Econometric Theory

John Stachurski

January 10, 2014

Contents

Preface

This is a quick course on modern econometric and statistical theory, focusing on fundamental ideas and general principles. The course is intended to be concise—suitable for learning concepts and results—rather than an encyclopedic treatment or a reference manual. It includes background material in probability and statistics that budding econometricians often need to build up.

Most of the topics covered here are standard, and I have borrowed ideas, results and exercises from many sources, usually without individual citations to that effect. Some of the large sample theory is new—although similar results have doubtless been considered elsewhere. The large sample theory has been developed to illuminate more clearly the kinds of conditions that allow the law of large numbers and central limit theorem to function, and to make large sample theory accessible to students without knowledge of measure theory.

The other originality is in the presentation of otherwise standard material. In my humble opinion, many of the mathematical arguments found here are neater, more precise and more insightful than other econometrics texts at a similar level.

Finally, the course also teaches programming techniques via an extensive set of examples, and a long discussion of the statistical programming environment R in the appendix. Even if only for the purpose of understanding theory, good programming skills are important. In fact, one of the best ways to understand a result in econometric theory is to first work your way through the proof, and then run a simulation which shows the theory in action.

These notes have benefitted from the input of many students. In particular, I wish to thank without implication Blair Alexander, Frank Cai, Yiyong Cai, Patrick Carvalho, Paul Kitney, Bikramaditya Datta, Stefan Webb and Varang Wiriyawit.

Part I

Background Material

Chapter 1

Probability

Probability theory forms the foundation stones of statistics and econometrics. If you want to be a first class statistician/econometrician, then every extra detail of probability theory that you can grasp and internalize will prove an excellent investment.

1.1 Probability Models

We begin with the basic foundations of probability theory. What follows will involve a few set operations, and you might like to glance over the results on set operations (and the definition of functions) in §13.1.

1.1.1 Sample Spaces

In setting up a model of probability, we usually start with the notion of a **sample space**, which, in general, can be any nonempty set, and is typically denoted Ω. We can think of Ω as being the collection of all possible outcomes in a random experiment. A typical element of Ω is denoted ω. The general idea is that a realization of uncertainty will lead to the selection of a particular $\omega \in \Omega$.

Example 1.1.1. Let $\Omega := \{1, \ldots, 6\}$ represent the six different faces of a dice. A realization of uncertainty corresponds to a roll of the dice, with the outcome being an integer ω in the set $\{1, \ldots, 6\}$.

The specification of all possible outcomes Ω is one part of our probability model. The other thing we need to do is to assign probabilities to outcomes. The obvious thing to do here is to assign a probability to every ω in Ω, but it turns out, for technical reasons beyond the scope of this text (see, e.g., Williams, 1991), that is is not the right way forward. Instead, the standard approach is to assign probabilities to *subsets* of Ω. In the language of probability theory, subsets of Ω are called **events**. The set of all events is usually denoted by $\mathcal{F}$, and we follow this convention.[1]

Example 1.1.2. Let Ω be any sample space. Two events we always find in $\mathcal{F}$ are Ω itself and the empty set $\emptyset$. (The empty set is regarded as being a subset of *every* set, and hence $\emptyset \subset \Omega$). In this context, Ω is called the certain event because it always occurs (regardless of which outcome ω is selected, $\omega \in \Omega$ is true by definition). The empty set $\emptyset$ is called the impossible event.

In all of what follows, if B is an event (i.e., $B \in \mathcal{F}$), then the notation $\mathbb{P}(B)$ will represent the probability that event B occurs. The way you should think about it is this: $\mathbb{P}(B)$ *represents the probability that when uncertainty is resolved and some* $\omega \in \Omega$ *is selected by "nature," the statement* $\omega \in B$ *is true.*

Example 1.1.3. Continuing example 1.1.1, let B be the event $\{1, 2\}$. The number $\mathbb{P}(B)$ represents the probability that the face ω selected by the roll is either 1 or 2.

The second stage of our model construction is to assign probabilities to elements of $\mathcal{F}$. In order to make sure our model of probability is well behaved, it's best to put certain restrictions on $\mathbb{P}$. (For example, we wouldn't want to have a B with $\mathbb{P}(B) = -93$, as negative probabilities don't make much sense.) These restrictions are imposed in the next definition:

A **probability** $\mathbb{P}$ on $(\Omega, \mathcal{F})$ is a function that associates to each event in $\mathcal{F}$ a number in $[0, 1]$, and, in addition, satisfies

1. $\mathbb{P}(\Omega) = 1$, and
2. Additivity: $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B)$ whenever $A, B \in \mathcal{F}$ with $A \cap B = \emptyset$.

[1] I'm skirting technical details here. In many common situations, we take $\mathcal{F}$ to be a proper subset of the set of all subsets of Ω. In particular, we exclude a few troublesome subsets of Ω from $\mathcal{F}$, because they are so messy that assigning probabilities to these sets cause problems for the theory. See §1.1.2 for more discussion.

Together, the triple $(\Omega, \mathcal{F}, \mathbb{P})$ is called a **probability space**. It describes a set of events and their probabilities for a given experiment. *Confession: I've simplified the standard definition of a probability space slightly—in ways that will never matter to us in this course—to avoid technical discussions that we don't need to go into. See §1.1.2 if you'd like to know more.*

Remark 1.1.1. Additivity of $\mathbb{P}$ was defined for two sets, but this implies additivity over any disjoint finite union. In particular, if $A_1, \ldots, A_J$ are disjoint in the sense that $A_i \cap A_j = \emptyset$ whenever $i \neq j$, then $\mathbb{P}(\cup_{j=1}^J A_j) = \sum_{j=1}^J \mathbb{P}(A_j)$ also holds. See exercise 1.5.1.

The axioms in the definition of $\mathbb{P}$ are fairly sensible. For starters, we are restricting the probability of any event to be between zero and one. Second, it is clear why we require $\mathbb{P}(\Omega) = 1$, since the realization ω will always be chosen from Ω by its definition. Third, the additivity property is natural: To find the probability of a given event, we can determine all the different (i.e., *disjoint*) ways that the event could occur, and then sum their probabilities.

Example 1.1.4. Continuing example 1.1.1, let $\Omega := \{1, \ldots, 6\}$ represent the six different faces of a dice, and, for $A \in \mathcal{F}$, let

$$\mathbb{P}(A) := \#A/6, \quad \text{where} \quad \#A := \text{ number of elements in } A \tag{1.1}$$

For example, given this definition of $\mathbb{P}$, we see that $\mathbb{P}\{2,4,6\} = 3/6 = 1/2$. It is simple to check that $\mathbb{P}$ is a probability on $(\Omega, \mathcal{F})$. Let's check additivity. Suppose that A and B are two disjoint subsets of $\{1, \ldots, 6\}$. In this case we must have $\#(A \cup B) = \#A + \#B$, since, by disjointness, the number of elements in the union is just the number contributed by A plus the number contributed by B. As a result,

$$\mathbb{P}(A \cup B) = \#(A \cup B)/6 = (\#A + \#B)/6 = \#A/6 + \#B/6 = \mathbb{P}(A) + \mathbb{P}(B)$$

The additivity property is very intuitive in this setting. For example, if we roll the dice, the probability of getting an even number is the probability of getting a 2 plus that of getting a 4 plus that of getting a 6. Formally,

$$\begin{aligned}\mathbb{P}\{2,4,6\} &= \mathbb{P}[\{2\} \cup \{4\} \cup \{6\}] \\ &= \mathbb{P}\{2\} + \mathbb{P}\{4\} + \mathbb{P}\{6\} = 1/6 + 1/6 + 1/6 = 1/2\end{aligned}$$

To finish off the proof that $\mathbb{P}$ is a probability on $(\Omega, \mathcal{F})$, it only remains to check that $0 \leq \mathbb{P}(B) \leq 1$ for any $B \in \mathcal{F}$, and that $\mathbb{P}(\Omega) = 1$. These details are left to you.

Example 1.1.5. Consider a memory chip in a computer, made up of billions of tiny switches. Imagine that a random number generator accesses a subset of N switches, setting each one to "on" or "off" at random. One sample space for this experiment is

$$\Omega_0 := \{(b_1, \ldots, b_N) : b_n \in \{\text{on, off}\} \text{ for each } n\}$$

Letting zero represent off and one represent on, we can also use the more practical space

$$\Omega := \{(b_1, \ldots, b_N) : b_n \in \{0, 1\} \text{ for each } n\}$$

Thus, Ω is the set of all binary sequences of length N. As our probability, we define

$$\mathbb{P}(A) := 2^{-N}(\#A)$$

To see that this is indeed a probability on $(\Omega, \mathcal{F})$ we need to check that $0 \leq \mathbb{P}(A) \leq 1$ for all $A \subset \Omega$, that $\mathbb{P}(\Omega) = 1$, and that $\mathbb{P}$ is additive. Exercise 1.5.7 asks you to confirm that $\mathbb{P}$ is additive. That $\mathbb{P}(\Omega) = 1$ follows from the fact that the number of binary sequences of length N is 2^N.

Now let's go back to the general case, where $(\Omega, \mathcal{F}, \mathbb{P})$ is an arbitrary probability space. From the axioms above, we can derive a suprising number of properties. Let's list the key ones, starting with the next fact.

Fact 1.1.1. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, and let $A, B \in \mathcal{F}$. If $A \subset B$, then

1. $\mathbb{P}(B \setminus A) = \mathbb{P}(B) - \mathbb{P}(A)$;
2. $\mathbb{P}(A) \leq \mathbb{P}(B)$;
3. $\mathbb{P}(A^c) := \mathbb{P}(\Omega \setminus A) = 1 - \mathbb{P}(A)$; and
4. $\mathbb{P}(\varnothing) = 0$.

These claims are not hard to prove. For example, regarding the part 1, if $A \subset B$, then we have $B = (B \setminus A) \cup A$. (Sketching the Venn diagram will help confirm this equality in your mind.) Since $B \setminus A$ and A are disjoint, additivity of $\mathbb{P}$ now gives

$$\mathbb{P}(B) = \mathbb{P}(B \setminus A) + \mathbb{P}(A) \qquad (\text{whenever } A \subset B)$$

This equality implies parts 1–4 of fact 1.1.1. Rearranging gives part 1, while nonnegativity of $\mathbb{P}$ gives part 2. Specializing to $B = \Omega$ gives part 3, and setting $B = A$ gives part 4.

The property that if $A \subset B$, then $\mathbb{P}(A) \leq \mathbb{P}(B)$ is called **monotonicity**, and is fundamental. If $A \subset B$, then we know that B occurs whenever A occurs (because if ω lands in A, then it also lands in B). Hence, the probability of B should be larger. Many crucial ideas in probability boil down to this one point.

Fact 1.1.2. If A and B are any (not necessarily disjoint) events, then

$$\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$$

In particular, for any $A, B \in \mathcal{F}$, we have $\mathbb{P}(A \cup B) \leq \mathbb{P}(A) + \mathbb{P}(B)$.

If A and B are events, then the **conditional probability of A given B** is

$$\mathbb{P}(A \,|\, B) := \mathbb{P}(A \cap B)/\mathbb{P}(B) \tag{1.2}$$

It represents the probability that A will occur, given the information that B has occurred. For the definition to make sense, it requires that $\mathbb{P}(B) > 0$. Events A and B are called **independent** if $\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$. If A and B are independent, then the conditional probability of A given B is just the probability of A.

Example 1.1.6. Consider an experiment where we roll a dice twice. A suitable sample space is the set of pairs (i, j), where i and j are between 1 and 6. The first element i represents the outcome of the first roll, while the second element j represents the outcome of the second roll. Formally,

$$\Omega := \{(i, j) : i, j \in \{1, \ldots, 6\}\}$$

For our probability, let's define $\mathbb{P}(E) := \#E/36$, where $\#E$ is the number of elements in $E \subset \Omega$. (In this case, elements are pairs, so $\#E$ is the number of pairs in E.) Now consider the events

$$A := \{(i, j) \in \Omega : i \text{ is even}\} \quad \text{and} \quad B := \{(i, j) \in \Omega : j \text{ is even}\}$$

In this case we have

$$A \cap B = \{(i, j) \in \Omega : i \text{ and } j \text{ are even}\}$$

With a bit of work we can verify that $\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$, indicating that A and B are independent under the probability $\mathbb{P}$. To check this, we need to be able to count the number of elements in A, B and $A \cap B$. The basic principle for counting ordered tuples is that the total number of possible tuples is the product of the number of possibilities for each tuple. For example, the number of distinct tuples

$$(i, j, k) \text{ where } i \in I, j \in J \text{ and } k \in K$$

is $(\#I) \times (\#J) \times (\#K)$. Hence, the number of elements in A is $3 \times 6 = 18$, the number of elements in B is $6 \times 3 = 18$, and the number of elements in $A \cap B$ is $3 \times 3 = 9$. As a result,

$$\mathbb{P}(A \cap B) = 9/36 = 1/4 = (18/36) \times (18/36) = \mathbb{P}(A)\mathbb{P}(B)$$

Thus, A and B are independent, as claimed.

A very useful result is the **law of total probability**, which says that if $A \in \mathcal{F}$ and $B_1, \ldots, B_M$ is a partition of Ω (i.e., $B_m \in \mathcal{F}$ for each m, the B_m's are mutually disjoint in the sense that $B_j \cap B_k$ is empty when $j \neq k$, and $\cup_{m=1}^{M} B_m = \Omega$) with $\mathbb{P}(B_m) > 0$ for all m, then

$$\mathbb{P}(A) = \sum_{m=1}^{M} \mathbb{P}(A \,|\, B_m) \cdot \mathbb{P}(B_m)$$

The proof is quite straightforward, although you should check that the manipulations of intersections and unions work if you have not seen them before:

$$\begin{aligned}\mathbb{P}(A) = \mathbb{P}[A \cap (\cup_{m=1}^{M} B_m)] &= \mathbb{P}[\cup_{m=1}^{M} (A \cap B_m)] \\ &= \sum_{m=1}^{M} \mathbb{P}(A \cap B_m) = \sum_{m=1}^{M} \mathbb{P}(A \,|\, B_m) \cdot \mathbb{P}(B_m)\end{aligned}$$

Example 1.1.7. Here's an informal example of the law of total probability: Suppose I flip a coin to decide whether to take part in a poker game. Being a bad player, the chance of losing money when I play is 2/3. The overall chance of losing money (LM) that evening is

$$\mathbb{P}\{\text{LM}\} = \mathbb{P}\{\text{LM} \,|\, \text{play}\}\mathbb{P}\{\text{play}\} + \mathbb{P}\{\text{LM} \,|\, \text{don't play}\}\mathbb{P}\{\text{don't play}\}$$

which is $(2/3) \times (1/2) + 0 \times (1/2) = 1/3$.

1.1.2 Technical Details

Okay, as alluded to above, in my presentation of probability spaces, I've swept some technical details under the carpet to make the presentation smooth. These details won't affect anything that follows, and this whole course can be completed successfully without knowing anything about them. Hence you can skip this section on first reading. Nevertheless, if you intend to keep going deeper into probability and statistics, eventually you will have to work your way through them. So let's note them for the record.

Assigning probabilities to all subsets of Ω in a consistent way can be quite a difficult task, so in practice we permit our set of events $\mathcal{F}$ to be a *sub*-collection of the subsets of Ω, and only assign probabilities to elements of $\mathcal{F}$. Thus, the first stage of our model construction is to choose (i) a sample space Ω, and (ii) a collection of its subsets $\mathcal{F}$ that we want to assign probabilities to.

When we choose $\mathcal{F}$, usually we don't just choose freely, because doing so will make it hard to form a consistent theory. One restriction we always put on $\mathcal{F}$ is to require that it contains the empty set $\emptyset$ and the whole set Ω. (In the definition of $\mathbb{P}$, we require that $\mathbb{P}(\Omega) = 1$. Hence we need Ω to be an element of $\mathcal{F}$, the events we assign probability to.)

Another sensible restriction concerns complements. For example, let's suppose that $A \in \mathcal{F}$, so that $\mathbb{P}(A)$ is well defined, and represents the "probability of event A." Now, given that we can assign a probability to the event A, it would be a bit odd if we couldn't assign a probability to the event "not A", which corresponds to A^c. So normally we require that if $A \in \mathcal{F}$, then $A^c \in \mathcal{F}$. When this is true, we say that $\mathcal{F}$ is "closed under the taking of complements".

Also, let's suppose that A and B are both in $\mathcal{F}$, so we assign probabilities to these events. In this case, it would be natural to think about the probability of the event "A and B", which corresponds to $A \cap B$. So we also require that if A and B are in $\mathcal{F}$, then $A \cap B$ is also in $\mathcal{F}$. We say that $\mathcal{F}$ is "closed under the taking of intersections."

Perhaps we should also require that if A and B are in $\mathcal{F}$, then $A \cup B$ is also in $\mathcal{F}$? Actually, we don't have to, because (see fact 13.1.1 on page 362),

$$A \cup B = (A^c \cap B^c)^c$$

Thus, if $\mathcal{F}$ is closed under the taking of complements and intersections, then $\mathcal{F}$ is automatically closed under the taking of unions.

There is one more restriction that's typically placed on $\mathcal{F}$, which is the property of being closed under "countable" unions. We won't go into details. Suffice to say that when $\mathcal{F}$ satisfies all these properties, it is called a "σ-algebra."

Finally, in standard probability theory, there is another restriction placed on $\mathbb{P}$ that I have not mentioned, called **countable additivity**. The definition of countable additivity is that if $A_1, A_2, \ldots$ is a disjoint sequence of sets in $\mathcal{F}$ (disjoint means that $A_i \cap A_j = \emptyset$ for any $i \neq j$), then

$$\mathbb{P}(\cup_i A_i) := \mathbb{P}\{\omega \in \Omega : \omega \in A_i \text{ for some } i\} = \sum_{i=1}^{\infty} \mathbb{P}(A_i)$$

Why strengthen additivity to countable additivity? Countable additivity works behind the scenes to make probability theory run smoothly (expectations operators are suitably continuous, and so on). None of these details will concern us in this course.

If you wish, you can learn all about σ-algebras and countable additivity in any text on measure theory. I recommend Williams (1991).

1.1.3 Random Variables

If you've done an elementary probability course, you might have been taught that a random variable is a "value that changes randomly," or something to that effect. To work more deeply with these objects, however, we need a sounder definition. For this reason, mathematicians define **random variables** to be functions from Ω into $\mathbb{R}$. Thus, in this formal model, *random variables convert outcomes in sample space into numerical outcomes.* This is useful, because numerical outcomes are easy to manipulate and interpret.[2]

To visualize the definition, consider a random variable x, and imagine that "nature" picks out an element ω in Ω according to some probability. The random variable now sends this ω into $x(\omega) \in \mathbb{R}$. In terms of example 1.1.5, a random number generator picks out an ω, which is a binary sequence. A random variable x converts this sequence into a real number. Depending on the conversion rule (i.e., depending on the definition of x), the outcome $x(\omega)$ might simulate a Bernoulli random variable, a uniform random variable, a normal random variable, etc.

Example 1.1.8. Recall example 1.1.5, with sample space

$$\Omega := \{(b_1, \ldots, b_N) : b_n \in \{0, 1\} \text{ for each } n\}$$

The set of events and probability were defined as follows:

$$\mathcal{F} := \text{ all subsets of } \Omega \quad \text{and} \quad \mathbb{P}(A) := 2^{-N}(\#A)$$

Consider a random variable x on Ω that returns the first element of any given sequence. That is,

$$x(\omega) = x(b_1, \ldots, b_N) = b_1$$

[2] I'm skipping some technical details again. The definition of random variables on infinite Ω is actually a bit more subtle. In practice, when identifying random variables with the class of all functions from Ω to $\mathbb{R}$, we typically exclude some particularly nasty (i.e., complicated and generally badly behaved) functions. The remaining "nice" functions are our random variables. In this course we will never meet the nasty functions, and there's no need to go into further details. Those who want to know more should consult any text on measure theory (e.g., Williams, 1991).

Then x is a Bernoulli random variable (i.e., x takes only the values zero and one). The probability that $x = 1$ is $1/2$. Indeed,

$$\begin{aligned}\mathbb{P}\{x = 1\} &:= \mathbb{P}\{\omega \in \Omega : x(\omega) = 1\} \\ &= \mathbb{P}\{(b_1, \ldots, b_N) : b_1 = 1\} \\ &= 2^{-N} \times \#\{(b_1, \ldots, b_N) : b_1 = 1\}\end{aligned}$$

The number of length N binary sequences with $b_1 = 1$ is 2^{N-1}, so $\mathbb{P}\{x = 1\} = 1/2$.

Example 1.1.9. Consider the sample space

$$\Omega := \{(b_1, b_2, \ldots) : b_n \in \{0, 1\} \text{ for each } n\}$$

Ω is called the set of all infinite binary sequences. (This is an infinite version of the sample space in example 1.1.5. Imagine a computer with an infinite amount of memory.) Consider an experiment where we flip a coin until we get a "heads". We let 0 represent tails and 1 represent heads. The experiment of flipping until we get a heads can be modeled with the random variable

$$x(\omega) = x(b_1, b_2, \ldots) = \min\{n : b_n = 1\}$$

As per the definition, x is a well-defined function from Ω into $\mathbb{R}$.[3]

Let's go back to the general case, with arbitrary probability space $(\Omega, \mathcal{F}, \mathbb{P})$, and talk a bit more about Bernoulli (i.e., binary) random variables. There is a generic way to create Bernoulli random variables, using **indicator functions**. If Q is a statement, such as "on the planet Uranus, there exists a tribe of three-headed monkeys," then $\mathbb{1}\{Q\}$ is considered as equal to one when the statement Q is true, and zero when the statement Q is false. Hence, $\mathbb{1}\{Q\}$ is a binary indicator of the truth of the statement Q.

In general (in fact always), a **Bernoulli random variable** has the form

$$x(\omega) = \mathbb{1}\{\omega \in C\}$$

where C is some subset of Ω. Thus, x is a binary random variable indicating whether or not the event C occurs (one means "yes" and zero means "no").

[3] Actually, that's not strictly true. What if $\omega = \omega_0$ is an infinite sequence containing only zeros? Then $\{n : b_n = 1\} = \emptyset$. The convention here is to set $x(\omega_0) = \min\{n : b_n = 1\} = \min \emptyset = \infty$. But then x is not a map into $\mathbb{R}$, because it can take the value ∞. However, it turns out that this event has probability zero, and hence we can ignore it. For example, we can set $x(\omega_0) = 0$ without changing anything significant. Now we're back to a well-defined function from Ω to $\mathbb{R}$.

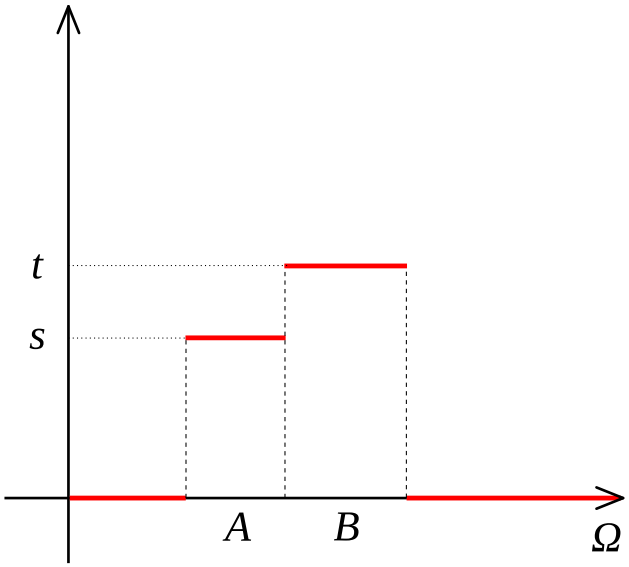


Figure 1.1: Simple function $x(\omega) = s\mathbb{1}\{\omega \in A\} + t\mathbb{1}\{\omega \in B\}$

From Bernoulli random variables we can create "discrete" random variables. A **discrete random variable** is a random variable with finite range.[4] We can create discrete random variables by taking "linear combinations" of Bernoulli random variables. (A linear combination of certain elements is formed by multiplying these elements by scalars and then summing the result.) For example, let A and B be disjoint subsets of Ω. The random variable

$$x(\omega) = s\mathbb{1}\{\omega \in A\} + t\mathbb{1}\{\omega \in B\} \tag{1.3}$$

is a discrete random variable taking the value s when ω falls in A, t when ω falls in B, and zero otherwise. (Check it!) Figure 1.1 shows a graph of x when $\Omega = \mathbb{R}$.

It turns out that *any* discrete random variable can be created by taking linear combinations of Bernoulli random variables. In particular, we can also define a discrete random variable as a random variable having the form

$$x(\omega) = \sum_{j=1}^{J} s_j \mathbb{1}\{\omega \in A_j\} \tag{1.4}$$

We will work with this expression quite a lot. In doing so, we will always assume that

[4] See §13.1.1 for the definition of "range." Our usage is not entirely standard, in that many texts call random variables with countably infinite range discrete as well.

- the scalars $s_1, \ldots, s_J$ are distinct, and
- the sets $A_1, \ldots, A_J$ form partition of Ω.[5]

Given these assumptions, we then have

- $x(\omega) = s_j$ if and only if $\omega \in A_j$.
- $\{x = s_j\} = A_j$.
- $\mathbb{P}\{x = s_j\} = \mathbb{P}(A_j)$.

The second two statements follow from the first. Convince yourself of these results before continuing.

Before finishing this section, let's clarify a common notational convention with random variables that we've adopted above and that will be used below. With a random variable x, we often write

$$\{x \text{ has some property}\}$$

as a shorthand for

$$\{\omega \in \Omega : x(\omega) \text{ has some property}\}$$

We'll follow this convention, but you should translated it backwards and forwards in your mind if you're not yet familiar with the notation. To give you an example, consider the claim that, for any random variable x,

$$\mathbb{P}\{x \leq a\} \leq \mathbb{P}\{x \leq b\} \quad \text{whenever} \quad a \leq b \tag{1.5}$$

This is intuitively obvious. The mathematical argument goes as follows: Observe that

$$\{x \leq a\} := \{\omega \in \Omega : x(\omega) \leq a\} \subset \{\omega \in \Omega : x(\omega) \leq b\} =: \{x \leq b\}$$

(The inclusion $\subset$ must hold, because if ω is such that $x(\omega) \leq a$, then, since $a \leq b$, we also have $x(\omega) \leq b$. Hence any ω in the left-hand side is also in the right-hand side.) The result in (1.5) now follows from monotonicity of $\mathbb{P}$ (fact 1.1.1 on page 5).

[5] That is, $A_i \cap A_j = \emptyset$ when $i \neq j$, and $\cup_j A_j = \Omega$.

1.1.4 Expectations

Our next task is to define expectations for an arbitrary random variable x on probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Roughly speaking, $\mathbb{E}[x]$ is defined as the "sum" of all possible values of x, weighted by their probabilities. (Here "sum" is in quotes because there may be an infinite number of possibilities.) The expectation $\mathbb{E}[x]$ of x also represents the "average" value of x over a "very large" sample. (This is a theorem, not a definition—see the law of large numbers below.)

Let's start with the definition when x is a discrete random variable. Let x be the discrete random variable $x(\omega) = \sum_{j=1}^{J} s_j \mathbb{1}\{\omega \in A_j\}$, as defined in (1.4). In this case, the **expectation** of x is defined as

$$\mathbb{E}[x] := \sum_{j=1}^{J} s_j \mathbb{P}(A_j) \tag{1.6}$$

The definition is completely intuitive: For this x, given our assumption that the sets A_j's are a partition of Ω and the s_j's are distinct, we have

$$A_j = \{x = s_j\} := \{\omega \in \Omega : x(\omega) = s_j\}$$

Hence, (1.6) tells us that

$$\mathbb{E}[x] = \sum_{j=1}^{J} s_j \mathbb{P}\{x = s_j\}$$

Thus, the expectation is the sum of the different values that x may take, weighted by their probabilities.

How about arbitrary random variables, with possibly infinite range? Unfortunately, the full definition of expectation for these random variables involves measure theory, and we can't treat in in detail. But the short story is that any arbitrary random variable x can be approximated by a sequence of discrete variables x_n. The expectation of discrete random variables was defined in (1.6). The expectation of the limit x is then defined as

$$\mathbb{E}[x] := \lim_{n \to \infty} \mathbb{E}[x_n] \tag{1.7}$$

When things are done carefully (details omitted), this value doesn't depend on the particular approximating sequence $\{x_n\}$, and hence the value $\mathbb{E}[x]$ is well defined.

Let's list some facts about expectation, and then discuss them one by one.

Fact 1.1.3. Expectation of indicators equals probability of event: For any $A \in \mathcal{F}$,

$$\mathbb{E}\left[\mathbb{1}\{\omega \in A\}\right] = \mathbb{P}(A) \tag{1.8}$$

Fact 1.1.4. Expectation of a constant is the constant: If $\alpha \in \mathbb{R}$, then $\mathbb{E}\left[\alpha\right] = \alpha$.

Fact 1.1.5. Linearity: If x and y are random variables and α and β are constants, then

$$\mathbb{E}\left[\alpha x + \beta y\right] = \alpha\mathbb{E}\left[x\right] + \beta\mathbb{E}\left[y\right]$$

Fact 1.1.6. Monotonicity: If x, y are random variables and $x \leq y$,[6] then $\mathbb{E}\left[x\right] \leq \mathbb{E}\left[y\right]$.

Fact 1.1.3 follows from the definition in (1.6). We have

$$\mathbb{1}\{\omega \in A\} = 1 \times \mathbb{1}\{\omega \in A\} + 0 \times \mathbb{1}\{\omega \in A^c\}$$

Applying (1.6), we get

$$\mathbb{E}\left[\mathbb{1}\{\omega \in A\}\right] = 1 \times \mathbb{P}(A) + 0 \times \mathbb{P}(A^c) = \mathbb{P}(A)$$

Fact 1.1.4 say that the expectation of a constant α is just the value of the constant. The idea here is that the constant α should be understood in this context as the constant random variable $\alpha\mathbb{1}\{\omega \in \Omega\}$. From our definition (1.6), the expectation of this "constant" is indeed equal to its value α:

$$\mathbb{E}\left[\alpha\right] := \mathbb{E}\left[\alpha\mathbb{1}\{\omega \in \Omega\}\right] = \alpha\mathbb{P}(\Omega) = \alpha$$

Now let's think about linearity (fact 1.1.5). The way this is proved, is to first prove linearity for discrete random variables, and then extend the proof to arbitrary random variables via (1.7). We'll omit the last step, which involves measure theory. We'll also omit the full proof for discrete random variables, since it's rather long. Instead, let's cover a quick sketch of the argument that still provides most of the intuition.

Suppose we take the random variable x in (1.4) and double it, producing the new random variable $y = 2x$. More precisely, for each ω, we set $y(\omega) = 2x(\omega)$. (Whatever happens with x, we're going to double it and y will return that value.) In that case, we have $\mathbb{E}\left[y\right] = 2\mathbb{E}\left[x\right]$. To see this, observe that

$$y(\omega) = 2x(\omega) = 2\left[\sum_{j=1}^{J} s_j \mathbb{1}\{\omega \in A_j\}\right] = \sum_{j=1}^{J} 2s_j \mathbb{1}\{\omega \in A_j\}$$

[6] The statement $x \leq y$ means that x is less than y for any realization of uncertainty. Formally, it means that $x(\omega) \leq y(\omega)$ for all $\omega \in \Omega$.

Hence, applying (1.6),

$$\mathbb{E}[y] = \sum_{j=1}^{J} 2s_j \mathbb{P}(A_j) = 2\left[\sum_{j=1}^{J} s_j \mathbb{P}(A_j)\right] = 2\mathbb{E}[x]$$

What we have shown is that $\mathbb{E}[2x] = 2\mathbb{E}[x]$. Looking back over our argument, we can see that there is nothing special about the number 2 here—we could have used any constant. In other words,

$$\text{For any constant } \gamma, \text{ we have } \mathbb{E}[\gamma x] = \gamma \mathbb{E}[x]$$

Another aspect of linearity of expectations is additivity, which says that given random variables x and y, the statement $\mathbb{E}[x+y] = \mathbb{E}[x] + \mathbb{E}[y]$ is always true. Instead of giving the full proof, let's show this for the Bernoulli random variables

$$x(\omega) = \mathbb{1}\{\omega \in A\} \quad \text{and} \quad y(\omega) = \mathbb{1}\{\omega \in B\} \tag{1.9}$$

Consider the sum $x+y$. By this, we mean the random variable $z(\omega) = x(\omega) + y(\omega)$. More succinctly, $(x+y)(\omega) := x(\omega) + y(\omega)$. We claim that $\mathbb{E}[x+y] = \mathbb{E}[x] + \mathbb{E}[y]$. To see that this is the case, note first that

$$(x+y)(\omega) = \mathbb{1}\{\omega \in A \setminus B\} + \mathbb{1}\{\omega \in B \setminus A\} + 2\mathbb{1}\{\omega \in A \cap B\}$$

(To check this, just go through the different cases for ω, and verify that the right hand side of this expression agrees with $x(\omega) + y(\omega)$. Sketching a Venn diagram will help.) Therefore, by the definition of expectation,

$$\mathbb{E}[x+y] = \mathbb{P}(A \setminus B) + \mathbb{P}(B \setminus A) + 2\mathbb{P}(A \cap B) \tag{1.10}$$

Now observe that

$$A = (A \setminus B) \cup (A \cap B)$$

It follows (why?) that

$$\mathbb{E}[x] := \mathbb{P}(A) = \mathbb{P}(A \setminus B) + \mathbb{P}(A \cap B)$$

Performing a similar calculation with y produces

$$\mathbb{E}[y] := \mathbb{P}(B) = \mathbb{P}(B \setminus A) + \mathbb{P}(A \cap B)$$

Adding these two produces the value on the right-hand side of (1.10), and we have now confirmed that $\mathbb{E}[x+y] = \mathbb{E}[x] + \mathbb{E}[y]$.

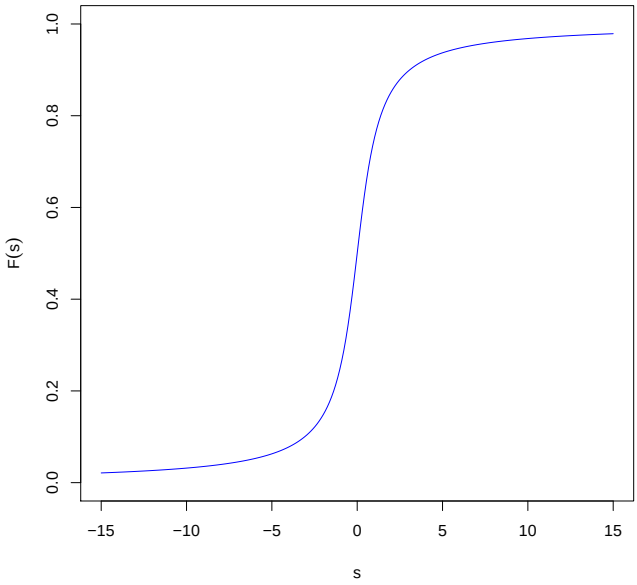


Figure 1.2: Cauchy cdf

1.2 Distributions

All random variables have distributions. Distributions summarize the probabilities of different outcomes for the random variable in question, and allow us to compute expectations. In this section, we describe the link between random variables and distributions.

1.2.1 CDFs

A **cumulative distribution function** (cdf) on $\mathbb{R}$ is a right-continuous, monotone increasing function $F\colon \mathbb{R} \to [0,1]$ satisfying $\lim_{s\to-\infty} F(s) = 0$ and $\lim_{s\to\infty} F(s) = 1$. ($F$ is monotone increasing if $F(s) \leq F(s')$ whenever $s \leq s'$, and right continuous if $F(s_n) \downarrow F(s)$ whenever $s_n \downarrow s$.)

Example 1.2.1. The function $F(s) = \arctan(s)/\pi + 1/2$ is a cdf—one variant of the Cauchy distribution. A plot is given in figure 1.2.

Let x be a random variable on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$, and consider the function

$$F_x(s) := \mathbb{P}\{x \leq s\} := \mathbb{P}\{\omega \in \Omega : x(\omega) \leq s\} \qquad (s \in \mathbb{R}) \tag{1.11}$$

It turns out that this function is *always a cdf*.[7] We say that F_x **is the cdf of** x, or, alternatively, that F_x **is the distribution of** x, and write $x \sim F_x$.

We won't go through the proof that the function F_x defined by (1.11) is a cdf. Note however that monotonicity is immediate from (1.5) on page 12.

Fact 1.2.1. If $x \sim F$, then $\mathbb{P}\{a < x \leq b\} = F(b) - F(a)$ for any $a \leq b$.

Proof: If $a \leq b$, then $\{a < x \leq b\} = \{x \leq b\} \setminus \{x \leq a\}$ and $\{x \leq a\} \subset \{x \leq b\}$. Applying fact 1.1.1 on page 5 gives the desired result.

A cdf F is called **symmetric** if $F(-s) = 1 - F(s)$ for all $s \in \mathbb{R}$.[8] The proof of the next fact is an exercise (exercise 1.5.12).

Fact 1.2.2. Let F be a cdf and let $x \sim F$. If F is symmetric and $\mathbb{P}\{x = s\} = 0$ for all $s \in \mathbb{R}$, then the distribution $F_{|x|}$ of the absolute value $|x|$ is given by

$$F_{|x|}(s) := \mathbb{P}\{|x| \leq s\} = 2F(s) - 1 \qquad (s \geq 0)$$

1.2.2 Densities and Probability Mass Functions

Cdfs are important because every random variable has a well-defined cdf via (1.11). However, they can be awkward to manipulate mathematically, and plotting cdfs is not a very good way to convey information about probabilities. For example, consider figure 1.2. The amount of probability mass in different regions of the x-axis is determined by the slope of the cdf. Research shows that humans are poor at extracting quantitative information from slopes. They do much better with *heights*, which leads us into our discussion of densities and probability mass functions.

Densities and probability mass functions correspond to two different, mutually exclusive cases. The first (density) case arises when the increase of the cdf in question is smooth, and contains no jumps. The second (probability mass function) case

[7] The following is also true: For every cdf F, there exists a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and a random variable $x\colon \Omega \to \mathbb{R}$ such that the distribution of x is F. Exercise 1.5.14 gives some hints on how the construction works.

[8] Thus, the probability that $x \leq -s$ is equal to the probability that $x > s$. Centered normal distributions and t-distributions have this property.

arises when the increase consists of jumps alone. Let's have a look at these two situations, starting with the second case.

The pure jump case occurs when the cdf represents a discrete random variable. To understand this, suppose that x takes values $s_1, \ldots, s_J$. Let $p_j := \mathbb{P}\{x = s_j\}$. We then have $0 \leq p_j \leq 1$ for each j, and $\sum_{j=1}^J p_j = 1$ (exercise 1.5.13). A finite collection of numbers $p_1, \ldots, p_J$ such that $0 \leq p_j \leq 1$ and $p_1 + \cdots + p_J = 1$ is called a **probability mass function** (pmf). The cdf corresponding to this random variable is

$$F_x(s) = \sum_{j=1}^J \mathbb{1}\{s_j \leq s\} p_j \tag{1.12}$$

How do we arrive at this expression? Because, for this random variable,

$$\mathbb{P}\{x \leq s\} = \mathbb{P}\left\{ \bigcup_{j \text{ s.t. } s_j \leq s} \{x = s_j\} \right\} = \sum_{j \text{ s.t. } s_j \leq s} \mathbb{P}\{x = s_j\} = \sum_{j=1}^J \mathbb{1}\{s_j \leq s\} p_j$$

Visually, F_x is a step function, with a jump up of size p_j at point s_j. Figure 1.3 gives an example with $J = 2$.

The other case of interest is the density case. A **density** is a nonnegative function p that integrates to 1. For example, suppose that F is a smooth cdf, so that the derivative F' exists. Let $p := F'$. By the fundamental theorem of calculus, we then have

$$\int_r^s p(t)dt = \int_r^s F'(t)dt = F(s) - F(r)$$

From the definition of cdfs, we can see that p is nonnegative and $\int_{-\infty}^{+\infty} p(s)ds = 1$. In other words, p is a density. Also, taking the limit as $r \to -\infty$ we obtain

$$F(s) = \int_{-\infty}^s p(t)dt$$

which tells us that F can be recovered from p.

More generally, if F is the cdf of random variable x and $p \colon \mathbb{R} \to [0, \infty)$ satisfies

$$F(s) = \int_{-\infty}^s p(t)dt \qquad \text{for all } s \in \mathbb{R}$$

then p is called the **density of** x.

Not every random variable has a density. The exact necessary and sufficient condition for a density to exist is that F is "absolutely continuous." The most important special

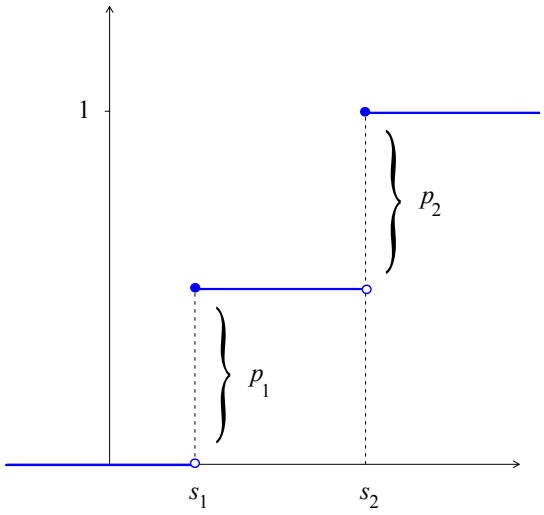


Figure 1.3: Discrete cdf

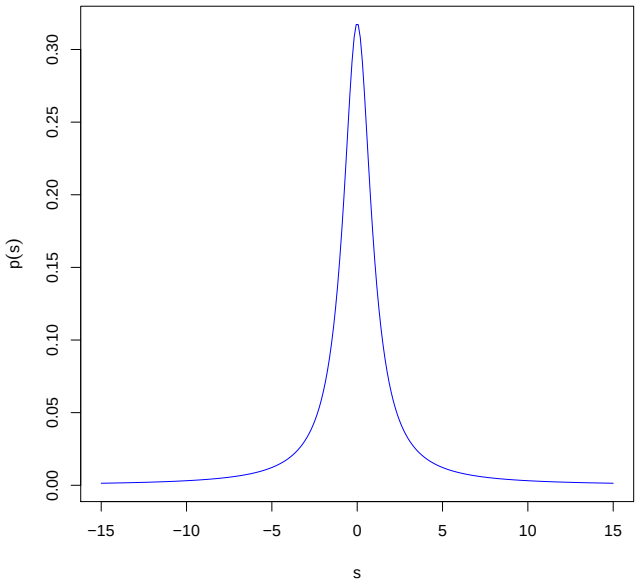


Figure 1.4: Cauchy density

case of absolute continuity is when F is differentiable. (See any text on measure theory for details.) For our purposes, you can think of absolute continuity meaning that F does not have any jumps. In particular, discrete random variables are out, as is any random variable putting positive probability mass on a single point.[9]

Fact 1.2.3. If x has a density, then $\mathbb{P}\{x = s\} = 0$ for all $s \in \mathbb{R}$.

As discussed above, cdfs are useful because every random variable has one, but pmfs and densities are nicer to work with, and visually more informative. For example, consider figure 1.4, which shows the density corresponding to the Cauchy cdf in figure 1.2. Information about probability mass is now conveyed by height rather than slope, which is easier for us humans to digest.

1.2.3 The Quantile Function

Let F be any cdf on $\mathbb{R}$. Suppose that F is strictly increasing, so that the inverse function F^{-1} exists:

$$F^{-1}(q) := \text{ the unique } s \text{ such that } F(s) = q \qquad (0 < q < 1) \tag{1.13}$$

The inverse of the cdf is called the **quantile function**, and has many applications in probability and statistics.

Example 1.2.2. The quantile function associated with the Cauchy cdf in example 1.2.1 is $F^{-1}(q) = \tan[\pi(q - 1/2)]$. See figure 1.5.

Things are a bit more complicated when F is not strictly increasing, as the inverse F^{-1} is not well defined. (If F is not strictly increasing, then there exists at least two distinct points s and s' such that $F(s) = F(s')$.) This problem is negotiated by setting

$$F^{-1}(q) := \inf\{s \in \mathbb{R} : F(s) \geq q\} \qquad (0 < q < 1)$$

This expression is a bit more complicated, but in the case where F is strictly increasing, it reduces to (1.13).

The value $F^{-1}(1/2)$ is called the **median** of F.

[9] In elementary texts, random variables with densities are often called "continuous random variables." The notation isn't great, because "continuous" here has nothing to do with the usual definition of continuity of functions.

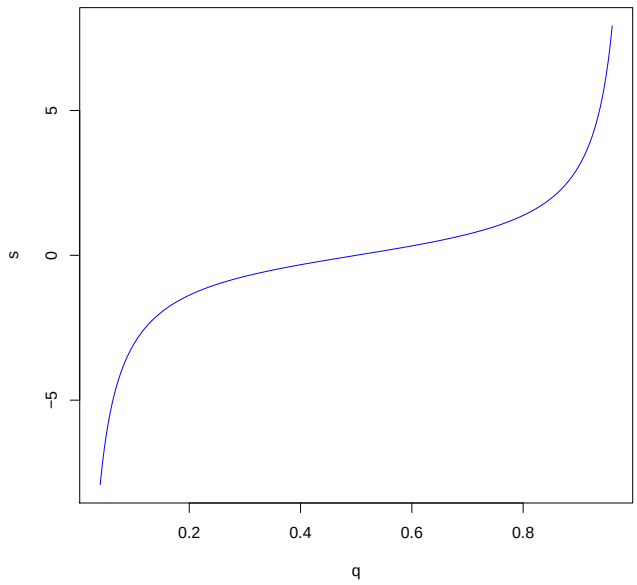


Figure 1.5: Cauchy quantile function

The quantile function features in hypothesis testing, where it can be used to define critical values (see §5.3). An abstract version of the problem is as follows: Let $x \sim F$, where F is strictly increasing, differentiable (so that a density exists and x puts no probability mass on any one point) and symmetric. Given $\alpha \in (0,1)$, we want to find the c such that $\mathbb{P}\{-c \leq x \leq c\} = 1 - \alpha$ (see figure 1.6). The solution is given by $c := F^{-1}(1 - \alpha/2)$. That is,

$$c = F^{-1}(1 - \alpha/2) \implies \mathbb{P}\{|x| \leq c\} = 1 - \alpha \tag{1.14}$$

To see this, fix $\alpha \in (0,1)$. From fact 1.2.2, we have

$$\mathbb{P}\{|x| \leq c\} = 2F(c) - 1 = 2F[F^{-1}(1 - \alpha/2)] - 1 = 1 - \alpha$$

In the case where F is the standard normal cdf Φ, this value c is usually denoted by $z_{\alpha/2}$. We will adopt the same notation:

$$z_{\alpha/2} := \Phi^{-1}(1 - \alpha/2) \tag{1.15}$$

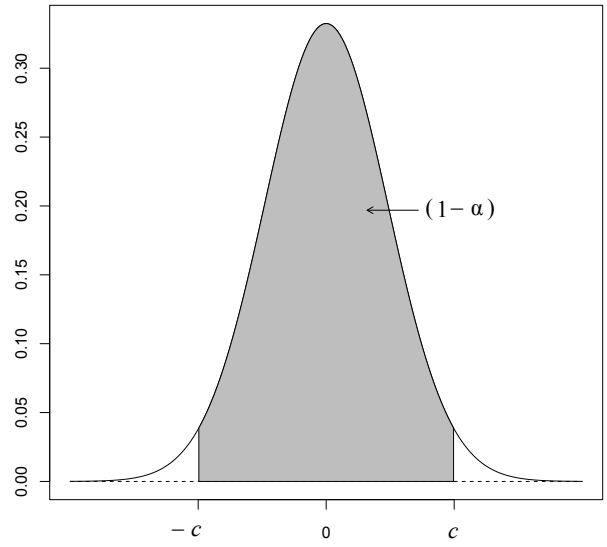


Figure 1.6: Finding critical values

1.2.4 Expectations from Distributions

Until now, we've been calculating expectations using the expectation operator $\mathbb{E}$, which was defined from a given probability $\mathbb{P}$ in §1.1.4. One of the most useful facts about distributions is that one need not know about x or $\mathbb{E}$ to calculate $\mathbb{E}[x]$—knowledge of the distribution of x is sufficient.

Mathematically, this is an interesting topic, and a proper treatment requires measure theory (see, e.g., Williams, 1991). Here I'll just tell you what you need to know for the course, and then follow that up with a bit of intuition.

In all of what follows, h is an arbitrary function from $\mathbb{R}$ to $\mathbb{R}$.

Fact 1.2.4. If x is a discrete random variable taking values $s_1, \ldots, s_J$ with probabilities $p_1, \ldots, p_J$, then

$$\mathbb{E}[h(x)] = \sum_{j=1}^{J} h(s_j) p_j \tag{1.16}$$

Fact 1.2.5. If x is a "continuous" random variable with density p, then

$$\mathbb{E}[h(x)] = \int_{-\infty}^{\infty} h(s) p(s) ds \tag{1.17}$$

It's convenient to have a piece of notation that captures both of these cases. As a result, if $x \sim F$, then we will write

$$\mathbb{E}\left[h(x)\right] = \int h(s)F(ds)$$

The way you should understand this expression is that when F is differentiable with derivative $p = F'$, then $\int h(s)F(ds)$ is defined as $\int_{-\infty}^{\infty} h(s)p(s)ds$. If, on the other hand, F is the step function $F(s) = \sum_{j=1}^{J} \mathbb{1}\{s_j \leq s\}p_j$ corresponding to the discrete random variable in fact 1.2.4, then $\int h(s)F(ds)$ is defined as $\sum_{j=1}^{J} h(s_j)p_j$.

Example 1.2.3. Suppose that $h(x) = x^2$. In this case, $\mathbb{E}\left[h(x)\right]$ is the second moment of x. If we know the density p of x, then fact 1.2.5 can be used to evaluate that second moment by solving the integral on the right-hand side of (1.17).

Just for the record, let me note that if you learn measure theory you will come to understand that, for a given cdf F, the expression $\int h(s)F(ds)$ has its own precise definition, as the "Lebesgue-Stieltjes" integral of h with respect to F. In the special case where F is differentiable with $p = F'$, one can *prove* that $\int h(s)F(ds) = \int_{-\infty}^{\infty} h(s)p(s)ds$, where the left hand side is the Lebesgue-Stieltjes integral, and the right hand side is the ordinary (Riemann) integral you learned in high school. An similar statement holds for the discrete case. However, this is not the right place for a full presentation of the Lebesgue-Stieltjes integral. We want to move on to statistics.

Although we're skipping a lot of technical details here, we can at least prove fact 1.2.4. This is the discrete case, where x is of the form $x(\omega) = \sum_{j=1}^{J} s_j \mathbb{1}\{\omega \in A_j\}$, and $p_j := \mathbb{P}\{x = s_j\} = \mathbb{P}(A_j)$. As usual, the values $\{s_j\}$ are distinct and the sets $\{A_j\}$ are a partition of Ω. As we saw in §1.1.4, the expectation is

$$\mathbb{E}\left[x\right] = \sum_{j=1}^{J} s_j \mathbb{P}(A_j) = \sum_{j=1}^{J} s_j \mathbb{P}\{x = s_j\} = \sum_{j=1}^{J} s_j p_j$$

Now let $h\colon \mathbb{R} \to \mathbb{R}$. You should be able to convince yourself that

$$h(x(\omega)) = \sum_{j=1}^{J} h(s_j)\mathbb{1}\{\omega \in A_j\}$$

(Pick an arbitrary A_j and check that the left- and right-hand sides are equal when $\omega \in A_j$.) This is a discrete random variable, which we can take the expectation of

using (1.6) (page 13). We get

$$\mathbb{E}[h(x)] = \sum_{j=1}^{J} h(s_j)\mathbb{P}(A_j) = \sum_{j=1}^{J} h(s_j)p_j$$

Fact 1.2.4 is now confirmed.

1.2.5 Common Distributions

Let's list a few well-known distributions that will come up in this course.

Let $a < b$. The **uniform distribution** on interval $[a, b]$ is the distribution associated with the density

$$p(s; a, b) := \frac{1}{b-a} \qquad (a \leq s \leq b)$$

(If $s < a$ or $s > b$, then $p(s; a, b) := 0$.) The mean is

$$\int_a^b s\, p(s; a, b) ds = \frac{a+b}{2}$$

The **univariate normal density** or **Gaussian density** is a function p of the form

$$p(s) := p(s; \mu, \sigma) := (2\pi\sigma^2)^{-1/2} \exp\left\{-\frac{1}{2}(s-\mu)^2\sigma^{-2}\right\}$$

for some $\mu \in \mathbb{R}$ and $\sigma > 0$. We represent this distribution symbolically by $\mathcal{N}(\mu, \sigma^2)$. The distribution $\mathcal{N}(0, 1)$ is called the **standard normal distribution**

It is well-known that if $x \sim \mathcal{N}(\mu, \sigma^2)$, then $\mathbb{E}[x] = \mu$, and $\text{var}[x] = \sigma^2$. Hence the two parameters separately define the mean and the variance (or standard deviation), and this is one of many attractive features of the distribution.

Fact 1.2.6. If $x_1, \ldots, x_N$ are normally distributed and $\alpha_0, \ldots, \alpha_N$ are any constants, then $\alpha_0 + \sum_{n=1}^{N} \alpha_n x_n$ is also normally distributed.

The **chi-squared distribution with k degrees of freedom** is the distribution with density

$$p(s; k) := \frac{1}{2^{k/2}\Gamma(k/2)} s^{k/2-1} e^{-s/2} \qquad (s \geq 0)$$

where Γ is the Gamma function (details omitted). If x has a distribution described by this density, then we write $x \sim \chi^2(k)$.

Student's t-distribution with k degrees of freedom, or, more simply, the t-distribution with k degrees of freedom, is the distribution on $\mathbb{R}$ with density

$$p(s;k) := \frac{\Gamma(\frac{k+1}{2})}{(k\pi)^{1/2}\Gamma(\frac{k}{2})}\left(1+\frac{s^2}{k}\right)^{-(k+1)/2}$$

The **F-distribution** with parameters k_1, k_2 is the distribution with the unlikely looking density

$$p(s;k_1,k_2) := \frac{\sqrt{(k_1 s)^{k_1} k_2^{k_2} / [k_1 s + k_2^{k_1+k_2}]}}{sB(k_1/2, k_2/2)} \qquad (s \geq 0)$$

where B is the Beta function (details omitted). The F-distribution arises in certain hypothesis tests, some of which we will examine later.

1.3 Dependence

[roadmap]

1.3.1 Joint Distributions

Consider a collection of N random variables $x_1, \ldots, x_N$. For each individual random variable $x_n \colon \Omega \to \mathbb{R}$, the distribution F_n of x_n is

$$F_n(s) := \mathbb{P}\{x_n \leq s\} \qquad (-\infty < s < \infty) \tag{1.18}$$

This distribution tells us about the random properties of x_n viewed as a single entity. But we often want to know about the relationships between the variables $x_1, \ldots, x_N$, and outcomes for the group of variables as a whole. To quantify these things, we define the **joint distribution** of $x_1, \ldots, x_N$ to be

$$F(s_1, \ldots, s_N) := \mathbb{P}\{x_1 \leq s_1, \ldots, x_N \leq s_N\} \qquad (-\infty < s_n < \infty; n = 1, \ldots, N)$$

In this setting, the distribution F_n of x_n is sometimes called the **marginal distribution**, in order to distinguish it from the joint distribution.

The **joint density** of $x_1, \ldots, x_N$, if it exists, is a function $p \colon \mathbb{R}^N \to [0, \infty)$ satisfying

$$\int_{-\infty}^{t_N} \cdots \int_{-\infty}^{t_1} p(s_1, \ldots, s_N) ds_1 \cdots ds_N = F(t_1, \ldots, t_N) \tag{1.19}$$

for all $t_n \in \mathbb{R}$, $n = 1, \ldots, N$.

Typically, the joint distribution cannot be determined from the N marginal distributions alone, since the marginals do not tell us about the interactions between the different variables. Once special case where we can tell the joint from the marginals is when there is no interaction. This is called independence, and we treat it in the next section.

From joint densities we can construct conditional densities. The **conditional density** of $x_{k+1}, \ldots, x_N$ given $x_1 = s_1, \ldots, x_k = s_k$ is defined by

$$p(s_{k+1}, \ldots, s_N \,|\, s_1, \ldots, s_k) := \frac{p(s_1, \ldots, s_N)}{p(s_1, \ldots, s_k)} \tag{1.20}$$

Rearranging this expression we obtain a decomposition of the joint density:

$$p(s_1, \ldots, s_N) = p(s_{k+1}, \ldots, s_N \,|\, s_1, \ldots, s_k) p(s_1, \ldots, s_k) \tag{1.21}$$

This decomposition is useful in many situations.

1.3.2 Independence

Let $x_1, \ldots, x_N$ be a collection of random variables with $x_n \sim F_n$, where F_n is a cdf. The variables $x_1, \ldots, x_N$ are called **identically distributed** $F_n = F_m$ for all n, m. They are called **independent** if, given any $s_1, \ldots, s_N$, we have

$$\mathbb{P}\{x_1 \leq s_1, \ldots, x_N \leq s_N\} = \mathbb{P}\{x_1 \leq s_1\} \times \cdots \times \mathbb{P}\{x_N \leq s_N\} \tag{1.22}$$

Equivalently, if F is the joint distribution of $x_1, \ldots, x_N$ and F_n is the marginal distribution of x_n, then independence states that

$$F(s_1, \ldots, s_N) = F_1(s_1) \times \cdots \times F_N(s_N) = \prod_{n=1}^{N} F_n(s_n)$$

We use the abbreviation IID for collections of random variables that are both independent and identically distributed.

Example 1.3.1. Consider a monkey throwing darts at a dartboard. Let x denote the horizontal location of the dart relative to the center of the board, and let y denote the vertical location. (For example, if $x = -1$ and $y = 3$, then the dart is 1cm to the left of the center, and 3cm above.) At first pass, we might suppose that x and y are independent and identically distributed.

Fact 1.3.1. If $x_1, \ldots, x_M$ are independent and $\mathbb{E}|x_m|$ is finite for each m, then

$$\mathbb{E}\left[\prod_{m=1}^{M} x_m\right] = \prod_{m=1}^{M} \mathbb{E}[x_m]$$

We won't prove the last fact in the general case, as this involves measure theory. However, we can illustrate the idea by showing that $\mathbb{E}[xy] = \mathbb{E}[x]\mathbb{E}[y]$ when x and y are independent and defined by (1.9). In this case, it can be shown (details omitted) that the random variables x and y are independent precisely when the events A and B are independent. Now observe that

$$(xy)(\omega) := x(\omega)y(\omega) = s\mathbb{1}\{\omega \in A\}t\mathbb{1}\{\omega \in B\} = st\mathbb{1}\{\omega \in A \cap B\}$$

Hence, by the definition of expectations, we have

$$\mathbb{E}[xy] = st\mathbb{P}(A \cap B) = st\mathbb{P}(A)\mathbb{P}(B) = s\mathbb{P}(A)t\mathbb{P}(B) = \mathbb{E}[x]\mathbb{E}[y]$$

Fact 1.3.2. If x and y are independent and g and f are any functions, then $f(x)$ and $g(y)$ are independent.

An important special case of the "independence means multiply" rule is as follows.

Fact 1.3.3. If random variables $x_1, \ldots, x_N$ are independent, and each has density p_n, then the joint density p exists, and is the product of the marginal densities:

$$p(s_1, \ldots, s_N) = \prod_{n=1}^{N} p_n(s_n)$$

Here are some useful facts relating independence and certain common distributions.

Fact 1.3.4. If $x_1, \ldots, x_k \overset{\text{IID}}{\sim} \mathcal{N}(0,1)$, then

$$Q := \sum_{i=1}^{k} x_i^2 \sim \chi^2(k)$$

Fact 1.3.5. If $Q_1, \ldots, Q_J$ are independent with $Q_j \sim \chi^2(k_j)$, then $\sum_{j=1}^{J} Q_j \sim \chi^2(\sum_j k_j)$.

Fact 1.3.6. If Z and Q are two random variables such that

1. $Z \sim \mathcal{N}(0,1)$,

2. $Q \sim \chi^2(k)$, and
3. Z and Q are independent,

then $Z(k/Q)^{1/2}$ has the t-distribution with k degrees of freedom.

Fact 1.3.7. If $Q_1 \sim \chi^2(k_1)$ and $Q_2 \sim \chi^2(k_2)$ are independent, then

$$\frac{Q_1/k_1}{Q_2/k_2}$$

is distributed as $F(k_1, k_2)$.

1.3.3 Variance and Covariance

Let $x \sim F$. For $k \in \mathbb{N}$, the **k-th moment of x** is defined as $\mathbb{E}[x^k] = \int s^k F(ds)$. If $\mathbb{E}[|x|^k] < \infty$ then the k-th moment is said to exist. For a random variable with the Cauchy distribution, even the first moment does not exist. For the normal distribution, every moment exists.

Fact 1.3.8. If the k-th moment of x exists, then so does the j-th moment for all $j \leq k$.

The **variance** of random variable x is defined as

$$\text{var}[x] := \mathbb{E}[(x - \mathbb{E}[x])^2]$$

This gives a measure of the dispersion of x. (Not all random variables have a well defined variance, but in general we'll just talk about the variance of a given random variable without adding the caveat "assuming it exists.") The **standard deviation** of x is $\sqrt{\text{var}[x]}$.

The **covariance** of random variables x and y is defined as

$$\text{cov}[x, y] := \mathbb{E}[(x - \mathbb{E}[x])(y - \mathbb{E}[y])]$$

Fact 1.3.9. If $x_1, \ldots, x_N$ are random variables and $\alpha_1, \ldots, \alpha_N$ are constant scalars, then

$$\text{var}\left[\sum_{n=1}^{N} \alpha_n x_n\right] = \sum_{n=1}^{N} \alpha_n^2 \,\text{var}[x_n] + 2 \sum_{n<m} \alpha_n \alpha_m \,\text{cov}[x_n, x_m]$$

In particular, if α and β are real numbers and x and y are random variables, then $\operatorname{var}[\alpha] = 0$,[10] $\operatorname{var}[\alpha + \beta x] = \beta^2 \operatorname{var}[x]$, and

$$\operatorname{var}[\alpha x + \beta y] = \alpha^2 \operatorname{var}[x] + \beta^2 \operatorname{var}[y] + 2\alpha\beta \operatorname{cov}[x, y]$$

Given two random variables x and y with finite variances σ_x^2 and σ_y^2 respectively, the **correlation** of x and y is defined as

$$\operatorname{corr}[x, y] := \frac{\operatorname{cov}[x, y]}{\sigma_x \sigma_y}$$

If $\operatorname{corr}[x, y] = 0$, we say that x and y are **uncorrelated**. For this to occur, it is necessary and sufficient that $\operatorname{cov}[x, y] = 0$. Positive correlation means that $\operatorname{corr}[x, y]$ is positive, while negative correlation means that $\operatorname{corr}[x, y]$ is negative.

Fact 1.3.10. Given any two random variables x, y and positive constants α, β, we have

$$-1 \leq \operatorname{corr}[x, y] \leq 1 \quad \text{and} \quad \operatorname{corr}[\alpha x, \beta y] = \operatorname{corr}[x, y]$$

Fact 1.3.11. If x and y are independent, then $\operatorname{cov}[x, y] = \operatorname{corr}[x, y] = 0$.

Note that the converse is not true: One can construct examples of dependent random variables with zero covariance.

1.3.4 Best Linear Predictors

As a little exercise that starts moving us in the direction of statistics, let's consider the problem of predicting the value of a random variable y given knowledge of the value of a second random variable x. Thus, we seek a function f such that $f(x)$ is close to y on average. To measure the "average distance" between $f(x)$ and y, we will use the mean squared deviation between $f(x)$ and y, which is

$$\mathbb{E}\left[(y - f(x))^2\right]$$

As we will learn in chapter 3, the minimizer of the mean squared deviation over all functions of x is obtained by choosing $f(x) = \mathbb{E}[y \mid x]$, where the right-hand size is the conditional expectation of y given x. However, the conditional expectation may

[10] Here $\operatorname{var}[\alpha]$ should be understood as $\operatorname{var}[\alpha \mathbb{1}\{\omega \in \Omega\}]$, as was the case when we discussed fact 1.1.4 on page 14.

be nonlinear and complicated, so let's now consider the simpler problem of finding a good predictor of y within a small and well-behaved class of functions. The class of functions we will consider is the set of "linear" functions

$$\mathcal{L} := \{ \text{ all functions of the form } \ell(x) = \alpha + \beta x \}$$

(While elementary courses refer to these functions as linear, in fact they are not linear unless $\alpha = 0$ (see §2.1.3). The class of functions $\mathcal{L}$ is more correctly known as the set of **affine** functions.) Thus, we consider the problem

$$\min_{\ell \in \mathcal{L}} \mathbb{E}\left[(y - \ell(x))^2\right] = \min_{\alpha,\beta \in \mathbb{R}} \mathbb{E}\left[(y - \alpha - \beta x)^2\right] \tag{1.23}$$

Expanding the square on the right-hand side and using linearity of $\mathbb{E}$, the objective function becomes

$$\psi(\alpha, \beta) := \mathbb{E}\left[y^2\right] - 2\alpha\mathbb{E}\left[y\right] - 2\beta\mathbb{E}\left[xy\right] + 2\alpha\beta\mathbb{E}\left[x\right] + \alpha^2 + \beta^2\mathbb{E}\left[x^2\right]$$

Computing the derivatives and solving the equations

$$\frac{\partial \psi(\alpha, \beta)}{\partial \alpha} = 0 \quad \text{and} \quad \frac{\partial \psi(\alpha, \beta)}{\partial \beta} = 0$$

We obtain (exercise 1.5.25) the minimizers

$$\beta^* := \frac{\operatorname{cov}[x, y]}{\operatorname{var}[x]} \quad \text{and} \quad \alpha^* := \mathbb{E}\left[y\right] - \beta^*\mathbb{E}\left[x\right] \tag{1.24}$$

The best linear predictor is therefore

$$\ell^*(x) := \alpha^* + \beta^* x$$

If you've studied elementary linear least squares regression before, you will realize that α^* and β^* are the "population" counterparts for the coefficient estimates in the regression setting. We'll talk more about the connections in the next chapter.

1.4 Asymptotics

In statistics, we often want to know how our tests and procedures will perform as the amount of data we have at hand becomes large. To this end, we now investigate the limiting properties of sequences of random variables. We begin by discussing three modes of convergence for random variables, all of which are used routinely in econometrics.

1.4.1 Modes of Convergence

Let $\{x_n\}_{n=1}^{\infty}$ be a sequence of random variables. We say that $\{x_n\}_{n=1}^{\infty}$ converges to random variable x **in probability** if

$$\text{for any } \delta > 0, \quad \mathbb{P}\{|x_n - x| > \delta\} \to 0 \quad \text{as} \quad n \to \infty$$

In symbols, this convergence is represented by $x_n \xrightarrow{p} x$. In almost all the applications we consider, the limit x will be a constant. The next example illustrates the definition for this case.

Example 1.4.1. If $x_n \sim \mathcal{N}(\alpha, 1/n)$, then $x_n \xrightarrow{p} \alpha$. That is, for any $\delta > 0$, we have $\mathbb{P}\{|x_n - \alpha| > \delta\} \to 0$. Fixing $\delta > 0$, the probability $\mathbb{P}\{|x_n - \alpha| > \delta\}$ is shown in figure 1.7 for two different values of n, where it corresponds to the size of the shaded areas. This probability collapses to zero as $n \to \infty$, decreasing the variance and causing the density to become more peaked.

A full proof of the convergence result in example 1.4.1 can be found by looking at the normal density and bounding tail probabilities. However, a much simpler proof can also be obtained by exploiting the connection between convergence in probability and convergence in mean squared error. The details are below.

Fact 1.4.1. Regarding convergence in probability, the following statements are true:

1. If $g\colon \mathbb{R} \to \mathbb{R}$ is continuous and $x_n \xrightarrow{p} x$, then $g(x_n) \xrightarrow{p} g(x)$.
2. If $x_n \xrightarrow{p} x$ and $y_n \xrightarrow{p} y$, then $x_n + y_n \xrightarrow{p} x + y$ and $x_n y_n \xrightarrow{p} xy$.
3. If $x_n \xrightarrow{p} x$ and $\alpha_n \to \alpha$, then $x_n + \alpha_n \xrightarrow{p} x + \alpha$ and $x_n \alpha_n \xrightarrow{p} x\alpha$.[11]

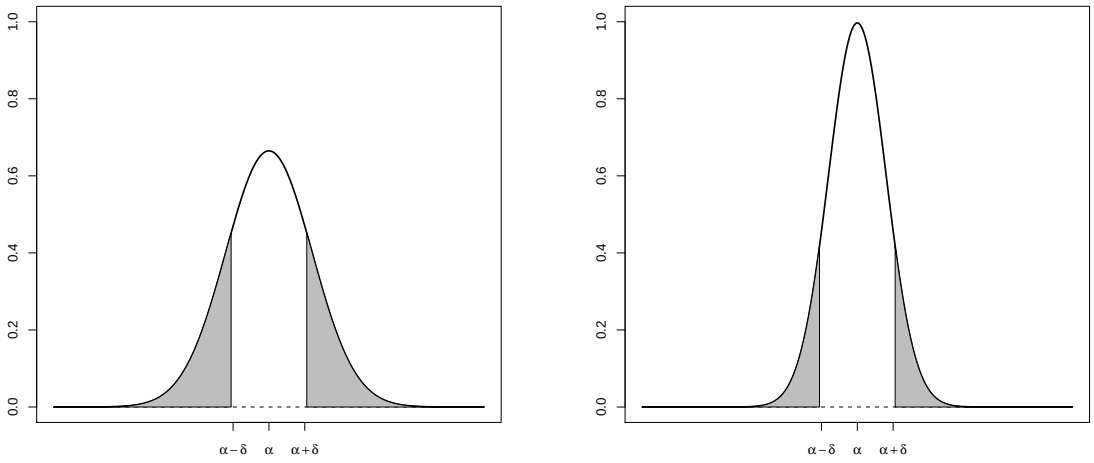
We say that $\{x_n\}$ converges to x **in mean square** if

$$\mathbb{E}\left[(x_n - x)^2\right] \to 0 \quad \text{as } n \to \infty$$

In symbols, this convergence is represented by $x_n \xrightarrow{ms} x$.

Fact 1.4.2. Regarding convergence in mean square, the following statements are true:

[11] Here $\{\alpha_n\}$ is a nonrandom scalar sequence.



(a) $n = 10$ (b) $n = 20$

Figure 1.7: $\mathbb{P}\{|x_n - \alpha| > \delta\}$ for $x_n \sim \mathcal{N}(\alpha, 1/n)$

1. If $x_n \overset{ms}{\to} x$, then $x_n \overset{p}{\to} x$.
2. If α is constant, then $x_n \overset{ms}{\to} \alpha$ if and only if $\mathbb{E}[x_n] \to \alpha$ and $\text{var}[x_n] \to 0$.

Part 1 of fact 1.4.2 follows from Chebychev's inequality, which states that for any random variable y with finite second moment and any $\delta > 0$, we have

$$\mathbb{P}\{|y| \geq \delta\} \leq \frac{\mathbb{E}[y^2]}{\delta^2} \tag{1.25}$$

(See exercise 1.5.29.) Using monotonicity of $\mathbb{P}$ and then applying (1.25) to $y = x_n - x$, we obtain

$$\mathbb{P}\{|x_n - x| > \delta\} \leq \mathbb{P}\{|x_n - x| \geq \delta\} \leq \frac{\mathbb{E}[(x_n - x)^2]}{\delta^2}$$

Part 1 of fact 1.4.2 follows. Part 2 is implied by the equality

$$\mathbb{E}[(x_n - \alpha)^2] = \text{var}[x_n] + (\mathbb{E}[x_n] - \alpha)^2$$

Verification of this equality is an exercise.

Example 1.4.2. In example 1.4.1, we stated that if $x_n \sim \mathcal{N}(\alpha, 1/n)$, then $x_n \overset{p}{\to} \alpha$. This follows from parts 1 and 2 of fact 1.4.2, since $\mathbb{E}[x_n] = \alpha$ and $\text{var}[x_n] = 1/n \to 0$.

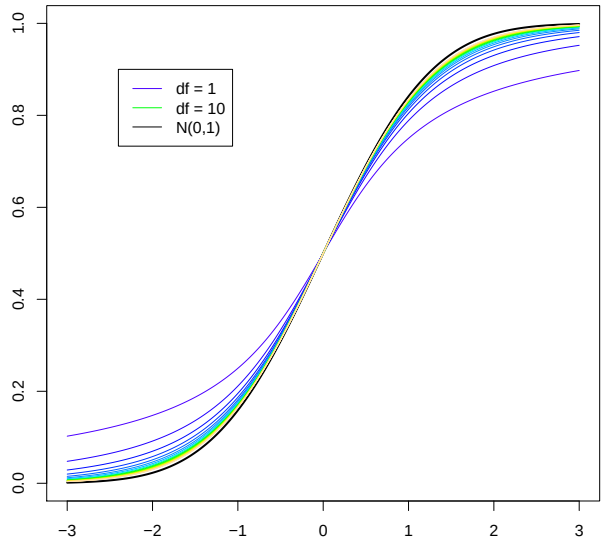


Figure 1.8: t-distribution with k df converges to $\mathcal{N}(0,1)$ as $k \to \infty$

Let $\{F_n\}_{n=1}^{\infty}$ be a sequence of cdfs, and let F be a cdf. We say that F_n **converges weakly** to F if, for any s such that F is continuous at s, we have

$$F_n(s) \to F(s) \quad \text{as} \quad n \to \infty$$

Example 1.4.3. It is well-known that the cdf of the t-distribution with k degrees of freedom converges to the standard normal cdf as $k \to \infty$. This convergence is illustrated in figure 1.8.

Sometimes densities are easier to work with than cdfs. In this connection, note that pointwise convergence of densities implies weak convergence of the corresponding distribution functions:

Fact 1.4.3. Let $\{F_n\}_{n=1}^{\infty}$ be a sequence of cdfs, and let F be a cdf. Suppose that all these cdfs are differentiable, and let p_n and p be the densities of F_n and F respectively. If $p_n(s) \to p(s)$ for all $s \in \mathbb{R}$, then F_n converges weakly to F.

Let $\{x_n\}_{n=1}^{\infty}$ and x be random variables, where $x_n \sim F_n$ and $x \sim F$. We say that x_n converges **in distribution** to x if F_n converges weakly to F. In symbols, this convergence is represented by $x_n \xrightarrow{d} x$.

Fact 1.4.4. Regarding convergence in distribution, the following statements are true:

1. If $g\colon \mathbb{R} \to \mathbb{R}$ is continuous and $x_n \xrightarrow{d} x$, then $g(x_n) \xrightarrow{d} g(x)$.
2. If $x_n \xrightarrow{p} x$, then $x_n \xrightarrow{d} x$.
3. If α is constant and $x_n \xrightarrow{d} \alpha$, then $x_n \xrightarrow{p} \alpha$.

The next result is sometimes known as **Slutsky's theorem**.

Fact 1.4.5. If α is constant, $x_n \xrightarrow{p} \alpha$ and $y_n \xrightarrow{d} y$, then $x_n + y_n \xrightarrow{d} \alpha + y$ and $x_n y_n \xrightarrow{d} \alpha y$.

1.4.2 The Law of Large Numbers

Two of the most important theorems in both probability and statistics are the law of large numbers and the central limit theorem. In their simplest forms, these theorems deal with averages of independent and identically distributed (IID) sequences. The law of large numbers tells us that these averages converge in probability to the mean of the distribution in question. The central limit theorem tells us that a simple transform of the average converges to a normal distribution.

Let's start with the **law of large numbers**, which relates to the sample mean

$$\bar{x}_N := \frac{1}{N} \sum_{n=1}^{N} x_n$$

of a given sample $x_1, \ldots, x_N$

Theorem 1.4.1. *Let $\{x_n\}$ be an* IID *sequence of random variables with common distribution F. If the first moment $\int |s| F(ds)$ is finite, then*

$$\bar{x}_N \xrightarrow{p} \mathbb{E}[x_n] = \int s F(ds) \quad \text{as} \quad N \to \infty \tag{1.26}$$

To prove theorem 1.4.1, we can use fact 1.4.2 on page 31. In view of this fact, it suffices to show that $\mathbb{E}[\bar{x}_N] \to \int s F(ds)$ and $\operatorname{var}[\bar{x}_N] \to 0$ as $N \to \infty$. These steps are left as an exercise (exercise 1.5.31). When you do the exercise, note to yourself exactly where independence bites.[12]

[12] The proof involves a bit of cheating, because it assumes that the variance of each x_n is finite. This second moment assumption is not necessary for the result, but it helps to simplify the proof.

Example 1.4.4. To illustrate the law of large numbers, consider flipping a coin until 10 heads have occurred. The coin is not fair: The probability of heads is 0.4. Let x be the number of tails observed in the process. It is known that such an x has the negative binomial distribution, and, with a little bit of googling, we find that the mean $\mathbb{E}[x]$ is 15. This means that if we simulate many observations of x and take the sample mean, we should get a value close to 15. Code to do this is provided in listing 1. Can you see how this program works?[13] An improved implementation is given in listing 2. The generation of a single observation has been wrapped in a function called `f`. To generate multiple observations, we have used the R function `replicate`, which is handy for simulations.

Listing 1 Illustrates the LLN

```
num.repetitions <- 10000
outcomes <- numeric(num.repetitions)
for (i in 1:num.repetitions) {
    num.tails <- 0
    num.heads <- 0
    while (num.heads < 10) {
        b <- runif(1)
        num.heads <- num.heads + (b < 0.4)
        num.tails <- num.tails + (b >= 0.4)
    }
    outcomes[i] <- num.tails
}
print(mean(outcomes))
```

At first glance, the law of large numbers (1.26) appears to only be a statement about the sample mean, but actually it can be applied to functions of the random variable as well. For example, if $h\colon \mathbb{R} \to \mathbb{R}$ is such that $\int |h(s)|F(ds)$ is finite, then

$$\frac{1}{N}\sum_{n=1}^{N} h(x_n) \xrightarrow{p} \mathbb{E}[h(x_n)] = \int h(s)F(ds) \tag{1.27}$$

This can be confirmed by letting $y_n := h(x_n)$ and then applying theorem 1.4.1.

[13] Hint: If u is uniform on $[0,1]$ and $q \in [0,1]$, then $\mathbb{P}\{u \leq q\} = q$. This fact is used to simulate the coin flips. Also recall that the logical values TRUE and FALSE are treated as 1 and 0 respectively in algebraic expressions.

Listing 2 Second version of listing 1

```
f <- function(q) {
    num.tails <- 0
    num.heads <- 0
    while (num.heads < 10) {
        b <- runif(1)
        num.heads <- num.heads + (b < q)
        num.tails <- num.tails + (b >= q)
    }
    return(num.tails)
}
outcomes <- replicate(10000, f(0.4))
print(mean(outcomes))
```

Also, the law of large numbers applies to probabilities as well as expectations. To see this, let $x \sim F$, fix $B \subset \mathbb{R}$, and consider the probability $\mathbb{P}\{x \in B\}$. Let h be the function defined by $h(s) = \mathbb{1}\{s \in B\}$ for all $s \in \mathbb{R}$. Using the principle that expectations of indicator functions equal probabilities of events (page 14), we have

$$\mathbb{E}[h(x)] = \mathbb{E}[\mathbb{1}\{x \in B\}] = \mathbb{P}\{x \in B\}$$

It now follows from (1.27) that if $\{x_n\}$ is an IID sample from F, then

$$\frac{1}{N}\sum_{n=1}^{N} \mathbb{1}\{x_n \in B\} \xrightarrow{p} \mathbb{P}\{x_n \in B\} \tag{1.28}$$

The left hand side is the fraction of the sample that falls in the set B, and (1.28) tells us that this fraction converges to the probability that $x_n \in B$.

1.4.3 The Central Limit Theorem

The **central limit theorem** is another classical result from probability theory. It is arguably one of the most beautiful and important results in all of mathematics. Relative to the LLN, it requires an additional second moment condition.

Theorem 1.4.2. *Assume the conditions of theorem 1.4.1. If, in addition, the second moment $\int s^2 F(ds)$ is finite, then*

$$\sqrt{N}(\bar{x}_N - \mu) \xrightarrow{d} y \sim \mathcal{N}(0, \sigma^2) \quad \text{as} \quad N \to \infty \tag{1.29}$$

where $\mu := \int sF(ds) = \mathbb{E}[x_n]$ *and* $\sigma^2 := \int (s-\mu)^2 F(ds) = \text{var}[x_n]$.

Another common statement of the central limit theorem is as follows: If all the conditions of theorem 1.4.2 are satisfied, then

$$z_N := \sqrt{N}\left\{\frac{\bar{x}_N - \mu}{\sigma}\right\} \xrightarrow{d} z \sim \mathcal{N}(0,1) \quad \text{as} \quad N \to \infty \tag{1.30}$$

Exercise 1.5.32 asks you to confirm this via theorem 1.4.2 and fact 1.4.4.

The central limit theorem tells us about the distribution of the sample mean when N is large. Arguing informally, for N large we have

$$\sqrt{N}(\bar{x}_N - \mu) \approx y \sim \mathcal{N}(0, \sigma^2)$$

$$\therefore \quad \bar{x}_N \approx \frac{y}{\sqrt{N}} + \mu \sim \mathcal{N}\left(\mu, \frac{\sigma^2}{N}\right)$$

Here $\approx$ means that the distributions are approximately equal. We see that $\bar{x}_N$ is approximately normal, with mean equal to $\mu := \mathbb{E}[x_1]$ and variance converging to zero at a rate proportional to $1/N$.

The convergence in (1.30) is illustrated by listing 3, the output of which is given in figure 1.9. The listing generates 5,000 observations of the random variable z_N defined in (1.30), where each x_n is $\chi^2(5)$. (The mean of this distribution is 5, and the variance is $2 \times 5 = 10$.) The observations of z_N are stored in the vector `outcomes`, and then histogrammed. The last line of the listing superimposes the density of the standard normal distribution over the histogram. As expected, the fit is pretty good.

Before finishing this section, we briefly note the following extension to the central limit theorem:

Theorem 1.4.3. *Assume the conditions of theorem 1.4.2. If* $g\colon \mathbb{R} \to \mathbb{R}$ *is differentiable at* μ *and* $g'(\mu) \neq 0$*, then*

$$\sqrt{N}\{g(\bar{x}_N) - g(\mu)\} \xrightarrow{d} \mathcal{N}(0, g'(\mu)^2\sigma^2) \quad \text{as} \quad N \to \infty \tag{1.31}$$

This theorem is used frequently in statistics, to obtain the asymptotic distribution of certain kinds estimators. The technique is referred to as the **delta method**. The proof of theorem 1.4.3 is based on Taylor expansion of g around the point μ. Some of the details are given in exercise 1.5.39. One word of warning when applying this theorem: In many situations, a rather large value of N is required for the convergence in (1.31) to occur.

Listing 3 Illustrates the CLT

```
num.replications <- 5000
outcomes <- numeric(num.replications)
N <- 1000
k <- 5      # Degrees of freedom
for (i in 1:num.replications) {
    xvec <- rchisq(N, k)
    outcomes[i] <- sqrt(N / (2 * k)) * (mean(xvec) - k)
}
hist(outcomes, breaks=50, freq=FALSE)
curve(dnorm, add=TRUE, lw=2, col="blue")
```

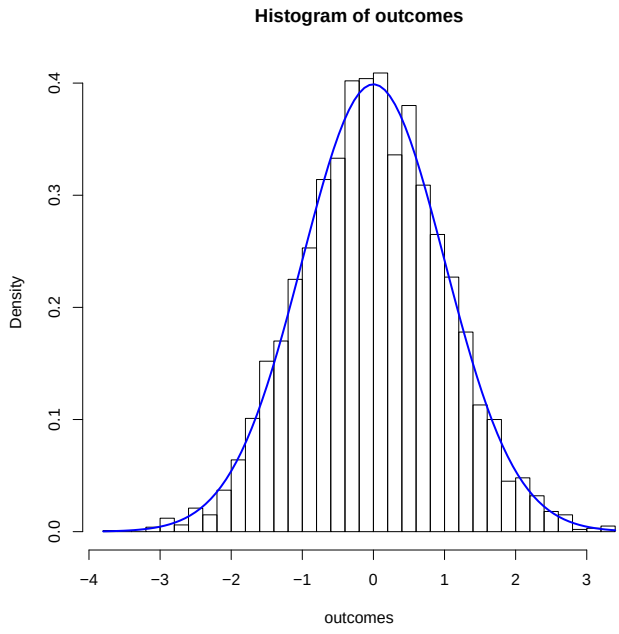


Figure 1.9: Illustration of the CLT

1.5 Exercises

Ex. 1.5.1. Suppose that $\mathbb{P}$ is a probability on $(\Omega, \mathcal{F})$, so that $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B)$ whenever A and B are disjoint. Show that if A, B and C are disjoint, then $\mathbb{P}(A \cup B \cup C) = \mathbb{P}(A) + \mathbb{P}(B) + \mathbb{P}(C)$.

Ex. 1.5.2. Prove fact 1.1.2: $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$ for any A, B.[14]

Ex. 1.5.3. Given sample space $\Omega := \{1, 2, 3\}$, let $A := \{1\}$, $B := \{2\}$ and $C := \{3\}$. Let $\mathbb{P}(A) = \mathbb{P}(B) = 1/3$. Compute $\mathbb{P}(C)$, $\mathbb{P}(A \cup B)$, $\mathbb{P}(A \cap B)$, $\mathbb{P}(A^c)$, $\mathbb{P}(A^c \cup B^c)$ and $\mathbb{P}(A \mid B)$. Are A and C independent?

Ex. 1.5.4. A dice is designed so that the probability of getting face m is qm, where $m \in \{1, \ldots, 6\}$ and q is a constant. Compute q.

Ex. 1.5.5. Let Ω be a nonempty finite set, and let ω_0 be a fixed element of Ω. For each $A \subset \Omega$, define $\mathbb{P}(A) := \mathbb{1}\{\omega_0 \in A\}$. Is $\mathbb{P}$ a probability on Ω? Why or why not?

Ex. 1.5.6. Let Ω be any sample space, and let $\mathbb{P}$ be a probability on the subsets $\mathcal{F}$. Let $A \in \mathcal{F}$. Show that if $\mathbb{P}(A) = 0$ or $\mathbb{P}(A) = 1$, then A is independent of every other event in $\mathcal{F}$. Show that if A is independent of itself, then either $\mathbb{P}(A) = 0$ or $\mathbb{P}(A) = 1$. Show that if A and B are independent, then A^c and B^c are also independent.

Ex. 1.5.7. Let $\mathbb{P}$ and Ω be defined as in example 1.1.5. Show that $\mathbb{P}$ is additive, in the sense that if A and B are disjoint events, then $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B)$.

Ex. 1.5.8. Let $\mathbb{P}$ and Ω be defined as in example 1.1.5. Let A be the event that the first switch is on, and let B be the event that the second switch is on. Show that A and B are independent under $\mathbb{P}$.

Ex. 1.5.9. Show that when Ω is finite, a random variable x on Ω can only take on a finite set of values (i.e., has finite range).[15]

Ex. 1.5.10. Recall F_x defined in (1.11). We claimed that F_x is a cdf, which implies that $\lim_{s \to \infty} F_x(s) = 1$. Verify this when x is the finite-valued random variable in (1.4).

Ex. 1.5.11. Recall F_x defined in (1.11). Suppose that x is the finite-valued random variable in (1.4). Show that $\lim_{s \to -\infty} F_x(s) = 0$. If you can, show that F is right-continuous.

[14] Hint: Sketching the Venn diagram, convince yourself that $A = [(A \cup B) \setminus B] \cup (A \cap B)$. Finish the proof using the definition of a probability and fact 1.1.1 (page 5).

[15] Hint: Have a look at the definition of a function in §13.1.1.

Ex. 1.5.12. Prove the claim in fact 1.2.2 on page 17.

Ex. 1.5.13. Let x be a discrete random variable taking values $s_1, \ldots, s_J$, and let $p_j := \mathbb{P}\{x = s_j\}$. Show that $0 \leq p_j \leq 1$ for each j, and $\sum_{j=1}^{J} p_j = 1$.

Ex. 1.5.14. This exercise describes the **inverse transform** method for generating random variables with arbitrary distribution from uniform random variables. The uniform cdf on $[0, 1]$ is given by $F(s) = 0$ if $s < 0$, $F(s) = s$ if $0 \leq s \leq 1$, and $F(s) = 1$ if $s > 1$. Let G be another cdf on $\mathbb{R}$. Suppose that G is strictly increasing, and let G^{-1} be the inverse (quantile). Show that if $u \sim F$, then $G^{-1}(u) \sim G$.

Ex. 1.5.15. Let $x \sim F$ where F is the uniform cdf on $[0, 1]$. Give an expression for the cdf G of the random variable $y = x^2$.

Ex. 1.5.16. Let $y \sim F$, where F is a cdf. Show that $F(s) = \mathbb{E}\left[\mathbb{1}\{y \leq s\}\right]$ for any s.

Ex. 1.5.17. Confirm monotonicity of expectations (fact 1.1.6 on page 14) for the special case where x and y are the random variables in (1.9).

Ex. 1.5.18. Prove fact 1.3.8. (Existence of k-th moment implies existence of j-th moment for all $j \leq k$.)

Ex. 1.5.19. Confirm the expression for variance of linear combinations in fact 1.3.9.

Ex. 1.5.20. Let x and y be scalar random variables. With reference to fact 1.3.10 on page 29, is it true that $\operatorname{corr}[\alpha x, \beta y] = \operatorname{corr}[x, y]$ for *any* constant scalars α and β? Why or why not?

Ex. 1.5.21. Confirm the claim in fact 1.3.11: If x and y are independent, then $\operatorname{cov}[x, y] = \operatorname{corr}[x, y] = 0$.

Ex. 1.5.22. Let x_1 and x_2 be random variables with densities p_1 and p_2. Let q be their joint density. Show that x_1 and x_2 are independent whenever $q(s, s') = p_1(s)p_2(s')$ for every $s, s' \in \mathbb{R}$.

Ex. 1.5.23. Fact 1.3.2 tells us that if x and y are independent random variables and g and f are any two functions, then $f(x)$ and $g(y)$ are independent. Prove this for the case where $f(x) = 2x$ and $g(y) = 3y - 1$.

Ex. 1.5.24. Let x and y be independent uniform random variables on $[0, 1]$. Let $z := \max\{x, y\}$. Compute the cdf, density and mean of z.[16] In addition, compute the cdf of $w := \min\{x, y\}$.

[16]Hint: Fix $s \in \mathbb{R}$ and compare the sets $\{z \leq s\}$ and $\{x \leq s\} \cap \{y \leq s\}$. What is the relationship between these two sets?

Ex. 1.5.25. Confirm the solutions in (1.24).

Ex. 1.5.26. Consider the setting of §1.3.4. Let α^*, β^* and ℓ^* be as defined there. Let the prediction error u be defined as $u := y - \ell^*(x)$. Show that

1. $\mathbb{E}[\ell^*(x)] = \mathbb{E}[y]$
2. $\text{var}[\ell^*(x)] = \text{corr}[x, y]^2 \, \text{var}[y]$
3. $\text{var}[u] = (1 - \text{corr}[x, y]^2) \, \text{var}[y]$

Ex. 1.5.27. Continuing on from exercise 1.5.26, show that $\text{cov}[\ell^*(x), u] = 0$.

Ex. 1.5.28. Let $\{x_n\}$ be a sequence of random variables satisfying $x_n = y$ for all n, where y is a single random variable. Show that if $\mathbb{P}\{y = -1\} = \mathbb{P}\{y = 1\} = 0.5$, then $x_n \xrightarrow{p} 0$ fails. Show that if $\mathbb{P}\{y = 0\} = 1$, then $x_n \xrightarrow{p} 0$ holds.

Ex. 1.5.29. Prove Chebychev's inequality (1.25). In particular, show that if x is a random variable with finite second moment and $\delta > 0$, then $\mathbb{P}\{|x| \geq \delta\} \leq \frac{\mathbb{E}[x^2]}{\delta^2}$.

Ex. 1.5.30. We saw in fact 1.4.4 that if $x_n \xrightarrow{p} x$, then $x_n \xrightarrow{d} x$. Show that the converse is not generally true. In other words, give an example of a sequence of random variables $\{x_n\}$ and random variable x such that x_n converges to x in distribution, but not in probability.

Ex. 1.5.31. In this exercise, we complete the proof of the LLN on page 34. Let $\{x_n\}$ be an IID sequence of random variables with common distribution F. Show that $\mathbb{E}[\bar{x}_N] \to \int s F(ds)$ and $\text{var}[\bar{x}_N] \to 0$ as $N \to \infty$.

Ex. 1.5.32. Confirm (1.30) via theorem 1.4.2 and fact 1.4.4.

Ex. 1.5.33 (Computational)**.** Provided that we can at least generate uniform random variables, the inverse transform method (see exercise 1.5.14) can be (and is) used to generate random variables with arbitrary distribution G. Pick three different continuous distributions G_1, G_2 and G_3 available in R. Using Q-Q plots,[17] examine for each G_i whether the random variables generated via inverse transform do appear equally distributed to the random variables generated from G_i using R's built in algorithms (accessed through r*name*, where *name* is one of `norm`, `lnorm`, etc.).

Ex. 1.5.34 (Computational)**.** Using numerical integration, show that the 8th moment of the standard normal density is approximately 105.

[17] Look them up if you don't know what they are. In R, see the documentation on `qqplot`.

Ex. 1.5.35 (Computational)**.** Using numerical integration and a `for` loop, compute the first 10 moments of the exponential distribution with mean 1. (The exponential distribution has one parameter. If the mean is 1, the value of the parameter is pinned down to what value?)

Ex. 1.5.36 (Computational)**.** Using a `for` loop, plot the chi-squared density for $k = 1, 2, 3, 4, 5$, all on the same figure. Use different colors for different k, and include a legend.

Ex. 1.5.37 (Computational)**.** Replicate the simulation performed in listing 3, but this time for $N = 2$. Why is the fit not as good?

Ex. 1.5.38 (Computational)**.** Replicate the simulation performed in listing 3, but this time for uniformly distributed random variables on $[-1, 1]$. Compare histograms and normal density plots in the manner of figure 1.9. Use the appropriate mean and variance in the normal density. Produce plots for $N = 1$, $N = 2$, $N = 5$ and $N = 200$.

Ex. 1.5.39. This exercise covers some of the proof behind theorem 1.4.3 on page 37. Suppose that $\{t_n\}$ is a sequence of random variables, θ is a constant, and

$$\sqrt{n}(t_n - \theta) \xrightarrow{d} \mathcal{N}(0, \sigma^2) \quad \text{as} \quad n \to \infty$$

Let $g \colon \mathbb{R} \to \mathbb{R}$ be differentiable at θ with $g'(\theta) \neq 0$. Taking a first order Taylor expansion of g around θ, we can write $g(t_n) = g(\theta) + g'(\theta)(t_n - \theta) + R(t_n - \theta)$, where $R(t_n - \theta)$ is a remainder term. It turns out that under these conditions we have $\sqrt{n}R(t_n - \theta) \xrightarrow{p} 0$. The details are omitted. Using this fact, prove carefully that $\sqrt{n}\{g(t_n) - g(\theta)\} \xrightarrow{d} \mathcal{N}(0, g'(\theta)^2\sigma^2)$.

1.5.1 Solutions to Selected Exercises

Solution to Exercise 1.5.1. If A, B and C are disjoint, then $A \cup B$ and C are also disjoint, and $A \cup B \cup C = (A \cup B) \cup C$. As a result, using additivity over pairs,

$$\mathbb{P}(A \cup B \cup C) = \mathbb{P}((A \cup B) \cup C) = \mathbb{P}(A \cup B) + \mathbb{P}(C) = \mathbb{P}(A) + \mathbb{P}(B) + \mathbb{P}(C)$$

This result can be extended to an arbitrary number of sets by using induction. □

Solution to Exercise 1.5.2. Pick any sets $A, B \in \mathcal{F}$. To show that

$$\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$$

we start by decomposing A into the union of two disjoint sets: $A = [(A \cup B) \setminus B] \cup (A \cap B)$. Using additivity of $\mathbb{P}$, we then have

$$\mathbb{P}(A) = \mathbb{P}[(A \cup B) \setminus B] + \mathbb{P}(A \cap B)$$

Since $B \subset (A \cup B)$, we can apply part 1 of fact 1.1.1 (page 5) to obtain

$$\mathbb{P}(A) = \mathbb{P}(A \cup B) - \mathbb{P}(B) + \mathbb{P}(A \cap B)$$

Rearranging this expression gives the result that we are seeking. □

Solution to Exercise 1.5.3. First, $\mathbb{P}(C) = 1/3$ as $1 = \mathbb{P}(\Omega) = \mathbb{P}(A \cup B \cup C) = \mathbb{P}(A) + \mathbb{P}(B) + \mathbb{P}(C) = 1/3 + 1/3 + \mathbb{P}(C)$, and hence $\mathbb{P}(C) = 1/3$. In addition, $\mathbb{P}(A \cup B) = 2/3$, $\mathbb{P}(A \cap B) = 0$, $\mathbb{P}(A^c) = 2/3$, $\mathbb{P}(A^c \cup B^c) = \mathbb{P}((A \cap B)^c) = \mathbb{P}(\Omega) = 1$, and $\mathbb{P}(A \cap C) = 0 \neq 1/9 = \mathbb{P}(A)\mathbb{P}(C)$. Therefore A is not independent of C. □

Solution to Exercise 1.5.4. When the dice is rolled one face must come up, so the sum of the probabilities is one. More formally, letting $\Omega = \{1, \ldots, 6\}$ be the sample space, we have

$$\mathbb{P}\{1, \ldots, 6\} = \mathbb{P} \cup_{m=1}^{6} \{m\} = \sum_{m=1}^{6} \mathbb{P}\{m\} = \sum_{m=1}^{6} qm = 1$$

Solving the last equality for q, we get $q = 1/21$. □

Solution to Exercise 1.5.5. To show that $\mathbb{P}$ is a probability on Ω we need to check that

1. $\mathbb{1}\{\omega_0 \in A\} \in [0, 1]$ for every $A \subset \Omega$.
2. $\mathbb{1}\{\omega_0 \in \Omega\} = 1$
3. If $A \cap B = \emptyset$, then $\mathbb{1}\{\omega_0 \in A \cup B\} = \mathbb{1}\{\omega_0 \in A\} + \mathbb{1}\{\omega_0 \in B\}$

1 is immediate from the definition of an indicator function. 2 holds because $\omega_0 \in \Omega$. Regarding 3, pick any disjoint A and B. If $\omega_0 \in A$, then $\omega_0 \notin B$, and we have

$$\mathbb{1}\{\omega_0 \in A \cup B\} = 1 = \mathbb{1}\{\omega_0 \in A\} + \mathbb{1}\{\omega_0 \in B\}$$

If $\omega_0 \in B$, then $\omega_0 \notin A$, and once again we have

$$\mathbb{1}\{\omega_0 \in A \cup B\} = 1 = \mathbb{1}\{\omega_0 \in A\} + \mathbb{1}\{\omega_0 \in B\}$$

Finally, if ω_0 is in neither A nor B, then

$$\mathbb{1}\{\omega_0 \in A \cup B\} = 0 = \mathbb{1}\{\omega_0 \in A\} + \mathbb{1}\{\omega_0 \in B\}$$

We have shown that 1–3 hold, and hence $\mathbb{P}$ is a probability on Ω. □

Solution to Exercise 1.5.6. Suppose that $\mathbb{P}(A) = 0$ and that $B \in \mathcal{F}$. We claim that $\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$, or, in this case, $\mathbb{P}(A \cap B) = 0$. Using nonnegativity and monotonicity of $\mathbb{P}$ (fact 1.1.1), we obtain

$$0 \leq \mathbb{P}(A \cap B) \leq P(A) = 0$$

Therefore $\mathbb{P}(A \cap B) = 0$ as claimed.

Now suppose that $\mathbb{P}(A) = 1$. We claim that $\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$, or, in this case, $\mathbb{P}(A \cap B) = \mathbb{P}(B)$. In view of fact 1.1.2 on page 6, we have

$$\mathbb{P}(A \cap B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cup B)$$

Since $\mathbb{P}(A) = 1$, it suffices to show that $\mathbb{P}(A \cup B) = 1$. This last equality is implied by monotonicity of $\mathbb{P}$, because $1 = \mathbb{P}(A) \leq \mathbb{P}(A \cup B) \leq 1$.

Next, suppose that A is independent of itself. Then $\mathbb{P}(A) = \mathbb{P}(A \cap A) = \mathbb{P}(A)\mathbb{P}(A) = \mathbb{P}(A)^2$. If $a = a^2$, then either $a = 0$ or $a = 1$.

Finally, let A and B be independent. We have

$$\mathbb{P}(A^c \cap B^c) = \mathbb{P}((A \cup B)^c) = 1 - \mathbb{P}(A \cup B)$$

Applying fact 1.1.2 and independence, we can transform the right-hand side to obtain

$$\mathbb{P}(A^c \cap B^c) = (1 - \mathbb{P}(A))(1 - \mathbb{P}(B)) = \mathbb{P}(A^c)\mathbb{P}(B^c)$$

In other words, A^c and B^c are independent. □

Solution to Exercise 1.5.7. The proof is almost identical to the proof of additivity in example 1.1.4 (page 4). □

Solution to Exercise 1.5.8. The proof of independence is essentially the same as the proof of independence of A and B in example 1.1.6 (page 6). □

Solution to Exercise 1.5.10. We are assuming that x has finite range, and hence takes only finitely many different values. Let m be the largest such value. For this m, we have

$$\lim_{s\to\infty} F_x(s) \geq F_x(m) = \mathbb{P}\{\omega \in \Omega : x(\omega) \leq m\} = \mathbb{P}(\Omega) = 1$$

(The inequality is due to the fact that F_x is increasing.) On the other hand,

$$\lim_{s\to\infty} F_x(s) = \lim_{s\to\infty} \mathbb{P}\{x \leq s\} \leq \lim_{s\to\infty} \mathbb{P}(\Omega) = 1$$

From these two inequalities we get $1 \leq \lim_{s\to\infty} F_x(s) \leq 1$, which is equivalent to $\lim_{s\to\infty} F_x(s) = 1$. □

Solution to Exercise 1.5.12. Fix $s \geq 0$. Using additivity over disjoint sets, we have

$$F_{|x|}(s) := \mathbb{P}\{|x| \leq s\} = \mathbb{P}\{-s \leq x \leq s\} = \mathbb{P}\{x = -s\} + \mathbb{P}\{-s < x \leq s\}$$

By assumption, $\mathbb{P}\{x = -s\} = 0$. Applying fact 1.2.1 on page 17 then yields

$$F_{|x|}(s) = \mathbb{P}\{-s < x \leq s\} = F(s) - F(-s)$$

The claim $F_{|x|}(s) = 2F(s) - 1$ now follows from the definition of symmetry. □

Solution to Exercise 1.5.13. That $0 \leq p_j \leq 1$ for each j follows immediately from the definition of $\mathbb{P}$. In addition, using additivity of $\mathbb{P}$, we have

$$\sum_{j=1}^{J} p_j = \sum_{j=1}^{J} \mathbb{P}\{x = s_j\} = \mathbb{P} \cup_{j=1}^{J} \{x = s_j\} = \mathbb{P}(\Omega) = 1 \tag{1.32}$$

(We are using the fact that the sets $\{x = s_j\}$ disjoint. Why is this always true? Look carefully at the definition of a function given in §13.1.1.) □

Solution to Exercise 1.5.14. Let $z := G^{-1}(u)$. We want to show that $z \sim G$. Since G is monotone increasing we have $G(a) \leq G(b)$ whenever $a \leq b$. As a result, for any $s \in \mathbb{R}$,

$$\mathbb{P}\{z \leq s\} = \mathbb{P}\{G^{-1}(u) \leq s\} = \mathbb{P}\{G(G^{-1}(u)) \leq G(s)\} = \mathbb{P}\{u \leq G(s)\} = G(s)$$

We have shown that $z \sim G$ as claimed. □

Solution to Exercise 1.5.15. Evidently $G(s) = 0$ when $s < 0$. For $s \geq 0$ we have

$$\mathbb{P}\{x^2 \leq s\} = \mathbb{P}\{|x| \leq \sqrt{s}\} = \mathbb{P}\{x \leq \sqrt{s}\} = F(\sqrt{s})$$

Thus, $G(s) = F(\sqrt{s})\mathbb{1}\{s \geq 0\}$. □

Solution to Exercise 1.5.17. If $x(\omega) := \mathbb{1}\{\omega \in A\} \leq \mathbb{1}\{\omega \in B\} =: y(\omega)$ for any $\omega \in \Omega$, then $A \subset B$. (If $\omega \in A$, then $x(\omega) = 1$. Since $x(\omega) \leq y(\omega) \leq 1$, we then have $y(\omega) = 1$, and hence $\omega \in B$.) Using (1.8) and monotonicity of $\mathbb{P}$, we then have

$$\mathbb{E}[x] = \mathbb{E}[\mathbb{1}\{\omega \in A\}] = \mathbb{P}(A) \leq \mathbb{P}(B) = \mathbb{E}[\mathbb{1}\{\omega \in B\}] = \mathbb{E}[y]$$

as was to be shown. □

Solution to Exercise 1.5.18. Let a be any nonnegative number, and let $j \leq k$. If $a \geq 1$, then $a^j \leq a^k$. If $a < 1$, then $a^j \leq 1$. Thus, for any $a \geq 0$, we have $a^j \leq a^k + 1$, and for any random variable x we have $|x|^j \leq |x|^k + 1$. Using monotonicity of expectations (fact 1.1.6 on page 14) and $\mathbb{E}[1] = 1$, we then have $\mathbb{E}[|x|^j] \leq \mathbb{E}[|x^k|] + 1$. Hence the j-th moment exists whenever the k-th moment exists. □

Solution to Exercise 1.5.23. Let $u := f(x) = 2x$ and $v := g(y) = 3y - 1$, where x and y are independent. Independence of u and v can be confirmed via (1.22) on page 26. Fixing s_1 and s_2 in $\mathbb{R}$, we have

$$\begin{aligned}\mathbb{P}\{u \leq s_1, v \leq s_2\} &= \mathbb{P}\{x \leq s_1/2, y \leq (s_2+1)/3\} \\ &= \mathbb{P}\{x \leq s_1/2\}\mathbb{P}\{y \leq (s_2+1)/3\} = \mathbb{P}\{u \leq s_1\}\mathbb{P}\{v \leq s_2\}\end{aligned}$$

Thus u and v are independent as claimed. □

Solution to Exercise 1.5.24. As in the statement of the exercise, x and y are independent uniform random variables on $[0, 1]$, $z := \max\{x, y\}$ and $w := \min\{x, y\}$. As a first step to the proofs, you should convince yourself that if a, b and c are three numbers, then

- $\max\{a, b\} \leq c$ if and only if $a \leq c$ and $b \leq c$
- $\min\{a, b\} \leq c$ if and only if $a \leq c$ or $b \leq c$

Using these facts, next convince yourself that, for any $s \in \mathbb{R}$,

- $\{z \leq s\} = \{x \leq s\} \cap \{y \leq s\}$
- $\{w \leq s\} = \{x \leq s\} \cup \{y \leq s\}$

(For each, equality, show that if ω is in the right-hand side, then ω is in the left-hand side, and vice versa.) Now, for $s \in [0,1]$, we have

$$\mathbb{P}\{z \leq s\} = \mathbb{P}[\{x \leq s\} \cap \{y \leq s\}] = \mathbb{P}\{x \leq s\}\mathbb{P}\{y \leq s\} = s^2$$

By differentiating we get the density $p(s) = 2s$, and by integrating $\int_0^1 sp(s)ds$ we get $\mathbb{E}[z] = 2/3$. Finally, regarding the cdf of w, for $s \in [0,1]$ we have

$$\begin{aligned}\mathbb{P}\{w \leq s\} &= \mathbb{P}[\{x \leq s\} \cup \{y \leq s\}] \\ &= \mathbb{P}\{x \leq s\} + \mathbb{P}\{y \leq s\} - \mathbb{P}[\{x \leq s\} \cap \{y \leq s\}]\end{aligned}$$

Hence $\mathbb{P}\{w \leq s\} = 2s - s^2$. □

Solution to Exercise 1.5.27. Using $y = \ell^*(x) + u$ and the results form exercise 1.5.26, we have

$$\begin{aligned}\operatorname{var}[\ell^*(x) + u] &= \operatorname{var}[y] \\ &= \operatorname{corr}[x,y]^2 \operatorname{var}[y] + (1 - \operatorname{corr}[x,y]^2) \operatorname{var}[y] \\ &= \operatorname{var}[\ell^*(x)] + \operatorname{var}[u]\end{aligned}$$

It follows (why?) that $\operatorname{cov}[\ell^*(x), u] = 0$ as claimed. □

Solution to Exercise 1.5.28. From the definition of convergence in probability (see §1.4.1), the statement $x_n \xrightarrow{p} 0$ means that, given any $\delta > 0$, we have $\mathbb{P}\{|x_n| > \delta\} \to 0$. Consider first the case where $\mathbb{P}\{y = -1\} = \mathbb{P}\{y = 1\} = 0.5$. Take $\delta = 0.5$. Then, since $x_n = y$ for all n,

$$\mathbb{P}\{|x_n| > \delta\} = \mathbb{P}\{|y| > 0.5\} = 1$$

Thus, the sequence does not converge to zero. Hence $x_n \xrightarrow{p} 0$ fails. On the other hand, if $\mathbb{P}\{y = 0\} = 1$, then for any $\delta > 0$ we have

$$\mathbb{P}\{|x_n| > \delta\} = \mathbb{P}\{|y| > \delta\} = 0$$

This sequence does converge to zero (in fact it's constant at zero), and $x_n \xrightarrow{p} 0$ holds. □

Solution to Exercise 1.5.29. Pick any random variable x and $\delta > 0$. By considering what happens at an arbitrary $\omega \in \Omega$, you should be able to convince yourself that

$$x^2 = \mathbb{1}\{|x| \geq \delta\}x^2 + \mathbb{1}\{|x| < \delta\}x^2 \geq \mathbb{1}\{|x| \geq \delta\}\delta^2$$

Using fact 1.1.6 (page 14), fact 1.1.3 (page 14) and rearranging completes the proof that $\mathbb{P}\{|x| \geq \delta\} \leq \frac{\mathbb{E}[x^2]}{\delta^2}$. □

Solution to Exercise 1.5.30. We want to give an example of a sequence of random variables $\{x_n\}$ and random variable x such that x_n converges to x in distribution, but not in probability. Many examples can be found by using IID sequences. For example, if $\{x_n\}_{n=1}^{\infty}$ and x are IID standard normal random variables, then x_n and x have the same distribution for all n, and hence x_n converges in distribution to x. However, $z_n := x_n - x$ has distribution $\mathcal{N}(0, 2)$ for all n. Letting z be any random variable with distribution $\mathcal{N}(0, 2)$ and δ be any strictly positive constant, we have $\mathbb{P}\{|x_n - x| \geq \delta\} = \mathbb{P}\{|z| \geq \delta\} > 0$. Thus, $\mathbb{P}\{|x_n - x| \geq \delta\}$ does not converge to zero. □

Solution to Exercise 1.5.31. By linearity of expectations,

$$\mathbb{E}[\bar{x}_N] = \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}[x_n] = \frac{N}{N}\int sF(ds) = \int sF(ds)$$

This confirms that $\mathbb{E}[\bar{x}_N] \to \int sF(ds)$ as claimed. To see that $\operatorname{var}[\bar{x}_N] \to 0$ as $N \to \infty$, let σ^2 be the common variance of each x_n. Using fact 1.3.9, we obtain

$$\operatorname{var}\left[\frac{1}{N}\sum_{n=1}^{N} x_n\right] = \frac{1}{N^2}\sum_{n=1}^{N}\sigma^2 + \frac{2}{N^2}\sum_{n<m}\operatorname{cov}[x_n, x_m]$$

By independence, this reduces to $\operatorname{var}[\bar{x}_N] = \sigma^2/N$, which converges to zero. □

Chapter 2

Linear Algebra

The first part of this chapter is mainly about solving systems of linear equations, while the second deals with random matrices. We start our story from the beginning, with the notions vectors and matrices.

2.1 Vectors and Matrices

[roadmap]

2.1.1 Vectors

An important set for us will be, for arbitrary $N \in \mathbb{N}$, the set of all N-vectors, or vectors of length N. This set is denoted by $\mathbb{R}^N$, and a typical element is of the form

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \quad \text{where } x_n \in \mathbb{R} \text{ for each } n$$

Here $\mathbf{x}$ has been written vertically, as a column of numbers. We could also write $\mathbf{x}$ horizontally, like so: $\mathbf{x} = (x_1, \ldots, x_N)$. At this stage, we are viewing vectors just as sequences of numbers, so it makes no difference whether they are written vertically or horizontally. Later, when we come to deal with *matrix* algebra, we will distinguish between column (vertical) and row (horizontal) vectors.

The vector of ones will be denoted **1**, while the vector of zeros will be denoted **0**:

$$\mathbf{1} := \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \qquad \mathbf{0} := \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

For elements of $\mathbb{R}^N$ there are two fundamental algebraic operations: addition and scalar multiplication. If $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^N$, then the **sum** is defined by

$$\mathbf{x} + \mathbf{y} :=: \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} + \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} := \begin{pmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_N + y_N \end{pmatrix}$$

If $\alpha \in \mathbb{R}$, then the **scalar product** of α and $\mathbf{x}$ is defined to be

$$\alpha\mathbf{x} := \begin{pmatrix} \alpha x_1 \\ \alpha x_2 \\ \vdots \\ \alpha x_N \end{pmatrix}$$

Thus, addition and scalar multiplication are defined in terms of ordinary addition and multiplication in $\mathbb{R}$, and computed element-by-element, by adding and multiplying respectively. Figures 2.1 and 2.1 show examples of vector addition and scalar multiplication in the case $N = 2$. In the figure, vectors are represented as arrows, starting at the origin and ending at the location in $\mathbb{R}^2$ defined by the vector.

We have defined addition and scalar multiplication of vectors, but not subtraction. Subtraction is performed element by element, analogous to addition. The definition can be given in terms of addition and scalar multiplication. $\mathbf{x} - \mathbf{y} := \mathbf{x} + (-1)\mathbf{y}$. An illustration of this operation is given in figure 2.3. The way to remember this is to draw a line from $\mathbf{y}$ to $\mathbf{x}$, and then shift it to the origin.

The **inner product** of two vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^N$ is denoted by $\mathbf{x}'\mathbf{y}$, and defined as the sum of the products of their elements:

$$\mathbf{x}'\mathbf{y} := \sum_{n=1}^{N} x_n y_n = \mathbf{y}'\mathbf{x}$$

The (euclidean) **norm** of a vector $\mathbf{x} \in \mathbb{R}^N$ is defined as

$$\|\mathbf{x}\| := \sqrt{\mathbf{x}'\mathbf{x}} := \left(\sum_{n=1}^{N} x_n^2 \right)^{1/2} \tag{2.1}$$

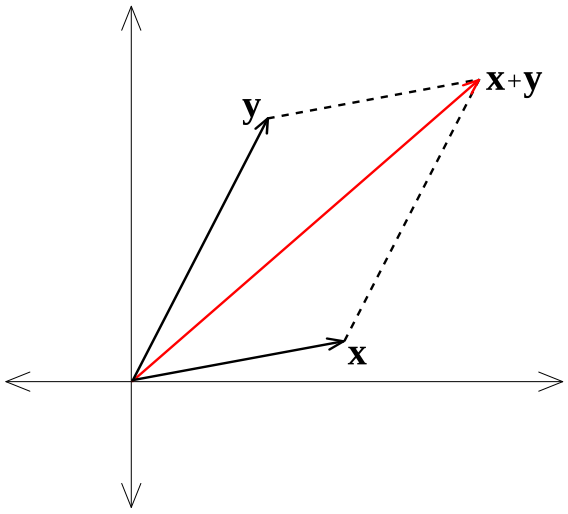


Figure 2.1: Vector addition

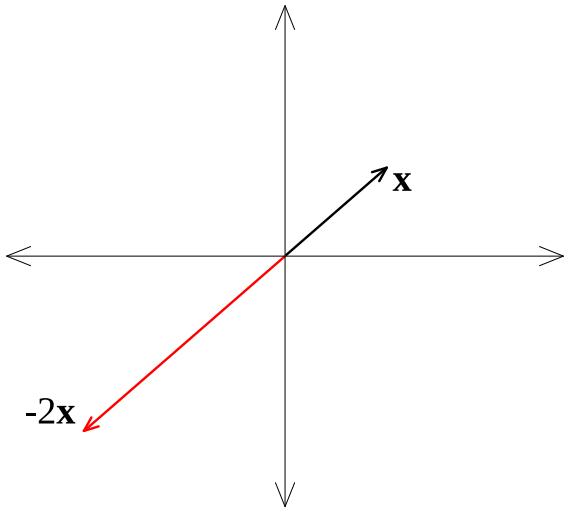


Figure 2.2: Scalar multiplication

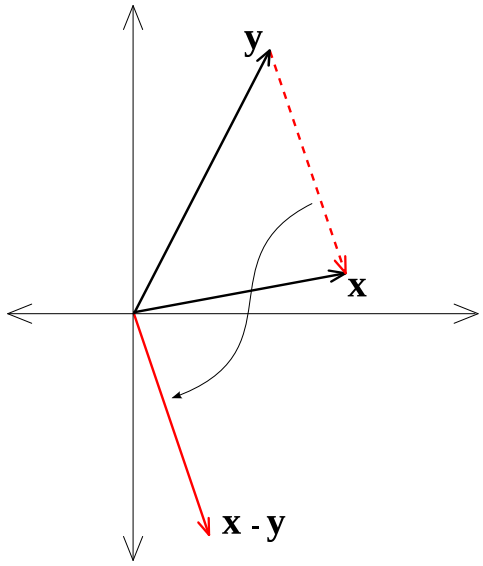


Figure 2.3: Difference between vectors

and represents the length of the vector $\mathbf{x}$. (In the arrow representation of vectors in figures 2.1–2.3, the norm of the vector is equal to the length of the arrow.)

Fact 2.1.1. For any $\alpha \in \mathbb{R}$ and any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$, the following properties are satisfied by the norm:

1. $\|\mathbf{x}\| \geq 0$ and $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = \mathbf{0}$.
2. $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|$.
3. $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$.
4. $|\mathbf{x}'\mathbf{y}| \leq \|\mathbf{x}\|\|\mathbf{y}\|$.

The third property is called the **triangle inequality**, while the fourth is called the **Cauchy-Schwartz inequality**.

Given two vectors $\mathbf{x}$ and $\mathbf{y}$, the value $\|\mathbf{x} - \mathbf{y}\|$ has the interpretation of being the "distance" between these points. To see why, consult figure 2.3 again.

2.1.2 Matrices

A $N \times K$ **matrix** is a rectangular array $\mathbf{A}$ of real numbers with N rows and K columns, written in the following way:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1K} \\ a_{21} & a_{22} & \cdots & a_{2K} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NK} \end{pmatrix}$$

Often, the values a_{nk} in the matrix represent coefficients in a system of linear equations, such as

$$\begin{gathered} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1K}x_K = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2K}x_K = b_2 \\ \vdots \\ a_{N1}x_1 + a_{N2}x_2 + \cdots + a_{NK}x_K = b_N \end{gathered}$$

We'll explore this relationship further after some more definitions.

In matrix $\mathbf{A}$, the symbol a_{nk} stands for the element in the n-th row of the k-th column. For obvious reasons, the matrix $\mathbf{A}$ is also called a vector if either $N = 1$ or $K = 1$. In the former case, $\mathbf{A}$ is called a **row vector**, while in the latter case it is called a **column vector**. If $\mathbf{A}$ is $N \times K$ and $N = K$, then $\mathbf{A}$ is called **square**. If, in addition $a_{nk} = a_{kn}$ for every k and n, then $\mathbf{A}$ is called **symmetric**.

When convenient, we will use the notation $\text{row}_n(\mathbf{A})$ to refer to the n-th row of $\mathbf{A}$, and $\text{col}_k(\mathbf{A})$ to refer to it's k-th column.

For a square matrix $\mathbf{A}$, the N elements of the form a_{nn} for $n = 1, \ldots, N$ are called the **principal diagonal**:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{pmatrix}$$

$\mathbf{A}$ is called **diagonal** if the only nonzero entries are on the principal diagonal. (Clearly every diagonal matrix is symmetric.) If, in addition to being diagonal, each element a_{nn} along the principal diagonal is equal to 1, then $\mathbf{A}$ is called the **identity**

matrix, and denoted by **I**:

$$\mathbf{I} := \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

Just as was the case for vectors, a number of algebraic operations are defined for matrices. The first two, scalar multiplication and addition, are immediate generalizations of the vector case: For $\gamma \in \mathbb{R}$, we let

$$\gamma \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1K} \\ a_{21} & a_{22} & \cdots & a_{2K} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NK} \end{pmatrix} := \begin{pmatrix} \gamma a_{11} & \gamma a_{12} & \cdots & \gamma a_{1K} \\ \gamma a_{21} & \gamma a_{22} & \cdots & \gamma a_{2K} \\ \vdots & \vdots & & \vdots \\ \gamma a_{N1} & \gamma a_{N2} & \cdots & \gamma a_{NK} \end{pmatrix}$$

while

$$\begin{pmatrix} a_{11} & \cdots & a_{1K} \\ a_{21} & \cdots & a_{2K} \\ \vdots & \vdots & \vdots \\ a_{N1} & \cdots & a_{NK} \end{pmatrix} + \begin{pmatrix} b_{11} & \cdots & b_{1K} \\ b_{21} & \cdots & b_{2K} \\ \vdots & \vdots & \vdots \\ b_{N1} & \cdots & b_{NK} \end{pmatrix} := \begin{pmatrix} a_{11} + b_{11} & \cdots & a_{1K} + b_{1K} \\ a_{21} + b_{21} & \cdots & a_{2K} + b_{2K} \\ \vdots & \vdots & \vdots \\ a_{N1} + b_{N1} & \cdots & a_{NK} + b_{NK} \end{pmatrix}$$

In the latter case, the matrices have to have the same number of rows and columns in order for the definition to make sense.

Now let's look at multiplication of matrices. If **A** and **B** are two matrices, then their product **AB** is formed by taking as it's i, j-th element the inner product of the i-th row of **A** and the j-th column of **B**. For example, consider the following product.

$$\begin{pmatrix} a_{11} & \cdots & a_{1K} \\ a_{21} & \cdots & a_{2K} \\ \vdots & \vdots & \vdots \\ a_{N1} & \cdots & a_{NK} \end{pmatrix} \begin{pmatrix} b_{11} & \cdots & b_{1J} \\ b_{21} & \cdots & b_{2J} \\ \vdots & \vdots & \vdots \\ b_{K1} & \cdots & b_{KJ} \end{pmatrix} = \begin{pmatrix} c_{11} & \cdots & c_{1J} \\ c_{21} & \cdots & c_{2J} \\ \vdots & \vdots & \vdots \\ c_{N1} & \cdots & c_{NJ} \end{pmatrix}$$

Here c_{11} is computed as

$$c_{11} = \text{row}_1(\mathbf{A})' \, \text{col}_1(\mathbf{B}) = \sum_{k=1}^{K} a_{1k} b_{k1}$$

There are many good tutorials for multiplying matrices on the web (try Wikipedia, for example), so I'll leave it to you to get a feeling for this operation.

Since inner products are only defined for vectors of equal length, this requires that the length of the rows of $\mathbf{A}$ is equal to the length of the columns of $\mathbf{B}$. Put differently, the number of columns of $\mathbf{A}$ is equal to the number of rows of $\mathbf{B}$. In other words, if $\mathbf{A}$ is $N \times K$ and $\mathbf{B}$ is $J \times M$, then we require $K = J$. The resulting matrix $\mathbf{AB}$ is $N \times M$. Here's the rule to remember:

$$\text{product of } N \times K \text{ and } K \times M \text{ is } N \times M$$

From the definition, it is clear that multiplication is not commutative, in that $\mathbf{AB}$ and $\mathbf{BA}$ are not generally the same thing. Indeed $\mathbf{BA}$ is not well-defined unless $N = M$ also holds. Even in this case, the two are not generally equal.

Other than that, multiplication behaves pretty much as we'd expect. In particular, for conformable matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, we have

- $\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$
- $\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$
- $(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC}$

(Here, we are using the word "conformable" to indicate dimensions are such that the operation in question makes sense. For example, we'll say "for two conformable matrices $\mathbf{A}$ and $\mathbf{B}$, the product $\mathbf{AB}$ satisfies xyz" if the dimensions of $\mathbf{A}$ and $\mathbf{B}$ are such that the product is well defined; and similarly for addition, etc.)

2.1.3 Linear Functions

One way to view matrices is as objects representing coefficients in linear systems of equations. Another way to view matrices is as *functions*, or mappings from one space to another. We will see that these two perspectives are very closely related. Let's begin with a definition: A function $f : \mathbb{R}^N \to \mathbb{R}^M$ is called **linear** if

$$f(\alpha\mathbf{x} + \beta\mathbf{y}) = \alpha f(\mathbf{x}) + \beta f(\mathbf{y}) \text{ for any } \mathbf{x}, \mathbf{y} \in \mathbb{R}^N \text{ and } \alpha, \beta \in \mathbb{R}$$

Example 2.1.1. The function $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = x^2$ is *non*linear, because if we take $\alpha = \beta = x = y = 1$, then $f(\alpha x + \beta y) = f(2) = 4$, while $\alpha f(x) + \beta f(y) = 1 + 1 = 2$.

Example 2.1.2. The function $f\colon \mathbb{R} \to \mathbb{R}$ defined by $f(x) = 2x$ is linear, because if we take any α, β, x, y in $\mathbb{R}$, then

$$f(\alpha x + \beta y) = 2(\alpha x + \beta y) = \alpha 2x + \beta 2y = \alpha f(x) + \beta f(y)$$

Example 2.1.3. The affine function $f\colon \mathbb{R} \to \mathbb{R}$ defined by $f(x) = 1 + 2x$ is *non*linear, because if we take $\alpha = \beta = x = y = 1$, then $f(\alpha x + \beta y) = f(2) = 5$, while $\alpha f(x) + \beta f(y) = 3 + 3 = 6$.

Now let's return to matrices. When we think of an $N \times K$ matrix $\mathbf{A}$ as a mapping, we are considering the operation of sending a vector $\mathbf{x} \in \mathbb{R}^K$ into a new vector $\mathbf{y} = \mathbf{A}\mathbf{x}$ in $\mathbb{R}^N$. In this sense, $\mathbf{A}$ defines a function from $\mathbb{R}^K$ to $\mathbb{R}^N$. Among the collection of all functions from $\mathbb{R}^K$ to $\mathbb{R}^N$, these functions defined by matrices have a special property: they are all linear. Moreover, it turns out that the functions defined by matrices are the *only* linear functions. In other words, the set of linear functions from $\mathbb{R}^K$ to $\mathbb{R}^N$ and the set of $N \times K$ matrices are essentially the same thing.

Let's look at some of the details. Take a fixed $N \times K$ matrix $\mathbf{A}$ and consider the function $f\colon \mathbb{R}^K \to \mathbb{R}^N$ defined by $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$. To see that f is a linear function, pick any $\mathbf{x}$, $\mathbf{y}$ in $\mathbb{R}^K$, and any scalars α and β. The rules of matrix addition and scalar multiplication tell us that

$$f(\alpha\mathbf{x} + \beta\mathbf{y}) := \mathbf{A}(\alpha\mathbf{x} + \beta\mathbf{y}) = \mathbf{A}\alpha\mathbf{x} + \mathbf{A}\beta\mathbf{y} = \alpha\mathbf{A}\mathbf{x} + \beta\mathbf{A}\mathbf{y} =: \alpha f(\mathbf{x}) + \beta f(\mathbf{y})$$

In other words, f is linear.

How about some more examples of a linear functions, to help you grasp the intuition? As I mentioned just above, I can't give you any more examples because there aren't any more. The next theorem states this result.

Theorem 2.1.1. *If $\mathbf{A}$ is an $N \times K$ matrix and $f\colon R^K \to \mathbb{R}^N$ is defined by $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$, then the function f is linear. Conversely, if $f\colon \mathbb{R}^K \to \mathbb{R}^N$ is linear, then there exists an $N \times K$ matrix $\mathbf{A}$ such that $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$ for all $\mathbf{x} \in \mathbb{R}^K$.*

The proof of the second part of theorem 2.1.1 is an exercise (exercise 2.6.3).

2.1.4 Maps and Linear Equations

In §2.1.3, we started to think about matrices as maps. That is, given an $N \times K$ matrix $\mathbf{A}$, we can identify $\mathbf{A}$ with the function $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$. Now let's think about linear

equations. The canonical problem is as follows: Given $\mathbf{A}$ and a fixed vector $\mathbf{b}$, we want to find a vector $\mathbf{x}$ such that $\mathbf{Ax} = \mathbf{b}$. The best way to think about this is that we want to invert the mapping $f(\mathbf{x}) = \mathbf{Ax}$, because what we want to do is find an $\mathbf{x}$ such that $f(\mathbf{x}) = \mathbf{b}$. That is, we want to find $f^{-1}(\mathbf{b})$, the preimage of $\mathbf{b}$ under f. Now if you know anything about inverting functions, you will know that there are various potential problems here. For one, there may be no $\mathbf{x}$ such that $f(\mathbf{x}) = \mathbf{b}$. Secondly, there may be multiple $\mathbf{x}$ with $f(\mathbf{x}) = \mathbf{b}$. These problems concern existence and uniqueness of solutions respectively.

Before tackling this problem directly, let's go back to the general problem of inverting functions in the one-dimensional case, the advantage being that we can graph the function and gain visual intuition. Consider figure 2.4, which graphs a one-dimensional function $f\colon [0,1] \to \mathbb{R}$. The set $[0,1]$ over which f is defined is called the **domain** of f. The red interval is called the **range** of f, and consists of all y such that $f(x) = y$ for some x in the domain $[0,1]$. More generally, for an arbitrary function $f\colon X \to Y$, the range of f is

$$\operatorname{rng}(f) := \{y \in Y : f(x) = y \text{ for some } x \in X\}$$

Returning to the problems of existence and uniqueness of solutions, have another look at the function f in figure 2.4. Evidently, the equation $f(x) = b$ has a solution if and only if $b \in \operatorname{rng}(f)$. In figure 2.5, b falls outside $\operatorname{rng}(f)$ and there is no solution. The other issue we must consider is uniqueness. Even if the equation $f(x) = b$ has a solution, the solution may not be unique. Figure 2.6 gives an example. Here both x_1 and x_2 solve $f(x) = b$.

Let's return now to the matrix setting. Let $f(\mathbf{x}) = \mathbf{Ax}$, where $\mathbf{A}$ is a given matrix, and consider the equation $\mathbf{Ax} = \mathbf{b}$ for some fixed $\mathbf{b}$. When will this equation have a solution? In view of our preceding discussion, the answer is that there will be a solution if and only if $\mathbf{b}$ is in $\operatorname{rng}(\mathbf{A}) := \operatorname{rng}(f)$. This is more likely if the range of f is "large." One way to check this by looking at something called the *rank* of $\mathbf{A}$, which is a measure of the size of its range. In turn, the rank is related to whether or not the columns of $\mathbf{A}$ are *linearly independent* or not. Conveniently, this last question turns out to be closely connected to the issue of uniqueness of solutions. To grasp these ideas takes a bit of effort but is certainly worthwhile. Let's get started.

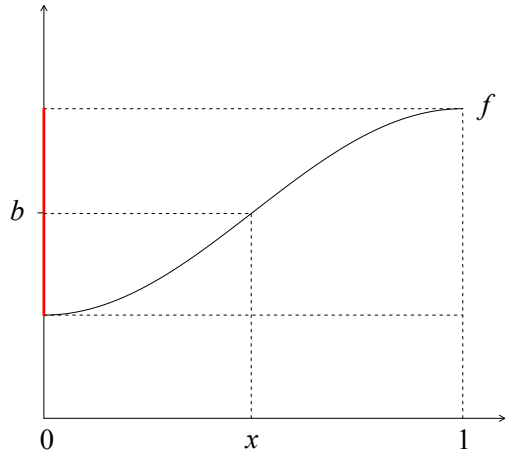


Figure 2.4: Preimage of b under f

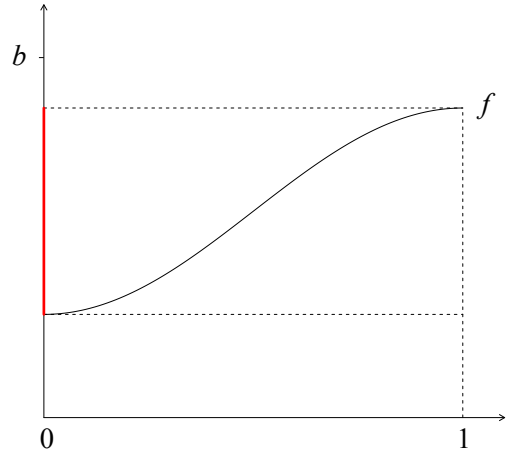


Figure 2.5: No solution

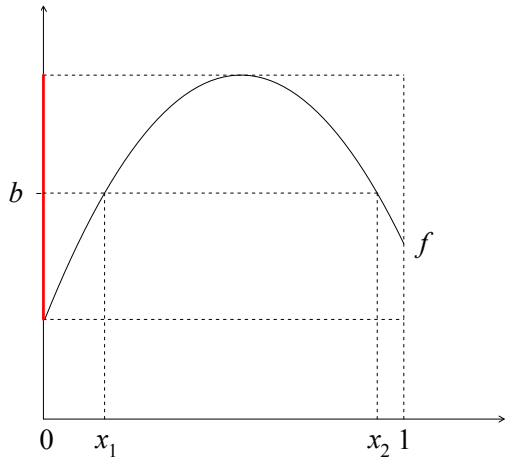


Figure 2.6: Multiple solutions

2.2 Span, Dimension and Independence

Motivated by the preceding discussion, we now introduce the important notions of linear subspaces, spans, linear independence, basis and dimension.

2.2.1 Spans and Linear Subspaces

Given K vectors $\mathbf{x}_1, \ldots, \mathbf{x}_K$ in $\mathbb{R}^N$, we can form **linear combinations**, which are vectors of the form

$$\mathbf{y} = \sum_{k=1}^{K} \alpha_k \mathbf{x}_k = \alpha_1 \mathbf{x}_1 + \cdots + \alpha_K \mathbf{x}_K$$

for some collection $\alpha_1, \ldots, \alpha_K$ of K real numbers.

Fact 2.2.1. Inner products of linear combinations satisfy the following rule:

$$\left(\sum_{k=1}^{K} \alpha_k \mathbf{x}_k \right)' \left(\sum_{j=1}^{J} \beta_j \mathbf{y}_j \right) = \sum_{k=1}^{K} \sum_{j=1}^{J} \alpha_k \beta_j \mathbf{x}_k' \mathbf{y}_j$$

The set of all linear combinations of $X := \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ is called the **span** of X, and denoted by $\text{span}(X)$:

$$\text{span}(X) := \left\{ \text{all vectors } \sum_{k=1}^{K} \alpha_k \mathbf{x}_k \text{ such that } \boldsymbol{\alpha} := (\alpha_1, \ldots, \alpha_K) \in \mathbb{R}^K \right\}$$

Let Y be any subset of $\mathbb{R}^N$, and let X be as above. If $Y \subset \text{span}(X)$, we say that the vectors $X := \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ span the set Y, or that X is a **spanning set** for Y. This is a particularly nice situation when Y is large but X is small, because it means that all the vectors in the large set Y are "described" by the small number of vectors in X.

Example 2.2.1. Let $X = \{\mathbf{1}\} = \{(1,1)\} \subset \mathbb{R}^2$. The span of X is all vectors of the form (α, α) with $\alpha \in \mathbb{R}$. This constitutes a line in the plane. Since we can take $\alpha = 0$, it follows that the origin $\mathbf{0}$ is in $\text{span}(X)$. In fact $\text{span}(X)$ is the unique line in the plane that passes through both $\mathbf{0}$ and the vector $\mathbf{1} = (1,1)$.

Example 2.2.2. Consider the vectors $\{\mathbf{e}_1, \ldots, \mathbf{e}_N\} \subset \mathbb{R}^N$, where $\mathbf{e}_n$ has all zeros except for a 1 as the n-th element. The case of $\mathbb{R}^2$, where $\mathbf{e}_1 := (1,0)$ and $\mathbf{e}_2 := (0,1)$, is illustrated in figure 2.7. The vectors $\mathbf{e}_1, \ldots, \mathbf{e}_N$ are called the **canonical basis vectors** of $\mathbb{R}^N$—we'll see why later on. One reason is that $\{\mathbf{e}_1, \ldots, \mathbf{e}_N\}$ spans all of $\mathbb{R}^N$. To see this in the case of $N = 2$ (check general N yourself), observe that for any $\mathbf{y} \in \mathbb{R}^2$, we have

$$\mathbf{y} := \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ y_1 \end{pmatrix} = y_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + y_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = y_1 \mathbf{e}_1 + y_2 \mathbf{e}_2$$

Thus, $\mathbf{y} \in \text{span}\{\mathbf{e}_1, \mathbf{e}_2\}$ as claimed. Since $\mathbf{y}$ is just an arbitrary vector in $\mathbb{R}^2$, we have shown that $\{\mathbf{e}_1, \mathbf{e}_2\}$ spans $\mathbb{R}^2$.

Example 2.2.3. Consider the set $P := \{(x_1, x_2, 0) \in \mathbb{R}^3 : x_1, x_2 \in \mathbb{R}\}$. Graphically, P corresponds to the flat plane in $\mathbb{R}^3$, where the height coordinate is always zero. If we take $\mathbf{e}_1 = (1,0,0)$ and $\mathbf{e}_2 = (0,1,0)$, then given $\mathbf{y} = (y_1, y_2, 0) \in P$ we have $\mathbf{y} = y_1 \mathbf{e}_1 + y_2 \mathbf{e}_2$. In other words, any $\mathbf{y} \in P$ can be expressed as a linear combination of $\mathbf{e}_1$ and $\mathbf{e}_2$, and $\{\mathbf{e}_1, \mathbf{e}_2\}$ is a spanning set for P.

Fact 2.2.2. Let X and Y be any two finite subsets of $\mathbb{R}^N$. If $X \subset Y$, then we have $\text{span}(X) \subset \text{span}(Y)$.

One of the key features of the span of a set X is that it is "closed" under the linear operations of vector addition and scalar multiplication, in the sense that if we take

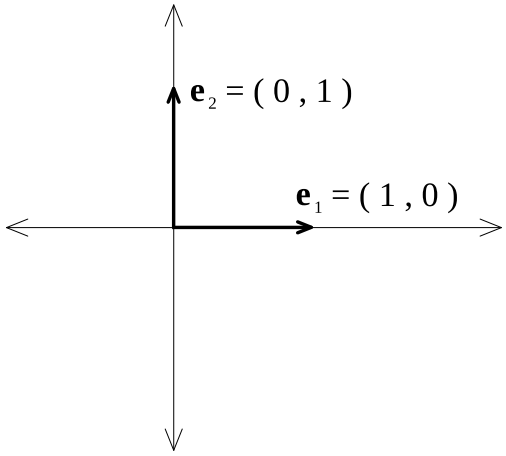


Figure 2.7: Canonical basis vectors in $\mathbb{R}^2$

elements of the span and combine them using these operations, the resulting vectors are still in the span. For example, to see that the span is closed under vector addition, observe that if $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ and $\mathbf{y}$, $\mathbf{z}$ are both in $\text{span}(X)$, then we can write them as

$$\mathbf{y} = \sum_{k=1}^{K} \alpha_k \mathbf{x}_k \quad \text{and} \quad \mathbf{z} = \sum_{k=1}^{K} \beta_k \mathbf{x}_k$$

for a suitable scalars $\alpha_k, \beta_k \in \mathbb{R}$. It then follows that

$$\mathbf{y} + \mathbf{z} = \sum_{k=1}^{K} (\alpha_k + \beta_k) \mathbf{x}_k \in \text{span}(X)$$

Hence $\text{span}(X)$ is closed under vector addition as claimed. Another easy argument shows that $\text{span}(X)$ is closed under scalar multiplication.

The notion of a set being closed under scalar multiplication and vector addition is important enough to have its own name: A set $S \subset \mathbb{R}^N$ with this property is called a linear subspace of $\mathbb{R}^N$. More succinctly, a nonempty subset S of $\mathbb{R}^N$ is called a **linear subspace** if, for any $\mathbf{x}$ and $\mathbf{y}$ in S, and any α and β in $\mathbb{R}$, the linear combination $\alpha\mathbf{x} + \beta\mathbf{y}$ is also in S.

Example 2.2.4. It follows immediately from the proceeding discussion that if X is any finite nonempty subset of $\mathbb{R}^N$, then $\text{span}(X)$ is a linear subspace of $\mathbb{R}^N$. For this reason, $\text{span}(X)$ is often called the **linear subspace spanned by X.**

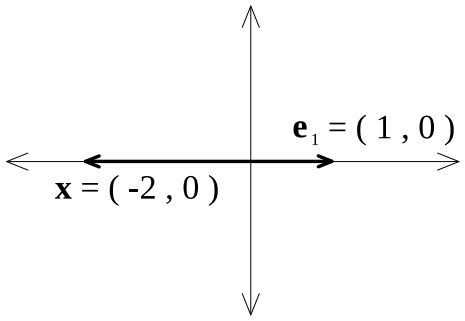


Figure 2.8: The vectors $\mathbf{e}_1$ and $\mathbf{x}$

Example 2.2.5. In $\mathbb{R}^3$, lines and planes that pass through the origin are linear subspaces. Other linear subspaces of $\mathbb{R}^3$ are the singleton set containing the zero element $\mathbf{0}$, and the set $\mathbb{R}^3$ itself.

Fact 2.2.3. Let S be a linear subspace of $\mathbb{R}^N$. The following statements are true:

1. The origin $\mathbf{0}$ is an element of S.
2. If X is a finite subset of S, then $\text{span}(X) \subset S$.

2.2.2 Linear Independence

In some sense, the span of $X := \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ is a measure of the "diversity" of the vectors in X—the more diverse are the elements of X, the greater is the set of vectors that can be represented as linear combinations of its elements. In fact, if X is not very diverse, then some "similar" elements may be redundant, in the sense that one can remove an element $\mathbf{x}_i$ from the collection X without reducing its span.

Let's consider two extremes. First consider the vectors $\mathbf{e}_1 := (1, 0)$ and $\mathbf{e}_2 := (0, 1)$ in $\mathbb{R}^2$ (figure 2.7). As we saw in example 2.2.2, the span $\{\mathbf{e}_1, \mathbf{e}_2\}$ is all of $\mathbb{R}^2$. With just these two vectors, we can span the whole plane. In algebraic terms, these vectors are relatively diverse. We can also see their diversity in the fact that if we remove one of the vectors from $\{\mathbf{e}_1, \mathbf{e}_2\}$, the span is no longer all of $\mathbb{R}^2$. In fact it is just a line in $\mathbb{R}^2$. Hence both vectors have their own role to play in forming the span.

Now consider the pair $\mathbf{e}_1$ and $\mathbf{x} := -2\mathbf{e}_1 = (-2, 0)$, as shown in figure 2.8. This pair is not very diverse. In fact, if $\mathbf{y} \in \text{span}\{\mathbf{e}_1, \mathbf{x}\}$, then, for some α_1 and α_2,

$$\mathbf{y} = \alpha_1\mathbf{e}_1 + \alpha_2\mathbf{x} = \alpha_1\mathbf{e}_1 + \alpha_2(-2)\mathbf{e}_1 = (\alpha_1 - 2\alpha_2)\mathbf{e}_1 \in \text{span}\{\mathbf{e}_1\}$$

In other words, any element of span$\{\mathbf{e}_1, \mathbf{x}\}$ is also an element of span$\{\mathbf{e}_1\}$. We can kick $\mathbf{x}$ out of the set $\{\mathbf{e}_1, \mathbf{x}\}$ without reducing the span.

Let's translate these ideas into formal definitions. In general, the set of vectors $X := \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ in $\mathbb{R}^N$ is called **linearly dependent** if one (or more) vector(s) can be removed without changing span(X). We call X **linearly independent** if it is not linearly dependent.

To see this definition in a slightly different light, suppose that $X := \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ is linearly dependent, with

$$\text{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_K\} = \text{span}\{\mathbf{x}_2, \ldots, \mathbf{x}_K\}$$

Since $\mathbf{x}_1 \in \text{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ certainly holds, this equality implies that

$$\mathbf{x}_1 \in \text{span}\{\mathbf{x}_2, \ldots, \mathbf{x}_K\}$$

Hence, there exist constants $\alpha_2, \ldots, \alpha_K$ with

$$\mathbf{x}_1 = \alpha_2 \mathbf{x}_2 + \cdots \alpha_K \mathbf{x}_K$$

In other words, $\mathbf{x}_1$ can be expressed as a linear combination of the other elements in X. This is a general rule: Linear dependence means that at least one vector in the set can be written as a linear combination of the others. Linear independence means the opposite is true. The following fact clarifies matters further:

Fact 2.2.4 (Definitions of linear independence)**.** The following statements are all equivalent:

1. The set $X := \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ is linearly independent.
2. If X_0 is a proper subset of X, then span(X_0) is a proper subset of span(X).[1]
3. No vector in X can be written as a linear combination of the others.
4. If $\sum_{k=1}^K \alpha_k \mathbf{x}_k = \mathbf{0}$, then $\alpha_1 = \alpha_2 = \cdots = \alpha_K = 0$.
5. If $\alpha_j \neq 0$ for some j, then $\sum_{k=1}^K \alpha_k \mathbf{x}_k \neq \mathbf{0}$.

[1] A is a proper subset of B if $A \subset B$ and $A \neq B$.

Part 5 is just the contrapositive of part 4, and hence the two are equivalent. (See §13.3 if you don't know what a contrapositive is.) The equivalence of part 4 and part 3 might not be immediately obvious, but the connection is clear when you think about it. To say that $(\alpha_1, \ldots, \alpha_K) = \mathbf{0}$ whenever $\sum_{k=1}^K \alpha_k \mathbf{x}_k = \mathbf{0}$ means precisely that no $\mathbf{x}_k$ can be written as a linear combination of the other vectors. For example, if there does exist some $\alpha_j \neq 0$ with $\sum_{k=1}^K \alpha_k \mathbf{x}_k = \mathbf{0}$, then $\mathbf{x}_j = \sum_{k \neq j} (-\alpha_k / \alpha_j) \mathbf{x}_k$.

Example 2.2.6. The set of canonical basis vectors in example 2.2.2 is linearly independent. Indeed, if $\alpha_j \neq 0$ for some j, then $\sum_{k=1}^K \alpha_k \mathbf{e}_k = (\alpha_1, \ldots, \alpha_K) \neq \mathbf{0}$.

One reason for our interest in the concept of linear independence lies in the following problem: We know when a point in $\mathbb{R}^N$ can be expressed as a linear combination of some fixed set of vectors X. This is true precisely when that point is in the span of X. What we do not know is when that representation is unique. It turns out that the relevant condition is independence:

Theorem 2.2.1. *Let $X := \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ be any collection of vectors in $\mathbb{R}^N$, and let $\mathbf{y}$ be any vector in $\operatorname{span}(X)$. If X is linearly independent, then there exists one and only one set of scalars $\alpha_1, \ldots, \alpha_K$ such that $\mathbf{y} = \sum_{k=1}^K \alpha_k \mathbf{x}_k$.*

Proof. Since $\mathbf{y}$ is in the span of X, we know that there exists at least one such set of scalars. Suppose now that there are two. In particular, suppose that

$$\mathbf{y} = \sum_{k=1}^K \alpha_k \mathbf{x}_k = \sum_{k=1}^K \beta_k \mathbf{x}_k$$

It follows from the second equality that $\sum_{k=1}^K (\alpha_k - \beta_k) \mathbf{x}_k = \mathbf{0}$. Using fact 2.2.4, we conclude that $\alpha_k = \beta_k$ for all k. In other words, the representation is unique. □

2.2.3 Dimension

In essence, the dimension of a linear subspace is the minimum number of vectors needed to span it. To understand this idea more clearly, let's look at an example. Consider the plane

$$P := \{(x_1, x_2, 0) \in \mathbb{R}^3 : x_1, x_2 \in \mathbb{R}\} \tag{2.2}$$

from example 2.2.3. Intuitively, this plane is a "two-dimensional" subset of $\mathbb{R}^3$. This intuition agrees with the definition above. Indeed, P cannot be spanned by one vector, for if we take a single vector in $\mathbb{R}^3$, then the span created by that singleton is

only a line in $\mathbb{R}^3$, not a plane. On the other hand, P *can* be spanned by two vectors, as we saw in example 2.2.3.

While P can also be spanned by three or more vectors, it turns out that one of the vectors will always be redundant—it does not change the span. In other words, any collection of 3 or more vectors from P will be linearly dependent. The following theorem contains the general statement of this idea:

Theorem 2.2.2. *If S is a linear subspace of $\mathbb{R}^N$ spanned by K vectors, then every linearly independent subset of S has at most K vectors.*

Put differently, if S is spanned by K vectors, then any subset of S with more than K vectors will be linearly dependent. This result is sometimes called the *exchange theorem*. The proof is not overly hard, but it is a little long. Readers keen to learn more will find it in most texts on linear algebra.

We now come to a key definition. If S is a linear subspace of $\mathbb{R}^N$ and B is some finite subset of $\mathbb{R}^N$, then B is called a **basis** of S if B spans S and is linearly independent.

Example 2.2.7. The pair $\{\mathbf{e}_1, \mathbf{e}_2\}$ is a basis for the set P defined in (2.2).

Example 2.2.8. Consider the set of canonical basis vectors $\{\mathbf{e}_1, \ldots, \mathbf{e}_N\} \subset \mathbb{R}^N$ described in example 2.2.8. This set is linearly independent, and its span is equal to all of $\mathbb{R}^N$. As a result, $\{\mathbf{e}_1, \ldots, \mathbf{e}_N\}$ is a basis for $\mathbb{R}^N$—as anticipated by the name.

Theorem 2.2.3. *If S is a linear subspace of $\mathbb{R}^N$, then every basis of S has the same number of elements.*

Proof. Let B_1 and B_2 be two bases of S, with K_1 and K_2 elements respectively. By definition, B_2 is a linearly independent subset of S. Moreover, S is spanned by the set B_1, which has K_1 elements. Applying theorem 2.2.2, we see that B_2 has at most K_1 elements. That is, $K_2 \leq K_1$. Repeating the same argument while reversing the roles of B_1 and B_2 we obtain $K_1 \leq K_2$. Hence $K_1 = K_2$. □

Theorem 2.2.3 states that if S is a linear subspace of $\mathbb{R}^N$, then every basis of S has the same number of elements. This common number is called the **dimension** of S, and written as $\dim(S)$. For example, if P is the plane in (2.2), then $\dim(P) = 2$, because the set $\{\mathbf{e}_1, \mathbf{e}_2\}$ is a basis, and this set contains two elements. The whole space $\mathbb{R}^N$ is N dimensional, because the canonical basis vectors form a basis, and there are N canonical basis vectors.

In $\mathbb{R}^3$, a line through the origin is a one-dimensional subspace, while a plane through the origin is a two-dimensional subspace.

Fact 2.2.5. The only N-dimensional linear subspace of $\mathbb{R}^N$ is $\mathbb{R}^N$.

If we take a set of K vectors, then how large will its span be in terms of dimension? The next lemma answers this question.

Lemma 2.2.1. *Let $X := \{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \subset \mathbb{R}^N$. Then*

1. $\dim(\text{span}(X)) \leq K$.
2. $\dim(\text{span}(X)) = K$ *if and only if X is linearly independent.*

Proof. Regarding part 1, let B be a basis of $\text{span}(X)$. By definition, B is a linearly independent subset of $\text{span}(X)$. Since $\text{span}(X)$ is spanned by K vectors, theorem 2.2.2 implies that B has no more than K elements. Hence, $\dim(\text{span}(X)) \leq K$.

Regarding part 2, suppose first that X is linearly independent. Then X is a basis for $\text{span}(X)$. Since X has K elements, we conclude that $\dim(\text{span}(X)) = K$.

Conversely, if $\dim(\text{span}(X)) = K$ then X must be linearly independent. For if X is not linearly independent, then exists a proper subset X_0 of X such that $\text{span}(X_0) = \text{span}(X)$. By part 1 of this theorem, we then have $\dim(\text{span}(X_0)) \leq \#X_0 \leq K - 1$. Therefore, $\dim(\text{span}(X)) \leq K - 1$. Contradiction. □

Part 2 of lemma 2.2.1 is important in what follows, and also rather intuitive. It says that the span of a set will be large when it's elements are algebraically diverse.

2.3 Matrices and Equations

[Roadmap]

2.3.1 Rank

Let's now connect matrices to our discussion of span, linear independence and dimension. We will be particularly interested in solving equations of the form $\mathbf{Ax} = \mathbf{b}$ for unknown $\mathbf{x}$. We take $\mathbf{A}$ to be an $N \times K$ matrix. As discussed in §2.1.3–2.1.4, we can view this matrix as a mapping $f(\mathbf{x}) = \mathbf{Ax}$ from $\mathbb{R}^K$ to $\mathbb{R}^N$. If $\mathbf{b} \in \mathbb{R}^N$ is given and we are looking for an $\mathbf{x}$ to solve $\mathbf{Ax} = \mathbf{b}$, then we know at least one such $\mathbf{x}$ will

exist if $\mathbf{b}$ is in the range of f. Since $\mathbf{A}$ and f are essentially the same thing, we will denote the range by $\text{rng}(\mathbf{A})$ instead of $\text{rng}(f)$. That is,

$$\text{rng}(\mathbf{A}) := \{\mathbf{A}\mathbf{x} : \mathbf{x} \in \mathbb{R}^K\}$$

Just a little bit of thought will convince you that this is precisely the span of the columns of $\mathbf{A}$:

$$\text{rng}(\mathbf{A}) = \text{span}(\text{col}_1(\mathbf{A}), \ldots, \text{col}_K(\mathbf{A}))$$

For obvious reasons, this set is sometimes called the **column space** of $\mathbf{A}$. Being defined as a span, it is obviously a linear subspace of $\mathbb{R}^N$.

As stated above, for the system $\mathbf{A}\mathbf{x} = \mathbf{b}$ to have a solution, we require that $\mathbf{b} \in \text{rng}(\mathbf{A})$. If we want to check that this is true, we'll probably be wanting to check that $\text{rng}(\mathbf{A})$ is suitably "large." The obvious measure of size for a linear subspace such as $\text{rng}(\mathbf{A})$ is its dimension. The dimension of $\text{rng}(\mathbf{A})$ is known as the **rank** of $\mathbf{A}$. That is,

$$\text{rank}(\mathbf{A}) := \dim(\text{rng}(\mathbf{A}))$$

Furthermore, $\mathbf{A}$ is said to have **full column rank** if $\text{rank}(\mathbf{A})$ is equal to K, the number of its columns. Why do we say "full" rank here? Because, by definition, $\text{rng}(\mathbf{A})$ is the span by K vectors, and hence, by part 1 of lemma 2.2.1, we have $\dim(\text{rng}(\mathbf{A})) \leq K$. In other words, the rank of $\mathbf{A}$ is less than or equal to K. $\mathbf{A}$ is said to have full column rank when this maximum is achieved.

When is this maximum achieved? By part 2 of lemma 2.2.1, this while be the case precisely when the columns of $\mathbf{A}$ are linearly independent. Thus, the matrix $\mathbf{A}$ is of full column rank if and only if the columns of $\mathbf{A}$ are linearly independent. By fact 2.2.4 on page 63, the next characterization is also equivalent.

Fact 2.3.1. $\mathbf{A}$ is of full column rank if and only if the only $\mathbf{x}$ satisfying $\mathbf{A}\mathbf{x} = \mathbf{0}$ is $\mathbf{x} = \mathbf{0}$.

Let's return to the problem of solving the system $\mathbf{A}\mathbf{x} = \mathbf{b}$ for some fixed $\mathbf{b} \in \mathbb{R}^N$. For existence of a solution we need $\mathbf{b} \in \text{rng}(\mathbf{A})$, and this range will be large when $\mathbf{A}$ is full column rank. So the property of $\mathbf{A}$ being full column rank will be connected to the problem of existence. Even better, the full column rank condition is exactly what we need for uniqueness as well, as follows immediately from theorem 2.2.1. In matrix terminology, theorem 2.2.1 translates to the following result:

Fact 2.3.2. If $\mathbf{A}$ has full column rank and $\mathbf{b} \in \text{rng}(\mathbf{A})$, then the system of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$ has a unique solution.

2.3.2 Square Matrices

Now let $N = K$, so that $\mathbf{A}$ is a square $N \times N$ matrix. Suppose that $\mathbf{A}$ is full column rank, so that its columns are independent. In that case, fact 2.2.5 on page 66 implies that $\text{rng}(\mathbf{A})$ is equal to $\mathbb{R}^N$, because $\text{rng}(\mathbf{A})$ is a linear subspace of $\mathbb{R}^N$ by definition, and its dimension is N by the full column rank assumption.

Since $\text{rng}(\mathbf{A}) = \mathbb{R}^N$, it follows immediately that, for any $\mathbf{b} \in \mathbb{R}^N$, the system $\mathbf{Ax} = \mathbf{b}$ has a solution. Moreover, by fact 2.3.2, the solution is unique. To repeat, if $\mathbf{A}$ is square and full column rank, then for any $\mathbf{b} \in \mathbb{R}^N$, the there is a unique $\mathbf{x} \in \mathbb{R}^N$ such that $\mathbf{Ax} = \mathbf{b}$.

Since this problem is so important, there are several different ways of describing it. First, a square matrix $\mathbf{A}$ is called **invertible** if there exists a second square matrix $\mathbf{B}$ such that $\mathbf{AB} = \mathbf{BA} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. If this is the case, then $\mathbf{B}$ is called the **inverse** of $\mathbf{A}$, and written as $\mathbf{A}^{-1}$. Invertibility of $\mathbf{A}$ is exactly equivalent to the existence of a unique solution $\mathbf{x}$ to the system $\mathbf{Ax} = \mathbf{b}$ for all $\mathbf{b} \in \mathbb{R}^N$. Indeed, the existence of an inverse $\mathbf{A}^{-1}$ means that we can take our solution as

$$\mathbf{x}_b = \mathbf{A}^{-1}\mathbf{b} \tag{2.3}$$

since $\mathbf{Ax}_b = \mathbf{AA}^{-1}\mathbf{b} = \mathbf{Ib} = \mathbf{b}$.

In addition, to each square matrix $\mathbf{A}$, we can associate a unique number $\det(\mathbf{A})$ called the **determinant** of $\mathbf{A}$. The determinant crops up in many ways, but the general definition is a bit fiddly to state and hence we omit it. (Look it up if you are interested, but note that it won't be required in what follows.) If $\det(\mathbf{A}) = 0$, then $\mathbf{A}$ is called **singular**. Otherwise $\mathbf{A}$ is called **nonsingular**.

It turns out that nonsingularity is also equivalent to invertibility. Let's summarize our discussion:

Fact 2.3.3. For $N \times N$ matrix $\mathbf{A}$, the following are equivalent:

1. $\mathbf{A}$ is of full column rank.
2. The columns of $\mathbf{A}$ are linearly independent.
3. $\mathbf{A}$ is invertible.
4. $\mathbf{A}$ is nonsingular.
5. For each $\mathbf{b} \in \mathbb{R}^N$, the system $\mathbf{Ax} = \mathbf{b}$ has unique solution $\mathbf{A}^{-1}\mathbf{b}$.

The next fact collects useful results about the inverse.

Fact 2.3.4. If $\mathbf{A}$ and $\mathbf{B}$ are invertible and $\alpha \neq 0$, then

1. $\det(\mathbf{A}^{-1}) = (\det(\mathbf{A}))^{-1}$,
2. $(\mathbf{A}^{-1})^{-1} = \mathbf{A}$,
3. $(\alpha\mathbf{A})^{-1} = \alpha^{-1}\mathbf{A}^{-1}$, and
4. $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$.

2.3.3 Other Properties of Matrices

The **transpose** of $N \times K$ matrix $\mathbf{A}$ is a $K \times N$ matrix $\mathbf{A}'$ such that the first column of $\mathbf{A}'$ is the first row of $\mathbf{A}$, the second column of $\mathbf{A}'$ is the second row of $\mathbf{A}$, and so on. For example, given

$$\mathbf{A} := \begin{pmatrix} 10 & 40 \\ 20 & 50 \\ 30 & 60 \end{pmatrix} \qquad \mathbf{B} := \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix} \tag{2.4}$$

the transposes are

$$\mathbf{A}' = \begin{pmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \end{pmatrix} \qquad \mathbf{B}' := \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

Fact 2.3.5. For conformable matrices $\mathbf{A}$ and $\mathbf{B}$, transposition satisfies the following:

1. $(\mathbf{AB})' = \mathbf{B}'\mathbf{A}'$
2. $(\mathbf{A} + \mathbf{B})' = \mathbf{A}' + \mathbf{B}'$
3. $(c\mathbf{A})' = c\mathbf{A}'$ for any constant c.

Note that a square matrix $\mathbf{C}$ is symmetric precisely when $\mathbf{C}' = \mathbf{C}$. Note also that $\mathbf{A}'\mathbf{A}$ and $\mathbf{AA}'$ are well-defined and symmetric.

Fact 2.3.6. For each square matrix $\mathbf{A}$, we have

1. $\det(\mathbf{A}') = \det(\mathbf{A})$, and
2. $(\mathbf{A}^{-1})' = (\mathbf{A}')^{-1}$ whenever the inverse exists.

The **trace** of a square matrix is the sum of the elements on its principal diagonal. That is,

$$\operatorname{trace} \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{pmatrix} = \sum_{n=1}^{N} a_{nn}$$

Fact 2.3.7. Transposition does not alter trace: $\operatorname{trace}(\mathbf{A}) = \operatorname{trace}(\mathbf{A}')$.

Fact 2.3.8. If $\mathbf{A}$ and $\mathbf{B}$ are $N \times N$ matrices and α and β are two scalars, then

$$\operatorname{trace}(\alpha\mathbf{A} + \beta\mathbf{B}) = \alpha \operatorname{trace}(\mathbf{A}) + \beta \operatorname{trace}(\mathbf{B})$$

Moreover, if $\mathbf{A}$ is $N \times M$ and $\mathbf{B}$ is $M \times N$, then $\operatorname{trace}(\mathbf{AB}) = \operatorname{trace}(\mathbf{BA})$.

The rank of a matrix can be difficult to determine. One case where it is easy is where the matrix is idempotent. A square matrix $\mathbf{A}$ is called **idempotent** if $\mathbf{AA} = \mathbf{A}$.

Fact 2.3.9. If $\mathbf{A}$ is idempotent, then $\operatorname{rank}(\mathbf{A}) = \operatorname{trace}(\mathbf{A})$.

2.3.4 Quadratic Forms

Let $\mathbf{A}$ be $N \times N$ and symmetric, and let $\mathbf{x}$ be $N \times 1$. The expression

$$\mathbf{x}'\mathbf{A}\mathbf{x} = \sum_{j=1}^{N} \sum_{i=1}^{N} a_{ij} x_i x_j$$

is called a **quadratic form** in $\mathbf{x}$. Notice that if $\mathbf{A} = \mathbf{I}$, then this reduces to $\|\mathbf{x}\|^2$, which is positive whenever $\mathbf{x}$ is nonzero. The next two definitions generalize this idea: An $N \times N$ symmetric matrix $\mathbf{A}$ is called

- **nonnegative definite** if $\mathbf{x}'\mathbf{A}\mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^N$, and
- **positive definite** if $\mathbf{x}'\mathbf{A}\mathbf{x} > 0$ for all $\mathbf{x} \in \mathbb{R}^N$ with $\mathbf{x} \neq \mathbf{0}$.

As we have just seen, the identity matrix is positive definite.

Fact 2.3.10. If $\mathbf{A}$ is nonnegative definite, then each element a_{nn} on the principal diagonal is nonnegative.

Fact 2.3.11. If $\mathbf{A}$ is positive definite, then:

1. Each element a_{nn} on the principal diagonal is positive.
2. $\mathbf{A}$ is full column rank and invertible, with $\det(\mathbf{A}) > 0$.

To see that positive definiteness implies full column rank, consider the following argument: If $\mathbf{A}$ is positive definite, then $\mathbf{A}$ must be full column rank, for if not there exists a $\mathbf{x} \neq \mathbf{0}$ with $\mathbf{Ax} = \mathbf{0}$ (fact 2.3.1). But then $\mathbf{x}'\mathbf{Ax} = 0$ for nonzero $\mathbf{x}$. This contradicts the definition of positive definiteness.

2.4 Random Vectors and Matrices

A **random vector** $\mathbf{x}$ is just a sequence of K random variables $(x_1, \ldots, x_K)$. Each realization of $\mathbf{x}$ is an element of $\mathbb{R}^K$. The distribution (or cdf) of $\mathbf{x}$ is the joint distribution F of $(x_1, \ldots, x_K)$. That is,

$$F(\mathbf{s}) :=: F(s_1, \ldots, s_K) := \mathbb{P}\{x_1 \leq s_1, \ldots, x_K \leq s_K\} :=: \mathbb{P}\{\mathbf{x} \leq \mathbf{s}\} \tag{2.5}$$

for each $\mathbf{s}$ in $\mathbb{R}^K$. (Here and in what follows, the statement $\mathbf{x} \leq \mathbf{s}$ means that $x_n \leq s_n$ for $n = 1, \ldots, K$.)

Just as some but not all distributions on $\mathbb{R}$ have a density representation (see §1.2.2), some but not all distributions on $\mathbb{R}^K$ can be represented by a density. We say that $f : \mathbb{R}^K \to \mathbb{R}$ is the density of random vector $\mathbf{x} := (x_1, \ldots, x_K)$ if

$$\int_B f(\mathbf{s})\, d\mathbf{s} = \mathbb{P}\{\mathbf{x} \in B\} \tag{2.6}$$

for every subset B of $\mathbb{R}^K$.[2] Most of the distributions we work with in this course have density representations.

[2] Actually, some subsets of $\mathbb{R}^K$ are so messy that it's not possible to integrate over them, so we only require (2.6) to hold for a large but suitably well-behaved class of sets called the *Borel* sets. See any text on measure theory for details.

For random vectors, the definition of independence mirrors the scalar case. In particular, a collection of random vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$ is called **independent** if, given any $\mathbf{s}_1, \ldots, \mathbf{s}_N$, we have

$$\mathbb{P}\{\mathbf{x}_1 \leq \mathbf{s}_1, \ldots, \mathbf{x}_N \leq \mathbf{s}_N\} = \mathbb{P}\{\mathbf{x}_1 \leq \mathbf{s}_1\} \times \cdots \times \mathbb{P}\{\mathbf{x}_N \leq \mathbf{s}_N\}$$

We note the following multivariate version of fact 1.3.2:

Fact 2.4.1. If $\mathbf{x}$ and $\mathbf{y}$ are independent and g and f are any functions, then $f(\mathbf{x})$ and $g(\mathbf{y})$ are also independent.

A **random $N \times K$ matrix** $\mathbf{X}$ is a rectangular $N \times K$ array of random variables. In this section, we briefly review some properties of random vectors and matrices.

2.4.1 Expectations for Vectors and Matrices

Let $\mathbf{x} = (x_1, \ldots, x_K)$ be a random vector taking values in $\mathbb{R}^K$ with $\mu_k := \mathbb{E}[x_k]$ for all $k = 1, \ldots, K$. The **expectation** $\mathbb{E}[\mathbf{x}]$ of vector $\mathbf{x}$ is defined as the vector of expectations:

$$\mathbb{E}[\mathbf{x}] := \begin{pmatrix} \mathbb{E}[x_1] \\ \mathbb{E}[x_2] \\ \vdots \\ \mathbb{E}[x_K] \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_K \end{pmatrix} =: \boldsymbol{\mu}$$

More generally, if $\mathbf{X}$ is a random $N \times K$ matrix, then its expectation $\mathbb{E}[\mathbf{X}]$ is the matrix of the expectations:

$$\mathbb{E}[\mathbf{X}] := \begin{pmatrix} \mathbb{E}[x_{11}] & \mathbb{E}[x_{12}] & \cdots & \mathbb{E}[x_{1K}] \\ \mathbb{E}[x_{21}] & \mathbb{E}[x_{22}] & \cdots & \mathbb{E}[x_{2K}] \\ \vdots & \vdots & & \vdots \\ \mathbb{E}[x_{N1}] & \mathbb{E}[x_{N2}] & \cdots & \mathbb{E}[x_{NK}] \end{pmatrix}$$

Expectation of vectors and matrices maintains the linearity of scalar expectations:

Fact 2.4.2. If $\mathbf{X}$ and $\mathbf{Y}$ are random and $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are conformable constant matrices, then

$$\mathbb{E}[\mathbf{A} + \mathbf{B}\mathbf{X} + \mathbf{C}\mathbf{Y}] = \mathbf{A} + \mathbf{B}\mathbb{E}[\mathbf{X}] + \mathbf{C}\mathbb{E}[\mathbf{Y}]$$

The **covariance** between random $N \times 1$ vectors $\mathbf{x}$ and $\mathbf{y}$ is

$$\operatorname{cov}[\mathbf{x}, \mathbf{y}] := \mathbb{E}\left[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{y} - \mathbb{E}[\mathbf{y}])'\right]$$

The **variance-covariance matrix** of random vector $\mathbf{x}$ with $\boldsymbol{\mu} := \mathbb{E}[\mathbf{x}]$ is defined as

$$\operatorname{var}[\mathbf{x}] := \operatorname{cov}(\mathbf{x}, \mathbf{x}) = \mathbb{E}\left[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])'\right] = \mathbb{E}\left[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})'\right]$$

Expanding this out, we get

$$\operatorname{var}[\mathbf{x}] = \begin{pmatrix} \mathbb{E}[(x_1 - \mu_1)(x_1 - \mu_1)] & \cdots & \mathbb{E}[(x_1 - \mu_1)(x_N - \mu_N)] \\ \mathbb{E}[(x_2 - \mu_2)(x_1 - \mu_1)] & \cdots & \mathbb{E}[(x_2 - \mu_2)(x_N - \mu_N)] \\ \vdots & \vdots & \vdots \\ \mathbb{E}[(x_N - \mu_N)(x_1 - \mu_1)] & \cdots & \mathbb{E}[(x_N - \mu_N)(x_N - \mu_N)] \end{pmatrix}$$

The j,k-th term is the scalar covariance between x_j and x_k. As a result, the principle diagonal contains the variance of each x_n.

Some simple algebra yields the alternative expressions

$$\operatorname{cov}[\mathbf{x}, \mathbf{y}] = \mathbb{E}[\mathbf{x}\mathbf{y}'] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}]' \quad \text{and} \quad \operatorname{var}[\mathbf{x}] = \mathbb{E}[\mathbf{x}\mathbf{x}'] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{x}]'$$

Fact 2.4.3. For any random vector $\mathbf{x}$, the variance-covariance matrix $\operatorname{var}[\mathbf{x}]$ is square, symmetric and nonnegative definite.

Fact 2.4.4. For any random vector $\mathbf{x}$, any constant conformable matrix $\mathbf{A}$ and any constant conformable vector $\mathbf{a}$, we have

$$\operatorname{var}[\mathbf{a} + \mathbf{A}\mathbf{x}] = \mathbf{A} \operatorname{var}[\mathbf{x}] \mathbf{A}'$$

2.4.2 Multivariate Gaussians

The **multivariate normal density** or **Gaussian density** in $\mathbb{R}^N$ is a function p of the form

$$p(\mathbf{s}) = (2\pi)^{-N/2} \det(\boldsymbol{\Sigma})^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{s} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{s} - \boldsymbol{\mu})\right\}$$

where $\boldsymbol{\mu}$ is any $N \times 1$ vector and $\boldsymbol{\Sigma}$ is a symmetric, positive definite $N \times N$ matrix. In symbols, we represent this distribution by $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Although we omit the derivations, it can be shown that if $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu} \quad \text{and} \quad \operatorname{var}[\mathbf{x}] = \boldsymbol{\Sigma}$$

We say that $\mathbf{x}$ is **normally distributed** if $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for some $N \times 1$ vector $\boldsymbol{\mu}$ and symmetric, positive definite $N \times N$ matrix $\boldsymbol{\Sigma}$. We say that $\mathbf{x}$ is **standard normal** if $\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{\Sigma} = \mathbf{I}$.

Fact 2.4.5. $N \times 1$ random vector $\mathbf{x}$ is normally distributed if and only if $\mathbf{a}'\mathbf{x}$ is normally distributed in $\mathbb{R}$ for every constant $N \times 1$ vector $\mathbf{a}$.[3]

Fact 2.4.6. If $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then $\mathbf{a} + \mathbf{A}\mathbf{x} \sim \mathcal{N}(\mathbf{a} + \mathbf{A}\boldsymbol{\mu}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}')$.

Here, the fact that $\mathbf{a} + \mathbf{A}\mathbf{x}$ has mean $\mathbf{a} + \mathbf{A}\boldsymbol{\mu}$ and variance-covariance matrix $\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}'$ is not surprising. What is important is that normality is preserved .

Fact 2.4.7. Normally distributed random variables are independent if and only if they are uncorrelated. In particular, if both x and y are normally distributed and $\operatorname{cov}[x, y] = 0$, then x and y are independent.

Fact 2.4.8. If $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then $(\mathbf{x} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \sim \chi^2(k)$, where $k :=$ length of $\mathbf{x}$.

Fact 2.4.9. If $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\mathbf{A}$ is a conformable idempotent and symmetric matrix with $\operatorname{rank}(\mathbf{A}) = j$, then $\mathbf{x}'\mathbf{A}\mathbf{x} \sim \chi^2(j)$. (In view of fact 2.3.9, when using this result it is sufficient to show that $\operatorname{trace}(\mathbf{A}) = j$.)

2.5 Convergence of Random Matrices

As a precursor to time series analyais, we extend the probabilistic notions of convergence discussed in §1.4.1 to random vectors and matrices.

2.5.1 Convergence in Probability

We already have a notion of scalar convergence in probability (see §1.4.1). Extending this to the matrix case, let $\{\mathbf{X}_n\}_{n=1}^{\infty}$ be a sequence of random $I \times J$ matrices. We say that $\mathbf{X}_n$ converges to a random $I \times J$ matrix $\mathbf{X}$ **in probability** (and write $\mathbf{X}_n \xrightarrow{p} \mathbf{X}$) if every element of $\mathbf{X}_n$ converges to the corresponding element of $\mathbf{X}$ in probability in the scalar sense. That is,

$$\mathbf{X}_n \xrightarrow{p} \mathbf{X} \text{ whenever } x_{ij}^n \xrightarrow{p} x_{ij} \text{ for all } i \text{ and } j$$

Here x_{ij}^n is the i, j-th element of $\mathbf{X}_n$ and x_{ij} is the i, j-th element of $\mathbf{X}$.

[3] If $\mathbf{a} = \mathbf{0}$ then we can interpret $\mathbf{a}'\mathbf{x}$ as a "normal" random variable with zero variance.

If we are dealing with vectors ($I = 1$ or $J = 1$ in the previous definition), then the condition for convergence has the form

$$\mathbf{x}_n := \begin{pmatrix} x_1^n \\ \vdots \\ x_K^n \end{pmatrix} \xrightarrow{p} \begin{pmatrix} x_1 \\ \vdots \\ x_K \end{pmatrix} =: \mathbf{x} \text{ whenever } x_k^n \xrightarrow{p} x_k \text{ for all } k$$

With vectors, we can also consider norm deviation. In this connection, we have the following result.

Fact 2.5.1. If $\{\mathbf{x}_n\}$ is a sequence of random vectors in $\mathbb{R}^K$ and $\mathbf{x}$ is a random vector in $\mathbb{R}^K$, then

$$\mathbf{x}_n \xrightarrow{p} \mathbf{x} \text{ if and only if } \|\mathbf{x}_n - \mathbf{x}\| \xrightarrow{p} 0$$

In other words, each element of $\mathbf{x}_n$ converges in probability to the corresponding element of $\mathbf{x}$ if and only if the norm distance between the vectors goes to zero in probability. Although fact 2.5.1 is stated in terms of vectors, the same result is in fact true for matrices if we regard matrices as vectors. In other words, if we take an $N \times K$ matrix $\mathbf{A}$, we can think of it as a vector in $\mathbb{R}^{N \times K} = \mathbb{R}^{NK}$ by stacking all the columns into one long column, or rows into one long row—it doesn't matter which. Thinking of matrices this way, fact 2.5.1 is applicable: $\mathbf{X}_n \xrightarrow{p} \mathbf{X}$ if and only if $\|\mathbf{X}_n - \mathbf{X}\| \xrightarrow{p} 0$.[4]

Fact 2.5.2. Assuming conformability, the following statements are true:

1. If $\mathbf{X}_n \xrightarrow{p} \mathbf{X}$ and $\mathbf{X}_n$ and $\mathbf{X}$ are invertible, then $\mathbf{X}_n^{-1} \xrightarrow{p} \mathbf{X}^{-1}$.
2. If $\mathbf{X}_n \xrightarrow{p} \mathbf{X}$ and $\mathbf{Y}_n \xrightarrow{p} \mathbf{Y}$, then
$$\mathbf{X}_n + \mathbf{Y}_n \xrightarrow{p} \mathbf{X} + \mathbf{Y}, \quad \mathbf{X}_n\mathbf{Y}_n \xrightarrow{p} \mathbf{XY} \quad \text{and} \quad \mathbf{Y}_n\mathbf{X}_n \xrightarrow{p} \mathbf{YX}$$
3. If $\mathbf{X}_n \xrightarrow{p} \mathbf{X}$ and $\mathbf{A}_n \to \mathbf{A}$, then
$$\mathbf{X}_n + \mathbf{A}_n \xrightarrow{p} \mathbf{X} + \mathbf{A}, \quad \mathbf{X}_n\mathbf{A}_n \xrightarrow{p} \mathbf{XA} \quad \text{and} \quad \mathbf{A}_n\mathbf{X}_n \xrightarrow{p} \mathbf{AX}$$

In part 3 of fact 2.5.2, the matrices $\mathbf{A}_n$ and $\mathbf{A}$ are nonrandom. The convergence $\mathbf{A}_n \to \mathbf{A}$ means that each element of $\mathbf{A}_n$ converges in the usual scalar sense to the corresponding element of $\mathbf{A}$:

$$\mathbf{A}_n \to \mathbf{A} \text{ means } a_{ij}^n \to a_{ij} \text{ for all } i \text{ and } j$$

[4] There are various notions of matrix norms. The one defined here is called the **Frobenius norm**.

Alternatively, we can stack the matrices into vectors and take the norms, as discussed above. Then we say that $\mathbf{A}_n \to \mathbf{A}$ if $\|\mathbf{A}_n - \mathbf{A}\| \to 0$. The two definitions can be shown to be equivalent.

As an example of how fact 2.5.2 can be used, let's establish convergence of the quadratic form

$$\mathbf{a}'\mathbf{X}_n\mathbf{a} \xrightarrow{p} \mathbf{a}'\mathbf{X}\mathbf{a} \quad \text{whenever } \mathbf{a} \text{ is a conformable constant vector and } \mathbf{X}_n \xrightarrow{p} \mathbf{X} \tag{2.7}$$

This follows from two applications of fact 2.5.2. Applying fact 2.5.2 once we get $\mathbf{a}'\mathbf{X}_n \xrightarrow{p} \mathbf{a}'\mathbf{X}$. Applying it a second time yields the convergence in (2.7).

2.5.2 Convergence in Distribution

Now let's extend the notion of convergence in distribution to random vectors. The definition is almost identical to the scalar case, with only the obvious modifications. Let $\{F_n\}_{n=1}^{\infty}$ be a sequence of cdfs on $\mathbb{R}^K$, and let F be a cdf on $\mathbb{R}^K$. We say that F_n converges to F **weakly** if, for any $\mathbf{s}$ such that F is continuous at $\mathbf{s}$, we have

$$F_n(\mathbf{s}) \to F(\mathbf{s}) \quad \text{as } n \to \infty$$

Let $\{\mathbf{x}_n\}_{n=1}^{\infty}$ and $\mathbf{x}$ be random vectors in $\mathbb{R}^K$, where $\mathbf{x}_n \sim F_n$ and $\mathbf{x} \sim F$. We say that $\mathbf{x}_n$ converges **in distribution** to $\mathbf{x}$ if F_n converges weakly to F. In symbols, this convergence is represented by $\mathbf{x}_n \xrightarrow{d} \mathbf{x}$.

As discussed above, convergence of $\mathbf{x}_n$ to $\mathbf{x}$ in probability simply requires that the elements of $\mathbf{x}_n$ converge in probability (in the scalar sense) to the corresponding elements of $\mathbf{x}$. For convergence in distribution this is not generally true:

$$x_k^n \xrightarrow{d} x_k \text{ for all } k \text{ does } \textit{not} \text{ imply } \mathbf{x}_n := \begin{pmatrix} x_1^n \\ \vdots \\ x_K^n \end{pmatrix} \xrightarrow{d} \begin{pmatrix} x_1 \\ \vdots \\ x_K \end{pmatrix} =: \mathbf{x}$$

Put differently, convergence of the marginals does not necessarily imply convergence of the joint distribution. (As you might have guessed, one setting where convergence of the marginals implies convergence of the joint is when the elements of the vectors are independent, and the joint is just the product of the marginals.)

The fact that elementwise convergence in distribution does not necessarily imply convergence of the vectors is problematic, because vector convergence is harder to work with than scalar convergence. Fortunately, we have the following results, which provide a link from scalar to vector convergence:

Fact 2.5.3. Let $\mathbf{x}_n$ and $\mathbf{x}$ be random vectors in $\mathbb{R}^K$.

1. If $\mathbf{a}'\mathbf{x}_n \xrightarrow{d} \mathbf{a}'\mathbf{x}$ for any $\mathbf{a} \in \mathbb{R}^K$, then $\mathbf{x}_n \xrightarrow{d} \mathbf{x}$.
2. If $\mathbf{a}'\mathbf{x}_n \xrightarrow{p} \mathbf{a}'\mathbf{x}$ for any $\mathbf{a} \in \mathbb{R}^K$, then $\mathbf{x}_n \xrightarrow{p} \mathbf{x}$.

The second of these results is quite straightforward to prove (exercise 8.5.2). The first is more difficult (the standard argument uses characteristic functions). It is often referred as the Cramer-Wold device.

As in the scalar case (fact 1.4.4), convergence in distribution is preserved under continuous transformations:

Fact 2.5.4 (Continuous mapping theorem)**.** Let $\mathbf{x}_n$ and $\mathbf{x}$ be random vectors in $\mathbb{R}^K$. If $g\colon \mathbb{R}^K \to \mathbb{R}^J$ is continuous and $\mathbf{x}_n \xrightarrow{d} \mathbf{x}$, then $g(\mathbf{x}_n) \xrightarrow{d} g(\mathbf{x})$.

Another result used routinely in econometric theory is the vector version of Slutsky's theorem:

Fact 2.5.5 (Slutsky's theorem)**.** Let $\mathbf{x}_n$ and $\mathbf{x}$ be random vectors in $\mathbb{R}^K$, let $\mathbf{Y}_n$ be random matrices, and let $\mathbf{C}$ be a constant matrix. If $\mathbf{Y}_n \xrightarrow{p} \mathbf{C}$ and $\mathbf{x}_n \xrightarrow{d} \mathbf{x}$, then

$$\mathbf{Y}_n\mathbf{x}_n \xrightarrow{d} \mathbf{C}\mathbf{x} \quad \text{and} \quad \mathbf{Y}_n + \mathbf{x}_n \xrightarrow{d} \mathbf{C} + \mathbf{x}$$

whenever the matrices are conformable.

2.5.3 Vector LLN and CLT

With the above definitions of convergence in hand, we can move on to the next topic: Vector LLN and CLT. The scalar LLN and CLT that we discussed in §1.4 extend to the vector case in a natural way. For example, we have the following result:

Theorem 2.5.1. *Let $\{\mathbf{x}_n\}$ be an* IID *sequence of random vectors in $\mathbb{R}^K$ with $\mathbb{E}\left[\|\mathbf{x}_n\|^2\right] < \infty$. Let $\boldsymbol{\mu} := \mathbb{E}\left[\mathbf{x}_n\right]$ and let $\Sigma := \operatorname{var}[\mathbf{x}_n]$. For this sequence we have*

$$\bar{\mathbf{x}}_N := \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n \xrightarrow{p} \boldsymbol{\mu} \quad \textit{and} \quad \sqrt{N}\left(\bar{\mathbf{x}}_N - \boldsymbol{\mu}\right) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \Sigma) \tag{2.8}$$

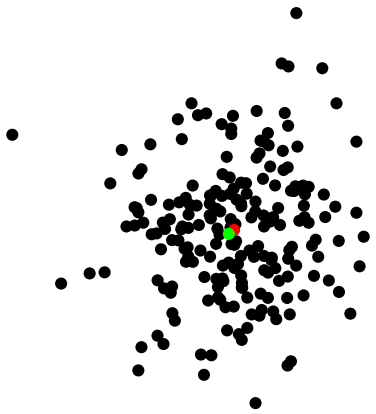


Figure 2.9: LLN, vector case

Figure 2.9 illustrates the LLN in two dimensions. The green dot is the point $\mathbf{0} = (0, 0)$ in $\mathbb{R}^2$. The black dots are IID observations $\mathbf{x}_1, \ldots, \mathbf{x}_N$ of a random vector with mean $\boldsymbol{\mu} = \mathbf{0}$. The red dot is the sample mean $\frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n$. (Remember that we are working with vectors here, so the summation and scalar multiplication in the sample mean $\bar{\mathbf{x}}_N$ is done elementwise—in this case for two elements. In particular, the sample mean is a linear combination of the observations $\mathbf{x}_1, \ldots, \mathbf{x}_N$.) By the vector LLN, the red dot converges to the green dot.

The vector LLN in theorem 2.5.1 follows from the scalar LLN. To see this, let $\mathbf{x}_n$ be as in theorem 2.5.1, let $\mathbf{a}$ be any constant vector in $\mathbb{R}^K$ and consider the scalar sequence $\{y_n\}$ defined by $y_n = \mathbf{a}'\mathbf{x}_n$. The sequence $\{y_n\}$ inherets the IID property from $\{\mathbf{x}_n\}$.[5] By the scalar LLN (theorem 1.4.1) we have

$$\frac{1}{N}\sum_{n=1}^{N} y_n \xrightarrow{p} \mathbb{E}[y_n] = \mathbb{E}[\mathbf{a}'\mathbf{x}_n] = \mathbf{a}'\mathbb{E}[\mathbf{x}_n] = \mathbf{a}'\boldsymbol{\mu}$$

But

$$\frac{1}{N}\sum_{n=1}^{N} y_n = \frac{1}{N}\sum_{n=1}^{N} \mathbf{a}'\mathbf{x}_n = \mathbf{a}'\left[\frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n\right] = \mathbf{a}'\bar{\mathbf{x}}_N$$

Since $\mathbf{a}$ was arbitrary, we have shown that

$$\mathbf{a}'\bar{\mathbf{x}}_N \xrightarrow{p} \mathbf{a}'\boldsymbol{\mu} \text{ for any } \mathbf{a} \in \mathbb{R}^K$$

The claim $\bar{\mathbf{x}}_N \xrightarrow{p} \boldsymbol{\mu}$ now follows from fact 2.5.3.

[5] Functions of independent random variables are themselves independent (fact 1.3.2, page 27).

The vector CLT in theorem 2.5.1 also follows from the scalar case. The proof is rather similar to the vector LLN proof we have just completed. See exercise 8.5.5.

2.6 Exercises

Ex. 2.6.1. Given two vectors $\mathbf{x}$ and $\mathbf{y}$, show that $|\|\mathbf{x}\| - \|\mathbf{y}\|| \leq \|\mathbf{x} - \mathbf{y}\|$.[6]

Ex. 2.6.2. Use the first property in fact 2.1.1 to show that if $\mathbf{y} \in \mathbb{R}^N$ is such that $\mathbf{y}'\mathbf{x} = 0$ for every $\mathbf{x} \in \mathbb{R}^N$, then $\mathbf{y} = \mathbf{0}$.

Ex. 2.6.3. Prove the second part of theorem 2.1.1. In particular, show that if $f : \mathbb{R}^K \to \mathbb{R}^N$ is linear, then there exists an $N \times K$ matrix $\mathbf{A}$ such that $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$ for all $\mathbf{x} \in \mathbb{R}^K$.[7]

Ex. 2.6.4. Show that if S and S' are two linear subspaces of $\mathbb{R}^N$, then $S \cap S'$ is also a linear subspace.

Ex. 2.6.5. Show that every linear subspace of $\mathbb{R}^N$ contains the origin $\mathbf{0}$.

Ex. 2.6.6. Show that the vectors $(1, 1)$ and $(-1, 2)$ are linearly independent.[8]

Ex. 2.6.7. Find two unit vectors (i.e., vectors with norm equal to one) that are orthogonal to $(1, -2)$.

Ex. 2.6.8. Let $\mathbf{a} \in \mathbb{R}^N$ and let $A := \{\mathbf{x} \in \mathbb{R}^N : \mathbf{a}'\mathbf{x} = 0\}$. Show that A is a linear subspace of $\mathbb{R}^N$.

Ex. 2.6.9. Let Q be the subset of $\mathbb{R}^3$ defined by

$$Q := \{(x_1, x_2, x_3) \in \mathbb{R}^3 : x_2 = x_1 + x_3\}$$

Is Q a linear subspace of $\mathbb{R}^3$? Why or why not?

Ex. 2.6.10. Let Q be the subset of $\mathbb{R}^3$ defined by

$$Q := \{(x_1, x_2, x_3) \in \mathbb{R}^3 : x_2 = 1\}$$

Is Q a linear subspace of $\mathbb{R}^3$? Why or why not?

[6] Hint: Use the triangle inequality.
[7] Hint: $\text{col}_k(\mathbf{A}) = f(\mathbf{e}_k)$.
[8] Hint: Look at the different definitions of linear independence. Choose the one that's easiest to work with in terms of algebra.

Ex. 2.6.11. Let $X := \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ be a linearly independent subset of $\mathbb{R}^N$. Is it possible that $\mathbf{0} \in X$? Why or why not?

Ex. 2.6.12. Prove facts 2.3.1 and 2.3.10.

Ex. 2.6.13. Show that for any two conformable matrices $\mathbf{A}$ and $\mathbf{B}$, we have $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$.[9]

Ex. 2.6.14. Let $\mathbf{A}$ be a constant $N \times N$ matrix. Assuming existence of the inverse $\mathbf{A}^{-1}$, show that $(\mathbf{A}')^{-1} = (\mathbf{A}^{-1})'$.

Ex. 2.6.15. Show that if $\mathbf{e}_i$ and $\mathbf{e}_j$ are the i-th and j-th canonical basis vectors of $\mathbb{R}^N$ respectively, and $\mathbf{A}$ is an $N \times N$ matrix, then $\mathbf{e}_i'\mathbf{A}\mathbf{e}_j = a_{ij}$, the i, j-th element of $\mathbf{A}$.

Ex. 2.6.16. Let

$$\mathbf{A} := \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

Show that $\mathbf{A}$ is nonnegative definite.

Ex. 2.6.17. Show that for any matrix $\mathbf{A}$, the matrix $\mathbf{A}'\mathbf{A}$ is well-defined (i.e., multiplication is possible), square, and nonnegative definite.

Ex. 2.6.18. Show that if $\mathbf{A}$ and $\mathbf{B}$ are positive definite and $\mathbf{A} + \mathbf{B}$ is well defined, then it is also positive definite.

Ex. 2.6.19. Let $\mathbf{A}$ be $N \times K$. Show that if $\mathbf{Ax} = \mathbf{0}$ for all $K \times 1$ vectors $\mathbf{x}$, then $\mathbf{A} = \mathbf{0}$ (i.e., every element of $\mathbf{A}$ is zero).

Ex. 2.6.20. Let $\mathbf{I}_N$ be the $N \times N$ identity matrix.

1. Explain briefly why $\mathbf{I}_N$ is full column rank.
2. Show that $\mathbf{I}_N$ is the inverse of itself.
3. Let $\mathbf{A} := \alpha\mathbf{I}_N$. Give a condition on α such that $\mathbf{A}$ is positive definite.

Ex. 2.6.21. Let $\mathbf{X} := \mathbf{I}_N - 2\mathbf{u}\mathbf{u}'$, where $\mathbf{u}$ is an $N \times 1$ vector with $\|\mathbf{u}\| = 1$. Show that $\mathbf{X}$ is symmetric and $\mathbf{XX} = \mathbf{I}_N$.

Ex. 2.6.22. Let $\mathbf{1}$ be an $N \times 1$ vector of ones. Consider the matrix

$$\mathbf{Z} := \frac{1}{N}\mathbf{1}\mathbf{1}'$$

[9] Hint: Look at the definition of the inverse! Always look at the definition, and then show that the object in question has the stated property.

1. Show that if $\mathbf{x}$ is any $N \times 1$ vector, then $\mathbf{Zx}$ is a vector with all elements equal to the sample mean of the elements of $\mathbf{x}$.

2. Show that $\mathbf{Z}$ is idempotent.

Ex. 2.6.23. Show that if $\mathbf{x}$ is a random vector with $\mathbb{E}[\mathbf{x}\mathbf{x}'] = \mathbf{I}$ and $\mathbf{A}$ is a conformable constant matrix, then $\mathbb{E}[\mathbf{x}'\mathbf{A}\mathbf{x}] = \text{trace}(\mathbf{A})$.

Ex. 2.6.24. Let $\mathbf{x}$ be random and let $\mathbf{a}$ be constant. Show that if $\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}$ and $\text{var}[\mathbf{x}] = \boldsymbol{\Sigma}$, then $\mathbb{E}[\mathbf{a}'\mathbf{x}] = \mathbf{a}'\boldsymbol{\mu}$ and $\text{var}[\mathbf{a}'\mathbf{x}] = \mathbf{a}'\boldsymbol{\Sigma}\mathbf{a}$.

Ex. 2.6.25. Let $\mathbf{x}$ be a random $K \times 1$ vector. Show that $\mathbb{E}[\mathbf{x}\mathbf{x}']$ is nonnegative definite.

Ex. 2.6.26. Let $\mathbf{x} = (x_1, \ldots, x_N) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_N)$.

1. Are x_1 and x_2 independent? Why or why not?

2. What is the distribution of x_1^2? Why?

3. What is the distribution of x_1^2/x_2^2? Why?

4. What is the distribution of $x_1[2/(x_2^2 + x_3^2)]^{1/2}$? Why?

5. What is the distribution of $\|\mathbf{x}\|^2$? Why?

6. If $\mathbf{a}$ is an $N \times 1$ constant vector, what is the distribution of $\mathbf{a}'\mathbf{x}$?

Ex. 2.6.27 (Computational)**.** Write a function in R called `innerprod` that computes the inner product of any two vectors using `sum`. Write another function called `norm`, also using `sum`, that computes the norm of any vector. Choose two vectors and a scalar, and check that the properties in fact 2.1.1 all hold.

Ex. 2.6.28 (Computational)**.** Form a 4×5 matrix $\mathbf{A}$, the elements of which are chosen randomly via uniform sampling from $\{1, 2, 3, 4\}$ with replacement (look up the online documentation on the `sample` function). Compute the row sums and column sums by either pre- or postmultiplying by a vector of ones. (Which is which?) Check against the built-in functions `colSums` and `rowSums`.

Ex. 2.6.29 (Computational)**.** Using the matrix $\mathbf{A}$ in exercise 2.6.28, create a new matrix $\mathbf{B}$ such that the j-th column of $\mathbf{B}$ is the j-th column of $\mathbf{A}$ minus the column mean of the j-th column of $\mathbf{A}$. The built-in function `colMeans` can be used to obtain the column means.

2.6.1 Solutions to Selected Exercises

Solution to Exercise 2.6.8. Let $\mathbf{x}, \mathbf{y} \in A$ and let $\alpha, \beta \in \mathbb{R}$. We must show that $\mathbf{z} := \alpha\mathbf{x} + \beta\mathbf{y} \in A$, or, equivalently, $\mathbf{a}'\mathbf{z} = \mathbf{a}'(\alpha\mathbf{x} + \beta\mathbf{y}) = 0$. This is immediate, because $\mathbf{a}'(\alpha\mathbf{x} + \beta\mathbf{y}) = \alpha\mathbf{a}'\mathbf{x} + \beta\mathbf{a}'\mathbf{y} = 0 + 0 = 0$. □

Solution to Exericse 2.6.9. If $\mathbf{a} := (1, -1, 1)$, then Q is all $\mathbf{x}$ with $\mathbf{a}'\mathbf{x} = 0$. This set is a linear subspace of $\mathbb{R}^3$, as shown in exercise 2.6.8. □

Solution to Exercise 2.6.11. If X has more than one element, then it is not possible. To see this, suppose (without any loss of generality) that $\mathbf{x}_1 = \mathbf{0} \in X$. Then

$$\mathbf{x}_1 = \mathbf{0} = \sum_{k=2}^{K} 0\mathbf{x}_k$$

In other words, $\mathbf{x}_1$ is a linear combination of the elements of X. This contradicts linear independence. □

Solution to Exercise 2.6.20. The solutions are as follows: (1) $\mathbf{I}_N$ is full column rank because its columns are the canonical basis vectors, which are independent. (2) By definition, $\mathbf{B}$ is the inverse of $\mathbf{A}$ if $\mathbf{BA} = \mathbf{AB} = \mathbf{I}_N$. It follows immediately that $\mathbf{I}_N$ is the inverse of itself. (3) A sufficient condition is $\alpha > 0$. If this holds, then given $\mathbf{x} \neq \mathbf{0}$, we have $\mathbf{x}'\alpha\mathbf{I}_N\mathbf{x} = \alpha\|\mathbf{x}\|^2 > 0$. □

Solution to Exercise 2.6.21. First, $\mathbf{X}$ is symmetric because

$$\mathbf{X}' = (\mathbf{I}_N - 2\mathbf{u}\mathbf{u}')' = \mathbf{I}_N' - 2(\mathbf{u}\mathbf{u}')' = \mathbf{I}_N - 2(\mathbf{u}')'\mathbf{u}' = \mathbf{I}_N - 2\mathbf{u}\mathbf{u}' = \mathbf{X}$$

Second, $\mathbf{XX} = \mathbf{I}_N$, because

$$\begin{aligned}\mathbf{XX} = (\mathbf{I}_N - 2\mathbf{u}\mathbf{u}')(\mathbf{I}_N' - 2\mathbf{u}\mathbf{u}') &= \mathbf{I}_N\mathbf{I}_N - 2\mathbf{I}_N 2\mathbf{u}\mathbf{u}' + (2\mathbf{u}\mathbf{u}')(2\mathbf{u}\mathbf{u}') \\ &= \mathbf{I}_N - 4\mathbf{u}\mathbf{u}' + 4\mathbf{u}\mathbf{u}'\mathbf{u}\mathbf{u}' = \mathbf{I}_N - 4\mathbf{u}\mathbf{u}' + 4\mathbf{u}\mathbf{u}' = \mathbf{I}_N\end{aligned}$$

The second last equality is due to the assumption that $\mathbf{u}'\mathbf{u} = \|\mathbf{u}\|^2 = 1$. □

Solution to Exercise 2.6.26. First note that since $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_N)$ we have $\operatorname{cov}[x_i, x_j] = 0$ for all $i \neq j$. Since uncorrelated normal random variables are independent, we

then have $x_1, \ldots, x_N \overset{\text{IID}}{\sim} \mathcal{N}(0,1)$. Since sums of squares of independent standard normals are chi-squared, we have in particular that

$$\sum_{n=1}^{k} x_n^2 \sim \chi^2(k) \tag{2.9}$$

for any $k \leq N$. The solutions to the exercise can now be given:

1. Yes, for the reason just described.
2. $x_1^2 \sim \chi^2(1)$ by (2.9)
3. $x_1^2/x_2^2 \sim F(1,1)$, because if $Q_1 \sim \chi^2(k_1)$ and $Q_2 \sim \chi^2(k_2)$ and Q_1 and Q_2 and independent, then $(Q_1/k_1)/(Q_2/k_2) \sim F(k_1, k_2)$.
4. $x_1[2/(x_2^2 + x_3^2)]^{1/2} \sim t(2)$, because if $Z \sim \mathcal{N}(0,1)$ and $Q \sim \chi^2(k)$ and Z and Q are independent, then $Z(k/Q)^{1/2} \sim t(k)$.
5. $\|\mathbf{x}\|^2 = \sum_{n=1}^{N} x_N^2 \sim \chi^2(N)$ by (2.9).
6. Linear combinations of normals are normal, so $y := \mathbf{a}'\mathbf{x}$ is normal. Evidently $\mathbb{E}[y] = \mathbb{E}[\mathbf{a}'\mathbf{x}] = \mathbf{a}'\mathbb{E}[\mathbf{x}] = 0$. Using independence, we obtain

$$\operatorname{var}[y] = \sum_{n=1}^{N} a_n^2 \operatorname{var}[x_n] = \sum_{n=1}^{N} a_n^2$$

Hence $y \sim \mathcal{N}(0, \sum_{n=1}^{N} a_n^2)$.

□

Chapter 3

Projections

This chapter provides further background in linear algebra for studying OLS with multiple regressors. At the heart of the chapter is the orthogonal projection theorem, which lies behind many of the key results in OLS theory. The theory of projections also allows us to define conditional expectations, and determine the properties of the conditioning operation.

3.1 Orthogonality and Projection

[roadmap]

3.1.1 Orthogonality

Let $\mathbf{x}$ and $\mathbf{z}$ be two vectors in $\mathbb{R}^N$. If $\mathbf{x}'\mathbf{z} = 0$, then $\mathbf{x}$ and $\mathbf{z}$ are said to be **orthogonal**, and we write $\mathbf{x} \perp \mathbf{z}$. In $\mathbb{R}^2$, $\mathbf{x}$ and $\mathbf{z}$ are orthogonal when they are perpendicular to one another (figure 3.1). If $\mathbf{x}$ is a vector and S is a set, then we say that **$\mathbf{x}$ is orthogonal to S** if $\mathbf{x} \perp \mathbf{z}$ for all $\mathbf{z} \in S$. In this case we write $\mathbf{x} \perp S$. Figure 3.2 illustrates.

The first thing you need to know about orthogonal vectors is the Pythagorean Law:

Theorem 3.1.1. *If $\mathbf{x}_1, \ldots, \mathbf{x}_K$ are vectors in $\mathbb{R}^N$ and $\mathbf{x}_i \perp \mathbf{x}_j$ whenever $i \neq j$, then*

$$\|\mathbf{x}_1 + \cdots + \mathbf{x}_K\|^2 = \|\mathbf{x}_1\|^2 + \cdots + \|\mathbf{x}_K\|^2$$

Orthogonality and linear independence are related. For example,

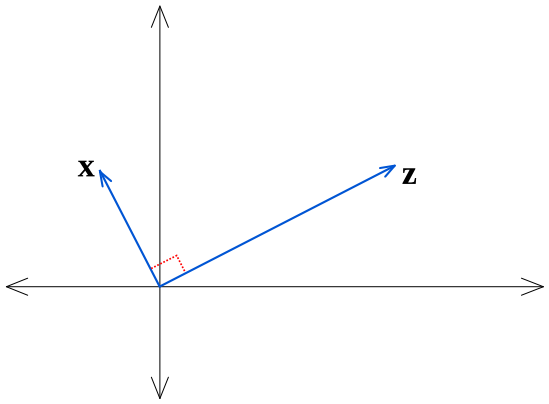


Figure 3.1: $\mathbf{x} \perp \mathbf{z}$

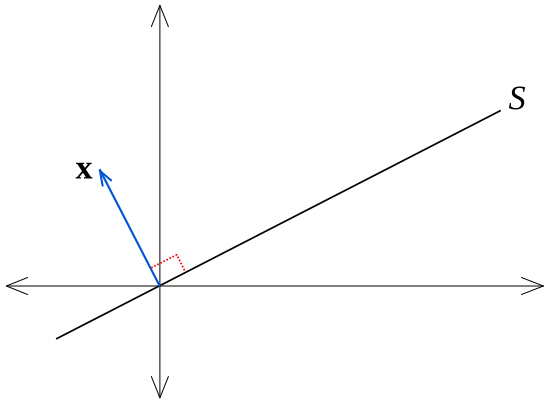


Figure 3.2: $\mathbf{x} \perp S$

Fact 3.1.1. If V is a finite set with $\mathbf{x} \perp \mathbf{y}$ for all distinct pairs $\mathbf{x}, \mathbf{y} \in V$, and, moreover, $\mathbf{0} \notin V$, then V is linearly independent.

3.1.2 Projections

One problem that comes up in many different contexts is approximation of an element $\mathbf{y}$ of $\mathbb{R}^N$ by an element of a given subspace S of $\mathbb{R}^N$. Stated more precisely, the problem is, given $\mathbf{y}$ and S, to find the closest element $\hat{\mathbf{y}}$ of S to $\mathbf{y}$. Closeness is in terms of euclidean norm, so $\hat{\mathbf{y}}$ is the minimizer of $\|\mathbf{y} - \mathbf{z}\|$ over all $\mathbf{z} \in S$:

$$\hat{\mathbf{y}} := \operatorname*{argmin}_{\mathbf{z} \in S} \|\mathbf{y} - \mathbf{z}\|$$

Existence of a minimizer is not immediately obvious, suggesting that $\hat{\mathbf{y}}$ may not be well-defined. However, it turns out that we need not be concerned, as $\hat{\mathbf{y}}$ always exists (given any S and $\mathbf{y}$). The next theorem states this fact, as well as providing a way to identify $\hat{\mathbf{y}}$.

Theorem 3.1.2 (Orthogonal Projection Theorem, Part 1)**.** *Let $\mathbf{y} \in \mathbb{R}^N$ and let S be a subspace of $\mathbb{R}^N$. The closest point in S to $\mathbf{y}$ is the unique vector $\hat{\mathbf{y}} \in S$ such that $\mathbf{y} - \hat{\mathbf{y}} \perp S$.*

The vector $\hat{\mathbf{y}}$ in theorem 3.1.2 is called the **orthogonal projection of y onto S**. Although we do not prove the theorem here, the intuition is easy to grasp from a graphical presentation. Figure 3.3 illustrates. Looking at the figure, we can see that the closest point $\hat{\mathbf{y}}$ to $\mathbf{y}$ within S is indeed the one and only point in S such that $\mathbf{y} - \hat{\mathbf{y}}$ is orthogonal to S.

Holding S fixed, we can think of the operation

$$\mathbf{y} \;\mapsto\; \text{the orthogonal projection of } \mathbf{y} \text{ onto } S$$

as a *function* from $\mathbb{R}^N$ to $\mathbb{R}^N$.[1] The function is typically denoted by $\mathbf{P}$, so that $\mathbf{P}(\mathbf{y})$ or $\mathbf{P}\mathbf{y}$ represents the orthogonal projection $\hat{\mathbf{y}}$. In general, $\mathbf{P}$ is called the **orthogonal projection onto S**. Figure 3.4 illustrates the action of $\mathbf{P}$ on two different vectors.

Using this notation, we can restate the orthogonal projection theorem, as well as adding some properties of $\mathbf{P}$:

Theorem 3.1.3 (Orthogonal Projection Theorem 2)**.** *Let S be any linear subspace, and let $\mathbf{P} \colon \mathbb{R}^N \to \mathbb{R}^N$ be the orthogonal projection onto S. The function $\mathbf{P}$ is linear. Moreover, for any $\mathbf{y} \in \mathbb{R}^N$, we have*

[1] Confirm in your mind that we are describing a functional relationship, as defined in §13.1.1.

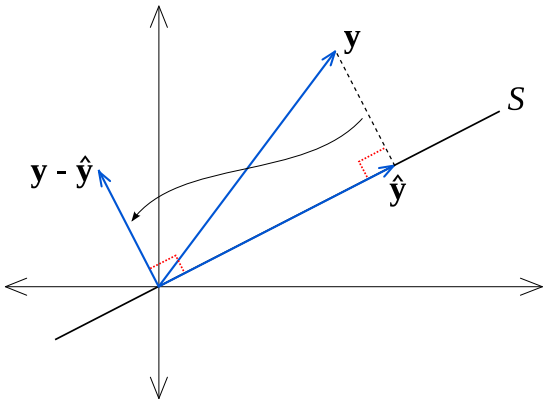


Figure 3.3: Orthogonal projection

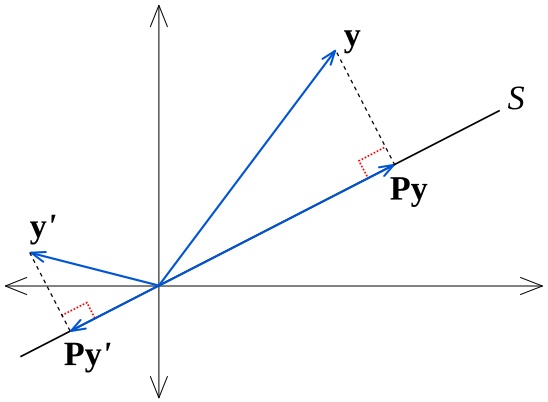


Figure 3.4: Orthogonal projection under **P**

1. $\mathbf{Py} \in S$,
2. $\mathbf{y} - \mathbf{Py} \perp S$,
3. $\|\mathbf{y}\|^2 = \|\mathbf{Py}\|^2 + \|\mathbf{y} - \mathbf{Py}\|^2$,
4. $\|\mathbf{Py}\| \leq \|\mathbf{y}\|$, *and*
5. $\mathbf{Py} = \mathbf{y}$ *if and only if* $\mathbf{y} \in S$.

These results are not difficult to prove, given theorem 3.1.2. Linearity of $\mathbf{P}$ is left as an exercise (exercise 3.4.6). Parts 1 and 2 follow directly from theorem 3.1.2. To see part 3, observe that $\mathbf{y}$ can be decomposed as

$$\mathbf{y} = \mathbf{Py} + \mathbf{y} - \mathbf{Py}$$

Part 3 now follows from parts 1–2 and the Pythagorean law. (Why?) Part 4 follows from part 3. (Why?) Part 5 is obvious from the definition of $\mathbf{Py}$ as the closest point to $\mathbf{y}$ in S.

There's one more very important property of $\mathbf{P}$ that we need to make note of: Suppose we have two linear subspaces S_1 and S_2 of $\mathbb{R}^N$, where $S_1 \subset S_2$. What then is the difference between (a) first projecting a point onto the bigger subspace S_2, and then projecting the result onto the smaller subspace S_1, and (b) projecting directly to the smaller subspace S_1? The answer is none—we get the same result.

Fact 3.1.2. Let S_1 and S_2 be two subspaces of $\mathbb{R}^N$, and let $\mathbf{y} \in \mathbb{R}^N$. Let $\mathbf{P}_1$ and $\mathbf{P}_2$ be the projections onto S_1 and S_2 respectively. If $S_1 \subset S_2$, then

$$\mathbf{P}_1\mathbf{P}_2\mathbf{y} = \mathbf{P}_2\mathbf{P}_1\mathbf{y} = \mathbf{P}_1\mathbf{y}$$

There's yet another way of stating the orthogonal projection theorem, which is also informative. Given $S \subset \mathbb{R}^N$, the **orthogonal complement** of S is defined as

$$S^\perp := \{\mathbf{x} \in \mathbb{R}^N : \mathbf{x} \perp S\}$$

In other words, $S^\perp$ is the set of all vectors that are orthogonal to S. Figure 3.5 gives an example in $\mathbb{R}^2$.

Fact 3.1.3. Given any S, the orthogonal complement $S^\perp$ is always a linear subspace.

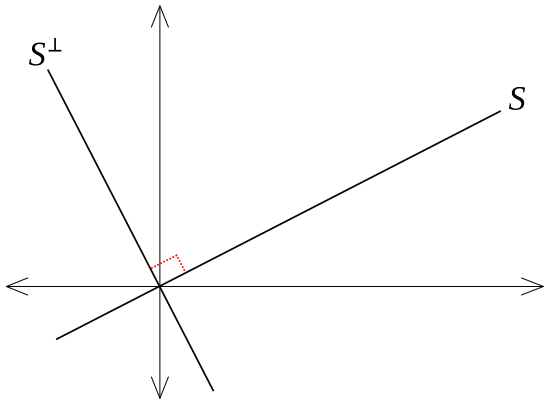


Figure 3.5: Orthogonal complement of S

This is easy enough to confirm: Looking back at the definition of linear subspaces, we see that the following statement must be verified: Given $\mathbf{x}, \mathbf{y} \in S^\perp$ and $\alpha, \beta \in \mathbb{R}$, the vector that $\alpha\mathbf{x} + \beta\mathbf{y}$ is also in $S^\perp$. Clearly this is the case, because if $\mathbf{z} \in S$, then

$$\begin{aligned}(\alpha\mathbf{x} + \beta\mathbf{y})'\mathbf{z} &= \alpha\mathbf{x}'\mathbf{z} + \beta\mathbf{y}'\mathbf{z} \qquad (\because \text{ linearity of inner products}) \\ &= \alpha \times 0 + \beta \times 0 = 0 \qquad (\because \ \mathbf{x}, \mathbf{y} \in S^\perp \text{ and } \mathbf{z} \in S)\end{aligned}$$

We have shown that $\alpha\mathbf{x} + \beta\mathbf{y} \perp \mathbf{z}$ for any $\mathbf{z} \in S$, thus confirming that $\alpha\mathbf{x} + \beta\mathbf{y} \in S^\perp$.

Fact 3.1.4. For $S \subset \mathbb{R}^N$, we have $S \cap S^\perp = \{\mathbf{0}\}$.

Now, let's look at the orthogonal projection theorem again. Our interest was in projecting $\mathbf{y}$ onto S. However, we have just learned that $S^\perp$ is itself a linear subspace, so we can also project $\mathbf{y}$ onto $S^\perp$. Just as we used $\mathbf{P}$ to denote the function sending $\mathbf{y}$ into its projection onto S, so we'll use $\mathbf{M}$ to denote the function sending $\mathbf{y}$ into its projection onto $S^\perp$. The result we'll denote by $\hat{\mathbf{u}}$, so that $\hat{\mathbf{u}} := \mathbf{M}\mathbf{y}$. Figure 3.6 illustrates. The figure suggests that we will have $\mathbf{y} = \hat{\mathbf{y}} + \hat{\mathbf{u}}$, and indeed that is the case. The next theorem states this somewhat more mathematically.

Theorem 3.1.4 (Orthogonal Projection Theorem 3)**.** *Let S be a linear subspace of $\mathbb{R}^N$. If $\mathbf{P}$ is the orthogonal projection onto S and $\mathbf{M}$ is the orthogonal projection onto $S^\perp$, then $\mathbf{P}\mathbf{y}$ and $\mathbf{M}\mathbf{y}$ are orthogonal, and*

$$\mathbf{y} = \mathbf{P}\mathbf{y} + \mathbf{M}\mathbf{y}$$

If S_1 and S_2 are two subspaces of $\mathbb{R}^N$ with $S_1 \subset S_2$, then $S_2^\perp \subset S_1^\perp$. This means that the result in fact 3.1.2 is reversed for $\mathbf{M}$.

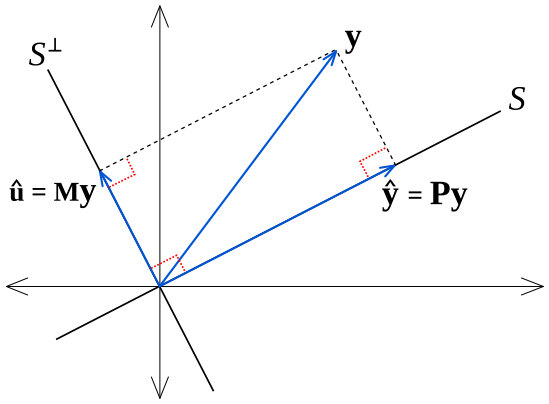


Figure 3.6: Orthogonal projection

Fact 3.1.5. Let S_1 and S_2 be two subspaces of $\mathbb{R}^N$ and let $\mathbf{y} \in \mathbb{R}^N$. Let $\mathbf{M}_1$ and $\mathbf{M}_2$ be the projections onto $S_1^\perp$ and $S_2^\perp$ respectively. If $S_1 \subset S_2$, then,

$$\mathbf{M}_1\mathbf{M}_2\mathbf{y} = \mathbf{M}_2\mathbf{M}_1\mathbf{y} = \mathbf{M}_2\mathbf{y}$$

Fact 3.1.6. $\mathbf{Py} = \mathbf{0}$ if and only if $\mathbf{y} \in S^\perp$, and $\mathbf{My} = \mathbf{0}$ if and only if $\mathbf{y} \in S$.[2]

3.2 Overdetermined Systems of Equations

When we get to multivariate linear regression, we will see that, mathematically speaking, the problem we are presented with is one of solving what is called an *overdetermined* system of equations. In turn, overdetermined systems of equations are usually solved using orthogonal projection. Let's have a quick look at how this is done. We begin with a system of equations such as $\mathbf{X}\boldsymbol{\beta} = \mathbf{y}$, where $\mathbf{X}$ is $N \times K$, $\boldsymbol{\beta}$ is $K \times 1$, and $\mathbf{y}$ is $N \times 1$. We regard the matrix $\mathbf{X}$ and the vector $\mathbf{y}$ as given, and seek a $\boldsymbol{\beta} \in \mathbb{R}^K$ that solves this equation. Throughout this section, we maintain the assumption that $\mathbf{X}$ is full column rank.

If $K = N$, then the full column rank assumption and fact 2.3.3 imply that this system has precisely one solution. However, we are going to study the case when $N > K$. In this case, the system of equations is said to be **overdetermined**. This corresponds to

[2]For example, if $\mathbf{Py} = \mathbf{0}$, then $\mathbf{y} = \mathbf{Py} + \mathbf{My} = \mathbf{My}$. Hence $\mathbf{M}$ does not shift $\mathbf{y}$. If an orthogonal projection onto a subspace doesn't shift a point, that's because the point is already in that subspace (see, e.g., theorem 3.1.3). In this case the subspace is $S^\perp$, and we conclude that $\mathbf{y} \in S^\perp$.

the situation where the number of equations (equal to N) is larger than the number of unknowns (the K elements of $\boldsymbol{\beta}$). Intuitively, in such a situation, we may not be able find a $\boldsymbol{\beta}$ that satisfies all N equations.

To understand this problem, recall from §2.3.1 that $\mathbf{X}$ can be viewed as a mapping from $\mathbb{R}^K$ to $\mathbb{R}^N$, and its range is the linear subspace of $\mathbb{R}^N$ spanned by the columns of $\mathbf{X}$:

$$\text{rng}(\mathbf{X}) := \{\text{all vectors } \mathbf{X}\boldsymbol{\beta} \text{ with } \boldsymbol{\beta} \in \mathbb{R}^K\} :=: \text{column space of } \mathbf{X}$$

As discussed in §2.3.1, a solution to $\mathbf{X}\boldsymbol{\beta} = \mathbf{y}$ exists precisely when $\mathbf{y}$ lies in $\text{rng}(\mathbf{X})$. In general, given our assumption that $K < N$, this outcome is unlikely.[3] As a result, the standard approach is to admit that an exact solution may not exist, and instead focus on finding a $\boldsymbol{\beta} \in \mathbb{R}^K$ such that $\mathbf{X}\boldsymbol{\beta}$ is as close to $\mathbf{y}$ as possible. Closeness is defined in the euclidean sense, so the problem is to minimize $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|$ over the set of all $\boldsymbol{\beta} \in \mathbb{R}^K$. Using the orthogonal projection theorem, the minimizer is easy to identify:

Theorem 3.2.1. *The minimizer of* $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|$ *over all* $\boldsymbol{\beta} \in \mathbb{R}^K$ *is* $\hat{\boldsymbol{\beta}} := (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$.

Proof. If we can show that $\mathbf{X}\hat{\boldsymbol{\beta}}$ is the closest point in $\text{rng}(\mathbf{X})$ to $\mathbf{y}$, we then have

$$\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\| \leq \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\| \text{ for any } \boldsymbol{\beta} \in \mathbb{R}^K$$

which is all we need to prove. To verify that $\hat{\mathbf{y}} := \mathbf{X}\hat{\boldsymbol{\beta}}$ is in fact the closest point in $\text{rng}(\mathbf{X})$ to $\mathbf{y}$, recall the orthogonal projection theorem (page 86). By this theorem, $\hat{\mathbf{y}} := \mathbf{X}\hat{\boldsymbol{\beta}}$ is the closest point in $\text{rng}(\mathbf{X})$ to $\mathbf{y}$ when

1. $\hat{\mathbf{y}} \in \text{rng}(\mathbf{X})$, and
2. $\mathbf{y} - \hat{\mathbf{y}} \perp \text{rng}(\mathbf{X})$

Here 1 is true by construction, and 2 translates to claim

$$\mathbf{y} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \perp \mathbf{X}\boldsymbol{\beta} \qquad \text{for all } \boldsymbol{\beta} \in \mathbb{R}^K$$

[3] Why is it unlikely that $\mathbf{y}$ lies in the range of $\mathbf{X}$? Since $\mathbf{X}$ is assumed to be full column rank, the range of $\mathbf{X}$ is a K-dimensional subspace of $\mathbb{R}^N$, while $\mathbf{y}$ is any point in $\mathbb{R}^N$. In a sense, for $K < N$, all K-dimensional subspaces of $\mathbb{R}^N$ are "small," and the "chance" of $\mathbf{y}$ happening to lie in this subspace is likewise small. For example, consider the case where $N = 3$ and $K = 2$. Then the column space of $\mathbf{X}$ forms a 2 dimensional plane in $\mathbb{R}^3$. Intuitively, this set has no volume because planes have no "thickness," and hence the chance of a randomly chosen $\mathbf{y}$ lying in this plane is near zero. More formally, if $\mathbf{y}$ is drawn from a continuous distribution over $\mathbb{R}^3$, then the probability that it falls in this plane is zero, due to the fact that planes in $\mathbb{R}^3$ are always of "Lebesgue measure zero."

This is true, because if $\boldsymbol{\beta} \in \mathbb{R}^K$, then

$$(\mathbf{X}\boldsymbol{\beta})'[\mathbf{y} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}] = \boldsymbol{\beta}'[\mathbf{X}'\mathbf{y} - \mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}] = \boldsymbol{\beta}'[\mathbf{X}'\mathbf{y} - \mathbf{X}'\mathbf{y}] = 0$$

The proof of theorem 3.2.1 is done. □

Notice that theorem 3.2.1 implicitly assumes that $\mathbf{X}'\mathbf{X}$ is invertible. This is justified, however, because $\mathbf{X}$ is assumed to be full column rank. (Exercise 3.4.9.)

Remark 3.2.1. On an intuitive level, why do we need full column rank for $\mathbf{X}$? Full rank means that the columns of $\mathbf{X}$ are linearly independent. Let's drop this assumption and consider what happens. The set $\text{rng}(\mathbf{X})$ is still a linear subspace, and the orthogonal projection theorem still gives us a closest point $\hat{\mathbf{y}}$ to $\mathbf{y}$ in $\text{rng}(\mathbf{X})$. Since $\hat{\mathbf{y}} \in \text{rng}(\mathbf{X})$, there still exists a vector $\hat{\boldsymbol{\beta}}$ such that $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$. The problems is that now there may exist two such vectors—or even an infinity.

Let's tie this discussion in to theorem 3.1.4 on page 89. We define the **projection matrix** $\mathbf{P}$ associated with $\mathbf{X}$ as

$$\mathbf{P} := \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \tag{3.1}$$

We also define the **annihilator** $\mathbf{M}$ associated with $\mathbf{X}$ as

$$\mathbf{M} := \mathbf{I} - \mathbf{P} \tag{3.2}$$

where $\mathbf{I}$ is, as usual, the identity matrix (in this case $N \times N$). Given these definitions, we then have

$$\mathbf{P}\mathbf{y} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}} = \hat{\mathbf{y}}$$

and

$$\mathbf{M}\mathbf{y} = (\mathbf{I} - \mathbf{P})\mathbf{y} = \mathbf{y} - \mathbf{P}\mathbf{y} = \mathbf{y} - \hat{\mathbf{y}}$$

The projection matrix and the annihilator correspond to the two projections $\mathbf{P}$ and $\mathbf{M}$ in theorem 3.1.4. $\mathbf{P}$ projects onto $\text{rng}(\mathbf{X})$, while $\mathbf{M}$ projects onto the orthogonal complement of $\text{rng}(\mathbf{X})$. In particular, to find the closest element of $\text{rng}(\mathbf{X})$ to a given vector $\mathbf{y}$ in $\mathbb{R}^N$, we can just premultiply $\mathbf{y}$ by $\mathbf{P} := \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$.

Fact 3.2.1. Both $\mathbf{P}$ and $\mathbf{M}$ are symmetric and idempotent.

The proof is an exercise (exercise 3.4.10). Idempotence is rather intuitive here, because both $\mathbf{P}$ and $\mathbf{M}$ represent orthogonal projections onto linear subspaces. Such projections map vectors into their respective subspaces. Applying the mapping a second time has no effect, because the vector is already in the subspace.

Fact 3.2.2. The annihilator $\mathbf{M}$ associated with $\mathbf{X}$ satisfies $\mathbf{MX} = \mathbf{0}$.

The proof is an exercise. (Exercise 3.4.11.) The intuition is as follows: The j-th column of $\mathbf{MX}$ is $\mathbf{M}\mathbf{x}_j$, where $\mathbf{x}_j$ is the j-th column of $\mathbf{X}$. Since $\mathbf{x}_j$ is in $\text{rng}(\mathbf{X})$, it gets mapped into the zero vector by $\mathbf{M}$. This follows from fact 3.1.6 on page 90, but it's also quite intuitive in light of figure 3.6 (where S corresponds to $\text{rng}(\mathbf{X})$).

3.3 Conditioning

The main purpose of this section is to introduce conditional expectations and study their properties. The definition of conditional expectations given in elementary probability texts is often cumbersome to work with, and fails to provide the big picture. In advanced texts, there are several different approaches to presenting conditional expectations. The one I present here is less common than the plain vanila treatment, but it is, to my mind, by far the most intuitive. As you might expect given the location of this discussion, the presentation involves orthogonal projection.

3.3.1 The Space L_2

Suppose we want to predict the value of a random variable u using another variable v. In this case we'd want u and v to be similar to each other in some sense. Since it helps to think geometrically, we usually talk about "closeness" instead of similarity, but the meaning is the same. A natural measure of closeness is mean squared error (MSE). The mean squared error of v as a predictor of u is defined as $\mathbb{E}\left[(u-v)^2\right]$. For the purposes of this section, it will be more convenient if we make a slight adjustment, replacing the mean squared error with the root mean squared error (RMSE), which is, as the name suggests, the square root of the MSE. Since we'll be using it a lot, let's give the RMSE its own notation:

$$|||u - v||| := \sqrt{\mathbb{E}\left[(u-v)^2\right]}$$

More generally, if we define

$$|||z||| := \sqrt{\mathbb{E}\left[z^2\right]} \tag{3.3}$$

and regard this as the "norm" of the random variable z, then the RMSE between u and v is the "norm" of the random variable $u - v$.

In fact the random variable "norm" $|||\cdot|||$ defined in (3.3) behaves very much like the euclidean norm $\|\cdot\|$ over vectors defined in (2.1) on page 50. If $\mathbf{z}$ is a vector in $\mathbb{R}^N$ and z is a random variable with density f, then the definitions of the two norms written side by side look pretty similar:

$$\|\mathbf{z}\| = \left(\sum_{n=1}^{N} z_n^2 \right)^{1/2} \quad \text{and} \quad |||z||| = \left(\int s^2 f(s) ds \right)^{1/2}$$

More importantly, all the properties of the euclidean norm $\|\cdot\|$ given in fact 2.1.1 (page 52) carry over to then "norm" $|||\cdot|||$ if we replace vectors with random variables. So let's stop calling $|||\cdot|||$ a "norm," and just start calling it a norm.[4]

Unlike the situation with the euclidean norm, there is a risk here that $|||z|||$ may not be defined because $\mathbb{E}[z^2] = \infty$. So for the purposes of this section, let's restrict attention to random variables with finite second moment. The standard name of this set of random variables is L_2. That is,

$$L_2 := \{ \text{ all random variables } x \text{ with } \mathbb{E}[x^2] < \infty \}$$

We can draw another parallel with the euclidean norm. As we saw in §2.1.1, the euclidean norm is defined in terms of the inner product on $\mathbb{R}^N$. If $\mathbf{x}$ and $\mathbf{y}$ are two vectors in $\mathbb{R}^N$, then the inner product is $\mathbf{x}'\mathbf{y}$, and the norm of vector $\mathbf{x}$ is $\|\mathbf{x}\| = \sqrt{\mathbf{x}'\mathbf{x}}$. Similarly, for random variables x and y, we define

$$\text{inner product of } x \text{ and } y \ := \ \mathbb{E}[xy]$$

As for the euclidean case, you can see here that the norm $|||x|||$ of x is precisely the square root of the inner product of x with itself.

As in the euclidean case, if the inner product of x and y is zero, then we say that x and y are **orthogonal**, and write $x \perp y$. This terminology is used frequently in econometrics (often by people who aren't actually sure why the term "orthogonal" is used—which puts you one step ahead of them). Clearly, if either x or y is zero mean, then orthogonality of x and y is equivalent to $\text{cov}[x, y] = 0$.

[4] One caveat is that while $\|\mathbf{x}\| = 0$ implies that $\mathbf{x} = \mathbf{0}$, it is not true that $|||z||| = 0$ implies z is the zero random variable (i.e., $z(\omega) = 0$ for all $\omega \in \Omega$). However, we can say that if $|||z||| = 0$, then the set $E := \{\omega \in \Omega : |z(\omega)| > 0\}$ satisfies $\mathbb{P}(E) = 0$. In this sense, z differs from the zero random variable only in a trivial way.

3.3.2 Measurability

What's the main point of the discussion in the previous section? By providing the collection of random variables L_2 with a norm, we've made it look rather similar to euclidean vector space $\mathbb{R}^N$. The advantage of this is that we have a lot of geometric intuition about the vector space $\mathbb{R}^N$. Since L_2 with its norm $|||\cdot|||$ behaves a lot like $\mathbb{R}^N$ with its norm $\|\cdot\|$, that same geometric intuition concerning vectors can be applied to the study of random variables. For example, we will see that the orthogonal projection theorem carries over to L_2, and this is precisely how we will study conditional expectation.

Recall that, in the case of $\mathbb{R}^N$, orthogonal projection starts with a linear subspace S of $\mathbb{R}^N$. Once we have this subspace, we think about how to project onto it. In fact S is the crucial component here, because once we select S, we implictly define the orthogonal projection mapping $\mathbf{P}$ that projects onto S (see theorems 3.1.2 and 3.1.3). So when I tell you that conditional expectation is characterized by orthogonal projection, you will understand that the first thing we need to think about is the linear subspaces that we want to project onto. It is to this topic that we now turn.

The first step is a definition at the very heart of probability theory: measurability. Let $x_1, \ldots, x_p$ be some collection of random variables, and let $\mathcal{G} := \{x_1, \ldots, x_p\}$. Thus, $\mathcal{G}$ is a set of random variables, often referred to in what follows as the **information set**. We will say that another random variable z is **$\mathcal{G}$-measurable** if there exists a (nonrandom) function $g\colon \mathbb{R}^p \to \mathbb{R}$ such that

$$z = g(x_1, \ldots, x_p)$$

Informally, what this means is that once the values of the random variables $x_1, \ldots, x_p$ have been realized, the variable z is completely determined (i.e., no longer random) and it's realized value can be calculated (assuming that we can calculate the functional form g). You might like to imagine it like this: Uncertainty is realized, in the sense that some ω is selected from the sample space Ω. Suppose that we don't get to view ω itself, but we do get to view certain random outcomes. For example, we might get to observe the realized values $x_1(\omega), \ldots, x_p(\omega)$. If z is $\mathcal{G}$-measurable, we can now calculate the realized value $z(\omega)$ of z, even without knowning ω, because we can compute $z(\omega) = g(x_1(\omega), \ldots, x_p(\omega))$.[5]

[5] A technical note: In the definition of measurability above, where we speak of existence of the function g, it is additional required that the function g is "Borel measurable." For the purposes of this course, we can regard non-Borel measurable functions as a mere theoretical curiosity. As such, the distinction will be ignored. See any text on measure theory for further details.

As a matter of notation, if $\mathcal{G} = \{x\}$ and y is $\mathcal{G}$-measurable, then we will also say that y is x-measurable.

Example 3.3.1. Let x and z be two random variables. If $z = 2x + 3$, then z is x-measurable. To see this formally, we can write $z = g(x)$ when $g(x) = 2x + 3$. Less formally, when x is realized, the value of z can be calculated.

Example 3.3.2. Let $x_1, \ldots, x_N$ be random variables and let $\bar{x}_N$ be their sample mean. If $\mathcal{G} = \{x_1, \ldots, x_N\}$, then $\bar{x}_N := N^{-1} \sum_{n=1}^{N} x_n$ is clearly $\mathcal{G}$-measurable.

Example 3.3.3. If x and y are independent, then y is not x-measurable. Indeed, if y was x-measurable, then we would have $y = g(x)$ for some function g. This contradicts independence of x and y.

Example 3.3.4. Let x, y and z be three random variables with $z = x + y$. Suppose that x and y are independent. Then z is not x-measurable. Intuitively, even if we know the realized value of x, the realization of z cannot be computed until we know the realized value of y. Formally, if z is x-measurable then $z = g(x)$ for some function g. But then $y = g(x) - x$, so y is x-measurable. This contradicts independence of x and y.

Example 3.3.5. Let $y = \alpha$, where α is a constant. This degenerate random variable is $\mathcal{G}$-measurable for any information set $\mathcal{G}$, because y is already deterministic. For example, if $\mathcal{G} = \{x_1, \ldots, x_p\}$, then we can take $y = g(x_1, \ldots, x_p) = \alpha + \sum_{i=1}^{p} 0 x_i$.

If x and y are known given the information in $\mathcal{G}$, then a third random variable that depends on only on x and y is likewise known given $\mathcal{G}$. Hence $\mathcal{G}$-measurability is preserved under the taking of sums, products, etc. In particular,

Fact 3.3.1. Let α, β be any scalars, and let x and y be random variables. If x and y are both $\mathcal{G}$-measurable, then $u := xy$ and $v := \alpha x + \beta y$ are also $\mathcal{G}$-measurable.

Let $\mathcal{G}$ and $\mathcal{H}$ be two information sets with $\mathcal{G} \subset \mathcal{H}$. In this case, if random variable z is $\mathcal{G}$ measurable, *then it is also $\mathcal{H}$-measurable*. This follows from our intuitive definition of measurability: If the value z is known once the variables in $\mathcal{G}$ are known, then it is certainly known when the extra information provided by $\mathcal{H}$ is available. The next example helps to clarify.

Example 3.3.6. Let x, y and z be three random variables, let $\mathcal{G} = \{x\}$, and let $\mathcal{H} = \{x, y\}$. Suppose that $z = 2x + 3$, so that z is $\mathcal{G}$-measurable. Then z is also $\mathcal{H}$-measurable. Informally, we can see that z is deterministic once the variables in $\mathcal{H}$ are realized. Formally, we can write $z = g(x, y)$, where $g(x, y) = 2x + 3 + 0y$. Hence z is also $\mathcal{H}$-measurable as claimed.

Let's note this idea as a fact:

Fact 3.3.2. If $\mathcal{G} \subset \mathcal{H}$ and z is $\mathcal{G}$-measurable, then z is $\mathcal{H}$-measurable.

We started off this section by talking about projecting onto linear subspaces. Recall that $S \subset \mathbb{R}^N$ is called a linear subspace of $\mathbb{R}^N$ if, given arbitrary scalars α, β and vectors $\mathbf{x}, \mathbf{y}$ in S, the linear combination $\alpha\mathbf{x} + \beta\mathbf{y}$ is again in S. Similarly $S \subset L_2$ is called a **linear subspace of L_2** if, given arbitrary scalars α, β and random variables x, y in S, the random variable $\alpha x + \beta y$ is also in S.

For conditional expectations, the subspaces of interest are the subspaces of measurable random variables. In particular, given $\mathcal{G} \subset L_2$, we define

$$L_2(\mathcal{G}) := \{\text{the set of all } \mathcal{G}\text{-measurable random variables in } L_2\}$$

In view of fact 3.3.1, we have the following important result:

Fact 3.3.3. For any $\mathcal{G} \subset L_2$, the set $L_2(\mathcal{G})$ is a linear subspace of L_2.

From fact 3.3.2 we see that, in the sense of set inclusion, the linear subspace is increasing with respect to the information set.

Fact 3.3.4. If $\mathcal{G} \subset \mathcal{H}$, then $L_2(\mathcal{G}) \subset L_2(\mathcal{H})$.

3.3.3 Conditional Expectation

Now it's time to define conditional expectations. Let $\mathcal{G} \subset L_2$ and y be some random variable in L_2. The **conditional expectation** of y given $\mathcal{G}$ is written as $\mathbb{E}[y \mid \mathcal{G}]$ or $\mathbb{E}^{\mathcal{G}}[y]$, and defined as the closest $\mathcal{G}$-measurable random variable to y.[6] More formally,

$$\mathbb{E}[y \mid \mathcal{G}] := \underset{z \in L_2(\mathcal{G})}{\operatorname{argmin}} \|\|y - z\|\| \tag{3.4}$$

This definition makes a lot of sense. Our intuitive understanding of the conditional expectation $\mathbb{E}[y \mid \mathcal{G}]$ is that it is the best predictor of y given the information contained in $\mathcal{G}$. The definition in (3.4) says the same thing. It simultaneously restricts $\mathbb{E}[y \mid \mathcal{G}]$ to be $\mathcal{G}$-measurable, so we can actually compute it once the variables in $\mathcal{G}$ are realized, and selects $\mathbb{E}[y \mid \mathcal{G}]$ as the closest such variable to y in terms of RMSE.

[6] I prefer the notation $\mathbb{E}^{\mathcal{G}}[y]$ to $\mathbb{E}[y \mid \mathcal{G}]$ because, as we will see, $\mathbb{E}^{\mathcal{G}}$ is a function (an orthogonal projection) from L_2 to L_2, and the former notation complements this view. However, the notation $\mathbb{E}[y \mid \mathcal{G}]$ is a bit more standard, so that's the one we'll use.

While the definition makes sense, it still leaves many open questions. For example, there are many situations where minimizers don't exist, or, if the do exist, there are lots of them. So is our definition really a definition? Moreover, even assuming we do have a proper definition, how do we actually go about computing conditional expectations in practical situations? And what properties do conditional expectations have?

These look like tricky questions, but fortunately the orthogonal projection theorem comes to the rescue. The orthogonal projection theorem in L_2 is almost identical to the orthogonal projection theorem we gave for $\mathbb{R}^N$. Given a linear subspace S of L_2 and a random variable y in L_2, there is a unique $\hat{y} \in S$ such that

$$|||y - \hat{y}||| \leq |||y - z||| \text{ for all } z \in S$$

The variable $\hat{y} \in S$ is called the **orthogonal projection** of y onto S.[7] Just as for the $\mathbb{R}^N$ case, the projection is characterized by two properties:

$$\hat{y} \text{ is the orthogonal projection of } y \text{ onto } S \text{ if and only if } \hat{y} \in S \text{ and } y - \hat{y} \perp S$$

As for $\mathbb{R}^N$, we can think of $y \mapsto \hat{y}$ as a function, which we denote by $\mathbf{P}$, so that $\mathbf{P}y$ is the orthogonal projection of y onto S for arbitrary $y \in L_2$. Moreover, $\mathbf{P}$ satisfies all the properties in theorem 3.1.3 (page 86). Let's state this as a theorem for the record.

Theorem 3.3.1. *Given a linear subspace S of L_2, the function*

$$\mathbf{P}y := \operatorname*{argmin}_{z \in S} |||y - z||| \tag{3.5}$$

is a well-defined linear function from L_2 to S. Given any $y \in L_2$, we have

1. $\mathbf{P}y \in S$,
2. $y - \mathbf{P}y \perp S$, *and*
3. $\mathbf{P}y = y$ *if and only if* $y \in S$.

[7] There are two small caveats I should mention. First, we actually require that S is a "closed" linear subspace of L_2, which means that if $\{x_n\} \subset S$, $x \in L_2$ and $|||x_n - x||| \to 0$, then $x \in S$. For the subspaces we consider here, this condition is always true. Second, when we talk about uniqueness in L_2, we do not distinguish between elements x and x' of L_2 such that $\mathbb{P}\{x = x'\} = 1$. A nice treatment of orthogonal projection in Hilbert spaces (of which L_2 is one example) is provided in Cheney (2001, chapter 2). Most other books covering Hilbert space will provide some discussion.

Comparing (3.4) and (3.5), we see that $y \mapsto \mathbb{E}[y \mid \mathcal{G}]$ is exactly the orthogonal projection function $\mathbf{P}$ in the special case where the subspace S is the $\mathcal{G}$-measurable functions $L_2(\mathcal{G})$.

Okay, so $\mathbb{E}[y \mid \mathcal{G}]$ is the orthogonal projection of y onto $L_2(\mathcal{G})$. That's kind of neat, but what does it actually tell us? Well, it tells us quite a lot. For starters, theorem 3.3.1 implies that $\mathbb{E}[y \mid \mathcal{G}]$ is always well defined and unique. Second, it gives us a useful characterization of $\mathbb{E}[y \mid \mathcal{G}]$, because we now know that $\mathbb{E}[y \mid \mathcal{G}]$ is the unique point in L_2 such that $\mathbb{E}[y \mid \mathcal{G}] \in L_2(\mathcal{G})$ and $y - \mathbb{E}[y \mid \mathcal{G}] \perp z$ for all $z \in L_2(\mathcal{G})$. Rewriting these conditions in a slightly different way, we can give an alternative (and equivalent) definition of conditional expectation: $\mathbb{E}[y \mid \mathcal{G}] \in L_2$ is the **conditional expectation** of y given $\mathcal{G}$ if

1. $\mathbb{E}[y \mid \mathcal{G}]$ is $\mathcal{G}$-measurable, and
2. $\mathbb{E}[\mathbb{E}[y \mid \mathcal{G}]\, z] = \mathbb{E}[yz]$ for all $\mathcal{G}$-measurable $z \in L_2$.

This definition seems a bit formidable, but it's not too hard to use. Before giving an application, let's bow to common notation and define

$$\mathbb{E}[y \mid x_1, \ldots, x_p] := \mathbb{E}[y \mid \mathcal{G}]$$

Also, let's note the following "obvious" fact:

Fact 3.3.5. Given $\{x_1, \ldots, x_p\}$ and y in L_2, there exists a function $g\colon \mathbb{R}^p \to \mathbb{R}$ such that $\mathbb{E}[y \mid x_1, \ldots, x_p] = g(x_1, \ldots, x_p)$.

This is obvious because, by definition, $\mathbb{E}[y \mid \mathcal{G}]$ is $\mathcal{G}$-measurable. At the same time, it's worth keeping in mind: A conditional expectation with respect to a collection of random variables is some function of those random variables.

Example 3.3.7. If x and w are independent and $y = x + w$, then $\mathbb{E}[y \mid x] = x + \mathbb{E}[w]$.

Let's check this using the second definition of conditional expectations given above. To check that $x + \mathbb{E}[w]$ is indeed the conditional expectation of y given $\mathcal{G} = \{x\}$, we need to show that $x + \mathbb{E}[w]$ is x-measurable and that $\mathbb{E}[(x + \mathbb{E}[w])\, z] = \mathbb{E}[yz]$ for all x-measurable z. The first claim is clearly true, because $x + \mathbb{E}[w]$ is a deterministic function of x. The second claim translates to the claim that

$$\mathbb{E}[(x + \mathbb{E}[w])g(x)] = \mathbb{E}[(x + w)g(x)] \tag{3.6}$$

for any function g. Verifying this equality is left as an exercise (exercise 3.4.12)

The next example shows that when x and y are linked by a conditional density (remember: densities don't always exist), then our definition of conditional expectation reduces to the one seen in elementary probability texts. The proof of the claim in the example is the topic of exercise 3.4.17.

Example 3.3.8. If x and y are random variables and $p(y \mid x)$ is the conditional density of y given x, then

$$\mathbb{E}[y \mid x] = \int t p(t \mid x) dt$$

There are some additional goodies we can harvest using the fact that conditional expectation is an orthogonal projection.

Fact 3.3.6. Let x and y be random variables in L_2, let α and β be scalars, and let $\mathcal{G}$ and $\mathcal{H}$ be subsets of L_2. The following properties hold.

1. Linearity: $\mathbb{E}[\alpha x + \beta y \mid \mathcal{G}] = \alpha \mathbb{E}[x \mid \mathcal{G}] + \beta \mathbb{E}[y \mid \mathcal{G}]$.
2. If $\mathcal{G} \subset \mathcal{H}$, then $\mathbb{E}[\mathbb{E}[y \mid \mathcal{H}] \mid \mathcal{G}] = \mathbb{E}[y \mid \mathcal{G}]$ and $\mathbb{E}[\mathbb{E}[y \mid \mathcal{G}]] = \mathbb{E}[y]$.
3. If y is independent of the variables in $\mathcal{G}$, then $\mathbb{E}[y \mid \mathcal{G}] = \mathbb{E}[y]$.
4. If y is $\mathcal{G}$-measurable, then $\mathbb{E}[y \mid \mathcal{G}] = y$.
5. If x is $\mathcal{G}$-measurable, then $\mathbb{E}[xy \mid \mathcal{G}] = x\mathbb{E}[y \mid \mathcal{G}]$.

Checking of these facts is mainly left to the exercises. Most are fairly straightforward. For example, consider the claim that if y is $\mathcal{G}$-measurable, then $\mathbb{E}[y \mid \mathcal{G}] = y$. In other words, we are saying that if $y \in L_2(\mathcal{G})$, then y is projected into itself. This is immediate from the last statement in theorem 3.3.1.

The fact that if $\mathcal{G} \subset \mathcal{H}$, then $\mathbb{E}[\mathbb{E}[y \mid \mathcal{H}] \mid \mathcal{G}] = \mathbb{E}[y \mid \mathcal{G}]$ is called the "tower" property of conditional expectations (by mathematicians), or the law of iterated expectations (by econometricians). The law follows from the property of orthogonal projections given in fact 3.1.2 on page 88: Projecting onto the bigger subspace $L_2(\mathcal{H})$ and from there onto $L_2(\mathcal{G})$ is the same as projecting directly onto the smaller subspace $L_2(\mathcal{G})$.

3.3.4 The Vector/Matrix Case

Conditional expectations of random matrices are defined using the notion of conditional expectations for scalar random variables. For example, given random matrices $\mathbf{X}$ and $\mathbf{Y}$, we set

$$\mathbb{E}\left[\mathbf{Y} \mid \mathbf{X}\right] := \begin{pmatrix} \mathbb{E}\left[y_{11} \mid \mathbf{X}\right] & \mathbb{E}\left[y_{12} \mid \mathbf{X}\right] & \cdots & \mathbb{E}\left[y_{1K} \mid \mathbf{X}\right] \\ \mathbb{E}\left[y_{21} \mid \mathbf{X}\right] & \mathbb{E}\left[y_{22} \mid \mathbf{X}\right] & \cdots & \mathbb{E}\left[y_{2K} \mid \mathbf{X}\right] \\ \vdots & \vdots & & \vdots \\ \mathbb{E}\left[y_{N1} \mid \mathbf{X}\right] & \mathbb{E}\left[y_{N2} \mid \mathbf{X}\right] & \cdots & \mathbb{E}\left[y_{NK} \mid \mathbf{X}\right] \end{pmatrix}$$

where

$$\mathbb{E}\left[y_{nk} \mid \mathbf{X}\right] := \mathbb{E}\left[y_{nk} \mid x_{11}, \ldots, x_{\ell m}, \ldots, x_{LM}\right]$$

We also define

$$\operatorname{cov}[\mathbf{x}, \mathbf{y} \mid \mathbf{Z}] := \mathbb{E}\left[\mathbf{x}\mathbf{y}' \mid \mathbf{Z}\right] - \mathbb{E}\left[\mathbf{x} \mid \mathbf{Z}\right]\mathbb{E}\left[\mathbf{y} \mid \mathbf{Z}\right]'$$

and

$$\operatorname{var}[\mathbf{x} \mid \mathbf{Z}] := \mathbb{E}\left[\mathbf{x}\mathbf{x}' \mid \mathbf{Z}\right] - \mathbb{E}\left[\mathbf{x} \mid \mathbf{Z}\right]\mathbb{E}\left[\mathbf{x} \mid \mathbf{Z}\right]'$$

Using the definitions, one can show that all of the results on conditional expectations in fact 3.3.6 continue to hold in the current setting, replacing scalars with vectors and matrices. We state necessary results for convenience:

Fact 3.3.7. Let $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ be random matrices, and let $\mathbf{A}$ and $\mathbf{B}$ be constant matrices. Assuming conformability of matrix operations, the following results hold:

1. $\mathbb{E}\left[\mathbf{Y} \mid \mathbf{Z}\right]' = \mathbb{E}\left[\mathbf{Y}' \mid \mathbf{Z}\right]$.
2. $\mathbb{E}\left[\mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{Y} \mid \mathbf{Z}\right] = \mathbf{A}\mathbb{E}\left[\mathbf{X} \mid \mathbf{Z}\right] + \mathbf{B}\mathbb{E}\left[\mathbf{Y} \mid \mathbf{Z}\right]$.
3. $\mathbb{E}\left[\mathbb{E}\left[\mathbf{Y} \mid \mathbf{X}\right]\right] = \mathbb{E}\left[\mathbf{Y}\right]$ and $\mathbb{E}\left[\mathbb{E}\left[\mathbf{Y} \mid \mathbf{X}, \mathbf{Z}\right] \mid \mathbf{X}\right] = \mathbb{E}\left[\mathbf{Y} \mid \mathbf{X}\right]$.
4. If $\mathbf{X}$ and $\mathbf{Y}$ are independent, then $\mathbb{E}\left[\mathbf{Y} \mid \mathbf{X}\right] = \mathbb{E}\left[\mathbf{Y}\right]$.
5. If g is a (nonrandom) function, so that $g(\mathbf{X})$ is a matrix depending only on $\mathbf{X}$, then
   - $\mathbb{E}\left[g(\mathbf{X}) \mid \mathbf{X}\right] = g(\mathbf{X})$
   - $\mathbb{E}\left[g(\mathbf{X})\,\mathbf{Y} \mid \mathbf{X}\right] = g(\mathbf{X})\mathbb{E}\left[\mathbf{Y} \mid \mathbf{X}\right]$
   - $\mathbb{E}\left[\mathbf{Y}\,g(\mathbf{X}) \mid \mathbf{X}\right] = \mathbb{E}\left[\mathbf{Y} \mid \mathbf{X}\right] g(\mathbf{X})$

3.3.5 An Exercise in Conditional Expectations

Let x and y be two random variables. We saw that $\mathbb{E}[y \mid x]$ is a function f of x such that $f(x)$ is the best predictor of y in terms of root mean squared error. Since monotone increasing transformations do not affect minimizers, f also minimizes the mean squared error. In other words, f solves

$$\min_{g \in G} \mathbb{E}\left[(y - g(x))^2\right] \tag{3.7}$$

where G is the set of functions from $\mathbb{R}$ to $\mathbb{R}$. From this definition of conditional expectations, we employed the orthogonal projection theorem to deduce various properties of conditional expectations. We can also reverse this process, showing directly that $f(x) := \mathbb{E}[y \mid x]$ solves (3.7), given the various properties of conditional expectations listed in fact 3.3.6. To begin, suppose that the properties in fact 3.3.6 hold, and fix an arbitrary $g \in G$. We have

$$\begin{aligned}(y - g(x))^2 &= (y - f(x) + f(x) - g(x))^2 \\ &= (y - f(x))^2 + 2(y - f(x))(f(x) - g(x)) + (f(x) - g(x))^2\end{aligned}$$

Let's consider the expectation of the cross-product term. From the law of iterated expectations (fact 3.3.6), we obtain

$$\mathbb{E}\left\{(y - f(x))(f(x) - g(x))\right\} = \mathbb{E}\left\{\mathbb{E}\left[(y - f(x))(f(x) - g(x)) \mid x\right]\right\} \tag{3.8}$$

We can re-write the term inside the curly brackets on the right-hand side of (3.8) as

$$(f(x) - g(x))\mathbb{E}\left[(y - f(x)) \mid x\right]$$

(Which part of fact 3.3.6 are we using here?) Regarding the second term in this product, we have (by which facts?) the result

$$\mathbb{E}[y - f(x) \mid x] = \mathbb{E}[y \mid x] - \mathbb{E}[f(x) \mid x] = \mathbb{E}[y \mid x] - f(x) = \mathbb{E}[y \mid x] - \mathbb{E}[y \mid x] = 0$$

We conclude that the expectation in (3.8) is $\mathbb{E}[0] = 0$. It then follows that

$$\begin{aligned}\mathbb{E}\left[(y - g(x))^2\right] &= \mathbb{E}\left[(y - f(x))^2 + 2(y - f(x))(f(x) - g(x)) + (f(x) - g(x))^2\right] \\ &= \mathbb{E}\left[(y - f(x))^2\right] + \mathbb{E}\left[(f(x) - g(x))^2\right]\end{aligned}$$

Since $(f(x) - g(x))^2 \geq 0$ we have $\mathbb{E}\left[(f(x) - g(x))^2\right] \geq 0$, and we conclude that

$$\mathbb{E}\left[(y - g(x))^2\right] \geq \mathbb{E}\left[(y - f(x))^2\right] =: \mathbb{E}\left[(y - \mathbb{E}[y \mid x])^2\right]$$

Since g was an arbitrary element of G, we conclude that

$$f = \operatorname*{argmin}_{g \in G} \mathbb{E}\left[(y - g(x))^2\right]$$

3.4 Exercises

Ex. 3.4.1. Prove the Pythagorean law in theorem 3.1.1.[8]

Ex. 3.4.2. Prove theorem 3.1.4 using theorems 3.1.2–3.1.3.

Ex. 3.4.3. Prove fact 3.1.4: If $S \subset \mathbb{R}^N$, then $S \cap S^\perp = \{\mathbf{0}\}$.

Ex. 3.4.4. Prove fact 3.1.2.

Ex. 3.4.5. Let $\mathbf{x}$ and $\mathbf{y}$ be any two $N \times 1$ vectors.

1. Show that $\|\mathbf{x} + \mathbf{y}\|^2 = \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 + 2\mathbf{x}'\mathbf{y}$
2. Explain the connection between this equality and the Pythagorean Law.

Ex. 3.4.6. Let $\mathbf{P}$ be the orthogonal projection described in theorem 3.1.3 (page 86). Confirm that $\mathbf{P}$ is a linear function from $\mathbb{R}^N$ to $\mathbb{R}^N$, as defined in §2.1.3.

Ex. 3.4.7. Let $S := \{(x_1, x_2, x_3) \in \mathbb{R}^3 : x_3 = 0\}$, and let $\mathbf{y} := \mathbf{1} := (1, 1, 1)$. Using the orthogonal projection theorem, find the closest point in S to $\mathbf{y}$.

Ex. 3.4.8. Let $\mathbf{P}$ be the orthogonal projection described in theorem 3.1.3 (page 86). Is it true that $\mathbf{Px} \neq \mathbf{Py}$ whenever $\mathbf{x} \neq \mathbf{y}$? Why or why not?[9]

Ex. 3.4.9. Show that when $N \times K$ matrix $\mathbf{X}$ is full column rank, the matrix $\mathbf{X}'\mathbf{X}$ is invertible.[10]

Ex. 3.4.10. Show by direct computation that $\mathbf{P}$ and $\mathbf{M}$ in (3.1) and (3.2) are both symmetric and idempotent.

Ex. 3.4.11. Verify fact 3.2.2 (i.e., $\mathbf{MX} = \mathbf{0}$) directly using matrix algebra.

Ex. 3.4.12. Show that the equality in (3.6) holds when x and w are independent.

Ex. 3.4.13. In fact 3.3.6, it is stated that if y is independent of the variables in $\mathcal{G}$, then $\mathbb{E}[y \mid \mathcal{G}] = \mathbb{E}[y]$. Prove this using the (second) definition of the conditional expectation $\mathbb{E}[y \mid \mathcal{G}]$. To make the proof a bit simpler, you can take $\mathcal{G} = \{x\}$.

[8] Hint: See fact 2.2.1.

[9] Hint: Sketch the graph and think about it visually.

[10] Hint: This is non-trivial. In view of fact 2.3.11, it suffices to show that $\mathbf{X}'\mathbf{X}$ is positive definite. Make use of the full column rank assumption. Look at the different equivalent conditions for linear independence of a set of vectors.

Ex. 3.4.14. Confirm the claim in fact 3.3.6 that if x is $\mathcal{G}$-measurable, then $\mathbb{E}[xy \mid \mathcal{G}] = x\mathbb{E}[y \mid \mathcal{G}]$.

Ex. 3.4.15. Let $\operatorname{var}[y \mid x] := \mathbb{E}[y^2 \mid x] - (\mathbb{E}[y \mid x])^2$. Show that

$$\operatorname{var}[y] = \mathbb{E}[\operatorname{var}[y \mid x]] + \operatorname{var}[\mathbb{E}[y \mid x]]$$

Ex. 3.4.16. Show that the conditional expectation of a constant α is α. In particular, using the results in fact 3.3.6 (page 100) as appropriate, show that if α is a constant and $\mathcal{G}$ is any information set, then $\mathbb{E}[\alpha \mid \mathcal{G}] = \alpha$.

Ex. 3.4.17. Prove the claim in example 3.3.8. (Warning: The proof is a little advanced and you should be comfortable manipulating double integrals.)

3.4.1 Solutions to Selected Exercises

Solution to Exercise 3.4.3. Let $S \subset \mathbb{R}^N$. We aim to show that $S \cap S^\perp = \{\mathbf{0}\}$. Fix $\mathbf{a} \in S \cap S^\perp$. Since $\mathbf{a} \in S^\perp$, we know that $\mathbf{a}'\mathbf{s} = 0$ for any $\mathbf{s} \in S$. Since $\mathbf{a} \in S$, we have in particular, $\mathbf{a}'\mathbf{a} = \|\mathbf{a}\|^2 = 0$. As we saw in fact 2.1.1, the only such vector is $\mathbf{0}$. □

Solution to Exercise 3.4.6. Fix $\alpha, \beta \in \mathbb{R}$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$. The claim is that

$$\mathbf{P}(\alpha\mathbf{x} + \beta\mathbf{y}) = \alpha\mathbf{P}\mathbf{x} + \beta\mathbf{P}\mathbf{y}$$

To verify this equality, we need to show that the right-hand side is the orthogonal projection of $\alpha\mathbf{x} + \beta\mathbf{y}$ onto S. Going back to theorem 3.1.2, we need to show that (i) $\alpha\mathbf{P}\mathbf{x} + \beta\mathbf{P}\mathbf{y} \in S$ and (ii) for any $\mathbf{z} \in S$, we have

$$(\alpha\mathbf{x} + \beta\mathbf{y} - (\alpha\mathbf{P}\mathbf{x} + \beta\mathbf{P}\mathbf{y}))'\mathbf{z} = 0$$

Here (i) is immediate, because $\mathbf{P}\mathbf{x}$ and $\mathbf{P}\mathbf{y}$ are in S by definition; and, moreover S is a linear subspace. To see that (ii) holds, just note that

$$(\alpha\mathbf{x} + \beta\mathbf{y} - (\alpha\mathbf{P}\mathbf{x} + \beta\mathbf{P}\mathbf{y}))'\mathbf{z} = \alpha(\mathbf{x} - \mathbf{P}\mathbf{x})'\mathbf{z} + \beta(\mathbf{y} - \mathbf{P}\mathbf{y})'\mathbf{z}$$

By definition, the projections of $\mathbf{x}$ and $\mathbf{y}$ are orthogonal to S, so we have $(\mathbf{x} - \mathbf{P}\mathbf{x})'\mathbf{z} = (\mathbf{y} - \mathbf{P}\mathbf{y})'\mathbf{z} = 0$. We are done. □

Solution to Exercise 3.4.7. Let $\mathbf{x} = (x_1, x_2, x_3)$ be the closest point in S to $\mathbf{y}$. Note that $\mathbf{e}_1 \in S$ and $\mathbf{e}_2 \in S$. By the orthogonal projection theorem we have (i) $\mathbf{x} \in S$, and (ii) $\mathbf{y} - \mathbf{x} \perp S$. From (i) we have $x_3 = 0$. From (ii) we have

$$\langle \mathbf{y} - \mathbf{x}, \mathbf{e}_1 \rangle = 0 \quad \text{and} \quad \langle \mathbf{y} - \mathbf{x}, \mathbf{e}_2 \rangle = 0$$

These equations can be expressed more simply as $1 - x_1 = 0$ and $1 - x_2 = 0$. We conclude that $\mathbf{x} = (1, 1, 0)$. □

Solution to Exercise 3.4.8. It is false to say that $\mathbf{Px} \neq \mathbf{Py}$ whenever $\mathbf{x} \neq \mathbf{y}$: We can find examples of vectors $\mathbf{x}$ and $\mathbf{y}$ such that $\mathbf{x} \neq \mathbf{y}$ but $\mathbf{Px} = \mathbf{Py}$. Indeed, if we fix any $\mathbf{y}$ and then set $\mathbf{x} = \mathbf{Py} + \alpha \mathbf{My}$ for some constant α, you should be able to confirm that $\mathbf{Px} = \mathbf{Py}$, and also that $\mathbf{x} \neq \mathbf{y}$ when $\alpha \neq 1$. □

Solution to Exercise 3.4.9. Let $\mathbf{A} = \mathbf{X}'\mathbf{X}$. It suffices to show that $\mathbf{A}$ is positive definite, since this implies that its determinant is strictly positive, and any matrix with nonzero determinant is invertible. To see that $\mathbf{A}$ is positive definite, pick any $\mathbf{b} \neq \mathbf{0}$. We must show that $\mathbf{b}'\mathbf{Ab} > 0$. To see this, observe that

$$\mathbf{b}'\mathbf{Ab} = \mathbf{b}'\mathbf{X}'\mathbf{Xb} = (\mathbf{Xb})'\mathbf{Xb} = \|\mathbf{Xb}\|^2$$

By the properties of norms, this last term is zero only when $\mathbf{Xb} = \mathbf{0}$. But this is not true, because $\mathbf{b} \neq \mathbf{0}$ and $\mathbf{X}$ is full column rank (see fact 2.2.4, part 5). □

Solution to Exercise 3.4.12. Let g be any function from $\mathbb{R} \to \mathbb{R}$. Given independence of x and w (and applying fact 1.3.2 on page 27), we have

$$\begin{aligned} \mathbb{E}\left[(x + \mathbb{E}[w])g(x)\right] &= \mathbb{E}[xg(x)] + \mathbb{E}[w]\mathbb{E}[g(x)] \\ &= \mathbb{E}[xg(x)] + \mathbb{E}[wg(x)] \\ &= \mathbb{E}[(x + w)g(x)] \end{aligned}$$

This confirms (3.6). □

Solution to Exercise 3.4.13. Let y be independent of x. From the (second) definition of conditional expectation, to show that $\mathbb{E}[y \mid x] = \mathbb{E}[y]$ we need to show that

1. $\mathbb{E}[y]$ is $\mathcal{G}$-measurable, and
2. $\mathbb{E}[\mathbb{E}[y]g(x)] = \mathbb{E}[yg(x)]$ for any function $g\colon \mathbb{R} \to \mathbb{R}$.

Part 1 is immediate, because $\mathbb{E}[y]$ is constant (see example 3.3.5 on page 96). Regarding part 2, if g is any function, then by facts 1.3.1 and 1.3.2 (see page 27) we have $\mathbb{E}[yg(x)] = \mathbb{E}[y]\mathbb{E}[g(x)]$. By linearity of expectations, $\mathbb{E}[y]\mathbb{E}[g(x)] = \mathbb{E}[\mathbb{E}[y]g(x)]$. □

Solution to Exercise 3.4.14. We need to show that if x is $\mathcal{G}$-measurable, then $\mathbb{E}[xy \mid \mathcal{G}] = x\mathbb{E}[y \mid \mathcal{G}]$. To confirm this, we must show that

1. $x\mathbb{E}[y \mid \mathcal{G}]$ is $\mathcal{G}$-measurable, and
2. $\mathbb{E}[x\mathbb{E}[y \mid \mathcal{G}]z] = \mathbb{E}[xyz]$ for any $z \in L_2(\mathcal{G})$.

Regarding part 1, $\mathbb{E}[y \mid \mathcal{G}]$ is $\mathcal{G}$-measurable by definition, and x is $\mathcal{G}$-measurable by assumption, so $x\mathbb{E}[y \mid \mathcal{G}]$ is $\mathcal{G}$-measurable by fact 3.3.1 on page 96. Regarding part 2, fix $z \in L_2(\mathcal{G})$, and let $u := xz$. Since $x \in L_2(\mathcal{G})$, we have $u \in L_2(\mathcal{G})$. We need to show that

$$\mathbb{E}[\mathbb{E}[y \mid \mathcal{G}]u] = \mathbb{E}[yu]$$

Since $u \in L_2(\mathcal{G})$, this is immediate from the definition of $\mathbb{E}[y \mid \mathcal{G}]$. □

Solution to Exercise 3.4.16. By fact 3.3.6 (page 100), we know that if α is $\mathcal{G}$-measurable, then $\mathbb{E}[\alpha \mid \mathcal{G}] = \alpha$. Example 3.3.5 on page 96 tells us that α is indeed $\mathcal{G}$-measurable. □

Solution to Exercise 3.4.17. As in example 3.3.8, let x and y be random variables where $p(y \mid x)$ is the conditional density of y given x. Let $g(x) := \int t p(t \mid x) dt$. The claim is that $\mathbb{E}[y \mid x] = g(x)$. To prove this, we need to show that $g(x)$ is x-measurable, and that

$$\mathbb{E}[g(x)h(x)] = \mathbb{E}[yh(x)] \quad \text{for any function} \quad h \colon \mathbb{R} \to \mathbb{R} \tag{3.9}$$

The first claim is obvious. Regarding (3.9), let h be any such function. Using the notation in (1.20) on page 26, we can write

$$\begin{aligned}
\mathbb{E}[g(x)h(x)] &= \mathbb{E}\left[\int t p(t \mid x) dt \; h(x)\right] \\
&= \int \int t p(t \mid s) dt \; h(s) p(s) ds \\
&= \int \int t \frac{p(s,t)}{p(s)} dt \; h(s) p(s) ds \\
&= \int \int t \, h(s) p(s,t) dt ds
\end{aligned}$$

This is equal to the right-hand side of (3.9), and the proof is done. □

Part II

Foundations of Statistics

Chapter 4

Statistical Learning

Econometrics is just statistics applied to economic problems—nothing more and nothing less. We should probably call it "statistical economics," but I guess people feel that the term "econometrics" has a better ring to it. The only cost of using the term "econometrics" is that we are sometimes fooled into thinking that we work on a distinct discipline, separate from statistics. This is not true.

The next two chapters provides a short, concise review of the foundations of modern statistics, including parametric and nonparametric methods, empirical distributions, hypothesis testing and confidence intervals.

4.1 Inductive Learning

In the modern world we have lots of data, but still lack fundamental knowledge on how many systems work, or how different economic variables are related to one another. What then is the process of extracting knowledge from data? Under what conditions will this process be successful?

4.1.1 Generalization

The fundamental problem of statistics is learning from data. Learning from data concerns *generalization*. A finite set of data is observed, and, on the basis of this data, one seeks to make more general statements. For example, suppose that a certain drug is tested on 1,000 volunteers, and found to produce the desired effect in 95%

of cases. On the basis of this study, the drug company claims that the drug is highly effective. The implication of their claim is that we can *generalize* to the wider population. The interest is not so much in what happened to the volunteers themselves, but rather on what the outcome for the volunteers implies *for other people*.

Another word for generalization is *induction*. Inductive learning is where reasoning proceeds from the specific to the general—as opposed to deductive learning, which proceeds from general to specific.

Example 4.1.1. You show a child pictures of dogs in a book and say 'dog'. After a while, the child sees a dog on the street and says 'dog'. The child has *generalized* from specific examples. Hence, the learning is inductive. If, on the other hand, you had told the child that dogs are hairy, four legged animals that stick their tongues out when hot, and the child determined the creature was a dog on this basis, then the nature of the learning process could be called deductive.

Here are some typical statistical problems, phrased in more mathematical language:

Example 4.1.2. N random values $x_1, \ldots, x_N$ are drawn from a given but unknown cdf F. We wish to learn about F from this sample.

Example 4.1.3. Same as example 4.1.2, but now we only care about learning the mean of F—or the standard deviation, or the median, etc.

Example 4.1.4 (Regression)**.** We observe "inputs" $x_1, \ldots, x_N$ to some "system," as well as the corresponding "outputs" $y_1, \ldots, y_N$. Given this data, we wish to compute a function f such that, given a new input/output pair (x, y), the value $f(x)$ will be a good guess of the corresponding output y. (Here we imagine that y is observed after x or not at all, and hence the need to predict y from x.)

In these examples, the problem lies in the fact that we do not know the underlying distributions. If, in example 4.1.4, we knew the joint distribution of (x, y) pairs, then we could work out the conditional expectation $\mathbb{E}[y \mid x]$. As we'll see in §3.3.3, there is a natural sense in which the conditional expectation is the best predictor of y given x. In statistical applications, however, we don't know the distributions. All we have is the observations. We must do the best we can given the information contained in this sample.

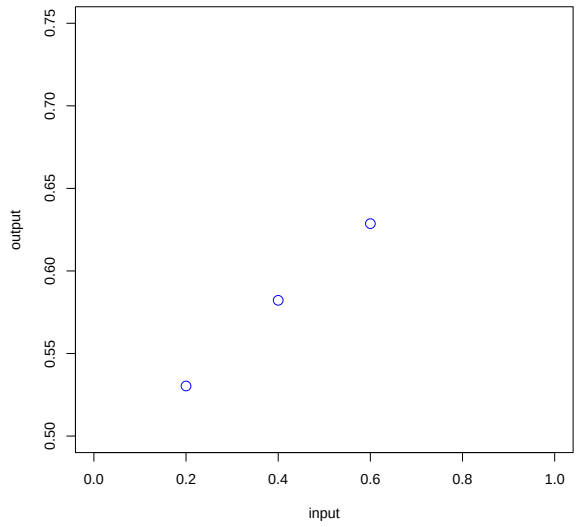


Figure 4.1: Generalization requires knowledge

4.1.2 Data is not Enough

As a rule, statistical learning requires more than just data. Ideally, data is combined with a theoretical model that encapsulates our knowledge of the system we are studying. The data is often used to pin down parameter values for the model. This is called fitting the model to the data. If our model is good, then combining model with data allows to gain an understanding of how the system works.

Even when we have no formal model of how the system works, we still need to combine the data with *some* assumptions in order to generalize. Figure 4.1 helps to illustrate this idea. Consider the regression setting of example 4.1.4, and suppose we observe the blue dots as our data. Now make a subjective guess as to the likely value of the output, given that the input value is 0.8. Was your guess something like the red dot in Figure 4.2? It looks reasonable to me too.

But why does it look reasonable? Because our brain picks up a pattern: The blue dots lie roughly on a straight line. We instictively predict that the red dot will lie on the same line, or at least we feel it would be natural for that to occur. One way or another, our brains have been trained (or are hard-wired?) to think in straight lines. And even though this thought process is subconscious, in the end what we are doing is bringing our own assumptions into play.

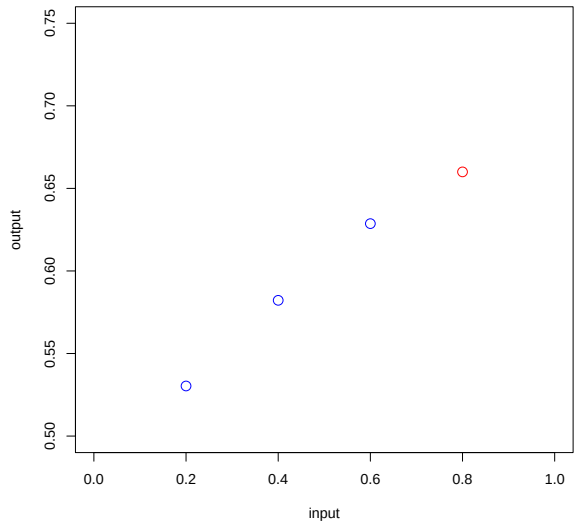


Figure 4.2: Generalization requires knowledge

Obviously our assumption about the linear relationship could be completely wrong. After all, we haven't even talked about the kind of system we are observing here. Maybe the functional relationship between inputs and outputs is totally different to what we perceived from these few data points. Ideally, our assumptions should be based on sound theory and understanding of the system we are studying, rather than some subconscious feeling that straight lines are most likely.[1]

Either way, regardless of the process that led to our assumptions, the point is that *we cannot forecast the new observation from the data alone*. We have to make *some* assumptions as to the functional relationship in order to come up with a guess of likely output given input 0.8. Those assumptions may come from knowledge of the system, or they may come from subconscious preference for straight lines. Either way, we are adding something to the data in order to make inference about likely outcomes.

If the assumptions we add to the data are to some extent correct, this injection of prior knowledge into the learning process allows us to *generalize* from the observed

[1] In 1929, the economist Irving Fisher famously declared that "Stocks have reached what looks like a permanently high plateau." Perhaps Dr Fisher based his projection on subconscious attraction to straight lines, rather than some deeper understanding of the underlying forces generating the time series of equity prices he observed.

data points. Thus, roughly speaking, the rule is

$$\text{statistical learning} = \text{prior knowledge} + \text{data}$$

4.2 Statistics

"Statistics" sounds like an odd name for a section. Isn't this whole course about statistics? Yes, sure it is, but here we're using the term *statistic* with a special meaning. Specifically, a statistic is any function of a given data set. To repeat:

- A **statistic** is an observable function of the sample data.

For example, suppose that we have data $x_1, \ldots, x_N$, which might represent the price of a Big Mac in N different countries, or the closing price of one share in Google over N consecutive days. Common statistics used to summarize the data are the **sample mean**

$$\bar{x}_N :=: \bar{x} := \frac{1}{N} \sum_{n=1}^{N} x_n$$

the **sample variance**

$$s_N^2 :=: s^2 := \frac{1}{N-1} \sum_{n=1}^{N} (x_n - \bar{x})^2 \tag{4.1}$$

and the **sample standard deviation**

$$s_N :=: s := \sqrt{s^2} = \left[\frac{1}{N-1} \sum_{n=1}^{N} (x_n - \bar{x})^2 \right]^{1/2} \tag{4.2}$$

For positive integer k, the **sample k-th moment** is given by

$$\frac{1}{N} \sum_{n=1}^{N} x_n^k$$

If we have bivariate data $(x_1, y_1), \ldots, (x_N, y_N)$, then the **sample covariance** is

$$\frac{1}{N-1} \sum_{n=1}^{N} (x_n - \bar{x})(y_n - \bar{y}) \tag{4.3}$$

and the **sample correlation** is the sample covariance divided by the product of the two sample standard deviations. With some rearranging, this becomes

$$\frac{\sum_{n=1}^{N}(x_n - \bar{x})(y_n - \bar{y})}{\sqrt{\sum_{n=1}^{N}(x_n - \bar{x})^2 \sum_{n=1}^{N}(y_n - \bar{y})^2}} \tag{4.4}$$

R has functions for all of these common statistics. The sample mean, sample variance and sample standard deviation are calculated using the functions `mean`, `var` and `sd` respectively. Sample covariance and sample correlation can be computed using `cov` and `cor`:

```
> x <- rnorm(10)
> y <- rnorm(10)
> cov(x, y)
[1] 0.001906421
> cor(x, y)
[1] 0.004054976
```

Perhaps the most important thing to remember about statistics is that, being functions of the sample, *they are also random variables.* This might not be clear, since, we tend to think of the data as a fixed set of numbers in a file on our hard disk, determined by some previous historical outcome. Statistics are deterministic functions of these numbers, and we only observe one value of any particular statistic—one sample mean, one sample variance, etc. However, the way that statisticians think about it is that they imagine designing the statistical exercise *prior to observing the data*. At this stage, the data is regarded as a collection of random variables—even though these variables may have been previously determined in some historical data set. Hence, each statistic is also a random quantity (i.e., random variable).

More formally, if we look at the sample mean, for example, when we write $\bar{x}_N := N^{-1}\sum_{n=1}^{N} x_n$, what we actually mean is

$$\bar{x}_N(\omega) := \frac{1}{N}\sum_{n=1}^{N} x_n(\omega) \qquad (\omega \in \Omega) \tag{4.5}$$

Hence $\bar{x}_N$ is a function from $\Omega \to \mathbb{R}$. Put differently, $\bar{x}_N$ is a random variable.

Being random variables, statistics have expectations, variances and so on. For example, consider the sample mean $\bar{x} := N^{-1}\sum_n x_n$ of an identically distributed sample. Each observation x_n is drawn from some fixed distribution F with unknown mean

μ. In that case, the mean of $\bar{x}$ is also μ, because, from linearity of expectations,

$$\mathbb{E}[\bar{x}] = \mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N} x_n\right] = \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}[x_n] = \mu \tag{4.6}$$

Since we don't actually know what μ is, this result might not seem very helpful, but it is: It tells us that $\bar{x}$ is a useful predictor of the unknown quantity μ, in the sense that it's "most likely" outcome is this unknown quantity. We say that $\bar{x}$ is *unbiased* for μ. The next session discusses this and other properties of estimators.

4.2.1 Vector Statistics

Statistics can be vector valued, or even matrix valued. For example, if $\mathbf{x}_1, \ldots, \mathbf{x}_N$ are random vectors of equal length, then the sample mean is the random vector defined by

$$\bar{\mathbf{x}} := \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

and the **sample variance-covariance matrix** is defined as

$$\mathbf{Q} := \frac{1}{N-1}\sum_{n=1}^{N}[(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})'] \tag{4.7}$$

In R, the sample variance-covariance matrix is obtained using `cov`, which acts on matrices. In relation to $\mathbf{Q}$ in (4.7), the observed vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$ are treated as row vectors of the matrix. This is a standard convention in statistics: rows of a matrix are observations of a random vector. Hence, to obtain the sample variance covariance matrix $\mathbf{Q}$ of observations $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$, we stack them into rows and use `cov`:

```
> cov(rbind(x1, x3, x2))
```

More commonly, we will be working with a data matrix `X` where the rows are observations, and we can just use `cov(X)`.

4.2.2 Estimators and their Properties

Estimators are just statistics—that is, functions of the data. However, when we talk about estimators, we have in mind the idea of estimating a specific quantity

of interest. In other words, to discuss an estimator, we need to also specify what it is we're trying to estimate. For example, suppose we wish to estimate the mean $\mu := \int s\, F(ds)$ of F given observations $x_1, \ldots, x_N$ from F. A common way to do this is to use the sample mean $\bar{x}$ of the observations $x_1, \ldots, x_N$. In this setting, $\bar{x}$ is regarded as an *estimator* of μ.

In any given problem, there are always many estimators we can use. For example, when estimating the mean in the preceding problem we can also use the so-called mid-range estimator

$$m_N := \frac{\min_n x_n + \max_n x_n}{2}$$

Which is better, the sample mean or the mid-range estimator?

More generally, let us consider the problem of what makes a good estimator. Not surprisingly, that depends on how you define "good." To begin the discussion, first we collect some terminology. Let $\hat{\theta}$ be an estimator of θ. The **bias** of $\hat{\theta}$ is defined as

$$\text{bias}[\hat{\theta}] := \mathbb{E}\,[\hat{\theta}] - \theta \tag{4.8}$$

The estimator $\hat{\theta}$ is called **unbiased** for θ if its bias is zero, or $\mathbb{E}\,[\hat{\theta}] = \theta$. The **mean squared error** of a given estimator $\hat{\theta}$ of some fixed quantity θ is

$$\text{mse}[\hat{\theta}] := \mathbb{E}\,[(\hat{\theta} - \theta)^2] \tag{4.9}$$

Low mean squared error means that probability mass is concentrated around θ.[2]

Example 4.2.1. As we saw in (4.6), if $x_1, \ldots, x_N$ is identically distributed, then the sample mean $\bar{x}$ is always an unbiased estimator of the mean. As a result, the mean squared error of $\bar{x}$ is equal to the variance. If, in addition, the random variables $x_1, \ldots, x_N$ in question are uncorrelated, then the variance of $\bar{x}$ is

$$\text{var}[\bar{x}] = \text{var}\left[\frac{1}{N}\sum_{n=1}^{N} x_n\right] = \frac{\sigma^2}{N} \tag{4.10}$$

where $\sigma^2 := \int (s - \theta)^2 F(ds)$ is the common variance of each x_n.

Example 4.2.2. For an IID sample, the sample variance s^2 is an unbiased estimator of the variance (exercise 4.7.2).

[2] In the definition of mean squared error, we are implicitly assuming that the expression on the right hand side of (4.9) is finite. This may not be true for certain estimators, simply because the second moment of the estimator $= \infty$.

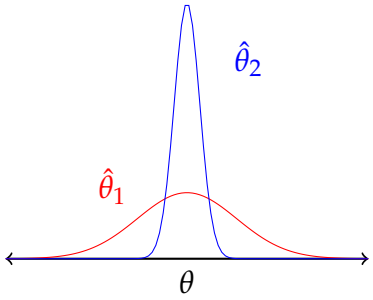


Figure 4.3: Unbiased estimators

Example 4.2.3. The mid-range estimator m_N may be biased as an estimator of the mean of a distribution, depending on the distribution in question. Bias in the log-normal case is illustrated in the following simulation:

```
> mr <- function(x) return((min(x) + max(x)) / 2)
> observations <- replicate(5000, mr(rlnorm(20)))
> mean(observations)   # Sample mean of mid-range
[1] 3.800108
> exp(1/2)       # Mean of lognormal
[1] 1.648721
```

By the LLN, the sample mean of m_N is close to $\mathbb{E}[m_N]$. This is clearly a long way from the mean of the lognormal density.

For an unbiased estimator $\hat{\theta}$, low variance is desirable. Low variance means that probability mass for the random variable $\hat{\theta}$ is concentrated around its mean (see figure 4.3). This is precisely what we want, since the mean of an unbiased estimator is the quantity θ that we wish to estimate.

One issue here is that low variance is a bit hard to quantify. For example, consider the variance of the sample mean, as given in (4.10). Is that low or is it not? One way to approach this kind of question is to take the class of unbiased estimators of a given quantity θ, and find the estimator in the class with the lowest variance. For given θ and given data $x_1, \ldots, x_N$, the estimator in the set of unbiased estimators

$$U_\theta := \{\text{all statistics } \hat{\theta} \text{ with } \mathbb{E}[\hat{\theta}] = \theta\}$$

that has the lowest variance within this class (i.e., set) is called the **minimum variance unbiased estimator**.

While minimum variance unbiased estimators are nice in theory, it's certainly possible that no minimizer exists. Second, even if such an estimator does exist in this class, it may be hard to determine in practice. Hence, there is a tendency to focus on smaller classes than U_θ, and find the best estimator in that class. For example, the estimator in the set of *linear* unbiased estimators

$$U_\theta^\ell := \{\text{all linear statistics } \hat{\theta} \text{ with } \mathbb{E}[\hat{\theta}] = \theta\}$$

with the lowest variance—if it exists—is called the **best linear unbiased estimator**, or **BLUE**. Here "linear" means that $\hat{\theta}$ is a linear function of the data $x_1, \ldots, x_N$. Linearity will be defined formally in §2.1.3. For now let's just look at an example.

Example 4.2.4. Let $x_1, \ldots, x_N \overset{\text{IID}}{\sim} F$, where F has finite mean $\mu \neq 0$ and variance σ^2. Take it on trust for now that the set of linear estimators of μ is given by

$$\left\{\text{all statistics of the form } \hat{\mu} = \sum_{n=1}^N \alpha_n x_n, \text{ where } \alpha_n \in \mathbb{R} \text{ for } n = 1, \ldots, N\right\}$$

Hence, the set linear *unbiased* estimators of μ is given by

$$U_\mu^\ell := \left\{\text{all } \hat{\mu} = \sum_{n=1}^N \alpha_n x_n \text{ with } \alpha_n \in \mathbb{R}, \;\; n = 1, \ldots, N \text{ and } \mathbb{E}\left[\sum_{n=1}^N \alpha_n x_n\right] = \mu\right\}$$

Using linearity of expectations, we see that this set can be re-written as

$$U_\mu^\ell := \left\{\text{all } \hat{\mu} = \sum_{n=1}^N \alpha_n x_n \text{ with } \sum_{n=1}^N \alpha_n = 1\right\}$$

By fact 1.3.9 on page 28, the variance of an element of this class is given by

$$\text{var}\left[\sum_{n=1}^N \alpha_n x_n\right] = \sum_{n=1}^N \alpha_n^2 \,\text{var}[x_n] + 2\sum_{n<m} \alpha_n \alpha_m \,\text{cov}[x_n, x_m] = \sigma^2 \sum_{n=1}^N \alpha_n^2$$

where the last equality is due to independence. To find the BLUE, we need to solve

$$\text{minimize } \sigma^2 \sum_{n=1}^N \alpha_n^2 \text{ over all } \alpha_1, \ldots, \alpha_N \text{ with } \sum_{n=1}^N \alpha_n = 1$$

To solve this constrained optimization problem, we can use the Lagrangian, setting

$$L(\alpha_1, \ldots, \alpha_N; \lambda) := \sigma^2 \sum_{n=1}^N \alpha_n^2 - \lambda\left[\sum_{n=1}^N \alpha_n - 1\right]$$

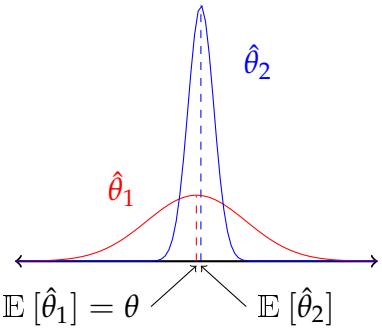


Figure 4.4: Biased and unbiased estimators

where λ is the Lagrange multiplier. Differentiating with respect to α_n and setting the result equal to zero, the minimizer α_n^* satisfies

$$\alpha_n^* = \lambda / (2\sigma^2) \qquad n = 1, \ldots, N$$

In particular, each α_n^* takes the same value, and hence, from the constraint $\sum_n \alpha_n^* = 1$, we have $\alpha_n^* = 1/N$. Using these values, our estimator becomes

$$\sum_{n=1}^N \alpha_n^* x_n = \sum_{n=1}^N (1/N) x_n = \bar{x}$$

We conclude that, under our assumptions, the sample mean is the best linear unbiased estimator of μ.

Returning to the general case, note that while classical statistics puts much emphasis on unbiased estimators, in recent years the use of biased estimators has become very common. To understand why, it's important to bear in mind that what we seek is an estimator of θ that is close to θ with high probability, and far away with low probability. In this sense, an unbiased estimator is not necessarily better than a biased one. For example, consider the two estimators $\hat{\theta}_1$ and $\hat{\theta}_2$ of θ depicted figure 4.4. The unbiased estimator $\hat{\theta}_1$ has higher variance than the biased estimator $\hat{\theta}_2$, and it is not clear that its performance will be better.

An overall measure of performance that takes into account both bias and variance is mean squared error, as defined in (4.9). Indeed, we can (exercise 4.7.1) decompose mean squared error into the sum of variance and squared bias:

$$\text{mse}[\hat{\theta}] = \text{var}[\hat{\theta}] + \text{bias}[\hat{\theta}]^2 \tag{4.11}$$

In many situations we find a trade-off between bias and variance: We can lower variance at the cost of extra bias and vice-versa.

4.2.3 Asymptotic Properties

Notice in (4.10) how, under the IID assumption, the variance of the sample mean converges to zero in N. Since the sample mean is unbiased for the mean, this suggests that in the limit, all probability mass concentrates on the mean—which is the value that the sample mean seeks to estimate. This is a useful piece of information. In order to formalize it, as well as generalize to other estimators, let's now consider asymptotic properties.

Let $\hat{\theta}_N$ be an estimator of a quantity $\theta \in \mathbb{R}$, where N denotes the size of the sample from which $\hat{\theta}_N$ is constructed. We say that $\hat{\theta}_N$ is

- **asymptotically unbiased** for θ if $\mathbb{E}[\hat{\theta}_N] \to \theta$ as $N \to \infty$
- **consistent** for θ if $\hat{\theta}_N \xrightarrow{p} \theta$ as $N \to \infty$
- **asymptotically normal** if $\sqrt{N}(\hat{\theta}_N - \theta) \xrightarrow{d} \mathcal{N}(0, v(\theta))$ for some $v(\theta) > 0$.

In the last definition, $v(\theta)$ is called the **asymptotic variance** of $\hat{\theta}_N$.

All of these properties are desirable. The desirability of asymptotic normality is somewhat less obvious than that of consistency, but asymptotic normality is important for two reasons. First, it provides a means of forming confidence intervals and hypothesis tests, as explained in chapter 5. Second, it gives us an idea of the *rate* of convergence of $\hat{\theta}_N$ to θ. To see this, observe that if $\hat{\theta}_N$ is asymptotically normal, then $\sqrt{N}(\hat{\theta}_N - \theta)$ does not diverge to infinity. This means that the term $\hat{\theta}_N - \theta$ goes to zero at least fast enough to offset the diverging term $\sqrt{N}$. To emphasize this point, we sometimes say that an asymptotically normal estimator $\hat{\theta}_N$ is **$\sqrt{N}$-consistent**.

Example 4.2.5. The sample mean $\bar{x}_N$ of any identically distributed sample is asymptotically unbiased for the common mean μ because it is unbiased. If the random variables $x_1, \ldots, x_N$ in the sample are also independent, then we can apply the law of large numbers, which implies that $\bar{x}_N$ is consistent for μ (see (1.26) on page 34). If, in addition, $\mathbb{E}[x_n^2] < \infty$, then we can also apply the central limit theorem, which implies that $\bar{x}_N$ is asymptotically normal (see (1.29) on page 36).

Example 4.2.6. As another example of consistency, let's consider the sample standard deviation $s_N = s$ defined in (4.2). Let $x_1, \ldots, x_N$ be an IID sample as before, with each x_n having mean μ, variance σ^2 and standard deviation $\sigma = \sqrt{\sigma^2}$. Fact 1.4.1 on page 31 tells us that if g is continuous and $s_N^2 \xrightarrow{p} \sigma^2$, then $g(s_N^2) \xrightarrow{p} g(\sigma^2)$. Taking

$g(x) = \sqrt{x}$, we see that if $s_N^2 \xrightarrow{p} \sigma^2$, then $s_N = \sqrt{s_N^2} \xrightarrow{p} \sqrt{\sigma^2} = \sigma$, which is what we want to show. Hence it suffices to prove that $s_N^2 \xrightarrow{p} \sigma^2$. To see that this is the case, note that

$$s_N^2 = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \bar{x}_N)^2 = \frac{N}{N-1}\,\frac{1}{N}\,\sum_{n=1}^{N}(x_n - \bar{x}_N)^2$$
$$= \frac{N}{N-1}\,\frac{1}{N}\,\sum_{n=1}^{N}[(x_n - \mu) - (\bar{x}_N - \mu)]^2$$

Expanding out the square, we get

$$s_N^2 = \frac{N}{N-1}\left[\frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2 - 2\frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)(\bar{x}_N - \mu) + (\bar{x}_N - \mu)^2\right]$$
$$= \frac{N}{N-1}\left[\frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2 - (\bar{x}_N - \mu)^2\right]$$

By the law of large numbers,

$$\frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2 \xrightarrow{p} \sigma^2 \quad \text{and} \quad (\mu - \bar{x}_N) \xrightarrow{p} 0$$

Applying the various results in fact 1.4.1 (page 31), we then have

$$s_N^2 = \frac{N}{N-1}\left[\frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2 - (\mu - \bar{x}_N)^2\right] \xrightarrow{p} 1 \times [\sigma^2 - 0] = \sigma^2$$

Hence the sample variance and sample standard deviation are consistent estimators of the variance and standard deviation respectively.

4.3 Maximum Likelihood

How does one come up with an estimator having nice properties, such as unbiasedness, consistency, etc.? Sometimes we can just use intuition. For example, it's natural to use the sample mean to estimate the mean of a random variable. In more complicated settings, however, more systematic approaches are required. One such approach is the celebrated **principle of maximum likelihood**.

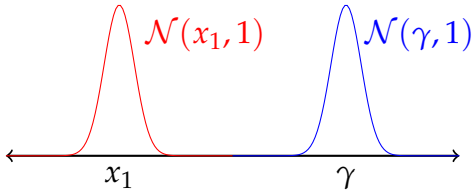


Figure 4.5: Maximizing the likelihood

4.3.1 The Idea

To motivate the methodology, suppose I present you with a single draw x_1 from distribution $\mathcal{N}(\mu, 1)$, where the μ is unknown, and the standard deviation is known and equal to one for simplicity. Your job is to make a guess $\hat{\mu}$ of the mean μ of the distribution, given the observation x_1. Since a guess $\hat{\mu}$ of μ also pins down the distribution $\mathcal{N}(\hat{\mu}, 1)$, we could equivalently say that your job is to guess the distribution that generated the observation x_1.

In guessing this distribution, if we centered it around some number γ much larger than x_1, then our observed data point x_1 would be an "unlikely" outcome for this distribution. See figure 4.5. The same logic would apply if we centered the density at a point much smaller than x_1.[3] In fact, in the absence of any additional information, the most obvious guess would be that the normal density is centered on x_1. To center the density on x_1, we must choose the mean to be x_1. In other words, our guess of μ is $\hat{\mu} = x_1$.

Maximum likelihood leads to the same conclusion. The density of x_1 is

$$p(s; \mu) := (2\pi)^{-1/2} \exp\left\{-\frac{(s-\mu)^2}{2}\right\} \qquad (s \in \mathbb{R})$$

Consider plugging the observed value x_1 into this density:

$$p(x_1; \mu) = (2\pi)^{-1/2} \exp\left\{-\frac{(x_1-\mu)^2}{2}\right\}$$

Even though were dealing with a continuous random variable here, let's think of $p(x_1; \mu)$ as representing the "probability" of realizing our sample point x_1. The principle of maximum likelihood suggests that we take as our guess $\hat{\mu}$ of μ the value that

[3] Given that our distribution can be represented by a density, all individual outcomes $s \in \mathbb{R}$ have probability zero, and in this sense, all outcomes are equally unlikely. What's meant by the statement about x_1 being relatively "unlikely" is that there is little probability mass in the neighborhood of x_1.

maximizes this probability. In some sense, this is the "most likely" μ given the sample. A little thought will convince you that $\hat{\mu} = x_1$ is the maximizer. That is,

$$\hat{\mu} = x_1 = \operatorname*{argmax}_{-\infty<\mu<\infty} p(x_1; \mu)$$

This coincides with our intuitive discussion regarding figure 4.5.

The same principle applies when we have $x_1, \ldots, x_N \overset{\text{IID}}{\sim} \mathcal{N}(\mu, 1)$, where μ is unknown. By independence, the joint density of this sample is obtained as the product of the marginal densities. Plugging the sample values into the joint density, one then maximizes the joint density with respect to μ. The maximizer

$$\hat{\mu} := \operatorname*{argmax}_{-\infty<\mu<\infty} (2\pi)^{-N/2} \prod_{n=1}^{N} \exp\left\{-\frac{(x_n - \mu)^2}{2}\right\} \tag{4.12}$$

is called the maximum likelihood estimate. As you are asked to show in exercise 4.7.4, the maximizer $\hat{\mu}$ is precisely the sample mean of $x_1, \ldots, x_N$.

We can generalize these ideas in several ways. Let's suppose now that the data $x_1, \ldots, x_N$ has *joint* density p in the sense of (1.19). We will assume that $p = p(\cdot; \theta)$ is known up to a vector of parameters $\theta \in \Theta \subset \mathbb{R}^K$. In other words, the functional form of p is known, and each choice of θ pins down a particular density $p = p(\cdot; \theta)$, but the value of θ in the density $p(\cdot; \theta)$ that generated the data is unknown. In this setting, the **likelihood function** is p evaluated at the sample $x_1, \ldots, x_N$, and regarded as a function of θ:

$$L(\theta) := p(x_1, \ldots, x_N; \theta) \qquad (\theta \in \Theta) \tag{4.13}$$

The principle of maximum likelihood tells us to estimate θ using the maximizer of $L(\theta)$ over $\theta \in \Theta$. Alternatively, we can maximize the **log likelihood function** (see §13.2), defined as the log of L:

$$\ell(\theta) := \ln(L(\theta)) \qquad (\theta \in \Theta)$$

The **maximum likelihood estimate** (MLE) $\hat{\theta}$ is the maximizer of $L(\theta)$, or, equivalently, of $\ell(\theta)$:

$$\hat{\theta} := \operatorname*{argmax}_{\theta\in\Theta} L(\theta) = \operatorname*{argmax}_{\theta\in\Theta} \ell(\theta) \tag{4.14}$$

(In the preceding discussion, p was a density function, but it can be a probability mass function as well. Exercise 4.7.6 treats one example.)

To implement this method, we obviously need to know the joint density p of the data. As we saw in (4.12), when the data points are independent this is easy because the joint density p is the product of the marginals. More generally, if each x_n is drawn independently from fixed arbitrary (marginal) density $p_n(\cdot;\theta)$ on $\mathbb{R}$, then

$$L(\theta) = \prod_{n=1}^{N} p_n(x_n;\theta) \quad \text{and} \quad \ell(\theta) = \sum_{n=1}^{N} \ln p_n(x_n;\theta) \tag{4.15}$$

4.3.2 Conditional Maximum Likelihood

In many statistical problems we wish to learn about the relationship between one or more "input" variables and an "output" or "response" variable y. (See example 4.1.4 on page 109.) In the case of scalar input, we observe inputs $x_1, \ldots, x_N$ and corresponding outputs $y_1, \ldots, y_N$. Given this data, we wish to estimate the relationship between x and y. For the following theoretical discussion, we suppose that the pairs (x_n, y_n) are independent of each other and share a common density p:

$$\mathbb{P}\{x_n \leq \bar{s},\, y_n \leq \bar{t}\} = \int_{-\infty}^{\bar{t}} \int_{-\infty}^{\bar{s}} p(s,t)\, ds dt \quad \text{for all } \bar{s}, \bar{t} \in \mathbb{R}$$

Let's suppose that in investigating the relationship between x and y, we have decided that the conditional density of y given x has the form $f_\theta(y|x)$, where $\theta \in \Theta$ is a vector of parameters. How can we choose θ by maximum likelihood?

The principle of maximum likelihood tells us to maximize the log likelihood function formed from the joint density of the sample, which in this case is

$$\ell(\theta) = \sum_{n=1}^{N} \ln p(x_n, y_n)$$

Letting g be the marginal density of x, we can use the decomposition (1.21) on page 26 to write $p(s,t) = f_\theta(t \,|\, s) g(s)$, and re-express the log likelihood as

$$\ell(\theta) = \sum_{n=1}^{N} \ln[f_\theta(y_n|x_n) g(x_n)]$$

Since the function g enters into this expression, it might seem like we need to specify the marginal distribution of x in order to maximize ℓ. However, if g is not a function of θ then this is unnecessary, since

$$\sum_{n=1}^{N} \ln[f_\theta(y_n|x_n) g(x_n)] = \sum_{n=1}^{N} \ln f_\theta(y_n|x_n) + \sum_{n=1}^{N} \ln g(x_n)$$

By assumption, the second term is independent of θ, and as such it does not affect the maximizer. As a result, the MLE is

$$\operatorname*{argmax}_{\theta \in \Theta} \sum_{n=1}^{N} \ln f_\theta(y_n | x_n)$$

The term $\sum_{n=1}^{N} \ln f_\theta(y_n|x_n)$ is formally referred to as the **conditional log likelihood**, but in applications most people just call it the log likelihood.

Example 4.3.1. Consider a binary response model with scalar input x. To be concrete, we will imagine that binary (i.e., Bernoulli) random variables $y_1, \ldots, y_N$ indicate whether or not a sample of married women participate in the labor force. We believe that the decision y_n of the n-th individual is influenced by a variable x_n measuring income from the rest of the household (e.g., the salary of their spouse). Let $q(s)$ be the probability that $y = 1$ (indicates participation) when $x = s$. Often this is modelled by taking $q(s) = F(\theta s)$, where θ is an unknown parameter and F is a cdf. We can then write

$$\mathbb{P}\{y = t \,|\, x = s\} = F(\theta s)^t (1 - F(\theta s))^{1-t} \quad \text{for } s \in \mathbb{R} \text{ and } t \in \{0, 1\}$$

Taking this expression as the conditional density of y given x, the (conditional) log likelihood is therefore

$$\begin{aligned} \ell(\theta) &= \sum_{n=1}^{N} \ln[F(\theta x_n)^{y_n} (1 - F(\theta x_n))^{1-y_n}] \\ &= \sum_{n=1}^{N} y_n \ln F(\theta x_n) + \sum_{n=1}^{N} (1 - y_n) \ln(1 - F(\theta x_n)) \end{aligned}$$

If F is the standard normal cdf, then the binary response model is called the **probit** model. If F is the logistic cdf $F(s) = 1/(1 - e^{-s})$, then it's called the **logit** model.[4]

4.3.3 Comments on Maximum Likelihood

Let's finish with some general comments on maximum likelihood. Maximum likelihood theory formed the cornerstone of early to mid 20th Century statistics. Analyzed by a series of brilliant statisticians, maximum likelihood estimators were

[4] To find the MLE, we can differentiate ℓ with respect to θ to obtain the first order condition, but there is no analytical solution for either the probit or logit case. Instead, numerical optimization is required. However, ℓ can be shown to be concave on $\mathbb{R}$, which means that most hill climbing algorithms will converge to the global maximum. Some discussion of numerical optimization is given in §8.3.3.

shown to be good estimators under a variety of different criteria. For example, under a bunch of regularity conditions that we won't delve into, MLEs are

1. consistent,
2. asymptotically normal, and
3. have small variance, at least asymptotically.

These results are genuinely remarkable. For details, see, for example, Dasgupta (2008, chapter 16).

In the last several decades, however, many statisticians have become increasingly dissatisfied with the limitations of maximum likelihood, and other approaches have become more popular. The most important criticism of maximum likelihood is that the statistician must bring a lot of knowledge to the table in order to form an estimator. If we look at (4.14), we see that to determine the MLE we must first specify the likelihood function L, which in turn requires the joint distribution of the sample as a function of θ. Thus, to pin down the MLE we need to know the parametric class of the density from which the sample was drawn. All of the nice properties of the MLE mentioned above are entirely dependent correct specification of the joint distribution. This is a very big caveat indeed.

Discussion of these issues continues in §4.4.

4.4 Parametric vs Nonparametric Estimation

[roadmap]

4.4.1 Classes of Distributions

We used the terminology "parametric class" in the preceding discussion. Consider, for example, the set $\mathcal{D}$ of all normal densities. That is,

$$\mathcal{D} := \left\{ \text{all } p \text{ s.t. } p(s; \mu, \sigma) = (2\pi\sigma^2)^{-1/2} \exp\left\{ -\frac{(s-\mu)^2}{2\sigma^2} \right\}, \ \mu \in \mathbb{R}, \ \sigma > 0 \right\}$$

The set $\mathcal{D}$ is an example of a parametric class. In this case the parametric class is the set of all normal densities p that can be formed by different choices of μ and σ.

The parameters are μ and σ, and a particular choice of the parameters determines (parameterizes) an element of the class $\mathcal{D}$.

More generally, a **parametric class** of densities

$$\mathcal{D} := \{p_\theta\}_{\theta \in \Theta} :=: \{p_\theta : \theta \in \Theta\}$$

is a set of densities p_θ indexed by a vector of parameters $\theta \in \Theta \subset \mathbb{R}^K$. In the previous example, $\theta = (\mu, \sigma)$, and $\Theta \subset \mathbb{R}^2$. Not all classes of densities are parametric, however. For example, consider the set $\mathcal{D}'$ of all densities p with finite second moment. In other words,

$$\mathcal{D}' := \left\{ \text{all } p \colon \mathbb{R} \to \mathbb{R} \text{ s.t. } p \geq 0, \int p(s)ds = 1, \int s^2 p(s)ds < \infty \right\}$$

This is a large set of densities that cannot be expressed as a parametric class. In such cases, we say that the class of densities is **nonparametric**.

Classical methods of inference such as maximum likelihood are parametric in nature. In this setting, we typically assume that:

- The data is generated by an unknown density.
- The density belongs to parametric class $\mathcal{D} = \{p_\theta\}_{\theta \in \Theta}$.
- We know the class, but $\theta \in \Theta$ is unknown.

Nonparametric statistical techniques assume instead that the unknown density is in some nonparametric class of densities. Often this class will be very broad. It may even be the set of all densities.

4.4.2 Parametric Estimation

To understand the difference between parametric and nonparametric estimation, let's look at an extended example. In the example, we let f be a density that is a mixture of two normals. In particular,

$$f(s) := \frac{1}{2}(2\pi)^{-1/2} \exp\left\{-\frac{(s+1)^2}{2}\right\} + \frac{1}{2}(2\pi)^{-1/2} \exp\left\{-\frac{(s-1)^2}{2}\right\} \tag{4.16}$$

To understand the density f, suppose that we flip a fair coin. If we get heads then we draw x from $\mathcal{N}(-1, 1)$. If we get tails then we draw x from $\mathcal{N}(1, 1)$. The random variable x then has density f. A plot of f is given in figure 4.6.

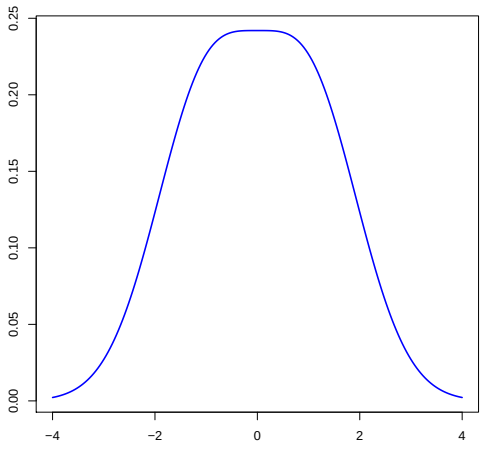


Figure 4.6: The true density f

Now consider an econometrician who does not know f, but has instead access to an IID sample $x_1, \ldots, x_N$ from f. His (or her) objective is to estimate the whole density f, based on the sample. Assuming that he is trained in traditional econometrics/statistics, his instinct will be to choose a particular parametric class for f, and then estimate the parameters in order to pin down a particular density in that class.

Let's suppose that, not having any particular pointers from theory as to the nature of the distribution, our econometrician decides in the end to model the density as being normally distributed. In other words, he makes the assumption that $f \in \mathcal{D}$, where $\mathcal{D}$ is the class of normal densities, as defined above.

He now wants to form an estimator $\hat{f} \in \mathcal{D}$ of f, based on the data. This involves determining two parameters, μ and σ. He does this in the obvious way: He estimates μ via $\hat{\mu} := \bar{x}$, the sample mean, and σ via $\hat{\sigma} := s$, the sample standard deviation. Plugging these into the density, his estimate becomes $\hat{f}(s) := p(s; \hat{\mu}, \hat{\sigma})$, where p is the normal density.

Let's generate 200 independent observations $x_1, \ldots, x_{200}$ from f and see how this procedure goes. The estimator $\hat{f}$ of f proposed by our econometrician is the black density in figure 4.7. The density is superimposed over the histogram and the original true density f (in blue).

Let's now consider whether $\hat{f}$ a is good estimator of f. On balance, one would probably have to say no. Even though the fit in figure 4.7 (i.e., the deviation between

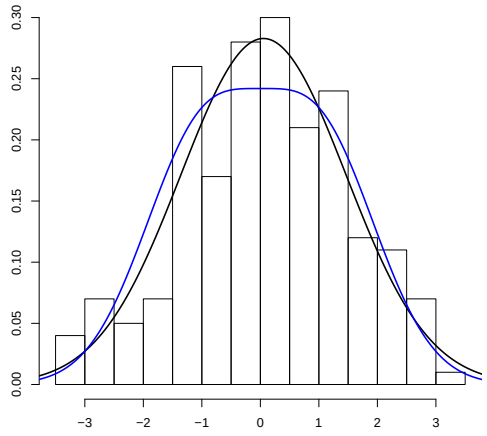


Figure 4.7: Estimate $\hat{f}$ (black line) and true f (blue line)

the black and blue lines) might be thought of as reasonable, the estimator $\hat{f}$ does not have good properties. In particular, $\hat{f}$ will not converge to f as the sample size goes to infinity.[5] The problem is the mistaken original assumption that $f \in \mathcal{D}$. *There is no element of $\mathcal{D}$ that can approximate f well.* We can see this by rerunning the simulation with ten times more observations (2,000 instead of 200). The result is given in figure 4.8. As expected, the fit is not much better.

4.4.3 Nonparametric Kernel Density Estimation

Let's now look at a standard nonparametric approach to the same problem. Our next econometrician is in the same position as the econometrician in the previous section: IID data $x_1, \ldots, x_N$ is generated from the density f in (4.16) and presented to her. She does not have knowledge of f, and seeks to construct a estimate $\hat{f}$ of f on the basis of the data. Unlike the previous econometrician, however, she does not presume to know the parametric class that the density f belongs to. How can she proceed?

[5] In general, one would say that $\hat{f}$ is not *consistent*. I haven't used this terminology, because consistency was defined for real-valued estimators, not *function-valued* estimators like $\hat{f}$. However, one can define a notion of convergence in probability for functions, and then give a definition of consistency that applies here. See any advanced text on density estimation.

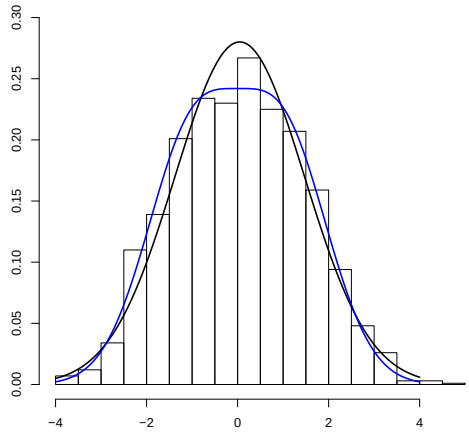


Figure 4.8: Sample size = 2,000

One approach to estimating f without making assumptions about the parametric class is to use a **kernel density estimator**. Let $K\colon \mathbb{R} \to \mathbb{R}$ be a density function, and let δ be a positive real number. From the sample, we then define

$$\hat{f}_N(s) :=: \hat{f}(s) := \frac{1}{N\delta} \sum_{n=1}^{N} K\left(\frac{s - x_n}{\delta}\right) \tag{4.17}$$

Here K is called the **kernel function** of the estimator, and δ is called the **bandwidth**. Exercise 4.7.7 asks you to confirm that $\hat{f}$ is indeed a density.[6]

A common choice for K is the standard normal density. For this choice of K, the function $\hat{f}$ in (4.17) is illustrated in figure 4.9 for the case of two sample points, x_1 and x_2. Centered on sample point x_n we place a smooth "bump" drawn in red, which is the function

$$g_n(s) := \frac{1}{N\delta} K\left(\frac{s - x_n}{\delta}\right) \qquad (n = 1, 2)$$

Summing these two bumps gives $\hat{f} = g_1 + g_2$, drawn in black.

In R, nonparametric kernel density estimates can be produced using the function `density`. Try, for example,

[6] Although we have not specified a parametric class, our choice of K and δ are associated with some assumptions about the shape and form of f. For example, if K is taken to be Gaussian, then $\hat{f}_N$ will have exponentially decreasing tails.

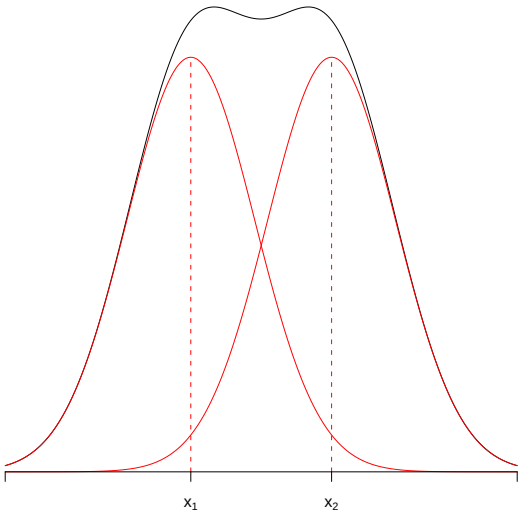


Figure 4.9: Function $\hat{f}$ when $N = 2$

```
> plot(density(runif(200)))
```

To learn more, type `?density`.

Returning to our estimation problem, figure 4.10 shows a kernel density estimate of f from 200 sample points, using the default settings in R. Figure 4.11 shows the estimate with 2,000 sample points. (The code for generating figure 4.11 is given in listing 4.) The fit in figure 4.11 is much better than the comparable parametric fit in figure 4.8, and, unlike the parametric case, further increases in sample size continue to improve the fit. Indeed, it can be proved in that, for any density f, the nonparametric kernel density estimator converges to f in the following sense:

Theorem 4.4.1. *Let f and K be any densities on $\mathbb{R}$, and let $\{x_n\}_{n=1}^{\infty} \overset{\text{IID}}{\sim} f$. Let $\hat{f}_N$ be as defined in (4.17). If the bandwidth δ_N is a sequence depending on N and satisfying $\delta_N \to 0$ and $N\delta_N \to \infty$ as $N \to \infty$, then*

$$\mathbb{E}\left[\int |\hat{f}_N(s) - f(s)|\, ds\right] \to 0 \qquad \text{as } N \to \infty$$

For a proof, see Devroye and Lugosi (2001, theorem 9.2).

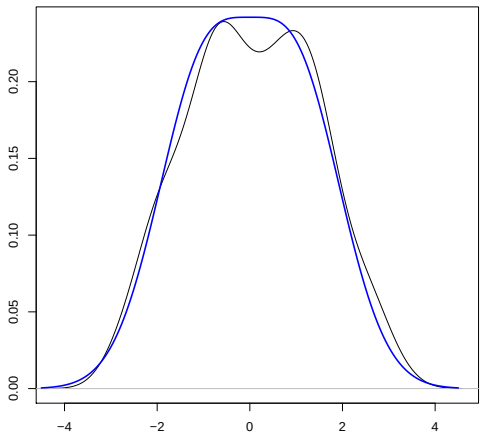


Figure 4.10: Nonparametric, sample size = 200

On the other hand, if you experiment with the code in listing 4, you will see that, for small sample sizes (try 10), the nonparametric estimate is actually poorer than the parametric alternative. The good theoretical results we have discussed are all asymptotic, and nonparametric methods generally need large sample sizes. This stands to reason: Nonparametric methods have little structure in the form of prior knowledge, and hence require abundant data.

4.4.4 Commentary

In some fields of science, researchers have considerable knowledge about parametric classes and specific functional forms. For example, the theory of Brownian motion describes how the location of a tiny particle in liquid is approximately normally distributed. Hence, the underlying theory provides the exact parametric class of the density.

Here's another example, from a regression perspective: An engineer is interested in studying the effect of a certain load on the length of a spring. Classical physics tells her that the relationship is approximately proportional. This provides a functional form that the engineer can use to estimate a regression function.

When underlying theory provides us with knowledge of functional forms, as in the two examples above, the parametric paradigm is excellent. Classical statistics

Listing 4 The source code for figure 4.11

```
set.seed(1234)


fden <- function(x) {  # The density function of f
   return(0.5 * dnorm(x, mean=-1) + 0.5 * dnorm(x, mean=1))
}


fsamp <- function(N) { # Generates N draws from f
   observations <- numeric(N)
   u <- runif(N)
   for (i in 1:N) {
       if (u[i] < 0.5) {
          observations[i] <- rnorm(1, mean=-1)
       }
       else observations[i] <- rnorm(1, mean=1)
   }
   return(observations)
}

observations <- fsamp(2000)
xgrid <- seq(-4.5, 4.5, length=200)
plot(density(observations), main="", xlab="", ylab="")
lines(xgrid, fden(xgrid), col="blue", lwd=2)
```

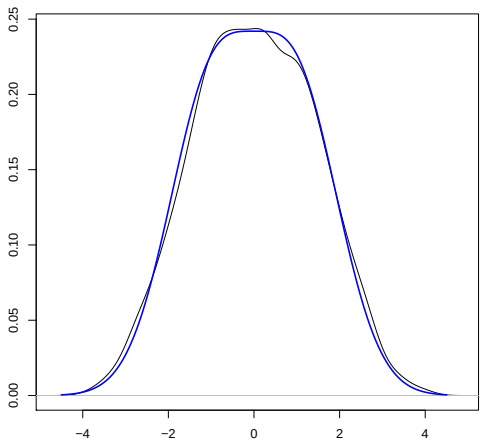


Figure 4.11: Nonparametric, sample size = 2,000

yields many well behaved estimators. Moreover, all the knowledge embodied in the functional forms helps us to conduct inference: Generalization is about combining knowledge and data, and the more knowledge we have the better.

What about econometrics?

Unfortunately, no such elegant equivalent of the theory of Brownian motion exists for the movement of economic variables such as stock prices, or exchange rates. Similarly, an econometrician trying to estimate the effect of human capital on GDP does not have the same information as the engineer studying load on a spring, who knows that her relationship is proportional. What does classical economics tell the econometrician about the precise functional form relating human capital to GDP? Not a lot, to be honest.

Overall, economics and other social sciences are generally messier than the physical sciences, and the econometrician usually comes to the table with much less knowledge of parametric classes and functional forms. This suggests that nonparametric techniques will become increasingly popular in econometrics. At the same time, there is no firm dividing line between parametric and nonparametric methods. Often, a flexible parametric class of densities with many parameters can do a great job of approximating a given nonparametric class, and be more convenient to work with. For this reason, modern statistics is something of a mix, blending both parametric and nonparametric methods. These notes follow this mixed approach.

4.5 Empirical Distributions

Recall from (1.12) on page 18 that if x is a discrete random variable taking values $s_1, \ldots, s_J$ with probabilities $p_1, \ldots, p_J$, then the cdf of x is

$$F(s) = \mathbb{P}\{x \leq s\} = \sum_{j=1}^{J} \mathbb{1}\{s_j \leq s\} p_j \tag{4.18}$$

In this section, we're going to look at an important kind of discrete distribution. To describe it, let $x_1, \ldots, x_N$ be draws from some unknown distribution F. The **empirical distribution** of the sample is the discrete distribution that puts equal probability on each sample point. Since there are N sample points, that means that probability $1/N$ is placed on each point x_n. The concept of the empirical distribution is a bit slippery because it's a *random* distribution, depending on the sample. Nevertheless, it's a very useful object.

Throughout this section, we'll work with the same fixed observations $x_1, \ldots, x_N \sim F$, and x^e will denote a random variable with the corresponding empirical distribution. That is, x^e is a random variable taking each of the values $x_1, \ldots, x_N$ with uniform probability $1/N$.

The cdf for the empirical distribution is called the **empirical cumulative distribution function**, or **ecdf**. Throughout the text, we will denote it by F_N. Invoking (4.18), we see that the ecdf can be written as

$$F_N(s) = \mathbb{P}\{x^e \leq s\} = \sum_{n=1}^{N} \mathbb{1}\{x_n \leq s\} \frac{1}{N}$$

It's more common to put the $1/N$ term at the start, so let's use this as our definition:

$$F_N(s) := \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}\{x_n \leq s\} \qquad (s \in \mathbb{R})$$

If you think about it, you will see that we can also write

$$F_N(s) = \text{ fraction of the sample less than or equal to } s$$

Graphically, F_N is a step function, with an upward jump of $1/N$ at each data point. Figure 4.12 shows an example with $N = 10$ and each data point drawn independently from a Beta(5, 5) distribution.

In R, the ecdf is implemented by the function `ecdf`. Try this example:

```
plot(ecdf(rnorm(20)))
```

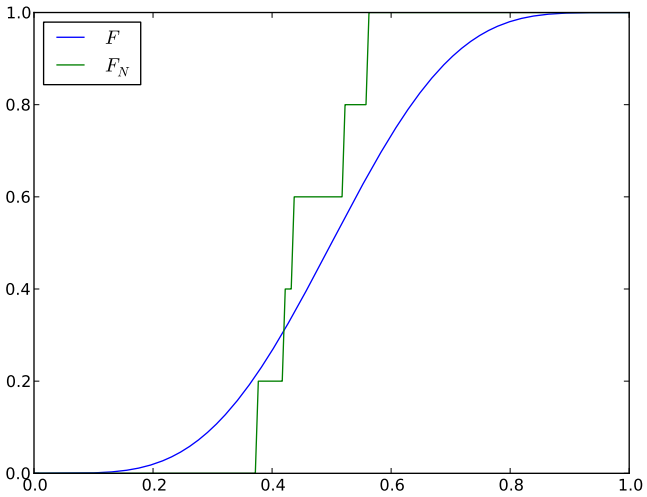


Figure 4.12: F_N and F with $N = 10$

4.5.1 Plug in Estimators

Continuing the proceeding discussion with the same notation, note that if h is a function from $\mathbb{R}$ into $\mathbb{R}$, then by (1.16) on page 22, its expectation with respect to the empirical distribution is

$$\int h(s) F_N(ds) := \mathbb{E}\left[h(x^e)\right] = \sum_{n=1}^{N} h(x_n) \frac{1}{N} = \frac{1}{N} \sum_{n=1}^{N} h(x_n) \tag{4.19}$$

For example, the mean of the empirical distribution is the sample mean of $x_1, \ldots, x_N$:

$$\int s F_N(ds) := \mathbb{E}\left[x^e\right] = \frac{1}{N} \sum_{n=1}^{N} x_n =: \bar{x}_N$$

If the sample is IID, then by the law of large numbers, the value of the expression (4.19) converges in probability to the expectation $\mathbb{E}\left[h(x_1)\right] = \int h(s) F(ds)$. In other words,

$$\text{for } h\colon \mathbb{R} \to \mathbb{R} \text{ and large } N \text{ we have } \quad \int h(s) F_N(ds) \approx \int h(s) F(ds)$$

This suggests an approach for producing estimators: Whenever we want to estimate a quantity θ such that

1. $\theta = \int h(s)F(ds)$ for some $h\colon \mathbb{R} \to \mathbb{R}$, and
2. the function h is known,

then we replace the cdf F with the ecdf F_N and use the resulting statistic

$$\hat{\theta}_N := \int h(s)F_N(ds) = \frac{1}{N}\sum_{n=1}^{N} h(x_n)$$

The estimator $\hat{\theta}_N$ is called the **plug in estimator** of $\theta = \int h(s)F(ds)$, because F_N is plugged in to the expression in place of F. Notice that plug in estimators are nonparametric, in the sense that we need no parametric class assumption in order to form the estimator.

Example 4.5.1. The plug in estimator of the k-th moment $\int s^k F(ds)$ of F is the sample k-th moment

$$\int s^k F_N(ds) = \frac{1}{N}\sum_{n=1}^{N} x_n^k$$

Example 4.5.2. The plug in estimator of the variance

$$\int \left[t - \int sF(ds)\right]^2 F(dt)$$

is

$$\int \left[t - \int sF_N(ds)\right]^2 F_N(dt) = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x}_N)^2$$

This differs slightly from the sample variance s_N^2 defined on page 112. However, the deviation is negligible when N is large.

Remark 4.5.1. Although we have defined the plug in estimator as an estimator of quantities θ that can be expressed as integrals using F, the term "plug in estimator" is often used more generally for any estimator produced by replacing F with F_N. For example, in this terminology, the plug in estimator of the median $F^{-1}(0.5)$ is $F_N^{-1}(0.5)$.

4.5.2 Properties of the ecdf

Once again, let $x_1, \ldots, x_N$ be IID draws from some fixed underlying distribution F, and let F_N be the corresponding ecdf. Perhaps the most important single fact about

the ecdf F_N is that it converges to the cdf F as $N \to \infty$. Indeed, from (1.28) on page 36, we have

$$F_N(s) := \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}\{x_n \leq s\} \xrightarrow{p} \mathbb{P}\{x_n \leq s\} =: F(s)$$

In fact, a stronger statement is true. The following theorem is sometimes called the fundamental theorem of statistics, or the Glivenko-Cantelli theorem:

Theorem 4.5.1 (FTS). *If $x_1, \ldots, x_N$ are* IID *draws from some cdf F, and F_N is the corresponding ecdf, then*

$$\sup_{s \in \mathbb{R}} |F_N(s) - F(s)| \xrightarrow{p} 0$$

(Here "sup" is roughly equivalent to "max"—see the appendix for more discussion.) Thus, we see that the maximal deviation between the two functions goes to zero in probability.[7] Figures 4.13–4.15 illustrate the idea. Each picture shows 10 observations of F_N, depending on 10 different observations of the data $x_1, \ldots, x_N$.

The theorem tells us that, at least in the IID setting, if we have an infinite amount of data, then we can learn the underlying distribution without having to impose any assumptions. This is certainly a nice result. However, we should bear in mind that in reality we only ever have a finite amount of data. As such, assumptions are still required to generalize from this data.

4.6 Empirical Risk Minimization

Empirical risk minimization is an inductive principle that is essentially nonparametric in nature. Except in special cases, it does not require specification of the parametric form of the underlying density in order to form the estimator. Instead, it starts with loss function, which states the subjective loss (opposite of utility) from incorrect prediction.

4.6.1 The ERM Principle

To understand the principle, consider a setting where we repeatedly observe an input x to a system, followed by an output y. Both are random variables, and we

[7] In fact, the theorem tells us that convergence occurs "almost surely," which is a stronger notion that in probability. The details are omitted.

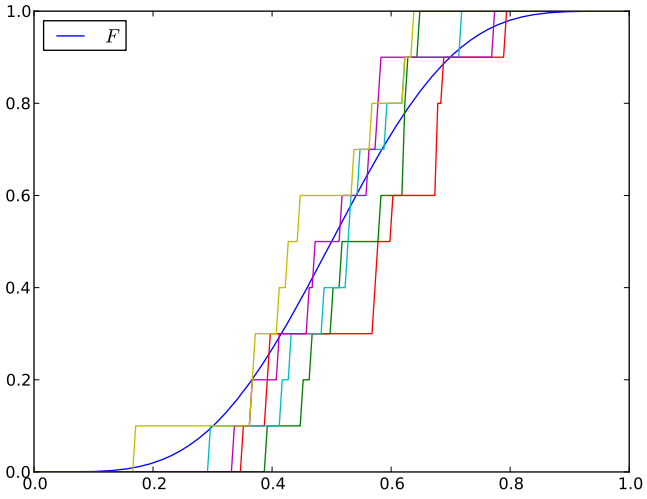


Figure 4.13: Realizations of F_N with $N = 10$

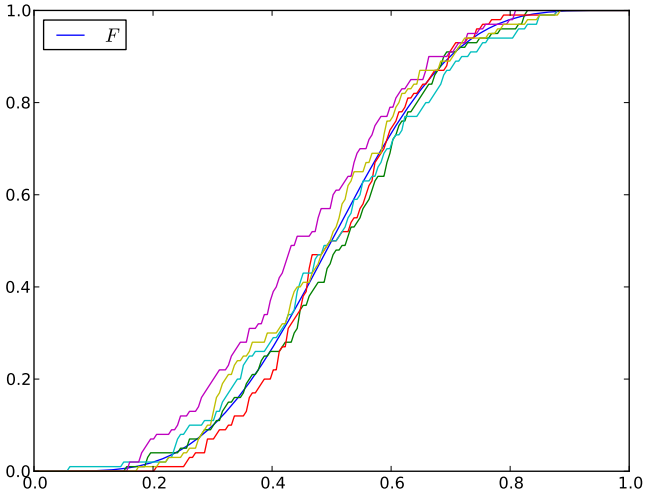


Figure 4.14: Realizations of F_N with $N = 100$

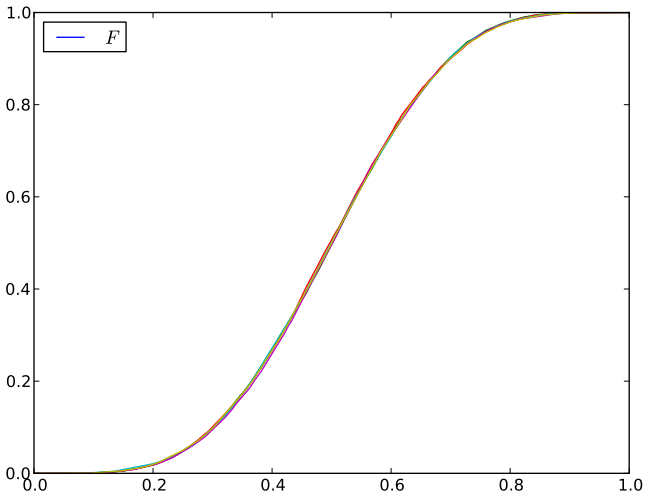


Figure 4.15: Realizations of F_N with $N = 1000$

believe there is some stable relation between them. In particular, we assume that the input-output pairs $(x_1, y_1), \ldots, (x_N, y_N)$ we observe are IID draws from joint density p. Suppose our aim is to predict new output values from observed input values. A natural way to treat this problem is to choose a function f such that $f(x)$ is our prediction of y once x is observed. Incorrect prediction incurs a loss. Letting y be the actual outcome, the size of this subjective loss is taken to be $L(y, f(x))$. The function L is called the **loss function**. Common choices for the loss function include:

- The quadratic loss function $L(y, f(x)) = (y - f(x))^2$
- The absolute deviation $L(y, f(x)) = |y - f(x)|$
- The discrete loss function $L(y, f(x)) = \mathbb{1}\{y \neq f(x)\}$

The discrete loss function is typically used when y takes only finitely many possible values. A unit loss is incurred if our guess is incorrect. No loss is incurred otherwise.

Returning to the general problem, with arbitrary loss function L, one might consider choosing f so as to minimize the expected loss

$$R(f) := \mathbb{E}\left[L(y, f(x))\right] :=: \int \int L(t, f(s)) p(s, t) ds dt \tag{4.20}$$

In (4.20), the expected loss is called the **risk**, and R is called the **risk function**. If we knew the joint density p, then, at least in principle, we could evaluate $R(f)$ for any f by calculating the double integral in (4.20). By repeating this calculation for different f, we could search for a minimizer.

However, in the statistical setting, there is an obvious difficultly: We don't know p. Hence $R(f)$ cannot be evaluated, let alone minimized.

All is not lost. While we don't know p, we do have at hand the observed input-output pairs $(x_1, y_1), \ldots, (x_N, y_N)$, which are independent draws from p. Draws from p give us information about p. For example, the law of large numbers tells us for large N we have

$$R(f) := \mathbb{E}\left[L(y, f(x))\right] \approx \frac{1}{N}\sum_{n=1}^{N} L(y_n, f(x_n))$$

This motivates us to replace the risk function R with the **empirical risk function**

$$\hat{R}(f) := \frac{1}{N}\sum_{n=1}^{N} L(y_n, f(x_n)) \tag{4.21}$$

In particular, we obtain an estimate $\hat{f}$ of the true minimizer $\operatorname{argmin}_f R(f)$ by solving

$$\hat{f} = \operatorname*{argmin}_{f\in\mathcal{F}} \hat{R}(f) = \operatorname*{argmin}_{f\in\mathcal{F}} \frac{1}{N}\sum_{n=1}^{N} L(y_n, f(x_n)) \tag{4.22}$$

This inductive principle, which produces an estimate of the risk-minimizing function by minimizing the empirical risk, is called **empirical risk minimization** (ERM).

Notice that in (4.22) we are minimizing over a set of functions $\mathcal{F}$. This set of functions is called the **hypothesis space**, and is a class of candidate functions chosen by the econometrician or researcher. At first pass, it might seem that we should $\mathcal{F}$ to be the set of *all* functions $f\colon \mathbb{R} \to \mathbb{R}$, or at least take it to be as large as possible. After all, if the risk minimizing function $f^* := \operatorname{argmin}_f R(f)$ is not in $\mathcal{F}$, as visualized in figure 4.16, then the solution to (4.22) is not equal to f^*, and we are making a sub-optimal choice.

Although this reasoning seems logical, it turns out that setting $\mathcal{F}$ to be the set of all functions from $\mathbb{R} \to \mathbb{R}$ is a bad idea. In fact, we want to be quite restrictive in our choice of $\mathcal{F}$. These ideas are explored in detail in §4.6.2.

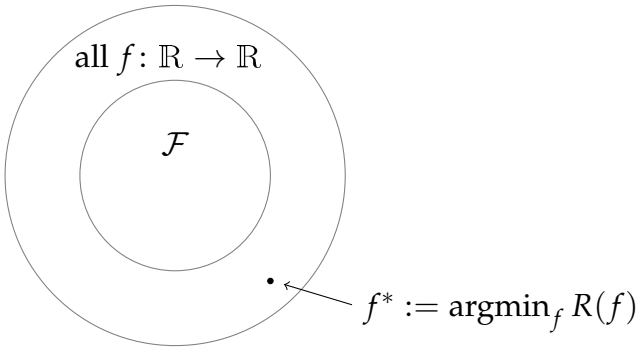


Figure 4.16: Choosing the hypothesis space

4.6.2 ERM and Least Squares

Specializing to the quadratic loss function $L(y, \hat{y}) = (y - \hat{y})^2$, and observing that the term $\frac{1}{N}$ makes no difference to the solution $\hat{f}$ (see §13.2), the ERM problem becomes

$$\min_{f \in \mathcal{F}} \sum_{n=1}^{N} (y_n - f(x_n))^2 \tag{4.23}$$

For obvious reasons, this optimization problem is called the **least squares problem**. If we specialize $\mathcal{F}$ to be the set of affine functions

$$\mathcal{L} := \{ \text{ all functions of the form } \ell(x) = \alpha + \beta x \} \tag{4.24}$$

then the problem reduces to the **linear least squares problem**

$$\min_{\ell \in \mathcal{L}} \sum_{n=1}^{N} (y_n - \ell(x_n))^2 = \min_{\alpha, \beta} \sum_{n=1}^{N} (y_n - \alpha - \beta x_n)^2 \tag{4.25}$$

This is the empirical risk counterpart to the risk minimization problem (1.23) on page 30. Direct differentiation and simple manipulations show that the minimizers of the empirical risk are

$$\hat{\beta} = \frac{\sum_{n=1}^{N} (x_n - \bar{x})(y_n - \bar{y})}{\sum_{n=1}^{N} (x_n - \bar{x})^2} \quad \text{and} \quad \hat{\alpha} = \frac{1}{N} \sum_{n=1}^{N} y_n - \hat{\beta} \frac{1}{N} \sum_{n=1}^{N} x_n \tag{4.26}$$

Comparing with (1.24) on page 30, which gives the minimizers of the risk, we see that in this case the minimizers of the empirical risk are the sample analogues of the minimizers of the risk.

Now let's return to the issue of hypothesis space mentioned above: Why would we want to minimize the empirical risk over a restricted hypothesis space such as $\mathcal{L}$, rather than the entire set of functions from $\mathbb{R}$ to $\mathbb{R}$? After all, minimizing the empirical risk over a bigger set of functions makes the empirical risk smaller.[8] Isn't that desirable?

The answer is: not necessarily. The reason is that, while the function $\hat{f}$ obtained by minimizing empirical risk over a large set of functions will make the *empirical* risk $\hat{R}(\hat{f})$ small, the actual risk $R(\hat{f})$ may not be. The underlying problem is that we are attempting to minimize expected loss on the basis of a sample mean, rather using the expectation from the actual distribution. We need to be careful about reading "too much" into this particular sample.

Let's illustrate this point by way of an example, where empirical risk is minimized over progressively larger hypothesis spaces. In the example, the model we will consider is one that generates input-output pairs via

$$x \sim U[-1,1] \quad \text{and then} \quad y = \cos(\pi x) + u \quad \text{where} \quad u \sim N(0,1) \tag{4.27}$$

where $U[-1,1]$ is the uniform distribution on the interval $[-1,1]$. Our hypothesis spaces for predicting y from x will be sets of polynomial functions. To fix notation, let $\mathcal{P}_d$ be the set of all polynomials of degree d. That is,

$$\mathcal{P}_d := \{ \text{ all functions } f_d(x) = c_0 x^0 + c_1 x^1 + \cdots c_d x^d \text{ where each } c_i \in \mathbb{R} \}$$

Clearly

$$\mathcal{P}_1 \subset \mathcal{P}_2 \subset \mathcal{P}_3 \subset \cdots$$

because if f is a polynomial of degree d, then f can be represented as a polynomial of degree $d+1$ just by setting the last coefficient c_{d+1} to zero:

$$\begin{aligned} \mathcal{P}_d \ni f_d(x) &= c_0 x^0 + c_1 x^1 + \cdots c_d x^d \\ &= c_0 x^0 + c_1 x^1 + \cdots c_d x^d + 0 x^{d+1} \in \mathcal{P}_{d+1} \end{aligned}$$

Also, the set of linear functions $\mathcal{L}$ defined in (4.24) is equal to $\mathcal{P}_1$.

If we seek to predict y from x using quadratic loss and the set $\mathcal{P}_d$ as our candidate functions, the risk minimization problem is

$$\min_{f \in \mathcal{P}_d} R(f) \quad \text{where} \quad R(f) = \mathbb{E}\left[(y - f(x))^2\right] \tag{4.28}$$

[8] Intuitively, if we expand the set of candidates, then we can find a smaller value. Formally, if A is any set, $g\colon A \to \mathbb{R}$, and $D \subset D' \subset A$, then $\inf_{a \in D'} g(a) \leq \inf_{a \in D} g(a)$ always holds.

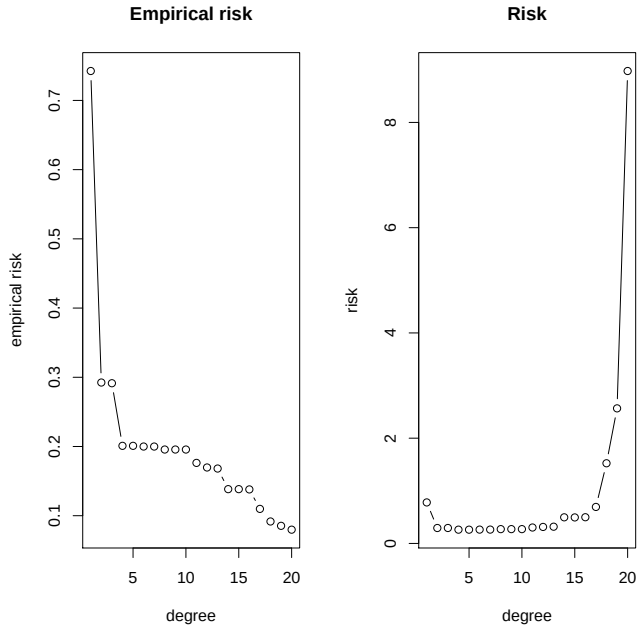


Figure 4.17: Risk and empirical risk as a function of d

while the empirical risk minimization problem is

$$\min_{f \in \mathcal{P}_d} \hat{R}(f) \quad \text{where} \quad \hat{R}(f) = \frac{1}{N} \sum_{n=1}^{N} (y_n - f(x_n))^2 \tag{4.29}$$

To illustrate the difference between risk and empirical risk, we first generate $N = 25$ data points from the model (4.27). Taking this as our data set, we then solve (4.29) repeatedly, once for each d in $1, 2, \ldots, 15$. The solution to the d-th minimization problem is denoted $\hat{f}_d$, and is, by construction, a polynomial of degree d. Finally, we compare the risk $R(\hat{f}_d)$ and empirical risk $\hat{R}(\hat{f}_d)$.[9] The results are in figure 4.17.

Analysing the figure, we see that, as expected, empirical risk falls monotonically with d. This must be the case because minimizing a function over larger and larger domains produces smaller and smaller values. On the other hand, the risk decreases slightly and then increases rapidly. For large d, the minimizer $\hat{f}_d$ of the empirical risk is associated with high risk in the sense of large expected loss.

[9]The risk $R(\hat{f}_d)$ is evaluated by substituting $\hat{f}_d$ into the expression for R in (4.28) and calculating the expectation numerically.

We can get a feeling for what is happening by plotting the data and the functions. In figures 4.18–4.21, the N data points are plotted alongside the function $y = \cos(\pi x)$ from the true model (4.27) in black, and fitted polynomial $\hat{f}_d$ in red. The function $y = \cos(\pi x)$ is the risk minimizer, and represents the ideal prediction function. In figure 4.18 we have $d = 1$, and the fitted polynomial $\hat{f}_1$ is the linear regression line. In figures 4.19, 4.20 and 4.21 we have $d = 3$, $d = 11$ and $d = 14$ respectively, and the fitted polynomials are $\hat{f}_3$, $\hat{f}_{11}$ and $\hat{f}_{14}$.

When $d = 1$, the hypothesis space $\mathcal{P}_d = \mathcal{P}_1$ is quite small. There is no function in this class that can do a good job of fitting the underlying model. This situation is called "under-fitting," and is reflected in the poor fit of the red line to the black line in figure 4.18. When $d = 3$, the class of functions $\mathcal{P}_d = \mathcal{P}_3$ is considerably larger. Given that the data is relatively noisy, and that we only have 25 observations, the fit of the function is fairly good (figure 4.19). If we look at the risk for $d = 3$ on the right-hand side of figure 4.17, we see that it is lower than for $d = 1$.

On the other hand, if we take d up to 11 or even 14, the fit to the underlying model is poor, and the risk is high. Examining the corresponding figures (figures 4.20 and 4.21), we see that the fitted polynomial has been able to fit the observed data closely, passing near to many of the data points. In a sense it is paying too much attention to this particular realization of the data. When a new input x is drawn, the prediction $\hat{f}_{14}(x)$ is likely to be a poor predictor of y, and the risk is correspondingly high. This situation is called "over-fitting."

What can we learn from this discussion? The main lesson is that the choice of the hypothesis space $\mathcal{F}$ in the empirical risk minimization problem (4.23) is crucial. This is the problem of **model selection**.

In real statistical applications, we do not have the luxury of knowing the true model when we choose $\mathcal{F}$. In response, many researchers simply choose $\mathcal{F} = \mathcal{L}$, the set of linear functions. This may or may not be a good choice. Ideally, the hypothesis space should be carefully chosen on the basis of economic theory: $\mathcal{F}$ should be the set of "reasonable" candidate descriptions of the relationship between x and y, given our knowledge of the economic system we are modelling. Once again, the message is that statistical learning equals prior knowledge plus data.

The problem of model selection is discussed in more depth in chapter 10.

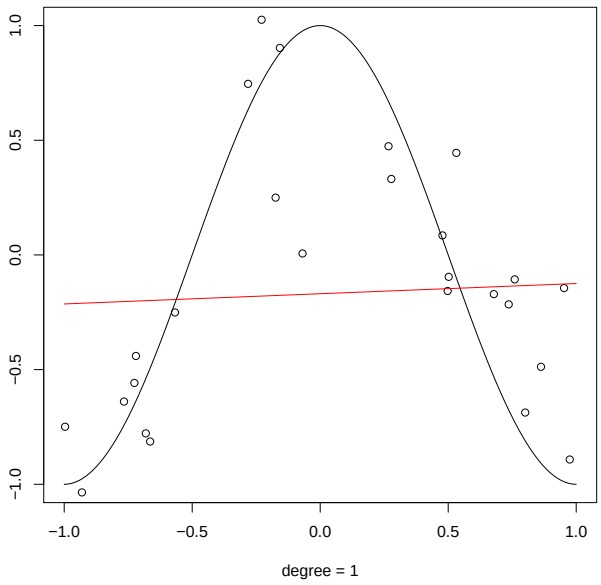


Figure 4.18: Fitted polynomial, $d = 1$

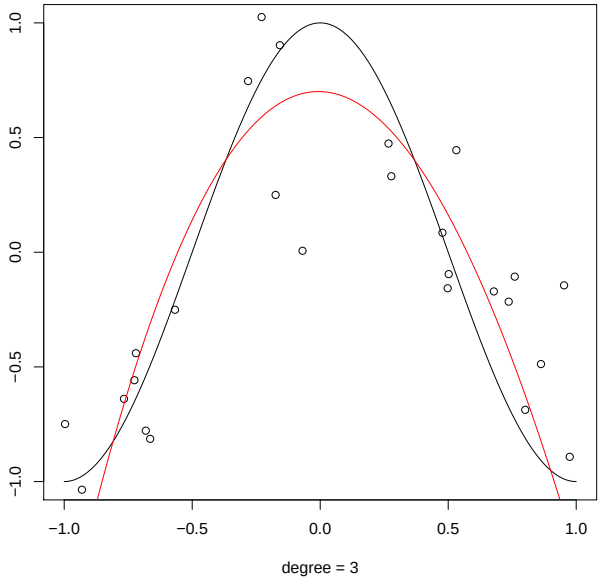


Figure 4.19: Fitted polynomial, $d = 3$

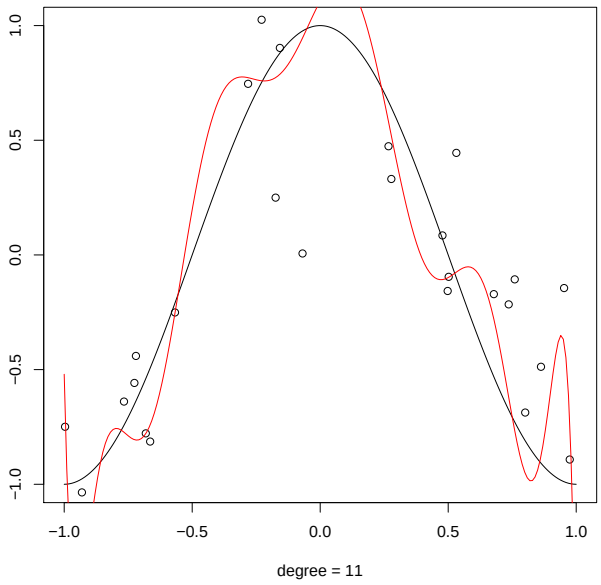


Figure 4.20: Fitted polynomial, $d = 11$

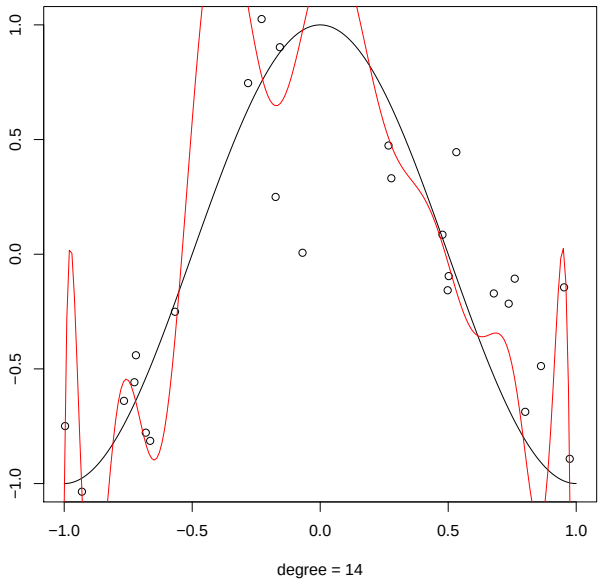


Figure 4.21: Fitted polynomial, $d = 14$

4.6.3 Other Applications of ERM

In our discussion of ERM so far, we have talked about finding functions to predict y given x. A simpler situation is where we observe only y and seek to predict it. In this case the object we seek to calculate is just a constant (a prediction of y) rather than a function (that predicts y from any given x). This makes the learning problem simpler.

Suppose, for example, that we observe $y_1, \ldots, y_N \overset{\text{IID}}{\sim} F$, where F is an unknown cdf. For the sake of concreteness, let's imagine that each observation is the monetary payoff of a particular game at a casino. We want to predict the payoff of the next draw. Letting α be our prediction and L be the loss function, the corresponding risk is $\mathbb{E}[L(\alpha, y)]$. If we specialize to the quadratic loss, this becomes

$$R(\alpha) = \mathbb{E}[(\alpha - y)^2] = \int (\alpha - s)^2 F(ds)$$

The empirical risk is

$$\hat{R}(\alpha) = \frac{1}{N} \sum_{n=1}^{N} (\alpha - y_n)^2 = \int (\alpha - s)^2 F_N(ds) \tag{4.30}$$

Here F_N is the empirical distribution of the sample. Minimizing $\hat{R}(\alpha)$ with respect to α, we obtain our prediction of y as

$$\alpha^* := \operatorname*{argmin}_{\alpha} \hat{R}(\alpha) = \frac{1}{N} \sum_{n=1}^{N} y_n$$

Thus, at least with quadratic loss, ERM leads to the sample mean, which is the most natural predictor of y.

As this last example helps to clarify, the ERM principle is essentially nonparametric in nature. The empirical risk is determined only by the loss function and the empirical distribution. Unlike maximum likelihood, say, we usually don't have to specify the parametric class of the unknown distributions in order to solve the ERM problem.

At the same time, we can recover many parametric techniques as special cases of empirical risk minimization. One is maximum likelihood. To see this, suppose that x is drawn from unknown density q. We wish to learn the density q by observing draws from this density. We take our loss function to be $L(p, x) = -\ln p(x)$. In other words, if our guess of q is p and the value x is drawn, then our loss is $-\ln p(x)$.

Loosely speaking, if p puts small probabilities on regions where x is realized, then we suffer large loss. Hence, the loss function encourages us to choose p close to q.

Our choice of loss leads to the risk function

$$R(p) = \mathbb{E}\left[L(p,x)\right] = -\int \ln[p(s)]q(s)ds$$

Although it may not be obvious at first glance, minimizing this risk function yields the unknown density q. To see this, let us first transform our expression for the risk function to

$$R(p) = \int \ln\left[\frac{q(s)}{p(s)}\right] q(s)ds - \int \ln[q(s)]q(s)ds$$

(Can you see why the two expressions for $R(p)$ are equal?) The term on the far right is called the entropy of the density q, and does not involve p. Hence, minimization of the risk comes down to minimization of

$$D(q,p) := \int \ln\left[\frac{q(s)}{p(s)}\right] q(s)ds$$

This quantity is called the **Kullback-Leibler (KL) deviation** between q and p. The KL deviation is possibly infinite, always nonnegative, and zero if and only if $p = q$.[10] It follows that the unique minimizer of the risk is the true density q.

Now suppose that we observe IID draws $x_1, \ldots, x_N$ from q. To estimate q, the ERM principle indicates we should solve

$$\hat{p} := \operatorname*{argmin}_p \hat{R}(p) = \operatorname*{argmin}_p \left\{\frac{1}{N}\sum_{n=1}^{N} -\ln p(x_n)\right\} = \operatorname*{argmax}_p \left\{\sum_{n=1}^{N} \ln p(x_n)\right\}$$

To make the connection with maximum likelihood, let's now add the assumption that the unknown density lies in some parametric class $\{p(\cdot;\theta)\}_{\theta\in\Theta}$. Suppose that we know the parametric class, but the true value θ generating the data is unknown. Choosing our estimate $\hat{p}$ of q now reduces to choosing an estimate $\hat{\theta}$ of θ. Re-writing our optimization problem for this case, we obtain

$$\hat{\theta} = \operatorname*{argmax}_\theta \left\{\sum_{n=1}^{N} \ln p(x_n;\theta)\right\} = \operatorname*{argmax}_\theta \ \ell(\theta)$$

where ℓ is the log-likelihood. It follows from this expression that the ERM estimator is precisely the maximum likelihood estimator.

[10] More precisely, $D(q,p) = 0$ if and only if $p = q$ almost everywhere. Equality almost everywhere is a basic concept from measure theory that is (only just) beyond the scope of these notes. Note that if two density are equal almost everywhere then they share the same cdf, and hence represent the same distribution.

4.6.4 Concluding Comments

The ERM principle is a very general principle for solving statistical problems and producing estimators. For such a general method it is difficult to give a set of strong results showing that ERM produces good estimators. Indeed, there will be instances when ERM produces poor estimators, as discussed in §4.6.2. Having said that, some rather general consistency results have been obtained. The details are beyond the level of these notes. Some discussion can be found in [19].

4.7 Exercises

Ex. 4.7.1. Confirm (4.11): Show that, for any estimator $\hat{\theta}$ of θ, we have $\text{mse}[\hat{\theta}] = \text{var}[\hat{\theta}] + \text{bias}[\hat{\theta}]^2$.

Ex. 4.7.2. Confirm that for an IID sample $x_1, \ldots, x_N$ with variance σ^2, the sample variance s_N^2 defined in (4.1) is unbiased for σ^2.[11]

Ex. 4.7.3. Let $x_1, \ldots, x_N$ be IID with mean μ and variance σ^2. Let $\bar{x}_N$ be the sample mean, and let σ_N be a consistent estimator of σ. What is the limiting distribution of

$$y_N := N \left(\frac{\bar{x}_N - \mu}{\sigma_N} \right)^2$$

Ex. 4.7.4. Confirm that the maximizer of (4.12) is the sample mean of $x_1, \ldots, x_N$.

Ex. 4.7.5. Let $x_1, \ldots, x_N \overset{\text{IID}}{\sim} F$, where F is a cdf. Let $m_4 < \infty$ be the 4-th moment. That is,

$$m_4 := \int s^4 F(ds)$$

Define the plug in estimator of m_4. Is the estimator consistent? Why or why not?

Ex. 4.7.6. Suppose we are playing a slot machine (one-armed bandit) that either pays one dollar or nothing, with each payoff independent of the previous outcome. Let μ be the probability of winning (i.e., receiving one dollar). Having observed 100 plays $x_1, \ldots, x_{100}$, where $x_n \in \{0, 1\}$, a natural estimator of μ is the fraction of wins, which is just the sample mean $= \bar{x}$. Use the principle of maximum likelihood to obtain the same conclusion.[12]

[11] The sample standard deviation s (4.2) is typically biased. To see why, look up Jensen's inequality.

[12] Hint: Each x_n is a Bernoulli random variable, the probability mass function for which can be written as $p(s; \mu) := \mu^s (1 - \mu)^{1-s}$ for $s \in \{0, 1\}$.

Ex. 4.7.7. Show that $\hat{f}$ in (4.17) is a density for every N, every $\delta > 0$ and every realization of the sample.[13]

Ex. 4.7.8. Let x be a random variable with $\mu := \mathbb{E}[x]$. Consider the risk function given by $R(\theta) = \mathbb{E}[(\theta - x)^2]$. Show that μ is the minimizer of $R(\theta)$ over all $\theta \in \mathbb{R}$, without using differentiation.[14]

Ex. 4.7.9 (Computational)**.** Let $x_1, \ldots, x_N$ be IID and uniformly distributed on the interval $[0, 1]$. Let $\bar{x}_N$ be the sample mean. What is the expectation and variance of $\bar{x}_N$? For $N = 1, 2, 10, 500$, simulate 10,000 observations of the random variable $\bar{x}_N$. Histogram the observations, using one histogram for each value of N. (For example, the first histogram should be of 10,000 observations of $\bar{x}_1$.) What do you observe about these four distributions? What interpretation can you give?

Ex. 4.7.10 (Computational)**.** Extending your results in exercise 1.5.24, determine the cdf of $z := \max\{u_1, \ldots, u_N\}$, where $u_1, \ldots, u_N$ are N independent random variables uniformly distributed on $[0, 1]$. Check this by generating 1,000 draws of y and plotting the ecdf, along with your expression for the cdf. The ecdf should be close to the cdf. In the simulation, set $N = 5$.

Ex. 4.7.11 (Computational)**.** Implement the ecdf as your own user-defined function in R, based on the definition (i.e., that it reports the fraction of the sample falling below a given point).

Ex. 4.7.12. Let $\hat{\theta}_N$ be an estimator of θ. Show that if $\hat{\theta}_N$ is asymptotically normal, then $\hat{\theta}_N$ is consistent for θ. (Warning: This exercise is harder, and requires a bit of experience with analysis to solve properly.)

4.7.1 Solutions to Selected Exercises

Solution to Exercise 4.7.1. Adding and subtracting $\mathbb{E}[\hat{\theta}]$, we get

$$\text{mse}[\hat{\theta}] := \mathbb{E}[(\hat{\theta} - \theta)^2] = \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}] + \mathbb{E}[\hat{\theta}] - \theta)^2]$$

Expanding the square and minor manipulations yield the desired result. □

[13] Hint: You need to show that $\hat{f}$ is nonnegative and integrates to one. Showing that $\int \hat{f}(s)ds = 1$ is the hard part. Try a change-of-variables argument.

[14] Hint: Use the add and subtract strategy.

Solution to Exercise 4.7.2. We showed in §4.2.3 that

$$s^2 = \frac{N}{N-1}\left[\frac{1}{N}\sum_{n=1}^{N}(x_n-\mu)^2 - (\bar{x}-\mu)^2\right]$$

Taking expectations, applying the result $\operatorname{var}[\bar{x}] = \mathbb{E}\left[(\bar{x}-\mu)^2\right] = \sigma^2/N$ from (4.10) and rearranging gives $\mathbb{E}\left[s^2\right] = \sigma^2$ as claimed. □

Solution to Exercise 4.7.3. Let

$$w_N := \sqrt{N}\frac{\bar{x}_N - \mu}{\sigma_N} = \frac{\sigma}{\sigma_N}\sqrt{N}\frac{\bar{x}_N - \mu}{\sigma}$$

Since $\sigma_N \xrightarrow{p} \sigma$ by assumption, fact 1.4.1 on page 31 yields $\sigma/\sigma_N \xrightarrow{p} \sigma/\sigma = 1$. Applying the central limit theorem and Slutsky's theorem (fact 1.4.5 on page 34) together, we then have $w_N \xrightarrow{d} z \sim \mathcal{N}(0,1)$. By the continuous mapping theorem (fact 1.4.4 on page 34), $y_N = w_N^2$ converges in distribution to z^2. By fact 1.3.4 on page 1.3.4, the distribution of z^2 is $\chi^2(1)$. □

Solution to Exercise 4.7.5. The plug in estimator of m_4 is the sample fourth moment. The sample fourth moment is consistent for m_4 under the IID assumption by the law of large numbers, given the stated assumption that $m_4 < \infty$. □

Solution to Exercise 4.7.6. Each x_n is a Bernoulli random variable, with pmf given by $p(s;\mu) := \mu^s(1-\mu)^{1-s}$ for $s \in \{0,1\}$. By independence, the joint distribution is the product of the marginals, and hence the log likelihood is

$$\ell(\mu) = \sum_{n=1}^{N} \log p(x_n;\mu) = \sum_{n=1}^{N}\left[x_n \log\mu + (1-x_n)\log(1-\mu)\right]$$

Differentiating with respect to μ and setting the result equal to zero yields $\hat{\mu} = \bar{x}$ as claimed. □

Solution to Exercise 4.7.7. The nonnegativity of $\hat{f}$ is obvious. To show that $\int \hat{f}(s)ds = 1$, it's enough to show that

$$\int K\left(\frac{s-a}{\delta}\right)ds = \delta$$

for any given number a. This equality can be obtained by the change of variable $u := (s-a)/\delta$, which leads to

$$\int K\left(\frac{s-a}{\delta}\right)ds = \int K(u)\delta du = \delta\int K(u)du$$

Since K is a density, the proof is done. □

Solution to Exercise 4.7.8. Adding and subtracting μ, we can express $R(\theta)$ as

$$R(\theta) = \mathbb{E}\left\{[(\theta - \mu) + (\mu - x)]^2\right\}$$

Expanding this out and using $\mathbb{E}[x] = \mu$, we obtain $R(\theta) = (\theta - \mu)^2 + \text{var}[x]$. Evidently a minimum is obtained when $\theta = \mu$. □

Solution to Exercise 4.7.12. Fix $\delta > 0$. It suffices to show that for any positive number ϵ we have

$$\lim_{N\to\infty} \mathbb{P}\{|\hat{\theta}_N - \theta| > \delta\} \leq \epsilon \tag{4.31}$$

(If $a \geq 0$ and $a \leq \epsilon$ for any $\epsilon > 0$, then $a = 0$.) To establish (4.31), fix $\epsilon > 0$. Let z be standard normal, and choose M such that $\mathbb{P}\{|z| \geq M\} \leq \epsilon$. For N such that $\sqrt{N}\delta \geq M$ we have

$$\mathbb{P}\{|\hat{\theta}_N - \theta| > \delta\} = \mathbb{P}\{\sqrt{N}|\hat{\theta}_N - \theta| > \sqrt{N}\delta\} \leq \mathbb{P}\{\sqrt{N}|\hat{\theta}_N - \theta| > M\}$$

Taking $N \to \infty$, applying asymptotic normality, the continuous mapping theorem (fact 1.4.4 on page 34) and the definition of M gives (4.31). □

Chapter 5

Methods of Inference

[roadmap]

5.1 Making Inference about Theory

[roadmap]

5.1.1 Sampling Distributions

In §4.2 we emphasized the fact that statistics, and hence estimators, are random variables. For example, if $\hat{\theta}$ is an estimator of some quantity θ, then $\hat{\theta}$, being a statistic, must be an observable function of the data. If $x_1, \ldots, x_N$ is the data, and $\hat{\theta} = T(x_1, \ldots, x_N)$ for known function T, then $\hat{\theta}$ is the random variable

$$\hat{\theta}(\omega) = T(x_1(\omega), \ldots, x_N(\omega)) \qquad (\omega \in \Omega)$$

(See, for example, (4.5) on page 113.)

Being a random variable, $\hat{\theta}$ has a distribution, which is the cdf $G(s) = \mathbb{P}\{\hat{\theta} \leq s\}$. The distribution G of $\hat{\theta}$ is sometimes called its **sampling distribution**. Usually this distribution will depend on unknown quantities. For example, if $x_1, \ldots, x_N \overset{\text{IID}}{\sim} F$ for some unknown cdf F, then the distribution of $\hat{\theta}$ will depend on F and T. Thus, the sampling distribution depends partly on the known object T, and partly on the uknown object F.

Let's look at two examples. First, suppose that $x_1, \ldots, x_N$ is an IID sample from the normal distribution $\mathcal{N}(\theta, \sigma^2)$, where both θ and σ are unknown. Consider the sample mean $\bar{x}$ as an estimator of the mean θ. Combining fact 1.2.6 on page 24, (4.6) on page 114 and (4.10) on page 115, we obtain

$$x_1, \ldots, x_N \overset{\text{IID}}{\sim} \mathcal{N}(\theta, \sigma^2) \implies \bar{x} \sim \mathcal{N}(\theta, \sigma^2/N) \tag{5.1}$$

The right-hand side is the sampling distribution of $\bar{x}$. Although this expression provides some information, the sampling distribution still depends on the unknown quantities θ and σ.

As a second example, suppose that we are interested in the average length of time between incoming telephone calls at a call center during business hours. We model the duration x between two calls as having the exponential distribution, with density $f(s) = 1 - \exp(-\lambda s)$. The parameter λ is unknown. We propose to monitor consecutive calls until observations $x_1, \ldots x_N$ are recorded. Using these observations, we will estimate mean duration using the sample mean $\bar{x} = N^{-1} \sum_n x_n$.

Since sums of independent exponential random variables are known to have the gamma distribution, $\sum_{n=1}^N x_n$ must be gamma. Since scalar multiples of gammas are again gamma, $\bar{x} = N^{-1} \sum_{n=1}^N x_n$ must also be gamma. Thus, under our assumptions, when the data is collected and $\bar{x}$ is evaluated, its value will be a draw from a gamma distribution. Which particular gamma distribution depends on the unknown quantity λ.

5.1.2 Comparing Theory with Outcomes

In chapter 4, we were interested in estimating and predicting. We looked at ways to find estimators, and at the properties of these estimators. In this chapter, we are going to consider a different style of problem. The problem is one where we hold a belief or theory concerning the probabilities generating the data, and we are interested in whether the observed data provides evidence for or against that theory.

To illustrate, suppose again that we have $x_1, \ldots, x_N \overset{\text{IID}}{\sim} \mathcal{N}(\theta, \sigma^2)$, where both θ and σ are unknown. Suppose that some economic theory that implies a specific value θ_0 for the unknown parameter θ (prices should be equal to marginal cost, excess profits should be equal to zero, etc.) In this case, our interest in observing $\hat{\theta}$ will be: What light does this realization $\hat{\theta}$ shed on our theory?

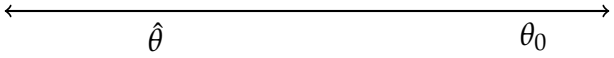


Figure 5.1: Theoretical and realized values

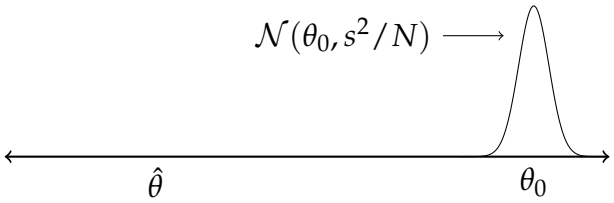


Figure 5.2: Estimated sampling distribution when the theory is correct

The naive answer is: The realization $\hat{\theta}$ appears to contradict our theory when it's a long way from our hypothesized value θ_0. But this is not really an answer until we specify what "a long way" is. For example, consider figure 5.1. Is θ_0 a long way from $\hat{\theta}$?

To determine what "a long way" means, we can look at the sampling distribution of $\hat{\theta}$. In the present case, this distribution is $\mathcal{N}(\theta, \sigma^2/N)$, as shown in (5.1). Although the parameters θ and σ are not known, our theory specifies that θ should be equal to θ_0, and the second parameter σ^2 can be estimated consistently by the sample variance s^2. Plugging the hypothesized mean θ_0 and the estimated variance s^2 into the sampling distribution gives the density in figure 5.2. Looking at this figure, we can see that θ_0 and $\hat{\theta}$ can indeed be regarded as a long way apart, in the sense that if our theory was correct, then $\hat{\theta}$ would be a realization from way out in the tail of its own distribution. Thus, the realization $\hat{\theta}$ is "unlikely" when our theory is true, and this fact can be construed as evidence against the theory.

In the following sections we formalize these ideas.

5.2 Confidence Sets

The idea of confidence sets is to provide a set of parameter values, distributions, or models that are "plausible" given the observed outcome of a statistic.

5.2.1 Parametric Examples

Suppose that we have a parametric class of models $\{M_\theta\}$ indexed by parameter $\theta \in \Theta \subset \mathbb{R}$. For example, the models might describe a set of cdfs $\{F_\theta\}$, or a family of proposed regression functions $\{f_\theta\}$ in a regression problem. We anticipate observing a sample $\mathbf{x} := (x_1, \ldots, x_N)$ generated by one of the models M_θ. Let F_θ be the joint distribution of the sample vector $\mathbf{x}$ when the data is generated by M_θ. We use the notation $\mathbb{P}_\theta$ to refer to probabilities for $\mathbf{x}$. For example, given $B \subset \mathbb{R}^N$, we let $\mathbb{P}_\theta\{\mathbf{x} \in B\}$ be the probability that $\mathbf{x} \in B$ given $\mathbf{x} \sim F_\theta$.[1]

Fix $\alpha \in (0,1)$. A random set $C(\mathbf{x}) \subset \Theta$ is called a $1-\alpha$ **confidence set** for θ if

$$\mathbb{P}_\theta\{\theta \in C(\mathbf{x})\} \geq 1 - \alpha \quad \text{for all} \quad \theta \in \Theta$$

Remember, it is the set that is random here, not the parameter θ. The "for all" statement is necessary since we don't actually know what the true value of θ is, and hence the experiment should be designed such that $\mathbb{P}_\theta\{\theta \in C(\mathbf{x})\} \geq 1 - \alpha$ regardless of which θ is generating the data. The interpretation of the confidence set is that, if we conduct all our statistical experiments with a fixed value of α, then our confidence sets will contain the true parameter about $(1-\alpha) \times 100\%$ of the time.

If $C(\mathbf{x})$ is an interval in $\mathbb{R}$, then $C(\mathbf{x})$ is also called a **confidence interval**.

Example 5.2.1. Let $x_1, \ldots, x_N \overset{\text{IID}}{\sim} \mathcal{N}(\theta, \sigma^2)$, where $\theta \in \Theta = \mathbb{R}$ is unknown. Suppose for the moment that σ is known. We wish to form a confidence interval for θ. By (5.1), we have

$$\sqrt{N}\frac{(\bar{x}_N - \theta)}{\sigma} \sim \mathcal{N}(0,1) \tag{5.2}$$

and hence, applying (1.14) on page 21,

$$\mathbb{P}_\theta\left\{\frac{\sqrt{N}}{\sigma}|\bar{x}_N - \theta| \leq z_{\alpha/2}\right\} = 1 - \alpha \quad \text{when} \quad z_{\alpha/2} := \Phi^{-1}(1 - \alpha/2)$$

Here Φ is the standard normal cdf. Some rearranging gives

$$\mathbb{P}_\theta\left\{\bar{x}_N - \frac{\sigma}{\sqrt{N}}z_{\alpha/2} \leq \theta \leq \bar{x}_N + \frac{\sigma}{\sqrt{N}}z_{\alpha/2}\right\} = 1 - \alpha$$

Since this argument is true regardless of the value of θ, we conclude that if $e_n := \sigma z_{\alpha/2}/\sqrt{N}$, then $C(\mathbf{x}) := (\bar{x}_N - e_n,\ \bar{x}_N + e_n)$ is a $1-\alpha$ confidence interval for θ

[1] For those readers who prefer more formal definitions, let $\mathbb{P}_\theta\{\mathbf{x} \in B\} := \int \mathbb{1}\{\mathbf{s} \in B\}F_\theta(d\mathbf{s})$.

Example 5.2.2. Continuing on from example 5.2.1, a more realistic situation is that σ is also unknown. In that case, a natural approach is to replace σ with the sample standard deviation s_N. In this case, (5.2) becomes

$$\sqrt{N}\frac{(\bar{x}_N - \theta)}{s_N} \sim F_{N-1} \tag{5.3}$$

where F_{N-1} is the cdf of the t-distribution with $N-1$ degrees of freedom.[2] The reasoning in example 5.2.1 now goes through when $z_{\alpha/2}$ is replaced by $t_{\alpha/2} := F_{N-1}^{-1}(1-\alpha/2)$, and we obtain

$$\mathbb{P}_\theta \left\{ \bar{x}_N - \frac{s_N}{\sqrt{N}} t_{\alpha/2} \leq \theta \leq \bar{x}_N + \frac{s_N}{\sqrt{N}} t_{\alpha/2} \right\} = 1 - \alpha$$

In example 5.2.2, note that since the standard deviation of $\bar{x}_N$ is $\sigma/\sqrt{N}$, the term $s_N/\sqrt{N}$ is a sample estimate of the standard deviation of the estimator $\bar{x}$. It is often called the standard error. More generally, if $\hat{\theta}$ is an estimator of some quantity θ, then the **standard error** is

$$\text{se}(\hat{\theta}) := \text{ a sample estimate of the standard deviation of } \hat{\theta}$$

Of course this is not really a formal definition because we haven't specified *which* estimate of the standard deviation we are talking about, but nevertheless the terminology is very common.[3] Using our new notation, we can write the confidence interval in example 5.2.2 as

$$C(\mathbf{x}) := (\bar{x}_N - \text{se}(\bar{x}_N) t_{\alpha/2},\ \bar{x}_N + \text{se}(\bar{x}_N) t_{\alpha/2}) \tag{5.4}$$

5.2.2 Asymptotic Confidence Sets

As the degrees of freedom increases, quantiles of the t-distribution converge to those of the normal distribution. Hence, for large N, we can replace $t_{\alpha/2}$ with $z_{\alpha/2}$ in (5.4) and the approximation will be accurate. This is part of a more general phenomenon related to the central limit theorem. Recall from §4.2.3 that an estimator $\hat{\theta}_N$ of $\theta \in \mathbb{R}$ is called asymptotically normal if

$$\sqrt{N}(\hat{\theta}_N - \theta) \xrightarrow{d} \sqrt{v(\theta)}\, Z \quad \text{as} \quad N \to \infty \tag{5.5}$$

[2] The proof of this statement is clearly related to fact 1.3.6 on page 27. The details are a bit fiddly. We omit them because more general results are established in chapter 7.

[3] Sometimes the term "standard error" is used to refer to the standard deviation of the estimator, rather than an estimate of the standard deviation.

where $v(\theta)$ is some positive constant and Z is standard normal. The constant $v(\theta)$ is called the asymptotic variance of $\hat{\theta}_N$. Many estimators have this property. For example, most maximum likelihood estimators are asymptotically normal, as are smooth transformations of sample means (theorem 1.4.3 on page 37).

Now suppose we have a sequence of statistics $\text{se}(\hat{\theta}_N)$ such that

$$\sqrt{N}\,\text{se}(\hat{\theta}_N) \xrightarrow{p} \sqrt{v(\theta)} \quad \text{as} \quad N \to \infty \tag{5.6}$$

As exercise 5.5.1 asks you to show, (5.5) and (5.6) imply that

$$\frac{\hat{\theta}_N - \theta}{\text{se}(\hat{\theta}_N)} \xrightarrow{d} Z \quad \text{as} \quad N \to \infty \tag{5.7}$$

As a result, for large N, we can write

$$\hat{\theta}_N \approx \theta + \text{se}(\hat{\theta}_N) Z \tag{5.8}$$

where the approximate equality is in terms of distribution. We see that $\text{se}(\hat{\theta}_N)$ is an approximation to the standard deviation of $\hat{\theta}_N$, which explains our choice of notation. As before, $\text{se}(\hat{\theta}_N)$ is referred to as the standard error of the estimator.

A sequence of random sets $C_N(\mathbf{x}) \subset \Theta$ is said to form an **asymptotic $1-\alpha$ confidence set** for θ if

$$\lim_{N\to\infty} \mathbb{P}_\theta\{\theta \in C_N(\mathbf{x})\} \geq 1 - \alpha \quad \text{for all} \quad \theta \in \Theta$$

For our asymptotically normal estimator $\hat{\theta}_N$, the sequence

$$C_N(\mathbf{x}) := (\hat{\theta}_N - \text{se}(\hat{\theta}_N) z_{\alpha/2},\, \hat{\theta}_N + \text{se}(\hat{\theta}_N) z_{\alpha/2}) \tag{5.9}$$

can be used because, rearranging, taking the limit and applying (5.7),

$$\begin{aligned}
\lim_{N\to\infty} \mathbb{P}_\theta\{\theta \in C_N(\mathbf{x})\} &= \lim_{N\to\infty} \mathbb{P}_\theta\left\{\hat{\theta}_N - \text{se}(\hat{\theta}_N) z_{\alpha/2} \leq \theta \leq \hat{\theta}_N + \text{se}(\hat{\theta}_N) z_{\alpha/2}\right\} \\
&= \lim_{N\to\infty} \mathbb{P}_\theta\left\{-z_{\alpha/2} \leq \frac{\hat{\theta}_N - \theta}{\text{se}(\hat{\theta}_N)} \leq z_{\alpha/2}\right\} \\
&= 1 - \alpha
\end{aligned}$$

Looking at (5.9) gives a good indication of why standard errors are normally reported along with the point estimate. For example, we have the following useful rule of thumb: If $\alpha = 0.05$, then $z_{\alpha/2} \approx 2$, so there is a 95% likelihood that the true parameter lies within 2 standard deviations of the observed value of our estimator.

5.2.3 A Nonparametric Example

Confidence sets can be obtained in nonparametric settings too. A nice example is confidence sets for the ecdf. In § 4.5.2, we learned that if $x_1, \ldots, x_N$ are IID draws from some cdf F, and F_N is the corresponding ecdf, then $\sup_s |F_N(s) - F(s)| \xrightarrow{p} 0$. This is the fundamental theorem of statistics (theorem 4.5.1). In 1933, A.N. Kolmogorov used an extension of the central limit theorem to obtain an asymptotic distribution for this term. In particular, he showed that

$$\sqrt{N} \sup_{s \in \mathbb{R}} |F_N(s) - F(s)| \xrightarrow{d} K \tag{5.10}$$

where K is the **Kolmogorov** distribution

$$K(s) := \frac{\sqrt{2\pi}}{s} \sum_{i=1}^{\infty} \exp\left[-\frac{(2i-1)^2 \pi^2}{8s^2}\right] \qquad (s \geq 0) \tag{5.11}$$

Notice that, like in the CLT, the limiting distribution K is independent of the cdf F that generates the data.

We can use this result to produce an asymptotic $1 - \alpha$ confidence set for F. To do so, let $\mathfrak{F}$ be the set of all cdfs on $\mathbb{R}$, let $k_{1-\alpha} := K^{-1}(1-\alpha)$, and let

$$C_N(\mathbf{x}) := \left\{ F \in \mathfrak{F} \ : \ F_N(s) - \frac{k_{1-\alpha}}{\sqrt{N}} \leq G(s) \leq F_N(s) + \frac{k_{1-\alpha}}{\sqrt{N}} \ \text{ for all } \ s \in \mathbb{R} \right\}$$

The set $C_N(\mathbf{x}) \subset \mathfrak{F}$ is an asymptotic $1 - \alpha$ confidence set for F. Indeed, rearranging the expression, we get

$$\begin{aligned} \{F \in C_N(\mathbf{x})\} &= \left\{ -k_{1-\alpha} \leq \sqrt{N}(F_N(s) - F(s)) \leq k_{1-\alpha} \ \text{ for all } \ s \right\} \\ &= \left\{ \sqrt{N}|F_N(s) - F(s)| \leq k_{1-\alpha} \ \text{ for all } \ s \right\} \\ &= \left\{ \sup_s \sqrt{N}|F_N(s) - F(s)| \leq k_{1-\alpha} \right\} \end{aligned}$$

Hopefully the last equality is clear.[4] Applying (5.10) now confirms our claim:

$$\lim_{N \to \infty} \mathbb{P}\{F \in C_N(\mathbf{x})\} = \lim_{N \to \infty} \mathbb{P}\left\{ \sup_s \sqrt{N}|F_N(s) - F(s)| \leq k_{1-\alpha} \right\} = 1 - \alpha$$

[4] If $g \colon D \to \mathbb{R}$ and $g(s) \leq M$ for all $s \in D$, then $\sup_{s \in D} g(s) \leq M$. Conversely, if $\sup_{s \in D} g(s) \leq M$, then $g(s) \leq M$ for all $s \in D$.

Given our data $x_1, \ldots, x_N$ and the corresponding ecdf F_N, we can present the confidence set $C_N(\mathbf{x})$ visually by plotting the lower bound function $F_N(s) - k_{1-\alpha}/\sqrt{N}$ and the the upper bound function $F_N(s) + k_{1-\alpha}/\sqrt{N}$. This is done in figure 5.3 for $\alpha = 0.05$. The data is generated using an arbitrary distribution F (the t-distribtion with 2 degrees of freedom). In figures (a) and (b), the true function F is not shown. In (c) and (d), F is shown in red. The preceding theory tells us that realizations of the confidence set will catch the true function F about 95 times in 100.

The code for producing (d) of figure 5.3 is shown in listing 5. In the code, you will see that the value we used for $k_{1-\alpha} = k_{0.95} := K^{-1}(0.95)$ was 1.36. This value was obtained numerically from the definition of K in (5.11). The technique used was to truncate the infinite sum in (5.11) at 20 to provide an approximation to K, and then search for an s satisfying $K(s) = 1 - \alpha$, or, equivalently, $f(s) = 0$ when $f(s) := K(s) - 1 + \alpha$. The root of f was found using the R univariate root-finding function `uniroot`. See listing 6.

Listing 5 The source code for figure 5.3

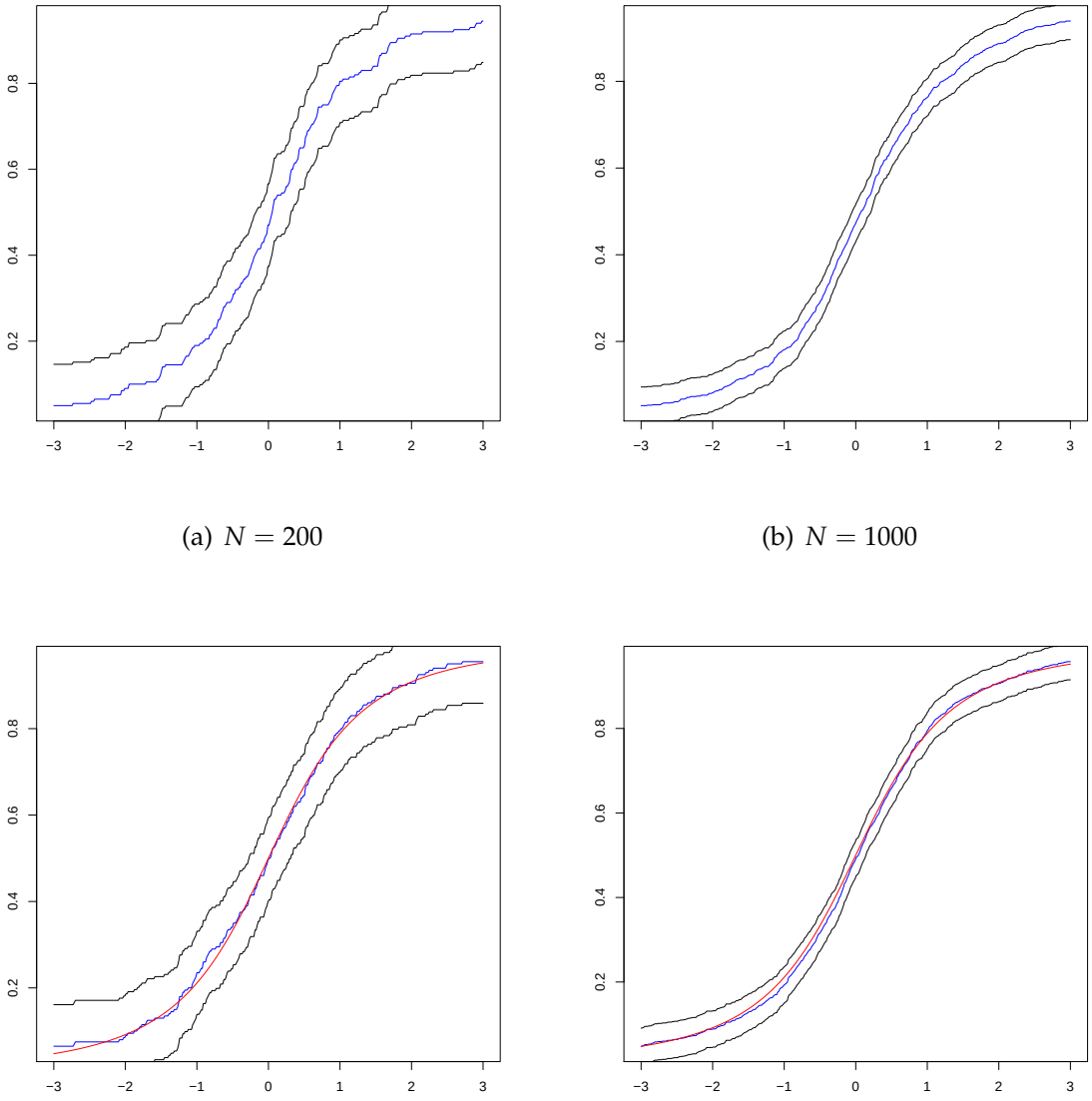
```
samp_size <- 1000
grid_size <- 400
xgrid <- seq(-3, 3, length=grid_size)

FN <- function(s, X) return(mean(X <= s))  # ecdf

X <- rt(samp_size, 2)  # RVs from t-dist with 2 DF
Y <- numeric(length(xgrid))
for (i in 1:length(xgrid)) Y[i] <- FN(xgrid[i], X)
Y_upper <- Y + 1.36 / sqrt(samp_size)
Y_lower <- Y - 1.36 / sqrt(samp_size)
plot(xgrid, Y, type="l", col="blue", xlab="", ylab="")
lines(xgrid, Y_upper)
lines(xgrid, Y_lower)
lines(xgrid, pt(xgrid, 2), col="red")
```

5.3 Hypothesis Tests

[roadmap]



(a) $N = 200$

(b) $N = 1000$

(c) $N = 200$, true F shown in red

(d) $N = 1000$, true F shown in red

Figure 5.3: Confidence sets for the ecdf

Listing 6 Finding the value of $k_{1-\alpha}$

```
K <- function(s) {  # Kolmogorov cdf
    a <- sqrt(2 * pi) / s
    b <- 0
    for (i in 1:20) {
        b <- b + exp(- (2 * i - 1)^2 * (pi^2 / (8 * s^2)))
    }
    return (a * b)
}

alpha <- 0.05
f <- function(s) return(K(s) - 1 + alpha)
ur <- uniroot(f, lower=1.2, upper=1.5)
print(ur)
```

5.3.1 The Framework

Hypothesis testing begins with the specification of a **null hypothesis**, which is a statement that the observed data is being generated by a certain model, or one of a certain class of models. For example, suppose we observe data pairs (x_n, y_n). Our null hypothesis might be that all pairs are generated by a particular functional relationship $y = f(x) + u$, where f is a specific function and u has a specific distribution. Alternatively, our null hypothesis might be that the distribution of u belongs to a certain parametric class of distributions, and that the function f lies in some particular set of functions $\mathcal{H}$, such as the increasing functions, or the twice differentiable functions, or the set of 3rd order polynomials.

One might imagine that the standard procedure in statistics is to show the validity of the null hypothesis, but it is not. Rather, a hypothesis test is an attempt *reject* the null. Karl Popper (1902–1994) was a major originator and proponent of this approach. To illustrate why we should focus on rejection, Popper used the example of testing the theory that all swans are white. (Consider this to be our null hypothesis.) It is futile to attempt to prove this theory correct: Regardless of how many white swans we find, no amount can ever confirm the claim that that *all* swans on planet earth are white. On the other hand, a single black swan can show the claim to be false. In this sense, attempting to falsify a theory (i.e., reject the null) is more constructive than attempting to confirm it.

Although we attempt to reject the null hypothesis, we do so only if strong evidence against it is observed. To understand how this convention came about, suppose that we have a collection of theories about how the economy works. The procedure would then be to step through the theories, at each stage taking correctness of the theory as the null hypothesis and attempting to reject. If the theory is rejected then we can discard it, and go on to the remaining theories. This is a useful process of elimination. However, we don't want to mistakenly discard a good theory. Hence, we only reject when there is strong evidence against the null hypothesis.

5.3.2 Constructing Tests

Let's look at testing in the parametric setting of §5.2.1. We have a parametric class of models $\{M_\theta\}$ indexed by parameter $\theta \in \Theta \subset \mathbb{R}$. We let F_θ be the joint distribution of the sample vector $\mathbf{x}$ when the data is generated by M_θ. We use the notation $\mathbb{P}_\theta$ to refer to probabilities for $\mathbf{x}$. For example, given $B \subset \mathbb{R}^N$, we let $\mathbb{P}_\theta\{\mathbf{x} \in B\}$ be the probability that $\mathbf{x} \in B$ given $\mathbf{x} \sim F_\theta$. A null hypothesis is a specification of the set of models $\{M_\theta\}$ that we believe generated $\mathbf{x}$. This amounts to specifying a subset Θ_0 of Θ. The null hypothesis is often written as

$$H_0 \colon \theta \in \Theta_0$$

If Θ_0 is a singleton, then the null hypothesis is called a **simple hypothesis**. If not, then the null hypothesis is called a **composite hypothesis**.

A test of the null hypothesis amounts to a test of whether the observed data was generated by M_θ for some $\theta \in \Theta_0$. Formally, a **test** is a binary function ϕ mapping the observed data $\mathbf{x}$ into $\{0, 1\}$. The decision rule is[5]

$$\begin{aligned} &\text{if } \phi(\mathbf{x}) = 1, \text{ then reject } H_0 \\ &\text{if } \phi(\mathbf{x}) = 0, \text{ then do not reject } H_0 \end{aligned}$$

Remark 5.3.1. Note that, prior to implementation of the test, $\phi(\mathbf{x})$ is to be considered as a random variable, the distribution of which depends on the distribution of $\mathbf{x}$ and the function ϕ. Note also that failing to reject H_0 should not be confused with accepting H_0! More on this below.

[5]Some texts identify tests with a rejection region, which is a subset R of $\mathbb{R}^N$. ($\mathbb{R}^N$ is the set of N-vectors—see chapter 2 for more.) The null is rejected if $\mathbf{x} \in R$. This is equivalent to our formulation: If a rejection region R is specified, then we take ϕ to be defined by $\phi(\mathbf{x}) := \mathbb{1}\{\mathbf{x} \in R\}$. Conversely, if ϕ is specified, then we take R as equal to $\{\mathbf{s} \in \mathbb{R}^N : \phi(\mathbf{s}) = 1\}$.

Remark 5.3.2. Although we are using the language of parametric hypothesis testing, this is only for convenience. We can also think of θ as an arbitrary index over a (possibly nonparametric) class of models.

The outcome of our test depends on the random sample $\mathbf{x}$, and, being random, its realization can be misleading. There are two different ways in which the realization can mislead us. First, we can mistakenly reject the null hypothesis when it is in fact true. This is called **type I error**. Second, we can fail to reject the null hypothesis when it is false. This is called **type II error**.

The **power function** associated with the test ϕ is the function

$$\beta(\theta) = \mathbb{P}_\theta\{\phi(\mathbf{x}) = 1\} \qquad (\theta \in \Theta)$$

In other words, $\beta(\theta)$ is the probability that the test rejects when the data is generated by M_θ. Ideally, we would like $\beta(\theta) = 0$ when $\theta \in \Theta_0$, and $\beta(\theta) = 1$ when $\theta \notin \Theta_0$. In practice, this is usually difficult to achieve.

As discussed above, we tend to be conservative in rejecting the null, because we don't want to discard good theories. For this reason, it is traditional to keep the probability of type I error small. Then, if our test tells us to reject the null, it's unlikely the null is true. Because of this, the standard procedure is to choose a small number α such as 0.05 or 0.01, and then adjust the test such that

$$\beta(\theta) \leq \alpha \quad \text{for all } \theta \in \Theta_0 \tag{5.12}$$

If (5.12) holds, then the test is said to be of **size** α.

In constructing tests, a common (but not universal) set up is to define a real-valued **test statistic** T and a **critical value** c, and then set[6]

$$\phi(\mathbf{x}) := \mathbb{1}\{T(\mathbf{x}) > c\} \tag{5.13}$$

The pair (T, c) then defines the test, and the rule becomes:

$$\text{Reject } H_0 \text{ if and only if } T(\mathbf{x}) > c$$

[6] A minor point: A test statistic is a kind of statistic, in the sense that it is a function of the data. Usually, statistics are thought of as being computable given the data. This means that they do not depend on any unknown quantities. In the case of a test statistic, it is common to allow the test statistic to depend on unknown parameters (a subset of Θ), with the caveat that the values of these unknown parameters are all pinned down by the null hypothesis. (Otherwise the test cannot be implemented.)

Example 5.3.1. Suppose we "know" that $x_1, \ldots, x_N \overset{\text{IID}}{\sim} \mathcal{N}(\theta, 1)$ for some $\theta \in \mathbb{R}$, but the value of θ is unknown. The null hypothesis is $\theta \leq 0$, or $\Theta_0 = (-\infty, 0]$. To construct a test, we seek a suitable test statistic. Since we want to make inference about the mean, a natural choice for our statistic is the sample mean, so that

$$T(\mathbf{x}) :=: T(x_1, \ldots, x_N) := \bar{x}_N$$

Each $c \in \mathbb{R}$ now gives us a test via (5.13), with power function $\beta(\theta) = \mathbb{P}_\theta\{\bar{x}_N > c\}$. We can obtain a clearer expression for the power function by observing that $\bar{x}_N \sim \mathcal{N}(\theta, 1/N)$. As a result, if Φ is the cdf of the standard normal distribution on $\mathbb{R}$ and $z \sim \Phi$, then

$$\mathbb{P}_\theta\{\bar{x}_N \leq c\} = \mathbb{P}\{\theta + N^{-1/2}z \leq c\} = \mathbb{P}\{z \leq N^{1/2}(c - \theta)\} = \Phi[N^{1/2}(c - \theta)]$$

$$\therefore \quad \beta(\theta) = 1 - \Phi[N^{1/2}(c - \theta)] \tag{5.14}$$

Given c, the power function is increasing in θ, because higher θ pushes up the mean of $\bar{x}_N$, making the event $\{\bar{x}_N > c\}$ more likely. Given θ, the function is decreasing in c, because higher c makes the event $\{\bar{x}_N > c\}$ less likely. A plot of β is presented in figure 5.4 for two different values of c.[7]

5.3.3 Choosing Critical Values

Let's think a bit more about the test in (5.13). Typically, the choice of T is suggested by the problem. For example, if our hypothesis is a statement about the second moment of a random variable, then we might take T to be the sample second moment. Once T is fixed, we need to adjust c such that (T, c) attains the appropriate size. Thus, the standard procedure is to:

1. choose a desired size α according to our tolerance for type I error,
2. identify a suitable test statistic T, and
3. choose a critical value c so that (T, c) is of size α.

In performing the last step, it's common to choose c such that (5.12) holds with equality. In this case, the problem is to choose c to solve

$$\alpha = \sup_{\theta \in \Theta_0} \mathbb{P}_\theta\{T(\mathbf{x}) > c\} \tag{5.15}$$

[7] N is fixed at 10.

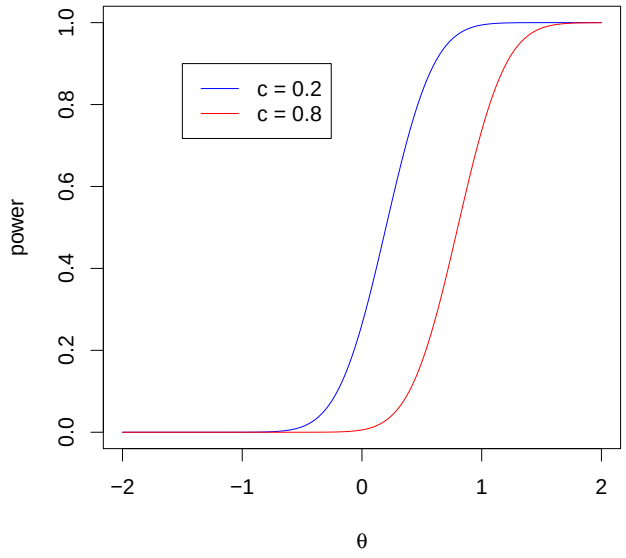


Figure 5.4: The power function β

Figure 5.5 gives an illustration. In the figure, we've taken Θ_0 to be the two element set $\{\theta_a, \theta_b\}$. The blue line gives an imaginary distribution of $T(\mathbf{x})$ when $\mathbf{x} \sim F_{\theta_a}$, represented as a density. The black line gives the same for θ_b. Assuming that a value of α be prescribed, the next step is to determine the value of c such that (5.15) holds. Here, we choose c such that the largest of the two shaded areas is equal to α.

Example 5.3.2. Let's look again at example 5.3.1, where $x_1, \ldots, x_N \overset{\text{IID}}{\sim} \mathcal{N}(\theta, 1)$ for θ unknown, and our null hypothesis is $\theta \leq 0$. Given α, our task is to find the appropriate critical value c so that the test (T, c) is of size α. To solve for c given α we use (5.15). Applying the expression for the power function in (5.14), this becomes

$$\alpha = \sup_{\theta \leq 0} \{1 - \Phi[N^{1/2}(c - \theta)]\}$$

The right-hand side is increasing in θ, so the supremum is obtained by setting $\theta = 0$. Setting $\theta = 0$ and solving for c, we obtain

$$c(\alpha) := N^{-1/2} \Phi^{-1}(1 - \alpha)$$

where Φ^{-1} is the quantile function of the standard normal distribution. In R, this function can be evaluated using `qnorm`. For example,

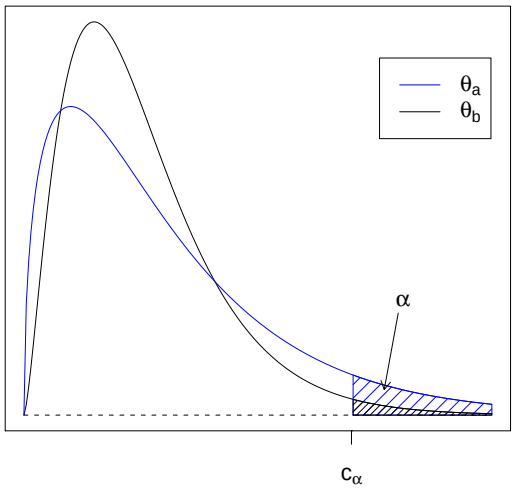


Figure 5.5: Determining the critical value

```
> alpha = 0.05
> qnorm(1 - alpha)
[1] 1.644854
```

Since Φ^{-1} is increasing, we see that smaller α corresponds to larger $c(\alpha)$—we reduce the probability of type I error by increasing the critical value that the mean $\bar{x}_N$ must obtain for rejection to occur. Also, higher N brings down $c(\alpha)$: More data allows us to reduce the critical value without increasing the probability of rejecting a true null.

5.3.4 p-Values

Typically, a test that rejects at size 0.05 will also reject at size 0.1, but may not reject at size 0.01. Lower α means less tolerance for type I error, and forces the critical value to become larger. Hence, for a fixed value of the test statistic, the result of the test may switch from reject to accept. A natural question, then, is: What is the smallest value of α at which we can still reject a given test statistic? This value is called the p-value.

Let's give a formal definition. Consider again the general parametric setting of §5.3. We have a null hypothesis is $H_0 : \theta \in \Theta_0$ and, for each $\alpha \in (0,1)$, a test $(T, c(\alpha))$ of size α. We assume here that $c(\alpha)$ is determined via the relationship in (5.15). In this setting, the p-value of the test is defined as

$$p(\mathbf{x}) := \inf\{\alpha \in (0,1) \,:\, c(\alpha) < T(\mathbf{x})\}$$

Roughly speaking, this is the α at which the test switches from accept to reject. Typically $\alpha \mapsto c(\alpha)$ is continuous, and in this case the expression for $p(\mathbf{x})$ reduces to

$$p(\mathbf{x}) := \text{ the } \alpha \text{ such that } c(\alpha) = T(\mathbf{x}) \tag{5.16}$$

Example 5.3.3. Recall the test (5.21). Here $c(\alpha) := \Phi^{-1}(1 - \alpha/2)$, and $c(\alpha)$ is continuous in α, so we can apply the definition of $p(\mathbf{x})$ in (5.16). To solve for $p(\mathbf{x})$, then, we need to solve for α in the expression $\Phi^{-1}(1 - \alpha/2) = |t_N(\mathbf{x})|$. With a bit of rearranging and an application of symmetry (page 17), we obtain

$$p(\mathbf{x}) = 2\Phi(-|t_N(\mathbf{x})|) \tag{5.17}$$

5.3.5 Asymptotic Tests

As we saw in examples 5.3.1 and 5.3.2, constructing appropriate tests requires knowledge of the distribution of the test statistic. In many cases, however, we know relatively little about the distribution of the test statistic. Often we don't want to make parametric assumptions about the distribution of the underlying data $\mathbf{x}$. And even if we make such assumptions, it may still be hard to work out the implied distribution of a given statistic $T(\mathbf{x})$.

Fortunately, for many problems, the central limit theorem and its consequences provide an elegant solution: Even if we don't know the distribution of the test statistic, we may still be able to determine it's *asymptotic* distribution via the CLT. Once we know the asymptotic distribution, we have an idea of the power of the test, at least for large samples, and can hopefully choose critical values to obtain an appropriate size.

To go down this path we need to switch to a notion of asymptotic size, rather than finite sample size. Writing β_N instead of β to emphasize the fact that the power function typically depends on sample size, a test is called **asymptotically of size α** if

$$\lim_{N \to \infty} \beta_N(\theta) \leq \alpha \quad \text{for all } \theta \in \Theta_0 \tag{5.18}$$

To illustrate how we might go about constructing a test which is asymptotically of size α, consider the data presented in figure 5.6. The histogram is of standardized daily returns on the Nikkei 225 index from January 1984 until May 2009. Here "standardized" means that we have subtracted the sample mean and divided by the sample deviation. If the original returns were normally distributed, then the standardized returns would be approximately standard normal. The code for producing the histogram is in the first part of listing 7. The data file can be downloaded at

```
http://johnstachurski.net/emet/nikkei_daily.txt
```

The standard normal density has been superimposed on the histogram in blue. We can see that the fit is not particularly good. The histogram suggests that the density of returns is more peaked and has heavier tails than the normal density. This is a common observation for asset price returns.

Let's think about making this analysis more precise by constructing an asymptotic test. The construction of the test essentially "inverts" the confidence set in §5.2.3. To begin, let Φ be the standard normal cdf, and let the null hypothesis be that standardized returns are IID draws from Φ. Let α be given, and let $k_{1-\alpha} = K^{-1}(1-\alpha)$ be the $1-\alpha$ quantile of the Kolmogorov distribution K, as defined in (5.11). Finally, let F_N be the ecdf of the data. If the null hypothesis is true, then, by (5.10) on page 159, we have

$$\sqrt{N} \sup_{s \in \mathbb{R}} |F_N(s) - \Phi(s)| \xrightarrow{d} K \tag{5.19}$$

For the test

$$\phi_N(\mathbf{x}) := \mathbb{1}\left\{ \sqrt{N} \sup_{s \in \mathbb{R}} |F_N(s) - \Phi(s)| > k_{1-\alpha} \right\}$$

let $\beta_N(\Phi)$ be the value of the power function when the null hypothesis is true. By (5.19), we have

$$\lim_{N\to\infty} \beta_N(\Phi) = \lim_{N\to\infty} \mathbb{P}\left\{ \sqrt{N} \sup_{s \in \mathbb{R}} |F_N(s) - \Phi(s)| > k_{1-\alpha} \right\} = \alpha$$

Hence (5.18) is verified, and the test is asymptotically of size α.

The value of the statistic $\sqrt{N} \sup_{s\in\mathbb{R}} |F_N(s) - \Phi(s)|$ produced by listing 7 is 5.67. If $\alpha = 0.05$, then, as shown in §5.2.3, the critical value $k_{1-\alpha}$ is 1.36. The test statistic exceeds the critical value, and the null hypothesis is rejected.

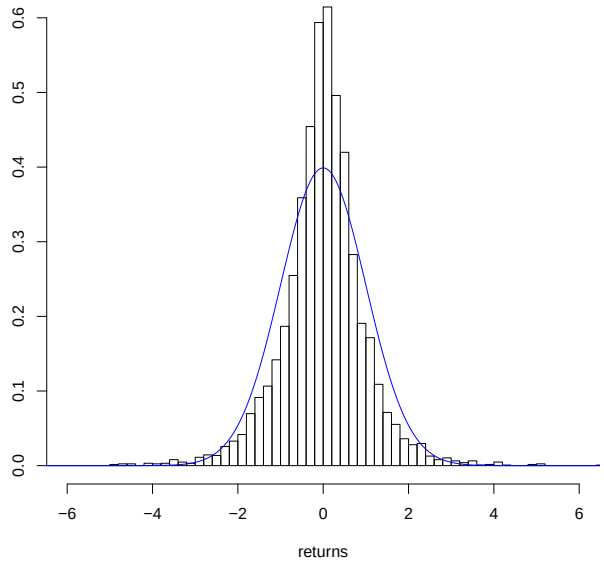


Figure 5.6: Standardized daily returns, Nikkei 225

Looking back on what we've done, there's an obvious weakness in our approach. Our null hypothesis was that standardized returns are IID draws from the standard normal density. The rejection suggests this null is false, but it may be the IID assumption rather than the normality assumption that makes our null a poor fit to the data. The test we've implemented can be modified to tackle the case of dependent data, but such a discussion is beyond the scope of these notes. See Negri and Nishiyama (2010) for one such test and many references.

5.4 Use and Misuse of Testing

[roadmap]

5.4.1 Testing and Model Selection

To to think about testing and model selection in the context of econometrics, it's interesting to discuss the current state of hypothesis testing in macroeconomics in

Listing 7 Testing normality

```
nikkei <- read.table("nikkei_daily.txt", header=T, sep=",")
dates <- as.Date(rev(nikkei$Date))
close <- rev(nikkei$Close)
returns <- diff(log(close))
data <- (returns - mean(returns)) / sd(returns)
hist(data, breaks=100, prob=T, xlab="returns",
    ylab="", main="", xlim=c(-6,6))
xgrid <- seq(min(data), max(data), length=400)
lines(xgrid, dnorm(xgrid), col="blue")

FN <- function(s) return(mean(data <= s))  # ecdf
FN <- Vectorize(FN)
m <- max(abs(FN(xgrid) - pnorm(xgrid)))
print(sqrt(length(data)) * m)
```

particular. You are probably aware that in the 1970s, the rational expectations revolution ushered in a brand new class of macroeconomic models. One of the first things the proponents of rational expectations did was to test these models against data. The results were disappointing. Thomas Sargent's account runs as follows (Evans and Honkapohja, 2005):

> My recollection is that Bob Lucas and Ed Prescott were initially very enthusiastic about rational expectations econometrics. After all, it simply involved imposing on ourselves the same high standards we had criticized the Keynesians for failing to live up to. But after about five years of doing likelihood ratio tests on rational expectations models, I recall Bob Lucas and Ed Prescott telling me that those tests were rejecting too many good models.

As a result, many proponents of these "good" models have moved away from formal hypothesis testing in favor of so-called calibration. The consensus of this school of though can be paraphrased as "No model is a true description of the real world. Hence, *I already know that my model is wrong*. Rejection of my model using standard inference tells me nothing new."

This line of argument runs contrary to the standard paradigm under which most of science and statistical testing takes place. The standard paradigm recognizes that all

models are wrong. Given that all models are wrong, we try to cull old ones that perform poorly and produce new ones that perform better. The process of culling takes place by statistical rejection. Some models are easily rejected. Others are harder to reject. The overall outcome resembles survival of the fittest. Those models that explain observed phenomena effectively, and, at the same time, are *most difficult to reject*, will survive.[8]

On the other hand, it should be acknowledged that social science is not physics, and our models tend to be more imperfect. For any given model a poor fit to the data can usually be found along some dimension, thus enabling rejection. But perhaps some of these models are still useful in guiding our thinking, and perhaps they represent the best models we have for now. None of these are easy questions.

5.4.2 Accepting the Null?

power and lack of power. failing to reject doesn't mean the null is right.

example: unit root tests.

example: diagnostic tests for regression.

5.5 Exercises

Ex. 5.5.1. Show that (5.7) is valid when (5.5) and (5.6) hold.

Ex. 5.5.2. Let $\alpha \in (0,1)$ and set $z_{\alpha/2} := \Phi^{-1}(1-\alpha/2)$. Show that if $t_N(\mathbf{x}) \xrightarrow{d} \mathcal{N}(0,1)$ whenever H_0 is true, and $T_N(\mathbf{x}) = |t_N(\mathbf{x})|$, the sequence of tests $\phi_N(\mathbf{x}) := \mathbb{1}\{T_N(\mathbf{x}) > z_{\alpha/2}\}$ is asymptotically of size α.

Ex. 5.5.3. Let $x_1, \ldots, x_N$ be an IID sample with mean θ and variance σ^2. Assume that both θ and σ are unknown. We wish to test the hypothesis $H_0 \colon \theta = \theta_0$. Consider the statistic

$$t_N := \sqrt{N} \left\{ \frac{\bar{x}_N - \theta_0}{s_N} \right\} \tag{5.20}$$

[8]Incidentally, given this description of model selection, it might appear there is a way for us to produce a model that survives forever in the pool of currently acceptable theories, without ever being culled via rejection: Just make sure that the model has no testable implications. However, this strategy does not work, because a model with no testable implications cannot be considered as a theory of anything. The standard definition of a "scientific" theory, as proposed by Karl Popper, is that the theory has one or more testable implications. In other words, the theory can be *falsified*.

With reference to exercise 5.5.2, show that the sequence of tests

$$\phi_N(\mathbf{x}) := \mathbb{1}\{|t_N| > z_{\alpha/2}\} \tag{5.21}$$

is asymptotically of size α.

Ex. 5.5.4. If a test is well designed, then the rejection probability will converge to one as $N \to \infty$ whenever H_0 is false:

$$\beta_N(\theta) \to 1 \text{ as } N \to \infty \quad \text{whenever} \quad \theta \in \Theta_1$$

Such a test is said to be **consistent**. Show that the test in exercise 5.5.3 is consistent whenever $\alpha \in (0, 1)$.[9]

Ex. 5.5.5 (Computational)**.** The chi-squared goodness of fit test is used to test whether or not a given data set $x_1, \ldots, x_N$ was generated from a particular discrete distribution, described by a pmf $p_1, \ldots, p_J$ over values $1, \ldots, J$. More precisely, let $x_1, \ldots, x_N$ be N random variables, each x_n taking integer values between 1 and J. The null hypothesis of the test is that the sample $x_1, \ldots, x_N$ is IID with $\mathbb{P}\{x_n = j\} = p_j$ for $n \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, J\}$. The test statistic is given by

$$X := N \sum_{j=1}^{J} \frac{(q_j - p_j)^2}{p_j} \tag{5.22}$$

where q_j is the fraction of the sample $x_1, \ldots, x_N$ taking the value j. Write a function in R called `chsqts` that takes two arguments `observations` and `p`, where `observations` is a vector storing the sample $x_1, \ldots, x_N$, and `p` is a vector storing the values $p_1, \ldots, p_J$. The function call `chsqts(observations, p)` should return the value X in equation (5.22).[10]

Ex. 5.5.6 (Computational)**.** This exercise continues on from exercise 5.5.5. Under the null hypothesis, X in (5.22) is asymptotically chi-squared with $J - 1$ degrees of freedom. Let $J = 3$ with $p_1 = 0.2$, $p_2 = 0.2$ and $p_3 = 0.6$. Let $N = 20$. By repeatedly simulating 20 IID observations $x_1, \ldots, x_{20}$ from this pmf,[11] generate 5,000 independent observations of the statistic X, and store them in a vector called `obsX`. Plot the ecdf of `obsX`. In the same figure, plot the chi-squared cdf with 2 degrees of freedom. The functions should be close.[12]

[9] Hint: In essence, you need to show that the absolute value of the test statistic $|t_N|$ in (5.20) diverges to infinity when $\theta \neq \theta_0$. Try the add and subtract strategy, replacing the expresion $\bar{x}_N - \theta_0$ in (5.20) with $(\bar{x}_N - \theta) + (\theta - \theta_0)$.

[10] R has a built-in function called `chisq.test` for implementing the chi-squared goodness of fit test. Do not use this built-in function in your solution.

[11] In particular, draw each x_n such that $\mathbb{P}\{x_n = j\} = p_j$ for $j = 1, 2, 3$.

[12] You can improve the fit further by taking N larger. The reason is that the fit is only asymptotic, rather than exact, and $N = 20$ is not a large sample.

5.5.1 Solutions to Selected Exercises

Solution to Exercise 5.5.1. We aim to show that $(\hat{\theta}_N - \theta)/\operatorname{se}(\hat{\theta}_N)$ converges in distribution to a standard normal when $\sqrt{N}(\hat{\theta}_N - \theta) \xrightarrow{d} \mathcal{N}(0, v(\theta))$ and $\sqrt{N}\operatorname{se}(\hat{\theta}_N) \xrightarrow{p} \sqrt{v(\theta)}$. To see this, observe that

$$\frac{\hat{\theta}_N - \theta}{\operatorname{se}(\hat{\theta}_N)} = \xi_N \eta_N \quad \text{where} \quad \xi_N := \frac{\sqrt{N}(\hat{\theta}_N - \theta)}{\sqrt{v(\theta)}} \quad \text{and} \quad \eta_N := \frac{\sqrt{v(\theta)}}{\sqrt{N}\operatorname{se}(\hat{\theta}_N)}$$

Using the various rules for converges in probabilty and distribution (check them carefully—see facts 1.4.1 and 1.4.4) we obtain $\xi_N \xrightarrow{d} Z \sim \mathcal{N}(0,1)$ and $\eta_N \xrightarrow{p} 1$. Applying Slutsky's theorem (page 34) now gives $\xi_N \eta_N \xrightarrow{d} Z$, as was to be shown. □

Solution to Exercise 5.5.2. Fix $\alpha \in (0,1)$ and let $z \sim \mathcal{N}(0,1)$. In view of (1.14) on page 21, we have $\mathbb{P}\{|z| > z_{\alpha/2}\} = \alpha$. If H_0 is true, then $t_N(\mathbf{x}) \xrightarrow{d} z$ by assumption. Since $g(s) := |s|$ is continuous, fact 1.4.4 on page 34 implies that $|t_N(\mathbf{x})| \xrightarrow{d} |z|$. As a result, we have

$$\lim_{N\to\infty} \beta_N(\theta) = \lim_{N\to\infty} \mathbb{P}_\theta\{|t_N(\mathbf{x})| > z_{\alpha/2}\} = \mathbb{P}\{|z| > z_{\alpha/2}\} = \alpha$$

This confirms (5.18), and the exercise is done. □

Solution to Exercise 5.5.3. In §4.2.2 we showed that the sample standard deviation s_N defined in (4.2) is consistent for the standard deviation σ. Appealing to (1.30) on page 37 and fact 1.4.5 on page 34, we can see that

$$t_N := \sqrt{N}\left\{\frac{\bar{x}_N - \theta_0}{s_N}\right\} = \frac{\sigma}{s_N}\sqrt{N}\left\{\frac{\bar{x}_N - \theta_0}{\sigma}\right\}$$

is asymptotically standard normal whenever H_0 is true (i.e., $\theta_0 = \theta$). It follows that exercise 5.5.2 can be applied. In particular, we can state that for $z_{\alpha/2} := \Phi^{-1}(1 - \alpha/2)$, the test is asymptotically of size α. □

Chapter 6

Linear Least Squares

[roadmap]

6.1 Least Squares

[roadmap]

6.1.1 Multivariate Least Squares and ERM

Let's consider a multivariate version of the regression problem we studied in §4.6. Suppose we repeatedly observe a vector input $\mathbf{x}$ to a system, followed by a scalar output y. Both are random, and $\mathbf{x}$ takes values in $\mathbb{R}^K$. We assume that the input-output pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ we observe are all draws from some common joint distribution on $\mathbb{R}^{K+1}$. This distribution is unknown to us. Our aim is to predict new output values from observed input values. In particular, our problem for this chapter is to

$$\text{choose a function } f\colon \mathbb{R}^K \to \mathbb{R} \text{ such that } f(\mathbf{x}) \text{ is a good predictor of } y$$

Throughout this chapter, we will be using quadratic loss as our measure of "goodness", so our expected loss (risk) from given function f is

$$R(f) := \mathbb{E}\left[(y - f(\mathbf{x}))^2\right] \tag{6.1}$$

As we learned in chapter 3, the minimizer of the risk is $f^*(\mathbf{x}) = \mathbb{E}[y \,|\, \mathbf{x}]$. Since the underlying distribution is not known, we cannot compute this conditional expectation. Instead, we will use the principle of empirical risk minimization instead, replacing the risk function with the empirical risk before minimizing. In other words, we solve

$$\min_{f \in \mathcal{F}} \hat{R}(f) \quad \text{where} \quad \hat{R}(f) := \frac{1}{N} \sum_{n=1}^{N} (y_n - f(\mathbf{x}_n))^2$$

This is the general **least squares** problem. The function f is chosen from a set of candidate functions $\mathcal{F}$ mapping $\mathbb{R}^K$ into $\mathbb{R}$. As before, $\mathcal{F}$ is called the hypothesis space.

When we presented the theory of empirical risk minimization in §4.6, we assumed that the input-output pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ were independent of each other. If this is true, then, for fixed f, the scalar random variables $(y_n - f(\mathbf{x}_n))^2$ are also independent (fact 2.4.1 on page 72), and, by the law of large numbers,

$$\hat{R}(f) := \frac{1}{N} \sum_{n=1}^{N} (y_n - f(\mathbf{x}_n))^2 \xrightarrow{p} \mathbb{E}[(y - f(\mathbf{x}))^2] = R(f) \tag{6.2}$$

This gives a fundamental justification for the empirical risk minimization principle: for large N, the empirical risk and the true risk are close.

Let us note at this stage that (6.2) can hold under much weaker assumptions than independence. For example, (6.2) can hold when the input-output pairs form a time series (each n is a point in time), and correlation between the input-output pairs is present, provided that this correlation dies out sufficiently quickly over time. An extensive discussion of this ideas is given in chapter 8. For now, we will not make any particular assumptions. Just keep in mind that validity of (6.2) is a minimal requirement to justify the approach that we are taking.

Let's now turn to the hypothesis space $\mathcal{F}$. If we take $\mathcal{F}$ to be the set of *all* functions from $\mathbb{R}^K$ to $\mathbb{R}$, we will usually be able to make the empirical risk $\hat{R}(f)$ arbitrarily small by choosing a function f such that $y_n - f(\mathbf{x}_n)$ is very small for all n. However, as we discussed extensively in §4.6.2, this is not the same think as making the *risk* small, which is what we actually want to minimize. Thus, $\mathcal{F}$ must be restricted, and in this chapter we consider the case $\mathcal{F} = \mathcal{L}$, where $\mathcal{L}$ is the *linear* functions from $\mathbb{R}^K$ to $\mathbb{R}$. That is,

$$\mathcal{L} := \{ \text{ all functions } \ell \colon \mathbb{R}^K \to \mathbb{R}, \text{ where } \ell(\mathbf{x}) = \mathbf{b}'\mathbf{x} \text{ for some } \mathbf{b} \in \mathbb{R}^K \} \tag{6.3}$$

The problem we need to solve is then $\min_{\ell \in \mathcal{L}} \sum_{n=1}^{N} (y_n - \ell(\mathbf{x}_n))^2$, or, more simply,

$$\min_{\mathbf{b} \in \mathbb{R}^K} \sum_{n=1}^{N} (y_n - \mathbf{b}' \mathbf{x}_n)^2 \tag{6.4}$$

(The constant $1/N$ has been dropped since it does not affect the minimizer.) This is the multivariate version of (4.25) on page 141. Although our presentation is rather modern, the idea of choosing $\mathbf{b}$ to minimize (6.4) is intuitive, and this optimization problem has a very long tradition. It dates back at least as far as Carl Gauss's work on the orbital position of Celes, published in 1801.

One might well ask whether the choice $\mathcal{F} = \mathcal{L}$ is suitable for most problems we encounter. This is an excellent question. It may not be. However, setting $\mathcal{F} = \mathcal{L}$ allows us to obtain an analytical expression for the minimizer, which greatly simplifies computations. The derivation is in §6.1.2 below. Moreover, the technique has a very natural extension from $\mathcal{L}$ to a very broad class of functions, as described in §6.2.1.

6.1.2 The Least Squares Estimator

The next step is to solve (6.4). In fact, armed with our knowledge of overdetermined systems (see §3.2), we already have all the necessary tools. This will be more obvious after we switch to matrix notation. To do this, let

$$\mathbf{y} := \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}, \quad \mathbf{x}_n := \begin{pmatrix} x_{n1} \\ x_{n2} \\ \vdots \\ x_{nK} \end{pmatrix} = n\text{-th observation of all regressors} \tag{6.5}$$

and

$$\mathbf{X} := \begin{pmatrix} \mathbf{x}_1' \\ \mathbf{x}_2' \\ \vdots \\ \mathbf{x}_N' \end{pmatrix} :=: \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1K} \\ x_{21} & x_{22} & \cdots & x_{2K} \\ \vdots & \vdots & & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NK} \end{pmatrix} \tag{6.6}$$

We will use the notational convention $\text{col}_k(\mathbf{X}) :=$ the k-th column of $\mathbf{X}$. In other words, $\text{col}_k(\mathbf{X})$ is all observations of the k-th regressor. Throughout this chapter, we will always maintain the following assumption on $\mathbf{X}$, which is usually satisfied in applications unless you're doing something silly.

Assumption 6.1.1. $N > K$ and the matrix $\mathbf{X}$ is full column rank.

Let's transform the minimization problem (6.4) into matrix form. Note that, for any $\mathbf{b} \in \mathbb{R}^K$, we have

$$\mathbf{X}\mathbf{b} = \begin{pmatrix} \mathbf{x}_1'\mathbf{b} \\ \mathbf{x}_2'\mathbf{b} \\ \vdots \\ \mathbf{x}_N'\mathbf{b} \end{pmatrix} = \begin{pmatrix} \mathbf{b}'\mathbf{x}_1 \\ \mathbf{b}'\mathbf{x}_2 \\ \vdots \\ \mathbf{b}'\mathbf{x}_N \end{pmatrix}$$

Regarding the objective function in (6.4), with a little bit of effort, you will be able to verify that

$$\sum_{n=1}^{N} (y_n - \mathbf{b}'\mathbf{x}_n)^2 = \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2$$

Moreover, since increasing transforms don't affect minimizers (see §13.2), we have

$$\operatorname*{argmin}_{\mathbf{b} \in \mathbb{R}^K} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 = \operatorname*{argmin}_{\mathbf{b} \in \mathbb{R}^K} \|\mathbf{y} - \mathbf{X}\mathbf{b}\| \tag{6.7}$$

In summary, any solution to the right-hand side of (6.7) is a minimizer of (6.4) and vice versa. The significance is that we already know now to solve for the minimizer on the right-hand side of (6.7). By theorem 3.2.1 (page 91), the solution is

$$\hat{\boldsymbol{\beta}} := (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \tag{6.8}$$

Traditionally, this random vector $\hat{\boldsymbol{\beta}}$ is called the **least squares estimator**, or the **OLS estimator**. (Right now, this terminology doesn't fit well with our presentation, since we haven't really assumed that $\hat{\boldsymbol{\beta}}$ is an *estimator* of anything in particular. In chapter 7 we will add some parametric structure to the underlying model, and $\hat{\boldsymbol{\beta}}$ will become an estimator of an unknown parameter vector $\boldsymbol{\beta}$.)

6.1.3 Standard Notation

There's a fair bit of notation associated with linear least squares estimation. Let's try to collect it in one place. First, let $\mathbf{P}$ and $\mathbf{M}$ be the projection matrix and annihilator associated with $\mathbf{X}$, as defined on page 92. The vector $\mathbf{P}\mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}}$ is often denoted $\hat{\mathbf{y}}$, and called the **vector of fitted values**:

$$\hat{\mathbf{y}} := \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{P}\mathbf{y}$$

The n-th fitted value $\hat{y}_n$ is the prediction $\mathbf{x}_n' \hat{\boldsymbol{\beta}}$ associated with OLS estimate and the n-th observation $\mathbf{x}_n$ of the input vector.

The vector $\mathbf{My}$ is often denoted $\hat{\mathbf{u}}$, and called the **vector of residuals**:

$$\hat{\mathbf{u}} := \mathbf{My} = \mathbf{y} - \hat{\mathbf{y}}$$

The vector of residuals corresponds to the error that occurs when $\mathbf{y}$ is approximated by its orthogonal projection $\mathbf{Py}$. From theorem 3.1.4 on page 89 we have

$$\mathbf{My} \perp \mathbf{Py} \quad \text{and} \quad \mathbf{y} = \mathbf{Py} + \mathbf{My} \tag{6.9}$$

In other words, $\mathbf{y}$ can be decomposed into two orthogonal vectors $\mathbf{Py}$ and $\mathbf{My}$, where the first represents the best approximation to $\mathbf{y}$ in $\text{rng}(\mathbf{X})$, and the second represents the error.

Related to the fitted values and residuals, we have some standard definitions:

- **Total sum of squares** $:=: \text{TSS} := \|\mathbf{y}\|^2$.
- **Sum of squared residuals** $:=: \text{SSR} := \|\mathbf{My}\|^2$.
- **Explained sum of squares** $:=: \text{ESS} := \|\mathbf{Py}\|^2$.

By (6.9) and the Pythagorean law (page 84) we have the following fundamental relation:

$$\text{TSS} = \text{ESS} + \text{SSR} \tag{6.10}$$

6.2 Transformations of the Data

[roadmap]

6.2.1 Basis Functions

Let's now revisit the decision to set $\mathcal{F} = \mathcal{L}$ made in §6.1.1. As we saw in §4.6.2, the choice of $\mathcal{F}$ is crucial. In that section, we considered data generated by the model

$$y = \cos(\pi x) + u \quad \text{where} \quad u \sim N(0, 1) \tag{6.11}$$

We imagined that this model was unknown to us, and attempted to minimize risk (expected quadratic loss) by minimizing empirical risk with different choices of hypothesis space $\mathcal{F}$. We saw that if $\mathcal{F}$ is too small, then no function in $\mathcal{F}$ provides a good fit to the model, and both the empirical risk and the risk are large (figure 4.18 on page 145). This is called underfitting. Conversely, if $\mathcal{F}$ is too big, then the empirical risk can be made small, but this risk itself is large (figure 4.21 on page 146). We are paying too much attention to one particular data set, causing overfitting.

As we learned in §3.3, in our quadratic loss setting, the minimizer of the risk is the conditional expectation of y given $\mathbf{x}$. Since our goal is to make the risk small, it would be best if $\mathcal{F}$ contained the conditional expectation. As our information is limited by the quantity of data, we would also like $\mathcal{F}$ to be small, so that the minimizer of the empirical risk has to be "close" to the risk minimizer $\mathbb{E}[y \mid \mathbf{x}]$. Of course choosing $\mathcal{F}$ in this way is not easy since $\mathbb{E}[y \mid \mathbf{x}]$ is unknown. The ideal case is where theory guides us, providing information about $\mathbb{E}[y \mid \mathbf{x}]$.

From such theory or perhaps from more primitive intuition, we may suspect that, for the problem at hand, the conditional expectation $\mathbb{E}[y \mid \mathbf{x}]$ is nonlinear. For example, many macroeconomic phenomena have a distinctly self-reinforcing flavor (poverty traps, dynamic network effects, deleveraging-induced debt deflation, etc.), and self-reinforcing dynamics are inherently nonlinear. This seems to suggest that setting $\mathcal{F} = \mathcal{L}$ is probably not appropriate.

Fortunately, it is easy to extend our previous analysis to a broad class of nonlinear functions. To do so, we first *transform* the data using some arbitrary nonlinear function $\boldsymbol{\phi} \colon \mathbb{R}^K \to \mathbb{R}^J$. The action of $\boldsymbol{\phi}$ on $\mathbf{x} \in \mathbb{R}^K$ is

$$\mathbf{x} \mapsto \boldsymbol{\phi}(\mathbf{x}) = \begin{pmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \vdots \\ \phi_J(\mathbf{x}) \end{pmatrix} \in \mathbb{R}^J$$

In this context, the individual functions $\phi_1, \ldots, \phi_J$ mapping $\mathbb{R}^K$ into $\mathbb{R}$ are referred to as **basis functions**. Now we apply linear least squares estimation to the transformed data. Formally, we choosing the hypothesis space to be

$$\mathcal{F}_{\boldsymbol{\phi}} := \{\text{all functions } \ell \circ \boldsymbol{\phi}, \text{ where } \ell \text{ is a linear function from } \mathbb{R}^J \text{ to } \mathbb{R}\}$$

The empirical risk minimization problem is then

$$\min_{\ell} \sum_{n=1}^{N} \{y_n - \ell(\boldsymbol{\phi}(\mathbf{x}_n))\}^2 = \min_{\gamma \in \mathbb{R}^J} \sum_{n=1}^{N} (y_n - \gamma' \boldsymbol{\phi}(\mathbf{x}_n))^2 \tag{6.12}$$

Switching to matrix notation, let

$$\boldsymbol{\phi}_n := \boldsymbol{\phi}(\mathbf{x}_n) \quad \text{and} \quad \boldsymbol{\Phi} := \begin{pmatrix} \boldsymbol{\phi}_1' \\ \boldsymbol{\phi}_2' \\ \vdots \\ \boldsymbol{\phi}_N' \end{pmatrix} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \cdots & \phi_J(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \cdots & \phi_J(\mathbf{x}_2) \\ \vdots & \cdots & \vdots \\ \phi_1(\mathbf{x}_N) & \cdots & \phi_J(\mathbf{x}_N) \end{pmatrix} \in \mathbb{R}^{N \times J}$$

In transforming (6.12) into matrix form, the objective function can be expressed as

$$\sum_{n=1}^{N} (y_n - \boldsymbol{\gamma}' \boldsymbol{\phi}_n)^2 = \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\gamma}\|^2$$

Once again, increasing functions don't affect minimizers, and the problem (6.12) becomes

$$\operatorname*{argmin}_{\boldsymbol{\gamma} \in \mathbb{R}^J} \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\gamma}\| \tag{6.13}$$

Assuming that $\boldsymbol{\Phi}$ is full column rank, the solution is

$$\hat{\boldsymbol{\gamma}} := (\boldsymbol{\Phi}'\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}'\mathbf{y}$$

Example 6.2.1. Suppose that $K = 1$, and x_n is scalar valued. Consider the mononomial basis functions $\phi_j(x) = x^{j-1}$, so that

$$\boldsymbol{\gamma}' \boldsymbol{\phi}(x_n) = \boldsymbol{\gamma}' \boldsymbol{\phi}_n = \boldsymbol{\gamma}' \begin{pmatrix} x_n^0 \\ x_n^1 \\ \vdots \\ x_n^{J-1} \end{pmatrix} = \sum_{j=1}^{J} \gamma_j x_n^{j-1} \tag{6.14}$$

This case corresponds to univariate polynomial regression, as previously discussed in §4.6.2. There is a theorem proved by Karl Weierstrass in 1885 which states that, for any given continuous function f on a closed interval of $\mathbb{R}$, there exists a polynomial function g such that g is arbitrarily close to f. On an intuitive level, this means that if we take J large enough, the relationship in (6.14) is capable of representing pretty much any (one-dimensional) nonlinear relationship we want.

In most of this chapter, we return to the elementary case of regressing y on $\mathbf{x}$, rather than on some nonlinear transformation $\boldsymbol{\phi}(\mathbf{x})$. However, no loss of generality is entailed, as we can just imagine that the data has already been transformed, and $\mathbf{x}$ is the result. Similarly, we'll use $\mathbf{X}$ to denote the data matrix instead of Φ, and $\hat{\boldsymbol{\beta}}$ to denote the least squares estimator $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$.

6.2.2 The Intercept

There's one special transformation that is worth treating in more detail: Adding an intercept to our regression. To add an intercept, we use the transformation

$$\boldsymbol{\phi}(\mathbf{x}) = \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix} = \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_K \end{pmatrix}$$

We'll call the resulting matrix $\mathbf{X}$ instead of $\boldsymbol{\Phi}$. (As mentioned at the end of the last section, we're going to use $\mathbf{X}$ to denote the data matrix from now on, even though the data it contains may have been subject to some transformations. What this means in the present case is that the first column of $\mathbf{X}$ is now $\mathbf{1}$, and each of the remaining columns of $\mathbf{X}$ contains N observations on one of the non-constant regressors.) In practice, adding an intercept means fitting an extra parameter, and this extra degree of freedom allows a more flexible fit in our regression.

One consequence of adding an intercept is that the vector of residuals must sum to zero. To see why this is the case, observe that

$$\mathbf{1}'\mathbf{M}\mathbf{y} = \mathbf{1}'(\mathbf{I} - \mathbf{P})\mathbf{y} = \mathbf{1}'\mathbf{y} - \mathbf{1}'\mathbf{P}\mathbf{y} = \mathbf{1}'\mathbf{y} - (\mathbf{P}'\mathbf{1})'\mathbf{y} = \mathbf{1}'\mathbf{y} - (\mathbf{P}\mathbf{1})'\mathbf{y}$$

where the last equality uses the fact the $\mathbf{P}$ is symmetric (exercise 3.4.10). Moreover, as exercise 6.4.5 asks you to confirm, we have $\mathbf{P}\mathbf{1} = \mathbf{1}$ whenever $\mathbf{1} \in \operatorname{rng}(\mathbf{X})$. Clearly $\mathbf{1} \in \operatorname{rng}(\mathbf{X})$ holds, since $\mathbf{1}$ is a column vector of $\mathbf{X}$.[1] Therefore,

$$\mathbf{1}'\mathbf{M}\mathbf{y} = \mathbf{1}'\mathbf{y} - (\mathbf{P}\mathbf{1})'\mathbf{y} = \mathbf{1}'\mathbf{y} - \mathbf{1}'\mathbf{y} = 0$$

In other words, the vector of residuals sums to zero.

It's also the case that if the regression contains the intercept, then the mean of the fitted values $\hat{\mathbf{y}} = \mathbf{P}\mathbf{y}$ is equal to the mean of $\mathbf{y}$. This follows from the previous argument, because we now have

$$\frac{1}{N}\sum_{n=1}^{N} \hat{y}_n = \frac{1}{N}\mathbf{1}'\hat{\mathbf{y}} = \frac{1}{N}\mathbf{1}'\mathbf{P}\mathbf{y} = \frac{1}{N}\mathbf{1}'\mathbf{y} = \frac{1}{N}\sum_{n=1}^{N} y_n$$

[1] Remember that $\operatorname{rng}(\mathbf{X})$ is the span of the columns of $\mathbf{X}$, and clearly each column is in the span.

6.3 Goodness of Fit

In traditional econometrics, goodness of fit refers to the in-sample fit of the model to the data (i.e., how well the model fits the observed data, as opposed to the potentially more important question of how well the model would predict new y from new $\mathbf{x}$). The most common measure of goodness of fit is the **coefficient of determination**. It is usually represented by the symbol R^2 (read "R squared"), and defined as

$$R^2 := \frac{\text{ESS}}{\text{TSS}} = \frac{\|\mathbf{P}\mathbf{y}\|^2}{\|\mathbf{y}\|^2}$$

For any $\mathbf{X}$ and $\mathbf{y}$ we have $0 \leq R^2 \leq 1$. The nontrivial inequality $R^2 \leq 1$ follows from the fact that for $\mathbf{P}$ and any point $\mathbf{y}$, we always have $\|\mathbf{P}\mathbf{y}\| \leq \|\mathbf{y}\|$. This was discussed in theorem 3.1.3 (page 86), and the geometric intuition can be seen in figure 3.4. The closer is $\mathbf{y}$ to the subspace $\text{rng}(\mathbf{X})$ that $\mathbf{P}$ projects onto, the closer $\|\mathbf{P}\mathbf{y}\| / \|\mathbf{y}\|$ will be to one. An extreme case is when $R^2 = 1$, a so-called **perfect fit**. If $R^2 = 1$, then, as exercise 6.4.7 asks you to verify, we must have $\mathbf{P}\mathbf{y} = \mathbf{y}$ and $\mathbf{y} \in \text{rng}(\mathbf{X})$.

Historically, R^2 has often been viewed as a one-number summary of the success of a regression model. As many people have noted, there are all sorts of problems with viewing R^2 in this way. In this section we cover some of the issues.

6.3.1 R Squared and Empirical Risk

One issue that arises when we equate high R^2 with successful regression is that we can usually make R^2 arbitrarily close to one in a way that nobody would consider good science: By putting as many regressors as we can think of into our regression. Intuitively, as we add regressors we increase the column space of $\mathbf{X}$, expanding it out towards $\mathbf{y}$, and hence increasings R^2. Put differently, the larger the column space $\mathbf{X}$, the better we can approximate a given vector $\mathbf{y}$ with an element of that column space.

To see this more formally, consider two groups of regressors, $\mathbf{X}_a$ and $\mathbf{X}_b$. We assume that $\mathbf{X}_b$ is larger, in the sense that every column of $\mathbf{X}_a$ is also a column of $\mathbf{X}_b$. Let $\mathbf{P}_a$ and $\mathbf{P}_b$ be the projection matrices corresponding to $\mathbf{X}_a$ and $\mathbf{X}_b$ respectively. Let $\mathbf{y}$ be given, and let R_a^2 and R_b^2 be the respective R squareds:

$$R_a^2 := \frac{\|\mathbf{P}_a\mathbf{y}\|^2}{\|\mathbf{y}\|^2} \quad \text{and} \quad R_b^2 := \frac{\|\mathbf{P}_b\mathbf{y}\|^2}{\|\mathbf{y}\|^2}$$

Since $\mathbf{X}_b$ is larger than $\mathbf{X}_a$, it follows that the column space of $\mathbf{X}_a$ is contained in the column space of $\mathbf{X}_b$ (exercise 6.4.11). It then follows from fact 3.1.2 on page 88 that $\mathbf{P}_a\mathbf{P}_b\mathbf{y} = \mathbf{P}_a\mathbf{y}$. Using this fact, and setting $\mathbf{y}_b := \mathbf{P}_b\mathbf{y}$, we obtain

$$\frac{R_a^2}{R_b^2} = \left(\frac{\|\mathbf{P}_a\mathbf{y}\|}{\|\mathbf{P}_b\mathbf{y}\|}\right)^2 = \left(\frac{\|\mathbf{P}_a\mathbf{P}_b\mathbf{y}\|}{\|\mathbf{P}_b\mathbf{y}\|}\right)^2 = \left(\frac{\|\mathbf{P}_a\mathbf{y}_b\|}{\|\mathbf{y}_b\|}\right)^2 \leq 1$$

where the final inequality follows from theorem 3.1.3 on page 86. Hence $R_b^2 \geq R_a^2$, and regressing with $\mathbf{X}_b$ produces (weakly) larger R^2.

Let's look at this phenomenon from a more statistical perspective. There is a close connection between R^2 and empirical risk. (See (4.21) on page 140 for the definition of the latter.) The R^2 of a regression of $\mathbf{y}$ on $\mathbf{X}$ can be written as $R^2 = 1 - \text{SSR}/\|\mathbf{y}\|^2$. If we alter the regressors, we change $\text{SSR} = \|\mathbf{M}\mathbf{y}\|^2$ while leaving other terms constant. In particular, if we raise R^2 by including more regressors, then the increase in R^2 occurs because SSR is falling. Since

$$\text{SSR} = \|\mathbf{M}\mathbf{y}\|^2 = \sum_{n=1}^{N}(y_n - \hat{\boldsymbol{\beta}}'\mathbf{x}_n)^2$$

we see that SSR is proportional to the empirical risk

$$\hat{R}(f) = \frac{1}{N}\sum_{n=1}^{N}(y_n - f(\mathbf{x}_n))^2$$

of our fitted function $f(\mathbf{x}) = \hat{\boldsymbol{\beta}}'\mathbf{x}$. Thus, the increase in R^2 is due to a fall in empirical risk. If we can drive the empirical risk to zero, then $\text{SSR} = 0$, $R^2 = 1$ and we have a perfect fit.

We can understand what is happening by recalling our discussion of empirical risk minimization in §4.6.2. As we discussed, when we use empirical risk minimization, our true goal is to minimize risk, rather than empirical risk. Empirical risk is just proxy for risk, the latter being unobservable. Moreover, if we allow ourselves unlimited flexibility in fitting functional relationships, we can make the empirical risk arbitrarily small, but this does not guarantee small risk. In fact, this excess of attention to the data set we have in hand often causes the risk to explode. Such an outcome was presented in figure 4.17 on page 143.

Let's look at a small simulation that illustrates the idea. Suppose that x_n and y_n are draws from a uniform distribution on $[0, 1]$. We will draw x_n and y_n independently,

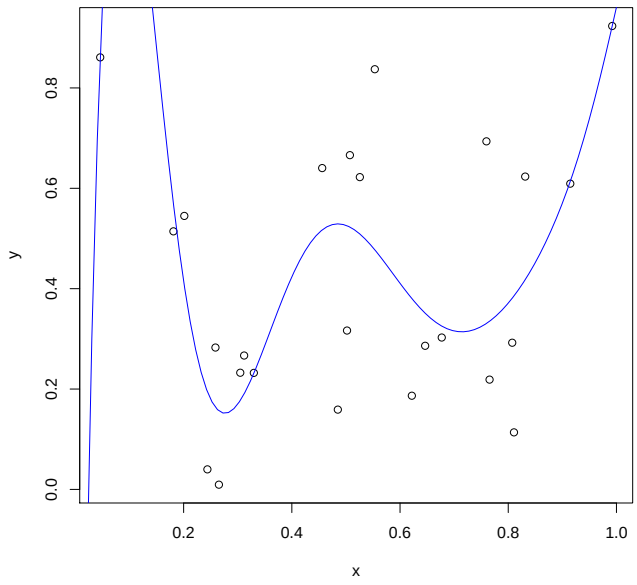


Figure 6.1: Minimizing empirical risk on independent data

so there is no relationship at all between these two variables. We will fit the polynomial $\beta_1 + \beta_2 x + \beta_3 x^2 + \cdots \beta_K x^{K-1}$ to the data for successively larger K. In other words, we will regress $\mathbf{y}$ on $\mathbf{X} = (x_n^{k-1})_{k,n}$ where $n = 1, \ldots, N$ and $k = 1, \ldots, K$. Incrementing K by one corresponds to one additional regressor.

A simulation and plot is shown in figure 6.1. Here $K = 8$, so we are fitting a relatively flexible polynomial. Since x and y are independent, the best guess of y given x is just the mean of y, which is 0.5. Nevertheless, the polynomial minimizes empirical risk by getting close to the sample points *in this particular draw of observations*. This reduces SSR and increases the R^2. Indeed, for this regression, the R^2 was 0.87. Increasing K to 25, I obtained an R^2 of 0.95. (The code is given in listing 8.) By this measure, the regression is very successful, even though we know there is actually no relationship whatsoever between x and y.

6.3.2 Centered R squared

Another issue with R^2 is that it is not invariant to certain kinds of changes of units. This problem is easily rectified, by using the so-called centered R^2 in place of R^2.

Listing 8 Generating the sequence of R^2

```
set.seed(1234)
N <- 25
y <- runif(N)
x <- runif(N)
X <- rep(1, N)

KMAX <- 25
for (K in 1:KMAX) {
    X <- cbind(X, x^K)
    results <- lm(y ~ 0 + X)
    Py2 <- sum(results$fitted.values^2)
    y2 <- sum(y^2)
    cat("K =", K, "R^2 =", Py2 / y2, "\n")
}
```

The main ideas are presented below.

First, R^2 is invariant to changes of units that involve rescaling of the regressand $\mathbf{y}$ (dollars versus cents, kilometers versus miles, etc.) because if $\mathbf{y}$ is scaled by $\alpha \in \mathbb{R}$, then

$$\frac{\|\mathbf{P}\alpha\mathbf{y}\|^2}{\|\alpha\mathbf{y}\|^2} = \frac{\|\alpha\mathbf{P}\mathbf{y}\|^2}{\|\alpha\mathbf{y}\|^2} = \frac{\alpha^2\|\mathbf{P}\mathbf{y}\|^2}{\alpha^2\|\mathbf{y}\|^2} = \frac{\|\mathbf{P}\mathbf{y}\|^2}{\|\mathbf{y}\|^2}$$

On the other hand, when the regression contains an intercept, R^2 is not invariant to changes of units that involve addition or subtraction (actual inflation versus inflation in excess of a certain level, income versus income over a certain threshold, etc.). To see this, let's compare the R^2 associated with $\mathbf{y}$ with that associated with $\mathbf{y} + \alpha\mathbf{1}$, where $\alpha \in \mathbb{R}$. The R^2 in the latter case is

$$\frac{\|\mathbf{P}(\mathbf{y}+\alpha\mathbf{1})\|^2}{\|\mathbf{y}+\alpha\mathbf{1}\|^2} = \frac{\|\mathbf{P}\mathbf{y}+\alpha\mathbf{P}\mathbf{1}\|^2}{\|\mathbf{y}+\alpha\mathbf{1}\|^2} = \frac{\|\mathbf{P}\mathbf{y}+\alpha\mathbf{1}\|^2}{\|\mathbf{y}+\alpha\mathbf{1}\|^2} = \frac{\alpha^2\|\mathbf{P}\mathbf{y}/\alpha+\mathbf{1}\|^2}{\alpha^2\|\mathbf{y}/\alpha+\mathbf{1}\|^2} = \frac{\|\mathbf{P}\mathbf{y}/\alpha+\mathbf{1}\|^2}{\|\mathbf{y}/\alpha+\mathbf{1}\|^2}$$

where the second inequality follows from the fact that $\mathbf{1} \in \text{rng}(\mathbf{X})$. Taking the limit as $\alpha \to \infty$, we find that the R squared converges to one. In other words, we can make the R squared as large as we like, just by a change of units.

For this reason, many economists and statisticians use the **centered R squared** rather than the R squared, at least when the regression contains an intercept. For the

purposes of this section, let's assume that this is the case (or, more generally, that $\mathbf{1} \in \text{rng}(\mathbf{X})$). The centered R squared is defined as

$$R_c^2 := \frac{\|\mathbf{P}\mathbf{M}_c\mathbf{y}\|^2}{\|\mathbf{M}_c\mathbf{y}\|^2} = \frac{\|\mathbf{M}_c\mathbf{P}\mathbf{y}\|^2}{\|\mathbf{M}_c\mathbf{y}\|^2} \tag{6.15}$$

where

$$\mathbf{M}_c := \mathbf{I} - \mathbf{1}(\mathbf{1}'\mathbf{1})^{-1}\mathbf{1}' = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}' \tag{6.16}$$

is the annihilator associated with $\mathbf{1}$. The equality of the two expressions for R_c^2 is left as an exercise (exercise 6.4.12). Hopefully it is clear that adding a constant to each element of $\mathbf{y}$ will have no effect on R_c^2.

The centered R squared can be re-written (exercise 6.4.13) as

$$R_c^2 = \frac{\sum_{n=1}^N (\hat{y}_n - \bar{y})^2}{\sum_{n=1}^N (y_n - \bar{y})^2} \tag{6.17}$$

It is a further exercise (exercise 6.4.16) to show that, in the case of the simple regression, the centered R squared is equal to the square of the sample correlation between the regressor and regressand, as defined in (4.4) on page 113. Thus, centered R squared can be thought of as a measure of correlation. As discussed below, correlation should not be confused with causation.

6.3.3 A Note on Causality

You have probably heard R^2 interpreted as measuring the "explanatory power" of the regressors in a particular regression. The idea is that regression amounts to decomposing $\mathbf{y}$ into the sum of two orthogonal parts, the fitted values $\mathbf{P}\mathbf{y}$ and the residuals $\mathbf{M}\mathbf{y}$. By the Pythagorean theorem, the squared norm $\|\mathbf{y}\|^2 =: \text{TSS}$ of $\mathbf{y}$ can then be represented $\|\mathbf{P}\mathbf{y}\|^2 =: \text{ESS}$ plus $\|\mathbf{M}\mathbf{y}\|^2 =: \text{SSR}$, as in (6.10). This is sometimes paraphrased as "the total variation in $\mathbf{y}$ is the sum of explained and unexplained variation." The value of R^2 is then claimed to be the fraction of the variation in $\mathbf{y}$ "explained" by the regressors.

This is a poor choice of terminology, because the notion that the regressors "explain" variation in $\mathbf{y}$ suggests causation, and R^2 says nothing about causation *per se*. Instead, R^2 is better thought of as a measure of correlation (see §6.3.2). As has been observed on many occasions, correlation and causation are not the same thing. Some informal examples are as follows:

- We often see car crashes and ambulances together (correlation). This does not imply that ambulances cause crashes.
- It has been observed that motorcycles fitted with ABS are less likely to be involved in accidents than those without ABS. Does that mean fitting ABS to a given motorcycle will reduce the probability that bike is involved in an accident? Perhaps, but another likely explanation is selection bias in the sample—cautious motorcyclists choose bikes with ABS, while crazy motorcyclists don't.
- Suppose we observe that people sleeping with their shoes on often wake up with headaches. One possible explanation is that that wearing shoes to bed causes headaches. A more likely explanation is that both phenomena are caused by too many pints at the pub the night before.

Identifying causality in statistical studies can be an extremely difficult problem. This is especially so in the social sciences, where properly controlled experiments are often costly or impossible to implement. (If we stand someone on a bridge and tell them to jump, are they more likely to do so? Try asking your national research body to fund that experiment.) An excellent starting point for learning more is Freedman (2009).

6.4 Exercises

Ex. 6.4.1. Argue that the sample mean of a random sample $y_1, \ldots, y_N$ from a given distribution F can be viewed as a least squares estimator of the mean of F.

Ex. 6.4.2. Let's show that $\hat{\boldsymbol{\beta}}$ solves the least squares problem in a slightly different way: Let $\mathbf{b}$ be any $K \times 1$ vector, and let $\hat{\boldsymbol{\beta}} := (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$.

1. Show that $\|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 = \|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\|^2 + \|\mathbf{X}(\hat{\boldsymbol{\beta}} - \mathbf{b})\|^2$.
2. Using this equality, argue that $\hat{\boldsymbol{\beta}}$ is the minimizer of $\|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2$ over all $K \times 1$ vectors $\mathbf{b}$.

Ex. 6.4.3. Verify that $\sum_{n=1}^{N}(y_n - \mathbf{b}'\mathbf{x}_n)^2 = \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2$.

Ex. 6.4.4. Show carefully that any solution to $\min_{\mathbf{b}\in\mathbb{R}^K} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2$ is also a solution to $\min_{\mathbf{b}\in\mathbb{R}^K} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|$, and vice versa.[2]

[2] You can use the ideas on optimization in the appendix, but provide your own careful argument. Make sure you use the *definition* of a minimizer in your argument.

Ex. 6.4.5. Explain why $\mathbf{P1} = \mathbf{1}$ whenever $\mathbf{1} \in \text{rng}(\mathbf{X})$.[3]

Ex. 6.4.6. Show that $\mathbf{PM} = \mathbf{MP} = \mathbf{0}$. Using this fact (and not the orthogonal projection theorem), show that the vector of fitted values and the vector of residuals are orthogonal.

Ex. 6.4.7. Show that if $R^2 = 1$, then $\mathbf{Py} = \mathbf{y}$ and $\mathbf{y} \in \text{rng}(\mathbf{X})$.

Ex. 6.4.8. Suppose that the regression contains an intercept, so that the first column of $\mathbf{X}$ is $\mathbf{1}$. Let $\bar{y}$ be the sample mean of $\mathbf{y}$, and let $\bar{\mathbf{x}}$ be a $1 \times K$ row vector such that the k-th element of $\bar{\mathbf{x}}$ is the sample mean of the k-th column of $\mathbf{X}$. Show that $\bar{y} = \bar{\mathbf{x}}\hat{\boldsymbol{\beta}}$.

Ex. 6.4.9. Show that if $R^2 = 1$, then every element of the vector of residuals is zero.

Ex. 6.4.10. Suppose the regression contains an intercept. Let $\mathbf{M}_c$ be as defined in (6.16). Show that

$$\|\mathbf{My}\|^2 = \|\mathbf{M}_c\mathbf{y}\|^2 - \|\mathbf{PM}_c\mathbf{y}\|^2 \tag{6.18}$$

always holds.[4]

Ex. 6.4.11. Let $\mathbf{X}_a$ and $\mathbf{X}_b$ be $N \times K_a$ and $N \times K_b$ respectively. Suppose that every column of $\mathbf{X}_a$ is also a column of $\mathbf{X}_b$. Show that $\text{rng}(\mathbf{X}_a) \subset \text{rng}(\mathbf{X}_b)$.

Ex. 6.4.12. Confirm the equality of the two alternative expressions for R_c^2 in (6.15).

Ex. 6.4.13. Verify the expression for R_c^2 in (6.17).

Ex. 6.4.14. Show that the coefficient of determination R^2 is invariant to a rescaling of the regressors (where all elements of the data matrix $\mathbf{X}$ are scaled by the same constant).

Ex. 6.4.15. Let $\mathbf{x} := (x_1, \ldots, x_N)$ and $\mathbf{y} := (y_1, \ldots, y_N)$ be sequences of scalar random variables. Show that the sample correlation $\hat{\rho}$ between $\mathbf{x}$ and $\mathbf{y}$ (defined in (4.4) on page 113) can be written as

$$\hat{\rho} = \frac{(\mathbf{M}_c\mathbf{x})'(\mathbf{M}_c\mathbf{y})}{\|\mathbf{M}_c\mathbf{x}\| \, \|\mathbf{M}_c\mathbf{y}\|}$$

Ex. 6.4.16. (Quite hard) Show that, in the case of the simple regression model (see §7.3.3), R_c^2 is equal to the square of the sample correlation between $\mathbf{x}$ and $\mathbf{y}$.

[3] Hint: Refresh your memory of theorem 3.1.3 on page 86.
[4] Hints: See fact 3.1.5 on page 90 and the Pythagorean law (page 84).

Ex. 6.4.17 (Computational)**.** Build an arbitrary data matrix $\mathbf{X}$ by simulation. Construct $\mathbf{M}$ and calculate $\mathbf{MX}$. The resulting matrix should have all entries very close to (but not exactly) zero. Test the orthogonality of $\mathbf{My}$ and $\mathbf{Py}$. The inner product should be very close to (but not exactly) zero.

Ex. 6.4.18 (Computational)**.** Build an arbitrary data set $\mathbf{X}$, $\mathbf{y}$ by simulation. Run a regression with the intercept, and record the values of the estimated coefficients of the non-constant (i.e., $k \geq 2$) regressors. Confirm that these values are equal to the estimated coefficients of the no-intercept regression after all variables have been centered around their mean.

Ex. 6.4.19 (Computational)**.** To be written: [R squared increases–see file rsquared.R]

Ex. 6.4.20 (Computational)**.** In §12.6.4, we discussed how the direct method of computing the OLS estimate via the expression $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ may be problematic in some settings. The difficulty is inverting the matrix $(\mathbf{X}'\mathbf{X})^{-1}$ when it is large and almost singular.[5] To see this in action, calculate the coefficients of (6.14) for successively higher and higher values of K, where each regression uses the same set of observations $\mathbf{x} = (x_1, \ldots, x_N)$ and $\mathbf{y} = (y_1, \ldots, y_N)$. (Example 6.2.1 explains how to estimate the coefficients using multiple regression.) Let $N = 20$ and let $\mathbf{x}$ be an evenly spaced grid on $[0, 1]$. Generate $\mathbf{y}$ as $y_n = x_n + \mathcal{N}(0, 1)$. In each round of the loop, calculate the regression coefficients using first `lm` and then the direct method (computing $(\mathbf{X}'\mathbf{X})^{-1}$ via the function `solve`). Set the program running in an infinite loop,[6] where each iteration, print the current degree of the polynomial and the coefficients from the two methods. You should find that the direct method fails first.

6.4.1 Solutions to Selected Exercises

Solution to Exercise 6.4.1. Letting $\mu := \mathbb{E}[y_n]$ we can write $y_n = \mu + u_n$ when $u_n := y_n - \mu$. In other words, $\mathbf{y} = \mathbf{1}\mu + \mathbf{u}$. The OLS estimate of μ is

$$\hat{\mu} := (\mathbf{1}'\mathbf{1})^{-1}\mathbf{1}'\mathbf{y} = \frac{1}{N}\mathbf{1}'\mathbf{y} = \frac{1}{N}\sum_{n=1}^{N} y_n = \bar{y}_N$$

Reading right to left, the sample mean of $\mathbf{y}$ is the OLS estimate of the mean. □

[5] In this setting, the inversion routine involves calculating many very small numbers. Since the amount of memory allocated to storing each of these numbers is fixed, the result of the calculations may be imprecise.

[6] To set up an infinite loop, start with `while(TRUE)`. To exist from an infinite loop running in the terminal, use control-C.

Solution to Exercise 6.4.2. Part 2 follows immediately from part 1. Regarding part 1, observe that

$$\|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 = \|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} + \mathbf{X}(\hat{\boldsymbol{\beta}} - \mathbf{b})\|^2$$

By the Pythagorean law, the claim

$$\|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 = \|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\|^2 + \|\mathbf{X}(\hat{\boldsymbol{\beta}} - \mathbf{b})\|^2$$

will be confirmed if $\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} \perp \mathbf{X}(\hat{\boldsymbol{\beta}} - \mathbf{b})$. This follows from the definition of $\hat{\boldsymbol{\beta}}$, because for arbitrary $\mathbf{a} \in \mathbb{R}^K$ we have

$$(\mathbf{a}\mathbf{X})'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \mathbf{a}'(\mathbf{X}'\mathbf{y} - \mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}) = \mathbf{a}'(\mathbf{X}'\mathbf{y} - \mathbf{X}'\mathbf{y}) = 0$$

□

Solution to Exercise 6.4.4. Let $\hat{\boldsymbol{\beta}}$ be a solution to

$$\min_{\mathbf{b} \in \mathbb{R}^K} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2$$

which is to say that

$$\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\|^2 \leq \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 \quad \text{for any } \mathbf{b} \in \mathbb{R}^K$$

If a and b and nonnegative constants with $a \leq b$, then $\sqrt{a} \leq \sqrt{b}$, and hence

$$\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\| \leq \|\mathbf{y} - \mathbf{X}\mathbf{b}\| \quad \text{for any } \mathbf{b} \in \mathbb{R}^K$$

In other words, $\hat{\boldsymbol{\beta}}$ is a solution to

$$\min_{\mathbf{b} \in \mathbb{R}^K} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|$$

The "vice versa" argument follows along similar lines. □

Solution to Exercise 6.4.7. If $R^2 = 1$, then, by (6.10) on page 179, we have $\|\mathbf{M}\mathbf{y}\|^2 = 0$, and hence have $\mathbf{M}\mathbf{y} = \mathbf{0}$. Since $\mathbf{M}\mathbf{y} + \mathbf{P}\mathbf{y} = \mathbf{y}$, this implies that $\mathbf{P}\mathbf{y} = \mathbf{y}$. But then $\mathbf{y} \in \text{rng}(\mathbf{X})$ by 5 of theorem 3.1.3. □

Solution to Exercise 6.4.9. If $R^2 = 1$, then $\|\mathbf{P}\mathbf{y}\|^2 = \|\mathbf{y}\|^2$, and hence

$$\|\mathbf{y}\|^2 = \|\mathbf{P}\mathbf{y}\|^2 + \|\mathbf{M}\mathbf{y}\|^2 = \|\mathbf{y}\|^2 + \|\mathbf{M}\mathbf{y}\|^2$$

Therefore $\|\mathbf{M}\mathbf{y}\|^2 = 0$, and, by fact 2.1.1 on page 52, $\mathbf{M}\mathbf{y} = \mathbf{0}$. □

Solution to Exercise 6.4.10. Theorem 3.1.4 tells us that for any conformable vector $\mathbf{z}$ we have $\mathbf{z} = \mathbf{P}\mathbf{z} + \mathbf{M}\mathbf{z}$. where the two vectors on the right-hand side are orthogonal. Letting $\mathbf{z} = \mathbf{M}_c\mathbf{y}$, we obtain

$$\mathbf{M}_c\mathbf{y} = \mathbf{P}\mathbf{M}_c\mathbf{y} + \mathbf{M}\mathbf{M}_c\mathbf{y}$$

From fact 3.1.5 we have $\mathbf{M}\mathbf{M}_c\mathbf{y} = \mathbf{M}\mathbf{y}$. Using this result, orthogonality and the Pythagorean law, we obtain

$$\|\mathbf{M}_c\mathbf{y}\|^2 = \|\mathbf{P}\mathbf{M}_c\mathbf{y}\|^2 + \|\mathbf{M}\mathbf{y}\|^2$$

Rearranging gives (6.18) □

Solution to Exercise 6.4.11. It suffices to show that if $\mathbf{z} \in \operatorname{rng}(\mathbf{X}_a)$, then $\mathbf{z} \in \operatorname{rng}(\mathbf{X}_b)$. Let $\mathbf{x}_1, \ldots, \mathbf{x}_J$ be the columns of $\mathbf{X}_a$ and let $\mathbf{x}_1, \ldots, \mathbf{x}_{J+M}$ be the columns of $\mathbf{X}_b$. If $\mathbf{z} \in \operatorname{rng}(\mathbf{X}_a)$, then

$$\mathbf{z} = \sum_{j=1}^{J} \alpha_j \mathbf{x}_j$$

for some scalars $\alpha_1, \ldots, \alpha_J$. But then

$$\mathbf{z} = \sum_{j=1}^{J} \alpha_j \mathbf{x}_j + \sum_{j=J+1}^{J+M} 0\,\mathbf{x}_j$$

In other words, $\mathbf{z} \in \operatorname{rng}(\mathbf{X}_b)$. □

Solution to Exercise 6.4.12. It is sufficient to show that

$$\mathbf{P}\mathbf{M}_c = \mathbf{M}_c\mathbf{P}$$

Since $\mathbf{1} \in \operatorname{rng}(\mathbf{X})$ by assumption, we have $\mathbf{P}\mathbf{1} = \mathbf{1}$, and $\mathbf{1}'\mathbf{P} = (\mathbf{P}'\mathbf{1})' = (\mathbf{P}\mathbf{1})' = \mathbf{1}'$. Therefore $\mathbf{P}\mathbf{1}\mathbf{1}' = \mathbf{1}\mathbf{1}'\mathbf{P}$, and

$$\mathbf{P}\mathbf{M}_c = \mathbf{P} - \frac{1}{N}\mathbf{P}\mathbf{1}\mathbf{1}' = \mathbf{P} - \frac{1}{N}\mathbf{1}\mathbf{1}'\mathbf{P} = \mathbf{M}_c\mathbf{P}$$

□

Solution to Exercise 6.4.13. It suffices to show that

$$\|\mathbf{P}\mathbf{M}_c\mathbf{y}\|^2 = \sum_{n=1}^{N} (\hat{y}_n - \bar{y})^2$$

This is the case because

$$\sum_{n=1}^{N}(\hat{y}_n - \bar{y})^2 = \|\mathbf{P}\mathbf{y} - \mathbf{1}\bar{y}\|^2 = \|\mathbf{P}\mathbf{y} - \mathbf{P}\mathbf{1}\bar{y}\|^2 = \|\mathbf{P}(\mathbf{y} - \mathbf{1}\bar{y})\|^2 = \|\mathbf{P}\mathbf{M}_c\mathbf{y}\|^2$$

□

Solution to Exercise 6.4.14. This follows immediately from the definition of R^2, and the fact that, for any $\alpha \neq 0$,

$$\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X} = \frac{\alpha^2}{\alpha^2}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X} = (\alpha\mathbf{X})((\alpha\mathbf{X})'(\alpha\mathbf{X}))^{-1}(\alpha\mathbf{X})$$

□

Solution to Exercise 6.4.16. From exercise 6.4.15, the squared sample correlation between $\mathbf{x}$ and $\mathbf{y}$ can be written as

$$\hat{\rho}^2 = \frac{[(\mathbf{M}_c\mathbf{x})'(\mathbf{M}_c\mathbf{y})]^2}{\|\mathbf{M}_c\mathbf{x}\|^2\|\mathbf{M}_c\mathbf{y}\|^2}$$

On the other hand, we have

$$R_c^2 = \frac{\|\mathbf{M}_c\mathbf{P}\mathbf{y}\|^2}{\|\mathbf{M}_c\mathbf{y}\|^2}$$

Therefore it suffices to show that, for the simple linear regression model in §7.3.3, we have

$$\|\mathbf{M}_c\mathbf{P}\mathbf{y}\| = \frac{|(\mathbf{M}_c\mathbf{x})'(\mathbf{M}_c\mathbf{y})|}{\|\mathbf{M}_c\mathbf{x}\|} \tag{6.19}$$

Let $\mathbf{X} = (\mathbf{1}, \mathbf{x})$ be the data matrix, where the first column is $\mathbf{1}$ and the second column is $\mathbf{x}$. Let

$$\hat{\beta}_1 := \bar{y} - \hat{\beta}_2\bar{x} \quad \text{and} \quad \hat{\beta}_2 := \frac{(\mathbf{M}_c\mathbf{x})'(\mathbf{M}_c\mathbf{y})}{\|\mathbf{M}_c\mathbf{x}\|^2}$$

be the OLS estimators of β_1 and β_2 respectively (see §7.3.3). We then have

$$\mathbf{P}\mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{1}\hat{\beta}_1 + \mathbf{x}\hat{\beta}_2$$

$$\therefore \quad \mathbf{M}_c\mathbf{P}\mathbf{y} = \mathbf{M}_c\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{M}_c\mathbf{x}\hat{\beta}_2$$

$$\therefore \quad \|\mathbf{M}_c\mathbf{P}\mathbf{y}\| = \|\mathbf{M}_c\mathbf{x}\hat{\beta}_2\| = |\hat{\beta}_2|\|\mathbf{M}_c\mathbf{x}\| = \frac{|(\mathbf{M}_c\mathbf{x})'(\mathbf{M}_c\mathbf{y})|}{\|\mathbf{M}_c\mathbf{x}\|^2}\|\mathbf{M}_c\mathbf{x}\|$$

Cancelling $\|\mathbf{M}_c\mathbf{x}\|$ we get (6.19). This completes the proof. □

Part III

Econometric Models

Chapter 7

Classical OLS

In this chapter we continue our study of linear least squares and multivariate regression, as begun in chapter 6. To say more about whether linear least squares estimation is a "good" procedure or otherwise, we need to make more assumptions on the process that generates our data. Not surprisingly, for some sets of assumptions on the data generating process, the performance of linear least squares estimation is good, while for other assumptions the performance is poor. The main purpose of this chapter is to describe the performance of linear least squares estimation under the classical OLS assumptions, where OLS stands for **ordinary least squares**.

To some people (like me), the standard OLS assumptions are somewhat difficult to swallow. At the same time, the results obtained from these assumptions form the bread and butter of econometrics. As such, they need to be understood.

7.1 The Model

[roadmap]

7.1.1 The OLS Assumptions

In chapter 6 we assumed that the observed input-output pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ were draws from some common joint distribution on $\mathbb{R}^{K+1}$. To work with the classical OLS model, we have to impose (much!) more structure. For starters, we will

assume that the input-output pairs all satisfy

$$y_n = \boldsymbol{\beta}' \mathbf{x}_n + u_n \tag{7.1}$$

where $\boldsymbol{\beta}$ is an unknown $K \times 1$ vector of parameters, and $u_1, \ldots, u_N$ are unobservable random variables. The model is called linear because the deterministic part of the output value $\boldsymbol{\beta}' \mathbf{x}$ is linear as a function of $\mathbf{x}$. Letting $\mathbf{u} := (u_1, u_2, \ldots, u_N)'$ be the column vector formed by the N realizations of the shock, the N equations in (7.1) can be expressed as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u} \tag{7.2}$$

Here are three very traditional examples:

Example 7.1.1. The Cobb-Douglas production function relates capital and labor inputs with output via $y = Ak^\gamma \ell^\delta$ where A is a random, firm-specific productivity term and γ and δ are parameters. Taking logs yields the linear regression model

$$\ln y = \beta + \gamma \ln k + \delta \ln \ell + u$$

where the random term $\ln A$ is represented by $\beta + u$.

Example 7.1.2. The elementary Keynesian consumption function is of the form $C = \beta_1 + \beta_2 D$, where C is consumption and D is household disposable income. Letting $\boldsymbol{\beta} := (\beta_1, \beta_2)$ and $\mathbf{x} := (1, D)$ and adding an error term u, we obtain the linear regression model $C = \boldsymbol{\beta}' \mathbf{x} + u$.

Example 7.1.3. Okun's law is an empirical rule of thumb relating output to unemployment. A commonly estimated form of Okun's law is $U = \beta_1 + \beta_2 G + u$, where G is the growth rate of GDP over a given period, and U is the change in the unemployment rate over the same period (i.e., end value minus starting value). The parameter β_2 is expected to be negative. The value $-\beta_1/\beta_2$ is thought of as the rate of GDP growth necessary to maintain a stable unemployment rate.

Let $\mathbf{M}$ be the annihilator associated with $\mathbf{X}$. The proof of the next fact is an exercise (exercise 7.6.1).

Fact 7.1.1. When (7.2) holds we have $\mathbf{My} = \mathbf{Mu}$ and $\text{SSR} = \mathbf{u}'\mathbf{Mu}$.

Regarding the shocks $u_1, \ldots, u_N$, the classical OLS model makes two assumptions, one concerning the first moments, and the other concerning the variance. The first assumption is as follows:

Assumption 7.1.1. Together, $\mathbf{u}$ and $\mathbf{X}$ satisfy $\mathbb{E}[\mathbf{u} \mid \mathbf{X}] = \mathbf{0}$.

This assumption expresses a number of ideas in very compact form. Important details are recorded in the next fact, the proof of which is an exercise (exercise 7.6.2).

Fact 7.1.2. Given assumption 7.1.1, we have:

1. $\mathbb{E}[\mathbf{u}] = \mathbf{0}$.
2. $\mathbb{E}[u_m \mid x_{nk}] = 0$ for any m, n, k.
3. $\mathbb{E}[u_m x_{nk}] = 0$ for any m, n, k.
4. $\text{cov}[u_m, x_{nk}] = 0$ for any m, n, k.

To consider the variance of $\hat{\boldsymbol{\beta}}$, we will need an assumption on second moments of the shock. The most standard assumption is as follows:

Assumption 7.1.2. Together, $\mathbf{u}$ and $\mathbf{X}$ satisfy $\mathbb{E}[\mathbf{u}\mathbf{u}' \mid \mathbf{X}] = \sigma^2 \mathbf{I}$, where σ is a positive constant.

The value of the parameter σ is unknown, in the same sense that the vector of coefficients $\boldsymbol{\beta}$ is unknown. To understand assumption 7.1.2, note that

$$\text{var}[\mathbf{u} \mid \mathbf{X}] := \mathbb{E}[\mathbf{u}\mathbf{u}' \mid \mathbf{X}] - \mathbb{E}[\mathbf{u} \mid \mathbf{X}]\mathbb{E}[\mathbf{u}' \mid \mathbf{X}] = \mathbb{E}[\mathbf{u}\mathbf{u}' \mid \mathbf{X}]$$

where the second equality is due to assumption 7.1.1. Hence, assumption 7.1.2 implies that the conditional variance-covariance matrix is the constant matrix $\sigma^2 \mathbf{I}$.

Fact 7.1.3. Given assumption 7.1.2, we have the following results:

1. $\text{var}[\mathbf{u}] = \mathbb{E}[\mathbf{u}\mathbf{u}'] = \sigma^2 \mathbf{I}$.
2. Shocks are **homoskedastic**: $\mathbb{E}[u_i^2 \mid \mathbf{X}] = \mathbb{E}[u_j^2 \mid \mathbf{X}] = \sigma^2$ for any i, j in $1, \ldots, N$.
3. Distinct shocks are uncorrelated: $\mathbb{E}[u_i u_j \mid \mathbf{X}] = 0$ whenever $i \neq j$.

7.1.2 The OLS Estimators

The standard estimator of the unknown parameter vector $\boldsymbol{\beta}$ is the estimator $\hat{\boldsymbol{\beta}}$ defined in (6.8). To repeat:

$$\hat{\boldsymbol{\beta}} := (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \tag{7.3}$$

The standard estimator of the parameter σ^2 introduced in assumption 7.1.2 is

$$\hat{\sigma}^2 := \frac{\text{SSR}}{N-K}$$

We will show below that, under the classical OLS assumptions given in §7.1.1, both estimators have nice properties. As a precursor to the arguments, note that, applying (7.2), we obtain

$$\hat{\boldsymbol{\beta}} - \boldsymbol{\beta} := (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} - \boldsymbol{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) - \boldsymbol{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}$$

This deviation is known as the **sampling error** of $\hat{\boldsymbol{\beta}}$. Adding $\boldsymbol{\beta}$ to both sides yields the useful expression

$$\hat{\boldsymbol{\beta}} = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u} \tag{7.4}$$

7.2 Variance and Bias

In this section we will investigate the properties of the estimators $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ under the standard OLS assumptions.

7.2.1 Bias

Under the linearity assumption $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u}$ and the standard assumptions on the shock, $\hat{\boldsymbol{\beta}}$ is an unbiased estimator of $\boldsymbol{\beta}$, and $\hat{\sigma}^2$ is an unbiased estimator of σ^2:

Theorem 7.2.1. *Under assumption 7.1.1, we have* $\mathbb{E}[\hat{\boldsymbol{\beta}}] = \mathbb{E}[\hat{\boldsymbol{\beta}} \mid \mathbf{X}] = \boldsymbol{\beta}$.

Theorem 7.2.2. *Under assumptions 7.1.1–7.1.2, we have* $\mathbb{E}[\hat{\sigma}^2] = \mathbb{E}[\hat{\sigma}^2 \mid \mathbf{X}] = \sigma^2$.

Proof of theorem 7.2.1. By (7.4) and assumption 7.1.1, we have

$$\mathbb{E}[\hat{\boldsymbol{\beta}} \mid \mathbf{X}] = \boldsymbol{\beta} + \mathbb{E}[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u} \mid \mathbf{X}] = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbb{E}[\mathbf{u} \mid \mathbf{X}] = \boldsymbol{\beta}$$

This proves $\mathbb{E}[\hat{\boldsymbol{\beta}} \mid \mathbf{X}] = \boldsymbol{\beta}$, and hence $\mathbb{E}[\hat{\boldsymbol{\beta}}] = \mathbb{E}[\mathbb{E}[\hat{\boldsymbol{\beta}} \mid \mathbf{X}]] = \mathbb{E}[\boldsymbol{\beta}] = \boldsymbol{\beta}$. □

Proof of theorem 7.2.2. For reasons that will become apparent, we start the proof by showing that $\text{trace}(\mathbf{M}) = N - K$. To see that this is so, observe that, recalling fact 2.3.8,

$$\text{trace}(\mathbf{P}) = \text{trace}[\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'] = \text{trace}[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}] = \text{trace}[\mathbf{I}_K] = K$$

$$\therefore \quad \text{trace}(\mathbf{M}) = \text{trace}(\mathbf{I}_N - \mathbf{P}) = \text{trace}(\mathbf{I}_N) - \text{trace}(\mathbf{P}) = N - K$$

Now let $m_{ij}(\mathbf{X})$ be the i, j-th element of $\mathbf{M}$. Applying fact 7.1.1 on page 196, we have

$$\mathbb{E}\left[\text{SSR} \mid \mathbf{X}\right] = \mathbb{E}\left[\mathbf{u}'\mathbf{M}\mathbf{u} \mid \mathbf{X}\right] = \mathbb{E}\left[\sum_{i=1}^{N}\sum_{j=1}^{N} u_i u_j m_{ij}(\mathbf{X}) \mid \mathbf{X}\right] = \sum_{i=1}^{N}\sum_{j=1}^{N} m_{ij}(\mathbf{X})\mathbb{E}\left[u_i u_j \mid \mathbf{X}\right]$$

In view of assumption 7.1.2, this reduces to

$$\sum_{n=1}^{N} m_{nn}(\mathbf{X})\sigma^2 = \text{trace}(\mathbf{M})\sigma^2 = (N-K)\sigma^2$$

Hence $\mathbb{E}\left[\hat{\sigma}^2 \mid \mathbf{X}\right] = \sigma^2$, and $\mathbb{E}\left[\hat{\sigma}^2\right] = \mathbb{E}\left[\mathbb{E}\left[\hat{\sigma}^2 \mid \mathbf{X}\right]\right] = \mathbb{E}\left[\sigma^2\right] = \sigma^2$. □

7.2.2 Variance of $\hat{\boldsymbol{\beta}}$

Now that $\hat{\boldsymbol{\beta}}$ is known to be unbiased, we want to say something about the variance.

Theorem 7.2.3. *Under assumptions 7.1.1–7.1.2, we have* $\text{var}[\hat{\boldsymbol{\beta}} \mid \mathbf{X}] = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$.

Proof. If $\mathbf{A} := (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$, then $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta} + \mathbf{A}\mathbf{u}$, and

$$\text{var}[\hat{\boldsymbol{\beta}} \mid \mathbf{X}] = \text{var}[\boldsymbol{\beta} + \mathbf{A}\mathbf{u} \mid \mathbf{X}] = \text{var}[\mathbf{A}\mathbf{u} \mid \mathbf{X}]$$

Since $\mathbf{A}$ is a function of $\mathbf{X}$, we can treat it as non-random given $\mathbf{X}$, and hence, by fact 2.4.4 on page 73, we have

$$\text{var}[\mathbf{A}\mathbf{u} \mid \mathbf{X}] = \mathbf{A}\,\text{var}[\mathbf{u} \mid \mathbf{X}]\mathbf{A}' = \mathbf{A}(\sigma^2\mathbf{I})\mathbf{A}'$$

Moreover,

$$\mathbf{A}(\sigma^2\mathbf{I})\mathbf{A}' = \sigma^2\mathbf{A}\mathbf{A}' = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$$

$$\therefore \quad \text{var}[\hat{\boldsymbol{\beta}} \mid \mathbf{X}] = \text{var}[\mathbf{A}\mathbf{u} \mid \mathbf{X}] = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$$

□

7.2.3 The Gauss-Markov Theorem

Now that $\hat{\boldsymbol{\beta}}$ is known to be unbiased for $\boldsymbol{\beta}$ under the OLS assumptions, the next step is to show that $\hat{\boldsymbol{\beta}}$ has low variance. Although we obtained an expression for the variance in theorem 7.2.3, it's not clear whether this is low or high. The natural way to answer this question is to compare the variance of $\hat{\boldsymbol{\beta}}$ with that of other unbiased estimators. This leads us to the famous Gauss-Markov theorem.

Theorem 7.2.4 (Gauss-Markov)**.** *If* $\mathbf{b}$ *is any other linear unbiased estimator of* $\boldsymbol{\beta}$*, then* $\operatorname{var}[\mathbf{b} \mid \mathbf{X}] \geq \operatorname{var}[\hat{\boldsymbol{\beta}} \mid \mathbf{X}]$*, in the sense that* $\operatorname{var}[\mathbf{b} \mid \mathbf{X}] - \operatorname{var}[\hat{\boldsymbol{\beta}} \mid \mathbf{X}]$ *is nonnegative definite.*

The theorem is often summarized by stating that $\hat{\boldsymbol{\beta}}$ is BLUE. BLUE stands for Best Linear Unbiased Estimator, and was discussed previously in §4.2.2.

There are a couple of points to clarify. First, the meaning of linearity: Although it's not immediately clear from the statement of the theorem, here linearity of $\mathbf{b}$ means that $\mathbf{b}$ is linear as a function of $\mathbf{y}$ (taking $\mathbf{X}$ as fixed). In view of theorem 2.1.1 on page 56, this is equivalent to requiring that $\mathbf{b} = \mathbf{C}\mathbf{y}$ for some matrix $\mathbf{C}$. The matrix $\mathbf{C}$ is allowed to depend on $\mathbf{X}$ (i.e., be a function of $\mathbf{X}$), but not $\mathbf{y}$.

Second, how to interpret the statement that $\operatorname{var}[\mathbf{b} \mid \mathbf{X}] - \operatorname{var}[\hat{\boldsymbol{\beta}} \mid \mathbf{X}]$ is positive definite? Matrices have no standard ordering, and hence it's hard to say when one random vector has "larger" variance than another. But nonnegative definiteness of the difference is a natural criterion. In particular, all elements of the principle diagonal of a nonnegative definite matrix are themselves nonnegative, so the implication is that $\operatorname{var}[b_k \mid \mathbf{X}] \geq \operatorname{var}[\hat{\beta}_k \mid \mathbf{X}]$ for all k.

Third, the meaning of unbiasedness: In this theorem, it means that, regardless of the value of $\boldsymbol{\beta}$ (i.e., for any $\boldsymbol{\beta} \in \mathbb{R}^K$), we have $\mathbb{E}[\mathbf{b} \mid \mathbf{X}] = \mathbb{E}[\mathbf{C}\mathbf{y} \mid \mathbf{X}] = \boldsymbol{\beta}$.

Proof of theorem 7.2.4. Let $\mathbf{b} = \mathbf{C}\mathbf{y}$, as described above, and let $\mathbf{D} := \mathbf{C} - \mathbf{A}$, where $\mathbf{A} := (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$. Then

$$\mathbf{b} = \mathbf{C}\mathbf{y} = \mathbf{D}\mathbf{y} + \mathbf{A}\mathbf{y} = \mathbf{D}(\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) + \hat{\boldsymbol{\beta}} = \mathbf{D}\mathbf{X}\boldsymbol{\beta} + \mathbf{D}\mathbf{u} + \hat{\boldsymbol{\beta}} \tag{7.5}$$

Taking conditional expectations and using the fact that $\mathbf{D}$ is a function of $\mathbf{X}$, we obtain

$$\begin{aligned}\mathbb{E}[\mathbf{b} \mid \mathbf{X}] &= \mathbb{E}[\mathbf{D}\mathbf{X}\boldsymbol{\beta} \mid \mathbf{X}] + \mathbb{E}[\mathbf{D}\mathbf{u} \mid \mathbf{X}] + \mathbb{E}[\hat{\boldsymbol{\beta}} \mid \mathbf{X}] \\ &= \mathbf{D}\mathbf{X}\,\mathbb{E}[\boldsymbol{\beta} \mid \mathbf{X}] + \mathbf{D}\,\mathbb{E}[\mathbf{u} \mid \mathbf{X}] + \mathbb{E}[\hat{\boldsymbol{\beta}} \mid \mathbf{X}] = \mathbf{D}\mathbf{X}\boldsymbol{\beta} + \mathbf{0} + \boldsymbol{\beta}\end{aligned}$$

In light of the fact that $\mathbf{b}$ is unbiased, and, in particular, $\mathbb{E}[\mathbf{b} \mid \mathbf{X}] = \boldsymbol{\beta}$ for any given $\boldsymbol{\beta}$, we have

$$\begin{aligned} \boldsymbol{\beta} &= \mathbf{D}\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta} \quad \text{for all } \boldsymbol{\beta} \in \mathbb{R}^K \\ \therefore \quad \mathbf{0} &= \mathbf{D}\mathbf{X}\boldsymbol{\beta} \quad \text{for all } \boldsymbol{\beta} \in \mathbb{R}^K \end{aligned}$$

In light of exercise 2.6.19 on page 80, we conclude that $\mathbf{D}\mathbf{X} = \mathbf{0}$. Combining this result with (7.5), we obtain the expression $\mathbf{b} = \mathbf{D}\mathbf{u} + \hat{\boldsymbol{\beta}}$. Roughly speaking, this says that $\mathbf{b}$ is equal to the OLS estimator plus some zero mean noise.

To complete the proof, observe that

$$\operatorname{var}[\mathbf{b} \mid \mathbf{X}] = \operatorname{var}[\mathbf{D}\mathbf{u} + \hat{\boldsymbol{\beta}} \mid \mathbf{X}] = \operatorname{var}[(\mathbf{D} + \mathbf{A})\mathbf{u} \mid \mathbf{X}] = (\mathbf{D} + \mathbf{A}) \operatorname{var}[\mathbf{u} \mid \mathbf{X}](\mathbf{D} + \mathbf{A})'$$

Using assumption 7.1.2 and fact 2.3.5, the right-hand side of this expression becomes

$$\sigma^2(\mathbf{D} + \mathbf{A})(\mathbf{D}' + \mathbf{A}') = \sigma^2(\mathbf{D}\mathbf{D}' + \mathbf{D}\mathbf{A}' + \mathbf{A}\mathbf{D}' + \mathbf{A}\mathbf{A}')$$

Since

$$\mathbf{D}\mathbf{A}' = \mathbf{D}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} = \mathbf{0}(\mathbf{X}'\mathbf{X})^{-1} = \mathbf{0}$$

and since

$$\mathbf{A}\mathbf{A}' = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} = (\mathbf{X}'\mathbf{X})^{-1}$$

we now conclude that

$$\operatorname{var}[\mathbf{b} \mid \mathbf{X}] = \sigma^2[\mathbf{D}\mathbf{D}' + (\mathbf{X}'\mathbf{X})^{-1}] = \sigma^2\mathbf{D}\mathbf{D}' + \operatorname{var}[\hat{\boldsymbol{\beta}} \mid \mathbf{X}]$$

Since $\sigma^2\mathbf{D}\mathbf{D}'$ is nonnegative definite (why?), the proof is now done. □

7.3 The FWL Theorem

The Frisch-Waugh-Lovell (FWL) Theorem yields, among other things, an explicit expression for an arbitrary sub-vector of the OLS estimator $\hat{\boldsymbol{\beta}}$. While that might not sound terribly exciting, it turns out to have many useful applications.

7.3.1 Statement and Proof

Continuing to work with our linear regression model, let's take $\mathbf{y}$ and $\mathbf{X}$ as given, implying an OLS estimate $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$. Recall that $\mathbf{y}$ can be decomposed as

$$\mathbf{y} = \mathbf{P}\mathbf{y} + \mathbf{M}\mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}} + \mathbf{M}\mathbf{y} \tag{7.6}$$

We can write (7.6) in a slightly different way by partitioning the regressors and the estimated coefficients into two classes: Let $\mathbf{X}_1$ be a matrix consisting of the first K_1 columns of $\mathbf{X}$, and let $\mathbf{X}_2$ be a matrix consisting of the remaining $K_2 := K - K_1$ columns of $\mathbf{X}$. Similarly, let

1. $\hat{\boldsymbol{\beta}}_1$ be the $K_1 \times 1$ vector consisting of the first K_1 elements of $\hat{\boldsymbol{\beta}}$, and
2. $\hat{\boldsymbol{\beta}}_2$ be the $K_2 \times 1$ vector consisting of the remaining K_2 elements of $\hat{\boldsymbol{\beta}}$.

We can then rewrite (7.6) as

$$\mathbf{y} = \mathbf{X}_1\hat{\boldsymbol{\beta}}_1 + \mathbf{X}_2\hat{\boldsymbol{\beta}}_2 + \mathbf{M}\mathbf{y} \tag{7.7}$$

Let $\mathbf{P}_1 := \mathbf{X}_1(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'$ be the projection onto the column space of $\mathbf{X}_1$, and let $\mathbf{M}_1 := \mathbf{I} - \mathbf{P}_1$ be the corresponding annihilator, projecting onto the orthogonal complement of the column space of $\mathbf{X}_1$. With this notation we have the following result:

Theorem 7.3.1 (FWL theorem)**.** *The $K_2 \times 1$ vector $\hat{\boldsymbol{\beta}}_2$ can be expressed as*

$$\hat{\boldsymbol{\beta}}_2 = (\mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2)^{-1}\mathbf{X}_2'\mathbf{M}_1\mathbf{y} \tag{7.8}$$

The theorem gives an explicit analytical expression for our arbitrarily chosen subset $\hat{\boldsymbol{\beta}}_2$ of the OLS estimate $\hat{\boldsymbol{\beta}}$. Before discussing it's implications, let's present the proof.

Proof of theorem 7.3.1. Premultiplying both sides of (7.7) by $\mathbf{X}_2'\mathbf{M}_1$, we obtain

$$\mathbf{X}_2'\mathbf{M}_1\mathbf{y} = \mathbf{X}_2'\mathbf{M}_1\mathbf{X}_1\hat{\boldsymbol{\beta}}_1 + \mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2\hat{\boldsymbol{\beta}}_2 + \mathbf{X}_2'\mathbf{M}_1\mathbf{M}\mathbf{y} \tag{7.9}$$

The first and last terms on the right-hand side are zero. This is clear for the first term, because $\mathbf{M}_1$ is the annihilator associated with $\mathbf{X}_1$. Hence $\mathbf{M}_1\mathbf{X}_1 = \mathbf{0}$. Regarding the last term, it suffices to show that the transpose of the term is $\mathbf{0}'$. To see this, observe that

$$(\mathbf{X}_2'\mathbf{M}_1\mathbf{M}\mathbf{y})' = \mathbf{y}'\mathbf{M}'\mathbf{M}_1'\mathbf{X}_2 = \mathbf{y}'\mathbf{M}\mathbf{M}_1\mathbf{X}_2 = \mathbf{y}'\mathbf{M}\mathbf{X}_2 = \mathbf{0}'$$

In the first equality we used the usual property of transposes (fact 2.3.5), in the second we used symmetry of $\mathbf{M}$ and $\mathbf{M}_1$ (exercise 3.4.10 or direct calculation), in the third we used fact 3.1.5 on page 90, and in the fourth we used the fact that $\mathbf{M}$ is the annihilator for $\mathbf{X}$, and hence $\mathbf{M}\mathbf{X}_2 = \mathbf{0}$.

In light of the above, (7.9) becomes

$$\mathbf{X}_2'\mathbf{M}_1\mathbf{y} = \mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2\hat{\boldsymbol{\beta}}_2$$

To go from this equation to (7.8), we just need to check that $\mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2$ is invertible. The proof of this last fact is left as an exercise (exercise 7.6.10). □

7.3.2 Intuition

As exercise 7.6.8 asks you to show, the expression for $\hat{\boldsymbol{\beta}}_2$ in theorem 7.3.1 can be rewritten as

$$\hat{\boldsymbol{\beta}}_2 = [(\mathbf{M}_1\mathbf{X}_2)'\mathbf{M}_1\mathbf{X}_2]^{-1}(\mathbf{M}_1\mathbf{X}_2)'\mathbf{M}_1\mathbf{y} \tag{7.10}$$

Close inspection of this formula confirms the following claim: There is another way to obtain $\hat{\boldsymbol{\beta}}_2$ besides just regressing $\mathbf{y}$ on $\mathbf{X}$ and then extracting the last K_2 elements: We can also regress $\mathbf{M}_1\mathbf{y}$ on $\mathbf{M}_1\mathbf{X}_2$ to produce the same result.

To get some feeling for what this means, let's look at a special case, where $\mathbf{X}_2$ is the single column $\text{col}_K(\mathbf{X})$, containing the observations on the K-th regressor. In view of the preceding discussion, the OLS estimate $\hat{\beta}_K$ can be found by regressing

$$\tilde{\mathbf{y}} := \mathbf{M}_1\mathbf{y} = \text{residuals of regressing } \mathbf{y} \text{ on } \mathbf{X}_1$$

on

$$\tilde{\mathbf{x}}_K := \mathbf{M}_1 \,\text{col}_K(\mathbf{X}) = \text{residuals of regressing } \text{col}_K(\mathbf{X}) \text{ on } \mathbf{X}_1$$

Loosely speaking, these two residual terms $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{x}}_K$ can be thought of as the parts of $\mathbf{y}$ and $\text{col}_K(\mathbf{X})$ that are "not explained by" $\mathbf{X}_1$. Thus, on an intuitive level, the process for obtaining the OLS estimate $\hat{\beta}_K$ is:

1. Remove effects of all other regressors from $\mathbf{y}$ and $\text{col}_K(\mathbf{X})$, producing $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{x}}_K$.
2. Regress $\tilde{\mathbf{y}}$ on $\tilde{\mathbf{x}}_K$.

This is obviously different from the process for obtaining the coefficient of the vector $\text{col}_K(\mathbf{X})$ in a simple univariate regression, the latter being just

1. Regress $\mathbf{y}$ on $\text{col}_K(\mathbf{X})$.

In words, the difference between the univariate least squares estimated coefficient of the K-th regressor and the multiple regression OLS coefficient is that the multiple regression coefficient $\hat{\beta}_K$ measures the *isolated relationship* between x_K and y, without taking into account indirect channels involving other variables.

We can illustrate this idea further with a small simulation. Suppose that

$$y = x_1 + x_2 + u \quad \text{where} \quad u \overset{\text{IID}}{\sim} \mathcal{N}(0,1)$$

If we generate N independent observations from this model and regress y the observations of (x_1, x_2), then, provided that N is sufficiently large, the coefficients for x_1 and x_2 will both be close to unity.[1] However, if we regress y on x_1 alone, then the coefficient for x_1 will depend on the relationship between x_1 and x_2. For example:

```
> N <- 1000
> beta <- c(1, 1)
> X1 <- runif(N)
> X2 <- 10 * exp(X1) + rnorm(N)
> X <- cbind(X1,  X2)
> y <- X %*% beta + rnorm(N)
> results <- lm(y ~ 0 + X1)
> results$coefficients
      X1
30.76840
```

Here the coefficient for x_1 is much larger than unity, because an increase in x_1 tends to have a large positive effect on x_2, which in turn increases y. The coefficient in the univariate regression reflects this total effect.

7.3.3 Simple Regression

As an application of the FWL theorem, let's derive the familiar expression for the slope coefficient in simple regression. Simple regression is a special case of multivariate regression, where the intercept is included (i.e., $\mathbf{1}$ is the first column of $\mathbf{X}$) and $K = 2$. For simplicity, the second column of $\mathbf{X}$ will be denoted simply by $\mathbf{x}$. As we saw in (4.26) on page 141, the OLS estimates are

$$\hat{\beta}_2 = \frac{\sum_{n=1}^N (x_n - \bar{x})(y_n - \bar{y})}{\sum_{n=1}^N (x_n - \bar{x})^2} \quad \text{and} \quad \hat{\beta}_1 = \bar{y} - \hat{\beta}_2 \bar{x}$$

where $\bar{x}$ is the sample mean of $\mathbf{x}$ and $\bar{y}$ is the sample mean of $\mathbf{y}$. The coefficient $\hat{\beta}_2$ is known as the slope coefficient, while $\hat{\beta}_1$ is called the intercept coefficient.

We can rewrite $\hat{\beta}_2$ more succinctly as

$$\hat{\beta}_2 = [(\mathbf{x} - \bar{x}\mathbf{1})'(\mathbf{x} - \bar{x}\mathbf{1})]^{-1}(\mathbf{x} - \bar{x}\mathbf{1})'(\mathbf{y} - \bar{y}\mathbf{1}) \tag{7.11}$$

[1] The reason is that, in this setting, the OLS estimator is consistent for the coefficients. A proof can be found in chapter 7.

By the FWL theorem (equation 7.10), we also have

$$\hat{\beta}_2 = [(\mathbf{M}_c\mathbf{x})'\mathbf{M}_c\mathbf{x}]^{-1}(\mathbf{M}_c\mathbf{x})'\mathbf{M}_c\mathbf{y} \tag{7.12}$$

where $\mathbf{M}_c$ is the annihilator associated with the single regressor $\mathbf{1}$, as defined in (6.16). It is straightforward to show that for this annihilator $\mathbf{M}_c$ and any $\mathbf{z}$, we have $\mathbf{M}_c\mathbf{z} = \mathbf{z} - \bar{z}\mathbf{1}$. In other words, the annihilator associated with $\mathbf{1}$ converts vectors into deviations around their mean. (The "c" in $\mathbf{M}_c$ reminds us that $\mathbf{M}_c$ centers vectors around their mean.)

It is now easy to see that the right-hand sides of (7.11) and (7.12) coincide.

7.3.4 Centered Observations

Let's generalize the discussion in the preceding section to the case where there are multiple non-constant regressors. The only difference to the preceding case is that instead of one column $\mathbf{x}$ of observations on a single non-constant regressor, we have a matrix $\mathbf{X}_2$ containing multiple columns, each a vector of observations on a non-constant regressor.

If the OLS estimate $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ is partitioned into $(\hat{\beta}_1, \hat{\boldsymbol{\beta}}_2)$, then we can write

$$\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{1}\beta_1 + \mathbf{X}_2\hat{\boldsymbol{\beta}}_2$$

Applying the FWL theorem (equation 7.10) once more, we can write $\hat{\boldsymbol{\beta}}_2$ as

$$\hat{\boldsymbol{\beta}}_2 = [(\mathbf{M}_c\mathbf{X}_2)'\mathbf{M}_c\mathbf{X}_2]^{-1}(\mathbf{M}_c\mathbf{X}_2)'\mathbf{M}_c\mathbf{y}$$

where $\mathbf{M}_c$ is the annihilator in (6.16). As we saw in the last section, $\mathbf{M}_c\mathbf{y}$ is $\mathbf{y}$ centered around its mean. Similarly, $\mathbf{M}_c\mathbf{X}_2$ is a matrix formed by taking each column of $\mathbf{X}_2$ and centering it around its mean.

What we have shown is this: In an OLS regression with an intercept, the estimated coefficients of the non-constant (i.e., non-intercept) regressors are equal to the estimated coefficients of a zero-intercept regression performed after all variables have been centered around their mean.

7.3.5 Precision of the OLS Estimates

Let's return to the regression problem, with assumptions 7.1.1–7.1.2 in force. In theorem 7.2.3, we showed that the variance-covariance matrix of the OLS estimate $\hat{\boldsymbol{\beta}}$

given $\mathbf{X}$ is $\sigma^2(\mathbf{X}'\mathbf{X})^{-1}$. The scalar variances of the individual OLS coefficient estimates $\hat{\beta}_1, \ldots, \hat{\beta}_K$ are given by the principle diagonal of this matrix. Since any one of these OLS estimates $\hat{\beta}_k$ is unbiased (theorem 7.2.1), small variance of $\hat{\beta}_k$ corresponds to probability mass concentrated around the true parameter β_k. In this case, we say that the estimator has high **precision**. (Precision of an estimator is sometimes defined as the inverse of the variance, although definitions do vary.)

The Gauss-Markov theorem tells us that, at least as far as unbiased linear estimators go, the OLS estimates will have low variance. Put differently, if we fix the regression problem and vary the estimators, the OLS estimators will have the most precision. However, we want to think about precision a different way: If we hold the estimation technique fixed (use only OLS) and consider different regression problems, which problems will have high precision estimates, and which will have low precision estimates?

To answer this question, let's focus on the variance of a fixed coefficient β_k. We can write the regression model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u}$ as

$$\mathbf{y} = \mathbf{X}_1\boldsymbol{\beta}_1 + \operatorname{col}_k(\mathbf{X})\beta_k + \mathbf{u} \tag{7.13}$$

where $\operatorname{col}_k(\mathbf{X})$ is the vector of observations of the k-th regressor, $\mathbf{X}_1$ contains as its columns the observations of the other regressors, and $\hat{\boldsymbol{\beta}}_1$ is the OLS estimates of the corresponding coefficients. From the FWL theorem, we can then express $\hat{\beta}_k$ as

$$\hat{\beta}_k = (\operatorname{col}_k(\mathbf{X})'\mathbf{M}_1 \operatorname{col}_k(\mathbf{X}))^{-1} \operatorname{col}_k(\mathbf{X})'\mathbf{M}_1\mathbf{y} \tag{7.14}$$

where $\mathbf{M}_1$ is the annihilator corresponding to $\mathbf{X}_1$. That is, $\mathbf{M}_1 := \mathbf{I} - \mathbf{P}_1$ where $\mathbf{P}_1$ is the matrix $\mathbf{X}_1(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'$ projecting onto the column space of $\mathbf{X}_1$. Applying $\mathbf{M}_1$ to both sides of (7.13), we obtain

$$\mathbf{M}_1\mathbf{y} = \mathbf{M}_1 \operatorname{col}_k(\mathbf{X})\beta_k + \mathbf{M}_1\mathbf{u}$$

Substituting this into (7.14), we obtain a second expression for $\hat{\beta}_k$ in terms of the shock vector $\mathbf{u}$:

$$\hat{\beta}_k = \beta_k + (\operatorname{col}_k(\mathbf{X})'\mathbf{M}_1 \operatorname{col}_k(\mathbf{X}))^{-1} \operatorname{col}_k(\mathbf{X})'\mathbf{M}_1\mathbf{u} \tag{7.15}$$

Some calculations then show (exercise 7.6.11) that

$$\operatorname{var}[\hat{\beta}_k \mid \mathbf{X}] = \sigma^2(\operatorname{col}_k(\mathbf{X})'\mathbf{M}_1 \operatorname{col}_k(\mathbf{X}))^{-1} = \sigma^2\|\mathbf{M}_1 \operatorname{col}_k(\mathbf{X})\|^{-2} \tag{7.16}$$

Thus, the variance of $\hat{\beta}_k$ depends on two components, the variance σ^2 of the shock u, and the norm of the vector $\mathbf{M}_1 \operatorname{col}_k(\mathbf{X})$.

The variance in σ^2 is in some sense unavoidable: Some data is noisier that other data. The larger the variance in the unobservable shock, the harder it will be to estimate β_k with good precession. The term $\|\mathbf{M}_1 \operatorname{col}_k(\mathbf{X})\|^{-2}$ is more interesting. The vector $\mathbf{M}_1 \operatorname{col}_k(\mathbf{X})$ is the residuals from regressing $\operatorname{col}_k(\mathbf{X})$ on $\mathbf{X}_1$, and $\|\mathbf{M}_1 \operatorname{col}_k(\mathbf{X})\|$ is the norm of this vector. If this norm is small, then the variance of $\hat{\beta}_k$ will be large.

When will this norm be small (and the variance of $\hat{\beta}_k$ correspondingly large)? This will be the case when $\operatorname{col}_k(\mathbf{X})$ is "almost" a linear combination of the other regressors. To see this, suppose that $\operatorname{col}_k(\mathbf{X})$ is indeed "almost" a linear combination of the other regressors. This implies that

$$\mathbf{P}_1 \operatorname{col}_k(\mathbf{X}) \approx \operatorname{col}_k(\mathbf{X})$$

because $\mathbf{P}_1$ projects into the column space of the other regressors $\mathbf{X}_1$, so we are saying that $\operatorname{col}_k(\mathbf{X})$ is "almost in" that span. Now if $\mathbf{P}_1 \operatorname{col}_k(\mathbf{X}) \approx \operatorname{col}_k(\mathbf{X})$, then $\|\mathbf{P}_1 \operatorname{col}_k(\mathbf{X}) - \operatorname{col}_k(\mathbf{X})\| \approx 0$, and hence

$$\|\mathbf{M}_1 \operatorname{col}_k(\mathbf{X})\| = \|\operatorname{col}_k(\mathbf{X}) - \mathbf{P}_1 \operatorname{col}_k(\mathbf{X})\| \approx 0$$

In other words, the norm of $\mathbf{M}_1 \operatorname{col}_k(\mathbf{X})$ is small, and hence the variance of $\hat{\beta}_k$ is large.

This situation is sometimes referred to as **multicollinearity**. As we have just seen, multicollinearity is associated with poor precision in estimates of the coefficients.

7.4 Normal Errors

In this section, we're going to strengthen and augment our previous assumptions by specifying the parametric class of the error vector $\mathbf{u}$. Once this class is specified, we can determine the distribution of the OLS estimate $\hat{\boldsymbol{\beta}}$ up to the unknown parameters $\sigma^2, \beta_1, \ldots, \beta_K$. (In other words, if values for these parameters are specified, then the distribution of $\hat{\boldsymbol{\beta}}$ is fully specified.) This will allow us to test hypotheses about the coefficients.

Because of its many attractive properties, the normal distribution is our go-to distribution, at least for the case where we have no information that suggests another distribution, or contradicts the normality assumption. Following this grand tradition, we will assume that $\mathbf{u}$ is a normally distributed element of $\mathbb{R}^N$.

A normal distribution in $\mathbb{R}^N$ is fully specified by its mean and variance-covariance matrix. In this case, given that we're strengthening our previous assumptions, we

have no choice here. From assumption 7.1.1, the mean is $\mathbb{E}[\mathbf{u}] = \mathbf{0}$, and, from assumption 7.1.2, the variance-covariance matrix is $\mathbb{E}[\mathbf{u}\mathbf{u}'] = \sigma^2\mathbf{I}$. We must then have $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$.

Furthermore, assumption 7.1.1 implies that shocks and regressors are uncorrelated (see fact 7.1.2 on page 197). To make life a bit easier for ourselves, we'll go a step further and assume they are independent.

Assumption 7.4.1. $\mathbf{X}$ and $\mathbf{u}$ are independent, and $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$.

Notice that assumption 7.4.1 implies both assumption 7.1.1 and assumption 7.1.2.

7.4.1 Preliminary Results

Assumption 7.4.1 also implies that the conditional distribution of $\hat{\boldsymbol{\beta}}$ given $\mathbf{X}$ is normal, since $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}$, and linear combinations of normals are normal. The next theorem records this result.

Theorem 7.4.1. *Under assumption 7.4.1, the distribution of $\hat{\boldsymbol{\beta}}$ given $\mathbf{X}$ is $\mathcal{N}(\boldsymbol{\beta}, \sigma^2(\mathbf{X}'\mathbf{X})^{-1})$.*

It follows from theorem 7.4.1 that the distribution of individual coefficient $\hat{\beta}_k$ given $\mathbf{X}$ is also normal. This can be established directly from (7.15), and the variance is given in (7.16). However, we will use theorem 7.4.1 instead, since it gives an expression for the variance which is easier to compute. To do this, let $\mathbf{e}_k$ be the k-th canonical basis vector, and observe that

$$\mathbf{e}_k'\hat{\boldsymbol{\beta}} \sim \mathcal{N}(\mathbf{e}_k'\boldsymbol{\beta}, \sigma^2\mathbf{e}_k'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{e}_k)$$

In other words,

$$\hat{\beta}_k \sim \mathcal{N}(\beta_k, \sigma^2\mathbf{e}_k'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{e}_k) \tag{7.17}$$

Note that $\mathbf{e}_k'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{e}_k$ is the (k,k)-th element of the matrix $(\mathbf{X}'\mathbf{X})^{-1}$.[2] It then follows that

$$z_k := \frac{\hat{\beta}_k - \beta_k}{\sigma\sqrt{\mathbf{e}_k'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{e}_k}} \sim \mathcal{N}(0,1) \tag{7.18}$$

Our second preliminary result concerns the distribution of $\hat{\sigma}^2$, or more precisely, of

$$Q := (N-K)\frac{\hat{\sigma}^2}{\sigma^2} \tag{7.19}$$

[2] See exercise 2.6.15 on page 80.

Theorem 7.4.2. *Under assumption 7.4.1, the distribution of Q given $\mathbf{X}$ is $\chi^2(N-K)$.*

Proof. To see that $Q \sim \chi^2(N-K)$ given $\mathbf{X}$, observe that

$$Q = \frac{(\mathbf{Mu})'(\mathbf{Mu})}{\sigma^2} = \frac{\mathbf{u}'\mathbf{Mu}}{\sigma^2} = (\sigma^{-1}\mathbf{u})'\mathbf{M}(\sigma^{-1}\mathbf{u})$$

Since $\sigma^{-1}\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the expression on the far right is $\chi^2(N-K)$, as follows from fact 2.4.9 and our previous result that $\operatorname{trace}(\mathbf{M}) = \operatorname{rank}(\mathbf{M}) = N-K$ (recall that for idempotent matrices, trace and rank are equal—see fact 2.3.9). □

7.4.2 The t-test

Let's consider the problem of testing a hypothesis about an individual coefficient β_k. Specifically, we consider the null hypothesis

$$H_0 : \beta_k = \beta_k^0 \quad \text{against} \quad H_1 : \beta_k \neq \beta_k^0$$

where β_k^0 is any number. If we knew σ^2, we could test H_0 via (7.18). Since we don't, the standard methodology is to replace σ^2 with its estimator $\hat{\sigma}^2$, and determine the distribution of the resulting test statistic. Our next result implements this idea. In doing so, we will make use of the following notation:

$$\operatorname{se}(\hat{\beta}_k) := \sqrt{\hat{\sigma}^2 \mathbf{e}_k'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{e}_k}$$

The term $\operatorname{se}(\hat{\beta}_k)$ is called the **standard error** of $\hat{\beta}_k$. It can be regarded as the sample estimate of the standard deviation of $\hat{\beta}_k$. Replacing this standard deviation with its sample estimate $\operatorname{se}(\hat{\beta}_k)$ and β_k with β_k^0 in (7.18), we obtain the **t-statistic**

$$t_k := \frac{\hat{\beta}_k - \beta_k^0}{\operatorname{se}(\hat{\beta}_k)} \tag{7.20}$$

The distribution of this statistic under the null is described in the next theorem.

Theorem 7.4.3. *Let assumption 7.4.1 hold. If the null hypothesis H_0 is true, then, conditional on $\mathbf{X}$, the distribution of the t-statistic in (7.20) is Student's t with $N-K$ degrees of freedom.*

Proof. Suppose that the null hypothesis is true. We then have

$$t_k = \frac{\hat{\beta}_k - \beta_k}{\text{se}(\hat{\beta}_k)} = \frac{\hat{\beta}_k - \beta_k}{\sqrt{\sigma^2 \mathbf{e}_k'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{e}_k}} \sqrt{\frac{1}{\hat{\sigma}^2/\sigma^2}} = z_k \sqrt{\frac{1}{\hat{\sigma}^2/\sigma^2}}$$

where z_k is defined in (7.18). Multiplying and dividing by the square root of $N - K$, we can write this as

$$t_k = z_k \sqrt{\frac{N-K}{(N-K)\hat{\sigma}^2/\sigma^2}} = z_k \sqrt{\frac{N-K}{Q}}$$

where Q is defined in (7.19). In view of fact 1.3.6 on page 27, we know that t_k is Student's t with $N - K$ degrees of freedom if z_k is standard normal, Q is $\chi^2(N-K)$ and z_k and Q are independent. The first two results were established in §7.4.1, so it remains only to show that z_k and Q are independent.

To see this, note that if $\mathbf{a}$ and $\mathbf{b}$ are independent random vectors and f and g are two functions, then $f(\mathbf{a})$ and $g(\mathbf{b})$ are likewise independent. Since we can write z_k as a function of $\hat{\boldsymbol{\beta}}$ and Q as a function of $\hat{\mathbf{u}}$, it suffices to show that $\hat{\boldsymbol{\beta}}$ and $\hat{\mathbf{u}}$ are independent. Since both are normally distributed given $\mathbf{X}$, fact 2.4.7 on page 74 implies that they will be independent whenever their covariance is zero. This is exercise 7.6.5, which completes the proof of theorem 7.4.3. □

Recall that a test is a test statistic T and a critical value c, with the rule: Reject H_0 if $T > c$. For T we take $T = |t_k|$. Let a desired size α be given. In view of (5.15) on page 165, we choose $c = c_\alpha$ to solve $\alpha = \mathbb{P}_\theta\{T > c\}$, or

$$1 - \alpha = \mathbb{P}_\theta\{|t_k| \leq c\}$$

From (1.14) on page 21, we know that the solution is $c_\alpha = F^{-1}(1 - \alpha/2)$, where F is the Student's t cdf with $N - K$ degrees of freedom. In view of example 5.3.3, the corresponding p-value is $2F(-|t_k|)$.

Let's look at an example. The most common implementation of the t-test is the test that a given coefficent is equal to zero. For the k-th coefficient β_k, this leads to the statistic

$$t_k := \frac{\hat{\beta}_k}{\text{se}(\hat{\beta}_k)}$$

This statistic is sometimes called the Z-score. To illustrate further, an application with simulated data is given in listing 9. If you run the program, you will find that

the Z-scores calculated by the `zscore` function agree with the "t value" column of the summary table produced by R's `summary` function (the last line of listing 9). It is left as an exercise to check that the p-values in the same table agree with the formula $2F(-|t_k|)$ given in the last paragraph.

Listing 9 Calculating Z-scores

```
set.seed(1234)
N <- 50; K <- 3
beta <- rep(1, K)
X <- cbind(runif(N), runif(N), runif(N))
u <- rnorm(N)
y <- X %*% beta + u

betahat <- solve(t(X) %*% X) %*% t(X) %*% y
residuals <- y - X %*% betahat
sigmahat <- sqrt(sum(residuals^2) / (N - K))

# Compute t-stat (Z-score) for k-th regressor
zscore <- function(k) {
    se <- sigmahat * sqrt(solve(t(X) %*% X)[k, k])
    return(betahat[k] / se)
}
# Print t-stats
for (k in 1:3) {
    cat("t-stat, k =", k, ":", zscore(k), "\n")
}
# For comparison:
print(summary(lm(y ~ X - 1)))
```

7.4.3 The F-test

The t-test is used to test hypotheses about individual regressors. For hypotheses concerning multiple regressors, the most common test is the F-test. The F-test can test quite general hypotheses, but for simplicity we will focus on null hypotheses that restrict a subset of the coefficients to be zero.

In what follows, we let $\mathbf{X}_1$, $\mathbf{X}_2$, $\hat{\boldsymbol{\beta}}_1$, $\hat{\boldsymbol{\beta}}_2$, $\mathbf{P}_1$ and $\mathbf{M}_1$ be as defined in §7.3.1. As our null hypothesis we take

$$H_0 \colon \boldsymbol{\beta}_2 = \mathbf{0} \quad \text{against} \quad H_1 \colon \boldsymbol{\beta}_2 \neq \mathbf{0}$$

Since

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u} = \mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{X}_2\boldsymbol{\beta}_2 + \mathbf{u} \tag{7.21}$$

it follows that under the null hypothesis we have

$$\mathbf{y} = \mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{u} \tag{7.22}$$

Letting

$$\text{USSR} := \|\mathbf{M}\mathbf{y}\|^2 \quad \text{and} \quad \text{RSSR} := \|\mathbf{M}_1\mathbf{y}\|^2$$

be the sums of squared residuals for the unrestricted regression (7.21) and restricted regression (7.22) respectively, the standard test statistic for our null hypothesis is

$$F := \frac{(\text{RSSR} - \text{USSR})/K_2}{\text{USSR}/(N-K)} \tag{7.23}$$

Large residuals in the restricted regression (7.22) relative to those in (7.21) result in large values for F, which translates to evidence against the null hypothesis.

Theorem 7.4.4. *Let assumption 7.4.1 hold. If the null hypothesis is true, then, conditional on* $\mathbf{X}$*, the statistic* F *defined in* (7.23) *has the* F *distribution, with parameters* $(K_2, N-K)$*.*

Proof. Let $Q_1 := (\text{RSSR} - \text{USSR})/\sigma^2$ and let $Q_2 := \text{USSR}/\sigma^2$, so that

$$F = \frac{Q_1/K_2}{Q_2/(N-K)}$$

In view of fact 1.3.7 on page 28, it now suffices to show that, under the null hypothesis,

(a) Q_1 is chi-squared with K_2 degrees of freedom.

(b) Q_2 is chi-squared with $N-K$ degrees of freedom.

(c) Q_1 and Q_2 are independent.

Part (b) was established in theorem 7.4.2. Regarding part (a), observe that, under the null hypothesis,

- $\text{USSR} = \|\mathbf{M}\mathbf{y}\|^2 = \|\mathbf{M}(\mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{u})\|^2 = \|\mathbf{M}\mathbf{u}\|^2 = \mathbf{u}'\mathbf{M}\mathbf{u}$, and

- RSSR $= \|\mathbf{M}_1\mathbf{y}\|^2 = \|\mathbf{M}_1(\mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{u})\|^2 = \|\mathbf{M}_1\mathbf{u}\|^2 = \mathbf{u}'\mathbf{M}_1\mathbf{u}$

It follows that

$$\text{RSSR} - \text{USSR} = \mathbf{u}'\mathbf{M}_1\mathbf{u} - \mathbf{u}'\mathbf{M}\mathbf{u} = \mathbf{u}'(\mathbf{M}_1 - \mathbf{M})\mathbf{u}$$

Using the definitions of $\mathbf{M}$ and $\mathbf{M}_1$, we then obtain

$$Q_1 = \frac{\text{RSSR} - \text{USSR}}{\sigma^2} = \frac{\mathbf{u}'(\mathbf{I} - \mathbf{P}_1 - \mathbf{I} + \mathbf{P})\mathbf{u}}{\sigma^2} = (\sigma^1\mathbf{u})'(\mathbf{P} - \mathbf{P}_1)(\sigma^{-1}\mathbf{u})$$

It is an exercise to show that $(\mathbf{P} - \mathbf{P}_1)$ is symmetric and idempotent.[3] Applying the techniques in the proof of theorem 7.2.2, we see that

$$\text{rank}(\mathbf{P} - \mathbf{P}_1) = \text{trace}(\mathbf{P} - \mathbf{P}_1) = \text{trace}(\mathbf{P}) - \text{trace}(\mathbf{P}_1) = K - K_1 = K_2$$

Via fact 2.4.9, we conclude that $Q_1 \sim \chi^2(K_2)$, as was to be shown.

To complete the proof, it remains to show that, under the null hypothesis and taking $\mathbf{X}$ as given, Q_1 and Q_2 are independent. To see this, observe that Q_1 is a function of $(\mathbf{P} - \mathbf{P}_1)\mathbf{u}$, while Q_2 is a function of $\mathbf{M}\mathbf{u}$. Since both $(\mathbf{P} - \mathbf{P}_1)\mathbf{u}$ and $\mathbf{M}\mathbf{u}$ are normal given $\mathbf{X}$, it suffices to show that their covariance is zero. This is the case, because

$$\text{cov}[(\mathbf{P} - \mathbf{P}_1)\mathbf{u}, \mathbf{M}\mathbf{u} \mid \mathbf{X}] = \mathbb{E}\left[(\mathbf{P} - \mathbf{P}_1)\mathbf{u}(\mathbf{M}\mathbf{u})' \mid \mathbf{X}\right] = \mathbb{E}\left[(\mathbf{P} - \mathbf{P}_1)\mathbf{u}\mathbf{u}'\mathbf{M} \mid \mathbf{X}\right]$$

Since $\mathbf{P}$, $\mathbf{P}_1$ and $\mathbf{M}$ are just functions of $\mathbf{X}$, this becomes

$$(\mathbf{P} - \mathbf{P}_1)\mathbb{E}\left[\mathbf{u}\mathbf{u}' \mid \mathbf{X}\right]\mathbf{M} = \sigma^2(\mathbf{P} - \mathbf{P}_1)\mathbf{M} = \sigma^2(\mathbf{P} - \mathbf{P}_1)(\mathbf{I} - \mathbf{P})$$

Using idempotence and fact 3.1.2, the matrix product on the right is

$$(\mathbf{P} - \mathbf{P}_1)(\mathbf{I} - \mathbf{P}) = \mathbf{P} - \mathbf{P}^2 - \mathbf{P}_1 + \mathbf{P}_1\mathbf{P} = \mathbf{P} - \mathbf{P} - \mathbf{P}_1 + \mathbf{P}_1 = \mathbf{0}$$

This completes the proof of independence, and hence of theorem 7.4.4. □

The most common implementation of the F test is the test that all coefficients of non-constant regressors are zero. In this case (7.21) becomes

$$\mathbf{y} = \mathbf{1}\beta_1 + \mathbf{X}_2\boldsymbol{\beta}_2 + \mathbf{u} \tag{7.24}$$

where $\boldsymbol{\beta}_2$ is the vector of coefficients corresponding to the non-constant regressors. Since $\mathbf{X}_1 = \mathbf{1}$, we then have $\mathbf{M}_1 = \mathbf{M}_c$, where the latter is defined in (6.16) on

Listing 10 Calculating the F statistic

```
set.seed(1234)
N <- 50; K <- 3
beta <- rep(1, K)
x2 <- runif(N); x3 <- runif(N)
X <- cbind(rep(1, N), x2, x3)
u <- rnorm(N)
y <- X %*% beta + u

betahat <- solve(t(X) %*% X) %*% t(X) %*% y
residuals <- y - X %*% betahat
ussr <- sum(residuals^2)
rssr <- sum((y - mean(y))^2)

Fa <- (rssr - ussr) / 2
Fb <- ussr / (N - K)

cat("F =", Fa / Fb, "\n")

# For comparison:
print(summary(lm(y ~ x2 + x3)))
```

page 187, and hence RSSR is the squared norm of $\mathbf{y} - \bar{y}\mathbf{1}$. An application with simulated data is given in listing 10. If you run the program, you will find that the F statistic calculated by the theoretical formula agrees with the F statistic produced by R's `summary` function.

It is an exercise to show that in the case of (7.24), the F statistic in (7.23) can be rewritten as

$$F = \frac{R_c^2}{1 - R_c^2} \frac{N - K}{K_2} \tag{7.25}$$

where R_c^2 is the centered R squared defined in §6.3.2. Can you provide some intuition as to why large F is evidence against the null?

7.5 When the Assumptions Fail

The standard OLS assumptions are very strict, and the results we have obtained are sensitive to their failure. For example, if our basic assumption $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u}$ is not true, then pretty much all the results discussed in this chapter are invalid. In what follows, let's be polite and consider situations where the standard OLS assumptions are only slightly wrong.

7.5.1 Endogeneity Bias

Even if the model is correctly specified, the OLS estimates can be biased when assumption 7.1.1 (i.e., $\mathbb{E}[\mathbf{u} \,|\, \mathbf{X}] = \mathbf{0}$) fails. Assumption 7.1.1 is sometimes called an exogeneity assumption. When it fails, the bias is called **endogeneity bias**. There are many sources of endogeneity bias. We will look at two examples.

As a first example, consider again the Cobb-Douglas example on page 196, which yields the regression model

$$\ln y_n = \beta + \gamma \ln k_n + \delta \ln \ell_n + u_n$$

Here y is output, k is capital, ℓ is labor, and subscript n indicates observation on the n-th firm. The term u_n is a firm specific productivity shock. A likely problem here is that the productivity shocks are positively correlated, and, moreover, the firm will

[3] Hint: See fact 3.1.2 on page 88.

choose higher levels of both capital and labor when it anticipates high productivity in the current period. This will lead to endogeneity bias.

To illustrate this, suppose that $u_{n,-1}$ is the productivity shock received by firm n last period, and this value is observable to the firm. Suppose that productivity follows a random walk, with $u_n = u_{n,-1} + \eta_n$, where η_n is zero mean white noise. As a result, the firm forecasts period n productivity as $\mathbb{E}[u_n \mid u_{n,-1}] = u_{n,-1}$. Finally, suppose that the firm increases labor input when productivity is anticipated to be high, with the specific relationship $\ell_n = a + b\mathbb{E}[u_n \mid u_{n,-1}]$ for $b > 0$. When all shocks are zero mean we then have

$$\mathbb{E}[\ell_n u_n] = \mathbb{E}[(a + bu_{n,-1})(u_{n,-1} + \eta_n)] = \mathbb{E}[bu_{n,-1}^2]$$

This term will be strictly positive whenever $u_{n,-1}$ has positive variance. Thus, the conditions of fact 7.1.2 (page 197) fail, and therefore assumption 7.1.1 does not hold (because assumption 7.1.1 implies fact 7.1.2).

This source of endogeneity bias in estimating production functions has been discussed many times in the literature. The best solution is better modeling. For an illustration of careful modeling (and discussion of other potential problems with estimating production functions) see the paper of Olley and Pakes (1996).

As a second example of endogeneity bias, suppose next that we have in hand data that is generated according to the simple AR(1) model

$$y_0 = 0 \quad \text{and} \quad y_n = \beta y_{n-1} + u_n \quad \text{for } n = 1, \ldots, N \tag{7.26}$$

Here we assume that $\{u_n\}_{n=1}^N \overset{\text{IID}}{\sim} \mathcal{N}(0, \sigma^2)$. The unknown parameters are β and σ^2. Letting

$$\mathbf{y} := (y_1, \ldots, y_N), \quad \mathbf{x} := (y_0, \ldots, y_{N-1}) \quad \text{and} \quad \mathbf{u} := (u_1, \ldots, u_N)$$

we can write the N equations in (7.26) as

$$\mathbf{y} = \beta \mathbf{x} + \mathbf{u}$$

Suppose that we now estimate β by regressing $\mathbf{y}$ on $\mathbf{x}$, obtaining the OLS estimate

$$\hat{\beta} := (\mathbf{x}'\mathbf{x})^{-1}\mathbf{x}'\mathbf{y} = \frac{\mathbf{x}'\mathbf{y}}{\mathbf{x}'\mathbf{x}}$$

It turns out that $\hat{\beta}$ is a biased estimate of β. The source of the bias is failure of the exogeneity assumption. For example, fact 7.1.2 tells us that if assumption 7.1.1

holds, then we must have $\mathbb{E}[u_m x_{n+1}] = 0$ for any m and n. In the current set-up, this equates to $\mathbb{E}[u_m y_n] = 0$. We now show that this fails whenever $\beta \neq 0$ and $n \geq m$. To see this, observe that (exercise 7.6.15) we can write y_n as

$$y_n = \sum_{j=0}^{n-1} \beta^j u_{n-j} \tag{7.27}$$

and, therefore,

$$\mathbb{E}[y_n u_m] = \sum_{j=0}^{n-1} \beta^j \mathbb{E}[u_{n-j} u_m] = \beta^{n-m}\sigma^2 \quad \text{whenever } n \geq m \tag{7.28}$$

It follows that when $\beta \neq 0$, assumption 7.1.1 must fail.

To help illustrate the bias in our estimate of β, let's generate the data 10,000 times, compute $\hat{\beta}$ on each occasion, and then take the sample mean. The code to do this is given in Listing 11, with $N = 20$ and $\beta = 0.9$. The resulting sample mean was 0.82. Since the number of replications is very large (i.e., 10,000), this will be very close to $\mathbb{E}[\hat{\beta}]$. In fact, the asymptotic 95% confidence interval for our estimate of $\mathbb{E}[\hat{\beta}]$ is (0.818, 0.824).

This bias towards small values is reinforced by the histogram of the observations given in figure 7.1, which is skewed to the left.

7.5.2 Misspecification and Bias

The linearity assumption $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u}$ can fail in many ways, and when it does the estimator $\hat{\boldsymbol{\beta}}$ will typically be biased. Let's look at one possible failure, where the model is still linear, but some variables are omitted. In particular, let's suppose that the data generating process is in fact

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\theta} + \mathbf{u} \tag{7.29}$$

We'll also assume that $\boldsymbol{\theta} \neq \mathbf{0}$, and that $\mathbb{E}[\mathbf{u} \mid \mathbf{X}, \mathbf{Z}] = \mathbf{0}$.

Suppose that we are aware of the relationship between $\mathbf{y}$ and $\mathbf{X}$, but unaware of the relationship between $\mathbf{y}$ and $\mathbf{Z}$. This will lead us to mistakenly ignore $\mathbf{Z}$, and simply regress $\mathbf{y}$ on $\mathbf{X}$. Our OLS estimator will then be given by the usual expression $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$. Substituting in (7.29), we get

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\theta} + \mathbf{u}) = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Z}\boldsymbol{\theta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}$$

Listing 11 Generates observations of $\hat{\beta}$

```
N <- 20
y <- numeric(N)
y_zero <- 0
beta <- 0.9
num.reps <- 10000
betahat.obs <- numeric(num.reps)

for (j in 1:num.reps) {
    u <- rnorm(N)
    y[1] <- beta * y_zero + u[1]
    for (t in 1:(N-1)) {
        y[t+1] <- beta * y[t] + u[t+1]
    }
    x <- c(y_zero, y[-N])  # Lagged y
    betahat.obs[j]  <- sum(x * y) / sum(x^2)
}
print(mean(betahat.obs))
```

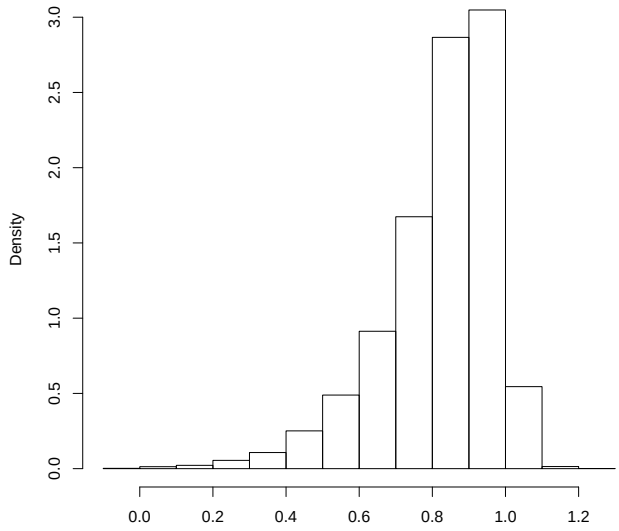


Figure 7.1: Observations of $\hat{\beta}$

We now have

$$\mathbb{E}\left[\hat{\boldsymbol{\beta}}\right] = \mathbb{E}\left[\mathbb{E}\left[\hat{\boldsymbol{\beta}} \,|\, \mathbf{X}, \mathbf{Z}\right]\right] = \boldsymbol{\beta} + \mathbb{E}\left[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Z}\right]\boldsymbol{\theta}$$

If the columns of $\mathbf{X}$ and $\mathbf{Z}$ are orthogonal, then $\mathbf{X}'\mathbf{Z} = \mathbf{0}$, the last term on the right-hand side drops out, and $\hat{\boldsymbol{\beta}}$ is unbiased. If this is not the case (typically it won't be), then $\hat{\boldsymbol{\beta}}$ is a biased estimator of $\boldsymbol{\beta}$.

7.5.3 Heteroskedasticity

[to be written]

7.6 Exercises

Throughout these exercises, we maintain assumptions 7.1.1 and 7.1.2. In particular, we assume that $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u}$, $\mathbb{E}\left[\mathbf{u} \,|\, \mathbf{X}\right] = \mathbf{0}$, and $\mathbb{E}\left[\mathbf{u}\mathbf{u}' \,|\, \mathbf{X}\right] = \sigma^2\mathbf{I}$, where σ is a positive constant.

A general hint for the exercises is to remember that, as stated in fact 3.3.7, when computing expectations conditional on $\mathbf{X}$, you can treat $\mathbf{X}$ or any function of $\mathbf{X}$ as constant. For example, matrices and vectors that depend only on $\mathbf{X}$ and non-random vectors/matrices, such as $\mathbf{X}\boldsymbol{\beta}$ or $\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$, can be regarded as constant.

Ex. 7.6.1. Show that $\mathbf{My} = \mathbf{Mu}$ and $\text{SSR} = \mathbf{u}'\mathbf{Mu}$.

Ex. 7.6.2. Prove the claims in fact 7.1.2.

Ex. 7.6.3. Confirm part 1 of fact 7.1.3: Show that $\text{var}[\mathbf{u}] = \mathbb{E}[\mathbf{uu}'] = \sigma^2\mathbf{I}$.

Ex. 7.6.4. Show that $\text{cov}[\mathbf{Py}, \mathbf{My} \mid \mathbf{X}] = 0$.

Ex. 7.6.5. Show that, under assumptions 7.1.1–7.1.2, we have $\text{cov}[\hat{\boldsymbol{\beta}}, \hat{\mathbf{u}} \mid \mathbf{X}] = \mathbf{0}$.

Ex. 7.6.6. Show that $\mathbb{E}[\mathbf{Py} \mid \mathbf{X}] = \mathbf{X}\boldsymbol{\beta}$ and $\text{var}[\mathbf{Py} \mid \mathbf{X}] = \sigma^2\mathbf{P}$.

Ex. 7.6.7. Show that $\mathbb{E}[\mathbf{My} \mid \mathbf{X}] = \mathbf{0}$ and $\text{var}[\mathbf{My} \mid \mathbf{X}] = \sigma^2\mathbf{M}$.

Ex. 7.6.8. Show that the two expressions for $\hat{\boldsymbol{\beta}}_2$ in (7.8) and (7.10) are equal.[4]

Ex. 7.6.9. In the proof of theorem 7.2.2 we used a clever trick with the trace to show that $\text{trace}(P) = K$. An alternative way to obtain the same result is to observe that, since P is idempotent, the trace is equal to the rank (fact 2.3.9 on page 70). Prove directly that the rank of P equals K.

Ex. 7.6.10. (Hard) At the end of the proof of theorem 7.3.1, it was claimed that the matrix $\mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2$ is invertible. Verify this claim.

Ex. 7.6.11. Confirm (7.16) on page 206.

Ex. 7.6.12. Using (7.16), show that for the simple OLS model $\mathbf{y} = \beta_1\mathbf{1} + \beta_2\mathbf{x} + \mathbf{u}$, the variance of $\hat{\beta}_2$ given $\mathbf{x}$ is $\sigma^2 / \sum_{n=1}^{N}(x_n - \bar{x})^2$.

Ex. 7.6.13. Show that in the case of (7.24), the F statistic in (7.23) can be rewritten as (7.25).[5]

Ex. 7.6.14. Suppose that assumption 7.4.1 holds, so that $\mathbf{X}$ and $\mathbf{u}$ are independent, and $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. Show that, conditional on $\mathbf{X}$,

1. $\mathbf{Py}$ and $\mathbf{My}$ are normally distributed, and
2. $\mathbf{Py}$ and $\mathbf{My}$ are independent.

Ex. 7.6.15. Verify the expression for y_n in (7.27).

[4] Hint: Use the symmetry and idempotence of the matrix $\mathbf{M}_1$.
[5] Hint: Note that $\mathbf{My} = \mathbf{MM}_c\mathbf{y}$ by fact 3.1.5 on page 90.

7.6.1 Solutions to Selected Exercises

Solution to Exercise 7.6.1. We have

$$\mathbf{My} = \mathbf{M}(\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) = \mathbf{Mu}$$

$$\therefore \quad \text{SSR} := \|\mathbf{My}\|^2 = \|\mathbf{Mu}\|^2 = (\mathbf{Mu})'(\mathbf{Mu}) = \mathbf{u}'\mathbf{M}'\mathbf{Mu} = \mathbf{u}'\mathbf{MMu}$$

by symmetry of $\mathbf{M}$. The result now follows from idempotence of $\mathbf{M}$. □

Solution to Exercise 7.6.2. The respective proofs for claims 1–4 are as follows:

1. $\mathbb{E}[\mathbf{u}] = \mathbb{E}[\mathbb{E}[\mathbf{u} \mid \mathbf{X}]] = \mathbb{E}[\mathbf{0}] = \mathbf{0}$
2. $\mathbb{E}[u_m \mid x_{nk}] = \mathbb{E}[\mathbb{E}[u_m \mid \mathbf{X}] \mid x_{nk}] = \mathbb{E}[0 \mid x_{nk}] = 0$
3. $\mathbb{E}[u_m x_{nk}] = \mathbb{E}[\mathbb{E}[u_m x_{nk} \mid x_{nk}]] = \mathbb{E}[x_{nk}\mathbb{E}[u_m \mid x_{nk}]] = 0$
4. $\text{cov}[u_m, x_{nk}] = \mathbb{E}[u_m x_{nk}] - \mathbb{E}[u_m]\mathbb{E}[x_{nk}] = 0$

□

Solution to Exercise 7.6.3. By definition,

$$\text{var}[\mathbf{u}] = \mathbb{E}[\mathbf{uu}'] - \mathbb{E}[\mathbf{u}]\mathbb{E}[\mathbf{u}']$$

Since $\mathbb{E}[\mathbf{u}] = \mathbb{E}[\mathbb{E}[\mathbf{u} \mid \mathbf{X}]] = \mathbf{0}$, this reduces to $\text{var}[\mathbf{u}] = \mathbb{E}[\mathbf{uu}']$. Moreover,

$$\mathbb{E}[\mathbf{uu}'] = \mathbb{E}[\mathbb{E}[\mathbf{uu}' \mid \mathbf{X}]] = \mathbb{E}[\sigma^2\mathbf{I}] = \sigma^2\mathbf{I}$$

□

Solution to Exercise 7.6.4. Note that

- $\mathbf{My} = \mathbf{M}(\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) = \mathbf{Mu}$
- $\mathbf{Py} = \mathbf{P}(\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) = \mathbf{X}\boldsymbol{\beta} + \mathbf{Pu}$
- $\mathbb{E}[\mathbf{My} \mid \mathbf{X}] = \mathbb{E}[\mathbf{Mu} \mid \mathbf{X}] = \mathbf{M}\mathbb{E}[\mathbf{u} \mid \mathbf{X}] = \mathbf{0}$

Using these facts, we obtain

$$\operatorname{cov}[\mathbf{Py}, \mathbf{My} \mid \mathbf{X}] = \operatorname{cov}[\mathbf{X}\boldsymbol{\beta} + \mathbf{Pu}, \mathbf{Mu} \mid \mathbf{X}] = \mathbb{E}\left[(\mathbf{X}\boldsymbol{\beta} + \mathbf{Pu})(\mathbf{Mu})' \mid \mathbf{X}\right]$$

From linearity of expectations and symmetry of $\mathbf{M}$, this becomes

$$\operatorname{cov}[\mathbf{Py}, \mathbf{My} \mid \mathbf{X}] = \mathbb{E}\left[\mathbf{X}\boldsymbol{\beta}\mathbf{u}'\mathbf{M} \mid \mathbf{X}\right] + \mathbb{E}\left[\mathbf{Puu}'\mathbf{M} \mid \mathbf{X}\right]$$

Regarding the first term on the right-hand side, we have

$$\mathbb{E}\left[\mathbf{X}\boldsymbol{\beta}\mathbf{u}'\mathbf{M} \mid \mathbf{X}\right] = \mathbf{X}\boldsymbol{\beta}\mathbb{E}\left[\mathbf{u}' \mid \mathbf{X}\right]\mathbf{M} = \mathbf{0}$$

Regarding the second term on the right-hand side, we have

$$\mathbb{E}\left[\mathbf{Puu}'\mathbf{M} \mid \mathbf{X}\right] = \mathbf{P}\mathbb{E}\left[\mathbf{uu}' \mid \mathbf{X}\right]\mathbf{M} = \mathbf{P}\sigma^2\mathbf{IM} = \sigma^2\mathbf{PM} = \mathbf{0}$$

$$\therefore \quad \operatorname{cov}[\mathbf{Py}, \mathbf{My} \mid \mathbf{X}] = \mathbf{0}$$

□

Solution to Exercise 7.6.5. Since $\mathbf{M}$ is a function of $\mathbf{X}$, we have $\mathbb{E}\left[\mathbf{Mu} \mid \mathbf{X}\right] = \mathbf{M}\mathbb{E}\left[\mathbf{u} \mid \mathbf{X}\right] = \mathbf{0}$. As a result,

$$\begin{aligned}
\operatorname{cov}[\hat{\boldsymbol{\beta}}, \hat{\mathbf{u}} \mid \mathbf{X}] &= \mathbb{E}\left[\hat{\boldsymbol{\beta}}\,\hat{\mathbf{u}}' \mid \mathbf{X}\right] - \mathbb{E}\left[\hat{\boldsymbol{\beta}} \mid \mathbf{X}\right]\mathbb{E}\left[\hat{\mathbf{u}} \mid \mathbf{X}\right]' \\
&= \mathbb{E}\left[\hat{\boldsymbol{\beta}}\,(\mathbf{Mu})' \mid \mathbf{X}\right] \\
&= \mathbb{E}\left[\boldsymbol{\beta}(\mathbf{Mu})' + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}\,(\mathbf{Mu})' \mid \mathbf{X}\right] \\
&= \mathbb{E}\left[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}\,(\mathbf{Mu})' \mid \mathbf{X}\right] \\
&= \mathbb{E}\left[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{uu}'\mathbf{M} \mid \mathbf{X}\right] \\
&= \sigma^2(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{M}
\end{aligned}$$

Since $\mathbf{X}'\mathbf{M} = (\mathbf{MX})' = \mathbf{0}'$ we have $\operatorname{cov}[\hat{\boldsymbol{\beta}}, \hat{\mathbf{u}} \mid \mathbf{X}] = \mathbf{0}$, and the proof is done. □

Solution to Exercise 7.6.6. Regarding the claim that $\mathbb{E}\left[\mathbf{Py} \mid \mathbf{X}\right] = \mathbf{X}\boldsymbol{\beta}$, our previous results and linearity of expectations gives

$$\mathbb{E}\left[\mathbf{Py} \mid \mathbf{X}\right] = \mathbb{E}\left[\mathbf{X}\boldsymbol{\beta} + \mathbf{Pu} \mid \mathbf{X}\right] = \mathbf{X}\boldsymbol{\beta} + \mathbf{P}\mathbb{E}\left[\mathbf{u} \mid \mathbf{X}\right] = \mathbf{X}\boldsymbol{\beta}$$

Regarding the claim that $\operatorname{var}[\mathbf{Py} \mid \mathbf{X}] = \sigma^2\mathbf{P}$, our rules for manipulating variances yield

$$\operatorname{var}[\mathbf{Py} \mid \mathbf{X}] = \operatorname{var}[\mathbf{X}\boldsymbol{\beta} + \mathbf{Pu} \mid \mathbf{X}] = \operatorname{var}[\mathbf{Pu} \mid \mathbf{X}] = \mathbf{P}\operatorname{var}[\mathbf{u} \mid \mathbf{X}]\mathbf{P}' = \mathbf{P}\sigma^2\mathbf{IP}'$$

Using symmetry and idempotence of $\mathbf{P}$, we obtain $\operatorname{var}[\mathbf{Py} \mid \mathbf{X}] = \sigma^2\mathbf{P}$. □

Solution to Exercise 7.6.7. Similar to the solution of exercise 7.6.6. □

Solution to Exercise 7.6.9. Since $\operatorname{rank}(\mathbf{X}) = \dim(\operatorname{rng}(\mathbf{X})) = K$, to establish that $\operatorname{rank}(\mathbf{P}) = \dim(\operatorname{rng}(\mathbf{P})) = K$, it suffices to show that $\operatorname{rng}(\mathbf{X}) = \operatorname{rng}(\mathbf{P})$. To see this, suppose first that $\mathbf{z} \in \operatorname{rng}(\mathbf{X})$. Since $\mathbf{P}$ is the projection onto $\operatorname{rng}(\mathbf{X})$, we then have $\mathbf{z} = \mathbf{P}\mathbf{z}$ (see theorem 3.1.3 on page 86), and hence $\mathbf{z} \in \operatorname{rng}(\mathbf{P})$. Conversely, if $\mathbf{z} \in \operatorname{rng}(\mathbf{P})$, then, $\mathbf{z} = \mathbf{P}\mathbf{a}$ for some $\mathbf{a} \in \mathbb{R}^N$. By definition, $\mathbf{P}$ maps every point into $\operatorname{rng}(\mathbf{X})$, so we conclude that $\mathbf{z} \in \operatorname{rng}(\mathbf{X})$. □

Solution to Exercise 7.6.10. To see that the matrix $\mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2$ is invertible, note that, in view of idempotence and symmetry of $\mathbf{M}_1$,

$$\mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2 = \mathbf{X}_2'\mathbf{M}_1\mathbf{M}_1\mathbf{X}_2 = \mathbf{X}_2'\mathbf{M}_1'\mathbf{M}_1\mathbf{X}_2 = (\mathbf{M}_1\mathbf{X}_2)'\mathbf{M}_1\mathbf{X}_2$$

In view of fact 2.3.11, to show that this matrix is invertible, it suffices to show that the matrix is positive definite. So take any $\mathbf{a} \neq 0$. We need to show that

$$\mathbf{a}'(\mathbf{M}_1\mathbf{X}_2)'\mathbf{M}_1\mathbf{X}_2\mathbf{a} = (\mathbf{M}_1\mathbf{X}_2\mathbf{a})'\mathbf{M}_1\mathbf{X}_2\mathbf{a} = \|\mathbf{M}_1\mathbf{X}_2\mathbf{a}\|^2 > 0$$

Since the only vector with zero norm is the zero vector, it now suffices to show that $\mathbf{M}_1\mathbf{X}_2\mathbf{a}$ is non-zero. From fact 3.1.6 on page 90, we see that $\mathbf{M}_1\mathbf{X}_2\mathbf{a} = \mathbf{0}$ only when $\mathbf{X}_2\mathbf{a}$ is in the column span of $\mathbf{X}_1$. Thus, the proof will be complete if we can show that $\mathbf{X}_2\mathbf{a}$ is not in the column span of $\mathbf{X}_1$.

Indeed, $\mathbf{X}_2\mathbf{a}$ is not in the column span of $\mathbf{X}_1$. For if it were, then we could write $\mathbf{X}_1\mathbf{b} = \mathbf{X}_2\mathbf{a}$ for some $\mathbf{b} \in \mathbb{R}^{K_1}$. Rearranging, we get $\mathbf{X}\mathbf{c} = \mathbf{0}$ for some non-zero $\mathbf{c}$ (recall $\mathbf{a} \neq 0$). This contradicts linear independence of the columns of $\mathbf{X}$. □

Solution to Exercise 7.6.11. Repeating (7.15) we have

$$\hat{\beta}_k = \beta_k + (\operatorname{col}_k(\mathbf{X})'\mathbf{M}_1 \operatorname{col}_k(\mathbf{X}))^{-1} \operatorname{col}_k(\mathbf{X})'\mathbf{M}_1\mathbf{u} \tag{7.30}$$

Since β_k is constant, taking the variance of (7.30) conditional on $\mathbf{X}$ we obtain

$$\operatorname{var}[\hat{\beta}_k \mid \mathbf{X}] = \operatorname{var}[\mathbf{A}\mathbf{u} \mid \mathbf{X}] \quad \text{where } \mathbf{A} := (\operatorname{col}_k(\mathbf{X})'\mathbf{M}_1 \operatorname{col}_k(\mathbf{X}))^{-1} \operatorname{col}_k(\mathbf{X})'\mathbf{M}_1$$

Since $\mathbf{A}$ is a function of $\mathbf{X}$, we can treat it as constant given $\mathbf{X}$, and we obtain

$$\operatorname{var}[\hat{\beta}_k \mid \mathbf{X}] = \mathbf{A} \operatorname{var}[\mathbf{u} \mid \mathbf{X}]\mathbf{A}' = \mathbf{A}\sigma^2\mathbf{I}\mathbf{A}' = \sigma^2\mathbf{A}\mathbf{A}'$$

To complete the proof, we just observe that

$$\begin{aligned}\mathbf{A}\mathbf{A}' &= (\operatorname{col}_k(\mathbf{X})'\mathbf{M}_1 \operatorname{col}_k(\mathbf{X}))^{-1} \operatorname{col}_k(\mathbf{X})'\mathbf{M}_1\mathbf{M}_1' \operatorname{col}_k(\mathbf{X})(\operatorname{col}_k(\mathbf{X})'\mathbf{M}_1 \operatorname{col}_k(\mathbf{X}))^{-1} \\ &= (\operatorname{col}_k(\mathbf{X})'\mathbf{M}_1 \operatorname{col}_k(\mathbf{X}))^{-1}\end{aligned}$$

where the last equality is due to symmetry and idempotence of $\mathbf{M}_1$ (recall that $\mathbf{M}_1$ is an annihilator). We conclude that (7.16) is valid. □

Solution to Exercise 7.6.12. For the model $\mathbf{y} = \beta_1\mathbf{1} + \beta_2\mathbf{x} + \mathbf{u}$, the matrix $\mathbf{M}_1$ is the centering annihilator $\mathbf{M}_c$ associated with $\mathbf{1}$, and $\operatorname{col}_k(\mathbf{X})$ is just $\mathbf{x}$. Hence, from (7.16), we have

$$\operatorname{var}[\hat{\beta}_2 \mid \mathbf{X}] = \sigma^2(\mathbf{x}'\mathbf{M}_c\mathbf{x})^{-1}$$

Using symmetry and idempotence of $\mathbf{M}_c$, this becomes

$$\operatorname{var}[\hat{\beta}_2 \mid \mathbf{X}] = \sigma^2[(\mathbf{M}_c\mathbf{x})'\mathbf{M}_c\mathbf{x}]^{-1} = \sigma^2 / \sum_{n=1}^{N}(x_n - \bar{x})^2$$

Finally, since the only random variables in $\mathbf{X}$ are the random variables in $\mathbf{x}$, we can write

$$\operatorname{var}[\hat{\beta}_2 \mid \mathbf{x}] = \sigma^2 / \sum_{n=1}^{N}(x_n - \bar{x})^2$$

as was to be shown. □

Solution to Exercise 7.6.13. We need to show that in the special case (7.24) we have

$$\frac{(\text{RSSR} - \text{USSR})/K_2}{\text{USSR}/(N-K)} = \frac{R_c^2}{1-R_c^2}\frac{N-K}{K_2}$$

or, equivalently,

$$\frac{\text{RSSR} - \text{USSR}}{\text{USSR}} = \frac{R_c^2}{1-R_c^2} \tag{7.31}$$

Consider first the left-hand side of (7.31). In the case of (7.24), this becomes

$$\frac{\text{RSSR} - \text{USSR}}{\text{USSR}} = \frac{\|\mathbf{M}_c\mathbf{y}\|^2 - \|\mathbf{M}\mathbf{y}\|^2}{\|\mathbf{M}\mathbf{y}\|^2}$$

On the other hand, regarding the right-hand side of (7.31), the definition of R_c^2 and some minor manipulation gives

$$\frac{R_c^2}{1-R_c^2} = \frac{\|\mathbf{P}\mathbf{M}_c\mathbf{y}\|^2}{\|\mathbf{M}_c\mathbf{y}\|^2 - \|\mathbf{P}\mathbf{M}_c\mathbf{y}\|^2}$$

Hence, to establish (7.31), we need to show that

$$\frac{\|\mathbf{M}_c\mathbf{y}\|^2 - \|\mathbf{M}\mathbf{y}\|^2}{\|\mathbf{M}\mathbf{y}\|^2} = \frac{\|\mathbf{P}\mathbf{M}_c\mathbf{y}\|^2}{\|\mathbf{M}_c\mathbf{y}\|^2 - \|\mathbf{P}\mathbf{M}_c\mathbf{y}\|^2}$$

This is can be established using (6.18). □

Solution to Exercise 7.6.14. We have shown previously that $\mathbf{M}\mathbf{y} = \mathbf{M}\mathbf{u}$ and $\mathbf{P}\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{P}\mathbf{u}$. Since we are conditioning on $\mathbf{X}$ we can treat it as constant. When $\mathbf{X}$ is constant, $\mathbf{P}$, $\mathbf{M}$ and $\mathbf{X}\boldsymbol{\beta}$ are all constant. Since linear (or affine) transformations of normal random vectors are normal, $\mathbf{M}\mathbf{u}$ and $\mathbf{X}\boldsymbol{\beta} + \mathbf{P}\mathbf{u}$ are both normally distributed.

It remains to show that $\mathbf{P}\mathbf{y}$ and $\mathbf{M}\mathbf{y}$ are independent given $\mathbf{X}$. Since they are normally distributed given $\mathbf{X}$, we need only show that they are uncorrelated given $\mathbf{X}$. This was already proved in exercise 7.6.4. □

Chapter 8

Time Series Models

The purpose of this chapter is to move away from the IID restriction and towards the study of data that has some dependence structure over time. Such data is very common in economics and finance. Our first step is to introduce and study common time series models. Next, we use the techniques developed in this process to move from finite sample OLS to large sample OLS. Large sample OLS theory is in many ways more attractive and convincing than its finite sample counterpart.

8.1 Some Common Models

In this section we introduce some of the most common time series models, working from specific to more general.

8.1.1 Linear Models

In time series as in other fields, the easiest models to analyze are the linear models. Of the linear time series models, the friendliest is the scalar Gaussian AR(1) model, which takes the form

$$x_{t+1} = \alpha + \rho x_t + w_{t+1} \quad \text{with } \{w_t\} \overset{\text{IID}}{\sim} \mathcal{N}(0, \sigma^2) \text{ and } x_0 \text{ given} \tag{8.1}$$

Here α, ρ and σ are parameters. The random variable x_t is called the **state variable**. Note that (8.1) fully defines the time t state x_t as a random variable for each t. (We'll spell out how this works in some detail below.) Since x_t is a well-defined random

variable, it has a distribution Π_t defined by $\Pi_t(s) := \mathbb{P}\{x_t \leq s\}$. This distribution is often called the **marginal distribution** of x_t. In what follows, we will be interested in the dynamics both of the state process $\{x_t\}$ and also of the corresponding distribution sequence $\{\Pi_t\}$.

The scalar Gaussian AR(1) model (8.1) generalizes in serveral different directions. For example, we can remove the assumption that the shocks are normally distributed, in which case it is simply called the scalar AR(1) model. The scalar AR(1) model further generalizes to $\mathbb{R}^K$, yielding the vector AR(1) model, or VAR(1). The dynamics of the process are given by

$$\mathbf{x}_{t+1} = \mathbf{a} + \mathbf{\Lambda}\mathbf{x}_t + \mathbf{w}_{t+1} \quad \text{with } \{\mathbf{w}_t\} \overset{\text{IID}}{\sim} \phi \text{ and } \mathbf{x}_0 \text{ given} \tag{8.2}$$

where $\mathbf{a}$ is a $K \times 1$ column vector, $\mathbf{\Lambda}$ is a $K \times K$ matrix and ϕ is some distribution on $\mathbb{R}^K$. If $\phi = \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ for some a symmetric, positive definite $K \times K$ matrix $\mathbf{\Sigma}$, then (8.2) is called the Gaussian VAR(1). The vector $\mathbf{x}_t$ is called the **state vector**, and the space $\mathbb{R}^K$ in which it takes values is called the **state space**.

Another common generalization of the scalar AR(1) model is the scalar AR(p) model, where the next state x_{t+1} is a (linear) function not just of the current state x_t, but of the last p previous states. For example, the AR(2) process has dynamics

$$x_{t+1} = \alpha + \rho x_t + \gamma x_{t-1} + w_{t+1}$$

Although x_{t+1} is a function of *two* lagged states, x_t and x_{t-1}, we can reformulate it as a first order model. To begin, we define an additional state variable y_t via $y_t = x_{t-1}$. The dynamics can then be expressed as

$$\begin{aligned} x_{t+1} &= \alpha + \rho x_t + \gamma y_t + w_{t+1} \\ y_{t+1} &= x_t \end{aligned}$$

We can write this in matrix form as

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} \rho & \gamma \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_t \\ y_t \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} w_{t+1}$$

Notice that this is a special case of the VAR(1) model in (8.2). The message is that higher order processes can be reduced to first order processes by increasing the number of state variables. For this reason, in what follows we concentrate primarily on first order models.

8.1.2 Nonlinear Models

The previous examples are all linear models. Linear models are simple, but this is not always a good thing. Their simplicity means they cannot always capture the kinds of dynamics we observe in data. Moreover, many theoretical modeling exercises produce models that are not linear. In this section we introduce several popular nonlinear models.

One well-known nonlinear model is the p-th order autoregressive conditional heteroskedasticity model (ARCH(p) model), the ARCH(1) version of which has dynamics

$$x_{t+1} = (\alpha_0 + \alpha_1 x_t^2)^{1/2} w_{t+1}, \qquad \{w_t\} \overset{\text{IID}}{\sim} \mathcal{N}(0,1) \tag{8.3}$$

The model arose as an effort to model evolution of returns on a given asset. In the model, returns x_t are first written as the product of an IID shock w_t and a time-varying volatility component σ_t. That is, $x_t = \sigma_t w_t$. The evolution of σ_t is specified by $\sigma_{t+1}^2 = \alpha_0 + \alpha_1 x_t^2$. Combining these equations gives the dynamics for x_t displayed in (8.3).

In recent years, econometricians studying asset prices have moved away from the ARCH model in favor of a generalized ARCH (GARCH) model, the simplest of which is the GARCH(1,1) process

$$\begin{aligned} x_t &= \sigma_t w_t \\ \sigma_{t+1}^2 &= \alpha_0 + \alpha_1 x_t^2 + \alpha_2 \sigma_t^2 \end{aligned}$$

Another popular nonlinear model is the smooth transition threshold autoregression (STAR) model

$$x_{t+1} = g(x_t) + w_{t+1} \tag{8.4}$$

where g is of the form

$$g(s) := (\alpha_0 + \rho_0 s)(1 - \tau(s)) + (\alpha_1 + \rho_1 s)\tau(s)$$

Here $\tau \colon \mathbb{R} \to [0,1]$ is an increasing function satisfying $\lim_{s \to -\infty} \tau(s) = 0$ and $\lim_{s \to \infty} \tau(s) = 1$. When s is small we have $\tau(s) \approx 0$, and $g(s) \approx \alpha_0 + \rho_0 s$. When s is large we have $\tau(s) \approx 1$, and $g(s) \approx \alpha_1 + \rho_1 s$. Thus, the dynamics transition between two different linear models, with the smoothness of the transition depending on the shape of τ.

8.1.3 Markov Models

All of the previous examples in this section are special cases of a general class of process called Markov processes. The general formulation for a (first order, time homogeneous) **Markov process** looks as follows:

$$\mathbf{x}_{t+1} = G(\mathbf{x}_t, \mathbf{w}_{t+1}) \quad \text{with} \quad \mathbf{x}_0 \sim \Pi_0 \tag{8.5}$$

Here we assume that $\{\mathbf{w}_t\}_{t\geq 1}$ is an IID sequence of $\mathbb{R}^M$-valued shocks with common density ϕ, and that G is a given function mapping the current state $\mathbf{x}_t \in \mathbb{R}^K$ and shock $\mathbf{w}_{t+1} \in \mathbb{R}^M$ into the new state $\mathbf{x}_{t+1} \in \mathbb{R}^K$. The initial condition $\mathbf{x}_0$ and the shocks $\{\mathbf{w}_t\}_{t\geq 1}$ are also assumed to be independent of each other. The density Π_0 is the distribution of $\mathbf{x}_0$. As before, we'll let

$$\Pi_t(\mathbf{s}) := \mathbb{P}\{\mathbf{x}_t \leq \mathbf{s}\}$$

represent the marginal distribution of the state $\mathbf{x}_t$. Where necessary, π_t will represent the corresponding density.

By repeated use of (8.5), we obtain the sequence of expressions

$$\begin{aligned}
\mathbf{x}_1 &= G(\mathbf{x}_0, \mathbf{w}_1) \\
\mathbf{x}_2 &= G(G(\mathbf{x}_0, \mathbf{w}_1), \mathbf{w}_2) \\
\mathbf{x}_3 &= G(G(G(\mathbf{x}_0, \mathbf{w}_1), \mathbf{w}_2), \mathbf{w}_3) \\
&\vdots
\end{aligned}$$

Continuing in this fashion, we see that, for any t, the state vector $\mathbf{x}_t$ can be written as a function of $\mathbf{x}_0$ and the shocks $\mathbf{w}_1, \ldots, \mathbf{w}_t$. In other words, for each t, there exists a function H_t such that

$$\mathbf{x}_t = H_t(\mathbf{x}_0, \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_t) \tag{8.6}$$

Although there may be no neat expression for the function H_t, equation (8.6) clarifies the fact that (8.5) pins down each $\mathbf{x}_t$ as a well-defined random variable, depending on the initial condition and the shocks up until date t.

Example 8.1.1. A simple example is the scalar linear AR(1) process

$$x_{t+1} = \alpha + \rho x_t + w_{t+1}$$

In this case, there is a neat expression for H_t in (8.6). Indeed, for all $t \geq 0$ we have

$$x_t = \alpha \sum_{k=0}^{t-1} \rho^k + \sum_{k=0}^{t-1} \rho^k w_{t-k} + \rho^t x_0 \tag{8.7}$$

The proof is an exercise.

Fact 8.1.1. For the Markov process (8.5), the current state $\mathbf{x}_t$ and future shocks $\mathbf{w}_{t+j}$ are independent for every t and every $j > 0$.

Fact 8.1.1 follows from fact 2.4.1 on page 72. In particular, $\mathbf{x}_t$ is a function of the random variables $\mathbf{x}_0, \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_t$, and these are all, by the IID assumption, independent of $\mathbf{w}_{t+j}$.

It's worth mentioning that, although it's often not made explicit, behind the all random vectors $\{\mathbf{x}_t\}$ lies a single sample space Ω and a probability $\mathbb{P}$. The idea is that an element ω of Ω is selected by "nature" at the start of the "experiment" (with the probability that ω lies in $E \subset \Omega$ equal to $\mathbb{P}(E)$). This determines the initial condition $\mathbf{x}_0$ and the shocks $\mathbf{w}_t$ as

$$\mathbf{x}_0(\omega), \mathbf{w}_1(\omega), \mathbf{w}_2(\omega), \mathbf{w}_3(\omega), \ldots$$

From these, each state vector $\mathbf{x}_t$ is determined via

$$\mathbf{x}_t(\omega) = H_t(\mathbf{x}_0(\omega), \mathbf{w}_1(\omega), \mathbf{w}_2(\omega), \ldots, \mathbf{w}_t(\omega))$$

where H_t is the function in (8.6).

An important object in Markov process theory is the **transition density**, or stochastic kernel, which is the conditional density of the next period state given the current state, and will be denoted by p.[1] In particular,

$$p(\cdot \,|\, \mathbf{s}) := \text{ the conditional density of } \mathbf{x}_{t+1} \text{ given } \mathbf{x}_t \text{ equals } \mathbf{s} \tag{8.8}$$

We can usually derive an expression for the transition density in terms of the model. For example, suppose that the shock is additive, so that

$$\mathbf{x}_{t+1} = G(\mathbf{x}_t, \mathbf{w}_{t+1}) = g(\mathbf{x}_t) + \mathbf{w}_{t+1} \quad \text{with} \quad \{\mathbf{w}_t\} \overset{\text{IID}}{\sim} \phi \tag{8.9}$$

for some function g and density ϕ. In this case, the transition density has the form

$$p(\mathbf{s}' \,|\, \mathbf{s}) = \phi(\mathbf{s}' - g(\mathbf{s})) \tag{8.10}$$

How does one arrive at expression (8.10)? Let's go through the argument for the scalar case, where the model is $x_{t+1} = g(x_t) + w_{t+1}$ with $w_{t+1} \sim \phi$. Let Φ be the

[1] I'm being a little careless here, because this density may not in fact exist. (For example, take the process $x_{t+1} = G(x_t, w_{t+1})$ where $G(s, w) = 0$ for all s and w. In this case the random variable x_{t+1} is equal to zero with probability one. Such a random variable does not have a density. See the discussion in §1.2.2.) However, the density will exist in most applications, and in all cases we consider.

cdf corresponding to ϕ, so that the derivative of Φ is ϕ. Now, we claim that the density of x_{t+1} when x_t equals constant s is given by $p(s' \mid s) := \phi(s' - g(s))$. This is equivalent to the claim that the density of $x' = g(s) + w$ is equal to $\phi(s' - g(s))$. Letting F be the cdf of $x' = g(s) + w$, we have

$$F(s') := \mathbb{P}\{x' \leq s'\} = \mathbb{P}\{g(s) + w \leq s'\} = \mathbb{P}\{w \leq s' - g(s)\} = \Phi(s' - g(s))$$

The density we seek is the derivative of this expression with respect to s', which is $\phi(s' - g(s))$. In other words, $p(s' \mid s) = \phi(s' - g(s))$ as claimed.

8.1.4 Martingales

Loosely speaking, a martingale is a stochastic process evolving over time such that the best guess of the next value given the current value is the current value. Martingales arise naturally in many kinds of economic and financial models. Moreover, since the mid 20th Century, martingales have contributed to much progress in the foundations of probability theory.

To give a more formal definition, we need first to introduce the notion of a filtration, which is an increasing sequence of information sets. Recall from §3.3.2 that an information set is just a set of random variables or vectors. Let $\{\mathcal{F}_t\}$ be a sequence of information sets. That is, $\mathcal{F}_t$ is an information set for each t. The sequence $\{\mathcal{F}_t\}$ is called a **filtration** if, in addition, it satisfies $\mathcal{F}_t \subset \mathcal{F}_{t+1}$ for all t. Intuitively, $\mathcal{F}_t$ contains the information available at time t, and the requirement that the sequence be increasing reflects the idea that more and more information is revealed over time.[2]

Example 8.1.2. Let $\{\mathbf{x}_t\}$ be a sequence of random vectors, and let

$$\mathcal{F}_0 := \emptyset, \quad \mathcal{F}_1 := \{\mathbf{x}_1\}, \quad \mathcal{F}_2 := \{\mathbf{x}_1, \mathbf{x}_2\}, \quad \mathcal{F}_3 := \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}, \quad \cdots$$

Then $\{\mathcal{F}_t\}$ is a filtration. In fact this is the canonical example of a filtration.

Now let $\{m_t\}$ be a scalar stochastic process (i.e., a sequence of random variables) and let $\{\mathcal{F}_t\}$ be a filtration. We say that $\{m_t\}$ is **adapted** to the filtration $\{\mathcal{F}_t\}$ if m_t is $\mathcal{F}_t$-measurable for every t. (For the definition of measurability see §3.3.2.) In many applications, $\mathcal{F}_t$ represents the variables we know as of time t, and if $\{m_t\}$ is adapted to $\mathcal{F}_t$, then we can compute m_t at time t as well.

[2] If you learn measure theory, you will learn that $\{\mathcal{F}_t\}$ is actually best thought of as an increasing sequence of σ-algebras. A presentation along these lines is beyond the scope of these notes. However, the underlying meaning is almost identical.

Example 8.1.3. If $\{\mathcal{F}_t\}$ is the filtration defined by

$$\mathcal{F}_0 := \emptyset, \quad \mathcal{F}_1 := \{x_1\}, \quad \mathcal{F}_2 := \{x_1, x_2\}, \quad \mathcal{F}_3 := \{x_1, x_2, x_3\}, \quad \cdots$$

and $m_t := t^{-1} \sum_{j=1}^t x_j$, then $\{m_t\}$ is adapted to $\{\mathcal{F}_t\}$.

Fact 8.1.2. If $\{m_t\}$ is adapted to $\{\mathcal{F}_t\}$, then $\mathbb{E}[m_t \mid \mathcal{F}_{t+j}] = m_t$ for any $j \geq 0$.

Hopefully the reason is clear: By adaptedness, we know that m_t is $\mathcal{F}_t$-measurable. From the definition of a filtration and fact 3.3.2 on page 97 it follows that m_t is $\mathcal{F}_{t+j}$-measurable. The result in fact 8.1.2 now follows from fact 3.3.6 on page 100.

To define martingales we let $\{m_t\}$ be a sequence of random variables adapted to a filtration $\{\mathcal{F}_t\}$, and satisfying $\mathbb{E}[|m_t|] < \infty$ for all t. In this setting, we say that $\{m_t\}$ is a **martingale** with respect to $\{\mathcal{F}_t\}$ if

$$\mathbb{E}[m_{t+1} \mid \mathcal{F}_t] = m_t \quad \text{for all} \quad t$$

We say that $\{m_t\}$ is a **martingale difference sequence** with respect to $\{\mathcal{F}_t\}$ if

$$\mathbb{E}[m_{t+1} \mid \mathcal{F}_t] = 0 \quad \text{for all} \quad t$$

A martingale difference sequence is so named because if $\{m_t\}$ is a martingale with respect to $\{\mathcal{F}_t\}$, then $d_t = m_t - m_{t-1}$ is a martingale difference sequence with respect to $\{\mathcal{F}_t\}$. See exercise 8.5.8.

Note that the unconditional mean of a martingale difference sequence is always zero, because

$$\mathbb{E}[m_t] = \mathbb{E}[\mathbb{E}[m_t \mid \mathcal{F}_{t-1}]] = \mathbb{E}[0] = 0$$

Example 8.1.4. The classic example of a martingale is a random walk. Let $\{\eta_t\}$ be an IID sequence of random variables with $\mathbb{E}[\eta_1] = 0$, and let $m_t := \sum_{j=1}^t \eta_j$. For example, η_t might be the payoff on the t-th round of a game (e.g., poker), and m_t is the wealth of a gambler after the t-th round. (We are assuming that wealth starts at zero and may take arbitrarily large negative values without the gambler getting ejected from the game and knee-capped by the mafia.) In this case,

$$m_t = m_{t-1} + \eta_t = \sum_{j=1}^t \eta_j$$

is a martingale with respect to $\mathcal{F}_t := \{\eta_1, \ldots, \eta_t\}$. That $\{m_t\}$ is adapted to $\{\mathcal{F}_t\}$ follows immediately from the definition of m_t and $\mathcal{F}_t$. Moreover,

$$\begin{aligned}
\mathbb{E}\left[m_{t+1} \mid \mathcal{F}_t\right] &= \mathbb{E}\left[\eta_1 + \cdots \eta_t + \eta_{t+1} \mid \mathcal{F}_t\right] \\
&= \mathbb{E}\left[\eta_1 \mid \mathcal{F}_t\right] + \cdots \mathbb{E}\left[\eta_t \mid \mathcal{F}_t\right] + \mathbb{E}\left[\eta_{t+1} \mid \mathcal{F}_t\right] \\
&= \eta_1 + \cdots \eta_t + \mathbb{E}\left[\eta_{t+1} \mid \mathcal{F}_t\right] \\
&= \eta_1 + \cdots \eta_t + \mathbb{E}\left[\eta_{t+1}\right] \\
&= \eta_1 + \cdots \eta_t = m_t
\end{aligned}$$

Example 8.1.5. A famous example of a martingale in economic theory is Robert Hall's hypothesis that consumption is a martingale (Hall, 1978). To understand his hypothesis, consider an Euler equation of the form

$$u'(c_t) = \mathbb{E}_t\left[\frac{1 + r_{t+1}}{1 + \rho} u'(c_{t+1})\right]$$

where u' is the derivative of a utility function u, r_t is an interest rate and ρ is a discount factor. The "time t" expectation $\mathbb{E}_t[\cdot]$ can be thought of as a conditional expectation $\mathbb{E}\left[\cdot \mid \mathcal{F}_t\right]$, where $\mathcal{F}_t$ contains all variables observable at time t. Specializing to the case $r_{t+1} = \rho$ and $u(c) = c - ac^2/2$, the Euler equation reduces to

$$c_t = \mathbb{E}_t[c_{t+1}] =: \mathbb{E}\left[c_{t+1} \mid \mathcal{F}_t\right]$$

Thus, under the theory, consumption is a martingale with respect to $\{\mathcal{F}_t\}$.

8.2 Dynamic Properties

The time series models discussed above can display very different dynamics from the simple IID data processes considered earlier in this text. This has profound implications for asymptotic theory, such as the law of large numbers or central limit theorem. In this section we try to unravel some of the mysteries, starting with a very simple case.

8.2.1 A Simple Example: Linear AR(1)

Consider again the scalar Gaussian AR(1) process

$$x_{t+1} = \alpha + \rho x_t + w_{t+1} \quad \text{with} \quad \{w_t\} \overset{\text{IID}}{\sim} \mathcal{N}(0, 1)$$

For simplicity, the variance of the shock w_t has been set to one. In order to learn about the dynamics of this process, let's begin with some simulated time series and see what we observe. In all simulations, we take $\alpha = 1$.

Six individual time series $\{x_t\}$ are shown in figure 8.1, each generated using a different value of ρ. The code for generating the figure is given in listing 12. As suggested by the figure (experiment with the code to verify it for yourself), the simulated time paths are quite sensitive to the value of the coefficient ρ. Whenever ρ is outside the interval $(-1, 1)$, the series tend to diverge. If, on the other hand, $|\rho| < 1$, then the process does not diverge. For example, if you look at the time series for $\rho = 0.9$ in figure 8.1, you will see that, after an initial burn in period where the series is affected by the initial condition x_0, the process settles down to random motion within a band (between about 5 and 15 in this case).

Listing 12 Code for figure 8.1

```
# Generates an AR(1) time series starting from x = init
ar1ts <- function(init, n, alpha, rho) {
    x <- numeric(n)
    x[1] <- init
    w <- rnorm(n-1)
    for (t in 1:(n-1)) {
        x[t+1] <- alpha + rho * x[t] + w[t]
    }
    return(x)
}

rhos <- c(0.1, -0.1, 0.9, -0.9, 1.1, -1.1)
N <- 200
par(mfrow=c(3,2))  # Arrangement of figures
for (rho in rhos) {
    plot(ar1ts(0, N, 1, rho), type="l",
        xlab=paste("rho = ", rho), ylab="")
}
```

We can investigate this phenomenon analytically by looking at expression (8.7). Since the shocks $\{w_t\}$ are assumed to be normal, it follows from this expression and fact 1.2.6 on page 24 that x_t will be normally distributed whenever x_0 is either normal or constant. Let's assume that this is the case. In particular, let's assume that

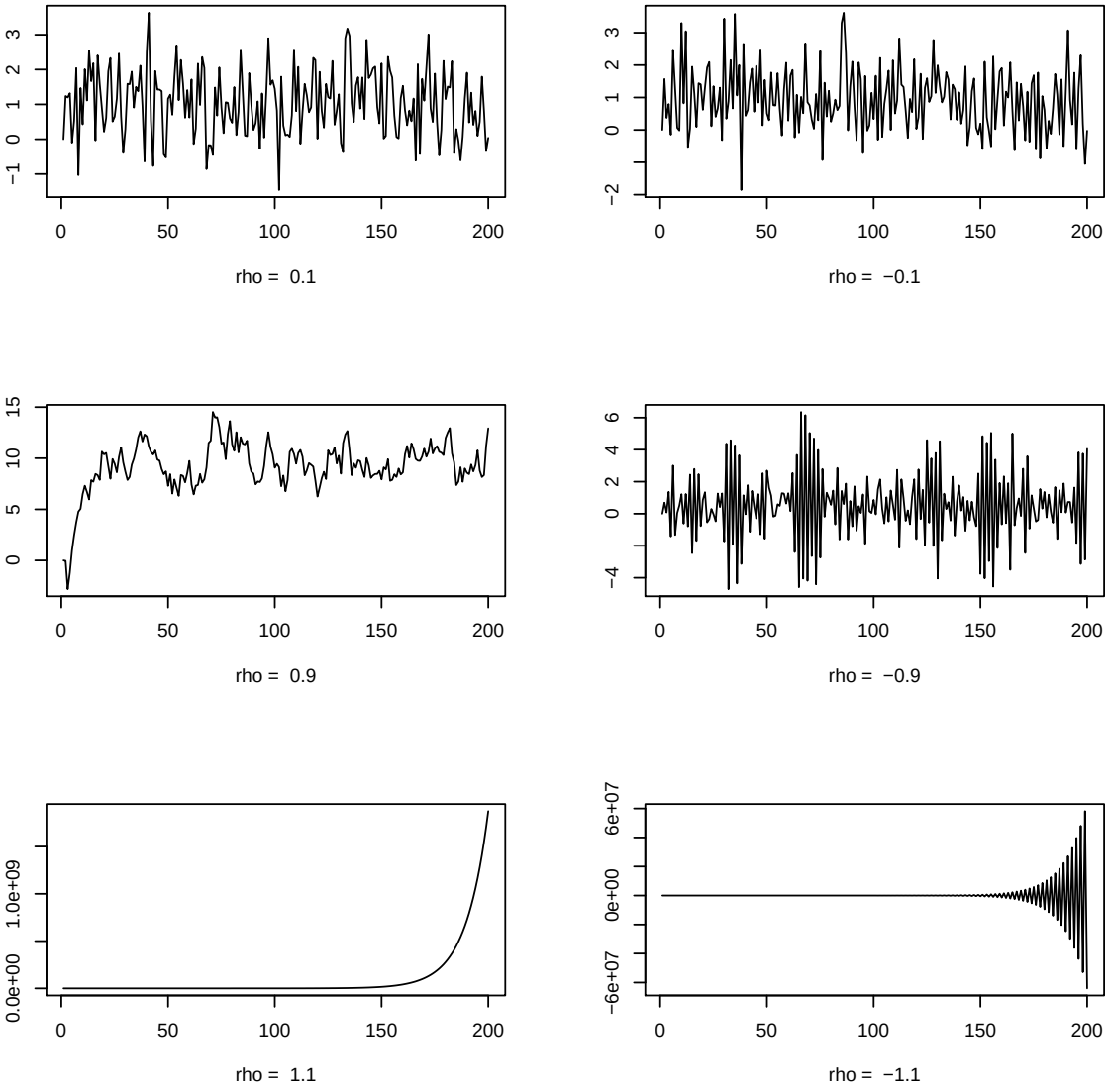


Figure 8.1: Dynamics of the linear AR(1) model

$x_0 \sim \mathcal{N}(\mu_0, \sigma_0^2)$, where μ_0 and σ_0 are given constants. Applying our usual rules for expectation and variance to (8.7), we can also see that

$$\mu_t := \mathbb{E}[x_t] = \alpha \sum_{k=0}^{t-1} \rho^k + \rho^t \mu_0 \quad \text{and} \quad \sigma_t^2 := \text{var}[x_t] = \sum_{k=0}^{t-1} \rho^{2k} + \rho^{2t} \sigma_0^2$$

Since x_t is normal and we've now found the mean and variance, we've pinned down the marginal distribution Π_t for x_t. In particular, we have shown that

$$\Pi_t = \mathcal{N}(\mu_t, \sigma_t^2) = \mathcal{N}\left(\alpha \sum_{k=0}^{t-1} \rho^k + \rho^t \mu_0, \sum_{k=0}^{t-1} \rho^{2k} + \rho^{2t} \sigma_0^2 \right)$$

Notice that if $|\rho| \geq 1$, then the mean and variance diverge. If, on the other hand, $|\rho| < 1$, then

$$\mu_t \to \mu_\infty := \frac{\alpha}{1-\rho} \quad \text{and} \quad \sigma_t^2 \to \sigma_\infty^2 := \frac{1}{1-\rho^2}$$

In this case, it seems likely that the marginal distribution $\Pi_t = \mathcal{N}(\mu_t, \sigma_t^2)$ of x_t converges weakly (see the definition in §2.5.2) to $\Pi_\infty := \mathcal{N}(\mu_\infty, \sigma_\infty^2)$. Using fact 1.4.3 on page 33, one can then show that this is indeed the case. That is,

$$\Pi_t = \mathcal{N}(\mu_t, \sigma_t^2) \xrightarrow{d} \Pi_\infty := \mathcal{N}(\mu_\infty, \sigma_\infty^2) := \mathcal{N}\left(\frac{\alpha}{1-\rho}, \frac{1}{1-\rho^2} \right) \tag{8.11}$$

Observe that this limit does not depend on the starting values μ_0 and σ_0^2. In other words, Π_∞ does not depend on Π_0.

Figures 8.2 and 8.3 illustrate convergence of Π_t to Π_∞, and of the corresponding densities π_t to π_∞, when $\alpha = 0$ and $\rho = 0.9$. The initial distribution in the figure is $\Pi_0 := \mathcal{N}(\mu_0, \sigma_0^2)$ with arbitrarily chosen constants $\mu_0 = -6$ and $\sigma_0^2 = 4.2$. For both the sequence of cdfs and the sequence of densities, convergence is from left to right. The code is given in listing 13, and if you experiment with different choices of μ_0 and σ_0, you will see that convergence to the same distribution Π_∞ always occurs. The fact that $\Pi_t \to \Pi_\infty$ for any choice of Π_0 is called **global stability**, or **ergodicity**. A more formal definition is given below.[3]

Besides being the limiting distribution of the sequence $\{\Pi_t\}$, the distribution Π_∞ has another special property: If we start with $\Pi_0 = \Pi_\infty$, then we will have $\Pi_t = \Pi_\infty$

[3]The term ergodicity is sometimes used to signify that the process satisfies the law of large numbers, as described in the next section. However, as will be discussed at length there, global stability and the law of large numbers are closely related.

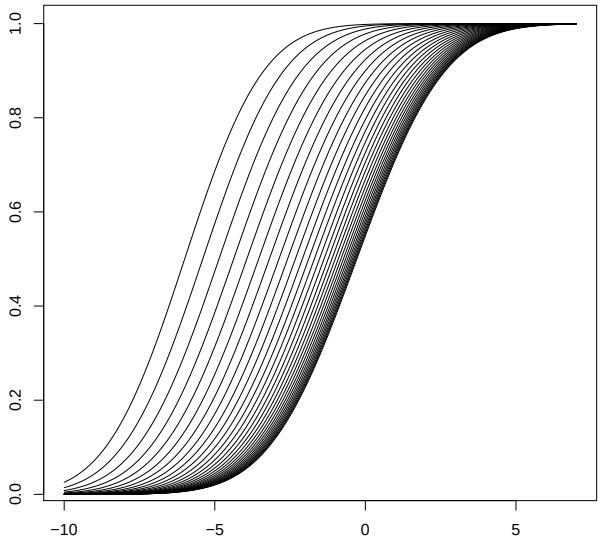


Figure 8.2: Convergence of cdfs

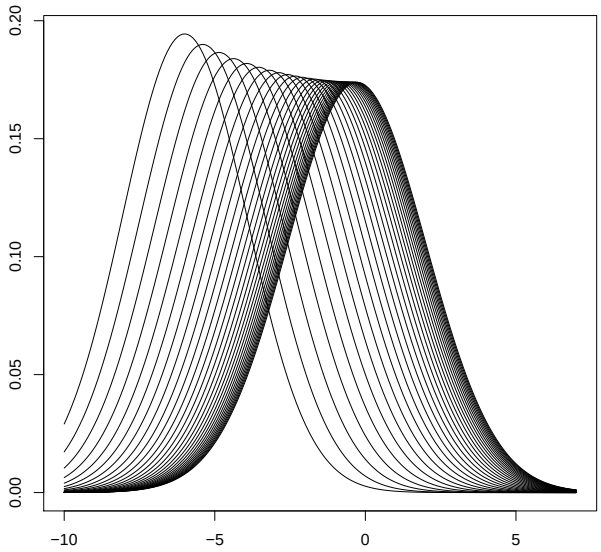


Figure 8.3: Convergence of densities

Listing 13 Code for figure 8.3

```
rho <- 0.9
N <- 30  # Number of densities to plot
mu <- -6; sigma2 <- 0.8 / (1 - rho^2)  # mu_0 and sigma^2_0

xgrid <- seq(-10, 7, length=200)
plot(xgrid, dnorm(xgrid, mean=mu, sd=sqrt(sigma2)),
     type="l", xlab="", ylab="", main="")

for (i in 2:N) {
     mu <- rho * mu
     sigma2 <- rho^2 * sigma2 + 1
     lines(xgrid, dnorm(xgrid, mean=mu, sd=sqrt(sigma2)))
}
```

for all t. For example, if, in figure 8.2 we had started at $\Pi_0 = \Pi_\infty$, then we would see only one curve, which corresponds to Π_∞. The sequence of distributions is constant. For this reason, Π_∞ is called the **stationary distribution** of the process. Note also that for our model, we have only one stationary distribution. In particular, if we start at *any* other cdf $\Pi_0 \neq \Pi_\infty$, then we will see motion in the figure as the sequence Π_t converges to Π_∞.

The fact that if $\Pi_0 = \Pi_\infty$, then $\Pi_t = \Pi_\infty$ for all t is a very important point, and as such it's worth checking analytically as well. The way to do this is by induction, showing that if $\Pi_t = \Pi_\infty$, then $\Pi_{t+1} = \Pi_\infty$ is also true.[4] To verify the latter, one can use the relation $x_{t+1} = \alpha + \rho x_t + w_{t+1}$. The details are left as an exercise (exercise 8.5.7).

Thus, taking $\Pi_0 = \Pi_\infty$ implies that x_t has the same marginal distribution Π_∞ for every t. In other words, the sequence of random variables $\{x_t\}$ is *identically distributed*. It is not, however, IID, because x_t and x_{t+j} are not independent (unless $\rho = 0$). We'll say more about this in just a moment.

[4]The logic is as follows: Suppose we know that (a) $\Pi_0 = \Pi_\infty$, and (b) $\Pi_t = \Pi_\infty$ implies $\Pi_{t+1} = \Pi_\infty$. Then (a) and (b) together imply that $\Pi_1 = \Pi_\infty$. Next, using (b) again, we get $\Pi_2 = \Pi_\infty$. Using (b) one more time we get $\Pi_3 = \Pi_\infty$, and so on. Hence $\Pi_t = \Pi_\infty$ for all t, as was to be shown.

8.2.2 Linear AR(1) Continued: The LLN

For consistency of statistical procedures, some version of the LLN is almost always necessary. The scalar and vector LLNs we have considered so far used the IID assumption. In particular, they required zero correlation between elements of the sequence. If we admit nonzero correlation, then, as we'll see below, the LLN can easily fail.

Fortunately, the global stability concept we have just investigated provides one way to obtain the LLN without IID data. Let's think about this how this connection might work, starting with the AR(1) model

$$x_{t+1} = \alpha + \rho x_t + w_{t+1} \quad \text{with} \quad \{w_t\} \overset{\text{IID}}{\sim} \mathcal{N}(0, \sigma^2)$$

Suppose for now that $\alpha = 0$, $\rho = 1$ and $\sigma = 0$. Setting $\sigma = 0$ means that $w_t \sim \mathcal{N}(0, 0)$, or, in other words, w_t is identically equal to zero. In this case, the dynamics reduce to $x_{t+1} = x_t$, and hence

$$x_t = x_{t-1} = x_{t-2} = \cdots = x_1$$

If x_0 has some given distribution Π_∞, then clearly

$$\Pi_t(s) := \mathbb{P}\{x_t \leq s\} = \mathbb{P}\{x_1 \leq s\} = \Pi_\infty(s) \quad \text{for all} \quad t$$

This tells us that $\{x_t\}$ is identically distributed, with common distribution Π_∞. (The distribution Π_∞ is stationary for this process, which is why I chose the symbol Π_∞. In fact *every* distribution is stationary for this process, because if we fix any distribution and let x_0 have that distribution, then x_t will have that same distribution for all t, as we just established.) However, the sequence $\{x_t\}$ does not satisfy the LLN. Indeed,

$$\bar{x}_T := \frac{1}{T} \sum_{t=1}^{T} x_t = \frac{1}{T} \sum_{t=1}^{T} x_1 = x_1$$

and x_1, being a random variable, does not generally converge to anything.[5]

So under what conditions does the sample mean of the AR(1) process converge to the common mean of x_t? One necessary condition is that the sequence $\{x_t\}$ does indeed have a common mean. So let's restrict attention to the case where the parameters α, ρ and σ are such that some stationary distribution Π_∞ does exist,[6] and

[5] The only exception is if x_0 is a degenerate random variable, putting all its probability mass on a single point. If this is not clear, go back to exercise 1.5.28.

[6] For example, if $|\rho| \geq 1$ and $\sigma > 0$, then no stationary distribution exists. If $|\rho| < 1$, then a unique stationary distribution always exists, for any values of α and σ.

start the process off with $\Pi_0 = \Pi_\infty$, so that $\Pi_t = \Pi_\infty$ for all t. Hence $\{x_t\}$ is identically distributed. We want to know when

$$\bar{x}_T \xrightarrow{p} \mathbb{E}[x_t] = \mu_\infty := \int s \Pi_\infty(ds) \tag{8.12}$$

We saw in the previous example ($\alpha = 0$, $\rho = 1$, $\sigma = 0$) that the mere fact that $\{x_t\}$ is identically distributed is not enough. We need something more. To investigate this, let's recall the proof of the LLN for IID sequences in theorem 1.4.1 (page 34). In the theorem, we saw that when $\{x_n\}$ is an IID sequence of random variables with common mean μ and variance σ^2, we have

$$\bar{x}_N := \frac{1}{N} \sum_{n=1}^{N} x_n \xrightarrow{p} \mathbb{E}[x_n] = \mu$$

For the proof, we just observe that since $\mathbb{E}[\bar{x}_N] = \mu$, fact 1.4.2 on page 31 implies that the result will hold whenever $\operatorname{var}[\bar{x}_N] \to 0$ as $N \to \infty$. In view of fact 1.3.9 on page 28, we have

$$\begin{aligned} \operatorname{var}\left[\frac{1}{N}\sum_{n=1}^{N} x_n\right] &= \frac{1}{N^2}\sum_{n=1}^{N} \operatorname{var}[x_n] + \frac{2}{N^2}\sum_{n<m} \operatorname{cov}[x_n, x_m] \\ &= \frac{\sigma^2}{N} + \frac{2}{N^2}\sum_{n<m} \operatorname{cov}[x_n, x_m] \end{aligned}$$

In the IID case, $\operatorname{cov}[x_n, x_m] = 0$ for all $n < m$, and hence the convergence $\operatorname{var}[\bar{x}_N] \to 0$ does indeed hold.

Now let's weaken the assumption of zero correlation, and try to think about whether the LLN can be salvaged. When correlation is non-zero, the question of whether or not $\operatorname{var}[\bar{x}_N] \to 0$ depends whether or not most of the terms $\operatorname{cov}[x_n, x_m]$ are small. This will be the case if the covariances die out relatively quickly, so that $\operatorname{cov}[x_n, x_{n+j}] \approx 0$ when j is large. Furthermore, the property that correlations die out over time is closely related with global stability. For example, let's take $\alpha = 0$ and $\sigma = 1$, so the dynamics are

$$x_{t+1} = \rho x_t + w_{t+1} \quad \text{with} \quad \{w_t\} \overset{\text{IID}}{\sim} \mathcal{N}(0,1) \tag{8.13}$$

We saw in (8.11) that, under the assumption $-1 < \rho < 1$, the model has a unique, globally stable stationary distribution given by

$$\Pi_\infty := \mathcal{N}\left(0, \frac{1}{1-\rho^2}\right)$$

It turns out that the stability condition $-1 < \rho < 1$ is precisely what we need for the covariances to die out. Indeed, fixing $j \geq 1$ and iterating with (8.13), we obtain

$$x_{t+j} = \rho^j x_t + \sum_{k=1}^{j} \rho^{j-k} w_{t+k}$$

we now have

$$\begin{aligned}
\operatorname{cov}[x_{t+j}, x_t] = \mathbb{E}[x_{t+j} x_t] &= \mathbb{E}\left[\left(\rho^j x_t + \sum_{k=1}^{j} \rho^{j-k} w_{t+k}\right) x_t\right] \\
&= \rho^j \mathbb{E}[x_t^2] + \sum_{k=1}^{j} \rho^{j-k} \mathbb{E}[w_{t+k} x_t] \\
&= \rho^j \mathbb{E}[x_t^2] + \sum_{k=1}^{j} \rho^{j-k} \mathbb{E}[w_{t+k}] \mathbb{E}[x_t] \\
&= \rho^j \mathbb{E}[x_t^2]
\end{aligned}$$

(In the second last equality, we used the fact that x_t depends only on current and lagged shocks (see (8.7) on page 229), and hence x_t and w_{t+k} are independent.) Since $|\rho| < 1$ we now have

$$\operatorname{cov}[x_{t+j}, x_t] = \rho^j \mathbb{E}[x_t^2] = \frac{\rho^j}{1 - \rho^2} \to 0 \quad \text{as} \quad j \to \infty$$

To summarize our discussion, to make the LLN work we need covariances to go to zero as we compare elements of the sequence that are more and more separated in time. The condition for covariances going to zero is exactly the condition we require for global stability (i.e., $|\rho| < 1$).

The LLN result (8.12) in the AR(1) model (8.13) is illustrated in figure 8.4. In the simulation, we set $\rho = 0.8$. Note that for this model, the mean $\int s \Pi_\infty(ds)$ of Π_∞ is zero. The grey line is a simulated time series of the process. The blue line is the plot of $\bar{x}_T$ against T. As anticipated by the LLN, we see that $\bar{x}_T$ converges to zero. The code for producing figure 8.4 is given in listing 14.

8.2.3 Markov Process Dynamics

Studying the dynamics of the AR(1) process has helped us build intuition, but now we need to look at more general (and complex) cases. Let's look again at the Markov

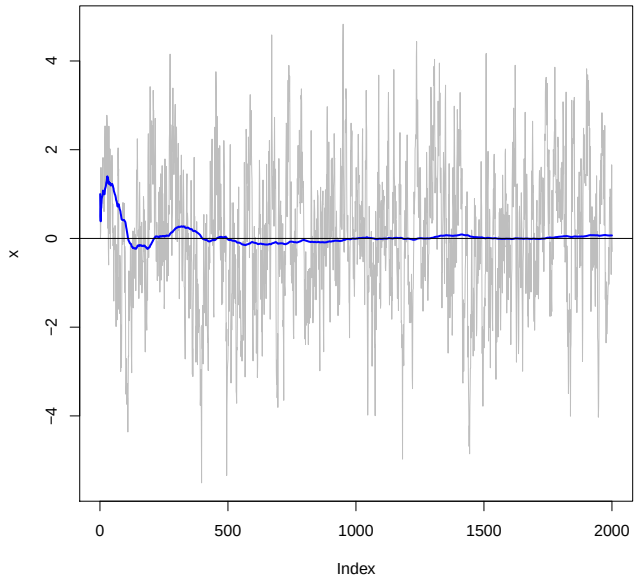


Figure 8.4: LLN in the AR(1) case

Listing 14 Code for figure 8.4

```
rho = 0.8
N <- 2000

x <- numeric(N)
sample_mean <- numeric(N)
x[1] <- 1
for (t in 2:N) x[t+1] <- rho * x[t] + rnorm(1)
for (T in 1:N) sample_mean[T] <- sum(x[1:T] / T)

plot(x, col="gray", type="l")
lines(sample_mean, col="blue", lwd=2)
abline(0, 0)
```

process (8.5) on page 229, which includes the AR(1) model as a special case. For each $t \geq 1$, let π_t denote the density of $\mathbf{x}_t$. In particular, for $B \subset \mathbb{R}^K$, we have

$$\int_B \pi_t(\mathbf{s})\, d\mathbf{s} = \mathbb{P}\{\mathbf{x}_t \in B\} = \mathbb{P}\{H_t(\mathbf{x}_0, \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_t) \in B\}$$

where H_t is defined on page 229.[7] Let's work towards finding conditions for global stability, so that this sequence of densities will converge to a unique limit for all initial π_0. First we need some definitions. These definitions include and formalize the stability-related definitions given for the scalar AR(1) model.

Let's start with the definition of stationary distributions. Let $p(\cdot \,|\, \mathbf{s})$ be the transition density of the process, as defined in (8.8). Given p, a density π_∞ on $\mathbb{R}^K$ is called **stationary** for the process (8.15) if

$$\pi_\infty(\mathbf{s}') = \int p(\mathbf{s}' \,|\, \mathbf{s}) \pi_\infty(\mathbf{s})\, d\mathbf{s} \quad \text{for all } \mathbf{s}' \in \mathbb{R}^K \tag{8.14}$$

A stationary density π_∞ is called **globally stable** if the sequence of marginal densities $\{\pi_t\}_{t=0}^\infty$ converges to π_∞ for any choice of initial density π_0.

The interpretation of (8.14) is as follows. Suppose that $\mathbf{x}_t \sim \pi_\infty$. Informally, the probability that $\mathbf{x}_{t+1} = \mathbf{s}'$ should be equal to the probability that $\mathbf{x}_{t+1} = \mathbf{s}'$ given $\mathbf{x}_t = \mathbf{s}$, summed over all possible $\mathbf{s}$ and weighted by the probability that $\mathbf{x}_t = \mathbf{s}$. This is the right-hand side of (8.14). The equality in (8.14) says that this probability is $\pi_\infty(\mathbf{s}')$. In other words, $\mathbf{x}_{t+1} \sim \pi_\infty$ whenever $\mathbf{x}_t \sim \pi_\infty$. This is the reason π_∞ is called a "stationary" distribution.

As a consequence of the preceding discussion, we have the following result:

Fact 8.2.1. Let $\{\mathbf{x}_t\}$ be a Markov process with stationary distribution π_∞. If $\mathbf{x}_0 \sim \pi_\infty$, then $\{\mathbf{x}_t\}$ is *identically distributed*, with common marginal distribution π_∞.

While stability or instability of linear models like the AR(1) process is relatively easy to study, for the more general Markov model we are studying now, the analysis can be somewhat tricky. Here is a useful result pertaining to the additive shock model

$$\mathbf{x}_{t+1} = g(\mathbf{x}_t) + \mathbf{w}_{t+1} \tag{8.15}$$

Here $\{\mathbf{w}_t\}_{t \geq 1}$ is an IID sequence of $\mathbb{R}^K$-valued shocks with common density ϕ, and $\mathbf{x}_0$ has density π_0.

[7] In the presentation below we're going to work with densities rather than cdfs because the presentation is a little easier.

Theorem 8.2.1. *If the density ϕ has finite mean and is strictly positive everywhere on $\mathbb{R}^K$, the function g is continuous and there exist positive constants λ and L such that $\lambda < 1$ and*

$$\|g(\mathbf{s})\| \leq \lambda \|\mathbf{s}\| + L \quad \textit{for all } \mathbf{s} \in \mathbb{R}^K \tag{8.16}$$

then (8.15) has a unique stationary distribution that is globally stable.

Theorem 8.2.1 is a very handy result, although it is but one example of a condition ensuring global stability.[8] We'll look at how to check the conditions of theorem 8.2.1 in a moment, but before that let's consider the connection between this stability theorem and the LLN. We saw in the scalar AR(1) case that this connection is rather close. This caries over to the general Markov case treated here. Indeed, the stability conditions in theorem 8.2.1 are sufficient for the LLN:

Theorem 8.2.2. *Suppose that the conditions of theorem 8.2.1 hold. Let π_∞ be the unique stationary density. Let $\{\mathbf{x}_t\}$ be an generated by (8.15) with $\mathbf{x}_0 \sim \pi_\infty$. If $h\colon \mathbb{R}^K \to \mathbb{R}$ is any function such that $\int |h(\mathbf{s})| \pi_\infty(\mathbf{s}) d\mathbf{s} < \infty$, then*[9]

$$\frac{1}{T} \sum_{t=1}^{T} h(\mathbf{x}_t) \xrightarrow{p} \mathbb{E}\left[h(\mathbf{x}_t)\right] = \int h(\mathbf{s}) \pi_\infty(\mathbf{s}) d\mathbf{s} \quad \textit{as} \quad T \to \infty$$

To apply theorems 8.2.1 and 8.2.2, we need to be able to check the conditions of theorem 8.2.1. The best way to learn to do this is by looking at examples.

Example 8.2.1. Here's a variation on the scalar "threshold autoregressive" model:

$$x_{t+1} = \rho |x_t| + (1 - \rho^2)^{1/2} w_{t+1} \quad \text{with} \quad -1 < \rho < 1 \quad \text{and} \quad \{w_t\} \overset{\text{IID}}{\sim} \mathcal{N}(0,1)$$

The conditions of theorem 8.2.1 are satisfied. To see this, we can rewrite the model as

$$x_{t+1} = g(x_t) + v_{t+1}, \quad g(s) = \rho |s| \quad \text{and} \quad \{v_t\} \overset{\text{IID}}{\sim} \mathcal{N}(0, 1 - \rho^2)$$

Clearly the distribution $\mathcal{N}(0, 1 - \rho^2)$ of v_t has finite mean and a density that is everywhere positive on $\mathbb{R}$. Moreover, $|g(s)| = |\rho||s|$, so that (8.16) is satisfied with

[8] In fact the conditions of theorem 8.2.1 are rather strong (in order to make the statement of the theorem straightforward). There are many other conditions for this kind of stability, based on a variety of different criteria. For further discussion of the Markov case see Stachurski (2009) and the references therein.

[9] The condition that $\mathbf{x}_0 \sim \pi_\infty$ is actually unnecessary, but it means that $\mathbb{E}\left[h(\mathbf{x}_t)\right] = \int h(\mathbf{s}) \pi_\infty(\mathbf{s}) d\mathbf{s}$ for all t, which makes the result a little easier to digest. See Stachurski (2009) for a more formal discussion of this LLN, and references containing proofs.

$\lambda = |\rho|$ and $L = 0$. By assumption, we have $\lambda < 1$, and hence all the conditions of theorem 8.2.1 are satisfied, and a unique, globally stable stationary density exists. While for many Markov processes the stationary density has no known closed form solution, in the present case the stationary density is known to have the form $\pi_\infty(s) = 2\phi(s)\Phi(qs)$, where $q := \rho(1-\rho^2)^{-1/2}$, ϕ is the standard normal density and Φ is the standard normal cdf.

Example 8.2.2. Consider again the VAR(1) process from (8.2), with

$$\mathbf{x}_{t+1} = \mathbf{a} + \mathbf{\Lambda}\mathbf{x}_t + \mathbf{w}_{t+1} \tag{8.17}$$

To study the dynamics of this process, it's useful to recall the definition of the **spectral norm** of $\mathbf{\Lambda}$, which is given by[10]

$$\rho(\mathbf{\Lambda}) := \max_{\mathbf{s}\neq 0} \frac{\|\mathbf{\Lambda}\mathbf{s}\|}{\|\mathbf{s}\|}$$

Let us consider the drift condition (8.16) applied to the law of motion (8.17). In this case, $g(\mathbf{s}) = \mathbf{a} + \mathbf{\Lambda}\mathbf{s}$, and, using the triangle inequality for the norm (fact 2.1.1 on page 52), we have

$$\|g(\mathbf{s})\| = \|\mathbf{a} + \mathbf{\Lambda}\mathbf{s}\| \leq \|\mathbf{a}\| + \|\mathbf{\Lambda}\mathbf{s}\|$$

Let $\lambda := \rho(\mathbf{\Lambda})$ and let $L := \|\mathbf{a}\|$. We then have

$$\|g(\mathbf{s})\| \leq \|\mathbf{\Lambda}\mathbf{s}\| + L = \frac{\|\mathbf{\Lambda}\mathbf{s}\|}{\|\mathbf{s}\|}\|\mathbf{s}\| + L \leq \rho(\mathbf{\Lambda})\|\mathbf{s}\| + L =: \lambda\|\mathbf{s}\| + L$$

We can now see that the drift condition (8.16) will be satisfied whenever the spectral norm of $\mathbf{\Lambda}$ is less than one.

Example 8.2.3. Consider the STAR model introduced in §8.1.2, where the function g is given by

$$g(s) := (\alpha_0 + \rho_0 s)(1 - \tau(s)) + (\alpha_1 + \rho_1 s)\tau(s)$$

and $\tau\colon \mathbb{R} \to [0,1]$. Applying the triangle inequality $|a+b| \leq |a| + |b|$, we obtain

$$\begin{aligned} |g(s)| &\leq |(\alpha_0 + \rho_0 s)(1 - \tau(s))| + |(\alpha_1 + \rho_1 s)\tau(s)| \\ &\leq |\alpha_0| + |\alpha_1| + |\rho_0| \cdot |s(1 - \tau(s))| + |\rho_1| \cdot |s\tau(s)| \end{aligned}$$

[10] Readers familiar with the notion of eigenvalues might have seen the spectral norm defined as the square root of the largest eigenvalue of $\mathbf{\Lambda}$. The two definitions are equivalent. The second definition is the most useful for when it comes to numerical computation.

Letting $L := |\alpha_0| + |\alpha_1|$ and $\lambda := \max\{|\rho_0|, |\rho_1|\}$, we then have

$$|g(s)| \leq \lambda |s| + L$$

If both $|\rho_0|$ and $|\rho_1|$ are strictly less than one, the condition (8.16) is satisfied. If, in addition, τ is continuous, then g is continuous. If the distribution of the shock has, say, a normal distribution, then it has an everywhere positive density on $\mathbb{R}$, and all the conditions of theorem 8.2.1 are satisfied. Hence the process is globally stable.

One interesting special case of theorem 8.2.2 is if we take $h(\mathbf{x}_t) = \mathbb{1}\{\mathbf{x}_t \leq \mathbf{s}\}$ for some fixed $\mathbf{s} \in \mathbb{R}^K$. By (1.8) on page 14, we have

$$\mathbb{E}\left[\mathbb{1}\{\mathbf{x}_t \leq \mathbf{s}\}\right] = \mathbb{P}\{\mathbf{x}_t \leq \mathbf{s}\} = \Pi_\infty(\mathbf{s})$$

so in this case theorem 8.2.2 yields

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{1}\{\mathbf{x}_t \leq \mathbf{s}\} \xrightarrow{p} \Pi_\infty(\mathbf{s})$$

In other words, the ecdf converges to the stationary cdf Π_∞, just as for the IID case.

The ecdf is a nonparametric estimator of the cdf. Provided that the cdf Π_∞ has a density, the same idea works for the standard nonparametric density estimator discussed in §4.4.3. Let's test this for the model in example 8.2.1. For this model, the stationary density is known to be $\pi_\infty(s) = 2\phi(s)\Phi(qs)$, where $q := \rho(1-\rho^2)^{-1/2}$, ϕ is the standard normal density and Φ is the standard normal cdf. This is the blue line in figure 8.5. (The value of ρ is 0.95.) Next, we generate a time series, and plot the nonparametric kernel density estimate $\frac{1}{T\delta}\sum_{t=1}^{T} K(\frac{s-x_t}{\delta})$ as a black line, using R's default choice of K and δ. Here $T = 5000$, and the fit is pretty good. The code for producing figure 8.5 is given in listing 15.

8.2.4 Martingale Difference LLN and CLT

Next we consider asymptotics for martingale difference sequences. Martingale difference sequences are important to us because they are good candidates for the LLN and CLT. To see this, suppose that $\{m_t\}$ is a martingale difference sequence with respect to filtration $\{\mathcal{F}_t\}$. Suppose further that $\{m_t\}$ is identically distributed, and $\mathbb{E}[m_1^2] < \infty$. If the variables $\{m_t\}$ are also independent, then the classical LLN and CLT apply (theorems 1.4.1 and 1.4.2 respectively). Here we do not wish to assume

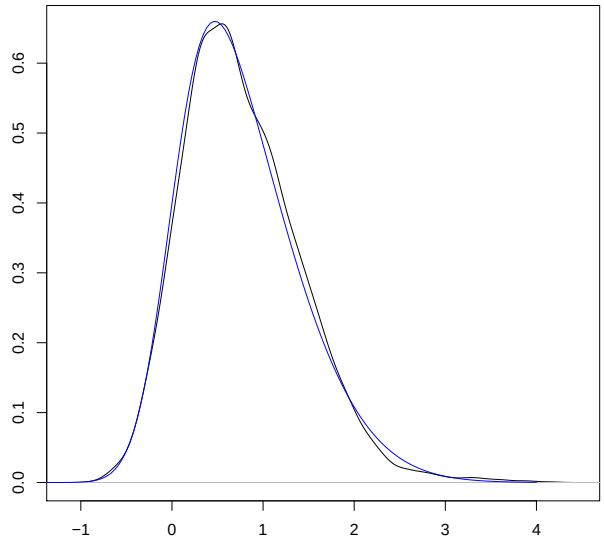


Figure 8.5: Stationary density and nonparametric estimate

Listing 15 Code for figure 8.5

```
rho <- 0.95
delta <- rho / sqrt(1 - rho^2)

T <- 5000
x <- numeric(T)
x[1] = 0
for (t in 2:T) {
    x[t] = rho * abs(x[t-1]) + sqrt(1 - rho^2) * rnorm(1)
}
plot(density(x), xlab="", ylab="", main="")

xgrid <- seq(-4, 4, length=200)
pistar <- function(s) {
    return(2 * dnorm(s) * pnorm(s * delta))
}
lines(xgrid, pistar(xgrid), col="blue")
```

independence, but with martingale difference sequences we do at least have zero correlation. To see this, fix $t \geq 0$ and $j \geq 1$. We have

$$\operatorname{cov}[m_{t+j}, m_t] = \mathbb{E}\left[m_{t+j} m_t\right] = \mathbb{E}\left[\mathbb{E}\left[m_{t+j} m_t \mid \mathcal{F}_{t+j-1}\right]\right]$$

Since $t + j - 1 \geq t$ and $\{\mathcal{F}_t\}$ is a filtration, we know that m_t is $\mathcal{F}_{t+j-1}$-measurable, and hence

$$\mathbb{E}\left[\mathbb{E}\left[m_{t+j} m_t \mid \mathcal{F}_{t+j-1}\right]\right] = \mathbb{E}\left[m_t \mathbb{E}\left[m_{t+j} \mid \mathcal{F}_{t+j-1}\right]\right] = \mathbb{E}\left[m_t \cdot 0\right] = 0$$

This confirms that martingale difference sequences are uncorrelated, as claimed.

Given the lack of correlation, we might hope that the LLN and CLT will still hold in some form. Indeed, the following result is true:

Theorem 8.2.3. *Let $\{m_t\}$ be identically distributed. If $\{m_t\}$ is a martingale difference sequence with respect to some filtration $\{\mathcal{F}_t\}$, then*

$$\frac{1}{T} \sum_{t=1}^{T} m_t \xrightarrow{p} 0 \quad \text{as} \quad T \to \infty \tag{8.18}$$

If, in addition, $\gamma^2 := \mathbb{E}\left[m_t^2\right]$ is positive and finite, and

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[m_t^2 \mid \mathcal{F}_{t-1}\right] \xrightarrow{p} \gamma^2 \quad \text{as} \quad T \to \infty$$

then

$$\sqrt{T}\left[\frac{1}{T} \sum_{t=1}^{T} m_t\right] = T^{-1/2} \sum_{t=1}^{T} m_t \xrightarrow{d} \mathcal{N}(0, \gamma^2) \quad \text{as} \quad T \to \infty \tag{8.19}$$

The LLN result in (8.18) can be proved in exactly the the same way we proved the classical LLN in theorem 1.4.1. Theorem 8.2.3 is a consequence of a martingale CLT proved in Durrett (1996, theorem 7.4).[11] We will use theorem 8.2.3 in our large sample OLS theory below.

8.3 Maximum Likelihood for Markov Processes

[roadmap]

[11] Deriving theorem 8.2.3 from the result in Durrett requires some measure theory, and is beyond the scope of these notes. If you know measure theory then you should be able to work out the proof.

8.3.1 The Likelihood Function

If we have IID scalar observations $x_1, \ldots, x_N$ from common density p_θ, then, as we saw in §4.3.1, the joint density is the product of the marginals, and the maximum likelihood estimate (MLE) of θ is

$$\hat{\theta} := \operatorname{argmax} L(\theta) \quad \text{where} \quad L(\theta) = \prod_{n=1}^{N} p_\theta(x_n) \tag{8.20}$$

If, on the other hand, our data $x_1, \ldots, x_T$ is a time series where the independence assumption does not hold, then the joint density is no longer the product of the marginals. To obtain a convenient expression for the joint density in this general case, let's begin by constructing a joint density for the first three data points x_1, x_2, x_3. Using (1.21) on page 26, we can write the joint density as

$$p(s_1, s_2, s_3) = p(s_3 \mid s_1, s_2) p(s_1, s_2)$$

Applying (1.21) again, this time to $p(s_1, s_2)$, we get

$$p(s_1, s_2, s_3) = p(s_3 \mid s_1, s_2) p(s_2 \mid s_1) p(s_1)$$

Extending this from $T = 3$ to general T we get[12]

$$p(s_1, \ldots, s_T) = p(s_1) \prod_{t=1}^{T-1} p(s_{t+1} \mid s_1, \ldots, s_t)$$

We can specialize further if we are dealing with a Markov process. Suppose that $x_1, \ldots, x_T$ are observations of a globally stable Markov process with transition density $p(s_{t+1} \mid s_t)$ and stationary density π_∞. If the process we are observing has been running for a while, then, given global stability, it is not unreasonable to assume that $x_1 \sim \pi_\infty$. In this case, our expression for the joint density becomes

$$p(s_1, \ldots, s_T) = \pi_\infty(s_1) \prod_{t=1}^{T-1} p(s_{t+1} \mid s_t)$$

where we are using the fact that, for a (first order, time homogeneous) Markov process, $p(s_{t+1} \mid s_1, \ldots, s_t) = p(s_{t+1} \mid s_t)$.[13] Finally, nothing in this expression changes if

[12] Check it by induction if you wish.

[13] This is pretty much the defining property of a first order Markov process, and follows from the general expression $x_{t+1} = G(x_t, w_{t+1})$ given on page 229.

we shift to the vector case, so the joint density of a Markov process $\mathbf{x}_1, \ldots, \mathbf{x}_T$ with transition density $p(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t)$ and $\mathbf{x}_1 \sim \pi_\infty$ has the form

$$p(\mathbf{s}_1, \ldots, \mathbf{s}_T) = \pi_\infty(\mathbf{s}_1) \prod_{t=1}^{T-1} p(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t) \tag{8.21}$$

Turning to the likelihood function, let's suppose now that p depends on an unknown parameter vector $\theta \in \Theta$, and write p_θ. Since the stationary density π_∞ is determined by p_θ (see (8.14) on page 243) we indicate this dependence by writing it as π_∞^θ. The log-likelihood function is then given by

$$\ell(\theta) = \ln \pi_\infty^\theta(\mathbf{x}_1) + \sum_{t=1}^{T-1} \ln p_\theta(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t)$$

In practice it is common to drop the first term in this expression, particularly when the data size is large. There are two reasons. First, if the data size is large, then there are many elements in the sum, and the influence of a single element is likely to be negligible. Second, even though the stationary density π_∞^θ is formally defined by (8.14), for a great many processes there is no known analytical expression for this density.[14] Here we'll follow this convention and, abusing notation slightly, write

$$\ell(\theta) = \sum_{t=1}^{T-1} \ln p_\theta(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t) \tag{8.22}$$

8.3.2 Example: The ARCH Case

Recall the ARCH process

$$x_{t+1} = (a + b x_t^2)^{1/2} w_{t+1}, \qquad \{w_t\} \overset{\text{IID}}{\sim} \mathcal{N}(0,1) \tag{8.23}$$

The transition density for this model is the density of x_{t+1} given $x_t = s$. If $x_t = s$, then $x_{t+1} \sim \mathcal{N}(0, a + bs^2)$, and hence

$$p(s' \,|\, s) = (2\pi(a + bs^2))^{-1/2} \exp\left\{ -\frac{(s')^2}{2(a + bs^2)} \right\}$$

[14] In this situation it is still possible to compute the density numerically, using simulation. The discussion surrounding figure 8.5 gives some idea. A better technique is discussed in Stachurski and Martin (2008).

Since $a + bs^2$ is the conditional variance of x_{t+1}, the parameters a and b are restricted to be nonnegative. Moreover, if $b < 1$, then one can show that the process is globally stable.[15] From (8.22), the log-likelihood function is

$$\ell(a,b) = \sum_{t=1}^{T-1} \left\{ -\frac{1}{2}\ln(2\pi(a+bx_t^2)) - \frac{x_{t+1}^2}{2(a+bx_t^2)} \right\} \tag{8.24}$$

Rearranging, dropping terms that do not depend on a or b, and multiplying by 2 (an increasing transformation), we can rewrite this (abusing notation again) as

$$\ell(a,b) = -\sum_{t=1}^{T-1} \left\{ \ln z_t + \frac{x_{t+1}^2}{z_t} \right\} \quad \text{where} \quad z_t := a + bx_t^2 \tag{8.25}$$

Let's run some simulations to see what this function looks like. In the simulations we will set $T = 500$ and $a = b = 0.5$. Thus, we imagine the situation where, unbeknownst to us, the true parameter values are $a = b = 0.5$, and we observe a time series $x_1, \ldots, x_{500}$ generated by these parameters. In order to estimate a and b, we form the likelihood function (8.25), and obtain the MLEs $\hat{a}$ and $\hat{b}$ as the vector $(\hat{a}, \hat{b})$ that maximizes $\ell(a,b)$.

Four different simulations of ℓ are given in figure 8.6. In each figure, a separate data set $x_1, \ldots, x_{500}$ is generated using the true parameter values $a = b = 0.5$, and the function ℓ in (8.25) is then plotted. Since the graph of the function is three dimensional (i.e, the function has two arguments), we have plotted it using contour lines and a color map. Lighter colors refer to larger values. The horizontal axis is a values, and the vertical axis is b values. The code for producing one of these figures (modulo randomness) is given in listing 16. The function `arch_like(theta, data)` represents ℓ in (8.25), with `theta` corresponding to (a,b) and `data` corresponding to the time series $x_1, \ldots, x_T$.

In each of the four simulations, a rough guess of the MLEs can be obtained just by looking for maximizers in the figures. For example, in simulation (a), the MLEs look to be around $\hat{a} = 0.44$ and $\hat{b} = 0.61$. To get more accurate estimates, we can use some form of analytical or numerical optimization. For this problem, we don't have any analytical expressions for the MLEs because setting the two partial derivatives of ℓ in (8.25) to zero does not yield neat expressions for $\hat{a}$ and $\hat{b}$. On the other hand, there are many numerical routines we can use to obtain the MLEs for a given data set.

[15] Unfortunately, theorem 8.2.1 (page 244) cannot be used to check global stability, because the shock is not additive. If you wish to verify global stability then have a look at the techniques in Chapter 8 of Stachurski (2009).

The simplest approach is to use one of R's inbuilt optimization routines. For example, given the definition of `arch_like` in listing 16 and a sequence of observations $x_1, \ldots, x_T$ stored in a vector `xdata`, the function `arch_like` can be optimized numerically via the commands

```
start_theta <- c(0.65, 0.35)  # An initial guess of (a,b)
neg_like <- function(theta) {
    return(-arch_like(theta, xdata))  # xdata is the data
}
opt <- optim(start_theta, neg_like, method="BFGS")
```

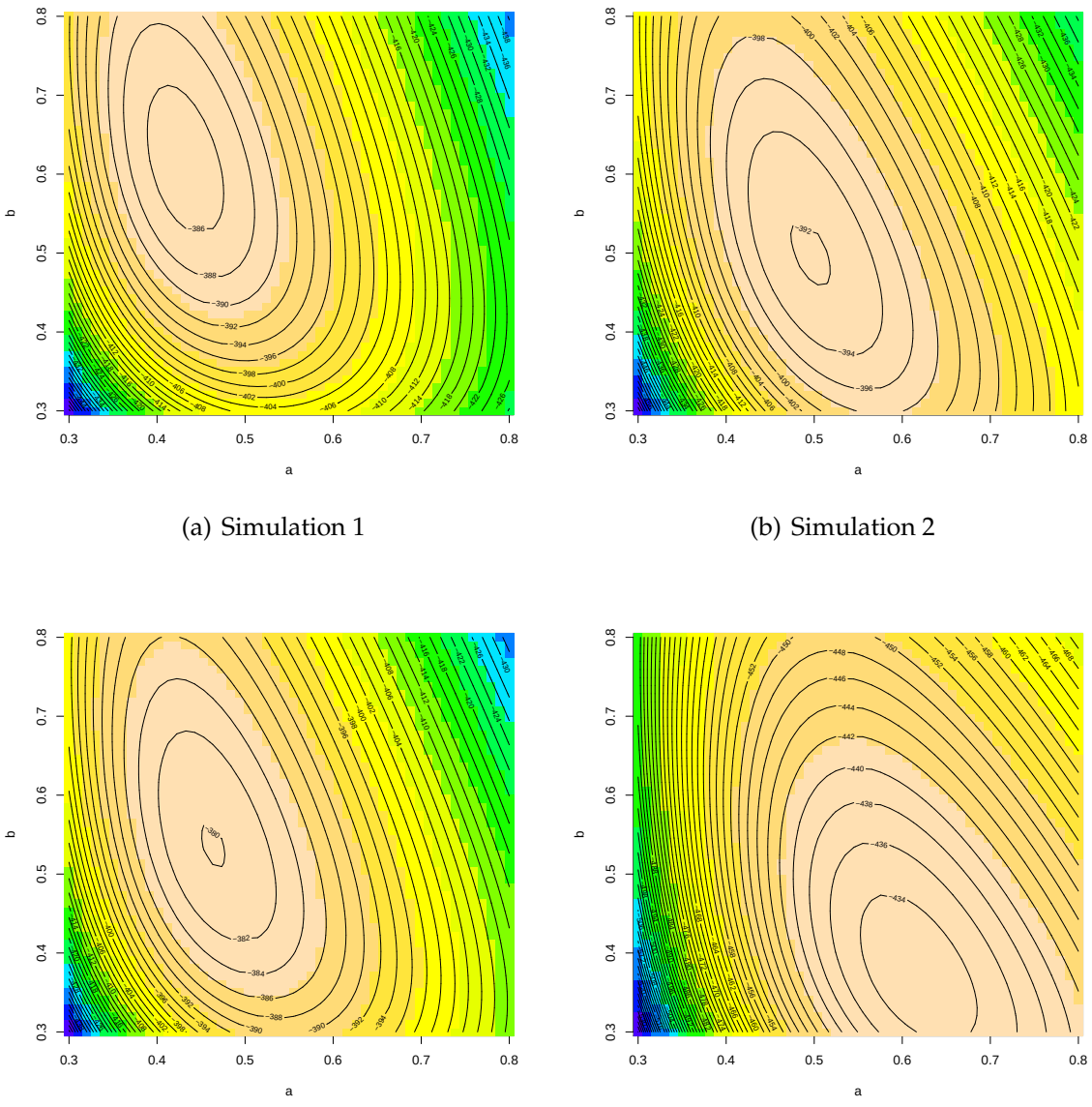
Here `optim` is an built-in R function for numerical optimization of multivariate functions. Most built-in functions in most languages perform minimization rather than maximization, and `optim` is no exception. For this reason, the function that we pass to `optim` is `neg_like`, which is -1 times ℓ. The first argument to `optim` is a vector of starting values (a guess of the MLEs). The last argument tells `optim` to use the BFGS routine, which is variation on the Newton-Raphson algorithm. The return value of `optim` is a list, and the approximate minimizing vector is one element of this list (called `par`).

In this particular set up, for most realizations of the data and starting values, you will find that the algorithm converges to a good approximation to the global optimizer. However, there's no guarantee that it will. In case of problems, it's useful to know how these kinds of algorithms work, and how to code up simple implementations on your own. The next section will get you started.

8.3.3 The Newton-Raphson Algorithm

The Newton-Raphson algorithm is a *root-finding* algorithm. In other words, given a function $g\colon \mathbb{R} \to \mathbb{R}$, the algorithm searches for points $\bar{s} \in \mathbb{R}$ such that $g(\bar{s}) = 0$. Any root-finding algorithm can be used to optimize differentiable functions because, for differentiable functions, interior optimizers are always roots of the objective function's first derivative.

To describe the algorithm, let's begin with the root-finding problem and then specialize to optimization. To begin, let $g\colon \mathbb{R} \to \mathbb{R}$, and let s_0 be some initial point in $\mathbb{R}$ that we think (hope) is somewhere near a root. We don't know how to jump from s_0 straight to a root of g (otherwise there would be no problem to solve), but what we can do is move to the root of the function which forms the *tangent line* to g at s_0. In



(a) Simulation 1

(b) Simulation 2

(c) Simulation 3

(d) Simulation 4

Figure 8.6: Simulations of the function (8.25) with $T = 500$

Listing 16 Code for figure 8.6

```
arch_like <- function(theta, data) {
    Y <- data[-1]                #  All but first element
    X <- data[-length(data)]     #  All but last element
    Z <- theta[1] + theta[2] * X^2
    return(-sum(log(Z) + Y^2 / Z))
}

sim_data <- function(a, b, n=500) {
    x <- numeric(n)
    x[1] = 0
    w = rnorm(n)
    for (t in 1:(n-1)) {
        x[t+1] = sqrt(a + b * x[t]^2) * w[t]
    }
    return(x)
}

xdata <- sim_data(0.5, 0.5)   #  True parameters

K <- 50
a <- seq(0.3, 0.8, length=K)
b <- seq(0.3, 0.8, length=K)
M <- matrix(nrow=K, ncol=K)
for (i in 1:K) {
    for (j in 1:K) {
        theta <- c(a[i], b[j])
        M[i,j] <- arch_like(theta, xdata)
    }
}
image(a, b, M, col=topo.colors(12))
contour(a, b, M, nlevels=40, add=T)
```

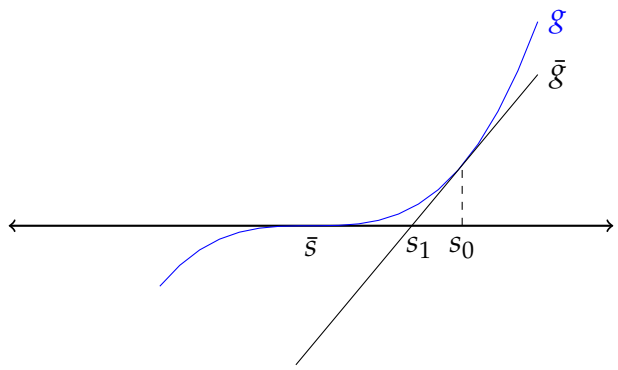


Figure 8.7: The first step of the NR algorithm

other words, we replace g with its linear approximation around s_0, which is given by

$$\tilde{g}(s) := g(s_0) + g'(s_0)(s - s_0) \qquad (s \in \mathbb{R})$$

and solve for the root of $\tilde{g}$. This point is represented as s_1 in figure 8.7, and the value is easily seen to be $s_1 := s_0 - g(s_0)/g'(s_0)$. The point s_1 is taken as our next guess of the root, and the procedure is repeated, taking the tangent of g at s_1, solving for the root, and so on. This generates a sequence of points $\{s_k\}$ satisfying

$$s_{k+1} = s_k - \frac{g(s_k)}{g'(s_k)}$$

There are various results telling us that when g is suitably well-behaved and s_0 is sufficiently close to a given root $\bar{s}$, then sequence $\{s_k\}$ will converge to $\bar{s}$.[16]

To move from general root-finding to the specific problem of optimization, suppose now that $g\colon \mathbb{R} \to \mathbb{R}$ is a differentiable function we wish to maximize. We know that if s^* is a maximizer of g, then $g'(s^*) = 0$. Hence it is natural to begin our search for maximizers by looking for roots to this equation. This can be done by applying the Newton-Raphson algorithm to g', which yields the sequence

$$s_{k+1} = s_k - \frac{g'(s_k)}{g''(s_k)} \tag{8.26}$$

We can extend this algorithm to the multivariate case as well. Let's suppose that g is a function of two arguments. In particular, suppose that g is twice differentiable

[16] In practical situations we often have no way of knowing whether the conditions are satisfied, and there have been many attempts to make the procedure more robust. The R function `optim` described above is a child of this process.

and $g\colon \mathbb{R}^2 \to \mathbb{R}$. The **gradient vector** and **Hessian** of g at $(x, y) \in \mathbb{R}^2$ are defined as

$$\nabla g(x,y) := \begin{pmatrix} g_1'(x,y) \\ g_2'(x,y) \end{pmatrix} \quad \text{and} \quad \nabla^2 g(x,y) := \begin{pmatrix} g_{11}''(x,y) & g_{12}''(x,y) \\ g_{21}''(x,y) & g_{22}''(x,y) \end{pmatrix}$$

Here g_i' is the first partial of g with respect to its i-th argument, g_{ij}'' second derivative cross-partial, and so on.

By analogy with (8.26), the Newton-Raphson algorithm for this two dimensional case is the algorithm that generates the sequence $\{(x_k, y_k)\}$ defined by

$$(x_{k+1}, y_{k+1}) = (x_k, y_k) - [\nabla^2 g(x_k, y_k)]^{-1} \nabla g(x_k, y_k) \tag{8.27}$$

from some initial guess (x_0, y_0).[17]

For the sake of the exercise, let's apply this to maximization of (8.25). Let z_t be as defined in (8.25). The first partials are then

$$\frac{\partial \ell}{\partial a}(a,b) := \sum_{t=1}^{T-1} \left[\frac{x_{t+1}^2}{z_t^2} - \frac{1}{z_t} \right], \quad \frac{\partial \ell}{\partial b}(a,b) := \sum_{t=1}^{T-1} x_t^2 \left[\frac{x_{t+1}^2}{z_t^2} - \frac{1}{z_t} \right]$$

while the second partials are

$$\frac{\partial^2 \ell}{\partial a^2}(a,b) := \sum_{t=1}^{T-1} \left[\frac{1}{z_t^2} - 2\frac{x_{t+1}^2}{z_t^3} \right], \quad \frac{\partial^2 \ell}{\partial b^2}(a,b) := \sum_{t=1}^{T-1} x_t^4 \left[\frac{1}{z_t^2} - 2\frac{x_{t+1}^2}{z_t^3} \right]$$

The cross-partial is

$$\frac{\partial^2 \ell}{\partial a \partial b}(a,b) := \sum_{t=1}^{T-1} x_t^2 \left[\frac{1}{z_t^2} - 2\frac{x_{t+1}^2}{z_t^3} \right]$$

From these expressions we can easily form the gradient vector and the Hessian, pick an initial guess, and iterate according to (8.27). Figure 8.8 show four iterations of this procedure, starting from $(a_0, b_0) = (0.65, 0.35)$.[18] In this case the convergence is quick, and we are already close to the global optimum.

Replication of this figure (modulo randomness) is left as an exercise.

8.4 Models with Latent Variables

To be written. Max likelihood with latent variables. GARCH, HMM, Markov switching, factor models?

[17] We are assuming that the Hessian matrix is invertible.

[18] As before, the simulation uses $a = b = 0.5$ and $T = 500$.

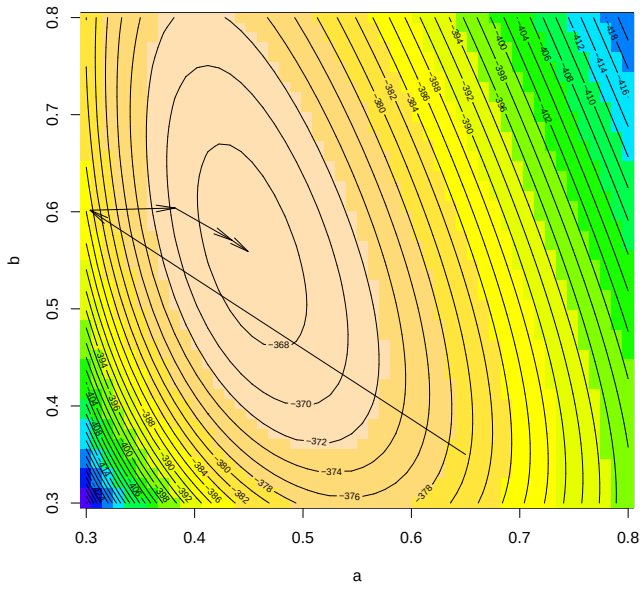


Figure 8.8: Newton-Raphson iterates

8.5 Exercises

Ex. 8.5.1. Using fact 1.4.1 (page 31) as appropriate, prove the following part of fact 2.5.2: If $\mathbf{X}_n \xrightarrow{p} \mathbf{X}$ and $\mathbf{Y}_n \xrightarrow{p} \mathbf{Y}$, then $\mathbf{X}_n\mathbf{Y}_n \xrightarrow{p} \mathbf{XY}$ whenever the matrices are conformable.

Ex. 8.5.2. Confirm the following claim in fact 2.5.3: If $\mathbf{a}'\mathbf{x}_n \xrightarrow{p} \mathbf{a}'\mathbf{x}$ for every $\mathbf{a} \in \mathbb{R}^K$, then $\mathbf{x}_n \xrightarrow{p} \mathbf{x}$.

Ex. 8.5.3. Let $\{\mathbf{x}_n\}$ be a sequence of vectors in $\mathbb{R}^2$, where $\mathbf{x}_n := (x_n, y_n)$ for each n. Suppose that $\mathbf{x}_n \xrightarrow{p} \mathbf{0}$ (i.e., $x_n \xrightarrow{p} 0$ and $y_n \xrightarrow{p} 0$). Show that $\|\mathbf{x}_n\| \xrightarrow{p} 0$.

Ex. 8.5.4. Verify fact 2.5.1 on page 75. (Note that exercise 8.5.3 is a warm up to this exercise.)

Ex. 8.5.5. Confirm the claim $\sqrt{N}(\bar{\mathbf{x}}_N - \boldsymbol{\mu}) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \Sigma)$ in theorem 2.5.1.

Ex. 8.5.6. Let $\{\mathbf{x}_n\}$ be an IID sequence of random vectors in $\mathbb{R}^K$ with $\mathbb{E}[\mathbf{x}_n] = \mathbf{0}$ and $\text{var}[\mathbf{x}_n] = \mathbf{I}_K$. Let

$$\bar{\mathbf{x}}_N := \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n \quad \text{and} \quad y_N := N \cdot \|\bar{\mathbf{x}}_N\|^2$$

What is the asymptotic distribution of $\{y_N\}$?

Ex. 8.5.7. Suppose that $x_{t+1} = \alpha + \rho x_t + w_{t+1}$, where

$$x_t \sim \pi_\infty := \mathcal{N}\left(\frac{\alpha}{1-\rho}, \frac{1}{1-\rho^2}\right) \quad \text{and} \quad w_{t+1} \sim \mathcal{N}(0,1)$$

Show that $x_{t+1} \sim \pi_\infty$ also holds.

Ex. 8.5.8. Let $\{\mathcal{F}_t\}$ be a filtration. Show that if $\{m_t\}$ is a martingale with respect to $\{\mathcal{F}_t\}$, then $d_t = m_t - m_{t-1}$ is a martingale difference sequence with respect to $\{\mathcal{F}_t\}$.

Ex. 8.5.9. Let $\{\mathcal{F}_t\}$ be a filtration, and let $\{m_t\}$ be a martingale with respect to $\{\mathcal{F}_t\}$. Let $\eta_t := m_t + 1$, let $\kappa_t := 2m_t$, and let $\gamma_t := m_t^2$.

1. Is $\{\eta_t\}$ a martingale with respect to $\{\mathcal{F}_t\}$?
2. Is $\{\kappa_t\}$ a martingale with respect to $\{\mathcal{F}_t\}$?
3. Is $\{\gamma_t\}$ a martingale with respect to $\{\mathcal{F}_t\}$?

If yes, give a proof. If not, give a counterexample.[19]

Ex. 8.5.10. Let $\{\eta_t\}$ be an IID sequence of scalar random variables with $\mathbb{E}[\eta_1] = 0$ and $\text{var}[\eta_1] = \sigma^2 > 0$. Let $\{\mathcal{F}_t\}$ be the filtration defined by $\mathcal{F}_t := \{\eta_1, \ldots, \eta_t\}$, and let $z_t := t \cdot \eta_t$ for each t.

1. Is $\{z_t\}$ IID? Why or why not?
2. Is $\{z_t\}$ a martingale difference sequence with respect to $\{\mathcal{F}_t\}$? Why or why not?

Ex. 8.5.11. Let $\{\eta_t\}$ be an IID sequence of scalar random variables with

$$\mathbb{P}\{\eta_1 = 1\} = \mathbb{P}\{\eta_1 = -1\} = 0.5$$

and let $\{\mathcal{F}_t\}$ be the filtration defined by $\mathcal{F}_t := \{\eta_1, \ldots, \eta_t\}$. Let

$$m_t := \sum_{j=1}^{t} \eta_j \quad \text{and} \quad \kappa_t := m_t^2 - t$$

Show that $\{\kappa_t\}$ is a martingale with respect to $\{\mathcal{F}_t\}$.

[19] To give a counterexample, you need to give a specific example of the pair $\{m_t\}$ and $\{\mathcal{F}_t\}$ where the stated property fails. Look in the course notes for specific examples of martingales.

Ex. 8.5.12. Consider the scalar sequence $x_{t+1} = \rho x_t + w_{t+1}$, where $\{w_t\} \overset{\text{IID}}{\sim} \mathcal{N}(0,1)$ and $x_0 = 0$. Let $\mathcal{F}_t := \{w_1, \ldots, w_t\}$. Give conditions on ρ such that

1. $\{x_t\}$ is a martingale with respect to $\{\mathcal{F}_t\}$.
2. $\{x_t\}$ is a martingale difference sequence with respect to $\{\mathcal{F}_t\}$.

Ex. 8.5.13. Consider again the scalar Markov sequence $x_{t+1} = \rho x_t + w_{t+1}$. Assume that $\{w_t\}$ is IID, having Student's t-distribution with 2 degrees of freedom, and that $-1 < \rho < 1$. Prove that this process has a unique, globally stable stationary distribution using theorem 8.2.1.

8.5.1 Solutions to Selected Exercises

Solution to Exercise 8.5.1. Let $\mathbf{X}_n \xrightarrow{p} \mathbf{X}$ and $\mathbf{Y}_n \xrightarrow{p} \mathbf{Y}$. To prove that $\mathbf{X}_n \mathbf{Y}_n \xrightarrow{p} \mathbf{XY}$, we need to show that the i,j-th element of $\mathbf{X}_n \mathbf{Y}_n$ converges in probability to the i,j-th element of $\mathbf{XY}$. By hypothesis, we have

$$x_{ik}^n \xrightarrow{p} x_{ik} \quad \text{and} \quad y_{kj}^n \xrightarrow{p} y_{kj} \quad \text{for all} \quad k$$

Applying fact 1.4.1 on page 31 twice, we obtain

$$x_{ik}^n y_{kj}^n \xrightarrow{p} x_{ik} y_{kj} \quad \text{for all} \quad k$$

and then

$$\sum_k x_{ik}^n y_{kj}^n \xrightarrow{p} \sum_k x_{ik} y_{kj}$$

In other words, the i,j-th element of $\mathbf{X}_n \mathbf{Y}_n$ converges in probability to the i,j-th element of $\mathbf{XY}$. □

Solution to Exercise 8.5.2. If $\mathbf{a}' \mathbf{x}_n \xrightarrow{p} \mathbf{a}' \mathbf{x}$ for every $\mathbf{a} \in \mathbb{R}^K$, then we know in particular that this convergence holds for the canonical basis vectors. Hence

$$\mathbf{e}_k' \mathbf{x}_n \xrightarrow{p} \mathbf{e}_k' \mathbf{x} \quad \text{for every } k$$

$$\therefore \quad x_n^k \xrightarrow{p} x^k \quad \text{for every } k \quad \text{(elementwise convergence)}$$

$$\therefore \quad \mathbf{x}_n \xrightarrow{p} \mathbf{x} \quad \text{(vector convergence, by definition)}$$

□

Solution to Exercise 8.5.3. From fact 1.4.1 on page 31, we know that if $g\colon \mathbb{R} \to \mathbb{R}$ is continuous and $\{u_n\}$ is a scalar sequence of random variables with $u_n \xrightarrow{p} u$, then $g(u_n) \xrightarrow{p} g(u)$. We also know that if $u_n \xrightarrow{p} u$ and $v_n \xrightarrow{p} v$, then $u_n + v_n \xrightarrow{p} u + v$. By assumption, we have

$$
\begin{aligned}
& x_n \xrightarrow{p} 0 \quad \text{and} \quad y_n \xrightarrow{p} 0 \\
\therefore \quad & x_n^2 \xrightarrow{p} 0^2 = 0 \quad \text{and} \quad y_n^2 \xrightarrow{p} 0^2 = 0 \\
\therefore \quad & \|\mathbf{x}_n\|^2 = x_n^2 + y_n^2 \xrightarrow{p} 0 + 0 = 0 \\
\therefore \quad & \|\mathbf{x}_n\| = \sqrt{\|\mathbf{x}_n\|^2} \xrightarrow{p} \sqrt{0} = 0
\end{aligned}
$$

□

Solution to Exercise 8.5.4. Let $\{\mathbf{x}_n\}$ be a sequence of random vectors in $\mathbb{R}^K$ and $\mathbf{x}$ be a random vector in $\mathbb{R}^K$. We need to show that

$$
x_k^n \xrightarrow{p} x_k \text{ for all } k \iff \|\mathbf{x}_n - \mathbf{x}\| \xrightarrow{p} 0
$$

A special case of this argument can be found in the solution to exercise 8.5.3. The general case is similar: Suppose first that $x_k^n \xrightarrow{p} x_k$ for all k. Combining the various results about scalar convergence in probability in fact 1.4.1 (page 31), one can then verify (details left to you) that

$$
\|\mathbf{x}_n - \mathbf{x}\| := \sqrt{\sum_{k=1}^K (x_k^n - x_k)^2} \xrightarrow{p} 0 \qquad (n \to \infty)
$$

Regarding the converse, suppose now that $\|\mathbf{x}_n - \mathbf{x}\| \xrightarrow{p} 0$. Fix $\epsilon > 0$ and arbitrary k. From the definition of the norm we see that $|x_k^n - x_k| \leq \|\mathbf{x}_n - \mathbf{x}\|$ is always true, and hence

$$
\begin{aligned}
& |x_k^n - x_k| > \epsilon \implies \|\mathbf{x}_n - \mathbf{x}\| > \epsilon \\
\therefore \quad & \{|x_k^n - x_k| > \epsilon\} \subset \{\|\mathbf{x}_n - \mathbf{x}\| > \epsilon\} \\
\therefore \quad & 0 \leq \mathbb{P}\{|x_k^n - x_k| > \epsilon\} \leq \mathbb{P}\{\|\mathbf{x}_n - \mathbf{x}\| > \epsilon\} \to 0
\end{aligned}
$$

The proof is done. □

Solution to Exercise 8.5.5. Define

$$
\mathbf{z}_n := \sqrt{N}\,(\bar{\mathbf{x}}_N - \boldsymbol{\mu}) \quad \text{and} \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \Sigma)
$$

We need to show that $\mathbf{z}_n \xrightarrow{d} \mathbf{z}$. To do this, we apply the Cramer-Wold device (fact 2.5.3, page 77) and the scalar CLT (theorem 1.4.2, page 36). To begin, fix $\mathbf{a} \in \mathbb{R}^K$. Observe that

$$\mathbf{a}'\mathbf{z}_n := \sqrt{N}\left(\bar{y}_n - \mathbb{E}\left[y_n\right]\right)$$

where $y_n := \mathbf{a}'\mathbf{x}_n$. Since y_n is IID (in particular, functions of independent random variables are independent) and

$$\operatorname{var}[y_n] = \operatorname{var}[\mathbf{a}'\mathbf{x}_n] = \mathbf{a}' \operatorname{var}[\mathbf{x}_n]\mathbf{a} = \mathbf{a}'\Sigma\mathbf{a}$$

the scalar CLT yields

$$\mathbf{a}'\mathbf{z}_n \xrightarrow{d} \mathcal{N}(0, \mathbf{a}'\Sigma\mathbf{a})$$

Since $\mathbf{a}'\mathbf{z} \sim \mathcal{N}(0, \mathbf{a}'\Sigma\mathbf{a})$, we have shown that $\mathbf{a}'\mathbf{z}_n \xrightarrow{d} \mathbf{a}'\mathbf{z}$. Since $\mathbf{a}$ was arbitrary, the Cramer-Wold device tells us that $\mathbf{z}_n$ converges in distribution to $\mathbf{z}$. □

Solution to Exercise 8.5.6. By assumption, $\{\mathbf{x}_n\}$ is an IID sequence in $\mathbb{R}^K$ with $\mathbb{E}\left[\mathbf{x}_n\right] = \mathbf{0}$ and $\operatorname{var}[\mathbf{x}_n] = \mathbf{I}_K$. It follows from the vector central limit theorem that

$$\sqrt{N}\bar{\mathbf{x}}_N \xrightarrow{d} \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_K)$$

Letting $g(\mathbf{s}) := \|\mathbf{s}\|^2$ and applying the continuous mapping theorem (fact 2.5.4 on page 77), we obtain

$$y_N = \|\sqrt{N}\bar{\mathbf{x}}_N\|^2 \xrightarrow{d} \|\mathbf{z}\|^2 = \sum_{k=1}^{K} z_k^2$$

From fact 1.3.4 on page 27 we conclude that $y_N \xrightarrow{d} \chi^2(K)$. □

Solution to Exercise 8.5.7. To simplify the algebra, I'll solve the case where $\alpha = 0$, and leave further details to you. Since linear combinations of normals are normal, we know that x_{t+1} is normal. Thus, it remains only to show that $\mathbb{E}\left[x_{t+1}\right] = 0$ and $\operatorname{var}[x_{t+1}] = 1/(1-\rho^2)$. The first claim is true because, by linearity of expectations,

$$\mathbb{E}\left[x_{t+1}\right] = \mathbb{E}\left[\rho x_t + w_{t+1}\right] = \rho\mathbb{E}\left[x_t\right] + \mathbb{E}\left[u\right] = 0$$

The second claim also holds, because, from the rule for variance of linear combinations,

$$\operatorname{var}[x_{t+1}] = \operatorname{var}[\rho x_t + w_{t+1}] = \rho^2 \operatorname{var}[x_t] + \operatorname{var}[w_{t+1}] + 2\rho \operatorname{cov}[x_t, w_{t+1}]$$

Since x_t and w_{t+1} are independent (x_t is a function of current and lagged shocks only), the final term is zero, and we get

$$\text{var}[x_{t+1}] = \rho^2 \frac{1}{1-\rho^2} + 1 = \frac{1}{1-\rho^2}$$

□

Solution to Exercise 8.5.8. We need to show that

1. $\{d_t\}$ is adapted to $\{\mathcal{F}_t\}$, and
2. $\mathbb{E}[d_{t+1} \mid \mathcal{F}_t] = 0$

To prove that $\{d_t\}$ is adapted to $\{\mathcal{F}_t\}$, we need to show that d_t is a function of variables in $\mathcal{F}_t$. To see this, observe first that m_t is a function of variables in $\mathcal{F}_t$. Second, m_{t-1} is a function of variables in $\mathcal{F}_{t-1}$, and, by the definition of filtrations, every variable in $\mathcal{F}_{t-1}$ is also in $\mathcal{F}_t$. Hence, m_{t-1} is also is a function of variables in $\mathcal{F}_t$. Since both m_t and m_{t-1} are functions of variables in $\mathcal{F}_t$, clearly $d_t = m_t - m_{t-1}$ is also a function of variables in $\mathcal{F}_t$.[20]

Finally, since $\mathbb{E}[m_{t+1} \mid \mathcal{F}_t] = m_t$ (recall that $\{m_t\}$ is a martingale) and $\mathbb{E}[m_t \mid \mathcal{F}_t] = m_t$ (by fact 8.1.2), we have

$$\mathbb{E}[d_{t+1} \mid \mathcal{F}_t] = \mathbb{E}[m_{t+1} - m_t \mid \mathcal{F}_t] = \mathbb{E}[m_{t+1} \mid \mathcal{F}_t] - \mathbb{E}[m_t \mid \mathcal{F}_t] = m_t - m_t = 0$$

□

Solution to Exercise 8.5.9. Regarding part 1, $\{\eta_t\}$ a martingale with respect to $\{\mathcal{F}_t\}$. Firstly, $\{\eta_t\}$ adapted to $\{\mathcal{F}_t\}$, because $\{m_t\}$ adapted to $\{\mathcal{F}_t\}$ by assumption, so m_t is a function of variables in $\mathcal{F}_t$. Hence $\eta_t = m_t + 1$ is a function of variables in $\mathcal{F}_t$.[21] Moreover, using the fact that $\{m_t\}$ is a martingale with respect to $\{\mathcal{F}_t\}$, we have

$$\mathbb{E}[\eta_{t+1} \mid \mathcal{F}_t] = \mathbb{E}[m_{t+1} + 1 \mid \mathcal{F}_t] = \mathbb{E}[m_{t+1} \mid \mathcal{F}_t] + 1 = m_t + 1 = \eta_t$$

Regarding part 2, $\{\kappa_t\}$ a martingale with respect to $\{\mathcal{F}_t\}$. The proof is similar to part 1, and hence omitted.

[20] The way to think about this intuitively is to think about whether or not d_t can be computed on the basis of information available at time t. Since both m_t and m_{t-1} can be computed at time t, their difference $d_t = m_t - m_{t-1}$ can also be computed.

[21] Once again, the way to remember this is to recognize that since the value of m_t can be computed at time t (by assumption), the value of $\eta_t = m_t + 1$ can also be computed.

Regarding part 3, $\{\gamma_t\}$ is not generally a martingale with respect to $\{\mathcal{F}_t\}$. For example, we saw in the course notes that if $\{\xi_t\}$ is an IID sequence of random variables with $\mathbb{E}[\xi_1] = 0$, $m_t := \sum_{j=1}^t \xi_j$ and $\mathcal{F}_t := \{\xi_1, \ldots, \xi_t\}$, then $\{m_t\}$ is a martingale with respect to $\{\mathcal{F}_t\}$. However, the process $\{\gamma_t\}$ given by $\gamma_t = m_t^2$ is not a martingale whenever $\sigma^2 := \mathbb{E}[\xi_1^2]$ is strictly positive. To see this, observe that

$$\gamma_{t+1} = m_{t+1}^2 = \left(\sum_{j=1}^t \xi_j + \xi_{t+1} \right)^2 = (m_t + \xi_{t+1})^2 = m_t^2 + m_t \xi_{t+1} + \xi_{t+1}^2$$

$$\therefore \quad \mathbb{E}[\gamma_{t+1} \mid \mathcal{F}_t] = \mathbb{E}[m_t^2 \mid \mathcal{F}_t] + \mathbb{E}[m_t \xi_{t+1} \mid \mathcal{F}_t] + \mathbb{E}[\xi_{t+1}^2 \mid \mathcal{F}_t]$$

Since

$$\mathbb{E}[m_t \xi_{t+1} \mid \mathcal{F}_t] = m_t \mathbb{E}[\xi_{t+1} \mid \mathcal{F}_t] = m_t \mathbb{E}[\xi_{t+1}] = 0$$

and

$$\mathbb{E}[\xi_{t+1}^2 \mid \mathcal{F}_t] = \mathbb{E}[\xi_{t+1}^2] = \sigma^2$$

we now have

$$\mathbb{E}[\gamma_{t+1} \mid \mathcal{F}_t] = \mathbb{E}[m_t^2 \mid \mathcal{F}_t] + \sigma^2 = m_t^2 + \sigma^2 = \gamma_t + \sigma^2 > \gamma_t$$

Hence $\{\gamma_t\}$ is not a martingale with respect to $\{\mathcal{F}_t\}$. □

Solution to Exercise 8.5.12. It is clear from (8.6) on page 229 that $\{x_t\}$ is adapted to the filtration for all values of ρ.

Regarding part 1, if $\rho = 1$, then $\{x_t\}$ is the random walk in example 8.1.4. Hence $\{x_t\}$ is a martingale with respect to $\{\mathcal{F}_t\}$. Regarding part 2, if $\rho = 0$, then $x_t = w_t$, and

$$\mathbb{E}[x_{t+1} \mid \mathcal{F}_t] = \mathbb{E}[w_{t+1} \mid \mathcal{F}_t] = \mathbb{E}[w_{t+1}] = 0$$

Hence $\{x_t\}$ is a martingale difference sequence with respect to $\{\mathcal{F}_t\}$. □

Chapter 9

Large Sample OLS

[roadmap]

9.1 Consistency

Let's return to the linear multivariate regression problem studied in chapters 6 and 7. Although the large sample theory we develop here has applications in a cross-sectional environment with no correlation between observations, for additional generality we will imagine ourselves to be in a time series setting. To remind us of this, observations will be indexed by t rather than n, and the sample size will be denoted by T rather than N.

The only assumption we will retain from chapter 7 is the linear model assumption (see 7.1.1). In particular, we assume that our data $(y_1, \mathbf{x}_1), \ldots, (y_T, \mathbf{x}_T)$ is generated by the linear model

$$y_t = \mathbf{x}_t' \boldsymbol{\beta} + u_t \tag{9.1}$$

where $\boldsymbol{\beta}$ is a K-vector of unknown coefficients, and u_t is an unobservable shock. We let $\mathbf{y}$ be the T-vector of observed outputs, so that y_t is the t-th element of the $T \times 1$ vector $\mathbf{y}$, and $\mathbf{u}$ be the vector of shocks, so that u_t is the t-th element of the $T \times 1$ vector $\mathbf{u}$. We let $\mathbf{X}$ be the $T \times K$ matrix

$$\mathbf{X} := \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1K} \\ x_{21} & x_{22} & \cdots & x_{2K} \\ \vdots & \vdots & & \vdots \\ x_{T1} & x_{T2} & \cdots & x_{TK} \end{pmatrix}$$

We continue to assume that $\mathbf{X}$ is full column rank (i.e., $\text{rank}(\mathbf{X}) = K$).

We will estimate the parameter vector $\boldsymbol{\beta}$ via least squares. The expression for the OLS estimate is unchanged:

$$\hat{\boldsymbol{\beta}}_T := (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

Notice also that we are subscripting $\hat{\boldsymbol{\beta}}$ with T to make the dependence of the estimator on T explicit. We can make this dependence clearer by rewriting the estimator in a different way. Multiplying and dividing by T, we get

$$\hat{\boldsymbol{\beta}}_T = \left[\frac{1}{T}\mathbf{X}'\mathbf{X}\right]^{-1} \cdot \frac{1}{T}\mathbf{X}'\mathbf{y}$$

Expanding out the matrix products (exercise 9.3.1), we obtain

$$\hat{\boldsymbol{\beta}}_T = \left[\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t\mathbf{x}_t'\right]^{-1} \cdot \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t y_t \tag{9.2}$$

Also, taking our usual expression $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}$ for the sampling error and performing a similar manipulation, we get

$$\hat{\boldsymbol{\beta}}_T - \boldsymbol{\beta} = \left[\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t\mathbf{x}_t'\right]^{-1} \cdot \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t u_t \tag{9.3}$$

9.1.1 Assumptions

Let's now study the properties of this estimator in the time series setting. In this setting, we abandon assumption 7.1.1, which is the exogeneity assumption $\mathbb{E}[\mathbf{u} \mid \mathbf{X}] = \mathbf{0}$. The reason is that this assumption excludes too many models. For example, we showed in §7.5.1 that the assumption fails when we try to estimate the simple AR(1) model $y_{t+1} = \beta y_t + u_{t+1}$ by setting $x_t = y_{t-1}$, thereby producing the regression model

$$y_t = \beta x_t + u_t, \qquad t = 1, \ldots, T \tag{9.4}$$

The problem is that for this specification of (9.1), the regressor is correlated with lagged values of the shock.

We know that under assumption 7.1.1, the OLS estimator is unbiased for $\boldsymbol{\beta}$ (theorem 7.2.1). In fact assumption 7.1.1 is close to the minimum requirement for unbiasedness, and without it there is little chance of establishing this property. Instead we will aim for a large sample property: consistency of $\hat{\boldsymbol{\beta}}$. To this end, we make the following assumptions:

Assumption 9.1.1 (Ergodic regressors)**.** The sequence $\mathbf{x}_1, \ldots, \mathbf{x}_T$ is identically distributed, and $\mathbf{\Sigma_{xx}} := \mathbb{E}[\mathbf{x}_1 \mathbf{x}_1']$ is positive definite. Moreover,

$$\frac{1}{T} \sum_{t=1}^{T} \mathbf{x}_t \mathbf{x}_t' \xrightarrow{p} \mathbf{\Sigma_{xx}} \quad \text{as} \quad T \to \infty \tag{9.5}$$

Two remarks: First, although the observations $\mathbf{x}_1, \ldots, \mathbf{x}_T$ in assumption 9.1.1 are required to be identically distributed, they are not assumed to be IID—some correlation is allowed. Second, we are implicity assuming that $\mathbf{\Sigma_{xx}} := \mathbb{E}[\mathbf{x}_1 \mathbf{x}_1']$ exists. This is a second moment assumption.

Example 9.1.1. Let's look at a scalar example. Let $\{x_t\}$ be the Markov process in example 8.2.1 on page 244. To repeat,

$$x_{t+1} = \rho |x_t| + (1 - \rho^2)^{1/2} w_{t+1} \quad \text{with} \quad -1 < \rho < 1 \quad \text{and} \quad \{w_t\} \overset{\text{IID}}{\sim} \mathcal{N}(0,1)$$

As discussed in example 8.2.1, the model has a unique, globally stable stationary distribution, given by $\pi_\infty(s) = 2\phi(s)\Phi(qs)$, where $q := \rho(1-\rho^2)^{-1/2}$, ϕ is the standard normal density and Φ is the standard normal cdf. Let's assume that x_0 has density π_∞.[1] In this case, all of the conditions in assumption 9.1.1 are satisfied. Exercise 9.3.2 asks you to step through the details.

Assumption 9.1.2 (Weak exogeneity)**.** The shocks $\{u_t\}$ are IID with $\mathbb{E}[u_t] = 0$ and $\mathbb{E}[u_t^2] = \sigma^2$. Moreover, the shocks are independent of contemporaneous and lagged regressors:

$$u_t \text{ is independent of } \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t \text{ for all } t$$

Remark: Assumption 9.1.2 permits dependence between current shocks and future regressors. It is desirable to admit this possibility in a time series setting, because current shocks usually feed into future state variables.

Example 9.1.2. For example, in the AR(1) regression (9.4), this will be the case whenever the shock process $\{u_t\}$ is IID, because the contemporaneous and lagged regressors $x_1, \ldots, x_t$ are equal to the lagged state variables $y_0, \ldots, y_{t-1}$, which in turn are functions of only y_0 and $u_1, \ldots, u_{t-1}$, and therefore independent of u_t.

[1] In the econometric setting, it is standard to assume that the first data point is drawn from the stationary distribution. This seems justified when the process has been running for a long time, and hence the distribtion of the state has converged to the stationary distribution by the time the first data point is observed.

One consequence of assumption 9.1.2 is that we have

$$\mathbb{E}\left[u_s u_t \mid \mathbf{x}_1, \ldots, \mathbf{x}_t\right] = \begin{cases} \sigma^2 & \text{if} \quad s = t \\ 0 & \text{if} \quad s < t \end{cases} \tag{9.6}$$

The proof is an exercise (exercise 9.3.3)

One of the most important consequences of assumptions 9.1.1 and 9.1.2 for us is that linear functions of $\{\mathbf{x}_t u_t\}$ now form a martingale difference sequence. This allows us to apply the LLN and CLT results in theorem 8.2.3 (page 248). From this LLN and CLT, we will be able to show that the OLS estimator is consistent and asymptotically normal.

Lemma 9.1.1. *If assumptions 9.1.1 and 9.1.2 both hold, then, for any constant vector* $\mathbf{a} \in \mathbb{R}^K$*, the sequence* $\{m_t\}$ *defined by* $m_t = \mathbf{a}'\mathbf{x}_t u_t$ *is*

1. *identically distributed with* $\mathbb{E}\left[m_1^2\right] = \sigma^2 \mathbf{a}' \mathbf{\Sigma}_{\mathbf{xx}} \mathbf{a}$*, and*
2. *a martingale difference sequence with respect to the filtration defined by*

$$\mathcal{F}_t := \{\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{x}_{t+1}, u_1, \ldots, u_t\} \tag{9.7}$$

Proof. First let's check part 1. That $\{m_t\}$ is identically distributed follows from the assumption that $\{u_t\}$ and $\{\mathbf{x}_t\}$ are identically distributed, and that $\mathbf{x}_t$ and u_t are independent.[2] Regarding the second moment $\mathbb{E}\left[m_1^2\right]$, we have

$$\mathbb{E}\left[m_1^2\right] = \mathbb{E}\left[\mathbb{E}\left[u_1^2 (\mathbf{a}'\mathbf{x}_1)^2 \mid \mathbf{x}_1\right]\right] = \mathbb{E}\left[(\mathbf{a}'\mathbf{x}_1)^2 \mathbb{E}\left[u_1^2 \mid \mathbf{x}_1\right]\right]$$

From independence of u_1 and $\mathbf{x}_1$, the inner expectation is σ^2. Moreover,

$$(\mathbf{a}'\mathbf{x}_1)^2 = \mathbf{a}'\mathbf{x}_1 \mathbf{a}'\mathbf{x}_1 = \mathbf{a}'\mathbf{x}_1\mathbf{x}_1'\mathbf{a}$$

$$\therefore \quad \mathbb{E}\left[m_1^2\right] = \mathbb{E}\left[\mathbf{a}'\mathbf{x}_1\mathbf{x}_1'\mathbf{a}\ \sigma^2\right] = \sigma^2 \mathbf{a}' \mathbb{E}\left[\mathbf{x}_1\mathbf{x}_1'\right]\mathbf{a} = \sigma^2 \mathbf{a}' \mathbf{\Sigma}_{\mathbf{xx}} \mathbf{a}$$

To check part 2, note that $\{m_t\}$ is adapted to $\{\mathcal{F}_t\}$, since $m_t := u_t \mathbf{a}'\mathbf{x}_t$ is a function of variables in $\mathcal{F}_t$. Moreover, we have

$$\mathbb{E}\left[m_{t+1} \mid \mathcal{F}_t\right] = \mathbb{E}\left[u_{t+1}\mathbf{a}'\mathbf{x}_{t+1} \mid \mathcal{F}_t\right] = \mathbf{a}'\mathbf{x}_{t+1}\mathbb{E}\left[u_{t+1} \mid \mathcal{F}_t\right] = \mathbf{a}'\mathbf{x}_{t+1}\mathbb{E}\left[u_{t+1}\right] = 0$$

(Here the second equality follows from the fact that $\mathbf{x}_{t+1} \in \mathcal{F}_t$, while the third follows from the independence in assumption 9.1.2.) This confirms that $\{m_t\}$ is a martingale difference sequence with respect to $\{\mathcal{F}_t\}$. □

[2] The distribution of m_t depends on the joint distribution of $\mathbf{a}'\mathbf{x}_t$ and u_t. Since $\mathbf{a}'\mathbf{x}_t$ and u_t are independent, their joint distribution is just the product of their marginal distributions. Since $\{u_t\}$ and $\{\mathbf{x}_t\}$ are identically distributed, these marginal distributions do not dependend on t.

9.1.2 Consistency of $\hat{\boldsymbol{\beta}}_T$

Under the assumptions of the previous section, the OLS estimator is consistent for the parameter vector $\boldsymbol{\beta}$. In particular, we have the following result:

Theorem 9.1.1. *If assumptions 9.1.1 and 9.1.2 both hold, then* $\hat{\boldsymbol{\beta}}_T \xrightarrow{p} \boldsymbol{\beta}$ *as* $T \to \infty$.

Proof. It suffices to show that the expression on the right-hand side of (9.3) converges in probability to $\mathbf{0}$. As a first step, let's show that

$$\frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_t u_t \xrightarrow{p} \mathbf{0} \tag{9.8}$$

is true. In view of fact 2.5.3 on page 77, it suffices to show that, for any $\mathbf{a} \in \mathbb{R}^K$, we have

$$\mathbf{a}' \left[\frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_t u_t \right] \xrightarrow{p} \mathbf{a}'\mathbf{0} = 0 \tag{9.9}$$

If we define $m_t := \mathbf{a}'\mathbf{x}_t u_t$, then (9.9) can be written as $T^{-1}\sum_{t=1}^{T} m_t$. Since $\{m_t\}$ is an identically distributed martingale difference sequence (see lemma 9.1.1 on page 267), the convergence $T^{-1}\sum_{t=1}^{T} m_t \xrightarrow{p} 0$ follows from theorem 8.2.3 (page 248). We have now verified (9.8).

Now let us return to the expression on the right-hand side of (9.3). By assumption 9.1.1 and fact 2.5.2, we see that

$$\left[\frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_t \mathbf{x}_t' \right]^{-1} \xrightarrow{p} {\boldsymbol{\Sigma}_{\mathbf{xx}}}^{-1} \quad \text{as} \quad T \to \infty$$

Appealing to fact 2.5.2 once more, we obtain

$$\hat{\boldsymbol{\beta}}_T - \boldsymbol{\beta} = \left[\frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_t \mathbf{x}_t' \right]^{-1} \cdot \frac{1}{T}\sum_{t=1}^{T} u_t \mathbf{x}_t \xrightarrow{p} {\boldsymbol{\Sigma}_{\mathbf{xx}}}^{-1}\mathbf{0} = \mathbf{0}$$

The proof of theorem 9.1.1 is now done. □

9.1.3 Consistency of $\hat{\sigma}_T^2$

To estimate the variance σ^2 of the error terms, we previously used the expression $\hat{\sigma}_2 := \text{SSR}/(N-K)$, where N was the sample size. In the current setting T is the

sample size, and, since T is assumed to be large relative to K, we have $1/(T-K) \approx 1/T$. Hence for our new expression we just take $\hat{\sigma}_T^2 = \text{SSR}/T$. (None of the following theory is affected if we use $\text{SSR}/(T-K)$ instead.) In summary,

$$\hat{\sigma}_T^2 := \frac{1}{T}\sum_{t=1}^{T} \hat{u}_t^2 =: \frac{1}{T}\sum_{t=1}^{T}(y_t - \mathbf{x}_t'\hat{\boldsymbol{\beta}}_T)^2 \tag{9.10}$$

Theorem 9.1.2. *If assumptions 9.1.1 and 9.1.2 both hold, then $\hat{\sigma}_T^2 \xrightarrow{p} \sigma^2$ as $T \to \infty$.*

Proof. We have

$$\hat{\sigma}_T^2 = \frac{1}{T}\sum_{t=1}^{T}(y_t - \mathbf{x}_t'\hat{\boldsymbol{\beta}}_T)^2 = \frac{1}{T}\sum_{t=1}^{T}[u_t + \mathbf{x}_t'(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_T)]^2$$

Expanding out the square, we get

$$\hat{\sigma}_T^2 = \frac{1}{T}\sum_{t=1}^{T} u_t^2 + 2(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_T)'\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t u_t + (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_T)'\left[\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t\mathbf{x}_t'\right](\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_T)$$

By assumption 9.1.2 and the law of large numbers, the first term on the right-hand side converges in probability to σ^2. Hence it suffices to show that the second and third term converge in probability to zero as $T \to \infty$ (recall fact 1.4.1 on page 31). These results follow from repeated applications of fact 2.5.2 on page 75, combined with various convergence results we have already established. The details are left as an exercise. □

9.2 Asymptotic Normality

Under our assumptions, we will now show that the term $\sqrt{T}(\hat{\boldsymbol{\beta}}_T - \boldsymbol{\beta})$ is asymptotically normal. From this information we can develop asymptotic tests and confidence intervals.

9.2.1 Asymptotic Normality of $\hat{\boldsymbol{\beta}}$

Let's begin by establishing asymptotic normality of the OLS estimator:

Theorem 9.2.1. *Under assumptions 9.1.1 and 9.1.2, the OLS estimator $\hat{\boldsymbol{\beta}}_T =: \hat{\boldsymbol{\beta}}$ satisfies*

$$\sqrt{T}(\hat{\boldsymbol{\beta}}_T - \boldsymbol{\beta}) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{\Sigma}_{\mathbf{xx}}^{-1}) \quad \textit{as} \quad T \to \infty$$

Proof. Using the expression (9.3) we obtain

$$\sqrt{T}(\hat{\boldsymbol{\beta}}_T - \boldsymbol{\beta}) = \left[\frac{1}{T} \sum_{t=1}^{T} \mathbf{x}_t \mathbf{x}_t' \right]^{-1} \cdot \; T^{-1/2} \sum_{t=1}^{T} u_t \mathbf{x}_t \tag{9.11}$$

Let $\mathbf{z}$ be a random variable satisfying $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{\Sigma}_{\mathbf{xx}})$. Suppose we can show that

$$T^{-1/2} \sum_{t=1}^{T} u_t \mathbf{x}_t \xrightarrow{d} \mathbf{z} \quad \text{as} \quad T \to \infty \tag{9.12}$$

If (9.12) is valid, then, applying assumption 9.1.1 along with fact 2.5.5, we obtain

$$\sqrt{T}(\hat{\boldsymbol{\beta}}_T - \boldsymbol{\beta}) = \left[\frac{1}{T} \sum_{t=1}^{T} \mathbf{x}_t \mathbf{x}_t' \right]^{-1} \cdot \; T^{-1/2} \sum_{t=1}^{T} u_t \mathbf{x}_t \;\xrightarrow{d}\; \boldsymbol{\Sigma}_{\mathbf{xx}}^{-1} \mathbf{z}$$

In view of fact 2.4.6 on page 74 and symmetry of $\boldsymbol{\Sigma}_{\mathbf{xx}}^{-1}$,[3] we have

$$\boldsymbol{\Sigma}_{\mathbf{xx}}^{-1} \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{xx}}^{-1} \operatorname{var}[\mathbf{z}] \boldsymbol{\Sigma}_{\mathbf{xx}}^{-1}) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{xx}}^{-1} \sigma^2 \boldsymbol{\Sigma}_{\mathbf{xx}} \boldsymbol{\Sigma}_{\mathbf{xx}}^{-1}) = \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{\Sigma}_{\mathbf{xx}}^{-1})$$

This completes the proof of theorem 9.2.1, conditional on the assumption that (9.12) is valid. Let's now check that (9.12) is valid.

By the Cramer-Wold device (fact 2.5.3 on page 77), it suffices to show that for any $\mathbf{a} \in \mathbb{R}^K$ we have

$$\mathbf{a}' \left[T^{-1/2} \sum_{t=1}^{T} u_t \mathbf{x}_t \right] \xrightarrow{d} \mathbf{a}' \mathbf{z} \tag{9.13}$$

Fixing $\mathbf{a}$ and letting $m_t := u_t \mathbf{a}' \mathbf{x}_t$, the expression on the left of (9.13) can be rewritten as follows:

$$\mathbf{a}' \left[T^{-1/2} \sum_{t=1}^{T} u_t \mathbf{x}_t \right] = T^{-1/2} \sum_{t=1}^{T} u_t \mathbf{a}' \mathbf{x}_t =: T^{-1/2} \sum_{t=1}^{T} m_t$$

Since $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{\Sigma}_{\mathbf{xx}})$, to establish (9.13) we need to show that

$$T^{-1/2} \sum_{t=1}^{T} m_t \xrightarrow{d} \mathcal{N}(0, \sigma^2 \mathbf{a}' \boldsymbol{\Sigma}_{\mathbf{xx}} \mathbf{a}) \tag{9.14}$$

[3] Remember that the transpose of the inverse is the inverse of the transpose, and the transpose of $\boldsymbol{\Sigma}_{\mathbf{xx}}$ is just $\boldsymbol{\Sigma}_{\mathbf{xx}}$, since all variance-covariance matrices are symmetric.

From lemma 9.1.1, we already know that $\{m_t\}$ is an identically distributed with $\mathbb{E}[m_1^2] = \sigma^2 \mathbf{a}' \boldsymbol{\Sigma}_{\mathbf{xx}} \mathbf{a}$, and a martingale difference sequence with respect to the filtration defined by

$$\mathcal{F}_t := \{\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{x}_{t+1}, u_1, \ldots, u_t\}$$

In view of the martingale difference CLT in theorem 8.2.3, the result (9.14) will hold whenever

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[m_t^2 \mid \mathcal{F}_{t-1}] \xrightarrow{p} \sigma^2 \mathbf{a}' \boldsymbol{\Sigma}_{\mathbf{xx}} \mathbf{a} \quad \text{as } T \to \infty \tag{9.15}$$

Since $\mathbf{x}_t \in \mathcal{F}_{t-1}$, we have

$$\mathbb{E}[m_t^2 \mid \mathcal{F}_{t-1}] = \mathbb{E}[u_t^2 (\mathbf{a}' \mathbf{x}_t)^2 \mid \mathcal{F}_{t-1}] = (\mathbf{a}' \mathbf{x}_t)^2 \mathbb{E}[u_t^2 \mid \mathcal{F}_{t-1}] = \sigma^2 (\mathbf{a}' \mathbf{x}_t)^2 = \sigma^2 \mathbf{a}' \mathbf{x}_t \mathbf{x}_t' \mathbf{a}$$

The right-hand side of (9.15) is therefore

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[m_t^2 \mid \mathcal{F}_{t-1}] = \frac{1}{T} \sum_{t=1}^{T} (\sigma^2 \mathbf{a}' \mathbf{x}_t \mathbf{x}_t' \mathbf{a}) = \sigma^2 \mathbf{a}' \left[\frac{1}{T} \sum_{t=1}^{T} \mathbf{x}_t \mathbf{x}_t' \right] \mathbf{a} \xrightarrow{p} \sigma^2 \mathbf{a}' \boldsymbol{\Sigma}_{\mathbf{xx}} \mathbf{a}$$

where the convergence in probability is due to assumption 9.1.1 and (2.7). This verifies (9.15), and completes the proof of theorem 9.2.1. □

9.2.2 Large Sample Tests

In §7.4.2 we considered the problem of testing a hypothesis about an individual coefficient β_k. Let's consider this problem again in the large sample setting. The hypothesis to be tested is

$$H_0 : \beta_k = \beta_k^0 \quad \text{against} \quad H_1 : \beta_k \neq \beta_k^0$$

In the finite sample theory of §7.4.2, we showed that if the error terms are normally distributed, then the expression $(\hat{\beta}_k - \beta_k) / \operatorname{se}(\hat{\beta}_k)$ has the t-distribution with $N - K$ degrees of freedom. In the large sample case, we can use the central limit theorem to show that the same statistic is asymptotically normal. (In a sense, this is not surprising, because the t-distribution converges to the standard normal distribution as the degrees of freedom converges to infinity. However, we cannot use this result directly, as our model assumptions are quite different.)

Theorem 9.2.2. *Let $\hat{\sigma}_T$ be as defined in (9.10) on page 269. Let assumptions 9.1.1 and 9.1.2 hold, and let*

$$\operatorname{se}(\hat{\beta}_k^T) := \sqrt{\hat{\sigma}_T^2 \mathbf{e}_k' (\mathbf{X}'\mathbf{X})^{-1} \mathbf{e}_k}$$

Under the null hypothesis H_0, we have

$$z_k^T := \frac{\hat{\beta}_k^T - \beta_k^0}{\text{se}(\hat{\beta}_k^T)} \xrightarrow{d} \mathcal{N}(0,1) \quad as \quad T \to \infty \tag{9.16}$$

Proof. Recall from theorem 9.2.1 that

$$\sqrt{T}(\hat{\boldsymbol{\beta}}_T - \boldsymbol{\beta}) \xrightarrow{d} \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{\Sigma}_{\mathbf{xx}}^{-1}) \quad \text{as} \quad T \to \infty$$

where, as usual, $\boldsymbol{\beta}$ is the true parameter vector. It now follows (from which facts?) that

$$\mathbf{e}_k'[\sqrt{T}(\hat{\boldsymbol{\beta}}_T - \boldsymbol{\beta})] \xrightarrow{d} \mathbf{e}_k'\mathbf{z} \sim \mathcal{N}(\mathbf{e}_k'\mathbb{E}[\mathbf{z}], \mathbf{e}_k' \operatorname{var}[\mathbf{z}]\mathbf{e}_k)$$

In other words, we have $\sqrt{T}(\hat{\beta}_k^T - \beta_k) \xrightarrow{d} \mathcal{N}(0, \sigma^2 \mathbf{e}_k' \boldsymbol{\Sigma}_{\mathbf{xx}}^{-1} \mathbf{e}_k)$. Making the obvious transformation, we obtain

$$\frac{\sqrt{T}(\hat{\beta}_k^T - \beta_k)}{\sqrt{\sigma^2 \mathbf{e}_k' \boldsymbol{\Sigma}_{\mathbf{xx}}^{-1} \mathbf{e}_k}} \xrightarrow{d} \mathcal{N}(0,1) \tag{9.17}$$

We have already shown that $(T^{-1}\sum_{t=1}^T \mathbf{x}_t \mathbf{x}_t')^{-1} \xrightarrow{p} \boldsymbol{\Sigma}_{\mathbf{xx}}{}^{-1}$. Applying (2.7), we then have

$$T\mathbf{e}_k'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{e}_k = \mathbf{e}_k' \left[\frac{1}{T}\sum_{t=1}^T \mathbf{x}_t \mathbf{x}_t'\right]^{-1} \mathbf{e}_k \xrightarrow{p} \mathbf{e}_k' \boldsymbol{\Sigma}_{\mathbf{xx}}{}^{-1} \mathbf{e}_k$$

Recall that $\hat{\sigma}_T^2 \xrightarrow{p} \sigma^2$, as shown in theorem 9.1.2. Using this result plus fact 1.4.1 on page 31, we then have

$$1/\sqrt{\hat{\sigma}_T^2 \, T\mathbf{e}_k'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{e}_k} \xrightarrow{p} 1/\sqrt{\sigma^2 \mathbf{e}_k' \boldsymbol{\Sigma}_{\mathbf{xx}}{}^{-1} \mathbf{e}_k}$$

In view of (9.17) and fact 1.4.5 on page 34, we then have

$$\frac{\sqrt{T}(\hat{\beta}_k^T - \beta_k)}{\sqrt{\hat{\sigma}_T^2 \, T\mathbf{e}_k'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{e}_k}} \xrightarrow{d} \mathcal{N}(0,1)$$

Assuming validity of the null hypothesis (so that $\beta_k^0 = \beta^k$) and cancelling $\sqrt{T}$, we have established (9.16). □

9.3 Exercises

Ex. 9.3.1. Verify expression (9.2). Recall here that $\mathbf{x}_t$ is the t-th row of $\mathbf{X}$.

Ex. 9.3.2. In example 9.1.1 it was claimed that the threshold process studied in that example satisfies all of the conditions of assumption 9.1.1. Verify that this is the case.

Ex. 9.3.3. Verify the claim that (9.6) holds when assumption 9.1.2 is valid.

Ex. 9.3.4. Let $K \times 1$ random vector $\hat{\boldsymbol{\theta}}_T$ be an estimator of $\boldsymbol{\theta}$. Suppose that this estimator is asymptotically normal, in the sense that

$$\sqrt{T}(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \mathbf{C})$$

where $\mathbf{C}$ is symmetric and positive definite. It is know that for such a $\mathbf{C}$ there exists a $K \times K$ matrix $\mathbf{Q}$ such that $\mathbf{QCQ}' = \mathbf{I}$. Let $\hat{\mathbf{Q}}_T$ be a consistent estimator of $\mathbf{Q}$. Show that

$$T\|\hat{\mathbf{Q}}_T(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta})\|^2 \xrightarrow{d} \chi^2(K) \tag{9.18}$$

Ex. 9.3.5 (To be written. derive ols estimator of ρ in the model of example 8.2.1. (see zhao.) is it consistent?)**.**

Ex. 9.3.6 (as in example 9.1.2, but u_t itself AR(1). then assumptions fail. make this an exercise?)**.**

9.3.1 Solutions to Selected Exercises

Solution to Exercise 9.3.2. The process $\{x_t\}$ studied in example 9.3.2 is identically distributed as a result of our assumption that $x_0 \sim \pi_\infty$ (see fact 8.2.1 on page 243). It remains to check the conditions on $\boldsymbol{\Sigma}_{\mathbf{xx}}$. In the present case,

$$\boldsymbol{\Sigma}_{\mathbf{xx}} = \mathbb{E}\left[x_t^2\right] = \int_{-\infty}^{\infty} s^2 \pi_\infty(s) ds = \int_{-\infty}^{\infty} s^2 2\phi(s)\Phi(qs) ds$$

where $q := \rho(1-\rho^2)^{-1/2}$, ϕ is the standard normal density and Φ is the standard normal cdf. To verify that $\boldsymbol{\Sigma}_{\mathbf{xx}}$ is positive definite, we need to check that the term on the right-hand side is strictly positive. This is clearly true, because the function inside the integral is strictly positive everywhere but zero. To be careful, we should also check that $\boldsymbol{\Sigma}_{\mathbf{xx}}$ is finite, and this is also true because $\Phi(qs) \leq 1$, and hence

$$\int_{-\infty}^{\infty} s^2 2\phi(s)\Phi(qs) ds \leq \int_{-\infty}^{\infty} s^2 2\phi(s) ds = 2\int_{-\infty}^{\infty} s^2 \phi(s) ds = 2$$

Finally, we need to show that

$$\frac{1}{T}\sum_{t=1}^{T} x_t^2 \xrightarrow{p} \boldsymbol{\Sigma}_{\mathbf{xx}} = \mathbb{E}\left[x_t^2\right] \tag{9.19}$$

Since the conditions of theorem 8.2.1 (page 244) are satisfied, we can appeal to theorem 8.2.2 (page 244). This theorem confirms that the convergence in (9.19) is valid. □

Solution to Exercise 9.3.3. Suppose that assumption 9.1.2 (page 266) is valid. We need to show that

$$\mathbb{E}\left[u_s u_t \mid \mathbf{x}_1, \ldots, \mathbf{x}_t\right] = \begin{cases} \sigma^2 & \text{if} \quad s = t \\ 0 & \text{if} \quad s < t \end{cases}$$

On one hand, if $s = t$, then $\mathbb{E}\left[u_t^2 \mid \mathbf{x}_1, \ldots, \mathbf{x}_t\right] = \mathbb{E}\left[u_t^2\right] = \sigma^2$ by independence. On the other hand, if $s < t$, then

$$\begin{aligned} \mathbb{E}\left[u_s u_t \mid \mathbf{x}_1, \ldots, \mathbf{x}_t\right] &= \mathbb{E}\left[\mathbb{E}\left[u_s u_t \mid \mathbf{x}_1, \ldots, \mathbf{x}_t, u_s\right] \mid \mathbf{x}_1, \ldots, \mathbf{x}_t\right] \\ &= \mathbb{E}\left[u_s \mathbb{E}\left[u_t \mid \mathbf{x}_1, \ldots, \mathbf{x}_t, u_s\right] \mid \mathbf{x}_1, \ldots, \mathbf{x}_t\right] \\ &= \mathbb{E}\left[u_s \mathbb{E}\left[u_t\right] \mid \mathbf{x}_1, \ldots, \mathbf{x}_t\right] \\ &= \mathbb{E}\left[u_s 0 \mid \mathbf{x}_1, \ldots, \mathbf{x}_t\right] = 0 \end{aligned}$$

□

Solution to Exercises 9.3.4. By assumption we have

$$\hat{\mathbf{Q}}_T \xrightarrow{p} \mathbf{Q} \quad \text{and} \quad \sqrt{T}(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}) \xrightarrow{d} \mathbf{z}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$. Applying Slutsky's theorem, we obtain

$$\hat{\mathbf{Q}}_T \sqrt{T}(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}) \xrightarrow{d} \mathbf{Q}\mathbf{z} \tag{9.20}$$

Clearly $\mathbf{Q}\mathbf{z}$ is normally distributed with mean $\mathbf{0}$. Moreover,

$$\operatorname{var}[\mathbf{Q}\mathbf{z}] = \mathbf{Q} \operatorname{var}[\mathbf{z}] \mathbf{Q}' = \mathbf{Q}\mathbf{C}\mathbf{Q}' = \mathbf{I}$$

In other words, $\mathbf{Q}\mathbf{z}$ is standard normal. As a result, $\|\mathbf{Q}\mathbf{z}\|^2 \sim \chi^2(K)$. Applying the continuous mapping theorem to (9.20), we obtain

$$\|\hat{\mathbf{Q}}_T \sqrt{T}(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta})\|^2 \xrightarrow{d} \|\mathbf{Q}\mathbf{z}\|^2 \sim \chi^2(K)$$

This is equivalent to (9.18).

Incidentally, it should be clear that (9.18) can be used to test the null hypothesis that $\boldsymbol{\theta} = \boldsymbol{\theta}_0$. Under the null hypothesis, we have

$$T\|\hat{\mathbf{Q}}_T(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}_0)\|^2 \xrightarrow{d} \chi^2(K)$$

Fixing α and letting c be the $1-\alpha$ quantile of the $\chi^2(K)$ distribution, we reject the null if

$$T\|\hat{\mathbf{Q}}_T(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}_0)\|^2 > c$$

This test is asymptotically of size α. □

Chapter 10

Further Topics

[roadmap]

10.1 Model Selection

[roadmap]

- Don't use tests

10.1.1 Ridge Regression

We are going to begin our discussion of model selection by introducing ridge regression. Ridge regression is an important method in its own right, with connections to many areas of statistics and approximation theory. Moreover, it immediately presents us will a class of models we need to choose between, and hence a model selection problem.

Let's begin by putting ourselves in the classical OLS setting of chapter 7. In particular, we will assume that $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u}$, where $\boldsymbol{\beta}$ is unknown, and $\mathbf{u}$ is unobservable, has unknown distribution, but satisfies $\mathbb{E}[\mathbf{u} \mid \mathbf{X}] = \mathbf{0}$ and $\mathbb{E}[\mathbf{u}\mathbf{u}' \mid \mathbf{X}] = \sigma^2\mathbf{I}$ for some unknown $\sigma > 0$. In traditional OLS theory, we estimate $\boldsymbol{\beta}$ with the OLS estimator

$$\hat{\boldsymbol{\beta}} := (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \operatorname*{argmin}_{\mathbf{b}\in\mathbb{R}^K} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 = \operatorname*{argmin}_{\mathbf{b}\in\mathbb{R}^K} \sum_{n=1}^{N} (y_n - \mathbf{x}_n'\mathbf{b})^2$$

In many ways, $\hat{\boldsymbol{\beta}}$ is a natural choice for estimating $\boldsymbol{\beta}$. Firstly, it minimizes the empirical risk corresponding to the natural risk function $R(f) = \mathbb{E}\left[(y - f(\mathbf{x}))^2\right]$ when the hypothesis space $\mathcal{F}$ is the set of linear functions. Second, under our current assumptions, it is unbiased for $\boldsymbol{\beta}$ (theorem 7.2.1 on page 198), and, moreover, it has the lowest variance among all linear unbiased estimators of $\boldsymbol{\beta}$ (see the Gauss-Markov theorem on page 200).

These results, and, in particular, the Gauss-Markov theorem are much celebrated foundation stones of standard OLS theory. But at least some of this celebration is misplaced. Rather than looking at whether an estimator is best linear unbiased, a better way to evaluate the estimator is to consider its mean squared error, which tells us directly how much probability mass the estimator puts around the object it's trying to estimate. (This point was illustrated in figure 4.4 on page 118.) In the vector case, the mean squared error of a estimator $\hat{\mathbf{b}}$ of $\boldsymbol{\beta}$ is defined as

$$\text{mse}(\hat{\mathbf{b}}) := \mathbb{E}\left[\|\hat{\mathbf{b}} - \boldsymbol{\beta}\|^2\right]$$

It is an exercise (exercise 10.4.1) to show that the mean squared error can also be expressed as

$$\text{mse}(\hat{\mathbf{b}}) = \mathbb{E}\left[\|\hat{\mathbf{b}} - \mathbb{E}\left[\hat{\mathbf{b}}\right]\|^2\right] + \|\mathbb{E}\left[\hat{\mathbf{b}}\right] - \boldsymbol{\beta}\|^2 \tag{10.1}$$

This equation is analogous to (4.11) on page 118, and tells us that the mean squared error is the sum of "variance" and "bias." To minimize mean squared error we face a trade off between these two terms. In many situations involving trade off, the optimal choice is not at either extreme, but somewhere in the middle. Many estimation techniques exhibit this property: Mean squared error is at its minimum not when bias is zero, but rather when some small amount of bias is admitted.

Applying this idea to the OLS setting, it turns out that we can find a (biased) linear estimator that has lower mean squared error that $\hat{\boldsymbol{\beta}}$. The estimator is defined as the solution to the modified least squares problem

$$\min_{\mathbf{b} \in \mathbb{R}_K} \left\{ \sum_{n=1}^{N} (y_n - \mathbf{x}_n' \mathbf{b})^2 + \lambda \|\mathbf{b}\|^2 \right\} \tag{10.2}$$

where $\lambda \geq 0$ is called the regularization parameter. In solving (10.2), we are minimizing the empirical risk plus a term that penalizes large values of $\|\mathbf{b}\|$. The effect is to "shrink" the solution relative to the unpenalized solution $\hat{\boldsymbol{\beta}}$. A bit of calculus shows that the solution to (10.2) is

$$\hat{\boldsymbol{\beta}}_\lambda := (\mathbf{X}'\mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}'\mathbf{y} \tag{10.3}$$

Minimizing the objective in (10.2) is certainly a less obvious approach than simply minimizing the empirical risk $\sum_{n=1}^{N}(y_n - \mathbf{x}_n'\mathbf{b})^2 = \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2$. One indication as to why it might be a good idea comes from regularization theory. To illustrate regularization, suppose that $\mathbf{A}\mathbf{b} = \mathbf{c}$ is an overdetermined system, where $\mathbf{A}$ is $N \times K$ with $N > K$. Let $\mathbf{b}^*$ be the least squares solution: $\mathbf{b}^* = \text{argmin}_{\mathbf{b}} \|\mathbf{A}\mathbf{b} - \mathbf{c}\|^2$. Suppose in addition that $\mathbf{c}$ cannot be calculated perfectly, due to some form of measurement error. Instead we observe $\mathbf{c}_0 \approx \mathbf{c}$. In the absence of additional information, you might guess that the best way to compute an approximation to $\mathbf{b}^*$ is to solve $\min_{\mathbf{b}} \|\mathbf{A}\mathbf{b} - \mathbf{c}_0\|^2$, obtaining the least squares solution to the system $\mathbf{A}\mathbf{b} = \mathbf{c}_0$. Surprisingly, it turns out that this is not always the case, especially when the columns of $\mathbf{A}$ are almost linearly dependent. Instead, one often does better by minimizing $\|\mathbf{A}\mathbf{b} - \mathbf{c}_0\|^2 + \lambda\|\mathbf{b}\|^2$ for some small but positive λ. This second approach is called **Tikhonov regularization**.

While the theory of Tikhonov regularization is too deep to treat in detail here, we can illustrate the rather surprising benefits of regularization with a simulation. In our simulation, $\mathbf{A}$ will be chosen fairly arbitrarily, but such that the columns are quite close to being linearly dependent. To simplify, we first set $\mathbf{b}^* := (10, 10, \ldots, 10)$, and then set $\mathbf{c} := \mathbf{A}\mathbf{b}^*$. By construction, $\mathbf{b}^*$ is then a solution to the system $\mathbf{A}\mathbf{b}^* = \mathbf{c}$, and also the least squares solution (because it solves $\min_{\mathbf{b}} \|\mathbf{A}\mathbf{b} - \mathbf{c}\|^2$).

When measuring $\mathbf{c}$ we will corrupt it with a Gaussian shock. In particular, we draw $\mathbf{c}_0 \sim \mathcal{N}(\mathbf{c}, \sigma^2\mathbf{I})$ where σ is a small positive number. We then plot the ordinary least squares solution based on $\mathbf{c}_0$, which minimizes $\|\mathbf{A}\mathbf{b} - \mathbf{c}_0\|^2$, and the regularized solution, which minimizes $\|\mathbf{A}\mathbf{b} - \mathbf{c}_0\|^2 + \lambda\|\mathbf{b}\|^2$. The former are plotted against their index in black, while the latter are plotted in blue. The true solution $\mathbf{b}^*$ is plotted in red. The result is figure 10.1. The figure shows 20 solutions each for the ordinary and regularized solutions, corresponding to 20 draws of $\mathbf{c}_0$. The regularized solutions are clearly better on average.

Of course, this result is dependent on a reasonable choice for λ. If you experiment with the code (listing 17), you will see that for very small values of λ, the regularized solutions are almost the same as the unregularized solutions. Conversely, very large values of λ pull the regularized solutions too close to the zero vector. Such a situation is depicted in figure 10.2, where the value of λ has been increased by a factor of 50. We can see from the figure that the variance of the regularized solutions has fallen even further, but they are now substantially biased.

Tikhonov regularization gives some understanding as to why the ridge regression estimator $\hat{\boldsymbol{\beta}}_\lambda$ can perform better than $\hat{\boldsymbol{\beta}}$. While $\hat{\boldsymbol{\beta}}$ is the solution to the least squares

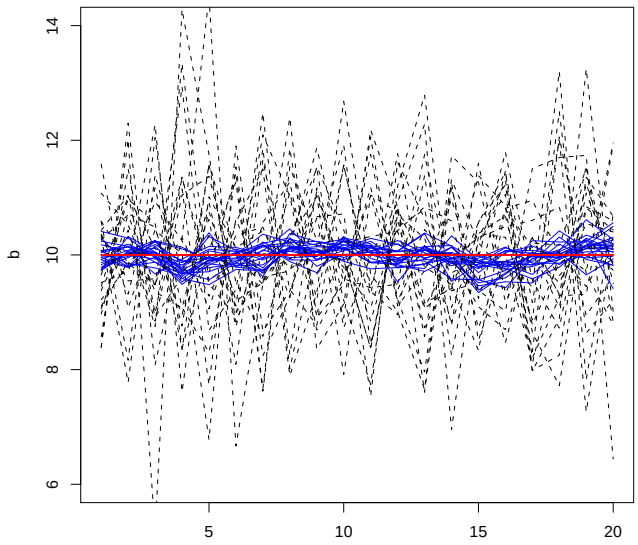


Figure 10.1: Effect of Tikhonov regularization, $\lambda = 1$

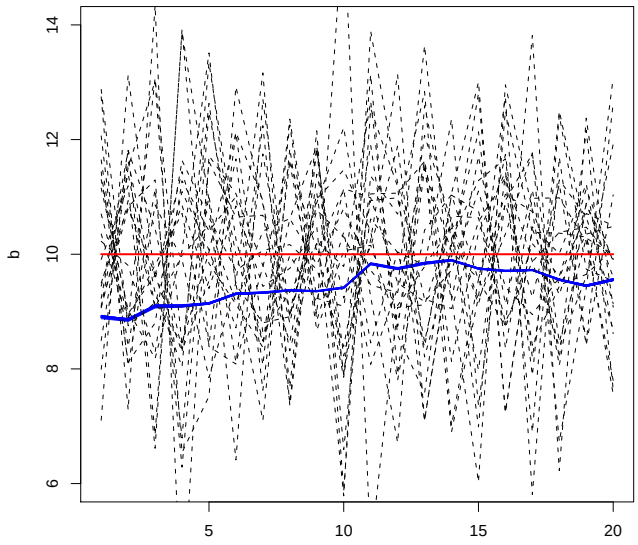


Figure 10.2: Effect of Tikhonov regularization, $\lambda = 50$

Listing 17 Effect of Tikhonov regularization

```
sigma <- 0.5    # Parameterizes measurement error for c
lambda <- 1     # Regularization parameter
numreps <- 20   # Number of times to solve system

# Construct an arbitrary N x K matrix A
N <- 40; K <- 20
A <- matrix(nrow=N, ncol=K)
A[,1] <- 1
for (i in 2:K) {
    A[,i] <- A[,i-1] + rnorm(N, sd=0.1)
}

Ap <- t(A)              # A transpose
bstar <- rep(10, K)     # True solution
c <- A %*% bstar        # Corresponding c

# Create empty plot
plot(bstar, type="n", ylim=c(6, 14), xlab="", ylab="b")
# Plot the solutions
for (j in 1:numreps) {
    # Observe c with error
    c0 <- c + rnorm(N, sd=sigma)
    # Compute the regularized solution
    b1 <- solve((Ap %*% A + lambda * diag(K)), Ap %*% c0)
    lines(b1, col="blue")
    # Compute the standard least squares solution
    b2 <- lm(c0 ~ 0 + A)$coefficients
    lines(b2, col="black", lty=2)
}
lines(bstar, lwd=2, col="red")
```

problem $\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2$, the ridge regression estimator $\hat{\boldsymbol{\beta}}_\lambda$ is the solution to the regularized problem $\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \lambda\|\mathbf{b}\|^2$. Since $\mathbf{y}$ is indeed a noisy observation, we can expect that the regularized estimator will sometimes perform better.

In fact, it turns out that $\hat{\boldsymbol{\beta}}_\lambda$ can *always* outperform $\hat{\boldsymbol{\beta}}$, in the sense that there always exists a $\lambda > 0$ such that $\text{mse}(\hat{\boldsymbol{\beta}}_\lambda) < \text{mse}(\hat{\boldsymbol{\beta}})$. This was proved by Hoerl and Kennard (1970), and the details of the argument can be found there. As mentioned above, this implies that the estimator $\hat{\boldsymbol{\beta}}_\lambda$ is biased (see exercise 10.4.2). The reduction in mean squared error over the least squares estimator occurs because, for some intermediate value of λ, the variance of $\hat{\boldsymbol{\beta}}_\lambda$ falls by more than enough to offset the extra bias.

It is worth emphasizing two things before we move on. One is that, with the right choice of λ, the ridge regression estimator $\hat{\boldsymbol{\beta}}_\lambda$ outperforms $\hat{\boldsymbol{\beta}}$ even though all of the classical OLS assumptions are completely valid. The other is that the right choice of λ is an important and nontrivial problem. This problem falls under the heading of model selection, which is the topic treated in the next few sections.

10.1.2 Subset Selection and Ridge Regression

One problem frequently faced in specific regression problems is which variables to include. For example, if we are comparing crime rates across different cites, we can think of any number of variables that might be relevant (median wage, unemployment, police density, population, etc., etc.). The same is true if we are trying to model credit default rates for some group of firms or individuals, educational attainment across schools, adoption of technologies, demand for certain products, and so on. And a similar problem of variable selection arises in time series models, where we want to know how many lags of the state variables to include. The general problem is known as **subset selection**, since we are trying to choose the right subset of all candidate regressors.

This problem takes on another dimension (no pun intended) when we start to think about basis functions. Given a set of covariates $\mathbf{x}$, we have the option to map this into a larger vector $\boldsymbol{\phi}(\mathbf{x})$ using basis functions, as discussed in §6.2.1. For example, given a single covariate x, we may consider mapping it into $\boldsymbol{\phi}(x) = (1, x, x^2, \ldots, x^d)$ and regressing y on $\boldsymbol{\phi}(x)$. This amounts to polynomial regression, where the hypothesis space is

$$\mathcal{F} := \left\{ \text{all functions of the form } f(x) = b_0 x^0 + b_1 x^1 + \cdots + b_d x^d \right\}$$

Polynomial regression was discussed extensively in §4.6.2. As we saw in figures 4.18–4.21 (page 145), a good choice of d is crucial. Choosing d is another example of the subset selection problem because we are trying to decide whether to include the regressor x^j for some given j.

Subset selection can be viewed from the lens of empirical risk minimization. Suppose that we are trying to model a given system with output y and inputs $\mathbf{x}$. We imagine that $\mathbf{x}$ is a large set of K candidate regressors. We will also suppose that we have already applied any transformations we think might be necessary. For example, $\mathbf{x}$ might include not just the original regressors but squares of the regressors, cross-products, and so on. (In other words, we have already applied the basis functions.) If we want to include all regressors then we can minimize empirical risk (i.e., solve the least squares problem) over the hypothesis space of linear functions from $\mathbb{R}^K$ to $\mathbb{R}$:

$$\mathcal{L} := \left\{ \text{ all functions of the form } f(\mathbf{x}) = \mathbf{b}'\mathbf{x} \text{ with } b_1, \ldots, b_K \in \mathbb{R} \right\}$$

Doing so produces the usual OLS estimator.

Now suppose that we wish to exclude some subset of regressors $\{x_i, \ldots, x_j\}$. Let $I \subset \{1, \ldots, K\}$ be the set of indices of the regressors we want to exclude. Regressing y on the remaining regressors $\{x_k\}_{k \notin I}$ is equivalent to minimizing the empirical risk over the hypothesis space

$$\mathcal{L}_{-I} := \left\{ \text{ all functions } f(\mathbf{x}) = \mathbf{b}'\mathbf{x} \text{ with } b_k = 0 \text{ for all } k \in I \right\}$$

Once again, we are back to the problem of choosing a suitable hypothesis space over which to minimize empirical risk.

The subset selection problem has been tackled by many researchers. Well-known approaches include those based on the Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC) and Mallow's C_p statistic. For example, Mallow's C_p statistic consists of two terms, one increasing in the size of the empirical risk, and the other increasing in $\#I$, the size of the subset selected. The objective is to minimize the statistic, which involves trading off poor fit (large empirical risk) against excess complexity of the hypothesis space (large $\#I$).

One of the problems with subset selection is that there is usually a large number of possible subsets. With K regressors, there are 2^K subsets to step through. To avoid this problem, one alternative is to use ridge regression. With ridge regression, the regularization term leads us to choose an estimate with smaller norm. What

this means in practice is that the coefficients of less helpful regressors are driven towards zero. The effect is to "almost exclude" those regressors. Of course, the model selection problem is not solved, because we still need to choose the value of the regularization parameter λ. However, the problem has been reduced to tuning a single parameter, rather than searching over 2^K subsets.

We can illustrate the idea by reconsidering the regression problem discussed in §4.6.2. Figures 4.18–4.21 (see page 145) showed the fit we obtained by minimizing empirical risk over larger and larger hypothesis spaces. The hypothesis spaces were the sets $\mathcal{P}_d$ of degree d polynomials for different values of d. For each d we minimized the empirical risk over $\mathcal{P}_d$, which translates into solving

$$\min_{\mathbf{b}} \sum_{n=1}^{N} (y_n - \mathbf{b}'\boldsymbol{\phi}(x_n))^2 \quad \text{where} \quad \boldsymbol{\phi}(x) = (x^0, x^1, \ldots, x^d)$$

As discussed above, choosing the right d is a essentially a subset selection problem, because we are deciding what powers of x to include as regressors. Figure 4.17 (page 143) showed that intermediate values of d did best at minimizing risk.

We can do a very similar thing using ridge regression. First, let's take as our hypothesis space the relatively large space $\mathcal{P}_{14}$. This space is certainly large enough to provide a good fit to the data, but with empirical risk minimization the result is overfitting (see figure 4.21 on page 146). Here, instead of using empirical risk minimization, we solve the regularized problem

$$\min_{\mathbf{b}} \sum_{n=1}^{N} \left\{ (y_n - \mathbf{b}'\boldsymbol{\phi}(x_n))^2 + \lambda \|\mathbf{b}\|^2 \right\}$$

for different values of λ. The data used here is exactly the same data used in the original figures 4.18–4.21 from §4.6.2. The solution for each λ we denote by $\hat{\boldsymbol{\beta}}_\lambda$, which is the ridge regression estimator, and the resulting prediction function we denote by $\hat{f}_\lambda$, so that $\hat{f}_\lambda(x) = \hat{\boldsymbol{\beta}}_\lambda'\boldsymbol{\phi}(x)$.

The function $\hat{f}_\lambda$ is plotted in red for increasingly larger values of λ over figures 10.3–10.6. The black line is the risk minimizing function. In figure 10.3, the value of λ is too small to impose any real restriction, and the procedure overfits. In figure 10.4, the value of λ is a bit larger, and the fit is good. In figures 10.5 and 10.6, the value of λ is too large, and all coefficients are shrunk towards zero.

As in §4.6.2, we can compute the risk of each function $\hat{f}_\lambda$, since we know the underlying model (see (4.27) on page 142). The risk is plotted against λ in figure 10.7. The x-axis is on log-scale. On the basis of what we've seen so far, it's not surprising that risk is smallest for small but nonzero values of λ.

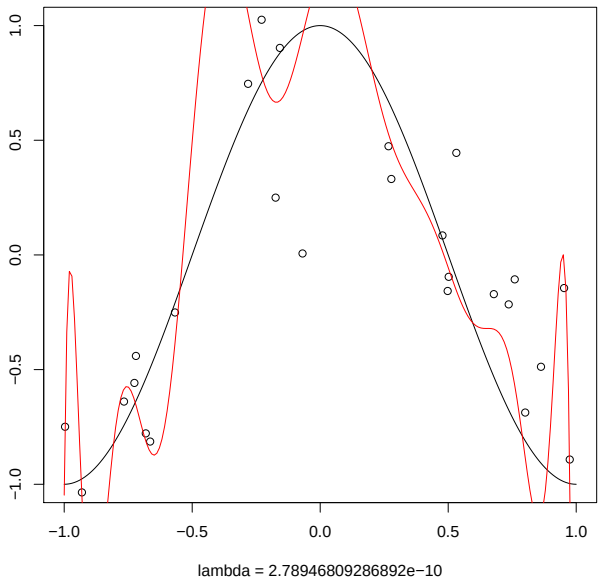


Figure 10.3: Fitted polynomial, $\lambda \approx 0$

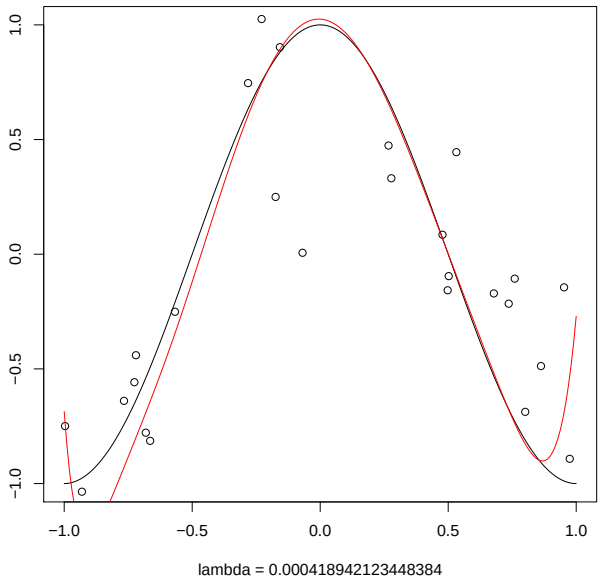


Figure 10.4: Fitted polynomial, $\lambda \approx 0.0004$

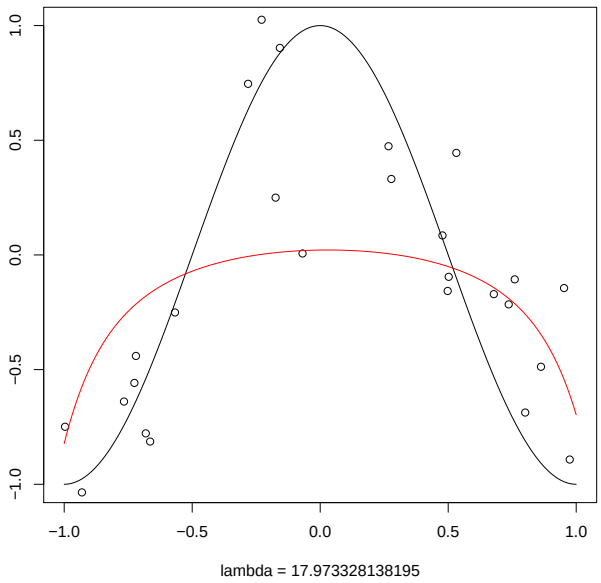


Figure 10.5: Fitted polynomial, $\lambda \approx 18$

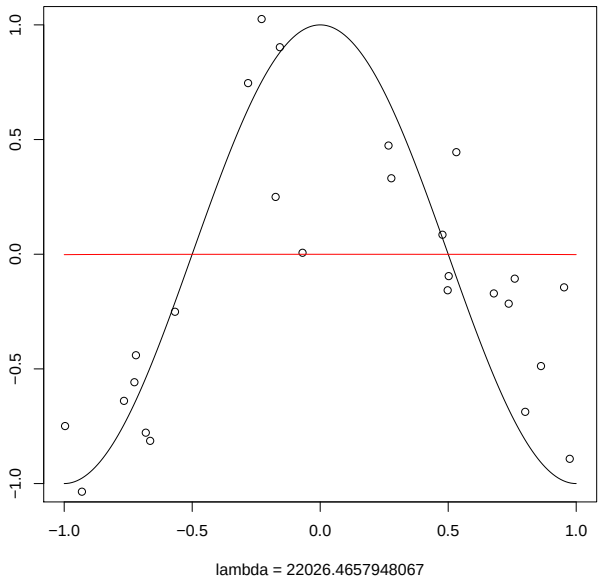


Figure 10.6: Fitted polynomial, $\lambda \approx 2200$

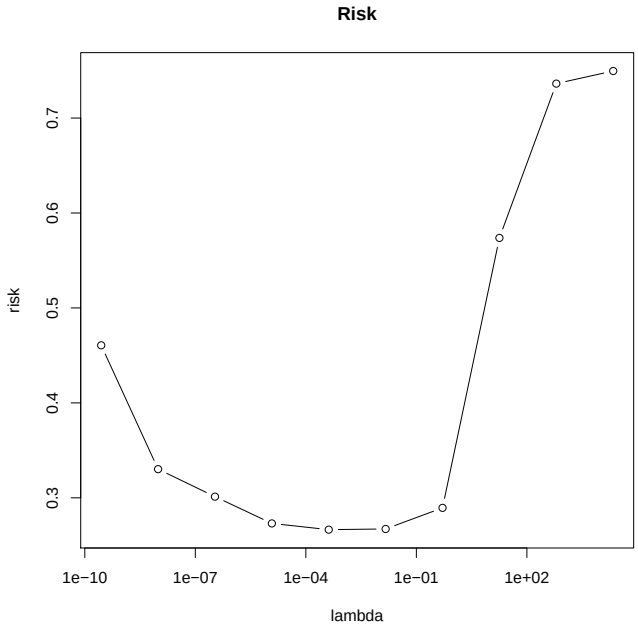


Figure 10.7: The risk of $\hat{f}_\lambda$ plotted against λ

10.1.3 A Bayesian Perspective

The ideal case with model selection is that we have clear guidance from theory on which regressors to include, which to exclude, which functional forms to use, which values of our regularization parameter to choose, and so on. If theory or prior knowledge provides this information then every effort should be made to exploit it. One technique for injecting prior information into statistical estimation is via Bayesian analysis. Bayesian methods are currently very popular in econometrics and other fields of statistics (such as machine learning), and perhaps a future version of these notes will give them more attention. Nevertheless, the brief treatment we present in this section does provide useful intuition on their strengths and weaknesses. In what follows, we focus on Bayesian linear regression.

To begin, let's recall **Bayes' formula**, which states that for any sets A and B we have

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(B \mid A)\mathbb{P}(A)}{\mathbb{P}(B)}$$

This formula follows easily from the definition of conditional probability on page 6. An analogous statement holds true for densities, although the derivation is a bit

more involved.

The main idea of Bayesian analysis is to treat parameters as random variables, in the sense of being unknown quantities for which we hold subjective beliefs regarding their likely values. These subjective beliefs are called **priors**. Suppose for example that we observe input-output pairs $(y_1, \mathbf{x}_1), \ldots, (y_N, \mathbf{x}_N)$. We assume that the pairs satisfy $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u}$. To simplify the presentation we will assume that $\mathbf{X}$ is non-random. (Taking $\mathbf{X}$ to be random leads to the same conclusions but with a longer derivation. See, for example, Bishop, 2006, chapter 3). As before, $\mathbf{u}$ is random and unobservable. The new feature provided by the Bayesian perspective is that we take $\boldsymbol{\beta}$ to be random (and unobservable) as well. While $\mathbf{u}$ and $\boldsymbol{\beta}$ are unobservable random quantities, let's suppose that we have subjective prior beliefs regarding their likely values, expressed in the form of probability distributions. Here we will take the priors to be $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \sigma\mathbf{I})$ and $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \tau\mathbf{I})$.

Given our model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u}$, our prior on $\mathbf{u}$ implies that the density of $\mathbf{y}$ given $\boldsymbol{\beta}$ is $\mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma\mathbf{I})$. In generic notation, we can write our distributions as

$$p(\mathbf{y} \mid \boldsymbol{\beta}) = \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma\mathbf{I}) \quad \text{and} \quad p(\boldsymbol{\beta}) = \mathcal{N}(\mathbf{0}, \tau\mathbf{I}) \tag{10.4}$$

Applying Bayes formula to the pair $(\mathbf{y}, \boldsymbol{\beta})$, we obtain

$$p(\boldsymbol{\beta} \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \boldsymbol{\beta}) p(\boldsymbol{\beta})}{p(\mathbf{y})} \tag{10.5}$$

The left-hand side is called the **posterior** density of $\boldsymbol{\beta}$ given the data $\mathbf{y}$, and represents our new beliefs updated from the prior on the basis of the data $\mathbf{y}$.

Often we wish to summarize the information contained in the posterior, by looking at the "most likely" value of $\boldsymbol{\beta}$ given our priors and the information contained in the data. We can do this by looking either at the mean of the posterior, or at its maximum value. The maximizer of the posterior is called the **maximum a posteriori probability (MAP) estimate**. Taking logs of (10.5) and dropping the term that does not contain $\boldsymbol{\beta}$, it can be expressed as

$$\hat{\boldsymbol{\beta}}_M := \operatorname*{argmax}_{\boldsymbol{\beta}} \{\ln p(\mathbf{y} \mid \boldsymbol{\beta}) + \ln p(\boldsymbol{\beta})\} \tag{10.6}$$

Inserting the distributions in (10.4), dropping constant terms and multiplying by -1, we obtain (exercise 10.4.3) the expression

$$\hat{\boldsymbol{\beta}}_M = \operatorname*{argmin}_{\boldsymbol{\beta}} \left\{ \sum_{n=1}^{N} (y_n - \mathbf{x}_n'\boldsymbol{\beta})^2 + \frac{\sigma^2}{\tau^2} \|\boldsymbol{\beta}\|^2 \right\} \tag{10.7}$$

This is exactly equivalent to the penalized least squares problem (10.2) on page 277, where the regularization parameter λ is equal to $(\sigma/\tau)^2$. In view of (10.3), the solution is

$$\hat{\boldsymbol{\beta}}_M := (\mathbf{X}'\mathbf{X} + (\sigma/\tau)^2\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}$$

Thus, Bayesian estimation provides a principled derivation of the penalized least squares method commonly known as ridge regression. Previously, we justified ridge regression via Tikhonov regularization. Here, Bayesian analysis provides the same regularization, where regularization arises out of combining prior knowledge with the data. Moreover, at least in principle, the value $(\sigma/\tau)^2$ is part of our prior knowledge, and hence there is no model selection problem.

In practice one can of course question the assertion that we have so much prior knowledge that the regularization parameter $\lambda := (\sigma/\tau)^2$ is pinned down. If not, then we are back at the model selection problem. In the next section we forgo the assumption that this strong prior knowledge is available, and consider a more automated approach to choosing λ.

10.1.4 Cross-Validation

The most natural way to think about model selection is to think about minimizing risk. Recall that, given loss function L and a system producing input-output pairs $(y, \mathbf{x}) \in \mathbb{R}^{K+1}$ with joint density p, the risk of a function $f\colon \mathbb{R}^K \to \mathbb{R}$ is the expected loss

$$R(f) := \mathbb{E}\left[L(y, f(\mathbf{x}))\right] = \int\int L(t, f(\mathbf{s}))p(t, \mathbf{s})\, dt\, d\mathbf{s}$$

that occurs when we use $f(\mathbf{x})$ to predict y. Now suppose that we observe N IID input-output pairs

$$\mathcal{D} := \{(y_1, \mathbf{x}_1), \ldots (y_N, \mathbf{x}_N)\}$$

Intuitively, given a selection of models (or estimation procedures), we would like to find the one that takes this data set $\mathcal{D}$ and returns a predictor $\hat{f}$ such that $\hat{f}$ has lower risk than the predictors returned by the other models. Here we have to be a bit careful in defining risk, because if we simply define the risk as $\mathbb{E}\left[L(y, \hat{f}(\mathbf{x}))\right]$ then we are taking expectation over all randomness, including that in $\hat{f}$, which depends implicitly on the data set $\mathcal{D}$. That's not a bad idea per-se, and we discuss it further in §10.1.5. But what we want to do for now is just take the data set as given, and see how well we can do in terms of predicting new values as evaluated by expected

loss. Hence we define the risk of $\hat{f}$ as the expected loss taking $\mathcal{D}$ (and hence $\hat{f}$) as given:

$$R(\hat{f} \mid \mathcal{D}) := \mathbb{E}\left[L(y, \hat{f}(\mathbf{x})) \mid \mathcal{D}\right] = \int \int L(t, \hat{f}(\mathbf{s}))p(t, \mathbf{s})\, dt\, d\mathbf{s}$$

If we have a collection of models M indexed by m, and $\hat{f}_m$ is the predictor produced by fitting model m with data $\mathcal{D}$, then we would like to find the model m^* such that

$$R(\hat{f}_{m^*} \mid \mathcal{D}) \leq R(\hat{f}_m \mid \mathcal{D}) \quad \text{for all} \quad m \in M$$

The obvious problem with this idea is that risk is unobservable. If we knew the joint density p then we could calculate it, but then again, if we knew p there would be no need to estimate anything in the first place.

Looking at this problem, you might have the following idea: Although we don't know p, we do have the data $\mathcal{D}$, which consists of IID draws from p. From the law of large numbers, we know that expectations can be approximated by averages over IID draws, so we could approximate $R(\hat{f} \mid \mathcal{D})$ by

$$\frac{1}{N} \sum_{n=1}^{N} L(y_n, \hat{f}(\mathbf{x}_n))$$

where the pairs $(y_n, \mathbf{x}_n)$ are from the training data $\mathcal{D}$. However, if you think about it for a moment more, you will realize that this is just the empirical risk, and the empirical risk is a very biased estimator of the risk. This point was discussed extensively in §4.6.2. See, in particular, figure 4.17 on page 143. The point that figure made was that complex models tend to overfit, producing low empirical risk, but high risk. In essence, the problem is that we are using the data $\mathcal{D}$ twice, for conflicting objectives. First, we are using it to fit the model, producing $\hat{f}$. Second, we are using it to evalute the predictive ability of $\hat{f}$ on new observations.

So what we really need is fresh data. New data will tell us how $\hat{f}$ performs out of sample. If we had J new observations $(y_j^v, \mathbf{x}_j^v)$, then we could estimate the risk by

$$\frac{1}{J} \sum_{j=1}^{J} L(y_j^v, \hat{f}(\mathbf{x}_j^v))$$

Of course, this is not really a solution, because we don't have any new data in general. One way that statisticians try to work around this problem is to take $\mathcal{D}$ and split it into two disjoint subsets, called the **training set** and the **validation set**. The training set is used to fit $\hat{f}$ and the validation set is used to estimate the risk of $\hat{f}$. We then repeat this for all models, and choose the one with lowest estimated risk.

Since data is scarce, a more common procedure is **cross-validation**, which attempts to use the whole data set for both fitting the model and estimating the risk. To illustrate the idea, suppose that we partition the data set $\mathcal{D}$ into two subsets $\mathcal{D}_1$ and $\mathcal{D}_2$. First, we use $\mathcal{D}_1$ as the training set and $\mathcal{D}_2$ as the validation set. Next, we use $\mathcal{D}_2$ as the training set, and $\mathcal{D}_1$ as the validation set. The estimate of the risk is the average of the estimates of the risk produced in these two steps.

Of course, we could divide the data into more than two sets. The extreme is to partition $\mathcal{D}$ into N subsets, each with one element $(y_n, \mathbf{x}_n)$. This procedure is called **leave-one-out cross validation**. Letting $\mathcal{D}_{-n} := \mathcal{D} \setminus \{(y_n, \mathbf{x}_n)\}$, the data set $\mathcal{D}$ with just the n-th data point $(y_n, \mathbf{x}_n)$ omitted, the algorithm can be expressed as follows:

Algorithm 1: Leave-one-out cross-validation

for $n = 1, \ldots, N$ **do**
  fit $\hat{f}_{-n}$ using data $\mathcal{D}_{-n}$;
  set $r_n := L(y_n, \hat{f}_{-n}(\mathbf{x}_n))$;
end
return $r := \frac{1}{N} \sum_{n=1}^{N} r_n$

At each step inside the loop, we fit the model using all but the n-th data point, and then try to predict the n-th data point using the fitted model. The prediction quality is evaluated in terms of loss. Repeating this n times, we then return an estimate of the risk, using the average loss. On an intuitive level, the procedure is attractive because we are using the available data quite intensively, but still evaluating based on out-of-sample prediction.

In terms of model selection, the idea is to run each model through the cross-validation procedure, and then select the one that produces the lowest value of r, the estimated risk. Let's illustrate this idea, by considering again the ridge regression procedure used in §10.1.2. In this problem, the set of models is indexed by λ, the regularization parameter in the ridge regression. The data set $\mathcal{D}$ is the set of points shown in figures 10.3–10.6. For each λ, the fitted function $\hat{f}_\lambda$ is

$$\hat{f}_\lambda(x) = \hat{\boldsymbol{\beta}}_\lambda' \boldsymbol{\phi}(x) \quad \text{where} \quad \hat{\boldsymbol{\beta}}_\lambda := \operatorname*{argmin}_{\mathbf{b}} \sum_{n=1}^{N} \left\{ (y_n - \mathbf{b}' \boldsymbol{\phi}(x_n))^2 + \lambda \|\mathbf{b}\|^2 \right\}$$

Recall here that $\boldsymbol{\phi}(x) = (x^0, x^1, \ldots, x^d)$ with d fixed at 14, so we are fitting a polynomial of degree 14 to the data by minimizing regularized least squares error. The amount of regularization is increasing in λ. The resulting functions $\hat{f}_\lambda$ were shown

for different values of λ in figures 10.3–10.6. Intermediate values of λ produced the best fit in terms of minimizing risk (see figures 10.4 and 10.7).

In that discussion, we used the fact that we knew the underlying model to evaluate the risk. Since we could evaluate risk, we were able to determine which values of λ produced low risk (figure 10.7). In real estimation problems, risk is unobservable, and we need to choose λ on the basis of the data alone (assuming we don't have prior knowledge, as in the Bayesian case—see §10.1.3). Let's see how a data-based procedure such as cross-validation performs in terms of selecting a good value of λ.

In this experiment, for each λ in the grid

```
lambda <- exp(seq(-22, 10, length=10))
```

we perform leave-one-out cross-validation as in algorithm 1. The fit at each step within the loop is via ridge regression, omitting the n-th data point, and the resulting polynomial is used to predict y_n from x_n. The prediction error is measured by squared loss. In other words, for each λ in the grid, we use the following algorithm to estimate the risk:

Algorithm 2: Leave-one-out cross-validation for ridge regression

for $n = 1, \ldots, N$ **do**
  set $\hat{\boldsymbol{\beta}}_{\lambda,-n} := \operatorname{argmin}_{\mathbf{b}} \sum_{i \neq n} \{(y_i - \mathbf{b}' \boldsymbol{\phi}(x_i))^2 + \lambda \|\mathbf{b}\|^2\}$;
  set $r_{\lambda,n} := (y_n - \hat{\boldsymbol{\beta}}'_{\lambda,-n} \boldsymbol{\phi}(x_n))^2$;
end
return $r_\lambda := \frac{1}{N} \sum_{n=1}^{N} r_{\lambda,n}$

The value of λ producing the smallest estimated risk r_λ is around 0.015. This is in fact very close to the value that minimizes the actual risk (see figure 10.7 on page 286). The associated function $\hat{f}_\lambda$ is plotted in red in figure 10.8, and indeed the fit is excellent. In this instance, our fully automated procedure is very successful.

10.1.5 The Theory of Model Selection

Generalization error.

Minimizer is regression function.

Variance-bias interpretation?

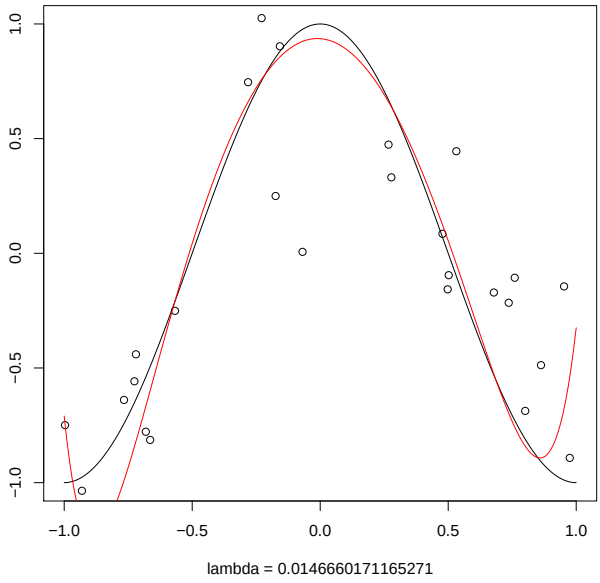


Figure 10.8: Fitted polynomial, $\lambda \approx 0.015$

Best is prior knowledge: choose the regression function.

Is there one algorithm that outperforms all others over all possible specifications of the DGP?

10.2 Method of Moments

- IV, method of moments, GMM, simulated method of moments.

Change following to vector case.

GMM is a generalization of the plug in estimator (which is a special case of ERM).

Suppose we want to estimate unknown quantity

$$\theta := \int h(s)F(ds) = \mathbb{E}\left[h(x)\right]$$

The plug in estimator is

$$\hat{\theta} := \int h(s)F_N(ds) = \frac{1}{N}\sum_{n=1}^{N} h(x_n)$$

The method of moments is a generalization of this idea. We want to estimate θ where

$$g(\theta) = \mathbb{E}\left[h(x)\right] \tag{10.8}$$

The method of moments estimator $\hat{\theta}$ is the solution to

$$g(\hat{\theta}) = \frac{1}{N}\sum_{n=1}^{N} h(x_n) \tag{10.9}$$

We can express the same thing slightly differently, replacing (10.8) with

$$\mathbb{E}\left[g(\theta) - h(x)\right] = 0 \tag{10.10}$$

and (10.9) with

$$\frac{1}{N}\sum_{n=1}^{N}[g(\hat{\theta}) - h(x_n)] = 0 \tag{10.11}$$

Generalized method of moments extends this idea further, by replacing (10.10) with the more general expression

$$\mathbb{E}\left[G(\theta, x)\right] = 0 \tag{10.12}$$

and (10.11) with the empirical counterpart

$$\frac{1}{N}\sum_{n=1}^{N} G(\hat{\theta}, x_n) = 0 \tag{10.13}$$

10.3 Breaking the Bank

- Gaussian copula

10.4 Exercises

Ex. 10.4.1. Verify the claim in (10.1) that $\text{mse}(\hat{\mathbf{b}}) = \mathbb{E}\left[\|\hat{\mathbf{b}} - \mathbb{E}\left[\hat{\mathbf{b}}\right]\|^2\right] + \|\mathbb{E}\left[\hat{\mathbf{b}}\right] - \boldsymbol{\beta}\|^2$.

Ex. 10.4.2. Derive the expectation of the ridge regression estimator $\hat{\boldsymbol{\beta}}_\lambda$. In particular, show that $\hat{\boldsymbol{\beta}}_\lambda$ is a biased estimator of $\boldsymbol{\beta}$ when $\lambda > 0$.

Ex. 10.4.3. Verify (10.7) on page 287 using (10.4) and (10.6).

Part IV

Appendices

Chapter 11

Appendix A: An R Primer

These notes use the statistical programming language R for applications and illustration of various statistical and probabilistic concepts. This chapter gives a quick introduction.

11.1 The R Language

[roadmap]

11.1.1 Why R?

It probably behooves me to say a few words about why I want you to learn R. R is not trivial to learn. And perhaps you like using Eviews/STATA/whatever because it's nice and simple: point and click, and it's done. Who needs to learn a fancy programming language? The short answer is that computers and computing are radically changing econometrics and statistics, and opening up a world of new possibilities. Let's leave point and click regression for the high-school kiddies. Real programming skills will set you free.

Here's an example. Let's say that you are interested in development and microfinance. A new development in this area is the arrival of Kiva, a nonprofit organization that allows people to make small loans over the Internet, mostly to local entrepreneurs in developing countries. Kiva has been good enough to make almost all its data freely available for download. This is done through a web-based API

which returns HTTP web queries in XML or JSON (Javascript Object Notation—a text based data exchange format suitable for parsing by computers). If you want to get your hands on Kiva's data and manipulate it, you need to know a bit about computing.

The Kiva example illustrates the need for programming skills when obtaining and parsing data. The other side of the story is what gets done to the data once its stored on our computer. Once again, good programming skills are what give you the freedom to do what you want. Computers have opened up a whole new world of statistical techniques, and not all can or will be conveniently canned and packaged in a point and click interface. You will need to tell your computer what to do with a sequence of written text commands—a program.

R is a good way to jump into more computer intensive statistical techniques. It's programming language is simple and robust enough to learn easily, and descriptive enough to do almost anything you want. Either way, in this course you'll be exposed to R. Whether you're converted or not, R is well worth a look.

(Finally, if you're already a programmer, you might be wondering why I've chosen R over other programming languages popular in econometrics, such as GAUSS or Matlab. The short answer is that R is certainly as good as any other programming language for statistics, and it's also free. Moreover, it's rapidly becoming the default language of statistics within statistics departments around the world. That said, the main aim here is to teach you how to write programs for statistics and econometrics. Almost all the skills you learn in this course are portable to other languages. The important this is that you *learn how to code up your ideas.*)

11.1.2 Introducing R

What is R? The R homepage (`http://www.r-project.org/`) introduces R as a language and environment for statistical computing and graphics, designed as an open source implementation of S (the latter being a statistics language developed at Bell Laboratories. The fact that R is open source means that R is

- free as in "free beer"—it costs nothing to download
- free as in "free speech"—owned by the community, for the community

Despite being free, R is every bit as good as commercial statistical packages.

Of course R is not perfect. Someone once said that the best thing about R is that it was written by statisticians...and the worst thing about R is that it was written by statisticians. That's pretty accurate: It's a great environment to jump into and do serious analysis, but the language is a little quirky relative to some of the more elegant modern programming languages (Python, etc.). It also has a steep learning curve relative to point-and-click style environments such as Eviews and STATA.

On the other hand, R contains a complete, well-structured programming language, which allows users to tackle arbitrarily sophisticated problems. It has excellent graphics and visual presentation, combining sensible defaults with good user control. It implements a vast range of statistical functions, tests and graphical techniques, including linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, and so on. More can be found by perusing the available packages on `http://cran.r-project.org/`.

11.1.3 Getting Started

R can be downloaded from `http://www.r-project.org/`. Typically, the latest version and add on packages will be found on CRAN—the Comprehensive R Archive Network. CRAN can be accessed from the R homepage, where you will be directed to a local mirror of the archive.

A quick Google search will provide lots of information on getting R up and running. (Try searching YouTube too—at the time of writing there are some helpful videos.)

I'll assume that installation has gone fine, and you have just fired up R for the first time. You should now be greeted with information on the version number, copyright, etc., followed by a prompt like so:

```
>
```

The first thing you need to know is how to quit:

```
> q()
```

If you're prompted to save your workspace, say no for now.

The second thing you need to know is how to get help. There's an interactive help system that can be accessed as follows:

```
> help(plot)
```

or

```
> ?plot
```

You'll be presented with a manual page for the `plot` function. On my computer (which is running Linux—might be different for Windows or Mac), pressing "q" exits the manual page, and returns you to the prompt. If `help` doesn't turn anything up, you can try `help.search`:

```
> help.search("plotting")
```

Overall, the help system is fairly good, and I consult it often. However, it can be technical, so general Internet searches may be useful too.

Now we've covered help and quitting, let's learn a bit about the command interface. We are not in the land of point-and-click here, which might make some people nervous. But the command line is highly efficient once you get used to it. Here are a few tips.

First, try typing

```
> plo
```

and press the tab key twice. You'll be presented with a list of possible expansions. This is useful if you're not sure what related commands are available. Also, if the command you've begun is uniquely identified by what you've typed so far, it will be expanded by the tab key to the full command.

Once we start working with our own variables, the same expansion technique will work for them. This is particularly helpful with long names. If you have a variable called

```
interwar.consumption.in.southern.alabama
```

then typing the first few letters and tabbing out will save a lot of time.

Another useful feature of the command line is the *command history*, which is accessed via the up and down arrow keys. For example, to recall a previously entered command from the current session, press the up arrow key until it reappears. (Try it and you'll see.) You can now press enter to re-run the command as is, or edit the command and then run (use the left and right arrow keys).

11.2 Variables and Vectors

Now we have some feel for the command line, let's see how we can put R to good use. The simplest way to use R is as a fancy calculator. For example, to add 12 and 23 we type

```
> 12 + 23
```

and get the response

```
[1] 35
```

(Just ignore the `[1]` for now.) To multiply we type

```
> 12 * 23
```

To raise 12 to the power of 23 we type

```
> 12^23
```

and so on.

Parentheses are used in a natural way. For example:

```
> 12 * (2 + 10)
[1] 144
```

This indicates to the R interpreter that it should first add 2 and 10, and then multiply the result by 12.

So far so good, but to do anything more interesting we'll need to use *variables*

11.2.1 Variables

At the heart of any programming language is the concept of a variable. A **variable** is a name (often a symbol such as x or y) associated with a value. For example,

```
> x <- 5
```

binds the *name* `x` to the *value* 5. The value 5 is an object stored somewhere in your computers' memory, and the name `x` is associated to that little patch of memory. (The equal sign = can be used as a substitute for the assignment operator `<-`, but the latter is more common.) Figure 11.1 illustrates the idea.

Now, when you use the symbol `x`, R retrieves that value and uses it in place of `x`:

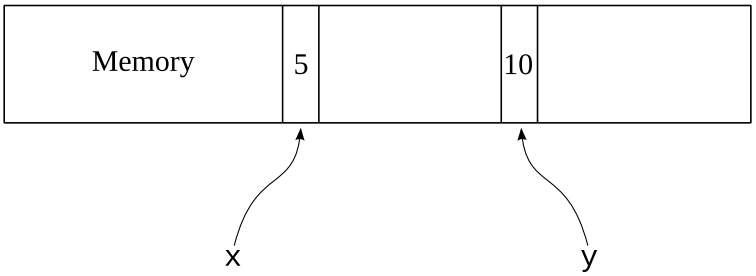


Figure 11.1: Variables stored in memory

```
> x <- 5
> x
[1] 5
> x + 3
[1] 8
> x
[1] 5
> y <- x + x
> y
[1] 10
```

Notice how typing the variable name by itself causes R to return the value of the variable. Notice also that assignment works from right to left: in the statement `y <- x + x`, the R interpreter first evaluates the expression `x + x` and then binds the name `y` to the result. Understanding this is important for interpreting commands such as

```
> x <- x * 2
```

First the r.h.s. is evaluated to obtain 10, and then the name `x` is bound to this number. Hence, the value of `x` is now 10.

As before, we are using the `*` symbol to multiply, as in `x * 2`. It cannot be omitted:

```
> x <- x 2
Error: unexpected numeric constant in "x <- x 2"
> x <- x2
Error: object 'x2' not found
```

Exponential and division are as follows:

```
> x^3
[1] 1000
> x / 3
[1] 3.333333
```

We can use `ls` to see what variables we've created so far, and `rm` if we decide that one of them should be deleted:

```
> ls()
[1] "x" "y"
> rm(x)
> ls()
[1] "y"
```

Incidentally, up until now we've used simple names for our variables, such as x and y. More complex names can be used to help us remember what our variables stand for:

```
> number_of_observations <- 200
```

or

```
> number.of.observations <- 200
```

Some rules to remember: `x1` is a legitimate variable name, but `1x` is not (variables can't start with numbers). Also, R is case sensitive (`a` and `A` are distinct names, etc.)

11.2.2 Vectors

So far, the variables we've created have been scalar-valued. Now let's create some vectors. A **vector** is an array of values such as numbers. (Actually, there are other possibilities besides numbers, as we'll see soon enough.) Vectors are important in R. If fact, the "scalar" variables we've created so far are stored internally as vectors of length 1.

```
> a <- c(2, 5, 7.3, 0, -1)
> a
[1]  2.0  5.0  7.3  0.0 -1.0
```

Here we've created a vector `a` using the `c` function, which *concatenates* the numbers 2, 5, 7.3, 0, and -1 into a vector. The resulting vector is of length 5, and we can verify this as follows:

```
> length(a)
[1] 5
```

The c function can also concatenate vectors:

```
> a <- c(2, 4)
> b <- c(6, 8)
> a_and_b <- c(a, b)
> a_and_b
[1] 2 4 6 8
```

On thing we often want to do is create vectors of **regular sequences**. Here's one way:

```
> b <- 1:5
> b
[1] 1 2 3 4 5
> b <- 5:1
> b
[1] 5 4 3 2 1
```

If we need to be a bit more flexible, we can use the function seq:

```
> b <- seq(-1, 1, length=5)
> b
[1] -1.0 -0.5  0.0  0.5  1.0
```

Try help(seq) to learn more. To generate a constant array, try rep:

```
> z <- rep(0, 5)
> z
[1] 0 0 0 0 0
```

We can also generate vectors of random variables. For example

```
> x <- rnorm(3)       # 3 draws from N(0,1)
> y <- rlnorm(30)     # 30 lognormals
> z <- runif(300)     # 300 uniforms on [0,1]
```

We'll learn more about this later on.

11.2.3 Indices

Using `[k]` after the name of a vector references the k-th element:

```
> a <- c(2, 5, 7.3, 0, -1)
> a[3]
[1] 7.3
> a[3] <- 100
> a
[1]   2   5 100   0  -1
```

Entering a negative index returns all but the indicated value:

```
> a <- c(2, 5, 7.3, 0, -1)
> a[-1]
[1]  5.0  7.3  0.0 -1.0
```

There are other ways to extract several elements at once. The most common is by putting a vector of integers inside the square brackets like so:

```
> a[1:4]
[1] 2.0 5.0 7.3 0.0
```

11.2.4 Vector Operations

Now let's look at some operations that we can perform on vectors. Most of these can be performed on scalar variables as well, but remember that scalar variables are just vectors of length 1 in R, so there's no need to distinguish.

To begin, consider the vector

```
> x <- 1:5     # Same as x <- c(1, 2, 3, 4, 5)
```

We can obtain the sum of all elements via

```
> sum(x)
[1] 15
```

and the minimal and maximal values via

```
> min(x)
[1] 1
> max(x)
[1] 5
```

To get the average of the values we use

```
> mean(x)
[1] 3
```

while the median is obtained by

```
> median(x)
[1] 3
```

The sample variance and standard deviation are obtained by `var(x)` and `sd(x)` respectively.

So far we've looked at functions that take a vector and return a single number. There are many others that transform the vector in question into a new vector of equal length. For example, the `log` function returns the natural log of each element of the vector:

```
> log(x)
[1] 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379
```

This is a common pattern: The function acts *elementwise* (i.e., element by element) on its argument. Here are some more examples, returning the exponential and sine of x respectively:

```
> exp(x)
[1]   2.718282   7.389056  20.085537  54.598150 148.413159
> sin(x)
[1]  0.8414710  0.9092974  0.1411200 -0.7568025 -0.9589243
```

Naturally, we can perform two or more operations at once:

```
> abs(cos(x))
[1] 0.5403023 0.4161468 0.9899925 0.6536436 0.2836622
> round(sqrt(x), 1)
[1] 1.0 1.4 1.7 2.0 2.2
```

Now let's look at arithmetic operations. In general, standard arithmetic operations like addition and multiplication are performed elementwise. For example

```
> a <- 1:4
> b <- 5:8
> a
[1] 1 2 3 4
> b
[1] 5 6 7 8
> a + b
[1]  6  8 10 12
> a * b
[1]  5 12 21 32
```

How about if we want to multiply each element of a by 2? One way would be to enter

```
> a * rep(2, 4)    # 4 because length(a) = 4
[1] 2 4 6 8
```

or, more generally,

```
> a * rep(2, length(a))
[1] 2 4 6 8
```

However, there's a nicer way to do it:

```
> a * 2
[1] 2 4 6 8
```

The same principle works for addition, division and so on.

11.3 Graphics

R has strong graphics capabilities when it comes to producing statistical figures. There are many different ways to create such figures. A common one is the `plot()` command. Here's an example

```
> x <- seq(-3, 3, length=200)
> y <- cos(x)
> plot(x, y)
```

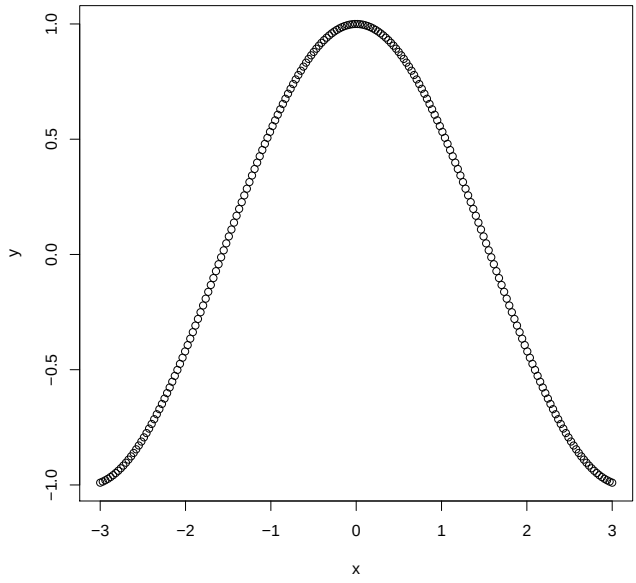


Figure 11.2: Illustration of the `plot` command

This produces a *scatter plot* of x and y, as in figure 11.2. If we prefer blue we use the command `plot(x, y, col="blue")` instead, producing figure 11.3. To produce a blue line, try `plot(x, y, col="blue", type="l")`.

If you give only one vector to `plot()` it will be interpreted as a time series. For example, try

```
> x <- rnorm(40)
> plot(x, type="l")
```

11.3.1 High-Level Graphical Commands

In R, graphical commands are either "high-level" or "low-level." The function `plot()` is an example of a high-level command. Low-level commands do things like add points, lines and text to a plotting area. High-level commands are built on top of low-level commands, offering a convenient interface to the creation of common statistical figures.

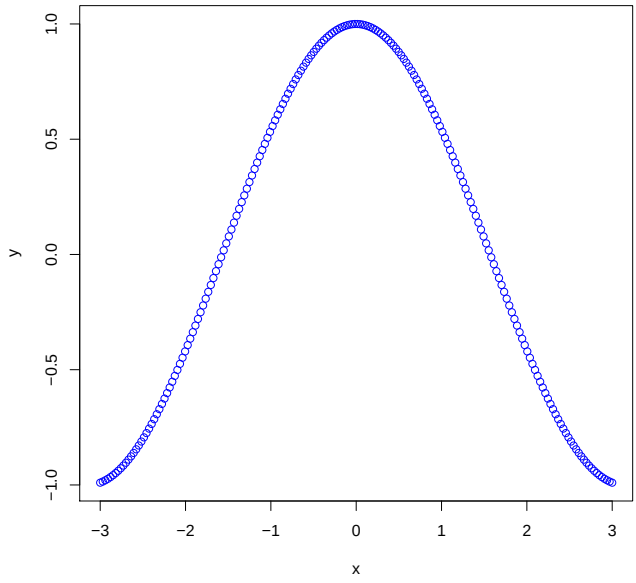


Figure 11.3: Illustration of `plot` again

The important thing to remember about creating figures in R is that the commands for a given figure should contain one and only one high-level command, followed optionally by multiple low-level commands.

Let's discuss some more examples of high-level commands. Histograms are a really common way to investigate a univariate data sample. To produce a histogram we use `hist`:

```
> x <- rlnorm(100)  # lognormal density
> hist(x)
```

The output is shown in figure 11.4.

Figure 11.5 is a bit fancier. The code for producing it is

```
> x <- rnorm(1000)
> hist(x, breaks=100, col="midnightblue")
```

If you want a background color too, then use `par`, which sets **par**ameters for the figure. See the on-line help for details (type "`?par`" without the quotes).

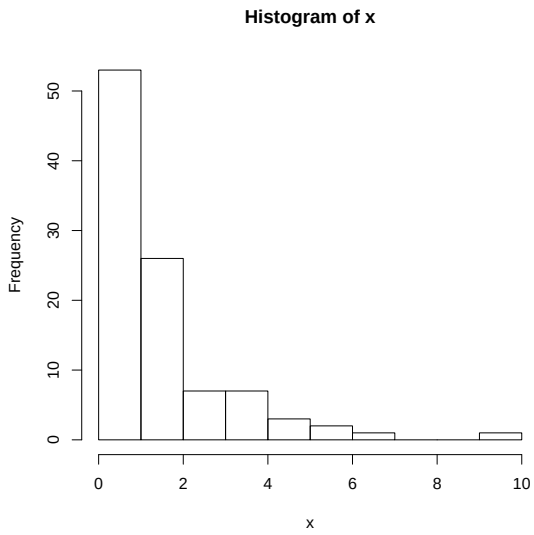


Figure 11.4: Illustration of the `hist` command

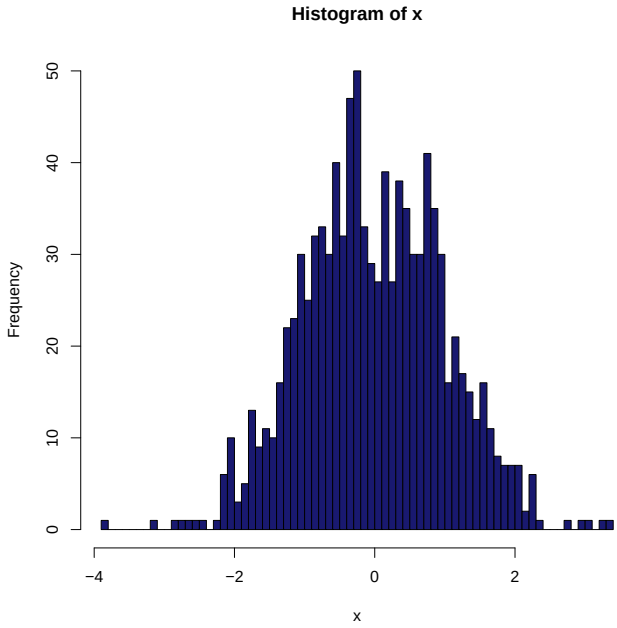


Figure 11.5: Illustration of `hist` again

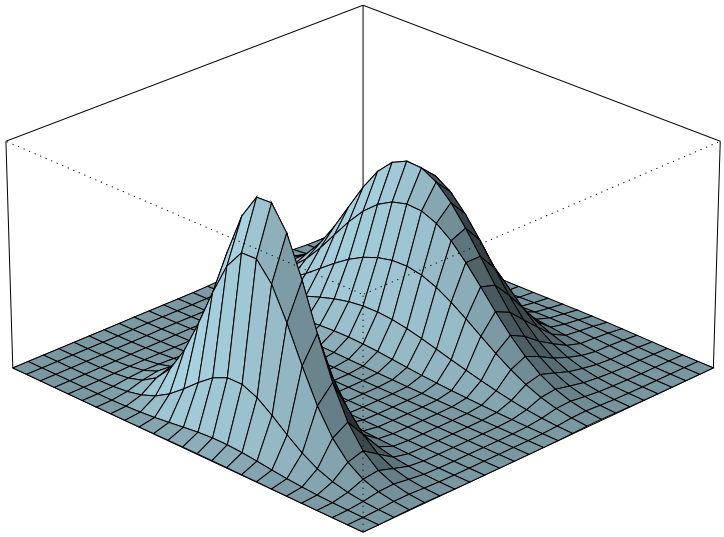


Figure 11.6: Plotting in 3D

There are several other high-level plotting functions for presenting statistical data. For example, `barplot` produces bar plots, `boxplot` produces box plots, `pie` produces pie charts, and so on. We will meet some of them as we go along.

Still more high-level graphics functions are available if you want to dig deeper. For example, `contour` produces contour maps of 3D data, while `persp` produces 3D graphs. A 3D graph produced using `persp` is shown in figure 11.6. Details are omitted.

11.3.2 Low-Level Graphical Commands

High-level graphics commands are built on top of low-level commands. Low-level commands are also available to the user, and can be utilized to *add additional components to figures that were initially produced by a high-level command.* Examples include `points`, which adds points, `lines`, which adds lines described by x and y coordinates, `text`, which adds text, `abline`, which adds straight lines, `polygon`, which adds polygons (useful for filling regions), `arrows`, which adds arrows, and `legend`, which adds a legend.

The following code gives an example.

```
> x <- seq(-2 * pi, 2 * pi, length=200)
```

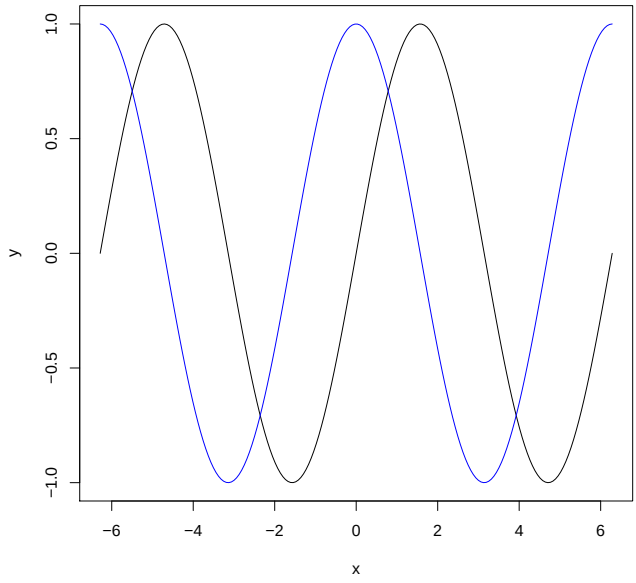


Figure 11.7: Adding points and lines

```
> y <- sin(x)
> z <- cos(x)
> plot(x, y, type="l")
> lines(x, z, col="blue")
```

First we call the high-level graphics command `plot`, and then the low-level command `lines`. The resulting output is shown in figure 11.7. You can experiment to see what happens if you issues these commands in the reverse order, or replace the word `lines` with `plot`.

11.3.3 Getting Hardcopy

R provides range of *graphics drivers* to produce hardcopy. For example, to produce a histogram as a PDF, try

```
> x <- rlnorm(100)
> pdf("foo.pdf")    # Write to a file called foo.pdf
> hist(x)
```

```
> dev.off()
```

The command `dev.off` signals to the graphics driver that you have finished adding components to foo.pdf. R now writes the PDF output to a file called foo.pdf in your current working directory. To see what your current working directory is, type `getwd`.

11.4 Data Types

[roadmap]

11.4.1 Basic Data Types

Like most programming languages, R can work with and perform operations on various kinds of data, such as numbers, text and Boolean values (see below). We can investigate the type of a variable by using either the `mode` or `class` function:

```
> b <- c(3, 4)
> class(b)
[1] "numeric"
> mode(b)
[1] "numeric"
```

Here both `mode` and `class` return `"numeric"`, which indicates that the elements of the vector are stored as **floating point numbers**. Floating point numbers (or floats) are a computer's approximation to real numbers—see §13.1 for a discussion of the latter.

Although `mode` and `class` returned the same answer in the previous example, we will see that this is not always the case. In general, `mode` refers to the primitive data type, whereas `class` is more specific. We'll talk more about classes below.

Another common data type is **strings**. Strings are pieces of text—any of the alphanumeric characters and other symbols on your keyboard. In R, strings have mode "character."

```
> x <- "foobar"  # Bind name x to string "foobar"
> mode(x)
[1] "character"
```

We can concatenate strings using the `paste` function:

```
> paste("foo", "bar")
[1] "foo bar"
```

By default, the two strings are separated by a space, but we can eliminate the space as follows:

```
> paste("foo", "bar", sep="")  # Separate by empty string
[1] "foobar"
```

Here's a more useful example of `paste`:

```
> paste("Today is ", date())
[1] "Today is  Mon Feb 28 11:22:11 2011"
```

In the code above, you will have noticed that, when we work with strings, we usually write them between quote marks. Why do we do this? The reason is that if we don't add quote marks, then R interprets the sequence of letters as the name of a variable. For example,

```
> x <- foobar
Error: object 'foobar' not found
```

Here the interpreter looks for the variable `foobar`, and, not finding it, issues an error message.

Here's another example. In §11.3, the command

```
> hist(x, breaks=100, col="midnightblue")
```

was used to produce figure 11.5. Here `"midnightblue"` is a string, that's passed as an argument to the function `hist`. Why can't we just type

```
> hist(x, breaks=100, col=midnightblue)  # Wrong!
```

instead? Because then R would think that `midnightblue` is a variable, and look for it in the current environment.

There are a few other basic data types besides numeric and character. One is **Boolean**, which holds the logical values TRUE and FALSE. As with other types, Boolean values can be stored in vectors. For example:

```
> x <- TRUE
> mode(x)
[1] "logical"
> x <- c(TRUE, TRUE, FALSE)
> mode(x)
[1] "logical"
```

As we'll soon see, Boolean vectors are very important in R. To save a bit of typing, you can use the abbreviations `T` and `F`:

```
> x <- c(T, T, F)
> x
[1]  TRUE  TRUE FALSE
```

One thing to remember about vectors is that in any one vector, all data must be of the same type. For example, suppose that we try to make a vector with two different modes:

```
> x <- c(1.1, "foobar")
> mode(x)
[1] "character"
> x
[1] "1.1"    "foobar"
```

We see that the numeric value `1.1` has been converted to a character string, in order to ensure all elements have the same type.

At this point, an obvious question is: What if we want to store several different data types as a single object? For example, let's say we have data on employees at a given firm, and we want to store their surname and salary together. How should we accomplish this?

In R, one way to do it is to use a **list**. A list is like a vector, except that it's elements have "names" that can be used to access them, and, in addition, there's no restriction on the data types of the elements. Here's an example, where we create a list using the `list` function:

```
> employee1 <- list(surname="Smith", salary=50000)
```

Here we've given the names `surname` and `salary` to the two elements of the list. The elements can now be accessed via these names using the "$" symbol:

```
> employee1$surname
[1] "Smith"
> employee1$salary
[1] 50000
```

If you're dealing with a list and you can't remember the names of the elements, you can extract them with `names`:

```
> names(employee1)
[1] "surname" "salary"
```

Although we won't have much cause to use the `list` function during this course, we will still be creating quite a lot of lists. The reason is that when R functions need to return a whole lot of different information, they do so using a list. For example, whenever we run a regression using the standard linear regression function `lm`, this function will return a list that contains information about the estimated coefficients, the residuals, and so on.

One final point on basic data types is that sometimes we need to test the type of a variable to make sure it will work in a given operation. If not, we may need to change its type. For this purpose, R provides a collection of `is.` and `as.` functions. Here's an example:

```
> x <- c("100", "200")
> is.numeric(x)
[1] FALSE
> sum(x)
Error in sum(x) : invalid type
> x <- as.numeric(x)  # Convert x to numeric
> is.numeric(x)
[1] TRUE
> sum(x)
[1] 300
```

11.4.2 Data Frames

In statistical programming we work a lot with data sets, reading in data, storing it, selecting subsets, making changes, and so on. In R, data is usually stored in **data frames**. Data frames are special kinds of lists that are used to store "columns"

of related data. Typically, each column corresponds to observations on a particular variable. Data frames are a bit like matrices (which we'll meet later), but the columns can hold different data types (numeric, character, etc).

Let's start with a simple and light-hearted example. On the financial blog "Alphaville," Tracy Alloway noted the correlation between speaking fees paid by major financial firms to economist Lawrence Summers (Director of the White House National Economic Council) and the relative stability of their share price during the GFC. The firms in question were Goldman Sachs, Lehman Bros, Citigroup and JP Morgan, who paid Summers speaking fees of (roughly) $140,000, $70,000, $50,000 and $70,000 respectively. Over the year to April 2009, their share prices fell by 35%, 100%, 89% and 34% respectively.

Let's record this in a data frame. We can do this in different ways. Perhaps the easiest is to first put the data in vectors

```
> fee <- c(140000, 70000, 50000, 70000)
> price <- c(-35, -100, -89, -34)
```

and then build a data frame:

```
> summers <- data.frame(fee, price)
```

As discussed above, a data frame is a list:

```
> mode(summers)
[1] "list"
```

But it's not just a list, it's a special kind of list called a data frame:

```
> class(summers)
[1] "data.frame"
```

We'll talk more about the `class` function in §11.4.3 below. For now, let's have a look at our data frame:

```
> summers
    fee price
1 140000   -35
2  70000  -100
3  50000   -89
4  70000   -34
```

We see that R has numbered the rows for us, and used the variable names as names for the columns. We can produce more descriptive column names as follows:

```
> summers <- data.frame(Speak.Fee=fee, Price.Change=price)
> summers
        Speak.Fee     Price.Change
1        140000         -35
2         70000        -100
3         50000         -89
4         70000         -34
```

Since `summers` is a kind of list, with columns of the data frame corresponding to elements of the list, the columns can be accessed using their names:

```
> summers$Speak.Fee
[1] 140000  70000  50000  70000
> summers$Price.Change
[1]  -35 -100  -89  -34
```

(Are you remembering to use the TAB key to expand these long names?) On the other hand, the column names by themselves won't work:

```
> Speak.Fee
Error: object "Speak.Fee" not found
```

This is because these variables are not part of the current workspace, but rather they are "hidden" inside the data frame. This *data encapsulation* is deliberate, and helps us things organized. It's the same idea with directories (folders) on computers: Files are grouped in different directories to keep the organized by topic. Just as we can have the same file name in different directories, we can have the same column name in different data frames.

Returning to our data frame, we can also access the data in the data frame using "matrix style" index notation. For example,

```
> summers[1, 2]  # First row, second column
[1] -35
> summers[2,]   # All of second row
    Speak.Fee Price.Change
2     70000          -100
> summers[,2]   # All of second column
[1]  -35 -100  -89  -34
```

One thing we could do to make the data frame more descriptive is to replace the row numbers with the names of the banks. This is done through the `row.names` function, which acts on data frames. Let's see how this works:

```
> row.names(summers)
[1] "1" "2" "3" "4"
> firm <- c("Goldman", "Lehman", "Citi", "JP Morgan")
> row.names(summers) <- firm
> row.names(summers)
[1] "Goldman"   "Lehman"    "Citi"      "JP Morgan"
```

Now the `summers` data frame looks as follows:

```
> summers
             Speak.Fee        Price.Change
Goldman          140000                 -35
Lehman            70000                -100
Citi              50000                 -89
JP Morgan         70000                 -34
```

One of the nice things about data frames is that many R functions know how to interact with them directly. For example, if we enter

```
> plot(summers)
```

we immediately get a scatter plot. If we use the function `summary` we get a summary of the data:

```
> summary(summers)
    Speak.Fee     Price.Change
Min.   : 50000   Min.   :-100.00
1st Qu.: 65000   1st Qu.: -91.75
Median : 70000   Median : -62.00
Mean   : 82500   Mean   : -64.50
3rd Qu.: 87500   3rd Qu.: -34.75
Max.   :140000   Max.   : -34.00
```

Here, we plot, run a linear regression and then add the line of best fit to the plot:

```
> plot(summers)
> reg <- lm(Price.Change ~ Speak.Fee, data=summers)
> abline(reg)
```

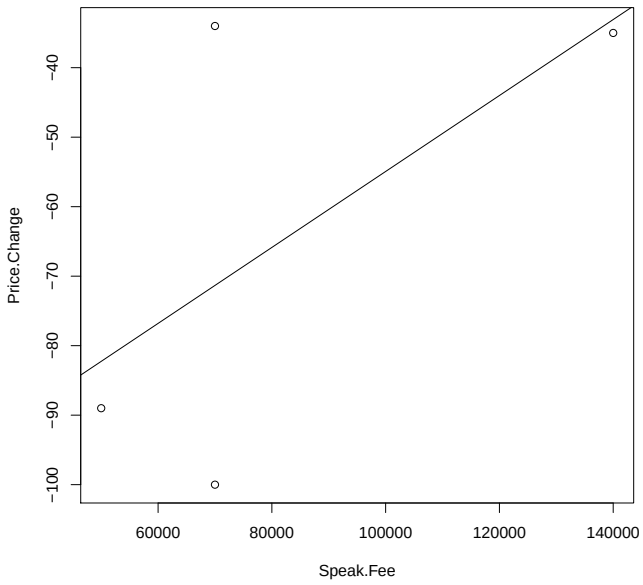


Figure 11.8: Scatter plot and line of best fit

The resulting plot in figure 11.8 shows positive correlation. (Again, this is lighthearted, don't read too much into this.) More discussion of univariate linear regression is given in §11.5.

11.4.3 A Note on Classes

In the proceeding example, we created a data frame called `summers` and ran a regression with the code

```
> reg <- lm(Price.Change ~ Speak.Fee, data=summers)
```

If you now type

```
> summary(reg)
```

you'll be presented with a nice table giving you estimated coefficients and other summary statistics. We'll talk more about this output in §11.5, but for now I'd like you to notice that the function `summary` was used previously in the code

```
> summary(summers)
```

which gave a basic description of the data in the data frame `summers`. What's interesting here is that we are using the same function `summary` with two very different arguments, and each time R gives an appropriate result. It's useful to know how this works.

On one hand, we can check that both the data frame `summers` and the object `reg` returned by the regression are lists:

```
> mode(reg)
[1] "list"
> mode(summers)
[1] "list"
```

However, the `summary` function needs to distinguish between these objects, so that it knows what kind of information to return. The way this is accomplished is that the two objects are given addition type information beyond their mode. This second, more specific, type information is called the objects **class**:

```
> class(reg)
[1] "lm"
> class(summers)
[1] "data.frame"
```

The function `summary`, when passed an object such as `reg`, first investigates its class. Once it knows the class of the object, it knows what action to perform. Functions like `summary`, that act differently on different objects according to their class, are called **generic functions**.

11.5 Simple Regressions in R

Let's have a quick look at the basic technique for running regressions in R.

11.5.1 The `lm` Function

To set up an example, let's generate some data:

```
> N <- 25
> x <- seq(0, 1, length=N)
> y <- 5 + 10 * x + rnorm(N)
```

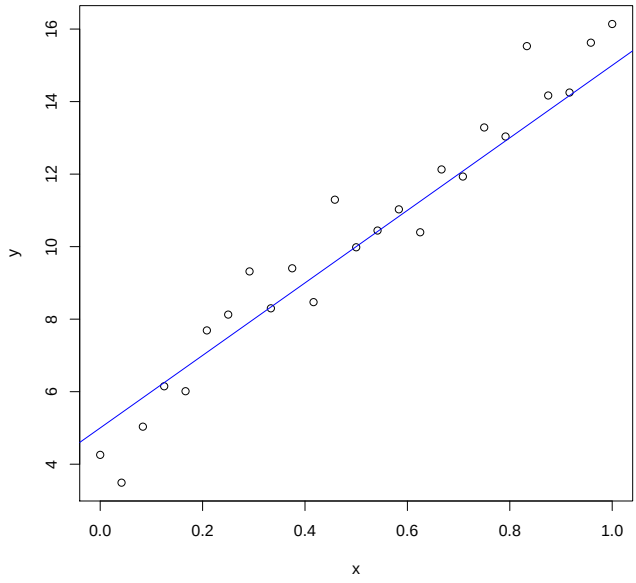


Figure 11.9: The data

The data is plotted in figure 11.9, along with the function $y = 5 + 10x$ (in blue). We can regress y on x as follows:

```
> results <- lm(y ~ x)
> class(results)
[1] "lm"
```

The function `lm` is the standard function for linear, least squares regression. It returns an object of class `lm`, which, as discussed in §11.4.3, is a kind of list. In this example, we have bound the name `results` to this list.

The list object returned by a called to `lm` includes as its elements various various other vectors and lists containing the information produced by the regression. For example, the coefficients are an element of the list:

```
> results$coefficients
(Intercept)           x
   4.491593   11.456543
```

To see the full list, use `names(results)`.

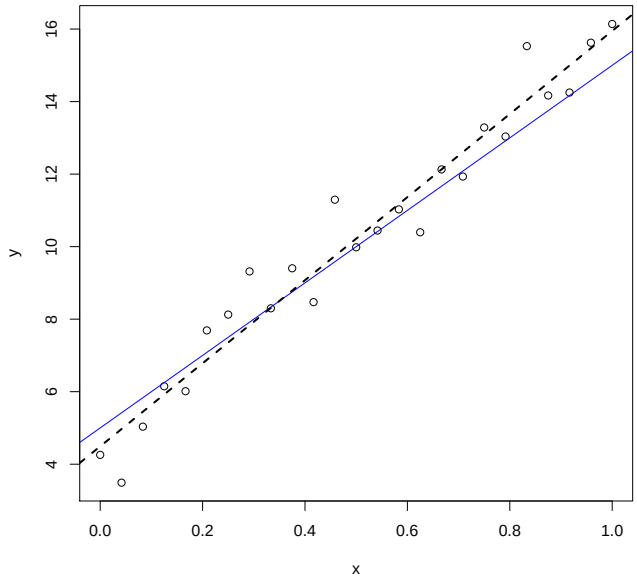


Figure 11.10: The regression line

There are in fact various "extractor" functions for obtaining coefficients, fitted values and so on from `results`. Try, for example,

```
> coef(results)
> fitted.values(results)
> residuals(results)
```

On top of these extractor functions that obtain low level information about the output of the regression, several generic functions perform further calculations based on the results to provide information about the regression. We learned about `summary` in §11.4.3. Another such function is `abline`, which adds the regression line to a plot. In figure 11.10 we've added a regression line via the call

```
> abline(results, lty=2, lw=2)  # Dashed, double thickness
```

11.5.2 Formulas

Now let's have another look at the syntax of our call to `lm`. The argument we used was `y ~ x`. This argument is called a **formula**, which is a special syntax used for

specifying statistical models in R. Let's look at some further examples. First, if you happen to have another predictor z, then you can regress y on x and z via the call

```
> lm(y ~ x + z)
```

What's important to remember is that the "+" symbol does *not* represent addition per se—it is part of the formula, and indicates that y should be regressed on x *and* z.

In regressions such as the one above, the intercept is included by default. We can also include it explicitly via the call

```
> lm(y ~ 1 + x + z)
```

but this is exactly equivalent to the last call. To remove the intercept we can use the formula

```
> lm(y ~ 0 + x + z)  # Or lm(y ~ x + z - 1)
```

Finally, we sometimes want R to evaluate the expressions in our formulas as ordinary arithmetic operations. For example, suppose that we want to regress y on the square of x, rather than x itself. This can be achieved by the call

```
> lm(y ~ I(x^2))
```

The function I indicates that the operation x^2 should be treated as an arithmetic operation, rather than be regarded as part of the formula.

Chapter 12

Appendix B: More R Techniques

Now let's dig a bit deeper into R. In this chapter, we'll learn about working with files, and also how to code up simple programs.

12.1 Input and Output

Writing programs and getting data in and out of R involves sending information to the screen and working with files. Let's run through the basics of how this is done.

12.1.1 The Current Working Directory

Often, when you are working with R, you will have a directory (folder) that contains files related to the project you are working on—data files, script files containing R programs, figures you have created, and so on. On the other hand, when you start R, an internal variable is initialized that stores the **current working directory**. This is the directory where R looks for any files you reference at the command line, and writes files that you create. If you are using something other than a Linux machine, these two directories will not be automatically matched. This can cause confusion.

On part of this confusion is that some operating systems use the backslash symbol \ to separate directories in a path, while others use the forward slash /. Regardless of your operating system, however, R uses /. For example, if you are using a Windows machine and have a directory called `"C:\Foobar"` on your computer, then R will

refer to this directory as "C:/Foobar". Please be aware of this in what follows, and convert forward slashes to backslashes as necessary.

Let's look at an example. I've created a file on my computer called "test.txt". It is a text file that contains the single line

```
10 9 8 7 6 5 4 3 2 1
```

The full path to the file on my Linux machine is

```
/home/john/emet_project/test.txt
```

If I start R in "/home/john" then this will be my current working directory. This can be checked with the command getwd:

```
> getwd()
[1] "/home/john"
```

Now I try to read the data in "test.txt" into a vector with the scan function:

```
> x <- scan("test.txt")
```

but receive an error message telling me the file cannot be found. The problem is that "test.txt" is not in the current working directory.

There are a few ways I can rectify this problem, including shifting the file into the current working directory, but the best solution is to change the current working directory to where the file is. This can be accomplished as follows:

```
> setwd("/home/john/emet_project/")
```

and the call to scan now succeeds:

```
> x <- scan("test.txt")
Read 10 items
> x
 [1] 10  9  8  7  6  5  4  3  2  1
```

If you want to see the contents of your current working directory, you can do so with the dir function:

```
> dir()
[1] "test.txt"
```

If I next create a figure and save it, the file will be saved in the current working directory:

```
> pdf("foo.pdf")
> plot(x)
> dev.off()
null device
          1
> dir()
[1] "foo.pdf"  "test.txt"
```

Two points to finish this section: First, it's tedious to have to manually set the current working directory every time you start R. There are work-arounds for all operating systems. Googling will tell you what you need to know. Second, there may be times when you want to read data from a file that is located somewhere on your computer, and it's easiest to locate that file by point and click. In that case, try the `file.choose` function, as in the following example:

```
x <- scan(file=file.choose())
```

On Windows machines, this should open up a dialog box which will allow you to select the file.

12.1.2 Reading Data in

We have already me the `scan` function, which is a low-level I/O function for reading data from files. The `scan` function returns either a vector or a list, depending on the contents of the file. When working with data sets, a much more common function to use is the `read.table` function, which attempts to read the data from a file into a data frame.

Let's look at an example. I've created a text file on my computer called `"testdf.txt"` with two columns of data that looks as follows:

```
X  Y
1 10
2 20
3 30
```

This file is in my current working directory, and I can read it into a data frame as follows:

```
> df <- read.table("testdf.txt", header=TRUE)
> df
  X  Y
1 1 10
2 2 20
3 3 30
> class(df)
[1] "data.frame"
```

Here I've set `header=TRUE` because the first row contains column names.

The `read.table` function has many options, which you can investigate `?read.table`. You can skip lines of crud at the start of your file using `skip`, work with comma separated values via `sep` and so on. (R can also handle many foreign data file formats. For example, R can read and write data files in the formats used by Excel, STATA, SPSS and SAS. Please look up the documentation as required.)

Another thing you can do with `read.table` and other input functions is read data directly from the Internet by giving the URL instead of a filename:

```
> read.table("http://johnstachurski.net/emet/testdf.txt",
   header=TRUE)
```

On your home installation this command should work fine. At work or at university it may not, because many office and university computers are behind firewalls, where HTTP traffic is routed through a proxy server. If R doesn't know about the proxy, then you can set it as follows:

```
Sys.setenv(http_proxy="http://user:pass@proxy:8080")
```

Here `user` is your username, `pass` is your password, and `proxy` is your proxy server. (One place to find those details is in your browser, which must be aware of the proxy server if it's working. However, you will not be able to see your password.)

If the above command still doesn't help, you can always save the file in question to your local hard disk with your browser, and then proceed as before.

12.1.3 Other I/O Functions

There are several low level functions for input and output. One is `scan`, which we met in §12.1.1. This function is quite flexible, but the most common use is the one we mentioned: Reading a sequence of numbers from a text file and converting it to a vector. The vector can then be converted into a matrix if desired—we'll talk about this process later.

If no file name is given, then `scan` reads from the screen:

```
> x <- scan()
1: 1 12 3 55 128    # Me typing numbers in
6:                  # And hitting return a second time
Read 5 items
> x
[1]   1  12   3  55 128
```

Another way to get information in is via the `readline` function. For example, if other people are going to use your program, you can get information from them along the following lines:

```
> x <- readline("Enter the value of x: ")
```

Note that the result will be a string, so you may need to call a function such as `as.numeric` if you want to convert x to a numeric value.

The last few functions we have discussed pertain to input. Let's talk briefly about output. To write a whole data frame, try something along the lines of

```
> write.table(summers, "summers.txt")
```

There are many options to this command, and I leave you to investigate.

A lower level function for data output is `cat`. One important use for this function is as a substitute for the `print` function, with more detailed control. For example,

```
> x <- 3
> cat("The value of x is", x, "\n")
The value of x is 3
```

The final `"\n"` tells R to end with a new line. (Try without if you're not sure what I mean.) By specifying a file name, we can also write this information to the hard disk:

```
> cat("The value of x is", x, "\n", file="test.txt")
```

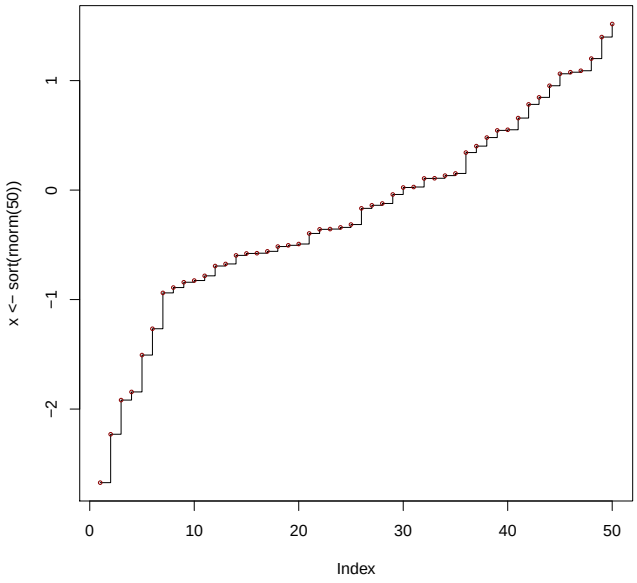


Figure 12.1: Step function

12.1.4 Scripting

Another kind of data you will want to read into R is sequences of commands, or a *programs*. Suppose for example that we type the next two commands at the prompt:

```
> plot(x <- sort(rnorm(50)), type = "s")
> points(x, cex = .5, col = "dark red")
```

On my computer, these commands produce the graph in figure 12.1. We might be rather happy with this figure, and decide to save these commands in a file. In interpreted languages such as R, a sequence of commands saved in a file is often called a **script**. A more generic term is **program**.

There are many reasons why you'll need to write and work with scripts. As the tasks you implement become longer and more complicated, typing the commands one-by-one at the prompt becomes impractical. When you have a long sequence of commands that need to be executed in a certain order, writing a script allows to you run the whole sequence of commands easily, isolate problems, make incremental improvements and so on. The resulting program can then be shared with colleagues, etc.

When creating a script for the first time, the most important thing to remember is that scripts are saved in **text files**. Students often confuse text files with word processor documents, but they are quite different. R will not be able to read a program written in a word processor such as MS Word unless you specifically save it as a text file.[1]

Usually, text files are manipulated with **text editors**. Pretty much all computing environments supply a basic text editor. On Windows, a simple one is Notepad, while on Mac OS you can use TextEdit. On Linux there are many options. Alternatively, depending on the version of R you are running, you might find that above the R command line there are several menus. In the "File" menu, you should have the option to create a script. Selecting this option opens up a simple text editor.

Good text editors include features such as syntax highlighting, automatic indentation, code completion, and so on. These features may sound minor, but in fact they make coding much faster and more enjoyable. There are plenty of good open source text editors. By all means, google around to find out about text editors that play well with R, and install a nice one on your home computer.

Now let's say that we've written a script in some text editor and we want to run it. For a short script, the most elementary way to run it is to open up the text file with the text editor, and then copy and paste into R's interpreter (i.e., into the command line, where the $>$ prompt is). While this is not a bad way to start off, a much faster way is to use the R function `source`, which reads in commands from a specified text file:

```
> source("foobar.R")
```

For serious scripting this second method is the only viable one. However, note that the script must be in the current working directory. See §12.1.1 for information on how to set the current working directory.

Once you are familiar with R, you will find that your work is a mix of reading in commands from programs (source files), and entering commands at the prompt (i.e., interacting directly with the interpreter). The latter is always useful for testing, getting help, and rapid prototyping of programs.

A final comment: Generally speaking, running commands via a script produces the same result as typing the commands at the prompt one by one. One exception is as

[1] Yes, if you save as text then you can actually use a word processor to write scripts. But please don't. For starters, no-one will take you seriously if you tell them you code in MS Word. It's just not cool. More importantly, word processors aren't designed for the job, and they don't do it well.

follows: If I have a variable `x` and I type `x` at the prompt and hit return, I get the value of the variable. In a script run via `source`, you will need an explicit call to `print` or `cat` to get the information to the screen.

12.2 Conditions

[roadmap]

12.2.1 Comparisons

We have already met the logical values `TRUE` and `FALSE`:

```
> x <- TRUE
> x
[1] TRUE
```

(In many situations, `TRUE` and `FALSE` can be shortened to `T` and `F`.) Some expressions in R *evaluate* to either `TRUE` or `FALSE`. For example:

```
> 2 > 3
[1] FALSE
> 3 >= 2
[1] TRUE
```

Here we're testing for strict and weak inequality respectively, using the **relational operators** > and >=. Testing for equality and inequality is as follows:

```
> 2 == 3   # Note double equal sign!!
[1] FALSE
> 2 != 3
[1] TRUE
```

The exclamation mark means "not," and reverses truth values. For example:

```
> is.numeric("foo")
[1] FALSE
> !is.numeric("foo")
[1] TRUE
```

Note the double equal sign when *testing* for equality. A single equal sign means *assignment* (i.e., is equivalent to <-). For example, consider:

```
> x <- 1
> x == 2    # Testing equality
[1] FALSE
> x = 2     # Assignment, equivalent to x <- 2
> x
[1] 2
```

While on the topic of greater than and less than, one interesting numerical object in R is `Inf`. `Inf` behaves much as the symbol ∞ does in comparisons, as well as arithmetic operations:

```
> 1 > Inf
[1] FALSE
> 10^100 > Inf
[1] FALSE
> Inf + Inf
[1] Inf
> Inf - Inf    #  Result is NaN (Not a Number)
[1] NaN
> 1 + Inf
[1] Inf
> 0 > -Inf
[1] TRUE
> 10 / Inf
[1] 0
```

Let's continue our discussion of the relational operators. When applied to vectors, these operators produce element by element comparisons. For example:

```
> x <- c(1, 2)
> y <- c(0, 10)
> x > y
[1]  TRUE FALSE
```

Here `x[1]` is compared against `y[1]` and `x[2]` is compared against `y[2]`.

Often, we want to compare a whole vector against a single value. For example, let's create a vector `x` and then ask which elements are greater than 1:

```
> x <- c(1, 4, 5, 9, 0)
> x > 1
[1] FALSE  TRUE  TRUE  TRUE FALSE
```

12.2.2 Boolean Operators

Relational operators can be combined with the **Boolean operators** `AND` and `OR`. To understand these operators, consider two statements `P` and `Q`, such as "2 is greater than 3," or "I live on Mars". Given statements `P` and `Q`, we can also consider the statements `P AND Q` and `P OR Q`. The statement `P AND Q` is true if both `P` and `Q` are true, and false otherwise. The statement `P OR Q` is false if both `P` and `Q` are false, and true otherwise.

In R, the operators `AND` and `OR` are represented by the symbols & and `|`respectively:

```
> 1 < 2 & 2 < 3  # AND: Both true, so true
[1] TRUE
> 1 < 2 & 2 < 1  # AND: One false, so false
[1] FALSE
> 1 < 2 | 2 < 1  # OR: One true, so true
[1] TRUE
> 1 < 2 | 2 < 3  # OR: Both true, so true
[1] TRUE
```

Try experimenting with different combinations.

The operators `AND` and `OR` can also be applied to vectors. As usual, the action is performed element by element:

```
> x <- c(1, 4, 5, 9, 0)
> x >= 5 & x <= 7  #  All x in the interval [5, 7]
[1] FALSE FALSE  TRUE FALSE FALSE
> x <= 1 | x > 5
[1]  TRUE FALSE FALSE  TRUE  TRUE
```

12.2.3 Boolean Arithmetic

As we saw earlier, the values `TRUE` and `FALSE` are primitive data types in R, of class **logical**:

```
> mode(TRUE)
[1] "logical"
```

One important property of logical values is that they *can be used in algebraic expressions,* where `TRUE` evaluates to one and `FALSE` evaluates to zero:

```
> FALSE + TRUE
[1] 1
> FALSE * TRUE
[1] 0
> sum(c(TRUE, TRUE, FALSE))
[1] 2
```

This is very handy. For example, if we want to know how many elements of a numerical vector `y` exceed 3, we can use the command

```
> sum(y > 3)
```

If we want to know the *fraction* of elements of `y` that exceed 3, we can use

```
> mean(y > 3)
```

Can you see how this works? Make sure that it's clear in your mind.

12.2.4 Conditional Extraction

One important fact regarding logical values is that vectors can be indexed by logical vectors. For example,

```
> y <- seq(2, 4, length=5)
> y
[1] 2.0 2.5 3.0 3.5 4.0
> index <- c(TRUE, FALSE, FALSE, FALSE, TRUE)
> y[index]      # Extract first and last element of y
[1] 2 4
```

This feature of vector indexing allows us to perform *conditional extraction* on vectors with very simple syntax. For example, if `y` is any vector, then

```
> y[y > 3]
```

returns all elements of y that exceed 3. This works because the expression inside the square brackets produces a logical vector, and only the elements of y corresponding to TRUE are extracted.

Here's an example of what we can achieve with conditional extraction. Let's suppose we have a data frame called wages, the first column of which records the sex of the individual, and the second of which records his or her salary. The first few lines of wages are as follows:

```
    sex salary
1    F  11.21
2    F  10.79
3    M   8.92
4    F  10.42
5    M   9.75
6    F   9.90
```

How can we compute the average wage for females? We can do it in one line, like so:

```
> mean(wages$salary[wages$sex=="F"])
[1] 10.03096
```

Take your time to think through how this works.

12.2.5 If-Then-Else

Next let's discuss the if-then-else construct. In the simplest case, we can combine logical expressions with the if statement to determine whether a piece of code should be executed or not. For example,

```
> if (2 > 3) print("foo")
```

prints nothing, while

```
> if (3 > 2) print("foo")
```

prints "foo". Conditionals are mainly used in programs, rather than at the command line. Listing 18 gives a (completely artificial) example of the full if-then-else construct, contained in a small program written in a text file. Try reproducing and then running it, and see what it does.

Listing 18 If-then-else

```
password <- readline("Enter your password: ")
if (password == "foobar") {
    print("Welcome")
    # Do something
} else {
    print("Access denied.")
    # Do something else
}
```

Often, the if-then-else syntax can be replaced by the convenient function `ifelse`. To illustrate the latter, consider the following:

```
> ifelse(1 > -1, "foo", "bar")   # Returns "foo"
[1] "foo"
> ifelse(-1 > 1, "foo", "bar")   # Returns "bar"
[1] "bar"
```

The first statement inside the brackets is evaluated. If true, the second value is returned. If false, the third value is returned.

The function `ifelse` is vectorized. For example, suppose we have a vector of data on years of schooling, including university:

```
> ys
[1] 10 12 15 12 16 17 11
```

We want to create a dummy (i.e., binary) variable in a new vector `tertiary` that has value 1 if more than 12 years of schooling has been attained (i.e., tertiary educated) and zero if not. This can be accomplished as follows:

```
> tertiary <- ifelse(ys > 12, 1, 0)
> tertiary
[1] 0 0 1 0 1 1 0
```

12.3 Repetition

The beauty of computers is that they can perform lots of small caculations quickly—much faster than a human. If a human needs to intervene and give the command for

each calculation explicitly, this kind of misses the point. What we want to do is provide a set of instructions at the start, detailing all the calculations in a parsimonious way. This is done using *loops*.

12.3.1 For Loops

The most common kind of loop is a `for` loop. Suppose for example that we want to sum the integers from 1 to 100. The next piece of code performs this task:

```
> x <- 0
> for (i in 1:100) {x <- x + i}
> x
[1] 5050
```

How does this work? First `x` is set to zero. Next, `i` is stepped through each element of the vector `1:100`, and the calculation `x <- x + i` is performed at each step. Another way to get R to do this would be to write it all out in full:

```
> x <- 0
> i <- 1
> x <- x + i
> i <- 2
> x <- x + i
. . .          # Many lines omitted
> i <- 100
> x <- x + i
```

You can see that would be more than a little tedious.

The previous example loop is just for illustration. As a matter of fact, you've already learned a simpler way of performing the same calculation:

```
> sum(1:100)
[1] 5050
```

In R, the latter is more efficient than the `for` loop. We'll talk about why that's the case in §12.5.1.

Let's look at another example of a `for` loop. Suppose we want to simulate flipping a coin 1,000 times, and count the number of heads we get in the process. Listing 19

shows how we can do this using a loop. Since the program is a bit longer, it's been written in a text file. This is why we need the explicit `print` call, to get the value of `num.heads` sent to the screen.

How does it work? The variable `i` is stepped through each element of the vector `1:1000`, and the commands inside the curly brackets are performed at each step. The coin flip is simulated by drawing a uniform random number between zero and one. If the number is less than 1/2, the outcome is regarded as heads.

Listing 19 A `for` loop

```
num.heads <- 0
for (i in 1:1000) {
    b <- runif(1)
    if (b < 0.5) num.heads <- num.heads + 1
}
print(num.heads)
```

Once again, there's an easier way to do this in R. For example,

```
> sum(runif(1000) < 0.5)
```

will also do the job. (Can you see how it works?) So will

```
> rbinom(1, size=1000, prob=0.5)
```

If you're not sure about this last one, you need to read up on the binomial distribution.

For loops are very often used to step through the indices of a vector. For example, let's say that we have a numeric vector with data on firm sizes (in number of employees) called `fsize`. We want to create a new vector `ftype` that replaces these numbers with the labels S, M, and L (small, medium and large), depending on whether the number of employees is in $[0, 500)$, $[500, 1000]$ or $[1000, \infty)$. This can be accomplished as in listing 20.

Once again, there are special functions in R that can be used to avoid this loop (see the function `cut`, for example).[2]

[2]Personally, I often favor explicit loops over specialized R functions, because I program in several languages, and my brain is more comfortable with generic—rather than R-specific—coding styles.

Listing 20 Another `for` loop

```
ftype <- character(0)  # Empty character vector

for (i in 1:length(fsize)) {
    if (fsize[i] < 500) ftype[i] <- "S"
    if (fsize[i] >= 500 & fsize[i] <= 1000) ftype[i] <- "M"
    if (fsize[i] > 1000) ftype[i] <- "L"
}
```

From the previous examples, it might seem that `for` loops are almost unnecessary in R, because there are many convenient functions that avoid the need for explicit looping. Often this is true, especially with short programs performing standard operations. However, for longer programs, explicit loops are pretty much essential. Suppose for example that we want to regress a variable such as inflation on all possible subsets of ten different regressors, and see which has the highest adjusted R squared. There are over 1,000 different possible subsets, and a `for` loop would be a natural choice to step through these possibilities.

12.3.2 While Loops

Now let's briefly look at `while` loops. Suppose we want to model flipping a coin until the time the first head appears. In other words, we want to simulate a random variable that returns the number of the flip resulting in the first head.[3] An implementation is given in listing 21. This loop continues to execute the statements inside the curly brackets until the condition `coin.face == 0` evaluates to `FALSE`. The logic of the program is illustrated in figure 12.2.

Now let's repeat this simulation 1,000 times, and calculate the average value of the random variable over these repetitions.[4] To do this, we put our `while` loop inside a `for` loop, as seen in listing 22. In the listing, the second line creates an empty vector `outcomes`. As the `for` loop progresses, this vector is populated with the result of each individual simulation. At the end we take the mean and print it.

[3] This random variable has the so-called *negative binomial* distribution, which can be simulated using `rnbinom`. We'll produce our own implementation for the sake of the exercise.

[4] This gives us an estimate of the expected value—more on this later.

Listing 21 A `while` loop

```
flip.num <- 0
coin.face <- 0
while (coin.face == 0) {
    flip.num <- flip.num + 1
    b <- runif(1)
    if (b < 0.5) coin.face <- 1
}
print(flip.num)
```

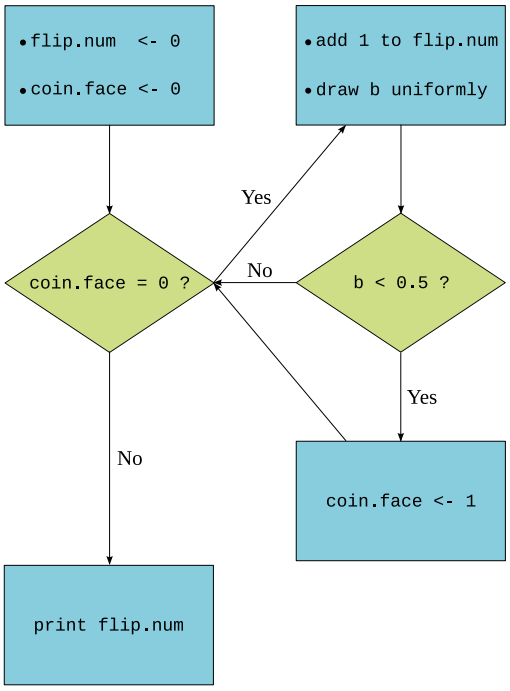


Figure 12.2: Flow chart

Listing 22 Loop inside a loop

```
num.repetitions <- 1000
outcomes <- rep(0, num.repetitions)  # Set up empty vector
for (i in 1:num.repetitions) {
    flip.num <- 0
    coin.face <- 0
    while (coin.face == 0) {
        flip.num <- flip.num + 1
        b <- runif(1)
        if (b < 0.5) coin.face <- 1
    }
    outcomes[i] <- flip.num  # Record result of i-th flip
}
print(mean(outcomes))
```

12.4 Functions

One aspect of programming that separates good and bad programmers is their use of functions. As the next step along our journey to becoming good programmers, let's spend some time learning the mechanics of building our own functions.

12.4.1 Built-In Functions

R provides numerous **functions** as part of the R environment, many of which we've already met. For example, `sum` is a function. It takes a vector as its **argument** (also called **parameter**), and **returns** the sum of that vector's elements:

```
> y <- sum(x)  # x is argument, y gets return value
```

Another built-in function is `integrate`, which performs numerical integration. For example,

```
> integrate(cos, -1, 5)
```

computes the integral $\int_{-1}^{5} \cos(x)dx$ numerically by evaluating the function `cos` at a number of points between -1 and 5, fitting a polynomial to those points, and returning the integral of this approximating polynomial. On my computer, this function call returns the output

```
-0.1174533 with absolute error < 4.3e-14
```

(You can verify that this is approximately correct by computing $-\sin(5)+\sin(-1)$.) The function `integrate` takes three arguments: a function representing the integrand, a number representing the lower bound of integration, and a number representing the upper bound of integration. This example illustrates the fact that R functions can take any object as an argument—a vector, a string, a data frame, or even another function.

Let's look at a third example: the `plot` function. Typically, this function receives at least two arguments, and often many more. Consider, for example, the function call

```
> plot(a, b, col="blue")  # a and b are vectors
```

The first two arguments are called **positional arguments**. Their meaning is determined from their order in the function call. In particular, since `a` appears before `b`, R knows that the elements of `a` correspond to x-axis values, while those of `b` correspond to y-axis values. The ordering is clearly important here, because if `a` and `b` differ, then so will the output of the call

```
> plot(b, a, col="blue")  # Order of a and b reversed
```

The argument `"blue"` is a **named argument**, with `col` being the name of the argument. Named arguments serve two purposes. First, if there are many arguments, then distinguishing by names rather than position makes it easier to remember the roles of the different arguments. Second, named arguments have a default value attached, which means that if such an argument is not supplied, the function can revert to a sensible default.

Note that we write `col="blue"` rather that `col<-"blue"`. When assigning values to positional arguments, one must use = rather than `<-`.

12.4.2 User-Defined Functions

The functions `sum` and `plot` are examples of **built-in functions**, which are part of the R environment. It's often convenient to define our own **user-defined functions**. In fact, when we start writing longer programs, user-defined functions become almost indispensable. Further discussion of "why" is left to §12.4.4. For now let's concentrate on "how."

We begin with an example. In §12.4.1 we used the built-in function `integrate` to calculate the approximate integral of the cosine function over $[-1,5]$ via the call

```
> integrate(cos, -1, 5)
```

Now let's suppose that, for whatever reason, we want to compute $\int_{-1}^{5} x^2 \cos(x) dx$. We can do this by creating our own *user-defined* function that represents $y = x^2 \cos(x)$ and then passing it to `integrate`. The first step is to create the function:

```
> f <- function(x) return(x * x * cos(x))
```

Here `f` is just a name that we've chosen arbitrarily, while `function(x)` indicates that we are creating a function with one argument, called `x`. The built in function `return` determines what value the function will return when we call it.

Let's test out our function to check that it works as expected:

```
> f(3)
[1] -8.909932
> 3 * 3 * cos(3)
[1] -8.909932
```

In the first line we are calling our function, using the function name followed by the argument in brackets. The return value is the right one, as we confirmed in the third and fourth line.

Simple R functions like the one we have just defined are much like the mathematical notion of a function. For example, just as $y = x^2 \cos(x)$ and $y = z^2 \cos(z)$ describe exactly the same functional relationship, the variable name `x` can be any name here. For example,

```
> f <- function(z) return(z * z * cos(z))
```

creates the same function as did the previous function definition.

Anyway, we are now ready to perform the integration, passing `f` as the first argument to `integrate`:

```
> integrate(f, -1, 5)
-18.97950 with absolute error < 2.5e-13
```

Now let's create some more complicated functions. In doing so, we can have as many arguments as we like. For example, the code

```
> f <- function() print("foo")
> f()
[1] "foo"
```

creates and calls a function with no arguments, and no specified return value.[5] For an example with two arguments, consider

```
> g <- function(x, y) return(x^2 + 2 * x * y + y^2)
> g(2, 3)
[1] 25
```

We can also create functions with named arguments:

```
> h <- function(x, intercept=0) return(intercept + 2 * x)
```

Here the statement `intercept=0` in the definition of the function `h` indicates that `intercept` is a named argument with default value 0. If the function is called without specifying a value for `intercept`, the default value will be used:

```
> h(1)
[1] 2
> h(1, intercept=3)
[1] 5
```

Note that, although functions can have many arguments, they always have just one return value (i.e., they always return a single object). If you want to send back multiple pieces of data from your user-defined function, then bundle that data into a vector or a list.

Many functions are larger than the ones we've described, with each call involving a sequence of commands. To create such a function, we enclose the commands in curly brackets. For example, the function in listing 23 packages the simulation in listing 21 in a more convenient form. Notice the curly brackets in the first and last line, which indicate the start and end of the function body respectively. Commands inside these brackets are executed at each function call.

The function in listing 23 has an argument `q`, that represents the probability of heads for our (biased) coin. The function returns the number of flips it took to obtain the first heads. Here's how the function is called:

```
> f(.01)
[1] 408
> f(.9)
[1] 1
```

Why did calling `f` with a small number return a big number?

[5] Actually, `f` returns the string `"foo"`, even though we did not specify a return value.

Listing 23 A longer function

```
f <- function(q) {   #  q = the probability of heads
    flip.num <- 0
    coin.face <- 0
    while (coin.face == 0) {
        flip.num <- flip.num + 1
        b <- runif(1)
        if (b < q) coin.face <- 1  # with prob q
    }
    return(flip.num)
}
```

12.4.3 Variable Scope

One technical issue about functions needs to be mentioned: Variables defined inside a function are **local variables**. For example, consider the following code:

```
> x <- 1
> f <- function() {x <- 2; print(x)}
> f()
[1] 2
> x
[1] 1
```

You might find it surprising to see that even after the function call `f`, which involved binding `x` to 2, when we query the value of `x` at the end we find it is still bound to 1.

Here's how it works: The first assignment `x <- 1` binds the name `x` to 1. This variable has what is called **global scope**, because it is created outside of any function. The next assignment `x <- 2` occurs inside the body of the function `f`. This variable has **local scope**. When we execute the function call `f`, the local variable `x` is created in a separate environment specific to that function call, and bound to the value 2. Any use of the name `x` inside that environment is resolved by first looking for the variable name inside this environment, and, if it is not found, then looking in the global environment. In this case, the name `x` is found in the local environment, and the local value 2 is printed.

Once execution of the function call finishes, the local environment created for that function call is destroyed, and the local `x` is lost. Execution returns to the global

environment. Now, when we query the value of `x`, R looks for this variable name in the global environment. In this case, the value returned is 1.

Almost all programming languages differentiate between local and global variables. The reason is data encapsulation: If you call some function in R that implements a complex operation, that function will likely declare lots of variables that you have no prior knowledge of. It would not be a happy situation in those variable names conflicted with the variable names that you are using in your global environment.

12.4.4 Why Functions?

The single most important reason to use functions is that they break programs down into smaller logical components. Each of these logical components can be designed individually, considering only the task it must perform. This process of reducing programs to functions fits our brains well because it corresponds to the way that solve complex problems in our heads: By breaking them down into smaller pieces.

Related to this point is the fact that, because they are used to solve specific tasks, functions encourage code reuse. For example, suppose for some reason that R had no `integrate` function for numerical integration, and everyone had to write their own. This would involve an enormous duplication of effort, since numerical integration is common to many statistical problems. Moreover, if there's just one implementation that everyone uses, more time can be spent making that one implementation as good as possible.

12.5 General Programming Tips

Let's finish up our introduction to the R language by covering some general pointers for writing programs in R (and other related languages).

12.5.1 Efficiency

If you persist with statistics and econometrics, sooner or later you will bump up against the limits of what your computer can do. Computers may be getting faster, but data sets are also getting larger, and the programming problems tackled by econometricians are becoming increasingly complex. Throwing money at these

problems by buying new hardware often makes little difference. If you are faced with such a problem, then you will almost always need to look at the efficiency of your code.

In this course we won't be dealing with huge data sets or enormous computational problems. However, it's worth understanding the basics of how interpreted languages like R work, in order that code can be structured appropriately.

All standard computer programs must be converted into **machine code** before they can be executed by the CPU. In a compiled language such as C, Fortran or Java, this is done in one pass of the entire program, prior to execution by the user. On the other hand, in an interpreted language such as MATLAB or R, individual commands are converted to machine code on the fly.

Once off compilation prior to execution is efficient for two reasons. First, the compiler is able to see the program as a whole, and optimize the machine code accordingly. Second, interpreted languages must pay the overhead of continually calling the machine code compiler, whereas compiled languages like C need do this only once. As a result, a language like C can be hundreds of times faster than R in certain operations.

Why don't we all just program in C then, if it's so much faster? Go ahead and try, and you will soon find out: Programming statistics in C is a painful experience. R may not be optimized for computers, but it is optimized for humans, and human time is far more valuable than computer time.

In fact, R is written mainly in C. You can think of R as a friendly interface to C, suitable for statistical calculations. The benefit is that most of the necessary C routines have been coded up for you, and all you have to do to use them is type in intuitive commands at the prompt. The cost is that you lose low-level control relative to the alternative option of hand-coding C yourself.

Even if, like most people, you decide to write statistical procedures in R rather than hand-code them in C, you can still learn from the preceding discussion. In particular, we can learn that to program R efficiently, we need to pack big batches of operations into individual commands. This allows R to pass the whole operation out to optimized machine code, pre-compiled from purpose-built C routines.

To illustrate these ideas, suppose we write our own naive function to obtain the square of all elements in a vector `x`, as in listing 24. Let's compare it against the natural, vectorized operation in R:

Listing 24 A slow loop

```
f <- function(x) {
    y <- numeric(length(x))
    for (i in 1:length(x)) {
        y[i] <- x[i]^2
    }
    return(y)
}
```

```
> n <- 10^6
> x <- rnorm(n)
> system.time(x^2)
   user  system elapsed
  0.024   0.004   0.046
> system.time(f(x))
   user  system elapsed
  3.412   0.012   3.423
```

We see that our function is over 100 times slower, because R blindly steps through the instructions in the `for` loop, translating into machine code as it goes. On the other hand, the vectorized method allows R to see the problem as a whole, and pass it to the compiler in an optimal way.

As a rule of thumb, vectorized calculations are far more efficient than explicit loops. When R calculations can be vectorized, the performance of R is often quite similar to that of C, Fortran or Java. When operations are not vectorized, it can be far slower.

12.5.2 Clarity and Debugging

Having talked about efficiency, it's now very important to note that *very little of your code needs to be optimized.* Often, 99% of your CPU time will be used by a tiny subset of the code that you write. Only this code needs to be optimized, and only if excessive run-time of the program justifies the extra work. Once you've spent a few days debugging your programs, it will become very clear to you that, for the vast majority of your code, clarity is the priority.

Clarity is crucial, since writing and reading programs is not an easy business for

human beings. That said, there are some things you can do to make it easier on yourself and others who read your programs. One is to add comments to your programs. (A comment is a # symbol, followed by some text.) The text can be anything, but the idea is to make a useful comment on your code. Adding comments is helpful to others who read your code, and to you when you come back to your code after several weeks, months or years.

Another useful technique to improve the clarity of your code is to use indentation (i.e., whitespace at the start of a line of code). For example, the indentation in listings 22 and 23 helps to separate different logical parts of the program for the reader. (Like comments, this is irrelevant to the R interpreter: Whitespace is ignored.)

Despite your best efforts at clarity, however, you will still find that a lot of your programming time is spend hunting down bugs. Errors come in two main classes: **syntax errors,** which are flaws in the syntax of your instructions, and cause R to issue an error message. The other kinds of errors are **semantic errors,** which are logical mistakes that cause your program to operate incorrectly. These can be very difficult to track down, because you don't have the benefit of an error message.

Debugging is a bit of an art, and I won't say much about it, apart from suggesting that, should you have a bug in one of your programs, a good first step is to fill the program with calls to `cat` such as

```
cat("x =", x, "and y =", y, "\n")
```

so that you can keep track of your variables during execution. Although this technique for tracking your variables is not very sophisticated, it can be extremely helpful.

12.6 More Statistics

[Roadmap]

12.6.1 Distributions in R

R has handy functions for accessing all the common distributions. These functions have the form

```
lettername
```

where "name" is one of the named distributions in R, such as

`norm` (normal), `lnorm` (log normal), `unif` (uniform), etc.

and "letter" is one of `p` , `d`, `q` or `r`. The meanings of the letters are

- `p` cumulative distribution function
- `d` density function
- `q` quantile function
- `r` generates random variables

Here are a couple of examples:

```
> pnorm(2, mean=.1, sd=2)  # F(2), cdf of N(.1, 4)
> qcauchy(1/2)  # median of cauchy distribution
> runif(100, 2, 4)  # 100 uniform r.v.s on [2, 4]
```

See the documentation for further details on these functions.

With respect to random number generation, you should be aware that "random" numbers generated by a computer are not truly random. In fact they are not random at all—they are generated in a purely deterministic way according to specified rules. By clever design of these rules, it is possible to generate deterministic sequences the statistical properties of which resemble independent draws from common distributions. These sequences are called **pseudo random numbers.**

Pseudo random numbers follow well defined patterns determined by initial conditions. By default, these initial conditions come from the system clock and are different each time you call on the random number generator. This is a good thing if you want new draws each time. However, sometimes it's helpful to set the initial conditions, such as when you run a simulation experiment, and you want others to be able to reproduce your results.

The way to do this is via the function `set.seed`, which sets the initial conditions (seed) for the random number generator. Here's an example of usage:

```
> set.seed(123)
> runif(5)   # First draw
[1] 0.2875775 0.7883051 0.4089769 0.8830174 0.9404673
> runif(5)   # Draw again, without resetting seed
[1] 0.0455565 0.5281055 0.8924190 0.5514350 0.4566147
> set.seed(123)
> runif(5)   # Back to start again
[1] 0.2875775 0.7883051 0.4089769 0.8830174 0.9404673
```

12.6.2 Working with Vectors

Vectors in R are rather similar to the abstract mathematical notion of a vector we've just discussed. In particular, they have no notion of being either row vectors or column vectors. This can be observed by displaying the `dim` attribute associated with a given vector via the function `dim`. Here's an example:

```
> x <- rnorm(20)   # Create an arbitrary vector x in R
> dim(x)           # By default, x is a flat vector
NULL
```

If we wish to, we can alter the `dim` attribute to make `x` a column or row vector.

```
> dim(x) <- c(20, 1)   # Make x a column vector
> dim(x)
[1] 20  1
```

This is useful for performing matrix multiplication, an operation that distinguishes between row and column vectors.

As we saw in §11.2.4, scalar multiplication and vector addition are performed in a natural way:

```
> x <- 1:4           # Vector
> y <- 5:8           # Vector
> a <- 1             # Scalar
> b <- .5            # Scalar
> a * x              # Scalar multiplication
> x + y              # Vector addition
> a * x + b * y      # Both together
```

12.6.3 Working with Matrices

Let's look at how to work with matrices in R. Most often matrices are read in from data files, or created from other computations. We can also create them from scratch if we need to. Here, we create the matrices **A** and **B** defined on page 69:

```
> A <- matrix(c(10, 20, 30, 40, 50, 60), nrow=3, ncol=2)
> A
     [,1] [,2]
```

```
[1,]   10   40
[2,]   20   50
[3,]   30   60
> B <- matrix(c(1, 2, 3, 4, 5, 6), nrow=2, ncol=3)
> B
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

Alternatively, we can create `B` in two steps:

```
> B <- c(1, 2, 3, 4, 5, 6)
> class(B)  #  B is a flat vector with no dimension
[1] "numeric"
> dim(B) <- c(2, 3)  # Set the dimension attribute
> B
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> class(B)      # Now B is a matrix
[1] "matrix"
```

Finally, here's a third way that I use often:

```
> B <- matrix(nrow=2, ncol=3)
> B
     [,1] [,2] [,3]
[1,]   NA   NA   NA
[2,]   NA   NA   NA
> B[1,] <- c(1, 3, 5)
> B[2,] <- c(2, 4, 6)
> B
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

Internally, matrices in R are closely related to data frames. In particular, we can access elements of the matrices using square brackets followed by row and column numbers:

```
> A[3, 2]   #  Third row, second column
[1] 60
> A[3,]     #  Third row, all columns
[1] 30 60
> A[, 2]    #  All rows, second column
[1] 40 50 60
```

A and **B** are not conformable for addition, but $\mathbf{A}'$ and **B** are. To take the transpose of **A** we use the transpose function `t`:

```
> t(A)
     [,1] [,2] [,3]
[1,]   10   20   30
[2,]   40   50   60
```

Addition is now straightforward:

```
> t(A) + B
     [,1] [,2] [,3]
[1,]   11   23   35
[2,]   42   54   66
```

Notice that the multiplication symbol `*` gives element by element multiplication, as follows

```
> t(A) * B
     [,1] [,2] [,3]
[1,]   10   60  150
[2,]   80  200  360
```

This is natural, because it's consistent with the algebraic operations on vectors we saw earlier. Matrix multiplication is different, and uses the symbol combination `%*%`:

```
> A %*% B
     [,1] [,2] [,3]
[1,]   90  190  290
[2,]  120  260  400
[3,]  150  330  510
```

As we learned above, the product **AB** is formed by taking as it's i, j-th element the inner product of the i-th row of **A** and the j-th column of **B**. For example:

```
> A[2,] %*% B[,3]
     [,1]
[1,]  400
```

We can also check the claim in fact 2.3.5 that $(\mathbf{AB})' = \mathbf{B}'\mathbf{A}'$:

```
> t(A %*% B)
     [,1] [,2] [,3]
[1,]   90  120  150
[2,]  190  260  330
[3,]  290  400  510
> t(B) %*% t(A)
     [,1] [,2] [,3]
[1,]   90  120  150
[2,]  190  260  330
[3,]  290  400  510
```

Let's have a look at solving linear systems of equations. Suppose now that $\mathbf{A}$ is the matrix

```
> A
     [,1] [,2] [,3]
[1,]   58   25    3
[2,]   43   97   90
[3,]   18   32   80
```

and $\mathbf{b}$ is the column vector

```
> b
     [,1]
[1,]    1
[2,]    2
[3,]    3
```

We are interesting in solving the system of equations $\mathbf{Ax} = \mathbf{b}$. According to fact 2.3.3, a unique solution will exist provided that $\mathbf{A}$ has nonzero determinant (and is therefore of full rank). Let's check this:

```
> det(A)
[1] 236430
```

The inverse of $\mathbf{A}$ can be calculated as follows:

```
> Ainv <- solve(A)
> Ainv
              [,1]          [,2]         [,3]
[1,]  0.020640359 -0.008053124  0.00828575
[2,] -0.007697839  0.019396862 -0.02153280
[3,] -0.001564945 -0.005946792  0.01924883
```

In view of (2.3), we can solve for $\mathbf{x}$ as $\mathbf{A}^{-1}\mathbf{b}$:

```
> x <- Ainv %*% b
> x
             [,1]
[1,]  0.02939136
[2,] -0.03350252
[3,]  0.04428795
```

While this is valid in theory, it turns out that there are more efficient ways to do this numerically. As a rule, computing inverses of matrices directly is relatively unstable numerically, in the sense that round off errors have large effects. Other techniques are available that mitigate this problem. In R, the preferred method is to use the formula `solve(A, b)`:

```
> solve(A, b)
             [,1]
[1,]  0.02939136
[2,] -0.03350252
[3,]  0.04428795
```

In this case we can see that there was no difference in the results produced by the two techniques, but for large matrices the latter method is significantly more accurate and computationally efficient.

Let's conclude this section with a few tips for working with matrices and vectors. First, suppose that we want to use a vector in a matrix operation such as matrix multiplication. Since a vector in R has no dimension attribute, how will R know whether to treat it as a column vector or a row vector? (Clearly the result will depend on which choice is made.) The answer is that R will make an educated guess, depending on which of the two choices is conformable. The next piece of code illustrates:

```
> A <- matrix(c(1, 2, 3, 4), nrow=2)
```

```
> A
     [,1] [,2]
[1,]    1    3
[2,]    2    4
> x <- c(5, 6)
> dim(x)   #  x is a flat vector, with no dimension
NULL
> A %*% x
     [,1]
[1,]   23
[2,]   34
> x %*% A
     [,1] [,2]
[1,]   17   39
```

In the first case x is treated as a column vector, while in the second it is treated as a row vector.

The case `x %*% x` is ambiguous, and R always gives the inner product.

Sometimes we need to combine matrices or vectors. Here's an example that forms a matrix by stacking two vectors row-wise and then column-wise.

```
> a <- c(1, 2, 3)
> b <- c(4, 5, 6)
> cbind(a, b)
     a b
[1,] 1 4
[2,] 2 5
[3,] 3 6
> rbind(a, b)
  [,1] [,2] [,3]
a    1    2    3
b    4    5    6
```

The function `diag` provides simple way to extract the diagonal elements of a matrix:

```
> A
 [,1] [,2]
[1,]    1    3
[2,]    2    4
```

```
> diag(A)
[1] 1 4
```

(The trace can now be obtained by summing.) Somewhat confusingly, the same function is also used to create diagonal matrices:

```
> diag(3)
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
```

Last but not least, we often need to create matrices of zeros or ones. Here's how:

```
> matrix(1, 3, 2)
     [,1] [,2]
[1,]    1    1
[2,]    1    1
[3,]    1    1
> matrix(0, 4, 4)
     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    0    0    0    0
[3,]    0    0    0    0
[4,]    0    0    0    0
```

[internally, matrices are just flat arrays, so some vectorized operations can be performed one-off. example, want to center each row around its mean. write function, use apply.]

12.6.4 Multiple Regression in R

Let's look briefly at running multiple regressions in R. The standard method is to use the function `lm`, which we discussed in §11.5. Let's now recall the method, and compare it with the result of direct calculations according to the theoretical results we have derived. As a starting point, we generate some synthetic data:

```
> set.seed(1234)
> N <- 500
```

```
> beta <- c(1, 1, 1)
> X <- cbind(rep(1, N), runif(N), runif(N))
> y <- X %*% beta + rnorm(N)
```

The seed for the random number generator has been set so that the same pseudo-random numbers are produced each time the program is run. The matrix $\mathbf{X}$ consists of a vector of ones as the first column, and then two columns of observations on non-constant regressors. Hence, $K = 3$. The coefficient vector $\boldsymbol{\beta}$ has been set to $(1, 1, 1)$ for simplicity.

We can run an OLS regression in R as follows.

```
> results <- lm(y ~ 0 + X)
```

In the call to `lm`, the term `0` indicates that a constant term should not be added—in this case, because we already set up $\mathbf{X}$ to have a constant vector as the first column.

As a first step, let's compare the estimate of the coefficient vector produced by `lm` to the one we get when using the theoretical formula directly:

```
> results$coefficients
       X1        X2        X3
0.9662670 0.9532119 1.0045879
> solve(t(X) %*% X) %*% t(X) %*% y
          [,1]
[1,] 0.9662670
[2,] 0.9532119
[3,] 1.0045879
```

Here, exactly the same results are produced. However, you should be aware that the direct method is less stable numerically. Numerical error will be zero or negligible in mosts settings, but may be significant when data sets are large and multicollinearity is present in the data. For a discussion of multicollinearity, see §7.3.5. For an illustration of this numerical instability, see exercise 6.4.20.

Let's also check our theoretical methods for computing $\hat{\mathbf{y}}$ and $\hat{\mathbf{u}}$. Since these vectors are too large to print out and compare, we'll look instead at the squared norms of the vectors, which correspond to the ESS and SSR respectively:

```
> yhat <- results$fitted.values
> uhat <- results$residuals
> P <- X %*% solve(t(X) %*% X) %*% t(X)
```

```
> M <- diag(N) - P
> sum(yhat * yhat)  # Fitted values calculated by lm()
[1] 2002.222
> sum((P %*% y) * (P %*% y))  # Fitted vals, direct method
[1] 2002.222
> sum(uhat * uhat)  # Residuals calculated by lm()
[1] 458.4417
> sum((M %*% y) * (M %*% y))  # Residuals, direct method
[1] 458.4417
```

Again, the results of `lm` and the results of our direct calculations from our theoretical results are in exact agreement.

Regarding the coefficient of determination, the calculation by `lm` can be viewed using the `summary` function:

```
> summary(results)
# Some lines omitted
Multiple R-squared: 0.8137
```

According to the theory, this should agree with $\|\hat{\mathbf{y}}\|^2 / \|\mathbf{y}\|^2$. Let's check this:

```
> sum(yhat * yhat) / sum(y * y)
[1] 0.8136919
```

The results are in agreement.

12.7 Exercises

Ex. 12.7.1. How could you use `sum` to determine whether a numeric vector `x` contains the value 10?

Ex. 12.7.2. Suppose we are processing vector of zeros and ones. The vectors corresponds to the employment histories of individuals. In a given vector, 1 means that the individual was employed at the associated point in time, while 0 means unemployed. Write a function that takes such a vector (of arbitrary length) and compute the longest (consecutive) period of employment.

[new exercise: computing a Lorenz curve from a data set on the web. and maybe, computing and plotting a set of Lorenz curves from a collection of data sets, using

a loop. make this a computer lab as well? to replace computer lab on chaotic dynamics? Lorenz curves can be calculated with for loops, or using built in piecewise linear function approximation.]

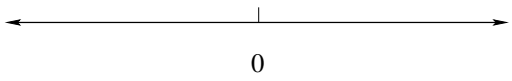
[add more exercises!]

Chapter 13

Appendix C: Analysis

13.1 Sets and Functions

In the course we often refer to the **real numbers**. This set is denoted by $\mathbb{R}$, and we understand it to contain "all of the numbers." $\mathbb{R}$ can be visualized as the "continuous" real line:



It contains both the rational and the irrational numbers.

What's "real" about the real numbers? Well, "real" is in contrast to "imaginary," where the latter refers to the set of imaginary numbers. Actually, the imaginary numbers are no more imaginary (or less real) than any other kind of numbers, but we don't need to talk any more about this.

$\mathbb{R}$ is an example of a **set**. A set is a collection of objects viewed as a whole. (In this case the objects are numbers.) Other examples of sets are the set of all rectangles in the plane, or the set of all monkeys in Japan.

If A is a set, then the statement $x \in A$ means that x is contained in (alternatively, is an element of) A. If B is another set, then $A \subset B$ means that any element of A is also an element of B, and we say that A is a **subset** of B. The statement $A = B$ means that A and B contain the same elements (each element of A is an element of B and

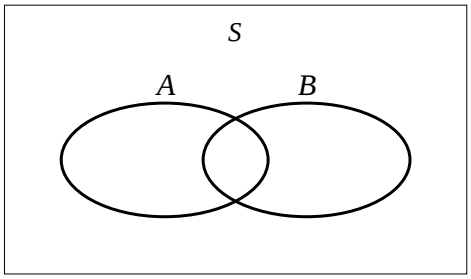


Figure 13.1: Sets A and B in S

vice versa). For example, if $\mathbb{I}$ is the irrational numbers,[1] then $\mathbb{I} \subset \mathbb{R}$. Also, $0 \in \mathbb{R}$, $\pi \in \mathbb{R}$, $-3 \in \mathbb{R}$, $e \in \mathbb{R}$, and so on.

Commonly used subsets of $\mathbb{R}$ include the intervals. For arbitrary a and b in $\mathbb{R}$, the **open inverval** (a, b) is defined as

$$(a, b) := \{x \in \mathbb{R} : a < x < b\}$$

while the **closed inverval** $[a, b]$ is defined as

$$[a, b] := \{x \in \mathbb{R} : a \leq x \leq b\}$$

We also use half open intervals such as $[a, b) := \{x \in \mathbb{R} : a \leq x < b\}$, half lines such as $(-\infty, b) = \{x \in \mathbb{R} : x < b\}$, and so on.

Let S be a set and let A and B be two subsets of S, as illustrated in figure 13.1. The **union** of A and B is the set of elements of S that are in A or B or both:

$$A \cup B := \{x \in S : x \in A \text{ or } x \in B\}$$

Here and below, "or" is used in the mathematical sense. It means "and/or". The **intersection** of A and B is the set of all elements of S that are in both A and B:

$$A \cap B := \{x \in S : x \in A \text{ and } x \in B\}$$

The set $A \setminus B$ is all points in A that are not points in B:

$$A \setminus B := \{x \in S : x \in A \text{ and } x \notin B\}$$

The **complement** of A is the set of elements of S that are not contained in A:

$$A^c := S \setminus A :=: \{x \in S : x \notin A\}$$

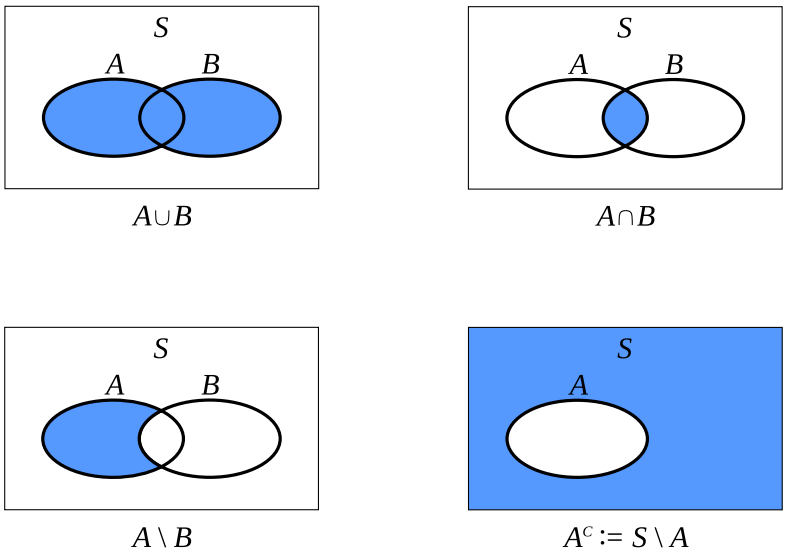


Figure 13.2: Unions, intersection and complements

Here $x \notin A$ means that x is not an element of A. Figure 13.2 illustrate these definitions.

For example, since $\mathbb{R}$ consists of the irrationals $\mathbb{I}$ and the rationals $\mathbb{Q}$, we have

$$\mathbb{Q} \subset \mathbb{R},\ \mathbb{I} \subset \mathbb{R},\ \mathbb{Q} \cup \mathbb{I} = \mathbb{R},\ \mathbb{Q}^c = \mathbb{I},\ \text{etc.}$$

Also,

$$\mathbb{N} := \{1, 2, 3, \ldots\} \subset \mathbb{Q} \subset \mathbb{R}$$

The **empty set** is, unsurprisingly, the set containing no elements. It is denoted by $\emptyset$. If the intersection of A and B equals $\emptyset$, then A and B are said to be **disjoint**.

The next fact lists some well known rules for set theoretic operations.

Fact 13.1.1. Let A and B be subsets of S. The following statements are true:

1. $A \cup B = B \cup A$ and $A \cap B = B \cap A$.
2. $(A \cup B)^c = B^c \cap A^c$ and $(A \cap B)^c = B^c \cup A^c$.

[1]The **irrationals** are those numbers such as π and $\sqrt{2}$ that cannot be expressed as fractions of whole numbers.

3. $A \setminus B = A \cap B^c$.

4. $(A^c)^c = A$.

13.1.1 Functions

Let A and B be sets. A **function** f from A to B is a rule that associates to each element a of A one and only one element of B. That element of B is usually called the **image of a under f**, and written $f(a)$. If we write $f\colon A \to B$, this means that f is a function from A to B.

For an example of a function, think of the hands on an old school clock. Let's say we know it's the morning. Each position of the two hands is associated with one and only one time in the morning. If we don't know it's morning, however, one position of the hands is associated with two different times, am and pm. The relationship is no longer functional.

Figure 13.3 is instructive. Top right is not a function because the middle point on the left-hand side is associated with two different points (images). Bottom right is not a function because the top point on the left-hand side is not associated with any image. From the definition, this is not allowed.

13.1.2 Convergence and Continuity

Let $\{x_n\}_{n=1}^{\infty}$ be a sequence of real numbers. (For each $n = 1, 2, \ldots$ we have a corresponding $x_n \in \mathbb{R}$.) We say that x_n converges to 0 if, given any neighborhood of 0, the sequence points are eventually in that neighborhood. More formally (we won't use the formal definition, so feel free to skip this), given any $\epsilon > 0$, there exists an $N \in \mathbb{N}$ such that $|x_n| < \epsilon$ whenever $n \geq N$. Symbolically, $x_n \to 0$.

Now let $\{\mathbf{x}_n\}_{n=1}^{\infty}$ be a sequence of vectors in $\mathbb{R}^N$. We say that **$\mathbf{x}_n$ converges to $\mathbf{x} \in \mathbb{R}$** if $\|\mathbf{x}_n - \mathbf{x}\| \to 0$. Symbolically, $\mathbf{x}_n \to \mathbf{x}$. This is the fundamental notion of convergence in $\mathbb{R}^N$. Whole branches of mathematics are built on this idea.

Let $A \subset \mathbb{R}^N$ and $B \subset \mathbb{R}^M$. A function $f\colon A \to B$ is called **continuous at $\mathbf{x}$** if $f(\mathbf{x}_n) \to f(\mathbf{x})$ whenever $\mathbf{x}_n \to \mathbf{x}$, and **continuous** if it is continuous at $\mathbf{x}$ for all $\mathbf{x} \in A$. Figure 13.4 illustrates. The notion of continuity is also massively important to mathematical analysis. However, we won't be doing any formal proofs using the definition—we just state it for the record.

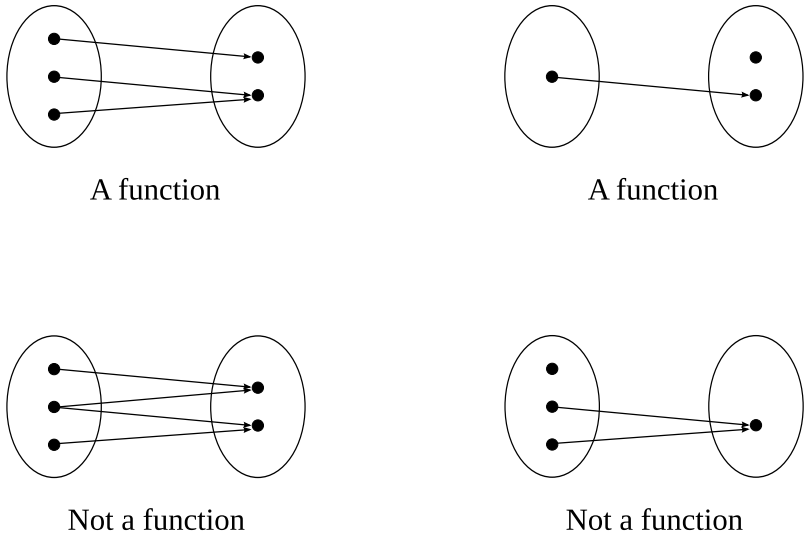


Figure 13.3: Function and non-functions

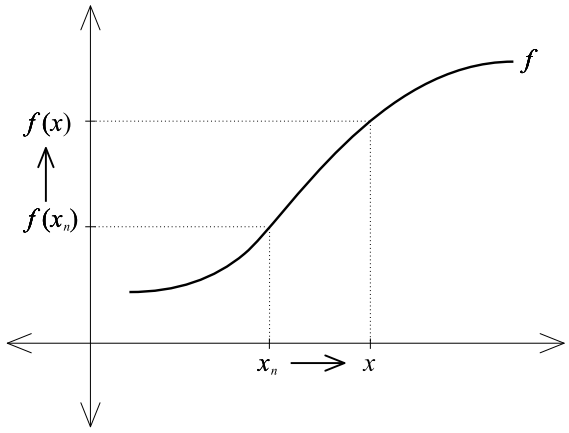


Figure 13.4: Continuity

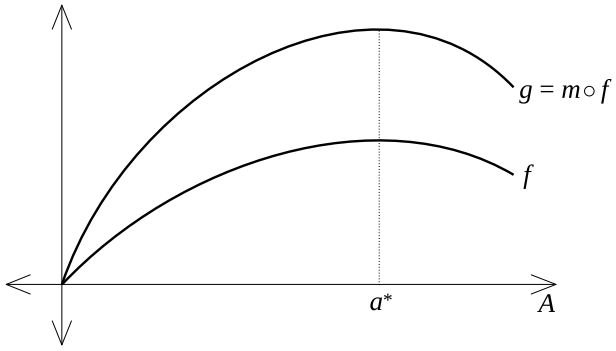


Figure 13.5: Monotone transforms preserve maximizers

13.2 Optimization

Monotone increasing transformations of functions do not affect maximizers:

Let A be any set, and let $f\colon A \to \mathbb{R}$. That is, f is a function such that $f(a)$ is a number for each $a \in A$. A **maximizer** of f on A is a point $a^* \in A$ such that

$$f(a^*) \geq f(a) \text{ for all } a \in A$$

Now let m be a **monotone increasing function**, in the sense that if $x \leq x'$, then $m(x) \leq m(x')$, and let g be the function defined by $g(a) = m(f(a))$. Our claim is this:

> *Any maximizer of f on A is also a maximizer of g on A.*

It's easy to see why this is the case. Let $a \in A$. Since a^* is a maximizer of f, it must be the case that $f(a) \leq f(a^*)$. Since m is monotone increasing, this implies that $m(f(a)) \leq m(f(a^*))$. Given that a was chosen arbitrarily, we have now shown that

$$g(a^*) \geq g(a) \text{ for all } a \in A$$

In other words, a^* is a maximizer of g on A.

Before finishing this topic, let's recall the notions of supremum and infimum. To illustrate, consider the function $f\colon (0,1) \to (0,1)$ defined by $f(x) = x$. It should be clear that f has no maximiser on $(0,1)$: given any $a^* \in (0,1)$, we can always choose another point $a^{**} \in (0,1)$ such that $a^{**} = f(a^{**}) > f(a^*) = a^*$. No maximizer exists and the optimization problem $\max_{x \in (0,1)} f(x)$ has no solution.

To get around this kind of problem, we often the notion of supremum instead. If A is a set, then the **supremum** $s := \sup A$ is the unique number s such that $a \leq s$ for every $a \in A$, and, moreover, there exists a sequence $\{x_n\} \subset A$ such that $x_n \to s$. For example, 1 is the supremum of both $(0,1)$ and $[0,1]$. The **infimum** $i := \inf A$ is the unique number i such that $a \geq i$ for every $a \in A$, and, moreover, there exists a sequence $\{x_n\} \subset A$ such that $x_n \to i$. For example, 0 is the supremum of both $(0,1)$ and $[0,1]$.

One can show that the supremum and infimum of any bounded set A exist, and any set A when the values $-\infty$ and ∞ are admitted as a possible infima and supremum.

Returning to our original example with $f(x) = x$, while $\max_{x\in(0,1)} f(x)$ is not well defined, $\sup_{x\in(0,1)} f(x) := \sup\{f(x) : x \in (0,1)\} = \sup(0,1) = 1$.

13.3 Logical Arguments

To be written:

- Role of counterexamples. Many logical statements are of the form "if it's an A, then it's a B". (Examples.) The statement may be correct or incorrect. To show it's correct, we take an arbitrary A, and prove that it's a B. To show it's false, we provide a counterexample. (Give examples of this process.)

- Contrapositive. Simple example with Venn diagram. Explain how it's useful with an example from the text. (fact 2.2.4?)

- Proof by induction. Simple examples. Relate to (7.27) and (8.7). Relate to discussion of statistial learning and induction.

Bibliography

[1] Amemiya, T. (1994): *Introduction to Statistics and Econometrics,* Harvard UP.

[2] Bishop, C.M. (2006): *Pattern Recognition and Machine Learning,* Springer.

[3] Casella, G. and R. L. Berger (1990): *Statistical Inference,* Duxbury Press, CA.

[4] Cheney, W. (2001): *Analysis for Applied Mathematics,* Springer.

[5] Dasgupta, A. (2008): *Asymptotic Theory of Statistics and Probability,* Springer.

[6] Devroye, Luc and Gabor Lugosi (2001): "Combinatorial Methods in Density Estimation" Springer-Verlag, New York.

[7] Doeblin, W. (1938): "Exposé de la theorie des chaîns simples constantes de Markov à un nombre fini d'états," *Rev. Math. Union Interbalkanique,* 2, 77–105.

[8] Durrett, R. (1996): *Probability: Theory and Examples*, Duxbury Press.

[9] Evans, G. and S. Honkapohja (2005): "An Interview with Thomas Sargent," *Macroeconomic Dynamics*, 9, 561–583.

[10] Freedman, D. A. (2009): *Statistical Models: Theory and Practice,* Cambridge UP.

[11] Greene, W. H. (1993): *Econometric Analysis,* Prentice-Hall, New Jersey.

[12] Hall, R. E. (1978): "Stochastic implications of the life cycle-permanent income hypothesis," *Journal of Political Economy* 86 (6), 971–87.

[13] Hayashi, F. (2000): *Econometrics,* Princeton UP.

[14] Hoerl, A. E. and R. W. Kennard (1970): "Ridge Regression: Biased Estimation for Nonorthogonal Problems," *Technometrics,* 12 (1), 55–67.

[15] Olley, S. and A. Pakes (1996): "The Dynamics of Productivity in the Telecommunications Equipment Industry," *Econometrica*, 64 (6), 1263-97.

[16] Negri, I and Y. Nishiyama (2010): "Review on Goodness of Fit Tests for Ergodic Diffusion processes by Different Sampling Schemes," *Economic Notes*, 39, 91–106.

[17] Stachurski, J. and V. Martin (2008): "Computing the Distributions of Economic Models via Simulation," *Econometrica*, 76 (2), 443–450.

[18] Stachurski, J. (2009): *Economic Dynamics: Theory and Computation*, MIT Press.

[19] Vapnik, V. N. (2000): *The Nature of Statistical Learning Theory*, Springer-Verlag, New York.

[20] Wasserman, L. (2004): *All of Statistics*, Springer, New York.

[21] Williams, D. (2001): *Probabiliity with Martingales*, Cambridge Mathematical Textbooks.

Index