

# Econometrics Notes

In progress, Chapters 1 to 11 available.

Anthony Tay

2023-10-31



# Table of contents

<b>Preface</b>	<b>1</b>
What is Econometrics? . . . . .	1
Mathematical Prerequisites . . . . .	2
Software . . . . .	2
<b>1 A Brief Introduction to R</b>	<b>3</b>
1.1 Getting Set Up . . . . .	3
1.2 Data Types . . . . .	5
1.2.1 Arithmetic and Logical Operators . . . . .	6
1.3 Data Structures . . . . .	7
1.3.1 Vectors . . . . .	7
1.3.2 Factor Datatype . . . . .	10
1.3.3 Data Frames . . . . .	10
1.3.4 Matrices, Lists . . . . .	11
1.3.5 Time Series . . . . .	12
1.4 Importing Data . . . . .	13
1.5 Plotting Data . . . . .	15
1.6 More on the R Environment . . . . .	16
1.7 User-Defined Functions, Conditional Statements, Loops . . . . .	17
<b>2 Miscellaneous Mathematics Topics</b>	<b>21</b>
2.1 The Summation Notation . . . . .	21
2.1.1 Rules for Summation Notation . . . . .	22
2.1.2 Some Useful Formulas . . . . .	23
2.1.3 Double Summations . . . . .	25
2.1.4 Exercises . . . . .	26
2.2 An Introduction to Matrices . . . . .	27
2.2.1 Definitions and Notation . . . . .	27
2.2.2 Addition, Scalar Multiplication and Transpose . . . . .	29
2.2.3 Exercises . . . . .	30
2.2.4 Matrix Multiplication . . . . .	31
2.2.5 Exercises . . . . .	32
2.2.6 Partitioned Matrices . . . . .	35
2.2.7 Determinants and Inverses . . . . .	36
2.2.8 Exercises . . . . .	37
2.2.9 Matrices in R . . . . .	39
2.3 A Brief Review of Optimization Theory . . . . .	42
2.3.1 Functions of One Variable . . . . .	42
2.3.2 Functions of Many Variables . . . . .	46

2.3.3	Exercises . . . . .	48
2.4	Application: Fitting a Straight Line by Least Squares . . . . .	49
2.4.1	Algebraic Properties . . . . .	52
2.5	Exercises . . . . .	55
<b>3</b>	<b>Probability and Expectations Review</b>	<b>57</b>
3.1	Random Variables, Mean and Variance . . . . .	57
3.2	Joint and Conditional Distributions . . . . .	60
3.2.1	Bayes' Theorem . . . . .	62
3.2.2	Mean and Variance, Covariance . . . . .	63
3.2.3	Conditional Means and Variances . . . . .	66
3.2.4	Law of Iterated Expectations . . . . .	67
3.2.5	Independent Random Variables . . . . .	69
3.2.6	Exercises . . . . .	71
3.3	A Few More Distributions . . . . .	72
3.3.1	Geometric Distribution . . . . .	73
3.3.2	Uniform Distribution . . . . .	74
3.3.3	Mean, Variance and Other Moments . . . . .	75
3.3.4	The Normal Distribution . . . . .	77
3.3.5	The Log-Normal Distribution . . . . .	80
3.3.6	The Chi-squared, Student-t, and F Distributions . . . . .	81
3.3.7	The Bivariate Normal Distribution . . . . .	84
3.3.8	Exercises . . . . .	87
3.4	Prediction . . . . .	88
<b>4</b>	<b>Statistics Review</b>	<b>89</b>
4.1	Estimation . . . . .	89
4.1.1	Unbiased Estimators . . . . .	89
4.1.2	Efficiency . . . . .	91
4.1.3	Mean Square Error . . . . .	92
4.2	A Coin Toss Example . . . . .	92
4.3	Hypothesis Testing . . . . .	94
4.4	Asymptotic Analysis . . . . .	96
4.4.1	Consistency and the Law of Large Numbers . . . . .	97
4.4.2	Asymptotic Normality . . . . .	99
4.4.3	The Central Limit Theorem . . . . .	101
4.5	Exercises . . . . .	104
4.6	Prediction . . . . .	105
4.6.1	Exercises . . . . .	106
<b>5</b>	<b>Simple Linear Regression</b>	<b>109</b>
5.1	The Simple Linear Regression Framework . . . . .	109
5.2	Ordinary Least Squares . . . . .	112

5.2.1	Statistical Properties of OLS Estimators . . . . .	116
5.3	Prediction . . . . .	120
5.4	Hypothesis Testing . . . . .	124
5.5	Asymptotic Results . . . . .	127
5.6	When Baseline Assumptions are Violated . . . . .	129
5.6.1	Heteroskedasticity . . . . .	129
5.6.2	Endogeneity . . . . .	129
5.7	Exercises . . . . .	134
<b>6</b>	<b>Multiple Linear Regression</b>	<b>137</b>
6.1	OLS Estimation of the Multiple Linear Regression Model . . . . .	139
6.2	Algebraic Properties of OLS Estimators . . . . .	141
6.3	Statistical Properties of OLS Estimators . . . . .	143
6.4	Hypothesis Testing . . . . .	148
6.5	Exercises . . . . .	155
<b>7</b>	<b>Heteroskedasticity and Specification Tests</b>	<b>157</b>
7.1	An Example . . . . .	157
7.2	Weighted Least Squares . . . . .	160
7.3	Testing for Heteroskedasticity . . . . .	165
7.4	Some Additional Regression Tests . . . . .	166
7.4.1	RESET test for functional form misspecification . . . . .	166
7.4.2	Testing Nonnested Alternatives . . . . .	168
7.4.3	Testing for Normality of Noise Terms . . . . .	169
7.5	Exercises . . . . .	171
<b>8</b>	<b>More Matrix Algebra</b>	<b>173</b>
8.1	Rank . . . . .	173
8.1.1	A Geometric Viewpoint . . . . .	173
8.1.2	The Rank of a Matrix . . . . .	180
8.1.3	Finding the Rank of a Matrix in R . . . . .	181
8.1.4	Exercises . . . . .	181
8.2	Diagonalization of Symmetric Matrices . . . . .	182
8.2.1	Exercises . . . . .	185
8.3	Differentiation of Matrix Forms . . . . .	185
8.3.1	Definitions . . . . .	185
8.3.2	Basic Differentiation Formulas . . . . .	186
8.3.3	Exercises . . . . .	187
8.4	Vectors and Matrices of Random Variables . . . . .	188
8.4.1	Expectations and Variance-Covariance Matrices . . . . .	188
8.4.2	The Multivariate Normal Distribution . . . . .	190
8.4.3	Exercises . . . . .	190
8.5	An Application of the Eigendecomposition of a Symmetric Matrix . . . . .	191

<b>9</b>	<b>Least Squares with Matrix Algebra</b>	<b>193</b>
9.1	The Setup . . . . .	193
9.2	Ordinary Least Squares . . . . .	195
9.3	Algebraic Properties of OLS Estimators . . . . .	197
9.4	Statistical Properties of OLS Estimators. . . . .	199
9.5	Hypothesis Testing . . . . .	201
9.6	Asymptotic Properties . . . . .	206
9.7	Exercises . . . . .	212
9.8	Appendix . . . . .	215
<b>10</b>	<b>Instrumental Variables and Generalized Method of Moments</b>	<b>217</b>
10.1	Instrumental Variables and the IV Estimator . . . . .	217
10.2	A Simultaneous Equation Example . . . . .	219
10.2.1	A Two-Stage Least Squares Perspective . . . . .	220
10.3	Multiple Instruments . . . . .	223
10.4	Generalized Method of Moments . . . . .	227
10.4.1	Optimal GMM . . . . .	230
10.4.2	Hypothesis Testing after GMM . . . . .	233
10.4.3	GMM Estimation in Stata . . . . .	237
10.5	Exercises . . . . .	239
<b>11</b>	<b>Introduction to Time Series Regressions</b>	<b>241</b>
11.1	Overview . . . . .	242
11.2	Some Simple Time Series Models . . . . .	243
11.2.1	Transformations . . . . .	243
11.2.2	Sample Autocorrelation Function . . . . .	244
11.2.3	Trend . . . . .	245
11.2.4	Seasonality . . . . .	250
11.2.5	Cycles . . . . .	257
11.3	Time Series Regressions . . . . .	259
11.3.1	Dynamic Specifications . . . . .	259
11.3.2	Assumptions . . . . .	261
11.3.3	Standard Errors for Dynamically Incomplete Models . . . . .	262
11.3.4	Dynamically Complete Models . . . . .	265
11.3.5	Testing for Autocorrelation . . . . .	265
11.3.6	Regression with Trending and Persistent Series . . . . .	266
11.3.7	Spurious Regressions . . . . .	270
11.4	Exercises . . . . .	272
11.5	Appendix . . . . .	274
	<b>References</b>	<b>277</b>

## Preface

These notes were written to accompany the econometrics courses that I teach at the School of Economics, Singapore Management University (SMU):

- ECON207 Intermediate Econometrics (BSc Econ)
- ECON682 Econometric Analysis (Econometrics core for MSc Econ / MSc Fin. Econ.)
- ECON6001 Time Series Econometrics (MSc Economics - Quantitative Economics Track)

There are (will be) about 20 to 30 chapters in total; the specific chapters you will use are listed in your course outline.

## What is Econometrics?

Econometrics draws on statistics, economic theory, and mathematics to develop tools for estimating economic relationships, for the purposes of decision making, prediction and forecasting, inferring causal effects, evaluating the efficacy of policy interventions and initiatives, testing the validity of economic theories and their underlying assumptions, and answering a multitude of questions that are ultimately empirical in nature. Examples include:

- Pricing decisions by firms require knowledge of the price sensitivity of demand for their products. These are provided by estimates of the products' price elasticities of demand.
- Monetary authorities / central banks build empirical forecasting models of the economy to help anticipate outcomes such as high inflation or economic recessions and predict the outcome of potential policy responses.
- House prices that are very much higher than that predicted by an empirical model linking house prices to economic fundamentals may indicate imbalances in the economy that require policy intervention.
- There is a long list of public initiatives undertaken by authorities to encourage certain behaviors in people and firms, or to improve economic, health, educational and other outcomes in populations. To what extent do they work?
- Many theories in various fields such as industrial organization, economic growth, economic geography among others, assume constant returns to scale in production. Are such assumptions in line with empirical evidence, or would they fall when tested against data.
- Estimates of the economic effect of climate change must factor in adaptation by industries. While we can expect industries to at least try to adapt, is there evidence that they are able to do so effectively and quickly enough?

Such applications present many challenges. The challenge in forecasting applications is to find predictors that have stable relationships with the variable being forecast, and to determine and estimate the form of these relationships. In some cases there are many potential predictors, each limited in predictive ability on its own, but perhaps powerful in totality. The challenge is to estimate usable forecasting relationships with those predictors.

Causal inference – empirically teasing out causal relations from correlative ones – must deal with confounding effects. For example, the causal link from years of education to earnings is tangled up with the effects of individual characteristics such as ability, work experience at the

time of sampling, family background, among others things, all of which drive both earnings and the decision to pursue more years of education. Any attempt to interpret a correlation between years of education and earnings as a causal effect must somehow control for these factors. The ideal situation is if we could hold everything fixed apart from the candidate “causal” variable  $x$  and observe what happens to the ‘explained’ variable  $y$  when we change  $x$ , but of course this is impossible. What are the alternatives? In some applications, one might be able to employ a randomized controlled trial (RCT) wherein subjects are randomly assigned into a treatment group and a non-treatment group. The randomization breaks the link between the confounding characteristics and the treatment, and enables one to interpret the correlation between treatment and outcome as evidence of causation. In most cases, however, researchers have to depend on observational data, where information regarding a sample drawn from a population is observed without any intervention from the researcher. In these cases, clever methods must be devised to tease out causality from correlation.

Econometric methods must also take into consideration the data structures found in economic data – whether data is made up of a sample from a population taken at some point in time (we call this “cross-sectional data”), or several cross-sections resampled over multiple periods (“pooled cross-sections”) or the same cross-sectional sample re-observed over multiple periods (“panel or logistical data”), or observations of variables taken over multiple time periods (“time-series data”), and so on. In some applications, the researcher has to take special steps to counter the complications that arise because of this structure. In other examples, the features of certain data structures can be exploited to assist in empirical causal inference. Other data related issues include measurement error, and the fact that we often are only able to employ data that are, at best, proxies of the actual variables we would like to study.

Econometricians have always relied on computers to implement their formulas. This reliance has further increased as computer-based statistical methods – where algorithms have replaced formulas – have become more important over the past few decades. The econometrician now must add computing skills, in addition to economic theory, mathematics and statistics, to her list of competencies.

## Mathematical Prerequisites

I assume that the reader is able to do simple differentiation and integration. What we need of optimization theory is reviewed in a section in the mathematics review chapter. We will use a considerable amount of matrix algebra, and detailed notes are provided on this topic. We will, of course, use probability theory and statistics extensively. These notes include chapter-length reviews of both.

## Software

The computations in these notes were done in R. Data are available from your course webpages, and I assume these are stored in a ‘data’ folder in your working directory. There are many introductions to R on the web. I will proceed on the assumption that you have studied some of these, and that you have a working installation on your computer. I recommend running R within the RStudio Integrated Development Environment (IDE). The chapter “Introduction to R” contains brief instructions on installing R and RStudio, and a quick primer on using R.



# Chapter 1

## A Brief Introduction to R

### 1.1 Getting Set Up

First download and install the R software from The R Project for Statistical Computing website. Then download and install RStudio from the RStudio website (go to Products, RStudio under the Open Source tab, and download the Open Source Edition of RStudio Desktop). RStudio is an “integrated development environment” (IDE) comprising a set of programs that help you to develop and run R code. You do not need RStudio to run R, but this is what we will do.

When you first run RStudio, you will see an RStudio desktop open up with three or four windows. There will be one with tabs such as **Files**, **Plots**, **Packages**, **Help**, and **Viewer**, another window with tabs marked **Console**, **Terminal**, **Jobs**, and a third window with **Environments**, **History**, **Connections**, and other tabs. If you go to the menu and select **File>New File>R Script**, a fourth window will open up with an **Untitled1** tab. This is the Editor window, and the **Untitled1** tab is a blank **R Script** file.

You will issue commands in the **Console** tab, or write your instructions in a R script file and execute them from the Editor window. If you ask R to display the results of calculations, these will show up in the **Console** tab. Graphics produced will show up in the **Plots** tab. Objects created will be listed in the **Environment** tab. Executing commands from an R script is best practice; you can save the commands, modify them, correct errors and redo computations easily. Use the console to test commands, make inquiries of objects, and for other one-off actions.

**Activity:** Go to your console tab, type `x <- 4.5` and press enter. Type `x` and enter.

```
x <- 4.5
x

[1] 4.5
```

You have just used the **assignment operator** `<-` to create an object named `x` containing the number 4.5.<sup>1</sup> The object `x` now appears in your **Environment** tab. You can also use the `=` operator, but we almost always use `<-` for assignment and `=` when giving values to parameters in functions. The second line prints the value of `x` to your console.

If you make a calculation, say `2+2`, and don’t assign the result to a name, the result of that calculation is displayed in the console window, but is thereafter inaccessible, lost in computer memory until overwritten by R.

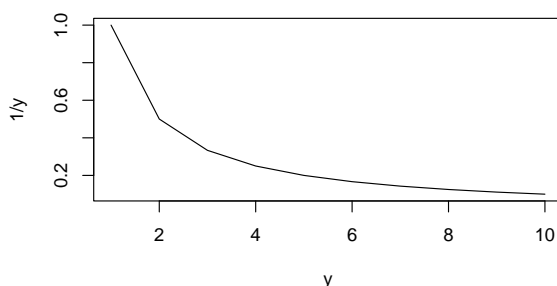
---


<sup>1</sup>Technically, R stores the value 4.5 as binary code somewhere in your computer, creates a name `x`, and a pointer linking the name to the memory location where the value is stored. But at this stage, you can think of it as having created an object named `x` with the value 4.5.

**Activity:** Open a new R Script in the Editor window (or use the ‘untitled’ script if it is already open), and type in the following lines.

```
# Any line or part of line following a '#' is ignored by R
# We use this feature to add comments to code

y <- seq(1,10)      # Create an integer sequence from 1 to 10
plot(y,1/y,type='l') # Create a line plot
```



On your Editor window, click on the **Source** button (alternatively, select all lines and hit **Ctrl+Enter**). This will cause all of the lines in the R script to run one after the other. A line plot will appear on your **Plots** tab. If you select **Export>Copy to Clipboard**, the plot appears in a pop-up window. If you click **Copy Plot**, the plot is placed on your clipboard and you can then paste the plot into, say, a Word document. New plots are placed over current plots. Use the arrows in the plot window to go back and forth between plots. Press the red circle with a white X to erase the current plot. Press the broom icon to erase all plots. To save your script, click on the floppy disk icon  and save the file with an appropriate name. The saved file will have the “.R” extension.

In R, you store your data in **vectors**, **matrices**, **lists**, **data frames** (and variants of it), and **time series** objects (and variants of it). These are different **data structures**, i.e., different ways that R can organize your data.

You will work with different kinds of **data types**, including **integer** (whole numbers, without decimals), **double** (or **floating-point** for numbers with decimals), **character** (for text data), **logical** (or **boolean**, to indicate TRUE or FALSE), and **complex** (for complex numbers). Numbers such as 1 and 2 can be stored either as integers or doubles. To force a whole number to be an integer, we append an L after the number, e.g., 1L. Integer and double are also collectively known as **numeric** data types. There is also a **factor** data type for categorical data.

All actions in R are carried out using **functions** such as **seq()** and **plot()**, and **operators** such as **<-** and **:** (operators are actually also functions). Functions in R are sets of instructions designed to perform certain tasks. Pre-written R functions are organized into **packages**. Every installation of R comes with some packages pre-installed (including the **base**, **datasets**, **graphics** and **stats** packages) and which are automatically loaded every time you start R. There are many other packages written by independent programmers that provide additional functionality and that are not pre-installed in R. To access these functions, you have first to

install the package into your R installation. You can then load the package into any R session that requires the functions in that package. Finally, you can write your own functions.<sup>2</sup>

To find out more about any given R function, enter `? function_name` into the console and the relevant documentation will come up, e.g.,

```
? seq # Enter this and see what happens
? `:` # With this sort of operators, you have to use surround them by backticks
```

As we mentioned earlier, objects that you create go into your “environment”. To see what is in your environment, use `ls()`. To remove all objects in your environment, use `rm(list=ls())`. Try the following line-by-line.

```
ls()          # ls ~ list objects
rm(list=ls()) # rm ~ remove. Environment is now clear.
```

Most of the data that you work with will be imported from an external files (`.csv`, `.xlsx`, etc.) and stored data frames. We will import some later. You will also want to “create” data within R. For instance, you may want to create a sequence of integers, or a value that will be used as a constant in your work, or generate a sequence of random numbers and so on.

## 1.2 Data Types

**Activity:** Run the following commands and queries as suggested, line by line.

```
## The following illustrate some of the major data types in R
a1 <- 1L          # Integer
a2 <- 4           # Integer or Double?
a3 <- 2.3         # Double
a4 <- TRUE        # Logical
a5 <- "Two"       # Character, making up a "string"
a6 <- "12"        # Another string
a7 <- 2+1i        # Complex

## Use typeof() to query the data type of the object
typeof(a1)        # Try with the others objects you created

## The is.integer(), is.double(), is.numeric(), is.character(), is.complex()
## functions make more specific queries as to the object's data type. An example
## is shown. Try each of the query function on each of the objects above.
is.integer(a2)

## In some cases, you can "coerce" R to change data types using functions like
## as.integer(), as.double(), as.numeric(), as.logical(), as.character(), as.complex().
## An example is shown below. Try these functions of each of the objects created so far.
as.integer(a4)

## Sometimes R will do the coercion for you
3 + TRUE
```

<sup>2</sup>We will use the packages `tidyverse` (Wickham et al. (2019)) for data management and plotting, `readxl` (Wickham and Bryan (2023)) for importing data, `car` (Fox and Weisberg (2019)) and `sandwich` (Zeileis, Köll, and Graham (2020)) for econometrics related algorithms, and `patchwork` (Pedersen (2023)), `latex2exp` (Meschiari (2023)), `gridExtra` (Auguie (2015)) and `plot3D` (Karline (2015)) for additional plotting functionality.

There are **special values** in R:

- **NA** stands for “Not Available” or “Missing”. It is by default a logical datatype, but can be converted to other data types.
- **NULL** is an empty object, with no datatype.
- **Inf** stands for “Infinity” and comes about when you do operations like  $1/0$ . It is by default a **numeric** datatype (specifically **double**, but coerce-able to **complex**).
- **NaN** stands for “Not a Number” and comes about when you do operations like **Inf-Inf**. It is, ironically, of **numeric** datatype by default (specifically **double**, coerce-able to **complex**).

The following are examples of how these values can arise, or how they may be used.

```
1/0      # This will give you Inf
Inf - Inf # Gives NaN. Inf - Inf is NOT equal to zero
0/0      # Also gives NaN. 0/0 is NOT equal to one. Please.
a <- NA   # Basically saying the data that's suppose to be there is missing
```

**NULL**, **NA**, **Inf** and **NaN** are reserved words. You cannot use them as names of objects. Other reserved words include: **if**, **else**, **while**, **repeat**, **for**, **next**, **in**, **function**, **break**, **TRUE**, **FALSE**.

Computers can only store real numbers up to some degree of accuracy:

**Activity:** The `sqrt()` function returns the square root of a number. Execute the following code. Do the results surprise you?

```
sqrt(2)
sqrt(2)*sqrt(2)
sqrt(2)*sqrt(2) == 2 # use == to make equality comparisons
sqrt(2)*sqrt(2) - 2
```

The **e-16** in the output stands for  $\times 10^{-16}$ . Computers, of course, cannot store irrational numbers to infinite accuracy, and this can lead to surprising results when making comparisons, or inaccurate results when performing complicated tasks that involve a very large number of calculations. For now, just bear this in mind. The degree of accuracy is generally not going to be an issue for us (except when making comparisons).

### 1.2.1 Arithmetic and Logical Operators

The **arithmetic operators** include: **Addition** (+), **Subtraction** (-), **Multiplication** (\*), **Division** (/), and **Exponent** (^). The usual **operator precedence** apply: operations in parentheses are evaluated first, followed by ^, followed by (\*,/), followed by (+/-). Ties between multiplication and division, and between addition and subtraction, are broken by evaluating from left to right. Always use parentheses when in doubt.

**Activity:** Enter  $8/2 * (2+2)$ . Do you agree with the result?

```
8 / 2 * (2+2)
```

You may recognize this from an internet meme, asking what is  $8 \div 2(2+2)$  and the answer depends on whether you treat  $2(2+2)$  as a single entity. If I say “4 divided by  $2n$ ” do I mean “4 divided by  $(2n)$ ” or “4 divided by 2, times  $n$ ”. I mean the former. In R, you cannot write  $8/2(2+2)$ , you have to write  $8/2*(2+2)$  which means 8 divided by 2 times 4.

The **relational operators** are:

- Less than <
- Greater than >
- Less than or equal to <=
- Greater than or equal to >=
- Equal to ==
- Not equal to !=

Comparisons using relational operators result in the logical outcomes TRUE or FALSE.

```
2 != 3
```

```
[1] TRUE
```

There are usually a number of different ways to make a comparison, e.g.,

```
2 != 3
!(2 == 3)
```

The ! is the logical operator “not”, or negation. The **logical operators** are:

- Logical Negation: !
- Logical And: &, &&
- Logical Or: |, ||

We will use & and | for now, and explain && and || later.

Let A and B be two statements, each of which are either true or false. If A is true, !A is false. If A is false, then !A is true. The statement A & B is true only if both are true. If one or both statements are false, then A & B is false. The statement A | B is true if one or both are true. If both statements are false, then A | B is false.

In mathematics and computer programming, “or” is always non-exclusive. A or B is true means either (i) A is true, (ii) B is true, or (iii) both are true. Also note that ‘and’ takes precedence over ‘or’, so the statement “A or B and C” means “A or (B and C)”. Question: will R evaluate the following as true or false?

```
# Is the following TRUE or FALSE
(1 < 2) | (2 < 3) & (4 < 2)
# What about this?
(1 < 2) | (4 < 2) & (6 < 5)
```

The order of precedence is, from highest to lowest: not, and, or, implies, equivalent to. Use parentheses to ensure the order is as you want it.

## 1.3 Data Structures

### 1.3.1 Vectors

The **vector** datatype is the most basic data structure in R. It is an ordered set of data items. Even single values are stored as a vector.

**Activity:** Earlier we created the data objects `x` and `y`. Enter the following commands one at a time, and study the outcome.

```
is.vector(x) # query if object is data type
length(x)    # how many items are in it?
typeof(x)    # what data type does it contain?
## Repeat the above with the object "y"
```

**Activity:** The following commands all produce vectors. Run the following lines one at a time. After each line output the variable to your console, and study them

```
b01 <- 5
b02 <- 1:26 # `:` ~ colon operator, gives integers from:to
b03 <- seq(from=2, to=15, by=2) # the seq() function is more flexible
b04 <- c(37, 42, 29, pi) # c() ~ "combine" things in a vector or list
b05 <- c("Q1", "Q2", "Q3", "Q4") # A piece of character data is a "string"
b06 <- rep(1, times=5) # rep() ~ "replicate". Can simply say rep(1,5)
b07 <- rep(b05, times=4) # what happens here?
b08 <- rep(1980:1983, each=4) # and here?
b09 <- c(1<2, 2==4, 4>=3, 1+1==2) # gives logical values!
b10 <- c(1+1i, 0+1i, 2+3i) # complex numbers!!
b11 <- letters # built-in vector, like "pi" in a04
b12 <- LETTERS # built-in vector
b13 <- month.abb # built-in vector
b14 <- month.name # built-in vector
```

Use `is.vector()` to verify that all the objects you just created are vectors. Use `typeof()` to check their data types. Use `is.integer()`, `is.double()`, `is.numeric()`, `is.character()`, `is.logical()`, `is.complex()` to query to data type of the elements of an object. For example:

```
is.vector(b01) # Should return TRUE
typeof(b05)    # Should return 'character'
is.double(b02) # Should return FALSE. R has opted to store these as integer.
is.integer(b02)
is.logical(b09)
```

A few things to remember about R vectors:

- In matrix algebra, we have row vectors and column vector:

$$\text{a row vector: } \begin{bmatrix} 1 & 2 & 4 & 8 \end{bmatrix} \quad \text{a column vector: } \begin{bmatrix} 1 \\ 2 \\ 4 \\ 8 \end{bmatrix}.$$

In R, vectors have no shape: they are simply ordered collections of data items, one item following another, but not organized into a row or a column. Vectors have length (here meaning “number of items”) but no dimension.

- There are no scalars. The object `b01` is just a vector (of length 1).
- Each vector can only hold data of a single datatype. You cannot mix datatypes in a vector

You access elements of a vector using the “extract and replace” operator `[]`.

**Activity:** What do the following do?

```
b12[2]           # indexing from R starts with 1. This returns the 2nd item in b.
b12[c(1,1,3)]    # returns the 1st, 1st, and 3rd items
b12[22:26]       # returns 22nd to 26th items
b12[-(1:3)]      # negative indices remove items. Cannot mix with positive indices

head(b12,5)      #
head(b12,-5)     # Frequently helpful if accessing the start or end of vectors
tail(b12,5)      # Check them out!
tail(b12,-5)     #

c01 <- 1:4        # A new vector
c02 <- c01[c(2,4)] # Copies 2nd and 4th elements of c01 into c02
c02              # Check it out.
c01[2] <- 20      # What does this do?
c01              # 2nd element of c01 has been changed,
c02              # but c02 is not changed. It is its own object.
```

**Activity:** What happens in the next activity is a bit tough to figure out. First find out about the `%` operator. Then try to figure out what the following lines do? Remember `b02` is 1, 2, ..., 26 and `b11` is a, b, ..., z. The point of this activity is to show that you can extract from a vector using logical values.

```
i <- !(b02 %% 2)      # First check out b02 %% 2, then check out !(b02 %% 2)
evenletters <- b11[i] # Then see what 'evenletters' is
```

You can also give names to the positions of elements in a vector, and access the elements by their position name.

**Activity:** Try the following.

```
names(b02) <- b12
b02
b02[c("A","C","D","C")]
```

Since a vector can hold data of one type only, if you attempt to mix data types in a vector, R will try to coerce the data types “upwards” – logicals become integers or higher, integers become doubles or higher, doubles become complex or higher, complex becomes character. In the first vector in the following example, we try to mix a logical, double, and complex values. The result is a complex vector. In the second case, we mix a logical with an integer and a character. The result is a character vector.

```
c1 <- c(F, 4.5, 1+1i)
c2 <- c(T, 1, "r")
```

### 1.3.2 Factor Datatype

The **factor** datatype is used for **categorical** variables. The following vectors contain the names of a sample of people, their ages, the region of Singapore they live in<sup>3</sup>, and birth month.

```
name <- c("Abe", "Ben", "Claire", "Daniel", "Edwin",
          "Fred", "Gina", "Harry", "Ivy", "Judy")
age <- c(16, 24, 16, 23, 25, 40, 33, 31, 31, 60)
region <- c("West", "North-East", "West", "Central", "East",
            "North-East", "West", "West", "East", "North")
bmonth <- c("Apr", "Jun", "Oct", "Jan", "Apr",
            "Sep", "Jun", "Jul", "Aug", "Apr")
```

Both `name`, `region`, and `bmonth` are currently character vectors. We can convert `region` into factor datatype.

```
region <- factor(region)
region
```

```
[1] West      North-East West      Central   East      North-East
[7] West      West      East      North
```

Levels: Central East North North-East West

We'll convert `bmonth` into an ordered factor data type:

```
bmonth <- factor(bmonth, levels=month.abb, ordered=TRUE) # remember what month.abb is?
bmonth
```

```
[1] Apr Jun Oct Jan Apr Sep Jun Jul Aug Apr
```

12 Levels: Jan < Feb < Mar < Apr < May < Jun < Jul < Aug < Sep < ... < Dec

### 1.3.3 Data Frames

Most of the time, you will store your data for analysis in a data structure called a **data frame**, or one of its variants. You can think of this as a rectangular “spreadsheet” of data, each column containing data on some variable, with different data types allowed per columns.

```
customers <- data.frame(Name=name, Age=age, Region=region, BMonth = bmonth)
customers
```

	Name	Age	Region	BMonth
1	Abe	16	West	Apr
2	Ben	24	North-East	Jun
3	Claire	16	West	Oct
4	Daniel	23	Central	Jan
5	Edwin	25	East	Apr
6	Fred	40	North-East	Sep
7	Gina	33	West	Jun
8	Harry	31	West	Jul
9	Ivy	31	East	Aug
10	Judy	60	North	Apr

---

<sup>3</sup>For purposes of urban planning, Singapore’s Urban Redevelopment Authority (URA) divides the country into five regions: Central, East, North, North-East and West. These are further subdivided into 55 planning areas.



You can access the contents of this data frame in various ways, illustrated below.

```
customers[1:3,] # All columns of the first three rows
```

	Name	Age	Region	BMonth
1	Abe	16	West	Apr
2	Ben	24	North-East	Jun
3	Claire	16	West	Oct

```
customers[age==16,c("Name", "BMonth")] # Name and Birth month of customers aged 16
```

	Name	BMonth
1	Abe	Apr
3	Claire	Oct

```
customers$Name[6:10] # Names of all customers 6 to 10
```

```
[1] "Fred" "Gina" "Harry" "Ivy" "Judy"
```

### 1.3.4 Matrices, Lists

Other useful data structures include **matrices** and **lists**. A matrix is a vector given a “dimension attribute.” The following code creates a matrix with two rows from a vector.

```
mat1 <- matrix(c(1,2,3,4,5,6), nrow=2)
mat1
```

	[,1]	[,2]	[,3]
[1,]	1	3	5
[2,]	2	4	6

```
attributes(mat1)
```

```
$dim
[1] 2 3
```

Notice that the matrix is filled up by columns. This is the default. To fill by rows, use the `byrow==TRUE` option

Lists are like vectors, except that you can have different data types *and* even different data structures in a list (including other lists). You access items in a list using `[[..]]`. In the following code, we create a list of six items, from previously defined objects.

```
mylist <- list(first=b01, second=b02, third=b03, fourth=b04, fifth=b05, sixth=mat1)
mylist
```

```
$first
[1] 5
```

```
$second
 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
```

```
$third
[1] 2 4 6 8 10 12 14
```

```
$fourth
[1] 37.000000 42.000000 29.000000 3.141593
```

```
$fifth
[1] "Q1" "Q2" "Q3" "Q4"
```

```
$sixth
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

We gave names to the items in the list when creating the list. This is optional. The following are some examples of how items in a list can be accessed.

```
mylist[[3]]

[1] 2 4 6 8 10 12 14
```

```
mylist[["second"]]

 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
```

```
mylist[[6]][1,1:2]

[1] 1 3
```

In the last example, `mylist[[3]]` returns a matrix, and `mylist[[3]][1,1:2]` returns the (1,1)th and (1,2)th items of this matrix.

### 1.3.5 Time Series

Another data structure is **time-series**, for holding data ordered in time. The following example converts a numerical vector of random numbers into a “quarterly” time series.

```
set.seed(13)      # for replicability, use own choice of integer
u <- runif(8)      # generates a vector of 8 numbers from a U(0,1) distribution
u.ts <- ts(u, start=c(2010,1), frequency = 4)
u.ts

      Qtr1      Qtr2      Qtr3      Qtr4
2010 0.71032245 0.24613730 0.38963444 0.09138367
2011 0.96206454 0.01093333 0.57429518 0.76439799
```

The `ts()` function converts a vector to time series. The `frequency=4` indicates that the data are quarterly (4 observations per year), and the `start` option then gives the starting period.

You can use the `class()` function to query an object as to its data structure.

```
class(name)      # For vectors, this function returns the datatype.
class(age)       #   E.g., class(age) returns "numeric" instead of "vector".
class(region)    #   You should read that as "age is 'a numeric vector'".
class(bmonth)
class(customers)
class(mat1)
class(mylist)
class(ts)
```

```
[1] "character"
[1] "numeric"
[1] "factor"
[1] "ordered" "factor"
[1] "data.frame"
[1] "matrix" "array"
[1] "list"
[1] "function"
```

The `class()` function returns the “class” attribute which identifies the data structure of the object. You should see what you get when you apply the `attribute()` function to the objects listed above, for example:

```
attributes(region)
```

```
$levels
[1] "Central" "East" "North" "North-East" "West"

$class
[1] "factor"
```

Notice that `class(m)` returns "matrix" "array". An R array is a data structure with more than two dimensions. Matrices are 2-dimensional arrays.

## 1.4 Importing Data

Most of the time, we will read in our data from an external file.

**Example 1.1.** I assume you have the data set **Anscombe.xlsx** (available on course website) stored in a ‘data’ sub-folder of your working directory. We will use the function `read_excel()` from the package `readxl` to read in the data. If the package has not yet been installed, install it with the command

```
install.packages("readxl") # don't forget the quotes
```

You only have to do install a package once (unless you want to update it). Thereafter, just load the package with `library()` whenever you want to use the functions in this package.

```
library(readxl) # No quotes!
```

Now we read in the data:

```
df2 <- read_excel("data\\Anscombe.xlsx")
df2
```

```
# A tibble: 11 x 8
```

	x1	y1	x2	y2	x3	y3	x4	y4
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	10	8.04	10	9.14	10	7.46	8	6.58
2	8	6.95	8	8.14	8	6.77	8	5.76
3	13	7.58	13	8.74	13	12.7	8	7.71
4	9	8.81	9	8.77	9	7.11	8	8.84
5	11	8.33	11	9.26	11	7.81	8	8.47
6	14	9.96	14	8.1	14	8.84	8	7.04
7	6	7.24	6	6.13	6	6.08	8	5.25
8	4	4.26	4	3.1	4	5.39	19	12.5
9	12	10.8	12	9.13	12	8.15	8	5.56
10	7	4.82	7	7.26	7	6.42	8	7.91
11	5	5.68	5	4.74	5	5.73	8	6.89

Investigating a large data frame by simply printing it out to screen is not feasible. You can use `head()` and `tail()` to print only the first few or last few observations. Alternatively, you can use `str()` to give you a summary of the data frame (`str` = structure).

```
head(df2,3)
```

```
# A tibble: 3 x 8
```

	x1	y1	x2	y2	x3	y3	x4	y4
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	10	8.04	10	9.14	10	7.46	8	6.58
2	8	6.95	8	8.14	8	6.77	8	5.76
3	13	7.58	13	8.74	13	12.7	8	7.71

```
tail(df2,3)
```

```
# A tibble: 3 x 8
```

	x1	y1	x2	y2	x3	y3	x4	y4
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	12	10.8	12	9.13	12	8.15	8	5.56
2	7	4.82	7	7.26	7	6.42	8	7.91
3	5	5.68	5	4.74	5	5.73	8	6.89

```
str(df2)
```

```
tibble [11 x 8] (S3: tbl_df/tbl/data.frame)
 $ x1: num [1:11] 10 8 13 9 11 14 6 4 12 7 ...
 $ y1: num [1:11] 8.04 6.95 7.58 8.81 8.33 ...
 $ x2: num [1:11] 10 8 13 9 11 14 6 4 12 7 ...
 $ y2: num [1:11] 9.14 8.14 8.74 8.77 9.26 8.1 6.13 3.1 9.13 7.26 ...
 $ x3: num [1:11] 10 8 13 9 11 14 6 4 12 7 ...
 $ y3: num [1:11] 7.46 6.77 12.74 7.11 7.81 ...
 $ x4: num [1:11] 8 8 8 8 8 8 8 19 8 8 ...
 $ y4: num [1:11] 6.58 5.76 7.71 8.84 8.47 7.04 5.25 12.5 5.56 7.91 ...
```

The `read_excel()` function reads data into a modified data frame called a `tibble`. This modification is part of the larger “tidyverse” initiative. For the moment, we can treat the two data structures (tibble vs data frame) as essentially the same thing. We will use the tidyverse suite of packages for data wrangling, and for graphics.

## 1.5 Plotting Data

R comes with a very good base graphics package pre-installed (and automatically loaded whenever you start an R session). We used the `plot()` function from this package earlier. There is another package called `ggplot2` that contains many functions for producing very good graphics (`gg` = Grammar of Graphics). We will use both in these notes, but for now we use the latter.

You can install the `ggplot2` package separately, but we will instead install the `tidyverse` package which includes several packages, `ggplot2` being one of them.

```
install.packages("tidyverse") # don't forget the quotes
```

Once the tidyverse package is installed, you can load it into your R session if you need to use it. Remember you don’t need to re-install packages once you have done so (unless you are updating the package). However, you do need to load the package every time you start an R session, should you be planning to use the functions in that package in the session.

```
library(tidyverse) # no quotes!
```

```
-- Attaching packages ----- tidyverse 1.3.2 --
v ggplot2 3.4.0      v purrr   0.3.5
v tibble  3.1.7      v dplyr   1.0.10
v tidyr   1.2.0      v stringr 1.4.0
v readr   2.1.2      v forcats 0.5.2
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

In addition to `ggplot2`, there are other packages that are helpful for constructing plots. One such package used in this book is `patchwork`. We assume you have already installed this package. In the example below, we use these `ggplot` and `patchwork` to plot the data that we just imported into R.

```
library(patchwork)
p1 <- df2 %>% ggplot() + geom_point(aes(x=x1, y=y1), size=1) + theme_classic()
p2 <- df2 %>% ggplot() + geom_point(aes(x=x1, y=y1), size=1) + theme_classic()
p3 <- df2 %>% ggplot() + geom_point(aes(x=x3, y=y3), size=1) + theme_classic()
p4 <- df2 %>% ggplot() + geom_point(aes(x=x4, y=y4), size=1) + theme_classic()
(p1 | p2) / (p3 | p4) # this is from patchwork package
```

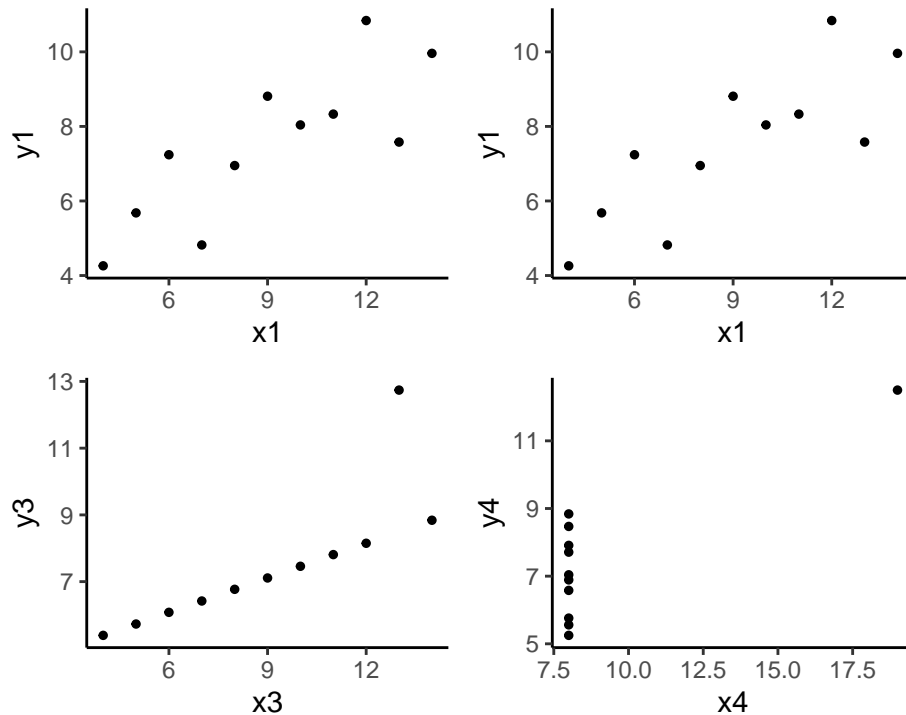


Figure 1.1: Anscombe Quartet.

In the code above, we created four separate figures, named `p1`, `p2`, `p3`, `p4` and used the `patchwork` library to create a composite figure comprising the four plots. When creating the individual scatterplots, we used the **pipe operator** `%>%` to “send” the dataframe / tibble to the `ggplot` function, and then ‘added’ a scatterplot with the `geom_point()` function. The `aes` option (which stands for aesthetics) is used to indicate the x-variable, y-variable, color-variable, and so on. The `theme_classic()` function is used to create a certain “look” for the plots.

The pipe operator `%>%` is helpful when doing several things to a data frame in sequence, and can help create very readable code. This operator is not part of base R, but is provided by the package `magrittr` which is included in the `dplyr` package which is included in the `tidyverse` package.

## 1.6 More on the R Environment

In your Environment tab, look for the menu button marked “Global Environment” and click on the little black triangle on the right of it. You will see a large list of “environments”, most of which are packages that were loaded in your R session, either automatically or by yourself using the `library()` command. The “Global Environment”, which contains all the variables that you created in your session, is always first. The packages are ordered as they were loaded (latest on top). To see all the functions in a loaded package, say the package `ggplot2`, you can use the command `ls("package:ggplot2")`. Just entering `ls()` will list the contents of the Global Environment.

One issue that you should pay attention to is ‘masking’. When we loaded the `tidyverse` package we saw two warnings: that `dplyr::filter()` masks `stats::filter()` and `dplyr::lag()` masks `stats::lag()`. Both `dplyr` and `stats` packages have a `lag()` function. Because the `dplyr`

package was loaded on top of the `stats` package, the `dplyr` version ‘masks’ the `stats` version, and calling `lag()` will call the `dplyr` version. However, the two versions behave differently: the `stats` version requires the input to be a time series object, whereas the input to the `dplyr` version *cannot* be a time series object. Worse, `lag(x,1)` in one means something quite different from `lag(x,1)` in the other. We illustrate this issue in the next example. To be explicit about which version you wish to use, indicate the package using `::`, as in `stats::lag()`.

**Example 1.2.** In this example, we create a vector `1:10`, and convert it into a **time series object** from 2019Q1 to 2021Q2. We then apply the `dplyr` version to the vector, and the `stats` version to the time series.

```
x <- 1:10
x
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

```
lag(x,1) # dplyr version is used
```

```
[1] NA 1 2 3 4 5 6 7 8 9
```

```
x.ts <- ts(1:10, start=c(2019,1), frequency=4)
x.ts
```

	Qtr1	Qtr2	Qtr3	Qtr4
2019	1	2	3	4
2020	5	6	7	8
2021	9	10		

```
stats::lag(x.ts,1)
```

	Qtr1	Qtr2	Qtr3	Qtr4
2018				1
2019	2	3	4	5
2020	6	7	8	9
2021	10			

We see that the `dplyr` version lags the data whereas the `stats` version creates a “leading” series. To use the `stats` version to lag, we have to say `stats::lag(x.ts,-1)`.

## 1.7 User-Defined Functions, Conditional Statements, Loops

You can define your own functions.

**Example 1.3.** A one-line function to calculate the area of a circle.

```
area_circle <- function(r){pi*r^2}
area_circle(2)
```

```
[1] 12.56637
```

**Example 1.4.** A more complicated function

```

circle_summary <- function(r=1){
  if (!is.numeric(r)){
    stop("Error: Input is not numeric.")
  } else if (r<=0 | is.nan(r) | is.infinite(r)) {
    print("Error: Please input a positive finite value for the radius.")
    return(NULL)
  } else {
    result = list("radius" = r, "area"=pi*r^2, "circumference"=2*pi*r)
    return(result)
  }
}

```

When the set of instructions is executed, a function object named `circle_summary` appears in your environment. Thereafter we can call it whenever we want to use it.

```

A1 = circle_summary(); A1    # radius defaults to 1

$radius
[1] 1

$area
[1] 3.141593

$circumference
[1] 6.283185

A2 = circle_summary(2); A2;

$radius
[1] 2

$area
[1] 12.56637

$circumference
[1] 12.56637

A3 = circle_summary(-1); A3

[1] "Error: Please input a positive finite value for the radius."
NULL

A4 = circle_summary("two"); A4

```

```

Error in circle_summary("two"): Error: Input is not numeric.
Error in eval(expr, envir, enclos): object 'A4' not found

```

The `circle_summary()` function requires one input `r`, which has the default value of 1. The function also contains “if-else” statements that carry out the following conditional actions:

- check if you put in a non-numeric value;
- if you did, print a error message and stop;



- If you did not input a non-numeric, check if it is negative or `NaN` or `Inf`;
- If so, print a different error message and return `NULL` (but don't stop the program);
- If the numeric value is not negative and not `NaN` and not `Inf`, then return a list comprising the radius, area and circumference of the circle.

Blocks of code are bound with “{...}”. The way we placed the braces is somewhat conventional. Indentations and writing long commands over several lines also help with readability.

Every function call has its “own namespace”:

**Example 1.5.** In the following example, the assignment of the value 3 to `x` inside the function does not change the value of `x` outside of the function.

```
an_example_function <- function(x){
  cat("x =", x, "was passed into the function.\n")
  x <- 3;
  cat("The function changes the value to: x = ", x, ".\n", sep="")
}
x <- 1
cat("The declared value of x: x = ", x, ".\n", sep="")
an_example_function(x)
cat("The value outside the function remains unchanged: x = ", x, ".\n")
```

The declared value of x: x = 1.

x = 1 was passed into the function.

The function changes the value to: x = 3.

The value outside the function remains unchanged: x = 1 .

We use the function `cat()` to print to screen (cat == “concatenate and print”). The special code “\n” refers to a line break. The function automatically adds a space between entries. To tell the function not to add the space, set the option `sep=""`.

Another essential programming technique is the “for-loop”. The following code, which contains a loop and a nested loop, illustrates how they work.

**Example 1.6.** Can you figure out what is going on in the program below?

```
for (A in c(TRUE,FALSE)){
  for (B in c(TRUE,FALSE)){
    cat("A is",A,"and B is",B,"then A & B is",A & B, "\n")
  }
}
```

A is TRUE and B is TRUE then A & B is TRUE

A is TRUE and B is FALSE then A & B is FALSE

A is FALSE and B is TRUE then A & B is FALSE

A is FALSE and B is FALSE then A & B is FALSE

Finally, we illustrate the “while” loop:

```
x <- 0
while (x<10){
  cat("x = ", x, ", x < 10 is ", x<10, " so we enter the loop.\n", sep="")
  x=x+2 # this means replace current value of x with current value + 2
}
cat("x = ", x, ", x < 10 is ", x<10, " so we skip the loop.\n", sep="")
```

```
x = 0, x < 10 is TRUE so we enter the loop,
x = 2, x < 10 is TRUE so we enter the loop,
x = 4, x < 10 is TRUE so we enter the loop,
x = 6, x < 10 is TRUE so we enter the loop,
x = 8, x < 10 is TRUE so we enter the loop,
x = 10, x < 10 is FALSE so we skip the loop.
```

Can you see the danger of inadvertently entering an infinite loop?

## Chapter 2

### Miscellaneous Mathematics Topics

We briefly review some math prerequisites, specifically: how to use the summation notation, an introduction to matrices, and a little optimization theory. We follow up on matrix algebra in later chapters. The R code uses the `tidyverse` and `patchwork` libraries.

```
library(tidyverse)
library(patchwork) # for laying out plots
```

#### 2.1 The Summation Notation

We use the uppercase sigma “ $\Sigma$ ” in the following way to denote summation. For a set of numbers  $\{x_1, x_2, \dots, x_n\}$ , define

$$\sum_{i=1}^n x_i = x_1 + x_2 + \dots + x_n.$$

**Example 2.1.** The **sample mean** of a set of numbers  $\{x_1, x_2, \dots, x_n\}$  is

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i .$$

**Example 2.2.** Write  $4 + 8 + 12 + 16 + 20 + 24$  in summation notation. Ans:  $\sum_{i=1}^6 4i$ .

**Example 2.3.** The **present value** of a future amount of money is the amount today that, if invested at a certain rate, would return that future sum. Suppose the following payments are to be made:  $a_1$  at the end of the first period,  $a_2$  at the end of the second period, and so on, for  $n$  periods. At a fixed interest rate of  $r$  per period, the present value of the payments is

$$\frac{a_1}{1+r} + \frac{a_2}{(1+r)^2} + \dots + \frac{a_n}{(1+r)^n} = \sum_{i=1}^n \frac{a_i}{(1+r)^i} .$$

**Example 2.4.**  $\sum_{i=1}^n c = \underbrace{c + c + \dots + c}_{n \text{ terms, one for each } i} = nc$ .

In the first example, the **index of summation**  $i$  enters as subscripts that identify the terms of the summation, but otherwise does not enter into the computation of the sum. In the second example, the value of the index is used in the computation of the terms. In the third example, the index is used both ways. In the fourth example there is no index in the terms. We run through  $i = 1$  to  $n$  regardless.

**Example 2.5.** Let  $i = 1, 2, \dots, n$  represent a “basket” of goods, and

- $q_{0i}$  be the quantity of good  $i$  purchased in period 0 (the “base year”),
- $p_{0i}$  be the price of good  $i$  in the base year,
- $p_{ti}$  be the price of good  $i$  in period  $t$ .

The **Laspeyres price index** is defined as

$$Laspeyres_t = \frac{\sum_{i=1}^n p_{ti} q_{0i}}{\sum_{i=1}^n p_{0i} q_{0i}}.$$

In other words, the Laspeyres price index tracks the relative cost over time of a bundle of goods put together in the base year. The Consumer Price Index uses this methodology. For details, see International Monetary Fund et al. (2020). An alternative index is the “**Paasche Price Index**” which is defined as

$$Paasche_t = \frac{\sum_{i=1}^n p_{ti} q_{ti}}{\sum_{i=1}^n p_{0i} q_{ti}}.$$

where  $q_{ti}$  is the quantity of good  $i$  purchased in period  $t$ . The intention is to track the cost of an evolving basket of goods. It has the disadvantage that the basket of goods has to be redefined each period, which can be an expensive exercise if the basket is intended to be reflective of the quantities of goods consumed by a representative member of an economy. An advantage is that the updated quantities would reflect the effects of the price changes.

Expressions using the summation notation are not unique; more than one expression can be used to represent any given sum.

**Example 2.6.** Write  $1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11}$  in summation notation.

$$\text{Ans: } \sum_{i=1}^6 (-1)^{i-1} \frac{1}{2i-1}. \text{ Alternative Ans: } \sum_{i=0}^5 (-1)^i \frac{1}{2i+1}.$$

### 2.1.1 Rules for Summation Notation

The summation notation greatly simplifies notation but this is only helpful if you know how to manipulate expressions written with it. There are only two rules to learn:

- $\sum_{i=1}^n (a_i + b_i) = \sum_{i=1}^n a_i + \sum_{i=1}^n b_i$ ,
- $\sum_{i=1}^n (ca_i) = c \sum_{i=1}^n a_i$ , where  $c$  is some constant.

**Example 2.7.**  $\sum_{i=1}^n (x_i - \bar{x}) = 0$ , where  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ .

*Proof:*  $\sum_{i=1}^n (x_i - \bar{x}) = \sum_{i=1}^n x_i - \sum_{i=1}^n \bar{x} = n\bar{x} - n\bar{x} = 0$ .

That is, the sum of deviations of any set of numbers from its sample mean is always zero.

**Example 2.8.** Given  $n$  pairs of numbers  $(x_i, y_i)$ ,  $i = 1, 2, \dots, n$ , we have

$$\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^n (x_i - \bar{x})y_i = \sum_{i=1}^n x_i(y_i - \bar{y}). \quad (2.1)$$

*Proof:* For the first equality in (2.1), we have

$$\begin{aligned} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) &= \sum_{i=1}^n (x_i - \bar{x})y_i - \sum_{i=1}^n (x_i - \bar{x})\bar{y} \\ &= \sum_{i=1}^n (x_i - \bar{x})y_i - \bar{y} \underbrace{\sum_{i=1}^n (x_i - \bar{x})}_{=0} \\ &= \sum_{i=1}^n (x_i - \bar{x})y_i. \end{aligned}$$

The second equality (2.1) can be shown in similar fashion.

The sum in (2.1) appears in the **sample covariance** of observations  $\{x_i, y_i\}_{i=1}^n$  of two variables, defined as

$$s_{xy} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}).$$

If for each observation  $i$ ,  $x_i$  and  $y_i$  tend to be either both above or both below their respective sample means, then the product  $(x_i - \bar{x})(y_i - \bar{y})$  will be positive for most of the observations, and  $s_{xy}$  will likely be positive. If the variables tend to appear on opposite sides of their respective means, then  $s_{xy}$  will likely be negative. If there are no tendencies in the relative sizes of the two variables, then  $s_{xy}$  should be close to zero. We will explain in a later chapter why the sum is divided by  $n-1$  and not  $n$ .

### 2.1.2 Some Useful Formulas

For every positive integer  $n$ , we have

$$\begin{aligned} \sum_{i=1}^n i &= 1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2} \\ \sum_{i=1}^n i^2 &= 1^2 + 2^2 + 3^2 + \dots + n^2 = \frac{n(n+1)(2n+1)}{6} \\ \sum_{i=1}^n i^3 &= 1^3 + 2^3 + 3^3 + \dots + n^3 = \left( \sum_{i=1}^n i \right)^2 \end{aligned}$$

These can be proven by induction, or derived directly. We can get the first equation from

$$\begin{array}{ccccccc} 2 \sum_{i=1}^n i & = & 1 & + & 2 & + & \dots & + & n \\ & & + & n & + & (n-1) & + & \dots & + & 1 & = & n(n+1). \end{array}$$

For  $\sum_{i=1}^n i^2$ , we can use the fact that  $i^3 - (i-1)^3 = 3i^2 - 3i + 1$ . Summing both sides over  $i = 1, 2, \dots, n$  gives

$$\sum_{i=1}^n (i^3 - (i-1)^3) = 3 \sum_{i=1}^n i^2 - 3 \sum_{i=1}^n i + \sum_{i=1}^n 1.$$

The “telescopic sum” on the left hand side adds to  $n^3$ , therefore

$$n^3 = 3 \sum_{i=1}^n i^2 - 3 \sum_{i=1}^n i + \sum_{i=1}^n 1 = 3 \sum_{i=1}^n i^2 - 3 \frac{n(n+1)}{2} + n,$$

which can be solved for  $\sum_{i=1}^n i^2$ . The same trick can be used for  $\sum_{i=1}^n i^3$ : sum

$$i^4 - (i-1)^4 = 4i^3 - 6i^2 + 4i - 1$$

from  $i = 1$  to  $n$  on both sides to get

$$n^4 = 4 \sum_{i=1}^n i^3 - 6 \sum_{i=1}^n i^2 + 4 \sum_{i=1}^n i - n,$$

then plug in the formulas for  $\sum_{i=1}^n i^2$  and  $\sum_{i=1}^n i$ , and solve for  $\sum_{i=1}^n i^3$ . This trick can be used recursively to obtain expressions for  $\sum_{i=1}^n i^4$ ,  $\sum_{i=1}^n i^5$ , etc.

*Arithmetic Series:*

$$\sum_{i=1}^n (a + (i-1)d) = na + d \sum_{i=1}^{n-1} i = na + \frac{n(n-1)d}{2}. \quad (2.2)$$

*Geometric Series:*

$$\sum_{i=1}^n ar^{i-1} = a + ar + ar^2 + \dots + ar^{n-1} = \frac{a(1-r^n)}{1-r}. \quad (2.3)$$

To derive (2.3), let  $S = \sum_{i=1}^n ar^{i-1}$ . We have

$$\begin{aligned} S &= a + ar + ar^2 + \dots + ar^{n-1} \\ rS &= ar + ar^2 + \dots + ar^{n-1} + ar^n. \end{aligned}$$

Subtracting the second equation from the first gives  $(1-r)S = a(1-r^n)$  which you can solve for  $S$ .

*The Binomial Formula* For any integer  $n$ ,

$$\begin{aligned} (a+b)^n &= \binom{n}{0} a^n b^0 + \binom{n}{1} a^{n-1} b^1 + \dots + \binom{n}{n-1} a^1 b^{n-1} + \binom{n}{n} a^0 b^n \\ &= \sum_{i=0}^n \binom{n}{i} a^{n-i} b^i \end{aligned}$$

where

$$\binom{n}{i} = \frac{n!}{(n-i)!i!}.$$

The binomial formula can be proven by induction. It certainly holds for  $n = 1$ . For the induction step, we want to show that if the formula holds for  $n$ , then it also holds for  $n + 1$ . We have

$$\begin{aligned}
 (a + b)^{n+1} &= (a + b)^n(a + b) \\
 &= \binom{n}{0}a^{n+1}b^0 + \binom{n}{1}a^nb^1 + \binom{n}{2}a^{n-1}b^2 + \cdots + \binom{n}{n-1}a^2b^{n-1} + \binom{n}{n}a^1b^n \\
 &+ \binom{n}{0}a^nb^1 + \binom{n}{1}a^{n-1}b^2 + \cdots + \binom{n}{n-2}a^2b^{n-1} + \binom{n}{n-1}a^1b^n + \binom{n}{n}a^0b^{n+1} \\
 &= \binom{n}{0}a^{n+1}b^0 + \left[\binom{n}{1} + \binom{n}{0}\right]a^nb^1 + \left[\binom{n}{2} + \binom{n}{1}\right]a^{n-1}b^2 + \cdots \\
 &\quad + \left[\binom{n}{n-1} + \binom{n}{n-2}\right]a^2b^{n-1} + \left[\binom{n}{n} + \binom{n}{n-1}\right]a^1b^n + \binom{n}{n}a^0b^{n+1}.
 \end{aligned}$$

The binomial formula for  $n + 1$  follows from

$$\binom{n}{0} = \binom{n+1}{0} = 1 = \binom{n}{n} = \binom{n+1}{n+1}$$

and (see exercises)

$$\binom{n}{i} + \binom{n}{i-1} = \binom{n+1}{i}, \quad i = 1, 2, \dots, n.$$

### 2.1.3 Double Summations

Suppose we want to write the sum  $S$  of all of the terms in the rectangular block of numbers below using summation notation:

$$\begin{array}{cccc}
 a_{11} & a_{12} & \cdots & a_{1n} \\
 a_{21} & a_{22} & \cdots & a_{2n} \\
 \vdots & \vdots & \ddots & \vdots \\
 a_{m1} & a_{m2} & \cdots & a_{mn}
 \end{array}$$

We can first add up each row, then add up the row totals:

$$S = \sum_{j=1}^n a_{1j} + \sum_{j=1}^n a_{2j} + \cdots + \sum_{j=1}^n a_{mj} = \sum_{i=1}^m \left( \sum_{j=1}^n a_{ij} \right).$$

Alternatively, we can add up the columns, then add up the column totals, i.e.,

$$S = \sum_{i=1}^m a_{i1} + \sum_{i=1}^m a_{i2} + \cdots + \sum_{i=1}^m a_{in} = \sum_{j=1}^n \left( \sum_{i=1}^m a_{ij} \right).$$

Obviously it doesn't matter which approach we take. In other words, the order of summation does not matter. The parentheses make it clear which summation is to be done first, but we can leave out the parentheses and just write

$$\sum_{i=1}^m \sum_{j=1}^n a_{ij} \quad \text{or} \quad \sum_{j=1}^n \sum_{i=1}^m a_{ij}$$

with the understanding that the summations are carried out from the inner summation to the outer summation.

**Example 2.9.** Expand  $\sum_{i=1}^m \sum_{j=1}^n ij^2$ .

*Solution:*

$$\begin{aligned} \sum_{i=1}^m \left( \sum_{j=1}^n ij^2 \right) &= \sum_{i=1}^m \left( i \sum_{j=1}^n j^2 \right) \\ &= \left( \sum_{i=1}^m i \right) \left( \sum_{j=1}^n j^2 \right) \\ &= (1 + 2 + \cdots + m)(1^2 + 2^2 + \cdots + n^2). \end{aligned}$$

We cannot interchange the order of summation if the limits of the inner summation depend on the index of the outer summation.

**Example 2.10.** Suppose we have the triangular array of numbers below:

$$\begin{array}{cccc} a_{11} & & & \\ a_{21} & a_{22} & & \\ \vdots & \vdots & \ddots & \\ a_{m1} & a_{m2} & \cdots & a_{mm} \end{array}$$

We can write the sum of the elements of this array as

$$\sum_{i=1}^m \sum_{j=1}^i a_{ij}.$$

We added up the rows first, then added up the total. In this example, we cannot interchange the order of summation because the inner upper limit depends on the index of the outer summation; the expression  $\sum_{j=1}^i \sum_{i=1}^m a_{ij}$  simply makes no sense. If we want to add up the columns first, and then add up the total, we would write  $\sum_{j=1}^m \sum_{i=j}^m a_{ij}$ .

**Example 2.11.** Suppose we have the triangular array of numbers below:

$$\begin{array}{cccc} a_{11} & & & \\ a_{21} & a_{22} & & \\ \vdots & \vdots & \ddots & \\ a_{m1} & a_{m2} & \cdots & a_{mm} \\ \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{array}$$

where  $n \geq m$ . We can write the sum of the elements of this array as

$$\sum_{j=1}^m \sum_{i=j}^n a_{ij}.$$

#### 2.1.4 Exercises

In all of the exercises,  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$  and  $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ .

**Exercise 2.1.** Write  $2 + 3/2 + 4/3 + 5/4 + 6/5$  using the summation notation in two ways:

- with the index of summation  $i$  starting at 1,
- with the index of summation  $i$  starting at 2.



**Exercise 2.2.** Show that

- $\sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n (x_i - \bar{x})x_i = \sum_{i=1}^n x_i^2 - n\bar{x}^2$ ,
- $\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^n x_i y_i - n\bar{x}\bar{y}$ ,
- $\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^n x_i y_i$  if  $\bar{x} = 0$  or  $\bar{y} = 0$  (or both),
- $\sum_{i=1}^n (x_i - \bar{x})(x_i - 1) = \sum_{i=1}^n (x_i - \bar{x})(x_i - 1000000)$ .

**Exercise 2.3.** Prove that

- $\sum_{i=1}^n i^3 = \left(\sum_{i=1}^n i\right)^2$  using the identity

$$i^4 - (i-1)^4 = 4i^3 - 6i^2 + 4i - 1.$$

$$\text{b. } \sum_{i=1}^n i^4 = \frac{n(n+1)(2n+1)(3n^2+3n-1)}{30}.$$

**Exercise 2.4.** Show that for  $i = 1, 2, \dots, n$ , we have

$$\binom{n}{i} + \binom{n}{i-1} = \binom{n+1}{i}.$$

**Exercise 2.5.** Run the R code below, with your choice of numbers in the `x` vector. You can put whatever numbers you want, and as many of them as you want. What is the value of `sum(x-mean(x))`?

```
x <- c(pi, 2, 4.2, 54, 12.1212, 16)
sum(x-mean(x))
```

## 2.2 An Introduction to Matrices

### 2.2.1 Definitions and Notation

A **matrix** is a rectangular collection of numbers. The following is a matrix with  $m$  rows and  $n$  columns:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

Such a matrix is said to have “dimension”  $(m \times n)$ . The number that appears in the  $(i, j)$ th position, i.e., in the  $i$ th row and  $j$ th column, is called the  $(i, j)$ th element/entry/component of the matrix. We count rows from top to bottom, and columns from left to right.

- If  $m = n$ , the matrix is a **square matrix**,
- If  $m = 1$  and  $n > 1$ , we have a **row vector**,
- If  $m > 1$  and  $n = 1$ , we have a **column vector**.

The term “vector” is used in many ways in mathematics. Sometimes a vector refers to an ordered list of numbers  $(x_1, x_2, \dots, x_n)$ . Such an object has no dimension. It is merely an ordered sequence of length  $n$ . Column and row vectors, on the other hand, are two-dimensional objects. In the context of matrix algebra, the word “vector” alone usually means a column vector, but not always.

- If  $m = n = 1$ , then we have a **scalar**.

**Example 2.12.** A row vector  $c = [c_1 \ c_2 \ \cdots \ c_n]$ , a column vector  $b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$ .

A  $(2 \times 2)$  square matrix  $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ .

Matrices and vectors are often written in bold lettering, or with some sort of mark to distinguish them from scalars and other objects. We will not do so in these notes, and the reader will have to rely on context to distinguish scalars from vectors and matrices. Where context is unclear, we will be more explicit. Some additional notation:

- It is often convenient to indicate an  $(m \times n)$  matrix by  $(a_{ij})_{m \times n}$ .
- To refer to the  $(i, j)$ th element of a matrix  $A$ , we sometimes write  $[A]_{ij}$ .

Two matrices of the same dimension  $(m \times n)$  are said to be equal if each of their corresponding elements are equal, i.e.,

$$A = B \Leftrightarrow [A]_{ij} = [B]_{ij} \text{ for all } i = 1, 2, \dots, m; j = 1, 2, \dots, n.$$

Two matrices of different dimensions cannot be equal. A **zero matrix** is one whose elements are all zero. It is simply written as 0 although sometimes subscripts are added to indicate the dimension of the zero matrix.

The **diagonal** elements of an  $(n \times n)$  square matrix refer to the  $(i, i)$ th elements of the matrix, i.e., to the elements  $[A]_{ii}$ ,  $i = 1, 2, \dots, n$ . A **diagonal matrix** is a square matrix with off-diagonal elements equal to zero, i.e., a square matrix  $A$  is diagonal if  $[A]_{ij} = 0$  for all  $i \neq j$ ,  $i, j = 1, 2, \dots, n$ . Diagonal matrices are sometimes written  $\text{diag}(a_1, a_2, \dots, a_n)$ .

**Example 2.13.** The matrix

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \text{diag}(1, 4, 0)$$

is a diagonal matrix. Note that there is nothing in the definition of a diagonal matrix that says its diagonal elements cannot be zero.

An **identity matrix** is a square matrix with diagonal elements equal to one and off-diagonal elements equal to zero, i.e.,

$$I_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}.$$

An identity matrix is always written  $I$ . A subscript is sometimes added to indicate its dimension, although this is often left out. We will see shortly that the identity matrix plays a role in matrix algebra akin to the role played by the number “1” in the real number system.

A **symmetric matrix** is a square matrix  $A$  such that  $[A]_{ij} = [A]_{ji}$  for all  $i, j = 1, 2, \dots, n$ .

**Example 2.14.** The matrix  $A = \begin{bmatrix} 1 & 3 & 2 \\ 3 & 4 & 6 \\ 2 & 6 & 3 \end{bmatrix}$  is symmetric. The matrix  $B = \begin{bmatrix} 1 & 3 & 2 \\ 7 & 4 & 6 \\ 2 & 6 & 3 \end{bmatrix}$  is not.

### 2.2.2 Addition, Scalar Multiplication and Transpose

*Addition:* Let  $A = (a_{ij})_{m \times n}$  and  $B = (b_{ij})_{m \times n}$ . Then

$$A + B = (a_{ij} + b_{ij})_{m \times n}.$$

That is, addition of matrices is defined as element-by-element addition.

**Example 2.15.**  $\begin{bmatrix} 1 & 4 \\ 3 & 2 \\ 6 & 5 \end{bmatrix} + \begin{bmatrix} 6 & 9 \\ 1 & 2 \\ 1 & 10 \end{bmatrix} = \begin{bmatrix} 1+6 & 4+9 \\ 3+1 & 2+2 \\ 6+1 & 5+10 \end{bmatrix} = \begin{bmatrix} 7 & 13 \\ 4 & 4 \\ 7 & 15 \end{bmatrix}.$

Matrices being added together obviously have to have the same dimensions. It should also be obvious that

$$\begin{aligned} A + B &= B + A, \\ (A + B) + C &= A + (B + C). \end{aligned}$$

This means that *as far as addition is concerned*, we can manipulate matrices in the same way we manipulate ordinary numbers (as long as the matrices being added have the same dimensions).

*Scalar Multiplication:* Let  $A = (a_{ij})_{m \times n}$ , and let  $\alpha$  be a scalar. Then we define

$$\alpha A = A\alpha = (\alpha a_{ij})_{m \times n},$$

i.e., the product of a scalar and a matrix is defined to be the multiplication of each element of the matrix by the scalar.

**Example 2.16.**  $b \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} = \begin{bmatrix} ba_{11} & ba_{12} \\ ba_{21} & ba_{22} \\ ba_{31} & ba_{32} \end{bmatrix}.$

We can use scalar multiplication to define **matrix subtraction**:

$$A - B = A + (-1)B.$$

*Transpose:* When we transpose a matrix, we write its rows as its columns, and its columns as its rows. That is, the transpose of an  $(m \times n)$  matrix  $A$ , denoted either by  $A^T$  or  $A'$ , is defined by

$$[A^T]_{ij} = [A]_{ji} \text{ for all } i = 1, 2, \dots, m, j = 1, 2, \dots, n.$$

**Example 2.17.**  $\begin{bmatrix} 1 & 4 \\ 3 & 2 \\ 6 & 5 \end{bmatrix}^T = \begin{bmatrix} 1 & 3 & 6 \\ 4 & 2 & 5 \end{bmatrix}.$

We can use the transpose operator to define symmetric matrices: a symmetric matrix is simply one where  $A^T = A$ . We will often write a column vector

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

as  $x = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}^T$  or  $x^T = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}$  in order to use space more efficiently.

### 2.2.3 Exercises

**Exercise 2.6.** Let  $A = \begin{bmatrix} 7 & 13 \\ 4 & 4 \\ 7 & 15 \end{bmatrix}$ . What is the dimension of  $A$ ? What is  $[A]_{12}$ ? What is  $[A]_{31}$ ?

**Exercise 2.7.** Suppose  $A = (a_{ij})_{2 \times 4}$  where  $a_{ij} = i + j$ . Write out the matrix in full.

**Exercise 2.8.** Write out in full the matrices:

- i.  $(a_{ij})_{4 \times 4}$  where  $a_{ij} = 1$  when  $i = j$ , 0 otherwise.
- ii.  $(a_{ij})_{4 \times 4}$  where  $a_{ij} = 0$  if  $i \neq j$  (fill the rest of the entries with “\*”).
- iii.  $(a_{ij})_{5 \times 5}$  where  $a_{ij} = 0$  if  $i < j$  (fill the rest of the entries with “\*”).
- iv.  $(a_{ij})_{5 \times 5}$  where  $a_{ij} = 0$  if  $i > j$  (fill the rest of the entries with “\*”).

*These are all square matrices. Matrix (iii) is a “lower triangular matrix” and (iv) is an “upper triangular matrix” (so we have in (iii) and (iv) matrices that are square and triangular!)*

**Exercise 2.9.** Give an example of a  $(4 \times 4)$  matrix such that  $[A]_{ij} = [A]_{ji}$ .

**Exercise 2.10.** What is  $u$  and  $v$  if

$$\begin{bmatrix} u + 2v & 1 & 3 \\ 9 & 0 & 4 \\ 3 & 4 & 7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 3 \\ 9 & 0 & u + v \\ 3 & 4 & 7 \end{bmatrix} ?$$

**Exercise 2.11.** Let  $v_1, v_2, v_3, v_4$  represent cities and suppose there are one-way flights from  $v_1$  to  $v_2$  and  $v_3$ , from  $v_2$  to  $v_3$  and  $v_4$ , and two-way flights between  $v_1$  and  $v_4$ . Write out a matrix  $A$  such that  $[A]_{ij} = 1$  if there is a flight from  $v_i$  to  $v_j$ , and zero otherwise.

**Exercise 2.12.** What is the dimension of the matrix  $\begin{bmatrix} 1 & 8 & 3 \\ 9 & 1 & 9 \\ 0 & 0 & 0 \end{bmatrix}$ ?

**Exercise 2.13.** Let  $A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$  and  $B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$ . Is  $A = B$ ?

**Exercise 2.14.** If  $2A = \begin{bmatrix} 3 & 4 \\ 2 & 8 \\ 1 & 5 \end{bmatrix}$ , what is  $A$ ? If  $B - \frac{1}{2} \begin{bmatrix} 3 & 4 \\ 1 & 8 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} 6 & 4 \\ 2 & 5 \\ 3 & 1 \end{bmatrix}$ , what is  $B$ ?

**Exercise 2.15.** Which of the following matrices are symmetric?

$$\begin{array}{lll} \text{a.} & \begin{bmatrix} 1 & 2 & 3 & 5 \\ 2 & 5 & 4 & b \\ 3 & 4 & 3 & 3 \\ 5 & b & 3 & 1 \end{bmatrix} & \text{b.} & \begin{bmatrix} 1 & 2 & 3 & 5 \\ 0 & 5 & 4 & b \\ 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \text{c.} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \text{d.} & \begin{bmatrix} 1 & 1 & 3 & 5 \\ 2 & 5 & 4 & b \\ 3 & 4 & 3 & 3 \\ 5 & b & 3 & 1 \end{bmatrix} & \text{e.} & \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} & & \end{array}$$

**Exercise 2.16.** True or False?

- Symmetric matrices must be square.
- A scalar is symmetric
- If  $A$  is symmetric, then  $\alpha A$  is symmetric.
- The sum of symmetric matrices is symmetric.
- If  $(A^T)^T = A$ , then  $A$  is symmetric.

**Exercise 2.17.**

- Find  $A$  and  $B$  if they simultaneously satisfy

$$2A + B = \begin{bmatrix} 1 & 2 & 1 \\ 4 & 3 & 0 \end{bmatrix} \quad \text{and} \quad A + 2B = \begin{bmatrix} 4 & 2 & 3 \\ 5 & 1 & 1 \end{bmatrix}.$$

- If  $A + B = C$  and  $3A - 2B = 0$  simultaneously, find  $A$  and  $B$  in terms of  $C$ .

### 2.2.4 Matrix Multiplication

Let  $A$  be  $(m \times n)$  and  $B$  be  $(n \times p)$  – here we require the number of columns in  $A$  and the number of rows in  $B$  to be the same. Then the product  $AB$  is defined as the  $(m \times p)$  matrix whose  $(i, j)$ th element is defined by

$$[AB]_{ij} = \sum_{k=1}^n a_{ik} b_{kj}.$$

That is, the  $(i, j)$ th element of the product  $AB$  is defined as the sum of the product of the elements of the  $i$ th row of  $A$  with the corresponding elements in the  $j$ th column of  $B$ . For example, the  $(1, 1)$ th element of  $AB$  is

$$[AB]_{11} = \sum_{k=1}^n a_{1k} b_{k1} = a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} + \cdots + a_{1n}b_{n1}.$$

The  $(2, 3)$ th element of  $AB$  is

$$[AB]_{2,3} = \sum_{k=1}^n a_{2k} b_{k3} = a_{21}b_{13} + a_{22}b_{23} + a_{23}b_{33} + \cdots + a_{2n}b_{n3},$$

and so on. Visually, for a product of a  $(3 \times 3)$  matrix into a  $(3 \times 2)$  matrix, we have

$$\begin{aligned}
 \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} &= \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & \bullet \\ \bullet & \bullet \\ \bullet & \bullet \end{bmatrix} \\
 \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} &= \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} \\ \bullet & \bullet \\ \bullet & \bullet \end{bmatrix} \\
 \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} &= \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & \bullet \\ \bullet & \bullet \end{bmatrix}
 \end{aligned}$$

and so on.

**Example 2.18.** Let  $A = \begin{bmatrix} 2 & 8 \\ 3 & 0 \\ 5 & 1 \end{bmatrix}$  and  $B = \begin{bmatrix} 4 & 7 \\ 6 & 9 \end{bmatrix}$ . Then

$$AB = \begin{bmatrix} 2 & 8 \\ 3 & 0 \\ 5 & 1 \end{bmatrix} \begin{bmatrix} 4 & 7 \\ 6 & 9 \end{bmatrix} = \begin{bmatrix} (2)(4) + (8)(6) & (2)(7) + (8)(9) \\ (3)(4) + (0)(6) & (3)(7) + (0)(9) \\ (5)(4) + (1)(6) & (5)(7) + (1)(9) \end{bmatrix} = \begin{bmatrix} 56 & 86 \\ 12 & 21 \\ 26 & 44 \end{bmatrix}.$$

**Example 2.19.** The simultaneous equations

$$\begin{aligned}
 2x_1 - x_2 &= 4 \\
 x_1 + 2x_2 &= 2
 \end{aligned}$$

can be written in matrix form as

$$\begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}, \text{ or } Ax = b$$

where  $A = \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix}$ ,  $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ , and  $b = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$ .

## 2.2.5 Exercises

The following exercises illustrate very important aspects of matrix multiplication. You should work through each exercise and be sure to understand the point being made.

**Exercise 2.18.** Let  $A = \begin{bmatrix} 2 & 8 \\ 3 & 0 \\ 5 & 1 \end{bmatrix}$ ,  $B = \begin{bmatrix} 2 & 0 \\ 3 & 8 \end{bmatrix}$  and  $C = \begin{bmatrix} 7 & 2 \\ 6 & 3 \end{bmatrix}$ .

- Compute the matrices  $BC$ ,  $CB$ , and  $AB$ ,
- Can  $BA$  even be computed?

*Remark:* This exercise shows that for any two matrices  $A$  and  $B$ ,  $AB \neq BA$  in general. That

is, we have to distinguish between pre-multiplication and post-multiplication. In the product  $AB$ , we say that  $B$  is pre-multiplied by  $A$ , or that  $A$  is post-multiplied by  $B$ .

**Exercise 2.19.** Show that  $x^T x \geq 0$  for any vector  $x = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}^T$ . When will  $x^T x = 0$ ?

*Remark:* For any column vector  $x$ , the product  $x^T x$  is the sum of the squares of its elements.

**Exercise 2.20.**

- a. Compute  $\begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} -2 & 4 \\ 1 & -2 \end{bmatrix}$ .
- b. Let  $A = \begin{bmatrix} 1 & b \\ -\frac{1}{b} & -1 \end{bmatrix}$  where  $b \neq 0$ . Compute  $A^2$ , i.e., compute the product  $AA$ .

*Remark:* This exercise shows that you can multiply two non-zero matrices and end up with a zero matrix. Therefore  $AB = 0$  does **not** imply  $A = 0$  or  $B = 0$ . It is even possible for the square of a non-zero matrix to be a zero matrix. Of course, if  $A = 0$  or  $B = 0$ , then  $AB = 0$ .

Matrix multiplication therefore does not behave like the usual multiplication of numbers: the order of multiplication matters, and  $AB = 0$  does not imply  $A = 0$  or  $B = 0$ . In other words matrix multiplication *does* behave like regular multiplication of numbers, as the next exercise shows.

**Exercise 2.21.**

- a. Prove that  $(AB)C = A(BC)$  where  $A$ ,  $B$ , and  $C$  are  $(m \times n)$ ,  $(n \times p)$  and  $(p \times q)$  respectively.
- b. Prove that  $A(B + C) = AB + AC$  where  $A$  is  $(m \times n)$ , and  $B$  and  $C$  are  $(n \times p)$ .
- c. Prove that  $(A + B)C = AC + BC$  where  $A$  and  $B$  are  $(m \times n)$  and  $C$  is  $(n \times p)$ .

We give the proof for part (a).

$$\begin{aligned} [(AB)C]_{ij} &= \sum_{k=1}^p [AB]_{ik} [C]_{kj} \\ &= \sum_{k=1}^p \left( \sum_{l=1}^n [A]_{il} [B]_{lk} \right) [C]_{kj} \\ &= \sum_{l=1}^n [A]_{il} \left( \sum_{k=1}^p [B]_{lk} [C]_{kj} \right) \\ &= \sum_{l=1}^n [A]_{il} [BC]_{lj} \\ &= [A(BC)]_{ij}. \end{aligned}$$

**Exercise 2.22.** Let  $A$  be an  $(m \times n)$  matrix, and let  $I_n$  and  $I_m$  be identity matrices of dimensions  $(n \times n)$  and  $(m \times m)$  respectively. Show that  $I_m A = A I_n = A$ .

**Exercise 2.23.** Show that

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = b_1 \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{41} \end{bmatrix} + b_2 \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \\ a_{42} \end{bmatrix} + b_3 \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \\ a_{43} \end{bmatrix}.$$

In other words,  $Ab$  is a “linear combination” of the columns of  $A$ , with weights given in  $b$ .

**Exercise 2.24.**

- a. For  $A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \end{bmatrix}$  and  $B = \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{bmatrix}$ , prove that  $(AB)^T = B^T A^T$  by multiplying out the matrices.

*Remark: This result holds generally. For any  $(m \times n)$  matrix  $A$  and any  $(n \times p)$  matrix  $B$ , we have  $(AB)^T = B^T A^T$ . We want to show that the  $(i, j)$ th element of  $(AB)^T$  is equal to the  $(i, j)$ th element of  $B^T A^T$ . By definition of the transpose, the  $(i, j)$ th element of  $(AB)^T$  is the  $(j, i)$ th element of  $AB$ , therefore*

$$[(AB)^T]_{ij} = [AB]_{ji} = \sum_{k=1}^n a_{jk} b_{ki} = \sum_{k=1}^n b_{ki} a_{jk} = \sum_{k=1}^n [B^T]_{ik} [A^T]_{kj} = [B^T A^T]_{ij}.$$

- b. Prove that  $(ABC)^T = C^T B^T A^T$ .

**Exercise 2.25.** Let  $X$  be a general  $(n \times k)$  matrix. Explain why  $X^T X$  is square and symmetric.

*Remark: The matrix  $X^T X$  is encountered frequently in econometrics.*

**Exercise 2.26.** The trace of an  $(n \times n)$  matrix  $A$  is defined to be

$$\text{tr}(A) = \sum_{i=1}^n a_{ii}.$$

That is, the trace of a square matrix is simply the sum of its diagonal elements. The trace of a scalar is the scalar itself.

- a. If  $A$  and  $B$  are square matrices of the same dimensions, show that

$$\text{tr}(A + B) = \text{tr}(A) + \text{tr}(B).$$

- b. If  $A$  is a square matrix, show that  $\text{tr}(A^T) = \text{tr}(A)$ .  
 c. If  $A$  is  $(m \times n)$  and  $B$  is  $(n \times m)$ , show that  $\text{tr}(AB) = \text{tr}(BA)$ .  
 d. If  $x$  is an  $(n \times 1)$  column vector, show that  $x^T x = \text{tr}(xx^T)$ . Show this by  
 i. direct multiplication,  
 ii. using the result in part(c) and the fact that the trace of a scalar is the scalar itself.

**Exercise 2.27.** Let  $i_n$  be an  $(n \times 1)$  vector of ones, i.e.,  $i_n = [1 \ 1 \ \cdots \ 1]^T$ . Show that the sample mean of the elements of the column vector  $y = [y_1 \ y_2 \ \cdots \ y_n]^T$  can be written

$$\bar{y} = (i_n^T i_n)^{-1} i_n^T y.$$

**Exercise 2.28.** Prove that  $A(\alpha B) = (\alpha A)B = \alpha(AB)$ .



### 2.2.6 Partitioned Matrices

We can partition the contents of an  $(m \times n)$  matrix into blocks of submatrices. For instance, we can write

$$A = \begin{bmatrix} 1 & 3 & 2 & 6 \\ 2 & 8 & 2 & 1 \\ 3 & 1 & 2 & 4 \\ 4 & 2 & 1 & 3 \\ 3 & 1 & 1 & 7 \end{bmatrix} = \left[ \begin{array}{c|ccc} 1 & 3 & 2 & 6 \\ 2 & 8 & 2 & 1 \\ \hline 3 & 1 & 2 & 4 \\ 4 & 2 & 1 & 3 \\ 3 & 1 & 1 & 7 \end{array} \right] = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

where  $A_{11}$  is  $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ ,  $A_{21}$  is  $\begin{bmatrix} 3 \\ 4 \\ 3 \end{bmatrix}$ ,  $A_{12}$  is  $\begin{bmatrix} 3 & 2 & 6 \\ 8 & 2 & 1 \end{bmatrix}$ , and  $A_{22}$  is  $\begin{bmatrix} 1 & 2 & 4 \\ 2 & 1 & 3 \\ 1 & 1 & 7 \end{bmatrix}$ .

Of course, there are many ways of partitioning any given matrix:

$$A = \begin{bmatrix} 1 & 3 & 2 & 6 \\ 2 & 8 & 2 & 1 \\ 3 & 1 & 2 & 4 \\ 4 & 2 & 1 & 3 \\ 3 & 1 & 1 & 7 \end{bmatrix} = \left[ \begin{array}{c|ccc} 1 & 3 & 2 & 6 \\ 2 & 8 & 2 & 1 \\ \hline 3 & 1 & 2 & 4 \\ 4 & 2 & 1 & 3 \\ 3 & 1 & 1 & 7 \end{array} \right] = \left[ \begin{array}{cc|cc} 1 & 3 & 2 & 6 \\ 2 & 8 & 2 & 1 \\ \hline 3 & 1 & 2 & 4 \\ 4 & 2 & 1 & 3 \\ \hline 3 & 1 & 1 & 7 \end{array} \right].$$

It can be shown that addition and multiplication of partitioned matrices can be carried out as though the blocks are elements, as long as the matrices are partitioned conformably.

*Addition of Partitioned Matrices* Consider two  $(m \times n)$  matrices  $A$  and  $B$  partitioned in the following manner:

$$A = \begin{bmatrix} \underbrace{A_{11}}_{m_1 \times n_1} & \underbrace{A_{12}}_{m_1 \times n_2} \\ \underbrace{A_{21}}_{m_2 \times n_1} & \underbrace{A_{22}}_{m_2 \times n_2} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} \underbrace{B_{11}}_{m_1 \times n_1} & \underbrace{B_{12}}_{m_1 \times n_2} \\ \underbrace{B_{21}}_{m_2 \times n_1} & \underbrace{B_{22}}_{m_2 \times n_2} \end{bmatrix}$$

where  $n_1 + n_2 = n$  and  $m_1 + m_2 = m$ . We emphasize that  $A$  and  $B$  must be of the same size and partitioned identically. Then

$$A + B = \begin{bmatrix} \underbrace{A_{11} + B_{11}}_{m_1 \times n_1} & \underbrace{A_{12} + B_{12}}_{m_1 \times n_2} \\ \underbrace{A_{21} + B_{21}}_{m_2 \times n_1} & \underbrace{A_{22} + B_{22}}_{m_2 \times n_2} \end{bmatrix}. \quad (2.4)$$

*Multiplication of Partitioned Matrices* Now consider two matrices  $A$  and  $B$  with dimensions  $(m \times p)$  and  $(p \times n)$  respectively. Suppose they are partitioned as follows:

$$A = \begin{bmatrix} \underbrace{A_{11}}_{m_1 \times p_1} & \underbrace{A_{12}}_{m_1 \times p_2} \\ \underbrace{A_{21}}_{m_2 \times p_1} & \underbrace{A_{22}}_{m_2 \times p_2} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} \underbrace{B_{11}}_{p_1 \times n_1} & \underbrace{B_{12}}_{p_1 \times n_2} \\ \underbrace{B_{21}}_{p_2 \times n_1} & \underbrace{B_{22}}_{p_2 \times n_2} \end{bmatrix}.$$

In particular, the partition is such that the column-wise partition of  $A$  matches the row-wise

partition of  $B$ . Then

$$AB = \begin{bmatrix} \underbrace{A_{11}}_{m_1 \times p_1} & \underbrace{A_{12}}_{m_1 \times p_2} \\ \underbrace{A_{21}}_{m_2 \times p_1} & \underbrace{A_{22}}_{m_2 \times p_2} \end{bmatrix} \begin{bmatrix} \underbrace{B_{11}}_{p_1 \times n_1} & \underbrace{B_{12}}_{p_1 \times n_2} \\ \underbrace{B_{21}}_{p_2 \times n_1} & \underbrace{B_{22}}_{p_2 \times n_2} \end{bmatrix} = \begin{bmatrix} \underbrace{A_{11}B_{11} + A_{12}B_{21}}_{m_1 \times n_1} & \underbrace{A_{11}B_{12} + A_{12}B_{22}}_{m_1 \times n_2} \\ \underbrace{A_{21}B_{11} + A_{22}B_{21}}_{m_2 \times n_1} & \underbrace{A_{21}B_{12} + A_{22}B_{22}}_{m_2 \times n_2} \end{bmatrix}. \quad (2.5)$$

*Transposition of Partitioned Matrices* It is straightforward to show that

$$A = \begin{bmatrix} \underbrace{A_{11}}_{m_1 \times n_1} & \underbrace{A_{12}}_{m_1 \times n_2} \\ \underbrace{A_{21}}_{m_2 \times n_1} & \underbrace{A_{22}}_{m_2 \times n_2} \end{bmatrix} \Rightarrow A^T = \begin{bmatrix} \underbrace{A_{11}^T}_{n_1 \times m_1} & \underbrace{A_{21}^T}_{n_1 \times m_2} \\ \underbrace{A_{12}^T}_{n_2 \times m_1} & \underbrace{A_{22}^T}_{n_2 \times m_2} \end{bmatrix}. \quad (2.6)$$

### 2.2.7 Determinants and Inverses

Suppose  $A$  is a square matrix of dimension  $(n \times n)$ . The inverse of  $A$ , if it exists, is the matrix which we will denote as  $A^{-1}$ , such that

$$A^{-1}A = I$$

**Example 2.20.** The inverse of the matrix

$$A = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \quad \text{is} \quad A^{-1} = \frac{1}{-2} \begin{bmatrix} 4 & -3 \\ -2 & 1 \end{bmatrix}.$$

This can be verified by direct multiplication:

$$A^{-1}A = \frac{1}{-2} \begin{bmatrix} 4 & -3 \\ -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

If  $A^{-1}A = I$ , it will also be true that  $AA^{-1} = I$ .

The formula for the inverse of an arbitrary  $(2 \times 2)$  matrix  $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$  is

$$A^{-1} = \frac{1}{|A|} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \quad \text{where} \quad |A| = a_{11}a_{22} - a_{12}a_{21}. \quad (2.7)$$

You can easily verify this by direct multiplication. It is worth your while to commit (2.7) to memory. The expression  $|A|$  in (2.7) is called the determinant of the  $(2 \times 2)$  matrix  $A$ . Notice that if  $|A| = 0$ , then the inverse will not exist (in that case, we say that  $A$  is ‘singular’). If  $|A| \neq 0$ , then the inverse will exist.

**Example 2.21.** The matrix  $A = \begin{bmatrix} 1 & 3 \\ 2 & 6 \end{bmatrix}$  has determinant zero:

$$|A| = (1)(6) - (2)(3) = 0.$$

It does not have an inverse.

When will  $|A| = 0$ ? For the  $(2 \times 2)$  case, it will be when one or more row or columns are all zero, or if one row is a multiple of the other, or if one column is a multiple of the other.

We will omit the formula for the determinant and inverse of larger square matrices, but the same story applies: the inverse of a square matrix exists if and only if it has a non-zero determinant. We will discuss a way of computing the determinant of a general square matrix in a later chapter.

One application of matrix inverse is in solving simultaneous equations, e.g.,

$$\begin{aligned} 2x_1 - x_2 &= 4 \\ x_1 + 2x_2 &= 2 \end{aligned}$$

which can be written in matrix form as  $Ax = b$  where

$$A = \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \text{ and } b = \begin{bmatrix} 4 \\ 2 \end{bmatrix}.$$

Since

$$A^{-1} = \frac{1}{5} \begin{bmatrix} 2 & 1 \\ -1 & 2 \end{bmatrix},$$

we can simply (pre-)multiply both sides of  $Ax = b$  by  $A^{-1}$  to get the solution:

$$Ax = b \Rightarrow A^{-1}Ax = A^{-1}b \Rightarrow x = A^{-1}b.$$

For our specific example, we have

$$x = A^{-1}b = \frac{1}{5} \begin{bmatrix} 2 & 1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}.$$

### 2.2.8 Exercises

**Exercise 2.29.** Let

$$A = \left[ \begin{array}{c|ccc} 1 & 3 & 2 & 6 \\ 2 & 8 & 2 & 1 \\ \hline 3 & 1 & 2 & 4 \\ 4 & 2 & 1 & 3 \\ 3 & 1 & 1 & 7 \end{array} \right] \quad \text{and} \quad B = \left[ \begin{array}{c|cc} 2 & 0 & 1 \\ \hline 3 & 1 & 3 \\ 1 & 5 & 4 \\ 4 & 1 & 1 \end{array} \right].$$

Verify the partitioned matrix multiplication formulas by multiplying in the usual way, then multiplying using (Eq. 2.5). Verify the transposition formula (2.6) for both matrices.

**Exercise 2.30.** Consider the following two partitions of an  $(m \times n)$  matrix  $A$ :

$$A = \left[ \begin{array}{c|c|c|c} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{array} \right] = [A_1 \quad A_2 \quad \cdots \quad A_n] \quad \text{and} \quad A = \left[ \begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{array} \right] = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_m^T \end{bmatrix}.$$

Let  $c = [c_1 \ c_2 \ \dots \ c_m]^T$ ,  $b = [b_1 \ b_2 \ \dots \ b_n]^T$ . Show that

- $c^T A = c_1 a_1^T + c_2 a_2^T + \dots + c_m a_m^T$ , i.e.,  $c^T A$  is a linear combination of the rows of  $A$ .
- $Ab = b_1 A_1 + b_2 A_2 + \dots + b_n A_n$ , i.e.,  $Ab$  is a linear combination of the columns of  $A$ .

**Exercise 2.31.** Let  $X$  be a  $(n \times 3)$  data matrix containing  $n$  observations of three variables:

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & x_{n3} \end{bmatrix}$$

where  $x_{ij}$  represents the  $i$ th observation of variable  $j$ . We can partition this matrix to emphasize the variables by writing  $X$  as  $X = [X_1 \ X_2 \ X_3]$  where

$$X_1 = \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \\ \vdots \\ x_{n1} \end{bmatrix}, X_2 = \begin{bmatrix} x_{12} \\ x_{22} \\ x_{32} \\ \vdots \\ x_{n2} \end{bmatrix}, \text{ and } X_3 = \begin{bmatrix} x_{13} \\ x_{23} \\ x_{33} \\ \vdots \\ x_{n3} \end{bmatrix}.$$

Alternatively, we can partition the data matrix to emphasize observations:

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ x_3^T \\ \vdots \\ x_n^T \end{bmatrix}.$$

where

$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \end{bmatrix} \quad i = 1, 2, \dots, n,$$

is the column vector containing the  $i$ th observations of all three variables. Show that the matrix  $X^T X$  can be written as

$$\begin{aligned} X^T X &= \begin{bmatrix} X_1^T X_1 & X_1^T X_2 & X_1^T X_3 \\ X_2^T X_1 & X_2^T X_2 & X_2^T X_3 \\ X_3^T X_1 & X_3^T X_2 & X_3^T X_3 \end{bmatrix} \\ &= \begin{bmatrix} \sum_{i=1}^n x_{i1}^2 & \sum_{i=1}^n x_{i1}x_{i2} & \sum_{i=1}^n x_{i1}x_{i3} \\ \sum_{i=1}^n x_{i1}x_{i2} & \sum_{i=1}^n x_{i2}^2 & \sum_{i=1}^n x_{i2}x_{i3} \\ \sum_{i=1}^n x_{i1}x_{i3} & \sum_{i=1}^n x_{i2}x_{i3} & \sum_{i=1}^n x_{i3}^2 \end{bmatrix} \\ &= \sum_{i=1}^n x_i x_i^T. \end{aligned}$$

### 2.2.9 Matrices in R

We have already learnt how to create matrices in R. Here are three matrices:

```
A = matrix(c(2,3,5,9,0,1),3,2); A
```

```
      [,1] [,2]
[1,]    2    9
[2,]    3    0
[3,]    5    1
```

```
B = matrix(c(2,3,0,8),2,2); B
```

```
      [,1] [,2]
[1,]    2    0
[2,]    3    8
```

```
C = matrix(c(7,6,2,3),2,2); C
```

```
      [,1] [,2]
[1,]    7    2
[2,]    6    3
```

```
D = matrix(0,2,2); D
```

```
      [,1] [,2]
[1,]    0    0
[2,]    0    0
```

```
I3 = diag(c(1,1,1)); I3
```

```
      [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
```

Feeding an R vector into `diag()` creates a diagonal matrix. Feeding a square matrix into `diag()` draws out the diagonal elements:

```
diag(C)
```

```
[1] 7 3
```

In R, the `*` operator refers to element-by-element multiplication.

```
B*C
```

```
      [,1] [,2]
[1,]   14    0
[2,]   18   24
```

To do *matrix multiplication*, we use the `%%` operator. Of course, the matrices must be compatible for multiplication.

```
B%%C
```

```
      [,1] [,2]
[1,]   14    4
[2,]   69   30
```

```
A%%B
```

```
      [,1] [,2]
[1,]   31   72
[2,]    6    0
[3,]   13    8
```

```
B%%A
```

```
Error in B %% A: non-conformable arguments
```

We can use `*` for scalar multiplication

```
3*B
```

```
      [,1] [,2]
[1,]    6    0
[2,]    9   24
```

Addition and subtraction can be done with the `+` and `-` operators. A matrix can be transposed using the function `t()`.

**Example 2.22.** We transpose the matrix  $A$  defined earlier.

```
A
```

```
      [,1] [,2]
[1,]    2    9
[2,]    3    0
[3,]    5    1
```

```
t(A)
```

```
      [,1] [,2] [,3]
[1,]    2    3    5
[2,]    9    0    1
```

The determinant of a square matrix can be obtained using the `det()` function:

**Example 2.23.** Consider the matrices

$$A = \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 3 & 4 \\ 3 & 1 & 2 \\ 6 & 2 & 1 \end{bmatrix}, \quad \text{and} \quad E = \begin{bmatrix} 2 & 2 & 4 \\ 2 & 1 & 3 \\ 2 & 5 & 7 \end{bmatrix},$$

Their determinants are:

```

A = matrix(c(2,1,-1,2), 2, 2); A; cat("Determinant is", det(A), "\n\n")

      [,1] [,2]
[1,]     2  -1
[2,]     1   2
Determinant is 5

D = matrix(c(0,3,6,3,1,2,4,2,1), 3, 3); D; cat("Determinant is", det(D), "\n\n")

      [,1] [,2] [,3]
[1,]     0   3   4
[2,]     3   1   2
[3,]     6   2   1
Determinant is 27

E = matrix(c(2,2,2,2,1,5,4,3,7), 3, 3); E; cat("Determinant is", det(E), "\n\n")

      [,1] [,2] [,3]
[1,]     2   2   4
[2,]     2   1   3
[3,]     2   5   7
Determinant is 0

```

To calculate the inverse of square matrices, use the `solve()` function

```

solve(A)

      [,1] [,2]
[1,]  0.4  0.2
[2,] -0.2  0.4

solve(D)

      [,1]      [,2]      [,3]
[1,] -0.1111111  0.1851852  0.07407407
[2,]  0.3333333 -0.8888889  0.44444444
[3,]  0.0000000  0.6666667 -0.33333333

solve(E) # not going to work, since det(E)=0

```

Error in solve.default(E): Lapack routine dgesv: system is exactly singular: U[3,3] = 0

To solve a system of linear equations  $Ax = b$  in R, you can use `solve(A)%*%b` or `solve(A,b)`.

```

A = matrix(c(2,1,-1,2), 2, 2);
b = matrix(c(4,2),2,1)
x = solve(A,b)
x

      [,1]
[1,]     2
[2,]     0

```

The trace of a matrix can be computed with `sum(diag())`:

```
sum(diag(A)); sum(diag(D)); sum(diag(E))
```

```
[1] 4
[1] 2
[1] 10
```

## 2.3 A Brief Review of Optimization Theory

Many estimators used in econometrics are based on the optimization (i.e., finding the minimum or maximum point) of some objective function. Optimization can be complicated when dealing with ‘poorly behaved’ functions, but is straightforward for certain classes of functions.

### 2.3.1 Functions of One Variable

The **minimum point** of a function  $f$  is the point  $x^*$  in the domain of  $f$  such that  $f(x) \geq f(x^*)$  for all  $x$  in its domain. If it is the case that  $f(x) > f(x^*)$  for all  $x \neq x^*$  in its domain, then  $x^*$  is said to be a **strict minimum point**. **Maximum points** and **strict maximum points** are defined similarly, with the reverse inequalities.

Consider for the moment functions  $f$  that satisfy the following two conditions:

1. the domain of  $f$  is an open interval  $(a, b)$ ,
2. the function is twice differentiable, i.e.,  $f'(x)$  and  $f''(x)$  exist for every  $x$  in  $(a, b)$ .

The first tool for finding the maximum or minimum points of such functions is the fact that minimum and maximum points, *if* they exist, must satisfy

$$f'(x^*) = 0. \quad (2.8)$$

The rough intuition is that if the slope of the function at  $x^*$  is not zero, then moving  $x^*$  to the left or right will lead to higher or lower values of the function. We call any  $x^*$  that satisfies (2.8) a “stationary point”.

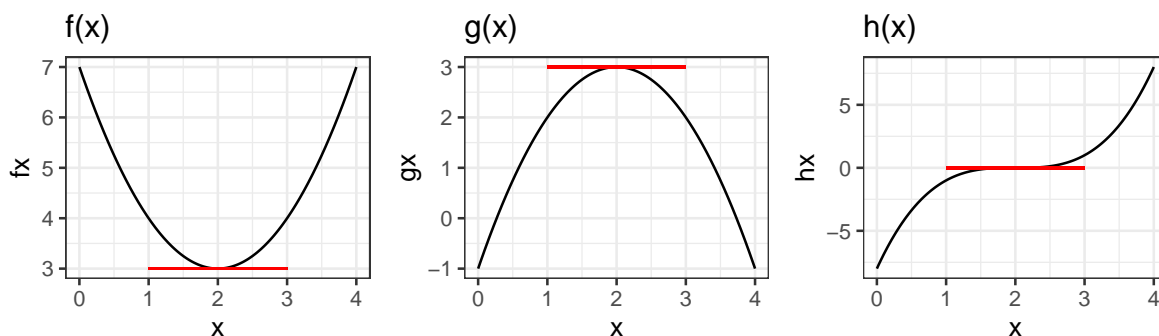
This result is useful because, as a necessary condition, it helps us sieve out all *candidate* minimum and maximum points. However, the result is not sufficient because it can also pick out points that are neither maximum or minimum points.

**Example 2.24.** Consider the functions  $f(x) = (x - 2)^2 + 3$ ,  $g(x) = -(x - 2)^2 + 3$  and  $h(x) = (x - 2)^3$ . All three are defined and twice-differentiable over the entire real line  $(-\infty, \infty)$ . The three functions are shown below:

```
x <- seq(0,4, by=0.01)
fx <- (x-2)^2 + 3
gx <- -(x-2)^2 + 3
hx <- (x-2)^3
dat <- data.frame(x=x,fx=fx,gx=gx,hx=hx)
p1 <- ggplot(data=dat) + geom_line(aes(x=x,y=fx)) + ggtitle("f(x)") +
  geom_segment(aes(x=1,y=3,xend=3,yend=3), color='red') + theme_bw()
p2 <- ggplot(data=dat) + geom_line(aes(x=x,y=gx)) + ggtitle("g(x)") +
  geom_segment(aes(x=1,y=3,xend=3,yend=3), color='red') + theme_bw()
p3 <- ggplot(data=dat) + geom_line(aes(x=x,y=hx)) + ggtitle("h(x)") +
```



```
geom_segment(aes(x=1,y=0,xend=3,yend=0), color='red') + theme_bw()
(p1 | p2 | p3)
```



In all three cases, the point  $x^* = 2$  satisfies  $f'(x^*) = 0$ ,  $g'(x^*) = 0$  and  $h'(x^*) = 0$ , but in the case of  $h(x)$ ,  $x^* = 2$  is neither a maximum or minimum point (it is an *inflection point*).

Quite often we find ourselves in situations where our function behaves like  $f(x)$  or  $g(x)$  in the example above, in which case the first order condition does pick out the strict minimum or strict maximum point. For twice-differentiable functions we can use the second-order derivative to see if we are indeed working with such functions. If

$$3a. \quad f''(x) > 0 \text{ for all } x \in (a, b)$$

then the point  $x^*$  such that  $f'(x^*) = 0$  gives the minimum point of the function. The condition 3a says that moving from left to right the slope of the function always increases, so the function arcs upwards (we say the function is *convex*, or *convex upwards*). If

3b.  $f''(x) < 0$  for all  $x \in (a, b)$ , then the slope of the function is always decreases as  $x$  increases. The function therefore arcs downwards (i.e., is *concave*, or *concave downwards*), and the stationary point then gives a strict maximum point.

We refer to  $f'(x^*) = 0$  as the “first-order condition”, and  $f''(x^*) > 0$  as the “second-order condition” for a minimum, and  $f''(x^*) < 0$  as the second-order condition for a maximum.

**Example 2.25.** The point  $x = 2$  is the minimum point of  $f(x) = (x - 2)^2 + 3$  since  $f'(x) = 2(x - 2) = 0$  at  $x = 2$  and  $f''(x) = 2 > 0$  for all  $x$ . The point  $x = 2$  is the maximum point of  $g(x) = -(x - 2)^2 - 3$  since  $f'(x) = -2(x - 2) = 0$  at  $x = 2$  and  $f''(x) = -2 < 0$  for all  $x$ . The first derivative of the function  $h(x) = (x - 2)^3$  is zero at  $x = 2$ :  $h'(x) = 3(x - 2)^2 = 0$  at  $x = 2$ . However, it fails the second order condition for both a maximum and a minimum as  $h''(x) = 6(x - 2)$  is neither always positive or always negative.

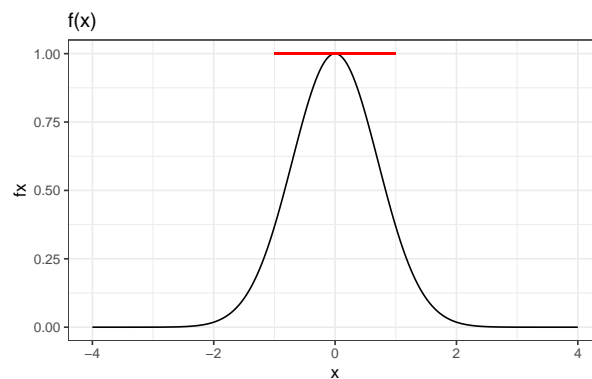
Note the  $f(x^*) = 0$  together with  $f''(x^*) > 0$  or  $f''(x^*) < 0$  is sufficient to guarantee that  $x^*$  is a minimum or maximum point, but they are not necessary conditions. In  $h(x)$  above, the point  $x = 2$  turned out to be neither a maximum or a minimum of the function  $h(x) = (x - 2)^3$ , but we could not have claimed this solely on the basis of the function’s failure to satisfy the second-order condition. It is possible for a function to fail the second-order condition and yet yield a maximum or minimum point at the point  $x^*$  where the first derivative is zero.

**Example 2.26.** Suppose we wish to maximize the function  $f(x) = e^{-x^2}$  defined over the open interval  $(-\infty, \infty)$ . It is twice-differentiable everywhere on this interval. The first derivative is  $f'(x) = -2xe^{-x^2}$  which is zero at  $x = 0$  so  $x = 0$  is a candidate maximum point. However, the second derivative

$$f''(x) = 2e^{-x^2}(2x^2 - 1)$$

is not strictly positive or strictly negative everywhere:  $f''(x) < 0$  when  $-1/\sqrt{2} < x < 1/\sqrt{2}$ , and  $f''(x) > 0$  when  $x < -1/\sqrt{2}$  or  $x > 1/\sqrt{2}$ . It fails to satisfy the second-order condition for a maximum (or a minimum). Yet the point  $x = 0$  is a maximum point (see plot below).

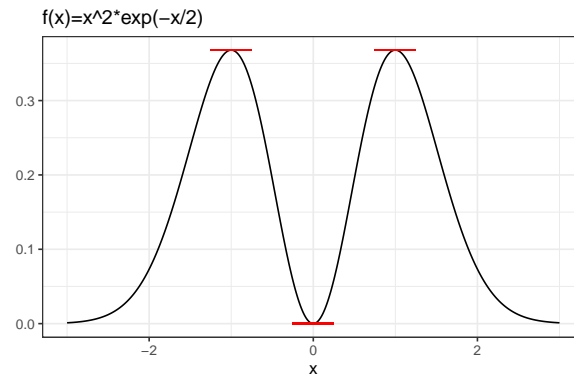
```
x <- seq(-4,4, by=0.01)
fx <- exp(-x^2)
dat <- data.frame(x=x,fx=fx)
ggplot(data=dat) + geom_line(aes(x=x,y=fx)) + ggtitle("f(x)") +
  geom_segment(aes(x=-1,y=1,xend=1,yend=1), color='red') +
  theme(plot.title = element_text(size = 10)) + theme_bw()
```



We have noted that the first order condition (2.8) cannot distinguish between maximum and minimum points. It also cannot distinguish between strict vs non-strict maximum/minimum points.

**Example 2.27.** Consider the function  $f(x) = x^2 \exp(-x^2)$  shown below.

```
x <- seq(-3,3, by=0.01)
f <- function(x){x^2*exp(-x^2)}
dat <- data.frame(x=x,fx=f(x))
x1 <- -1; x2 <- 0; x3 <- 1
ggplot(data=dat) + geom_line(aes(x=x,y=fx)) +
  ggtitle("f(x)=x^2*exp(-x/2)") + ylab("") +
  geom_segment(aes(x=x1-0.25,y=f(x1),xend=x1+0.25,yend=f(x1)), color='red') +
  geom_segment(aes(x=x2-0.25,y=0,xend=x2+0.25,yend=0), color='red') +
  geom_segment(aes(x=x3-0.25,y=f(x3),xend=x3+0.25,yend=f(x3)), color='red') +
  theme(plot.title = element_text(size = 9)) + theme_bw()
```

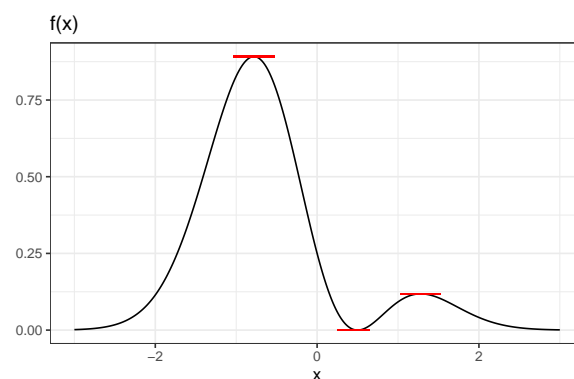


This function has three stationary points. The points  $x = -1$  and  $x = 1$  are (non-strict) maximum points. The point  $x = 0$  is a strict minimum point.

We have shown an example where the first order condition yields neither a maximum point nor minimum point. Sometimes the first order condition yields points that are maximum or minimum, but only “locally” so.

**Example 2.28.** Take the function  $f(x) = (x - 0.5)^2 \exp(-x^2)$  shown below.

```
x <- seq(-3,3, by=0.01)
f <- function(x){(x-0.5)^2*exp(-x^2)}
dat <- data.frame(x=x,fx=f(x))
x1 <- (1-sqrt(17))/4; x2 <- 0.5; x3 <- (1+sqrt(17))/4
ggplot(data=dat) + geom_line(aes(x=x,y=fx)) + ggtitle("f(x)") + ylab("") +
  geom_segment(aes(x=x1-0.25,y=f(x1),xend=x1+0.25,yend=f(x1)), color='red') +
  geom_segment(aes(x=x2-0.25,y=0,xend=x2+0.15,yend=0), color='red') +
  geom_segment(aes(x=x3-0.25,y=f(x3),xend=x3+0.25,yend=f(x3)), color='red') +
  theme(plot.title = element_text(size = 9)) + theme_bw()
```



In this example the first order condition picks out three candidate maximum and minimum points. The points  $x = (1 - \sqrt{17})/4 \approx -0.78$  is a strict maximum point, where as  $x = 0.5$  is a strict minimum point. The point  $x = (1 + \sqrt{17})/4 \approx 1.28$  is strictly speaking not a maximum point, but it is if we consider only a small enough neighborhood about  $x = (1 + \sqrt{17})/4$ . We call this a **local maximum point**. The first order condition picks out local maximum and local minimum points, in addition to the “global” ones.

Concave/convex functions need not have maximum/minimum points.

**Example 2.29.** Let  $f(x) = 1 - 1/x$ ,  $x \in (0, \infty)$ . This function is twice-differentiable over its domain which is an open interval. We have  $f'(x) = 1/x^2$  and  $f''(x) = -2/x^3$ . The second derivative is negative for all  $x \in (0, \infty)$ . However, there is no point  $x^*$  such that  $f'(x^*) = 0$ . The function is concave, but it is strictly increasing and has no maximum point (even though it is bounded from above!).

The discussion above assume twice-differentiable functions defined over open intervals. Things can get a little more complicated once we depart from this scenario, as the following examples show:

**Example 2.30.** The function  $f(x) = |x|$  has a minimum point at  $x = 0$ . Yet there are no points at which  $f'(x) = 0$  (the function is not differentiable at  $x = 0$ ).

**Example 2.31.** Consider the function  $f(x) = x^2$  defined over the restricted domain  $x \in [1, 2]$ . Note that the domain is now not an open interval. At no point in the domain do we have  $f'(x) = 0$ . Yet  $x = 2$  is a global maximum point, and  $x = 1$  is a global minimum point.

For the moment, we will not need to go beyond the class of twice-differentiable concave/convex functions defined over an open interval. However, we will have to consider functions of many variables.

### 2.3.2 Functions of Many Variables

We will only consider the simplest cases. First consider a function  $f(x, y)$ , and assume that the first and second partial derivatives exist everywhere in its domain, which we take to be  $\mathbb{R}^2$ . The first order condition in this case is that a minimum or maximum point  $(x^*, y^*)$ , if it exists, must satisfy

$$\begin{aligned} f'_x(x^*, y^*) &= 0 \\ f'_y(x^*, y^*) &= 0 \end{aligned} \tag{2.9}$$

Intuitively, if the slope of the function at  $(x^*, y^*)$  in any given direction is not zero, then we can increase or decrease the function by moving along that direction. If (2.9) holds, then the slope of the function in the  $x$ -direction and the  $y$ -direction are both zero at the stationary point  $(x^*, y^*)$ . This in turn guarantees that the slope in every direction is zero.

Like the univariate case, (2.9) is only necessary, not sufficient. The plots below are three-dimensional plots, from left to right, of the functions

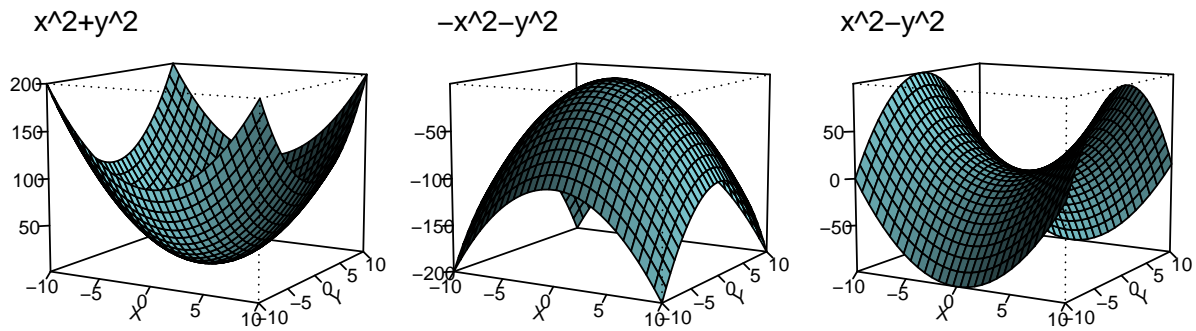
$$f(x, y) = x^2 + y^2, \quad f(x, y) = -x^2 - y^2 \quad \text{and} \quad f(x, y) = x^2 - y^2.$$

```
x <- y <- seq(-10,10,length.out=30)
funcs <- list(NA,NA,NA)
funcs[[1]] <- function(x,y){r <- x^2 + y^2}
funcs[[2]] <- function(x,y){r <- -x^2 - y^2}
funcs[[3]] <- function(x,y){r <- x^2 - y^2}
fneqs<-c("x^2+y^2", "-x^2-y^2", "x^2-y^2")
op <- par(bg="white", oma=c(0,0,0,0), mfrow=c(1,3))
```

```

for (i in 1:3){
  z <- outer(x,y,funcs[[i]])
  z[is.na(z)] <- 1
  par(mar=c(0,2,0,2), pin=c(2,2.5))
  persp(x,y,z,theta=30, phi=10,r=10, expand=0.8, col="cadetblue1",
        ltheta=120, shade=0.75,ticktype="detailed", xlab="X", ylab="Y", zlab="")
  mtext(fneqs[i],side=3, line=-3, adj=0)
}
mtext("Three functions with stationary point at (0,0)", line=-3, side=1, outer=TRUE)

```



Three functions with stationary point at (0,0)

All three have a stationary point at  $(x^*, y^*) = (0, 0)$ , but in the first case we have a minimum, in the second a maximum, and in the third case neither. In the first case, the function is convex throughout. In the second, the function is concave throughout, and in the third case neither. For the most part we will deal with concave or convex functions. How do we tell if a function is concave or convex? For functions of one variable we look at the second derivative. For a function of many variables, we look at the matrix of second-order partial derivatives. For example, for the function  $f(x, y, z)$ , we would look at the matrix

$$H_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial z \partial x} & \frac{\partial^2 f}{\partial z \partial y} & \frac{\partial^2 f}{\partial z^2} \end{bmatrix}$$

This matrix, which is symmetric, is called the **Hessian** of the function  $f$ . If it is the case that the Hessian satisfies

$$a' H_f a > 0 \quad (2.10)$$

for all non-zero vectors  $a$ , then the function is convex (we also say that the Hessian is **positive definite**). In this case we say that the “second-order condition” for a minimum holds, and the stationary point is a minimum point. If

$$a' H_f a < 0 \quad (2.11)$$

then the Hessian is **negative definite**, and the function is concave. The second-order condition for a maximum holds, and the stationary point is the maximum point. If neither condition holds, then the second order condition is silent about the status of the stationary point.

**Example 2.32.** For the function  $f(x, y) = x^2 + y^2$ , we have

$$H_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}.$$

Therefore

$$a' H_f a = \begin{bmatrix} a_1 & a_2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = 2a_1^2 + 2a_2^2$$

which is strictly positive for all non-zero vectors  $a$ .

For the function  $f(x, y) = -x^2 - y^2$ , we have

$$a' H_f a = \begin{bmatrix} a_1 & a_2 \end{bmatrix} \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = -2(a_1^2 + a_2^2)$$

which is strictly negative for all non-zero vectors  $a$ . In the case of  $f(x, y) = x^2 - y^2$ , we have

$$a' H_f a = \begin{bmatrix} a_1 & a_2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = 2(a_1^2 - a_2^2)$$

which can be negative or positive or zero, depending on the values of  $a_1$  and  $a_2$ .

The first and second order conditions stated in this section extend to functions of more than two variables.

### 2.3.3 Exercises

**Exercise 2.32.** Find the first and second derivatives of the function

$$f(x) = \frac{x}{1+x^2}.$$

Show that  $f'(x) = 0$  at  $x = 1$  and  $x = -1$ . Find the regions over which  $f''(x)$  is positive, and the regions over which  $f''(x)$  is negative. What argument can you give to prove that  $x = 1$  is a global maximum point, and  $x = -1$  is a global minimum point of the function?

**Exercise 2.33.** Let  $\{X_i\}_{i=1}^N$  be a sample of  $N$  observations of a random variable  $X$  with population mean  $E[X] = \mu$ . Show that the sample mean  $\hat{\mu} = \frac{1}{N} \sum_{i=1}^N X_i$  minimizes the function

$$f(\hat{\mu}) = \sum_{i=1}^N (X_i - \hat{\mu})^2.$$

**Exercise 2.34.** Show that  $f(x) = 2x^3 - 6x$  has stationary points at  $x = 1$  and  $x = -1$ . Show that  $x = -1$  is a local maximum point, and  $x = 1$  is a local minimum point.

**Exercise 2.35.** Let  $A$  be the following matrix, which can be decomposed into the product of two matrices as shown:

$$A = \begin{bmatrix} 5 & 5 \\ 5 & 10 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 2 & 1 \end{bmatrix}.$$

Explain why this shows  $A$  is positive definite. Find the stationary point of the function

$$f(x, y) = 5x^2 + 10xy + 10y^2.$$

Is this point a maximum point, a minimum point, or neither?

## 2.4 Application: Fitting a Straight Line by Least Squares

For an application of the ideas we have just reviewed, consider the problem of fitting a straight line through a scatterplot of  $n$  points  $\{X_i, Y_i\}_{i=1}^n$ . For illustration, we will use data in the file `ols01.csv` which comprises 10 observations of variables  $X$  and  $Y$ .

```
# The function read_csv() is from tidyverse::readr
# The option show_col_types=F shuts some automated messages from read_csv()
df <- read_csv("data\\ols01.csv", show_col_types=F)
glimpse(df)      # tidyverse version of str() to quickly explore tibble
```

Rows: 10

Columns: 2

\$ X <dbl> 2.514333, 5.169248, 1.731986, 3.421461, 4.028095, 4.577327, 8.194569~

\$ Y <dbl> 7.639444, 10.668647, 3.110330, 1.846599, 11.782167, 10.582858, 15.45~

The `glimpse()` function gives you a quick look at the data. Although the data is presented in transposed form, there are 2 columns of data, 10 rows each. The data are plotted in Fig. 2.1.

```
p1 <- df %>%
  ggplot(aes(x=X,y=Y)) + geom_point(size=1) + theme_minimal() +
  xlim(0,10) + ylim(0,16) + coord_fixed(ratio=1/3) +
  theme(axis.title=element_text(size=10))
p1
```

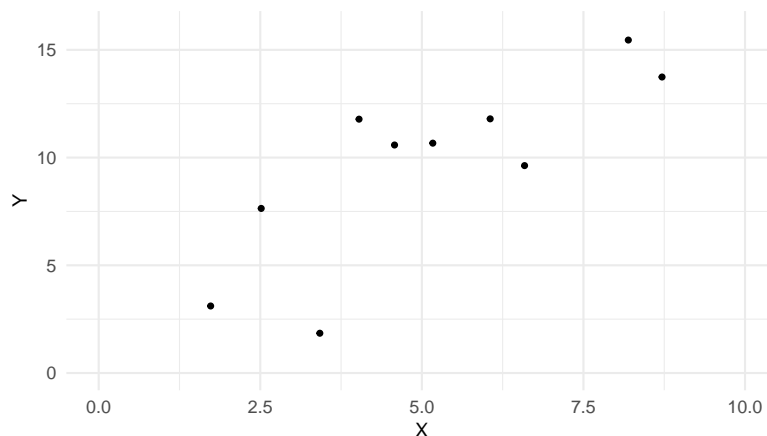


Figure 2.1: The data set `ols01.csv`.

There are many ways to fit a straight line through these points, including

- i. minimizing the sum of vertical distances from each point to the line;
- ii. minimizing the sum of horizontal distances from each point to the line;
- iii. minimizing the sum perpendicular distances from each point to the line;
- iv. use the square distances rather than the absolute distances;
- v. give different weights to each observation.

You can think of many other variations. For this exercise, we will minimize the sum of the *squared vertical* distances of each observation to the fitted line (we will refer to this as “ordinary least squares” or “OLS”). The vertical distances are marked out in Fig. 2.2. For the moment there is no particular reason for choosing this method, and no statistical / probabilistic / econometric meaning to this exercise at all. We are simply fitting a straight line to the data points, and exploring the mathematical properties of such a line. Everything we say in this section will apply to any set of data points  $\{X_i, Y_i\}_{i=1}^n$ , regardless of the source of the data.

Write the line to be fitted as

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$$

where  $\hat{\beta}_0$  and  $\hat{\beta}_1$  are the  $y$ -intercept and slope of the line respectively. Choosing a straight line means choosing values for these two objects. For each  $i = 1, 2, \dots, n$ , let  $\hat{Y}_i$  be the  $y$ -value of this line at  $X = X_i$  (the points  $(X_i, \hat{Y}_i)$  are shown on Fig. 2.2 as hollow circles). Then the vertical distances from the data points to the line, are  $Y_i - \hat{Y}_i$  which we will denote by  $\hat{\epsilon}_i$ :

$$\hat{\epsilon}_i = Y_i - \hat{Y}_i \quad \text{where} \quad \hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i.$$

We will call  $\hat{Y}_i$  the “fitted values” of  $Y$ , and  $\hat{\epsilon}_i$  the “residuals”. The residual for the fourth observation in our data set is marked out in Fig. 2.2.<sup>1</sup>

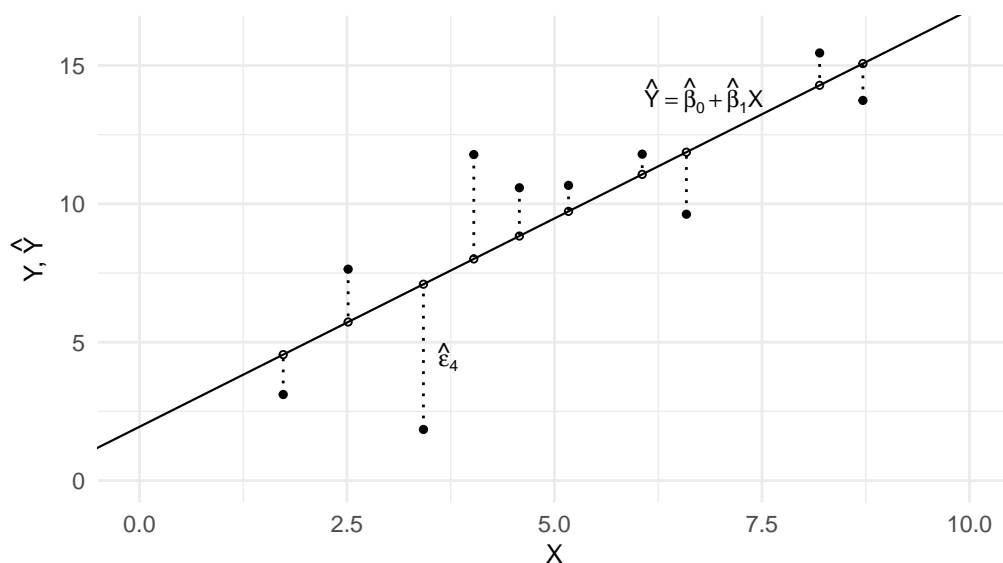


Figure 2.2: Fitting a Straight Line to Data by Least Squares.

<sup>1</sup>I’ll show code for most of the figures in this book, but I’ll skip the code for this figure as it is a bit distracting.



The sum of squared vertical distances, or “sum of squared residuals” can then be written as

$$SSR = \sum_{i=1}^N \hat{\epsilon}_i^2 = \sum_{i=1}^N (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i)^2. \quad (2.12)$$

We want to

- derive the formulas for  $\hat{\beta}_0$  and  $\hat{\beta}_1$  that minimizes the expression in (5.5).
- understand the algebraic properties of OLS estimators.

Once again, we are not making any statistical or econometric interpretations at this point.

Let  $\hat{\beta}_0^{ols}$  and  $\hat{\beta}_1^{ols}$  be the values of  $\hat{\beta}_0$  and  $\hat{\beta}_1$  that minimizes SSR in (5.5). These can be found by solving the first order conditions:

$$\begin{aligned} \left. \frac{\partial SSR}{\partial \hat{\beta}_0} \right|_{\hat{\beta}_0^{ols}, \hat{\beta}_1^{ols}} &= -2 \sum_{i=1}^N (Y_i - \hat{\beta}_0^{ols} - \hat{\beta}_1^{ols} X_i) = 0 \\ \left. \frac{\partial SSR}{\partial \hat{\beta}_1} \right|_{\hat{\beta}_0^{ols}, \hat{\beta}_1^{ols}} &= -2 \sum_{i=1}^N (Y_i - \hat{\beta}_0^{ols} - \hat{\beta}_1^{ols} X_i) X_i = 0 \end{aligned} \quad (2.13)$$

where the notation  $\left. \frac{\partial SSR}{\partial \hat{\beta}_0} \right|_{\hat{\beta}_0^{ols}, \hat{\beta}_1^{ols}}$  refers to the derivative  $\frac{\partial SSR}{\partial \hat{\beta}_0}$  evaluated at  $\hat{\beta}_0^{ols}$  and  $\hat{\beta}_1^{ols}$ , and likewise for  $\left. \frac{\partial SSR}{\partial \hat{\beta}_1} \right|_{\hat{\beta}_0^{ols}, \hat{\beta}_1^{ols}}$ .

We can solve the equations in (2.13) in the following way. Divide the first equation in (2.13) by  $N$  and solve for  $\hat{\beta}_0^{ols}$  to get

$$\hat{\beta}_0^{ols} = \bar{Y} - \hat{\beta}_1^{ols} \bar{X}. \quad (2.14)$$

Then substitute (2.14) into the second equation. This gives

$$\sum_{i=1}^N ((Y_i - \bar{Y}) - \hat{\beta}_1^{ols} (X_i - \bar{X})) X_i = 0$$

which we can solve to get

$$\hat{\beta}_1^{ols} = \frac{\sum_{i=1}^N (Y_i - \bar{Y}) X_i}{\sum_{i=1}^N (X_i - \bar{X}) X_i}. \quad (2.15)$$

We can substitute this expression back into (2.14) to get the full formula for  $\hat{\beta}_0^{ols}$ , but we'll leave (2.14) as it is.

We can show that the second order condition for a global minimum is satisfied, so that  $\hat{\beta}_0^{ols}$  and  $\hat{\beta}_1^{ols}$  does in fact minimize the SSR. This is left as an exercise.

The fitted line is therefore

$$\hat{Y} = \hat{\beta}_0^{ols} + \hat{\beta}_1^{ols} X$$

where  $\hat{\beta}_0^{ols}$  and  $\hat{\beta}_1^{ols}$  are as given in (2.14) and (2.15). For our data, we have

```
beta1hat = sum((df$X - mean(df$X)) * df$Y) / sum((df$X - mean(df$X))^2)
beta0hat = mean(df$Y) - beta1hat*mean(df$X)
cat("Intercept:", round(beta0hat,3), "; Slope:", round(beta1hat,3))
```

Intercept: 1.943 ; Slope: 1.506

### 2.4.1 Algebraic Properties

From this point on, we will refer to the fitted line as the “sample regression line”. The fitted values  $\hat{Y}_i$  and residuals  $\hat{\epsilon}_i$  should be understood to be the “least squares” or “OLS” fitted values and residuals, i.e.,

$$\hat{Y}_i = \hat{\beta}_0^{ols} + \hat{\beta}_1^{ols} X_i \quad \text{and} \quad \hat{\epsilon}_i = Y_i - \hat{Y}_i = Y_i - \hat{\beta}_0^{ols} - \hat{\beta}_1^{ols} X_i.$$

Ideally we ought to place “ols” superscripts on these, to distinguish them from fitted values and residuals obtained from other fitting methods, but we will neglect this detail. We will leave the superscripts on  $\hat{\beta}_0^{ols}$  and  $\hat{\beta}_1^{ols}$ . We call  $\{Y_i\}$  the regressand, and  $\{X_i\}$  the regressor.

The sample regression line has a number of useful algebraic properties.

[P1] If all of the  $X$  observations are of the same value, i.e.,  $X_1 = X_2 = \dots = X_N$ , then  $\bar{X}$  will have this same value, and  $\sum_{i=1}^N (X_i - \bar{X})X_i$  will be zero. As a result, we will not be able to compute (2.15) (nor (2.14)). This is the case where all of your data points line up in a vertical straight line.

[P2] Since

$$\sum_{i=1}^N (X_i - \bar{X})(Y_i - \bar{Y}) = \sum_{i=1}^N (X_i - \bar{X})Y_i \quad \text{and} \quad \sum_{i=1}^N (X_i - \bar{X})^2 = \sum_{i=1}^N (X_i - \bar{X})X_i,$$

we can write (2.15) as

$$\hat{\beta}_1^{ols} = \frac{\sum_{i=1}^N (X_i - \bar{X})Y_i}{\sum_{i=1}^N (X_i - \bar{X})X_i} = \frac{\sum_{i=1}^N (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^N (X_i - \bar{X})^2}. \quad (2.16)$$

If you divide both numerator and denominator by  $N - 1$ , the numerator becomes the sample covariance of  $X_i$  and  $Y_i$ , and the denominator becomes the sample variance of  $X_i$ .

[P3] The first equation in the first order condition (2.13) can be written as

$$\sum_{i=1}^N (Y_i - \hat{\beta}_0^{ols} - \hat{\beta}_1^{ols} X_i) = \sum_{i=1}^N \hat{\epsilon}_i = 0. \quad (2.17)$$

It follows that the sample mean of the least squares residuals is zero.

[P4] The second equation in the first order condition (2.13) can be written as

$$\sum_{i=1}^N (Y_i - \hat{\beta}_0^{ols} - \hat{\beta}_1^{ols} X_i)X_i = \sum_{i=1}^N \hat{\epsilon}_i X_i = 0. \quad (2.18)$$

We say that the OLS residuals  $\hat{\epsilon}_i$  and the regressors  $X_i$  are **orthogonal**. This result and [P3] imply that the fitted values and the residuals are also orthogonal.

$$\sum_{i=1}^N \hat{Y}_i \hat{\epsilon}_i = \hat{\beta}_0^{ols} \sum_{i=1}^N \hat{\epsilon}_i + \hat{\beta}_1^{ols} \sum_{i=1}^N X_i \hat{\epsilon}_i = 0. \quad (2.19)$$

[P5] Because the residuals have sample mean zero, it follows from (2.18) that the sample covariance between the residuals and the regressors is zero. This is because the sample covariance is  $\frac{1}{N} \sum_{i=1}^N (\hat{\epsilon}_i - \bar{\epsilon}_i)(X_i - \bar{X})$  and

$$\begin{aligned} \sum_{i=1}^N (\hat{\epsilon}_i - \bar{\epsilon}_i)(X_i - \bar{X}) &= \sum_{i=1}^N (\hat{\epsilon}_i - \bar{\epsilon}_i)X_i \quad (\text{why?}) \\ &= \sum_{i=1}^N \hat{\epsilon}_i X_i \quad (\text{why?}) \end{aligned}$$

[P6] Summing the equation

$$Y_i = \hat{\beta}_0^{ols} + \hat{\beta}_1^{ols} X_i + \hat{\epsilon}_i$$

over  $i = 1, 2, \dots, N$  and dividing by  $N$  gives

$$\bar{Y} = \hat{\beta}_0^{ols} + \hat{\beta}_1^{ols} \bar{X}$$

since the residuals have sample mean zero. This means that the sample regression line (the fitted line) passes through the point  $(\bar{X}, \bar{Y})$ .

[P7] Similarly, taking sample means on both sides of  $Y_i = \hat{Y}_i + \hat{\epsilon}_i$  gives

$$\bar{Y} = \bar{\hat{Y}}, \quad \text{where } \bar{\hat{Y}} = (1/N) \sum_{i=1}^N \hat{Y}_i.$$

In words, the OLS fitted values and the regressand both have the same sample mean.

[P8] We have the useful decomposition

$$\sum_{i=1}^N (Y_i - \bar{Y})^2 = \sum_{i=1}^N (\hat{Y}_i - \bar{\hat{Y}})^2 + \sum_{i=1}^N \hat{\epsilon}_i^2. \quad (2.20)$$

We read (2.20) as “Sum of Squared Total = Sum of Squared Explained + Sum of Squared Residuals” or “SST = SSE + SSR”. It is essentially a variance decomposition result. To get this result, subtract  $\bar{Y}$  on both sides, then square and sum both sides:

$$\begin{aligned} Y_i &= \hat{Y}_i + \hat{\epsilon}_i \\ Y_i - \bar{Y} &= \hat{Y}_i - \bar{Y} + \hat{\epsilon}_i \\ (Y_i - \bar{Y})^2 &= (\hat{Y}_i - \bar{Y})^2 + \hat{\epsilon}_i^2 + 2(\hat{Y}_i - \bar{Y})\hat{\epsilon}_i \\ \sum_{i=1}^N (Y_i - \bar{Y})^2 &= \sum_{i=1}^N (\hat{Y}_i - \bar{Y})^2 + \sum_{i=1}^N \hat{\epsilon}_i^2 + 2 \sum_{i=1}^N (\hat{Y}_i - \bar{Y})\hat{\epsilon}_i. \end{aligned}$$

Since  $\bar{Y} = \bar{\hat{Y}}$ , replace  $\sum_{i=1}^N (\hat{Y}_i - \bar{Y})^2$  with  $\sum_{i=1}^N (\hat{Y}_i - \bar{\hat{Y}})^2$ . Complete the proof by noting that

$$\sum_{i=1}^N (\hat{Y}_i - \bar{Y})\hat{\epsilon}_i = \sum_{i=1}^N \hat{Y}_i \hat{\epsilon}_i - \bar{Y} \sum_{i=1}^N \hat{\epsilon}_i = 0.$$

The identity in (2.20) forms the basis of the classic “goodness-of-fit”  $R^2$  measure. We can think of the SST as a measure of the total “variation” in  $Y_i$ . Dividing by  $N - 1$  gives you the sample variance of  $Y_i$ . The SST is decomposed into the total “variation in  $\hat{Y}_i$  and the residuals. Dividing (2.20) throughout by SST, we have

$$1 = \frac{SSE}{SST} + \frac{SSR}{SST}$$

from which we can define

$$R^2 = 1 - \frac{SSR}{SST}. \quad (2.21)$$

Since  $1 - SSR/SST = SSE/SST$ , the  $R^2$  has the interpretation as the *proportion* of variation in  $Y_i$  that is accounted for (sometimes the word “explained” is used) by  $\hat{Y}_i$ , or by  $X_i$ , since  $\hat{Y}_i$  is just a linear function of  $X_i$ .

By construction,  $R^2$  lies between 0 and 1 (inclusive). An  $R^2$  of one indicates a perfect fit, since  $R^2 = 1$  only when  $SSR = 0$ , which means that  $\hat{\epsilon}_i = 0$  for all  $i$ . On the other hand,  $R^2 = 0$  when  $SSR = SST$ , which means that  $SSE = \sum_{i=1}^N (\hat{Y}_i - \bar{Y})^2 = 0$ , or  $\hat{Y}_i = \bar{Y}$  for all  $i$  (this implies also that  $\hat{\beta}_1 = 0$ ). All intermediate fits result in values of  $R^2$  strictly between 0 and 1. For our data set and regression, we have

```
e_hat <- df$Y - Y_hat
SSR <- sum(e_hat^2) # we defined e_hat = residuals mdl earlier
SST <- sum((df$Y - mean(df$Y))^2)
Rsqr <- 1 - SSR/SST
print(paste("The R-squared for the sample regression line is:", round(Rsqr, 3)))
```

```
[1] "The R-squared for the sample regression line is: 0.644"
```

That is, the fitted line accounts for around 64.5 percent of the variation in  $Y_i$ .

As you might have guessed, there are built-in function in R for making all the computations demonstrated here. The usual function for calculating the coefficient values is `lm()` from the (auto-loaded) package `stats`.

```
mdl <- lm(Y~X, data=df) # Y~X means regress Y on X, including an intercept term.
coef(mdl) # "mdl" contains a lot of stuff, coef(mdl) picks out the coefficient vector.
```

```
(Intercept)          X
1.943349      1.506138
```

We can get the fitted values and residuals using

```
Y_hat <- fitted(mdl)
e_hat <- residuals(mdl)
```

We can extract the  $R^2$  from the `summary.lm` object returned when `summary()` function is applied to the `lm` object `mdl`:

```
summary(mdl)$r.squared
```

```
[1] 0.6444797
```

## 2.5 Exercises

**Exercise 2.36.** Show that  $\hat{\beta}_1^{ols}$  in (2.15) can be written as  $\hat{\beta}_1^{ols} = \frac{\sum_{i=1}^N X_i Y_i - N \bar{X} \bar{Y}}{\sum_{i=1}^N X_i^2 - N \bar{X}^2}$ .

**Exercise 2.37.** The second order condition for the OLS minimization of the SSR in (5.5) is that

$$H = \begin{bmatrix} \frac{\partial^2 SSR}{\partial \hat{\beta}_0^2} & \frac{\partial^2 SSR}{\partial \hat{\beta}_0 \partial \hat{\beta}_1} \\ \frac{\partial^2 SSR}{\partial \hat{\beta}_0 \partial \hat{\beta}_1} & \frac{\partial^2 SSR}{\partial \hat{\beta}_1^2} \end{bmatrix} \text{ is positive definite.}$$

i.e.,  $c^T H c > 0$  for all  $c \neq 0$ . Show that this condition is satisfied. *Hint: Show that*

$$H = 2 \begin{bmatrix} N & \sum_{i=1}^N X_i \\ \sum_{i=1}^N X_i & \sum_{i=1}^N X_i^2 \end{bmatrix} = 2(X^T X) \quad \text{where} \quad X = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_N \end{bmatrix}$$

and explain why this implies that  $H$  is positive definite, assuming  $\sum_{i=1}^N (X_i - \bar{X})^2 \neq 0$ .

**Exercise 2.38.** In order to fit the straight line to your data points by least squares, you require the condition that  $\sum_{i=1}^N (X_i - \bar{X})^2 \neq 0$ , i.e.,  $X_i$  cannot all be equal to a single constant value. What is  $\hat{\beta}_0^{ols}$  and  $\hat{\beta}_1^{ols}$  if this condition is met, but  $Y_i = c$  for all  $i = 1, \dots, N$ ? What is the  $R^2$  in this case?

**Exercise 2.39.** Suppose you fit a straight line to your data with the *additional constraint that the line must pass through the origin*. In other words, you fit the sample regression line

$$\hat{Y}_i = \hat{\beta}_1 X_i$$

to your data points. Find the value of  $\hat{\beta}_1$  that minimizes the sum of squared residuals, where the residuals are now  $\hat{\epsilon}_i = Y_i - \hat{\beta}_1 X_i$ .

Which algebraic properties of the sample regression line [P1]-[P8] continue to hold and which are lost? Will the  $R^2$  as defined in (2.21) still necessarily lie between 0 and 1? (Hint: it can go below 0, but why?) What does it mean if it goes below zero?

**Exercise 2.40.** (Continuing with the straight line passing through the origin.) The formula for  $\hat{\beta}_1$  when fitting the straight line with no intercept  $\hat{Y} = \hat{\beta}_1 X$  is

$$\hat{\beta}_1^{ols} = \frac{\sum_{i=1}^N X_i Y_i}{\sum_{i=1}^N X_i^2}.$$

For this formula to be feasible, we require  $\sum_{i=1}^N X_i^2 \neq 0$ , but *OLS estimation is still feasible if*  $\sum_{i=1}^N (X_i - \bar{X})^2 = 0$ . In other words, we no longer require variation in the  $X_i$ . Why is this so?



## Chapter 3

### Probability and Expectations Review

We describe the outcomes of “experiments” as random variables. By “experiment”, we mean any sort of activity (human or otherwise) that can result in a range of outcomes, and such that which outcomes occur can be thought of (at some level) as random. This chapter contains a few examples to review the idea of random variables and related concepts. The R code in this chapter uses the `tidyverse` and `patchwork` packages.

```
library(tidyverse)
library(patchwork)
library(latex2exp)
```

#### 3.1 Random Variables, Mean and Variance

For our first example, suppose there are two urns A and B each containing 100 balls numbered 1 to 10. We will call a ball that is numbered  $i$  an “ $i$ -ball” (so we have 1-balls, 2-balls, 3-balls, and so on). The number of  $i$ -balls in each urn is shown below.

```
UrnA <- c(3, 4, 8, 15, 20, 20, 15, 8, 4, 3);
UrnB <- c(16, 50, 16, 8, 4, 2, 1, 1, 1, 1);
ballnum = 1:10
ballnames <- paste0(ballnum, "-Ball")
names(UrnA) <- ballnames
names(UrnB) <- ballnames
cat("Urn A:\n"); UrnA
cat("Urn B:\n"); UrnB
```

Urn A:

1-Ball	2-Ball	3-Ball	4-Ball	5-Ball	6-Ball	7-Ball	8-Ball	9-Ball	10-Ball
3	4	8	15	20	20	15	8	4	3

Urn B:

1-Ball	2-Ball	3-Ball	4-Ball	5-Ball	6-Ball	7-Ball	8-Ball	9-Ball	10-Ball
16	50	16	8	4	2	1	1	1	1

Suppose you pick one ball at random from Urn A (suppose the balls are well mixed, you don’t look, etc.). It seems reasonable to think that the chance, or probability, of drawing a 1-ball, 2-ball, 3-ball, etc. is 0.03, 0.04, 0.08, and so on. If  $X$  be the number on the ball that is drawn, then  $X$  is a **random variable**, with **probability distribution function (pdf)**  $p(x) = Pr[X = x]$  as shown in Fig. 3.1(a) below. Likewise, if  $Y$  is the number on a ball drawn from Urn B, then it too is a random variable, with pdf as shown in Fig. 3.1(b).

```
pdfX = data.frame(x=ballnum, probx=UrnA/100)
pdfY = data.frame(y=ballnum, proby=UrnB/100)
my_theme <- theme_bw() + theme(axis.title=element_text(size=10), aspect.ratio = 0.8)
p1 <- pdfX %>% mutate(x=as.factor(x)) %>% ggplot(aes(x=x, y=probx)) + ggtitle("Urn A") +
  geom_bar(stat="identity", width=0.2) + ylim(0,0.52) + ylab("P(X=x)") + my_theme
p2 <- pdfY %>% mutate(y=as.factor(y)) %>% ggplot(aes(x=y, y=proby)) + ggtitle("Urn B") +
  geom_bar(stat="identity", width=0.2) + ylim(0,0.52) + ylab("P(Y=y)") + my_theme
(p1|p2)
```

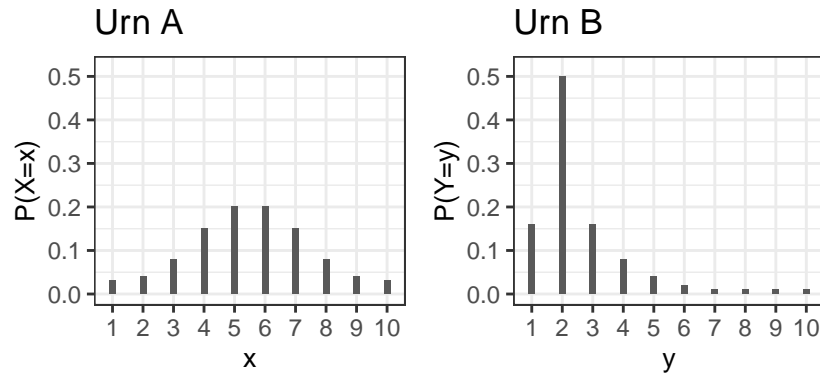


Figure 3.1: Probability distributions of value of ball from Urn A and Urn B.

It seems the “central location” as well as the “spread” of the two probability distributions are quite different. Can we come up with What indicators can we use to describe these features?

Suppose a random variable  $X$  takes possible values  $x_i$  with probabilities  $p(x_i) = \Pr[X = x_i]$ ,  $i = 1, 2, \dots, N$ . For central location the following indicators are popular:

$$\text{mean: } E[X] = \sum_{i=1}^N x_i \Pr[X = x_i] ,$$

$$\text{median: } \text{mdn}[X] = m \text{ such that } \Pr[X \leq m] = 0.5 \text{ and } \Pr[X \geq m] = 0.5 , \quad (3.1)$$

$$\text{mode: } \text{mo}[X] = x_j \text{ such that } \Pr[X = x_j] \geq \Pr[X = x_i] \text{ for all } i = 1, 2, \dots, N .$$

The definitions are for a random variable taking  $n$  possible values. Later we extend the definition to other kinds of random variables.

The **mean**, also called the **expected value**, is a weighted average. Imagine that the probabilities are weights resting on a plank under which you place a pivot. The mean is the location of the pivot such that the probabilities on either side balances and the plank rests horizontally on the pivot. For this reason, the mean is called a **moment**. For our random variables:

```
mu_X <- sum(pdfX$x*pdfX$probx)
mu_Y <- sum(pdfY$y*pdfY$proby)
cat("E[X] =", round(mu_X,3), "and E[Y] =", round(mu_Y,3), ".")
```

$E[X] = 5.5$  and  $E[Y] = 2.62$  .

The definition of the mean is easy to extend to functions of  $X$ , e.g., the mean of  $g(X)$  is just

$$E[g(X)] = \sum_{i=1}^N g(x_i) \Pr[X = x_i] . \quad (3.2)$$

The median is a value such that the probabilities on either side of it add to 0.5. One problem with this measure is that there may be many values or none (for  $X$ , all values from 5 to 6 inclusive are medians, and we cannot calculate the median of  $Y$ ). The mode is a value that has the highest probability. Again, the mode may not be unique. The mode of  $Y$  is 2, but the modes of  $X$  are 5 and 6.



For spread, a popular indicator is the variance:

$$\begin{aligned}\text{variance: } \text{var}[X] &= \sum_{i=1}^n (x_i - E[X])^2 \Pr[X = x_i] \\ &= E[(X - E[X])^2].\end{aligned}\tag{3.3}$$

The second line in Eq. 3.3 is just a restatement of the first, using the definition of an expectation.

The **variance** is the probability-weighted average of the squared deviations of the possible values from the mean. If the bulk of the probabilities are on values near the mean, then small  $(x_i - E[X])^2$  will be given more weight and the variance will be smaller. For our random variables:

```
var_X <- sum((pdfX$x-mu_X)^2*pdfX$probx)
var_Y <- sum((pdfY$y-mu_Y)^2*pdfY$proby)
cat("var[X] =", round(var_X,3), "and var[Y] =", round(var_Y,3), ".")
```

var[X] = 3.97 and var[Y] = 2.676 .

Expanding the square in Eq. 3.3, we get an alternative way of computing the variance:

$$\begin{aligned}\sum_{i=1}^n (x_i - E[X])^2 \Pr[X = x_i] &= \sum_{i=1}^n (x_i^2 - 2E[X]x_i + E[X]^2) \Pr[X = x_i] \\ &= \sum_{i=1}^n x_i^2 \Pr[X = x_i] - 2E[X] \sum_{i=1}^n x_i \Pr[X = x_i] + E[X]^2 \sum_{i=1}^n \Pr[X = x_i] \\ &= \sum_{i=1}^n x_i^2 \Pr[X = x_i] - 2E[X]^2 + E[X]^2 \\ &= E[X^2] - E[X]^2\end{aligned}\tag{3.4}$$

This version is often easier to use. The variance is also known as the “second central moment”. The square root of the variance is the **standard deviation**. There are other measures such as the “inter-quartile range” but we won’t be using these.

The mean and variance are popular measures because they are easy to work with and manipulate. For instance, it is easy to show from the definitions of the mean and variance in Eq. 3.1 and Eq. 3.3 that for any random variable  $X$ , we have

- $E[aX + b] = aE[X] + b$
- $\text{var}[aX + b] = a^2 \text{var}[X]$

These are left as an exercise.

Suppose the game is: you get to pick one ball from one urn, and you win in dollars 100 times the number on the ball. To pick from Urn A, you first have to pay \$500. To pick from Urn B, you only pay \$250. Which urn would you choose to play, or would you not play? (Just asking.)

### 3.2 Joint and Conditional Distributions

Suppose you live in a place with no seasons, and seemingly random variations in temperature and precipitation from day to day, with the following probabilities.

```
Weather = matrix(c(0.05, 0.15, 0.35, 0.10, 0.15, 0.20), nrow=2, byrow=T)
colnames(Weather) <- c("Cool", "Warm", "Hot")
rownames(Weather) <- c("Dry", "Rainy")
cat("Joint Probability Distribution of Temperature and Precipitation:\n")
Weather
SumWeather <- sum(Weather)
cat("Total Probabilities = ", SumWeather)
```

Joint Probability Distribution of Temperature and Precipitation:

	Cool	Warm	Hot
Dry	0.05	0.15	0.35
Rainy	0.10	0.15	0.20
Total Probabilities =	1		

This is an example of a **joint probability distribution**<sup>1</sup> for two variables: Temperature (Temp) taking values ‘Cool’, ‘Warm’ and ‘Hot’, and Precipitation (Prpc) taking values ‘Dry’ and ‘Rainy’. It gives the probabilities of joint Temperature-Precipitation events like ‘Cool and Rainy’, ‘Hot and Dry’, and so on. In notation

$$\Pr[\text{Temp} = \text{Cool}, \text{Prpc} = \text{Dry}] = 0.05, \Pr[\text{Temp} = \text{Warm}, \text{Prpc} = \text{Rainy}] = 0.15, \text{ etc.}$$

with probabilities adding to one. So, out of every hundred days, we should see about 20 Hot and Rainy days, 5 Cool and Dry days, etc. In what follows, I will just write  $\Pr[\text{Cool}, \text{Dry}]$  for  $\Pr[\text{Temp} = \text{Cool}, \text{Prpc} = \text{Dry}]$ .

What is the probability of Dry days (regardless of temperature)? “Dry regardless of temperature” means Dry & Cool, Dry & Warm, and Dry & Hot all count, so we add up the probabilities in the first row to get  $\Pr[\text{Dry}] = 0.55$ . Likewise,  $\Pr[\text{Rainy}] = 0.45$ . These two probabilities make up the **marginal**, or **unconditional**, probability distribution for Precipitation. To get the marginal or unconditional probability distribution for Temperature, we add up the columns. We get:  $\Pr[\text{Cool}] = 0.15$ ,  $\Pr[\text{Warm}] = 0.3$ ,  $\Pr[\text{Hot}] = 0.55$ . We show these two marginal distributions in the last column and row of the table below:

```
cat("Joint and Marginal Probability Distribution of Prec and Temp:\n")
MargPrpc <- rowSums(Weather)
MargTemp <- colSums(Weather)
WeatherWithMarg <- cbind(Weather, MargPrpc)
WeatherWithMarg <- rbind(WeatherWithMarg, MargTemp=c(MargTemp, SumWeather) )
WeatherWithMarg
```

Joint and Marginal Probability Distribution of Prec and Temp:

	Cool	Warm	Hot	MargPrpc
Dry	0.05	0.15	0.35	0.55
Rainy	0.10	0.15	0.20	0.45
MargTemp	0.15	0.30	0.55	1.00

---

<sup>1</sup>This is obviously a simplified description of the weather, discretized for expositional purposes!

We can ask a different sort of question about the weather in this place. E.g., what is the precipitation like on Cool days? From the “Cool” column in the joint distribution, we see that on Cool days it is twice as likely to be Rainy than Dry. We want to describe precipitation on Cool days as a probability distribution, and total probabilities must add to one, so we divide the “Cool” column of the joint distribution by  $\Pr[\text{Cool}]$ . This gives the **Conditional Probability Distribution** of Precipitation given Temperature = Cool. In notation:

$$\begin{aligned}\Pr[\text{Dry} \mid \text{Cool}] &= \frac{\Pr[\text{Dry, Cool}]}{\Pr[\text{Cool}]} = \frac{0.05}{0.15} = 1/3 \\ \Pr[\text{Rainy} \mid \text{Cool}] &= \frac{\Pr[\text{Rainy, Cool}]}{\Pr[\text{Cool}]} = \frac{0.1}{0.15} = 2/3\end{aligned}\tag{3.5}$$

We can make similar calculations for Warm and Hot days to get the conditional distributions for Precipitation given Temperature = Warm, and given Temperature = Hot. All three conditional distributions are shown in the columns of the output below:

```
PrpcGivenTemp = Weather
PrpcGivenTemp[, 'Cool'] <- Weather[, 'Cool'] / MargTemp['Cool']
PrpcGivenTemp[, 'Warm'] <- Weather[, 'Warm'] / MargTemp['Warm']
PrpcGivenTemp[, 'Hot'] <- Weather[, 'Hot'] / MargTemp['Hot']
rownames(PrpcGivenTemp) <- c("Pr[Dry|Temp]", "Pr[Rainy|Temp]")
cat("Conditional Distribution of Precipitation Given Temperature (columns):\n")
round(PrpcGivenTemp,3)
```

Conditional Distribution of Precipitation Given Temperature (columns):

	Cool	Warm	Hot
Pr[Dry Temp]	0.333	0.5	0.636
Pr[Rainy Temp]	0.667	0.5	0.364

On warm days, it is fifty-fifty whether it is Dry or Rainy. On Hot days, it is more likely to be Dry than Rainy. We emphasize that each column is a distribution, so the conditional distribution of Precipitation given Temperature is really a collection of three distribution different distributions.

Similar calculations gives the conditional distribution of Temperature given Precipitation, e.g., what are the probabilities each of Cool, Warm, and Hot days given it is Dry? By the same reasoning as before,

$$\begin{aligned}\Pr[\text{Cool} \mid \text{Dry}] &= \frac{\Pr[\text{Dry, Cool}]}{\Pr[\text{Dry}]} = \frac{0.05}{0.55} = 1/11 \\ \Pr[\text{Warm} \mid \text{Dry}] &= \frac{\Pr[\text{Dry, Warm}]}{\Pr[\text{Dry}]} = \frac{0.15}{0.55} = 3/11 \\ \Pr[\text{Hot} \mid \text{Dry}] &= \frac{\Pr[\text{Dry, Hot}]}{\Pr[\text{Dry}]} = \frac{0.35}{0.55} = 7/11\end{aligned}\tag{3.6}$$

Likewise for Rainy days. The two conditional distributions of temperature given precipitation are shown in the rows of the output below, with probabilities rounded to 3 decimal places.

```

TempGivenPrpcp = Weather
TempGivenPrpcp['Dry',] <- Weather['Dry',] / MargPrpcp['Dry']
TempGivenPrpcp['Rainy',] <- Weather['Rainy',] / MargPrpcp['Rainy']
colnames(TempGivenPrpcp) <- c("Pr[Cool|Prpcp]", "Pr[Warm|Prpcp]", "Pr[Hot|Prpcp]")
cat("Conditional Distribution of Temperature Given Precipitation (rows):\n")
round(TempGivenPrpcp,3)

```

Conditional Distribution of Temperature Given Precipitation (rows):

	Pr[Cool Prpcp]	Pr[Warm Prpcp]	Pr[Hot Prpcp]
Dry	0.091	0.273	0.636
Rainy	0.222	0.333	0.444

### 3.2.1 Bayes' Theorem

There is a tidy relationship between the conditional distributions. From the first equations of Eq. 3.5 and Eq. 3.6, we can deduce that

$$\Pr[\text{Dry} | \text{Cool}] \Pr[\text{Cool}] = \Pr[\text{Cool} | \text{Dry}] \Pr[\text{Dry}] ,$$

since both are equal to  $\Pr[\text{Cool}, \text{Dry}]$ . It follows that

$$\Pr[\text{Dry} | \text{Cool}] = \frac{\Pr[\text{Cool} | \text{Dry}] \Pr[\text{Dry}]}{\Pr[\text{Cool}]} . \quad (3.7)$$

In our example, we found that  $\Pr[\text{Dry} | \text{Cool}] = 1/3$  whereas  $\Pr[\text{Cool} | \text{Dry}] = 1/11$  which goes to show that the two can be very different. Furthermore we can verify that  $1/3 = ((1/11) \times 0.55)/0.15$ , where  $\Pr[\text{Dry}] = 0.55$  and  $\Pr[\text{Cool}] = 0.15$ .

Since

$$\begin{aligned} \Pr[\text{Cool}] &= \Pr[\text{Cool}, \text{Dry}] + \Pr[\text{Cool}, \text{Rainy}] \\ &= \Pr[\text{Cool} | \text{Dry}] \Pr[\text{Dry}] + \Pr[\text{Cool} | \text{Rainy}] \Pr[\text{Rainy}] , \end{aligned}$$

another version of Eq. 3.7 is

$$\Pr[\text{Dry} | \text{Cool}] = \frac{\Pr[\text{Cool} | \text{Dry}] \Pr[\text{Dry}]}{\Pr[\text{Cool} | \text{Dry}] \Pr[\text{Dry}] + \Pr[\text{Cool} | \text{Rainy}] \Pr[\text{Rainy}]} .$$

You can write similar equations for the other combinations of temperature and precipitation.

Eq. 3.7 is an example of **Bayes' Theorem**. If  $A$  and  $B$  are two events such that  $\Pr[A] \neq 0$  and  $\Pr[B] \neq 0$ , then

$$\Pr[A|B] = \frac{\Pr[B|A] \Pr[A]}{\Pr[B]} .$$

Likewise, we have

$$\Pr[B|A] = \frac{\Pr[A|B] \Pr[B]}{\Pr[A]} .$$

The following example might be a helpful illustration of the theorem.

**Example 3.1.** Imagine that an infectious disease enters a population of 101,000 people but that a large percentage – 100,000 out of 101,000 members of the population have vaccinated themselves against this disease. Of the 1000 not vaccinated, 50 percent (500 people) caught the disease. Only one percentage of these 100,000 vaccinated people (1000 people) eventually caught the disease, so the vaccine is effective.

Of those that caught the disease, more were vaccinated than un-vaccinated, but this is simply because a large proportion of the population received the vaccine, and 1 percent of 100,000 is more than 50 percent of 1000. The proportion of those that caught the disease being vaccinated (1000 out of 1500, or 0.67) is analogous to the *probability of having been vaccinated conditional on catching the disease*,  $\Pr[\text{vaccinated} | \text{infected}]$ , but this proportion doesn't say much about the effectiveness of the vaccine, if that is what we are interested in. What we want instead is the *probability of being infected conditional on being vaccinated*, i.e.,  $\Pr[\text{infected} | \text{vaccinated}]$ , which is 1000/100,000.

How do we get from

$$\Pr[\text{vaccinated} | \text{infected}] = \frac{1000}{1500} \quad \text{to} \quad \Pr[\text{infected} | \text{vaccinated}] = \frac{1000}{100,000} ?$$

If we multiply by the ratio of the percentage of the population that caught the disease  $\Pr[\text{infected}]$  to the percentage of those who got vaccinated  $\Pr[\text{vaccinated}]$ , then

$$\begin{aligned} \Pr[\text{infected} | \text{vaccinated}] &= \frac{\Pr[\text{vaccinated} | \text{infected}] \Pr[\text{infected}]}{\Pr[\text{vaccinated}]} \\ &= \frac{\frac{1000}{1500} \frac{1500}{101,000}}{\frac{100,000}{101,000}} = \frac{1000}{100,000} = 0.01. \end{aligned}$$

### 3.2.2 Mean and Variance, Covariance

At this point we cannot calculate things like means and variance for precipitation and temperature because we have only expressed the outcomes as categories, not quantities. Suppose Dry = 0mm of rain per day, and Rainy = 30mm of rain per day, and Hot=34 deg C, Warm=28C and Cool=22C. Then the joint distribution is

$$p(\text{temp}, \text{prcp}) = \Pr[\text{Temp} = \text{temp}, \text{Prcp} = \text{prcp}],$$

where  $\text{temp} = 22, 28, 34$  and  $\text{prcp} = 0, 30$ , as defined in the table below:

```
prcp <- c(0,30); temp <- c(22, 28, 34)
rownames(Weather) <- paste0(prcp, "mm"); colnames(Weather) <- paste0(temp, "C")
cat("Joint Distribution Pr[Temp = temp, Prcp = prcp]\n")
Weather
```

```
Joint Distribution Pr[Temp = temp, Prcp = prcp]
      22C  28C  34C
0mm  0.05 0.15 0.35
30mm 0.10 0.15 0.20
```

The marginal distributions are

```
names(MargPrcp) <- NULL
tbl <- cbind(prcp, MargPrcp); colnames(tbl) <- c("prcp(mm)", "Pr[Prcp=prcp]"); tbl
cat("\n")
names(MargTemp) <- NULL
tbl <- cbind(temp, MargTemp); colnames(tbl) <- c("temp(C)", "Pr[Temp=temp]"); tbl
```

```
      prcp(mm) Pr[Prcp=prcp]
[1,]         0          0.55
[2,]        30          0.45
```

```
      temp(C) Pr[Temp=temp]
[1,]        22          0.15
[2,]        28          0.30
[3,]        34          0.55
```

From these we can calculate the mean and variance of Prcp and Temp:

```
muPrcp <- sum(prcp*MargPrcp); muTemp <- sum(temp*MargTemp)
varPrcp <- sum((prcp-muPrcp)^2*MargPrcp); varTemp <- sum((temp-muTemp)^2*MargTemp)
cat(" E[Prcp] =", muPrcp, "mm"); cat(" "); cat(" E[Temp] =", muTemp, "C\n");
cat("var[Prcp] =", varPrcp, "sqr mm"); cat(" "); cat("var[Temp] =", varTemp, "sqr C\n");
cat(" sd[Prcp] =", round(sqrt(varPrcp),2), "mm"); cat(" ");
cat(" sd[Temp] =", round(sqrt(varTemp),2), "C")
```

```
E[Prcp] = 13.5 mm          E[Temp] = 30.4 C
var[Prcp] = 222.75 sqr mm  var[Temp] = 19.44 sqr C
sd[Prcp] = 14.92 mm       sd[Temp] = 4.41 C
```

Note the units of measurements on these moments, which are often not included when the moments are reported.

Do you notice a negative relationship between temperature and precipitation? One indicator of such an association is the **covariance** between two variables  $X$  and  $Y$  defined as

$$\begin{aligned} cov[X, Y] &= \sum_{i=1}^N \sum_{j=1}^M (x_i - E[X])(y_j - E[Y]) \Pr[X = x_i, Y = y_j] \\ &= E[(X - E[X])(Y - E[Y])] \\ &= E[XY] - E[X]E[Y] \end{aligned} \tag{3.8}$$

where we assume  $X$  has  $N$  possible values and  $Y$  has  $M$  possible values. The second expression is just a restatement of the first. Notice that the “outer” expectation is with respect to the *joint* probabilities. To make this clear, we might write

$$cov[X, Y] = E_{X,Y}[(X - E_X[X])(Y - E_Y[Y])],$$

but the subscripts are usually omitted. For  $E_X[X]$  and  $E_Y[Y]$ , it doesn’t matter whether you

use the marginal or joint probabilities, since

$$\begin{aligned} E_{X,Y}[X] &= \sum_{i=1}^N \sum_{j=1}^M x_i \Pr[X = x_i, Y = y_j] \\ &= \sum_{i=1}^N x_i \sum_{j=1}^M \Pr[X = x_i, Y = y_j] = \sum_{i=1}^N x_i \Pr[X = x_i] = E_X[X]. \end{aligned}$$

The third expression in Eq. 3.8 can be obtained by multiplying  $x_i - E[X]$  and  $y_j - E[Y]$  in the first expression.

Eq. 3.8 works in capturing positive or negative associations in the following way: if there is more probability placed on events where  $x_i$  and  $y_j$  are both above or both below their means, then events where the product of the deviation from means is positive is given more weight, and the covariance will be positive. If more probability is placed on events where  $x_i$  and  $y_j$  are on opposites sides of their means, then events where the product of the deviation from means is negative is given more weight, and the covariance will be negative. For our example:

```
# Find out what the outer() function does!
devfrmmeans <- outer(prcp - muPrcp, temp - muTemp)
covar <- sum(devfrmmeans * Weather)
cat("cov[Temp,Prcp] =", covar, "mm C")
```

```
cov[Temp,Prcp] = -14.4 mm C
```

so in fact yes, there is an association between lower temperatures and higher precipitation.

Notice again the unit of measurement on the covariance. By convention this is usually omitted, but covariance is not “scale-less”, and changing the unit of measurement changes the covariance. For instance, if we measure temperature in Fahrenheit ( $F = 32 + \frac{9}{5}C$ ), then the covariance becomes:

```
# Find out what the outer() function does!
devfrmmeans <- outer(prcp - muPrcp, (temp - muTemp)*9/5+32)
covar <- sum(devfrmmeans * Weather)
cat("cov[Temp(F),Prcp] =", round(covar,2), "mm F")
```

```
cov[Temp(F),Prcp] = -25.92 mm F
```

For this reason, we usually use the **correlation** instead of covariance, where

$$\text{corr}[X, Y] = \frac{\text{cov}[X, Y]}{\sqrt{\text{var}[X]}\sqrt{\text{var}[Y]}}.$$

This scales the covariance so that the result lies between  $-1$  and  $1$ , a consequence of an inequality called the “Cauchy-Schwarz Inequality”. Furthermore, the correlation is scale-less, so changing the unit of measure does not change its value. For our example:

```
corr <- covar / (sqrt(varPrcp)*sqrt(varTemp))
cat("corr[Temp,Prcp] =", round(corr,2))
```

```
corr[Temp,Prcp] = -0.39
```

The correlation is negative, but moderate.

### 3.2.3 Conditional Means and Variances

Our conditional distributions are:

```
tbl <- cbind(prcp,PrpcpGivenTemp)
rownames(tbl) <- NULL
colnames(tbl) <- c("prcp", "Pr[Prcp=prcp|Temp=22]", "Pr[Prcp=prcp|Temp=28]", "Pr[Prcp=prcp|Temp=34]")
cat("Conditional Distribution Pr[Prcp | Temp]\n")
round(tbl, 3)
cat("\n")
tbl <- cbind(temp,t(TempGivenPrcp))
rownames(tbl) <- NULL
colnames(tbl) <- c("temp", "Pr[Temp=temp|Prcp=0]", "Pr[Temp=temp|Prcp=30]")
cat("Conditional Distribution Pr[Temp | Prcp]\n")
round(tbl, 3)
```

Conditional Distribution Pr[Prcp | Temp]

	prcp	Pr[Prcp=prcp Temp=22]	Pr[Prcp=prcp Temp=28]	Pr[Prcp=prcp Temp=34]
[1,]	0	0.333	0.5	0.636
[2,]	30	0.667	0.5	0.364

Conditional Distribution Pr[Temp | Prcp]

	temp	Pr[Temp=temp Prcp=0]	Pr[Temp=temp Prcp=30]
[1,]	22	0.091	0.222
[2,]	28	0.273	0.333
[3,]	34	0.636	0.444

For any two random variables  $X$  and  $Y$  taking  $N$  and  $M$  values respectively, we can calculate the conditional means and variances from the conditional distribution, e.g., for any  $i$ ,

$$E[Y|X = x_i] = \sum_{j=1}^M y_j \Pr[Y = y_j|X = x_i],$$

$$\text{var}[Y|X = x_i] = \sum_{j=1}^M (y_j - E[Y|X = x_i])^2 \Pr[Y = y_j|X = x_i].$$

The conditional standard deviation is the square root of the conditional variance. These conditional moments tell us how  $Y$  behave when  $X$  is at some particular value. Is the expected value of  $Y$  higher or lower when  $X$  is higher? Is the variance of  $Y$  different at different values of  $X$ ?

The conditional mean and variance of Temperature on Dry and Rainy Days in our example are:

```
muTempGivenDry <- sum(temp*TempGivenPrcp["Dry",])
varTempGivenDry <- sum((temp-muTempGivenDry)^2*TempGivenPrcp["Dry",])
cat("E[Temp|Prcp=0] =", round(muTempGivenDry, 2), "C.\n")
cat("var[Temp|Prcp=0] =", round(varTempGivenDry, 2), "sqr C.\n")
cat("std.dev.[Temp|Prcp=0] =", round(sqrt(varTempGivenDry), 2), "C.\n")
```

E[Temp|Prcp=0] = 31.27 C.

var[Temp|Prcp=0] = 15.47 sqr C.

std.dev.[Temp|Prcp=0] = 3.93 C.



```
muTempGivenRainy <- sum(temp*TempGivenPrcp["Rainy",])
varTempGivenRainy <- sum((temp-muTempGivenRainy)^2*TempGivenPrcp["Rainy",])
cat("E[Temp|Prcp=30] =", round(muTempGivenRainy, 2), "C.\n")
cat("var[Temp|Prcp=30] =", round(varTempGivenRainy, 2), "sqr C.\n")
cat("std.dev.[Temp|Prcp=30] =", round(sqrt(varTempGivenRainy), 2), "C.\n")
```

E[Temp|Prcp=30] = 29.33 C.

var[Temp|Prcp=30] = 22.22 sqr C.

std.dev.[Temp|Prcp=30] = 4.71 C.

The mean temperature is slightly higher on Dry days than Rainy days, but the temperature variance is larger on Rainy days. The calculation of the conditional mean and variance of precipitation given temperature is left as an exercise for you.

Manipulation of conditional expectations and variances follows one simple principle: whatever is being conditioned on can be treated as “fixed” (i.e., like a constant) as far as that expectation or variance is concerned.

### Example 3.2.

- $E[aXY|X] = aXE[Y|X]$ ,  $var[aXY|X] = a^2X^2var[Y|X]$ ,
- $E[aX|X] = aX$  (contrast with  $E[aX] = aE[X]$ , a constant),
- $var[aX|X] = 0$  (contrast with  $var[aX] = a^2var[X]$ ),
- If  $Y = \beta_0 + \beta_1X + \epsilon$  with  $E[\epsilon|X] = 0$  and  $var[\epsilon|X] = \sigma^2$ , then

$$E[Y|X] = \beta_0 + \beta_1X \text{ and } var[Y|X] = \sigma^2.$$

### 3.2.4 Law of Iterated Expectations

In our weather example, we have two values of the conditional mean of Temp, one for Dry days and one for Rainy days. But precipitation is a random variable with probabilities  $\Pr[\text{Dry}] = 0.55$  and  $\Pr[\text{Rainy}] = 0.45$ . Therefore  $E[\text{Temp} | \text{Prcp}]$  is itself a random variable:

```
CondTempPrcp = cbind(prcp, MargPrcp, c(muTempGivenDry, muTempGivenRainy))
colnames(CondTempPrcp) <- c("prcp", "Pr(Prcp=prcp)", "E[Temp|Prcp=prcp]")
CondTempPrcp %>% round(2)
```

	prcp	Pr(Prcp=prcp)	E[Temp Prcp=prcp]
[1,]	0	0.55	31.27
[2,]	30	0.45	29.33

We can take the mean and variance of the conditional mean of Temp given Prcp, over all possible values of Prcp.

```
EETemp <- sum(CondTempPrcp[, "E[Temp|Prcp=prcp]"]*CondTempPrcp[, "Pr(Prcp=prcp)"])
varETemp <- sum((CondTempPrcp[, "E[Temp|Prcp=prcp]"]-EETemp)^2*CondTempPrcp[, "Pr(Prcp=prcp)"])
cat("E[E[Temp|Prcp]] =", EETemp, "\n")
```

E[E[Temp|Prcp]] = 30.4

```
cat("var[E[Temp|Prcp]] =", varETemp)
```

var[E[Temp|Prcp]] = 0.9309091

Notice that the  $E[E[\text{Temp} \mid \text{Prep}]] = E[\text{Temp}]$ . This is not a coincidence, but an instance of the **Law of Iterated Expectations**. Given two variables  $X$  and  $Y$  with  $N$  and  $M$  possible values respectively, we have

$$E_X[E_{Y|X}[Y|X]] = E_Y[Y].$$

The Law of Iterated Expectations says (roughly speaking) that we can get the ‘overall’ expectation of  $Y$  by taking the conditional expectations of  $Y$  for each possible value of  $X$ , and then taking the mean of all of those conditional expectations.

We demonstrate two results implied by the Law of Iterated Expectations:

- i. If  $E[Y|X] = c$ , then  $E[Y] = c$ ,
- ii. If  $E[Y|X] = c$ , then  $\text{cov}[X, Y] = 0$ .

Result (i) says that if the expected value of  $Y$  is  $c$  for every possible value of  $X$ , then the “overall” mean must be that same constant, and (ii) says that  $E[Y|X] = c$  is a sufficient condition for  $\text{cov}[X, Y] = 0$ .

The derivation of these results is straightforward: For (i), if  $E[Y|X] = c$ , then

$$E[Y] = E[E[Y|X]] = E[c] = c.$$

For (ii), we note that

$$E[YX] = E[E[YX|X]] = E[XE[Y|X]] = E[cX] = cE[X].$$

Therefore

$$\text{cov}[X, Y] = E[XY] - E[X]E[Y] = cE[X] - cE[X] = 0.$$

Although constant conditional mean implies zero covariance, the converse does not necessarily hold. For instance, suppose  $X$  is zero mean and has a symmetric distribution (which together implies that  $E[X^3] = 0$ ). Suppose  $Y = X^2$ . Then  $E[Y|X] = X^2$  but

$$\text{cov}[X, Y] = E[XY] - E[X]E[Y] = E[XE[Y|X]] - 0E[Y] = E[X^3] = 0.$$

Two quick remarks. First, the idea of conditional distributions, conditional mean, etc. can be extended to conditioning on more than one variable. The Law of Iterated Expectations can also be extended to more than two variables. For example, for random variables  $W$ ,  $X$  and  $Y$ , we have

$$E[X|Y] = E[E[X|W, Y]|Y].$$

Second, while we do not have a “Law of Iterated Variance”, we do have the following:

$$\text{var}[Y] = E[\text{var}[Y|X]] + \text{var}[E[Y|X]]. \quad (3.9)$$

### 3.2.5 Independent Random Variables

Consider two other places A and B with slightly different weather patterns. In Place A,

```
WeatherA = matrix(c(0.05, 0.3, 0.05, 0.2, 0.2, 0.2), nrow=2, byrow=T)
colnames(WeatherA) <- c("22C (Cool)", "28C (Warm)", "34C (Hot)")
rownames(WeatherA) <- c("0mm (Dry)", "30mm (Rainy)")
cat("Joint Probability Distribution of Temperature and Precipitation (Place A):\n")
WeatherA
SumWeatherA <- sum(WeatherA)
cat("Total Probabilities = ", SumWeatherA)
```

Joint Probability Distribution of Temperature and Precipitation (Place A):

	22C (Cool)	28C (Warm)	34C (Hot)
0mm (Dry)	0.05	0.3	0.05
30mm (Rainy)	0.20	0.2	0.20
Total Probabilities = 1			

The probability of Dry days on Place B is 0.4 and the probability of Rainy days is 0.6. Dividing each row by their respective total probabilities gives the conditional distribution of Temperature given Precipitation:

```
TempGivenPrpcA <- WeatherA
TempGivenPrpcA["0mm (Dry)", ] <-
  TempGivenPrpcA["0mm (Dry)", ] / sum(TempGivenPrpcA["0mm (Dry)", ])
TempGivenPrpcA["30mm (Rainy)", ] <-
  TempGivenPrpcA["30mm (Rainy)", ] / sum(TempGivenPrpcA["30mm (Rainy)", ])
cat("Conditional Probability Pr[Temperature | Precipitation] (Place A):\n")
round(TempGivenPrpcA,3)
```

Conditional Probability Pr[Temperature | Precipitation] (Place A):

	22C (Cool)	28C (Warm)	34C (Hot)
0mm (Dry)	0.125	0.750	0.125
30mm (Rainy)	0.333	0.333	0.333

Without much more calculations we can claim that

- $E[\text{Temp}|\text{Prpc} = 0] = E[\text{Temp}|\text{Prpc} = 30] = 28$  (why?)
- $\text{cov}[\text{Temp}, \text{Prpc}] = 0$ . (why?)

The first claim comes because the conditional distributions of Temp are symmetric about 28C. The second claim comes because the constant conditional mean implies zero covariance, which we proved earlier. However, if you calculate  $E[\text{Prpc}|\text{Temp} = \text{temp}]$  for  $\text{temp} = 22, 28, 34$ , you will find that the expected value of Precipitation given Temperature is *not* constant. This shows that while constant conditional expectation implies zero covariance, zero covariance does not imply constant conditional mean.

If we calculate the conditional variance of Temperature given Precipitation, we will get:

```
cat("var[Temp|Prpc=0] =", sum((temp-28)^2*TempGivenPrpcA["0mm (Dry)",]),"\n")
cat("var[Temp|Prpc=30] =", sum((temp-28)^2*TempGivenPrpcA["30mm (Rainy)",]))
```

```
var[Temp|Prpc=0] = 9
```

```
var[Temp|Prpc=30] = 24
```

So although Temperature and Precipitation are not correlated in Place A, the two variables are not unrelated. Temperature has a higher variance on Rainy Days than on Dry days.

Now we turn our attention to Place B, where

```
WeatherB = matrix(c(0.05, 0.15, 0.05, 0.15, 0.45, 0.15), nrow=2, byrow=T)
colnames(WeatherB) <- c("22C (Cool)", "28C (Warm)", "34C (Hot)")
rownames(WeatherB) <- c("0mm (Dry)", "30mm (Rainy)")
cat("Joint Probability Distribution of Temperature and Precipitation (Place B):\n")
WeatherB
SumWeatherB <- sum(WeatherB)
cat("Total Probabilities = ", SumWeatherB)
```

Joint Probability Distribution of Temperature and Precipitation (Place B):

	22C (Cool)	28C (Warm)	34C (Hot)
0mm (Dry)	0.05	0.15	0.05
30mm (Rainy)	0.15	0.45	0.15
Total Probabilities =	1		

The probability of Rainy days here is 0.75, and the probability of Dry days is 0.25. If we calculate the conditional distribution of Temperature given Precipitation, we get

```
TempGivenPrpB <- WeatherB
TempGivenPrpB["0mm (Dry)", ] <-
  TempGivenPrpB["0mm (Dry)", ] / sum(TempGivenPrpB["0mm (Dry)", ])
TempGivenPrpB["30mm (Rainy)", ] <-
  TempGivenPrpB["30mm (Rainy)", ] / sum(TempGivenPrpB["30mm (Rainy)", ])
cat("Conditional Probability Pr[Temperature | Precipitation] (Place B):\n")
round(TempGivenPrpB,3)
```

Conditional Probability Pr[Temperature | Precipitation] (Place B):

	22C (Cool)	28C (Warm)	34C (Hot)
0mm (Dry)	0.2	0.6	0.2
30mm (Rainy)	0.2	0.6	0.2

The conditional distributions are the same, which means the temperature distribution in Place B is the same regardless of precipitation, and in fact is the same as the unconditional distribution of temperature:

$$\Pr[\text{Temp} = \text{temp} | \text{Prp} = \text{prcp}] = \Pr[\text{Temp} = \text{temp}] \quad \text{for all } \text{temp}, \text{prcp}.$$

The conditional expectation and conditional variance of Temp given Prcp will also be the same here (since the conditional distributions are the same). Naturally the covariance will be zero.

In such a case, we say that Temperature and Precipitation are independent. Two random variables  $X$  and  $Y$  are **independent** if

$$\Pr[Y = y_j | X = x_i] = \Pr[Y = y_j] \quad \text{for all } i, j.$$

In this case it must also be that

$$\begin{aligned} \Pr[X = x_i | Y = y_j] &= \Pr[X = x_i] \quad \text{for all } i, j \quad (\text{why?}) \\ \Pr[X = x_i, Y = y_j] &= \Pr[X = x_i] \Pr[Y = y_j] \quad \text{for all } i, j \quad (\text{why?}) \end{aligned}$$

You can verify for yourself that  $\Pr[\text{Prp} = \text{prcp} | \text{Temp} = \text{temp}] = \Pr[\text{Prp} = \text{prcp}]$  for all temperatures and precipitation in Place B.

## 3.2.6 Exercises

**Exercise 3.1.** Show that  $E[aX + b] = aE[X] + b$  and  $\text{var}[aX + b] = a^2\text{var}[X]$  where  $X$  is a random variable and  $a$  and  $b$  are constants.

**Exercise 3.2.** Starting from the definition  $\text{cov}[X, Y] = E[(X - E[X])(Y - E[Y])]$  and using the properties of expectations, show that  $\text{cov}[X, Y] = E[XY] - E[X]E[Y]$ .

**Exercise 3.3.** For random variables  $X_1, X_2$  and  $X_3$  and constants  $a_1, a_2$  and  $a_3$ , show that

$$\begin{aligned} & \text{var}[a_1X_1 + a_2X_2 + a_3X_3] \\ &= a_1^2\text{var}[X_1] + a_2^2\text{var}[X_2] + a_3^2\text{var}[X_3] \\ & \quad + 2a_1a_2\text{cov}[X_1, X_2] + 2a_1a_3\text{cov}[X_1, X_3] + 2a_2a_3\text{cov}[X_2, X_3] \\ &= \sum_{i=1}^3 \sum_{j=1}^3 a_i a_j \text{cov}[X_i, X_j]. \end{aligned}$$

**Exercise 3.4.** Show that

$$\text{cov}[a_1X_1 + a_2X_2, b_1Y_1 + b_2Y_2 + b_3Y_3] = \sum_{i=1}^2 \sum_{j=1}^3 a_i b_j \text{cov}[X_i, Y_j].$$

**Exercise 3.5.** Explain why the correlation coefficient always lies between  $-1$  and  $1$ , inclusive.

*Hint: For arbitrary  $\alpha$ , we have  $\text{var}[X - \alpha Y] \geq 0$ . Expand  $\text{var}[X - \alpha Y]$  and let  $\alpha = \text{cov}[X, Y]/\text{var}[Y]$*

**Exercise 3.6.** Suppose  $X$  and  $Y$  are two random variables with the joint pdf

$$\begin{array}{c|ccccc} & 6 & 5.5 & 5 & 4.5 & 4 & 3.5 & 3 \\ Y & 0 & 0 & 0 & 0 & \frac{1}{20} & \frac{2}{20} & \frac{1}{20} \\ & 0 & 0 & 0 & \frac{1}{20} & \frac{2}{20} & \frac{1}{20} & 0 \\ & 0 & \frac{1}{20} & \frac{2}{20} & \frac{1}{20} & 0 & 0 & 0 \\ & \frac{1}{20} & \frac{2}{20} & \frac{1}{20} & 0 & 0 & 0 & 0 \\ & \frac{2}{20} & \frac{1}{20} & 0 & 0 & 0 & 0 & 0 \\ & \frac{1}{20} & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline & 1 & 2 & 3 & 4 & 5 & & \\ & & & & & & X & \end{array} \quad (3.10)$$

- Find the marginal distributions, means and variances of  $X$  and  $Y$ .
- Show that the covariance is equal to 1 and the correlation is equal to 0.8944.
- Find the conditional distribution, mean and variance of  $Y$  given  $X$ .
- Find the conditional distribution, mean and variance of  $X$  given  $Y$ .
- Are the two variables independent?

**Exercise 3.7.** Show that if  $E[Y|X] = a + bX$ , then

$$b = \frac{\text{cov}[X, Y]}{\text{var}[X]} \quad \text{and} \quad a = E[Y] - bE[X].$$

**Exercise 3.8.** Prove Eq. 3.9. Use this relationship to show that

- $\text{var}[Y] = E[\text{var}[Y|X]]$  if  $E[Y|X]$  is constant.
- $\text{var}[Y|X] \leq \text{var}[Y]$  if  $\text{var}[Y|X]$  is constant.

**Exercise 3.9.** Suppose  $Y$  and  $X$  have the following joint distribution function:

	10	0	0	0	0	$\frac{1}{10}$
	9	0	0	0	$\frac{1}{10}$	0
	8	0	0	$\frac{1}{10}$	0	0
	7	0	$\frac{1}{10}$	0	0	0
	6	$\frac{1}{10}$	0	0	0	0
Y	5	$\frac{1}{10}$	0	0	0	0
	4	0	$\frac{1}{10}$	0	0	0
	3	0	0	$\frac{1}{10}$	0	0
	2	0	0	0	$\frac{1}{10}$	0
	1	0	0	0	0	$\frac{1}{10}$
		1	2	3	4	5
				X		

- Find the marginal distribution of  $X$  and of  $Y$ .
- Find the conditional distribution, conditional mean, and conditional variance of  $Y$  given  $X$ .
- Find  $\text{cov}[X, Y]$ .
- In what way is the conditional distribution of  $Y$  related to  $X$ ?

**Exercise 3.10.** Suppose  $Y$  and  $X$  have the following joint pdf:

	5	0.01	0.04	0.03	0.01	0.01
	4	0.02	0.08	0.06	0.02	0.02
Y	3	0.04	0.16	0.12	0.04	0.04
	2	0.02	0.08	0.06	0.02	0.02
	1	0.01	0.04	0.03	0.01	0.01
		1	2	3	4	5
				X		

Are the variables independent? Are they identically distributed (i.e., do they have the same marginal distributions?) Change the probabilities in the joint pdf of  $X$  and  $Y$  so that the two variables are independently and identically distributed.

### 3.3 A Few More Distributions

We have seen the Bernoulli and Binomial Distributions, which are distributions of “discrete” random variables with finite number of possible outcomes. In this section we will look at a few random variables with countably infinite (discrete, but infinite) possible outcomes, as well as “uncountably infinite” number of possible outcomes (or outcomes over some continuum, like from 0 to 1). The latter type of random variables are called continuous random variables.

### 3.3.1 Geometric Distribution

**Example 3.3.** Suppose an experiment with two outcomes “Success” and “Failure” with probabilities  $p$  and  $1-p$  respectively is carried out a number of times. Suppose that the outcome of one experiment does not affect the outcome of subsequent experiments. Then the number of “Failures” before a “Success” is observed is a random variable  $X$  with the **geometric distribution**, i.e., with pdf

$$f_X(x) = p(1-p)^x, x = 0, 1, 2, \dots \quad (3.11)$$

We write  $X \sim \text{Geometric}(p)$ . Probabilities over all possible outcomes must sum to one, and we leave it as an exercise to show that this is the case for Eq. 3.11. Sometimes we describe the probabilistic behavior of a random variable using a **cumulative distribution function (cdf)** instead of a probability density function. The cdf is defined as

$$F_X(x) = \Pr[X \leq x], x \in \mathbb{R}.$$

For discrete random variables,  $F_X(x)$  is just the sum of all of the probabilities for  $X = x_i$  where  $x_i \leq x$ . For continuous random variables, we have

$$F_X(x) = \int_{-\infty}^x f_X(u) du.$$

Whereas the pdf is defined over the possible outcomes for  $X$ , the cdf is defined over the real line. Sometimes it is easier to use the pdf to describe the probabilistic behavior of a random variable, sometimes it is easier to use the cdf. For continuous variables the derivative of the cdf gives the pdf.

The cdf of a Geometric random variable is

$$F_X(x) = \begin{cases} 0, & x < 0 \\ 1 - (1-p)^{k+1}, & k \leq x < k+1, k = 0, 1, 2, \dots \end{cases} \quad (3.12)$$

Eq. 3.12 follows from the formula for geometric series:

$$\Pr[X \leq k] = \sum_{i=0}^k p(1-p)^i = \frac{p - p(1-p)^{k+1}}{1 - (1-p)} = 1 - (1-p)^{k+1}, k = 0, 1, 2, \dots$$

The Geometric pdf and cdf with 0.25 is shown in Fig. 3.2 (for  $x$  up to 10 only).

```
plot_theme <- theme_minimal() +
  theme(aspect.ratio = 1:1, plot.title = element_text(size = 12))

## Set parameter
p = 0.25
x <- 0:10
fx_geom <- dgeom(x,prob=p)
Fx_geom <- pgeom(x,prob=p)

## Plot the PDF
p1 <- ggplot() +
  geom_col(data=data.frame(x, fx_geom), aes(x=x, y=fx_geom), color="black", width=0.01) +
```

```

ggtitle('(a) pdf') + scale_x_continuous(breaks=seq(0,11,2)) +
ylim(0,0.3) + ylab(NULL) + plot_theme

Fx_geom_1 <- c(0,Fx_geom[1:length(x)-1])    ## Addition stuff for plotting discrete CDF
xstart <- c(-0.9,x)
xend <- c(x,10.4)
ystart <- c(0,Fx_geom)
yend <- c(Fx_geom_1,Fx_geom[length(x)])
df2a <- data.frame(x, Fx_geom)
df2b <- data.frame(x, Fx_geom_1)
df2c <- data.frame(xstart, ystart, xend, yend)
## Plot the CDF
p2 <- ggplot() +
  geom_point(data=df2a, aes(x=x,y=Fx_geom), color="black") +          # the black dots
  geom_point(data=df2b, aes(x=x,y=Fx_geom_1), color="black", shape=1) + # the hollow dots
  geom_segment(data=df2c, aes(x=xstart,y=ystart,xend=xend, yend=yend)) + # the vertical lines
  ggtitle('(b) cdf') + scale_x_continuous(breaks=seq(0,10.4,2)) +
  ylim(0,1) + ylab(NULL) + plot_theme
p1 | p2 + plot_annotation("Geometric PDF and CDF with p=0.25")

```

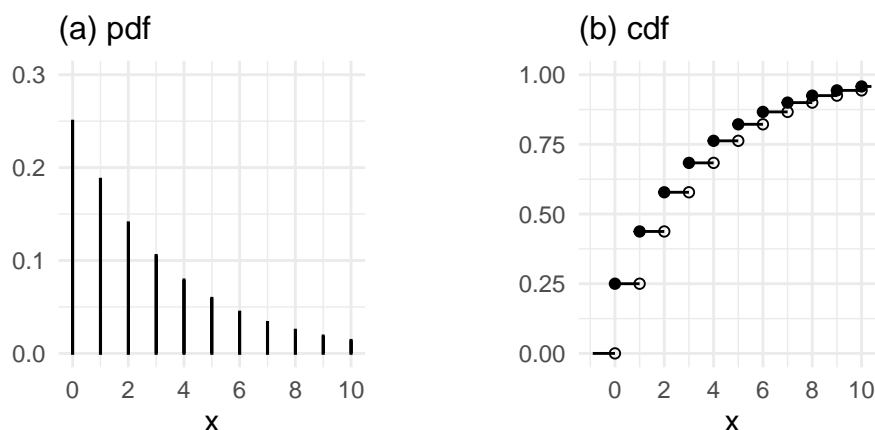


Figure 3.2: Geometric distribution pdf and cdf

The probability density function of a discrete random variable is sometimes called a **probability mass function** and the cdf a **cumulative probability mass function**.

### 3.3.2 Uniform Distribution

**Example 3.4.** A random variable  $X$  has a  $\text{Uniform}(0, 1)$  distribution, written  $X \sim U(0, 1)$ , if its pdf is

$$f_X(x) = 1, \quad x \in [0, 1]. \quad (3.13)$$

Whereas the probability density/mass function of a discrete random variable has the interpretation as  $f_X(x) = \Pr[X = x]$ , this interpretation must be modified for continuous random variables. For continuous variables, the probability of obtaining an outcome between  $a$  and  $b$  is



the area between the pdf and the  $x$ -axis from  $x = a$  to  $x = b$ . That is,

$$\Pr[a \leq X \leq b] = \int_a^b f_X(u) du.$$

For the Uniform (0,1) distribution,  $\Pr[a \leq X \leq b]$  is straightforward to compute. If  $X \sim U(0, 1)$ , then  $\Pr[0.1 < X < 0.3] = 0.2$ ,  $\Pr[X = 0.5] = 0$ ,  $\Pr[0 \leq X \leq 1] = 1$ , and so on. The cdf of the Uniform (0,1) random variable is

$$F_X(x) = \Pr[X \leq x] = \begin{cases} 0, & x < 0 \\ x, & 0 \leq x \leq 1 \\ 1, & x > 1. \end{cases}$$

```
## PDF
p1 <- ggplot() +
  geom_segment(data=data.frame(xstart=0, xend=1, ystart=1, yend=1),
    aes(x=xstart, y=ystart, xend=xend, yend=yend)) + ggtitle('(a) pdf') +
  xlim(-0.5, 1.5) + ylim(0, 1.2) + xlab("x") + ylab(NULL) + plot_theme
# CDF
p2 <- ggplot() +
  geom_segment(data=data.frame(xstart=c(-0.5, 0, 1), xend=c(0, 1, 1.5),
    ystart=c(0, 0, 1), yend=c(0, 1, 1)),
    aes(x=xstart, y=ystart, xend=xend, yend=yend)) + ggtitle('(a) cdf') + xlim(-0.5, 1.5) +
    ylim(0, 1.2) + xlab("x") + ylab(NULL) + plot_theme
p1 | p2
```

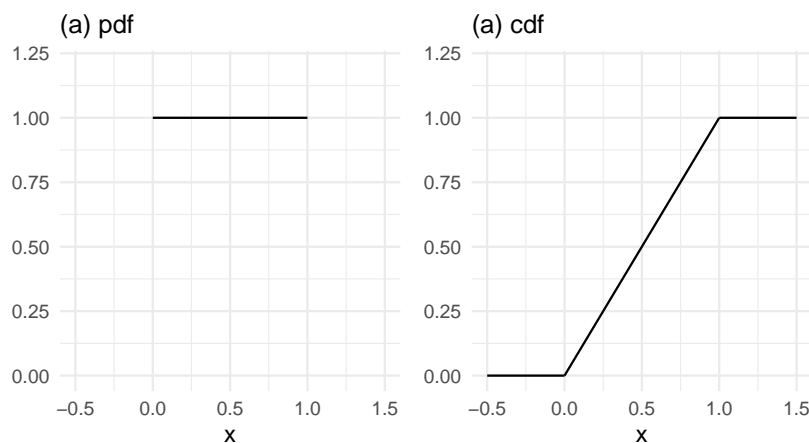


Figure 3.3: Uniform[0,1] distribution pdf and cdf

### 3.3.3 Mean, Variance and Other Moments

We have earlier introduced the mean and variance of discrete random variables with a finite number of outcomes. We extend these definitions to discrete random variables with countably infinite number of outcomes, and continuous random variables.

- If  $X$  has a finite number of possible outcomes  $x_1, x_2, \dots, x_n$  with probabilities  $p_1, p_2, \dots, p_n$  respectively, then

$$E[X] = \sum_{i=1}^n x_i p_i$$

$$\text{var}[X] = \sum_{i=1}^n (x_i - E[X])^2 p_i$$

- If  $X$  has a infinitely countable number of possible outcomes  $x_1, x_2, \dots$  with probabilities  $p_1, p_2, \dots$  respectively, then

$$E[X] = \sum_{i=1}^{\infty} x_i p_i$$

$$\text{var}[X] = \sum_{i=1}^{\infty} (x_i - E[X])^2 p_i$$

- If the possible values of  $X$  range over a continuum, and it has pdf  $f_X(x)$ , then

$$E[X] = \int x f_X(x) dx$$

$$\text{var}[X] = \int (x - E[X])^2 f_X(x) dx \quad (3.14)$$

where the integrals are over the range of possible values of  $X$ . We can also write the variance definition as

$$\text{var}[X] = E[(X - E[X])^2]. \quad (3.15)$$

Sometimes easier to use

$$\text{var}[X] = E[X^2] - E[X]^2. \quad (3.16)$$

which you can derive from Eq. 3.15. The square root of the variance is called the **standard deviation** of  $X$ .

### Example 3.5.

- If  $X \sim \text{Bernoulli}(p)$ , then  $E[X] = 1 \cdot p + 0 \cdot (1 - p) = p$  and  $\text{var}[X] = p(1 - p)$ .
- If  $X \sim \text{Geometric}(p)$ , then

$$E[X] = \sum_{x=0}^{\infty} x p (1 - p)^x = \frac{1 - p}{p} \quad (\text{see exercises}) .$$

Furthermore,  $E[X^2] = \sum_{x=0}^{\infty} x^2 p (1 - p)^x = \frac{(1 - p)^2 + (1 - p)}{p^2}$ , therefore

$$\text{var}[X] = \frac{1 - p}{p^2} \quad (\text{see exercises}) .$$

- If  $X \sim \text{Uniform}(0, 1)$ , then

$$E[X] = \int_0^1 x dx = \frac{x^2}{2} \Big|_0^1 = \frac{1}{2} .$$

Furthermore,  $E[X^2] = \int_0^1 x^2 dx = \frac{x^3}{3} \Big|_0^1 = \frac{1}{3}$ , therefore

$$\text{var}[X] = \frac{1}{3} - \frac{1}{4} = \frac{1}{12}.$$

The important properties of the mean and variance continue to hold, i.e.,

- $E[aX + b] = aE[X] + b$
- $\text{var}[aX + b] = a^2 \text{var}[X]$

where  $a$  and  $b$  are constants.

Besides the mean and variance, we occasionally refer to higher moments. The skewness coefficient of a random variable is defined to be

$$S = \frac{E[(X - E[X])^3]}{\sigma^3}$$

where  $\sigma$  is the standard deviation of  $X$ . It is used as a measure of symmetry. If a distribution is symmetric about the mean, then corresponding negative and positive deviations from mean receive the same weight, and retain their signs when raised to the third power. They therefore cancel out when summed or integrated, resulting in  $S = 0$ . E.g., the skewness coefficient of the normal distribution is zero.

The **kurtosis** of a random variable  $X$  is defined as

$$K = \frac{E[(X - E[X])^4]}{\sigma^4}.$$

When raised to the fourth power, small deviation from means ( $< 1$ ) become very small and do not contribute to  $K$  whereas large deviations from mean contribute substantially. The kurtosis is therefore a measure of how “fat-tailed” a distribution is. It turns out that  $K = 3$  for a normal-distributed random variable. A random variable with  $K > 3$  is said to have **excess kurtosis**, or have a “fat-tailed” distribution.

### 3.3.4 The Normal Distribution

A random variable  $X$  has the **normal distribution**  $\text{Normal}(\mu, \sigma^2)$  if it takes possible values over the entire real line, and its pdf is

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ -\frac{(x - \mu)^2}{2\sigma^2} \right\}, \quad x \in \mathbb{R}. \quad (3.17)$$

The pdf of the normal distribution has the familiar symmetric bell-shape, centered at  $\mu$ , which is centered at  $\mu$ . The variance is  $\sigma^2$ . The normal distribution with mean 0 and variance 1 is called the **standard normal distribution**; it has no parameters, and has a special place in probability theory for reasons you will see in the next chapter.

Fig. 3.4 show 5 normal pdfs. The three centered at zero have mean zero. The thinner of these has variance  $1/2$ , and the flatter, broader one as variance 4. The one in bold is the pdf of the standard normal. On either side are pdfs of normal variates with variance 1 and means -6 (left) and 6 (right).

```

x = seq(-10,10,0.01)
# dnorm(x,mu,sd) gives normal pdf at x with mean mu and standard deviation sd
fx_data = data.frame(x=x, fx_norm1=dnorm(x,0,1),
                     fx_norm2 = dnorm(x,0,sqrt(0.25)),
                     fx_norm3 = dnorm(x,0,sqrt(4)),
                     fx_norm4 = dnorm(x,-6,1),
                     fx_norm5 = dnorm(x,6,1))

p1 <- ggplot(data = fx_data) +
  geom_line(aes(x=x, y=fx_norm1), linewidth=1) +
  geom_line(aes(x=x, y=fx_norm2), linetype='dashed') +
  geom_line(aes(x=x, y=fx_norm3), linetype='dotted') +
  geom_line(aes(x=x, y=fx_norm4), color='blue', linewidth=0.75) +
  geom_line(aes(x=x, y=fx_norm5), color='magenta', linewidth=0.75) +
  annotate('text', -7.9, 0.4, label="Normal(-6,1)", color='blue', size=3) +
  annotate('text', 7.8, 0.4, label="Normal(6,1)", color='magenta', size=3) +
  annotate('text', 2.6, 0.25, label="Normal(0,1)", fontface='bold', size=3) +
  annotate('text', 2.3, 0.6, label="Normal(0,1/2)", size=3) +
  annotate('text', 4.3, 0.1, label="Normal(0,4)", size=3) +
  plot_theme + ylim(0,0.8) + ylab(NULL) + theme(aspect.ratio = 0.5)
p1

```

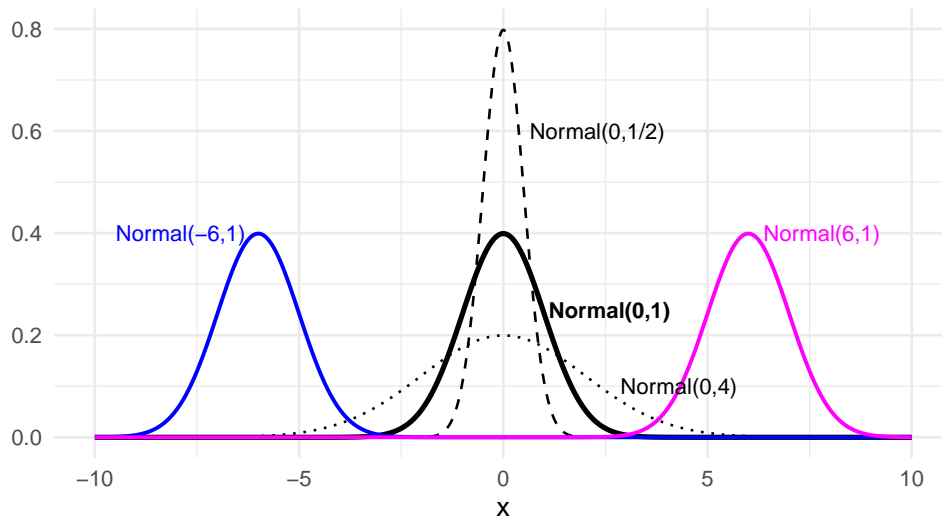


Figure 3.4: Normal pdf, various means and variances

Substituting  $\mu = 0$  and  $\sigma^2 = 1$  into Eq. 3.17 gives the pdf of the standard normal distribution, which is given the special notation  $\phi(\cdot)$ :

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{x^2}{2}\right\}, \quad x \in \mathbb{R}. \quad (3.18)$$

The normal distribution has the property that if  $X \sim \text{Normal}(\mu, \sigma^2)$ , then

$$aX + b \sim \text{Normal}(a\mu + b, a^2\sigma^2)$$

The change in mean and variance will be true for all random variables; the important part here is that the distribution itself doesn't change under the variable transformation. An important application of this result is:

$$\frac{X - \mu}{\sigma} \sim \text{Normal}(0, 1).$$

The cdf of the normal distribution is

$$F_X(x) = \Pr[X \leq x] = \int_{-\infty}^x \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} dx, \quad x \in \mathbb{R}. \quad (3.19)$$

The normal cdf does not have a “closed form” expression (there isn't a simple formula for it) but there are algorithms that can calculate it. The cdf of the standard normal is denoted  $\Phi(x)$ .

```
x = seq(-4,4,0.01)
# pnorm(x,mu,sd) gives normal pdf at x with mean mu and standard deviation sd
fx_data = data.frame(x=x, fx_pnorm1 = pnorm(x,0,1))
p1 <- ggplot(data = fx_data) + geom_line(aes(x=x, y=fx_pnorm1), linewidth=0.75) +
  plot_theme + ylim(0,1.1) + ylab(NULL) + theme(aspect.ratio = 0.5)
p1
```

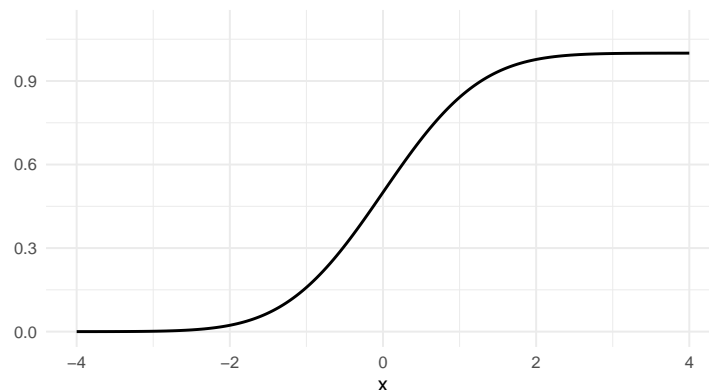


Figure 3.5: Standard normal cdf

It is sometimes useful to express the pdf of a non-standard normal distribution in terms of the pdf and cdf of a standard normal. This is easily done given that linear transformations do not change the distributional form of a normal random variate. If  $X \sim N(\mu, \sigma^2)$ , then its cdf can be written as

$$F_X(x) = \Phi\left(\frac{x - \mu}{\sigma}\right),$$

since

$$F_X(x) = \Pr(X \leq x) = \Pr\left(\frac{X - \mu}{\sigma} \leq \frac{x - \mu}{\sigma}\right) = \Phi\left(\frac{x - \mu}{\sigma}\right).$$

Since the pdf is the derivative of the cdf, we have

$$f_X(x) = \frac{1}{\sigma} \phi\left(\frac{x - \mu}{\sigma}\right).$$

The R function for the cdf of a normal variate with mean  $\mu$  and standard deviation  $\sigma$  is `pnorm()`.

```
cat("pnorm(1,1,2) =", pnorm(1, mean=1, sd=2))
cat("\npnorm(2,1,2) =", pnorm(2, 1, 2))
cat("\npnorm(0,0,1) =", pnorm(0, 0, 1))
cat("\npnorm(-1.96,0,1) =", pnorm(-1.96, 0, 1))
cat("\npnorm(1.96,0,1) =", pnorm(1.96, 0, 1))
```

```
pnorm(1,1,2) = 0.5
pnorm(2,1,2) = 0.6914625
pnorm(0,0,1) = 0.5
pnorm(-1.96,0,1) = 0.0249979
pnorm(1.96,0,1) = 0.9750021
```

To get the value of  $q$  such that  $X \leq q$ , use `qnorm()`:

```
cat("qnorm(0.025,0,1) =", qnorm(0.025, mean=0, sd=1))
```

```
qnorm(0.025,0,1) = -1.959964
```

### 3.3.5 The Log-Normal Distribution

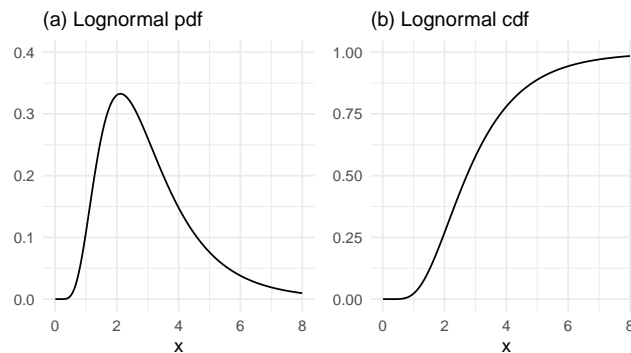
The **log-normal distribution** is sometimes used to describe the probabilistic behavior of stock prices. A random variable  $X$  has the log-normal distribution with distribution  $\mu$  and  $\sigma^2$  if its probability density function is

$$p_X(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left\{-\frac{(\ln x - \mu)^2}{2\sigma^2}\right\}, \quad x \in (0, \infty). \quad (3.20)$$

The log-normal distribution is so-called because if  $X \sim \text{Lognormal}(\mu, \sigma^2)$ , then  $Y = \ln X$  has the **normal distribution**  $\text{Normal}(\mu, \sigma^2)$

The log-normal cdf does not have ‘closed form’ expressions, but as with the normal distribution, that are computer algorithms that can compute the log-normal cdf to a good approximation.

```
x = seq(0.01,8,0.01)
fx_lnorm = dlnorm(x,1,1/2)
Fx_lnorm = plnorm(x,1,1/2)
p1 <- ggplot() +
  geom_line(data = data.frame(x,fx_lnorm), aes(x=x, y=fx_lnorm)) +
  plot_theme + ylim(0,0.4) + ylab(NULL) + ggtitle('(a) Lognormal pdf')
p2 <- ggplot() +
  geom_line(data = data.frame(x,Fx_lnorm), aes(x=x, y=Fx_lnorm)) +
  plot_theme + ylim(0,1) + ylab(NULL) + ggtitle('(b) Lognormal cdf')
(p1 | p2)
```

Figure 3.6: The Log-Normal pdf and cdf with  $\mu=1$ ,  $\sigma=1/2$ 

From the perspective of the validity of possible outcomes, the log-normal distribution, which takes possible values in  $(0, \infty)$ , is more appropriate than, say, the normal distribution, since prices cannot take negative values. However, there are many other distributions with possible outcomes restricted to  $(0, \infty)$ . Whether the log-normal distribution is the appropriate distribution for a given stock price is an empirical question.

- If  $X \sim \text{Lognormal}(\mu, \sigma^2)$ , then

$$E[X] = \exp\left\{\mu + \frac{\sigma^2}{2}\right\}$$

$$\text{var}[X] = [\exp(\sigma^2) - 1] \exp(2\mu + \sigma^2)$$

### 3.3.6 The Chi-squared, Student-t, and F Distributions

We mention three univariate distributions that are related to the normal distribution. We will not need to use the expression of the pdf or cdf of these distributions, but you will need to know the properties that we list here.

#### 3.3.6.1 Chi-square distribution

If  $X \sim N(0, 1)$ , then  $X^2$  has the “**Chi-squared distribution** with one degree of freedom”. If  $X_1, X_2, \dots, X_k$  are independent standard normal variates, then  $\sum_{i=1}^k X_i^2$  is Chi-squared distribution with  $k$  degrees of freedom, denoted  $\chi_{(k)}^2$ . If  $X \sim \chi_{(k)}^2$ , then  $E[X] = k$  and  $\text{var}[X] = 2k$ .

```
x = seq(0.05, 30, 0.01)
df_chi <- data.frame(x = x, fx_chi1 = dchisq(x,1), fx_chi5 = dchisq(x,5),
                    fx_chi10 = dchisq(x,10), fx_chi20 = dchisq(x,20))
p1 <- ggplot(data = df_chi) +
  geom_line(aes(x=x, y=fx_chi1), linewidth=0.5) +
  geom_line(aes(x=x, y=fx_chi5), linewidth=0.5) +
  geom_line(aes(x=x, y=fx_chi10), linewidth=0.5) +
  geom_line(aes(x=x, y=fx_chi20), linewidth=0.5) +
  annotate('text', 2.3, 0.23, label=TeX("$\\chi^2_{(1)}")) +
  annotate('text', 5, 0.17, label=TeX("$\\chi^2_{(5)}")) +
  annotate('text', 10, 0.12, label=TeX("$\\chi^2_{(10)}")) +
  annotate('text', 21, 0.08, label=TeX("$\\chi^2_{(20)}")) +
```

```
plot_theme + ylim(0,0.25) + ylab(NULL) + theme(aspect.ratio = 0.5)
```

p1

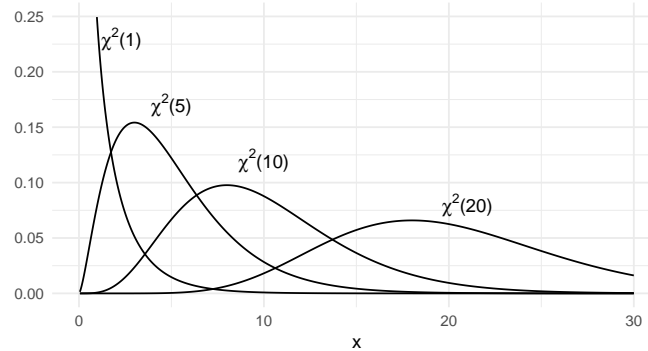


Figure 3.7: The  $\chi^2$  distribution

### 3.3.6.2 Student-t distribution

If  $X$  and  $W$  are independent variables with  $X \sim N(0, 1)$  and  $W \sim \chi^2(k)$ , then

$$\frac{X}{\sqrt{W/k}} \sim t_{(v)}$$

where  $t_{(v)}$  denotes the “**Student-t distribution** with  $v$  degrees of freedom”. A student-t random variate has zero mean, and variance  $\frac{v}{v-2}$  (the variance does not exist unless  $v > 2$ ). The Student-t pdf is similar to that of the standard normal pdf in that it is symmetrically bell-shaped and centered about zero. However, it has fatter tails than a normal distribution (its kurtosis is always greater than 3). This means that a Student-t random variable has greater probability of extreme realizations than a comparable normal variate. The Student-t pdf has the property that it converges to the standard normal pdf as  $v \rightarrow \infty$ . Fig. 3.8 shows the student-t pdf with degree-of-freedom parameter  $v=1, 5, 10$ , and  $20$ , and also the pdf of the standard normal. The  $t(1)$  and  $t(5)$  distributions are indicated, with the  $t(10)$  and  $t(20)$  distributions “between” the  $t(5)$  and the  $\text{Normal}(0,1)$  pdf.

```
x = seq(-5, 5, 0.01)
df_t <- data.frame(x = x, fx_t1 = dt(x,1), fx_t5 = dt(x,5), fx_t10 = dt(x,10),
                  fx_t20 = dt(x,20), fx_norm = dnorm(x,0,1))
p1 <- ggplot(data = df_t) +
  geom_line(aes(x=x, y=fx_t1), linewidth=0.5) +
  geom_line(aes(x=x, y=fx_t5), linewidth=0.5) +
  geom_line(aes(x=x, y=fx_t10), linewidth=0.5) +
  geom_line(aes(x=x, y=fx_t20), linewidth=0.5) +
  geom_line(aes(x=x, y=fx_norm), linewidth=0.6, color="magenta") +
  annotate('text', 0.8, 0.13, label="t(1)") +
  annotate('text', 0.08, 0.34, label="t(5)") +
  annotate('text', 0.89, 0.38, label="N(0,1)", color="magenta") +
  plot_theme + ylim(0, 0.42) + ylab(NULL) +
  theme(aspect.ratio = 0.5)
```

p1



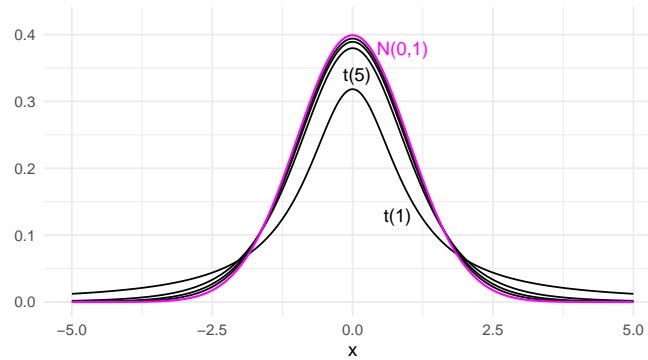


Figure 3.8: The student t distribution

The following table compares the tail probabilities of the normal and the t distribution.

```
norm_vs_t_tail <- cbind(
  pnorm(c(-2.57, -1.96, -1.64)),
  pt(c(-2.57, -1.96, -1.64), 1),
  pt(c(-2.57, -1.96, -1.64), 5),
  pt(c(-2.57, -1.96, -1.64), 10),
  pt(c(-2.57, -1.96, -1.64), 20),
  pt(c(-2.57, -1.96, -1.64), 30))
colnames(norm_vs_t_tail) <- c("N(0,1)", "t(1)", "t(5)", "t(10)", "t(20)", "t(30)")
rownames(norm_vs_t_tail) <- c("P[X<-2.57]", "P[X<-1.96]", "P[X<-1.64]")
round(norm_vs_t_tail, 4)
```

	N(0,1)	t(1)	t(5)	t(10)	t(20)	t(30)
P[X<-2.57]	0.0051	0.1181	0.0250	0.0139	0.0091	0.0077
P[X<-1.96]	0.0250	0.1502	0.0536	0.0392	0.0320	0.0297
P[X<-1.64]	0.0505	0.1743	0.0810	0.0660	0.0583	0.0557

### 3.3.6.3 F distribution

If  $W_1$  and  $W_2$  are independent chi-squared random variables with degrees of freedom  $v_1$  and  $v_2$  respectively, then

$$\frac{W_1/v_1}{W_2/v_2} \sim F_{(v_1, v_2)}$$

where  $F_{(v_1, v_2)}$  denotes the “**F distribution** with  $v_1$  and  $v_2$  degrees of freedom”. If  $X \sim F_{(v_1, v_2)}$ , then

$$E[X] = \frac{v_2}{v_2 - 2}$$

$$\text{var}[X] = 2 \left( \frac{v_2}{v_2 - 2} \right)^2 \frac{v_1 + v_2 - 2}{v_1(v_2 - 4)}.$$

The F-distribution is also related to the t- and chi-squared distributions in that

- If  $X \sim t_{(v)}$ , then  $X^2 \sim F_{(1, v)}$ ,
- If  $X \sim F_{(v_1, v_2)}$ , then the pdf of  $v_1 X$  converges to the  $\chi^2(v_1)$  pdf as  $v_2 \rightarrow \infty$ .

Fig. 3.9 shows the  $F_{(3, 10)}$  pdf.

```

x <- seq(0,5,0.01)
Fpdf <- df(x, df1=3, df2=20)
dat2 <- data.frame(x , Fpdf)
p1 <- ggplot(data = dat2) +
  geom_line(aes(x=x, y = Fpdf), colour="black") +
  ggtitle(TeX("$F_{(3,20)}$ pdf")) + ylab(NULL) + theme_minimal() +
  theme(aspect.ratio = 0.5)

p1

```

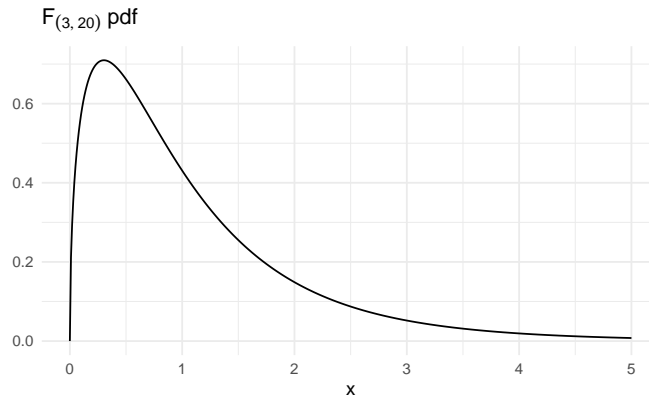


Figure 3.9: The F pdf

### 3.3.7 The Bivariate Normal Distribution

Two random variables  $X$  and  $Y$  have the bivariate normal distribution if their joint pdf have the form

$$f_{X,Y}(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho_{xy}^2}} \exp\left\{-\frac{1}{2}\frac{\tilde{x}^2 - 2\rho_{xy}\tilde{x}\tilde{y} + \tilde{y}^2}{1-\rho_{xy}^2}\right\} \quad (3.21)$$

where  $\tilde{x} = \frac{x - \mu_x}{\sigma_x}$  and  $\tilde{y} = \frac{y - \mu_y}{\sigma_y}$ . The bivariate normal distribution has five parameters, with the following interpretation:

- $\mu_x$  : unconditional mean of  $X$ ,
- $\mu_y$  : unconditional mean of  $Y$ ,
- $\sigma_x^2$  : unconditional variance of  $X$ ,
- $\sigma_y^2$  : unconditional variance of  $Y$ , and
- $\rho_{xy}$  : correlation coefficient of  $X$  and  $Y$ , i.e.,  $\rho = \frac{\sigma_{xy}}{\sigma_x\sigma_y}$  where  $\sigma_{xy} = \text{cov}[X, Y]$ .

We write  $(X, Y) \sim \text{Normal}_2(\mu_x, \mu_y, \sigma_x^2, \sigma_y^2, \rho_{xy})$

Contour plots are helpful for visualizing bivariate normal distributions. We show the contour plots of a bivariate normal distribution with

$$(\mu_x, \mu_y, \sigma_x^2, \sigma_y^2, \rho) = (1, 0, 1, 2, 0.9).$$

in Fig. 3.10(a). Alternatively, one can look at the 3-d plot of the bivariate pdf in Fig. 3.10(b).

```

x <- seq(-3,5,0.2)
y <- seq(-4,4,0.2)
mesh_xy <- expand.grid(x,y)
binorm1 <- function(mesh_xy, mu1, mu2, sg1, sg2, rho){
  r <- (2*pi*sg1*sg2*sqrt(1-rho^2))^( -1)*exp(-1/(2*(1-rho^2))*
    (((mesh_xy[,1]-mu1)/sg1)^2 - 2*rho*(mesh_xy[,1]-mu1)*(mesh_xy[,2]-mu2)/sg1/sg2
    + ((mesh_xy[,2]-mu2)/sg2)^2))
}
binorm2 <- function(x,y, mu1, mu2, sg1, sg2, rho){
  r <- (2*pi*sg1*sg2*sqrt(1-rho^2))^( -1)*exp(-1/(2*(1-rho^2))*
    (((x-mu1)/sg1)^2 - 2*rho*(x-mu1)*(y-mu2)/sg1/sg2 + ((y-mu2)/sg2)^2))
}
par(mfrow=c(1,2))
mu1 <- 1; mu2 <- 0; sg1 <- 1; sg2 <- sqrt(2); rho = 0.7
par(mar=c(2,1.5,1.5,1.5)) # Contour
f_xy <- binorm1(mesh_xy, mu1, mu2, sg1, sg2, rho)
f_xy <- matrix(f_xy, byrow=FALSE, nrow=length(x))
p1 <- contour(x,y,f_xy,nlevels=12, xlab="", ylab="", main="(a) Contour Plot",
  cex.lab=0.6, cex.axis=0.6, cex.main=0.6)
title(ylab="y", line=-1, cex.lab=0.6)
title(xlab="x", line=-1, cex.lab=0.6)
par(mar=c(0,1,0,0)) # 3-d
z <- outer(x,y,binorm2, mu1, mu2, sg1, sg2, rho)
p2 <- persp(x,y,z,theta=20, phi=30, r=10, expand=0.8, col="lightblue",
  ltheta=0, shade=0.75, ticktype="detailed", xlab="x", ylab="y", zlab="",
  cex.lab=0.6, cex.axis=0.6)
title(main="(b) 3-d Plot", cex.main=0.6, line=-1)

```

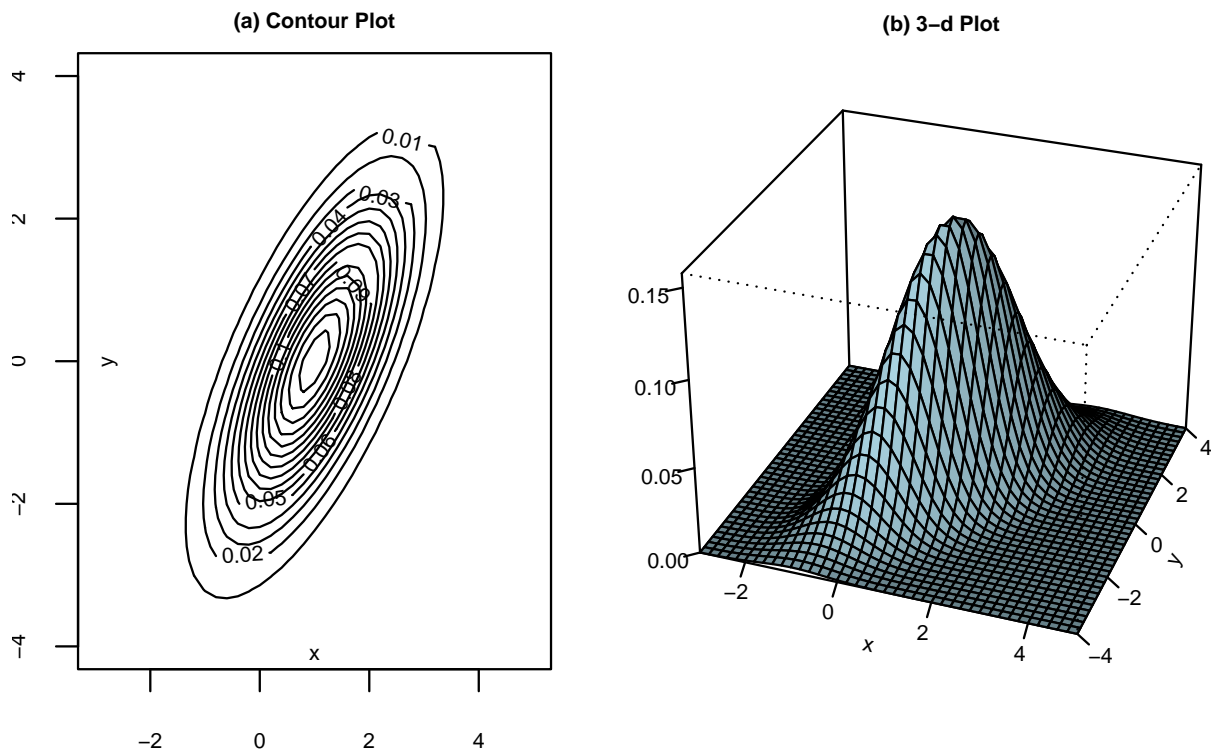


Figure 3.10: Bivariate Normal Distribution (parameter values given in text)

The marginal and conditional distributions of a bivariate normal random variables are also normal. To see this, we “complete the square” on  $\tilde{x}^2 - 2\rho_{xy}\tilde{x}\tilde{y} + \tilde{y}^2$  to get

$$\begin{aligned}\tilde{x}^2 - 2\rho_{xy}\tilde{x}\tilde{y} + \tilde{y}^2 &= (\tilde{x} - \rho_{xy}\tilde{y})^2 + (1 - \rho_{xy}^2)\tilde{y}^2 \\ &= \left[ \frac{x - \mu_x}{\sigma_x} - \frac{\sigma_{xy}}{\sigma_x\sigma_y} \frac{y - \mu_y}{\sigma_y} \right]^2 + (1 - \rho_{xy}^2) \left( \frac{y - \mu_y}{\sigma_y} \right)^2 \\ &= \frac{1}{\sigma_x^2} \left[ x - \mu_x - \frac{\sigma_{xy}}{\sigma_y^2} (y - \mu_y) \right]^2 + (1 - \rho_{xy}^2) \left( \frac{y - \mu_y}{\sigma_y} \right)^2 \\ &= \frac{1}{\sigma_x^2} [x - (\alpha + \beta y)]^2 + (1 - \rho_{xy}^2) \left( \frac{y - \mu_y}{\sigma_y} \right)^2\end{aligned}$$

where  $\alpha = \mu_x - \beta\mu_y$  and  $\beta = \frac{\sigma_{xy}}{\sigma_y^2}$ . Then the pdf can be written as

$$\begin{aligned}f_{X,Y}(x,y) &= \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho_{xy}^2}} \exp \left\{ -\frac{1}{2} \frac{1}{1-\rho_{xy}^2} (\tilde{x}^2 - 2\rho_{xy}\tilde{x}\tilde{y} + \tilde{y}^2) \right\} \\ &= \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho_{xy}^2}} \times \exp \left\{ -\frac{1}{2} \frac{1}{1-\rho_{xy}^2} \left[ \frac{1}{\sigma_x^2} [x - (\alpha + \beta y)]^2 + (1 - \rho_{xy}^2) \left( \frac{y - \mu_y}{\sigma_y} \right)^2 \right] \right\} \\ &= \underbrace{\frac{1}{\sqrt{2\pi}\sqrt{\sigma_x^2(1-\rho_{xy}^2)}} \exp \left\{ -\frac{1}{2} \frac{[x - (\alpha + \beta y)]^2}{\sigma_x^2(1-\rho_{xy}^2)} \right\}}_A \times \underbrace{\frac{1}{\sqrt{2\pi}\sigma_y} \exp \left\{ -\frac{1}{2} \left( \frac{y - \mu_y}{\sigma_y} \right)^2 \right\}}_B\end{aligned}$$

If we compare expressions  $A$  and  $B$  with the expression for the Normal pdf, we see that  $B$  is a normal pdf  $f_Y(y)$  with mean  $\mu_y$  and variance  $\sigma_y$ , and if we take  $y$  as fixed, then  $A$  is a (conditional) normal pdf  $f_{X|Y}(x|y)$  with mean  $\alpha + \beta y$  and variance  $\sigma_x^2 - \sigma_{xy}/\sigma_y^2$ . That is, if  $X$  and  $Y$  have the bivariate normal distribution Eq. 3.21, then

- the marginal distribution of  $Y$  is  $\text{Normal}(\mu_y, \sigma_y^2)$ ,
- the conditional distribution of  $X$  given  $Y$  is  $\text{Normal}(\mu_{x|y}, \sigma_{x|y})$  where

$$\begin{aligned}\mu_{x|y} &= \mu_x + \frac{\sigma_{xy}}{\sigma_y^2}(y - \mu_y), \\ \sigma_{x|y} &= \sigma_x^2 - \frac{\sigma_{xy}^2}{\sigma_y^2}.\end{aligned}$$

The conditional mean can be written as  $\mu_{x|y} = \alpha_x + \beta_x y$  where  $\alpha_x = \mu_x - \beta_x \mu_y$  and  $\beta_x = \frac{\sigma_{xy}}{\sigma_y^2}$ .

Similarly,

- the marginal distribution of  $X$  is  $\text{Normal}(\mu_x, \sigma_x^2)$ ,
- the conditional distribution of  $Y$  given  $X$  is  $\text{Normal}(\mu_{y|x}, \sigma_{y|x})$  where

$$\begin{aligned}\mu_{y|x} &= \mu_y + \frac{\sigma_{xy}}{\sigma_x^2}(x - \mu_x), \\ \sigma_{y|x} &= \sigma_y^2 - \frac{\sigma_{xy}^2}{\sigma_x^2}.\end{aligned}$$

The conditional mean can be written as  $\mu_{y|x} = \alpha_y + \beta_y x$  where  $\alpha_y = \mu_y - \beta_y \mu_x$  and  $\beta_y = \frac{\sigma_{xy}}{\sigma_x^2}$ .

It follows immediately from the decomposition of the bivariate normal pdf that  $X$  and  $Y$  are independent if they are bivariate normal and uncorrelated (see exercises). It can also be shown that if  $X$  and  $Y$  are bivariate normal, then

- $aX + bY \sim \text{Normal}(a\mu_x + b\mu_y, a^2\sigma_x^2 + b^2\sigma_y^2 + 2ab\sigma_{xy})$ .

We omit the proof of this last result.

### 3.3.8 Exercises

**Exercise 3.11.** Use the fact that

$$\sum_{j=0}^{\infty} jr^j = \frac{r}{(1-r)^2} \quad \text{and} \quad \sum_{j=0}^{\infty} j^2 r^j = \frac{r^2 + r}{(1-r)^3}$$

to derive the mean and variance of the Geometric distribution.

**Exercise 3.12.** Show that  $X$  and  $Y$  are independent if they are bivariate normal and uncorrelated. *Hint: show that  $f_{X,Y}(x,y) = f_Y(y)f_X(x)$  when  $\rho_{xy} = 0$ .*

**Exercise 3.13.** The function `pnorm()`, when evaluated at  $x$ , returns the CDF of the normal pdf at  $x$ , i.e., it returns the probability  $\Pr[X \leq x]$ . The quantile function `qnorm()`, when evaluated at probability  $p$ , returns the value of  $x$  for which  $\Pr[X \leq x] = p$ . For example:

```
pnorm(0, mean=0, sd=1)    # Pr[X <= 0] for X~N(0,1)
```

```
[1] 0.5
```

```
qnorm(0.5, mean=0, sd=1)  # c such that Pr[X<=c]=0.5 when X~N(0,1)
```

```
[1] 0
```

The corresponding functions for the t, chi-sq, and F distributions are `pt(x,df)` and `qt(p,df)`, `pchisq(x,df)` and `qchisq(p,df)`, and `pf(x,df1,df2)` and `qf(p,df1,df2)` respectively. Find:

- i.  $\Pr[X \leq -2.5]$  when  $X \sim N(0,1)$
- ii.  $\Pr[X \leq -2.5]$  when  $X \sim t_{(5)}$
- iii.  $c$  such that  $\Pr[X > c] = 0.05$  when  $X \sim \chi_{(5)}^2$ .
- iv.  $\Pr[-1.96 \leq X \leq 1.96]$  when  $X \sim N(0,1)$
- v.  $c$  such that  $\Pr[-c \leq X \leq c] = 0.95$  when  $X \sim N(0,1)$
- vi.  $c$  such that  $\Pr[-c \leq X \leq c] = 0.95$  when  $X \sim t_{(12)}$
- vii.  $c$  such that  $\Pr[-c \leq X \leq c] = 0.95$  when  $X \sim t_{(100)}$
- viii.  $c$  such that  $\Pr[X > c] = 0.05$  when  $X \sim F_{(5,8)}$ .
- ix.  $5c$  such that  $\Pr[X > c] = 0.05$  when  $X \sim F_{(5,80)}$ .
- x.  $5c$  such that  $\Pr[X > c] = 0.05$  when  $X \sim F_{(5,8000)}$ .

### 3.4 Prediction

The prediction problem is, roughly: given the realization of  $X$ , what is  $Y$ ? In other words, if I tell you that the realization of  $X$  is  $x$  (this is your “information set”), what can you tell me about  $Y$ ? Often what is wanted is some estimate of what  $Y$  will turn out to be, e.g., what is the price of a house in a particular city given that it is in such-and-such location, is 15 years old, and has four rooms? The desired answer is some value, such as, 1.8 million dollars. This is a “point prediction”. Sometimes we want other types of prediction, such as “what is the probability that inflation will exceed 5 percent over the next quarter?” This is a “probabilistic forecast”, and the information set would presumably be current and past realizations of all variables that the forecaster deems relevant for inflation.

Suppose what we want is a point prediction of  $Y$  given knowledge that  $X = x$ , and suppose we have a squared error loss function, meaning that if our prediction is  $\hat{y}(x)$  and the realization of  $Y$  turns out to be  $y$ , then the cost of prediction error is  $(y - \hat{y}(x))^2$ . Suppose the prediction is made to minimize *expected* loss, i.e., we choose  $\hat{y}(x)$  to minimize the (Conditional) Mean Squared Prediction Error (MSPE)

$$MSPE(Y|X = x) = E[(Y - \hat{y}(x))^2|X = x]. \quad (3.22)$$

The question is how to choose  $\hat{y}(x)$ .

Unsurprisingly, the optimal prediction (the prediction that minimizes Eq. 3.22) is the conditional mean  $E[Y|X = x]$ , as we now show. Write the MSPE as

$$\begin{aligned} MSPE(Y|X = x) &= E[(Y - E[Y|X = x] + E[Y|X = x] - \hat{y}(x))^2|X = x] \\ &= E[(Y - E[Y|X = x])^2|X = x] + E[(E[Y|X = x] - \hat{y}(x))^2|X = x] + \\ &\quad 2E[(Y - E[Y|X = x])(E[Y|X = x] - \hat{y}(x))|X = x] \\ &= E[(Y - E[Y|X = x])^2|X = x] + (E[Y|X = x] - \hat{y}(x))^2. \end{aligned}$$

The last equality holds because the second term in the RHS of the second line is fixed given  $X = x$ , and the third term is zero.<sup>2</sup> The first term on the RHS of the last line does not depend on the prediction, and since the second term is non-negative, the prediction that minimizes  $MSPE(Y|X = x)$  is the conditional expectation  $E[Y|X = x]$ . Since  $E[Y|X]$  minimizes conditional  $MSPE$  for all possible values  $x$  of  $X$ , it also minimizes the unconditional MSPE.

Of course in practice we do not know the conditional distribution  $E[Y|X = x]$ , and will have to estimate it from previously taken observations of  $Y$  and  $X$ .

---

<sup>2</sup>We can obtain the same expression by noting that the MSPE is the expectation of a square of a random variable, which is equal to the variance of the random variable plus the square of its mean.

## Chapter 4

### Statistics Review

Many problems in statistics involve estimating the population mean of a variable and testing hypotheses regarding its value. Often the sample average of the observations are used as an estimate of the population mean. We do so because we know *a priori* that in many circumstances the sample mean is a good estimator (i.e., a good estimation rule) for the population mean.

In this chapter, we review the ideas behind parameter estimation and hypothesis testing using the example of estimating and testing a population mean, including an extended discussion of what “good” means in the context of parameter estimation. We then go through in detail a simple application to illustrate the theory. The R code in this chapter uses the `tidyverse` and `patchwork` packages.

```
library(tidyverse)
library(patchwork)
```

#### 4.1 Estimation

Suppose  $Y$  is a random variable with mean  $\mu$  and variance  $\sigma^2$ . You are interested in

- estimating  $\mu$ ,
- getting some idea how “good” (accurate, precise, etc.) your estimate is,
- testing if  $\mu$  is equal to some hypothesized value  $\mu_0$ , i.e., checking to see if you have statistical evidence to reject  $\mu = \mu_0$ .

Suppose you will be able to obtain a random sample of  $N$  observations  $Y_1, Y_2, \dots, Y_N$  of this variable, but *suppose you haven’t yet obtained this sample*, so for the moment,  $Y_1, Y_2, \dots, Y_N$  are all random variables. We will nonetheless continue to refer to them as the sample. The term “random sample” means that your sample are i.i.d. random variables of  $X$ , so they are all uncorrelated with each other, and  $E[Y_i] = \mu$  and  $\text{var}[Y_i] = \sigma^2$  for all  $i$ .

##### 4.1.1 Unbiased Estimators

Since you are estimating a population mean, suppose you propose to use the sample mean

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^N Y_i = \bar{Y} \quad (4.1)$$

as an estimator for  $\mu$ .

We will switch between the notation  $\hat{\mu}$  and  $\bar{Y}$ , using the former especially when we want to highlight the fact that we are estimating the population mean. Take note to distinguish between the population mean and the sample mean. The population mean is the constant  $E[Y] = \mu$ . The sample mean is the expression in Eq. 4.1, which because it is made of random variables, is itself a random variable. You are proposing to use the sample mean as an *estimator* for the population mean. When you do finally collect your sample realizations, you will plug in those realizations into Eq. 4.1 to calculate your **estimate** of the population mean. An estimate (an actual number) is a realization of the estimator (a random variable).

Is your proposal to use the sample mean as an estimator for the population mean a good one? First, what is your criteria for deciding what makes a good estimator?

One criterion used is unbiasedness. The estimator  $\hat{\theta}$  is said to be **unbiased** for  $\theta$  if

$$E[\hat{\theta}] = \theta.$$

That is, the distribution of your estimator is centered about the population mean. You can interpret this to mean that by following the estimation rule  $\hat{\theta}$  you will not be systematically over- or under-estimating your target parameter.

**Theorem 4.1.** *Suppose you have an i.i.d. sample  $Y_1, Y_2, \dots, Y_N$  of a random variable  $Y$  with mean  $E[Y] = \mu$  and variance  $\text{var}[Y] = \sigma^2$ . Then*

$$E[\bar{Y}] = E\left[\frac{1}{N} \sum_{i=1}^N Y_i\right] = \frac{1}{N} \sum_{i=1}^N E[Y_i] = \frac{1}{N} N\mu = \mu \quad (4.2)$$

and

$$\text{var}[\bar{Y}] = \text{var}\left[\frac{1}{N} \sum_{i=1}^N Y_i\right] = \frac{1}{N^2} \sum_{i=1}^N \text{var}[Y_i] = \frac{1}{N^2} N\sigma^2 = \frac{\sigma^2}{N}. \quad (4.3)$$

Eq. 4.2 says that the sample mean is an unbiased estimator for the population mean. Theorem 4.1 also provides an expression for the variance of the sample mean. For the variance, we made use of the “independence” assumption, so the sample is an uncorrelated sample, and the variance of the sum became the sum of the variance. Note that for the unbiasedness part, we did not require uncorrelated samples.

**Example 4.1** (An example of a biased estimator). We should always provide an estimate of the variance (or standard error) of our estimator.<sup>1</sup> For the sample mean, this requires estimating  $\sigma^2 = \text{var}[Y]$ , and using this estimate in Eq. 4.3. Since  $\text{var}[Y] = E[(Y - E[Y])^2] = E[Y^2] - E[Y]^2$ , it seems reasonable to consider estimating  $\sigma^2$  with

$$\widetilde{\sigma^2} = \frac{1}{N} \sum_{i=1}^N (Y_i - \bar{Y})^2 = \frac{1}{N} \sum_{i=1}^N Y_i^2 - \bar{Y}^2. \quad (4.4)$$

However, this is a biased estimator for  $\sigma^2$ . Using  $E[Y_i^2] = \text{var}[Y_i] + E[Y_i]^2 = \sigma^2 + \mu^2$  and  $E[\bar{Y}^2] = \text{var}[\bar{Y}] + E[\bar{Y}]^2 = \sigma^2/N + \mu^2$ , we have

$$E[\widetilde{\sigma^2}] = \frac{1}{N} \sum_{i=1}^N E[Y_i^2] - E[\bar{Y}^2] = \sigma^2 + \mu^2 - \frac{\sigma^2}{N} - \mu^2 = \frac{N-1}{N} \sigma^2.$$

The estimator Eq. 4.4 therefore systematically under-estimates the variance of the sample observations. If your sample size  $N$  is large, the bias may be negligible for all intents and purposes. Nonetheless, in this case it is easy to derive an unbiased estimator, namely

$$\widehat{\sigma^2} = \frac{N}{N-1} \widetilde{\sigma^2} = \frac{1}{N-1} \sum_{i=1}^N (Y_i - \bar{Y})^2. \quad (4.5)$$

---

<sup>1</sup>The square root of the estimator variance is the *standard error* of the estimator, and can be thought of as a measure of the size of the estimation error one might expect to make.



The intuition for why the divisor in Eq. 4.5 has to be  $N - 1$  instead of  $N$  is that the deviations from sample mean always sum to zero. This means that there are only  $N - 1$  ‘free’ deviations from sample mean. For example, given  $\sum_{i=1}^N (Y_i - \bar{Y})$  and the first  $N - 1$  deviations  $(Y_i - \bar{Y})$ ,  $i = 1, 2, \dots, N - 1$ , you can determine the  $N$ th deviation as  $(Y_N - \bar{Y}) = -\sum_{i=1}^{N-1} (Y_i - \bar{Y})$ . One “degree of freedom” was lost because we had to use the observations to compute the sample mean in order to compute the deviations from sample mean.

#### 4.1.2 Efficiency

Unbiasedness is obviously a desirable property for an estimator, but on its own, it is hardly sufficient justification. After all, the estimator  $\hat{\mu}_1 = Y_1$ , where you only pick the first observation and throw away the rest, is also unbiased:  $E[\hat{\mu}_1] = E[Y_1] = \mu$ . Of course we don’t want to use just one observation; the whole point of taking the average of several observations is so that positive and negative errors cancel out, leading to a better estimator.

The idea of positive and negative errors cancelling out is reflected in the variance of an estimator, which is a measure of how *precise* the estimator is. Since  $\text{var}[\bar{Y}] = \frac{\sigma^2}{N}$ , using all  $N$  observations produces an estimator that is much more precise than using just a single observation, which gives a variance of  $\sigma^2$ . Even using just two observations reduces the estimator variance by half.

Obviously we want our sample size to be as large as possible. But we can in fact go further and claim that the sample mean, under the conditions we have stated, is the most precise among all “linear unbiased estimators”. In the context of statistical estimation, a linear estimator is one that takes the form  $\sum_{i=1}^N a_i Y_i$ , and such an estimator will be unbiased if the  $a_i$ ’s sum to one. The sample mean is a linear unbiased estimator<sup>2</sup> with  $a_i = 1/N$  for  $i = 1, 2, \dots, N$ . It is the linear estimator with the smallest variance; we cannot tweak the weights to give us a more precise estimator.

The following argument proves that the sample mean has the smallest variance among all linear unbiased estimators. Suppose we have a linear unbiased estimator  $\sum_{i=1}^N a_i Y_i$  where  $a_i \neq 1/N$  for at least one  $i$ . Write  $a_i = 1/N + b_i$  where the  $b_i$ ’s are not all zero, but sum to zero. This means that the estimator differs from the sample mean, but is nonetheless unbiased, since  $\sum_{i=1}^N a_i = 1$ . Then

$$\begin{aligned} \text{var}[\tilde{Y}] &= \text{var} \left[ \sum_{i=1}^N \left( \frac{1}{N} + b_i \right) Y_i \right] \\ &= \sum_{i=1}^N \left( \frac{1}{N} + b_i \right)^2 \text{var}[Y_i] \\ &= \text{var}[\bar{Y}] + \sigma^2 \sum_{i=1}^N b_i^2. \end{aligned} \tag{4.6}$$

Since the  $b_i$ ’s are not all zero,  $\sum_{i=1}^N b_i^2 > 0$ , so  $\text{var}[\tilde{Y}] > \text{var}[\bar{Y}]$ . We say that the sample mean is the most **efficient** among all linear unbiased estimators (we also say it is the ‘best linear unbiased’ estimator).

---

<sup>2</sup>An example of a non-linear estimator is the geometric mean  $(Y_1 \times Y_2 \times \dots \times Y_N)^{1/N}$  which, incidentally, is biased downwards, and can only be used for positive random variables.

### 4.1.3 Mean Square Error

What we have done so far is taken a “lexicographical” approach to the criterion of unbiasedness and efficiency. We have justified the sample mean as being unbiased, and then argued that among all *linear* unbiased estimators, it is the most precise. Is there a criteria that combines both on an equal footing?

The *mean square error* (MSE) of an estimator  $\hat{\theta}$  is defined as

$$MSE[\hat{\theta}] = E[(\hat{\theta} - \theta)^2].$$

The MSE can be decomposed into the estimator variance and square of the bias:

$$\begin{aligned} E[(\hat{\theta} - \theta)^2] &= \text{var}[\hat{\theta} - \theta] + E[(\hat{\theta} - \theta)]^2 \\ &= \text{var}[\hat{\theta}] + (E[\hat{\theta}] - \theta)^2 \\ &= \text{var}[\hat{\theta}] + \text{bias}^2 \end{aligned}$$

where  $\text{bias} = E[\hat{\theta}] - \theta$ .

We might want an estimator that minimizes the MSE. Earlier, we found an estimator whose bias is zero, and then conditional on unbiasedness, minimum variance. Minimizing MSE considers both variance and bias simultaneously. In particular, we may find an estimator that is slightly biased, but has a much smaller variance, thereby reducing MSE. In other words, there may have a beneficial bias-variance trade-off.

It is easy to show that the sample mean minimizes the *sample* mean square error: suppose we choose  $\hat{\mu}$  to minimize

$$\text{Sample MSE} = \sum_{i=1}^N (Y_i - \hat{\mu})^2.$$

The first order condition is  $-2 \sum_{i=1}^N (Y_i - \hat{\mu}) = 0$  which you can solve for  $\hat{\mu} = \bar{Y}$ . The second derivative of the sample MSE with respect to  $\hat{\mu}$  is  $2 > 0$  so the sample mean solves the minimization problem. However, it does not necessarily minimize the *population* MSE  $E[(\hat{\mu} - \mu)^2]$ , and we may be able to find another estimator that produces a lower MSE.

We will see examples of this in the exercises.

## 4.2 A Coin Toss Example

How would we test if a coin is fair – that a random toss of the coin is as likely to show heads as it would tails? We could toss the coin a number of times and see if the frequency of heads is ‘reasonably’ close to half. How far from half should the frequency be for us to claim that the coin is not fair?

We can model the “experiment” of tossing the coin by describing the outcome as a random variable  $Y$  with two possible values 0 and 1, where 0 indicates tails and 1 indicates heads, and where the probability of  $Y = 1$  is  $p$  and the probability of  $Y = 0$  is  $1 - p$ . Such a random variable is said to have a *Bernoulli* distribution,  $Y \sim \text{Bern}(p)$ . The hypothesis that the coin is fair is  $p = 0.5$ .

Suppose that each observation in our (as yet hypothetical) sample  $\{Y_1, Y_2, \dots, Y_N\}$  is such that no one toss affects the outcome of any other, and that each of the  $Y_i$  is Bernoulli with the

same  $p$ . That is, we assume that the  $Y_i$ 's are independently and identically distributed with distribution  $\text{Bern}(p)$ :

$$Y_i \stackrel{iid}{\sim} \text{Bern}(p), i = 1, 2, \dots, N.$$

When might a sample of coin tosses not be i.i.d.? Perhaps one lazily flips a coin back and forth, so we simply alternate between heads and tails (the observations are not uncorrelated). Or perhaps one (somehow) damages the coin during the flipping process, so that  $p$  changes (observations are no longer identically distributed). Some individuals have been known to be skillful enough to willfully control the outcome of a coin toss. We rule out all such cases.

This application nicely fits our estimation theory of the previous section. We have an i.i.d. sample of observations of a random variable  $Y$  with mean  $p$ . The only difference here is that there is not a separate parameter for the variance, which is  $p(1-p)$ . This does not cause any problems for us; we simply replace all instances of  $\sigma^2$  with  $p(1-p)$ . Another difference is that we have more information in this example than in the previous section. In particular, we know the whole distribution of  $Y$ , whereas in the previous section we did not.

Since  $p$  is the population mean of  $Y$ , we estimate it using the sample mean

$$\hat{p} = \bar{Y}.$$

In this application, the sample mean is just the frequency of heads in the sample. From the theory discussed earlier we know that the sample mean is unbiased, and minimum variance among all linear unbiased estimators.

In fact, for this particular example the sample mean has the lowest variance among all unbiased estimators, linear or not. This is because the variance of the sample mean achieves a theoretical lower bound for unbiased estimators known as the Cramer-Rao Lower Bound. We omit a discussion of this for now.

The variance of  $\hat{p} = \bar{Y}$  is  $\text{var}[Y]/N$ . Since  $\text{var}[Y] = p(1-p)$ , and  $\hat{p}$  is an unbiased estimator for  $p$ , perhaps we can estimate  $\text{var}[Y]$  using

$$\widehat{\text{var}[Y]} = \hat{p}(1 - \hat{p}) = \hat{p} - \hat{p}^2?$$

This is a biased estimator. Although  $\hat{p}$  is unbiased,  $E[\hat{p}^2] = \text{var}[\hat{p}] + E[\hat{p}]^2 = \text{var}[\hat{p}] + p^2 > p^2$ . Therefore  $E[\hat{p} - \hat{p}^2] < p - p^2$ . In fact, it is easy to show that

$$\hat{p} - \hat{p}^2 = \bar{Y} - \bar{Y}^2 = \frac{1}{N} \sum_{i=1}^N (Y_i - \bar{Y})^2 \quad (4.7)$$

which we already know to be a downward biased estimator of  $\text{var}[Y]$ .

Since we also know that  $\frac{1}{N-1} \sum_{i=1}^N (Y_i - \bar{Y})^2$  is an unbiased estimator for  $\text{var}[Y]$ , we can compute this directly, or use

$$\widehat{\text{var}[Y]} = \frac{N}{N-1} \hat{p}(1 - \hat{p}).$$

An unbiased estimator for  $\text{var}[\hat{p}]$  is then

$$\widehat{\text{var}[\hat{p}]} = \frac{1}{N} \frac{N}{N-1} \hat{p}(1 - \hat{p}) = \frac{1}{N-1} \hat{p}(1 - \hat{p}).$$

The following are 20 simulated tosses of three coins with  $p = 0.5$  (Coin 1),  $p = 0.6$  (Coin 2), and  $p = 0.9$  (Coin 3), and the corresponding estimates, and (estimates of) the estimator variances and standard errors.

```
set.seed(13) # Initialize random number generator for replicability
N <- 20
Coin1 <- rbinom(N,1,0.5) # 20 tosses of a fair coin
Coin2 <- rbinom(N,1,0.6) # 20 tosses of a slightly unfair coin
Coin3 <- rbinom(N,1,0.9) # 20 tosses of a very warped coin!
Tosses <- rbind(Coin1, Coin2, Coin3) # place outcomes into three rows
p_hat <- apply(Tosses,1,mean) # apply mean function to each row of Tosses
var_p_hat <- p_hat*(1-p_hat)/(N-1) # as per formula derived in notes
se_p_hat <- sqrt(var_p_hat)
cat("Coin Tosses\n")
Tosses
d <- cbind(p_hat,var_p_hat,se_p_hat)
cat("\nEstimation Results\n")
round(d, 4)
```

Coin Tosses

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]
Coin1	1	0	0	0	1	0	1	1	1	0	1	1	1
Coin2	1	1	0	1	1	0	1	1	1	0	1	1	0
Coin3	1	1	0	1	1	0	1	1	1	1	1	1	1

	[,14]	[,15]	[,16]	[,17]	[,18]	[,19]	[,20]
Coin1	1	1	0	0	1	1	1
Coin2	0	1	1	1	1	0	1
Coin3	0	1	1	1	1	1	1

Estimation Results

	p_hat	var_p_hat	se_p_hat
Coin1	0.65	0.0120	0.1094
Coin2	0.70	0.0111	0.1051
Coin3	0.85	0.0067	0.0819

### 4.3 Hypothesis Testing

To test if the population mean is equal to some specific numerical value  $\mu_0$ , we check if the sample mean is ‘improbably far’ from  $\mu_0$ . If it is, we construe this as evidence that the “null hypothesis”  $H_0 : E[Y] = \mu_0$  is false, and reject it in favor of the alternative  $H_A : E[Y] \neq \mu_0$ . But how far is “improbably” far? To provide an answer to this question fully, we need to derive the distribution of the sample mean when  $\mu = \mu_0$ , and to do so we need to know the distribution of  $Y$ . If all you know is that  $E[Y] = \mu$  and  $var[Y] = \sigma^2$ , then you do not have enough information to derive the distribution of the sample mean. We will explain how to find an approximation to the finite sample distribution in this situation later in the chapter.

In the case of the coin toss example, the structure of the problem does give us enough information to derive the finite sample distribution of the sample mean. Suppose  $N = 2$ . Then the possible values of the sample mean are 0,  $1/2$  and 1, corresponding to sample outcomes (0, 0), (0, 1) or (1, 0), and (1, 1) respectively. The corresponding probabilities are  $(1 - p)^2$ ,  $2p(1 - p)$ , and  $p^2$ . For  $N = 3$ , the possible outcomes for the sample mean are:

- 0, corresponding to outcome  $(0, 0, 0)$ , which occurs with probability  $(1 - p)^3$ ;
- $1/3$ , corresponding to outcomes with 1 head out of 3 tosses. There are  $\binom{3}{1} = 3$  such outcomes, so the probability is  $3p(1 - p)^2$ .
- $2/3$ , corresponding to outcomes with 2 heads out of 3 tosses. There are  $\binom{3}{2} = 3$  such outcomes, so the probability is  $3p^2(1 - p)$ .
- 1, corresponding to outcome  $(1, 1, 1)$  which occurs with probability  $p^3$ .

For a sample of size  $N$ , the possible values of the sample mean are  $i/N$ ,  $i = 0, 1, \dots, N$ , each corresponding to a set of  $\binom{N}{i}$  outcomes comprising  $i$  heads out of  $N$  tosses, so the probability of obtaining a sample mean of  $i/N$  is

$$\Pr\left[\bar{Y} = \frac{i}{N}\right] = \binom{N}{i} p^i (1 - p)^{N-i}, i = 0, 1, 2, \dots, N. \quad (4.8)$$

We can use Eq. 4.8 to help us decide whether or not to reject the hypothesis that the coin is fair. Suppose we have a sample of 20 coin tosses, and suppose that the coin is in fact fair, i.e., that  $p$  is indeed equal to 0.5. The following is the probability distribution function of the sample mean  $f(i/N) = \Pr[\bar{Y} = i/N]$ ,  $i = 0, 1, \dots, N$  calculated using Eq. 4.8.

```
p <- 0.5
N <- 20
i <- 0:N      # i integers from 0 to 20
phat <- 0:N/N  # possible values of sample means
Pr_phat <- choose(N,i)*p^i*(1-p)^(N-i)
dim(Pr_phat) <- c(1,N+1) # make into row vector for presentation
colnames(Pr_phat) = paste0("p_hat=",i/N)
rownames(Pr_phat) = "Prob"
noquote(format(Pr_phat, scientific=T,digits=6)) # another way to print to screen
```

```
      p_hat=0      p_hat=0.05  p_hat=0.1   p_hat=0.15  p_hat=0.2   p_hat=0.25
Prob 9.53674e-07  1.90735e-05  1.81198e-04  1.08719e-03  4.62055e-03  1.47858e-02
      p_hat=0.3   p_hat=0.35  p_hat=0.4   p_hat=0.45  p_hat=0.5   p_hat=0.55
Prob 3.69644e-02  7.39288e-02  1.20134e-01  1.60179e-01  1.76197e-01  1.60179e-01
      p_hat=0.6   p_hat=0.65  p_hat=0.7   p_hat=0.75  p_hat=0.8   p_hat=0.85
Prob 1.20134e-01  7.39288e-02  3.69644e-02  1.47858e-02  4.62055e-03  1.08719e-03
      p_hat=0.9   p_hat=0.95  p_hat=1
Prob 1.81198e-04  1.90735e-05  9.53674e-07
```

```
library(latex2exp)
my_theme <- theme_bw() + theme(axis.title=element_text(size=10), aspect.ratio = 0.8)
bar_dat <- data.frame(phat=phat, Pr_phat = as.vector(Pr_phat))
bar_dat %>% ggplot(aes(x=phat, y=Pr_phat)) +
  geom_bar(stat="identity", width=0.015) + ylab(TeX("$\\Pr[\\hat{p}=i/20]$")) +
  xlab("possible values of sample mean, i/20, i=0,1,...,20") +
  my_theme + theme(aspect.ratio = 0.5)
```

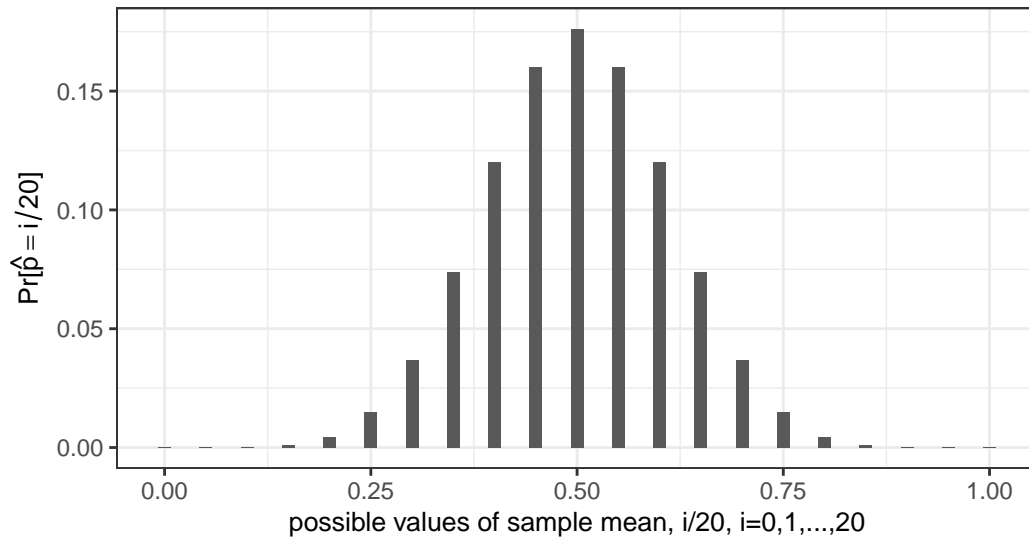


Figure 4.1: Distribution of sample mean,  $N=20$ ,  $p=0.5$ .

Notice that there is non-zero probability on every possible outcome of the sample mean. This means that any reasonable decision rule that we use to reject or not reject the null hypothesis will carry a non-zero probability of rejection even when the null hypothesis is true (we call this a “Type 1 error”). For example, suppose we use the rule “Reject  $p = 0.5$  in favor of the alternative  $p \neq 0.5$  if the frequency of heads  $\hat{p}$  is less than 0.3 or greater than 0.7”, which seems not unreasonable. We can calculate from the table above that in using this rule, there is a probability of approximately

```
round(sum(Pr_phat[i/N<0.3])+sum(Pr_phat[i/N>0.7]),4)
```

```
[1] 0.0414
```

that we reject the null even though  $p$  is in fact equal to 0.5. We can reduce the probability of Type 1 error by allowing for a larger range for  $\hat{p}$  (perhaps reject if  $\hat{p} < 0.05$  or  $\hat{p} > 0.95$ ), but then the test loses power to reject a false hypothesis (i.e., the probability of failing to reject a wrong hypothesis – a “Type 2 error” – increases). In practice, researchers usually opt for rules such that the probability of an incorrect rejection of a true hypothesis is around 0.01, or 0.05, or 0.10.

#### 4.4 Asymptotic Analysis

Our discussion so far has been “finite sample analysis”. Asymptotic analysis refers to results that apply “in the limit”, as the sample size goes to infinity. It serves to approximate the properties of estimators in large samples. We continue to focus on the sample mean, which we now denote as  $\bar{Y}_N$  to indicate the sample size used to calculate it.

#### 4.4.1 Consistency and the Law of Large Numbers

We have mentioned the desirability of larger sample sizes. For the general problem of estimating the population mean of a random variable  $Y$  with mean  $\mu$  and variance  $\sigma^2$  using the sample mean, we have  $\text{var}[\bar{Y}_N] = \sigma^2/N \rightarrow 0$  as  $N \rightarrow \infty$ . Since  $\bar{Y}_N$  is unbiased, and its variance collapses to zero as the sample size goes to infinity, the estimator converges to the population mean as the sample size grows larger and larger.

The convergence of  $\bar{Y}_N$  to  $\mu$  is not quite the same as the convergence of, say, the deterministic sequence  $1/N$  to zero. In the latter case, I know that if  $N$  is large enough, then  $1/N$  will *definitely* be within a certain distance of 0. For instance, if  $N > 1000$ , then  $1/N < 0.001$  *for sure*. In the case of  $\bar{Y}_N$ , which is a sequence of random variables, we cannot make such a definite claim.

The convergence concept we use for random variables is “convergence in probability”. A sequence of random variables  $X_N$  is said to **converge in probability** to some value  $c$  as  $N \rightarrow \infty$  if for any  $\epsilon > 0$  (no matter how small), the *probability* that  $|X_N - c| > \epsilon$  goes to zero as  $N \rightarrow \infty$ . This allows for some probability that the distance between  $X_N$  and  $c$  exceeds  $\epsilon$  at any sample size  $N$ , but as  $N$  increases towards infinity, this probability becomes vanishingly small. We write  $\text{plim} X_N = c$  or  $X_N \xrightarrow{p} c$ . We can extend this definition to “convergence in probability to a random variable”: we say that  $X_N \xrightarrow{p} Z_N$  if  $X_N - Z_N \xrightarrow{p} 0$ .

In the context of parameter estimation, we say that an estimator is **consistent** if it converges in probability to the true value of the parameter it is estimating. The sample mean  $\bar{Y}_N$  is a consistent estimator for  $\mu$  under quite general conditions. This result is known as a **Law of Large Numbers**. There are several laws of large numbers, each describing a set of conditions which, if met, guarantee the consistency of the sample mean. We state one such law below:

**Theorem 4.2** (Khinchine’s Weak Law of Large Numbers, WLLN). *If  $\{Y_i\}_{i=1}^N$  are i.i.d. with  $E[Y_i] = \mu < \infty$  for all  $i$ , then  $\bar{Y}_N \xrightarrow{p} \mu$ .*

There are other kinds of convergence concepts for sequences of random variables, but for the moment we consider only convergence in probability. The theorem above is referred to as a *weak* law of large numbers because the convergence concept used is convergence in probability, and there are “stronger” forms of probabilistic convergence.

The following is a result concerning convergence in probability that is used frequently:

**Proposition 4.1.** *If  $g(\cdot)$  is a continuous function, then*

$$X_N \xrightarrow{p} c \Rightarrow g(X_N) \xrightarrow{p} g(c). \quad (4.9)$$

*That is, if  $\text{plim} X_N$  exists, and  $g(\cdot)$  is continuous, then  $\text{plim} g(X_N) = g(\text{plim} X_N)$ .*

For example, if  $X_N$  converges in probability to  $c$ , then  $X_N^2 \xrightarrow{p} c^2$ .

Result Eq. 4.9 extends to continuous functions of multiple variables. This implies that if  $X_N \xrightarrow{p} c_x$  and  $Z_N \xrightarrow{p} c_z$ , then

- $X_N + Z_N \xrightarrow{p} c_x + c_z$ ,
- $X_N Z_N \xrightarrow{p} c_x c_z$ ,
- $X_N / Z_N \xrightarrow{p} c_x / c_z$ , as long as  $c_z$  is not zero,

and similarly for other continuous functions of  $X_N$  and  $Z_N$ .

**Example 4.2.** Suppose  $\{Y_i\}_{i=1}^N$  is an i.i.d. sample, with  $E[Y_i] = \mu < \infty$  and  $\text{var}[Y_i] = \sigma^2 < \infty$  for all  $i$ . We show that the biased estimator

$$\widetilde{\sigma}_N^2 = \frac{1}{N} \sum_{i=1}^N (Y_i - \bar{Y}_N)^2 = \frac{1}{N} \sum_{i=1}^N Y_i^2 - \bar{Y}_N^2$$

is consistent for the population variance  $\sigma^2$ . Since  $\{Y_i\}_{i=1}^N$  are i.i.d., so are  $\{Y_i^2\}_{i=1}^N$ . Furthermore,  $E[Y_i^2] = \sigma^2 + \mu^2 < \infty$ , so  $\frac{1}{N} \sum_{i=1}^N Y_i^2 \xrightarrow{p} \sigma^2 + \mu^2$ . Since  $\bar{Y}_N \xrightarrow{p} \mu$  and “power of two” is a continuous function, we have  $\bar{Y}_N^2 \xrightarrow{p} \mu^2$ . Therefore  $\widetilde{\sigma}_N^2$  converges in probability to  $\sigma^2 + \mu^2 - \mu^2 = \sigma^2$ . This example shows that consistent estimators can be biased in finite samples.

**Example 4.3.** Since  $\widehat{\sigma}_N^2 = \frac{N}{N-1} \widetilde{\sigma}_N^2$ , and because  $\frac{N}{N-1} \rightarrow 1$  and  $\widetilde{\sigma}_N^2 \xrightarrow{p} \sigma^2$ , we have  $\widehat{\sigma}_N^2 \xrightarrow{p} \sigma^2$ .

**Example 4.4.** Since both  $\widehat{\sigma}_N^2$  and  $\widetilde{\sigma}_N^2$  are consistent estimators for  $\sigma^2$ , both  $(\widehat{\sigma}_N^2)^{1/2}$  and  $(\widetilde{\sigma}_N^2)^{1/2}$  are consistent estimators for  $\sigma$ .

Unbiasedness, as opposed to consistency, generally does not carry over to non-linear functions of estimators. We saw earlier that  $E[\hat{p}^2] > p^2$  despite  $E[\hat{p}] = p$ . The following is another example.

**Example 4.5.** Unbiasedness of an estimator  $\hat{\theta}$  for some parameter  $\theta$  does *not* imply unbiasedness of  $\hat{\theta}^{1/2}$  for  $\theta^{1/2}$ . In fact, we can see from  $\text{var}[\hat{\theta}^{1/2}] = E[\hat{\theta}] - E[\hat{\theta}^{1/2}]^2$  that  $E[\hat{\theta}^{1/2}]^2 < E[\hat{\theta}]$ . If  $\hat{\theta}$  is unbiased, we have  $E[\hat{\theta}^{1/2}]^2 < \theta$ . Taking square roots then gives

$$E[\hat{\theta}^{1/2}] < \theta^{1/2}.$$

Both  $(\widehat{\sigma}_N^2)^{1/2}$  and  $(\widetilde{\sigma}_N^2)^{1/2}$  are biased (but consistent) estimators for  $\sigma$ .

It may seem that unbiasedness (together with efficiency) is a more relevant way to judge an estimator than consistency since we never have infinite sample sizes, but consistency is still useful as it captures the idea of convergence to the population parameter as sample size increases. Furthermore, in more complex applications it can be difficult or impossible to find unbiased estimators, but straightforward to find consistent ones. We have also seen that it is easy to find consistent estimators of continuous functions of parameters once we have consistent estimators for the parameters.

Earlier we derived the distribution of the sample mean in the coin toss example, and calculated this distribution for a fair coin with sample size 20. We repeat this exercise, this time for a coin with  $p = 0.25$ , for sample sizes of 5, 10, 20, 100, 200 and 400. We present the probability distribution functions graphically in Fig. 4.2. The convergence in probability of the sample mean to the true value of  $p$  can be seen in these graphs.

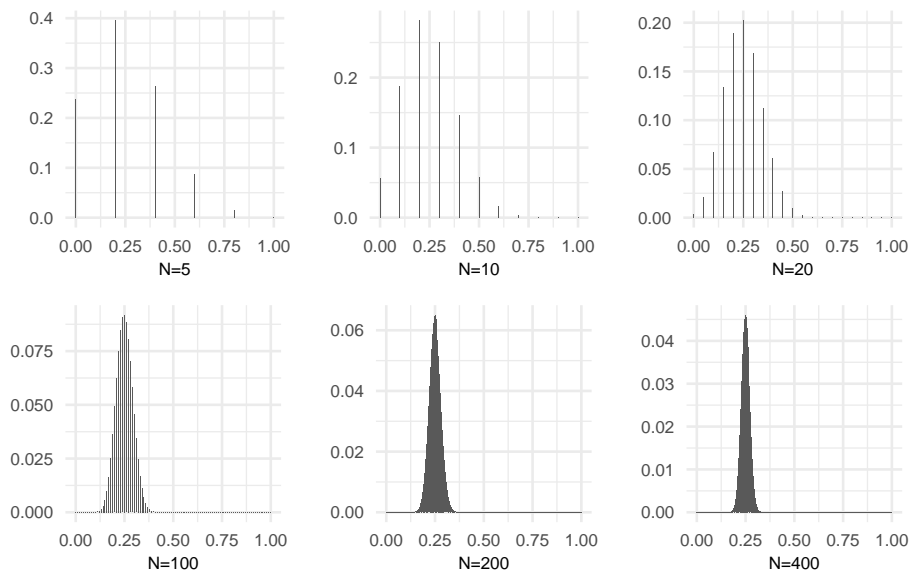
```
p <- 0.25 # Assume probability of heads = 0.25
samplesize <- c(5,10,20,100,200,400)
pdfs <- list() # to store our plots in the loop
for (j in 1:6){
  N <- samplesize[j]
  i <- 0:N
  phat <- 0:N/N
```



```

Pr_phat <- choose(N,i)*p^i*(1-p)^(N-i)
data <- data.frame(phat=phat,Pr_phat=Pr_phat)
pdfs[[j]] <- ggplot(data,aes(x=phat,y=Pr_phat))+
  geom_bar(stat="identity", width=0.005)+
  ylab("")+xlab(paste0('N=',N)) + theme_minimal() +
  theme(axis.text=element_text(size=7),
        axis.title=element_text(size=7))
}
(pdfs[[1]] | pdfs[[2]] | pdfs[[3]]) /
(pdfs[[4]] | pdfs[[5]] | pdfs[[6]])

```

Figure 4.2: Consistency of sample mean to  $p=0.25$ .

#### 4.4.2 Asymptotic Normality

The distribution of the sample mean in the example above, with  $p = 0.25$ , is unsurprisingly skewed in small samples because of the low probability of heads relative to tails. However, the shape of the distribution appears to quickly become quite symmetric as sample size grows, and appears to converge to a familiar bell-shaped distribution. Of course, in the limit the distribution collapses to a degenerate one with all of the probability at  $p = 0.25$ . This is because the variance of the sample mean,  $\text{var}[\hat{p}] = p(1-p)/N$  goes to zero as  $N \rightarrow \infty$ . Suppose, however, that we scale the sample mean (after subtracting  $p$ ) by  $\sqrt{N}$ , i.e., suppose we look at the distribution of

$$\sqrt{N}(\hat{p} - p). \quad (4.10)$$

This random variable has mean 0 and a non-collapsing variance  $Np(1-p)/N = p(1-p)$ . We can then talk about the shape of Eq. 4.10 as  $N \rightarrow \infty$  without the distribution collapsing to a single point. The plots below show the same distributions as above, but after centering and scaling as in Eq. 4.10.

```

p <- 0.25 # Assume probability of heads = 0.25
samplesize <- c(5,10,20,100,200,400)
pdfs <- list() # to store our plots in the loop
for (j in 1:6){
  N <- samplesize[j]
  i <- 0:N
  phat <- 0:N/N
  phat_scaled <- sqrt(N)*(phat - p)
  Pr_phat_scaled <- choose(N,i)*p^i*(1-p)^(N-i)
  data <- data.frame(phat_scaled=phat_scaled,Pr_phat_scaled=Pr_phat_scaled)
  pdfs[[j]] <- ggplot(data,aes(x=phat_scaled,y=Pr_phat_scaled)) +
    geom_bar(stat="identity", width=0.05) + ylab("") +
    xlab(paste0('N=',N)) + xlim(c(-2.5,2.5)) + theme_minimal() +
    theme(axis.text=element_text(size=8),
          axis.title=element_text(size=7))
}
(pdfs[[1]] | pdfs[[2]] | pdfs[[3]]) /
(pdfs[[4]] | pdfs[[5]] | pdfs[[6]])

```

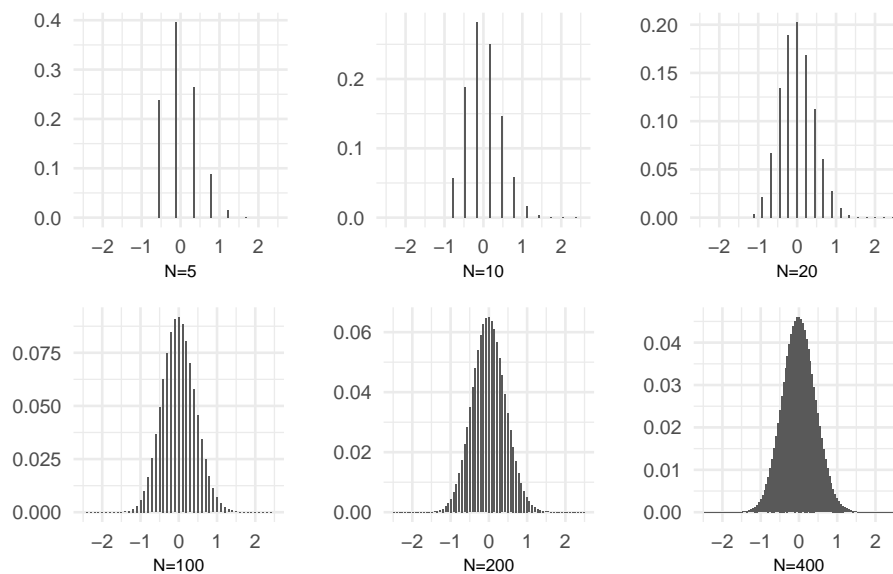


Figure 4.3: Convergence of distribution to normal (pdf view).

The distributions appear to take the shape of a normal distribution as sample size increases. Of course, the sample mean in the coin toss example is a discrete random variable, whereas a normally distributed random variable is a continuous one. The notion of a discrete pdf converging to a continuous one is best thought of in terms of their cdfs. In the figure below, we juxtapose the cdf of the distribution of  $\sqrt{N}(\hat{p} - p)$  at  $N = 400$  (which is a step function) with the cdf of the normal distribution with mean 0 and variance  $p(1 - p)$  where  $p = 0.25$ .

```

p <- 0.25 # Assume probability of heads = 0.25
N = 400
i <- 0:N

```

```

phat <- 0:N/N
phat_scaled <- sqrt(N)*(phat - p)
cdf_phat_scaled <- cumsum(choose(N,i)*p^i*(1-p)^(N-i))
cdf_norm <- pnorm(phat_scaled,0,sqrt(p*(1-p)))
data1 <- data.frame(phat_scaled=phat_scaled,
                    cdf_phat_scaled=cdf_phat_scaled)
data2 <- data.frame(phat_scaled=phat_scaled,
                    cdf_norm=cdf_norm)

ggplot()+
  geom_step(data=data1,aes(x=phat_scaled,y=cdf_phat_scaled,color="blue"),
            direction="vh")+
  geom_line(data=data2,aes(x=phat_scaled,y=cdf_norm,color="red"))+
  xlab(TeX("$\\sqrt{N}*(\\hat{p}-p)$", N = 400, p=0.25))+ylab("")+
  xlim(c(-2.5,2.5))+ylim(c(0,1.1)) + theme_minimal() +
  theme(axis.text=element_text(size=7), axis.title=element_text(size=8)) +
  scale_colour_manual(
    name = '',
    values = c('blue'='blue','red'='red'),
    labels = c('emp. cdf. phat_scaled','cdf. N(0,p(1-p))'))

```

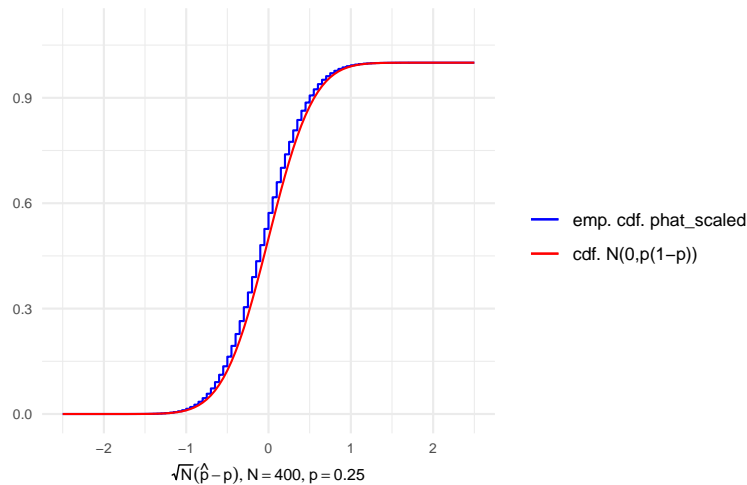


Figure 4.4: Convergence of distribution of sample mean to normal (cdf view).

#### 4.4.3 The Central Limit Theorem

The convergence of the cdf of the (centered and scaled) sample mean in the coin toss example to a normal cdf is an instance of the **Central Limit Theorem** (CLT), a key result in probability theory. As with the Law of Large Numbers, there are many CLTs, each listing out a set of conditions under which convergence to normality is guaranteed. We state one such CLT:

**Theorem 4.3** (Lindeberg-Levy CLT). *If  $\{Y_i\}_{i=1}^N$  are i.i.d. with  $E[Y_i] = \mu < \infty$  and  $\text{var}[Y_i] = \sigma^2 < \infty$  for all  $i$ , then*

$$\sqrt{N}(\bar{Y}_N - \mu) \xrightarrow{d} \text{Normal}(0, \sigma^2)$$

where  $\xrightarrow{d}$  means **convergence in distribution**, meaning that the cdf of the random variable on the left converges point-wise to the cdf of the distribution indicated on the right.

Our plots of the distribution of  $\sqrt{N}(\hat{p}_N - p)$  in the coin toss example suggests convergence in distribution to  $\text{Normal}(0, p(1 - p))$ . The sample  $\{Y_i\}$  in the coin toss example does in fact meet the requirements of the Lindeberg-Levy CLT, so we can claim that  $\sqrt{N}(\hat{p}_N - p) \xrightarrow{d} \text{Normal}(0, p(1 - p))$

Sometimes we want to indicate that a sequence of random variables  $X_N$  converges in distribution to the cdf of some random variable  $X$ . To so do, we write  $X_N \xrightarrow{d} X$ .

**Proposition 4.2** (Properties of convergence in distribution).

- (a) If  $g(\cdot)$  is a continuous function and  $X_N \xrightarrow{d} X$ , then  $g(X_N) \xrightarrow{d} g(X)$ .
- (b) If  $X_N \xrightarrow{p} X$ , then  $X_N \xrightarrow{d} X$ .
- (c) If  $a_N \xrightarrow{p} a$  and  $X_N \xrightarrow{d} X$ , then  $a_N X_N \xrightarrow{d} aX$ .

**Example 4.6.** If  $X_N \xrightarrow{d} X \sim N(0, 1)$ , then  $X_N^2 \xrightarrow{d} X^2 \sim \chi_{(1)}^2$ , since the square of a standard normal is  $\chi^2(1)$ .

**Example 4.7.** If  $\sqrt{N}(\bar{Y}_N - \mu) \xrightarrow{d} \text{Normal}(0, \sigma^2)$  and  $s_N^2$  is any consistent estimator of  $\sigma^2$ , then  $1/s_N = (1/s_N^2)^{1/2}$  converges in probability to  $1/\sigma$ , and therefore

$$t = \frac{\sqrt{N}(\bar{Y}_N - \mu)}{s_N} = \frac{\bar{Y}_N - \mu}{\sqrt{s_N^2/N}} \xrightarrow{d} N(0, 1). \quad (4.11)$$

If  $\sqrt{N}(\bar{Y}_N - \mu) \xrightarrow{d} \text{Normal}(0, \sigma^2)$ , we would be justified, in large enough samples, to say that the distribution of  $\sqrt{N}(\bar{Y}_N - \mu)$  is approximately  $\text{Normal}(0, \sigma^2)$ , or that  $\bar{Y}$  is approximately  $N(\mu, \sigma^2/N)$ . This last statement is sometimes written  $\bar{Y}_N \overset{a}{\sim} N(\mu, \sigma^2/N)$ , where the “a” stands for “approximately” (some take “a” to stand for “asymptotically”).

Result Eq. 4.11 is useful for hypotheses testing when one is unable or unwilling to make an assumption regarding the distribution of the sample. Suppose  $\{Y_i\}_{i=1}^N$  is an i.i.d. sample with  $E[Y_i] = \mu$  and  $\text{var}[Y_i] = \sigma^2$ . The sample mean  $\bar{Y}$  is a consistent estimator for  $\mu$  and  $\widehat{\sigma^2} = \frac{1}{N-1} \sum_{i=1}^N (Y_i - \bar{Y})^2$  is a consistent estimator for  $\sigma^2$ . To test the null hypothesis  $H_0 : \mu = \mu_0$ , where  $\mu_0$  is some numerical value, we need to compute the distribution of the sample mean, but you cannot do this unless you know the distribution of each  $Y_i$ . Result Eq. 4.11, however, tells us that if our sample size is large enough, then under the null hypothesis,

$$t = \frac{\bar{Y}_N - \mu_0}{\sqrt{s_N^2/N}} \overset{a}{\sim} \text{Normal}(0, 1). \quad (4.12)$$

It suggests that we use the decision rule “reject the null if  $|t| > c_\alpha$ ” where  $c_\alpha$  is that value such that  $\Pr[|t| > c_\alpha] = \alpha$ , where  $\alpha$  is the chosen “level of significance” of the test, i.e., the probability of rejecting the null when it is true, and where the value  $c_\alpha$  is found from the normal distribution. For 0.01, 0.05, 0.10 levels of significance, the appropriate values of  $c_\alpha$  are approximately

```
round(qnorm(c(0.995, 0.975, 0.95)), 3)
```

```
[1] 2.576 1.960 1.645
```

respectively. The 0.05 level of significance test, in particular, says to reject  $H_0 : \mu = \mu_0$  if the

absolute distance from the sample mean to the hypothesized value  $\mu_0$  is more than 1.96 (or approximately 2) standard errors.

A test based on the statistic in Eq. 4.12 and rejection values (or ‘critical values’) based on the Normal(0, 1) distribution, would be an approximate test in the sense that the true significance level may not be exactly  $\alpha$ , as intended. Nonetheless, it is a way forward in a situation where an exact test is unavailable. Even where the exact distribution is available, such as in our coin toss example, the approximate test using Eq. 4.12 can be a convenient approximation.

**Example 4.8.** Earlier we showed an example of 20 tosses of three coins, where the true value of  $p$  for coins 1, 2, and 3 are 0.5, 0.6, and 0.9 respectively. We replicate the results below, this time also computing the corresponding t-statistics for the hypothesis  $H_0 : p = 0.5$ , i.e., we compute

$$t = \frac{\hat{p} - 0.5}{\sqrt{\frac{\hat{p}(1-\hat{p})}{N-1}}}$$

```
set.seed(13) # Initialize random number generator for replicability
N <- 20
Coin1 <- rbinom(N,1,0.5) # 20 tosses of a fair coin
Coin2 <- rbinom(N,1,0.6) # 20 tosses of a slightly unfair coin
Coin3 <- rbinom(N,1,0.9) # 20 tosses of a very warped coin!
Tosses <- rbind(Coin1, Coin2, Coin3) # place outcomes into three rows
p_hat <- apply(Tosses,1,mean) # apply mean function to each row of Tosses
var_p_hat <- p_hat*(1-p_hat)/(N-1) # as per formula derived in notes
se_p_hat <- sqrt(var_p_hat)
t_stat <- (p_hat-0.5)/se_p_hat
p_val <- 2*(1-pnorm(t_stat,0,1))
# Output
d <- cbind(p_hat,se_p_hat,t_stat,p_val)
round(d,4)
```

	p_hat	se_p_hat	t_stat	p_val
Coin1	0.65	0.1094	1.3708	0.1704
Coin2	0.70	0.1051	1.9024	0.0571
Coin3	0.85	0.0819	4.2726	0.0000

Using the asymptotic tests, we make the following (approximate) conclusions:

- we (correctly) fail to reject the hypothesis that Coin 1 is fair at any of the usual levels of significance.
- we (correctly) reject the hypothesis for Coin 2 at 0.1 level of significance, but (incorrectly) fail to reject at the 0.05 level of significance.
- Fairness of Coin 3 is resoundingly (and correctly) rejected at all conventional levels of significance.

The result for Coin 2 illustrates the fact that it can be hard to reject a mildly incorrect hypothesis. All tests have poor power in such cases. The results for Coin 1 and Coin 3 turned out to be correct in this example, but it should be remembered that there were non-zero probabilities of rejecting fairness for Coin 1, and not rejecting fairness for Coin 3.

In addition to the t-statistic, we also compute the p-value, i.e., the probability that the t-

statistic, prior to realization, would exceed the value realized, in absolute terms. We reject a hypothesis at  $\alpha$  level of significance if the p-value is smaller than  $\alpha$ .

The statistic in Eq. 4.12 is sometimes called a  $z$ -statistic, and often called a  $t$ -statistic because its exact distribution would be the  $t$ -distribution (with degree of freedom  $N - 1$ ) if the sample were normally distributed, i.e., if  $Y_i \sim \text{Normal}(\mu, \sigma^2)$  for all  $i$ , then

$$t = \frac{\bar{Y}_N - \mu}{\sqrt{\widehat{\sigma}_N^2/N}} \sim t_{(N-1)}.$$

Since  $Y_i$  is not normal in the coin toss example, this result does not apply, and we rely on the asymptotic result. Of course, the  $t$ -distribution is itself approximately standard normal when  $N$  is large, so in this case, the  $t$ -statistic converges to the standard normal for *two* reasons: because of the central limit theorem, and because the  $t$ -distribution anyway converges to the standard normal as the degree of freedom goes to infinity. The usefulness of result Eq. 4.12 is in its applicability regardless of the distribution of the sample, when sample sizes are large enough.

## 4.5 Exercises

**Exercise 4.1.** Prove the last equality in Eq. 4.6.

**Exercise 4.2.** Prove the last equality in Eq. 4.4.

**Exercise 4.3.** Suppose  $\hat{\theta}$  is an unbiased estimator. Prove that  $g(\hat{\theta})$  is an unbiased estimator of  $g(\theta)$  if  $g(\theta)$  has the form  $g(\theta) = a + b\theta$ .

**Exercise 4.4.** A function  $g(\cdot)$  that is differentiable and concave has the property that the function lies on or under every tangent line. The functions  $g(x) = \sqrt{x}$ ,  $x \geq 0$  and  $g(x) = \ln x$ ,  $x > 0$  are both examples of concave functions. Follow the steps below to prove **Jensen's Inequality**, which says that if  $g(\cdot)$  is concave, then

$$E[g(X)] \leq g(E[X]).$$

*Step 1:* Let  $l(x) = a + bx$  be the tangent line of the concave function  $g(x)$  at the point  $(E[X], g(E[X]))$ , i.e.,  $l(x) = a + bx$  satisfies

$$l(x) = a + bx \geq g(x) \quad \text{and} \quad l(E[X]) = a + bE[X] = g(E[X]).$$

*Step 2:* Prove Jensen's Inequality by using the fact that if  $f_1(x) \leq f_2(x)$ , then  $E[f_1(X)] \leq E[f_2(X)]$ .

**Exercise 4.5.** Prove the last equality in Eq. 4.7.

**Exercise 4.6.** For the coin toss example, compute the probability of rejecting the null hypothesis  $H_0 : p = 0.5$  using the rule "Reject  $H_0$  if  $\hat{p} = \bar{Y}$  is strictly greater than 0.7 or strictly less than 0.3" for values of  $p = 0.05, 0.1, \dots, 0.90, 0.95$  when  $N = 20$ . Plot these probabilities against  $p$ . Repeat the exercise for  $N = 50$ .

*The Bias-Variance Trade-off*

**Exercise 4.7.** It can be shown that if  $Y_i$ ,  $i = 1, 2, \dots, N$  are i.i.d. draws from a normal distribution with mean  $\mu$  and variance  $\sigma^2$ , then the variance of the unbiased variance estimator  $\widehat{\sigma^2}$  defined in Eq. 4.5 is  $\text{var}[\widehat{\sigma^2}] = \frac{2\sigma^4}{N-1}$ . Because  $\widehat{\sigma^2}$  is an unbiased estimator, its MSE is also  $\frac{2\sigma^4}{N-1}$ .

- Show that the **biased** estimator  $\widetilde{\sigma^2}$  defined in Eq. 4.4 has a smaller variance than  $\widehat{\sigma^2}$ .
- Show that  $\text{MSE}[\widetilde{\sigma^2}] = \frac{2N-1}{N^2}\sigma^4$ .
- Show that  $\text{MSE}[\widetilde{\sigma^2}] < \text{MSE}[\widehat{\sigma^2}]$ .

*This is an example where MSE can be improved by trading off some bias for a reduced variance. Note that the arguments here have assumed that  $Y$  is normally distributed.*

**Exercise 4.8.** Consider the coin toss example with  $N = 10$ . Let  $\hat{p} = (1/10) \sum_{i=1}^{10} Y_i$ , and let  $\tilde{p} = (1/11) \sum_{i=1}^{10} Y_i$  be an alternative estimator. We know  $\hat{p}$  is unbiased, and therefore  $\tilde{p}$  is biased (downwards).

- Show that the variance of  $\tilde{p}$  is  $(10/121)p(1-p)$  (which is lower than  $\text{var}[\hat{p}] = p(1-p)/10$ ).
- Find the *MSE* of  $\hat{p}$ .
- Find the *MSE* of  $\tilde{p}$ .
- Show that  $\text{MSE}[\tilde{p}] < \text{MSE}[\hat{p}]$  if  $p < 21/31$ .

*Remark: The estimator  $\tilde{p}$  is biased but it has a lower variance than the unbiased  $\hat{p}$ . It turns out that this bias-variance trade-off is favorable toward minimizing MSE only if  $p < 21/31$ . If it is believed that  $p < 21/31$ , and the objective is minimizing MSE, then the alternate estimator may be preferred.*

## 4.6 Prediction

Suppose you have an iid sample  $Y_1, Y_2, \dots, Y_N$  of draws from random variable  $Y$  with mean  $E[Y] = \mu$  and variance  $\text{var}[Y] = \sigma^2$ . Your interest is in predicting the value of the next independent draw  $Y_{N+1}$ . In the previous chapter we learnt that the point prediction that minimizes the mean squared prediction error is the expectation conditional on the information set  $X$ . In this application, the  $X$  is just your sample  $Y_1, Y_2, \dots, Y_N$  and the optimal prediction is the conditional expectation  $E[Y_{N+1}|Y_1, \dots, Y_N]$ . Since it is assumed that the sample is iid, the conditional mean reduces to the unconditional mean  $E[Y_{N+1}] = \mu$ . You decide to estimate the sample mean from your iid sample, and use this as your predictor, i.e., you choose

$$\widehat{Y}_{N+1} = \bar{Y} = \frac{1}{N} \sum_{i=1}^N Y_i.$$

What is the MSPE for this predictor? The conditional MSPE when predicting  $Y_{N+1}$  using  $\bar{Y}$  estimated over  $Y_1, \dots, Y_N$  is

$$\begin{aligned} \text{MSPE}(Y_{N+1}, \bar{Y}|X) &= E[(Y_{N+1} - \bar{Y})^2|X] \\ &= E[(Y_{N+1} - \mu + \mu - \bar{Y})^2|X] \\ &= E[(Y_{N+1} - \mu)^2|X] + E[(\mu - \bar{Y})^2|X] + 2E[(Y_{N+1} - \mu)(\mu - \bar{Y})|X] \\ &= E[(Y_{N+1} - \mu)^2] + (\mu - \bar{Y})^2 + 2(\mu - \bar{Y})E[(Y_{N+1} - \mu)|X] \\ &= \text{var}[Y_{N+1}] + (\bar{Y} - \mu)^2 \end{aligned}$$

where we have used the assumption that  $Y_{N+1}$  is independent of the sample  $X = \{Y_1, \dots, Y_N\}$ . Taking expectation over  $X$ , we get the unconditional MSPE to be

$$\begin{aligned} MSPE(Y_{N+1}, \bar{Y}) &= E[MSPE(Y_{N+1}, \bar{Y}|X)] \\ &= E[\text{var}[Y_{N+1}]] + E[(\bar{Y} - \mu)^2] \\ &= \text{var}[Y_{N+1}] + \text{var}[\bar{Y}]. \end{aligned}$$

That is, the MSPE is

$$MSPE(Y_{N+1}, \bar{Y}) = \sigma^2 + \frac{\sigma^2}{N}.$$

The first term is due to the fact that we are predicting a new observation outside of the sample. The second term comes from the variance of the sample mean. We can estimate the MSPE by replacing  $\sigma^2$  with  $\widehat{\sigma}^2$ , i.e.,

$$\widehat{MSPE}(Y_{N+1}, \bar{Y}) = \widehat{\sigma}^2 + \frac{\widehat{\sigma}^2}{N}. \quad (4.13)$$

Often the prediction is reported as

$$\text{prediction} \pm 2\sqrt{\widehat{MSPE}}.$$

This is often called the “0.95 Prediction Interval”.<sup>3</sup> Some people use 1.96 instead of 2.

Two remarks: first, since the  $MSPE(Y_{N+1}, \bar{Y}|X) = E[(Y_{N+1} - \hat{\mu})^2]$ , one might be tempted to estimate it with

$$\frac{1}{N} \sum_{i=1}^N (Y_i - \bar{Y})^2 \quad \text{or} \quad \frac{1}{N-1} \sum_{i=1}^N (Y_i - \bar{Y})^2$$

This is, of course, merely an estimate of  $\sigma^2$ , and would underestimate the MSPE. This is because you are using “in-sample” or “training” data to estimate MSPE, and the result does not generalize to out-of-sample predictions.

#### 4.6.1 Exercises

The following questions are based on the above prediction example.

**Exercise 4.9.** Explain why using

$$\widehat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^N (Y_i - \bar{Y})^2$$

in Eq. 4.13 results in an unbiased estimate of  $\widehat{MSPE}(Y_{N+1}, \bar{Y})$ .

---

<sup>3</sup>The 0.95 Confidence Interval, on the other hand, uses only the sample error from estimating the mean, i.e.,

$$\hat{\mu} \pm 2\sqrt{\widehat{\sigma}^2/N}.$$

This is the 0.95 confidence interval for the sample mean, and is a measure of uncertainty about the estimate of the sample mean, not the prediction.



**Exercise 4.10.** Suppose instead of the estimating the MSPE using Eq. 4.13, you did the following instead: divide your sample of  $N$  observations into two subsamples of sizes  $N_1$  and  $N_2$ . In particular, divide your  $N$  observations into observations  $i = 1, 2, \dots, N_1$  and  $j = N_1 + 1, N_1 + 2, \dots, N_1 + N_2$ . (Perhaps  $N_2$  might be  $N/5$ , but the size of split is not so important). Then estimate  $\mu$  over the  $N_1$  sample, and estimate your MSPE over the  $N_2$  sample, i.e.,

$$\tilde{\mu} = \frac{1}{N_1} \sum_{i=1}^{N_1} Y_i \quad \text{and} \quad \widetilde{MSPE}(Y_{N+1}, \tilde{\mu}) = \frac{1}{N_2} \sum_{j=N_1+1}^{N_1+N_2} (Y_j - \tilde{\mu})^2.$$

Show that  $\widetilde{MSPE}(Y_{N+1}, \tilde{\mu})$  is an unbiased estimator of  $MSPE(Y_{N+1}, \tilde{\mu})$ , which is

$$MSPE(Y_{N+1}, \tilde{\mu}) = \sigma^2 + \frac{\sigma^2}{N_1}.$$

*Remark:* An “advantage” of this method of estimating MSPE is that we did not have to estimate  $\sigma^2$ , but this is a hardly an advantage since we have unbiased estimators of  $\sigma^2$ . The major disadvantage of this method is that you are estimating the sample mean using a smaller sample, which increases its sampling variability, and increases the MSPE. Since we have easily derived the formula for the MSPE in this simple application, there is no advantage to the method proposed here. However, in more advanced applications, the form of the MSPE might not be so easy to derive. In those applications, and assuming a large enough sample size, it may be advantageous to estimate the MSPE in the manner described in this exercise. The “cross-validation” method used in machine learning is an extension of the method described here. The method proposed here may also be useful in the situation where the loss function is not squared error.

**Exercise 4.11.** Suppose instead of the sample mean, you use some other (possibly biased) estimator  $\hat{g}(X)$  where  $X = \{Y_1, \dots, Y_N\}$  to predict an independent observation  $Y_{N+1}$ . Show that the MSPE is

$$MSPE(Y_{N+1}, \hat{g}(X)) = \sigma^2 + \text{var}[\hat{g}(X)] + (E[\hat{g}(X)] - \mu)^2.$$

*Hint:* Write

$$\begin{aligned} MSPE(Y_{N+1}, \hat{g}(X)) &= E[(Y_{N+1} - \hat{g}(X))^2] \\ &= E[(Y_{N+1} - \mu + \mu - E[\hat{g}(X)] + E[\hat{g}(X)] - \hat{g}(X))^2] \\ &= E[A^2 + B^2 + C^2 + 2AB + 2AC + 2BC] \end{aligned}$$

where  $A = Y_{N+1} - \mu$ ,  $B = \mu - E[\hat{g}(X)]$  and  $C = E[\hat{g}(X)] - \hat{g}(X)$ . Then compute the expectation by first taking expectations conditional on  $X$ , then take expectations over  $X$ . When taking expectations conditional of  $X$ ,  $\hat{g}(X)$  is fixed. You should find that the last three terms drop out, and the  $E[A^2] = \sigma^2$ ,  $E[B^2]$  is the squared bias, and  $E[C^2]$  is the variance of  $\hat{g}(X)$ . When  $\hat{g}(X)$  is the sample mean, which is unbiased, the MSPE reduces to  $\sigma^2 + \sigma^2/N$ .

**Exercise 4.12.** In the coin toss example, suppose you want to use  $N$  independent coin tosses to predict the outcome of the next coin toss. Suppose your loss function is squared error. What is your optimal point prediction?

**Exercise 4.13.** Continuing with the coin toss example, suppose you have estimated the probability of Heads to be  $\hat{p}$ . Suppose now that your prediction  $\hat{Y}_{N+1}$  for the next toss must be either ‘heads’ or ‘tails’ (i.e., 1 or 0, you cannot give fractional answers) and your loss function is “0-1”, meaning that your loss is 1 if you are wrong, and 0 if you are right, i.e.,

$$L(Y_{N+1}, \hat{Y}_{N+1}) = \begin{cases} 1, & \text{if } Y_{N+1} \neq \hat{Y}_{N+1} \\ 0, & \text{if } Y_{N+1} = \hat{Y}_{N+1}. \end{cases}$$

Show that your optimal prediction is  $\hat{Y}_{N+1} = 1$  if  $\hat{p} \geq 0.5$ ,  $\hat{Y}_{N+1} = 0$  otherwise.

*Hint: You want to choose  $\hat{Y}_{N+1} = 1$  or 0 to minimize your expected loss, which is*

$$L(Y_{N+1} = 1, \hat{Y}_{N+1})Pr[Y_{N+1} = 1] + L(Y_{N+1} = 0, \hat{Y}_{N+1})Pr[Y_{N+1} = 0].$$

*Compare this value for the cases  $\hat{Y}_{N+1} = 1$  and  $\hat{Y}_{N+1} = 0$  and ask: which choice minimizes expected loss when  $Pr[Y_{N+1} = 1] \geq 0.5$ ? Which choice minimizes expected loss when  $Pr[Y_{N+1} = 0] < 0.5$ ?*

## Chapter 5

### Simple Linear Regression

Simple linear regression is a framework for developing empirical models of the form

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X \quad (5.1)$$

for the purpose of prediction, inferring causality from  $X$  to  $Y$ , testing hypotheses regarding  $X$  and  $Y$ , among other applications. This chapter describes and studies this framework. The simple linear regression framework will in many instances turn out to be insufficient for predictive, causal inference and testing applications, but it is a good place to start, both as a first step towards mastering the principles and technicalities of more advanced frameworks, and as a way to better understand the issues involved in these applications.

We focus on cross-sectional regressions in this chapter. Time series regressions are discussed in a later chapter. Our initial discussions will also focus on the problem of prediction, although we will also discuss hypothesis testing and issues involved in trying to interpret regression results as causal effects.

The R code in this chapter uses the following packages.

```
library(tidyverse) # For data handling and visualization
library(patchwork) # ]
library(gridExtra) # ] For plot management
library(latex2exp) # ]
```

#### 5.1 The Simple Linear Regression Framework

Suppose you have observations  $\{X_i, Y_i\}_{i=1}^N$  and you want to build an empirical model of the form Eq. 5.1 for predicting  $Y$  for new observations at given values of  $X$ . For example, you want to predict the price  $Y$  of new houses to be built  $X = x$  distance from town center, or for predicting how much new customers with income levels  $X = x$  are likely to spend on some product. We'll keep  $X$  and  $Y$  generic for the moment.

You know that the point prediction that minimize mean squared prediction error is the conditional mean (and suppose that minimizing mean squared prediction error is your objective). You plan to use the empirical model Eq. 5.1 as an estimate of the conditional expectation

$$E[Y|X] \approx \hat{E}[Y|X] = \hat{\beta}_0 + \hat{\beta}_1 X.$$

New observations with  $X = x$  will be predicted to have  $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 x$ . Estimating Eq. 5.1 amounts to using your data to come up with appropriate values of  $\hat{\beta}_0$  and  $\hat{\beta}_1$ . What method should you use? If you choose to use ordinary least squares (OLS) as described in Section 2.4, under what conditions would that give you “good” estimates of the conditional expectation? Are you able to give some indication of how much your predictions can be trusted? Is the supposed linear form of the conditional expectation appropriate in the first place?

We answer these questions by first laying out a set of assumptions, and showing that OLS works well in under these assumptions. Then we ask what happens when one or more of the assumptions fail. Later chapters will discuss how the basic framework can be modified to such situations. In practice, diagnostic assessments are made to ascertain if the assumptions in the basic framework are appropriate for your particular application. These will usually include both data-based approaches (are the assumptions consistent with the data?) and as well as assessments based on a knowledge of economics. Assumptions and estimation methods are then modified accordingly.

Let  $X$  and  $Y$  be your variables of interest, and let  $\{X_i, Y_i\}_{i=1}^N$  be your data sample. We suppose for the moment that you have a cross-sectional sample, i.e., you have data on a sample of individuals (this could be individual people, individual firms, ...) that can be considered to have been collected at a single point in time. We consider other data structures in later chapters.

**Assumption Set A:** Suppose that (A1) there are two values  $\beta_0$  and  $\beta_1$  such that the random variable  $\epsilon$ , defined as

$$\epsilon = Y - \beta_0 - \beta_1 X,$$

satisfies

$$(A2) \quad E[\epsilon|X] = 0,$$

$$(A3) \quad \text{var}[\epsilon|X] = \sigma^2.$$

Suppose also that

$$(A4) \quad \{X_i, Y_i\}_{i=1}^N \text{ is a random sample from the population of interest, and}$$

$$(A5) \quad \sum_{i=1}^N (X_i - \bar{X})^2 > 0.$$

It is important to understand that we are placing ourselves in a position prior to observing data, and treating the  $X_i$  and  $Y_i$  in the sample as random variables. We want to know if the methods to be proposed for building the empirical model will work in general situations conforming to the scenario described in Assumption Set A.

Assumption A2 implies that

$$E[Y|X] = \beta_0 + \beta_1 X \tag{5.2}$$

We call Eq. 5.2 the Population Regression Function (PRF). For obvious reasons, we call  $\beta_0$  the “intercept”, and  $\beta_1$  the “slope coefficient”. The variable  $\epsilon$  is called the “error” or “noise” term. The variable  $Y$  is the “dependent/explained/response/predicted variable”, or “regressand”. The variable  $X$  is the “independent/explanatory/control variable” or “regressor”. In predictive applications, it is often called the “predictor”. Assumption A2 also implies that

$$\text{cov}[X, \epsilon] = 0,$$

and

$$\beta_0 = E[Y] - \beta_1 E[X] \quad \text{and} \quad \beta_1 = \frac{\text{cov}[X, Y]}{\text{var}[X]}.$$

Assumption A3 is called “conditional homoskedasticity”. It says that the variance of the noise term does not depend on  $X$ . If A3 holds, we should expect that the spread of the sample realizations about the regression line should be more or less even, so that, roughly speaking, each observation is equally informative about the regression line. If A3 does not hold, we say that there is “conditional heteroskedasticity” in the noise term.<sup>1</sup> Incidentally, if A2 holds, then we can write A3 as  $E[\epsilon^2|X] = \sigma^2$ .

**Example 5.1.** The file `heterosk.csv` contains observations on three variables  $X$ ,  $Y$  and  $Z$ . We plot observations of  $Y$  against  $X$  in panel (a), and  $Z$  against  $X$  in panel (b) using data in `heterosk.csv`. Visually, assumption A3 appears appropriate if your data behaves as in (a) in Fig. 5.1, but almost surely does not hold if your data behaves as in (b). In the latter case, there is strong visual evidence that the variance of  $\epsilon$  increases with  $X$ .

```
df_het <- read_csv("data\\heterosk.csv", col_types = c("n", "n", "n"))
plt_het1a <- ggplot(data=df_het) + geom_point(aes(x=x, y=z), size=1) +
  ggtitle("(a)") + theme_classic()
plt_het1b <- ggplot(data=df_het) + geom_point(aes(x=x, y=y), size=1) +
  ggtitle("(b)") + theme_classic()
plt_het1a | plt_het1b
```

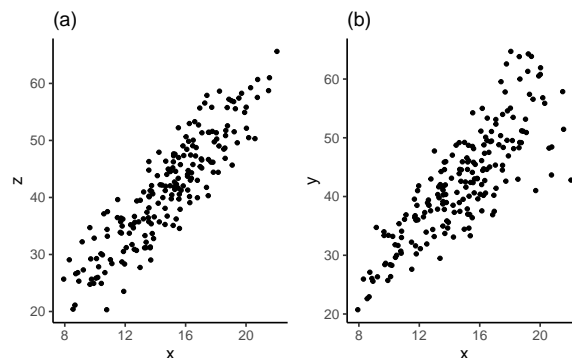


Figure 5.1: Data sets with and without heteroskedasticity

One application where the data may appear as in (b) is when  $Y$  is expenditure and  $X$  is income. Low income earners may have to spend most of their earnings on necessary purchases, with little room for variation, whereas high income earners have considerably more discretion in how much of their income to spend or save.

There is a lot packed into the phrase “random sample from the population of interest”. “Random sample” is usually taken to mean that the sample  $\{X_i, Y_i\}_{i=1}^N$  are independently and identically distributed draws. From a data sampling perspective, the term means that you have a representative draw from the population of interest, without favoring draws from segments of the population with certain characteristics. Suppose you are measuring the distribution of heights in an adult population of a certain country. If your sampling process somehow makes it more likely to sample males than females. The result will be that the distribution of heights

<sup>1</sup>The “-scedasticity” part of the words homoskedasticity and heteroskedasticity come from an ancient greek word that can be translated to “scatter”. “Homo-” and “Hetero-” come from words translating to “equal” and “different” respectively.

in your sample will not be representative of your population. You want each member of the population to have an equal chance of getting sampled, so that your sample has the same mix of characteristics as the entire population. If your sample comprises whole families, then there will be dependence in heights within members of the same family. In the latter case, you could still get a representative sample of the population, but you would need a much larger sample, and calculations of statistics like variances will need to take the dependence into consideration.

We will discuss each of these assumptions in greater detail shortly, when they might fail to hold, and the consequences. For now, we consider OLS estimation of the population regression function, and the properties of the estimators under Assumption Set A.

## 5.2 Ordinary Least Squares

Under Assumption A2, we can write

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i, \quad i = 1, 2, \dots, N.$$

For any estimators  $\hat{\beta}_0$  and  $\hat{\beta}_1$  (whether obtained by OLS or otherwise), define the **fitted values** to be

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i, \quad i = 1, 2, \dots, N \quad (5.3)$$

and the **residuals** to be

$$\hat{\epsilon}_i = Y_i - \hat{Y}_i = Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i, \quad i = 1, 2, \dots, N. \quad (5.4)$$

Ordinary Least Squares (OLS) chooses  $\hat{\beta}_0^{ols}$  and  $\hat{\beta}_1^{ols}$  to minimize the **sum of squared residuals** (SSR):

$$\text{OLS: Choose } \hat{\beta}_0, \hat{\beta}_1 \text{ to minimize } SSR = \sum_{i=1}^N \hat{\epsilon}_i^2 = \sum_{i=1}^N (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i)^2. \quad (5.5)$$

We have already seen in Section 2.4 how the minimization problem (5.5) can be solved. The first order conditions are:

$$\begin{aligned} \left. \frac{\partial SSR}{\partial \hat{\beta}_0} \right|_{\hat{\beta}_0^{ols}, \hat{\beta}_1^{ols}} &= -2 \sum_{i=1}^N (Y_i - \hat{\beta}_0^{ols} - \hat{\beta}_1^{ols} X_i) = 0 \\ \left. \frac{\partial SSR}{\partial \hat{\beta}_1} \right|_{\hat{\beta}_0^{ols}, \hat{\beta}_1^{ols}} &= -2 \sum_{i=1}^N (Y_i - \hat{\beta}_0^{ols} - \hat{\beta}_1^{ols} X_i) X_i = 0 \end{aligned} \quad (5.6)$$

where we use the notation  $\left. \frac{\partial SSR}{\partial \hat{\beta}_0} \right|_{\hat{\beta}_0^{ols}, \hat{\beta}_1^{ols}}$  to refer to the derivative  $\frac{\partial SSR}{\partial \hat{\beta}_0}$  evaluated at  $\hat{\beta}_0^{ols}$  and  $\hat{\beta}_1^{ols}$ , and likewise for  $\left. \frac{\partial SSR}{\partial \hat{\beta}_1} \right|_{\hat{\beta}_0^{ols}, \hat{\beta}_1^{ols}}$ . Solving the first order conditions in Eq. 5.6 gives

$$\begin{aligned} \hat{\beta}_0^{ols} &= \bar{Y} - \hat{\beta}_1^{ols} \bar{X} \\ \hat{\beta}_1^{ols} &= \frac{\sum_{i=1}^N (Y_i - \bar{Y}) X_i}{\sum_{i=1}^N (X_i - \bar{X}) X_i} = \frac{\sum_{i=1}^N (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^N (X_i - \bar{X})^2}. \end{aligned} \quad (5.7)$$

You showed in an exercise that the second-order condition for a strict global minimum holds, so that  $\hat{\beta}_0^{ols}$  and  $\hat{\beta}_1^{ols}$  do in fact solve the minimization problem (5.5). The OLS fitted values and OLS residuals are

$$\begin{aligned}\hat{Y}_i^{ols} &= \hat{\beta}_0^{ols} + \hat{\beta}_1^{ols} X_i \\ \hat{\epsilon}_i^{ols} &= Y_i - \hat{Y}_i^{ols} = Y_i - \hat{\beta}_0^{ols} - \hat{\beta}_1^{ols} X_i\end{aligned}$$

for  $i = 1, 2, \dots, N$ . The OLS Sample Regression Function (SRF) is the line

$$\hat{Y}^{ols} = \hat{\beta}_0^{ols} + \hat{\beta}_1^{ols} X.$$

In all that follows, we will drop the “OLS” superscript from the estimators, fitted values and residuals. These are assumed to be OLS estimators, fitted values and residuals. Where we need to discuss non-OLS estimators, we will indicate those estimators in some way.

We have seen that several algebraic identities hold under OLS estimation. We summarize them briefly below.

- The first-order conditions can also be written as

$$\begin{aligned}\sum_{i=1}^N \hat{\epsilon}_i &= 0 \\ \sum_{i=1}^N X_i \hat{\epsilon}_i &= 0\end{aligned}\tag{5.8}$$

It follows that OLS residuals have zero sample mean and zero sample covariance with the regressors:

$$\begin{aligned}\frac{1}{N} \sum_{i=1}^N \hat{\epsilon}_i &= 0 \\ \text{sample cov}[\hat{\epsilon}_i, X_i] &= \frac{1}{N} \sum_{i=1}^N (\hat{\epsilon}_i - \bar{\hat{\epsilon}})(X_i - \bar{X}) = \frac{1}{N} \sum_{i=1}^N X_i \hat{\epsilon}_i = 0.\end{aligned}$$

We describe the condition  $\sum_{i=1}^N X_i \hat{\epsilon}_i = 0$  by saying that the OLS residuals  $\hat{\epsilon}_i$  and the regressors  $X_i$  are **orthogonal**.

- The fitted values and the residuals are also orthogonal:

$$\sum_{i=1}^N \hat{Y}_i \hat{\epsilon}_i = \hat{\beta}_0 \sum_{i=1}^N \hat{\epsilon}_i + \hat{\beta}_1 \sum_{i=1}^N X_i \hat{\epsilon}_i = 0.$$

- The regressand and fitted values always have the same sample average

$$\bar{Y} = \bar{\hat{Y}}, \quad \text{where } \bar{\hat{Y}} = (1/N) \sum_{i=1}^N \hat{Y}_i$$

and the sample regression line (the fitted line) passes through the point  $(\bar{X}, \bar{Y})$ , i.e.,

$$\bar{Y} = \hat{\beta}_0 + \hat{\beta}_1 \bar{X}.$$

- We have the variance decomposition

$$\sum_{i=1}^N (Y_i - \bar{Y})^2 = \sum_{i=1}^N (\hat{Y}_i - \bar{\hat{Y}})^2 + \sum_{i=1}^N \hat{\epsilon}_i^2, \quad (5.9)$$

which is often read as “Sum of Squared Total = Sum of Squared Explained + Sum of Squared Residuals” or “SST = SSE + SSR”. From this decomposition, we define the Goodness-of-Fit measure

$$R^2 = 1 - \frac{SSR}{SST}. \quad (5.10)$$

The  $R^2$  has the interpretation as the *proportion* of variation in  $Y_i$  that is accounted<sup>2</sup> for by  $\hat{Y}_i$ , or by  $X_i$ , since  $\hat{Y}_i$  is just a linear function of  $X_i$ .

Many of these properties require that the intercept term be included in the regression.

**Example 5.2.** Suppose the PRF is

$$E[Y|X] = 2 + 1.5X$$

This is plotted as a blue dashed line on the left panel in Fig. 5.2. You do not observe this line. All you observe are the data points shown as black dots.

```
# Simulated Data
set.seed(888)                                # For replicability
X <- rnorm(10, mean=5, sd=2)                 # Simulating some data
Y <- 2 + 1.5*X + rnorm(10, mean=0, sd=3)
df <- data.frame(X, Y)
# Fit OLS and get predicted values
Ybar <- mean(df$Y); Xbar <- mean(df$X)
b1hat <- cov(df$X, df$Y)/var(df$X)
b0hat <- Ybar - b1hat*Xbar
df <- df %>%
  mutate(Yhat = b0hat + b1hat*X, # add fitted values to df data frame
         ehat = Y - Yhat) %>%
  arrange(X)
# Plot data and lines
p1 <- df %>% ggplot(aes(x=X,y=Y)) +
  geom_point(size=1.5) + geom_line(aes(x=X, y=Yhat), size=0.6) +
  theme_minimal() + ylab(TeX('Y, $\hat{Y}$')) + theme(aspect.ratio = 0.8) +
  geom_abline(intercept=2, slope=1.5, col='blue', lty='dashed', lwd=0.6) +
  geom_segment(x=df$X, xend=df$X, y=df$Y, yend=df$Yhat, lty='dotted') +
  annotate(geom="text", x=4.5, y=11.5,
          label=TeX("$E[Y|X]=\beta_0 + \beta_1 X$"), col="blue")
p1 + gridExtra::tableGrob(round(df[,c("X","Y")],4), theme=ttheme_minimal()) +
  plot_layout(widths=c(2.5,1))
```

<sup>2</sup>Sometimes the word “explained” is used instead.



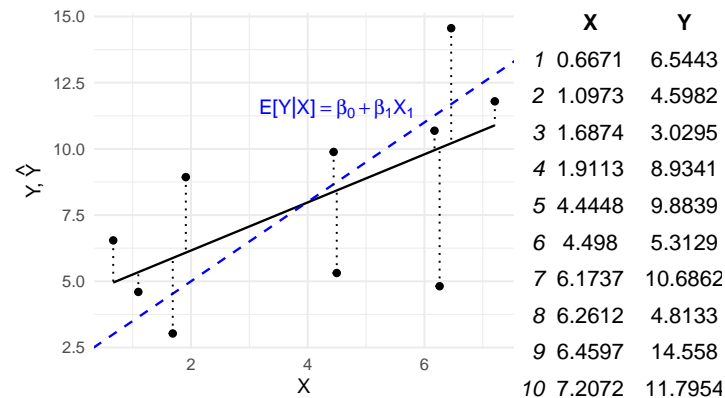


Figure 5.2: Simple Linear Regression

The sample regression function is shown in black, obtained by ordinary least squares. The vertical distances from the data points to the estimated line are shown as dashed line. OLS chooses the black line to minimize the sum of squared lengths of these dotted lines. The parameter estimates are

```
print(c("b0hat" = b0hat, "b1hat" = b1hat))

b0hat    b1hat
4.3473024 0.9078149
```

The residuals have zero sample mean and are uncorrelated with the regressors:

```
## results will be up to computer precision
cat("sample mean of residuals: ", mean(df$ehat), "\n")
cat("sample covariance, residuals and regressor: ", cov(df$ehat, df$X), "\n")

sample mean of residuals: 0
sample covariance, residuals and regressor: 2.607868e-16
```

If you plot the point  $(\bar{X}, \bar{Y})$  in the figure, you will find that it lies on the estimated line (we did not do this in the figure). The  $R^2$  for this regression is

```
SSR <- sum(df$ehat^2) # we defined SSR
SST <- sum((df$Y - mean(df$Y))^2)
Rsqr <- 1 - SSR/SST
Rsqr

[1] 0.3687383
```

In Example 5.2, the estimated regression line does not coincide with the true population regression line. Of course, this will be the case in general, since the sample regression line is only an estimate of the population regression line. The question is how the OLS procedure described here will perform on average in situations where the circumstances described in Assumption Set A hold. In the next section, we argue that, from a certain perspective, you cannot do much better than OLS.

### 5.2.1 Statistical Properties of OLS Estimators

It turns out that OLS produces unbiased and efficient estimators of  $\beta_0$  and  $\beta_1$ , and therefore of  $E[Y|X]$ , under Assumption Set A. In the arguments below we focus on  $\beta_1$ ; similar arguments hold for  $\beta_0$ . We first note that  $\hat{\beta}_1$  can be written as

$$\begin{aligned}
 \hat{\beta}_1 &= \frac{\sum_{i=1}^N (X_i - \bar{X})Y_i}{\sum_{i=1}^N (X_i - \bar{X})^2} \\
 &= \sum_{i=1}^N w_i Y_i \quad \text{where} \quad w_i = \frac{(X_i - \bar{X})}{\sum_{i=1}^N (X_i - \bar{X})^2} \\
 &= \sum_{i=1}^N w_i (\beta_0 + \beta_1 X_i + \epsilon_i) \\
 &= \beta_0 \sum_{i=1}^N w_i + \beta_1 \sum_{i=1}^N w_i X_i + \sum_{i=1}^N w_i \epsilon_i \\
 &= \beta_1 + \sum_{i=1}^N w_i \epsilon_i.
 \end{aligned} \tag{5.11}$$

The second line says that  $\hat{\beta}_1$  is a “linear estimator”. The last line uses the fact that

$$\begin{aligned}
 \sum_{i=1}^N w_i &= \sum_{i=1}^N \left\{ \frac{(X_i - \bar{X})}{\sum_{i=1}^N (X_i - \bar{X})^2} \right\} = \frac{\sum_{i=1}^N (X_i - \bar{X})}{\sum_{i=1}^N (X_i - \bar{X})^2} = 0 \\
 \sum_{i=1}^N w_i X_i &= \sum_{i=1}^N \left\{ \frac{(X_i - \bar{X})X_i}{\sum_{i=1}^N (X_i - \bar{X})^2} \right\} = \frac{\sum_{i=1}^N (X_i - \bar{X})^2}{\sum_{i=1}^N (X_i - \bar{X})^2} = 1.
 \end{aligned}$$

The form of  $\hat{\beta}_1$  in Eq. 5.11 is useful because it expresses  $\hat{\beta}_1$  in terms of  $\beta_1$  which enables a comparison of the two.

Assumption Set A implies that our data sample satisfies  $E[\epsilon_i|X_i] = 0$  and  $E[\epsilon_i^2|X_i] = \sigma^2$ . Independence of our sample observations allows us to extend these statements to

$$E[\epsilon_i|X_1, X_2, \dots, X_N] = 0 \text{ for all } i = 1, 2, \dots, N, \tag{5.12}$$

and

$$E[\epsilon_i^2|X_1, X_2, \dots, X_N] = \sigma^2 \text{ for all } i = 1, 2, \dots, N. \tag{5.13}$$

Furthermore, we also have

$$E[\epsilon_i \epsilon_j|X_1, X_2, \dots, X_N] = 0 \text{ for all } i = 1, 2, \dots, N. \tag{5.14}$$

We will use Eq. 5.12 - Eq. 5.14 in the derivation of the properties of OLS estimators.

*Unbiasedness* Under Assumption Set A,  $\hat{\beta}_1$  is unbiased, i.e.,  $E[\hat{\beta}_1] = \beta_1$ . *Proof:* From Eq. 5.11 we get

$$E[\hat{\beta}_1|X_1, X_2, \dots, X_N] = \beta_1 + \sum_{i=1}^N w_i E[\epsilon_i|X_1, X_2, \dots, X_N] = \beta_1. \tag{5.15}$$

Since the conditional expectation is the constant  $\beta_1$ , the unconditional mean is also  $\beta_1$ .

Unbiasedness of  $\hat{\beta}_1$  means that  $\hat{\beta}_1$  does not *systematically* underestimate or overestimate  $\beta_1$ . Of course, in any given application there will be sampling error. For instance, we clearly underestimated the slope in Example 5.3, although in a non-simulated application you would not know this since you do not observe the PRF. Unbiasedness of  $\hat{\beta}_1$  means that in repeated application under similar circumstances,  $\hat{\beta}_1$  will estimate  $\beta_1$  correctly “on average”.

**Example 5.3.** We replicate the simulation exercise in Example 5.2 two hundred times. For each replication, we collect the estimated  $\beta_1$ .

```
set.seed(888)
nreps <- 200
betas1 <- rep(NA, nreps)
X <- rnorm(10, mean=5, sd=2)
for (i in 1:nreps) {
  Y <- 2 + 1.5*X + rnorm(10, mean=0, sd=3)
  dfsim <- data.frame(X, Y)
  mdlsim <- lm(Y~X, data=dfsim)
  betas1[i] <- coef(summary(mdlsim))[2, "Estimate"]
}
cat("The mean of the simulated betahat estimates is", round(mean(betas1), 3), "\n")
```

The mean of the simulated betahat estimates is 1.492

The average  $\hat{\beta}_1$  obtained over the 200 replications is approximately 1.492 which is quite close to the true value of 1.5. In practice, of course, you only have one estimate, that for the data sample that you have. Nonetheless, this simulation exercise illustrates the fact the  $\hat{\beta}_1$  is unbiased.<sup>3</sup>

Notice that the proof of unbiasedness of  $\hat{\beta}_1$  uses the condition in Eq. 5.12, and one of the key assumptions in Assumption Set A underlying this condition is Assumption A2  $E[\epsilon|X] = 0$ . The estimator  $\hat{\beta}_1$  will be biased if this assumption is violated. We will see examples where this assumption fails to hold. The proof of unbiasedness, on the other hand, did not make use of assumption A3 (conditional homoskedasticity) in any way, which means that violation of A3 will not lead to bias in  $\hat{\beta}_1$ .

We would like to characterize the precision with which we are able to estimate  $\beta_1$ . The conditional variance of  $\hat{\beta}_1$  under Assumption Set A is

$$\begin{aligned} \text{var}[\hat{\beta}_1|X_1, X_2, \dots, X_N] &= \text{var}\left[\beta_1 + \sum_{i=1}^N w_i \epsilon_i \middle| X_1, X_2, \dots, X_N\right] \\ &= \sum_{i=1}^N w_i^2 \text{var}[\epsilon_i|X_1, X_2, \dots, X_N] \\ &= \sigma^2 \sum_{i=1}^N w_i^2 \\ &= \frac{\sigma^2}{\sum_{i=1}^N (X_i - \bar{X})^2} \end{aligned} \tag{5.16}$$

<sup>3</sup>Our simulation experiment actually illustrates “conditional unbiasedness” as in Eq. 5.15. To show “unconditional unbiasedness” in the simulation experiment, move the line `X <- rnorm(10, mean=5, sd=2)` from just above the `for` statement to just below it (i.e., to just above `Y <- 2 + 1.5*X + rnorm(10, mean=0, sd=3)`).

The derivation of  $\text{var}[\hat{\beta}_1]$  makes use of the conditional homoskedasticity assumption, and the assumption that the error terms are uncorrelated (which came from the random sampling assumption). If these assumptions do not hold (in fact, if any of the assumptions do not hold), then the formula derived in Eq. 5.16 will be incorrect. The variance derived in Eq. 5.16 is a conditional variance. Assumption Set A does not give us enough information to calculate the unconditional variance of  $\hat{\beta}_1$ .

In order to put a number on  $\text{var}[\beta_1|X_1, \dots, X_N]$ , we will need an estimate of  $\sigma^2$ . An unbiased estimator for  $\sigma^2$  in the simple linear regression model (unbiased under Assumption Set A) is

$$\hat{\sigma}^2 = \frac{1}{N-2} \sum_{i=1}^N \hat{\epsilon}_i^2. \quad (5.17)$$

We shall leave the proof of unbiasedness of  $\hat{\sigma}^2$  until later. For now, we simply note that the SSR is divided by  $N-2$  instead of  $N$  because two ‘degrees-of-freedom’ were used in computing  $\hat{\beta}_0$  and  $\hat{\beta}_1$ , and these are used in the computation of  $\hat{\epsilon}_i$ . With  $\hat{\sigma}^2$ , we can estimate the conditional variance of  $\hat{\beta}_1$  with

$$\widehat{\text{var}}[\hat{\beta}_1|X_1, \dots, X_N] = \frac{\hat{\sigma}^2}{\sum_{i=1}^N (X_i - \bar{X})^2} \quad (5.18)$$

The standard error of  $\hat{\beta}_1$  is the square root of Eq. 5.18. For the data in Example 5.2:

```
sigma2hat <- SSR/(10-2) # SSR calculated earlier
cat("residual variance: ", round(c(sigma2hat),3), "\n")
cat("residual s.e.: ", round(sqrt(sigma2hat),3), "\n")
estvarb1hat <- sigma2hat/sum((df$X-mean(df$X))^2)
cat("b1hat variance: ", round(c(estvarb1hat),3), "\n")
cat("b1hat s.e.: ", round(sqrt(estvarb1hat),3), "\n")
```

```
residual variance:  9.849
residual s.e.:    3.138
b1hat variance:   0.176
b1hat s.e.:      0.42
```

Since the data in Example 5.2 is simulated with  $\sigma^2 = 9$ , the true conditional variance and standard error of  $\hat{\beta}_1$  are

```
var_betahat <- 9/sum((df$X - mean(df$X))^2)
cat("True b1hat variance: ", round(var_betahat,3), "\n")
cat("True b1hat s.e.: ", round(sqrt(var_betahat),3), "\n")
```

```
True b1hat variance:  0.161
True b1hat s.e.:    0.401
```

In our simulation exercise in Example 5.3, the standard deviation of the  $\hat{\beta}_1$  obtained over the 200 replications is 0.395 which is very close to 0.401.

```
cat("Standard deviation of simulated betahats is", round(sd(betahats),3), "\n")
```

```
Standard deviation of simulated betahats is 0.395
```

The formula for  $\text{var}[\hat{\beta}_1|X_1, X_2, \dots, X_N]$  tells us that the estimator for  $\hat{\beta}_1$  is more precise (has smaller variance) when (i)  $\sigma^2$  is smaller, (ii)  $N$  is larger (since the denominator is a sum of  $N$  non-negative terms), and (iii) if there is more variation in  $X_i$ . This is intuitive; you should get more precise estimators if (i) the data are less noisy, (ii) you have more observations, and if (iii) there is more variation in  $X_i$ . When  $\sigma^2$  is smaller, your data is more informative about the PRF. If you have more observations, you will be able to estimate your parameters more precisely. Since  $\beta_1$  measures how  $Y_i$  changes as  $X_i$  changes, it helps if  $X_i$  changes a lot in your sample.

*Efficiency* OLS estimators, under Assumption Set A, are efficient in the sense that they have the lowest variance among all linear unbiased estimators. We show this for  $\hat{\beta}_1$ . Let  $\tilde{\beta}_1$  be another estimator of the form  $\tilde{\beta}_1 = \sum_{i=1}^N a_i Y_i$  such that  $E[\tilde{\beta}_1] = \beta_1$  and where  $a_i$  are weights constructed from  $\{X_i\}_{i=1}^N$ . We want to relate this estimator to  $\hat{\beta}_1$  so write  $\tilde{\beta}_1$  as

$$\begin{aligned}\tilde{\beta}_1 &= \sum_{i=1}^N (w_i + v_i) Y_i \\ &= \sum_{i=1}^N w_i Y_i + \sum_{i=1}^N v_i Y_i \\ &= \beta_1 + \sum_{i=1}^N w_i \epsilon_i + \sum_{i=1}^N v_i (\beta_0 + \beta_1 X_i + \epsilon_i) \\ &= \beta_1 + \beta_0 \sum_{i=1}^N v_i + \beta_1 \sum_{i=1}^N v_i X_i + \sum_{i=1}^N (w_i + v_i) \epsilon_i\end{aligned}\tag{5.19}$$

where  $w_i$  are the OLS weights previously defined. We want  $\tilde{\beta}_1$  to be unbiased, so we limit our choice of  $v_i$  to those such that  $\sum_{i=1}^N v_i = 0$  and  $\sum_{i=1}^N X_i v_i = 0$ , which guarantees unbiasedness of  $\tilde{\beta}_1$ . Then Eq. 5.19 becomes

$$\tilde{\beta}_1 = \beta_1 + \sum_{i=1}^N (w_i + v_i) \epsilon_i.$$

Taking conditional variance gives

$$\begin{aligned}\text{var}[\tilde{\beta}_1|X_1, \dots, X_N] &= \sum_{i=1}^N (w_i + v_i)^2 \text{var}[\epsilon_i|X_1, \dots, X_N] \\ &= \sigma^2 \sum_{i=1}^N (w_i^2 + v_i^2 + 2w_i v_i) \\ &= \text{var}[\hat{\beta}_1] + \sigma^2 \sum_{i=1}^N v_i^2\end{aligned}\tag{5.20}$$

since  $\text{var}[\hat{\beta}_1] = \sigma^2 \sum_{i=1}^N w_i^2$  and  $\sum_{i=1}^N w_i v_i = 0$  (why?)

In other words, you will not be able to find another linear estimator for  $\beta_1$  with smaller variance than  $\hat{\beta}_1$ . We summarize the unbiasedness and minimum variance result by saying that  $\hat{\beta}_1$  is a “Best Linear Unbiased Estimator”, or BLUE. The result is also referred to as the “Gauss-Markov Theorem”. The result applies also to  $\hat{\beta}_0$  (and to the multiple regression case). We present the general result later.

We have so far only presented results for  $\hat{\beta}_1$ . What about  $\hat{\beta}_0$ . It remains true that  $\hat{\beta}_0$  is unbiased, i.e.,  $E[\hat{\beta}_0] = \beta_0$  under Assumption Set A, and that this property does not require conditional homoskedasticity. Furthermore, we have

$$\text{var}[\hat{\beta}_0|X_1, \dots, X_N] = \frac{\sigma^2 \sum_{i=1}^N X_i^2}{N \sum_{i=1}^N (X_i - \bar{X})^2}$$

and

$$\text{cov}[\hat{\beta}_0, \hat{\beta}_1|X_1, \dots, X_N] = \frac{-\sigma^2 \bar{X}}{\sum_{i=1}^N (X_i - \bar{X})^2}.$$

We will derive these results later. The sign of the correlation (which depends on  $\bar{X}$ ) is intuitive, since the estimated regression line always passes through the point  $(\bar{X}, \bar{Y})$ .

### 5.3 Prediction

Since  $\hat{\beta}_0$  and  $\hat{\beta}_1$  are unbiased, predictions based on OLS estimators will also be (conditionally) unbiased. Suppose there is a new independent observation  $(Y_0, X_0)$  from the same population, so

$$Y_0 = \beta_0 + \beta_1 X_0 + \epsilon_0$$

with  $E[\epsilon_0|X_0] = 0$ . You only observe  $X_0$ , and you predict  $Y_0$  with  $\hat{Y}_0 = \hat{\beta}_0 + \hat{\beta}_1 X_0$ . Then

$$E[\hat{\beta}_0 + \hat{\beta}_1 X_0|X_0] = E[\hat{\beta}_0|X_0] + E[\hat{\beta}_1|X_0]X_0 = \beta_0 + \beta_1 X_0 = E[Y_0|X_0].$$

(The expectation is with respect to your estimation sample).

Furthermore, the MSPE will

$$\begin{aligned} E[(Y_0 - \hat{Y}_0)^2|X_0] &= E[(\epsilon_0 + (\beta_0 - \hat{\beta}_0) + (\beta_1 - \hat{\beta}_1)X_0)^2|X_0] \\ &= E[\epsilon_0^2|X_0] + E[(\beta_0 - \hat{\beta}_0) + (\beta_1 - \hat{\beta}_1)X_0]^2|X_0 \end{aligned}$$

The second line comes from the assumption that the new observation is independent of your sample. Since  $E[\epsilon_0|X_0] = 0$ , the first term in the second line is just the conditional variance of  $\epsilon_0$  which is  $\sigma^2$ . Likewise, since

$$E[(\beta_0 - \hat{\beta}_0) + (\beta_1 - \hat{\beta}_1)X_0|X_0] = 0$$

the second term in the second line is also just the variance of  $(\beta_0 - \hat{\beta}_0) + (\beta_1 - \hat{\beta}_1)X_0$ , which is

$$\text{var}[\hat{\beta}_0] + X_0^2 \text{var}[\hat{\beta}_1] + 2X_0 \text{cov}[\hat{\beta}_0, \hat{\beta}_1] = \left( \frac{1}{N} + \frac{(X_0 - \bar{X})^2}{\sum_{i=1}^N (X_i - \bar{X})^2} \right) \sigma^2 \quad (5.21)$$

(See exercises.) The root mean square prediction error is therefore

$$\text{RMPSE} = \left\{ \sigma^2 \left[ 1 + \frac{1}{N} + \frac{(X_0 - \bar{X})^2}{\sum_{i=1}^N (X_i - \bar{X})^2} \right] \right\}^{1/2}. \quad (5.22)$$

Some remarks:

- The “1” in the square brackets of Eq. 5.22 reflects the variance of the unpredictable element of the new observation. The other two terms reflects the mean squared error in estimating the conditional mean. In general mean squared error is variance plus squared bias, but because the estimators are unbiased, the mean squared estimation error is simply the variance. As  $N$  increases, the sampling error part gets smaller ( $1/N$  is smaller for larger  $N$ , and the denominator in the third term is the sum of  $N$  non-negative terms).
- Notice that the RMSPE gets larger the further  $X_0$  is from the sample mean of the predictor.
- Related to the previous point, the formula in Eq. 5.22 *assumes* correct specification of the conditional mean in the regression model.
- The predictions are usually reported as “prediction  $\pm 1.96$  RMSPE for a”0.95 prediction interval”, i.e., the 0.95 prediction interval is

$$\text{prediction} \pm 1.96 \left\{ \sigma^2 \left[ 1 + \frac{1}{N} + \frac{(X_0 - \bar{X})^2}{\sum_{i=1}^N (X_i - \bar{X})^2} \right] \right\}^{1/2}.$$

- Formula Eq. 5.22 gives the RMSPE for predictions of “new” observations not in the estimation sample, and takes into account (i) the sampling error when estimating the conditional mean, and (ii) the fact that our new prediction will include an unpredictable noise term with variance  $\sigma^2$ .
- If we wish to measure the 0.95 *confidence interval* around the estimated conditional mean, we would use Eq. 5.22 but without the “+1”, i.e., the 0.95 confidence interval is

$$\text{prediction} \pm 1.96 \left\{ \sigma^2 \left[ \frac{1}{N} + \frac{(X_0 - \bar{X})^2}{\sum_{i=1}^N (X_i - \bar{X})^2} \right] \right\}^{1/2}.$$

- Of course, we have to replace  $\sigma^2$  with an estimate of it.

**Example 5.4.** Fig. 5.3 shows a scatter-plot of 540 observations of log hourly earnings `log(earnings)` against years of schooling `s`, taken from the dataset `earnings.csv`. Years of schooling ranges from 7 years to 20 years. We estimate the simple linear regression model

$$\log(\text{earnings}_i) = \beta_0 + \beta_1 s_i + \epsilon_i$$

and use it to predict the `log(earnings)` of new observations at various levels of schooling `s`. The scatterplot includes the estimated regression line (i.e., the predictions), as well as the  $\pm 1.96$  RMSPE (or “prediction standard error”) band, with RMSPE calculated as in Eq. 5.22. We use the `predict.lm` function’s built-in capability to calculate the predictions and RMSPE instead of calculating it from scratch, although you are encouraged to try to replicate the result. The predictions and prediction bands can be obtained from `mdl_pred`. We also plot the 0.95 confidence intervals in red. Because we have a fairly large sample, the sampling error in estimating the conditional expectation is quite small, resulting in the very tight confidence interval band. However, years of schooling only explains fairly little on the variation in `log(earnings)`, so the prediction interval is quite large.

```
df_earn <- read_csv("data\\earnings.csv", show_col_types = FALSE) %>%
  select(c(earnings, s))
mdl_earn <- lm(data=df_earn, log(earnings)~s)
# We will predict log(earnings) at s = 5, 6, ..., 22, compiled in "new_data" below
new_data <- data.frame(s=seq(5,22,1))
mdl_pred <- predict(mdl_earn, new_data, interval = "prediction", level = 0.95)
mdl_pred <- cbind(new_data, mdl_pred)
mdl_ci <- predict(mdl_earn, new_data, interval = "confidence", level = 0.95)
mdl_ci <- cbind(new_data, mdl_ci)

ggplot() +
  geom_point(data=df_earn,aes(x=s, y=log(earnings)), size=1) +
  geom_line(data=mdl_pred, aes(x=s, y=fit), col="blue") +
  geom_line(data=mdl_pred, aes(x=s, y=upr), linetype="dotted", col="blue") +
  geom_line(data=mdl_pred, aes(x=s, y=lwr), linetype="dotted", col="blue") +
  geom_line(data=mdl_ci, aes(x=s, y=upr), linetype="dotted", col="red") +
  geom_line(data=mdl_ci, aes(x=s, y=lwr), linetype="dotted", col="red") +
  theme_minimal()
```

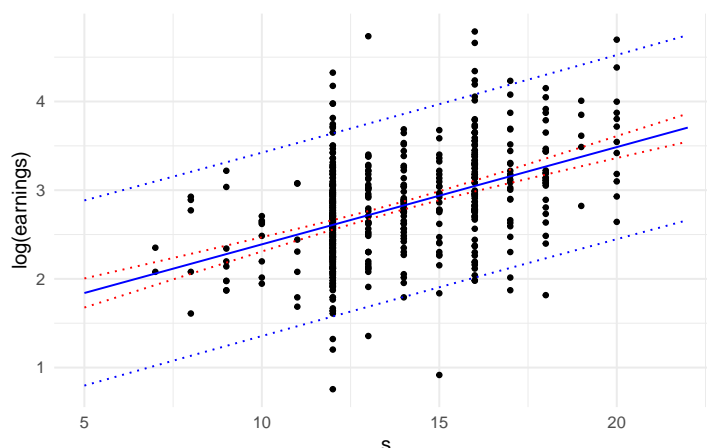
Figure 5.3:  $\log(\text{earnings})$  vs  $s$ 

Fig. 5.3 presents several interesting questions. Suppose we were interested in predicting a new observation at  $s = 16$ . In our data set we have several (in fact 88) observations at  $s = 16$ . Would it make sense to simply use the sample mean of those 88 observations as our prediction? You would still get unbiased estimators, although your prediction would be based on only 88 observations, and in situations where you have very few observations at a certain  $s$ , your prediction and estimates of the MSPE can be unreliable. On the other hand, if indeed the conditional expectation of  $\log(\text{earnings})$  is a linear function of  $s$ , then we can use the entire data set to estimate just two parameters which would reduce sampling errors considerably.

Nonetheless, using just the observations at  $s=16$  might actually be a sensible thing to do if we were very unsure about the form of the conditional expectation. If the conditional expectation is non-linear, then the linear regression line would only be an approximation (possibly a very poor one) and would give biased predictions. The sample mean at  $s=16$  would be unbiased, though it would have greater variance.



What we have is a bias-variance trade-off. If the conditional expectation is non-linear, but only slightly, then imposing the assumption would result in slightly biased predictions, but we would be able to draw on the whole data set to reduce sampling error. It may be the case that allowing for a slight bias might reduce variance sufficiently to reduce mean squared error, which is variance plus squared bias.

Another question: Why are we predicting  $\log(\text{earnings})$  and not  $\text{earnings}$ ? Probably it is the latter that we want, but the relationship between  $\text{earnings}$  and  $s$  is definitely not linear, see Fig. 5.4

```
ggplot() +
  geom_point(data=df_earn,aes(x=s, y=earnings), size=1)
```

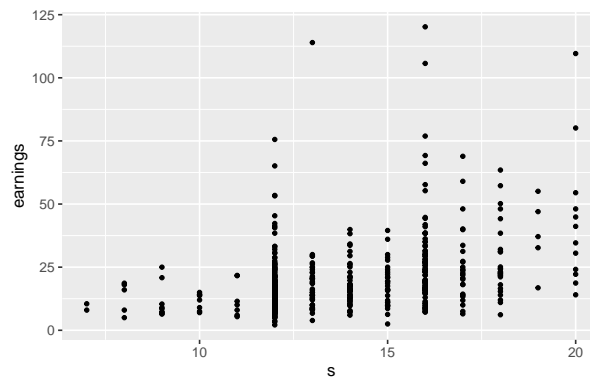


Figure 5.4:  $\log(\text{earnings})$  vs  $s$

We may be able to effectively model such non-linearity using a multiple linear regression framework, but often a simple transformation of one or more of the variables works well to linearize a relationship. As we see in Fig. 5.3, a linear relationship between  $\log(\text{earnings})$  and  $s$  is not entirely unreasonable, and using  $\log(\text{earnings})$  also alleviates some of the heteroskedasticity issues that we see in Fig. 5.4.

But then, how do we convert a prediction of  $\log Y$  to a prediction of  $Y$ ? One way would be to simply reverse the  $\log()$  transformation, and use

$$\exp(\widehat{\log Y})$$

as the prediction. However,  $\widehat{\log y}$  is an estimate of  $E[\log Y|X]$ , and because  $\exp()$  is a convex function,

$$\exp E[\log Y|X] \leq E[\exp \log Y|X]$$

In other words we would be systematically under-estimating  $E[\log Y|X]$ .

We know that if  $\log Y \sim \text{Normal}(\mu, \sigma^2)$ , then  $Y$  has the log-normal distribution with mean

$$E[Y] = \exp(\mu) \exp(\sigma^2/2).$$

If we assume normality of the error terms in the log-regression, i.e.,

$$\log Y = \beta_0 + \beta_1 X + \epsilon, \quad \epsilon \sim \text{Normal}(0, \sigma^2),$$

then

$$(\log Y)|X \sim \text{Normal}(\beta_0 + \beta_1 X, \sigma^2)$$

which means

$$E[Y|X] = \exp(\beta_0 + \beta_1 X) \exp(\sigma^2/2).$$

This suggests using the transformation

$$\exp(\widehat{\log Y}) \exp\left(\frac{\widehat{\sigma^2}}{2}\right)$$

to convert predictions of  $\log Y$  to predictions of  $Y$ .

## 5.4 Hypothesis Testing

We often want to test if  $\beta_1$  is equal to some value in population. For instance, is the price elasticity of a product equal to 1? This would be the hypothesis  $H_0 : \beta_1 = 1$  vs  $H_A : \beta_1 \neq 1$  in the regression

$$\ln Y_i = \beta_0 + \beta_1 \ln X_i + \epsilon_i$$

where  $X$  is the price and  $Y$  is quantity sold. We call  $H_0$  the “null hypothesis” and  $H_A$  the “alternative hypothesis”. Is a job training program effective? This would be the hypothesis  $H_0 : \beta_1 = 0$  vs  $H_A : \beta_1 \neq 0$  in the regression

$$Y = \beta_0 + \beta_1 X + \epsilon_i.$$

where  $X$  is an indicator of participation in a job training program and  $Y$  is some outcome variable.

The basic strategy for checking if  $\beta_1$  is equal to some value  $\beta_1^*$  in population is to check if  $\hat{\beta}_1$  is “improbably far” from  $\beta_1^*$ , given what we know about the distribution of  $\hat{\beta}_1$  “under the null”, i.e., when  $\beta_1 = \beta_1^*$ . In order to derive this distribution, at least in finite samples, we have to make an additional assumption regarding the conditional distribution of  $\epsilon$ . If we assume that

$$(A6) \epsilon|X \sim \text{Normal}(0, \sigma^2)$$

then under random sampling, we will have

$$\epsilon_i|X_1, X_2, \dots, X_N \sim \text{Normal}(0, \sigma^2).$$

Under the null hypothesis we have

$$\hat{\beta}_1 = \beta_1^* + \sum_{i=1}^N w_i \epsilon_i.$$

Since  $\hat{\beta}_1$  is a constant plus a linear combination of normally distributed terms, it is normally distributed (conditional on  $\{X_1, X_2, \dots, X_N\}$ ). We have already shown that  $E[\hat{\beta}_1|X_1, \dots, X_N]$  is unbiased and  $\text{var}[\hat{\beta}_1|X_1, \dots, X_N] = \sigma^2 / \sum_{i=1}^N (X_i - \bar{X})^2$ . Therefore under the null hypothesis

that  $\beta_1 = \beta_1^*$  we have

$$\hat{\beta}_1 | X_1, \dots, X_N \sim \text{Normal} \left( \beta_1^*, \frac{\sigma^2}{\sum_{i=1}^N (X_i - \bar{X})^2} \right).$$

That is,

$$\frac{\hat{\beta}_1 - \beta_1^*}{\sqrt{\frac{\sigma^2}{\sum_{i=1}^N (X_i - \bar{X})^2}}} \bigg| X_1, \dots, X_N \sim \text{Normal}(0, 1).$$

The remaining problem is that  $\sigma^2$  is unknown. It turns out that replacing  $\sigma^2$  with  $\widehat{\sigma}^2$  as calculated in Eq. 5.17, we have

$$t = \frac{\hat{\beta}_1 - \beta_1^*}{\sqrt{\frac{\widehat{\sigma}^2}{\sum_{i=1}^N (X_i - \bar{X})^2}}} \sim t_{(N-2)}. \quad (5.23)$$

We can drop the conditioning information, since the t-distribution does not depend on  $X_1, \dots, X_N$ . In other words, the result holds unconditionally.

We can then use Eq. 5.23 to test  $H_0 : \beta_1 = \beta_1^*$  in the usual way. For instance, for a 0.05 test, we can use the rule “reject  $H_0 : \beta_1 = \beta_1^*$  if the absolute value of  $t$  in Eq. 5.23 exceeds  $t_{0.025, N-2}$  where  $t_{0.025, N-2}$  is the 0.975 percentile of the  $t_{N-2}$  distribution. The denominator is just the square root of the usual estimator of  $\text{var}[\hat{\beta}_1]$ . If we are testing if  $\beta_1 = 0$ , then Eq. 5.23 is just the parameter estimate divided by its standard error.

**Example 5.5.** We continue with Example 5.2, where we fit the regression  $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$  on a simulated dataset with  $N = 10$  observations, and obtained the estimates

```
cat("beta1hat = ", round(b1hat,3), "\n")
cat("s.e.(beta1hat) = ", round(sqrt(estvarb1hat),3), "\n")
```

```
beta1hat = 0.908
s.e.(beta1hat) = 0.42
```

Suppose we are interested in testing the hypothesis  $H_0 : \beta_1 = 0$  vs the usual two-sided alternative, then the appropriate t-statistic is

```
t1 = b1hat / sqrt(estvarb1hat)
cat("t-stat(beta1=0) = ", round(t1,3))
```

```
t-stat(beta1=0) = 2.162
```

The critical value for the test at 0.01, 0.05 and 0.10 levels of significance is

```
round(c("0.10" = qt(0.95, 8), "0.05" = qt(0.975, 8), "0.01" = qt(0.995, 8)), 3)
```

```
0.10 0.05 0.01
1.860 2.306 3.355
```

We reject the hypothesis at 0.10 significance level, but not at 0.01 and 0.05 significance levels.

Alternatively, we can compute the p-value for this test.

```
cat("p-val (b1=0) is: ", round((1-pt(abs(t1), 8))*2,4), "\n")
```

```
p-val (b1=0) is: 0.0626
```

To test  $H_0 : \beta_1 = 1$  vs  $H_A : \beta_1 \neq 1$ , the t-statistic is

```
t2 = (b1hat-1) / sqrt(estvarb1hat)
cat("t-stat(beta1=1) = ", round(t2,3), "with p-value: ", round((1-pt(abs(t2), 8))*2,4), "\n")
```

```
t-stat(beta1=1) = -0.22 with p-value: 0.8317
```

We do not reject this test at any of the usual significance levels.

Most of the calculations here can be obtained via R's `lm()` function:

```
mdl <- lm(Y~X, data=df)
mdl_sum <- summary(mdl)
print(mdl_sum)
```

Call:

```
lm(formula = Y ~ X, data = df)
```

Residuals:

Min	1Q	Median	3Q	Max
-5.2180	-2.3236	0.8198	1.5689	4.3465

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	4.3473	1.9658	2.211	0.0580 .
X	0.9078	0.4200	2.162	0.0626 .

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.138 on 8 degrees of freedom

Multiple R-squared: 0.3687, Adjusted R-squared: 0.2898

F-statistic: 4.673 on 1 and 8 DF, p-value: 0.06262

The t-values reported are just the estimate divided by the standard error, and can be used to test (separately) the hypotheses  $\beta_0 = 0$  and  $\beta_1 = 0$ . The p-values indicate that we reject these hypotheses at 10%, but not at 5%. Since this is simulated data with both parameters set at  $\beta_0 = 4$  and  $\beta_1 = 1.5$ , we know that we make the wrong inference (we do not reject a false hypothesis) with the 0.05 and 0.01 tests. There is always the possibility of non-rejection of a false hypothesis, but in this case, this is due to the small sample size and relatively noisy data, leading to relatively large standard errors, and small t-values. For the test  $H_0 : \beta_1 = 1$  vs  $H_A : \beta_1 \neq 1$  you can use the `linearHypothesis()` function from the `car` package. The test reported is an “F-test” and not a t-test, although for tests of a single restriction they are equivalent (notice that the p-value is identical to the t-test reported previously for this hypothesis). We will discuss the F-test in a later chapter.

```
car::linearHypothesis(mdl, "X=1")
```

Linear hypothesis test

Hypothesis:

X = 1

Model 1: restricted model

Model 2: Y ~ X

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	9	79.269				
2	8	78.794	1	0.4746	0.0482	0.8317

## 5.5 Asymptotic Results

The results obtained so far are “finite sample” results, i.e., are valid for any given sample size  $N$  (when Assumption Set A holds). The OLS estimators also have good properties in the limit as  $N \rightarrow \infty$ , or “asymptotically”. We continue to focus of  $\hat{\beta}_1$ .

*Consistency*  $\hat{\beta}_1 \rightarrow_p \beta_1$ .

The starting point for showing this result is Eq. 5.11 which we restate here, expanding  $w_i$  and dividing the numerator and denominator of the last term by  $1/N$ :

$$\begin{aligned}\hat{\beta}_1 &= \beta_1 + \sum_{i=1}^N w_i \epsilon_i \\ &= \beta_1 + \frac{(1/N) \sum_{i=1}^N (X_i - \bar{X}) \epsilon_i}{(1/N) \sum_{i=1}^N (X_i - \bar{X})^2}.\end{aligned}\tag{5.24}$$

The last term is the ratio of sample covariance of  $X_i$  and  $\epsilon_i$  to the sample variance of  $X_i$ . If we assume that the conditions are such that the sample covariance of  $X_i$  and  $\epsilon_i$  converges in probability to the population covariance of  $X$  and  $\epsilon$ , and likewise that the sample variance of  $X_i$  converges in probability to the population variance of  $X$ , and that the population variance of  $X$  is not zero, then

$$\hat{\beta}_1 = \beta_1 + \frac{\overbrace{\frac{(1/N) \sum_{i=1}^N (X_i - \bar{X}) \epsilon_i}{(1/N) \sum_{i=1}^N (X_i - \bar{X})^2}}^{\rightarrow_p \text{cov}[X, \epsilon]=0}}{\rightarrow_p \text{var}[X] \neq 0} \rightarrow_p \beta_1\tag{5.25}$$

The assumption that  $\text{cov}[X, \epsilon] = 0$  comes directly from the assumption that  $E[\epsilon|X] = 0$ . Earlier we assumed that  $\sum_{i=1}^N (X_i - \bar{X})^2 \neq 0$ . We modify this assumption slightly to  $\text{var}[X] \neq 0$  (as noted earlier, the former is implied by the latter and random sampling). Convergence in probability comes about from the laws of large numbers. Earlier we stated the LLN as the convergence in probability of a sample mean to the population mean. The covariance of  $X$  and  $\epsilon$  is a population expectation (of the random variable  $(X - E[X])(\epsilon - E[\epsilon])$ ) and the sample covariance of  $X_i$  and  $\epsilon_i$  is a sample average of observations of this random variable. Likewise, for the sample variance in the denominator. As long as the conditions for the LLN hold for these sample means, convergence in probability will follow.

It can be shown that under a mild extension of Assumption Set A, we have

$$\sqrt{N}(\hat{\beta}_1 - \beta_1) \rightarrow_d \text{Normal}\left(0, \frac{\sigma^2}{\text{var}[X]}\right) \quad (5.26)$$

or

$$\frac{\sqrt{N}(\hat{\beta}_1 - \beta_1)}{\sqrt{\sigma^2/\text{var}[X]}} \rightarrow_d \text{Normal}(0, 1).$$

We can replace  $\sigma^2$  and  $E[X]$  with consistent estimates of the two. Replacing  $\sigma^2$  with  $\hat{\sigma}^2$  and using  $(1/N) \sum_{i=1}^N (X_i - \bar{X})^2$  as a consistent estimator for  $\text{var}[X]$ , we have

$$\frac{\hat{\beta}_1 - \beta_1}{\sqrt{\frac{\hat{\sigma}^2}{\sum_{i=1}^N (X_i - \bar{X})^2}}} \rightarrow_d \text{Normal}(0, 1).$$

The left hand side is just the t-statistic in Eq. 5.23. In other words, we can continue to use the t-statistic, but we choose the rejection region from the standard normal distribution rather than the t-distribution. This test would only be approximate, but if the sample size is large enough it should be sufficiently accurate.

We omit details of the proof of Eq. 5.26, but the intuition is as follows: write

$$\begin{aligned} \hat{\beta}_1 &= \beta_1 + \sum_{i=1}^N w_i \epsilon_i \\ &= \beta_1 + \left[ \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})^2 \right]^{-1} \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X}) \epsilon_i \end{aligned}$$

or

$$\begin{aligned} \sqrt{N}(\hat{\beta}_1 - \beta_1) &= \left[ \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})^2 \right]^{-1} \frac{1}{\sqrt{N}} \sum_{i=1}^N (X_i - \bar{X}) \epsilon_i \\ &= \left[ \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})^2 \right]^{-1} \frac{1}{\sqrt{N}} \sum_{i=1}^N b_i \epsilon_i \end{aligned}$$

where  $b_i = X_i - \bar{X}$ . The random variables  $\{b_i \epsilon_i\}_{i=1}^N$  have mean

$$E[b_i \epsilon_i] = E[E[b_i \epsilon_i | X_1, \dots, X_N]] = E[b_i E[\epsilon_i | X_1, \dots, X_N]] = 0$$

and variance

$$E[(b_i \epsilon_i)^2] = E[b_i^2 E[\epsilon_i^2 | X_1, \dots, X_N]] = \sigma^2 E[b_i^2].$$

If the required conditions for a relevant CLT hold, we have

$$\frac{1}{\sqrt{N}} \sum_{i=1}^N b_i \epsilon_i \rightarrow_d N(0, \sigma^2 E[b_i^2]).$$

Since  $(1/N) \sum_{i=1}^N (X_i - \bar{X})^2 \rightarrow_p \text{var}[X]$ , evidently  $E[b_i^2]$  is  $\text{var}[X]$ . Therefore

$$\begin{aligned} \sqrt{N}(\hat{\beta}_1 - \beta_1) &= \overbrace{\left[ \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})^2 \right]^{-1}}^{\rightarrow_p \text{var}[X]^{-1}} \overbrace{\frac{1}{\sqrt{N}} \sum_{i=1}^N b_i \epsilon_i}^{\rightarrow_d \text{Normal}(0, \sigma^2 \text{var}[X])} \\ &\rightarrow_d \text{Normal}(0, \sigma^2 \text{var}[X]^{-1}) \end{aligned}$$

## 5.6 When Baseline Assumptions are Violated

In summary, OLS estimation behaves well when Assumption Set A holds. What happens if one or more of the assumptions fail to hold?

### 5.6.1 Heteroskedasticity

The assumption of conditional homoskedasticity is that the variance of the noise terms do not depend on the regressors. The proof of unbiasedness (and consistency) of OLS estimators did not use this assumption in any way, which implies that violation of conditional homoskedasticity (such as in Example 5.1) does not affect unbiasedness. However, the derivation of the formulas for  $\text{var}[\hat{\beta}_1]$  and  $\text{var}[\hat{\beta}_0]$  do use this assumption, so the derived formulas for the estimator variances and t-tests are incorrect when conditional homoskedasticity fails to hold. Estimator variances computed using the formulas derived under homoskedasticity when the noise terms are heteroskedastic are unreliable. Furthermore, it turns out that OLS is no longer minimum variance among unbiased linear estimators, i.e., OLS is no longer “efficient”. Discussions about how OLS should be amended in response to conditional heteroskedasticity will have to wait until a later chapter, when we discuss heteroskedasticity in more detail.

### 5.6.2 Endogeneity

Recall that one of the key assumptions for unbiasedness of OLS estimators is that  $E[\epsilon_i | X_1, \dots, X_N] = 0$ . This assumption may fail to hold if  $E[\epsilon | X] = 0$  does not hold in population, or if there are sampling issues. “Endogeneity” is the word used to describe such situations.

**Example 5.6** (Truncated Sampling). Suppose  $\beta_1$  is positive so you have a positively sloped PRF. Suppose you have a “truncated sample” where you cannot observe any observation where  $Y_i > c$ . This means that the only observations with larger values of  $X_i$  that are included in your sample will be the ones with lower or negative values of  $\epsilon_i$ , since a large  $X_i$  together with large positive  $\epsilon_i$  makes  $Y_i > c$  more likely. This implies a negative correlation between  $X$  and  $\epsilon$ , and invalidates the assumption  $E[\epsilon_i | X_1, \dots, X_N] = 0$ . The following is an empirical illustration where the PRF has a positive slope, and observations with  $Y_i > 1500$  are unavailable. The plot in Fig. 5.5 shows the full (black circles) and truncated (red x's) samples. The estimated OLS sample regression line for the full data set (black) and the truncated data set (red) are shown, illustrating the downward bias in  $\hat{\beta}_1$ .

```
set.seed(13)
X <- rnorm(100, mean=50, sd=20)
Y <- 1220 + 4*X + rnorm(100, mean=0, sd=50)
df_notrunc <- data.frame(X, Y)
```

```
df_trunc <- filter(df_notrunc, Y<=1500)
ggplot() +
  geom_point(data=df_notrunc,aes(x=X,y=Y), pch=1, size=2) +
  geom_smooth(data=df_notrunc,aes(x=X,y=Y), method="lm", se=FALSE, col="black") +
  geom_point(data=df_trunc, aes(x=X,y=Y), pch=4, size=2,col='red') +
  geom_smooth(data=df_trunc,aes(x=X,y=Y), method="lm", se=FALSE, col="red") +
  theme_minimal()
```

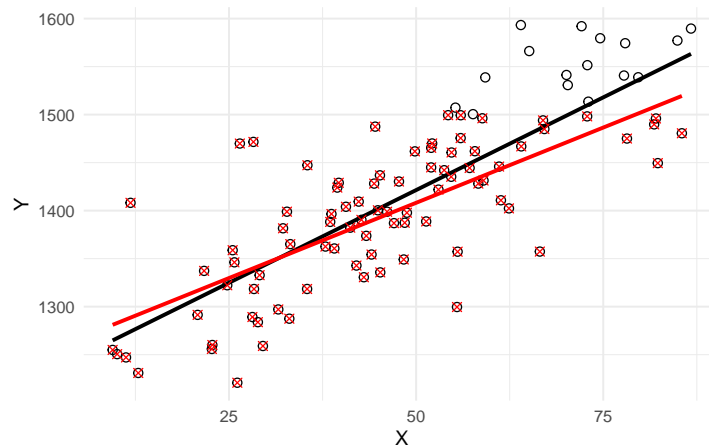


Figure 5.5: A truncated data set.

**Example 5.7** (Measurement Error). Another kind of sampling issue is measurement error. Suppose  $Y = \beta_0 + \beta_1 X + \epsilon$  describes the relationship between  $Y$  and  $X$ , but  $X$  is only observed with error, i.e., you observe  $X^* = X + u$ . Assume that the measurement error  $u$  is independent of  $X$ . Then

$$\begin{aligned} Y &= \beta_0 + \beta_1 X + \epsilon \\ &= \beta_0 + \beta_1 (X^* - u) + \epsilon \\ &= \beta_0 + \beta_1 X^* + (\epsilon - \beta_1 u) \\ &= \beta_0 + \beta_1 X^* + v \end{aligned}$$

where  $v = \epsilon - \beta_1 u$ . You proceed with what appears to be the only feasible option to you, which is to run the regression

$$Y = \beta_0 + \beta_1 X^* + v,$$

but since  $u$  is correlated with  $X^*$ , the assumption  $E[v|X^*] = 0$  does not hold. In the simulated example below, we have a positively sloped PRF, shown in red, and measurement error in the regressor. Since  $\beta_1$  is positive,  $X^*$  and  $v$  are negatively correlated, meaning that the error term  $v$  will tend to be positive for smaller  $X^*$  and negative for larger  $X^*$ . This tendency is visible in Fig. 5.6 below. The red circles are the sample you would have observed with no measurement error. The black circles are the same sample points, but with measurement error in the  $X_i$ 's.

```
set.seed(13)
X <- rnorm(100, mean=50, sd=20)
Y <- 1220 + 4*X + rnorm(100, mean=0, sd=10)
```



```

Xstar <- X + rnorm(100, mean=0, sd=10)
df_measerr <- data.frame(X, Y, Xstar)
ggplot() +
  geom_point(data=df_measerr, aes(x=Xstar, y=Y), pch=1, size=2) +
  geom_point(data=df_measerr, aes(x=X, y=Y), pch=1, size=2, col="red") +
  geom_abline(intercept=1220, slope=4, col='red') +
  geom_smooth(data=df_measerr, aes(x=Xstar, y=Y), method="lm", col='black',
              se=FALSE, linewidth=0.8) + xlab("X, Xstar") +
  theme_minimal()

```

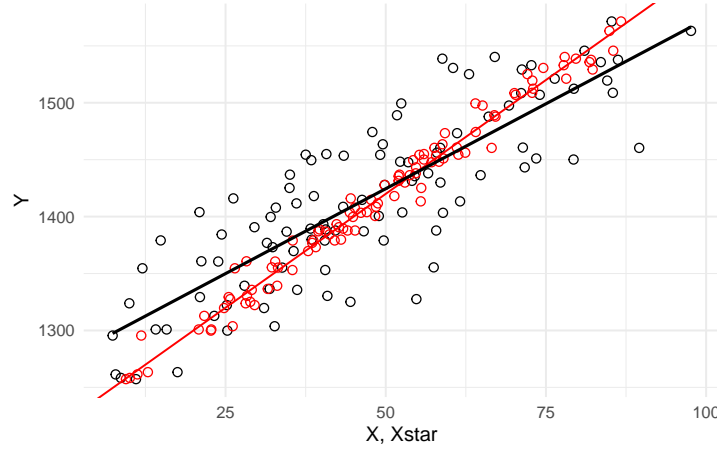


Figure 5.6: A data set with measurement error in the regressor.

**Example 5.8** (Simultaneity Bias). Suppose the market for a good is governed by the following demand and supply equations

$$Q_t^d = \delta_0 + \delta_1 P_t + \epsilon_t^d \quad (\text{Demand Eq } \delta_1 < 0)$$

$$Q_t^s = \alpha_0 + \alpha_1 P_t + \epsilon_t^s \quad (\text{Supply Eq } \alpha_1 > 0)$$

$$Q_t^s = Q_t^d \quad (\text{Market Clearing})$$

where  $Q$  and  $P$  represent log quantities and log prices respectively, so  $\delta_1$  and  $\alpha_1$  represent price elasticities of demand and supply respectively. Suppose the demand shock  $\epsilon_t^d$  and supply shock  $\epsilon_t^s$  are iid noise terms with zero means and variances  $\sigma_d^2$  and  $\sigma_s^2$  respectively, and are mutually uncorrelated. Market clearing means that observed quantity and prices occur at the intersection of the demand and supply equations, i.e., observed quantity and prices are such that

$$\delta_0 + \delta_1 P_t + \epsilon_t^d = \alpha_0 + \alpha_1 P_t + \epsilon_t^s.$$

which we can solve to get

$$P_t = \frac{\alpha_0 - \delta_0}{\delta_1 - \alpha_1} + \frac{\epsilon_t^s - \epsilon_t^d}{\delta_1 - \alpha_1}. \quad (5.27)$$

Substituting this expression for prices into either the demand or supply equation gives

$$Q_t = \left( \delta_0 + \delta_1 \frac{\alpha_0 - \delta_0}{\delta_1 - \alpha_1} \right) + \frac{\delta_1 \epsilon_t^s - \alpha_1 \epsilon_t^d}{\delta_1 - \alpha_1}. \quad (5.28)$$

Equations Eq. 5.27 and Eq. 5.28 imply

$$\text{var}[P_t] = \frac{\sigma_s^2 + \sigma_d^2}{(\delta_1 - \alpha_1)^2} \quad \text{and} \quad \text{cov}[P_t, Q_t] = \frac{\delta_1 \sigma_s^2 + \alpha_1 \sigma_d^2}{(\delta_1 - \alpha_1)^2}.$$

This means that in a regression of  $Q_t = \beta_0 + \beta_1 P_t + \epsilon_t$ , we will get

$$\hat{\beta}_1 \rightarrow_p \frac{\text{cov}[Q_t, P_t]}{\text{var}[P_t]} = \frac{\delta_1 \sigma_s^2 + \alpha_1 \sigma_d^2}{\sigma_s^2 + \sigma_d^2}$$

which is neither the price elasticity of demand nor the price elasticity of supply.

The problem in this example is that prices and quantities are simultaneously determined by the intersection of the demand and supply functions; both prices and quantities are “endogenous” variables. The consequence of this is that regardless of whether you view the regression of  $Q_t$  on  $P_t$  as estimating the demand or supply equation, the noise term in the regression will be correlated with  $P_t$ . A supply shock shifts the supply function and changes both  $Q_t$  and  $P_t$ . Likewise, a demand shock shifts the demand function and again changes both  $Q_t$  and  $P_t$ . The use of the term “endogeneity” comes from applications like these, but it is now used for all situations where the noise term is correlated with one or more of the regressors.

**Example 5.9** (Omitted Variables). Suppose  $X$  and  $Z$  are “causal” variables for  $Y$ , and we wish to measure the effect  $X$  has on  $Y$ . Suppose

$$Y = \alpha_0 + \alpha_1 X + \alpha_2 Z + w \tag{5.29}$$

where  $\alpha_2 \neq 0$  and  $E[w|X, Z] = 0$ . That is,

$$E[Y|X, Z] = \alpha_0 + \alpha_1 X + \alpha_2 Z.$$

If we omit  $Z$  from Eq. 5.29 and write it as

$$Y = \alpha_0 + \alpha_1 X + \epsilon$$

we would be subsuming the variable  $Z$  into the noise term  $\epsilon$

$$\epsilon = \alpha_2 Z + w.$$

If  $X$  and  $Z$  are correlated, then  $\text{cov}[X, \epsilon] \neq 0$ . In a regression of  $Y$  on  $X$ , the OLS estimator  $\hat{\alpha}_1$  will be biased and inconsistent for  $\alpha_1$ . We have not specified enough detail to derive an expression for the bias, but it is easy to show inconsistency. The OLS estimator for  $\hat{\alpha}_1$  is

$$\hat{\alpha}_1 = \alpha_1 + \frac{\sum_{i=1}^N (X_i - \bar{X}) \epsilon_i}{\sum_{i=1}^N (X_i - \bar{X})^2}$$

which will converge in probability to

$$\alpha_1 + \frac{\text{cov}[X, \epsilon]}{\text{var}[X]} \neq \alpha_1.$$

That is, the OLS estimator  $\hat{\alpha}_1$  will be inconsistent for  $\alpha_1$  and will misrepresent the degree of causality from  $X$  to  $Y$ .

The following is a more detailed example of a situation described in Example 5.9

**Example 5.10.** Suppose the true relationship of the variables  $X$ ,  $Y$  and  $Z$  is given by

$$\begin{aligned} Y &= \alpha_0 + \alpha_1 Z + u \\ X &= Z + v \end{aligned} \tag{5.30}$$

That is, both  $Y$  and  $X$  are driven by a third variable  $Z$  but are otherwise not connected. Assume the noise terms  $u$  and  $v$  are independent of each other, with  $u \sim N(0, \sigma_u^2)$  and  $v \sim N(0, \sigma_v^2)$ . Suppose also that  $Z \sim N(0, \sigma_Z^2)$ . In this example, there are in fact values  $\beta_0$  and  $\beta_1$  such that assumptions A1 to A3 in assumption set A hold. It can be shown (see exercises) that

$$E[Z|X] = \frac{\sigma_Z^2}{\sigma_Z^2 + \sigma_v^2} X. \tag{5.31}$$

(Intuition for Eq. 5.31 :  $Z$  obviously has information about  $X$ , but given  $X$  one also gets information about  $Z$ . If  $\sigma_v^2 = 0$  obviously  $Z = X$ . On the other hand, if  $\sigma_v^2$  is very large, then the information content in  $X$  about  $Z$  is small, and the expected value should be close to the unconditional mean of  $Z$  which is zero.) From Eq. 5.31, we get

$$E[Y|X] = \alpha_0 + \alpha_1 E[Z|X] + E[u|X] = \alpha_0 + \frac{\alpha_1 \sigma_Z^2}{\sigma_Z^2 + \sigma_v^2} X. \tag{5.32}$$

It follows that  $E[\epsilon|X] = 0$  where  $\epsilon = Y - \beta_0 - \beta_1 X$  with  $\beta_0 = \alpha_0$  and  $\beta_1 = \frac{\alpha_1 \sigma_Z^2}{\sigma_Z^2 + \sigma_v^2}$ . It is also straightforward to show that the conditional variance is a constant.

Since Assumption Set A holds, OLS estimation of a regression of  $Y_i$  on  $X_i$  will produce an unbiased and consistent estimator for  $\beta_1$ , but this non-zero  $\beta_1$  does not indicate causality of  $X$  on  $Y$ . In this example, it is  $Z$  that drives both  $Y$  and  $X$ . Any movements in  $X$  resulting from  $v$  but not  $Z$  will not result in any response in  $Y$ . But because  $Z$  drives both  $X$  and  $Y$ , one will observe a correlation between  $X$  and  $Y$ .

The next example is an illustration of the previous two examples.

**Example 5.11.** We use the data in **earnings.xlsx** for this example. This data set contains a sample of 540 individuals with information including earnings, height, sex, years of schooling, among other variables. We run a regression of  $\ln(\text{earnings})$  on  $\text{height}$ .

```
df_cause <- read_csv("data\\earnings.csv")
mdl_cause <- lm(log(earnings)~height, data=df_cause)
summary(mdl_cause)
```

Call:

```
lm(formula = log(earnings) ~ height, data = df_cause)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.21599	-0.37662	-0.01233	0.33506	2.06550

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.026104	0.387961	0.067	0.946
height	0.040871	0.005722	7.143	2.98e-12 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.563 on 538 degrees of freedom

Multiple R-squared: 0.08663, Adjusted R-squared: 0.08493

F-statistic: 51.03 on 1 and 538 DF, p-value: 2.976e-12

We find that the effect of *height* on  $\ln(\text{earnings})$  is statistically significant (and economically significant at 4% per inch), and “explains” over 8.6% of the variation in  $\ln(\text{earnings})$ . However, all this seems unlikely to be a true indication of “causality” in the sense of height causing earnings, *ceteris paribus*. In particular, the sample includes observations on both males and females. The result most likely reflects the wage gap between males and females. This is picked up by *height* since there is also a systematic difference in the heights of males and females.

One of the main applications of regression analysis is to empirically explore if one variable causes another, or at least the extent to which one variable causes another. However, what simple linear regression captures is correlation between two variables, and correlation does not necessarily reflect causality.

The ideal scenario for measuring causation of  $X$  to  $Y$  would be to literally hold everything else fixed, change  $X$ , and see how  $Y$  changes, but this is obviously impossible to do here. In some applications we may be able to sample in such a way so that the noise  $\epsilon$  is uncorrelated with  $X$  by construction (this is the approach of randomized controlled trials). With observational data, we have to find some other way to ‘control’ for confounding factors.

The solution to the omitted variable problem is (wherever possible) to include all relevant explanatory variables into the regression. This takes us to the multiple linear regression framework, which we cover in the next chapter. The solution to the other endogeneity problems require more advanced techniques.

## 5.7 Exercises

**Exercise 5.1.** What is the OLS estimator for  $\beta_0$  in the linear regression model with no regressor, i.e., in the regression  $Y_i = \beta_0 + \epsilon_i$ ? What is the  $R^2$  for this regression?

*Remark: All of our regressions will include the intercept term, unless explicitly stated otherwise.*

**Exercise 5.2.** You should have found the  $\hat{\beta}_0$  in Exercise 5.1 to be the sample mean  $\bar{Y}$ . Suppose you choose to estimate  $\beta_0$  using some other measure of location (perhaps the median of  $Y_i$ ). What can you say about the  $R^2$  in this case?

**Exercise 5.3.** Show for the simple linear regression model (with intercept term included) that

the  $R^2$  is the square of the correlation coefficient of  $Y_i$  and  $\hat{Y}_i$ , i.e.,

$$R^2 = \left[ \frac{\sum_{i=1}^N (Y_i - \bar{Y})(\hat{Y}_i - \bar{\hat{Y}})}{\sqrt{\sum_{i=1}^N (Y_i - \bar{Y})^2} \sqrt{\sum_{i=1}^N (\hat{Y}_i - \bar{\hat{Y}})^2}} \right]^2$$

This is where the name “ $R^2$ ” comes from. Show also that

$$R^2 = \left[ \frac{\sum_{i=1}^N (Y_i - \bar{Y})(X_i - \bar{X})}{\sqrt{\sum_{i=1}^N (Y_i - \bar{Y})^2} \sqrt{\sum_{i=1}^N (X_i - \bar{X})^2}} \right]^2$$

The first of these two expressions will hold for OLS estimation of the multiple linear regression model. The second is specific to the simple linear regression model.

**Exercise 5.4.** Explain why  $\sum_{i=1}^N w_i v_i = 0$  in Eq. 5.20.

**Exercise 5.5.** Show that  $E[Z|X] = \frac{\sigma_Z^2}{\sigma_Z^2 + \sigma_v^2} X$  in Example 5.9). (*Hint: Look up the notes on the Bivariate Normal Distribution.*)

**Exercise 5.6.** Show for the simple linear regression (with intercept) that any observation  $(X_i, Y_i)$  such that  $X_i = \bar{X}$  does not numerically affect the value of the OLS estimator  $\hat{\beta}_1$ .

**Exercise 5.7.** Suppose we wish to estimate a simple linear regression on a data set  $\{X_i, Y_i\}_{i=1}^{80}$ , but because of a clerical error, twenty additional rows of zeros were added to the excel file containing the data. That is, instead of estimating the regression on  $\{X_i, Y_i\}_{i=1}^{80}$ , the regression was estimated on  $\{X_i, Y_i\}_{i=1}^{100}$  where  $(X_i, Y_i) = (0, 0)$  for  $i = 81, \dots, 100$ . Explain why this error does not affect the numerical value of  $\hat{\beta}_1$ . Will it affect the numerical value of  $\hat{\beta}_0$ ? What about their variances?

**Exercise 5.8.** Suppose you have a data set  $\{X_i, Y_i\}_{i=1}^N$  where  $X_i$  is binary, with  $N_0$  observations where  $X_i = 0$  and  $N_1$  observations where  $X_i = 1$ ,  $N_0 + N_1 = N$ . You estimate a simple linear regression model on this data set using OLS. Show that  $\hat{\beta}_0$  is equal to the sample mean of  $Y_i$  over all observations where  $X_i = 0$ , i.e.,  $\hat{\beta}_0 = (1/N_0) \sum_{i: X_i=0} Y_i$  and  $\hat{\beta}_1$  is equal to the sample mean of  $Y_i$  over all observations where  $X_i = 1$  minus the sample mean of  $Y_i$  over all observations where  $X_i = 0$ , i.e.,

$$\hat{\beta}_1 = \frac{1}{N_1} \sum_{i: X_i=1} Y_i - \frac{1}{N_0} \sum_{i: X_i=0} Y_i.$$

**Exercise 5.9.** We have shown that measurement error in the regressor leads to inconsistent estimators. Does measurement error in the regressand  $Y$  also result in inconsistent estimators?

**Exercise 5.10.** The data set **Anscombe.xlsx** contains for four pairs of variables:  $(X1, Y1)$ ,  $(X2, Y2)$ ,  $(X3, Y3)$  and  $(X4, Y4)$ .

- Regress  $Y1$  on  $X1$ ,  $Y2$  on  $X2$ ,  $Y3$  on  $X3$  and  $Y4$  on  $X4$  in four separate regressions (all with intercept term). Report the results as given in `coef(summary())`, and the  $R^2$  for all four regressions. What do you observe? For each regression, plot the data and the fitted SRF in one diagram (one figure per regression) and plot the residuals against the regressors in a separate diagram (also one each per regression), and comment on the plots. This exercise shows you the importance of visually inspecting your regressions.

- b. For each of the four regressions in part (a), verify that the residuals sum to zero and are orthogonal to the regressor.

**Exercise 5.11.** Derive Eq. 5.22.

**Exercise 5.12.** Refer to the `earnings.csv` data set which contains observations of a sample of adult workers. The variable `earnings` is hourly earnings and `age` is the age of workers. Consider the regression

$$\log(\text{earnings}) = \beta_0 + \beta_1 \text{age} + \epsilon.$$

- a. What is the interpretation of  $\beta_1$ ? b. Does  $\beta_0$  have an economic interpretation? c. What would be the interpretation of  $\beta_0$  be in the regression

$$\log(\text{earnings}) = \beta_0 + \beta_1(\text{age} - 21) + \epsilon?$$

## Chapter 6

### Multiple Linear Regression

Multiple linear regression extends the simple linear regression framework to multiple regressors:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_{K-1} X_{K-1} + \epsilon. \quad (6.1)$$

Here  $X_1, X_2, \dots, X_{K-1}$  represent different variables, although one may be some transformation of another. That is, it may be that  $X_1$  and  $X_2$  represent completely different random variables, such as **age** and work experience **wexp**, or it may be that  $X_2$  is a function of  $X_1$ , e.g.,  $X_2 = X_1^2$ . In either case, when speaking of the multiple linear regression model generically, we will denote the regressors as  $X_1, X_2, \dots, X_{K-1}$ .<sup>1</sup> The variable  $\epsilon$  is again a catch-all noise term. The parameter  $\beta_0$  is the intercept, and  $\beta_1, \dots, \beta_{K-1}$  are the “slope coefficients”. The term “coefficients” will refer to both the intercept and the slope coefficients.

The following examples illustrate the usefulness of extending the simple linear regression to multiple linear regression.

**Example 6.1.** Suppose  $X$  and  $Z$  are correlated variables, but only  $Z$  is a true causal variable of  $Y$ . To fix ideas, suppose  $X$  is height,  $Z$  is gender and  $Y$  is wage. Estimating the regression

$$Y = \beta_0 + \beta_1 X + \epsilon$$

on data from a population where there is a gender wage gap will generally result in a significant estimate of  $\beta_1$ . This reflects only the common correlation between both  $Y$  and  $X$  with  $Z$ . As we will see, the multiple linear regression model

$$Y = \beta_0 + \beta_1 X + \beta_2 Z + \epsilon$$

can help separate the effects of  $X$  and  $Z$  on  $Y$ . If both  $X$  and  $Z$  are *bona fide* causal variables, the multiple linear regression model will be helpful in measuring the *extent* of causality between the two variables on the dependent variable.

**Example 6.2.** The simple linear regression model assumes a linear conditional expectation, but it may be that the conditional expectation is non-linear in the variables. In some cases, transformations to the regressor or the regressand (or both) suffices. For instance, the “log-linear” model

$$\ln Y = \beta_0 + \beta_1 \ln X + \epsilon$$

may fit the data well. In other cases, however, we might need greater flexibility in specifying the form of the conditional expectation. The multiple linear regression framework gives us a good

---

<sup>1</sup>If we use  $X_1, X_2$ , etc. to denote different variables, then the  $i$ th observation of regressor  $X_j$  will be denoted  $X_{j,i}$ . If we use  $X, Y, Z$ , to denote different variables, then the  $i$ th observation of these variables will be denoted  $X_i, Y_i, Z_i$ .

deal of flexibility. For example, we can have specifications such as

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon, \quad (6.2)$$

or

$$Y = \beta_0 + \beta_1 X + \beta_2 D.X + \epsilon, \quad (6.3)$$

where  $D$  is a binary variable that is equal to one if  $X$  is greater or equal to some threshold  $\xi$ , and zero otherwise.

**Example 6.3.** It may be that there are several variables that are good predictor for the dependent variable. Multiple linear regression models allow us a flexible way to use multiple predictors using specifications such as

$$Y = \beta_0 + \beta_1 X + \beta_2 Z + \epsilon$$

or even

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 Z + \beta_4 Z^2 + \beta_5 Z.X + \epsilon \quad (6.4)$$

The regressors  $D.X$  in Eq. 6.3 and  $Z.X$  in Eq. 6.4 are called interaction terms.

**Example 6.4.** The ability to specify flexible non-linear relationships in the multiple linear regression framework is also helpful in causal applications. It may well be that the relationship between dependent variable and a causal variable is non-linear. For instance, e.g., the rate of increase in earnings as a worker gets older may decline with age, in which case the specification

$$\ln earnings = \beta_0 + \beta_1 age + \beta_2 age^2 + \beta_3 wexp + \epsilon \quad (6.5)$$

may be appropriate. In Eq. 6.5, we have

$$\frac{\delta \ln earnings}{\delta age} = \beta_1 + \beta_2 age.$$

which allows the percentage annual increase in earnings to depend on age. We can also allow the rate of increase to depend also on other variables which specifications such as

$$\ln earnings = \beta_0 + \beta_1 age + \beta_2 age^2 + \beta_3 wexp + \beta_4 wexp.age + \epsilon \quad (6.6)$$

where

$$\frac{\delta \ln earnings}{\delta age} = \beta_1 + \beta_2 age + \beta_4 wexp.$$

Multiple regression models are usually estimated using ordinary least squares. In this chapter, we focus on the multiple linear regression with two regressors

$$Y = \beta_0 + \beta_1 X + \beta_2 Z + \epsilon.$$

The general multi-regressor case is best dealt with using matrix algebra, which we leave for a later chapter. We use the two regressor case to build intuition regarding issues such as bias-variance tradeoffs, how the inclusion of an additional variable helps to “control” for the confounding effect



of that variable, and basic ideas about joint hypotheses testing. We continue to assume that you are working with cross-sectional data. Be reminded that the variables  $Y$ ,  $X$  and  $Z$  may be transformations of the variables of interest. Furthermore,  $X$  and  $Z$  may be transformations of the *same* variable, e.g., we may have  $Z = X^2$ .

*We use the following packages in this chapter.*

```
library(tidyverse)
library(patchwork)
library(readxl)
library(car)
```

## 6.1 OLS Estimation of the Multiple Linear Regression Model

Let  $\{Y_i, X_i, Z_i\}_{i=1}^N$  be your sample. For any estimators  $\hat{\beta}_0$ ,  $\hat{\beta}_1$  and  $\hat{\beta}_2$  (whether or not obtained by OLS), define the **fitted values** to be

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i + \hat{\beta}_2 Z_i \quad (6.7)$$

and the **residuals** to be

$$\hat{\epsilon}_i = Y_i - \hat{Y}_i = Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i - \hat{\beta}_2 Z_i \quad (6.8)$$

for  $i = 1, 2, \dots, N$ . The OLS method chooses  $(\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2)$  to be those values of  $(\beta_0, \beta_1, \beta_2)$  that minimize the **sum of squared residuals**  $SSR = \sum_{i=1}^N (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i - \hat{\beta}_2 Z_i)^2$ , i.e.,

$$\hat{\beta}_0^{ols}, \hat{\beta}_1^{ols}, \hat{\beta}_2^{ols} = \operatorname{argmin}_{\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2} \sum_{i=1}^N (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i - \hat{\beta}_2 Z_i)^2. \quad (6.9)$$

The phrase “ $\operatorname{argmin}_{\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2}$ ” means “the values of  $\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2$  that minimize ...”. The OLS estimators can be found by solving the first order conditions:

$$\begin{aligned} \left. \frac{\partial SSR}{\partial \hat{\beta}_0} \right|_{\hat{\beta}_0^{ols}, \hat{\beta}_1^{ols}, \hat{\beta}_2^{ols}} &= -2 \sum_{i=1}^N (Y_i - \hat{\beta}_0^{ols} - \hat{\beta}_1^{ols} X_i - \hat{\beta}_2^{ols} Z_i) = 0, \\ \left. \frac{\partial SSR}{\partial \hat{\beta}_1} \right|_{\hat{\beta}_0^{ols}, \hat{\beta}_1^{ols}, \hat{\beta}_2^{ols}} &= -2 \sum_{i=1}^N (Y_i - \hat{\beta}_0^{ols} - \hat{\beta}_1^{ols} X_i - \hat{\beta}_2^{ols} Z_i) X_i = 0, \\ \left. \frac{\partial SSR}{\partial \hat{\beta}_2} \right|_{\hat{\beta}_0^{ols}, \hat{\beta}_1^{ols}, \hat{\beta}_2^{ols}} &= -2 \sum_{i=1}^N (Y_i - \hat{\beta}_0^{ols} - \hat{\beta}_1^{ols} X_i - \hat{\beta}_2^{ols} Z_i) Z_i = 0. \end{aligned} \quad (6.10)$$

We can also write the first order conditions as

$$\sum_{i=1}^N \hat{\epsilon}_i^{ols} = 0, \quad \sum_{i=1}^N \hat{\epsilon}_i^{ols} X_i = 0, \quad \text{and} \quad \sum_{i=1}^N \hat{\epsilon}_i^{ols} Z_i = 0. \quad (6.11)$$

Instead of solving the three-equation three-unknown system Eq. 6.10 directly, we are going to take an alternative but entirely equivalent approach. This alternative approach is indirect, but more illustrative. We focus on the estimation of  $\hat{\beta}_1^{ols}$ . You can get the solution for  $\hat{\beta}_2^{ols}$  by switching  $X_i$  with  $Z_i$  in the steps shown. After obtaining  $\hat{\beta}_1^{ols}$  and  $\hat{\beta}_2^{ols}$ , you can use the first

equation in Eq. 6.10 to compute

$$\hat{\beta}_0^{ols} = \bar{Y} - \hat{\beta}_1^{ols} \bar{X} - \hat{\beta}_2^{ols} \bar{Z}.$$

We begin with the following “auxiliary” regressions:

1. Regress  $X_i$  on  $Z_i$ , and collect the residuals  $r_{i,x|z}$  from this regression, i.e., compute

$$r_{i,x|z} = X_i - \hat{\delta}_0 - \hat{\delta}_1 Z_i, \quad i = 1, 2, \dots, N$$

where  $\hat{\delta}_0$  and  $\hat{\delta}_1$  are the OLS estimators for the intercept and slope coefficients from a regression of  $X_i$  on a constant and  $Z_i$ .

2. Regress  $Y_i$  on  $Z_i$ , and collect the residuals  $r_{i,y|z}$  from this regression, i.e., compute

$$r_{i,y|z} = Y_i - \hat{\alpha}_0 - \hat{\alpha}_1 Z_i, \quad i = 1, 2, \dots, N$$

where  $\hat{\alpha}_0$  and  $\hat{\alpha}_1$  are the OLS estimators for the intercept and slope coefficients from a regression of  $Y_i$  on a constant and  $Z_i$ .

The OLS estimator  $\hat{\beta}_1^{ols}$  obtained from solving the first order conditions Eq. 6.10 turns out to be equal to the OLS estimator of the coefficient on  $r_{i,x|z}$  in a regression of  $r_{i,y|z}$  on  $r_{i,x|z}$  (you can exclude the intercept term here; the sample means of both residuals are zero by construction, so the estimator for the intercept term if included will also be zero). In other words,

$$\hat{\beta}_1^{ols} = \frac{\sum_{i=1}^N r_{i,x|z} r_{i,y|z}}{\sum_{i=1}^N r_{i,x|z}^2}. \quad (6.12)$$

To see this, note that since  $\{r_{i,x|z}\}_{i=1}^N$  are OLS residuals from a regression of  $X_i$  on an intercept term and  $Z_i$ , we have  $\sum_{i=1}^N r_{i,x|z} = 0$  and  $\sum_{i=1}^N r_{i,x|z} Z_i = 0$ . This implies

$$\sum_{i=1}^N r_{i,x|z} \hat{X}_i = \sum_{i=1}^N r_{i,x|z} (\hat{\delta}_0 + \hat{\delta}_1 Z_i) = 0$$

and furthermore,

$$\sum_{i=1}^N r_{i,x|z} X_i = \sum_{i=1}^N r_{i,x|z} (\hat{X}_i + r_{i,x|z}) = \sum_{i=1}^N r_{i,x|z}^2.$$

Now consider the sum  $\sum_{i=1}^N r_{i,x|z} r_{i,y|z}$ . We have

$$\begin{aligned} \sum_{i=1}^N r_{i,x|z} r_{i,y|z} &= \sum_{i=1}^N r_{i,x|z} (Y_i - \hat{\alpha}_0 - \hat{\alpha}_1 Z_i) \\ &= \sum_{i=1}^N r_{i,x|z} Y_i \\ &= \sum_{i=1}^N r_{i,x|z} (\hat{\beta}_0^{ols} + \hat{\beta}_1^{ols} X_i + \hat{\beta}_2^{ols} Z_i + \hat{\epsilon}_i) \\ &= \hat{\beta}_1^{ols} \sum_{i=1}^N r_{i,x|z}^2 + \sum_{i=1}^N r_{i,x|z} \hat{\epsilon}_i. \end{aligned} \quad (6.13)$$

Finally, we note that the first order conditions Eq. 6.10 imply that

$$\begin{aligned}\sum_{i=1}^N r_{i,x|z} \hat{\epsilon}_i &= \sum_{i=1}^N \hat{\epsilon}_i (X_i - \hat{\delta}_0 - \hat{\delta}_1 Z_i) \\ &= \sum_{i=1}^N \hat{\epsilon}_i X_i - \hat{\delta}_0 \sum_{i=1}^N \hat{\epsilon}_i - \hat{\delta}_1 \sum_{i=1}^N \hat{\epsilon}_i Z_i = 0.\end{aligned}$$

Therefore

$$\sum_{i=1}^N r_{i,x|z} r_{i,y|z} = \hat{\beta}_1^{ols} \sum_{i=1}^N r_{i,x|z}^2$$

which gives Eq. 6.12.

Note that in order for Eq. 6.12 to be feasible, we require  $\sum_{i=1}^N r_{i,x|z}^2 \neq 0$ . This means that  $X_i$  and  $Z_i$  cannot be *perfectly* correlated (positively or negatively). They can be correlated, just not perfectly so. Furthermore, in the auxiliary regression of  $X_i$  on  $Z_i$ , we require some variation in  $Z_i$ , i.e., it cannot be that all the  $Z_i$ ,  $i = 1, 2, \dots, N$  have the same value  $c$ . Similarly, to derive  $\hat{\beta}_2^{ols}$ , we require variation in  $X_i$ . All this is perfectly intuitive. If there is no variation in  $X_i$  in the sample, we cannot measure how  $Y_i$  changes with  $X_i$ . Similarly for  $Z_i$ . If  $X_i$  and  $Z_i$  are perfectly correlated, we will not be able to tell whether a change in  $Y_i$  is due to a change in  $X_i$  or in  $Z_i$ , since they move in perfect lockstep. We can summarize all of these requirements by saying that there is no  $(c_1, c_2, c_3) \neq (0, 0, 0)$  such that  $c_1 + c_2 X_i + c_3 Z_i = 0$  for all  $i = 1, 2, \dots, N$ .

The argument presented shows the essence of how confounding factors are ‘controlled’ in multiple regression analysis. Suppose we want to measure how  $Y_i$  is affected by  $X_i$ . If  $Z_i$  is an important determinant of  $Y_i$  that is correlated with  $X_i$ , but omitted from the regression, then the measurement of the influence of  $X_i$  on  $Y_i$  will be distorted. In an experiment, we would control for  $Z_i$  by literally holding it fixed. In applications in economics, this is impossible. What multiple regression analysis does instead is to strip out all variation in  $Y_i$  and  $X_i$  that are correlated with  $Z_i$ , and then measure the correlation in the remaining variation in  $Y_i$  and  $X_i$ .

## 6.2 Algebraic Properties of OLS Estimators

We drop the ‘OLS’ superscript in our notation of the OLS estimators, residuals, and fitted values from this point, and write  $\hat{\beta}_0$ ,  $\hat{\beta}_1$ ,  $\hat{\beta}_2$ ,  $\hat{\epsilon}_i$  and  $\hat{Y}_i$  for  $\hat{\beta}_0^{ols}$ ,  $\hat{\beta}_1^{ols}$ ,  $\hat{\beta}_2^{ols}$ ,  $\hat{\epsilon}_i^{ols}$  and  $\hat{Y}_i^{ols}$  respectively. We will reinstate the ‘ols’ superscript whenever we need to emphasize that OLS was used, or when comparing OLS estimators to estimators derived in another way.

Many of the algebraic properties carry over from the simple linear regression model.

1. We have already noted that the first order conditions can be written as

$$\sum_{i=1}^N \hat{\epsilon}_i = 0, \quad \sum_{i=1}^N X_i \hat{\epsilon}_i = 0 \quad \text{and} \quad \sum_{i=1}^N Z_i \hat{\epsilon}_i = 0.$$

2. This implies that the fitted values  $\hat{Y}_i$  and the residuals are also uncorrelated.
3. The first equation in the first order conditions Eq. 6.10 implies that the point  $(\bar{X}, \bar{Y}, \bar{Z})$  lies on the sample regression function.
4.  $\bar{Y} = \widehat{\bar{Y}}$  continues to hold.
5. The above properties imply that the  $SST = SSE + SSR$  equality continues to hold in the

multiple regression case

$$\sum_{i=1}^N (Y_i - \bar{Y})^2 = \sum_{i=1}^N (\hat{Y}_i - \bar{\hat{Y}})^2 + \sum_{i=1}^N \hat{\epsilon}_i^2. \quad (6.14)$$

As in the simple linear regression case, we can use Eq. 6.14 to define the goodness-of-fit measure:

$$R^2 = 1 - \frac{SSR}{SST}. \quad (6.15)$$

It should be noted that the  $R^2$  will never decrease as we add more variables to the regression. This is because OLS minimizes  $SSR$ , and therefore maximizes  $R^2$ . For example, the  $R^2$  from the regression  $Y = \beta_0 + \beta_1 X + \beta_2 Z + u$  will never be less than the  $R^2$  from the regression  $Y = \alpha_0 + \alpha_1 X + \epsilon$ , and will generally be greater, unless it so happens that  $\hat{\beta}_0 = \hat{\alpha}_0$ ,  $\hat{\beta}_1 = \hat{\alpha}_1$  and  $\hat{\beta}_2 = 0$ . For this reason, the “Adjusted  $R^2$ ”

$$\text{Adj.-}R^2 = 1 - \frac{\frac{1}{N-K} SSR}{\frac{1}{N-1} SST}$$

is sometimes used, where  $K$  is the number of regressors (including the intercept term; for the 2-regressor case that we are focussing on,  $K = 3$ ). The idea is to use unbiased estimates of the variances of  $\epsilon$  and  $Y$ . Since both  $SSR$  and  $N - K$  decrease when additional variables (and parameters) are added into the model, the adjusted  $R^2$  will increase only if  $SSR$  falls enough to lower the value of  $SSR/(N - k)$ . The adjusted  $R^2$  may be used as an alternate measure of goodness-of-fit, but it should not be used as a model selection tool, for reasons we shall come to later later in the chapter.

6. In the derivation Eq. 6.13 of the OLS estimator  $\hat{\beta}_1$ , we noted that

$$\begin{aligned} \sum_{i=1}^N r_{i,x|z} r_{i,y|z} &= \sum_{i=1}^N r_{i,x|z} (Y_i - \hat{\alpha}_0 - \hat{\alpha}_1 Z_i) \\ &= \sum_{i=1}^N r_{i,x|z} Y_i. \end{aligned}$$

This implies that the estimator can also be written as

$$\hat{\beta}_1 = \frac{\sum_{i=1}^N r_{i,x|z} r_{i,y|z}}{\sum_{i=1}^N r_{i,x|z}^2} = \frac{\sum_{i=1}^N r_{i,x|z} Y_i}{\sum_{i=1}^N r_{i,x|z}^2} \quad (6.16)$$

which is the formula for the simple linear regression of  $Y_i$  on  $r_{i,x|z}$ . In other words, you can also get the OLS estimator  $\hat{\beta}_1$  by regressing  $Y_i$  on  $r_{i,x|z}$  without first stripping out the covariance between  $Y_i$  and  $Z_i$ .

7. The expression Eq. 6.16 shows that  $\hat{\beta}_1$  is a linear estimator, i.e.,

$$\hat{\beta}_1 = \sum_{i=1}^N w_i Y_i$$

where here the weights are  $w_i = r_{i,x|z} / \sum_{i=1}^N r_{i,x|z}^2$ . Note that the weights  $w_i$  are made up

solely of observations  $\{X_i\}_{i=1}^N$  and  $\{Z_i\}_{i=1}^N$ , since they are the residuals from a regression of  $X_i$  on  $Z_i$ . Furthermore, the weights have the following properties:

$$\begin{aligned}\sum_{i=1}^N w_i &= 0, \\ \sum_{i=1}^N w_i Z_i &= \frac{\sum_{i=1}^N r_{i,x|z} Z_i}{\sum_{i=1}^N r_{i,x|z}^2} = 0, \\ \sum_{i=1}^N w_i X_i &= \frac{\sum_{i=1}^N r_{i,x|z} X_i}{\sum_{i=1}^N r_{i,x|z}^2} = 1, \\ \sum_{i=1}^N w_i^2 &= \frac{\sum_{i=1}^N r_{i,x|z}^2}{(\sum_{i=1}^N r_{i,x|z}^2)^2} = \frac{1}{\sum_{i=1}^N r_{i,x|z}^2}.\end{aligned}$$

8. If the sample correlation between  $X_i$  and  $Z_i$  is zero, then the coefficient estimate  $\hat{\delta}_1$  in the auxiliary regression where we regressed  $X$  on  $Z$  would be zero, and  $\hat{\delta}_0$  would be equal to the sample mean of  $\bar{X}$ . In other words, we would have  $r_{i,x|z} = X_i - \bar{X}$ , so

$$\hat{\beta}_1 = \frac{\sum_{i=1}^N r_{i,x|z} Y_i}{\sum_{i=1}^N r_{i,x|z}^2} = \frac{\sum_{i=1}^N (X_i - \bar{X}) Y_i}{\sum_{i=1}^N (X_i - \bar{X})^2}$$

This is, of course, just the OLS estimator for the coefficient on  $X_i$  in the *simple* linear regression of  $Y$  on  $X$ . In other words, if the sample correlation between  $X_i$  and  $Z_i$  is zero, then including  $Z_i$  in the regression would not change the value of the simple linear regression estimator for the coefficient on  $X_i$ . We will see shortly that including the additional variable may nonetheless reduce the estimator variance.

### 6.3 Statistical Properties of OLS Estimators

We list Assumption Set B below, which is an adaptation of Assumption Set A to the two-regressor case. With these assumptions, the OLS estimators will again be unbiased and efficient. We will leave the proof of many of these results to a later chapter, when we deal with the general case. In this section, we focus on the OLS estimator variance, and in particular on the trade-off between the benefits of including more variables and the cost of doing so in terms of higher estimator variance.

**Assumption Set B:** Suppose that (B1) there are values  $\beta_0$ ,  $\beta_1$  and  $\beta_2$  such that the random variable  $\epsilon$ , defined as

$$\epsilon = Y - \beta_0 - \beta_1 X - \beta_2 Z$$

satisfies

$$(B2) \quad E[\epsilon|X, Z] = 0,$$

$$(B3) \quad \text{var}[\epsilon|X, Z] = \sigma^2.$$

Suppose also that your data

$$(B4) \quad \{X_i, Y_i, Z_i\}_{i=1}^N \text{ is a random sample from the population, and}$$

$$(B5) \quad c_1 + c_2 X_i + c_3 Z_i = 0 \text{ for all } i = 1, 2, \dots, N \text{ only if } (c_1, c_2, c_3) = (0, 0, 0).$$

Assumption B2 implies that

$$E[Y|X] = \beta_0 + \beta_1 X + \beta_2 Z. \quad (6.17)$$

As in the simple linear regression case, the assumptions imply

- $E[\epsilon_i|x, z] = 0$  for all  $i = 1, \dots, N$ ,
- $E[\epsilon_i^2|x, z] = \sigma^2$  for all  $i = 1, \dots, N$ ,
- $E[\epsilon_i \epsilon_j|x, z] = 0$  for all  $i \neq j$ ,  $i, j = 1, \dots, N$  where we use the notation  $x$  to denote  $X_1, X_2, \dots, X_N$ , and  $z$  to denote  $Z_1, Z_2, \dots, Z_N$ .

OLS is unbiased, since

$$\begin{aligned} \hat{\beta}_1 &= \sum_{i=1}^N w_i Y_i \\ &= \sum_{i=1}^N w_i (\beta_0 + \beta_1 X_i + \beta_2 Z_i + \epsilon_i) \\ &= \beta_1 + \sum_{i=1}^N w_i \epsilon_i. \end{aligned}$$

Taking conditional expectations gives

$$E[\hat{\beta}_1|x, z] = \beta_1 + \sum_{i=1}^N w_i E[\epsilon_i|x, z] = \beta_1.$$

It follows that the unconditional mean is  $E[\hat{\beta}_1] = \beta_1$ .

The conditional variance of  $\hat{\beta}_1$  under Assumption Set B is

$$\begin{aligned} \text{var}[\hat{\beta}_1|x, z] &= \text{var} \left[ \beta_1 + \sum_{i=1}^N w_i \epsilon_i \middle| x, z \right] \\ &= \sum_{i=1}^N w_i^2 \text{var}[\epsilon_i|x, z] \\ &= \frac{\sigma^2}{\sum_{i=1}^N r_{i,x|z}^2}. \end{aligned} \quad (6.18)$$

Since the  $R^2$  from the regression of  $X_i$  on  $Z_i$  is

$$R_{x|z}^2 = 1 - \frac{\sum_{i=1}^N r_{i,x|z}^2}{\sum_{i=1}^N (X_i - \bar{X})^2},$$

we can also write  $\text{var}[\hat{\beta}_1|x, z]$  as

$$\text{var}[\hat{\beta}_1|x, z] = \frac{\sigma^2}{(1 - R_{x|z}^2) \sum_{i=1}^N (X_i - \bar{X})^2}. \quad (6.19)$$

Expression Eq. 6.19 clearly shows the trade-offs involved in adding a second regressor. Suppose the true data generating process is

$$Y = \beta_0 + \beta_1 X + \beta_2 Z + \epsilon, \quad E[\epsilon|X, Z] = 0, \quad \text{var}[\epsilon|X, Z] = \sigma^2$$

but you ran the regression

$$Y = \beta_0 + \beta_1 X + u.$$

If  $X$  and  $Z$  are correlated, then  $X$  and  $u$  are correlated, and you will get biased estimates of  $\beta_1$ . By estimating the multiple linear regression, you are able to get an unbiased estimate of  $\beta_1$  by controlling for  $Z$ . However, the variance of the OLS estimator for  $\beta_1$  changes from

$$\text{var}[\hat{\beta}_1|x] = \frac{\sigma_u^2}{\sum_{i=1}^N (X_i - \bar{X})^2}$$

in the simple linear regression to the expression in Eq. 6.19 for the multiple linear regression. Since  $\sigma^2$  is the variance of  $\epsilon$ , and  $\sigma_u^2$  is the variance of a combination of the uncorrelated variables  $Z$  and  $\epsilon$ , we have  $\sigma^2 < \sigma_u^2$ . This has the effect of reducing the estimator variance (which is good!). However, since  $0 < 1 - R_{x|z}^2 < 1$ , the denominator in the variance expression is smaller in the multiple linear regression case than in the simple linear regression case. This is because in the multiple regression, we have stripped out all variation in  $X$  that is correlated with  $Z$ , resulting in *reduced effective variation* in  $X$ , which in turn increases the estimator variance. In general (and especially in causal applications), one would usually consider the trade-off to be in favor of the multiple regression. However, if  $X$  and  $Z$  are highly correlated ( $R_{x|z}^2$  close to 1), then the reduction in effective variation in  $X$  may be so severe that the estimator variance becomes very large. This tends to reduce the size of the t-statistic, leading to rejection of statistical significance even in cases where the size of the estimate itself may suggest strong *economic* significance.

To compute a numerical estimate for the conditional variance of  $\hat{\beta}_1$ , we have to estimate  $\sigma^2$ . An unbiased estimator for  $\sigma^2$  in the two-regressor case is

$$\widehat{\sigma^2} = \frac{1}{N-3} \sum_{i=1}^N \hat{\epsilon}_i^2. \quad (6.20)$$

The *SSR* is divided by  $N-3$  because three ‘degrees-of-freedom’ were used in computing  $\hat{\beta}_0$ ,  $\hat{\beta}_1$  and  $\hat{\beta}_2$  and these were used in the computation of  $\hat{\epsilon}_i$ . We shall again leave the proof of unbiasedness of  $\widehat{\sigma^2}$  for when we deal with the general case. We estimate the conditional variance of  $\hat{\beta}_1$  using

$$\widehat{\text{var}}[\hat{\beta}_1|x, z] = \frac{\widehat{\sigma^2}}{(1 - R_{x|z}^2) \sum_{i=1}^N (X_i - \bar{X})^2} \quad (6.21)$$

The standard error of  $\hat{\beta}_1$  is the square root of Eq. 6.21.

**Example 6.5.** The dataset `multireg_eg.csv` contains three variables  $X$ ,  $Y$  and  $Z$ . The variable  $Z$  takes integer values from 1 to 5. Fig. 6.1 shows two versions of a scatterplot of  $Y$  on  $X$ , the one in panel b uses shapes to reflect observations associate with different values of  $Z$ .

```
df <- read_csv("data\\multireg_eg.csv", col_types = c("n", "n", "n"))
p1 <- ggplot(data=df) + geom_point(aes(x=X, y=Y)) + theme_minimal()
p2 <- ggplot(data=df) + geom_point(aes(x=X, y=Y, shape=as.factor(Z))) +
  theme_minimal() + theme(legend.position = "bottom") + labs(shape='Z')
p1|p2
```

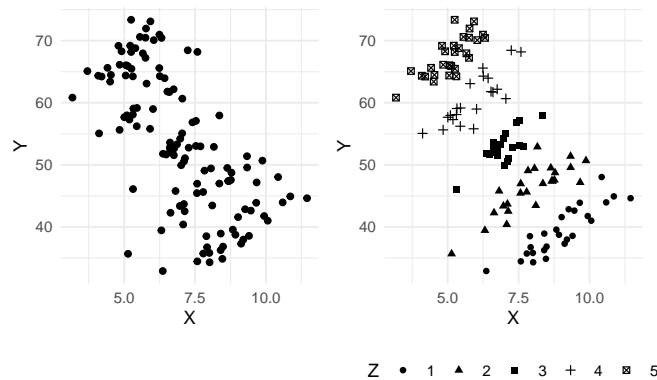


Figure 6.1: The effects of a confounding variable.

There is a clear negative relationships between  $Y$  and  $X$ . However, in panel (b) we see that  $Y$  and  $X$  are in fact *positively* correlated when  $Z$  is fixed at some specific value. However, there is a positive relationship between  $Y$  and  $Z$ , and a negative one between  $Z$  and  $X$ , the net effect of which is to sweep the scatter in the northwest direction as  $Z$  increases, turning a positive correlation between  $Y$  and  $X$  for fixed values of  $Z$  to a negative one overall.

We run two regressions below. The first is a simple linear regression of  $Y$  on  $X$ . The second is a multiple linear regression of  $Y$  on  $X$  and  $Z$ .

```
mdl1 <- lm(Y~X, data=df)
coef(summary(mdl1))
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	82.892162	3.2234151	25.715634	3.394337e-50
X	-4.237024	0.4480503	-9.456581	3.882573e-16

```
cat("R-squared:", summary(mdl1)$r.squared, "\n\n")
```

R-squared: 0.431125

```
mdl2 <- lm(Y~X+Z, data=df)
coef(summary(mdl2))
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.133354	2.0745669	0.5463086	5.858941e-01
X	3.122025	0.2048571	15.2400162	1.473978e-29
Z	10.110311	0.2369353	42.6711993	3.639969e-73

```
cat("R-squared:", summary(mdl2)$r.squared, "\n\n")
```

R-squared: 0.9656532



The simple linear regression shows the negative relationship between  $Y$  and  $X$  when viewed over all outcomes of  $Z$ . The multiple regression disentangles the effect of  $X$  and  $Z$  on  $Y$ . In this case, inclusion of  $Z$  has also reduced the standard error on the estimate of the coefficient on  $X$  despite the reduced variation in  $X$ . This is because  $Z$  accounts for a very substantial proportion of the variation in  $Y$ , as can be seen from the substantial increase in  $R^2$  when it is included (i.e., including  $W$  reduces the variance of the noise term by a lot).

We replicate below the multi-step approach to obtaining the coefficient estimate on  $X$ :

```
mdl1a <- lm(Y~Z, data=df)
r_yz <- residuals(mdl1a)
mdl1b <- lm(X~Z, data=df)
r_xz <- residuals(mdl1b)
df$r_yz <- r_yz
df$r_xz <- r_xz
coef(summary(lm(r_yz~r_xz-1, data=df))) # "-1" in the formula means exclude the intercept
```

	Estimate	Std. Error	t value	Pr(> t )
r_xz	3.122025	0.2031283	15.36972	4.896607e-30

The numerical estimate of the coefficient on  $r\_xz$  is identical to that on  $X$  in the previous regression. The standard errors are similar, but not the same. We emphasize that the auxiliary regression approach is for illustrative purposes only. The standard errors, t-statistic, etc. should all be taken from the previous (multiple) regression.

The plots in Fig. 6.2 illustrate the effect of ‘controlling’ for  $Z$ .

```
plot_theme <- theme_minimal() + theme(legend.position = "none")
p1 <- ggplot(data=df) + geom_point(aes(x=X,y=Y, shape=as.factor(Z))) +
  xlim(c(2.5,12.5)) + ylim(c(30,80)) + plot_theme
p2 <- ggplot(data=df) + geom_point(aes(x=r_xz,y=Y, shape=as.factor(Z))) +
  xlim(c(-5,5)) + ylim(c(30,80)) + plot_theme
p3 <- ggplot(data=df) + geom_point(aes(x=r_xz,y=r_yz, shape=as.factor(Z))) +
  xlim(c(-5,5)) + ylim(c(-25,25)) + plot_theme
p1|p2|p3
```

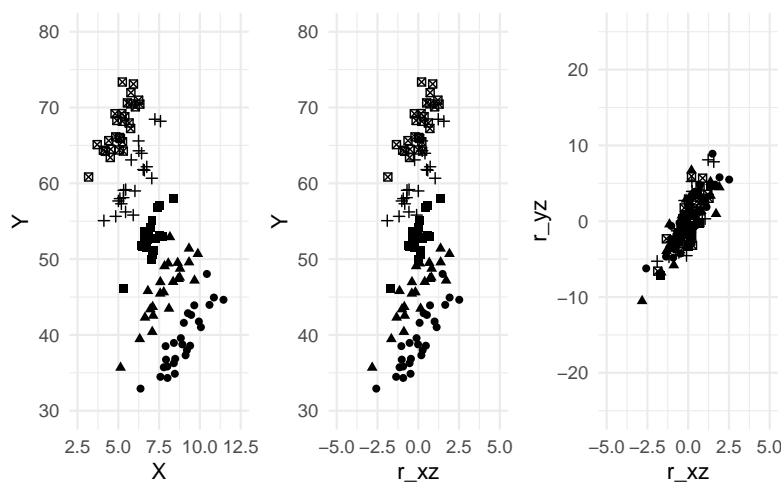


Figure 6.2: Controlling for a confounding variable

The range of the y-axis in the three plots are the same (30 to 80 in the first two, -25 to 25 in the third). Likewise the range of the x-axis is the same across all three plots (2.5 to 12.5 in the first, -5 to 5 in the second and third). This allows you to see the reduced variation in the variables. When we regress  $X$  on  $Z$  and take the residuals, we remove the effect of  $Z$  on  $X$  and also center the residuals around zero (OLS residuals always have sample mean zero). You can see from the second diagram that the variation in  $X$  is reduced, which tends to increase the estimator variance. You can also see that the negative slope has been turned into a positive one, albeit with a lot of noise. By removing the effect of  $Z$  on  $Y$  (which we do when we include  $Z$  in the regression), we reduce the variation on  $Y$ . This *reduces* the estimator variance. The slope coefficient in the simple regression of the data in the last panel gives the effect of  $X$  on  $Y$ , controlling for  $Z$ .

## 6.4 Hypothesis Testing

To test if  $\beta_k$  is equal to some value  $r_k$  in population, we can again use the  $t$ -statistic as in the simple linear regression case:

$$t = \frac{\hat{\beta}_k - r_k}{\sqrt{\widehat{var}[\hat{\beta}_k]}}.$$

If the noise terms are conditionally normally distributed, then the  $t$ -statistic has the  $t$ -distribution, with degrees-of-freedom  $N - K$  where  $K = 3$  in the two-regressor case with intercept. If we do not assume normality of the noise terms, then (as long as the necessary CLTs apply) we use instead the approximate test, using the  $t$ -statistic as defined above, but using the rejection region derived from the standard normal distribution. You can also test whether a linear combination of the parameters are equal to some value. For example, in the regression

$$Y = \beta_0 + \beta_1 X + \beta_2 Z + \epsilon$$

you can test, say,  $H_0 : \beta_1 + \beta_2 = 1$  vs  $H_A : \beta_1 + \beta_2 \neq 1$ . The  $t$ -statistic in this case is

$$t = \frac{\hat{\beta}_1 + \hat{\beta}_2 - 1}{\sqrt{\widehat{var}[\hat{\beta}_1 + \hat{\beta}_2]}}.$$

To compute this you will need the covariance of  $\hat{\beta}_1$  and  $\hat{\beta}_2$ , the derivation of which we leave for a later chapter.

**Example 6.6.** Suppose the production technology of a firm can be characterized by the “Cobb-Douglas Production Function”:

$$Q(L, K) = AL^\alpha K^\beta$$

where  $Q(L, K)$  is the quantity produced using  $L$  units of labor and  $K$  units of capital. The constants  $A$ ,  $\alpha$  and  $\beta$  are the parameters of the model. If we multiply the amount of labor and capital by  $c$ , we get

$$Q(cL, cK) = A(cL)^\alpha (cK)^\beta = c^{\alpha+\beta} AL^\alpha K^\beta.$$

The sum  $\alpha + \beta$  therefore represents the ‘returns to scale’. If  $\alpha + \beta = 1$ , then there is constant returns to scale, e.g., doubling the amount of labor and capital ( $c = 2$ ) results in the doubling

of total production. If  $\alpha + \beta > 1$  then there is increasing returns to scale, and if  $\alpha + \beta < 1$ , we have decreasing returns to scale. A logarithmic transformation of the production function gives

$$\ln Q = \ln A + \alpha \ln L + \beta \ln K.$$

If we have observations  $\{Q_i, L_i, K_i\}_{i=1}^N$  of the quantities produced and amount of labor and capital employed by a set of similar firms in an industry, we could estimate the production function for that industry using the regression

$$\ln Q_i = \ln A + \alpha \ln L_i + \beta \ln K_i + \epsilon_i.$$

A test for constant returns to scale would be the test

$$H_0 : \alpha + \beta = 1 \text{ vs } H_A : \alpha + \beta \neq 1.$$

**Example 6.7.** In the previous chapter, we estimated  $\ln(\text{earnings})$  on *height* using data in `earnings.xlsx` and obtained a statistically significant effect of height on earnings. We conjectured that the regression may be measuring a ‘gender gap’ in wages rather than a ‘height gap’, with *height* acting as a proxy for the sex of the subjects. We now attempt to control for the sex of the subjects by including a dummy variable *male* which is one when an observation is of a male subject, zero otherwise.

```
df_earnings <- read_excel("data\\earnings.xlsx")
mdl_earnings <- lm(log(earnings) ~ height + male, data=df_earnings)
summary(mdl_earnings)
```

Call:

```
lm(formula = log(earnings) ~ height + male, data = df_earnings)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-2.17607	-0.35889	-0.01983	0.33112	2.13062

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.003213	0.539157	1.861	0.06333 .
height	0.025079	0.008332	3.010	0.00273 **
male	0.183117	0.070562	2.595	0.00971 **

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.56 on 537 degrees of freedom

Multiple R-squared: 0.09794, Adjusted R-squared: 0.09458

F-statistic: 29.15 on 2 and 537 DF, p-value: 9.564e-13

Inclusion of the *male* dummy variable has reduced the size of the estimate of the *height* coefficient to 2.5 percent per inch in height (previously it was estimated at four percent). The estimate is still quite economically significant, and also still statistically significant. Perhaps *height* does have a direct effect on  $\ln(\text{earnings})$ , but it is more likely that there are yet more factors that need to be controlled for.

In some cases, we may wish to test multiple hypotheses, e.g., in the two-variable regression, we may wish to test

$$H_0 : \beta_1 = 0 \text{ and } \beta_2 = 0 \text{ vs } H_A : \beta_1 \neq 0 \text{ or } \beta_2 \neq 0.$$

One possibility would be to do individual  $t$ -tests for each of the two hypotheses, but we should be aware that two individual 5% tests is not equivalent to a joint 5% test. The following example illustrates this problem.

**Example 6.8.** We generate 100 observations of three uncorrelated variables  $X$ ,  $Y$  and  $Z$ . We regress  $Y$  on  $X$  and  $Z$ , and collect the  $t$ -statistics on  $X$  and  $Z$ . We repeat the experiment 1000 times (with different draws each time, of course, but the same parameters).

```
set.seed(3)
nreps <- 1000
tx <- tz <- rep(NA,nreps)
N <- 100
for (i in 1:nreps){
  X <- rnorm(N, mean=0, sd=2)
  Z <- rnorm(N, mean=0, sd=2)
  Y <- rnorm(N, mean=0, sd=2)
  df_test <- data.frame(X,Y,Z)
  mdlsim <- lm(Y~X+Z, data=df_test)
  tx[i] <- coef(summary(mdlsim))[2,'t value']
  tz[i] <- coef(summary(mdlsim))[3,'t value']
}
rjt_x <- sum(tx<qt(0.025,N-3) | tx>qt(0.975,N-3))/nreps
cat("Freq. of rejection of Beta_X = 0:", rjt_x, "\n")
```

Freq. of rejection of Beta\_X = 0: 0.058

```
rjt_z <- sum(tz<qt(0.025,N-3) | tz>qt(0.975,N-3))/nreps
cat("Freq. of rejection of Beta_Z = 0:", rjt_z, "\n")
```

Freq. of rejection of Beta\_Z = 0: 0.054

```
rjt_x_or_z <- sum(tz<qt(0.025,197) | tz>qt(0.975,197) |
  tx<qt(0.025,197) | tx>qt(0.975,197))/nreps
cat("Freq. of rejection of Beta_X = 0 and Beta_Z = 0 using two t-tests:", rjt_x_or_z, "\n")
```

Freq. of rejection of Beta\_X = 0 and Beta\_Z = 0 using two t-tests: 0.112

When using a 5%  $t$ -test, we reject the (true) hypothesis that  $\beta_x = 0$  in about 6% of the experiments, close to 5%. These rejections are regardless of whether the  $t$ -test for  $\beta_z = 0$  rejects or does not reject. Likewise, the 5%  $t$ -test for  $\beta_z = 0$  rejects the hypothesis 5.5% of the time, roughly five percent. However, if we say we reject  $\beta_x = 0$  and  $\beta_z = 0$  if either  $t$ -tests rejects the corresponding hypothesis, then the frequency of rejection is much larger, roughly double.

We plot the t-stats below, indicating the critical values for the individual tests. The proportion of points above the upper horizontal line or below the lower one is about 0.05. Similarly, the proportion of points to the left of the left vertical line or to the right of the right one is roughly 0.05. The number of point that meet either of the two sets of criteria is much larger, roughly the sum of the two proportions.

```
df_t <- data.frame(tx,tz)
ggplot(data=df_t) + geom_point(aes(x=tx,y=tz)) +
  geom_hline(yintercept = qt(0.025,N-3), lty='dashed', col='blue') +
  geom_hline(yintercept = qt(0.975,N-3), lty='dashed', col='blue') +
  geom_vline(xintercept = qt(0.025,N-3), lty='dashed', col='blue') +
  geom_vline(xintercept = qt(0.975,N-3), lty='dashed', col='blue') +
  theme_minimal()
```

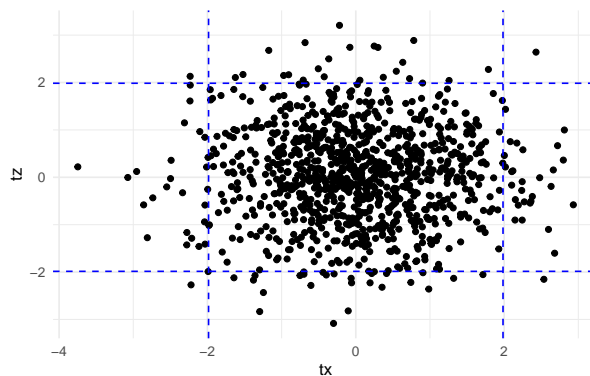


Figure 6.3: Rejection rate when compounding two t-tests

To jointly test multiple hypotheses, we can use the  $F$ -test. Suppose in the regression

$$Y = \beta_0 + \beta_1 X + \beta_2 Z + \epsilon$$

we wish to jointly test the hypotheses

$$H_0 : \beta_1 = 1 \text{ and } \beta_2 = 0 \text{ vs } H_A : \beta_1 \neq 1 \text{ or } \beta_2 \neq 0 \text{ (or both)}.$$

Suppose we run the regression twice, once unrestricted, and another time with the restrictions in  $H_0$  imposed. The regression with the restrictions imposed is

$$Y = \beta_0 + X + \epsilon$$

so the restricted OLS estimator for  $\beta_0$  is the sample mean of  $Y_i - X_i$ , i.e.,

$$\hat{\beta}_{0,r} = (1/N) \sum_{i=1}^N (Y_i - X_i).$$

Calculate the  $SSR$  from both equations. The “unrestricted  $SSR$ ” is

$$SSR_{ur} = \sum_{i=1}^N \hat{\epsilon}_i^2$$

where  $\hat{\epsilon}_i = Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i - \hat{\beta}_2 Z_i$ . The restricted  $SSR$  is

$$SSR_r = \sum_{i=1}^N \hat{\epsilon}_{i,r}^2$$

where  $\hat{\epsilon}_{i,r} = Y_i - \hat{\beta}_{0,r} - X_i$ . Since OLS minimizes  $SSR$ , imposing restrictions will generally increase the  $SSR$ , and never decrease it, i.e.,

$$SSR_r \geq SSR_{ur}.$$

It can be shown that if the hypotheses in  $H_0$  are true (and the noise terms are normally distributed), then

$$F = \frac{(SSR_r - SSR_{ur})/J}{SSR_{ur}/(N-K)} \sim F_{(J, N-K)} \quad (6.22)$$

where  $J$  is the number of restrictions being tested (in our example,  $J = 2$ ) and  $K$  is the number of coefficients to be estimated (including intercept; in our example,  $K = 3$ ). The  $F$ -statistic is always non-negative. The idea is that if the hypotheses in  $H_0$  are true, then imposing the restrictions on the regression would not increase the  $SSR$  by much, and  $F$  will be close to zero. On the other hand, if one or more of the hypotheses in  $H_0$  are false, then imposing them into the regression will cause the  $SSR$  to increase substantially, and the  $F$  statistic will be large. We take a very large  $F$ -statistic, meaning

$$F > F_{\alpha, J, N-K},$$

as statistical evidence that one or more of the hypothesis is false, where  $F_{\alpha, J, N-K}$  is the  $(1 - \alpha)$ -percentile of the  $F_{J, N-K}$  distribution and where  $\alpha$  is typically 0.10, 0.05 or 0.01,

Since  $R^2 = 1 - SSR/SST$ , we can write the  $F$ -statistic in terms of  $R^2$  instead of  $SSR$ . You are asked in an exercise to show that the  $F$ -statistic can be written as

$$F = \frac{(R_{ur}^2 - R_r^2)/J}{(1 - R_{ur}^2)/(N-K)}.$$

Imposing restrictions cannot increase  $R^2$ , and in general will decrease it. The  $F$ -test essentially tests if the  $R^2$  drops significantly when the restrictions are imposed. If the hypotheses being tested are true, then the drop should be slight. If one or more are false, the drop should be substantial, resulting in a large  $F$ -statistic.

If you cannot assume that the noise terms are conditionally normally distributed, then you will have to use an asymptotic approximation. It can be shown that

$$JF \rightarrow_d \chi^2_{(J)}$$

as  $N \rightarrow \infty$ , where  $J$  is the number of hypotheses being jointly tested, and  $F$  is the  $F$ -statistic Eq. 6.22. We refer to this as the “Chi-square Test”.

**Example 6.9.** We continue with Example 6.5. We estimate the model using `lm()` and store the results in `mdl`. We use the `summary()` function to display the results.

```
df <- read_csv("data\\multireg_eg.csv", col_types = c("n", "n", "n"))
mdl <- lm(Y~X+Z, data=df)
summary(mdl)
```

Call:

```
lm(formula = Y ~ X + Z, data = df)
```

Residuals:

Min	1Q	Median	3Q	Max
-4.369	-1.396	-0.077	1.246	6.068

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.1334	2.0746	0.546	0.586
X	3.1220	0.2049	15.240	<2e-16 ***
Z	10.1103	0.2369	42.671	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.07 on 117 degrees of freedom

Multiple R-squared: 0.9657, Adjusted R-squared: 0.9651

F-statistic: 1645 on 2 and 117 DF, p-value: < 2.2e-16

The t-statistics are for testing (separately)  $\beta_x = 0$  and  $\beta_z = 0$ . The F-statistic that is reported is for testing both of these hypotheses jointly, i.e.,  $H_0 : \beta_x = 0$  and  $\beta_z = 0$  versus the alternative that one or both do not hold, and the p-value listed next to the F-statistic is the probability that an  $F_{(2,117)}$  random variable exceeds the computed F-statistic. In this example, we resounding reject the null that both coefficients are zero. The residual standard error is the square root of  $\widehat{\sigma^2}$ , the multiple R-squared is the  $R^2$  discussed earlier. The “Adjusted R-Squared” is the modified  $R^2$  as previously discussed.

As an illustration of the general F test, suppose instead we wish to test that  $\beta_0 = 1$  and  $\beta_z = 3\beta_x$ . The restricted regression is

$$Y = 1 + \beta_x X + 3\beta_x Z + \epsilon = 1 + \beta_x (X + 3Z) + \epsilon.$$

The OLS estimator for the only parameter in the restricted regression,  $\beta_x$ , can be obtained from a regression of  $Y_i - 1$  on  $(X_i + 3Z_i)$  with no intercept term. We have

```
mdlr <- lm((Y-1)~I(X+3*Z)-1, data=df)
coef(mdlr)
```

```
I(X + 3 * Z)
3.276943
```

The restricted residuals can be computed as

$$\hat{\epsilon}_{i,r} = Y_i - 1 - \hat{\beta}_{x,r}(X_i + 3Z_i)$$

The F-statistic and associate p-value is

```
ehat_r <- df$Y - 1 - coef(mdlr)[1] * (df$X + 3*df$Z)
SSR_r <- sum(ehat_r^2)
ehat_ur <- residuals(mdl)
SSR_ur <- sum(ehat_ur^2)
df1 <- 2
df2 <- nobs(mdl) - length(coef(mdl))
F <- ((SSR_r-SSR_ur)/2)/(SSR_ur/(nobs(mdl)-length(coef(mdl))))
Fpval <- 1-pf(F, df1, df2)
X2 <- df1*F
X2pval <- 1-pchisq(X2, df1)
cat("Unrestricted SSR:", round(SSR_ur,6), "\n")
cat("Restricted SSR:", round(SSR_r,6), "\n")
cat("F-stat:",round(F,6))
cat("    p-val:",round(Fpval,6), "\n")
cat("Chi-sq:",round(X2,6))
cat("    p-val:",round(X2pval,6), "\n")
```

```
Unrestricted SSR: 501.2613
```

```
Restricted SSR: 553.7018
```

```
F-stat: 6.12009    p-val: 0.002966
```

```
Chi-sq: 12.24018    p-val: 0.002198
```

The function `linearHypothesis()` in the `car` package can also be used to carry out the F- and Chi-sq tests.

```
linearHypothesis(mdl,c('(Intercept)=1','Z-3*X = 0'), test="F")
```

Linear hypothesis test

Hypothesis:

```
(Intercept) = 1
- 3 X + Z = 0
```

Model 1: restricted model

Model 2: Y ~ X + Z

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	119	553.70				
2	117	501.26	2	52.44	6.1201	0.002966 **

```
---
```

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1



```
linearHypothesis(mdl,c('(Intercept)=1','Z-3*X = 0'), test="Chisq")
```

Linear hypothesis test

Hypothesis:

(Intercept) = 1

- 3 X + Z = 0

Model 1: restricted model

Model 2: Y ~ X + Z

```

Res.Df    RSS Df Sum of Sq Chisq Pr(>Chisq)
1      119 553.70
2      117 501.26  2      52.44 12.24   0.002198 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## 6.5 Exercises

**Exercise 6.1.** Each of the following regressions produces a sample regression function whose slope is  $\hat{\beta}_1$  when  $X_i < \xi$  and  $\hat{\beta}_1 + \hat{\alpha}_1$  when  $X_i \geq \xi$ . Which of them produces a sample regression function that is continuous at  $\xi$ ?

- $Y_i = \beta_0 + \beta_1 X_i + \alpha_1 D_i X_i + \epsilon$  where  $D_i$  is a dummy variable with  $D_i = 1$  if  $X_i > \xi$ ,  $D_i = 0$  otherwise;
- $Y_i = \beta_0 + \alpha_0 D_i + \beta_1 X_i + \alpha_1 D X_i + \epsilon$ ;
- $Y_i = \beta_0 + \beta_1 X_i + \alpha_1 (X_i - \xi)_+ + \epsilon_i$  where

$$(X_i - \xi)_+ = \begin{cases} X_i - \xi & \text{if } X_i > \xi, \\ 0 & \text{if } X_i \leq \xi. \end{cases}$$

**Exercise 6.2.** The following is a “piecewise quadratic regression” model

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 (X - \xi)_+^2 + \epsilon, \quad E[\epsilon|X] = 0.$$

where

$$(X_i - \xi)_+^2 = \begin{cases} (X_i - \xi)^2 & \text{if } X_i > \xi, \\ 0 & \text{if } X_i \leq \xi. \end{cases}$$

Show that the PRF  $E[Y|X]$  is “piecewise quadratic”, following one quadratic equation when  $X \leq \xi$ , and another when  $x > \xi$ . Show that the PRF is continuous, with continuous first derivative.

**Exercise 6.3.** Suppose your estimated sample regression function is

$$\widehat{wage} = -68.28 + 4.163 \text{ age} - 0.052 \text{ age}^2$$

where  $\hat{\alpha}_0$  and  $\hat{\alpha}_1$  are positive and  $\hat{\alpha}_2$  is negative. At what age are wages predicted to start declining with age? Does the intercept have any reasonable economic interpretation?

**Exercise 6.4.** Prove Eq. 6.14.

**Exercise 6.5.** Show that the  $F$ -statistic in Eq. 6.22 can be written as

$$F = \frac{(R_{ur}^2 - R_r^2)/J}{(1 - R_{ur}^2)/(N - K)}$$

where  $R_{ur}$  and  $R_r$  are the  $R^2$  from the unrestricted and restricted regressions respectively,  $J$  is the number of restrictions being tested,  $N$  is the number of observations used in the regression, and  $K$  is the number of coefficient parameters in the unrestricted regression model (including intercept). What does this expression simplify to when testing that all the slope coefficients (excluding the intercept) are equal to zero?

**Exercise 6.6.** Modify the code in Example 6.8 to collect the F-statistic for jointly testing  $\beta_x = 0$  and  $\beta_z = 0$ . Show that the 5% F-test is empirically correctly sized, meaning that the frequency of rejection in the simulation is in fact around 5%.

**Exercise 6.7.** Suppose

$$\begin{aligned} Y &= \alpha_0 + \alpha_1 X + \alpha_2 Z + u \\ Z &= \delta_1 X + v \end{aligned}$$

where  $u$  and  $v$  are independent zero-mean noise terms. Suppose you have a random sample  $\{Y_i, X_i, Z_i\}_{i=1}^N$  and you ran the regression

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i.$$

Show that the OLS estimator  $\hat{\beta}_1$  will be biased for  $\alpha_1$ . What is its expectation? Show that the prediction rule

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$$

still provides unbiased predictions, but that the prediction error variance is greater than the prediction error variance from using the prediction rule

$$Y = \hat{\alpha}_0 + \hat{\alpha}_1 X + \hat{\alpha}_2 Z$$

where  $\hat{\alpha}_0$ ,  $\hat{\alpha}_1$ , and  $\hat{\alpha}_2$  are the OLS estimators for  $\alpha_0$ ,  $\alpha_1$  and  $\alpha_2$  in the regression

$$Y_i = \alpha_0 + \alpha_1 X_i + \alpha_2 Z_i + u_i$$

**Exercise 6.8.** Verify all of the results reported in Example 6.9 by calculating them directly using the formulas developed in the notes (in particular, verify the coefficient estimates, standard errors, t-statistics and associated p-values, the residual standard error, the multiple R-squared and Adjusted R-squared, the F-statistic and the corresponding p-value).

## Chapter 7

### Heteroskedasticity and Specification Tests

We now deal with situations where the conditional variance of the noise term depends on the regressors, i.e., where we have conditional heteroskedasticity. We have already seen that this will not cause bias or inconsistency; we know this because the proofs of unbiasedness and consistency do not make use of the constant conditional variance assumption. However, the derivations of the formulas for the OLS estimator variances and the proof of efficiency of OLS estimators do make use of the constant conditional variance assumption, and so these results no longer apply if there is conditional heteroskedasticity. In this chapter, we present an alternative estimation approach that aims to provide more efficient estimators under this situation. We also present a way of estimating the variance of the OLS coefficient estimators that allows for conditional heteroskedasticity. Finally, we discuss tests for heteroskedasticity, specification form, and normality of noise terms.

We use the following packages in this chapter.

```
library(tidyverse)
library(patchwork)
library(readxl)
library(sandwich)
```

#### 7.1 An Example

We begin with a simple illustrative example where we can directly show all of the consequences of heteroskedasticity.

**Example 7.1.** Suppose

$$Y_i = \beta_1 X_i + \epsilon_i, \quad i = 1, 2, \dots, N \quad (7.1)$$

such that

$$E[\epsilon_i | \mathbf{x}] = 0, \quad E[\epsilon_i^2 | \mathbf{x}] = \sigma^2 X_i^2, \quad \text{and} \quad E[\epsilon_i \epsilon_j | \mathbf{x}] = 0$$

for all  $i \neq j$ ,  $i = 1, 2, \dots, N$ . We also assume that  $\sum_{i=1}^N X_i^2 \neq 0$ . (Here  $X_i$  represents the  $i$ th observation of variable  $X$ , and  $\mathbf{x}$  represents all of the observations of  $X_i$ .) The OLS estimator for  $\beta_1$  in this example is

$$\hat{\beta}_1^{ols} = \frac{\sum_{i=1}^N X_i Y_i}{\sum_{i=1}^N X_i^2} \quad (7.2)$$

which is unbiased for  $\beta_1$ : writing Eq. 7.2 as

$$\hat{\beta}_1^{ols} = \frac{\sum_{i=1}^N X_i Y_i}{\sum_{i=1}^N X_i^2} = \frac{\sum_{i=1}^N X_i (\beta_1 X_i + \epsilon_i)}{\sum_{i=1}^N X_i^2} = \beta_1 + \frac{\sum_{i=1}^N X_i \epsilon_i}{\sum_{i=1}^N X_i^2}$$

and taking conditional expectations gives

$$E[\hat{\beta}_1^{ols} | \mathbf{x}] = \beta_1 + \frac{\sum_{i=1}^N X_i E[\epsilon_i | \mathbf{x}]}{\sum_{i=1}^N X_i^2} = \beta_1.$$

Under our assumptions, the variance of the OLS estimator is

$$\begin{aligned} \text{var}[\hat{\beta}_1^{ols} | \mathbf{x}] &= \frac{\sum_{i=1}^N X_i^2 \text{var}[\epsilon_i | \mathbf{x}]}{\left(\sum_{i=1}^N X_i^2\right)^2} \\ &= \frac{\sum_{i=1}^N X_i^2 (\sigma^2 X_i^2)}{\left(\sum_{i=1}^N X_i^2\right)^2} = \frac{\sigma^2 \sum_{i=1}^N X_i^4}{\left(\sum_{i=1}^N X_i^2\right)^2}. \end{aligned} \quad (7.3)$$

Had you not realized that there is heteroskedasticity, you would have estimated  $\text{var}[\hat{\beta}_1 | \mathbf{x}]$  using the usual OLS formula for the variance when an intercept is excluded, which is

$$\widehat{\text{var}}[\hat{\beta}_1^{ols} | \mathbf{x}] = \frac{s^2}{\sum_{i=1}^N X_i^2}, \quad s^2 = \frac{1}{N-1} \sum_{i=1}^N \hat{\epsilon}_i^2, \quad \hat{\epsilon}_i = Y_i - \hat{\beta}_1 X_i.$$

This variance estimator is based on the assumption of conditional homoskedasticity, and is therefore inappropriate for this example. The form is wrong, and it is not immediately clear what  $s^2$  is estimating.

Furthermore, it turns out that the OLS estimator is inefficient. We show this by presenting a more efficient linear unbiased estimator. Weight each observation by  $1/X_i$  and run the regression

$$\frac{Y_i}{X_i} = \beta_1 + \frac{\epsilon_i}{X_i} = \beta_1 + \epsilon_i^*. \quad (7.4)$$

That is, simply regress  $Y_i/X_i$  on a constant. The modified noise terms in this regression will continue to have zero conditional expectation

$$E[\epsilon_i/X_i | \mathbf{x}] = (1/X_i)E[\epsilon_i | \mathbf{x}] = 0$$

and remain uncorrelated (exercise). Furthermore, its conditional variance is now constant:

$$\text{var}[\epsilon_i/X_i | \mathbf{x}] = (1/X_i^2)\text{var}[\epsilon_i | \mathbf{x}] = \sigma^2.$$

OLS estimation applied to this modified regression model Eq. 7.4 gives the estimator

$$\hat{\beta}_1^{wls} = \frac{1}{N} \sum_{i=1}^N \frac{Y_i}{X_i}. \quad (7.5)$$

The ‘wls’ superscript stands for “weighted least squares”. Since Eq. 7.5 can be written as

$$\hat{\beta}_1^{wls} = \sum_{i=1}^N \left( \frac{1}{NX_i} \right) Y_i,$$

it is a linear estimator. Since this estimator arises out a linear regression with homoskedastic and uncorrelated noise terms that have zero conditional expectation, it is unbiased and minimum variance among all linear unbiased estimators.

The fact that  $\hat{\beta}_1^{wls}$  is BLU suggests that  $\hat{\beta}_1^{ols}$  is not. In our simple example, it is straightforward to demonstrate this fact directly. Since  $\hat{\beta}_1^{wls}$  is a sample mean of observations of a random variable with variance  $\sigma^2$ , its variance is

$$\text{var}[\hat{\beta}_1^{wls}] = \frac{\sigma^2}{N}. \quad (7.6)$$

It can be shown (see exercises) that

$$\frac{\sum_{i=1}^N X_i^4}{\left(\sum_{i=1}^N X_i^2\right)^2} \geq \frac{1}{N}, \quad (7.7)$$

therefore

$$\text{var}[\hat{\beta}_1^{wls}] \leq \text{var}[\hat{\beta}_1^{ols}].$$

The reason OLS estimators are inefficient when there is conditional heteroskedasticity is that some observations are less informative about the population regression line than others, but OLS makes no use of this fact. Information ignored leads to inefficiency. The weighted least squares approach, on the other hand, uses this information directly, by assigning less weight to noisier observations, and more weight to observations whose noise terms have lower variance.

We have shown, in the context of a simple example, that the WLS estimator dominates the OLS estimator under conditional heteroskedasticity in the sense that the WLS estimator is (like OLS) linear and unbiased, but more precise. However, to get the WLS estimator we had to assume that the form of heteroskedasticity is known (in our example, we assumed  $\sigma_i^2 = \sigma^2 X_i^2$ ). There will be situations where we are not quite so sure about the form of heteroskedasticity, and may prefer to stay with OLS despite its inefficiency. The problem with doing so is that the usual formulas for the variance of the estimators is incorrect. Is there a way to correctly estimate  $\text{var}[\hat{\beta}_1^{ols}]$  under heteroskedasticity? Under quite general conditions, the answer is yes. Suppose, for our example, that we assume

$$E[\epsilon_i^2 | x] = \sigma_i^2$$

but without further specifying the form of the heteroskedasticity. The variance of the OLS estimator is

$$\text{var}[\hat{\beta}_1^{ols} | x] = \frac{\sum_{i=1}^N X_i^2 \text{var}[\epsilon_i | x]}{\left(\sum_{i=1}^N X_i^2\right)^2} = \frac{\sum_{i=1}^N X_i^2 \sigma_i^2}{\left(\sum_{i=1}^N X_i^2\right)^2}.$$

It turns out for this example that the variance estimator

$$\widehat{\text{var}}_{HC}[\hat{\beta}_1^{ols} | x] = \frac{\sum_{i=1}^N X_i^2 \hat{\epsilon}_i^2}{\left(\sum_{i=1}^N X_i^2\right)^2} \quad \text{where} \quad \hat{\epsilon}_i = Y_i - \beta_1^{ols} X_i$$

is consistent for the variance of  $\hat{\beta}_1^{ols}$ . We call this the heteroskedasticity-consistent, or heteroskedasticity-robust variance estimator. We discuss heteroskedasticity-robust variance estimators in more detail in a later chapter.

## 7.2 Weighted Least Squares

The idea of weighting observations to account for heteroskedasticity extends to the multiple linear regression model, but for the moment, we stay with the simple linear regression model (with intercept reinstated).

**Assumption Set C:** Suppose you have

- (C1)  $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$ , such that
- (C2)  $E[\epsilon_i|x] = 0$  for all  $i = 1, 2, \dots, N$ ,
- (C3)  $E[\epsilon_i^2|x] = \sigma_i^2 = \sigma^2 \eta_i(x)$  for all  $i = 1, 2, \dots, N$ ,
- (C4)  $E[\epsilon_i \epsilon_j|x] = 0$  for all  $i \neq j$ ,  $i, j = 1, 2, \dots, N$ ,
- (C5)  $c_1 + c_2 X_i = 0$  for all  $i = 1, 2, \dots, N$  only if  $(c_1, c_2) = (0, 0)$ .

We assume conditional heteroskedasticity whose form is known up to some constant factor. The idea of weighted least squares is to weight each observation so that the weighted noise terms are no longer heteroskedastic. That is, we modify the regression equation to

$$\frac{Y_i}{\sqrt{\eta_i(x)}} = \beta_0 \frac{1}{\sqrt{\eta_i(x)}} + \beta_1 \frac{X_i}{\sqrt{\eta_i(x)}} + \frac{\epsilon_i}{\sqrt{\eta_i(x)}}$$

which we can write as

$$Y_i^* = \beta_0 X_{0,i}^* + \beta_1 X_i^* + \epsilon_i^* \quad (7.8)$$

where

$$Y_i^* = Y_i / \sqrt{\eta_i(x)}, \quad X_{0,i}^* = 1 / \sqrt{\eta_i(x)}, \quad \text{and} \quad X_i^* = X_i / \sqrt{\eta_i(x)}.$$

Since  $\eta_i(x)$  is fixed conditional on the regressors, it is straightforward to see that Assumptions C2 and C4 will continue to hold for  $\epsilon_i^*$ . Furthermore the transformed noise term is conditionally homoskedastic:

$$E[\epsilon_i^{*2}|x] = E\left[\left(\frac{\epsilon_i}{\sqrt{\eta_i(x)}}\right)^2 \middle| x\right] = \frac{\sigma^2 \eta_i(x)}{\eta_i(x)} = \sigma^2 \quad \text{for all } i = 1, 2, \dots, N.$$

Therefore OLS on Eq. 7.8 will produce BLU estimators of the coefficients. The OLS estimators on the transformed regression equation in Eq. 7.8 are called the Weighted Least Squares (WLS) estimators of the coefficients. It is equivalent to choosing  $\hat{\beta}_0$  and  $\hat{\beta}_1$  to minimize a sum of weighted squared residuals

$$\text{WLS: Choose } \hat{\beta}_0, \hat{\beta}_1 \text{ to minimize } \sum_{i=1}^N \omega_i \hat{\epsilon}_i^2 = \sum_{i=1}^N \omega_i (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i)^2 \quad (7.9)$$

where the weights are

$$\omega_i = 1/\eta_i(x).$$

The actual form of the transformed equation depends on  $\eta_i(x)$ .

**Example 7.2.** If in Assumption Set C we have  $\eta_i(x) = X_i^2$ , then the appropriate transformed regression equation is

$$\frac{Y_i}{|X_i|} = \beta_0 \frac{1}{|X_i|} + \beta_1 \frac{X_i}{|X_i|} + \frac{\epsilon_i}{|X_i|}. \quad (7.10)$$

If  $X_i$  is always positive, then we can write this equation as

$$\frac{Y_i}{X_i} = \beta_0 \frac{1}{X_i} + \beta_1 + \frac{\epsilon_i}{X_i}. \quad (7.11)$$

In this case, the transformed equation is a simple linear regression of  $Y_i/X_i$  on a constant and  $1/X_i$ . It should be emphasized that although  $\beta_1$  is the intercept term in the transformed equation, it retains its interpretation as the slope coefficient in the original equation. Likewise, the coefficient  $\beta_0$  retains its interpretation as the intercept term in the original equation.

Regardless of the transformation applied to the regression, the coefficients always retain their interpretations from the original un-transformed equation, and it is the un-transformed regression that is the one of interest. The transformed equation is merely used to obtain efficient estimators of the coefficients. After obtaining  $\hat{\beta}_0^{wls}$  and  $\hat{\beta}_1^{wls}$ , you should report your results as

$$\hat{Y} = \hat{\beta}_0^{wls} + \hat{\beta}_1^{wls} X.$$

Likewise, the WLS fitted values are

$$\hat{Y}_i^{wls} = \hat{\beta}_0^{wls} + \hat{\beta}_1^{wls} X_i$$

and the residuals are

$$\hat{\epsilon}_{i,wls} = Y_i - \hat{Y}_i^{wls} = Y_i - \hat{\beta}_0^{wls} - \hat{\beta}_1^{wls} X_i.$$

For the purposes of assessing goodness-of-fit, we should use the WLS residuals:

$$R_{wls}^2 = 1 - \frac{\sum_{i=1}^N \hat{\epsilon}_{i,wls}^2}{\sum_{i=1}^N (Y_i - \bar{Y})^2}.$$

This R-squared will generally be less than the R-squared from OLS estimation (why?) and may even be negative. However, it is the un-transformed equation that we are ultimately interested in. On the other hand, the standard errors and test statistics *should* be based on the estimated transformed equation, since it is the noise term of the transformed equation that meets the required assumptions.

One difficulty with weighted least squares is that we generally do not know the form of the heteroskedasticity. Should  $\eta(x)$  be  $|X_i|$  or  $X_i^2$  or some other function of the regressors? One informal way of investigating this question is to first estimate the equation by OLS (which still gives consistent estimators of the coefficients) then visually exploring the relationship of the squared OLS residuals (serving as a proxy for the noise variance) against  $X_i$ . After estimating the transformed equation, we can confirm the choice of  $\eta(x)$  by testing for heteroskedasticity in the residuals of the transformed equation (not rejecting homoskedasticity would indicate our choice was adequate). We will discuss tests for heteroskedasticity shortly.

**Example 7.3.** We estimate a simple linear regression on the values  $y$  and  $x$  (with intercept) in the data set *heterosk.csv*. Fig. 7.1 displays a plot of  $y$  on  $x$  that shows heteroskedasticity, with the conditional variance is increasing with  $x$ .

```
df_het <- read_csv("data\\heterosk.csv", col_types = c("n", "n", "n"))
plt_het1 <- ggplot(data=df_het) + geom_point(aes(x=x, y=y), size=1) + theme_classic()
plt_het1
```

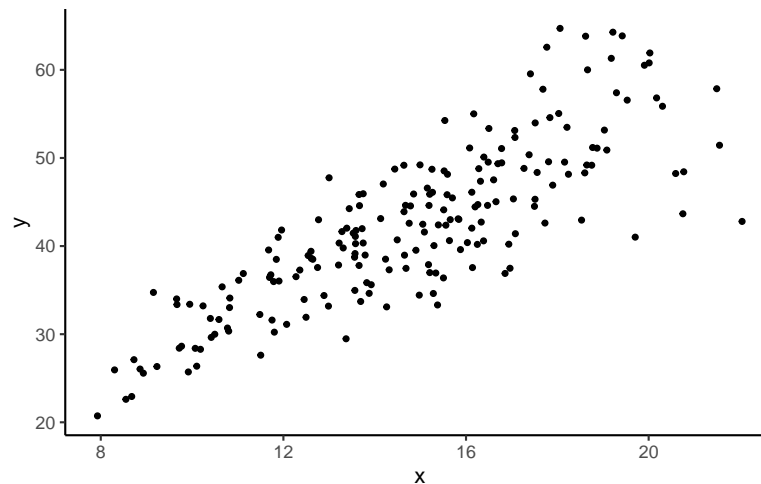


Figure 7.1: A data set with heteroskedasticity.

OLS estimation of this regression gives the following output.

```
ols <- lm(y~x, data=df_het)
sum_ols <- summary(ols)
coef(sum_ols)
cat("R-squared: ", sum_ols$r.squared, "\n")
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	6.285792	1.7817576	3.52786	5.207066e-04
x	2.430744	0.1177712	20.63955	3.034896e-51

R-squared: 0.6826877

Because there is obvious heteroskedasticity in this example, we should not trust the standard errors, t-statistics and p-values presented above. If we wish to stick with OLS, then we have to calculate the heteroskedasticity-robust standard errors (which we have not yet discussed how to do, except in the simple linear regression without intercept). For the time being, we will use the function `vcovHC()` from the `sandwich` package to obtain heteroskedasticity-robust standard errors (explanations in a later chapter!). The heteroskedasticity-robust standard errors are the square root of the diagonal elements of the matrix `rbst_V` in the example below.

```
rbst_V <- vcovHC(ols, type="HC0")
rbst_se <- sqrt(diag(rbst_V))
rbst_output <- coef(sum_ols)
colnames(rbst_output) <- c("Estimate", "rbst-se", "rbst-t", "p-val")
rbst_output[, 'rbst-se'] <- rbst_se
rbst_output[, 'rbst-t'] <- rbst_output[, 'Estimate']/rbst_se
```



```
rbst_output[, 'p-val'] <- 2*(1-pt(abs(rbst_output[, 'rbst-t']), sum_ols$df[2]))
round(rbst_output, 6)
```

```
      Estimate  rbst-se    rbst-t    p-val
(Intercept) 6.285792 1.863926  3.372339 0.000896
x            2.430744 0.136242 17.841435 0.000000
```

Now we assume that  $\text{var}[\hat{\epsilon}_i] = \sigma^2 X_i^2$  and run WLS. The appropriate transformed regression is as given in Eq. 7.11.

```
df_het$ystar <- df_het$y/df_het$x      # Transformed y
df_het$x0star <- 1/df_het$x           # Transformed intercept term
wls1 <- lm(ystar~x0star, data=df_het)
sum_wls1 <- summary(wls1)
coef(sum_wls1)                        # Print Coefficients
cat("R-squared: ", sum_wls1$r.squared, "\n") # Print R-squared of Transformed Eq
```

```
      Estimate Std. Error  t value    Pr(>|t|)
(Intercept)  2.53166   0.1023702 24.730430 1.839271e-62
x0star       4.82571   1.4065451  3.430896 7.321779e-04
R-squared:   0.05611379
```

Our estimated equation is

$$\hat{Y} = 4.8257 + 2.5317 X, \quad N = 200$$

(1.4065) (0.1024)

The standard errors are lower than in the OLS regression, which is not unexpected. The R-squared in the output above refers to the fit of the transformed equation which is not very useful. We calculate the R-squared for the un-transformed regression below

```
ehat <- df_het$y - coef(wls1)[2] - coef(wls1)[1]*df_het$x
ssr <- sum(ehat^2)
sst <- sum((df_het$y - mean(df_het$y))^2)
R2 <- 1 - ssr/sst
cat("R-squared: ", R2, "\n")
```

```
R-squared: 0.6814966
```

The R-squared from the WLS regression is lower than in the OLS regression, as expected, but only slightly so. Finally, we plot the residuals from the transformed regression against  $x$ . There does not appear to be any correlation between the variance of the residuals of the weighted regression and  $x$ , which suggests that our assumption regarding the form of heteroskedasticity in  $\epsilon_i$  is reasonable.

```
plot(df_het$x, residuals(wls1))
```

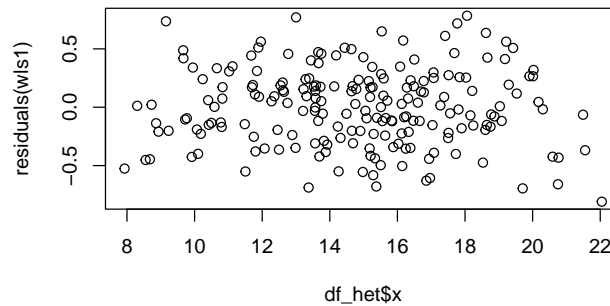


Figure 7.2: Homoskedastic errors.

We can also use the function `lm()` to carry out weighted least squares. Note that the option `weights` refer to weights on the squared residuals, as in the  $\omega_i$  in Eq. 7.9.

```
df_het$wt <- 1/df_het$x^2
wls2 <- lm(y~x,data=df_het, weights=wt)
sum_wls2 <- summary(wls2)
coef(sum_wls2)
cat("R-squared: ", sum_wls2$r.squared,"\n")
```

```
      Estimate Std. Error  t value    Pr(>|t|)
(Intercept)  4.82571    1.4065451   3.430896 7.321779e-04
x            2.53166    0.1023702  24.730430 1.839271e-62
R-squared:   0.755433
```

Notice that the R-squared provided when using `lm()` for WLS is different from what we previously obtained. The R-squared provided here is a “weighted R-squared”, obtained in the following way: let  $\bar{Y}_{wls}$  be the WLS estimator (with the same weights as previously) of  $\beta_0$  from the regression  $Y_i = \beta_0 + \epsilon_i$ . In other words,  $\bar{Y}_{wls}$  is the weighted mean of  $\{Y_i\}_{i=1}^N$ . Then

$$\text{weighted-}R^2 = \frac{\sum_{i=1}^N w_i (\hat{Y}_i^{wls} - \bar{Y}_{wls})^2}{\sum_{i=1}^N w_i (Y_i^{wls} - \bar{Y}_{wls})^2}.$$

In other words, it is the weighted ESS divided by the weighted TSS, centered on the weighted mean of  $\{Y_i\}_{i=1}^N$ . We replicate the `lm()` weighted R-squared below:

```
wls0 <- lm(y~1, data=df_het, weights=wt) # Regression on intercept only
sstw <- sum(df_het$wt * (df_het$y - coef(wls0))^2)
ssew <- sum(df_het$wt*(wls2$fitted.values - coef(wls0))^2)
WeightedR2 <- ssew/sstw
WeightedR2
```

```
[1] 0.755433
```

The problem of specifying a form of the heteroskedasticity becomes more challenging in the multivariate regression case. Suppose

$$Y_i = \beta_0 + \beta_1 X_{1,i} + \dots + \beta_{K-1,i} X_{K-1,i} + \epsilon_i$$

such that

$$E[\epsilon_i^2 | x_1, \dots, x_{K-1}] = \sigma_i^2 = \sigma^2 \eta_i(x_1, \dots, x_{K-1})$$

for all  $i = 1, 2, \dots, N$ . Any heteroskedasticity is likely to depend on more than one regressor, to different degrees, so there will be parameters to estimate. E.g., we might have something like

$$\eta_i = \exp(\alpha_1 X_{1,i} + \dots + \alpha_{K-1} X_{K-1,i})$$

(the exponentiation is to ensure the variance is positive). Then to implement weighted least squares, we first have to estimate the parameters in the variance equation. One way to do this is to first estimate the main equation using OLS, and obtain the OLS residuals. Then regress the log of the squared residuals on a constant and the regressors to estimate  $\sigma^2$  and the  $\alpha$  parameters, and then finally compute the “fitted variances”  $\widehat{\sigma}_i^2$  and weight the squared residuals by  $1/\widehat{\sigma}_i^2$ .

### 7.3 Testing for Heteroskedasticity

It may be of interest to test whether heteroskedasticity is an issue in the first place. The following are some possible tests. All involve first estimating the main regression by OLS and obtaining the OLS residuals  $\hat{\epsilon}_i$ .

1. Run the regression

$$\hat{\epsilon}_i^2 = \alpha_0 + \alpha_1 X_{1,i} + \dots + \alpha_{K-1} X_{K-1,i} + u_i$$

and test  $H_0 : \alpha_1 = \dots = \alpha_{K-1} = 0$  using an F-test.

2. An alternative is to use an “LM” test after running the regression above: under the null hypothesis, we have

$$NR_\epsilon^2 \stackrel{a}{\sim} \chi_{(K)}^2.$$

3. To allow for possible non-linear forms we can include powers of regressors and interaction terms between them in the variance regression:

$$\begin{aligned} \hat{\epsilon}_i^2 = & \alpha_0 + \alpha_1 X_{1,i} + \dots + \alpha_{K-1} X_{K-1,i} \\ & + \delta_1 X_{1,i}^2 + \dots + \delta_{K-1} X_{K-1,i}^2 \\ & + \gamma_{12} X_{1,i} X_{2,i} + \dots + u_i \end{aligned}$$

then testing if all of the coefficients (not including the intercept) are zero. Obviously you lose degrees of freedom quickly as the number of regressors grow.

4. One way around this problem is to run the regression

$$\hat{\epsilon}_i^2 = \alpha_0 + \alpha_1 \hat{Y}_{i,ols} + \alpha_2 \hat{Y}_{i,ols}^2 + u_i$$

where  $\hat{Y}_{i,ols}$  refers to the OLS fitted values from the main equation. The hypothesis that there is no heteroskedasticity is  $H_0 : \alpha_1 = \alpha_2 = 0$  using an F-tests or an LM test.

The first two is often referred to as Breusch-Pagan tests for heteroskedasticity. The last is referred to as the White test for heteroskedasticity.

**Example 7.4.** We apply the Breusch-Pagan test for heteroskedasticity to the regression

$$\ln(\text{earnings}_i) = \beta_0 + \beta_1 \text{wexp}_i + \beta_2 \text{tenure}_i + \epsilon_i.$$

```
df_earn <- read_excel("data\\earnings.xlsx")      #--Read Data
mdl <- lm(log(earnings)~wexp+tenure, data=df_earn) #--Main Equation
cat("Main Regression\n")                          #--Main Regr Output Title
round(summary(mdl)$coefficients,4)                #--Main Regr Output
```

Main Regression

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	2.4734	0.0980	25.2385	0.0000
wexp	0.0127	0.0060	2.1294	0.0337
tenure	0.0147	0.0041	3.5596	0.0004

```
df_earn$ehat <- residuals(mdl)                  #--Get OLS Residuals
heteq <- lm((ehat^2)~wexp+tenure, data=df_earn)  #--BP-Test Regression
cat("Heteroskedasticity Test Regression\n")      #--Test Regr Output Title
round(summary(heteq)$coefficients,4)            #--Test Regr Output
```

Heteroskedasticity Test Regression

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.5596	0.0927	6.0375	0.0000
wexp	-0.0121	0.0057	-2.1471	0.0322
tenure	-0.0035	0.0039	-0.8877	0.3751

```
## BP, F-version
f_het <- summary(heteq)$fstatistic #--Retrieve F-Stat (stat, df1, df2)
cat("BP-F Stat: ", f_het[1], "      p-val: ", 1-pf(f_het[1], f_het[2], f_het[3]), "\n")
```

BP-F Stat: 3.855297      p-val: 0.02175567

```
## BP, LM-version
lm_het <- nobs(heteq)*summary(heteq)$r.squared # Calc. LM Stat.
lm_pval <- 1 - pchisq(lm_het, 2) # 2 restrictions
cat("BP-LM Stat: ", lm_het, "      p-val: ", lm_pval, "\n", sep="")
```

BP-LM Stat: 7.643914      p-val: 0.02188493

There is some evidence of heteroskedasticity, and the individual t-tests suggest that the noise variance decreases with work experience.

## 7.4 Some Additional Regression Tests

We mention a few other tests associated with regressions.

### 7.4.1 RESET test for functional form misspecification

Given a regression specification

$$Y_i = \beta_0 + \beta_1 X_{1,i} + \cdots + \beta_{K-1} X_{K-1,i} + \epsilon_i,$$

the Regression Equation Specification Error Test (or “RESET Test”) checks if adding powers ( $X_{1,i}^2, X_{2,i}^2, \dots$ ) and interaction terms ( $X_{1,i}X_{2,i}, X_{1,i}X_{3,i}$ , etc.) of the regressors can significantly improve the fit. It is interpreted as a test of adequacy of the *functional form specification* of the original regression, and not as saying anything about the whether or not certain variables should or should not be included. Similar to the White test for heteroskedasticity, the RESET test does this by adding the squares, cubes, and possibly higher powers of the OLS fitted values  $\hat{Y}_{i,ols}$  into the regression specification and tests if these additions have significant explanatory power, i.e., the test equation is

$$Y_i = \beta_0 + \beta_1 X_{1,i} + \dots + \beta_{K-1} X_{K-1,i} + \alpha_2 \hat{Y}_{i,ols}^2 + \dots + \alpha_p \hat{Y}_{i,ols}^p + \epsilon_i, \quad (7.12)$$

and the hypothesis of adequacy of the functional form specification is

$$H_0 : \alpha_2 = \dots = \alpha_p = 0.$$

The test equation cannot include  $\hat{Y}_{i,ols}$  (see exercises), and often only the second or second and third powers are included. An F-test (or t-test, if only the second power is included) can be used to test the hypothesis. We illustrate the RESET test in the next example.

**Example 7.5.** We apply the RESET test to the regression

$$\ln(\text{earnings}_i) = \beta_0 + \beta_1 \text{wexp}_i + \beta_2 \text{tenure}_i + \epsilon_i.$$

using data in `earnings.xlsx`.

```
df_earn <- read_excel("data\\earnings.xlsx")
mdl_base <- lm(log(earnings)~wexp+tenure, data=df_earn)
df_earn$yhat <- fitted(mdl_base)
mdl_test <- lm(log(earnings)~wexp+tenure+I(yhat^2), data=df_earn)
cat("Base Regression:\n")
round(summary(mdl_base)$coefficients, 4)
cat("\nTest Regression:\n")
round(summary(mdl_test)$coefficients, 4)
```

Base Regression:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	2.4734	0.0980	25.2385	0.0000
wexp	0.0127	0.0060	2.1294	0.0337
tenure	0.0147	0.0041	3.5596	0.0004

Test Regression:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	23.4033	8.3837	2.7915	0.0054
wexp	0.2544	0.0970	2.6233	0.0090
tenure	0.3069	0.1171	2.6206	0.0090
I(yhat^2)	-3.4656	1.3881	-2.4967	0.0128

The hypothesis of adequacy of the functional form in the base regression is rejected.

### 7.4.2 Testing Nonnested Alternatives

Regression specifications such as

$$\begin{aligned} \text{[A]} \quad Y_i &= \beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} + \epsilon_i \\ \text{and [B]} \quad Y_i &= \beta_0 + \beta_1 \ln X_{1,i} + \beta_2 \ln X_{2,i} + \epsilon_i \end{aligned}$$

are “non-nested alternatives”, i.e., one is not a special case of the other. One way of testing which specification fits better is to construct a “super-model” that includes both [A] and [B] as restricted cases, i.e.,

$$\text{[A]} \quad Y_i = \beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} + \beta_3 \ln X_{1,i} + \beta_4 \ln X_{2,i} + \epsilon_i$$

and to test for coefficient significance. This approach is often plagued by multicollinearity problems. An alternative is to fit both models separately, collect their fitted values, and include each fitted value series as a regressor in the other specification, i.e., regress

$$\begin{aligned} \text{[A']} \quad Y_i &= \beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} + \delta_1 \hat{Y}_i^B + \epsilon_i \\ \text{and [B']} \quad Y_i &= \beta_0 + \beta_1 \ln X_{1,i} + \beta_2 \ln X_{2,i} + \delta_2 \hat{Y}_i^A + \epsilon_i \end{aligned}$$

and test (separately) if the coefficients on the fitted values are statistically significant. The idea is to see if each specification has anything to add to the other. If  $\delta_1 = 0$  is rejected and  $\delta_2 = 0$  is not, then this suggests that [B] is a better specification (the result does not suggest [B] is the *best* specification, just better than [A].) Likewise, [A] is preferred to [B] if  $\delta_2 = 0$  is rejected and  $\delta_1 = 0$  is not. It may be that both are rejected, which suggests that neither specification is adequate. If neither are rejected, then it appears that there is little in the data to distinguish between the two specifications. Note that the dependent variable in both alternatives must be the same.

We compare the specifications

$$\begin{aligned} \text{[A]} \quad \ln(\text{earnings}_i) &= \beta_0 + \beta_1 \text{wexp}_i + \beta_2 \text{tenure}_i + \epsilon_i \\ \text{and [B]} \quad \ln(\text{earnings}_i) &= \beta_0 + \beta_1 \ln(\text{wexp}_i) + \beta_2 \ln(\text{tenure}_i) + \epsilon_i. \end{aligned}$$

```
mdlA <- lm(log(earnings)~wexp+tenure, data=df_earn)
mdlB <- lm(log(earnings)~log(wexp)+log(tenure), data=df_earn)
df_earn$yhatA <- fitted(mdlA)
df_earn$yhatB <- fitted(mdlB)
cat("Model A plus yhatB:\n")
mdlAplusB <- lm(log(earnings)~wexp+tenure+yhatB, data=df_earn)
round(summary(mdlAplusB)$coefficients,4)
cat("\nModel B plus yhatA:\n")
mdlBplusA <- lm(log(earnings)~log(wexp)+log(tenure)+yhatA, data=df_earn)
round(summary(mdlBplusA)$coefficients,4)
```

Model A plus yhatB:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.1193	0.8700	-0.1371	0.8910

wexp	-0.0030	0.0079	-0.3747	0.7080
tenure	0.0000	0.0064	-0.0024	0.9981
yhatB	1.0607	0.3537	2.9990	0.0028

Model B plus yhatA:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	2.3694	0.9658	2.4534	0.0145
log(wexp)	0.1671	0.0962	1.7375	0.0829
log(tenure)	0.0925	0.0326	2.8386	0.0047
yhatA	-0.0606	0.4121	-0.1471	0.8831

It appears that the specification [B] is preferred over specification [A].

### 7.4.3 Testing for Normality of Noise Terms

The finite sample justification for the t- and F-tests depend on the normality of the noise terms. One way to test this is to use the fact that if a random variable has the normal distribution, then its skewness coefficient is zero (because it is symmetric), and its kurtosis coefficient is three: if  $X \sim \text{Normal}(\mu, \sigma^2)$ , then

$$S = E[(X - \mu)^3]/\sigma^3 = 0$$

$$Kur = E[(X - \mu)^4]/\sigma^4 = 3.$$

The kurtosis coefficient, being the expectation of a fourth moment, emphasizes larger deviations from mean over small deviations from mean (deviations from mean less than one become very small when raised to the fourth power). A kurtosis coefficient greater than 3 suggests higher probability of large deviations from mean, relative to a comparable normally distributed random variable. The skewness and kurtosis coefficients can be estimated using

$$\hat{S} = \frac{\frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})^3}{\left[ \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})^2 \right]^{3/2}}$$

$$\widehat{Kur} = \frac{\frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})^4}{\left[ \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})^2 \right]^2}.$$

The Jarque-Bera statistic applies this idea to regression residuals, using the statistic

$$JB = \frac{N - K}{6} \left( \hat{S}^2 + \frac{1}{4} (\widehat{Kur} - 3)^2 \right)$$

which is approximately  $\chi_{(2)}^2$  in large samples under the null. Some implementations ignore the degree-of-freedom correction and use  $N$  in the numerator instead of  $N - K$ .

We test for normality of the residuals in the regression

$$\ln(\text{earnings}) = \beta_0 + \beta_1 \ln(\text{wexp}_i) + \beta_2 \ln(\text{tenure}_i) + \epsilon_i$$

```
Skew <- function(x){
  # Returns Skewness Coefficient
  return(mean((x-mean(x))^3)/(mean((x-mean(x))^2)^(3/2)))
}
```

```

Kurt <- function(x){
  # Returns Kurtosis Coefficient
  return(mean((x-mean(x))^4)/(mean((x-mean(x))^2)^2))
}

JB <- function mdl){
  # requires lm object, returns JB Stat, p-val, Skewness and Kurtosis Coef.
  N <- nobs(mdl)
  K <- summary(mdl)$df[1]
  ehat <- residuals(mdl)
  JBSkew <- Skew(ehat)
  JBKurt <- Kurt(ehat)
  JBstat <- ((N-K)/6*(JBSkew^2 + (1/4)*(JBKurt-3)^2))
  JBpval <- 1-pchisq(JBstat,2)
  return(list("JBstat"=JBstat,
             "JBpval"=JBpval,
             "Skewness"=JBSkew,
             "Kurtosis"=JBKurt))
}

df_earn <- read_excel("data\\earnings.xlsx")
mdl <- lm(log(earnings)~log(wexp)+log(tenure), data=df_earn)
JBtest <- JB(mdl)
fmt <- function(x){format(round(x,4),nsmall=4)}
cat("JB:", fmt(JBtest$JBstat),
    " p-val:", fmt(JBtest$JBpval),
    " Skewness:", fmt(JBtest$Skewness),
    " Kurtosis:", fmt(JBtest$Kurtosis),"\n")

```

```
JB: 40.8122  p-val: 0.0000  Skewness: 0.4470  Kurtosis: 4.0123
```

The null of normality is rejected. There appears to be a slight skewness to the right, and ‘excess kurtosis’ (kurtosis in excess of 3). A histogram of the OLS residuals is shown below.

```
hist(residuals(mdl), 20)
```

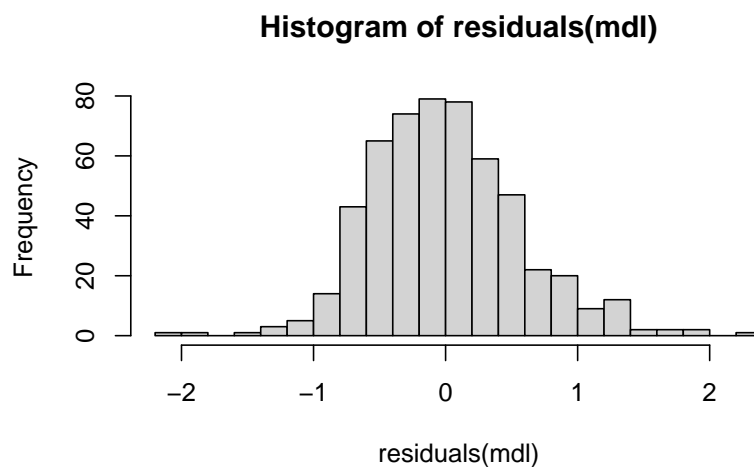


Figure 7.3: Residual histogram.



## 7.5 Exercises

**Exercise 7.1.** Show in Example 7.1 that the modified noise terms are uncorrelated, i.e.,

$$E[\epsilon_i^* \epsilon_j^* | \mathbf{x}] = 0$$

for all  $i \neq j$ ;  $i, j = 1, 2, \dots, N$ .

**Exercise 7.2.** In the notes we claimed that

$$\frac{\sum_{i=1}^N X_i^4}{\left(\sum_{i=1}^N X_i^2\right)^2} \geq \frac{1}{N}.$$

Prove this by showing that for any set of values  $\{z_i\}_{i=1}^N$ , we have

$$N \sum_{i=1}^N z_i^2 - \left(\sum_{i=1}^N z_i\right)^2 \geq 0.$$

(*Hint: start with the fact that  $\sum_{i=1}^N (z_i - \bar{z})^2 \geq 0$ .)* Then substitute  $X_i^2$  for  $z_i$ . When will equality hold?

**Exercise 7.3.**

- Calculate the heteroskedasticity-robust standard errors for the regression in Example 7.4. Compare the robust standard errors with the OLS standard errors.
- Estimate (using `lm()`) the main equation in Example 7.4 using WLS, assuming

$$\text{var}[\epsilon_i | \text{wexp}, \text{tenure}] = \sigma^2 \text{wexp}_i.$$

Compare the WLS estimation results with the OLS estimation results (with heteroskedasticity robust standard errors).

**Exercise 7.4.**

- Why is it that we cannot include  $\hat{Y}_{i,ols}$  in the RESET test equation?
- Add the third power of the OLS fitted value in the RESET test equation in Example 7.5. What happens to the statistical significance of the original regressors? Can you explain the likely cause? *Hint: what is the correlation between  $\hat{y}^2$  and  $\hat{y}^3$ ?*



## Chapter 8

### More Matrix Algebra

We cover four topics in matrix algebra: matrix rank, diagonalization of matrices, differentiation of matrix forms, and vectors and matrices of random variables.

#### 8.1 Rank

##### 8.1.1 A Geometric Viewpoint

We consider vectors and matrices from a geometric perspective, leading up to the concept of the **rank** of a matrix. We view a 2-dimensional vector

$$v_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

as an arrow from the origin to the point  $(x_1, y_1)$ . For the moment, we focus on column vectors. Fig. 8.1 shows three vectors, where  $v_2$  and  $v_3$  are scalar multiples of  $v_1$ .

$$v_1 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, v_2 = -1.5 \begin{bmatrix} 2 \\ 3 \end{bmatrix}, v_3 = 2 \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

```
par(mar=c(4,4,0,0))
v1 <- c(2,3)
v2 <- -1.5*v1
v3 <- 2*v1
plot(NA,xlim=c(-8,8), ylim=c(-7,7), xlab="X", ylab="Y", cex.lab=0.9, cex.axis=0.9)
grid()
arrows(0,0,v1[1],v1[2], lwd=3, length=0.1)
text(v1[1], v1[2]-2, "v1")
arrows(0,0,v2[1],v2[2], lwd=1, length=0.1, col='red')
text(v2[1], v2[2]-1, "-1.5v1", col='red')
arrows(0,0,v3[1],v3[2], lwd=1, length=0.1, col='blue')
text(v3[1]+1.5, v3[2], "2v1", col='blue')
```

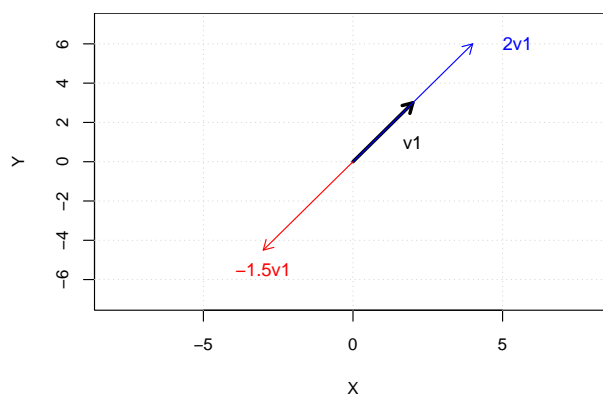


Figure 8.1: Scalar multiple of a vector.

Multiplying a non-zero vector by a scalar changes its length (stretch or shrink). If the scalar is negative, the vector direction is reversed. Other than possibly reversing direction, there is no rotation. If we take the set of *all* vectors of the form  $cv_1$ ,  $c \in \mathbb{R}$ , we get a straight line, which we can think of as a “one-dimensional space” in the two dimensional space made up of all two-dimensional vectors.

Now view a  $(2 \times 2)$  matrix as a collection of two column vectors

$$A = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} & \begin{bmatrix} v_{12} \\ v_{22} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} v_1 & v_2 \end{bmatrix}.$$

Consider linear combinations of the two vectors  $v_1$  and  $v_2$ :

$$c_1 v_1 + c_2 v_2 = \begin{bmatrix} v_1 & v_2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = Ac.$$

The result is a third vector that is a diagonal of the parallelogram formed by  $c_1 v_1$  and  $c_2 v_2$ . This is illustrated in Fig. 8.2 for  $c_1 = 2$  and  $c_2 = 1.5$ .

$$v_1 = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, v_2 = \begin{bmatrix} 4 \\ 1 \end{bmatrix}, v_3 = 2v_1 + 1.5v_2 = \begin{bmatrix} 8 \\ 7.5 \end{bmatrix}.$$

```
par(mar=c(4,4,0,0))
plot(c(-1,9), c(-1,9), type="n", xlab="X", ylab="Y", asp=1, cex.lab=0.9, cex.axis=0.9)
grid()
v1 <- c(1,3); v2 <- c(4,1); v3 <- 2*v1+1.5*v2
arrows(0,0,v1[1],v1[2], lwd=3, length=0.1); text(v1[1]-0.5, v1[2]+0.5, "v1")
arrows(0,0,v2[1],v2[2], lwd=3, length=0.1); text(v2[1], v2[2]-0.55, "v2")
arrows(0,0,2*v1[1],2*v1[2], lwd=1, length=0.1, col='red');
text(2*v1[1]-0.5, 2*v1[2]+0.5, "2v1", col='red')
arrows(0,0,1.5*v2[1],1.5*v2[2], lwd=1, length=0.1, col='red')
text(1.5*v2[1], 1.5*v2[2]-0.55, "1.5v2", col='red')
arrows(0,0,v3[1],v3[2], lwd=3, length=0.1); text(v3[1]-0.5, v3[2]+0.5, "v3")
segments(2*v1[1],2*v1[2],v3[1],v3[2], lwd=1, lty=2, col='red')
segments(1.5*v2[1],1.5*v2[2],v3[1],v3[2], lwd=1, lty=2, col='red')
```

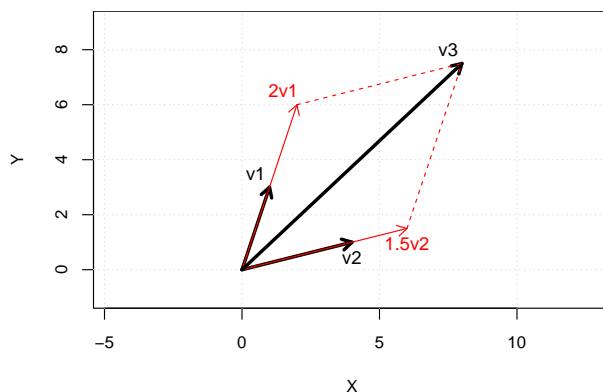


Figure 8.2: Scalar multiple of a vector.

Now we consider, given two vectors  $v_1$  and  $v_2$ , the set of **all** linear combinations of the form  $c_1v_1 + c_2v_2$ , where  $c_1 \in \mathbb{R}$  and  $c_2 \in \mathbb{R}$ . We consider in particular the following cases:

- Case 1:  $v_1$  and  $v_2$  are non-zero vectors, and it is not the case that  $v_1 = bv_2$ . The example in Fig. 8.2 illustrates this case. In such situations, the set of all linear combinations  $c_1v_1 + c_2v_2$  fills the entire 2-d space. Put differently, given any vector in the  $x$ - $y$  plane, you can find  $c_1$  and  $c_2$  such that that vector is equal to  $c_1v_1 + c_2v_2$ . We say  $v_1$  and  $v_2$  **spans** the entire 2-d space, and that the matrix  $A = \begin{bmatrix} v_1 & v_2 \end{bmatrix}$  has **column rank** two. Since the column rank of  $A$  is equal to the number of columns in  $A$ , we also say  $A$  has **full column rank**.
- Case 2:  $v_1 = bv_2$  (i.e.,  $v_1$  and  $v_2$  lie on the same line) or if one of the vectors is a zero vector and the other is not. In this case, then the set of all possible linear combinations  $c_1v_1 + c_2v_2$  will be the line coincident with  $v_1$  and  $v_2$ , or the line coincident with the non-zero vector. The vectors  $v_1$  and  $v_2$  do not span the entire (two-dimensional) space, but span only a (one-dimensional) line. The matrix  $A = \begin{bmatrix} v_1 & v_2 \end{bmatrix}$  will take the form

$$A = \begin{bmatrix} v_{11} & bv_{11} \\ v_{21} & bv_{21} \end{bmatrix}, \begin{bmatrix} v_{11} & 0 \\ v_{21} & 0 \end{bmatrix} \text{ or } \begin{bmatrix} 0 & v_{12} \\ 0 & v_{22} \end{bmatrix}$$

We say the matrix  $A$  has column rank one.

- Case 3:  $v_1 = v_2 = 0$ . In this case, all linear combinations  $c_1v_1 + c_2v_2$  result in the zero vector. We say that the matrix

$$A = \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

has column rank 0.

Notice that in cases 2 and 3, we have

$$c_1v_1 + c_2v_2 = 0 \text{ for some } (c_1, c_2) \neq (0, 0), \text{ or } Ac = 0 \text{ for some } c \neq 0 \quad (8.1)$$

whereas in case 1

$$c_1v_1 + c_2v_2 \neq 0 \text{ for all } (c_1, c_2) \neq (0, 0), \text{ i.e., } Ac \neq 0 \text{ for all } c \neq 0. \quad (8.2)$$

We say the vectors are **linearly dependent** if (8.1) holds, and **linearly independent** if (8.2) holds.

We can view a  $(2 \times m)$  matrix,  $m > 2$ , as a collection of  $m$  2-d vectors. These are vectors ‘living’ in 2-d space. Again, we consider the set of all linear combinations of the form

$$c_1v_1 + c_2v_2 + \cdots + c_mv_m.$$

Depending on the values of the vectors, the vectors might span the entire 2-d space, or only a single (1-d) line, or just the origin if the vectors are all zero-vectors. However, 2-d vectors cannot span a three (or more) dimensional space, no matter how many of them there are. In other words, a  $(2 \times m)$  matrix  $A$ , with  $m > 2$ , may have column rank 2, 1, or 0, but cannot have column rank greater than 2. Put yet another way, a set of three or more 2-d vectors must

be linearly dependent. We will be able to write each one as some combination of the others.

We view a 3-dimensional vector

$$v_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

as an arrow in a three-dimensional cartesian space from the origin to the point  $(x_1, y_1, z_1)$ . The diagram below illustrates<sup>1</sup> the vector

$$v_1 = \begin{bmatrix} 3 \\ 2 \\ 2 \end{bmatrix}.$$

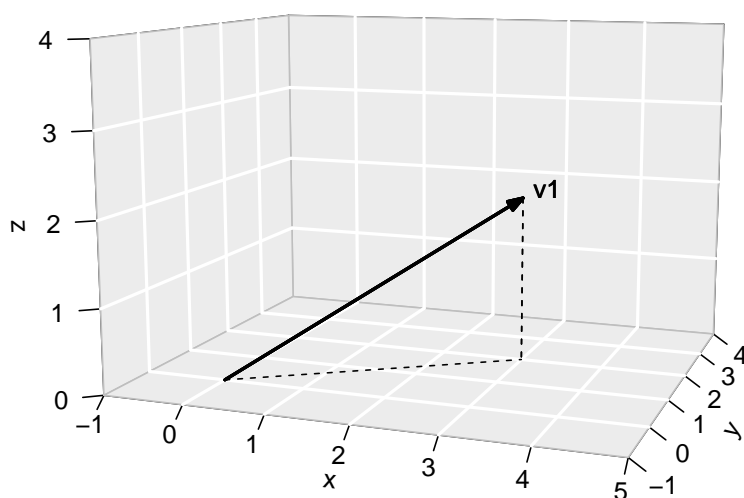


Figure 8.3: A 3-d vector.

Multiplying a vector by a scalar changes its length, but not the direction, except if the scalar is negative, in which case the vector direction is flipped. All vectors  $cv_1$  for any scalar  $c$  will lie on a single line.

View a  $(3 \times 2)$  matrix as a collection of two 3-d column vectors

$$A = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \\ v_{31} & v_{32} \end{bmatrix} = \left[ \begin{bmatrix} v_{11} \\ v_{21} \\ v_{31} \end{bmatrix} \quad \begin{bmatrix} v_{12} \\ v_{22} \\ v_{32} \end{bmatrix} \right] = [v_1 \quad v_2]$$

Consider linear combinations of the two 3-d vectors  $v_1$  and  $v_2$ :

$$c_1 v_1 + c_2 v_2.$$

The result is a third vector that is a diagonal of the parallelogram formed by  $c_1 v_1$  and  $c_2 v_2$ .

---

<sup>1</sup>The code uses the package `plot3D`. The code for the figures are not shown, to avoid distracting from the main discussion.

This is illustrated in Fig. 8.4, which shows two vectors

$$v_1 = \begin{bmatrix} 3 \\ 2 \\ 2 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 0 \\ 4 \\ 3 \end{bmatrix}$$

and their sum.

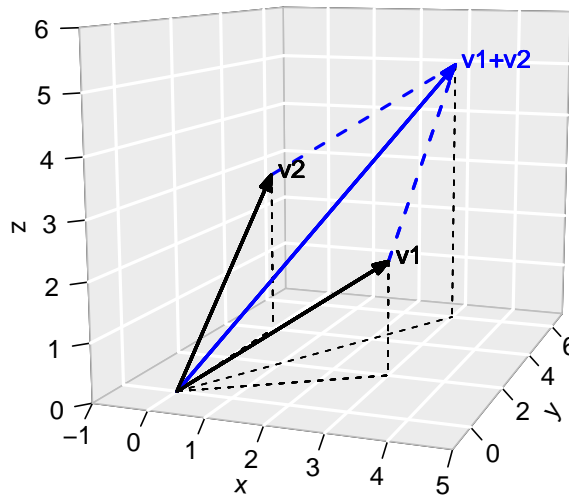


Figure 8.4: Linear combination of two 3-d vectors.

The three vectors  $v_1$ ,  $v_2$  and  $v_1 + v_2$  all lie on the same plane. In fact, if we consider the set of **all** linear combinations of  $v_1$  and  $v_2$

$$c_1 v_1 + c_2 v_2$$

we will find that this set makes up the entire plane containing  $v_1$  and  $v_2$ . We say that these two 3-d vectors span the (2-dimensional) plane just described.

If the two vectors  $v_1$  and  $v_2$  are such that  $v_2 = c v_1$ , then the two vectors lie on a line, and their linear combinations would also lie on the same line:

$$\text{if } v_1 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}, v_2 = \begin{bmatrix} 2 \\ 4 \\ 4 \end{bmatrix}, \quad \text{then } v_3 = c_1 v_1 + c_2 v_2 = (c_1 + 2c_2) \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}.$$

Fig. 8.5 shows the vectors  $v_1$ ,  $v_2$  and  $v_1 + v_2$ . The set of linear combinations of  $v_1$  and  $v_2$  spans a (one-dimensional) line, not a two-dimensional plane. The same is true if one of the vectors is the zero vector, and the other is non-zero. If both are zero vectors, then the linear combinations are always just the zero vector. We say that the  $(3 \times 2)$  matrix

$$A = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \\ v_{31} & v_{32} \end{bmatrix}$$

has **column rank** two if its columns span a plane. It has column rank one if its columns span

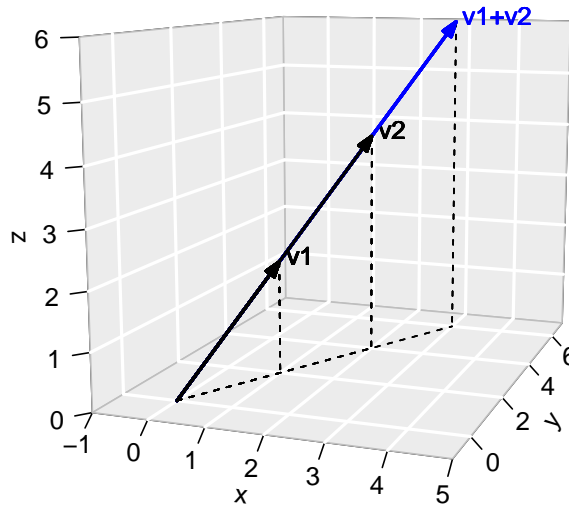


Figure 8.5: A one-dimensional line spanned by two vectors.

a line, and it has column rank zero if both columns are zero vectors.

Linear combinations of two 3-dimensional vectors, of course, cannot span the entire space – the highest dimensional space it can span is a two-dimensional plane. To span the entire three dimensional space, we need three vectors, and these three vectors cannot all lie on a plane or a line, and none of the vectors can be the zero vector. We say that the  $(3 \times 3)$  matrix

$$A = \begin{bmatrix} v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ v_{31} & v_{32} & v_{33} \end{bmatrix} = [v_1 \quad v_2 \quad v_3]$$

has column rank three, or full column rank, if the three column vectors that make up the matrix span the entire three dimensional space.

If the three column vectors span only 2-dimensions or less, then the three vectors all lie on the same plane or on a line, or are all zero. A compact way to describe these situations as a whole is that there is some  $(c_1, c_2, c_3) \neq (0, 0, 0)$  such that

$$c_1 v_1 + c_2 v_2 + c_3 v_3 = 0,$$

that is, there is some  $c \neq 0$  such that  $Ac = 0$ . If the vectors are non-zero and all lie on a plane or a line, then one can be written as a linear combination of the other, thus

$$v_i = c_j v_j + c_k v_k \quad \text{where } i, j, k = 1, 2, 3, \quad \text{with } i \neq j \neq k \neq i, \quad (8.3)$$

or  $v_i - c_j v_j - c_k v_k = 0$ . If one is a zero vector, say,  $v_1$ , then we can write  $v_1 = 0v_2 + 0v_3$ , or  $v_i - 0v_j - 0v_k = 0$ . In all of these cases we have found  $c \neq 0$  such that  $Ac = 0$ . In these cases we say that the vectors  $v_1, v_2$  and  $v_3$  are linearly dependent. On the other hand, if the three vectors span the entire 3-d space, then we cannot write one as a linear combination of the others. An expression like (8.3) will hold only when  $c = 0$ , i.e.,  $Ac \neq 0$  for all  $c \neq 0$ . In this case we say that the vectors are linearly independent.



What if we have a  $(3 \times n)$  matrix where  $n > 3$ ? These vectors that make up the matrix may span the entire 3-d space (the matrix has rank 3), a 2-d plane (the matrix has column rank 2), or a 1-d line (the matrix has column rank 1). If all of the vectors are zero vectors, the matrix has column rank zero. Three-dimensional vectors, of course, cannot span a space of dimension greater than three. The column rank of an  $(3 \times n)$  matrix cannot exceed 3.

We lose the literal geometric viewpoint once we enter the realm of higher-dimensional vectors, but the geometric intuition carries over. For instance, the two vectors

$$v_1 = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 3 \end{bmatrix}, v_2 = \begin{bmatrix} 2 \\ 4 \\ 4 \\ 6 \end{bmatrix},$$

span only a 1-d “line” (in 4-d space). Since  $v_2 = 2v_1$ , every linear combination is also just a multiple of  $v_1$ :

$$c_1 v_1 + c_2 v_2 = c_1 v_1 + 2c_2 v_1 = (c_1 + 2c_2)v_1.$$

The matrix

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 2 & 4 \\ 3 & 6 \end{bmatrix}$$

therefore only has column rank one. On the other hand, the columns of the matrix

$$B = \begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 2 & 4 \\ 3 & 5 \end{bmatrix}$$

span a 2-d “plane” in 4-d space; its column rank is two. A  $(4 \times 3)$  matrix can have column ranks 0, 1, 2, or 3. The following matrices have column ranks 2 and 3 respectively:

$$C = \begin{bmatrix} 1 & 3 & 6 \\ 2 & 2 & 8 \\ 3 & 1 & 10 \\ 4 & 1 & 13 \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 3 & 5 & 6 \end{bmatrix}.$$

A  $(4 \times 4)$  matrix can have column rank 0 to 4. A  $(4 \times n)$  matrix, where  $n > 4$  can have column ranks 0 up to 4. In general, the rank of an  $(m \times n)$  matrix cannot exceed the minimum of  $m$  and  $n$ :

$$\text{col.rank}(A) \leq \min(m, n).$$

If  $\text{col.rank}(A) = n$ , then  $Ac \neq 0$  for all  $c \neq 0$ . If  $\text{col.rank}(A) < n$ , then there is a  $c \neq 0$  such that  $Ac = 0$ .

### 8.1.2 The Rank of a Matrix

We could have done all this by treating a matrix as a collection of row vectors. For example,

$$A = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \\ v_{31} & v_{32} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} v_{11} & v_{12} \end{bmatrix} \\ \begin{bmatrix} v_{21} & v_{22} \end{bmatrix} \\ \begin{bmatrix} v_{31} & v_{32} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \mathbf{v}_3^T \end{bmatrix}$$

Linear combinations of the component row vectors can be written  $c^T A$ . We can talk about the linear dependence or linear independence of the row vectors, and the dimension of the space spanned by the vectors. All this leads to the concept of **row rank**. In general, the rank of an  $(m \times n)$  matrix cannot exceed the minimum of  $m$  and  $n$ :

$$\text{row.rank}(A) \leq \min(m, n).$$

If  $\text{row.rank}(A) = m$ , then  $c^T A \neq 0$  for all  $c \neq 0$ .

Finally, we note that the row and column ranks of a matrix are always equal. Suppose the column rank of a matrix  $A$  is  $r \leq \min(m, n)$ . This means you can find  $r$  linearly independent columns in  $A$ . Collect these columns into the  $(m \times r)$  matrix  $C$ . Since every column in  $A$  can be written as a linear combination of the columns of  $C$ , we can write  $A = CR$  for some  $(r \times n)$  matrix  $R$ . This in turn says that every row in  $A$  can be written as a linear combination of the rows of  $R$ . Since there are only  $r$  rows in  $R$ , it must be that the row rank of  $A$  is less than or equal to  $r$ . That is,

$$\text{row.rank}(A) \leq \text{col.rank}(A). \quad (8.4)$$

Applying the same argument to  $A^T$ , we get

$$\text{row.rank}(A^T) \leq \text{col.rank}(A^T).$$

Since the row rank of a transpose is the column rank of the original matrix, we have

$$\text{col.rank}(A) \leq \text{row.rank}(A). \quad (8.5)$$

Inequalities (8.4) and (8.5) imply

$$\text{row.rank}(A) = \text{col.rank}(A).$$

We can therefore speak unambiguously of the **rank** of a matrix  $A$ .

We state without proof a few results regarding the rank of a matrix:

- For any matrices  $A$  and  $B$  for which  $AB$  exists, we have

$$\text{rank}(AB) \leq \min(\text{rank}(A), \text{rank}(B)).$$

- A square matrix  $A$  has an inverse if and only if it is full rank.
- If  $A$  is a full rank  $(m \times m)$  matrix, and  $B$  is an  $(m \times n)$  matrix of rank  $r$ , then  $\text{rank}(AB) = r$ .

- For any  $(n \times k)$  matrix  $A$ , we have

$$\text{rank}(A^T A) = \text{rank}(A).$$

This means that  $A^T A$  will have an inverse if and only if  $A$  has full column rank.

### 8.1.3 Finding the Rank of a Matrix in R

To find the rank of a matrix in R, we can feed the matrix into the `qr()` function, which returns a number of things, including the matrix's rank.

**Example 8.1.** Use R to find the rank of the matrix

$$E = \begin{bmatrix} 2 & 2 & 4 \\ 2 & 1 & 3 \\ 2 & 5 & 7 \end{bmatrix}.$$

```
E <- matrix(c(2,2,2,2,1,5,4,3,7),3,3)
E_qr <- qr(E)
E_qr$rank
```

```
[1] 2
```

### 8.1.4 Exercises

**Exercise 8.1.** The following matrices have rank one:

$$A = \begin{bmatrix} a_{11} & \alpha a_{11} \\ a_{21} & \alpha a_{21} \end{bmatrix}, B = \begin{bmatrix} a_{11} & 0 \\ a_{21} & 0 \end{bmatrix}, C = \begin{bmatrix} 0 & a_{12} \\ 0 & a_{22} \end{bmatrix}.$$

In each case find a non-zero  $(2 \times 1)$  vector  $c$  such that  $Ac = 0$ . Show that in each case the determinant of the matrix is zero (so that the inverse does not exist).

**Exercise 8.2.** Suppose the matrix

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

is such that  $a_{11}a_{22} - a_{12}a_{21} \neq 0$ . Show that it cannot be that

- one or more of the rows or columns have all zero entries;
- one of the rows / columns is a multiple of the other row / column.
- Show that the only  $c$  such that  $Ac = 0$  is  $c = 0$ .

**Exercise 8.3.** What is the rank of the following matrices? (Try without using R.)

$$\text{i. } A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{ii. } B = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 1 & 7 \\ 1 & 0 & 2 \end{bmatrix}$$

**Exercise 8.4.** Let  $\{X_i\}_{i=1}^N$  be a set of numbers, and consider the matrix

$$A = \begin{bmatrix} N & \sum_{i=1}^N X_i \\ \sum_{i=1}^N X_i & \sum_{i=1}^N X_i^2 \end{bmatrix}$$

Show that  $A$  is rank one if and only if  $X_i$  is constant over all  $i$ , i.e.,  $X_i = b$ ,  $i = 1, 2, \dots, N$ .

**Exercise 8.5.** Suppose  $A$  is a  $(n \times n)$  diagonal matrix with  $r$  non-zero terms and  $n - r$  zero terms in its diagonal. What is its rank?

## 8.2 Diagonalization of Symmetric Matrices

A square matrix  $A$  is **diagonalizable** if it can be written in the form

$$A = C\Lambda C^{-1} \quad \text{where } \Lambda \text{ is diagonal.} \quad (8.6)$$

For such matrices, we have  $C^{-1}AC = \Lambda$ . We will not prove this statement, except to note that the diagonal elements of  $\Lambda$  are the *eigenvalues* of  $A$ , and the columns of  $C$  are the corresponding *eigenvectors* of  $A$ . The decomposition (8.6) is called the *eigen-decomposition* of the matrix  $A$ .

If the matrix  $A$  is symmetric, then we have in addition that  $C^{-1} = C^T$ , i.e., a symmetric matrix  $A$  can be written as

$$A = C\Lambda C^T \quad \text{where } C^T C = C C^T = I \text{ and } \Lambda \text{ is diagonal.} \quad (8.7)$$

We say that  $A$  is **orthogonally diagonalizable**. Symmetric matrices are the only such matrices, i.e., a square matrix is orthogonally diagonalizable if and only if it is symmetric. Furthermore, the values of  $C$  and  $\Lambda$  will be real; if  $A$  is not symmetric, the eigenvalues and eigenvectors may be complex-valued.

If (8.7) holds for the  $(n \times n)$  symmetric matrix  $A$ , we can write

$$A = C\Lambda C^T = \begin{bmatrix} c_1 & c_2 & \dots & c_n \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \begin{bmatrix} c_1^T \\ c_2^T \\ \dots \\ c_n^T \end{bmatrix} = \sum_{i=1}^n \lambda_i c_i c_i^T.$$

where  $c_1, c_2, \dots, c_n$  are the  $(n \times 1)$  columns of  $C$ , and  $\lambda_1, \lambda_2, \dots, \lambda_n$  are the diagonal elements of  $\Lambda$ .

A matrix  $C$  such that  $C^T C = I$  is called orthogonal (sometimes orthonormal) because

$$C^T C = \begin{bmatrix} c_1^T \\ c_2^T \\ \dots \\ c_n^T \end{bmatrix} \begin{bmatrix} c_1 & c_2 & \dots & c_n \end{bmatrix} = \begin{bmatrix} c_1^T c_1 & c_1^T c_2 & \dots & c_1^T c_n \\ c_2^T c_1 & c_2^T c_2 & \dots & c_2^T c_n \\ \vdots & \vdots & \ddots & \vdots \\ c_n^T c_1 & c_n^T c_2 & \dots & c_n^T c_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}.$$

In other words, the columns of  $C$  are unit length vectors, and orthogonal to each other.

**Example 8.2.** For the matrix

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 4 \\ 1 & 4 & 1 \end{bmatrix}.$$

the matrices  $C$  and  $\Lambda$  are

$$C \approx \begin{bmatrix} -0.1437 & 0.9688 & 0.2021 \\ -0.7331 & -0.2414 & 0.6359 \\ -0.6648 & 0.0568 & -0.7449 \end{bmatrix}, \Lambda \approx \begin{bmatrix} 5.6272 & 0 & 0 \\ 0 & 1.0586 & 0 \\ 0 & 0 & -2.6858 \end{bmatrix}$$

which we can find from the following code

```
A=matrix(c(1,0,1,0,2,4,1,4,1), nrow=3)
E=eigen(A)
C <- E$vectors # E$vectors is C
Lambda <- diag(E$values) # E$values is diag values of \Lambda

cat("C = \n"); round(C,4)

C =
      [,1]      [,2]      [,3]
[1,] -0.1437  0.9688  0.2021
[2,] -0.7331 -0.2414  0.6359
[3,] -0.6648  0.0568 -0.7449

cat("\nLambda = \n"); round(Lambda,4)

Lambda =
      [,1]      [,2]      [,3]
[1,] 5.6272 0.0000 0.0000
[2,] 0.0000 1.0586 0.0000
[3,] 0.0000 0.0000 -2.6858

cat("\nVerifying A = C Lambda t(C): \n"); round(C %*% Lambda %*% t(C), 4)

Verifying A = C Lambda t(C):
      [,1] [,2] [,3]
[1,] 1 0 1
[2,] 0 2 4
[3,] 1 4 1

cat("\nVerifying C %*% t(C) = I: \n"); round(C %*% t(C), 4)

Verifying C %*% t(C) = I:
      [,1] [,2] [,3]
[1,] 1 0 0
[2,] 0 1 0
[3,] 0 0 1
```

The columns of  $C$  are the eigenvectors, and the diagonal elements of  $\Lambda$  are the eigenvalues of  $A$ .

The diagonalization result (8.6) has many applications:

- Suppose you want to find the 100th power of  $A$ . Since  $A = C\Lambda C^{-1}$ , we have

$$A = C\Lambda C^{-1}C\Lambda C^{-1} \dots C\Lambda C^{-1} = C\Lambda^{100}C^{-1}.$$

This is a much more efficient (and accurate) way of computing large powers of matrices than brute force multiplication.

- Since  $AC = C\Lambda$  and  $C$  is full rank,  $A$  and  $\Lambda$  have the same rank. Since  $\Lambda$  is diagonal, its rank is just the number of non-zero elements in its diagonal. Therefore the rank of  $A$  is the number of non-zero elements in the diagonal of  $\Lambda$ .
- The determinant of the product of square matrices is the product of their determinants. Since  $AC = C\Lambda$ , we have  $|A||C| = |C||\Lambda|$ , from which it follows that  $|A| = |\Lambda|$ . Furthermore, the determinant of a diagonal matrix is just the product of the diagonal elements.
- Recall that a  $(n \times n)$  matrix  $A$  is positive definite if  $c^T A c > 0$  for all vectors  $c \neq 0$ . If  $A$  is symmetric, so that (8.7) holds, then we have

$$c^T A c = c^T C C^T A C C^T c = b^T \Lambda b = \sum_{i=1}^n b_i^2 \lambda_i$$

where  $b = C^T c$ . Since  $C^T$  is invertible, it has full rank, which means that  $b = C^T c \neq 0$  for all  $c \neq 0$ , and it follows that  $\sum_{i=1}^n b_i^2 \lambda_i > 0$  if  $\lambda_i > 0$  for all  $i$ . If  $\lambda_i \leq 0$  for one or more  $i$ , then we can find  $b \neq 0$  (and therefore a  $c \neq 0$ ) such that  $\sum_{i=1}^n b_i^2 \lambda_i \leq 0$ . In other words,  $A$  is positive definite if and only if the diagonal elements of  $\Lambda$  are positive.

- Since the diagonal elements of  $\Lambda$  are positive if  $A$  is symmetric and positive definite, we can write  $\Lambda = \Lambda^{1/2} \Lambda^{1/2}$ . The matrix  $\Lambda^{1/2}$  is the diagonal matrix whose diagonal elements are the square root of the diagonal elements of  $\Lambda$ . Then we have

$$A = C\Lambda^{1/2}\Lambda^{1/2}C^T \quad \text{or} \quad \Lambda^{-1/2}C^T A C\Lambda^{-1/2} = I.$$

If we let  $P = \Lambda^{-1/2}C^{-1}$ , then we can write

$$PAP^T = I \quad \text{or} \quad A = P^{-1}(P^T)^{-1} = P^{-1}(P^{-1})^T \quad \text{or} \quad A^{-1} = P^T P.$$

*What are these eigenvalues and eigenvectors?* Think of a square matrix  $A$  as something that transforms one vector into another, i.e.,  $Ax_1 = x_2$ . In general, the new vector  $x_2$  will have a different length and a different direction from  $x_1$ , i.e., in general there will be scaling and rotation. However, for any given matrix  $A$ , there will be certain vectors  $x$  such that

$$Ax = \lambda x \tag{8.8}$$

where  $\lambda$  is a scalar. Recall that a scalar multiple of a vector only scales the vector, and reverses its direction if the scalar is negative. What (8.8) says is that  $Ax$  only stretches or shrinks the vector  $x$ , without rotation (apart from possibly reversing the direction). Vectors  $x$  for which (8.8) holds are called the eigenvectors of  $A$ , and the corresponding  $\lambda$ s are the eigenvalues.

## 8.2.1 Exercises

**Exercise 8.6.** A  $(n \times n)$  matrix  $A$  is **positive semidefinite** if  $c^T A c \geq 0$  for all vectors  $c \neq 0$ . Explain why a symmetric matrix  $A$  is positive semidefinite if and only if  $\lambda_i$  is non-negative for all  $i$ .

**Exercise 8.7.** A square matrix  $A$  is said to be **idempotent** if  $AA = A$ . For example, suppose  $X$  is  $(N \times K)$  such that the  $(K \times K)$  matrix  $X^T X$  has an inverse (i.e.,  $X$  has full column rank). Then the matrix  $A = X(X^T X)^{-1} X^T$  is idempotent, since

$$AA = X(X^T X)^{-1} X^T X (X^T X)^{-1} X^T = X(X^T X)^{-1} X^T = A.$$

Now suppose that  $A$  is idempotent *and symmetric*, and consider the diagonalization of this matrix:

$$C^T A C = \Lambda.$$

- Explain why the diagonal elements of  $\Lambda$  can only take values 1 and 0. (*Hint: We have  $AC = C\Lambda$ , therefore  $AAC = AC\Lambda = C\Lambda^2$ . Since  $AA = A$ , we have  $AC = C\Lambda^2$ . Since  $C\Lambda = C\Lambda^2$  and  $C$  is full rank, therefore  $\Lambda = \Lambda^2$ .)*
- Show that the rank of a symmetric idempotent matrix  $A$  is equal to its trace. (*Hint:  $\text{tr}(A) = \text{tr}(C\Lambda C^T) = \text{tr}(C^T C \Lambda) = \text{tr}(\Lambda)$ .*)

## 8.3 Differentiation of Matrix Forms

## 8.3.1 Definitions

This topic is easier than it sounds. What we mean by differentiation of matrix forms is merely a set of formulas that organize partial derivatives of functions written with matrices. For instance, for the function  $y = f(x_1, x_2, \dots, x_n)$ , which we write as  $f(x)$  where  $x^T = [x_1 \ x_2 \ \dots \ x_n]$ , we *define* the derivative of  $y$  with respect to the vector  $x$  to be

$$\frac{dy}{dx} = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_n} \end{bmatrix} \quad \text{and} \quad \frac{dy}{dx^T} = \begin{bmatrix} \frac{\partial y}{\partial x_1} & \frac{\partial y}{\partial x_2} & \dots & \frac{\partial y}{\partial x_n} \end{bmatrix}.$$

If  $y = f(x_1, x_2, x_3, x_4)$  and if

$$X = \begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix},$$

then we define

$$\frac{dy}{dX} = \begin{bmatrix} \frac{\partial y}{\partial x_1} & \frac{\partial y}{\partial x_2} \\ \frac{\partial y}{\partial x_3} & \frac{\partial y}{\partial x_4} \end{bmatrix}.$$

Likewise if  $X$  is an  $(m \times n)$  matrix.

If  $y$  is an  $m$  vector of multivariable functions

$$y = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_m(x) \end{bmatrix}$$

where  $x^T = [x_1 \ x_2 \ \dots \ x_n]$ , then we define

$$\frac{dy}{dx^T} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}.$$

**Example 8.3.** Let

$$X = \begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix}$$

and  $y = |X| = x_1x_4 - x_2x_3$ . Then

$$\frac{dy}{dX} = \begin{bmatrix} \frac{\partial y}{\partial x_1} & \frac{\partial y}{\partial x_2} \\ \frac{\partial y}{\partial x_3} & \frac{\partial y}{\partial x_4} \end{bmatrix} = \begin{bmatrix} x_4 & -x_3 \\ -x_2 & x_1 \end{bmatrix}.$$

Since

$$X^{-1} = \frac{1}{|X|} \begin{bmatrix} x_4 & -x_2 \\ -x_3 & x_1 \end{bmatrix}$$

we have the formula

$$\frac{d}{dX}|X| = |X|(X^{-1})^T.$$

This result holds for general non-singular square matrices (proof omitted).

### 8.3.2 Basic Differentiation Formulas

If  $y = f(x) = Ax$  where  $A = (a_{ij})_{mn}$  is a  $(m \times n)$  matrix of constants and  $x = [x_1 \ x_2 \ \dots \ x_n]^T$  is an  $(n \times 1)$  vector of variables, then

$$\frac{dy}{dx^T} = A. \quad (8.9)$$

*Proof:* The  $i$ th element of  $Ax$  is  $\sum_{k=1}^n a_{ik}x_k$ . Therefore the  $(i, j)$ th element of  $\frac{dy}{dx^T}$  is

$$\frac{\partial}{\partial x_j} \sum_{k=1}^n a_{ik}x_k = a_{ij}$$

which says that  $\frac{dy}{dx^T} = A$ . Result (8.9) is the matrix analogue of the univariate differentiation rule  $\frac{d}{dx}ax = a$ .



If  $y = f(x) = x^T A x$  where  $A = (a_{jk})_{nn}$  is a  $(n \times n)$  matrix of constants, then

$$\frac{dy}{dx} = \frac{d}{dx} x^T A x = (A + A^T)x. \quad (8.10)$$

*Proof:*  $y = x^T A x = \sum_{j=1}^n \sum_{k=1}^n a_{jk} x_j x_k$ . The derivative  $\frac{dy}{dx}$  is the  $(n \times 1)$  vector whose  $i$ th element is

$$\frac{d}{dx_i} \sum_{j=1}^n \sum_{k=1}^n a_{jk} x_j x_k = \sum_{k=1}^n a_{ik} x_k + \sum_{j=1}^n a_{ji} x_j.$$

The first sum after the equality is the product of the  $i$ th row of  $A$  into  $x$ . The second sum after the inequality is the product of the  $i$ th row of  $A^T$  into  $x$ . In other words,  $\frac{dy}{dx} = (A + A^T)x$ .

It may be helpful to derive this formula by direct differentiation in a special case, say, where  $A$  is  $(3 \times 3)$ . You are asked to do this in an exercise. Note that if  $A$  is symmetric, then (8.10) becomes

$$\frac{dy}{dx} = \frac{d}{dx} x^T A x = (A + A^T)x = 2Ax. \quad (8.11)$$

The result then becomes directly comparable to the univariate differentiation rule  $\frac{d}{dx} ax^2 = 2ax$ .

If  $y = f(x)$  is a scalar valued function of an  $(n \times 1)$  vector of variables, then

$$\frac{d}{dx^T} \left( \frac{dy}{dx} \right) = \frac{d}{dx^T} \begin{bmatrix} \frac{dy}{dx_1} \\ \frac{dy}{dx_2} \\ \vdots \\ \frac{dy}{dx_n} \end{bmatrix} = \begin{bmatrix} \frac{d^2 y}{dx_1^2} & \frac{d^2 y}{dx_1 dx_2} & \cdots & \frac{d^2 y}{dx_1 dx_n} \\ \frac{d^2 y}{dx_2 dx_1} & \frac{d^2 y}{dx_2^2} & \cdots & \frac{d^2 y}{dx_2 dx_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{d^2 y}{dx_n dx_1} & \frac{d^2 y}{dx_n dx_2} & \cdots & \frac{d^2 y}{dx_n^2} \end{bmatrix}. \quad (8.12)$$

In other words, we get the Hessian matrix of  $y$ . We write  $\frac{d}{dx^T} \left( \frac{dy}{dx} \right)$  as

$$\frac{d}{dx^T} \left( \frac{dy}{dx} \right) = \frac{d^2 y}{dx dx^T}.$$

### 8.3.3 Exercises

**Exercise 8.8.** Show that if  $y = f(x) = x^T A$  where  $A = (a_{ij})_{mn}$  is a  $(m \times n)$  matrix of constants and  $x = [x_1 \ x_2 \ \dots \ x_m]^T$  is an  $(m \times 1)$  vector of variables, then

$$\frac{dy}{dx} = A.$$

**Exercise 8.9.** If  $c$  is an  $n$ -dimensional vector of constants and  $x$  is an  $n$ -dimensional vector of variables, show that

$$\frac{d}{dx} c^T x = c.$$

**Exercise 8.10.** Let  $A = (a_{ij})_{33}$  be a  $(3 \times 3)$  matrix of constants, and  $x$  be a  $(3 \times 1)$  vector of

variables. Multiply out  $x^T A x$  in full, and show by direct differentiation that

$$\frac{dy}{dx} = (A + A^T)x.$$

**Exercise 8.11.** Use the fact that  $\frac{d}{dX}|X| = |X|(X^{-1})^T$  for a general square matrix  $X$  to show that if  $|X| > 0$ , then

$$\frac{d}{dX} \ln |X| = (X^{-1})^T.$$

**Exercise 8.12.** Show that if  $A = (a_{ij})_{nn}$  is an  $(n \times n)$  matrix of constants and  $x$  is an  $n$ -dimensional vector of variables, then

$$\frac{d^2(x^T A x)}{dx dx^T} = A + A^T.$$

**Exercise 8.13.** Show that for any  $(n \times n)$  square matrix  $A$ , we have

$$\frac{d \operatorname{tr}(A)}{dA} = I_n.$$

## 8.4 Vectors and Matrices of Random Variables

Organizing large numbers of random variables using matrix algebra provides convenient formulas for manipulating their expectations, variances and covariances, and for expressing their joint pdf.

### 8.4.1 Expectations and Variance-Covariance Matrices

The expectation of a vector  $x$  of  $M$  random variables

$$x = \begin{bmatrix} X_1 & X_2 & \dots & X_M \end{bmatrix}^T$$

is defined as the vector of their expectations, i.e.,

$$E[x] = \begin{bmatrix} E[X_1] & E[X_2] & \dots & E[X_M] \end{bmatrix}^T.$$

Likewise, if  $X$  is a matrix of random variables

$$X = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1N} \\ X_{21} & X_{22} & \dots & X_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ X_{M1} & X_{M2} & \dots & X_{MN} \end{bmatrix}$$

then

$$E[X] = \begin{bmatrix} E[X_{11}] & E[X_{12}] & \dots & E[X_{1N}] \\ E[X_{21}] & E[X_{22}] & \dots & E[X_{2N}] \\ \vdots & \vdots & \ddots & \vdots \\ E[X_{M1}] & E[X_{M2}] & \dots & E[X_{MN}] \end{bmatrix}.$$

With these definitions, we can define the **variance-covariance matrix** of a vector  $x$  of random

variables. Let

$$\tilde{x} = x - E[x] = \begin{bmatrix} X_1 - E[X_1] \\ X_2 - E[X_2] \\ \vdots \\ X_M - E[X_M] \end{bmatrix} = \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \\ \vdots \\ \tilde{X}_M \end{bmatrix}$$

Then

$$\begin{aligned} E[\tilde{x}\tilde{x}^T] &= E[(x - E[x])(x - E[x])^T] = E \begin{bmatrix} \tilde{X}_1^2 & \tilde{X}_1\tilde{X}_2 & \dots & \tilde{X}_1\tilde{X}_M \\ \tilde{X}_2\tilde{X}_1 & \tilde{X}_2^2 & \dots & \tilde{X}_2\tilde{X}_M \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{X}_M\tilde{X}_1 & \tilde{X}_M\tilde{X}_2 & \dots & \tilde{X}_M\tilde{X}_M \end{bmatrix} \\ &= \begin{bmatrix} E[\tilde{X}_1^2] & E[\tilde{X}_1\tilde{X}_2] & \dots & E[\tilde{X}_1\tilde{X}_M] \\ E[\tilde{X}_2\tilde{X}_1] & E[\tilde{X}_2^2] & \dots & E[\tilde{X}_2\tilde{X}_M] \\ \vdots & \vdots & \ddots & \vdots \\ E[\tilde{X}_M\tilde{X}_1] & E[\tilde{X}_M\tilde{X}_2] & \dots & E[\tilde{X}_M\tilde{X}_M] \end{bmatrix} \\ &= \begin{bmatrix} \text{var}[X_1] & \text{cov}[X_1, X_2] & \dots & \text{cov}[X_1, X_M] \\ \text{cov}[X_1, X_2] & \text{var}[X_2] & \dots & \text{cov}[X_2, X_M] \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}[X_1, X_M] & \text{cov}[X_2, X_M] & \dots & \text{var}[X_M] \end{bmatrix}. \end{aligned} \quad (8.13)$$

In other words,  $E[(x - E[x])(x - E[x])^T]$  is a symmetric matrix containing the variances of all of the variables in  $x$ , and their covariances. We denote this matrix by  $\text{var}[X]$ .

If  $X$  is a random variable (singular), then  $E[aX + b] = aE[X] + b$  and  $\text{var}[aX + b] = a^2\text{var}[X]$ . The following are the matrix analogues: let  $x$  be an  $(M \times 1)$  vector of random variables, and  $X$  be a  $(K \times M)$  matrix of random variables. Let  $A = (a_{km})_{KM}$  be a  $(K \times M)$  matrix of constants, and  $b$  be a  $(K \times 1)$  vector of constants. Then

- $E[Ax + b] = AE[x] + b$

Proof:  $Ax + b$  is a  $(K \times 1)$  vector whose  $k$ th element is  $\sum_{m=1}^M (a_{km}x_m + b_k)$ , and the expectation of this term is

$$E \left[ \sum_{m=1}^M (a_{km}x_m + b_k) \right] = \sum_{m=1}^M a_{km}E[x_m] + b_k$$

which in turn is the  $k$ th element of the vector  $AE[x] + b$ .

- $\text{var}[Ax + b] = A\text{var}[x]A^T$

Proof: Since  $(Ax + b) - E[(Ax + b)] = A(x - E[x]) = A\tilde{x}$ , we have

$$\text{var}[Ax + b] = E[(A\tilde{x})(A\tilde{x})^T] = E[A\tilde{x}\tilde{x}^T A^T] = AE[\tilde{x}\tilde{x}^T]A^T = A\text{var}[x]A^T.$$

If  $X$  is a random variable (singular), we have  $\text{var}[X] = E[X^2] - E[X]^2$ . The matrix analogue of this result is

$$\text{var}[x] = E[xx^T] - E[x]E[x]^T \quad (8.14)$$

(see exercises).

Given a vector of random variables  $x$ , the linear combination  $c^T x$  has variance-covariance matrix  $c^T \text{var}[x] c$ . Since variances cannot be negative, we have  $c^T \text{var}[x] c \geq 0$  for all  $c$ . This means that  $\text{var}[x]$  is a **positive semidefinite** matrix. If there is a linear combination with zero variance, then at least one or more of the variables is actually a constant, or at least one or more of the variables is a linear combination of the others. Otherwise we have  $c^T \text{var}[x] c > 0$  for all  $c \neq 0$ , i.e.,  $\text{var}[x]$  is positive definite.

Since  $\text{var}[x]$  is symmetric and positive definite, we can find a  $C$  such that  $C^T \text{var}[x] C = \Lambda$  where  $\Lambda$  is diagonal. But  $C^T \text{var}[x] C$  is the variance of  $C^T x$ . In other words,  $C^T x$  is a vector of *uncorrelated* random variables, obtained from the (possibly) correlated random variables in  $x$ . Furthermore, we have  $\text{var}[Px] = I$  where  $P = \Lambda^{-1/2} C^T$ .

### 8.4.2 The Multivariate Normal Distribution

We presented the pdf of a bivariate normal distribution in an earlier chapter. We present here the pdf of a general multivariate normal distribution and some associated results. A  $(K \times 1)$  vector of random variables  $x$  is said to have a multivariate normal distribution with mean  $\mu$  and variance-covariance matrix  $\Sigma$ , denoted  $\text{Normal}_K(\mu, \Sigma)$ , if its pdf has the form

$$f(x) = (2\pi)^{-K/2} |\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right\}$$

We list a few results below, omitting proofs:

- If  $\Sigma$  is diagonal, then the variables are independent.
- If  $x \sim \text{Normal}_K(\mu, \Sigma)$ , then  $Ax + b \sim \text{Normal}_K(A\mu + b, A\Sigma A^T)$ .
- If we partition  $x$  as

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \text{Normal}_K \left( \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right)$$

where  $x_1$  is  $(K_1 \times 1)$  and  $x_2$  is  $(K_2 \times 1)$ , with  $K_1 + K_2 = K$ , then the marginal distribution of  $x_1$  is  $\text{Normal}_{K_1}(\mu_1, \Sigma_{11})$ , and the conditional distribution of  $x_2$  given  $x_1$  is

$$x_2 | x_1 \sim \text{Normal}_{K_2}(\mu_{2|1}, \Sigma_{22|1})$$

where

$$\mu_{2|1} = \mu_2 + \Sigma_{21} \Sigma_{11}^{-1} (x_1 - \mu_1) \quad \text{and} \quad \Sigma_{22|1} = \Sigma_{22} - \Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12}.$$

- If  $x \sim \text{Normal}_K(0, I)$  and  $A$  is symmetric and idempotent with rank  $J$ , then the scalar  $x^T A x$  is distributed  $\chi_{(J)}^2$ .
- If  $x \sim \text{Normal}_K(\mu, \Sigma)$ , then

$$(x - \mu)^T \Sigma^{-1} (x - \mu) \sim \chi_{(K)}^2.$$

### 8.4.3 Exercises

**Exercise 8.14.** Show that  $\text{var}[x] = E[xx^T] - E[x]E[x]^T$ .

**Exercise 8.15.** Show that  $E[\text{trace}[X]] = \text{trace}[E[X]]$ .

## 8.5 An Application of the Eigendecomposition of a Symmetric Matrix

Suppose  $X$  is a data matrix containing  $N$  observations of  $K$  variables

$$\begin{bmatrix} X_{1,1} & X_{2,1} & \cdots & X_{K,1} \\ X_{1,2} & X_{2,2} & \cdots & X_{K,2} \\ \vdots & \vdots & \ddots & \vdots \\ X_{1,N} & X_{2,N} & \cdots & X_{K,N} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_K \end{bmatrix} \quad \text{where } \mathbf{x}_k = \begin{bmatrix} X_{k,1} \\ X_{k,2} \\ \vdots \\ X_{k,N} \end{bmatrix}.$$

Suppose that these variables have had their sample means removed, i.e., that  $X_{k,i}$  is actually  $X_{k,i} - \bar{X}_k$  where  $\bar{X}_k = (1/N) \sum_{i=1}^N X_{k,i}$ . Furthermore, we assume that the variables have been standardized so that their individual sample variances are equal to 1.

Recall also that if you post-multiply  $X$  by a  $(K \times 1)$  vector  $c$ , you get a linear combination of the vectors of  $X$ :

$$Xc = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_K \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_K \end{bmatrix} = \sum_{k=1}^K c_k \mathbf{x}_k.$$

You can think of this as a sample of observations of a new random variable  $Z = c_1 X_1 + c_2 X_2 + \cdots c_K X_K$ .

If you have a collection of  $J$  number of such  $c$  vectors, say

$$C = \begin{bmatrix} c_1 & c_2 & \cdots & c_J \end{bmatrix}$$

where each of the  $c_1, c_2, \dots, c_J$  are now  $(K \times 1)$  vectors, then  $XC$  contains the observations of  $J$  new variables, each of which is a linear combination of the  $X$  variables:

$$XC = X \begin{bmatrix} c_1 & c_2 & \cdots & c_J \end{bmatrix} = \begin{bmatrix} Xc_1 & Xc_2 & \cdots & Xc_J \end{bmatrix}$$

Now consider the matrix

$$X^T X = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_K^T \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_K \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^T \mathbf{x}_1 & \mathbf{x}_1^T \mathbf{x}_2 & \cdots & \mathbf{x}_1^T \mathbf{x}_K \\ \mathbf{x}_2^T \mathbf{x}_1 & \mathbf{x}_2^T \mathbf{x}_2 & \cdots & \mathbf{x}_2^T \mathbf{x}_K \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_K^T \mathbf{x}_1 & \mathbf{x}_K^T \mathbf{x}_2 & \cdots & \mathbf{x}_K^T \mathbf{x}_K \end{bmatrix}.$$

Since the observations have been de-means, the terms  $\mathbf{x}_k^T \mathbf{x}_j = \sum_{i=1}^N X_{k,i} X_{j,i}$  are just  $N$  times the sample covariance of  $X_k$  and  $X_j$ , and  $\mathbf{x}_k^T \mathbf{x}_k = \sum_{i=1}^N X_{k,i}^2$  is just  $N$  times the sample variance of  $X_k$ . In other words,  $(1/N)X^T X$  is the *sample* variance-covariance matrix of the variables. To simplify notation, I will just drop the  $(1/N)$  and refer to  $X^T X$  as the sample variance-covariance matrix which summarizes the correlations between the variables in your data matrix  $X$ .

Since  $X^T X$  is symmetric, we can write

$$X^T X = C \Lambda C^T \quad \text{where } CC^T = I \quad (8.15)$$

and where  $\Lambda$  is a diagonal matrix containing the eigenvalues of  $X^T X$ , and the columns of  $C$  are the corresponding eigenvectors. Since

$$C\Lambda C^T = \begin{bmatrix} c_1 & c_2 & \dots & c_n \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \begin{bmatrix} c_1^T \\ c_2^T \\ \dots \\ c_n^T \end{bmatrix} = \sum_{i=1}^n \lambda_i c_i c_i^T$$

and since the terms of a summation can be added in any order, we can rearrange the columns of  $C$  in any order, as long as we also re-arrange the diagonals of  $\Lambda$  accordingly. We usually arrange it such that the eigenvalues are in descending order  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_K$ .

The decomposition (8.15) can be re-written as

$$C^T X^T X C = \Lambda$$

or

$$(XC)^T (XC) = \Lambda$$

This is just the sample covariance matrix of the variables whose observations appear in the columns of  $XC$ . Each of the columns in  $XC$  are just linear combinations of the columns of  $X$ . Since  $\Lambda$  is diagonal, this says that the  $K$  columns of  $XC$  are orthogonal. In other words, we have created  $K$  *uncorrelated* new variables, each of which is a linear combination of the (possibly correlated)  $X$  variables. Furthermore, writing the new variables as

$$XC = \begin{bmatrix} Xc_1 & Xc_2 & \dots & Xc_K \end{bmatrix}$$

we see that  $\lambda_1$  is the sample variance of  $Xc_1$ ,  $\lambda_2$  is the sample variance of  $Xc_2$ , and so on.

The new variables in  $XC$  are called the **principal components** of  $X$ . These are often used as a dimension reduction technique. Suppose  $X$  contains  $N$  observations of many many variables, so  $K$  is large. Quite often we find that only a few of the  $\lambda_i$ 's associated with  $X$  are large, and the rest very small. In other words, only a few of the principal components of  $X$  have substantial variation, the rest have very little variation. Sometimes just two or three of the principal components associated with the largest eigenvalues account for the bulk of variation in the data. If that is the case, then we have effectively reduced the number of variables from  $K$  to just two or three. The difficulty is that these principal components are often hard to interpret.

## Chapter 9

### Least Squares with Matrix Algebra

The mathematics of least squares is best expressed in matrix form. Proofs of results are much more concise and more general, and we can draw on the insights of linear algebra to understand least squares at a deeper level. Furthermore, the same mathematics applies to a vast number of advanced linear models including multiple equation models, and is useful even for non-linear ones, so it is well worth the time and effort to master the mathematics of least squares estimation expressed using matrix algebra. The objective of this chapter is to help you become familiarized with the mathematics of least squares estimation of linear models, and to provide proofs of results previously omitted or only proven partially.

*We use the following packages in this chapter.*

```
library(readxl)
library(car)
library(sandwich)
```

#### 9.1 The Setup

Suppose that you have a sample  $\{Y_i, X_{1,i}, X_{2,i}, \dots, X_{K-1,i}\}_{i=1}^N$  such that

$$Y_i = \beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} + \dots + \beta_{K-1} X_{K-1,i} + \epsilon_i, \quad i = 1, 2, \dots, N. \quad (9.1)$$

We use  $X_{k,i}$  to denote the  $i$ th observation of variables  $X_k$ . We can write the regression in matrix form as

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{bmatrix} = \begin{bmatrix} 1 & X_{1,1} & X_{2,1} & \dots & X_{K-1,1} \\ 1 & X_{1,2} & X_{2,2} & \dots & X_{K-1,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{1,N} & X_{2,N} & \dots & X_{K-1,N} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{K-1} \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix} \quad (9.2)$$

or simply

$$y = X\beta + \varepsilon \quad (9.3)$$

where  $y$  is the  $(N \times 1)$  vector  $[Y_1 \ Y_2 \ \dots \ Y_N]^T$ ,  $X$  is the  $(N \times K)$  matrix of regressors,  $\beta$  is the  $(K \times 1)$  coefficient vector, and  $\varepsilon$  is the  $(N \times 1)$  vector of noise terms. We will use  $\{\epsilon_i\}_{i=1}^N$  to denote a sample of the noise variable  $\epsilon$ . We organize the  $\{\epsilon_i\}_{i=1}^N$  into the vector  $\varepsilon$ , as in (9.2) and (9.3). We assume throughout that  $N > K$ .

We can partition the regressor matrix by observation:

$$X = \begin{bmatrix} 1 & X_{1,1} & X_{2,1} & \dots & X_{K-1,1} \\ 1 & X_{1,2} & X_{2,2} & \dots & X_{K-1,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{1,N} & X_{2,N} & \dots & X_{K-1,N} \end{bmatrix} = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix} \quad (9.4)$$

and write the model as

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{bmatrix} = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix} \beta + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}$$

or

$$Y_i = x_i^T \beta + \epsilon_i, \quad i = 1, 2, \dots, N.$$

It is sometimes helpful to partition the regressor matrix by variable:

$$X = \left[ \begin{array}{c|c|c|c|c} 1 & X_{1,1} & X_{2,1} & \dots & X_{K-1,1} \\ 1 & X_{1,2} & X_{2,2} & \dots & X_{K-1,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{1,N} & X_{2,N} & \dots & X_{K-1,N} \end{array} \right] = [i_N \quad x_1 \quad x_2 \quad \dots \quad x_{K-1}] \quad (9.5)$$

where  $i_N$  is the  $(N \times 1)$  vector of ones, and  $x_k$  is the vector of observations of variable  $X_k$ . Feasibility of OLS estimation will require  $X$  to have full column rank, i.e., that there is no non-zero  $(K \times 1)$  vector  $c$  such that

$$Xc = c_0 + c_1 x_1 + c_2 x_2 + \dots + c_{K-1} x_{K-1} = 0.$$

In other words, we must assume that there is variation in each of the variables (apart from the constant vector), and that no one variable can be written as a linear combination of the other variables. The full column rank assumption implies that  $X^T X$  is non-singular (i.e., has an inverse).

We continue to assume that the noise terms  $\epsilon_i$  have zero mean conditional on all observations of all regressors:

$$E[\epsilon_i | x_1, x_2, \dots, x_{K-1}] = 0.$$

In matrix form, we can write this even more simply as

$$E[\epsilon | X] = 0_{N \times 1}.$$

In the basic model, the noise terms were assumed to be conditionally homoskedasticity and uncorrelated:

$$\begin{aligned} \text{var}[\epsilon_i | x_1, x_2, \dots, x_{K-1}] &= \sigma^2 \quad \text{for all } i = 1, 2, \dots, N, \\ \text{cov}[\epsilon_i \epsilon_j | x_1, x_2, \dots, x_{K-1}] &= 0 \quad \text{for all } i \neq j, i, j = 1, 2, \dots, N. \end{aligned}$$

The (conditional) variance-covariance matrix of  $\epsilon$  contains the conditional variances and covariances of the noise terms:

$$\text{var}[\epsilon] = E[\epsilon \epsilon^T | X] = \begin{bmatrix} \text{var}[\epsilon_1 | X] & \text{cov}[\epsilon_1, \epsilon_2 | X] & \dots & \text{cov}[\epsilon_1, \epsilon_N | X] \\ \text{cov}[\epsilon_2, \epsilon_1 | X] & \text{var}[\epsilon_2 | X] & \dots & \text{cov}[\epsilon_2, \epsilon_N | X] \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}[\epsilon_N, \epsilon_1 | X] & \text{cov}[\epsilon_N, \epsilon_2 | X] & \dots & \text{var}[\epsilon_N | X] \end{bmatrix}.$$



In the case of homoskedastic uncorrelated noise terms, the variance-covariance matrix of  $\varepsilon$  is simply

$$E[\varepsilon\varepsilon^T|X] = \begin{bmatrix} \sigma^2 & 0 & \dots & 0 \\ 0 & \sigma^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma^2 \end{bmatrix} = \sigma^2 I$$

where  $I$  is the  $(N \times N)$  identity matrix (we will generally not indicate the dimensions of identity matrices, and leave the reader to deduce dimensions from context). If the errors are conditionally heteroskedastic with  $\text{var}[\epsilon_i|X] = \sigma_i^2$  but uncorrelated, then the conditional variance-covariance matrix of  $\varepsilon$  becomes

$$E[\varepsilon\varepsilon^T|X] = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_N^2 \end{bmatrix} = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_N^2).$$

If there is correlation between the errors, then the var-cov matrix of  $\varepsilon$  will no longer be diagonal. We write the general variance matrix of  $\varepsilon$  as

$$E[\varepsilon\varepsilon^T|X] = \Omega,$$

although we will have to impose some structure on  $\Omega$  for the analysis to be feasible. We list the assumptions of the basic model below:

**Assumption Set D** The sample  $\{Y_i, X_{1,i}, X_{2,i}, \dots, X_{K-1,i}\}_{i=1}^N$  satisfies

$$(D1) \ y = X\beta + \varepsilon,$$

$$(D2) \ E[\varepsilon|X] = 0,$$

$$(D3) \ E[\varepsilon\varepsilon^T|X] = \sigma^2 I,$$

$$(D4) \ Xc \neq 0 \text{ for all } c \neq 0,$$

where  $y$ ,  $X$  and  $\varepsilon$  are as defined in this section.

## 9.2 Ordinary Least Squares

Let  $\hat{\beta}$  denote some estimator for  $\beta$ . Then the fitted values associated with these estimators are

$$\hat{y} = X\hat{\beta}$$

and the residuals are

$$\hat{\varepsilon} = y - \hat{y} = y - X\hat{\beta}.$$

The sum of squared residuals is then

$$\begin{aligned} SSR &= \hat{\varepsilon}^T \hat{\varepsilon} = (y - X\hat{\beta})^T (y - X\hat{\beta}) \\ &= y^T y - \hat{\beta}^T X^T y - y^T X \hat{\beta} + \hat{\beta}^T X^T X \hat{\beta} \\ &= y^T y - 2\hat{\beta}^T X^T y + \hat{\beta}^T X^T X \hat{\beta} \end{aligned}$$

where we have used the fact that  $\hat{\beta}^T X^T y$  is the transpose of  $y^T X \hat{\beta}$ , and the transpose of a scalar is the scalar itself. OLS estimators are those that minimize SSR:

$$\hat{\beta}^{ols} = \operatorname{argmin}_{\hat{\beta}} SSR.$$

The first-order conditions are

$$\left. \frac{\partial SSR}{\partial \hat{\beta}} \right|_{\hat{\beta}^{ols}} = -2X^T y + 2X^T X \hat{\beta}^{ols} = 0.$$

This implies

$$X^T X \hat{\beta}^{ols} = X^T y$$

which, given our assumption that  $X$  is full column rank, can be solved for  $\hat{\beta}^{ols}$ :

$$\hat{\beta}^{ols} = (X^T X)^{-1} X^T y.$$

The second partial derivatives of the SSR

$$\frac{\partial^2 SSR}{\partial \hat{\beta} \partial \hat{\beta}^T} = 2X^T X$$

is positive definite, since the assumption of full column rank of  $X$  means that  $Xc \neq 0$  for all  $c \neq 0$ , from which it follows that

$$c^T X^T X c = (Xc)^T Xc > 0.$$

The FOC can also be written as

$$X^T (y - X \hat{\beta}^{ols}) = X^T \hat{\varepsilon}^{ols} = 0. \quad (9.6)$$

Partitioning  $X^T$  “by variable” as in (9.5), we can see that (9.6) says that OLS residuals sum to zero, and are orthogonal to each of the regressors:

$$X^T \hat{\varepsilon}^{ols} = \begin{bmatrix} i_N^T \\ x_1^T \\ x_2^T \\ \vdots \\ x_{K-1}^T \end{bmatrix} \hat{\varepsilon}^{ols} = \begin{bmatrix} i_N^T \hat{\varepsilon}^{ols} \\ x_1^T \hat{\varepsilon}^{ols} \\ x_2^T \hat{\varepsilon}^{ols} \\ \vdots \\ x_{K-1}^T \hat{\varepsilon}^{ols} \end{bmatrix} = 0.$$

Partitioning  $X$  by observation, as in (9.4), we can also write the OLS estimator as

$$\begin{aligned}\hat{\beta}^{ols} &= (X^T X)^{-1} X^T y \\ &= \left\{ \begin{bmatrix} x_1 & x_2 & \dots & x_N \end{bmatrix} \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix} \right\}^{-1} \begin{bmatrix} x_1 & x_2 & \dots & x_N \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{bmatrix} \\ &= \left( \sum_{i=1}^N x_i x_i^T \right)^{-1} \sum_{i=1}^N x_i Y_i.\end{aligned}$$

This form emphasizes the role that sample averages play in the estimation of  $\beta$ :

$$\hat{\beta}^{ols} = \left( \frac{1}{N} \sum_{i=1}^N x_i x_i^T \right)^{-1} \left( \frac{1}{N} \sum_{i=1}^N x_i Y_i \right).$$

### 9.3 Algebraic Properties of OLS Estimators

We list here some algebraic properties of OLS estimators. These hold as long as Assumption D4 holds. From this point onwards, we drop the ‘OLS’ superscript from the OLS estimators, fitted values and residuals, and write  $\hat{\beta}$ ,  $\hat{Y}$  and  $\hat{\varepsilon}$  for  $\hat{\beta}^{ols}$ ,  $\hat{Y}^{ols}$  and  $\hat{\varepsilon}^{ols}$ . We will reinstate the superscript whenever context demands it so.

1. OLS estimators are linear estimators:  $\hat{\beta} = (X^T X)^{-1} X^T y = Ay$  means that each OLS estimator  $\hat{\beta}_k$ ,  $k = 0, 1, \dots, K-1$  can be written as

$$\hat{\beta}_k = \sum_{i=1}^N a_{k,i} Y_i$$

where  $a_{k,i}$ ,  $i = 1, 2, \dots, N$  are the elements of the  $k$ th row of  $A = (X^T X)^{-1} X^T$ .

2. The OLS estimators can also be written as

$$\begin{aligned}\hat{\beta} &= (X^T X)^{-1} X^T y \\ &= (X^T X)^{-1} X^T (X\beta + \varepsilon) \\ &= \beta + (X^T X)^{-1} X^T \varepsilon.\end{aligned}$$

3. The OLS fitted values can be written as

$$\hat{y} = X\hat{\beta} = X(X^T X)^{-1} X^T y.$$

The matrix  $X(X^T X)^{-1} X^T$  is sometimes called the ‘hat’ matrix (because it puts a ‘hat’ on  $y$ ). It is also called the “Projection” matrix (since it projects  $y$  onto the space spanned by

the columns of  $X$ ) and denoted  $P$ . It has the convenient property that it is symmetric:

$$\begin{aligned} P^T &= (X(X^T X)^{-1} X^T)^T \\ &= (X^{TT}[(X^T X)^{-1}]^T X^T) \\ &= X(X^T X)^{-1} X^T = P \end{aligned}$$

where we have used the fact that  $(X^T X)^{-1}$  is symmetric (why is  $(X^T X)^{-1}$  symmetric?). The matrix  $P$  is also idempotent, meaning that  $PP = P$ :

$$PP = X(X^T X)^{-1} X^T X(X^T X)^{-1} X^T = X(X^T X)^{-1} X^T = P.$$

Symmetric and idempotent matrices have the convenient property that their rank is equal to their trace, which is easy to compute. Since

$$\text{tr}(P) = \text{tr}(X(X^T X)^{-1} X^T) = \text{tr}(X^T X(X^T X)^{-1}) = \text{tr}(I_K) = K,$$

the rank of  $P$  is  $K$ .

4. The OLS residuals can be written as

$$\hat{\varepsilon} = y - \hat{y} = (I - X(X^T X)^{-1} X^T)y.$$

The matrix  $I - X(X^T X)^{-1} X^T$  is also symmetric and idempotent, and its trace, and therefore its rank, is  $N - K$  (see exercises). It is often denoted by  $M$ , and has the property that it “eliminates  $X$ ” in the sense that

$$MX = (I - X(X^T X)^{-1} X^T)X = X - X = 0.$$

As a consequence of this, we have

$$MP = MX(X^T X)^{-1} X^T = 0.$$

Of course, you can also see this from  $MP = (I - P)P = P - PP = P - P = 0$ .

5. We have already noted from the FOC that the OLS residuals sum to zero, and are orthogonal to each of the regressors. Since  $y = \hat{y} + \hat{\varepsilon}$ , it follows that  $\bar{Y} = \widehat{\bar{Y}}$ . Furthermore,  $\hat{y}^T \hat{\varepsilon} = 0$ . That is, the fitted values and the residuals are orthogonal. We can also use the fact that  $MP = PM = 0$ :

$$\hat{y}^T \hat{\varepsilon} = y^T P M y = 0.$$

For those who are not uncomfortable thinking about  $N$ -dimensional vectors in geometric terms, this means the fitted values and residuals are at “right-angles” in  $N$ -dimensional space. The length of a vector  $y$  is  $\sqrt{y^T y}$ . Using orthogonality of the fitted values and residuals, we get

$$\begin{aligned} y^T y &= \hat{y}^T \hat{y} + 2\hat{y}^T \hat{\varepsilon} + \hat{\varepsilon}^T \hat{\varepsilon} \\ &= \hat{y}^T \hat{y} + \hat{\varepsilon}^T \hat{\varepsilon}. \end{aligned} \tag{9.7}$$

This is just Pythagoras’s Theorem (in  $N$ -dimensional space).

6. One useful application of the fact that  $M$  eliminates  $X$  is to derive a formula linking the residuals to the noise terms. We have

$$\hat{\varepsilon} = My = M(X\beta + \varepsilon) = M\varepsilon$$

This result, and the fact that  $M$  is symmetric and idempotent, means that the sum of squared residuals can be written as

$$\hat{\varepsilon}^T \hat{\varepsilon} = (M\varepsilon)^T M\varepsilon = \varepsilon^T M^T M\varepsilon = \varepsilon^T M\varepsilon.$$

#### 9.4 Statistical Properties of OLS Estimators.

Under Assumption Set D,  $\hat{\beta}$  is unbiased. Using  $\hat{\beta} = \beta + (X^T X)^{-1} X^T \varepsilon$ , we have

$$E[\hat{\beta}|X] = \beta + (X^T X)^{-1} X^T E[\varepsilon|X] = \beta$$

which implies  $E[\hat{\beta}] = \beta$ .

The proof of unbiasedness uses Assumption D2 directly, and Assumptions D1 and D4 indirectly, but it does not make any use of Assumption D3. Unbiasedness of OLS estimators does not depend on the the structure of the variance-covariance matrix of the noise terms.

The variances and covariances of all of the OLS coefficient estimators can be obtained by computing the (conditional) variance-covariance matrix of  $\hat{\beta}$ :

$$\begin{aligned} \text{var}[\hat{\beta}|X] &= E[(\hat{\beta} - E[\hat{\beta}])(\hat{\beta} - E[\hat{\beta}])^T|X] \\ &= E[(\hat{\beta} - \beta)(\hat{\beta} - \beta)^T|X] \\ &= E[(X^T X)^{-1} X^T \varepsilon \varepsilon^T X (X^T X)^{-1}|X] \\ &= (X^T X)^{-1} X^T E[\varepsilon \varepsilon^T|X] X (X^T X)^{-1}. \end{aligned}$$

In the general case  $E[\varepsilon \varepsilon^T|X] = \Omega$ , this becomes

$$\text{var}[\hat{\beta}|X] = (X^T X)^{-1} X^T \Omega X (X^T X)^{-1}.$$

To get any further we would have to put more structure on  $\Omega$ . If we have uncorrelated but possibly heteroskedastic noise terms, then

$$\begin{aligned} \text{var}[\hat{\beta}|X] &= (X^T X)^{-1} X^T \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_N^2) X (X^T X)^{-1} \\ &= \left( \sum_{i=1}^N x_i x_i^T \right)^{-1} \begin{bmatrix} x_1 & x_2 & \dots & x_N \end{bmatrix} \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_N^2 \end{bmatrix} \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix} \left( \sum_{i=1}^N x_i x_i^T \right)^{-1}. \quad (9.8) \\ &= \left( \sum_{i=1}^N x_i x_i^T \right)^{-1} \left( \sum_{i=1}^N \sigma_i^2 x_i x_i^T \right) \left( \sum_{i=1}^N x_i x_i^T \right)^{-1} \end{aligned}$$

If we further assume conditional homoskedasticity as in Assumption D3, then  $\text{var}[\hat{\beta}|X]$  further simplifies to

$$\begin{aligned}\text{var}[\hat{\beta}|X] &= \left( \sum_{i=1}^N x_i x_i^T \right)^{-1} \left( \sum_{i=1}^N \sigma^2 x_i x_i^T \right) \left( \sum_{i=1}^N x_i x_i^T \right)^{-1} \\ &= \sigma^2 \left( \sum_{i=1}^N x_i x_i^T \right)^{-1} \\ &= \sigma^2 (X^T X)^{-1}.\end{aligned}\tag{9.9}$$

Under conditional homoskedasticity, an unbiased estimator for  $\sigma^2$  is

$$\widehat{\sigma^2} = \frac{\hat{\varepsilon}^T \hat{\varepsilon}}{N - K} = \frac{\sum_{i=1}^N \hat{\varepsilon}_i^2}{N - K}.$$

To prove unbiasedness of this estimator, we note that

$$\begin{aligned}E[\hat{\varepsilon}^T \hat{\varepsilon}|X] &= E[\varepsilon^T M \varepsilon|X] = E[\text{tr}(\varepsilon^T M \varepsilon)|X] \\ &= E[\text{tr}(\varepsilon \varepsilon^T M)|X] = \text{tr}(E[\varepsilon \varepsilon^T M|X]) \\ &= \text{tr}(E[\varepsilon \varepsilon^T|X] M) = \text{tr}(\sigma^2 M) \\ &= \sigma^2 (N - K).\end{aligned}$$

This implies that  $E[\hat{\varepsilon}^T \hat{\varepsilon}] = \sigma^2 (N - K)$  and unbiasedness of  $\widehat{\sigma^2}$  follows. We therefore estimate  $\text{var}[\hat{\beta}]$  using

$$\widehat{\text{var}}[\hat{\beta}|X] = \widehat{\sigma^2} (X^T X)^{-1}.$$

**Example 9.1.** In the simple linear regression  $Y_i = \beta_0 + \beta_1 X_{1,i} + \varepsilon_i$ ,  $i = 1, 2, \dots, N$ , we have

$$X = \begin{bmatrix} 1 & X_{1,1} \\ 1 & X_{1,2} \\ \vdots & \vdots \\ 1 & X_{1,N} \end{bmatrix} \quad \text{and} \quad X^T X = \begin{bmatrix} N & \sum_{i=1}^N X_{1,i} \\ \sum_{i=1}^N X_{1,i} & \sum_{i=1}^N X_{1,i}^2 \end{bmatrix}.$$

The OLS estimators  $\hat{\beta}_0$  and  $\hat{\beta}_1$  can be found from

$$\hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix} = (X^T X)^{-1} X^T y.$$

If Assumption Set D holds, their variances and covariances can be found from

$$\text{var}[\hat{\beta}|X] = \begin{bmatrix} \text{var}[\hat{\beta}_0|X] & \text{cov}[\hat{\beta}_0, \hat{\beta}_1|X] \\ \text{cov}[\hat{\beta}_0, \hat{\beta}_1|X] & \text{var}[\hat{\beta}_1|X] \end{bmatrix} = \sigma^2 (X^T X)^{-1}.$$

We will discuss estimation of the variance-covariance matrix under conditional heteroskedasticity when we discuss asymptotic properties of OLS estimators. In the meantime, we continue our discussion under the assumption of conditional homoskedasticity.

Under Assumption Set D (with conditional homoskedasticity), the OLS estimators are the

most efficient estimators among all linear unbiased estimators, i.e.,

$$\text{var}[c^T \hat{\beta}|X] \leq \text{var}[c^T \tilde{\beta}|X] \quad (9.10)$$

for all unbiased estimators of the form  $\tilde{\beta} = By$ . In other words, each individual  $\hat{\beta}_k$  has the smallest variance among all linear unbiased estimators of  $\beta_k$ , and all linear combinations of  $\hat{\beta}$  will have a smaller variance than the same linear combination of any other linear unbiased estimator of  $\beta$ .

**Example 9.2.** Suppose  $Y_i = x_i^T \beta + \epsilon_i$ , with Assumption Set D holding. The prediction of  $Y_i$  at  $x_i = x_0$  is  $\hat{Y}(x_0) = x_0^T \hat{\beta}$ . This is a linear combination of the estimators in  $\hat{\beta}$ . Since the OLS estimators  $\hat{\beta}$  are most efficient among all linear unbiased estimators, this prediction rule gives us the most precise linear unbiased prediction of  $Y$  at  $x = x_0$ .

To prove (9.10), let  $\tilde{\beta} = By$  where  $B$  comprises constants and elements of  $X$ , and such that  $\tilde{\beta}$  is unbiased. Write  $B = D + (X^T X)^{-1} X^T$ , so

$$\tilde{\beta} = DX\beta + D\epsilon + \beta + (X^T X)^{-1} X^T \epsilon.$$

We have already assumed  $E[\epsilon|X] = 0$ . To ensure unbiasedness of  $\tilde{\beta}$ , we have also to assume that  $DX = 0$ . We make this assumption. Then

$$\begin{aligned} \text{var}[\tilde{\beta}|X] &= E[(\tilde{\beta} - \beta)(\tilde{\beta} - \beta)^T|X] \\ &= E[(D + (X^T X)^{-1} X^T) \epsilon \epsilon^T (D + (X^T X)^{-1} X^T)^T|X] \\ &= \sigma^2[(X^T X)^{-1} + DD^T] \\ &= \text{var}[\hat{\beta}] + \sigma^2 DD^T. \end{aligned}$$

Result (9.10) follows immediately.

## 9.5 Hypothesis Testing

Suppose in addition to (D1)-(D4) we also assume that the noise terms are normally distributed. We can write this as

$$(D5) \quad \epsilon|X \sim \text{Normal}_N(0, \sigma^2 I).$$

As written, assumption D5 actually subsumes D2 and D3, but we will keep D2, D3, and D5 as separate statements. Obviously if you have conditional heteroskedasticity or correlation in the noise terms, then the variance expression in D5 must be modified accordingly.

With assumption D5, we have

$$\hat{\beta}|X \sim \text{Normal}_K(\beta, \sigma^2 (X^T X)^{-1}) \quad (9.11)$$

since  $\hat{\beta}$  is a constant plus a linear combination of normally distributed noise terms. This can be used to develop t and F-tests of linear hypotheses concerning elements of  $\beta$ . A general single

linear hypothesis can be written as

$$H_0 : r^T \beta = r_0 \quad \text{vs} \quad r^T \beta \neq r_0.$$

**Example 9.3.** To test  $\beta_1 + \beta_2 = 1$  in the regression

$$Y_i = \beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} + \epsilon_i,$$

set  $r^T = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$  and  $r_0 = 1$ .

From (9.11), we have

$$r^T \hat{\beta} | X \sim \text{Normal}(r^T \beta, r^T [\sigma^2 (X^T X)^{-1}] r).$$

If the null hypothesis  $r^T \beta = r_0$  holds, then

$$r^T \hat{\beta} | X \sim \text{Normal}(r_0, r^T [\sigma^2 (X^T X)^{-1}] r)$$

and

$$\frac{r^T \hat{\beta} - r_0}{\sqrt{r^T [\sigma^2 (X^T X)^{-1}] r}} \sim \text{Normal}(0, 1).$$

Furthermore, it can be shown that if we replace  $\sigma^2$  with  $\widehat{\sigma^2}$ , then

$$\begin{aligned} t &= \frac{r^T \hat{\beta} - r_0}{\sqrt{r^T [\widehat{\sigma^2} (X^T X)^{-1}] r}} \\ &= \frac{r^T \hat{\beta} - r_0}{\sqrt{r^T \widehat{\text{var}}[\hat{\beta} | X] r}} \sim t_{(N-K)}. \end{aligned} \tag{9.12}$$

This can be used to test the hypothesis  $H_0 : r^T \beta = r_0$  in the usual way.

To test multiple hypotheses jointly, write the hypotheses as

$$H_0 : R\beta = r_0 \quad \text{vs} \quad H_A : R\beta \neq r_0$$

where now  $R$  is a  $(J \times K)$  matrix, and  $r_0$  is a  $(J \times 1)$  vector.

**Example 9.4.** To test the hypotheses

$$H_0 : \beta_1 + \beta_2 = 1 \text{ and } \beta_3 = 0 \quad \text{vs} \quad H_A : \beta_1 + \beta_2 \neq 1 \text{ or } \beta_3 \neq 0 \text{ (or both),}$$

set the matrices  $R$  and  $r_0$  to

$$R = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad r_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

To test multiple hypotheses jointly, we can again compare the sum of squared residuals from the restricted and unrestricted regressions, as explained in a previous chapter.



**Example 9.5.** We continue with Example 9.4. The restricted regression is

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 X_{1,i} + (1 - \beta_1) X_{2,i} + 0 X_{3,i} + \epsilon_i \\ &= \beta_0 + \beta_1 (X_{1,i} - X_{2,i}) + X_{2,i} + \epsilon_i. \end{aligned}$$

To estimate this, regress  $Y_i - X_{2,i}$  on a constant as  $X_{1,i} - X_{2,i}$ . Then calculate the restricted OLS residuals

$$\hat{\epsilon}_{i,r} = Y_i - \hat{\beta}_{0,r} - \hat{\beta}_{1,r}(X_{1,i} - X_{2,i}) - X_{2,i}$$

and finally calculate  $SSR_r = \sum_{i=1}^N \hat{\epsilon}_{i,r}^2 = \hat{\epsilon}_r^T \hat{\epsilon}_r$ , where  $\hat{\epsilon}_r$  is the vector of restricted OLS residuals.

It can be shown that if the null is true, then

$$F = \frac{(\hat{\epsilon}_r^T \hat{\epsilon}_r - \hat{\epsilon}^T \hat{\epsilon})/J}{\hat{\epsilon}^T \hat{\epsilon}/(N-K)} \sim F_{(J, N-K)} \quad (9.13)$$

where  $J$  is the number of restrictions being tested and  $\hat{\epsilon}$  is the vector of unrestricted OLS residuals. You would reject  $H_0$  if the  $F$ -statistic is “improbably large”, meaning that  $F > F_{\alpha, J, N-K}$  where  $F_{\alpha, J, N-K}$  is the  $1 - \alpha$  percentile of the  $F_{J, N-K}$  distribution. Typically  $\alpha = 0.01, 0.05$  or  $0.1$ .

In practice you do not have to compute the restricted regression. It can be shown that

$$\hat{\epsilon}_r^T \hat{\epsilon}_r - \hat{\epsilon}^T \hat{\epsilon} = (R\hat{\beta} - r_0)^T [R(X^T X)^{-1} R^T]^{-1} (R\hat{\beta} - r_0) \quad (9.14)$$

where  $\hat{\beta}$  is the unrestricted OLS estimators (we show this in an Appendix). Furthermore, since the denominator of the  $F$ -statistic is  $\hat{\sigma}^2$ , we can write the  $F$ -statistic as

$$\begin{aligned} F &= (R\hat{\beta} - r_0)^T [R(\hat{\sigma}^2(X^T X)^{-1}) R^T]^{-1} (R\hat{\beta} - r_0)/J \\ &= (R\hat{\beta} - r_0)^T [R \widehat{\text{var}}[\hat{\beta}|X] R^T]^{-1} (R\hat{\beta} - r_0)/J \\ &\sim F_{(J, N-K)} \end{aligned} \quad (9.15)$$

**Example 9.6.** We use data in `earnings.xlsx` to to estimate the equation

$$\ln(\text{earnings}_i) = \beta_0 + \beta_1 \text{height}_i + \beta_2 \text{male}_i + \beta_3 \text{wexp}_i + \beta_4 \text{tenure}_i + \epsilon_i$$

We compute the OLS estimates and associated statistics from the formulas derived in this chapter. The first four rows of the  $X$  matrix are

```
df_earnings <- read_excel("data\\earnings.xlsx")
y <- as.matrix(log(df_earnings$earnings))
N <- length(y)
X <- as.matrix(cbind("const"=rep(1,length(y)),
                    df_earnings[,c('height','male','wexp','tenure')]))
K <- dim(X)[2]
head(X,4)
```

	const	height	male	wexp	tenure
[1,]	1	67	1	22.384610	2.750000
[2,]	1	67	1	8.903846	2.384615

```
[3,]    1    69    1 13.250000 5.750000
[4,]    1    72    1 18.250000 6.134615
```

The following code implement the formulas derived earlier.

```
XTX <- t(X)%*%X # t() for transpose
XTXinv <- solve(XTX) # solve() to get inverse
XTy <- t(X)%*%y
bhat <- solve(XTX,XTy) # beta_hat, alt: XTX_1 %*% XTy, given mtd preferred
yhat <- X%*%bhat # fitted values
ehat <- y - yhat # residuals
s2hat <- sum(ehat^2)/(N-K) # or t(ehat)%*%ehat
varbhat <- s2hat*XTXinv # var(bhat)
sebhat <- sqrt(diag(varbhat)) # se(bhat)
tbhat <- bhat/sebhat
pval <- 2*(1-pt(abs(tbhat),N-K))
star <- rep("", K)
star[pval<0.1] <- "."
star[pval<0.05] <- "*"
star[pval<0.01] <- "***"
star[pval<0.001] <- "****"
Results <- data.frame("Estimate"=bhat, "Std. Err."=sebhat, "t-stat"=tbhat, "p-val"=pval, star)
names(Results)[K] <- ""
print(Results, digits=6, right=FALSE)
cat("---\nSignif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1\n")

      Estimate Std..Err. t.stat p.val
const 0.98423994 0.53948678 1.824400 0.068649145 .
height 0.02271202 0.00824686 2.754021 0.006087062 **
male   0.17052245 0.07035690 2.423678 0.015694822 *
wexp   0.00541464 0.00585844 0.924245 0.355775183
tenure 0.01335911 0.00397157 3.363681 0.000824198 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As an illustration, we test the hypothesis that  $\beta_3 = \beta_4$ , i.e.,  $\beta_3 - \beta_4 = 0$ .

```
r <- matrix(c(0,0,0,1,-1), ncol=1)
t <- t(r)%*%bhat / sqrt(t(r)%*%varbhat%*%r)
tpval <- 2*(1-pt(abs(t),N-K))
cat("H0 wage = tenure: t-stat ",round(t,4),",", pval", round(tpval,6))

H0 wage = tenure: t-stat -0.9779 , pval 0.328545
```

We do not reject the hypothesis. To jointly test the hypotheses  $\beta_3 - \beta_4 = 0$  and  $\beta_1 = 0$ , we use the F-test, with

$$R = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \quad \text{and} \quad r_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

```
R <- matrix(c(0,1,0,0,0,0,0,0,1,-1), nrow=2, byrow=TRUE)
r0 <- matrix(c(0,0),2,1)
J = length(r0)
```

```
Rb <- R %*% bhat - r0
F <- t(Rb) %*% solve(R%*%varbhat%*%t(R)) %*% Rb / J
Fpval <- (1-pf(F,J,N-K))
cat("H0 height = 0 and wage = tenure: F-stat ",round(F,4),", pval", round(Fpval,6))
```

H0 height = 0 and wage = tenure: F-stat 4.3802 , pval 0.012975

The following are the results using the `lm()` function for the regression, and the `linearHypothesis()` function from the `car` package for the general t- and F-tests.

```
mdl <- lm(log(earnings)~height+male+wexp+tenure, data=df_earnings)
summary(mdl)
```

Call:

```
lm(formula = log(earnings) ~ height + male + wexp + tenure, data = df_earnings)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-2.18146	-0.35799	-0.02521	0.32146	2.13918

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.984240	0.539487	1.824	0.068649 .
height	0.022712	0.008247	2.754	0.006087 **
male	0.170522	0.070357	2.424	0.015695 *
wexp	0.005415	0.005858	0.924	0.355775
tenure	0.013359	0.003972	3.364	0.000824 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5528 on 535 degrees of freedom

Multiple R-squared: 0.1244, Adjusted R-squared: 0.1179

F-statistic: 19.01 on 4 and 535 DF, p-value: 1.255e-14

```
linearHypothesis(mdl,c('wexp=tenure')) ## Testing one hypothesis
```

Linear hypothesis test

Hypothesis:

wexp - tenure = 0

Model 1: restricted model

Model 2: log(earnings) ~ height + male + wexp + tenure

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	536	163.77				
2	535	163.48	1	0.29223	0.9564	0.3285

```
linearHypothesis(mdl,c('height=0','wexp=tenure')) ## Testing two hypotheses jointly
```

Linear hypothesis test

Hypothesis:

```

height = 0
wexp - tenure = 0

Model 1: restricted model
Model 2: log(earnings) ~ height + male + wexp + tenure

   Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     537 166.15
2     535 163.48  2     2.6769 4.3802 0.01297 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Notice that the `linearHypothesis()` function returns an F-statistic even when testing a single hypothesis. It can be shown (see exercises) that the F-statistic for a test of a single hypothesis is the square of the corresponding t-statistic. The two tests will produce identical p-values.

## 9.6 Asymptotic Properties

In addition to showing consistency and asymptotic normality of OLS estimators, the asymptotic properties discussed in this section deal with two additional issues: how do we do hypothesis testing if we cannot assume that the noise terms are normally distributed? And how do we estimate the variance-covariance matrix of  $\hat{\beta}$  if there is conditional heteroskedasticity? (We have up to this point in the chapter only discussed estimation of  $\text{var}[\hat{\beta}]$  under conditional homoskedasticity. We briefly mentioned heteroskedasticity-robust standard errors in previous chapters; we develop the theory in this chapter.)

Recall that a “Law of Large Numbers” gives conditions under which a sample mean converges in probability to the true mean, e.g., Khinchine’s LLN says that if  $\{Z_i\}_{i=1}^N$  are iid with mean  $E[Z_i] = \mu < \infty$ , then  $\bar{Z} \rightarrow_p \mu$ . A “Central Limit Theorem” gives conditions that guarantee convergence in distribution, e.g., if  $\{Z_i\}_{i=1}^N$  are iid with  $E[Z_i] = \mu < \infty$  and  $\text{var}[Z_i] = \sigma^2 < \infty$ , then

$$\sqrt{N}(\bar{Z} - \mu) \rightarrow_d N(0, \sigma^2).$$

In particular, if  $\mu = 0$ , then

$$\sqrt{N}\bar{Z} = \frac{1}{\sqrt{N}} \sum_{i=1}^N Z_i \rightarrow_d N(0, \sigma^2).$$

In this section, we will use multivariate versions of these rules. First we extend the definition of convergence in probability and convergence in distribution to sequences of vectors and matrices of random variables: convergence in probability of a sequence of  $(M \times K)$ -dimensional matrices of random variables means convergence in probability element-by-element. Convergence in distribution of a sequence of  $(K \times 1)$  vectors of random variables means convergence to some multivariate distribution. We have the following results:

- If  $A_N$  is  $(M \times K)$  such that  $A_N \rightarrow_p A$  and  $g(\cdot)$  is a continuous function, then  $g(A_N) \rightarrow_p g(A)$ .
- If  $A_N$  is  $(M \times K)$  such that  $A_N \rightarrow_p A$ , and  $\bar{Z}_N$  is  $(K \times 1)$  such that  $\bar{Z}_N \rightarrow_p \mu$ , then

$$A_N \bar{Z}_N \rightarrow_p A\mu.$$

- If  $\bar{Z}_N \rightarrow_d Z$  (meaning that the distribution of  $\bar{Z}_N$  converges to the distribution of  $Z$ ), then  $A_N \bar{Z}_N \rightarrow_d AZ$ . In the special case that  $Z \sim \text{Normal}_K(0, \Sigma)$ , we have

$$A_N \bar{Z}_N \rightarrow_d \text{Normal}_M(0, A\Sigma A^T).$$

Now let  $Z_i$  be  $(K \times 1)$  and let  $\{Z_i\}_{i=1}^N$  be an iid sample with  $E[Z_i] = \mu$  for all  $i$ . Let  $\bar{Z}_N$  be the sample mean computed from a sample of size  $N$ . Then

$$\bar{Z} = \frac{1}{N} \sum_{i=1}^N Z_i = \begin{bmatrix} (1/N) \sum_{i=1}^N Z_{1,i} \\ (1/N) \sum_{i=1}^N Z_{2,i} \\ \vdots \\ (1/N) \sum_{i=1}^N Z_{K,i} \end{bmatrix} \rightarrow_p \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_K \end{bmatrix} = \mu.$$

If  $\{Z_i\}_{i=1}^N$  is iid with  $E[Z_i] = 0$  and  $\text{var}[Z_i] = \Sigma$  for all  $i$ , then

$$\sqrt{N} \bar{Z} = \frac{1}{\sqrt{N}} \sum_{i=1}^N Z_i = \begin{bmatrix} (1/\sqrt{N}) \sum_{i=1}^N Z_{1,i} \\ (1/\sqrt{N}) \sum_{i=1}^N Z_{2,i} \\ \vdots \\ (1/\sqrt{N}) \sum_{i=1}^N Z_{K,i} \end{bmatrix} \rightarrow_d \text{Normal}_K(0, \Sigma).$$

If  $Z \sim \text{Normal}_K(0, \Sigma)$ , then we write  $\sqrt{N} \bar{Z} \rightarrow_d Z$ .

We will work with the following form of  $\hat{\beta}$ :

$$\begin{aligned} \hat{\beta} &= \left( \frac{1}{N} \sum_{i=1}^N x_i x_i^T \right)^{-1} \left( \frac{1}{N} \sum_{i=1}^N x_i Y_i \right) \\ &= \beta + \left( \frac{1}{N} \sum_{i=1}^N x_i x_i^T \right)^{-1} \left( \frac{1}{N} \sum_{i=1}^N x_i \epsilon_i \right), \end{aligned} \tag{9.16}$$

The asymptotic properties of the OLS estimators depends on the what happens to the sample means in (9.16) as  $N \rightarrow \infty$ . We state a set of assumptions that is a slightly modified version of the basic assumptions we have been using.

### Assumption Set E

(E1) The sample  $\{Y_i, X_{1,i}, X_{2,i}, \dots, X_{K-1,i}\}_{i=1}^N$  are iid draws from a  $K$ -dimensional distribution with the following properties:

- (E2)  $E[x_i x_i^T] = \Sigma_{XX}$  is finite and non-singular,
- (E3)  $E[\epsilon_i | x_i] = 0$  where  $\epsilon_i = Y_i - x_i^T \beta$  for some constants  $\beta$ , and
- (E4)  $E[\epsilon_i^2 x_i x_i^T] = S$  finite and non-singular.

*Remarks:*

- Assumption E1 is the random sampling assumption. Since the draws are iid, the random variables in the  $(K \times K)$  matrix  $x_i x_i^T$  and the  $(K \times 1)$  vector  $x_i \epsilon_i$  are also iid over  $i$ .
- Assumption E2 is the assumption that there are no perfect correlations among the regressors in expectation. Together with the random sampling assumption, it guarantees

that

$$\frac{1}{N} \sum_{i=1}^N x_i x_i^T$$

converges in probability to something that is finite and has an inverse.

- The assumption E3 says a number of things: first,  $\epsilon_i$  is zero mean for all  $i$ . Second, it says that  $\epsilon_i$  is uncorrelated with each of the  $K - 1$  regressors in  $x_i$ , i.e.,  $E[\epsilon_i x_i] = 0$ . This, and the random sampling assumption, means that

$$\frac{1}{N} \sum_{i=1}^N x_i \epsilon_i \rightarrow_p 0.$$

The implication  $E[\epsilon_i x_i] = 0$  of Assumption D3 is the key to obtaining consistent OLS coefficient estimators.

- Since  $E[\epsilon_i x_i] = 0$ , the expectation  $E[\epsilon_i^2 x_i x_i^T]$  in Assumption D4 is the variance-covariance matrix of  $x_i \epsilon_i$ .
- The assumptions impose unconditional homoskedasticity, but **allow for conditional heteroskedasticity**. Since  $\{Y_i, X_{1,i}, X_{2,i}, \dots, X_{K-1,i}\}_{i=1}^N$  are iid draws,  $\{\epsilon_i\}_{i=1}^N$  are also iid. The “identically distributed” part implies  $\text{var}[\epsilon]$  is constant, so there is unconditional homoskedasticity. However, we allow for conditional heteroskedasticity, i.e., the *conditional* variance may depend on the regressors. For example, suppose  $E[\epsilon_i^2 | x_i] = \sigma^2 X_{1,i}^2$  so the noise variance depends on  $X_{1,i}$  (there is conditional heteroskedasticity). However,  $E[E[\epsilon_i^2 | x_i]] = \sigma^2 E[X_{1,i}^2]$ , which is constant if  $X_{1,i}$  is iid.
- We do *not* assume that the noise terms are normally distributed.

Given Assumption Set E,  $\hat{\beta}$  is consistent:

$$\hat{\beta} = \beta + \underbrace{\left( \frac{1}{N} \sum_{i=1}^N x_i x_i^T \right)^{-1}}_{\rightarrow_p \Sigma_{XX}^{-1}} \underbrace{\left( \frac{1}{N} \sum_{i=1}^N x_i \epsilon_i \right)}_{\rightarrow_p 0} \rightarrow_p \beta$$

This is basically the general version of the simple linear regression argument that

$$\hat{\beta}_1 = \beta_1 + \frac{\text{sample cov}(X_i, \epsilon_i)}{\text{sample var}(X_i)} \rightarrow_p \beta_1$$

if  $\text{cov}[X_i, \epsilon_i] = 0$  and  $\text{var}[X_i]$  is finite, and if their sample counterparts converge in probability to them.

Since  $\hat{\beta}_1$  is consistent, its distribution is essentially degenerate in the limit. To talk about limiting distributions, we have to scale  $\hat{\beta}_1$ . We use

$$\sqrt{N}(\hat{\beta} - \beta) = \left( \frac{1}{N} \sum_{i=1}^N x_i x_i^T \right)^{-1} \left( \frac{1}{\sqrt{N}} \sum_{i=1}^N x_i \epsilon_i \right),$$

Our assumptions and the CLT imply

$$\frac{1}{\sqrt{N}} \sum_{i=1}^N x_i \epsilon_i \rightarrow_d \text{Normal}_K(0, S),$$

therefore

$$\sqrt{N}(\hat{\beta} - \beta) = \underbrace{\left( \frac{1}{N} \sum_{i=1}^N x_i x_i^T \right)^{-1}}_{\rightarrow_p \Sigma_{XX}^{-1}} \underbrace{\left( \frac{1}{\sqrt{N}} \sum_{i=1}^N x_i \epsilon_i \right)}_{\rightarrow_d \text{Normal}_K(0, S)} \rightarrow_d \text{Normal}_K(0, \Sigma_{XX}^{-1} S \Sigma_{XX}^{-1})$$

That is,  $\hat{\beta}$  is consistent, with asymptotic variance  $\text{avar}[\hat{\beta}] = \Sigma_{XX}^{-1} S \Sigma_{XX}^{-1}$ . This result justifies the approximation

$$\text{var}[\hat{\beta}] \approx (1/N) \Sigma_{XX}^{-1} S \Sigma_{XX}^{-1}.$$

An obvious estimator for  $\Sigma_{XX}$  is

$$\hat{\Sigma}_{XX} = \frac{1}{N} \sum_{i=1}^N x_i x_i^T = (1/N) X^T X$$

Some additional assumptions (see advanced econometrics textbooks) guarantee

$$\hat{S} = \frac{1}{N} \sum_{i=1}^N \hat{\epsilon}_i^2 x_i x_i^T \rightarrow_p S. \quad (9.17)$$

This allows us to consistently estimate the asymptotic variance of  $\hat{\beta}$  by

$$\widehat{\text{avar}}[\hat{\beta}] = \left( \frac{1}{N} \sum_{i=1}^N x_i x_i^T \right)^{-1} \left( \frac{1}{N} \sum_{i=1}^N \hat{\epsilon}_i^2 x_i x_i^T \right) \left( \frac{1}{N} \sum_{i=1}^N x_i x_i^T \right)^{-1},$$

and justifies the use of

$$\begin{aligned} \widehat{\text{var}}_{HC0}[\hat{\beta}] &= \frac{1}{N} \widehat{\text{avar}}[\hat{\beta}] \\ &= \frac{1}{N} \left( \frac{1}{N} \sum_{i=1}^N x_i x_i^T \right)^{-1} \left( \frac{1}{N} \sum_{i=1}^N \hat{\epsilon}_i^2 x_i x_i^T \right) \left( \frac{1}{N} \sum_{i=1}^N x_i x_i^T \right)^{-1} \\ &= \left( \sum_{i=1}^N x_i x_i^T \right)^{-1} \left( \sum_{i=1}^N \hat{\epsilon}_i^2 x_i x_i^T \right) \left( \sum_{i=1}^N x_i x_i^T \right)^{-1} \end{aligned} \quad (9.18)$$

as an estimator for the variance of  $\hat{\beta}$ . The variance estimator in (9.18) is a “Heteroskedasticity-Consistent Variance Estimator”. There are several versions. The version presented in (9.18) is often referred to as “HC0”, which is why we label it as such. Other versions will be discussed in the exercises. Because of its form, (9.18) is often called a “sandwich” estimator.

If there is conditional heteroskedasticity in the noise terms, the usual OLS variance estimator  $\widehat{\sigma}^2 (X^T X)^{-1}$  is not appropriate since  $\widehat{\sigma}^2 ((1/N) X^T X)^{-1}$  is not a consistent estimator for the asymptotic variance. On the other hand, the variance estimator (9.18) remains consistent even

if in fact the noise terms are conditionally homoskedastic. In this sense it is safer to use (9.18) for estimating the estimator variance if there is any possibility of conditional heteroskedasticity.

We have already noted that OLS estimators are efficient if there is conditional homoskedasticity. If there is conditional heteroskedasticity, then OLS estimators are no longer efficient. (The formula (9.18) allows us to estimate the estimator *variance* consistently, but doesn't do anything about the efficiency of  $\hat{\beta}$  itself.) We have already addressed how we might try to get efficient estimators in the previous chapter.

We can use the heteroskedasticity consistent variance estimator to construct heteroskedasticity robust  $t$  and  $F$  statistics, by replacing  $\widehat{\text{var}}[\hat{\beta}|X]$  in (9.12) and (9.15) with the heteroskedasticity robust variance estimator in (9.18).

**Example 9.7.** We compute below the HC0 Heteroskedasticity Robust covariance matrix for the regression in Example 9.6.

```

hatS = 0
for (i in 1:N){
  xi = as.matrix(X[i,])
  hatS = hatS + ehat[i]^2*xi%*t(xi)
}
varhat_HC0 <- XTXinv %*% hatS %*% XTXinv ## XTXinv computed earlier
round(varhat_HC0,6)

```

	const	height	male	wexp	tenure
const	0.281450	-0.004322	0.024655	-0.000212	0.000364
height	-0.004322	0.000069	-0.000390	-0.000004	-0.000005
male	0.024655	-0.000390	0.004595	-0.000040	0.000033
wexp	-0.000212	-0.000004	-0.000040	0.000033	-0.000006
tenure	0.000364	-0.000005	0.000033	-0.000006	0.000013

The function `hccm()` from the `car` package also calculates this, as does the `vcovHC()` function from the `sandwich` package. We use the latter.

```

varhat_HC0_v2 = vcovHC(mdl,type="HC0")
round(varhat_HC0_v2,6)

```

	(Intercept)	height	male	wexp	tenure
(Intercept)	0.281450	-0.004322	0.024655	-0.000212	0.000364
height	-0.004322	0.000069	-0.000390	-0.000004	-0.000005
male	0.024655	-0.000390	0.004595	-0.000040	0.000033
wexp	-0.000212	-0.000004	-0.000040	0.000033	-0.000006
tenure	0.000364	-0.000005	0.000033	-0.000006	0.000013

The heteroskedasticity-robust standard errors are the square roots of the diagonal of this matrix. The table below presents the heteroskedasticity robust  $t$ -stats and recomputes the corresponding  $p$ -values.

```

sebhat <- sqrt(diag(varhat_HC0)) # se(bhat)
tbhat <- bhat/sebhat
pval <- 2*(1-pt(abs(tbhat),N-K))
star <- rep("", K)
star[pval<0.1] <- "."

```



```

star[pval<0.05] <- "*"
star[pval<0.01] <- "**"
star[pval<0.001] <- "***"
Results <- data.frame("Estimate"=bhat, "HCO Std. Err."=sebhat, "t-val"=tbhat, "p-val"=pval, star)
names(Results)[K] <- ""
print(Results, digits=6, right=FALSE)
cat("---\nSignif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1\n")

```

```

      Estimate   HCO.Std.Err. t.val   p.val
const 0.98423994 0.53051905   1.85524 0.064111865 .
height 0.02271202 0.00828080   2.74273 0.006297216 **
male   0.17052245 0.06778799   2.51553 0.012177295 *
wexp   0.00541464 0.00574455   0.94257 0.346326070
tenure 0.01335911 0.00365123   3.65880 0.000278498 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Comparing these results with the previous ones, we find that the heteroskedasticity-robust standard errors are not very different from those estimated using the usual OLS formulas under conditional homoskedasticity. This suggests that heteroskedasticity is not a significant issue in this regression. Nonetheless, for illustration purposes we use the heteroskedasticity-robust robust t- and F-tests to test the hypotheses in Example 9.6. In the code below, we compute the robust F-test manually, as well as using the `linearHypothesis()` function from the `car` package.

```

F <- t(Rb) %*% solve(R%*%varhat_HCO%*%t(R)) %*% Rb / J
Fpval <- (1-pf(F,J,N-K))
cat("H0 height = 0 and wage = tenure: F-stat ",round(F,4),",", pval", round(Fpval,6))

```

```
H0 height = 0 and wage = tenure: F-stat  4.3413 , pval 0.013481
```

```
linearHypothesis(mdl,c('height=0','wexp=tenure'), vcov=varhat_HCO)
```

Linear hypothesis test

Hypothesis:

height = 0

wexp - tenure = 0

Model 1: restricted model

Model 2: log(earnings) ~ height + male + wexp + tenure

Note: Coefficient covariance matrix supplied.

```

  Res.Df Df    F Pr(>F)
1     537
2     535  2 4.3413 0.01348 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

### 9.7 Exercises

**Exercise 9.1.** Find expressions for  $\text{var}[\hat{\beta}_0|X]$  and  $\text{cov}[\hat{\beta}_0, \hat{\beta}_1|X]$  in the simple linear regression

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i, \quad i = 1, 2, \dots, N$$

where  $E[\epsilon_i|X] = 0$  and  $E[\epsilon_i^2|X] = \sigma^2$  for all  $i = 1, 2, \dots, N$ , and  $E[\epsilon_i \epsilon_j|X] = 0$  for all  $i \neq j$ ,  $i, j = 1, 2, \dots, N$ . What is the sign of  $\text{cov}[\hat{\beta}_0, \hat{\beta}_1|X]$ ?

**Exercise 9.2.** Let  $X$  be a  $(N \times K)$  full column rank matrix. Show that

$$M = I - X(X^T X)^{-1} X^T$$

is symmetric and idempotent, with rank  $N - K$ .

**Exercise 9.3.** Show that the residuals from the regression  $Y_i = \beta_0 + \epsilon_i$ ,  $i = 1, 2, \dots, N$  can be written as

$$\hat{\epsilon} = M_0 y$$

where

$$M_0 = I - i_N(i_N^T i_N)^{-1} i_N^T.$$

Show by direct computation that  $M_0 y$  is the vector of deviations from means, i.e.,

$$M_0 y = (I - i_N(i_N^T i_N)^{-1} i_N^T) y = \begin{bmatrix} Y_1 - \bar{Y} \\ Y_2 - \bar{Y} \\ \vdots \\ Y_N - \bar{Y} \end{bmatrix}.$$

Show that  $y^T M_0 y = \sum_{i=1}^N (Y_i - \bar{Y})^2$

**Exercise 9.4.** Show that

$$y^T M_0 y = \hat{y}^T M_0 \hat{y} + \hat{\epsilon}^T \hat{\epsilon}.$$

This is the  $SST = SSR + SSE$  equality that forms the basis of the  $R^2$ .

**Exercise 9.5.** Consider the regression  $E^i = X\beta + \varepsilon$ , where  $E^i$  is the  $(N \times 1)$  vector comprising all zeros except for a ‘1’ in the  $i$ th position. Let the matrix  $X$  be  $(N \times K)$  full column rank.

a. Show that the fitted values  $\widehat{E}^i$  has the expression

$$\widehat{E}^i = X(X^T X)^{-1} x_i$$

where  $x_i^T$  is the  $i$ th row of the  $X$  matrix.

b. Define the ‘leverage’ of observation  $i$  to be

$$h_i = x_i^T (X^T X)^{-1} x_i.$$

Show that  $0 \leq h_i \leq 1$ . *Hint: Use part (a) and the “Pythagoras’s Theorem” result in (9.7).*

c. Explain why  $\sum_{i=1}^N h_i$  is the trace of the matrix  $P = X(X^T X)^{-1} X^T$ . Show that  $\sum_{i=1}^N h_i = K$ . (In other words, the “average value” of  $h_i$  is  $K/N$ .)

*Remark: it can be shown that*

$$\hat{\beta} - \hat{\beta}_{-i} = \left( \frac{1}{1 - h_i} \right) (X^T X)^{-1} x_i \hat{\epsilon}_i$$

where  $\hat{\beta}_{-i}$  is the OLS estimator obtained when observation  $i$  is left out. An observation with  $h_i$  close to 1 therefore has very high leverage, or “influential”.

**Exercise 9.6.** Consider the linear regression  $y = X\beta + \varepsilon$  where  $E[\varepsilon|X] = 0$  and  $E[\varepsilon\varepsilon^T|X] = \sigma^2 I$ . The fact that  $\widetilde{\sigma}^2 = \frac{1}{N} \sum_{i=1}^N \hat{\epsilon}_i^2$  is biased implies that each individual  $\hat{\epsilon}_i^2$  must in general be a biased estimator for  $\sigma^2$ . Show that

$$E[\hat{\epsilon}_i^2] = (1 - h_i)\sigma^2.$$

*Hint: use  $\hat{\epsilon}_i^2 = (E^i)^T \hat{\varepsilon} \hat{\varepsilon}^T E^i$  and  $\hat{\varepsilon} = M\varepsilon$  where  $M = I - X(X^T X)^{-1} X^T$ .*

**Exercise 9.7.** The “HC0” version of the heteroskedasticity-consistent standard errors given in (9.18) is sometimes criticized for not taking into consideration the fact that  $K$  degrees of freedom are used in the computation of  $\hat{\epsilon}_i$ . Another version proposes to take this into account by estimating  $S$  with

$$\hat{S}_1 = \frac{1}{N - K} \sum_{i=1}^N \hat{\epsilon}_i^2 x_i x_i^T$$

which also consistently estimates  $S$ .

- a. Show that using  $\hat{S}_1$  instead of  $\hat{S}$  in Eq. 9.17 results in the Heteroskedasticity-Consistent variance estimator

$$\widehat{\text{var}}_{HC1}[\hat{\beta}] = \left( \sum_{i=1}^N x_i x_i^T \right)^{-1} \left( \frac{N}{N - K} \sum_{i=1}^N \hat{\epsilon}_i^2 x_i x_i^T \right) \left( \sum_{i=1}^N x_i x_i^T \right)^{-1}. \quad (9.19)$$

Amend the code in Example 9.7 to use the HC1 version of the variance estimator, and verify your results using the `vcovHC()` function.

- b. Another version, based on the result in Exercise 9.6, is

$$\widehat{\text{var}}_{HC2}[\hat{\beta}] = \left( \sum_{i=1}^N x_i x_i^T \right)^{-1} \left( \sum_{i=1}^N \frac{\hat{\epsilon}_i^2}{1 - h_i} x_i x_i^T \right) \left( \sum_{i=1}^N x_i x_i^T \right)^{-1}. \quad (9.20)$$

Amend the code in Example 9.7 to use the HC2 version of the variance estimator, and verify your results using the `vcovHC()` function.

- c. The result in Exercise 9.6, of course, assumes conditional homoskedasticity. Yet another version is

$$\widehat{\text{var}}_{HC3}[\hat{\beta}] = \left( \sum_{i=1}^N x_i x_i^T \right)^{-1} \left( \sum_{i=1}^N \frac{\hat{\epsilon}_i^2}{(1 - h_i)^2} x_i x_i^T \right) \left( \sum_{i=1}^N x_i x_i^T \right)^{-1}. \quad (9.21)$$

This version puts more weight on observations that are more influential. Amend the code in Example 9.7 to use the HC3 version of the variance estimator, and verify your results using the `vcovHC()` function.

**Exercise 9.8.** Consider the linear regression model under Assumption Set E. Show, by application of the Law of Iterated Expectations, that if the noise terms are conditionally homoskedastic, i.e., if  $E[\epsilon_i^2|x_i] = \sigma^2$ , then

$$S = E[\epsilon_i^2 x_i x_i^T] = \sigma^2 \Sigma_{XX}.$$

and that the asymptotic variance of  $\hat{\beta}$  then becomes

$$\text{avar}[\hat{\beta}] = \sigma^2 \Sigma_{XX}^{-1}.$$

*Remark:* this result is in accordance with the fact that under conditional homoskedasticity,  $\text{var}[\hat{\beta}|X] = \sigma^2 (X^T X)^{-1}$ .

**Exercise 9.9.** Suppose we have  $y = X\beta + \varepsilon$  with  $E[\varepsilon|X] = 0$ . Suppose  $\text{var}[\varepsilon|X]$  is possibly heteroskedastic and correlated, but known up to some parameter  $\sigma^2$ , i.e.,  $\text{var}[\varepsilon|X] = \sigma^2 \Omega$ , where  $\Omega$  is known (it may involve elements of  $X$ , but with no unknown parameters). Of course,  $\Omega$  must be symmetric and positive definite. For example, we may have

$$\text{var}[\varepsilon|X] = \sigma^2 \Omega_1 = \sigma^2 \begin{bmatrix} 1/X_{k,1}^2 & 0 & 0 & \dots & 0 \\ 0 & 1/X_{k,2}^2 & 0 & \dots & 0 \\ 0 & 0 & 1/X_{k,3}^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1/X_{k,N}^2 \end{bmatrix}$$

where  $X_{k,i}$  is the  $i$ th observation of variable  $X_k$ . Another example is

$$\text{var}[\varepsilon|X] = \sigma^2 \Omega_2 = \frac{\sigma^2}{1-0.9^2} \begin{bmatrix} 1 & 0.9 & 0.9^2 & \dots & 0.9^{N-1} \\ 0.9 & 1 & 0.9 & \dots & 0.9^{N-2} \\ 0.9^2 & 0.9 & 1 & \dots & 0.9^{N-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.9^{N-1} & 0.9^{N-2} & 0.9^{N-3} & \dots & 1 \end{bmatrix}$$

Since  $\Omega$  is symmetric and positive definite, we can find  $P$  such that

$$P\Omega P^T = I$$

a. Find  $P$  such that  $P\Omega_1 P^T = I$ .

b. Verify that  $P\Omega_2 P^T = I$  where

$$P = \begin{bmatrix} \sqrt{1-0.9^2} & 0 & 0 & \dots & 0 & 0 \\ -0.9 & 1 & 0 & \dots & 0 & 0 \\ 0 & -0.9 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -0.9 & 1 \end{bmatrix}$$

*Remark:* Transforming the regression  $y = X\beta + \varepsilon$ ,  $E[\varepsilon|X] = 0$ ,  $\text{var}[\varepsilon|X] = \sigma^2 \Omega$ , by premultiply-

ing by  $P$ , where  $P$  is the matrix such that  $P\Omega P^T = I$ , produces the regression

$$Py = PX\beta + P\varepsilon$$

$$\text{or} \quad y^* = X^*\beta + \varepsilon^*$$

where  $y^* = Py$ ,  $X^* = PX$  and  $\varepsilon^* = P\varepsilon$ . Let  $\hat{\beta}^{gl_s}$  be the estimator of  $\beta$  obtained by applying OLS to the transformed regression. This is called the “Generalized Least Squares”, or GLS, estimator for  $\beta$ . In general,  $\Omega$  will contain parameters that must (somehow) be estimated. Methods where  $\Omega$  is replaced by some estimated  $\hat{\Omega}$  are called “Feasible Generalized Least Squares” or FGLS.\*

### Exercise 9.10.

- Show that the  $F$ -statistic for testing a single restriction, i.e., when  $J = 1$ , is equal to the square of the  $t$ -statistic for testing the same hypothesis. (Hint: compare (9.12) and (9.15) after setting  $R = r^T$  in the  $F$ -statistic.)
- Suppose you have the regressions

$$[A] \quad Y_i = \beta_0 + \beta_1 X_{1,i} + \dots + \beta_{K-1} X_{K-1,i} + \epsilon_i,$$

$$[B] \quad Y_i = \beta_0 + \beta_1 X_{1,i} + \dots + \beta_{K-1} X_{K-1,i} + \beta_K X_{K,i} + \epsilon_i.$$

Two possible ways of deciding whether or not to include variable  $X_{K,i}$  in the regression is to do a  $t$ -test (or an  $F$ -test) of the hypothesis  $\beta_K = 0$ . Another way is to see whether or not the Adjusted- $R^2$  in [B] is greater than the Adjusted- $R^2$  in [A]. Show that the latter method is equivalent to including  $X_{K,i}$  if the absolute value of the  $t$ -statistic for  $\hat{\beta}_K$  in [B] is greater than 1. *Hint: use the version of the  $F$ -statistic in (9.13).*

**Exercise 9.11.** Suppose  $Y_i = x_i^T \beta + \epsilon_i$ ,  $i = 1, 2, \dots, N$ , with  $E[\varepsilon|X] = 0$  and  $E[\varepsilon\varepsilon^T|X] = \sigma^2 I$ . Let  $\hat{\beta}$  be the OLS estimator for  $\beta$ . Suppose we predict  $Y$  at  $x = x_0$  using  $\hat{Y}(x_0) = x_0^T \hat{\beta}$ . The prediction error is

$$\hat{e}(x_0) = Y(x_0) - \hat{Y}(x_0) = x_0^T \beta + \epsilon_0 - x_0^T \hat{\beta} = x_0^T (\beta - \hat{\beta}) + \epsilon_0.$$

- Derive an expression for the prediction error variance.
- Specialize your answer in (a) to the simple linear regression  $Y_i = \beta_0 + \beta_1 X_{1,i} + \epsilon_i$ , predicting  $Y$  at  $X_1 = x_1^0$ . Show that the prediction error variance is

$$\hat{e}(x_1^0) = \sigma^2 \left( 1 + \frac{1}{N} + \frac{(x_1^0 - \bar{X}_1)^2}{\sum_{i=1}^N (X_{1,i} - \bar{X}_1)^2} \right).$$

## 9.8 Appendix

In this appendix, we prove the equality (9.14). Let the regression model be  $y = X\beta + \varepsilon$ , and let  $\hat{\beta}^r$  be the least squares estimator for  $\beta$  subject to the restriction that  $R\beta = r$ . We first show that

$$\hat{\beta}^r = \hat{\beta}^{ols} + (X^T X)^{-1} R^T [R(X^T X)^{-1} R^T]^{-1} (r - R\hat{\beta}^{ols})$$

where  $\hat{\beta}^{ols}$  is the usual unrestricted OLS estimator. The restricted SSR minimization problem is

$$\hat{\beta}^r = \operatorname{argmin}_{\hat{\beta}} (y - X\hat{\beta})^T (y - X\hat{\beta}) \text{ subject to } R\hat{\beta} - r = 0.$$

The Lagrangian is

$$L = (y - X\hat{\beta})^T (y - X\hat{\beta}) + 2(r^T - \hat{\beta}^T R^T) \lambda.$$

The FOC is

$$\begin{aligned} \left. \frac{\partial L}{\partial \hat{\beta}} \right|_{\hat{\beta}^r, \hat{\lambda}} &= -2X^T y + 2X^T X \hat{\beta}^r - 2R^T \hat{\lambda} = 0 \\ \left. \frac{\partial L}{\partial \hat{\lambda}} \right|_{\hat{\beta}^r, \hat{\lambda}} &= 2(r - R\hat{\beta}^r) = 0 \end{aligned}$$

The second equation in the FOC merely says that the restriction must hold. The first equation in the FOC implies

$$\hat{\beta}^r = (X^T X)^{-1} X^T y + (X^T X)^{-1} R^T \hat{\lambda} = \hat{\beta}^{ols} + (X^T X)^{-1} R^T \hat{\lambda}.$$

Multiplying throughout by  $R$  gives

$$R\hat{\beta}^r = R\hat{\beta}^{ols} + R(X^T X)^{-1} R^T \hat{\lambda}.$$

It follows that

$$\begin{aligned} \hat{\lambda} &= [R(X^T X)^{-1} R^T]^{-1} (R\hat{\beta}^r - R\hat{\beta}^{ols}) \\ &= [R(X^T X)^{-1} R^T]^{-1} (r - R\hat{\beta}^{ols}), \end{aligned}$$

and therefore

$$\hat{\beta}^r = \hat{\beta}^{ols} + (X^T X)^{-1} R^T [R(X^T X)^{-1} R^T]^{-1} (r - R\hat{\beta}^{ols}). \quad (9.22)$$

Now let

$$\begin{aligned} \hat{\varepsilon}_r &= y - X\hat{\beta}^r \\ &= y - X\hat{\beta}^{ols} + X\hat{\beta}^{ols} - X\hat{\beta}^r \\ &= \hat{\varepsilon}_{ols} + X(\hat{\beta}^{ols} - \hat{\beta}^r). \end{aligned} \quad (9.23)$$

Since (unrestricted) OLS residuals are orthogonal to the regressors, we have

$$\hat{\varepsilon}_{ols}^T \hat{\varepsilon}_r = \hat{\varepsilon}_{ols}^T \hat{\varepsilon}_{ols} + \hat{\varepsilon}_{ols}^T X(\hat{\beta}^{ols} - \hat{\beta}^r) = \hat{\varepsilon}_{ols}^T \hat{\varepsilon}_{ols}.$$

Therefore

$$(\hat{\varepsilon}_r - \hat{\varepsilon}_{ols})^T (\hat{\varepsilon}_r - \hat{\varepsilon}_{ols}) = \hat{\varepsilon}_r^T \hat{\varepsilon}_r - \hat{\varepsilon}_{ols}^T \hat{\varepsilon}_{ols}. \quad (9.24)$$

Finally, use (9.22), (9.23) and (9.24) to show (9.14).

## Chapter 10

### Instrumental Variables and Generalized Method of Moments

We present the concept of instrumental variables, and an estimation method called Generalized Method of Moments (GMM). This extended framework enables consistent estimation of economic relationships in situations where there are endogeneity problems, i.e., where (for whatever reason) there are correlations between the noise term and one or more regressors. However, it requires the availability of good instrumental variables, which may not be so easy to find.

The R code in this chapter uses the packages

```
library(tidyverse)
library(readxl)
library(lmtest)
library(sandwich)
library(car)
```

We will also use Stata to verify our computations.

#### 10.1 Instrumental Variables and the IV Estimator

Suppose the regression equation of interest is

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i, \quad i = 1, 2, \dots, N.$$

We continue to assume independent observations, but suppose that  $X_i$  and  $\epsilon_i$  are correlated, with the consequence that the OLS estimator for  $\beta_1$  is inconsistent (and biased). Suppose, however, that there are observations of a third variable  $Z_i$  that are correlated with  $X_i$  but not with  $\epsilon_i$ . We can use this variable to estimate  $\beta_1$  consistently.

We continue to assume that  $\epsilon_i$  is zero mean for all  $i$  (this is an innocuous assumption since the regression includes an intercept term). We therefore have the following “moment conditions” for all  $i$ :

$$\begin{aligned} E[\epsilon_i] &= E[Y_i - \beta_0 - \beta_1 X_i] = 0 \\ E[\epsilon_i Z_i] &= E[(Y_i - \beta_0 - \beta_1 X_i) Z_i] = 0. \end{aligned} \tag{10.1}$$

Define the “IV” estimators of  $\beta_0$  and  $\beta_1$  to be those values that solve the sample analogue of the moment conditions:

$$\begin{aligned} (1/N) \sum_{i=1}^N (Y_i - \hat{\beta}_0^{iv} - \hat{\beta}_1^{iv} X_i) &= 0 \\ (1/N) \sum_{i=1}^N (Y_i - \hat{\beta}_0^{iv} - \hat{\beta}_1^{iv} X_i) Z_i &= 0. \end{aligned} \tag{10.2}$$

This gives the estimators

$$\begin{aligned} \hat{\beta}_0^{iv} &= \bar{Y} - \hat{\beta}_1^{iv} \bar{X} \\ \hat{\beta}_1^{iv} &= \frac{\sum_{i=1}^N (Z_i - \bar{Z})(Y_i - \bar{Y})}{\sum_{i=1}^N (Z_i - \bar{Z})(X_i - \bar{X})} \end{aligned} \tag{10.3}$$

which are consistent for  $\beta_0$  and  $\beta_1$  respectively. We show consistency of  $\hat{\beta}_1^{iv}$ . Write  $\hat{\beta}_1^{iv}$  as

$$\hat{\beta}_1^{iv} = \beta_1 + \frac{\sum_{i=1}^N (Z_i - \bar{Z})\epsilon_i}{\sum_{i=1}^N (Z_i - \bar{Z})(X_i - \bar{X})}. \quad (10.4)$$

Dividing the numerator and denominator of the second term by  $N$  shows that it is the ratio of the sample covariance of  $Z_i$  and  $\epsilon_i$  to the sample covariance of  $Z_i$  and  $X_i$ . As sample sizes increase, these sample moments converge in probability to their population counterparts. By assumption, the population covariance of  $Z_i$  and  $\epsilon_i$  is zero, whereas the population covariance of  $Z_i$  and  $X_i$  is not zero. Therefore  $\hat{\beta}_1^{iv}$  converges in probability to  $\beta_1$ .

Another way to see consistency of  $\hat{\beta}_1^{iv}$  is to note that (10.1) uniquely identifies  $\beta_0$  and  $\beta_1$ . Write (10.1) as

$$\begin{aligned} E[Y_i] - \beta_0 - \beta_1 E[X_i] &= 0 \\ E[Z_i Y_i] - \beta_0 E[Z_i] - \beta_1 E[Z_i X_i] &= 0. \end{aligned} \quad (10.5)$$

Solving gives

$$\begin{aligned} \beta_0 &= E[Y_i] - \beta_1 E[X_i] \\ \beta_1 &= \frac{E[Z_i Y_i] - E[Z_i]E[Y_i]}{E[Z_i X_i] - E[Z_i]E[X_i]} = \frac{\text{cov}[Z_i, Y_i]}{\text{cov}[Z_i, X_i]}. \end{aligned} \quad (10.6)$$

The solution requires  $\text{cov}[Z_i, X_i] \neq 0$ , which we assume. Because the sample moments in (10.2) converge to their population counterparts,  $\hat{\beta}_0^{iv}$  and  $\hat{\beta}_1^{iv}$  converge to their population values.

Although  $\hat{\beta}_1^{iv}$  is consistent, it is biased. This is easily seen from (10.4). Taking conditional expectations does not remove the second term, since  $E[\epsilon_i | x, z] \neq 0$ .

Because  $X_i$  is correlated with  $\epsilon_i$ , we say it is “endogenous” (no matter what the reason for the correlation). Because  $Z_i$  is uncorrelated with  $\epsilon_i$ , we say it is “exogenous”. Because we use  $Z_i$  to identify our equation, we call it an “instrumental variable”. A valid instrumental variable, or “instrument”, is one that is exogenous but correlated with the regressor. The IV estimator is also an example of what we would call a “Method of Moments” estimator.

If  $X_i$  is not endogenous, then we can use it “as its own instrument”, i.e., set  $Z_i = X_i$ . The sample moment conditions in (10.2) then become

$$\begin{aligned} \sum_{i=1}^N (Y_i - \hat{\beta}_0^{iv} - \hat{\beta}_1^{iv} X_i) &= 0 \\ \sum_{i=1}^N (Y_i - \hat{\beta}_0^{iv} - \hat{\beta}_1^{iv} X_i) X_i &= 0 \end{aligned} \quad (10.7)$$

which you will recognize as the first-order conditions for OLS estimation.

Instruments can arise from natural experiments, and incidental features of specific applications. Some examples of instruments include proximity to college, quarter of birth, and parents’ years of schooling as instruments for subject’s years of schooling; variation in state cigarette taxes for maternal smoking; Vietnam war draft lottery number for veteran status, sibling sex composition for fertility (see Angrist and Krueger (2001) for a discussion of instruments from natural experiments). They can also arise from structural characteristics of particular economic relationships. We consider a supply and demand example below.



## 10.2 A Simultaneous Equation Example

Suppose that the intention is to estimate the demand function for a certain good. Suppose that the market for the good can be represented by the demand and supply system:

$$\begin{aligned} Q_t^d &= \delta_0 + \delta_1 P_t + \epsilon_t^d && \text{(Demand Eq } \delta_1 < 0) \\ Q_t^s &= \alpha_0 + \alpha_1 P_t + \epsilon_t^s && \text{(Supply Eq } \alpha_1 > 0) \\ Q_t^s &= Q_t^d, && \text{(Market Clearing)} \end{aligned} \quad (10.8)$$

As discussed in Chapter 5, the observed quantities and prices occur at the intersection of the demand and supply equations. Because of this, observed prices and quantities are not representative of either the demand or supply functions.

We illustrate this phenomenon with a simulation of the case where  $\alpha_0 = 10$ ,  $\alpha_1 = 5$ ,  $\delta_0 = 80$  and  $\delta_1 = -4$ , and where the mutually independent demand and supply shocks  $\epsilon_t^d$  and  $\epsilon_t^s$  are iid  $\text{Normal}(0, 6)$  and  $\text{Normal}(0, 8)$  respectively. The demand and supply shocks lead to shifts in the demand and supply functions. We draw the demand and supply functions for 20 periods and indicate their intersection points, which are the data that we observe.

```
set.seed(9)
nsim <- 20
ed <- rnorm(nsim, 0, 6); es <- rnorm(nsim, 0, 8)
p <- seq(from=3, to=12, by=0.1)
a0 <- 10; a1 <- 5; b0 <- 80; b1 <- -4
plt1 <- ggplot()
for (i in 1:nsim){
  qs <- a0 + a1*p + es[i]
  qd <- b0 + b1*p + ed[i]
  peq <- (a0-b0)/(b1-a1) + (es[i]-ed[i])/(b1-a1)
  qeq <- (b0+b1*(a0-b0)/(b1-a1)) + (b1*es[i]-a1*ed[i])/(b1-a1)
  dat <- data.frame("qs"=qs, "qd"=qd, "p"=p) %>% arrange(p)
  plt1 <- plt1 +
    geom_line(data=dat, aes(y=qs,x=p), color="gray", alpha=0.4) +
    geom_line(data=dat, aes(y=qd,x=p), color="gray", alpha=0.4) +
    annotate("point", y=qeq, x=peq, size=2) + ylab("q") +
    theme_classic() + theme(axis.title.y = element_text(angle = 0))
}; plt1
```

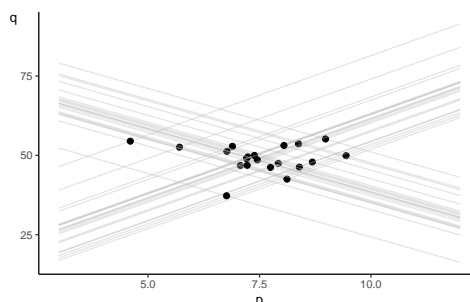


Figure 10.1: Demand and Supply Example

You can see that the observed prices and quantities reflect neither the demand nor the supply functions, and a regression of quantity on price will produce a slope coefficient that will turn out to be some average of  $\alpha_1$  and  $\beta_1$ . The issue here is that both prices and quantity are simultaneously determined. Price, in particular, is not an exogenous variable. Variation in the data comes about because of the demand and supply shocks, both of which affect both quantities and prices. In a regression of quantity on price, the regressor (price) will be correlated with the regression noise term.

We have already demonstrated the inconsistency of the OLS estimator mathematically in chapter 5. We review the argument briefly. Solving for quantity and prices gives

$$\begin{aligned} P_t &= \frac{\alpha_0 - \delta_0}{\delta_1 - \alpha_1} + \frac{\epsilon_t^s - \epsilon_t^d}{\delta_1 - \alpha_1} \\ Q_t &= \left( \delta_0 + \delta_1 \frac{\alpha_0 - \delta_0}{\delta_1 - \alpha_1} \right) + \frac{\delta_1 \epsilon_t^s - \alpha_1 \epsilon_t^d}{\delta_1 - \alpha_1}. \end{aligned} \quad (10.9)$$

which implies

$$\text{var}[P_t] = \frac{\sigma_s^2 + \sigma_d^2}{(\delta_1 - \alpha_1)^2} \quad \text{and} \quad \text{cov}[P_t, Q_t] = \frac{\delta_1 \sigma_s^2 + \alpha_1 \sigma_d^2}{(\delta_1 - \alpha_1)^2}. \quad (10.10)$$

The OLS estimator of  $\beta_1$  in the regression  $Q_t = \beta_0 + \beta_1 P_t + \epsilon_t$  will converge to the ratio of  $\text{cov}[P_t, Q_t]$  and  $\text{var}[P_t]$ :

$$\hat{\beta}_1^{ols} \rightarrow_p \frac{\text{cov}[Q_t, P_t]}{\text{var}[P_t]} = \frac{\delta_1 \sigma_s^2 + \alpha_1 \sigma_d^2}{\sigma_s^2 + \sigma_d^2}$$

which is neither the price elasticity of demand nor the price elasticity of supply.

Suppose, however, that there is some observable variable  $r_t$  that shifts the supply function but not the demand function. That is, suppose the market for our good is represented by the equations

$$\begin{aligned} Q_t^d &= \delta_0 + \delta_1 P_t + \epsilon_t^d && \text{(Demand Eq } \delta_1 < 0) \\ Q_t^s &= \alpha_0 + \alpha_1 P_t + \alpha_2 r_t + \epsilon_t^s && \text{(Supply Eq } \alpha_1 > 0) \\ Q_t^s &= Q_t^d && \text{(Market Clearing)} \end{aligned} \quad (10.11)$$

where  $\alpha_2 \neq 0$  and  $r_t$  is uncorrelated with the demand shocks. Because (by assumption)  $r_t$  is uncorrelated with the demand shock, and because  $r_t$  is correlated with prices (changes in  $r_t$  shift the supply function, and this changes price), it is a valid instrument. The IV estimators for  $\delta_0$  and  $\delta_1$  are then

$$\begin{aligned} \hat{\delta}_0^{iv} &= \bar{Q} - \hat{\delta}_1^{iv} \bar{P} \\ \hat{\delta}_1^{iv} &= \frac{\sum_{i=1}^T (r_t - \bar{r})(Q_t - \bar{Q})}{\sum_{i=1}^T (r_t - \bar{r})(P_t - \bar{P})}. \end{aligned} \quad (10.12)$$

As argued previously, these are consistent (though biased) estimators.

### 10.2.1 A Two-Stage Least Squares Perspective

We continue with the demand-supply example in (10.11), and view the estimator  $\hat{\delta}_1^{iv}$  from the perspective of the following two-step procedure:

- First regress the endogenous regressor  $P_t$  onto the exogenous regressor  $r_t$ :

$$P_t = \phi_0 + \phi_1 r_t + u_t$$

and collect the fitted values

$$\hat{P}_t = \hat{\phi}_0 + \hat{\phi}_1 r_t,$$

where

$$\hat{\phi}_1 = \frac{\sum_{i=1}^T (P_t - \bar{P})(r_t - \bar{r})}{\sum_{i=1}^T (r_t - \bar{r})^2}. \quad (10.13)$$

- Next regress  $Q_t$  on  $\hat{P}_t$  (with intercept)

Call the estimator of the coefficient on  $\hat{P}_t$  in this regression  $\hat{\delta}_1^{2sls}$  where “2sls” stands for “2-Stage Least Squares”. We have

$$\hat{\delta}_1^{2sls} = \frac{\sum_{i=1}^T (Q_t - \bar{Q})(\hat{P}_t - \bar{\hat{P}})}{\sum_{i=1}^T (\hat{P}_t - \bar{\hat{P}})^2}. \quad (10.14)$$

This estimator turns out to be identical to  $\hat{\delta}_1^{iv}$ . Since  $\hat{P}_t = \hat{\phi}_0 + \hat{\phi}_1 r_t$ , we have

$$\hat{P}_t - \bar{\hat{P}} = \hat{\phi}_1 (r_t - \bar{r}), \quad (10.15)$$

therefore

$$\hat{\delta}_1^{2sls} = \frac{\hat{\phi}_1 \sum_{i=1}^T (Q_t - \bar{Q})(r_t - \bar{r})}{\sum_{i=1}^T (\hat{\phi}_1 (r_t - \bar{r}))^2}.$$

Summing (10.15) over all observations gives

$$\sum_{i=1}^T (\hat{P}_t - \bar{\hat{P}})^2 = \hat{\phi}_1^2 \sum_{t=1}^T (r_t - \bar{r})^2 = \hat{\phi}_1 \sum_{i=1}^T (P_t - \bar{P})(r_t - \bar{r})$$

where we used the expression (10.13) for  $\hat{\phi}_1$ . It follows that

$$\hat{\delta}_1^{2sls} = \frac{\sum_{i=1}^T (Q_t - \bar{Q})(r_t - \bar{r})}{\sum_{i=1}^T (P_t - \bar{P})(r_t - \bar{r})}$$

which is the same expression as  $\hat{\delta}_1^{iv}$ .

The intuition is as follows. Imagine that we can “shut down” the supply and demand shocks. Then the demand function does not shift, whereas the supply function shifts as  $r_t$  varies. The intersection points then map out the demand function, as illustrated below:

```
r <- rnorm(nsim, 2, 3)
a2 <- 4
plt2 <- ggplot()
for (i in 1:nsim){
  qs <- a0 + a1*p + a2*r[i]
  qd <- b0 + b1*p
```

```

peq <- (a0-b0)/(b1-a1) + a2*r[i]/(b1-a1)
qeq <- (b0+b1*(a0-b0)/(b1-a1)) + b1*a2*r[i]/(b1-a1)
dat <- data.frame("qs"=qs, "qd"=qd, "p"=p) %>% arrange(p)
plt2 <- plt2 +
  geom_line(data=dat, aes(y=qs,x=p), color="gray", alpha=0.4) +
  geom_line(data=dat, aes(y=qd,x=p), color="gray", alpha=0.4) +
  annotate("point", y=qeq, x=peq, size=2) +
  theme_minimal()
}
plt2

```

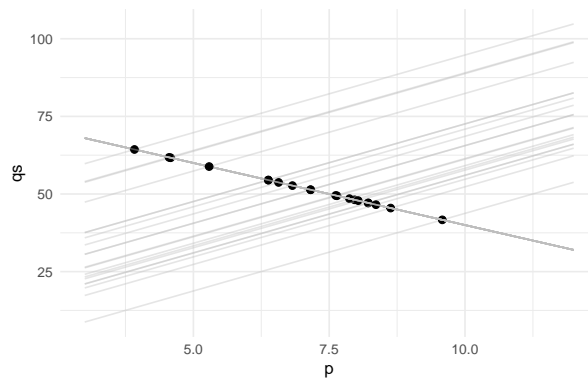


Figure 10.2: Demand and Supply with Variation in Exogenous Variable Only

In other words, movements in  $r_t$  can help to ‘trace out’ the demand function. The problem, however, is that the “real data” also contains movements due to the demand and supply shocks. We get around this problem by isolating movements in  $P_t$  that are due to solely to movements in  $r_t$ . This is done in the first stage regression where we regress  $P_t$  on  $r_t$ . Since the fitted values  $\hat{P}_t$  is equal to  $\hat{\phi}_0 + \hat{\phi}_1 r_t$ , and  $r_t$  is exogenous, so is  $\hat{P}_t$ . We get a consistent estimator of  $\delta_1$  by regressing  $Q_t$  on  $\hat{P}_t$ , i.e., by employing only that part of  $P_t$  that is uncorrelated with the demand and supply shocks.

Although IV estimation gives consistent estimators despite the presence of endogenous regressors, it does so at a cost. Since  $\hat{\delta}_1^{iv}$  is, in essence, obtained from a regression of  $Q_t$  on  $\hat{P}_t$ , and because there is less variation in  $\hat{P}_t$  than in  $P_t$  (why?), there is a reduction in effective variation in the regressor. This results in an increase in the variance of the estimator as compared to if we had regressed  $Q_t$  on  $P_t$ . If  $P_t$  is endogenous, then this would seem to be a good tradeoff, since the alternative is an inconsistent estimator. However, if the correlation between  $P_t$  and  $r_t$  is weak, the reduction in effective variation will be substantial, and this will result in very imprecise estimators. This loss of precision has to be weighed against the perceived degree of endogeneity in the regressor.

IV estimators can behave very poorly if instruments are not valid (in the sense of being poorly correlated with the endogenous regressors) and if there is some degree of endogeneity. We can get some intuition for why IV estimators are likely to perform poorly in these circumstances by referring to Eq. 10.4.

### 10.3 Multiple Instruments

We discuss situations where we have multiple endogenous regressor and multiple instruments. We revert to generic notation for our regressions, and continue to assume a sample of independent draws of size  $N$  throughout. We first note that having exogenous and endogenous regressors in the regression at the same time does not present any particular difficulty.

Suppose the regression equation of interest is:

$$Y_i = \beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} + \epsilon_i$$

where  $X_{1,i}$  is an exogenous regressor (not correlated with the noise term) and  $X_{2,i}$  is an endogenous regressor (correlated with the noise term). Suppose there is a variable  $Z_{1,i}$  that is correlated with  $X_{2,i}$  and uncorrelated with the noise term. Then we have the moment conditions

$$\begin{aligned} E[\epsilon_i] &= E[Y_i - \beta_0 - \beta_1 X_{1,i} - \beta_2 X_{2,i}] = 0 \\ E[\epsilon_i X_{1,i}] &= E[(Y_i - \beta_0 - \beta_1 X_{1,i} - \beta_2 X_{2,i}) X_{1,i}] = 0 \\ E[\epsilon_i Z_{1,i}] &= E[(Y_i - \beta_0 - \beta_1 X_{1,i} - \beta_2 X_{2,i}) Z_{1,i}] = 0. \end{aligned} \tag{10.16}$$

The sample analogue of (10.16) is

$$\begin{aligned} \sum_{i=1}^N (Y_i - \hat{\beta}_0^{iv} - \hat{\beta}_1^{iv} X_{1,i} - \hat{\beta}_2^{iv} X_{2,i}) &= 0 \\ \sum_{i=1}^N (Y_i - \hat{\beta}_0^{iv} - \hat{\beta}_1^{iv} X_{1,i} - \hat{\beta}_2^{iv} X_{2,i}) X_{1,i} &= 0 \\ \sum_{i=1}^N (Y_i - \hat{\beta}_0^{iv} - \hat{\beta}_1^{iv} X_{1,i} - \hat{\beta}_2^{iv} X_{2,i}) Z_{1,i} &= 0. \end{aligned} \tag{10.17}$$

The IV estimators are those values  $\hat{\beta}_0^{iv}$ ,  $\hat{\beta}_1^{iv}$  and  $\hat{\beta}_2^{iv}$  that solve (10.17). Of course, not all three-equation systems in three unknowns can be solved. The condition that  $Z_{1,i}$  be correlated with the endogenous regressor is required for the system to be solvable.

What if we have more than one instrument? Suppose we have  $Z_{1,i}$  and  $Z_{2,i}$  that are not correlated with the regression noise term, but correlated with the endogenous regressor? (Imagine that there are two exogenous variables that shift the only supply function in our demand-supply example.) In this case we have more moment equations than parameters:

$$\begin{aligned} E[\epsilon_i] &= E[Y_i - \beta_0 - \beta_1 X_{1,i} - \beta_2 X_{2,i}] = 0 \\ E[\epsilon_i X_{1,i}] &= E[(Y_i - \beta_0 - \beta_1 X_{1,i} - \beta_2 X_{2,i}) X_{1,i}] = 0 \\ E[\epsilon_i Z_{2,i}] &= E[(Y_i - \beta_0 - \beta_1 X_{1,i} - \beta_2 X_{2,i}) Z_{1,i}] = 0 \\ E[\epsilon_i Z_{3,i}] &= E[(Y_i - \beta_0 - \beta_1 X_{1,i} - \beta_2 X_{2,i}) Z_{2,i}] = 0. \end{aligned} \tag{10.18}$$

The sample analogues of the moment conditions are:

$$\begin{aligned}
\sum_{i=1}^N (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_{1,i} - \hat{\beta}_2 X_{2,i}) &= 0 \\
\sum_{i=1}^N (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_{1,i} - \hat{\beta}_2 X_{2,i}) X_{1,i} &= 0 \\
\sum_{i=1}^N (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_{1,i} - \hat{\beta}_2 X_{2,i}) Z_{1,i} &= 0 \\
\sum_{i=1}^N (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_{1,i} - \hat{\beta}_2 X_{2,i}) Z_{2,i} &= 0.
\end{aligned} \tag{10.19}$$

We cannot solve four moment conditions for three parameters, unless the moment conditions are dependent. However, we can set  $\hat{\beta}_0$ ,  $\hat{\beta}_1$  and  $\hat{\beta}_2$  so that the values on the left hand side of the equations in (10.19) are as close to zero as possible, in some sense. For instance, we can set  $\hat{\beta}_0$ ,  $\hat{\beta}_1$  and  $\hat{\beta}_2$  so that the sum of the square of the four values on the LHS of (10.19) are minimized. We will label these estimators “MM” (for “Method of Moments”).

At this point, it is easier to switch to matrix algebra. Write

$$y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{bmatrix}, \quad X = \begin{bmatrix} 1 & X_{1,1} & X_{2,1} \\ 1 & X_{1,2} & X_{2,2} \\ \vdots & \vdots & \vdots \\ 1 & X_{1,N} & X_{2,N} \end{bmatrix}, \quad \varepsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & X_{1,1} & Z_{1,1} & Z_{2,1} \\ 1 & X_{1,2} & Z_{1,2} & Z_{2,2} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & X_{1,N} & Z_{1,N} & Z_{2,N} \end{bmatrix}.$$

Let  $x_i^T = [1 \quad X_{1,i} \quad X_{2,i}]$  and  $z_i^T = [1 \quad X_{1,i} \quad Z_{1,i} \quad Z_{2,i}]$ . The regression equation is then

$$y = X\beta + \varepsilon$$

or

$$Y_i = x_i^T \beta + \epsilon_i, \quad i = 1, 2, \dots, N,$$

and where

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}.$$

The population moment conditions in (10.18) can be written as

$$E[z_i(Y_i - \beta_0 - x_i^T \beta_1)] = E[z_i \epsilon_i] = 0.$$

The sample analogue (10.19) can be written as

$$Z^T(y - X\hat{\beta}^{mm}) = Z^T y - Z^T X\hat{\beta}^{mm} = 0.$$

The “Sum of Squared Moments” is then

$$(Z^T y - Z^T X\hat{\beta}^{mm})^T (Z^T y - Z^T X\hat{\beta}^{mm}). \tag{10.20}$$

Minimizing (10.20), we get

$$\hat{\beta}^{mm} = (X^T Z Z^T X)^{-1} X^T Z Z^T y \quad (10.21)$$

(see exercises) where we assume that  $X^T Z Z^T X$  is invertible. We can also write  $\hat{\beta}^{mm}$  as

$$\begin{aligned} \hat{\beta}^{mm} &= (X^T Z Z^T X)^{-1} X^T Z Z^T (X\beta + \varepsilon) \\ &= \beta + (X^T Z Z^T X)^{-1} X^T Z Z^T \varepsilon \\ &= \beta + \left[ \left( \frac{1}{N} X^T Z \right) \left( \frac{1}{N} Z^T X \right) \right]^{-1} \left( \frac{1}{N} X^T Z \right) \left( \frac{1}{N} Z^T \varepsilon \right). \end{aligned}$$

Note that

$$\frac{1}{N} X^T Z = \frac{1}{N} \sum_{i=1}^N x_i z_i^T, \quad \frac{1}{N} Z^T \varepsilon = \frac{1}{N} \sum_{i=1}^N z_i \varepsilon_i, \text{ etc.}$$

Roughly speaking,  $\hat{\beta}^{mm}$  will be consistent if the sample covariances in  $\frac{1}{N} \sum_{i=1}^N z_i \varepsilon_i$  converge in probability to their population counterparts, which are zero if the instruments (the variables in  $Z$ ) are exogenous. As with the IV estimator, the MM estimator described here is biased. Taking conditional expectations (given  $X$  and  $Z$ ) of  $\hat{\beta}^{mm}$  does not remove the term  $(X^T Z Z^T X)^{-1} X^T Z Z^T \varepsilon$ , as we do not have  $E[\varepsilon|X, Z] = 0$ .

If there are as many variables in  $Z$  as there are in  $X$  (i.e., if we have as many instruments as endogenous regressors), then  $X^T Z$  is square, and assuming that  $X^T Z$  is invertible, (10.21) reduces to

$$\hat{\beta}^{iv} = (Z^T X)^{-1} Z^T y \quad (10.22)$$

which we refer to as the IV estimator, cf. (10.3).

What if take a two-stage least squares approach? (And how would we do it?) The first stage regression would be:

1. Regress each endogenous regressor on all of the exogenous variables (exogenous regressors and instruments). In the context of our specific example (one exogenous regressor  $X_{1,i}$ , one endogenous regressor  $X_{2,i}$ , and two instruments  $Z_{1,i}$  and  $Z_{2,i}$ ), this would mean regressing  $X_{2,i}$  on a constant,  $X_{1,i}$ ,  $Z_{1,i}$  and  $Z_{2,i}$ , and computing the fitted values

$$\hat{X}_{2,i} = \hat{\phi}_0 + \hat{\phi}_1 X_{1,i} + \hat{\phi}_2 Z_{1,i} + \hat{\phi}_3 Z_{2,i}.$$

2. In the second step, replace the endogenous regressors with the *fitted* endogenous regressors

$$Y_i = \beta_0 + \beta_1 X_{1,i} + \beta_2 \hat{X}_{2,i} + u_i$$

and estimate by OLS to get the 2SLS estimators  $\hat{\beta}_0^{2sls}$ ,  $\hat{\beta}_1^{2sls}$  and  $\hat{\beta}_2^{2sls}$ .

We can write the two steps above using matrix algebra as follows. In the first step, we regress  $X$  on  $Z$  to get the estimator  $\hat{B} = (Z^T Z)^{-1} Z^T X$ . The fitted values are then  $\hat{X} = Z \hat{B} = Z(Z^T Z)^{-1} Z^T X$ . These might seem like odd statements since  $X$  contains three columns: recall

that

$$X = \begin{bmatrix} 1 & X_{1,1} & X_{2,1} \\ 1 & X_{1,2} & X_{2,2} \\ \vdots & \vdots & \vdots \\ 1 & X_{1,N} & X_{2,N} \end{bmatrix} \quad \text{and} \quad Z = \begin{bmatrix} 1 & X_{1,1} & Z_{1,1} & Z_{2,1} \\ 1 & X_{1,2} & Z_{1,2} & Z_{2,2} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & X_{1,N} & Z_{1,N} & Z_{2,N} \end{bmatrix}.$$

Regressing  $X$  on  $Z$  means regressing each column of  $X$  on  $Z$ , and the columns of  $\hat{B}$  contain the estimators from each of these regressions. Likewise, the columns of the fitted *matrix*  $\hat{X}$  contain the fitted values from each of the regressions. Regressing a column of ones on  $Z$  simply returns the column of ones as the fitted values (think about it). Regressing the column of  $X_{1,i}$  observations on  $Z$  will likewise simply return the column of  $X_{1,i}$  observations. Regressing the column of  $X_{2,i}$  observations on  $Z$  is exactly the first stage regression that we described earlier.

In the second step, we regress  $y$  on  $\hat{X}$  to get the 2SLS estimator

$$\begin{aligned} \hat{\beta}^{2sls} &= (\hat{X}^T \hat{X})^{-1} \hat{X}^T y \\ &= (X^T Z (Z^T Z)^{-1} Z^T Z (Z^T Z)^{-1} Z^T X)^{-1} X^T Z (Z^T Z)^{-1} Z^T y \\ &= (X^T Z (Z^T Z)^{-1} Z^T X)^{-1} X^T Z (Z^T Z)^{-1} Z^T y. \end{aligned} \quad (10.23)$$

The 2SLS estimator (10.23) and what we have called the MM estimator are different, but both are consistent. It is easy to show that if  $X$  and  $Z$  have the same number of columns (so  $X^T Z$  is square) then they both reduce to what we have called the IV estimator.

The formulas derived extend without change to the case where there are  $K$  exogenous regressors (in addition to the constant),  $G$  endogenous regressors, and  $M$  instrumental variables, where  $M \geq G$ . That is, suppose we are interested in estimating the regression

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 X_{1,i}^k + \dots \beta_K X_{K,i}^k + \beta_{K+1} X_{K+1,i}^g + \dots + \beta_{K+G} X_{K+G,i}^g + \epsilon_i \\ &= x_i^T \beta + \epsilon_i \end{aligned}$$

where we use the  $k$  and  $g$  superscripts to denote the exogenous and endogenous regressors, and the  $K + G + 1$  vector  $x_i$  is

$$x_i^T = [1 \quad X_{1,i}^k \quad \dots \quad X_{K,i}^k \quad X_{K+1,i}^g \quad \dots \quad X_{K+G,i}^g].$$

Suppose you are able to find  $M \geq G$  instruments  $Z_{1,i}, Z_{2,i}, \dots, Z_{M,i}$ . Define the  $K + M + 1$  vector  $z_i$  as

$$z_i^T = [1 \quad X_{1,i}^k \quad \dots \quad X_{K,i}^k \quad Z_{1,i} \quad \dots \quad Z_{M,i}].$$

Let

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix} \quad \text{and} \quad Z = \begin{bmatrix} z_1^T \\ z_2^T \\ \vdots \\ z_N^T \end{bmatrix}.$$

Then the formulas (10.21), (10.22) and (10.23) for the “Method of Moments”, IV and 2SLS estimators all continue to apply. The inverses in those formulas must exist, of course. This requires that the instruments be correlated with the endogenous variables, and also that  $M \geq G$ .



If  $M = G$ , then the formulas all reduce to the IV estimator. If  $Z = X$ , then the IV estimator is simply the OLS estimator. If  $M = G$ , we say that the regression is **just identified**. If  $M > G$ , we say that the regression is **over-identified**.

### 10.4 Generalized Method of Moments

Finally we define the GMM estimator as those that minimize a *weighted* sum of squared moments, i.e.,

$$\hat{\beta}_W^{gmm} = \operatorname{argmin}_{\hat{\beta}} J(W_N)$$

where

$$\begin{aligned} J(W_N) &= (Z^T y - Z^T X \hat{\beta})^T W_N (Z^T y - Z^T X \hat{\beta}) \\ &= y^T Z Z^T y - 2 \hat{\beta}^T X^T Z W_N Z^T y + \hat{\beta}^T X^T Z W_N Z^T X \hat{\beta} \end{aligned} \quad (10.24)$$

and where  $W_N$  is a symmetric positive-definite matrix of “weights” (the weights may be data dependent). The resulting estimator depends on the choice of weights, which is the reason for the subscript  $W$  in  $\hat{\beta}_W^{gmm}$ . Minimizing (10.24), we get

$$\hat{\beta}_W^{gmm} = (X^T Z W_N Z^T X)^{-1} X^T Z W_N Z^T y. \quad (10.25)$$

To derive the properties of this estimator, we make the following assumptions:

**Assumption Set F:**

(F1) The equation to be estimated is

$$Y_i = x_i^T \beta + \epsilon_i, \quad i = 1, 2, \dots, N$$

where  $x_i$  is the vector

$$x_i^T = \begin{bmatrix} 1 & X_{1,i}^k & \dots & X_{K,i}^k & X_{K+1,i}^g & \dots & X_{K+G,i}^g \end{bmatrix}$$

and where  $X_{1,i}^k, \dots, X_{K,i}^k$  are  $K$  variables known to be exogenous, and  $X_{K+1,i}^g, \dots, X_{K+G,i}^g$  are  $G$  variables thought to be endogenous.

(F2) There are  $M$  instruments  $Z_{1,i}, \dots, Z_{M,i}$  such that the vector  $z_i$  defined as

$$z_i^T = \begin{bmatrix} 1 & X_{1,i}^k & \dots & X_{K,i}^k & Z_{1,i} & \dots & Z_{M,i} \end{bmatrix}$$

has the following properties:

(F3) the  $((K + M + 1) \times (K + G + 1))$  matrix  $E[z_i z_i^T] = \Sigma_{zx}$  has full column rank,

(F4)  $E[\epsilon_i | z_i] = 0$ ,

(F5)  $E[\epsilon_i^2 z_i z_i^T] = S$  is finite and non-singular.

Furthermore, assume

(F6) the unique random variables in  $\{x_i, z_i\}$  are i.i.d., and

(F7)  $W_N$  is chosen such that  $W_N \rightarrow_p W$  symmetric and positive-definite.

Let  $X$  and  $Z$  be as previously defined. The estimator (10.25) is consistent:

$$\begin{aligned}
\hat{\beta}_W^{gmm} &= (X^T Z W_N Z^T X)^{-1} X^T Z W_N Z^T y \\
&= \beta + (X^T Z W_N Z^T X)^{-1} X^T Z W_N Z^T \varepsilon \\
&= \beta + [(\frac{1}{N} X^T Z) W_N (\frac{1}{N} Z^T X)]^{-1} (\frac{1}{N} X^T Z) W_N (\frac{1}{N} Z^T \varepsilon) \\
&\rightarrow_p \beta + (\Sigma_{zx}^T W \Sigma_{zx})^{-1} \Sigma_{zx}^T W 0 \\
&= \beta
\end{aligned}$$

since under our assumptions

$$\frac{1}{N} Z^T X = \frac{1}{N} \sum_{i=1}^N z_i x_i^T \rightarrow_p \Sigma_{zx}$$

and

$$\frac{1}{N} Z^T \varepsilon = \frac{1}{N} \sum_{i=1}^N z_i \varepsilon_i \rightarrow_p 0.$$

Furthermore,

$$\begin{aligned}
\sqrt{N}(\hat{\beta}_W^{gmm} - \beta) &= [(\frac{1}{N} X^T Z) W_N (\frac{1}{N} Z^T X)]^{-1} (\frac{1}{N} X^T Z) W_N (\frac{1}{\sqrt{N}} Z^T \varepsilon) \\
&\rightarrow_d \text{Normal}(0, (\Sigma_{zx}^T W \Sigma_{zx})^{-1} \Sigma_{zx}^T W S W \Sigma_{zx} (\Sigma_{zx}^T W \Sigma_{zx})^{-1})
\end{aligned}$$

since under our assumptions,

$$\frac{1}{\sqrt{N}} Z^T \varepsilon \rightarrow_d \text{Normal}(0, S).$$

The asymptotic variance of the GMM estimator is

$$\text{avar}[\hat{\beta}_W^{gmm}] = (\Sigma_{zx}^T W \Sigma_{zx})^{-1} \Sigma_{zx}^T W S W \Sigma_{zx} (\Sigma_{zx}^T W \Sigma_{zx})^{-1}. \quad (10.26)$$

We approximate the finite sample variance of  $\hat{\beta}_W^{gmm}$  by

$$\text{var}[\hat{\beta}_W^{gmm}] \approx \frac{1}{N} (\Sigma_{zx}^T W \Sigma_{zx})^{-1} \Sigma_{zx}^T W S W \Sigma_{zx} (\Sigma_{zx}^T W \Sigma_{zx})^{-1}$$

To operationalize this estimator, we need to estimate the various elements in the formula for  $\text{var}[\hat{\beta}_W^{gmm}]$ . Assume for the moment that we know  $W$ . For  $\Sigma_{zx}$  we can use

$$\hat{\Sigma}_{zx} = \frac{1}{N} \sum_{i=1}^N z_i x_i^T = \frac{1}{N} Z^T X.$$

For  $S$  we can use

$$\hat{S} = \frac{1}{N} \sum_{i=1}^N \hat{\varepsilon}_i^2 z_i z_i^T.$$

In other words, we can estimate the variance of  $\hat{\beta}_W^{gmm}$  with

$$\begin{aligned}\widehat{var}[\hat{\beta}_W^{gmm}] &= \frac{1}{N}(\hat{\Sigma}_{zx}^T W \hat{\Sigma}_{zx})^{-1} \hat{\Sigma}_{zx}^T W \hat{S} W \hat{\Sigma}_{zx} (\hat{\Sigma}_{zx}^T W \hat{\Sigma}_{zx})^{-1} \\ &= (X^T Z W Z^T X)^{-1} X^T Z W \left[ \sum_{i=1}^N \hat{\epsilon}_i^2 z_i z_i^T \right] W Z^T X (X^T Z W Z^T X)^{-1}\end{aligned}\quad (10.27)$$

This variance estimator is heteroskedasticity-robust.

Remarks:

1. In the just-identified case, the GMM estimator reduces to the IV estimator regardless of  $W_N$ :

$$\begin{aligned}\hat{\beta}_W^{gmm} &= (X^T Z W_N Z^T X)^{-1} X^T Z W_N Z^T y \\ &= (Z^T X)^{-1} Z^T y \\ &= \hat{\beta}^{iv}.\end{aligned}\quad (10.28)$$

The asymptotic variance (10.26) becomes

$$\text{avar}[\hat{\beta}^{iv}] = \Sigma_{zx}^{-1} S (\Sigma_{zx}^T)^{-1}.$$

The variance estimator (10.27) becomes

$$\widehat{var}[\hat{\beta}^{iv}] = (Z^T X)^{-1} \left[ \sum_{i=1}^N \hat{\epsilon}_i^2 z_i z_i^T \right] (X^T Z)^{-1}.$$

2. If we choose  $W_N = I$ , the identity matrix, then

$$\begin{aligned}\hat{\beta}_I^{gmm} &= (X^T Z W_N Z^T X)^{-1} X^T Z W_N Z^T y \\ &= (X^T Z Z^T X)^{-1} X^T Z Z^T y \\ &= \hat{\beta}^{mm},\end{aligned}$$

the “MM” estimator presented earlier. The variance estimator (10.27) becomes

$$\widehat{var}[\hat{\beta}^{mm}] = (X^T Z Z^T X)^{-1} X^T Z \left[ \sum_{i=1}^N \hat{\epsilon}_i^2 z_i z_i^T \right] Z^T X (X^T Z Z^T X)^{-1}. \quad (10.29)$$

3. If we choose  $W_N = ((1/N)Z^T Z)^{-1}$ , then the GMM estimator becomes the 2SLS estimator:

$$\begin{aligned}\hat{\beta}_{(Z^T Z)^{-1}}^{gmm} &= (X^T Z W_N Z^T X)^{-1} X^T Z W_N Z^T y \\ &= (X^T Z (Z^T Z)^{-1} Z^T X)^{-1} X^T Z (Z^T Z)^{-1} Z^T y \\ &= \hat{\beta}^{2sls}.\end{aligned}$$

Since  $[(1/N)Z^T Z]^{-1} \rightarrow_p \Sigma_{zz}^{-1}$ , the asymptotic variance of the GMM estimator becomes

$$\text{avar}[\hat{\beta}^{2sls}] = (\Sigma_{zx}^T \Sigma_{zz}^{-1} \Sigma_{zx})^{-1} \Sigma_{zx}^T \Sigma_{zz}^{-1} S \Sigma_{zz}^{-1} \Sigma_{zx} (\Sigma_{zx}^T \Sigma_{zz}^{-1} \Sigma_{zx})^{-1}. \quad (10.30)$$

where  $\Sigma_{zz} = E[z_i z_i^T]$ .

Note that expression (10.30) allows for conditional heteroskedasticity, but if there is conditional homoskedasticity, then

$$S = E[\epsilon_i^2 z_i z_i^T] = E[E[\epsilon_i^2 z_i z_i^T | z_i]] = E[E[\epsilon_i^2 | z_i] z_i z_i^T] = \sigma^2 E[z_i z_i^T] = \sigma^2 \Sigma_{zz}$$

and the asymptotic variance reduces to

$$\text{avar}[\hat{\beta}^{2sls}] = \sigma^2 (\Sigma_{zx}^T \Sigma_{zz}^{-1} \Sigma_{zx})^{-1}. \quad (10.31)$$

We can estimate the estimator variance by

$$\widehat{\text{var}}[\hat{\beta}^{2sls}] = \widehat{\sigma^2} (X^T Z (Z^T Z)^{-1} Z^T X)^{-1}.$$

where  $\widehat{\sigma^2}$  is some consistent estimator for  $\sigma^2$ .

### 10.4.1 Optimal GMM

It turns out that the (asymptotically) optimal weight is to choose  $W_N$  such that  $W_N \rightarrow_p S^{-1}$ . This is because the asymptotic variance of  $\hat{\beta}^{gmm}$  then becomes

$$\begin{aligned} \text{avar}[\hat{\beta}^{gmm}] &= (\Sigma_{zx}^T W \Sigma_{zx})^{-1} \Sigma_{zx}^T W S W \Sigma_{zx} (\Sigma_{zx}^T W \Sigma_{zx})^{-1} \\ &= (\Sigma_{zx}^T S^{-1} \Sigma_{zx})^{-1} \end{aligned}$$

and it can be shown that

$$(\Sigma_{zx}^T W \Sigma_{zx})^{-1} \Sigma_{zx}^T W S W \Sigma_{zx} (\Sigma_{zx}^T W \Sigma_{zx})^{-1} - (\Sigma_{zx}^T S^{-1} \Sigma_{zx})^{-1} \quad (10.32)$$

is positive definite for any symmetric positive definite  $W$ . A natural weight matrix to choose is therefore

$$W_N = \hat{S}^{-1} = \left[ \frac{1}{N} \sum_{i=1}^N \hat{\epsilon}_i^2 z_i z_i^T \right]^{-1}.$$

The problem with this is that we need to have the residuals  $\hat{\epsilon}_i$  in order to compute  $\hat{S}^{-1}$ , but we need an estimator of  $\beta$  in order to compute the residuals. One solution is to take a two-step approach:

- Compute  $\hat{\beta}_W^{gmm}$  for some (non-optimal) weighting matrix  $W_N$ . A common choice is to use  $W_N = ((1/N) Z^T Z)^{-1}$ , which gives the (inefficient but consistent) 2SLS estimator  $\hat{\beta}^{2sls}$ . Then calculate  $\hat{S}^{-1}$  using the residuals

$$\hat{\epsilon}_i = Y_i - x_i^T \hat{\beta}^{2sls}.$$

- Calculate the optimal GMM estimator as

$$\hat{\beta}^{gmm} = (X^T Z \hat{S}^{-1} Z^T X)^{-1} X^T Z \hat{S}^{-1} Z^T y.$$

The asymptotic variance of the Optimal GMM estimator is

$$\text{avar}[\hat{\beta}^{gmm}] = (\Sigma_{zx}^T S^{-1} \Sigma_{zx})^{-1}. \quad (10.33)$$

The variance estimator becomes

$$\begin{aligned} \widehat{\text{var}}[\hat{\beta}^{gmm}] &= \frac{1}{N} \left\{ \left( \frac{1}{N} \sum_{i=1}^N z_i x_i^T \right)^T \left[ \frac{1}{N} \sum_{i=1}^N \hat{\epsilon}_i^2 z_i z_i^T \right]^{-1} \left( \frac{1}{N} \sum_{i=1}^N z_i x_i^T \right) \right\}^{-1} \\ &= \left\{ X^T Z \left[ \sum_{i=1}^N \hat{\epsilon}_i^2 z_i z_i^T \right]^{-1} Z^T X \right\}^{-1}. \end{aligned} \quad (10.34)$$

Notice that the asymptotic variance of the optimal GMM estimator *under conditional homoskedasticity* becomes

$$\text{avar}[\hat{\beta}^{gmm}] = \sigma^2 (\Sigma_{zx}^T \Sigma_{zz}^{-1} \Sigma_{zx})^{-1}$$

which is the same as the asymptotic variance of the 2SLS estimator under conditional homoskedasticity. This says that 2SLS is efficient under conditional homoskedasticity. The 2SLS estimator and the two-step implementation of optimal GMM are not numerically identical, but they are both efficient.

**Example 10.1.** We estimate the equation

$$\log(\text{earnings}_i) = \beta_0 + \beta_1 s_i + \beta_2 \text{wexp}_i + \epsilon_i$$

using data in **earnings.xlsx**. The variable  $s_i$  is years of schooling and  $\text{wexp}_i$  is work experience. It is thought that years of schooling may be endogenous because of omitted unobserved factors (e.g., ability). We will use parents' years of schooling  $sm_i$  and  $sf_i$  as instruments. We use only the "females observations" in our data set. First we show the OLS results (with robust standard errors).

```
df <- read_excel("data\\earnings.xlsx")
df_f <- df %>% filter(male==0)
mdl_ols <- lm(log(earnings)~s+wexp, data=df_f)
lmtest::coeftest(mdl_ols, vcov.=sandwich::vcovHC(mdl_ols, type='HC2'))
```

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.7144138	0.2024091	3.5296	0.00049 ***
s	0.1091864	0.0133491	8.1793	1.168e-14 ***
wexp	0.0264365	0.0052837	5.0034	1.023e-06 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The following code contain functions for computing 2SLS and GMM estimators:

```
# Function to computes the general GMM estimator (with user defined weight W)
gmm <- function(y,X,Z,W){
  N <- length(y)
  ZX <- t(Z)%*%X
  Zy <- t(Z)%*%y
  invXZWZX <- solve(t(ZX)%*%W%*ZX)
  # Calculate Estimator
  b_gmm <- invXZWZX%*%t(ZX)%*%W%*%Zy
  # Calculate Estimator Variance
  ehat <- y - X%*%b_gmm
  s2 <- (1/N)*sum(ehat^2)
  hatS <- 0
  for (i in 1:N){
    zi <- as.matrix(Z[i,])
    hatS <- hatS + ehat[i]^2 * zi%*%t(zi)
  }
  b_gmm_var <- invXZWZX%*%t(ZX)%*%W%*%hatS%*%W%*%ZX%*%invXZWZX
  b_gmm_se <- sqrt(diag(b_gmm_var))
  result <- list("bhat"=t(b_gmm), "bhatse"=b_gmm_se, "bhatvar"=b_gmm_var)
}

# The following function computes the Optimal GMM estimator (two-step approach)
gmm_opt <- function(y,X,Z,initW){
  N <- length(y)
  ZX <- t(Z)%*%X
  Zy <- t(Z)%*%y
  # Get an estimator for hatS
  invXZWZX <- solve(t(ZX)%*%initW%*ZX)
  b_tsls <- invXZWZX%*%t(ZX)%*%initW%*%Zy
  ehat <- y - X%*%b_tsls
  hatS <- 0
  for (i in 1:N){
    zi <- as.matrix(Z[i,])
    hatS <- hatS + ehat[i]^2 * zi%*%t(zi)
  }
  invhatS <- solve(hatS)
  # Calculate Optimum GMM
  b_gmm_opt <- solve(t(ZX)%*%invhatS%*ZX)%*%t(ZX)%*%invhatS%*%Zy
  # Update hatS and calculate GMM variance
  ehatgmm <- y - X%*%b_gmm_opt
  hatSgmm <- 0
  for (i in 1:N){
    zi <- as.matrix(Z[i,])
    hatSgmm <- hatSgmm + ehatgmm[i]^2 * zi%*%t(zi)
  }
  invhatSgmm <- solve(hatSgmm)
  b_gmm_var <- solve(t(ZX)%*%invhatSgmm%*ZX)
  b_gmm_se <- sqrt(diag(b_gmm_var))
  result <- list("bhat"=t(b_gmm_opt), "bhatse"=b_gmm_se,
                "bhatvar"=b_gmm_var, "hatS"= hatS)
}
```

The 2SLS estimate of our regression equation is:

```
## Main body of program
y <- log(df_f$earnings); yname <- "log(earnings)"
N <- length(y)
X <- as.matrix(data.frame("cons"=rep(1,N), "s"=df_f$s, "wexp"=df_f$wexp))
Z <- as.matrix(data.frame("cons"=rep(1,N), "wexp"=df_f$wexp, "sf"=df_f$sf, "sm"=df_f$sm))
W <- solve(t(Z)%*%Z)

#TSLS
mdl_2sls <- gmm(y,X,Z,W)
rslts.tsls <- rbind(mdl_2sls$bhat,mdl_2sls$bhatse)
rownames(rslts.tsls)<-c("Est.", "S.E.")
cat("Method: TSLS\nDep Var:", yname, "\nInstruments:", colnames(Z), "\n")
rslts.tsls
```

```
Method: TSLS
Dep Var: log(earnings)
Instruments: cons wexp sf sm
              cons      s      wexp
Est. 0.2737045 0.1403965 0.027413246
S.E. 0.4329824 0.0306525 0.005349077
```

The GMM estimate of our regression equation is:

```
# GMM
mdl_gmm <- gmm_opt(y,X,Z,W)
rslts.gmm <- rbind(mdl_gmm$bhat,mdl_gmm$bhatse)
rownames(rslts.gmm)<-c("Est.", "S.E.")
cat("Method: GMM\nDep Var:", yname, "\nInstruments:", colnames(Z), "\n")
rslts.gmm
```

```
Method: GMM
Dep Var: log(earnings)
Instruments: cons wexp sf sm
              cons      s      wexp
Est. 0.2640190 0.14124994 0.027482277
S.E. 0.4336357 0.03068906 0.005353824
```

The 2SLS and GMM estimates are very similar. Both 2SLS and GMM estimates of the coefficient on  $s_i$  are larger than the corresponding OLS estimate.

### 10.4.2 Hypothesis Testing after GMM

#### Testing Linear Restrictions

We can do the usual t- and F-tests after GMM estimation. Since the F-statistic is the square of the t-statistic when testing single hypotheses, we focus on the F-statistic. Furthermore, we will use the asymptotic version (the chi-sq version), usually called the “Wald Test”. To (jointly) test the  $J$  hypotheses  $H_0 : R\beta = r$  where  $R$  is a  $J \times (K + 1)$  matrix and  $r$  is  $(K + 1) \times 1$ , the

statistic is

$$W = (R\hat{\beta}^{gmm} - r)^T (R \hat{var}[\hat{\beta}^{gmm}] R^T)^{-1} (R\hat{\beta}^{gmm} - r) \sim_a \chi^2_{(J)}$$

We continue with Example 10.1. There we estimated the regression

$$\log(earnings_i) = \beta_0 + \beta_1 s_i + \beta_2 wexp_i + \epsilon_i,$$

using GMM with  $wexp_i$ ,  $sm_i$  and  $sf_i$  as instruments. We test  $H_0 : \beta_1 = \beta_2 = 0$  versus the alternative that one or both of these restrictions do not hold.

```
R = matrix(c(0,1,0,0,0,1), nrow=2, byrow=TRUE)
r = matrix(c(0,0), ncol=1)
b = matrix(mdl_gmm$bhat, ncol=1)
V = as.matrix(mdl_gmm$bhatvar)
F_stat = t(R%*%b-r)%*%solve(R%*%V%*%t(R))%*%(R%*%b-r)
cat("F:",F_stat,", p-value:", 1-pchisq(F_stat,nrow(R)))
```

F: 41.75286 , p-value: 8.579873e-10

Now we test  $H_0 : \beta_1 = 0.1$

```
R = matrix(c(0,1,0), nrow=1, byrow=TRUE)
r = matrix(c(0.1), ncol=1)
b = matrix(mdl_gmm$bhat, ncol=1)
V = as.matrix(mdl_gmm$bhatvar)
F_stat = t(R%*%b-r)%*%solve(R%*%V%*%t(R))%*%(R%*%b-r)
cat("F:",F_stat,", p-value:", 1-pchisq(F_stat,nrow(R)))
```

F: 1.806672 , p-value: 0.1789079

### Testing for Weak Instruments

We have mentioned that weak instruments (those that are poorly correlated with the endogenous regressors) will result in estimators with poor finite sample properties (high variance, possibly large finite sample biases). To check for weak instruments, we run the “first stage regression” (as though doing 2SLS manually), i.e., we regress the endogenous regressor on all of the exogenous variables (both those included in regression, as well as all the instruments), and test for significance of the instruments in the first stage regression. Research has shown that the F-statistic should be large (on the order of 20 or so) for GMM estimators to have good finite sample properties.

**Example 10.2.** The “First Stage Regression” in Example 10.1 is

$$s_i = \delta_0 + \delta_1 wexp_i + \delta_2 sm_i + \delta_3 sf_i$$

and the hypothesis of invalid instrument is  $H_0 : \delta_2 = \delta_3 = 0$ .



```
## First Stage
mdl_firststage <- lm(s~wexp+sm+sf,data=df_f)
car::linearHypothesis(mdl_firststage, c('sm=0','sf=0'),
                      vcov=sandwich::vcovHC(mdl_firststage,type="HC1"))
```

Linear hypothesis test

Hypothesis:

sm = 0

sf = 0

Model 1: restricted model

Model 2: s ~ wexp + sm + sf

Note: Coefficient covariance matrix supplied.

```
Res.Df Df      F    Pr(>F)
1      268
2      266  2 26.325 3.703e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

It appears that  $sm_i$  and  $sf_i$  are valid instruments.

### Tests of Overidentifying Restrictions

Recall the GMM objective function

$$J(W_N) = (Z^T y - Z^T X \hat{\beta})^T W_N (Z^T y - Z^T X \hat{\beta})$$

and the general GMM estimator

$$\hat{\beta}_W^{gmm} = (X^T Z W_N Z^T X)^{-1} X^T Z W_N Z^T y.$$

Recall also that the matrix  $Z^T X$  is square in the just-identified case, and if it is invertible, then  $\hat{\beta}_W^{gmm}$  reduces to  $\hat{\beta}_W^{gmm} = (Z^T X)^{-1} Z^T y$ . In that case, we have

$$J(W_N) = (Z^T y - Z^T X \hat{\beta})^T W_N (Z^T y - Z^T X \hat{\beta}) = 0$$

since

$$Z^T y - Z^T X \hat{\beta} = Z^T y - Z^T X (Z^T X)^{-1} Z^T y = 0.$$

In the over-identified case,  $J(W_N)$  will generally be greater than zero. However, if the moment conditions hold, then the sample moments should hold approximately, and  $J(W_N)$  will be close to zero. It can be shown that if  $W_N$  is chosen optimally, then under the null that all moment conditions hold, i.e., that all the presumed exogenous regressors and instruments in fact exogenous, we have

$$J \sim_a \chi_{(M-G)}^2$$

where  $M - G$  is the number of “overidentifying restriction”, i.e., the number of excess instruments. This is the “Test of Overidentifying Restrictions”, and significant a  $J$  statistic indicates that one or more of the moment conditions do not hold: perhaps one (or more) of the presumed included exogenous regressors is actually endogenous, or one of the instruments is not exogenous, or some combination of these situations.

**Example 10.3.** We continue with Example 10.1. We have an overidentified situation (with one excess instrument), so we can carry out the Test of Overidentified Restrictions.

```
# This function calculates the J statistic
gmm_overid <- function(y,X,Z,W,bhat){
  ZX <- t(Z)%*%X
  Zy <- t(Z)%*%y
  J <- t(Zy - ZX%*%bhat)%*%W%*(Zy - ZX%*%bhat)
  Jpval <- 1-pchisq(J,ncol(Z)-ncol(X))
  result <- list("J"=J, "J-pval"=Jpval)
}
Jtest <- gmm_overid(y,X,Z,solve(mdl_gmm$hatS), matrix(mdl_gmm$bhat, ncol=1))
cat("J:", Jtest$J, "    p-value:", Jtest$`J-pval`)
```

J: 0.4940986 p-value: 0.4821047

The J-statistic does not indicate any misspecification.

### Testing Endogeneity

If all of the endogenous variables can be treated as exogenous, then OLS is the preferred estimation method, since it is more efficient than GMM. If we have valid instruments, we can test if one or more (or all) of the endogenous regressors can be treated as exogenous. Suppose our regression  $Y_i = x_i^T \beta + \epsilon_i$  contains the following regressors

$$x_i^T = [1 \quad X_{1,i}^k \quad \dots \quad X_{K,i}^k \quad X_{K+1,i}^g \quad \dots \quad X_{K+G,i}^g]$$

and where  $X_{1,i}^k, \dots, X_{K,i}^k$  are  $K$  variables thought to be exogenous, and  $X_{K+1,i}^g, \dots, X_{K+G,i}^g$  are  $G$  variables thought to be endogenous. Suppose we have the  $M$  instruments  $Z_{1,i}, \dots, Z_{M,i}$ . The vector  $z_i$  is

$$z_i^T = [1 \quad X_{1,i}^k \quad \dots \quad X_{K,i}^k \quad Z_{1,i} \quad \dots \quad Z_{M,i}] .$$

Suppose that in fact all the variables in  $z_i$  are exogenous, so the population moment conditions  $E[z_i^T \epsilon_i] = 0$  hold. Now suppose we wish to ask if some of the presumed endogenous variables can be treated as exogenous. If, say, the variable  $X_{K+1,i}^g$  is in fact exogenous, then we can add it to the vector  $z_i$ , i.e., the vector  $z_i$  becomes

$$\tilde{z}_i^T = [1 \quad X_{1,i}^k \quad \dots \quad X_{K,i}^k \quad X_{K+1,i}^g \quad Z_{1,i} \quad \dots \quad Z_{M,i}]$$

and the population moment conditions become  $E[\tilde{z}_i^T \epsilon_i] = 0$ .

Consider estimating the regression equation once using  $z_i$  as instruments, and another time with  $\tilde{z}_i$  as instruments. Let  $J_z$  and  $J_{\tilde{z}}$  be the respective  $J$ -statistics. If in fact  $X_{K+1,i}^g$  is

exogenous, both  $J$ -statistics should be close in value, with  $J_{\tilde{z}}$  larger than  $J_z$  (since more moment conditions are involved when using  $\tilde{z}_i$ ). If  $X_{K+1,i}^g$  is not exogenous, then we can expect the difference between  $J_z$  and  $J_{\tilde{z}}$  to be significant. Under the null that  $X_{K+1,i}^g$  is exogenous, the difference-in- $J$  statistic (which we denote by “ $C$ ”) will be approximately  $\chi_{(Q)}^2$  where  $Q$  is the number of additional moments being tested (in our example,  $Q = 1$ ).

$$C = J_{\tilde{z}} - J_z \sim_a \chi_Q^2.$$

We refer to this test as a test for endogeneity. One minor complication of this is that in order to ensure that  $C > 0$ , the  $\hat{S}$  used in the computation of  $J_z$  has to be the submatrix of the  $\hat{S}$  used to compute  $J_{\tilde{z}}$ , with the selected rows/columns corresponding to the variables in  $z_i$ .

**Example 10.4.** We continue with Example 10.1 and test if  $s_i$  can be treated as exogenous.

```
# C-Statistic, checking if "s" is endogenous

#-- Z when "s" is exogenous
Zr <- as.matrix(data.frame("cons"=rep(1,N), "wexp"=df_f$wexp,
                           "sf"=df_f$sf, "sm"=df_f$sm, "s"=df_f$s))

Wr <- solve(t(Zr)%*%Zr)
mdla <- gmm_opt(y,X,Zr,Wr)
mdla_Jstat <- gmm_overid(y,X,Zr,solve(mdla$hatS),t(mdla$bhat))
subhatS <- mdla$hatS[1:4,1:4] # Get appropriate submatrix
mdlb <- gmm(y,X,Z,solve(subhatS))
mdlb_Jstat <- gmm_overid(y,X,Z,solve(subhatS),t(mdlb$bhat))

C_stat <- mdla_Jstat$J - mdlb_Jstat$J
cat("C:",C_stat,", p-value:", 1-pchisq(C_stat,ncol(Zr)-ncol(Z)))
```

C: 1.467645 , p-value: 0.2257176

We do not reject the null that  $s_i$  is exogenous, suggesting that perhaps we could have just used OLS in the first place. The GMM estimator, although less precise, is nonetheless consistent, and does increase the coefficient on  $s_i$  substantially, which is in agreement with our view that there is an omitted variable problem in this application. The regressor  $s_i$  is significant in both GMM and OLS regressions; the issue is in the magnitude of the coefficient on  $s_i$ .

### 10.4.3 GMM Estimation in Stata

Finally, we verify all of the computations above using STATA.

```
import excel "data\earnings.xlsx", sheet("earnings") firstrow
gen ln_earn = ln(earnings)
regress ln_earn wexp s if male==0, vce(hc2)
ivregress 2sls ln_earn wexp (s = sf sm) if male==0
ivregress gmm ln_earn wexp (s = sf sm) if male==0, wmatrix(robust)
estat firststage
estat overid
estat endog
```

```
. import excel "data\earnings.xlsx", sheet("earnings") first(14 vars, 540 obs)

. gen ln_earn = ln(earnings)

. regress ln_earn wexp s if male==0, vce(hc2)
```

		Robust				
ln_earn	Coefficient	std. err.	z	P> z	[95% conf. interval]	
s	.1412499	.0306891	4.60	0.000	.0811005	.2013994
wexp	.0274823	.0053538	5.13	0.000	.016989	.0379756
_cons	.2640191	.4336357	0.61	0.543	-.5858913	1.113929

Endogenous: s

Exogenous: wexp sf sm

. estat firststage

First-stage regression summary statistics

		Adjusted	Partial	Robust	
Variable	R-sq.	R-sq.	R-sq.	F(2,266)	Prob > F
s	0.1874	0.1782	0.1841	26.3249	0.0000

. estat overid

Test of overidentifying restriction:

Hansen's J chi2(1) = .494099 (p = 0.4821)

. estat endog

Test of endogeneity (orthogonality conditions)

H0: Variables are exogenous

GMM C statistic chi2(1) = 1.46765 (p = 0.2257)

## 10.5 Exercises

**Exercise 10.1.** Let

$$\begin{aligned}
 Y_i &= \delta_0 + \delta_1 X_i + \epsilon_i \\
 &= \delta_0 + \delta_1 (\hat{X}_i + \hat{v}_i) + \epsilon_i \\
 &= \delta_0 + \delta_1 \hat{X}_i + u_i
 \end{aligned}$$

where  $u_i = \delta_1 \hat{v}_i + \epsilon_i$ , and where  $\hat{X}_i$  are the OLS fitted values from a regression of  $X_i$  on the (valid) instrument  $Z_i$ , i.e.,

$$\hat{X}_i = \hat{\phi}_0 + \hat{\phi}_1 Z_i, \quad \hat{\phi}_1 = \frac{\sum_{i=1}^N (Z_i - \bar{Z})(X_i - \bar{X})}{\sum_{i=1}^N (Z_i - \bar{Z})^2}, \quad \hat{\phi}_0 = \bar{X} - \hat{\phi}_1 \bar{Z}.$$

Assume the data  $\{X_i, Y_i, Z_i\}_{i=1}^N$  are iid. Show that

$$\text{var}[\hat{\delta}_1 | \mathbf{x}, \mathbf{z}] = \frac{\sigma_u^2}{R_{xz}^2 \sum_{i=1}^N (X_i - \bar{X})^2}$$

where  $R_{xz}^2$  is the  $R^2$  from the regression of  $X_i$  on  $Z_i$ . Explain why an instrument that is poorly correlated with  $X_i$  results in an imprecise estimator.

**Exercise 10.2.** Show that the estimator (10.21) minimizes the sum of squared moments (10.20). Show that (10.21) reduces to the IV estimator (10.22) if  $X$  and  $Z$  have the same number of columns.

**Exercise 10.3.** Show that the 2SLS estimator (10.23) reduces to the IV estimator (10.22) if  $X$  and  $Z$  have the same number of columns.

**Exercise 10.4.** Show that the GMM estimator (10.25) minimizes the “weighted sum of squared moments” (10.24). Show that (10.25) reduces the IV estimator (10.22) if  $X$  and  $Z$  have the same number of columns, regardless of the choice of weighting matrix  $W_N$ .

**Exercise 10.5.** Show that (10.32) is positive definite by showing that it can be written as  $ASA^T$  where

$$A = (\Sigma_{zx}^T W \Sigma_{zx})^{-1} \Sigma_{zx}^T W - (\Sigma_{zx}^T S^{-1} \Sigma_{zx})^{-1} \Sigma_{zx}^T S^{-1}.$$

## Chapter 11

### Introduction to Time Series Regressions

Time series data are observations of variables made repeatedly over time, often at regular intervals (annually, quarterly, monthly, daily,...). The main issue with regressions with time series data is that the observations generally cannot be thought of as independent draws from a population. As you will see as this chapter progresses, the presence of *intertemporal correlations*, trends and other features can lead to a wide range of problems in regression analysis, if not properly addressed.

We use the following packages in this chapter, plus a few others.

```
library(fpp3);
library(patchwork);
library(ggfortify);
```

There are a number of time-series object types in R. We will primarily use the `tsibble` object type provided by the `fpp3` package. The `fpp3` package includes a number of the packages we have been using, plus a few additional ones which help in the analysis of time series. See Hyndman and Athanasopoulos (2021), Hyndman and Athanasopoulos (2023). The `tsibble` is a time-series version of the `tibble` object type. We will occasionally use the `ts` object type provided in base R. The following example creates a vector of random numbers and converts it into a `ts` object, then converts it into a `tsibble` object.

```
set.seed(13)
x <- rnorm(12,0,1)
x.ts <- ts(x, start=c(1990,1),freq=4) #Define as Quarterly data starting at 1990Q1
x.ts
```

	Qtr1	Qtr2	Qtr3	Qtr4
1990	0.5543269	-0.2802719	1.7751634	0.1873201
1991	1.1425261	0.4155261	1.2295066	0.2366797
1992	-0.3653828	1.1051443	-1.0935940	0.4618709

```
x.tsbl <- as_tsibble(x.ts)
x.tsbl
```

```
# A tsibble: 12 x 2 [1Q]
  index value
  <qtr> <dbl>
1 1990 Q1  0.554
2 1990 Q2 -0.280
3 1990 Q3  1.78
4 1990 Q4  0.187
5 1991 Q1  1.14
6 1991 Q2  0.416
7 1991 Q3  1.23
```

```

8 1991 Q4  0.237
9 1992 Q1 -0.365
10 1992 Q2  1.11
11 1992 Q3 -1.09
12 1992 Q4  0.462

```

Data frames can be converted directly to `tsibbles`. We do this in the upcoming example, where the `Period` column in the data frame is first converted to a `yearmonth` object, and set as the index of the `tsibble`.

### 11.1 Overview

Many economic time series display cyclical behavior, such as in panel (a) of Fig. 11.1, which shows the time series plot of the quarterly series  $Y$  from the dataset `ts_02.xlsx`. Cyclical behavior implies “serial correlation” (a.k.a. “autocorrelation”) meaning that  $\text{cov}[Y_t, Y_{t-k}] \neq 0$  for some  $k \neq 0$ . The scatterplot in panel (b) of  $Y_t$  against  $Y_{t-1}$  shows that consecutive observations are very highly correlated. By definition, i.i.d. observations cannot display serial correlation.

```

ts02 <- readxl::read_excel("data\\ts_02.xlsx") %>%
  mutate(Period=yearquarter(Period), Y_1=l原因ag(Y,1)) %>%
  as_tsibble(index=Period)
p1 <- ts02 %>% autoplot(Y) + theme_minimal() + xlab("")
p2 <- ts02 %>% ggplot(aes(x=Y_1, y=Y), size=0.5) + geom_point() + theme_minimal() +
  ylab("Y(t)") + xlab("Y(t-1)")
(p1 | p2) + plot_layout(widths=c(1.5,1)) + plot_annotation(tag_levels="a")

```

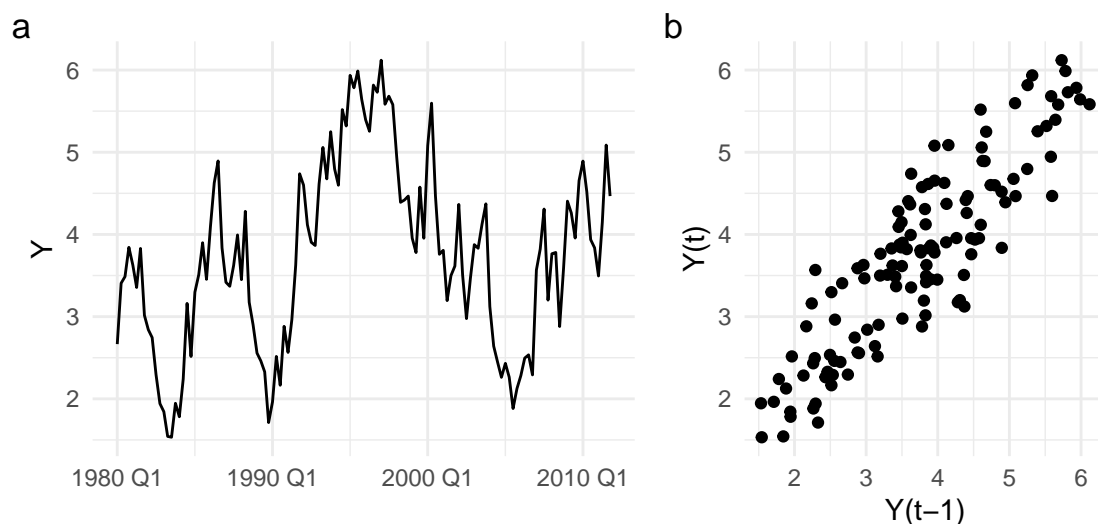


Figure 11.1: Series  $Y$  from `ts_02.xlsx`.

Economic time series often also display features such as trends and seasonality. Panel (a) of Fig. 11.2 shows a plot of Singapore’s monthly Industrial Production Index from 1983M1 to 2017M12, data in `ts_01.xlsx`. The upward trend seems the most dominant feature, though cyclical deviations from trend are also obvious. The size of the fluctuations are also increasing over time, which is not unusual in upward trending economic time series. Seasonality – repetitive



patterns that occur with regular periods – is also present. We zoom in on the sub-period 1990M1 to 1995M12 in Fig 2(b) where there seasonality is easier to see. There appears to be an annual pattern, with a sharp drop near the start of the year (usually in February) followed by a sharp positive response in the following month.

```
ts01 <- readxl::read_excel("data\\ts_01.xlsx") %>%
  select(DATE, IP_SG) %>%
  mutate(DATE=yearmonth(DATE)) %>%
  as_tsibble(index=DATE)
p1 <- ts01 %>%
  autoplot(IP_SG) + theme_minimal() + xlab("")
p2 <- ts01 %>%
  filter(DATE>=yearmonth("1990M1") & DATE<=yearmonth("1995M12")) %>%
  autoplot(IP_SG) + theme_minimal() + xlab("")
(p1 | p2) + plot_annotation(tag_levels="a")
```

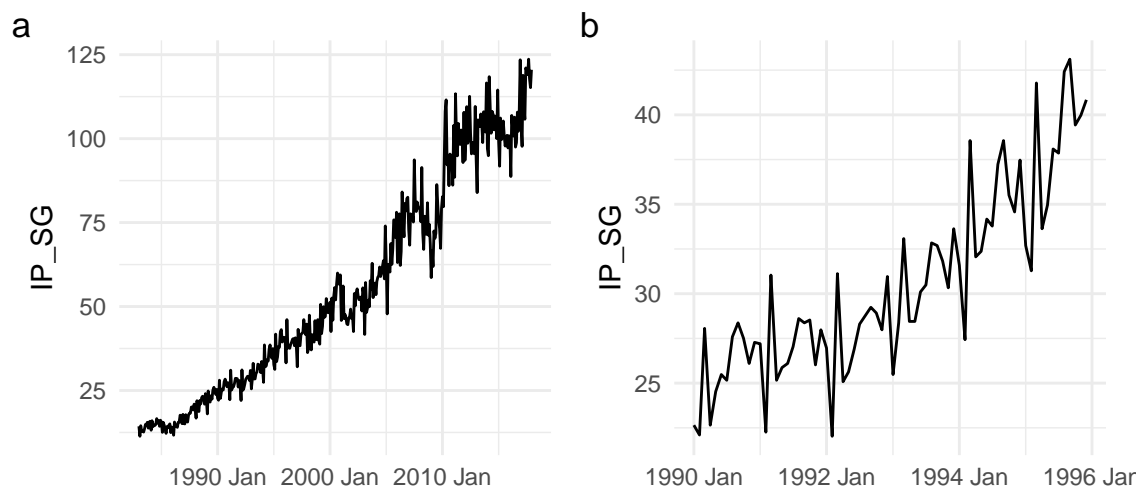


Figure 11.2: SG Industrial Production (IP\_SG).

## 11.2 Some Simple Time Series Models

We first discuss some basic tools for time series analysis, and a few simple *time series models* for cycles, trends and seasonality, so that we have a vocabulary for discussing such features and their consequences.

### 11.2.1 Transformations

We often apply some sort of transformation to a time series prior to analysis. For instances, we might work with the log of the time series rather than the series itself. Since the log function is strictly increasing, this does not affect the sign of the period-to-period changes in the time series. But as the log function is strictly-concave, applying a log-transformation attenuates fluctuations in the series, with larger fluctuations receiving greater attenuation. Log transformations therefore help to control the tendency of fluctuations in trending time series to increase over time.

Log transformations also linearize exponential trends, which is common in economic time series. Fig. 11.3 shows the log transformed IP\_SG.

```
p3 <- ts01 %>% autoplot(log(IP_SG)) + theme_minimal() + xlab("")
p3
```

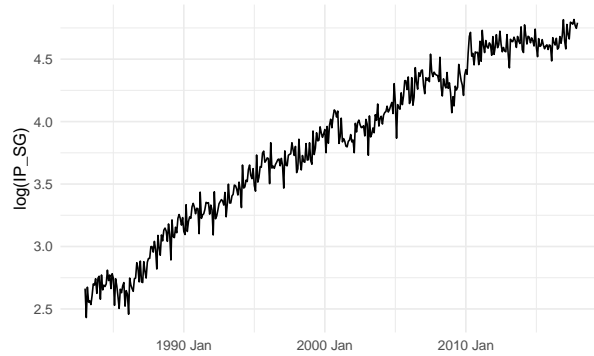


Figure 11.3:  $\log(\text{IP\_SG})$ .

Yet another advantage of the log transformation is that period-to-period percentage changes can be approximated by the first difference of log-transformed data (the “log-difference”):

$$\frac{Y_t - Y_{t-1}}{Y_{t-1}} \approx \ln Y_t - \ln Y_{t-1}.$$

This arises from the first-order linear approximation of the log function around  $Y_{t-1}$ . Alternatively, the log-difference can be interpreted as a continuous growth rate.

### 11.2.2 Sample Autocorrelation Function

We can summarize the serial correlation (or autocorrelation) in a time series by calculating its *sample autocorrelation function*. Given a time series  $Y_t$ ,  $t = 1, 2, \dots, T$ , define the *sample autocovariance function* to be

$$\hat{\gamma}_k = \frac{1}{T} \sum_{t=k+1}^T (Y_t - \bar{Y})(Y_{t-k} - \bar{Y}), \quad k = 0, 1, 2, \dots$$

where  $\bar{Y}$  is the sample mean defined in the usual way. This is, of course, just the usual sample covariance formula, except that here we measure a variable’s covariance *with itself* at some lag. Note that the summation index starts at  $t = k + 1$  instead of  $t = 1$ , because we need to take  $k$  lags of the variable. Despite only adding up  $T - k$  terms, we divide by  $T$  instead of  $T - k$  (we won’t get into the reasons why here). This is called a sample autocovariance *function* because we are computing the sample autocovariances at  $k = 0, 1, 2, \dots$ , (i.e., we have a function of  $k$ ). The sample autocovariance of  $Y_t$  at lag 0 is just the sample variance of  $Y_t$ . Finally, the sample autocorrelation function is defined as

$$\hat{\rho}_k = \frac{\hat{\gamma}_k}{\hat{\gamma}_0}.$$

Fig. 11.4 shows the sample acf for the  $Y_t$  series shown in Fig. 11.1. The dotted bands are the  $\pm 1.96$  standard errors of the sample acf, which helps us to determine significance of the autocorrelations. We see that the autocorrelations of this series decline as we consider observations further apart.

```
p1 <- ts02 %>% ACF(Y) %>% autoplot() + ylim(c(-1,1)) + theme_minimal()
p1
```

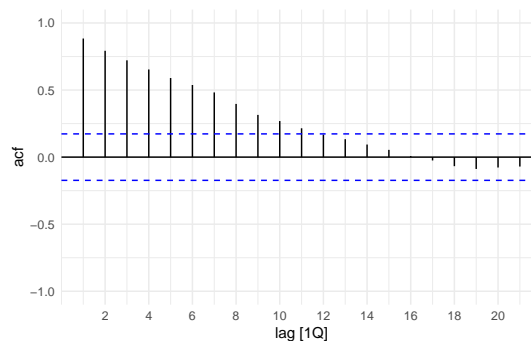


Figure 11.4: ACF: Series Y.

### 11.2.3 Trend

There are a number of ways to model trend. We can model a trending series as a “deterministic trend” process such as

$$Y_t = \beta_0 + \beta_1 t + \epsilon_t, t = 1, 2, \dots, T, \quad (11.1)$$

where for the moment we take  $\epsilon_t$  to be some zero-mean i.i.d. noise term. Here we indicate dates as an integer series, although this might represent some regular period like months or quarters. We can use any deterministic function of  $t$  for the trend. For example we can have a quadratic deterministic trend

$$Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \epsilon_t, t = 1, 2, \dots, T. \quad (11.2)$$

We fit using OLS the quadratic deterministic trend model to the  $\log(\text{IP\_SG})$  series. The fitted line and the residuals are shown in Fig. 11.5.

```
ts01 <- ts01 %>% mutate(t=1:length(ts01$IP_SG), tsq = t^2)
mdl_dqt <- lm(log(IP_SG) ~ t + tsq, data=ts01)
mdl_dqt %>% summary() %>% coef()
df_plot <- ts01 %>% mutate("Fitted"=fitted(mdl_dqt), "Residuals"=residuals(mdl_dqt))
p1 <- autoplot(df_plot, log(IP_SG), size=0.5) +
  autolayer(df_plot, Fitted) + theme_minimal() + xlab("")
p2 <- autoplot(df_plot, Residuals) + theme_minimal() + xlab("")
(p1 | p2) + plot_annotation(tag_levels = 'a')
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	2.519434e+00	1.529628e-02	164.70896	0.000000e+00
t	7.774090e-03	1.677910e-04	46.33197	1.517867e-166
tsq	-5.833467e-06	3.859553e-07	-15.11436	1.812010e-41

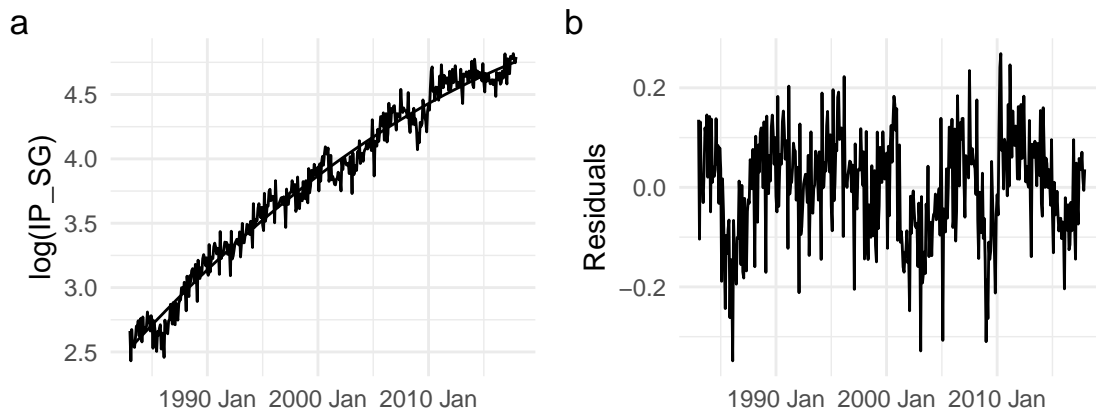


Figure 11.5: SG Quadratic Deterministic Trend, Fit and Residuals.

Of course, no series is likely to be a pure deterministic trend plus iid noise. In Fig. 11.5(b), the residuals display cycles and seasonality, and can be thought of as a de-trended version of the Industrial Production series. Sometimes the purpose of estimating a trend is precisely to obtain a de-trended series, so we can focus on the remaining components.

Instead of specifying a particular functional form for the trend, a more flexible way would be to use non-parametric methods. See Appendix A for a description of one such method.

Another way to model trend in a series is to say that on average the series changes every period by some value  $\alpha$ , i.e.,

$$Y_t - Y_{t-1} = \alpha + \epsilon_t,$$

where again for the moment we assume  $\epsilon_t$  to be some iid zero-mean noise term. We often denote the first difference  $Y_t - Y_{t-1}$  by  $\Delta Y_t$ . The model above is often written as

$$Y_t = \alpha + Y_{t-1} + \epsilon_t. \quad (11.3)$$

If  $\alpha > 0$ , then  $Y_t$  increases by an average of  $\alpha$  every period, and therefore trends upwards. The process (11.3) is called a **random walk with drift** if  $\alpha \neq 0$ , or a **random walk without drift** (or simply a “random walk”) if  $\alpha = 0$ .

There is an essential difference between the random walk approach (11.3) and deterministic trend approaches such as in (11.1) or (11.2). We can better understand (11.3) better by following the process starting from some fixed  $Y_0$ . Suppose that the variance of  $\epsilon_t$  is  $\sigma^2$ . We have

$$\begin{aligned} Y_1 &= \alpha + Y_0 + \epsilon_1 & \text{var}[Y_1|Y_0] &= \sigma^2 \\ Y_2 &= \alpha + Y_1 + \epsilon_2 = Y_0 + 2\alpha + \epsilon_1 + \epsilon_2 & \text{var}[Y_2|Y_0] &= 2\sigma^2 \\ Y_3 &= \alpha + Y_2 + \epsilon_3 = Y_0 + 3\alpha + \epsilon_1 + \epsilon_2 + \epsilon_3 & \text{var}[Y_3|Y_0] &= 3\sigma^2 \\ &\vdots & & \vdots \\ Y_t &= \alpha + Y_{t-1} + \epsilon_t = Y_0 + \alpha t + \epsilon_1 + \epsilon_2 + \epsilon_3 + \cdots + \epsilon_t & \text{var}[Y_t|Y_0] &= t\sigma^2 \end{aligned}$$

We see that if  $Y_t$  follows (11.3), then it contains a linear deterministic trend (if  $\alpha \neq 0$ ), but

unlike the linear deterministic trend process (11.1) there is also an accumulation of noise terms, resulting in a variance that increases steadily over time.

We simulate and plot in Fig. 11.6 one hundred series each of the “pure” linear deterministic trend process (11.1) with  $\beta_0 = 100$  and  $\beta_1 = 0.3$  (panel (a)), the random walk with drift (11.3) with  $\alpha = 0.3$  (panel (b)), and the “pure” random walk (11.3) with  $\alpha = 0$  (panel (c)). In the latter two cases, we start the process off at  $Y_0 = 100$ . For all three cases, we set the variance of the noise term at 1, and simulate series of 200 observations.

```
set.seed(20);
R <- 100; bigT <- 200; b0 <- 100; b1 <- 0.3; a0 <- 0.3; sigma <- 1; Y0 <- 100
dttrnd <- matrix(rep(0,R*bigT),ncol=R)
rwwd <- rwwod <- dttrnd
timeidx <- 1:bigT
for (r in 1:R){
  epsln <- rnorm(bigT,0,1)
  epscum <- cumsum(epsln)
  dttrnd[,r] <- b0 + b1*timeidx + epsln
  rwwd[,r] <- 100 + a0*timeidx + epscum
  rwwod[,r] <- 100 + epscum
}
theme1 <- theme_minimal() + theme(legend.position = "none")
p1 <- dttrnd %>% as.ts() %>% autoplot(facets=F, size=0.5) +
  ggtitle("Det. Trend") + ylim(c(50,200)) + theme1
p2 <- rwwd %>% as.ts() %>% autoplot(facets=F, size=0.5) +
  ggtitle("RW with drift") + ylim(c(50,200)) + theme1
p3 <- rwwod %>% as.ts() %>% autoplot(facets=F, size=0.5) +
  ggtitle("RW without drift") + ylim(c(50,200)) + theme1
(p1 | p2 | p3) + plot_annotation(tag_levels = 'a')
```

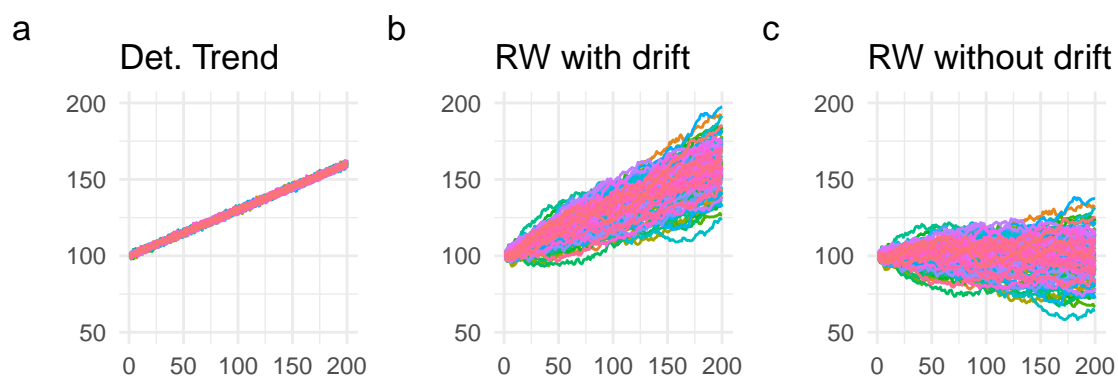


Figure 11.6: Simulated Deterministic Trends and Random Walks.

Because the ‘pure’ deterministic trend process is simply a deterministic function of  $t$  plus a non-accumulating noise term, such a process is very predictable. The random walk with drift, while trending upwards, is much less predictable because of the increasing variance. The random

walk without drift has no tendency to trend upwards, but also shows increasing variance.

Despite not containing a deterministic trend, the increasing variance in the random walk without drift means that in any finite sample one can observe a wide range of behaviors, including outcomes that appear to trend upwards or downwards, despite the fact that the average period-to-period growth rate is zero. Fig. 11.7 shows a few series drawn from the 200 simulations in panel (c) of Fig. 11.6. We refer to this behavior, which comes about because of the increasing variance, as the “stochastic trend”.

```
rwwood[,c(15, 20, 40, 65, 70, 95)] %>% as.ts() %>% autoplot(facet=F) + theme1
```

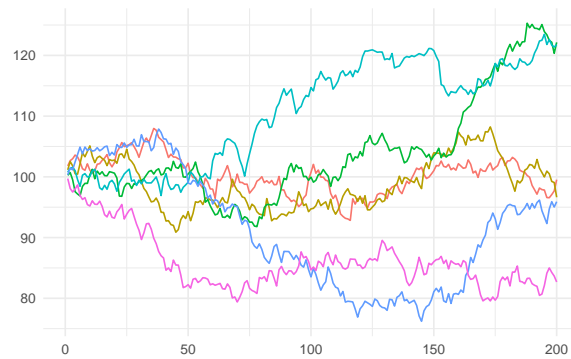


Figure 11.7: Selected Simulated Random Walks Without Drift.

In the random walk with drift, we have both the linear deterministic trend (due to the non-zero  $\alpha$ ) and the accumulating errors that lead to increasing variances. We refer to the linear deterministic trend part as the “drift” (hence the name “random walk with drift”), and the “increasing variance” part as the “stochastic trend”. The parameter  $\alpha$  is called the drift parameter.

- Deterministic Trend:  $Y_t = f(t) + \epsilon_t$ .
- Random Walk:

$$Y_t = \alpha + Y_{t-1} + \epsilon_t = \underbrace{Y_0 + \alpha t}_{\text{linear det. trend}} + \underbrace{\sum_{t=1}^T \epsilon_t}_{\text{stoc. trend}},$$

“with drift” if  $\alpha \neq 0$ , “without drift” otherwise.

We can estimate  $\alpha$  in the random walk with drift simply as the sample mean of  $\Delta Y_t$ . We estimate this for the log(IP\_SG) model below.

```
a0 <- mean(diff(log(ts01$IP_SG)))
df_plot2 <- ts01 %>% mutate(Fitted=log(log(IP_SG))+a0, Residuals=log(IP_SG)-Fitted)
p1 <- autoplot(df_plot2, log(IP_SG), size=0.5) +
  autolayer(df_plot2, Fitted) + theme_minimal() + xlab("")
p2 <- autoplot(df_plot2, Residuals) + theme_minimal() + xlab("")
(p1 | p2) + plot_annotation(tag_levels = 'a')
```

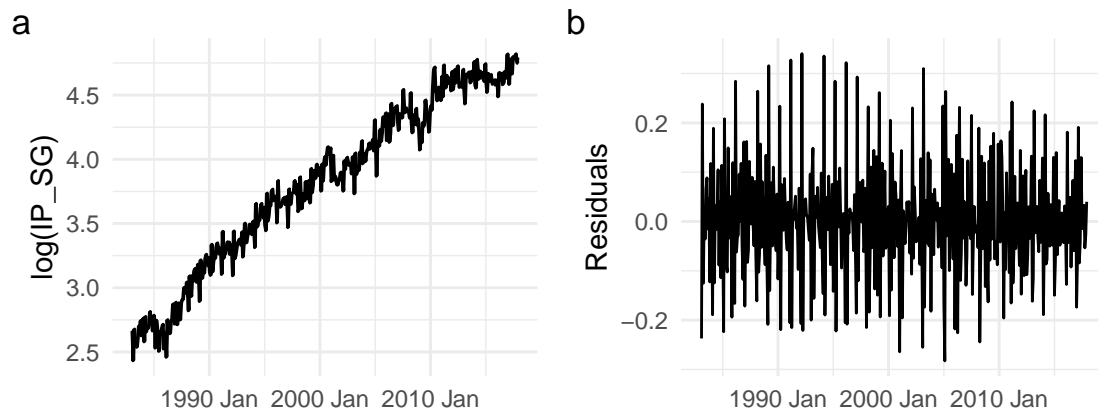
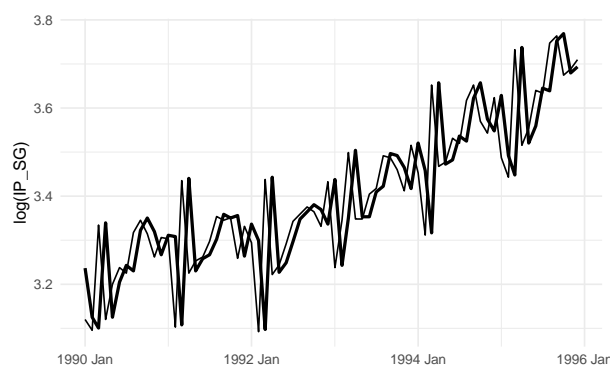


Figure 11.8: SG\_IP Random Walk with Drift, Fit and Residuals.

It is hard to make out the actual  $\log(\text{IP\_SG})$  and the fitted values in Fig. 11.8. We zoom in on a subsample in Fig. 11.9. The fitted values appear to be simply the lagged actual values. By construction, the fitted values are the lagged values plus the estimated growth rate  $\hat{\alpha}$ .

```
df_plot2sub <- df_plot2 %>%
  filter(
    DATE >= yearmonth("1990M1") & DATE <= yearmonth("1995M12")
  )
autoplot(df_plot2sub, log(IP_SG)) +
  autolayer(df_plot2sub, Fitted, size=1) +
  theme_minimal() + xlab("")
```

Figure 11.9:  $\log(\text{IP\_SG})$  Actual and Fitted (bold), Subsample.

To remove the stochastic trend and drift from a random walk process, simply take the first difference  $\Delta Y_t = Y_t - Y_{t-1}$ , or take the residuals from the fitted random walk model as in Fig. 11.8. In the latter the estimated growth rate is also removed.

While we have taken  $\epsilon_t$  to be i.i.d., for the moment, this will not be the case in most applications. We see in the residuals in Fig. 11.8 that there is seasonality, and less prominently, some cyclical patterns in the monthly growth rates.

Incidentally, trends show up in the sample acf as highly persistent autocorrelation. The following is the sample acf of the  $\log(\text{IP\_SG})$  series.

```
ts01 %>% ACF(log(IP_SG)) %>% autoplot() + ylim(c(-1,1)) + theme_minimal()
```

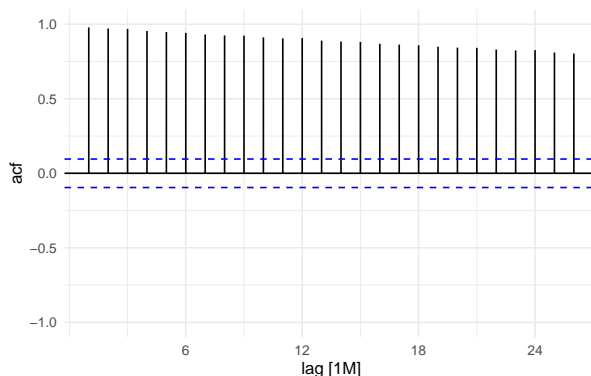


Figure 11.10: Sample ACF of IP\_SG.

Stochastic trends, even without drift, are also highly persistent processes. The following is the sample acf of one of the simulated random walks without drift that was shown in Fig. 11.6(c).

```
rwwod[,20] %>% as.ts() %>% as_tsibble() %>% ACF(value) %>%  
  autoplot() + ylim(c(-1,1)) + theme_minimal() + xlab("lags")
```

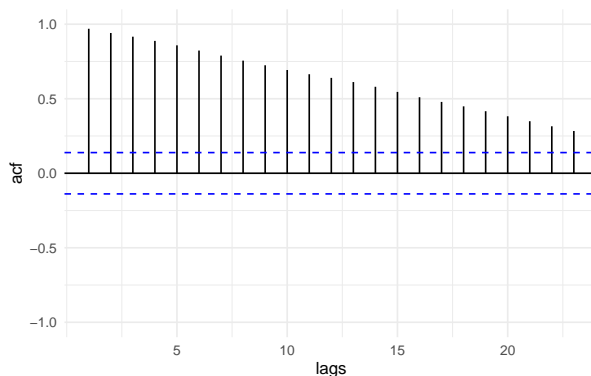


Figure 11.11: Sample ACF, RW without Drift.

We set aside for a later chapter the question of how to detect the presence of stochastic trend in a data series.

#### 11.2.4 Seasonality

Seasonals are patterns that occur with regular periods, often for ‘mechanical’ reasons, such as housing starts always being higher in the summer than in the winter, or tourist arrivals systematically peaking in the summer and at the end of the year.

Fig. 11.12 shows the “seasonal plot” for the IP\_SG growth rate, as measured by the first difference of the log-transformed IP\_SG series. We see that there is a very regular annual down-up-down pattern in industrial production growth over the Feb-Apr period, arising from the fewer number of calendar days in February, together with the Chinese New Year holidays which usually happen in February.



```
ts01 %>% gg_season(difference(log(IP_SG)), labels = "both") + theme_minimal()
```

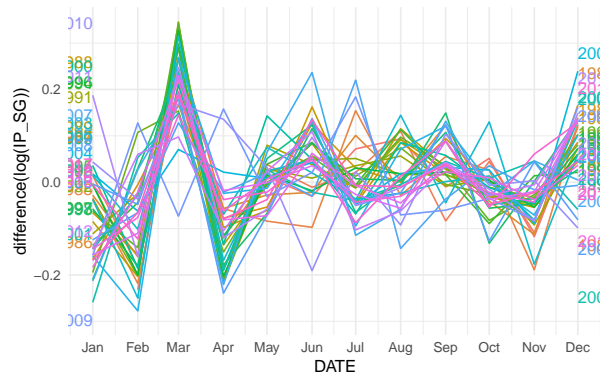


Figure 11.12: Seasonal Plot:  $d(\log(\text{IP\_SG}))$ .

Seasonality can also show up in the sample acf of a series. Fig. 11.13 shows the sample acf of the IP\_SG growth rate series.

```
ts01 %>% ACF(difference(log(IP_SG)), lag_max=48) %>% autoplot() +  
  ylim(c(-1,1)) + theme_minimal()
```

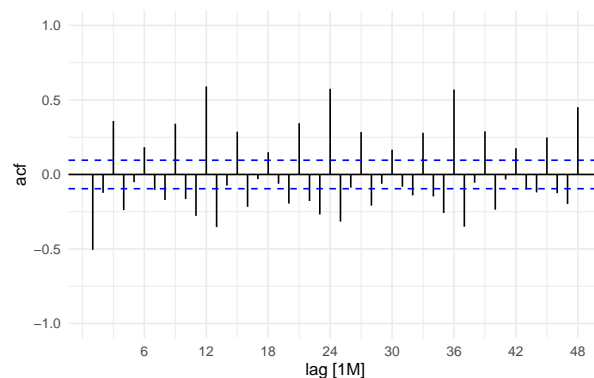


Figure 11.13: Sample ACF of  $d(\log(\text{IP\_SG}))$ .

Seasonality shows up as significant autocorrelations at multiples of the “seasonal lag”, which for monthly data is 12. These ‘seasonal spikes’ did not show up in the sample acf of trending  $\log(\text{IP\_SG})$  series in Fig. 11.10 because the seasonality (and all other features of the series) was overwhelmed by the trend, which is the dominant feature.

One way of modelling seasonal data is by using seasonal indicator variables (a.k.a. “seasonal dummy variables” or “seasonal dummies”). For monthly data, these are binary variables marking the month of the observation: the January dummy variable takes ‘1’ for all January observations, ‘0’ for all other observations; the February dummy variable takes ‘1’ for all February observations, ‘0’ for all others, and so on. The following 12 columns show the first 18 observations for each of the twelve monthly seasonal dummies for the IP\_SG data series.

```

ts01 <- ts01 %>%
  mutate(d01=ifelse(month(Date)==1,1,0),
         d02=ifelse(month(Date)==2,1,0),
         d03=ifelse(month(Date)==3,1,0),
         d04=ifelse(month(Date)==4,1,0),
         d05=ifelse(month(Date)==5,1,0),
         d06=ifelse(month(Date)==6,1,0),
         d07=ifelse(month(Date)==7,1,0),
         d08=ifelse(month(Date)==8,1,0),
         d09=ifelse(month(Date)==9,1,0),
         d10=ifelse(month(Date)==10,1,0),
         d11=ifelse(month(Date)==11,1,0),
         d12=ifelse(month(Date)==12,1,0))
ts01 %>% select(d01,d02,d03,d04,d05,d06,d07,d08,d09,d10,d11,d12) %>%
  filter(Date>=yearmonth("1985M1") & Date<=yearmonth("1986M6")) %>%
  knitr::kable()

```

d01	d02	d03	d04	d05	d06	d07	d08	d09	d10	d11	d12	DATE
1	0	0	0	0	0	0	0	0	0	0	0	1985 Jan
0	1	0	0	0	0	0	0	0	0	0	0	1985 Feb
0	0	1	0	0	0	0	0	0	0	0	0	1985 Mar
0	0	0	1	0	0	0	0	0	0	0	0	1985 Apr
0	0	0	0	1	0	0	0	0	0	0	0	1985 May
0	0	0	0	0	1	0	0	0	0	0	0	1985 Jun
0	0	0	0	0	0	1	0	0	0	0	0	1985 Jul
0	0	0	0	0	0	0	1	0	0	0	0	1985 Aug
0	0	0	0	0	0	0	0	1	0	0	0	1985 Sep
0	0	0	0	0	0	0	0	0	1	0	0	1985 Oct
0	0	0	0	0	0	0	0	0	0	1	0	1985 Nov
0	0	0	0	0	0	0	0	0	0	0	1	1985 Dec
1	0	0	0	0	0	0	0	0	0	0	0	1986 Jan
0	1	0	0	0	0	0	0	0	0	0	0	1986 Feb
0	0	1	0	0	0	0	0	0	0	0	0	1986 Mar
0	0	0	1	0	0	0	0	0	0	0	0	1986 Apr
0	0	0	0	1	0	0	0	0	0	0	0	1986 May
0	0	0	0	0	1	0	0	0	0	0	0	1986 Jun

We can model seasonality with seasonal dummies using the specification

$$Y_t = \beta_1 d_{1,t} + \beta_2 d_{2,t} + \cdots + \beta_{12} d_{12,t} + \epsilon_t \quad (11.4)$$

where  $d_{1,t}$  is the January dummy,  $d_{2,t}$  is the February dummy, and so on, and where we again (for the moment) take  $\epsilon_t$  to be a zero-mean iid noise term. This specification allows the mean of  $Y_t$  to depend on the ‘season’ (in this case, the month). For January observations, only  $d_{1,t} = 1$ , the other dummies are zero, therefore  $E[Y_t] = \beta_1$  for January observations. For February observations, only  $d_{2,t} = 1$ , the other dummies are zero, therefore  $E[Y_t] = \beta_2$  for February

observations, and so on.

Notice there is no intercept term in (11.4). This is because all of the dummies add up to a vector of ones. Including a constant will result in perfect collinearity (this is known as the dummy variable trap). If we wish to include the intercept, we will have to drop one of the dummy variables, as in (11.5) below, where we drop the January dummy.

$$Y_t = \alpha_0 + \alpha_2 d_{2,t} + \cdots + \alpha_{12} d_{12,t} + \epsilon_t \quad (11.5)$$

This changes the interpretation of the coefficients slightly. The parameter  $\alpha_0$  is now the January mean of  $Y_t$  and serves as the reference month. The February mean of  $Y_t$  is now  $\alpha_0 + \alpha_2$ , so  $\alpha_2$  is the difference between the February mean and the January mean. The other coefficients are interpreted similarly. We explore yet another equivalent specification in the exercises.

The seasonal dummies can be used in conjunction with other models we have discussed so far. For instance, we can have a quadratic deterministic trend model with seasonal dummies:

$$Y_t = \alpha_0 + \alpha_2 d_{2,t} + \cdots + \alpha_{12} d_{12,t} + \beta_1 t + \beta_2 t^2 + \epsilon_t \quad (11.6)$$

We fit (11.6) to the log(IP\_SG) model using OLS; the fitted values and residuals are shown in Fig. 11.14.

```
## recall we have previously added t and the seasonal dummies to ts01
mdl_dqt <- lm(log(IP_SG)~t+tsq+d02+d03+d04+d05+d06+d07+d08+d09+d10+d11+d12, data=ts01)
mdl_dqt %>% summary() %>% coef()
df_plot1 <- ts01 %>% mutate("Fitted"=fitted(mdl_dqt), "Residuals"=residuals(mdl_dqt))
p1 <- autoplot(df_plot1, log(IP_SG), size=0.5) +
  autolayer(df_plot1, Fitted) + theme_minimal() + xlab("")
p2 <- autoplot(df_plot1, Residuals) + theme_minimal() + xlab("")
(p1 | p2) + plot_annotation(tag_levels = 'a')
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	2.490454e+00	1.965402e-02	126.71473330	0.000000e+00
t	7.768060e-03	1.455858e-04	53.35728414	1.456574e-185
tsq	-5.832981e-06	3.348699e-07	-17.41864454	3.794588e-51
d02	-1.013478e-01	2.156942e-02	-4.69868204	3.590501e-06
d03	1.131723e-01	2.156951e-02	5.24686286	2.497552e-07
d04	-6.292830e-04	2.156967e-02	-0.02917445	9.767398e-01
d05	-2.884859e-03	2.156988e-02	-0.13374479	8.936707e-01
d06	4.298396e-02	2.157016e-02	1.99275152	4.695768e-02
d07	3.467546e-02	2.157049e-02	1.60754144	1.087130e-01
d08	4.523024e-02	2.157089e-02	2.09681856	3.662779e-02
d09	8.296558e-02	2.157135e-02	3.84610093	1.393025e-04
d10	5.959898e-02	2.157187e-02	2.76281065	5.991151e-03
d11	6.966586e-03	2.157245e-02	0.32293906	7.469076e-01
d12	8.191373e-02	2.157309e-02	3.79703297	1.687982e-04

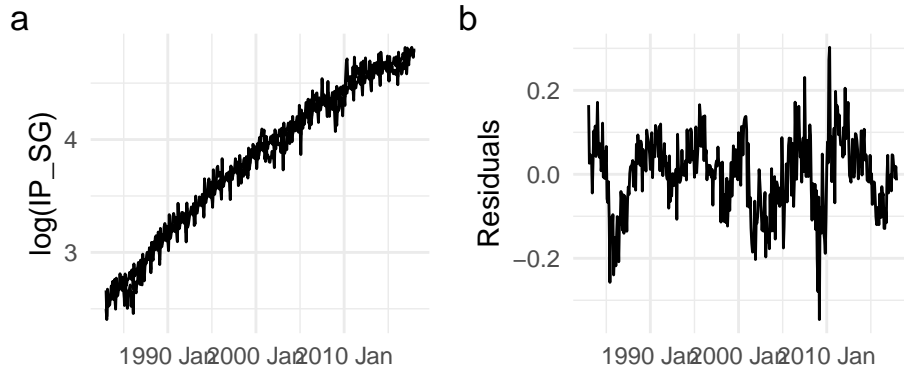


Figure 11.14: SG\_IP Quad. Det. Trend with Seas. Dummies, Fit and Residuals.

Fig. 11.15 zooms in on a subsample of the fit. The residuals in Fig. 11.14 can be thought of as de-trended “seasonally-adjusted”  $\log(\text{IP\_SG})$ .

```
df_plot1sub <- df_plot1 %>%
  filter(DATE>=yearmonth("1990M1") & DATE<=yearmonth("2000M12"))
autoplot(df_plot1sub, log(IP_SG)) + autolayer(df_plot1sub, Fitted, size=1) +
  theme_minimal() + xlab("")
```

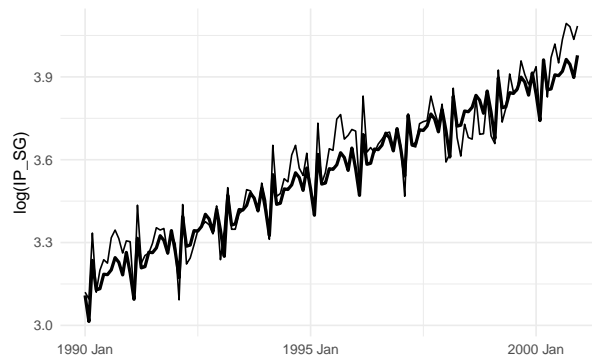


Figure 11.15:  $\log(\text{IP\_SG})$  Actual and Fitted (bold), subsample.

We can also fit a random walk with seasonal dummies by fitting the seasonal dummy model to the first differences:

$$Y_t - Y_{t-1} = \alpha_0 + \alpha_2 d_{2,t} + \cdots + \alpha_{12} d_{12,t} + \epsilon_t. \quad (11.7)$$

The fitted values can be obtained as

$$\hat{Y}_t = Y_{t-1} + \hat{\alpha}_0 + \hat{\alpha}_2 d_{2,t} + \cdots + \hat{\alpha}_{12} d_{12,t}, \quad t = 2, 3, \dots, T.$$

We fit this model to the  $\log(\text{IP\_SG})$  series, output shown below; fitted values and residuals are shown in Fig. 11.16. Fig. 11.17 zooms in on a smaller subsample. Presumably after removing trend and seasonality, only cyclical behavior remains. Notice that what we get after detrending

and seasonal adjustment can look very different, depending on what we assume about the trend and how we de-seasonalize. Compare Fig. 11.14(b) and Fig. 11.16(b).

```
ts01a <- ts01 %>% mutate(lIP_SG=log(IP_SG),
                        dlIP_SG=log(IP_SG)-log(lIP_SG)) %>% filter(
  DATE>=yearmonth("1985M2"))
mdl_rwseas <- lm(dlIP_SG~d02+d03+d04+d05+d06+d07+d09+d10+d11+d12,
  data=ts01a)
df_plot2 <- ts01a %>% mutate(Fitted=log(lIP_SG)+fitted(mdl_rwseas),
  Residuals=log(IP_SG)-Fitted)
p1 <- autoplot(df_plot2, log(IP_SG), size=0.5) +
  autolayer(df_plot2, Fitted) + theme_minimal() + xlab("")
p2 <- autoplot(df_plot2, Residuals) + theme_minimal() + xlab("")
(p1 | p2)
```

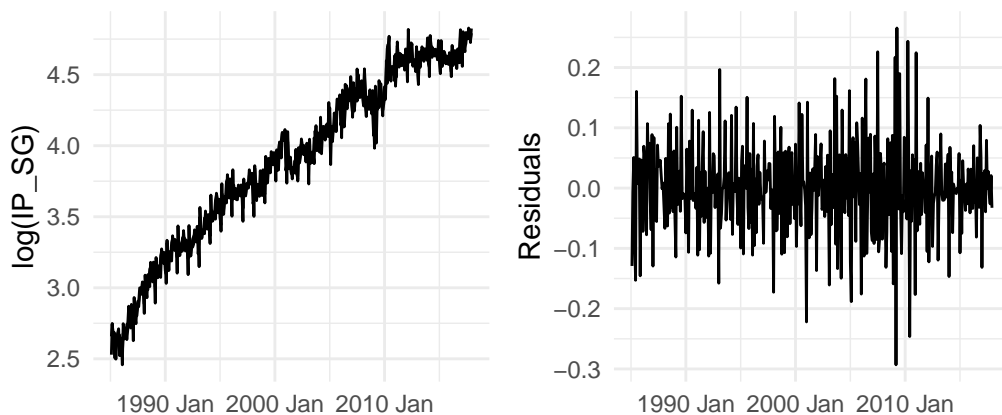


Figure 11.16: SG\_IP Random Walk with Drift, Fit and Residuals.

```
df_plot2sub <- df_plot2 %>%
  filter(
    DATE>=yearmonth("1990M1") & DATE<=yearmonth("1995M12"))
autoplot(df_plot2sub, log(IP_SG)) + autolayer(df_plot2sub, Fitted, size=1) +
  theme_minimal() + xlab("") +
  plot_annotation(subtitle="Fig 17. log(IP_SG) Actual and Fitted (bold), subsample")
```

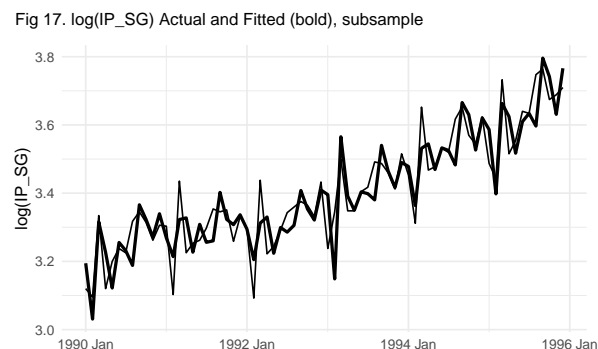


Figure 11.17: log(IP\_SG) Actual and Fitted (bold), subsample.

There are other ways to model seasonality. For instance, we can consider a “seasonal random walk”, which for monthly data would be

$$Y_t = Y_{t-12} + \epsilon_t.$$

This sort of seasonality can be dealt with by taking ‘seasonal differences’, i.e.,  $\Delta_{12}Y_t = Y_t - Y_{t-12}$ . This is in fact a fairly common approach when dealing with seasonal data.

Seasonally-adjusted (s.a.) versions of time series are often provided by official statistics agencies, sometimes together with the Non-Seasonally Adjusted (n.s.a.) versions, sometimes in place of it. Seasonal adjustment is often done by first estimating a “trend-cycle” (using sophisticated versions of the moving average method described in Appendix A), then estimating the seasonal component from the series with trend-cycle removed. This results in a decomposition of the original series into a trend-cycle component, a seasonal component, and an ‘irregular’ component. The seasonally-adjusted version of the data is obtained by removing the seasonal component from the original.

We apply one such method (called “X-11”) to  $\log(\text{IP\_SG})$ , with default settings. Fig. 11.18 shows the decomposition of  $\log(\text{IP\_SG})$  into its various components, and Fig. 11.19 shows the original and seasonally adjusted series.<sup>1</sup> Panel (b) zooms in on a sub-sample to better illustrate the seasonally adjusted series. It also highlights that care is needed when using default settings. The early year seasonal pattern in  $\text{IP\_SG}$  is driven by the Chinese New Year holidays, which typically occurs in February. Occasionally it falls in January, as it did in 1993. This might lead to what appears to be big ‘shocks’ where there were only moderate ones.

```
ipsg_dcmp <- ts01 %>% model(x11=X_13ARIMA_SEATS(log(IP_SG) ~ x11())) %>% components()
autoplot(ipsg_dcmp) + theme_minimal()
```

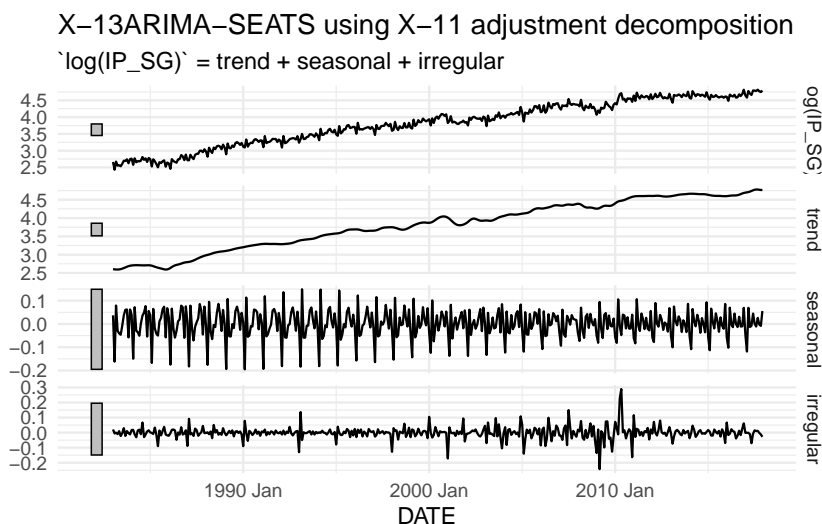


Figure 11.18: Decomposition of  $\log(\text{IP\_SG})$  using X-11.

<sup>1</sup>Decompositions can be “additive” or “multiplicative”. In this case it is additive, and the seasonally-adjusted series is obtained by subtracting the seasonal component from the original.

```

p2 <- ipsg_dcmp %>%
  ggplot(aes(x = DATE)) +
  geom_line(aes(y = `log(IP_SG)`), size=0.5) +
  geom_line(aes(y = season_adjust), size=1) + theme_minimal()
p3 <- ipsg_dcmp %>%
  filter(DATE >= yearmonth("1990M1") & DATE <= yearmonth("1995M12")) %>%
  ggplot(aes(x = DATE)) +
  geom_line(aes(y = `log(IP_SG)`), size=0.5) +
  geom_line(aes(y = season_adjust), size=1) + theme_minimal()
(p2 | p3) + plot_annotation(tag_levels = "a")

```

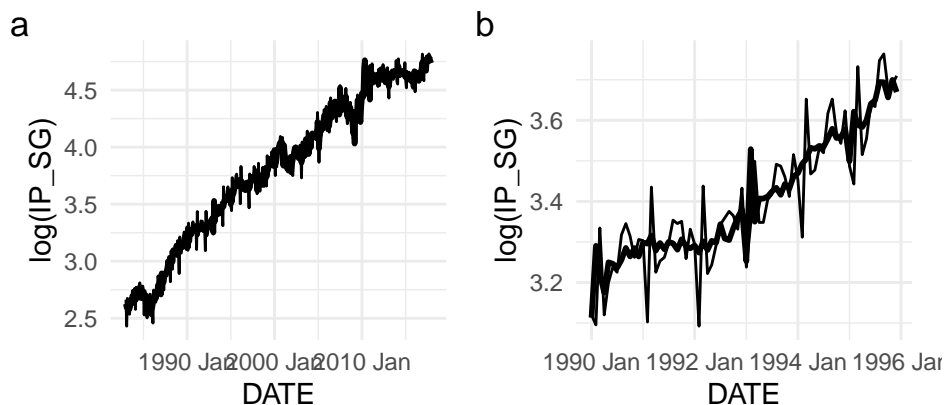


Figure 11.19:  $\log(\text{IP\_SG})$  sa. (bold) and nsa.

### 11.2.5 Cycles

Cycles are somewhat harder to define, and we will not attempt a definition here. Instead we will focus on serial correlation or autocorrelation, which as we pointed out earlier, can manifest as cyclical behavior in a time series. Returning to the series that was displayed in Fig. 11.1, the scatterplot in Fig. 11.1(b) suggested that perhaps a model such as

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \epsilon_t \quad (11.8)$$

may describe the behavior of the series well. Such a process is called an Autoregression of Order 1, or AR(1). In fact, the series in Fig. 11.1 was simulated from such a model, with  $\epsilon_t$  as some i.i.d. noise term.

One will recognize the Random Walk as an example of such an AR(1), with  $\beta_1 = 1$ , and we saw that such a process will have a stochastic trend, and also a drift if  $\beta_0 \neq 0$ . However, if  $-1 < \beta_1 < 1$ , then the AR(1) in (11.8) behaves quite differently. The following is another simulation of such a series, with  $\beta_1 = 0.7$ . This time we simulate a lengthy series, to illustrate a point about such series.

```

simAR <- function(a0, a1, bigT){
  burn = 100

```

```

Z <- rep(0,bigT+burn)
u <- rnorm(bigT+burn,0,1)
for (t in 2:(bigT+burn)){
  Z[t] <- a0 + a1*Z[t-1] + u[t]
}
return(Z[(burn+1):(burn+bigT)])
}
set.seed(13)
bigT=1000; a0 = 1; a1 = 0.8
X <- simAR(a0,a1,bigT)
dfplot <- data.frame(t=(1:bigT),X=X)
dfplot %>% ggplot() + geom_line(aes(x=t,y=X)) + theme_minimal()

```

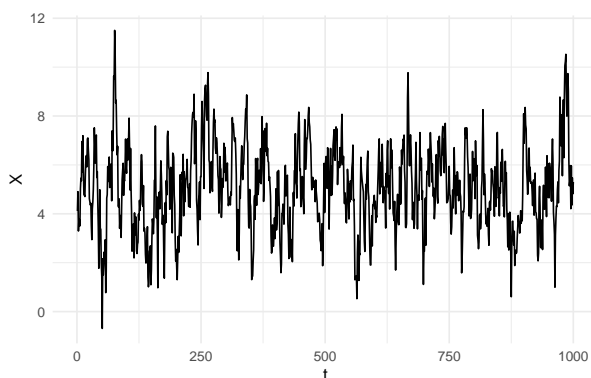


Figure 11.20: A Simulated AR(1).

There are cycles in this series of the boom-bust form (because we set  $\beta_1$  between zero and one), though the period and amplitude of each ‘cycle’ is not regular. Notice also that the series fluctuates around some constant value, and the overall size of the fluctuations appear fairly stable. In Fig. 11.21, we plot the sample acf for the full sample, as well as for three subsamples. We notice that the sample autocorrelations die off fairly quickly. Also, we see that the sample acf is very stable over the entire sample. This should be unsurprising, since the time series was simulated from a single model across the 1000 observations.

```

X <- as_tsibble(as.ts(X, freq=1))
Xearly <- X %>% filter(index>=1 & index<=300)
Xmiddle <- X %>% filter(index>=351 & index<=650)
Xlate <- X %>% filter(index>=700 & index<=1000)
p1 <- ACF(X,value,lag_max=16) %>% autoplot() + ylim(c(-1,1)) + theme_minimal()
p2 <- ACF(Xearly,value,lag_max=16) %>% autoplot() + ylim(c(-1,1)) + theme_minimal()
p3 <- ACF(Xmiddle,value,lag_max=16) %>% autoplot() + ylim(c(-1,1)) + theme_minimal()
p4 <- ACF(Xlate,value,lag_max=16) %>% autoplot() + ylim(c(-1,1)) + theme_minimal()
(p1 | p2) / (p3 | p4)

```



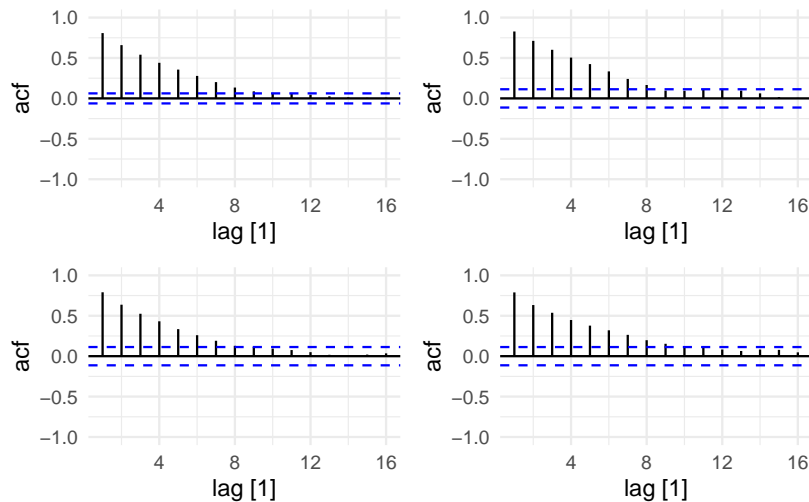


Figure 11.21: ACF of X, various subsamples.

A time series  $Y_t$  is said to be covariance-stationary if it has

- a constant and finite mean  $E[Y_t] < \infty$  for all  $t$ ,
- a constant and finite variance  $\text{var}[Y_t] < \infty$  for all  $t$ , and
- an autocorrelation function  $\text{cov}[Y_t, Y_{t-k}]$  that is finite and that may depend on  $k$  but not on  $t$ .

Covariance-stationary processes are processes that are “stable” in the sense given in its definition. A process whose autocorrelations dies off reasonably quickly as we consider observations further apart is said to be “weakly dependent”.<sup>2</sup> Processes like the random walk are not weakly dependent, but are “persistent”. Trending processes are not covariance-stationary. Seasonal processes may or may not be stationary, and may or may not be weakly dependent.

### 11.3 Time Series Regressions

We come now to time series regressions where we have ‘variables’ on the right-hand-side, rather than (only) functions of the time index or dummy variables. For the most part, we will begin by assuming that our variables are covariance-stationary and weakly dependent. If dealing with trending data, we assume that we have made whatever transformations are necessary (e.g., taking first differences, using seasonally-adjusted data) to obtain covariance-stationarity. Later we add deterministic trends and seasonal dummies to the mix.

#### 11.3.1 Dynamic Specifications

We first note that in time series regressions, we have the possibility of including lagged regressors, and also lagged dependent variables into the specification. Of course, we could simply have the “static” specification

$$Y_t = \alpha_0 + \alpha_1 X_t + \epsilon_t,$$

but in general we will want to consider “dynamic” specifications such as the following:

<sup>2</sup>There are formal ways to define “weakly dependent”, but we will make do with our informal characterization for now.

- “Distributed Lag Models”:

$$Y_t = \alpha_0 + \beta_0 X_t + \beta_1 X_{t-1} + \dots + \beta_q X_{t-q} + \epsilon_t. \quad (11.9)$$

Such models are useful because there is always the possibility that the influence of an explanatory variable may take several periods to fully work out. For instance, the effect of a change in interest rates on inflation may take several quarters to fully take effect. In (11.9), suppose there is a one-period only one-unit shock in  $X_t$ . The immediate effect on  $Y_t$  is  $\beta_0$ , but the effect on  $Y_t$  does not end there. Even though this is one-period only impulse in  $X$ , there is a lingering effect: the effect on  $Y_{t+1}$  is  $\beta_1$ , and that on  $Y_{t+2}$  is  $\beta_2$ , and so on. If there is a permanent change in  $X_t$  by one unit, then the effect on  $Y_t$  is cumulative: the total effect (or “long-run cumulative dynamic multiplier”) is  $\beta_0 + \beta_1 + \dots + \beta_q$ .

It should be noted that even when lags of  $X_t$  are included in the specification, it may well be that the noise term  $\epsilon_t$  are still not i.i.d., i.e., there may still be serial correlation in the noise term.

- We have already seen the stationary autoregressive model of order 1:

$$Y_t = \alpha_0 + \alpha_1 Y_{t-1} + \epsilon_t, \quad |\alpha_1| < 1.$$

Such models are used primarily to model cyclical dynamics in the data. There may be more than one lag of the dependent variable.

- “Autoregressive Distributed Lag (ARDL) models”:

$$Y_t = \alpha_0 + \alpha_1 Y_{t-1} + \beta_0 X_t + \beta_1 X_{t-1} + \dots + \beta_q X_{t-q} + \epsilon_t. \quad (11.10)$$

Such specifications imply an ‘infinite distributed lag structure’ for the effect of  $X_t$  or  $Y_t$ . Since (11.10) is assumed to hold for all  $t$ , we have

$$Y_{t-1} = \alpha_0 + \alpha_1 Y_{t-2} + \beta_0 X_{t-1} + \beta_1 X_{t-2} + \dots + \beta_q X_{t-q-1} + \epsilon_{t-1}.$$

Substituting into (11.10) gives

$$Y_t = \alpha_0(1 + \alpha_1) + \alpha_1^2 Y_{t-2} + \beta_0 X_t + (\beta_1 + \alpha_1 \beta_0) X_{t-1} + \dots + \alpha_1 \beta_q X_{t-q-1} + \epsilon_t + \alpha_1 \epsilon_{t-1}.$$

Continuing this process by substituting  $Y_{t-2}$ , then  $Y_{t-3}$  and so on, we get the infinite distributed lag structure on  $X_t$ .

We mentioned for the distributed lag model (11.9) that the noise term may contain cyclical dynamics. There is a close connection between such models and the autoregressive distributed lag specification. Suppose

$$Y_t = \alpha_0 + \beta_0 X_t + \beta_1 X_{t-1} + u_t, \quad u_t = \rho u_{t-1} + \epsilon_t, \quad |\rho| < 1 \quad (11.11)$$

where  $\epsilon_t$  is iid. This is a model with “covariance-stationary AR(1) errors”. Such models imply

an ARDL specification. Since

$$\rho Y_{t-1} = \rho\alpha_0 + \rho\beta_0 X_{t-1} + \rho\beta_1 X_{t-2} + \rho u_{t-1}, \quad (11.12)$$

subtracting (11.12) from (11.11) gives

$$Y_t - \rho Y_{t-1} = \alpha_0(1 - \rho) + \beta_0 X_t + (\beta_1 - \rho\beta_0)X_{t-1} + \rho\beta_1 X_{t-2} + u_t - \rho u_{t-1}$$

i.e.,

$$Y_t = \alpha_0(1 - \rho) + \rho Y_{t-1} + \beta_0 X_t + (\beta_1 - \rho\beta_0)X_{t-1} + \rho\beta_1 X_{t-2} + \epsilon_t.$$

If a dynamic model has i.i.d. errors, we will refer to it as a dynamically correct model. Of course, the dynamic structure in the error term might be much more complicated than a simple AR(1) model, and it may well be that even an ARDL specification might not be sufficient for obtaining a dynamically complete model.

### 11.3.2 Assumptions

We will write our (potentially dynamic) linear regression as

$$Y_t = x_t^T \beta + \epsilon_t$$

where  $x_t$  is a vector of regressors. Although this looks like a static specification, bear in mind that the vector  $x_t$  may contain lagged regressors or even lagged dependent variable. E.g., in the ARDL model (11.10), we have

$$x_t^T = [1 \quad Y_{t-1} \quad X_t \quad X_{t-1} \quad \dots \quad X_{t-q}]$$

and

$$\beta = [\alpha_0 \quad \alpha_1 \quad \beta_0 \quad \beta_1 \quad \dots \quad \beta_q]^T.$$

We will consider the usual OLS assumptions, and how they must be modified for dynamic time series regressions.

- First, in cross-sectional regressions we often assume iid draws. As we have already discussed, this is usually an untenable assumption for time series data. We allow for non-iid behavior in our time series, but for the moment, we will assume that the variables in our regression are covariance-stationary and weakly-dependent.
- In cross-sectional regressions, we usually assume  $E[\epsilon_i | x_1, x_2, \dots, x_N] = 0$ . Recall that this is the key assumption for unbiased OLS estimators. For time series regressions this assumption becomes

$$E[\epsilon_t | x_1, x_2, \dots, x_T] = 0. \quad (11.13)$$

We will refer to this assumption as “strong exogeneity”.

It turns out that this assumption is often too strong time series data. For example, take the AR(1) model

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \epsilon_t, \quad t = 2, 3, \dots, T$$

and suppose  $\epsilon_t$  are i.i.d. noise terms. Assumption (Eq. 11.13) then becomes

$$E[\epsilon_t | Y_1, Y_2, \dots, Y_{T-1}] = 0 \quad (11.14)$$

but this is impossible. If  $\epsilon_t$  is uncorrelated with  $Y_{t-1}$ , then it is definitely correlated with  $Y_t$ , whereas (11.14) says that  $\epsilon_t$  is uncorrelated with *all*  $Y_t$ ,  $t = 2, \dots, T$ . This argument applies for all specifications with lagged dependent variables. Strong exogeneity cannot hold in any specification that includes a lagged dependent variable as a regressor. Even if you do not have lagged dependent variables, strong exogeneity may still not hold. For instance, you may be forced to omit variables in your equation that can help forecast future outcomes of your regressor. Strong exogeneity will also not hold in such cases.

OLS estimators will be biased if strong exogeneity does not hold (since it is a necessary condition for unbiasedness of OLS estimators). Fortunately, it can be shown that OLS estimators of time series regression with covariance-stationary weakly-dependent variables will remain consistent as long as our noise terms satisfy **contemporaneous exogeneity**:

$$E[\epsilon_t | x_t] = 0 \quad (11.15)$$

(note that the conditioning information ends with  $t$ , not  $T$ ). This is a much weaker assumption, and can hold even if you have lagged dependent variables. We will assume that we have enough variables, and lags of variables included in the regression to allow this assumption to hold.

We have discussed homoskedasticity vs heteroskedasticity in cross-sectional regressions. For our discussion of time series regressions, we shall allow for conditional heteroskedasticity, and simply note that this means our OLS regressions are not necessarily efficient. If we are willing to assume a form of heteroskedasticity, then we may be able to apply weighted least squares to obtain more efficient estimates.

We have also assumed uncorrelated errors for our cross-sectional regressions. For time series regressions this assumption would be that there is no serial correlation or autocorrelation in the noise terms, and may or may not be appropriate. For models without lagged dependent variables, such as the distributed lag model (11.9), the assumption of no serial correlation seems less likely to hold, and the question is how we have to adapt our OLS methods and formulas to accommodate this. For models with lagged dependent variables, we will argue that we ought to try to ensure our dynamic specification is rich enough that our errors are uncorrelated, although this may not always be possible.

### 11.3.3 Standard Errors for Dynamically Incomplete Models

Suppose our linear regression model is

$$Y_t = x_t^T \beta + \epsilon_t$$

where for the moment we assume that there is no lagged dependent variable in  $x_t$ . We assume that our variables are covariance-stationary and weakly-dependent. We assume contemporaneous exogeneity, but allow for conditional heteroskedasticity and serial correlation in the noise terms. Then our OLS estimator for  $\beta$  is consistent and asymptotically normal. We shall omit

the proof of this statement for this chapter.

Note that the usual standard formula for the variance of  $\hat{\beta}$ ,

$$\widehat{\text{var}}[\hat{\beta}] = \widehat{\sigma^2} \left( \sum_{t=1}^T x_t x_t^T \right)^{-1} \quad (11.16)$$

is based on assumptions of homoskedasticity and uncorrelated errors, so if those assumptions do not hold, this formula is unreliable. If we have heteroskedastic but uncorrelated errors, then we can use the heteroskedasticity-robust “sandwich” estimator for the variance of  $\hat{\beta}$ :

$$\widehat{\text{var}}[\hat{\beta}] = \left( \sum_{t=1}^T x_t x_t^T \right)^{-1} \left( \sum_{t=1}^T \hat{\epsilon}_t^2 x_t x_t^T \right) \left( \sum_{t=1}^T x_t x_t^T \right)^{-1}. \quad (11.17)$$

If there are concerns about correlation in the errors (actually, the issue arises if there is serial correlation in  $x_t \epsilon_t$ ) then we can use the “heteroskedasticity and autocorrelation robust” (HAC) variance estimator

$$\widehat{\text{var}}[\hat{\beta}] = \left( \sum_{t=1}^T x_t x_t^T \right)^{-1} \left( \sum_{t=1}^T \hat{\epsilon}_t^2 x_t x_t^T + \sum_{v=1}^q \left( 1 - \frac{v}{q+1} \right) (x_t x_{t-v}^T + x_{t-v} x_t^T) \epsilon_t \epsilon_{t-v} \right) \left( \sum_{t=1}^T x_t x_t^T \right)^{-1}. \quad (11.18)$$

or one of its variants. It is the middle portion of (11.17) that is extended to allow for serial correlation in  $x_t \epsilon_t$ .

We illustrate the HAC variance estimator with a simulated example where  $\{X_t, Y_t\}_{t=1}^{100}$  follow

$$\begin{aligned} X_t &= 0.8 + 0.8X_{t-1} + \epsilon_t, \quad \epsilon_t \sim_{iid} N(0, 1) \\ Y_t &= 0.8 + 0X_t + u_t, \quad u_t = 0.95u_{t-1} + v_t, \quad v_t \sim_{iid} N(0, 1). \end{aligned}$$

In this example, there is no relation between  $Y_t$  and  $X_t$ . The  $Y_t$  series is a constant plus an AR(1) noise term. The  $X_t$  regressor is also an AR(1). We run a regression of  $Y_t$  on  $X_t$  and calculate the variance-covariance matrix of  $\hat{\beta}_0$  and  $\hat{\beta}_1$  three different ways, using (11.16), the “HC2” version of (11.17), and a version of (11.18) called “Newey-West”. The standard errors are the square root of the diagonal of these variance-covariance matrices.

```
set.seed(1313) # seed chosen randomly
Xsim <- simAR(0.8,0.8,100)
Ysim <- simAR(0.8,0.95,100)
df <- as_tsibble(data.frame(Xsim,Ysim,t=1:100),index=t)
p1 <- df %>% autoplot(Xsim) + theme_minimal()
p2 <- df %>% autoplot(Ysim) + theme_minimal()
p3 <- df %>% ggplot() + geom_point(aes(x=Xsim,y=Ysim)) + theme_minimal()
(p1 | p2 | p3)
mdlsm <- lm(Ysim~Xsim, data=df)
cat("OLS with Usual Standard Errors\n")
mdlsm %>% lmtest::coeftest()
cat("OLS with Heteroskedasticity-Robust S.E.\n")
```

```
lmtest::coeftest(mdl$sim, vcov=sandwich::vcovHC(mdl$sim, type="HC2"))
cat("OLS with Heteroskedasticity and Autocorrelation (HAC) Robust S.E.\n")
lmtest::coeftest(mdl$sim, vcov=sandwich::NeweyWest)
```

OLS with Usual Standard Errors

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	14.24286	0.79751	17.8592	< 2.2e-16 ***
Xsim	0.53330	0.17928	2.9748	0.003692 **

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

OLS with Heteroskedasticity-Robust S.E.

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	14.24286	0.85027	16.7509	< 2e-16 ***
Xsim	0.53330	0.18137	2.9404	0.00409 **

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

OLS with Heteroskedasticity and Autocorrelation (HAC) Robust S.E.

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	14.24286	2.06874	6.8848	5.551e-10 ***
Xsim	0.53330	0.34783	1.5332	0.1284

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

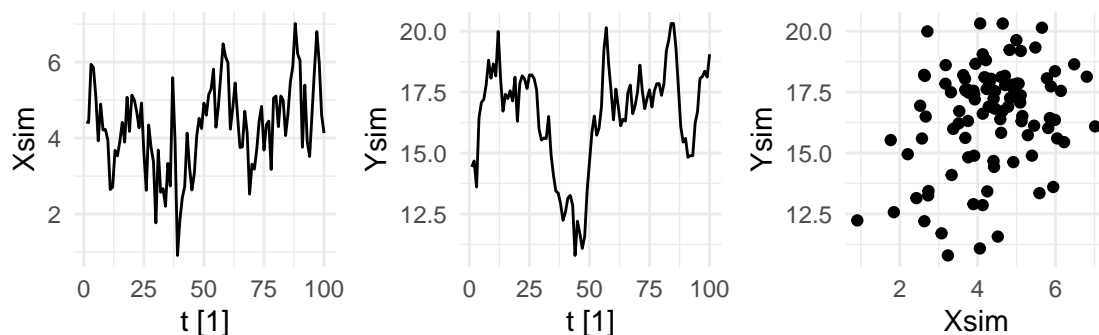


Figure 11.22: Xsim, Ysim and Scatterplot.

We see that the usual standard errors lead to incorrect inferences. The heteroskedasticity-robust standard errors are similar to the usual standard errors, which is not surprising since there is in fact no heteroskedasticity in this data set. They also lead to incorrect inference on the coefficient of `Xsim`. The HAC standard errors are substantially larger, resulting in smaller t-statistics, and correct inference. The HAC standard errors are appropriate as  $X_t\epsilon_t$  is correlated (see Fig. 11.23)

```
ACF(df, Xsim*residuals(mdlsim)) %>% autoplot() + theme_minimal()
```

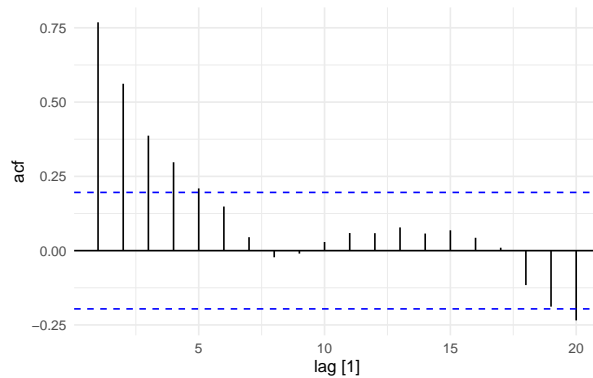


Figure 11.23: Correlation of Residuals Times Regressor.

#### 11.3.4 Dynamically Complete Models

Often we try to ensure that our time series models are dynamically complete, i.e., that sufficient number of lags of the dependent and independent variables are included so that there are no dynamics left in the noise term. One reason for this is if the model is being built for forecasting purposes. In that case we want to ensure that the full dynamic properties of the time series is accounted for by the model. Another reason is that sometimes we want to include lagged dependent variables in the specification, and the presence of both lagged dependent variables and serial correlation may lead to inconsistent estimators. For instance, suppose

$$Y_t = \alpha_0 + \alpha_1 Y_{t-1} + u_t$$

with  $|\alpha_1| < 1$  and where  $u_t = \epsilon_t + \theta\epsilon_{t-1}$ ,  $\epsilon_t \sim_{iid} (0, \sigma^2)$ . Then the error term  $u_t$  is correlated with  $Y_{t-1}$  (see exercises for details). This means that contemporaneous exogeneity does not hold, resulting in inconsistent OLS estimators.

#### 11.3.5 Testing for Autocorrelation

To check for autocorrelation in a regression, it is common practice to plot the sample acf of the residuals, i.e., after running the regression

$$Y_t = x_t^T \beta + \epsilon_t$$

we collect the residuals  $\hat{\epsilon}_t$  and compute its sample acf. This often gives us a good indication of the presence of serial correlation in the errors.

To more formally test for autocorrelation, one can run a regression of  $\hat{\epsilon}_t$  on lags of  $\hat{\epsilon}_t$ , and on the regressors, i.e., regress

$$\hat{\epsilon}_t \text{ on } \hat{\epsilon}_{t-1}, \dots, \hat{\epsilon}_{t-p}, x_{1,t}, \dots, x_{K-1,t}$$

where  $x_{1,t}, \dots, x_{K-1,t}$  are the regressors in  $x_t^T$ . Then test for the significance of the coefficients on the lagged residuals.

### 11.3.6 Regression with Trending and Persistent Series

If there is trend or seasonality in the time series in the regression, these must be accounted for. In the following example, we regress  $\log(\text{ELEC\_GEN\_SG})$  on  $\log(\text{IP\_SG})$  twice, once without seasonals or trend, and another time with both. That is, we run the regressions

$$Y_t = \alpha_0 + \beta_0 X_t + \epsilon_t \quad (11.19)$$

and

$$Y_t = \alpha_0 + \beta_0 X_t + \text{seas. dummies} + \delta_1 t + \delta_2 t^2 + \epsilon_t \quad (11.20)$$

```
ts03 <- readxl::read_excel("data\\ts_01.xlsx") %>%
  select(DATE, IP_SG, ELEC_GEN_SG) %>%
  mutate(DATE=yearmonth(DATE)) %>%
  as_tsibble(index=DATE)

## We use the TSLM() function in fpp3 which allows us to include seasonal dummies
## and trend simply by including season() and trend() in the regression formula
mdl_elec1 <- ts03 %>% model(TSLM(log(ELEC_GEN_SG)~log(IP_SG)))
report(mdl_elec1)
```

Series: ELEC\_GEN\_SG

Model: TSLM

Transformation: log(ELEC\_GEN\_SG)

Residuals:

	Min	1Q	Median	3Q	Max
	-0.23353	-0.06955	-0.01108	0.06348	0.28123

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	4.510243	0.028988	155.6	<2e-16 ***
log(IP_SG)	0.836081	0.007494	111.6	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.101 on 418 degrees of freedom

Multiple R-squared: 0.9675, Adjusted R-squared: 0.9674

F-statistic: 1.245e+04 on 1 and 418 DF, p-value: < 2.22e-16

```
mdl_elec2 <- ts03 %>% model(TSLM(log(ELEC_GEN_SG)~log(IP_SG)+season()+trend()+I(trend()^2)))
report(mdl_elec2)
```



Series: ELEC\_GEN\_SG

Model: TSLM

Transformation: log(ELEC\_GEN\_SG)

Residuals:

	Min	1Q	Median	3Q	Max
	-0.095765	-0.019684	-0.003458	0.021550	0.101889

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	6.235e+00	4.290e-02	145.325	< 2e-16 ***
log(IP_SG)	8.882e-02	1.701e-02	5.221	2.85e-07 ***
season()year2	-8.801e-02	7.592e-03	-11.593	< 2e-16 ***
season()year3	3.301e-02	7.640e-03	4.321	1.96e-05 ***
season()year4	2.397e-02	7.394e-03	3.242	0.00128 **
season()year5	6.398e-02	7.394e-03	8.654	< 2e-16 ***
season()year6	3.126e-02	7.430e-03	4.207	3.19e-05 ***
season()year7	5.519e-02	7.418e-03	7.441	6.04e-13 ***
season()year8	4.575e-02	7.434e-03	6.153	1.82e-09 ***
season()year9	1.455e-02	7.528e-03	1.933	0.05389 .
season()year10	4.584e-02	7.464e-03	6.142	1.95e-09 ***
season()year11	-9.464e-04	7.396e-03	-0.128	0.89824
season()year12	-3.467e-03	7.525e-03	-0.461	0.64524
trend()	7.748e-03	1.413e-04	54.846	< 2e-16 ***
I(trend()^2)	-8.901e-06	1.517e-07	-58.661	< 2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03093 on 405 degrees of freedom

Multiple R-squared: 0.997, Adjusted R-squared: 0.9969

F-statistic: 9768 on 14 and 405 DF, p-value: < 2.22e-16

The log(IP\_SG) variable is significant in both regressions, but its estimated coefficient in the regression with trend and seasonal dummies is one-tenth of that in the regression without trend and seasonality. Notice that the  $R^2$  is very high. This is usually the case in regressions with trend.

The model is dynamically incomplete. There is autocorrelation in the error terms, as can be seen from the ACF of the residuals, reported in Fig. 11.24 below.

```
p1 <- autoplot(residuals mdl_elec2, .resid) + theme_minimal()
p2 <- ACF(residuals mdl_elec2, .resid) %>% autoplot() + theme_minimal()
(p1 | p2) + plot_annotation(tag_levels = 'a')
```

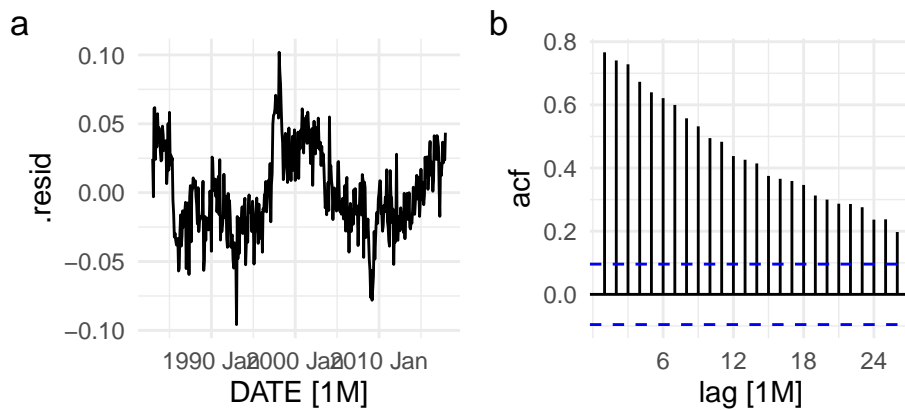


Figure 11.24: Residual and Residual ACF.

We add three lags of  $\log(\text{ELEC\_GEN\_SG})$  in the next model, i.e., we fit the model

$$Y_t = \alpha_0 + \beta_0 X_t + \gamma_1 Y_{t-1} + \gamma_2 Y_{t-2} + \gamma_3 Y_{t-3} + \text{seas. dummies} + \delta_1 t + \delta_2 t^2 + \epsilon_t \quad (11.21)$$

Fig. 11.25 shows the fit (in terms of  $\text{ELEC\_GEN\_SG}$ , not  $\log(\text{ELEC\_GEN\_SG})$ ), the residuals, and the ACF of the residuals.

```
mdl_elec3 <- ts03 %>% model(TSLM(log(ELEC_GEN_SG)~
                             log(IP_SG) +
                             lag(log(ELEC_GEN_SG),1)+
                             lag(log(ELEC_GEN_SG),2)+
                             lag(log(ELEC_GEN_SG),3)+
                             season()+trend()+I(trend()^2)))
report(mdl_elec3)
```

Series: ELEC\_GEN\_SG

Model: TSLM

Transformation:  $\log(\text{ELEC\_GEN\_SG})$

Residuals:

	Min	1Q	Median	3Q	Max
	-0.0600626	-0.0101659	-0.0001068	0.0115216	0.0646623

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	7.281e-01	2.029e-01	3.588	0.000375	***
log(IP_SG)	5.900e-02	1.015e-02	5.810	1.28e-08	***
lag(log(ELEC_GEN_SG), 1)	3.427e-01	4.624e-02	7.411	7.56e-13	***
lag(log(ELEC_GEN_SG), 2)	2.662e-01	4.697e-02	5.668	2.77e-08	***
lag(log(ELEC_GEN_SG), 3)	2.553e-01	4.609e-02	5.539	5.53e-08	***
season()year2	-7.666e-02	5.145e-03	-14.901	< 2e-16	***
season()year3	8.175e-02	6.241e-03	13.099	< 2e-16	***

```

season()year4      4.937e-02  7.095e-03   6.958 1.42e-11 ***
season()year5      8.335e-02  9.303e-03   8.959 < 2e-16 ***
season()year6      7.674e-03  5.207e-03   1.474 0.141365
season()year7      3.547e-02  5.743e-03   6.175 1.63e-09 ***
season()year8      1.584e-02  4.830e-03   3.280 0.001130 **
season()year9     -1.017e-02  5.507e-03  -1.847 0.065446 .
season()year10     2.630e-02  4.825e-03   5.450 8.83e-08 ***
season()year11    -2.249e-02  4.846e-03  -4.641 4.72e-06 ***
season()year12    -5.814e-03  5.513e-03  -1.055 0.292223
trend()           6.625e-04  2.737e-04   2.421 0.015939 *
I(trend()^2)     -9.391e-07  3.067e-07  -3.062 0.002346 **
---

```

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01825 on 399 degrees of freedom

Multiple R-squared: 0.999, Adjusted R-squared: 0.9989

F-statistic: 2.237e+04 on 17 and 399 DF, p-value: < 2.22e-16

```

p0 <- autoplot(ts03, ELEC_GEN_SG, color="blue") +
  autolayer(fitted mdl_elec3, .fitted, color="green") +
  theme_minimal()

p1 <- autoplot(residuals(mdl_elec3), .resid) + theme_minimal()
p2 <- ACF(residuals(mdl_elec3), .resid) %>%
  autoplot() + theme_minimal() + ylim(c(-1,1))
p0 / (p1 | p2) + plot_annotation(tag_levels = 'a')

```

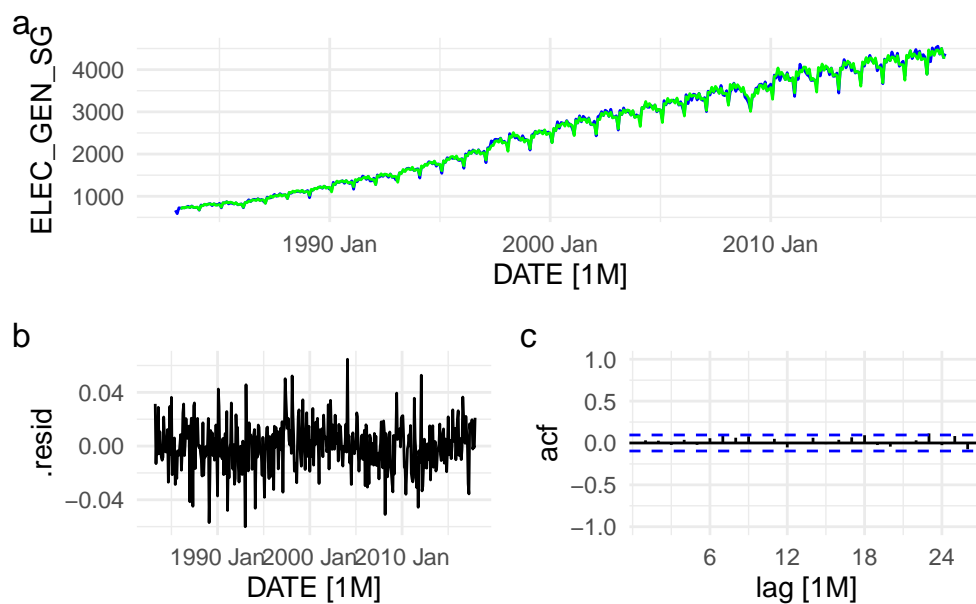


Figure 11.25: Residuals and Residual ACF.

### 11.3.7 Spurious Regressions

We run another experiment as an illustration of what can go wrong if one is not working with weakly dependent data. We simulate 200 pairs of unrelated series  $(X_t^{(r)}, Y_t^{(r)})_{t=1}^{100}$ ,  $r = 1, 2, \dots, R$  as random walks:

$$\begin{aligned} X_t^{(r)} &= \alpha_X + X_{t-1}^{(r)} + u_t^{(r)} \\ Y_t^{(r)} &= \alpha_Y + Y_{t-1}^{(r)} + v_t^{(r)} \end{aligned}$$

where  $u_t^{(r)}$  and  $v_t^{(r)}$  are independent  $\text{Normal}(0, 1)$  noise terms. We set  $\alpha_X = 0.5$  and  $\alpha_Y = 0.8$ . As explained earlier, these are trending series. For each replication  $r$ , we regress  $Y_t^{(r)}$  on  $X_t^{(r)}$ , with intercept, and collect the t-statistic on the coefficient of  $X_t^{(r)}$ . We replicate this experiment 200 times. The histogram of the 200 t-statistics is shown in panel (a) below.

```
simRW <- function(a0, bigT){
  Z <- rep(0, bigT)
  u <- rnorm(bigT, 0, 1)
  for (t in 2:bigT){
    Z[t] <- a0 + Z[t-1] + u[t]
  }
  return(Z)
}

simExp <- function(R, bigT, aX, aY){
  tstats <- rep(NA, R)
  for (r in 1:R){
    X <- simRW(aX, bigT)
    Y <- simRW(aY, bigT)
    df <- data.frame(X, Y)
    mdl <- lm(Y ~ X, data = df)
    tstats[r] <- summary(mdl)$coefficients[2, 't value']
  }
  return(tstats)
}

set.seed(13);
tstats1 <- simExp(200, 100, 0.4, 0.8)
df <- data.frame(tstats1)
ggplot(df, aes(x = tstats1)) +
  geom_histogram(binwidth = 3, color = "black", fill = "grey") + theme_minimal()
```

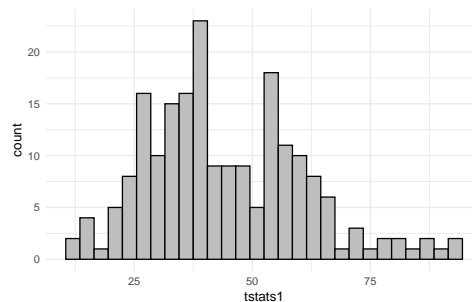


Figure 11.26: t-statistic Histogram.

Because  $X_t^{(r)}$  and  $Y_t^{(r)}$  are completely unrelated, one might expect the estimate of the  $X_t^{(r)}$  coefficient to be statistically insignificant. However, the histogram in Fig. 11.26 above shows that *all* of the t-statistics over our 200 replications are greater than 2. Given that the two series are trending, this is not too surprising. Fig. 11.27 shows the time series plots of one such pair (panels a and b), together with their scatterplot (panel c). It is clear that the linear regression of  $Y_t^{(r)}$  on  $X_t^{(r)}$  is merely capturing the fact that both series are increasing over time.

```
set.seed(13)
X <- simRW(0.4,100)
Y <- simRW(0.8,100)
dfplot <- data.frame(t=1:100,X,Y)
p1 <- dfplot %>% ggplot() + geom_point(aes(x=t,y=X)) + theme_minimal()
p2 <- dfplot %>% ggplot() + geom_point(aes(x=t,y=Y)) + theme_minimal()
p3 <- dfplot %>% ggplot() + geom_point(aes(x=X,y=Y)) + theme_minimal()
(p1 | p2 | p3) + plot_annotation(tag_levels = "a")
```

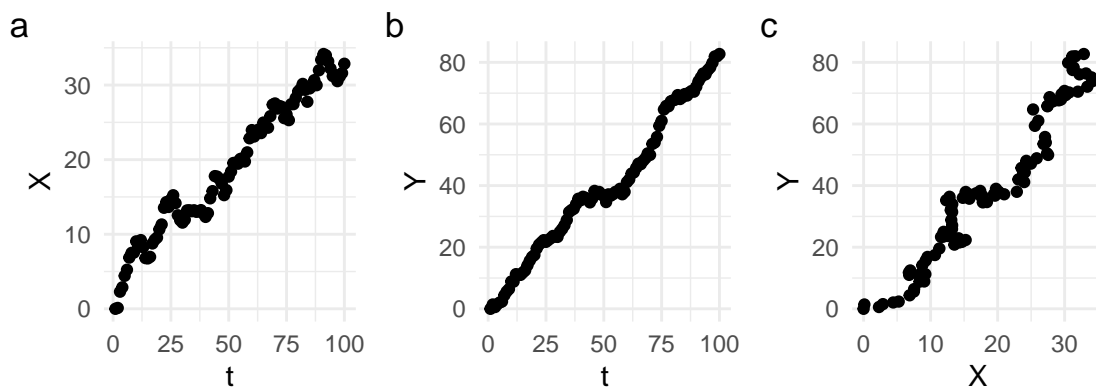


Figure 11.27: Trending Series are Highly Correlated.

The interesting thing about this result is that it persists even if  $\alpha_X$  and  $\alpha_Y$  are both set to zero, i.e., if

$$\begin{aligned} X_t^{(r)} &= X_{t-1}^{(r)} + u_t^{(r)} \\ Y_t^{(r)} &= Y_{t-1}^{(r)} + v_t^{(r)} \end{aligned}$$

so that there are no drifts in the series. Repeating the experiment with this change, we obtain the following t-statistic histogram over 200 replications.

```
set.seed(13);
tstats2 <- simExp(200,100,0,0)
df <- data.frame(tstats2)
ggplot(df, aes(x=tstats2)) +
  geom_histogram(binwidth=2, color="black",fill="grey") + theme_minimal()
```

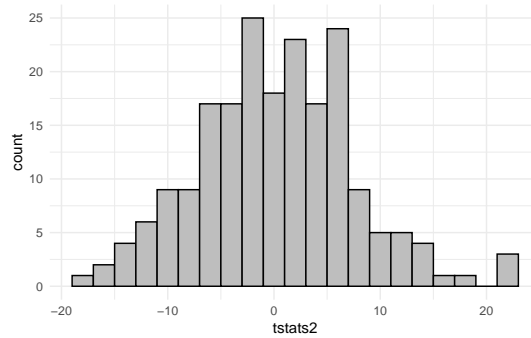


Figure 11.28: Histogram of t-statistics.

We find that in the vast majority (about 80%) of the replications, the t-statistic is greater than two, well in excess of 5%. This result is known as “spurious regressions”, and we leave a fuller discussion of this issue for a later chapter.

## 11.4 Exercises

### Exercise 11.1.

- Show that the first-order Taylor (i.e., linear) approximation to the (natural) log function leads to the approximation

$$\frac{Y_t - Y_{t-1}}{Y_{t-1}} \approx \ln Y_t - \ln Y_{t-1}.$$

- Show that  $\ln Y_t - \ln Y_{t-1}$  can also be viewed as the (exact) continuous growth rate over period  $t$  (assume that measurements of  $Y$  are taken at the end of each period).

**Exercise 11.2.** A popular transformation to apply to time series data is the Box-Cox transformation

$$y(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \ln(y) & \text{if } \lambda = 0 \end{cases}$$

for some given  $\lambda$ . This transformation is only used when  $y > 0$ . Show that

$$\frac{y^\lambda - 1}{\lambda} \rightarrow \ln(y) \quad \text{as } \lambda \rightarrow 0.$$

**Exercise 11.3.** The monthly seasonal dummy specification

$$Y_t = \delta_0 + \delta_2(d_{2,t} - \frac{1}{12}) + \delta_3(d_{3,t} - \frac{1}{12}) + \cdots + \delta_{12}(d_{12,t} - \frac{1}{12}) + \epsilon_t. \quad (11.22)$$

produces an equivalent fit to the specifications (11.4) and (11.5). Find interpretations for the parameters  $\delta_0, \delta_2, \dots, \delta_{12}$ .

**Exercise 11.4.** How would you test for the presence of seasonality using the seasonal dummy models (11.4), (11.5) and (11.22)? State the exact hypotheses to be tested.

**Exercise 11.5.** Suppose the price of a certain product is determined by supply and demand as specified below:

$$\begin{aligned} Q_t^s &= \alpha_0 + \alpha_1 P_t + \alpha_2 P_{t-1} + \epsilon_t^s && \text{(supply equation)} \\ Q_t^d &= \delta_0 + \delta_1 P_t + \epsilon_t^d && \text{(demand equation)} \\ Q_t^s &= Q_t^d && \text{(market clearing)} \end{aligned}$$

where the demand and supply shocks  $\epsilon_t^s$  and  $\epsilon_t^d$  are zero-mean independent noise term. Show that price  $P_t$  follows an AR(1) process by equating the supply and demand equations and solving for  $P_t$  in terms of  $P_{t-1}$  and the demand and supply shocks.

**Exercise 11.6.** Re-parameterize the distributed lag model

$$Y_t = \alpha_0 + \beta_0 X_t + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \epsilon_t$$

so that the long-run cumulative dynamic multiplier  $\beta_0 + \beta_1 + \beta_2$  appears as a coefficient on a regressor. *Remark: Estimating a version where the long-run cumulative dynamic multiplier is a coefficient on a regressor makes it much easier to obtain the standard errors on the sum  $\hat{\beta}_0 + \hat{\beta}_1 + \hat{\beta}_2$ .*

**Exercise 11.7.** Suppose

$$Y_t = \alpha_0 + \alpha_1 Y_{t-1} + u_t$$

with  $|\alpha_1| < 1$  and where  $u_t = \epsilon_t + \theta \epsilon_{t-1}$ ,  $\epsilon_t \sim_{iid} (0, \sigma^2)$ . In this exercise we will show that  $Y_{t-1}$  and  $u_t$  are correlated, so that even contemporaneous exogeneity does not hold.

- Show that  $E[u_t u_{t-1}] = \theta \sigma^2$ .
- Show that  $E[u_t u_{t-j}] = 0$  for all  $j > 1$ .
- Show that  $Y_t$  can be written as

$$Y_t = \frac{\alpha_0}{1 - \alpha_1} + u_t + \alpha_1 u_{t-1} + \alpha_1^2 u_{t-2} + \dots$$

- Show that  $E[u_t Y_{t-1}] = \theta \sigma^2$ .

**Exercise 11.8.** When we fit (11.20) on the  $\log(\text{ELEC\_GEN\_SG})$  series we obtained a very high  $R^2$ . This is often the case when fitting models with trend to models. Much of the explanatory power may be coming from the trend component. Run a regression of  $\log(\text{ELEC\_GEN\_SG})$  on the seasonal dummies and the quadratic trend and collect the residuals. Do the same with  $\log(\text{IP\_SG})$ . Now run a regression of the residuals from the  $\log(\text{ELEC\_GEN\_SG})$  regression on to the residuals from the  $\log(\text{IP\_SG})$  regression. Confirm that the coefficient on the  $\log(\text{IP\_SG})$  residuals are the same as the coefficient on  $\log(\text{IP\_SG})$  in (11.20). What is the  $R^2$  from the residual regression?

**Exercise 11.9.** In the regression (11.21), add one additional lags of  $\log(\text{ELEC\_GEN\_SG})$  and two lags of  $\log(\text{IP\_SG})$ . How would you amend the specification in (11.21)?

## 11.5 Appendix

There are methods that measure trend without requiring a decision on the form of the trend. The Hodrick-Prescott (HP) filter, popular among macroeconomists, is one such method. Denoting the trend as  $\tau_t$ , the method chooses  $\hat{\tau}_t$ ,  $t = 1, 2, \dots, T$  to minimize the sum of squared residuals  $\sum_{t=1}^T (y_t - \hat{\tau}_t)^2$ , while controlling for the overall level of “wigglyness” of the estimated trend. Specifically, the HP method sets:

$$\hat{\tau}_t^{hp} = \operatorname{argmin}_{\hat{\tau}_t} \left( \sum_{t=1}^T (y_t - \hat{\tau}_t)^2 + \lambda \sum_{t=2}^{T-1} [(\hat{\tau}_{t+1} - \hat{\tau}_t) - (\hat{\tau}_t - \hat{\tau}_{t-1})]^2 \right) \quad (11.23)$$

for some given value of  $\lambda > 0$ . The first term in the minimization is the sum of squared residuals. In the second summation,  $\hat{\tau}_t - \hat{\tau}_{t-1}$  is the change in the estimated trend from  $t - 1$  to  $t$ , and  $\hat{\tau}_{t+1} - \hat{\tau}_t$  is the change from period  $t$  to  $t + 1$ . The term

$$\sum_{t=2}^{T-1} [(\hat{\tau}_{t+1} - \hat{\tau}_t) - (\hat{\tau}_t - \hat{\tau}_{t-1})]^2$$

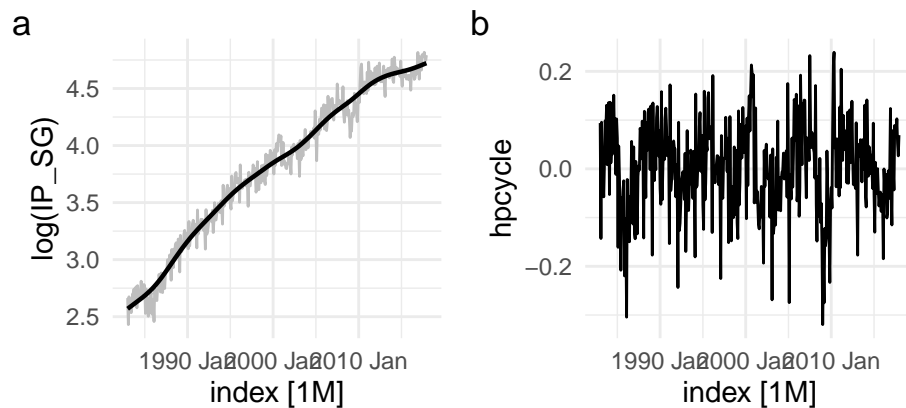
is therefore a measure of how quickly  $\hat{\tau}_t$  is change at  $t = 2, 3, \dots, T - 1$ , analogous to a second derivative. The second derivative measures how quickly a slope is changing – a large absolute second-derivative at a certain point indicates that the slope is changing very quickly at the point, i.e., the function is bending very sharply at that point. In the case of a straight time, the second derivative is zero everywhere.

We can get a sense of the HP filter works by imagining extreme values of  $\lambda$ . Setting  $\lambda = 0$ , the second term becomes irrelevant, and we can minimize (11.23) by setting  $\hat{\tau}_t^{hp} = y_t$  for all  $t$ . In other words, we simply connect the dots. Setting  $\lambda = \infty$ , the slightest bend in the proposed trend results in (11.23) becoming infinity, so minimization is achieved by fitting a straight line through the data.

The following is an application to the log(IP\_SG) with  $\lambda = 129600$ , the value of  $\lambda$  that has been recommended for monthly data. There series in Fig. 11.29(b) is the detrended log(IP\_SG) after the HP trend is removed.

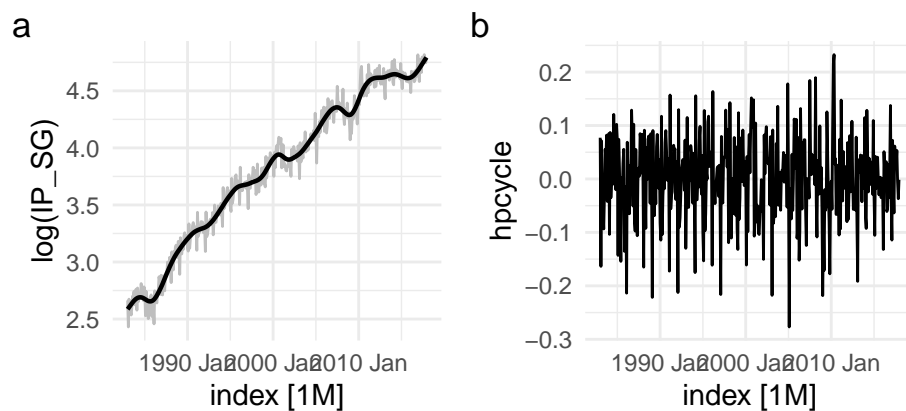
```
ts03.ts <- as.ts(ts03)
lipsg <- mFilter::hpfilter(log(ts03.ts[, 'IP_SG']), type="lambda", freq=129600)
hp_dat <- as_tsibble(ts.union("hpcycle"=lipsg$cycle,
                             "hptrend"=lipsg$trend,
                             "log(IP_SG)"=log(ts03.ts[, 'IP_SG'])), pivot_longer=F)
p1 <- autoplot(hp_dat, `log(IP_SG)` , color="grey") +
  autolayer(hp_dat, hptrend, size=0.8) +
  theme_minimal() + theme(legend.position = "bottom")
p2 <- autoplot(hp_dat, hpcycle) + theme_minimal()
(p1 | p2) + plot_annotation(tag_levels = 'a')
```



Figure 11.29: HP Filter Applied to  $\log(\text{IP\_SG})$ ,  $\lambda = 129600$ .

One difficulty with trend fitting and de-trending is that what remains (cycles, seasonalities, and other features) depends on what is taken out. It is sometimes difficult to tell what is trend (“long-term” movements) and what is cycle (“medium term” movements?). Repeating the exercise above with  $\lambda = 1600$  (the value usually recommended for quarterly data, even though we have monthly data) gives:

```
## uses mFilter package
lipsg <- mFilter::hpfilter(log(ts03.ts[, 'IP_SG']), type="lambda", freq=1600)
hp_dat <- as_tsibble(ts.union("hpcycle"=lipsg$cycle,
                             "hptrend"=lipsg$trend,
                             "log(IP_SG)"=log(ts03.ts[, 'IP_SG']))), pivot_longer=F)
p1 <- autoplot(hp_dat, `log(IP_SG)` , color="grey") +
  autolayer(hp_dat, hptrend, size=0.8) +
  theme_minimal() + theme(legend.position = "bottom")
p2 <- autoplot(hp_dat, hpcycle) + theme_minimal()
(p1 | p2) + plot_annotation(tag_levels = 'a')
```

Figure 11.30: HP Filter Applied to  $\log(\text{IP\_SG})$ ,  $\lambda = 1600$ 

The fitted trend fluctuates substantially more, and one suspects that it has picked up more

than just trend. The fitted line here might be better thought of as “trend and cycle”.

The HP filter is an example of a *non-parametric technique*. Another much more elementary non-parametric approach is to use a “moving-average”:

$$\hat{\tau}_t^{ma} = \frac{1}{2k+1} \sum_{j=-k}^k Y_{t+j}, t = k+1, \dots, T-k$$

for some chosen bandwidth  $k$ . Note that the specific version of the technique shown here does not provide estimates on either end of the series (there are versions that do). The “local” nature of this method means that the results should be considered “trend and cycle” unless a very large  $k$  is chosen, but then that would only provide estimates for a small portion of the data. The following smooths the  $\log(\text{IP\_SG})$  series with  $k = 10$

```
ma_dat <- ts03 %>% mutate("MA"=as.numeric(NA))
k <- 10
T <- dim(ma_dat)[1]
ma_dat[(k+1):(T-k), "MA"] <- zoo::rollmean(log(ma_dat[, "IP_SG"]), 2*k+1, align="center")
autoplot(ma_dat, log(IP_SG), color="darkgray") +
  autolayer(ma_dat, MA, size=0.8) + theme_minimal() +
  theme(legend.position = "bottom")
```

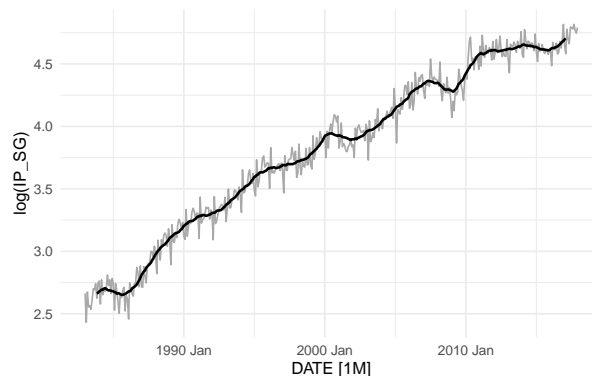


Figure 11.31: Moving Average Filter of  $\log(\text{IP\_SG})$ .

## References

- Angrist, Joshua D., and Alan B. Krueger. 2001. "Instrumental Variables and the Search for Identification: From Supply and Demand to Natural Experiments." *Journal of Economic Perspectives* 15 (4): 69–85. <https://doi.org/10.1257/jep.15.4.69>.
- Auguie, Baptiste. 2015. *gridExtra: Miscellaneous Functions for "Grid" Graphics*. <http://CRAN.R-project.org/package=gridExtra>.
- Fox, John, and Sanford Weisberg. 2019. *An R Companion to Applied Regression*. Third. Thousand Oaks CA: Sage. <https://socialsciences.mcmaster.ca/jfox/Books/Companion/>.
- Hyndman, Rob J, and George Athanasopoulos. 2021. *Forecasting: Principles and Practice*. Third. Melbourne, Australia: OTexts. <https://otexts.com/fpp3/>.
- . 2023. *Fpp3*.
- International Monetary Fund, International Labour Organization, Statistical Office of the European Union (Eurostat), United Nations Economic Commission for Europe, Organisation for Economic Co-operation and Development (OECD), and The World Bank. 2020. *Consumer Price Index Manual: Concepts and Methods 2020*. Washington, D.C.: International Monetary Fund.
- Karline, Soetaert. 2015. *plot3D*. <https://cran.r-project.org/web/packages/plot3D/index.html>.
- Meschiari, Stefano. 2023. *Latex2exp: Use LaTeX Expressions in Plots*.
- Pedersen, Thomas Lin. 2023. *Patchwork: The Composer of Plots*.
- Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Golemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. <https://doi.org/10.21105/joss.01686>.
- Wickham, Hadley, and Jennifer Bryan. 2023. *Readxl: Read Excel Files*.
- Zeileis, Achim, Susanne Köll, and Nathaniel Graham. 2020. "Various Versatile Variances: An Object-Oriented Implementation of Clustered Covariances in R." *J. Stat. Softw.* 95 (1). <https://doi.org/10.18637/jss.v095.i01>.

