

# **Empirical Methods**

Sven Otto

August 22, 2024

# Table of contents

<b>Welcome to the course!</b>	<b>3</b>
Course Materials . . . . .	3
<b>I A) Basic Principles</b>	<b>6</b>
<b>1 Data</b>	<b>7</b>
1.1 Datasets . . . . .	7
1.2 R programming language . . . . .	8
1.3 Datasets in R . . . . .	9
1.4 Importing data . . . . .	12
1.5 Data types . . . . .	13
1.6 Random variables . . . . .	15
1.7 Sampling . . . . .	16
1.8 R-codes . . . . .	18
<b>2 Summary Statistics</b>	<b>19</b>
2.1 Sample moments . . . . .	20
2.2 Empirical distribution . . . . .	23
2.3 Sample quantiles . . . . .	24
2.4 Density estimation . . . . .	26
2.5 Sample covariance . . . . .	28
2.6 R-codes . . . . .	30
<b>II B) Linear Regression</b>	<b>31</b>
<b>3 Least Squares</b>	<b>32</b>
3.1 Regression function . . . . .	32
3.2 Ordinary least squares (OLS) . . . . .	33
3.3 Regression plots . . . . .	33
3.4 Matrix notation . . . . .	36
3.5 R-squared . . . . .	38
3.6 Too many regressors . . . . .	39
3.7 Perfect multicollinearity . . . . .	41
3.8 Dummy variable trap . . . . .	42

3.9	R-codes . . . . .	42
<b>4</b>	<b>The Linear Model</b>	<b>43</b>
4.1	Assumptions . . . . .	44
4.2	OLS Estimator . . . . .	45
4.3	Marginal Effects . . . . .	46
4.4	Control Variables . . . . .	48
4.5	Polynomials . . . . .	49
4.6	Interactions . . . . .	50
4.7	Logarithms . . . . .	51
4.8	R-codes . . . . .	53
<b>5</b>	<b>Regression Inference</b>	<b>54</b>
5.1	Standardized coefficients . . . . .	54
5.2	Standard Errors . . . . .	55
5.2.1	Robust standard errors . . . . .	55
5.2.2	Classical standard errors . . . . .	56
5.2.3	Standard Errors in R . . . . .	57
5.3	Interval estimates . . . . .	57
5.3.1	Asymptotic Intervals . . . . .	57
5.3.2	Exact Intervals . . . . .	58
5.4	t-Tests . . . . .	59
5.5	Joint Testing . . . . .	60
5.5.1	Joint Hypotheses . . . . .	61
5.5.2	Wald Test . . . . .	61
5.5.3	F-Test . . . . .	62
5.5.4	Testing in R . . . . .	62
5.6	R-codes . . . . .	63
<b>6</b>	<b>Case Study I: Score Data</b>	<b>64</b>
6.1	Data Set Description . . . . .	64
6.2	Linear Regression . . . . .	65
6.3	Bad Controls . . . . .	66
6.4	Good Controls . . . . .	68
6.5	Nonlinear Specifications . . . . .	72
6.6	Interactions . . . . .	76
6.6.1	Two Binary Variables . . . . .	76
6.6.2	Continuous and Binary Variables . . . . .	78
6.6.3	Two Continuous Variables . . . . .	80
6.7	Nonlinearities in Score Regressions . . . . .	81
6.8	R-codes . . . . .	85

<b>7</b>	<b>Regression Diagnostics</b>	<b>86</b>
7.1	Leverage values . . . . .	86
7.2	Standardized residuals . . . . .	86
7.3	Diagnostics plots . . . . .	87
	Plot 1: Residuals vs Fitted . . . . .	88
	Plot 2: Normal Q-Q . . . . .	89
	Plot 3: Scale-Location . . . . .	90
	Plot 4: Residuals vs Leverage . . . . .	91
7.4	Diagnostics tests . . . . .	92
	7.4.1 Breusch-Pagan Test (Koenker’s version) . . . . .	93
	7.4.2 Jarque-Bera Test . . . . .	93
7.5	R-codes . . . . .	94
<b>III</b>	<b>C) Panel Data Methods</b>	<b>95</b>
<b>8</b>	<b>Panel Regression</b>	<b>96</b>
8.1	Panel Data . . . . .	96
8.2	Pooled Regression . . . . .	97
8.3	Pooled Regression Assumptions . . . . .	99
8.4	Pooled Regression Inference . . . . .	100
8.5	R-codes . . . . .	101
<b>9</b>	<b>Fixed Effects</b>	<b>102</b>
9.1	Time-constant Variables . . . . .	102
9.2	Fixed Effects Regression . . . . .	103
9.3	Differenced Estimator . . . . .	104
9.4	Within Estimator . . . . .	104
9.5	Time Fixed Effects . . . . .	106
9.6	Two-way Fixed Effects . . . . .	107
9.7	Comparison of panel models . . . . .	109
9.8	Dummy variable regression . . . . .	110
9.9	Panel R-squared . . . . .	113
9.10	R-codes . . . . .	114
<b>10</b>	<b>Case Study II: Drunk Driving</b>	<b>115</b>
10.1	Cross-sectional Regression . . . . .	118
10.2	“Before and After” Comparisons . . . . .	120
10.3	State Fixed Effects . . . . .	123
10.4	Year Fixed Effects . . . . .	124
10.5	Driving Laws and Economic Conditions . . . . .	126
10.6	Summary . . . . .	131
10.7	R-codes . . . . .	131

<b>IV D) Big Data Econometrics</b>	<b>132</b>
<b>11 Shrinkage Estimation</b>	<b>133</b>
11.1 Mean squared error . . . . .	133
11.2 A simple shrinkage estimator . . . . .	134
11.3 High-dimensional regression . . . . .	136
11.4 Ridge Regression . . . . .	136
11.5 Standardization . . . . .	137
11.6 Ridge Properties . . . . .	137
11.7 Mean squared prediction error . . . . .	138
11.8 Cross validation . . . . .	139
11.9 L2 Regularization: Ridge . . . . .	140
11.10 L1 Regularization: Lasso . . . . .	140
11.11 Implementation in R . . . . .	141
11.12 R-codes . . . . .	146
<b>12 Principal Component Regression</b>	<b>147</b>
12.1 Principal Components . . . . .	147
12.2 Analytical PCA Solution . . . . .	148
12.3 Sample principal components . . . . .	150
12.4 PCA in R . . . . .	150
12.5 Variance of principal components . . . . .	153
12.6 Linear regression with principal components . . . . .	154
12.7 Selecting the number of factors . . . . .	155
12.8 R-codes . . . . .	157
<b>13 Case Study III: Big Data</b>	<b>158</b>
13.1 Introduction . . . . .	158
13.2 Data Set Description . . . . .	158
13.3 Methods . . . . .	159
13.4 Data Preparation . . . . .	160
13.5 Cross-Validation for Tuning Parameters . . . . .	160
13.5.1 Ridge Regression . . . . .	160
13.5.2 Lasso Regression . . . . .	162
13.5.3 Principal Components Regression . . . . .	164
13.6 Summary . . . . .	167
13.7 R-codes . . . . .	169
<b>V E) Time Series Methods</b>	<b>170</b>
<b>14 Forecasting Models</b>	<b>171</b>
14.1 Basic time series models . . . . .	171

14.2	Dynamic regressions . . . . .	172
14.3	One-step ahead forecast . . . . .	172
14.4	Dynamic models in R . . . . .	173
14.4.1	An AR model for GDP . . . . .	173
14.4.2	An ADL model for gasoline prices . . . . .	178
14.5	Identification . . . . .	181
14.6	AR(1) process . . . . .	182
14.7	Autocorrelations of GDP . . . . .	184
14.8	R-codes . . . . .	184
<b>15</b>	<b>Time Series Inference</b>	<b>186</b>
15.1	Assumptions for time series regression . . . . .	186
15.2	Time series standard errors . . . . .	187
15.3	Spurious correlation . . . . .	189
15.3.1	Simulation evidence . . . . .	189
15.3.2	Real-world spurious correlations . . . . .	192
15.4	Testing for stationarity . . . . .	193
15.4.1	Dickey Fuller test . . . . .	193
15.4.2	Augmented Dickey Fuller test . . . . .	195
15.5	R-codes . . . . .	197
	<b>Appendices</b>	<b>198</b>
<b>A</b>	<b>OLS: Technical Details</b>	<b>198</b>
A.1	Probability toolbox . . . . .	198
A.2	Conditional Expectation . . . . .	200
A.3	Weak exogeneity . . . . .	200
A.4	Strict exogeneity . . . . .	200
A.5	Heteroskedasticity . . . . .	201
A.6	No autocorrelation . . . . .	201
A.7	Existence . . . . .	201
A.8	Unbiasedness . . . . .	202
A.9	Conditional variance . . . . .	202
A.10	Consistency . . . . .	202
A.11	Asymptotic normality . . . . .	203

# Welcome to the course!

Empirical Methods is a graduate-level course in regression analysis focusing on specialized econometric tools. We cover advanced topics such as panel data methods, generalized linear models, high-dimensional regression, instrumental variables, causal inference, and time series regression. Emphasis is on both theoretical understanding of the methods and practical applications using the R programming language.

## Course Materials

- [This webpage](#) and [its pdf version](#): the online script
- [eWhiteboard](#): the whiteboard notes
- [ILIAS](#): further course material
- [RScripts](#): codes from the lecture

## Literature

The course is primarily based on the following textbook:

- Stock, J.H. and Watson, M.W. (2019). **Introduction to Econometrics (Fourth Edition)**. Pearson.

The Global Edition of Stock and Watson (2019) is available [here](#). To view the book, please activate your Uni Köln VPN connection.

### Recommended reading to accompany the lecture:

Part	Reading
A – Basic Principles	Stock and Watson: Sections 1–3
B – Linear Regression	Stock and Watson: Sections 4–9 and 18–19
C – Panel Data Methods	Stock and Watson: Section 10
D – Big Data Econometrics	Stock and Watson: Section 14
E – Time Series Methods	Stock and Watson: Section 15

For specialized topics beyond Stock and Watson (2019), the following textbooks are recommended:

- James, G., Witten, D., Hastie, T., and Tibshirani, R. (2019). **An Introduction to Statistical Learning with Applications to R (Second Edition)**. Springer.
- Davidson, R., and MacKinnon, J.G. (2004). **Econometric Theory and Methods**. Oxford University Press.

James et al. (2019) is available for free [here](#) and [here](#). Davidson and MacKinnon (2004) is available for free on the author's webpage: [LINK](#). Printed versions of the books are available from the university library.

## Preparation

You should also be familiar with the basic concepts of **matrix algebra** and **probability theory**. Please consider the following refreshers:

[Crash Course in Matrix Algebra](#)

[Probability Theory for Econometricians](#)

We will be using the statistical programming language R. Please make sure you have **R** and **RStudio** installed before the class. [Here](#) you find the installation instructions for the software. If you are a beginner, please consider this short introduction, which contains many valuable resources:

[Getting Started with R](#)

## Assessment

The course will be graded by a 90-minute exam. More information about the assessment can be found on [ILIAS](#).

## Communication

Feel free to use the [ILIAS methods forum](#) to discuss lecture topics and ask questions. Please also let me know if you find any typos. Of course, you can reach me via e-mail: [sven.otto@uni-koeln.de](mailto:sven.otto@uni-koeln.de)



## Important Dates

Registration deadline exam 1	Jul 04, 2024
Exam 1	Jul 18, 2024, 16:00-17:30
Registration deadline exam 2	Aug 13, 2024
Exam 2 (alternate date)	Aug 20, 2024, 11:30-13:00

Please register for the exam on time. If you miss the registration deadline, you will not be able to take the exam (the Examinations Office is very strict about this). You only need to take one of the two exams to complete the course. The second exam will serve as a make-up exam for those who fail the first exam or do not take the first exam.

## Timetable

The course is held on Tuesdays from 14:00 to 15:30 in **Hörsaal XXI** and on Thursdays every two weeks from 16:00 to 17:30 in **Hörsaal VI**: [KLIPS TIMETABLE](#).

## R-Packages

To run the R code of the lecture script, you will need to install some additional packages.

```
install.packages(  
  c("AER", "plm", "dynlm", "glmnet", "moments", "urca",  
    "tidyverse", "stargazer", "BVAR",  
    "palmerpenguins", "kableExtra", "scatterplot3d"))
```

Some further datasets are contained in my package `teachingdata`, which is available in a GitHub repository:

```
install.packages("remotes")  
remotes::install_github("ottosven/teachingdata")
```

See the Ilias course on how to install `teachingdata2`.

## **Part I**

### **A) Basic Principles**

# 1 Data

## 1.1 Datasets

A **univariate dataset** is a sequence of observations  $Y_1, \dots, Y_n$ . These  $n$  observations can be organized into the **data vector**  $\mathbf{Y}$ , represented as  $\mathbf{Y} = (Y_1, \dots, Y_n)'$ . For example, if you conduct a survey and ask five individuals about their hourly earnings, your data vector might look like

$$\mathbf{Y} = \begin{pmatrix} 18.22 \\ 23.85 \\ 10.00 \\ 6.39 \\ 7.42 \end{pmatrix}.$$

Typically we have data on more than one variable, such as years of education and the gender. Categorical variables are often encoded as **dummy variables**, which are binary variables. The female dummy variable is defined as 1 if the gender of the person is female and 0 otherwise.

person	wage	education	female
1	18.22	16	1
2	23.85	18	0
3	10.00	16	1
4	6.39	13	0
5	7.42	14	0

A  **$k$ -variate dataset** is a collection of  $n$  vectors  $\mathbf{X}_1, \dots, \mathbf{X}_n$  containing data on  $k$  variables. The  $i$ -th vector  $\mathbf{X}_i = (X_{i1}, \dots, X_{ik})'$  contains the data on all  $k$  variables for individual  $i$ . Thus,  $X_{ij}$  represents the value for the  $j$ -th variable of individual  $i$ .

The full  $k$ -variate dataset is structured in the  $n \times k$  **data matrix**  $\mathbf{X}$ :

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}'_1 \\ \vdots \\ \mathbf{X}'_n \end{pmatrix} = \begin{pmatrix} X_{11} & \dots & X_{1k} \\ \vdots & \ddots & \vdots \\ X_{n1} & \dots & X_{nk} \end{pmatrix}$$


The  $i$ -th row in  $\mathbf{X}$  corresponds to the values from  $\mathbf{X}_i$ . Since  $\mathbf{X}_i$  is a column vector, we use the transpose notation  $\mathbf{X}'_i$ , which is a row vector. The data matrix and vectors for our example

are:

$$\mathbf{X} = \begin{pmatrix} 18.22 & 16 & 1 \\ 23.85 & 18 & 0 \\ 10.00 & 16 & 1 \\ 6.39 & 13 & 0 \\ 7.42 & 14 & 0 \end{pmatrix}, \quad \mathbf{X}_1 = \begin{pmatrix} 18.22 \\ 16 \\ 1 \end{pmatrix}, \quad \mathbf{X}_2 = \begin{pmatrix} 23.85 \\ 18 \\ 0 \end{pmatrix}, \dots$$

Vector and matrix algebra provide a compact mathematical representation of multivariate data and an efficient framework for analyzing and implementing statistical methods. We will use matrix algebra frequently throughout this course.

To refresh or enhance your knowledge of matrix algebra, please consult the following resources:

 **Crash Course on Matrix Algebra:**

[matrix.svenotto.com](http://matrix.svenotto.com)

Section 19.1 of the Stock and Watson book also provides a brief overview of matrix algebra concepts.

## 1.2 R programming language

The best way to learn statistical methods is to program and apply them yourself. Throughout this course, we will use the R programming language for implementing empirical methods and analyzing real-world datasets.

If you are just starting with R, it is crucial to familiarize yourself with its basics. Here's an introductory tutorial, which contains a lot of valuable resources:

 **Getting Started with R:**

[rintro.svenotto.com](http://rintro.svenotto.com)

For those new to R, I also recommend the interactive R package [SWIRL](#), which offers an excellent way to learn directly within the R environment. Additionally, two highly recommended online books are [Hands-On Programming with R](#) (with focus on programming) and [R for Data Science](#) (with focus on data analysis).

One of the best features of R is its extensive ecosystem of packages contributed by the statistical community. You find R packages for almost any statistical method out there and many statisticians provide R packages to accompany their research.

Maybe the most frequently used package is the `tidyverse` package, which provides a comprehensive suite of data management and visualization tools. You can install the package with the command `install.packages("tidyverse")` and you can load it with

```
library(tidyverse)
```

at the beginning of your code. We will explore several additional packages in the course of the lecture.

## 1.3 Datasets in R

R includes many built-in datasets and packages of datasets that can be loaded directly into your R environment. For illustration, we consider the `penguins` dataset available in the `palmerpenguins` package. To load this dataset into your R session, simply use:

```
data(penguins, package = "palmerpenguins")
```

```
class(penguins)
```

```
[1] "tbl_df"      "tbl"        "data.frame"
```

The `penguins` dataset is stored as a `data.frame`, R's most common data storage class for tabular data as in **X**. It organizes data in the form of a table, with variables as columns and observations as rows. The `penguins` object is also identified as a `tibble` (or `tbl_df`), the tidyverse version of a `data.frame`.

To inspect the structure of your dataset, you can use `str()` or `glimpse()`:

```
str(penguins)
```

```
tibble [344 x 8] (S3: tbl_df/tbl/data.frame)
 $ species      : Factor w/ 3 levels "Adelie","Chinstrap",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ island       : Factor w/ 3 levels "Biscoe","Dream",...: 3 3 3 3 3 3 3 3 3 3 ...
 $ bill_length_mm : num [1:344] 39.1 39.5 40.3 NA 36.7 39.3 38.9 39.2 34.1 42 ...
 $ bill_depth_mm : num [1:344] 18.7 17.4 18 NA 19.3 20.6 17.8 19.6 18.1 20.2 ...
 $ flipper_length_mm: int [1:344] 181 186 195 NA 193 190 181 195 193 190 ...
 $ body_mass_g   : int [1:344] 3750 3800 3250 NA 3450 3650 3625 4675 3475 4250 ...
 $ sex          : Factor w/ 2 levels "female","male": 2 1 1 NA 1 2 1 2 NA NA ...
 $ year         : int [1:344] 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
```

The dataset contains variables of various types: `fct(factor)` for categorical data, `dbl(numeric)` for real or continuous data, and `int(integer)` for integer or discrete data. The `head()` function displays its first few rows:

```
head(penguins)
```

```
# A tibble: 6 x 8
  species island  bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
  <fct>   <fct>          <dbl>         <dbl>          <int>        <int>
1 Adelie  Torgersen        39.1           18.7            181          3750
2 Adelie  Torgersen        39.5           17.4            186          3800
3 Adelie  Torgersen        40.3           18              195          3250
4 Adelie  Torgersen        NA             NA              NA           NA
5 Adelie  Torgersen        36.7           19.3            193          3450
6 Adelie  Torgersen        39.3           20.6            190          3650
# i 2 more variables: sex <fct>, year <int>
```

The pipe operator `|>` efficiently chains commands. It passes the output of one function as the input to another. For example:

```
penguins |> select(body_mass_g, bill_length_mm, species) |> summary()
```

```
  body_mass_g  bill_length_mm  species
Min.   :2700   Min.   :32.10   Adelie   :152
1st Qu.:3550   1st Qu.:39.23   Chinstrap: 68
Median :4050   Median :44.45   Gentoo   :124
Mean   :4202   Mean   :43.92
3rd Qu.:4750   3rd Qu.:48.50
Max.   :6300   Max.   :59.60
NA's   :2      NA's   :2
```

The `summary()` function presents a concise overview, showing absolute frequencies for categorical variables and descriptive statistics for numerical variables, along with information on missing values (NA). To exclude all rows with missing values, we can use `na.omit(penguins)`.

A dummy variable for the penguin species Gentoo can be created with the following command:

```
gentoo = ifelse(penguins$species == "Gentoo", 1, 0)
```

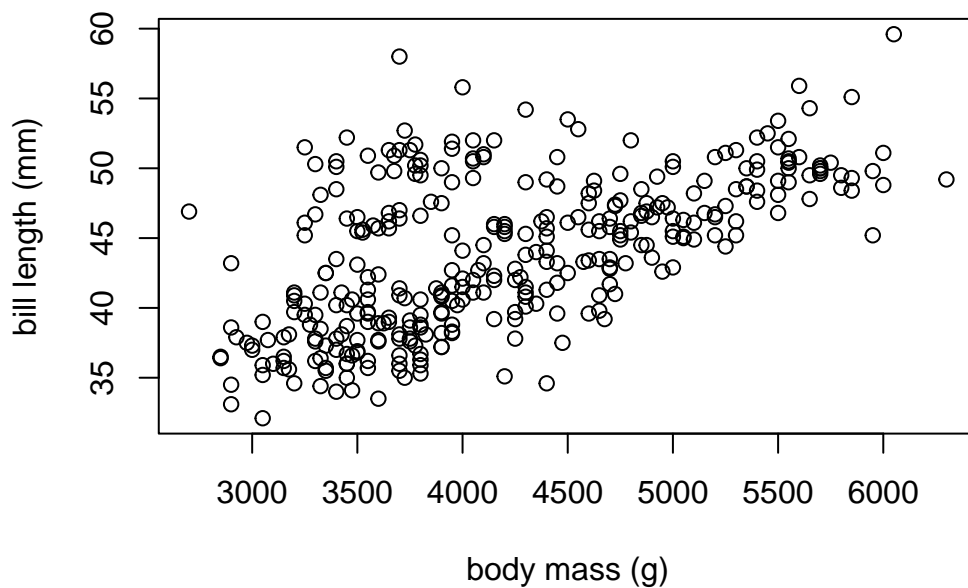
The `$` sign accesses a specific column of a data frame by name, as in `penguins$species` to select the variable `species` from `penguins`.

To convert factor variables into dummy variables efficiently, the `fastDummies` package's `dummy_cols()` function can be used. Let's create dummy variables for each of the three species.

```
library(fastDummies)
penguins.new = dummy_cols(penguins,select_columns = "species")
```

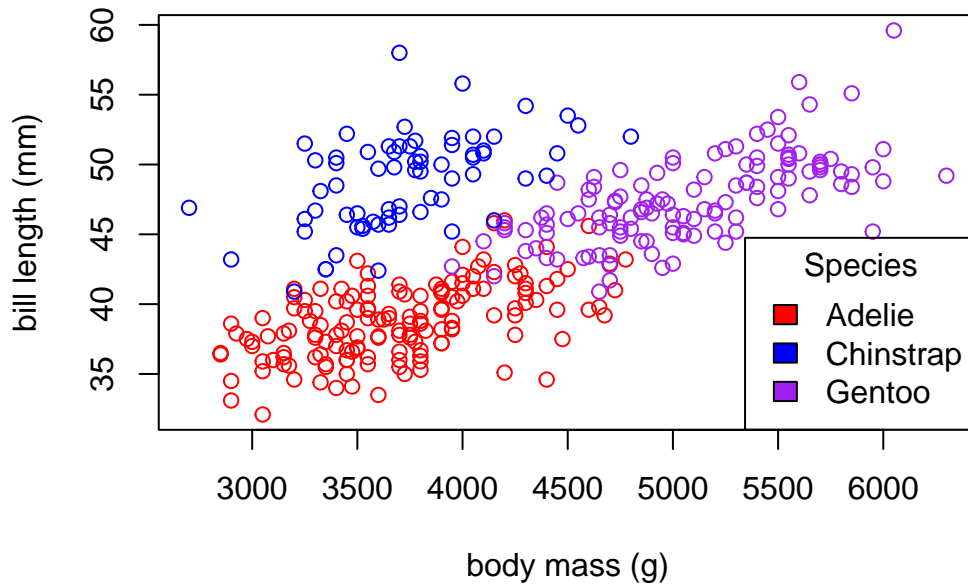
Scatterplots provide further insights:

```
plot(bill_length_mm ~ body_mass_g, data = penguins,
     xlab = "body mass (g)", ylab = "bill length (mm)")
```



We can assign unique colors to each species:

```
colors = c("red", "blue", "purple")
plot(bill_length_mm ~ body_mass_g, col = colors[species], data = penguins,
     xlab = "body mass (g)", ylab = "bill length (mm)")
legend("bottomright", legend = levels(penguins$species),
     fill = colors, title = "Species")
```



## 1.4 Importing data

The internet serves as a vast repository for data in various formats, with `csv` (comma-separated values), `xlsx` (Microsoft Excel spreadsheets), and `txt` (text files) being the most commonly used.

Many organizations, such as the German Bundesbank, the German Federal Statistical Office, the ECB (European Central Bank), Eurostat, and FRED (Federal Reserve Economic Data), offer economic datasets in these formats. These datasets can be accessed through their websites or via Application Programming Interfaces (APIs), which allow direct downloading of data into R. Accessing data via APIs often requires registering for an API token on the organization's website.

R supports various functions for different data formats:

- `read.csv()` for reading comma-separated values
- `read.csv2()` for semicolon-separated values (adopting the German data convention of using `'` as the decimal mark)
- `read.table()` for whitespace-separated files
- `read_excel()` for Microsoft Excel files (requires the `readxl` package)
- `read_stata()` for STATA files (requires the `haven` package)

The `rvest` package provides web scraping tools to extract data directly from HTML web pages.

In academic writing, it is crucial to provide enough information about data sources to ensure transparency and reproducibility.



Let's explore the CPS dataset from Bruce Hansen's website. The Current Population Survey (CPS) is a monthly survey conducted by the U.S. Census Bureau for the Bureau of Labor Statistics, primarily used to measure the labor force status of the U.S. population.

- Dataset: [cps09mar.txt](#)
- Description: [cps09mar\\_description.pdf](#)

```
cps = read.table("https://users.ssc.wisc.edu/~bhansen/econometrics/cps09mar.txt",
  col.names=c("age","female","hispanic","education","earnings","hours",
    "week", "union","uncov","region","race","marital")) |>
mutate(race = as.factor(race),
  region = as.factor(region),
  marital = as.factor(marital),
  experience = (age - education - 6), #years since graduation
  wage = earnings/(week*hours), #wage per hours
  married = ifelse(marital %in% c(1,2), 1, 0), #dummy
  college = ifelse(education >= 14, 1, 0),
  black = ifelse(race %in% c(2,6,10,11,12,15,16,19), 1, 0),
  asian = ifelse(race %in% c(4,8,11,13,14,16,17,18,19), 1, 0))
```

## 1.5 Data types

The most common types of economic data are:

- **Cross-sectional data:** Data collected on many entities without regard to time.
- **Time series data:** Data on a single entity collected over multiple time periods.
- **Panel data:** Data collected on multiple entities over multiple time points, combining features of both cross-sectional and time series data.

The cps data is an example of a cross-sectional dataset.

```
str(cps)
```

```
'data.frame':  50742 obs. of  18 variables:
 $ age      : int  52 38 38 41 42 66 51 49 33 52 ...
 $ female   : int  0 0 0 1 0 1 0 1 0 1 ...
 $ hispanic : int  0 0 0 0 0 0 0 0 0 0 ...
 $ education : int  12 18 14 13 13 13 16 16 16 14 ...
 $ earnings : int  146000 50000 32000 47000 161525 33000 37000 37000 80000 32000 ...
 $ hours    : int  45 45 40 40 50 40 44 44 40 40 ...
```

```

$ week      : int  52 52 51 52 52 52 52 52 52 52 ...
$ union     : int   0 0 0 0 1 0 0 0 0 0 ...
$ uncov     : int   0 0 0 0 0 0 0 0 0 0 ...
$ region    : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
$ race      : Factor w/ 20 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 1 1 ...
$ marital   : Factor w/ 7 levels "1","2","3","4",...: 1 1 1 1 1 5 1 1 1 1 ...
$ experience: num   34 14 18 22 23 47 29 27 11 32 ...
$ wage      : num  62.4 21.4 15.7 22.6 62.1 ...
$ married   : num   1 1 1 1 1 0 1 1 1 1 ...
$ college   : num   0 1 1 0 0 0 1 1 1 1 ...
$ black     : num   0 0 0 0 0 0 0 0 0 0 ...
$ asian     : num   0 0 0 0 0 0 0 0 0 0 ...

```

My repository [teachingdata](#) contains some recent time series datasets, for instance, the nominal GDP growth of Germany:

```

data("gdpgr", package="teachingdata")
str(gdpgr)

```

```

Time-Series [1:128] from 1992 to 2024: 0.0874 0.0651 0.0733 0.0586 0.0201 ...

```

The dataset `Fatalities` is a panel dataset. It contains variables related to traffic fatalities across different states and years in the United States:

```

data(Fatalities, package = "AER")
str(Fatalities)

```

```

'data.frame':  336 obs. of  34 variables:
 $ state      : Factor w/ 48 levels "al","az","ar",...: 1 1 1 1 1 1 1 2 2 2 ...
 $ year       : Factor w/ 7 levels "1982","1983",...: 1 2 3 4 5 6 7 1 2 3 ...
 $ spirits    : num   1.37 1.36 1.32 1.28 1.23 ...
 $ unemp      : num   14.4 13.7 11.1  8.9  9.8 ...
 $ income     : num  10544 10733 11109 11333 11662 ...
 $ emppop    : num   50.7 52.1 54.2 55.3 56.5 ...
 $ beertax    : num   1.54 1.79 1.71 1.65 1.61 ...
 $ baptist    : num   30.4 30.3 30.3 30.3 30.3 ...
 $ mormon     : num   0.328 0.343 0.359 0.376 0.393 ...
 $ drinkage   : num   19 19 19 19.7 21 ...
 $ dry        : num   25 23 24 23.6 23.5 ...
 $ youngdrivers: num   0.212 0.211 0.211 0.211 0.213 ...
 $ miles      : num   7234 7836 8263 8727 8953 ...

```

```

$ breath      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 1 ...
$ jail        : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 2 2 2 ...
$ service     : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 2 2 2 ...
$ fatal       : int   839 930 932 882 1081 1110 1023 724 675 869 ...
$ nfatal      : int   146 154 165 146 172 181 139 131 112 149 ...
$ sfatal      : int   99 98 94 98 119 114 89 76 60 81 ...
$ fatal1517   : int   53 71 49 66 82 94 66 40 40 51 ...
$ nfatal1517  : int    9 8 7 9 10 11 8 7 7 8 ...
$ fatal1820   : int   99 108 103 100 120 127 105 81 83 118 ...
$ nfatal1820  : int   34 26 25 23 23 31 24 16 19 34 ...
$ fatal2124   : int   120 124 118 114 119 138 123 96 80 123 ...
$ nfatal2124  : int   32 35 34 45 29 30 25 36 17 33 ...
$ afatal      : num   309 342 305 277 361 ...
$ pop         : num  3942002 3960008 3988992 4021008 4049994 ...
$ pop1517    : num  209000 202000 197000 195000 204000 ...
$ pop1820    : num  221553 219125 216724 214349 212000 ...
$ pop2124    : num  290000 290000 288000 284000 263000 ...
$ milestot   : num  28516 31032 32961 35091 36259 ...
$ unempus    : num   9.7 9.6 7.5 7.2 7 ...
$ emppopus   : num  57.8 57.9 59.5 60.1 60.7 ...
$ gsp        : num  -0.0221 0.0466 0.0628 0.0275 0.0321 ...

```

## 1.6 Random variables

Data is usually the result of a random experiment. The gender of the next person you meet, the daily fluctuation of a stock price, the monthly music streams of your favourite artist, the annual number of pizzas consumed - all of this information involves a certain amount of randomness.

In statistical sciences, we interpret a univariate dataset  $Y_1, \dots, Y_n$  as a sequence of random variables. Similarly, a multivariate dataset  $\mathbf{X}_1, \dots, \mathbf{X}_n$  is viewed as a sequence of random vectors.

Cross-sectional data is typically characterized by an **identical distribution** across its individual observations, meaning each element in the sequence  $\mathbf{X}_1, \dots, \mathbf{X}_n$  has the same distribution function.

For example, if  $Y_1, \dots, Y_n$  represent the wage levels of different individuals in Germany, each  $Y_i$  is drawn from the same distribution  $F$ , which in this context is the wage distribution across the country. Similarly, if  $\mathbf{X}_1, \dots, \mathbf{X}_n$  are bivariate random variables containing wages and years of education for individuals, each  $\mathbf{X}_i$  follows the same bivariate distribution  $G$ , which is the joint distribution of wages and education levels.

A primary goal of econometric methods and statistical inference is to gain insights about features of these true but unknown population distributions  $F$  or  $G$  using the available data. Econometric methods require certain assumptions about the sampling process and the underlying population distributions. Thus, a solid knowledge of probability theory is essential for econometric modelling.

For a recap on probability theory for econometricians, consider the following refresher:

💡 **Probability Theory for Econometricians:**

<https://probability.svenotto.com/>

Section 2 of the Stock and Watson book also provides a review of the most important concepts.

## 1.7 Sampling

The ideal scenario for data collection involves simple random sampling, where each individual in the population has an equal chance of being selected (independently and identically distributed).

### **i.i.d. sample**

An independently and identically distributed (i.i.d.) sample, or random sample, consists of a sequence of  $k$ -variate random vectors  $\mathbf{X}_1, \dots, \mathbf{X}_n$  that have the same probability distribution  $F$  and are mutually independent, i.e., for any  $i \neq j$  and for all  $\mathbf{a}, \mathbf{b} \in \mathbb{R}^k$ ,

$$P(\mathbf{X}_i \leq \mathbf{a}, \mathbf{X}_j \leq \mathbf{b}) = P(\mathbf{X}_i \leq \mathbf{a})P(\mathbf{X}_j \leq \mathbf{b}).$$

$F$  is called *population distribution* or *data-generating process (DGP)*.

The Current Population Survey (CPS) involves random interviews with individuals from the U.S. labor force may be regarded as an i.i.d. sample. Methods like survey sampling, administrative records, direct observation, web scraping, and field/laboratory experiments can yield i.i.d. sampling for economic cross-sectional datasets. In a random sample there is no inherent ordering that would introduce systematic dependencies.

Note that not all cross-sectional data comes from random sampling. For example, clustered sampling occurs when only specific groups (e.g., classrooms) are chosen randomly (students from the same classroom share the same environment and teacher's performance).

Time series and panel data are intrinsically not independent due to the sequential nature of the observations. We usually expect observations close in time to be strongly dependent and observations at greater distances to be less dependent.

For time series data we assume that there exists some underlying stochastic process represented as a doubly infinite sequence of random variables

$$\{Y_t\}_{t \in \mathbb{Z}} = \{\dots, Y_{-1}, Y_0, \underbrace{Y_1, \dots, Y_n}_{\text{observed part}}, Y_{n+1}, \dots\},$$

where the time series sample  $\{Y_1, \dots, Y_n\}$  is only the observed part of the process.

In order to learn from the observed part about the future (forecasting) or make inference on the dependence with other variables, we typically assume that the distribution of the time series sample does not depend on which time periods are observed, which excludes structural breaks or stochastic trends.

### Stationary time series

A time series process  $\{Y_i\}_{i \in \mathbb{Z}}$  is called **stationary** if the mean  $\mu$  and the autocovariances  $\gamma(\tau)$  do not depend on the time point  $i$ . That is,

$$\mu := E[Y_i] < \infty, \quad \text{for all } i,$$

and

$$\gamma(\tau) := \text{Cov}(Y_i, Y_{i-\tau}) < \infty \quad \text{for all } i \text{ and } \tau.$$

The quarterly nominal GDP is clearly nonstationary. It exhibits trending behavior and seasonalities. The annual nominal GDP growth rates can be regarded as a stationary time series.

Macroeconomic time series often indicate trending behavior and/or seasonalities. However, we can often use simple transformations to convert nonstationary time series into stationary series, such as differences (`diff(your_series, your_frequency)`) or growth rates (`diff(log(your_series), your_frequency)`).

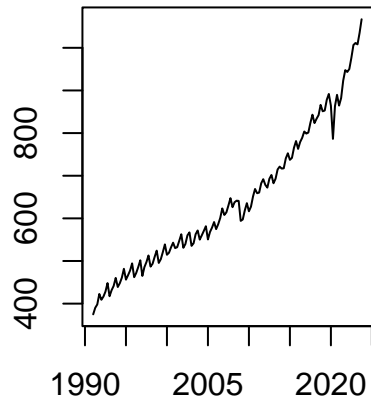
The frequency is the number of observed periods per time basis. Time series (`ts`) objects in R are defined in terms of a yearly time basis. Yearly time series have frequency 1, quarterly have frequency 4, and monthly have frequency 12.

Here are some common transformations:

- First differences:  $\Delta Y_i = Y_i - Y_{i-1}$
- Growth rates:  $\log(Y_i) - \log(Y_{i-1})$

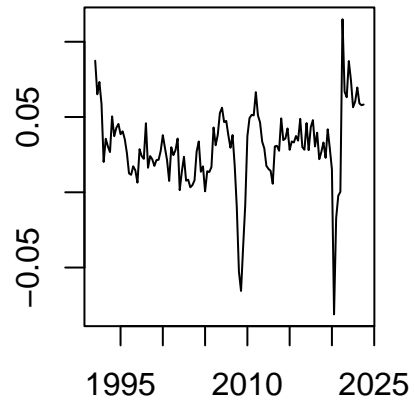
For seasonal data with frequency 4:

**Nominal GDP Germany**



Time

**Annual GDP growth Germany**



Time

- Fourth differences:  $\Delta Y_i = Y_i - Y_{i-4}$
- Annual growth rates:  $\log(Y_i) - \log(Y_{i-4})$

## 1.8 R-codes

[methods-sec01.R](#)

## 2 Summary Statistics

In this section you find an overview of the most important summary statistics commands. In the table below, `your_data` represents some univariate data (vector), and `your_df` represents a `data.frame` of multivariate data (matrix).

Statistic	Command
Sample Size (n)	<code>length(your_data)</code>
Maximum Value	<code>max(your_data)</code>
Minimum Value	<code>min(your_data)</code>
Total Sum	<code>sum(your_data)</code>
Mean	<code>mean(your_data)</code>
Variance	<code>var(your_data)</code>
Standard Deviation	<code>sd(your_data)</code>
Skewness	<code>skewness(your_data)</code> (requires <code>moments</code> package)
Kurtosis	<code>kurtosis(your_data)</code> (requires <code>moments</code> package)
Order statistics	<code>sort(your_data)</code>
Empirical CDF	<code>ecdf(your_data)</code>
Median	<code>median(your_data)</code>
p-Quantile	<code>quantile(your_data, p)</code>
Boxplot	<code>boxplot(your_data)</code>
Histogram	<code>hist(your_data)</code>
Kernel density estimator	<code>plot(density(your_data))</code>
Covariance	<code>cov(your_data1, your_data2)</code>
Correlation	<code>cor(your_data1, your_data2)</code>
Mean vector	<code>colMeans(your_df)</code>
Covariance matrix	<code>cov(your_df)</code>
Correlation matrix	<code>cor(your_df)</code>

*Note:* Ensure that your data does not contain missing values (NA's) for these commands. Use `na.omit()` or include `na.rm=TRUE` in functions to handle missing data.

## 2.1 Sample moments

### Mean

The sample mean (arithmetic mean) is the most common measure of central tendency:

$$\bar{Y} = \frac{1}{n} \sum_{i=1}^n Y_i$$

In i.i.d. samples, it converges in probability to the expected value as sample size grows (law of large numbers). I.e., it is a consistent estimator for the population mean:

$$\bar{Y} \xrightarrow{p} E[Y] \quad \text{as } n \rightarrow \infty.$$

The law of large numbers also holds for stationary time series with  $\gamma(\tau) \rightarrow 0$  as  $\tau \rightarrow \infty$ .

### Variance

The variance measures the spread around the mean. The sample variance is

$$\hat{\sigma}_Y^2 = \frac{1}{n} \sum_{i=1}^n (Y_i - \bar{Y})^2 = \overline{Y^2} - \bar{Y}^2,$$

and the adjusted sample variance is

$$s_Y^2 = \frac{1}{n-1} \sum_{i=1}^n (Y_i - \bar{Y})^2.$$

Note that `var(your_data)` computes  $s_Y^2$ , which is the conventional estimator for the population variance

$$\text{Var}[Y] = E[(Y - E[Y])^2] = E[Y^2] - E[Y]^2.$$

$s_Y^2$  is unbiased whereas  $\hat{\sigma}_Y^2$  is biased but has a lower sampling variance. For i.i.d. samples, both versions are consistent estimators for the population variance.

### Standard deviation

The standard deviation, the square root of the variance, is a measure of dispersion in the original unit of data. It quantifies the average distance data points typically deviate from the mean of their distribution.



The sample standard deviation and its adjusted version are the square roots of the corresponding variance formulas:

$$\hat{\sigma}_Y = \sqrt{\overline{Y^2} - \bar{Y}^2}, \quad s_Y = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (Y_i - \bar{Y})^2}$$

Note that `sd(your_data)` computes  $s_Y$  and not  $\hat{\sigma}_Y$ . Both versions are consistent estimators for the population standard deviation  $sd(Y) = \sqrt{Var[Y]}$  for i.i.d. samples.

## Skewness

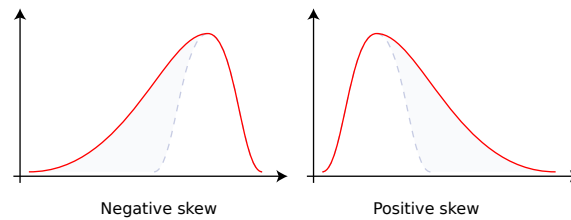
The skewness is a measure of asymmetry around the mean. The sample skewness is

$$\widehat{skew} = \frac{1}{n\hat{\sigma}_Y^3} \sum_{i=1}^n (Y_i - \bar{Y})^3.$$

It is a consistent estimator for the population skewness

$$skew = \frac{E[(Y - E[Y])^3]}{sd(Y)^3}.$$

A non-zero skewness indicates an asymmetric distribution, with positive values indicating a right tail and negative values a left tail. Below you find an illustration using density functions:



## Kurtosis

Kurtosis measures the heaviness of the tails of a distribution. It indicates how likely extreme outliers are. The sample kurtosis is

$$\widehat{kurt} = \frac{1}{n\hat{\sigma}_Y^4} \sum_{i=1}^n (Y_i - \bar{Y})^4.$$

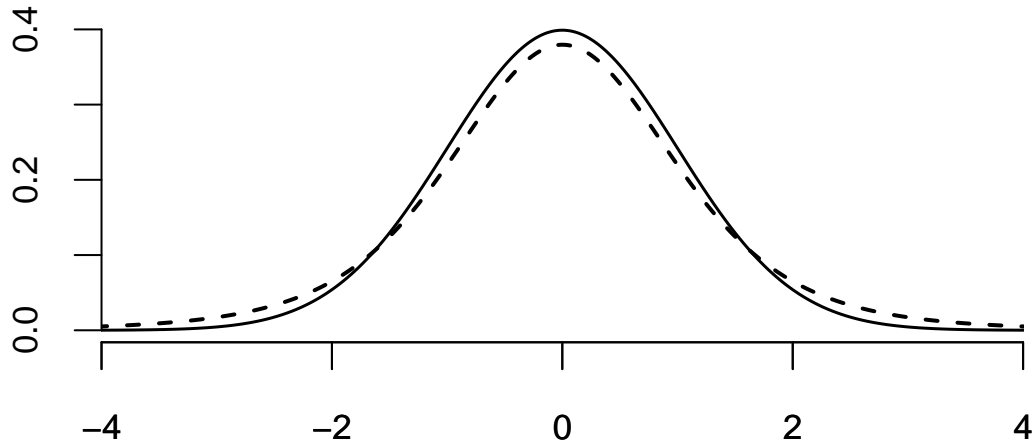
It is a consistent estimator for the population kurtosis

$$kurt = \frac{E[(Y - E[Y])^4]}{Var[Y]^2}.$$

The reference value is 3, which is the kurtosis of the standard normal distribution  $\mathcal{N}(0, 1)$ .

Values significantly above 3 indicate a distribution with heavy tails, such as the t-distribution  $t(5)$  with a kurtosis of 9, implying a higher likelihood of outliers compared to  $\mathcal{N}(0, 1)$ . Conversely, a distribution with kurtosis significantly below 3, such as the uniform distribution (kurt = 1.8), is called light-tailed, indicating fewer outliers. Both skewness and kurtosis are unit free measures.

Below you find the probability densities of the  $\mathcal{N}(0, 1)$  (solid) and the  $t(5)$  (dashed) distributions:



## Higher Moments

The  $r$ -th sample moment is

$$\bar{Y}^r = \frac{1}{n} \sum_{i=1}^n Y_i^r.$$

The sample mean is the first sample moment. The variance is the second minus the first squared sample moment (centered sample moment). The standard deviation, skewness, and kurtosis are also functions of the first four sample moments.

```
library(moments)
data(penguins, package="palmerpenguins")
Y = na.omit(penguins$body_mass_g)
length(Y)
max(Y)
min(Y)
sum(Y)
mean(Y)
var(Y)
```

```
sd(Y)
skewness(Y)
kurtosis(Y)
```

## 2.2 Empirical distribution

The distribution  $F$  of a random variable  $Y$  is defined by its **cumulative distribution function** (CDF)

$$F(a) = P(Y \leq a), \quad a \in \mathbb{R}.$$

With knowledge of  $F(\cdot)$ , you can calculate the probability of  $Y$  falling within any interval  $I \subseteq \mathbb{R}$ , or any countable union of such intervals, by applying the rules of probability.

The **empirical cumulative distribution function** (ECDF) is the sample-based counterpart of the CDF. It represents the proportion of observations within the sample that are less than or equal to a certain value  $a$ . To define the ECDF in mathematical terms, we use the concept of **order statistics**  $Y_{(h)}$ , which is the sample data arranged in ascending order such that

$$Y_{(1)} \leq Y_{(2)} \leq \dots \leq Y_{(n)}.$$

You can obtain the order statistics for your dataset using `sort(your_data)`.

The ECDF is then defined as

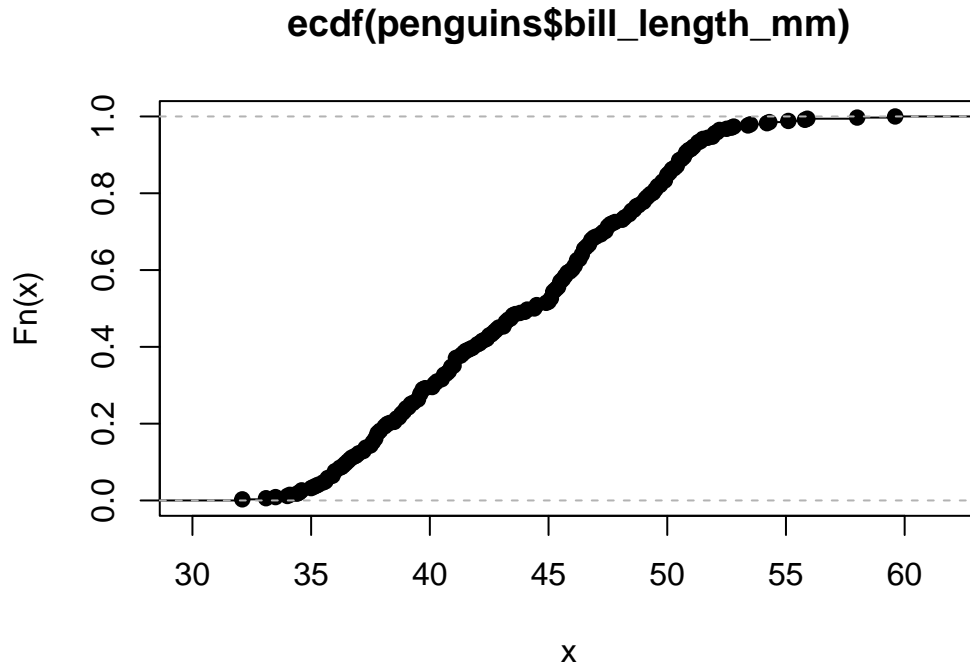
$$\widehat{F}(a) = \begin{cases} 0 & \text{for } a \in (-\infty, Y_{(1)}), \\ \frac{k}{n} & \text{for } a \in [Y_{(k)}, Y_{(k+1)}), \\ 1 & \text{for } a \in [Y_{(n)}, \infty). \end{cases}$$

The ECDF is always a step function with steps becoming arbitrarily small for continuous distributions as  $n$  increases. The ECDF is a consistent estimator for the CDF if the sample is i.i.d. (Glivenko–Cantelli theorem):

$$\sup_{a \in \mathbb{R}} |\widehat{F}(a) - F(a)| \xrightarrow{p} 0 \quad \text{as } n \rightarrow \infty.$$

```
data(penguins, package="palmerpenguins")
plot(ecdf(penguins$bill_length_mm))
```

Have a look at the ECDF's of the variables `wage`, `education`, and `female` from the `cps` data.



## 2.3 Sample quantiles

### Median

The median is a central value that splits the distribution into two equal parts.

For a continuous distribution, the population median is the value  $med$  such that  $F(med) = 0.5$ . In discrete distributions, if  $F$  is flat where it takes the value 0.5, the median isn't uniquely defined as any value within this flat region could technically satisfy the median condition  $F(med) = 0.5$ .

The empirical median of a sorted dataset is found at the point where the ECDF reaches 0.5. For an even-sized dataset, the median is the average of the two central observations:

$$\widehat{med} = \begin{cases} Y_{(\frac{n+1}{2})} & \text{if } n \text{ is odd} \\ \frac{1}{2}(Y_{(\frac{n}{2})} + Y_{(\frac{n}{2}+1)}) & \text{if } n \text{ is even} \end{cases}$$

The median corresponds to the 0.5-quantile of the distribution.

### Quantile

The population  $p$ -quantile is the value  $q_p$  such that  $F(q_p) = p$ . Similarly to the population median, population quantiles may not be unique for discrete distributions.

The empirical  $p$ -quantile  $\hat{q}_p$  is a value at which  $p$  percent of the data falls below it. It can be computed as the linear interpolation at  $h = (n - 1)p + 1$  between  $Y_{(\lfloor h \rfloor)}$  and  $Y_{(\lceil h \rceil)}$ :

$$\hat{q}_p = Y_{(\lfloor h \rfloor)} + (h - \lfloor h \rfloor)(Y_{(\lceil h \rceil)} - Y_{(\lfloor h \rfloor)}).$$

This interpolation scheme is standard in R, although multiple approaches exist for estimating quantiles (see [here](#)).

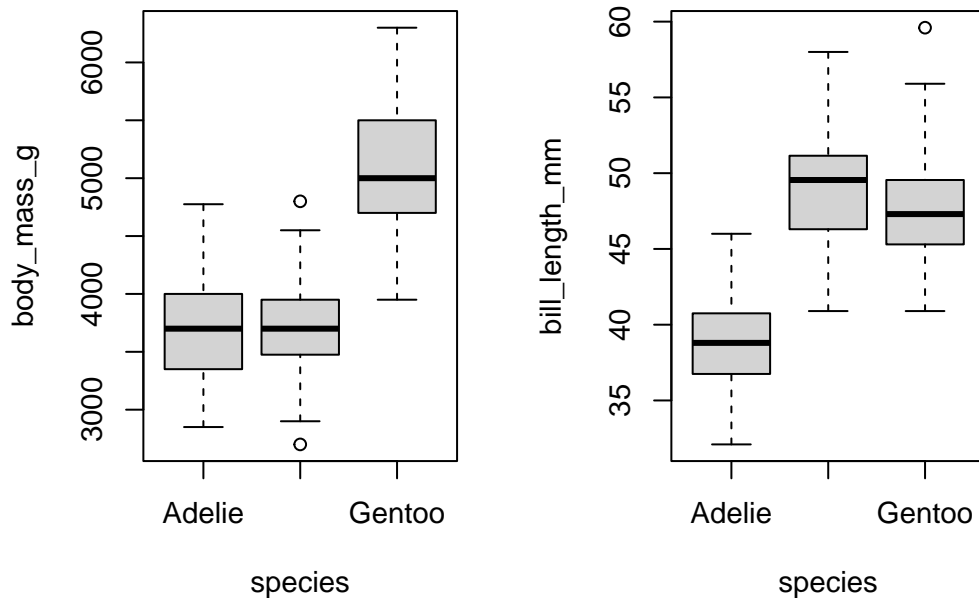
## Boxplot

Boxplots graphically represent the empirical distribution.

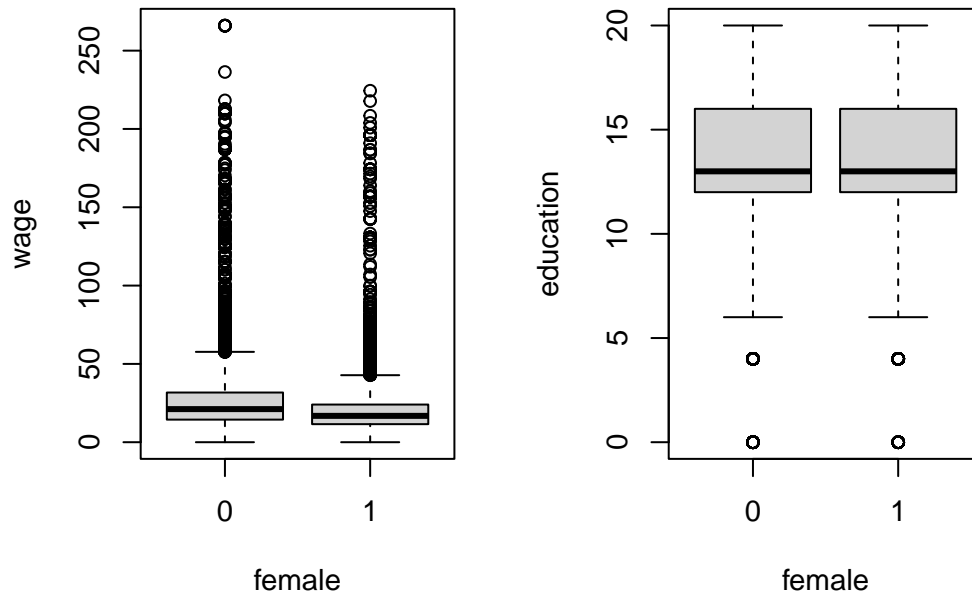
The box indicates the interquartile range ( $IQR = \hat{q}_{0.75} - \hat{q}_{0.25}$ ) and the median of the dataset. The upper whisker indicates the largest observation that does not exceed  $\hat{q}_{0.75} + 1.5IQR$ , and the lower whisker is the smallest observation that is greater or equal to  $\hat{q}_{0.25} - 1.5IQR$ . The points beyond the  $1.5IQR$  distance are plotted as single points and indicate potential outliers or the presence of a skewed or heavy tailed distribution.

Boxplots are helpful for comparing distributions across groups, such as differences in body mass or bill length among penguin species, or wage distributions by gender:

```
par(mfrow = c(1,2), cex=0.9)
boxplot(body_mass_g ~ species, data = penguins)
boxplot(bill_length_mm ~ species, data = penguins)
```



```
boxplot(wage ~ female, data = cps)
boxplot(education ~ female, data = cps)
```



## 2.4 Density estimation

A continuous random variable  $Y$  is characterized by a continuously differentiable CDF  $F(a) = P(Y \leq a)$ . The derivative is known as the probability density function (PDF), defined as  $f(a) = F'(a)$ . A simple method to estimate  $f$  is through the construction of a histogram.

### Histogram

A histogram divides the data range into  $B$  bins each of equal width  $h$  and counts the number of observations  $n_j$  within each bin. The histogram estimator of  $f(a)$  for  $a$  in the  $j$ -th bin is

$$\hat{f}(a) = \frac{n_j}{nh}.$$

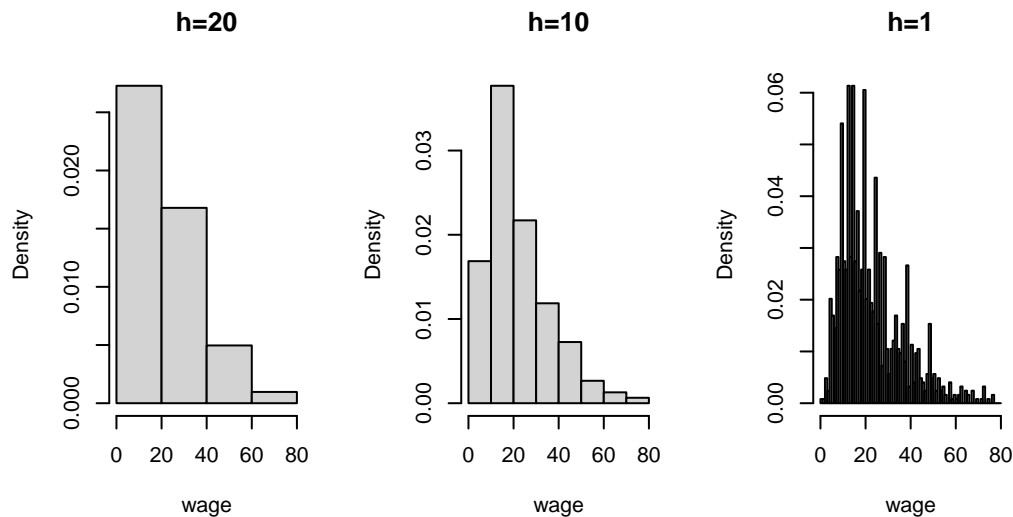
The histogram is the plot of these heights, displayed as rectangles, with their area normalized so that the total area equals 1. The appearance and accuracy of a histogram depend on the choice of bin width  $h$ .

Let's consider the subset of the CPS dataset of Asian women, excluding those with wages over \$80 for illustrative purposes:

```

library(tidyverse)
cps.new = cps |> filter(asian == 1, female == 1, wage < 80)
wage = cps.new$wage
par(mfrow = c(1,3))
hist(wage, breaks = seq(0,80,by=20), probability = TRUE, main = "h=20")
hist(wage, breaks = seq(0,80,by=10), probability = TRUE, main = "h=10")
hist(wage, breaks = seq(0,80,by=1), probability = TRUE, main = "h=1")

```



Running `hist(wage, probability=TRUE)` automatically selects a suitable bin width. `hist(wage)$breaks` shows the automatically selected break points, where the bin width is the distance between break points.

### Kernel density estimator

Suppose we want to estimate the wage density at  $a = 22$  and consider the histogram density estimate in the figure above with  $h = 10$ . It is based on the frequency of observations in the interval  $[20, 30)$  which is a skewed window about  $a = 22$ .

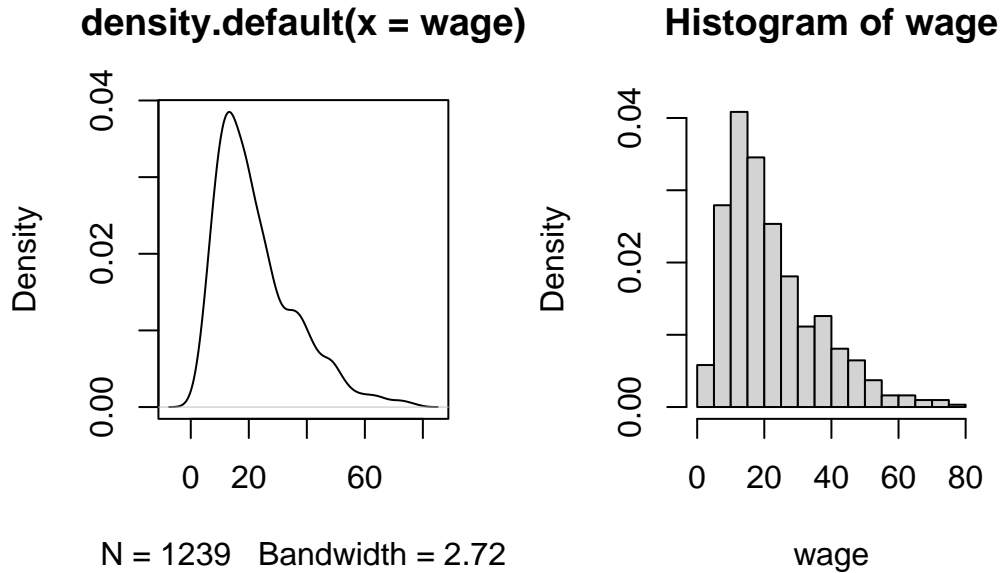
It seems more sensible to center the window at 22, for example  $[17, 27)$  instead of  $[20, 30)$ . It also seems sensible to give more weight to observations close to 22 and less to those at the edge of the window.

This idea leads to the **kernel density estimator** of  $f(a)$ , which is a smooth version of the histogram:

$$\hat{f}(a) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{X_i - a}{h}\right).$$

Here,  $K(u)$  represents a weighting function known as a kernel function, and  $h > 0$  is the **bandwidth**. A common choice for  $K(u)$  is the Gaussian kernel:

$$K(u) = \phi(u) = \frac{1}{\sqrt{2\pi}} \exp(-u^2/2).$$



The `density()` function in R automatically selects an optimal bandwidth, but it also allows for manual bandwidth specification via `density(wage, bw = your_bandwidth)`.

## 2.5 Sample covariance

Consider a multivariate dataset  $\mathbf{X}_1, \dots, \mathbf{X}_n$  represented as an  $n \times k$  data matrix  $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_n)'$ . For example, the following subset of the penguins dataset:

```
peng = penguins |>
  select(bill_length_mm, flipper_length_mm, body_mass_g) |>
  na.omit()
```

### Sample mean vector

The sample mean vector  $\bar{\mathbf{X}}$  is defined as

$$\bar{\mathbf{X}} = \frac{1}{n} \sum_{i=1}^n \mathbf{X}_i.$$



It is a consistent estimator for the population mean vector  $E[\mathbf{X}_i]$  if the sample is i.i.d..

```
colMeans(peng)
```

```
bill_length_mm flipper_length_mm    body_mass_g
      43.92193       200.91520       4201.75439
```

### Sample covariance matrix

The adjusted sample covariance matrix  $\widehat{\Sigma}$  is defined as the  $k \times k$  matrix

$$\widehat{\Sigma} = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{X}_i - \bar{\mathbf{X}})(\mathbf{X}_i - \bar{\mathbf{X}})',$$

where its  $(h, l)$  element is the pairwise **sample covariance** of variable  $h$  and  $l$  given by

$$s_{h,l} = \frac{1}{n-1} \sum_{i=1}^n (X_{ih} - \bar{X}_h)(X_{il} - \bar{X}_l), \quad \bar{X}_h = \frac{1}{n} \sum_{i=1}^n X_{ih}.$$

If the sample is i.i.d.,  $\widehat{\Sigma}$  is an unbiased and consistent estimator for the population covariance matrix  $E[(\mathbf{X}_i - E[\mathbf{X}_i])(\mathbf{X}_i - E[\mathbf{X}_i])']$ .

```
cov(peng)
```

```
              bill_length_mm flipper_length_mm body_mass_g
bill_length_mm      29.80705         50.37577      2605.592
flipper_length_mm   50.37577        197.73179      9824.416
body_mass_g        2605.59191       9824.41606  643131.077
```

### Sample correlation matrix

The correlation matrix is the matrix containing the pairwise **sample correlation coefficients**

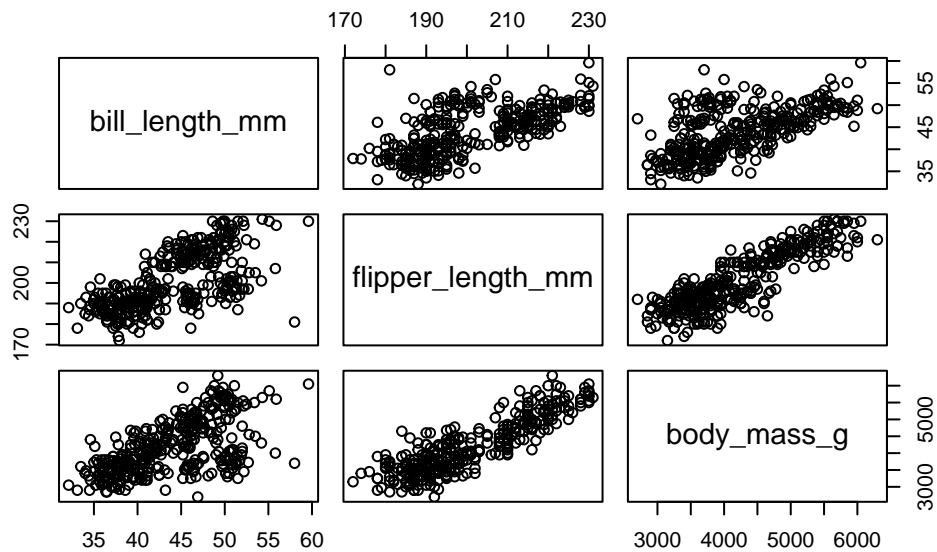
$$r_{h,l} = \frac{\sum_{i=1}^n (X_{ih} - \bar{X}_h)(X_{il} - \bar{X}_l)}{\sqrt{\sum_{i=1}^n (X_{ih} - \bar{X}_h)^2} \sqrt{\sum_{i=1}^n (X_{il} - \bar{X}_l)^2}}.$$

```
cor(peng)
```

	bill_length_mm	flipper_length_mm	body_mass_g
bill_length_mm	1.0000000	0.6561813	0.5951098
flipper_length_mm	0.6561813	1.0000000	0.8712018
body_mass_g	0.5951098	0.8712018	1.0000000

Both the covariance and correlation matrices are symmetric. The scatterplots of the full dataset visualize the positive correlations between the variables in the penguins data:

```
plot(peng)
```



## 2.6 R-codes

[methods-sec02.R](#)

## **Part II**

### **B) Linear Regression**

## 3 Least Squares

### 3.1 Regression function

The idea of regression analysis is to approximate a univariate dependent variable  $Y_i$  (also known the regressand or response variable) as a function of the  $k$ -variate vector of the independent variables  $\mathbf{X}_i$  (also known as regressors or predictor variables). The relationship is formulated as

$$Y_i \approx f(\mathbf{X}_i), \quad i = 1, \dots, n,$$

where  $Y_1, \dots, Y_n$  is a dataset for the dependent variable and  $\mathbf{X}_1, \dots, \mathbf{X}_n$  a corresponding dataset for the regressor variables.

The goal of the least squares method is to find the regression function that minimizes the squared difference between actual and fitted values of  $Y_i$ :

$$\min_{f(\cdot)} \sum_{i=1}^n (Y_i - f(\mathbf{X}_i))^2.$$

If the regression function  $f(\mathbf{X}_i)$  is linear in  $\mathbf{X}_i$ , i.e.,

$$f(\mathbf{X}_i) = b_1 + b_2 X_{i2} + \dots + b_k X_{ik} = \mathbf{X}_i' \mathbf{b}, \quad \mathbf{b} \in \mathbb{R}^k,$$

the minimization problem is known as the **ordinary least squares (OLS)** problem. To avoid the unrealistic constraint of the regression line passing through the origin, a constant term (intercept) is always included in  $\mathbf{X}_i$ , typically as the first regressor:

$$\mathbf{X}_i = (1, X_{i2}, \dots, X_{ik})'.$$

Despite its linear framework, linear regressions can be quite adaptable to nonlinear relationships by incorporating nonlinear transformations of the original regressors. Examples include polynomial terms (e.g., squared, cubic), interaction terms (combining continuous and categorical variables), and logarithmic transformations.

## 3.2 Ordinary least squares (OLS)

The **sum of squared errors** for a given coefficient vector  $\mathbf{b} \in \mathbb{R}^k$  is defined as

$$S_n(\mathbf{b}) = \sum_{i=1}^n (Y_i - f(\mathbf{X}_i))^2 = \sum_{i=1}^n (Y_i - \mathbf{X}'_i \mathbf{b})^2.$$

It is minimized by the **least squares coefficient vector**

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin}_{\mathbf{b} \in \mathbb{R}^k} \sum_{i=1}^n (Y_i - \mathbf{X}'_i \mathbf{b})^2.$$

### Least squares coefficients

If the  $k \times k$  matrix  $(\sum_{i=1}^n \mathbf{X}_i \mathbf{X}'_i)$  is invertible, the solution for the ordinary least squares problem is uniquely determined by

$$\hat{\boldsymbol{\beta}} = \left( \sum_{i=1}^n \mathbf{X}_i \mathbf{X}'_i \right)^{-1} \sum_{i=1}^n \mathbf{X}_i Y_i.$$

The **fitted values** or predicted values are

$$\hat{Y}_i = \hat{\beta}_1 + \hat{\beta}_2 X_{i2} + \dots + \hat{\beta}_k X_{ik} = \mathbf{X}'_i \hat{\boldsymbol{\beta}}, \quad i = 1, \dots, n.$$

The **residuals** are the difference between observed and fitted values:

$$\hat{u}_i = Y_i - \hat{Y}_i = Y_i - \mathbf{X}'_i \hat{\boldsymbol{\beta}}, \quad i = 1, \dots, n.$$

## 3.3 Regression plots

Let's examine the linear relationship between a penguin's body mass and its flipper length:

```
data(penguins, package="palmerpenguins")
fit1 = lm(formula = body_mass_g ~ flipper_length_mm, data = penguins)
coefficients(fit1)
```

```
(Intercept) flipper_length_mm
-5780.83136      49.68557
```

The fitted regression line is

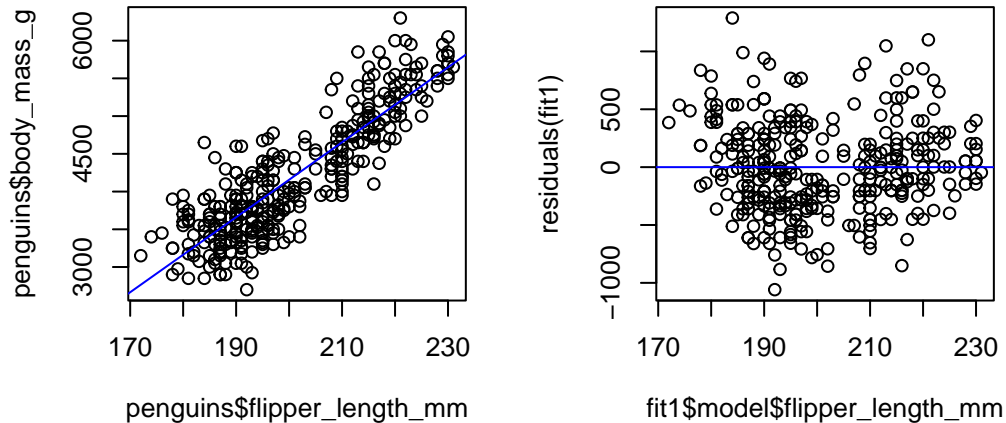
$$-5781 + 49.69 \text{ flipperlength.}$$

We can plot the regression line over a scatter plot of the data:

```

par(mfrow = c(1,2), cex=0.8)
plot(penguins$flipper_length_mm, penguins$body_mass_g)
abline(fit1, col="blue")
plot(fit1$model$flipper_length_mm, residuals(fit1))
abline(0,0,col="blue")

```



Let's include bill depth as an additional regressor:

```

fit2= lm(formula = body_mass_g ~ flipper_length_mm + bill_depth_mm,
         data = penguins)
coefficients(fit2)

```

(Intercept)	flipper_length_mm	bill_depth_mm
-6541.90750	51.54144	22.63414

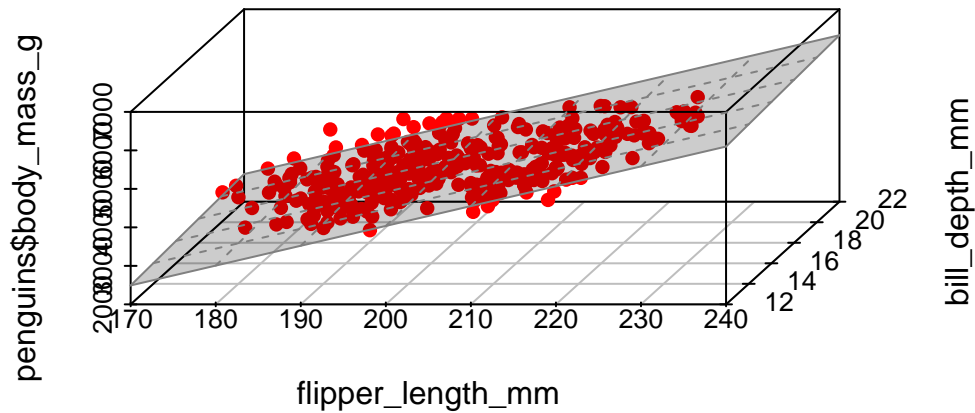
A 3D plot provides a visual representation of the resulting regression line (surface):

```

library(scatterplot3d) # package for 3d plots
Y = penguins$body_mass_g
X_2 = penguins$flipper_length_mm
X_3 = penguins$bill_depth_mm
plot3d <- scatterplot3d(x = penguins$flipper_length_mm,
                       y = penguins$bill_depth_mm,
                       z = penguins$body_mass_g,
                       angle = 60, scale.y = 0.8, pch = 16,
                       color = "red", xlab = "flipper_length_mm",
                       ylab = "bill_depth_mm",
                       main = "OLS Regression Surface")
plot3d$plane3d(fit2, lty.box = "solid", col=gray(.5), draw_polygon=TRUE)

```

## OLS Regression Surface



Adding the additional predictor bill length gives a model with dimensions beyond visual representation:

```
fit3 = lm(body_mass_g ~ flipper_length_mm + bill_depth_mm + bill_length_mm,  
          data = penguins)  
coefficients(fit3)
```

```
(Intercept) flipper_length_mm    bill_depth_mm    bill_length_mm  
-6424.76470      50.26922         20.04953         4.16182
```

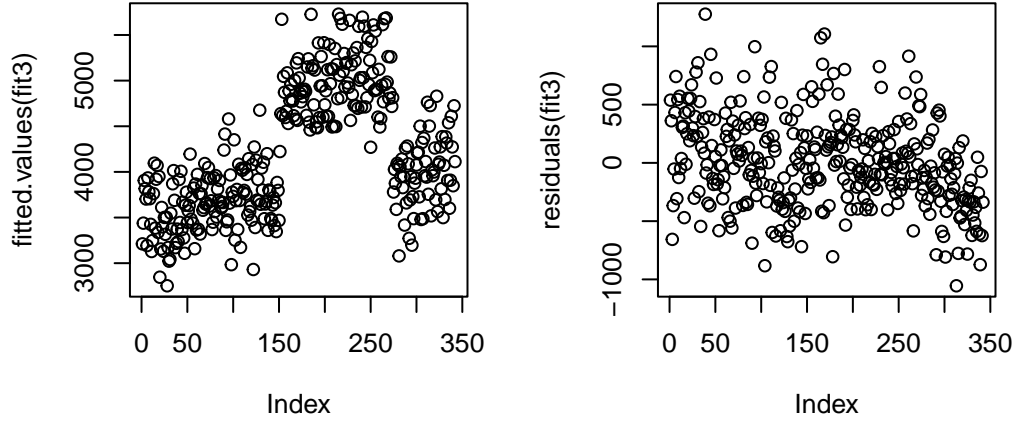
The fitted regression line now includes three predictors and four coefficients:

$$-6425 + 50.27 \text{ flipperlength} + 20.05 \text{ billdepth} + 4.16 \text{ billlength}$$

For models with multiple regressors, fitted values and residuals can still be visualized:

```
par(mfrow = c(1,2), cex=0.8)  
plot(fitted.values(fit3))  
plot(residuals(fit3))
```

The pattern of fitted values arises because the observations are sorted by penguin species.



### 3.4 Matrix notation

Matrix notation is convenient because it eliminates the need for summation symbols and indices. We define the response vector  $\mathbf{Y}$  and the regressor matrix (design matrix)  $\mathbf{X}$  as follows:

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} \mathbf{X}'_1 \\ \mathbf{X}'_2 \\ \vdots \\ \mathbf{X}'_n \end{pmatrix} = \begin{pmatrix} 1 & X_{12} & \dots & X_{1k} \\ \vdots & & & \vdots \\ 1 & X_{n2} & \dots & X_{nk} \end{pmatrix}$$

Note that  $\sum_{i=1}^n \mathbf{X}_i \mathbf{X}'_i = \mathbf{X}' \mathbf{X}$  and  $\sum_{i=1}^n \mathbf{X}_i Y_i = \mathbf{X}' \mathbf{Y}$ .

The least squares coefficient vector becomes

$$\hat{\boldsymbol{\beta}} = \left( \sum_{i=1}^n \mathbf{X}_i \mathbf{X}'_i \right)^{-1} \sum_{i=1}^n \mathbf{X}_i Y_i = (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' \mathbf{Y}.$$

The vector of fitted values can be computed as follows:

$$\hat{\mathbf{Y}} = \begin{pmatrix} \hat{Y}_1 \\ \vdots \\ \hat{Y}_n \end{pmatrix} = \mathbf{X} \hat{\boldsymbol{\beta}} = \underbrace{\mathbf{X} (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}'}_{=\mathbf{P}} \mathbf{Y} = \mathbf{P} \mathbf{Y}.$$

The **projection matrix**  $\mathbf{P}$  is also known as the *influence matrix* or *hat matrix* and maps observed values to fitted values.

The vector of residuals is given by

$$\hat{\mathbf{u}} = \begin{pmatrix} \hat{u}_1 \\ \vdots \\ \hat{u}_n \end{pmatrix} = \mathbf{Y} - \hat{\mathbf{Y}} = (\mathbf{I}_n - \mathbf{P}) \mathbf{Y}.$$



The diagonal entries of  $\mathbf{P}$ , given by

$$h_{ii} = \mathbf{X}'_i(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}_i,$$

are called **leverage values** or *hat values* and measure how far away the regressor values of the  $i$ -th observation  $X_i$  are from those of the other observations.

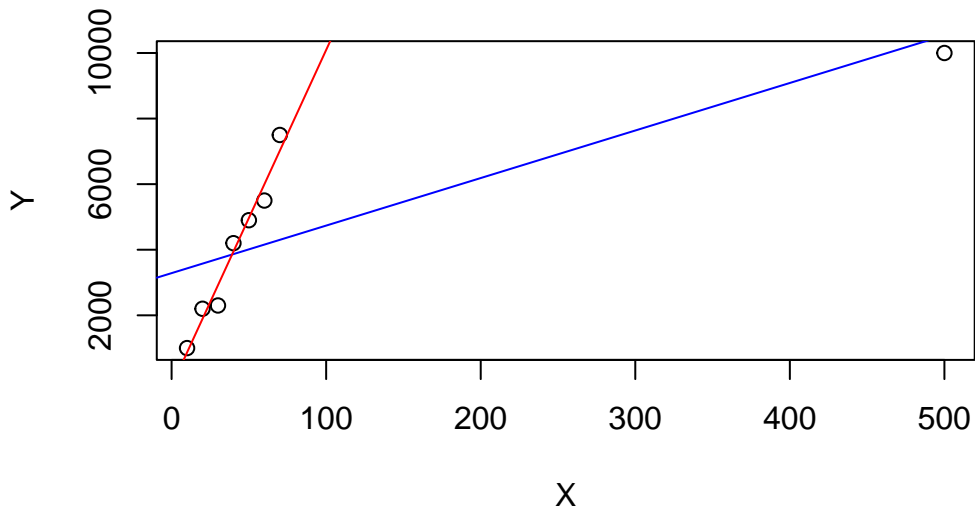
Properties of leverage values:

$$0 \leq h_{ii} \leq 1, \quad \sum_{i=1}^n h_{ii} = k.$$

A large  $h_{ii}$  occurs when the observation  $i$  has a big influence on the regression line, e.g., the last observation in the following dataset:

```
X=c(10,20,30,40,50,60,70,500)
Y=c(1000,2200,2300,4200,4900,5500,7500,10000)
plot(X,Y, main="OLS regression line with and without last observation")
abline(lm(Y~X), col="blue")
abline(lm(Y[1:7]~X[1:7]), col="red")
```

### OLS regression line with and without last observation



```
hatvalues(lm(Y~X))
```

```
1      2      3      4      5      6      7      8
0.1657356 0.1569566 0.1492418 0.1425911 0.1370045 0.1324820 0.1290237 0.9869646
```

### 3.5 R-squared

The residuals satisfy  $\mathbf{X}'\hat{\mathbf{u}} = \mathbf{0}$  and  $\widehat{\mathbf{Y}}'\hat{\mathbf{u}} = 0$ . The intercept in the regression model ensures  $\sum_{i=1}^n \hat{u}_i = 0$  and  $\sum_{i=1}^n \widehat{Y}_i = \sum_{i=1}^n Y_i$ .

Therefore, the sample variances have the following representations:

Dependent variable	$\hat{\sigma}_Y^2 = \frac{1}{n} \sum_{i=1}^n (Y_i - \bar{Y})^2$
Fitted values	$\hat{\sigma}_{\widehat{Y}}^2 = \frac{1}{n} \sum_{i=1}^n (\widehat{Y}_i - \bar{Y})^2$
Residuals	$\hat{\sigma}_{\widehat{u}}^2 = \frac{1}{n} \sum_{i=1}^n \hat{u}_i^2$
Analysis of variance formula	$\hat{\sigma}_Y^2 = \hat{\sigma}_{\widehat{Y}}^2 + \hat{\sigma}_{\widehat{u}}^2$

The larger the proportion of the explained sample variance, the better the fit of the OLS regression. This motivates the definition of the **R-squared coefficient**:

$$R^2 = \frac{\sum_{i=1}^n (\widehat{Y}_i - \bar{Y})^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} = 1 - \frac{\sum_{i=1}^n \hat{u}_i^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2}.$$

The R-squared describes the proportion of sample variation in  $\mathbf{Y}$  explained by  $\widehat{\mathbf{Y}}$ . Equivalently, it can be expressed as:  $R^2 = \hat{\sigma}_{\widehat{Y}}^2 / \hat{\sigma}_Y^2$  or  $R^2 = 1 - \hat{\sigma}_{\widehat{u}}^2 / \hat{\sigma}_Y^2$ . We have  $0 \leq R^2 \leq 1$ .

In a regression of  $Y_i$  on a single regressor  $Z_i$  with intercept (simple linear regression), the R-squared is equal to the squared sample correlation coefficient of  $Y_i$  and  $Z_i$ .

An R-squared of 0 indicates no sample variation in  $\widehat{\mathbf{Y}}$  (a flat regression line/surface), whereas a value of 1 indicates no variation in  $\hat{\mathbf{u}}$ , indicating a perfect fit. The higher the R-squared, the better the OLS regression fits the data.

However, a low R-squared does not necessarily mean the regression specification is bad. It just implies that there is a high share of unobserved heterogeneity in  $\mathbf{Y}$  that is not captured by the regressors  $\mathbf{X}$  linearly.

Conversely, a high R-squared does not necessarily mean a good regression specification. It just means that the regression fits the sample well. Too many unnecessary regressors lead to overfitting.

If  $k = n$ , we have  $R^2 = 1$  even if none of the regressors has an actual influence on the dependent variable.

We lose  $k$  degrees of freedom in the OLS regression since we have  $k$  regressors ( $k$  linear restrictions). Similar to the adjusted sample variance of  $Y$ ,  $s_Y^2 = \frac{1}{n-1} \sum_{i=1}^n (Y_i - \bar{Y})^2$ , where

we adjust for the fact that we lose 1 degree of freedom due to the sample mean (one linear restriction), the adjusted sample variance of the residuals is

$$s_{\hat{u}}^2 = \frac{1}{n-k} \sum_{i=1}^n \hat{u}_i^2.$$

By incorporating adjusted versions in the R-squared definition, we penalize regression specifications with large  $k$ . The **adjusted R-squared** is

$$\bar{R}^2 = 1 - \frac{\frac{1}{n-k} \sum_{i=1}^n \hat{u}_i^2}{\frac{1}{n-1} \sum_{i=1}^n (Y_i - \bar{Y})^2} = 1 - \frac{s_{\hat{u}}^2}{s_Y^2}.$$

The squareroot of the adjusted sample variance of the residuals is called the **standard error of the regression (SER)** or **residual standard error**:

$$SER := s_{\hat{u}} = \sqrt{\frac{1}{n-k} \sum_{i=1}^n \hat{u}_i^2}.$$

The R-squared should be used for interpreting the share of variation explained by the fitted regression line. The adjusted R-squared should be used for comparing different OLS regression specifications.

The commands `summary(fit)$r.squared` and `summary(fit)$adj.r.squared` return the R-squared and adjusted R-squared values, respectively. The *SER* can be returned by `summary(fit)$sigma`.

The `stargazer()` function can be used to produce nice regression outputs:

```
library(stargazer)
```

```
stargazer(fit1, fit2, fit3, type="latex", report="vc*", omit.stat = "f",
          star.cutoffs = NA, df=FALSE, omit.table.layout = "n",
          digits = 4, header = FALSE)
```

### 3.6 Too many regressors

OLS should be considered for regression problems with  $k \ll n$  (small  $k$  and large  $n$ ). When the number of predictors  $k$  approaches or equals the number of observations  $n$ , we run into the problem of overfitting. Specifically, at  $k = n$ , the regression line will perfectly fit the data.

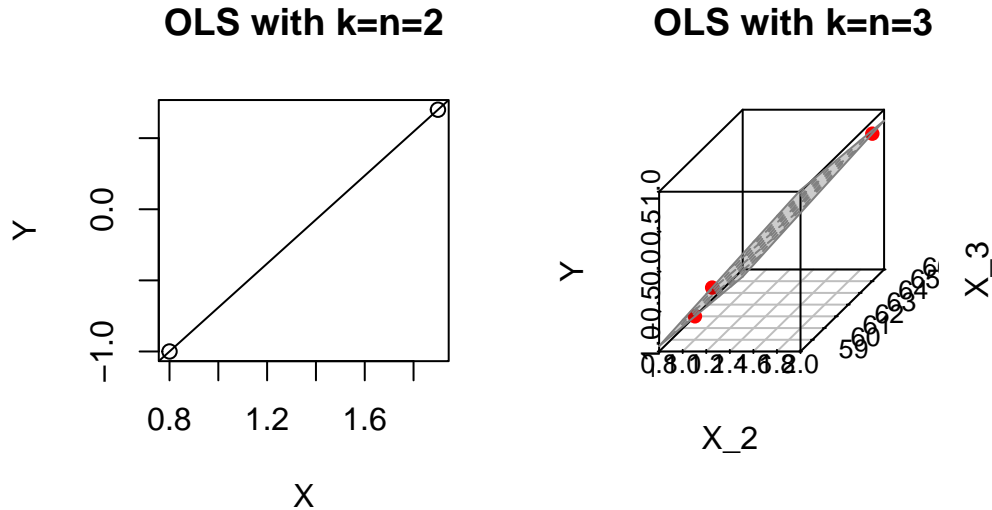
Table 3.2

	<i>Dependent variable:</i>		
	body_mass_g		
	(1)	(2)	(3)
flipper_length_mm	49.6856	51.5414	50.2692
bill_depth_mm		22.6341	20.0495
bill_length_mm			4.1618
Constant	-5,780.8310	-6,541.9080	-6,424.7650
Observations	342	342	342
R <sup>2</sup>	0.7590	0.7610	0.7615
Adjusted R <sup>2</sup>	0.7583	0.7596	0.7594
Residual Std. Error	394.2782	393.1784	393.4048

```

par(mfrow=c(1,2))
## k=n=2
Y = c(0.7,-1.0)
X = c(1.9,0.8)
fit1 = lm(Y~X)
plot(X,Y, main="OLS with k=n=2")
abline(fit1)
## k=n=3
# Some given data
Y = c(0.7,-1.0,-0.2)
X_2 = c(1.9,0.8,1.25)
X_3 = c(66, 62, 59)
fit2 = lm(Y ~ X_2 + X_3)
plot3d <- scatterplot3d(x = X_2, y = X_3, z = Y,
  angle = 33, scale.y = 0.8, pch = 16,
  color = "red",
  xlab = "X_2",
  ylab = "X_3",
  main = "OLS with k=n=3")
plot3d$plane3d(fit2, lty.box = "solid", col=gray(.5), draw_polygon=TRUE)

```



If  $k = n \geq 4$ , we can no longer visualize the OLS regression line, but the problem of a perfect fit is still present. If  $k > n$ , there exists no OLS solution because  $\mathbf{X}'\mathbf{X}$  is not invertible. Regression problems with  $k \approx n$  or  $k > n$  are called **high-dimensional regressions**.

### 3.7 Perfect multicollinearity

The only requirement for computing the OLS coefficients is the invertibility of the matrix  $\mathbf{X}'\mathbf{X}$ . As discussed above, a necessary condition is that  $k \leq n$ .

Another reason the matrix may not be invertible is if two or more regressors are perfectly collinear. Two variables are perfectly collinear if their sample correlation is 1 or -1. Multicollinearity arises if one variable is a linear combination of the other variables.

Common causes are duplicating a regressor or using the same variable in different units (e.g., GDP in both EUR and USD).

**Perfect multicollinearity** (or strict multicollinearity) arises if the regressor matrix does not have full column rank:  $\text{rank}(\mathbf{X}) < k$ . It implies  $\text{rank}(\mathbf{X}'\mathbf{X}) < k$ , so that the matrix is singular and  $\hat{\beta}$  cannot be computed.

**Near multicollinearity** occurs when two columns of  $\mathbf{X}$  have a sample correlation very close to 1 or -1. Then,  $(\mathbf{X}'\mathbf{X})$  is “near singular”, its eigenvalues are very small, and  $(\mathbf{X}'\mathbf{X})^{-1}$  becomes very large, causing numerical problems.

Multicollinearity means that at least one regressor is redundant and can be dropped.

### 3.8 Dummy variable trap

A common cause of strict multicollinearity is the inclusion of too many dummy variables. Let's add a dummy for each penguin species:

```
library(fastDummies)
penguins.new = dummy_cols(penguins,select_columns = "species")
fit4 = lm(body_mass_g ~ flipper_length_mm + species_Chinstrap
          + species_Gentoo + species_Adelie, data=penguins.new)
coefficients(fit4)
```

```
(Intercept) flipper_length_mm species_Chinstrap species_Gentoo
-4031.4769      40.7054      -206.5101      266.8096
species_Adelie
NA
```

Here, the dummy variables for penguin species are collinear with the intercept variable because  $D_{chinstrap} + D_{gentoo} + D_{adelie} = 1$ , leading to a singular matrix  $\mathbf{X}'\mathbf{X}$ . The dummy variable  $D_{adelie}$  is redundant because its value can always be recovered from  $D_{gentoo}$  and  $D_{chinstrap}$ .

The solution is to use one dummy variable less than factor levels, as R automatically does by omitting the last dummy variable. Note that the coefficient for species Adelle is NA.

Alternatively, we can incorporate the factor variable `species` directly in the regression formula as `lm()` automatically generates the correct amount of dummy variables:

```
fit5 = lm(body_mass_g ~ flipper_length_mm + species, data=penguins)
coefficients(fit5)
```

```
(Intercept) flipper_length_mm speciesChinstrap speciesGentoo
-4031.4769      40.7054      -206.5101      266.8096
```

### 3.9 R-codes

[methods-sec03.R](#)

## 4 The Linear Model

The previous section discussed OLS regression from a descriptive perspective. A regression model puts the regression problem into a stochastic framework.

Let  $\{(Y_i, \mathbf{X}'_i), i = 1, \dots, n\}$  be a sample from some joint population distribution, where  $Y_i$  is individual  $i$ 's dependent variable, and  $\mathbf{X}_i = (1, X_{i2}, \dots, X_{ik})'$  is the  $k \times 1$  vector of individual  $i$ 's regressor variables.

### Linear Regression Model

The linear regression model equation for individual  $i = 1, \dots, n$  is

$$Y_i = \beta_1 + \beta_2 X_{i2} + \dots + \beta_k X_{ik} + u_i$$

where  $\boldsymbol{\beta} = (\beta_1, \dots, \beta_k)'$  is the  $k \times 1$  vector of **regression coefficients** and  $u_i$  is the **error term** for individual  $i$ . In vector notation, we write

$$Y_i = \mathbf{X}'_i \boldsymbol{\beta} + u_i, \quad i = 1, \dots, n. \quad (4.1)$$

The error term represents further factors that affect the dependent variable and are not included in the model. These factors include measurement error, omitted variables, or unobserved/unmeasured variables.

The expression  $m(\mathbf{X}_i) = \mathbf{X}'_i \boldsymbol{\beta}$  is called the **population regression function**.

We can use matrix notation to describe the  $n$  individual regression equations together. Consider the  $n \times 1$  dependent variable vector  $\mathbf{Y}$ , the  $n \times k$  regressor matrix  $\mathbf{X}$ , and the vectors of coefficients and error terms given by

$$\underset{(k \times 1)}{\boldsymbol{\beta}} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_k \end{pmatrix}, \quad \underset{(n \times 1)}{\mathbf{u}} = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}.$$

The  $n$  equations of Equation 4.1 can be written together as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u}.$$

## 4.1 Assumptions

We assume that  $(Y_i, \mathbf{X}'_i)$ ,  $i = 1, \dots, n$ , satisfies Equation 4.1 with

- (A1) **conditional mean independence**:  $E[u_i | \mathbf{X}_i] = 0$
- (A2) **random sampling**:  $(Y_i, \mathbf{X}'_i)$  are i.i.d. draws from their joint population distribution
- (A3) **large outliers unlikely**:  $0 < E[Y_i^4] < \infty$ ,  $0 < E[X_{il}^4] < \infty$  for all  $l = 1, \dots, k$
- (A4) **no perfect multicollinearity**:  $\mathbf{X}$  has full column rank
- optional: (A5) **homoskedasticity**:  $Var[u_i | \mathbf{X}_i] = \sigma^2$
- optional: (A6) **normal errors**:  $u_i | \mathbf{X}_i$  is normally distributed

Assumptions (A1)–(A4) are required and (A5) and (A6) are optional. Model (A1)–(A4) is called **heteroskedastic linear regression model**, model (A1)–(A5) is called **homoskedastic linear regression model**, and model (A1)–(A6) is called **normal linear regression model**.

(A1)–(A2) define the properties of the regression model, (A3)–(A4) imply that OLS can be used to estimate the model, and (A5)–(A6) ensure that classical exact inference can be used without relying on robust large sample methods.

For all  $i, j = 1, \dots, n$ , the model has the following properties:

- (i) **Conditional expectation**: (A1) implies

$$E[Y_i | \mathbf{X}_i] = \mathbf{X}'_i \boldsymbol{\beta} = m(\mathbf{X}_i).$$

- (ii) **Weak exogeneity**: (A1) implies

$$E[u_i] = 0, \quad Cov(u_i, X_{il}) = 0.$$

- (iii) **Strict exogeneity**: (A1)+(A2) imply

$$E[u_i | \mathbf{X}] = 0, \quad Cov(u_i, X_{jl}) = 0.$$

- (iv) **Heteroskedasticity**: (A1)+(A2) imply

$$Var[u_i | \mathbf{X}] = E[u_i^2 | \mathbf{X}_i] =: \sigma_i^2$$



(v) **No autocorrelation:** (A1)+(A2) imply

$$E[u_i u_j | \mathbf{X}] = 0, \quad Cov(u_i, u_j) = 0, \quad i \neq j.$$

The errors have a diagonal conditional covariance matrix:

$$\mathbf{D} := Var[\mathbf{u} | \mathbf{X}] = \begin{pmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n^2 \end{pmatrix}.$$

## 4.2 OLS Estimator

The OLS coefficient vector  $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$  can be used to estimate  $\boldsymbol{\beta}$ . For all  $i = 1, \dots, n$  and  $l = 1, \dots, K$ , the OLS estimator has the following properties:

(i) **Existence:** (A4) implies that  $\mathbf{X}'\mathbf{X}$  is invertible and that  $\hat{\boldsymbol{\beta}}$  exists.

(ii) **Unbiasedness:** (A1)+(A2)+(A4) imply

$$E[\hat{\boldsymbol{\beta}} | \mathbf{X}] = \boldsymbol{\beta}.$$

(iii) **Sampling variance:** (A1)+(A2)+(A4) imply

$$Var[\hat{\boldsymbol{\beta}} | \mathbf{X}] = (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{D}\mathbf{X})(\mathbf{X}'\mathbf{X})^{-1}.$$

If (A5) holds as well, then  $\mathbf{D} = \mathbf{I}_n$  and  $Var[\hat{\boldsymbol{\beta}} | \mathbf{X}] = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$ .

(iv) **Normality:** (A1)+(A2)+(A4)+(A6) imply

$$\hat{\boldsymbol{\beta}} | \mathbf{X} \sim \mathcal{N}(\boldsymbol{\beta}, Var[\hat{\boldsymbol{\beta}} | \mathbf{X}])$$

(v) **Consistency:** (A1)–(A4) imply

$$\hat{\boldsymbol{\beta}} \xrightarrow{p} \boldsymbol{\beta} \quad \text{as } n \rightarrow \infty$$

since the bias is zero and the variance asymptotically tends to zero.

(vi) **Asymptotic variance:** Let  $\mathbf{Q} := E[\mathbf{X}_i \mathbf{X}_i']$  and  $\boldsymbol{\Omega} := E[u_i^2 \mathbf{X}_i \mathbf{X}_i']$ . (A1)–(A4) imply

$$Var[\hat{\boldsymbol{\beta}} | \mathbf{X}] = \frac{1}{n} \underbrace{\left( \frac{1}{n} \mathbf{X}'\mathbf{X} \right)^{-1}}_{\xrightarrow{p} \mathbf{Q}} \underbrace{\left( \frac{1}{n} \mathbf{X}'\mathbf{D}\mathbf{X} \right)}_{\xrightarrow{p} \boldsymbol{\Omega}} \underbrace{\left( \frac{1}{n} \mathbf{X}'\mathbf{X} \right)^{-1}}_{\xrightarrow{p} \mathbf{Q}} \xrightarrow{p} \mathbf{0},$$

and

$$Var[\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) | \mathbf{X}] \xrightarrow{p} \mathbf{Q}^{-1} \boldsymbol{\Omega} \mathbf{Q}^{-1}.$$

If (A5) holds as well, then  $\boldsymbol{\Omega} = \sigma^2 \mathbf{Q}$ , and  $\mathbf{Q}^{-1} \boldsymbol{\Omega} \mathbf{Q}^{-1} = \sigma^2 \mathbf{Q}^{-1}$ .

(vii) **Asymptotic normality:** (A1)–(A4) imply

$$\sqrt{n}(\hat{\beta} - \beta) \xrightarrow{D} \mathcal{N}(\mathbf{0}, \mathbf{Q}^{-1}\mathbf{\Omega}\mathbf{Q}^{-1}).$$

Technical details can be found in Appendix A.

### 4.3 Marginal Effects

For example, consider the regression model of hourly wage on education (years of schooling):

$$wage_i = \beta_1 + \beta_2 edu_i + u_i, \quad E[u_i | edu_i] = 0, \quad i = 1, \dots, n. \quad (4.2)$$

The population regression function is  $m(edu_i) = \beta_1 + \beta_2 edu_i$ . (A1) implies that

$$E[wage_i | edu_i] = \underbrace{\beta_1 + \beta_2 edu_i}_{=m(edu_i)} + \underbrace{E[u_i | edu_i]}_{=0}.$$

The average wage level of all individuals with  $z$  years of schooling is  $\beta_1 + \beta_2 z$ .

$$Cov(wage_i, edu_i) = \underbrace{Cov(m(edu_i), edu_i)}_{=\beta_2 Var[edu_i]} + \underbrace{Cov(u_i, edu_i)}_{=0}$$

The coefficient  $\beta_2$  is identified as

$$\beta_2 = \frac{Cov(wage_i, edu_i)}{Var[edu_i]} = Corr(wage_i, edu_i) \cdot \frac{sd(wage_i)}{sd(edu_i)}.$$

The coefficient describes the **correlative relationship** between education and wages.

The marginal effect of education is

$$\frac{\partial E[wage_i | edu_i]}{\partial edu_i} = \beta_2.$$

```
lm(wage ~ education, data = cps)
```

Call:

```
lm(formula = wage ~ education, data = cps)
```

Coefficients:

(Intercept)	education
-16.448	2.898

*Interpretation:* People with one more year of education are paid on average 2.90 USD more than people with one year less of education.

The marginal effect is a correlative effect and does not say where exactly a higher wage level for people with more education comes from. **Regression relationships do not necessarily imply a causal relationship.**

People with more education may earn more for a number of reasons. Maybe they are generally smarter or come from wealthier families, which leads to better paying jobs. Or maybe more education actually leads to higher earning

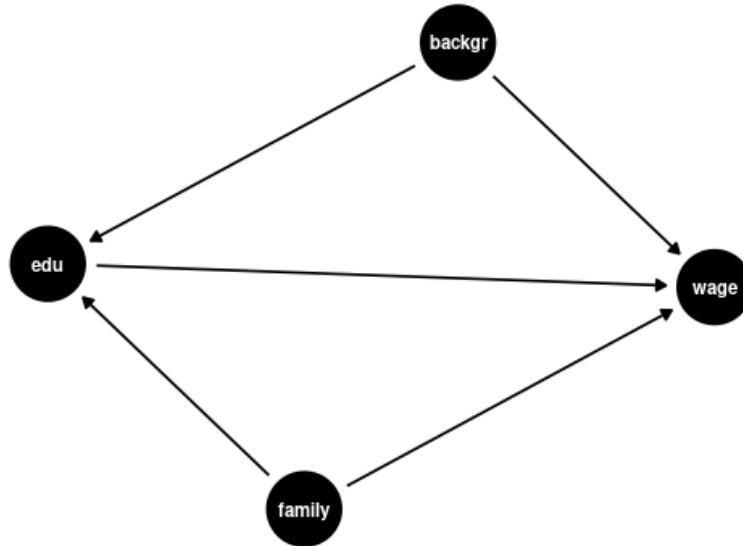


Figure 4.1: A DAG (directed acyclic graph) for the correlative and causal effects of edu on wage

The coefficient  $\beta_2$  is a measure of how strongly education and earnings are correlated.

This association could be due to other factors that correlate with both wages and education, such as family background (parental education, family income, ethnicity, structural racism) or personal background (gender, intelligence).

Notice: Correlation does not imply causation!

To disentangle the causal effect of education on wages from other correlative effects, we can include control variables.

## 4.4 Control Variables

To understand the causal effect of an additional year of education on wages, it is crucial to consider the influence of family and personal background. These factors, if not included in our analysis, are known as **omitted variables**. An omitted variable is one that:

- (i) it is correlated with the dependent variable (wage, in this scenario),
- (ii) correlated with the regressor of interest (education),
- (iii) omitted in the regression.

The presence of omitted variables means that we cannot be sure that the regression relationship between education and wages is purely causal. We say that we have **omitted variable bias** for the causal effect of the regressor of interest.

The coefficient  $\beta_2$  in Equation 4.2 measures the correlative or marginal effect, not the causal effect. This must always be kept in mind when interpreting regression coefficients.

We can include **control variables** in the linear regression model to reduce omitted variable bias so that we can interpret  $\beta_2$  as a **ceteris paribus marginal effect** (ceteris paribus means holding other variables constant).

For example, let's include years of experience as well as racial background and gender dummy variables for Black and female:

$$wage_i = \beta_1 + \beta_2 edu_i + \beta_3 ex_i + \beta_4 Black_i + \beta_5 fem_i + u_i.$$

In this case,

$$\beta_2 = \frac{\partial E[wage_i | edu_i, ex_i, Black_i, fem_i]}{\partial edu_i}$$

is the marginal effect of education on expected wages, holding experience, race, and gender fixed.

```
lm(wage ~ education + experience + black + female, data = cps)
```

Call:

```
lm(formula = wage ~ education + experience + black + female,  
    data = cps)
```

Coefficients:

(Intercept)	education	experience	black	female
-21.7095	3.1350	0.2443	-2.8554	-7.4363

*Interpretation:* Given the same experience, racial background, and gender, people with one more year of education are paid on average 3.14 USD more than people with one year less of education.

Note: It does not hold other unobservable characteristics (such as ability) or variables not included in the regression (such as quality of education) fixed, so an omitted variable bias may still be present.

Good control variables are variables that are determined before the level of education is determined. Control variables should not be the cause of the dependent variable of interest.

Examples of **good controls** for education are parental education level, region of residence, or educational industry/field of study.

A problematic situation is when the control variable is the cause of education. Bad controls are typically highly correlated with the independent variable of interest and irrelevant to the causal effect of that variable on the dependent variable.

Examples of **bad controls** for education are current job position, number of professional certifications obtained, or number of job offers.

A high correlation of the bad control with the variable education also causes a high variance of the OLS coefficient for education and leads to an imprecise coefficient estimate. This problem is called **imperfect multicollinearity**.

Bad controls make it difficult to interpret causal relationships. They may control away the effect you want to measure, or they may introduce additional reverse causal effects hidden in the regression coefficients.

## 4.5 Polynomials

A linear dependence on wages and experience is a strong assumption. We can reasonably expect a nonlinear marginal effect of another year of experience on wages. For example, the effect may be higher for workers with 5 years of experience than for those with 40 years of experience.

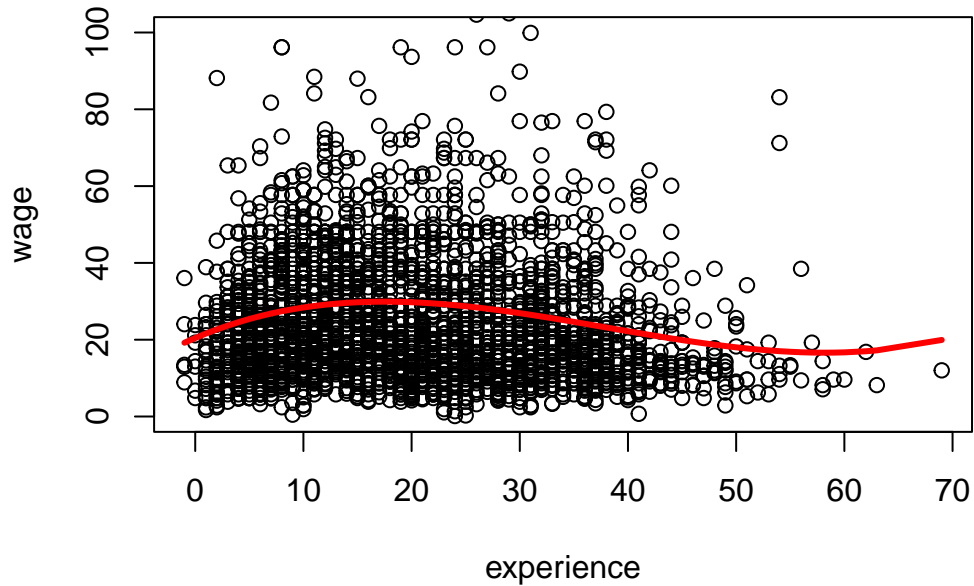
Polynomials can be used to specify a nonlinear regression function:

$$wage_i = \beta_1 + \beta_2 ex_i + \beta_3 ex_i^2 + \beta_4 ex_i^3 + u_i.$$

```
## we focus on Asian people only for illustration
cps.as = cps |> subset(asian == 1)
fit = lm(wage ~ experience + I(experience^2) + I(experience^3),
        data = cps.as)
coefficients(fit)
```

(Intercept)	experience	I(experience^2)	I(experience^3)
20.4547146896	1.2013241316	-0.0446897909	0.0003937551

```
plot(wage ~ experience, data = cps.as, ylim = c(0,100))
lines(sort(cps.as$experience),
      fitted(fit)[order(cps.as$experience)],
      col='red', type='l', lwd=3)
```



The marginal effect depends on the years of experience:

$$\frac{\partial E[wage_i | ex_i]}{\partial ex_i} = \beta_2 + 2\beta_3 ex_i + 3\beta_4 ex_i^2.$$

For instance, the additional wage for a worker with 11 years of experience compared to a worker with 10 years of experience is on average

$$1.43 + 2 \cdot (-0.042) \cdot 10 + 3 \cdot 0.0003 \cdot 10^2 = 0.68.$$

## 4.6 Interactions

A linear regression with interaction terms:

$$wage_i = \beta_1 + \beta_2 edu_i + \beta_3 fem_i + \beta_4 marr_i + \beta_5 (marr_i \cdot fem_i) + u_i$$

```
lm(wage ~ education + female + married + married:female, data = cps)
```

Call:

```
lm(formula = wage ~ education + female + married + married:female,  
    data = cps)
```

Coefficients:

(Intercept)	education	female	married	female:married
-17.886	2.867	-3.266	7.167	-5.767

The marginal effect of gender depends on the person's marital status:

$$\frac{\partial E[\text{wage}_i | \text{edu}_i, \text{female}_i, \text{married}_i]}{\partial \text{female}_i} = \beta_3 + \beta_5 \text{married}_i$$

*Interpretation:* Given the same education, unmarried women are paid on average 3.26 USD less than unmarried men, and married women are paid on average  $3.27 + 5.77 = 9.04$  USD less than married men.

The marginal effect of the marital status depends on the person's gender:

$$\frac{\partial E[\text{wage}_i | \text{edu}_i, \text{female}_i, \text{married}_i]}{\partial \text{married}_i} = \beta_4 + \beta_5 \text{female}_i$$

*Interpretation:* Given the same education, married men are paid on average 7.17 USD more than unmarried men, and married women are paid on average  $7.17 - 5.77 = 1.40$  USD more than unmarried women.

## 4.7 Logarithms

In the logarithmic specification

$$\log(\text{wage}_i) = \beta_1 + \beta_2 \text{edu}_i + u_i$$

we have

$$\frac{\partial E[\log(\text{wage}_i) | \text{edu}_i]}{\partial \text{edu}_i} = \beta_2.$$

This implies

$$\underbrace{\frac{\partial E[\log(\text{wage}_i) | \text{edu}_i]}{\partial \text{edu}_i}}_{\text{absolute change}} = \beta_2 \cdot \underbrace{\frac{\partial \text{edu}_i}{\partial \text{edu}_i}}_{\text{absolute change}}.$$

That is,  $\beta_2$  gives the average absolute change in log wages when education changes by 1.

Another interpretation can be given in terms of relative changes. Consider the following approximation:

$$E[wage_i | edu_i] \approx \exp(E[\log(wage_i) | edu_i]).$$

The left-hand expression is the conventional conditional mean, and the right-hand expression is the geometric mean. The geometric mean is slightly smaller because  $E[\log(Y)] < \log(E[Y])$ , but the difference is small unless the data is highly skewed.

The marginal effect of a change in  $edu$  on the geometric mean of  $wage$  is

$$\frac{\partial \exp(E[\log(wage_i) | edu_i])}{\partial edu_i} = \underbrace{\exp(E[\log(wage_i) | edu_i])}_{\text{outer derivative}} \cdot \beta_2.$$

Using the geometric mean approximation from above, we get

$$\underbrace{\frac{\partial E[wage_i | edu_i]}{E[wage_i | edu_i]}}_{\text{percentage change}} \approx \frac{\partial \exp(E[\log(wage_i) | edu_i])}{\exp(E[\log(wage_i) | edu_i])} = \beta_2 \cdot \underbrace{\frac{\partial edu_i}{edu_i}}_{\text{absolute change}}.$$

```
linear_model <- lm(wage ~ education, data = cps.as)
log_model <- lm(log(wage) ~ education, data = cps.as)
log_model
```

Call:

```
lm(formula = log(wage) ~ education, data = cps.as)
```

Coefficients:

```
(Intercept)    education
    1.3783         0.1113
```

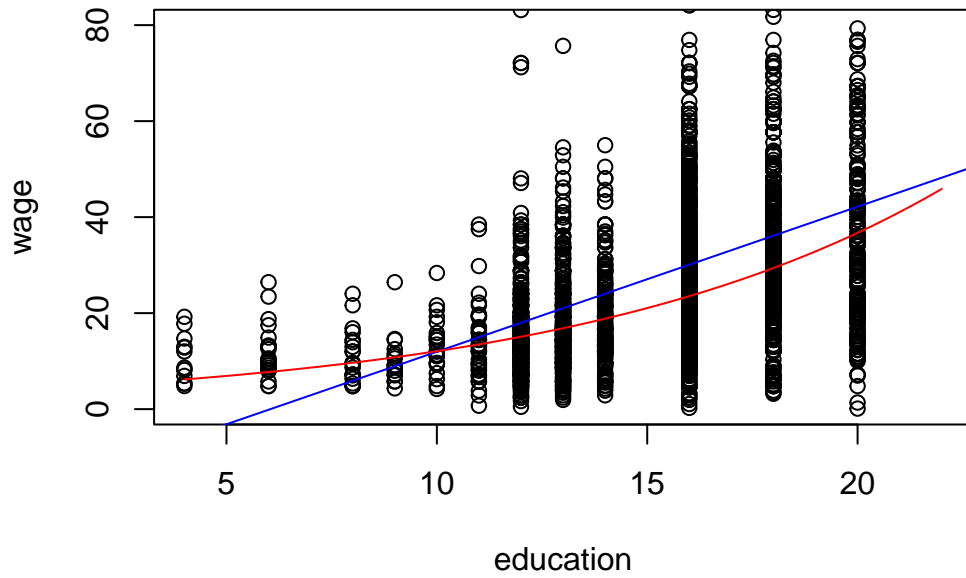
```
plot(wage ~ education, data = cps.as, ylim = c(0,80), xlim = c(4,22))
abline(linear_model, col="blue")
coef = coefficients(log_model)
curve(exp(coef[1]+coef[2]*x), add=TRUE, col="red")
```

*Interpretation:* A person with one more year of education has a wage that is 11.13% higher on average.

In addition to the linear-linear and log-linear specifications, we also have the linear-log specification

$$Y = \beta_1 + \beta_2 \log(X) + u$$





and the log-log specification

$$\log(Y) = \beta_1 + \beta_2 \log(X) + u.$$

*Linear-log interpretation:* When  $X$  is 1% higher, we observe, on average, a  $0.01\beta_2$  higher  $Y$ .

*Log-log interpretation:* When  $X$  is 1% higher, we observe, on average, a  $\beta_2\%$  higher  $Y$ .

## 4.8 R-codes

[methods-sec04.R](#)

## 5 Regression Inference

```
library(tidyverse)
library(kableExtra)
library(sandwich)
library(lmtest)
```

### 5.1 Standardized coefficients

The  $j$ -th OLS coefficient has the conditional standard deviation

$$sd(\hat{\beta}_j|\mathbf{X}) = \sqrt{[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{D}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}]_{jj}}.$$

Note that  $[\mathbf{A}]_{jj}$  indicates the  $j$ -th diagonal element of the matrix  $\mathbf{A}$ .

Under the homoskedasticity assumption (A5), the standard deviation simplifies to

$$sd(\hat{\beta}_j|\mathbf{X}) = \sqrt{\sigma^2[(\mathbf{X}'\mathbf{X})^{-1}]_{jj}}.$$

The coefficient is unbiased with  $E[\hat{\beta}_j|\mathbf{X}] = \beta_j$  and has the standardized representation

$$Z_j := \frac{\hat{\beta}_j - \beta_j}{sd(\hat{\beta}_j|\mathbf{X})}.$$

Under (A1)–(A4),  $\sqrt{n}(\hat{\beta}_j - \beta_j)$  converges to a normal distribution, and therefore

$$Z_j \xrightarrow{D} \mathcal{N}(0, 1) \quad \text{as } n \rightarrow \infty.$$

A direct consequence is that

$$\lim_{n \rightarrow \infty} P\left(\hat{\beta}_j - z_{(1-\frac{\alpha}{2})}sd(\hat{\beta}_j|\mathbf{X}) \leq \beta_j \leq \hat{\beta}_j + z_{(1-\frac{\alpha}{2})}sd(\hat{\beta}_j|\mathbf{X})\right) = 1 - \alpha,$$

where  $z_{(p)}$  is the  $p$ -quantile of the standard normal distribution. Thus,  $\hat{\beta}_j \pm z_{(1-\frac{\alpha}{2})}sd(\hat{\beta}_j|\mathbf{X})$  defines an asymptotic  $1 - \alpha$  confidence interval for  $\beta_j$ .

Under the normality assumption (A6), the OLS estimator  $\hat{\beta}_j$  is normal conditional on  $\mathbf{X}$ , which implies that  $Z_j \sim \mathcal{N}(0, 1)$  for any fixed sample size  $n$ . In this case,  $\hat{\beta}_j \pm z_{(1-\frac{\alpha}{2})} sd(\hat{\beta}_j|\mathbf{X})$  is an exact confidence interval for  $\beta_j$ .

Note that  $\mathbf{D}$  is unknown and  $sd(\hat{\beta}_j|\mathbf{X})$  is not computable in practice, so the confidence interval is not feasible.

## 5.2 Standard Errors

A standard error  $se(\hat{\beta}_j)$  for an estimator  $\hat{\beta}_j$  is an estimator of the standard deviation of the distribution of  $\hat{\beta}_j$ .

We say that the standard error is consistent if

$$\frac{se(\hat{\beta}_j)}{sd(\hat{\beta}_j|\mathbf{X})} \xrightarrow{p} 1.$$

This property ensures that, in practice, we can replace the unknown standard deviation with the standard error to apply inferential methods such as confidence intervals and t-tests.

To estimate the unknown standard deviation of the OLS estimator, the diagonal matrix  $\mathbf{D} = \text{diag}(\sigma_1^2, \dots, \sigma_n^2)$  is replaced by some sample counterpart  $\hat{\mathbf{D}} = \text{diag}(\hat{\sigma}_1^2, \dots, \hat{\sigma}_n^2)$ .

### 5.2.1 Robust standard errors

Various **heteroskedasticity-consistent (HC)** standard errors have been proposed in the literature:

HC type	weights
HC0	$\hat{\sigma}_i^2 = \hat{u}_i^2$
HC1	$\hat{\sigma}_i^2 = \frac{n}{n-k} \hat{u}_i^2$
HC2	$\hat{\sigma}_i^2 = \frac{\hat{u}_i^2}{1-h_{ii}}$
HC3	$\hat{\sigma}_i^2 = \frac{\hat{u}_i^2}{(1-h_{ii})^2}$

HC0 replaces the unknown variances with squared residuals, and HC1 is a bias-corrected version of HC0. HC2 and HC3 use the leverage values  $h_{ii}$  (the diagonal entries of the influence matrix  $\mathbf{P}$ ) and give less weight to influential observations.

HC1 and HC3 are the most common choices and can be written as

$$se_{hc1}(\hat{\beta}_j) = \sqrt{\left[ (\mathbf{X}'\mathbf{X})^{-1} \left( \frac{n}{n-k} \sum_{i=1}^n \hat{u}_i^2 \mathbf{X}_i \mathbf{X}_i' \right) (\mathbf{X}'\mathbf{X})^{-1} \right]_{jj}},$$

$$se_{hc3}(\hat{\beta}_j) = \sqrt{\left[ (\mathbf{X}'\mathbf{X})^{-1} \left( \sum_{i=1}^n \frac{\hat{u}_i^2}{(1-h_{ii})^2} \mathbf{X}_i \mathbf{X}_i' \right) (\mathbf{X}'\mathbf{X})^{-1} \right]_{jj}}.$$

All versions perform similarly well in large samples, but HC3 performs best in small samples and is the preferred choice.

HC standard errors are also known as **heteroskedasticity-robust standard errors** or simply **robust standard errors**.

Estimators for the full covariance matrix of  $\hat{\beta}$  have the form

$$\widehat{\mathbf{V}} = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}' \widehat{\mathbf{D}} \mathbf{X} (\mathbf{X}'\mathbf{X})^{-1}.$$

The HC3 covariance estimator can be written as

$$\widehat{\mathbf{V}}_{hc3} = (\mathbf{X}'\mathbf{X})^{-1} \left( \sum_{i=1}^n \frac{\hat{u}_i^2}{(1-h_{ii})^2} \mathbf{X}_i \mathbf{X}_i' \right) (\mathbf{X}'\mathbf{X})^{-1}.$$

### 5.2.2 Classical standard errors

**Classical standard errors** put equal weights on all observations:

$$\hat{\sigma}_i^2 = s_u^2 = \frac{1}{n-k} \sum_{j=1}^n \hat{u}_j^2.$$

This implies  $\widehat{\mathbf{D}} = s_u^2 \mathbf{I}_n$  and  $\mathbf{X}' \widehat{\mathbf{D}} \mathbf{X} = s_u^2 \mathbf{X}' \mathbf{X}$ . Therefore, the classical covariance matrix estimator reduces to

$$\widehat{\mathbf{V}}_{hom} = s_u^2 (\mathbf{X}'\mathbf{X})^{-1}.$$

The classical standard errors are

$$se_{hom}(\hat{\beta}_j) = \sqrt{s_u^2 [(\mathbf{X}'\mathbf{X})^{-1}]_{jj}}.$$

Classical standard errors are only valid under (A5) and are also known as **constant variance standard errors** or **homoskedasticity-only standard errors**. Classical standard errors should only be used if there are very good reasons for the error terms to be homoskedastic.

## 5.2.3 Standard Errors in R

The covariance matrix estimates can be computed using the `vcovHC()` function from the `sandwich` package. HC3 is the default version. The standard errors are the square roots of their diagonal entries.

```
fit = lm(wage ~ education + experience + black + female, data = cps)
hom = sqrt(diag(vcovHC(fit, "const")))
HC1 = sqrt(diag(vcovHC(fit, "HC1")))
HC3 = sqrt(diag(vcovHC(fit)))
tibble("Variable" = names(coefficients(fit)), hom, HC1, HC3) |>
  mutate_if(is.numeric, round, digits = 4) |>
  kbl(align = 'c')
```

Variable	hom	HC1	HC3
(Intercept)	0.4910	0.5666	0.5667
education	0.0305	0.0408	0.0409
experience	0.0072	0.0067	0.0067
black	0.2684	0.2243	0.2243
female	0.1670	0.1603	0.1604

## 5.3 Interval estimates

### 5.3.1 Asymptotic Intervals

A **confidence interval**  $I_{1-\alpha}$  for  $\beta_j$  with coverage probability  $1 - \alpha$  is asymptotically valid if

$$\lim_{n \rightarrow \infty} P(\beta_j \in I_{1-\alpha}) = 1 - \alpha.$$

Under (A1)–(A4), we can use

$$I_{1-\alpha} = [\hat{\beta}_j - z_{(1-\frac{\alpha}{2})} se_{hc}(\hat{\beta}_j); \hat{\beta}_j + z_{(1-\frac{\alpha}{2})} se_{hc}(\hat{\beta}_j)],$$

where  $se_{hc}(\hat{\beta}_j)$  is any HC-type standard error.  $z_{(p)}$  can be returned using `qnorm(p)`.

In practice, t-quantiles are often used instead of z-quantiles:

$$I_{1-\alpha} = [\hat{\beta}_j - t_{(1-\frac{\alpha}{2}, n-k)} se_{hc}(\hat{\beta}_j); \hat{\beta}_j + t_{(1-\frac{\alpha}{2}, n-k)} se_{hc}(\hat{\beta}_j)],$$

where  $t_{(p,m)}$  is the  $p$ -quantile of the t-distribution with  $m$  degrees of freedom.  $t_{(p,m)}$  can be returned using `qt(p,m)`.

Asymptotically, it makes no difference whether t- or z-quantiles are used. We have

$$t_{(1-\frac{\alpha}{2}, n-k)} > z_{(1-\frac{\alpha}{2})}$$

for any fixed  $n$ , which makes the t-based confidence intervals a little wider (conservative), but asymptotically they coincide because

$$\lim_{n \rightarrow \infty} t_{(1-\frac{\alpha}{2}, n-k)} = z_{(1-\frac{\alpha}{2})}.$$

You can use the `coefci()` function from the `lmtest` package. `coefci(fit)` calculates classical confidence intervals, `coefci(fit, vcov. = vcovHC)` uses HC3 standard errors, and `coefci(fit, vcov. = vcovHC, df=Inf)` considers z-quantiles instead of t-quantiles.

```
coefci(fit, vcov. = vcovHC)
```

	2.5 %	97.5 %
(Intercept)	-22.8201704	-20.5988645
education	3.0549552	3.2151008
experience	0.2311859	0.2574641
black	-3.2951083	-2.4157606
female	-7.7505755	-7.1219793

You can use `qt(p, df = nu)` and `qnorm(p)` to get the t- and z-quantiles, where  $p$  is the probability and  $nu$  is the degrees of freedom. The CDF values for the standard normal and t-distributions can be calculated using `pt()` and `pnorm()`.

### 5.3.2 Exact Intervals

An exact confidence interval  $I_{1-\alpha}$  for  $\beta_j$  satisfies

$$P(\beta_j \in I_{1-\alpha}) = 1 - \alpha$$

for any sample size  $n$ .

Exact confidence intervals for the regression coefficients are only available if the homoskedasticity and normality assumptions (A5) and (A6) hold. In this case,

$$\frac{(n-k)s_u^2}{\sigma^2} \sim \chi_{n-k}^2,$$

which implies that

$$\frac{se_{hom}(\hat{\beta}_j)}{sd(\hat{\beta}_j|\mathbf{X})} \sim \sqrt{\chi_{n-k}^2 / (n-k)}.$$

Replacing the true standard deviation with the classical standard error in the standardized OLS coefficient  $Z_j$  yields

$$\frac{\hat{\beta}_j - \beta_j}{se_{hom}(\beta_j)} = \frac{Z_j}{se_{hom}(\hat{\beta}_j)/sd(\hat{\beta}_j|\mathbf{X})} \sim \frac{\mathcal{N}(0, 1)}{\sqrt{\chi_{n-k}^2/(n-k)}} = t_{n-k}.$$

Therefore,

$$I_{1-\alpha, hom} = [\hat{\beta}_j - t_{(1-\frac{\alpha}{2}, n-k)} se_{hom}(\hat{\beta}_j); \hat{\beta}_j + t_{(1-\frac{\alpha}{2}, n-k)} se_{hom}(\hat{\beta}_j)]$$

is an exact confidence interval for  $\beta_j$  under (A1)–(A6).

## 5.4 t-Tests

The **t-statistic** is the OLS estimator standardized with the standard error. Under (A1)–(A4) we have

$$T = \frac{\hat{\beta}_j - \beta_j}{se_{hc}(\hat{\beta}_j)} \xrightarrow{D} \mathcal{N}(0, 1).$$

This result can be used to test the hypothesis  $H_0 : \beta_j = \beta_j^0$ . The t-statistic for this hypothesis is

$$T_0 = \frac{\hat{\beta}_j - \beta_j^0}{se_{hc}(\hat{\beta}_j)},$$

which satisfies  $T_0 = T \xrightarrow{D} \mathcal{N}(0, 1)$  under  $H_0$ .

The **two-sided t-test** for  $H_0$  against  $H_1 : \beta_j \neq \beta_j^0$  is given by the test decision

$$\begin{aligned} \text{do not reject } H_0 & \quad \text{if } |T_0| \leq t_{(1-\frac{\alpha}{2}, n-k)}, \\ \text{reject } H_0 & \quad \text{if } |T_0| > t_{(1-\frac{\alpha}{2}, n-k)}. \end{aligned}$$

The value  $t_{(1-\frac{\alpha}{2}, n-k)}$  is called the **critical value**.

This test is asymptotically of size  $\alpha$ :

$$\lim_{n \rightarrow \infty} P(\text{we reject } H_0 | H_0 \text{ is true}) = \alpha.$$

We can also use the critical value  $z_{(1-\frac{\alpha}{2})}$  instead of  $t_{(1-\frac{\alpha}{2}, n-k)}$  to get an asymptotically valid test of size  $\alpha$ .

If (A5)–(A6) hold, and  $se_{hom}(\hat{\beta}_j)$  is used instead of  $se_{hc}(\hat{\beta}_j)$ , then the t-quantile based t-test is of exact size  $\alpha$ .

**p-values** provide a quick alternative way to make the test decision. The t-test decision rule is equivalent to

$$\begin{aligned} &\text{reject } H_0 && \text{if p-value} < \alpha \\ &\text{do not reject } H_0 && \text{if p-value} \geq \alpha, \end{aligned}$$

where

$$p\text{-value} = 2(1 - F(|T_0|)),$$

and  $F$  is the CDF of  $t_{n-k}$  or  $\mathcal{N}(0, 1)$ , depending on whether the t- or z-quantile critical values are used.

The p-values can be calculated using  $2*(1-pt(abs(T0), n-k))$  and  $2*(1-pnorm(abs(T0), n-k))$ , where  $T_0$  is the t-statistic for  $H_0$ .

```
coeftest(fit, vcov. = vcovHC)
```

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-21.7095175	0.5666566	-38.312	< 2.2e-16 ***
education	3.1350280	0.0408533	76.739	< 2.2e-16 ***
experience	0.2443250	0.0067036	36.447	< 2.2e-16 ***
black	-2.8554345	0.2243222	-12.729	< 2.2e-16 ***
female	-7.4362774	0.1603553	-46.374	< 2.2e-16 ***

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

`coeftest()` is another function from the `lmtest` package and works similarly to `coefci()`. You can specify different standard errors: `coeftest(fit, vcov. = vcovHC, type = "HC1")`. `coeftest(fit)` returns the t-test results for classical standard errors which is identical to the output of the base-R command `summary(fit)`.

To represent very small numbers where there are  $n$  zero digits before the first nonzero digit after the decimal point, R uses scientific notation in the form  $e-n$ . For example,  $2.2e-16$  means  $0.00000000000000022$ .

## 5.5 Joint Testing

When multiple hypotheses are to be tested, repeated t-tests will not yield valid inferences.

Each t-test has a probability of falsely rejecting  $H_0$  (type I error) of  $\alpha$ , but if multiple t-tests are used on different coefficients, then the probability of falsely rejecting at least once (joint type I error probability) is greater than  $\alpha$  (multiple testing problem).



### 5.5.1 Joint Hypotheses

Consider the general hypothesis

$$H_0 : \mathbf{R}\boldsymbol{\beta} = \mathbf{r},$$

where  $\mathbf{R}$  is a  $q \times k$  matrix with  $\text{rank}(\mathbf{R}) = q$  and  $\mathbf{r}$  is a  $q \times 1$  vector.

Let's look at a linear regression with  $k = 3$ :

$$Y_i = \beta_1 + \beta_2 X_{i2} + \beta_3 X_{i3} + u_i$$

- Example 1: The hypothesis  $H_0 : (\beta_2 = 0 \text{ and } \beta_3 = 0)$  implies  $q = 2$  constraints and is translated to  $H_0 : \mathbf{R}\boldsymbol{\beta} = \mathbf{r}$  with

$$\mathbf{R} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{r} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

- Example 2: The hypothesis  $H_0 : \beta_2 + \beta_3 = 1$  implies  $q = 1$  constraint and is translated to  $H_0 : \mathbf{R}\boldsymbol{\beta} = \mathbf{r}$  with

$$\mathbf{R} = (0 \quad 1 \quad 1), \quad \mathbf{r} = (1).$$

In practice, the most common multiple hypothesis tests are tests of whether multiple coefficients are equal to zero, which is a test of whether those regressors should be included in the model.

### 5.5.2 Wald Test

The Wald distance is the vector  $\mathbf{d} = \mathbf{R}\hat{\boldsymbol{\beta}} - \mathbf{r}$ , and the Wald statistic is the squared standardized Wald distance vector:

$$W = (\mathbf{R}\hat{\boldsymbol{\beta}} - \mathbf{r})' (\mathbf{R}\widehat{\mathbf{V}}\mathbf{R}')^{-1} (\mathbf{R}\hat{\boldsymbol{\beta}} - \mathbf{r}).$$

Under  $H_0$  we have

$$W \xrightarrow{D} \chi_q^2.$$

The test decision for the **Wald test**:

$$\begin{aligned} \text{do not reject } H_0 & \quad \text{if } W \leq \chi_{(1-\alpha, q)}^2, \\ \text{reject } H_0 & \quad \text{if } W > \chi_{(1-\alpha, q)}^2, \end{aligned}$$

where  $\chi_{(p, m)}^2$  is the  $p$ -quantile of the chi-squared distribution with  $m$  degrees of freedom.  $\chi_{(p, m)}^2$  can be returned using `qchisq(p, m)`.

### 5.5.3 F-Test

The  $F$  statistic is the Wald statistic scaled by by the number of constraints:

$$F = \frac{W}{q} = \frac{1}{q}(\mathbf{R}\hat{\boldsymbol{\beta}} - \mathbf{r})'(\mathbf{R}\widehat{\mathbf{V}}\mathbf{R}')^{-1}(\mathbf{R}\hat{\boldsymbol{\beta}} - \mathbf{r}).$$

The test decision for the **F-test**:

$$\begin{aligned} \text{do not reject } H_0 & \quad \text{if } F \leq F_{(1-\alpha, q, n-k)}, \\ \text{reject } H_0 & \quad \text{if } F > F_{(1-\alpha, q, n-k)}, \end{aligned}$$

where  $F_{(p, m_1, m_2)}$  is the  $p$ -quantile of the F distribution with  $m_1$  degrees of freedom in the numerator and  $m_2$  degrees of freedom in the denominator.  $F_{(p, m_1, m_2)}$  can be returned using `qf(p, m1, m2)`.

For single constraint ( $q = 1$ ) hypotheses of the form  $H_0 : \beta_j = \beta_j^0$ , the Wald test is equivalent to a t-test using the z-quantile, and the F-test is equivalent to a t-test using the t-quantile.

The Wald and the F-test are asymptotically equivalent and have asymptotic sizes  $\alpha$  under (A1)–(A4) when a HC version of the covariance matrix estimator  $\widehat{\mathbf{V}}$  is used. The  $F$  test is slightly more conservative for small samples.

In the special case of homoscedastic and normal errors (A5)–(A6), the  $F$  test has exact size  $\alpha$  when  $\widehat{\mathbf{V}}_{hom}$  is used, similar to the exact t-test.

### 5.5.4 Testing in R

In our regression from above, we can test whether the two coefficients for `experience` and `female` are both zero. The `waldtest()` function from the `lmtest` package allows you to specify the names of the variables directly.

```
waldtest(fit, c("experience", "female"), vcov = vcovHC)
```

```
Wald test
```

```
Model 1: wage ~ education + experience + black + female
```

```
Model 2: wage ~ education + black
```

```
  Res.Df Df      F    Pr(>F)
1   50737
2   50739  -2 1490.9 < 2.2e-16 ***
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
waldtest(fit, c("experience", "female"), vcov = vcovHC, test = "Chisq")
```

Wald test

Model 1: wage ~ education + experience + black + female

Model 2: wage ~ education + black

	Res.Df	Df	Chisq	Pr(>Chisq)
1	50737			
2	50739	-2	2981.8	< 2.2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

An alternative is to fit a nested model and apply the function to the fitted models. The following command will produce the same output as above:

```
fit2 = lm(wage ~ education + black, data = cps)
waldtest(fit, fit2, vcov = vcovHC)
```

User-specified constraints of the general form  $R\beta = r$  can be tested with the `linearHypothesis()` function from the `car` package.

## 5.6 R-codes

[methods-sec05.R](#)

## 6 Case Study I: Score Data

```
library(AER) # for the dataset
library(sandwich) # robust standard errors
library(lmtest) # robust inference
library(stargazer) # regression outputs
library(tidyverse) # data management
```

### 6.1 Data Set Description

The California School data set (CASchools) is included in the R package `AER`. This dataset contains information on various characteristics of schools in California, such as test scores, teacher salaries, and student demographics.

```
# load the the data set
data(CASchools)
# get an overview
summary(CASchools)
```

Upon examination we find that the dataset contains mostly numeric variables, but it lacks two important ones we're interested in: **average test scores** and **student-teacher ratios**. However, we can calculate them using the available data.

To find the student-teacher ratio, we divide the total number of students by the number of teachers. For the average test score, we just need to average the math and reading scores. In the next code chunk, we'll demonstrate how to create these variables as vectors and add them to the `CASchools` dataset.

```
# compute student-teacher ratio and append it to CASchools
CASchools$STR <- CASchools$students/CASchools$teachers

# compute test score and append it to CASchools
CASchools$score <- (CASchools$read + CASchools$math)/2
```

If we ran `summary(CASchools)` again we would find the two variables of interest as additional variables named `STR` and `score`.

## 6.2 Linear Regression

Let's suppose we were interested in the following regression model

$$TestScore = \beta_0 + \beta_1 STR + \beta_2 english + u$$

In this regression, we aim to explore how test scores (`score`) are influenced by student-teacher ratio (`STR`) and the percentage of English learners (`english`). The variable `english` indicates the proportion of students who may require additional support or resources to improve their English language skills within each school.

We would run this model in R using the `lm()` function and explore the regression estimates with `coefest()`.

```
# run the model
model <- lm(score ~ STR + english, data = CASchools)
# report estimates
coefest(model, vcov. = vcovHC)
```

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	686.032245	8.812242	77.8499	< 2e-16 ***
STR	-1.101296	0.437066	-2.5197	0.01212 *
english	-0.649777	0.031297	-20.7617	< 2e-16 ***

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The `coefest()` function in R, along with suitable options such as `vcov. = vcovHC` for robust standard errors, automatically includes statistics such as standard errors,  $t$ -statistics, and  $p$ -values, which is exactly what we need to test hypotheses about single coefficients ( $\beta_j$ ) in regression models.

We can also compute confidence intervals for individual coefficients in the multiple regression model by using the function `coefci()`. This function computes confidence intervals at the 95% level by default.

```
# compute confidence intervals for all coefficients in the model
coefci(model, vcov. = vcovHC)
```

	2.5 %	97.5 %
(Intercept)	668.7102930	703.3541961
STR	-1.9604231	-0.2421682
english	-0.7112962	-0.5882574

To obtain confidence intervals at a different level, say 90%, we set the argument `level` in our call of `coefci()` accordingly.

```
coefci(model, vcov. = vcovHC, level = 0.9)
```

	5 %	95 %
(Intercept)	671.5051238	700.5593652
STR	-1.8218062	-0.3807851
english	-0.7013703	-0.5981834

The output above shows that zero is not an element of the confidence interval for the coefficient on `STR`, so we can reject the null hypothesis at significance levels of 5% and 10% (Note that rejection at the 5% level implies rejection at the 10% level anyway).

We can bring this conclusion further via the  $p$ -value for `STR`:  $0.01 < 0.01212 < 0.05$ , which indicates that this coefficient estimate is significant at the 5% level but not at the 1% level.

### 6.3 Bad Controls

Let's suppose now that we are interested in investigating the average effect on test scores of reducing the student-teacher ratio when the expenditures per pupil and the percentage of english learning pupils are held constant.

Let us augment our model by an additional regressor `expenditure`, that is a measure for the total expenditure per pupil in the district. For this model, we will include `expenditure` as measured in thousands of dollars. Our new model would be

$$TestScore = \beta_0 + \beta_1 STR + \beta_2 english + \beta_3 expenditure + u$$

Let us now estimate the model:

```
# scale expenditure to thousands of dollars
CASchools$expenditure <- CASchools$expenditure/1000

# estimate the model
model <- lm(score ~ STR + english + expenditure, data = CASchools)
coeftest(model, vcov. = vcovHC)
```

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	649.577947	15.668623	41.4572	< 2e-16 ***
STR	-0.286399	0.487513	-0.5875	0.55721
english	-0.656023	0.032114	-20.4278	< 2e-16 ***
expenditure	3.867901	1.607407	2.4063	0.01655 *

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The estimated impact of a one-unit change in the student-teacher ratio on test scores, while holding expenditure and the proportion of English learners constant, is  $-0.29$ . It is much smaller than the estimated coefficient in our initial model where we didn't include expenditure.

Additionally, this coefficient of *STR* is no longer statistically significant, even at a 10% significance level, as indicated by a  $p$ -value of 0.56. This lack of significance for  $\beta_1$  may stem from a larger standard error resulting from the inclusion of expenditure in the model, leading to less precise estimation of the coefficient on *STR*. This scenario highlights the challenge of dealing with strongly correlated predictors.

Note that *expenditure* can be classified as a **bad control** because higher expenditure per pupil may be the cause of a decrease in the student-teacher ratio. By adding *expenditure* to the regression we are controlling away our causal effect of *STR* on *score*.

The correlation between *STR* and *expenditure* can be determined using the `cor()` function.

```
# compute the sample correlation between 'STR' and 'expenditure'  
cor(CASchools$STR, CASchools$expenditure)
```

```
[1] -0.6199822
```

This indicates a moderately strong negative correlation between the two variables.

The estimated model is

$$\widehat{TestScore} = 649.58 - 0.29 STR - 0.66 english + 3.87 expenditure$$

(15.67)      (0.49)      (0.03)      (1.61)

Could we reject the hypothesis that *both* the *STR* coefficient and the *expenditure* coefficient are zero? To answer this, we need to conduct **joint hypothesis tests**, which involve placing restrictions on multiple regression coefficients. This differs from individual  $t$ -tests, where restrictions are applied to a single coefficient.

To test whether both coefficients are zero, we will conduct a heteroskedasticity-robust  $F$ -test. To do this in R, we can use the function `waldtest()` contained in the package `lmtest`.

```
waldtest(model, c("STR", "expenditure"), vcov = vcovHC)
```

Wald test

Model 1: `score ~ STR + english + expenditure`

Model 2: `score ~ english`

	Res.Df	Df	F	Pr(>F)	
1	416				
2	418	-2	5.2617	0.005537	**

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The output reveals that the  $F$ -statistic for this joint hypothesis test is 5.26 and the corresponding  $p$ -value is about 0.0055. We can therefore reject the null hypothesis that both coefficients are zero at the 1% level of significance. Notice that the individual  $t$ -tests for `STR` and `expenditure` are insignificant at the 1% level.

## 6.4 Good Controls

In order to reduce the risk of omitted variable bias, it is essential to include control variables in regression models. In our case, we are interested in estimating the causal effect of a change in the student-teacher ratio on test scores.

By including `english` as control variable, we aimed to control for unobservable student characteristics which correlate with the student-teacher ratio and are assumed to have an impact on test score. Including `expenditure` was actually not a good idea because it is highly correlated with `STR` (imperfect multicollinearity) and may be the cause of the student-teacher ratio (bad control).

There are other interesting control variables to observe:

- `lunch`: the share of students that qualify for a subsidized or even a free lunch at school.
- `calworks`: the percentage of students that qualify for the *CalWorks* income assistance program.

Students eligible for *CalWorks* live in families with a total income below the threshold for the subsidized lunch program, so both variables are indicators for the share of economically disadvantaged children. We suspect both indicators are highly correlated.



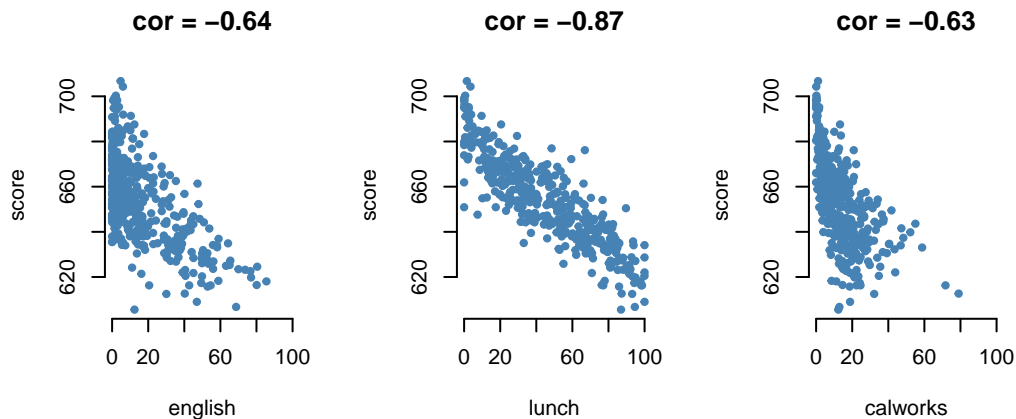
```
# estimate the correlation between 'calworks' and 'lunch'  
cor(CASchools$calworks, CASchools$lunch)
```

```
[1] 0.7394218
```

If they are highly correlated as we just confirmed, there is no standard way to proceed when deciding which variable to use. It may not be a good idea to use both variables as regressors in view of collinearity, but as long as we are only interested in the coefficient of **STR** we do not care whether the coefficients of **calworks** and **lunch** have an imperfect multicollinearity problem.

Let's first explore further these control variables and how they correlate with the dependent variable by plotting them against test scores.

```
correlations = round(cor(CASchools$score, CASchools |>  
                        select(english, lunch, calworks)),2)  
par(mfrow = c(1,3), pch = 20, col = "steelblue", bty="n")  
plot(score ~ english, data = CASchools, xlim = c(0, 100),  
     main = paste("cor =", correlations[1]))  
plot(score ~ lunch, data = CASchools, xlim = c(0, 100),  
     main = paste("cor =", correlations[2]))  
plot(score ~ calworks, data = CASchools, xlim = c(0, 100),  
     main = paste("cor =", correlations[3]))
```



We shall consider five different model equations:

$$\text{TestScore} = \beta_0 + \beta_1 \text{STR} + u, \quad (6.1)$$

$$\text{TestScore} = \beta_0 + \beta_1 \text{STR} + \beta_2 \text{english} + u, \quad (6.2)$$

$$\text{TestScore} = \beta_0 + \beta_1 \text{STR} + \beta_2 \text{english} + \beta_3 \text{lunch} + u, \quad (6.3)$$

$$\text{TestScore} = \beta_0 + \beta_1 \text{STR} + \beta_2 \text{english} + \beta_4 \text{calworks} + u, \quad (6.4)$$

$$\text{TestScore} = \beta_0 + \beta_1 \text{STR} + \beta_2 \text{english} + \beta_3 \text{lunch} + \beta_4 \text{calworks} + u. \quad (6.5)$$

The best way to report regression results is in a table. The `stargazer` package is very convenient for this purpose. It provides a function that generates professionally looking HTML and LaTeX tables that satisfy scientific standards. One simply has to provide one or multiple object(s) of class `lm`. The rest is done by the function `stargazer()`.

```
# estimate different model specifications
spec1 <- lm(score ~ STR, data = CASchools)
spec2 <- lm(score ~ STR + english, data = CASchools)
spec3 <- lm(score ~ STR + english + lunch, data = CASchools)
spec4 <- lm(score ~ STR + english + calworks, data = CASchools)
spec5 <- lm(score ~ STR + english + lunch + calworks, data = CASchools)

# gather robust standard errors in a list
rob_se <- list(sqrt(diag(vcovHC(spec1))),
               sqrt(diag(vcovHC(spec2))),
               sqrt(diag(vcovHC(spec3))),
               sqrt(diag(vcovHC(spec4))),
               sqrt(diag(vcovHC(spec5))))

stargazer(spec1, spec2, spec3, spec4, spec5,
           font.size = "footnotesize",
           se = rob_se,
           type="latex",
           omit.stat = "f", header = FALSE)
```

Each column in this table contains most of the information provided also by `coefTest()` and `summary()` for each of the models under consideration. Each of the coefficient estimates includes its standard error in parenthesis and one, two or three asterisks representing their significance levels (10% , 5% and 1%). Although  $t$ -statistics are not reported, one may compute them manually simply by dividing a coefficient estimate by the corresponding standard error. At the bottom of the table summary statistics for each model and a legend are reported.

From the model comparison we observe that including control variables approximately cuts the coefficient on *STR* in half. Additionally, the estimation seems to remain unaffected by the

Table 6.1

	<i>Dependent variable:</i>				
	score				
	(1)	(2)	(3)	(4)	(5)
STR	-2.280*** (0.524)	-1.101** (0.437)	-0.998*** (0.274)	-1.308*** (0.343)	-1.014*** (0.273)
english		-0.650*** (0.031)	-0.122*** (0.033)	-0.488*** (0.030)	-0.130*** (0.037)
lunch			-0.547*** (0.024)		-0.529*** (0.039)
calworks				-0.790*** (0.070)	-0.048 (0.062)
Constant	698.933*** (10.461)	686.032*** (8.812)	700.150*** (5.641)	697.999*** (7.006)	700.392*** (5.615)
Observations	420	420	420	420	420
R <sup>2</sup>	0.051	0.426	0.775	0.629	0.775
Adjusted R <sup>2</sup>	0.049	0.424	0.773	0.626	0.773
Residual Std. Error	18.581 (df = 418)	14.464 (df = 417)	9.080 (df = 416)	11.654 (df = 416)	9.084 (df = 415)

Note:

\*p<0.1; \*\*p<0.05; \*\*\*p<0.01

specific set of control variables employed. Thus, the inference drawn is that, under all other conditions held constant, reducing the student-teacher ratio by one unit is associated with an estimated average rise in test scores of roughly 1 point.

Incorporating student characteristics as controls increased both  $R^2$  and  $\bar{R}^2$  from about 0.05 (spec1) to about 0.77 (spec3 and spec5), indicating these variables' suitability as predictors for test scores.

We also observe that the coefficients for some of the control variables are not significant in some models. For example in spec5, the coefficient on *calworks* is not significantly different from zero at the 10% level.

Lastly, we see that the effect on the estimate (and its standard error) of the coefficient on *STR* when adding *calworks* to the base specification spec3 is minimal. Hence, we can identify *calworks* as an unnecessary control variable, especially considering the incorporation of *lunch* in this model.

## 6.5 Nonlinear Specifications

Sometimes a nonlinear regression function is better suited for estimating a population relationship. Let's have a look at an example that explores the relationship between the income of schooling districts and their test scores.

We start our analysis by computing the correlation between both variables.

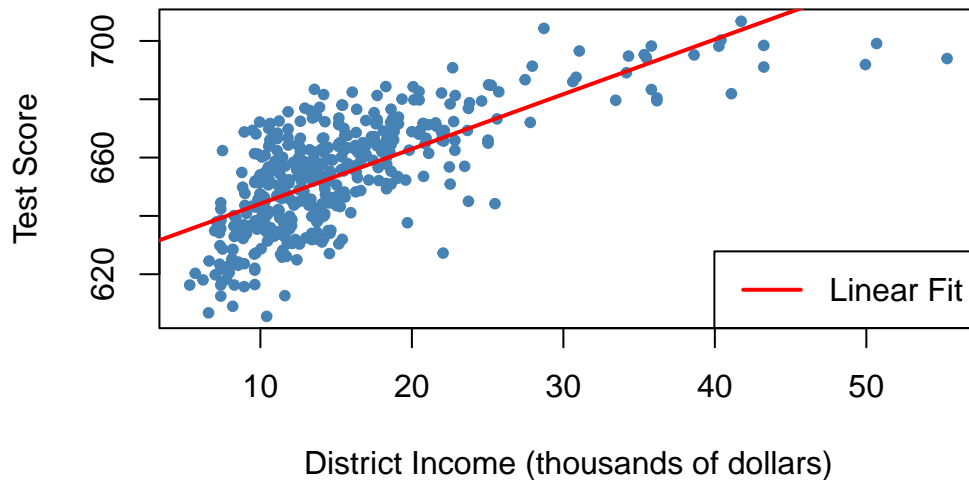
```
cor(CASchools$income, CASchools$score)
```

```
[1] 0.7124308
```

Income and test score are positively correlated: school districts with above-average income tend to achieve above-average test scores. But does a linear regression adequately model the data? To investigate this further, let's visualize the data by plotting it and adding a linear regression line.

```
# Fit a simple linear model and plot observations with the regression line
linear_model <- lm(score ~ income, data = CASchools)
plot(CASchools$income, CASchools$score, col = "steelblue", pch = 20,
      xlab = "District Income (thousands of dollars)", ylab = "Test Score",
      main = "Test Score vs. District Income and a Linear OLS Regression Function")
abline(linear_model, col = "red", lwd = 2) # Add regression line
legend("bottomright", "Linear Fit", col = "red", lwd = 2) # Add legend
```

## Test Score vs. District Income and a Linear OLS Regression Fit



The plot shows that the linear regression line seems to overestimate the true relationship when income is either very high or very low and it tends to underestimate it for the middle income group. Luckily, Ordinary Least Squares (OLS) isn't limited to linear regressions of the predictors. We have the flexibility to model test scores as a function of income and the square of income.

This leads us to the following regression model:

$$TestScore_i = \beta_0 + \beta_1 income_i + \beta_2 income_i^2 + u_i$$

which is a *quadratic regression model*. Here we treat  $income^2$  as an additional explanatory variable.

```
# fit the quadratic Model
quadratic_model <- lm(score ~ income + I(income^2), data = CASchools)

# obtain the model summary
coeftest(quadratic_model, vcov. = vcovHC)
```

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	607.3017435	2.9242237	207.6796	< 2.2e-16 ***
income	3.8509939	0.2711045	14.2048	< 2.2e-16 ***

```
I(income^2)  -0.0423084    0.0048809   -8.6681 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated function is

$$\widehat{TestScore} = 607.3 + 3.85 \text{ income}_i - 0.0423 \text{ income}_i^2$$

(2.93)
(0.27)
(0.00489)

We will now draw the same scatter plot as for the linear model and add the regression line for the quadratic model. Since `abline()` only plots straight lines, it cannot be used here, but we can use `lines()` function instead, which is suitable for plotting nonstraight lines (see `?lines`). The most basic call of `lines()` is `lines(x_values, y_values)` where `x_values` and `y_values` are vectors of the same length that provide coordinates of the points to be sequentially connected by a line.

This requires sorted coordinate pairs according to the X-values. We may use the function `order()` to sort the fitted values of score according to the observations of income, obtained from our quadratic model.

```
# Plot observations and add linear and quadratic regression lines
plot(CASchools$income, CASchools$score, col="steelblue", pch=20,
     xlab="District Income (thousands of dollars)", ylab="Test Score",
     main="Estimated Linear and Quadratic Regression Functions")
# Linear regression line
abline(linear_model, col="green", lwd=2)
# Quadratic regression line
lines(CASchools$income[order(CASchools$income)],
     fitted(quadratic_model)[order(CASchools$income)], col="red", lwd=2)
legend("bottomright", c("Quadratic Fit", "Linear Fit"), lwd=2, col=c("red", "green"))
```

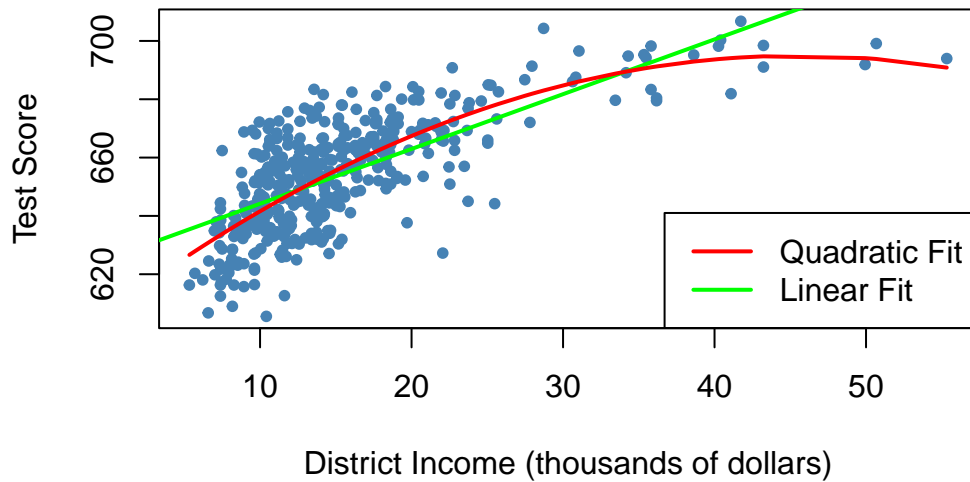
As the plot shows, the quadratic function appears to provide a better fit to the data compared to the linear function.

Another approach to estimate a concave nonlinear regression function involves using a logarithmic regressor.

```
# estimate a level-log model
LinearLog_model <- lm(score ~ log(income), data = CASchools)

# compute robust summary
coeftest(LinearLog_model, vcov = vcovHC)
```

## Estimated Linear and Quadratic Regression Functions



t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	557.8323	3.8622	144.433	< 2.2e-16 ***
log(income)	36.4197	1.4058	25.906	< 2.2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The estimated regression model is

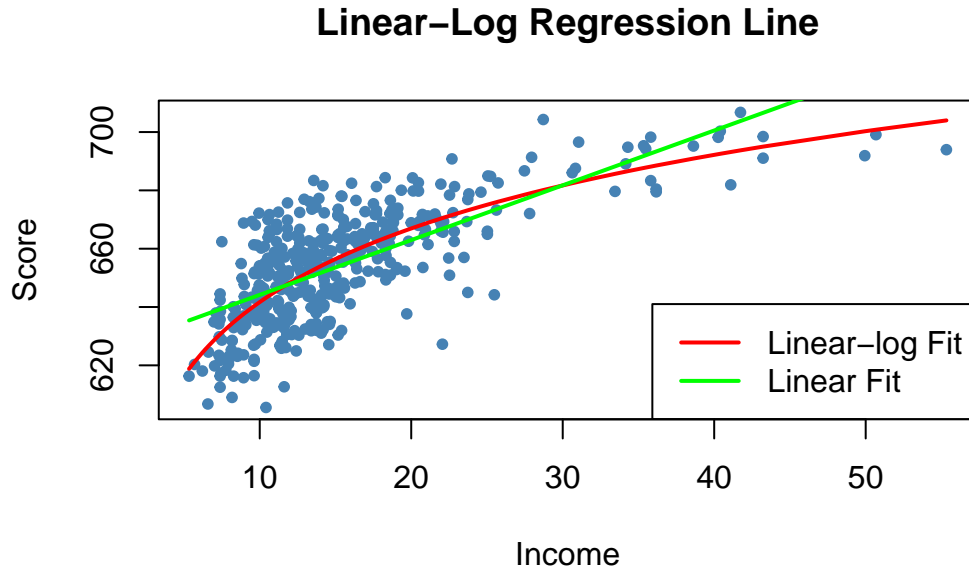
$$\widehat{TestScore} = 557.8 + 36.42 \log(income)$$

(3.86)      (1.41)

We plot this function

```
# Draw a scatterplot with linear and linear-log regression lines
plot(score ~ income, data = CASchools, col = "steelblue", pch = 20,
      ylab="Score", xlab="Income", main = "Linear-Log Regression Line")
order_id <- order(CASchools$income)
# Linear-log regression line
lines(CASchools$income[order_id], fitted(LinearLog_model)[order_id],
      col = "red", lwd = 2)
# Linear regression line
lines(CASchools$income[order_id], fitted(linear_model)[order_id],
```

```
col = "green", lwd = 2)
legend("bottomright", c("Linear-log Fit", "Linear Fit"),
      lwd = 2, col = c("red", "green"))
```



We can interpret  $\hat{\beta}_1$  as follows: a 1% increase in income is associated with an average increase in test scores of  $0.01 \cdot 36.42 = 0.36$  points.

## 6.6 Interactions

Sometimes it is interesting to learn how the effect on  $Y$  of a change in an independent variable depends on the value of another independent variable.

For example, we may ask if districts with many English learners benefit differently from a decrease in the student-teacher ratio compared to those with fewer English learning students. We can assess this by using a multiple regression model and including an interaction term.

We consider three cases: when both independent variables are binary, when one is binary and the other is continuous, and when both are continuous.

### 6.6.1 Two Binary Variables

Let

$$HiSTR = \begin{cases} 1, & \text{if } STR \geq 20, \\ 0, & \text{else,} \end{cases} \quad HiEL = \begin{cases} 1, & \text{if english} \geq 10, \\ 0, & \text{else.} \end{cases}$$



In R, we construct these dummies as follows

```
# append HiSTR to CASchools
CASchools$HiSTR <- as.numeric(CASchools$STR >= 20)

# append HiEL to CASchools
CASchools$HiEL <- as.numeric(CASchools$english >= 10)
```

We now estimate the model

$$TestScore = \beta_0 + \beta_1 HiSTR + \beta_2 HiEL + \beta_3 HiSTR \cdot HiEL + u_i.$$

We can simply indicate `HiEL * HiSTR` inside the `lm()` formula to add the interaction term to the model. Note that this adds `HiEL`, `HiSTR` and their interaction as regressors, whereas indicating `HiEL:HiSTR` only adds the interaction term.

```
# estimate the model with a binary interaction term
bi_model <- lm(score ~ HiSTR * HiEL, data = CASchools)

# print a robust summary of the coefficients
coeftest(bi_model, vcov. = vcovHC)
```

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	664.1433	1.3908	477.5272	< 2.2e-16 ***
HiSTR	-1.9078	1.9416	-0.9826	0.3264
HiEL	-18.3155	2.3453	-7.8094	4.721e-14 ***
HiSTR:HiEL	-3.2601	3.1360	-1.0396	0.2991

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The estimated regression model is

$$\widehat{TestScore} = 664.1 - 1.9 HiSTR - 18.3 HiEL - 3.3 (HiSTR \cdot HiEL)$$

(1.39)      (1.94)                      (2.35)                      (3.14)

According to this model, when moving from a school district with a low student-teacher ratio to one with a high ratio, the average effect on test scores depends on the percentage of English learners (`HiEL`), and can be computed as  $-1.9 - 3.3 \cdot HiEL$ .

This is, for districts with fewer English learners ( $HiEL = 0$ ), the expected decrease in test scores is 1.9 points. However, for districts with a higher proportion of English learners ( $HiEL = 1$ ), the predicted decrease in test scores is  $1.9 + 3.3 = 5.2$  points.

We can estimate the mean test score conditional on all possible combination of the included binary variables

$HiSTR$	$HiEL$	$E[score HiSTR, HiEL]$	$\widehat{score}$
0	0	$\beta_0$	664.1
0	1	$\beta_0 + \beta_2$	645.8
1	0	$\beta_0 + \beta_1$	662.2
1	1	$\beta_0 + \beta_1 + \beta_2 + \beta_3$	640.6

### 6.6.2 Continuous and Binary Variables

This specification where the interaction term includes a continuous variable ( $X_i$ ) and a binary variable ( $D_i$ ) allows for the slope to depend on the binary variable. There are three different possibilities:

1. Different intercepts, same slope:

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 D_i + u_i$$

2. Different intercepts and slopes:

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 D_i + \beta_3 (X_i \cdot D_i) + u_i$$

3. Same intercept, different slopes:

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 (X_i \cdot D_i) + u_i.$$

Does the effect on test scores of cutting the student-teacher ratio depend on whether the percentage of students still learning English is high or low?

One way to answer this question is to use a specification that allows for two different regression lines, depending on whether there is a high or a low percentage of English learners. This is achieved using the different intercept/different slope specification. We estimate the regression model

$$\widehat{TestScore}_i = \beta_0 + \beta_1 STR_i + \beta_2 HiEL_i + \beta_3 (STR_i \cdot HiEL_i) + u_i$$

```
# estimate the model
bci_model <- lm(score ~ STR + HiEL + STR * HiEL, data = CASchools)

# print robust summary of coefficients
coeftest(bci_model, vcov. = vcovHC)
```

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	682.24584	12.07126	56.5182	<2e-16 ***
STR	-0.96846	0.59943	-1.6156	0.1069
HiEL	5.63914	19.88866	0.2835	0.7769
STR:HiEL	-1.27661	0.98557	-1.2953	0.1959

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

$$\widehat{TestScore} = 682.2 - 0.97 STR + 5.6 HiEL - 1.28 (STR \cdot HiEL).$$

(12.07)
(0.60)
(19.89)
(0.99)

The estimated regression line for districts with a low fraction of English learners ( $HiEL = 0$ ) is

$$\widehat{TestScore} = 682.2 - 0.97 STR_i$$

while the one for districts with a high fraction of English learners ( $HiEL = 1$ ) is

$$\begin{aligned} \widehat{TestScore} &= 682.2 + 5.6 - 0.97 STR_i - 1.28 STR_i \\ &= 687.8 - 2.25 STR_i. \end{aligned}$$

The expected rise in test scores after decreasing the student-teacher ratio by one unit is roughly 0.97 points in districts with a low proportion of English learners, but 2.25 points in districts with a high concentration of English learners.

The coefficient on the interaction term, “ $STR \cdot HiEL$ ”, indicates that the contrast between these effects amounts to 1.28 points.

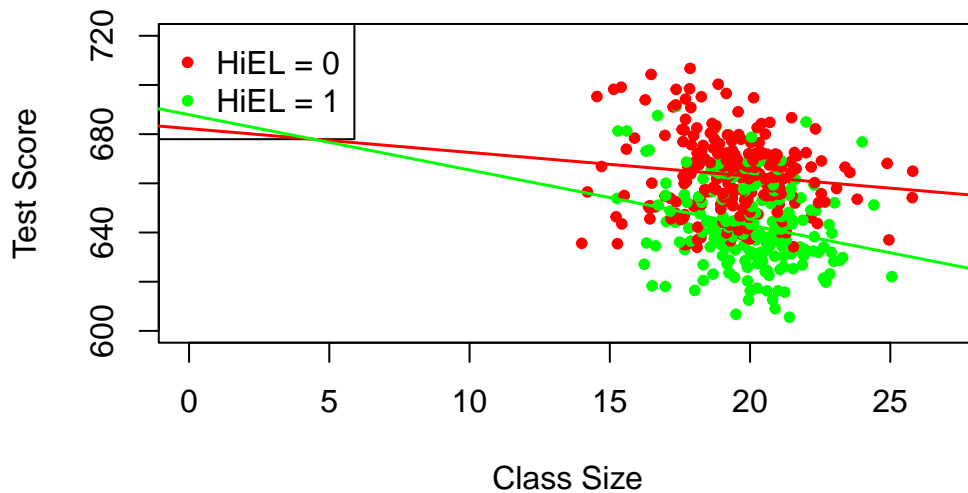
We now plot both regression lines from the model by using different colors to differentiate each of the  $STR$  levels.

```

# Determine observations with English learners >= 10%
id <- CASchools$english >= 10

# Plot observations with different colors for HiEL status and draw regression lines
plot(CASchools$STR, CASchools$score, xlim = c(0, 27), ylim = c(600, 720), pch = 20,
     col = ifelse(id, "green", "red"), xlab = "Class Size", ylab = "Test Score")
legend("topleft", pch = 20, col = c("red", "green"), legend = c("HiEL = 0", "HiEL = 1"))
abline(coef = c(bci_model$coefficients[1], bci_model$coefficients[2]),
       col = "red", lwd = 1.5)
abline(coef = c(bci_model$coefficients[1] + bci_model$coefficients[3],
               bci_model$coefficients[2] + bci_model$coefficients[4]),
       col = "green", lwd = 1.5)

```



### 6.6.3 Two Continuous Variables

Let's now examine the interaction between the continuous variables student-teacher ratio (*STR*) and the percentage of English learners (*english*).

```

# estimate regression model including the interaction between 'english' and 'STR'
cci_model <- lm(score ~ STR + english + english * STR, data = CASchools)

# print summary
coeftest(cci_model, vcov. = vcovHC)

```

t test of coefficients:

```

                Estimate Std. Error t value Pr(>|t|)
(Intercept) 686.3385268  11.9378561  57.4926 < 2e-16 ***
STR          -1.1170184   0.5965151  -1.8726  0.06183 .
english      -0.6729119   0.3865378  -1.7409  0.08245 .
STR:english   0.0011618    0.0191576   0.0606  0.95167
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

The estimated regression function is

$$\widehat{TestScore} = 686.3 - 1.12 STR - 0.67 english + 0.0012 (STR \cdot english).$$

(11.94)
(0.60)
(0.39)
(0.02)

Before proceeding with the interpretations, let us explore the quartiles of *english*

```
summary(CASchools$english)
```

```

    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.000   1.941   8.778  15.768  22.970  85.540

```

When the percentage of English learners is at the median (*english* = 8.778), the slope of the line is estimated to be  $(-1.12 + 0.0012 \cdot 8.778 = -1.12)$ . When the percentage of English learners is at the 75th percentile (*english* = 22.97), this line is estimated to be slightly flatter, with a slope of  $-1.12 + 0.0012 \cdot 22.97 = -1.09$ .

In other words, for a district with 8.78% English learners, the estimated effect of a one-unit reduction in the student-teacher ratio is to increase on average test scores by 1.11 points, but for a district with 23% English learners, reducing the student-teacher ratio by one unit is predicted to increase test scores on average by 1.09 points.

However, it is important to note from the output of `coefTest()` that the estimated coefficient on the interaction term ( $\beta_3$ ) is not statistically significant at the 10% level, so we cannot reject the null hypothesis  $H_0 : \beta_3 = 0$ .

## 6.7 Nonlinearities in Score Regressions

This section examines three key questions about test scores and the student-teacher ratio.

- First, it explores if reducing the student-teacher ratio affects test scores differently based on the number of English learners, even when considering economic differences across districts.

- Second, it investigates if this effect varies depending on the student-teacher ratio.
- Lastly, it aims to determine the expected impact on test scores when the student-teacher ratio decreases by two students per teacher, considering both economic factors and potential nonlinear relationships.

We will answer these questions considering the previously explained nonlinear regression specifications, extended to include two measures of the economic background of the students: the percentage of students eligible for a subsidized lunch (*lunch*) and the logarithm of average district income (*log(income)*).

The logarithm of district income is used following our previous empirical analysis, which suggested that this specification captures the nonlinear relationship between scores and income.

We leave out the expenditure per pupil (*expenditure*) from our analysis because including it would suggest that spending changes with the student-teacher ratio (in other words, we would not be holding expenditures per pupil constant).

We will consider 7 different model specifications:

```
# estimate all models
TS_mod1 <- lm(score ~ STR + english + lunch, data = CASchools)
TS_mod2 <- lm(score ~ STR + english + lunch + log(income), data = CASchools)
TS_mod3 <- lm(score ~ STR + HiEL + HiEL:STR, data = CASchools)
TS_mod4 <- lm(score ~ STR + HiEL + HiEL:STR + lunch + log(income), data = CASchools)
TS_mod5 <- lm(score ~ STR + I(STR^2) + I(STR^3) + HiEL + lunch + log(income),
              data = CASchools)
TS_mod6 <- lm(score ~ STR + I(STR^2) + I(STR^3) + HiEL + HiEL:STR + HiEL:I(STR^2) + HiEL:I(S
              + lunch + log(income), data = CASchools)
TS_mod7 <- lm(score ~ STR + I(STR^2) + I(STR^3) + english + lunch + log(income),
              data = CASchools)

# gather robust standard errors in a list
rob_se <- list(sqrt(diag(vcovHC(TS_mod1))),
              sqrt(diag(vcovHC(TS_mod2))),
              sqrt(diag(vcovHC(TS_mod3))),
              sqrt(diag(vcovHC(TS_mod4))),
              sqrt(diag(vcovHC(TS_mod5))),
              sqrt(diag(vcovHC(TS_mod6))),
              sqrt(diag(vcovHC(TS_mod7))))

stargazer(TS_mod1, TS_mod2, TS_mod3, TS_mod4,
          TS_mod5, TS_mod6, TS_mod7,
          font.size = "footnotesize",
```

```

se = rob_se,
type="latex",
omit.stat = "f", df=FALSE, header = FALSE)

```

Table 6.3

	<i>Dependent variable:</i>						
	(1)	(2)	(3)	score			(7)
STR	-0.998*** (0.274)	-0.734*** (0.261)	-0.968 (0.599)	-0.531 (0.350)	64.339** (27.295)	83.702*** (31.506)	65.285** (27.708)
english	-0.122*** (0.033)	-0.176*** (0.034)					-0.166*** (0.035)
I(STR <sup>2</sup> )					-3.424** (1.373)	-4.381*** (1.597)	-3.466** (1.395)
I(STR <sup>3</sup> )					0.059*** (0.023)	0.075*** (0.027)	0.060*** (0.023)
lunch	-0.547*** (0.024)	-0.398*** (0.034)		-0.411*** (0.029)	-0.420*** (0.029)	-0.418*** (0.029)	-0.402*** (0.034)
log(income)		11.569*** (1.841)		12.124*** (1.823)	11.748*** (1.799)	11.800*** (1.809)	11.509*** (1.834)
HiEL			5.639 (19.889)	5.498 (10.012)	-5.474*** (1.046)	816.076** (354.100)	
STR:HiEL			-1.277 (0.986)	-0.578 (0.507)		-123.282** (54.290)	
I(STR <sup>2</sup> ):HiEL						6.121** (2.752)	
I(STR <sup>3</sup> ):HiEL						-0.101** (0.046)	
Constant	700.150*** (5.641)	658.552*** (8.749)	682.246*** (12.071)	653.666*** (10.053)	252.050 (179.724)	122.353 (205.050)	244.809 (181.899)
Observations	420	420	420	420	420	420	420
R <sup>2</sup>	0.775	0.796	0.310	0.797	0.801	0.803	0.801
Adjusted R <sup>2</sup>	0.773	0.794	0.305	0.795	0.798	0.799	0.798
Residual Std. Error	9.080	8.643	15.880	8.629	8.559	8.547	8.568

Note:

\*p<0.1; \*\*p<0.05; \*\*\*p<0.01

What can be concluded from the results presented?

- First, we see the estimated coefficient on *STR* is highly significant in all models except from specifications (3) and (4).
- When we add  $\log(\text{income})$  to model (1) in the second specification, all coefficients remain highly significant while the coefficient on the new regressor is also statistically significant at the 1% level. Additionally, the coefficient on *STR* is now 0.27 higher than in model (1), suggesting a possible mitigation of omitted variable bias when including  $\log(\text{income})$  as regressor. For these reasons, it makes sense to keep this variable in other models too.
- Models (3) and (4) include the interaction term between *STR* and *HiEL*, first without control variables in the third specification and then controlling for economic factors in the fourth. The estimated coefficient for the interaction term is not significant at any common level in any of these models, nor is the coefficient on the dummy variable *HiEL*. Hence, despite accounting for economic factors, we cannot reject the null hypotheses that the impact of the student-teacher ratio on test scores remains consistent across districts with high and low proportions of English learning students.
- In regression (5) we have included quadratic and cubic terms for *STR*, while omitting the interaction term between *STR* and *HiEL*, since it was not significant in specification (4). The results indicate high levels of significance for these estimated coefficients and we can therefore assume the presence of a nonlinear effect of the student-teacher ratio on test scores. This could be also verified with an *F*-test of  $H_0 : \beta_2 = \beta_3 = 0$ .
- Regression (6) further examines whether the proportion of English learners influences the student-teacher ratio, incorporating the interaction terms  $\text{HiEL} \cdot \text{STR}$ ,  $\text{HiEL} \cdot \text{STR}^2$  and  $\text{HiEL} \cdot \text{STR}^3$ . Each individual *t*-test confirms significant effects. To validate this, we perform a robust *F*-test to assess  $H_0 : \beta_5 = \beta_6 = \beta_7 = 0$ .

```
# check joint significance of the interaction terms
waldtest(TS_mod6,
         c("STR:HiEL", "I(STR^2):HiEL", "I(STR^3):HiEL"),
         vcov = vcovHC)
```

Wald test

```
Model 1: score ~ STR + I(STR^2) + I(STR^3) + HiEL + HiEL:STR + HiEL:I(STR^2) +
         HiEL:I(STR^3) + lunch + log(income)
```

```
Model 2: score ~ STR + I(STR^2) + I(STR^3) + HiEL + lunch + log(income)
```

```
Res.Df Df      F Pr(>F)
1      410
2      413 -3 2.1885 0.08882 .
```

---

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



- With a  $p$ -value of 0.08882 we can just reject the null hypothesis at the 10% level. This provides only weak evidence that the regression functions are different for districts with high and low percentages of English learners.
- In model (7), we employ a continuous measure for the proportion of English learners instead of a dummy variable (thus omitting interaction terms). We note minimal alterations in the coefficient estimates for the remaining regressors. Consequently, we infer that the findings observed in model (5) are robust and not influenced significantly by the method used to measure the percentage of English learners.

We can now address the initial questions raised in this section:

- First, in the linear models, the impact of the percentage of English learners on changes in test scores due to variations in the student-teacher ratio is minimal, a conclusion that holds true even after accounting for students' economic backgrounds. Although the cubic specification (6) suggests that the relationship between student-teacher ratio and test scores is influenced by the proportion of English learners, the magnitude of this influence is not significant.
- Second, while controlling for students' economic backgrounds, we identify nonlinearities in the association between student-teacher ratio and test scores.
- Lastly, under the **linear specification** (2), a reduction of two students per teacher in the student-teacher ratio is projected to increase test scores by approximately 1.46 points. As this model is linear, this effect remains consistent regardless of class size. For instance, assuming a student-teacher ratio of 20, the **nonlinear model** (5) indicates that the reduction in student-teacher ratio would lead to an increase in test scores by

$$\begin{aligned}
& 64.33 \cdot 18 + 18^2 \cdot (-3.42) + 18^3 \cdot (0.059) \\
& - (64.33 \cdot 20 + 20^2 \cdot (-3.42) + 20^3 \cdot (0.059)) \\
& \approx 3.3
\end{aligned}$$

points. If the ratio was 22, a reduction to 20 leads to a predicted improvement in test scores of

$$\begin{aligned}
& 64.33 \cdot 20 + 20^2 \cdot (-3.42) + 20^3 \cdot (0.059) \\
& - (64.33 \cdot 22 + 22^2 \cdot (-3.42) + 22^3 \cdot (0.059)) \\
& \approx 2.4
\end{aligned}$$

points. This suggests that the effect is more evident in smaller classes.

## 6.8 R-codes

[methods-sec06.R](#)

## 7 Regression Diagnostics

This section discusses some graphical and analytical regression diagnostic techniques for detecting outliers and assessing whether the assumptions of our regression model are met.

### 7.1 Leverage values

Leverage values  $h_{ii}$  indicate how much influence an observation  $\mathbf{X}_i$  has on the regression fit. They are calculated as

$$h_{ii} = \mathbf{X}'_i(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}_i$$

and represent the diagonal entries of the hat-matrix

$$\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'.$$

A low leverage implies the presence of many regressor observations similar to  $\mathbf{X}_i$  in the sample, while a high leverage indicates a lack of similar observations near  $\mathbf{X}_i$ .

An observation with a high leverage  $h_{ii}$  but a response value  $Y_i$  that is close to the true regression line  $\mathbf{X}'_i\boldsymbol{\beta}$  (indicating a small error  $u_i$ ) is considered a **good leverage point**. It positively influences the model, especially in data-sparse regions.

Conversely, a **bad leverage point** occurs when both  $h_{ii}$  and the error  $u_i$  are large, indicating both unusual regressor and response values. This can misleadingly impact the regression fit.

The actual error term is unknown, but standardized residuals can be used to differentiate between good and bad leverage points.

### 7.2 Standardized residuals

Many regression diagnostic tools rely on the residuals of the OLS estimation  $\hat{u}_i$  because they provide insight into the properties of the unknown error terms  $u_i$ .

Under the homoskedastic linear regression model (A1)–(A5), the errors are independent and have the property

$$\text{Var}[u_i|\mathbf{X}] = \sigma^2.$$

Since  $\mathbf{P}\mathbf{X} = \mathbf{X}$  and, therefore,

$$\hat{\mathbf{u}} = (\mathbf{I}_n - \mathbf{P})\mathbf{Y} = (\mathbf{I}_n - \mathbf{P})(\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) = (\mathbf{I}_n - \mathbf{P})\mathbf{u},$$

the residuals have a different property:

$$\text{Var}[\hat{\mathbf{u}}|\mathbf{X}] = \sigma^2(\mathbf{I}_n - \mathbf{P}).$$

The  $i$ -th residual satisfies

$$\text{Var}[\hat{u}_i|\mathbf{X}] = \sigma^2(1 - h_{ii}),$$

where  $h_{ii}$  is the  $i$ -th leverage value.

Under the assumption (A5), the variance of  $\hat{u}_i$  depends on  $\mathbf{X}$ , while the variance of  $u_i$  does not. Dividing by  $\sqrt{1 - h_{ii}}$  removes the dependency:

$$\text{Var}\left[\frac{\hat{u}_i}{\sqrt{1 - h_{ii}}}\middle|\mathbf{X}\right] = \sigma^2$$

The **standardized residuals** are defined as follows:

$$r_i := \frac{\hat{u}_i}{\sqrt{s_u^2(1 - h_{ii})}}.$$

Standardized residuals are available using the R command `rstandard()`.

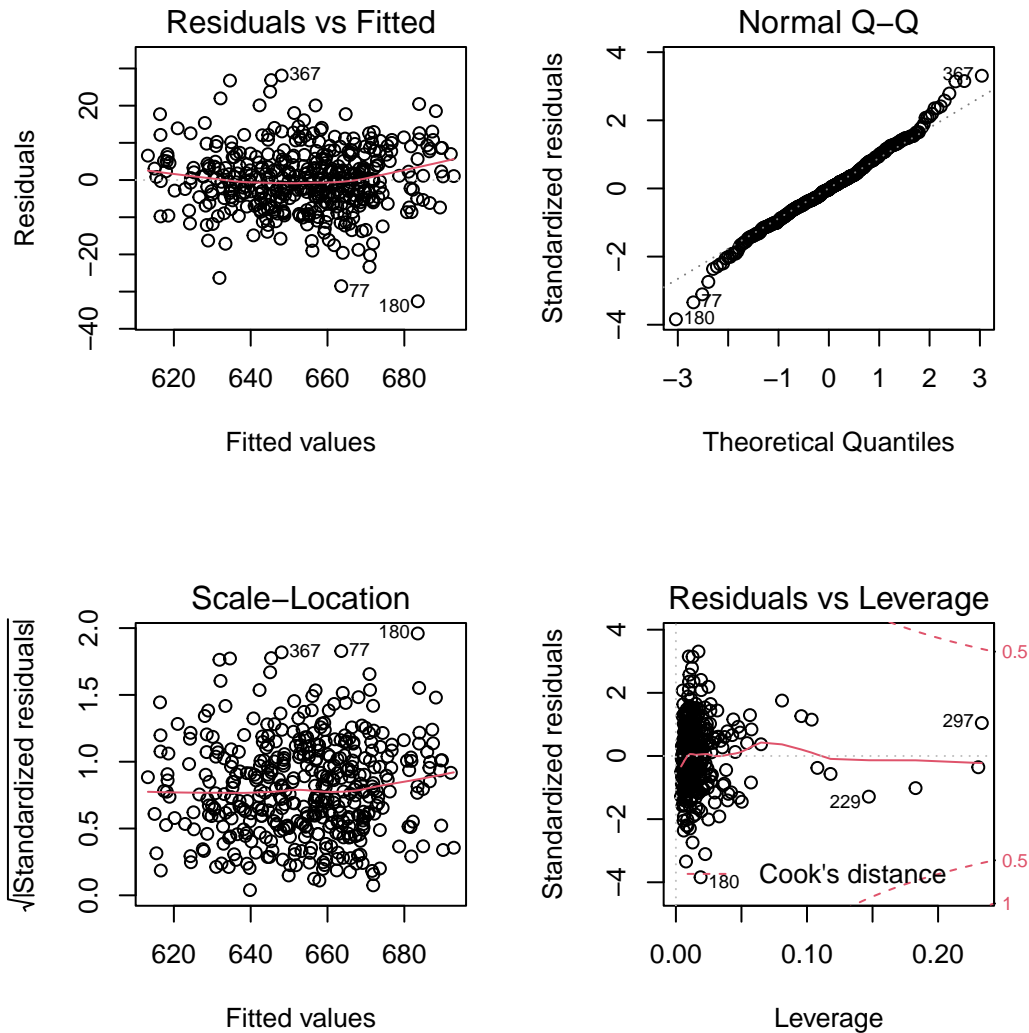
## 7.3 Diagnostics plots

Let's consider the `CASchools` dataset from the previous subsection:

```
library(AER)
data(CASchools)
CASchools$STR <- CASchools$students/CASchools$teachers
CASchools$score <- (CASchools$read + CASchools$math)/2
TS_mod7 <- lm(score ~ STR + I(STR^2) + I(STR^3)
              + english + lunch + log(income),
              data = CASchools)
```

The `plot()` function applied to an `lm` object returns four diagnostics plots:

```
par(mfrow=c(2,2))
plot(TS_mod7)
```



These plots show different scatterplots of the fitted values  $\widehat{Y}_i$ , residuals  $\widehat{u}_i$ , quantiles of the standard normal distribution, leverage values, and standardized residuals.

The red solid line indicates a local scatterplot smoother, which is a smooth locally weighted line through the points on the scatterplot to visualize the general pattern of the data.

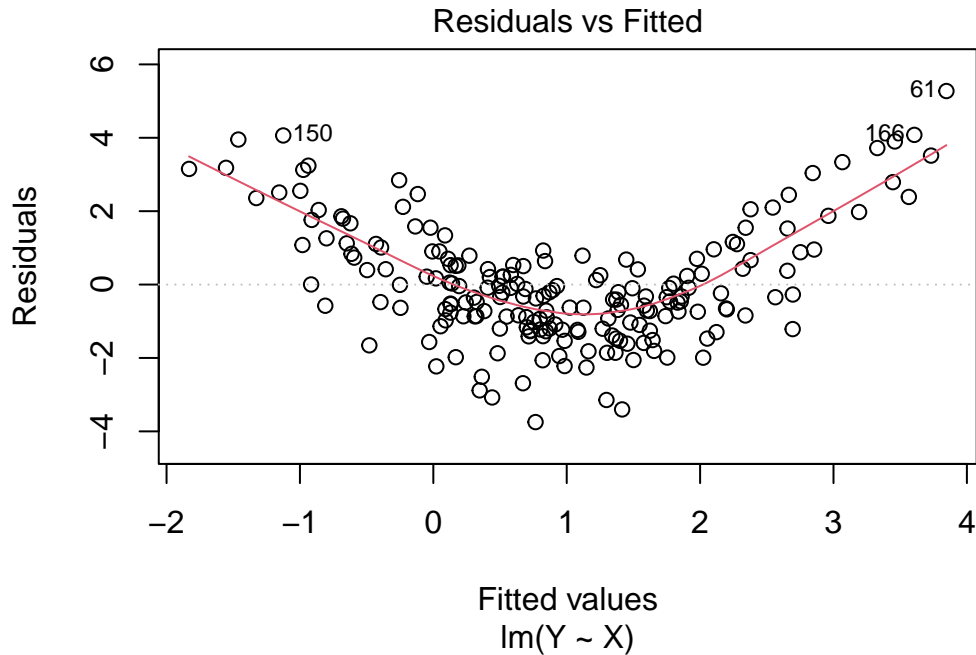
### Plot 1: Residuals vs Fitted

This plot indicates whether there are strong hidden nonlinear relationships between the response and the regressors that are not captured by the model. If a linear model is estimated but the relationship is nonlinear, then the assumption (A1)  $E[u_i | \mathbf{X}_i] = 0$  is violated.

The residuals serve as a proxy for the unknown error terms. If you find equally spread residuals around a horizontal line without distinct patterns, that is a good indication you don't have non-linear relationships.

In the CASchools regression, there is only little indication for an omitted non-linear relationship. Here is an example of a strong omitted nonlinear pattern:

```
# Set seed for reproducibility
set.seed(1)
# Simulate normally distributed regressors
X = rnorm(200)
# Simulate response nonlinearly
Y = X + X^2 + rnorm(200)
# Omit the nonlinearity in the regression
plot(lm(Y ~ X), which = 1)
```



## Plot 2: Normal Q-Q

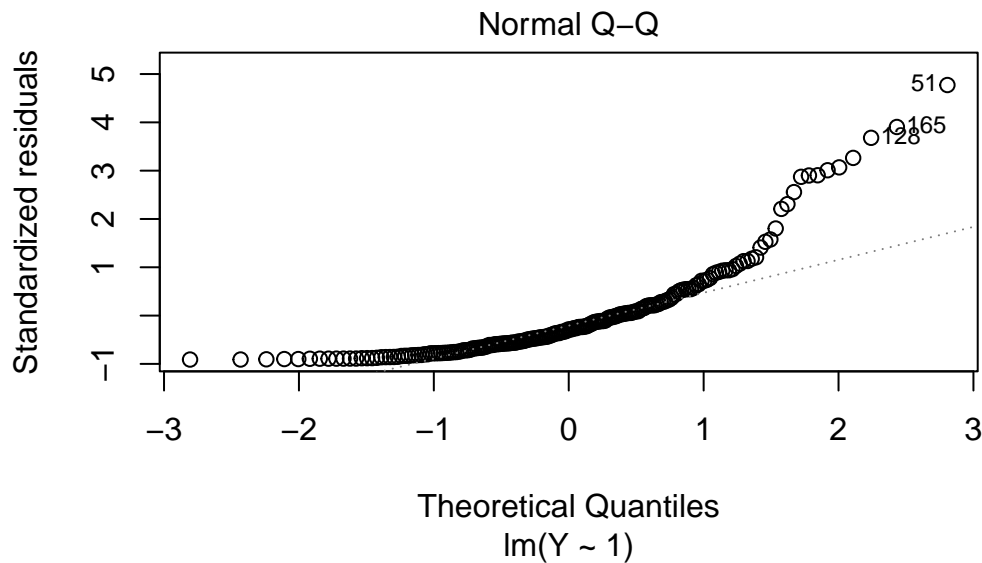
The QQ plot is a graphical tool to help us assess if the errors are conditionally normally distributed, i.e. whether assumption (A6) is satisfied.

Let  $r_{(i)}$  be the order statistics of the standardized residuals (sorted standardized residuals). The QQ plot plots the ordered standardized residuals  $u_{(i)}^*$  against the  $((i - 0.5)/n)$ -quantiles of the standard normal distribution.

If the residuals are lined well on the straight dashed line, there is indication that the distribution of the residuals is close to a normal distribution.

In the `CASchools` regression, we see a slight deviation from normality in the tails. Here is an extrem example with a strong deviation from normality:

```
# Exponentially distributed response variable
Y = rexp(200)
# Intercept only regression model
plot(lm(Y ~ 1), which = 2)
```



### Plot 3: Scale-Location

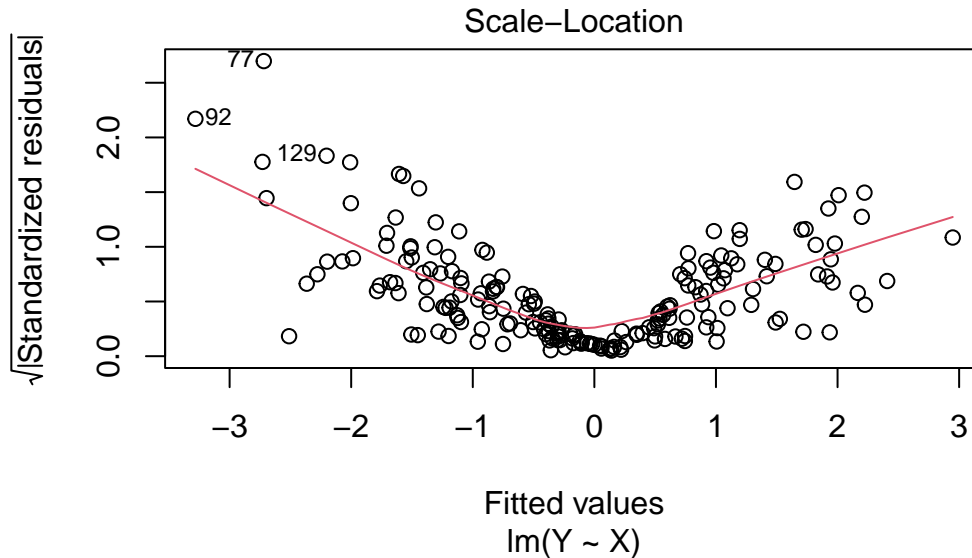
This plot shows if error terms are spread equally along the ranges of regressor values, which is how you can check the assumption of homoskedasticity (A5).

If you see a horizontal line with equally spread points, there is no indication for heteroskedasticity.

In the `CASchools` regression, we have some indication for weak heteroskedasticity. Here is an example with extreme heteroskedasticity:

```
## simulate regressor values
X = rnorm(200)
## error variance varies with the regressor value
u = rnorm(200)*X^2
```

```
## response value
Y = X + u
plot(lm(Y ~ X), which = 3)
```



#### Plot 4: Residuals vs Leverage

Plotting standardized residuals against leverage values provides a graphical tool for detecting outliers. High leverage points have a strong influence on the regression fit. High leverage values with standardized residuals close to 0 are good leverage points, and high leverage values with large standardized residuals are bad leverage points.

The plot also shows Cook's distance thresholds. Cook's distance for observation  $i$  is defined as

$$D_i = \frac{(\hat{\beta}_{(-i)} - \hat{\beta})' \mathbf{X}' \mathbf{X} (\hat{\beta}_{(-i)} - \hat{\beta})}{k s_u^2},$$

where

$$\hat{\beta}_{(-i)} = \hat{\beta} - (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}_i \hat{u}_i (1 - h_{ii})^{-1}$$

is the  $i$ -th leave-one-out estimator (the OLS estimator when the  $i$ -th observation is left out).

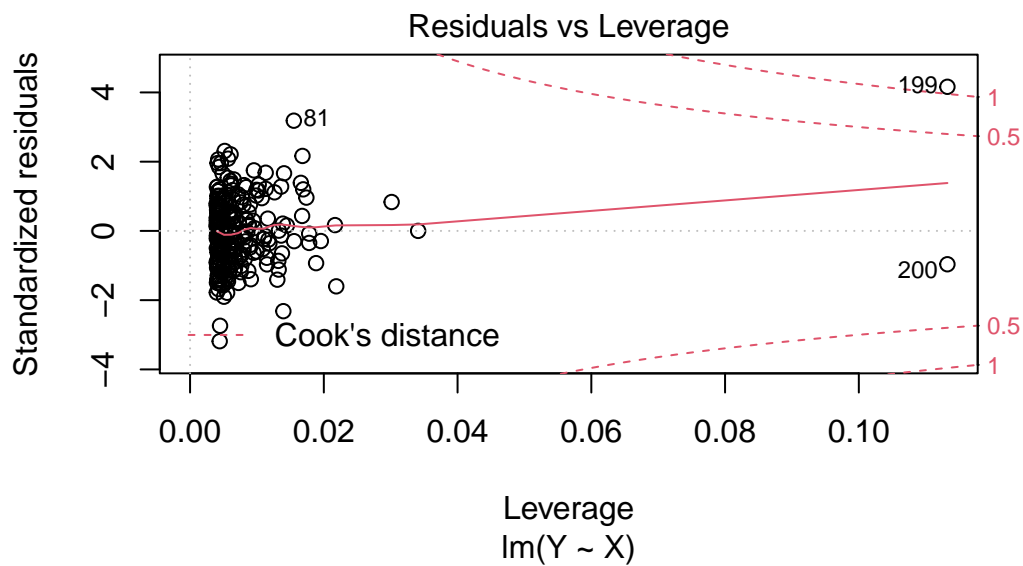
We should pay special attention to points outside Cook's distance thresholds of 0.5 and 1 and check for measurement errors or other anomalies.

Here is an example with two high leverage points. Observation  $i = 200$  is a good leverage point and  $i = 199$  is a bad leverage point:

```

## simulate regressors and errors
X = rnorm(250)
u = rnorm(250)
## set some unusual observations manually
X[199] = 6
X[200] = 6
u[199] = 5
u[200] = 0
## define dependent variable
Y = X + u
## residuals vs leverage plot
plot(lm(Y ~ X), which = 5)

```



## 7.4 Diagnostics tests

The asymptotic properties of the OLS estimator and inferential methods using HC-type standard errors do not depend on the validity of the homoskedasticity and normality assumptions (A5)–(A6).

However, if you are interested in exact inference, verifying the assumptions (A5)–(A6) becomes crucial, especially in small samples.



### 7.4.1 Breusch-Pagan Test (Koenker's version)

Under homoskedasticity, the variance of the error term does not depend on the values of the regressors.

To test for heteroskedasticity, we regress the squared residuals on the regressors.

$$\hat{u}_i^2 = \mathbf{X}'_i \boldsymbol{\gamma} + v_i, \quad i = 1, \dots, n. \quad (7.1)$$

Here,  $\boldsymbol{\gamma}$  are the auxiliary coefficients and  $v_i$  are the auxiliary error terms. Under homoskedasticity, the regressors should not be able to explain any variation in the residuals.

Let  $R_{aux}^2$  be the r-squared coefficient of the auxiliary regression of Equation 7.1. The test statistic:

$$BP = nR_{aux}^2$$

Under the null hypothesis of homoskedasticity, we have

$$BP \xrightarrow{D} \chi_{k-1}^2$$

Test decision rule: Reject  $H_0$  if  $BP$  exceeds  $\chi_{(1-\alpha, k-1)}^2$ .

In R we can apply the `bptest()` function from the `lmtest` package to the `lm` object of our regression.

### 7.4.2 Jarque-Bera Test

A general property of any normally distributed random variable is that it has a skewness of 0 and a kurtosis of 3.

Under (A5)–(A6), we have  $u_i \sim \mathcal{N}(0, \sigma^2)$ , which implies  $E[u_i^3] = 0$  and  $E[u_i^4] = 3\sigma^4$ .

Consider the sample skewness and the sample kurtosis of the residuals from your regression:

$$\widehat{skew}_{\hat{u}} = \frac{1}{n\hat{\sigma}_{\hat{u}}^3} \sum_{i=1}^n \hat{u}_i^3, \quad \widehat{kurt}_{\hat{u}} = \frac{1}{n\hat{\sigma}_{\hat{u}}^4} \sum_{i=1}^n \hat{u}_i^4$$

Jarque-Bera test statistic and null distribution if (A5)–(A6) hold:

$$JB = n \left( \frac{1}{6} (\widehat{skew}_{\hat{u}})^2 + \frac{1}{24} (\widehat{kurt}_{\hat{u}} - 3)^2 \right) \xrightarrow{D} \chi_2^2.$$

Test decision rule: Reject the null hypothesis of normality if  $JB$  exceeds  $\chi_{(1-\alpha, 2)}^2$ .

The Jarque-Bera test is sensitive to outliers.

In R we apply use the `jarque.test()` function from the `moments` package to the residual vector from our regression.

## 7.5 R-codes

[methods-sec07.R](#)

## **Part III**

### **C) Panel Data Methods**

## 8 Panel Regression

```
library(plm) # estimating panel models
library(lmtest) # regression inference
```

### 8.1 Panel Data

Panel data is data collected from multiple individuals at multiple points in time.

Individuals in a typical economic panel data application are people, households, firms, schools, regions, or countries. Time periods are often measured in years (annual data), but may have other frequencies.

$Y_{it}$  denotes a variable for individual  $i$  at time period  $t$ . We index observations by both individuals  $i = 1, \dots, n$  and the time period  $t = 1, \dots, T$ .

Multivariate panel data with  $k$  variables can be written as  $X_{1,it}, \dots, X_{k,it}$ , or, in vector form,

$$\mathbf{X}_{it} = \begin{pmatrix} X_{1,it} \\ X_{2,it} \\ \vdots \\ X_{k,it} \end{pmatrix}.$$

In a *balanced panel*, each individual  $i = 1, \dots, n$  has  $T$  observations. The total number of observations is  $nT$ . In typical economic panel datasets we have  $n > T$  (more individuals than time points) or  $n \approx T$  (roughly the same number of individuals as time points).

Often panel data have some missing data for at least one time period for at least one entity. In this case, we call it an *unbalanced panel*. Notation for unbalanced panels is tedious, so we focus here only on balanced panels. Statistical software can handle unbalanced panel data in much the same way as balanced panel data.

## 8.2 Pooled Regression

The simplest regression model for panel data is the pooled regression.

Consider a panel dataset with dependent variable  $Y_{it}$  and  $k$  independent variables  $X_{1,it}, \dots, X_{k,it}$  for  $i = 1, \dots, n$  and  $t = 1, \dots, T$ .

The first regressor variable represents an intercept (i.e.  $X_{1,it} = 1$ ). We stack the regressor variables into the  $k \times 1$  vector

$$\mathbf{X}_{it} = \begin{pmatrix} 1 \\ X_{2,it} \\ \vdots \\ X_{k,it} \end{pmatrix}.$$

The idea of pooled regression is to pool all observations over  $i = 1, \dots, n$  and  $t = 1, \dots, T$  and run a regression on the combined  $nT$  observations.

### Pooled Panel Regression Model

The pooled linear panel regression model equation for individual  $i = 1, \dots, n$  and time  $t = 1, \dots, T$  is

$$Y_{it} = \mathbf{X}'_{it}\boldsymbol{\beta} + u_{it},$$

where  $\boldsymbol{\beta} = (\beta_1, \dots, \beta_k)'$  is the  $k \times 1$  vector of **regression coefficients** and  $u_{it}$  is the **error term** for individual  $i$  at time  $t$ .

The pooled OLS estimator is

$$\hat{\boldsymbol{\beta}}_{\text{pool}} = \left( \sum_{i=1}^n \sum_{t=1}^T \mathbf{X}_{it}\mathbf{X}'_{it} \right)^{-1} \left( \sum_{i=1}^n \sum_{t=1}^T \mathbf{X}_{it}Y_{it} \right).$$

Similar to linear regression, we can combine the regressors into a pooled regressor matrix of order  $nT \times k$ :

$$\mathbf{X} = (\mathbf{X}_{11}, \dots, \mathbf{X}_{1T}, \mathbf{X}_{21}, \dots, \mathbf{X}_{2T}, \dots, \mathbf{X}_{n1}, \dots, \mathbf{X}_{nT})'.$$

The dependent variable vector is of the order  $nT \times 1$ :

$$\mathbf{Y} = (Y_{11}, \dots, Y_{1T}, Y_{21}, \dots, Y_{2T}, \dots, Y_{n1}, \dots, Y_{nT})'.$$

In matrix notation, the pooled OLS estimator becomes

$$\hat{\boldsymbol{\beta}}_{\text{pool}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}.$$

To illustrate the pooled OLS estimator, consider the **Grunfeld** dataset, which provides investment, capital stock, and firm value data for 10 firms over 20 years.

```
data(Grunfeld, package = "plm")
head(Grunfeld)
```

```
  firm year  inv  value capital
1     1 1935 317.6 3078.5     2.8
2     1 1936 391.8 4661.7    52.6
3     1 1937 410.6 5387.1   156.9
4     1 1938 257.7 2792.2    209.2
5     1 1939 330.8 4313.2    203.4
6     1 1940 461.2 4643.9    207.2
```

```
fit1 = lm(inv~capital, data=Grunfeld)
fit1
```

Call:

```
lm(formula = inv ~ capital, data = Grunfeld)
```

Coefficients:

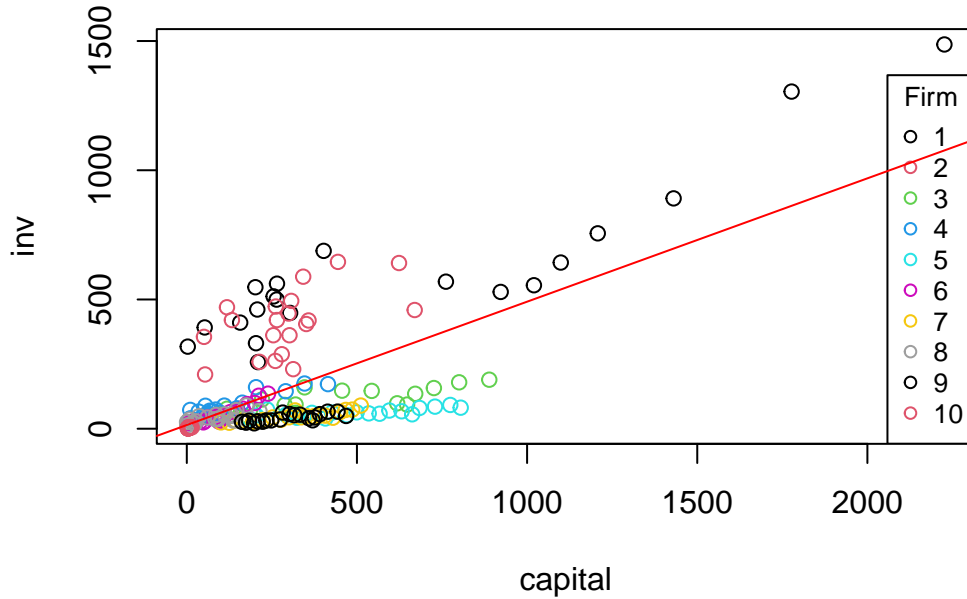
```
(Intercept)      capital
    14.2362         0.4772
```

In principle, the same assumptions can be made as for the linear regression model. However, in view of (A2), the assumption that  $(Y_{it}, \mathbf{X}_{it})$  is independent of  $(Y_{i,t-1}, \mathbf{X}_{i,t-1})$  is unreasonable because we expect  $Y_{it}$  and  $Y_{i,t-1}$  to be correlated (autocorrelation) for the same firm  $i$ .

This can be seen in the graph below. The observations appear in clusters, with each firm forming a cluster.

```
plot(inv~capital, col=as.factor(firm), data = Grunfeld)
legend("bottomright", legend=1:10, col=1:10, pch = 1, title="Firm", cex=0.8)
abline(fit1, col = "red")
```

It is still reasonable to assume that the observations of different individuals are independent. For example, if the firms are randomly selected,  $Y_{it}$  and  $Y_{j,t-1}$  should be independent for  $i \neq j$ .



### 8.3 Pooled Regression Assumptions

We refine our assumptions for the pooled regression case:

- (A1-pool) **conditional mean independence:**  $E[u_{it} | \mathbf{X}_{i1}, \dots, \mathbf{X}_{iT}] = 0$ .
- (A2-pool) **random sampling:**  $(Y_{i1}, \dots, Y_{iT}, \mathbf{X}'_{i1}, \dots, \mathbf{X}'_{iT})$  are i.i.d. draws from their joint population distribution for  $i = 1, \dots, n$ .
- (A3-pool) **large outliers unlikely:**  $0 < E[Y_{it}^4] < \infty$ ,  $0 < E[X_{l,it}^4] < \infty$  for all  $l = 1, \dots, k$ .
- (A4-pool) **no perfect multicollinearity:**  $\mathbf{X}$  has full column rank.

Under (A1-pool)–(A4-pool),  $\hat{\boldsymbol{\beta}}_{pool}$  is consistent for  $\boldsymbol{\beta}$  and asymptotically normal:

$$\frac{\hat{\beta}_i - \beta_i}{sd(\hat{\beta}_i | \mathbf{X})} \xrightarrow{D} \mathcal{N}(0, 1) \quad \text{as } n \rightarrow \infty$$

However,  $sd(\hat{\beta}_i | \mathbf{X}) = \sqrt{Var[\hat{\beta}_i | \mathbf{X}]}$  is different than in the cross-sectional case because of the clustered structure.

The error covariance matrix is of the order  $nT \times nT$  and has the block matrix structure

$$\mathbf{D} = Var[\mathbf{u} | \mathbf{X}] = \begin{pmatrix} \mathbf{D}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{D}_n \end{pmatrix}$$

where  $\mathbf{0}$  indicates the  $T \times T$  matrix of zeros, and on the main diagonal we have the  $T \times T$  cluster-specific covariance matrices

$$\mathbf{D}_i = \begin{pmatrix} E[u_{i,1}^2|\mathbf{X}] & E[u_{i,1}u_{i,2}|\mathbf{X}] & \dots & E[u_{i,1}u_{i,T}|\mathbf{X}] \\ E[u_{i,2}u_{i,1}|\mathbf{X}] & E[u_{i,2}^2|\mathbf{X}] & \dots & E[u_{i,2}u_{i,T}|\mathbf{X}] \\ \vdots & \vdots & \ddots & \vdots \\ E[u_{i,T}u_{i,1}|\mathbf{X}] & E[u_{i,T}u_{i,2}|\mathbf{X}] & \dots & E[u_{i,T}^2|\mathbf{X}] \end{pmatrix}$$

for  $i = 1, \dots, n$ .

We have  $Var[\hat{\beta}|\mathbf{X}] = (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{D}\mathbf{X})(\mathbf{X}'\mathbf{X})^{-1}$  with  $\mathbf{X}'\mathbf{X} = \sum_{i=1}^n \sum_{t=1}^T \mathbf{X}_{it}\mathbf{X}'_{it}$  and

$$\mathbf{X}'\mathbf{D}\mathbf{X} = E\left[\sum_{i=1}^n \left(\sum_{t=1}^T \mathbf{X}_{it}u_{it}\right) \left(\sum_{t=1}^T \mathbf{X}_{it}u_{it}\right)' \middle| \mathbf{X}\right].$$

Therefore, to estimate  $Var[\hat{\beta}|\mathbf{X}]$ , we need a different estimator than in the cross-sectional case.

The cluster-robust covariance matrix estimator is

$$\hat{\mathbf{V}}_{\text{pool}} = (\mathbf{X}'\mathbf{X})^{-1} \sum_{i=1}^n \left(\sum_{t=1}^T \mathbf{X}_{it}\hat{u}_{it}\right) \left(\sum_{t=1}^T \mathbf{X}_{it}\hat{u}_{it}\right)' (\mathbf{X}'\mathbf{X})^{-1},$$

which is the cluster-robust analog of the HC0 sandwich estimator. The cluster-robust standard errors are the squareroots of the diagonal entries of  $\hat{\mathbf{V}}_{\text{pool}}$ .

## 8.4 Pooled Regression Inference

To compute the sandwich form  $\hat{\mathbf{V}}_{\text{pool}}$ , we can use the `plm` package. It provides the `plm()` function for estimating linear panel models. The column names of our data frame corresponding to the individual  $i$  and the time  $t$  are specified by the `index` option.

```
library(plm)
fit2 = plm(inv~capital,
           index = c("firm", "year"),
           model = "pooling",
           data=Grunfeld)
fit2
```

Model Formula: `inv ~ capital`

Coefficients:

(Intercept)	capital
14.23620	0.47722



`fit2` returns the same estimate as `fit1`, but is an object of the class `plm`. You can check it by comparing `class(fit1)` and `class(fit2)`.

The `vcovHC` function applied to a `plm` object returns the cluster-robust covariance matrix  $\widehat{V}_{\text{pool}}$ :

```
Vpool = vcovHC(fit2)
Vpool
```

```
              (Intercept)    capital
(Intercept) 786.5712535 0.34238311
capital      0.3423831 0.01584317
attr(,"cluster")
[1] "group"
```

```
coeftest(fit2, vcov. = Vpool)
```

t test of coefficients:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 14.23620    28.04588  0.5076 0.6122959
capital      0.47722     0.12587  3.7914 0.0001988 ***
---
```

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Alternatively, `coeftest(fit2, vcov. = vcovHC)` gives the same output. Notice the difference compared to `coeftest(fit1, vcov. = vcovHC)`, which does not take into account the clustered structure in the autocovariance matrix and uses  $\widehat{V}_{\text{HC3}}$ .

Similarly to the cross-sectional case, the functions `coefci()` and `linearHypothesis()` can be used for confidence intervals and F/Wald tests.

## 8.5 R-codes

[methods-sec08.R](#)

## 9 Fixed Effects

```
library(plm) # estimating panel models
library(lmtest) # regression inference
library(stargazer) # regression outputs
```

### 9.1 Time-constant Variables

Panel data allows us to control for variables that are constant over time, even if these variables are not directly observable.

Consider a basic panel regression model:

$$Y_{it} = \beta_1 + \beta_2 X_{it} + \beta_3 Z_i + u_{it}. \quad (9.1)$$

Here,  $Z_i$  represents a variable that does not change over time and is specific to an individual (e.g., gender, ethnicity, parental education).

For simplicity, assume here that observations are only available for two time periods ( $t = 1$  and  $t = 2$ ). We can focus on the changes between these periods.

Subtracting the right-hand side of Equation 9.1 at  $t = 1$  from  $t = 2$  gives

$$\begin{aligned} & \beta_1 + \beta_2 X_{i2} + \beta_3 Z_i + u_{i2} - (\beta_1 + \beta_2 X_{i1} + \beta_3 Z_i + u_{i1}) \\ & = \beta_2 \Delta X_{i2} + \Delta u_{i2}. \end{aligned}$$

The symbol  $\Delta$  represents first-differencing, i.e.  $\Delta X_{i2} = X_{i2} - X_{i1}$  and  $\Delta u_{i2} = u_{i2} - u_{i1}$ .

By **first-differencing** both sides of Equation 9.1, our model becomes

$$\Delta Y_{i2} = \beta_2 \Delta X_{i2} + \Delta u_{i2}. \quad (9.2)$$

$\beta_1$  and  $\beta_3 Z_i$  do not appear in the transformed model Equation 9.2 because they are time-constant and cancel out.

In this differenced model,  $\beta_2$  can be estimated by regressing  $\Delta Y_{i2}$  on  $\Delta X_{i2}$  without an intercept. This regression isolates the marginal effect of  $X_{it}$  on  $Y_{it}$  conditional on any unobserved

individual characteristics like  $Z_i$ .  $\beta_2$  is the marginal effect of  $X_{it}$  on  $Y_{it}$  given the same individual-specific time-constant characteristics.

We can control for any time-constant variable without actually observing it. This is a remarkable advantage over conventional cross-sectional regression or pooled panel regression.

We may combine the terms  $\beta_1$  and  $\beta_3 Z_i$  and define the **individual-specific** effect  $\alpha_i = \beta_1 + \beta_3 Z_i$ . The term  $\alpha_i$  is also called **individual fixed effect**. The fixed effect cancels out after taking first differences.

## 9.2 Fixed Effects Regression

Consider a panel dataset with dependent variable  $Y_{it}$ , a vector of  $k$  independent variables  $\mathbf{X}_{it}$ , and an individual fixed effect  $\alpha_i$  for  $i = 1, \dots, n$  and  $t = 1, \dots, T$ .

Because  $\alpha_i$  already represents any time-constant variable of individual  $i$ , we assume that all variables in  $\mathbf{X}_{it}$  are time-varying. That is,  $\mathbf{X}_{it}$  neither contains an intercept nor any time-constant variables like gender, birthplace, etc.

### Fixed-effects Regression

The fixed-effects regression model equation for individual  $i = 1, \dots, n$  and time  $t = 1, \dots, T$  is

$$Y_{it} = \alpha_i + \mathbf{X}'_{it}\boldsymbol{\beta} + u_{it}, \quad (9.3)$$

where  $\boldsymbol{\beta} = (\beta_1, \dots, \beta_k)'$  is the  $k \times 1$  vector of regression coefficients and  $u_{it}$  is the error term for individual  $i$  at time  $t$ .

The fixed effects regression assumptions are:

- (A1-fe) **conditional mean independence:**  $E[u_{it} | \mathbf{X}_{i1}, \dots, \mathbf{X}_{iT}, \alpha_i] = 0$ .
- (A2-fe) **random sampling:**  $(\alpha_i, Y_{i1}, \dots, Y_{iT}, \mathbf{X}'_{i1}, \dots, \mathbf{X}'_{iT})$  are i.i.d. draws from their joint population distribution for  $i = 1, \dots, n$ .
- (A3-fe) **large outliers unlikely:**  $0 < E[Y_{it}^4] < \infty$ ,  $0 < E[u_{it}^4] < \infty$ .
- (A4-fe) **no perfect multicollinearity:**  $\mathbf{X}$  has full column rank.

### 9.3 Differenced Estimator

The first-differencing transformation can be used to estimate Equation 9.3:

$$\Delta Y_{it} = Y_{i,t} - Y_{i,t-1}, \quad \Delta \mathbf{X}_{it} = \mathbf{X}_{i,t} - \mathbf{X}_{i,t-1}.$$

Taking first differences on both sides of Equation 9.3 implies

$$\Delta Y_{it} = (\Delta \mathbf{X}_{it})' \boldsymbol{\beta} + \Delta u_{it}, \quad (9.4)$$

where  $\Delta u_{it} = u_{i,t} - u_{i,t-1}$ . Notice that the fixed effect  $\alpha_i$  cancels out.

Hence, we can apply the OLS principle to Equation 9.4 to estimate  $\boldsymbol{\beta}$ . We regress the differenced dependent variable  $\Delta Y_{it}$  on the differenced regressors  $\Delta \mathbf{X}_{it}$  for  $i = 1, \dots, n$  and  $t = 2, \dots, T$ .

A problem with this differenced estimator is that the transformed error term  $\Delta u_{it}$  defines an artificial correlation structure, which makes the estimator non-optimal.  $\Delta u_{i,t+1} = u_{i,t+1} - u_{i,t}$  is correlated with  $\Delta u_{i,t} = u_{i,t} - u_{i,t-1}$  through  $u_{i,t}$ .

```
data(Grunfeld, package="plm")
fit.diff = plm(inv ~ capital-1,
              index = c("firm", "year"),
              effect = "individual",
              model = "fd",
              data=Grunfeld)
fit.diff
```

Model Formula: inv ~ capital - 1

Coefficients:

capital  
0.23078

### 9.4 Within Estimator

An efficient estimator can be obtained by a different transformation. The idea is to consider the individual specific means

$$\bar{Y}_i = \frac{1}{T} \sum_{t=1}^T Y_{it}, \quad \bar{\mathbf{X}}_i = \frac{1}{T} \sum_{t=1}^T \mathbf{X}_{it}, \quad \bar{u}_i = \frac{1}{T} \sum_{t=1}^T u_{it}.$$

Taking the means of both sides of Equation 9.3 implies

$$\bar{Y}_i = \alpha_i + \bar{\mathbf{X}}_i' \boldsymbol{\beta} + \bar{u}_i. \quad (9.5)$$

Then, subtracting Equation 9.5 from Equation 9.3 removes the fixed effect  $\alpha_i$  from the equation:

$$Y_{it} - \bar{Y}_i = (\mathbf{X}_{it} - \bar{\mathbf{X}}_i)' \boldsymbol{\beta} + (u_{it} - \bar{u}_i).$$

The deviations from the individual specific means are called **within transformations**:

$$\dot{Y}_{it} = Y_{it} - \bar{Y}_i, \quad \dot{\mathbf{X}}_{it} = \mathbf{X}_{it} - \bar{\mathbf{X}}_i, \quad \dot{u}_{it} = u_{it} - \bar{u}_i.$$

The within-transformed model equation is

$$\dot{Y}_{it} = \dot{\mathbf{X}}_{it}' \boldsymbol{\beta} + \dot{u}_{it}. \quad (9.6)$$

Hence, to estimate  $\boldsymbol{\beta}$ , we regress the within-transformed dependent variable  $\dot{Y}_{it}$  on the within-transformed regressors  $\dot{\mathbf{X}}_{it}$  for  $i = 1, \dots, n$  and  $t = 1, \dots, T$ .

The within estimator is also called **fixed effects estimator**:

$$\hat{\boldsymbol{\beta}}_{\text{fe}} = \left( \sum_{i=1}^n \sum_{t=1}^T \dot{\mathbf{X}}_{it} \dot{\mathbf{X}}_{it}' \right)^{-1} \left( \sum_{i=1}^n \sum_{t=1}^T \dot{\mathbf{X}}_{it} \dot{Y}_{it} \right).$$

```
fit.fe = plm(inv ~ capital,
             index = c("firm", "year"),
             effect = "individual",
             model = "within",
             data=Grunfeld)
fit.fe
```

Model Formula: inv ~ capital

Coefficients:

capital  
0.37075

Under (A2-fe), the collection of the within-transformed variables if individual  $i$ ,

$$(\dot{Y}_{i1}, \dots, \dot{Y}_{iT}, \dot{\mathbf{X}}_{i1}, \dots, \dot{\mathbf{X}}_{iT}, \dot{u}_{i1}, \dots, \dot{u}_{iT}),$$

forms an i.i.d. sequence for  $i = 1, \dots, n$ . The within-transformed variables satisfy (A1-pool)–(A4-pool).

Hence, we can apply the cluster-robust covariance matrix estimator of the pooled regression to the within-transformed variables:

$$\widehat{V}_{fe} = (\dot{\mathbf{X}}' \dot{\mathbf{X}})^{-1} \sum_{i=1}^N \left( \sum_{t=1}^T \dot{\mathbf{X}}_{it} \hat{u}_{it} \right) \left( \sum_{t=1}^T \dot{\mathbf{X}}_{it} \hat{u}_{it} \right)' (\dot{\mathbf{X}}' \dot{\mathbf{X}})^{-1},$$

where  $\hat{u}_{it}$  now represents the residuals of  $\hat{\beta}_{fe}$ , and  $\dot{\mathbf{X}}' \dot{\mathbf{X}} = \sum_{i=1}^N \sum_{t=1}^T \dot{\mathbf{X}}_{it} \dot{\mathbf{X}}_{it}'$

```
## cluster-robust covariance matrix
Vfe = vcovHC(fit.fe)
Vfe
```

```
              capital
capital 0.003796144
attr(,"cluster")
[1] "group"
```

```
## cluster-robust standard error
sqrt(Vfe)
```

```
              capital
capital 0.06161285
attr(,"cluster")
[1] "group"
```

```
## t-test
coeftest(fit.fe, vcov. = Vfe)
```

t test of coefficients:

```
              Estimate Std. Error t value Pr(>|t|)
capital 0.370750    0.061613    6.0174 9.018e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## 9.5 Time Fixed Effects

While individual-specific fixed effects allow to control for variables that are constant over time but vary across individuals, we can also control for variables that are constant across

individuals but vary over time. For example, if new government regulations are introduced at a certain point in time that affect all individuals.

We denote time fixed effects by  $\lambda_t$ . The time effects only regression equation is

$$Y_{it} = \lambda_t + \mathbf{X}'_{it}\boldsymbol{\beta} + u_{it}. \quad (9.7)$$

Here,  $\mathbf{X}_{it}$  does not contain any variable that is the same for all individuals, because these variables are captured by the time fixed effect.

To remove  $\lambda_t$  from the equation, we can subtract time specific means on both sides:

$$Y_{it} - \bar{Y}_{.t} = (\mathbf{X}_{it} - \bar{\mathbf{X}}_{.t})'\boldsymbol{\beta} + (u_{it} - \bar{u}_{.t}).$$

The time specific means are

$$\bar{Y}_{.t} = \frac{1}{n} \sum_{i=1}^n Y_{it}, \quad \bar{\mathbf{X}}_{.t} = \frac{1}{n} \sum_{i=1}^n \mathbf{X}_{it}, \quad \bar{u}_{.t} = \frac{1}{n} \sum_{i=1}^n u_{it}.$$

Hence, we regress  $Y_{it} - \bar{Y}_{.t}$  on  $\mathbf{X}_{it} - \bar{\mathbf{X}}_{.t}$  to estimate  $\boldsymbol{\beta}$  in Equation 9.7.

```
fit.timefe = plm(inv ~ capital,
  index = c("firm", "year"),
  effect = "time",
  model = "within",
  data=Grunfeld)
fit.timefe
```

Model Formula: inv ~ capital

Coefficients:

capital  
0.53826

## 9.6 Two-way Fixed Effects

We may include both individual fixed effects and time fixed effects. The two-way fixed effects regression equation is

$$Y_{it} = \alpha_i + \lambda_t + \mathbf{X}'_{it}\boldsymbol{\beta} + u_{it}. \quad (9.8)$$

Note that  $\lambda_t$  and  $\alpha_i$  capture any variable that is the same for all individuals or is time constant. Therefore, the variables in  $\mathbf{X}_{it}$  must vary both across individuals and over time.

We can use a combination of the different transformations to remove the fixed effects.

- Individual specific mean:

$$\bar{Y}_{.i} = \alpha_i + \bar{\lambda} + \bar{\mathbf{X}}_{.i}'\boldsymbol{\beta} + \bar{u}_{.i},$$

where  $\bar{\lambda} = \frac{1}{T} \sum_{t=1}^T \lambda_t$ .

- Time specific mean:

$$\bar{Y}_{.t} = \bar{\alpha} + \lambda_t + \bar{\mathbf{X}}_{.t}'\boldsymbol{\beta} + \bar{u}_{.t},$$

where  $\bar{\alpha} = \frac{1}{n} \sum_{i=1}^n \alpha_i$ .

- Total mean:

$$\bar{Y} = \frac{1}{nT} \sum_{i=1}^n \sum_{t=1}^T Y_{it} = \bar{\alpha} + \bar{\lambda} + \bar{\mathbf{X}}'\boldsymbol{\beta} + \bar{u},$$

where  $\bar{\mathbf{X}} = \frac{1}{nT} \sum_{i=1}^n \sum_{t=1}^T \mathbf{X}_{it}$  and  $\bar{u} = \frac{1}{nT} \sum_{i=1}^n \sum_{t=1}^T u_{it}$ .

To eliminate the individual and time fixed effects in Equation 9.8, we use the **two-way transformation**:

$$\begin{aligned} \ddot{Y}_{it} &= Y_{it} - \bar{Y}_{.i} - \bar{Y}_{.t} + \bar{Y} \\ \ddot{\mathbf{X}}_{it} &= \mathbf{X}_{it} - \bar{\mathbf{X}}_{.i} - \bar{\mathbf{X}}_{.t} + \bar{\mathbf{X}} \\ \ddot{u}_{it} &= u_{it} - \bar{u}_{.i} - \bar{u}_{.t} + \bar{u}. \end{aligned}$$

Applying the two-way transformation on both sides of Equation 9.8 gives

$$\ddot{Y}_{it} = \ddot{\mathbf{X}}_{it}'\boldsymbol{\beta} + \ddot{u}_{it}. \quad (9.9)$$

Hence, we estimate  $\boldsymbol{\beta}$  by regressing  $\ddot{Y}_{it}$  on  $\ddot{\mathbf{X}}_{it}$ .

```
fit.2wayfe = plm(inv ~ capital,
  index = c("firm", "year"),
  effect = "twoways",
  model = "within",
  data=Grunfeld)
fit.2wayfe
```

Model Formula: inv ~ capital

Coefficients:

```
capital
0.4138
```

Similarly to the pooled and fixed effects estimator, we can use the cluster-robust covariance matrix estimator and cluster-robust standard errors.



```
## cluster-robust covariance matrix
V2way = vcovHC(fit.2wayfe)
V2way
```

```

              capital
capital 0.003241852
attr(,"cluster")
[1] "group"
```

```
## cluster-robust standard error
sqrt(Vfe)
```

```

              capital
capital 0.06161285
attr(,"cluster")
[1] "group"
```

```
## t-test
coeftest(fit.2wayfe, vcov. = V2way)
```

t test of coefficients:

```

              Estimate Std. Error t value Pr(>|t|)
capital 0.413802    0.056937   7.2677 1.268e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## 9.7 Comparison of panel models

The fixed effects estimators are asymptotically normal under assumptions (A1-fe)–(A4-fe), and the clustered standard errors are consistent.

```
fit.pool1 = lm(inv~capital, data=Grunfeld)
fit.pool2 = plm(inv~capital,
               index = c("firm", "year"),
               model = "pooling",
               data=Grunfeld)
```

```
cluster_se = list(
  sqrt(diag(vcovHC(fit.pool1))),
  sqrt(diag(vcovHC(fit.pool2))),
  sqrt(diag(vcovHC(fit.fe))),
  sqrt(diag(vcovHC(fit.timefe))),
  sqrt(diag(vcovHC(fit.2wayfe)))
)
```

```
stargazer_output = stargazer(fit.pool1, fit.pool2, fit.fe, fit.timefe, fit.2wayfe,
  se = cluster_se,
  add.lines=list(
    c("Firm FE", "No", "No", "Yes", "No", "Yes"),
    c("Year FE", "No", "No", "No", "Yes", "Yes"),
    c("Clustered SE", "No", "Yes", "Yes", "Yes", "Yes")
  ),
  type="latex",
  omit.stat = "f", df=FALSE,
  dep.var.labels="Gross Investment",
  covariate.labels = "Capital Stock",
  header = FALSE,
  table.placement = "!h")
```

## 9.8 Dummy variable regression

An alternative way to estimate the fixed effects model is by an OLS regression of  $Y_{it}$  on  $\mathbf{X}_{it}$  and a full set of dummy variables, one for each individual in the sample.

For the time fixed effects model, we include a full set of dummy variables for each time point in the sample, and for the two-way fixed effects model, we include individual and time dummies.

This approach is algebraically equivalent to the within and two-way transformations. The coefficients for the auxiliary dummy variables are usually not reported. The coefficients for capital are the same as in the table above:

```
lm(inv ~ capital + factor(firm), data=Grunfeld)
```

Call:

```
lm(formula = inv ~ capital + factor(firm), data = Grunfeld)
```

Table 9.1

	<i>Dependent variable:</i>				
	Gross Investment				
	<i>OLS</i>		<i>panel</i>	<i>linear</i>	
	(1)	(2)	(3)	(4)	(5)
Capital Stock	0.477*** (0.078)	0.477*** (0.126)	0.371*** (0.062)	0.538*** (0.153)	0.414*** (0.057)
Constant	14.236 (19.393)	14.236 (28.046)			
Firm FE	No	No	Yes	No	Yes
Year FE	No	No	No	Yes	Yes
Clustered SE	No	Yes	Yes	Yes	Yes
Observations	200	200	200	200	200
R <sup>2</sup>	0.439	0.439	0.660	0.429	0.599
Adjusted R <sup>2</sup>	0.436	0.436	0.642	0.365	0.530
Residual Std. Error	162.850				

Note:

\*p<0.1; \*\*p<0.05; \*\*\*p<0.01

Coefficients:

```
(Intercept)      capital  factor(firm)2  factor(firm)3  factor(firm)4
367.6130         0.3707      -66.4553      -413.6821     -326.4410
factor(firm)5    factor(firm)6  factor(firm)7  factor(firm)8  factor(firm)9
-486.2784       -350.8656    -436.7832    -356.4725     -436.1703
factor(firm)10
-366.7313
```

```
lm(inv ~ capital + factor(year), data=Grunfeld)
```

Call:

```
lm(formula = inv ~ capital + factor(year), data = Grunfeld)
```

Coefficients:

```
(Intercept)      capital  factor(year)1936  factor(year)1937
```

39.2068	0.5383	22.4605	27.8993
factor(year)1938	factor(year)1939	factor(year)1940	factor(year)1941
-36.6889	-42.4012	-11.4293	5.3301
factor(year)1942	factor(year)1943	factor(year)1944	factor(year)1945
-26.2522	-36.3995	-32.3887	-33.0571
factor(year)1946	factor(year)1947	factor(year)1948	factor(year)1949
-3.6307	-57.8083	-73.1115	-106.8436
factor(year)1950	factor(year)1951	factor(year)1952	factor(year)1953
-105.8753	-69.2505	-76.6097	-67.6766
factor(year)1954			
-112.6339			

```
lm(inv ~ capital + factor(firm) + factor(year), data=Grunfeld)
```

Call:

```
lm(formula = inv ~ capital + factor(firm) + factor(year), data = Grunfeld)
```

Coefficients:

(Intercept)	capital	factor(firm)2	factor(firm)3
354.9166	0.4138	-51.2329	-402.9933
factor(firm)4	factor(firm)5	factor(firm)6	factor(firm)7
-303.7443	-479.3182	-327.4387	-422.4257
factor(firm)8	factor(firm)9	factor(firm)10	factor(year)1936
-332.2429	-421.0790	-339.0705	23.9405
factor(year)1937	factor(year)1938	factor(year)1939	factor(year)1940
32.9483	-27.0935	-30.7979	0.5826
factor(year)1941	factor(year)1942	factor(year)1943	factor(year)1944
19.5836	-8.6393	-17.5675	-13.7593
factor(year)1945	factor(year)1946	factor(year)1947	factor(year)1948
-13.5253	17.6985	-27.2407	-37.4300
factor(year)1949	factor(year)1950	factor(year)1951	factor(year)1952
-66.7623	-63.2855	-23.9098	-23.9138
factor(year)1953	factor(year)1954		
-5.1266	-40.1051		

## 9.9 Panel R-squared

We can decompose the total variation into within group variation and between group variation:

$$Y_{it} - \bar{Y} = \underbrace{Y_{it} - \bar{Y}_i}_{\text{within group}} + \underbrace{\bar{Y}_i - \bar{Y}}_{\text{between group}}$$

Two different R squared versions:

- Overall R-squared:

$$R_{ov}^2 = 1 - \frac{\sum_{i=1}^n \sum_{t=1}^T \hat{u}_{it}^2}{\sum_{i=1}^n \sum_{t=1}^T (Y_{it} - \bar{Y})^2}$$

*Interpretation:* Proportion of total sample variation in  $Y_{it}$  explained by the model (the usual R-squared).

- Within R-squared

$$R_{wit}^2 = 1 - \frac{\sum_{i=1}^n \sum_{t=1}^T \hat{u}_{it}^2}{\sum_{i=1}^n \sum_{t=1}^T (Y_{it} - \bar{Y}_i)^2}$$

*Interpretation:* Proportion of sample variation in  $Y_{it}$  within the individual units is explained by the model.

For a individual-specific fixed effects regression, consider the two equivalent fixed effects estimators from above:

```
## plm object
fit.fe = plm(inv ~ capital,
             index = c("firm", "year"),
             effect = "individual",
             model = "within",
             data=Grunfeld)

## lm object
fit.fe.lsdv = lm(inv ~ capital + factor(firm), data=Grunfeld)
```

The `summary(object)$r.squared` function applied to the plm object returns the within R-squared, and for the lm object it returns the overall R-squared:

```
## within R-squared
summary(fit.fe)$r.squared
```

```
      rsq      adjrsq
0.6597327 0.6417291
```

```
## overall R-squared
summary(fit.fe.lsdv)$r.squared
```

```
[1] 0.9184098
```

It is not a big surprise that the fixed effects model explains a lot of the total variation in  $Y_{it}$ . The equivalent LSDV model assigns each individual its own dummy variable and therefore, by construction, explains a lot of variation between individuals.

The within R squared is often more insightful because it reflects the model's ability to explain the variation within entities over time.

## 9.10 R-codes

[methods-sec09.R](#)

## 10 Case Study II: Drunk Driving

```
library(AER) # for the dataset
library(plm) # panel models
library(stargazer) # regression tables
```

The dataset `Fatalities` contains panel data for traffic fatalities in the United States. Among others, it contains variables related to traffic fatalities and alcohol, including the number of traffic fatalities, the type of drunk driving laws and the tax on beer, reporting their values for each state and each year.

Here we will study how effective various government policies designed to discourage drunk driving actually are in reducing traffic deaths.

The measure of traffic deaths we use is the fatality rate, which is the annual number of traffic fatalities per 10000 individuals within the state's population. The measure of alcohol taxes we use is the "real" tax on a case of beer, which is the beer tax, put into 1988 dollars by adjusting for inflation.

Let's take a look at the structure of the dataset first.

```
data(Fatalities, package = "AER")
class(Fatalities)
```

```
[1] "data.frame"
```

```
dim(Fatalities)
```

```
[1] 336 34
```

```
str(Fatalities)
```

Click here to view or hide `str(Fatalities)`

```

'data.frame':  336 obs. of  34 variables:
 $ state      : Factor w/ 48 levels "al","az","ar",...: 1 1 1 1 1 1 1 2 2 2 ...
 $ year      : Factor w/ 7 levels "1982","1983",...: 1 2 3 4 5 6 7 1 2 3 ...
 $ spirits   : num  1.37 1.36 1.32 1.28 1.23 ...
 $ unemp     : num  14.4 13.7 11.1 8.9 9.8 ...
 $ income    : num  10544 10733 11109 11333 11662 ...
 $ emppop   : num  50.7 52.1 54.2 55.3 56.5 ...
 $ beertax   : num  1.54 1.79 1.71 1.65 1.61 ...
 $ baptist   : num  30.4 30.3 30.3 30.3 30.3 ...
 $ mormon    : num  0.328 0.343 0.359 0.376 0.393 ...
 $ drinkage  : num  19 19 19 19.7 21 ...
 $ dry       : num  25 23 24 23.6 23.5 ...
 $ youngdrivers: num  0.212 0.211 0.211 0.211 0.213 ...
 $ miles     : num  7234 7836 8263 8727 8953 ...
 $ breath    : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ jail      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 2 2 2 ...
 $ service   : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 2 2 2 ...
 $ fatal     : int  839 930 932 882 1081 1110 1023 724 675 869 ...
 $ nfatal    : int  146 154 165 146 172 181 139 131 112 149 ...
 $ sfatal    : int  99 98 94 98 119 114 89 76 60 81 ...
 $ fatal1517 : int  53 71 49 66 82 94 66 40 40 51 ...
 $ nfatal1517 : int  9 8 7 9 10 11 8 7 7 8 ...
 $ fatal1820 : int  99 108 103 100 120 127 105 81 83 118 ...
 $ nfatal1820 : int  34 26 25 23 23 31 24 16 19 34 ...
 $ fatal2124 : int  120 124 118 114 119 138 123 96 80 123 ...
 $ nfatal2124 : int  32 35 34 45 29 30 25 36 17 33 ...
 $ afatal    : num  309 342 305 277 361 ...
 $ pop       : num  3942002 3960008 3988992 4021008 4049994 ...
 $ pop1517   : num  209000 202000 197000 195000 204000 ...
 $ pop1820   : num  221553 219125 216724 214349 212000 ...
 $ pop2124   : num  290000 290000 288000 284000 263000 ...
 $ milestot  : num  28516 31032 32961 35091 36259 ...
 $ unempus   : num  9.7 9.6 7.5 7.2 7 ...
 $ emppopus  : num  57.8 57.9 59.5 60.1 60.7 ...
 $ gsp       : num  -0.0221 0.0466 0.0628 0.0275 0.0321 ...

```

We can see the data has been effectively defined as a data frame, with 336 observations of 34 variables. Our panel index variables are `state` (individual,  $i$ ) and `year` (time,  $t$ ).

It's always good to have a quick look at the first few observations. The `head()` function in R, by default, shows the first six observations (rows) of a data frame or data set. However,



you can specify a different number of rows to display by providing the desired count as an argument to the function if needed, like `head(your_data_frame, n = 10)` to display the first 10 rows.

[Click here to view or hide head\(Fatalities\)](#)

```
# list the first few observations
head(Fatalities)
```

```

state year spirits unemp  income  emppop  beertax  baptist  mormon  drinkage
1    al  1982    1.37  14.4 10544.15 50.69204 1.539379 30.3557 0.32829   19.00
2    al  1983    1.36  13.7 10732.80 52.14703 1.788991 30.3336 0.34341   19.00
3    al  1984    1.32  11.1 11108.79 54.16809 1.714286 30.3115 0.35924   19.00
4    al  1985    1.28   8.9 11332.63 55.27114 1.652542 30.2895 0.37579   19.67
5    al  1986    1.23   9.8 11661.51 56.51450 1.609907 30.2674 0.39311   21.00
6    al  1987    1.18   7.8 11944.00 57.50988 1.560000 30.2453 0.41123   21.00
      dry youngdrivers  miles breath jail service fatal nfatal sfatal
1 25.0063  0.211572 7233.887   no   no     no   839   146   99
2 22.9942  0.210768 7836.348   no   no     no   930   154   98
3 24.0426  0.211484 8262.990   no   no     no   932   165   94
4 23.6339  0.211140 8726.917   no   no     no   882   146   98
5 23.4647  0.213400 8952.854   no   no     no  1081   172  119
6 23.7924  0.215527 9166.302   no   no     no  1110   181  114
fatal1517 nfatal1517 fatal1820 nfatal1820 fatal2124 nfatal2124  afatal
1         53          9         99         34         120         32 309.438
2         71          8        108         26         124         35 341.834
3         49          7        103         25         118         34 304.872
4         66          9        100         23         114         45 276.742
5         82         10        120         23         119         29 360.716
6         94         11        127         31         138         30 368.421
      pop  pop1517  pop1820  pop2124  milestot  unempus  emppopus          gsp
1 3942002 208999.6 221553.4 290000.1  28516    9.7    57.8 -0.02212476
2 3960008 202000.1 219125.5 290000.2  31032    9.6    57.9  0.04655825
3 3988992 197000.0 216724.1 288000.2  32961    7.5    59.5  0.06279784
4 4021008 194999.7 214349.0 284000.3  35091    7.2    60.1  0.02748997
5 4049994 203999.9 212000.0 263000.3  36259    7.0    60.7  0.03214295
6 4082999 204999.8 208998.5 258999.8  37426    6.2    61.5  0.04897637
```

```
# summarize the variables 'state' and 'year'
summary(Fatalities[, c("state", "year")])
```

```

state      year
al       : 7   1982:48
az       : 7   1983:48
ar       : 7   1984:48
ca       : 7   1985:48
co       : 7   1986:48
ct       : 7   1987:48
(Other):294  1988:48
```

Notice that the variable `state` is a factor variable with 48 levels (one for each of the 48 contiguous federal states of the U.S.). The variable `year` is also a factor variable that has 7 levels identifying the time period when the observation was made. This gives us  $7 \times 48 = 336$  observations in total.

Since all variables are observed for all entities (states) and over all time periods, the panel is *balanced*. If there were missing data for at least one entity in at least one time period we would call the panel *unbalanced*.

## 10.1 Cross-sectional Regression

Let's start by estimating simple regressions using data for years 1982 and 1988 that model the relationship between the beer tax (adjusted for 1988 dollars) and the traffic fatality rate, measured as the number of fatalities per 10000 inhabitants. Afterwards, we plot the data and add the corresponding estimated regression functions.

```
# define the fatality rate
Fatalities$fatal_rate = Fatalities$fatal / Fatalities$pop * 10000

# subset the data
Fatalities1982 = Fatalities |> subset(year == "1982")
Fatalities1988 = Fatalities |> subset(year == "1988")

# estimate simple regression models using 1982 and 1988 data
fatal1982_mod = lm(fatal_rate ~ beertax, data = Fatalities1982)
fatal1988_mod = lm(fatal_rate ~ beertax, data = Fatalities1988)

coeftest(fatal1982_mod, vcov. = vcovHC)
```

t test of coefficients:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.01038     0.15278 13.1586  <2e-16 ***
beertax      0.14846     0.14500  1.0238  0.3113
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(fatal1988_mod, vcov. = vcovHC)
```

t test of coefficients:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.85907     0.11786 15.7731 < 2.2e-16 ***
beertax      0.43875     0.14224  3.0847  0.003443 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated regression functions are

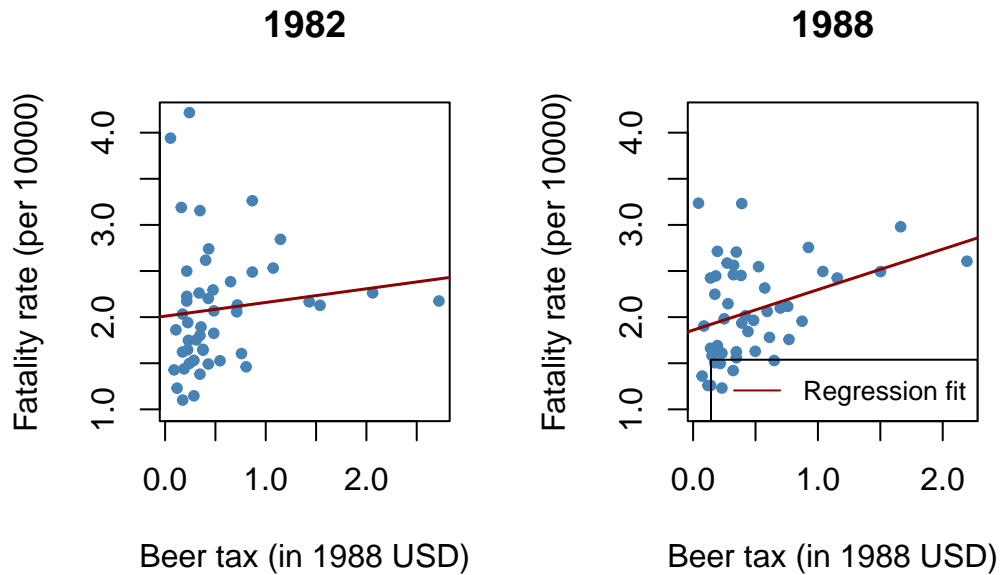
$$\widehat{FatalityRate} = 2.01 + 0.15 BeerTax \quad (1982 \text{ data})$$

(0.15)      (0.15)

$$\widehat{FatalityRate} = 1.86 + 0.44 BeerTax \quad (1988 \text{ data})$$

(0.12)      (0.14)

```
par(mfrow = c(1,2))
plot(fatal_rate~beertax, data = Fatalities1982,
     xlab = "Beer tax (in 1988 USD)", ylab = "Fatality rate (per 10000)",
     main = "1982", ylim = c(1, 4.2),
     pch = 20, col = "steelblue")
abline(fatal1982_mod, lwd = 1.5, col="darkred")
plot(fatal_rate~beertax, data = Fatalities1988,
     xlab = "Beer tax (in 1988 USD)", ylab = "Fatality rate (per 10000)",
     main = "1988", ylim = c(1, 4.2),
     pch = 20, col = "steelblue")
abline(fatal1988_mod, lwd = 1.5, col="darkred")
legend("bottomright",lty=1,col="darkred","Regression fit", cex = 0.8)
```



In both plots, each point represents observations of beer tax and fatality rate for a given state in the respective year. The regression results indicate a positive relationship between the beer tax and the fatality rate for both years.

The estimated coefficient on beer tax for the 1988 data is almost three times as large as for the 1982 dataset. This is contrary to our expectations: alcohol taxes are supposed to lower the rate of traffic fatalities. This is possibly due to omitted variable bias, since none of the models include any covariates, e.g., economic conditions.

Panel data methods could help here to account for omitted unobservable factors that vary from state to state but can be assumed to be constant over the observation period (e.g., attitudes toward drunk driving, road quality, density of cars on the road) and factors that vary from year to year but can be assumed to be constant for all states in a given year (e.g., changing national attitudes toward drunk driving, improvements in car safety over time).

## 10.2 “Before and After” Comparisons

Let’s suppose there are only  $T = 2$  time periods  $t = 1982, 1988$ . This allows us to analyze differences in changes of the fatality rate from year 1982 to 1988. We start by considering the population regression model:

$$\text{FatalityRate}_{it} = \beta_0 + \beta_1 \text{BeerTax}_{it} + \beta_2 Z_i + u_{it}$$

where the  $Z_i$  are state specific characteristics that differ between states but are constant over time. For  $t = 1982$  and  $t = 1988$  we have

$$FatalityRate_{i,1982} = \beta_0 + \beta_1 BeerTax_{i,1982} + \beta_2 Z_i + u_{i,1982},$$

$$FatalityRate_{i,1988} = \beta_0 + \beta_1 BeerTax_{i,1988} + \beta_2 Z_i + u_{i,1988}.$$

We can eliminate the  $Z_i$  by regressing the difference in the fatality rate between 1988 and 1982 on the difference in beer tax between those years:

$$\begin{aligned} & FatalityRate_{i,1988} - FatalityRate_{i,1982} \\ &= \beta_1(BeerTax_{i,1988} - BeerTax_{i,1982}) + u_{i,1988} - u_{i,1982} \end{aligned}$$

This regression model, where the difference in fatality rate between 1988 and 1982 is regressed on the difference in beer tax between those years, yields an estimate for  $\beta_1$  that is robust to a possible bias due to omission of  $Z_i$ , as these influences are eliminated from the model. Next we will estimate a regression based on the differenced data and plot the estimated regression function.

```
# compute the differences
diff_fatal_rate = Fatalities1988$fatal_rate - Fatalities1982$fatal_rate
diff_beertax = Fatalities1988$beertax - Fatalities1982$beertax

# estimate a regression using differenced data
fatal_diff_mod = lm(diff_fatal_rate ~ diff_beertax)
coeftest(fatal_diff_mod, vcov = vcovHC)
```

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.072037	0.067854	-1.0616	0.29394
diff_beertax	-1.040973	0.408288	-2.5496	0.01418 *

---  
 Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Including the intercept allows for a change in the mean fatality rate in the time between 1982 and 1988 in the absence of a change in the beer tax.

We obtain the OLS estimated regression function

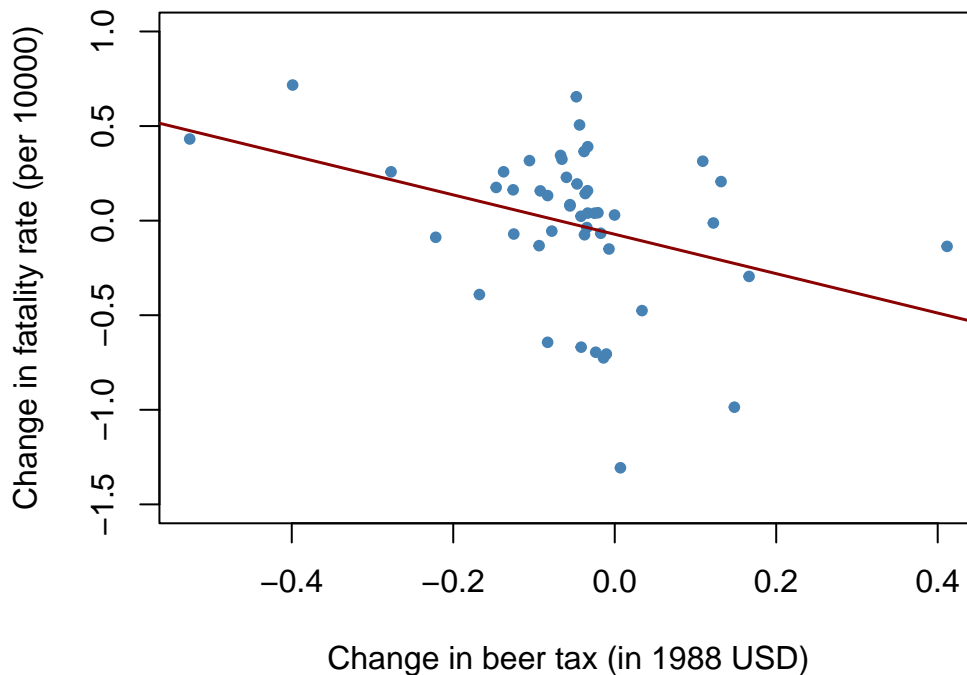
$$\begin{aligned}
 & FatalityRate_{i,1988} - \widehat{FatalityRate}_{i,1982} \\
 & = -0.072 - 1.04 (BeerTax_{i,1988} - BeerTax_{i,1982}) \\
 & \quad \quad \quad (-0.07) \quad \quad (0.41)
 \end{aligned}$$

```

plot(diff_fatal_rate ~ diff_beertax,
     xlab = "Change in beer tax (in 1988 USD)",
     ylab = "Change in fatality rate (per 10000)",
     main = "Changes in Traffic Fatality Rates and Beer Taxes in 1982-1988",
     ylim = c(-1.5, 1), cex.main=1,
     pch = 20, col = "steelblue")
abline(fatal_diff_mod, lwd = 1.5,col="darkred") # add the regression line to plot

```

**Changes in Traffic Fatality Rates and Beer Taxes in 1982-1988**



The estimated coefficient on beer tax is now negative and significantly different from zero at the 5% significance level. Its interpretation is that raising the beer tax by \$1 is associated with an average decrease of 1.04 fatalities per 10000 inhabitants. This is rather large as the average fatality rate is approximately 2 persons per 10000 inhabitants.

```
# mean fatality rate over all states and time periods
mean(Fatalities$fatal_rate)
```

```
[1] 2.040444
```

The outcome we obtained is likely to be a consequence of omitting factors in the single-year regression that influence the fatality rate and are correlated with the beer tax and change over time. The message is that we need to be more careful and control for such factors before drawing conclusions about the effect of a raise in beer taxes.

The approach presented in this section discards information for years 1983 to 1987. The fixed effects method allows us to use data for more than  $T = 2$  time periods and enables us to add control variables to the analysis.

### 10.3 State Fixed Effects

To estimate the relation between traffic fatality rates and beer taxes, the simple fixed effects model is

$$FatalityRate_{it} = \alpha_i + \beta_1 BeerTax_{it} + u_{it} \quad (10.1)$$

a regression of the traffic fatality rate on beer tax and 48 binary regressors (one for each federal state). In this model, we are using a fixed effects approach to account for the effect of each federal state.  $\alpha_i$  represents the state fixed effect. Including a fixed effect for each state means that we're estimating separate intercepts (or constant terms) for each state.

```
fatal_fe = plm(fatal_rate ~ beertax,
              index = c("state", "year"),
              effect = "individual",
              model = "within",
              data = Fatalities)
coeftest(fatal_fe, vcov. = vcovHC)
```

t test of coefficients:

```
          Estimate Std. Error t value Pr(>|t|)
beertax -0.65587    0.28837  -2.2744  0.02368 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated coefficient is again  $-0.6559$ . The estimated regression function is

$$\widehat{FatalityRate} = -0.66 \underset{(0.29)}{BeerTax} + StateFE \quad (10.2)$$

The coefficient on *BeerTax* is negative and statistically significant at the 5% level. Its interpretation is that states with a \$1 higher beer tax have, on average, 0.66 fewer traffic fatalities per 10000 people, given the same state-specific time-constant characteristics.

Although including state fixed effects eliminates the risk of bias due to omitted factors that vary across states but not over time, we suspect that there are other omitted variables that vary over time, making it difficult to interpret the coefficient as a causal effect.

If you prefer the `lm()` function, you can also use the following command:

```
fatal_fe_lm = lm(fatal_rate ~ beertax + factor(state) - 1, data = Fatalities)
```

The `-1` term tells R to exclude the intercept term that it would normally include by default. By doing this, we're essentially saying that we don't want to estimate an overall intercept for the model because we are already capturing the state-specific effects. This is a common practice in fixed effects models to avoid multicollinearity between the state-specific intercepts and the predictors.

While `fatal_fe_lm` and `fatal_fe` return the same coefficient estimate, `vcovHC(fatal_fe_lm)` returns the HC3 heteroskedasticity-robust covariance matrix and `vcovHC(fatal_fe)` returns the cluster-robust covariance matrix. The reason is that `fatal_fe_lm` is an `lm` object and `fatal_fe` is a `plm` object. Cluster-robust standard errors should be preferred due to the autocorrelation structure within each cluster (state).

## 10.4 Year Fixed Effects

Controlling for variables that are constant across entities but vary over time can be done by including time fixed effects. If there are *only* time fixed effects, the fixed effects regression model becomes

$$Y_{it} = \lambda_t + \beta_1 X_{it} + u_{it}$$

In some applications it is meaningful to include both entity (state) and time fixed effects. The **two-way fixed effects model** is

$$Y_{it} = \alpha_i + \lambda_t + \beta_1 X_{it} + u_{it}$$



The combined model allows to eliminate bias from unobservables that change over time but are constant over entities and it controls for factors that differ across entities but are constant over time.

Let's estimate the combined entity and time fixed effects model of the relation between fatalities and beer tax,

$$FatalityRate_{it} = \beta_1 BeerTax_{it} + StateFE_i + TimeFE_t + u_{it}$$

```
fatal_twoway = plm(fatal_rate ~ beertax,
                  index = c("state", "year"),
                  effect = "twoways",
                  model = "within",
                  data = Fatalities)
coeftest(fatal_twoway, vcov. = vcovHC)
```

t test of coefficients:

```
          Estimate Std. Error t value Pr(>|t|)
beertax -0.63998    0.34963 -1.8305  0.06824 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated regression function is

$$\widehat{FatalityRate} = \underset{(0.35)}{-0.64} BeerTax + StateFE + TimeFE \quad (10.3)$$

The result is close to the estimated coefficient for the regression model including only entity fixed effects, which was  $-0.66$ . Unsurprisingly, the coefficient is less precisely estimated, as we observe a slightly higher cluster-robust standard error for this new coefficient of  $-0.64$ . Nevertheless, it is still significantly different from zero at the 10% level.

We conclude that the estimated relationship between traffic fatalities and the real beer tax is not affected by omitted variable bias due to factors that are constant either over time or across states.

## 10.5 Driving Laws and Economic Conditions

There are two major sources of omitted variable bias that are not accounted for by all of the models of the relation between traffic fatalities and beer taxes that we have considered so far: economic conditions and driving laws.

Fortunately, `Fatalities` has data on state-specific legal drinking age (`drinkage`), punishment (`jail`, `service`) and various economic indicators like unemployment rate (`unemp`) and per capita income (`income`). We may use these covariates to extend the preceding analysis.

These covariates are defined as follows:

- `unemp`: a numeric variable stating the state specific unemployment rate.
- `log(income)`: the logarithm of real per capita income (in 1988 dollars).
- `miles`: the state average miles per driver.
- `drinkage`: the state specific minimum legal drinking age.
- `drinkagec`: a discretized version of `drinkage` that classifies states into four categories of minimal drinking age; 18, 19, 20, 21 and older. R denotes this as `[18,19)`, `[19,20)`, `[20,21)` and `[21,22]`. These categories are included as dummy regressors where `[21,22]` is chosen as the reference category.
- `punish`: a dummy variable with levels `yes` and `no` that measures if drunk driving is severely punished by mandatory jail time or mandatory community service (first conviction).

First, we define some relevant variables to include in our following regression models:

```
# discretize the minimum legal drinking age
Fatalities$drinkagec = factor(floor(Fatalities$drinkage))

# dummy for mandatory jail or community service
Fatalities$punish = ifelse(
  Fatalities$jail == "yes" | Fatalities$service == "yes",
  "yes", "no")
```

Next, we estimate six regression models using `plm()`.

```
# estimate six models
fat_mod1 = plm(fatal_rate ~ beertax,
               index = c("state", "year"),
               model = "pooling",
               data = Fatalities)

fat_mod2 = plm(fatal_rate ~ beertax,
```

```

        index = c("state", "year"),
        effect = "individual",
        model = "within",
        data = Fatalities)

fat_mod3 = plm(fatal_rate ~ beertax,
              index = c("state", "year"),
              effect = "twoways",
              model = "within",
              data = Fatalities)

fat_mod4 = plm(fatal_rate ~ beertax
              + drinkagec + punish + miles + unemp + log(income),
              index = c("state", "year"),
              effect = "twoways",
              model = "within",
              data = Fatalities)

fat_mod5 = plm(fatal_rate ~ beertax
              + drinkagec + punish + miles,
              index = c("state", "year"),
              effect = "twoways",
              model = "within",
              data = Fatalities)

fat_mod6 = plm(fatal_rate ~ beertax
              + drinkage + punish + miles + unemp + log(income),
              index = c("state", "year"),
              effect = "twoways",
              model = "within",
              data = Fatalities)

```

We use `stargazer()` to generate a comprehensive tabular presentation of the results.

```

# gather clustered standard errors in a list
rob_se = list(sqrt(diag(vcovHC(fat_mod1))),
              sqrt(diag(vcovHC(fat_mod2))),
              sqrt(diag(vcovHC(fat_mod3))),
              sqrt(diag(vcovHC(fat_mod4))),
              sqrt(diag(vcovHC(fat_mod5))),
              sqrt(diag(vcovHC(fat_mod6))))

```

```

stargazer(fat_mod1, fat_mod2, fat_mod3, fat_mod4, fat_mod5, fat_mod6,
          se = rob_se,
          type="latex",
          omit.stat = c("f", "rsq", "adj.rsq"),
          add.lines=list(
            c("State FE","no","yes","yes","yes","yes","yes"),
            c("Year FE","no","no","yes","yes","yes","yes"),
            c("Clustered SE","yes","yes","yes","yes","yes","yes"))
)

```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Do, Aug 22, 2024 - 16:08:58

While columns 2 and 3 recap the results of the regressions of Equation 10.1 and Equation 10.2, column 1 presents an estimate of the coefficient of interest in the naive OLS regression of the fatality rate on beer tax without any fixed effects. There we obtain a positive estimate for the coefficient on beer tax that is likely to be upward biased.

The sign of the estimate changes as we extend the model by both entity and time fixed effects in models 2 and 3. Nonetheless, as discussed before, the magnitudes of both estimates may be too large.

The model specifications 4 to 6 include covariates that shall capture the effect of overall state economic conditions as well as the legal framework. Nevertheless, considering **model 4** as the baseline specification including covariates, we observe **four interesting results**:

1. Including these covariates is not leading to a major reduction of the estimated effect of the beer tax. The coefficient is not significantly different from zero at the 10% level, which means that it is considered imprecise.
2. According to this regression model, the minimum legal drinking age is not associated with an effect on traffic fatalities: none of the three dummy variables are significantly different from zero at any common level of significance. Moreover, an *F*-Test of the joint hypothesis that all three coefficients are zero does not reject the null hypothesis. The next code chunk shows how to test this hypothesis:

```

# test if legal drinking age has no explanatory power (Wald test)
linearHypothesis(fat_mod4,
                 c("drinkagec19", "drinkagec20", "drinkagec21"),
                 vcov. = vcovHC)

```

Linear hypothesis test

Hypothesis:

Table 10.1

<i>Dependent variable:</i>						
fatal_rate						
	(1)	(2)	(3)	(4)	(5)	(6)
beertax	0.365*** (0.118)	-0.656** (0.288)	-0.640* (0.350)	-0.445 (0.288)	-0.690** (0.342)	-0.456 (0.298)
drinkagec19				-0.046 (0.057)	-0.065 (0.064)	
drinkagec20				0.004 (0.065)	-0.090 (0.075)	
drinkagec21				-0.028 (0.068)	0.010 (0.080)	
drinkage						-0.002 (0.021)
punishyes				0.038 (0.100)	0.085 (0.108)	0.039 (0.100)
miles				0.00001 (0.00001)	0.00002* (0.00001)	0.00001 (0.00001)
unemp				-0.063*** (0.013)		-0.063*** (0.013)
log(income)				1.816*** (0.616)		1.786*** (0.625)
Constant	1.853*** (0.117)					
State FE	no	yes	yes	yes	yes	yes
Year FE	no	no	yes	yes	yes	yes
Clustered SE	yes	yes	yes	yes	yes	yes
Observations	336	336	336	335	335	335

*Note:*

\*p<0.1; \*\*p<0.05; \*\*\*p<0.01

```
drinkagec19 = 0
drinkagec20 = 0
drinkagec21 = 0
```

Model 1: restricted model

Model 2: fatal\_rate ~ beertax + drinkagec + punish + miles + unemp + log(income)

Note: Coefficient covariance matrix supplied.

```
Res.Df Df  Chisq Pr(>Chisq)
1     276
2     273  3 1.1345    0.7688
```

3. There is no statistical evidence indicating an association between punishment for first offenders and drunk driving: the corresponding coefficient is not significant at the 10% level.

4. The coefficients on the economic variables representing employment rate and income per capita indicate an statistically significant association between these and traffic fatalities. We can check that the employment rate and income per capita coefficients are jointly significant at the 0.1% level.

```
# test if economic indicators have no explanatory power
linearHypothesis(fat_mod4,
                 c("log(income)", "unemp"),
                 vcov. = vcovHC)
```

Linear hypothesis test

```
Hypothesis:
log(income) = 0
unemp = 0
```

Model 1: restricted model

Model 2: fatal\_rate ~ beertax + drinkagec + punish + miles + unemp + log(income)

Note: Coefficient covariance matrix supplied.

```
Res.Df Df  Chisq Pr(>Chisq)
1     275
2     273  2 63.155 1.932e-14 ***
```

---

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 5 omits the economic factors. The result supports the notion that economic indicators should remain in the model as the coefficient on beer tax is sensitive to the inclusion of the latter.

Results for model 6 show that the legal drinking age has little explanatory power and that the coefficient of interest is not sensitive to changes in the functional form of the relation between drinking age and traffic fatalities.

## 10.6 Summary

We have not found statistical evidence to state that severe punishments and an increase in the minimum drinking age could lead to a reduction of traffic fatalities due to drunk driving.

Nonetheless, there seems to be a negative effect of alcohol taxes on traffic fatalities according to our model estimate. However, this estimate is not precise and cannot be interpreted as the causal effect of interest, as there still may be a bias.

There may be omitted variables that differ across states *and* change over time, and this bias remains even though we use a panel approach that controls for entity specific and time invariant unobservables.

## 10.7 R-codes

[methods-sec10.R](#)

## **Part IV**

### **D) Big Data Econometrics**



# 11 Shrinkage Estimation

Shrinkage estimation is a highly valuable technique in the context of high-dimensional regression analysis. It allows for the estimation of regression models with more regressors than observations.

## 11.1 Mean squared error

The key measure of estimation accuracy is the **mean squared error (MSE)**. The MSE of an estimator  $\hat{\theta}$  for a parameter  $\theta$  is

$$mse(\hat{\theta}) = E[(\hat{\theta} - \theta)^2].$$

The MSE can be decomposed into the variance plus squared bias:

$$mse(\hat{\theta}) = \underbrace{E[(\hat{\theta} - E[\hat{\theta}])^2]}_{=Var[\hat{\theta}]} + \underbrace{(E[\hat{\theta}] - \theta)^2}_{=bias(\hat{\theta})^2}$$

*Proof.* Subtracting and adding  $E[\hat{\theta}]$  gives

$$\begin{aligned}(\hat{\theta} - \theta)^2 &= (\hat{\theta} - E[\hat{\theta}] + E[\hat{\theta}] - \theta)^2 \\ &= (\hat{\theta} - E[\hat{\theta}])^2 + 2(\hat{\theta} - E[\hat{\theta}])(\underbrace{E[\hat{\theta}] - \theta}_{bias(\hat{\theta})}) + \underbrace{(E[\hat{\theta}] - \theta)^2}_{=bias(\hat{\theta})^2}.\end{aligned}$$

The middle term is zero after taking the expectation:

$$E[(\hat{\theta} - \theta)^2] = \underbrace{E[(\hat{\theta} - E[\hat{\theta}])^2]}_{=Var[\hat{\theta}]} + 2 \underbrace{E[\hat{\theta} - E[\hat{\theta}]]}_{=0} bias(\hat{\theta}) + bias(\hat{\theta})^2.$$

□

□

For instance, consider an i.i.d. sample  $X_1, \dots, X_n$  with population mean  $E[X_i] = \mu$  and variance  $Var[X_i] = \sigma^2$ . Let's study the sample mean

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n X_i$$

as an estimator of  $\mu$ . You will find that

$$E[\hat{\mu}] = \mu, \quad Var[\hat{\mu}] = \frac{\sigma^2}{n}.$$

*Proof.* By the linearity of the expectation, we have

$$E[\hat{\mu}] = \frac{1}{n} \sum_{i=1}^n \underbrace{E[X_i]}_{\mu} = \mu.$$

The independence of  $X_1, \dots, X_n$  implies

$$Var[\hat{\mu}] = \frac{1}{n^2} Var\left[\sum_{i=1}^n X_i\right] = \frac{1}{n^2} \sum_{i=1}^n Var[X_i] = \frac{\sigma^2}{n}$$

□

□

The sample mean is unbiased for  $\mu$ , i.e.,  $bias(\hat{\mu}) = E[\hat{\mu}] - \mu = 0$ . The MSE equals its variance:

$$mse(\hat{\mu}) = \frac{\sigma^2}{n}.$$

The sample mean is the best unbiased estimator for the population mean in the MSE sense, but there exists estimators with a lower MSE if we allow for a small bias.

## 11.2 A simple shrinkage estimator

Let us shrink our sample mean a bit towards 0 and define the alternative estimator

$$\tilde{\mu} = (1 - w)\hat{\mu}, \quad w \in [0, 1].$$

Setting the shrinkage weight to  $w = 0$  gives  $\tilde{\mu} = \hat{\mu}$  (no shrinkage) and  $w = 1$  gives  $\tilde{\mu} = 0$  (full shrinkage). Our shrinkage estimator has the bias

$$bias(\tilde{\mu}) = E[(1 - w)\hat{\mu}] - \mu = (1 - w) \underbrace{E[\hat{\mu}]}_{=\mu} - \mu = -w\mu.$$

The variance is

$$\text{Var}[\tilde{\mu}] = \text{Var}[(1-w)\hat{\mu}] = (1-w)^2 \text{Var}[\hat{\mu}] = (1-w)^2 \frac{\sigma^2}{n},$$

and the MSE is

$$\text{mse}(\tilde{\mu}) = \text{Var}[\tilde{\mu}] + \text{bias}(\tilde{\mu})^2 = (1-w)^2 \frac{\sigma^2}{n} + w^2 \mu^2.$$

The optimal weight in terms of the MSE is

$$w^* = \frac{1}{1 + n\mu^2/\sigma^2}$$

*Proof.* We take the derivative of  $\text{mse}(\tilde{\mu})$  across  $w$  to obtain the first order condition:

$$-2(1-w)\sigma^2/n + 2w\mu^2 = 0.$$

Solving for  $w$  gives  $w(1 + n\mu^2/\sigma^2) = 1$ . Then,  $w^*$  is the global minimum because the second derivative is  $2\sigma^2/n + 2\mu^2 > 0$ . □

□

For instance, if  $\mu = 1$ ,  $\sigma^2 = 1$ , and  $n = 99$ , we have  $w^* = 0.01$ .

The shrunked sample mean

$$\tilde{\mu}^* = (1-w^*)\hat{\mu} = \frac{n\mu^2/\sigma^2}{1 + n\mu^2/\sigma^2} \frac{1}{n} \sum_{i=1}^n X_i$$

has a lower MSE than the usual sample mean:

$$\text{mse}(\tilde{\mu}^*) = (1-w^*) \frac{\sigma^2}{n} + w^2 \mu^2 < \frac{\sigma^2}{n} = \text{mse}(\hat{\mu})$$

This is a remarkable result because it tells us that the sample mean is not the best we can do to estimate a population mean. The shrunked estimator is more efficient. Is biased, but the biased vanishes asymptotically since  $\lim_{n \rightarrow \infty} w^* = 0$ .

The optimal shrinkage parameter  $w^*$  is infeasible because we do not know  $\mu^2/\sigma^2$ . It is not very useful for empirical practice, and taking sample means is still recommended.

However, the shrinkage principle can be very useful in the context of high-dimensional regression.

## 11.3 High-dimensional regression

Least squares regression works well when the number of regressors  $k$  is small relative to the number of observations  $n$ . In a previous section on “too many regressors”, we discussed how ordinary least squares (OLS) can overfit when  $k$  is too large compared to  $n$ . Specifically, if  $k = n$ , the OLS regression line perfectly fits the data.

Many economic applications involve categorical variables that are transformed into a large number of dummy variables. If we include pairwise interaction terms among  $J$  variables, we get another  $\sum_{i=1}^{J-1} i = J(J-1)/2$  regressors (for example, 190 for  $J=20$  and 4950 for  $J=100$ ).

Accounting for further nonlinearities by adding squared and cubic terms or higher-order interactions can result in thousands or even millions of regressors. Many of these regressors may provide low informational value, but it is difficult to determine a priori which are relevant and which are irrelevant.

If  $k > n$ , the OLS estimator is not uniquely defined because  $\mathbf{X}'\mathbf{X}$  does not have full rank. If  $k \approx n$  the matrix  $\mathbf{X}'\mathbf{X}$  can be near singular, resulting in numerically unstable OLS coefficients or high variance.

For the vector-valued ( $k$ -variate) estimator  $\hat{\boldsymbol{\beta}}_{ols}$  the (conditional) MSE is

$$\begin{aligned} mse(\hat{\boldsymbol{\beta}}_{ols}|\mathbf{X}) &= E[(\hat{\boldsymbol{\beta}}_{ols} - \boldsymbol{\beta})'(\hat{\boldsymbol{\beta}}_{ols} - \boldsymbol{\beta})|\mathbf{X}] \\ &= Var[\hat{\boldsymbol{\beta}}_{ols}|\mathbf{X}] + bias(\hat{\boldsymbol{\beta}}_{ols}|\mathbf{X})(bias(\hat{\boldsymbol{\beta}}_{ols}|\mathbf{X}))', \end{aligned}$$

where, under random sampling, OLS is unbiased:

$$bias(\hat{\boldsymbol{\beta}}_{ols}|\mathbf{X}) = E[\hat{\boldsymbol{\beta}}_{ols}|\mathbf{X}] - \boldsymbol{\beta} = \mathbf{0}.$$

Consequently, the MSE of OLS equals its variance:

$$mse(\hat{\boldsymbol{\beta}}_{ols}|\mathbf{X}) = Var[\hat{\boldsymbol{\beta}}_{ols}|\mathbf{X}] = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{D}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}.$$

## 11.4 Ridge Regression

To avoid that  $(\mathbf{X}'\mathbf{X})^{-1}$  becomes very large or undefined for large  $k$ , we can introduce a shrinkage parameter  $\lambda$  and define the **ridge regression estimator**

$$\hat{\boldsymbol{\beta}}_{ridge} = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}_k)^{-1}\mathbf{X}'\mathbf{Y}. \quad (11.1)$$

This estimator is well defined and does not suffer from multicollinearity problems, even if  $k > n$ . The inverse  $(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}_k)^{-1}$  exists as long as  $\lambda > 0$ . For  $\lambda = 0$ , the ridge estimator coincides with the OLS estimator.

While the OLS estimator is motivated from the minimization problem

$$\min_{\boldsymbol{\beta}} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}),$$

the ridge estimator is the minimizer of

$$\min_{\boldsymbol{\beta}} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}'\boldsymbol{\beta}. \quad (11.2)$$

The minimization problem introduces a penalty for large values of  $\boldsymbol{\beta}$ . The solution is then shrunk towards zero by  $\lambda > 0$ .

## 11.5 Standardization

The regressors and dependent variables are typically standardized:

$$\tilde{X}_{ij} = \frac{X_{ij} - \bar{X}_j}{\sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_{ij} - \bar{X}_j)^2}}, \quad \bar{X}_j = \frac{1}{n} \sum_{i=1}^n X_{ij}$$

It is common practice to standardize the regressors (and dependent variable) in ridge regression.

Without standardization, variables with larger scales (i.e., larger variances) will disproportionately influence the penalty term through  $\lambda\boldsymbol{\beta}'\boldsymbol{\beta} = \lambda \sum_{j=1}^n \beta_j^2$ . Variables with smaller variance may be under-penalized, while those with larger variance may be over-penalized.

Standardization ensures that each variable contributes equally to the penalty term, making the penalty independent of the scale of the variables.

Standardizing makes the coefficient estimates more interpretable, as they will all be on the same scale, which helps in understanding the relative importance of each variable.

## 11.6 Ridge Properties

The bias of the ridge estimator is

$$\text{bias}(\hat{\boldsymbol{\beta}}_{\text{ridge}}|\mathbf{X}) = -\lambda(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}_k)^{-1}\boldsymbol{\beta},$$

and the covariance matrix is

$$\text{Var}[\hat{\boldsymbol{\beta}}_{\text{ridge}}|\mathbf{X}] = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}_k)^{-1}\mathbf{X}'\mathbf{D}\mathbf{X}(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}_k)^{-1}.$$

In the homoskedastic linear regression model, we have

$$mse(\hat{\boldsymbol{\beta}}_{ridge}|\mathbf{X}) < mse(\hat{\boldsymbol{\beta}}_{ols}|\mathbf{X})$$

if  $0 < \lambda < 2\sigma^2/\boldsymbol{\beta}'\boldsymbol{\beta}$ .

Similarly to the sample mean case, the upper bound  $2\sigma^2/\boldsymbol{\beta}'\boldsymbol{\beta}$  does not give practical guidance for selecting  $\lambda$  because  $\boldsymbol{\beta}$  and  $\sigma^2$  are unknown.

## 11.7 Mean squared prediction error

The optimal value for  $\lambda$  minimizes the MSE, but estimating the MSE of the ridge estimator is not straightforward because it depends on the parameter  $\boldsymbol{\beta}$  being estimated. Instead, it is better to focus on the out-of-sample mean squared prediction error (MSPE).

Let  $(Y_1, \mathbf{X}_1), \dots, (Y_n, \mathbf{X}_n)$  be our data set (in-sample observations) with ridge estimator Equation 11.1, and let  $(Y^{oos}, \mathbf{X}^{oos})$  be another observation pair (out-of-sample observation) that is independently drawn from the same population as  $(Y_1, \mathbf{X}_1), \dots, (Y_n, \mathbf{X}_n)$ .

The mean squared prediction error (MSPE) is

$$MSPE(\hat{\boldsymbol{\beta}}_{ridge}) = E[(Y^{oos} - (\mathbf{X}^{oos})'\hat{\boldsymbol{\beta}}_{ridge})^2].$$

Note that  $(Y^{oos}, \mathbf{X}^{oos})$  is independent of  $\hat{\boldsymbol{\beta}}_{ridge}$  because it has not been used for estimation.  $\hat{Y}(\mathbf{X}^{oos}) = (\mathbf{X}^{oos})'\hat{\boldsymbol{\beta}}_{ridge}$  is the predicted value of  $Y^{oos}$ .

To estimate the MSPE, we can use a **split sample**.

- 1) We divide our observations randomly into a training sample (in-sample) of size  $n_{train}$  and a testing sample (out-of-sample) of size  $n_{test}$  with  $n = n_{train} + n_{test}$ :

$$(Y_1^{ins}, \mathbf{X}_1^{ins}), \dots, (Y_{n_{train}}^{ins}, \mathbf{X}_{n_{train}}^{ins}), \quad (Y_1^{oos}, \mathbf{X}_1^{oos}), \dots, (Y_{n_{test}}^{oos}, \mathbf{X}_{n_{test}}^{oos})$$

- 2) We estimate  $\boldsymbol{\beta}$  using the training sample:

$$\hat{\boldsymbol{\beta}}_{ridge}^{ins} = \left( \sum_{i=1}^{n_{train}} \mathbf{X}_i^{ins}(\mathbf{X}_i^{ins})' + \lambda \mathbf{I}_k \right)^{-1} \sum_{i=1}^{n_{train}} \mathbf{X}_i^{ins} Y_i^{ins}.$$

- 3) We evaluate the empirical MSPE using the testing sample,

$$\widehat{MSPE}_{split} = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \left( Y_i^{oos} - (\mathbf{X}_i^{oos})'\hat{\boldsymbol{\beta}}_{ridge}^{ins} \right)^2 \quad (11.3)$$

Steps 2 and 3 are repeated for different values for  $\lambda$ . We select the value for  $\lambda$  that gives the smallest estimated MSPE.

## 11.8 Cross validation

A problem with the split sample estimator is that it highly depends on the choice of the two subsamples. An alternative is to select  $m$  subsamples (folds) and evaluate the MSPE using each fold separately:

### m-fold cross validation

- 1) Divide the sample into  $j = 1, \dots, m$  randomly chosen folds/subsamples of approximately equal size:

$$\begin{aligned} & (Y_1^{(1)}, \mathbf{X}_1^{(1)}), \dots, (Y_{n_1}^{(1)}, \mathbf{X}_{n_1}^{(1)}) \\ & (Y_1^{(2)}, \mathbf{X}_1^{(2)}), \dots, (Y_{n_2}^{(2)}, \mathbf{X}_{n_2}^{(2)}) \\ & \vdots \\ & (Y_1^{(m)}, \mathbf{X}_1^{(m)}), \dots, (Y_{n_m}^{(m)}, \mathbf{X}_{n_m}^{(m)}) \end{aligned}$$

- 2) Select  $j \in \{1, \dots, m\}$  as left-out test sample and use the other subsamples to compute the ridge estimator  $\hat{\beta}_{ridge}^{(-j)}$ , where the  $j$ -th fold is not used.
- 3) Compute Equation 11.3 using the  $j$ -th folds as a test sample, i.e.,

$$\widehat{MSPE}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} \left( Y_i^{(j)} - (\mathbf{X}_i^{(j)})' \hat{\beta}_{ridge}^{(-j)} \right)^2$$

- 4) The  $m$ -fold cross validation estimator is the weighted average over the  $m$  subsample estimates of the MSPE:

$$\widehat{MSPE}_{mfold} = \sum_{j=1}^m \frac{n_j}{n} \widehat{MSPE}_j,$$

where  $n = \sum_{j=1}^m n_j$  is the total number of observations.

- 5) Repeat these steps over a grid of tuning parameters for  $\lambda$ , and select the value for  $\lambda$  that minimizes  $\widehat{MSPE}_{mfold}$ .

Common values for  $m$  are  $m = 5$  and  $m = 10$ . The larger  $m$ , the less biased the estimation of the MSPE is, but also the more computationally expensive the cross validation becomes.

The largest possible value for  $m$  is  $m = n$ , where each observation represents a fold. This is also known as leave-one-out cross validation (LOOVC). LOOVC might be useful for small datasets but is often infeasible for large dataset because of the large computation time.

## 11.9 L2 Regularization: Ridge

The  $\ell_p$ -norm of a vector  $\mathbf{a} = (a_1, \dots, a_k)'$  is defined as

$$\|\mathbf{a}\|_p = \left( \sum_{j=1}^k |a_j|^p \right)^{1/p}.$$

Important special cases are the  $\ell_1$ -norm and  $\ell_2$ -norm:

$$\|\mathbf{a}\|_1 = \sum_{j=1}^k |a_j|, \quad \|\mathbf{a}\|_2 = \left( \sum_{j=1}^k a_j^2 \right)^{1/2} = \sqrt{\mathbf{a}'\mathbf{a}}.$$

The  $\ell_1$ -norm is the sum of absolute values, and the  $\ell_2$ -norm, also known as the Euclidean norm, represents the length of the vector in the Euclidean space.

Ridge regression is also called **L2 regularization** because it penalizes the sum of squared errors by the squared  $\ell_2$ -norm of the coefficient vector,  $\|\boldsymbol{\beta}\|_2^2 = \boldsymbol{\beta}'\boldsymbol{\beta}$ . Ridge is the solution to the minimization problem Equation 11.2, which can be written as

$$\hat{\boldsymbol{\beta}}_{ridge} = \operatorname{argmin}_{\boldsymbol{\beta}} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_2^2.$$

## 11.10 L1 Regularization: Lasso

An alternative approach is **L1 regularization**, also known as **lasso**. The lasso estimator is defined as

$$\hat{\boldsymbol{\beta}}_{lasso} = \operatorname{argmin}_{\boldsymbol{\beta}} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1,$$

where  $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^k |\beta_j|$ .

The **elastic net** estimator is a hybrid method. It combines L1 and L2 regularization using a weight  $0 \leq \alpha \leq 1$ :

$$\hat{\boldsymbol{\beta}}_{net,\alpha} = \operatorname{argmin}_{\boldsymbol{\beta}} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda(\alpha \|\boldsymbol{\beta}\|_1 + (1 - \alpha) \|\boldsymbol{\beta}\|_2^2).$$

This includes ridge ( $\alpha = 0$ ) and lasso ( $\alpha = 1$ ) as special cases.

Ridge has a closed form solution given by Equation 11.1. Lasso and elastic net with  $\alpha > 0$  require numerical solutions by means of quadratic programming. The solution typically involves some zero coefficients.



## 11.11 Implementation in R

Let's consider the `mtcars` dataset, which is available in base R. Have a look at `?mtcars` to see the data description. We estimate a ridge regression model to predict the variable `mpg` (miles per gallon) using the other variables. We consider the values  $\lambda = 0.5$  and  $\lambda = 2.5$ .

Ridge, lasso, and elastic net are implemented in the `glmnet` package. The `glmnet()` function requires matrix-valued data as input. The `model.matrix()` command is useful because it produces the regressor matrix  $\mathbf{X}$  and converts categorical variables into dummy variables.

```
library(glmnet)
Y = mtcars$mpg
X = model.matrix(mpg ~., data = mtcars)[,-1]
dim(X)
```

```
[1] 32 10
```

```
fit.ridge1 = glmnet(x=X, y=Y, alpha=0, lambda = 0.5)
fit.ridge1$beta
```

```
10 x 1 sparse Matrix of class "dgCMatrix"
      s0
cyl  -0.250698757
disp -0.001893223
hp   -0.013079878
drat  0.978514241
wt   -1.902328296
qsec  0.316107066
vs    0.472551434
am    2.113922488
gear  0.631836101
carb -0.661215998
```

```
fit.ridge2 = glmnet(x=X, y=Y, alpha=0, lambda = 2.5)
fit.ridge2$beta
```

```
10 x 1 sparse Matrix of class "dgCMatrix"
      s0
cyl  -0.368541841
disp -0.005184086
```

```

hp    -0.011710951
drat  1.052837310
wt    -1.264016952
qsec  0.164790158
vs    0.755205256
am    1.655241565
gear  0.546732963
carb  -0.560023425

```

You can use the command `coef(fit.ridge1)` to also display the intercept. By default, the regressors are standardized. You can turn off this setting by using the argument `standardize = FALSE`. The  $\ell_2$  norm of the coefficients is small for larger values of  $\lambda$ :

```

c(sqrt(sum(fit.ridge1$beta)),
  sqrt(sum(fit.ridge2$beta)))

```

```
[1] 1.297581 1.401902
```

The lasso estimator ( $\alpha = 1$ ) sets many coefficient equal to zero:

```

fit.lasso = glmnet(x=X, y=Y, alpha=1, lambda = 0.5)
coef(fit.lasso)

```

```

11 x 1 sparse Matrix of class "dgCMatrix"
              s0
(Intercept) 35.88689755
cyl         -0.85565434
disp         .
hp          -0.01411517
drat         0.07603453
wt          -2.67338139
qsec         .
vs           .
am          0.48651385
gear         .
carb        -0.10722338

```

The `cv.glmnet()` command estimates the optimal shrinkage parameter using 10-fold cross validation:

```
set.seed(123) ## for reproducibility
cv.glmnet(x=X, y=Y, alpha = 0)$lambda.min
```

```
[1] 2.746789
```

```
cv.glmnet(x=X, y=Y, alpha = 1)$lambda.min
```

```
[1] 0.8007036
```

We can use ridge and lasso to estimate linear models with more variables than observations. The command `^2` includes all pairwise interaction terms, which produces 55 variables in total. The dataset has  $n = 32$  observations.

```
X.large = model.matrix(mpg ~ . ^2, data = mtcars)[,-1]
dim(X.large)
```

```
[1] 32 55
```

```
fit.ridgelarge = glmnet(x=X.large, y=Y, alpha=0, lambda = 0.5)
coef(fit.ridgelarge)
```

```
56 x 1 sparse Matrix of class "dgCMatrix"
```

```
      s0
(Intercept) 1.315259e+01
cyl         -4.061218e-02
disp        -8.137358e-04
hp          -5.588290e-03
drat         4.386174e-01
wt          -5.547986e-01
qsec         2.308772e-01
vs           6.705889e-01
am           4.379822e-01
gear         8.788479e-01
carb        -1.537294e-01
cyl:disp     6.830897e-05
cyl:hp       1.351742e-04
cyl:drat     2.455464e-02
cyl:wt       -2.621868e-03
cyl:qsec     3.358094e-03
```

cyl:vs	1.591177e-01
cyl:am	6.102385e-02
cyl:gear	3.481957e-02
cyl:carb	7.499023e-04
disp:hp	8.592521e-06
disp:drat	-9.421536e-05
disp:wt	2.191122e-04
disp:qsec	-1.789464e-05
disp:vs	-1.280463e-03
disp:am	-9.043597e-03
disp:gear	-3.601317e-04
disp:carb	-1.255358e-04
hp:drat	-2.086003e-03
hp:wt	4.404097e-04
hp:qsec	-4.347470e-04
hp:vs	-1.858343e-02
hp:am	-2.604620e-03
hp:gear	-3.464491e-04
hp:carb	9.107116e-04
drat:wt	-1.766081e-01
drat:qsec	3.828881e-02
drat:vs	1.123963e-01
drat:am	5.047132e-02
drat:gear	8.294201e-02
drat:carb	-4.770358e-02
wt:qsec	-3.289204e-02
wt:vs	-3.239643e-01
wt:am	-4.197733e-01
wt:gear	-1.890703e-01
wt:carb	-1.497574e-02
qsec:vs	3.114409e-02
qsec:am	5.199239e-02
qsec:gear	7.035311e-02
qsec:carb	-1.859676e-02
vs:am	8.688134e-01
vs:gear	3.311330e-01
vs:carb	-2.768199e-01
am:gear	1.462749e-01
am:carb	1.588431e-01
gear:carb	8.165764e-03

```
fit.lassolarge = glmnet(x=X.large, y=Y, alpha=1, lambda = 0.5)
coef(fit.lassolarge)
```

56 x 1 sparse Matrix of class "dgCMatrix"

```
              s0
(Intercept) 23.655330629
cyl          -0.036308043
disp         .
hp           .
drat         .
wt          -1.301739306
qsec         .
vs           .
am           .
gear         .
carb         .
cyl:disp     .
cyl:hp       .
cyl:drat     .
cyl:wt       .
cyl:qsec     .
cyl:vs       .
cyl:am       .
cyl:gear     .
cyl:carb     .
disp:hp      .
disp:drat    .
disp:wt      .
disp:qsec    .
disp:vs      .
disp:am      .
disp:gear    .
disp:carb    .
hp:drat      .
hp:wt        .
hp:qsec      -0.001328046
hp:vs        .
hp:am        .
hp:gear      .
hp:carb      .
drat:wt      -0.337667877
drat:qsec    0.073725291
```

```
drat:vs      .
drat:am      .
drat:gear    .
drat:carb    .
wt:qsec      .
wt:vs        .
wt:am        .
wt:gear      .
wt:carb      .
qsec:vs      .
qsec:am      .
qsec:gear    0.041623415
qsec:carb    .
vs:am        2.429571498
vs:gear      .
vs:carb      .
am:gear      .
am:carb      .
gear:carb    .
```

## 11.12 R-codes

[methods-sec11.R](#)

# 12 Principal Component Regression

If two regressors are highly correlated, we can typically drop one of the regressors because they mostly contain the same information.

The idea of principal component regression is to exploit the correlations among the regressors to reduce their number while retaining as much of the original information as possible.

## 12.1 Principal Components

The principal components (PC) are linear combinations of the regressor variables that capture as much of the variation in the original variables as possible.

### Principal Components

Let  $\mathbf{X}_i$  be a  $k$ -variate vector of regressor variables.

The **first principal component** is  $P_{i1} = \mathbf{w}'_1 \mathbf{X}_i$ , where  $\mathbf{w}_1$  satisfies

$$\mathbf{w}_1 = \operatorname{argmax}_{\mathbf{w}'\mathbf{w}=1} \operatorname{Var}[\mathbf{w}'\mathbf{X}_i]$$

The **second principal component** is  $P_{i2} = \mathbf{w}'_2 \mathbf{X}_i$ , where  $\mathbf{w}_2$  satisfies

$$\mathbf{w}_2 = \operatorname{argmax}_{\substack{\mathbf{w}'\mathbf{w}=1 \\ \mathbf{w}'\mathbf{w}_1=0}} \operatorname{Var}[\mathbf{w}'\mathbf{X}_i]$$

The  $l$ -th **principal component** is  $P_{il} = \mathbf{w}'_l \mathbf{X}_i$ , where  $\mathbf{w}_l$  satisfies

$$\mathbf{w}_l = \operatorname{argmax}_{\substack{\mathbf{w}'\mathbf{w}=1 \\ \mathbf{w}'\mathbf{w}_1=\dots=\mathbf{w}'\mathbf{w}_{l-1}=0}} \operatorname{Var}[\mathbf{w}'\mathbf{X}_i]$$

A  $k$ -variate regressor vector  $\mathbf{X}_i$  has  $k$  principal components  $P_{i1}, \dots, P_{ik}$  and  $k$  corresponding **principal component weights** or **loadings**  $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k$ .

By definition, the principal components are descendingly ordered by their variance:

$$\operatorname{Var}[P_{i1}] \geq \operatorname{Var}[P_{i2}] \geq \dots \geq \operatorname{Var}[P_{ik}] \geq 0$$

The principal component weights are orthonormal:

$$\mathbf{w}'_i \mathbf{w}_j = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

Moreover,  $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k$  form an orthonormal basis for the  $k$ -dimensional vector space  $\mathbb{R}^k$ . The regressor vector admits the following decomposition into its principal components:

$$\mathbf{X}_i = \sum_{l=1}^k P_{il} \mathbf{w}_l \quad (12.1)$$

The decomposition of a dataset into its principal components is called **principal component analysis (PCA)**.

## 12.2 Analytical PCA Solution

In this subsection, we will use some matrix calculus and eigenvalue theory. To recap the relevant matrix algebra, the following resources will be useful:

- Eigenvalues and Eigenvectors: [https://matrix.svenotto.com/04\\_furtherconcepts.html](https://matrix.svenotto.com/04_furtherconcepts.html)
- Derivative rules for vectors: [https://matrix.svenotto.com/05\\_calculus.html](https://matrix.svenotto.com/05_calculus.html)

The maximization problem for the first principal component is

$$\max_{\mathbf{w}} \text{Var}[\mathbf{w}'\mathbf{X}_i] \quad \text{subject to } \mathbf{w}'\mathbf{w} = 1. \quad (12.2)$$

The variance of interest can be rewritten as

$$\begin{aligned} \text{Var}[\mathbf{w}'\mathbf{X}_i] &= E[(\mathbf{w}'(\mathbf{X}_i - E[\mathbf{X}_i]))^2] \\ &= E[(\mathbf{w}'(\mathbf{X}_i - E[\mathbf{X}_i]))(\mathbf{X}_i - E[\mathbf{X}_i])'\mathbf{w}] \\ &= \mathbf{w}'E[(\mathbf{X}_i - E[\mathbf{X}_i])(\mathbf{X}_i - E[\mathbf{X}_i])']\mathbf{w} \\ &= \mathbf{w}'\Sigma\mathbf{w} \end{aligned}$$

where  $\Sigma = \text{Var}[\mathbf{X}_i]$  is the population covariance matrix of  $\mathbf{X}_i$ . Thus, the constrained maximization problem Equation 12.2 has the Lagrangian

$$\mathcal{L}(\mathbf{w}, \lambda) = \mathbf{w}'\Sigma\mathbf{w} - \lambda(\mathbf{w}'\mathbf{w} - 1),$$

where  $\lambda$  is a Lagrange multiplier.

Recall the derivative rules for vectors: If  $\mathbf{A}$  is a symmetric matrix, then the derivative of  $\mathbf{a}'\mathbf{A}\mathbf{a}$  with respect to  $\mathbf{a}$  is  $2\mathbf{A}\mathbf{a}$ . Therefore, the first order condition with respect to  $\mathbf{w}$  is

$$\Sigma\mathbf{w} = \lambda\mathbf{w}. \quad (12.3)$$

The pair  $(\lambda, \mathbf{w})$  must satisfy the eigenequation Equation 12.3. The lagrange multiplier  $\lambda$  must be an eigenvalue of  $\Sigma$  and the weight vector  $\mathbf{w}$  must be a corresponding eigenvector. By the first order condition with respect to  $\lambda$ ,

$$\mathbf{w}'\mathbf{w} = 1,$$



the eigenvector should be normalized.

Therefore, the variance of interest is

$$\text{Var}[\mathbf{w}'\mathbf{X}_i] = \mathbf{w}'\Sigma\mathbf{w} = \mathbf{w}'(\lambda\mathbf{w}) = \lambda. \quad (12.4)$$

Consequently,  $\text{Var}[\mathbf{w}'\mathbf{X}_i]$  must be an eigenvalue of  $\Sigma$  and  $\mathbf{w}$  is a corresponding normalized eigenvector.

The expression  $\text{Var}[\mathbf{w}'\mathbf{X}_i] = \lambda$  is maximized if we use the largest eigenvalue  $\lambda = \lambda_1$ . Consequently, the variance of the first principal component  $P_{i1}$  is equal to the largest eigenvalue  $\lambda_1$  of  $\Sigma$ , and the first principal component weight  $\mathbf{w}_1$  is a normalized eigenvector corresponding to the eigenvalue  $\lambda_1$ .

Analogously, the second principal component weight  $\mathbf{w}_2$  must also be a normalized eigenvector of  $\Sigma$  with the additional restriction that it is orthogonal to  $\mathbf{w}_1$ . Therefore, it cannot be an eigenvector corresponding to the first eigenvalue, and we use the second largest eigenvalue  $\lambda = \lambda_2$  to maximize Equation 12.4.

The variance of the second principal component  $P_{i2}$  is equal to the second largest eigenvalue  $\lambda_2$  of  $\Sigma$ , and the second principal component weight  $\mathbf{w}_2$  is a corresponding normalized eigenvector.

To continue this pattern, the variance of the  $l$ -th principal component  $P_{il}$  is equal to the  $l$ -th largest eigenvalue  $\lambda_l$  of  $\Sigma$ , and the  $l$ -th principal component weight  $\mathbf{w}_l$  is a corresponding normalized eigenvector.

### Principal Components Solution

Let  $\Sigma$  be the covariance matrix of the  $k$ -variate vector of regressor variables  $\mathbf{X}_i$ , let  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k \geq 0$  be the descendingly ordered eigenvalues of  $\Sigma$ , and let  $\mathbf{v}_1, \dots, \mathbf{v}_k$  be corresponding orthonormal eigenvectors.

- The principal component weights are  $\mathbf{w}_l = \mathbf{v}_l$  for  $l = 1, \dots, k$
- The principal components are  $P_{il} = \mathbf{v}_l'\mathbf{X}_i$ , and they have the properties

$$\text{Var}[P_{il}] = \lambda_l, \quad \text{Cov}(P_{il}, P_{im}) = 0, \quad l \neq m.$$

Principal components are uncorrelated because

$$\begin{aligned} \text{Cov}(P_{im}, P_{il}) &= E[\mathbf{w}_m'(\mathbf{X}_i - E[\mathbf{X}_i])(\mathbf{X}_i - E[\mathbf{X}_i])'\mathbf{w}_l] \\ &= \mathbf{w}_m'\Sigma\mathbf{w}_l = \lambda_m\mathbf{w}_m'\mathbf{w}_l, \end{aligned}$$

where  $\mathbf{w}_m'\mathbf{w}_l = 1$  if  $m = l$  and  $\mathbf{w}_m'\mathbf{w}_l = 0$  if  $m \neq l$

## 12.3 Sample principal components

The covariance matrix  $\Sigma = \text{Var}[\mathbf{X}_i]$  is unknown in practice. Instead, we estimate it from the sample  $\mathbf{X}_1, \dots, \mathbf{X}_n$ :

$$\widehat{\Sigma} = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{X}_i - \bar{\mathbf{X}})(\mathbf{X}_i - \bar{\mathbf{X}})'$$

Let  $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \dots, \hat{\lambda}_k \geq 0$  be the eigenvalues of  $\widehat{\Sigma}$  and let  $\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_k$  be corresponding orthonormal eigenvectors. Then,

- The  $l$ -th sample principal component for observation  $i$  is

$$\widehat{P}_{il} = \widehat{\mathbf{w}}_l' \mathbf{X}_i$$

- The  $l$ -th sample principal component weight vector is

$$\widehat{\mathbf{w}}_l = \hat{\mathbf{v}}_l$$

- The (adjusted) sample variance of the  $l$ -th sample principal components series  $\widehat{P}_{1l}, \dots, \widehat{P}_{nl}$  is  $\hat{\lambda}_l$ , and the sample covariances of different principal components series are zero.

## 12.4 PCA in R

Let's compute the sample principal components of the `mtcars` dataset:

```
pca = prcomp(mtcars)
## the principal components are arranged by columns
pca$x |> head()
```

	PC1	PC2	PC3	PC4	PC5
Mazda RX4	-79.596425	2.132241	-2.153336	-2.7073437	-0.7023522
Mazda RX4 Wag	-79.598570	2.147487	-2.215124	-2.1782888	-0.8843859
Datsun 710	-133.894096	-5.057570	-2.137950	0.3460330	1.1061111
Hornet 4 Drive	8.516559	44.985630	1.233763	0.8273631	0.4240145
Hornet Sportabout	128.686342	30.817402	3.343421	-0.5211000	0.7365801
Valiant	-23.220146	35.106518	-3.259562	1.4005360	0.8029768
	PC6	PC7	PC8	PC9	PC10
Mazda RX4	-0.31486106	-0.098695018	0.07789812	-0.2000092	-0.29008191
Mazda RX4 Wag	-0.45343873	-0.003554594	0.09566630	-0.3533243	-0.19283553
Datsun 710	1.17298584	0.005755581	-0.13624782	-0.1976423	0.07634353
Hornet 4 Drive	-0.05789705	-0.024307168	-0.22120800	0.3559844	-0.09057039

```

Hornet Sportabout -0.33290957  0.106304777  0.05301719  0.1532714 -0.18862217
Valiant           -0.08837864  0.238946304 -0.42390551  0.1012944 -0.03769010
                PC11
Mazda RX4        -0.1057706
Mazda RX4 Wag   -0.1069047
Datsun 710       -0.2668713
Hornet 4 Drive   -0.2088354
Hornet Sportabout 0.1092563
Valiant          -0.2757693

```

```

## the principal components weights
pca$rotation |> head()

```

```

                PC1          PC2          PC3          PC4          PC5
mpg -0.038118199  0.009184847  0.98207085  0.047634784 -0.08832843
cyl  0.012035150 -0.003372487 -0.06348394 -0.227991962  0.23872590
disp 0.899568146  0.435372320  0.03144266 -0.005086826 -0.01073597
hp   0.434784387 -0.899307303  0.02509305  0.035715638  0.01655194
drat -0.002660077 -0.003900205  0.03972493 -0.057129357 -0.13332765
wt   0.006239405  0.004861023 -0.08491026  0.127962867 -0.24354296
                PC6          PC7          PC8          PC9          PC10
mpg -0.143790084 -0.039239174 -2.271040e-02 -0.002790139  0.030630361
cyl -0.793818050  0.425011021  1.890403e-01  0.042677206  0.131718534
disp 0.007424138  0.000582398  5.841464e-04  0.003532713 -0.005399132
hp   0.001653685 -0.002212538 -4.748087e-06 -0.003734085  0.001862554
drat 0.227229260  0.034847411  9.385817e-01 -0.014131110  0.184102094
wt   -0.127142296 -0.186558915 -1.561907e-01 -0.390600261  0.829886844
                PC11
mpg  0.0158569365
cyl -0.1454453628
disp -0.0009420262
hp   0.0021526102
drat 0.0973818815
wt   0.0198581635

```

```

## the standard deviation of the principal components
## are the squareroots of the sample eigenvalues
pca$sdev

```

```

[1] 136.5330479  38.1480776  3.0710166  1.3066508  0.9064862  0.6635411
[7]  0.3085791  0.2859604  0.2506973  0.2106519  0.1984238

```

Principal components are sensitive to the scaling of the data. Consequently, it is recommended to first scale each variable in the dataset to have mean zero and unit variance: `scale(mtcars)`. In this case,  $\Sigma$  is the correlation matrix.

```
pca = mtcars |> scale() |> prcomp()
pca$x |> head()
```

	PC1	PC2	PC3	PC4	PC5
Mazda RX4	-0.64686274	1.7081142	-0.5917309	0.11370221	0.9455234
Mazda RX4 Wag	-0.61948315	1.5256219	-0.3763013	0.19912121	1.0166807
Datsun 710	-2.73562427	-0.1441501	-0.2374391	-0.24521545	-0.3987623
Hornet 4 Drive	-0.30686063	-2.3258038	-0.1336213	-0.50380035	-0.5492089
Hornet Sportabout	1.94339268	-0.7425211	-1.1165366	0.07446196	-0.2075157
Valiant	-0.05525342	-2.7421229	0.1612456	-0.97516743	-0.2116654
	PC6	PC7	PC8	PC9	PC10
Mazda RX4	-0.01698737	-0.42648652	0.009631217	-0.14642303	0.06670350
Mazda RX4 Wag	-0.24172464	-0.41620046	0.084520213	-0.07452829	0.12692766
Datsun 710	-0.34876781	-0.60884146	-0.585255765	0.13122859	-0.04573787
Hornet 4 Drive	0.01929700	-0.04036075	0.049583029	-0.22021812	0.06039981
Hornet Sportabout	0.14919276	0.38350816	0.160297757	0.02117623	0.05983003
Valiant	-0.24383585	-0.29464160	-0.256612420	0.03222907	0.20165466
	PC11				
Mazda RX4	0.17969357				
Mazda RX4 Wag	0.08864426				
Datsun 710	-0.09463291				
Hornet 4 Drive	0.14761127				
Hornet Sportabout	0.14640690				
Valiant	0.01954506				

```
pca$rotation |> head()
```

	PC1	PC2	PC3	PC4	PC5	PC6
mpg	-0.3625305	0.01612440	-0.22574419	-0.022540255	-0.10284468	-0.10879743
cyl	0.3739160	0.04374371	-0.17531118	-0.002591838	-0.05848381	0.16855369
disp	0.3681852	-0.04932413	-0.06148414	0.256607885	-0.39399530	-0.33616451
hp	0.3300569	0.24878402	0.14001476	-0.067676157	-0.54004744	0.07143563
drat	-0.2941514	0.27469408	0.16118879	0.854828743	-0.07732727	0.24449705
wt	0.3461033	-0.14303825	0.34181851	0.245899314	0.07502912	-0.46493964
	PC7	PC8	PC9	PC10	PC11	
mpg	0.367723810	0.754091423	-0.23570162	-0.13928524	-0.12489563	
cyl	0.057277736	0.230824925	-0.05403527	0.84641949	-0.14069544	

```

disp  0.214303077 -0.001142134 -0.19842785 -0.04937979  0.66060648
hp    -0.001495989  0.222358441  0.57583007 -0.24782351 -0.25649206
drat  0.021119857 -0.032193501  0.04690123  0.10149369 -0.03953025
wt    -0.020668302  0.008571929 -0.35949825 -0.09439426 -0.56744870

```

```
pca$sdev
```

```

[1] 2.5706809 1.6280258 0.7919579 0.5192277 0.4727061 0.4599958 0.3677798
[8] 0.3505730 0.2775728 0.2281128 0.1484736

```

## 12.5 Variance of principal components

Since the sample principal components are uncorrelated, the total variation in the data is

$$\text{Var} \left[ \sum_{m=1}^k \widehat{P}_{im} \right] = \sum_{m=1}^k \text{Var}[\widehat{P}_{im}] = \sum_{m=1}^k \widehat{\lambda}_m.$$

The proportion of variance explained by the  $l$ -th principal component is

$$\frac{\text{Var}[\widehat{P}_{il}]}{\text{Var}[\sum_{m=1}^k \widehat{P}_{im}]} = \frac{\widehat{\lambda}_l}{\sum_{m=1}^k \widehat{\lambda}_m}$$

A scree plot is useful to see how much each principal component contributes to the total variation:

```

pcvar = pca$sdev^2
varexpl = pcvar/sum(pcvar)
varexpl

```

```

[1] 0.600763659 0.240951627 0.057017934 0.024508858 0.020313737 0.019236011
[7] 0.012296544 0.011172858 0.007004241 0.004730495 0.002004037

```

```
plot(varexpl)
```

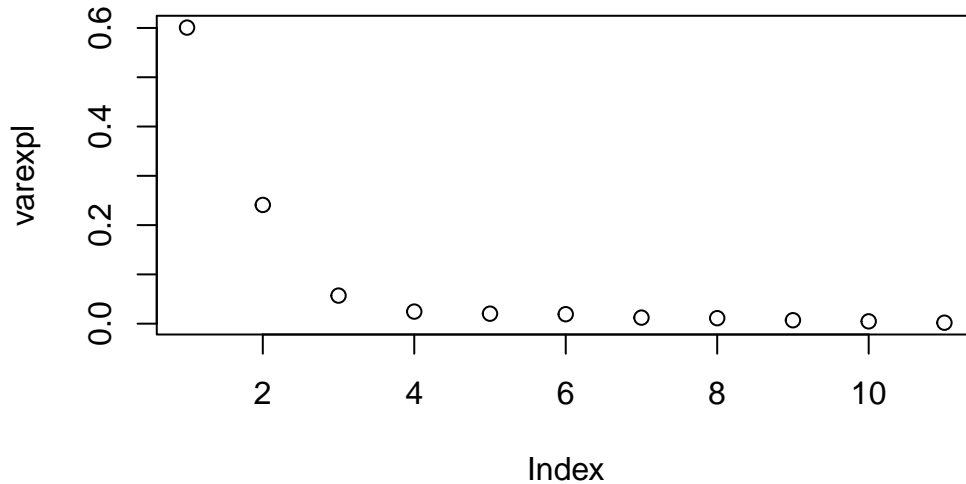
```
cumsum(varexpl)
```

```

[1] 0.6007637 0.8417153 0.8987332 0.9232421 0.9435558 0.9627918 0.9750884
[8] 0.9862612 0.9932655 0.9979960 1.0000000

```

The first principal component explains more than 60% of the variation, the first four explain more than 90% of the variation, the first 6 more than 95%, and the first 9 principal component more than 99% of the variation.



## 12.6 Linear regression with principal components

Principal components can be used to estimate the high-dimensional (large  $k$ ) linear regression model

$$Y_i = \mathbf{X}'_i \boldsymbol{\beta} + u_i, \quad i = 1, \dots, n.$$

Since the principal component weights  $\mathbf{w}_1, \dots, \mathbf{w}_k$  form a basis of  $\mathbb{R}^k$ , the regressors have the basis representation given by Equation 12.1. Similarly, we can represent the coefficient vector in terms of the principal component basis:

$$\boldsymbol{\beta} = \sum_{l=1}^k \theta_l \mathbf{w}_l, \quad \theta_l = \mathbf{w}'_l \boldsymbol{\beta}. \quad (12.5)$$

Inserting in the regression function gives

$$\mathbf{X}'_i \boldsymbol{\beta} = \sum_{l=1}^k \underbrace{\mathbf{X}'_i \mathbf{w}_l}_{=P_{il}} \theta_l,$$

and the regression equation becomes

$$Y_i = \sum_{l=1}^k P_{il} \theta_l + u_i. \quad (12.6)$$

This regression equation is convenient because the regressors  $P_{il}$  are uncorrelated, and OLS estimates for  $\theta_l$  can be inserted back into Equation 12.5 to get an estimate for  $\boldsymbol{\beta}$ .

When  $k$  is large, this approach is still prone to overfitting. The  $k$  principal components of  $\mathbf{X}_i$  explain 100% of its variance, but it may be reasonable to select a smaller number of principal components  $p < k$  that explain 95% or 99% of the variance.

The remaining  $k - p$  principal components explain only 5% or 1% of the variance. The idea is that we truncate the model by assuming that the remaining principal components contain only noise that is uncorrelated with  $Y_i$ .

**Assumption (PC):**  $E[P_{im}Y_i] = 0$  for all  $m = p + 1, \dots, k$ .

Because the principal components are uncorrelated, we have  $\theta_l = E[Y_i P_{il}] / E[P_{il}^2]$ , and, therefore  $\theta_m = 0$  for  $m = p + 1, \dots, k$ . Consequently,

$$\boldsymbol{\beta} = \sum_{l=1}^p \theta_l \boldsymbol{w}_l, \quad (12.7)$$

and Equation 12.6 becomes a factor model with  $p$  factors:

$$Y_i = \sum_{l=1}^p \theta_l P_{il} + u_i = \boldsymbol{P}'_i \boldsymbol{\theta} + u_i,$$

where  $\boldsymbol{P}_i = (P_{i1}, \dots, P_{ip})'$  and  $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)'$ . The least squares estimator of  $\boldsymbol{\theta}$  using the regressors  $\boldsymbol{P}_i$ ,  $i = 1, \dots, n$  can then be inserted to Equation 12.7 to obtain an estimate for  $\boldsymbol{\beta}$ .

In practice, the principal components are unknown and must be replaced by the first  $p$  sample principal components

$$\widehat{\boldsymbol{P}}_i = (\widehat{P}_{i1}, \dots, \widehat{P}_{ip})', \quad \widehat{P}_{il} = \widehat{\boldsymbol{w}}'_l \boldsymbol{X}_i.$$

The feasible least squares estimator for  $\boldsymbol{\theta}$  is

$$\widehat{\boldsymbol{\theta}} = (\widehat{\theta}_1, \dots, \widehat{\theta}_p)' = \left( \sum_{i=1}^n \widehat{\boldsymbol{P}}_i \widehat{\boldsymbol{P}}'_i \right)^{-1} \sum_{i=1}^n \widehat{\boldsymbol{P}}_i Y_i,$$

and the principal components estimator for  $\boldsymbol{\beta}$  is

$$\widehat{\boldsymbol{\beta}}_{pc} = \sum_{l=1}^p \widehat{\theta}_l \widehat{\boldsymbol{w}}_l.$$

## 12.7 Selecting the number of factors

To select the number of principal components, one practical approach is to choose those that explain a pre-specified percentage (90-99%) of the total variance.

```
Y = mtcars$mpg
X = model.matrix(mpg ~., data = mtcars)[,-1] |> scale()
## principal component analysis
pca = prcomp(X)
P = pca$x #full matrix of all principal components
```

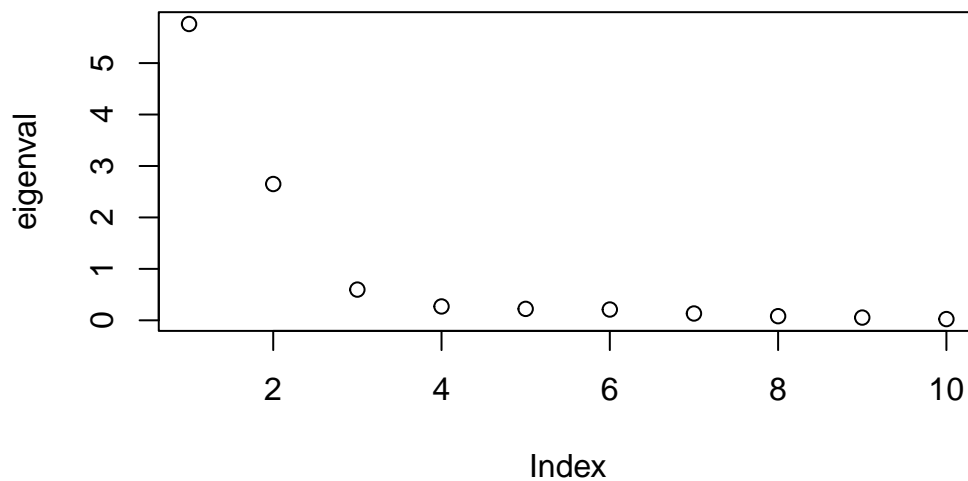
```
## variance explained
eigenval = pca$sdev^2
varexpl = eigenval/sum(eigenval)
cumsum(varexpl)
```

```
[1] 0.5760217 0.8409861 0.9007075 0.9276582 0.9498832 0.9708950 0.9841870
[8] 0.9922551 0.9976204 1.0000000
```

The first four principal components explain more than 92% of the variance, and the first seven more than 98%.

Another method involves creating a scree plot to display the eigenvalues (variances) for each principal component and identifying the point where the eigenvalues sharply drop (elbow point).

```
plot(eigenval)
```



We find an elbow at four principal components.

Selecting the number of principal components, similar to shrinkage estimation, involves balancing variance and bias. If the Assumption (PC) holds, the PC estimator is unbiased; if it doesn't, a small bias is introduced. Increasing the number of components  $p$  reduces bias but increases variance, while decreasing  $p$  reduces variance but increases bias.

Similarly to the shrinkage parameter in ridge and lasso estimation, the number of factors  $p$  can be treated as a tuning parameter. We can use  $m$ -fold cross validation to select  $p$  such that the MSE is minimized.



## 12.8 R-codes

[methods-sec12.R](#)

# 13 Case Study III: Big Data

```
library(readxl) # for reading Excel files
library(tidyverse) # for data manipulation and visualization
library(caret) # for cross-validation
library(glmnet) # for ridge and lasso regression
```

## 13.1 Introduction

In this case study, we will explore the empirical application of prediction methods using the test score data set from California elementary schools. The data set includes detailed information on various school and community characteristics, which allows us to experiment with different regression models and prediction techniques.

We aim to predict fifth-grade test scores using three different sets of predictors: a small set with only a few variables, a large set with many variables, and a very large set with an extensive number of predictors, including interactions, squares, and cubes of the main variables.

## 13.2 Data Set Description

The primary data set contains data on 3932 elementary schools in California from 2013. The raw data and its variable descriptions can be downloaded [here](#) (CA\_Schools\_EE14).

Data has been splitted into three sets based on the number of predictors:

1. **Small Data Set:** Contains 4 variables that have been commonly used in previous studies:
  - Student-teacher ratio (`str_s`)
  - Median income of the local population (`med_income_z`)
  - Teachers average years of experience (`te_avgyr_s`)
  - Instructional expenditures per student (`exp_1000_1999_d`)

2. **Large Data Set:** Contains 817 predictors including 38 main variables, their squares, cubes, and all possible pairwise interactions. The main variables cover student demographics, teacher characteristics, school funding, and expenditure metrics, including fractions of students by eligibility and ethnicity, teacher experience, school expenditures, and district-level financial data.
3. **Very Large Data Set:** Contains 2065 predictors, which includes additional demographic variables, their squares, cubes, and interactions with the binary variables describing school characteristics.

For simplicity, these data sets have been prepared for you and can be accessed directly from the xlsx files (see Ilias course):

```
data_large = read_xlsx("data_large.xlsx")
```

We will use the large data set here, but you can apply the same code to the small and the very large data set.

```
data_small = read_xlsx("data_small.xlsx")
data_verylarge = read_xlsx("data_verylarge.xlsx")
```

[Here](#) you find a script to create your own data sets based on the raw data.

## 13.3 Methods

We will use four different methods to estimate the predictive models:

1. Ordinary Least Squares (OLS)
2. Ridge Regression
3. Lasso Regression
4. Principal Components Regression (PCR)

These methods will be applied to each of the data sets to evaluate their performance in predicting out-of-sample test scores. The main metric used to assess the prediction accuracy is the Root Mean Squared Error (RMSE).

## 13.4 Data Preparation

The data is processed and divided into training and test sets (50/50 split), with 1966 observations each. The predictor variables are standardized, and the response variable is the average fifth-grade test score at the school.

```
## Select the data set to be used (large dataset in this case)
mydata = data_large

## Split 50/50 into training and test sets
set.seed(123) #for reproducibility
train_indices = sample(1:nrow(mydata), size = 0.5*nrow(mydata))
train_data = mydata[train_indices, ]
test_data = mydata[-train_indices, ]

## Standardize/scale the predictor variables
train_response = train_data$testscore
train_predictors = train_data |> select(-testscore) |> scale()
test_response = test_data$testscore
test_predictors = test_data |> select(-testscore) |> scale()
```

## 13.5 Cross-Validation for Tuning Parameters

### 13.5.1 Ridge Regression

For ridge regression, we perform 10-fold cross-validation to select the optimal shrinkage parameter ( $\lambda$ ). The tuning parameter that minimizes the cross-validated RMSE is chosen for the final model.

The `train()` function from the `caret` package can be used for cross validation. We set `alpha` to 0 for ridge regression and we try out different `lambdas` specified in the sequence given by `lambdagrid`.

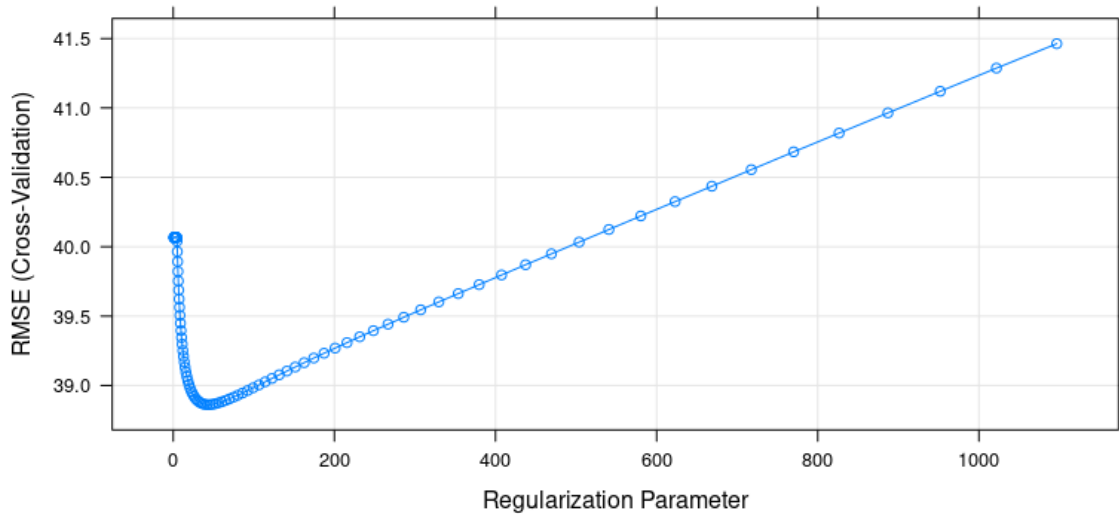
```
# grid for lambdas over which to cross validate. the finer the grid, the longer it takes
lambdagrid = exp(seq(0,7,length=100))

cv.ridge = train(
  x=train_predictors,
  y=train_response,
  method = "glmnet",
  metric = "RMSE",
```

```

tuneGrid = expand.grid(alpha = 0, lambda = lambdagrid),
trControl = trainControl(method = "cv", number = 10) # 10-fold cv
)
plot(cv.ridge) # plot the cv results for ridge

```



### *Cross-Validation Results for Ridge Regression*

```

# print best tuning parameters for ridge
cv.ridge$bestTune

```

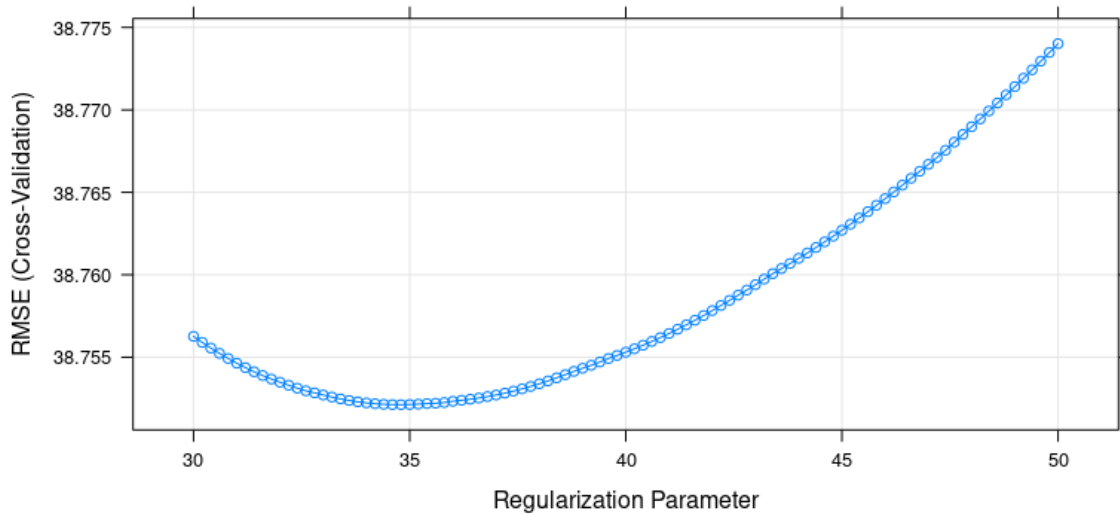
```
[1] 42.41384
```

We can also fine-tune lambda in a specific region to get a better picture, e.g. around 30-50:

```

cv.ridge.finetune = train(
  x=train_predictors,
  y=train_response,
  method = "glmnet",
  metric = "RMSE",
  tuneGrid = expand.grid(alpha = 0, lambda = seq(30, 50, length = 101)),
  trControl = trainControl(method = "cv", number = 10)
)
plot(cv.ridge.finetune)

```



```
# the result may be slightly different each time because the folds are sampled randomly
cv.ridge.finetune$bestTune
```

```
[1] 34.8
```

The built-in cross-validation method of the `glmnet` package gives a similar tuning parameter:

```
cv.ridge2 = cv.glmnet(x=train_predictors, y=train_response, alpha=0)
cv.ridge2$lambda.min
```

```
[1] 39.27491
```

### 13.5.2 Lasso Regression

Similar to ridge regression, we use 10-fold cross-validation to find the optimal shrinkage parameter for the lasso model.

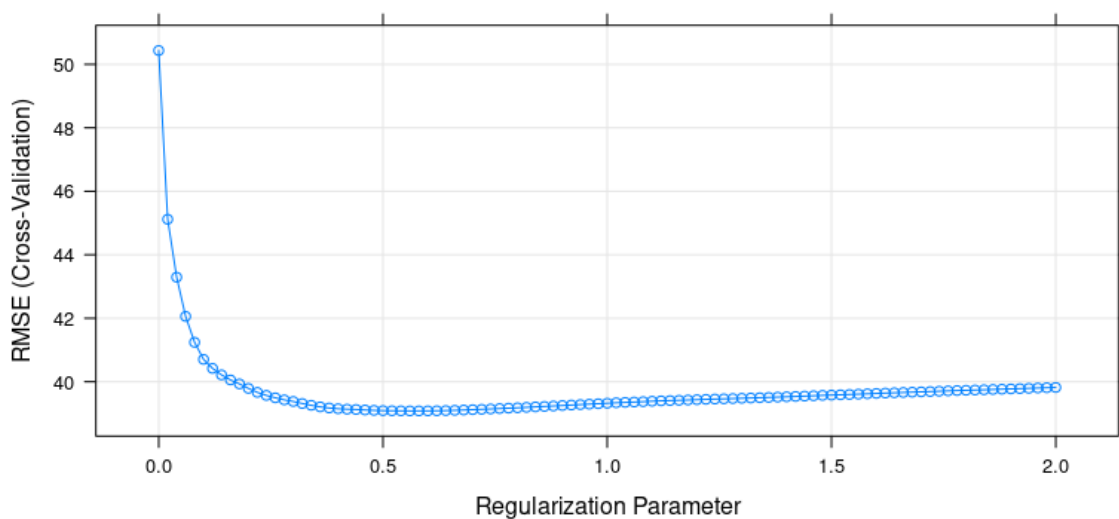
The procedure is the same, but now we set `alpha = 1` for lasso.

```

lambdagrid = seq(0, 2, length = 101)

cv.lasso = train(
  x = train_predictors,
  y = train_response,
  method = "glmnet",
  metric = "RMSE",
  tuneGrid = expand.grid(alpha = 1, lambda = lambdagrid),
  trControl = trainControl(method = "cv", number = 10) # 10-fold cv
)
plot(cv.lasso)

```



*Cross-Validation Results for Lasso Regression*

```

# print best tuning parameters for lasso
cv.lasso$bestTune

```

```
[1] 0.56
```

Again, we can alternatively use the `glmnet` package with its built-in cross-validation method:

```

cv.lasso2 = cv.glmnet(x=train_predictors, y=train_response, alpha=1)
cv.lasso2$lambda.min

```

```
[1] 0.4841995
```

### 13.5.3 Principal Components Regression

For principal component regression (PCR), we use principal components analysis to determine the number of components that explain a significant amount of variance in the predictors. We then use 10-fold cross-validation to select the number of principal components that balances bias and variance for the regression model.

```
## Principal Component Analysis
pca_result = prcomp(train_predictors)
X_pca = pca_result$x # Full matrix of all principal component scores
```

We then use a subset of the principal components as predictors in a regression model. Here, we start by using the first four principal components.

```
## Run a PC-regression with ncomp=4 principal components
ncomp = 4
data_pca = data.frame(y = train_response, X_pca[, 1:ncomp])
lm(y~., data = data_pca)
```

Call:

```
lm(formula = y ~ ., data = data_pca)
```

Coefficients:

(Intercept)	PC1	PC2	PC3	PC4
752.792	-2.480	-2.183	2.294	1.436

By regressing on the first few principal components, we reduce the dimensionality of the problem. This can help prevent overfitting, as the components capture the most important information from the original predictors while ignoring the noise.

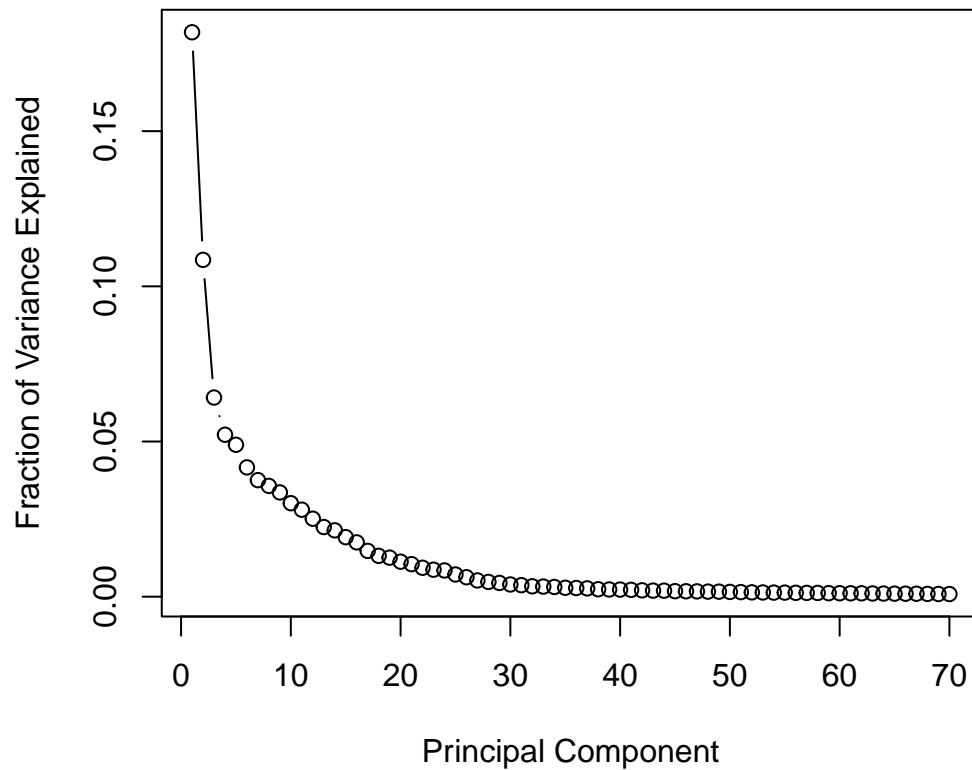
To decide how many principal components to use, we can plot the scree plot, which shows the fraction of total variance explained by each principal component.

```
## Scree Plot: Fraction of variance explained
var_explained = pca_result$sdev^2 / sum(pca_result$sdev^2)
plot(var_explained[1:70], type="b",
     xlab = "Principal Component", ylab = "Fraction of Variance Explained")
```

*Scree Plot: Fraction of variance explained*

The scree plot indicates an elbow around 30-40 components. We can also determine the number of principal components needed to explain a specific percentage of the total variance.





```
## number of components needed to explain 90% of variance  
which(cumsum(var_explained) > 0.90)[1]
```

```
[1] 34
```

```
## number of components needed to explain 95% of variance  
which(cumsum(var_explained) > 0.95)[1]
```

```
[1] 62
```

```
## number of components needed to explain 99% of variance  
which(cumsum(var_explained) > 0.99)[1]
```

```
[1] 157
```

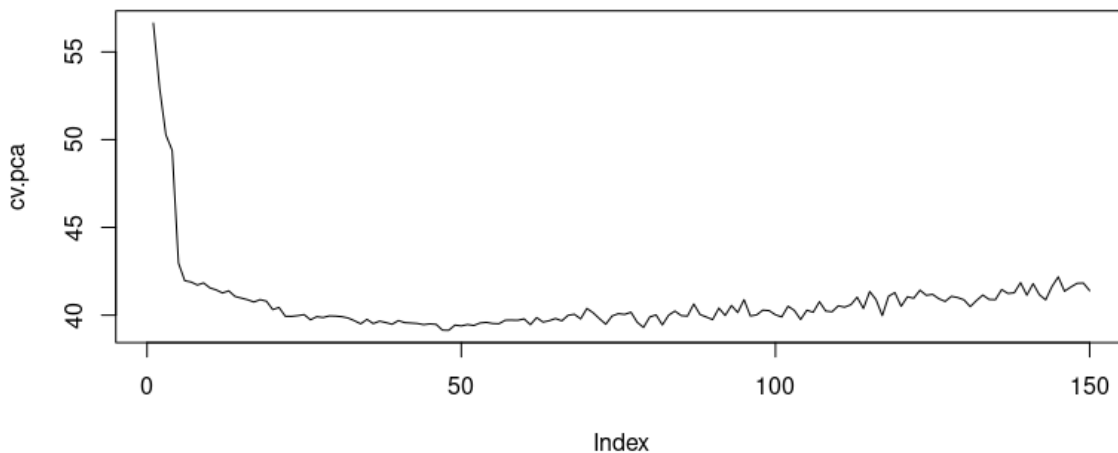
Retaining components that explain 90%-95% of the variance is a common practice to ensure that most of the underlying structure of the data is preserved while omitting unnecessary noise.

Finally, we use cross-validation to find the optimal number of principal components that minimizes the mean squared prediction error.

```
## PCR 10-fold cross-validation
myfunc.cv pca = function(p){
  data_pca = data.frame(y = train_response, X_pca[,1:p])
  cv = train(
    y ~ ., data = data_pca,
    method = "lm",
    metric = "RMSE",
    trControl = trainControl(method = "cv", number = 10)
  )
  return(cv$results$RMSE)
}
# Iterate function crossval over ncomp = 1, ..., maxcomp
maxcomp = 150 # select not more than number of variables (for data_small select <=4)
cv.pca = sapply(1:maxcomp, myfunc.cv pca) # sapply is useful for iterating over function argu

# Find the number of components with the lowest RMSPE
which.min(cv.pca)
plot(cv.pca, type="l")
```

[1] 48



*Cross-Validation Results for Principal Components Regression*

## 13.6 Summary

We can now summarize the tuning parameters that were determined through cross-validation for each predictive method and each data set.

Estimated  $\lambda$  or  $p$ :

Data Set	Ridge	Lasso	PCR
Small ( $k = 4$ )	3.33	0.4	4
Large ( $k = 817$ )	42.41	0.56	48
Very Large ( $k = 2065$ )	437.38	0.58	73

The parameters are selected by minimizing the MSPE through 10-fold cross-validation using the 1966 observations in the training sample.

We reserved a separate test sample of 1966 observations, independent of the training sample. To evaluate the predictive performance, we use the estimated models from the training sample to predict  $Y_i$  from  $\mathbf{X}_i$  for all  $i$  in the test sample and then assess the mean prediction errors. As a baseline competitor, we include the OLS predictor.

```
## OLS
fit.ols = lm(testscore ~., data = train_data)
oospred.ols = predict(fit.ols, newdata = test_data)

## Ridge
lambda.ridge = 42.41
fit.ridge = glmnet(x=train_predictors, y=train_response, alpha=0, lambda = lambda.ridge)
oospred.ridge = predict(fit.ridge, test_predictors)

## LASSO
lambda.lasso = 0.56
fit.lasso = glmnet(x=train_predictors, y=train_response, alpha=1, lambda = lambda.lasso)
oospred.lasso = predict(fit.lasso, test_predictors)

## PCA
p.pcr = 48
pca_result = prcomp(train_predictors)
data_pca = data.frame(y=train_response, pca_result$x[,1:p.pcr])
```

```

fit.pcr = lm(y~., data = data_pca)
## Estimated principal component weights
w = pca_result$rotation
## Principal components for the training data (coincides with pca_result$x):
P.train = train_predictors %*% w
## Principal components for the test data:
P.test = test_predictors %*% w
datapca.test = data.frame(y=test_response, P.test[,1:p.pcr])
## out of sample prediction
oospred.pca = predict(fit.pcr, newdata = datapca.test)

```

```

# Out-of-sample RMSPE computation
# OLS
sqrt(mean((test_response - oospred.ols)^2))

```

[1] 61.8867

```

# Ridge
sqrt(mean((test_response - oospred.ridge)^2))

```

[1] 39.10138

```

# Lasso
sqrt(mean((test_response - oospred.lasso)^2))

```

[1] 39.33968

```

# PCA
sqrt(mean((test_response - oospred.pca)^2))

```

[1] 39.82125

We can now present all out-of-sample MSPEs for all data sets in a summary table. We select the tuning parameters  $\lambda$  and  $p$  from the table above.

Data Set	OLS	Ridge	Lasso	PCR
Small	52.44	52.47	52.45	52.48

Data Set	OLS	Ridge	Lasso	PCR
Large	61.89	39.1	39.34	39.82
Very Large	-	39.39	39.42	39.95

OLS is infeasible in the very large dataset because  $k > n$ . Ridge, lasso, and PCR perform similarly well, in particular in the large and the very large data set.

## 13.7 R-codes

[methods-sec13.R](#)

## **Part V**

# **E) Time Series Methods**

# 14 Forecasting Models

## 14.1 Basic time series models

Consider two time series  $Y_t$  and  $Z_t$  for  $t = 1, \dots, T$ . The index  $t$  is used instead of  $i$  because observations correspond to time points, not individuals.  $T$  represents the sample size, i.e., the number of observed time periods.

Here are some core linear time series forecasting models:

- 1) **Autoregressive model**, AR( $p$ ):

$$Y_t = \alpha_0 + \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + \dots + \alpha_p Y_{t-p} + u_t$$

- 2) **Distributed lag model**, DL( $q$ ):

$$Y_t = \alpha + \delta_1 Z_{t-1} + \dots + \delta_q Z_{t-q} + u_t$$

- 3) **Autoregressive distributed lag model**, ADL( $p, q$ ):

$$Y_t = \alpha_0 + \alpha_1 Y_{t-1} + \dots + \alpha_p Y_{t-p} + \delta_1 Z_{t-1} + \dots + \delta_q Z_{t-q} + u_t$$

In these equations,  $p$  is the number of lags of the dependent variable  $Y_t$ ,  $q$  is the number of lags of the explanatory variable  $Z_t$ , and  $u_t$  is a mean zero error (shock) that is conditional mean independent of the regressors. These models can be estimated by OLS.

The AR, DL, and ADL models can be used for forecasting because the regressors lie in the past relative to the dependent variable. Further exogenous variables can also be included.

If the model parameters are known and the sample is given for  $t = 1, \dots, T$ , we can compute the out-of-sample predicted value for  $t = T + 1$ , which defines a population forecast for  $Y_{T+1}$  (1-step ahead forecast). E.g. in the ADL model, we have

$$Y_{T+1|T} = \alpha_0 + \alpha_1 Y_T + \dots + \alpha_p Y_{T-p+1} + \delta_1 Z_T + \dots + \delta_q Z_{T-q+1}.$$

Using estimated coefficients, we have the **1-step ahead forecast**

$$\widehat{Y}_{T+1|T} = \widehat{\alpha}_0 + \widehat{\alpha}_1 Y_T + \dots + \widehat{\alpha}_p Y_{T-p+1} + \widehat{\delta}_1 Z_T + \dots + \widehat{\delta}_q Z_{T-q+1}.$$

Because regression models with time series variables typically include lags of variables, we call them **dynamic regression models**.

## 14.2 Dynamic regressions

In general, let  $Y_t$  be the univariate dependent time series variable, and  $\mathbf{X}_t = (X_{1t}, \dots, X_{kt})'$  be the  $k$ -variate regressor time series vector. A time series regression is a linear regression model

$$Y_t = \mathbf{X}_t' \boldsymbol{\beta} + u_t, \quad t = 1, \dots, T, \quad (14.1)$$

where the error term satisfies  $E[u_t | \mathbf{X}_t] = 0$ .

The vector of regressors  $\mathbf{X}_t$  may contain multiple exogenous variables and its lags, but also lags of the dependent variable. E.g., in the ADL( $p, q$ ) model, we have  $k = p + q + 1$  and

$$\begin{aligned} \mathbf{X}_t &= (1, Y_{t-1}, \dots, Y_{t-p}, Z_{t-1}, \dots, Z_{t-q})', \\ \boldsymbol{\beta} &= (\alpha_0, \alpha_1, \dots, \alpha_p, \delta_1, \dots, \delta_q)'. \end{aligned}$$

The OLS estimator is

$$\hat{\boldsymbol{\beta}} = \left( \sum_{t=1}^T \mathbf{X}_t \mathbf{X}_t' \right)^{-1} \left( \sum_{t=1}^T \mathbf{X}_t Y_t \right).$$

To compute  $\mathbf{X}_1$  in  $\hat{\boldsymbol{\beta}}$  for dynamic models, we need a few additional observations at the beginning of the sample. I.e., for the ADL( $p, q$ ) model,  $Y_t$  must be observed from  $t = 1 - p, \dots, T$  and  $Z_t$  from  $t = 1 - q, \dots, T$ .

## 14.3 One-step ahead forecast

In forecasting models, the regressors contain only variables that lie in the past of  $t$ . Therefore,  $\mathbf{X}_{T+1}$  is known from the sample, and the **one-step ahead forecast** can be computed as

$$\widehat{Y}_{T+1|T} = \mathbf{X}_{T+1}' \hat{\boldsymbol{\beta}}.$$

The **forecast error** is

$$\begin{aligned} f_{T+1|T} &= Y_{T+1} - \widehat{Y}_{T+1|T} \\ &= \mathbf{X}_{T+1}' \boldsymbol{\beta} + u_{T+1} - \mathbf{X}_{T+1}' \hat{\boldsymbol{\beta}} \\ &= u_{T+1} + \mathbf{X}_{T+1}' (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) \\ &\approx u_{T+1}. \end{aligned}$$

The last step holds for large  $T$  if the OLS estimator  $\hat{\boldsymbol{\beta}}$  is consistent.

To obtain a  $(1 - \alpha)$ -forecast interval  $I_{(T+1|T; 1-\alpha)}$  with

$$\lim_{T \rightarrow \infty} P\left(Y_{T+1} \in I_{(T+1|T; 1-\alpha)}\right) = 1 - \alpha, \quad (14.2)$$



we require a distributional assumption for the error term. Unfortunately, the central limit theorem will not help us here. The most common assumption is to assume normally distributed errors  $u_t \sim \mathcal{N}(0, \sigma^2)$ , but also a t-distribution is possible if there is evidence that the errors have a higher kurtosis.

If the errors are normally distributed and the OLS estimator is consistent, it follows that

$$\lim_{T \rightarrow \infty} P\left(\frac{f_{T+1|T}}{s_{\hat{u}}} \leq c\right) = \Phi(c),$$

where  $\Phi$  is the standard normal CDF. Consequently, Equation 14.2 holds with

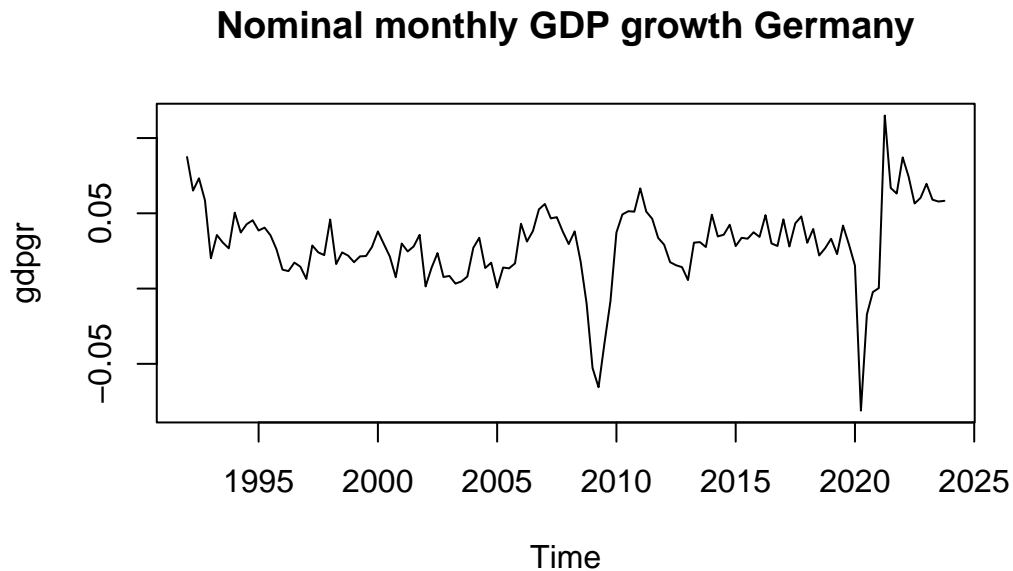
$$I_{(T+1|T; 1-\alpha)} = \left[ \widehat{Y}_{T+1|T} - z_{(1-\frac{\alpha}{2})} s_{\hat{u}}; \widehat{Y}_{T+1|T} + z_{(1-\frac{\alpha}{2})} s_{\hat{u}} \right],$$

where  $s_{\hat{u}}$  is the standard error of regression (SER).

## 14.4 Dynamic models in R

### 14.4.1 An AR model for GDP

```
library(dynlm) # for dynamic linear models
data(gdpgr, package = "teachingdata")
plot(gdpgr, main = "Nominal monthly GDP growth Germany")
```



Consider the AR(4) model for GDP growth:

$$gdp_t = \alpha_0 + \alpha_1 gdp_{t-1} + \alpha_2 gdp_{t-2} + \alpha_3 gdp_{t-3} + \alpha_4 gdp_{t-4} + u_t.$$

One challenge is to define the lagged regressors correctly. Because we have four lags, we need  $T + 4$  observations from  $t = -3, \dots, T$  to compute the OLS estimate. The `embed()` function is useful to get the regressor matrix with the shifted variables with lags from 1 to 4:

```
embed(gdpgr, 5)
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	0.0201337715	0.0586045514	0.0732642826	0.0651053628	0.0874092348
[2,]	0.0355929601	0.0201337715	0.0586045514	0.0732642826	0.0651053628
[3,]	0.0305325110	0.0355929601	0.0201337715	0.0586045514	0.0732642826
[4,]	0.0267275508	0.0305325110	0.0355929601	0.0201337715	0.0586045514
[5,]	0.0504532397	0.0267275508	0.0305325110	0.0355929601	0.0201337715
[6,]	0.0372759162	0.0504532397	0.0267275508	0.0305325110	0.0355929601
[7,]	0.0427747084	0.0372759162	0.0504532397	0.0267275508	0.0305325110
[8,]	0.0453798176	0.0427747084	0.0372759162	0.0504532397	0.0267275508
[9,]	0.0385844643	0.0453798176	0.0427747084	0.0372759162	0.0504532397
[10,]	0.0404915385	0.0385844643	0.0453798176	0.0427747084	0.0372759162
[11,]	0.0353187251	0.0404915385	0.0385844643	0.0453798176	0.0427747084
[12,]	0.0260446862	0.0353187251	0.0404915385	0.0385844643	0.0453798176
[13,]	0.0125448113	0.0260446862	0.0353187251	0.0404915385	0.0385844643
[14,]	0.0116162653	0.0125448113	0.0260446862	0.0353187251	0.0404915385
[15,]	0.0172743837	0.0116162653	0.0125448113	0.0260446862	0.0353187251
[16,]	0.0145381167	0.0172743837	0.0116162653	0.0125448113	0.0260446862
[17,]	0.0064074433	0.0145381167	0.0172743837	0.0116162653	0.0125448113
[18,]	0.0286181410	0.0064074433	0.0145381167	0.0172743837	0.0116162653
[19,]	0.0240593231	0.0286181410	0.0064074433	0.0145381167	0.0172743837
[20,]	0.0222180983	0.0240593231	0.0286181410	0.0064074433	0.0145381167
[21,]	0.0458560754	0.0222180983	0.0240593231	0.0286181410	0.0064074433
[22,]	0.0162997134	0.0458560754	0.0222180983	0.0240593231	0.0286181410
[23,]	0.0240238678	0.0162997134	0.0458560754	0.0222180983	0.0240593231
[24,]	0.0219259244	0.0240238678	0.0162997134	0.0458560754	0.0222180983
[25,]	0.0175312705	0.0219259244	0.0240238678	0.0162997134	0.0458560754
[26,]	0.0213872237	0.0175312705	0.0219259244	0.0240238678	0.0162997134
[27,]	0.0215996987	0.0213872237	0.0175312705	0.0219259244	0.0240238678
[28,]	0.0275603181	0.0215996987	0.0213872237	0.0175312705	0.0219259244
[29,]	0.0379630756	0.0275603181	0.0215996987	0.0213872237	0.0175312705
[30,]	0.0295828692	0.0379630756	0.0275603181	0.0215996987	0.0213872237
[31,]	0.0213309511	0.0295828692	0.0379630756	0.0275603181	0.0215996987

[32,]	0.0075237667	0.0213309511	0.0295828692	0.0379630756	0.0275603181
[33,]	0.0299392612	0.0075237667	0.0213309511	0.0295828692	0.0379630756
[34,]	0.0246649062	0.0299392612	0.0075237667	0.0213309511	0.0295828692
[35,]	0.0280194737	0.0246649062	0.0299392612	0.0075237667	0.0213309511
[36,]	0.0356734942	0.0280194737	0.0246649062	0.0299392612	0.0075237667
[37,]	0.0014322600	0.0356734942	0.0280194737	0.0246649062	0.0299392612
[38,]	0.0138416969	0.0014322600	0.0356734942	0.0280194737	0.0246649062
[39,]	0.0235678950	0.0138416969	0.0014322600	0.0356734942	0.0280194737
[40,]	0.0077007205	0.0235678950	0.0138416969	0.0014322600	0.0356734942
[41,]	0.0083826875	0.0077007205	0.0235678950	0.0138416969	0.0014322600
[42,]	0.0032922145	0.0083826875	0.0077007205	0.0235678950	0.0138416969
[43,]	0.0047364761	0.0032922145	0.0083826875	0.0077007205	0.0235678950
[44,]	0.0079743278	0.0047364761	0.0032922145	0.0083826875	0.0077007205
[45,]	0.0270819565	0.0079743278	0.0047364761	0.0032922145	0.0083826875
[46,]	0.0337685936	0.0270819565	0.0079743278	0.0047364761	0.0032922145
[47,]	0.0136382992	0.0337685936	0.0270819565	0.0079743278	0.0047364761
[48,]	0.0172059191	0.0136382992	0.0337685936	0.0270819565	0.0079743278
[49,]	0.0006541173	0.0172059191	0.0136382992	0.0337685936	0.0270819565
[50,]	0.0139693816	0.0006541173	0.0172059191	0.0136382992	0.0337685936
[51,]	0.0134547959	0.0139693816	0.0006541173	0.0172059191	0.0136382992
[52,]	0.0167457829	0.0134547959	0.0139693816	0.0006541173	0.0172059191
[53,]	0.0430703460	0.0167457829	0.0134547959	0.0139693816	0.0006541173
[54,]	0.0312473976	0.0430703460	0.0167457829	0.0134547959	0.0139693816
[55,]	0.0382467143	0.0312473976	0.0430703460	0.0167457829	0.0134547959
[56,]	0.0526367957	0.0382467143	0.0312473976	0.0430703460	0.0167457829
[57,]	0.0561884737	0.0526367957	0.0382467143	0.0312473976	0.0430703460
[58,]	0.0466371217	0.0561884737	0.0526367957	0.0382467143	0.0312473976
[59,]	0.0474469210	0.0466371217	0.0561884737	0.0526367957	0.0382467143
[60,]	0.0378900574	0.0474469210	0.0466371217	0.0561884737	0.0526367957
[61,]	0.0295752497	0.0378900574	0.0474469210	0.0466371217	0.0561884737
[62,]	0.0379954321	0.0295752497	0.0378900574	0.0474469210	0.0466371217
[63,]	0.0178515785	0.0379954321	0.0295752497	0.0378900574	0.0474469210
[64,]	-0.0099977546	0.0178515785	0.0379954321	0.0295752497	0.0378900574
[65,]	-0.0528038611	-0.0099977546	0.0178515785	0.0379954321	0.0295752497
[66,]	-0.0655685839	-0.0528038611	-0.0099977546	0.0178515785	0.0379954321
[67,]	-0.0361084433	-0.0655685839	-0.0528038611	-0.0099977546	0.0178515785
[68,]	-0.0083350789	-0.0361084433	-0.0655685839	-0.0528038611	-0.0099977546
[69,]	0.0372744742	-0.0083350789	-0.0361084433	-0.0655685839	-0.0528038611
[70,]	0.0492404647	0.0372744742	-0.0083350789	-0.0361084433	-0.0655685839
[71,]	0.0514080371	0.0492404647	0.0372744742	-0.0083350789	-0.0361084433
[72,]	0.0510942532	0.0514080371	0.0492404647	0.0372744742	-0.0083350789
[73,]	0.0665344115	0.0510942532	0.0514080371	0.0492404647	0.0372744742
[74,]	0.0511323253	0.0665344115	0.0510942532	0.0514080371	0.0492404647

[75,]	0.0463615981	0.0511323253	0.0665344115	0.0510942532	0.0514080371
[76,]	0.0336752941	0.0463615981	0.0511323253	0.0665344115	0.0510942532
[77,]	0.0291605087	0.0336752941	0.0463615981	0.0511323253	0.0665344115
[78,]	0.0175460213	0.0291605087	0.0336752941	0.0463615981	0.0511323253
[79,]	0.0154886280	0.0175460213	0.0291605087	0.0336752941	0.0463615981
[80,]	0.0142225002	0.0154886280	0.0175460213	0.0291605087	0.0336752941
[81,]	0.0056581603	0.0142225002	0.0154886280	0.0175460213	0.0291605087
[82,]	0.0305069664	0.0056581603	0.0142225002	0.0154886280	0.0175460213
[83,]	0.0308774823	0.0305069664	0.0056581603	0.0142225002	0.0154886280
[84,]	0.0276026912	0.0308774823	0.0305069664	0.0056581603	0.0142225002
[85,]	0.0490999652	0.0276026912	0.0308774823	0.0305069664	0.0056581603
[86,]	0.0346488227	0.0490999652	0.0276026912	0.0308774823	0.0305069664
[87,]	0.0358017884	0.0346488227	0.0490999652	0.0276026912	0.0308774823
[88,]	0.0424204059	0.0358017884	0.0346488227	0.0490999652	0.0276026912
[89,]	0.0282154475	0.0424204059	0.0358017884	0.0346488227	0.0490999652
[90,]	0.0337444820	0.0282154475	0.0424204059	0.0358017884	0.0346488227
[91,]	0.0331285814	0.0337444820	0.0282154475	0.0424204059	0.0358017884
[92,]	0.0373844847	0.0331285814	0.0337444820	0.0282154475	0.0424204059
[93,]	0.0343197078	0.0373844847	0.0331285814	0.0337444820	0.0282154475
[94,]	0.0487914477	0.0343197078	0.0373844847	0.0331285814	0.0337444820
[95,]	0.0299897045	0.0487914477	0.0343197078	0.0373844847	0.0331285814
[96,]	0.0282785948	0.0299897045	0.0487914477	0.0343197078	0.0373844847
[97,]	0.0459681771	0.0282785948	0.0299897045	0.0487914477	0.0343197078
[98,]	0.0279843861	0.0459681771	0.0282785948	0.0299897045	0.0487914477
[99,]	0.0433567397	0.0279843861	0.0459681771	0.0282785948	0.0299897045
[100,]	0.0479289263	0.0433567397	0.0279843861	0.0459681771	0.0282785948
[101,]	0.0304271605	0.0479289263	0.0433567397	0.0279843861	0.0459681771
[102,]	0.0395955660	0.0304271605	0.0479289263	0.0433567397	0.0279843861
[103,]	0.0219910435	0.0395955660	0.0304271605	0.0479289263	0.0433567397
[104,]	0.0268311490	0.0219910435	0.0395955660	0.0304271605	0.0479289263
[105,]	0.0330945264	0.0268311490	0.0219910435	0.0395955660	0.0304271605
[106,]	0.0228782682	0.0330945264	0.0268311490	0.0219910435	0.0395955660
[107,]	0.0418425360	0.0228782682	0.0330945264	0.0268311490	0.0219910435
[108,]	0.0292072118	0.0418425360	0.0228782682	0.0330945264	0.0268311490
[109,]	0.0152491384	0.0292072118	0.0418425360	0.0228782682	0.0330945264
[110,]	-0.0811063878	0.0152491384	0.0292072118	0.0418425360	0.0228782682
[111,]	-0.0171806194	-0.0811063878	0.0152491384	0.0292072118	0.0418425360
[112,]	-0.0023126329	-0.0171806194	-0.0811063878	0.0152491384	0.0292072118
[113,]	0.0003123391	-0.0023126329	-0.0171806194	-0.0811063878	0.0152491384
[114,]	0.1149645541	0.0003123391	-0.0023126329	-0.0171806194	-0.0811063878
[115,]	0.0668135553	0.1149645541	0.0003123391	-0.0023126329	-0.0171806194
[116,]	0.0631410541	0.0668135553	0.1149645541	0.0003123391	-0.0023126329
[117,]	0.0871829292	0.0631410541	0.0668135553	0.1149645541	0.0003123391

```
[118,] 0.0743265551 0.0871829292 0.0631410541 0.0668135553 0.1149645541
[119,] 0.0564924452 0.0743265551 0.0871829292 0.0631410541 0.0668135553
[120,] 0.0602844287 0.0564924452 0.0743265551 0.0871829292 0.0631410541
[121,] 0.0695948062 0.0602844287 0.0564924452 0.0743265551 0.0871829292
[122,] 0.0590362127 0.0695948062 0.0602844287 0.0564924452 0.0743265551
[123,] 0.0578294655 0.0590362127 0.0695948062 0.0602844287 0.0564924452
[124,] 0.0583002102 0.0578294655 0.0590362127 0.0695948062 0.0602844287
```

```
Y = embed(gdpgr,5)[,1]
X = embed(gdpgr,5)[,-1]
lm(Y~X)
```

```
Call:
lm(formula = Y ~ X)
```

```
Coefficients:
(Intercept)          X1          X2          X3          X4
    0.01377    0.61058    0.12867    0.15959   -0.37862
```

An alternative is the `dynlm()` function from the `dynlm` package (dynamic linear model). It has the option to use the lag operator  $L$

```
fitAR = dynlm(gdpgr ~ L(gdpgr) + L(gdpgr,2) + L(gdpgr,3) + L(gdpgr,4))
fitAR
```

```
Time series regression with "ts" data:
Start = 1993(1), End = 2023(4)
```

```
Call:
dynlm(formula = gdpgr ~ L(gdpgr) + L(gdpgr, 2) + L(gdpgr, 3) +
      L(gdpgr, 4))
```

```
Coefficients:
(Intercept)  L(gdpgr)  L(gdpgr, 2)  L(gdpgr, 3)  L(gdpgr, 4)
    0.01377    0.61058    0.12867    0.15959   -0.37862
```

You can also use `dynlm(gdpgr ~ L(gdpgr,1:4))`. The built-in function `ar.ols()` can be used as well, but it must be configured correctly:

```
ar.ols(gdpgr, aic=FALSE, order.max = 4, demean = FALSE, intercept = TRUE)
```

Let's predict the next value for the GDP growth,  $gdp_{T+1}$ . We use the regressors  $\mathbf{X}_{T+1} = (1, gdp_T, gdp_{T-1}, gdp_{T-2}, gdp_{T-3})'$ :

$$\widehat{gdp}_{T+1|T} = \mathbf{X}'_{T+1}\boldsymbol{\beta}.$$

```
## Define X_{T+1}
latestX = c(1, tail(gdpgr, 4))
## compute one-step ahead forecast
coef(fitAR) %*% latestX
```

```
      [,1]
[1,] 0.05101086
```

The above value is only a point forecast. Let's also compute 90% and 99% forecast intervals.

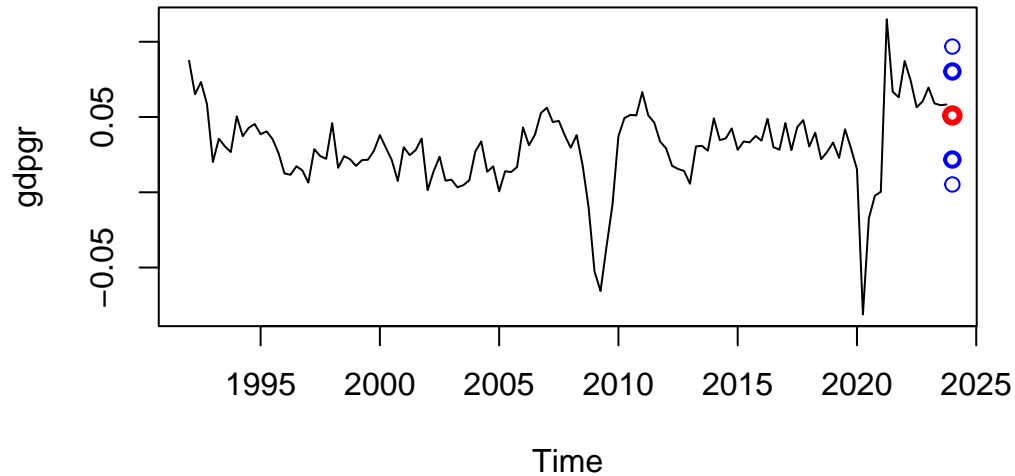
```
## One-step ahead point forecast
Yhat = coef(fitAR) %*% latestX
## standard error of regression
SER = summary(fitAR)$sigma
## Plot gdp growth
plot(gdpgr, main = "Forecast intervals for GDP growth")
## Plot point forecast
points(2024, Yhat, col="red", lwd = 3)
## Plot 90% forecast interval
points(2024, Yhat+SER*qnorm(0.95), col="blue", lwd=2)
points(2024, Yhat-SER*qnorm(0.95), col="blue", lwd=2)
## Plot 99% forecast interval
points(2024, Yhat+SER*qnorm(0.995), col="blue", lwd=1)
points(2024, Yhat-SER*qnorm(0.995), col="blue", lwd=1)
```

The forecast intervals are quite large, which is not too surprising given the simplicity of the model.

#### 14.4.2 An ADL model for gasoline prices

If  $X_t$  is a weekly price, then the return (the continuous growth rate) is  $\log(X_t) - \log(X_{t-1})$ , which is computed in R as `diff(log(X))`.

## Forecast intervals for GDP growth

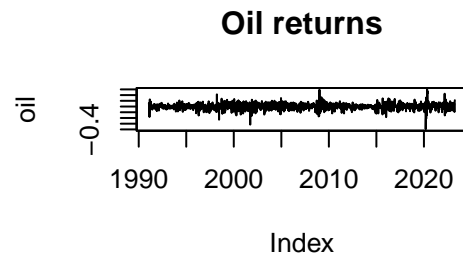
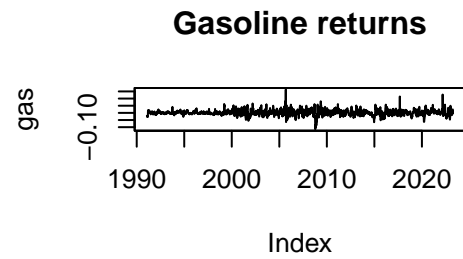
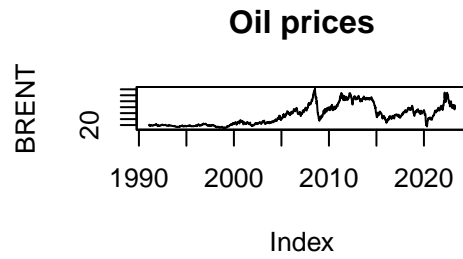
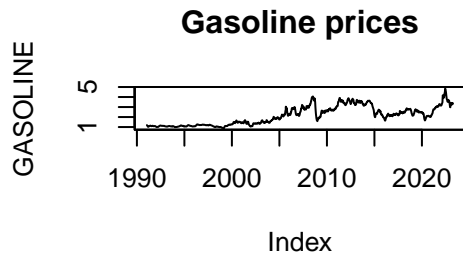


We consider an ADL(4,4) model regressing the weekly gasoline price returns on oil price returns:

$$\begin{aligned} gas_t = & \alpha_0 + \alpha_1 gas_{t-1} + \alpha_2 gas_{t-2} + \alpha_3 gas_{t-3} + \alpha_4 gas_{t-4} \\ & + \delta_1 oil_{t-1} + \delta_2 oil_{t-2} + \delta_3 oil_{t-3} + \delta_4 oil_{t-4} + u_t \end{aligned}$$

We can use the `zoo` class to assign time points to observations. The base R `ts` (time series) class can only handle time series with a fixed and regular number of observations per year such as yearly, quarterly, or monthly data. Weekly data do not have exactly the same number of observations per year, which is why we use the more flexible `zoo` class. `zoo` is part of the `AER` package. `zoo(mytimeseries, mydates)` defines a `zoo` object.

```
data(gasoil, package="teachingdata2")
GASOLINE = zoo(gasoil$gasoline, gasoil$date)
BRENT = zoo(gasoil$brent, gasoil$date)
gas = diff(log(GASOLINE))
oil = diff(log(BRENT))
par(mfrow = c(2,2))
plot(GASOLINE, main="Gasoline prices")
plot(BRENT, main="Oil prices")
plot(gas, main="Gasoline returns")
plot(oil, main="Oil returns")
```



```
fitADL = dynlm(gas ~ L(gas, 1:4) + L(oil, 1:4))
fitADL
```

Time series regression with "zoo" data:  
Start = 1991-02-25, End = 2023-04-03

Call:  
dynlm(formula = gas ~ L(gas, 1:4) + L(oil, 1:4))

Coefficients:

(Intercept)	L(gas, 1:4)1	L(gas, 1:4)2	L(gas, 1:4)3	L(gas, 1:4)4
0.0002527	0.3633626	0.0582818	0.0527356	-0.0143211
L(oil, 1:4)1	L(oil, 1:4)2	L(oil, 1:4)3	L(oil, 1:4)4	
0.1241477	0.0144996	0.0153132	0.0137106	

```
latestX = c(1, tail(gas,4), tail(oil,4))
## one-step ahead forecast
latestX %*% coef(fitADL)
```

```
[,1]
[1,] -0.002331957
```



## 14.5 Identification

Consider again the time series regression model of Equation 14.1. Under the regularity condition that the design matrix  $E[\mathbf{X}_t\mathbf{X}_t']$  is invertible (no multicollinearity), the coefficient vector  $\boldsymbol{\beta}$  can be written as

$$\boldsymbol{\beta} = (E[\mathbf{X}_t\mathbf{X}_t'])^{-1}E[\mathbf{X}_tY_t]. \quad (14.3)$$

In order for  $\boldsymbol{\beta}$  in Equation 14.3 to make sense, it must have same value for all time points  $t$ . That is,  $E[\mathbf{X}_t\mathbf{X}_t']$  and  $E[\mathbf{X}_tY_t]$  must be time invariant. To ensure this, we assume that the  $k+1$  vector  $\mathbf{Z}_t = (Y_t, \mathbf{X}_t)'$  is stationary.

Recall the definition of stationarity for a multivariate time series:

### Stationary univariate time series

A time series  $Y_t$  is called **stationary** if the **mean**  $\mu$  and the **autocovariance function**  $\gamma(\tau)$  do not depend on the time point  $t$ . That is,

$$\mu := E[Y_t] < \infty, \quad \text{for all } t,$$

and

$$\gamma(\tau) := Cov(Y_t, Y_{t-\tau}) < \infty \quad \text{for all } t \text{ and } \tau.$$

The **autocorrelation of order**  $\tau$  is

$$\rho(\tau) = \frac{Cov(Y_t, Y_{t-\tau})}{Var[Y_t]} = \frac{\gamma(\tau)}{\gamma(0)}, \quad \tau \in \mathbb{Z}.$$

The autocorrelations of stationary time series typically decay to zero quite quickly as  $\tau$  increases, i.e.,  $\rho(\tau) \rightarrow 0$  as  $\tau \rightarrow \infty$ . Observations close in time may be highly correlated, but observations farther apart have little dependence.

We define the stationarity concept for multivariate time series analogously:

### Stationary multivariate time series

A  $q$ -variate time series  $\mathbf{Z}_t = (Z_{1t}, \dots, Z_{qt})'$  is called **stationary** if each entry  $Z_{it}$  of  $\mathbf{Z}_t$  is a stationary time series, and, in addition, the **cross autocovariances** do not depend on  $t$ :

$$Cov(Z_{is}, Z_{j,s-\tau}) = Cov(Z_{it}, Z_{j,t-\tau}) < \infty$$

for all  $\tau \in \mathbb{Z}$  and for all  $s, t = 1, \dots, T$ , and  $i, j = 1, \dots, q$ .

The **mean vector** of  $\mathbf{Z}_t$  is

$$\boldsymbol{\mu} = (E[Z_{1t}], \dots, E[Z_{qt}])'$$

and the **autocovariance matrices** for  $\tau \geq 0$  are

$$\begin{aligned}\Gamma(\tau) &= E[(\mathbf{Z}_t - \boldsymbol{\mu})(\mathbf{Z}_{t-\tau} - \boldsymbol{\mu})'] \\ &= \begin{pmatrix} Cov(Z_{1,t}, Z_{1,t-\tau}) & \cdots & Cov(Z_{1,t}, Z_{q,t-\tau}) \\ \vdots & \ddots & \vdots \\ Cov(Z_{q,t}, Z_{1,t-\tau}) & \cdots & Cov(Z_{q,t}, Z_{q,t-\tau}) \end{pmatrix}\end{aligned}$$

A time series  $Y_t$  is **nonstationary** if the mean  $E[Y_t]$  or the autocovariances  $Cov(Y_t, Y_{t-\tau})$  change with  $t$ , i.e., if there exist time points  $s \neq t$  with

$$E[Y_t] \neq E[Y_s] \quad \text{or} \quad Cov(Y_t, Y_{t-\tau}) \neq Cov(Y_s, Y_{s-\tau})$$

for some  $\tau$ .

## 14.6 AR(1) process

To learn when a time series is stationary and when it is not, it is helpful to study the **autoregressive process of order one**, AR(1). It is defined as

$$Y_t = \phi Y_{t-1} + u_t, \tag{14.4}$$

where  $u_t$  is an i.i.d. sequence of increments with  $E[u_t] = 0$  and  $Var[u_t] = \sigma_u^2$ .

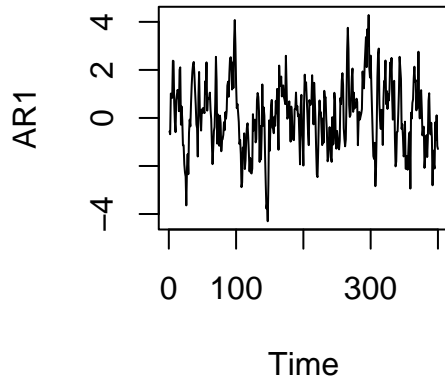
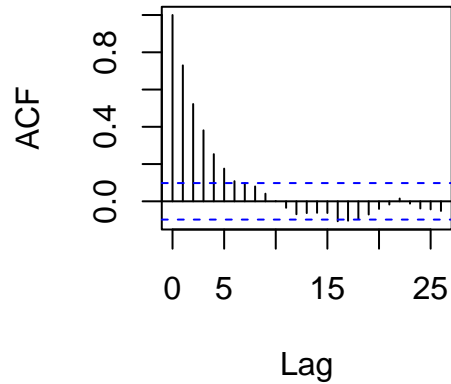
If  $|\phi| < 1$ , the AR(1) process is stationary with

$$\mu = 0, \quad \gamma(\tau) = \frac{\phi^\tau \sigma_u^2}{1 - \phi^2}, \quad \rho(\tau) = \phi^\tau, \quad \tau \geq 0.$$

Its autocorrelations  $\rho(\tau) = \phi^\tau$  decay exponentially in the lag order  $\tau$ .

Let's simulate a stationary AR(1) process. The function `filter(u, phi, "recursive")` computes Equation 14.4 for parameter `phi`, a given sequence `u` and starting value  $u_0 = 0$ .

```
## simulate AR1 with parameter phi=0.8,
## standard normal innovations, and T=400:
set.seed(123)
u = rnorm(400)
AR1 = stats::filter(u, 0.8, "recursive")
par(mfrow = c(1,2))
plot(AR1, main="Simulated AR(1) process")
acf(AR1)
```

**Simulated AR(1) process****Series AR1**

On the right hand side you find the values for the **sample autocorrelation function (ACF)**, which is defined as

$$\hat{\rho}(\tau) = \frac{\sum_{t=\tau+1}^T (Y_t - \bar{Y})(Y_{t-\tau} - \bar{Y})}{\sum_{t=1}^T (Y_t - \bar{Y})^2}.$$

The sample autocorrelations of the AR(1) process with parameter  $\phi = 0.8$  converge exponentially to 0 as  $\tau \rightarrow \infty$ .

The **simple random walk** is an example of a nonstationary time series process. It is an AR(1) process with  $\phi = 1$  and starting value  $Y_0 = 0$ , i.e.,

$$Y_t = Y_{t-1} + u_t, \quad t \geq 1.$$

By backward substitution, it can be expressed as the cumulative sum

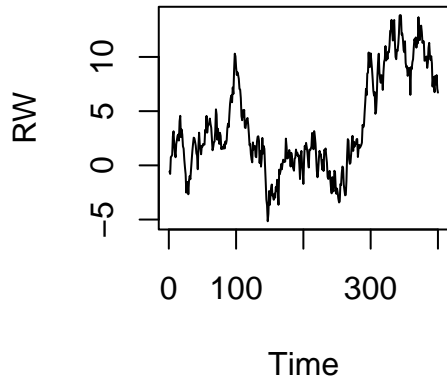
$$Y_t = \sum_{j=1}^t u_j.$$

It is nonstationary since  $Cov(Y_t, Y_{t-\tau}) = (t-\tau)\sigma_u^2$ , which depends on  $t$  and becomes larger as  $t$  gets larger.

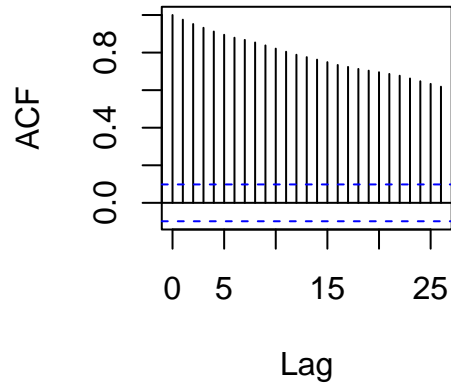
```
## simulate AR1 with parameter phi=1 (random walk):
RW = stats::filter(u, 1, "recursive")
par(mfrow = c(1,2))
plot(RW, main= "Simulated random walk")
acf(RW)
```

The ACF plots indicate the dynamic structure of the time series and whether they can be regarded as a stationary time series. The ACF of AR1 tends to zero quickly. It can be treated

### Simulated random walk



### Series RW



as stationary time series. The ACF of RW tends to zero very slowly, indicating a high persistence. This time series is non-stationary.

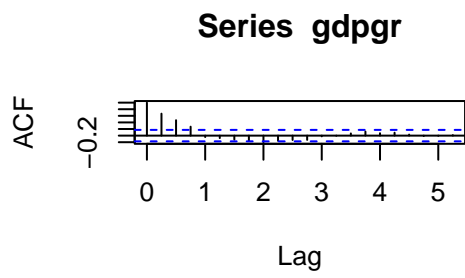
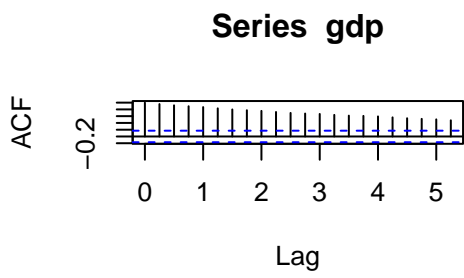
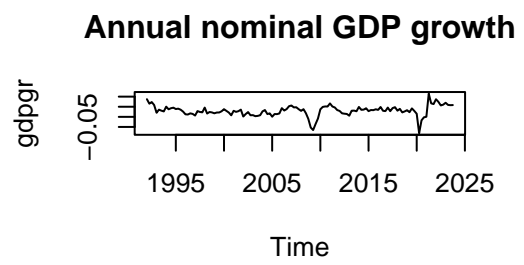
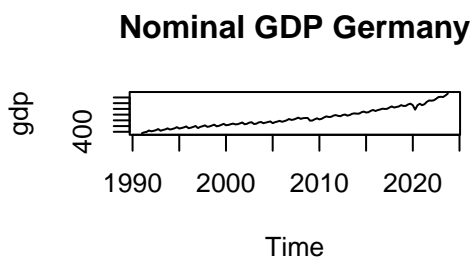
## 14.7 Autocorrelations of GDP

```
data(gdp, package="teachingdata")
par(mfrow = c(2,2))
plot(gdp, main="Nominal GDP Germany")
plot(gdpgr, main = "Annual nominal GDP growth")
acf(gdp)
acf(gdpgr)
```

The ACF plots indicate that nominal GDP is nonstationary, while GDP growth is stationary. The asymptotic normality result for OLS is not valid if nonstationary time series are used.

## 14.8 R-codes

[methods-sec14.R](#)



# 15 Time Series Inference

```
library(AER) # for sandwich, lmtest, and zoo
library(dynlm) # for dynamic regression
library(BVAR) # for the fred_qd data
```

In the previous section, we considered time series regression models tailored for forecasting, where the regressors are based on past data relative to the dependent variable.

Of course, the regressors may also be contemporaneous as in the **static time series regression**

$$Y_t = \alpha + \delta Z_t + u_t.$$

The ADL model can also be extended by a contemporaneous exogenous variable:

$$Y_t = \alpha_0 + \alpha_1 Y_{t-1} + \dots + \alpha_p Y_{t-p} + \delta_0 Z_t + \delta_1 Z_{t-1} + \dots + \delta_q Z_{t-q} + u_t.$$

Time series regressions have the general form

$$Y_t = \mathbf{X}'_t \boldsymbol{\beta} + u_t, \quad t = 1, \dots, T. \quad (15.1)$$

## 15.1 Assumptions for time series regression

Compared to cross-sectional regression, time series regressions require a stationarity condition instead of the i.i.d. assumption. Moreover, the error must be conditional mean independent of all past values, which indicates that the error represents the new information (shock) that was not available before time  $t$ . Variables that are conditional mean independent of the past are also called **martingale difference sequence**.

For the dynamic linear regression Equation 15.1 we make the following assumptions:

- (A1-dyn) **martingale difference sequence**:  $E[u_t | \mathbf{X}_t, \mathbf{X}_{t-1}, \dots] = 0$ .
- (A2-dyn) **stationary processes**:  $\mathbf{Z}_t = (Y_t, \mathbf{X}'_t)'$  is a stationary time series with the property that  $\mathbf{Z}_t$  and  $\mathbf{Z}_{t-\tau}$  become independent as  $\tau$  gets large.
- (A3-dyn) **large outliers unlikely**:  $0 < E[Y_t^4] < \infty$ ,  $0 < E[X_{tl}^4] < \infty$  for all  $l = 1, \dots, k$ .

- (A4-dyn) **no perfect multicollinearity**:  $\mathbf{X}$  has full column rank.

The precise mathematical statement for “becoming independent as  $\tau$  gets large” is omitted here. It can be formulated with respect to a so-called strong mixing condition. It essentially requires that the dependency between  $\mathbf{Z}_t$  and  $\mathbf{Z}_{t-\tau}$  decrease as  $\tau \rightarrow \infty$  with a certain rate so that  $\mathbf{Z}_t$  and  $\mathbf{Z}_{t-\tau}$  are “almost independent” if  $\tau$  is large enough.

Under (A1-dyn)–(A4-dyn), the OLS estimator  $\hat{\boldsymbol{\beta}}$  is consistent for  $\boldsymbol{\beta}$  and asymptotically normal.

## 15.2 Time series standard errors

We have

$$\frac{\hat{\beta}_l - \beta_l}{sd(\hat{\beta}_l|\mathbf{X})} \xrightarrow{D} \mathcal{N}(0, 1) \quad \text{as } T \rightarrow \infty.$$

The standard deviation  $sd(\hat{\beta}_l|\mathbf{X})$  is the squareroot of the  $(l, l)$ -entry of

$$Var[\hat{\boldsymbol{\beta}}|\mathbf{X}] = (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{D}\mathbf{X})(\mathbf{X}'\mathbf{X})^{-1},$$

where  $\mathbf{D} = Var[\mathbf{u}|\mathbf{X}]$ .

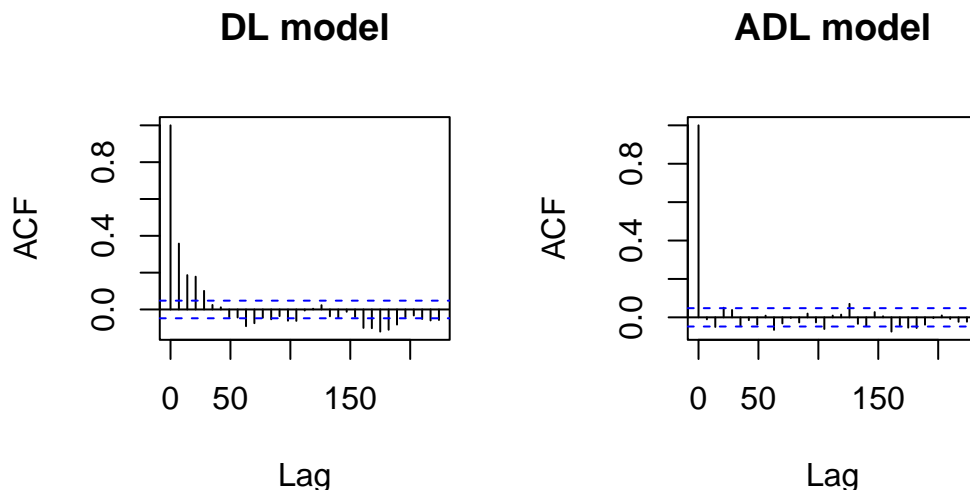
If the errors are uncorrelated, i.e.  $Cov(u_t, u_{t-\tau}) = 0$  for  $\tau \geq 1$ , the matrix  $\mathbf{D}$  is diagonal as in Section 5, and heteroskedasticity-consistent standard errors can be used. If the errors exhibit autocorrelation, then  $\mathbf{D}$  has an arbitrary form with off diagonal entries decaying slowly to zero as the distance to the main diagonal increases. In this case, heteroskedasticity and autocorrelation-consistent (HAC) standard errors must be used.

You can check potential autocorrelation in the errors by consulting the ACF plot for the residuals:

```
data(gasoil, package="teachingdata2")
gas = zoo(diff(log(gasoil$gasoline)), gasoil$date)
oil = zoo(diff(log(gasoil$brent)), gasoil$date)
DL = dynlm(gas ~ L(oil, 1:2))
ADL = dynlm(gas ~ L(gas, 1:2) + L(oil, 1:2))
par(mfrow=c(1,2))
acf(DL$residuals, main="DL model")
acf(ADL$residuals, main = "ADL model")
```

The residuals in the DL(2) model

$$gas_t = \alpha + \delta_1 oil_{t-1} + \delta_2 oil_{t-2} + u_t$$



indicate significant autocorrelation in the first few lags. The sample autocorrelations are above the blue dashed threshold.

The blue threshold indicates the critical value  $1.96/\sqrt{T}$  for a test for the null hypothesis  $H_0 : \rho(\tau) = 0$ .

We should use HAC standard errors:

```
coeftest(ADL, vcov. = vcovHC)
```

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	0.00022028	0.00038647	0.5700	0.56877	
L(gas, 1:2)1	0.37403773	0.06264798	5.9705	2.882e-09	***
L(gas, 1:2)2	0.11072881	0.04516219	2.4518	0.01432	*
L(oil, 1:2)1	0.12355493	0.01020577	12.1064	< 2.2e-16	***
L(oil, 1:2)2	0.00754501	0.01121716	0.6726	0.50127	

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The residuals in the ADL(2,2) model

$$gas_t = \alpha_0 + \alpha_1 gas_{t-1} + \alpha_2 gas_{t-2} + \delta_1 oil_{t-1} + \delta_2 oil_{t-2} + u_t$$

indicate no autocorrelation in the error term. We can use HC standard errors:



```
coefstest(DL, vcov. = vcovHAC)
```

t test of coefficients:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.00042119 0.00059160  0.7120   0.4766
L(oil, 1:2)1  0.17029082 0.01068906 15.9313 < 2.2e-16 ***
L(oil, 1:2)2  0.08437856 0.01128826  7.4749 1.239e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The following section highlights the importance of the variables being stationary in a time series regression.

## 15.3 Spurious correlation

Spurious correlation occurs when two unrelated time series  $Y_t$  and  $X_t$  have zero population correlation ( $Cov(Y_t, X_t) = 0$ ) but exhibit a large sample correlation coefficient due to coincidental patterns or trends within the sample data.

Here are some examples of nonsense correlations: [tylervigen.com/spurious-correlations](http://tylervigen.com/spurious-correlations).

Nonsense correlations may occur if the underlying time series process is nonstationary.

### 15.3.1 Simulation evidence

Let's simulate two independent AR(1) processes:

$$Y_t = \alpha Y_{t-1} + u_t, \quad X_t = \alpha X_{t-1} + v_t,$$

for  $t = 1, \dots, 200$ , where  $u_t$  and  $v_t$  are i.i.d. standard normal. If  $\alpha = 0.5$ , the processes are stationary. If  $\alpha = 1$ , the processes are nonstationary (random walk).

In any case, the population covariance is zero:

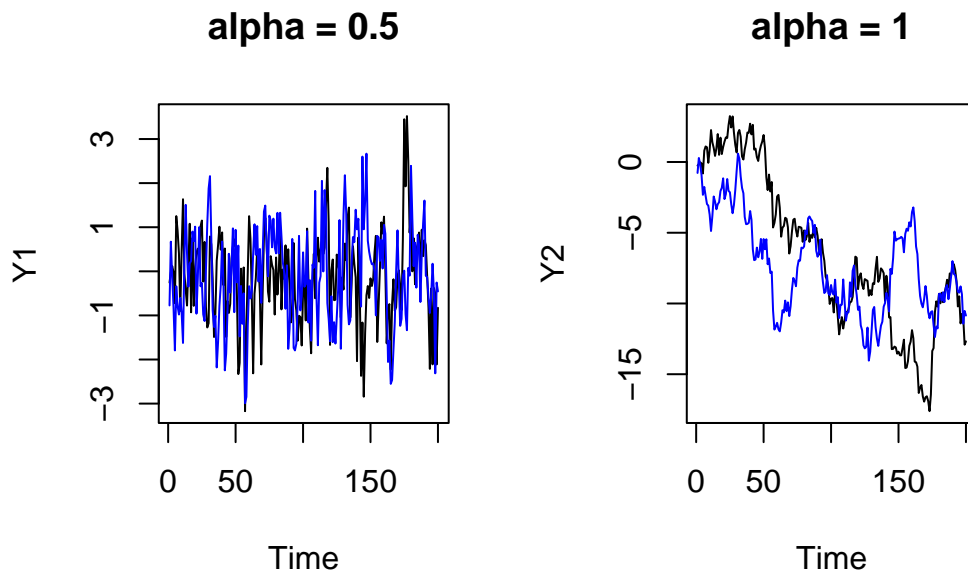
$$Cov(Y_t, X_t) = 0.$$

Therefore, we expect that the sample correlation is zero as well:

```

set.seed(121)
## Plot two independent AR(1) processes
u = rnorm(200)
v = rnorm(200)
Y1 = stats::filter(u, 0.5, "recursive")
X1 = stats::filter(v, 0.5, "recursive")
par(mfrow = c(1,2))
plot(Y1, main = "alpha = 0.5")
lines(X1, col="blue")
Y2 = stats::filter(u, 1, "recursive")
X2 = stats::filter(v, 1, "recursive")
plot(Y2, main = "alpha = 1")
lines(X2, col="blue")

```



```

## Squared sample correlation for alpha = 0.5:
cor(Y1,X1)^2

```

```
[1] 0.0214327
```

```

## Squared sample correlation for alpha = 1:
cor(Y2,X2)^2

```

```
[1] 0.325291
```

The squared sample correlation is equal to the R-squared of a simple regression of  $Y_t$  on  $X_t$ . The R-squared for the two independent stationary time series is close to zero, and the R-squared for the two independent nonstationary time series is unreasonably large.

The correlation of the differenced series is close to zero:

```
cor(diff(Y2), diff(X2))^2
```

```
[1] 0.02193921
```

Of course, a large sample correlation of two uncorrelated series could occur by chance. Let's repeat the simulation 10 times. Still, in many cases, the R-squared for the nonstationary series is much higher than expected:

```
## Simulate two independent AR(1) processes and R-squared
```

```
R2 = function(alpha, n=200){
```

```
  u = rnorm(n)
```

```
  v = rnorm(n)
```

```
  Y = stats::filter(u, alpha, "recursive")
```

```
  X = stats::filter(v, alpha, "recursive")
```

```
  return(cor(Y,X)^2)
```

```
}
```

```
## Get R-squared results with alpha = 0.5
```

```
c(R2(0.5, 200), R2(0.5, 200), R2(0.5, 200), R2(0.5, 200), R2(0.5, 200), R2(0.5, 200), R2(0.5, 200), R2(0.5, 200), R2(0.5, 200), R2(0.5, 200))
```

```
[1] 0.0052 0.0038 0.0014 0.0081 0.0020 0.0044 0.0096 0.0187 0.0102 0.0190
```

```
## Get R-squared results with alpha = 1
```

```
c(R2(1, 200), R2(1, 200), R2(1, 200), R2(1, 200), R2(1, 200), R2(1, 200), R2(1, 200), R2(1, 200), R2(1, 200), R2(1, 200))
```

```
[1] 0.0001 0.4746 0.3424 0.1782 0.4056 0.0385 0.1625 0.2406 0.3836 0.3570
```

Increasing the sample size to  $T = 1000$  gives a similar picture:

```
c(R2(1, 1000), R2(1, 1000), R2(1, 1000), R2(1, 1000), R2(1, 1000), R2(1, 1000), R2(1, 1000), R2(1, 1000), R2(1, 1000), R2(1, 1000))
```

```
[1] 0.2365 0.0019 0.0215 0.4754 0.0425 0.2173 0.0030 0.4104 0.6846 0.3555
```

The reason is that the OLS estimator is inconsistent if two independent random walks are regressed on each other. The key problem is that already simple moment statistics such as the sample mean or sample correlation are inconsistent for random walks. The behavior of the sample mean or OLS coefficients is driven by the stochastic path of the random walk.

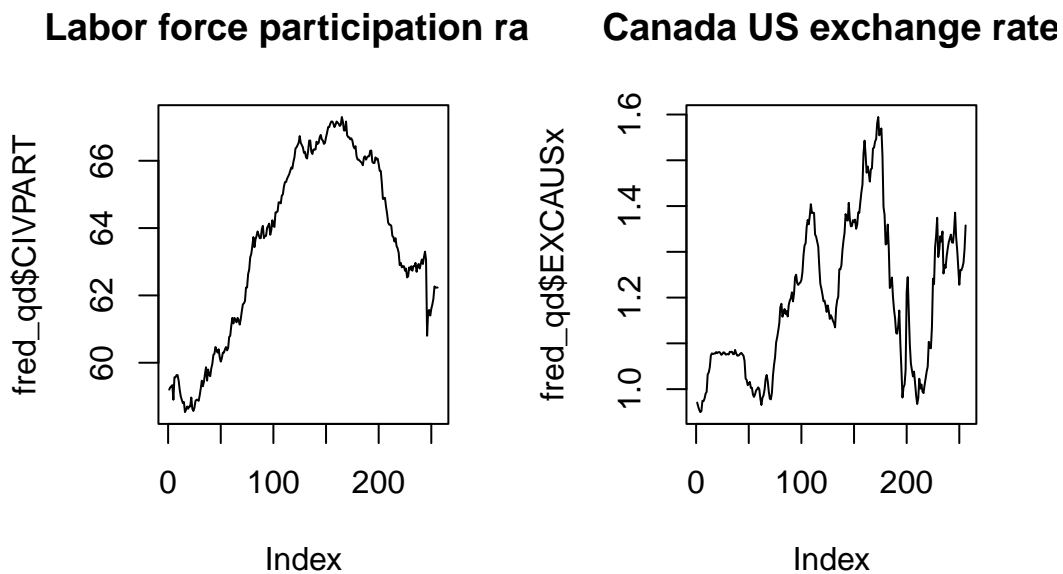
Two completely unrelated random walks might share common upward and downward drifts by chance, which can produce high sample correlations although the population correlation is zero.

### 15.3.2 Real-world spurious correlations

The [FRED-QD database](#) offers a comprehensive collection of quarterly U.S. macroeconomic time series data. A subset of this data is contained in the package `BVAR`. See the [appendix of this paper](#) for a detailed description of the data.

We expect no relationship between the labor force participation rate and the Canada US exchange rate. However, the sample correlation coefficient is extremely high:

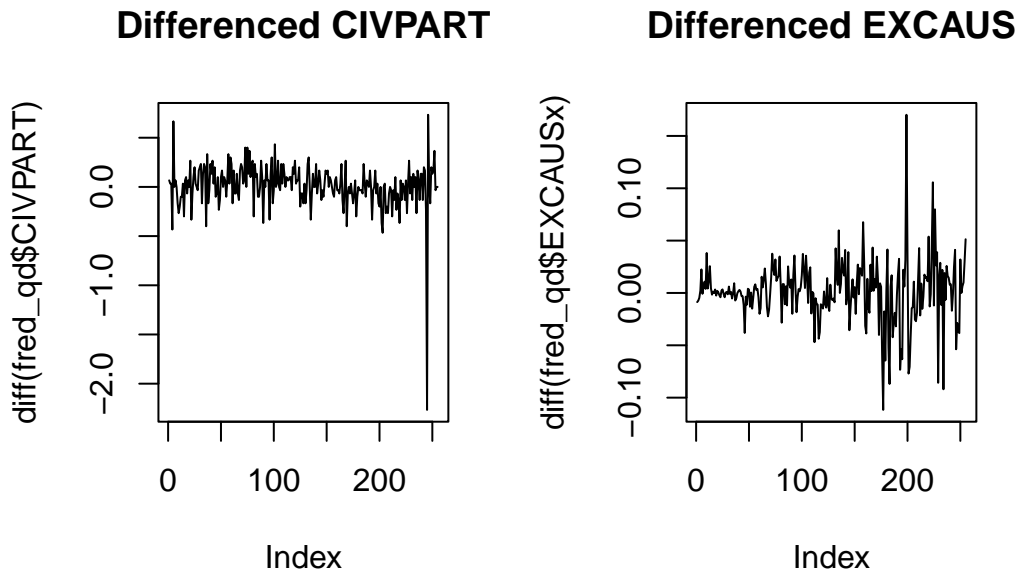
```
data(fred_qd, package = "BVAR")
par(mfrow=c(1,2))
plot(fred_qd$CIVPART, main="Labor force participation rate", type = "l")
plot(fred_qd$EXCAUSx, main="Canada US exchange rate", type = "l")
```



```
cor(fred_qd$CIVPART, fred_qd$EXCAUSx)
```

```
[1] 0.648879
```

```
plot(diff(fred_qd$CIVPART), main="Differenced CIVPART", type = "l")  
plot(diff(fred_qd$EXCAUSx), main="Differenced EXCAUS", type = "l")
```



```
cor(diff(fred_qd$CIVPART), diff(fred_qd$EXCAUSx))
```

```
[1] -0.03030723
```

The sample correlation of the differenced series indicates no relationship.

## 15.4 Testing for stationarity

The ACF plot provides a useful tool to decide whether a time series exhibits stationary or nonstationary behavior. We can also run a hypothesis test for the hypothesis that a series is nonstationary against the alternative that it is stationary.

### 15.4.1 Dickey Fuller test

Consider the AR(1) plus constant model:

$$Y_t = c + \phi Y_{t-1} + u_t, \quad (15.2)$$

where  $u_t$  is an i.i.d. zero mean sequence.

$Y_t$  is stationary if  $|\phi| < 1$  and nonstationary if  $\phi = 1$  (the cases  $\phi > 1$  and  $\phi \leq -1$  lead to exponential or oscillating behavior and are ignored here).

Let's consider the hypotheses

$$\underbrace{H_0 : \phi = 1}_{\text{nonstationarity}} \quad \text{vs.} \quad \underbrace{H_1 : |\phi| < 1}_{\text{stationarity}}$$

To test  $H_0$ , we can run a t-test for  $\phi = 1$ . The t-statistic is

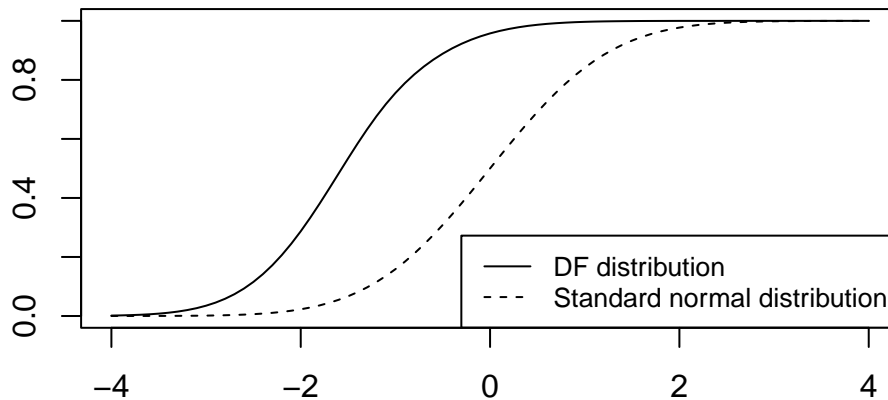
$$Z_{\hat{\phi}} = \frac{\hat{\phi} - 1}{se(\hat{\phi})},$$

where  $\hat{\phi}$  is the OLS estimator and  $se(\hat{\phi})$  is the homoskedasticity-only standard error.

Unfortunately, under  $H_0$  the time series regression assumptions are not satisfied because  $Y_t$  is a random walk. The OLS estimator is not normally distributed, but is consistent. It can be shown that the t-statistic does not converge to a standard normal distribution. Instead, it converges to the Dickey-Fuller distribution:

$$Z_{\hat{\phi}} \xrightarrow{D} DF$$

### Cumulative distribution functions



The critical values are much smaller:

	0.01	0.025	0.05	0.1
$\mathcal{N}(0, 1)$	-2.32	-1.96	-1.64	-1.28
$DF$	-3.43	-3.12	-2.86	-2.57

More quantiles for the DF distribution can be obtained from the function `qunitroot()` from the `urca` package.

We reject  $H_0$  if the t-statistic  $Z_{\hat{\phi}}$  is smaller than the corresponding critical value from the above table.

## 15.4.2 Augmented Dickey Fuller test

The assumption that  $\Delta Y_t = Y_t - Y_{t-1} = u_t$  in Equation 15.2 is i.i.d. is unreasonable in many cases. It is more realistic that

$$Y_t = c + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + u_t$$

for some lag order  $p$ . In this model,  $Y_t$  is nonstationary if  $\sum_{j=1}^p \phi_j = 1$ .

The equation can be rewritten as

$$\Delta Y_t = c + \psi Y_{t-1} + \theta_1 \Delta Y_{t-1} + \dots + \theta_{p-1} \Delta Y_{t-(p-1)} + u_t, \quad (15.3)$$

where  $\psi = \sum_{j=1}^p \phi_j - 1$  and  $\theta_i = -\sum_{j=i+1}^p \phi_j$ .

To test for nonstationarity, we formulate the null hypothesis  $H_0 : \sum_{j=1}^p \phi_j = 1$ , which is equivalent to  $H_0 : \psi = 0$ .

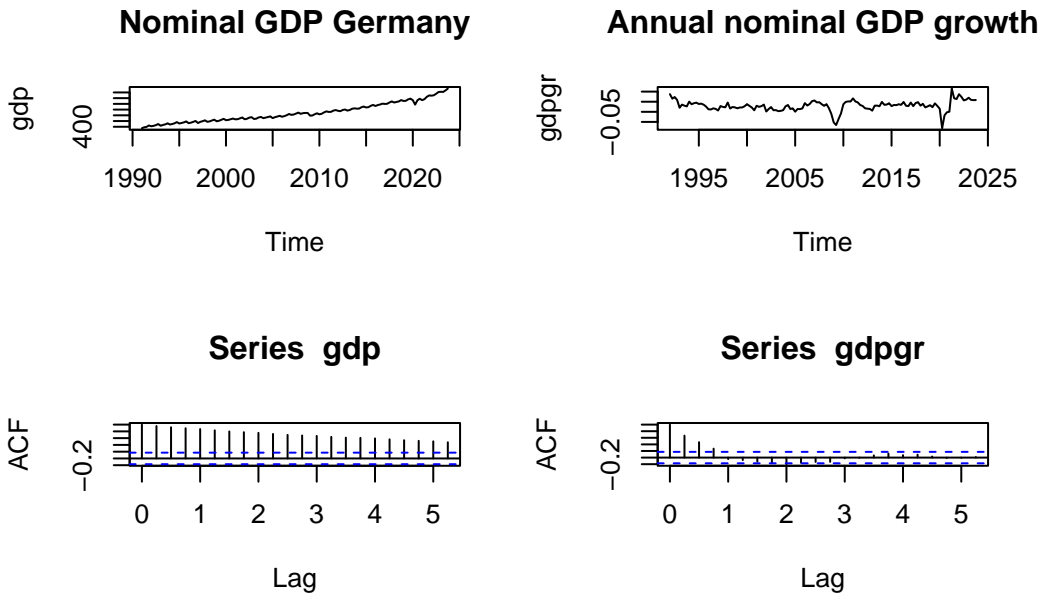
The t-statistic  $Z_{\hat{\psi}}$  from Equation 15.3 converges under  $H_0$  to the DF distribution as well. Therefore, we can reject the null hypothesis of nonstationarity, if  $Z_{\hat{\psi}}$  is smaller than the corresponding quantile from the DF distribution.

This test is called **Augmented Dickey-Fuller test (ADF)**.

```
data(gdp, package="teachingdata")
data(gdpgr, package="teachingdata")
par(mfrow = c(2,2))
plot(gdp, main="Nominal GDP Germany")
plot(gdpgr, main = "Annual nominal GDP growth")
acf(gdp)
acf(gdpgr)
```

We use the `ur.df()` function from the `urca` package with the option `type = "drift"` to compute the ADF test statistic.

```
library(urca)
ur.df(gdp, type = "drift", lags = 4)
```



```
#####
# Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
#####
```

The value of the test statistic is: 2.4235 7.8698

The ADF statistic  $Z_{\hat{\rho}}$  is the first value from the output. The critical value for  $\alpha = 0.05$  is -2.86. Hence, the ADF with  $p = 4$  does not reject the null hypothesis that GDP is nonstationary.

```
ur.df(gdpgr, type = "drift", lags = 4)
```

```
#####
# Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
#####
```

The value of the test statistic is: -4.1546 8.6402

The ADF statistic with  $p = 4$  is below -2.86, and the ADF test rejects the null hypothesis that GDP growth is nonstationary at the 5% significance level.

These results are in line with the observations from the ACF plots.



## 15.5 R-codes

[methods-sec15.R](#)

# A OLS: Technical Details

This section provides technical details about the linear model and the OLS estimator.

## A.1 Probability toolbox

We will use some results from probability theory. Let  $V$  and  $W$  be random variables vectors (or random vectors with compatible dimensions).

### 1.) Law of the iterated expectation (LIE):

$$E[V] = E[E[V|W]].$$

See Stock and Watson Section 2.3.

### 2.) Conditioning theorem (CT):

$$E[WW|W] = WE[V|W]$$

Moreover,  $E[g(W)V|W] = g(W)E[V|W]$  for any function  $g(\cdot)$ .

### 3.) Independence rule (IR):

If  $V$  and  $W$  are independent, then  $E[V] = E[V|W]$ . Moreover, if  $V$  and  $W_2$  are independent, then  $E[V|W_1] = E[V|W_1, W_2]$ .

### 4.) Functions of independent random variables:

If  $V$  and  $W$  are independent and  $g(\cdot)$  and  $h(\cdot)$  are functions, then  $g(V)$  and  $h(W)$  are independent.

### 5.) Cauchy-Schwarz inequality:

$$|E[VW]| \leq \sqrt{E[V^2]}\sqrt{E[W^2]}$$

See Stock and Watson Appendix 18.2.

### 6.) Convergence in probability

The sequence  $W_n$  convergence in probability to the constant  $C$ , written  $W_n \xrightarrow{p} C$ , if, for and  $\delta > 0$ ,

$$P(|W_n - C| > \delta) \rightarrow 0 \quad (\text{as } n \rightarrow \infty).$$

If  $W_n$  is a random vector or matrix, and  $C$  is a deterministic vector or matrix, then  $W_n \xrightarrow{p} C$  if each entry of  $W_n - C$  converges in probability to zero.  $W_n$  is called consistent for  $C$  if  $W_n \xrightarrow{p} C$ . A sufficient conditions for consistency is that both  $E[W_n] \rightarrow C$  and  $Var[W_n] \rightarrow 0$  as  $n \rightarrow \infty$ . See Stock and Watson Section 2.6 and 18.2.

### 7.) Law of large numbers

If  $W_n$  is an i.i.d. sequence with  $E[W_n^2] < \infty$  (or bounded second moment in each entry for vectors/matrices), then

$$\frac{1}{n} \sum_{i=1}^n W_n \xrightarrow{p} E[W_n]$$

See Stock and Watson Section 2.6 and 18.2.

### 8.) Convergence in distribution:

Let  $F_n$  be the cumulative distribution function (CDF) of  $W_n$  and let  $G$  be the CDF of  $V$ .  $W_n$  converges in distribution to  $V$ , written  $W_n \xrightarrow{d} V$ , if  $F_n(a) \rightarrow G(a)$  for all  $a$  at which  $G$  is continuous. If  $V$  is  $\mathcal{N}(\mu, \Sigma)$  distributed, we also write  $W_n \xrightarrow{d} \mathcal{N}(\mu, \Sigma)$ .

### 9.) Multivariate central limit theorem:

If  $W_n$  is an i.i.d. sequence of vectors with bounded second moments in each entry, then

$$\sqrt{n} \left( \frac{1}{n} \sum_{i=1}^n W_n - E[W_n] \right) \xrightarrow{d} \mathcal{N}(\mathbf{0}, Var[W_n]).$$

See Stock and Watson Section 19.2.

## 10. Continuous Mapping Theorem

Let  $g(\cdot)$  be a continuous function. If  $W_n \xrightarrow{p} C$ , then  $g(W_n) \xrightarrow{p} g(C)$ . Also, if  $W_n \xrightarrow{d} C$ , then  $g(W_n) \xrightarrow{d} g(C)$ . See Stock and Watson Section 18.2.

## 11. Slutsky's Theorem

If  $V_n \xrightarrow{p} C$  and  $W_n \xrightarrow{p} D$ , then  $V_n W_n \xrightarrow{p} CD$ . If  $V_n \xrightarrow{p} C$  and  $W_n \xrightarrow{d} W$ , then  $V_n W_n \xrightarrow{d} CW$ . See Stock and Watson Section 18.2.

## A.2 Conditional Expectation

Inserting the model equation  $Y_i = \mathbf{X}'_i \boldsymbol{\beta} + u_i$  gives

$$E[Y_i | \mathbf{X}_i] = E[\mathbf{X}'_i \boldsymbol{\beta} + u_i | \mathbf{X}_i] = \underbrace{E[\mathbf{X}'_i \boldsymbol{\beta} | \mathbf{X}_i]}_{\stackrel{(CT)}{=} \mathbf{X}'_i \boldsymbol{\beta}} + \underbrace{E[u_i | \mathbf{X}_i]}_{\stackrel{(A1)}{=} 0}$$

## A.3 Weak exogeneity

(A1) and the law of iterated expectations (LIE) imply

$$E[u_i] \stackrel{(LIE)}{=} E[\underbrace{E[u_i | \mathbf{X}_i]}_{=0}] = E[0] = 0,$$

and the conditioning theorem (CT) yields

$$\begin{aligned} Cov(u_i, X_{il}) &= E[u_i X_{il}] - \underbrace{E[u_i]}_{=0} E[X_{il}] \\ &\stackrel{(LIE)}{=} E[E[u_i X_{il} | \mathbf{X}_i]] \stackrel{(CT)}{=} E[\underbrace{E[u_i | \mathbf{X}_i]}_{=0} X_{il}] = 0. \end{aligned}$$

## A.4 Strict exogeneity

The i.i.d. assumption (A2) implies that  $\{(Y_i, \mathbf{X}'_i, u_i), i = 1, \dots, n\}$  is an i.i.d. collection since  $u_i = Y_i - \mathbf{X}'_i \boldsymbol{\beta}$  is a function of a random sample, and functions of independent variables

are independent as well. Therefore,  $u_i$  and  $\mathbf{X}_j$  are independent for  $i \neq j$ , and (IR) implies  $E[u_i|\mathbf{X}_1, \dots, \mathbf{X}_n] = E[u_i|\mathbf{X}_i]$ . Then,

$$E[u_i|\mathbf{X}] = E[u_i|\mathbf{X}_1, \dots, \mathbf{X}_n] \stackrel{(A2)}{=} E[u_i|\mathbf{X}_i] \stackrel{(A1)}{=} 0.$$

and

$$Cov(u_i, X_{jl}) = \underbrace{E[u_i X_{jl}]}_{=0} - \underbrace{E[u_i]}_{=0} E[X_{jl}] = 0.$$

## A.5 Heteroskedasticity

$$Var[u_i|\mathbf{X}] = E[u_i^2|\mathbf{X}] \stackrel{(A2)}{=} E[u_i^2|\mathbf{X}_i] =: \sigma_i^2 = \sigma^2(\mathbf{X}_i).$$

## A.6 No autocorrelation

(A2) implies that  $u_i$  is independent of  $u_j$  for  $i \neq j$ , and  $E[u_i|u_j, \mathbf{X}] = E[u_i|\mathbf{X}] = 0$ , which implies

$$E[u_i u_j|\mathbf{X}] \stackrel{(LIE)}{=} E[E[u_i u_j|u_j, \mathbf{X}]|\mathbf{X}] \stackrel{(CT)}{=} E[u_j \underbrace{E[u_i|u_j, \mathbf{X}]|\mathbf{X}}_{=0}] = 0,$$

and

$$Cov(u_i, u_j) = E[u_i u_j] \stackrel{(LIE)}{=} E[E[u_i u_j|\mathbf{X}]] = 0.$$

The conditional covariance matrix is

$$\mathbf{D} := Var[\mathbf{u}|\mathbf{X}] = E[\mathbf{u}\mathbf{u}'|\mathbf{X}] = \begin{pmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n^2 \end{pmatrix}.$$

## A.7 Existence

$$rank(\mathbf{X}) = k \quad \Leftrightarrow \quad rank(\mathbf{X}'\mathbf{X}) = k \quad \Leftrightarrow \quad \mathbf{X}'\mathbf{X} \text{ is invertible.}$$

## A.8 Unbiasedness

(A4) ensures that  $\hat{\beta}$  is well defined. The following decomposition is useful:

$$\begin{aligned}\hat{\beta} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \\ &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{X}\beta + \mathbf{u}) \\ &= (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{X})\beta + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u} \\ &= \beta + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}.\end{aligned}$$

The strict exogeneity implies  $E[\mathbf{u}|\mathbf{X}] = \mathbf{0}$ , and

$$E[\hat{\beta} - \beta|\mathbf{X}] = E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}|\mathbf{X}] \stackrel{(CT)}{=} (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \underbrace{E[\mathbf{u}|\mathbf{X}]}_{=\mathbf{0}} = \mathbf{0}.$$

By the (LIE),  $E[\hat{\beta}] = E[E[\hat{\beta}|\mathbf{X}]] = E[\beta] = \beta$ .

## A.9 Conditional variance

Recall the matrix rule  $Var[\mathbf{A}\mathbf{z}] = \mathbf{A}Var[\mathbf{z}]\mathbf{A}'$  if  $\mathbf{z}$  is a random vector and  $\mathbf{A}$  is a matrix. Then,

$$\begin{aligned}Var[\hat{\beta}|\mathbf{X}] &= Var[\beta + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}|\mathbf{X}] \\ &= Var[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}|\mathbf{X}] \\ &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'Var[\mathbf{u}|\mathbf{X}](\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \\ &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{D}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}.\end{aligned}$$

(A5) implies  $\mathbf{D} = \sigma^2\mathbf{I}_n$  and

$$Var[\hat{\beta}|\mathbf{X}] = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}.$$

## A.10 Consistency

Let  $\mathbf{Q} := E[\mathbf{X}_i\mathbf{X}_i']$  and

$$\mathbf{\Omega} := E[(\mathbf{X}_i u_i)(\mathbf{X}_i u_i)'] = E[E[u_i^2|\mathbf{X}_i]\mathbf{X}_i\mathbf{X}_i'] = E[\sigma_i^2\mathbf{X}_i\mathbf{X}_i'].$$

By (A3) and the Cauchy-Schwarz inequality, the entries of  $\mathbf{X}_i\mathbf{X}_i'$  and  $(\mathbf{X}_i u_i)(\mathbf{X}_i u_i)'$  have bounded second moments, and by (A2) these entries form i.i.d. sequences. Hence, the conditions for the Law of Large Numbers are satisfied, and we have

$$\begin{aligned}\frac{1}{n}\mathbf{X}'\mathbf{X} &= \frac{1}{n}\sum_{i=1}^n \mathbf{X}_i\mathbf{X}_i' \xrightarrow{p} \mathbf{Q}, \\ \frac{1}{n}\mathbf{X}'\mathbf{D}\mathbf{X} &= \frac{1}{n}\sum_{i=1}^n \sigma_i^2\mathbf{X}_i\mathbf{X}_i' \xrightarrow{p} \mathbf{\Omega}.\end{aligned}$$

Consequently,

$$Var[\hat{\boldsymbol{\beta}}|\mathbf{X}] = \underbrace{\frac{1}{n}}_{\rightarrow 0} \underbrace{\left(\frac{1}{n}\mathbf{X}'\mathbf{X}\right)^{-1}}_{\xrightarrow{p}\mathbf{Q}} \underbrace{\left(\frac{1}{n}\mathbf{X}'\mathbf{DX}\right)}_{\xrightarrow{p}\boldsymbol{\Omega}} \underbrace{\left(\frac{1}{n}\mathbf{X}'\mathbf{X}\right)^{-1}}_{\xrightarrow{p}\mathbf{Q}},$$

which implies that  $Var[\hat{\boldsymbol{\beta}}|\mathbf{X}] \xrightarrow{p} \mathbf{0}$  by the Continuous Mapping Theorem and Slutsky's Theorem. Since  $\hat{\boldsymbol{\beta}}$  is unbiased, the LIE implies  $Var[\hat{\boldsymbol{\beta}}] \rightarrow \mathbf{0}$ , and Chebyshev's inequality implies that  $\hat{\boldsymbol{\beta}} \xrightarrow{p} \boldsymbol{\beta}$ .

## A.11 Asymptotic normality

Since  $\mathbf{X}_i u_i$  is i.i.d. and has bounded second moments by (A2) and (A3), the Multivariate Central Limit Theorem implies

$$\frac{1}{\sqrt{n}}\mathbf{X}'\mathbf{u} = \frac{1}{n} \sum_{i=1}^n \mathbf{X}_i u_i \xrightarrow{D} \mathcal{N}(\mathbf{0}, \boldsymbol{\Omega})$$

because  $E[\mathbf{X}_i u_i] = \mathbf{0}$  by (A1) and the LIE, and  $Var[\mathbf{X}_i u_i] = \boldsymbol{\Omega}$ . Then, by Slutsky's Theorem and the Continuous Mapping Theorem,

$$\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) = \left(\frac{1}{n}\mathbf{X}'\mathbf{X}\right)^{-1} \left(\frac{1}{\sqrt{n}}\mathbf{X}'\mathbf{u}\right) \xrightarrow{D} \mathbf{Q}^{-1}\mathcal{N}(\mathbf{0}, \boldsymbol{\Omega}),$$

and the result follows from the fact that  $Var[\mathbf{Q}^{-1}\mathcal{N}(\mathbf{0}, \boldsymbol{\Omega})] = \mathbf{Q}^{-1}\boldsymbol{\Omega}\mathbf{Q}^{-1}$ .

Note that by the decomposition above,  $\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) = \sqrt{n}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}$ , which is a linear combination of  $\mathbf{u}$ . Since linear combinations of normal variables are normal,  $\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})$  is normal under (A6) conditional on  $\mathbf{X}$  with  $E[\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})] = \mathbf{0}$  by the unbiasedness, and

$$Var[\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})] = nE[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{DX}(\mathbf{X}'\mathbf{X})^{-1}] \stackrel{(A5)}{=} \sigma^2\mathbf{Q}^{-1}.$$