Research Module in Econometrics & Statistics

Prof. Dominik Liebl (dliebl@uni-bonn.de) and Christopher Walsh (cwalsh@uni-bonn.de)

2021-10-05

Contents

Preface

This is the script for the research module in econometrics & statistics.

Repo that makes this site: https://github.com/lidom/RM_ES_Script

Research Module in Econometrics and Statistics

Description:

Lecture-Phase: The lecture-phase of this research module is intended to introduce the students to fundamental concepts of (applied and mathematical) statistics, as well as, to provide the students with reasoning skills for communicating statistical results. Participation in the lecture-phase is strongly recommended and active participation is desirable.

Project-Phase: The students have the opportunity to choose among a set of specific projects. Topics suggested by the students are generally appreciated, but will be assessed with respect to their feasibility. Each project must focus on one specific statistical method/topic (for instance, panel data analysis, clustered standard errors, non-parametric regression, etc.) and should contain the following three parts:

1. A critical description of the statistical method and its theoretical properties
2. Monte Carlo simulation studies to assess the finite sample properties of the statistical method
3. One or more real-data applications to showcase the practical use of the statistical/econometric method.

Depending on the actual number of participants, it might be that the project work has to be carried out as a group task rather than as an individual task.

Software: Due to the mathematical contents, it is strongly recommended to use LaTeX for preparing the presentation slides and the term papers. Moreover, the Monte Carlo simulations and the real-data applications will make it necessary to work with advanced software such as R, Python, or Matlab. Short introductions to LaTeX and R will be given during the lecture-phase, but the students should be willing to work with such software.

Grading: The final grade will be a weighted average of the presentation (40%) and the research paper (60%).

Registration: You need to register for this course via BASIS.

Registration period: Oct. 12-19. (**Caution:** No timely registration means you cannot participate in this course)

Time Table:

Date	Time	Topics*
12.10.	14:15 - 15:45	General Introduction / Introduction to R / Topic Choices [Liebl/Walsh]
13.10.	14:15 - 15:45	Introduction to R (Flipped Classroom) / Topic Choices [Liebl/Walsh]
19.10.	14:15 - 15:45	Estimation Theory [Walsh]
20.10.	14:15 - 15:45	Estimation Theory [Walsh]
26.10.	14:15 - 15:45	Regression / Monte-Carlo Simulations (Flipped Classroom) [Walsh]
27.10.	14:15 - 15:45	Test Theory [Liebl]
02.11.	14:15 - 15:45	Test Theory [Liebl]
03.11.	14:15 - 15:45	Monte-Carlo Simulations (Flipped Classroom) [Liebl]
09.11.	14:15 - 15:45	How to Write and Present [Walsh]
18.01.	14:15 - 15:45	Presentations [Liebl/Walsh]
19.01.	14:15 - 15:45	Presentations [Liebl/Walsh]

* This is just the approximate general structure. Depending on your pre-knowledge, we may deviate from this.

- **Virtual Lecture-Room (Zoom):** Zoom-Meeting Link
  Meeting-ID: 914 0066 5510
  Password: 544848
- **Supervision meetings:** From Nov. to Jan.
- **Scheduling of appointments:** HERE

Presentations:

- For groups of 1-2: 15-25 minutes
- For groups of 3: 20-25 minutes

Term Paper:

- Every term paper should consist of the following parts:

  - Introduction of the general problem and a short overview about the relevant literature.
  - Description of the considered method(s).
  - Assessment of the method(s) by means of Monte-Carlo simulations.
  - Application to real data.
- Page Count:
  - For groups of 1-2: 10-15 pages (plus bibliography and appendix)
  - For groups of 3: 15-20 pages (plus bibliography and appendix)
  - Long tables, proofs, additional figures, etc. should be placed in the appendix.
  - Line-spacing: 1.5
- **Deadline for submission of slides:** Jan. 17, 2022, via e-mail to dliebl@uni-bonn.de

- **Deadline for submission of term papers:** Feb. 11, 2022, via e-mail to dliebl@uni-bonn.de

Topics

- **Regression Discontinuity Designs** Literature: Imbens and Lemieux (2008)
- **Robust Inference** Literature: Cameron et al. (2011)
- **Testing for Systematically Missing Values** Literature: Little (1988)
- **Instrumental Variables** Literature: Davidson and MacKinnon (2004) Ch. 8
- **Difference-in-Differences Estimations** Literature: Bertrand et al. (2004), Cerulli (2015) Ch. 3.4
- **Multiple Testing** Literature: F. Bretz (2010), Lehmann and Romano (2006) Ch. 9
- **Panel Data Analysis** Literature: Hsiao (2014), Greene (2003), and Baltagi (2008)
- **Nonparametric Regression** Literature: Li and Racine (2007), Fan and Gijbels (1996), and Wand and Jones (1994)

Chapter 1

Introduction to R

This tutorial aims to serve as an introduction to the software package R. Other very good and much more exhaustive tutorials and useful reference-cards can be found at the following links:

- Reference card for R commands (always useful)
- Matlab/R reference card (for those who are more familiar with Matlab)
- The official Introduction to R (very detailed)
- And many more at www.r-project.org (see "Documents")
- An R-package for learning R: www.swirl.com
- An excellent book project which covers also advanced issues such as "writing performant code" and "package development": adv-r.had.co.nz

- Another excellent book: R for Data Science

Some other tutorials:

- Introduction to data science
- Scraping the web using R
- Creating dynamic graphics

Why R?

- R is **free** of charge from: www.r-project.org
- The celebrated IDE **RStudio** for R is also **free** of charge: www.rstudio.com
- R is equipped with one of the most flexible and powerful graphics routines available anywhere.
  For instance, check out one of the following repositories:
  - Clean Graphs
  - R graph catalog
  - Publication Ready Plots
- Today, R is the de-facto standard for statistical science.

1.1 Short Glossary

Lets start the tutorial with a (very) short glossary:

- **Console**: The thing with the "**>**" sign at the beginning.
- **Script file**: An ordinary text file with suffix "**.R**". For instance, **yourFavoritFileName.R**.
- **Working directory**: The file-directory you are working in. Useful commands: with `getwd()` you get the location of your current working directory and `setwd()` allows you to set a new location for it.
- **Workspace**: This is a hidden file (stored in the working directory), where all objects you use (e.g., data, matrices, vectors, variables, functions, etc.) are stored. Useful commands: `ls()` shows all elements in our current workspace and `rm(list=ls())` deletes all elements in our current workspace.

1.2 First Steps

A good idea is to use a script file such as **yourFavoritFileName.R** in order to store your R commands. You can send single lines or marked regions of your R-code to the console by pressing the keys **STRG+ENTER**.

To begin with baby steps, do some simple computations:

```
2+2 # and all the others: *,/,-, ^2, ^3,...
```

```
## [1] 4
```

Note: Everything that is written after the **#**-sign is ignored by R, which is very useful to comment your code.

The **assignment operator** will be your most often used tool. Here an example to create a **scalar** variable:

```
x <- 4
x
```

```
## [1] 4
```

```
4 -> x # possible but unusual
x
```

```
## [1] 4
```

Note: The R community loves the **<-** assignment operator, which is a very unusual syntax. Alternatively, you can use the **=** operator.

And now a more interesting object - a **vector**:

```
y <- c(2,7,4,1)
y
```

```
## [1] 2 7 4 1
```

The command `ls()` shows the total content of your current workspace, and the command `rm(list=ls())` deletes all elements of your current workspace:

```
ls()
```

```
## [1] "x" "y"
```

```
rm(list=ls())
ls()
```

```
## character(0)
```

Note: RStudio's **Environment** pane also lists all the elements in your current workspace. That is, the command `ls()` becomes a bit obsolete when working with RStudio.

Let's try how we can compute with vectors and scalars in R.

```
x <- 4
y <- c(2,7,4,1)

x*y # each element in the vector, y, is multiplied by the scalar, x.
```

```
## [1]  8 28 16  4
```

```
y*y # this is a term by term product of the elements in y
```

```
## [1]  4 49 16  1
```

Performing vector multiplications as you might expect from your last math-course, e.g., an outer product: $y\,y^\top$:

```
y %*% t(y)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    4   14    8    2
## [2,]   14   49   28    7
## [3,]    8   28   16    4
## [4,]    2    7    4    1
```

Or an inner product $y^\top y$:

```
t(y) %*% y
```

```
##      [,1]
## [1,]   70
```

Note: Sometimes, R's treatment of vectors can be annoying. The product `y %*% y` is treated as the product `t(y) %*% y`.

The term-by-term execution as in the above example, `y*y`, is actually a central strength of R. We can conduct many operations **vector-wisely**:

```
y^2
```

```
## [1]  4 49 16  1
```

```
log(y)
```

```
## [1] 0.6931472 1.9459101 1.3862944 0.0000000
```

```
exp(y)
```

```
## [1]    7.389056 1096.633158   54.598150    2.718282
```

```
y-mean(y)
```

```
## [1] -1.5  3.5  0.5 -2.5
```

```
(y-mean(y))/sd(y) # standardization
```

```
## [1] -0.5669467  1.3228757  0.1889822 -0.9449112
```

This is a central characteristic of so called matrix based languages like R (or Matlab). Other programming languages often have to use **loops** instead:

```
N <- length(y)
1:N

y.sq <- numeric(N)
y.sq

for(i in 1:N){
  y.sq[i] <- y[i]^2
  if(i == N){
    print(y.sq)
  }
}
```

The `for()`-loop is the most common loop. But there is also a `while()`-loop and a `repeat()`-loop. However, loops in R can be rather slow, therefore, try to avoid them!

Useful commands to produce **sequences** of numbers:

```
1:10
-10:10
?seq # Help for the seq()-function
seq(from=1, to=100, by=7)
```

Using the sequence command `1:16`, we can go for our first **matrix**:

```
?matrix
A <- matrix(data=1:16, nrow=4, ncol=4)
```

```
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    5    9   13
## [2,]    2    6   10   14
## [3,]    3    7   11   15
## [4,]    4    8   12   16
```

```
A <- matrix(1:16, 4, 4)
```

Note that a matrix has always two **dimensions**, but a vector has only one dimension:

```
dim(A)    # Dimension of matrix A?
```

```
## [1] 4 4
```

```
dim(y)    # dim() does not operate on vectors.
```

```
## NULL
```

```
length(y) # Length of vector y?
```

```
## [1] 4
```

Lets play a bit with the matrix `A` and the vector `y`. As we have seen in the loop above, the `[]`-operator **selects elements** of vectors and matrices:

```
A[,1]
A[4,4]
y[c(1,4)]
```

This can be done on a more **logical** basis, too. For example, if you want to know which elements in the first column of matrix `A` are strictly greater than 2:

```
A[,1][A[,1]>2]
```

```
## [1] 3 4
```

```
# Note that this give you a boolean vector:
A[,1]>2
```

```
## [1] FALSE FALSE  TRUE  TRUE
```

```
# And you can use it in a non-sense relation, too:
y[A[,1]>2]
```

```
## [1] 4 1
```

Note: Logical operations return so-called **boolean** objects, i.e., either a `TRUE` or a `FALSE`. For instance, if we ask R whether `1>2` we get the answer `FALSE`.

1.3 Further Data Objects

Besides classical data objects such as scalars, vectors, and matrices there are three further data objects in R:

1. The **array**: As a matrix but with more dimensions. Here is an example of a $2 \times 2 \times 2$-dimensional `array`:

```
myFirst.Array <- array(c(1:8), dim=c(2,2,2)) # Take a look at it!
```

2. The **list**: In `lists` you can organize different kinds of data. E.g., consider the following example:

```
myFirst.List <- list("Some_Numbers" = c(66, 76, 55, 12, 4, 66, 8, 99),
                     "Animals"      = c("Rabbit", "Cat", "Elefant"),
                     "My_Series"    = c(30:1))
```

A very useful function to find specific values and entries within lists is the `str()`-function:

```
str(myFirst.List)
```

```
## List of 3
##  $ Some_Numbers: num [1:8] 66 76 55 12 4 66 8 99
##  $ Animals     : chr [1:3] "Rabbit" "Cat" "Elefant"
##  $ My_Series   : int [1:30] 30 29 28 27 26 25 24 23 22 21 ...
```

3. The **data frame**: A `data.frame` is a `list`-object but with some more formal restrictions (e.g., equal number of rows for all columns). As indicated by its name, a `data.frame`-object is designed to store data:

```
myFirst.Dataframe <- data.frame("Credit_Default"   = c( 0, 0, 1, 0, 1, 1),
                                "Age"              = c(35,41,55,36,44,26),
                                "Loan_in_1000_EUR" = c(55,65,23,12,98,76))
# Take a look at it!
```

1.4 Simple Regression Analysis using R

Alright, let's do some statistics with real data. You can download the data HERE. Save it on your computer, at a place where you can find it, and give the path (e.g. `"C:\textbackslash path\textbackslash auto.data.csv"`, which references to the data, to the *file*-argument of the function `read.csv()`:

```
# ATTENTION! YOU HAVE TO CHANGE "\" TO "/":
auto.data <- read.csv(file="C:/your_path/autodata.txt", header=TRUE)
head(auto.data)
```

If you have problems to read the data into R, go on with these commands. (For this you need a working internet connection!):

```
# install.packages("readr")
library("readr")
auto.data <- suppressMessages(read_csv(file = "https://cdn.rawgit.com/lidom/Teaching_Repo/bc692b5
# head(auto.data)
```

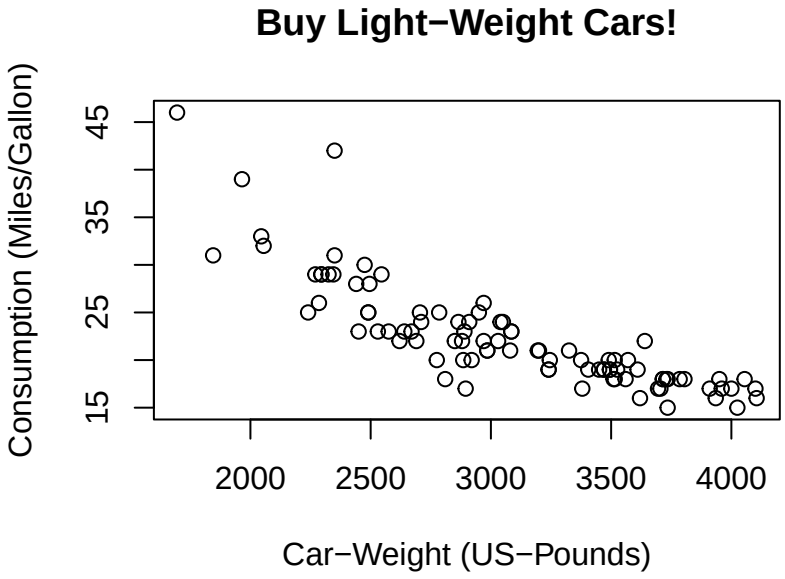
You can select specific variables of the `auto.data` using the `$`-operator:

```
gasolin.consumption      <- auto.data$MPG.city
car.weight               <- auto.data$Weight
## Take a look at the first elements of these vectors:
head(cbind(gasolin.consumption,car.weight))
```

```
##      gasolin.consumption car.weight
## [1,]                  25       2705
## [2,]                  18       3560
## [3,]                  20       3375
## [4,]                  19       3405
## [5,]                  22       3640
## [6,]                  22       2880
```

This is how you can produce your first plot:

```
## Plot the data:
plot(y=gasolin.consumption, x=car.weight,
     xlab="Car-Weight (US-Pounds)",
     ylab="Consumption (Miles/Gallon)",
     main="Buy Light-Weight Cars!")
```



As a first step, we might assume a simple kind of linear relationship between the variables `gasolin.consumption` and `car.weight`. Let us assume that the data was generated by the following simple regression model:

$$y_i = \alpha + \beta_1 x_i + \varepsilon_i, \quad i = 1, \ldots, n$$

where y_i denotes the gasoline-consumption, x_i the weight of car i, and ε_i is a mean zero constant variance noise term. (This is clearly a non-sense model!)

The command `lm()` computes the estimates of this linear regression model. The command (in fact it's a *method*) `summary()` computes further quantities of general interest from the *object* that was returned from the `lm()` function.

```
lm.result   <- lm(gasolin.consumption~car.weight)
lm.summary  <- summary(lm.result)
lm.summary
```

```
##
## Call:
## lm(formula = gasolin.consumption ~ car.weight)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7946 -1.9711  0.0249  1.1855 13.8278
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

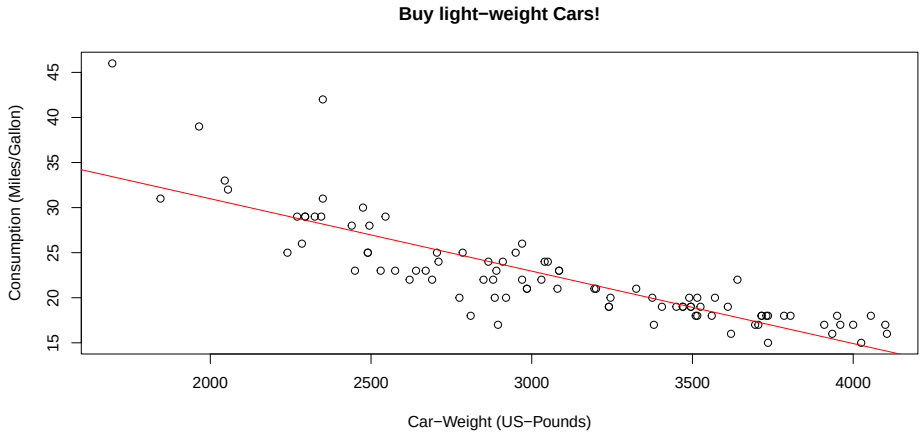
```
## (Intercept) 47.048353   1.679912   28.01   <2e-16 ***
## car.weight  -0.008032   0.000537  -14.96   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.038 on 91 degrees of freedom
## Multiple R-squared:  0.7109, Adjusted R-squared:  0.7077
## F-statistic: 223.8 on 1 and 91 DF,  p-value: < 2.2e-16
```

Of course, we want to have a possibility to access all the quantities computed so far, e.g., in order to plot the results. This can be done as following:

```
## Accessing the computed quantities
names(lm.summary) ## Alternatively: str(lm.summary)
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```
alpha <- lm.summary$coefficients[1]
beta  <- lm.summary$coefficients[2]

## Plot all:
plot(y=gasolin.consumption, x=car.weight,
     xlab="Car-Weight (US-Pounds)",
     ylab="Consumption (Miles/Gallon)",
     main="Buy light-weight Cars!")
abline(a=alpha,
       b=beta, col="red")
```



1.5 Programming in R

Let's write, i.e., program our own R-function for estimating linear regression models. In order to be able to validate our function, we start with **simulating data** for which we then *know* all true parameters. Simulating data is like being the "Data-God": For instance, we generate realizations of the error term ε_i, i.e., something which we *never* observe in real data.

Let us consider the following multiple regression model:

$$y_i = \beta_1 + \beta_2 x_{2i} + \beta_3 x_{3i} + \varepsilon_i, \quad i = 1, \dots, n,$$

where ε_i is a heteroscedastic error term

$$\varepsilon_i \sim N(0, \sigma_i^2), \quad \sigma_i = x_{3i},$$

and where for all $i = 1, \dots, n = 50$:

- $x_{2i} \sim N(10, 1.5^2)$
- x_{3i} comes from a t-distribution with 5 degrees of freedom and non-centrality parameter 2

```
set.seed(109) # Sets the "seed" of the random number generators:
n   <- 50     # Number of observations

## Generate two explanatory variables plus an intercept-variable:
X.1 <- rep(1, n)                  # Intercept
X.2 <- rnorm(n, mean=10, sd=1.5) # Draw realizations form a normal distr.
X.3 <- rt(n, df=5, ncp=2)         # Draw realizations form a t-distr.
X   <- cbind(X.1, X.2, X.3)       # Save as a Nx3-dimensional data matrix.
```

OK, we have regressors, i.e., data that we also have in real data sets.

Now we define the elements of the β-vector. Be aware of the difference: In real data sets we do not know the true β-vector, but try to estimate it. However, when simulating data, we determine (as "Data-Gods") the true β-vector and can compare our estimate $\hat{\beta}$ with the true β:

```
## Define the slope-coefficients
beta.vec  <- c(1,-5,5)
```

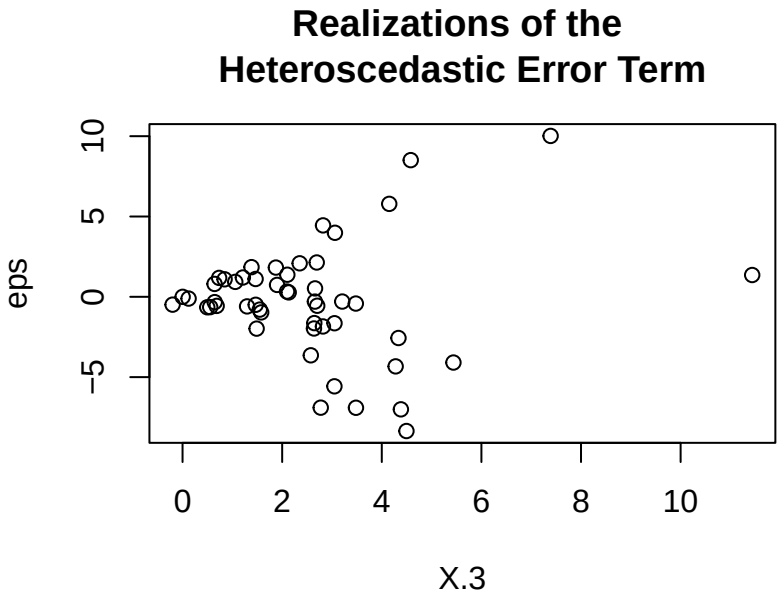
We still need to simulate realizations of the dependent variable y_i. Remember

that $y_i = \beta_1 x_{1i} + \beta_1 x_{2i} + \beta_3 x_{3i} + \varepsilon_i$. That is, we only need realizations from the error terms ε_i in order to compute the realizations from y_i. This is how you can simulate realizations from the heteroscedastic error terms ε_i:

```
## Generate realizations from the heteroscadastic error term
eps       <- (X.3)*rnorm(n, mean=0, sd=1)
```

Take a look at the heteroscedasticity in the error term:

```
plot(y=eps, x=X.3,
     main="Realizations of the \nHeteroscedastic Error Term")
```

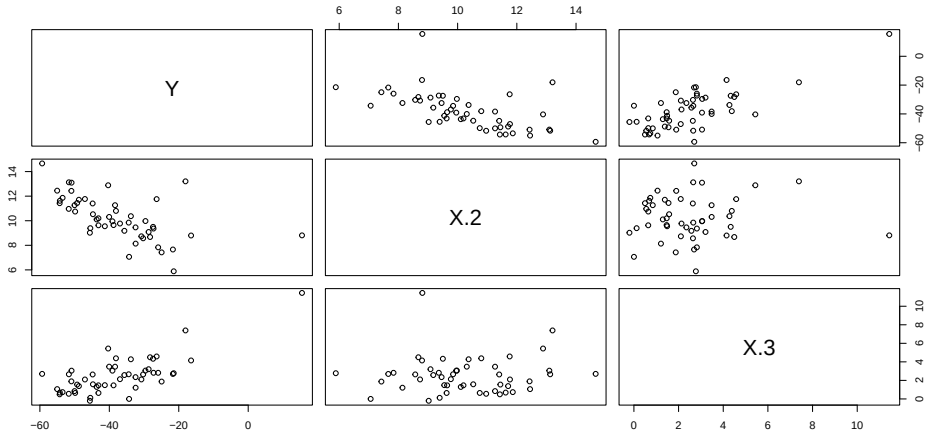


With the (pseudo-random) realizations from ε_i, we can finally generate realizations from the dependent variable y_i:

```
## Dependent variable:
y   <- X %*% beta.vec + eps
```

Let's take a look at the data:

```
mydata    <- data.frame("Y"=y, "X.1"=X.1, "X.2"=X.2, "X.3"=X.3)
pairs(mydata[,-2]) # The '-2' removes the intercept variable "X.1"
```



Once we have data, we can compute the OLS estimate of the true β vector. Remember the formula:

$$\hat{\beta} = (X^\top X)^{-1} X^\top y$$

In R-Code this is: $(X^\top X)^{-1}$ =`solve(t(X) %*% X)`, i.e.:

```
## Computation of the beta-Vector:
beta.hat <- solve(t(X) %*% X) %*% t(X) %*% y
beta.hat
```

```
##           [,1]
## X.1 -2.735042
## X.2 -4.685719
## X.3  5.091811
```

Well done. Using the above lines of code we can easily program our own `myOLSFun()` function!

```
myOLSFun <- function(y, x, add.intercept=FALSE){

  ## Number of Observations:
  n         <- length(y)

  ## Add an intercept to x:
  if(add.intercept){
    Intercept <- rep(1, n)
    x         <- cbind(Intercept, x)
  }
```

```
  ## Estimation of the slope-parameters:
  beta.hat.vec <- solve(t(x) %*% x) %*% t(x) %*% y

  ## Return the result:
  return(beta.hat.vec)
}

## Run the function:
myOLSFun(y=y, x=X)
```

```
##          [,1]
## X.1 -2.735042
## X.2 -4.685719
## X.3  5.091811
```

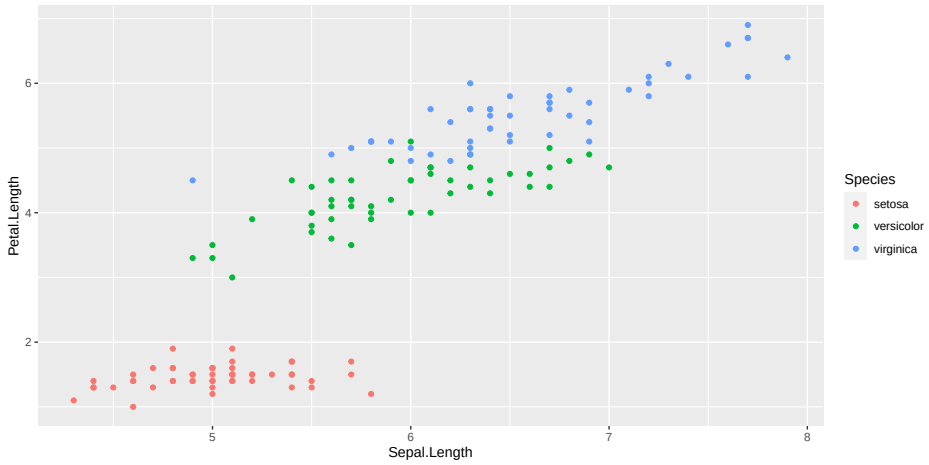
Can you extend the function for the computation of the covariance matrix of the slope-estimates, several measures of fits (R^2, adj.-R^2, etc.), t-tests, ...?

1.6 R-packages

One of the best features in R are its contributed packages. The list of all packages on CRAN is impressive! Take a look at it HERE

For instance, nice plots can be produced using the R-package is `ggplot2`. You can find an intro do this package HERE.

```
# install.packages("ggplot2")
library("ggplot2")

qplot(Sepal.Length, Petal.Length, data = iris, color = Species)
```



Of course, `ggplot2` concerns "only" plotting, but you'll find R-packages for almost any statistical method out there.

1.7 Tidyverse

The `tidyverse` package is a collection of packages that lets you import, manipulate, explore, visualize and model data in a harmonized and consistent way which helps you to be more productive.

Installing the `tidyverse` package:

```
install.packages("tidyverse")
```

To use the `tidyverse` package load it using the `library()` function:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v tibble  3.1.5     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v purrr   0.3.4     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Chick Weight Data

R comes with many datasets installed. We will use the `ChickWeight` dataset to learn about the tidyverse. The help system gives a basic summary of the

experiment from which the data was collect:

> *"The body weights of the chicks were measured at birth and every second day thereafter until day 20. They were also measured on day 21. There were four groups of chicks on different protein diets."*

You can get more information, including references by typing:

```
help("ChickWeight")
```

The Data: There are 578 observations (rows) and 4 variables:

- `Chick` – unique ID for each chick.
- `Diet` – one of four protein diets.
- `Time` – number of days since birth.
- `weight` – body weight of chick in grams.

Note: `weight` has a lower case `w` (recall R is case sensitive).

Store the data locally:

```
ChickWeight %>%
  select(Chick, Diet, Time, weight) %>%
  arrange(Chick, Diet, Time) %>%
  write_csv("ChickWeight.csv")
```

First we will import the data from a file called `ChickWeight.csv` using the `read_csv()` function from the `readr` package (part of the `tidyverse`). The first thing to do, outside of R, is to open the file `ChickWeight.csv` to check what it contains and that it makes sense. Now we can import the data as follows:

```
CW <- read_csv("ChickWeight.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Chick = col_double(),
##   Diet = col_double(),
##   Time = col_double(),
##   weight = col_double()
## )
```

If all goes well then the data is now stored in an R object called `CW`. If you get the following error message then you need to change the working directory to where the data is stored.

```
Error: 'ChickWeight.csv' does not exist in current
working directory ...
```

Changing the working directory: In RStudio you can use the menu bar ("Session - Set Working Directory - Choose Directory…"). Alternatively, you

can use the function `setwd()`.

Looking at the Dataset: To look at the data type just type the object (dataset) name:

```
CW
```

```
## # A tibble: 578 x 4
##    Chick  Diet  Time weight
##    <dbl> <dbl> <dbl>  <dbl>
##  1    18     1     0     39
##  2    18     1     2     35
##  3    16     1     0     41
##  4    16     1     2     45
##  5    16     1     4     49
##  6    16     1     6     51
##  7    16     1     8     57
##  8    16     1    10     51
##  9    16     1    12     54
## 10    15     1     0     41
## # ... with 568 more rows
```

If there are too many variables then not all them may be printed. To overcome this issue we can use the `glimpse()` function which makes it possible to see every column in your dataset (called a "data frame" in R speak).

```
glimpse(CW)
```

```
## Rows: 578
## Columns: 4
## $ Chick  <dbl> 18, 18, 16, 16, 16, 16, 16, 16, 16, 15, 15, 15, 15, 15, 15, 15,~
## $ Diet   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Time   <dbl> 0, 2, 0, 2, 4, 6, 8, 10, 12, 0, 2, 4, 6, 8, 10, 12, 14, 0, 2, 4~
## $ weight <dbl> 39, 35, 41, 45, 49, 51, 57, 51, 54, 41, 49, 56, 64, 68, 68, 67,~
```

The function `View()` allows for a spread-sheet type of view on the data:

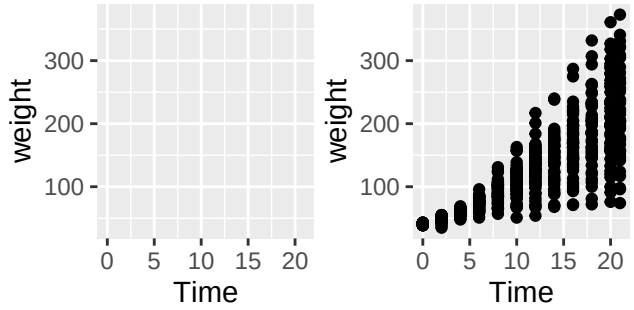
```
View(CW)
```

1.7.1 Tidyverse: Plotting Basics

To **visualise** the chick weight data, we will use the `ggplot2` package (part of the `tidyverse`). Our interest is in seeing how the *weight changes over time for the chicks by diet*. For the moment don't worry too much about the details just try to build your own understanding and logic. To learn more try different things even if you get an error messages.

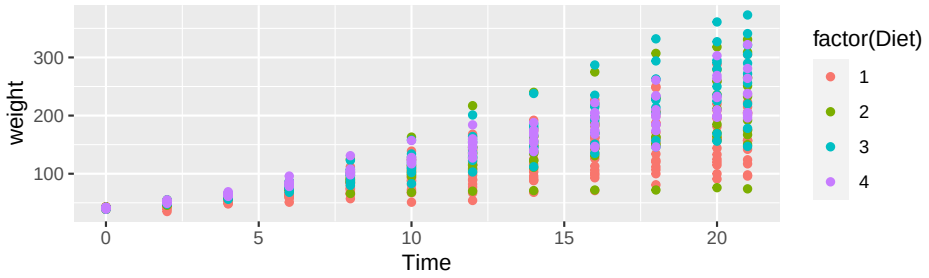
Let's plot the weight data (vertical axis) over time (horizontal axis).

```
# An empty plot (the plot on the left)
ggplot(CW, aes(Time, weight))
# With data (the plot on the right)
ggplot(CW, aes(Time, weight)) + geom_point()
```



Add color for `Diet`. The graph above does not differentiate between the diets. Let's use a different color for each diet.

```
# Adding colour for diet
ggplot(CW,aes(Time,weight,colour=factor(Diet))) +
  geom_point()
```



It is difficult to conclude anything from this graph as the points are printed on top of one another (with diet 1 underneath and diet 4 at the top).

Factor Variables: Before we continue, we have to make an important change to the `CW` dataset by making `Diet` and `Time` *factor variables*. This means that R will treat them as categorical variables (see the `<fct>` variables below) instead of continuous variables. It will simplify our coding. The next section will explain the `mutate()` function.

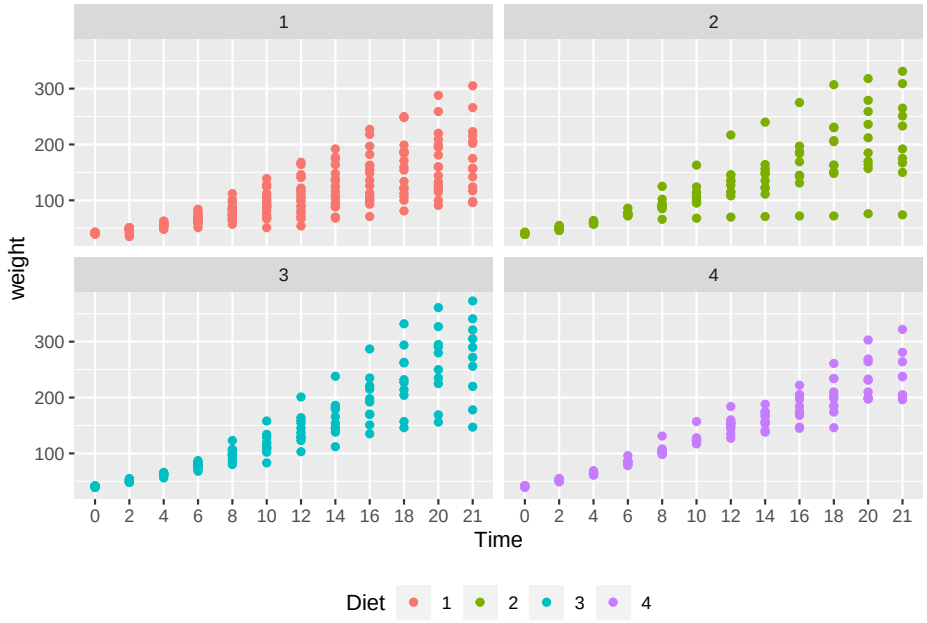
```
CW <- mutate(CW, Diet = factor(Diet))
CW <- mutate(CW, Time = factor(Time))
glimpse(CW)
```

```
## Rows: 578
## Columns: 4
```

```
## $ Chick  <dbl> 18, 18, 16, 16, 16, 16, 16, 16, 16, 15, 15, 15, 15, 15, 15, 15,~
## $ Diet   <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Time   <fct> 0, 2, 0, 2, 4, 6, 8, 10, 12, 0, 2, 4, 6, 8, 10, 12, 14, 0, 2, 4~
## $ weight <dbl> 39, 35, 41, 45, 49, 51, 57, 51, 54, 41, 49, 56, 64, 68, 68, 67,~
```

The `facet_wrap()` function: To plot each diet separately in a grid using `facet_wrap()`:

```
# Adding jitter to the points
ggplot(CW, aes(Time, weight, colour=Diet)) +
  geom_point() +
  facet_wrap(~Diet) +
  theme(legend.position = "bottom")
```

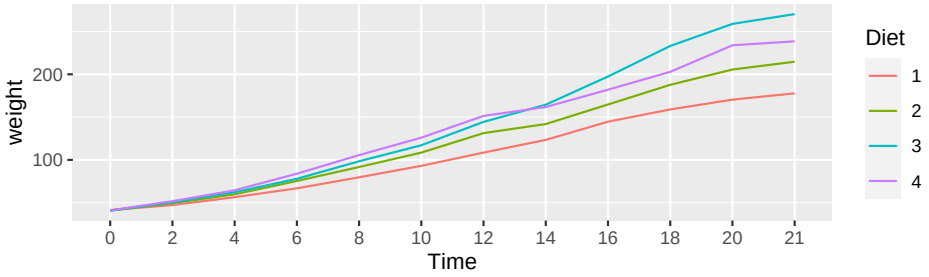


Interpretation: Diet 4 has the least variability but we can't really say anything about the mean effect of each diet although diet 3 seems to have the highest.

Next we will plot the **mean changes** over time for each diet using the `stat_summary()` function:

```
ggplot(CW, aes(Time, weight,
               group=Diet, colour=Diet)) +
  stat_summary(fun.y="mean", geom="line")
```

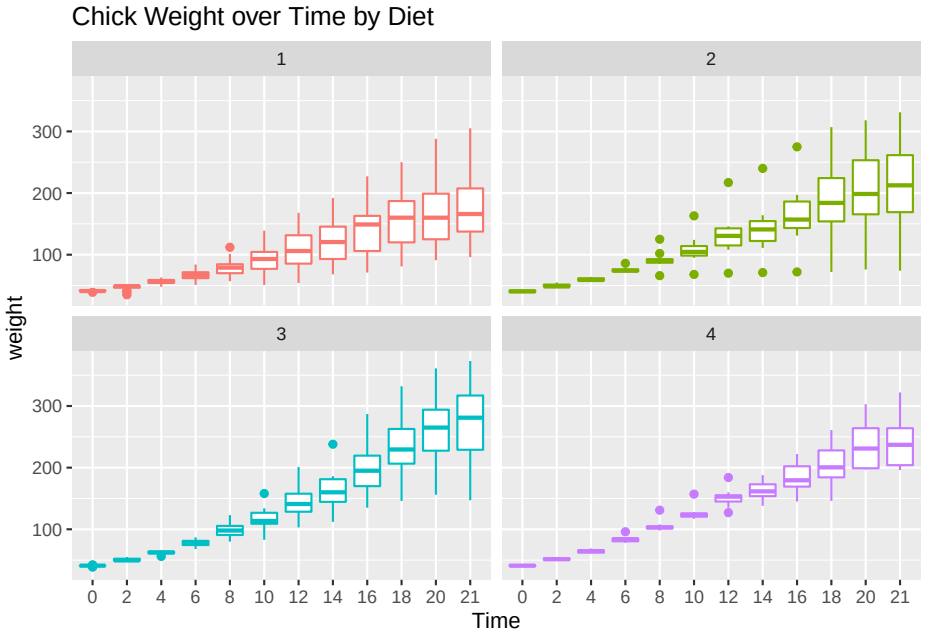
```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```



Interpretation: We can see that diet 3 has the highest mean weight gains by the end of the experiment. However, we don't have any information about the variation (uncertainty) in the data.

To see variation between the different diets we use `geom_boxplot` to plot a box-whisker plot. A note of caution is that the number of chicks per diet is relatively low to produce this plot.

```
ggplot(CW, aes(Time, weight, colour=Diet)) +
  facet_wrap(~Diet) +
  geom_boxplot() +
  theme(legend.position = "none") +
  ggtitle("Chick Weight over Time by Diet")
```

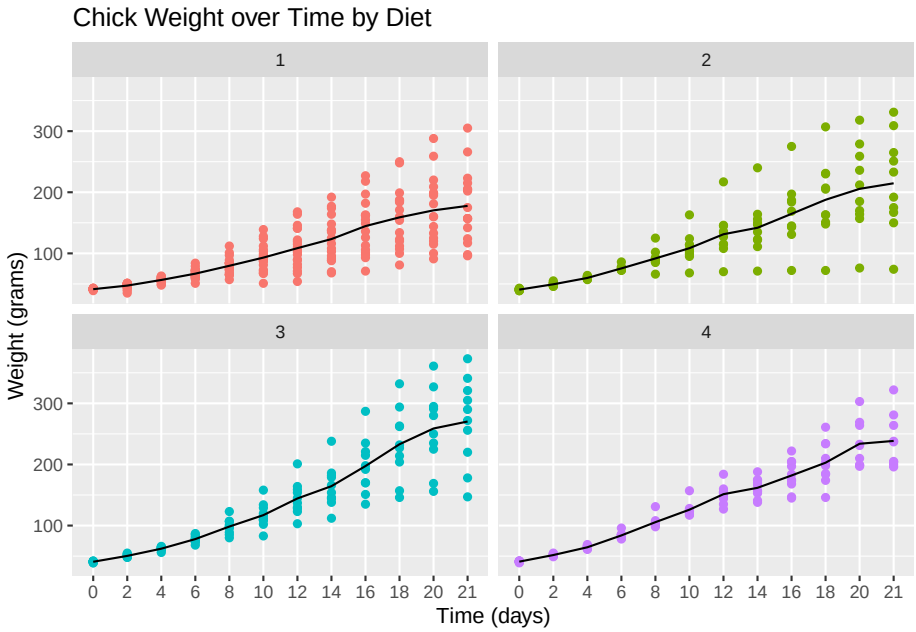


Interpretation: Diet 3 seems to have the highest "average" weight gain but it has more variation than diet 4 which is consistent with our findings so far.

Let's finish with a plot that you might include in a publication.

```
ggplot(CW, aes(Time, weight, group=Diet,
                              colour=Diet)) +
  facet_wrap(~Diet) +
  geom_point() +
  # geom_jitter() +
  stat_summary(fun.y="mean", geom="line",
               colour="black") +
  theme(legend.position = "none") +
  ggtitle("Chick Weight over Time by Diet") +
  xlab("Time (days)") +
  ylab("Weight (grams)")
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```



1.7.2 Tidyverse: Data Wrangling Basics

In this section we will learn how to wrangle (manipulate) datasets using the `tidyverse` package. Let's start with the `mutate()`, `select()`, `rename()`, `filter()` and `arrange()` functions.

`mutate()`: Adds a new variable (column) or modifies an existing one. We already used this above to create factor variables.

```
# Added a column
CWm1 <- mutate(CW, weightKg = weight/1000)
CWm1
```

```
## # A tibble: 578 x 5
##   Chick Diet  Time  weight weightKg
##   <dbl> <fct> <fct>  <dbl>    <dbl>
## 1    18 1     0         39    0.039
## 2    18 1     2         35    0.035
## 3    16 1     0         41    0.041
## # ... with 575 more rows
```

```
# Modify an existing column
CWm2 <- mutate(CW, Diet = str_c("Diet ", Diet))
CWm2
```

```
## # A tibble: 578 x 4
##   Chick Diet   Time  weight
##   <dbl> <chr>  <fct>  <dbl>
## 1    18 Diet 1 0         39
## 2    18 Diet 1 2         35
## 3    16 Diet 1 0         41
## # ... with 575 more rows
```

`select()`: Keeps, drops or reorders variables.

```
# Drop the weight variable from CWm1 using minus
select(CWm1, -weight)
```

```
## # A tibble: 578 x 4
##   Chick Diet  Time  weightKg
##   <dbl> <fct> <fct>    <dbl>
## 1    18 1     0        0.039
## 2    18 1     2        0.035
## 3    16 1     0        0.041
## # ... with 575 more rows
```

```
# Keep variables Time, Diet and weightKg
select(CWm1, Chick, Time, Diet, weightKg)
```

```
## # A tibble: 578 x 4
##   Chick Time  Diet  weightKg
##   <dbl> <fct> <fct>    <dbl>
## 1    18 0     1        0.039
## 2    18 2     1        0.035
## 3    16 0     1        0.041
```

```
## # ... with 575 more rows
```

`rename()`: Renames variables whilst keeping all variables.

```
rename(CW, Group = Diet, Weight = weight)
```

```
## # A tibble: 578 x 4
##   Chick Group Time  Weight
##   <dbl> <fct> <fct>  <dbl>
## 1    18 1     0         39
## 2    18 1     2         35
## 3    16 1     0         41
## # ... with 575 more rows
```

`filter()`: Keeps or drops observations (rows).

```
filter(CW, Time==21 & weight>300)
```

```
## # A tibble: 8 x 4
##   Chick Diet  Time  weight
##   <dbl> <fct> <fct>  <dbl>
## 1     7 1     21       305
## 2    29 2     21       309
## 3    21 2     21       331
## # ... with 5 more rows
```

For comparing values in vectors use: `<` (less than), `>` (greater than), `<=` (less than and equal to), `>=` (greater than and equal to), `==` (equal to) and `!=` (not equal to). These can be combined logically using `&` (and) and `|` (or).

`arrange()`: Changes the order of the observations.

```
arrange(CW, Chick, Time)
```

```
## # A tibble: 578 x 4
##   Chick Diet  Time  weight
##   <dbl> <fct> <fct>  <dbl>
## 1     1 1     0         42
## 2     1 1     2         51
## 3     1 1     4         59
## # ... with 575 more rows
```

```
arrange(CW, desc(weight))
```

```
## # A tibble: 578 x 4
##   Chick Diet  Time  weight
##   <dbl> <fct> <fct>  <dbl>
## 1    35 3     21       373
## 2    35 3     20       361
## 3    34 3     21       341
## # ... with 575 more rows
```

What does the `desc()` do? Try using `desc(Time)`.

1.7.3 The pipe operator %>%

In reality you will end up doing multiple data wrangling steps that you want to save. The pipe operator `%>%` makes your code nice and readable:

```
CW21 <- CW %>%
  filter(Time %in% c(0, 21)) %>%
  rename(Weight = weight) %>%
  mutate(Group = factor(str_c("Diet ", Diet))) %>%
  select(Chick, Group, Time, Weight) %>%
  arrange(Chick, Time)
CW21
```

```
## # A tibble: 95 x 4
##   Chick Group  Time  Weight
##   <dbl> <fct>  <fct>  <dbl>
## 1     1 Diet 1 0         42
## 2     1 Diet 1 21       205
## 3     2 Diet 1 0         40
## # ... with 92 more rows
```

Hint: To understand the code above we should read the pipe operator `%>%` as "then".

> Create a new dataset (object) called `CW21` using dataset `CW` ***then*** keep the data for days 0 and 21 ***then*** rename variable `weight` to `Weight` ***then*** create a variable called `Group` ***then*** keep variables `Chick`, `Group`, `Time` and `Weight` and ***then*** finally arrange the data by variables `Chick` and `Time`.

This is the same code:

```
CW21 <- CW %>%
  filter(., Time %in% c(0, 21)) %>%
  rename(., Weight = weight) %>%
  mutate(., Group=factor(str_c("Diet ",Diet))) %>%
```

```
  select(., Chick, Group, Time, Weight) %>%
  arrange(., Chick, Time)
```

The pipe operator, `%>%`, replaces the dots (`.`) with whatever is returned from code preceding it. For example, the dot in `filter(., Time %in% c(0, 21))` is replaced by `CW`. The output of the `filter(...)` then replaces the dot in `rename(., Weight = weight)` and so on. Think of it as a data assembly line with each function doing its thing and passing it to the next.

1.7.4 The `group_by()` function

From the data visualizations above we concluded that the diet 3 has the highest mean and diet 4 the least variation. In this section, we will quantify the effects of the diets using **summmary statistics**. We start by looking at the number of observations and the mean by **diet** and **time**.

```
mnsdCW <- CW %>%
  group_by(Diet, Time) %>%
  summarise(N = n(), Mean = mean(weight)) %>%
  arrange(Diet, Time)
```

```
## `summarise()` has grouped output by 'Diet'. You can override using the `.groups` arg
```

```
mnsdCW
```

```
## # A tibble: 48 x 4
## # Groups:   Diet [4]
##   Diet  Time      N  Mean
##   <fct> <fct> <int> <dbl>
## 1 1     0        20  41.4
## 2 1     2        20  47.2
## 3 1     4        19  56.5
## # ... with 45 more rows
```

For each distinct combination of `Diet` and `Time`, the chick weight data is summarized into the number of observations (`N`) and the mean (`Mean`) of `weight`.

Further summaries: Let's also calculate the standard deviation, median, minimum and maximum values but only at days 0 and 21.

```
sumCW <-  CW %>%
  filter(Time %in% c(0, 21)) %>%
  group_by(Diet, Time) %>%
  summarise(N = n(),
            Mean = mean(weight),
            SD = sd(weight),
            Median = median(weight),
            Min = min(weight),
```

```
            Max = max(weight)) %>%
  arrange(Diet, Time)
```

```
## `summarise()` has grouped output by 'Diet'. You can override using the `.groups` argument.
```

```
sumCW
```

```
## # A tibble: 8 x 8
## # Groups:   Diet [4]
##   Diet  Time      N  Mean     SD Median   Min   Max
##   <fct> <fct> <int> <dbl>  <dbl>  <dbl> <dbl> <dbl>
## 1 1     0        20  41.4  0.995   41      39    43
## 2 1     21       16 178.  58.7    166      96   305
## 3 2     0        10  40.7  1.49    40.5    39    43
## # ... with 5 more rows
```

Let's make the summaries "prettier", say, for a report or publication.

```
library("knitr") # to use the kable() function
prettySumCW <- sumCW %>%
 mutate(`Mean (SD)` = str_c(format(Mean, digits=1),
           " (", format(SD, digits=2), ")")) %>%
 mutate(Range = str_c(Min, " - ", Max)) %>%
 select(Diet, Time, N, `Mean (SD)`, Median, Range) %>%
 arrange(Diet, Time) %>%
 kable(format = "latex")
prettySumCW
```

Diet	Time	N	Mean (SD)	Median	Range
1	0	20	41 (0.99)	41.0	39 - 43
1	21	16	178 (58.70)	166.0	96 - 305
2	0	10	41 (1.5)	40.5	39 - 43
2	21	10	215 (78.1)	212.5	74 - 331
3	0	10	41 (1)	41.0	39 - 42
3	21	10	270 (72)	281.0	147 - 373
4	0	10	41 (1.1)	41.0	39 - 42
4	21	9	239 (43.3)	237.0	196 - 322

Interpretation: This summary table offers the same interpretation as before, namely that diet 3 has the highest mean and median weights at day 21 but a higher variation than group 4. However it should be noted that at day 21, diet 1 lost 4 chicks from 20 that started and diet 4 lost 1 from 10. This could be a sign of some health related issues.

1.8 Further Links

1.8.1 Further R-Intros

- https://eddelbuettel.github.io/gsir-te/Getting-Started-in-R.pdf
- https://www.datacamp.com/courses/free-introduction-to-r
- https://swcarpentry.github.io/r-novice-gapminder/
- https://support.rstudio.com/hc/en-us/articles/200526207-Using-Projects

1.8.2 Version Control (Git/GitHub)

- https://support.rstudio.com/hc/en-us/articles/200532077-Version-Control-with-Git-and-SVN
- http://happygitwithr.com/
- https://www.gitkraken.com/

1.8.3 R-Ladies

- https://rladies.org/

Chapter 2

Estimation Theory

2.1 Bias, Variance and MSE

Given a sample $X_1, \dots, X_n$ consider an estimator $\hat{\theta}_n \equiv \hat{\theta}(X_1, \dots, X_n)$ of a real-valued parameter $\theta \in \Omega \subset \mathbb{R}$.

The distribution of any estimator of course depends on the true parameter vector θ, i.e., more precisely,

$$\hat{\theta}_n \equiv \hat{\theta}(X_1, \dots, X_n; \theta).$$

This dependence is usually not explicitly written, but all properties of estimators discussed below have to hold for all possible parameter values $\theta \in \Omega$. This will go without saying.

Statistical inference requires to assess the accuracy of an estimator.

The **bias** of an estimator is defined by

$$\text{Bias}(\hat{\theta}_n) = E(\hat{\theta}_n) - \theta$$

An estimator is called **unbiased** if $E(\hat{\theta}_n) = \theta$ and hence $\text{Bias}(\hat{\theta}_n) = 0$ (for all possible $\theta \in \Omega$).

The **variance** of an estimator is given by

$$\text{var}(\hat{\theta}_n) = E\left((\hat{\theta}_n - E(\hat{\theta}_n))^2\right).$$

Performance of an estimator is most frequently evaluated with respect to the **quadratic loss** (also called L_2 loss)

$$(\hat{\theta}_n - \theta)^2.$$

The corresponding risk is the **Mean Squared Error (MSE)**

$$E\left((\hat{\theta}_n - \theta)^2\right) = \text{Bias}(\hat{\theta}_n)^2 + \text{var}(\hat{\theta}_n)$$

For an unbiased estimator the mean squared error is obviously equal to the variance of the estimator.

Example: Assume an i.i.d. sample $X_1, \ldots, X_n$ with mean $\mu = E(X_i)$ and variance $\sigma^2 = \text{var}(X_i) < \infty$.

- The sample mean $\bar{X}$ is an unbiased estimator of the true mean μ, since the equation
$$E(\bar{X}) = \mu$$
holds for any possible value of the true mean μ.
- The variance of the estimator $\bar{X}$ is given by
$$\text{var}(\bar{X}) = \sigma^2/n$$
- The mean squared error of the estimator $\bar{X}$ is given by
$$E\left((\bar{X} - \mu)^2\right) = \text{var}(\bar{X}) = \sigma^2/n$$

2.2 Consistency of Estimators

Asymptotic theory is concerned with theoretical results valid for "large sample sizes". Important keywords of asymptotic theory are:

- consistency
- rates of convergence
- asymptotic distributions

They all rely on elaborated concepts on the stochastic convergence of random variables.

Stochastic convergence. Let $\{Z_n\}_{n=1,2,3,\ldots}$ be a sequence of **random variables**. Mathematically, there are different kinds of convergence of $\{Z_n\}$ to a fixed value c. The three most important are:

- Convergence in probability (abbreviated $Z_n \to_P c$):

$$\lim_{n\to\infty} P\left(|Z_n - c| > \epsilon\right) = 0 \quad \text{for all} \quad \epsilon > 0$$

- Almost sure convergence (abbreviated $Z_n \to_{a.s.} c$):

$$P\left(\lim_{n\to\infty} Z_n = c\right) = 1$$

- Convergence in quadratic mean (abbreviated $Z_n \to_{q.m.} c$):

$$\lim_{n\to\infty} E\left((Z_n - c)^2\right) = 0$$

Note that:

- $Z_n \to_{a.s.} c$ implies $Z_n \to_P c$
- $Z_n \to_{q.m.} c$ implies $Z_n \to_P c$

Consistency of estimators. Based on a sample $X_1, \ldots, X_n$ let $\hat{\theta}_n \equiv \theta_n(X_1, \ldots, X_n)$ be an estimator of an unknown parameter θ.

- $\hat{\theta}_n$ is called "weakly consistent" if

$$\hat{\theta}_n \to_P \theta \quad \text{as} \quad n \to \infty$$

- $\hat{\theta}_n$ is called "strongly consistent" if

$$\hat{\theta}_n \to_{a.s.} \theta \quad \text{as} \quad n \to \infty$$

Remark: For most statistical estimation problems it is usually possible to define many different estimators. The real problem is to find a good estimator which approximates the true parameter θ with the maximal possible accuracy. Consistency is generally seen as a necessary condition which has to be satisfied by any reasonable estimator. In econometric practice usually only weak consistency is derived which generally follows from weak **laws of large numbers**.

Example: Assume again an i.i.d. sample $X_1, \ldots, X_n$ with mean $\mu = E(X_i)$ and variance $\sigma^2 = \text{var}(X_i) < \infty$. As stated above we then have

$$E\left((\bar{X} - \mu)^2\right) = \text{var}(\bar{X}) = \sigma^2/n \to 0 \quad \text{as } n \to \infty.$$

Therefore, $\bar{X} \to_{q.m.} \mu$. The latter implies that $\bar{X} \to_P \mu$, i.e. $\bar{X}$ is a (weakly) consistent estimator of μ.

2.3 Rates of Convergence

Rates of convergence quantify the (stochastic) order of magnitude of an estimation error in dependence of the sample size n. This order of magnitude is usually represented using the symbols: O_P and o_P.

Let $\{Z_n\}_{n=1,2,3,...}$ be a sequence of random variables, and let $\{c_n\}_{n=1,2,3,...}$ be a sequence of positive (deterministic) numbers.

- We will write $Z_n = O_p(c_n)$ if for any $\epsilon > 0$ there exist numbers $0 < M < \infty$ and m such that

$$P(|Z_n| \geq M \cdot c_n) \leq \epsilon \quad \text{for all} \quad n \geq m.$$

- We will write $Z_n = o_p(c_n)$ if

$$\lim_{n\to\infty} P(|Z_n| \geq \epsilon \cdot c_n) = 0 \quad \text{for all} \quad \epsilon > 0.$$

- With $c_n = 1$ for all n, $Z_n = O_p(1)$ means that the sequence $\{Z_n\}$ is **stochastically bounded**. I.e., for any $\epsilon > 0$ there exist number $0 < M < \infty$ and m such that

$$P(|Z_n| \geq M) \leq \epsilon \quad \text{for all} \quad n \geq m.$$

- With $c_n = 1$ for all n, $Z_n = o_P(1)$ is equivalent to $Z_n \to_P 0$, i.e., Z_n **converges in probability to zero**.

Note that:

- $Z_n = O_p(c_n)$ is equivalent to $Z_n/c_n = O_p(1)$
- $Z_n = o_p(c_n)$ is equivalent to $Z_n/c_n = o_p(1)$

Definition: An estimator $\hat{\theta} \equiv \hat{\theta}_n$ of a parameter θ possesses the **rate of convergence** n^{-r} if and only if r is the *largest positive number* with the property that

$$|\hat{\theta}_n - \theta| = O_P(n^{-r}).$$

The rate of convergence quantifies how fast the estimation error decreases when increasing the sample size n.

Unbiased estimators: Let $\hat{\theta}_n$ be an *unbiased* estimator of an unknown parameter θ satisfying $\text{var}(\hat{\theta}_n) = Cn^{-1}$ for some $0 < C < \infty$. Then $\hat{\theta}_n$ possesses the rate of convergence $n^{-1/2}$. This is a consequence of the Chebyshev inequality.

Chebyshev inequality: If Z denotes a random variable with mean μ and variance σ^2, then

$$P\left(|X-\mu| > \sigma \cdot m\right) \leq \frac{1}{m^2} \quad \text{for all} \quad m > 0$$

$$\Rightarrow P\left(|\hat{\theta}_n - \theta| > n^{-1/2}\sqrt{C} \cdot \frac{1}{\sqrt{\epsilon}}\right) \leq \epsilon \quad \text{for all} \quad \epsilon > 0$$

Example: Assume an i.i.d. sample $X_1, \ldots, X_n$ with mean $\mu = E(X_i)$ and variance $\sigma^2 = \text{var}(X_i) < \infty$. The sample mean $\bar{X}$ ($\equiv \bar{X}_n$) is an unbiased estimator of μ with variance $\text{var}(\bar{X}) = \sigma^2/n$. For large n we have by the central limit theorem that approximately $\sqrt{n}(\bar{X}-\mu) \sim N(0, \sigma^2)$. Therefore, for example:

- with $\epsilon = 0.05$ we obtain

$$P\left(|\bar{X}_n - \mu| \geq 1.96\sigma \cdot n^{-1/2}\right) = 0.05$$

- with $\epsilon = 0.01$ we obtain

$$P\left(|\bar{X}_n - \mu| \geq 2.64\sigma \cdot n^{-1/2}\right) = 0.01.$$

Generalizing this argument for all possible $\epsilon > 0$ we can conclude that $\bar{X} - \mu = O_P(n^{-1/2})$. On the other hand for any $r > 1/2$ we have $n^{-r}/n^{-1/2} \to 0$ as $n \to \infty$. Hence, for any constant $c > 0$

$$\begin{aligned} &P\left(|\bar{X}_n - \mu| \geq c\sigma \cdot n^{-r}\right) = \\ =&P\left(|\bar{X}_n - \mu| \geq (c\sigma \cdot n^{-1/2}) \cdot \frac{n^{-r}}{n^{-1/2}}\right) \to 1 \quad \text{as} \quad n \to \infty. \end{aligned}$$

Therefore $n^{-1/2}$ is the **rate of convergence** of $\bar{X}$.

Note that:

- Maximum-likelihood estimators of an unknown parameter usually possess the rate of convergence $n^{-1/2}$ (there are exceptions!).
- The situation is different, for instance, in nonparametric curve estimation problems. For example kernel estimators (of a density or regression function) only achieve the rate of convergence $n^{-2/5}$.

- The rate of convergence is an important criterion for selecting the best possible estimator for a given problem. For most parametric problems it is well known that the optimal (i.e. fastest possible) convergence rate is $n^{-1/2}$. In nonparametric regression or density estimation the optimal convergence rate is only $n^{-2/5}$, if the underlying function is twice continuously differentiable.

O_P-rules:

- We have

$$Z_n \to_P Z \qquad \text{if and only if} \qquad Z_n = Z + o_p(1)$$

  This follows from $Z_n = Z + (Z_n - Z)$ and $Z_n - Z \to_P 0$.

- If $Z_n = O_P(n^{-\delta})$ for some $\delta > 0$, then $Z_n = o_P(1)$

- If $Z_n = O_P(r_n)$, then $Z_n^\delta = O_P(r_n^\delta)$ for any $\delta > 0$. Similarly, $Z_n = o_P(r_n)$ implies $Z_n^\delta = o_P(r_n^\delta)$ for any $\delta > 0$.

- If $Z_n = O_P(r_n)$ and $V_n = O_P(s_n)$, then

$$Z_n + V_n = O_P(\max\{r_n, s_n\})$$
$$Z_n V_n = O_P(r_n s_n)$$

- If $Z_n = o_P(r_n)$ and $V_n = O_P(s_n)$, then $Z_n V_n = o_P(r_n s_n)$

- If $E(|Z_n|^k) = O(r_n)$, then $Z_n = O_p(|r_n|^{1/k})$ for $k = 1, 2, 3, ...$

2.4 Asymptotic Distributions

The practically most important version of stochastic convergence is convergence in distribution. Knowledge about the "asymptotic distribution" of an estimator allows to construct confidence intervals and tests.

Definition: Let Z_n be a sequence of random variables with corresponding distribution functions G_n. Then Z_n **converges in distribution** to a random variable Z with distribution function G, if

$$G_n(x) \to G(x) \quad \text{as} \quad n \to \infty$$

at all continuity points x of G (abbreviated: $Z_n \to_L Z$ or $Z_n \to_L G$ or "$\to_D$" instead of "$\to_L$").

In a vast majority of practically important situation the limiting distribution is the normal distribution. One then speaks of **asymptotic normality**. Asymptotic normality is usually a consequence of central limit theorems. The simplest result in this direction is the central limit theorem of Lindeberg-Levy.

Theorem (Lindeberg-Levy) Let $Z_1, Z_2, ...$ be a sequence of i.i.d. random variables with finite mean μ and variance $\sigma^2 < \infty$. Then

$$\sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n} Z_i - \mu\right) \to_L N(0, \sigma^2).$$

Example: Let $X_1, ..., X_n$ be independent random variables with $E(X_i) = \mu$, $Var(X_i) = \sigma^2$. Then the central limit theorem of Lindeberg-Levy implies that

$$\sqrt{n}(\bar{X} - \mu) \to_L N(0, \sigma^2) \quad \text{or equivalently} \quad \frac{\sqrt{n}(\bar{X} - \mu)}{\sigma} \to_L N(0, 1).$$

We can conclude that $\bar{X}$ is an "asymptotically normal estimator" of μ. If n is sufficiently large, then $\bar{X}$ is approximatively normal with mean μ and variance σ^2/n. Frequently used notations:

- $\bar{X} \sim AN(\mu, \sigma^2/n)$
- $\bar{X} \overset{a}{\sim} N(\mu, \sigma^2/n)$

Most estimators $\hat{\theta}_n$ used in parametric and nonparametric statistics are asymptotically normal. In parametric problems (with rate of convergence $n^{-1/2}$) one usually obtains

$$\sqrt{n}(\hat{\theta}_n - \theta) \to_L N(0, v^2),$$

where v^2 is the asymptotic variance of the estimator (often, but not necessarily, $v^2 = \lim_{n\to\infty} n \cdot \mathrm{var}(\hat{\theta}_n)$).

Multivariate generalization: The above concepts are easily generalized to estimators $\hat{\theta}_n$ of a multivariate parameter vector $\theta \in \mathbb{R}^p$. Consistency and rates of convergence then have to be derived separately for each element of the vector. Convergence in distribution is defined via convergence of the multivariate distribution functions. For standard estimators (e.g., maximum likelihood) in parametric problems one usually obtains

$$\sqrt{n}(\hat{\theta}_n - \theta) \to_L N_p(0, V),$$

where V is the asymptotic covariance matrix (usually, $V = \lim_{n\to\infty} n \cdot \mathrm{Cov}(\hat{\theta}_n)$).

Multivariate normality holds if and only if for any vector $c = (c_1, ..., c_p)' \in \mathbb{R}^p$ with $\sum_{j=1}^p c_j^2 = \|c\|_2^2 = 1$

$$\sqrt{n}\left(\sum_{j=1}^{p} c_j(\hat{\theta}_{jn} - \theta_j)\right) = \sqrt{n}\left(c'\hat{\theta}_n - c'\theta\right) \to_L N\left(0, v_c^2\right),$$

where

$$v_c^2 = c'Vc = \sum_{j=1}^{p}\sum_{k=1}^{p} c_j c_k V_{jk},$$

and where V_{jk} are the elements of the asymptotic covariance matrix V.

This condition is frequently called **"Cramer-Wold device"**. Using one-dimensional central limit theorems it can be verified for any vector c.

Example: Let $X_1 = (X_{11}, X_{12})', ..., X_n = (X_{n1}, X_{n2})'$ be i.i.d. two-dimensional random vectors with $E(X_i) = \mu = (\mu_1, \mu_2)'$ and $Cov(X_i) = \Sigma$. The Cramer-Wold device and Lindeberg-Levy's central limit theorem then imply that

$$\sqrt{n}\left(\bar{X} - \mu\right) \to_L N_2\left(0, \Sigma\right).$$

Note that asymptotic normality usually also holds for nonparametric curve estimators with convergence rates slower than $n^{-1/2}$.

2.5 Asymptotic Theory

Many estimation procedures in modern statistics rely on fairly general assumptions. For a given sample size n it is then often impossible to derive the exact distribution of θ_n. Necessary calculations are too complex, and finite sample distributions usually depend on unknown characteristics of the distribution of the underlying data.

The goal of asymptotic theory then is to derive reasonable approximations. For **large samples** such approximations are of course very accurate, for **small samples** there may exist a considerable approximation error. Therefore, for small samples the approximation quality of asymptotic approximations is usually studied by Monte-Carlo approximations.

Asymptotic theory is used in order to select an appropriate estimation procedure in complex situations. The idea is to determine the estimator which, at least for large sample sizes, provides the smallest possible estimation error. This leads to the concept of "asymptotically efficient" estimators.

Properties of an asymptotically efficient estimator θ_n:

- For the estimation problem to be considered θ_n is consistent and adopts the fastest possible rate of convergence (generally: $n^{-1/2}$ in parametric statistics; $n^{-2/5}$ can be achieved in nonparametric univariate curve estimation problems).
- In most regular situations one is additionally interested in a "best asymptotically normal" (BAN) estimator. Assume that $\sqrt{n}(\theta_n - \theta) \sim N(0, v^2)$. Then θ_n is a BAN-estimator if any alternative estimator $\tilde{\theta}_n$ with $\sqrt{n}(\tilde{\theta}_n - \theta) \sim N(0, \tilde{v}^2)$ possesses a larger asymptotic variance, i.e. $\tilde{v}^2 \geq v^2$.
- **Multivariate generalization:** An estimator θ_n with $\sqrt{n}(\theta_n - \theta) \sim N_p(0, V)$ is best asymptotically normal if

$$c'\tilde{V}c \geq c'Vc \quad \text{for all} \quad c \in \mathbb{R}^p, \|c\|_2^2 = 1$$

  for any other estimator $\tilde{\theta}_n$ satisfying $\sqrt{n}(\tilde{\theta}_n - \theta) \sim N_p(0, \tilde{V})$.

For most estimation problems in parametric statistics maximum-likelihood estimators are best asymptotically normal.

2.6 Mathematical tools

2.6.1 Taylor expansions

Taylors' theorem: Let f be a real-valued function which is $k+1$ continuously differentiable in the interior of an interval $[a,b]$. Consider a point $x_0 \in (a,b)$. For any other value $x \in (a,b)$ there exists some $\psi \in [x_0, x]$ such that

$$f(x) = f(x_0) + \sum_{r=1}^{k} \frac{1}{r!} f^{(r)}(x_0) \cdot (x - x_0)^r + \frac{1}{(k+1)!} f^{(k+1)}(\psi) \cdot (x - x_0)^{k+1}$$

Qualitative version of Taylors' formula:

$$f(x) = f(x_0) + \sum_{r=1}^{k} \frac{1}{r!} f^{(r)}(x_0) \cdot (x - x_0)^r + O((x - x_0)^{k+1})$$

Example: Let $f(x) = ln(x)$ und $x_0 = 1 \Rightarrow f'(x_0) = 1$, $f''(x_0) = -1$.

First order Taylor approximation: $f(x) = \tilde{f}(x) + O((x - x_0)^2)$, where $\tilde{f}(x) = x - x_0$

- $x = 1.05 \Rightarrow f(x) = 0.04879$, $\tilde{f}(x) = 0.05$ and $|f(x) - \tilde{f}(x)| = 0.00121$
- $x = 1.1 \Rightarrow f(x) = 0.09531$, $\tilde{f}(x) = 0.1$ and $|f(x) - \tilde{f}(x)| = 0.00469$
- $x = 1.5 \Rightarrow f(x) = 0.40546$, $\tilde{f}(x) = 0.5$ and $|f(x) - \tilde{f}(x)| = 0.09454$
- $x = 2 \Rightarrow f(x) = 0.69315$, $\tilde{f}(x) = 1$ and $|f(x) - \tilde{f}(x)| = 0.30685$

Second order Taylor approximation: $f(x) = \tilde{f}(x) + O((x - x_0)^3)$, where $\tilde{f}(x) = x - x_0 - \frac{1}{2}(x - x_0)^2$

- $x = 1.05 \Rightarrow f(x) = 0.04879$, $\tilde{f}(x) = 0.04875$ and $|f(x) - \tilde{f}(x)| = 0.00004$
- $x = 1.1 \Rightarrow f(x) = 0.09531$, $\tilde{f}(x) = 0.95$ and $|f(x) - \tilde{f}(x)| = 0.00031$
- $x = 1.5 \Rightarrow f(x) = 0.40546$, $\tilde{f}(x) = 0.375$ and $|f(x) - \tilde{f}(x)| = 0.03046$
- $x = 2 \Rightarrow f(x) = 0.69315$, $\tilde{f}(x) = 0.5$ and $|f(x) - \tilde{f}(x)| = 0.19315$

Multivariate generalization: $x_0, x \in \mathbb{R}^p$, $f'(x_0) \in \mathbb{R}^p$, $f''(x_0)$ a $p \times p$ Matrix.

First order Taylor approximation:

$$f(x) = f(x_0) + f'(x_0) \cdot (x - x_0) + O(\|x - x_0\|_2^2)$$

Second order Taylor approximation:

$$f(x) = f(x_0) + f'(x_0) \cdot (x - x_0) + \frac{1}{2}(x - x_0)^T f''(x_0)(x - x_0) + O(\|x - x_0\|_2^3)$$

2.6.2 Tools for deriving asymptotic distributions

Let $\{W_n\}$, $\{Z_n\}$ be sequences of random variables, then:

- $Z_n = W_n + o_P(1) \quad \Leftrightarrow \quad Z_n - W_n \to_P 0$. If additionally $W_n \to_L N(0, v^2)$ then $Z_n \to_L N(0, v^2)$.
- For any fixed constant $c \neq 0$: If $Z_n \to_P c$ and $W_n \to_L N(0, v^2)$, then

  $$cW_n \to_L N(0, c^2v^2) \quad \text{as well as} \quad V_n := Z_n \cdot W_n \to_L N(0, c^2v^2).$$

  Furthermore, If Z_n and c are positive (with probability 1) then also

  $$W_n/c \to_L N(0, v^2/c^2) \quad \text{as well as} \quad V_n := W_n/Z_n \to_L N(0, v^2/c^2).$$

- Multivariate generalization (C, Z_n $p \times p$ matrices; W_n p-dimensional random vectors): If $Z_n \to_P C$ as well as $W_n \to_L N_p(0, V)$, then

  $$\begin{aligned} CW_n &\to_L N_p(0, CVC') \quad \text{as well as} \\ V_n := Z_n \cdot W_n &\to_L N_p(0, CVC') \end{aligned}$$

2.6.3 The Delta-Method

A further tool which is frequently used in asymptotic statistics is the so-called delta-method.

Delta-Method: Let $\hat{\theta}_n$ be a sequence of estimators of a one-dimensional parameter θ satisfying $n^r(\hat{\theta}_n - \theta) \to_L N(0, v^2)$, and let $g(.)$ be a real-valued function which is continuously differentiable at θ and satisfies $g'(\theta) \neq 0$. Then

$$n^r \left(g(\hat{\theta}_n) - g(\theta) \right) \to_L N\left(0, g'(\theta)^2 v^2\right).$$

Example: Assume an i.i.d. random sample $X_1, \ldots, X_n$ from an exponential distribution. That is, the underlying density of X_i, $i = 1, \ldots, n$, is given by $f(x|\theta) = \theta \exp(-\theta x)$. We then have $\mu := E(X_i) = 1/\theta$ as well as $\sigma_X^2 := \text{var}(X_i) = 1/\theta^2$. The underlying parameter $\theta > 0$ is unknown and has to be estimated from the data.

The maximum-likelihood estimator of θ is $\widehat{\theta} = 1/\bar{X}$.

We know that $\sqrt{n}(\bar{X} - \frac{1}{\theta}) \rightarrow_L N(0, \frac{1}{\theta^2})$, but what's about the distribution of $1/\bar{X}$? For this purpose the delta-method can be applied with $g(x) = 1/x$. Then $g'(x) = -1/x^2$, $g'(1/\theta) = -\theta^2$, and consequently

$$n^{1/2}\left(\frac{1}{\bar{X}} - \theta\right) = n^{1/2}\left(g\left(\bar{X}\right) - g\left(\frac{1}{\theta}\right)\right) \rightarrow_L N\left(0, \theta^2\right).$$

Chapter 3

Statistical Hypothesis Testing

3.1 Hypotheses and Test-Statistics

Assume an independently and identically distributed (i.i.d.) random sample $X_1, \dots, X_n$, where the distribution of X_i, $i = 1, \dots, n$, depends on some unknown parameter $\theta \in \Omega$, and where Ω is some parameter space.

General Testing Problem:

$$H_0 : \theta \in \Omega_0$$

against

$$H_1 : \theta \in \Omega_1$$

H_0 is the null hypothesis, while H_1 is the alternative. $\Omega_0 \subset \Omega$ and $\Omega_1 \subset \Omega$ are used to denote the possible values of θ under H_0 and H_1. Necessarily, $\Omega_0 \cap \Omega_1 = \emptyset$.

For a large number of tests we have $\Omega \subseteq \mathbb{R}$ and the respective null hypothesis states that θ has a specific value $\theta_0 \in \mathbb{R}$, i.e., $\Omega_0 = \{\theta_0\}$ and $H_0 : \theta = \theta_0$. Depending on the alternative one then often distinguishes between one-sided ($\Omega_1 = (\theta_0, \infty)$ or $\Omega_1 = (-\infty, \theta_0)$) and two-sided tests ($\Omega_1 = \{\theta \in \mathbb{R} | \theta \neq \theta_0\}$).

The data $X_1, \dots, X_n$ is used in order to decide whether to accept or to reject H_0.

Test Statistic: Every statistical hypothesis test relies on a corresponding test statistic

$$T = T(X_1, \dots, X_n).$$

Any test statistic is a real valued random variable, and for given data the resulting observed value T_{obs} is used to decide between H_0 and H_1. Generally, the distribution of T under H_0 is analyzed in order to define a **rejection region** C:

- $T_{obs} \notin C \Rightarrow H_0$ is not rejected
- $T_{obs} \in C \Rightarrow H_0$ is rejected

For one-sided tests C is typically of the form $(-\infty, c_0]$ or $[c_1, \infty)$. For two-sided tests C typically takes the form of $(-\infty, c_0] \cup [c_1, \infty)$. The limits c_0 and c_1 of the respective intervals are called **critical values**, and are obtained from quantiles of the **null distribution**, i.e., the distribution of T under H_0.

Decision Errors:

Decision Errors	Verbal Definition	Probability of a Type I/II Error
Type I error	H_0 is rejected even though H_0 is true.	$P(T \in C \mid H_0 \text{ true})$
Type II error	The test fails to reject a false H_0.	$P(T \notin C \mid H_1 \text{ true})$

3.2 Significance Level, Size and p-Values

Significance Level: In a statistical significance test, the probability of a type I error is controlled by the *significance level* α (e.g., $\alpha = 5\%$).

$$P\left(\text{Type I error}\right) = \sup_{\theta \in \Omega_0} P(T \in C | \theta \in \Omega_0) \leq \alpha$$

Size: The *size* of a statistical test is defined as

$$\sup_{\theta \in \Omega_0} P(T \in C | \theta \in \Omega_0).$$

That is, the preselected significance level α is an upper bound for the size, which

may not be attained (i.e., size $< \alpha$) if, for instance, the relevant probability function is discrete.

Practically important significance levels:

- $\alpha = 0.05$: It is common to say that a test result is "significant" if a hypothesis test of level $\alpha = 0.05$ rejects H_0.
- $\alpha = 0.01$: It is common to say that a test result is "strongly significant" if a hypothesis test of level $\alpha = 0.01$ rejects H_0.

p-Value: The *p-value* is the probability of obtaining a test statistic at least as "extreme" as the one that was actually observed, assuming that the null hypothesis is true.

- For one-sided tests:
  - $P(T \geq T_{\text{obs}}|H_0 \text{ true})$ or
  - $P(T \leq T_{\text{obs}}|H_0 \text{ true})$
- For two-sided tests:
  - $2\min\{P(T \leq T_{\text{obs}}|H_0 \text{ true}), P(T \geq T_{\text{obs}}|H_0 \text{ true})\}$

Remarks:

- The p-value is random as it depends on the observed data. That is, different random samples will lead to different p-values.
- For given data, having determined the p-value of a test we also know the test decisions for all possible levels α:
  - $\alpha >$ p-value $\Rightarrow H_0$ is rejected
  - $\alpha <$ p-value $\Rightarrow H_0$ cannot be rejected

Example: Let $X_i \sim N(\mu, \sigma^2)$ independently for all $i = 1, \ldots, 5 = n$. Observed realizations from this i.i.d. random sample: $X_1 = 19.20$, $X_2 = 17.40$, $X_3 = 18.50$, $X_4 = 16.50$, $X_5 = 18.90$. That is, the empirical mean is given by $\bar{X} = 18.1$.

Testing problem: $H_0 : \mu = \mu_0$ against $H_1 : \mu \neq \mu_0$ (i.e., a two-sided test), where $\mu_0 = 17$.

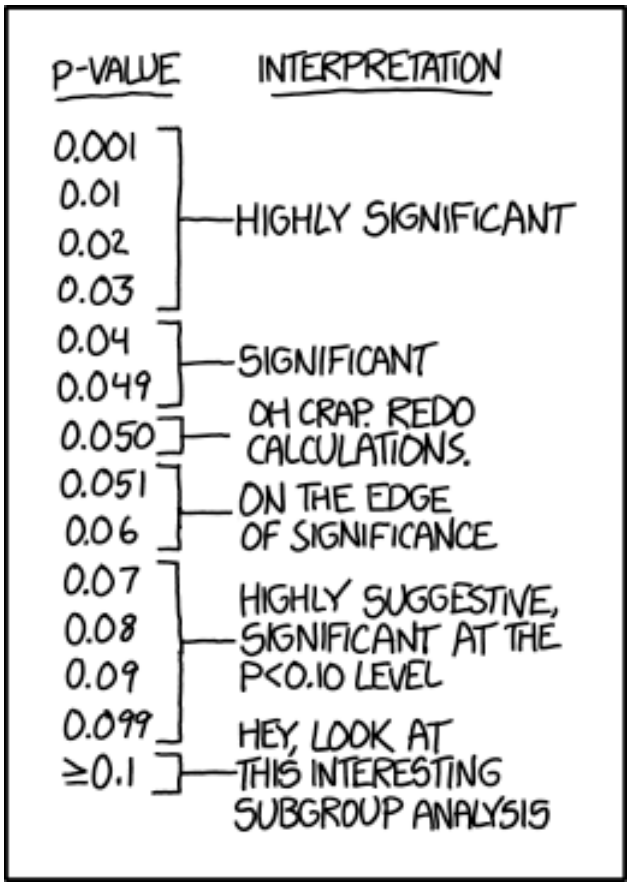


Figure 3.1: From: https://xkcd.com/1478/

Since the variance is unknown, we use the sample standard deviation, s, which then leads to the **t-test** for testing H_0. Test statistic of the t-test:

$$T = \frac{\sqrt{n}(\bar{X} - \mu_0)}{s},$$

where $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$ is the unbiased estimator of σ^2.

$$T_{obs} = \frac{\sqrt{5}(18.1 - 17)}{1.125} = 2.187$$

$$\Rightarrow \text{p-value} = 2\min\{P(T_{n-1} \leq 2.187),\ P(T_{n-1} \geq 2.187)\} = 0.094$$

The above computations in R

```
library("magrittr", quietly = TRUE)# for using the pipe-operator: %>%

X           <- c(19.20, 17.40, 18.50, 16.50, 18.90)
mu_0        <- 17         # hypothetical mean
n           <- length(X) # sample size
X_mean      <- mean(X)    # empirical mean
X_sd        <- sd(X)      # empirical sd
# t-test statistic
t_test_stat <- sqrt(n)*(X_mean - mu_0)/X_sd

# p-value for two-sided test
c(pt(q = t_test_stat, df = n-1, lower.tail = TRUE),
  pt(q = t_test_stat, df = n-1, lower.tail = FALSE)) %>%
  min * 2 -> p_value

p_value %>% round(., digits = 3)
```

```
## [1] 0.094
```

Of course, there is also a `t.test()` function in R:

```
t.test(X, mu = mu_0, alternative = "two.sided")
```

```
##
##  One Sample t-test
##
## data:  X
## t = 2.1869, df = 4, p-value = 0.09402
## alternative hypothesis: true mean is not equal to 17
```

```
## 95 percent confidence interval:
##  16.70347 19.49653
## sample estimates:
## mean of x
##      18.1
```

3.3 The Power Function

For every possible value $\theta \in \Omega_0 \cup \Omega_1$, all sample sizes n and each significance level α the corresponding value of the **power function** β is defined by the following probability:

$$\beta_{n,\alpha}(\theta) := P(\text{reject } H_0 \mid \theta \in \Omega_0 \cup \Omega_1)$$

Obviously, $\beta_{n,\alpha}(\theta) \leq \alpha$ for all $\theta \in \Omega_0$. Furthermore, for any $\theta \in \Omega_1$, $1 - \beta_{n,\alpha}(\theta)$ is the probability of committing a type II error.

The power function is an important tool for accessing the quality of a test and for comparing different test procedures.

Conservative Test: If possible, a test is constructed in such a way that size equals level, i.e., $\beta_{n,\alpha}(\theta) = \alpha$ for some $\theta \in \Omega_0$. In some cases, however, as for discrete test statistics or complex, composite null hypothesis, it is not possible to reach the level, and $\sup_{\theta \in \Omega_0} \beta_{n,\alpha}(\theta) < \alpha$. In this case the test is called *conservative*.

Unbiased Test: A significance test of level $\alpha > 0$ is called *unbiased* if $\beta_{n,\alpha}(\theta) \geq \alpha$ for all $\theta \in \Omega_1$.

Consistent Test: A significance test of level $\alpha > 0$ is called *consistent* if

$$\lim_{n \to \infty} \beta_{n,\alpha}(\theta) = 1$$

for all $\theta \in \Omega_1$.

Most Powerful Test: When choosing between different testing procedures for the same testing problem, one will usually prefer the *most powerful test*. Consider a fixed sample size n. For a specified $\theta \in \Omega_1$, a test with power function $\beta_{n,\alpha}(\theta)$ is said to be **most powerful** for θ if for any alternative test with power function $\beta^*_{n,\alpha}(\theta)$,

$$\beta_{n,\alpha}(\theta) \geq \beta^*_{n,\alpha}(\theta)$$

holds for all levels $\alpha > 0$.

Uniformly Most Powerful: A test with power function $\beta_{n,\alpha}(\theta)$ is said to be *uniformly most powerful* against the set of alternatives Ω_1 if for any alternative test with power function $\beta^*_{n,\alpha}(\theta)$,

$$\beta_{n,\alpha}(\theta) \geq \beta^*_{n,\alpha}(\theta) \quad \text{holds for all } \theta \in \Omega_1, \alpha > 0$$

Unfortunately, uniformly most powerful tests only exist for very special testing problems.

Example: Let $X_1, \ldots, X_n$ be an i.i.d. random sample. Assume that $n = 9$, and that $X_i \sim N(\mu, 0.18^2)$. Hence, in this simple example only the mean $\mu = E(X)$ is unknown, while the standard deviation has the known value $\sigma = 0.18$.

Testing problem: $H_0 : \mu = \mu_0$ against $H_1 : \mu \neq \mu_0$ for $\mu_0 = 18.3$ (i.e., a two-sided test).

Since the variance is known, a test may rely on the Gauss (or Z) test statistic:

$$Z = \frac{\sqrt{n}(\bar{X} - \mu_0)}{\sigma} = \frac{3(\bar{X} - 18.3)}{0.18}$$

Under H_0 we have $Z \sim N(0, 1)$, and for the significance level $\alpha = 0.05$ the null hypothesis is rejected if

$$|Z| \geq z_{1-\alpha/2} = 1.96,$$

where $z_{1-\alpha/2}$ denotes the $(1-\alpha/2)$-quantile of the standard normal distribution. Note that the size of this test equals its level $\alpha = 0.05$.

For determining the rejection region of a test it suffices to determine the distribution of the test statistic under H_0. But in order to calculate the power

function one needs to quantify the distribution of the test statistic for all possible values $\theta \in \Omega$. For many important problems this is a formidable task. For the Gauss test, however, it is quite easy. Note that for any (true) mean value $\mu \in \mathbb{R}$ the corresponding distribution of $Z \equiv Z_\mu = \sqrt{n(\bar{X} - \mu_0)}/\sigma$ is

$$Z_\mu = \frac{\sqrt{n}(\mu - \mu_0)}{\sigma} + \frac{\sqrt{n}(\bar{X} - \mu)}{\sigma} \sim N\left(\frac{\sqrt{n}(\mu - \mu_0)}{\sigma}, 1\right)$$

This implies that

$$\begin{aligned}\beta_{n,\alpha}(\mu) &= P\left(|Z_\mu| > z_{1-\alpha/2}\right) \\ &= 1 - \Phi\left(z_{1-\alpha/2} - \frac{\sqrt{n}(\mu - \mu_0)}{\sigma}\right) + \Phi\left(-z_{1-\alpha/2} - \frac{\sqrt{n}(\mu - \mu_0)}{\sigma}\right),\end{aligned}$$

where Φ denotes the distribution function of the standard normal distribution.

Implementing the power function of the two-sided Z-test in R:

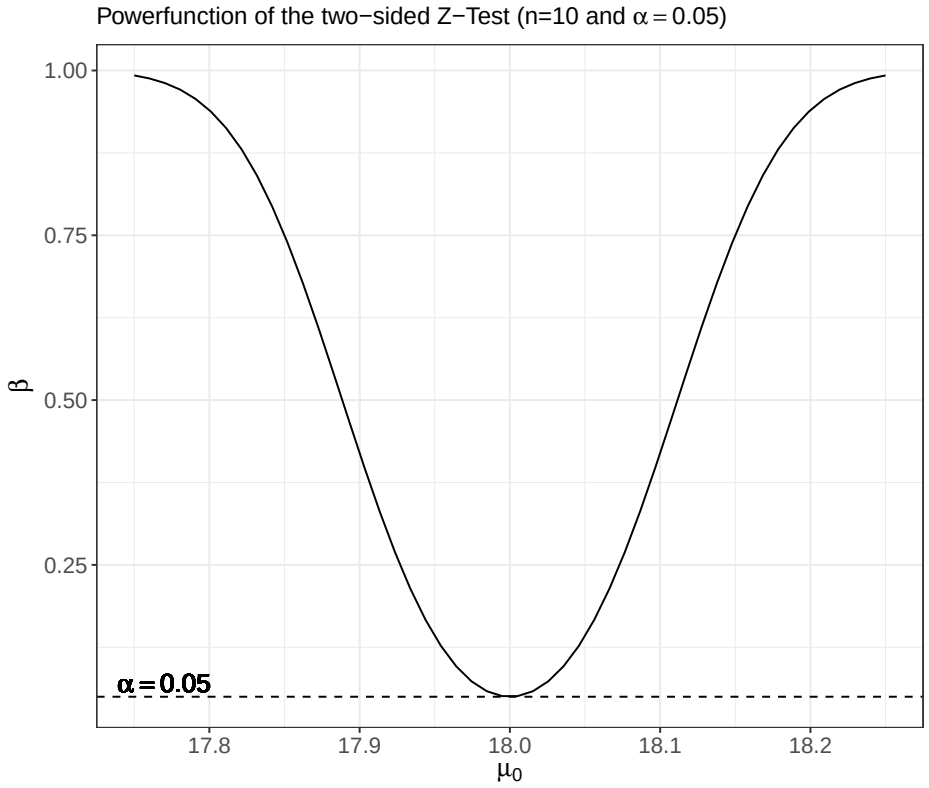
```
# The power function
beta_Ztest_TwoSided <- function(n, alpha, sigma, mu_0, mu){
  # (1-alpha/2)-quantile of N(0,1):
  z_upper        <- qnorm(p = 1-alpha/2)
  # location shift under H_1:
  location_shift <- sqrt(n) * (mu - mu_0)/sigma
  # compute power
  power          <- 1 - pnorm( z_upper - location_shift) +
                        pnorm(-z_upper - location_shift)
  return(power)
}

# Apply the function
n     <-  9
sigma <-  0.18
mu_0  <- 18.3
##
c(beta_Ztest_TwoSided(n = n, alpha = 0.05, sigma = sigma, mu_0 = mu_0, mu=18.35),
  beta_Ztest_TwoSided(n = n, alpha = 0.05, sigma = sigma, mu_0 = mu_0, mu=18.50),
  beta_Ztest_TwoSided(n = n, alpha = 0.01, sigma = sigma, mu_0 = mu_0, mu=18.50)) %>%
  round(., digits = 3)
```

```
## [1] 0.133 0.915 0.776
```

Plotting the graph of the power function

```
suppressPackageStartupMessages(
  library("tidyverse")
)
# Vectorize the function with respect to mu_0:
```

```
beta_Ztest_TwoSided <- Vectorize(FUN = beta_Ztest_TwoSided,
                                 vectorize.args = "mu_0")

mu_0_vec <- seq(from = 17.75, to = 18.25, len = 50)

beta_vec <- beta_Ztest_TwoSided(n     =   10,
                                alpha = 0.05,
                                sigma = 0.18,
                                mu    =  18,
                                mu_0  = mu_0_vec)

beta_df <- data.frame("mu_0"  = mu_0_vec,
                      "Beta"  = beta_vec)

ggplot(data = beta_df, aes(x=mu_0, y=Beta)) +
  geom_line() +
  geom_hline(yintercept = 0.05, lty=2) +
  geom_text(aes(x=17.77, y=0.07, label='alpha==0.05'), parse=TRUE, size=5) +
  labs(title = expression(
    paste("Powerfunction of the two-sided Z-Test (n=10 and ",alpha==0.05,")")),
       x = expression(paste(mu[0])),
       y = expression(paste(beta)), size=8)    +
  theme_bw() +
  theme(axis.text  = element_text(size=12),
          axis.title = element_text(size=14))
```



This example illustrates the power function of a sensible test, since:

- Under $H_0 : \mu = \mu_0$ we have $\beta_{n,\alpha}(\mu_0) = \alpha$.
- The test is unbiased, since $\beta_{n,\alpha}(\mu) \geq \alpha$ for any $\mu \neq \mu_0$.
- The test is consistent, since $\lim_{n\to\infty} \beta_{n,\alpha}(\mu) = 1$ for every fixed $\mu \neq \mu_0$.
- For fixed sample size n, $\beta_{n,\alpha}(\mu)$ increases as the distance $|\mu - \mu_0|$ increases.
- If $|\mu - \mu_0| > |\mu^* - \mu_0|$ then $\beta_{n,\alpha}(\mu) > \beta_{n,\alpha}(\mu^*)$.
- $\beta_{n,\alpha}(\mu)$ decreases as the significance level α of the test decreases. I.e., if $\alpha > \alpha^*$ then $\beta_{n,\alpha}(\mu) > \beta_{n,\alpha^*}(\mu)$.

Assuming that the basic assumptions (i.e., normality and known variance) are true, the above Gauss-test is the most prominent example of a *uniformly most powerful* test. Under its (restrictive) assumptions, no other possible test can achieve a larger value of $\beta_{n,\alpha}(\mu)$ for any possible value of μ.

3.4 Asymptotic Null Distributions

Generally, the underlying distributions are unknown. In this case it is usually not possible to compute the power function of a test for fixed n. (Exceptions are so called "distribution-free" tests in nonparametric statistics.) The only way out of this difficulty is to rely on large sample asymptotics and corresponding asymptotic distributions, which allow to approximate the power function and to study the **asymptotic efficiency** of a test. The finite sample behavior of a test for different sample sizes n is then evaluated by means of **simulation studies**.

For a real-valued parameter θ most tests of $H_0 : \theta = \theta_0$ rely on estimators $\hat{\theta}$ of θ. Under suitable regularity conditions on the underlying distribution, central limit theorems usually imply that

$$\sqrt{n}(\hat{\theta} - \theta) \to_D N(0, v^2) \quad \text{as} \quad n \to \infty,$$

where v^2 is the asymptotic variance of the estimator.

Often a consistent estimator $\hat{v}^2$ of v^2 can be determined from the data. For large n we then approximately have

$$\frac{\sqrt{n}(\hat{\theta} - \theta)}{v} \overset{a}{\sim} N(0, 1).$$

For a given α, a one-sided test of $H_0 : \theta = \theta_0$ against $H_1 : \theta > \theta_0$ then rejects H_0 if

$$Z = \frac{\sqrt{n}(\hat{\theta} - \theta_0)}{v} > z_{1-\alpha}.$$

The corresponding asymptotic approximation (valid for sufficiently large n) of the true power function is then given by

$$\beta_{n,\alpha}(\theta) = 1 - \Phi\left(z_{1-\alpha} - \frac{\sqrt{n}(\theta - \theta_0)}{v}\right)$$

Note that in practice the (unknown) true value v^2 is generally replaced by an estimator $\hat{v}^2$ determined from the data. As long as $\hat{v}^2$ is a consistent estimator of v^2 this leads to the same asymptotic power function. The resulting test is asymptotically unbiased and consistent.

Usually there are many different possible estimators for a parameter θ. Consider an alternative estimator $\tilde{\theta}$ of θ satisfying

$$\sqrt{n}(\tilde{\theta} - \theta) \to_D N(0, \tilde{v}^2) \quad \text{as} \quad n \to \infty.$$

If the asymptotic variance v^2 of the estimator $\hat{\theta}$ is smaller than the asymptotic variance $\tilde{v}^2$ of $\tilde{\theta}$, i.e., $v^2 < \tilde{v}^2$, then $\hat{\theta}$ is a **more efficient** estimator of θ. Then necessarily the test based on $\hat{\theta}$ is **more powerful** than the test based on $\tilde{\theta}$, since asymptotically for all $\theta > \theta_0$

$$\begin{aligned}\tilde{\beta}_{n,\alpha}(\theta) &= 1 - \Phi\left(z_{1-\alpha} - \frac{\sqrt{n}(\theta - \theta_0)}{\tilde{v}}\right) \\ &< 1 - \Phi\left(z_{1-\alpha} - \frac{\sqrt{n}(\theta - \theta_0)}{v}\right) = \beta_{n,\alpha}(\theta)\end{aligned}$$

Example: Let $X_1, \ldots, X_n$ be an iid random sample. Consider testing $H_0 : \mu = \mu_0$ against $H_1 : \mu > \mu_0$, where $\mu := E(X_i)$. For a given level α the t-test then rejects H_0 if

$$T = \frac{\sqrt{n}(\bar{X} - \mu_0)}{S} > t_{n-1;1-\alpha},$$

where $t_{n-1;1-\alpha}$ is the $1-\alpha$ quantile of a t-distributions with $n-1$-degrees of freedom. This is an exact test if the distribution of X_i is normal. In the general case, the justification of the t-test is based on asymptotic arguments. Under some regularity conditions the central limit theorem implies that

$$\sqrt{n}(\bar{X} - \mu) \to_D N(0, \sigma^2) \quad \text{as} \quad n \to \infty$$

with $\sigma^2 = Var(X_i)$. Moreover, S^2 is a consistent estimator of σ^2 and $t_{n-1;1-\alpha} \to z_{1-\alpha}$ as $n \to \infty$. Thus even if the distribution of X_i is non-normal, for sufficiently large n, $T = \frac{\sqrt{n}(\bar{X}-\mu_0)}{S}$ is approximately $N(0,1)$-distributed and the asymptotic power function of the t-test is given by

$$\beta_{n,\alpha}(\theta) = 1 - \Phi\left(z_{1-\alpha} - \frac{\sqrt{n}(\mu - \mu_0)}{\sigma}\right).$$

3.5 Multiple Comparisons

In statistics, the multiple comparisons, multiplicity or multiple testing problem occurs when one considers a set of statistical inferences simultaneously or infers a subset of parameters selected based on the observed values. Errors in inference, including confidence intervals that fail to include their corresponding population parameters or hypothesis tests that incorrectly reject the null hypothesis are more likely to occur when one considers the set as a whole.

In empirical studies often dozens or even hundreds of tests are performed for the same data set. When **searching** for significant test results, one may come up with **false discoveries**.

Example: m different, independent test of significance level $\alpha > 0$. (Independence means that the test statistics used are mutually independent – this is usually not true in practice). Let's assume that a common null hypothesis H_0 holds for each of the m tests. Then

$$P\begin{pmatrix}\text{Type I error}\\ \text{by at least}\\ \text{one of the } m \text{ tests}\end{pmatrix} = 1-(1-\alpha)^m =: \alpha_m > \alpha$$

Therefore, as m increases also the probability of a type I error increases:

Number of tests m	Probability of at least one type I error (α_m)
1	0.050
3	0.143
5	0.226
10	0.401
100	**0.994**

Analogous problem: Construction of m many $(1-\alpha)$ **confidence intervals**.

$$P\begin{pmatrix}\text{at least one of the } m \text{ confidence}\\ \text{intervals does not contain}\\ \text{the true parameter value}\end{pmatrix} = 1-(1-\alpha)^m > \alpha$$

This represents the general problem of multiple comparisons. In practice, it will not be true that all considered test statistics are mutually independent. (This even complicates the problem.) However, we will still have the effect that the probability of at least one falsely significant result increases with the number m of tests, but it will not be equal to $1-(1-\alpha)^m$.

A statistically rigorous **solution** of this problem consists in modifying the constructions of tests or confidence intervals in order to arrive at **simultaneous tests**:

$$P\begin{pmatrix}\text{Type I error by}\\ \text{at least one of the } m \text{ tests}\end{pmatrix} \leq \alpha$$

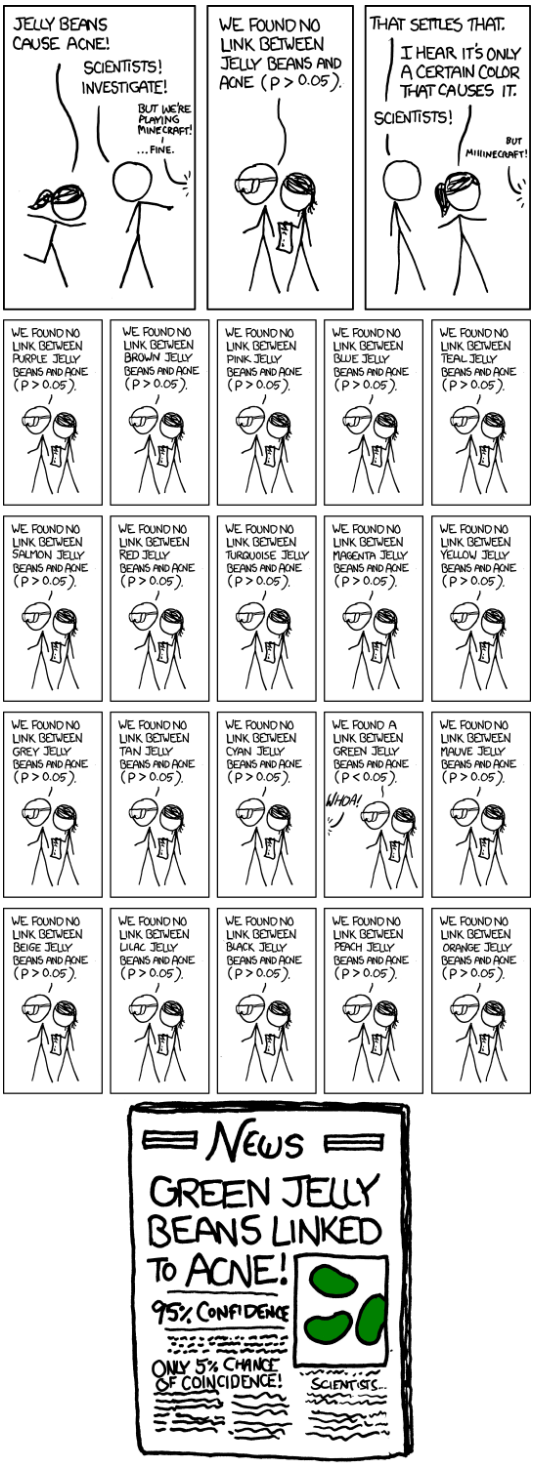


Figure 3.2: From: https://xkcd.com/882/

or **simultaneous confidence intervals**:

$$P\left(\begin{array}{c}\text{At least one of the } m \text{ confidence}\\ \text{intervals does not contain}\\ \text{the true parameter value}\end{array}\right) \leq \alpha$$

$$\Leftrightarrow \quad P\left(\begin{array}{c}\text{All confidence intervals}\\ \text{simultaneously contain the}\\ \text{true parameter values}\end{array}\right) \geq 1-\alpha$$

For certain problems (e.g., analysis of variance) there exist specific procedures for constructing simultaneous confidence intervals. However, the only generally applicable procedure seems to be the **Bonferroni correction**. It is based on Boole's inequality.

Theorem (Boole): Let $A_1, A_2, \ldots, A_m$ denote m different events. Then

$$P(A_1 \cup A_2 \cup \cdots \cup A_m) \leq \sum_{i=1}^{m} P(A_i).$$

This inequality also implies that:

$$P(A_1 \cap A_2 \cap \cdots \cap A_m) \geq 1 - \sum_{i=1}^{m} P(\bar{A}_i),$$

where $\bar{A}_i$ denotes the complementary event "not A_i".

Example: Bonferroni adjustment for m different tests of level $\alpha^* = \alpha/m$.

$$P\left(\begin{array}{c}\text{Type I error by}\\ \text{at least one of the } m \text{ tests}\end{array}\right) \leq \sum_{i=1}^{m} \alpha^* = \alpha$$

Analogously: Construction of m many $(1-\alpha^*)$-confidence intervals with $\alpha^* = \alpha/m$:

$$P\left(\begin{array}{c}\text{At least one of the } m \text{ confidence}\\ \text{intervals does not contain}\\ \text{the true parameter value}\end{array}\right) \leq \sum_{i=1}^{m} \alpha^* = \alpha$$

$$\Leftrightarrow \quad P\left(\begin{array}{c}\text{All confidence interval}\\ \text{simultaneously contain the}\\ \text{true parameter values}\end{array}\right) \geq 1 - \sum_{i=1}^{m} \alpha^* = 1-\alpha$$

Example: Regression analysis with $K = 100$ regressors, where none of the variables has an effect on the dependent variable y.

```
library("tidyverse", quietly = TRUE)
K <- 100 # number of regressors
n <- 500 # sample size

set.seed(123)

# Generate regression data, where none of the X-variables
# has an effect on the dependent variable Y:
my_df <- matrix(data = rnorm(n = n*K),
                nrow = n, ncol = K,
                dimnames = list(paste0("i.",1:n),
                                paste0("X.",1:K))) %>%
  as_tibble() %>%
  mutate(Y = rnorm(n)) %>% # Adding a Y-variable that is independent of the X-variables
  select(Y, everything())

# OLS regression
OLS_result_df <- lm(Y ~ . , data = my_df) %>%
  summary %>%
  broom::tidy()


Count_Signif <- OLS_result_df %>%
  filter(term != '(Intercept)') %>%
  count(p.value < 0.05)
```

```
## # A tibble: 2 x 2
##   `p.value < 0.05`     n
## * <lgl>            <int>
## 1 FALSE               96
## 2 TRUE                 4
```

3.6 R-Lab: The Gauss-Test

Let's reconsider the simplest test statistic you will ever meet: The **Gauss-Test** (Or "Z-Test").

Setup: Let $X_1, \dots, X_n$ be an iid random sample with $X_i \sim N(\mu, \sigma^2)$ and $\sigma^2 < \infty$.

Idea: Under the above setup, $\bar{X}_n = n^{-1} \sum_{i=1}^{n} X_i$ consistently estimates the (unknown) true mean value μ. That is, $\bar{X}_n \to_p \mu$.

- Under the null hypothesis (i.e., $\mu_0 = \mu$), the difference $\bar{X}_n - \mu_0$ should be

"small".

- Under the alternative hypothesis (i.e., $\mu_0 \neq \mu$), the difference $\bar{X}_n - \mu_0$ should be "large".

Under the null hypothesis H_0 we have that $\mu_0 = \mu$. Therefore:

$$Z = \frac{\sqrt{n}\,(\bar{X}_n - \mu_0)}{\sigma} = \underbrace{\frac{\sqrt{n}\,(\bar{X}_n - \mu)}{\sigma}}_{\sim N(0,1)}$$

Under the alternative H_1 we have that $\mu_0 \neq \mu$. Therefore:

$$\begin{aligned} Z &= \frac{\sqrt{n}\,(\bar{X}_n - \mu_0)}{\sigma} \\ &= \frac{\sqrt{n}\,(\bar{X}_n - \mu_0 + \mu - \mu)}{\sigma} \\ &= \frac{\sqrt{n}\,(\bar{X}_n - \mu)}{\sigma} + \frac{\sqrt{n}\,(\mu - \mu_0)}{\sigma} \sim N\left(\frac{\sqrt{n}\,(\mu - \mu_0)}{\sigma}, 1\right) \end{aligned}$$

The different distributions (under H_0 and H_1) of the test statistic Z can be investigated in the following dynamic plot:

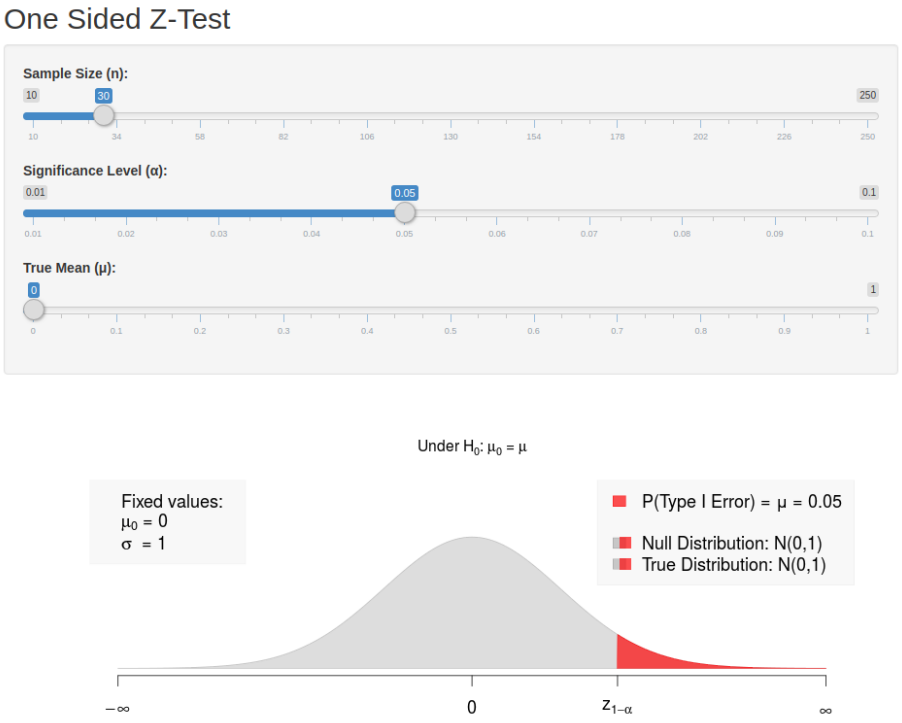


Figure 3.3: See: https://dliebl.shinyapps.io/Gauss-Test-Distr/

Chapter 4

Ordinary Least Squares: The Classical Linear Regression Model

4.1 Finite-Sample Properties

Notation:

- y_i dependent variable.
- x_{ik} kth independent variable (or regressor) with $k = 1, \dots, K$.
  Can be stochastic or deterministic.
- ε_i stochastic error term
- i indexes the ith individual with $i = 1, \dots, n$, where n is the sample size

Assumption 1.1: Linearity

$$y_i = \sum_{k=1}^{K} \beta_k x_{ik} + \varepsilon_i, \quad i = 1, \dots, n\,. \tag{4.1}$$

Usually, a constant (or intercept) is included, in this case $x_{i1} = 1$ for all i. In the following we will always assume that a constant is included in the linear model, unless otherwise stated. A special case of the above defined linear model is the so-called *simple linear model*, defined as

$$y_i = \beta_1 + \beta_2 x_i + \varepsilon_i, \quad i = 1, \dots, n\,.$$

Often it is convenient to write (4.1) using matrix notation

$$y_i = \mathbf{x}_i'\beta + \varepsilon_i, \quad i = 1, \dots, n, \tag{4.2}$$

where $\mathbf{x}_i = (x_{i1}, \dots, x_{iK})'$ and $\beta = (\beta_1, \dots, \beta_K)'$. Stacking all individual rows i leads to

$$\underset{(n\times 1)}{\mathbf{y}} = \underset{(n\times K)}{\mathbf{X}}\,\underset{(K\times 1)}{\beta} + \underset{(n\times 1)}{\varepsilon},$$

where

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} x_{11} & \dots & x_{1K} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nK} \end{pmatrix}, \quad \text{and} \quad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

We begin our analysis of the model in Eq. (4.2) under the framework of the so-called *classic assumptions*.

Assumption 1.2: Strict Exogeneity

$$\mathbb{E}(\varepsilon_i|\mathbf{X}) = 0$$

or equivalently stated for the vector ε

$$\mathbb{E}(\varepsilon|\mathbf{X}) = \mathbf{0}.$$

Notice that in the presence of a constant regressor, setting the expectation to zero is a normalization. Note that in econometrics, where we typically have to work with quasi-experimental data, strict exogeneity is a very strong assumption. It also cannot be fulfilled when the regressors include lagged dependent variables.

Some Implications of Strict Exogeneity:

- The unconditional mean of the error term is zero:

$$\mathbb{E}(\varepsilon_i) = 0 \quad (i = 1, \dots, n) \tag{4.3}$$

Proof From the *Law of Total Expectations* (i.e., $\mathbb{E}(\mathbb{E}(y|\mathbf{x})) = \mathbb{E}(y)$) it follows that

$$\mathbb{E}(\varepsilon_i) = \mathbb{E}(\mathbb{E}(\varepsilon_i|\mathbf{X})).$$

The strict exogeneity assumption then yields

$$\mathbb{E}(\mathbb{E}(\varepsilon_i|\mathbf{X})) = \mathbb{E}(0) = 0.$$

qed.

- Generally, two random variables x and y are said to be **orthogonal** if their cross moment is zero: $\mathbb{E}(xy) = 0$. Under strict exogeneity, the regressors are orthogonal to the error term for *all* observations, i.e.,

$$\mathbb{E}(x_{jk}\varepsilon_i) = 0 \quad (i, j = 1, \dots, n; k = 1, \dots, K) \tag{4.4}$$

Proof

$$\begin{aligned}\mathbb{E}(x_{jk}\varepsilon_i) &= \mathbb{E}(\mathbb{E}(x_{jk}\varepsilon_i|x_{jk})) \quad \text{(Law of Total Expect.)}\\ &= \mathbb{E}(x_{jk}\mathbb{E}(\varepsilon_i|x_{jk})) \quad \text{(Linearity of } \mathbb{E}\text{-operator)}\end{aligned}$$

Now, to show that $\mathbb{E}(x_{jk}\varepsilon_i) = 0$, we need to show that $\mathbb{E}(\varepsilon_i|x_{jk}) = 0$, which is done in the following: Since x_{jk} is an element of $\mathbf{X}$, the *Law of Iterated Expectations* (i.e., $\mathbb{E}(\mathbb{E}(y|\mathbf{x}, \mathbf{z})|\mathbf{x}) = \mathbb{E}(y|\mathbf{x})$) implies that

$$\mathbb{E}(\mathbb{E}(\varepsilon_i|\mathbf{X})|x_{jk}) = \mathbb{E}(\varepsilon_i|x_{jk}).$$

The strict exogeneity assumption yields

$$\mathbb{E}(\mathbb{E}(\varepsilon_i|\mathbf{X})|x_{jk}) = \mathbb{E}(0|x_{jk}) = 0.$$

I.e., we have that

$$\mathbb{E}(\varepsilon_i|x_{jk}) = 0,$$

which allows us to conclude that

$$\mathbb{E}(x_{jk}\varepsilon_i) = \mathbb{E}(x_{jk}\mathbb{E}(\varepsilon_i|x_{jk})) = \mathbb{E}(x_{jk}0) = 0.$$

qed

- Because the mean of the error term is zero ($\mathbb{E}(\varepsilon_i) = 0$ for all i), it follows that the orthogonality property ($\mathbb{E}(x_{jk}\varepsilon_i) = 0$, for all i, j, k) is equivalent to a zero-correlation property. I.e., that

$$Cov(\varepsilon_i, x_{jk}) = 0;\ i, j = 1, \dots, n; k = 1, \dots, K \tag{4.5}$$

Therefore, the strict exogeneity assumption implies the requirement that regressors are uncorrelated with the current ($i = j$), the past ($i < j$) and the future ($i > j$) error terms. Of course, this is usually found to be a too strong assumption - particularly in time-series contexts.

Proof

$$\begin{aligned} Cov(\varepsilon_i, x_{jk}) &= \mathbb{E}(x_{jk}\varepsilon_i) - \mathbb{E}(x_{jk})\,\mathbb{E}(\varepsilon_i) \quad \text{(Def. of Cov)} \\ &= \mathbb{E}(x_{jk}\varepsilon_i) \\ &= 0 \end{aligned}$$

Where the second equal sign holds since $\mathbb{E}(\varepsilon_i) = 0$ (see Eq. (4.3)) and the third because of orthogonality (see Eq. (4.4)). qed

Assumption 1.3: Rank Condition

$$rank(\mathbf{X}) = K \quad \text{a.s.}$$

This assumption demands that the event of one regressor being linearly dependent on the others occurs with a probability equal to zero. (This is the literal translation of the "almost surely (a.s.)" concept.) This assumption also implies the assumption that $n \geq K$.
This assumption is a bit dicey and its violation belongs to one of the classic problems in applied econometrics (keywords: multicollinearity, dummy variable trap, variance inflation). The violation of this assumption harms any economic interpretation as we cannot disentangle the regressors' individual effects on $\mathbf{y}$. Therefore, this assumption is often referred to as an *identification* assumption.

Assumption 1.4: Spherical Error

$$\begin{aligned} \mathbb{E}(\varepsilon_i^2|\mathbf{X}) &= \sigma^2 > 0 \\ \mathbb{E}(\varepsilon_i\varepsilon_j|\mathbf{X}) &= 0, \qquad i \neq j. \end{aligned}$$

Or more compactly written as,

$$\mathbb{E}(\varepsilon\varepsilon'|\mathbf{X}) = \sigma^2 I_n, \qquad \sigma^2 > 0.$$

Thus, we assume that, for a given realization of $\mathbf{X}$, the error process is uncorrelated ($\mathbb{E}(\varepsilon_i\varepsilon_j|\mathbf{X}) = 0$, for all $i \neq j$) and homoscedastic (same σ^2, for all i).

4.1.1 The Algebra of Least Squares

The OLS estimator $\mathbf{b}$ is defined as the minimizer of a specific loss function termed *the sum of squared residuals*

$$SSR(\mathbf{b}^*) = \sum_{i=1}^{n}(y_i - \mathbf{x}_i'\mathbf{b}^*)^2 \;=\; (\mathbf{y} - \mathbf{X}\mathbf{b}^*)'(\mathbf{y} - \mathbf{X}\mathbf{b}^*).$$

I.e., we have

$$\mathbf{b} := \arg\min_{\mathbf{b}^* \in \mathbb{R}^K} SSR(\mathbf{b}^*),$$

We can easily minimize $SSR(\mathbf{b}^*)$ in closed form:

$$\begin{aligned} SSR(\mathbf{b}^*) &= (\mathbf{y} - \mathbf{X}\mathbf{b}^*)'(\mathbf{y} - \mathbf{X}\mathbf{b}^*) \\ &= \mathbf{y}'\mathbf{y} - (\mathbf{X}\mathbf{b}^*)'\mathbf{y} - \mathbf{y}'\mathbf{X}\mathbf{b}^* + \mathbf{b}^{*'}\mathbf{X}'\mathbf{X}\mathbf{b}^* \\ &= \mathbf{y}'\mathbf{y} - 2\mathbf{y}'\mathbf{X}\mathbf{b}^* + \mathbf{b}^{*'}\mathbf{X}'\mathbf{X}\mathbf{b}^* \end{aligned}$$

$$\Rightarrow \quad \frac{d}{d\mathbf{b}^*} SSR(\mathbf{b}^*) = -2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\mathbf{b}^*$$

Setting the first derivative so zero yields the so-called *normal equations*

$$\mathbf{X}'\mathbf{X}\mathbf{b} = \mathbf{X}'\mathbf{y},$$

which lead to the OLS estimator

$$\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}, \tag{4.6}$$

where $(\mathbf{X}'\mathbf{X})^{-1}$ exists (a.s.) because of our full rank assumption (Assumption 3).
Often it is useful to express $\mathbf{b}$ (and similar other estimators) in sample moment notation:

$$\mathbf{b} = \mathbf{S}_{\mathbf{xx}}^{-1}\mathbf{s}_{\mathbf{xy}},$$

where $\mathbf{S}_{\mathbf{xx}} = n^{-1}\mathbf{X}'\mathbf{X} = n^{-1}\sum_i \mathbf{x}_i\mathbf{x}_i'$ and $\mathbf{s}_{\mathbf{xy}} = n^{-1}\mathbf{X}'\mathbf{y} = n^{-1}\sum_i \mathbf{x}_i y_i$. This notation is more convenient for developing our large sample results.

Some quantities of interest:

- The *(OLS) fitted value*: $\hat{y}_i = \mathbf{x}_i\mathbf{b}$
  In matrix notation: $\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{P}\mathbf{y}$
- The *(OLS) residual*: $\hat{\varepsilon}_i = y_i - \hat{y}_i$
  In matrix notation: $\hat{\varepsilon} = \mathbf{y} - \hat{\mathbf{y}} = (\mathbf{I}_n - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\,\mathbf{y} = \mathbf{M}\mathbf{y}$,

where $\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ is a so-called orthogonal projection matrix that projects any vector into the column space spanned by $\mathbf{X}$ and $\mathbf{M} = \mathbf{I}_n - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ is the associated orthogonal projection matrix that projects any vector into the vector space that is orthogonal to that spanned by $\mathbf{X}$. Projection matrices have some nice properties, listed in the following lemma.

Lemma 3.1.1 (Orthogonal Projection matrices) For $\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ and $\mathbf{M} = \mathbf{I}_n - \mathbf{P}$ with $\mathbf{X}$ being of full rank it holds:

- $\mathbf{P}$ and $\mathbf{M}$ are symmetric and idempotent, i.e.:
$$\mathbf{P}\mathbf{P} = \mathbf{P} \quad \text{and} \quad \mathbf{M}\mathbf{M} = \mathbf{M}.$$
- Further properties:
$$\mathbf{X}'\mathbf{P} = \mathbf{X}', \quad \mathbf{X}'\mathbf{M} = \mathbf{0}, \quad \text{and} \quad \mathbf{P}\mathbf{M} = \mathbf{0}.$$

Proofs follow directly from the definitions of $\mathbf{P}$ and $\mathbf{M}$.
Using these results we obtain the following proposition on the OLS residuals and OLS fitted values.

Proposition 3.1.2 (OLS residuals) For the OLS residuals and the OLS fitted values it holds that

$$\begin{aligned}\mathbf{X}'\hat{\varepsilon} &= \mathbf{0}, \quad \text{and}\\ \mathbf{y}'\mathbf{y} &= \hat{\mathbf{y}}'\hat{\mathbf{y}} + \hat{\varepsilon}'\hat{\varepsilon}.\end{aligned}$$

Proof The first result can be shown as following:

$$\begin{aligned}\mathbf{X}'\hat{\varepsilon} &= \mathbf{X}'\mathbf{M}\mathbf{y} \quad \text{(By Def. of } \mathbf{M}\text{)}\\ &= \mathbf{0}\mathbf{y} \quad \text{(By Lemma 3.1.1 part (ii))}\\ &= \underset{(K\times 1)}{\mathbf{0}}\end{aligned}$$

The second result follows from:

$$\begin{aligned}\mathbf{y}'\mathbf{y} &= (\mathbf{P}\mathbf{y} + \mathbf{M}\mathbf{y})'(\mathbf{P}\mathbf{y} + \mathbf{M}\mathbf{y}) \quad \text{(By Def. of } \mathbf{P} \text{ and } \mathbf{M}\text{)}\\ &= (\mathbf{y}'\mathbf{P}' + \mathbf{y}'\mathbf{M}')(\mathbf{P}\mathbf{y} + \mathbf{M}\mathbf{y})\\ &= \mathbf{y}'\mathbf{P}'\mathbf{P}\mathbf{y} + \mathbf{y}'\mathbf{M}'\mathbf{M}\mathbf{y} + \mathbf{0} \quad \text{(By Lemma 3.1.1 part (ii))}\\ &= \hat{\mathbf{y}}'\hat{\mathbf{y}} + \hat{\varepsilon}'\hat{\varepsilon}\end{aligned}$$

qed

The vector of residuals $\hat{\varepsilon}$ has only $n-K$ so-called *degrees of freedom*. The vector looses K degrees of freedom, since it has to satisfy the K linear restrictions ($\mathbf{X}'\hat{\varepsilon} = \mathbf{0}$). Particularly, in the case with intercept we have that $\sum_{i=1}^{n} \hat{\varepsilon_i} = \mathbf{0}$. This loss of K degrees of freedom also appears in the definition of the *unbiased* variance estimator

$$s^2 = \frac{1}{n-K} \sum_{i=1}^{n} \hat{\varepsilon_i^2}. \tag{4.7}$$

4.1.2 Coefficient of determination

The total sample variance of the dependent variable $\sum_{i=1}^{n} (y_i - \bar{y})^2$, where $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$, can be decomposed as following:

Proposition 3.1.3 (Variance decomposition) For the OLS regression of the linear model ((4.1) with intercept it holds that

$$\underbrace{\sum_{i=1}^{n} (y_i - \bar{y})^2}_{\text{total variance}} = \underbrace{\sum_{i=1}^{n} \left(\hat{y}_i - \bar{\hat{y}}\right)^2}_{\text{explained variance}} + \underbrace{\sum_{i=1}^{n} \hat{\varepsilon_i^2}}_{\text{unexplained variance}}.$$

Proof

- As a consequence of Prop. 3.1.2 we have for regressions with intercept: $\sum_{i=1}^{n} \varepsilon_i \hat{=} 0$. Hence, from $y_i = \hat{y}_i + \hat{\varepsilon_i}$ it follows that

$$\frac{1}{n} \sum_{i=1}^{n} y_i = \frac{1}{n} \sum_{i=1}^{n} \hat{y}_i + \frac{1}{n} \sum_{i=1}^{n} \hat{\varepsilon_i}$$
$$\bar{y} = \bar{\hat{y}}_i + 0$$

- From Prop. 3.1.2 we know that:

$$\begin{aligned}
\mathbf{y}'\mathbf{y} &= \hat{\mathbf{y}}'\hat{\mathbf{y}} + \hat{\varepsilon}'\hat{\varepsilon} \\
\mathbf{y}'\mathbf{y} - n\bar{y}^2 &= \hat{\mathbf{y}}'\hat{\mathbf{y}} - n\bar{y}^2 + \hat{\varepsilon}'\hat{\varepsilon} \\
\mathbf{y}'\mathbf{y} - n\bar{y}^2 &= \hat{\mathbf{y}}'\hat{\mathbf{y}} - n\bar{\hat{y}}^2 + \hat{\varepsilon}'\hat{\varepsilon} \quad \text{(By our result above.)} \\
\sum_{i=1}^{n} y_i^2 - n\bar{y}^2 &= \sum_{i=1}^{n} \hat{y}_i^2 - n\bar{\hat{y}}^2 + \sum_{i=1}^{n} \hat{\varepsilon}_i^2 \\
\sum_{i=1}^{n} (y_i - \bar{y})^2 &= \sum_{i=1}^{n} (\hat{y}_i - \bar{\hat{y}})^2 + \sum_{i=1}^{n} \hat{\varepsilon}_i^2
\end{aligned}$$

qed

The larger the proportion of the explained variance, the better is the fit of the model. This motivates the definition of the so-called R^2 coefficient of determination:

$$R^2 = \frac{\sum_{i=1}^{n} \left(\widehat{y}_i - \bar{\widehat{y}}\right)^2}{\sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2} \;=\; 1 - \frac{\sum_{i=1}^{n} \widehat{u}_i^2}{\sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2}$$

Obviously, we have that $0 \leq R^2 \leq 1$. The closer R^2 lies to 1, the better is the fit of the model to the observed data. However, a high/low R^2 does not mean a validation/falsification of the estimated model. Any relation (i.e., model assumption) needs a plausible explanation from relevant economic theory.

The most often criticized disadvantage of the R^2 is that additional regressors (relevant or not) will always increase the R^2.

Proposition 3.1.4 (R^2 increase)

Let R_1^2 and R_2^2 result from

$$\begin{aligned} \mathbf{y} &= \mathbf{X}_1\mathbf{b}_{11} + \widehat{\varepsilon_1} \quad \text{and} \\ \mathbf{y} &= \mathbf{X}_1\mathbf{b}_{21} + \mathbf{X}_2\mathbf{b}_{22} + \widehat{\varepsilon_2}. \end{aligned}$$

It then holds that $R_2^2 \geq R_1^2$.

Proof Consider the sum of squared residuals,

$$S(\mathfrak{b}_{21}, \mathfrak{b}_{22}) = (\mathbf{y} - \mathbf{X}_1\mathfrak{b}_{21} + \mathbf{X}_2\mathfrak{b}_{22})'(\mathbf{y} - \mathbf{X}_1\mathfrak{b}_{21} + \mathbf{X}_2\mathfrak{b}_{22})$$

By definition, this sum is minimized by the OLS estimators $\mathbf{b}_{21}$ and $\mathbf{b}_{22}$, i.e., $S(\mathbf{b}_{21}, \mathbf{b}_{22}) \leq S(\mathfrak{b}_{21}, \mathfrak{b}_{22})$. Consequently,

$$\widehat{\varepsilon}_2'\widehat{\varepsilon}_2 = S(\mathbf{b}_{21}, \mathbf{b}_{22}) \leq S(\mathbf{b}_{11}, 0) = \widehat{\varepsilon}_1'\widehat{\varepsilon}_1$$

which implies the statement:

$$R_2^2 = 1 - \frac{\widehat{\varepsilon}_2'\widehat{\varepsilon}_2}{\sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2} \geq 1 - \frac{\widehat{\varepsilon}_1'\widehat{\varepsilon}_1}{\sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2} = R_1^2$$

qed

Because of this, the R^2 cannot be used as a criterion for model selection. Possible solutions are given by penalized criterions such as the so-called *adjusted* R^2 defined as

$$
\begin{aligned}
\overline{R}^2 &= 1 - \frac{\frac{1}{n-K}\sum_{i=1}^{n}\widehat{u}_i^2}{\frac{1}{n-1}\sum_{i=1}^{n}\left(y_i-\bar{y}\right)^2} \\
&= 1 - \frac{n-1}{n-K}\left(1-R^2\right) \\
&= 1 - \frac{n-1}{n-K} + \frac{n-1}{n-K}R^2 \quad + \frac{K-1}{n-K}R^2 - \frac{K-1}{n-K}R^2 \\
&= 1 - \frac{n-1}{n-K} + R^2 \quad + \frac{K-1}{n-K}R^2 \\
&= -\frac{K-1}{n-K} + R^2 \quad + \frac{K-1}{n-K}R^2 \\
&= R^2 - \frac{K-1}{n-K}\left(1-R^2\right) \leq R^2
\end{aligned}
$$

The adjustment is in terms of degrees of freedom.

Partitioned regression model

Already in the first edition of Econometrica (1933) Frisch and Waugh pointed to an interesting property of multivariate linear regression analysis, which was later generalized to by Lovell (1963). The so-called Frisch-Waugh-Lovell (FWL) theorem points to a property of the OLS estimation method, which allows to gain a deeper understanding of the estimation method that is useful for the interpretation of the estimated coefficients.

$$
\mathbf{y} = \mathbf{X}_1\mathbf{b}_1 + \mathbf{X}_2\mathbf{b}_2 + \widehat{\varepsilon} = (\mathbf{X}_1, \mathbf{X}_2)\begin{pmatrix}\mathbf{b}_1 \\ \mathbf{b}_2\end{pmatrix} + \widehat{\varepsilon}, \tag{4.8}
$$

where $rank(\mathbf{X}_j) = K_j$ for $j = 1, 2$.
A regression of $\mathbf{y}$ only on $\mathbf{X}_2$ (not on $\mathbf{X}_1$), which however *takes into account the effect of* $\mathbf{X}_1$, has to be done as following:

$$
\mathbf{M}_1\mathbf{y} = \mathbf{M}_1\mathbf{X}_2\widehat{\beta}_2 + \widehat{\mathbf{v}}, \tag{4.9}
$$

where $\mathbf{M}_1 = \mathbf{I}_n - \mathbf{X}_1(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'$. Note that ((4.9) is a regression model full of residuals: The dependent variables $\mathbf{M}_1\mathbf{y}$ are the residuals from regressing $\mathbf{y}$ on $\mathbf{X}_1$ and the K_2 columns in the matrix of independent variables $\mathbf{M}_1\mathbf{X}_2$ are the residuals from the regressing $\mathbf{X}_2$ column-wise on $\mathbf{X}_1$. This means that the variables $\mathbf{M}_1\mathbf{y}$ and $\mathbf{M}_1\mathbf{X}_2$ contain only those parts of $\mathbf{y}$ and $\mathbf{X}_2$, which are orthogonal to $\mathbf{X}_1$; the effect of $\mathbf{X}_1$ is "*partialled out*". By the FWL theorem we have that:

Proposition 3.1.5 (Frisch-Waugh-Lovell theorem) For the equations ((4.8)) and (4.9)) it holds that:

$$
\widehat{\beta}_2 = \mathbf{b}_2 \quad \text{and} \quad \widehat{\varepsilon} = \widehat{\mathbf{v}}.
$$

Proof The OLS estimator $\hat{\beta}_2$ is given by

$$\begin{aligned}\hat{\beta}_2 &= \left((\mathbf{M}_1\mathbf{X}_2)'(\mathbf{M}_1\mathbf{X}_2)\right)^{-1}(\mathbf{M}_1\mathbf{X}_2)'\mathbf{y} \\ &= \left(\mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2\right)^{-1}\mathbf{X}_2'\mathbf{M}_1\mathbf{y} \end{aligned} \tag{4.10}$$

In the following, we show that $\hat{\beta}_2 = \mathbf{b}_2$:
From the normal equations for $\mathbf{b}$, we have that (using the partition $X = [\mathbf{X}_1, \mathbf{X}_2]$):

$$\begin{aligned}(\mathbf{X}\mathbf{X})^{-1}\mathbf{b} &= \mathbf{X}'\mathbf{y} \\ \begin{pmatrix}\mathbf{X}_1'\mathbf{X}_1 & \mathbf{X}_1'\mathbf{X}_2 \\ \mathbf{X}_2'\mathbf{X}_1 & \mathbf{X}_2'\mathbf{X}_2\end{pmatrix}\begin{pmatrix}\mathbf{b}_1 \\ \mathbf{b}_2\end{pmatrix} &= \begin{pmatrix}\mathbf{X}_1'\mathbf{y} \\ \mathbf{X}_2'\mathbf{y}\end{pmatrix},\end{aligned}$$

which is an equation system with two equations:

$$\mathbf{X}_1'\mathbf{X}_1\mathbf{b}_1 + \mathbf{X}_1'\mathbf{X}_2\mathbf{b}_2 = \mathbf{X}_1'\mathbf{y} \tag{4.11}$$

$$\mathbf{X}_2'\mathbf{X}_1\mathbf{b}_1 + \mathbf{X}_2'\mathbf{X}_2\mathbf{b}_2 = \mathbf{X}_2'\mathbf{y} \tag{4.12}$$

From (4.11):

$$\mathbf{b}_1 = \left(\mathbf{X}_1'\mathbf{X}_1\right)^{-1}\left(\mathbf{X}_1'\mathbf{y} - \mathbf{X}_1'\mathbf{X}_2\mathbf{b}_2\right) \tag{4.13}$$

Plugging (4.13) into (4.12) yields,

$$\begin{aligned}\mathbf{X}_2'\mathbf{X}_1\left\{\left(\mathbf{X}_1'\mathbf{X}_1\right)^{-1}\left(\mathbf{X}_1'\mathbf{y} - \mathbf{X}_1'\mathbf{X}_2\mathbf{b}_2\right)\right\} + \mathbf{X}_2'\mathbf{X}_2\mathbf{b}_2 &= \mathbf{X}_2'\mathbf{y} \\ -\mathbf{X}_2'\mathbf{X}_1\left(\mathbf{X}_1'\mathbf{X}_1\right)^{-1}\mathbf{X}_1'\mathbf{X}_2\mathbf{b}_2 + \mathbf{X}_2'\mathbf{X}_2\mathbf{b}_2 &= \mathbf{X}_2'\mathbf{y} - \mathbf{X}_2'\mathbf{X}_1\left(\mathbf{X}_1'\mathbf{X}_1\right)^{-1}\mathbf{X}_1'\mathbf{y} \\ \left(\mathbf{X}_2'\mathbf{X}_2 - \mathbf{X}_2'\mathbf{X}_1\left(\mathbf{X}_1'\mathbf{X}_1\right)^{-1}\mathbf{X}_1'\mathbf{X}_2\right)\mathbf{b}_2 &= \mathbf{X}_2'\left(I - \mathbf{X}_1\left(\mathbf{X}_1'\mathbf{X}_1\right)^{-1}\mathbf{X}_1'\right)\mathbf{y} \\ \mathbf{X}_2'\left(I - \mathbf{X}_1\left(\mathbf{X}_1'\mathbf{X}_1\right)^{-1}\mathbf{X}_1'\right)\mathbf{X}_2\mathbf{b}_2 &= \mathbf{X}_2'\left(I - \mathbf{X}_1\left(\mathbf{X}_1'\mathbf{X}_1\right)^{-1}\mathbf{X}_1'\right)\mathbf{y} \\ \mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2\mathbf{b}_2 &= \mathbf{X}_2'\mathbf{M}_1\mathbf{y} \\ \Leftrightarrow \mathbf{b}_2 &= \left(\mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2\right)^{-1}\mathbf{X}_2'\mathbf{M}_1\mathbf{y}\end{aligned} \tag{4.14}$$

From (4.10) and (4.14) it follows that $\hat{\beta}_2 = \mathbf{b}_2$ as stated by the proposition.
It remains to show that $\hat{\varepsilon} = \hat{\mathbf{v}}$:
Observe that

$$\hat{\mathbf{v}} = \mathbf{M}_1\mathbf{y} - \mathbf{M}_1\mathbf{X}_2\hat{\beta}_2.$$

But, using (4.13)

$$\begin{aligned}
\hat{\varepsilon} &= \mathbf{y} - \mathbf{X}_1\mathbf{b}_1 - \mathbf{X}_2\mathbf{b}_2 \\
&= \mathbf{y} - \mathbf{X}_1(\mathbf{X}_1'\mathbf{X}_1)^{-1}(\mathbf{X}_1'\mathbf{y} - \mathbf{X}_1'\mathbf{X}_2\mathbf{b}_2) - \mathbf{X}_2\mathbf{b}_2 \\
&= \mathbf{y} - \mathbf{P}_1\mathbf{y} - (\mathbf{X}_2\mathbf{b}_2 - \mathbf{P}_1\mathbf{X}_2\mathbf{b}_2) \\
&= \mathbf{M}_1\mathbf{y} - \mathbf{M}_1\mathbf{X}_2\mathbf{b}_2 \\
&= \hat{\mathbf{v}}
\end{aligned}$$

qed

4.1.3 Finite-Sample Properties of OLS

Notice that, by contrast to (the true but unknown) parameter vector β, $\mathbf{b}$ is a stochastic quantity, since it depends on ε through $\mathbf{y}$. The stochastic difference $\mathbf{b} - \beta$ is termed the **sampling error**:

$$\begin{aligned}
\mathbf{b} - \beta &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} - \beta \\
&= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{X}\beta + \varepsilon) - \beta \quad \text{(By Assumption 1)} \\
&= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}\beta + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\varepsilon - \beta \\
&= \beta + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\varepsilon - \beta \\
&= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\varepsilon\,,
\end{aligned}$$

where the first equality holds by Eq. (4.6)

The distribution of $\mathbf{b}$ depends (among others) on the sample size n, although this is not made explicitly by our notation. In this section, we focus on the case of a fix, finite sample size n.

Theorem *The OLS estimator* $\mathbf{b}$

- *is an unbiased estimator:* $\mathbb{E}(\mathbf{b}|\mathbf{X}) = \beta$
- *has variance:* $\mathbb{V}(\mathbf{b}|\mathbf{X}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$
- *(Gauss-Markov Theorem) is efficient in the class of all linear unbiased estimators. That is, for any unbiased estimator* $\tilde{\mathbf{b}}$ *that is linear in* $\mathbf{y}$*, we have:* $\mathbb{V}(\tilde{\mathbf{b}}|\mathbf{X}) \geq \mathbb{V}(\mathbf{b}|\mathbf{X})$ *in the matrix sense.*

While part (ii) and (iii) need all of the classical Assumptions 1.1-1.4, part (i) needs only the Assumptions 1.1-1.3.

Note that, by saying: "$\mathbb{V}(\tilde{\mathbf{b}}|\mathbf{X}) \geq \mathbb{V}(\mathbf{b}|\mathbf{X})$ in the matrix sense", we mean that $\mathbb{V}(\tilde{\mathbf{b}}|\mathbf{X}) - \mathbb{V}(\mathbf{b}|\mathbf{X}) = \mathbf{D}$, where $\mathbf{D}$ is a *positive semidefinite* $K \times K$ matrix, i.e., $\mathbf{a}'\mathbf{D}\mathbf{a} \geq 0$ for any K-dimensional vector $\mathbf{a}$. Observe that this implies that $\mathbb{V}(\tilde{\mathrm{b}}_k|\mathbf{X}) \geq \mathbb{V}(\mathrm{b}_k|\mathbf{X})$ for any $k = 1, \ldots, K$.

Proof

Part (i):

$$\begin{aligned}
\mathbb{E}(\mathbf{b}|\mathbf{X}) &= \mathbb{E}\left((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}|\mathbf{X}\right) \\
&= \mathbb{E}\left((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{X}\beta + \varepsilon)|\mathbf{X}\right) \\
&= \mathbb{E}\left((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}\beta + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\varepsilon|\mathbf{X}\right) \\
&= \beta + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbb{E}(\varepsilon|\mathbf{X}) \;=\; \beta,
\end{aligned}$$

where the last step follows from the strict exogeneity assumption.

Part (ii):

$$\begin{aligned}
\mathbb{V}(\mathbf{b}|\mathbf{X}) &= \mathbb{V}(\mathbf{b} - \beta|\mathbf{X}) \quad \text{(Since } \beta \text{ is not random)} \\
&= \mathbb{V}\left((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\varepsilon|\mathbf{X}\right) \\
&= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbb{V}(\varepsilon|\mathbf{X})\,\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} \\
&= \sigma^2(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'I_n\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} \\
&= \sigma^2(\mathbf{X}'\mathbf{X})^{-1}
\end{aligned}$$

Part (iii), Gauss-Markov:
Since $\tilde{\mathbf{b}}$ is assumed to be linear in $\mathbf{y}$, we can write

$$\tilde{\mathbf{b}} = \mathbf{C}\mathbf{y},$$

where $\mathbf{C}$ is some $K \times n$ matrix, which is a function of $\mathbf{X}$ and/or nonrandom components.
Adding a $K \times n$ zero matrix $\mathbf{0}$ yields

$$\tilde{\mathbf{b}} = \Big(\mathbf{C} \overbrace{-(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'}^{=\mathbf{0}}\Big)\mathbf{y}.$$

Let now $\mathbf{D} = \mathbf{C} - (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$, then

$$\begin{aligned}\tilde{\mathbf{b}} &= \mathbf{D}\mathbf{y} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \\ \tilde{\mathbf{b}} &= \mathbf{D}(\mathbf{X}\beta + \varepsilon) + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \\ \tilde{\mathbf{b}} &= \mathbf{D}\mathbf{X}\beta + \mathbf{D}\varepsilon + \mathbf{b} \qquad (4.15)\end{aligned}$$

$$\begin{aligned}\Rightarrow \quad \mathbb{E}(\tilde{\mathbf{b}}|\mathbf{X}) &= \mathbb{E}(\mathbf{D}\mathbf{X}\beta|\mathbf{X}) + \mathbb{E}(\mathbf{D}\varepsilon|\mathbf{X}) + \mathbb{E}(\mathbf{b}|\mathbf{X}) \\ &= \mathbf{D}\mathbf{X}\beta + \mathbf{0} + \beta \qquad (4.16)\end{aligned}$$

Since $\tilde{\mathbf{b}}$ is (by assumption) unbiased, we have that $\mathbb{E}(\tilde{\mathbf{b}}|\mathbf{X}) = \beta$. The latter, together with (4.16), implies that $\mathbf{D}\mathbf{X} = \mathbf{0}$. Plugging $\mathbf{D}\mathbf{X} = \mathbf{0}$ into (4.15) yields,

$$\begin{aligned}\tilde{\mathbf{b}} &= \mathbf{D}\varepsilon + \mathbf{b} \\ \tilde{\mathbf{b}} - \beta &= \mathbf{D}\varepsilon + (\mathbf{b} - \beta) \quad \text{(Adding a zero vector } \beta - \beta\text{)} \\ \tilde{\mathbf{b}} - \beta &= \mathbf{D}\varepsilon + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\varepsilon \quad \text{(Sampling error expression)} \\ \tilde{\mathbf{b}} - \beta &= \left(\mathbf{D} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\right)\varepsilon \qquad (4.17)\end{aligned}$$

So,

$$\begin{aligned}\mathbb{V}(\tilde{\mathbf{b}}|\mathbf{X}) &= \mathbb{V}(\tilde{\mathbf{b}} - \beta|\mathbf{X}) \quad \text{(Since } \beta \text{ is not random)} \\ &= \mathbb{V}((\mathbf{D} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\varepsilon|\mathbf{X}) \\ &= (\mathbf{D} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\mathbb{V}(\varepsilon|\mathbf{X})(\mathbf{D}' + X(\mathbf{X}'\mathbf{X})^{-1}) \\ &= \sigma^2(\mathbf{D} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')I_n(\mathbf{D}' + X(\mathbf{X}'\mathbf{X})^{-1}) \\ &= \sigma^2\left(\mathbf{D}\mathbf{D}' + (\mathbf{X}'\mathbf{X})^{-1}\right) \quad \text{(using that } \mathbf{D}\mathbf{X} = \mathbf{0}\text{)} \\ &\geq \sigma^2(\mathbf{X}'\mathbf{X})^{-1} \\ &= \mathbb{V}(\mathbf{b}|\mathbf{X}) \quad \text{(Since } \mathbf{D}\mathbf{D}' \text{ is pos. semidef.)}\end{aligned}$$

where the second equality uses Eq. (4.17).

Showing that $\mathbf{D}\mathbf{D}'$ is positive definite:

$$\mathbf{a}'\mathbf{D}\mathbf{D}'\mathbf{a} = (\mathbf{D}'\mathbf{a})'(\mathbf{D}'\mathbf{a}) = \tilde{\mathbf{a}}'\tilde{\mathbf{a}} \geq 0, \qquad (4.18)$$

where $\tilde{\mathbf{a}}$ is a K dimensional column-vector.
Remember:

- $(\mathbf{A} + \mathbf{B})' = \mathbf{A}' + \mathbf{B}'$
- $(\mathbf{A}\mathbf{B})' = \mathbf{B}'\mathbf{A}'$
- $\mathbf{A}' = \mathbf{A} \Leftrightarrow \mathbf{A}$ is a symmetric matrix qed

Theorem *Unbiasedness of* s^2

Under Assumptions 1.1-1.4, we have that:

$$\mathbb{E}(s^2|\mathbf{X}) = \sigma^2,$$

and hence $\mathbb{E}(s^2) = \sigma^2$, *provided that* $n > K$ *(otherwise* s^2 *isn't well defined).*

Proof:
In the following we show that $\mathbb{E}(s^2|\mathbf{X}) = \sigma^2$, where

$$s^2 = \frac{1}{n-K}\sum_{i=1}^{n}\hat{\varepsilon}_i^{\,2} = \frac{\hat{\varepsilon}'\hat{\varepsilon}}{n-K}.$$

In fact, it will be convenient to show the following equivalent statement:

$$\mathbb{E}(\hat{\varepsilon}'\hat{\varepsilon}|\mathbf{X}) = \sigma^2(n-K).$$

Note that

$$\begin{aligned}\hat{\varepsilon}'\hat{\varepsilon} &= (\mathbf{M}\mathbf{y})'\mathbf{M}\mathbf{y}\\ &= (\mathbf{M}(\mathbf{X}\beta+\varepsilon))'\mathbf{M}(\mathbf{X}\beta+\varepsilon)\\ &= (\mathbf{M}\varepsilon)'\mathbf{M}\varepsilon\\ &= \varepsilon'\mathbf{M}\varepsilon.\end{aligned}$$

First, we show that $\mathbb{E}(\varepsilon'\mathbf{M}\varepsilon|\mathbf{X}) = \sigma^2 trace(\mathbf{M})$, second, we show that $trace(\mathbf{M}) = n-K$.
1st Part:

$$\varepsilon'\mathbf{M}\varepsilon = \sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij}\varepsilon_i\varepsilon_j \quad (\text{All } m_{ij}\text{'s are functions of } \mathbf{X})$$

$$\begin{aligned}\Rightarrow \mathbb{E}(\varepsilon'\mathbf{M}\varepsilon|\mathbf{X}) &= \sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij}\mathbb{E}(\varepsilon_i\varepsilon_j|\mathbf{X})\\ &= \sum_{i=1}^{n} m_{ii}\sigma^2 = \sigma^2 trace(\mathbf{M}).\end{aligned}$$

2nd Part:

$$\begin{aligned} trace(\mathbf{M}) &= trace(I_n - P) \\ &= trace(I_n) - trace(P) \quad \text{(By linearity of } trace(.)) \\ &= n - trace(P) \\ &= n - trace(\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}') \\ &= n - trace(\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}) \\ &= n - trace(I_K) \\ &= n - K. \end{aligned}$$

Such that

$$\mathbb{E}(\hat{\varepsilon}'\hat{\varepsilon}|\mathbf{X}) = \sigma^2(n-K). \quad \square$$

Remember (trace-trick):

- $trace(AB) = trace(BA)$ qed

4.1.4 Hypothesis Testing under Normality

Assumption 1.5: Normality

$$\varepsilon|\mathbf{X} \sim N(\mathbf{0}, \sigma^2\mathbf{I}_n)$$

Strictly, speaking, the only aspect of this assumption is that ε is normally distributed. The assumption immediately implies that

$$\begin{aligned} ((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\varepsilon)|\mathbf{X} &\sim N(\mathbf{0}, \sigma^2(\mathbf{X}'\mathbf{X})^{-1}) \\ (\mathbf{b} - \beta)|\mathbf{X} &\sim N(\mathbf{0}, \sigma^2(\mathbf{X}'\mathbf{X})^{-1}) \end{aligned}$$

which inspires our test statistics. E.g., if we would know σ^2, we have

$$z_k = \frac{\mathrm{b}_k - \bar{\beta}_k}{\left(\sigma^2\left[(\mathbf{X}'\mathbf{X})^{-1}\right]_{kk}\right)^{1/2}} \overset{H_0}{\sim} N(0,1),$$

where $\bar{\beta}_k$ is some known value specified by the null hypothesis: H_0: $\mathrm{b}_k = \bar{\beta}_k$.

Usually, we do not know the value of σ^2 and have to estimate it. Plugging in the OLS estimate s^2 leads to

$$\text{t-ratio}_k = \frac{\mathrm{b}_k - \bar{\beta}_k}{\left(s^2\left[(\mathbf{X}'\mathbf{X})^{-1}\right]_{kk}\right)^{1/2}} \sim t_{n-K},$$

where t_{n-K} is the (Student) t-distribution with $n-K$ degrees of freedom.
Of course, **confidence intervals** for the single estimators $\widehat{\beta}_k$ can also be directly derived using the normality assumption (1.5):

$$CI_{1-\alpha} = \left[\mathbf{b}_k \pm t_{1-\frac{\alpha}{2},n-K}\; s^2\sqrt{[(\mathbf{XX})^{-1}]_{kk}}\right],$$

where $CI_{1-\alpha}$ contains the true unknown β_k with probability $1-\alpha$.
Testing linear combinations of hypotheses (so-called **linear restrictions**) on $\beta_1, \dots, \beta_K$:

$$\mathrm{H}_0 : \mathbf{R}\beta = \mathbf{r},$$

where the $(\#\mathbf{r} \times K)$ dimensional matrix $\mathbf{R}$ and the vector $\mathbf{r}$ are known and specified by the hypothesis, and $\#\mathbf{r}$ is the number of elements in $\mathbf{r}$ (i.e., the number of linear equations in the nullhypothesis). To make sure that there are no redundant equations it is required that $rank(\mathbf{R}) = \#\mathbf{r}$.
Based on the normality assumption we can test the nullhypothesis using the χ^2-distributed test statistic

$$\mathrm{W} = \frac{(\mathbf{Rb}-\mathbf{r})'(\mathbf{R}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}')^{-1}(\mathbf{Rb}-\mathbf{r})}{\sigma^2} \sim \chi^2_{\#\mathbf{r}},$$

where $\chi^2_{\#\mathbf{r}}$ denotes the χ^2-distribution with $\#\mathbf{r}$ degrees of freedom. If σ^2 is unknown we have to plug-in its estimator s^2, which then changes the distribution of the test statistic:

$$\mathrm{F} = \frac{(\mathbf{Rb}-\mathbf{r})'(\mathbf{R}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}')^{-1}(\mathbf{Rb}-\mathbf{r})}{s^2\#\mathbf{r}} \sim F_{\#\mathbf{r},n-K},$$

where $F_{\#\mathbf{r},n-K}$ is the F-distribution with $\#\mathbf{r}, n-K$ degrees of freedom.

4.2 Asymptotics under the Classic Regression Model

In this section we proof that the OLS estimators $\mathbf{b}$ and s^2 applied to the classic regression model (defined by Assumptions 1.1 to 1.4) are consistent estimators as $n \to \infty$. Even better, we can show that it is possible to drop the unrealistic normality assumption (Assumption 1.5.), but still to use the usual test statistics as long as the sample size n is large. Though, before we can formally state the asymptotic properties, we first need to adjust the rank assumption (Assumption 1.3), such that the full column rank of $\mathbf{X}$ is guaranteed for the limiting case as $n \to \infty$, too. Second, we need to assume that the sample $(y_i, \mathbf{x}_i)$ is iid, which allows us to apply Kolmogorov's strong LLN and Lindeberg-Levy's CLT.

Assumption 1.3*: $\mathbb{E}(\mathbf{x}_i\mathbf{x}_i') = \Sigma_{\mathbf{xx}}$,
such that the $(K \times K)$ matrix $\Sigma_{\mathbf{xx}}$ has full rank K (i.e., is nonsingular).

Assumption 1.5*: The sample $(\mathbf{x}_i, \varepsilon_i)$, equivalently $(y_i, \mathbf{x}_i)$, is iid for all $i = 1, \ldots, n$, with existing and finite first, second, third, and fourth moments.

Note that existence and finiteness of the first two moments of $\mathbf{x}_i$ is actually already implied by Assumption 1.3*.

Under the Assumptions 1.1, 1.2, 1.3*, 1.4, and, 1.5* we can show the following results.

Proposition 3.1.8 (Consistency of $\mathbf{S}_{\mathbf{xx}}^{-1}$)

$$\left(\frac{1}{n}\mathbf{X}'\mathbf{X}\right)^{-1} = \mathbf{S}_{\mathbf{xx}}^{-1} \quad \xrightarrow{p} \quad \Sigma_{\mathbf{xx}}^{-1}$$

Proof
1st Part:
Let us define $\bar{z}_{kl}$ as

$$[\mathbf{S}_{\mathbf{xx}}]_{kl} = \frac{1}{n}\sum_{i=1}^{n} \underbrace{x_{ik}x_{il}}_{z_{i,kl}} = \bar{z}_{kl}.$$

From:

$\mathbb{E}[z_{i,kl}] = [\mathbf{S}_{\mathbf{xx}}]_{kl}$ (By Assumption 1.3*)
and
$z_{i,kl}$ is iid and has four moments (By Assumption 1.5*)

it follows by Kolmogorov's strong law of large numbers that

$$\bar{z}_{kl} \xrightarrow{a.s.} [\Sigma_{\mathbf{xx}}]_{kl}, \quad \text{for any} \quad 1 \leq k, l \leq K.$$

Consequently, $\mathbf{S}_{\mathbf{xx}} \xrightarrow{a.s.} \Sigma_{\mathbf{xx}}$ element-wise.

2nd Part:
By the Continuous Mapping Theorem we have that also

$$(\mathbf{S}_{\mathbf{xx}})^{-1} \xrightarrow{a.s.} (\Sigma_{\mathbf{xx}})^{-1}.$$

3rd Part: Almost-Sure-Convergence implies Convergence-in-Probability ($\xrightarrow{a.s.} \Rightarrow \xrightarrow{p}$); see relations among modes of convergence. qed

Proposition 3.1.9 (Consistency of b)

$$\mathbf{b} \xrightarrow{p} \beta$$

Proof**
We show the equivalent result that $\mathbf{b} - \beta \xrightarrow{p} \mathbf{0}$.
Remember:

$$\begin{aligned}\mathbf{b} - \beta &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\varepsilon \\ &= (n^{-1}\mathbf{X}'\mathbf{X})^{-1}\frac{1}{n}\mathbf{X}'\varepsilon \\ &= (\mathbf{S}_{\mathbf{xx}})^{-1}\ \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\varepsilon_i\end{aligned}$$

From propositions 3.1.8: $(\mathbf{S}_{\mathbf{xx}})^{-1} \xrightarrow{p} (\Sigma_{\mathbf{xx}})^{-1}$.
Let us focus on element-by-element asymptotics of $\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\varepsilon_i$:
Define

$$\frac{1}{n}\sum_{i=1}^{n}\underbrace{x_{ik}\varepsilon_i}_{z_{ik}} = \bar{z}_{n,k}.$$

From:

$\mathbb{E}[z_{ik}] = \mathbb{E}[x_{ik}\varepsilon_i] = 0$ (By Str. Exog. Ass 1.2)
and
z_{ik} is iid and has four moments (By Assumption 1.5*)

it follows by Kolmogorov's strong law of large numbers that

$$\bar{z}_{n,k} = \frac{1}{n}\sum_{i=1}^{n}x_{ik}\varepsilon_i \xrightarrow{a.s.} 0 \quad \text{for any} \quad 1 \le k \le K.$$

Consequently, also

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\varepsilon_i \xrightarrow{a.s.} \underset{(K\times 1)}{\mathbf{0}} \quad \text{(element-wise)}.$$

Almost-Sure-Convergence implies Convergence-in-Probability ($\xrightarrow{a.s.} \Rightarrow \xrightarrow{p}$); see relations among modes of convergence:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\varepsilon_i \xrightarrow{p} \underset{(K\times 1)}{\mathbf{0}} \quad \text{(element-wise)}.$$

Final step: From

$$(\mathbf{S}_{\mathbf{xx}})^{-1} \xrightarrow{p} (\Sigma_{\mathbf{xx}})^{-1}$$

and

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\varepsilon_i \xrightarrow{p} \mathbf{0}$$

it follows by Slutsky's Theorem that

$$\mathbf{b} - \beta = (\mathbf{S_{xx}})^{-1} \ \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \varepsilon_i \xrightarrow{p} \mathbf{0}.$$

qed

Finally, we can show that the appropriately scaled (by $\sqrt{n}$) sampling error $\mathbf{b} - \beta$ of the OLS estimator is asymptotically normal distributed.

Proposition 3.1.10 (Sampling error limiting normality)

$$\sqrt{n}(\mathbf{b} - \beta) \xrightarrow{d} N(\mathbf{0}, \sigma^2 \Sigma_{\mathbf{xx}}^{-1}).$$

In order to show Proposition 3.1.10, we will need to use the so-called Cramér Wold Device on multivariate convergence in distribution:

Cramér Wold Device: Let $\mathbf{z}_n, \mathbf{z} \in \mathbf{R}^K$, then

$$\mathbf{z}_n \xrightarrow{d} \mathbf{z} \quad \text{if and only if} \quad \lambda' \mathbf{z}_n \xrightarrow{d} \lambda' \mathbf{z}$$

for any $\lambda \in \mathbb{R}^K$.

The Cramér Wold Device is needed, since convergence in distribution element-by-element $z_{k,n} \xrightarrow{d} z_k$ for $k = 1, \ldots, K$, does not imply multivariate convergence in distribution $\mathbf{z}_n \xrightarrow{d} \mathbf{z}$.

Proof Let's start with some rearrangements:

$$\begin{aligned}
\mathbf{b} - \beta &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\varepsilon \\
&= (n^{-1}\mathbf{X}'\mathbf{X})^{-1} \frac{1}{n}\mathbf{X}'\varepsilon \\
&= (\mathbf{S_{xx}})^{-1} \ \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \varepsilon_i \\
\Leftrightarrow \sqrt{n}(\mathbf{b} - \beta) &= (\mathbf{S_{xx}})^{-1} \ \left(\sqrt{n} \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \varepsilon_i \right)
\end{aligned}$$

From Proposition 3.1.8, we already know that

$$\left(\frac{1}{n} \mathbf{X}'\mathbf{X} \right)^{-1} = \mathbf{S}_{\mathbf{xx}}^{-1} \quad \xrightarrow{p} \quad \Sigma_{\mathbf{xx}}^{-1}.$$

What happens with

$$\sqrt{n}\underbrace{\frac{1}{n}\sum_{i=1}^{n}\overbrace{\mathbf{x}_i\varepsilon_i}^{\mathbf{z}_i}}_{\bar{\mathbf{z}}_n} = \sqrt{n}\,\bar{\mathbf{z}}_n \quad ?$$

In the following we show that $\sqrt{n}\,\bar{\mathbf{z}}_n \xrightarrow{d} N(\mathbf{0}, \sigma^2\,\Sigma_{\mathbf{xx}})$ using the Cramér Wold Device:
1st Moment:

$$\mathbb{E}(\lambda'\mathbf{z}_i) = \lambda' \underbrace{\begin{pmatrix}\mathbb{E}(\mathbf{x}_{i1}\varepsilon_i)\\ \vdots \\ \mathbb{E}(\mathbf{x}_{iK}\varepsilon_i)\end{pmatrix}}_{\mathbf{0}} = \lambda'\mathbf{0} = 0,$$

(By Str. Exog. Ass 1.2)

for any $\lambda \in \mathbb{R}^K$ and for all $i = 1, 2, ...$
2nd Moment:

$$\begin{aligned}\mathbb{V}(\lambda'\mathbf{z}_i) &= \lambda'\mathbb{V}(\mathbf{z}_i)\lambda \\ &= \lambda'\mathbb{E}(\varepsilon_i\mathbf{x}_i\mathbf{x}_i')\lambda \\ &= \lambda'\mathbb{E}(\mathbb{E}(\varepsilon_i\mathbf{x}_i\mathbf{x}_i'|\mathbf{X}))\lambda \\ &= \lambda'\mathbb{E}(\mathbf{x}_i\mathbf{x}_i'\underbrace{\mathbb{E}(\varepsilon_i|\mathbf{X})}_{=\sigma^2})\lambda \\ &\qquad\qquad \text{(Ass 1.4)} \\ &= \lambda'\sigma^2\underbrace{\mathbb{E}(\mathbf{x}_i\mathbf{x}_i')}_{\Sigma_{\mathbf{xx}}}\lambda = \sigma^2\lambda'\Sigma_{\mathbf{xx}}\lambda, \\ &\qquad\qquad \text{(Ass 1.3}^*\text{)}\end{aligned}$$

for any $\lambda \in \mathbb{R}^K$ and for all $i = 1, 2, ...$
From $\mathbb{E}(\lambda'\mathbf{z}_i) = 0$, $\mathbb{V}(\lambda'\mathbf{z}_i) = \sigma^2\lambda'\Sigma_{\mathbf{xx}}\lambda$, and $\mathbf{z}_i = (\mathbf{x}_i\varepsilon_i)$ being iid (Ass 1.5*), it follows by the Lindeberg-Levy's CLT and the Cramér Wold Device that

$$\begin{aligned}&\sqrt{n}\lambda'\bar{\mathbf{z}}_n \xrightarrow{d} N(0, \sigma^2\lambda'\Sigma_{\mathbf{xx}}\lambda) \quad \text{(By Lindeberg-Levy's CLT)} \\ \Leftrightarrow \quad &\underbrace{\sqrt{n}\bar{\mathbf{z}}_n}_{=\sqrt{n}\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\varepsilon_i} \xrightarrow{d} N(\mathbf{0}, \sigma^2\Sigma_{\mathbf{xx}}) \quad \text{(Cramér Wold Device)}\end{aligned}$$

Now, we can conclude the proof:
From $\mathbf{S}_{\mathbf{xx}}^{-1} \xrightarrow{p} \Sigma_{\mathbf{xx}}^{-1}$ (by Proposition 3.1.8 and
$\sqrt{n}\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\varepsilon_i \xrightarrow{d} N(\mathbf{0}, \sigma^2\Sigma_{\mathbf{xx}})$ it follows by Slutsky's Theorem that

$$\underbrace{(\mathbf{S}_{\mathbf{xx}})^{-1}\left(\sqrt{n}\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\varepsilon_i\right)}_{\sqrt{n}(\mathbf{b}-\beta)} \xrightarrow{d} N\left(\mathbf{0}, \underbrace{(\Sigma_{\mathbf{xx}}^{-1})\,(\sigma^2\Sigma_{\mathbf{xx}})\,(\Sigma_{\mathbf{xx}}^{-1})'}_{\sigma^2\Sigma_{\mathbf{xx}}^{-1}}\right)$$

qed

Chapter 5

Monte-Carlo Simulations

5.1 Checking Test Statistics

5.1.1 Simple Example: Gauss Test

First, we repeat some of the things we already know about the Gauss test statistic $Z = \frac{\sqrt{n}(\bar{X}-\mu_{X,0})}{\sigma_X}$ from Chapter.

Let us consider the simple case of the *one-sided* Gauss (or Z) test statistic under the following following setup:

- $X_1, \ldots, X_n$ i.i.d. random sample with $X_i \sim N(\mu_X, \sigma_X^2)$ and $\sigma_X^2 = 1$
- $\alpha = 0.05$ (Significance level)
- $n \in \{15, 30, 50\}$ (Different sample sizes)
- $\mu_{X,0} = 0$, i.e., $\Omega_0 = \{0\}$ and $\Omega_1 =]0, \infty[$.

Under the above setup, we know the **theoretical power function:**

$$\begin{aligned} \beta^Z_{n,\alpha}(\mu_X) &= \mathbb{P}(Z \geq z_{1-\alpha}) \\ &= 1 - \mathbb{P}(Z < z_{1-\alpha}) \\ &= 1 - \Phi_{\mu_Z,\sigma_Z^2}(z_{1-\alpha}), \end{aligned}$$

where Φ_{μ_Z,σ_Z^2} denotes the distribution function of a Gaussian distribution with mean and variance:

$$\begin{aligned} \mu_Z &= \frac{\sqrt{n}(\mu_X-\mu_{X,0})}{\sigma_X} = \sqrt{n}(\mu_X - 0) \\ \sigma_Z^2 &= 1. \end{aligned}$$

Furthermore, $z_{1-\alpha} = z_{0.95}$ is the 95% quantile of a standard normal distribution.

Computation in R: This is how you can use R in order to compute $\beta^Z_{n,\alpha}(\mu_X) = 1 - \Phi_{\mu_Z,1}(z_{1-\alpha})$:

```
Gauss.beta <- function(n,              # sample size
                       alpha=0.05,     # significance level
                       mu.X.true=0,    # The true mean of X_i
                       mu.X.null=0,    # The null-hypothesis mean of X_i
                       var.X    =1     # The assumed known var of X_i
                       ){
  ## Critical value:
  z_crit <- qnorm(1-alpha, mean=0, sd=1)
  ## Power:
  Phi <- pnorm(q    = z_crit,
               mean = sqrt(n)*(mu.X.true - mu.X.null)/sqrt(var.X),
               sd   = 1)
  power <- 1- Phi
  ## Return result:
  return(power)
}
```

For our purposes, it is convenient to vectorize the `Gauss.beta()` function with respect to its argument `mu.X.true`:

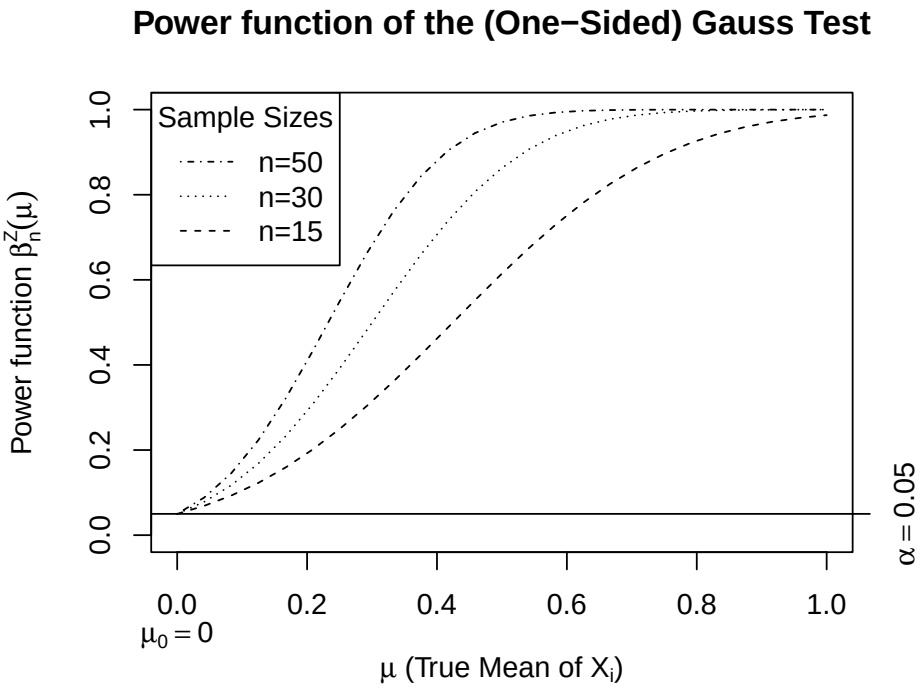
```
## Vectorization with respect to the argument `mu.X.true`:
Gauss.beta <- Vectorize(FUN=Gauss.beta, vectorize.args = "mu.X.true")
```

Plot: The function `Gauss.beta()` allows us now to easily produce a plot of the trajectories of the power function $\beta^Z_{n,0.05}(\mu_X)$ for $\mu_X \in \Omega_0 \cup \Omega_1$ and for the different sample sizes $n \in \{15, 30, 50\}$.

Here is the R-Code to do this:

```
## Sequence of different mu_X values (here from 0 to 1):
mu.X.true.seq   <- seq(0,1,len=25)

## Trajectories of the power function for different sample sizes:
## n=15
beta.n.15       <- Gauss.beta(n=15, mu.X.true=mu.X.true.seq)
## n=30
beta.n.30       <- Gauss.beta(n=30, mu.X.true=mu.X.true.seq)
## n=50
beta.n.50       <- Gauss.beta(n=50, mu.X.true=mu.X.true.seq)

## Plot
par(mar=c(5.1,4.1+1,4.1,2.1))
plot(y=0, x=0, type="n",
     ylim=c(0,1),
     xlim=range(mu.X.true.seq),
     xlab=expression(paste(mu," (True Mean of ", X[i],")")),
     ## Labels:
```

```
     ylab=expression(paste("Power function ",beta[n]^Z,(mu))),
     main="Power function of the (One-Sided) Gauss Test")
## Null-hypothesis mean:
mtext(text = expression(mu[0]==0), side = 1, line = 2, at = 0)
## Trajectories:
lines(y=beta.n.15, x=mu.X.true.seq, lty=2)
lines(y=beta.n.30, x=mu.X.true.seq, lty=3)
lines(y=beta.n.50, x=mu.X.true.seq, lty=4)
## Significance Level:
axis(4, at=0.05, labels = expression(alpha==0.05))
abline(h=0.05, lty=1)
## Legend:
legend("topleft", title = "Sample Sizes",
       legend = c("n=50","n=30","n=15"),
       lty=c(4:2))
```



5.1.2 Simulated Power Function

The power function is best suited to compare several test statistics with each other. Very often, however, it is impossible to compute the power function $\beta_{n,\alpha}(\theta)$ analytically. (The Gauss test is a rare exception.) The reason for this is that we often know only the distribution of a test statistic under the null hypothesis, but not under the alternative hypothesis. In fact, things can be

even worse: Very often, we only know the **asymptotic distribution** of a test statistic under the null hypothesis. That is, the null distribution is only known for the limiting case of $n \to \infty$.

Solution: Use **Monte-Carlo Simulations** in order approximate the power function.

For the sake of simplicity let's approximate the power function $\beta^Z_{n,\alpha}(\theta)$ of the one-sided Gauss-Test. This has the (didactic) advantage that we can compare our MC-approximated power function with the theoretical power function.

General Idea: MC-Simulations make use of the **Law of Large Numbers**. For instance, by the Strong Law of Large Numbers we know that the empirical mean

$$\bar{X}_m \to_{(a.s)} \mu_X$$

converges almost surely (a.s.) to the desired limit $\mathbb{E}(X) = \mu_X$ as $m \to \infty$. The only prerequisites are that X has finite first moments, i.e., $\mathbb{E}(X) = \mu_X < \infty$, and that $\bar{X}_m$ is constructed from an i.i.d. sample $X_1, \dots, X_m$. That is, MC-Simulations use averages in order to approximate mean values.

Approximating a Power Function (Theory):

Remember that

$$\beta^Z_{n,\alpha}(\mu_X) \quad = \quad \mathbb{P}(Z \geq z_{1-\alpha}).$$

Let us rewrite this probability using the following binary random variable:

$$V = 1_{(Z \geq z_{1-\alpha})},$$

where $1_{(\text{TRUE})} = 1$ and $1_{(\text{FALSE})} = 0$. Then we have that

$$\beta^Z_{n,\alpha}(\mu_X) \quad = \quad \mathbb{P}(Z \geq z_{1-\alpha}) \; = \; \mathbb{E}(V),$$

since

$$\mathbb{E}(V) \quad = \quad \underbrace{\mathbb{P}(V=1)}_{\mathbb{P}(Z \geq z_{1-\alpha}) \cdot 1} + \underbrace{\mathbb{P}(V=0) \cdot 0}_{=0}.$$

Now we have an expression for the power function $\beta^Z_{n,\alpha}(\mu_X)$ in terms of the population mean $\mathbb{E}(V)$.

By the **Law of Large Numbers** we know that we can use averages of i.i.d. random variables in order to approximate their population mean. That is,

$$\frac{1}{m} \sum_{j=1}^{m} V_j \to_{(a.s)} \mathbb{E}(V) \quad \text{as} \quad m \to \infty,$$

where $(V_1, \dots, V_m)$ is an iid random sample with $V_j \sim V = 1_{(Z \geq z_{1-\alpha})}$. This approximation can be made **arbitrarily accurate** as $m \to \infty$.

The MC-Simulation proceeds as following:

1. Choose a large number m, for instance, $m = 50,000$.
2. Generate realizations
$$(v_1, \dots, v_m)$$
from the MC random sample
$$(V_1, \dots, V_m) = (1_{(Z_1 \geq z_{1-\alpha})}, \dots, 1_{(Z_m \geq z_{1-\alpha})})$$
3. Approximate $\mathbb{E}(V) = \beta^Z_{n,\alpha}(\mu_X)$ using the empirical means.
$$\frac{1}{m} \sum_{j=1}^{m} v_j.$$

Approximating a Power Function (Practice):

Let's start with approximating only the following value of the power function for the one-sided Gauss-Test:

$$\beta^Z_{n=15,\alpha=0.05}(0.5) = 0.6147.$$

First, we set the Monte-Carlo sample size to $m = 50,000$.

Second, we need a realization from the random sample

$$(1_{(Z_1 \geq z_{1-\alpha})}, \dots, 1_{(Z_m \geq z_{1-\alpha})}).$$

In R you can do this as following:

```
set.seed(1009)
## Setup:
n         <- 15    # Sample Size
alpha     <-  0.05 # Significance Level
mu.X.true <-  0.5  # The (usually unknown) true mean of X_i
mu.X.null <-  0    # The null-hypothesis mean of X_i
var.X     <-  1    # The (assumed known)  var of X_i

## Critical value:
z_crit <- qnorm(1-alpha, mean=0, sd=1)

## Number of Monte-Carlo Repetitions:
m         <- 50000
```

```
## Container for Z-realizations:
Z         <- rep(NA, m)

## MC-Experiments:
for(j in 1:m){
  ## Generate X-sample:
  X.sample <- rnorm(n=n, mean=mu.X.true, sd=sqrt(var.X))
  ## Compute jth realization of Z:
  Z[j]     <- sqrt(n)*(mean(X.sample) - mu.X.null)/sqrt(var.X)
}

## Check Z>=z_crit:
head(Z >= z_crit)
```

```
## [1]  TRUE  TRUE  TRUE FALSE FALSE  TRUE
```

```
head(as.numeric(Z >= z_crit))
```

```
## [1] 1 1 1 0 0 1
```

Third, we need to compute the average

$$\frac{1}{m}\sum_{j=1}^{m} 1_{(Z_j \geq z_{1-\alpha})}$$

with respect to the simulated realizations $1_{(Z_1 \geq z_{1-\alpha})}, \ldots, 1_{(Z_m \geq z_{1-\alpha})}$.

In R you can do this as following:

```
## MC-Approximated Power:
MC_power_n15_mu0.5 <- mean(Z >= z_crit)
MC_power_n15_mu0.5
```

```
## [1] 0.6139
```

Observe that this approximation is really close to the true value: $\beta^{Z}_{n=15,\alpha=0.05}(0.5) = 0.6147 =$ `Gauss.beta(n=15, mu.X.true=0.5)`.

We can write all this as a practical R function `Gauss.MC.beta()`:

```
Gauss.MC.beta <- function(
    n         = 15,    # Sample Size
    alpha     =  0.05, # Significance Level
    mu.X.true =  0.5,  # The (usually unknown) true mean of X_i
    mu.X.null =  0,    # The null-hypothesis mean of X_i
    var.X     =  1,    # The (assumed known)  var of X_i
    ##
```

```
    m          = 50000  # Number of Monte-Carlo Repetitions:
    ){

  ## Critical value:
  z_crit <- qnorm(1-alpha, mean=0, sd=1)
  ## Container for Z-realizations:
  Z          <- rep(NA, m)

  ## MC-Experiments:
  for(j in 1:m){
    ## Generate X-sample:
    X.sample <- rnorm(n=n, mean=mu.X.true, sd=sqrt(var.X))
    ## Compute jth realization of Z:
    Z[j]     <- sqrt(n)*(mean(X.sample) - mu.X.null)/sqrt(var.X)
  }
  ## MC-Approx Power
  MC.power <- mean(c(as.numeric(Z >= z_crit)))
  ##
  return(MC.power)
}

## Vectorization:
Gauss.MC.beta <- Vectorize(FUN=Gauss.MC.beta, vectorize.args = "mu.X.true")
```

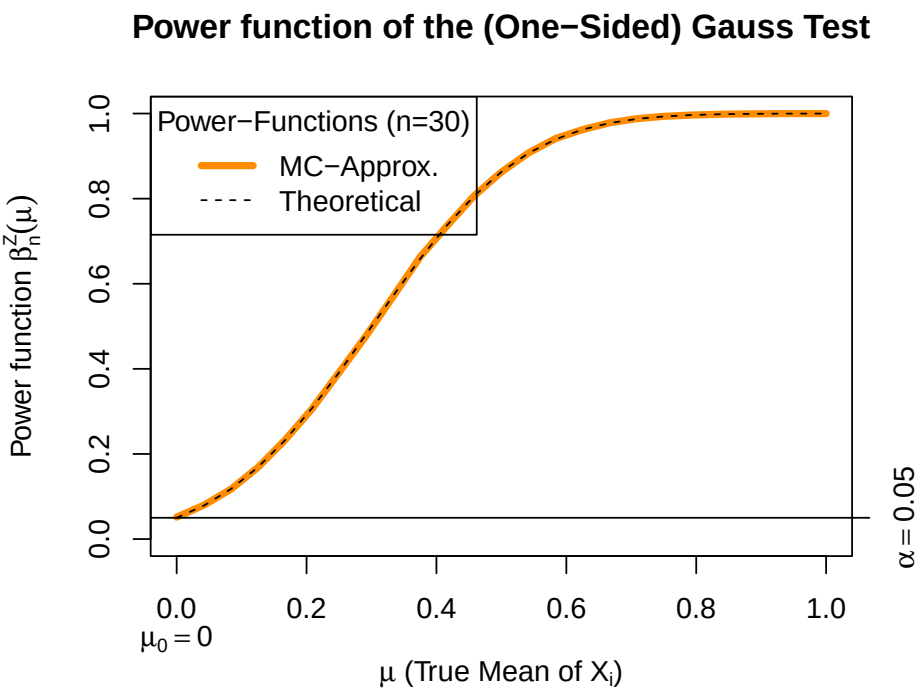
Plot:

The function `Gauss.MC.beta()` allows us now to compare the theoretical trajectory of the power function $\beta^Z_{n,0.05}(\mu_X)$ for $\mu_X \in \Omega_0 \cup \Omega_1$, e.g., for the sample size $n = 30$ with its simulated counterpart.

Here is the R-Code to do this:

```
## Sequence of different mu_X values (here from 0 to 1):
mu.X.true.seq   <- seq(0,1,len=25)

## Simulating the trajectory of the power function for n=30:
## Number of MC-Repetitions:
m <- 50000
## Try also:
## m <- 100

beta.MC.n.30       <- Gauss.MC.beta(n          = 30,
                                    mu.X.true = mu.X.true.seq,
                                    m          = m)

## Plot
par(mar=c(5.1,4.1+1,4.1,2.1))
```

```
plot(y=0, x=0, type="n",
     ylim=c(0,1),
     xlim=range(mu.X.true.seq),
     xlab=expression(paste(mu," (True Mean of ", X[i],")")),
     ## Labels:
     ylab=expression(paste("Power function ",beta[n]^Z,(mu))),
     main="Power function of the (One-Sided) Gauss Test")
## Null-hypothesis mean:
mtext(text = expression(mu[0]==0), side = 1, line = 2, at = 0)
## Trajectories:
lines(y=beta.MC.n.30, x=mu.X.true.seq, lty=1, lwd=4, col="darkorange")
lines(y=beta.n.30,    x=mu.X.true.seq, lty=2)
## Significance Level:
axis(4, at=0.05, labels = expression(alpha==0.05))
abline(h=0.05, lty=1)
## Legend:
legend("topleft", title = "Power-Functions (n=30)",
       legend = c("MC-Approx.","Theoretical"),
       lty=c(1:2), lwd=c(4,1), col=c("darkorange","black"))
```



5.2 Checking Parameter Estimators

For the following, we use our `myOLSFun()` function to compute OLS estimators.

```
myOLSFun <- function(y, x, add.intercept=FALSE){

  ## Number of Observations:
  n         <- length(y)

  ## Add an intercept to x:
  if(add.intercept){
    Intercept <- rep(1, n)
    x         <- cbind(Intercept, x)
  }

  ## Estimation of the slope-parameters:
  beta.hat.vec <- solve(t(x) %*% x) %*% t(x) %*% y

  ## Return the result:
  return(beta.hat.vec)
}
```

Let us consider the multiple regression model:

$$y_i = \beta_1 + \beta_2 x_{2i} + \beta_3 x_{3i} + \varepsilon_i, \quad i = 1, \dots, n,$$

where ε_i is a heteroscedastic error term

$$\varepsilon_i \sim N(0, \sigma_i^2), \quad \sigma_i = x_{3i},$$

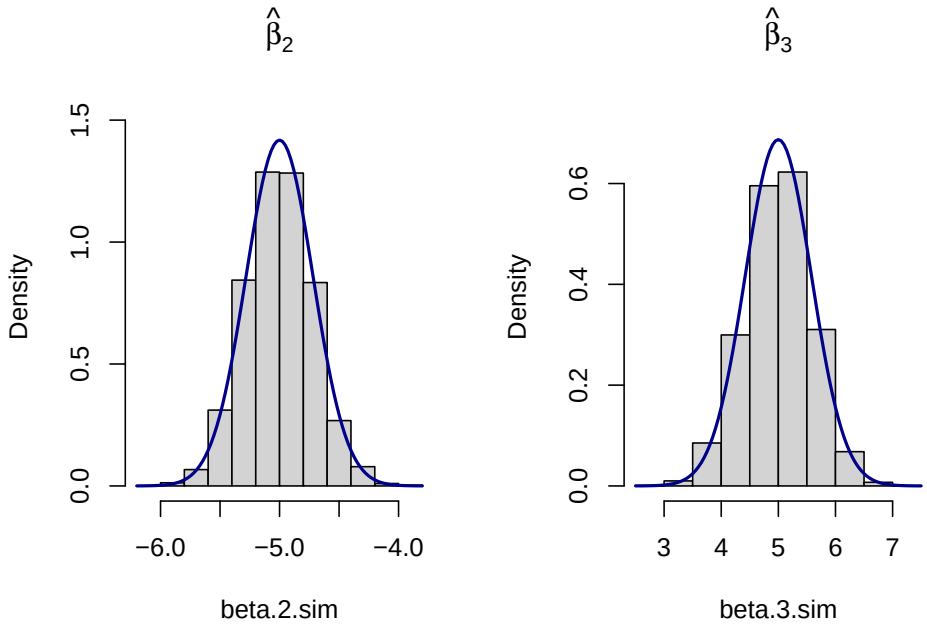
and where:

- $i = 1, \dots, n$ with $n = 50$
- $\beta_1 = 1$, $\beta_2 = -5$, and $\beta_3 = 5$
- $x_{2i} \sim N(10, 1.5^2)$
- x_{3i} comes from a t-distribution with 5 degrees of freedom and non-centrality parameter 2

This following code generates:

1. `m <- 5000` (pseudo) random samples from the above model.
2. Computes the OLS estimates $\widehat{\beta}_{2j}$ and $\widehat{\beta}_{3j}$ for each sample $j = 1, \dots, m$
3. Stores the estimation results in the data-vectors `beta.2.sim` and `beta.3.sim`
4. Plots the distribution of the estimation results using non-parametric density plots

```r
## Simulation parameters:
set.seed(109)            # Sets the "seed" of the random number generators
m           <- 5000      # Number of simulation runs
## Model parameters:
beta.vec    <- c(1,-5,5) # Slope coefficients
n           <- 50        # Number of observations
## Containers to save simulation results:
beta.2.sim  <- rep(NA,m)
beta.3.sim  <- rep(NA,m)

## Generate the regressors:
## Outside of the loop (i.e., 'conditional on X')
X.1 <- rep(1, n)
X.2 <- rnorm(n, mean=10, sd=1.5)     # Draw realizations form a normal distr.
X.3 <- rt(n, df=5, ncp=2)            # Draw realizations form a t-distr.
X   <- cbind(X.1, X.2, X.3)          # Save as a Nx3-dimensional matrix.

## Setup a progressbar
#pb <- txtProgressBar(min = 0, max = m, style = 3)

for(rpt in 1:m){
  eps <- (X.3)*rnorm(n, mean=0, sd=1) # heteroscadastic error term
  y   <- X %*% beta.vec + eps         # Dependent variable
  ## Estimation
  beta.hat <- myOLSFun(y=y,x=X)
  ## Save results
  beta.2.sim[rpt] <- beta.hat[2]
  beta.3.sim[rpt] <- beta.hat[3]
  ## Progress bar
  #setTxtProgressBar(pb, rpt)
}
#close(pb)# Close progressbar


# Theoretical variance covariance matrix of \hat{beta}
Var_beta_mat <- solve(t(X)%*%X) %*% t(X) %*% diag((X.3)^2) %*% X %*% solve(t(X)%*%X)

## Plot results
par(mfrow=c(1,2))
hist(beta.2.sim, prob=TRUE, main=expression(hat(beta)[2]), ylim=c(0,1.55))
curve(dnorm(x, mean=beta.vec[2], sd=sqrt(Var_beta_mat[2,2])),
      col="darkblue", lwd=2, add=TRUE, yaxt="n")
hist(beta.3.sim, prob=TRUE, main=expression(hat(beta)[3]), ylim=c(0,.75))
curve(dnorm(x, mean=beta.vec[3], sd=sqrt(Var_beta_mat[3,3])),
      col="darkblue", lwd=2, add=TRUE, yaxt="n")
```


$\hat{\beta}_2$
Density
0.0
0.5
1.0
1.5
−6.0
−5.0
−4.0
beta.2.sim
$\hat{\beta}_3$
Density
0.0
0.2
0.4
0.6
3
4
5
6
7
beta.3.sim

Chapter 6

How to Write

6.1 Five Common Writing Mistakes

The following is heavily based on a blog-post of Jacquelyn Gill.

1. The passive voice is being used. This is an understandable mistake, because not only are we taught to write passively in primary and secondary school science classes, but many of my colleagues still cling to the idea that solid, objective scientific writing must be in the passive voice. However, the passive voice …

- results in unnecessarily long sentences and
- makes your prose difficult to follow.

Why is the active voice better? Because science is an active process, done by human beings: "I" and "we" statements are appropriate when describing action. For instance, "We demonstrate the usefulness of our method" is much nicer to read than "The usefulness of the method is demonstrated". The active voice is more engaging to read by its very nature, which makes otherwise dry methods sections just a bit less tedious. We are a storytelling species – we like a little drama!

2. Extraneous, superfluous or otherwise unnecessary additions. *Example of bad writting:* It is entirely likely that your prose is padded with extraneous, superfluous, or otherwise unnecessary additions; furthermore, the utilization of such redundant verbiage is arguably obfuscating your points (thus, in order to improve the clarity of your writing, it is highly recommended that you eschew such stylistic choices, including run-on sentences filled with fluff, padding, and

filler).

Perhaps a consequence of assigning papers with word counts, text padding is one of the most common issues with student writing, especially for those writing their first manuscripts. My example sentence above has a few issues:

1. Double (or triple)-dipping with adjectives when one would do
2. It crams too much, abusing semi-colons and parentheses for useless purposes
3. It's full of "junk" phrases that serve no purpose whatsoever.

Tips:

- "In order to" is not necessary when "to" will do.
- "It is entirely likely that" can be replaced with a single word (likely): "Your prose is likely…"
- Words like "arguably" or "furthermore" or "thus" rarely do any heavy lifting in sentences, and are often implied anyway.
- "Highly" isn't needed in front of "likely."
- "Utilize" is rarely more appropriate than "use."

How to avoid unnecessary additions:

- Relentlessly go over your prose to **remove junk**.
- When you're over your word count on an abstract or conclusion section, look to cut sentence padding first, before you start cutting your cool ideas.
- Take opportunities to be creative, but not at the expense of clarity.

People often assume the thesaurus will help them sound smarter, but instead leaves your reader thinking, "wow, she/he really loves her/his thesaurus." Use fun words sparingly, and **aim for clarity**.

3. Your prose contains redundant parts. You keep making the same point over and over again. I often find myself reading prose that has the same idea presented in multiple ways – sometimes word for word, from one paragraph to the next! Regardless of how this happens, redundancy highlights the importance of taking a break from your work. Redundancy is also usually a symptom of poor organization; a lack of structure can lead to circular writing, because you don't know where you've come from and you don't have a clear sense of where you're going.

How to avoid redundancies:

- If you're revising your own work, you should be catching the places of redundant text parts.
- Reading your writing out loud helps (you're more likely to catch errors than if you skim the page reading to yourself).

- Get comfortable with deleting your writing. Cutting words/texts is part of the writing process, and sometimes it's the most effective way to make your writing better. That doesn't mean the initial writing was wasted – it's all part of what got you to cleaner, stronger prose.

4. Unclear antecedents. When you find yourself writing "this," check to make sure that "this" is clearly linked to an antecedent. Remember that your readers aren't in your head, and the connections may not be intuitive. In scientific writing, this problem (haha, see what I did there?) often happens in the beginning of sentences and paragraphs. "This is a problem, because…" What's the problem?

5. Your paragraph lacks of a topic sentence. Each paragraph should have a topic sentence anticipating the main message of the paragraph. Topic sentences help your reader to follow your writing. Furthermore, topic sentences are useful to diagnose structural problems in writing. The sequence of your topic sentences should represent the roadmap of your paper. Therefore, a very quick test to see if you've got organization issues is to check your topic sentences: the first sentence of your paragraph tells your reader what the paragraph is about, and every sentence should serve that topic sentence in some way.

It's worth thinking about the structure before you start writing. Otherwise, you end up taking more of a random walk than a straight line to your point, and it definitely shows. Think about organization early and often – your topic sentences can help you with it.

As with any rule, you can be a little creative here, but checking your topic sentences are a great way to check for structural issues in your writing.

6.2 Gregory Mankiw: How to Write Well

The following list of writing guidelines is taken from Gregory Mankiw's Blog:

- Stay focused. Remember the take-away points you want the reader to remember. If some material is irrelevant to these points, it should probably be cut.
- Keep sentences short. Short words are better than long words. Monosyllabic words are best.
- The passive voice is avoided by good writers.
- Positive statements are more persuasive than normative statements.

- Use adverbs sparingly.
- Avoid jargon. Any word you don't read regularly in a newspaper is suspect.
- Never make up your own acronyms.
- Avoid unnecessary words. For instance, in most cases, change
  - "in order to" to "to"
  - "whether or not" to "whether"
  - "is equal to" to "equals"
- Avoid "of course,"clearly," and "obviously." Clearly, if something is obvious, that fact will, of course, be obvious to the reader.
- The word "very" is very often very unnecessary.
- Keep your writing self-contained. Frequent references to things that have come before or will come later, can be distracting.
- Put details and digressions in footnotes. Then delete the footnotes.
- Buy a copy of Strunk and White's (Elements of Style)[https://en.wikipedia.org/wiki/The_Elements_of_Style]. Also, William Zinsser's *On Writing Well*. Read them again and again and again.
  - Zinsser's theory is that: "writing improves in direct ratio to the number of things we can keep out of it."
- Keep it simple.

6.3 Rob J Hyndman: Avoid Annoying a Referee

The following list on "How to avoid annoying a referee" is taken from Rob J Hyndman's Blog:

It's not a good idea to annoy the referees of your paper. They make recommendations to the editor about your work and it is best to keep them happy.

- Explain what you've done clearly, avoiding unnecessary jargon.
- Don't claim your paper contributes more than it actually does.
- Ensure all figures have clear (and well sized) captions and labels.
- Include citations to the referee's own work. Obviously you don't know who is going to referee your paper, but you should aim to cite the main work in the area. It places your work in context, and keeps the referees happy if they are the authors.
- Make sure the cited papers say what you think they say. Sight what you cite!
- Include proper citations for all software packages. If you are unsure how to cite an R package, try the command `citation("packagename")`.
- Never plagiarize from other papers – not even sentence fragments. Use your own words. I've refereed a thesis which had slabs taken from my own lecture notes including the typos.
- Don't plagiarize from your own papers. Either reference your earlier work, or provide a summary in new words.
- Provide enough detail so your work can be replicated. Where possible, provide the data and code. Make sure the code works.

- When responding to referee reports, make sure you answer everything asked of you. (See my earlier post "Always listen to reviewers")
- If you've revised the paper based on referees' comments, then thank them in the acknowledgements section.

6.4 LaTeX

Use LaTeX for scientific writing - particularly, if some math is involved! The probably easiest way to start with LaTeX is the online editor of overleaf.

- Overleaf allows you to start writing with LaTeX without installing software.
- Overleaf even allows for collaborative writing.

Alternatively, you can also install a LaTeX-distribtion on your computer. The following brief instructions on how to set up a LaTeX system on different operating systems is taken from Rob J Hyndman's Blog:

MS-Windows:

- Download and run the setup program for MikTeX. Choose the "basic" system.
- Download and run the installer program for TeXstudio.
- Then run TeXstudio and start typing.

Mac OS:

- Download and install MacTeX.
- Then run TeXshop and start typing.

Ubuntu:

- Install TexLive and TeXstudio through the software centre.
- Then run Texstudio and start typing.

To make sure everything is working ok, open sample.tex in TeXstudio (or TeXshop or TeXworks) to see an example of a LaTeX file. (You will also need sample.bib stored in the same folder.) Click on "Quick build" (or hit F1) and the file should be processed and appear in a separate window. Study the difference between the original file and the final product to learn some basic LaTeX commands.

For help with learning LaTeX, check out Rob J Hyndman's "Useful LaTeX links".

Chapter 7

How to Present

The following is heavily based on the paper:

Rob J Hyndman (2011), Giving a useR! Talk, The R Journal, 3(1), 69–71

Giving a talk/presentation is a balancing act in which you have to try to impart some ideas, provide sufficient background and keep the audience interested, all in a very short period of time. I've sat through more than my fair share of bad conference talks. Slides full of equations flashing past quickly, tables containing tiny figures that no-one can read, most of the audience lost on the third slide. Anyone who has attended even one conference will have seen such (bad) presentations.

The problems often stem from confusion about the purpose of the talk. Some speakers clearly think the aim of a talk is to impress the audience with their technical skills, even (or especially) if that means the audience does not understand what they are talking about. Others speakers appear to believe that talks are for those few members of the audience who are working in the same narrow research area – and so no attempt is made to provide an introduction to the topic. Still others see conference talks as an oral version of an academic paper, in all its painful detail.

7.1 The Aim of your Talk

- I like to think of talks/presentations as **advertisements for the associated paper**.
- The talk is not intended to cover everything you have done, or even to summarize what you have done.
- In giving a talk, I am hoping that:

  - everyone in the audience will have a clear idea of what I have been working on.
  - some of those in the audience will be motivated to read my paper or put into practice some of my advice.

The tiny tricky details are for the people who read the paper. Generally, there is no point discussing the tiny details of proofs, code or algorithm technicalities. Those who care will explore the details afterwards.

I tend to spend at least half the time going through a motivating example and reviewing the relevant background – most of the audience will need that context in order to understand what the talk is about. In fact, it is reasonable to assume that the audience knows about as much as you did at the start of your work in this area. That is, probably very little. So it is important to spend some time providing background information or you will lose the audience quickly. Do not assume the audience already knows what you have spent long hours learning on your own.

7.2 A Suggested Structure

1. Start with a motivating example demonstrating the problem you are trying to solve
2. Explain existing approaches to the problem and their weaknesses
3. Describe your main contributions
4. Show how your ideas solve the problem/example you started with

This structure will not necessarily work for every talk, but it is a good place to start. In particular, beginning with a motivating example is much better than setting up the problem algebraically.

For a 15 minute conference presentation, I divide the time approximately into 3/4/6/2 minute sections. Using this structure, you will have barely started on your own contributions when you are half way through your allocated time. Resist the temptation to trim the first two sections. The audience needs time to absorb the **purpose of your work** and the context in which it is set.

7.3 Preparing Slides

High quality slides are an important part of a good presentation. I recommend using the beamer package with LATEX.

Avoid distracting slide transitions and dazzling slide themes. You want the audience to focus on your content, not wonder how you implemented some gimmick. Save animation for aiding interpretation.

Use at least a **20-point font** so everyone in the room can read your material.

Limit the material on each slide

- Keep the number of words to a minimum.
- Do not include every detail of what you plan to say.
- Keep it simple.
- Resort to text only when illustrations fail you.
- Very often, a table can be replaced with a suitable graphic.
- If you must present tables, only show the essential information. No-one is going to read a slide full of tiny numbers.
- It is easy, but boring, to have bulleted lists summarizing your main points. Instead, use pictures and graphics as much as possible.
- Give only the most necessary mathematical details. When you use equations, define your notation.

Slides are not there to remind you what to say. Use a page of notes for that purpose. The slides are for the audience – make sure everything on your slides is there because it will help the audience understand what you are saying.

Slide numbers. It is useful to add slide numbers so the audience can refer back to specific slides in question time.

On your last slide, give your website or email address for people to contact you if they want to read the paper or just ask a question.

Refine the slides. I spend a lot of time going over my slides looking for ways to improve them. After a presentation is essentially complete, I go through all the slides to see what I can remove – less text is better. I also look for places where I can simplify the presentation, where I can replace text with graphics, and where the titles can be improved. I often spend almost as much time refining the slides as in creating the first version.

Always preview your slides on the computer being used for the talk. You will look foolish if symbols and Greek letters that looked OK on your computer translate into something unreadable on the big screen. Use pdf-slides!

7.4 Keeping to Time

Do not deliver a 30-minute talk in 15 minutes. Nothing irritates an audience more than a rushed presentation. It is like trying to get a drink out of a fire hydrant. Your objective is to engage the audience and have them understand your message. Do not flood them with more than they can absorb.

Cut your material. Present only as much material as can reasonably fit into the allocated time. Generally that means no more than one slide per minute. I tend to use an average of about 0.8 slides per minute of talking. It is helpful to use some slides as time-markers and make sure you are at the relevant slide at the right time.

Never go over time. Keep an eye out for signals from the session chair indicating when you need to conclude. If necessary, be prepared to cut your

talk short and finish with a quick summary.

Rehearsing is invaluable. Practise. Out loud. Standing up. Using a data projector. Get colleagues to listen to you, including some who are not knowledgeable on the topic of your talk; they will be able to point out places where you may not come across clearly. After the rehearsal, you may wish to **delete some of your material** to ensure the talk can be delivered within the allocated time.

Balance the amount of material you present with a reasonable pace of presentation. If you feel rushed when you practise, then you have too much material.

7.5 Giving the Presentation

Confidence. By the time you give the talk, you will have spent enough time preparing your slides and practising your talk that you should feel confident of giving a great presentation.

Talking. Talk at a pace that everybody in the audience can understand. Speak slowly, clearly, and loudly, especially if your English is accented. Speak loudly enough to be easily heard by those sitting in the back row

Engage the audience. Speak to them, not to the projector screen or to your notes. It helps to move around, look at your audience and smile.

Do not apologize

- Never apologize for your slides. Make apologies unnecessary by producing great slides in the first place. Do not say, "I know you can't see this, but ..." If the audience cannot read your slide, there is no point displaying it.
- Do not apologize for incomplete results either. Researchers understand that all research is incomplete. Just present the results and let the audience judge. It is okay to say, "work is on-going".

When finished, thank the audience for their attention. Stay for the entire session, for the courtesy and benefit of your audience and your co-speakers. Afterwards, be available for people to ask you questions.

Final Advice

My final advise for our last two chapters on writing and presenting:

> Everything should be made as simple as possible, but not simpler.

(Quote by Albert Einstein, or some other smart person)

Bibliography

Baltagi, B. (2008). *Econometric Analysis of Panel Data*. John Wiley & Sons.

Bertrand, M., Duflo, E., and Mullainathan, S. (2004). How much should we trust differences-in-differences estimates? *The Quarterly Journal of Economics*, 119(1):249–275.

Cameron, A. C., Gelbach, J. B., and Miller, D. L. (2011). Robust inference with multiway clustering. *Journal of Business & Economic Statistics*, 29(2):238–249.

Cerulli, G. (2015). *Econometric Evaluation of Socio-Economic Programs*. Advanced Studies in Theoretical and Applied Econometrics Series. Springer.

Davidson, R. and MacKinnon, J. G. (2004). *Econometric Theory and Methods*. Oxford University Press New York.

F. Bretz, T. Hothorn, P. W. (2010). *Multiple Comparisons Using R*. Chapman and Hall/CRC.

Fan, J. and Gijbels, I. (1996). *Local Polynomial Modelling and its Applications*, volume 66 of *Monographs on Statistics and Applied Probability*. Chapman & Hall/CRC, 1. edition.

Greene, W. (2003). *Econometric Analysis*. Pearson.

Hsiao, C. (2014). *Analysis of Panel Data*. Cambridge university press.

Imbens, G. W. and Lemieux, T. (2008). Regression discontinuity designs: A guide to practice. *Journal of Econometrics*, 142(2):615–635.

Lehmann, E. L. and Romano, J. P. (2006). *Testing statistical hypotheses*. Springer.

Li, Q. and Racine, J. (2007). *Nonparametric Econometrics: Theory and Practice*. Princeton University Press.

Little, R. J. A. (1988). A test of missing completely at random for multivariate data with missing values. *Journal of the American Statistical Association*, 83(404):1198–1202.

Wand, M. and Jones, M. (1994). *Kernel Smoothing*, volume 60. Chapman & Hall/CRC.