

Data Visualization in R & Introduction to R Markdown

Data Avicenna

Department of Economics
Universitas Gadjah Mada

May 21, 2025

Outline

- 1 Introduction
- 2 Types of Plots
- 3 Base R Plotting
- 4 Descriptive Stats
- 5 ggplot2
- 6 Plots for Publication
- 7 R Markdown
- 8 Assignment
- 9 References

Introduction to Data Visualization in R

- Data visualization is a crucial part of data analysis
- R provides multiple approaches for creating visualizations:
 - Visualising data using base R
 - Using **ggplot2** package
- Today we'll explore:
 - Base R plotting and its limitations
 - ggplot2 for more advanced, aesthetic visualizations
 - R Markdown for creating reproducible reports

Types of Plots

- **Barplot:** A barplot shows the relationship between a numeric and a categoric variable. Each entity of the categoric variable is represented as a bar. The size of the bar represents its numeric value.
- **Histogram:** A histogram takes as input a numeric variable only. The variable is cut into several bins, and the number of observation per bin is represented by the height of the bar.

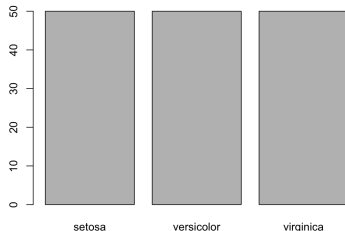


Figure: Barplot

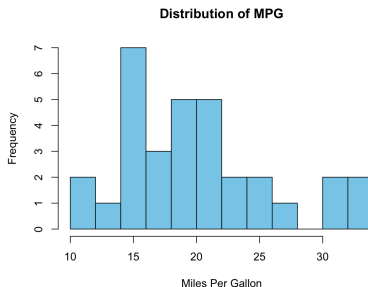


Figure: Histogram

Types of Plots

- **Line plot:** A line plot displays the evolution of one or several numeric variables. Data points are connected by straight line segments.
- **Box plot:** A boxplot gives a summary of one or several numeric variables. The line that divides the box into 2 parts represents the median of the data. The end of the box shows the upper and lower quartiles. The extreme lines show the highest and lowest value excluding outliers.

Time Series Example

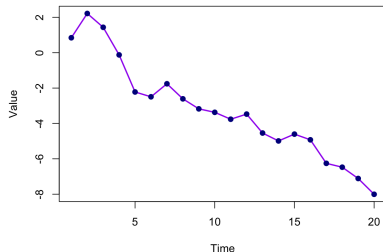


Figure: Line plot

MPG by Number of Cylinders

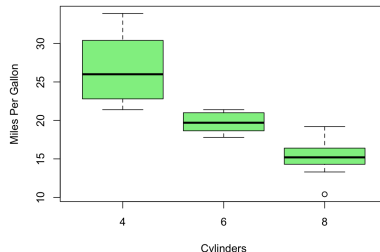


Figure: Box plot

Types of Plots

- **Scatterplot:** A scatterplot displays the relationship between 2 numeric variables. For each data point, the value of its first variable is represented on the X axis, the second on the Y axis.

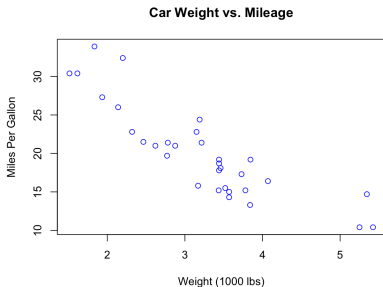


Figure: Scatterplot

Basic Plotting with Base R

Base R includes a set of functions for creating standard plots:

```
# Simple plots using plot()
plot(iris$Species) # Categorical variable
plot(iris$Petal.Length) # Quantitative variable
plot(iris$Species, iris$Petal.Width) # Cat x quant
plot(iris$Petal.Length, iris$Petal.Width) # Quant pair
plot(iris) # Scatterplot of all variables

# Simple scatter plot with options
plot(mtcars$wt, mtcars$mpg,
     main = "Car Weight vs. Mileage",
     xlab = "Weight (1000 lbs)",
     ylab = "Miles Per Gallon",
     pch = 1, # Type of plot points
     col = "blue")
```

Common Plot Types in Base R

Histogram

```
hist(mtcars$mpg,  
     breaks = 10,  
     col = "skyblue",  
     main = "Distribution of MPG",  
     xlab = "Miles Per Gallon")
```

Box plot

```
boxplot(mpg ~ cyl, data = mtcars,  
        col = "lightgreen",  
        main = "MPG by Number of Cylinders",  
        xlab = "Cylinders",  
        ylab = "Miles Per Gallon")
```


Bar Plots in Base R

```
# Create a table of frequencies
```

```
cylinders <- table(mtcars$cyl)
```

```
cylinders
```

```
# Bar plot
```

```
barplot(cylinders,  
        main = "Number of Cars by Cylinder Count",  
        xlab = "Number of Cylinders",  
        ylab = "Frequency",  
        col = "salmon")
```

Line Plots in Base R

```
# Line plots in R  
# Create some time series data  
time <- 1:20  
values <- rnorm(20) # Generates 20 random numbers  
  
# Line plot  
plot(time, values,  
      type = "o",  
      col = "purple",  
      lwd = 2,  
      main = "Time Series Example",  
      xlab = "Time",  
      ylab = "Value")  
  
# Add points to the line  
points(time, values, pch = 19, col = "darkblue")
```

Multiple Plots in Base R

```
# Create a 2x2 grid of plots
```

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

```
# Plot 1: Scatter plot
```

```
plot(mtcars$wt, mtcars$mpg, main = "Weight vs MPG", pch = 1)
```

```
# Plot 2: Histogram
```

```
hist(mtcars$mpg, main = "MPG Distribution")
```

```
# Plot 3: Box plot
```

```
boxplot(mtcars$mpg, main = "MPG Box Plot")
```

```
# Plot 4: Bar plot
```

```
barplot(table(mtcars$cyl), main = "Cylinders")
```

```
# Reset to 1x1 layout
```

```
par(mfrow = c(1, 1))
```

Descriptive Statistics


- To produce well-formatted descriptive statistics, we can use `datasummary_skim()` from the package `modelsummary`.
- In addition to common summary statistics like minimum, maximum, mean, median and standard deviation, it even comes with a small histogram of each distribution.
- The biggest advantage, however, is the option to produce nicely formatted output in a variety of formats, such as markdown.

Load Smoking Dataset

- We will be using the Smoking dataset again.
- Load dataset:
 - 1 Set your directory using `setwd()`
 - 2 Load your dataset using `read_excel()` (Or depending on your file format)
 - 3 Name your dataset **smoking_df**

modelsummary

```
# Load modelsummary package
# install.packages("modelsummary")
library(modelsummary)
datasummary_skim(smoking_df)
```

		Missing						
	Unique	Pct.	Mean	SD	Min	Median	Max	Hist
smoker	2	0	0.2	0.4	0.0	0.0	1.0	
smkban	2	0	0.6	0.5	0.0	1.0	1.0	
age	65	0	38.7	12.1	18.0	37.0	88.0	
hsdrop	2	0	0.1	0.3	0.0	0.0	1.0	
hsgrad	2	0	0.3	0.5	0.0	0.0	1.0	
colsome	2	0	0.3	0.4	0.0	0.0	1.0	
colgrad	2	0	0.2	0.4	0.0	0.0	1.0	
black	2	0	0.1	0.3	0.0	0.0	1.0	
hispanic	2	0	0.1	0.3	0.0	0.0	1.0	

Cross Tabulations

- Tabulates two variables at the same time.
- % and number of male and female that smoke and do not smoke:

		female	0	1	All
0	N		3239	1124	4363
	% row		74.2	25.8	100.0
1	N		4338	1299	5637
	% row		77.0	23.0	100.0
All	N		7577	2423	10000
	% row		75.8	24.2	100.0

- Column: 0 = non-smoker, 1 = smoker
- 25.8% of males smoke; 23% of females smoke.

Cross Tabulations Code

```
# Cross Tabulations
```

```
datasummary_crosstab(female ~ smoker, data = smoking_df)
```

```
# Double check % of male smokers
```

```
male_smokers <- filter(smoking_df, female==0) %>%
```

```
summarize(
```

```
  mean(smoker)*100
```

```
)
```

```
male_smokers # > 25.8
```


Why use ggplot2?

Base R plotting system has several limitations:

- Creating layered plots takes more steps.
- Harder to integrate with Tidyverse: If you're using dplyr, tidyr, and friends, ggplot2 just plugs right in.

A glimpse of ggplot2 vs Base R

```
library(ggplot2)
```

```
# Basic scatter plot with a linear regression line as a layer
```

```
ggplot(data = mtcars, aes(x = wt, y = mpg)) +
```

```
  geom_point(color = "blue") + # Layer 1: scatter plot
```

```
  geom_smooth(method = "lm", se = FALSE, color = "red") + # Layer 2: regression
```

```
  labs(title = "MPG vs Weight", x = "Weight", y = "Miles per Gallon") # Layer 3
```

```
# Base scatter plot
```

```
plot(mtcars$wt, mtcars$mpg,
```

```
  main = "MPG vs Weight",
```

```
  xlab = "Weight", ylab = "Miles per Gallon",
```

```
  pch = 16, col = "blue")
```

```
# Add a regression line (layer 2)
```

```
model <- lm(mpg ~ wt, data = mtcars)
```

```
abline(model, col = "red", lwd = 2)
```

```
# Add a legend (layer 3)
```

```
legend("topright", legend = c("Data", "Linear Fit"),
```

```
  col = c("blue", "red"), pch = c(16, NA), lty = c(NA, 1), lwd = c(NA, 2))
```

ggplot2

The ggplot2 code, however, looks quite different at first glance:

```
ggplot(data = mtcars,  
       mapping = aes(x = wt, y = mpg)) +  
geom_point()
```

- Two functions are necessary to create the plot.
- The plot object is being “piped” from function to function by using a special syntax (+ instead of %>%)
- The three necessary components (dataset, plot type, and aesthetic mapping) are reflected as
 - 1 argument data =
 - 2 geom_* function
 - 3 argument mapping =
 - 4 The argument mapping does not accept variables for “x” and “y” directly; you will have to wrap them inside the function aes()

ggplot2

As always in R, you can omit the argument names if you provide them in the default order (documented in the help file). Thus we can rewrite the previous code by omitting data and mapping:

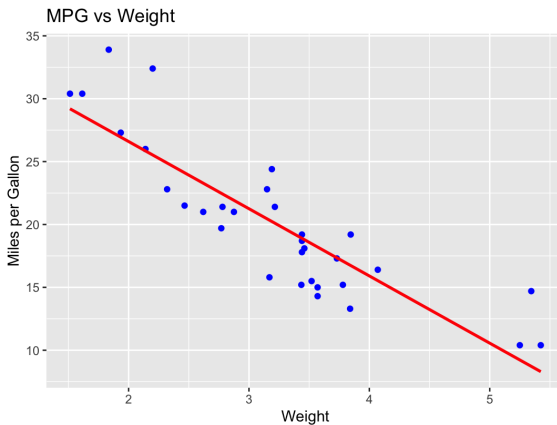
```
ggplot(mtcars, aes(x = wt, y = mpg)) +  
  geom_point()
```

Adding another layer: (Example: Linear fit)

```
ggplot(mtcars, aes(x = wt, y = mpg)) +  
  geom_point() +  
  geom_smooth(method = "lm")
```

ggplot2: Scatterplot with options and labels

```
ggplot(data = mtcars, aes(x = wt, y = mpg)) +  
  geom_point(color = "blue") + # 1:scatter plot  
  geom_smooth(method = "lm", se = FALSE, color = "red") + # 2:regression line  
  labs(title = "MPG vs Weight", x = "Weight", y = "Miles per Gallon") # 3:label
```

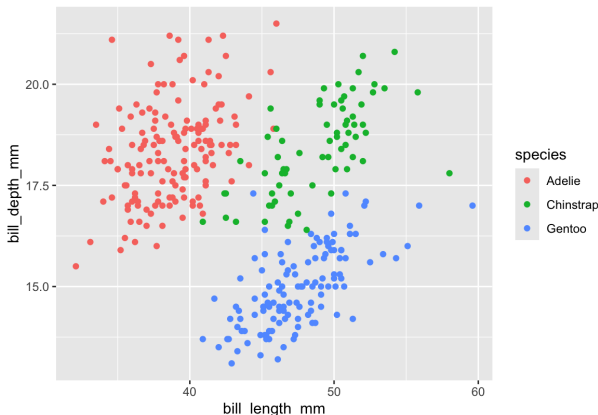


More exploration of ggplot2

- We will now use the penguins dataset from the package palmerpenguins.
- Install the palmerpenguins package:
`install.packages("palmerpenguins")`
- Load penguins dataset: `library(palmerpenguins)`

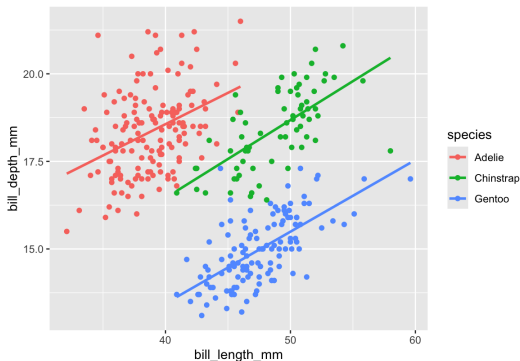
Color dots for different groups in scatterplots

```
ggplot(penguins, aes(x = bill_length_mm,  
  y = bill_depth_mm,  
  colour = species)) +  
geom_point()
```



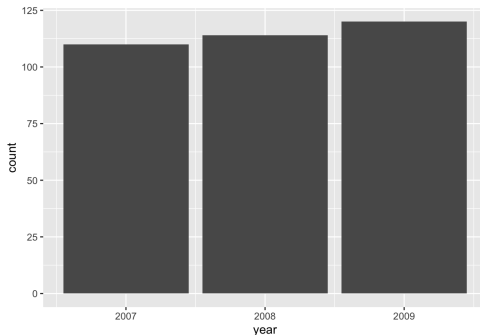
Color dots for different groups in scatterplots

```
ggplot(penguins, aes(x = bill_length_mm,  
                      y = bill_depth_mm,  
                      color = species)) +  
  geom_point() +  
  geom_smooth(method = "lm", se = FALSE)
```



Bar Charts in ggplot2

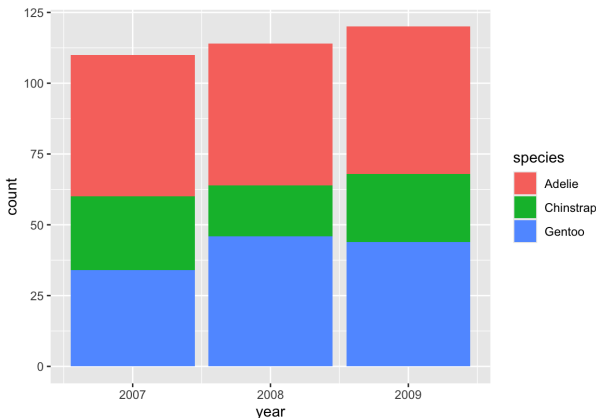
```
ggplot(penguins, aes(year)) +  
  geom_bar()
```



```
# Check number of obs for each year  
penguins %>%  
  count(year)
```

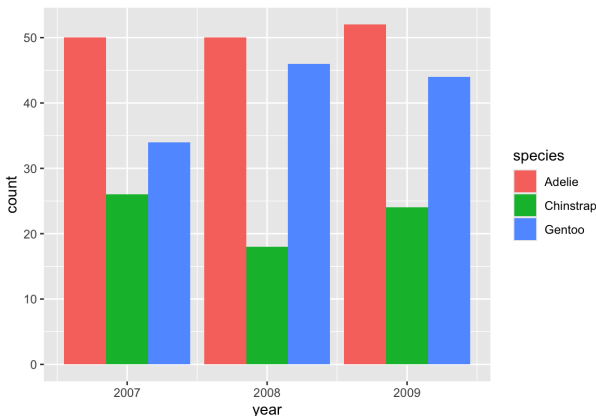
Stacked bar charts

```
ggplot(penguins, aes(year, fill = species)) +  
  geom_bar(position = "stack")
```



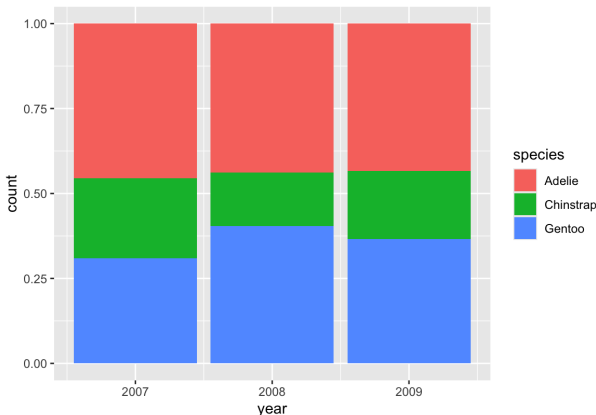
Other bar chart options

```
ggplot(penguins, aes(year, fill = species)) +  
  geom_bar(position = "dodge")
```



Other bar chart options

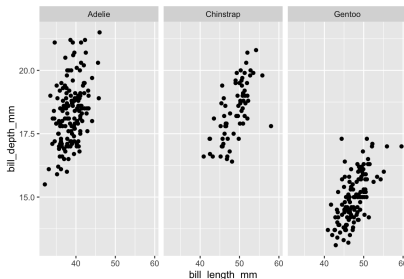
```
ggplot(penguins, aes(year, fill = species)) +  
  geom_bar(position = "fill")
```



Facet Wrap

We can visualize each group of observations separately using `facet_wrap()`.

```
# Create base plot
bill_plot <- ggplot(penguins, aes(bill_length_mm, bill_depth_mm)) +
  geom_point()
# Facet wrap
bill_plot +
  facet_wrap(~species)
```

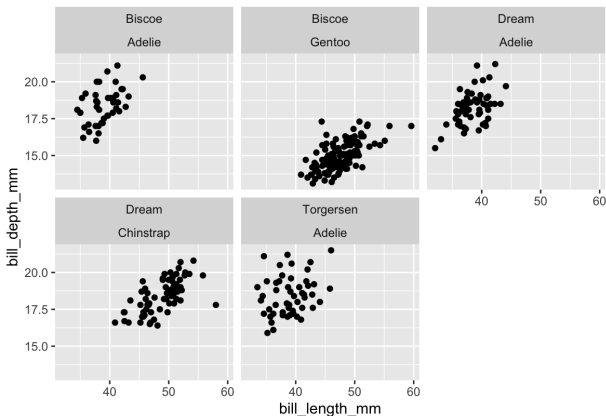


Facet Grid (Two-Sided Formula)

```
# Facet grid
```

```
bill_plot +
```

```
  facet_wrap(island~species) # two-sided formula
```



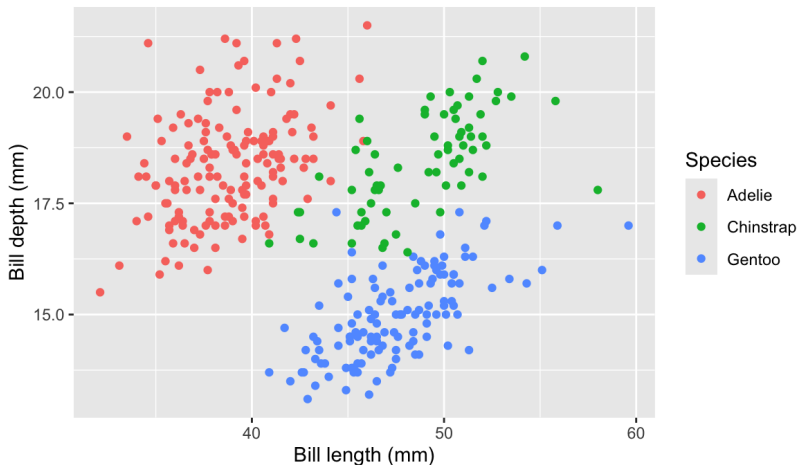
Making a Plot Publication Ready

```
# Create plot
bill_plot <- ggplot(penguins, aes(bill_length_mm, bill_depth_mm)) +
  geom_point(aes(color = species))

# Add labels
bill_plot <- bill_plot +
  labs(
    title = "Correlation between bill length and bill depth",
    subtitle = "Species seem to matter",
    y = "Bill depth (mm)",
    x = "Bill length (mm)",
    color = "Species",
    caption = "Source: palmerpenguins"
  )

# View plot
bill_plot
```

Species seem to matter



Source: palmerpenquins

Saving plots using ggsave

```
# Saving figures
```

```
# create new folder for plots
```

```
fs::dir_create("figures")
```

```
# save as pdf (vector graphic)
```

```
ggsave("figures/plot.pdf", bill_plot)
```

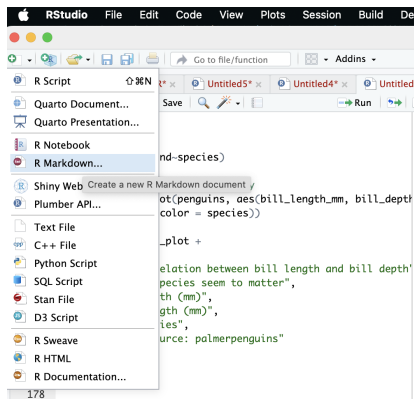
```
# save as png (raster graphic)
```

```
ggsave("figures/plot.png", bill_plot)
```

- R markdown lets you write text, add R code chunks, show the results of various plots in one document.
- You can write a great report using R markdown! The report can be saved as an HTML, PDF, or Word format.

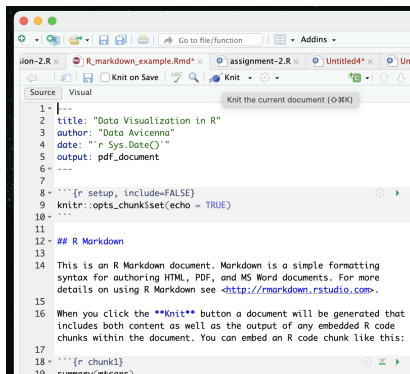
Generating an R Markdown

- In the top left corner of RStudio, click the icon that has a plus-sign on top of a white square.
- Then, you will see several options. Click "R Markdown".



Knitting an R Markdown

Knit your R Markdown to view your output. You can then save your output to your designated folder.



The screenshot shows the RStudio interface with a file explorer at the top displaying several files: 'ion-2.R', 'R_markdown_example.Rmd', 'assignment-2.R', 'Untitled4*', and 'Un...'. Below the file explorer, the 'Knit' button is highlighted. The main editor window shows the source code of an R Markdown document. The code includes a YAML header with fields for title, author, date, and output, followed by R setup code and a text block explaining R Markdown.

```
1 #---
2 title: "Data Visualization in R"
3 author: "Data Avicenna"
4 date: "r Sys.Date()"
5 output: pdf_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
11
12 ## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting
15 syntax for authoring HTML, PDF, and MS Word documents. For more
16 details on using R Markdown see <http://rmarkdown.rstudio.com>.
17
18 When you click the Knit button a document will be generated that
19 includes both content as well as the output of any embedded R code
20 chunks within the document. You can embed an R code chunk like this:
21
22 ```{r chunk1}
23 summary(mtcars)
24 ```
```

Assignment 3

You need to submit your answers to this assignment to obtain a training certificate. Submit your **HTML or PDF** of R Markdown to this link: <https://tinyurl.com/R-Assignment-3-2025>. Contact dataavicenna@mail.ugm.ac.id for any questions.

Submission deadline: **27 May 2025 at 23.59pm**

Assignment 3

Answer the following questions in R Markdown. Save your document in either an **HTML** or **PDF** format. Your very first R code chunk should be:

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
```

(Note: You can also modify my file "R_markdown_example.Rmd", where the setup code chunk at the top should not be changed)

Assignment 3

Use the penguins dataset from the "palmerpenguins" package to answer these questions in your R Markdown file:

- 1 Generate a bar graph using ggplot2 that shows the number of observations for each species in the penguins dataset. Then, create a stacked bar chart that shows the distribution of species across different islands. Make sure to include appropriate titles, labels, and a color legend. Write a brief interpretation of both visualizations.
- 2 Generate a scatterplot of bill length vs. flipper length using ggplot2. Add a fitted regression line for each species (use different colors). Include appropriate titles and axis labels. Briefly explain any patterns you observe in the relationship between these variables across different species and islands.

References

- From Data to Viz: <https://www.data-to-viz.com>
- Schmidt, S. S., & Turbanisch, F. (n.d.). *Economic Analysis with R*. University of Göttingen. Retrieved from <https://economic-analysis-with-r.uni-goettingen.de/>