Introdution to R

What is R?!

Franziska Faupel, Clara Filet & PD Dr. Oliver Nakoinz

An Introdution to R

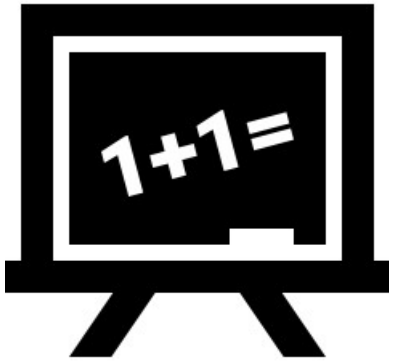
- R - What is it?!
  - *Why using R?*
  - *History of R*
  - *7 Levels of Learning*
  - *R Studio*
  - *Scripts*
- Working with R
  - *Function vs Objects*
  - *Vectors*
  - *Functions*
  - *Operators*
  - *Data Frames and Matrices*
- First Calculations and Plots in R
  - *Package Management*
  - *Loading Data*
  - *Data Management*
  - *Loops and Restrictions*
  - *Some Statistics*
  - *Digression: Tidy verse*
  - *Plotting in R*
- R and Spatial Data
  - *Load Spatial Objects*
  - *Create Spatial Objects*
  - *Manipulate Spatial Objects*
  - *Plot Spatial Objects*

R - What is it?!

- R is a language and environment for statistical computing and graphics
- Open Source Software: www.r-project.org
- extendible by packages (libraries)
- running with most operational systems
- interfaces to C, Python and Fortran etc.
- functional with object oriented extension
- vector based / dataframe
- developing and implementing own functions

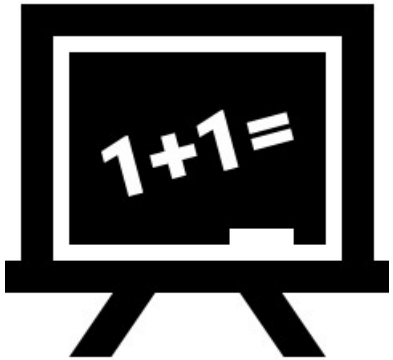
Why should I apply statistics?

- Data analysis is an essential tool of empirical research
- Highlight, explore, describe and compare patterns in Data



Why should I apply statistics?

- Data analysis is an essential tool of empirical research
- Highlight, explore, describe and compare patterns in Data



Why should I use programming to apply them?

- “Learning programming is learning to think”
- Forcing the problem in a logical and sequential structure
- Avoid the black box effect of press button softwares

Why using R?

- computation engine
  - *programming language with focus on data analysis*
  - *does all calculations*
  - *speeds up data management*
  - *visualises results*
  - *exports images in printable quality*
- packages with statistical functions (> 12’000)
  - *most statistical analysis are available*
  - *huge user community with great support*
  - *well documented applications and many free accessible books*
- user can become developer in seconds
  - *customizable with self written functions*
  - *automation*
- reproducible analysis and research
  - *scripts document the analysis*
  - *can be redone by a colleague*
  - *can be shared within your community*

History of R

- 1976: **S language by John Chambers** at Bell Laboratories (AT&T)

John Chambers

History of R

- 1976: **S language by John Chambers** at Bell Laboratories (AT&T)
- 1993: Free implementation of R under the name R by **Ross Ihaka and Robert Gentleman**
- 1997: R becomes **GNU project**
- 2000: **R1.0**
- 2015: **R Consortium**: R Foundation, Microsoft, R Studio, Google etc.

7 levels of using R

1. Asking a colleague to run R for you
2. Applying build in functions
3. Combining build in functions
4. Writing own algorithms
5. Writing own functions and objects
6. Developing efficient code
7. Writing own packages

Before you begin

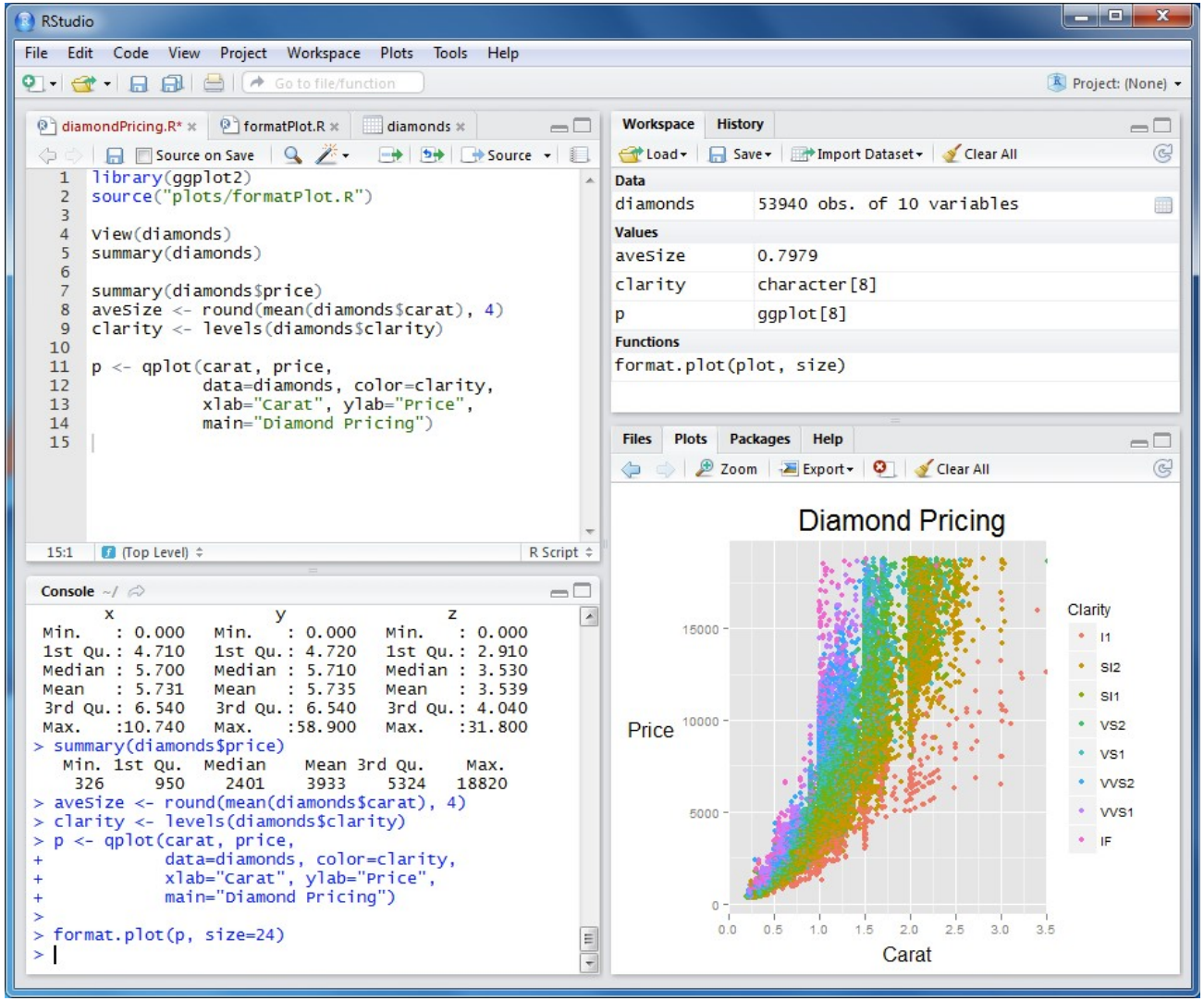
1. **Problem:** Put your question.
2. **Theory:** Which theoretical frame is adequate for solving the problem?
3. **Data:** Which data can be used for solving the problem? Are they available as database etc.? How can you access the data?
4. **Method:** Which methods can solve the problem? Learn the methods!
5. **Tool:** Learn the tools to apply the methods.
6. **Finally:** Check, validate and interpret the results. Plausible? Significant?

Now, you can begin

- **Loading** data
- **Exploring** data
- **Analysing** data
  - *Descriptive analysis*
  - *customized analyses*
- **Visualise** and export results
- **Report** and publication

How to start?

Please open R Studio!


RStudio
File Edit Code View Project Workspace Plots Tools Help
diamondPricing.R*
formatPlot.R
diamonds
Source on Save
Source
1 library(ggplot2)
2 source("plots/formatPlot.R")
3
4 view(diamonds)
5 summary(diamonds)
6
7 summary(diamonds$price)
8 aveSize <- round(mean(diamonds$carat), 4)
9 clarity <- levels(diamonds$clarity)
10
11 p <- qplot(carat, price,
12 data=diamonds, color=clarity,
13 xlab="Carat", ylab="Price",
14 main="Diamond Pricing")
15
15:1 (Top Level) R Script
Console ~/
x y z
Min. : 0.000 Min. : 0.000 Min. : 0.000
1st Qu.: 4.710 1st Qu.: 4.720 1st Qu.: 2.910
Median : 5.700 Median : 5.710 Median : 3.530
Mean : 5.731 Mean : 5.735 Mean : 3.539
3rd Qu.: 6.540 3rd Qu.: 6.540 3rd Qu.: 4.040
Max. :10.740 Max. :58.900 Max. :31.800
> summary(diamonds$price)
Min. 1st Qu. Median Mean 3rd Qu. Max.
326 950 2401 3933 5324 18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+ data=diamonds, color=clarity,
+ xlab="Carat", ylab="Price",
+ main="Diamond Pricing")
>
> format.plot(p, size=24)
>
Workspace History
Load Save Import Dataset Clear All
Data
diamonds 53940 obs. of 10 variables
Values
aveSize 0.7979
clarity character[8]
p ggplot[8]
Functions
format.plot(plot, size)
Files Plots Packages Help
Zoom Export Clear All
Diamond Pricing
Price
Carat
Clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF
15000
10000
5000
0
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5

Scipts?!

- working freehand
  - *flexible*
  - *impressive*
  - *badly documented*

- using scripts
  - *reusable*
  - *easier*
  - *exact documentation*
  - *reproducibility of the analysis*
  - *sharing code*
  - *automatic report generation*

Scripts: Some rules

- Each analysis has its own **script**.
- Each **script** has a header containing important information such as title, topic, purpose, author, date, etc.

Scripts: Script Header

```
######################################################################
## Didactic R-Script for Modelling Summer School
## ====================================================================
## Project: Modelling Summer School
## Author: O. Nakoinz, C. Filet & F. Faupel
## Version: 01
## Date of last changes: 10.09.2018
## Data: some.data
## Author of data: author.data
## Purpose: didactic
## Content: 1. preparation, 2. data import, ...
## Description: The script include ...
## Licence data: -
## Licence Script: GPL (http://www.gnu.org/licenses/gpl-3.0.html)
######################################################################
```

Scripts: Some rules

- For each **command**, you should learn the usage only once.
  - *Supplement the command with comments so that you can re-use the command for similar purposes without reading the manual again.*
  - *Copy, paste and adapt the commands from old scripts.*

```
# This is a comment
a <- c(3,6,7,4,9,7,3,3,3)       # This is a vector
median(a)                         # This is the "median" function
## [1] 4
```

- Use standardised names and denotations to ease the re-use of code.
- Use detailed **comments** to understand your analysis years later.
  - *It is useful to think that you should hand over the script to a colleague with limited knowledge to optimise comments and structure.*
- Use **versioning** of the scripts by adding a version number to the file name (-v01.xxx) or a version control system.

Scripts: Some rules

- Avoid special characters in file names, variable names, etc.
  - *subset Ä with Ae etc.*
  - *moreover, it is a good idea to encode the file in* `UTF8`.
- Short Style Guide:
  - *short, but meaningful appreciations for variables and functions*
  - *use underscores to subdivide names, like: objecttype_content*
  - *use white spaces before and after operators*
  - *use <− to assign not =*
  - *curly brackets never start in line*
- Use a convenient yet standardised folder structure for scripts, data, results and reports.

Scripts: Folder Structure

Use a convenient yet standardised folder structure for scripts, data, results and reports. During the Mosaic Summer School, we will work with the following structure.

- 1script
- 2data
- 3geodata
- 4ws
- 5figures
- 6results
- 7report
- 8lib
- etc.

Don’t Panic

- understand concepts
- adapt methods and techniques
- connect to experts
- establish a community
- develop a scientific network
- collaborative projects of participants

Summary: Why using R?

Advantages	Disadvantages
reproducibility and verifiablilty	high barriers to learn
flexible and extensible	`raw` syntax
many different packages (>12’000)	amount of available functions appear chaotic for beginners
customisable	
easier to conduct complex analysis	
figures can be created automatically	
paper can be created automatically	
independent platform with a great support	

Summary: R Studio

- Text Editor
- Console
- Data Explorer
- File Manager
- Image Viewer
- Documentation Viewer
- Project and Session Manager
- Workspace Explorer
- Package Manager
- Wizards and Assistants

Summary: Scripts

- **Scripts** are an exact documentation of the work.
- **Scripts** allow reproducing the analysis.
- **Scripts** allow reusing commands, algorithms etc.
- **Scripts** allow sharing code.
- **Scripts** allow the automation of complex analyses.
- **Scripts** allow the automation of report generation; for instance, in R with the packages `knitr` or `sweave`.
- **Scripts** allow rerunning and testing code snippets.
- **Scripts** make it easy to create complex solutions.
- **Scripts** allow limiting the active knowledge of the researcher to the most important aspects.