Introduction to R

Day 1

Sereina Herzog

Institute for Medical Informatics, Statistics and Documentation
Medical University of Graz

09.11.2023

Course Aim

- Introduction to R using RStudio
  - How to use R and RStudio
- Project structure
  - Using R as an example
- Report generation using Rmarkdown
  - Advantage of avoiding “copy & paste”
  - Reproducible reports
- Data visualization with R
  - Using ggplot for typical plots

⇒ **Help for self-help**

Data visualization

This is where we started to draw taller bars to please you guys.
GRUHN

Source: www.googleplussuomi.com

Purpose

- Exploring and presenting data in form of graphs
- Summarizing - data reduction (mean, variance, median etc.)
- Presenting data in form of tables and/or graphs

Scales

Categorical

- Nominal scale
  - The values of any two study units can be classified either as identical or non identical
    - E.g. hair colour, blood group
- Ordinal scale
  - Observation are still classified but some observations have “more” or are “greater than” other observations
    - E.g. school grades

Scales

Continious

- Numerical scales
  - Interval scale
    - Interval scale allows for the degree of difference between items, but not the ratio between them (e.g. dates, °C).
  - Ratio scale
    - A ratio scale possesses a meaningful (unique and non-arbitrary) zero value (e.g. weight, number of children).

Note:

- Numerical scales measured *continuous* (age) or *discrete* (no. of children)
- Nominal and ordinal scale are also known as *qualitative* measurement
- Numerical scale known as *quantitative* measurement

Summarizing data (values)

Common statistics used to summarize data and describe certain attributes of a set of data

- Measure of location (central tendency)
  - Mode
  - Median
  - Arithmetic mean
  - Geometric mean

- Measure of dispersion (spread of data)
  - Standard deviation
  - Variance
  - Interquantile range
  - Range

Summarizing data (graphs)

Visualize data in graphs

- Bar chart
- Histogram
- Box-and-whisker plot
- Time series plot
- Scatterplot
- . . .

When to use what

Idea of data cleaning

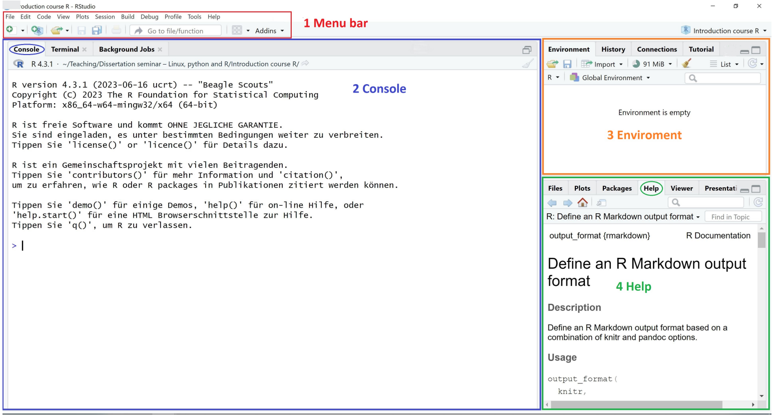
- Check data using
  - Key figures (e.g. median)
  - Graphs (e.g. histogram)
- Data quality
  - consult original source (e.g. patient health record, lab journal)
- Plausibility

R & RStudio

What is R and RStudio?

- R: The R Project for Statistical Computing ▸ Link R project
  - is an open-source programming languages
  - works with *R packages*
- RStudio
  - is an integrated development environment (IDE)
    - specifically designed for working with the R programming language
  - has a user-friendly interface
  - has code editing features
    - code completion feature
    - syntax-highlighting editor

RStudio - Interface



```
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R ist freie Software und kommt OHNE JEGLICHE GARANTIE.
Sie sind eingeladen, es unter bestimmten Bedingungen weiter zu verbreiten.
Tippen Sie 'license()' or 'licence()' für Details dazu.

R ist ein Gemeinschaftsprojekt mit vielen Beitragenden.
Tippen Sie 'contributors()' für mehr Information und 'citation()',
um zu erfahren, wie R oder R packages in Publikationen zitiert werden können.

Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe, oder
'help.start()' für eine HTML Browserschnittstelle zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.

> |
```

output_format {rmarkdown} R Documentation

Define an R Markdown output format

Description

Define an R Markdown output format based on a combination of knitr and pandoc options.

Usage

```
output_format(
  knitr,
```

RStudio - Getting started

- Open RStudio
- Work through ‘Day 1 - Exercise 1’ (together)

Data types and structures in R

- Data types
  - character
  - numeric (real or decimal)
  - integer
  - logical
  - complex
- Data structures
  - atomic vector (i.e. only holds data of a single data type)
  - list
  - matrix
  - data frame
  - factors
  - ...

Examine features in R

- Examine features
  - *class()* - what kind of object is it (high-level)?
  - *typeof()* - what is the object's data type (low-level)?
  - *length()* - how long is it? What about two dimensional objects?
  - *attributes()* - does it have any metadata?
  - ...

Example examing features (I)

```r
x <- "dataset"
typeof(x)
```

```
## [1] "character"
```

```r
attributes(x)
```

```
## NULL
```

Example examing features (II)

```r
y <- 1:10
y
```

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

```r
typeof(y)
```

```
## [1] "integer"
```

```r
length(y)
```

```
## [1] 10
```

Example examing features (III)

```r
z <- as.numeric(c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10))
z
```

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

```r
class(z)
```

```
## [1] "numeric"
```

```r
typeof(z)
```

```
## [1] "double"
```

Reproducibility

What is reproducibility in science?

▸ Link Reproducability

- Ability to reproduce results by a peer
- Requires data, methods, and procedures
- Increasingly, science is supposed to be reproducible

Why does it not happen, in practice?

Some opinions on whether reproducibility is needed:

- *“Ideally, yes but we don’t have time for this.”*
- *“If it gets published, yes.”*
- *“No need: I work on my own.”*
- *“For others to copy us? You crazy?!”*
- *“No way! We rigged the data, the method does not work, and we ran the analyses in Excel”.*

Main obstacles to reproducibility

- Lack of time: ultimately, reproducibility is faster
- Fear of plagiarism: low risks in practice
- Internal work, no need to share: almost never true

- **One good reason:** lack of tools to facilitate reproducibility

You never work alone

Be nice to your future selves!

Reproducibility with RStudio & R

- R with RMarkdown can be used to produce different types of documents [see: http://rmarkdown.rstudio.com/gallery.html]
  - standardised reports (`html`, `pdf`)
  - word documents (`.docx`)
  - slides for presentations (`html`, `pdf`, `powerpoint`)
  - journal articles. using the `rticles` package (`.pdf`)
  - ...

⇒ **making transparent and reproducible analysis**

Folder structure and R projects in RStudio

Folder structure

Suggestion how to structure your project folder

- project1
  - literature
  - reports
  - . . .
  - R
    - orig
    - Rdata
    - Rmarkdown
    - Routput
    - Rfiles

Hint: never touch the original data!

Folder structure

Idea: set path at the beginning of your file with syntax related to your *R* folder and everything else relative to that .

```
path <- "C:/myname/work/project1/R"
setwd(path)
```

For example, data `example0.csv` is in your `Rdata` folder

```
library(readr)
dat <- read_csv(file = "Rdata/example0.csv")
```

OR: use 'R project' option!

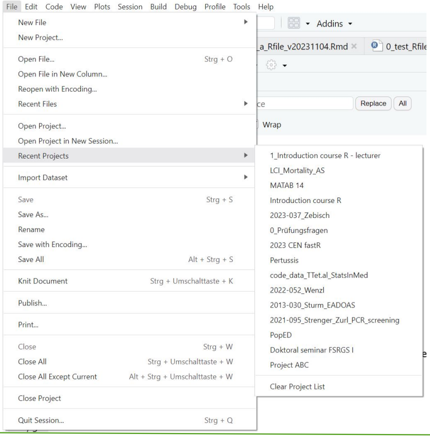
TO DO - Create folder structure

1) Generate following folder structure

- Course Introduction to R
  - slides
  - . . .
  - R
    - orig
    - Rdata
    - Rfiles
    - Rmarkdown
    - Rfiles

R project

- An R project
  - is a way to organize files and folders related to a specific analysis or project
    - easy to switch different projects
    - the working directory is the project's root folder



TO DO - Create R project

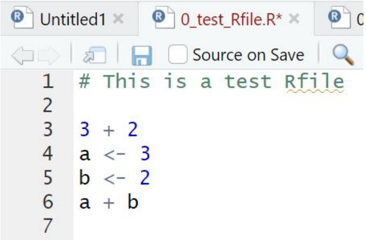
2) Generate a 'R project' (together)

- File → New Project... → Existing Directory

R files

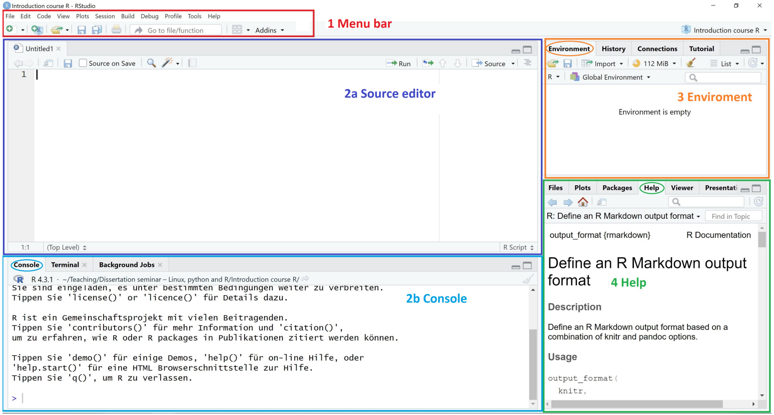
R files

- An R file (*.R*) is
  - a script written in R
  - contains code that can be executed within the R software environment



```
# This is a test Rfile

3 + 2
a <- 3
b <- 2
a + b

```

RStudio - Interface with open script



R 4.3.1 · ~/Teaching/Dissertation seminar – Linux, python and R/Introduction course R/

```
Sie sind eingeladen, es unter bestimmten Bedingungen weiter zu verbreiten.
Tippen Sie 'license()' or 'licence()' für Details dazu.

R ist ein Gemeinschaftsprojekt mit vielen Beitragenden.
Tippen Sie 'contributors()' für mehr Information und 'citation()',
um zu erfahren, wie R oder R packages in Publikationen zitiert werden können.

Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe, oder
'help.start()' für eine HTML Browserschnittstelle zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.

>
```

output_format {rmarkdown} R Documentation

Define an R Markdown output format

Description

Define an R Markdown output format based on a combination of knitr and pandoc options.

Usage

```
output_format(
  knitr,
```

R file - Getting started

- Switch to RStudio
- Work through ‘Day 1 - Exercise 2’ (together)

Rmarkdown

Rmarkdown

- Rmarkdown is a file type (.Rmd) supported within RStudio which can **combine plain text with R code** (‘R chunks’).
- Rmarkdown can combine the results of data analysis (including charts and tables) and the written text (interpretation, summary, comments, etc.) into a single, **reproducible document**.

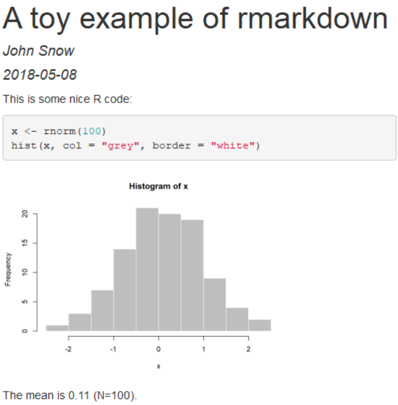
rmarkdown: toy example

```
---
title: “A toy example of rmarkdown”
author: “John Snow”
date: “2018-05-08”
output: html_document
---

This is some nice R code:
` ` `{r rnorm-example, verbatim = TRUE}
```

```r
x <- rnorm(100)
hist(x, col = "grey", border = "white")
```

```
` ` `

The mean is `r round(mean(x), 2)` (N= `r length(x)`).
```

rmarkdown: toy example



A toy example of rmarkdown

John Snow

2018-05-08

This is some nice R code:

```r
x <- rnorm(100)
hist(x, col = "grey", border = "white")
```

The mean is 0.11 (N=100).

Rmarkdown - Getting started

- Switch to RStudio
- Open Rmarkdown file
- Work through ‘Day 1 - Exercise 3’ (together)

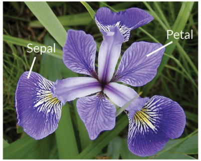
Data visualization with *ggplot*

Example - Iris

A famous iris data set gives the measurements in centimeters of the variables

- sepal length
- sepal width
- petal length
- petal width

for 50 flowers from each of 3 species of iris (*Iris setosa*, *versicolor*, and *virginica*).

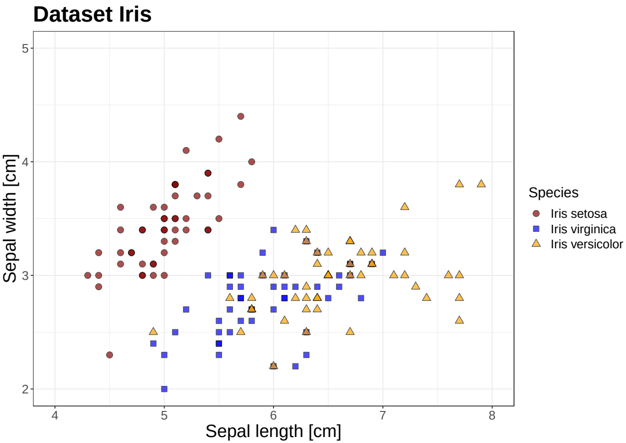


Iris Versicolor

Iris Setosa

Iris Virginica

Example - Iris

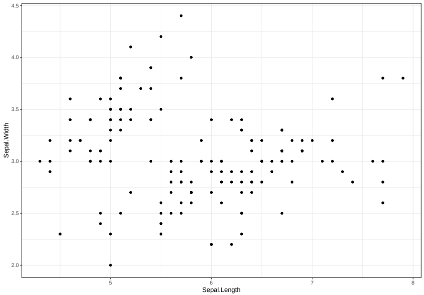


What is *ggplot*?

- powerful data visualization package in R
  - wide range of high-quality plots and graphics
  - provides a consistent syntax
  - a layered approach to building plots
- consists of three main components:
  - **data**
    - represents the dataset being visualized
  - **aesthetics** (aes)
    - define how variables are mapped to visual properties (e.g., x-axis, y-axis, color)
  - **geometric objects** (geom)
    - determine the type of plot (e.g., points, lines, bars)

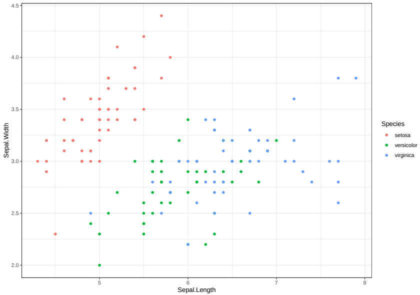
Example - Iris

```
ggplot(data = iris,
       aes(x = Sepal.Length, y = Sepal.Width)) +
    geom_point() +
    theme_bw()
```



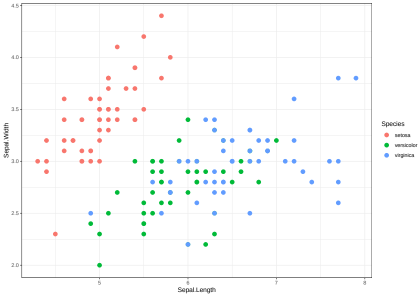
Example - Iris: including species as colour

```r
ggplot(data = iris,
       aes(x = Sepal.Length, y = Sepal.Width, colour = Species)) +
    geom_point() +
    theme_bw()
```



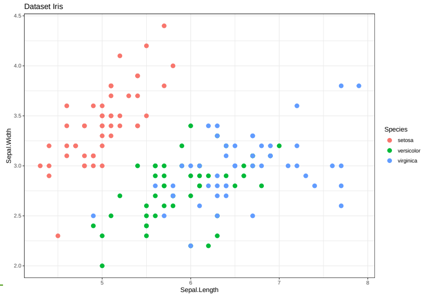
Example - Iris: increase point size

```
ggplot(data = iris,
       aes(x = Sepal.Length, y = Sepal.Width, colour = Species)) +
    geom_point(size = 3) +
    theme_bw()
```



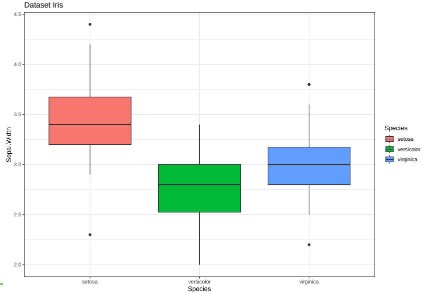
Example - Iris: adding title

```r
ggplot(data = iris,
       aes(x = Sepal.Length, y = Sepal.Width, colour = Species)) +
    geom_point(size = 3) +
    labs(title = "Dataset Iris") +
    theme_bw()
```



Example - Iris: using another geom

```
ggplot(data = iris,
       aes(x = Species, y = Sepal.Width, fill = Species)) +
    geom_boxplot() +
    labs(title = "Dataset Iris") +
    theme_bw()
```



ggplot - Getting started

- Switch to RStudio
- Open Rmd file: *day1_ex4_ggplot_v20231108.Rmd*
  - is on GitHub in folder ‘Course Introduction R 2023/Day1’
- Work through ‘Day 1 - Exercise 4’

Saving ggplots

```
plot_iris <-
  ggplot(data = iris,
         aes(x = Sepal.Length, y = Sepal.Width, colour = Species)) +
      geom_point() +
      theme_bw()

ggsave(filename = "../Routputs/example_iris.png", plot = plot_iris,
       units = "cm", width = 12, height = 7)
```

- Try to save your last plot in the 'Day 1 - Exercise 4'
    - test different formats and values for width/height

Chunk options in Rmarkdown

- See cheat sheet within RStudio
- Make copy of your ‘Day 1 - Exercise 4’ Rmarkdown file and try chunk options

Links

Links

- Introduction to R
  - R for Data Science (https://r4ds.hadley.nz/)
- Plots using ggplot
  - Overview with further links to course material: https://ggplot2.tidyverse.org/
- Display tables using flextable
  - flextable bool https://ardata-fr.github.io/flextable-book/
  - Function references https://davidgohel.github.io/flextable/reference/index.html
- Download R
  - CRAN (https://cran.r-project.org/)
- Download RStudio
  - RStudio Desktop (https://posit.co/download/rstudio-desktop/)