Introduction to R

Day 2

Sereina Herzog & Gudrun Pregartner

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

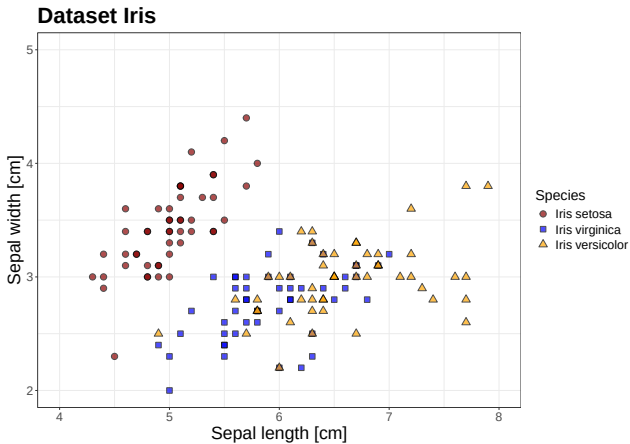
30.01.2026

Course Content - Introduction R (Day 2)

- Saving plots
- R Markdown

Saving plots in R

Example Iris



Example Iris

```
lab_species <- c("Iris setosa", "Iris virginica", "Iris versicolor")

plot_iris_f <-
  ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width, fill = Species)) +
      geom_point(aes(shape = Species), size = 3, alpha = 0.7) +

      scale_fill_manual(values = c("darkred", "blue", "orange"),
                        labels = lab_species) +
      scale_shape_manual(values = c(21, 22, 24),
                         labels = lab_species) +
      labs(title = "Dataset Iris") +
      xlab("Sepal length [cm]") +
      ylab("Sepal width [cm]") +
      coord_cartesian(xlim = c(4, 8), ylim = c(2, 5)) +
      theme_bw() +
      theme(plot.title = element_text(face = "bold", size = 25),
            axis.title = element_text(size = 20),
            axis.text = element_text(size = 14),
            legend.title = element_text(size = 16),
            legend.text = element_text(size = 14))

plot_iris_f
```

Saving plots

Several possibilities

Saving plots

Several possibilities

- plot image in RStudio and work with ‘Plots’ panel (lower right corner)

Saving plots

Several possibilities

- plot image in RStudio and work with ‘Plots’ panel (lower right corner)
- save your image in a specific format (e.g., jpeg(), png(), svg(), pdf())

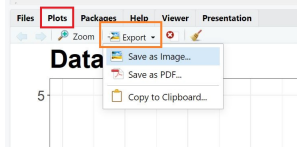
Saving plots

Several possibilities

- plot image in RStudio and work with ‘Plots’ panel (lower right corner)
- save your image in a specific format (e.g., jpeg(), png(), svg(), pdf())
- for ggplot images use ggsave()

Saving plots - ‘Plots’ panel

‘Plots’ panel (lower right corner) → Export → Save as Image or Save as PDF



Saving plots - jpeg()

```
jpeg(filename = "plot_iris_v20231109.jpg",
     width = 12, height = 7, units = "cm", res = 300)

  plot_iris_f

dev.off()
```

- Look at documentation for all options

Saving ggplots

```
ggsave(filename = "plot_iris_ggsave_v20231109.png",
       plot = plot_iris_f,
       units = "cm", width = 12, height = 7)
```

Saving plot - exercise

- Switch to RStudio
- Open R file: *day2_ex1_saving_ggplot_vYYYYMMD.R*
- Work through 'Day 2 - Exercise 1'

Introduction R Markdown

What is Markdown?

- Markdown is a lightweight markup language for creating formatted text using a plain-text editor

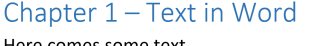
Microsoft Word vs Markdown

Comparison

Microsoft Word vs Markdown

Comparison

- e.g., *Microsoft Word*: write as you see, clicking buttons to format words and phrases, and see changes instantly



Microsoft Word vs Markdown

Comparison

- e.g., *Microsoft Word*: write as you see, clicking buttons to format words and phrases, and see changes instantly

Chapter 1 – Text in Word

Here comes some text

- in *Markdown-formatted file*: you need 'markdown syntax' indicating which words and phrases should look different

```
# Chapter 1 - Markdown
Here comes some text...
```

⇨

Chapter 1 - Markdown

Here comes some text...

Why Markdown?

- Markdown allows you to focus more on content and less on its presentation
- Markdown is simple and intuitive
- You need only a text editor

What is R markdown?

- An R Markdown document (.Rmd) is written in markdown and can **combine plain text with R code** ('R chunks').

What is R markdown?

- An R Markdown document (.Rmd) is written in markdown and can **combine plain text with R code** ('R chunks').
- R Markdown can combine the results of data analysis (including charts and tables) and the written text (interpretation, summary, comments, etc.) into a single, **reproducible document**.

Output formats for .Rmd

- **Documents**

Output formats for .Rmd

- **Documents**
  - html_document - HTML document
  - pdf_document - PDF document (via LaTeX template)
  - word_document - Microsoft Word document (docx)
  - odt_document - OpenDocument Text document
  - ...

Output formats for .Rmd

- **Documents**
  - html_document - HTML document
  - pdf_document - PDF document (via LaTeX template)
  - word_document - Microsoft Word document (docx)
  - odt_document - OpenDocument Text document
  - ...
- **Presentations** (slides)

Output formats for .Rmd

- **Documents**
  - html_document - HTML document
  - pdf_document - PDF document (via LaTeX template)
  - word_document - Microsoft Word document (docx)
  - odt_document - OpenDocument Text document
  - ...
- **Presentations** (slides)
  - beamer_presentation - PDF presentation with LaTeX Beamer
  - powerpoint_presentation: PowerPoint presentation
  - ...

Output formats for .Rmd

- **Documents**
  - html_document - HTML document
  - pdf_document - PDF document (via LaTeX template)
  - word_document - Microsoft Word document (docx)
  - odt_document - OpenDocument Text document
  - ...
- **Presentations** (slides)
  - beamer_presentation - PDF presentation with LaTeX Beamer
  - powerpoint_presentation: PowerPoint presentation
  - ...
- **More**
  - books
  - websites
  - ...

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}

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

```

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

# `rmarkdown`: toy example

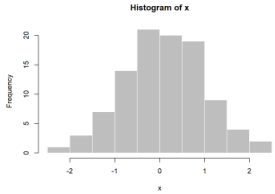
## A toy example of rmarkdown

*John Snow*

*2018-05-08*

This is some nice R code:

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



The mean is 0.11 (N=100).

R Markdown - Getting started (part I)

We focus first on Markdown syntax

- Switch to RStudio
- Open Rmd file: *day2_ex2_markdown_syntax_vYYYYMMDD.Rmd*
  - is on GitHub in R/Rmarkdown
- Work through 'Day 2 - Exercise 2' (together; no pdf file)
  - look at cheat sheet 'R Markdown Cheat Sheet' (page 1 'Write with Markdown')

Chunk options in R Markdown

- There are several options regarding code chunks, e.g.,

Chunk options in R Markdown

- There are several options regarding code chunks, e.g.,
  - **echo** display code in output document (default TRUE)

Chunk options in R Markdown

- There are several options regarding code chunks, e.g.,
  - **echo** display code in output document (default TRUE)
  - **eval** rund code in chunk (default TRUE)

Chunk options in R Markdown

- There are several options regarding code chunks, e.g.,
  - **echo** display code in output document (default TRUE)
  - **eval** rund code in chunk (default TRUE)
  - **fig.width** width of plot dimensions in inches (default 7)
  - ...

Chunk options in R Markdown

- There are several options regarding code chunks, e.g.,
  - **echo** display code in output document (default TRUE)
  - **eval** rund code in chunk (default TRUE)
  - **fig.width** width of plot dimensions in inches (default 7)
  - ...
- The options can be set globally and/or separately for each code chunk

Chunk options in R Markdown

- There are several options regarding code chunks, e.g.,
  - **echo** display code in output document (default TRUE)
  - **eval** rund code in chunk (default TRUE)
  - **fig.width** width of plot dimensions in inches (default 7)
  - ...
- The options can be set globally and/or separately for each code chunk
  - globally: use within a code chunk, e.g., *knitr::opts_chunk$set(echo = TRUE)*

Chunk options in R Markdown

- There are several options regarding code chunks, e.g.,
  - **echo** display code in output document (default TRUE)
  - **eval** rund code in chunk (default TRUE)
  - **fig.width** width of plot dimensions in inches (default 7)
  - ...
- The options can be set globally and/or separately for each code chunk
  - globally: use within a code chunk, e.g., *knitr::opts_chunk$set(echo = TRUE)*
  - for one code chunk: within the curly brackets, e.g., ` ``{r, echo = FALSE}`

Chunk options in R Markdown

- There are several options regarding code chunks, e.g.,
  - **echo** display code in output document (default TRUE)
  - **eval** rund code in chunk (default TRUE)
  - **fig.width** width of plot dimensions in inches (default 7)
  - ...
- The options can be set globally and/or separately for each code chunk
  - globally: use within a code chunk, e.g., *knitr::opts_chunk$set(echo = TRUE)*
  - for one code chunk: within the curly brackets, e.g., ` ``{r, echo = FALSE}`
- See cheat sheet within RStudio
  - Help → Cheat sheets → R Markdown Cheat Sheet

R Markdown - Getting started (part II)

Now we combine Markdown and 'R chunks'

- Switch to RStudio
- Open Rmd file: *day2_ex3_rmarkdown_vYYYYMMDD.Rmd*
  - is on GitHub in R/Rmarkdown
- Work through 'Day 2 - Exercise 3' (together; no pdf file)
  - Hint: use cheat sheet 'R Markdown Cheat Sheet'

R Markdown - Getting started (part III)

Create a report with the figures form ‘Day 1 - Exercise 3’

- Switch to RStudio
- You can use the solution provided in R file:
  day1_ex3_ggplot_vYYYYMMDD_SOLUTION.Rmd
  - is on GitHub in R/Rfiles
- Create a Rmd file containing the ‘final’ figures from each task - ‘Day 2 - Exercise 4’ (no pdf file; no starting file)
  - save your solution as *day2_ex4_report_figures_vYYYYMMDD.Rmd*

Data cleaning with tidyverse

Data table

- each **unit** (e.g. patient, mouse, cell) equals a row
- for each unit the measured **variables** (e.g. age, blood pressure, size) equal columns

Data table

- each **unit** (e.g. patient, mouse, cell) equals a row
- for each unit the measured **variables** (e.g. age, blood pressure, size) equal columns

id	gender	age	weight	height	smoking
1	1	35	70.5	185	0
2	2	36	65.3	170	0
3	2		90.1	164	1
4	1	21	72.0	177	0
5	1	66	89.4	175	0

Repeated measurements

wide format

id	gender	syst0	syst1
1	1	120	125
2	2	118	125
3	2		110

Repeated measurements

wide format

id	gender	syst0	syst1
1	1	120	125
2	2	118	125
3	2		110

long format

id	gender	syst	time
1	1	120	0
1	1	125	61
2	2	118	0
2	2	125	60
3	2		
3	2	110	59

What is tidyverse

- tidyverse is a collection of R packages designed for data science

What is tidyverse

- tidyverse is a collection of R packages designed for data science
  - they share an underlying design philosophy, grammar, and data structure

What is tidyverse

- tidyverse is a collection of R packages designed for data science
  - they share an underlying design philosophy, grammar, and data structure
    - *ggplot2* for data visualization

What is tidyverse

- tidyverse is a collection of R packages designed for data science
  - they share an underlying design philosophy, grammar, and data structure
    - *ggplot2* for data visualization
    - *readr* for data importation from various file sources

What is tidyverse

- tidyverse is a collection of R packages designed for data science
  - they share an underlying design philosophy, grammar, and data structure
    - *ggplot2* for data visualization
    - *readr* for data importation from various file sources
    - *tidyr* and *dplyr* useful for data cleaning

What is tidyverse

- tidyverse is a collection of R packages designed for data science
  - they share an underlying design philosophy, grammar, and data structure
    - *ggplot2* for data visualization
    - *readr* for data importation from various file sources
    - *tidyr* and *dplyr* useful for data cleaning
    - ...
  - all core packages can be loaded at once: *library(tidyverse)*

What is tidyverse

- tidyverse is a collection of R packages designed for data science
  - they share an underlying design philosophy, grammar, and data structure
    - *ggplot2* for data visualization
    - *readr* for data importation from various file sources
    - *tidyr* and *dplyr* useful for data cleaning
    - ...
  - all core packages can be loaded at once: *library(tidyverse)*
  - 'R for Data Science' (see slide with links)

Useful functions for data cleaning

- **select()** extracts columns and returns a tibble

Useful functions for data cleaning

- **select()** extracts columns and returns a tibble
- **arrange()** changes the ordering of the rows

Useful functions for data cleaning

- **select()** extracts columns and returns a tibble
- **arrange()** changes the ordering of the rows
- **filter()** picks cases based on their values

Useful functions for data cleaning

- **select()** extracts columns and returns a tibble
- **arrange()** changes the ordering of the rows
- **filter()** picks cases based on their values
- **mutate()** adds new variables that are functions of existing variables

What is %>% in Tidyverse?

%>% is used to emphasis a sequence of actions, rather than the object that the actions are being performed on

What is %>% in Tidyverse?

%>% is used to emphasis a sequence of actions, rather than the object that the actions are being performed on

```
dt_example %>%
  mutate(bmi = weight/(height^2)) %>%
  select(pat_id, sex, bmi)
```

What will we cover

- We will look at
  - importing data (example: .xlsx)
  - useful function for data cleaning
  - save R environment (.Rdata)
- We will work with .Rdata in a R Markdown file

Data cleaning - exercise

- Example Glucose:
  - Glucose tolerance was tested by administering 100g glucose drink
  - Glucose was tested before and 1 hour after administering
  - source: R package medicaldata

Data cleaning - exercise

- Example Glucose:
  - Glucose tolerance was tested by administering 100g glucose drink
  - Glucose was tested before and 1 hour after administering
  - source: R package medicaldata
- Switch to RStudio
- Open R file: *day2_ex5_datacleaning_vYYYYMMDD.R*
  - is on GitHub in R/Rfiles
- Work through 'Day 2 - Exercise 5' (together; no pdf file)

Links

Links (I)

- 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
- knit_child()
  - link (https://bookdown.org/yihui/rmarkdown-cookbook/child-document.html)

Links (II)

- Download R
  - CRAN (https://cran.r-project.org/)
- Download RStudio
  - RStudio Desktop (https://posit.co/download/rstudio-desktop/)