JEB157 Data Analysis in R
Week #1

Course information

&

Introduction to R and RStudio

Ladislav Krištoufek

Outline

1. Core course information
   - Basics
   - Schedule
   - Grading

2. R and RStudio

Outline

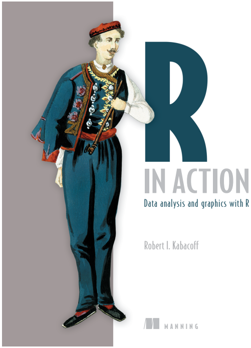
1. Core course information
   - Basics
   - Schedule
   - Grading

2. R and RStudio

Course information

- JEB157 Data Analysis in R
- Lectures/Seminars:
    - Ladislav Krištoufek (lectures/seminars)
    - Anna Drahozalová (assignments & group consultations)
    - Pre-recorder lectures/seminars (links in SIS, recorded in Loom)
    - Offline consultations (not mandatory), room 016:
        - 27 Oct (Week 4), 10 Nov (Week 6), 24 Nov (Week 8), 8 Dec (Week 10), and 22 Dec (Week 12)
        - between 11:00 and 12:30
- Contact: LK@fsv.cuni.cz

Study materials



- Available in the IES library. However, it is (quite) easily accessible online. There is also a GitHub page for the book.
- DataCamp courses (also Python, SQL, Power BI, Tableau, Excel, Docker, PyTorch, Spark, Git, Julia, Scala, and others).

Course aims . . .

- are:
  - to make you comfortable writing your own functions in R
  - to make you comfortable analyzing data in R
  - to enlarge your skill portfolio
  - to make you more attractive to employers
- are not:
  - to go deep into theory of the presented methods
  - to make you a proficient R coder

prg.ai

Course pre-requisites

- This is a mandatory bachelor's course (new accreditations).
- JEB142 Introductory Statistics is a pre-requisite for this course.
- No previous knowledge of R is assumed.

Outline

1. Core course information
    - Basics
    - Schedule
    - Grading

2. R and RStudio

Course schedule

- Week #1: Course information + Introduction to R and RStudio
- Week #2: Creating a dataset
- Week #3: Basic data management
- Week #4: Advanced data management
- Week #5: Getting started with graphs
- Week #6: Basic graphs
- Week #7: Basic statistics
- Week #8: Analysis of variance
- Week #9: Power analysis
- Week #10: Intermediate graphs
- Week #11: Resampling statistics and bootstrapping
- Week #12: Principal component analysis and factor analysis

Outline

1. Core course information
   - Basics
   - Schedule
   - Grading
2. R and RStudio

Grading

The final grade consists of two components:

- 3 skill tracks in DataCamp:
  - Skill Track “R Programming” (7.5 points) - by 19 November 2023 CET
  - Skill Track “Importing & Cleaning Data” (7.5 points) - by 10 December 2023 CET
  - Career Track “Data Analyst with R” (20 points) - by 4 February 2024 CET
- 3 assessments in DataCamp:
  - “Understanding and Interpreting Data” (5 points) - by 5 November 2023 CET
    - At least 120 score in DataCamp to pass and obtain 5 points.
  - “R Programming” (20 points) - by 19 November 2023 CET
  - “Importing & Cleaning Data” (20 points) - by 10 December 2023 CET
  - “Data Manipulation with R” (20 points) - by 4 February 2024 CET
    - To get the score, use the DataCamp score x and fit it to $(x - 60)/80 * 100\% \Rightarrow$ 140+ score means full points from the Assessment.
    - At least 50%, i.e. at least 10 points, from each assessment is a necessary (not a sufficient) condition for passing the Data Analysis in R course $\Rightarrow$ you need at least 100 score to pass the Assessment.
  - You can re-take the assessments twice a week during the whole semester (up till the deadline, of course). Remember that the last one counts (not necessarily the best one).

Grading

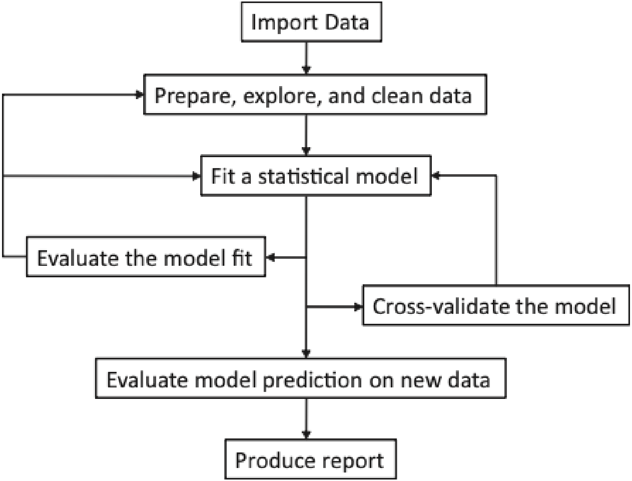
Grading scale (following Dean’s Provision 17/2018):

- A: $points > 90$
- B: $80 < points \leq 90$
- C: $70 < points \leq 80$
- D: $60 < points \leq 70$
- E: $50 < points \leq 60$
- F: $points \leq 50$

Why R?

- Simple but powerful and robust software.
- Huge user community (packages, helps, StackOverflow.com, and others).
- Freely available.
- Works on all major platforms.
- A comprehensive statistical platform, pretty much anything in data analysis can be done here.
- Works with any type of data.
- Has become quite popular with the employers but also in the research community.
- The skills (syntax/algorithmic thinking) are transferable.

Steps in a typical data analysis



RStudio

- RStudio is the most popular R interface.
- Clearly and well organized.
- Allows for efficient usage for various levels of users.

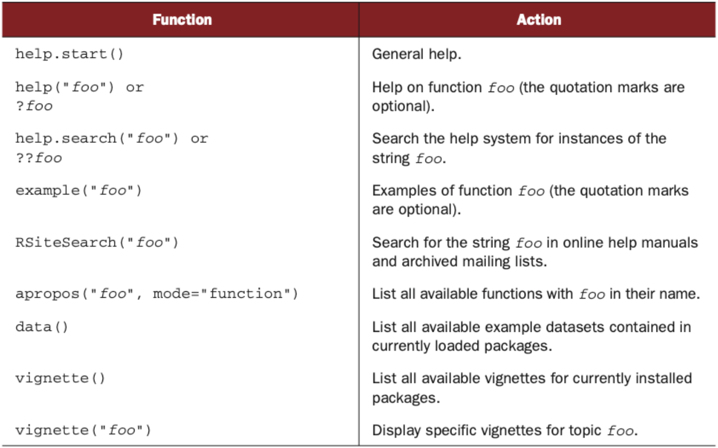
Getting R and RStudio

- It is necessary to install both.
- R is available on `cran.r-project.org`.
- RStudio is available on `rstudio.com`.

The very basics

- Case-sensitive.
- Various data types and structures (we will cover these next week).
- Most functionality is provided through built-in and user-created functions. Data objects are created in memory during the sessions.
- Basic functions are available by default. Other functions are contained in packages.
- Statements consist of functions and assignments. R uses the symbol $< -$ for assignments, rather than the typical $=$ (even though it can be used). E.g. *x <- rnorm(5)* creates a vector object named *x* containing five random draws from a standard normal distribution. The “arrow” can be set in the other direction, keeping the assigning logic (but it is not a standard way of writing code in R).
- Comments are preceded by the # symbol. Code on such line (or following the hash) is ignored by the R interpreter.

Getting help

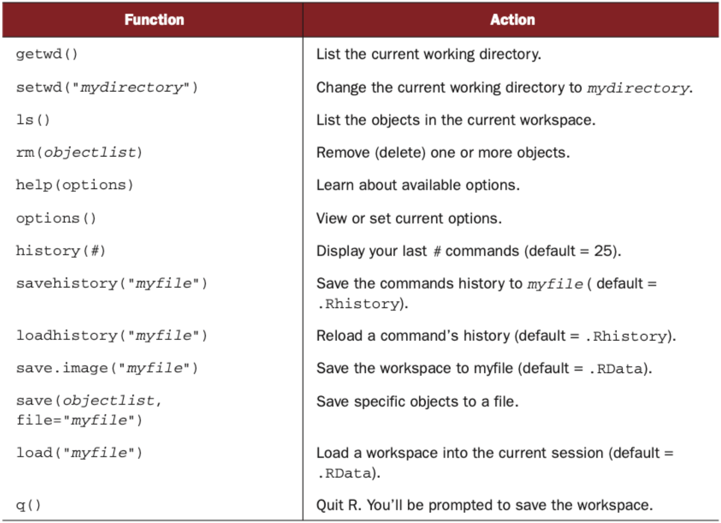


Function	Action
`help.start()`	General help.
`help("foo")` or `?foo`	Help on function *foo* (the quotation marks are optional).
`help.search("foo")` or `??foo`	Search the help system for instances of the string *foo*.
`example("foo")`	Examples of function *foo* (the quotation marks are optional).
`RSiteSearch("foo")`	Search for the string *foo* in online help manuals and archived mailing lists.
`apropos("foo", mode="function")`	List all available functions with *foo* in their name.
`data()`	List all available example datasets contained in currently loaded packages.
`vignette()`	List all available vignettes for currently installed packages.
`vignette("foo")`	Display specific vignettes for topic *foo*.

Workspace

- Your current working environment that includes any user-defined objects (vectors, matrices, functions, data frames, or lists).
- You can save an image of the current workspace which is then automatically reloaded.
- The current working directory is the directory R will read files from and save results to by default.
- Mind the difference between slash and backslash when/if spelling out the directory path.

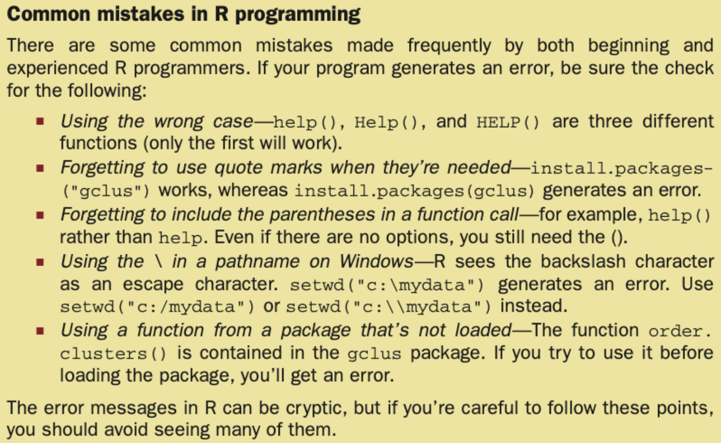
Workspace functions



Function	Action
getwd()	List the current working directory.
setwd("*mydirectory*")	Change the current working directory to *mydirectory*.
ls()	List the objects in the current workspace.
rm(*objectlist*)	Remove (delete) one or more objects.
help(options)	Learn about available options.
options()	View or set current options.
history(#)	Display your last # commands (default = 25).
savehistory("*myfile*")	Save the commands history to *myfile* (default = .Rhistory).
loadhistory("*myfile*")	Reload a command's history (default = .Rhistory).
save.image("*myfile*")	Save the workspace to myfile (default = .RData).
save(*objectlist*, file="*myfile*")	Save specific objects to a file.
load("*myfile*")	Load a workspace into the current session (default = .RData).
q()	Quit R. You'll be prompted to save the workspace.

Packages

- Rich library of ready-to-use package.
- Most or rather practically all standard statistical and econometrical methods are already coded. Sometimes, one needs to look.
- Needed packages need to be loaded at the beginning of the session or when needed via the *library()* function.
- Packages usually come with a detailed help. This can be clicked through in RStudio, or called via *help(package="package_name")*.



Common mistakes in R programming

There are some common mistakes made frequently by both beginning and experienced R programmers. If your program generates an error, be sure the check for the following:

- *Using the wrong case*—`help()`, `Help()`, and `HELP()` are three different functions (only the first will work).
- *Forgetting to use quote marks when they're needed*—`install.packages("gclus")` works, whereas `install.packages(gclus)` generates an error.
- *Forgetting to include the parentheses in a function call*—for example, `help()` rather than `help`. Even if there are no options, you still need the ().
- *Using the \ in a pathname on Windows*—R sees the backslash character as an escape character. `setwd("c:\mydata")` generates an error. Use `setwd("c:/mydata")` or `setwd("c:\\mydata")` instead.
- *Using a function from a package that's not loaded*—The function `order.clusters()` is contained in the `gclus` package. If you try to use it before loading the package, you'll get an error.

The error messages in R can be cryptic, but if you're careful to follow these points, you should avoid seeing many of them.

Next lecture

- Creating a dataset
  - Understanding datasets
  - Data structures
  - Data input
  - Annotating datasets
  - Useful functions for working with data objects