Lesson 1:
Data Management

Recap: R language

- Functions
- Objects
- Data Frames

Recap: R Studio

- Files, Plots, Packages, Help
- Environment & History
- Source
- Commander

Recap: Importing / Exporting

- Downloading and opening packages
- Importing excel files
- Exporting excel files

Important commands (recap)

COMMAND	EFFECT
`library(packagename)`	opens package from library (must be already installed)
`install.package("packagename")`	installs package into your library
`setwd("C:/Path_to_your_WD")`	sets your working directory (you can access files directly now)
`getwd()`	will display the path to your working directory
`help("function")` or `?function`	will display the manual page of given function
`#`	used for commenting as R will ignore anything after hash
`c(…)`	Combining values or strings to a vector (if using strings, put values in parentheses)

`factor(variable,`
`  levels = c(1, 2),`
`  labels = c("male", "female")`
	Turns numeric variable into a factor. Level 1 = male, level 2 = female.

`as.numeric()`	Turns character variable into numeric variable
`data.frame(var1, var2)`	Combines two variables into one data frame
`as.matrix()`	Turns data frame into matrix
`$` (Example: `table$column_1`)	Used to select a particular column in a table
`dataframe[rows, columns]`	To specify which rows and columns will be used

Data Management

1. Import Data
2. Check, if read correctly
3. Transform variables
4. Select subsets

1. Import Data

- `read_excel()` function for excel files (package: `readxl`)
- Also packages for .txt, SPSS files, .csv, etc.
- Clicking on “Files” also works!

2. Check, if read correctly

- Look at data via environment
- Display the first few rows: `head(data)`
- List all variables of the dataset: `ls(data)`
- Names of all variables of the dataset: `names(data)`
- Summary of a variable: `summary(data$variable)`
- Number of persons (rows): `nrow(data)`
- Number of variables (columns): `ncol(data)`
- Look at missing values: NA? -99?

3. Transform variables

- Were categorical variables imported as factors?
- Were numerical variables imported indeed as numerical ones?
- Important functions: `is.factor()` and `is.numeric()` will tell you `TRUE` or `FALSE`
- If not, use `factor()` and `as.numeric()` to change

  Example: `data$sex <- factor(data$sex)`

  new variable: factor

  old variable: numeric

  - Name of new and old variable is the same: old variable is written over

4. Select subsets

- Create subset via `subset()` command

  Example: `subset(data, data$alive == “yes”)`

  → selects all rows for which status alive is `TRUE`

- Delete a variable: `data$variable <- NULL`

  → variable is now “null”

4. Select subsets

- Create a new variable: `data$new_var`

  → `new_var` is included in dataset (but empty at the moment)

- Appoint values to new variable

  Example: `data$new_var[data$age < 50] <- “young”`

  - The new variable new_var is now ”young” for all persons with an age below 50

Preview: Descriptive Statistics

What we already know:

- `summary()` → descriptive information for numeric variables
- But how to report about categorical variables?

Example: calculate quantiles

- Load and open package `stats`
- Use `quantile(data$variable)` for quartile calculation

Important commands

COMMAND	EFFECT
`head(data$variable)`	prints first few rows of dataset, including variable names
`ls(data)`	creates a list of all variable names (alphabetized)
`names(data)`	prints all variables names (in order of appearance)
`summary(data$variable)`	summarizes descriptive statistics of variable
`nrow(data)`	number of rows
`ncol(data)`	number of columns
`subset(data, data$variable == "condition")`	creates a subset from dataset
`quantile(data$numeric_variable)`	(package stats) calculates quartiles of numeric variable

remember: you can always use `help()`, `?` or simply google a command to find out more!