Biostatistics & Epidemiological Data Analysis using R

2
Manipulating objects in R

Stefan Konigorski

Health Intervention Analytics Group, HPI

October 26, 2022

HPI Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Content

Block	Class	Content	Date
R, Data manipulation, Descriptives	1	Overview & Introduction to R and data analysis	2022.10.19
	2	First steps in data analysis using R	2022.10.26
	3	Second steps in data analysis using R	2022.11.02
Epidemiology & Statistics: concepts	4	Epidemiological study designs and study planning	2022.11.09
	5	Estimation	2022.11.16
	6	Hypothesis testing	2022.11.23
	7	Missing data	2022.11.30
Data analysis w/ regression models	8	Linear regression I	2022.12.07
	9	Linear regression II	2022.12.14
	10	Regression models for binary and count data	2023.01.11
	11	Analysis of variance & Linear mixed models I	2023.01.18
	12	Linear mixed models II & Meta analysis	2023.01.25
	13	Survival analysis	2023.02.01
	14	Causal inference & Data analysis challenge	2023.02.08

(see full schedule online)

Learning objectives

- Introduction to documentation using R Markdown.
- Learn which data checks can be important and how to do them in R.
- Learn and practice how to manipulate objects in R (variables=vectors and datasets=data frames) in order to create new variables, transform variables, and select subsets of variables or observations.

Main steps of a data analysis

1. Import dataset from an external file (e.g. xls, txt, SPSS file).
2. Import check: check if dataset has been read correctly.
3. Save dataset as R dataset (.Rdata), e.g. as `dat_raw.Rdata`.
4. Data check: check if data is correct/missing, and e.g. remove probands/variables or decide for imputation. Save corrected dataset as new dataset, e.g. `dat_corrected.Rdata`.
5. Transform variables, compute new variables, and/or select subset for final analysis. Save this again as new dataset, e.g. as `dat_final.Rdata`, and use in all further steps.
6. Descriptives to describe main characteristics of study sample.
7. Main analyses.
8. Secondary analyses.
9. Sensitivity analyses.

Review class 1 - import data

- Overview of R, RStudio, packages and help functions (homework 1, exercises 2-4).
- Different functions available to import csv, excel files, and many more (homework 1, exercises 5-7).
- In order to use some of them, the respective package has to be installed and loaded first!
- Save datasets as .RData files (homework 1, exercise 1).

Review class 1 - objects in R

Important objects

- Vectors, data frames
- Vectors = variables
- Data frames = rectangular matrices with observations in rows and variables in columns

What else to remember?

- Missing value in R: NA
- Access elements with `[.]` operator
- R objects have classes, e.g. data frame is a class, and also character, numeric, logical, factor, date are classes (of vectors)

Insert: Documentation of analyses and results

- Use R Markdown.
- See `R_2b_RMarkdown.pdf`.

Exercise 1

- See `R_2_exercises.Rmd`

Main steps of a data analysis

1. Import dataset from an external file (e.g. xls, txt, SPSS file).
2. Import check: check if dataset has been read correctly.
3. Save dataset as R dataset (.Rdata), e.g. as `dat_raw.Rdata`.
4. Data check: check if data is correct/missing, and e.g. remove probands/variables or decide for imputation. Save corrected dataset as new dataset, e.g. `dat_corrected.Rdata`.
5. Transform variables, compute new variables, and/or select subset for final analysis. Save this again as new dataset, e.g. as `dat_final.Rdata`, and use in all further steps.
6. Descriptives to describe main characteristics of study sample.
7. Main analyses.
8. Secondary analyses.
9. Sensitivity analyses.

Step 4 - Data check

Goal

Now check if data is correct or if there is something weird.

What to do if weird?

- Go back and check the raw data.
- Check if weird values are wrong, suspicious, or outliers?
- Transform variable, remove variable or observation?
- If many missing values, think about missing value imputation.
- Important: think critically in order not to bias your analysis (can check e.g. in sensitivity analysis)!!

Step 4 - Data check

How to check and transform?

I often use logical evaluations together with the `table()` function:

- Important logical operators:
  - logical EQUAL: ==
  - logical AND: &
  - logical OR: |
  - logical NOT: !
- In combination with functions to compare/evaluate values such as <, >, <=, >=, `is.na()`, and further specific functions to e.g. evaluate strings, many questions can be evaluated.
- The number of times this evaluations is true can be then displayed using the `table()` function.

Step 4 - Data check

How to check and transform?

Examples:

- Does anyone have age smaller than 0: `table(age < 0)`
- How many missing values does the variable age have: `table(is.na(age))`
- How people have a BMI of 0: `table(BMI == 0)`
- How people have insulin level of 0: `table(insulin == 0)`
- Are those people with BMI 0 the same people with insulin 0: `table((BMI == 0) & (insulin == 0))`

Exercise 2

- Do exercise 2a and 2b in `R_2_exercises.Rmd`.

Step 5 - Manipulate variables and observations

After checking if the data is correct,

- transform variables and
- select final sample
- in order to prepare the dataset that you will use in all your following analyses.

Step 5 - Transform variables

Examples

- Change variable type using the functions `as.numeric()`, `as.character()`, `as.factor()`, `as.numeric(as.character())`, `as.Date()`.
- Create new variable through mathematical operation, e.g.:
  - compute BMI from height and weight:
    `dat$BMI <- dat$weight/(dat$height^2)`
  - standardize variables with `scale()` function:
    `dat$BMI_z <- scale(dat$BMI)`
- Remove/add/replace values of variable with `[.]` operator, e.g.:
  - `dat$BMI[1] <- 20`
  - `dat$BMI[dat$BMI < 0] <- NA`

Step 5 - Transform data frame

Examples

- Same ideas as for transforming variables (columns of data frame = variables = vectors!)
- Select subset of data frame to filter variables/observations, or add columns/rows. This can be done using the `[, ]` operator, `data.frame()` function, and others, e.g.:
  - `dat[!dat$Age == 0, ]`
  - `dat_female <- dat[dat$Gender == "F", ]`
  - `dat_final <- data.frame(ID = dat_female$PatientId, Age = dat_female$Age, NoShow = dat_female$No-show)`
  - `subset()` function.

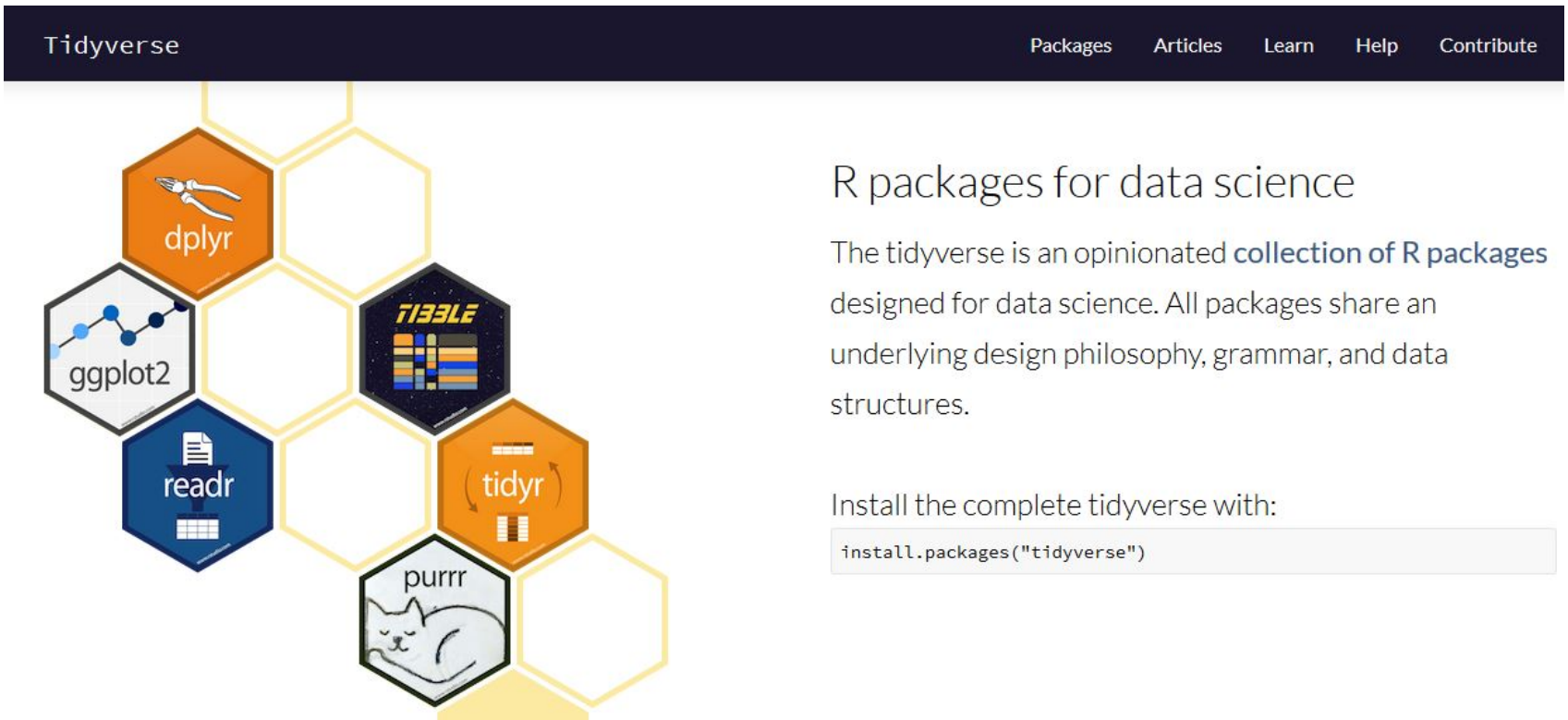
Step 5 - Transform variables and data frames

Exercise 3

See `R_2_exercises.Rmd`.

Tidyverse

- In R, in addition to the "classical" R programming, which we have mostly used so far, there are many new packages and functions that introduce new objects and structures how to program.
- Many are subsumed in the tidyverse (`www.tidyverse.org`):



Tidyverse

Overview and references

- Tidyverse covers the packages `dplyr`, `tidyr`, `readr`, `ggplot2` and others.
- Tidyverse manifesto: `https://cran.r-project.org/web/packages/tidyverse/vignettes/manifesto.html`.
- Overview of data import functions: `https://rawgit.com/rstudio/cheatsheets/master/data-import.pdf`.
- Overview of data manipulation functions: `https://dplyr.tidyverse.org/`
- See also books `https://r4ds.had.co.nz/` and `http://adv-r.had.co.nz/`.

Tidyverse

The pipe %>%

- From `magrittr` package (`https://magrittr.tidyverse.org`).
- Sends the output of the left-hand side function to the first argument of the right-hand side function.
- Simple example: `sum(1:8) %>% sqrt()`.
- Using the pipe, simple functions can be composed.

More complex example

```
Pima_diabetes %>%
  dplyr::select(Pregnancies, BMI) %>%
  dplyr::filter(Pregnancies > 10) %>%
  dplyr::summarize(avg_BMI_highP = mean(BMI), n = n())
```

Insert: Times and dates in R

- See `R_2c_dates_and_times_in_R.pdf`.
- Do exercise 4 in `R_2_exercises.Rmd`.

Homework

Homework

See file `R_2_homework.Rmd`

Questions?