Biostatistics & Epidemiological Data Analysis using R

1
Importing, checking & manipulating datasets in R

Stefan Konigorski

Health Intervention Analytics Group, HPI

October 28, 2021

HPI Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Learning objectives

- Get an overview about the main steps that are part of most data analyses in R.
- Learn how to load and import datasets.
- Learn some aspects what to look for, when checking if datasets have been imported correctly.
- Get an overview of different objects in R and how to work with them.

Main steps of a data analysis

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`.

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.

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.

Import, check, save datasets

Step 0: Input dataset

How would you input data collected from using the following questionnaire?

Aktion Saubere Hände

Hygiene Compliance Survey

ID _____ Station/ Department _____

Date _____

Profession	Indication	Action
☐ Physician	☐ Before visit	
	☐ Before asept	☐ Yes
☐ Nurse	☐ After inf	
	☐ After visit	☐ No
☐ Other	☐ After change	

Profession	Indication	Action
☐ Physician	☐ Before visit	
	☐ Before asept	☐ Yes
☐ Nurse	☐ After inf	
	☐ After visit	☐ No
☐ Other	☐ After change	

from `www.aktion-sauberehaende.de/fileadmin/ash/downloads/pdf/Compliance/Beobachtungsbogen_01.19.pdf`

Step 1: Import dataset

Functions to read external datasets into R

- `read.table()` to read text files.
- `read.csv()` to read csv files.

Step 1: Import dataset

Functions to read external datasets into R

- `read.table()` to read text files.
- `read.csv()` to read csv files.
- Equivalent to `read.table()`, but much faster/computationally efficient (for large datasets): `data.table::fread()`

Step 1: Import dataset

Functions to read external datasets into R

- `read.table()` to read text files.
- `read.csv()` to read csv files.
- Equivalent to `read.table()`, but much faster/computationally efficient (for large datasets): `data.table::fread()`
- `read_excel()` in `readxl` package to read xls and xlsx files.
- `spss.get()` in `Hmisc` package to read SPSS files.

Step 1: Import dataset

Functions to read external datasets into R

- `read.table()` to read text files.
- `read.csv()` to read csv files.
- Equivalent to `read.table()`, but much faster/computationally efficient (for large datasets): `data.table::fread()`
- `read_excel()` in `readxl` package to read xls and xlsx files.
- `spss.get()` in `Hmisc` package to read SPSS files.
- Example to import SPSS files: `R_1_example_import_SPSS.Rmd`.
- Alternatives: `readr::read_csv()`, `haven::read_sav()`, `foreign::read.spss()`, `haven::read_sas()`, ...
- ...

Step 1: Import dataset

Read files from

- Local directory, for example:

```
Pima_diabetes <-
read.csv(file = "C:/Users/stefan.konigorski/
                 Desktop/Pima_diabetes.csv")
```

- URL, for example:

```
read.csv(file = url("https://www.dropbox.com/s/
                     4s4rhf6abda6gw2/
                     Pima_diabetes.csv?dl=1"))
```

- Google sheets by using functions in the `googlesheets` package.
- ...

Step 1: Import dataset

Exercise 3

See `R_1_exercise_3.R`.

Step 1: Import dataset

Imported dataset:

RStudio Source Editor

Pima_diabetes

Filter

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	35	168	43.1	2.288	33	1
6	5	116	74	0	0	25.6	0.201	30	0

Step 1: Import dataset

Imported dataset:

RStudio Source Editor

Pima_diabetes

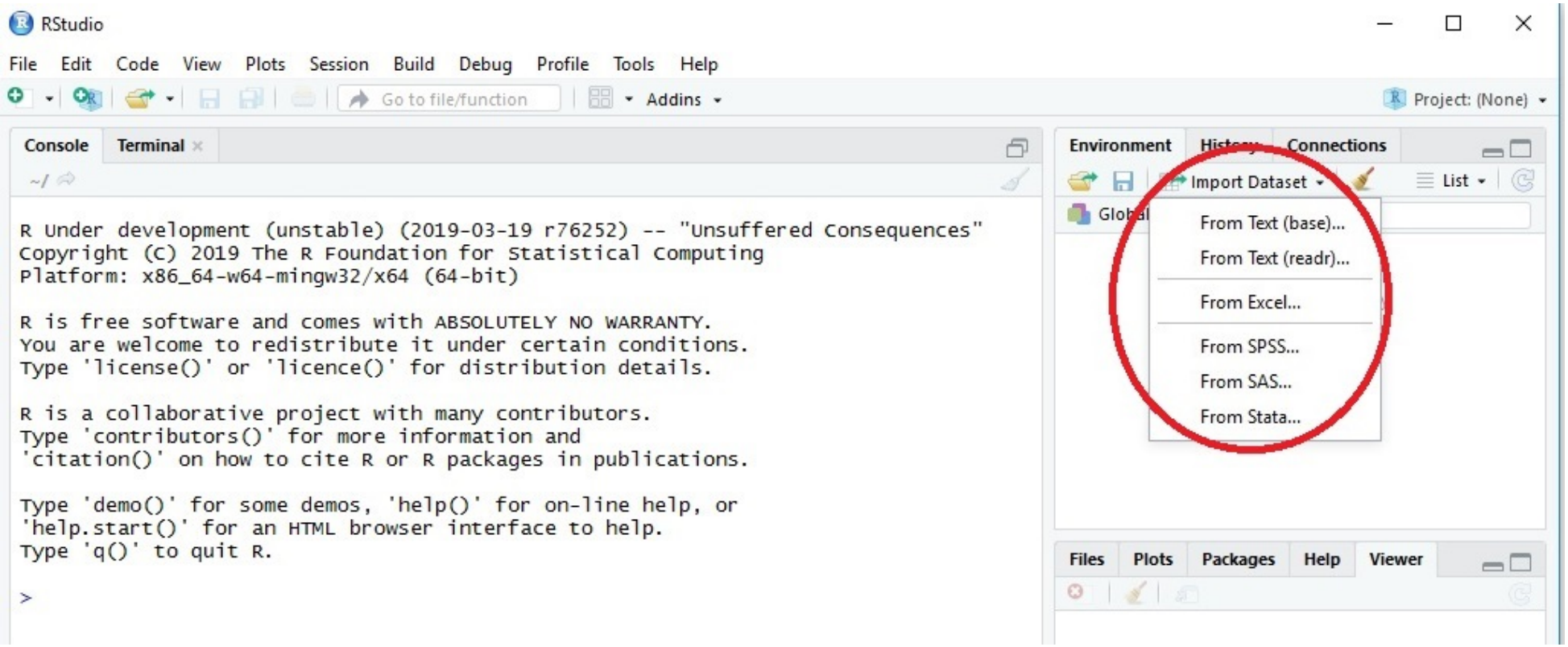
Filter

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	35	168	43.1	2.288	33	1
6	5	116	74	0	0	25.6	0.201	30	0

Inspect options how to display this dataset (sort, filter) - which don't change the data!

Step 1: Import dataset

Some of these functions to import datasets are also available using the graphical guide in RStudio:



Step 1: Import dataset

Important options when importing data

Default options in `read.csv()`:

- Does csv file contain header? `header = TRUE`
- How are data values separated? `sep = ","`
- How are decimal points described? `dec = "."`
- Are characters read as factors (default = yes)? `as.is, colClasses` options

Step 1: Import dataset

Important options when importing data

Default options in `read.csv()`:

- Does csv file contain header? `header = TRUE`
- How are data values separated? `sep = ","`
- How are decimal points described? `dec = "."`
- Are characters read as factors (default = yes)? `as.is, colClasses` options

Question: What is a factor? What kind of objects are there in R?

Insert: Objects in R

- Vector
- Matrix
- List
- Data frame

Insert: Objects in R - Vectors

Examples

- `c(1, 2, 3)`
- `c(1:5, NA, NA, NA, c(1, 2, 3), NA)`
- `Pima_diabetes$Pregnancies`
- `c("small", "big", "small")`
- `factor(c("small", "big", "small"))`
- `c(TRUE, TRUE, FALSE)`

Insert: Objects in R - Vectors

Examples

- `c(1, 2, 3)`
- `c(1:5, NA, NA, NA, c(1, 2, 3), NA)`
- `Pima_diabetes$Pregnancies`
- `c("small", "big", "small")`
- `factor(c("small", "big", "small"))`
- `c(TRUE, TRUE, FALSE)`

Assign values to object

Use `<-` to assign values to an object, e.g. `x <- 1:100`.

Insert: Objects in R - Vectors

Notes

- Types of vectors: numeric, character, logical.
- Can check by using the `is.numeric()`, `is.character()`, `is.logical()` functions, the `typeof()` function, or by looking in the environment panel in RStudio.
- All elements of a vector must have the same type. (What happens, if not?)

Insert: Objects in R - Vectors

Notes

- Types of vectors: numeric, character, logical.
- Can check by using the `is.numeric()`, `is.character()`, `is.logical()` functions, the `typeof()` function, or by looking in the environment panel in RStudio.
- All elements of a vector must have the same type. (What happens, if not?)

How to access elements of a vector

- Use `[.]` operator, for example:

Insert: Objects in R - Vectors

Notes

- Types of vectors: numeric, character, logical.
- Can check by using the `is.numeric()`, `is.character()`, `is.logical()` functions, the `typeof()` function, or by looking in the environment panel in RStudio.
- All elements of a vector must have the same type. (What happens, if not?)

How to access elements of a vector

- Use `[.]` operator, for example:
- `c(1, 2, 3)[2]`
- `c(1:5, NA, NA, NA, c(1, 2, 3), NA)[1:7]`
- `Pima_diabetes$Pregnancies[1:3]`

Insert: Objects in R - Lists

Examples

- `list(1, 2, 3)`
- `list(c(1, 2, 3))`
- `list(x = 1, y = 2, z = "small", zz = c(TRUE, TRUE, FALSE, FALSE))`

Insert: Objects in R - Lists

Examples

- `list(1, 2, 3)`
- `list(c(1, 2, 3))`
- `list(x = 1, y = 2, z = "small", zz = c(TRUE, TRUE, FALSE, FALSE))`

Notes

The elements of a list (e.g. `x`, `z`, `zz` in example above) can have different types and different lengths.

Insert: Objects in R - Lists

How to access elements of a list

- Use `[.]` or `[[.]]` operator, or access by name using the `$` operator. For example:

Insert: Objects in R - Lists

How to access elements of a list

- Use `[.]` or `[[.]]` operator, or access by name using the `$` operator. For example:
- `list(1, 2, 3)[1]`
- `list(1, 2, 3)[[1]]`
- `list(x = 1, y = 2, z = "small", zz = c(TRUE, TRUE, FALSE, FALSE))[4]`
- `list(x = 1, y = 2, z = "small", zz = c(TRUE, TRUE, FALSE, FALSE))[[4]]`
- `list(x = 1, y = 2, z = "small", zz = c(TRUE, TRUE, FALSE, FALSE))$zz`

Insert: Objects in R - Lists

How to access elements of a list

- Use `[.]` or `[[.]]` operator, or access by name using the `$` operator. For example:
- `list(1, 2, 3)[1]`
- `list(1, 2, 3)[[1]]`
- `list(x = 1, y = 2, z = "small", zz = c(TRUE, TRUE, FALSE, FALSE))[4]`
- `list(x = 1, y = 2, z = "small", zz = c(TRUE, TRUE, FALSE, FALSE))[[4]]`
- `list(x = 1, y = 2, z = "small", zz = c(TRUE, TRUE, FALSE, FALSE))$zz`

Difference between example 1 and 2? Try to add the number 1.

Insert: Objects in R - Data frames

Examples

- mtcars.
- The imported `Pima_diabetes` data frame.
- `data.frame(x = 1:4, y = c(TRUE, FALSE, TRUE, FALSE))`.

Insert: Objects in R - Data frames

Examples

- mtcars.
- The imported `Pima_diabetes` data frame.
- `data.frame(x = 1:4, y = c(TRUE, FALSE, TRUE, FALSE))`.

Notes

- Columns of data frames can have different types but must have the same length.
- tibbles: a modern take on data frames (`https://cran.r-project.org/web/packages/tibble/vignettes/tibble.html`).

Insert: Objects in R - Data frames

Further ways how to get info on R objects, e.g. data frames

- `str()`
- `summary()`

Insert: Objects in R - Data frames

Further ways how to get info on R objects, e.g. data frames

- `str()`
- `summary()`

How to access elements of a data frame

- `Pima_diabetes$Pregnancies`
- `Pima_diabetes[, 1]`
- `Pima_diabetes[1:2, ]`
- `Pima_diabetes[1:4, 1:4]`

Step 2 - Check imported dataset

- Check if dataset has been read correctly (not if the data is correct, do this later in step 4)[1] - with visual tests of raw data and imported data, and also "automatic checks" for larger data frames.

[1] Of course, these two check steps can also be combined.

Step 2 - Check imported dataset

- Check if dataset has been read correctly (not if the data is correct, do this later in step 4)[1] - with visual tests of raw data and imported data, and also "automatic checks" for larger data frames.
- Visually check first and last rows and first and last columns!
- Check if type of variable is correctly read it!
- Examples of automatic checks: sum all values in one column, use logical checks. ...
- Especially critical to check: dates, character strings (encoding?!), decimal numbers, missing values (-99?!)

[1] Of course, these two check steps can also be combined.

Step 3 - Save the checked imported dataset

Use `save()` function.

[2] See also "Files" tab in right lower panel.

Step 3 - Save the checked imported dataset

Use `save()` function.

Examples of `save()` function

- `save(Pima_diabetes, file = "Pima_diabetes_raw.RData")`
- `save(list(x = 1, y = c(TRUE, FALSE)), file = "list1.RData")`

[2] See also "Files" tab in right lower panel.

Step 3 - Save the checked imported dataset

Use `save()` function.

Examples of `save()` function

- `save(Pima_diabetes, file = "Pima_diabetes_raw.RData")`
- `save(list(x = 1, y = c(TRUE, FALSE)), file = "list1.RData")`

Where?

- Either without path as above (then it is saved in current workspace directory, can find where this is with `getwd()`, can change this with `setwd()`[2])
- or by filling in full path, e.g. `save(dat, file = "C:/Users/.../dat.RData")`

[2] See also "Files" tab in right lower panel.

Step 3 - Save and write

Further things to know:

- Objects can similarly be saved as .rda file.
- Also, single objects can be saved using the `saveRDS()` function.
- Save environment (= all elements in environment) using the `save.image()` function.
- R objects can also be exported, i.e. written to a csv file (using the `write.csv()` function) or xslx file (using the `writexl::write_xlsx()` or `xslx::write.xlsx()` functions).

Load R dataset

Once the data frame (or any other R object) has been saved as an R file (i.e. .RData file), it can simply be loaded with the `load()` function.

Homework

Homework

See file `R_1_homework.R`

Questions?