Chapter 1

Import and export of data sets

1.1 R data sets

To get a list of all available datasets provided by R, type `data()`. This data objects can be used for your own calculations and examples.

1.2 External data

Here we cover the basic for reading into R and exporting data from R. The most common and basic way to store tabular data in text format are *.csv* files. Each line in a .csv file represents a row of data, and single cells are separated by commas. Hence the name csv, comma seperated values.

1.2.1 Working directory

At all time, R internally points to a certain directory, called the *working directory*. If one wants to access a file in the filesystem from R, relative paths start from the working directory. It is convenient to put scripts with R code (ending in .R), and data files in the same directory and use that as the working directory. If you don't know what relative paths are and are interested, check out this short video `https://www.youtube.com/watch?v=ephId3mYu9o`.

1.2.2 CSV import

To read a .csv file in R use the *read.csv()* function. Here we read the *BackPain.csv* file in the data folder. Make sure that you set R's working directory to the same location as the BackPain.csv file. This can be done in RStudio with the Menu: Session → Set Working Directory → To source File Location. The following code reads the BackPain.csv and stores the data in the *backpain_data* object.

```
backpain_data <- read.csv("../data/BackPain.csv")
```

Saving tabular data in a .csv file works similarly. To save the same data with another filename:

```
write.csv(backpain_data, "../data/tmp_DifferentName.csv")
```

1.2.3 Excel

Its possible to directly work with excel files, but this requires to load an extension, called a *package*. One established package that can read and write xlsx file is called *openxlsx*. After installation, excel files can be accessed. Saving tabular data in a .csv file works similarly.

Another established package is *readxl*. It is part of the tidyverse and does a good job at reading excel files, but cannot write them.

The following code shows how to read and save the same data with another filename using openxlsx and readxl:

```
library(readxl)
library(openxlsx)

# read first sheet - readingxl
packpain_data <- read_excel("../data/BackPain.xlsx", sheet = 1)

# write again - openxlsx
write.xlsx(x = packpain_data, file  = "../data/tmp_DifferentName.xlsx")
```

1.3 Import and export file formats from other statistical software

Its possible to import datafiles from other statistical softwrae packages. One good package to use is *haven*, it may need to be installed. This package supports for example the *Stata* and *SPSS* formats:

```
library(haven)
aids <- read_dta("../data/Aids.dta") # read stata aids data
write_dta(aids, "../data/tmp_Aids.dta") # write stata

norsjo <- read_sav("../data/norsjo86.sav") # read spss save file
write_sav(norsjo, "../data/tmp_norsoj86") # write spss save
```

1.4 Import export using RStudio

R Studio provides a way to automatically generate the code for importing data from various file formats via FIle->Import Dataset. One can conveniently use a graphical file browser to go to select the file and import the data directly into the working environment. It works by pasting the code to do that into the console and executing it.

Own experimentation

What are the advantages uf using the R Studio menu, what are the disadvantages? If you publish an R script as part of a scientific article, Would you chose to include the code for importing the data as part of the script?

1.5 R binary files

Sometimes it is convenient to store R objects directly in binary format, without using one of the text-based formats such as *csv*. Binary means that the file format can not be edited with text editors and is harder to read for humans. Instead, the data is stored in a format that is related to how the data looks internally in the computer memory. Note that many file formats - regardless of whether they are text based or not - compress the data. For example, compression is the reason why excel files are smaller than .csv files with the same content, even if both are text based. Excel uses an xml, text-based file format and zip compression.

R provides several ways of saving to a binary format. One way is using `save()`. This function stores the specified objects in an *.RData* file.

```
data <- cars
var1 <- 42

# objects
save(data,var1, file = "../data/tmp_current_environment.RData")

# specify list of object character names
save(list = c("data", "var1"), file = "../data/tmp_current_environment.RData")

# directly specify the objects
save(data, var1, file = "../data/tmp_current_environment.RData")
```

The counterpart to `save()` is (load()). It can be used to load the *.RData* files back into R. Note that all the objects in such a workspace image file will autmatically be added as objects to the current working environment.

```
load(file = "../data/tmp_current_environment.RData")
```

Another way to load and save data is using the (saveRDS()) and `readRDS()` function. They operate on single objects, meaning that they save and read single R objects:

```
# save and load single objects
saveRDS(mtcars, file = "../data/tmp_data.rds")
readRDS(file = "../data/tmp_data.rds")

##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

This offers some advantage over `save()` and `load()`. Associating objects explicitly to filenames makes scripts easier to understand, write, and enhances reproducibility. When saving and loading whole working environments, it may not be clear what data has been loaded with the working environment and what data has been loaded prior to that using other means.

While binary formats in R generally results in very small files and are easy to load and save, using them can create problems when trying to load data with other software or different R versions.