Introduction to R

Getting Data into R

Patrick Toche[1]

[1]contact@patricktoche.com
https://github.com/ptoche/

May 2017

Getting Data Into R

- Datasets come in different formats, e.g. `.csv`, `.xls`, `.xlsx`, `.dta`, `.tsv`.
- Whether you intend to share the dataset or to analyze it with R, a good choice of format is "comma-separated-values (csv)" because it may easily be read by other software (including spreadsheet software) and because it results in small files. Tab-separated values (tsv) is an alternative that is also easily portable.
- Other formats, such as `.xlsx` and `.xls` (Excel), `.dta` (Stata), `.sas7bdat` and `.xpt` (SAS), `.sav` (SPSS), `.mat` (Matlab) or `.RData` and `.rda` (R) may carry more information (such as the type of variable, e.g. `int` or `char`) or may be compressed or even encrypted.
- For most purposes `.csv` is a good choice.

Getting Data Into R

- For most purposes `.csv` is a good choice.
- `.xls` and `.xlsx` are also very popular formats.
- In most cases, importing data from these formats is straightforward.
- In some cases, the spreadsheets may contain several sheets of data or non-standard characters, e.g. Chinese characters or invisible characters inadvertently inserted by non-standard inputting methods. In some cases, the datasets may even appear to be corrupted. There are dedicated packages to handle complicated situations.

Getting Data Into R

- Create a simple spreadsheet with LibreOffice or another software. Save it in the `.xls` format.
- In RStudio, under the `Environment` tab, you will see a `Import Dataset` command. This is equivalent to typing the following commands in your console:

```
library(readxl)
data_1 <- read_excel("path/to/dataset/data-1.xls")
View(data_1)
```

- With simple datasets and for a one-time use, it is often simpler to manually edit the dataset in your spreadsheet software. But with large datasets and/or datasets that will get updated over time, you may want to automate the process of downloading from an online repository, loading the data into R , editing. For this purpose, there are more powerful packages than `readxl`

Getting Data Into R

- Save the spreadsheet as a `.csv`. Make sure to select the appropriate options: The `Field Delimiter` should be set to the comma `,` and the `Text Delimiter` should be set to the double-quote `''`. Also select to save the data "as shown" rather than saving the formulas.
- Before saving, adjust the display precision to the data precision, otherwise some data will be lost. If your data is precise to the 5th decimal, make sure that all 5 decimals are displayed before saving.

```
library(readr)
data_1_csv <- read_excel("path/to/dataset/data-1.csv")
```

- Notice that I have named the dataset differently to make sure both datasets are loaded.

- You can compare the datasets with `str(data_1)` and `str(data_1_csv)` or by viewing the data into the console. This reveals that the variables were imported in different formats:

```
typeof(data_1$x)
## [1] "character"
typeof(data_1_csv$x)
## [1] "double"
```

- The `readxl` package imported our `.xls` dataset as characters, while the `readr` package imported our `.csv` dataset as integers. In this instance, the choice made by the `readr` package was a better one.
- We can change the data to the desired type, e.g.

```
data_1$x <- as.numeric(data_1$x)
data_1$y <- as.integer(data_1$y)
```