BootCamp 3: Data import

Lucien Baumgartner

10/3/2018

This is a short one.

data.table::fread() vs. base::read.csv()

Since we work with very big data sets encoded as CSV (comma separated values), we need some fast and efficient datahandling.

Base R has its own function to import data in .csv-format, namely `read.csv()`.

```
# first we set our working directory
setwd('~/ddj18-data/st-zh/Daten/')
```

```
bev <- read.csv('bevoelkerung.csv',
                stringsAsFactors = F)
```

Importing data with that function is waaaaaay slower than using `fread()` from the package `data.table` which provides a special dataframe-structure for powerful dataransformation operations. We will only use the aforementioned function, though.

To use the function, you need to install the package, if you haven't already, and load the library beforehand. Loading a library is equivalent to adding a book (yes, a real book) to your library at home in order for you to use all the knowldege stored in it (here: all the functions stored in it). Loading a library means loading functions to your workspace. I'll use the dataset `bevoelkerung.csv` as an example case, since it is the biggest dataset Statistik Zürich has provided you with.

```
install.packages('data.table')
```

```
# load the library
library(data.table)

# time used to import data via base function
system.time(
  bev <- read.csv('bevoelkerung.csv',
                  stringsAsFactors = F)
)
```

```
##    user  system elapsed
##  64.091   3.531  68.735
```

```
# time used to import data via fread
system.time(
  bev <- fread('bevoelkerung.csv')
  )
```

```
##    user  system elapsed
##  18.427   1.590   4.385
```

While `read.csv` needs over one minute (elpased time), `fread`, on the other hand, only 4s (NOTE:: I'm running a selenium scraper while compiling this script, so the import time jumps up to roughly 20s). Now that you have assigned the data to the object `bev`, you can use it to crunch some data.

`system.time()` is only used to show you how long an evaluation takes, you don't need to include that in your script.

Make sure to set the paramater `stringsAsFactors` to `FALSE` in `read.csv`, since `TRUE` is the default (in `fread`, from v 1.9.6+, the default is `FALSE`). Why? Because we want total control over the data. We *always* create perform class transformations ourselves, except for very special cases. If you work with factors, some operations will just fail or create NAs, while they work with characters. I will show some examples in another script.

If you get this error btw, you just set the path wrongly:

```
bev <- read.csv('wrong/path/bevoelkerung.csv',
                stringsAsFactors = F)
```

```
## Warning in file(file, "rt"): cannot open file 'wrong/path/
## bevoelkerung.csv': No such file or directory
```

```
## Error in file(file, "rt"): cannot open the connection
```