W05-1: Melting a data set

This worksheet will demand your competence in using loop functions to perform computations on a data frame. Especially the lapply function will need more training than this worksheet can provide but after completing it you should feel able to implement an lapply and for construction at least if you have some assisting tips.

Things you need for this worksheet

- R — the interpreter can be installed on any operation system. For Linux, you should use the r-cran packages supplied for your Linux distribution. If you use Ubuntu, this is one of many starting points. If you use windows, you could install R from the official CRAN web page.
- R Studio — we recommend to use R Studio for (interactive) programming with R. You can download R Studio from the official web page.
- your world bank data sets from W01-1: Getting and organizing data

Learning log assignments

As always, please add these entries to your today's learning log at teachwiki:

- Favorite aspect of the session (if any)
- Superfluous aspect of the session (if any)
- Eureka effect (if any)
- Links to what I've learned so far (if any)
- Questions (if any)

For more information see this short howto.

As today's special, please complete the following assignment:

As part of W01-1: Getting and organizing data you have downloaded two world bank data sets, one giving the GNI, another one the CO_2 emissions on a country level since 1960.

At the end of this worksheet you will find a section of code which will be the basis for today's assignment.

The script starts with the `list.files` function which gets you a list of all files matching "wb*2013.csv" in the present working directory. The pattern matches our naming pattern of the world bank data set and in your case it might be necessary to adapt it.

The second block of the script shows you an `lapply` example on how you can read one file after another and perform a conversion from wide (i.e. one column per year) to long format (i.e. one row per year per country) using the `melt` function of the `reshape2` library. Both world bank data sets are subsequently combined into one data frame.

For the first task, imagine that there is not `melt` function and of course you should also not search for

alternative functions. Instead do the following:

Please write an R script in which the melt-section of the given code is replaced by another `lapply` loop. The resulting data frame should have the same structure as the one resulting from the `melt` function.

As a result of this task, the structure of your combined data frame will look like this:

```
'data.frame': 13932 obs. of  7 variables:
 $ Country.Code    : Factor w/ 258 levels "ABW","AFG","AGO",..: 1 2 3 4 5 6
7 8 9 10 ...
 $ Year            : Factor w/ 54 levels "1960","1961",..: 1 1 1 1 1 1 1 1 1
1 ...
 $ Country.Name    : Factor w/ 258 levels "Afghanistan",..: 12 1 7 2 6 5 9
244 10 11 ...
 $ Indicator.Name.x: Factor w/ 1 level "CO<sub>2</sub> emissions (kt)": 1 1
1 1 1 1 1 1 1 1 ...
 $ CO<sub>2</sub>             : num  NA NA NA NA NA NA NA NA NA NA ...
 $ Indicator.Name.y: Factor w/ 1 level "GNI per capita, Atlas method
(current US$)": 1 1 1 1 1 1 1 1 1 1 ...
 $ gni             : num  NA NA NA NA NA NA NA NA NA NA ...
```

The `rep` function which allows the replication of elements in a vector (or data frame column) might be quite helpful for this task. Aside from that, the logic of the loop is similar to this.

Let's pretend that someone is interest in the correlation between CO_2 emissions and GNI on a country level.

Please extend your R script by a for loop which handles the computation of the correlation between both world bank indicators on a country level. The for loop should directly alter a data frame which is create prior to the execution of the for loop and which has three columns: Country.Name, Country.Code and Correlation.

The `unique` function will be quite helpful for the initialization of the new data frame (especially for the Country.x columns). The Correlation column can just be initialized with NAs.

Since the `cor` function would cause an error if not a single complete case is present (i.e. not a single overlapping year for which the CO_2 and the GNI value of a specific country is given) you have to add `use = "pairwise.complete.obs"` to the call of the function.

Please upload your final script to your learning log.

"Snippet 01"

```
library(reshape2)

#### Load and merge world bank data sets
```

```
########################################
# Load world bank data sets by scanning files in a directory, convert
them from
# long to wide format and combine both data sets by merging.
files <- list.files("data_raw", pattern = glob2rx("wb*2013.csv"),
                    full.names = TRUE)

wb <- lapply(files, function(x){
  df <- read.table(x, header = TRUE, sep = ",", skip = 2)

  # Start of melt-structure: the following section should be replaced
  ldf <- melt(df, id.vars = 1:3, measure.vars = 5:58,
              variable.name = "Year",
              value.name = substr(basename(x), regexpr("_",
basename(x)) + 1,
                                  regexpr("_", basename(x)) + 3))
  ldf$Year <- substr(ldf$Year, 2, 5)
  # End of of melt-structure

  return(ldf)
})
wb <- merge(wb[[1]], wb[[2]][,-1], by = c("Country.Code", "Year"))
wb <- wb[order(c(wb$Country, wb$Year)),]
str(wb)
```