03_Data_Preparation

Aleksandra Kudaeva

28 Februar 2019

Preparation of Air-Pollution Data

Preparation of air-pollution data required obtaining additional data and usage of several approximations. The whole process of data preparation can be devided in three parts: 1) downloading the data from different sources 2) data cleaning and reformating 3) merging and aggregating the data

Loading the data

Air-pollution data was downloaded from [*insert reference: "https://fbinter.stadt-berlin.de/fb/index.jsp%22*] in xls format. Initial column names are poorly adapted for R processing (e.g. "PM10-Belastung (berechnetes Jahresmittel [µg/m³]) 2015") as they are too long and contain special symbols. For the purpose of further analysis we rename all the columns according to the correspondence stated in a matching table (matching.csv)

```
# Read air-pollution data from excel
ap15 = read_excel("./SPL_BerlinDst_Data_Prep_3/Air_Pollution_2015.xls",
                  sheet = 1)

# Original names are too long and contain special symbols and spaces
mtch = read.csv2("./SPL_BerlinDst_Data_Prep_3/matching.csv",
                 sep = ";",
                 stringsAsFactors = FALSE)  # Matching table for short names

names(ap15) = mtch$new[match(names(ap15), mtch$old)]  # Rename variables
```

Downloaded table consists of 12374 rows containing values of different air-pollution indicators and aggregated index values for section of main Berlin Streets (1238). Average value of PM10 pollution is equal to 20.6 mg/mˆ3, and PM25 - 14,3 (on average in 2015), which is below dangerous threshold. Statistics were calculated by means of the script provided below:

```
# descriptory statistics of air pollution data table
ap15 %>% summarize(Sections = n(),                 # Number of street sections
                   Streets  = n_distinct(Street),  # Number of unique streets
                   avg_PM10 = mean(PM10_yearly),   # Average value of PM10
                   avg_PM25 = mean(PM25_yearly))   # Average value of PM25
```

The raw data does not contain district key. That is why, in order to bring Air-Pollution data to the same format as other variables described in previous chapters and merge tables we need to download additional information. Our solution was to scrap correspondence table from web-page [*insert reference: kauperts*]:

```
for (i in 1:dim(dstr)[1]) {
    # Generate a link to data for all the districts and sub-districts
    link=paste0("https://berlin.kauperts.de/Bezirke/",
                dstr$District[i],
                "/Ortsteile/",
```

```
                dstr$Sub.district[i],
                "/Strassen")

    # Download the data from the web-page with generated link
    webpage = read_html(link)
    tbls    = html_nodes(webpage, "table")
    tab     = html_table(tbls)[[1]]

    # Add columns for district and sub-district
    tab$District    = dstr$District[i]
    tab$SubDistrict = dstr$Sub.district[i]

    # Add created table to the table for the whole Berlin
    StrMtch = rbind(StrMtch, tab)
}
```

Downloaded table contains information on XXX streets from 12 districts of Berlin.

Reformating and cleaning

Another problem was different ways of writing street name, e.g. Adamstr./ Africanische Str. and so on. First step to unification of street names was replacements of german special symbols and bringing everything to the low case. For that purpose the function *ReplaceUmlauts* was written:

```
#replace all the Umlauts by latin equivalents
ReplaceUmlauts = function(clmn){
    #Description: Replaces german special symbols and turns to lower case
    #Author: Aleksandra Kudaeva
    #Input:  column where you want to replace umlauts
    #Output: column without umlauts (lower case)

    clmn = tolower(clmn)  #all strings to lower case

    #check if at least one element of a vector has any umlauts in it
    #replaces umlauts until there are no one left
    while(any(grepl("ä|ö|ü|ß",clmn)) == TRUE) {
        clmn %<>%
            sub("ä", "ae", .) %<>%
            sub("ö", "oe", .) %<>%
            sub("ü", "ue", .) %<>%
            sub("ß", "ss", .)
    }
    return(clmn)
}
```

Next step, was to substract unique part of street name (e.g. Adamstr. - Adam). THe following code illustrates work of replacement function and substitution procedure for air-pollution dataset (analogical procedure was performed for correspondence table):

```
# Street name formatting (in order to merge with street-index matching table)
ap15$str = ap15$Street %>%
    ReplaceUmlauts() %>%  # Replace umlauts and switch to lower case
```

```
    sub("str.$|str$|-strasse|strasse|-str.$", "", .) %>%  # Delete street ind.
    sub("ak |as |ad ", "", .)  # Delete AK, AS, AD in the beginning
```

Merging and summarizing