Project Worksheet Day 3

August 25, 2021

Now that we have discussed in the lecture how to deal with files and data frames, you are ready to read in your data and start processing it! Your project assignment for today is to read the data you need from your input files in a format that is suitable for further processing (in most cases, pandas dataframes will be most adequate) and to implement one first, simple data analysis function as a sort of "teaser" of what the program will do. This will also help you getting more familiar with your data set and how to work with it.

Tomorrow you will work on the code for the more complex analyses that you have planned. This is good practice generally: When you want to program something, create a simple but working example first, and extend it from there. Starting with too many details right away will only make it difficult to get it working.

Teaser Implementation

For the teaser for our running example on *"ICT kennis en vaardigheden"*, we might implement a function `load_data_from_file()` in the `data_handling.py` module as follows (you are welcome to reuse and adapt this code):

```
# module containing different data handling functions for CBS data sets

# required imports
import pandas as pd

# function for loading CBS data from a csv file
def load_data_from_file(file):

    # read content into dataframe
    df = pd.read_csv(file, sep=";")

    # trim whitespaces from all strings in the dataframe
    df_obj = df.select_dtypes(['object'])
    df[df_obj.columns] = df_obj.apply(lambda x: x.str.strip())

    # return the dataframe
    return df
```

The statement `df_obj.apply(lambda x: x.str.strip())` is used to remove surrounding white spaces from the string-typed entries in the data frame. We will discuss these "lambda functions" in more detail in one of the next lectures, for now you can simply use it.

We can then implement a function for calculating correlation for one specific case:

```
# demo function for calculating the correlation regarding two specific
skills between a subset of different age groups in the year 2018
```

```
def make_demo_correlation(cbsdata):

    # set parameters for the demo

    period = "2018JJ00"

    skill1 = "GebruikVanSpecifiekeSpreadsheets_39"

    skill2 = "GebruikVanSpreadsheetSoftware_36"

    #subset data frame to get the values for the requested year

    cbsdata_2018 = cbsdata[(cbsdata["Perioden"]==period) &
(cbsdata["Marges"]=="MW00000")]

    correlation = cbsdata_2018[skill1].corr(cbsdata_2018[skill2])

    return correlation
```

Of course, we need to call this function from the main program. That can be done as follows:

```
# import the other self-defined modules
import data_handling as dh

# for simplicity, set path to input data file here (must be in the same
directory)
inputfile = "83428NED_UntypedDataSet_06022019_212509.csv"

# load CBS data set and metadata into separate data frames
cbsdata = dh.load_data_from_file(inputfile)

# print message about what is going to happen
print("Calculate correlations of skills in using presentation "\
      "software among different age groups in 2018.")

# call function to make demo plot
correlation = make_demo_correlation(cbsdata)

# print correlations
print(correlation)
```

Output:

```
Calculate correlations of skills in using presentation software among
different age groups in 2018.
```