Exercise class 6

(week 13)

Introduction to Programming and Numerical Analysis

Class 4 and 8

Rosa Haslund Meyer

Spring 2024

KØBENHAVNS UNIVERSITET

Take-away's form this week's lectures

API's

Tips and comments on problem set 4

Problem set 4. Analysing data

Take-away's from this week's lectures

The lectures introduced you to different tools used when working with data:

- Importing/fetching data from **API's** using provided Python packages
- Combining datasets using **`pd.merge()`** and **`.join()`**
- Transforming data:
  - The split-apply-combine process:
    1. Split: divide the dataset into different units (in this case one for each municipality)
    2. Apply: compute, for example, the average employment rate of each unit (transform data)
    3. Combine: merge this new variable back onto the original dataset
  - `pd.df.apply()` method: Applies a function along an axis of the DataFrame
  - `pd.df.agg()` method: Aggregates using one or more operations over a specified axis (rows or columns)
  - `pd.df.transform()` method: Calls a function, a string, a list or a dict on df producing a DataFrame with same axis shape as df

Short on API's

An **application programming interface** (API) is a set of specific rules that lets computer programs communicate with each other to exchange data.

In this case think of the API as a communication line between your software (Python) and Statistic Denmark's database.

You can imagine API as a waiter in a restaurant. You are seated at a table and a waiter called API comes to you with a menu. After placing your order, the waiter API passes it to the kitchen for further processing. When the order is ready, the waiter returns to you with your order.

Tip for working with API's

The API's used during this course has **associated Python packages** that connects to the API and pulls and parses data for you. For example, in today's problem set you'll use **DstApi** for Statistics Denmark' database. Another example is the **Pandas Datareader** which you can check here.

However, not all API's has associated Python packages. For those API's which don't, you'll need the `requests`-library installed. This is not part of the course per se, but if you're interested check out this webpage or the DataCamp course *Intermediate - Importing Data in Python*.

When using API's always make sure to check documentation!

Problem set 4

If you didn't finish **problem set 3** last week, you should do so before working on problem set 4. You can refer to the uploaded slides from last week for notes and comments on a bug.

In problem set 4 you'll be fetching data from Statistics Denmark's database using **DstApi** – which you firstly must install (refer to first cell in notebook). The documentation for the API can be found here. If at any time you are unsure about which arguments to pass, for example for the `.get_data()`-method, see **documentation** for help.

As with last week's problem set, Pandas syntax is quite heavy, so feel free to look at answers but make sure to experiment and understand what's going on!

A few tips

Task 1:

Use the hints already in the cell. For example, when they ask you to download/fetch the table 'nah1', that's simply what you should do.

'nah1' is a table from Statistics Denmark's database, so you can't access it before fetching it using the API.

In step 3, you'll probably write a function to keep certain rows. Note the difference between the 'or' operator and the '|' operator. I wrote a comment on this in my uploaded solution guide here.

Task 2:

Google at documentation for the `pd.merge()-method`. I made a comment on the 'how' parameter in my uploaded notebook. If you don't already know SQL, make sure you understand this parameter!

A few tips

Task 3:

Look at the tips in the cells one line at a time. Ask yourself what each line is supposed to do. The first thing you should do is set the index to 'year' instead of '0,1,2…' – what do you expect to happen when doing this?

Task 4:

Start by printing the head of the table 'nah1' from the split-apply-combine approach. You should be able to get a table with the same variables and data, except you should get an extra column named 'index_transform'.

Problem:

For the problem, look at the graphs closely before you start programming! Start by merging, taking logs and then plot. Remember that you can always print the table!

Time for exercises

Problem set 4:

- Task 1: Importing and cleaning data
- Task 2-3: Merging using the `pd.merge()` and join method
- Task 4: Split-apply-combine-plot using `.transform()`
- Problem: The Housing Market
  1. Understanding ALL commands: Loading/importing and cleaning data
  2. Analysis: Reproducing graphs – merging, taking logs and plotting

Next time...

Video lectures:

- No videos this week!
- ... lots of time to catch up on things missed

Exercises – Data project:

- Introduction to data project
- Work on project