Exercise class 8

Introduction to Programming and Numerical Analysis

Class 3

Annasofie Marckstrøm Olesen

Spring 2023

UNIVERSITY OF COPENHAGEN

Take-aways from this week's lectures

Problem set 4

Intro to Data Project

This week's letures

This week you've seen a bunch of tools for working with data:

- Fetching data from **API**'s using provided Python-packages.
- Combining datasets using `pd.merge` and/or `.join`
- Transforming to data using the **split-apply-combine** approach, and `.apply`, `.transform` and `.agg` methods

Combined with your knowledge from previous lectures on cleaning data as well as presenting results in plots and tables, you have a pretty toolbox for empirical work in Python.

Tips for working with API's

An API (Application Programming Interface) is a **communication line** between your software (Python) and some other software (ie. Statistics Denmark's database).

Some API's have associated **Python packages** that can connect to the API and pull and parse data for you (ie. pydst for Statistics Denmark or pandas-datareader).

Otherwise, you need the `requests`-library. Not a mandatory part of this course, but check out this blog post or the DataCamp course *Intermediate Importing Data in Python* if you're interested.

In any case, make sure to **check out the documentation** for the API you're using.

Problem set 4

If you didn't get through problem set 3 last time, you should do that first. See last week's slides for notes on the problem set + a bug.

In problem set 4, you will be fetching data from Statistics Denmark using pydst. Documentation is here. If you are unsure about which arguments to pass to the `.get_data()`-method, see documentation for help.

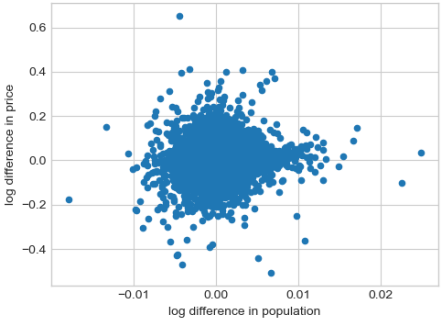
Working with Pandas can be very syntax-heavy, so it is okay to glance at the answers - but make sure you understand the syntax.

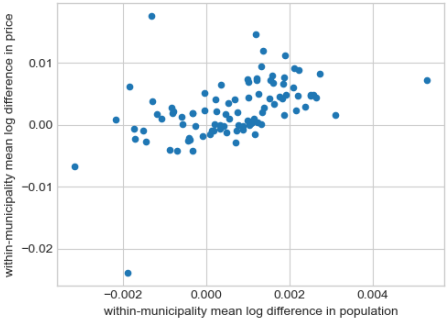
Plan

- Now-16.00: Work on tasks
- 16.00-16.15: Break
- 16.15-16.40: Work on problem
- 16.40 - 17.00: Finish together + Tips for data project

TIP THAT IS NOT IN PROBLEM SET 4:
If you just download the data directly from the API, it is not completely sorted by years and variables - make sure you **sort the data correctly**, otherwise the plots produced by the code will be unreadable.

Plots for the problem





Next time

Video lectures

- No video lectures

Exercise:

- Work on data project

A few words on the data project

Objective: Download and clean data then do some empirical analysis - but how, what and why is entirely up to you!

Choose something interesting but manageable. Since each project is different, the possibility to copy code from lectures will be limited.

You can get data from API's or by downloading manually to ie. a CSV-file.

For the analysis, focus on presenting your data in a nice way, ie. through a pretty figure or table. Think about the point you want to get across - how can you best illustrate that?

Inspiration for data

- Statistics Denmark, or the package pydst (see lecture on fetching data)
- Pandas-datareader can access many data sources, including Federal Reserve, NASDAQ, World Bank, Yahoo Finance etc. (see lecture on fetching data)
- Our World in Data, or the package owid-catalog
- Understat (European Football Leagues)
- IMDB-data and the Cinemagoer package
- FiveThirtyEight hosts all code and data on their GitHub
- This list of publicly available API's (You may need to interact with the API directly instead of using a package.)

You can also see previous years' data projects on the course GitHub by searching ie. "projects-2022".