



UNIVERSITY OF
COPENHAGEN

Eighth exercise class

Introduction to numerical programming and analysis

Jonathan Wenzel Pedersen

Spring 2023

1. Plan for today

2. Hints

Plan for today

What are we doing today?

Today we will be working with Problem set 4, which covers pulling down data from Denmark Statistics, cleaning and handling that data (including merging) and finally using that data.

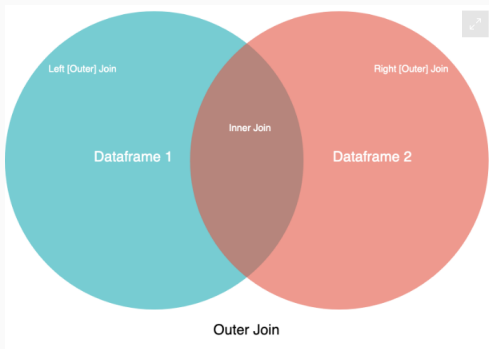
Hints

Downloading, tips

In problem 1 you have to download the data using pydst, pydst is a package that allows us to connect to Denmarks Statistics, it can be downloaded using the command *pip install git+https://github.com/Kristianuruplarsen/pydst.git*, if that does not work then clone the github and move the file PyDST into the problem set 4 folder (this will solve the problem).

Merge (Problem 2 and 3), tips

Merging can be done using the `.merge()` or `.join()` command, `.merge()` can be joined using inner, outer, left or right as described below. Remember for `.merge()` we use the command `on=""` to specify the column (or columns) which we merge on, while `.join()` uses the index so if you merge you need to change these beforehand



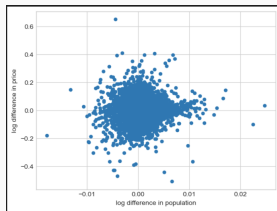
Problem 4, tips

Read through Jeppes code and try to understand it first, consider writing in notes what each line does. If your unsure what a line does either google it, run it or ask me.

Once you understand his code try to implement first three steps i.e. a, b and c using the *.transform()* method.

Problem 5, tips

Your result should look like this:



Remember that co-variation is the relationship between two values when it comes to relative changes i.e. how does a increase (or decrease) in one variable affect the other?

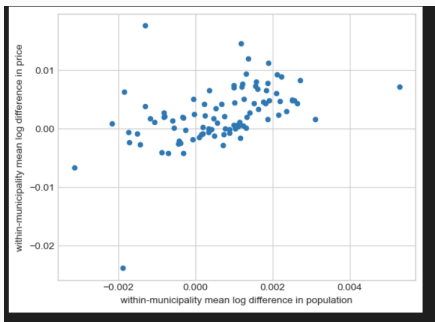
To solve this problem you need to do the following:

- Combine the two data-sets (you can use merge or join for this)
- Get the relative change, remember that

$$\ln(y_t) - \ln(y_{t-1}) \approx \frac{y_t - y_{t-1}}{y_{t-1}}, \text{ so consider using log and difference}$$

Problem 6, tips

Your result should look like this:



You can reuse most of the code from problem 6, this difference is that you want to take the mean for each municipality, try to do it in two steps using Groupby (Jeppes answer is done in