Seventh exercise class

Class 5

Introduction to numerical programming and analysis

Asker Nygaard Christensen

Spring 2021

Plan

1. Inaugural project
2. Next assignment -repeated from last time
3. Problem set 4, Q2.1-Q2.3
4. Q3.1-Q3.2 will be done together
5. Q3.3

Inaugural project

Inaugural project

Inaugural project

By now you should all have received feed-back.

I was generally really impressed by your projects. There weren't any glaring recurring mistakes, but I have made a small notebook, representing my most recurring comments.

Next assignment -repeated from last time

Data Analysis Project -repeated

It might already be a good idea, to start thinking about what you wanna do and what kinda data you wanna use in the next assignment

You can see the assignment at Github, but basically you have to download data from an online source and do some empirical analysis (key figures and graphs). You get to choose the data and what kind of analysis you wanna do yourself. There are LOADS of possibilities, so choose something you find interesting and seems manageable. Recreating a already existing figure using python is also good.

You can see all the previous projects by searching 'projects-2019' and 'projects-2020' at the NumEconCopenhagen github account.

Also if you wanna do some regressions you can use statsmodels, but regressions are not a prerequisite for doing the assignment, a beautiful figure is just as good.

Since these projects are idiosyncratic, you'll find less directly applicable code in the lectures and PS. But they are still good for inspiriation, along with googling your problems.

Ideas -repeated

- Investegating the home-field advantage using Covid-variation. This should be able to the job. But you can also download excel sheets from here (Inspiration from this paper)
- Twitter data (See for example This guide
- Google trends using pytrend
- You recreate graphs from 'Capital in the 21st century' and 'Captial & Ideology', either from QUANDLE or Piketty's personal website
- World Inequality Database
- Pandas_datareader, from L08 has the capability to load from multiple sources, including the World Bank, OECD and stock data.
- Denmarks statistics can be accessed using pydst, also L08
- There is even a python package for downloading IMDB data (A group actually used IMDB data last year, Credible threats, although they downloaded the data manually)

My tips on Pandas and data science generally -repeated

Data science can be an excruciating job. Because you're doing self-chosen projects, you won't be able to rely as much on the lecture and PS. Remember to reach out when you're having problems.

There are loads of python guides and stack-flow answers on the internet, so the right google search can also be your saviour.

You been introduced to many different ways of referencing data. I'd recommend using *.loc[I,columns]* mostly in the beginning, as it is the most versatile. Instead of the conditions implicit in *I*, you can also use a list of index-numbers.

When creating Boolean condition, I'd also recommend being explicit with your brackets. Also, remember '&' is the bit-wise 'and' operator, and '|' is the bit-wise 'or' operator (Depending on your keyboard is could be AltGr+'the key with | on it' (close to backspace), or Alt+i), also called pipe or vertical line.

Problem set 4, Q2.1-Q2.3

My notes on problem set 3 (also, see lecture 8)

Q 2.1 Import national account data from Denmark Statistics

See section 2.1 in lecture 8

In step 2 and 4, the '#'-hint uses *nah1_true*, you only need to write *nah1* (because that's the name you give the dataset in step 1).

Remember to look at the data between steps, to make sure your operations are doing what you expect them to do.

Step 2 For the first line, maybe have a look at the documentation of *.rename()*, and note that you're renaming the columns

Step 3 First notice that *[Y, C, G, I, X, M]* is the values of *var_dict*, so you do not need to make a new list

The answers creates the condition for a row to be kept, by looping through the elements and using the '|' (or)-operator. An easier approach is simply to use the *isin()*-method:

I = nah1.variable.isin(var_dict.values())

Step 4 Run the answer cell again, such that max year is 2020 (all DST Qs)

My notes on problem set 3

Q 2.2 **Merge**. See section 2.2 of lecture 8 on merging. Remember that joining is like merging, but on the index.

Q 2.3 **Split-apply-combine** can be a bit tricky, so have a look at section 4.2 in the lecture, and don't be afraid to peek at the answer.
Also, it's *nah1* not *nah1_final*.
The question mark in *h1_alt[?]*, is just what you want to call the column where your storing your data, the answers calls this *index_transform*.

Q3.1-Q3.2 will be done together

Q3.3

Q3.3

I think it is intentional there that is less help on the last question, so I won't write as explicit hints for this one.

But try to first think of the steps you need to go through: Merging the data sets. Calculating the data you want. Plotting.

The answer performs some of the calculations in the same lines as plotting, but I think it is easier to understand conceptually to separate it.

The variables you need to calculate are: The yearly log-difference within each municipality, and the average yearly log-difference for each municipality.

Also, you need to sort the data by municipality and date, before calculating, to make sure the yearly differences are between adjacent years.