Project documentation:

Clean and visualize data: Corona 7-day incidence and deaths cases

1. ReadMe

Hello lovely tutors,

If you follow the steps in this ReadMe, nothing will go wrong. But if not, we cannot rule out anything, not even a fourth Covid-19 wave. Speaking of Covid-19, have you ever wondered how the RKI comes up with its beautiful plots and diagrams?

No? Well, then get outta here!

Yes? Great. You have found the perfect Basic Python Final Project to satisfy your curiosity.

1. To start of your exploration, clone this repository on your local computer.
2. Open the editor, so that you see the Gib-Hub project "BasicPython_CoronaPlot" on your local computer.
3. Now, you have to add the RKI data. And because it is too big to store on GitHub, you have to trust me for once and follow this dropbox-link (don't worry, it's safe. Scout's honor!).
4. Download the two csv files (RKI_Corona_Landkreise & RKI_COVID19) and unpack them into a folder with the name "data". Next up, store this folder in the GitHub-project "BasicPython_CoronaPlot" alongside the folders "documentation", "loadAndClean", "main", and "visualization".
5. Now that you are finished with setting up the data, you can open the folder "main". Open "main.py" in your preferred IDE. Before running it, however, make sure that you have installed the following packages in your Python Interpreter: 'pandas', 'matplotlib', 'numpy'.

Have fun exploring the data!

2. Programming Journey

Our journey began when one of us explained that he wanted to perform some (surgical) operations on Corona patient (data). Everyone was quickly on board with this idea, as we had all chosen the course to become better data scientists. The Covid-19 data provided the perfect practice material for juggling csv files on our own; and, moreover, they were interesting to work with. We were particularly keen to compare the different variables, such as age, gender, and state.

After our requirements were approved by our handsome tutors, the first thing we did was to create a GitHub repository, because we had heard that this is exactly the way the bad boys program nowadays. However, we quickly learned that GitHub isn't that cool as everyone was telling us… there was simply far too little space for our beautiful data. So we had to add our data locally on our computer (just like you pretty ones). Good old dropbox helped out here.

After these minor teething problems, we were good to go. However, we quickly found out that not everyone in our team took math classes in school: Instead of calculating the 7-day incidence, we only calculated the case number of the last seven days. You could say that this led to our first major insight, which is how the 7-day incidence is actually calculated. Specifically, we had to sum over all values submitted in one day, then compute the 7-day window and shift right by 1, because the values from one day are calculated by the 7 days before (more details on that can be found in the file 'Calculate.py' in the function 'compute_incidence).

Now, let's be honest: the rest of the data cleaning was a piece of cake due to the solid python training we enjoyed this semester from our outrageously gorgeous tutors. Nevertheless, the next problem was already waiting along the way, disguised as an innocent hitchhiker with his thumbs up. When we stopped due to our good nature and asked for his name, he only replied: "Bar Plots!".

After getting into our car, he slowly thawed out, and started explaining to us long and hard why bar plots were the best kind of plot and why we should definitely use bar plots to plot our data. After all, we had also written into our Final Project Requirements that we were going to use bar plots and stacked bars for our project, which he rubbed under our collective noses several times. Alessandro in particular, however, kept pointing out to the stranger that bar plots with a plethora of data points (as we encountered with the RKI data), combined with a plethora of categories (as we had promised our charming tutors in the requirements) were simply a bad idea.

After hours of debating, we finally had enough—by this point the stranger was just lethargically muttering "bar plots, you need bar plots" into his unkempt beard. We stopped the car again and wished the stranger a good onward journey. When we were finally among ourselves again, we were sure: We will use line plots for the

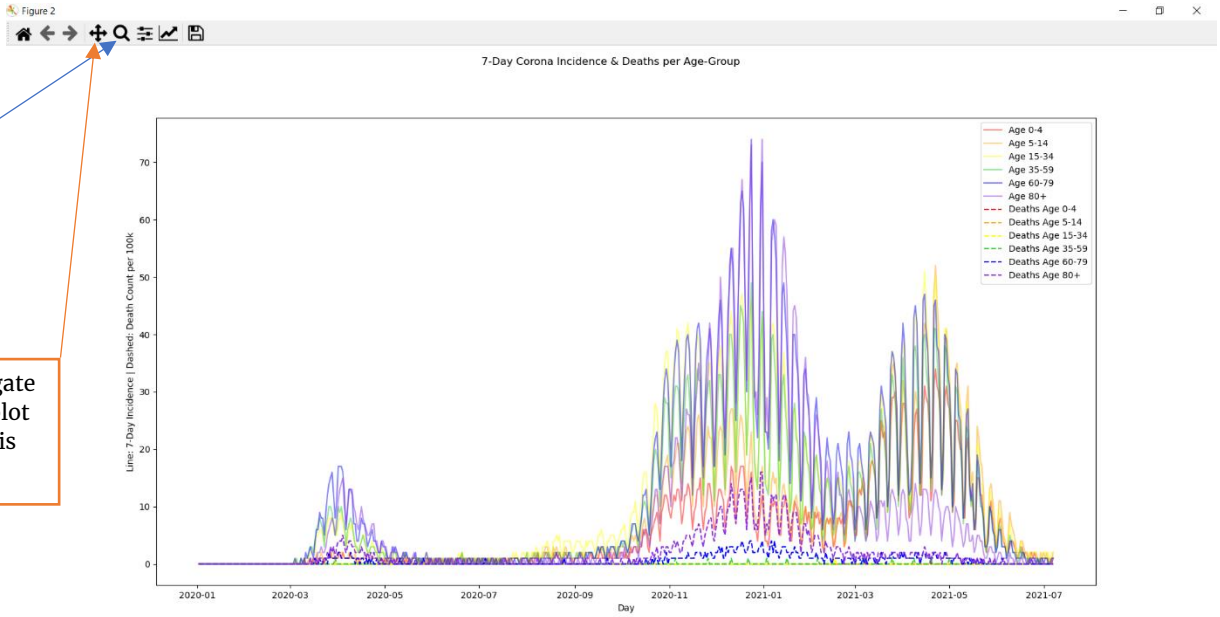
visualization of the Corona data. Too many variables and too much data left us no other choice—despite our full-bodied promises in the project requirements. This, in turn, turned out to be an insanely good decision in retrospect, especially for the category-intensive plots like 'incidence per age group' and 'incidence per state'. Since we had to include the number of deaths in addition to the incidence, at a certain point we could no longer think of bar plots. Additionally, line plots allowed us to divide the data into line and dashed plots (see results below).

A final nuisance became another topic, which actually should have already found its conclusion in elementary school: Upper and lower case. During the final error-testing, we were repeatedly annoyed by capitalizations of folder and file names, since the Python scripts sometimes didn't care about capitalization, but sometimes they did. Following naming conventions, we agreed to write folder names in lowercase and file names in uppercase. Except that GitHub absolutely objected to changing a damn folder name in the online repository. Let's face it Mr. GitHub: You're not as cool as everyone says you are.

In conclusion, we can say that we enjoyed the project very much and worked wonderfully together as a team. This beautiful self-efficacy experience of collaboration makes us look forward to further programming projects. And since this was the first "big" software project we participated in, even if we already had theoretical knowledge, this was a good practical exercise for planning and implementing further software projects. In terms of content, we were particularly able to improve our skills with the libraries 'pandas' and 'matplotlib'.

3. Results

In our exploration of the Covid-19 data, we plotted four different figures: 'Total incidence', 'Incidence per age group', 'Incidence per sex', and 'Incidence per state'. In this results section, let us take a look at 'Incidence per age group', as it is probably the most interesting one in its progression.



In this figure, we can see the 7-day incidence as a line plot, and the death count as a dashed plot. The y-axis shows the respective cases per 100k, and on the x-axis we see the individual days as a time interval from January '20 until July '21. The age groups are depicted using different color schemes.

As in each of the graphs, we first see the three Covid-19 waves, with the second and third clearly exceeding the first in magnitude. Of particular interest in this plot is the different incidence of age groups in each wave. Whereas in the first two waves the age group '80+' was particularly affected, this is extremely flattened in the third wave.

Exactly the opposite is true for the age groups '0-4' and '5-14'. In the first wave this age group is hardly of importance; in the second wave we see a significant rise among affected children; and in the third wave these two age groups are even more affected than the age group '80+'.

Nevertheless, the death figures show that the '80+' age group were the most affected in each wave, despite much lower incidence values in the third wave. The age groups '0-4' and '5-14' have such low death rates that they are not even visible in this plot. Thus, with the help of this plot, we can draw several conclusions:

1. The elderly ('80+') are most affected by Covid-19 and have the highest mortality rate.
2. The elderly ('80+') had the lowest incidence rates in the third wave, which is most likely explained by the vaccination that this age group had already received in the first three months of 2021.
3. Despite high incidence rates of children in the third wave, mortality rates have not increased significantly in this age group ('0-4' & '5-14').