Diary – Week 1: Topic: Project overview, and data processing (acquisition, cleaning, transformation, annotation, etc.).

I focused on this first week on what topic I want to visualize, from where do I get the data to visualize and try to make some tiny steps with the data.

The topic of my project is going to be about the Covid-19 situation. This is not shocking because of it the most pressing topic at the moment but it also delivers a lot of data to visualize.
Although, this might also be a little problem.
Many institutions worldwide provide data about the current virus situation. But when you look at the provided data they differ quite a lot, when you compare them with each other. Also, many of them are quite outdated. The most trustworthy data sources get probably provided by the WHO and John Hopkins University.
Here, for example, an outdated REST API from a chinese university: https://lab.isaaclin.cn/nCoV/en. It delivers data about corona via HTTP requests but the last update was done in February. Nonetheless, the provided data by the Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE) seems to be more than sufficient. Simply because they have a clear schedule when they are gonna update their data and it seems more accurate than other data sources. Although, I think it does not make much of a difference in this case.
Take a brief look at it here: https://github.com/CSSEGISandData/COVID-19.

The data on the page gets updated once a day at 23:59 (UTC). All data is stored in several CSV files, which is convenient to work with. A huge disadvantage is when the source gets only updated once a day at midnight, my graphs will always be one day behind. The data is stored on github, which makes it simple to acquire it (git clone/pull), even with various frameworks.
Besides that, I run into a blunder but more about this in the next paragraphs.

I have conducted a few tests with the latest file from "data/csse_covid_19_data/csse_covid_19_daily_reports/". The file contains a lot of useful information such as province, region, last update, confirmed, deaths, recovered, latitude and longitude. Even though I don't know yet what in detail I want to visualize but that information gives me many possibilities. For example, latitude and longitude give me the ability to display the data on geographical maps.
For testing, I developed a python script to preprocess the data, which basically looks like this:

```
df = df.read_csv(file)
df = df.drop(['FIPS', 'Admin2', 'Province_State', 'Last_Update', 'Combined_Key', 'Lat', 'Long_'], axis=1)
df = df.aggregate({'Confirmed': ['sum'], 'Deaths': ['sum'], 'Recovered': ['sum'], 'Active': ['sum']})
df = df.reset_index()
df = df.drop(['index'], axis=1)
print(df)
```

Goal was to simply sum up all confirmed, deaths, recovered and active cases worldwide. The outcome looks like this:

```
   Confirmed  Deaths  Recovered  Active
0     720117   33925     149082  395606
```

You can quickly tell that the numbers do not add up. The number of confirmed cases should be equal to deaths, recovered and active cases but it is not.

A short look at the file and the first row reveals quickly why.

Province_State	Country_Region	Confirmed	Deaths	Recovered	Active
Abbeville	South Carolina	3	0	0	0

The active column is wrong. 0 deaths and 0 recovered cases would lead to 3 active cases. This blunder happens throughout many rows.
The Hopkins University provides an online dashboard as well and there the calculation is correct.
As you can see here:
https://www.arcgis.com/apps/opsdashboard/index.html#/bda7594740fd40299423467b48e9ecf6

I assume that not every country delivers active numbers but some do and the university just calculates the correct value for their dashboard by themselves. So I will do the same and simply subtract the values of deaths and recovered cases from confirmed cases to get the correct value for active cases.

```
df = df.read_csv(file)
df = df.drop(['FIPS', 'Admin2', 'Province_State', 'Last_Update', 'Combined_Key', 'Lat', 'Long_', 'Active'], axis=1)
df = df.aggregate({'Confirmed': ['sum'], 'Deaths': ['sum'], 'Recovered': ['sum']})
df = df.reset_index()
df = df.drop(['index'], axis=1)
df['Active'] = df['Confirmed'] - (df['Deaths'] + df['Recovered'])
print(df)
```

The outcome now looks like this:

```
   Confirmed  Deaths  Recovered  Active
0     720117   33925     149082  537110
```

When we now add up the values of the latter three columns, the outcome is the value of the confirmed cases, which is correct.
This blunder also appears in other files in the same folder, but not in other folders.

After discovering and fixing this blunder, I decided to move on and think about how to visualize this kind of data. I have a bit of experience with JS and according to the requirements the visualization has to be online and interactive, so JS seems to be a valid choice.

To visualize my data (see picture below) I set up a web application with React and ‘React Vis’. It is still just a test run and I am not 100% sure yet what I am gonna visualize in detail but ‘React Vis’ allows me to not just visualize data but also to make it interactive.

Overall, the dataset seems to be sufficient to build interactive graphs. Depending on the graphs and how up-to-date I want to keep them, some preprocessing with python and processing of the data with React will be needed. I developed also other graphs already but more about this next week.

COVID-19 Statistics

Home | Country | Countries

Below a bar chart can be observed, which represents the newest covid-19 numbers of all countries combined.