Coronavirus Project

18/01/2021 to 25/01/2021

–

Group B

María Cagigas, Juan Guerrero y Kapil Dadlani

Data Science

The Bridge

General Vision

Group project based on worldwide Coronavirus data. We have explored and analysed coronavirus data from five countries, previously designated: France, Spain, India, Peru, United States.

Goals

The main goal is to use all the knowledge we have acquired to be able to get, clean, analyse and visualize data (EDA) to be able to draw our conclusions. Another goal is to be able to create an API so another group can access specific information related to our designated countries.

Specifications

In order to achieve the goals and make the delivery the most, all the requirements needed are specified:

Software

Visual Studio Code

Power Point

Adobe Acrobat Reader DC

Hardware

Processor = Minimum i5

RAM memory = 8GB onwards is recommended

Requirements

Python (Pandas, Numpy, Matplotlib, Seaborn libraries)

Fonts Power Point: Calibri (Cuerpo), Segoe UI Semilight, Segoe UI Black

Internet connection to import updated data from https://ourworldindata.org/coronavirus-source-data

Steps

I. Research the context

We researched the alarm states or state of emergency of all our given countries (France, Spain, India, Peru, United States). Some of them have had more than one alarm states, others have only had one. Some of the alarm states of the countries are still active and others have ended. To have control of all this in order to change the dates in our project, we had to check few websites with this information. We wanted to know, among other things, if the alarm states had an impact in the data of the coronavirus in each country.

II. Get Data

We got data from https://ourworldindata.org/coronavirus-source-data and also...

Spain: https://www.lamoncloa.gob.es/covid-19/Paginas/estado-de-alarma.aspx

France: https://www.aa.com.tr/en/europe/france-extends-covid-19-state-of-emergency-to-feb-16/2032593

India: https://www.brookings.edu/blog/future-development/2020/07/02/how-well-is-india-responding-to-covid-19/

https://www.indiatvnews.com/news/india/lockdown-unlock-in-india-covid19-pandemic-guidelines-restrictions-2020-coronavirus-lockdown-series-674925

Peru : https://www.gob.pe/8784-coronavirus-preguntas-y-respuestas-sobre-el-estado-de-emergencia

United States:
https://en.wikipedia.org/wiki/List_of_national_emergencies_in_the_United_States

III. Data Wrangling

The data was imported directly as a .csv file which we transformed to a pandas dataframe using pandas library:

```
data = pd.read_csv("https://covid.ourworldindata.org/data/owid-covid-data.csv")
```

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...
0	AFG	Asia	Afghanistan	2020-02-24	1.0	1.0	NaN	NaN	NaN	NaN	...
1	AFG	Asia	Afghanistan	2020-02-25	1.0	0.0	NaN	NaN	NaN	NaN	...
2	AFG	Asia	Afghanistan	2020-02-26	1.0	0.0	NaN	NaN	NaN	NaN	...
3	AFG	Asia	Afghanistan	2020-02-27	1.0	0.0	NaN	NaN	NaN	NaN	...
4	AFG	Asia	Afghanistan	2020-02-28	1.0	0.0	NaN	NaN	NaN	NaN	...
...	...	...	...	...	...	...	...	...	...	...	...
61850	ZWE	Africa	Zimbabwe	2021-01-19	28675.0	783.0	776.571	825.0	52.0	39.143	...
61851	ZWE	Africa	Zimbabwe	2021-01-20	29408.0	733.0	736.000	879.0	54.0	41.429	...
61852	ZWE	Africa	Zimbabwe	2021-01-21	30047.0	639.0	668.429	917.0	38.0	40.143	...
61853	ZWE	Africa	Zimbabwe	2021-01-22	30523.0	476.0	630.571	962.0	45.0	42.286	...
61854	ZWE	Africa	Zimbabwe	2021-01-23	31007.0	484.0	589.429	974.0	12.0	41.571	...

61855 rows × 55 columns

IV. Data Mining / Clean Data

We cleaned the dataframe mentioned in the data wrangling step. We filtered our dataframe with the 5 countries we were designated: France, Spain, India, Peru, United States. With these countries selected, we had 1776 rows and 55 columns. We then had to choose which columns we wanted to work with. So, the criteria we used to choose these columns was the combination of the amount of missing values they had (we chose the ones with the least missing values) and the amount of unique values in the columns. If there were only 5 values (one for each country), we didn´t add them to our final dataframe (except for “life expectancy” which was the only one we did add with only 5 values because we had to answer a question regarding this column).

These are the initial columns we had once we imported the data. This list represents the columns and the percentage rate of missing values each column has. Those which are highlighted are the ones we finally chose for our analysis:

iso_code	0.000000
hospital_beds_per_thousand	0.000000
female_smokers	0.000000
diabetes_prevalence	0.000000
cardiovasc_death_rate	0.000000
gdp_per_capita	0.000000
aged_70_older	0.000000
aged_65_older	0.000000
median_age	0.000000
population_density	0.000000
population	0.000000
life_expectancy	0.000000
human_development_index	0.000000
total_cases_per_million	0.000000
continent	0.000000
location	0.000000
date	0.000000
total_cases	0.000000
new_cases	0.056306
new_cases_per_million	0.056306
new_deaths_smoothed	1.463964
new_cases_smoothed	1.463964
new_cases_smoothed_per_million	1.463964
new_deaths_smoothed_per_million	1.463964
stringency_index	2.646396
total_deaths	8.220721
new_deaths	8.220721
new_deaths_per_million	8.220721
total_deaths_per_million	8.220721
reproduction_rate	13.682432
tests_units	17.004505
male_smokers	18.243243
new_tests_smoothed	18.918919
new_tests_smoothed_per_thousand	18.918919
extreme_poverty	20.608108
new_tests	33.727477
new_tests_per_thousand	33.727477
positive_rate	34.065315
tests_per_case	34.065315
total_tests	45.213964
total_tests_per_thousand	45.213964
hosp_patients_per_million	58.277027
hosp_patients	58.277027
icu_patients_per_million	58.840090
icu_patients	58.840090
handwashing_facilities	79.729730
weekly_hosp_admissions	92.004505
weekly_icu_admissions_per_million	92.004505
weekly_icu_admissions	92.004505
weekly_hosp_admissions_per_million	92.004505
new_vaccinations_smoothed	95.551802
new_vaccinations_smoothed_per_million	95.551802
total_vaccinations_per_hundred	96.340090

```
total_vaccinations                    96.340090
new_vaccinations                      97.184685
```

We then changed date column data to datatime.

Final dataframe:

```
data_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1776 entries, 24935 to 52171
Data columns (total 15 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   iso_code                         1776 non-null   object
 1   continent                        1776 non-null   object
 2   location                         1776 non-null   object
 3   date                             1776 non-null   datetime64[ns]
 4   total_cases                      1776 non-null   float64
 5   total_cases_per_million          1776 non-null   float64
 6   new_cases_smoothed               1750 non-null   float64
 7   new_cases_smoothed_per_million   1750 non-null   float64
 8   new_deaths_smoothed              1750 non-null   float64
 9   new_deaths_smoothed_per_million  1750 non-null   float64
 10  total_deaths                     1630 non-null   float64
 11  total_deaths_per_million         1630 non-null   float64
 12  life_expectancy                  1776 non-null   float64
 13  stringency_index                 1729 non-null   float64
 14  new_deaths                       1630 non-null   float64
dtypes: datetime64[ns](1), float64(11), object(3)
memory usage: 222.0+ KB
```

All functions to clean the data were kept in the mining_data_tb.py file and folders_tb.py.

V. Creating an API

We created an API to pass information (new_deaths) to another group. To do so, we used a web Framework: FLASK. We defined 3 functions with its decorators. One of them was for the main site, the other one was made to introduce the group id and it returned a token that you had to use in the next URL to finally get the json. To access our API, you had to be in our same WIFI connection (same router) and had to write our private ip, followed by ":" and the port we had chosen. After that, you had to access the group_id page with the B88 password. That would give you the token needed to access the data. Bellow, you can see the 2 URL´s you had to put in the navigator to access our API:

http://"private ip":6060/group_id?password=B88

http://"private ip":6060/token_id?password=B227766764

All this was coded in the server.py file in the "APIS" folder.

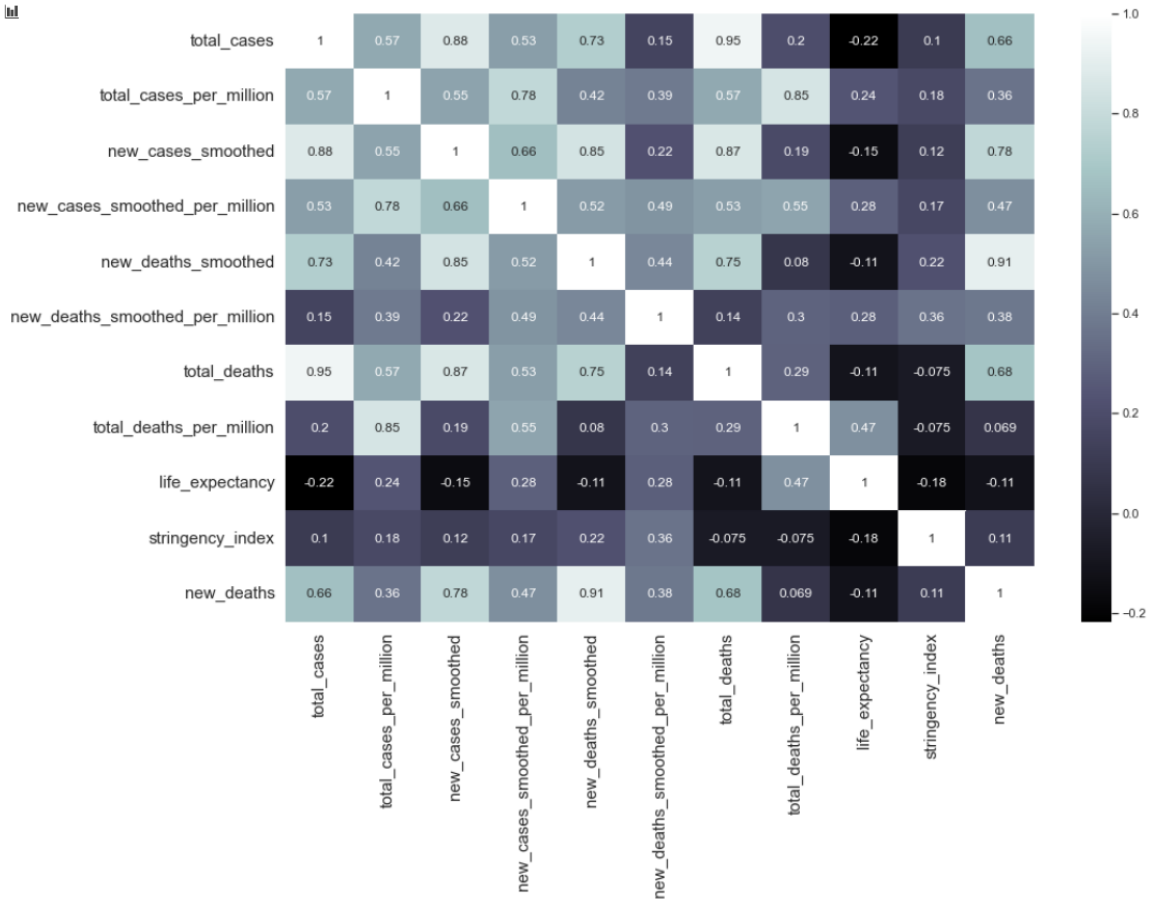
VI. Visualization

Libraries used: Matplotlib and Seaborn were the ones chosen. Firstly, we showed different tendencies for each column. In these tendencies we included the five countries we were designated with. We also included the states of alarm or state of emergency which all these countries put in response to the covid-19 pandemic crisis.

- Trends plotted (five countries in the same plot):
    - Total cases
    - Total cases per million
    - Life Expectancy
    - New cases smoothed
    - New cases smoothed by million
    - Stringency index
    - New deaths
    - New deaths smoothed per million
    - Total deaths
    - Total deaths per million

We also plotted the alarm state vertically for each trend and for each country.

VII. Correlation Matrix



Above, we can see the correlation matrix of the Coronavirus dataset.

As shown, the columns with the highest correlation between them are:

- Total deaths and Total Cases

Th columns with the least correlation between them are:

- Total Cases and Life Expectancy

VIII. Others

Answering the questions of option C:

a. What position do your countries occupy in comparison to the number of total infected, total deaths and life expectancy

- *Total infected*: the graphs plotted show the evolution of the percentage of total registered cases for each of the considered countries along time. In the present time the United States represents the 12.67% of total cases, India the 5.38%, France the 1.57%, Spain the 1.26% and Peru the 0.55%

- *Total deaths*: similarly as for the total cases, we plotted the evolution of the percentage of total deaths for each of our countries. As for today: United States represents the 9.84% of total deaths, India the 3.6%, France the 1.72%, Spain the 1.30% and Peru the 0.93%

- *Life expectancy*: in the global list for the 192 countries considered in this dataset, our countries were ranked in the following positions from highest life expectancy to lowest: Spain 8º position, France 15º, United States 42º, Peru 65º, India 136º.

b. What can you conclude about your data study?

There are many conclusions that can be extracted from this EDA, but we want to focus on the main three that caught our attention:

- *Strong restrictions (stringency index) had a positive impact in France and Spain since there were fewer infections and deaths.* In the case of France and Spain, we can see how when the new deaths and new cases curves rise steeply, restrictive measures are taken and there is a sudden drop of these two parameters after some time. Also, when these measures are relaxed, these curves begin to rise again. This seems to show a correlation between these parameters and the stringency index, suggesting these measures work in containing the virus.

- *Restrictions in Peru, India and United States had no impact in decreasing infections and deaths.* Contrary to what we've seen in the cases of France and Spain, the same does not work for India, Peru and the US. It is true that for this last country these measures had an impact in the decrease of the total deaths registered, but whilst they seem to be kept quite constant through all this time, the new cases and deaths seem to fluctuate regardless of it. For India and Peru measures were




implemented very early in their expansion of the virus and have kept relatively constant through time. Regardless of this, these curves have peaked and fallen. Given these two views we can't conclude that the more restrictions, the fewer infections and deaths: the results are divergent depending on the countries considered.

- *The death ratio (new cases vs deaths) was higher in the first wave.* For France, Spain and the US there seems to be three waves where new cases and deaths have risen steeply. It is worth noting that for the first one the new cases were much lower than for the other two, while the deaths were much higher. This could be for a number of reasons which suggest that the mortality of the virus was higher in the beginning or that there were much less cases registered tan there actually were.

c. Are the outliers or some rare data?

- Outliers:
    - **France**
      New cases smoothed → We can see negative values in April 2020.
      New deaths → We can see negative values in May, July, September, October and November.
    - **Spain**
      New deaths smoothed → We can see negative results in the end of May beginning of June.
      Total deaths → There is a rare value in this plot. We can see this in the fall in total cases at the end of May. This is wrong data as the total cases can never be less.
      New deaths → We can see a rare data (less than 0 value) at the end of May
    - **India**
      New deaths → Outlier in mid-June 2020
    - **Peru**
      New deaths → Outliers in July and August