## 

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

## 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.

## 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/>

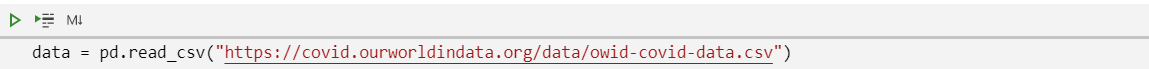
<https://www.indiatvnews.com/news/india/lockdown-unlock-in-india-covid19-pandemic-guidelines-restrictions-2020-coronavirus-lockdown-series-674925>

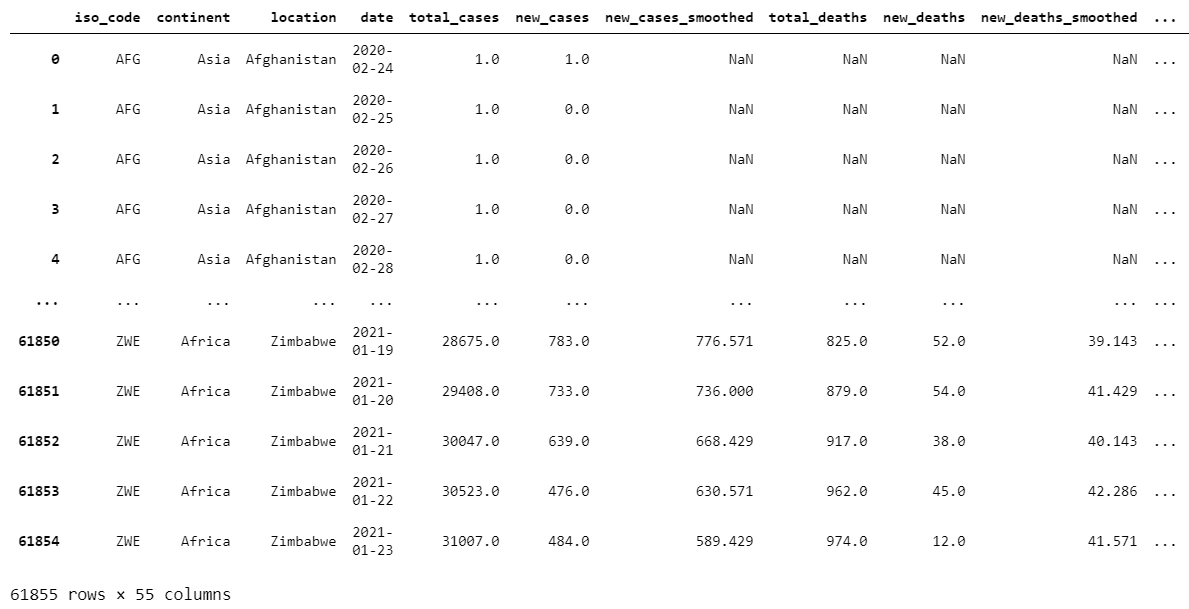
Peru : <https://www.gob.pe/8784-coronavirus-preguntas-y-respuestas-sobre-el-estado-de-emergencia>

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

## Data Wrangling

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





## 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

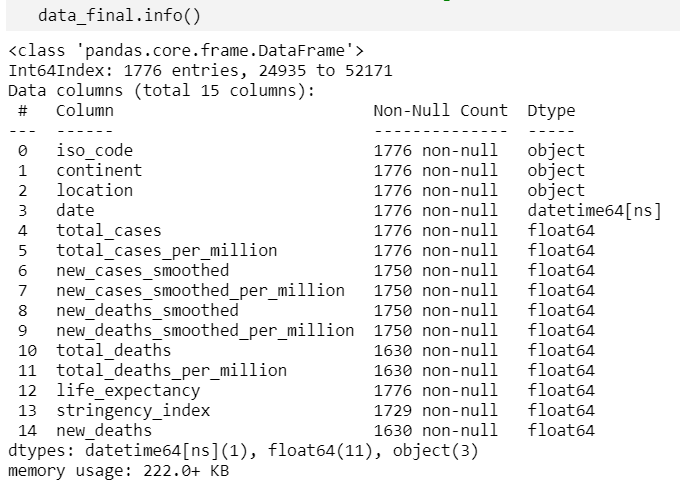
total\_vaccinations\_per\_hundred 96.340090

total\_vaccinations 96.340090

new\_vaccinations 97.184685

We then changed date column data to datatime.

Final dataframe:



All functions to clean the data were kept in the mining\_data\_tb.py file and folders\_tb.py.

## 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.

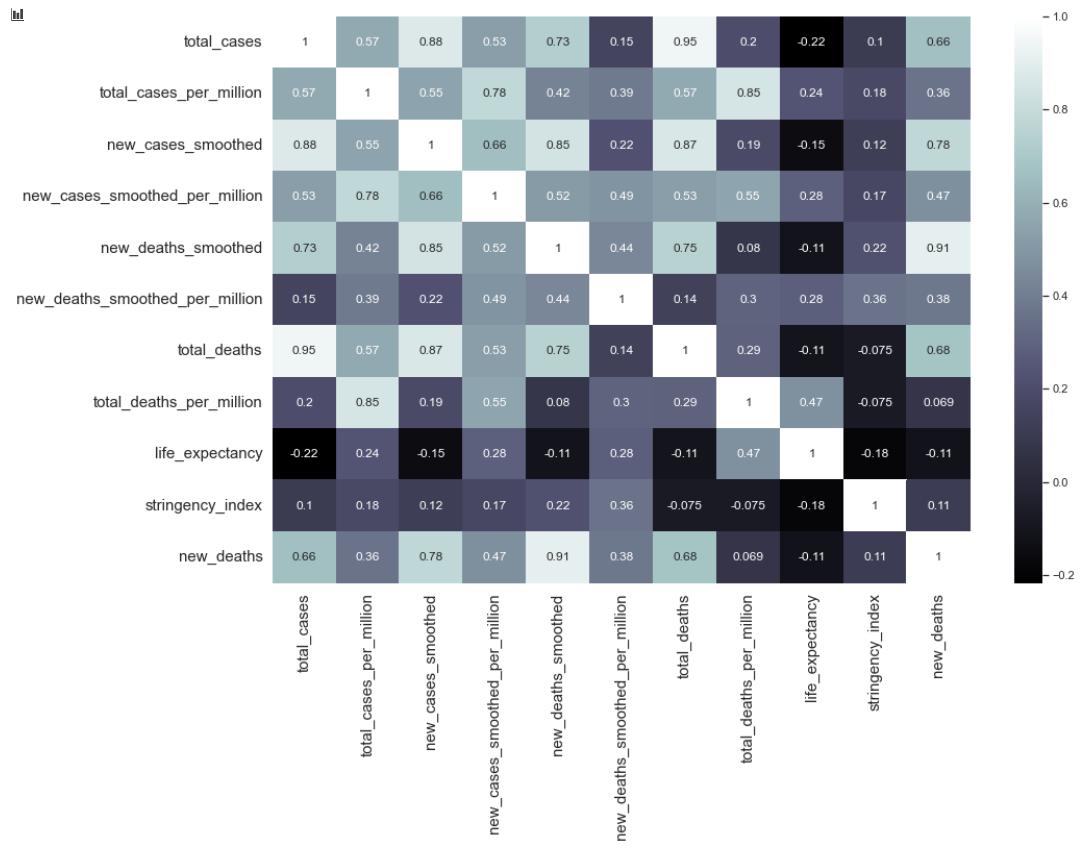
## Visualization

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.

## 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

## Others

## Answering the questions of option C:

1. **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.

1. **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.

1. **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