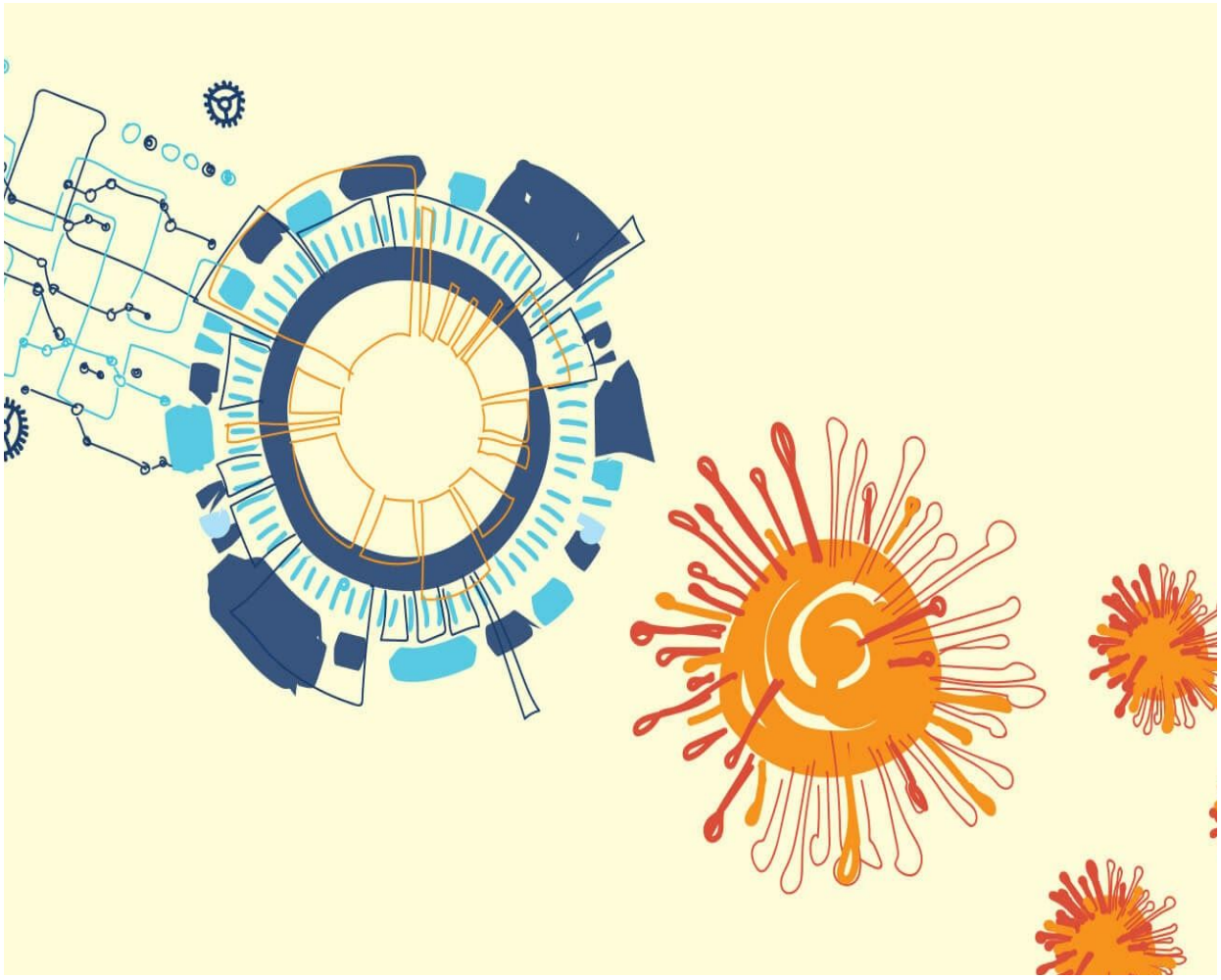

Data Science and Covid. Group Delivery.

23/07/2020 to 26/08/2020

— Group **B**

Jose Mª González, Cristina Segura, Javier A. Yestera

Data Science

The Bridge

General Vision

As part of our bootcamp training, we have done some research about the actual pandemic situation happening all around the planet. Countries were previously selected and assigned to groups, so we have focused our scope in India, Peru, France, Spain and USA

Goals

We aim to reach option A. Once finished all C and B requirements, we have also saved a collection of plots in different folders for each country. We have also used distributed modules for each functionality, analyzed and explained the consequences of alarm state measures taken in each country and plotted the progression measured in 10 days lapses.

Specifications

Software

Python 3 (Including Numpy, Pandas, Matplotlib libraries), Visual Studio Code, Adobe Acrobat or any .pdf Reader, Google Chrome or any suitable internet browser. Works in Windows, Mac and Linux Os

Hardware

AMD Ryzen 3 with 8Gb of Ram memory MacBook Pro i5 processor and 8Gb Ram memory.

Requirements

We need an internet connection in order to update our dataset from https://ourworldindata.org where data is free and we won´t need to register.

Steps

- ### Research the context

  We had to research the news and documents to find dates when every country declared the emergency state. Some countries also finished this state and others are still under quarantine measures, so we have been following the track in order to detect changes on this subject.

- ### Get Data / Data Mining / Clean Data

  We get data from the url https://ourworldindata.org/coronavirus-source-data . Emergency state dates are taken from Spanish government (Spain), Wikipedia (India), Peruvian embassy in Spain and the USA embassy in Peru (Peru), CNN news (USA). This research is located in our documentation folder.

  We have tried to automatize all steps using functions distributed in modules.

  open_csv(url) function to read and convert into dataframe the dataset contained in the given url. It also changes date data to date type and returns the Dataframe. Located in **folders_tb**.

  seleccion_paises(dataframe.paises) function from **mining_data_tb** module to select our assigned countries from the complete dataframe, extracting the desired values with a loc method. Args and kwargs are the dataframe and a list of selected countries.

  seleccion_columnas(dataframe, col1, col2) function in **mining_data_tb** module used to select the columns to analyze giving the columns position.

  borrar_previo(dataframe) is used to search the first detected case and delete previous values. Then, using pivot table it returns the dataframe sorted by date, starting when first infection was detected and ending with the last.

 mining_data_tb module also **has cero_nan(dataframe)** function to change Nan values for zeros by applying the fillna(0) method to the complete dataframe.

 We have detected several **outliers** in all countries data but we have kept them because countries are re defining the data collection. We have also plotted data in a month frequency in order to smooth these outliers,

Data Wrangling / Visualization and Backup

We have automatized data wrangling and visualization using functions that parse every dataframe column to analyze, selecting a country or a group of countries. Several plots are also saved in independent folders named as countries.

All these functions are located in **visualization_tb** module.

 paises_juntos() function compares the evolution of several selected countries in a timeline, creating a pivot table for each column, and then plotting, printing and saving a copy of plots in "../resources/plots/paises_juntos". It also updates the previous image if exists. Parameters are the Dataframe and a dictionary of countries.

 paises_juntos_mes() function is a variation of paises_juntos with a month period between samples. Saved plots are named as monthly.

 emergencias() from **visualization_tb** plots different trends for each column of selected country. It also plots with vertical lines the start end end (if declared) of the alarm state. Function parses a list made with selected columns. Other parameters are the known dates of emergency activation and deactivation. Plots include grouped countries bars plot, and individual countries plots. All files are saved and overwrite older versions in each country´s folder inside "..resources/plots"

 emergencias_mes() function is a variation of emergencias with a month period between samples. Saved plots are named as monthly

 total_paises() from **visualization_tb** function compares the ranking of countries by number of infected and number of death. Both absolute and by million of inhabitants. Parameters are Dataframe, selected columns, and the countries dictionary (containing names and colours). Plots are printed and then saved as *"barras_columname"* in "../resources/plots/paises_juntos"

 total_paises_orden() function returns the same as total_paises in reversed order so position of the countries is clearer. Plots are printed and then saved as *"barras_orden_columname"* in "../resources/plots/paises_juntos" .