Data Engineering Report - 16164571

For this project, I used the John Hopkins Covid 19 dataset which is available on GitHub at https://github.com/CSSEGISandData/COVID-19. This is a data set that starts at the beginning of the Corona Virus pandemic and is updated daily with statistics for all the countries around the world.

1. Data Exploration

Data Exploring involves the visual exploration of a data set to gain a better understanding for the data set and its characteristics within.

I focused on the general worldwide dataset for this project which gave CSV files for each daily worldwide report. In each CSV file, there was a lot of data at an initial glance. I immediately noticed data that would be useful for my graph implementations and I also noticed data that would not provide much value for my project.

I began by importing all the CSV files into a dictionary and I labeled them with the corresponding date to which they provided daily statistics for. I began to explore the data visually in my program and do this by printing out each Dataframe. Each data frame had around 4000 rows of data with 14 Columns providing a variety of stats. I established I would like to focus on Countries' cases over time, death rate, active cases, and also the cases confirmed on a given day. However, the dataset only provided the total number of confirmed cases each day but I calculated this number which will be talked about more in the data preprocessing section. Overall after exploring my data that there was a lot of data in the set and it was important for me to pick out the most important pieces for my implementation which I undertook in the pre-processing section.

Before I pre-processed my data I decided what data each graph needed.

Graph 1 - Examining the effect of a lockdown period on confirmed daily cases:

Country, Today's Cases, 5 Day Average (Based on today's cases) and Date.

Graph 2 - Comparing 5 European countries confirmed cases over the pandemic period:

Country, Confirmed Cases, Date.

Graph 3 - Death Cases and Spike detection in several European countries:

Country, Deaths, Date

Graph 4 - All European countries active cases yesterday:

Country, Active Cases, Date

2. Pre Processing

Data processing involves cleaning and normalising our data and the product of this processing is the final data set.

From this, I concluded the data frame columns I would need are Country_Region, Deaths, Active, Confirmed, Today's Cases, 5 Day Average, and Date. I create a new data frame from the original date frame of each date which just contains these required columns.

Three of these data columns were not provided for me which are Today's Cases, 5 Day Average, and Date. To calculate today's cases, I started from the start of the data record date and got the current day and previous days' confirmed cases. From here I subtracted them from each other and added this new figure into the new Today's Column of the current day's data frame. This gave us an accurate rise in cases each day.

To add the date I extracted the date from the original CSV file and added it into its own individual column for the corresponding data frame. I had to change the format of this date from m-d-Y to Y-m-d in order for it to work with my visualization tool, Bokeh.

Lastly to add the moving average I used a built-in data frame method .rolling() to calculate the rolling 5 day average of today's cases column.

```
#Adds a new column containing the 5 day rolling average of the daily cases
irelandCases['MA'] = irelandCases.iloc[:,4].rolling(window=5).mean()
```

Graph 1 required a specific date range so I created a method chooseDateRange(start,end) which returned a dictionary of the data frames between the two user-specified dates. As the graph needed just one country I also have a method that returns the data for a specified country.

```
selectedDates = chooseDateRange(start_date,end_date)
irelandCases = filterLocation('Ireland',selectedDates)
```

For graph 2 I used a similar method to filterLocation called getAllDates(location). This got all the data frames for the dates of a given country and preprocessed some of the necessary columns such as Date.

I used the same method to get all the dates for each country's data frame in graph 3. I used some of the already obtained data frames from graph 2 and combined them with newly processed countries to get the deaths for these countries.

Not much pre-processing had to be done for graph 4, I just calculated the date of yesterday, got all countries from the main imported dictionary where the countries name is in a list which contains European countries then plotted this data on my bar chart.

3. Standardization

Standardization involves the bringing of data into a common format and value range.

For each data frame, I convert each date format to a compatible format for bokeh to use. This means all the dates are in the same format and can allow the library to plot the dates on a DateTime axis.

I also converted the Active cases from a double to an integer as I felt there was no need for this figure to be a double.

The final piece of standardization in this project for me was about the graph axis. I ensured each graph axis displayed the values in a user friendly and easy to read and understand format e.g 1,000,000 rather than 1000000