Jamie Neeson

[jamie@allenrec.com](mailto:jamie@allenrec.com)

A picture containing clipart

Description automatically generated

Introduction to data analytics

Project Report

Date submitted: 21.12.2021

**UCDPA Project Report**

Analyzing COVID 19 confirmed cases and deaths

21.12.2021, submitted by Jamie Neeson

**GitHub URL**

<https://github.com/Jamie-Neeson/UCDPA_Jamie_Neeson>

**Abstract**

I have undertaken this project as part of the UCDPA Certification in Introduction to Analytics. As COVID 19 has become part of our everyday lives over the last two years and will most certainly be for the foreseeable future, I would like to take this opportunity to dive into data to get a better overall view of the global confirmed cases and deaths. The object of my project is to analyze COVID 19 confirmed case and deaths by using tools and functions such as Pandas, NumPy, Matplotlib and Plotly.

In this project, I have used a CSV approach for uploading my data for data analysis.

**Dataset**

The dataset I have used has been sourced from Kaggle: <https://www.kaggle.com/antgoldbloom/covid19-data-from-john-hopkins-university>

The dataset shows confirmed cases and deaths on a country level. As per Kaggle, the version of this dataset that I am using is aiming to be easier to analyze. The data is organized by column rather than by row.

This dataset in its raw state is good, however, there are many missing fields, and the layout is not the most effective for data analysis. I will need to clean the data along with manipulating the DataFrame to pivot it from a wide to a long format.

**Implementation Process**

Obtaining data:

The first stage of the data implementation process was to obtain my data. The dataset was chosen from Kaggle, was downloaded as a CSV file.

**Importing packages:**

As I was using Jupyter Notebook as my IDE, I needed to use the !pip install function to bring in Pandas, NumPy, Plotly and Matplotlib. Once this was completed and the requirements were satisfied, I could then import my packages.

**Loading in the CSV:**

For my analysis I created two DataFrames, one for the confirmed cases and one for the confirmed deaths.

**Viewing the data:**

* Once the data was loaded, I checked the dataset for its info, results showed my DataFrames both had 280 rows and 702, with a mixture of floats and int. I noted here that there were no datatypes as datetime so I knew if I wanted to use dates in my analysis I would need to convert ‘Date’ using the to\_datetime function.
* The information available to me was Country/Region, Province/State, an array of dates with a sum in each field along with coordinates for latitude and longitude.
* For my analysis I decided that I would remove the Province/State, the main reason here being this is where the majority of the missing data was, along with the coordinates for latitude and longitude.

**Data Cleaning and Manipulation:**

After spending time to view the data and to decide what data I needed and what format I needed it in to carry out my analysis I decided to:

1. Drop columns that I don't need:

Table

Description automatically generated

1. Group data by Country/Region and resetting the index. As there were multiple entries for the same country, I decided the best way to format the country data would be to group them together using the df.groupby function, aggregate the DataFrame using the .aggregate(np.sum).T function. Once this was completed I reset the DataFrame’s index.

Table

Description automatically generated with medium confidence

1. Melting/Pivoting the data. This was one of the most difficult things I completed during my analysis as it took many tries to get it right, after many broken datasets I got it right. By doing this, I was able to format the whole DataFrame into 3 columns. I did this by using the df.melt function which allows you to change the format where there is one or more columns as an identifier. Here, I used ‘Date’ as the id\_vars.

Graphical user interface, application, Word

Description automatically generated

1. Formatting data. Here I needed to convert ‘Date’ using the to\_datetime function and then I decided to reformat the date layout to be Day, Month and Year compared to the US format. For this I used the dt.strftime function ('%d/%m/%Y'). I found this very beneficial to be able to quickly read and understand the date.

Graphical user interface

Description automatically generated with low confidence

1. Establishing the maximum date entry. I needed to establish when the last data entry was on the DataFrame so I can run code to take into consideration for example the total entries in ‘Confirmed’ to a certain date. Creating max\_date was an easy way to get this done.



1. Creating the total confirmed cases and total confirmed cases, and total of active cases. I created this to get a quick figure for the total confirmed cases and deaths due to COVID 19 globally. This was the last step before moving into visualizing my data.

Graphical user interface, text, application, email

Description automatically generated

**Results – Data Visualization**

For my Data Visualization I have created 5 different visuals using Plotly. I choose to use Plotly over Matplotlib as I found it to be much more sophisticated for my data visualization. It allowed me to create interactive visuals like what you would expect to find on a dashboard on Tableau or Power BI. All in all, I found it as a much better tool, and I have highlighted in Fig, two of the same charts using Matplotlib and Plotly to see the clear different in the visualization output.

**Fig**: the below bar chart shows the top 10 countries with the highest COVID 19 cases. This is an interactive graph (when opened in Jupyter Notebook).

Graphical user interface, Teams

Description automatically generated



**Fig:** the below bar chart was coded using Matplotlib to display the same information as the above visualization that was created using Plotly.

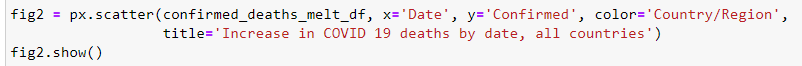
Chart

Description automatically generated



**Fig2**: the below line scatter graph shows the increase in deaths cause by COVID 19 across all countries. This is an interactive graph (when opened in Jupyter Notebook).Chart, line chart, histogram

Description automatically generated



**Fig3**: the below line graph shows the increase in COVID 19 cases within a certain country. For the example below it shows Ireland’s COVID 19 cases from 22.1.2020 to 11.12.2021. I have created the below code to allow the user to simply change the chosen country to display that country’s COVID 19 cases. This is an interactive graph (when opened in Jupyter Notebook).

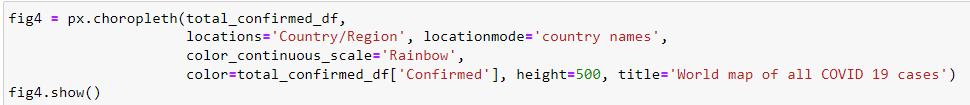
Chart, line chart

Description automatically generated



**Fig4**: the below shows an interactive choropleth map (when opened in Jupyter Notebook) what allows the user to highlight over a certain country and it will display the total COVID 19 cases within that country. The code can be easily changes to show the number of deaths. I opted to create this visualization over a standard heatmap. Map

Description automatically generated



**Fig5**: the below bubble scatter graph shows the total of deaths cause by COVID 19 across all countries. This is an interactive graph (when opened in Jupyter Notebook).

Chart

Description automatically generated

**Data Insights**

**Fig:**

Here we see the top 10 countries with the most COVID 19 cases. Unsurprisingly we see the US leading this table with over 20 million confirmed cases and India are 2nd what just over 10 million confirmed cases. It is surprising to see it that while Italy is within the top 10, ranked 8th, I would have expected to see them higher on the list even that they, apart from China, were on of the worst countries hit by the initial outbreak of COVID 19. In relation to the USA, given how late they were to implement any restrictions to protect their population from this outbreak it is not surprising to see them top the list. What does raise concern is that they have over double the cases compared to India, a much larger country where the living standard would be much lower than the USA with less widely available healthcare.

**Fig2:**

Here we see the graduate increase in the deaths caused by COVID 19. We can pinpoint from this graph the dates where death spikes nearly across all countries: 26.4.2020 (the start of the first wave world outside), 2.11.2020 to 2.9.2021. During this period, many countries eased their restrictions and opened up to international travel. A worrying insight from May 2021 as seen in this graph is that deaths continue to rise despite the rollout of the COVID 19 vaccine.

**Fig3:**

Here we are specifically looking at the increase in COVID 19 cases in Ireland, but the code allows the user to quickly change the desired country. The insights we get from this graph is that we can see a steep rise in COVID 19 cases in the run up to Christmas 2020 with a continual rise month on month until the most recent data entry. From April 2021 to Present there have been nearly 1000 COVID 19 related deaths in Ireland.

**Fig4:**

Here we can see a world map of all of the COVID 19 cases. When interacting with the map, we get insights that we expect such as the most cases in the USA, countries in South America have 5+ million cases and as already discussed, India with over 10 million. What I did not expect to see was such low confirmed COVID 19 cases in South Africa. Countries such as Chad only reporting 2113, Niger 3323 and Samalia 4714. The most unexpected insight from this map is that China has only has 95,963 confirmed cases of COVID 19. Unsurprisingly New Zealand, often hailed for the measures of combating COVID 19, have only reported 2162 confirmed COVID 19 cases.

**Fig:5**

Here we can see the total COVID 19 deaths by country. But now we can view this information in a bubble scatter plot. Key insights here include that the USA have over 350,000 reported COVID 19 deaths. Surprisingly, India who have the 2nd most cases have the 3rd number of total deaths. Brazil, who have nearly 3 million less COVID 19 case than India, have recorded nearly 50,000 more deaths. Mongolia is at the bottom of confirmed COVID 19 deaths with just 1.

**References**

<https://www.kaggle.com/antgoldbloom/covid19-data-from-john-hopkins-university>

<https://plotly.com/python/>

<https://matplotlib.org/>