

COVID DASHBOARD

Using Python - Bokeh

Ajay Dyavathi
Bhavya Nallapaneni
Dinesh Talagadadeevi
Shardul Vedpathak



Introduction

- Bokeh is a Python library for creating interactive visualisations for modern web browsers.
- It helps you build beautiful graphics, ranging from simple plots to complex dashboards with streaming datasets.
- With Bokeh, you can create JavaScript-powered visualisations without writing any JavaScript yourself.
- Bokeh is the art of creating visually stunning and interactive data stories, that captivate and engage the audience with ease.

COVID Dashboard

Objective

- The COVID-19 pandemic has had a significant impact on the world, affecting many aspects of our lives.
- Understanding how the virus is spreading across the globe brings awareness to the general public.
- In this project, we built a dashboard that can clearly visualise the spread of the virus and its impact.
- This dashboard displays the information of Total cases, Total deaths and Total vaccinations across the world.
- It can provide valuable insights into the impact of COVID-19 on a global scale, help identify trends and hotspots, and inform decision-making for public health policies and interventions.

COVID Dashboard

Data

- We obtained up to date information from World Health Organisation (WHO).
- The data is collected in 2 files:
 - 1. CSV file containing the Cases and Deaths information for each country (Cumulative and by-date).
 - 2. CSV file containing the Vaccination information for each country.
- We also needed the geometry information for each country to display the map, for which we have used a global map shape file.

COVID Dashboard

Data

	Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
0	2020-01-03	AF	Afghanistan	EMRO	0	0	0	0
1	2020-01-04	AF	Afghanistan	EMRO	0	0	0	0
2	2020-01-05	AF	Afghanistan	EMRO	0	0	0	0
3	2020-01-06	AF	Afghanistan	EMRO	0	0	0	0
4	2020-01-07	AF	Afghanistan	EMRO	0	0	0	0

Cases and Deaths Info

	Country	Total_Vaccinations_Per100
0	Afghanistan	43.269
1	Albania	106.700
2	Algeria	34.817
3	American Samoa	207.812
4	Andorra	203.100
5	Angola	76.740
6	Anguilla	164.005
7	Antigua and Barbuda	139.399
8	Argentina	242.617
9	Armenia	72.700

Vaccination Info

	Country	geometry
0	Costa Rica	MULTIPOLYGON (((-83.69650 10.93659, -83.68687 ...
1	Nicaragua	MULTIPOLYGON (((-85.70174 11.08088, -85.70242 ...
2	Dominican Republic	MULTIPOLYGON (((-71.75744 19.71011, -71.73827 ...
3	Sint Maarten	POLYGON ((-63.10700 18.06212, -63.08589 18.058...
4	Haiti	MULTIPOLYGON (((-71.75744 19.71011, -71.74861 ...
5	Saint Martin	POLYGON ((-63.01757 18.03339, -63.08589 18.058...
6	El Salvador	MULTIPOLYGON (((-90.09831 13.73140, -90.11431 ...
7	US Naval Base Guantanamo Bay	MULTIPOLYGON (((-75.09501 19.89723, -75.10766 ...
8	Guatemala	POLYGON ((-89.36162 14.41548, -89.36182 14.415...
9	Cuba	MULTIPOLYGON (((-75.09501 19.89723, -75.09495 ...

Shape Info

COVID Dashboard

Easy plotting

```
# Sourcing the merged data into a JSON Data Source object
geosource = GeoJSONDataSource(geojson=df.to_json())
```

```
# Creating a blank figure with given initialisers
plot_cases = figure(title="Total Covid Cases by Country",
                    width=1400,
                    height=650,
                    tools="wheel_zoom, pan, reset",
                    x_axis_type='mercator',
                    y_axis_type='mercator')
```

```
# Turning off grids and axes.
plot_cases.toolbar.logo = None
plot_cases.toolbar_location = "above"
plot_cases.xgrid.grid_line_color = None
plot_cases.ygrid.grid_line_color = None
plot_cases.xaxis.visible = False
plot_cases.yaxis.visible = False
plot_cases.axis.visible = False
```

```
# Using geosource to create patches with
# different color for each country.
patches = plot_cases.patches("xs", "ys",
                             fill_color="Color_cases",
                             line_color="black",
                             line_width=0.5,
                             source=geosource)
```

```
# Creating hover tool that displays info such as cases and deaths
hover_cases = HoverTool(tooltips=[
    ("Country", "@Country"),
    ("Total Cases", "@Total_cases"),
    ("Total Deaths", "@Total_deaths")])
plot_cases.add_tools(hover_cases)
```

```
# Adding the legend to the plot that shows the color scale.
legend_items = [(label, [plot_cases.square([np.nan], [np.nan],
                                           fill_color=color,
                                           line_color="black")]) \
                for label, color in blue_color_palette.items()]
```

```
# Displaying the plot
show(plot_cases)
```


COVID Dashboard

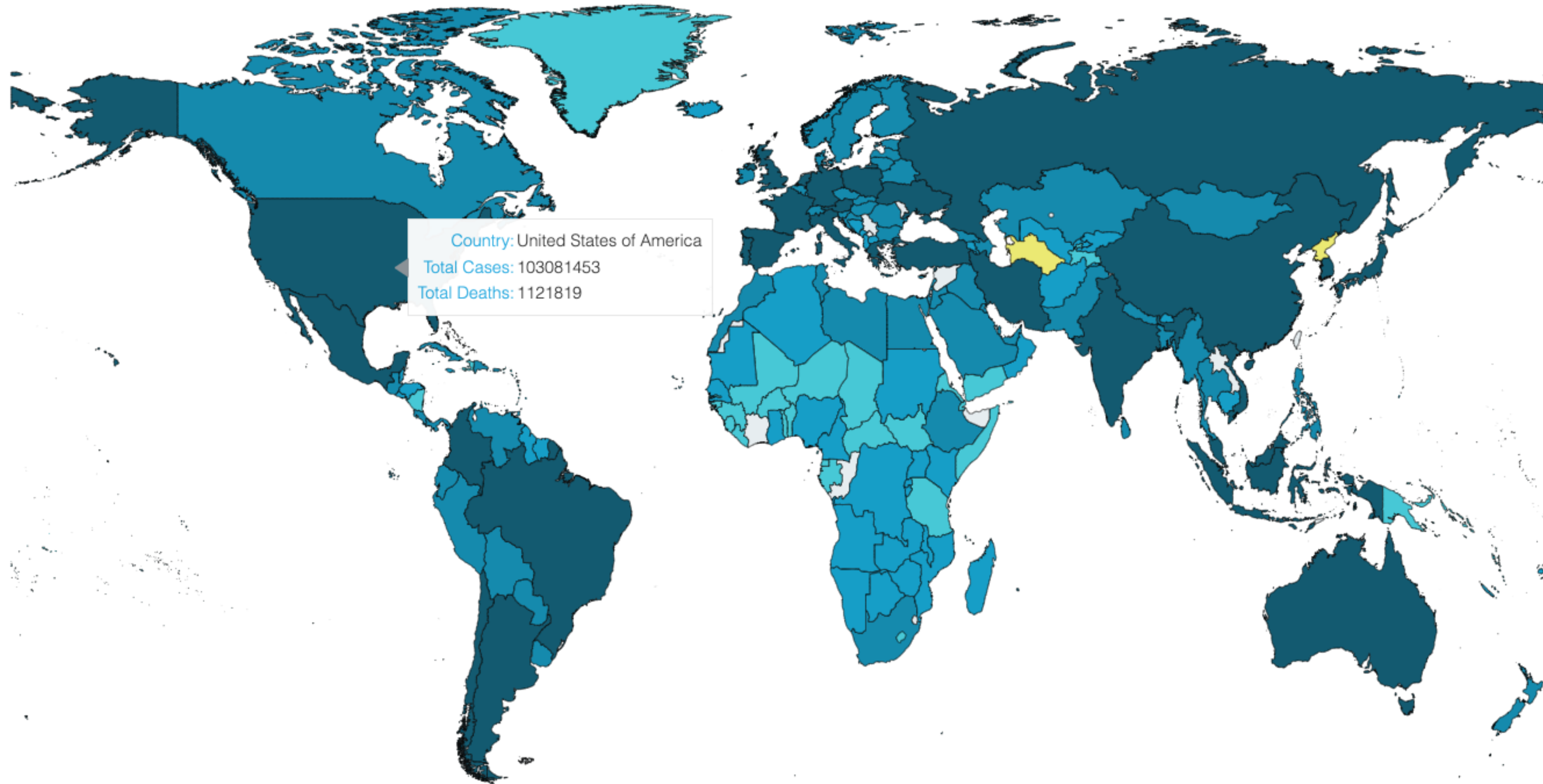
Show Cases

Show Deaths

Show Vaccinations

Total Covid Cases by Country

📍 | 09 | 🔄 | ⌵



COVID Dashboard

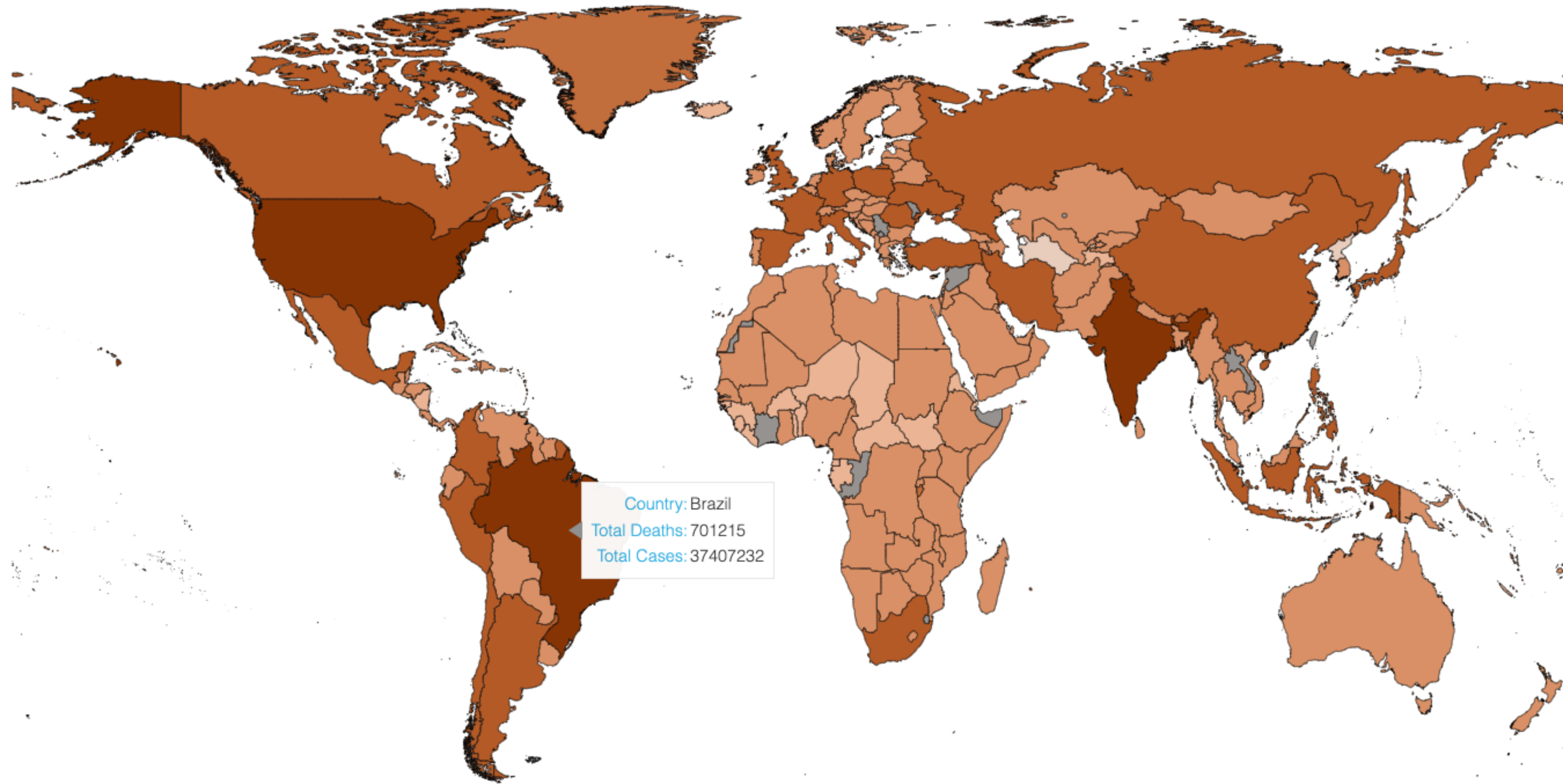
Show Cases

Show Deaths

Show Vaccinations

Total Covid Deaths by Country

📍 | 00 | 🔄 | ⋮



Total Deaths

- 0
- 1 - 50
- 51 - 500
- 501 - 50,000
- 50,001 - 50,0000
- 50,0000+
- Not Applicable

COVID Dashboard

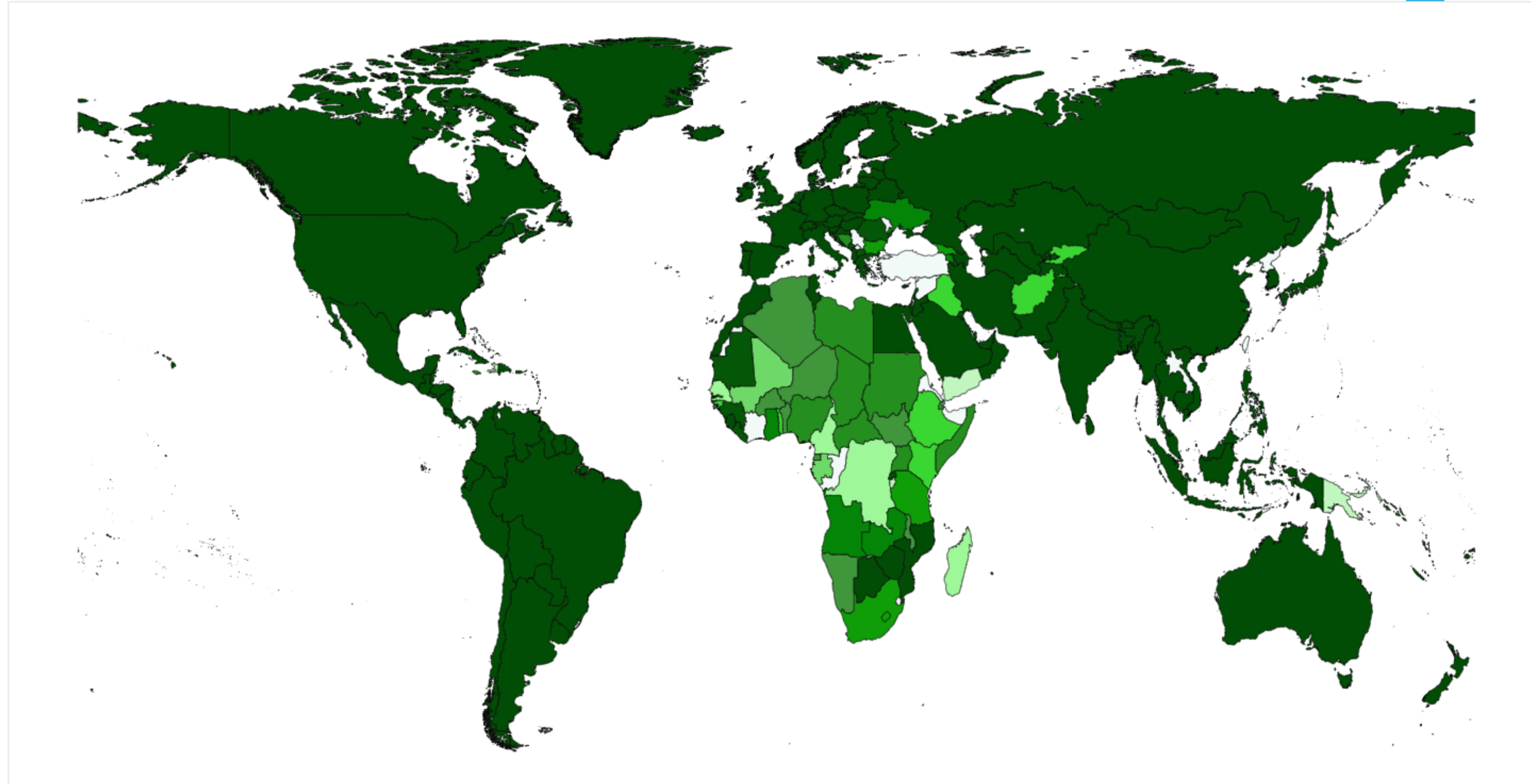
Show Cases

Show Deaths

Show Vaccinations

Total Covid Vaccinations per 100 by Country

⊕ | 09 | ↺ | ⋮



Total Vaccinations per 100

- 0 - 10
- 10 - 20
- 20 - 30
- 30 - 40
- 40 - 50
- 50 - 60
- 60 - 70
- 70 - 80
- 80 - 90
- 90 - 100
- Not Applicable

Bokeh

Advantages

- Very Flexible
- Highly Interactive
- Easy to use
- Open Source
- For the browser, with / without server.
- Able to produce Novel Graphics
- Can handle large and streaming datasets.

COVID Dashboard

Summary

- This dashboard was built using Bokeh python library, which is really to work with.
- As an advantage of Bokeh (with its JS engine), this dashboard automatically gets displayed in a web browser, making it easier to extend this to a website.
- Also, we took our time to pre-process the data carefully to be compatible across multiple files.
- Overall, we feel that we utilised the Bokeh library well enough to present the information in an easy way to understand.

COVID Dashboard

Member contributions

- Ajay - Geo Data processing & Plotting (cases).
- Bhavya - Data collection & Preprocessing.
- Dinesh - Plotting (deaths) & Tools addition.
- Shardul - Plotting (vaccinations) & legend.