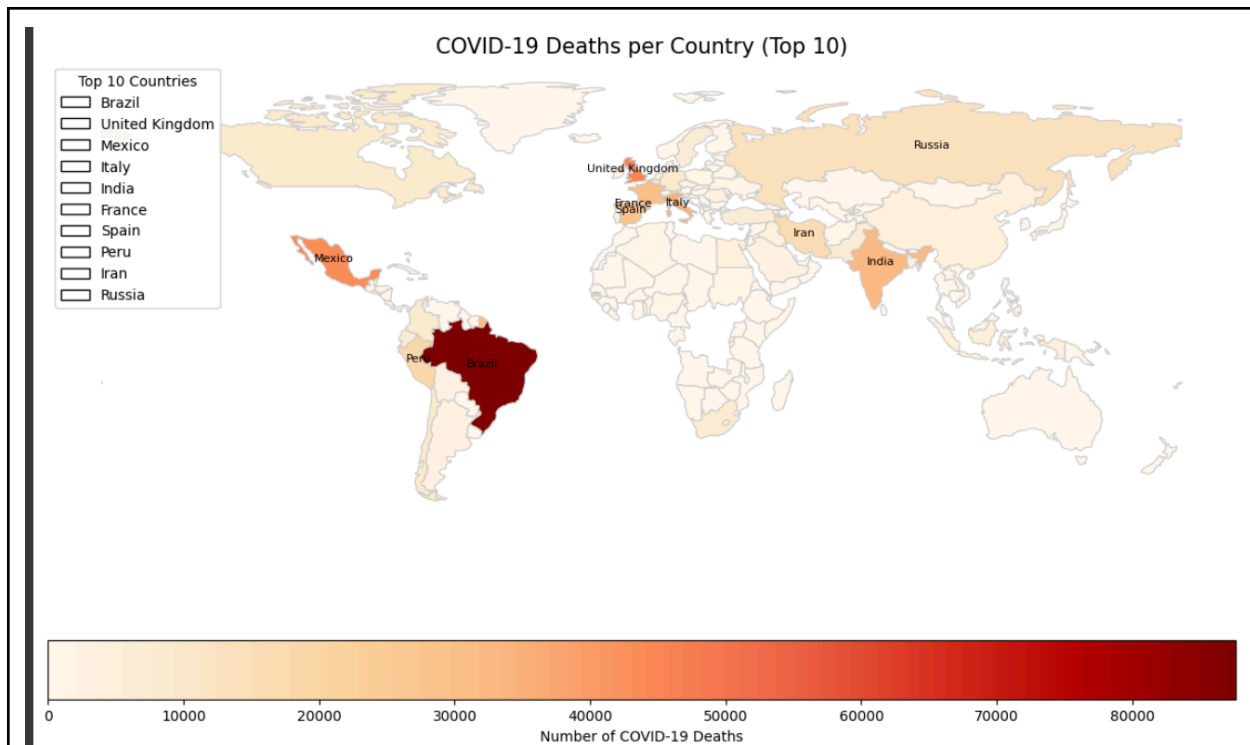


TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES
College of Computer Studies
Computer Science Department
ITE 405-CS33S1A - Applied Plotting, Charting, and Data Representation

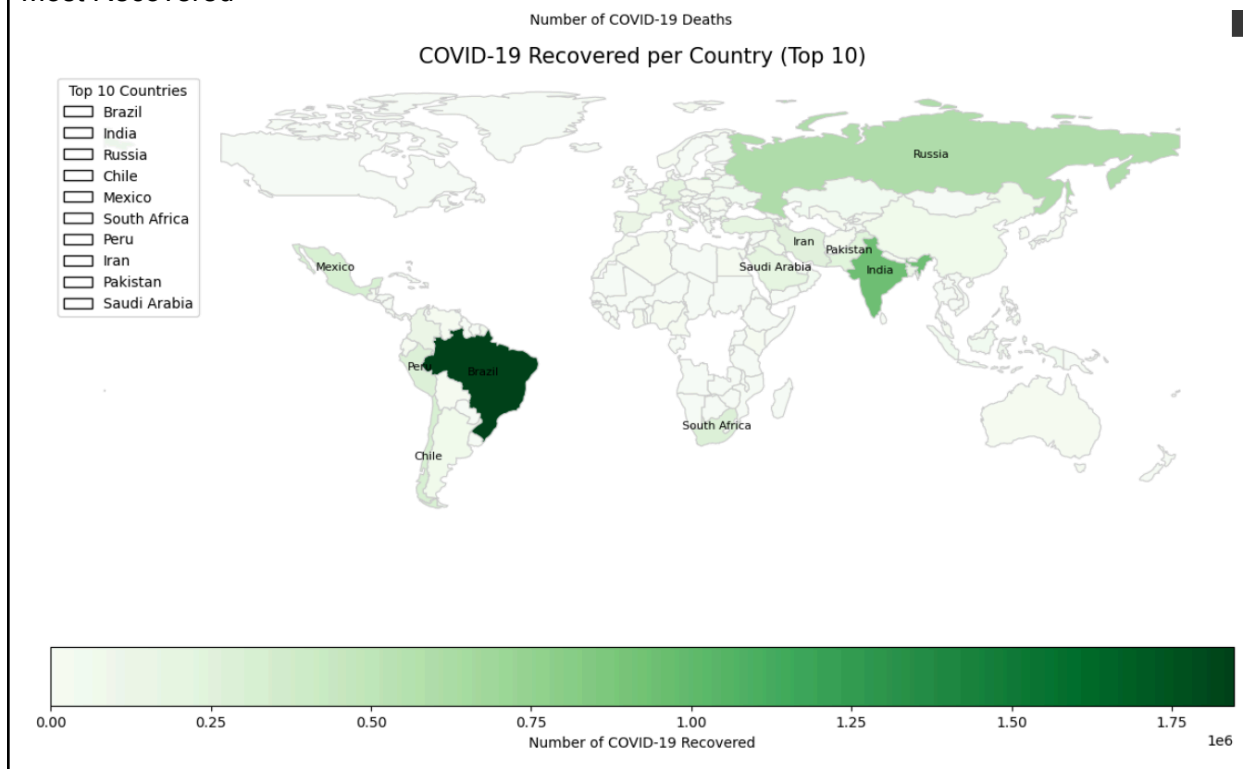
Name: Anioay, Kenneth Baniqued, Kevin De Ocampo, Deighro Evangelista, Danielle	Date: June 18, 2024
Program/Section: CS33S1	Instructor: Mr. Jess N. Garcia
Laboratory Exercise 4.3 Analyzing COVID-19 Data Using Heatmaps	
<p>Group Activity</p> <p>No. of Members: 2-4</p> <p>Objective:</p> <p>Students will analyze a COVID-19 data set and create five heatmaps to visualize different aspects of the data. This activity will help students understand how to interpret and present data effectively using heatmaps.</p> <p>Instructions:</p> <p>Obtain the Data Set</p> <p>Download the provided COVID-19 data set (Covid-19.csv)Download Covid-19.csv)</p> <p>Data Preparation</p> <p>Clean and preprocess the data to ensure it is ready for analysis.</p> <p>Create Heatmaps</p> <p>Create five different heatmaps</p> <p>Most deaths</p>	



Interpretation

In this heat map, among the top 10 countries with the most deaths it can be observed that Brazil has the most deaths, followed by the United Kingdom and Mexico.

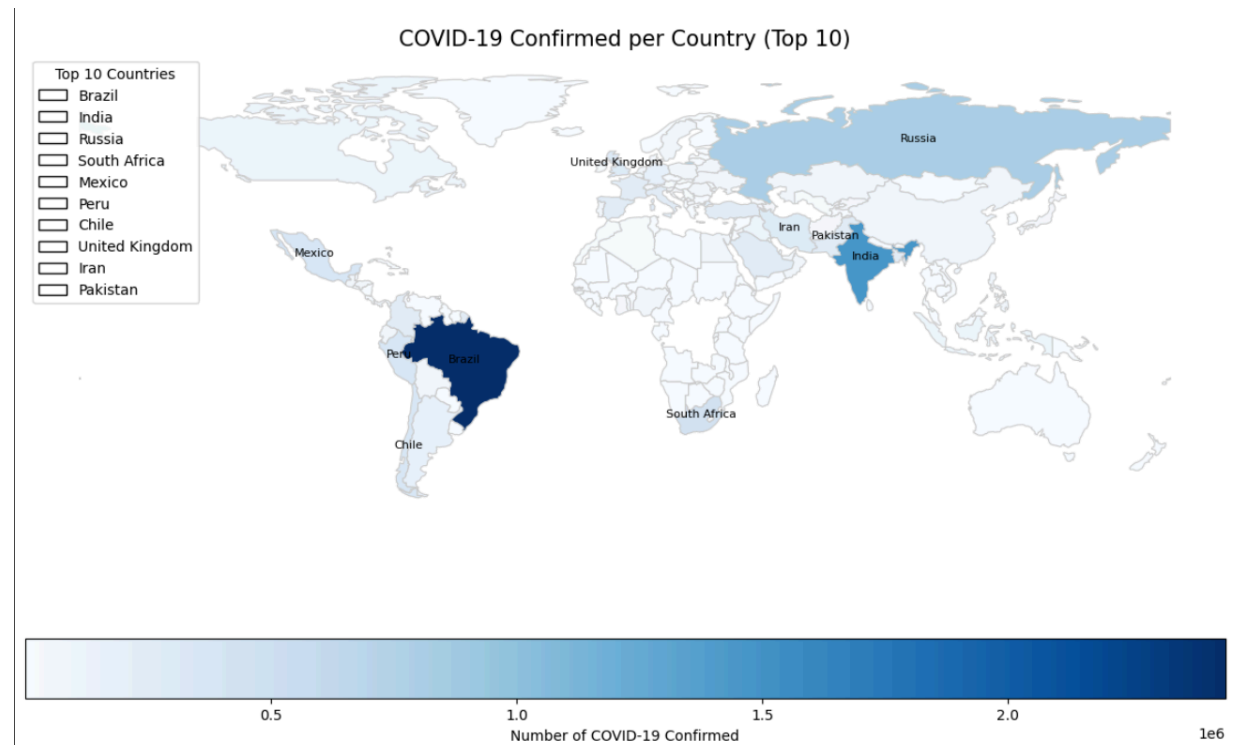
Most Recovered



Interpretation:

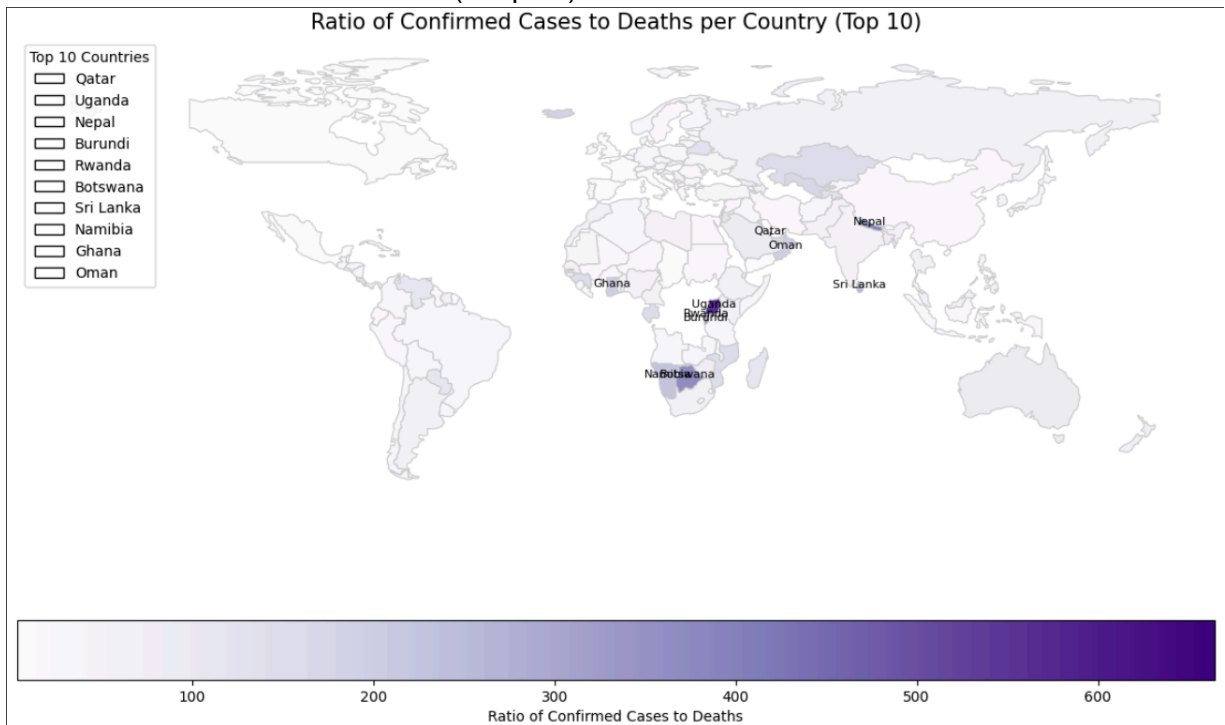
This heat map shows that Brazil has the most number of people who recovered from COVID-19 with India and Russia coming in 2nd and 3rd.

Most Confirmed

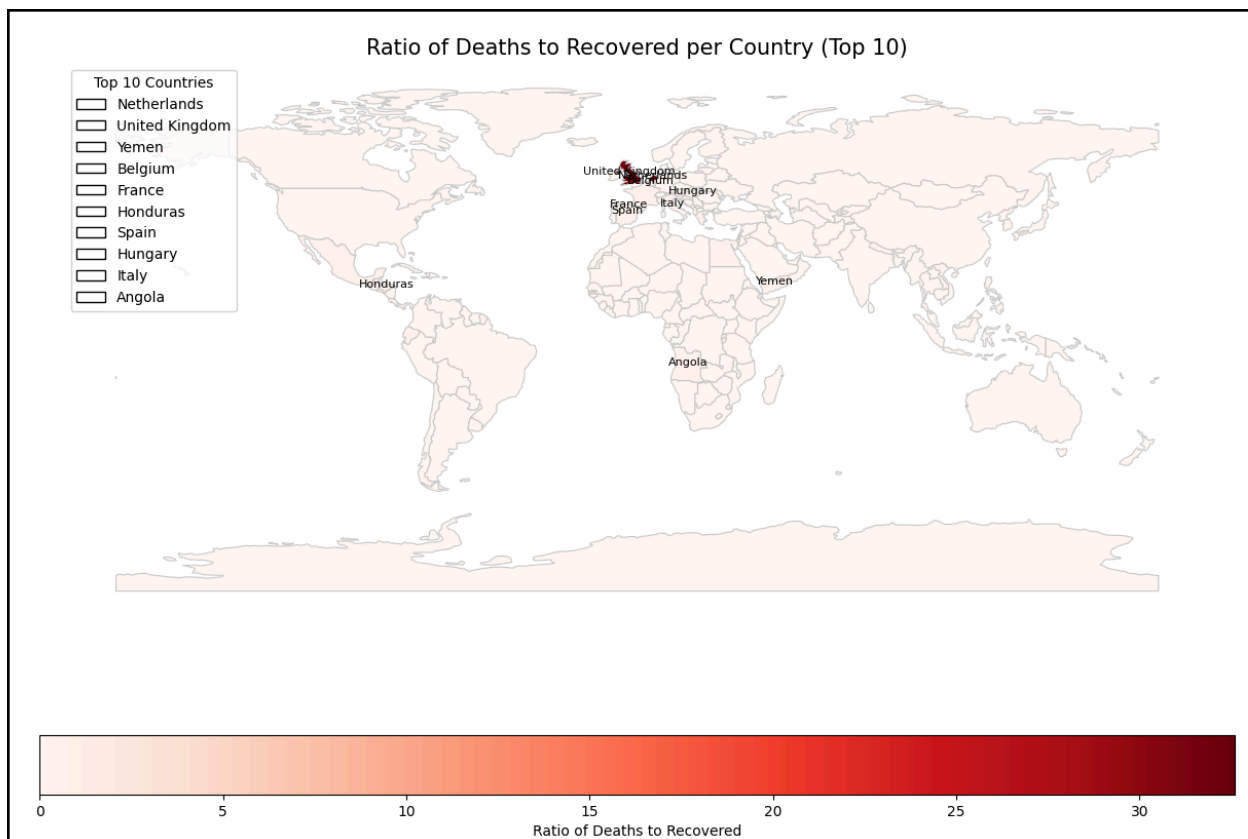


This heat map visualizes the ratio of confirmed COVID-19 cases to deaths per country, highlighting the top 10 countries with the highest ratios. Darker shades indicate higher ratios, with countries like Qatar, Uganda, Nepal, and others showing significantly high ratios, implying that these countries have a high number of confirmed cases relative to deaths.

Ratio of confirmed cases to deaths(eto pala)



The heatmap highlights the case fatality rates in different countries. Darker shades indicate higher ratios. Notable countries include Qatar, Uganda, Nepal, Burundi, and Rwanda. These nations have varying levels of impact, with different ratios of confirmed cases to deaths. A higher ratio could imply better management of the condition or underreporting of deaths; without additional context or data regarding testing rates and healthcare quality in these countries



Analysis and Interpretation

This is a heat map of a world map highlighting the top 10 countries by the death-to-recovery ratio from an event or disease. The countries are the Netherlands, UK, Yemen, Belgium, France, Honduras, Spain, Hungary, Italy, and Angola. A color scale from 0 to 30 shows a ratio that indicates the severity in each country.

Source Code:

```
import numpy as np
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
from matplotlib.patches import Patch
from matplotlib.lines import Line2D

# Function to create custom legend
def create_custom_legend(ax, top_10_countries):
    legend_elements = []
    for idx, row in top_10_countries.iterrows():
        patch = Patch(facecolor='none', edgecolor='black',
label=row['name'])
```

```

        legend_elements.append(patch)

    # Place the legend
    ax.legend(handles=legend_elements, loc='upper left', title="Top 10
Countries")

# COVID-19 Deaths per Country
deaths_per_country = df.groupby('Country')['Deaths'].sum().reset_index()
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
merged_data = world.merge(deaths_per_country, how='left',
left_on='name', right_on='Country')
top_10_countries = merged_data.sort_values(by='Deaths',
ascending=False).head(10)
fig, ax = plt.subplots(1, 1, figsize=(15, 10))
merged_data.plot(column='Deaths', cmap='OrRd', linewidth=0.8, ax=ax,
edgecolor='0.8', legend=True,
                legend_kwds={'label': "Number of COVID-19 Deaths",
'orientation': "horizontal"})
for idx, row in top_10_countries.iterrows():
    plt.annotate(text=row['name'], xy=(row['geometry'].centroid.x,
row['geometry'].centroid.y),
                horizontalalignment='center', fontsize=8,
color='black')
ax.set_title('COVID-19 Deaths per Country (Top 10)',
fontdict={'fontsize': '15', 'fontweight' : '3'})
ax.set_axis_off()
create_custom_legend(ax, top_10_countries)
plt.show()

# Most Recovered
recovered_per_country =
df.groupby('Country')['Recovered'].sum().reset_index()
merged_data_recovered = world.merge(recovered_per_country, how='left',
left_on='name', right_on='Country')
top_10_countries_recovered =
merged_data_recovered.sort_values(by='Recovered',
ascending=False).head(10)
fig, ax = plt.subplots(1, 1, figsize=(15, 10))

```

```

merged_data_recovered.plot(column='Recovered', cmap='Greens',
linewidth=0.8, ax=ax, edgecolor='0.8', legend=True,
                           legend_kwds={'label': "Number of COVID-19
Recovered", 'orientation': "horizontal"})
for idx, row in top_10_countries_recovered.iterrows():
    plt.annotate(text=row['name'], xy=(row['geometry'].centroid.x,
row['geometry'].centroid.y),
                 horizontalalignment='center', fontsize=8,
color='black')
ax.set_title('COVID-19 Recovered per Country (Top 10)',
fontdict={'fontsize': '15', 'fontweight' : '3'})
ax.set_axis_off()
create_custom_legend(ax, top_10_countries_recovered)
plt.show()

# Most Confirmed
confirmed_per_country =
df.groupby('Country')['Confirmed'].sum().reset_index()
merged_data_confirmed = world.merge(confirmed_per_country, how='left',
left_on='name', right_on='Country')
top_10_countries_confirmed =
merged_data_confirmed.sort_values(by='Confirmed',
ascending=False).head(10)
fig, ax = plt.subplots(1, 1, figsize=(15, 10))
merged_data_confirmed.plot(column='Confirmed', cmap='Blues',
linewidth=0.8, ax=ax, edgecolor='0.8', legend=True,
                           legend_kwds={'label': "Number of COVID-19
Confirmed", 'orientation': "horizontal"})
for idx, row in top_10_countries_confirmed.iterrows():
    plt.annotate(text=row['name'], xy=(row['geometry'].centroid.x,
row['geometry'].centroid.y),
                 horizontalalignment='center', fontsize=8,
color='black')
ax.set_title('COVID-19 Confirmed per Country (Top 10)',
fontdict={'fontsize': '15', 'fontweight' : '3'})
ax.set_axis_off()
create_custom_legend(ax, top_10_countries_confirmed)
plt.show()

```

```

# Ratio of Confirmed to Deaths
df['Confirmed_to_Deaths'] = df['Confirmed'] / df['Deaths'].replace({0:
np.nan})
ratio_confirmed_to_deaths_per_country =
df.groupby('Country')['Confirmed_to_Deaths'].mean().reset_index()
merged_data_ratio_confirmed_to_deaths =
world.merge(ratio_confirmed_to_deaths_per_country, how='left',
left_on='name', right_on='Country')
top_10_countries_ratio_confirmed_to_deaths =
merged_data_ratio_confirmed_to_deaths.sort_values(by='Confirmed_to_Death
s', ascending=False).head(10)
fig, ax = plt.subplots(1, 1, figsize=(15, 10))
merged_data_ratio_confirmed_to_deaths.plot(column='Confirmed_to_Deaths',
cmap='Purples', linewidth=0.8, ax=ax, edgecolor='0.8', legend=True,
legend_kwds={'label': "Ratio
of Confirmed Cases to Deaths", 'orientation': "horizontal"})
for idx, row in top_10_countries_ratio_confirmed_to_deaths.iterrows():
    plt.annotate(text=row['name'], xy=(row['geometry'].centroid.x,
row['geometry'].centroid.y),
horizontalalignment='center', fontsize=8,
color='black')
ax.set_title('Ratio of Confirmed Cases to Deaths per Country (Top 10)',
fontdict={'fontsize': '15', 'fontweight' : '3'})
ax.set_axis_off()
create_custom_legend(ax, top_10_countries_ratio_confirmed_to_deaths)
plt.show()

# Ratio of Deaths to Recovered
df['Deaths_to_Recovered'] = df['Deaths'] / df['Recovered'].replace({0:
np.nan})
ratio_deaths_to_recovered_per_country =
df.groupby('Country')['Deaths_to_Recovered'].mean().reset_index()
merged_data_ratio_deaths_to_recovered =
world.merge(ratio_deaths_to_recovered_per_country, how='left',
left_on='name', right_on='Country')
merged_data_ratio_deaths_to_recovered['Deaths_to_Recovered'] =
merged_data_ratio_deaths_to_recovered['Deaths_to_Recovered'].fillna(0)

```



```
top_10_countries_ratio_deaths_to_recovered =
merged_data_ratio_deaths_to_recovered.sort_values(by='Deaths_to_Recover
d', ascending=False).head(10)
fig, ax = plt.subplots(1, 1, figsize=(15, 10))
merged_data_ratio_deaths_to_recovered.plot(column='Deaths_to_Recovered',
cmap='Reds', linewidth=0.8, ax=ax, edgecolor='0.8', legend=True,
                                         legend_kwds={'label': "Ratio
of Deaths to Recovered", 'orientation': "horizontal"})
for idx, row in top_10_countries_ratio_deaths_to_recovered.iterrows():
    plt.annotate(text=row['name'], xy=(row['geometry'].centroid.x,
row['geometry'].centroid.y),
                horizontalalignment='center', fontsize=8,
color='black')
ax.set_title('Ratio of Deaths to Recovered per Country (Top 10)',
fontdict={'fontsize': '15', 'fontweight' : '3'})
ax.set_axis_off()
create_custom_legend(ax, top_10_countries_ratio_deaths_to_recovered)
plt.show()
```
