Assignment 3 - Visual Mapping

Weilin Ruan 50018083 Group5

Q1: What is the data type of each data attribute (*date, location, new_cases*, …)?

Data Types of Each Data Attribute

- date: String (can be converted to datetime for analysis)
- location: String
- new_cases: Integer
- new_deaths: Integer
- total_cases: Integer
- total_deaths: Integer
- weekly_cases: Integer
- weekly_deaths: Integer
- bi_weekly_cases: Integer
- bi_weekly_deaths: Integer

Q2. Describe what you have learned from the data and the visualization. For instance, which country has the most number of total cases/deaths? Which country has the highest case-fatality rate?

- **Country with Most Total Cases/Deaths**: From the visualization, you can quickly identify which country has the largest values for total_cases and total_deaths by looking at the top-right corner.
- **Highest Case-Fatality Rate**: The country with the highest case-fatality rate can be identified by hovering over the data points and comparing the calculated rates.

Extension of the Scatterplot

To add further information such as population and continent:

- **Population**: Use the size of each data point to represent the population. Larger circles can indicate larger populations.
- **Continent**: Use colors to differentiate between continents. Assign a different color to each continent to enable quick visual differentiation.

This mapping can be achieved by merging additional data containing population and continent information with the main dataset and using these attributes in the visual encoding.

Coding

To complete the task, we will visualize total_cases and total_deaths on the latest day in the dataset using a scatter plot. We will also include interactivity to display the country/region name and case-fatality rate on hover.

```
import pandas as pd
import plotly.express as px

# Load the data
data_url = 'https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/jhu/full_data.csv' # get newest data
df = pd.read_csv(data_url)
# df = pd.read_csv("full_data-1.csv") # or use local data


# Get the latest date in the data
latest_date = df['date'].max()

# Filter the data for the latest date
latest_data = df[df['date'] == latest_date].copy()

# Calculate case-fatality rate using .loc
latest_data.loc[:, 'case_fatality_rate'] = (latest_data['total_deaths'] / latest_data['total_cases']) * 100

# Create the scatter plot
fig = px.scatter(
    latest_data,
    x='total_cases',
    y='total_deaths',
    text='location',
    hover_data={
        'location': True,
        'case_fatality_rate': ':.2f',
        'total_cases': True,
        'total_deaths': True
    },
    labels={'total_cases': 'Total Cases', 'total_deaths': 'Total Deaths'},
    title='COVID-19 Total Cases vs Total Deaths',
    log_x=True,
    log_y=True
```

```
)

# Update layout for better display
fig.update_traces(marker=dict(size=20, opacity=0.7),
                  selector=dict(mode='markers+text'))
fig.update_layout(
    hovermode='closest',
    width=2000,  # Set the width of the figure
    height=1200   # Set the height of the figure
)

# Show the plot
fig.show()
```

Results:

Figure 1 interaction of the scatterplot

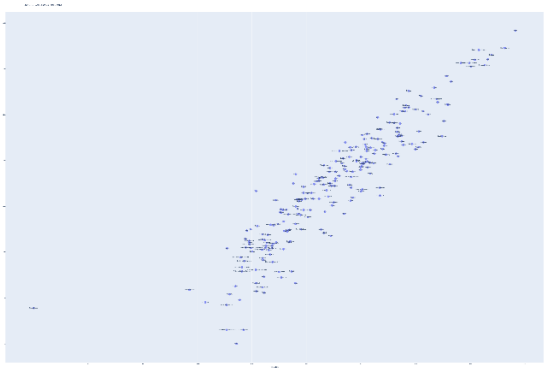

Figure 2 scatter plot snapshot