

Assignment 3 — Siru ZHONG

Questions

1. Data attribute type for new_cases

- **Data attribute type:** `float`, as it represents a count of new cases reported.
- **Colormap type:** `sequential`, as it is a quantitative variable, a sequential colormap would be appropriate as it shows a progress from low to high values.

2. Data attribute type for location

- **Data attribute type:** `string`, as it represents categorical non-ordered data.
- **Colormap type:** `categorical` colormap is suitable because it represents discrete, non-quantitative categories.

Coding

1. Scatter Plot with Colormap Based on new_cases

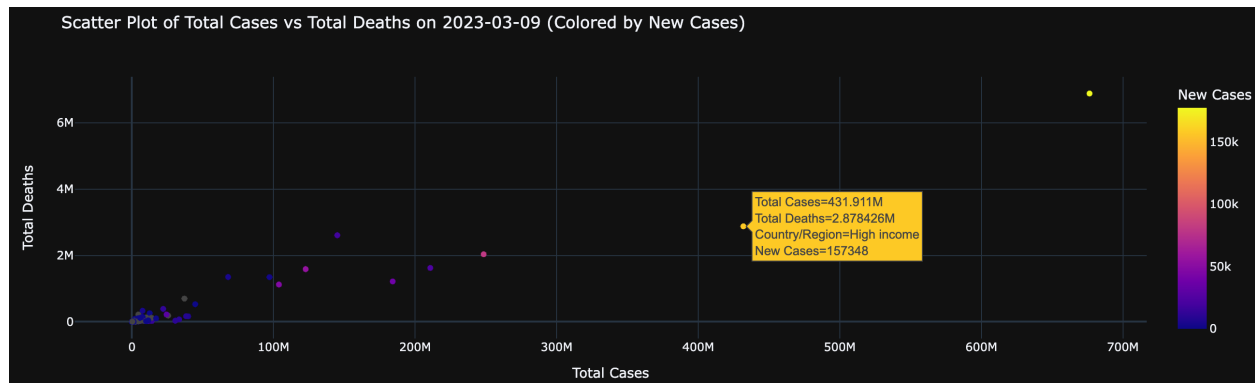
```
import pandas as pd
import plotly.express as px

# Reload the data from the uploaded file
data = pd.read_csv("full_data.csv")

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

# Create scatter plot for total_cases vs total_deaths with colors based on new_cases
fig_new_cases = px.scatter(
    latest_data,
    x='total_cases',
    y='total_deaths',
    color='new_cases', # Assigning colors based on new_cases using default sequential colormap
    hover_data=['location'],
    labels={
        'total_cases': 'Total Cases',
        'total_deaths': 'Total Deaths',
        'location': 'Country/Region',
        'new_cases': 'New Cases'
    },
    title=f"Scatter Plot of Total Cases vs Total Deaths on {latest_date} (Colored by New Cases)"
)

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



2. Scatter Plot with Colormap Based on location and Point Size Based on new_cases

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

# Load the data from the CSV file
data = pd.read_csv("full_data.csv")

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

# Handle NaN values in new_cases by replacing them with 0
latest_data['new_cases'].fillna(0, inplace=True)

# Calculate a size factor for the new_cases - this will control the size of the scatter plot points
# You might need to adjust the scaling factor to get a desirable visual result
latest_data['size'] = latest_data['new_cases'] / latest_data['new_cases'].max() * 100

# Create scatter plot for total_cases vs total_deaths
# Colors are based on location and sizes are based on new_cases
fig_location_new_cases = px.scatter(
    latest_data,
    x='total_cases',
    y='total_deaths',
    color='location', # Assigning colors based on location
    size='size', # Adjusting sizes based on new_cases
    hover_data=['location', 'new_cases'],
    labels={
        'total_cases': 'Total Cases',
        'total_deaths': 'Total Deaths',
        'new_cases': 'New Cases'
    },
    title=f"Scatter Plot of Total Cases vs Total Deaths on {latest_date} (Colored by Location, Sized by New Cases)"
)

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

