Data Visualization

Lecture 4

Data Visualization with Seaborn & Folium

Seaborn Regression Plots

- The regression plots in seaborn are primarily intended to add a visual guide that helps to emphasize patterns in a dataset during exploratory data analyses.

- Regression plots as the name suggests creates a regression line between 2 parameters and helps to visualize their linear relationships.

Seaborn Regression Plots

- Seaborn is not only a visualization library but also a provider of built-in datasets.

- We will use a dataset named ‘tips’.
  - This dataset contains information about the people who probably had food at the restaurant and whether or not they left a tip.
  - It also provides information about the gender of the people, whether they smoke, day, time and so on.

Seaborn Regression Plots

- Example:
  - Load the dataset and show the first 5 entries:

```
import seaborn as sns
dataset = sns.load_dataset('tips')
dataset.head()
```

  - Draw the regression plot (simple linear plot):

```
sns.set_style('whitegrid')
sns.lmplot(x ='total_bill', y ='tip', data = dataset)
```

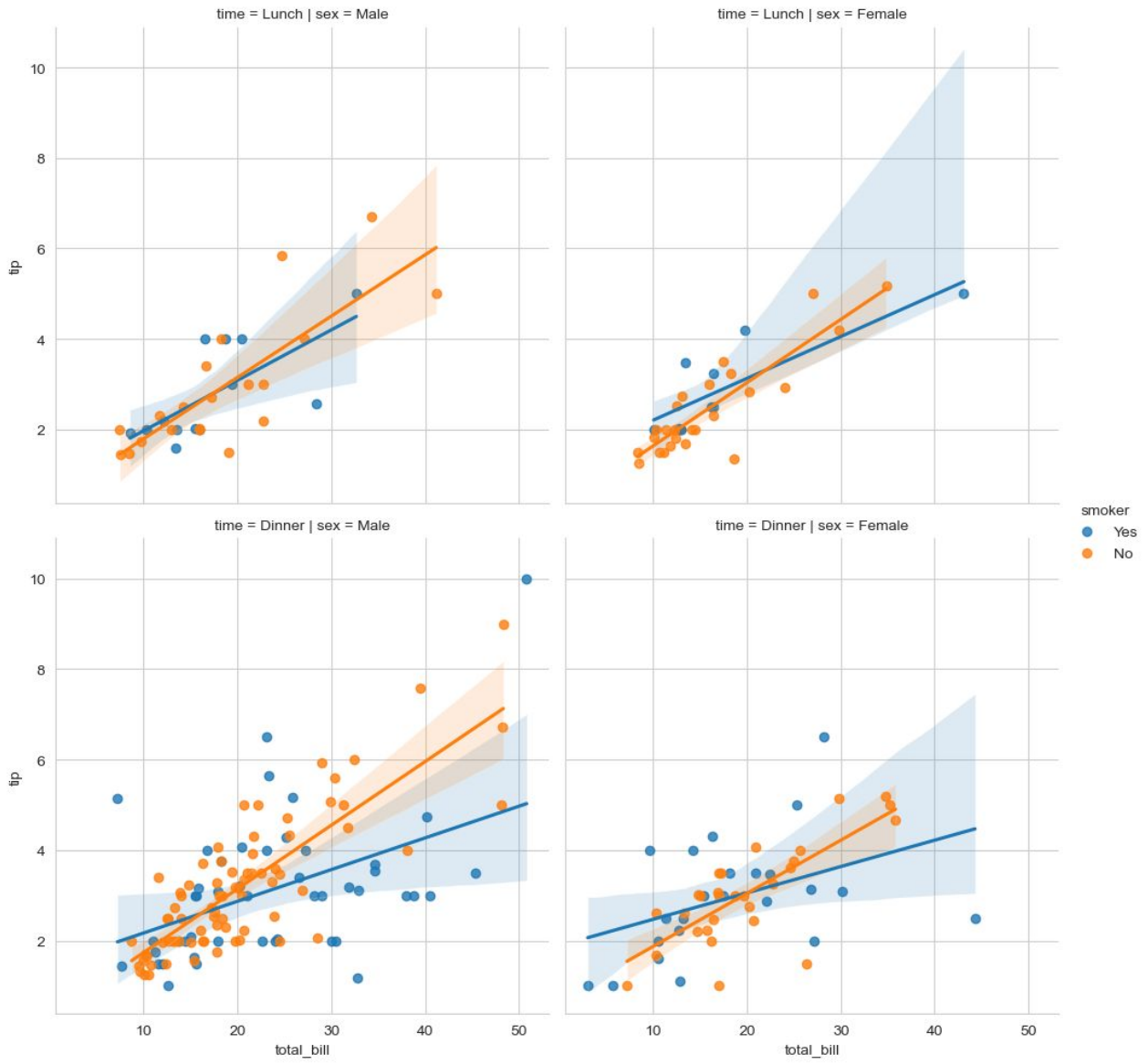
Seaborn Regression Plots

- Example:
  - Regression plot with more parameters:

```
import matplotlib.pyplot as plt
sns.lmplot(x ='total_bill', y ='tip', data = dataset, col ='sex', row ='time', hue ='smoker’)
plt.show()
```

Seaborn Regression Plots

- Example:
  - Regression plot with more parameters:



Seaborn Regression Plots

- Exercise:
  - The "iris" dataset from seaborn contains information about the physical characteristics of three species of iris flowers: Setosa, Versicolor, and Virginica.
  - It includes 150 observations and 5 variables such as sepal length, sepal width, petal length, petal width, and the species of the flower.
  - The goal of this dataset is to predict the species of iris flowers based on their physical characteristics.
  - Write a code that loads the dataset and draws the regression plot.

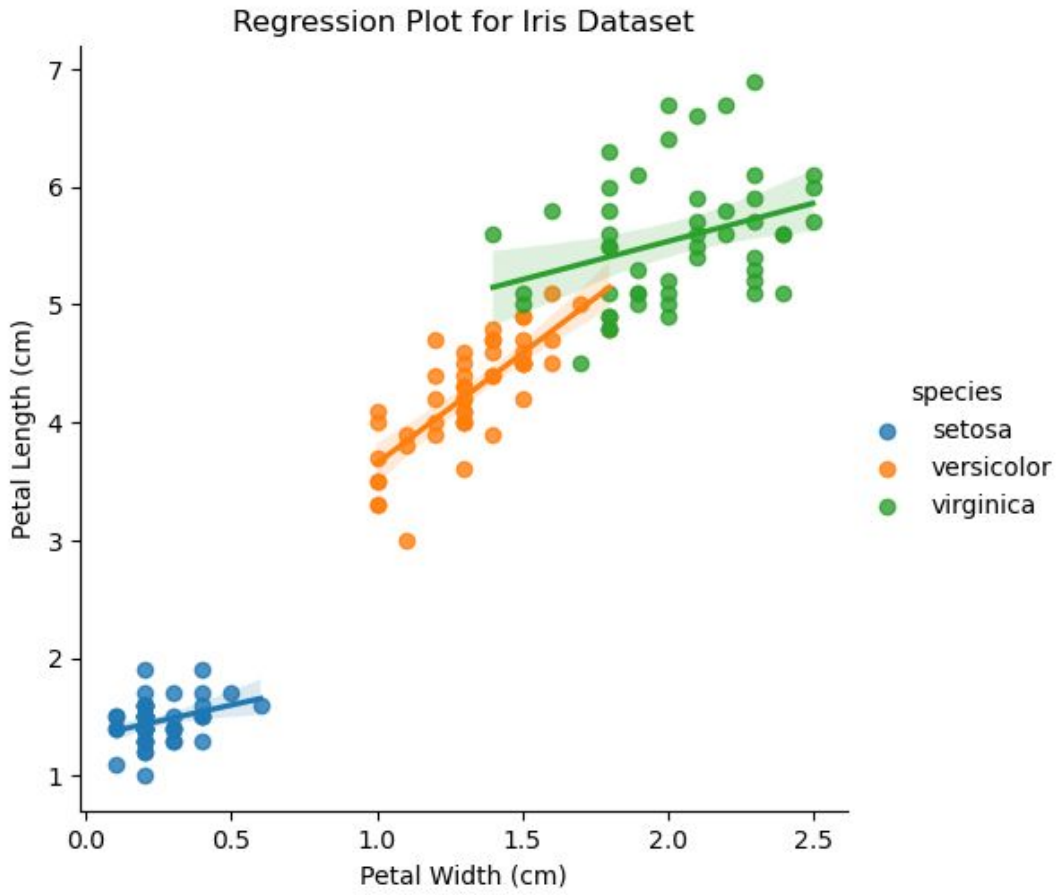
Seaborn Regression Plots

- Exercise:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Load the iris dataset from seaborn library
iris = sns.load_dataset("iris")

# Create the regression plot
sns.lmplot(x="petal_width", y="petal_length", hue="species", data=iris)

# Set the title and axes labels
plt.title("Regression Plot for Iris Dataset")
plt.xlabel("Petal Width (cm)")
plt.ylabel("Petal Length (cm)")

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

Seaborn Regression Plots

- Exercise:
  - Output:



Folium Geospatial Data

- Folium is a Python library used for visualizing geospatial data.
- It is easy to use and yet a powerful library.
- How to install Folium using Jupyter notebook?

```
!pip install folium
```

Folium Geospatial Data

- Let’s draw Egypt’s map:

```
import folium
m=folium.Map(location=[26.8206, 30.8025])
m
```

- You can see that these maps are interactive:
  - You can zoom in and out by clicking the positive and negative buttons in the top-left corner of the map.
  - You can also drag the map and see different regions.

Folium Geospatial Data

- We can resize the map and change the zoom level:

```
import folium
m=folium.Map(location=[26.8206, 30.8025])
M

from branca.element import Figure
fig=Figure(width=550,height=350)

m1=folium.Map(width=550,height=350,location=[26.8206,
30.8025],zoom_start=11,min_zoom=8,max_zoom=14)
fig.add_child(m1)
m1
```

Folium Geospatial Data

- We can resize the map and change the zoom level:

Folium Geospatial Data

- Layers and Tiles:
  - Each tileset shows different features of a map and is suitable for different purposes.
  - For example, Stamen Terrain features hill shading and natural vegetation colors. It showcases advanced labeling and linework generalization of dual-carriageway roads. While CartoDB Dark Matter shows the CartoDB Positron map in dark mode.

Folium Geospatial Data

- Layers and Tiles:

```
fig2=Figure(width=550,height=350)
m2=folium.Map(location=[26.8206, 30.8025])
fig2.add_child(m2)
folium.TileLayer('Stamen Terrain').add_to(m2)
folium.TileLayer('cartodbpositron').add_to(m2)
folium.TileLayer('cartodbdark_matter').add_to(m2)
folium.LayerControl().add_to(m2)
m2
```

Folium Geospatial Data

- Layers and Tiles:

Folium Geospatial Data

- Plotting Markers on the map:
  - Markers are the items used for marking a location on a map.
    - For example, when you use Google Maps for navigation, your location is marked by a marker and your destination is marked by another marker.
  - Markers are among the most important and helpful things on a map.
  - Folium gives a folium.Marker() class for plotting markers on a map.
    - Just pass the latitude and longitude of the location, mention the popup and tooltip and add it to the map.

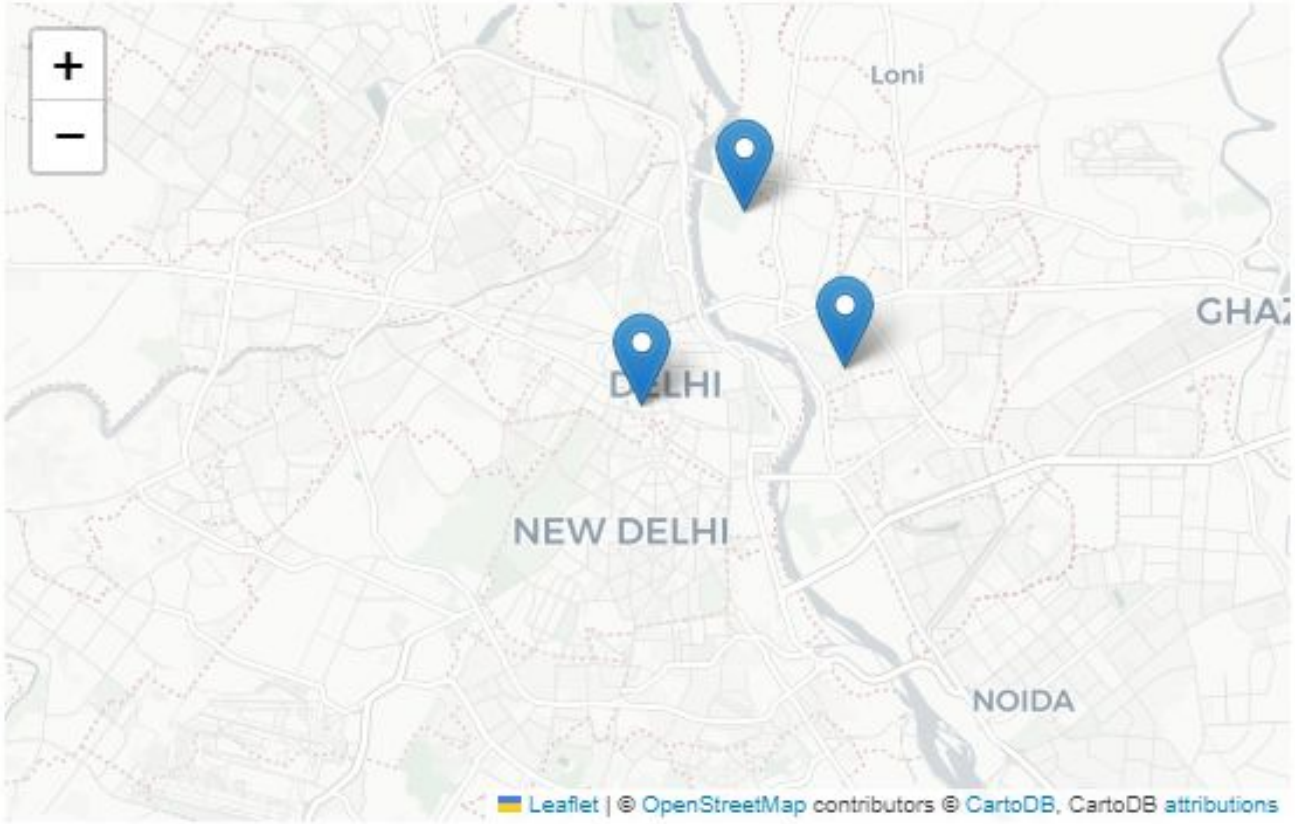
Folium Geospatial Data

- Plotting Markers on the map:

```
# Creating Basemap
fig3=Figure(width=550,height=350)
m3=folium.Map(location=[28.644800, 77.216721],tiles='cartodbpositron',zoom_start=11)
fig3.add_child(m3)

#Adding markers to the map
folium.Marker(location=[28.695800, 77.244721],popup='Default popup Marker1',tooltip='Click here to see Popup').add_to(m3)
folium.Marker(location=[28.645800,77.214721],popup='<strong>Marker3</strong>',tooltip='<strong>Click here to see Popup</strong>').add_to(m3)
folium.Marker(location=[28.655800, 77.274721],popup=‘<h3 style="color:green;">Marker2</h3>',tooltip='<strong>Click here to see Popup</strong>').add_to(m3)

m3
```

Folium Geospatial Data

- Plotting Markers on the map:



Folium Geospatial Data

- Change icon shape:
  - Folium gives the folium.Icon() class which can be used for creating custom icons for markers.

  - Icon() takes three arguments – color, prefix and icon:
    - Color is used for changing the color of the marker
    - Prefix is used for selecting the icon provider (fa for Fontawesome and glyphicon for Glyphicons)
    - Icon is used for selecting the icon name

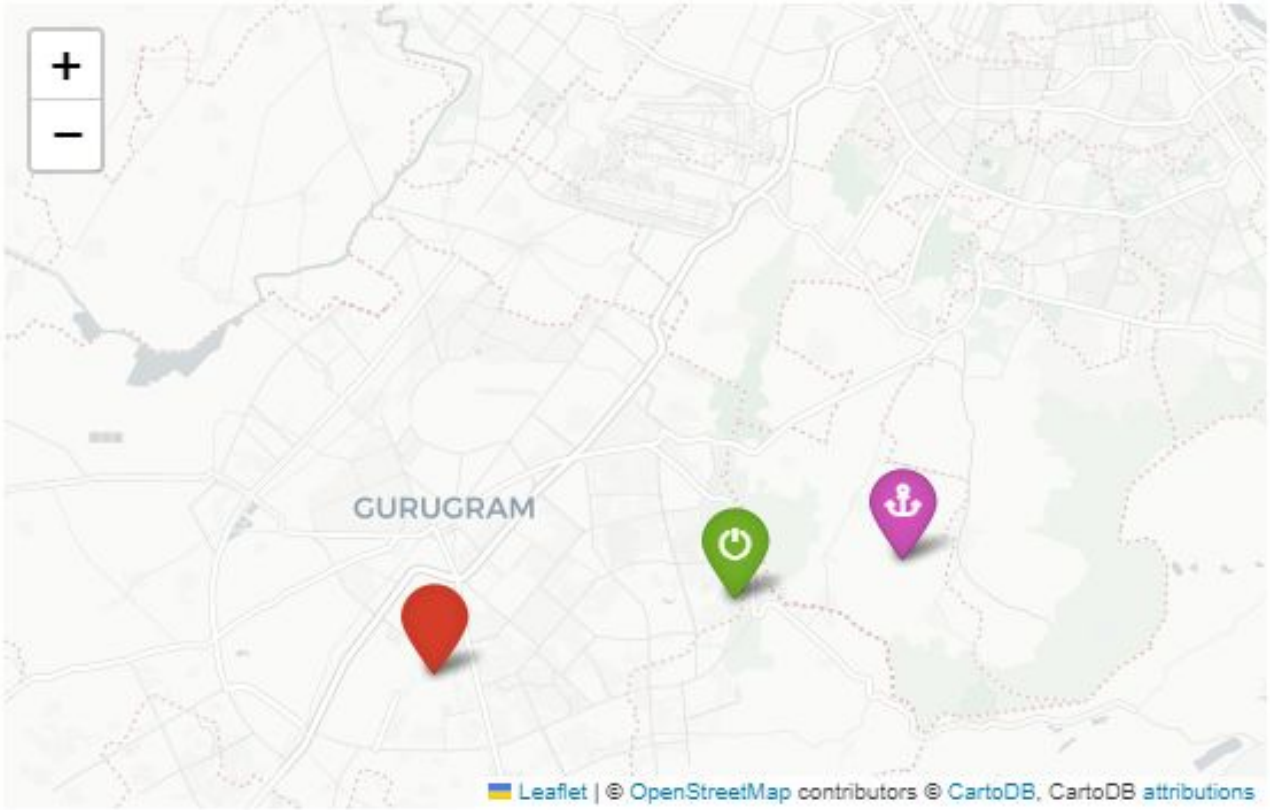
Folium Geospatial Data

- Change icon shape:

```
# Creating Basemap
fig4=Figure(height=350,width=550)
m4=folium.Map(location=[28.4911091,77.0867361],tiles='cartodbpositron',zoom_start=11)
fig4.add_child(m4)

# Adding Custom Markers
folium.Marker(location=[28.4211091,77.0267361],popup='Custom Marker 1',tooltip='<strong>Click here to see Popup</strong>',icon=folium.Icon(color='red',icon='none')).add_to(m4)
folium.Marker(location=[28.4411091,77.1167361],popup='Custom Marker 2',tooltip='<strong>Click here to see Popup</strong>',icon=folium.Icon(color='green',prefix='glyphicon',icon='off')).add_to(m4)
folium.Marker(location=[28.4511091,77.1667361],popup='Custom Marker 3',tooltip='<strong>Click here to see Popup</strong>',icon=folium.Icon(color='purple',prefix='fa',icon='anchor')).add_to(m4)

m4
```

Folium Geospatial Data

- Change icon shape:



Folium Geospatial Data

- Exercise:
  1. Create a map centered on Cairo (30.0444, 31.2357) with initial zoom level=12.
  2. Add markers for:
     - Tahrir Square (30.0444, 31.2357)
     - Pyramids of Giza (29.9792, 31.1342)
     - Citadel of Cairo (30.0295, 31.2615)
  3. Create a layer for museums:
     - Egyptian Museum (30.0478, 31.2336)
     - Museum of Islamic Art (30.0474, 31.2294)
  4. Create a layer for parks:
     - Al-Azhar Park (30.0326, 31.2645)
     - International Garden (29.9682, 31.2695)
  5. Add three different tile layers to the map: 'Stamen Toner’, 'cartodbpositron’, 'cartodbdark_matter'

Folium Geospatial Data

- Code:

```
import folium
# Create a map centered on Cairo
cairo_map = folium.Map(location=[30.0444, 31.2357], zoom_start=12)

# Add a marker for Tahrir Square
folium.Marker(location=[30.0444, 31.2357], popup='Tahrir Square').add_to(cairo_map)

# Add a marker for the Pyramids of Giza
folium.Marker(location=[29.9792, 31.1342], popup='Pyramids of Giza').add_to(cairo_map)

# Add a marker for the Citadel of Cairo
folium.Marker(location=[30.0295, 31.2615], popup='Citadel of Cairo').add_to(cairo_map)
```

Folium Geospatial Data

- Code:

```
# Create a layer group for museums
museums = folium.FeatureGroup(name='Museums')

folium.Marker(location=[30.0478, 31.2336], popup='Egyptian Museum').add_to(museums)

folium.Marker(location=[30.0474, 31.2294], popup='Museum of Islamic Art').add_to(museums)

# Add the museums layer group to the map
museums.add_to(cairo_map)
```

Folium Geospatial Data

- Code:

```
# Create a layer group for parks
parks = folium.FeatureGroup(name='Parks')

folium.Marker(location=[30.0326, 31.2645], popup='Al-Azhar Park').add_to(parks)

folium.Marker(location=[29.9682, 31.2695], popup='International Garden').add_to(parks)

parks.add_to(cairo_map)
```

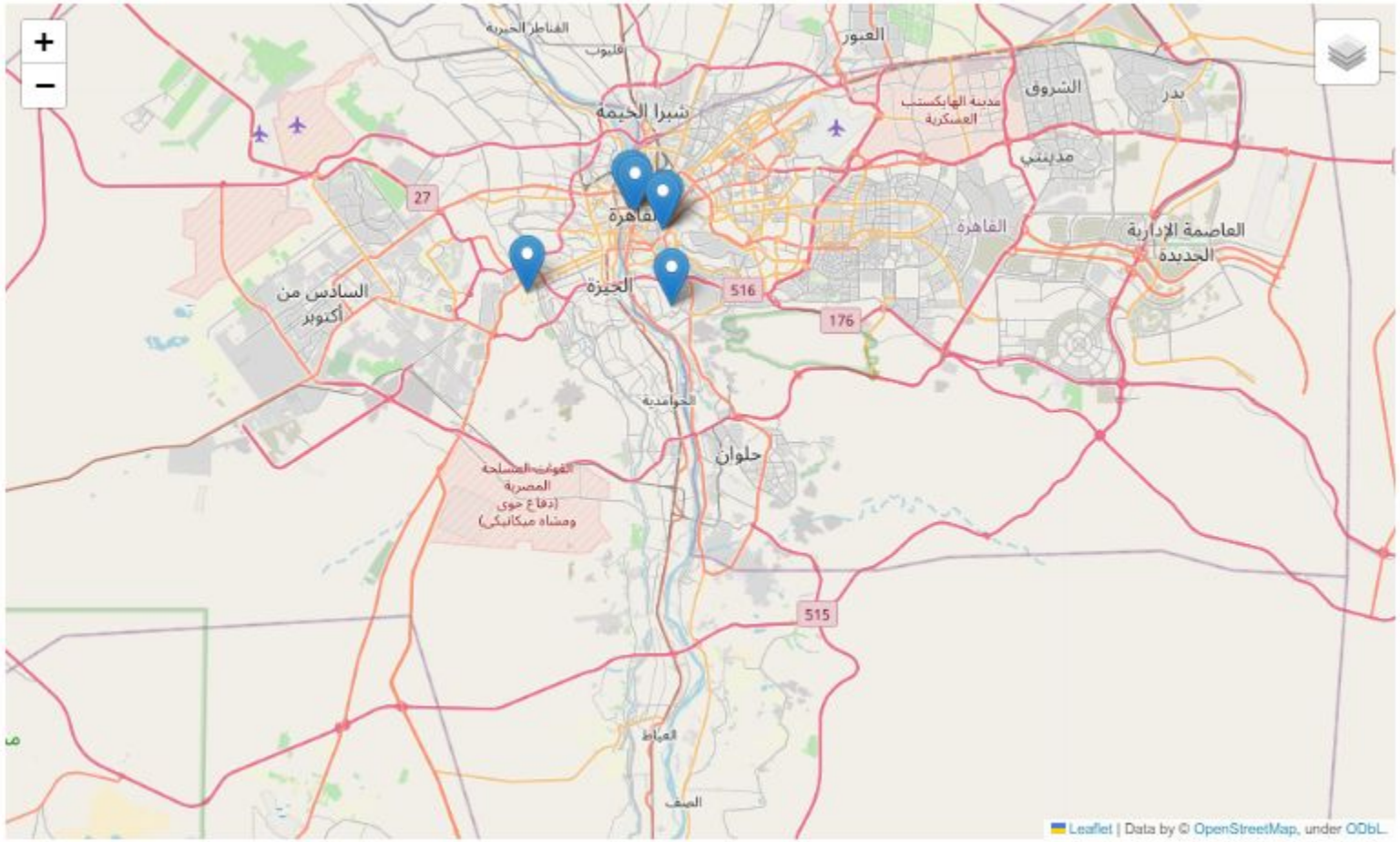
Folium Geospatial Data

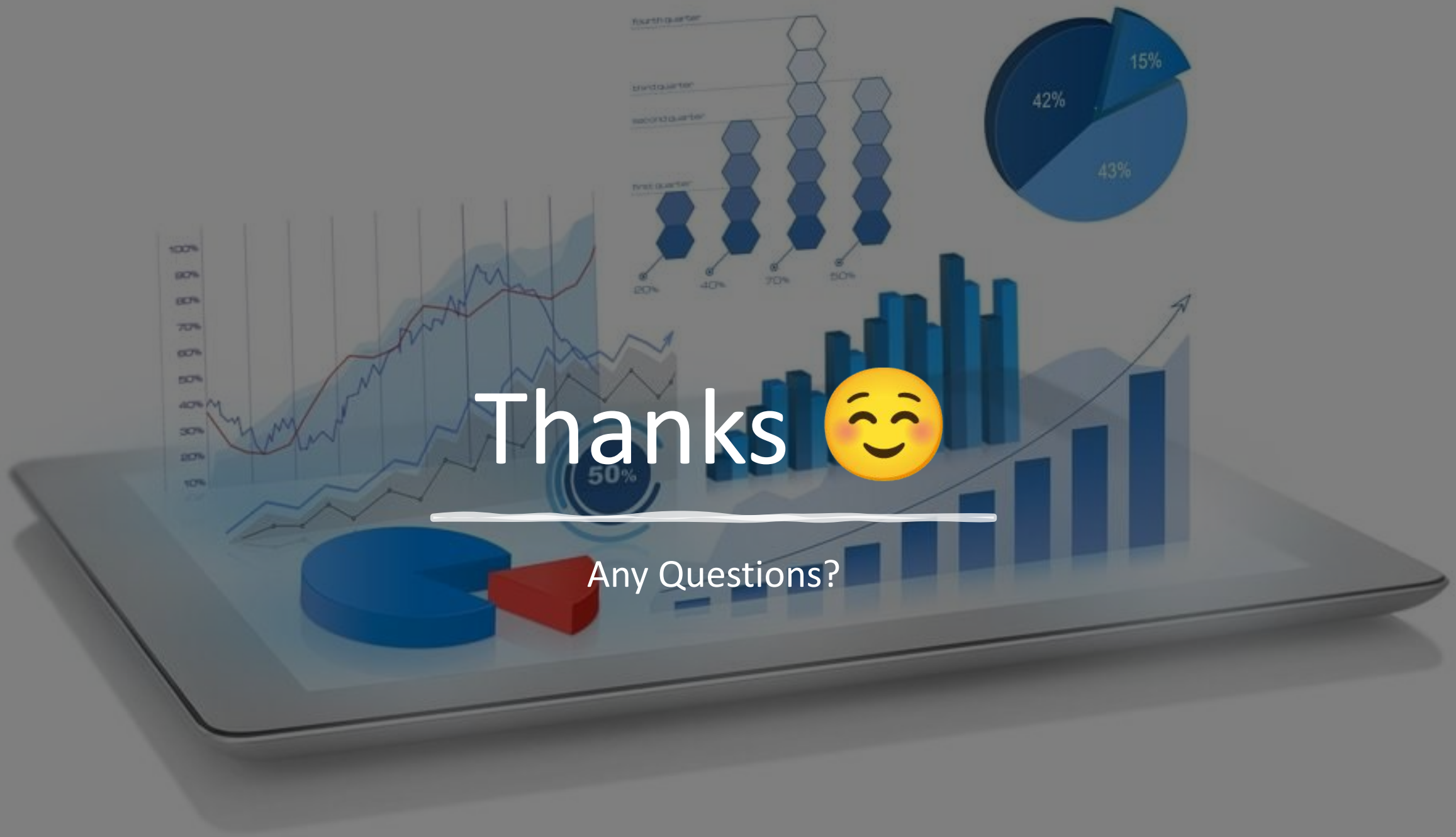
- Code:

```
# Add a tile layer for the map
folium.TileLayer('cartodbpositron').add_to(cairo_map)
folium.TileLayer('Stamen Toner').add_to(cairo_map)
folium.TileLayer('cartodbdark_matter').add_to(cairo_map)

# Add a layer control to the map
folium.LayerControl().add_to(cairo_map)

# Display the map
cairo_map
```

Folium Geospatial Data

- Output:





Thanks 😊

Any Questions?