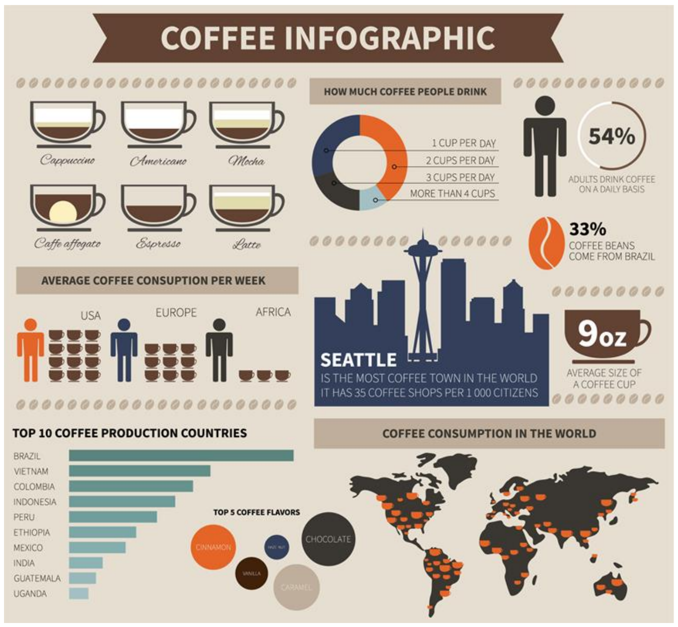
Data Visualization

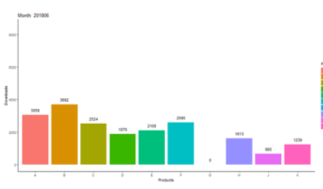
CCDATSCL | COM 221 - ML

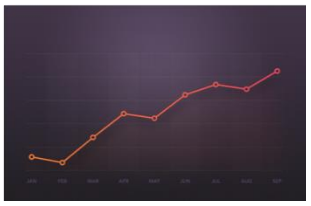
Data visualization is the practice of turning data into visual representation such as charts, plots, infographics, and even animations.

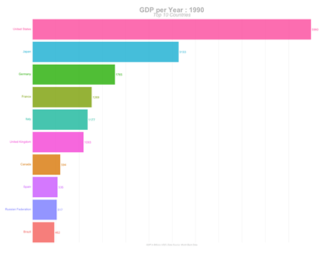
These visual displays of information communicate complex data relationships and data-driven insights in a way that is easy to understand.

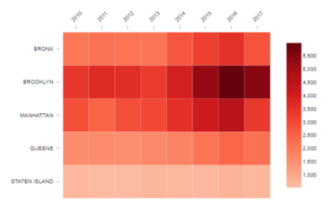
It is used to help people see patterns, trends, and insights more clearly than they could from raw numbers or tables.

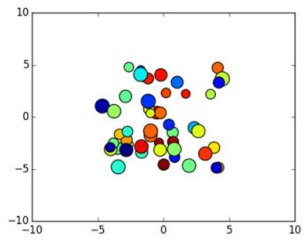

COFFEE INFOGRAPHIC
Cappuccino
Americano
Mocha
Caffe affogato
Espresso
Latte
HOW MUCH COFFEE PEOPLE DRINK
1 CUP PER DAY
2 CUPS PER DAY
3 CUPS PER DAY
MORE THAN 4 CUPS
54%
ADULTS DRINK COFFEE ON A DAILY BASIS
33%
COFFEE BEANS COME FROM BRAZIL
AVERAGE COFFEE CONSUPTION PER WEEK
USA
EUROPE
AFRICA
SEATTLE
IS THE MOST COFFEE TOWN IN THE WORLD
IT HAS 35 COFFEE SHOPS PER 1 000 CITIZENS
9oz
AVERAGE SIZE OF A COFFEE CUP
TOP 10 COFFEE PRODUCTION COUNTRIES
BRAZIL
VIETNAM
COLOMBIA
INDONESIA
PERU
ETHIOPIA
MEXICO
INDIA
GUATEMALA
UGANDA
TOP 5 COFFEE FLAVORS
CINNAMON
CHOCOLATE
VANILLA
CARAMEL
COFFEE CONSUMPTION IN THE WORLD


Column Chart

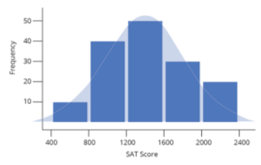

Line Chart

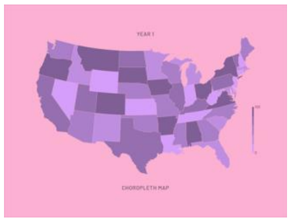

Bar Graph/Chart

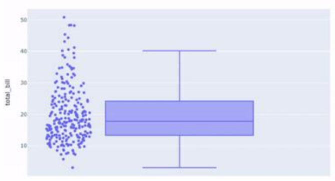

Heat Map



Scatterplot


Histogram


Chloropleth Map


Box / Whisker Plot

Data Visualization Tools in Python

plotly matplotlib

ggplot2 seaborn

Types of Data Visualization

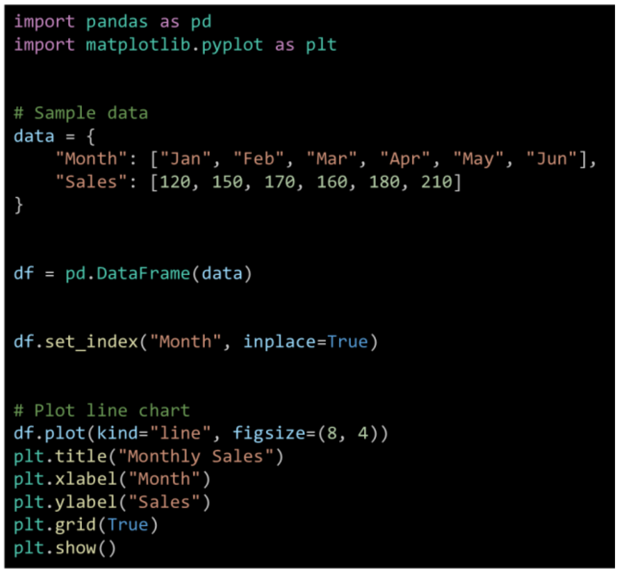
There are many types of data visualization. Some of the most common types used are:

COLUMN CHART

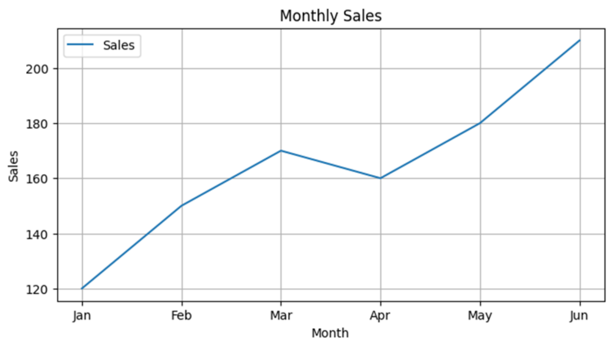
- They are a straightforward, time-tested method of comparing several collections of data. A column chart may be used to track data sets across time.

LINE CHART

- A line graph is used to show trends, development, or changes through time.
- As a result, it functions best when your data collection is continuous as opposed to having many beginnings and ends.

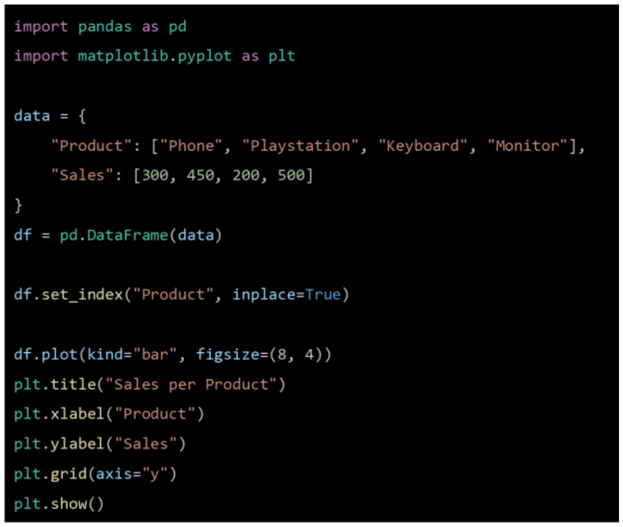


```python
import pandas as pd
import matplotlib.pyplot as plt


# Sample data
data = {
    "Month": ["Jan", "Feb", "Mar", "Apr", "May", "Jun"],
    "Sales": [120, 150, 170, 160, 180, 210]
}


df = pd.DataFrame(data)


df.set_index("Month", inplace=True)


# Plot line chart
df.plot(kind="line", figsize=(8, 4))
plt.title("Monthly Sales")
plt.xlabel("Month")
plt.ylabel("Sales")
plt.grid(True)
plt.show()
```

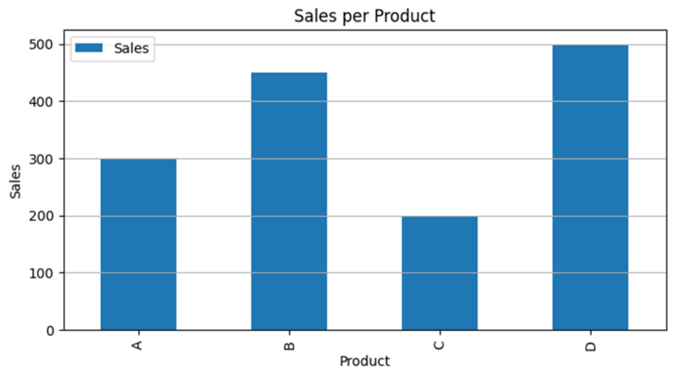


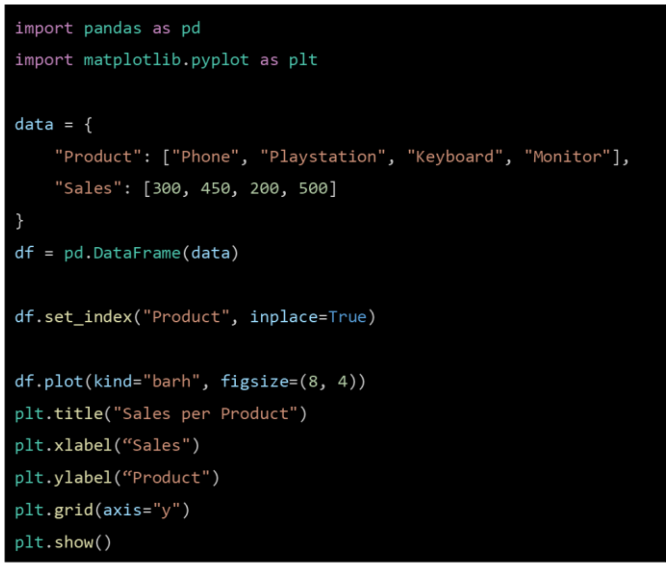
BAR CHART

- To compare data along two axes, use bar charts.
- A visual representation of the categories or subjects being measured is shown on one of the axes, which is numerical.

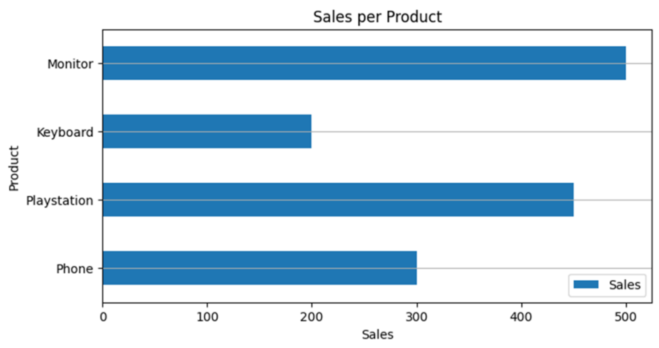


```python
import pandas as pd
import matplotlib.pyplot as plt

data = {
    "Product": ["Phone", "Playstation", "Keyboard", "Monitor"],
    "Sales": [300, 450, 200, 500]
}
df = pd.DataFrame(data)

df.set_index("Product", inplace=True)

df.plot(kind="bar", figsize=(8, 4))
plt.title("Sales per Product")
plt.xlabel("Product")
plt.ylabel("Sales")
plt.grid(axis="y")
plt.show()
```



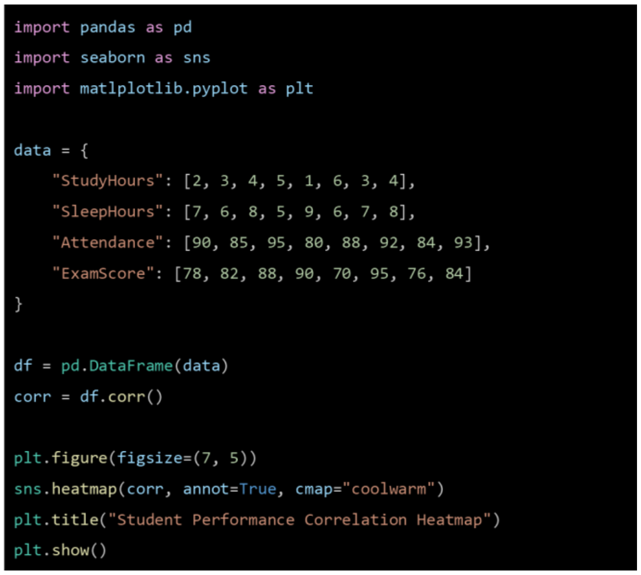


```python
import pandas as pd
import matplotlib.pyplot as plt

data = {
    "Product": ["Phone", "Playstation", "Keyboard", "Monitor"],
    "Sales": [300, 450, 200, 500]
}
df = pd.DataFrame(data)

df.set_index("Product", inplace=True)

df.plot(kind="barh", figsize=(8, 4))
plt.title("Sales per Product")
plt.xlabel(“Sales")
plt.ylabel(“Product")
plt.grid(axis="y")
plt.show()
```

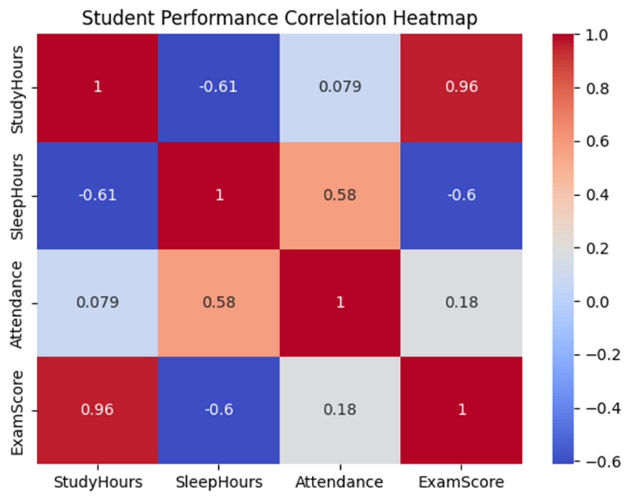


HEAT MAP

- A data visualization method that uses colors to denote values
- Each cell represents the correlation between two subjects.
- Brighter or darker colors indicate stronger or weaker relationships

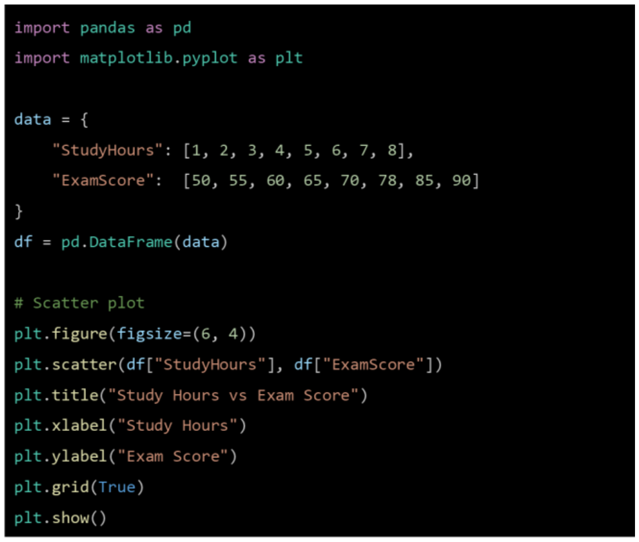


```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

data = {
    "StudyHours": [2, 3, 4, 5, 1, 6, 3, 4],
    "SleepHours": [7, 6, 8, 5, 9, 6, 7, 8],
    "Attendance": [90, 85, 95, 80, 88, 92, 84, 93],
    "ExamScore": [78, 82, 88, 90, 70, 95, 76, 84]
}

df = pd.DataFrame(data)
corr = df.corr()

plt.figure(figsize=(7, 5))
sns.heatmap(corr, annot=True, cmap="coolwarm")
plt.title("Student Performance Correlation Heatmap")
plt.show()
```

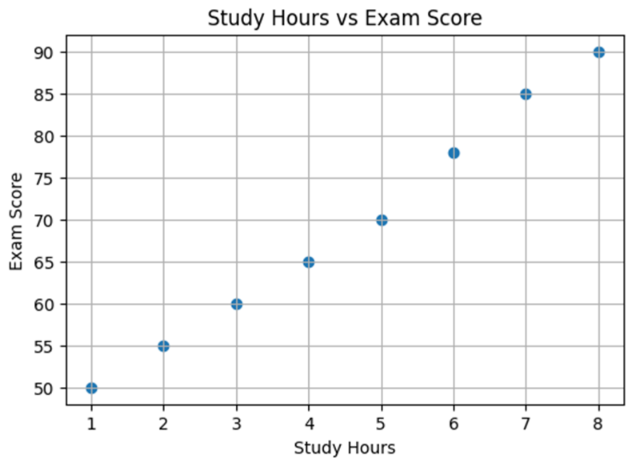


SCATTERPLOT

- The correlation between variables is examined using a scatter plot. At the point where the data's two values overlap, the data are represented on the graph as dots.

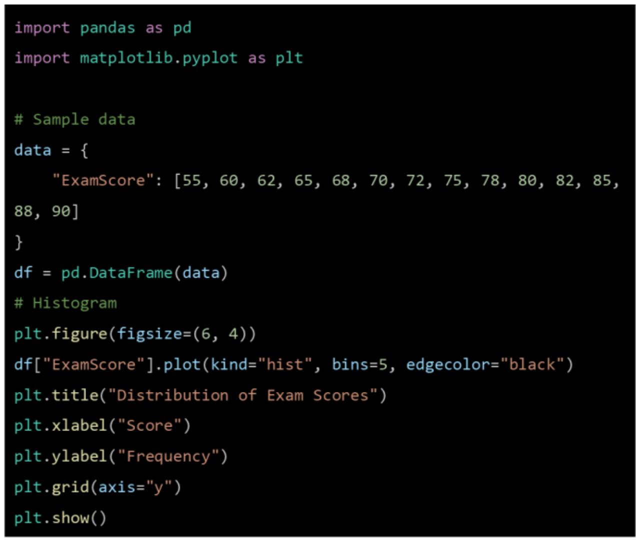


```python
import pandas as pd
import matplotlib.pyplot as plt

data = {
    "StudyHours": [1, 2, 3, 4, 5, 6, 7, 8],
    "ExamScore":  [50, 55, 60, 65, 70, 78, 85, 90]
}
df = pd.DataFrame(data)

# Scatter plot
plt.figure(figsize=(6, 4))
plt.scatter(df["StudyHours"], df["ExamScore"])
plt.title("Study Hours vs Exam Score")
plt.xlabel("Study Hours")
plt.ylabel("Exam Score")
plt.grid(True)
plt.show()
```

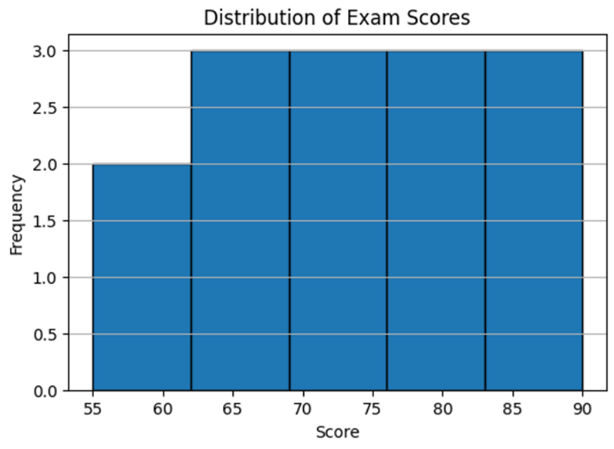


HISTOGRAM

- While a histogram and a bar graph are similar, they use distinct charting systems.
- The ideal sort of data visualization for frequency-based analysis of data ranges is a histogram.

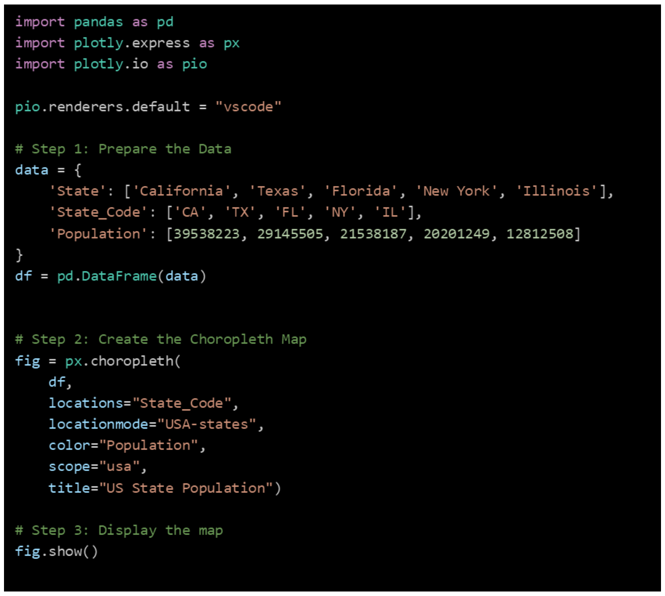


```python
import pandas as pd
import matplotlib.pyplot as plt

# Sample data
data = {
    "ExamScore": [55, 60, 62, 65, 68, 70, 72, 75, 78, 80, 82, 85,
88, 90]
}
df = pd.DataFrame(data)
# Histogram
plt.figure(figsize=(6, 4))
df["ExamScore"].plot(kind="hist", bins=5, edgecolor="black")
plt.title("Distribution of Exam Scores")
plt.xlabel("Score")
plt.ylabel("Frequency")
plt.grid(axis="y")
plt.show()
```

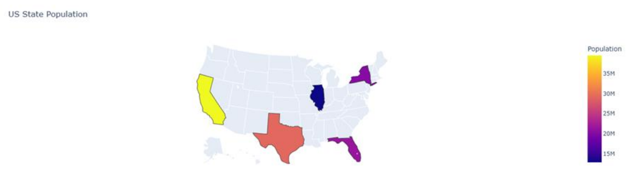


CHLOROPLETH

- The technique of color mapping symbology is used to create choropleth maps, which are themed maps used to display statistical data.
- It shows geographically segmented sections or regions that are colored, shaded, or patterned according to a data variable, known as enumeration units.

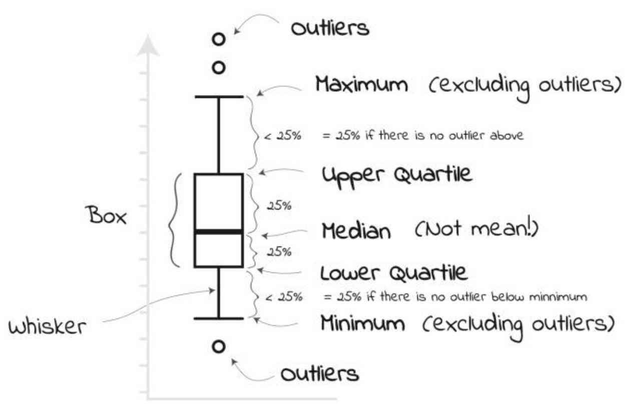


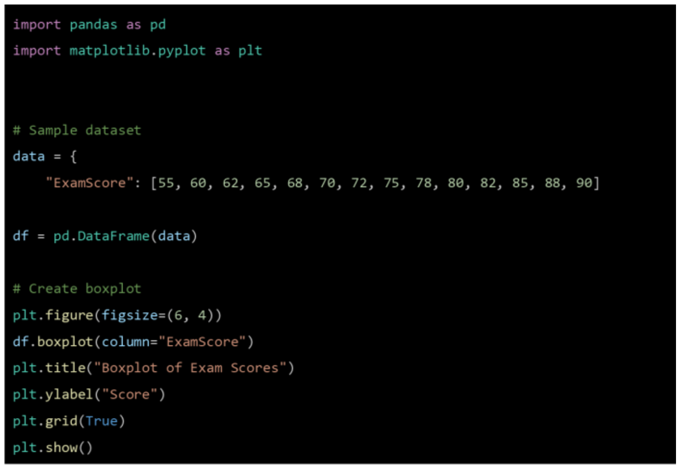
```python
import pandas as pd
import plotly.express as px
import plotly.io as pio

pio.renderers.default = "vscode"

# Step 1: Prepare the Data
data = {
    'State': ['California', 'Texas', 'Florida', 'New York', 'Illinois'],
    'State_Code': ['CA', 'TX', 'FL', 'NY', 'IL'],
    'Population': [39538223, 29145505, 21538187, 20201249, 12812508]
}
df = pd.DataFrame(data)


# Step 2: Create the Choropleth Map
fig = px.choropleth(
    df,
    locations="State_Code",
    locationmode="USA-states",
    color="Population",
    scope="usa",
    title="US State Population")

# Step 3: Display the map
fig.show()
```



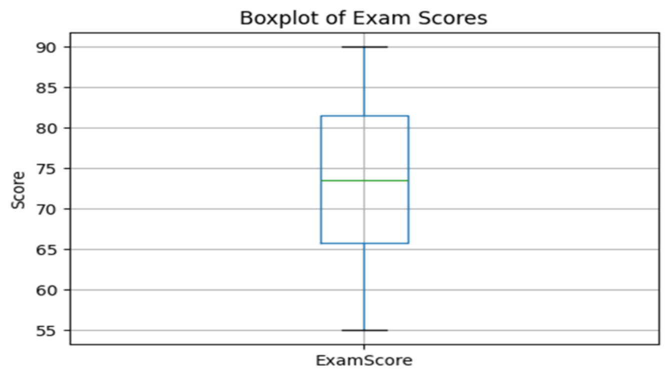
BOXPLOT

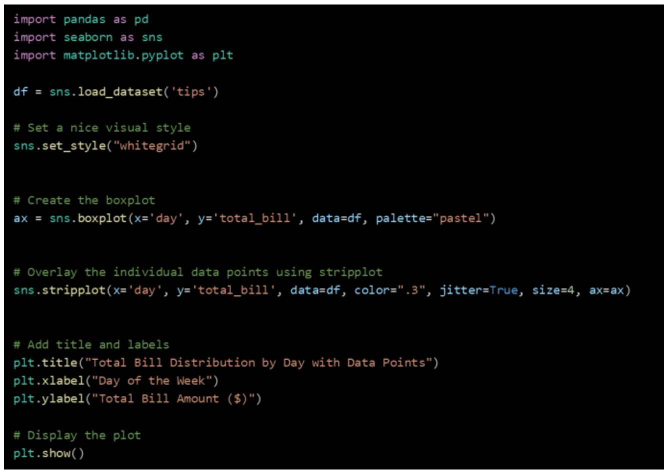
- A Box Plot is also known as a Box and Whisker Plot and it is a


Outliers
Maximum (excluding outliers)
< 25%
= 25% if there is no outlier above
Upper Quartile
25%
Box
Median (Not mean!)
25%
Lower Quartile
< 25%
= 25% if there is no outlier below minimum
Whisker
Minimum (excluding outliers)
Outliers

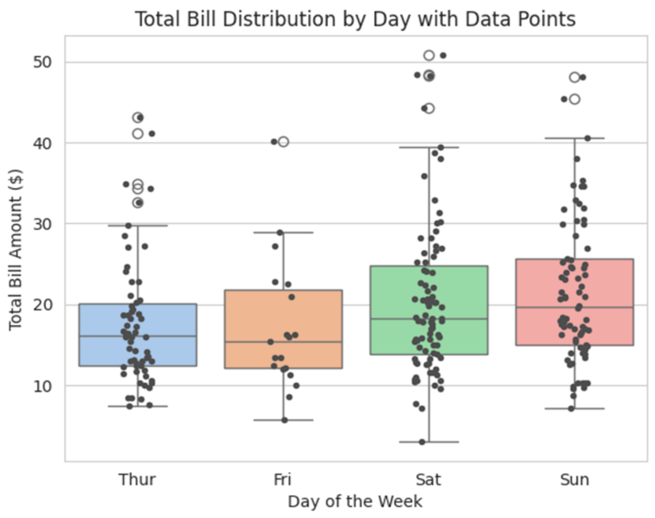


```python
import pandas as pd
import matplotlib.pyplot as plt


# Sample dataset
data = {
    "ExamScore": [55, 60, 62, 65, 68, 70, 72, 75, 78, 80, 82, 85, 88, 90]

df = pd.DataFrame(data)

# Create boxplot
plt.figure(figsize=(6, 4))
df.boxplot(column="ExamScore")
plt.title("Boxplot of Exam Scores")
plt.ylabel("Score")
plt.grid(True)
plt.show()
```





```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = sns.load_dataset('tips')

# Set a nice visual style
sns.set_style("whitegrid")


# Create the boxplot
ax = sns.boxplot(x='day', y='total_bill', data=df, palette="pastel")


# Overlay the individual data points using stripplot
sns.stripplot(x='day', y='total_bill', data=df, color=".3", jitter=True, size=4, ax=ax)


# Add title and labels
plt.title("Total Bill Distribution by Day with Data Points")
plt.xlabel("Day of the Week")
plt.ylabel("Total Bill Amount ($)")

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


Total Bill Distribution by Day with Data Points
Total Bill Amount ($)
Thur
Fri
Sat
Sun
Day of the Week