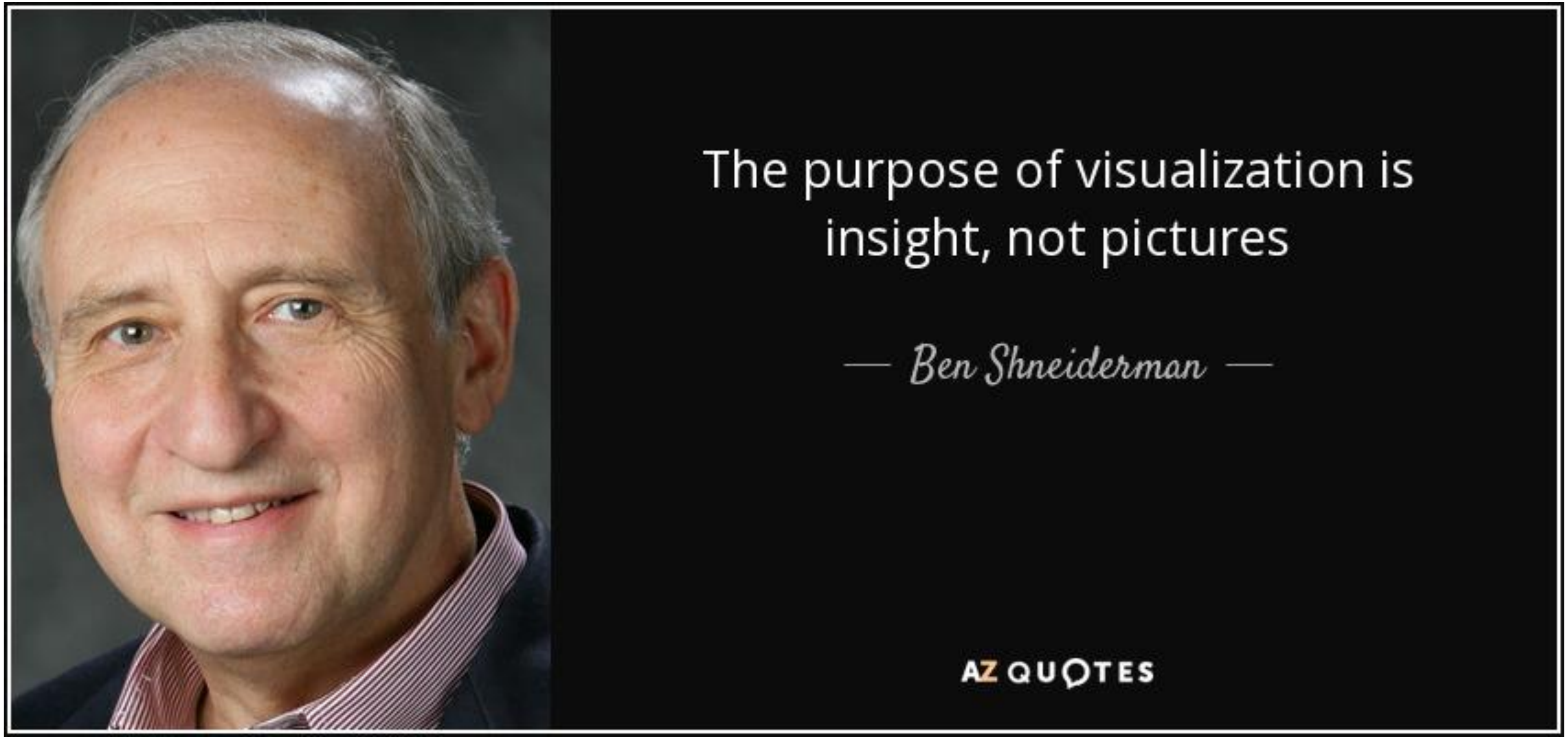
Day 02

Data Science with Python

WE'VE G
THE EN
SALES
SHAVED HEADS
WE FOUND THIS CORRELATION IN THE DATA. EVERYONE TAKE A RAZOR.
TOM FISH BURNE
© marketoonist.com



The purpose of visualization is insight, not pictures

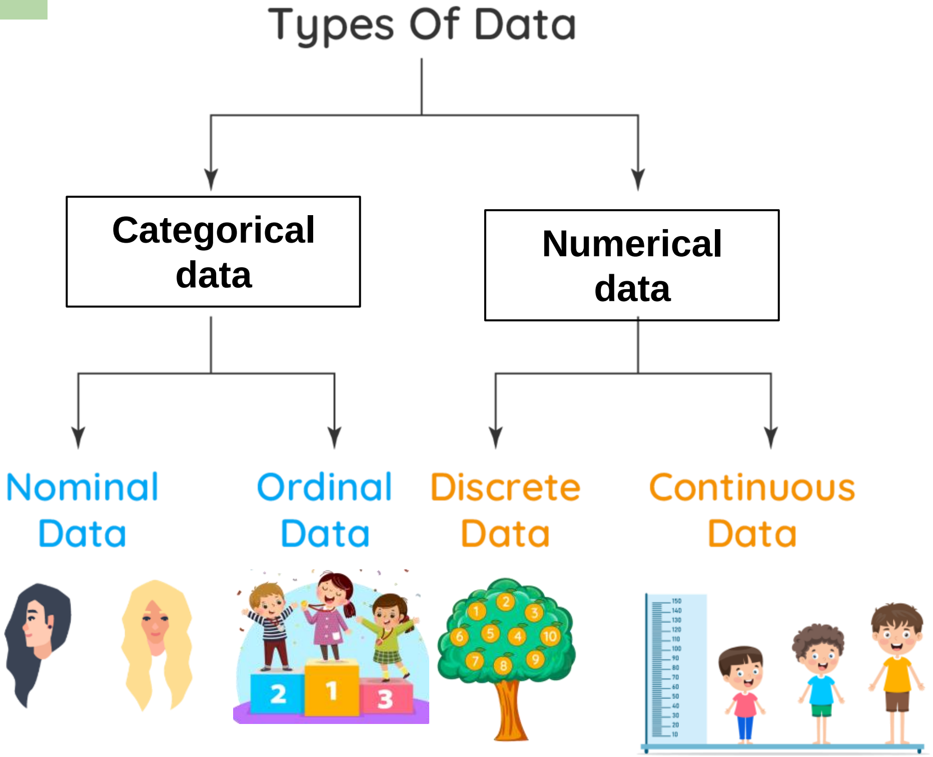
— Ben Shneiderman —

Data

- Collection of information gathered by observations, measurements, research, or analysis.

- It may comprise facts, figures, numbers, names, or even general descriptions of things.

- Can be organized in the form of graphs, charts, or tables for ease in our study.



Types Of Data

Categorical data

Numerical data

Nominal Data

Ordinal Data

Discrete Data

Continuous Data

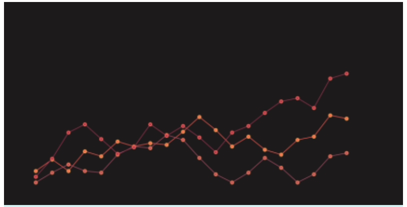
Chart types

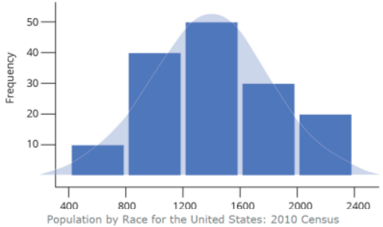
- **Line Chart:** showing *trends or patterns over time or continuous data*.
- Ex: *Stock prices, temperature variations, sales trends.*

- **Bar Chart**: comparing *categorical data or discrete values.*
- Ex: *Comparison of sales by product category, population by country, survey results*.

- **Histogram:** visualizing *distribution of continuous or discrete data*.
- Ex: *Age distribution, exam scores distribution, frequency of occurrence.*

- **Pie Chart**: representing *parts of a whole or proportions.*
- Ex: *Market share of different products, composition of a budget, demographic distribution*.





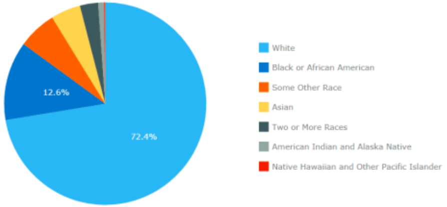
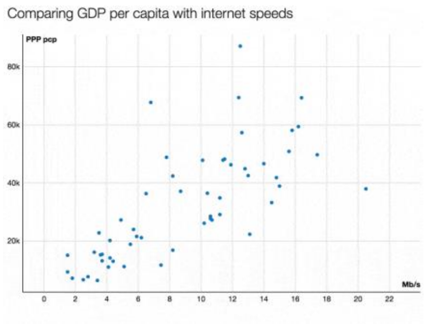
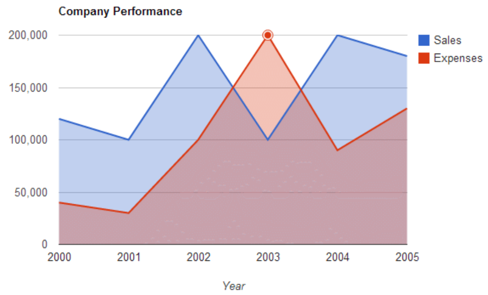


Chart types

- **Scatter Plot:** Suitable for visualizing the relationship between two *continuous variables*.
- Ex: *Correlation between height and weight, relationship between study time and exam scores.*



- **Area Chart:** Suitable for showing the *cumulative sum or proportions* over time.
- Ex: *Total sales over time, population growth over years, stacked area chart for market share.*



- **Box Plot:** Suitable for displaying the *distribution and variability of a dataset*.
- Ex: *Comparison of salaries across departments, distribution of test scores by subject.*

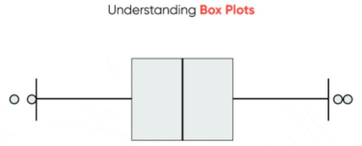
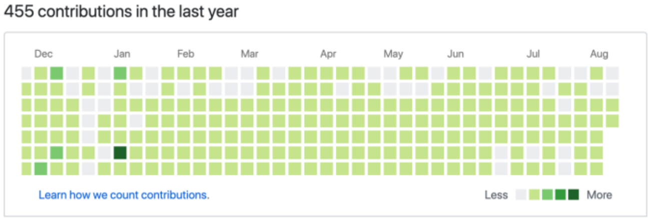


Chart types

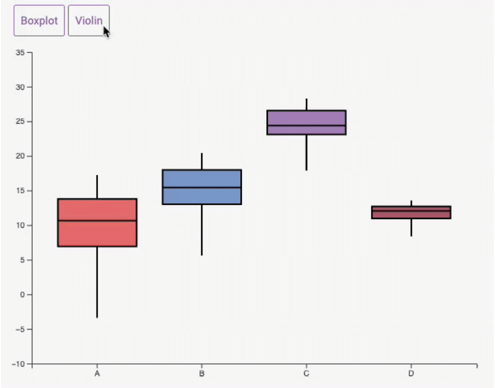
- **Heatmap:** Suitable for displaying relationships between two *categorical variables or visualizing correlation matrices*.
- Ex: *Confusion matrix in machine learning, correlation matrix of variables, visualizing a grid of data*.

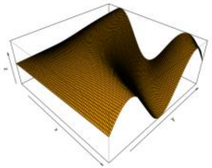
- **Violin Plot:** Suitable for displaying the *distribution and density of a dataset*.
- Ex: *Comparison of income distribution by occupation, distribution of housing prices by location.*

- **3D Plots:** Suitable for visualizing *three-dimensional data or relationships*.
- Ex: *3D surface plots, 3D scatter plots, contour plots.*


455 contributions in the last year

Learn how we count contributions.





Data types

- ***Continuous numerical data (e.g. temperature, height, weight)*** **-**
  - The most common chart types are ***line charts, scatter plots, and histograms***.
  - useful for showing trends over time or a continuous range of values.
  - ***Scatter plot*** > show the ***relationship between two continuous variables***.
  - ***Histogram*** > show the ***distribution of values in a dataset.***

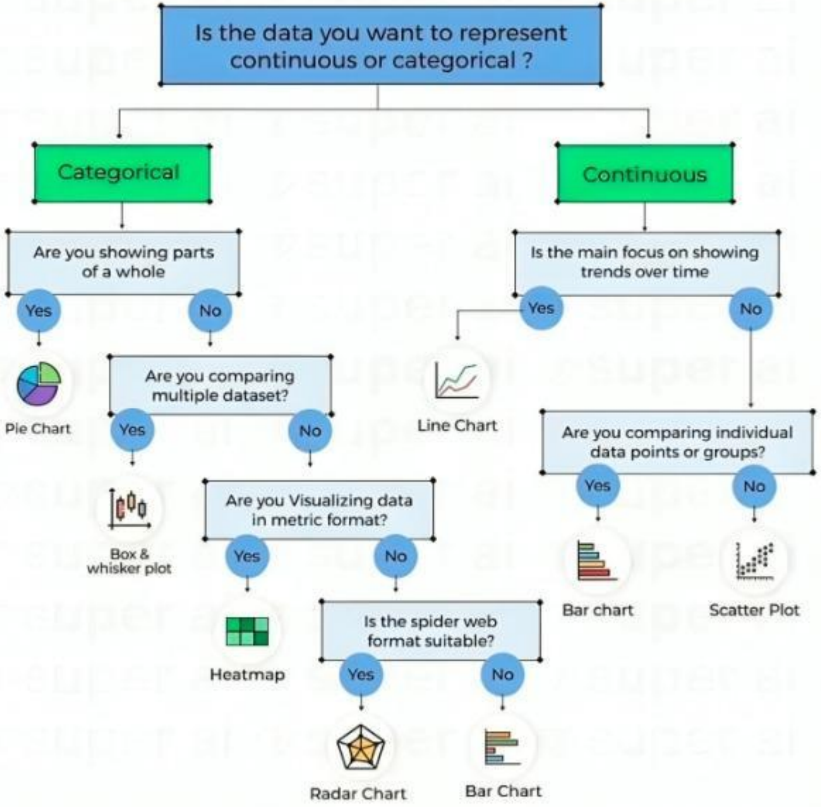
- ***Categorical data*** ***(e.g. colors, age groups, food cuisines, sports, genders, shapes) -***
  - Common chart types are ***bar charts and pie charts.***
  - ***Bar chart*** > compare the ***frequency of different categories***
  - ***Pie chart*** > ***show the proportion of each category in the dataset***.

Data types Cont.

- ***Time-series data (e.g. stock prices, weather patterns)*** **-**
  - ***Line charts*** are typically the most useful for ***visualizing trends over time.***

- ***Geographic data (e.g. city populations, sales by region)*** **-**
  - ***Choropleth*** maps are useful for showing data on a geographic map
  - ***Bubble maps*** can be used to show data points at specific geographic locations.

How to choose a **Right Graph** for **Data Visualization**



Is the data you want to represent continuous or categorical ?

Categorical

Are you showing parts of a whole

Yes

Pie Chart

No

Are you comparing multiple dataset?

Yes

Box & whisker plot

No

Are you Visualizing data in metric format?

Yes

Heatmap

No

Is the spider web format suitable?

Yes

Radar Chart

No

Bar Chart

Continuous

Is the main focus on showing trends over time

Yes

Line Chart

No

Are you comparing individual data points or groups?

Yes

Bar chart

No

Scatter Plot

visualizations approaches

- ***Procedural visualization libraries*** - require you to explicitly specify the steps needed to create the visualization.
    - Matplotlib, ggplot2 (for R), and D3.js (for JavaScript).
- ***Declarative visualization libraries*** - allow you to describe the visualization in terms of its intended output, without specifying the steps needed to create it.
    - Plotly

Grammar of Graphics

- **Data**: Raw information that you want to visualize
- **Aesthetics**: How the data variables are mapped to visual properties such as position, size, color, and shape.
- **Geometries**: Visual marks or shapes used to represent the data. - points, lines, bars, areas, and more.
- **Scales**: How the values of data variables are mapped to the visual range.
- **Facets**: Subsets of the data and display them in separate panels or subplots.

Data Visualization with Matplotlib Library

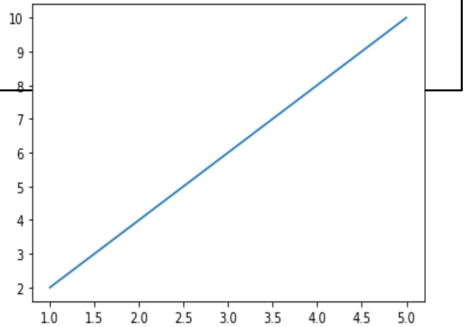
Introduction to Matplotlib

- Popular Python library used for creating static, animated, and interactive visualizations.
- It is widely used in data analysis, scientific research, and machine learning, among other fields.
- Matplotlib provides a variety of functions for creating ***line plots, scatter plots, bar charts, histograms, and more.***
- One of the most widely used data visualization libraries in Python.

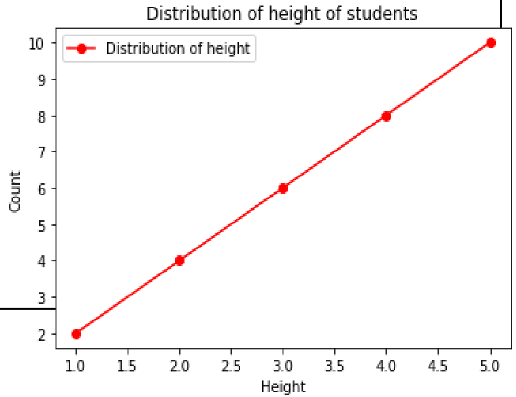
Line plots

- A line plot is a plot that displays data as a series of points connected by straight lines.

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

plt.plot(x, y)

plt.show()
```



```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

plt.plot(x, y,'pattern',c='',label=' ')

plt.xlabel(' ')
plt.ylabel(' ')
plt.title(' ')

plt.legend()
plt.show()
```



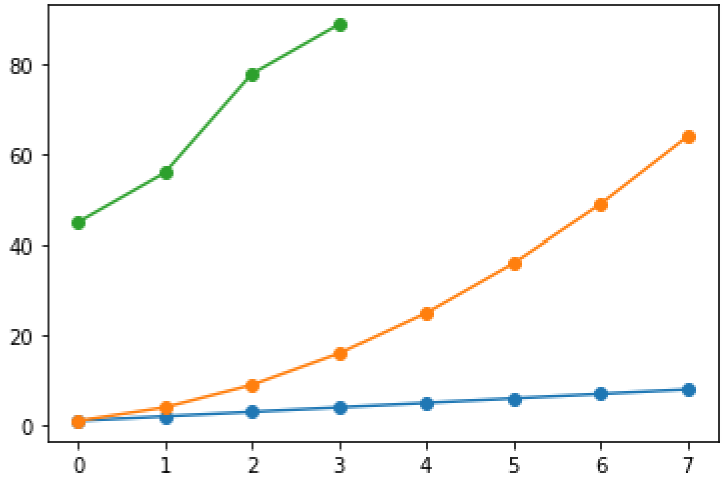
- 3rd argument - ‘pattern’

Plotting with matplotlib

- fg=plt.figure(figsize=(10,10))
- plt.gcf()
- gh=plt.gca()
- gh.axis([starting_valueX,ending_valueX,starting_valueY,ending_valueY])
- gh.get_children()

Line plots

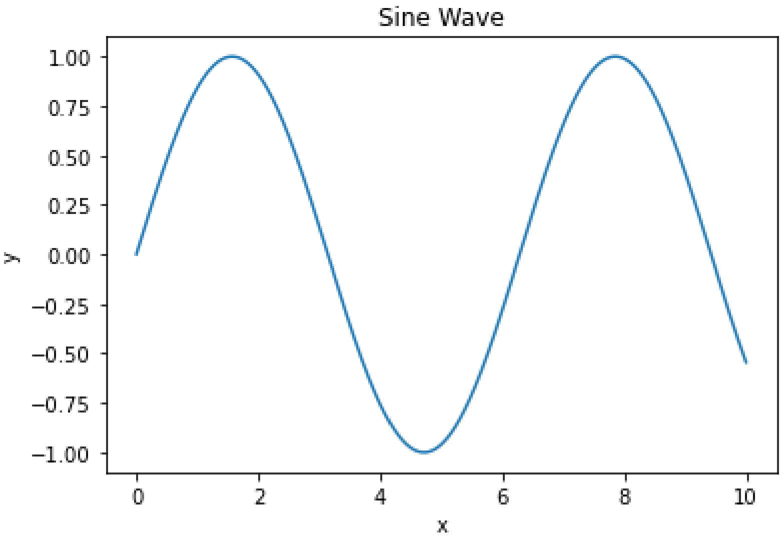
```
linear_data=np.array([1,2,3,4,5,6,7,8])

exponential_data=linear_data**2

x=[45,56,78,89]
```



```
import matplotlib.pyplot as plt
import numpy as np

linear_data=np.array([1,2,3,4,5,6,7,8])

exponential_data=linear_data**2

plt.figure()

plt.plot(linear_data,'-o',exponential_data,'-o')

plt.plot([45,56,78,89],'-o')
```

Plotting Mathematical Functions

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 10, 1000)
y = np.sin(x)

# Plot the sine wave
plt.plot(x, y)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Sine Wave')
plt.show()
```



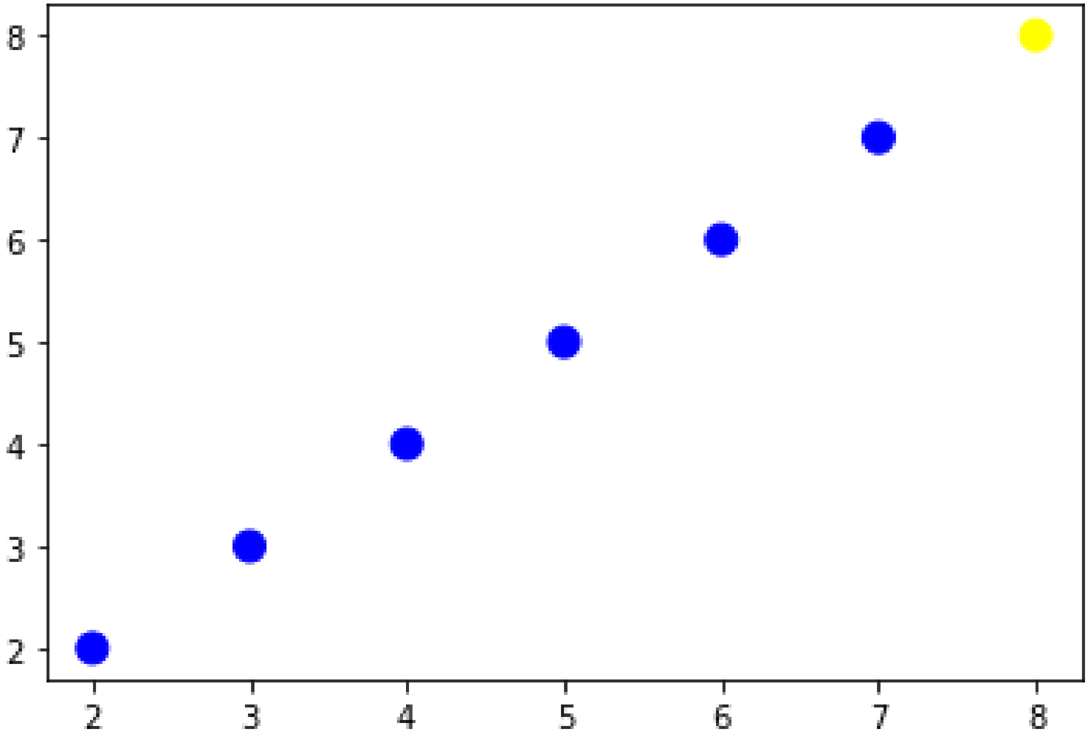
Scatterplots

Plot that displays data as a collection of points.

- plt.scatter(x, y, s=None, c=None)
- .xlabel()
- .ylabel()
- .title()

```
import matplotlib.pyplot as plt

x = np.random.randint(low=0, high=10000, size = 90)
y = np.random.randint(low=0, high=10000, size = 90)

plt.scatter(x, y)

plt.show()
```

Scatterplots

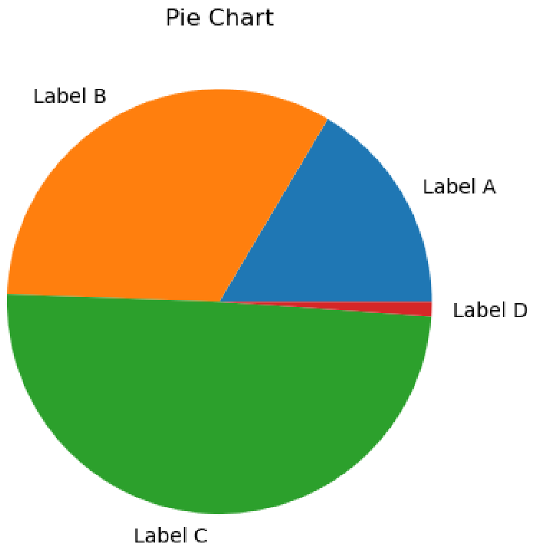


```
import numpy as np

y=np.array([2,3,4,5,6,7,8])
x=y

colors=['blue']*(len(y)-1)
colors.append('yellow')

plt.figure()

plt.scatter(x,y,s=100,c=colors)
```

Pie charts

- Pie charts are circular-shaped charts that record discrete data whereby pie represents the whole and the slices represent the parts of the whole.

```
import matplotlib.pyplot as plt

values = [15, 30, 45, 1]

labels = ['Label A', 'Label B', 'Label C', 'Label D']

# Create a pie chart
plt.pie(values, labels=labels)

plt.title('Pie Chart')

plt.show()
```

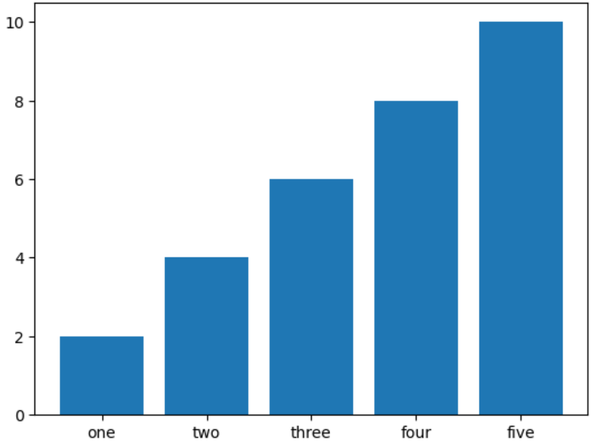


Bar charts

Plot that displays data as a series of bars, with the height of each bar representing the value of the data.

- .bar(x,y,width)

```
import matplotlib.pyplot as plt

x = ['one', 'two', 'three', 'four', 'five']
y = [2, 4, 6, 8, 10]

plt.bar(x, y)

plt.show()
```

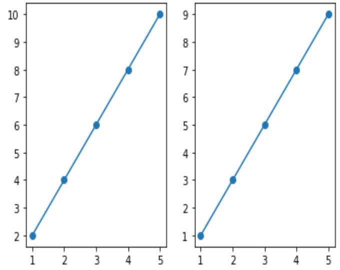


- stacked bar graph - .bar(x, ,width=,bottom= ,color=’ ’)
- horizontal stacked bar graph - .barh(x, ,height=,left= ,color=’ ’)

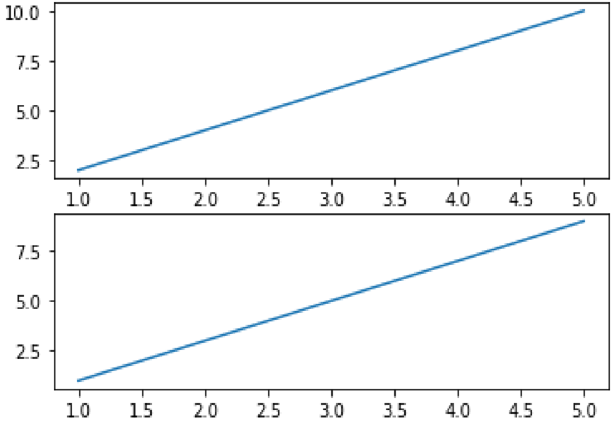
Subplots

Way to display multiple plots in a single figure. This is useful when you want to compare different data sets or display related data sets together.

- .subplot(rows,cols,plot number)
- fig, ((, ,),(, ,),(, ,))=.subplots()

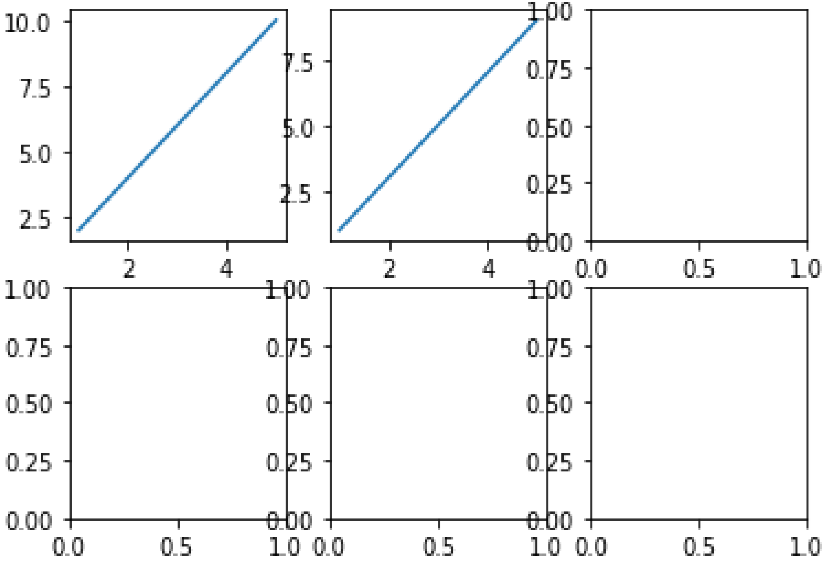


```
plt.subplot(1,2,1)

x = [1, 2, 3, 4, 5]
y1 = [2, 4, 6, 8, 10]
y2 = [1, 3, 5, 7, 9]

plt.plot(x,y1,'-o')
plt.subplot(1,2,2)
plt.plot(x,y2,'-o')
```



```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y1 = [2, 4, 6, 8, 10]
y2 = [1, 3, 5, 7, 9]

fig, axs = plt.subplots(2, 1)

axs[0].plot(x, y1)
axs[1].plot(x, y2)

plt.show()
```

Subplots



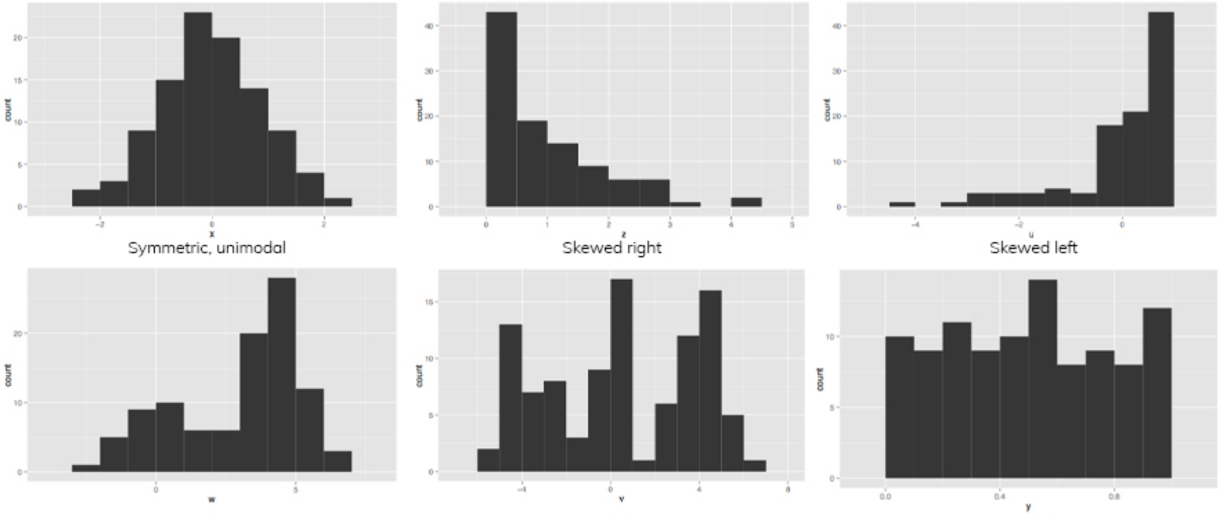
```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y1 = [2, 4, 6, 8, 10]
y2 = [1, 3, 5, 7, 9]

fig, axs = plt.subplots(2, 3)

axs[0,0].plot(x, y1)
axs[0,1].plot(x, y2)

plt.show()
```

Subplots

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

df = pd.read_csv('winequality-red.csv')

# Define the column pairs for the scatter plots
column_pairs = [('alcohol', 'pH'), ('fixed acidity', 'citric acid'), ('residual sugar', 'chlorides'), ('density', 'sulphates')]

# Create subplots
fig, axs = plt.subplots(2, 2, figsize=(10, 8))

# Iterate over the column pairs and plot each scatter plot
for i, (x_col, y_col) in enumerate(column_pairs):
    ax = axs[i // 2, i % 2]
    ax.scatter(df[x_col], df[y_col], color='steelblue', alpha=0.7)
    ax.set_xlabel(x_col)
    ax.set_ylabel(y_col)

# Adjust spacing between subplots
plt.tight_layout()

plt.show()
```

Histogram

- Common way to visualize the distribution of a dataset.
- They provide a graphical representation of the ***frequency of values in a dataset***, which can help identify patterns and anomalies in the data.



Symmetric, unimodal

Skewed right

Skewed left

Bimodal

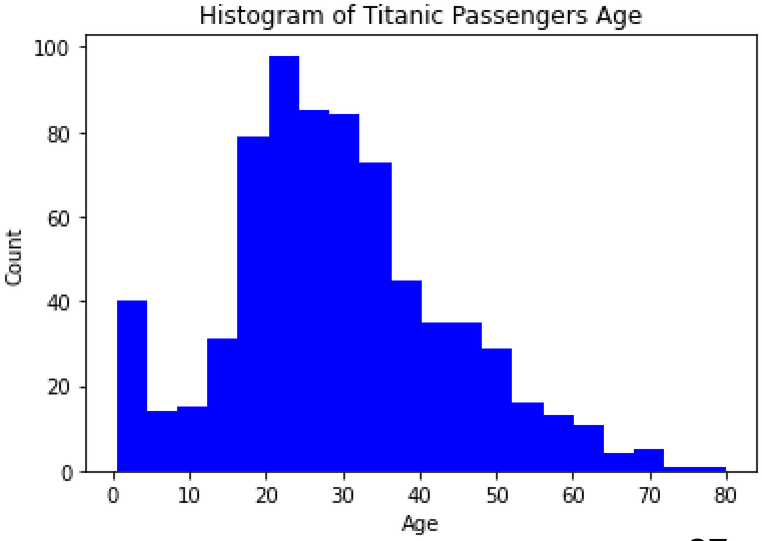
Multimodal

Symmetric

Histogram

- Common way to visualize the distribution of a dataset.
- They provide a graphical representation of the ***frequency of values in a dataset***, which can help identify patterns and anomalies in the data.

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

titanic_data = pd.read_csv(‘titanic.csv')

plt.hist(titanic_data['Age'], bins=20, color='blue')

plt.xlabel('Age')
plt.ylabel('Count')
plt.title('Histogram of Titanic Passengers Age')

plt.show()
```

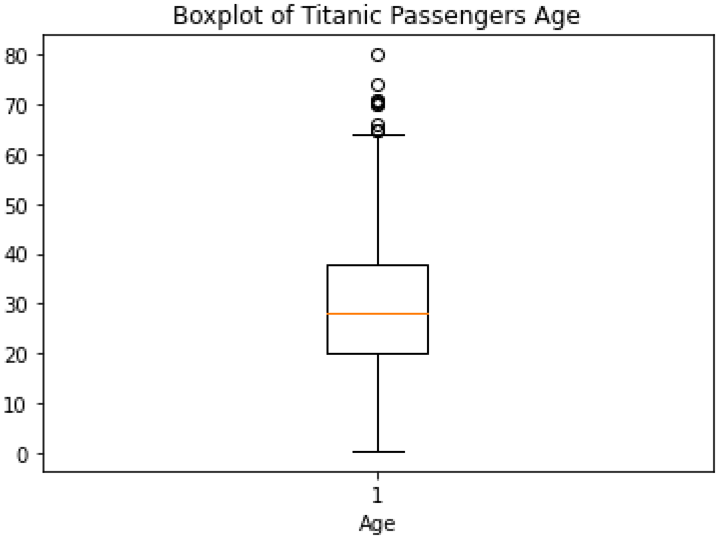


Box Plots

way of displaying the distribution of a dataset through its quartiles.

- Aggregate statistics - 5 number summary - min, max, median, 1st and 3rd quartiles
- plt.boxplot(, whis=[0,100])

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

titanic_data = pd.read_csv('titanic.csv')

plt.boxplot(titanic_data['Age'].dropna())

plt.xlabel('Age')
plt.title('Boxplot of Titanic Passengers Age')

plt.show()
```

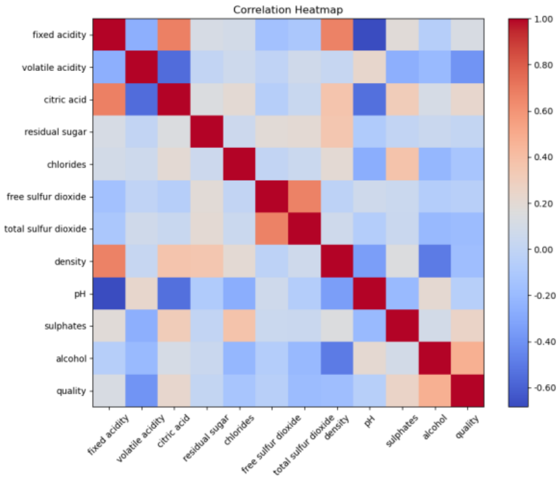


Heatmaps

- 3 dimensions

- way of displaying data in a color-coded matrix, where different colors represent different values.

```
import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_csv(‘winequality-red.csv')


# Calculate the correlation matrix
corr_matrix = df.corr()

# Create a heatmap
plt.figure(figsize=(10, 8))
plt.imshow(corr_matrix, cmap='coolwarm', aspect='auto')
plt.colorbar(format='%.2f')
plt.xticks(range(len(corr_matrix.columns)), corr_matrix.columns,
rotation=45)
plt.yticks(range(len(corr_matrix.columns)), corr_matrix.columns)
plt.title('Correlation Heatmap')

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



Data Visualization with Pandas Library

What is Pandas?

- A popular Python library used for data manipulation and analysis
- “Pandas” comes from both "Panel Data" and "Python Data Analysis"
- Built on top of other popular Python libraries, such as NumPy and matplotlib
- Can be used for a wide range of data-related tasks, such as data cleaning, transformation, exploration, analysis, and visualization.

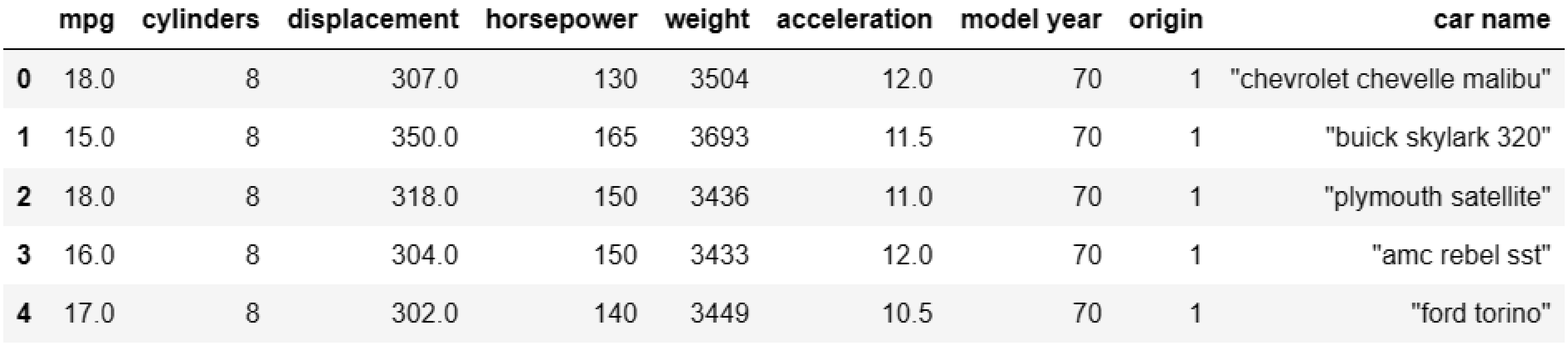
What is Pandas?

Types of visualizations

- Line plots
- Scatter plots
- Bar plots
- Histograms
- Box plots
- Area plots
- Pie charts
- Kernel density estimation plots
- Hexbin plots
- 3D scatter plots

Plotting with pandas

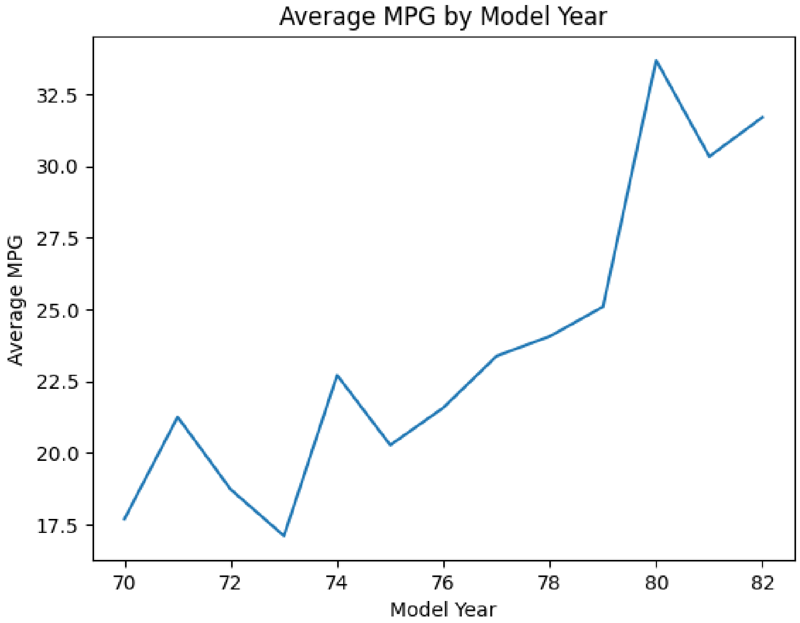
```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('mpg.csv')
df.head()
```



	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130	3504	12.0	70	1	"chevrolet chevelle malibu"
1	15.0	8	350.0	165	3693	11.5	70	1	"buick skylark 320"
2	18.0	8	318.0	150	3436	11.0	70	1	"plymouth satellite"
3	16.0	8	304.0	150	3433	12.0	70	1	"amc rebel sst"
4	17.0	8	302.0	140	3449	10.5	70	1	"ford torino"

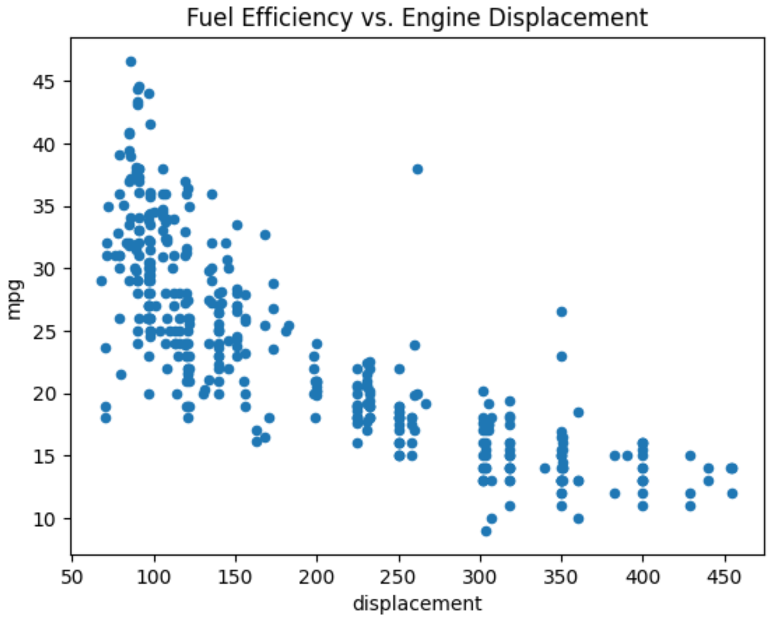
Line Plot

```
plt.figure()
df.groupby('model year')['mpg'].mean().plot(kind='line')
plt.xlabel('Model Year')
plt.ylabel('Average MPG')
plt.title('Average MPG by Model Year')
plt.show()
```



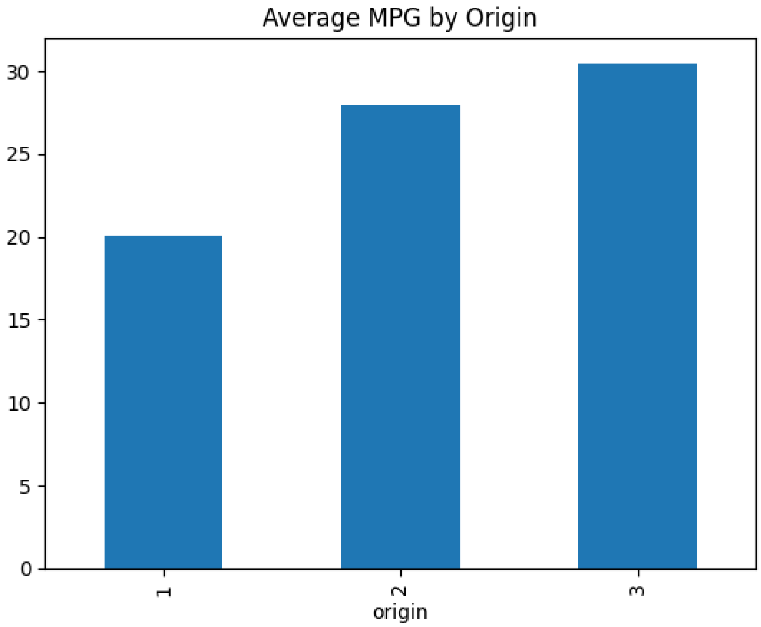
Scatter Plot

```
plt.figure()
df.plot(kind='scatter', x='displacement', y='mpg')
plt.title('Fuel Efficiency vs. Engine Displacement')
plt.savefig('scatter_plot.png')
plt.show()
```



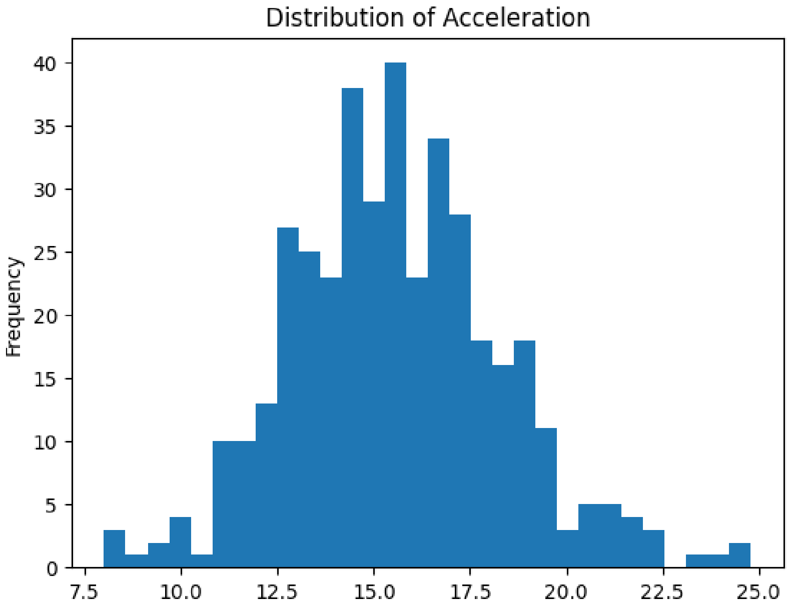
Bar Plot

```
plt.figure()
df.groupby('origin')['mpg'].mean().plot(kind='bar')
plt.title('Average MPG by Origin')
plt.show()
```



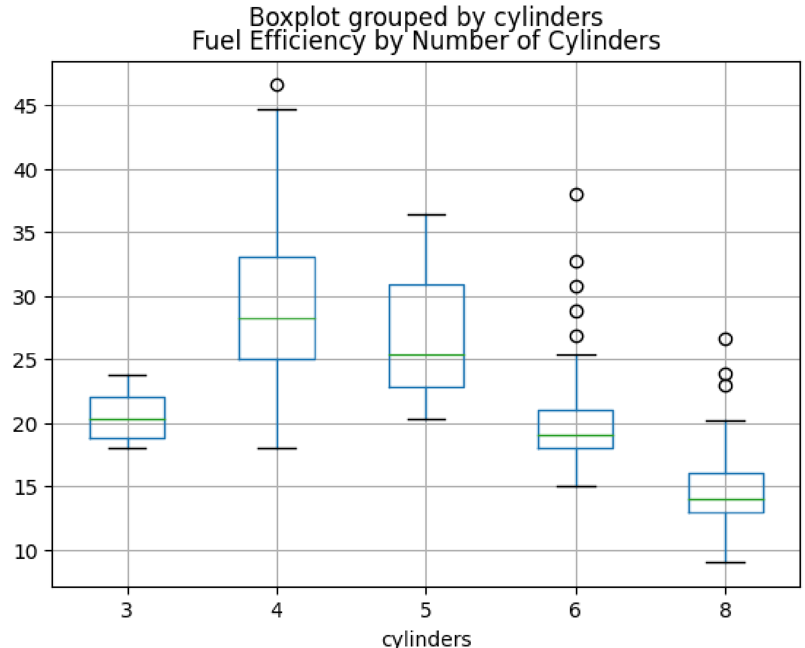
Histogram

```
plt.figure()
df['acceleration'].plot(kind='hist', bins=30)
plt.title('Distribution of Acceleration')
plt.show()
```



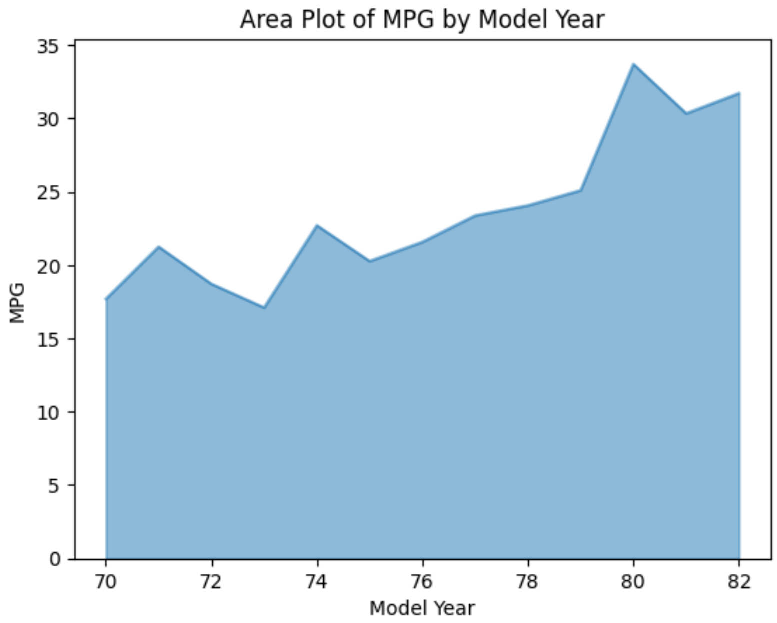
Box Plot

```
plt.figure()

df.boxplot(column='mpg', by='cylinders')

plt.title('Fuel Efficiency by Number of Cylinders')

plt.show()
```



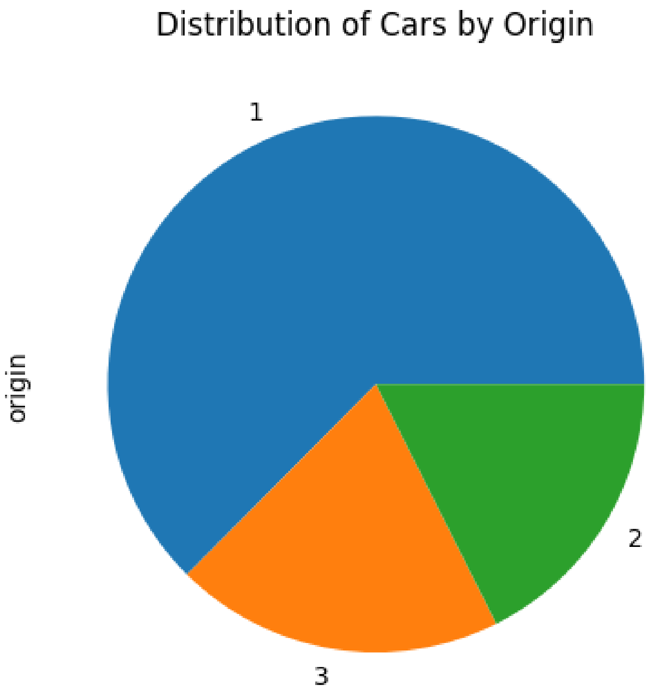
Area Plot

```
plt.figure()
df.groupby('model year')['mpg'].mean().plot(kind='area', alpha=0.5)
plt.title('Area Plot of MPG by Model Year')
plt.xlabel('Model Year')
plt.ylabel('MPG')
plt.show()
```



Pie Chart

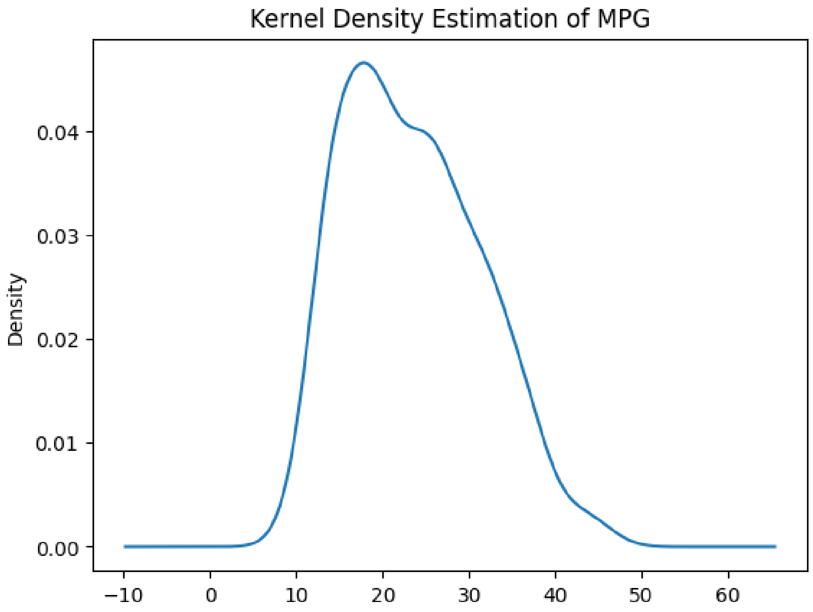
```
plt.figure()
df['origin'].value_counts().plot(kind='pie')
plt.title('Distribution of Cars by Origin')
plt.show()
```



Distribution of Cars by Origin

Kernel density estimation plots(KDE)

```
plt.figure()
df['mpg'].plot(kind='kde')
plt.title('Kernel Density Estimation of MPG')
plt.show()
```



Data Visualization with Seaborn Library

What is Seaborn

- A Python data visualization library based on matplotlib
- Plotting functions operate on data frames and arrays
- Graphs can be customized easily

Installation

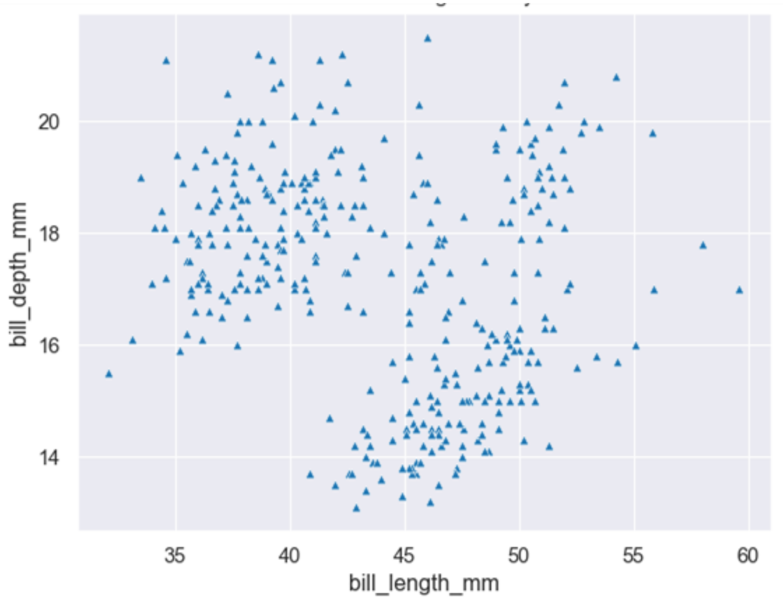
```
pip install seaborn

conda install seaborn
```

Plotting with Seaborn

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

#retreive the available datasets
sns.get_dataset_names()

sns.set_style("darkgrid")

penguins = sns.load_dataset('penguins')
penguins.head()
```

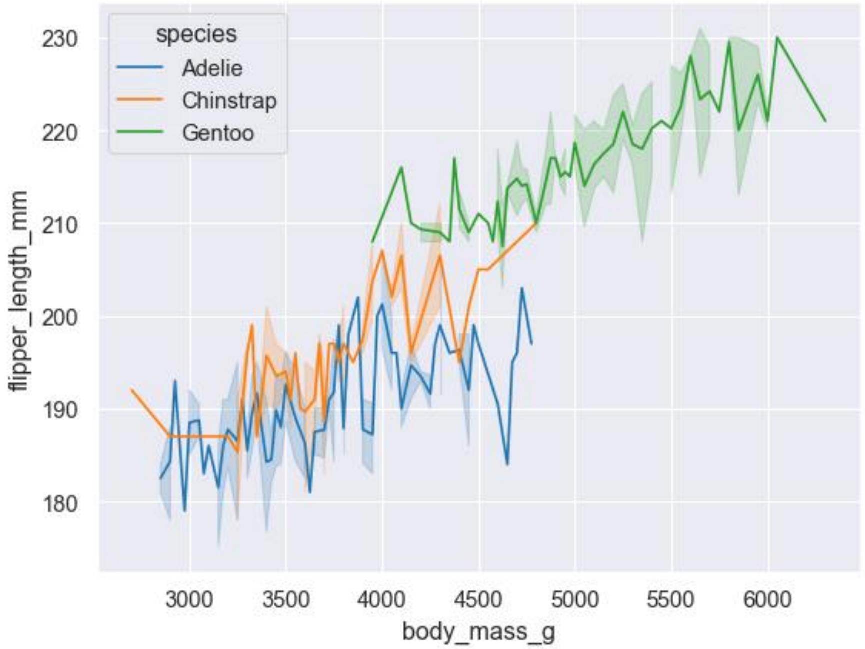
Scatter Plot

```
penguins = sns.load_dataset("penguins")
sns.scatterplot(x="bill_length_mm", y="bill_depth_mm", data=penguins, marker="^")
plt.show()
```



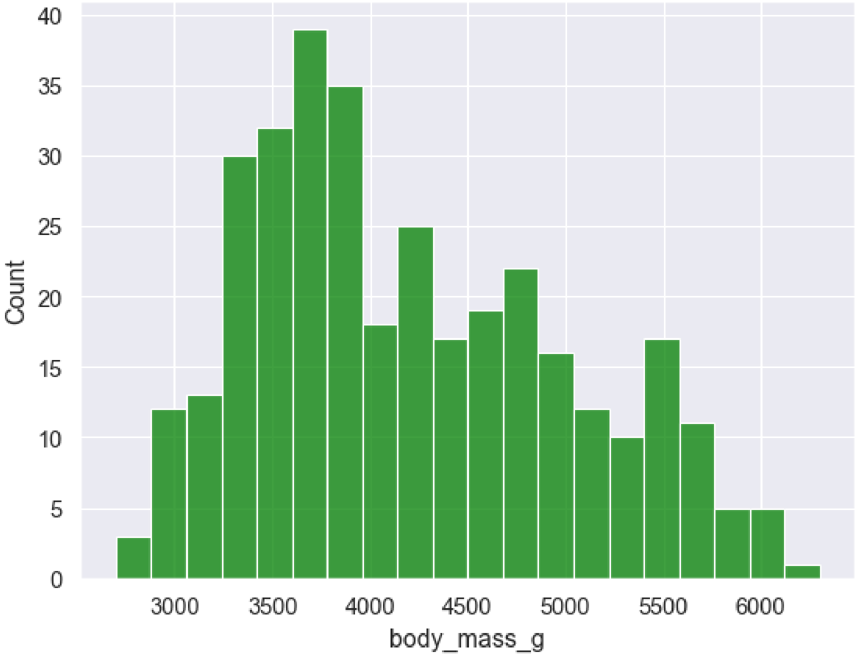
Line Plot

```
sns.lineplot(x="body_mass_g", y="flipper_length_mm", hue="species", data=penguins)
plt.show()
```



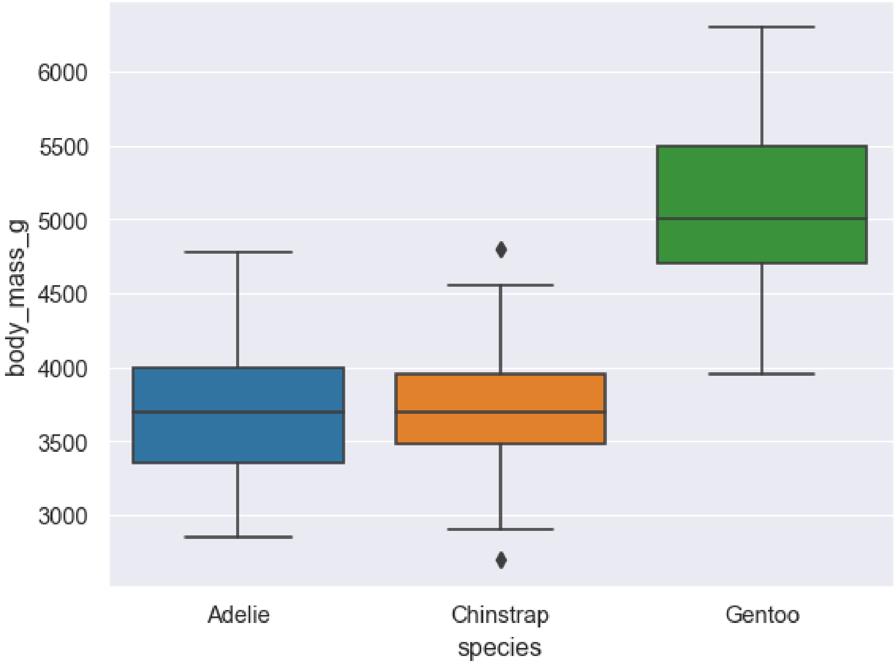
Histogram

```
sns.histplot(x="body_mass_g", data=penguins, color='green', bins = 20)
plt.show()
```



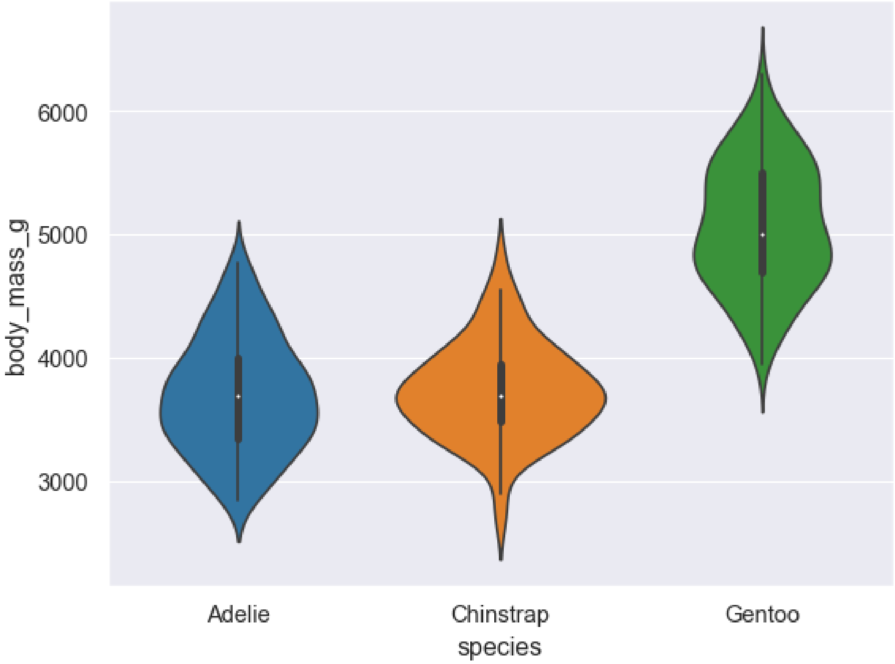
Box Plot

```
sns.boxplot(x="species", y="body_mass_g", data=penguins)

plt.show()
```



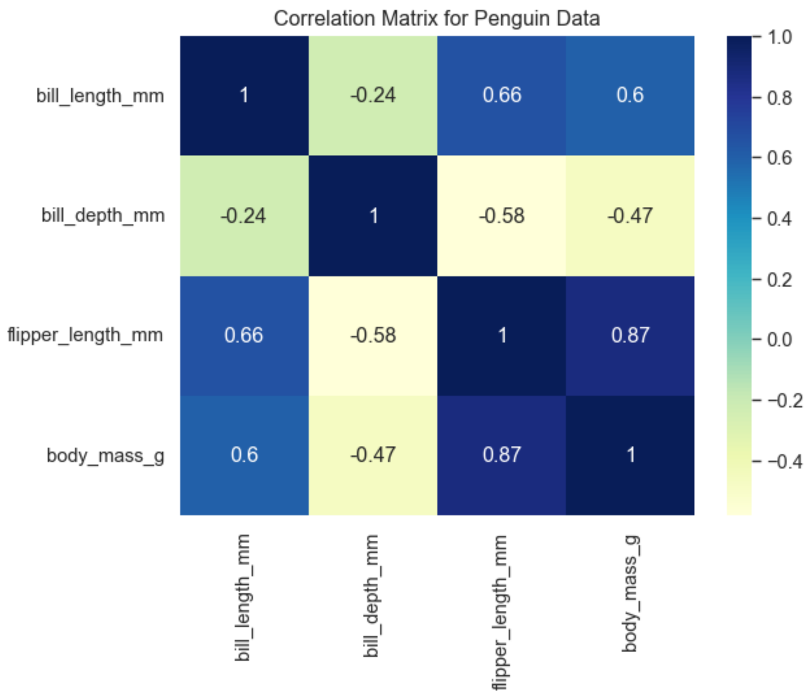
Violin Plot

```
sns.violinplot(x="species", y="body_mass_g", data=penguins)
plt.show()
```



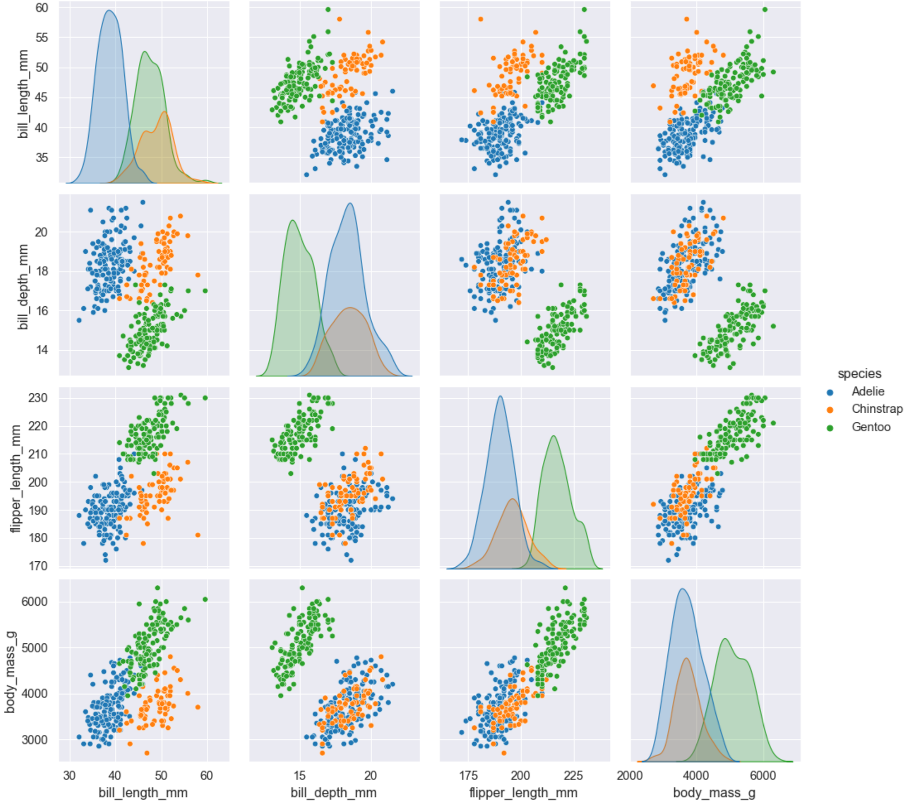
Heatmap

```
corr_matrix = penguins.corr()
sns.heatmap(corr_matrix, annot=True, cmap="YlGnBu")
plt.show()
```



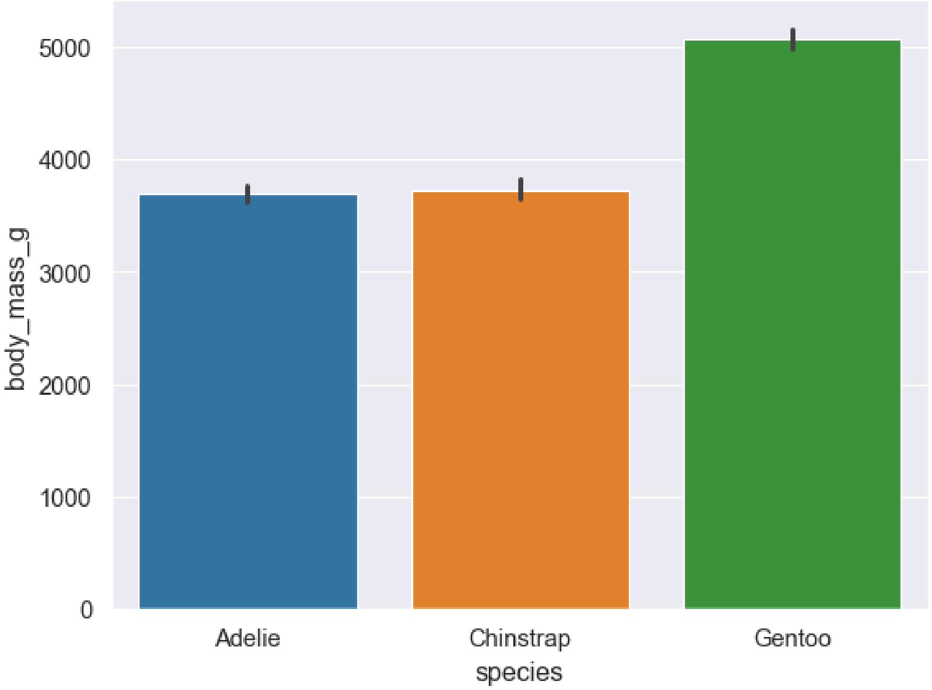
Pair Plot

```
sns.pairplot(data=penguins, hue="species")

plt.show()
```



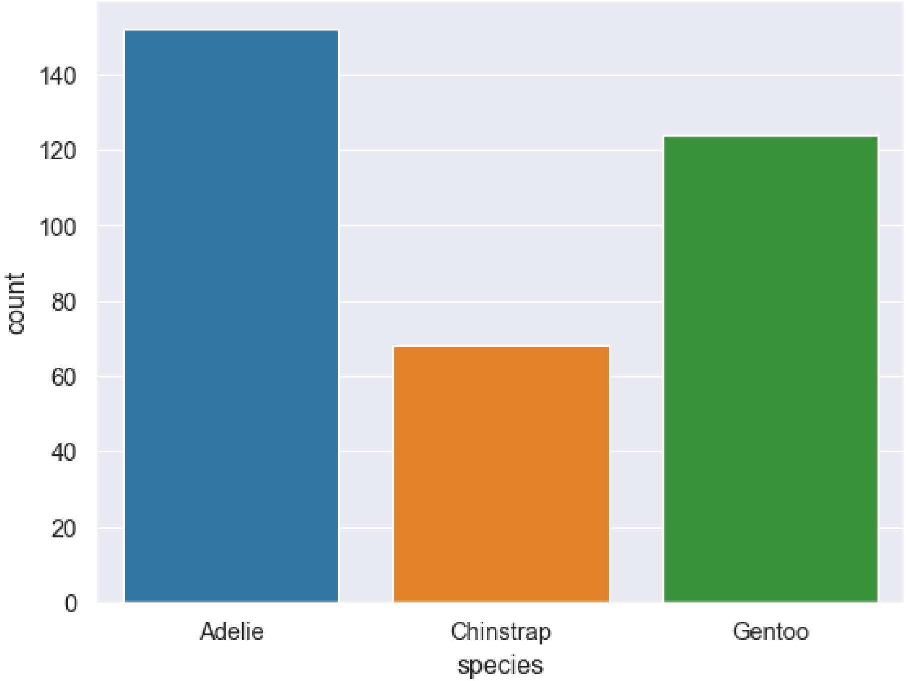
Bar Plot

```
sns.barplot(x="species", y="body_mass_g", data=penguins)
plt.show()
```



Count Plot

```
sns.countplot(x="species", data=penguins)
plt.show()
```



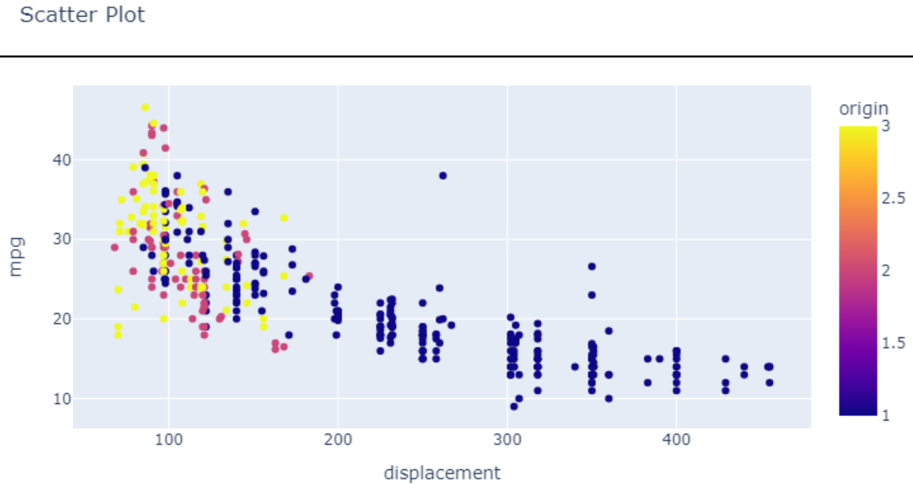
Data Visualization with Plotly Library

What is plotly?

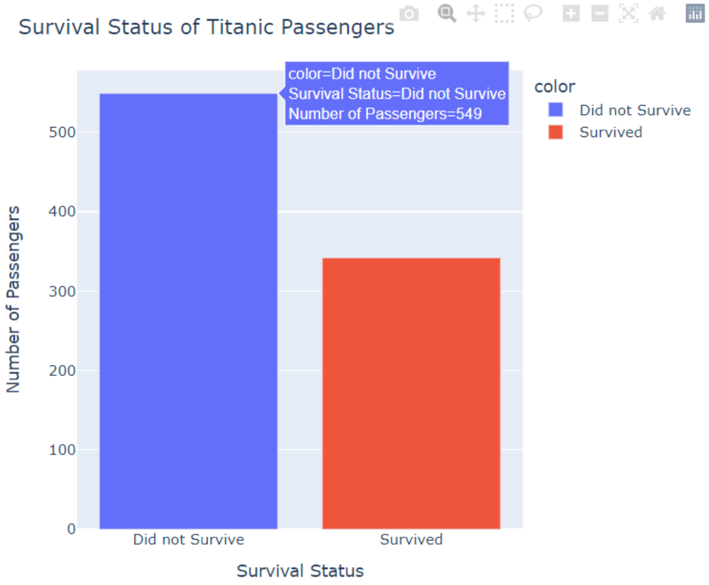
- Plotly is a Python library for creating interactive data visualizations.
- Has become a popular choice for data visualization in industry and academia.
- A wide range of chart types, including ***scatter plots, bar charts, line charts, and more.***
- Interactive features like ***hover effects, zooming, and panning.***
- Support for creating dashboards and web-based applications with Dash.
- Easy to use and customize.
- Installation - `pip install plotly`

Scatter plot

```
import pandas as pd
import plotly.express as px
df = pd.read_csv('mpg.csv')
fig = px.scatter(df, x='displacement', y='mpg', color='origin', title='Scatter
Plot')
fig.show()
```

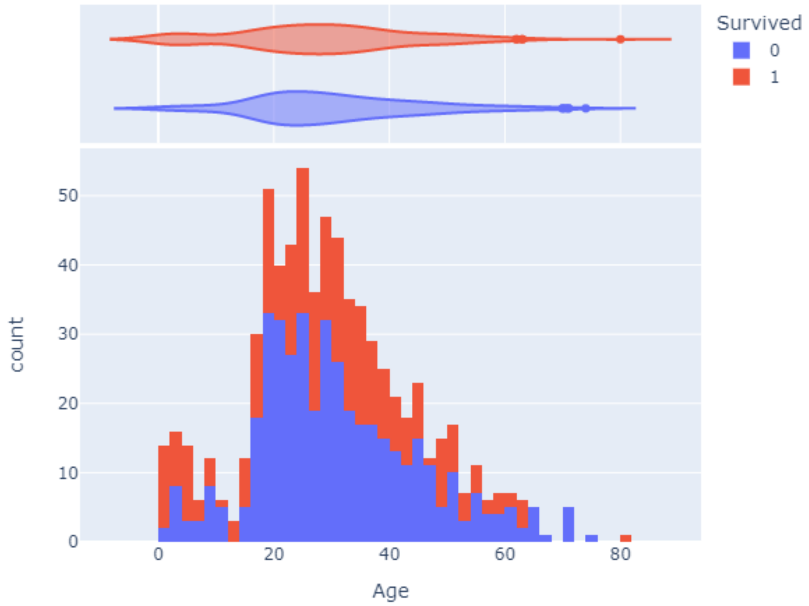


Bar chart



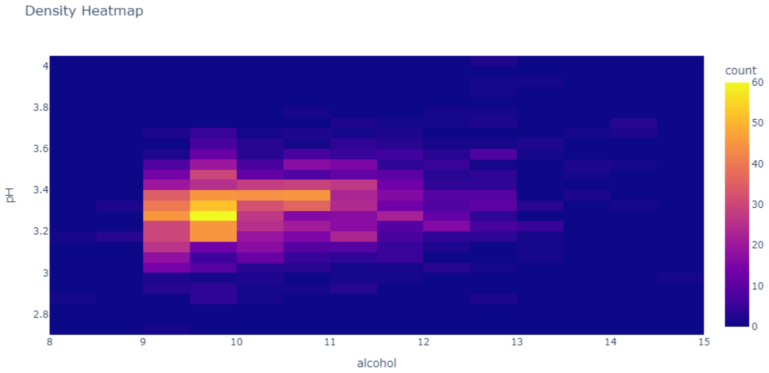
```
import plotly.express as px
import pandas as pd

titanic = pd.read_csv('titanic.csv')
survival_counts = titanic['Survived'].value_counts()

fig = px.bar(
    x=['Did not Survive', 'Survived'],
    y=survival_counts.values,
    labels={'x': 'Survival Status', 'y': 'Number of
Passengers'},
    color=['Did not Survive', 'Survived']
)

fig.update_layout(title='Survival Status of Titanic
Passengers')

fig.show()
```

Histogram



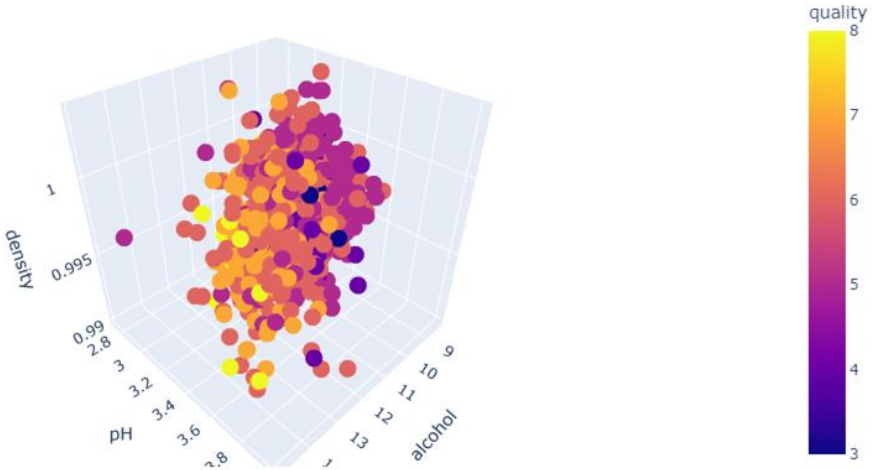
```
import plotly.express as px
df = pd.read_csv(‘titanic.csv')
fig = px.histogram(df, x="Age",
color="Survived", marginal="violin")

fig.show()
```

Heatmap



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

# Load the Wine Quality dataset from a CSV file
df = pd.read_csv(‘winequality-red.csv')

# Create a density heatmap using Plotly Express
fig = px.density_heatmap(df, x='alcohol', y='pH',
title='Density Heatmap')
fig.show()
```

3D plots



```
import plotly.express as px
df = pd.read_csv(‘winequality-red.csv')

fig = px.scatter_3d(df, x='alcohol',
y='pH',z='density',color='quality')
fig.show()
```

Activity 02