



Pantech e Learning
DIGITAL LEARNING SIMPLIFIED

Amazon Web Services

MLOps with AWS

Masterclass



Machine Learning

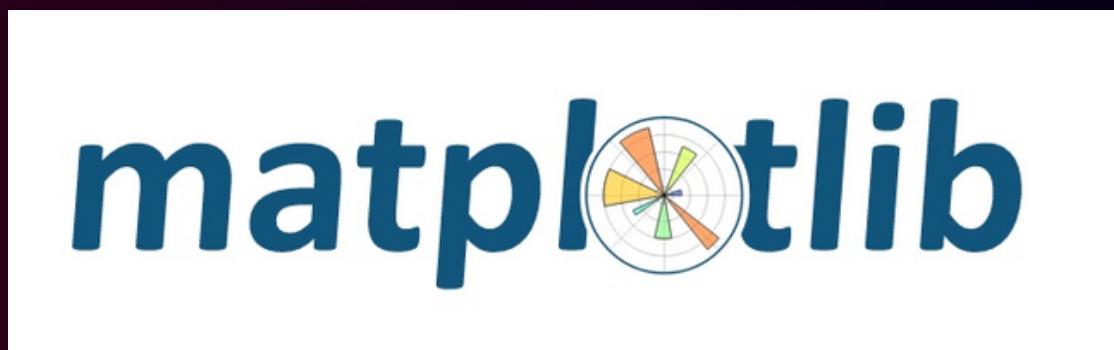
Operations with AWS

Day -9



Pantech e Learning
DIGITAL LEARNING SIMPLIFIED

Data Visualization Library



Data Visualization

- Data visualization is an essential part of data analysis because it helps to communicate complex information and patterns in a simple way.
- By presenting data visually, you can easily identify trends, patterns, and outliers that may not be apparent from raw data.
- Visualization also helps to explore and understand data more effectively, allowing you to make better-informed decisions and communicate your findings to others more clearly.

Data Visualization

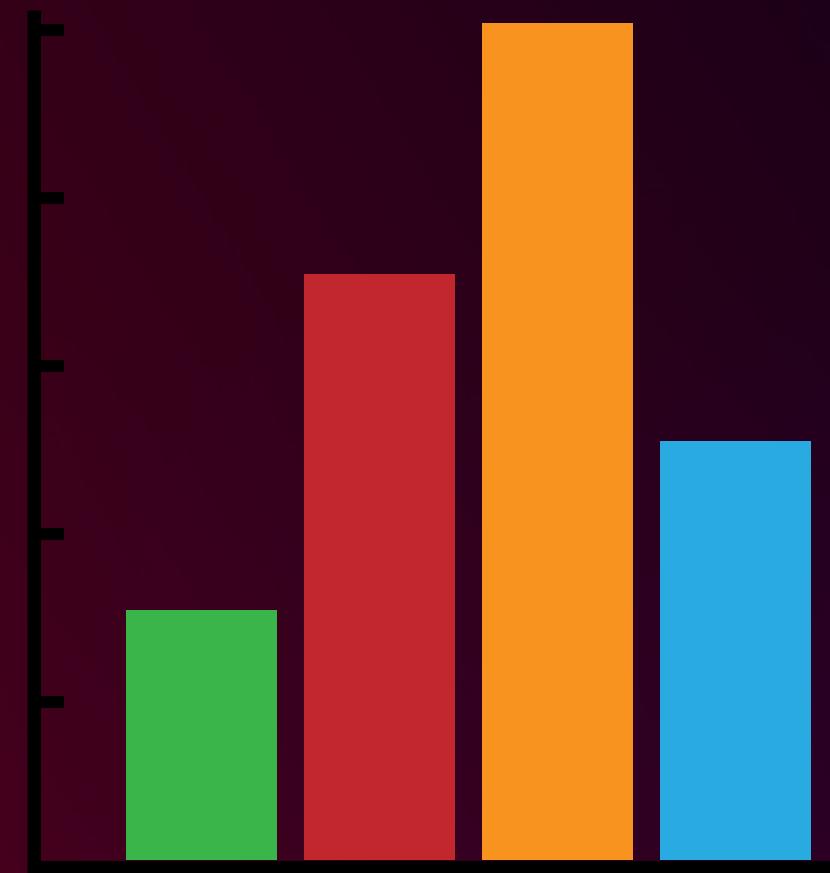
- line graph
- Scatter plot
- Bar plot
- Histogram
- Pie chart
- Heatmap
- Box Plot

Line Graph



- A Line Graph displays continuous data
- Every Line Graph consists of data points that are connected.
- The purpose of connecting their lines is to help illustrate a trend, for example, a change or other pattern.
- Uses of Line Graphs: When you want to show trends over time, for example, how house prices have increased over time

Bar Graph



- A Bar Graph represents discrete data with rectangular bars.
- The x-axis of a Bar Graph presents the discrete categories, and the y-axis shows a measured value.
- Uses of Bar Graphs: When you want to display data that are grouped into discrete categories When you want to compare differences among categories

Pie Chart



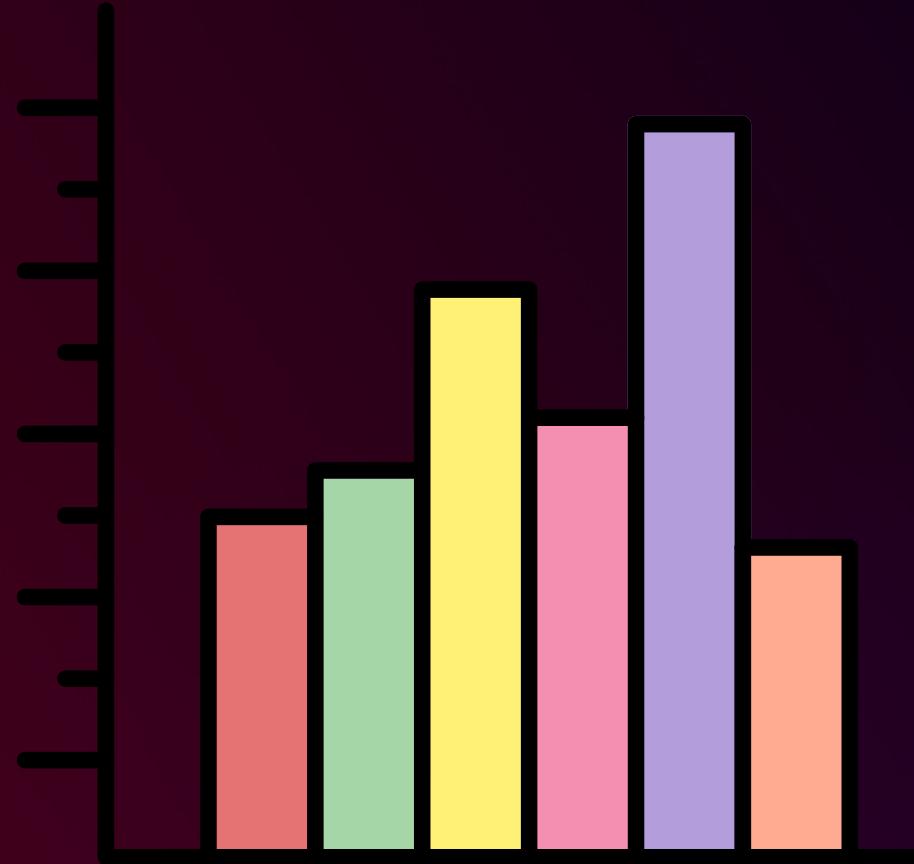
- A pie chart is a circular graph that is divided into slices to represent the relative sizes of different categories in a dataset.
- The size of each slice is proportional to the percentage of the total dataset that belongs to that category.
- They can also be helpful for highlighting the largest and smallest categories in the dataset.

Heat maps

- Heatmaps are a useful type of plot for visualizing the relationship between two variables in a dataset.
- A heatmap can be used to visualize the correlation between variables in a dataset.
- Correlation is a measure of the relationship between two variables.
- Heatmaps can be especially helpful for identifying patterns and trends in complex data.



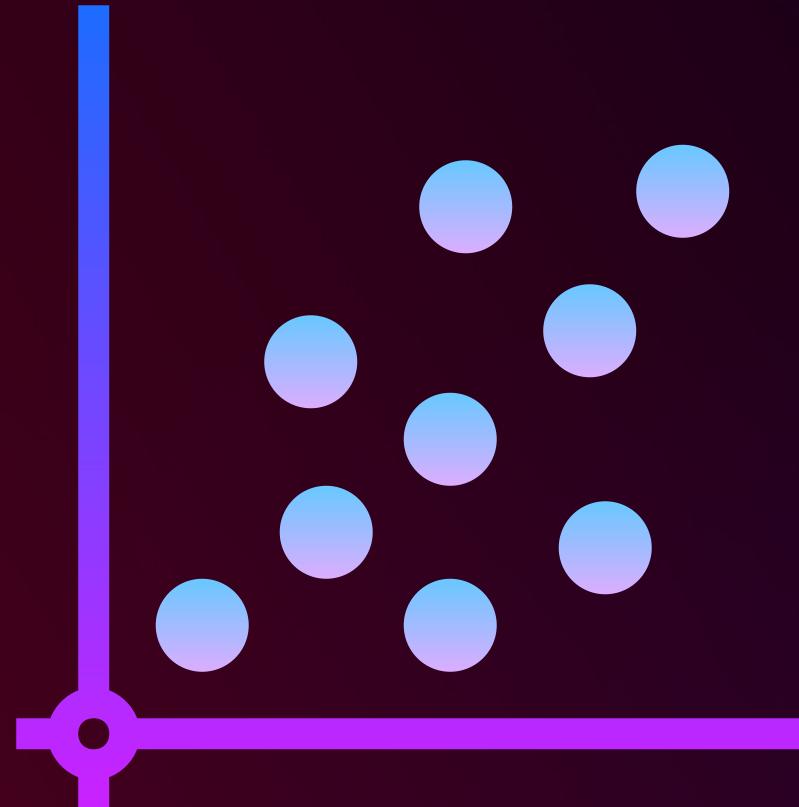
Histogram



- A Histogram shows continuous data in ordered rectangular columns, displays frequency distribution (shape) of a data set.
- Bar graph vs histogram: Bar Graphs represents categorical data and Histograms represent continuous data.
- Uses of Histograms: When the data are continuous. When you want to represent the shape of the data's distribution.

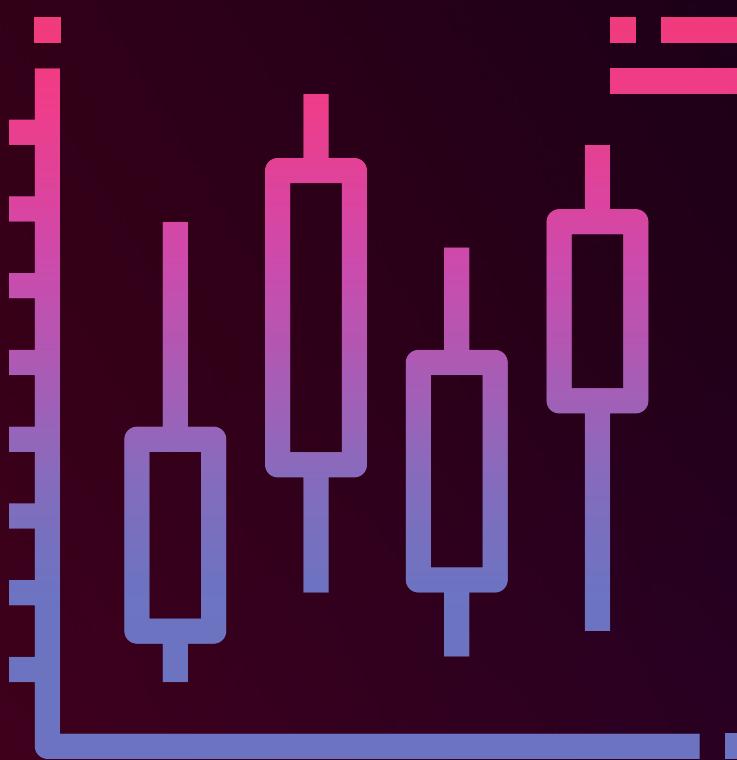
Scatter Plot

- A Scatter Plot is an x-y diagram that shows a relationship between two variables.
- The purpose is to show the relationship between dependent and independent variables
 - A scatter plot can be used to analyze trends over time.
 - For example, you might use a scatter plot to track the sales of a particular product over time and identify any trends, such as seasonal fluctuations or overall growth.



Box Plot

- A boxplot, also known as a box-and-whisker plot, is a useful type of plot for visualizing the distribution of a dataset.
- A boxplot can help you identify outliers in a dataset.
- A boxplot can be used to compare the distribution of different datasets.
- For example, you might use a boxplot to compare the distribution of the heights of men and women in a population.



Matplotlib

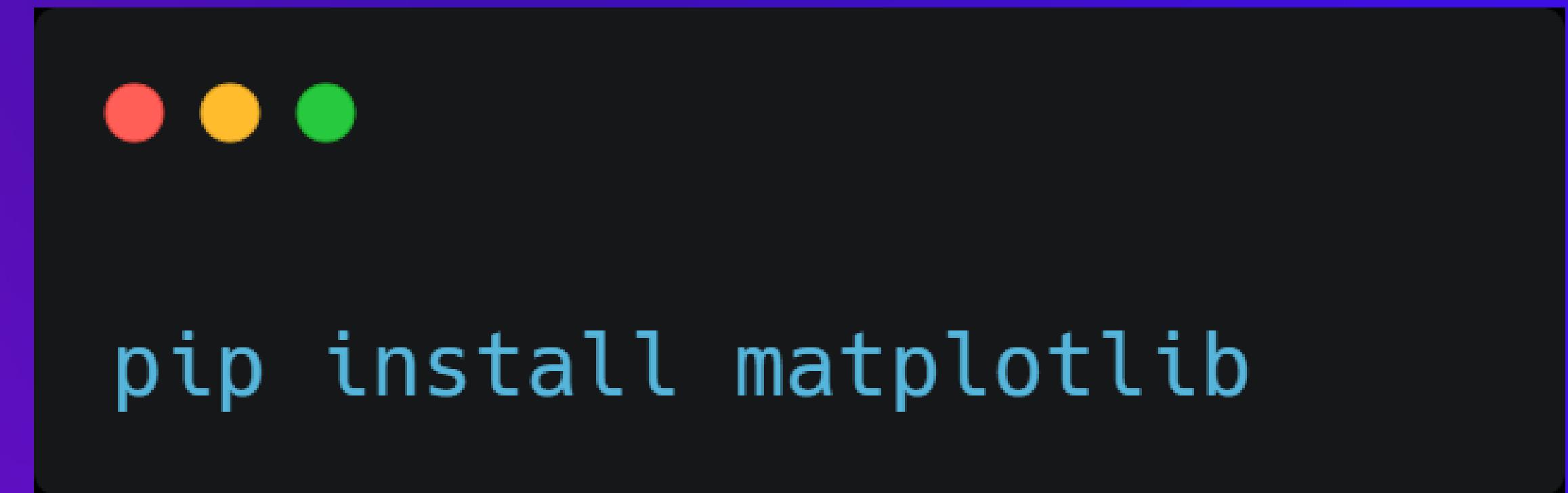


The logo for matplotlib features the word "matplotlib" in a bold, dark blue sans-serif font. The letter "o" in "plot" is replaced by a circular icon containing several colored segments (orange, yellow, green, blue) and radial lines, suggesting a pie chart or a sunburst diagram.

Matplotlib

- Matplotlib is a graph plotting library in python that serves as a visualization utility.
- It has a module named pyplot which makes things easy for plotting by providing many features.
- It is able to create different types of visualization reports like line plots, scatter plots, histograms, bar charts, pie charts, box plots, and many more different plots.

Installation

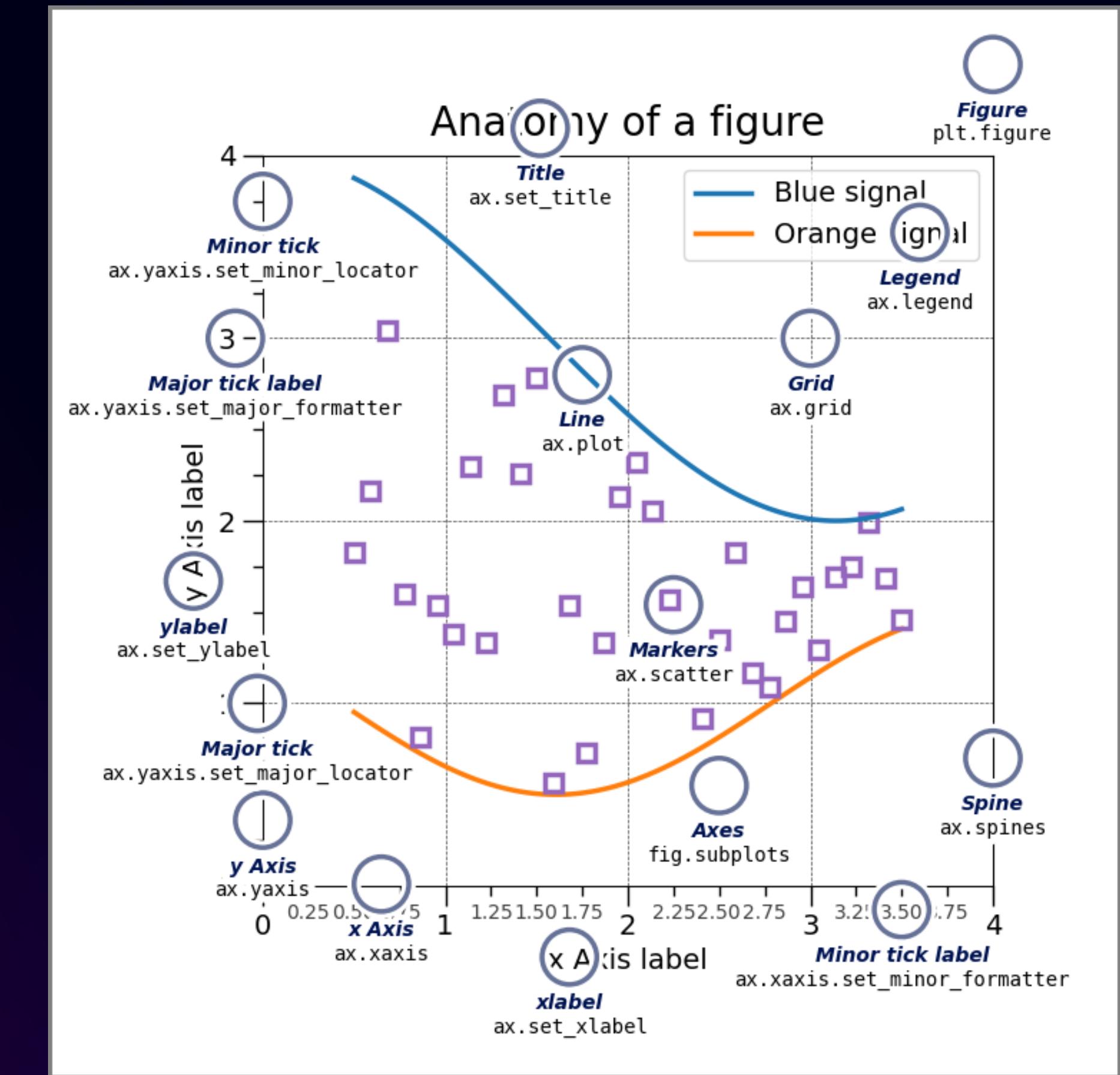


Import



```
import matplotlib.pyplot as plt
```

Anatomy of matplotlib figure



Line plot



```
x = np.array([0, 50])
```

```
y = np.array([0, 50])
```

```
plt.plot(x, y)
```

```
plt.show()
```

Draw Multiple Lines



```
x = np.array([1, 2, 6, 8])
```

```
y = np.array([3, 8, 1, 10])
```

```
plt.plot(x, y)
```

```
plt.show()
```

Line Style

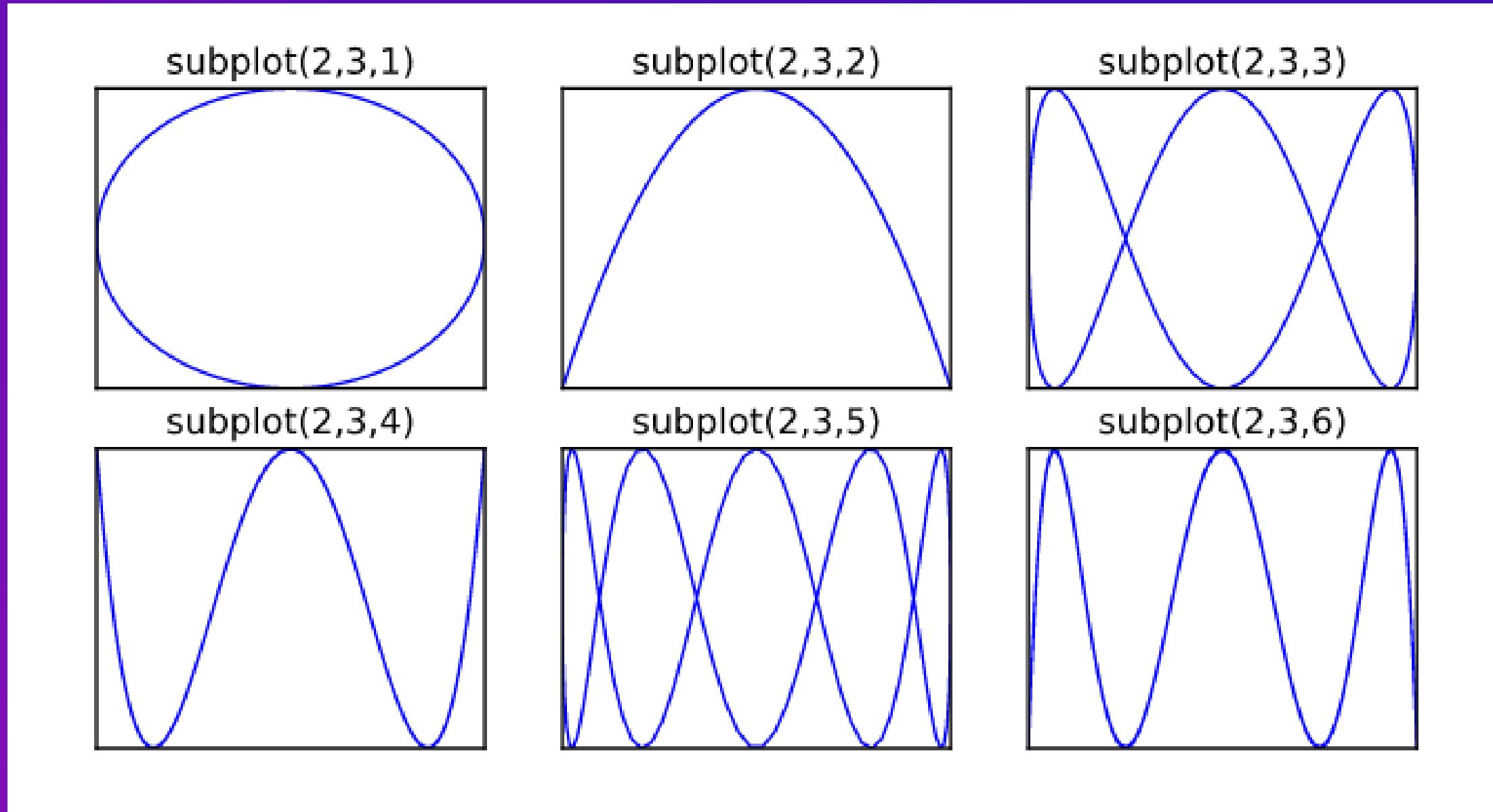


```
x = np.array([2, 6, 3, 10])  
  
plt.plot(x, linestyle = 'dotted')  
  
plt.show()
```

Labels, Title, Grid

```
● ● ●  
  
x = np.array([20, 25, 30, 35, 40, 45, 50, 55, 60, 65])  
  
y = np.array([25, 40, 55, 70, 85, 100, 115, 130, 145, 160])  
  
  
plt.plot(x, y)  
  
plt.xlabel("Average Pulse")  
  
plt.ylabel("Calorie Burnage")  
  
plt.title("Sports Watch Data")  
  
plt.grid()  
  
plt.show()
```

Sub plots



```
x = np.array([0, 1, 2, 3])
```

```
y = np.array([3, 8, 1, 10])
```

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

```
plt.plot(x,y)
```

```
x = np.array([0, 1, 2, 3])
```

```
y = np.array([10, 20, 30, 40])
```

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

```
plt.plot(x,y)
```

```
plt.show()
```

Scatter Plot

```
● ● ●  
  
x1 = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])  
  
y1 = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])  
  
plt.scatter(x1, y1)  
  
  
  
x2 = np.array([2,2,8,1,15,8,12,9,7,3,11,4,7,14,12])  
  
y2 = np.array([100,105,84,105,90,99,90,95,94,100,79,112,91,80,85])  
  
plt.scatter(x2, y2)  
  
  
plt.show()
```

Bar Plot

```
● ● ●  
  
x = np.array( [ "A", "B", "C", "D" ] )  
  
y = np.array( [ 3, 8, 1, 10 ] )  
  
plt.bar(x,y)
```

Horizontal Bars

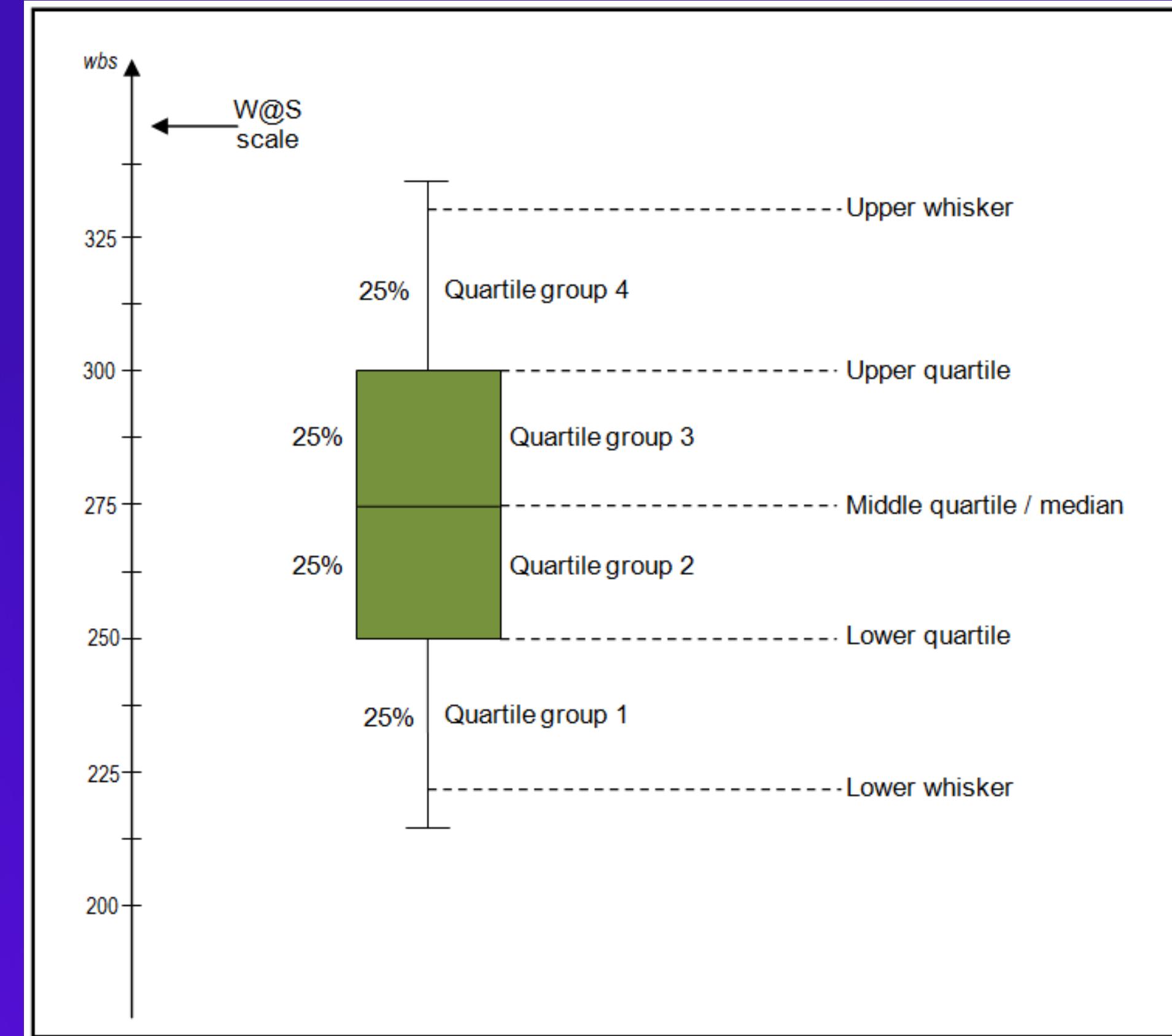
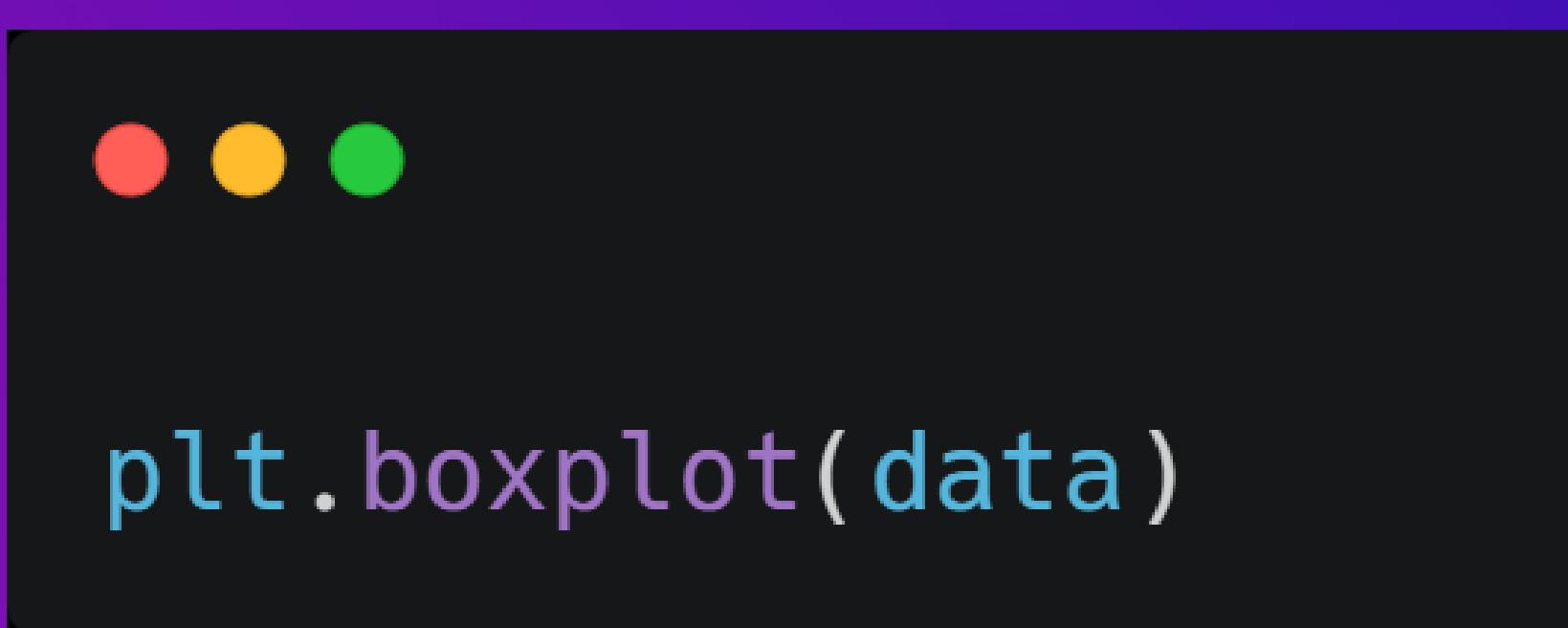


```
x = np.array( ["A", "B", "C", "D"] )  
  
y = np.array( [3, 8, 1, 10] )  
  
plt.barh(x,y)
```

Histogram

```
● ● ●  
  
x = [21,22,23,4,5,6,77,8,9,10,31,32,33,34,35,36,37,18,49,50,100]  
  
plt.hist(x, bins)  
plt.show()
```

Box plot



Pie Chart



```
y = np.array([35, 25, 25, 15])
```

```
plt.pie(y)
```

```
plt.show()
```

Pie Chart



```
y = np.array([35, 25, 25, 15])  
  
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]  
  
plt.pie(y, labels = mylabels)  
  
plt.legend()  
  
plt.show()
```

Seaborn



Thank you



Pantech e Learning
DIGITAL LEARNING SIMPLIFIED