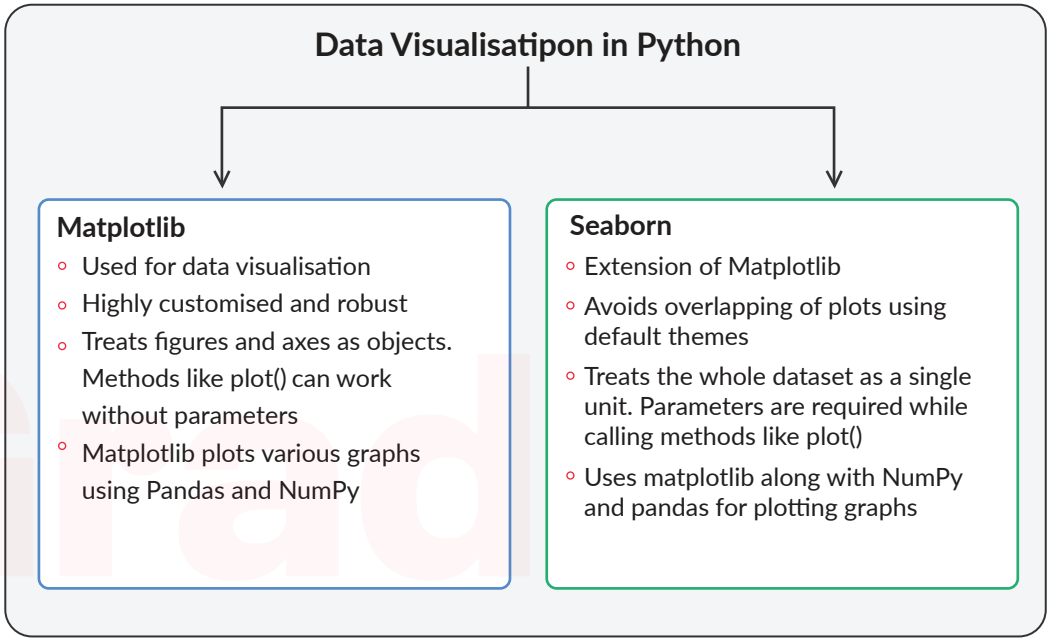
Data Visualisation in Python

Data visualisation is a highly important skill for anyone trying to extract and communicate insights from data. You learn to visualise data using two libraries in python, Matplotlib and Seaborn
These libraries are useful in deriving insights from data and highlighting trends, outliers and other important aspects

Common Interview Questions:

1. What is data visualization?
2. Which are the best libraries for data visualization in Python?
3. Why is data cleansing important for data visualization?
4. What is a scatter plot? Scatter plots are used for which type of data?
5. What features might be visible in scatter plots?
6. When do you use a histogram and a bar chart? Explain with examples.
7. Box plots are used for which type of data?
8. What information can you gain from a box plot?
9. What are stacked plots used for?
10. What is a heatmap in Python?



Different Scenarios	Useful Graphs
Categorical data composition	Bar plot, pie chart and donut chart
Data composition over time	Area chart
Two or more-dimensional data composition	Stacked columns chart
Data distribution	Histogram, scatter plot and boxplot
Statistical summary such as mean, median, outliers	Boxplot
Categorical values comparison	Column chart, grouped column chart and bar plot
Value comparison over time	Line chart and overlay line

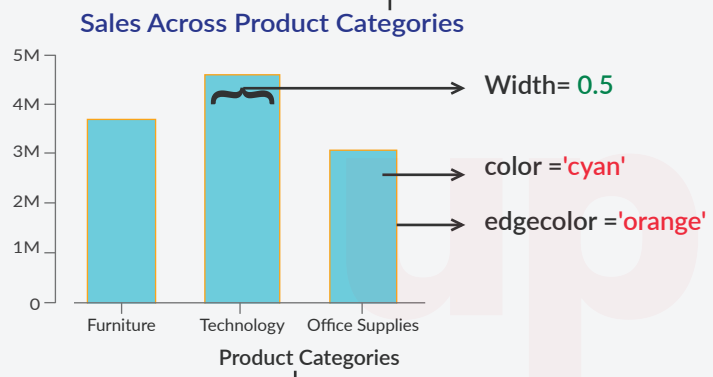
Data Visualisation in Python

Import Necessary Library

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

Bar plot: Compare the values of different categories in the data. Datatype: Categorical

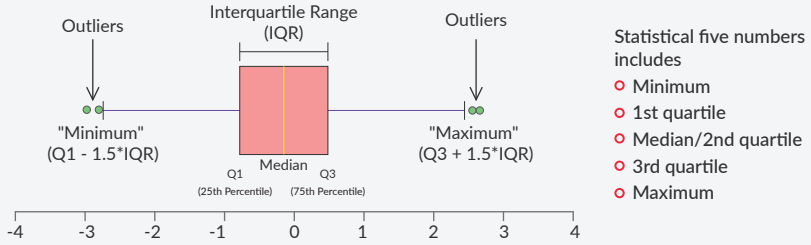
```
plt.bar(product_category, sales, width= 0.5, align='center',
edgecolor='Orange', color='cyan')
plt.title("Sales Across Product Categories\n", fontdict={'fontsize': 20,
'fontweight' : 5, 'color': 'Green' })
```




```
plt.xlabel("Product Category", fontdict={'fontsize': 12,
'fontweight' : 5, 'color' : 'Brown'})
plt.ylabel("Sales", fontdict {'fontsize': 12, 'fontweight' : 5,
'color' :  })
```

Box plot: Visual representation of the statistical five numbers summary of a given data set. Datatype: Numerical

```
sns.boxplot(data = df)
```



Statistical five numbers includes

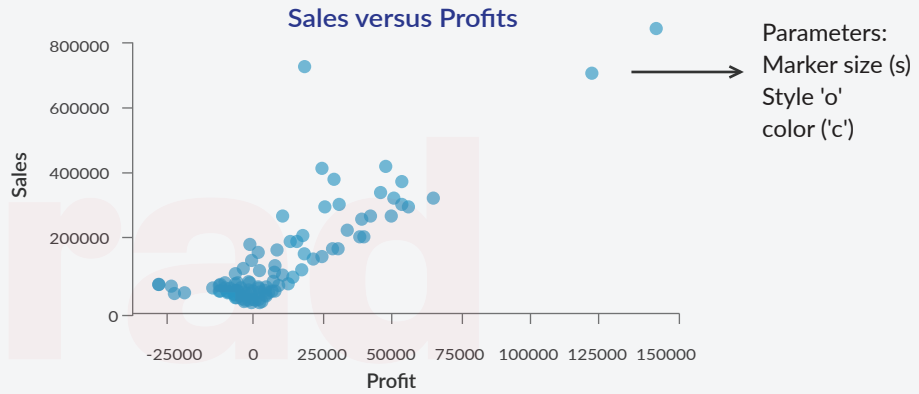
- Minimum
- 1st quartile
- Median/2nd quartile
- 3rd quartile
- Maximum

Data Types:

- Numerical: Data in the form of numbers
- Categorical: Data with two or more categories

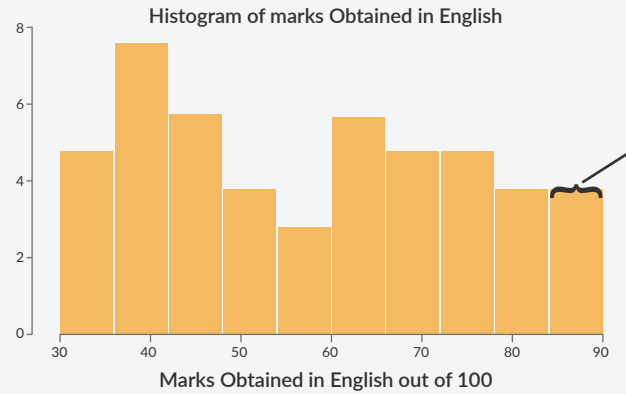
Scatter plot: To identify a relationship or pattern between two quantitative variables and the presence of outliers within them. Datatype: Numerical

```
plt.scatter(profit, sales,alpha= 0.7, s = 50)
plt.show()
```



Histogram: Frequency chart that records the number of occurrences of an entry or an element in a data set. It is useful to understand the distribution of a given series. Datatype: Numerical and categorical

```
sns.histplot(data+df,x ='English', edgecolor ='linen', alpha = 0.5, bins = 10)
```

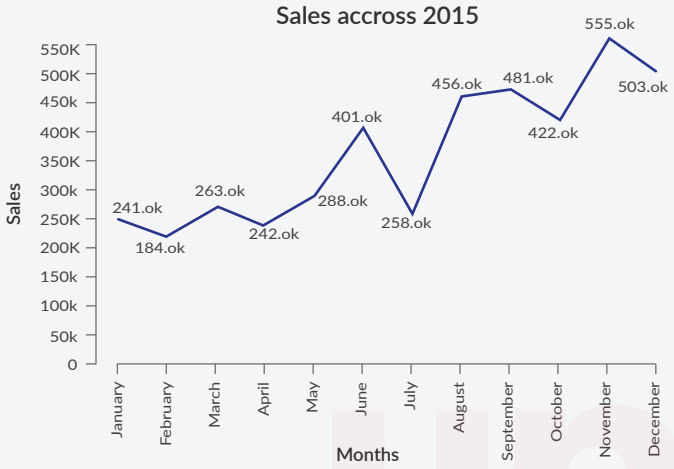


Note: 'alpha' varies between 0 and 1 and represents the transparency of a histogram

Data Visualisation in Python

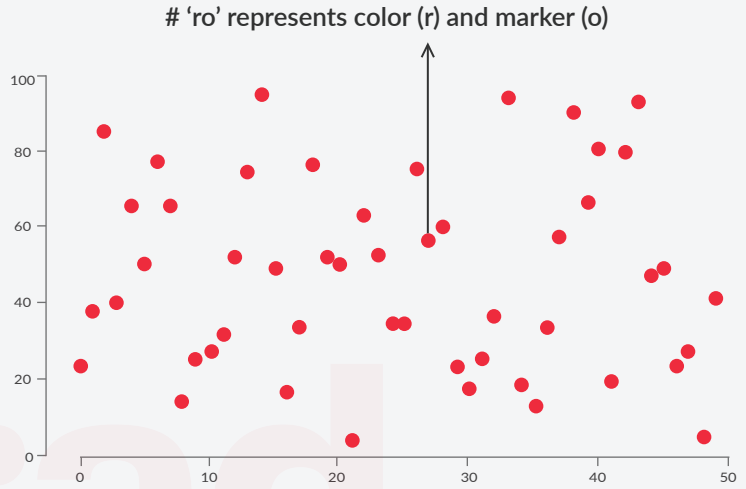
Line graph: Depicts the trend of a variable over a specified time period.
Datatype: Numerical and categorical

```
plt.plot(x_axis, y_axis)
```



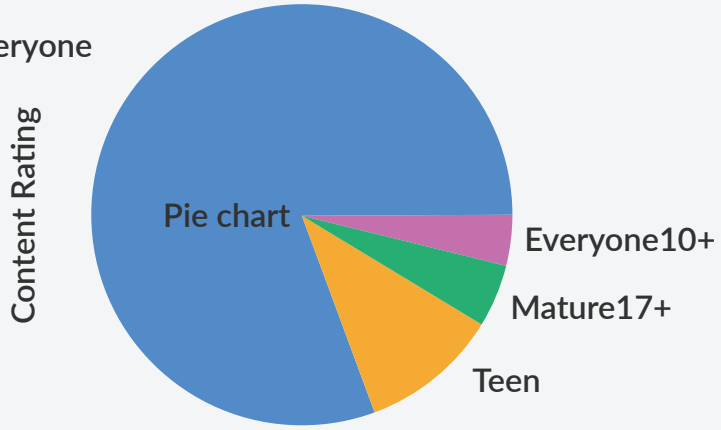
Caution: plt.plot also create scatter plot

```
y= np.random.randint(1,100, 50)
plt.plot(y, 'ro')
```

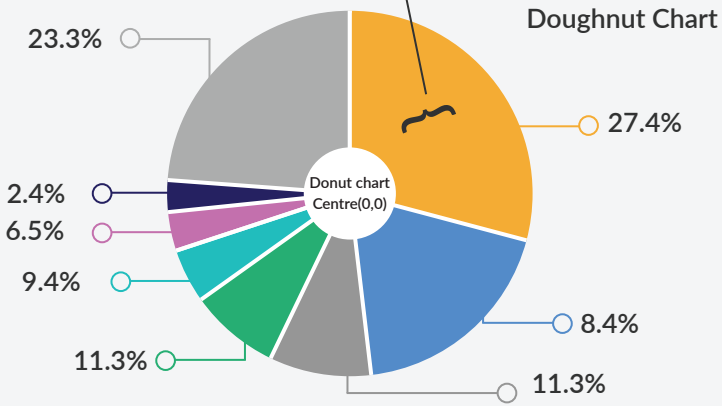


Pie-chart and donut chart: Percentage of data present in each category.
Datatype: Categorical

```
matplotlib.pyplot .pie(data)
```



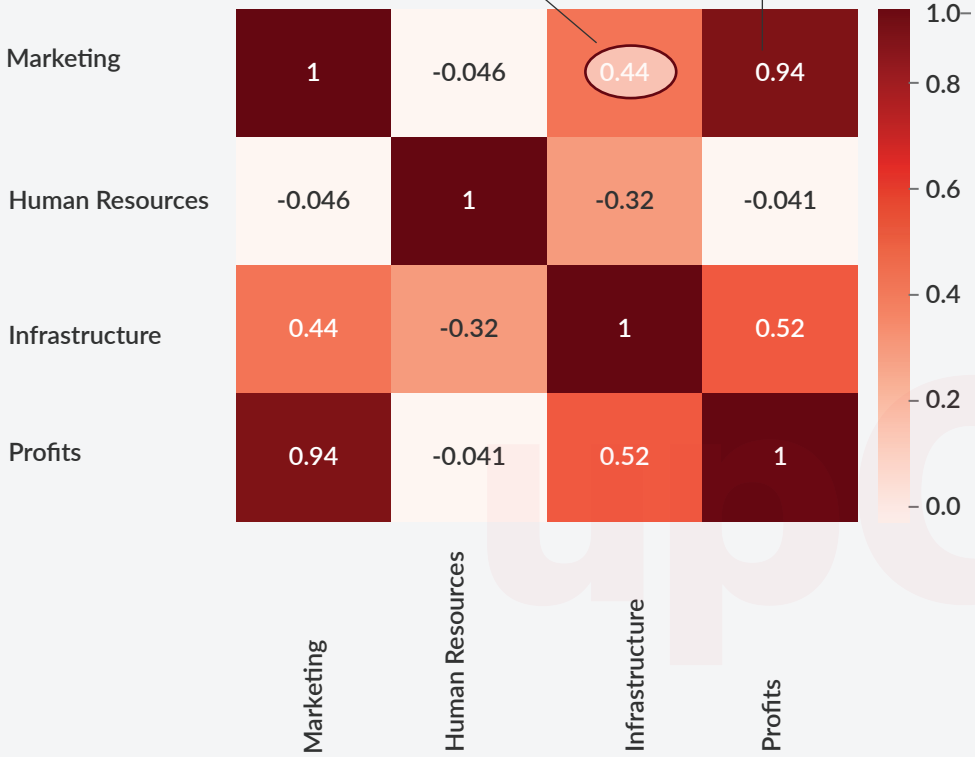

```
matplotlib.pyplot.pie(data)
#create circle
circle = plt.Circle( (0,0), 0.7, color='white')
P=plt.gcf()
#adding the circle to the centre of pie chart
p.gca().add_artist(circle)
```



Data Visualisation in Python

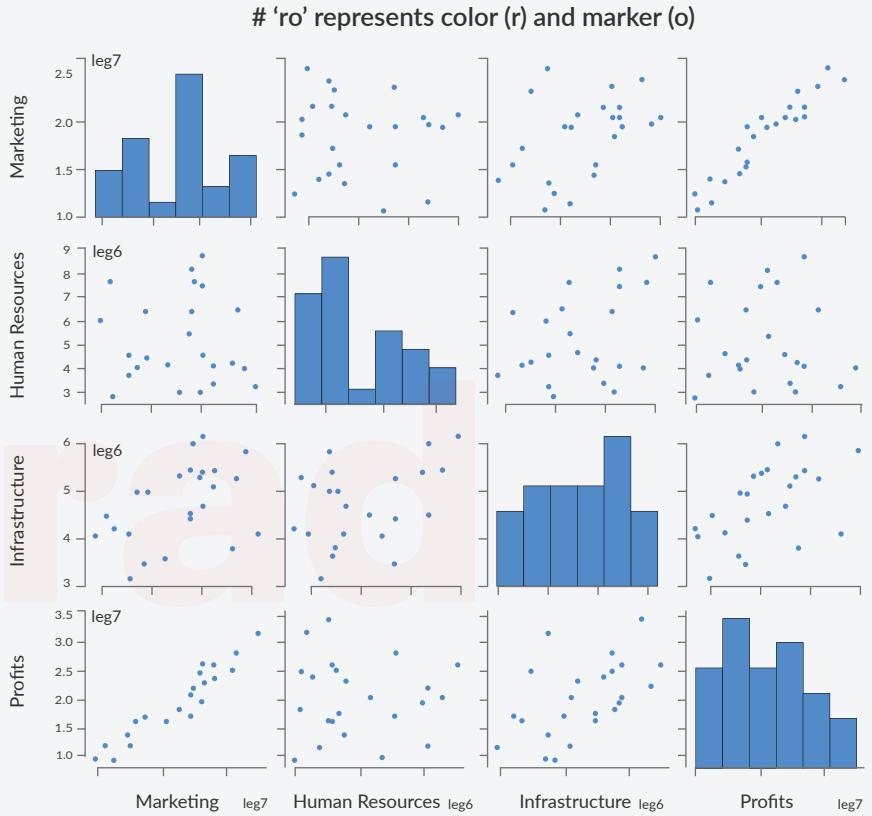
Heatmap: To show the correlation matrix

```
sns.heatmap(df.corr(), annot = True, camp = 'Reds')
```



Pair plot: To determine pairwise relationships in a dataset

```
sns.pairplot(df);
```



Note:

- Positive correlation value implies two variables are positively correlated
- Negative correlation value implies two variables are negatively correlated
- A variable is strongly correlated with itself that is the correlation value between the variable and itself is always 1