

EXP 4 INLAB

1. Importing Libraries and creating dataset:-

The data contains the courses taken by a student and the marks obtained by that student in those courses out of 100.

```
# Importing Libraries
import pandas as pd
import matplotlib.pyplot as plt

[2] # Creating dataset
data = {'EDA':80, 'SM':75, 'OOSE':98, 'OET':66, 'OEHM':43, 'CC':57}
courses = list(data.keys())
marks = list(data.values())

[3] # Showing Dataset
data

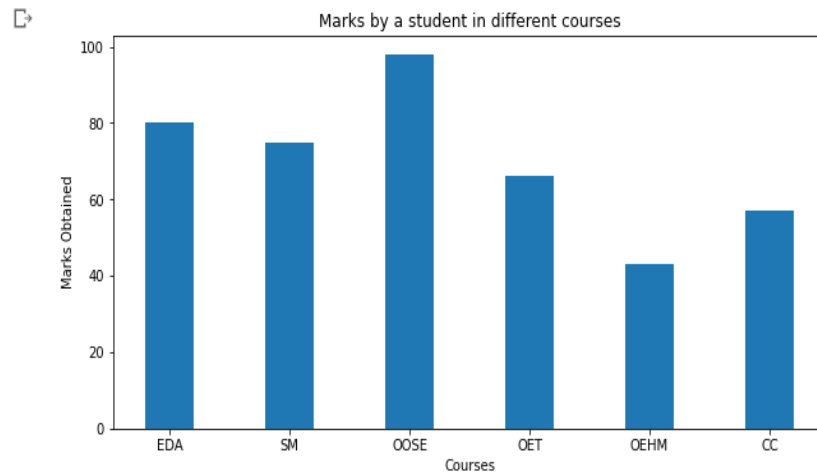
{'CC': 57, 'EDA': 80, 'OEHM': 43, 'OET': 66, 'OOSE': 98, 'SM': 75}
```

2. Bar Chart:-

A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. The bar plots can be plotted horizontally or vertically. A bar chart describes the comparisons between the discrete categories.

```
# Bar Chart
fig = plt.figure(figsize = (10, 5))

plt.bar(courses, marks, width = 0.4)
plt.xlabel("Courses")
plt.ylabel("Marks Obtained")
plt.title("Marks by a student in different courses")
plt.show()
```

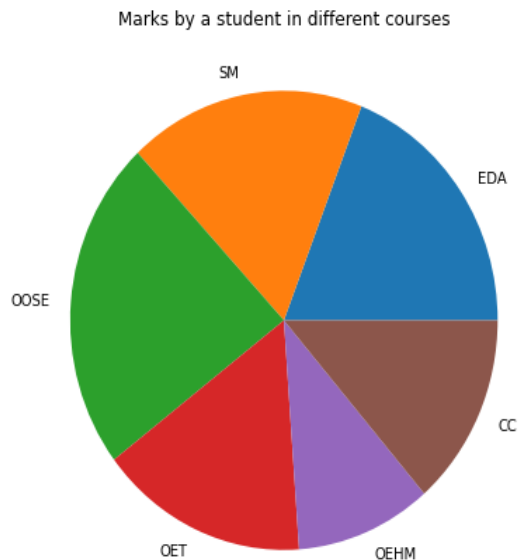


3. Pie Chart:-

A **Pie Chart** is a circular statistical plot that can display only one series of data. The area of the chart is the total percentage of the given data. The area of slices of the pie represents the percentage of the parts of the data. The slices of pie are called wedges. The area of the wedge is determined by the length of the arc of the wedge. The area of a wedge represents the relative percentage of that part with respect to whole data.

```
# Pie Chart
fig = plt.figure(figsize =(10, 7))

plt.pie(marks, labels = courses)
plt.title("Marks by a student in different courses")
plt.show()
```

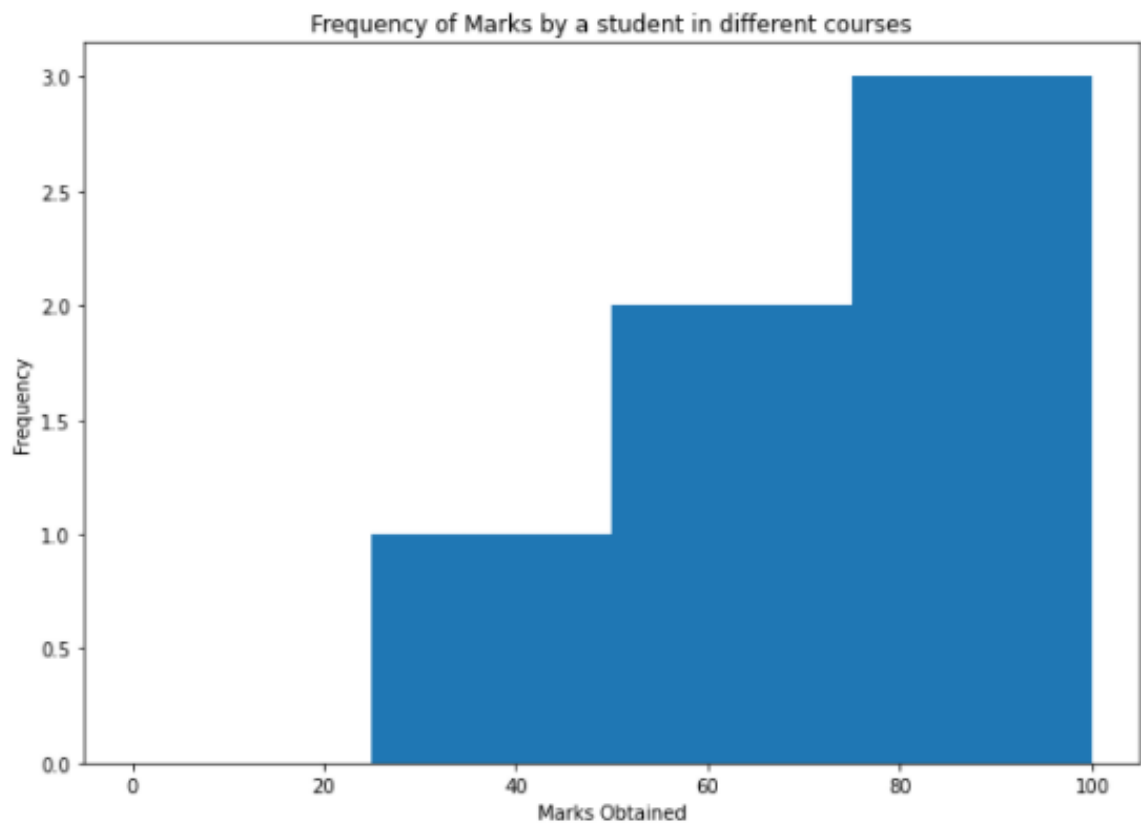


4. Histogram:-

A histogram is basically used to represent data provided in a form of some groups. It is an accurate method for the graphical representation of numerical data distribution. It is a type of bar plot where X-axis represents the bin ranges while Y-axis gives information about frequency.

```
[16] # Histogram
fig, ax = plt.subplots(figsize=(10, 7))
ax.hist(marks, bins = [0, 25, 50, 75, 100])

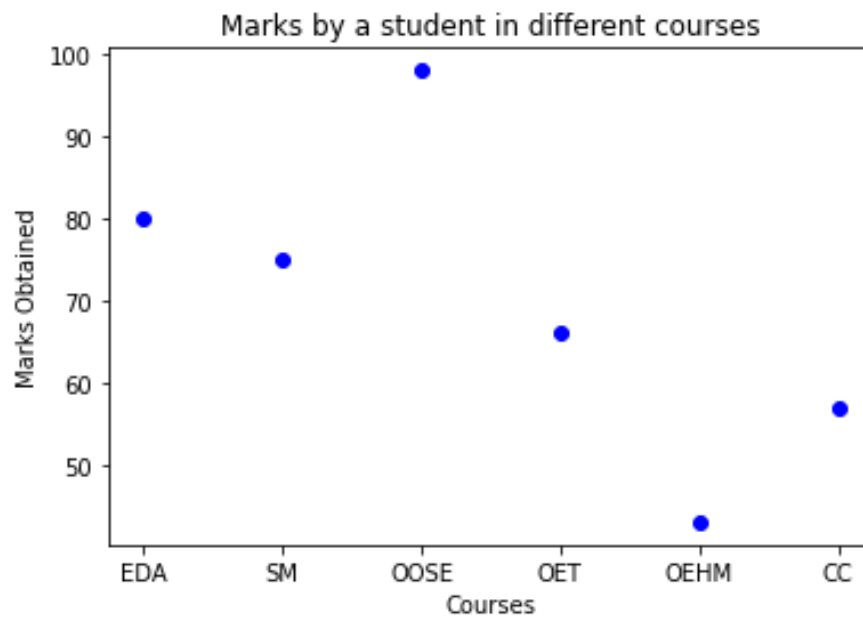
plt.ylabel("Frequency")
plt.xlabel("Marks Obtained")
plt.title("Frequency of Marks by a student in different courses")
plt.show()
```



5. Scatter Plot:-

Scatter plots are used to observe relationship between variables and uses dots to represent the relationship between them.

```
[18] # Scatter Plot
plt.scatter(courses, marks, c = "blue")
plt.xlabel("Courses")
plt.ylabel("Marks Obtained")
plt.title("Marks by a student in different courses")
plt.show()
```

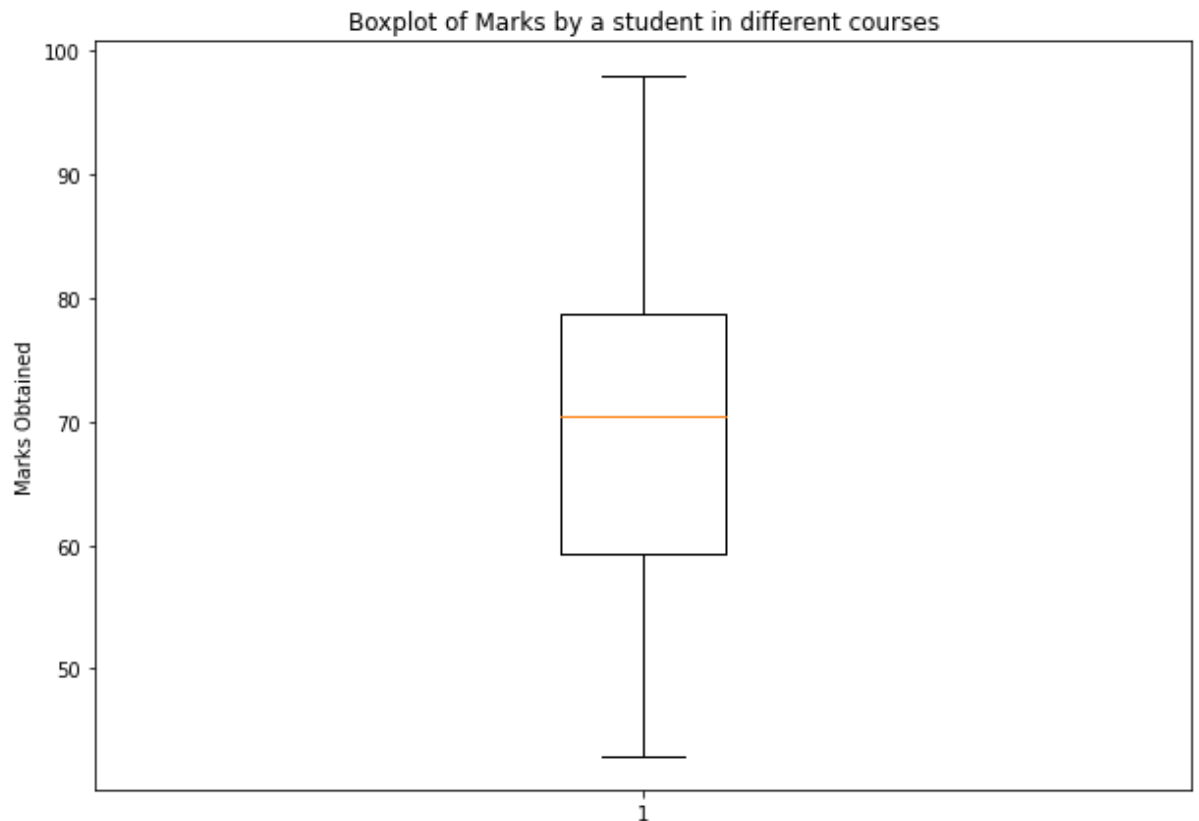


6. Quartile Plot (Boxplot):-

A **Box Plot** is also known as **Whisker plot** is created to display the summary of the set of data values having properties like minimum, first quartile, median, third quartile and maximum. In the box plot, a box is created from the first quartile to the third quartile; a vertical line is also there which goes through the box at the median.

```
[23] # Quartile Plot (Boxplot)
      fig = plt.figure(figsize =(10, 7))

      plt.boxplot(marks)
      plt.ylabel("Marks Obtained")
      plt.title("Boxplot of Marks by a student in different courses")
      plt.show()
```



7. Subplots:-

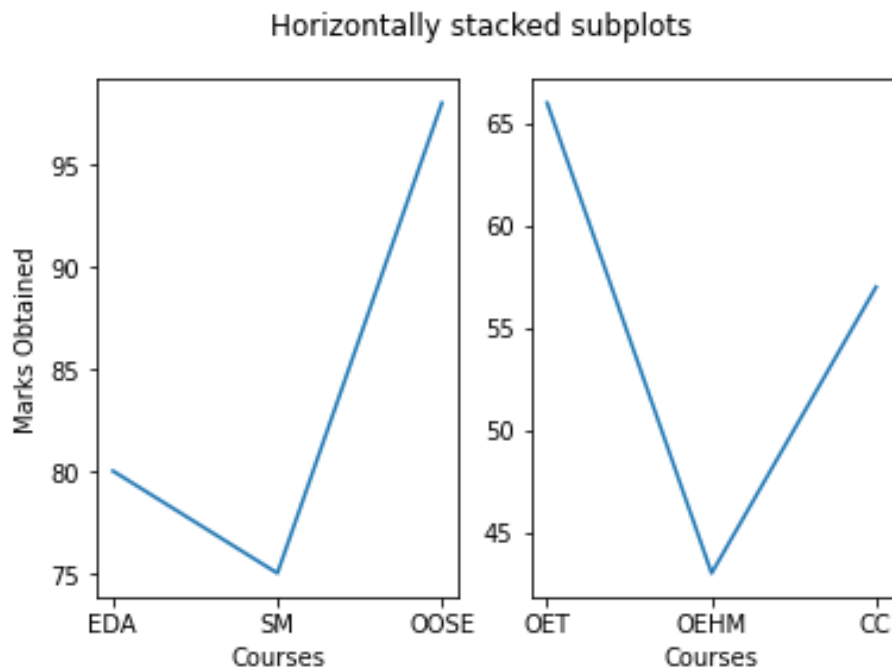
Subplots mean groups of axes that can exist in a single matplotlib figure. `subplots()` function in the matplotlib library, helps in creating multiple layouts of subplots. It provides control over all the individual plots that are created.

```
[34] # Subplots
fig, (ax1, ax2) = plt.subplots(1, 2)
fig.suptitle('Horizontally stacked subplots')

ax1.plot(courses[:3], marks[:3])
ax1.set_xlabel("Courses")
ax1.set_ylabel("Marks Obtained")

ax2.plot(courses[3:], marks[3:])
ax2.set_xlabel("Courses")

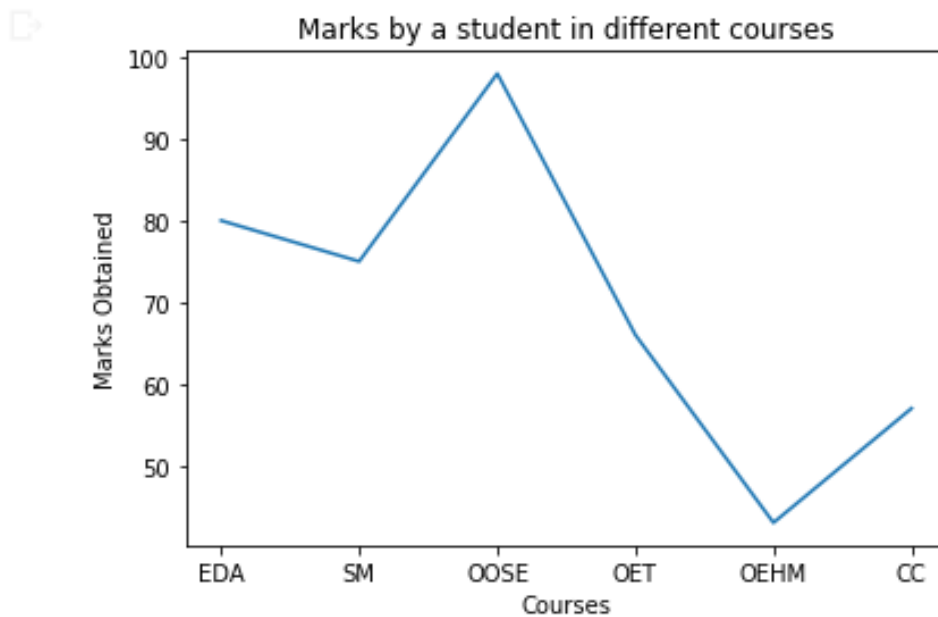
plt.show()
```



8. Line Chart:-

Line charts are used to represent the relation between two data X and Y on a different axis.

```
[35] # Line Chart
plt.plot(courses, marks)
plt.xlabel("Courses")
plt.ylabel("Marks Obtained")
plt.title("Marks by a student in different courses")
plt.show()
```

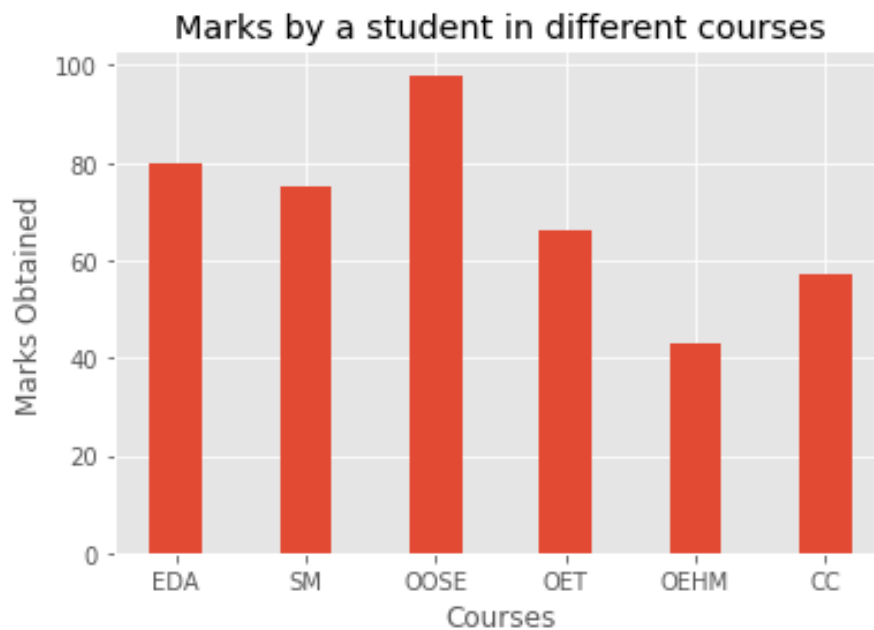


9. Ggplot:-

This example demonstrates the "ggplot" style, which adjusts the style to emulate ggplot (a popular plotting package for R).

```
[38] # Ggplot
plt.style.use('ggplot')

plt.bar(courses, marks, width = 0.4)
plt.xlabel("Courses")
plt.ylabel("Marks Obtained")
plt.title("Marks by a student in different courses")
plt.show()
```



10. QQ Plot:-

When the quantiles of two variables are plotted against each other, then the plot obtained is known as quantile – quantile plot or qqplot. This plot provides a summary of whether the distributions of two variables are similar or not with respect to the locations.

```
# QQ Plot
import numpy as np
import statsmodels.api as sm
import pylab as py

data_points = np.random.normal(0, 1, 100)

sm.qqplot(data_points, line = '45')
py.show()
```

