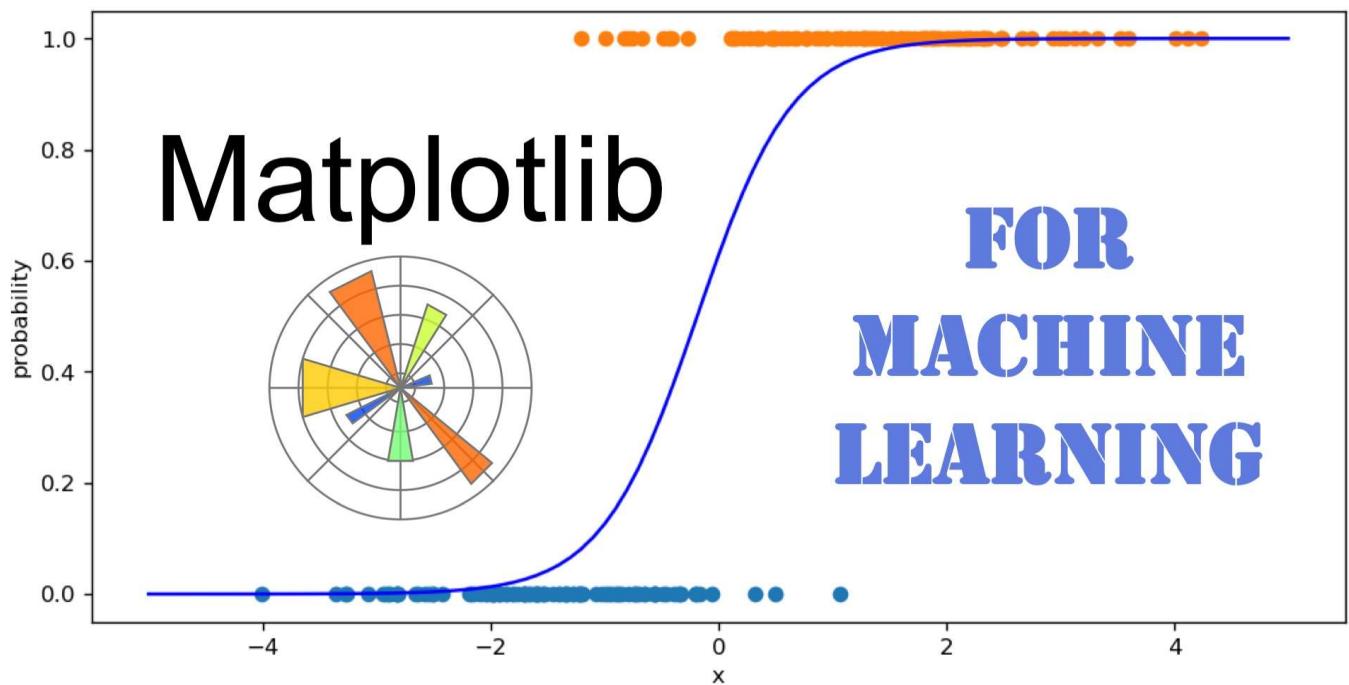


Goeduhub_Assignment/TASK: 6

By :- Mohit Mahi



This assignment is for visualization using Python Matplotlib Library

1. Load the necessary package for plotting using pyplot from matplotlib. Example - Days(x-axis) represents 8 days and Speed represents a car's speed. Plot a Basic line plot between days and car speed, put x axis label as days and y axis label as car speed and put title Car Speed Measurement.

Days=[1,2,3,4,5,6,7,8]

Speed=[60,62,61,58,56,57,46,63]

In []:

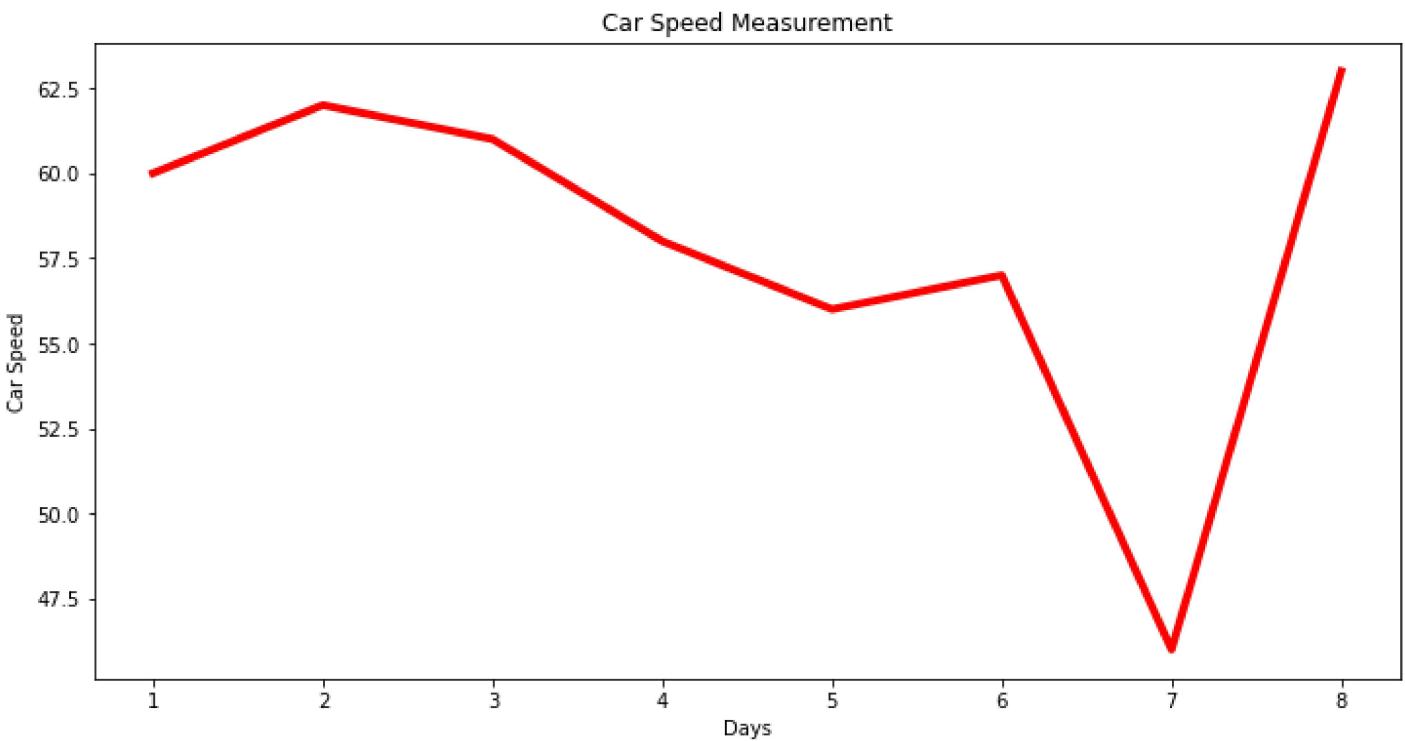
```
import numpy as np  
import matplotlib.pyplot as plt
```

In []:

```
Days = np.array([1,2,3,4,5,6,7,8])  
Speed = np.array([60,62,61,58,56,57,46,63])
```

In []:

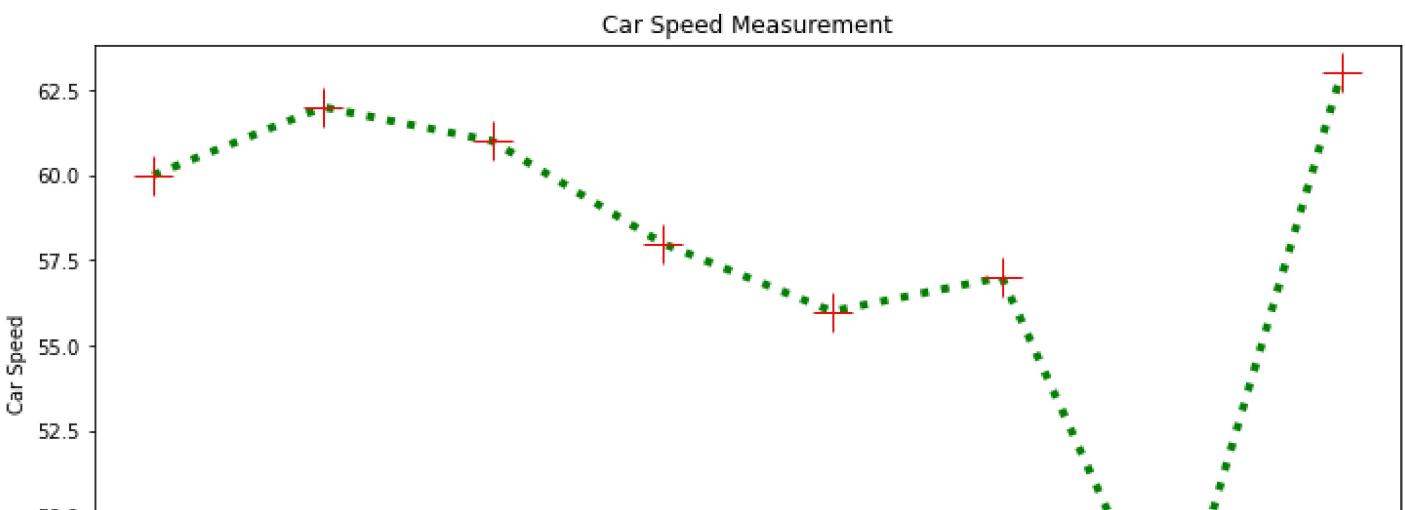
```
plt.figure(figsize=(12,6))
plt.plot(Days, Speed, linewidth = 4, color= 'red')
plt.xlabel('Days')
plt.ylabel('Car Speed')
plt.title('Car Speed Measurement')
plt.show()
```

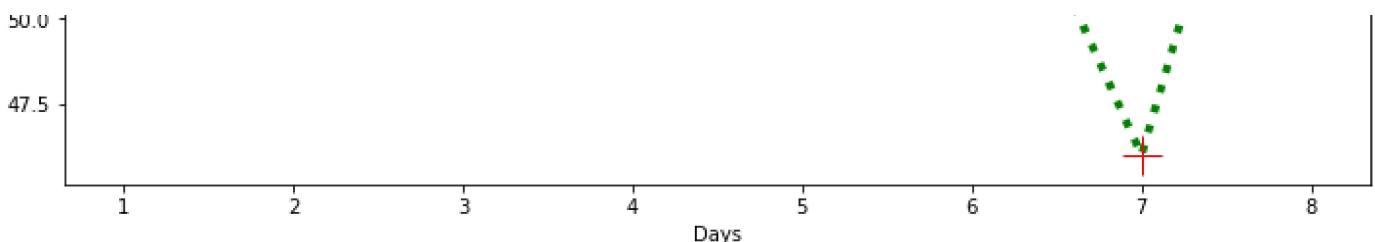


2. Now to above car data apply some string formats like line style example green dotted line, marker shape like +, change markersize, markerface color etc.

In []:

```
plt.figure(figsize=(12,6))
plt.plot(Days, Speed, linewidth = 4, color='green', linestyle = ':',
         marker = '+', markersize = 20, markeredgecolor = 'r')
plt.xlabel('Days')
plt.ylabel('Car Speed')
plt.title('Car Speed Measurement')
plt.show()
```





3. Plot Axes Labels, Chart title, Legend, Grid in Car minimum, Maximum and average speed in 8 days.

days=[1,2,3,4,5,6,7,8]

max_speed=[80,91,92,88,77,79,76,75]

min_speed=[42,43,40,42,33,36,34,35]

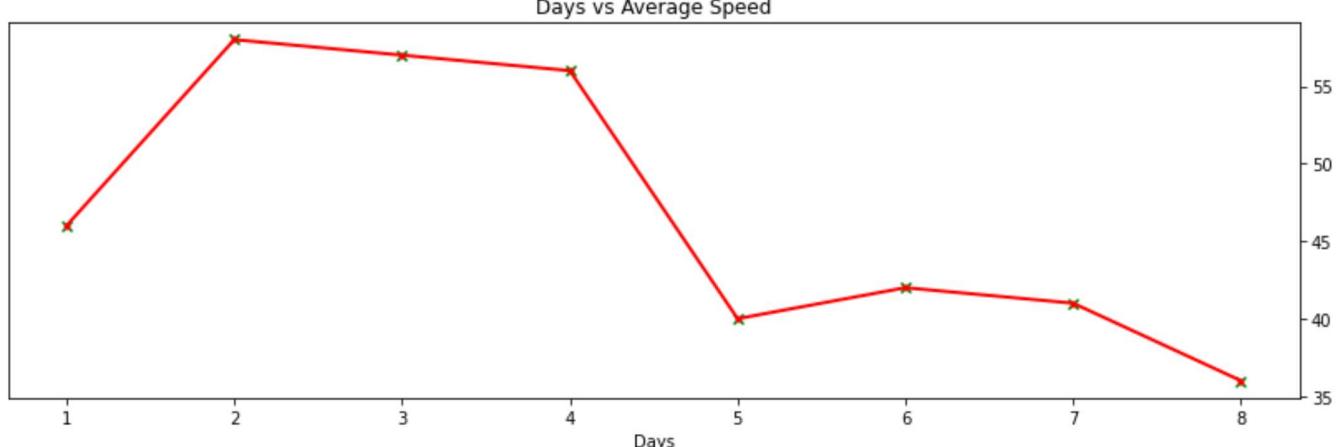
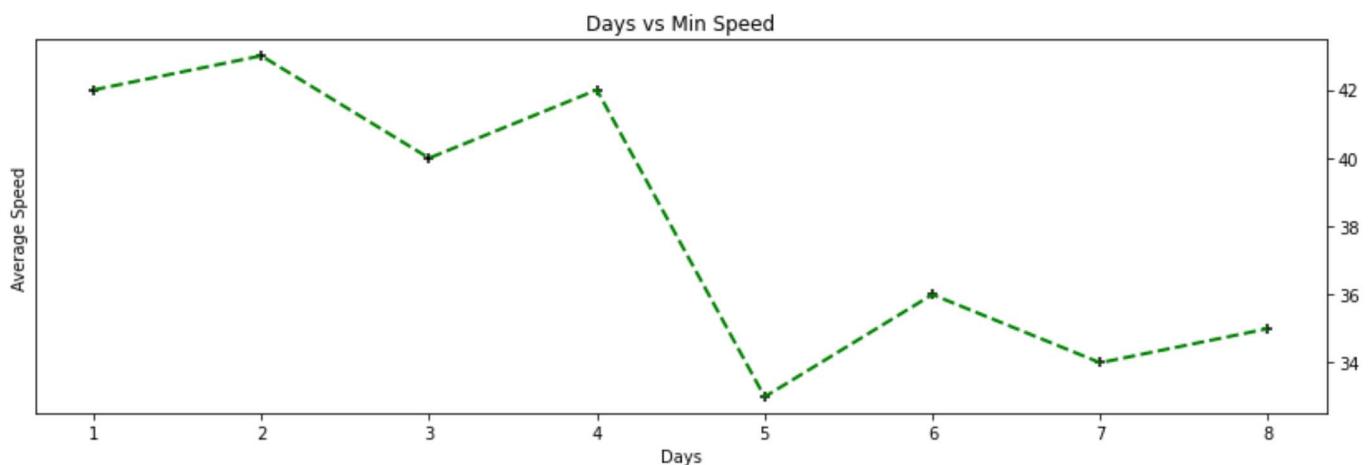
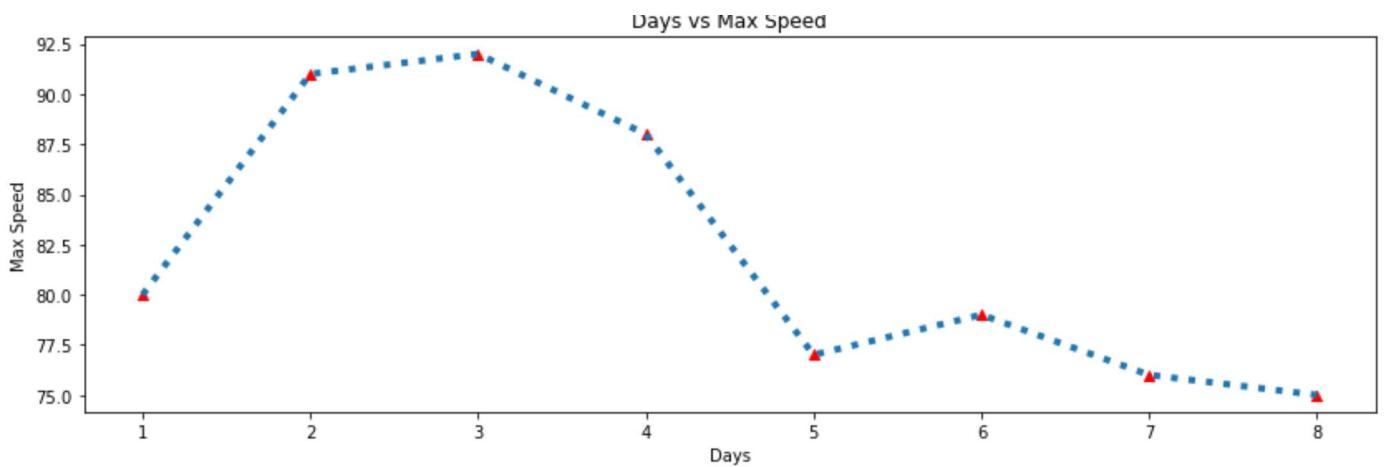
avg_speed=[46,58,57,56,40,42,41,36]

In []:

```
days = np.array([1,2,3,4,5,6,7,8])
max_speed = np.array([80,91,92,88,77,79,76,75])
min_speed = np.array([42,43,40,42,33,36,34,35])
avg_speed = np.array([46,58,57,56,40,42,41,36])
```

In []:

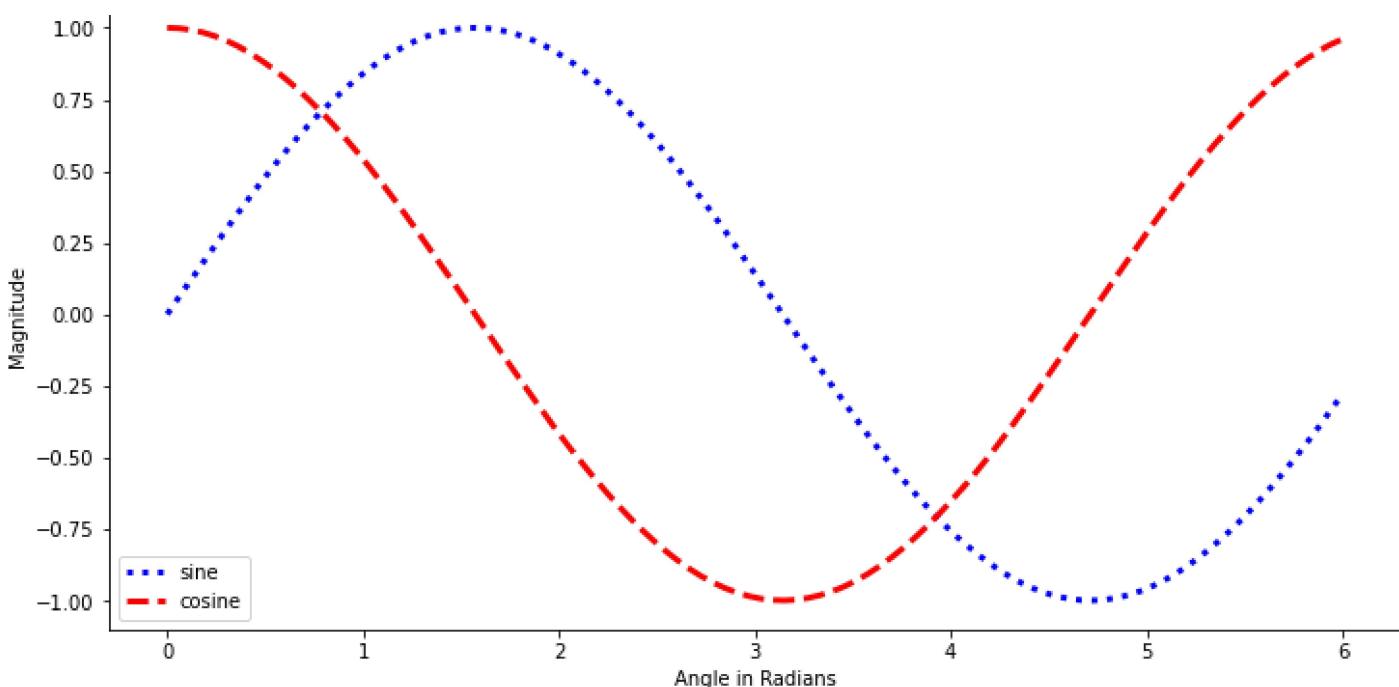
```
fig, (ax1, ax2, ax3) = plt.subplots(nrows=3, ncols=1, figsize = (12,12))
# customizing axis 1
ax1.plot(days,max_speed,':', linewidth=4)
ax1.scatter(days,max_speed,color='red',marker='^')
ax1.set_title('Days vs Max Speed')
ax1.set_xlabel('Days')
ax1.set_ylabel('Max Speed')
ax2.yaxis.tick_right()
# customizing axis 2
ax2.plot(days,min_speed, '--g', linewidth=2)
ax2.scatter(days,min_speed,color='black',marker='+')
ax2.set_title('Days vs Min Speed')
ax2.set_xlabel('Days')
ax2.set_ylabel('Min Speed')
ax2.yaxis.tick_right()
# customizing axis 3
ax3.plot(days,avg_speed, '.-r', linewidth=2)
ax3.scatter(days,avg_speed,color='green',marker='x')
ax3.set_title('Days vs Average Speed')
ax3.set_xlabel('Days')
ax3.set_ylabel('Average Speed')
ax3.yaxis.tick_right()
fig.suptitle('Plots of Speed', y = 1.05, fontsize=16)
fig.tight_layout()
```



4. Plotting a basic sine graph by adding more features. Adding Multiple plots by Superimposition like cosine wave.

In []:

```
x = np.linspace(0,6,100)
y = np.sin(x)
z = np.cos(x)
plt.figure(figsize=(12,6))
plt.plot(x,y,':b', linewidth = 3)
plt.plot(x,z,'--r', linewidth = 3)
plt.xlabel("Angle in Radians")
plt.ylabel("Magnitude")
plt.title("Sine/Cosine function curve")
plt.legend(['sine','cosine'])
plt.show()
```



5. Plot Simple bar chart showing popularity of Programming Languages.

Languages = ['Python', 'SQL', 'Java', 'C++', 'JavaScript']

Popularity = [56, 39, 34, 34, 29]

Security = [44 ,36 ,55, 50, 42]

Plot Multiple Bars showing Popularity and Security of major Programming Languages. Also Create Horizontal bar chart using barh function.

In []:

```
Languages = ['Python', 'SQL', 'Java', 'C++', 'JavaScript']
Popularity = np.array([56, 39, 34, 34, 29])
Security = np.array([44 ,36 ,55, 50, 42])
```

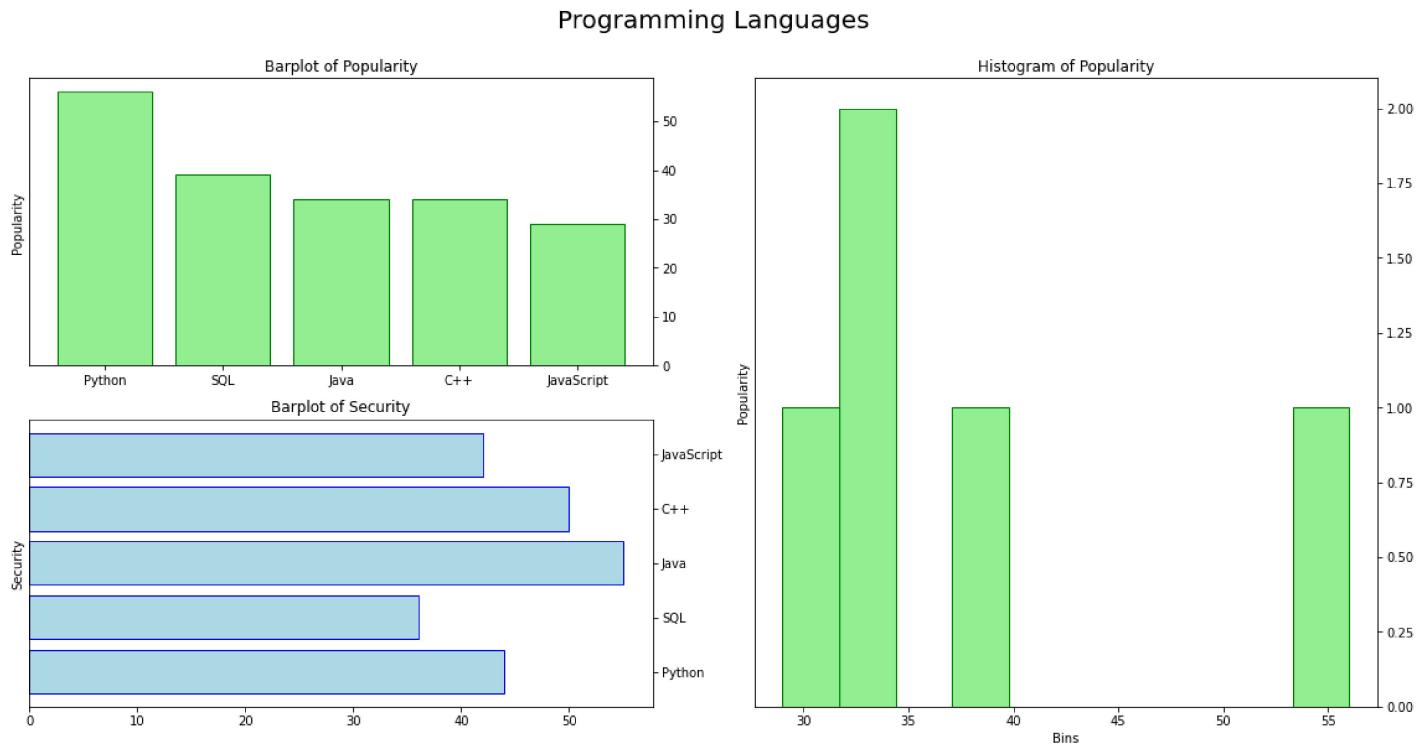
In []:

```
gridsize = (2, 4)
fig = plt.figure(figsize=(16, 8))
ax1 = plt.subplot2grid(gridsize, (0, 0), colspan=2, rowspan=1)
ax2 = plt.subplot2grid(gridsize, (1, 0), colspan=2, rowspan=1)
ax3 = plt.subplot2grid(gridsize, (0, 2), colspan=2, rowspan=2)
# customizing axis 3
ax1.bar(Languages,Popularity,color= 'lightgreen', edgecolor = 'g')
ax1.set_title('Barplot of Popularity')
ax1.set_ylabel('Popularity')
ax1.yaxis.tick_right()
# customizing axis 2
ax2.barh(Languages,Security,color= 'lightblue', edgecolor = 'b')
```

```

ax2.set_title('Barplot of Security')
ax2.set_ylabel('Security')
ax2.yaxis.tick_right()
# customizing axis 3
ax3.hist(Popularity,color= 'lightgreen', edgecolor = 'g')
ax3.set_title('Histogram of Popularity')
ax3.set_xlabel('Bins')
ax3.set_ylabel('Popularity')
ax3.yaxis.tick_right()
# super title of figure
fig.suptitle('Programming Languages', y = 1.05, fontsize=20)
# clean up whitespace padding
fig.tight_layout()

```



6. Plot Histogram, We have a sample data of Students marks of various Students, we will try to plot number of Students by marks range and try to figure out how many Students are average, below-average and Excellent.

Marks = [61,86,42,46,73,95,65,78,53,92,55,69,70,49,72,86,64]

Histogram showing Below Average, Average and Excellent distribution

40-60: Below Average

60-80: Average

80-100: Excellent

In []:

```

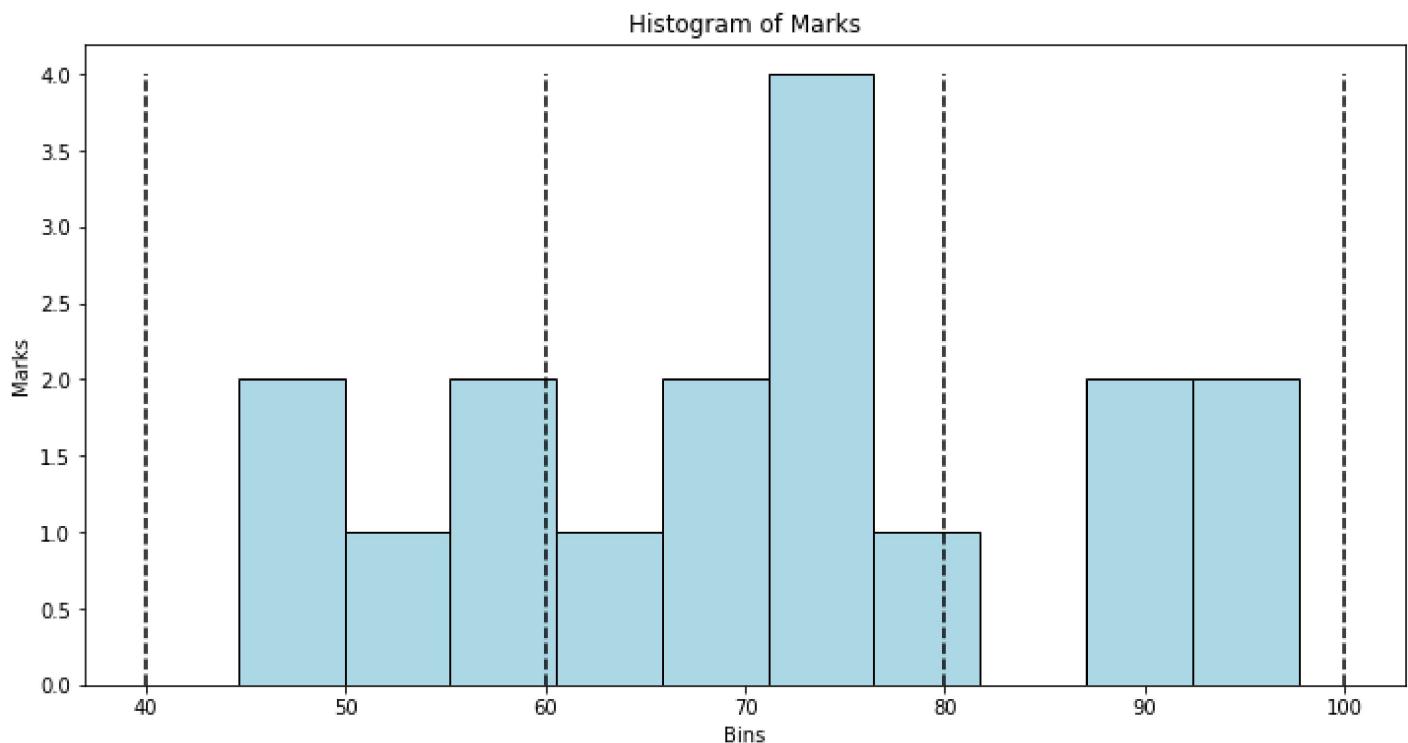
Marks = np.array([61,86,42,46,73,95,65,78,53,92,55,69,70,49,72,86,64])
below_average = Marks[np.logical_and(Marks >= 40,Marks < 60)]

```

```
average = Marks[np.logical_and(Marks >= 60, Marks < 80)]
excellent = Marks[np.logical_and(Marks >= 80, Marks < 100)]
```

In []:

```
plt.figure(figsize=(12, 6))
plt.hist(Marks, align='right', color='lightblue', edgecolor='black')
plt.xlabel('Bins')
plt.ylabel('Marks')
plt.title('Histogram of Marks')
plt.vlines([40, 60, 80, 100], ymin=0, ymax=4, colors='k', linestyles='dashed')
plt.show()
```



In []:

```
print(f'No of Below Average students : {below_average.size}')
print(f'No of Average students : {average.size}')
print(f'No of Excellent students : {excellent.size}')
```

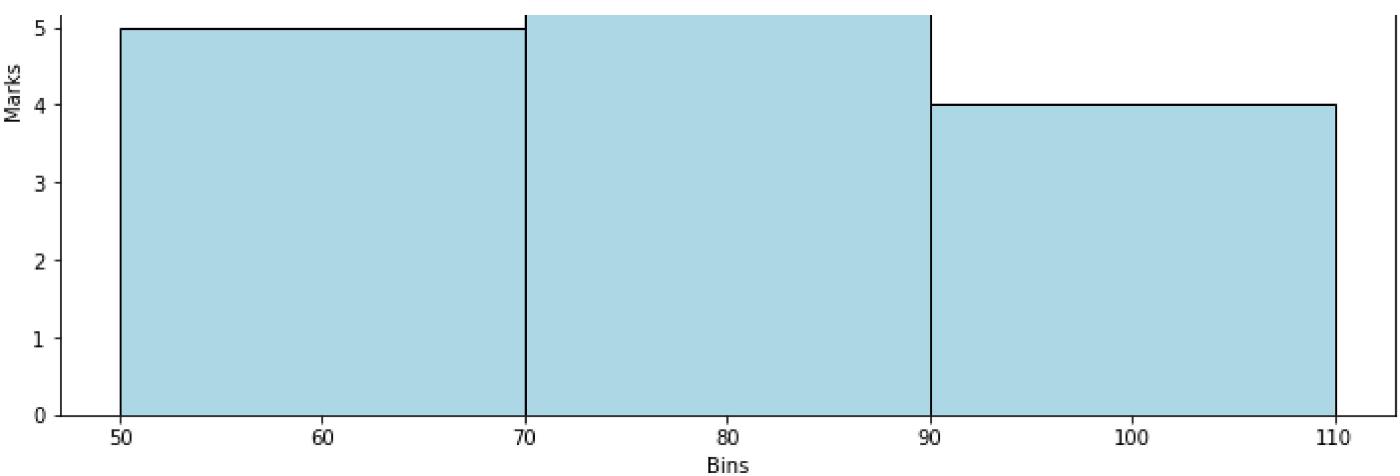
No of Below Average students : 5
No of Average students : 8
No of Excellent students : 4

In []:

```
plt.figure(figsize=(12, 6))
plt.hist(Marks, bins = [40, 60, 80, 100], align='right', color='lightblue',
edgecolor='black')
plt.xlabel('Bins')
plt.ylabel('Marks')
plt.title('Histogram of Marks')
plt.show()
```

Histogram of Marks





7. Titanic Data Set Download Data

Load the data file

(i) Create a pie chart presenting the male/female proportion

(ii) Create a scatterplot with the Fare paid and the Age, differ the plot color by gender

In []:

```
import seaborn as sns
import pandas as pd
```

In []:

```
df = sns.load_dataset('titanic')
```

In []:

```
df.head()
```

Out[]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton

◀

▶

In []:

```
df.columns
```

Out[]:

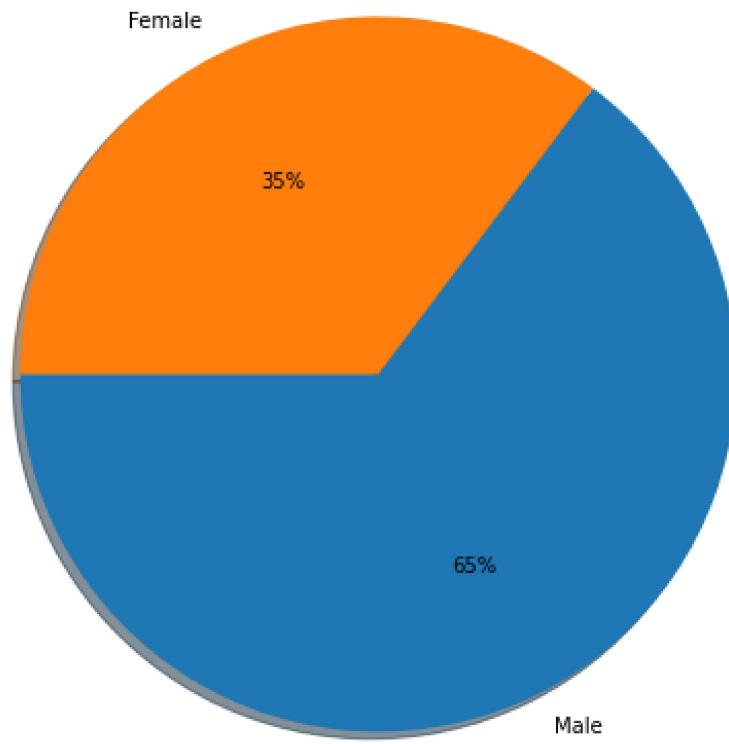
```
Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare',
       'embarked', 'class', 'who', 'adult_male', 'deck', 'embark_town',
       'alive', 'alone'],
      dtype='object')
```

In []:

```
plt.figure(figsize = (8, 8))
df.sex.value_counts().plot.pie(autopct='%.0f%%', labels=['Male', 'Female'], shadow = True,
startangle = 180)
```

Out[]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f937065c690>
```



In []:

```
df_male = df[df['sex'] == 'male']
df_male.sex.value_counts()
```

Out[]:

```
male    577
Name: sex, dtype: int64
```

In []:

```
df_female = df[df['sex'] == 'female']
df_female.sex.value_counts()
```

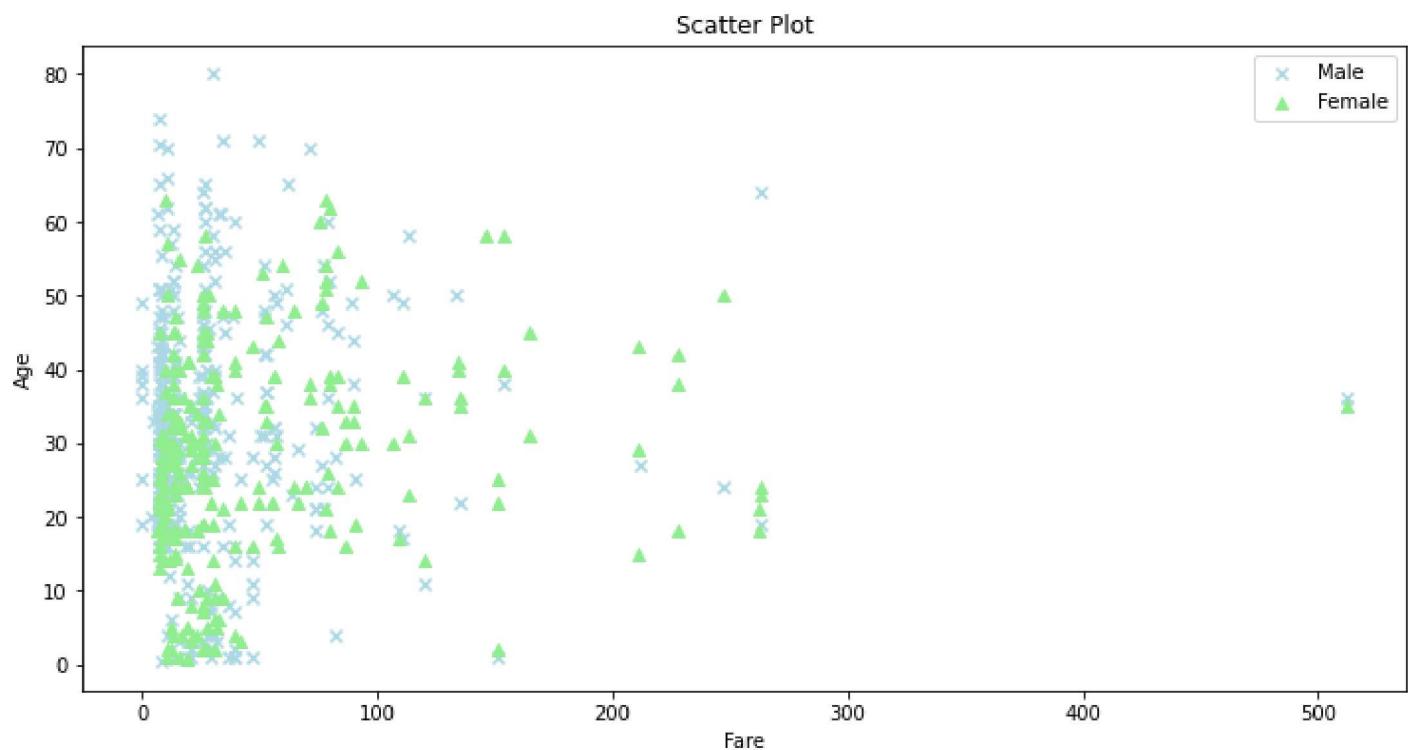
Out[]:

```
female    314
Name: sex, dtype: int64
```

In []:

```
plt.figure(figsize=(12,6))
plt.scatter(df_male.fare,df_male.age,
label='Male',color='lightblue',marker='x')
plt.scatter(df_female.fare,df_female.age,
label='Female',color='lightgreen',marker='^')
plt.xlabel('Fare')
plt.ylabel('Age')
```

```
plt.title('Scatter Plot')
plt.legend()
plt.show()
```



In []: