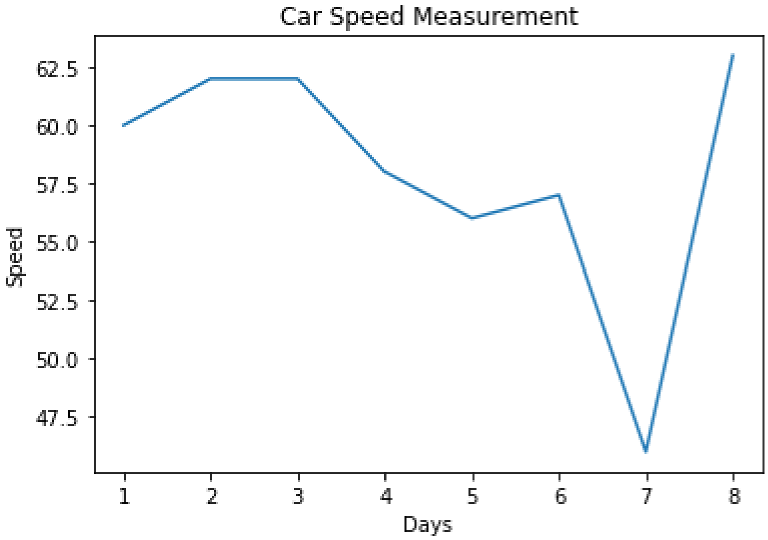
1. Load the necessary package for plotting using pyplot from matplotlib. Example - Days(x-axis) represents 8 days and Speed represents a car’s speed. Plot a Basic line plot between days and car speed, put x axis label as days and y axis label as car speed and put title Car Speed Measurement.

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

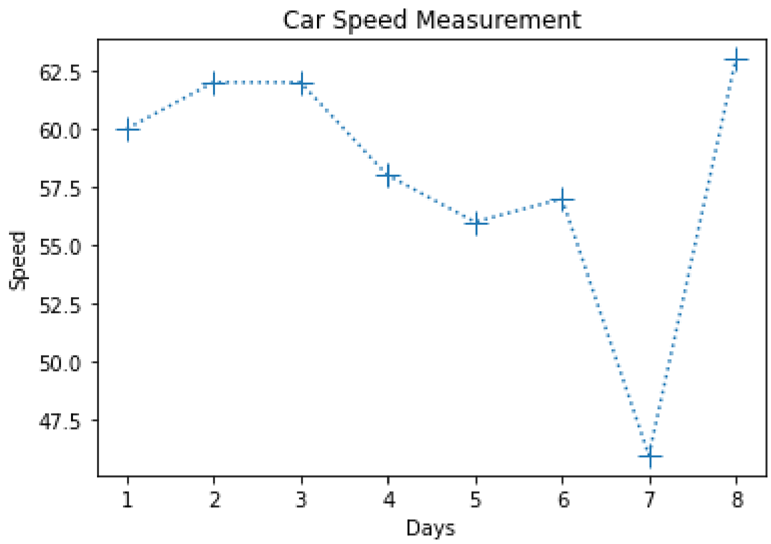
```
import matplotlib.pyplot as plt
```

```
Days=[1,2,3,4,5,6,7,8]
Speed=[60,62,62,58,56,57,46,63]
plt.plot(Days, Speed)
plt.xlabel("Days")
plt.ylabel("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.

```
plt.plot(Days, Speed, linestyle="dotted", marker="+", ms=12, mfc="red")
plt.xlabel("Days")
plt.ylabel("Speed")
plt.title("Car Speed Measurement")
plt.show()
```

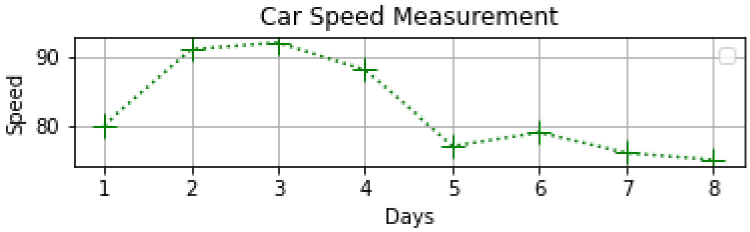


3. Plot Axes Labels, Chart title, Legend, Grid in Car minimum, Maximum and average speed in 8 days.

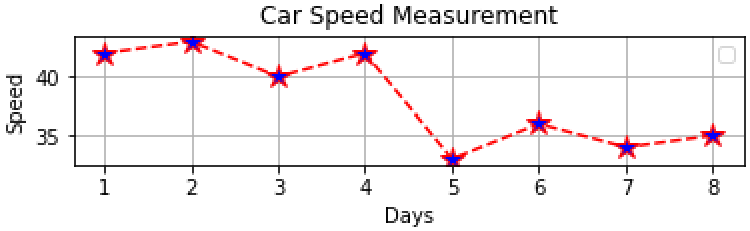
```
days=[1,2,3,4,5,6,7,8]
max_speed=[80,91,92,88,77,79,76,75]
min_speed=[42,43,40,42,33,36,34,35]
avg_speed=[46,58,57,56,40,42,41,36]
```

```
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]

plt.subplot(3,1,1)
plt.plot(Days, max_speed, color="green", linestyle="dotted", marker="+", ms=12, mfc="red")
plt.xlabel("Days")
plt.ylabel("Speed")
plt.title("Car Speed Measurement")
plt.legend()
plt.grid()
plt.show()

plt.subplot(3,1,2)
plt.plot(Days, min_speed, color="red", linestyle="dashed", marker="*", ms=12, mfc="blue")
plt.xlabel("Days")
plt.ylabel("Speed")
plt.title("Car Speed Measurement")
plt.legend()
plt.grid()
plt.show()

plt.subplot(3,1,3)
plt.subplot(3,1,3)
plt.plot(Days, avg_speed, color="blue", linestyle="dashdot", marker="o", ms=12, mfc="orange")
plt.xlabel("Days")
plt.ylabel("Speed")
plt.title("Car Speed Measurement")
plt.legend()
plt.grid()
plt.show()
```

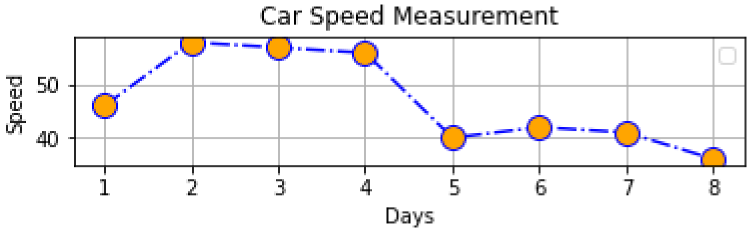
```
No handles with labels found to put in legend.
```



```
No handles with labels found to put in legend.
```

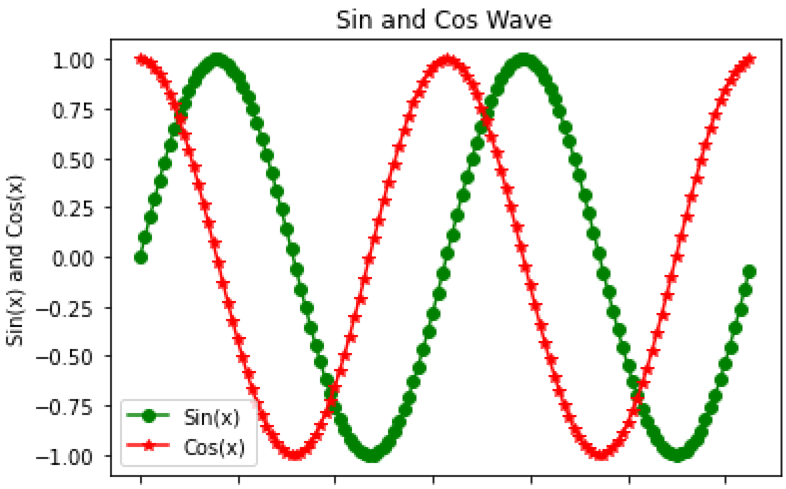


```
No handles with labels found to put in legend.
```



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

```
import numpy as np
x=np.arange(0,4*np.pi,0.1)
y=np.sin(x)
z=np.cos(x)
plt.plot(x,y,color='green',marker='o')
plt.plot(x,z,color='red',marker='*')
plt.xlabel("Sin and Cos Value from 0 to 4pi")
plt.ylabel("Sin(x) and Cos(x)")
plt.title("Sin and Cos Wave")
plt.legend(["Sin(x)","Cos(x)"])
plt.show()
```

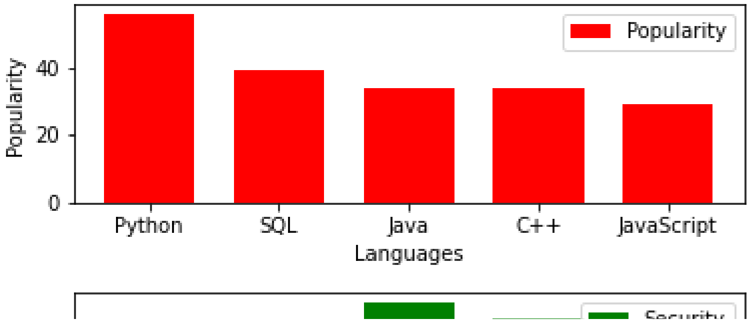


5. Plot Simple bar chart showing popularity of Programming Languages.

```
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.

```
Languages=["Python", "SQL", "Java", "C++", "JavaScript"]
Popularity=[56,39,34,34,29]
Security=[44,36,55,50,42]

plt.subplot(2,1,1)
plt.bar(Languages, Popularity, width=0.7, color="red", align="center")
plt.xlabel("Languages")
plt.ylabel("Popularity")
plt.legend(["Popularity"])
plt.show()

plt.subplot(2,1,2)
plt.bar(Languages, Security, width=0.7, color="green", align="center")
plt.xlabel("Languages")
plt.ylabel("Security")
plt.legend(["Security"], loc="upper right")
plt.show()
```



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]
```

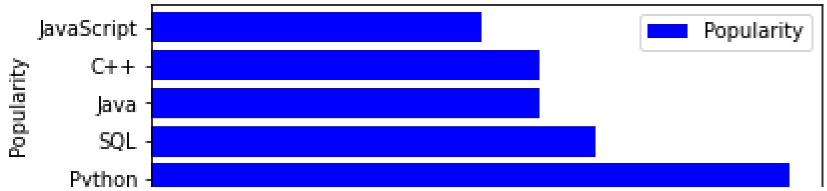
Histogram showing Below Average, Average and Execellent distribution

40-60: Below Average

60-80: Average

80-100: Excellent

```
# Creation of Horizontal Bar
plt.subplot(2,1,1)
plt.barh(Languages, Popularity, color="blue", align="center")
plt.xlabel("Languages")
plt.ylabel("Popularity")
plt.legend(["Popularity"])
plt.show()

plt.subplot(2,1,2)
plt.barh(Languages, Security, color="orange", align="center")
plt.xlabel("Languages")
plt.ylabel("Security")
plt.legend(["Security"], loc="lower right")
plt.show()
```



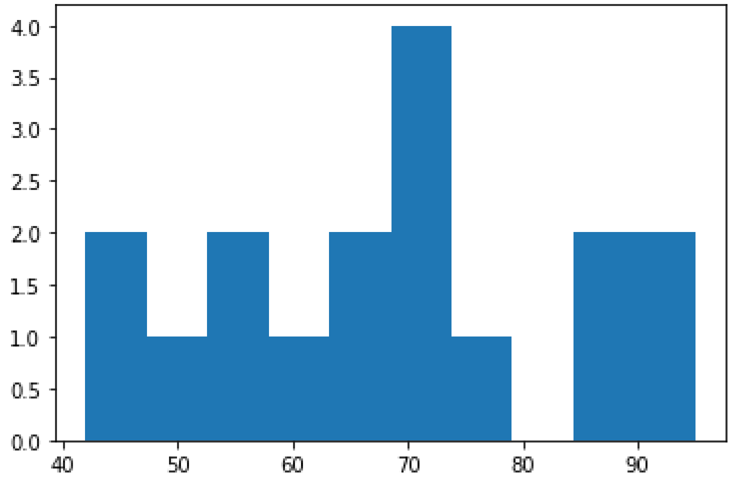
7. Titanic Data Set Download Data

   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

```
Marks = np.array([61,86,42,46,73,95,65,78,53,92,55,69,70,49,72,86,64])
below_avg = Marks[np.logical_and(Marks >= 40, Marks < 60)]
avg_marks = Marks[np.logical_and(Marks >= 60, Marks < 80)]
exe_marks = Marks[np.logical_and(Marks >= 80, Marks <= 100)]
plt.hist(Marks)
plt.show()
print('Below Average Students Are : ',below_avg.size)
print('Average Students Are : ',avg_marks.size)
print('Execelent Students Are : ',exe_marks.size)
```



```
Below Average Students Are :  5
Average Students Are :  8
Execelent Students Are :  4
```

```
import pandas as pd
df = pd.read_csv('/content/titanic_original.csv')
df.head()
```

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cab
0	1.0	1.0	Allen, Miss. Elisabeth Walton	female	29.0000	0.0	0.0	24160	211.3375	B
1	1.0	1.0	Allison, Master. Hudson Trevor	male	0.9167	1.0	2.0	113781	151.5500	C C

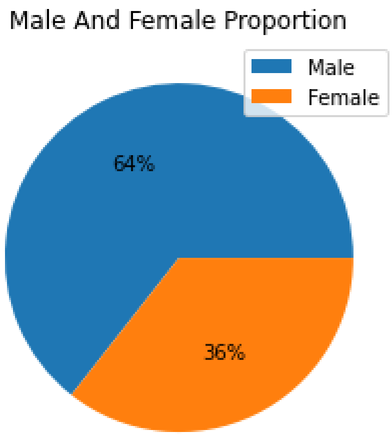
```
df['sex']
```

```
0       female
1         male
2       female
3         male
4       female
         ...
1305    female
1306      male
1307      male
1308      male
1309       NaN
Name: sex, Length: 1310, dtype: object
```

```
x = df.sex.value_counts()
x
```

```
male      843
female    466
Name: sex, dtype: int64
```

```
plt.pie(x, autopct='%.0f%%')
plt.legend(['Male','Female'])
plt.title('Male And Female Proportion')
plt.show()
```



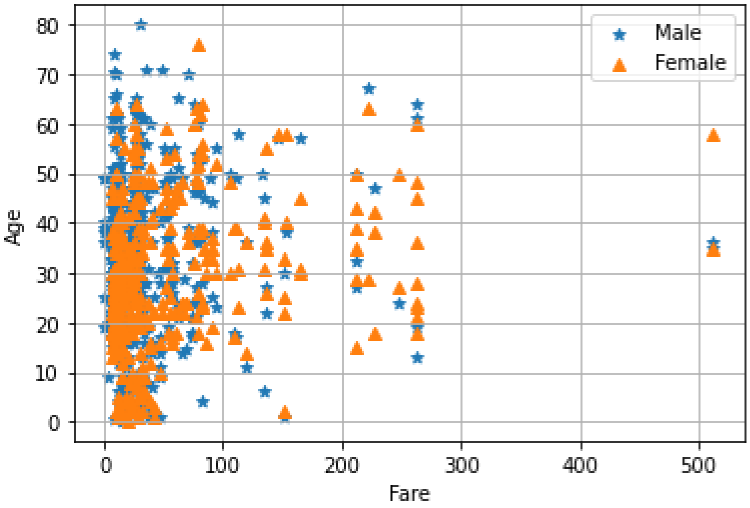
```
male = df[df['sex']=='male']
male.sex.value_counts()
```

```
male    843
Name: sex, dtype: int64
```

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

```
female    466
Name: sex, dtype: int64
```

```
plt.scatter(male.fare, male.age, label = 'Male', marker = '*')
plt.scatter(female.fare, female.age, label = 'Female', marker = '^')
plt.legend()
plt.xlabel('Fare')
plt.ylabel('Age')
plt.grid()
plt.show()
```



● ×