

ASSIGNMENT/TASK6

Shweta Jha(GO_STP_12574)

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]
```

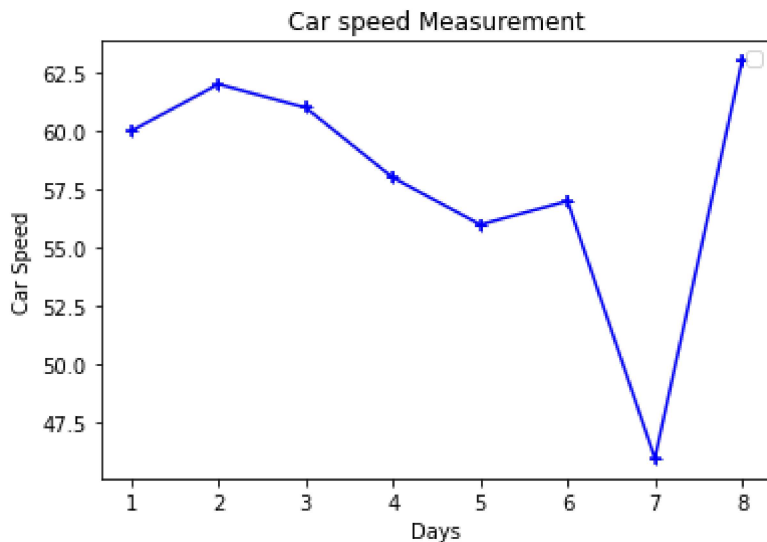
```
Speed=[60,62,61,58,56,57,46,63]
```

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

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

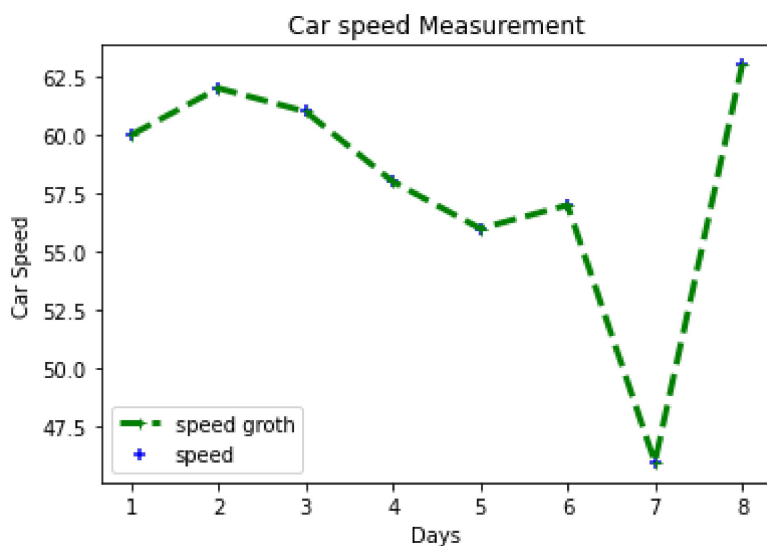
```
plt.xlabel('Days')  
plt.ylabel('Car Speed')  
plt.plot(Days,Speed,color='blue',marker='+')  
plt.scatter(Days,Speed,color='blue',marker='+')  
plt.title("Car speed Measurement")  
plt.legend()  
plt.show()
```

No handles with labels found to put in legend.



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.xlabel('Days')
plt.ylabel('Car Speed')
plt.plot(Days,Speed,"--",color='green',label="speed groth",marker='+',linewidth=3,markersize=
plt.scatter(Days,Speed,color='blue',marker='+',label="speed")
plt.title("Car speed Measurement")
plt.legend()
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.xlabel('Days')
```

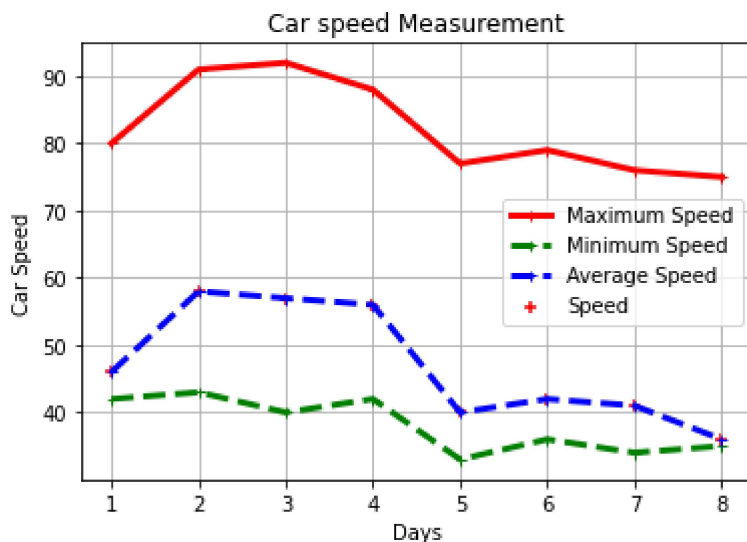
```
plt.ylabel('Car Speed')
```

```
plt.plot(days,max_speed,color='red',marker='+', label='Maximum Speed',linewidth=3,markersize=
```

```
plt.plot(days,min_speed,"--",color='green',marker='+', label='Minimum Speed',linewidth=3,mark
```

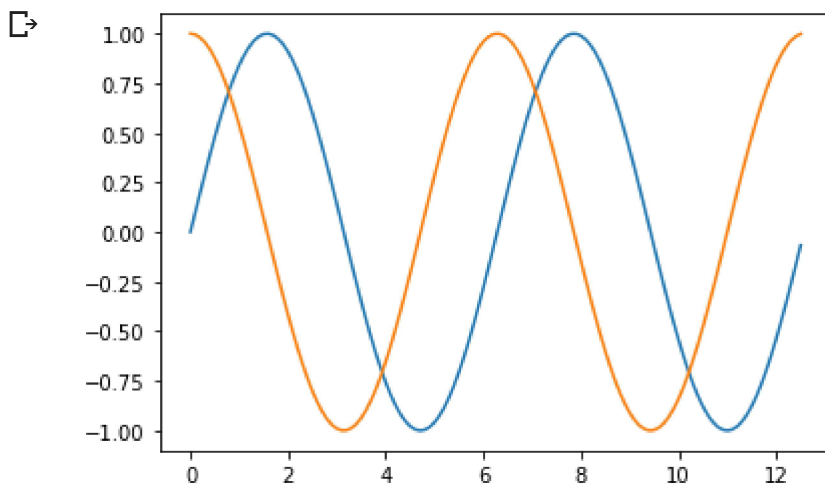
```
plt.plot(days,avg_speed,"--",color='blue',marker='+', label='Average Speed',linewidth=3,marke
```

```
plt.scatter(days,avg_speed,color='red',marker='+', label='Speed')
plt.title("Car speed Measurement")
plt.legend()
plt.grid()
plt.show()
```



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

```
x=np.arange(0,4*np.pi,0.1)
y=np.sin(x)
z=np.cos(x)
plt.plot(x,y,x,z)
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]
```

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

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

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

```
plt.bar(Languages, Popularity, color='b', label="Popularity")
```

```
plt.bar(Languages, Security, color='r', label="Security")
```

```
plt.legend()
```

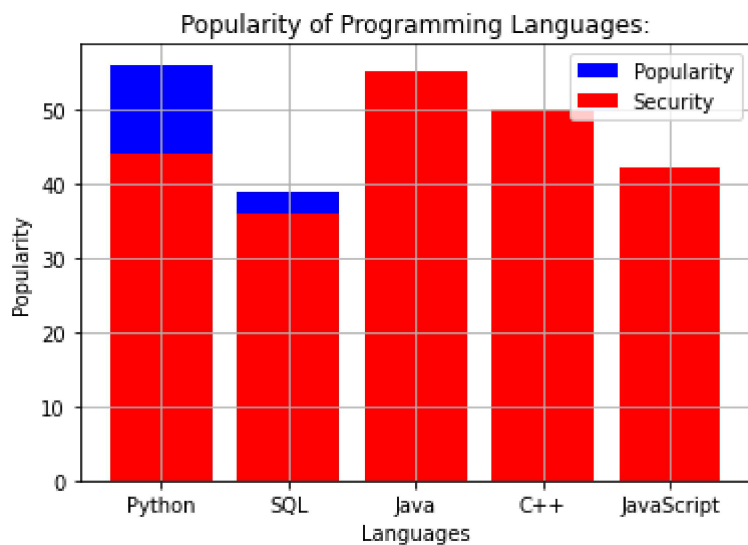
```
plt.xlabel('Languages')
```

```
plt.ylabel('Popularity')
```

```
plt.grid()
```

```
plt.title('Popularity of Programming Languages:')
```

```
plt.show()
```



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

```
plt.bar(Languages, Popularity, label="Popularity", color='b', width=.1)
```

```
plt.bar(Languages, Security, label="Security", color='r', width=.2)
```

```
plt.legend()
```

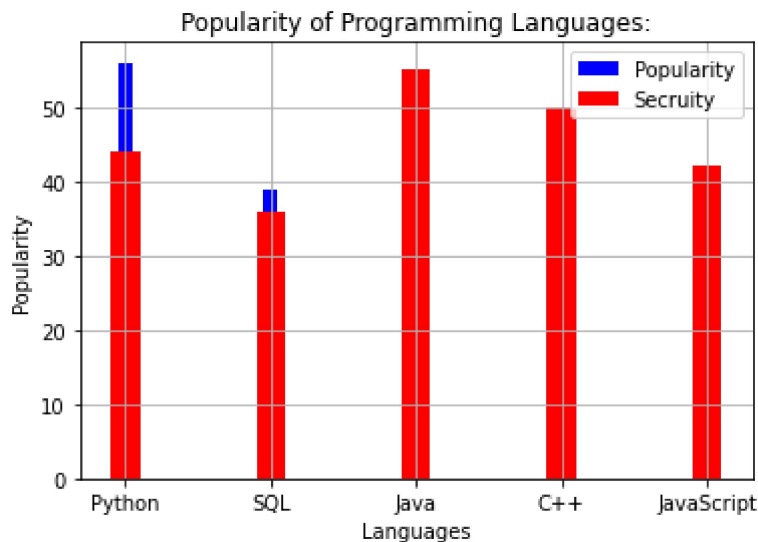
```
plt.xlabel('Languages')
```

```
plt.ylabel('Popularity')
```

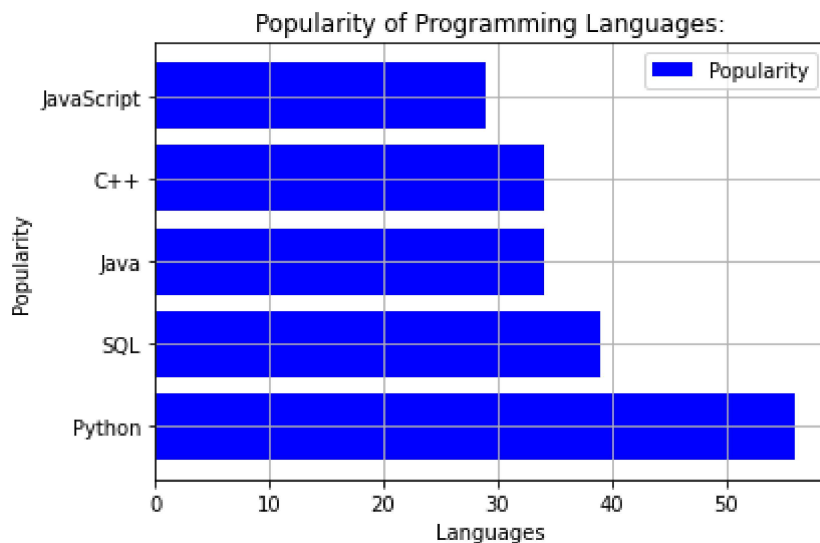
```
plt.grid()
```

```
plt.title('Popularity of Programming Languages:')
```

```
plt.show()
```



```
plt.barh(Languages,Popularity,label="Popularity",color='b')
plt.legend()
plt.xlabel('Languages')
plt.ylabel('Popularity')
plt.grid()
plt.title('Popularity of Programming Languages:')
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.

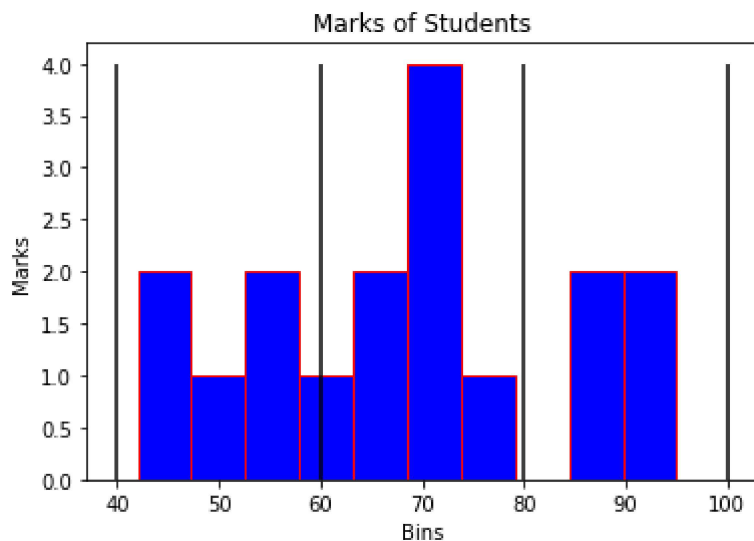
Marks = [61,86,42,46,73,95,65,78,53,92,55,69,70,49,72,86,64]

```
Marks = np.array([ 61,86,42,46,73,95,65,78,53,92,55,69,70,49,72,86,64])
plt.hist(Marks,color='b',edgecolor='r')
plt.vlines([40,60,80,100],ymin=0,ymax=4)
plt.xlabel('Marks')
```

```

plt.xlabel('Bins')
plt.ylabel('Marks')
plt.title("Marks of Students")
plt.show()
print("1,Average:",len(Marks[np.logical_and(Marks>=60,Marks<80)]))
print("2,Below Average:",len(Marks[np.logical_and(Marks>=40,Marks<60)]))
print("3,Excellent:",len(Marks[np.logical_and(Marks>=80,Marks<100)]))

```



```

1,Average: 8
2,Below Average: 5
3,Excellent: 4

```

Histogram showing Below Average, Average and Excellent distribution

40-60: Below Average

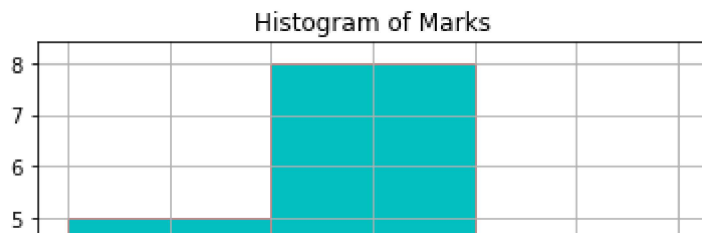
60-80: Average

80-100: Excellent

```

bins=[40,60,80,100]
plt.hist(Marks,bins,color='c',edgecolor='r')
plt.xlabel('Bins')
plt.ylabel('Marks')
plt.title("Histogram of Marks")
plt.grid()
plt.show()

```



7. Titanic Data Set Download Data

Load the data file

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



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

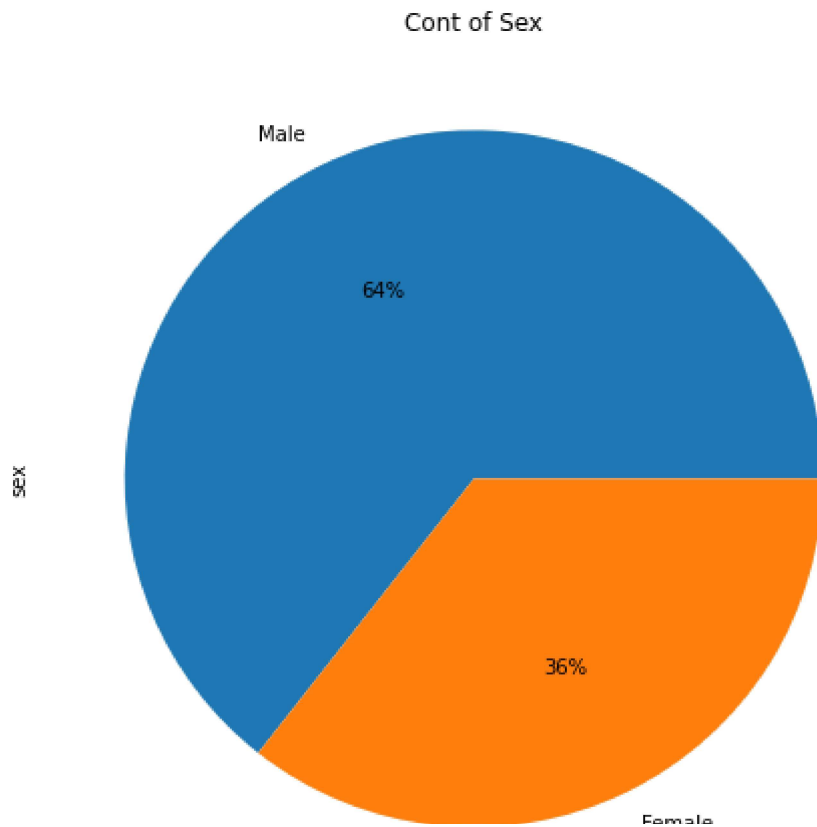
```
df =pd.read_csv('/content/titanic_original.csv')
df.head(5)
```

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	em
0	1.0	1.0	Allen, Miss. Elisabeth Walton	female	29.0000	0.0	0.0	24160	211.3375	B5	
1	1.0	1.0	Allison, Master. Hudson Trevor	male	0.9167	1.0	2.0	113781	151.5500	C22 C26	
2	1.0	0.0	Allison, Miss. Helen Loraine	female	2.0000	1.0	2.0	113781	151.5500	C22 C26	

```
df.columns
```

```
Index(['pclass', 'survived', 'name', 'sex', 'age', 'sibsp', 'parch', 'ticket',
      'fare', 'cabin', 'embarked', 'boat', 'body', 'home.dest'],
      dtype='object')
```

```
plt.figure(figsize=(8,8))
df.sex.value_counts().plot.pie(autopct='%0f%%',labels=['Male','Female'])
plt.title('Cont of Sex')
plt.show()
```



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

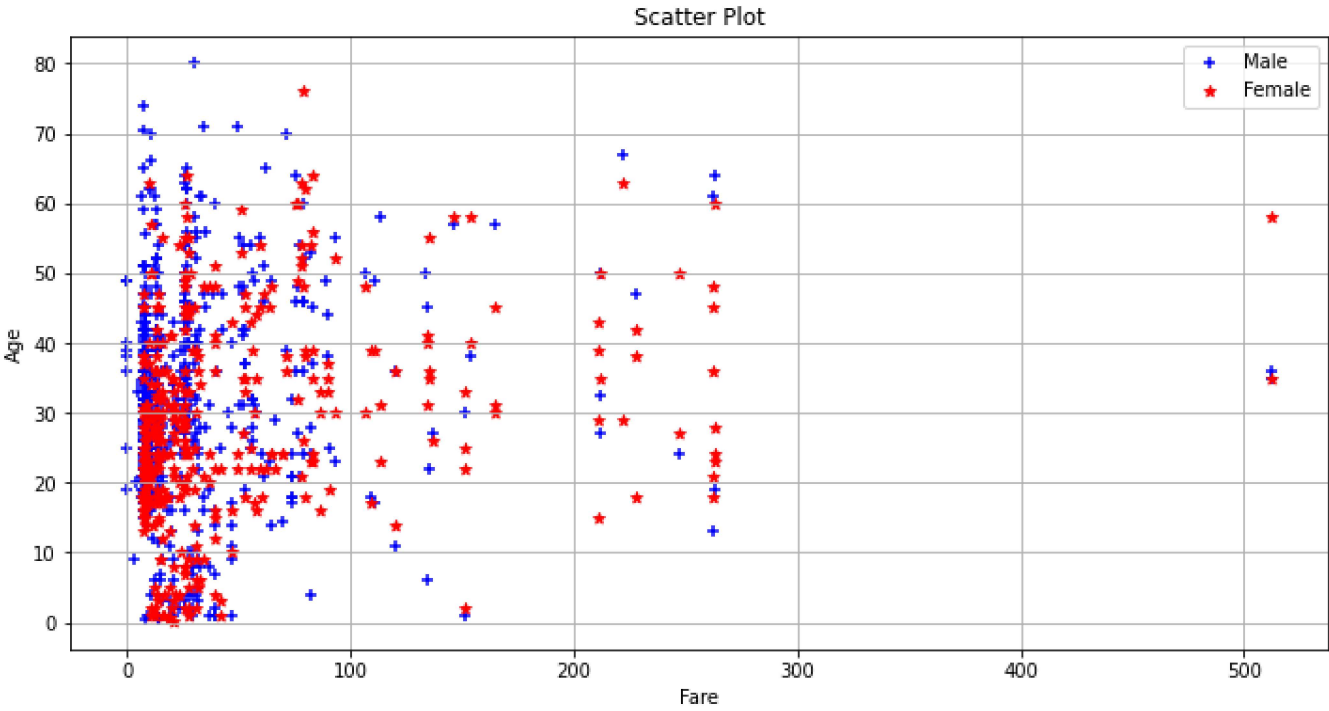
```
df_male=df[df['sex']=='male']
df_male.sex.value_counts()
```

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

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

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

```
plt.figure(figsize=(12,6))
plt.scatter(df_male.fare,df_male.age,label='Male',color='blue',marker='+')
plt.scatter(df_female.fare,df_female.age,label='Female',color='red',marker='*')
plt.xlabel("Fare")
plt.ylabel('Age')
plt.title('Scatter Plot')
plt.legend()
plt.grid()
plt.show()
```

✓ 0s completed at 2:32 AM

