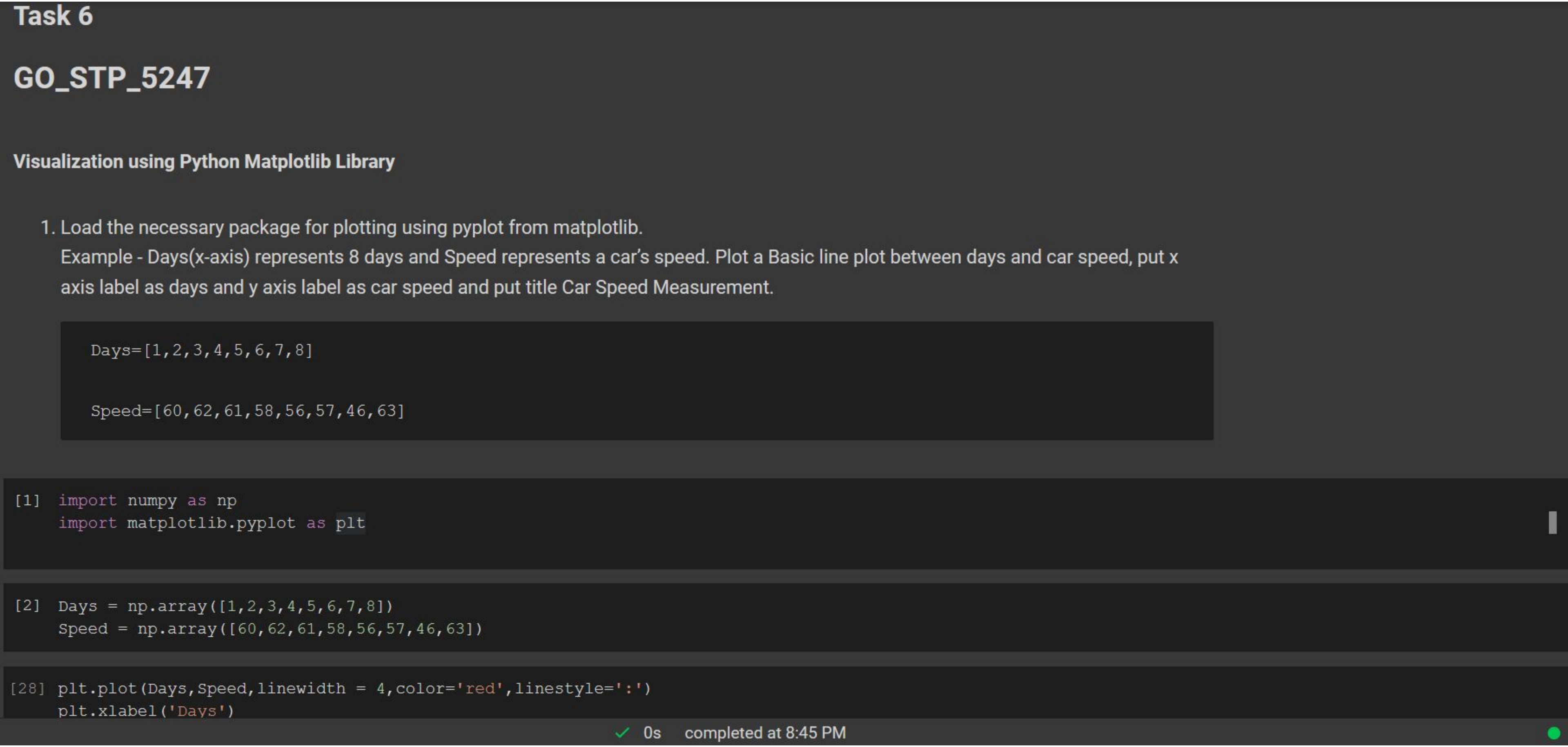


# Task 6

# GO_STP_5247

**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.

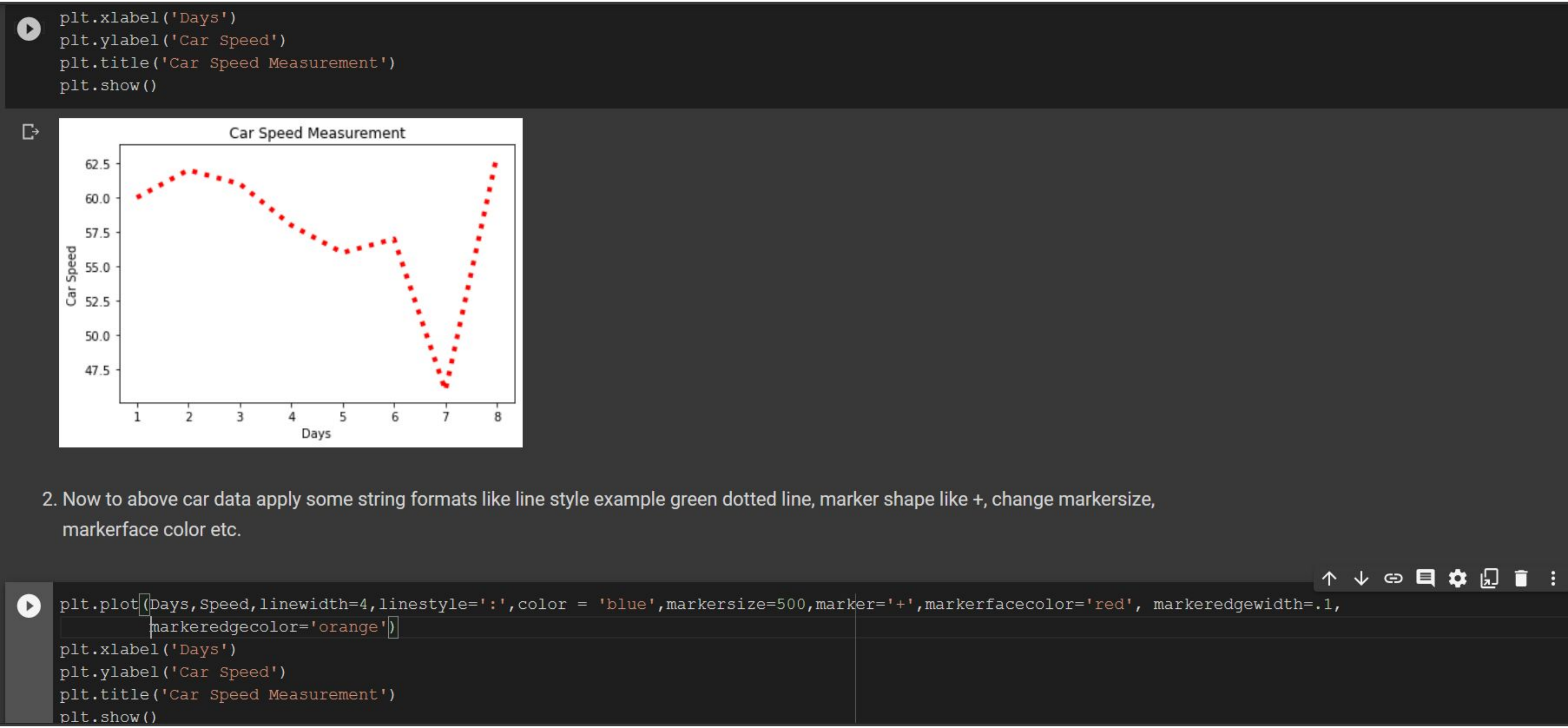
   ```
   Days=[1,2,3,4,5,6,7,8]


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

```
[1] import numpy as np
    import matplotlib.pyplot as plt
```

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

```
[28] plt.plot(Days,Speed,linewidth = 4,color='red',linestyle=':')
     plt.xlabel('Days')
```

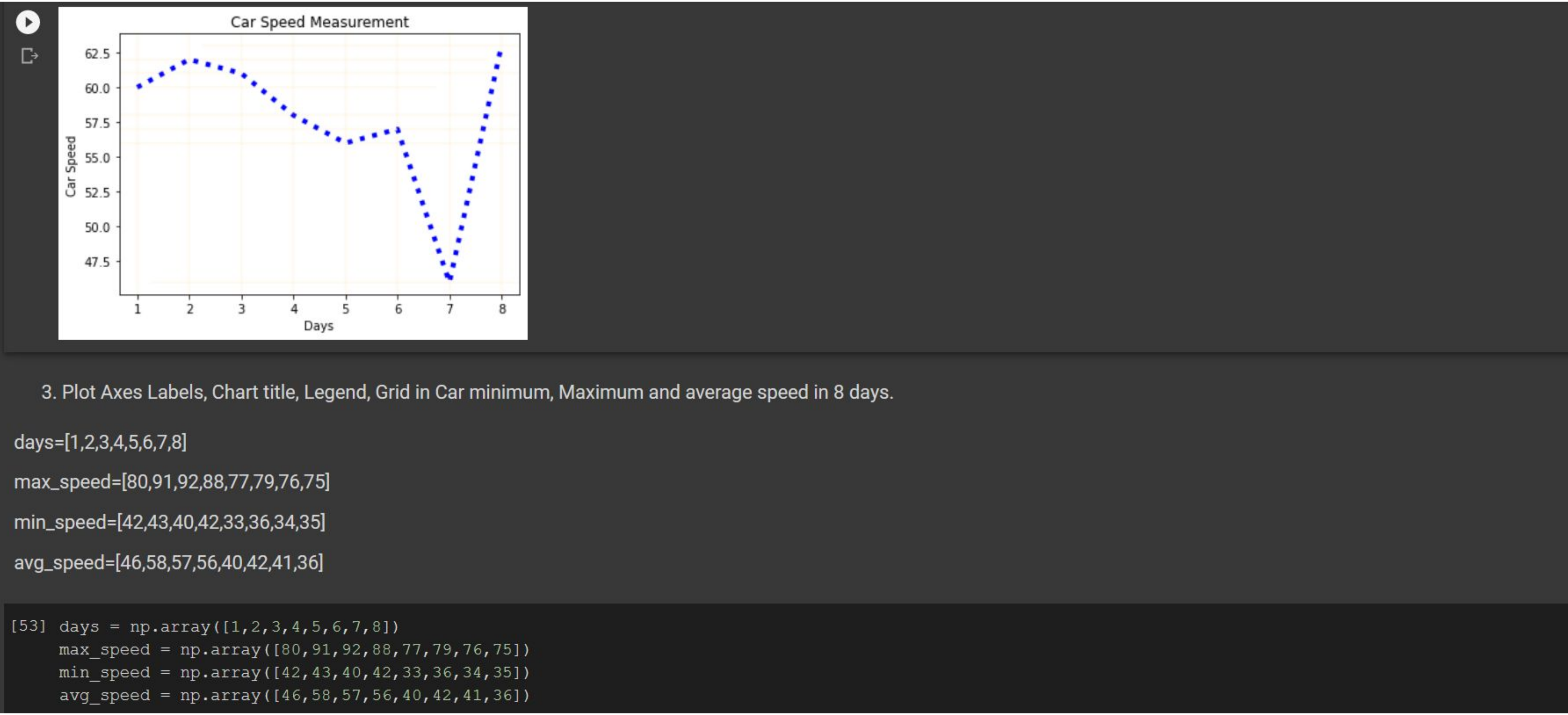
✓ 0s completed at 8:45 PM



```
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.

```
plt.plot(Days,Speed,linewidth=4,linestyle=':',color = 'blue',markersize=500,marker='+',markerfacecolor='red', markeredgewidth=.1,
         markeredgecolor='orange')
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.

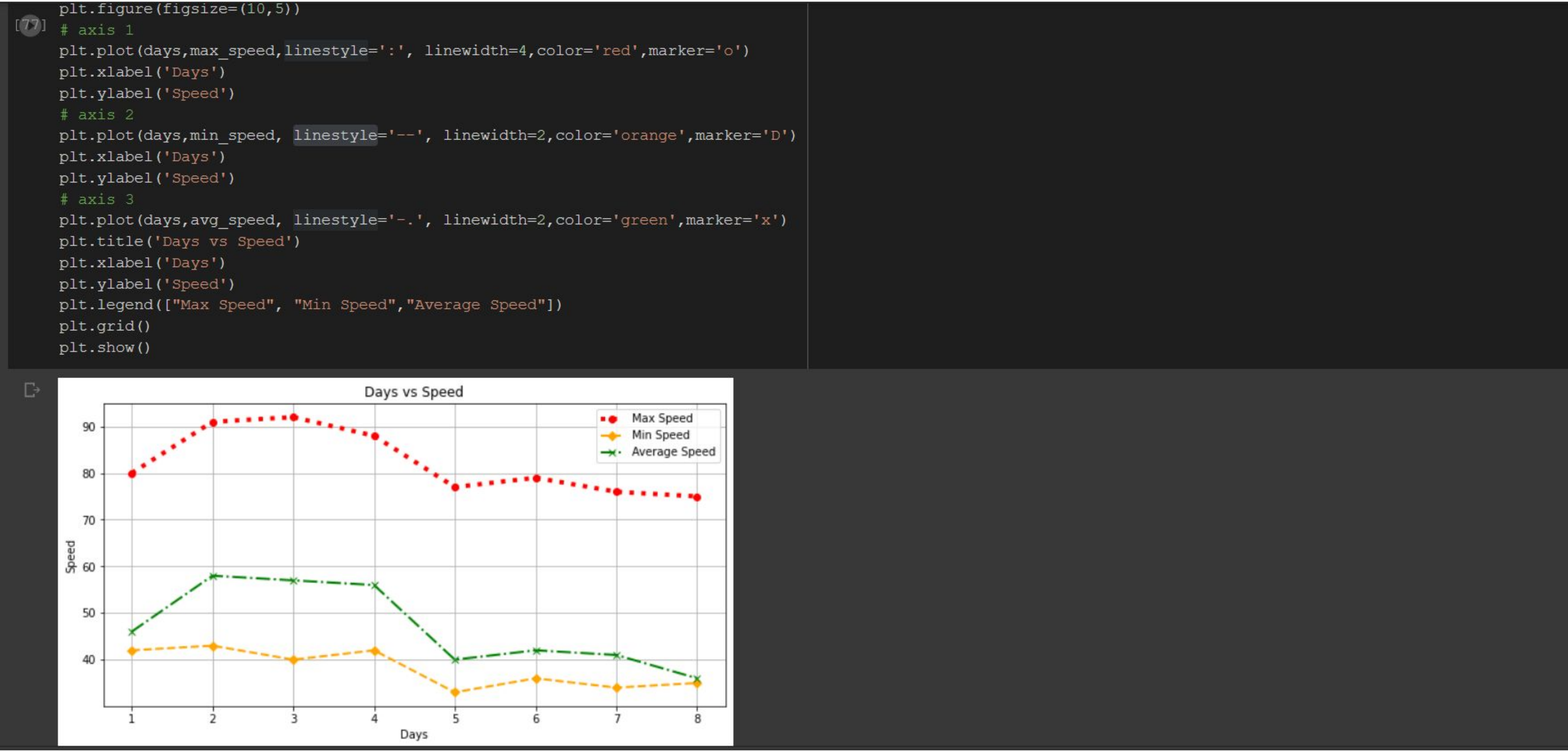
days=[1,2,3,4,5,6,7,8]

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

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

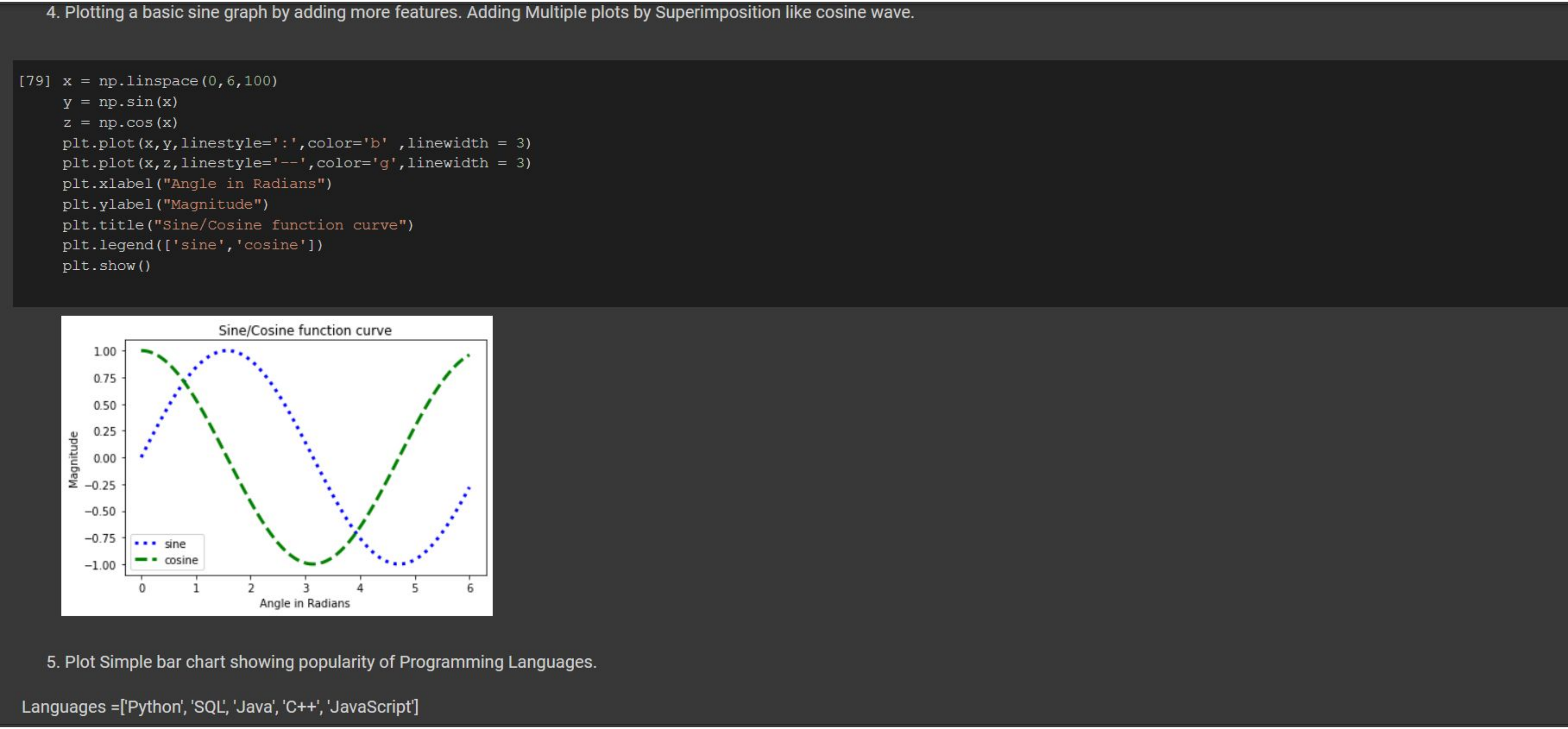
avg_speed=[46,58,57,56,40,42,41,36]

```
[53] 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])
```



[77]

```python
plt.figure(figsize=(10,5))
# axis 1
plt.plot(days,max_speed,linestyle=':', linewidth=4,color='red',marker='o')
plt.xlabel('Days')
plt.ylabel('Speed')
# axis 2
plt.plot(days,min_speed, linestyle='--', linewidth=2,color='orange',marker='D')
plt.xlabel('Days')
plt.ylabel('Speed')
# axis 3
plt.plot(days,avg_speed, linestyle='-.', linewidth=2,color='green',marker='x')
plt.title('Days vs Speed')
plt.xlabel('Days')
plt.ylabel('Speed')
plt.legend(["Max Speed", "Min Speed","Average Speed"])
plt.grid()
plt.show()
```

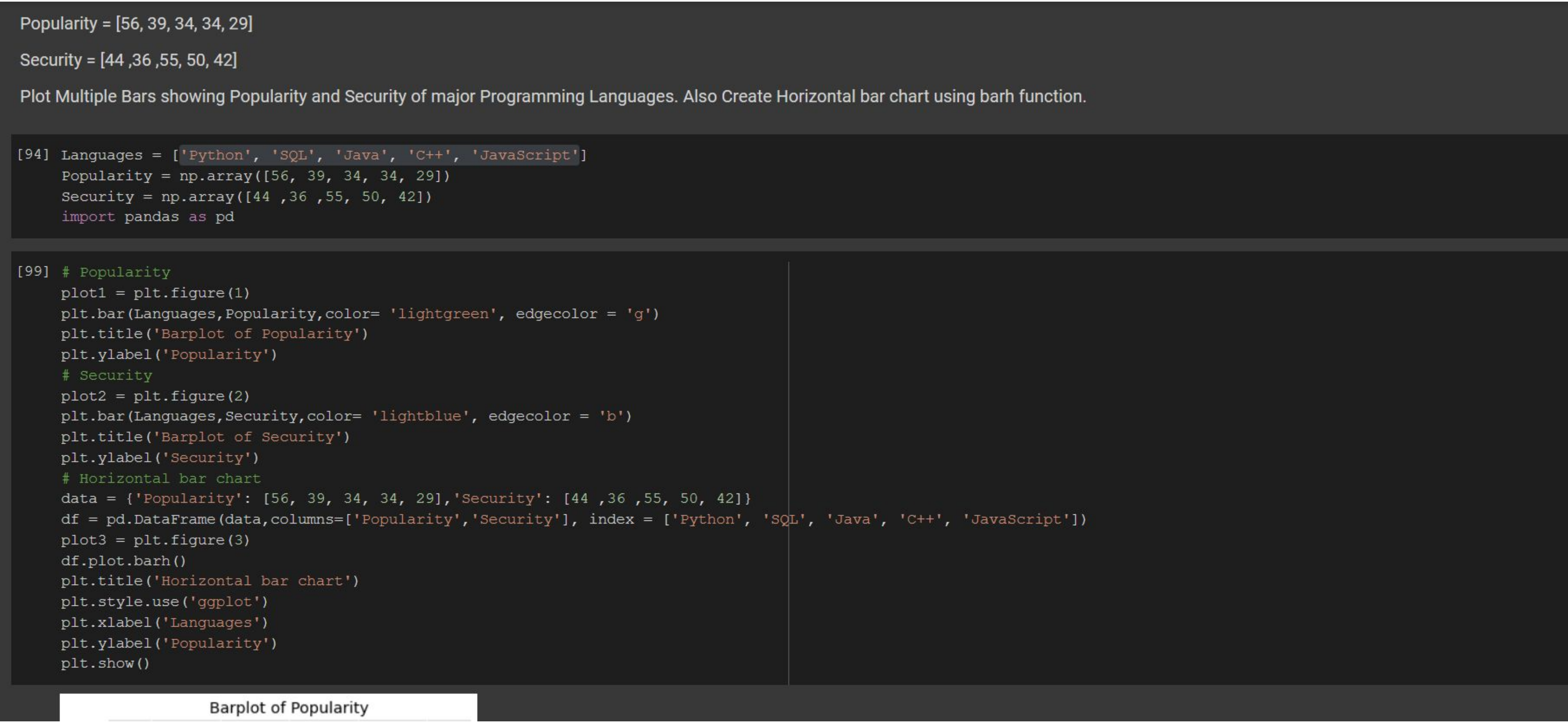


4. Plotting a basic sine graph by adding more features. Adding Multiple plots by Superimposition like cosine wave.

```
[79] x = np.linspace(0,6,100)
     y = np.sin(x)
     z = np.cos(x)
     plt.plot(x,y,linestyle=':',color='b' ,linewidth = 3)
     plt.plot(x,z,linestyle='--',color='g',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']



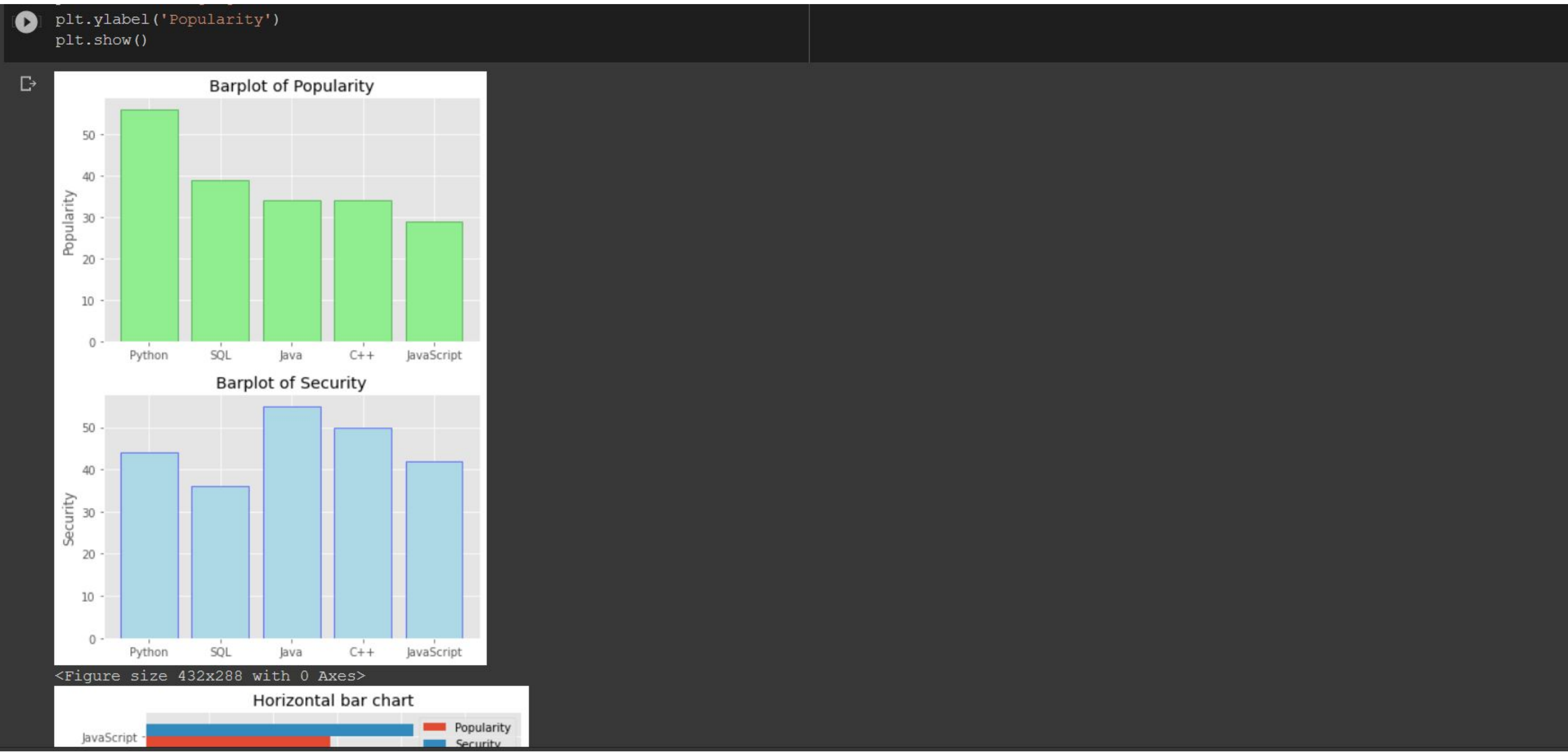
Popularity = [56, 39, 34, 34, 29]

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.

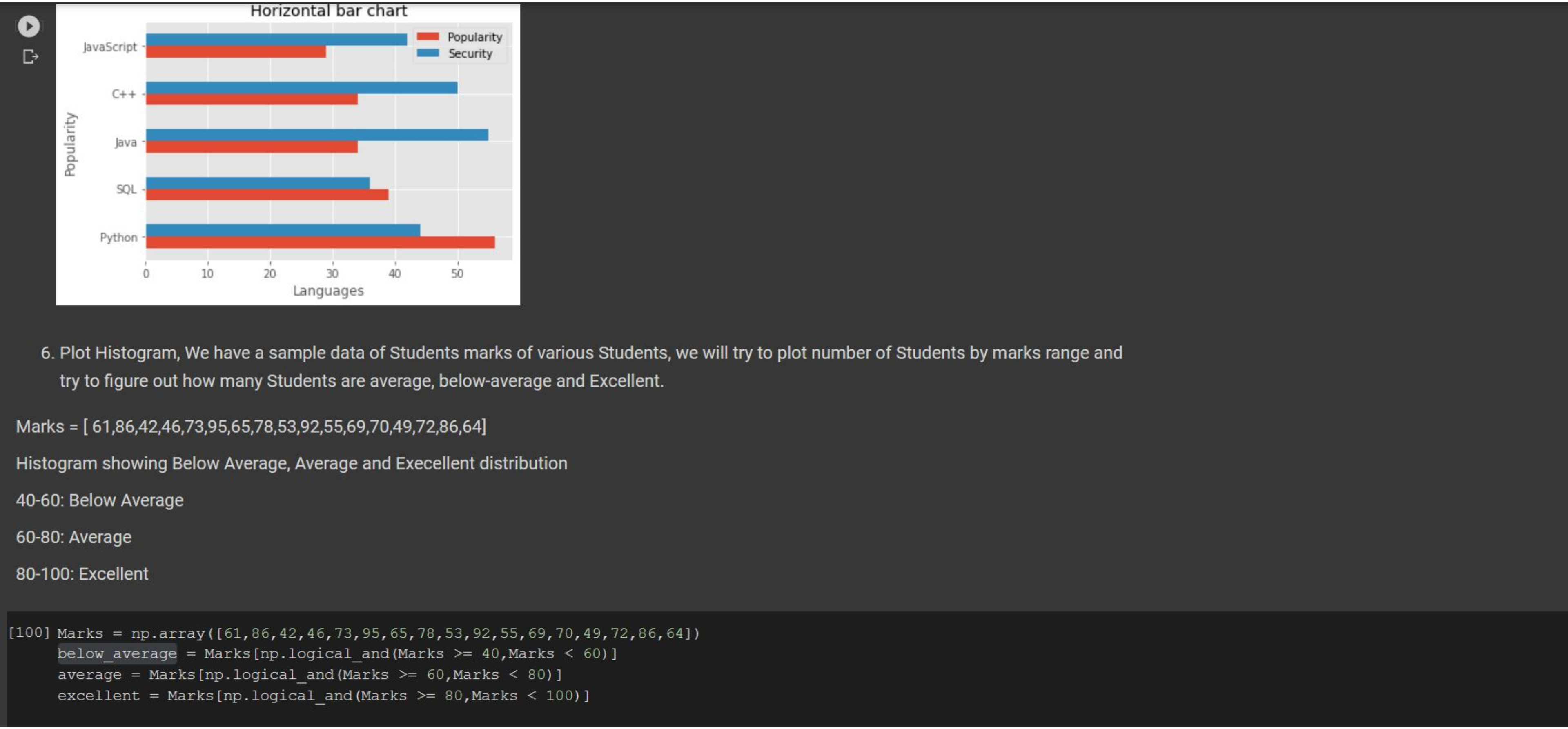
```
[94] Languages = ['Python', 'SQL', 'Java', 'C++', 'JavaScript']
     Popularity = np.array([56, 39, 34, 34, 29])
     Security = np.array([44 ,36 ,55, 50, 42])
     import pandas as pd
```

```
[99] # Popularity
     plot1 = plt.figure(1)
     plt.bar(Languages,Popularity,color= 'lightgreen', edgecolor = 'g')
     plt.title('Barplot of Popularity')
     plt.ylabel('Popularity')
     # Security
     plot2 = plt.figure(2)
     plt.bar(Languages,Security,color= 'lightblue', edgecolor = 'b')
     plt.title('Barplot of Security')
     plt.ylabel('Security')
     # Horizontal bar chart
     data = {'Popularity': [56, 39, 34, 34, 29],'Security': [44 ,36 ,55, 50, 42]}
     df = pd.DataFrame(data,columns=['Popularity','Security'], index = ['Python', 'SQL', 'Java', 'C++', 'JavaScript'])
     plot3 = plt.figure(3)
     df.plot.barh()
     plt.title('Horizontal bar chart')
     plt.style.use('ggplot')
     plt.xlabel('Languages')
     plt.ylabel('Popularity')
     plt.show()
```



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

```
<Figure size 432x288 with 0 Axes>
```



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]

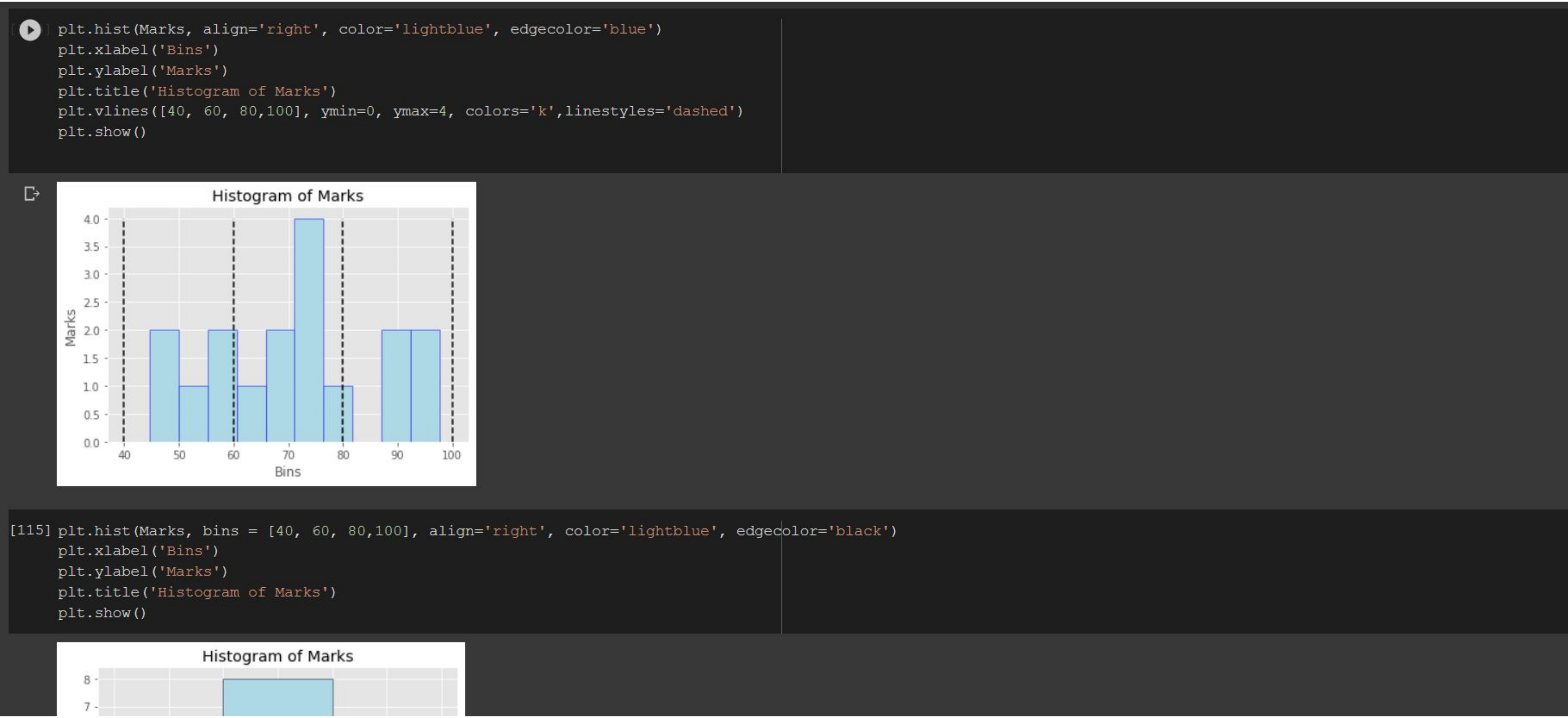
Histogram showing Below Average, Average and Exececllent distribution

40-60: Below Average

60-80: Average

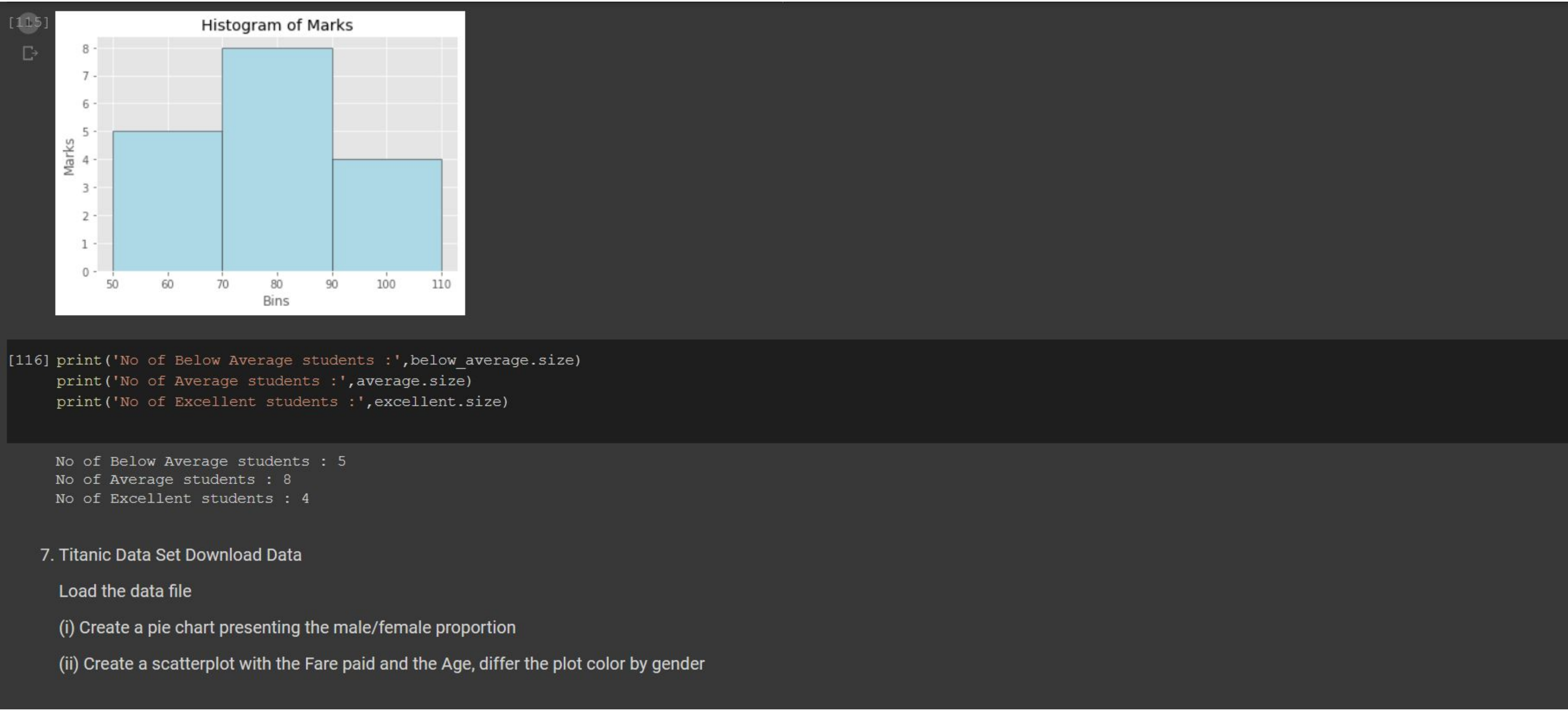
80-100: Excellent

```
[100] 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)]
```



```
plt.hist(Marks, align='right', color='lightblue', edgecolor='blue')
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()
```

```
[115] 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()
```



```
[116] print('No of Below Average students :',below_average.size)
      print('No of Average students :',average.size)
      print('No of Excellent students :',excellent.size)
```

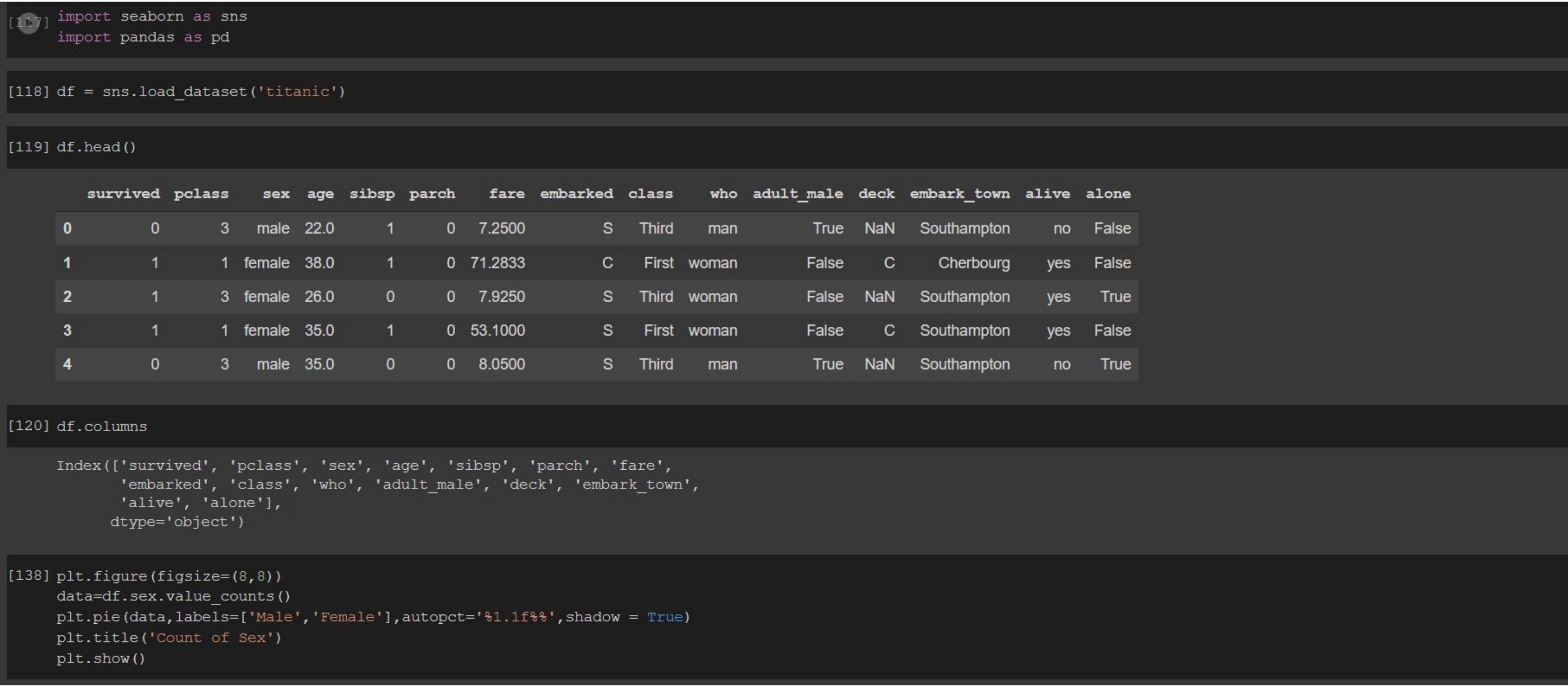
```
No of Below Average students : 5
No of Average students : 8
No of Excellent students : 4
```

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



```
[117] import seaborn as sns
      import pandas as pd
```

```
[118] df = sns.load_dataset('titanic')
```

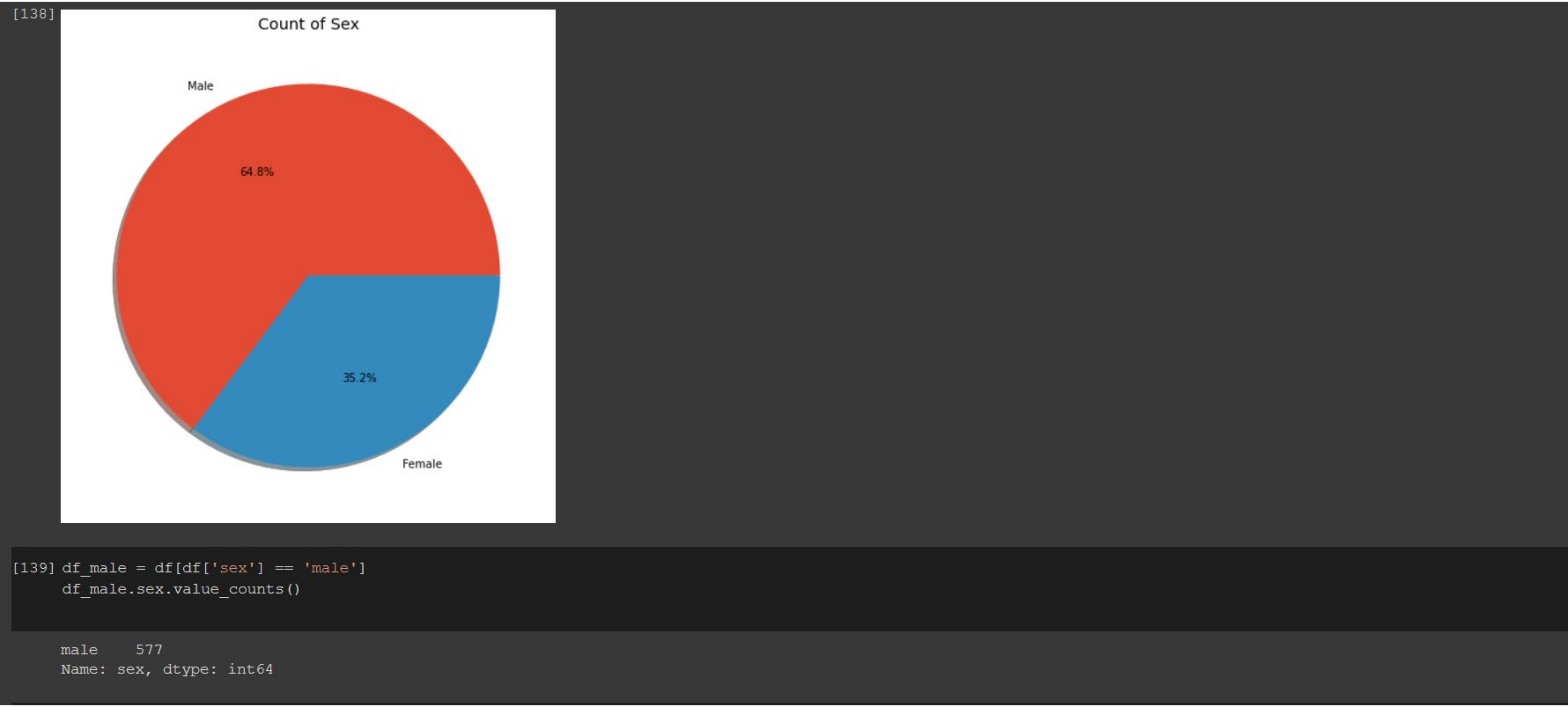
```
[119] df.head()
```

| | survived | pclass | sex | age | sibsp | parch | fare | embarked | class | who | adult_male | deck | embark_town | alive | alone |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 3 | male | 22.0 | 1 | 0 | 7.2500 | S | Third | man | True | NaN | Southampton | no | False |
| **1** | 1 | 1 | female | 38.0 | 1 | 0 | 71.2833 | C | First | woman | False | C | Cherbourg | yes | False |
| **2** | 1 | 3 | female | 26.0 | 0 | 0 | 7.9250 | S | Third | woman | False | NaN | Southampton | yes | True |
| **3** | 1 | 1 | female | 35.0 | 1 | 0 | 53.1000 | S | First | woman | False | C | Southampton | yes | False |
| **4** | 0 | 3 | male | 35.0 | 0 | 0 | 8.0500 | S | Third | man | True | NaN | Southampton | no | True |

```
[120] df.columns
```

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

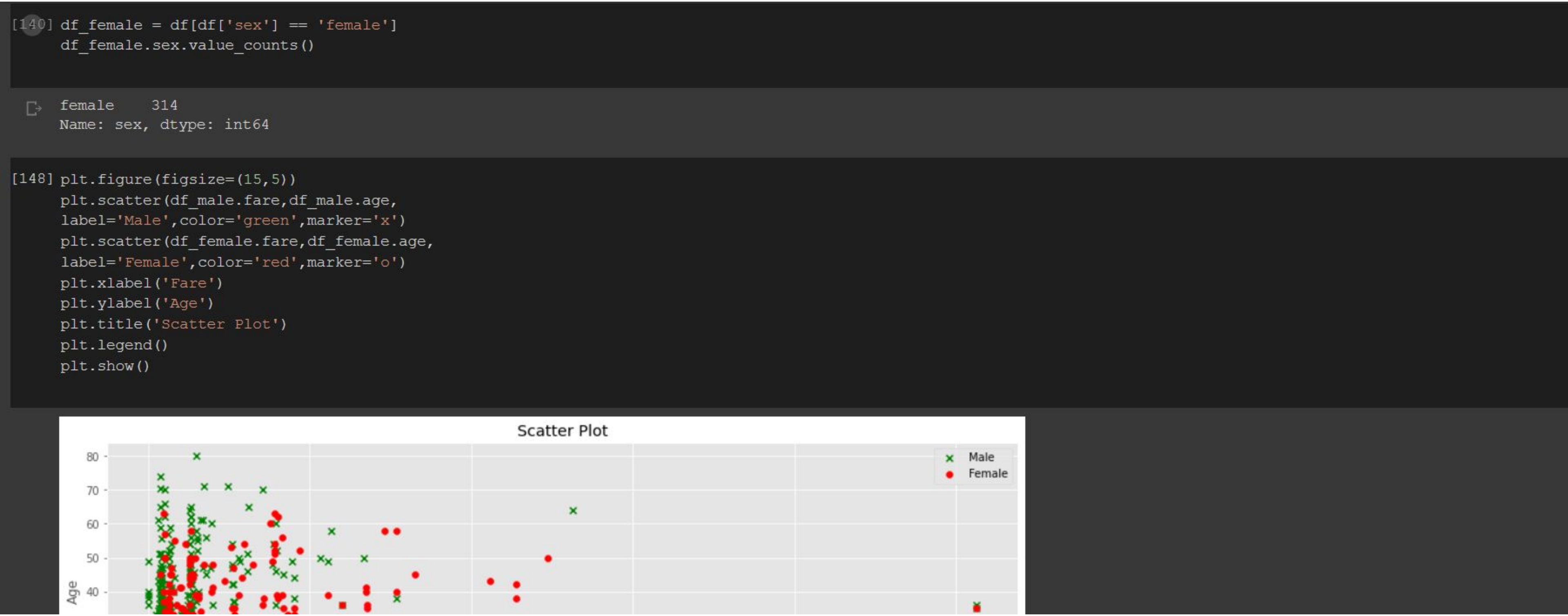
```
[138] plt.figure(figsize=(8,8))
      data=df.sex.value_counts()
      plt.pie(data,labels=['Male','Female'],autopct='%1.1f%%',shadow = True)
      plt.title('Count of Sex')
      plt.show()
```



[138]

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

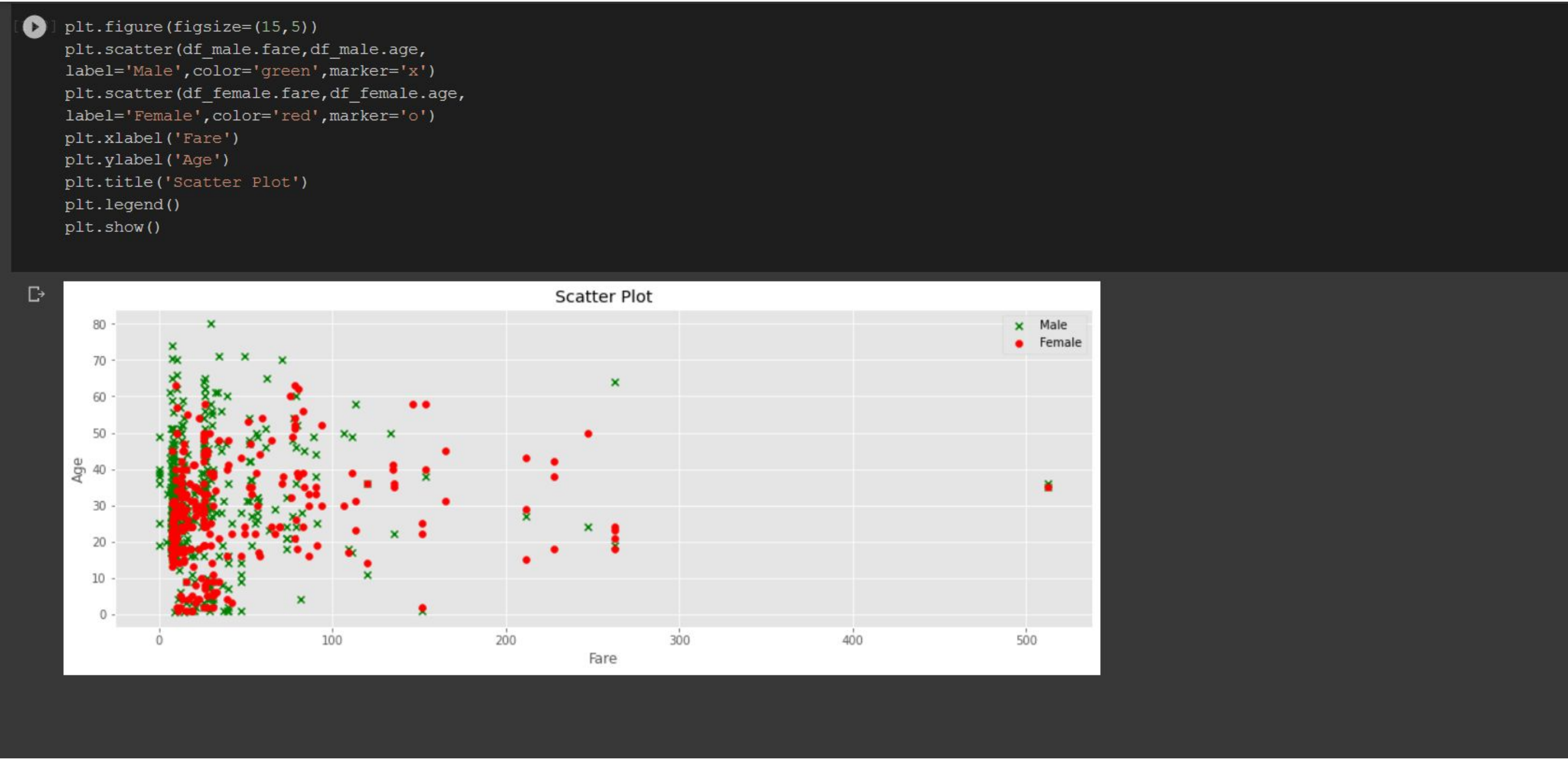
```
male    577
Name: sex, dtype: int64
```



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

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

```
[148] plt.figure(figsize=(15,5))
      plt.scatter(df_male.fare,df_male.age,
      label='Male',color='green',marker='x')
      plt.scatter(df_female.fare,df_female.age,
      label='Female',color='red',marker='o')
      plt.xlabel('Fare')
      plt.ylabel('Age')
      plt.title('Scatter Plot')
      plt.legend()
      plt.show()
```



```
plt.figure(figsize=(15,5))
plt.scatter(df_male.fare,df_male.age,
label='Male',color='green',marker='x')
plt.scatter(df_female.fare,df_female.age,
label='Female',color='red',marker='o')
plt.xlabel('Fare')
plt.ylabel('Age')
plt.title('Scatter Plot')
plt.legend()
plt.show()
```