Kmit

SONET

Data Science
(Level-I)

Ajeet K. Jain, M. Narsimlu
(ML TEAM)- SONET, KMIT, Hyderabad

Kmit SONET

Session - 23

This session deals with

Data Visualization

Data Interpretation

Matplotlib library

seaborn library

Types of graphs

Exercises on Matplotlib

Kmit SONET

Data Visualization

DATA SCIENCE

we can see the patterns and insights we’re looking for.

It becomes easier to grasp difficult concepts or identify new trends we may have missed.

We can use data visualizations to make an argument, or to support a hypothesis, or to explore our world in different ways.

Data visualization is the discipline of trying to understand data by placing it in a visual context

Data Visualization

The main benefits of data visualization are as follows:

It simplifies the complex quantitative information

It helps analyze and explore big data easily

It identifies the areas that need attention or improvement

It identifies the relationship between data points and variables

It explores new patterns and reveals hidden patterns in the data

Data Visualization

Three major considerations for Data Visualization:

Clarity

Accuracy

Efficiency

Clarity : Ensures that the data set is complete and relevant. This enables the data scientist to use the new patterns yield from the data in the relevant places.

Accuracy ensures using appropriate graphical representation to convey the right message.

Efficiency uses efficient visualization technique which highlights all the data points.

Data Visualization

Major factors that one would need to be aware of before visualizing the data.

Visual effect
Coordination System
Data Types and Scale
Informative Interpretation

Popular plotting libraries in Python

Python offers multiple graphing libraries that offers diverse features

• *matplotlib*	• to create 2D graphs and plots
• *pandas visualization*	• easy to use interface, built on Matplotlib
• *seaborn*	• provides a high-level interface for drawing attractive and informative statistical graphics
• *ggplot*	• based on R's ggplot2, uses Grammar of Graphics
• *plotly*	• can create interactive plots

Matplotlib.pyplot

- **matplotlib.pyplot** is a plotting library used for 2D graphics in python programming language.
- plotting with this Python data visualization library.
- plotting the data with the help of the plot() function
- show your plot using the show() function.

Matplotlib.pyplot

There are several advantages of using matplotlib to visualize data.

A multi-platform data visualization tool built on the numpy and scipy framework. Therefore, it's fast and efficient.
It possesses the ability to work well with many operating systems and graphic backends.
It possesses high-quality graphics and plots to print and view for a range of graphs such as histograms, bar charts, pie charts, scatter plots and heat maps.

Types of Graphs

- Types Of Plots
- – Line Graph
- – Bar Graph
- – Histogram
- – Scatter Plot

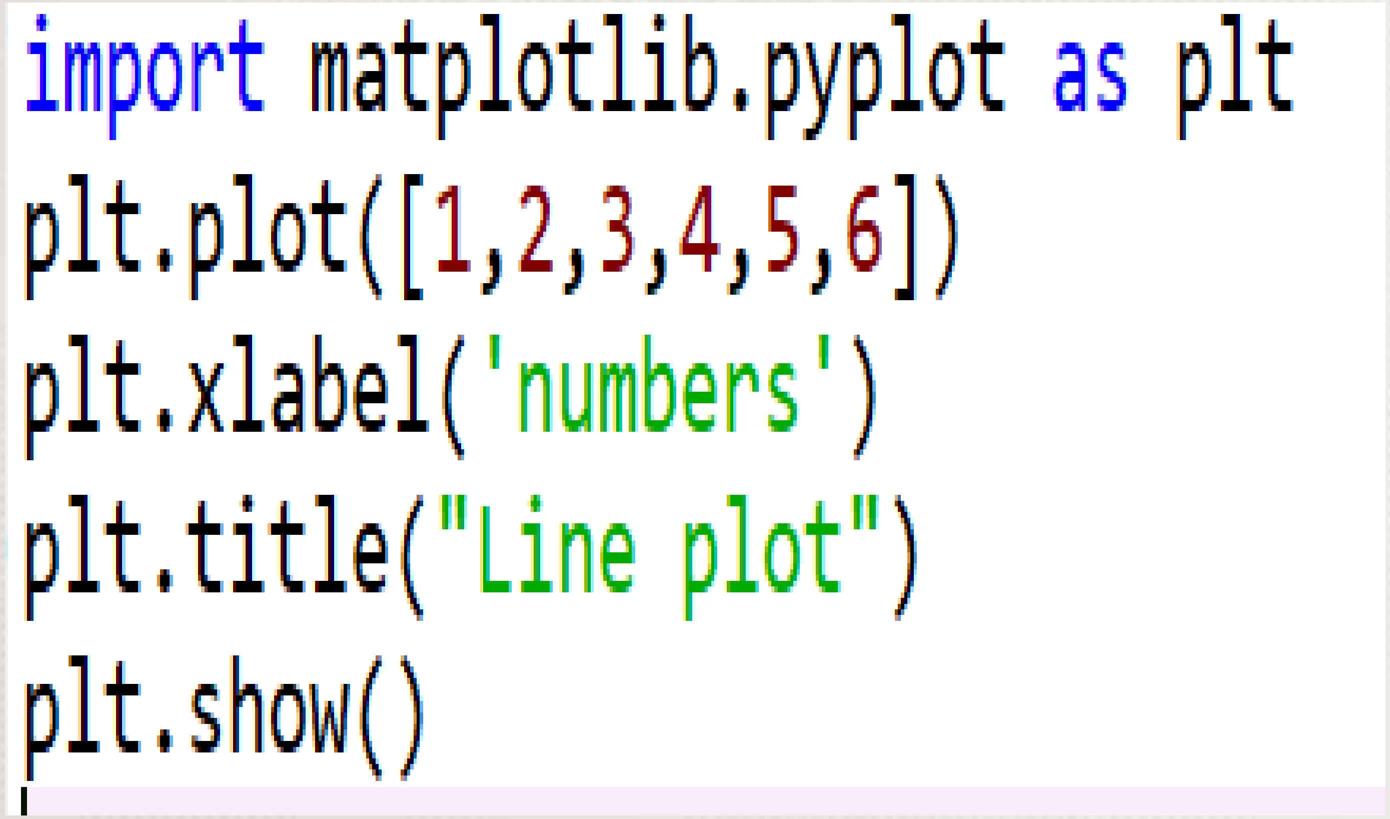
Line Plot-To create a line plot with text labels using plot().

KMIT SONET

Exercise -01

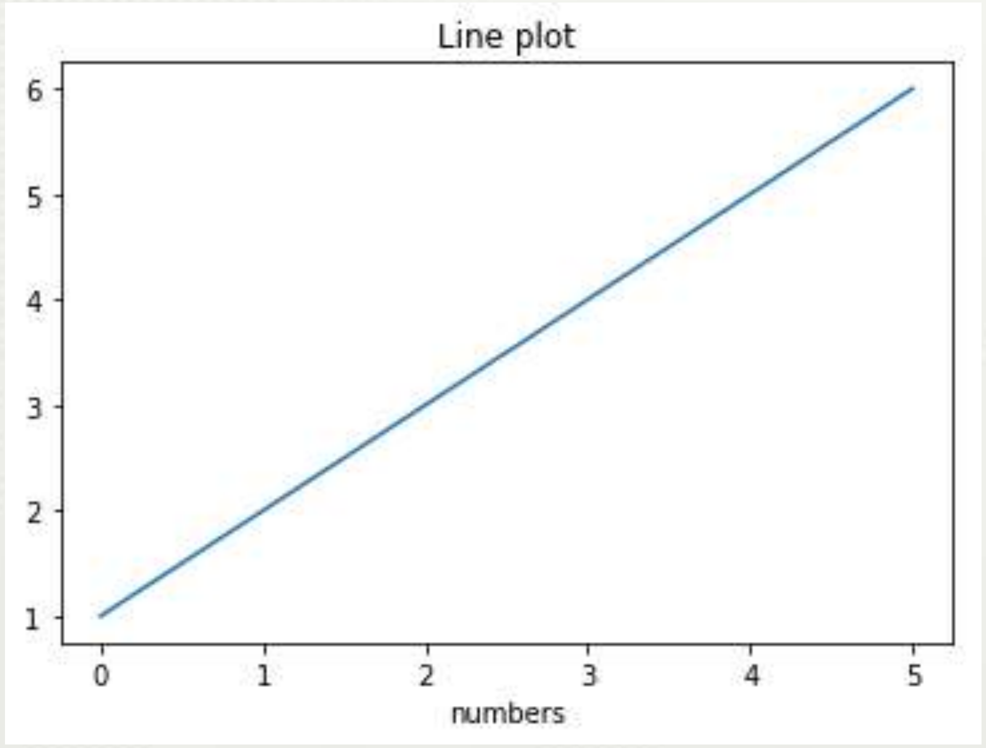
DATA SCIENCE

- Create a Line plot with integer numbers from 1 to 6 and name the title as Line graph



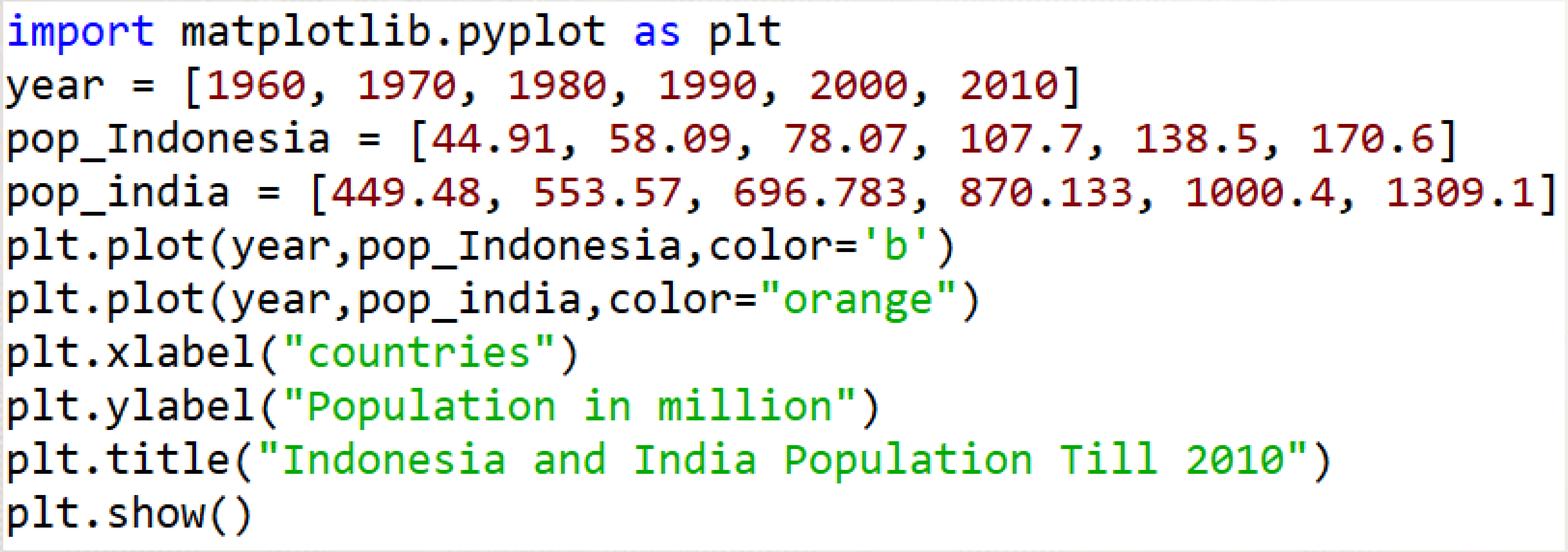
```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4,5,6])
plt.xlabel('numbers')
plt.title("Line plot")
plt.show()
```

Output



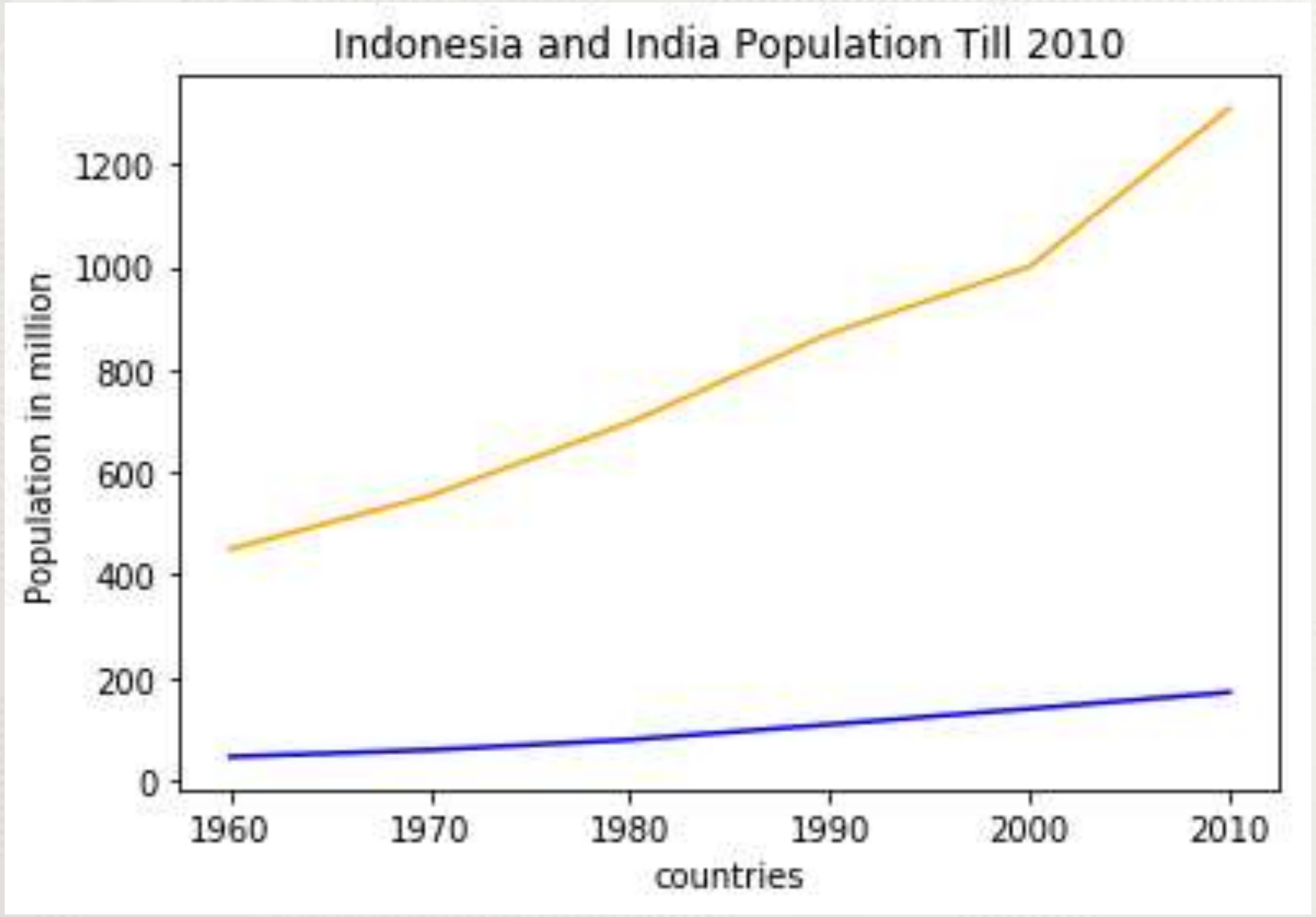
Exercise -02

- Create a pyplot which shows how population has been increased 1960 to 2010 between Indonesia and India.



```python
import matplotlib.pyplot as plt
year = [1960, 1970, 1980, 1990, 2000, 2010]
pop_Indonesia = [44.91, 58.09, 78.07, 107.7, 138.5, 170.6]
pop_india = [449.48, 553.57, 696.783, 870.133, 1000.4, 1309.1]
plt.plot(year,pop_Indonesia,color='b')
plt.plot(year,pop_india,color="orange")
plt.xlabel("countries")
plt.ylabel("Population in million")
plt.title("Indonesia and India Population Till 2010")
plt.show()
```

Output


Indonesia and India Population Till 2010
Population in million
countries
0
200
400
600
800
1000
1200
1960
1970
1980
1990
2000
2010

Line Plots with grids

- You can even try many styling techniques to create a better graph.
- What if you want to change the width or color of a particular line or what if you want to have some grid lines, there you need styling!
- let me show you how to add style to a graph using python matplotlib.

Exercise-03

- Create a line graph for the following data and perform following tasks:
- x = [5,8,10]
- y = [12,16,6]
- x2 = [6,9,11]
- y2 = [6,15,7]
- 1.create a line graph between x and y with blue color, label it as “line one”
- 2.create a line graph between x2 and y2 with red color, label it as “line two”
- 3.merge two lines using legend() method with grid color brown

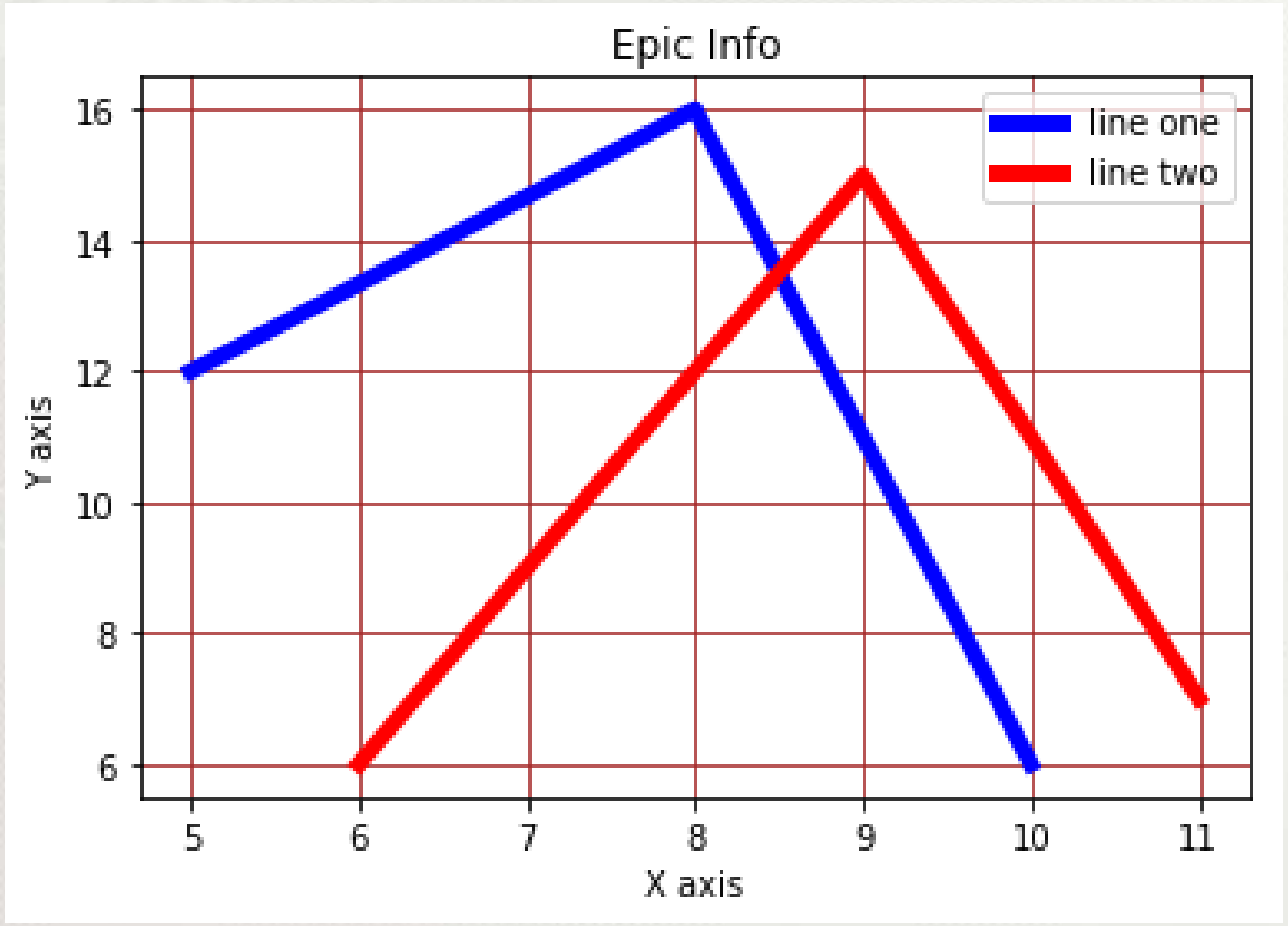
SONET

Solution

DATA SCIENCE

```
from matplotlib import pyplot as plt
x = [5,8,10]
y = [12,16,6]
x2 = [6,9,11]
y2 = [6,15,7]
plt.plot(x,y,'b',label='line one', linewidth=5)
plt.plot(x2,y2,'r',label='line two',linewidth=5)
plt.title('Epic Info')
plt.ylabel('Y axis')
plt.xlabel('X axis')
plt.legend()
plt.grid(True,color='brown')
plt.show()
```

Output



Epic Info
line one
line two
Y axis
X axis
16
14
12
10
8
6
5
6
7
8
9
10
11

Bar Graph

- A bar graph uses bars to compare data among different categories.
- It is well suited when you want to measure the changes over a period of time. It can be represented horizontally or vertically.
- Also, the important thing to keep in mind is that longer the bar, greater is the value.
- Use the bar() function to make bar charts, which includes customizations such as error bars.

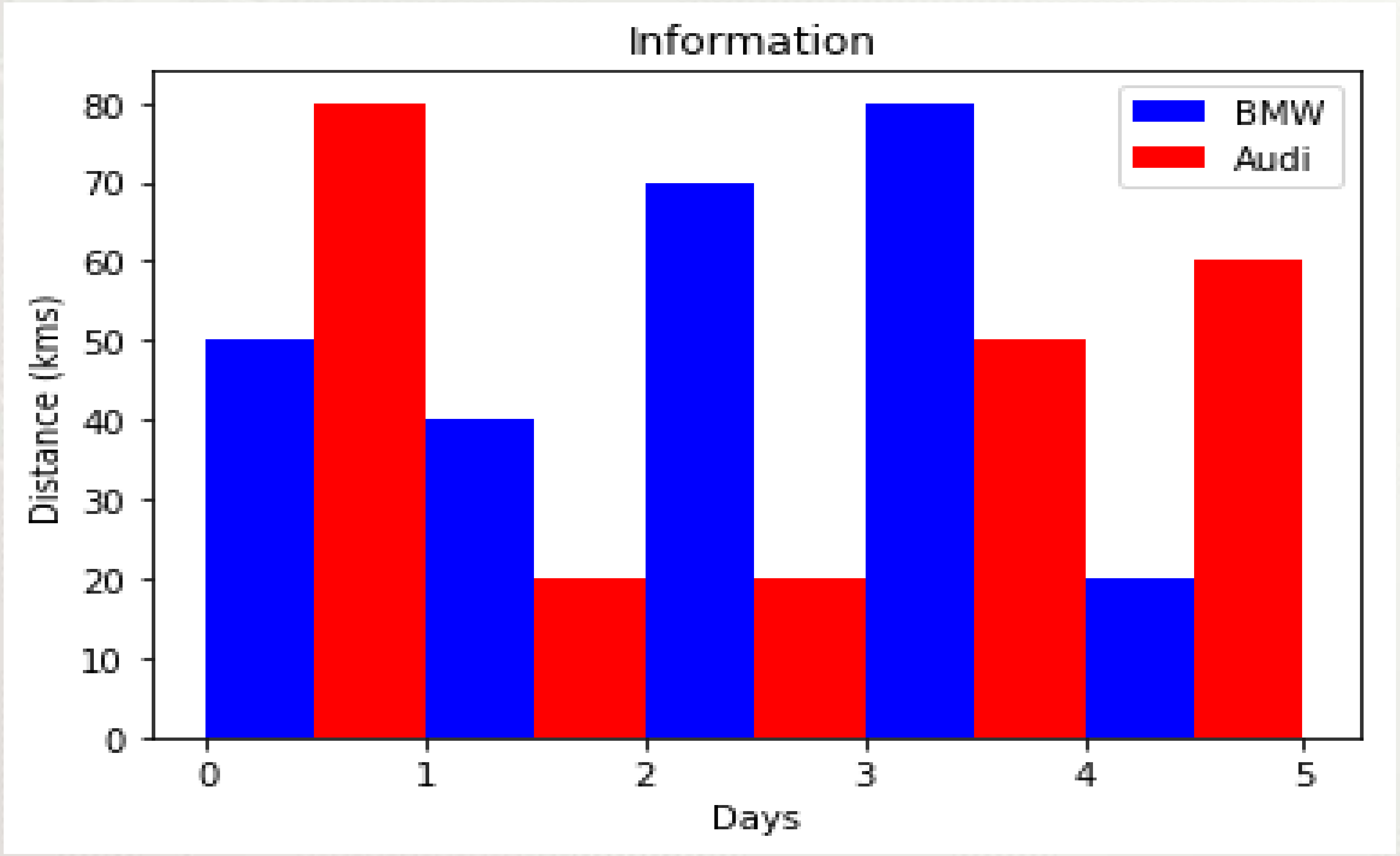
Exercise-4

- Create a bar graph for the following data:
- comparison between the distance covered by two cars BMW and Audi over a period of 5 days.
- BMW car:
- Days=[0.25,1.25,2.25,3.25,4.25]
- Distance= [50,40,70,80,20]
- Audi car:
- Days=[.75,1.75,2.75,3.75,4.75]
- Distance= [80,20,20,50,60]

Solution

```
from matplotlib import pyplot as plt
plt.bar([0.25,1.25,2.25,3.25,4.25],[50,40,70,80,20],
label="BMW",color='b',width=.5)
plt.bar([.75,1.75,2.75,3.75,4.75],[80,20,20,50,60],
label="Audi", color='r',width=.5)
plt.legend()
plt.xlabel('Days')
plt.ylabel('Distance (kms)')
plt.title('Information')
plt.show()
```

Output


Information
Distance (kms)
Days
BMW
Audi
0
10
20
30
40
50
60
70
80
0
1
2
3
4
5

Exercise-5

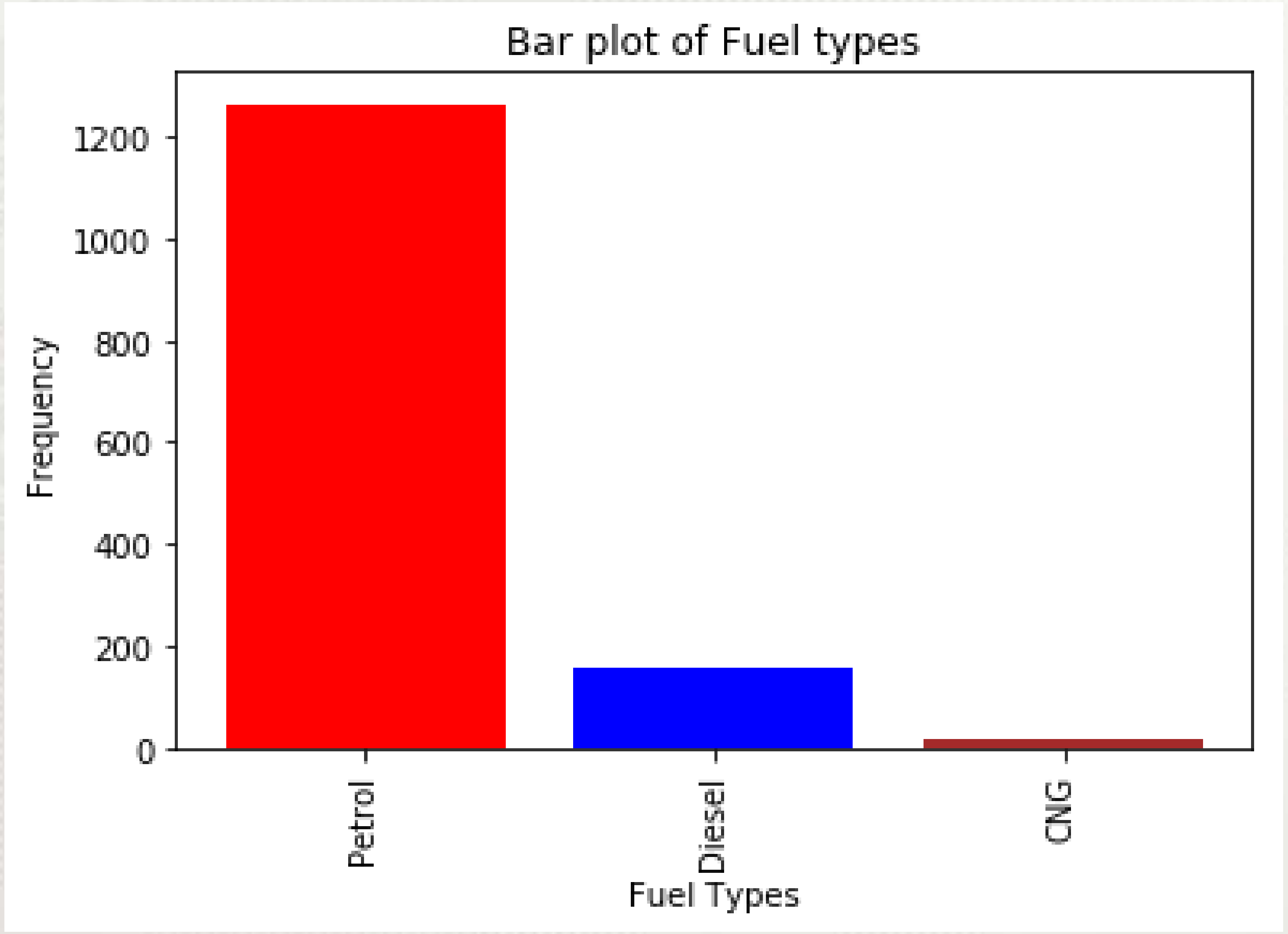
Read a Toyotacars dataset and perform the following tasks

1.Create a one way table on "Fuel_Type" feature
2.create a two list which consists of number of values of categories and categories
3.Find the length of categories list
4.create a bar graph betweeen index and count,add the colors as "red","blue","brown“ to the categories
5.Name of the graph title as "Bar plot of Fuel types
6.Name x label as "Fuel type" and y label as "Frequency"
7.Add sticks to categories as "Petrol","Diesel","CNG" with rotation "90"
8.show the graph

Exercise-5

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
data_cars=pd.read_csv("ToyotaCorolla.csv")
Fuel_tab=pd.crosstab(index=data_cars["Fuel_Type"],columns="counts")
print(Fuel_tab)
counts=[1264,155,17]
fueltype=["Petrol","Diesel","CNG"]
index=np.arange(len(fueltype))
plt.bar(index,counts,color=["red","blue","brown"])
plt.title("Bar plot of Fuel types")
plt.xlabel("Fuel Types")
plt.ylabel("Frequency")
plt.xticks(index,fueltype,rotation=90)
plt.show()
```

Output



Bar plot of Fuel types
Frequency
0
200
400
600
800
1000
1200
Petrol
Diesel
CNG
Fuel Types

Histogram

- Histograms are used to show a distribution whereas a bar chart is used to compare different entities.
- Histograms are useful when you have arrays or a very long list.
- Histograms are graphical representations of a frequency distribution of data.

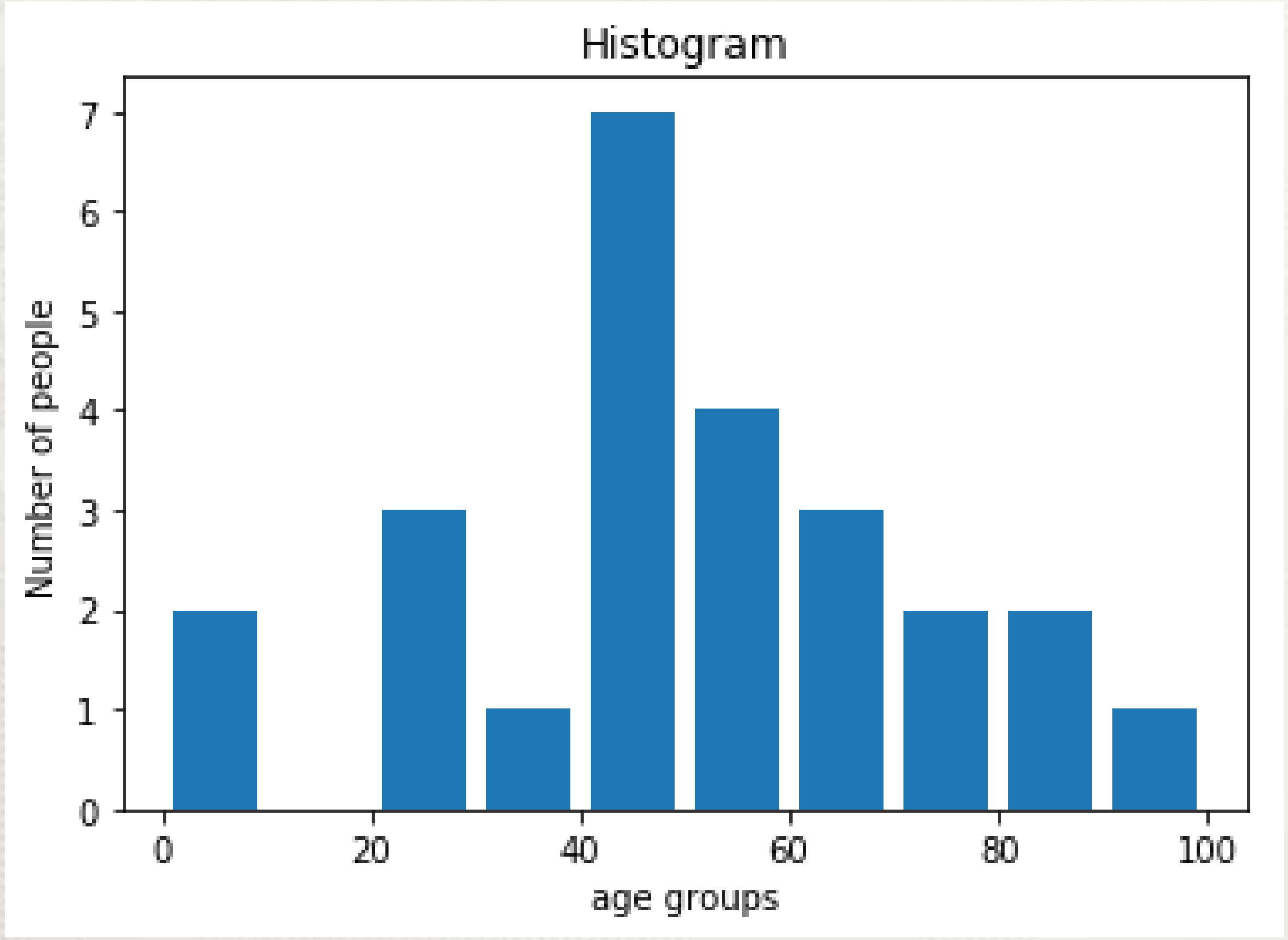
Exercise-5

- Create a histogram on the following data
- population_age =[22,55,62,45,21,22,34,42,42,4,2,102,95,85,55,110,120,
- 70,65,55,111,115,80,75,65,54,44,43,42,48]
- bins = [0,10,20,30,40,50,60,70,80,90,100]

Solution

```
import matplotlib.pyplot as plt
population_age =[22,55,62,45,21,22,34,42,42,4,2,102,95,85,55,110,120,
                 70,65,55,111,115,80,75,65,54,44,43,42,48]
bins = [0,10,20,30,40,50,60,70,80,90,100]
plt.hist(population_age, bins, histtype='bar', rwidth=0.8)
plt.xlabel('age groups')
plt.ylabel('Number of people')
plt.title('Histogram')
plt.show()
```

Output


Histogram
Number of people
age groups
0
1
2
3
4
5
6
7
0
20
40
60
80
100

Exercise-6

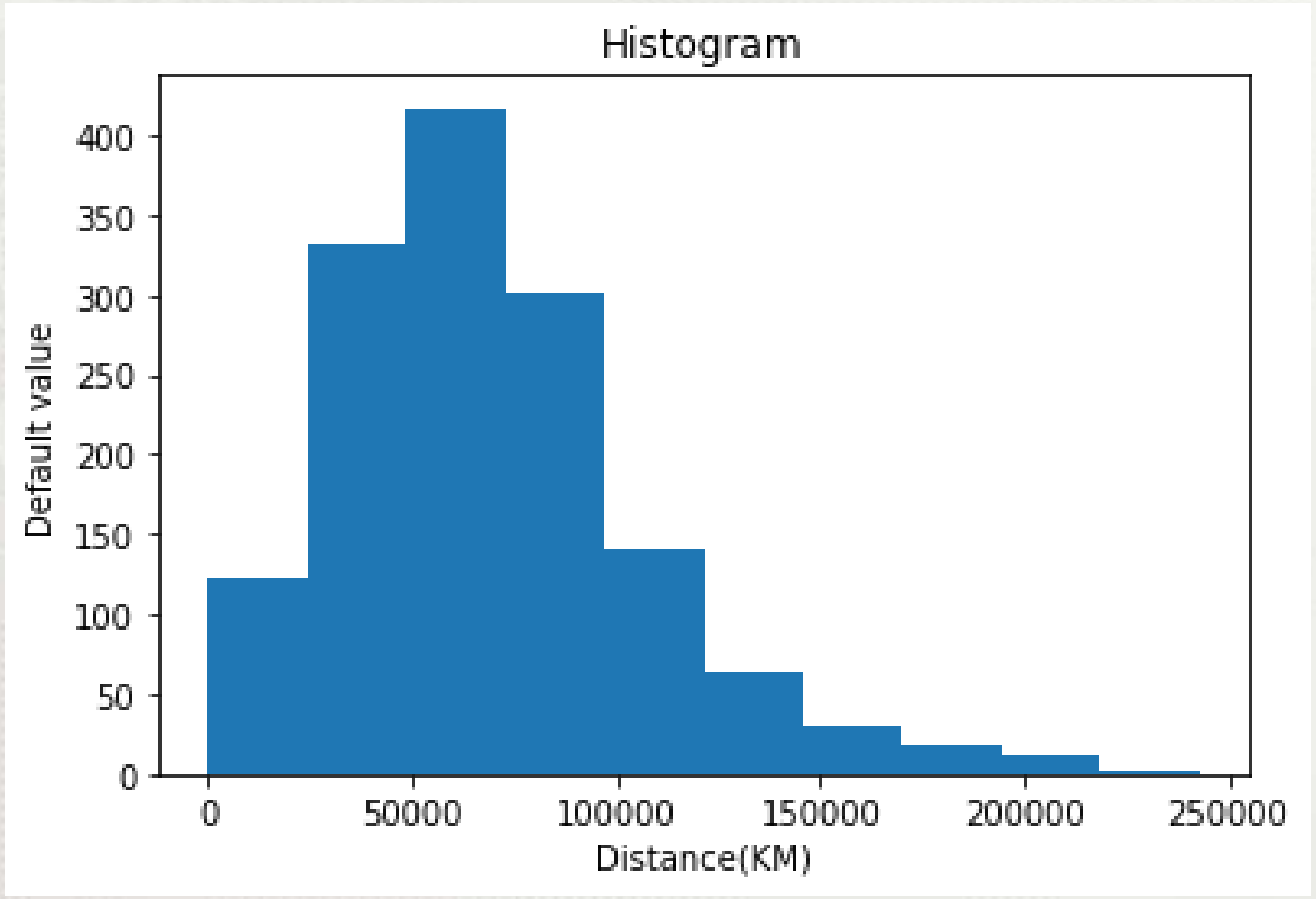
Read a Toyotacars dataset and perform the following tasks

1.Create a histogram on default argument the car which travelled no of KM.
2.create a histogram on with frequency and color of the bar and also edge color of the bar

1.Solution

```
import pandas as pd
import matplotlib.pyplot as plt
data_cars=pd.read_csv("ToyotaCorolla.csv")
plt.hist(data_cars["KM"])
plt.title("Histogram")
plt.xlabel("Distance(KM)")
plt.ylabel("Default value")
plt.show()
```

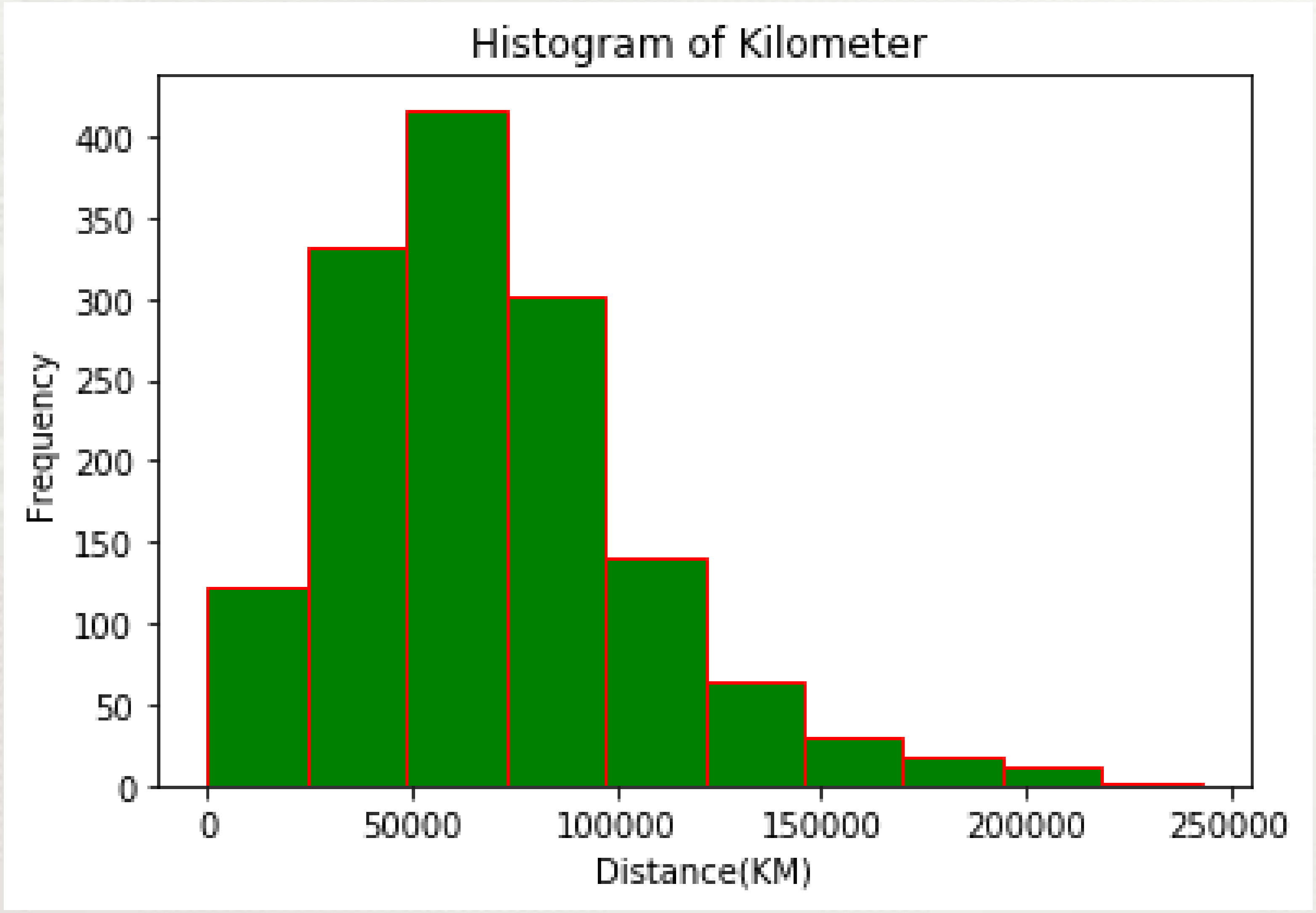
Output


Histogram
Default value
Distance(KM)
0
50
100
150
200
250
300
350
400
0
50000
100000
150000
200000
250000

2.Solution

```python
import pandas as pd
import matplotlib.pyplot as plt
data_cars=pd.read_csv("ToyotaCorolla.csv")
plt.hist(data_cars["KM"],bins=10,color="green",edgecolor="red")
plt.title("Histogram of Kilometer ")
plt.xlabel("Distance(KM)")
plt.ylabel("Frequency")
plt.show()
```

Output


Histogram of Kilometer
Frequency
Distance(KM)
0
50
100
150
200
250
300
350
400
0
50000
100000
150000
200000
250000

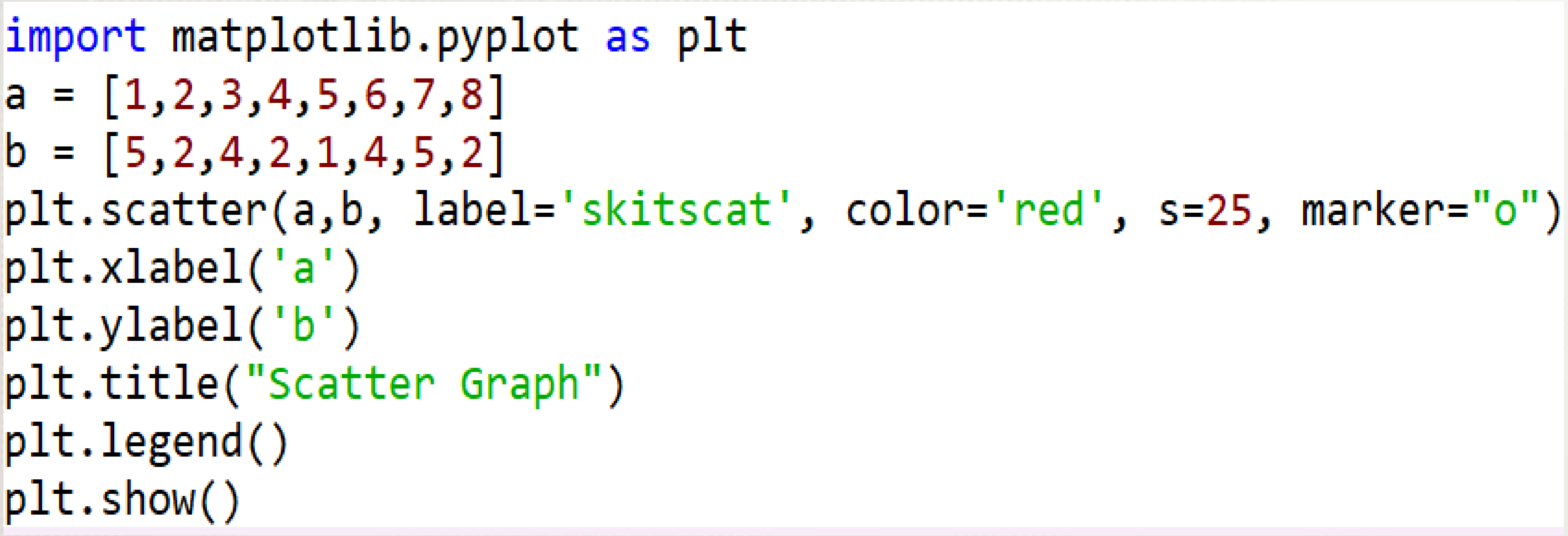
Scatter Plot

- Usually we need scatter plots in order to compare variables.

- How much one variable is affected by another variable to build a relation out of it.
- The scatter() function makes a scatter plot with (optional) size and color arguments.
- Scatter plots are used to convey the relationship between numerical variables
- The data is displayed as a collection of points, each having the value of one variable which determines the position on the horizontal axis and the value of other variable determines the position on the vertical axis.

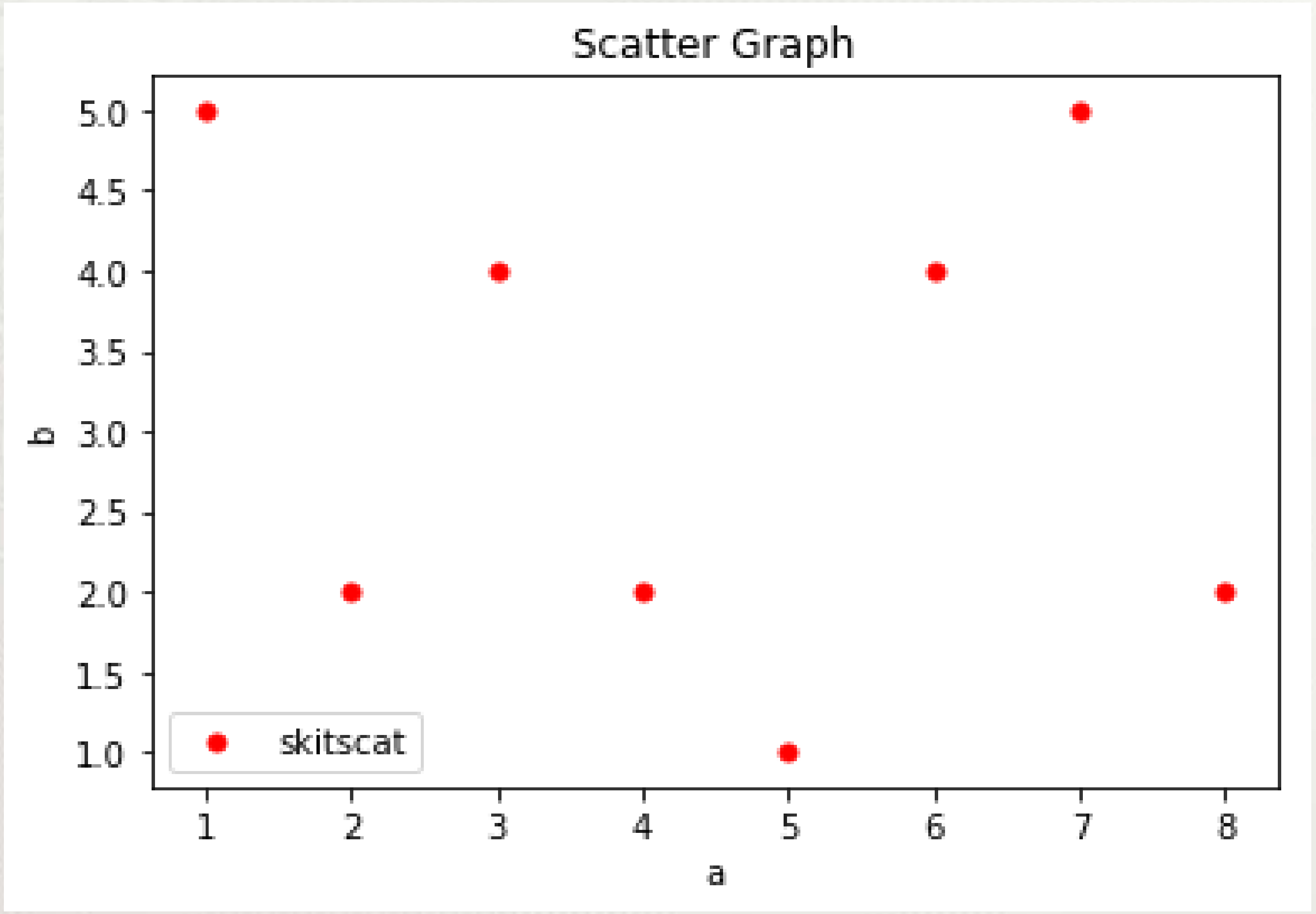
Exercise-6

- Create a scatter graph for the following data:
- a = [1,2,3,4,5,6,7,8]
- b = [5,2,4,2,1,4,5,2]



```
import matplotlib.pyplot as plt
a = [1,2,3,4,5,6,7,8]
b = [5,2,4,2,1,4,5,2]
plt.scatter(a,b, label='skitscat', color='red', s=25, marker="o")
plt.xlabel('a')
plt.ylabel('b')
plt.title("Scatter Graph")
plt.legend()
plt.show()
```

Exercise-6



Scatter-plot

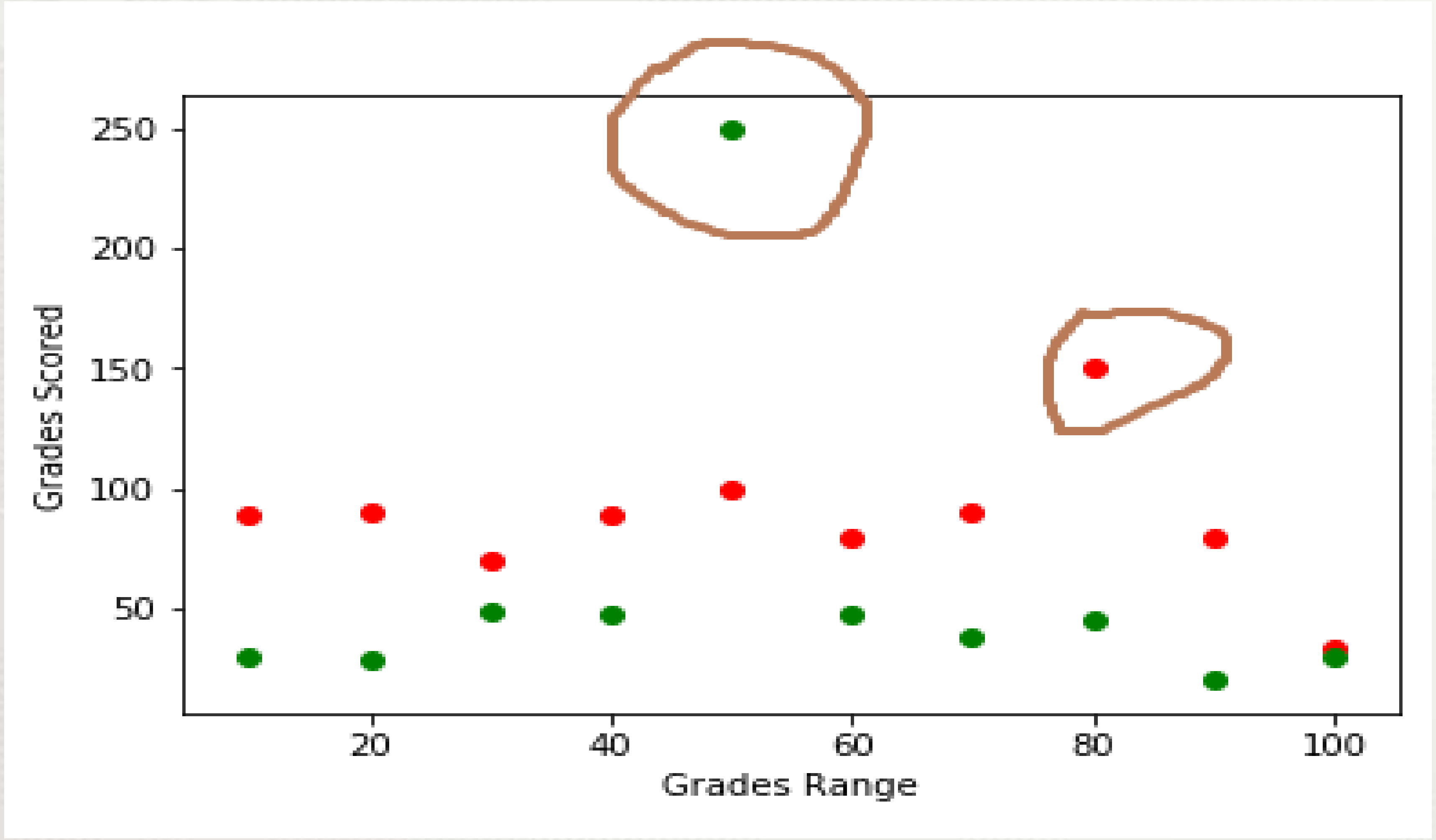
- Scatter Plots are usually used to represent the correlation between two or more variables. It also helps it identify **Outliers**, if any.

- Create a scatter plot for the following data
- girls_grades = [89, 90, 70, 89, 100, 80, 90, 150, 80, 34]
- boys_grades = [30, 29, 49, 48, 250, 48, 38, 45, 20, 30]
- grades_range = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
- 1.create a relation between grades ranges and girls grades features
- 2. create a relation between grades ranges and boys grades features

Exercise-7

```python
import matplotlib.pyplot as plt
import pandas as pd
girls_grades = [89, 90, 70, 89, 100, 80, 90, 150, 80, 34]
boys_grades = [30, 29, 49, 48, 250, 48, 38, 45, 20, 30]
grades_range = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
plt.scatter(grades_range, girls_grades,color='r')
plt.scatter(grades_range, boys_grades,color='g')
plt.xlabel('Grades Range')
plt.ylabel('Grades Scored')
plt.show()
```

Output



Exercise-8

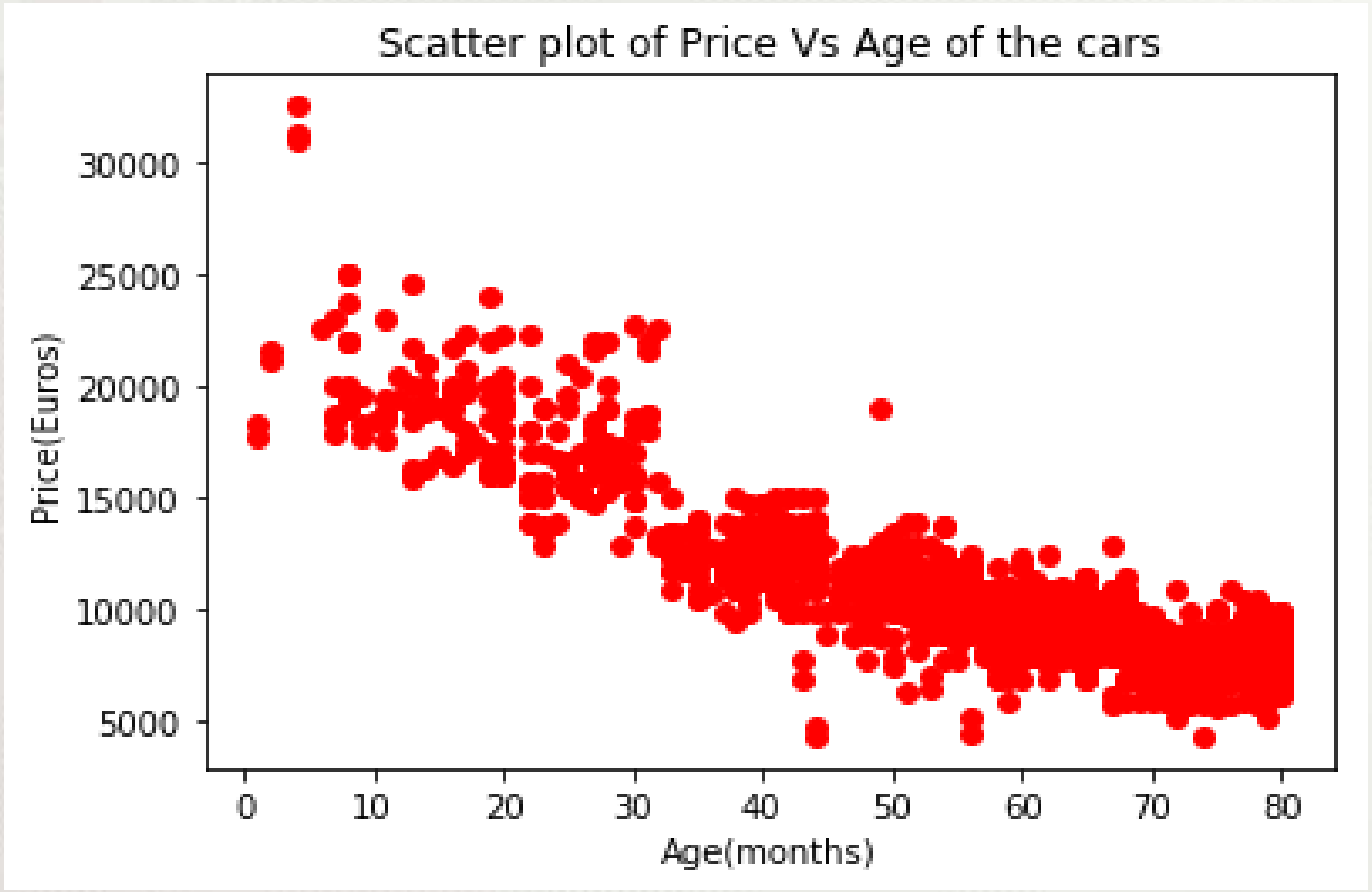
Read a Toyota cars dataset and perform the following tasks

1. Find missing values and display them
2. Remove missing values and display them
3. create a scatter plot between Age vs price

Solution

```
import pandas as pd
import matplotlib.pyplot as plt
data_cars=pd.read_csv("ToyotaCorolla.csv")
print(data_cars.isnull().sum())
print(data_cars)
data_cars.dropna(axis=0,inplace=True)
print(data_cars)
plt.scatter(data_cars["Age_08_04"],data_cars["Price"],c="red")
plt.title("Scatter plot of Price Vs Age of the cars")
plt.xlabel("Age(months)")
plt.ylabel("Price(Euros)")
plt.show()
```

Output



Scatter plot of Price Vs Age of the cars
Price(Euros)
Age(months)
30000
25000
20000
15000
10000
5000
0
10
20
30
40
50
60
70
80

Conclusion

You are aware of

Data Visualization

Data Interpretation

We will proceed with

Data Visualization

Kmit SONET

DATA SCIENCE

THANK YOU