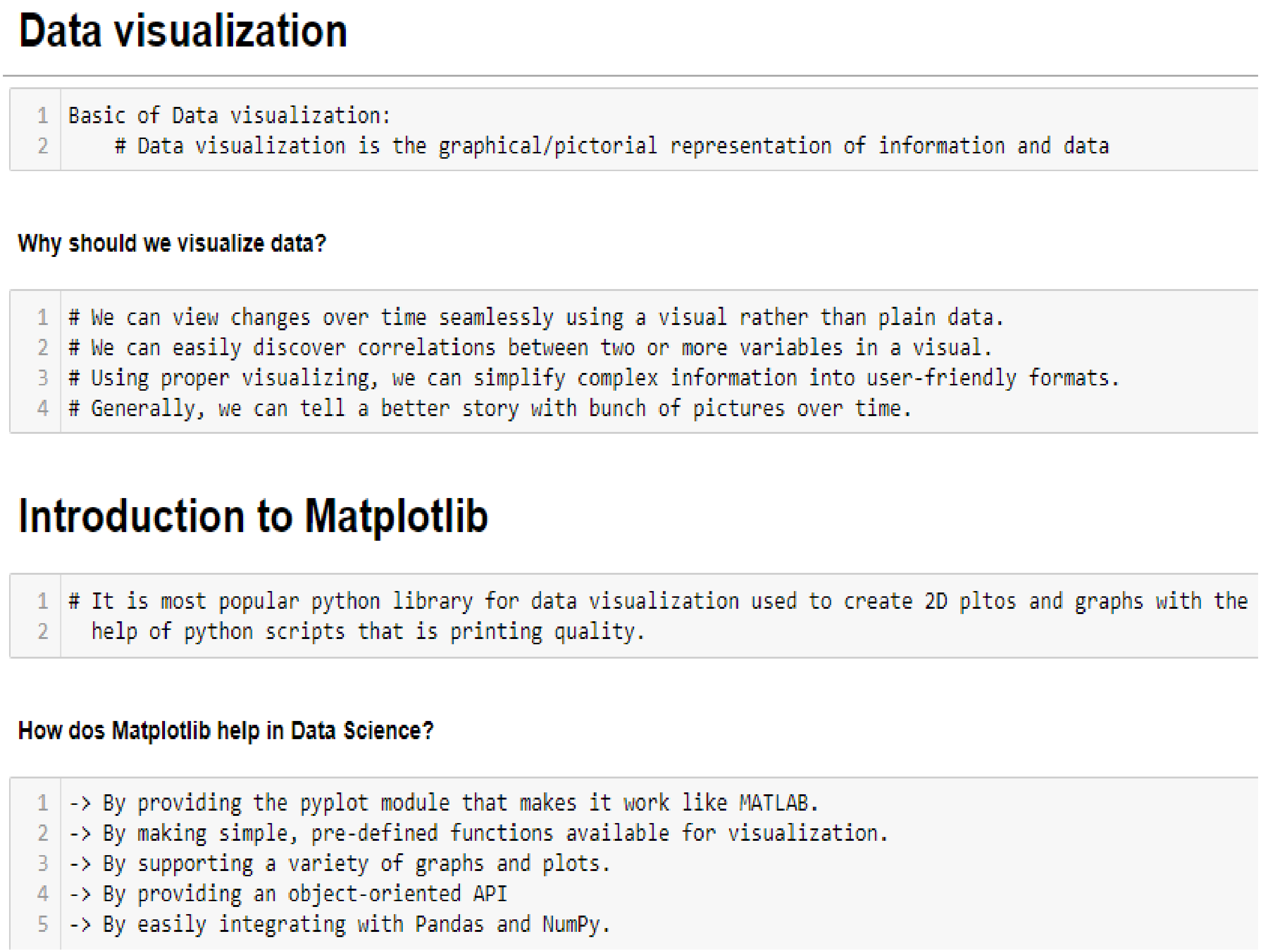


Data visualization

```
Basic of Data visualization:
    # Data visualization is the graphical/pictorial representation of information and data
```

Why should we visualize data?

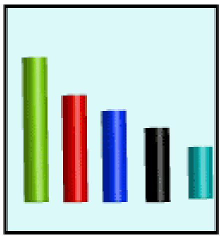
```
# We can view changes over time seamlessly using a visual rather than plain data.
# We can easily discover correlations between two or more variables in a visual.
# Using proper visualizing, we can simplify complex information into user-friendly formats.
# Generally, we can tell a better story with bunch of pictures over time.
```

Introduction to Matplotlib

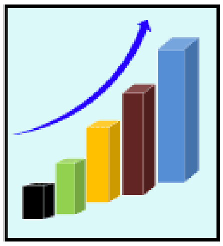
```
# It is most popular python library for data visualization used to create 2D pltos and graphs with the
  help of python scripts that is printing quality.
```

How dos Matplotlib help in Data Science?

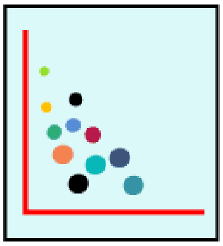
```
-> By providing the pyplot module that makes it work like MATLAB.
-> By making simple, pre-defined functions available for visualization.
-> By supporting a variety of graphs and plots.
-> By providing an object-oriented API
-> By easily integrating with Pandas and NumPy.
```



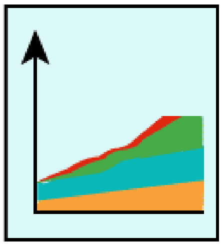
Bar Graph



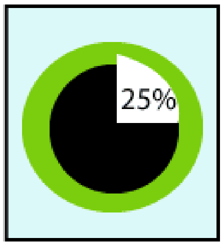
Histogram



Scatter Plot



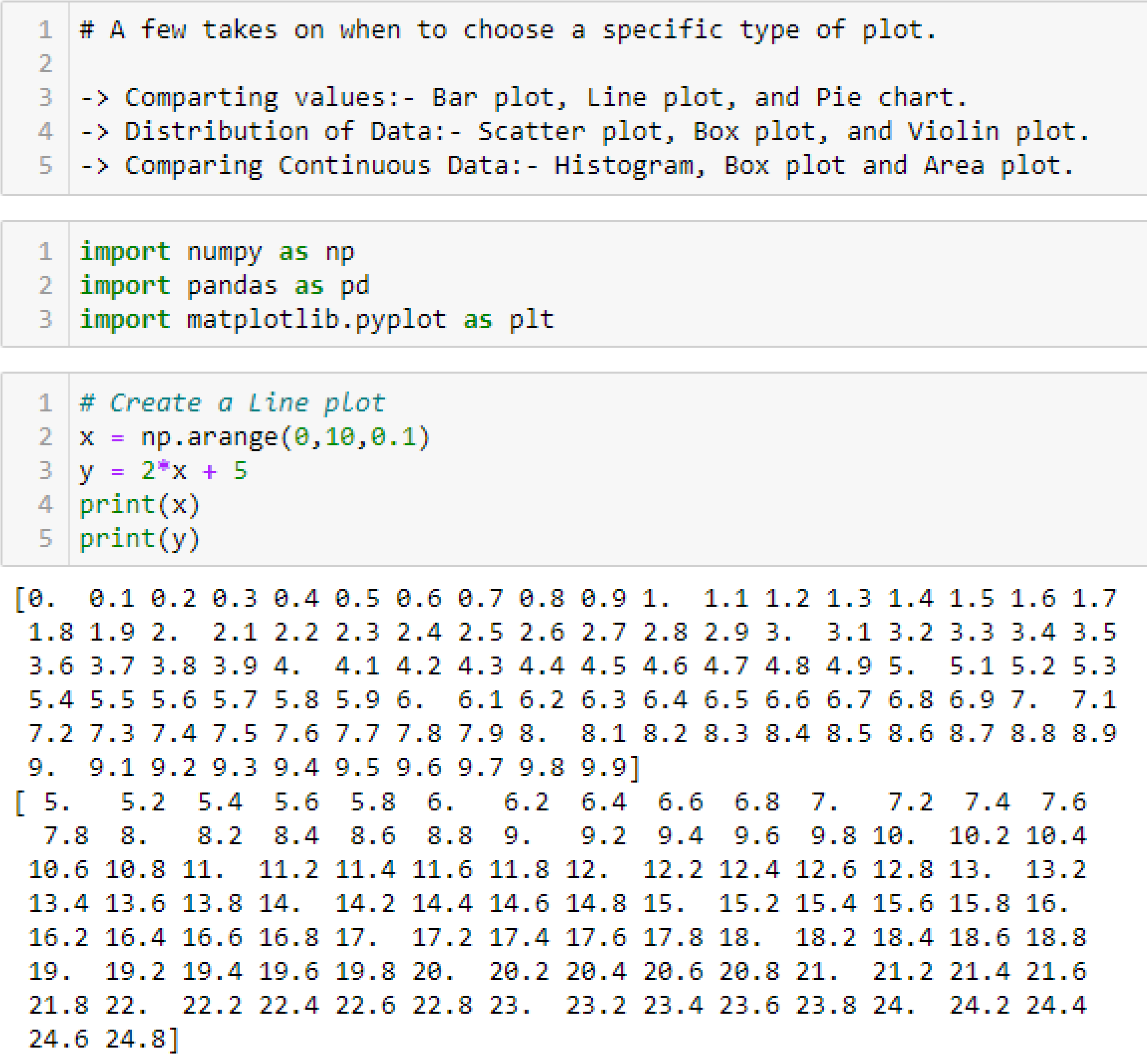
Area Plot



Pie Plot

There are five phases which are essential to make the decision for the organization:

Types of plots

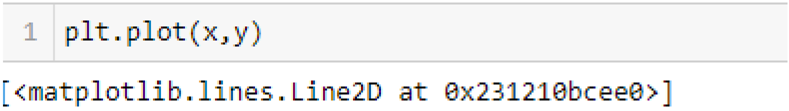


```
# A few takes on when to choose a specific type of plot.

-> Comparing values:- Bar plot, Line plot, and Pie chart.
-> Distribution of Data:- Scatter plot, Box plot, and Violin plot.
-> Comparing Continuous Data:- Histogram, Box plot and Area plot.
```

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

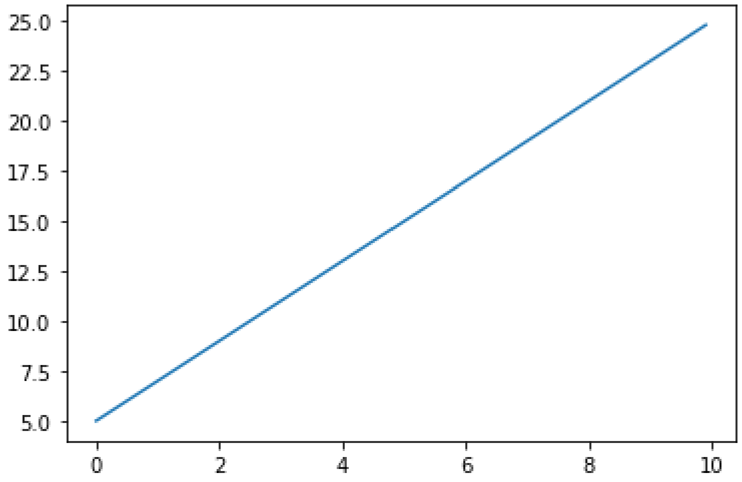
```
# Create a Line plot
x = np.arange(0,10,0.1)
y = 2*x + 5
print(x)
print(y)
```

```
[0.  0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7
 1.8 1.9 2.  2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3.  3.1 3.2 3.3 3.4 3.5
 3.6 3.7 3.8 3.9 4.  4.1 4.2 4.3 4.4 4.5 4.6 4.7 4.8 4.9 5.  5.1 5.2 5.3
 5.4 5.5 5.6 5.7 5.8 5.9 6.  6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8 6.9 7.  7.1
 7.2 7.3 7.4 7.5 7.6 7.7 7.8 7.9 8.  8.1 8.2 8.3 8.4 8.5 8.6 8.7 8.8 8.9
 9.  9.1 9.2 9.3 9.4 9.5 9.6 9.7 9.8 9.9]
[ 5.   5.2  5.4  5.6  5.8  6.   6.2  6.4  6.6  6.8  7.   7.2  7.4  7.6
  7.8  8.   8.2  8.4  8.6  8.8  9.   9.2  9.4  9.6  9.8 10.  10.2 10.4
 10.6 10.8 11.  11.2 11.4 11.6 11.8 12.  12.2 12.4 12.6 12.8 13.  13.2
 13.4 13.6 13.8 14.  14.2 14.4 14.6 14.8 15.  15.2 15.4 15.6 15.8 16.
 16.2 16.4 16.6 16.8 17.  17.2 17.4 17.6 17.8 18.  18.2 18.4 18.6 18.8
 19.  19.2 19.4 19.6 19.8 20.  20.2 20.4 20.6 20.8 21.  21.2 21.4 21.6
 21.8 22.  22.2 22.4 22.6 22.8 23.  23.2 23.4 23.6 23.8 24.  24.2 24.4
 24.6 24.8]
```

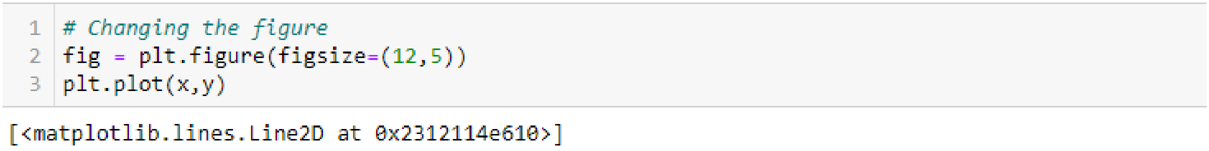


```
plt.plot(x,y)
```

```
[<matplotlib.lines.Line2D at 0x231210bcee0>]
```

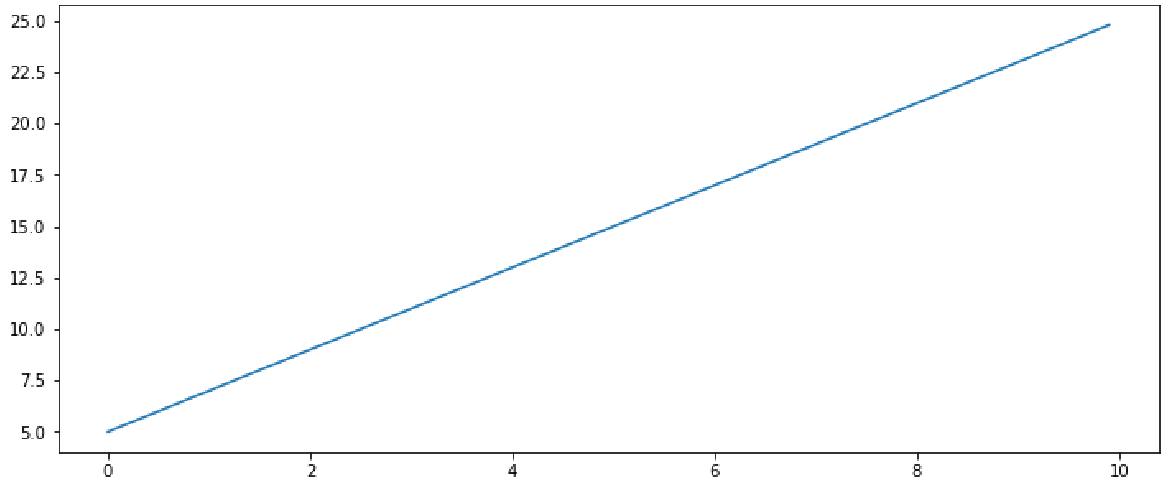


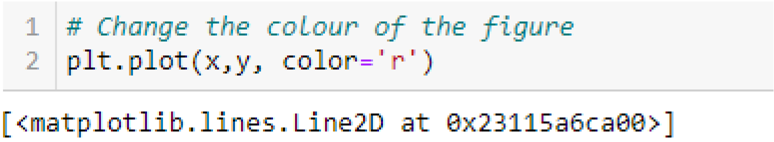
Customize the Line plot



```
# Changing the figure
fig = plt.figure(figsize=(12,5))
plt.plot(x,y)
```

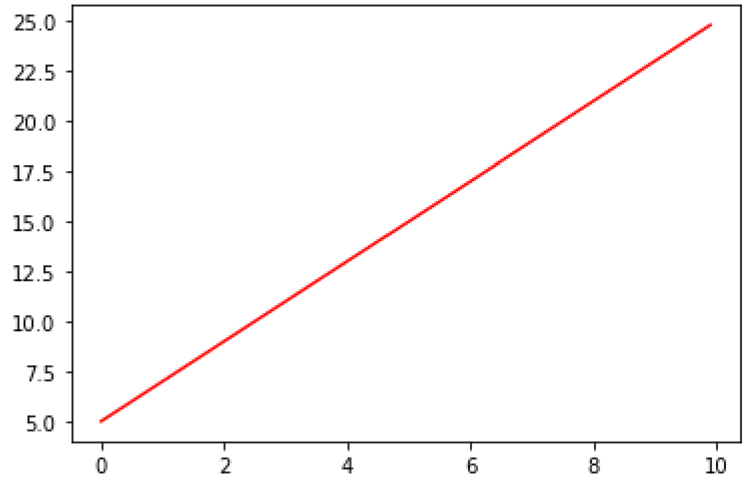
```
[<matplotlib.lines.Line2D at 0x2312114e610>]
```

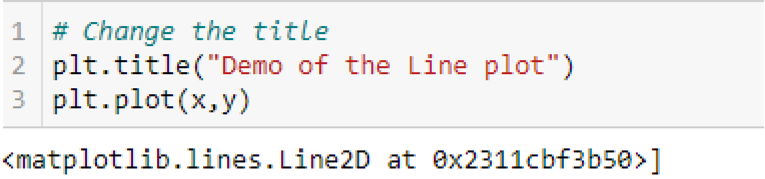




```
# Change the colour of the figure
plt.plot(x,y, color='r')
```

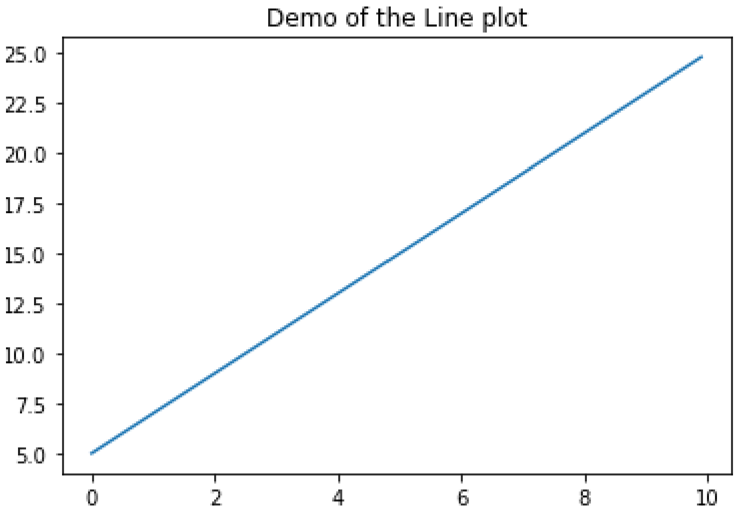
```
[<matplotlib.lines.Line2D at 0x23115a6ca00>]
```

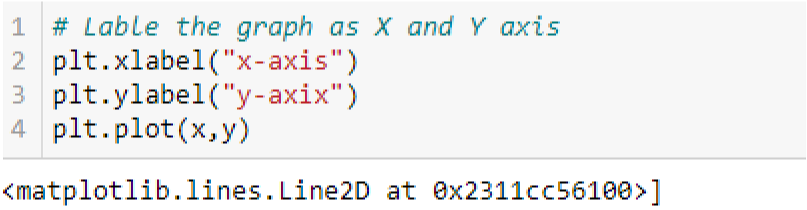




```
# Change the title
plt.title("Demo of the Line plot")
plt.plot(x,y)
```

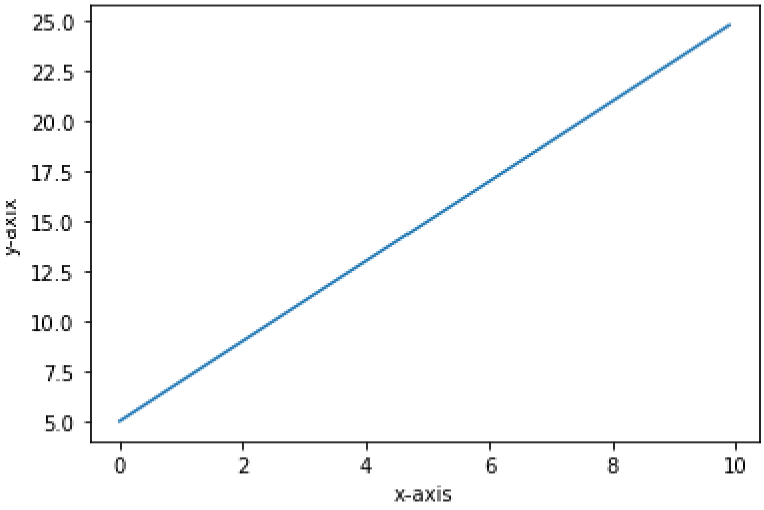
```
<matplotlib.lines.Line2D at 0x2311cbf3b50>]
```

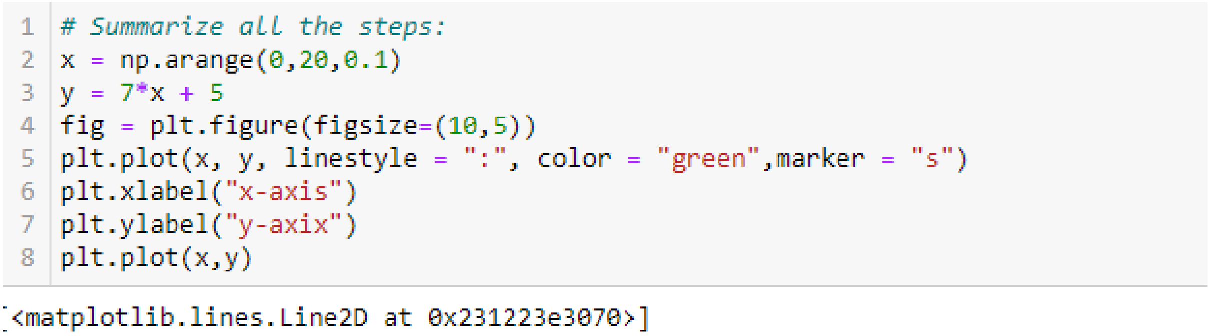




```
# Lable the graph as X and Y axis
plt.xlabel("x-axis")
plt.ylabel("y-axix")
plt.plot(x,y)
```

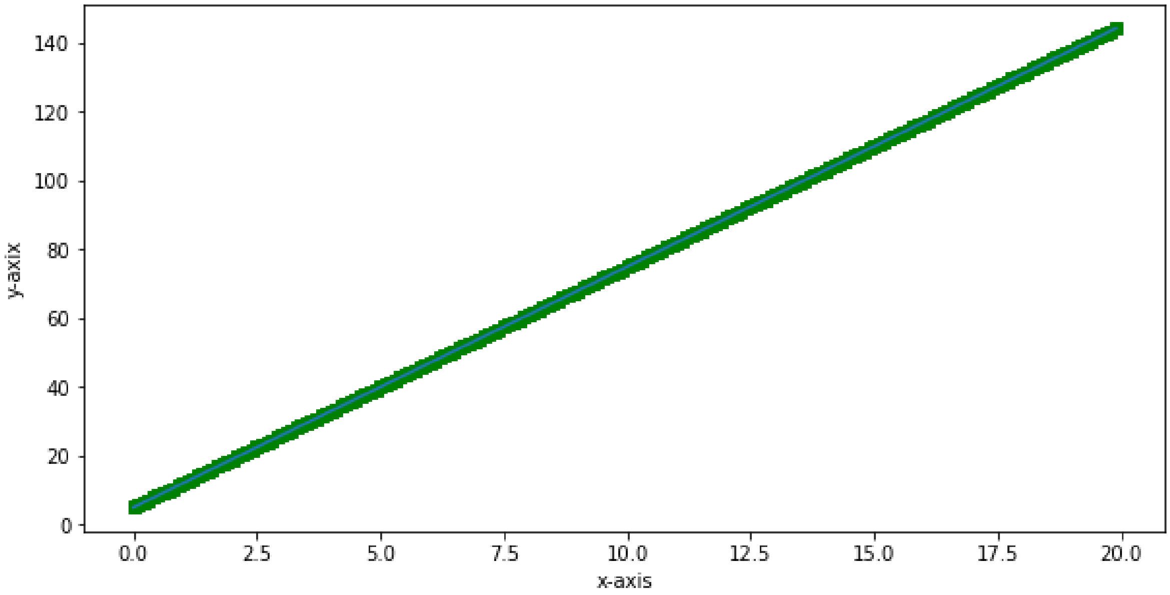
```
<matplotlib.lines.Line2D at 0x2311cc56100>]
```

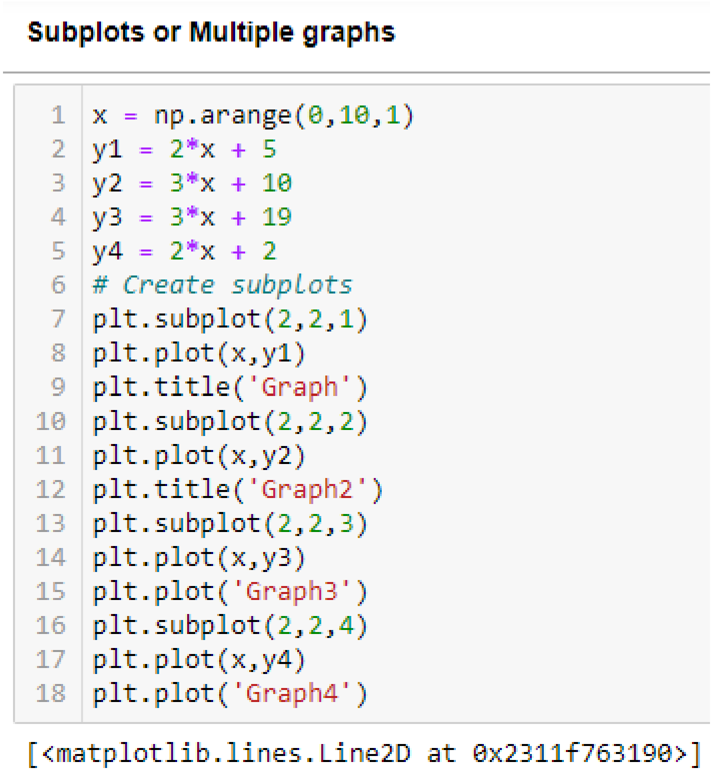




```
# Summarize all the steps:
x = np.arange(0,20,0.1)
y = 7*x + 5
fig = plt.figure(figsize=(10,5))
plt.plot(x, y, linestyle = ":", color = "green",marker = "s")
plt.xlabel("x-axis")
plt.ylabel("y-axix")
plt.plot(x,y)
```

```
[<matplotlib.lines.Line2D at 0x231223e3070>]
```

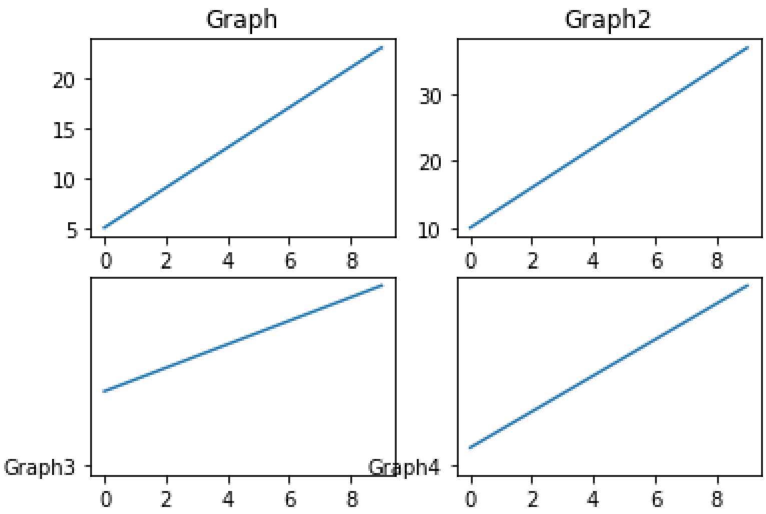


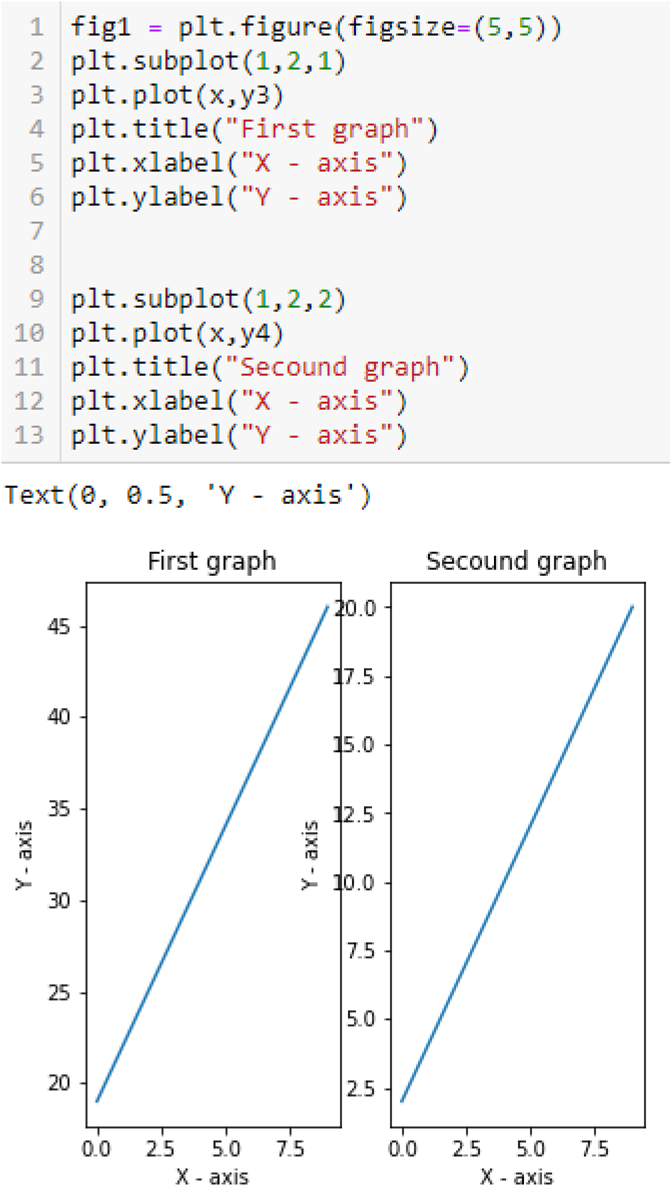


Subplots or Multiple graphs

```
x = np.arange(0,10,1)
y1 = 2*x + 5
y2 = 3*x + 10
y3 = 3*x + 19
y4 = 2*x + 2
# Create subplots
plt.subplot(2,2,1)
plt.plot(x,y1)
plt.title('Graph')
plt.subplot(2,2,2)
plt.plot(x,y2)
plt.title('Graph2')
plt.subplot(2,2,3)
plt.plot(x,y3)
plt.plot('Graph3')
plt.subplot(2,2,4)
plt.plot(x,y4)
plt.plot('Graph4')
```

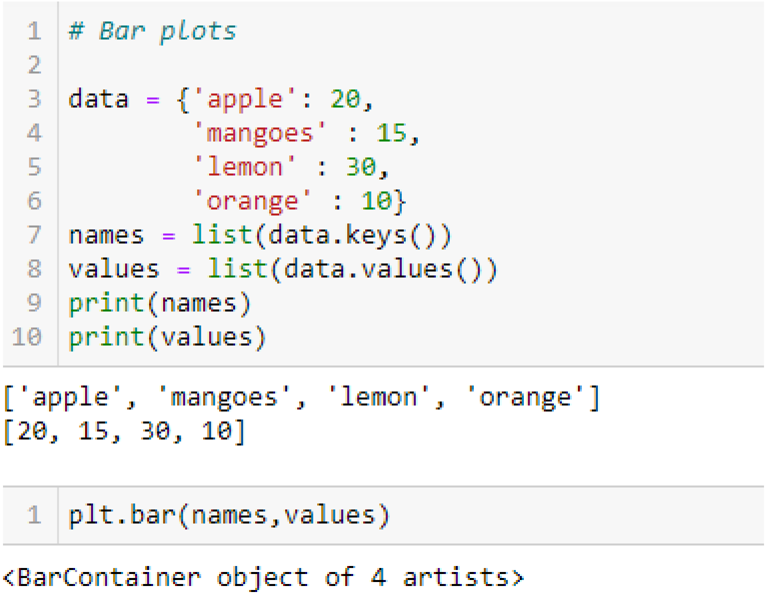
```
[<matplotlib.lines.Line2D at 0x2311f763190>]
```





```
fig1 = plt.figure(figsize=(5,5))
plt.subplot(1,2,1)
plt.plot(x,y3)
plt.title("First graph")
plt.xlabel("X - axis")
plt.ylabel("Y - axis")


plt.subplot(1,2,2)
plt.plot(x,y4)
plt.title("Secound graph")
plt.xlabel("X - axis")
plt.ylabel("Y - axis")
```

```
Text(0, 0.5, 'Y - axis')
```

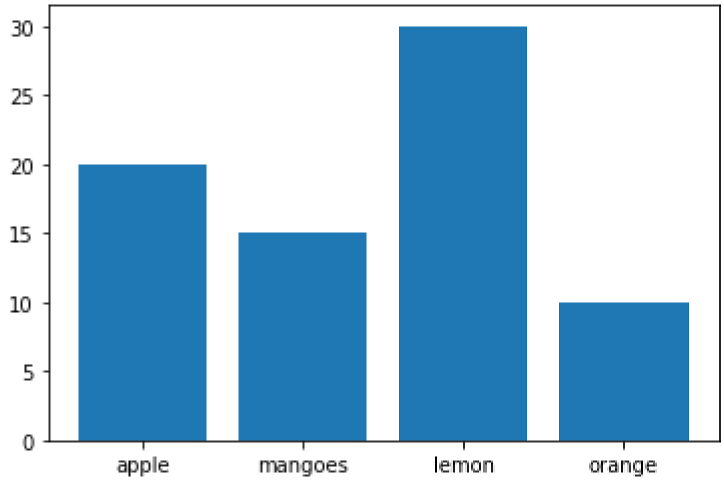


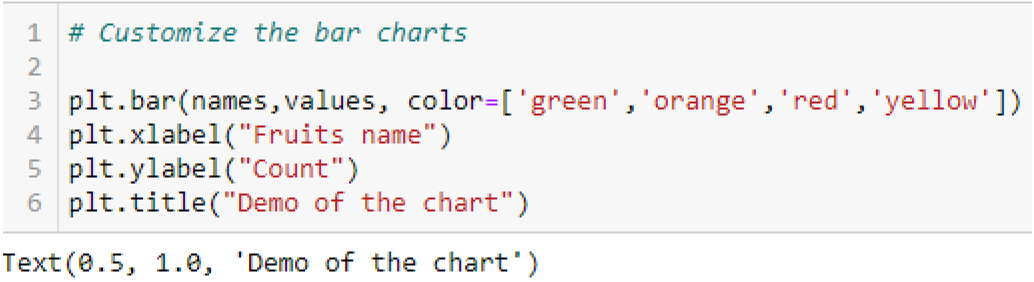
```python
# Bar plots

data = {'apple': 20,
        'mangoes' : 15,
        'lemon' : 30,
        'orange' : 10}
names = list(data.keys())
values = list(data.values())
print(names)
print(values)
```

```
['apple', 'mangoes', 'lemon', 'orange']
[20, 15, 30, 10]
```

```python
plt.bar(names,values)
```

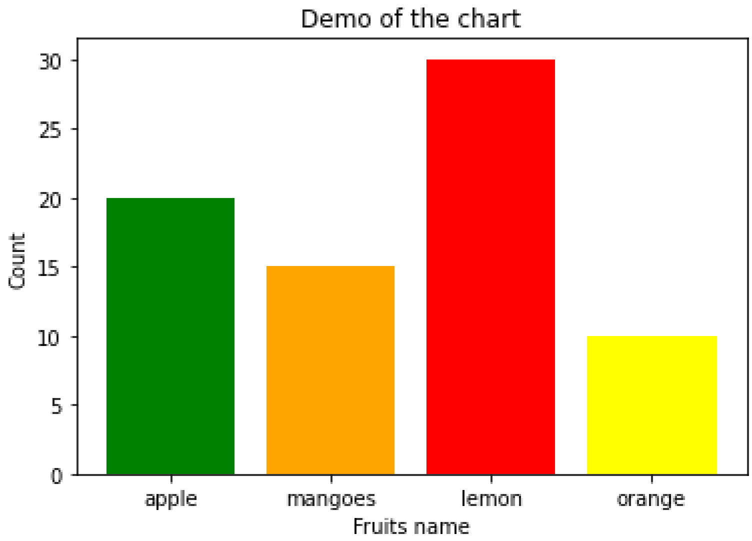
```
<BarContainer object of 4 artists>
```

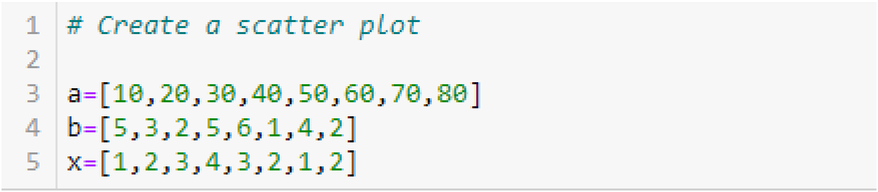




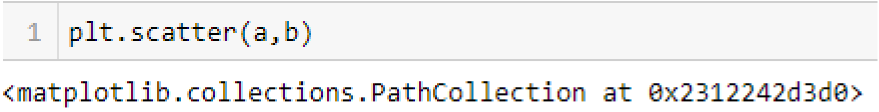
```
# Customize the bar charts

plt.bar(names,values, color=['green','orange','red','yellow'])
plt.xlabel("Fruits name")
plt.ylabel("Count")
plt.title("Demo of the chart")
```

```
Text(0.5, 1.0, 'Demo of the chart')
```



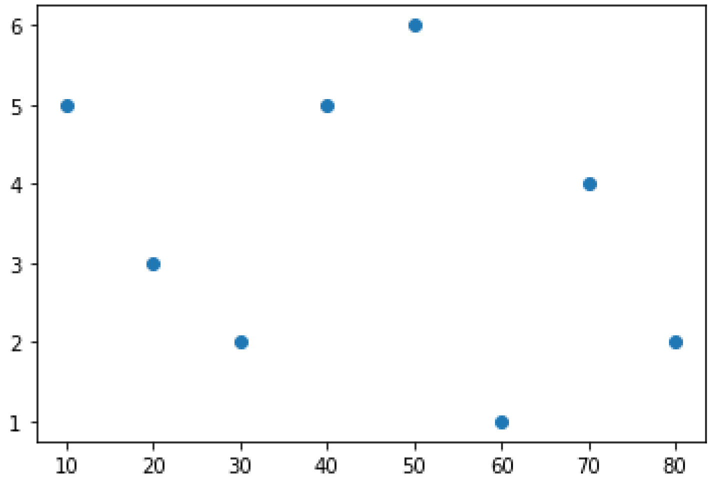


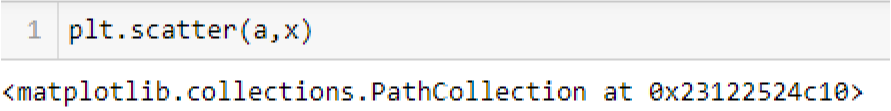
```
# Create a scatter plot

a=[10,20,30,40,50,60,70,80]
b=[5,3,2,5,6,1,4,2]
x=[1,2,3,4,3,2,1,2]
```



```
plt.scatter(a,b)
```

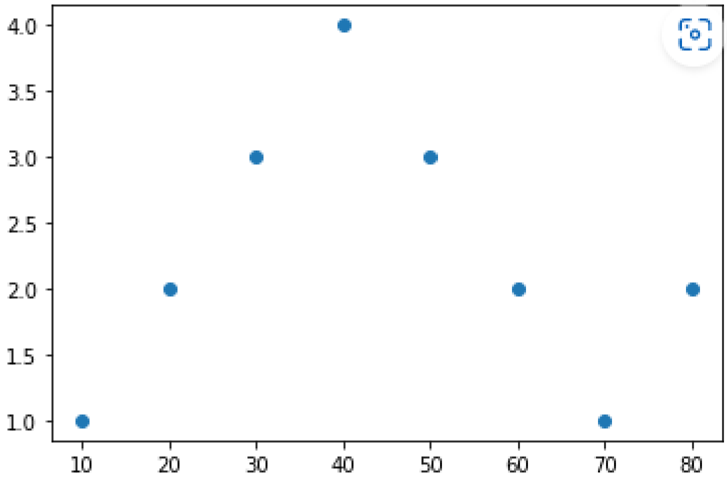
```
<matplotlib.collections.PathCollection at 0x2312242d3d0>
```

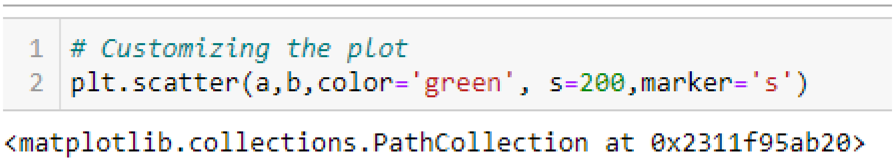




```
plt.scatter(a,x)
```

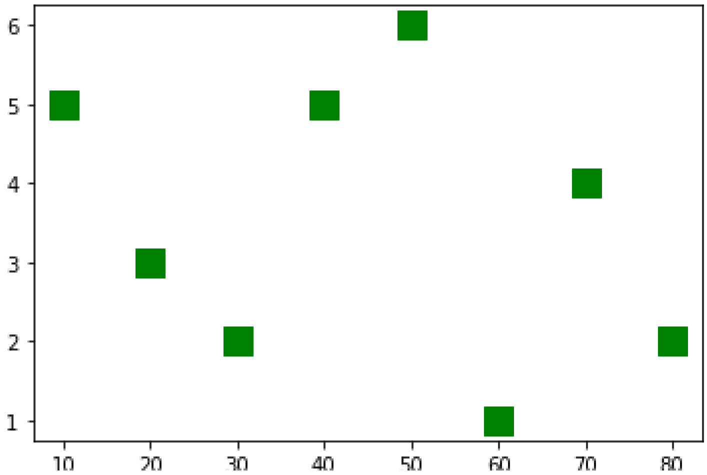
```
<matplotlib.collections.PathCollection at 0x23122524c10>
```

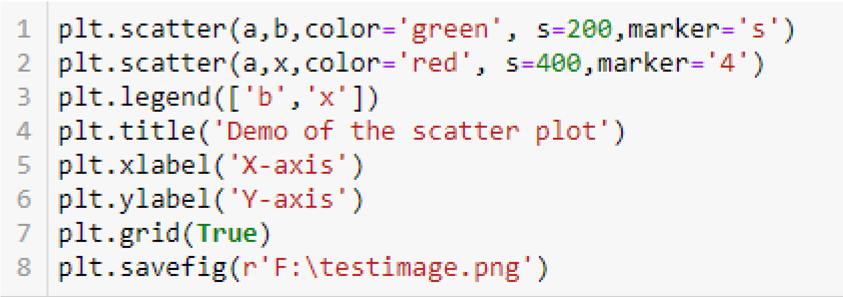




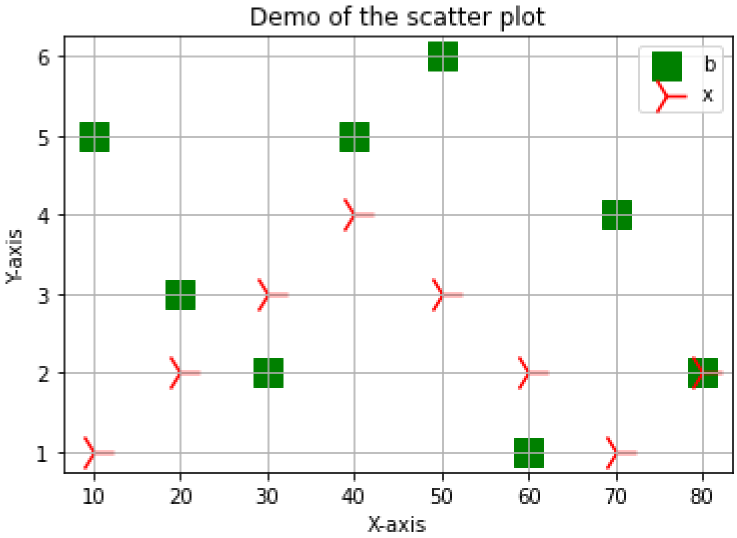
```
# Customizing the plot
plt.scatter(a,b,color='green', s=200,marker='s')
```

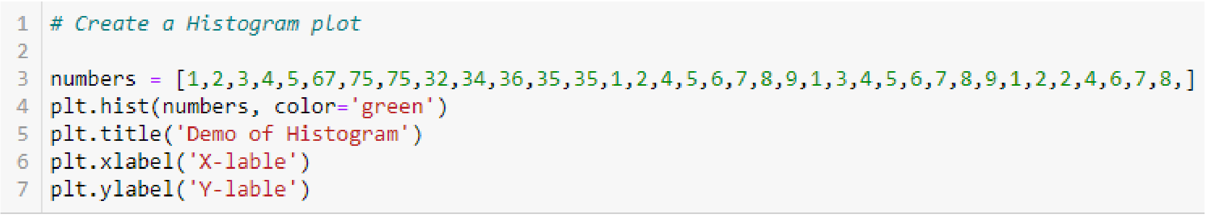
```
<matplotlib.collections.PathCollection at 0x2311f95ab20>
```





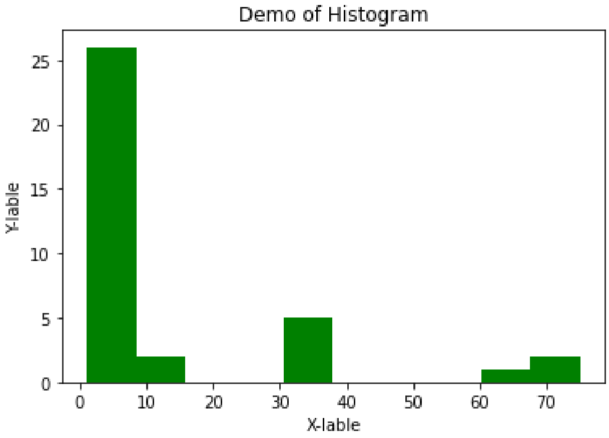
```python
plt.scatter(a,b,color='green', s=200,marker='s')
plt.scatter(a,x,color='red', s=400,marker='4')
plt.legend(['b','x'])
plt.title('Demo of the scatter plot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.grid(True)
plt.savefig(r'F:\testimage.png')
```

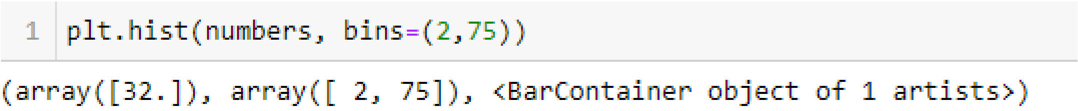




```python
# Create a Histogram plot

numbers = [1,2,3,4,5,67,75,75,32,34,36,35,35,1,2,4,5,6,7,8,9,1,3,4,5,6,7,8,9,1,2,2,4,6,7,8,]
plt.hist(numbers, color='green')
plt.title('Demo of Histogram')
plt.xlabel('X-lable')
plt.ylabel('Y-lable')
```

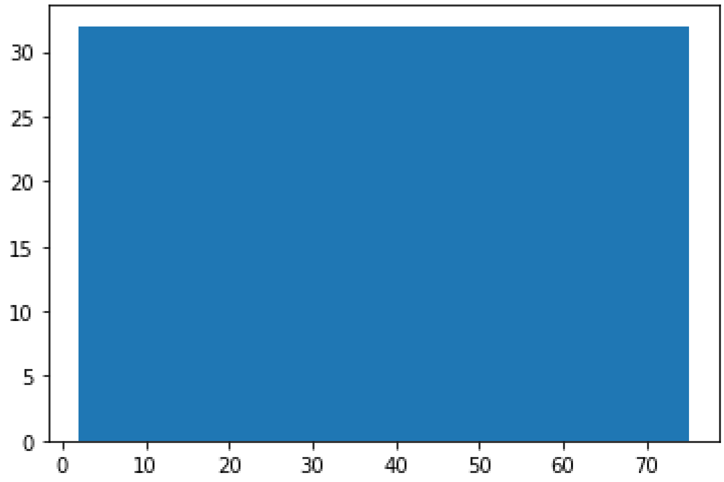
```
Text(0, 0.5, 'Y-lable')
```

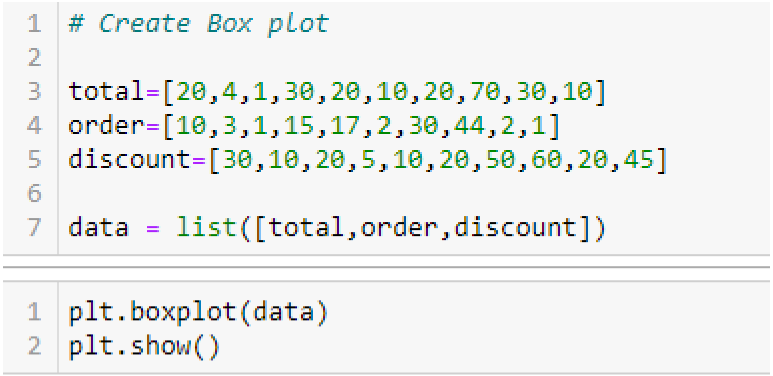




```
plt.hist(numbers, bins=(2,75))
```

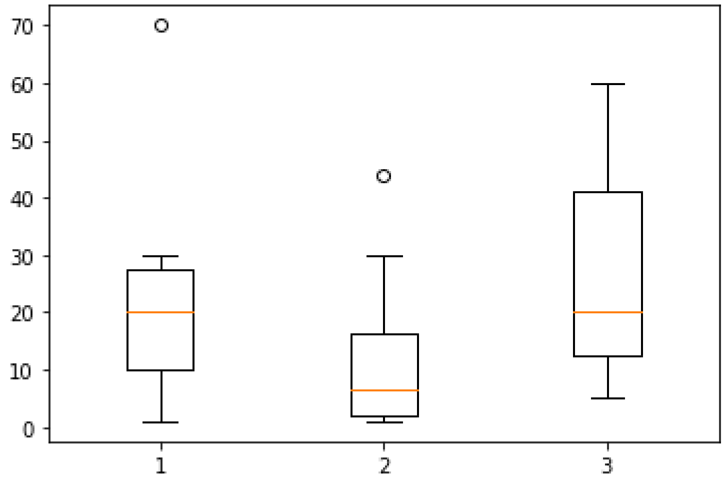
```
(array([32.]), array([ 2, 75]), <BarContainer object of 1 artists>)
```

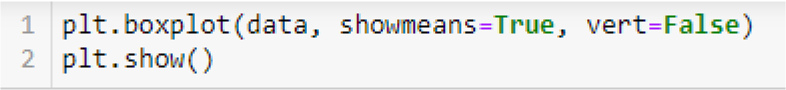




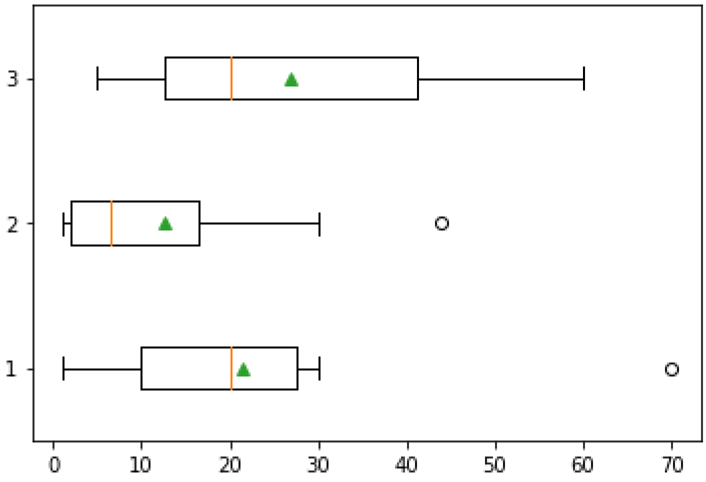
```
# Create Box plot

total=[20,4,1,30,20,10,20,70,30,10]
order=[10,3,1,15,17,2,30,44,2,1]
discount=[30,10,20,5,10,20,50,60,20,45]

data = list([total,order,discount])
```

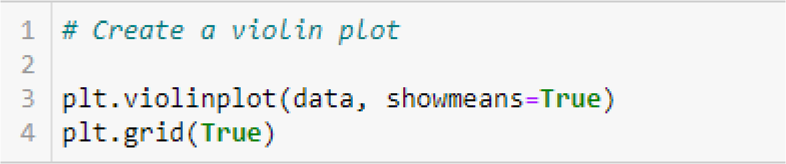
```
plt.boxplot(data)
plt.show()
```



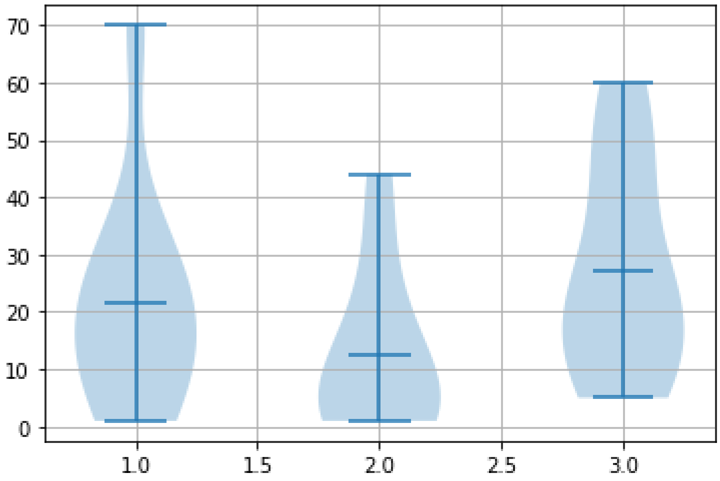


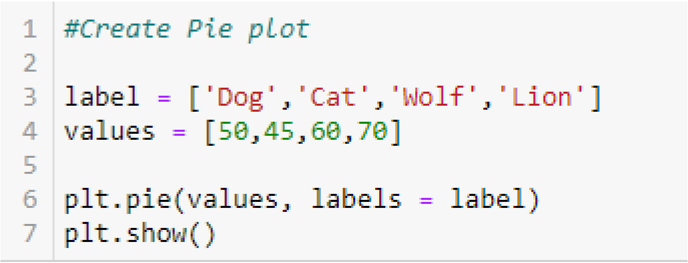
```
plt.boxplot(data, showmeans=True, vert=False)
plt.show()
```



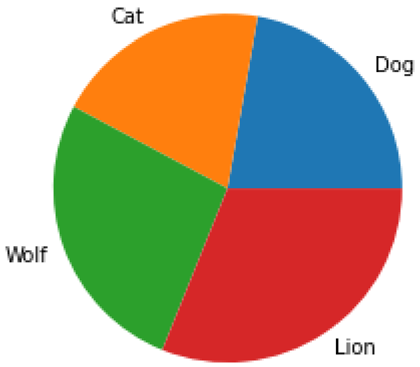


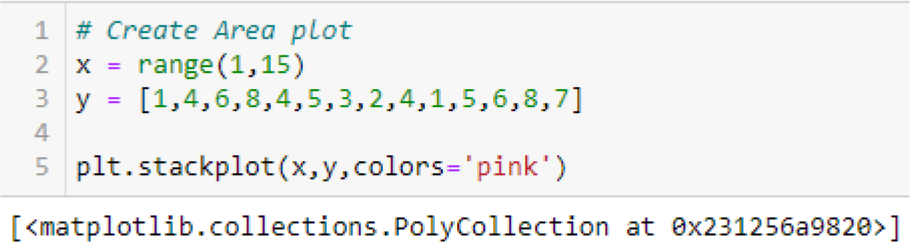
```
# Create a violin plot

plt.violinplot(data, showmeans=True)
plt.grid(True)
```





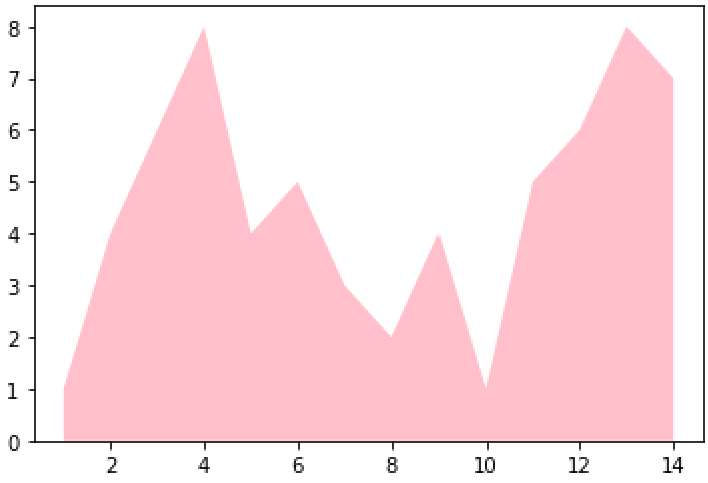
```
#Create Pie plot

label = ['Dog','Cat','Wolf','Lion']
values = [50,45,60,70]

plt.pie(values, labels = label)
plt.show()
```





```
# Create Area plot
x = range(1,15)
y = [1,4,6,8,4,5,3,2,4,1,5,6,8,7]

plt.stackplot(x,y,colors='pink')
```

```
[<matplotlib.collections.PolyCollection at 0x231256a9820>]
```



8
7
6
5
4
3
2
1
0
2
4
6
8
10
12
14