

Matplotlib

Matplotlib is a python library used for data visualization

Line Plot

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

In [2]: #pyplot is the sub module within the matplotlib library for plot diagram.

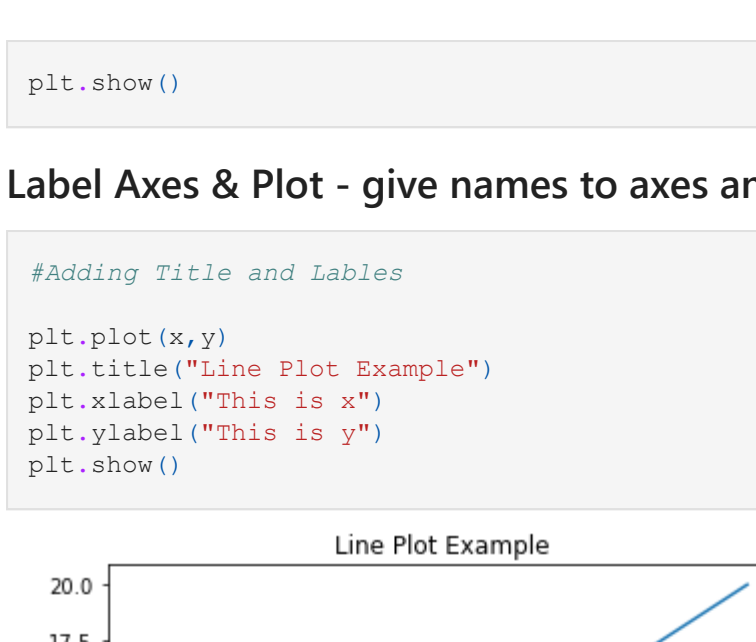
In [3]: x = np.arange(1,11)
x

Out[3]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])

In [4]: y = x*2
y

Out[4]: array([ 2,  4,  6,  8, 10, 12, 14, 16, 18, 20])

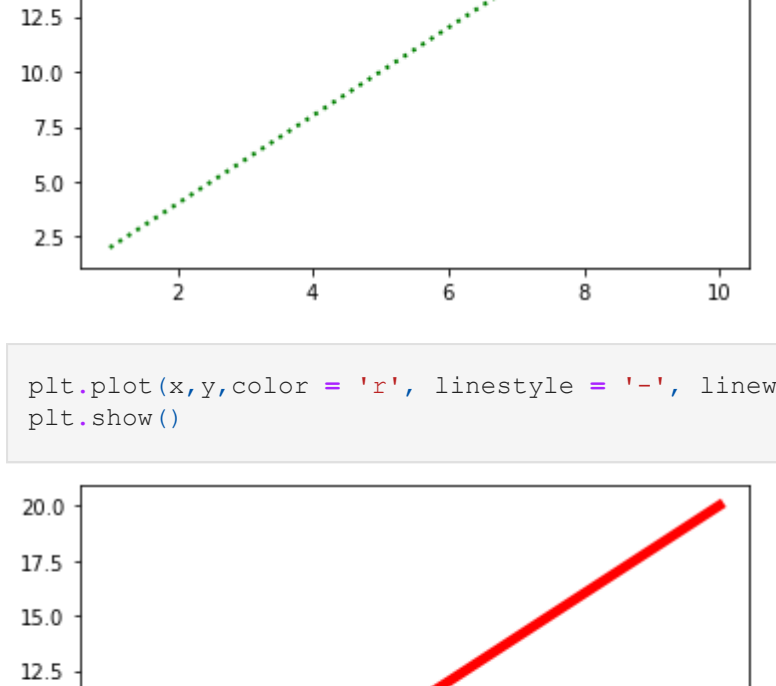
In [5]: plt.plot(x,y)
plt.show()
```



Label Axes & Plot - give names to axes and plot

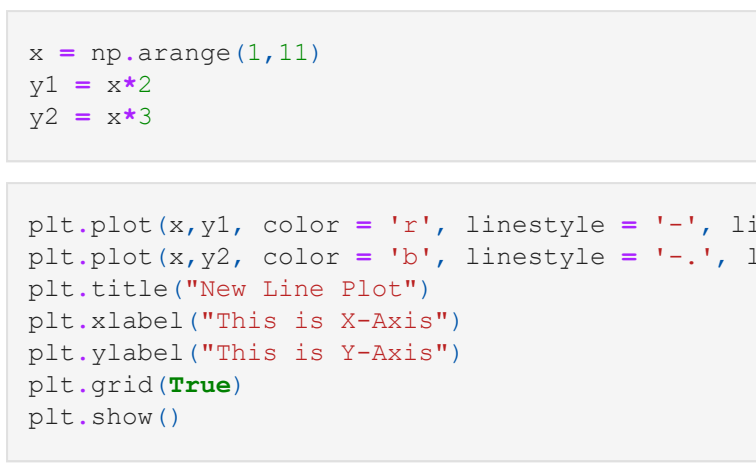
```
In [7]: #Adding Title and Labels

plt.plot(x,y)
plt.title("Line Plot Example")
plt.xlabel("This is x")
plt.ylabel("This is y")
plt.show()
```

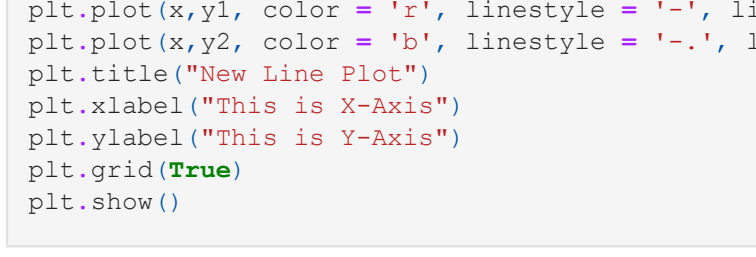


Changing line aesthetics - Line style, line colour, line width, etc.

```
In [8]: plt.plot(x,y,color = 'g', linestyle = ':', linewidth = 2)
plt.show()
```



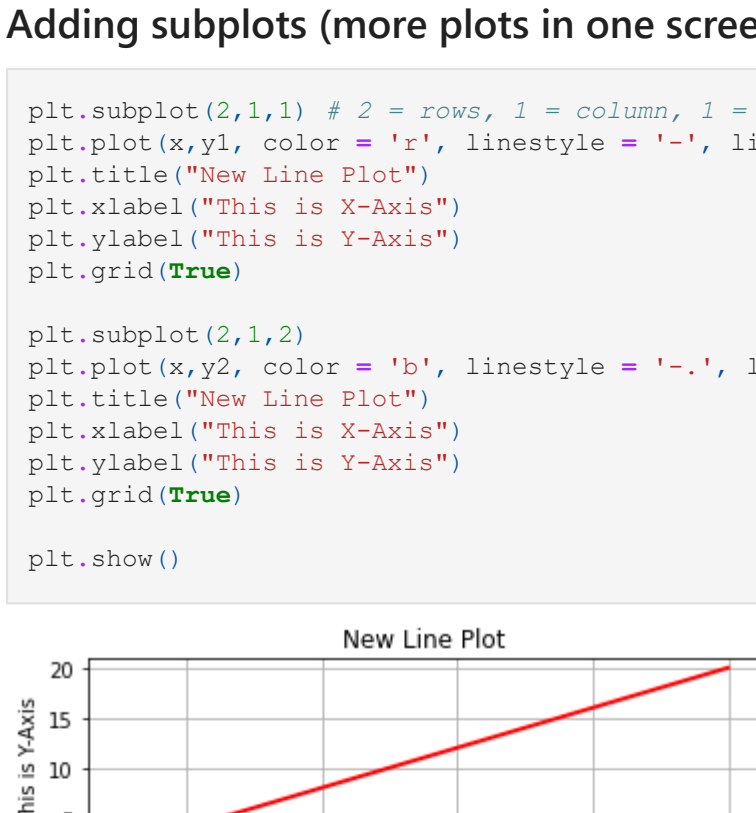
```
In [9]: plt.plot(x,y,color = 'r', linestyle = '-', linewidth = 5)
plt.show()
```



Adding two lines in the same plot

```
In [10]: x = np.arange(1,11)
y1 = x*2
y2 = x*3

In [13]: plt.plot(x,y1, color = 'r', linestyle = '-', linewidth = 2)
plt.plot(x,y2, color = 'b', linestyle = '--', linewidth = 4)
plt.title("New Line Plot")
plt.xlabel("This is X-Axis")
plt.ylabel("This is Y-Axis")
plt.grid(True)
plt.show()
```

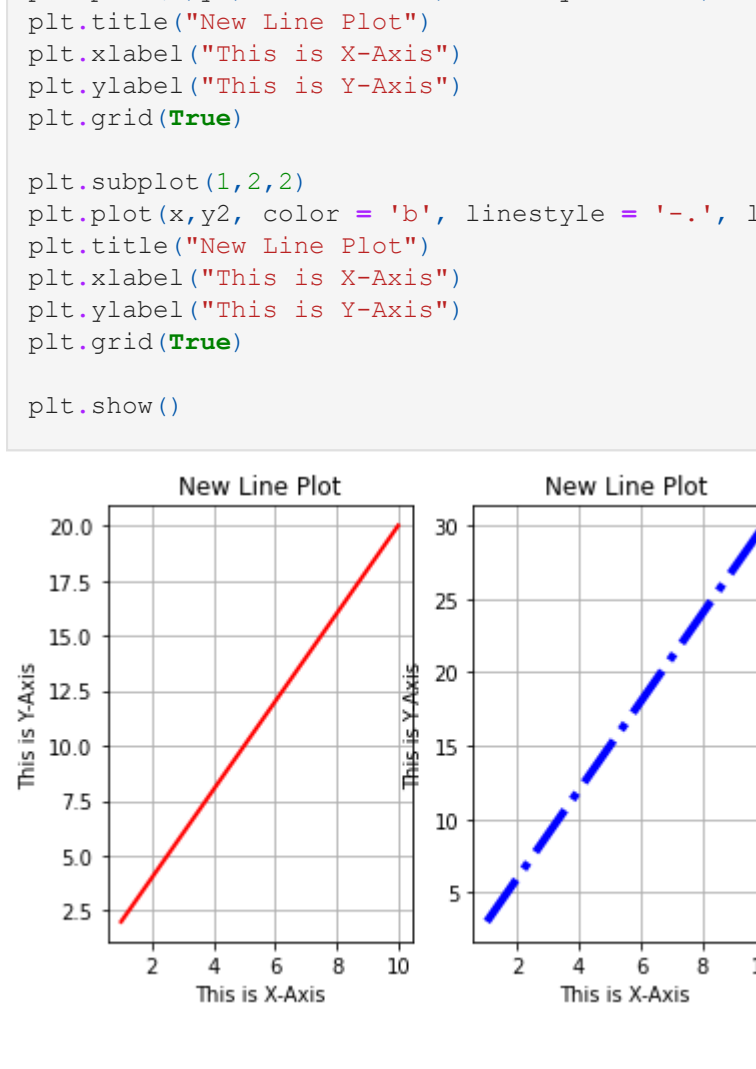


Adding subplots (more plots in one screen)

```
In [16]: plt.subplot(2,1,1) # 2 = rows, 1 = column, 1 = index
plt.plot(x,y1, color = 'r', linestyle = '-', linewidth = 2)
plt.title("New Line Plot")
plt.xlabel("This is X-Axis")
plt.ylabel("This is Y-Axis")
plt.grid(True)

plt.subplot(2,1,2)
plt.plot(x,y2, color = 'b', linestyle = '--', linewidth = 4)
plt.title("New Line Plot")
plt.xlabel("This is X-Axis")
plt.ylabel("This is Y-Axis")
plt.grid(True)

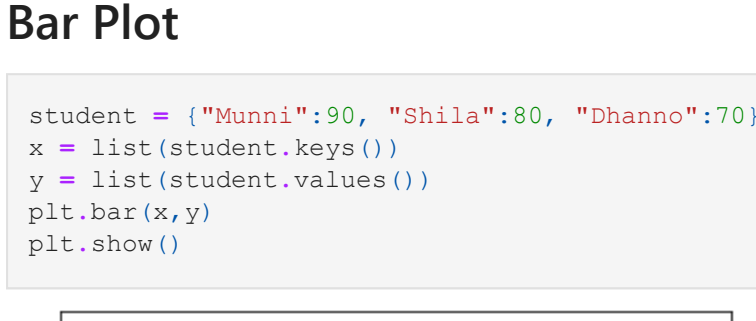
plt.show()
```



```
In [29]: plt.subplot(1,2,1) # 1 = row, 2 = column, 2 = index
plt.plot(x,y1, color = 'r', linestyle = '-', linewidth = 2)
plt.title("New Line Plot")
plt.xlabel("This is X-Axis")
plt.ylabel("This is Y-Axis")
plt.grid(True)

plt.subplot(1,2,2)
plt.plot(x,y2, color = 'b', linestyle = '--', linewidth = 4)
plt.title("New Line Plot")
plt.xlabel("This is X-Axis")
plt.ylabel("This is Y-Axis")
plt.grid(True)

plt.show()
```



Bar Plot

```
In [18]: student = {"Munni":90, "Shila":80, "Dhanno":70}
x = list(student.keys())
y = list(student.values())
plt.bar(x,y)
plt.show()
```



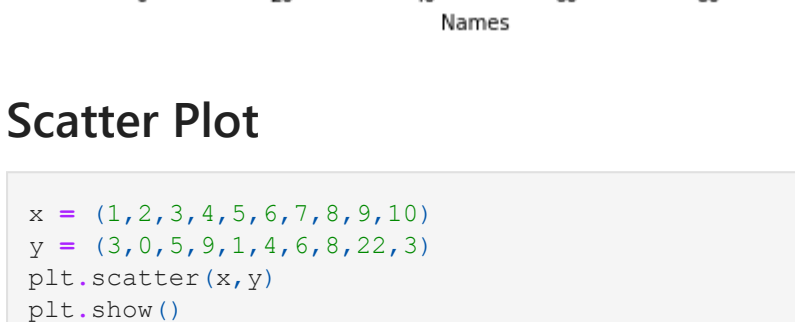
Adding Tiltles and Label

```
In [24]: plt.bar(x,y)
plt.title("Bar Plot Example")
plt.xlabel("Names")
plt.ylabel("Marks")
plt.grid(False) #We can pass false if we dont want grids in our any types of plot
plt.show()
```



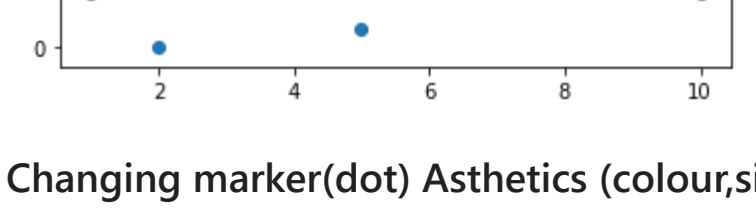
Horizontal Bar Plot

```
In [25]: plt.barh(x,y, color = 'r')
plt.title("Bar Plot Example")
plt.xlabel("Names")
plt.ylabel("Marks")
plt.show()
```



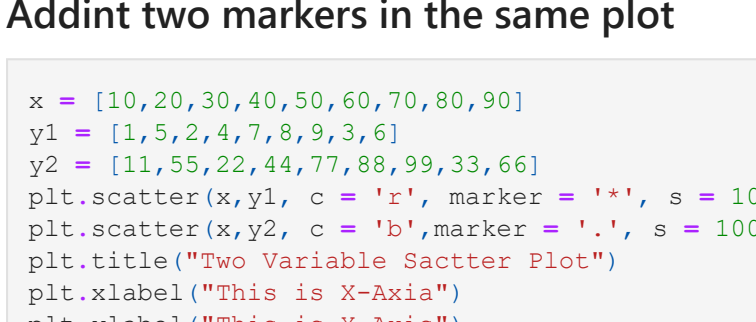
Scatter Plot

```
In [28]: x = (1,2,3,4,5,6,7,8,9,10)
y = (3,0,5,9,1,4,6,8,22,3)
plt.scatter(x,y)
plt.show()
```



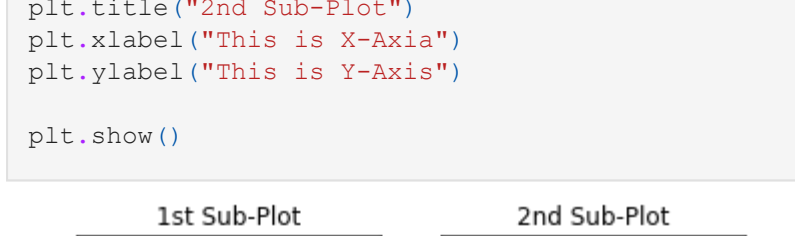
Changing marker(dot) Aesthetics (colour,size,type)

```
In [41]: x = (1,2,3,4,5,6,7,8,9,10)
y = (1,5,2,4,7,8,9,3,6)
plt.scatter(x,y,marker = '^', c = 'r', s = 100)
plt.show()
```



Addit two markers in the same plot

```
In [39]: x = [10,20,30,40,50,60,70,80,90]
y1 = [1,2,3,4,5,6,7,8,9]
y2 = [11,55,22,44,77,88,99,33,66]
plt.scatter(x,y1, c = 'r', marker = '^', s = 100)
plt.scatter(x,y2, c = 'b', marker = '+', s = 100)
plt.title("Two Variable Sactter Plot")
plt.xlabel("This is X-Axis")
plt.ylabel("This is Y-Axis")
plt.show()
```




Adding more than one plots(sub plots) in one screen

```
In [43]: x = [10,20,30,40,50,60,70,80,90]
y1 = [1,2,3,4,7,8,9,3,6]
y2 = [11,55,22,44,77,88,99,33,66]

plt.subplot(1,2,1)
plt.scatter(x,y1, c = 'r', marker = '^', s = 100)
plt.title("1st Sub-Plot")
plt.xlabel("This is X-Axis")
plt.ylabel("This is Y-Axis")

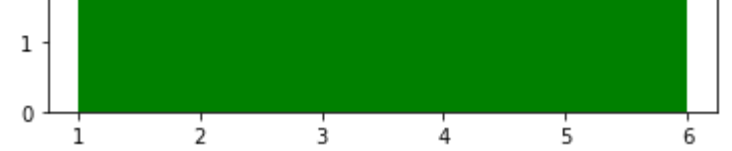
plt.subplot(1,2,2)
plt.scatter(x,y2, c = 'b', marker = '+', s = 100)
plt.title("2nd Sub-Plot")
plt.xlabel("This is X-Axis")
plt.ylabel("This is Y-Axis")

plt.show()
```



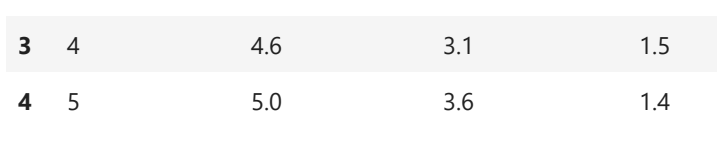
Histogram

```
In [52]: data = [1,1,1,2,2,3,3,3,4,4,4,5,5,5,6]
plt.hist(data, color = 'g', bins = 3)
plt.show()
```



Adding 'Bins'

```
In [53]: data = [1,1,1,2,2,3,3,3,4,4,4,5,5,5,6]
plt.hist(data, color = 'g', bins = 3)
plt.show()
```

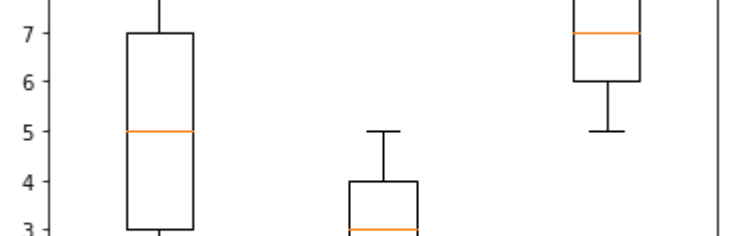


Working with a data set

```
In [61]: import pandas as pd
iris = pd.read_csv('iris.csv')
iris.head()
```

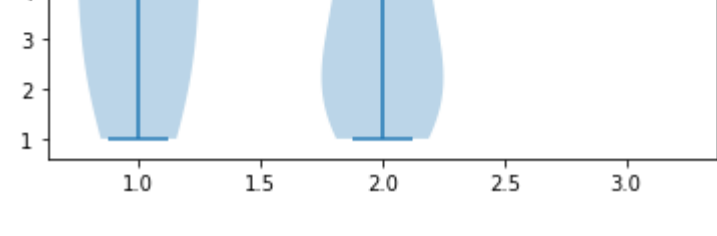
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```
In [72]: plt.hist(iris['SepalLengthCm'], color = 'r', bins = 100)
plt.show()
```



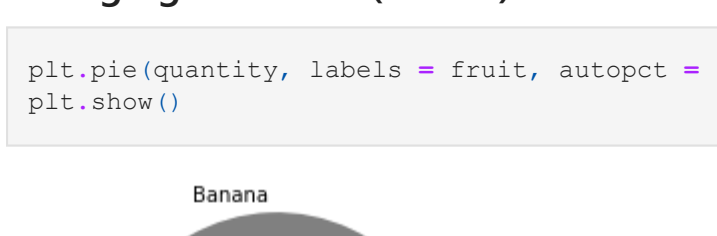
Box-Plot

```
In [74]: one = [1,2,3,4,5,6,7,8,9]
two = [1,2,3,5,4,3,2,1,5]
three = [9,7,6,8,9,6,5,7,9]
d1 = list((one,two,three))
plt.boxplot(d1)
plt.show()
```



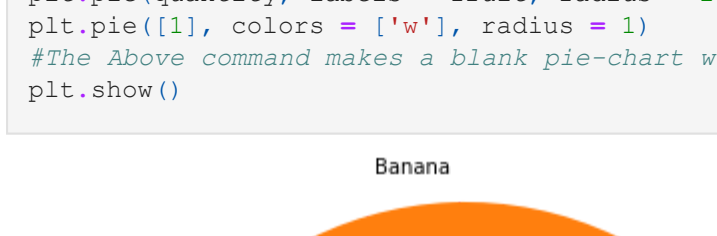
Violin-Plot

```
In [75]: one = [1,2,3,4,5,6,7,8,9]
two = [1,2,3,5,4,3,2,1,5]
three = [9,7,6,8,9,6,5,7,9]
d1 = list((one,two,three))
plt.violinplot(d1)
plt.show()
```




Pie-Chart

```
In [76]: fruit = ['Apple','Banana','Mango','Cherry']
quantity = [20,60,45,70]
plt.pie(quantity, labels = fruit)
plt.show()
```



Changing Asthetics (colour)

```
In [82]: plt.pie(quantity, labels = fruit, autopct = '%0.1f%%', colors = ['yellow','grey','pink','turquoise'])
plt.show()
```



DoughNut-Chart (modified pie chart)

```
In [92]: fruit = ['Apple','Banana','Mango','Cherry']
quantity = [20,60,45,70]
plt.pie(quantity, labels = fruit, radius = 2)
plt.pie(1, colors = ['w'], radius = 1)
#The Above command makes a blank pie-chart with white colour whoes radius is smaller than the outer circle's radius
plt.show()
```

