Dropping Rows

greatlearning
Learning for Life

```
iris.drop([1,2,3],axis=0)
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
0	5.1	3.5	1.4	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
5	5.4	3.9	1.7	0.4	setosa
6	4.6	3.4	1.4	0.3	setosa
7	5.0	3.4	1.5	0.2	setosa

More Pandas Functions

greatlearning
Learning for Life

Mean

```
In [5]: iris.mean()

Out[5]: Sepal.Length    5.843333
        Sepal.Width     3.057333
        Petal.Length    3.758000
        Petal.Width     1.199333
        dtype: float64
```

Minimum

```
In [13]: iris.min()

Out[13]: Sepal.Length        4.3
         Sepal.Width           2
         Petal.Length          1
         Petal.Width         0.1
         Species          setosa
         dtype: object
```

Median

```
In [6]: iris.median()

Out[6]: Sepal.Length    5.80
        Sepal.Width     3.00
        Petal.Length    4.35
        Petal.Width     1.30
        dtype: float64
```

Maximum

```
In [14]: iris.max()

Out[14]: Sepal.Length         7.9
         Sepal.Width          4.4
         Petal.Length         6.9
         Petal.Width          2.5
         Species        virginica
         dtype: object
```

More Pandas Functions

greatlearning
Learning for Life

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
def half(s):
    return s*0.5
```

```
iris[['Sepal.Length','Petal.Length']].apply(half)
```

	Sepal.Length	Petal.Length
0	2.55	0.70
1	2.45	0.70
2	2.35	0.65
3	2.30	0.75
4	2.50	0.70

More Pandas Functions

greatlearning
Learning for Life

Value_counts()

```
In [28]: iris['Species'].value_counts()

Out[28]: setosa        50
         virginica     50
         versicolor    50
         Name: Species, dtype: int64
```

sort_values()

```
In [29]: iris.sort_values(by='Sepal.Length')
```

Out[29]:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
13	4.3	3.0	1.1	0.1	setosa
42	4.4	3.2	1.3	0.2	setosa
38	4.4	3.0	1.3	0.2	setosa
8	4.4	2.9	1.4	0.2	setosa
41	4.5	2.3	1.3	0.3	setosa

Python Matplotlib

greatlearning
Learning for Life

Matplotlib is a python library used for data visualization

matplotlib

Line Plot

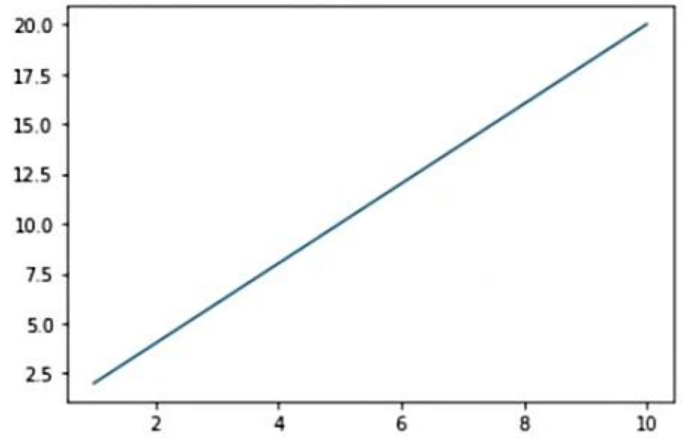
greatlearning
Learning for Life

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

```
In [2]: x=np.arange(1,11)
        x
Out[2]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

```
In [4]: y= 2*x
        y
Out[4]: array([ 2,  4,  6,  8, 10, 12, 14, 16, 18, 20])
```

```
In [6]: plt.plot(x,y)
        plt.show()
```

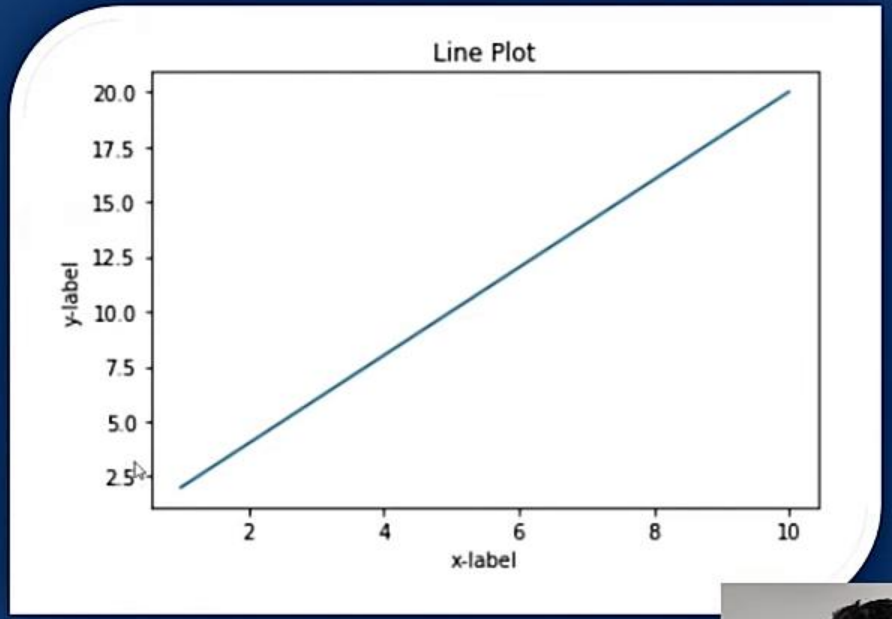


Line Plot

greatlearning
Learning for Life

Adding Title and Labels

```
In [8]: plt.plot(x,y)
        plt.title("Line Plot")
        plt.xlabel("x-label")
        plt.ylabel("y-label")
        plt.show()
```

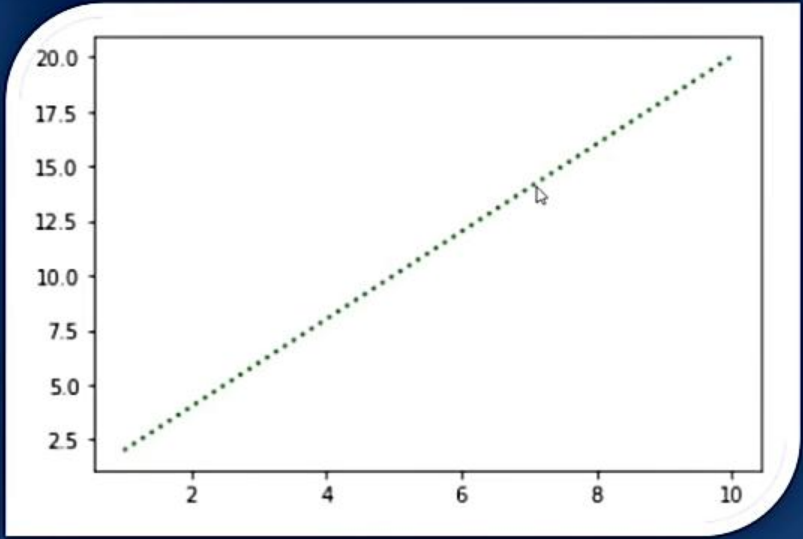


Line Plot

greatlearning
Learning for Life

Changing Line Aesthetics

```
In [10]: plt.plot(x,y,color='g',linestyle=':',linewidth=2)
         plt.show()
```



Line Plot

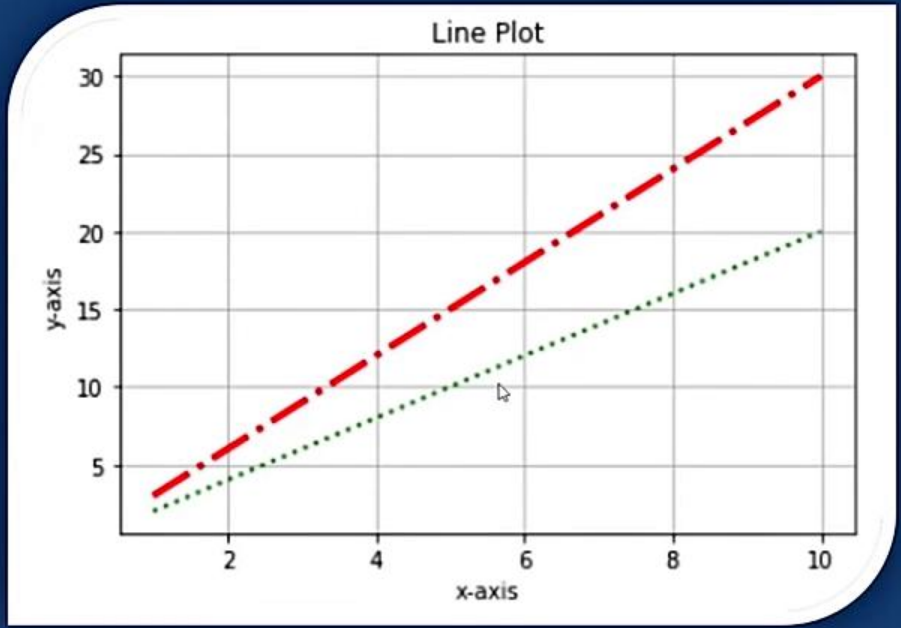
greatlearning
Learning for Life

Adding two lines in the same plot

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

```
In [11]: plt.plot(x,y1,color='g',linestyle=':',linewidth=2)
         plt.plot(x,y2,color='r',linestyle='-.',linewidth=3)
         plt.title("Line Plot")
         plt.xlabel("x-axis")
         plt.ylabel("y-axis")
         plt.grid(True)
         plt.show()
```

Line Plot



Line Plot

greatlearning
Learning for Life

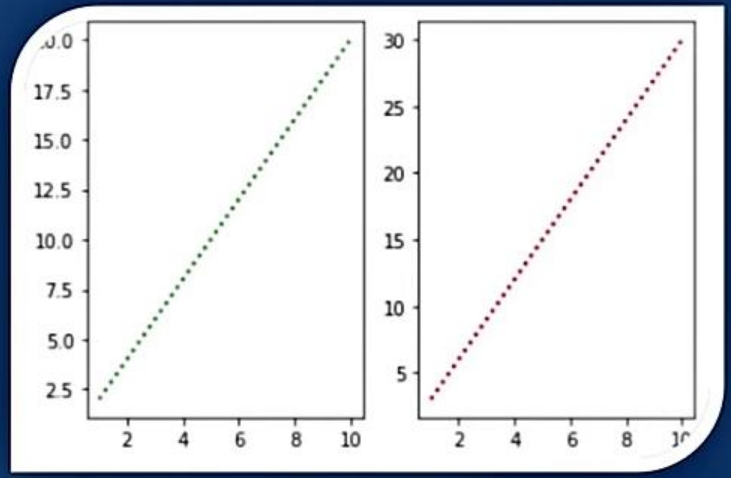
Adding sub-plots

```
x=np.arange(1,11)
y1=2*x
y2=3*x

plt.subplot(1,2,1)
plt.plot(x,y1,color='g',linestyle=':',linewidth=2)

plt.subplot(1,2,2)
plt.plot(x,y2,color='r',linestyle=':',linewidth=2)

plt.show()
```



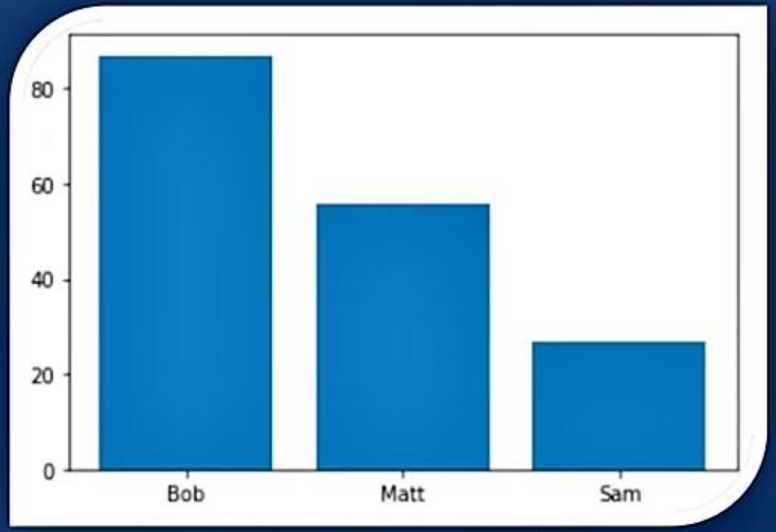
Bar Plot

greatlearning
Learning for Life

```
[39]: student = {"Bob":87,"Matt":56,"Sam":27}
```

```
In [40]: names = list(student.keys())
         values = list(student.values())
```

```
In [42]: plt.bar(names,values)
         plt.show()
```

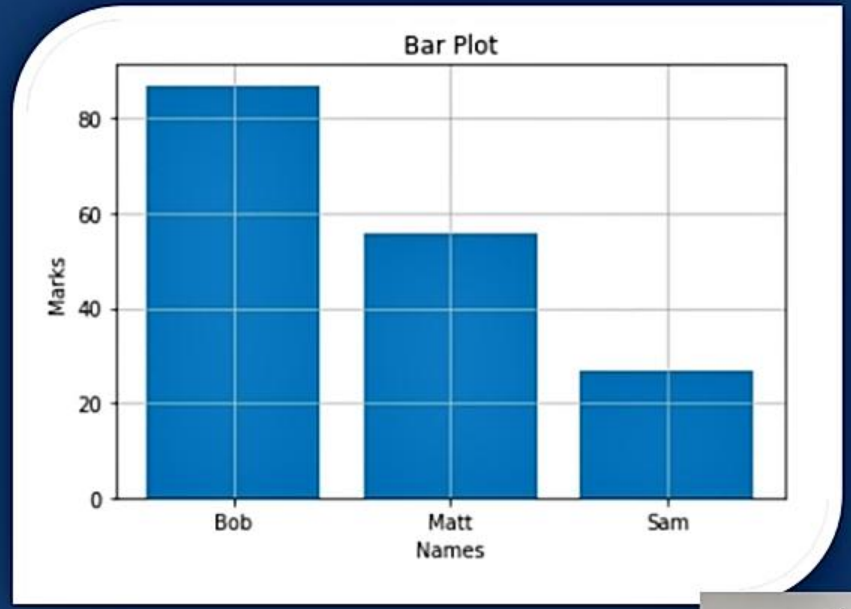


Bar Plot

greatlearning
Learning for Life

Adding Title and Labels

```
In [16]: plt.bar(names,values)
         plt.title("Bar Plot")
         plt.xlabel("Names")
         plt.ylabel("Marks")
         plt.grid(True)
         plt.show()
```

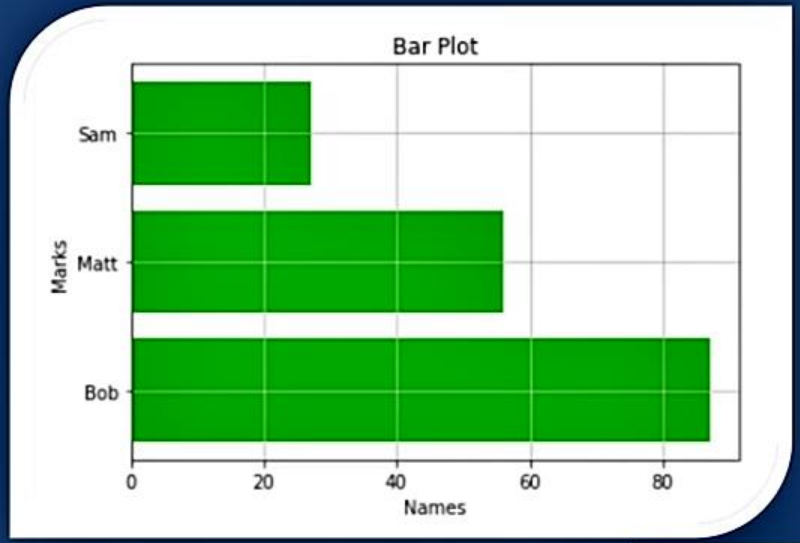


Horizontal Bar Plot

greatlearning
Learning for Life

Horizontal Bar Plot

```
In [44]: plt.barh(names,values,color='g')
         plt.title("Bar Plot")
         plt.xlabel("Names")
         plt.ylabel("Marks")
         plt.grid(True)
         plt.show()
```



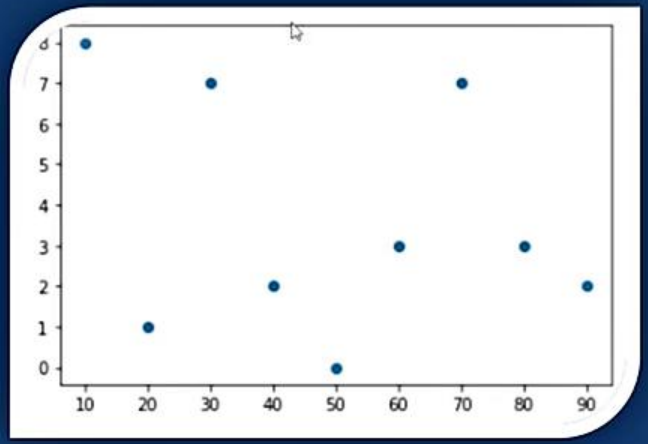
Scatter Plot

greatlearning
Learning for Life

Creating a basic scatter-plot

```
x=[10,20,30,40,50,60,70,80,90]
a=[8,1,7,2,0,3,7,3,2]

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

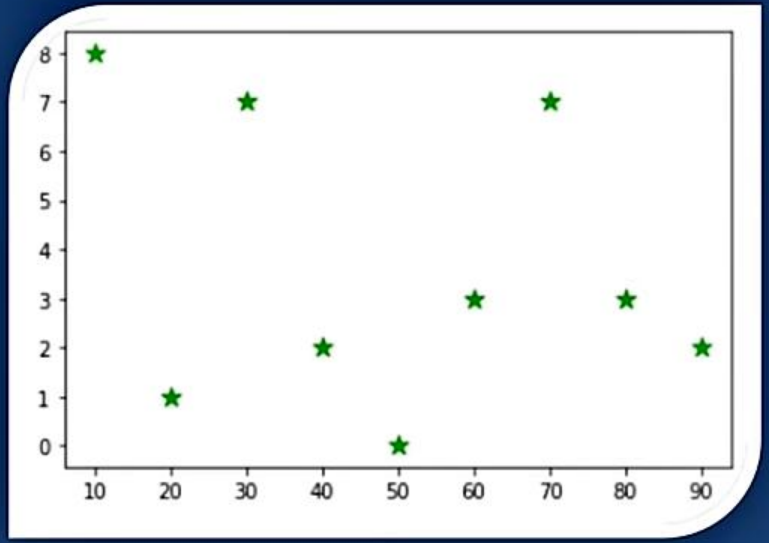


Scatter Plot

greatlearning
Learning for Life

Changing Mark Aesthetics

```
In [7]: x=[10,20,30,40,50,60,70,80,90]
        a=[8,1,7,2,0,3,7,3,2]
        plt.scatter(x,a,marker="*",c="g",s=100)
        plt.show()
```

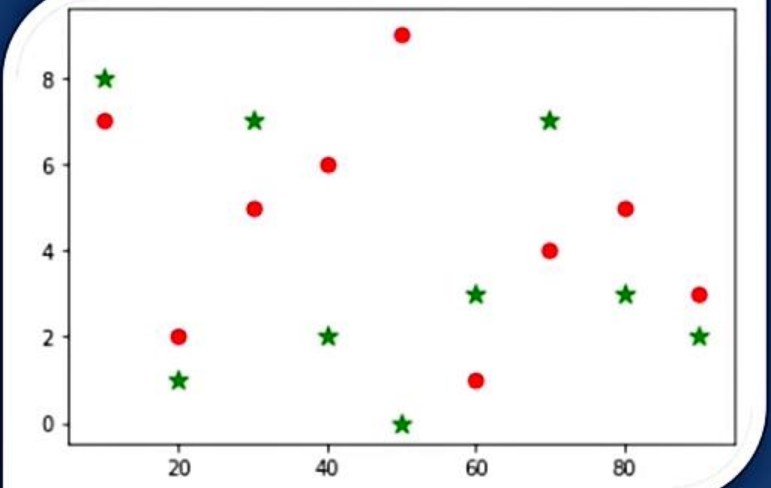


Scatter Plot

greatlearning
Learning for Life

Adding two markers in the same plot

```
In [10]: x=[10,20,30,40,50,60,70,80,90]
         a=[8,1,7,2,0,3,7,3,2]
         b=[7,2,5,6,9,1,4,5,3]
         plt.scatter(x,a,marker="*",c="g",s=100)
         plt.scatter(x,b,marker=".",c="r",s=200)
         plt.show()
```



Scatter Plot

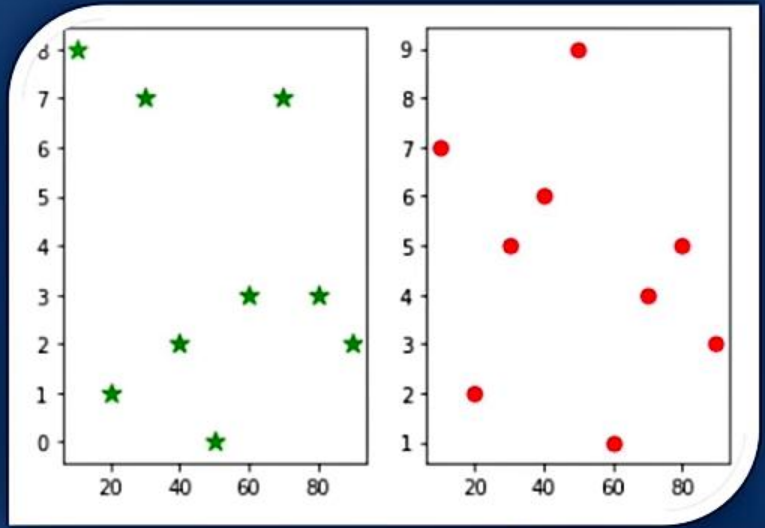
greatlearning
Learning for Life

Adding sub-plots

```
x=[10,20,30,40,50,60,70,80,90]
a=[8,1,7,2,0,3,7,3,2]
b=[7,2,5,6,9,1,4,5,3]

plt.subplot(1,2,1)
plt.scatter(x,a,marker="*",c="g",s=100)

plt.subplot(1,2,2)
plt.scatter(x,b,marker=".",c="r",s=200)
plt.show()
```



Histogram

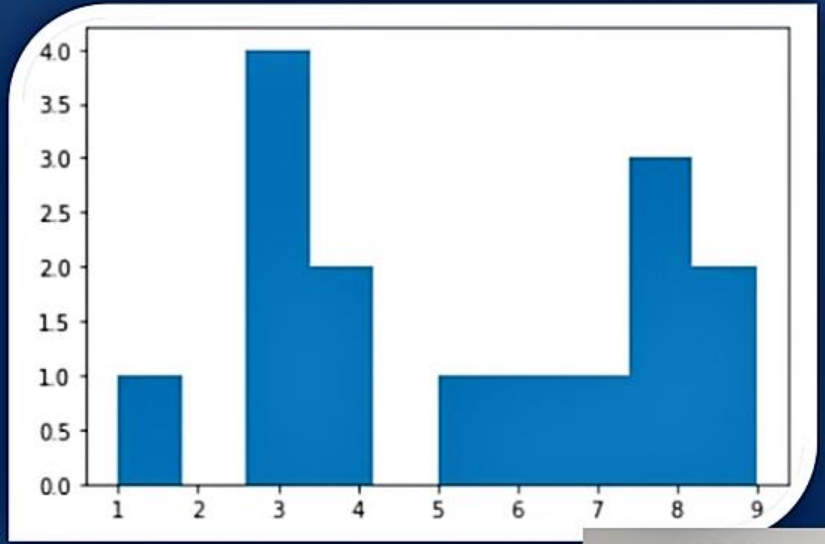
greatlearning
Learning for Life

Creating data

```
data = [1,3,3,3,3,9,9,5,4,4,8,8,8,6,7]
```

Making Histogram

```
plt.hist(data)
plt.show()
```

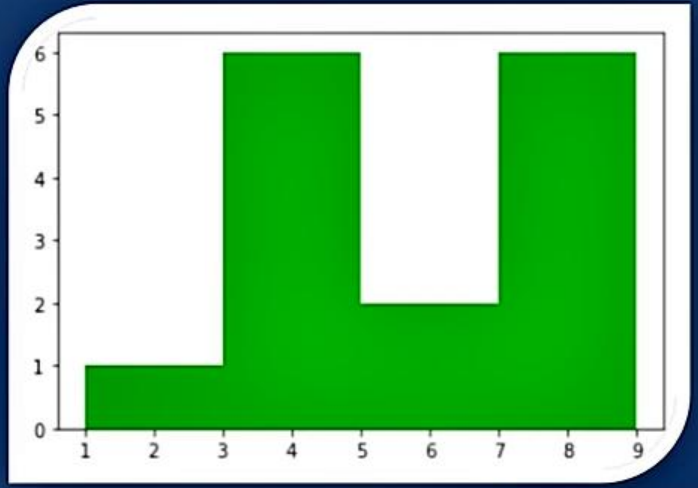


Histogram

greatlearning
Learning for Life

Changing Aesthetics

```
In [24]: plt.hist(data,color="g",bins=4)
         plt.show()
```



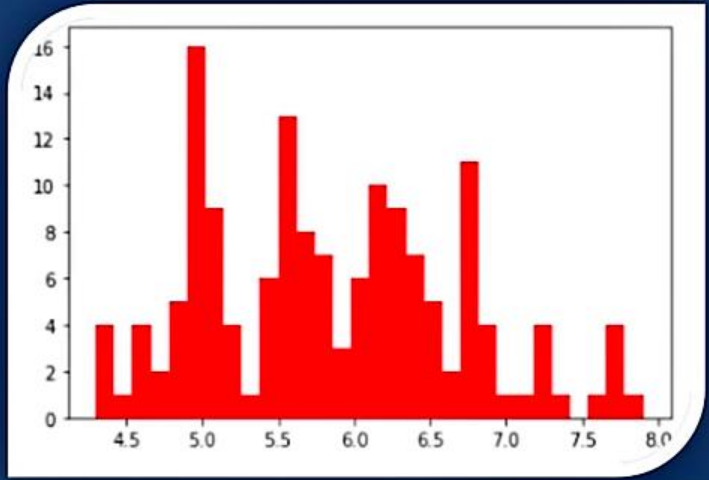
Histogram

greatlearning
Learning for Life

Working with a dataset

```
iris=pd.read_csv('iris.csv')
iris.head()
```

```
plt.hist(iris['Sepal.Length'],bins=30,color="r")
plt.show()
```



Box-Plot

greatlearning
Learning for Life

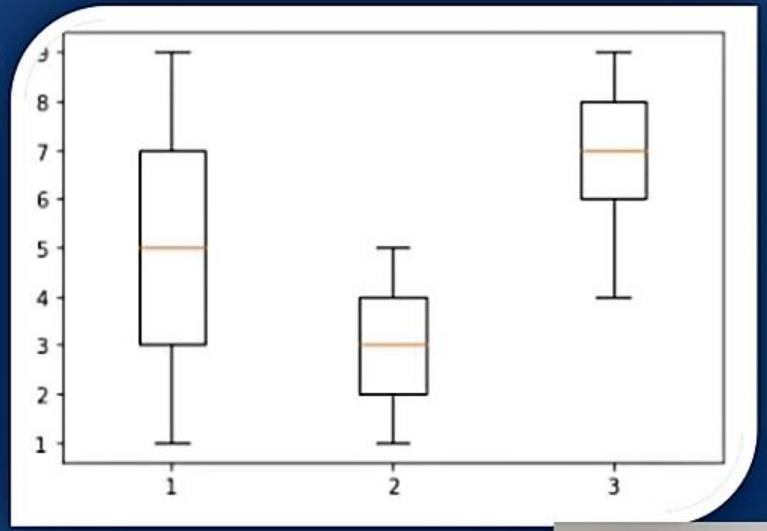
Creating data

```
one = [1,2,3,4,5,6,7,8,9]
two = [1,2,3,4,5,4,3,2,1]
three = [6,7,8,9,8,7,6,5,4]

data =  list([one,two,three])
```

Making Plot

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



Violin-Plot

greatlearning
Learning for Life

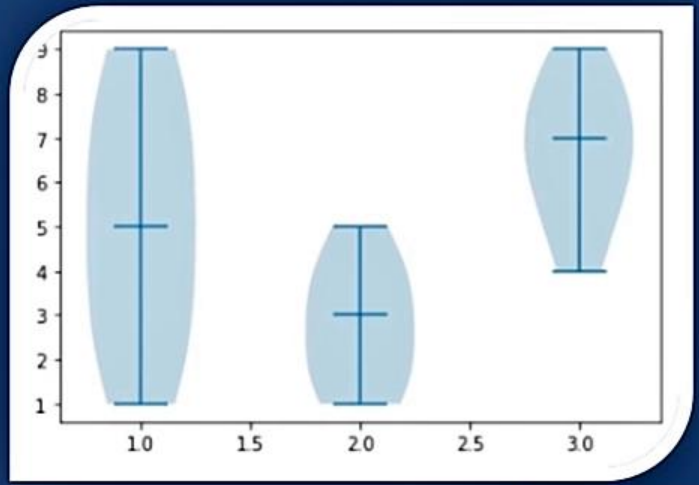
Creating data

```
one = [1,2,3,4,5,6,7,8,9]
two = [1,2,3,4,5,4,3,2,1]
three = [6,7,8,9,8,7,6,5,4]

data =  list([one,two,three])
```

Making Plot

```
plt.violinplot(data,showmedians=True)
plt.show()
```



Pie-Chart

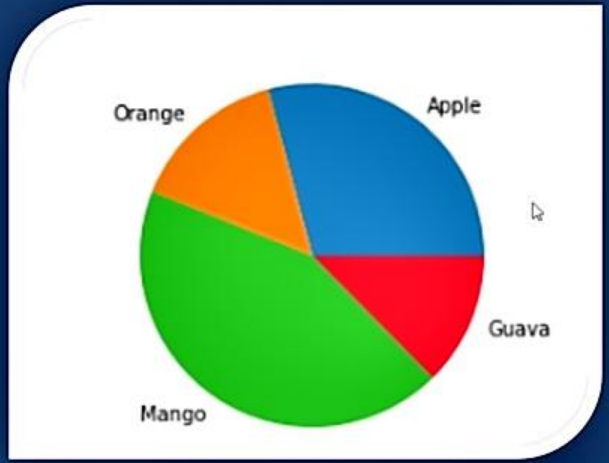
greatlearning
Learning for Life

Creating data

```
fruit = ['Apple','Orange','Mango','Guava']
quantity = [67,34,100,29]
```

Making Plot

```
plt.pie(quantity,labels=fruit)
plt.show()
```

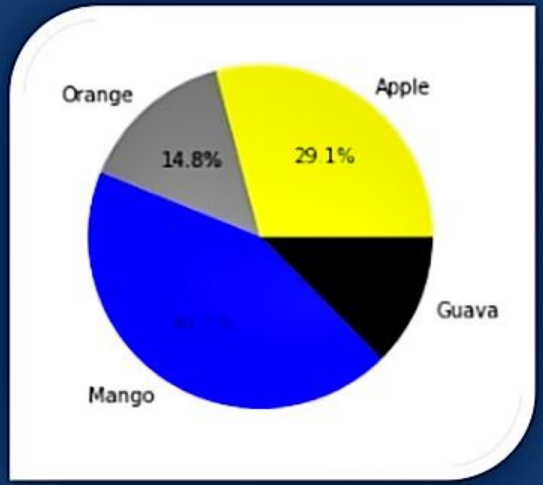


Pie-Chart

greatlearning
Learning for Life

Changing Aesthetics

```
plt.pie(quantity,labels=fruit,autopct='%0.1f%%'
        ,colors=['yellow','grey','blue','black'])
plt.show()
```



DoughNut-Chart

greatlearning
Learning for Life

Creating Data

```
fruit = ['Apple','Orange','Mango','Guava']
quantity = [67,34,100,29]
```

Making Plot

```
plt.pie(quantity,labels=fruit,radius=2)
plt.pie([1],colors=['w'],radius=1)
plt.show()
```