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

uses to import matplot and work with it we require pandas and numpy

```
x=np.linspace(0,5,11)

x

array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. ])
```

used for creating array btw range while deciding number of elements

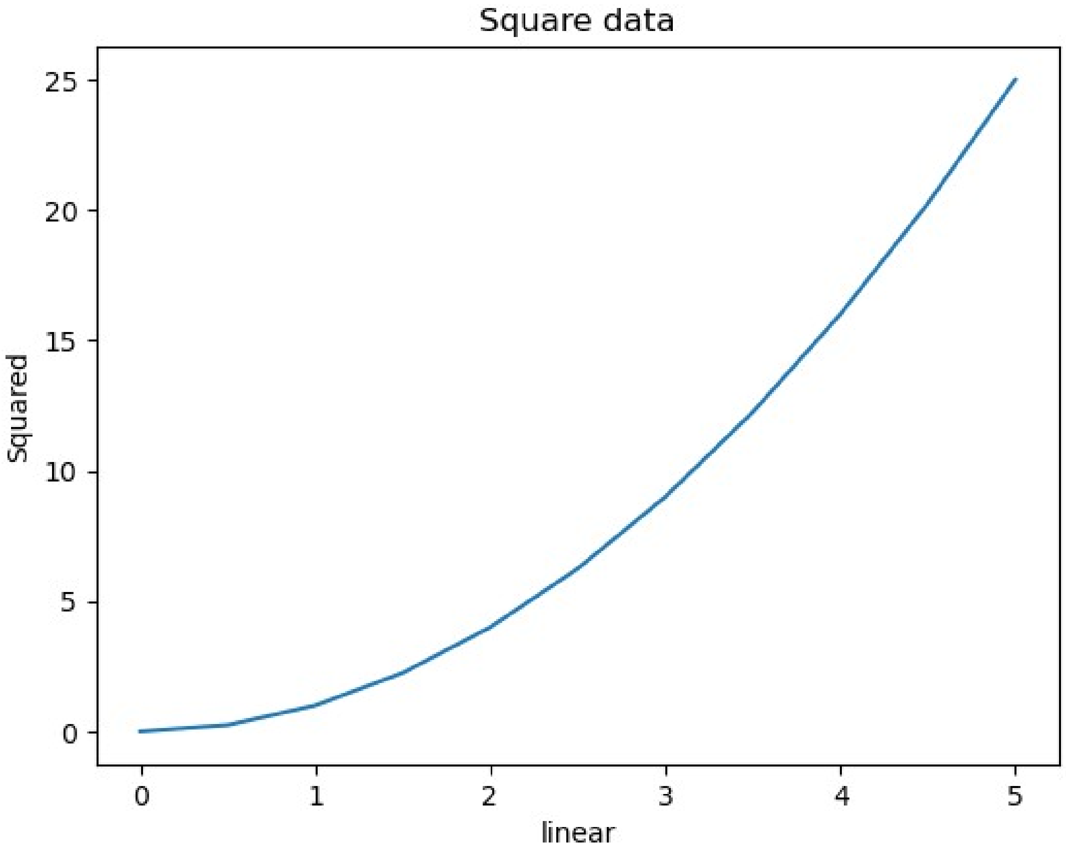
```
y=x**2
y

array([ 0.  ,  0.25,  1.  ,  2.25,  4.  ,  6.25,  9.  , 12.25, 16.  ,
       20.25, 25.  ])
```

plot used for ploting a 2d linear graph ,using x and y where it compares coresponding values of x and y to create a graph

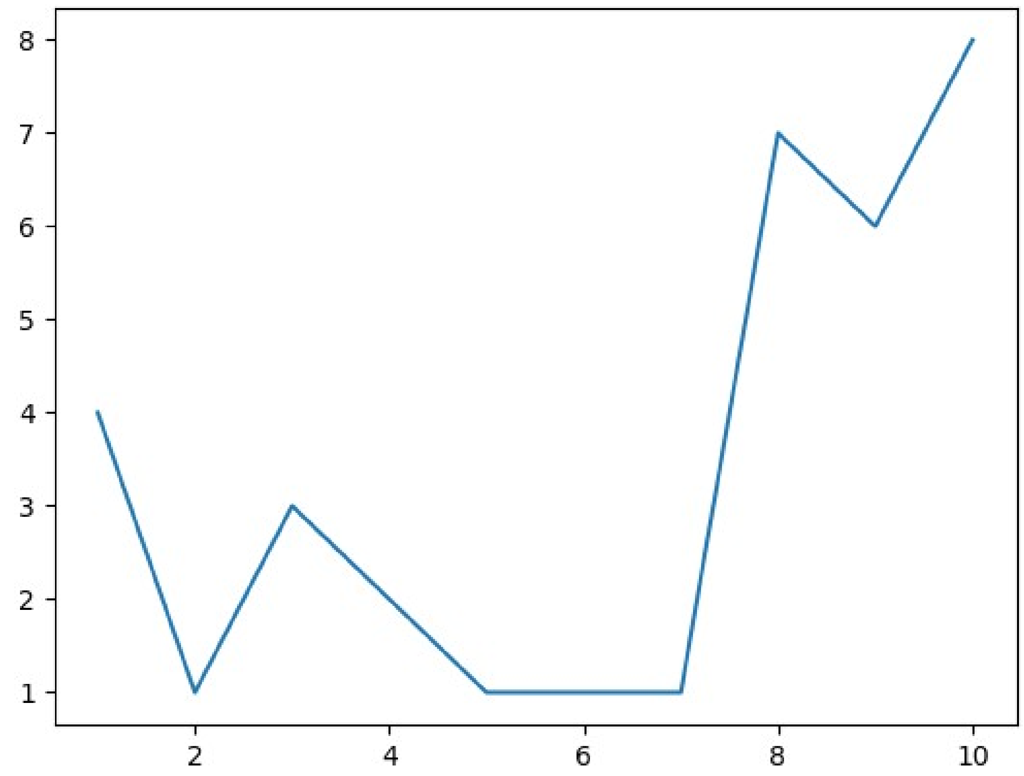
title is used for giveing the graph a tittle

LABEL(X,Y) IS USED FOR GIVING LABELS TO EACH AXIS

```
plt.plot(x,y)
plt.title("Square data")
plt.xlabel("linear")
plt.ylabel("Squared")

Text(0, 0.5, 'Squared')
```



```
a=np.array([1,2,3,4,5,6,7,8,9,10])
b=np.array([4,1,3,2,1,1,1,7,6,8])
c=a*2
d=b**3
plt.plot(a,b)

[<matplotlib.lines.Line2D at 0x1f4fddaee90>]
```

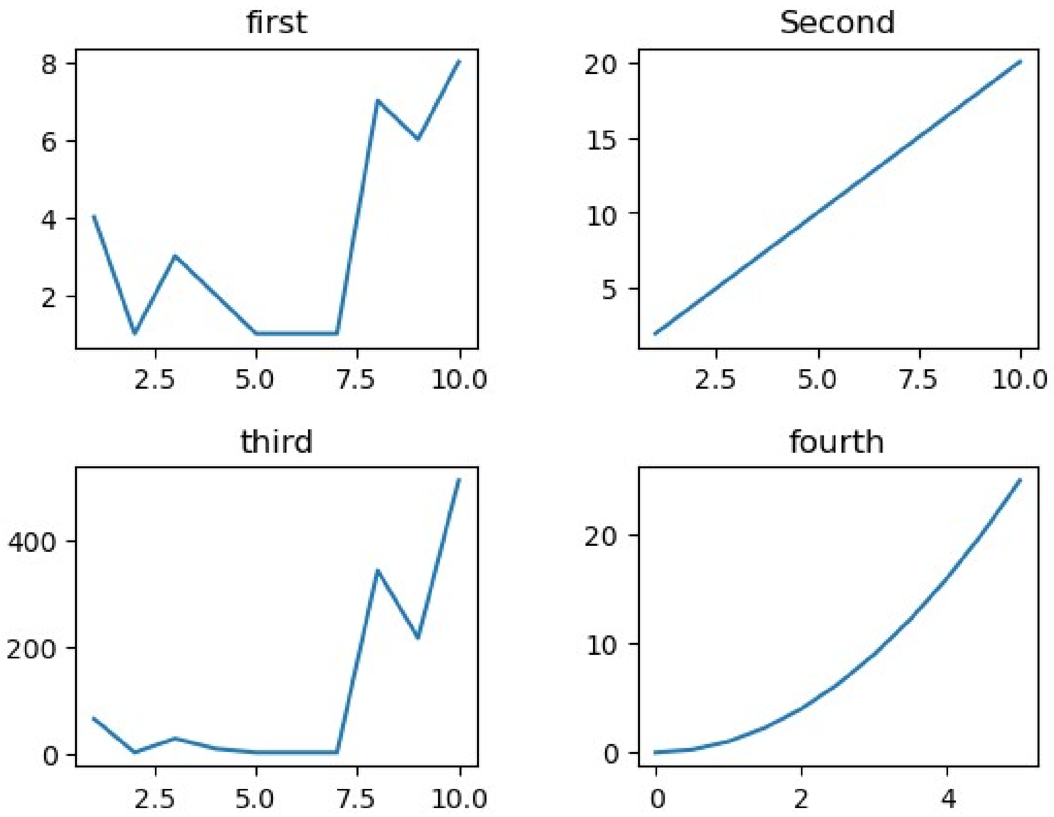


SUBPLOTS

you can create 4 plots together like this ,keeping each plot uder subplot to visulize that plot in that sub plot

```
plt.subplots_adjust(hspace=0.4, wspace=0.4) #used for spacing h for
vertical,w for horizontal
plt.subplot(2,2,1)
plt.plot(a,b)
plt.title("first")
plt.subplot(2,2,2)
plt.plot(a,c)
plt.title("Second")
plt.subplot(2,2,3)
plt.plot(a,d)
plt.title("third")
plt.subplot(2,2,4)
plt.plot(x,y)
plt.title("fourth")
```

```
Text(0.5, 1.0, 'fourth')
```



object oriented way

the white region is the canvas and whole stuff is image here

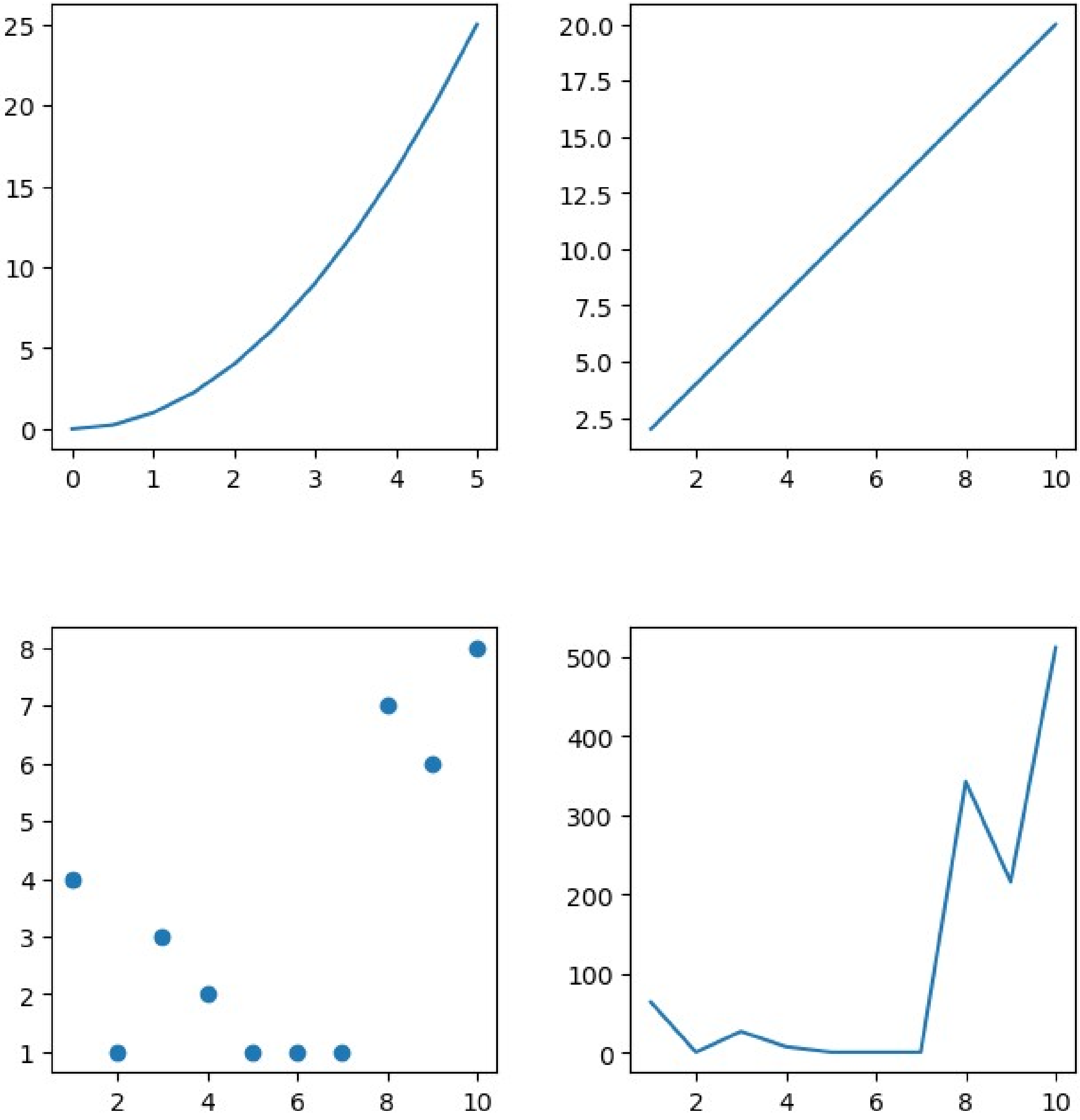
Matplotlib allows the aspect ratio ,DPI and figure size to be specified when the figure object is created.You can use the fig size and dpi keyword arguements

figsize is a tuple of the width and height of the figure in inches dpi is the dot-per-inch (pixel per inch)(more clear view more dpi)(increses load set to 100 mainly).

object oriented gives us power to access canvas and ,allow us to change

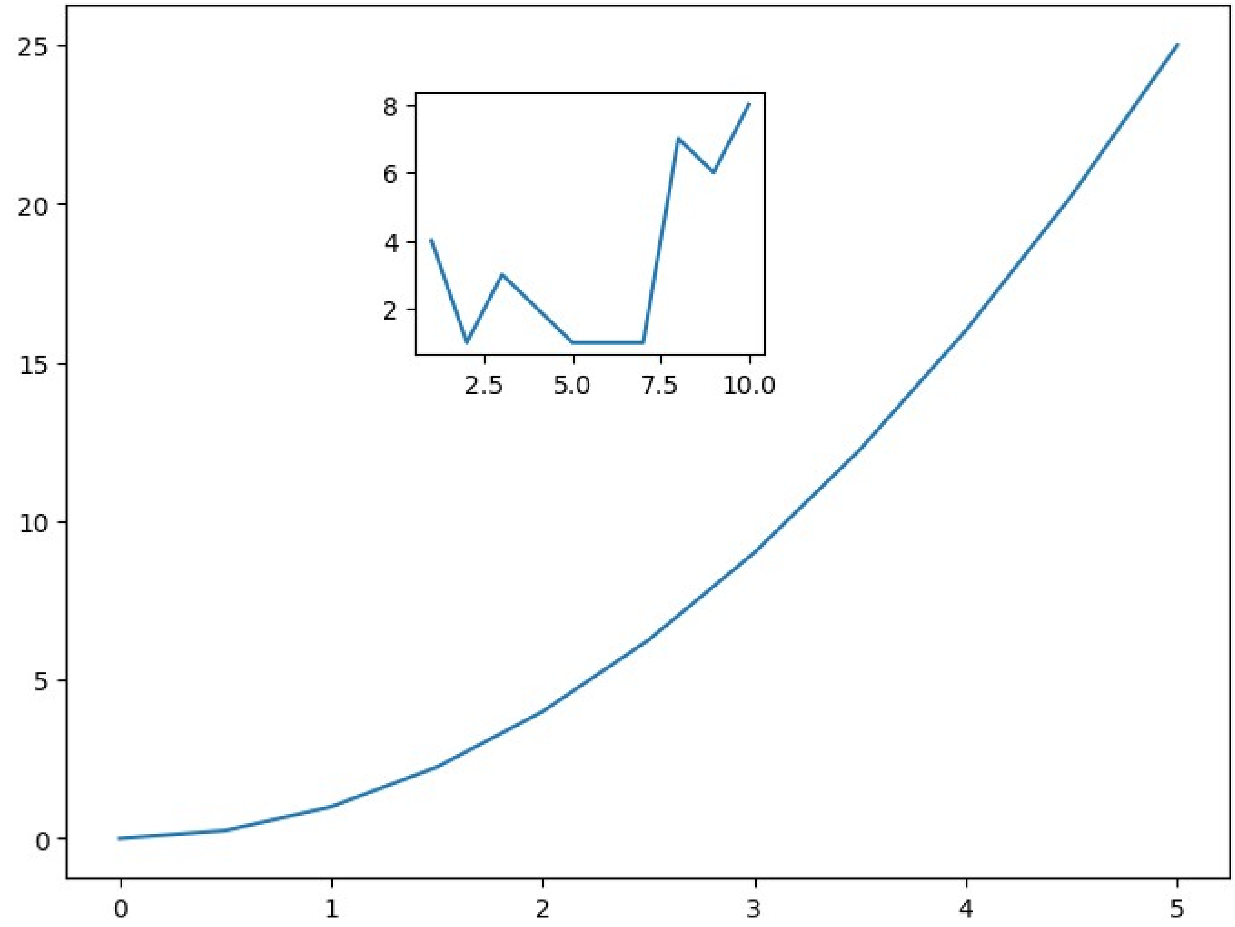
```
fig=plt.figure(figsize=(5,5),dpi=100)
axis1=fig.add_axes([0.05,0.7,0.5,0.5]) #[left margin,bottom
margin,size width,size height]
axis1.plot(x,y)
axis2=fig.add_axes([0.05,0,0.5,0.5])
axis2.scatter(a,b)
axis3=fig.add_axes([0.7,0.7,0.5,0.5])
axis3.plot(a,c)
axis4=fig.add_axes([0.7,0,0.5,0.5])
axis4.plot(a,d)
```

```
[<matplotlib.lines.Line2D at 0x1f4fd958cd0>]
```



we can create a figure plot like object with the help of plt.figure and create axes using fig.add_axes and to create multiple plots use in same fig cell and adjusting parameters

```
fig=plt.figure()
axis1=fig.add_axes([0,0,1,1]) #[left margin,bottom margin,size
width,size height]
axis1.plot(x,y)
axis2=fig.add_axes([0.3,0.6,0.3,0.3])
axis2.plot(a,b)

[<matplotlib.lines.Line2D at 0x1f4fef36210>]
```



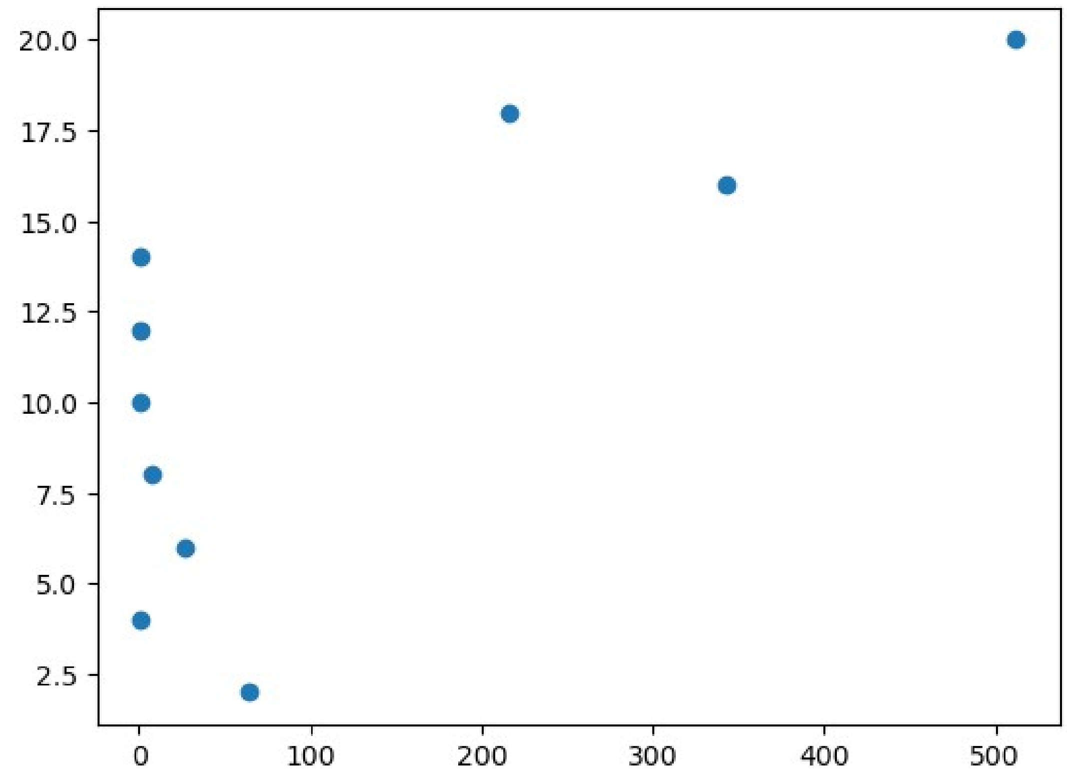
by adjusting left bottom height wisth you can manipulate each plot position and size

Types of Plot

Scatter Plot

```
plt.scatter(d,c)
```

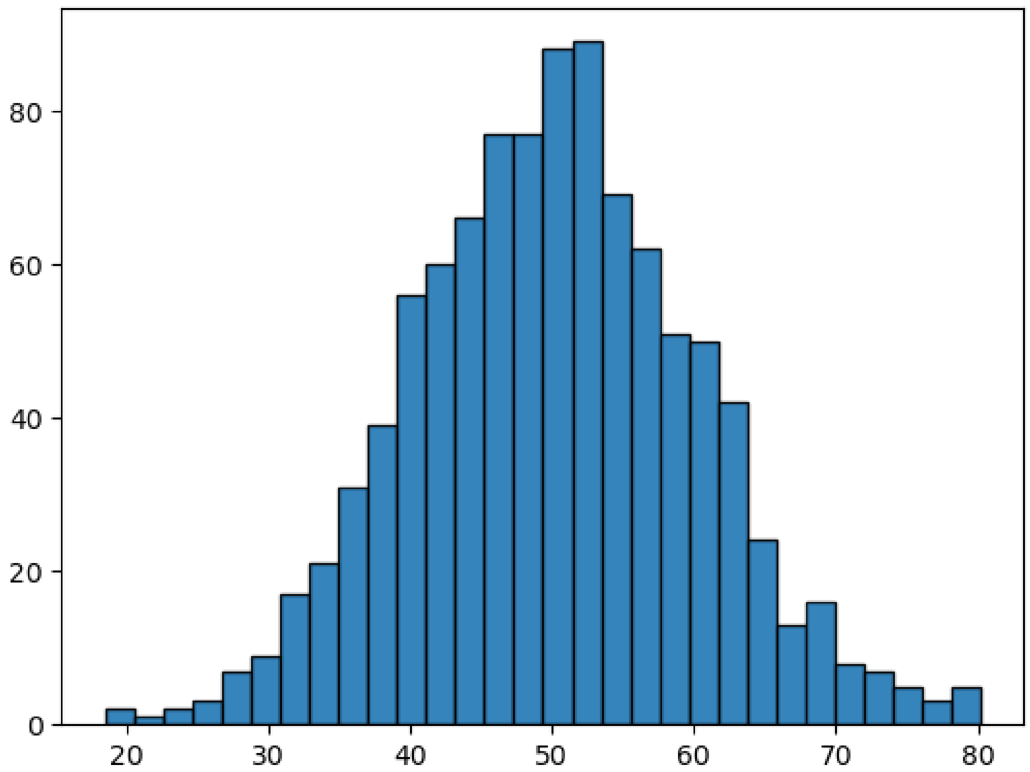
```
<matplotlib.collections.PathCollection at 0x1f4fefc8a50>
```



Histogram

```
data = np.random.normal(loc=50, scale=10, size=1000)
plt.hist(data, bins=30, edgecolor='black', alpha=0.9)#bin=x,alpha=edge
transparency
```

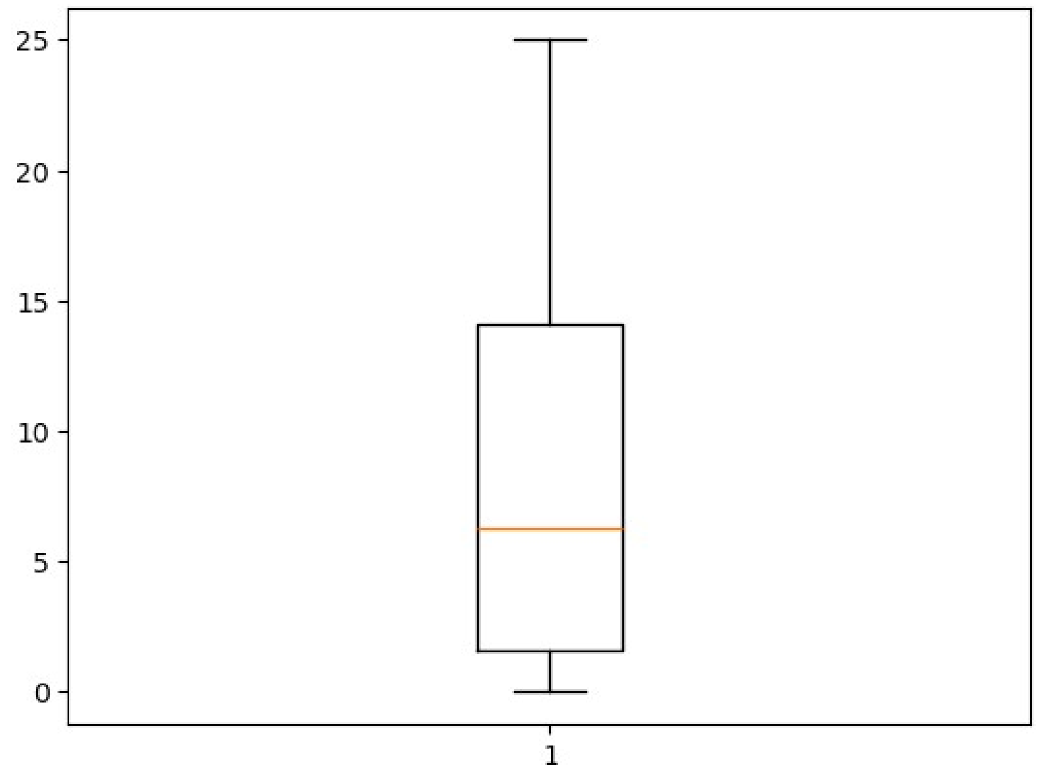
```
(array([ 2.,  1.,  2.,  3.,  7.,  9., 17., 21., 31., 39., 56., 60.,
66.,
        77., 77., 88., 89., 69., 62., 51., 50., 42., 24., 13., 16.,
8.,
         7.,  5.,  3.,  5.]),
 array([18.52889508, 20.58409537, 22.63929566, 24.69449596,
26.74969625,
        28.80489654, 30.86009684, 32.91529713, 34.97049743,
37.02569772,
        39.08089801, 41.13609831, 43.1912986 , 45.24649889,
47.30169919,
        49.35689948, 51.41209977, 53.46730007, 55.52250036,
57.57770065,
        59.63290095, 61.68810124, 63.74330153, 65.79850183,
67.85370212,
        69.90890241, 71.96410271, 74.019303  , 76.0745033 ,
78.12970359,
        80.18490388]),
 <BarContainer object of 30 artists>)
```



Box PLOT

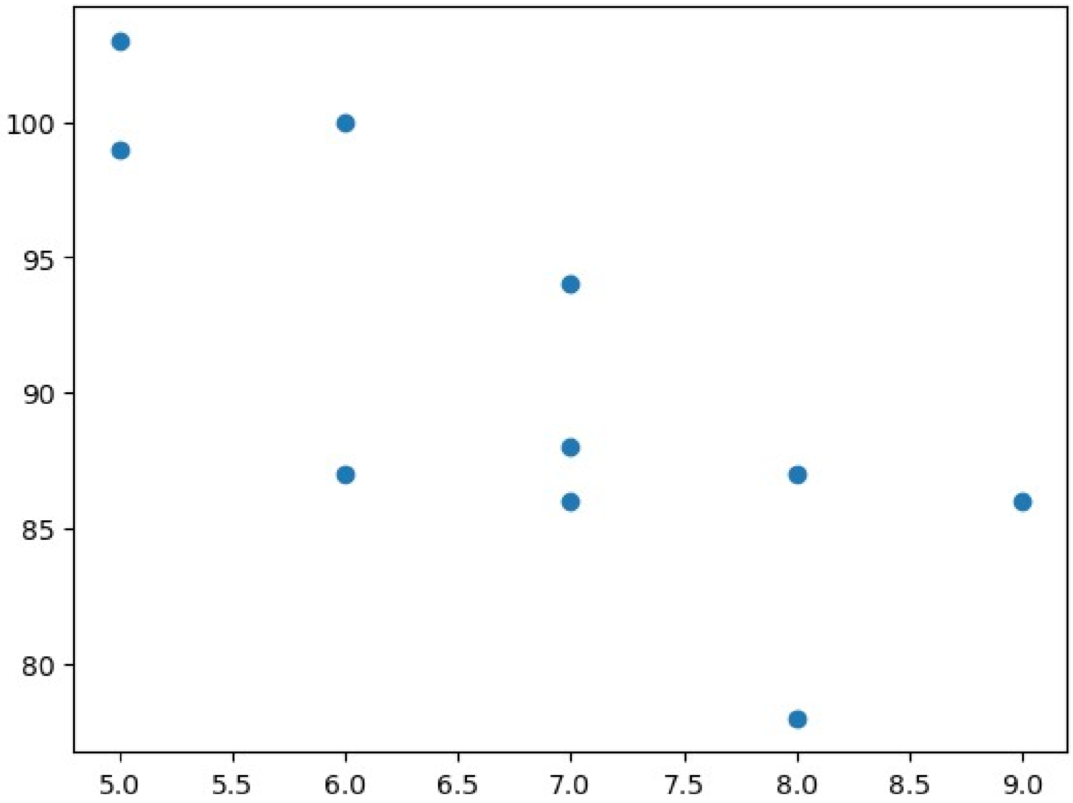
```
plt.boxplot(y)
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x1f48c883890>,
  <matplotlib.lines.Line2D at 0x1f48c8839d0>],
 'caps': [<matplotlib.lines.Line2D at 0x1f48c883b10>,
  <matplotlib.lines.Line2D at 0x1f48c883c50>],
 'boxes': [<matplotlib.lines.Line2D at 0x1f48c883750>],
 'medians': [<matplotlib.lines.Line2D at 0x1f48c883d90>],
 'fliers': [<matplotlib.lines.Line2D at 0x1f48c883ed0>],
 'means': []}
```



Saving PLOTS

```
x = np.array([5, 7, 8, 7, 6, 9, 5, 6, 7, 8])
y = np.array([99, 86, 87, 88, 100, 86, 103, 87, 94, 78])
fig=plt.figure()
scatter=fig.add_axes([0.1,0.1,0.8,0.8])
scatter.scatter(x, y)
plt.savefig("scatterplot.png")
```



working Images

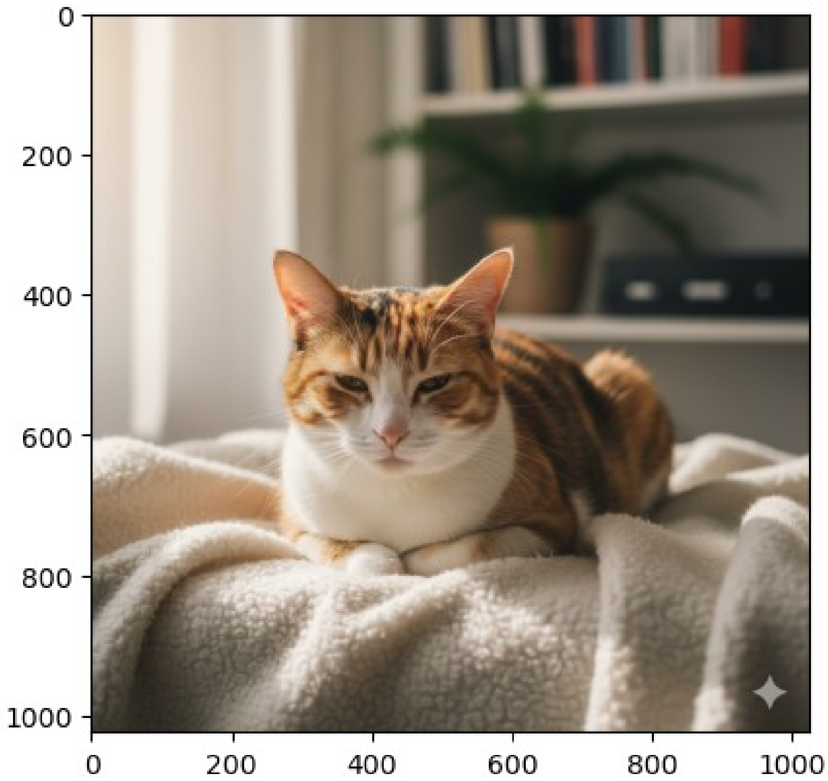
matplotlib helps in working with images which seaborn cannot

```
import matplotlib.image as mpimg

img=mpimg.imread('cat.png')
```

img

array of images basis on pixels

```
plt.imshow(img)

<matplotlib.image.AxesImage at 0x1f4905c0050>
```

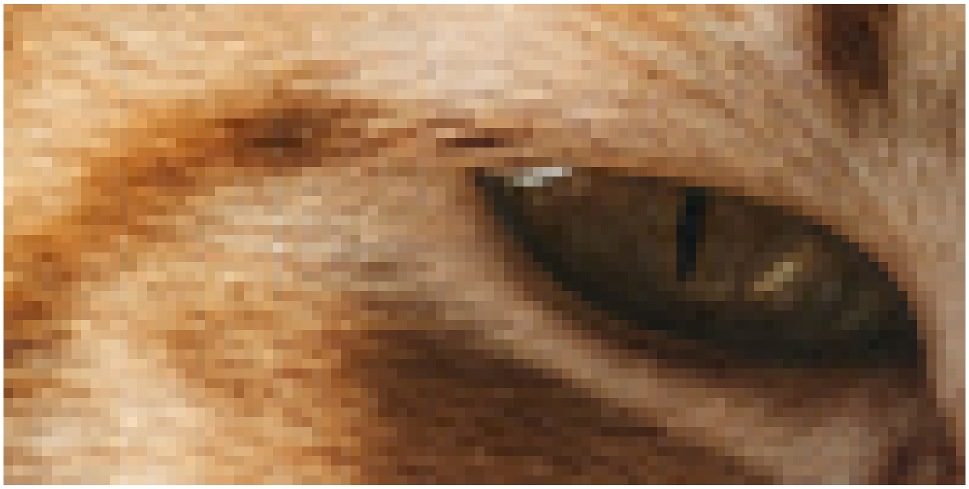


```
cropped_img=img[200:800,200:800] #crop: rows50-200,cols 100-300

plt.imshow(cropped_img)
plt.axis('off')
```

```
(np.float64(-0.5), np.float64(599.5), np.float64(599.5), np.float64(-
0.5))
```

```
cropped_img=img[500:550,300:400] #crop: rows50-200,cols 100-300
plt.imshow(cropped_img)#rows horizontal & columns verticals
plt.axis('off')
```

```
(np.float64(-0.5), np.float64(99.5), np.float64(49.5), np.float64(-
0.5))
```

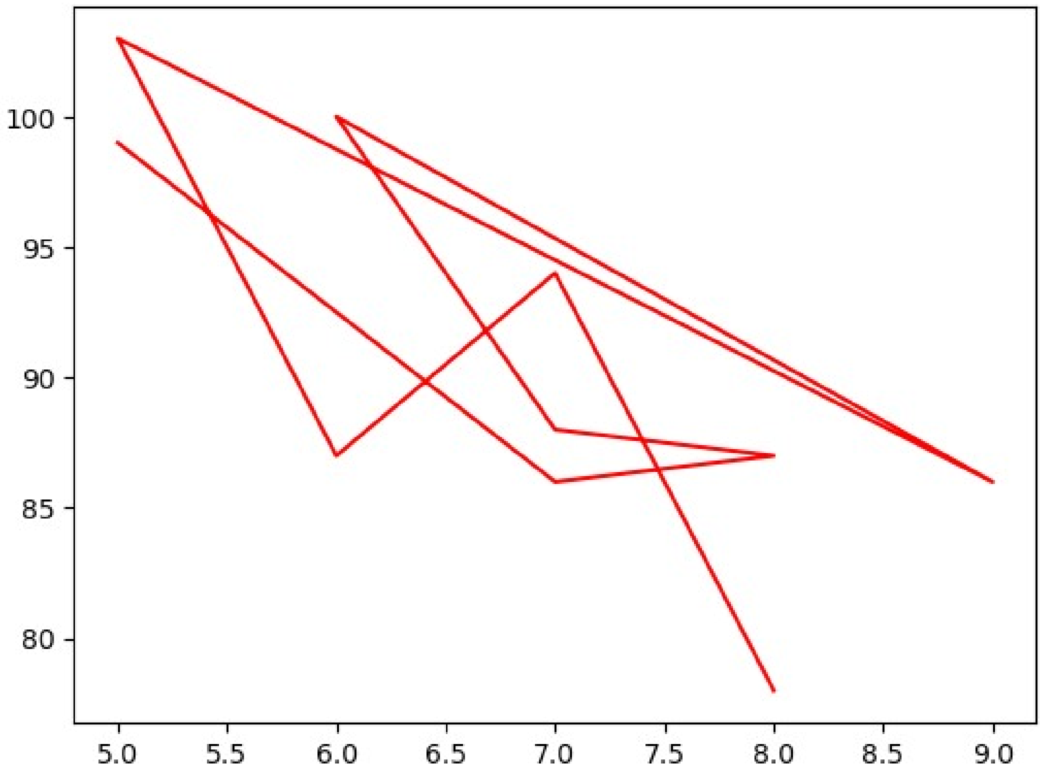


act as 2d array

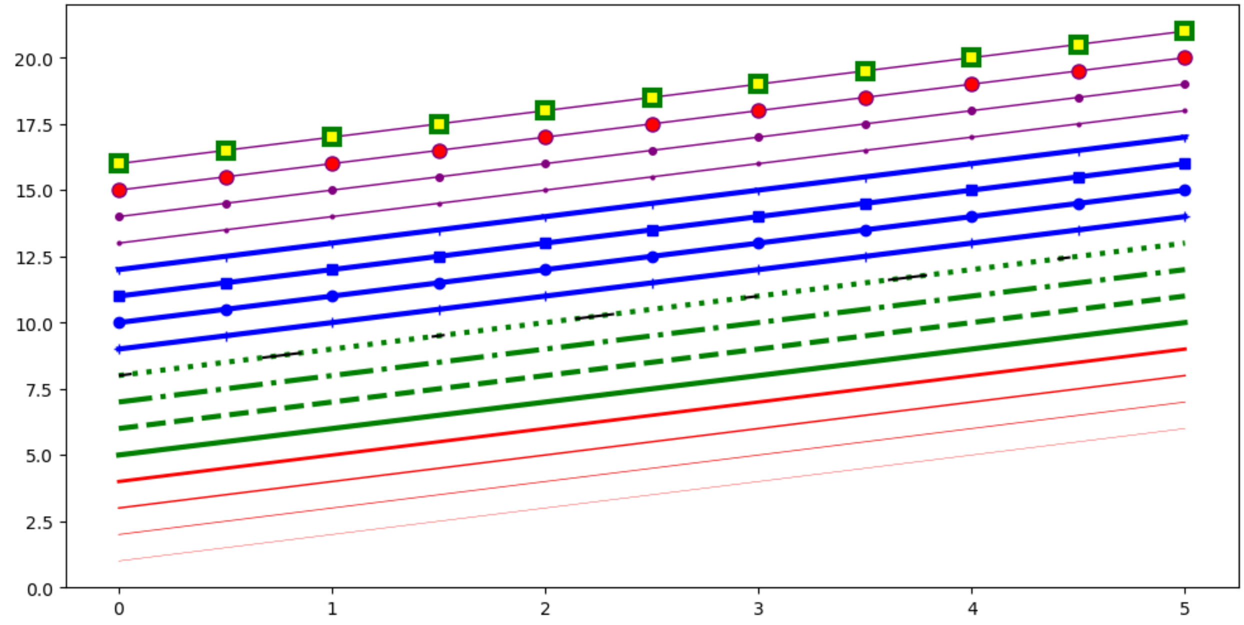
Optional designing

```
plt.plot(x,y,color="red")
```

```
[<matplotlib.lines.Line2D at 0x1f4ff62e710>]
```



```

# X-axis values
xaxis = np.linspace(0, 5, 11)

# Create figure and axis
fig, ax = plt.subplots(figsize=(12, 6))

# Line width examples
ax.plot(xaxis, xaxis+1, color="red", linewidth=0.25)
ax.plot(xaxis, xaxis+2, color="red", linewidth=0.50)
ax.plot(xaxis, xaxis+3, color="red", linewidth=1.00)
ax.plot(xaxis, xaxis+4, color="red", linewidth=2.00)

# Line style examples
ax.plot(xaxis, xaxis+5, color="green", lw=3, linestyle='-')
ax.plot(xaxis, xaxis+6, color="green", lw=3, ls='--')
ax.plot(xaxis, xaxis+7, color="green", lw=3, ls='-.')
ax.plot(xaxis, xaxis+8, color="green", lw=3, ls=':')

# Custom dash
line, = ax.plot(xaxis, xaxis+8, color="black", lw=1.50)
line.set_dashes([5, 50, 15, 50])  # line length, space length, ...

# Marker examples
ax.plot(xaxis, xaxis+9,  color="blue", lw=3, ls='-', marker='+')
ax.plot(xaxis, xaxis+10, color="blue", lw=3, ls='-', marker='o')
ax.plot(xaxis, xaxis+11, color="blue", lw=3, ls='-', marker='s')
ax.plot(xaxis, xaxis+12, color="blue", lw=3, ls='-', marker='1')

# Marker size and color examples
ax.plot(xaxis, xaxis+13, color="purple", lw=1, ls='-', marker='o',
markersize=2)
ax.plot(xaxis, xaxis+14, color="purple", lw=1, ls='-', marker='o',
markersize=4)
ax.plot(xaxis, xaxis+15, color="purple", lw=1, ls='-', marker='o',
markersize=8, markerfacecolor="red")
ax.plot(xaxis, xaxis+16, color="purple", lw=1, ls='-', marker='s',
markersize=8,
        markerfacecolor="yellow", markeredgewidth=3,
markeredgecolor="green")

plt.show()
```


20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
1
2
3
4
5