

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

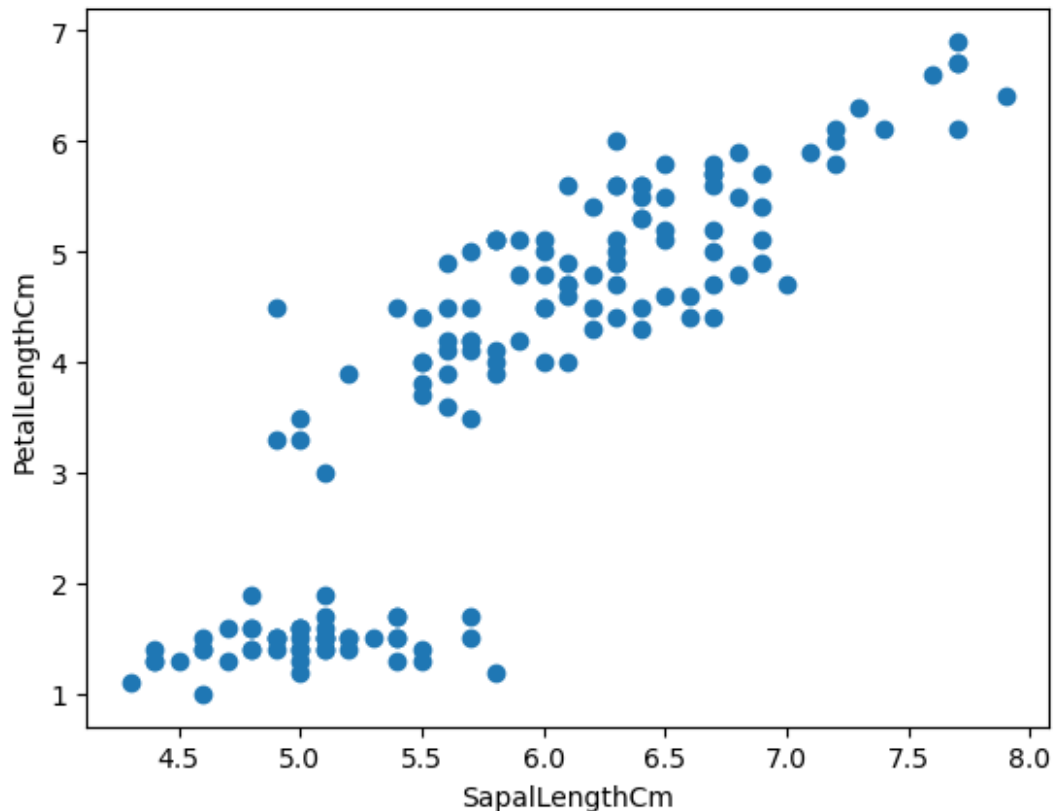
Colored Scatterplots

```
iris = pd.read_csv('iris.csv')
iris.sample(5)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	\
10	11	5.4	3.7	1.5	0.2	
149	150	5.9	3.0	5.1	1.8	
57	58	4.9	2.4	3.3	1.0	
61	62	5.9	3.0	4.2	1.5	
41	42	4.5	2.3	1.3	0.3	

	Species
10	Iris-setosa
149	Iris-virginica
57	Iris-versicolor
61	Iris-versicolor
41	Iris-setosa

```
plt.scatter(iris['SepalLengthCm'],iris['PetalLengthCm'])
plt.xlabel('SepalLengthCm')
plt.ylabel('PetalLengthCm')
Text(0, 0.5, 'PetalLengthCm')
```



```
# abhi sare tarah ke flower same color se dikh raha hai agar aap chaho
# to sare color ke lye alag alag color daal sakte ho abhi humare pass
# 3 types ke species hai i.e Iris-setosa, Iris-versicolor, Iris-
# virginica
```

```
### iss kaam ke lye hume apne species ko ek number assign karna hota
# hai suppose Iris-setosa =0 , Iris-versicolor =1 and Iris-virginica =2
```

```
iris['Species']=iris['Species'].replace({'Iris-setosa':0 , 'Iris-
versicolor' : 1 , 'Iris-virginica' : 2 })
iris.sample(5)
```

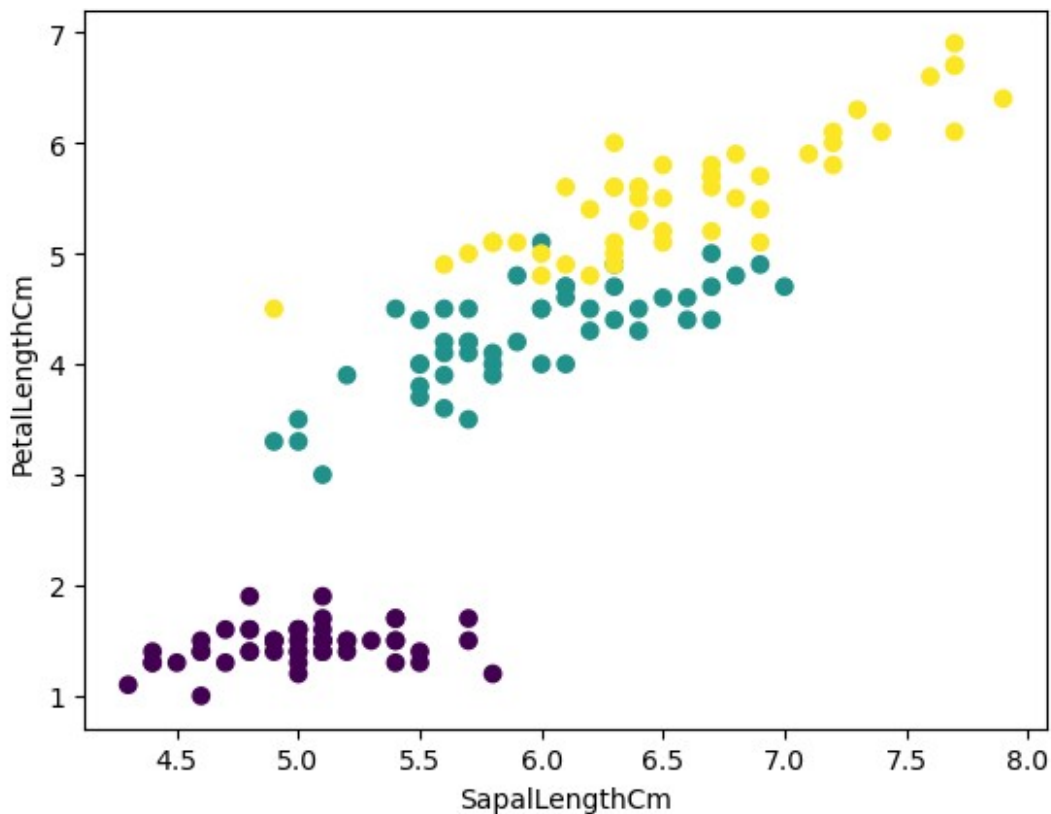
```
C:\Users\jayra\AppData\Local\Temp\ipykernel_20392\4177835182.py:1:
FutureWarning: Downcasting behavior in `replace` is deprecated and
will be removed in a future version. To retain the old behavior,
explicitly call `result.infer_objects(copy=False)`. To opt-in to the
future behavior, set `pd.set_option('future.no_silent_downcasting',
True)`
```

```
iris['Species']=iris['Species'].replace({'Iris-setosa':0 , 'Iris-
versicolor' : 1 , 'Iris-virginica' : 2 })
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
Species					
0	1	5.1	3.5	1.4	0.2
0					

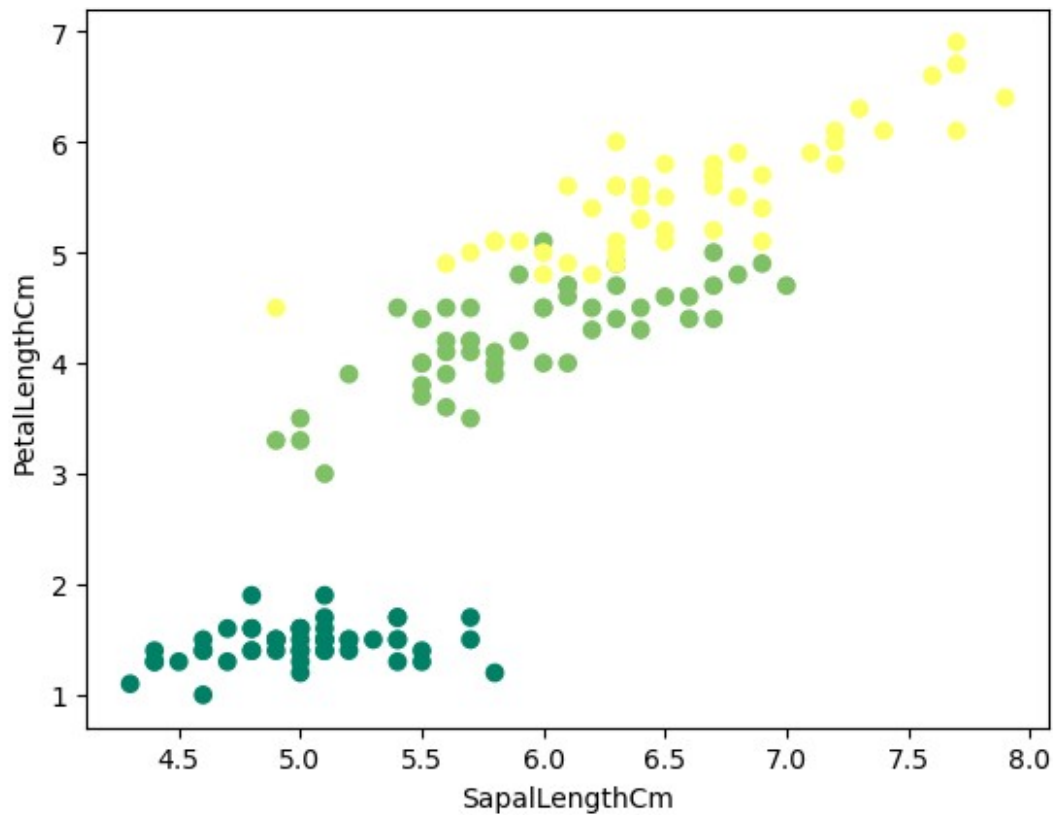
142	143	5.8	2.7	5.1	1.9
2					
108	109	6.7	2.5	5.8	1.8
2					
89	90	5.5	2.5	4.0	1.3
1					
68	69	6.2	2.2	4.5	1.5
1					

```
plt.scatter(iris['SepalLengthCm'],iris['PetalLengthCm'],c=iris['Species'])
plt.xlabel('SepalLengthCm')
plt.ylabel('PetalLengthCm')
Text(0, 0.5, 'PetalLengthCm')
```



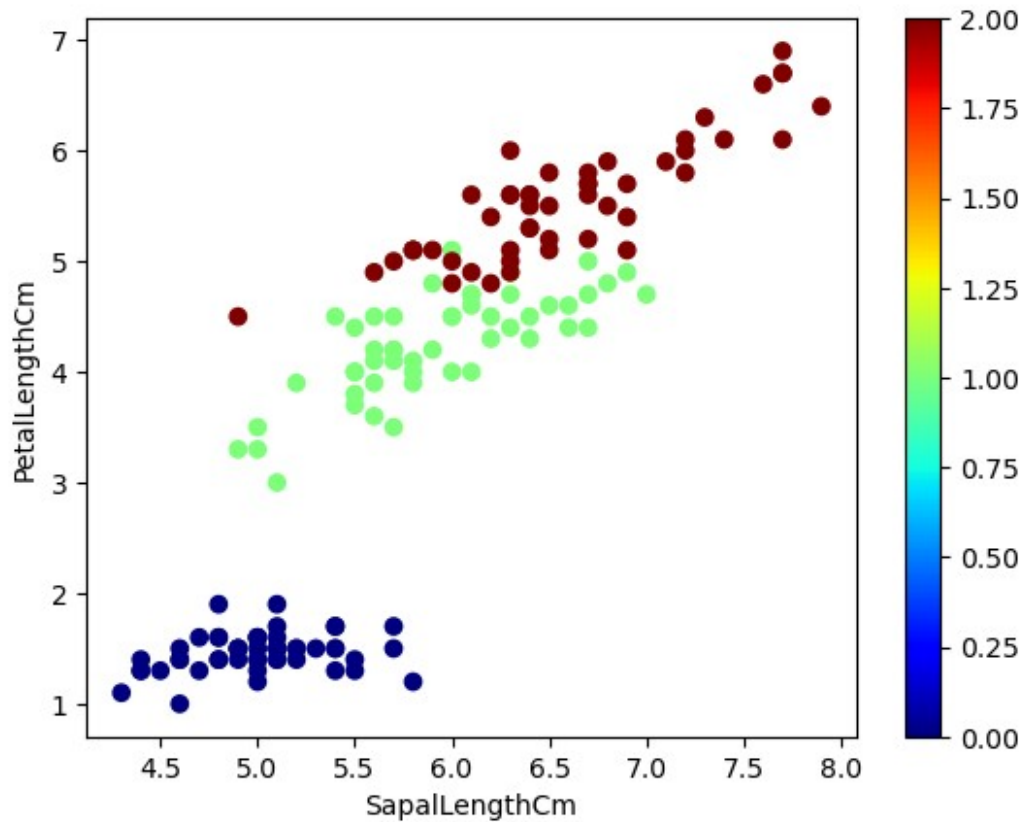
```
# aap ek alag parameter ke sath color bhi change kar sakte ho
plt.scatter(iris['SepalLengthCm'],iris['PetalLengthCm'],c=iris['Species'],
            cmap='summer') # viridis - default , jet, you can search ->
                             matplotlib cmap
plt.xlabel('SepalLengthCm')
plt.ylabel('PetalLengthCm')
```

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



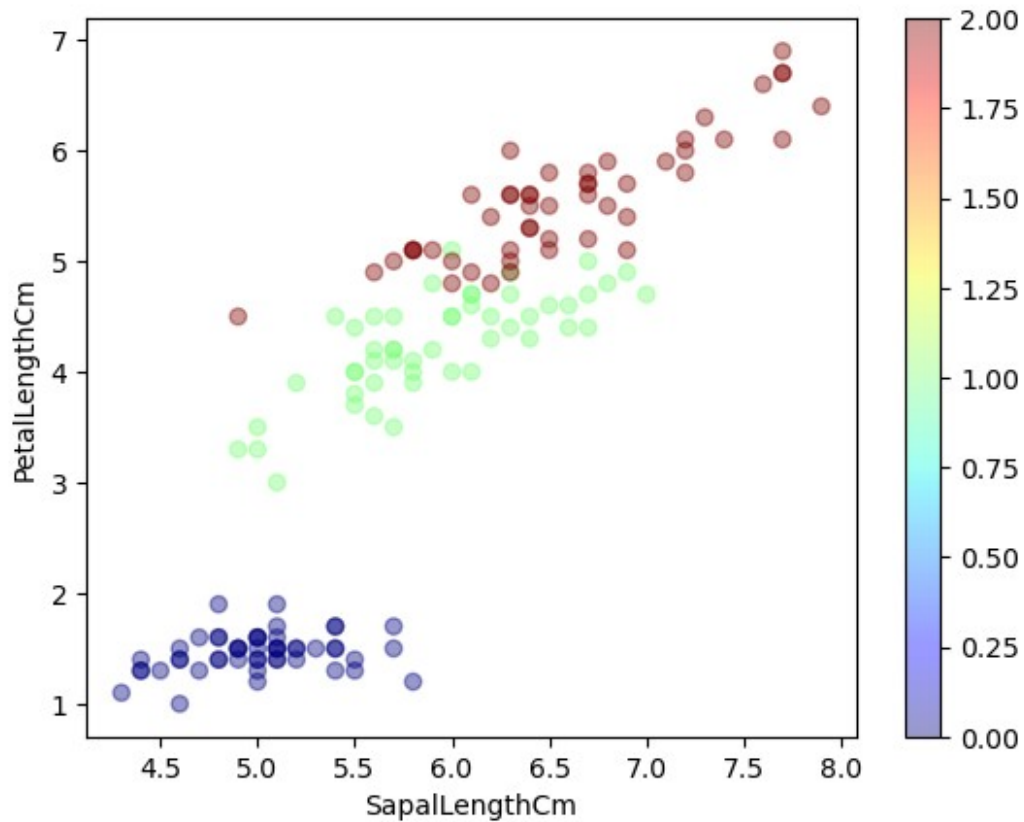
```
# you can add colorbar jisse ki pata chal jayega kon sa color kon sse  
# number ke species ko represent kar raha hai  
plt.scatter(iris['SepalLengthCm'],iris['PetalLengthCm'],c=iris['Species'],  
            cmap='jet')  
plt.xlabel('SepalLengthCm')  
plt.ylabel('PetalLengthCm')  
plt.colorbar()
```

```
<matplotlib.colorbar.Colorbar at 0x15c98eeb3b0>
```



```
# you can manage the opacity (using alpha)
plt.scatter(iris['SepalLengthCm'],iris['PetalLengthCm'],c=iris['Species'], cmap='jet',alpha=0.4)
plt.xlabel('SepalLengthCm')
plt.ylabel('PetalLengthCm')
plt.colorbar()

<matplotlib.colorbar.Colorbar at 0x15c99073ce0>
```

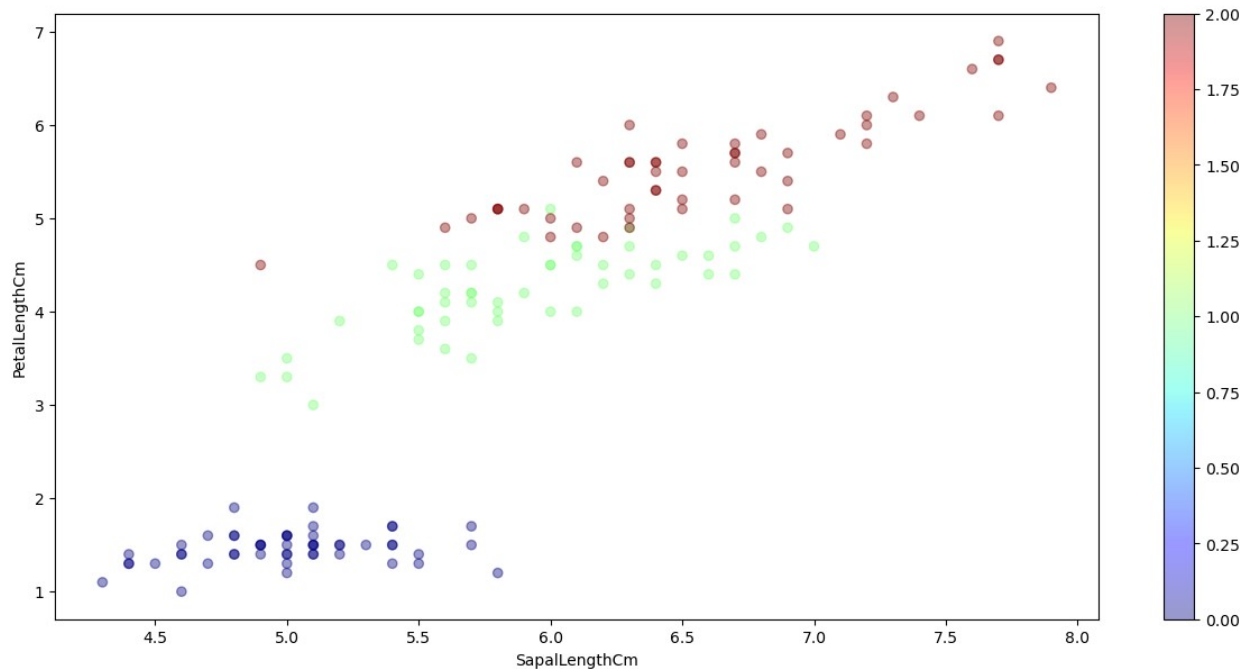


Plot Size

```
# what if graph ka size increase and decrease karna ho to
plt.figure(figsize=(15,7)) # isse humesa top pe hi likhna hota hai

plt.scatter(iris['SepalLengthCm'],iris['PetalLengthCm'],c=iris['Species'], cmap='jet',alpha=0.4)
plt.xlabel('SepalLengthCm')
plt.ylabel('PetalLengthCm')
plt.colorbar()

<matplotlib.colorbar.Colorbar at 0x15c9905bef0>
```



Annotations

```
batters = pd.read_csv("batter.csv")
```

```
batters.head()
```

	batter	runs	avg	strike_rate
0	V Kohli	6634	36.251366	125.977972
1	S Dhawan	6244	34.882682	122.840842
2	DA Warner	5883	41.429577	136.401577
3	RG Sharma	5881	30.314433	126.964594
4	SK Raina	5536	32.374269	132.535312

```
sample_df=batters.head(100).sample(25,random_state=5)
```

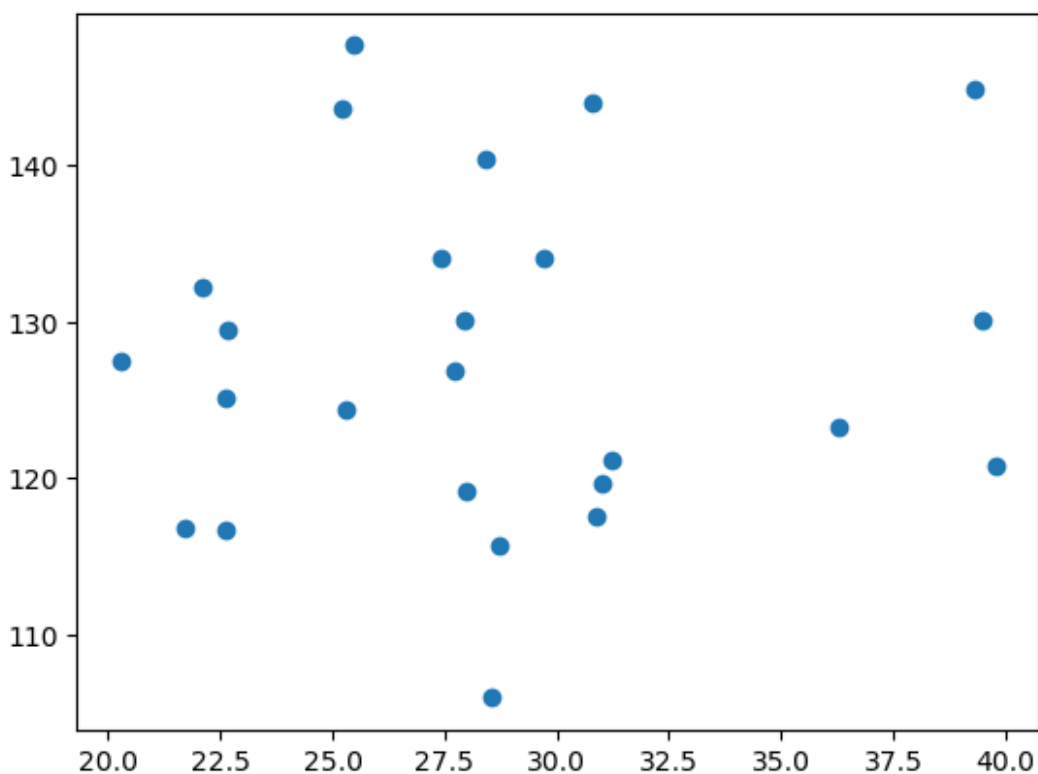
```
sample_df
```

	batter	runs	avg	strike_rate
66	KH Pandya	1326	22.100000	132.203390
32	SE Marsh	2489	39.507937	130.109775
46	JP Duminy	2029	39.784314	120.773810
28	SA Yadav	2644	29.707865	134.009123
74	IK Pathan	1150	21.698113	116.751269
23	JC Buttler	2832	39.333333	144.859335
10	G Gambhir	4217	31.007353	119.665153
20	BB McCullum	2882	27.711538	126.848592
17	KA Pollard	3437	28.404959	140.457703
35	WP Saha	2427	25.281250	124.397745

97	ST Jayasuriya	768	27.428571	134.031414
37	MA Agarwal	2335	22.669903	129.506378
70	DJ Hooda	1237	20.278689	127.525773
40	N Rana	2181	27.961538	130.053667
60	SS Tiwary	1494	28.730769	115.724245
34	JH Kallis	2427	28.552941	105.936272
42	KS Williamson	2105	36.293103	123.315759
57	DJ Bravo	1560	22.608696	125.100241
12	AM Rahane	4074	30.863636	117.575758
69	D Padikkal	1260	28.000000	119.205298
94	SO Hetmyer	831	30.777778	144.020797
56	PP Shaw	1588	25.206349	143.580470
22	PA Patel	2848	22.603175	116.625717
39	GJ Maxwell	2320	25.494505	147.676639
24	SS Iyer	2780	31.235955	121.132898

```
# sare scatter point ko ek name dena
plt.scatter(sample_df['avg'],sample_df['strike_rate'])

<matplotlib.collections.PathCollection at 0x15c9a0bdac0>
```

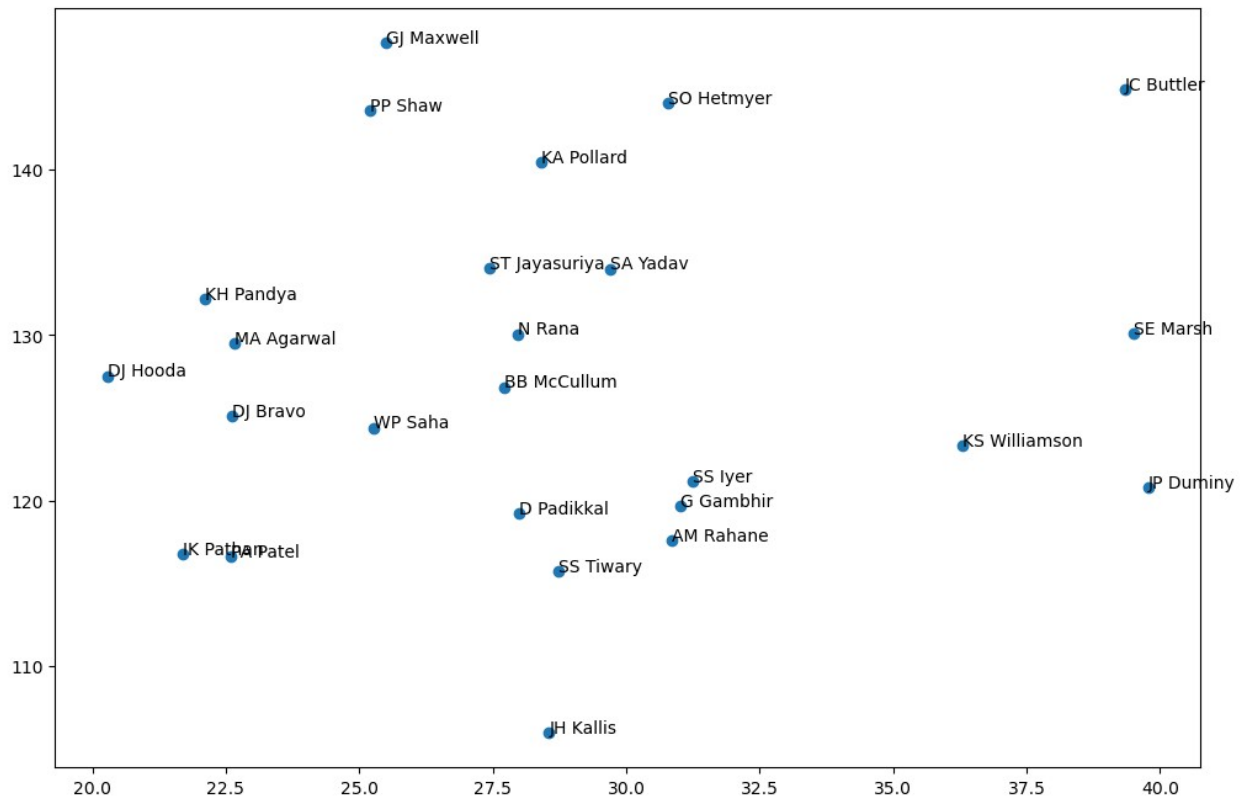


```
plt.figure(figsize=(12,8))

plt.scatter(sample_df['avg'],sample_df['strike_rate'])
for i in range(sample_df.shape[0]):
```



```
plt.text(sample_df['avg'].values[i],sample_df['strike_rate'].values[i],
sample_df['batter'].values[i])
```

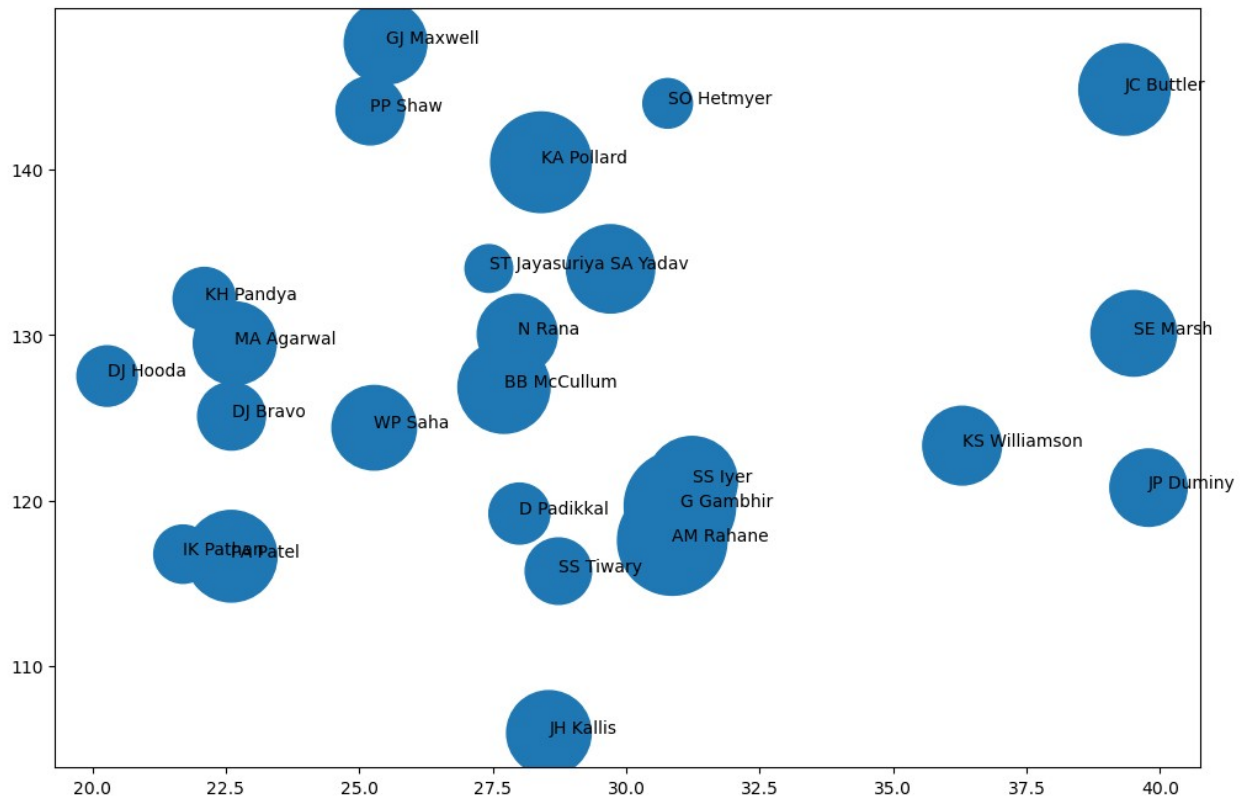


we want to customize the size of dot according to run jo jitna jyada run banaye hoga uska dot size utna bada

```
plt.figure(figsize=(12,8))

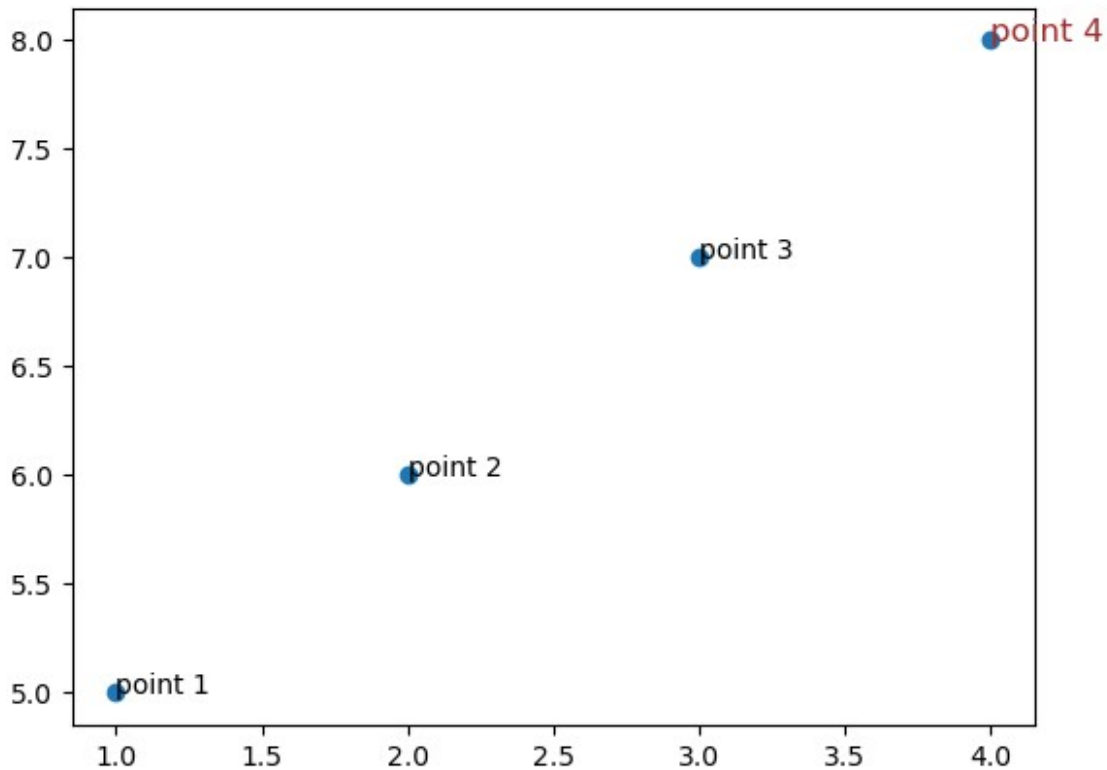
plt.scatter(sample_df['avg'],sample_df['strike_rate'],s=sample_df['run
s'])
for i in range(sample_df.shape[0]):

plt.text(sample_df['avg'].values[i],sample_df['strike_rate'].values[i],
sample_df['batter'].values[i])
```



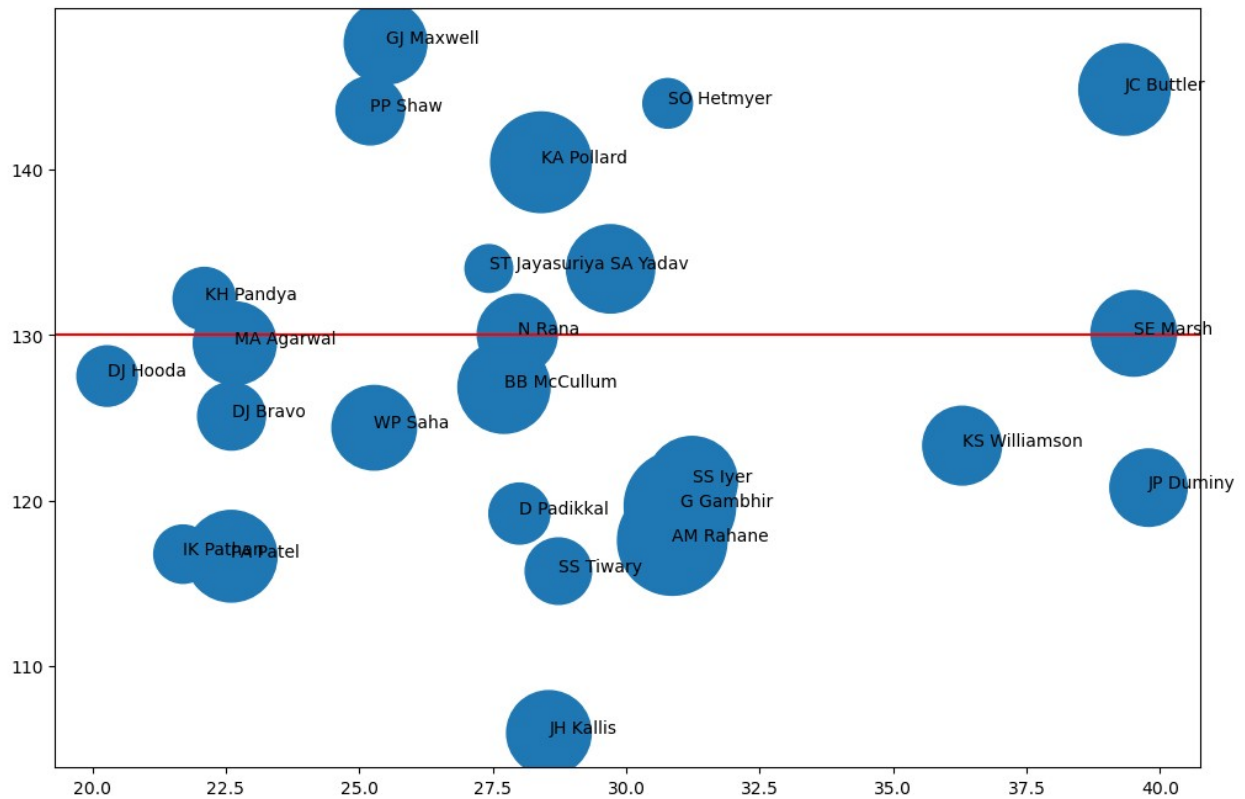
```
x = [1,2,3,4]
y = [5,6,7,8]

plt.scatter(x,y)
plt.text(1,5,'point 1')
plt.text(2,6,'point 2')
plt.text(3,7,'point 3')
plt.text(4,8,'point 4',fontdict={'size':12,'color':'brown'})
Text(4, 8, 'point 4')
```



horizontal and Vertical lines

```
# just imagine ki hum uss player ko salet karna chahate hain jiska  
strike rate grater than 130 hai  
plt.figure(figsize=(12,8))  
  
plt.scatter(sample_df['avg'],sample_df['strike_rate'],s=sample_df['runs'])  
  
plt.axhline(130,color='red')  
  
for i in range(sample_df.shape[0]):  
    plt.text(sample_df['avg'].values[i],sample_df['strike_rate'].values[i],  
            sample_df['batter'].values[i])
```



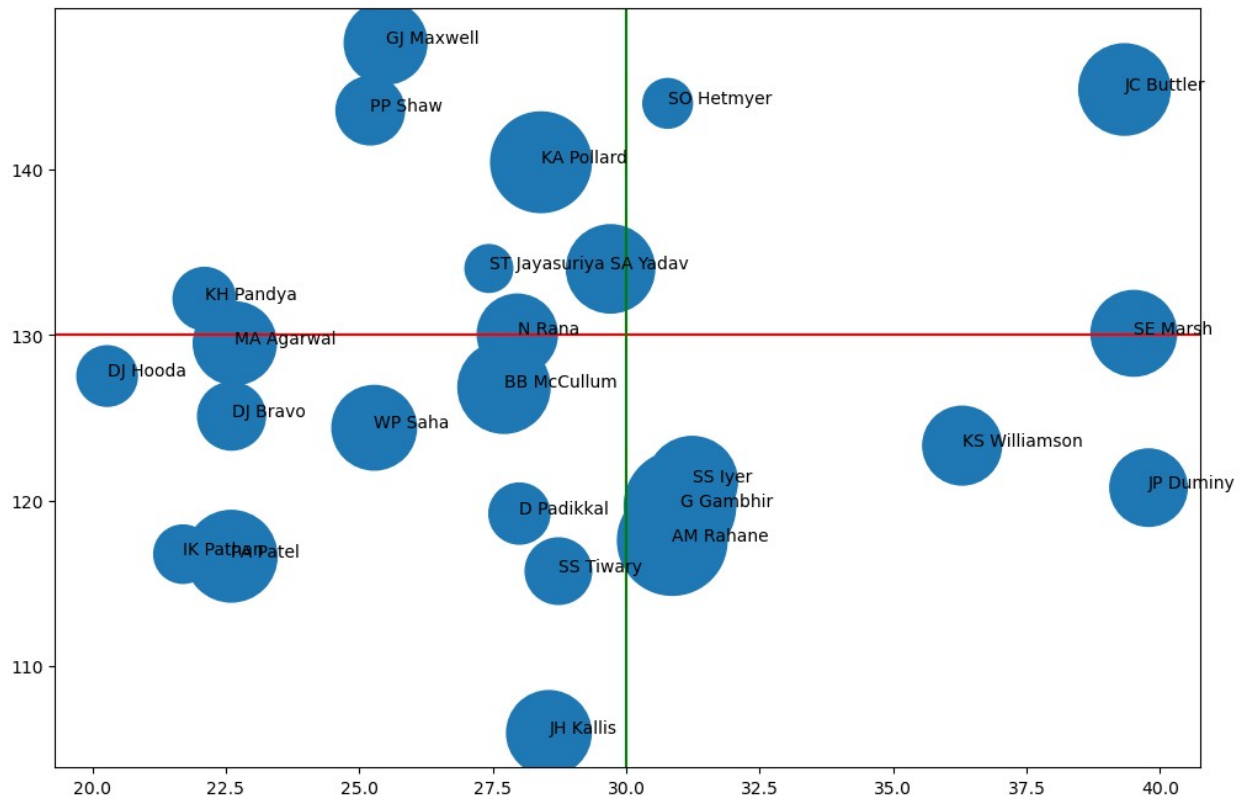
```
# not only we are focusing on strike rate lets focus on average
plt.figure(figsize=(12,8))

plt.scatter(sample_df['avg'],sample_df['strike_rate'],s=sample_df['runs'])

plt.axhline(130,color='red')
plt.axvline(30,color='green')

for i in range(sample_df.shape[0]):

plt.text(sample_df['avg'].values[i],sample_df['strike_rate'].values[i],sample_df['batter'].values[i])
```



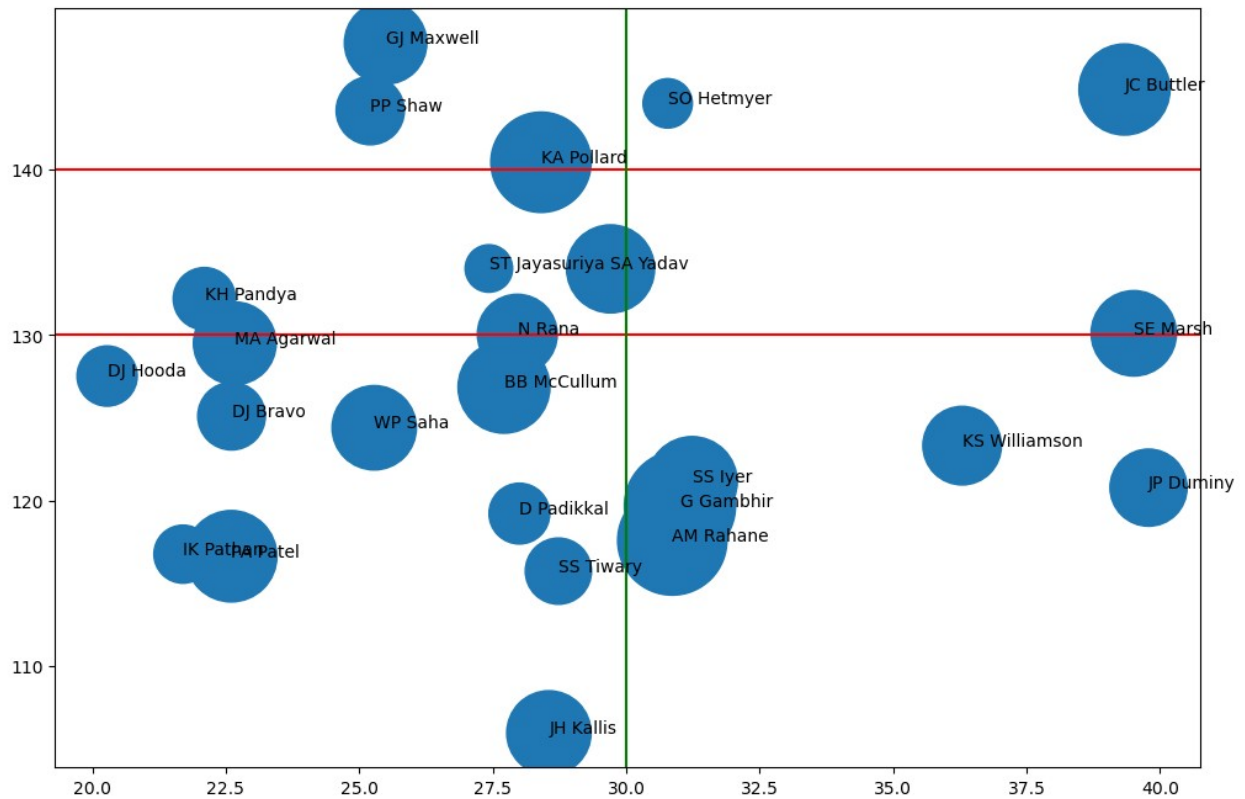
```
# multiple horizontal and vertical lines
plt.figure(figsize=(12,8))

plt.scatter(sample_df['avg'],sample_df['strike_rate'],s=sample_df['run
s'])

plt.axhline(130,color='red')
plt.axhline(140,color='red')
plt.axvline(30,color='green')

for i in range(sample_df.shape[0]):

plt.text(sample_df['avg'].values[i],sample_df['strike_rate'].values[i]
,sample_df['batter'].values[i])
```



Sub Plots

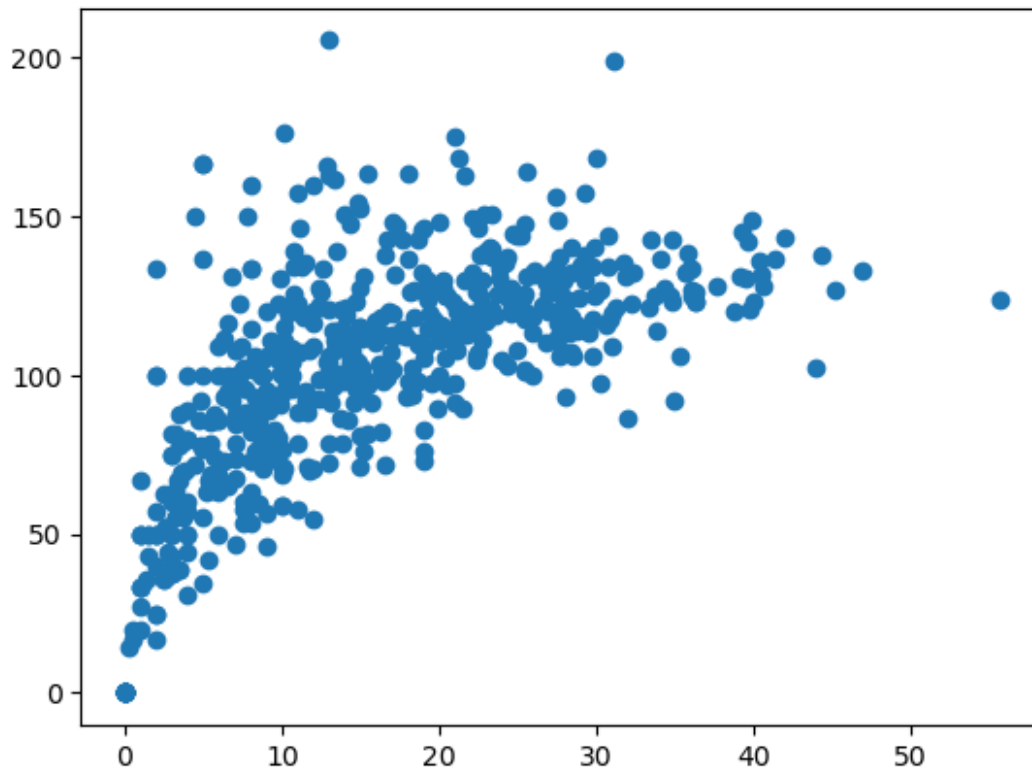
a diff way to plot graphs

```
batters.head()
```

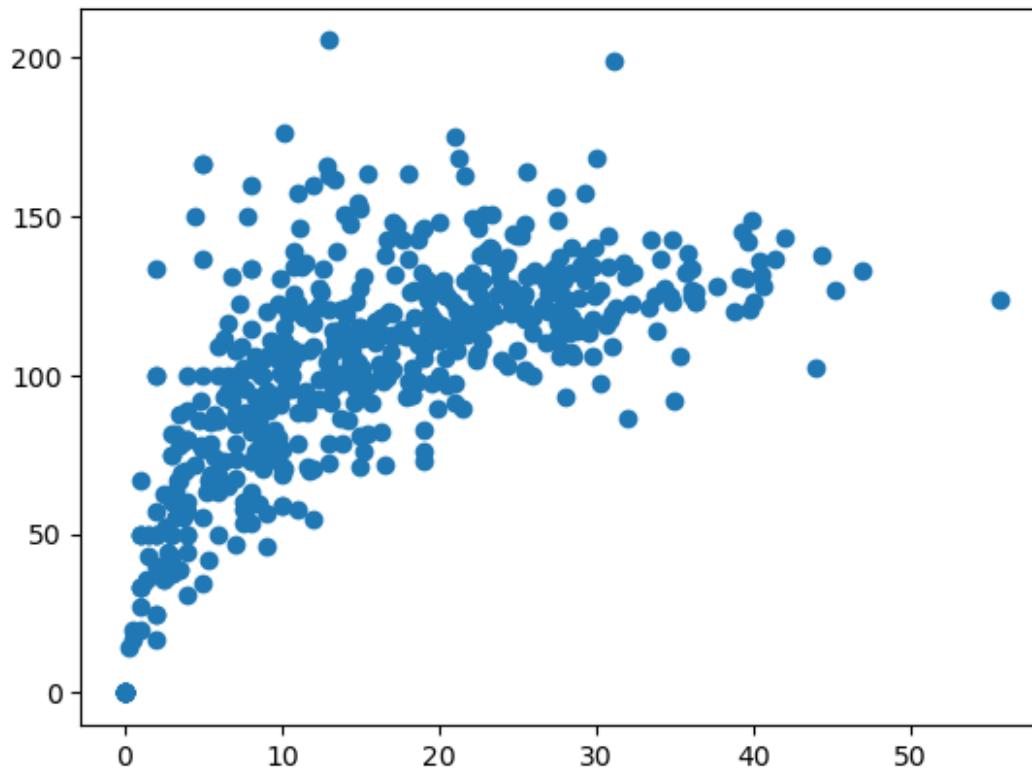
	batter	runs	avg	strike_rate
0	V Kohli	6634	36.251366	125.977972
1	S Dhawan	6244	34.882682	122.840842
2	DA Warner	5883	41.429577	136.401577
3	RG Sharma	5881	30.314433	126.964594
4	SK Raina	5536	32.374269	132.535312

```
plt.scatter(batters['avg'],batters['strike_rate'])
```

```
<matplotlib.collections.PathCollection at 0x15c9a24f3e0>
```



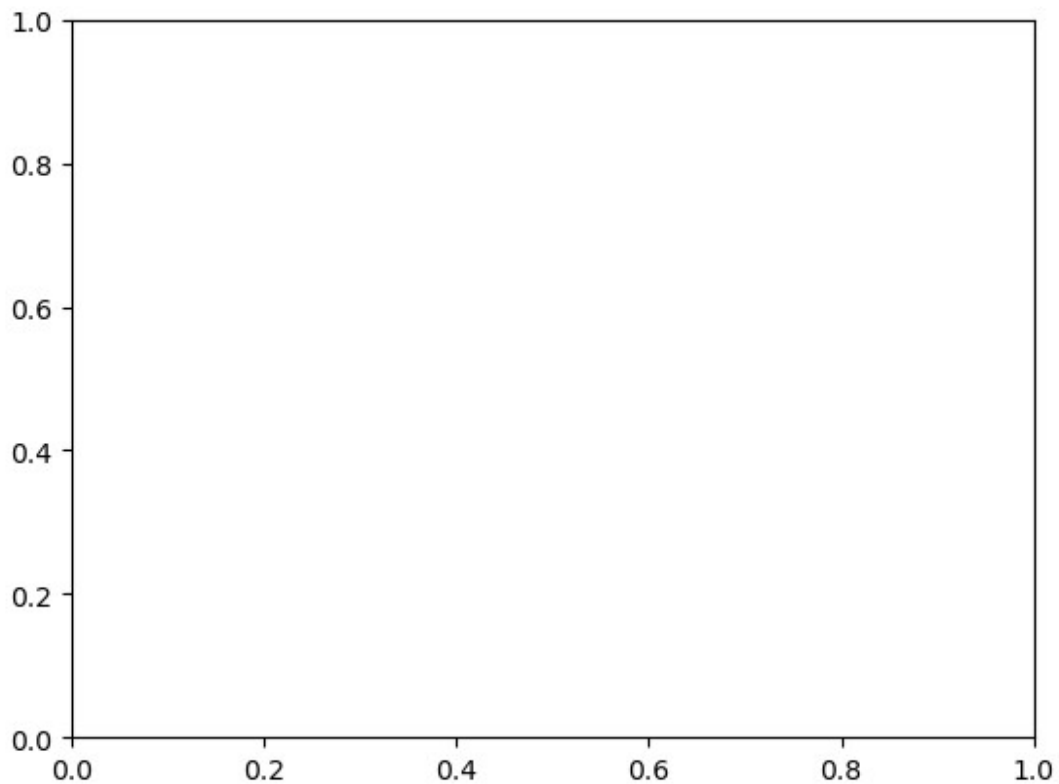
```
# diff way
fig, ax=plt.subplots()
ax.scatter(batters['avg'],batters['strike_rate'])
<matplotlib.collections.PathCollection at 0x15c9a3aea20>
```



```
# breakdown the above code  
plt.subplots()
```

```
# basically jab iss code ko run karte hain to hume do object milta hai  
one is figure object and another is axes object isse hi do  
# alag alag variable (fig and ax me store kye hain) abb iss axes ko  
pakad ke hum koi bhi graph plot kar sakte hain
```

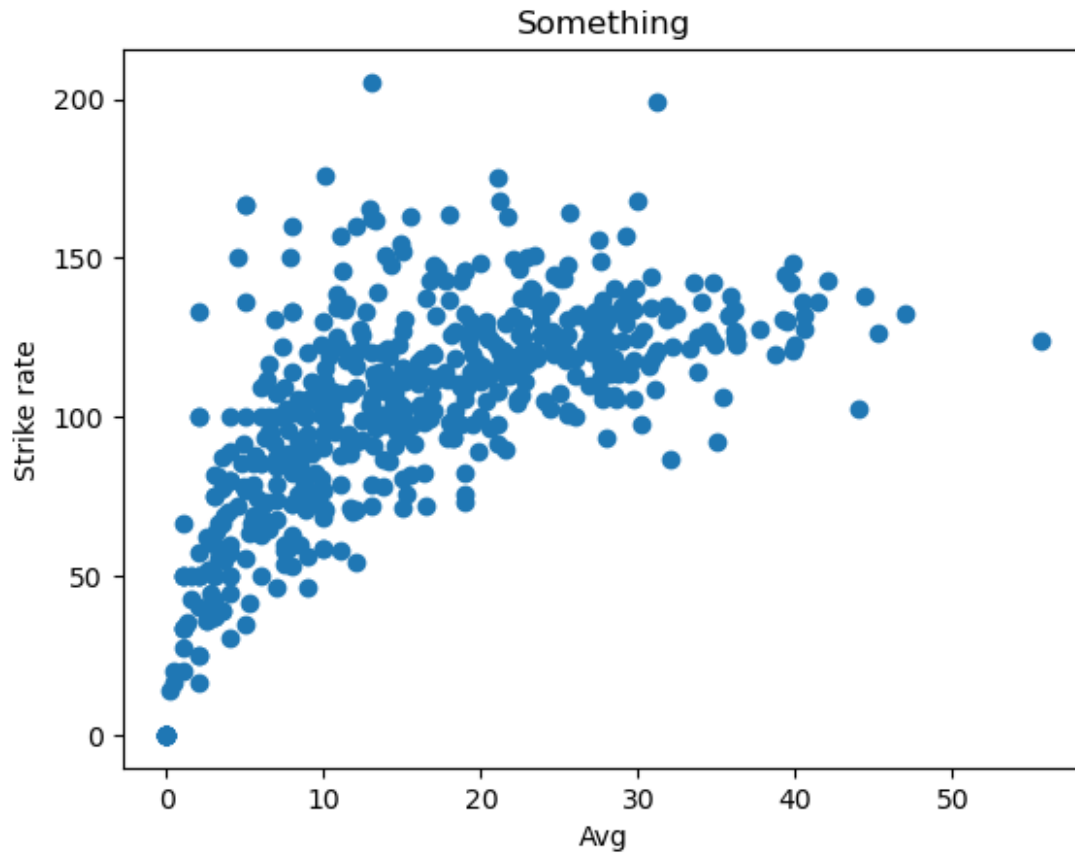
```
(<Figure size 640x480 with 1 Axes>, <Axes: >)
```

iss tarah se plot karne ke piche main reason hai ki aap ek sath me multiple graph plot kar sakte ho

how to set label and title in second method

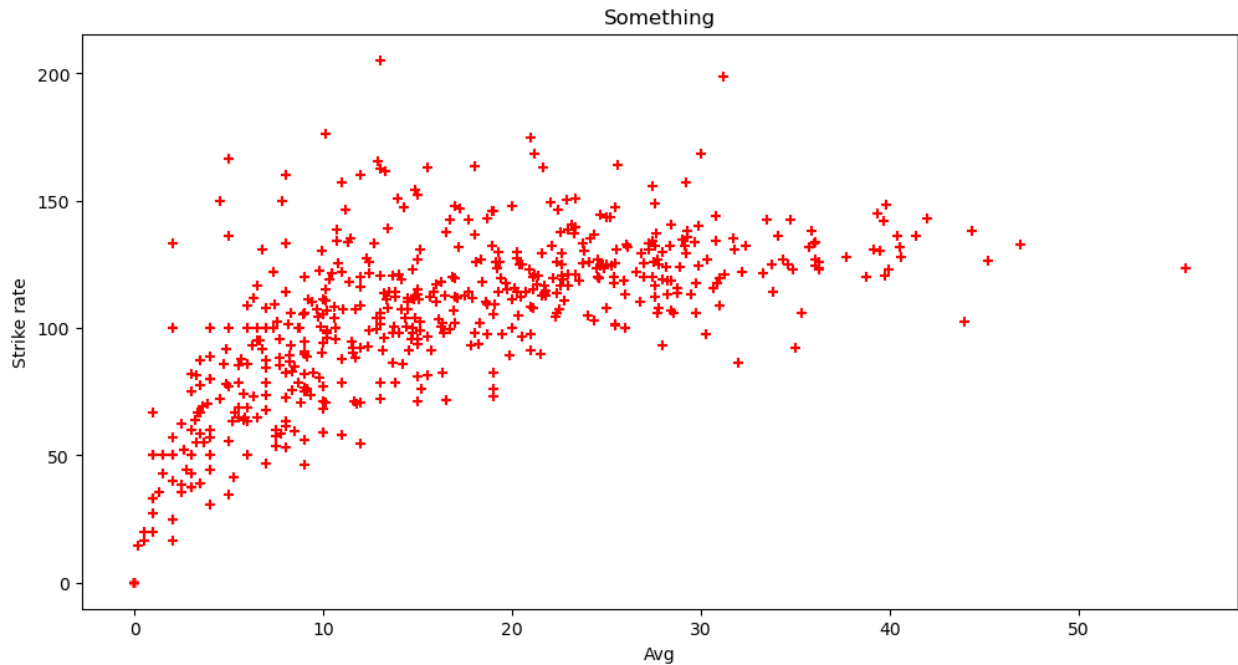
```
fig, ax=plt.subplots()
ax.scatter(batters['avg'],batters['strike_rate'])
ax.set_title('Something')
ax.set_xlabel('Avg')
ax.set_ylabel('Strike rate')
Text(0, 0.5, 'Strike rate')
```



```
# how to increase and decrease the graph size
fig, ax=plt.subplots(figsize=(12,6))

ax.scatter(batters['avg'],batters['strike_rate'],color='red',marker='+')
ax.set_title('Something')
ax.set_xlabel('Avg')
ax.set_ylabel('Strike rate')

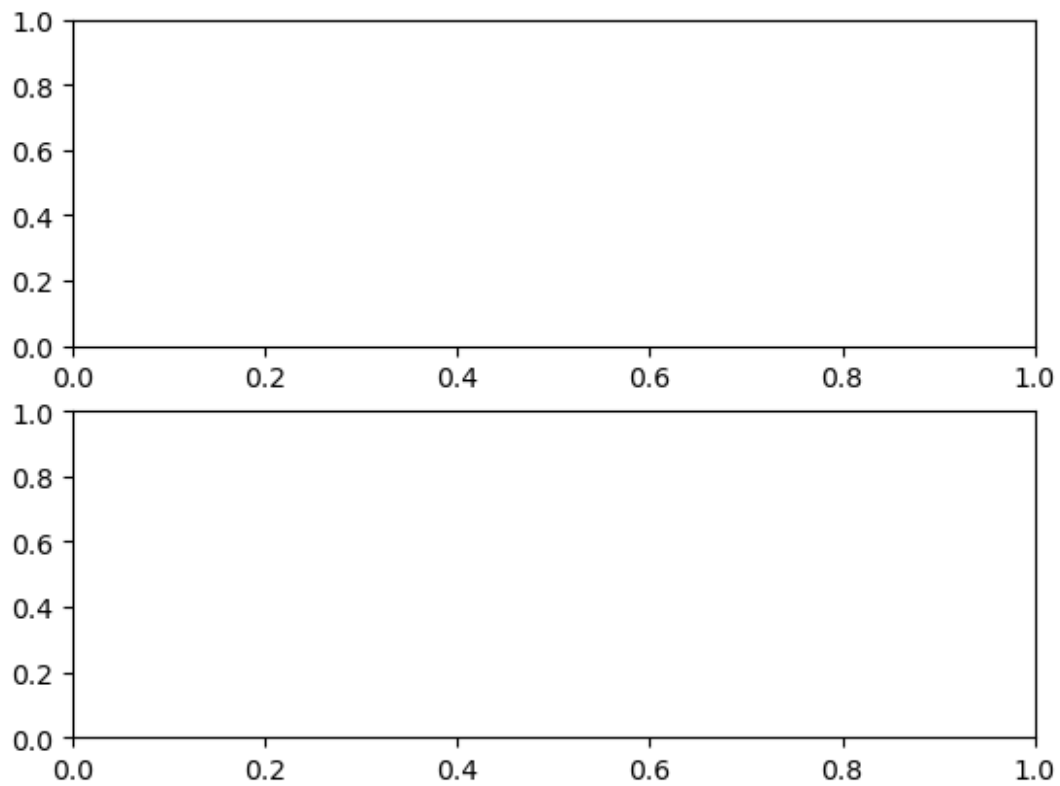
# plt.show() = fig.show()
Text(0, 0.5, 'Strike rate')
```



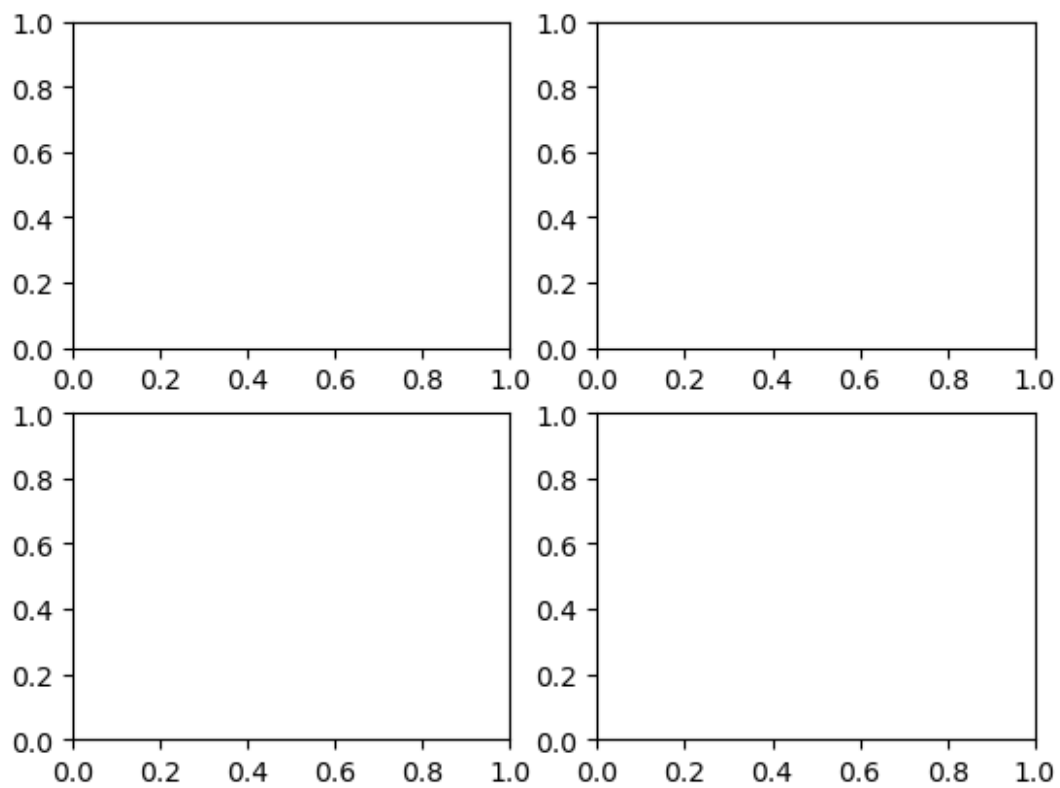
```
# plotting two graphs side by side one is for (avg vs strike_rete) and  
another is for (rus vs avg)
```

```
plt.subplots(nrows=2,ncols=1)
```

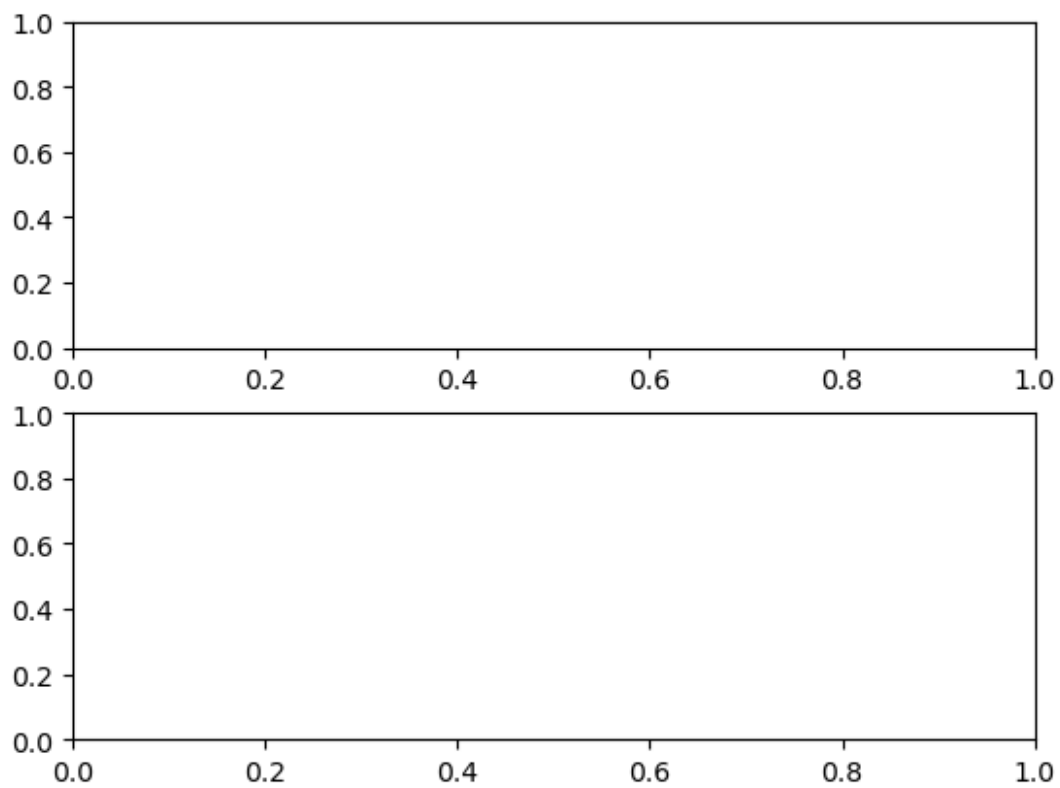
```
(<Figure size 640x480 with 2 Axes>, array([<Axes: >, <Axes: >],  
dtype=object))
```



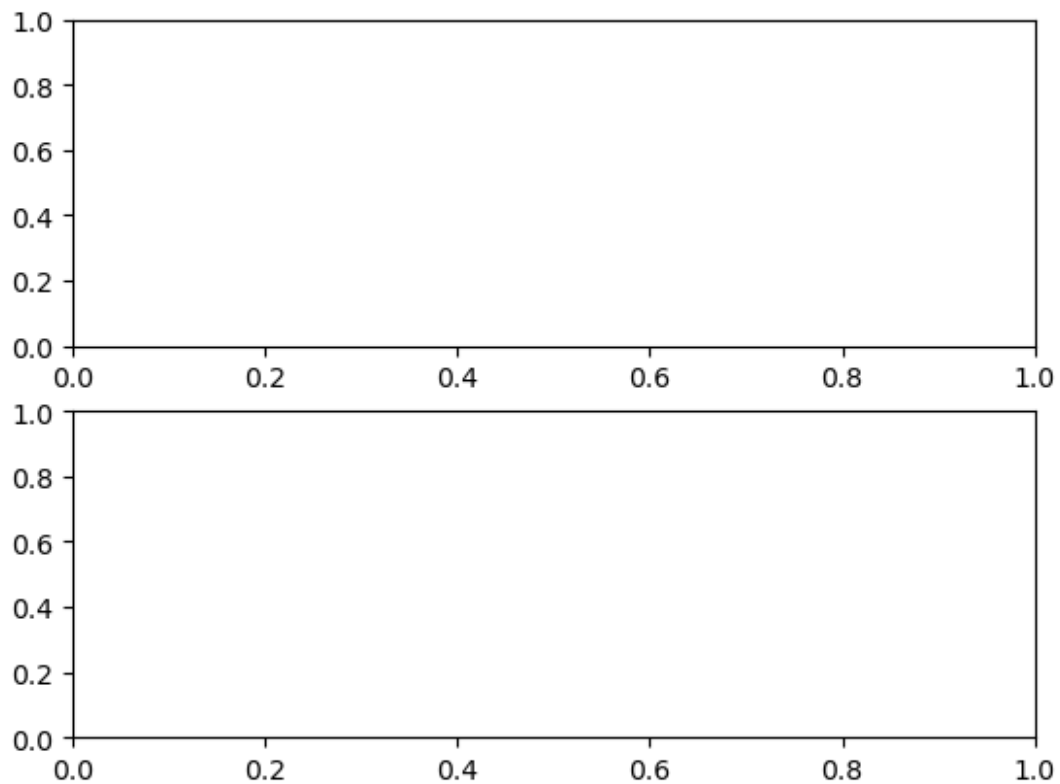
```
plt.subplots(nrows=2,ncols=2)    # total (nrows*ncols) graph bana sakte  
hain  
(<Figure size 640x480 with 4 Axes>,  
 array([[<Axes: >, <Axes: >],  
        [<Axes: >, <Axes: >]], dtype=object))
```



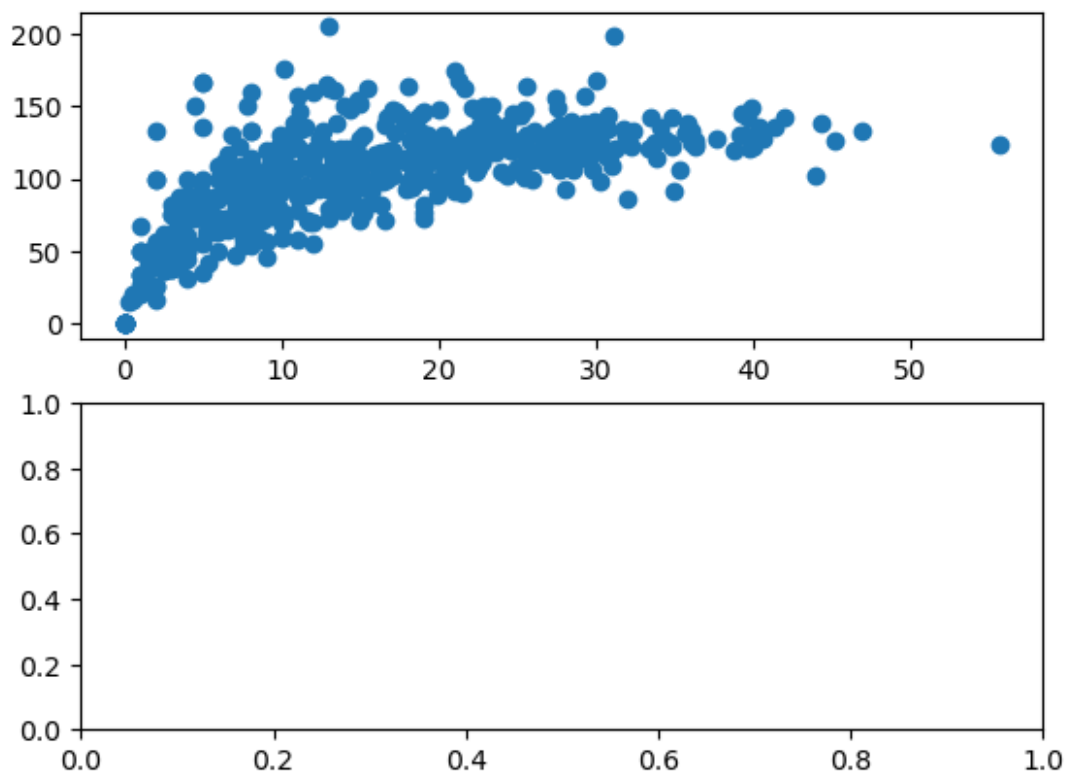
```
fig,ax=plt.subplots(nrows=2,ncols=1)
```



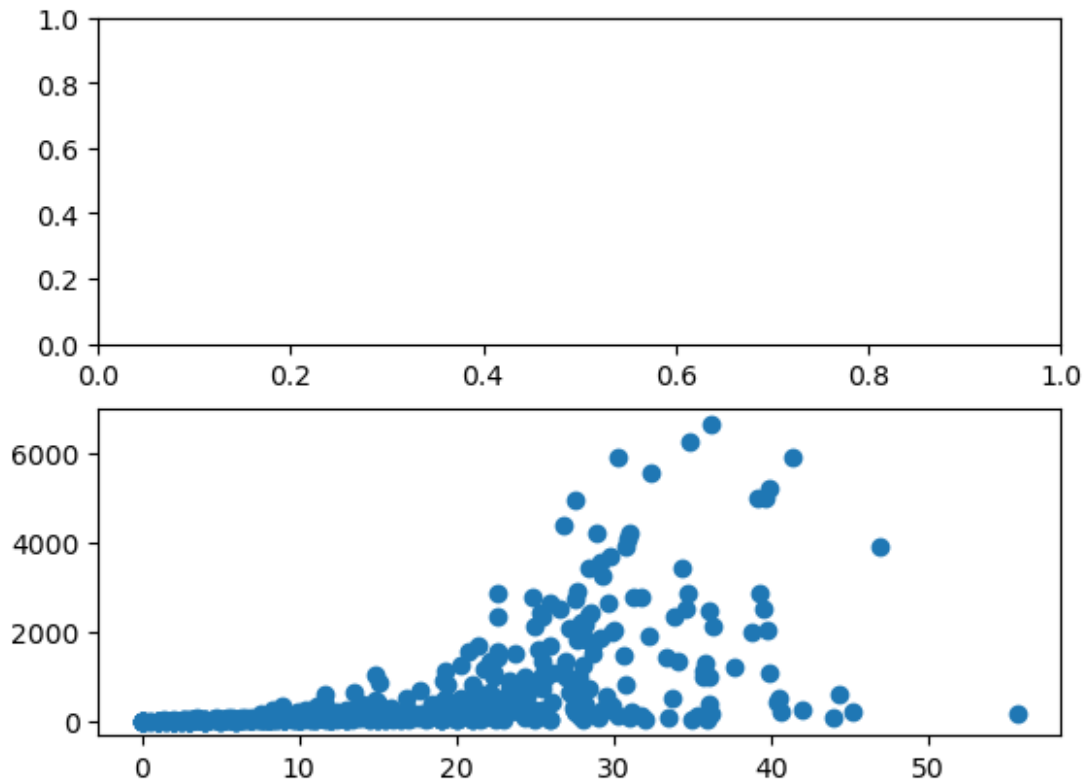
```
fig # fig obj me figure hai
```



```
ax # axes object me array store hoga jisme 2 items hoga  
array([<Axes: >, <Axes: >], dtype=object)  
fig,ax=plt.subplots(nrows=2,ncols=1)  
ax[0].scatter(batters['avg'],batters['strike_rate'])  
<matplotlib.collections.PathCollection at 0x15c9a3fb3e0>
```



```
fig,ax=plt.subplots(nrows=2,ncols=1)
ax[1].scatter(batters['avg'],batters['runs'])
<matplotlib.collections.PathCollection at 0x15c9aebdd30>
```

```
fig,ax=plt.subplots(nrows=2,ncols=1)
ax[0].scatter(batters['avg'],batters['strike_rate'])
ax[1].scatter(batters['avg'],batters['runs'])

ax[0].set_title('avg vs strike_rate')
ax[1].set_title('avg vs runs')

ax[0].set_xlabel('avg') # ye nahi bhi likh sakte hain qki dono ka x
axis same hi hai
ax[0].set_ylabel('strike_rate')

ax[1].set_xlabel('avg')
ax[1].set_ylabel('runs')

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



```
# abhi dono graph me x axis pe number dikh raha hai jabki dono ke lye
x-axis similar hai so upar bale se hum number hata
# sakte hain(by passing a parameter 'sharex')

fig,ax=plt.subplots(nrows=2,ncols=1,sharex=True,figsize=(10,6))

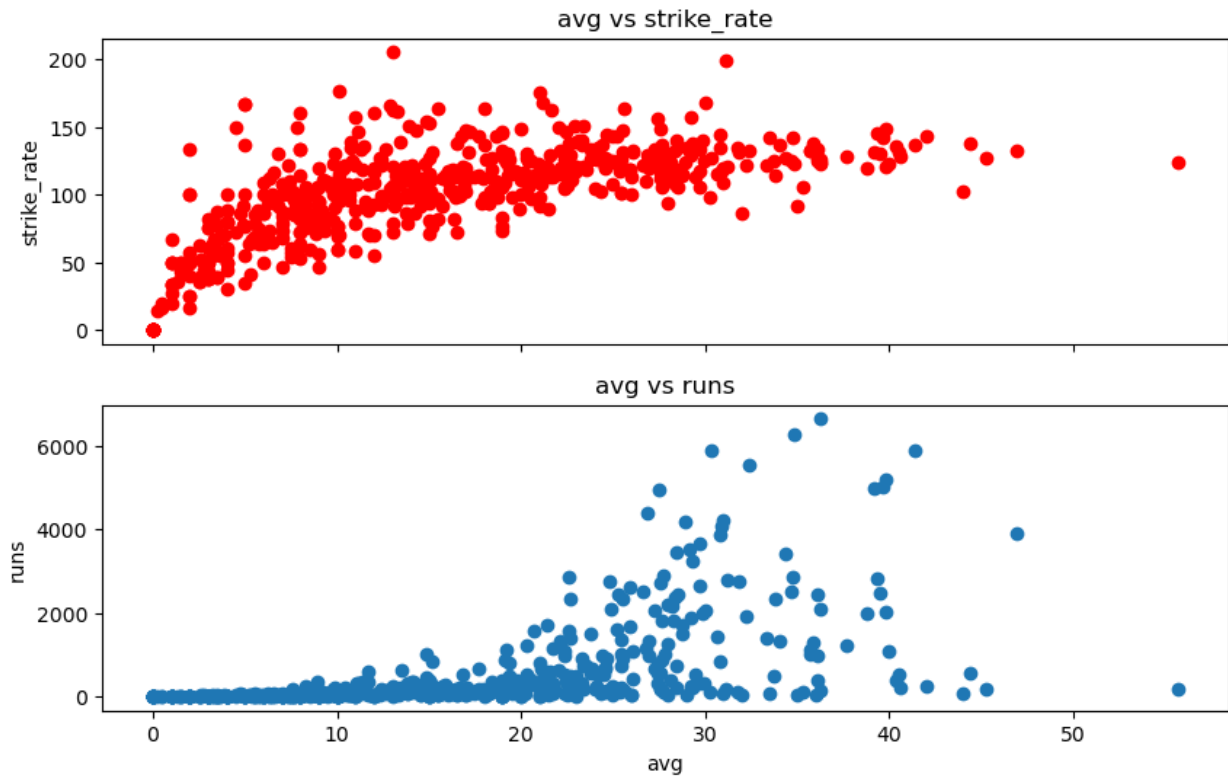
ax[0].scatter(batters['avg'],batters['strike_rate'],color='red')
ax[1].scatter(batters['avg'],batters['runs'])

ax[0].set_title('avg vs strike_rate')
ax[1].set_title('avg vs runs')

#ax[0].set_xlabel('avg')    # jab aap x-axes se number hata rahe ho to
ye remove karna hoga other wise ye graph me dikhega
ax[0].set_ylabel('strike_rate')

ax[1].set_xlabel('avg')
ax[1].set_ylabel('runs')

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



```
fig,ax=plt.subplots(nrows=2,ncols=2,sharex=True,figsize=(10,10))
ax[0].scatter(batters['avg'],batters['strike_rate'],color='red')
ax[1].scatter(batters['avg'],batters['runs'])

ax[0].set_title('avg vs strike_rate')
ax[1].set_title('avg vs runs')

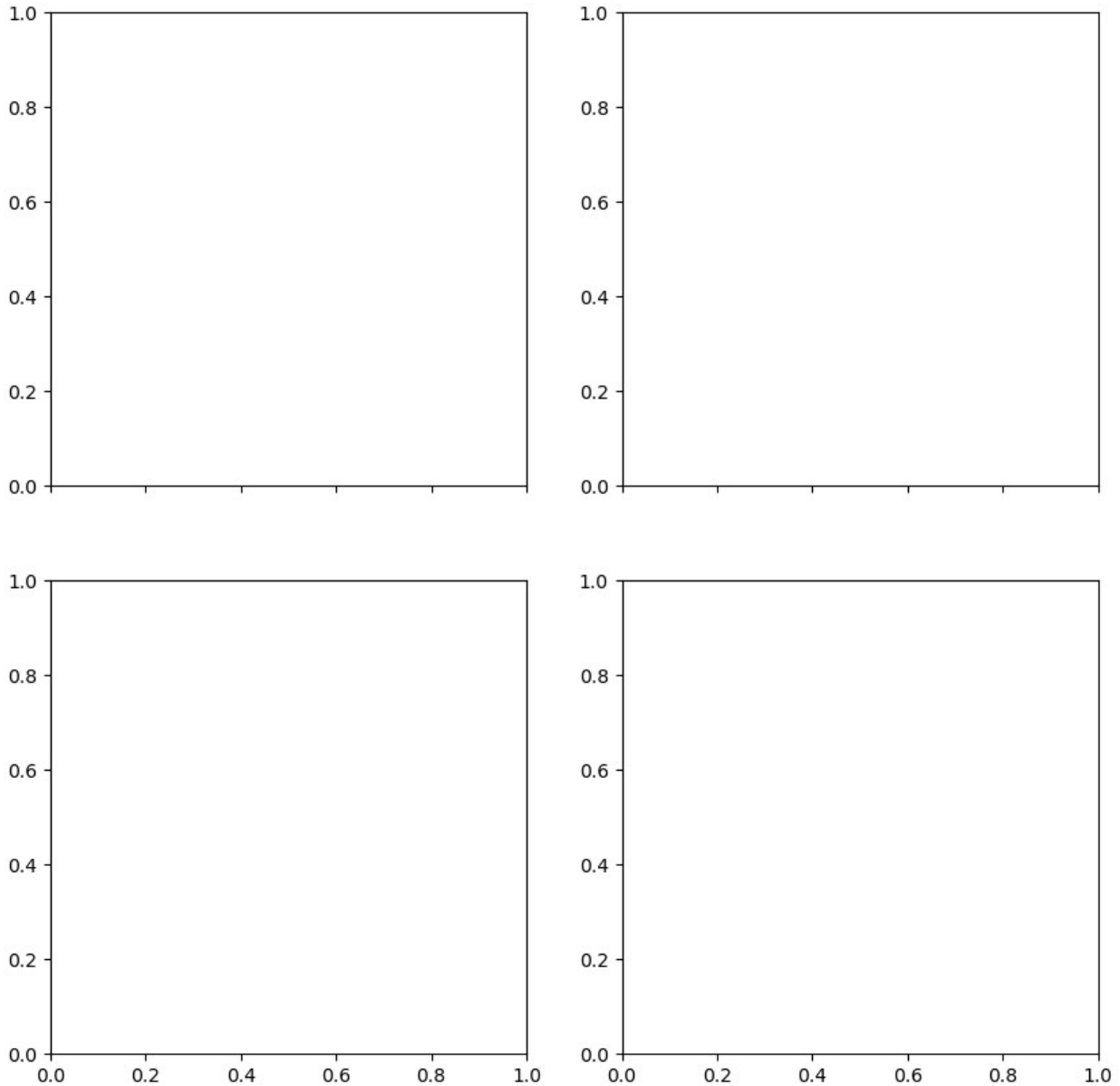
ax[0].set_xlabel('avg') # ye nahi bhi likh sakte hain qki dono ka x
axis same hi hai
ax[0].set_ylabel('strike_rate')

ax[1].set_xlabel('avg')
ax[1].set_ylabel('runs')
```

```
-----
-----
AttributeError                                Traceback (most recent call
last)
Cell In[84], line 2
      1
----> 2 fig,ax=plt.subplots(nrows=2,ncols=2,sharex=True,figsize=(10,10))
      3 ax[0].scatter(batters['avg'],batters['strike_rate'],color='red')
```

```
3 ax[1].scatter(batters['avg'],batters['runs'])
5 ax[0].set_title('avg vs strike_rate')
```

AttributeError: 'numpy.ndarray' object has no attribute 'scatter'



ax

```
# since ye ek 2D array hai to each graph ko fetch karne ke lye row and
column provide karna hoga
fig,ax=plt.subplots(nrows=2,ncols=2,sharex=True,figsize=(10,10))
ax[0][0].scatter(batters['avg'],batters['strike_rate'],color='red')
ax[0][1].scatter(batters['avg'],batters['runs'])
```

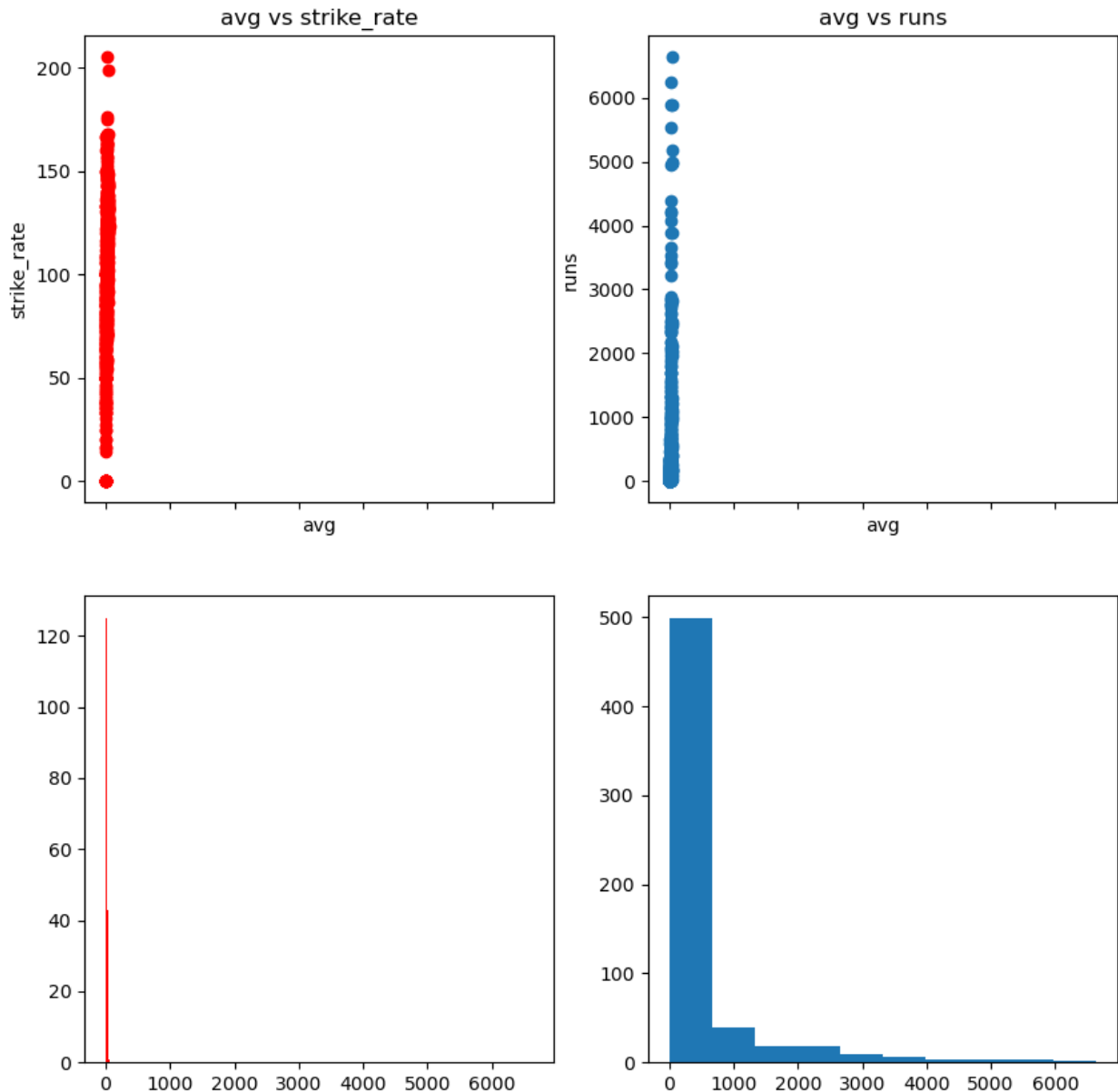
```
ax[1][0].hist(batters['avg'],color='red')
ax[1][1].hist(batters['runs'])
```

```
ax[0][0].set_title('avg vs strike_rate')
ax[0][1].set_title('avg vs runs')
```

```
ax[0][0].set_xlabel('avg')
ax[0][0].set_ylabel('strike_rate')
```

```
ax[0][1].set_xlabel('avg')
ax[0][1].set_ylabel('runs')
```

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



ye itna ajib sa islye ho gya qki humne sharex kar rakha hai

```
fig,ax=plt.subplots(nrows=2,ncols=2,figsize=(10,10))
ax[0][0].scatter(batters['avg'],batters['strike_rate'],color='red')
ax[0][1].scatter(batters['avg'],batters['runs'])

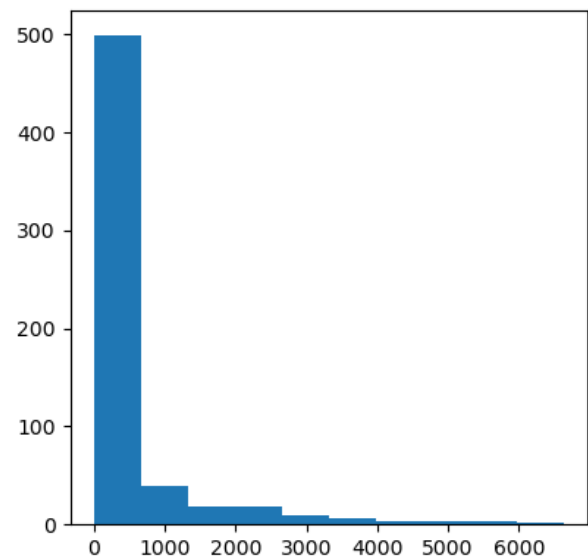
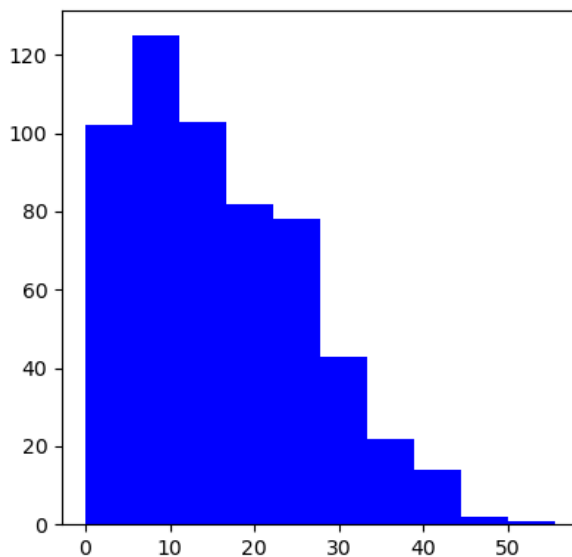
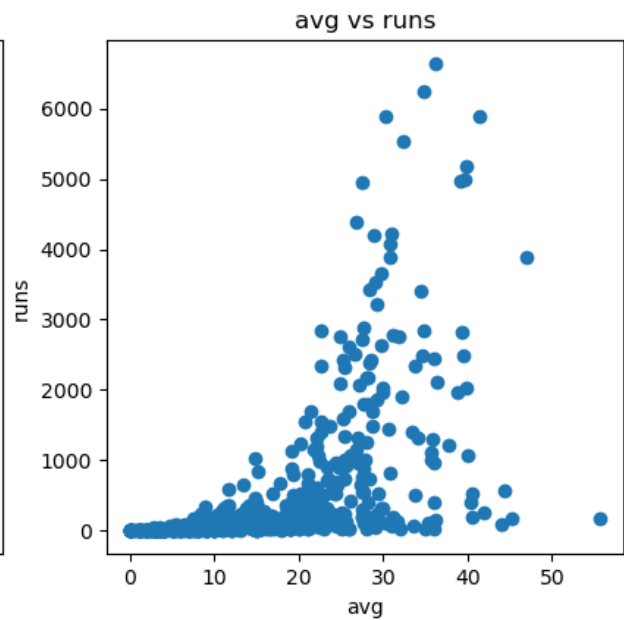
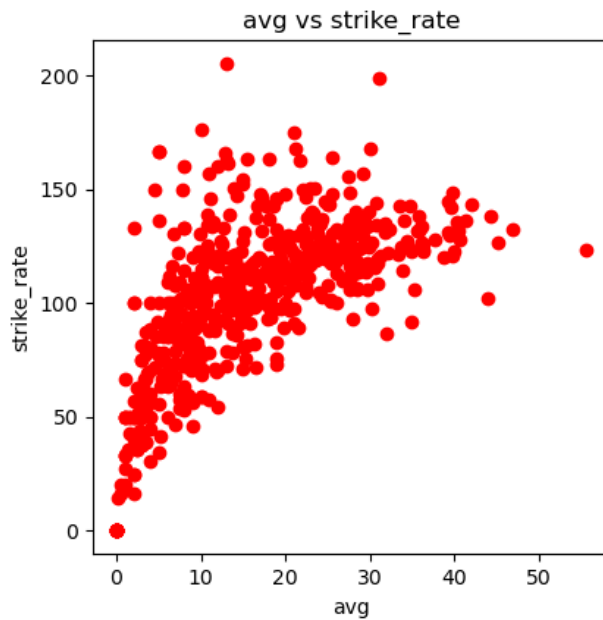
ax[1][0].hist(batters['avg'],color='blue')
ax[1][1].hist(batters['runs'])

ax[0][0].set_title('avg vs strike_rate')
ax[0][1].set_title('avg vs runs')
```

```
ax[0][0].set_xlabel('avg')
ax[0][0].set_ylabel('strike_rate')
```

```
ax[0][1].set_xlabel('avg')
ax[0][1].set_ylabel('runs')
```

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



one another way to do this

```

fig=plt.figure()
fig

<Figure size 640x480 with 0 Axes>

<Figure size 640x480 with 0 Axes>

fig=plt.figure()
ax1=fig.add_subplot(221)  # 211 --> 2-rows ,2-column, 1-> first graph
(hum comma se seperate bhi karke likh sakte hain)
ax1.scatter(batters['avg'],batters['strike_rate'],color='red')

ax2=fig.add_subplot(222)  #222 --> 2-rows ,2-column, 2-> second graph
ax2.hist(batters['avg'],color='blue')

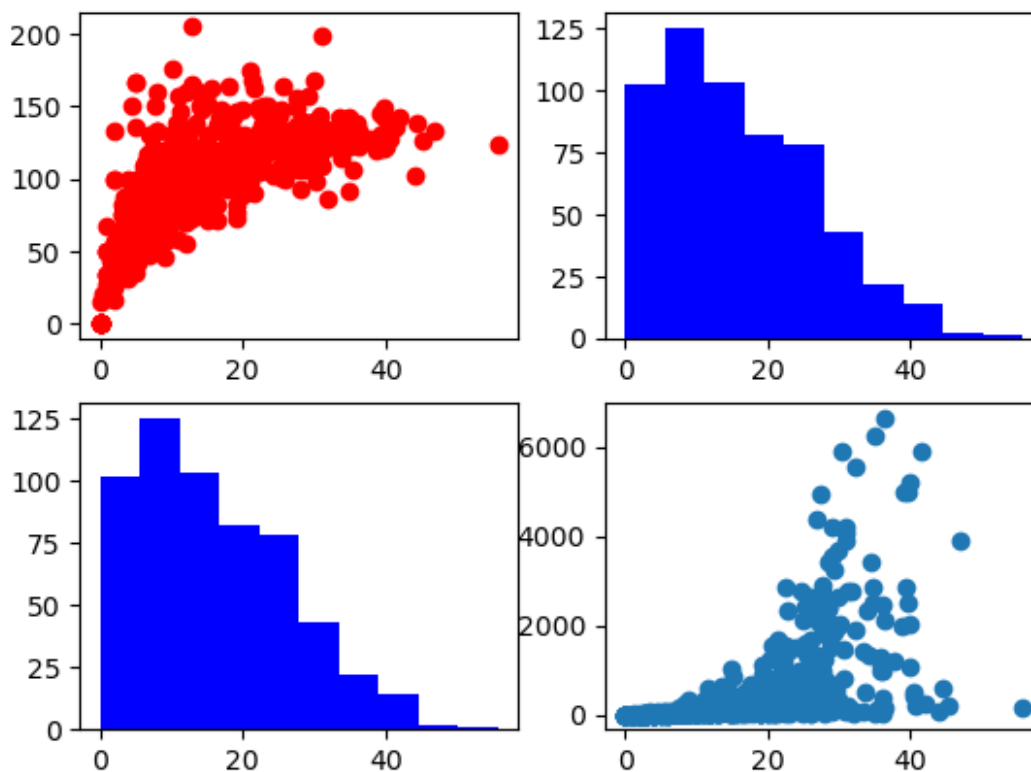
ax3=fig.add_subplot(223)  #223 ---> 2-rows , 2-columns , 3rd->graph
ax3.hist(batters['avg'],color='blue')

ax4=fig.add_subplot(224)  #224 ---> 2-rows ,2-columns , 4th- graph
ax4.scatter(batters['avg'],batters['runs'])

#(total graph = 1st number * 2nd number and 3rd number row wise graph
ke position ko batata hai)

<matplotlib.collections.PathCollection at 0x15c9f2148c0>

```



3D Scatter Plots

jab aap 3quantity ke bich me graph banate ho

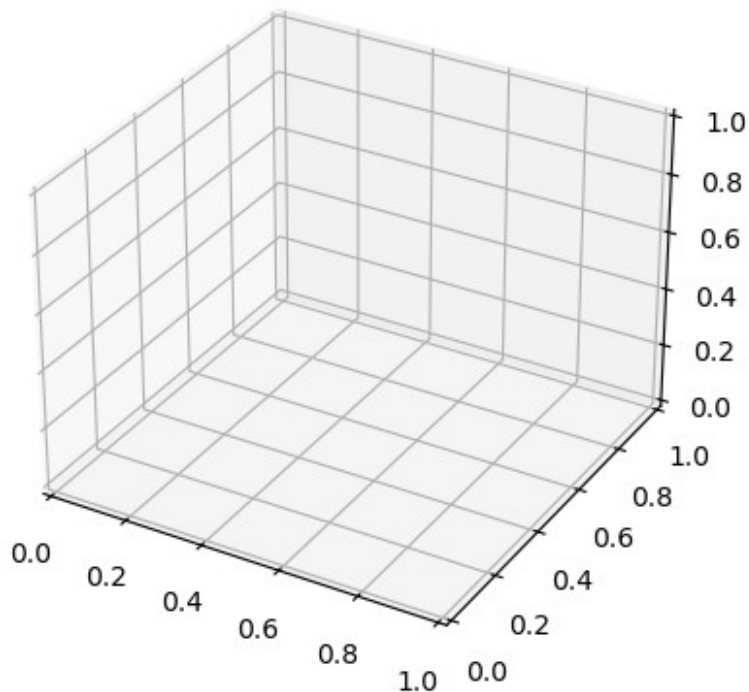
batters

	batter	runs	avg	strike_rate
0	V Kohli	6634	36.251366	125.977972
1	S Dhawan	6244	34.882682	122.840842
2	DA Warner	5883	41.429577	136.401577
3	RG Sharma	5881	30.314433	126.964594
4	SK Raina	5536	32.374269	132.535312
...
600	C Nanda	0	0.000000	0.000000
601	Akash Deep	0	0.000000	0.000000
602	S Ladda	0	0.000000	0.000000
603	V Pratap Singh	0	0.000000	0.000000
604	S Lamichhane	0	0.000000	0.000000

[605 rows x 4 columns]

fig=plt.figure()

ax=plt.subplot(projection='3d')



```

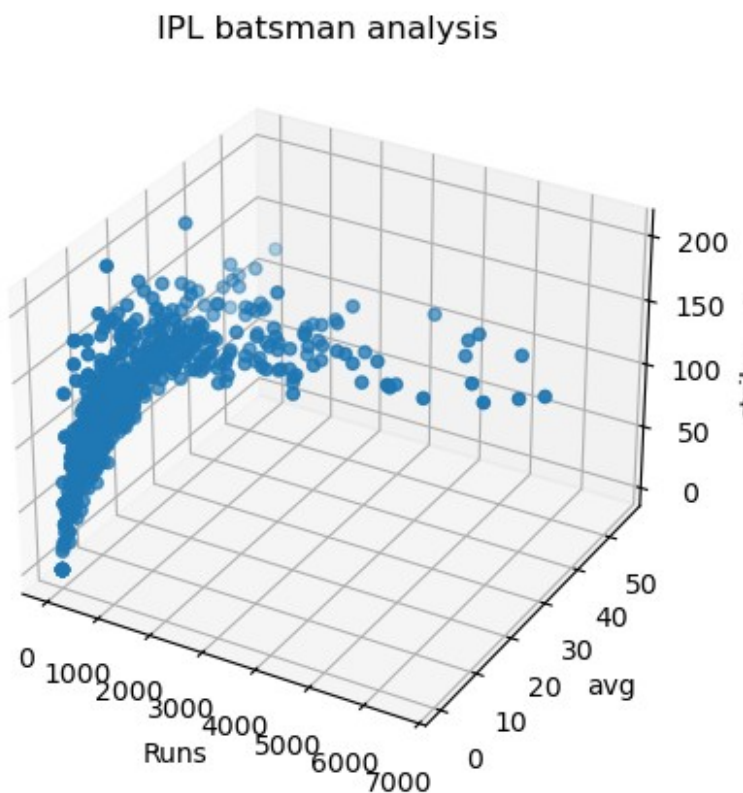
fig=plt.figure()

ax=plt.subplot(projection='3d')
ax.scatter3D(batters['runs'],batters['avg'],batters['strike_rate'])

ax.set_title('IPL batsman analysis')
ax.set_xlabel('Runs')
ax.set_ylabel('avg')
ax.set_zlabel('strike_rate')

# baki sab kuch pichle jaisa kar sakte hain
Text(0.5, 0, 'strike_rate')

```



3D Line Plot

- jo bhi scatter plot me point aata hai usko line se connect kar deta hai

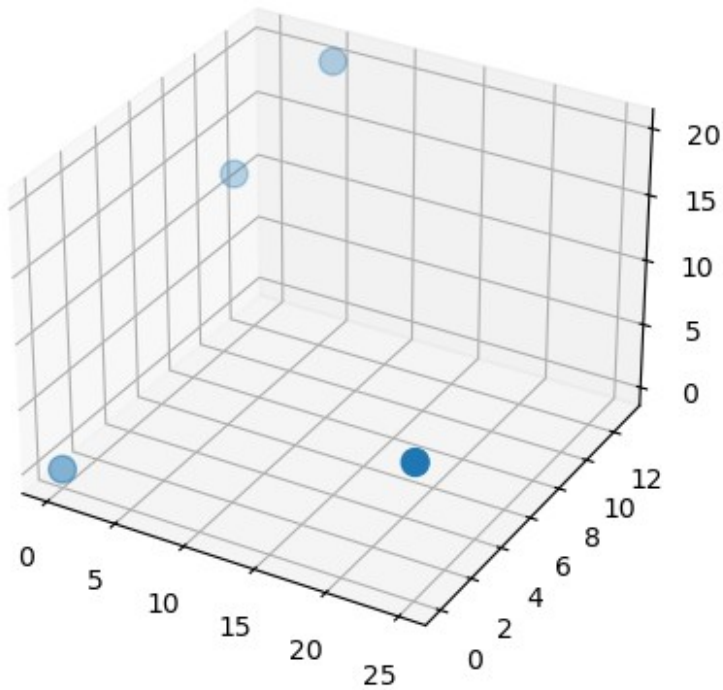
```

x = [0,1,5,25]
y = [0,10,13,0]
z = [0,13,20,9]

fig=plt.figure()

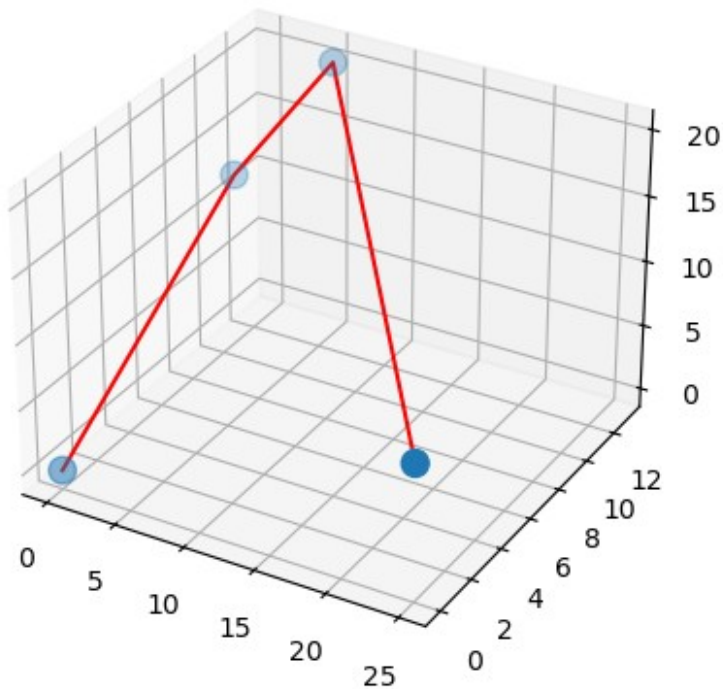
```

```
ax=plt.subplot(projection='3d')
ax.scatter3D(x,y,z,s=[100,100,100,100]) # s is for control the
scatter(point) size
<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x15c9d8abdd0>
```



```
x = [0,1,5,25]
y = [0,10,13,0]
z = [0,13,20,9]

fig=plt.figure()
ax=plt.subplot(projection='3d')
ax.scatter3D(x,y,z,s=[100,100,100,100])
ax.plot3D(x,y,z,color='red')
[<mpl_toolkits.mplot3d.art3d.Line3D at 0x15c9cc9da00>]
```



3D surface plot

```
x = np.linspace(-10,10,100)
y = np.linspace(-10,10,100)

xx, yy = np.meshgrid(x,y)
xx.shape
# meshgrid simply ek grid ready kar
deta hai
(100, 100)

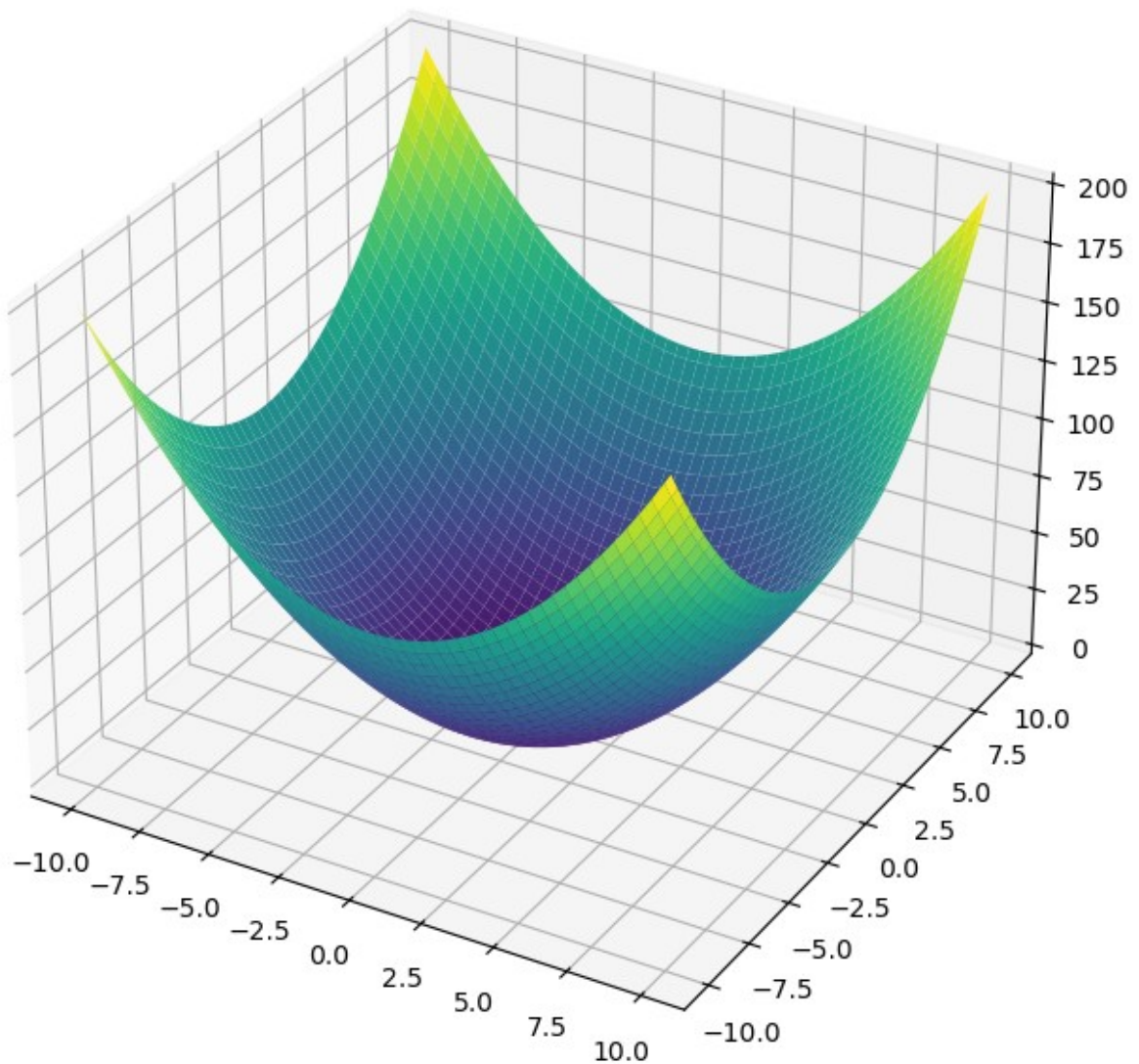
xx,yy = np.meshgrid(x,y)
xx.shape
(100, 100)

yy.shape
(100, 100)

# calculate z
# lets we are trying to crate a graph of  $z=a^2+b^2$ 

z = xx**2 + yy**2
z.shape
(100, 100)
```

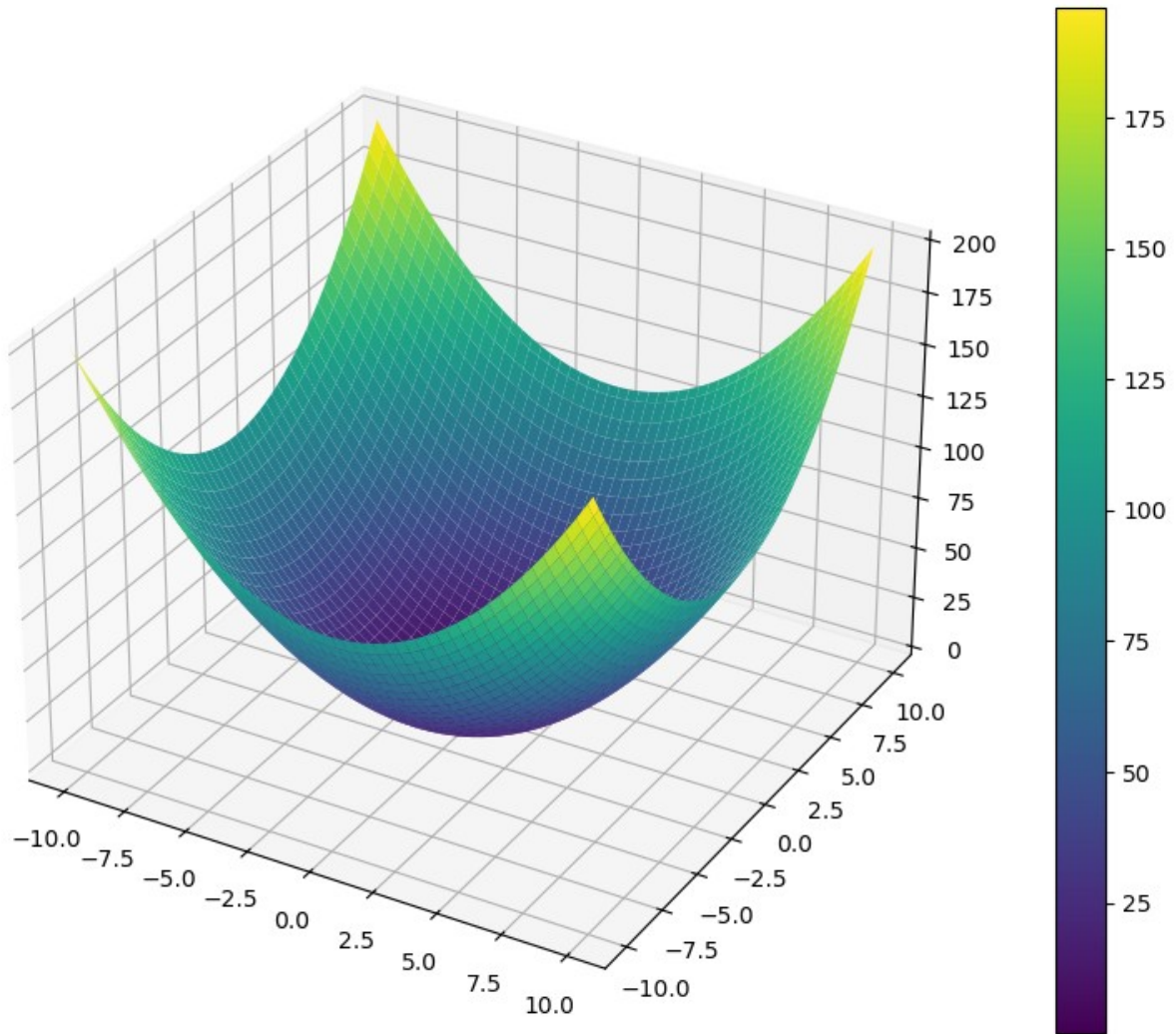
```
fig = plt.figure(figsize=(12,8))  
ax = plt.subplot(projection='3d')  
ax.plot_surface(xx,yy,z,cmap='viridis')  
<mpl_toolkits.mplot3d.art3d.Poly3DCollection at 0x15c9ff998e0>
```



```
fig = plt.figure(figsize=(12,8))  
ax = plt.subplot(projection='3d')
```

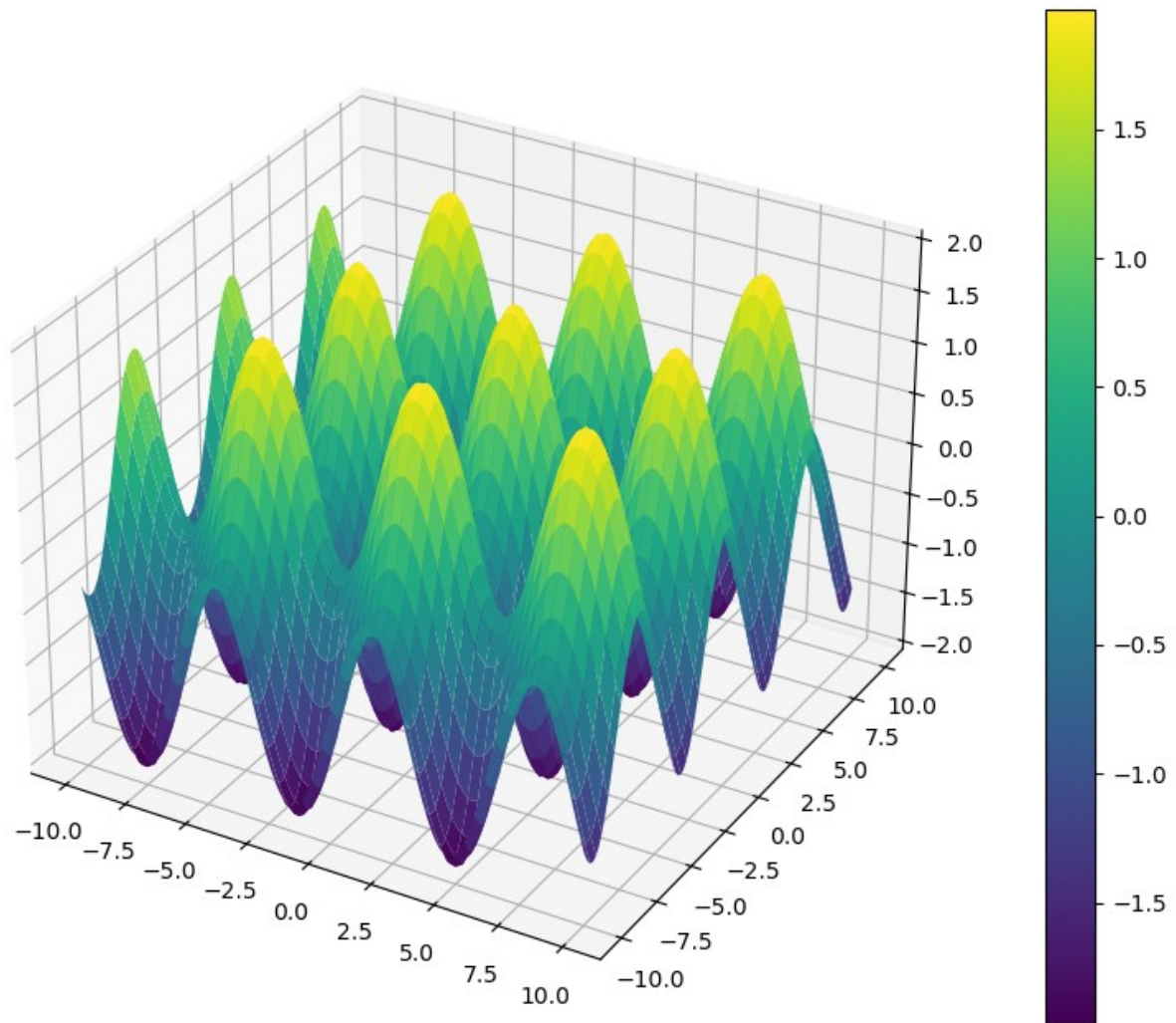
```
p=ax.plot_surface(xx,yy,z,cmap='viridis')
fig.colorbar(p)

<matplotlib.colorbar.Colorbar at 0x15c9ff996a0>
```



```
# just try on another function
z= np.sin(xx) + np.cos(yy)
fig = plt.figure(figsize=(12,8))
ax = plt.subplot(projection='3d')
p=ax.plot_surface(xx,yy,z,cmap='viridis')
fig.colorbar(p)

<matplotlib.colorbar.Colorbar at 0x15c9ff99f10>
```

```
# just try on another function
```

```
z = np.sin(xx) + np.log(yy)
```

```
fig = plt.figure(figsize=(12,8))
```

```
ax = plt.subplot(projection='3d')
```

```
p=ax.plot_surface(xx,yy,z,cmap='viridis')
```

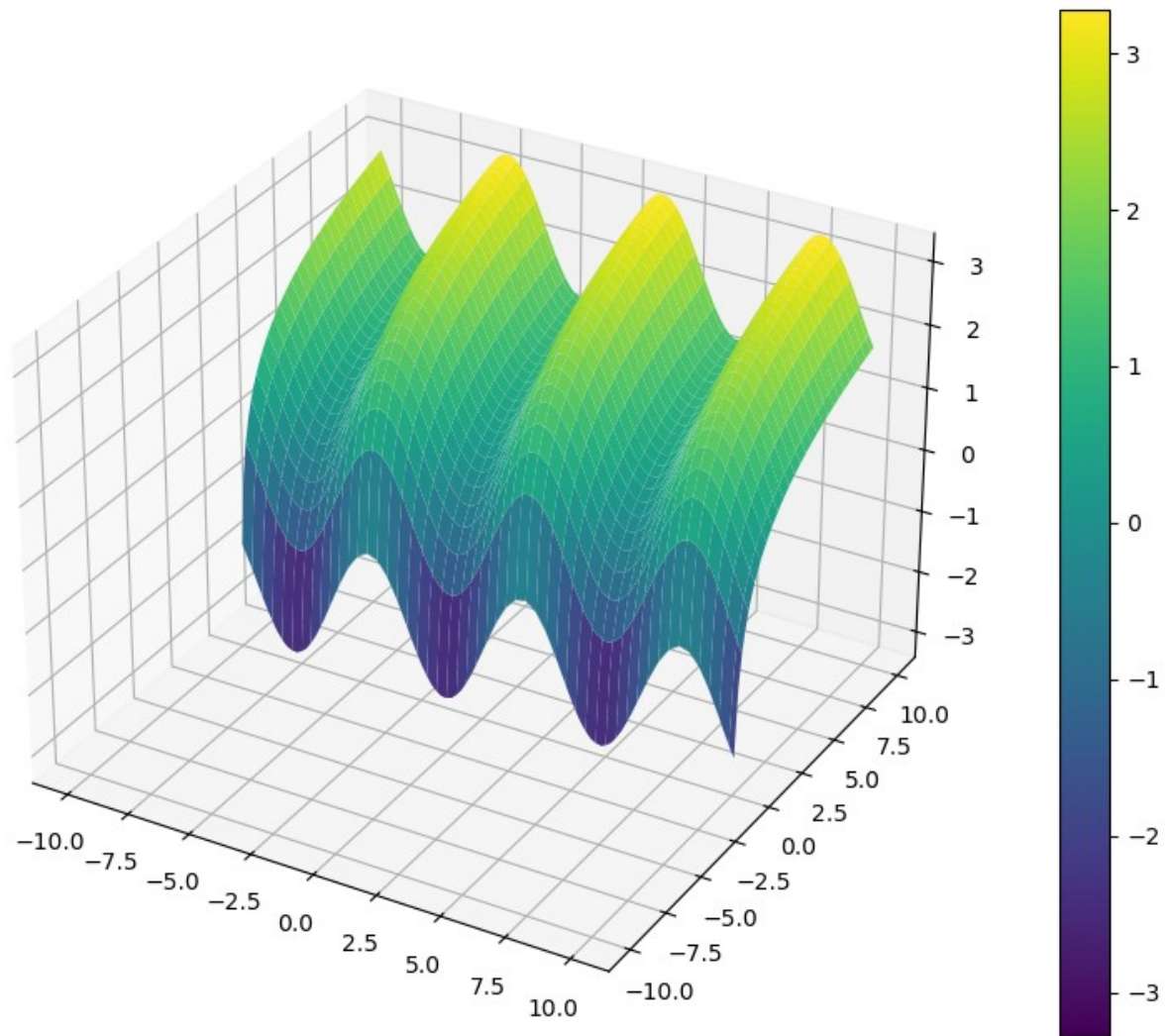
```
fig.colorbar(p)
```

```
C:\Users\jayra\AppData\Local\Temp\ipykernel_20392\3024872249.py:2:
```

```
RuntimeWarning: invalid value encountered in log
```

```
z = np.sin(xx) + np.log(yy)
```

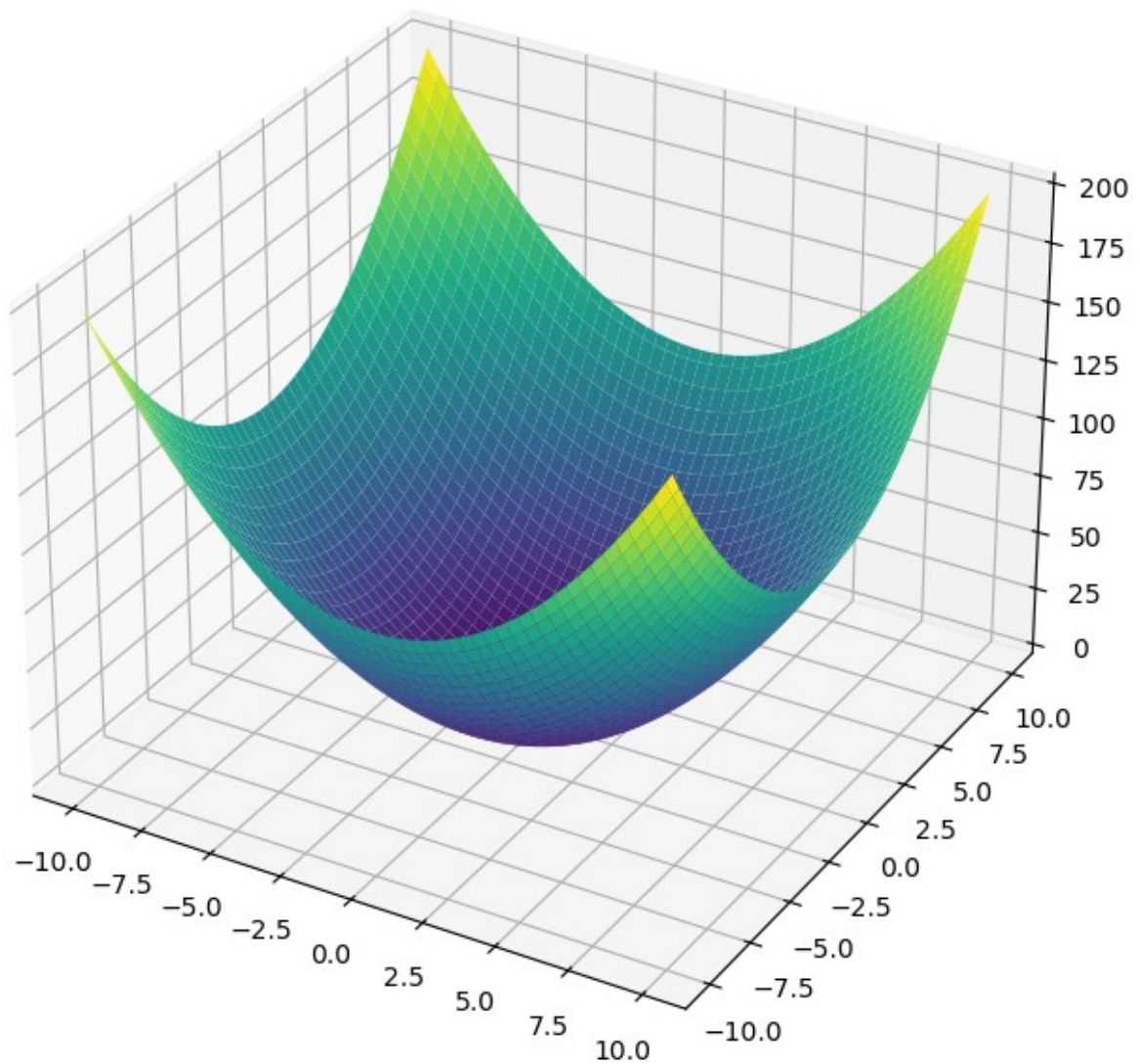
```
<matplotlib.colorbar.Colorbar at 0x15c9e050980>
```



Contour graph

- 3d graph ko 2d me represent karta hai

```
z = xx**2 + yy**2
fig = plt.figure(figsize=(12,8))
ax = plt.subplot(projection='3d')
ax.plot_surface(xx,yy,z,cmap='viridis')
<mpl_toolkits.mplot3d.art3d.Poly3DCollection at 0x15ca3e958e0>
```

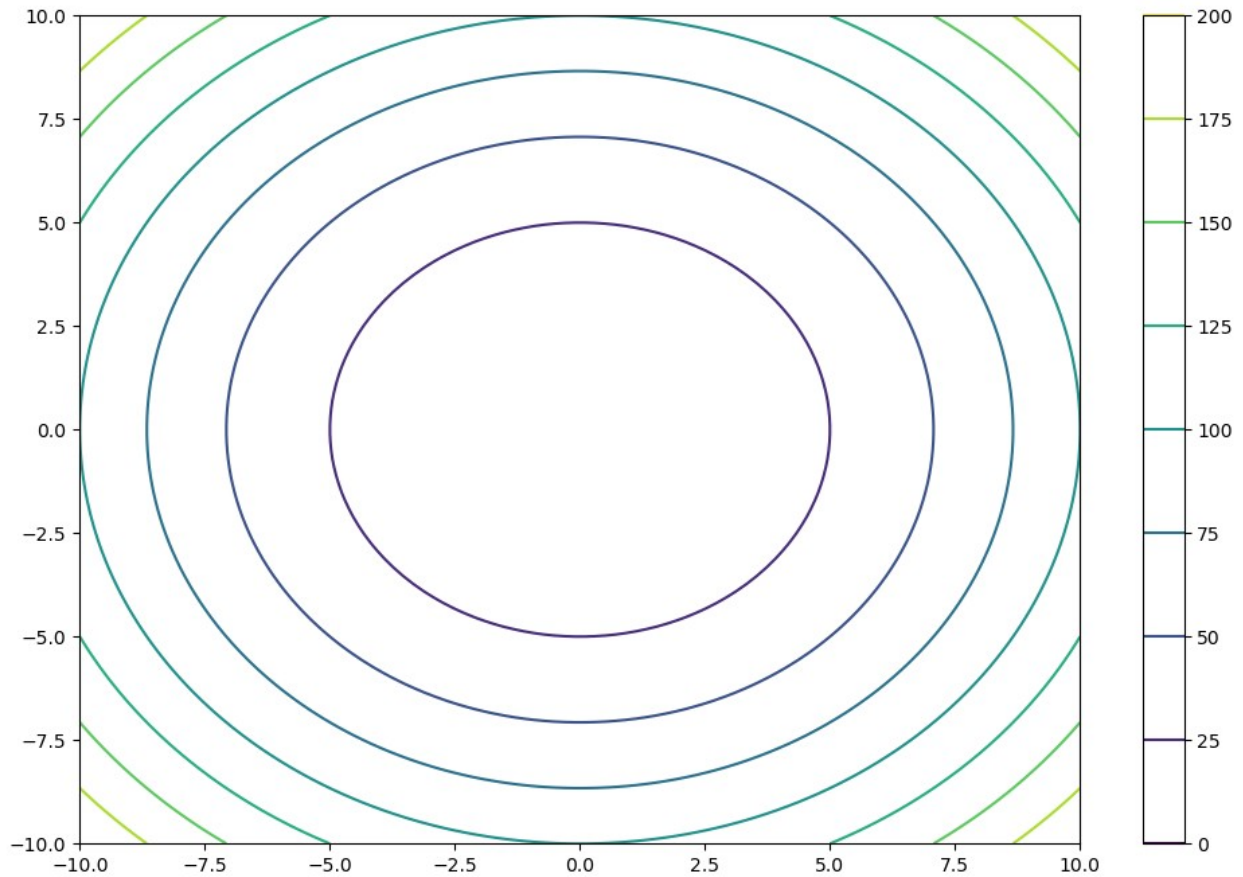



```
#lets do
fig = plt.figure(figsize=(12,8))
ax = plt.subplot()
p = ax.contour(xx,yy,z,cmap='viridis')
fig.colorbar(p)
```

isko aaise dekho ki aap 3D surface garap ke top pe khade ho and andar ki taraf dekh rahe ho to yellow wali jo line dikh rahi hai wo top ko

and purple
bali bottom ko dikha raha hai

<matplotlib.colorbar.Colorbar at 0x15ca2ad9100>

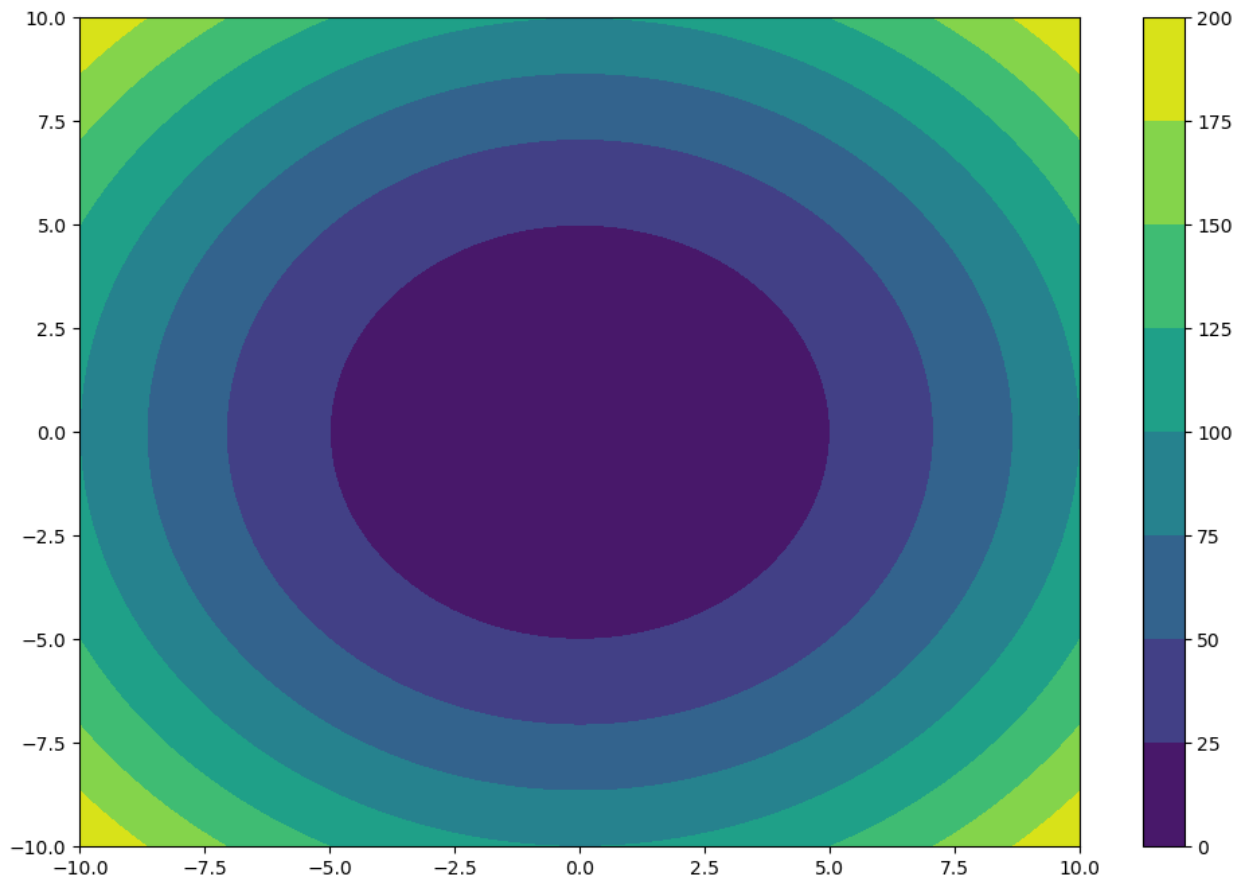


```
# filled contour graph
fig = plt.figure(figsize=(12,8))

ax = plt.subplot()

p = ax.contourf(xx,yy,z,cmap='viridis') # contourf -> contour fill
fig.colorbar(p)
```

<matplotlib.colorbar.Colorbar at 0x15ca3ada540>



```
# another example
```

```
z = np.sin(xx) + np.log(yy)
```

```
fig = plt.figure(figsize=(12,8))
```

```
ax = plt.subplot(projection='3d')
```

```
p=ax.plot_surface(xx,yy,z,cmap='viridis')
```

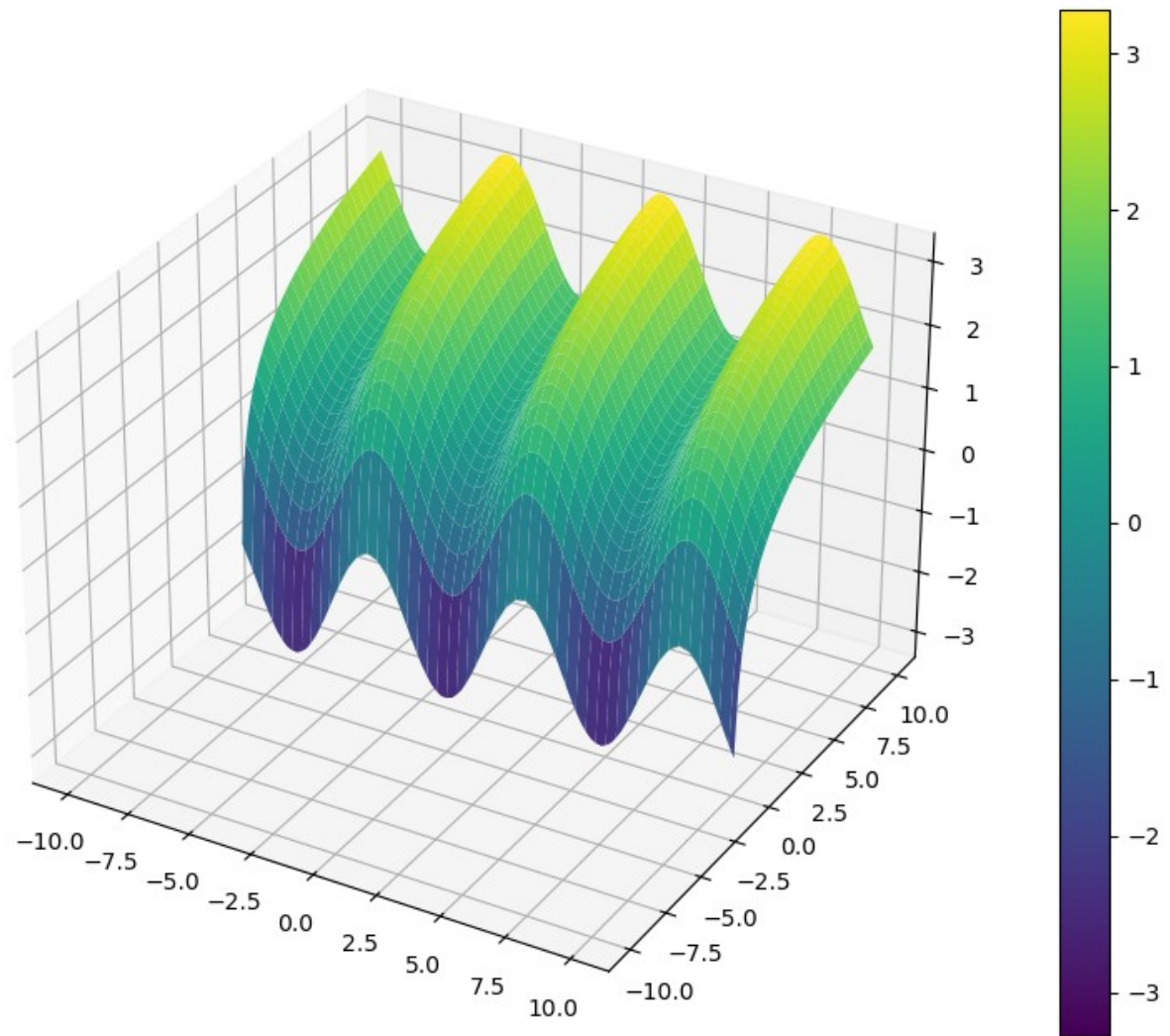
```
fig.colorbar(p)
```

```
C:\Users\jayra\AppData\Local\Temp\ipykernel_20392\3018327182.py:1:
```

```
RuntimeWarning: invalid value encountered in log
```

```
z = np.sin(xx) + np.log(yy)
```

```
<matplotlib.colorbar.Colorbar at 0x15c9a2bd2e0>
```



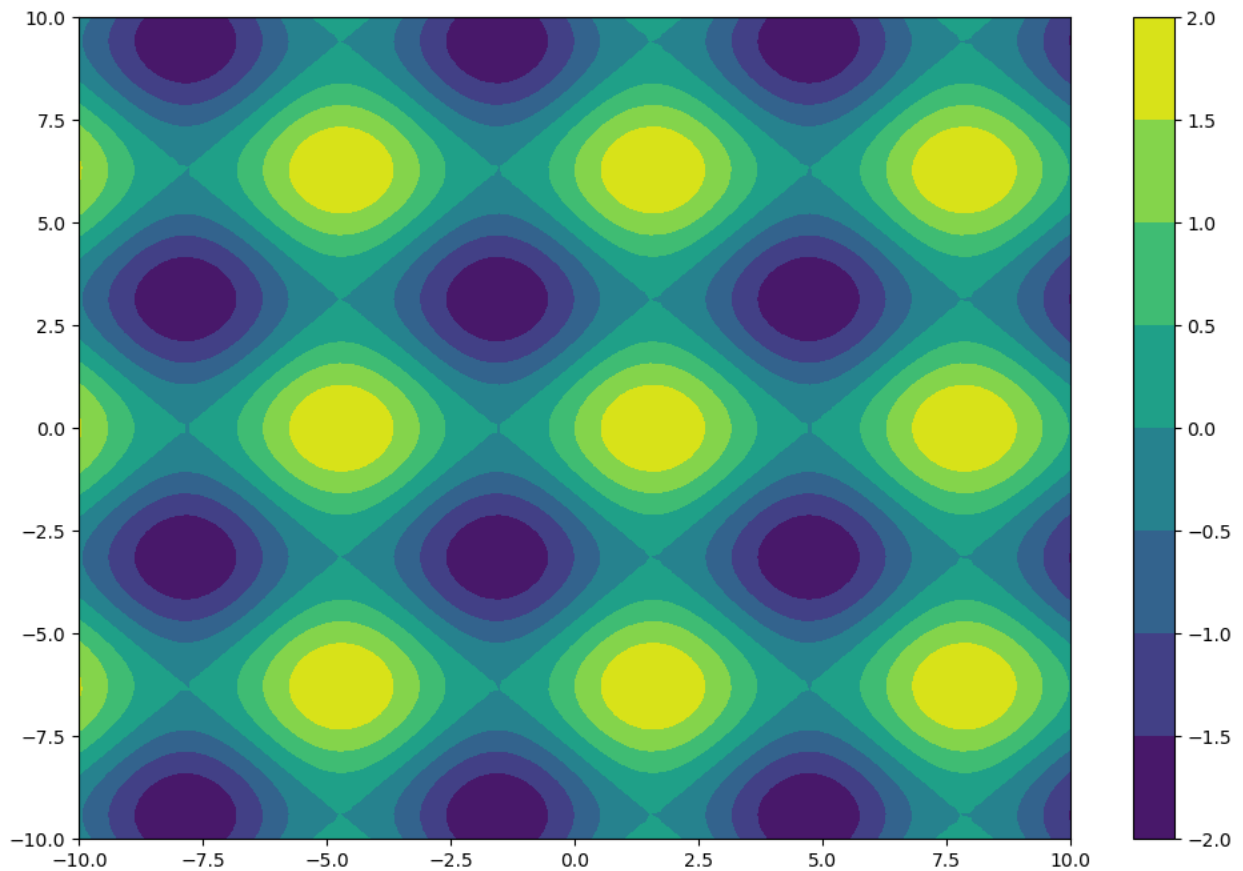
```
# lets try
z = np.sin(xx) + np.cos(yy)

fig = plt.figure(figsize=(12,8))

ax = plt.subplot()

p=ax.contourf(xx,yy,z,cmap='viridis')
fig.colorbar(p)

<matplotlib.colorbar.Colorbar at 0x15ca242a630>
```



Heatmap

- basically grid ko graph me convert karte hain

ek aaisa graph karna chahate hain jo ki show karega ki ipl ke har over ke kon se ball pe kitna six lagta hai

```
delivery = pd.read_csv('IPL_Ball_by_Ball_2008_2022.csv')
delivery.head()
```

	ID	innings	overs	ballnumber	batter	bowler	\
0	1312200	1	0	1	YBK Jaiswal	Mohammed Shami	
1	1312200	1	0	2	YBK Jaiswal	Mohammed Shami	
2	1312200	1	0	3	JC Buttler	Mohammed Shami	
3	1312200	1	0	4	YBK Jaiswal	Mohammed Shami	
4	1312200	1	0	5	YBK Jaiswal	Mohammed Shami	

	non-striker	extra_type	batsman_run	extras_run	total_run
non_boundary					
0	JC Buttler	NaN	0	0	0
0					
1	JC Buttler	legbyes	0	1	1

```

0
2   YBK Jaiswal      NaN      1      0      1
0
3   JC Buttler      NaN      0      0      0
0
4   JC Buttler      NaN      0      0      0
0

    isWicketDelivery player_out kind fielders_involved
BattingTeam
0              0      NaN  NaN      NaN  Rajasthan
Royals
1              0      NaN  NaN      NaN  Rajasthan
Royals
2              0      NaN  NaN      NaN  Rajasthan
Royals
3              0      NaN  NaN      NaN  Rajasthan
Royals
4              0      NaN  NaN      NaN  Rajasthan
Royals

delivery['ballnumber'].unique() # kuch over me 10 balls bhi dala gya
hai may be wide or noball ho sakta hai

array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10], dtype=int64)

# but hume only 6 ball hi chahiye
(delivery['ballnumber'].isin([1,2,3,4,5,6])) &
(delivery['batsman_run'] == 6)

0      False
1      False
2      False
3      False
4      False
...
225949  False
225950  False
225951  False
225952  False
225953  False
Length: 225954, dtype: bool

temp_df = delivery[(delivery['ballnumber'].isin([1,2,3,4,5,6])) &
(delivery['batsman_run'] == 6)]
temp_df

      ID  innings  overs  ballnumber  batter
bowler \
16     1312200      1      2          5  YBK Jaiswal  Mohammed
Shami

```

22	1312200	1	3	5	YBK Jaiswal	Yash
Dayal						
103	1312200	1	17	2	TA Boult	R Sai
Kishore						
107	1312200	1	17	6	OC McCoy	R Sai
Kishore						
142	1312200	2	3	5	MS Wade	M Prasidh
Krishna						
...
...						
225848	335982	1	19	2	BB McCullum	P
Kumar						
225852	335982	1	19	6	BB McCullum	P
Kumar						
225882	335982	2	4	4	JH Kallis	AB
Agarkar						
225932	335982	2	12	1	P Kumar	LR
Shukla						
225944	335982	2	13	6	P Kumar	SC
Ganguly						
	non-striker	extra_type	batsman_run	extras_run	total_run	
\						
16	JC Buttler	NaN	6	0	6	
22	JC Buttler	NaN	6	0	6	
103	R Parag	NaN	6	0	6	
107	R Parag	NaN	6	0	6	
142	Shubman Gill	NaN	6	0	6	
...	
225848	Mohammad Hafeez	NaN	6	0	6	
225852	Mohammad Hafeez	NaN	6	0	6	
225882	W Jaffer	NaN	6	0	6	
225932	Z Khan	NaN	6	0	6	
225944	SB Joshi	NaN	6	0	6	
	non_boundary	isWicketDelivery	player_out	kind		
fielders_involved	\					
16	0	0	NaN	NaN		
NaN						

22	0	0	NaN	NaN
NaN				
103	0	0	NaN	NaN
NaN				
107	0	0	NaN	NaN
NaN				
142	0	0	NaN	NaN
NaN				
...
..				
225848	0	0	NaN	NaN
NaN				
225852	0	0	NaN	NaN
NaN				
225882	0	0	NaN	NaN
NaN				
225932	0	0	NaN	NaN
NaN				
225944	0	0	NaN	NaN
NaN				

	BattingTeam
16	Rajasthan Royals
22	Rajasthan Royals
103	Rajasthan Royals
107	Rajasthan Royals
142	Gujarat Titans
...	...
225848	Kolkata Knight Riders
225852	Kolkata Knight Riders
225882	Royal Challengers Bangalore
225932	Royal Challengers Bangalore
225944	Royal Challengers Bangalore

[10276 rows x 17 columns]

```
grid=temp_df.pivot_table(index='overs',columns='ballnumber' ,values='batsman_run' ,aggfunc='count')
```

grid

ballnumber	1	2	3	4	5	6
overs						
0	9	17	31	39	33	27
1	31	40	49	56	58	54
2	75	62	70	72	58	76
3	60	74	74	103	74	71
4	71	76	112	80	81	72
5	77	102	63	86	78	80
6	34	56	49	59	64	38

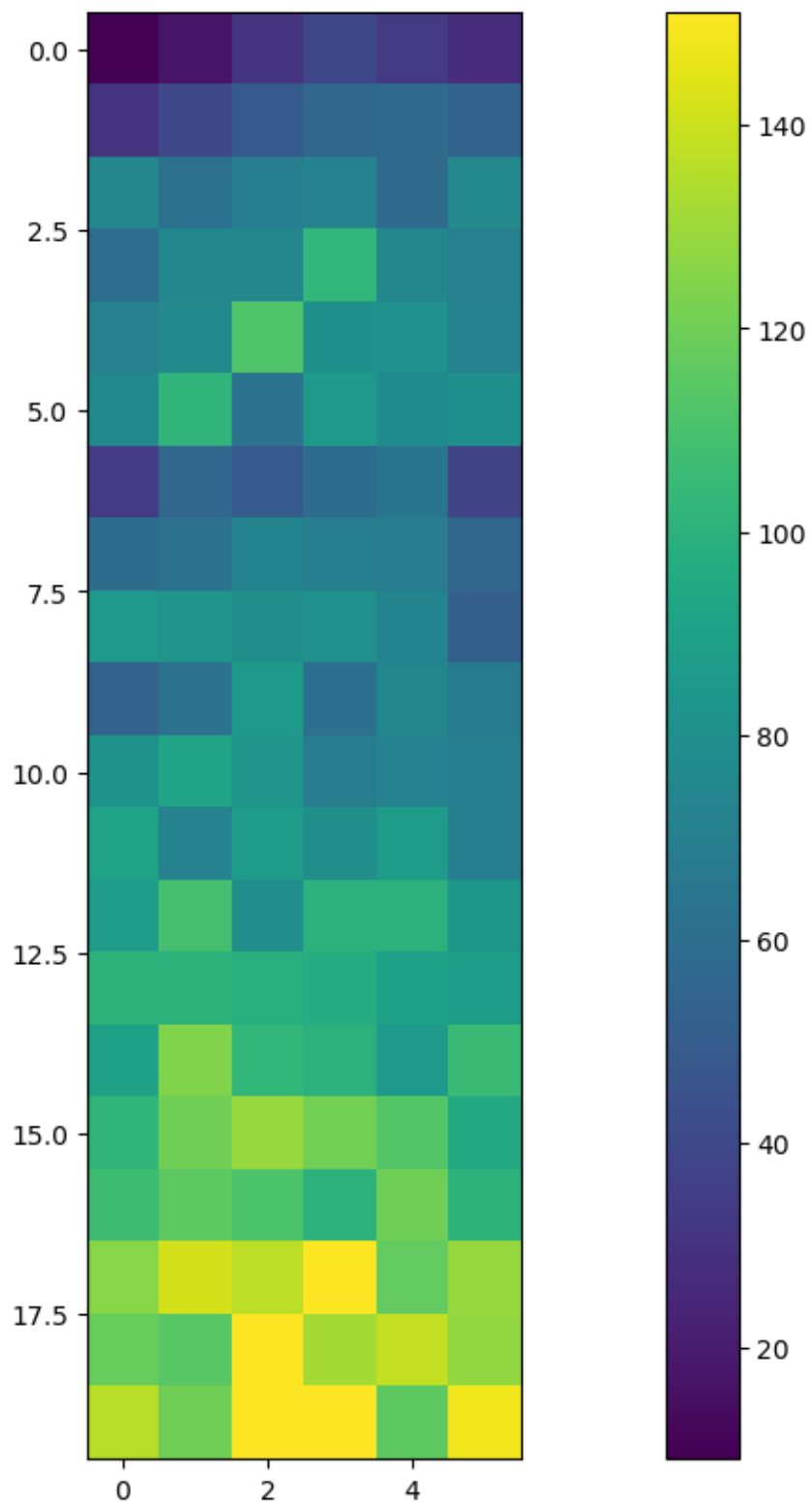
7	59	62	73	70	69	56
8	86	83	79	81	73	52
9	54	62	86	61	74	67
10	82	92	83	69	72	70
11	91	72	87	79	87	70
12	87	109	79	100	100	84
13	101	101	99	97	90	88
14	90	124	103	100	86	106
15	102	120	129	121	113	96
16	107	115	111	100	120	101
17	126	142	137	151	117	129
18	118	114	151	132	138	128
19	136	120	151	151	116	148

```
plt.figure(figsize=(20,10))
```

```
plt.imshow(grid)
```

```
plt.colorbar()
```

```
<matplotlib.colorbar.Colorbar at 0x15ca62475c0>
```

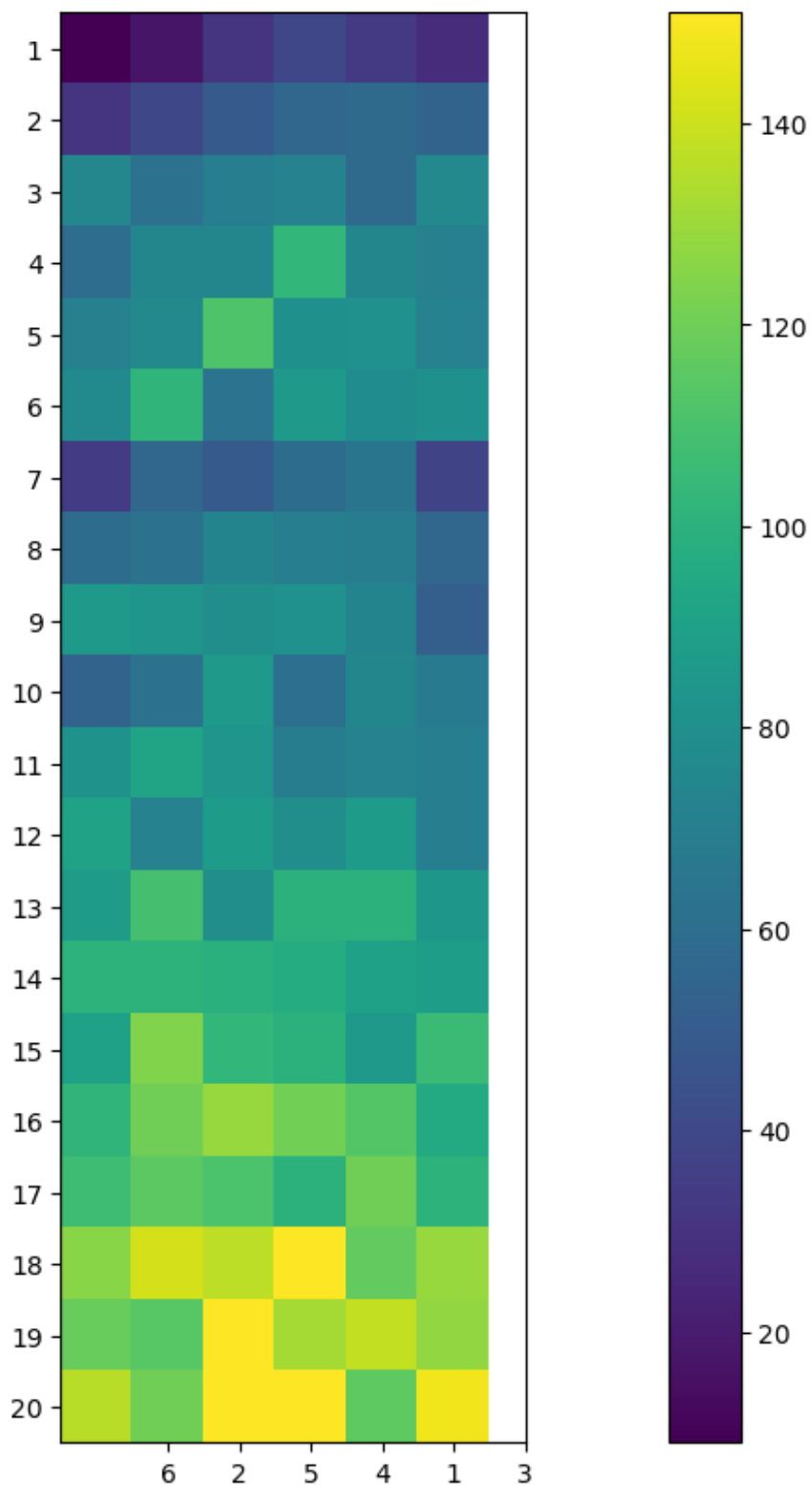


```
plt.figure(figsize=(20,10))
```

```
plt.imshow(grid)
plt.yticks(delivery['overs'].unique(),list(range(1,21)))

plt.xticks(temp_df['ballnumber'].unique(),list(range(1,7)))
plt.colorbar()

<matplotlib.colorbar.Colorbar at 0x15ca63ec680>
```



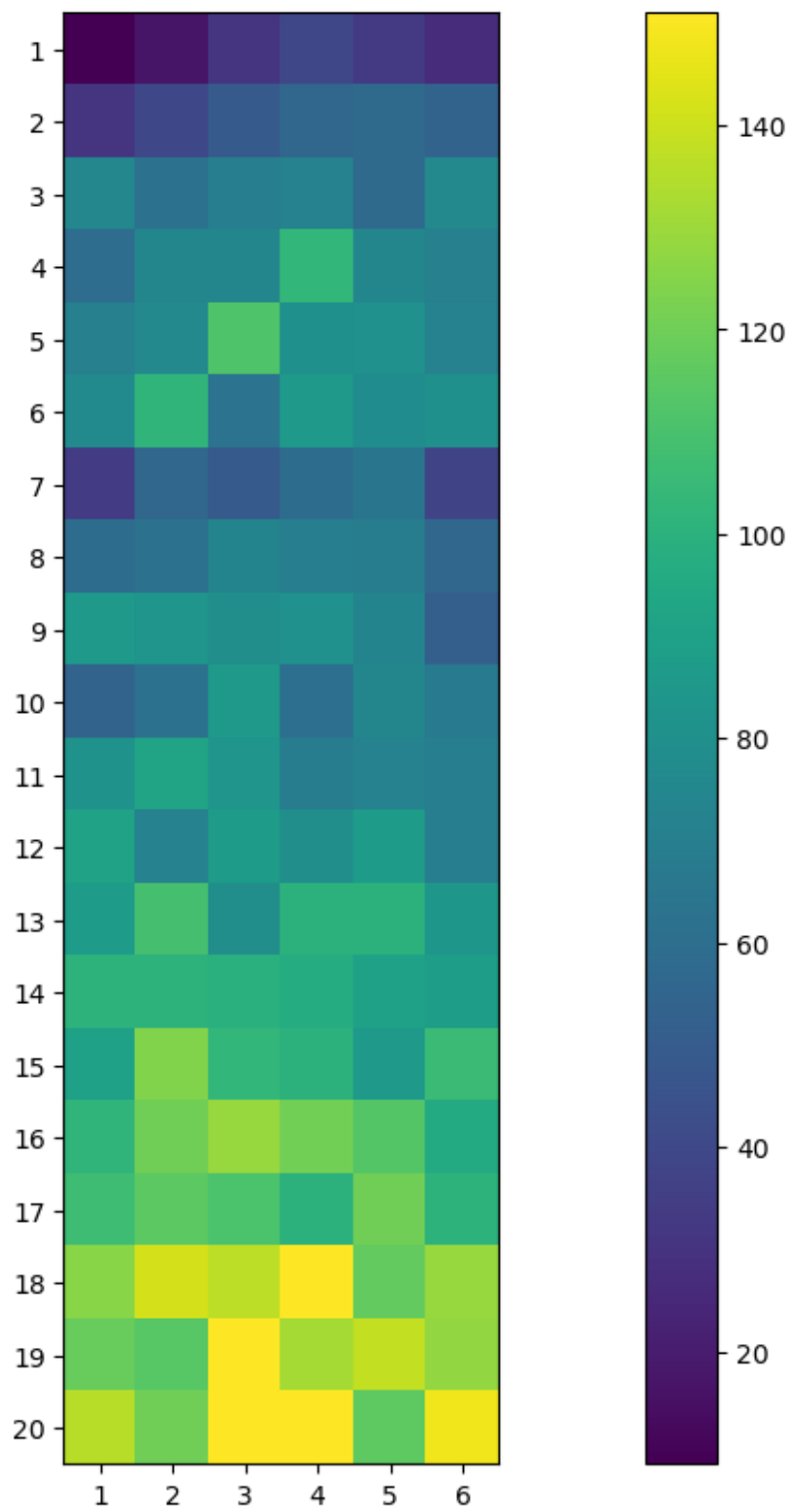
x ticks me order change ho gya may be data me hi order change hoga

```
# lets do another method
plt.figure(figsize=(20,10))

plt.imshow(grid)
plt.yticks(delivery['overs'].unique(),list(range(1,21)))

plt.xticks(np.arange(0,6),list(range(1,7)))
plt.colorbar()

<matplotlib.colorbar.Colorbar at 0x15ca66415b0>
```



DataFrame Plot()

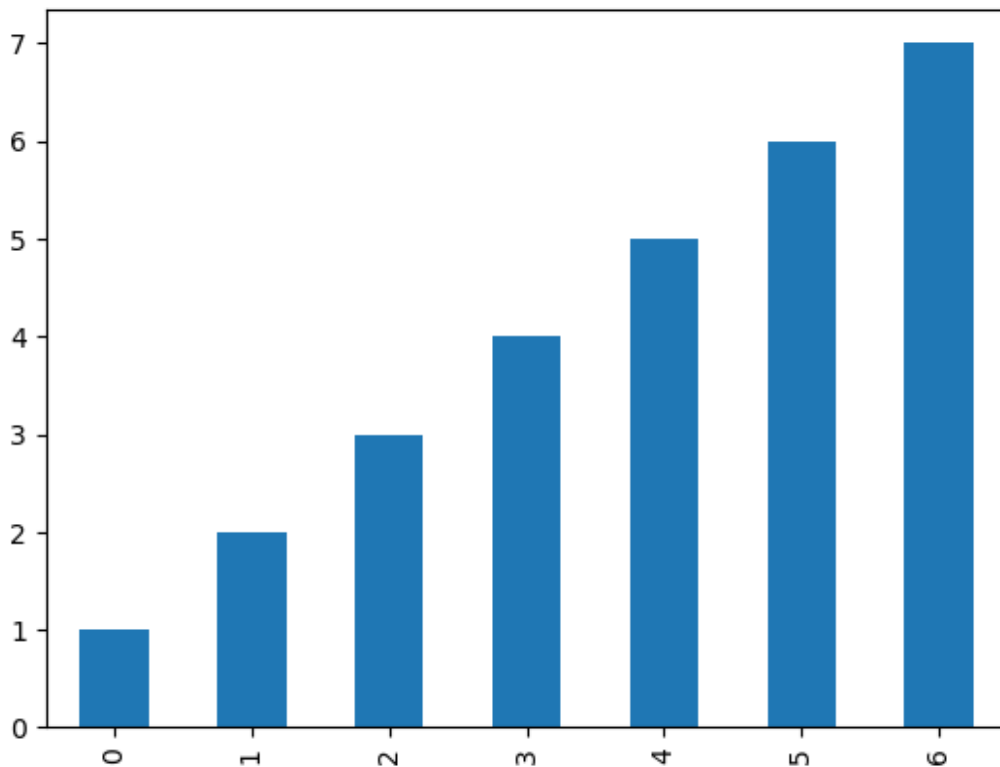
Pandas plot()

- pandas me hi hum plot kar sakte hain (but difference is ki hum utna aache se customize nahi kar payenge graph ko)

```
# on a series
```

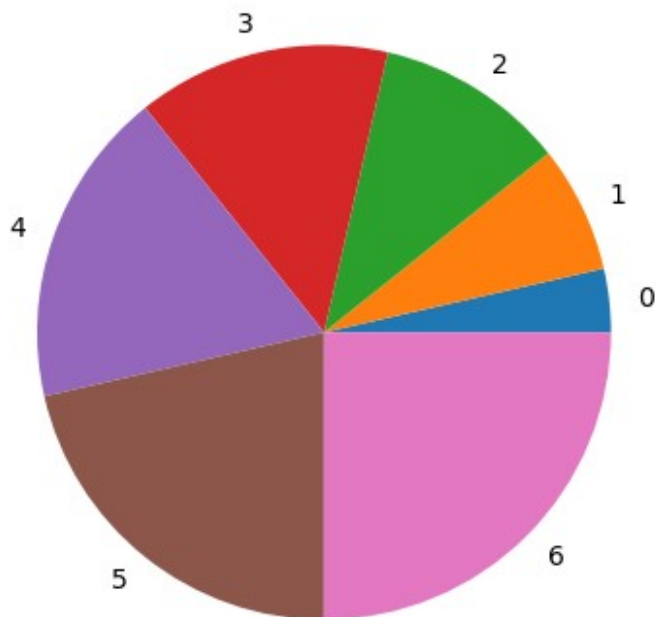
```
s=pd.Series([1,2,3,4,5,6,7])  
s.plot(kind='bar')
```

```
<Axes: >
```



```
s.plot(kind='pie')
```

```
<Axes: >
```



can be used on a dataframe as well

```
import seaborn as sns
tips = sns.load_dataset('tips')
tips
```

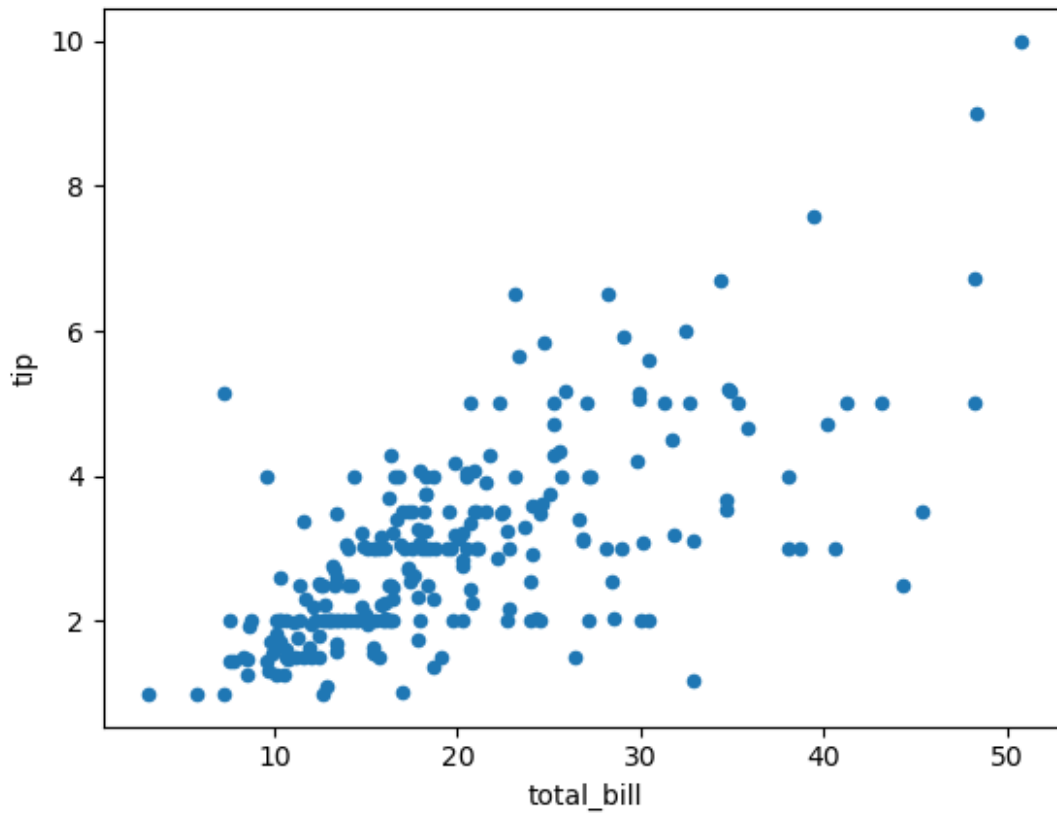
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

[244 rows x 7 columns]

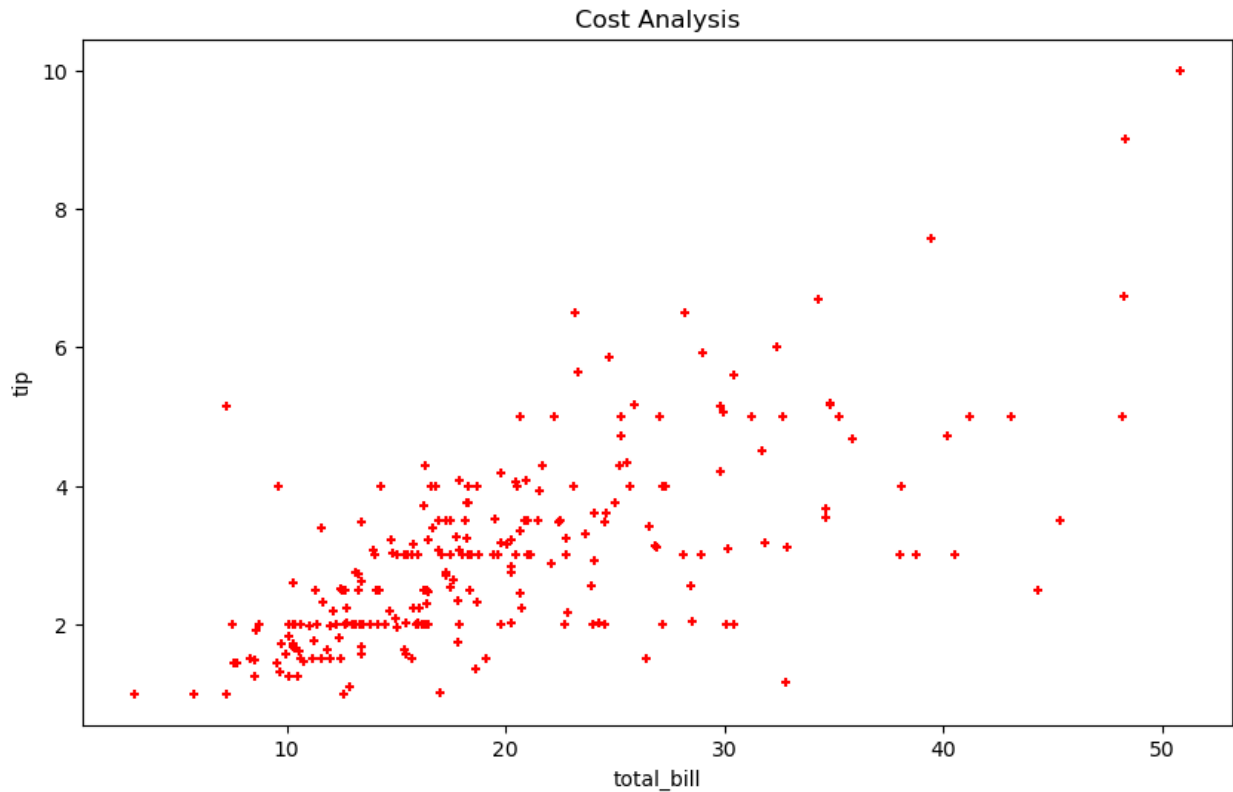
Scatter plot -> labels -> marker -> figsize -> color -> cmap

```
tips.plot(kind='scatter' , x= 'total_bill',y='tip')
```

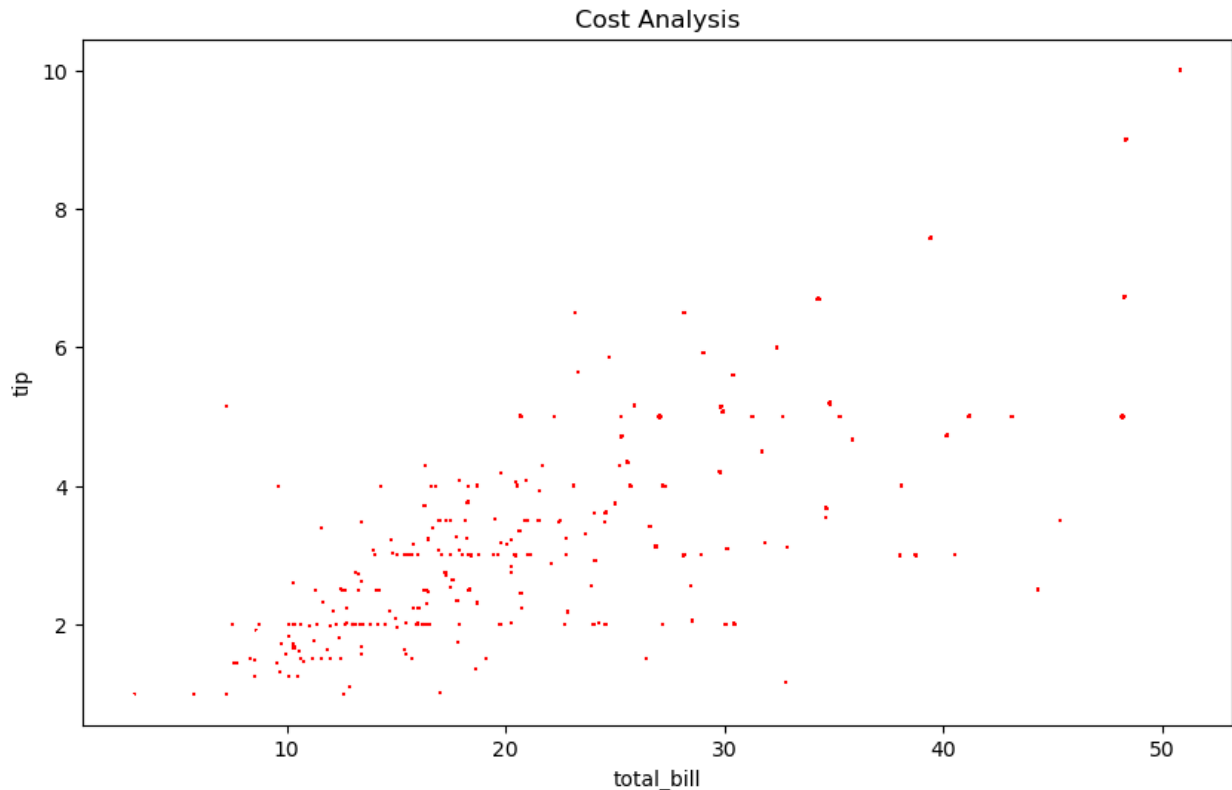
```
<Axes: xlabel='total_bill', ylabel='tip'>
```

```
tips.plot(kind='scatter' , x= 'total_bill',y='tip',title='Cost  
Analysis' ,color='red',marker='+',figsize=(10,6))  
<Axes: title={'center': 'Cost Analysis'}, xlabel='total_bill',  
ylabel='tip'>
```



```
# if size ke according point ko bada or chota dikhana chahate ho to  
# matplotlib me hum kisi data ko multiply kar lete the but isme nahi kar  
# sakte hain  
# hum origiunal data me change karke kar dakte hain  
tips.plot(kind='scatter' , x= 'total_bill',y='tip',title='Cost  
Analysis' ,color='red',marker='+',figsize=(10,6),s='size')  
  
<Axes: title={'center': 'Cost Analysis'}, xlabel='total_bill',  
ylabel='tip'>
```



```
tips.plot(kind='scatter' , x= 'total_bill',y='tip',title='Cost
Analysis' ,color='red',marker='+',figsize=(10,6),s='size'*100) #
multiply not allowed
```

```
-----
-----
ValueError                                Traceback (most recent call
last)
Cell In[281], line 1
----> 1 tips.plot(kind='scatter' , x= 'total_bill',y='tip',title='Cost
Analysis' ,color='red',marker='+',figsize=(10,6),s='size'*100) #
multiply not allowed
```

```
File ~\anaconda3\Lib\site-packages\pandas\plotting\_core.py:975, in
PlotAccessor.__call__(self, *args, **kwargs)
    973 if kind in self._dataframe_kinds:
    974     if isinstance(data, ABCDataFrame):
--> 975         return plot_backend.plot(data, x=x, y=y, kind=kind,
**kwargs)
    976     else:
    977         raise ValueError(f"plot kind {kind} can only be used
for data frames")
```

```
File ~\anaconda3\Lib\site-packages\pandas\plotting\_matplotlib\
__init__.py:71, in plot(data, kind, **kwargs)
```

```

69         kwargs["ax"] = getattr(ax, "left_ax", ax)
70 plot_obj = PLOT_CLASSES[kind](data, **kwargs)
--> 71 plot_obj.generate()
72 plot_obj.draw()
73 return plot_obj.result

```

```

File ~\anaconda3\Lib\site-packages\pandas\plotting\_matplotlib\
core.py:501, in MPLPlot.generate(self)
499 self._compute_plot_data()
500 fig = self.fig
--> 501 self._make_plot(fig)
502 self._add_table()
503 self._make_legend()

```

```

File ~\anaconda3\Lib\site-packages\pandas\plotting\_matplotlib\
core.py:1345, in ScatterPlot._make_plot(self, fig)
1343 else:
1344     label = None
-> 1345 scatter = ax.scatter(
1346     data[x].values,
1347     data[y].values,
1348     c=c_values,
1349     label=label,
1350     cmap=cmap,
1351     norm=norm,
1352     s=self.s,
1353     **self.kwds,
1354 )
1355 if cb:
1356     cbar_label = c if c_is_column else ""

```

```

File ~\anaconda3\Lib\site-packages\matplotlib\__init__.py:1473, in
_preprocess_data.<locals>.inner(ax, data, *args, **kwargs)
1470 @functools.wraps(func)
1471 def inner(ax, *args, data=None, **kwargs):
1472     if data is None:
-> 1473         return func(
1474             ax,
1475             *map(sanitize_sequence, args),
1476             **{k: sanitize_sequence(v) for k, v in
kwargs.items()})
1478     bound = new_sig.bind(ax, *args, **kwargs)
1479     auto_label = (bound.arguments.get(label_namer)
1480                  or bound.kwargs.get(label_namer))

```

```

File ~\anaconda3\Lib\site-packages\matplotlib\axes\_axes.py:4796, in
Axes.scatter(self, x, y, s, c, marker, cmap, norm, vmin, vmax, alpha,
linewidths, edgecolors, plotnonfinite, **kwargs)
4792 s = np.ma.ravel(s)
4793 if (len(s) not in (1, x.size) or

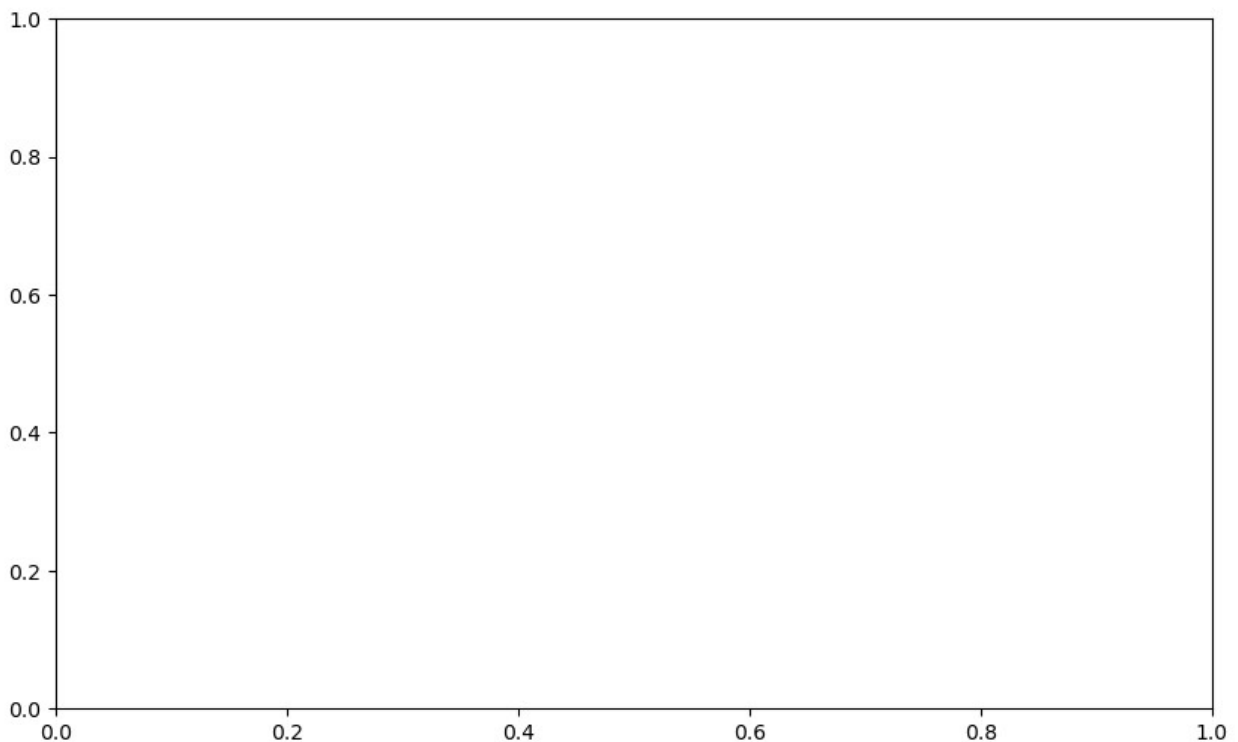
```

```

4794         (not np.issubdtype(s.dtype, np.floating) and
4795         not np.issubdtype(s.dtype, np.integer))):
-> 4796     raise ValueError(
4797         "s must be a scalar, "
4798         "or float array-like with the same size as x and y")
4800 # get the original edgecolor the user passed before we
normalize
4801 orig_edgecolor = edgecolors

```

ValueError: s must be a scalar, or float array-like with the same size as x and y



```
# lets make changes in ooriginal data
```

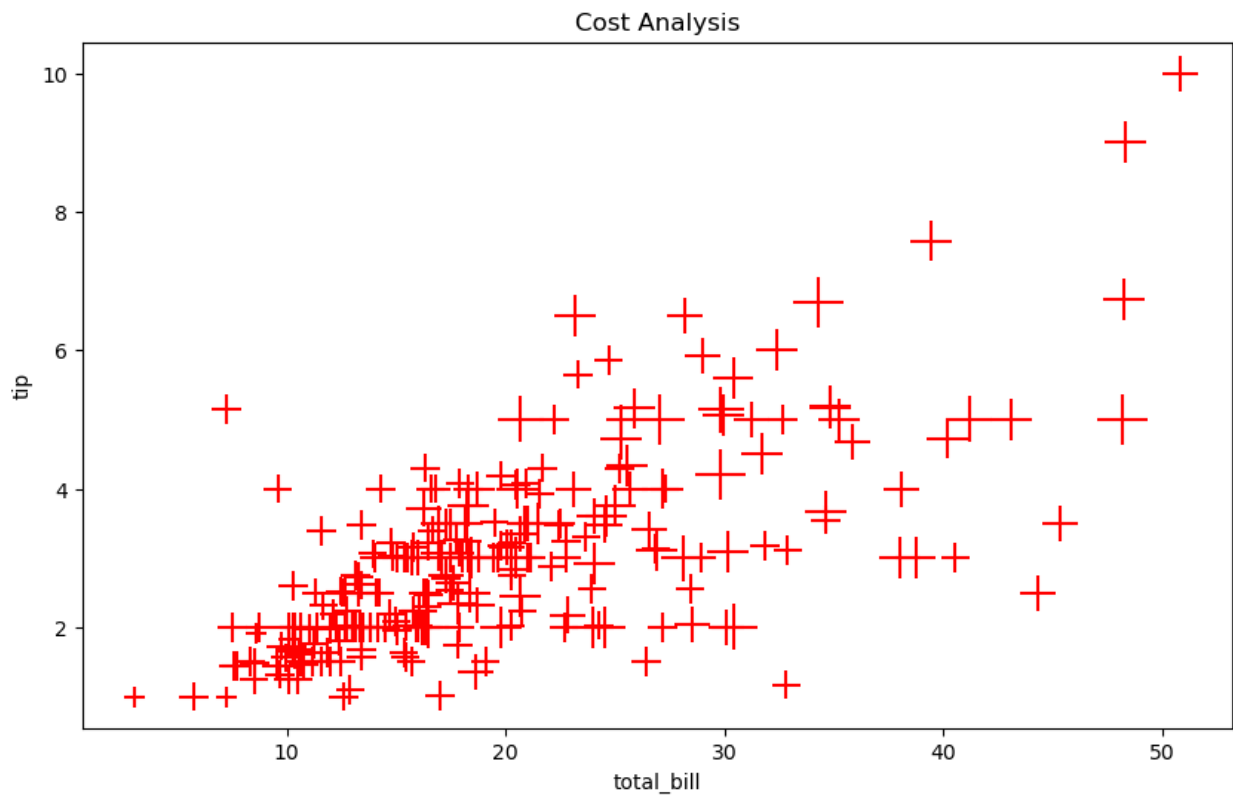
```
tips['size']=tips['size']*100
```

```
tips.head()
```

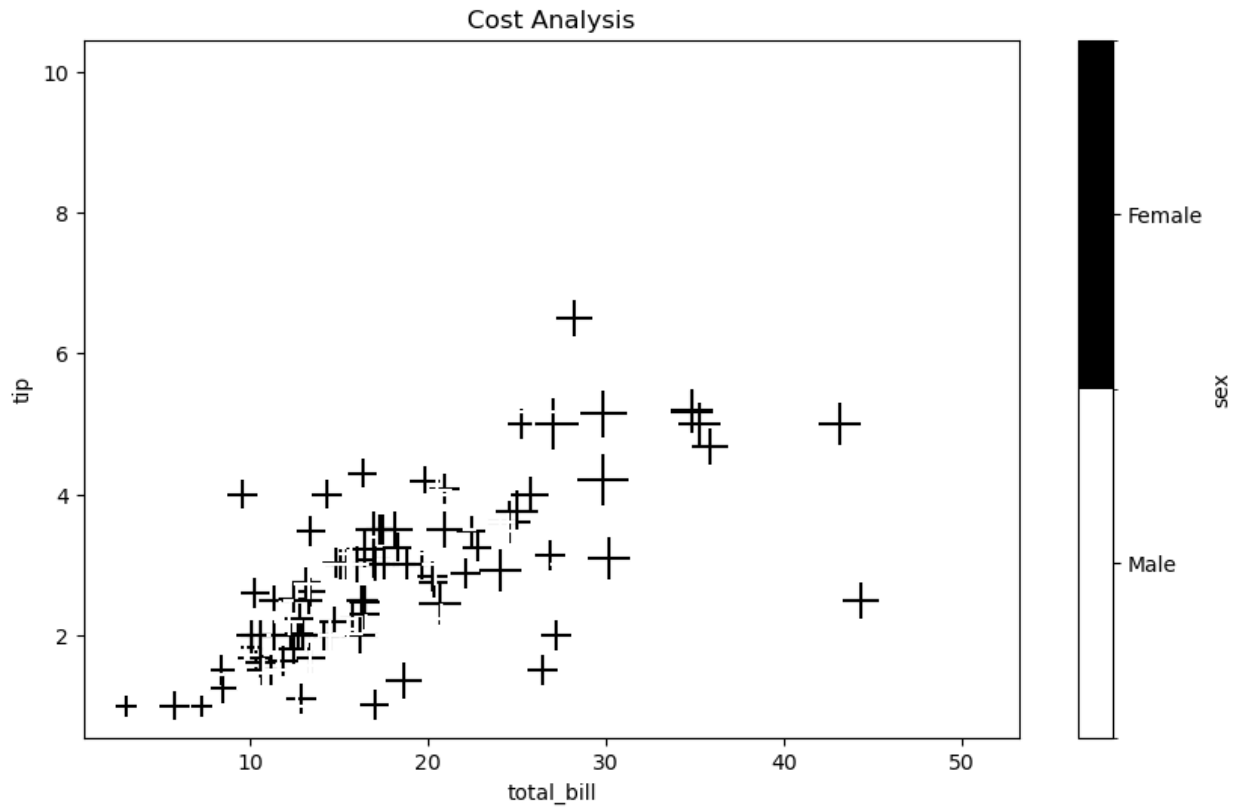
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	200
1	10.34	1.66	Male	No	Sun	Dinner	300
2	21.01	3.50	Male	No	Sun	Dinner	300
3	23.68	3.31	Male	No	Sun	Dinner	200
4	24.59	3.61	Female	No	Sun	Dinner	400

```
tips.plot(kind='scatter' , x= 'total_bill',y='tip',title='Cost
Analysis' ,color='red',marker='+',figsize=(10,6),s='size')
```

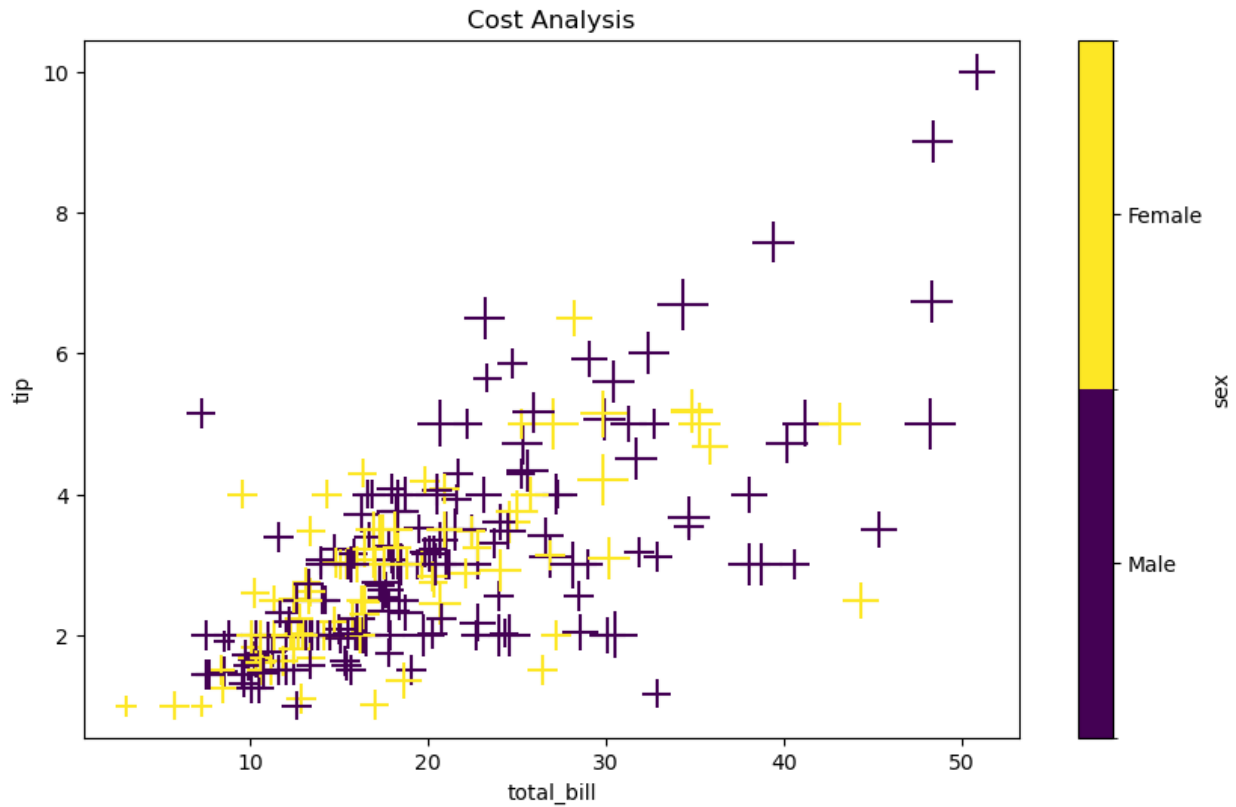
```
<Axes: title={'center': 'Cost Analysis'}, xlabel='total_bill',  
ylabel='tip'>
```



```
# suppose sex ke according hum color change jkarna chahate hain =>  
color parameter ko hata ke c parameter me jiske  
# basis pe change karna chahate ho pass kar do  
tips.plot(kind='scatter' , x= 'total_bill',y='tip',title='Cost  
Analysis' ,marker='+',figsize=(10,6),s='size',c='sex')  
  
<Axes: title={'center': 'Cost Analysis'}, xlabel='total_bill',  
ylabel='tip'>
```



```
# ypu can chage color --> using 'cmap'  
tips.plot(kind='scatter' , x= 'total_bill',y='tip',title='Cost  
Analysis' ,marker='+',figsize=(10,6),s='size',c='sex',cmap='viridis')  
  
<Axes: title={'center': 'Cost Analysis'}, xlabel='total_bill',  
ylabel='tip'>
```



2D plot

dataset =

https://raw.githubusercontent.com/m-mehdi/pandas_tutorials/main/weekly_stocks.csv

```
stocks =
pd.read_csv('https://raw.githubusercontent.com/m-mehdi/pandas_tutorials/main/weekly_stocks.csv')
stocks.head()
```

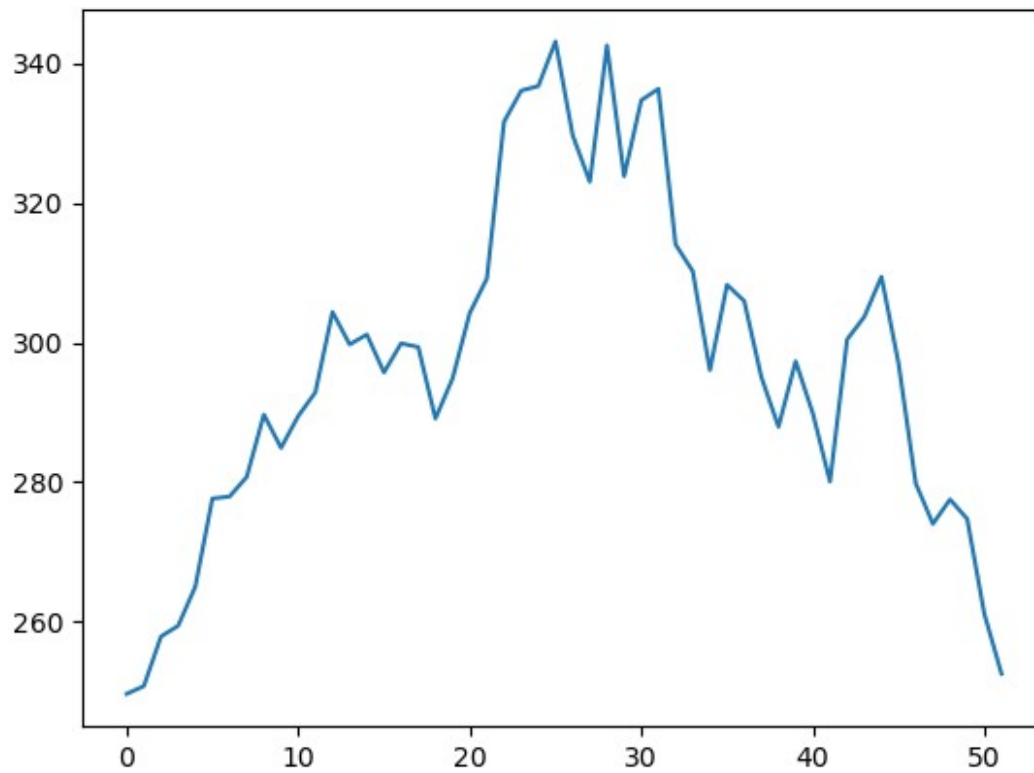
	Date	MSFT	FB	AAPL
0	2021-05-24	249.679993	328.730011	124.610001
1	2021-05-31	250.789993	330.350006	125.889999
2	2021-06-07	257.890015	331.260010	127.349998
3	2021-06-14	259.429993	329.660004	130.460007
4	2021-06-21	265.019989	341.369995	133.110001

line plot

for single columns

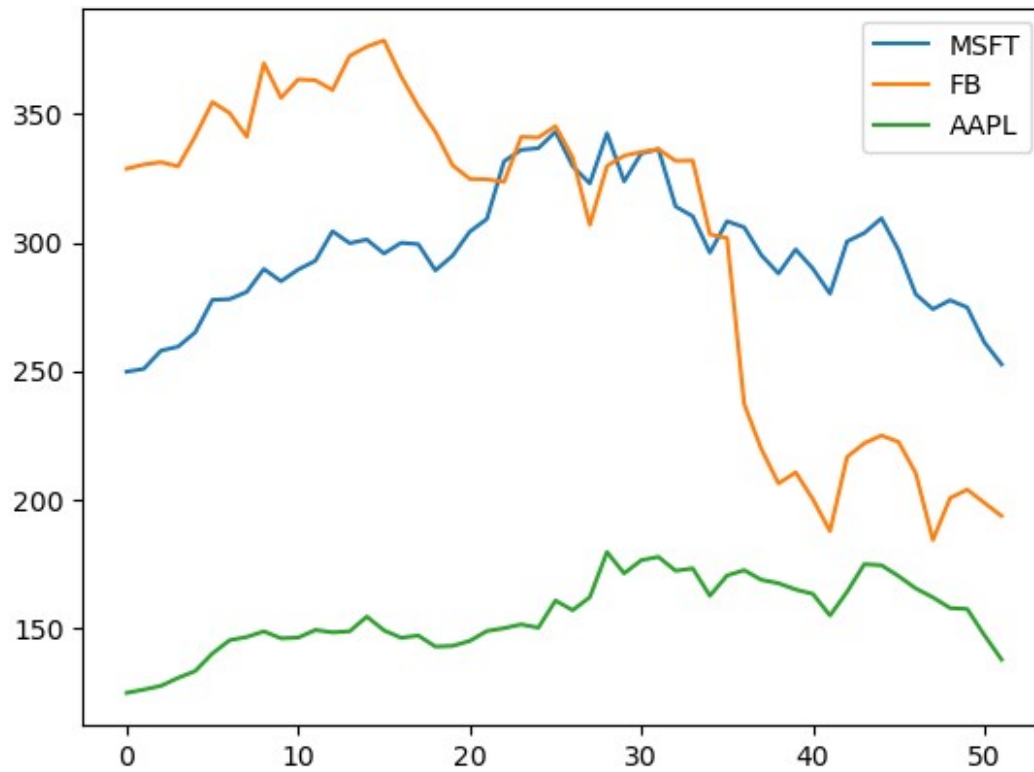
```
stocks['MSFT'].plot(kind='line')
```


<Axes: >

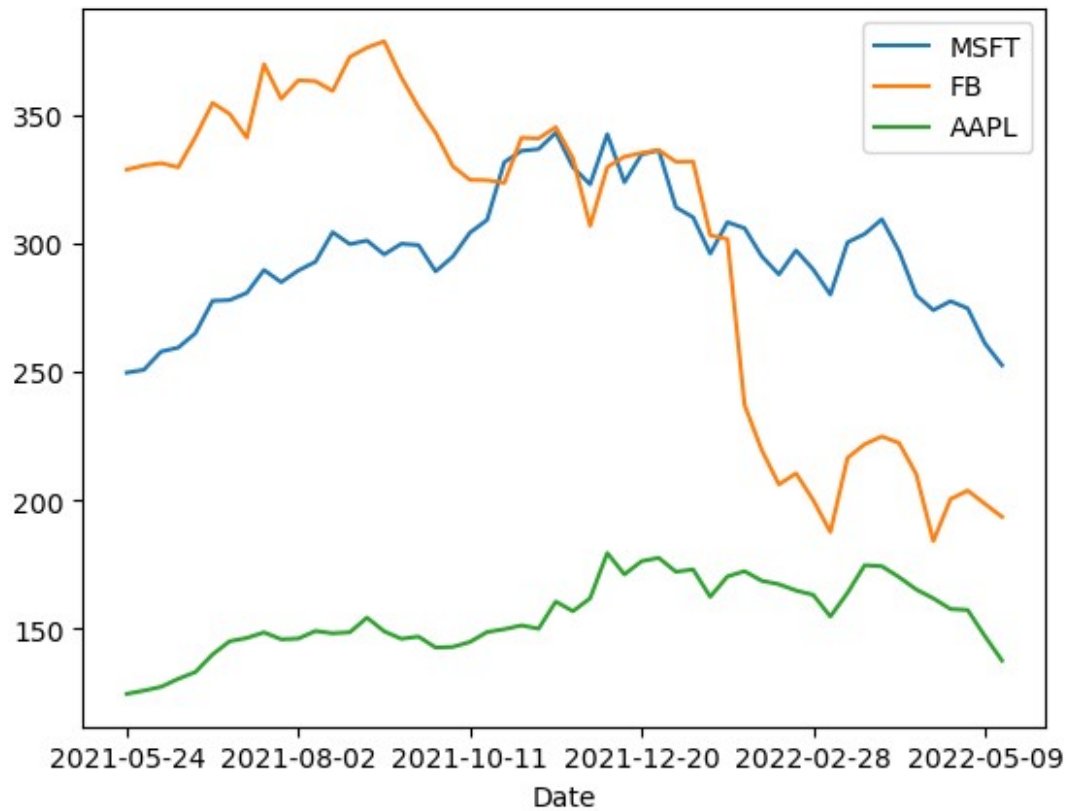


```
# for all the columns  
stocks.plot(kind='line')
```

<Axes: >



```
# x-axis pe date ko laa sakte hain
stocks.plot(kind='line',x='Date')
<Axes: xlabel='Date'>
```



```
# suppose kisi do ko draw karna hai
stocks[['Date', 'AAPL', 'FB']].plot(kind='line', x='Date')
<Axes: xlabel='Date'>
```



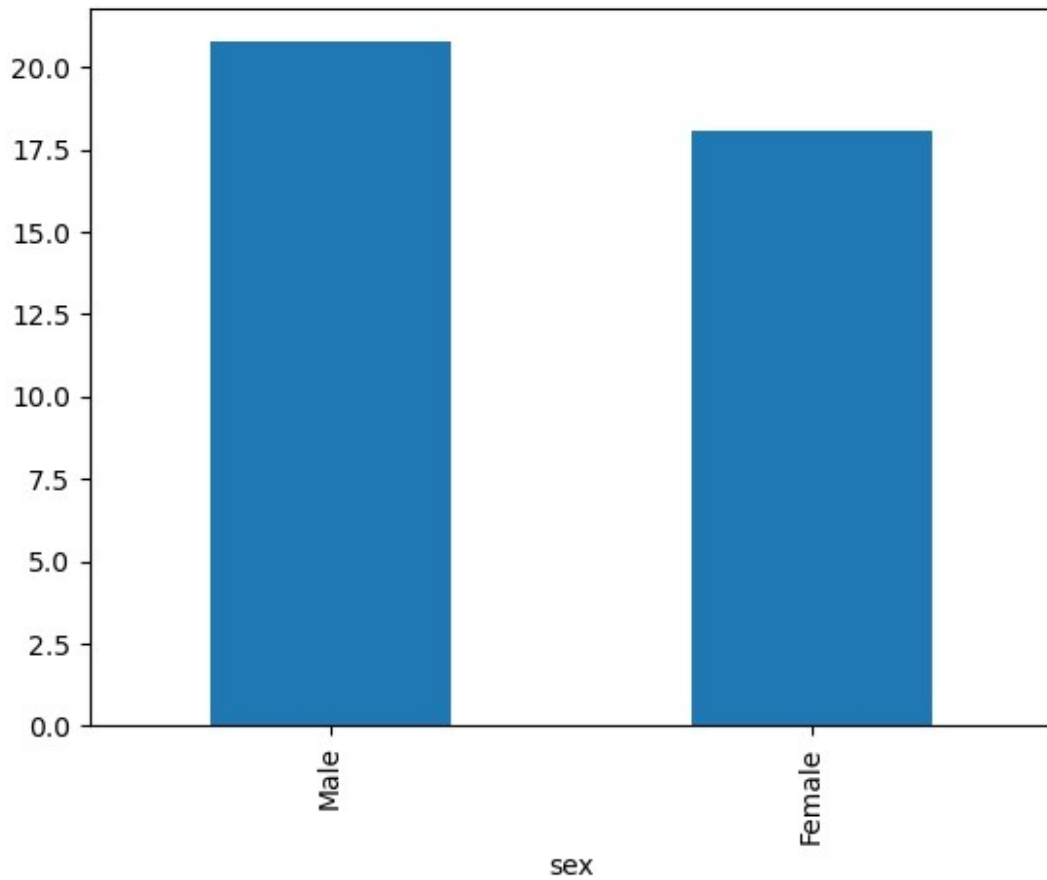
```
# bar chart -> single -> horizontal -> multiple
# using tips
```

```
tips.groupby('sex')['total_bill'].mean().plot(kind='bar')
```

C:\Users\jayra\AppData\Local\Temp\ipykernel_20392\273118215.py:3:
FutureWarning: The default of observed=False is deprecated and will be
changed to True in a future version of pandas. Pass observed=False to
retain current behavior or observed=True to adopt the future default
and silence this warning.

```
tips.groupby('sex')['total_bill'].mean().plot(kind='bar')
```

```
<Axes: xlabel='sex'>
```

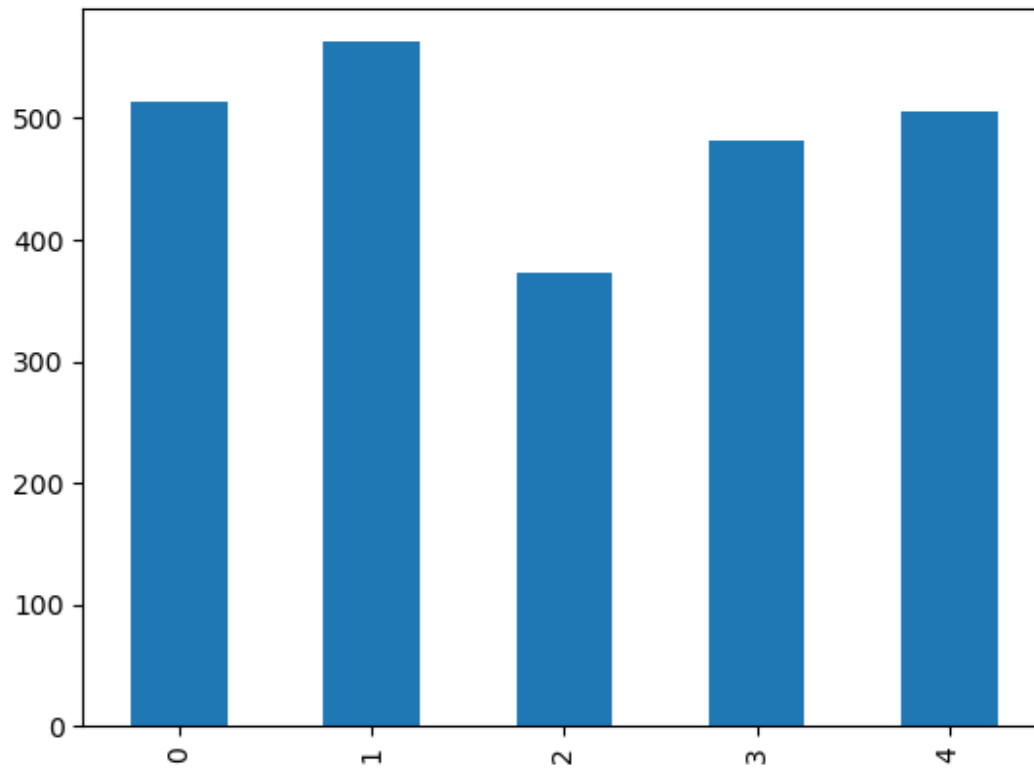


```
temp = pd.read_csv('batsman_season_record.csv')
temp
```

	batsman	2015	2016	2017
0	AB de Villiers	513	687	216
1	DA Warner	562	848	641
2	MS Dhoni	372	284	290
3	RG Sharma	482	489	333
4	V Kohli	505	973	308

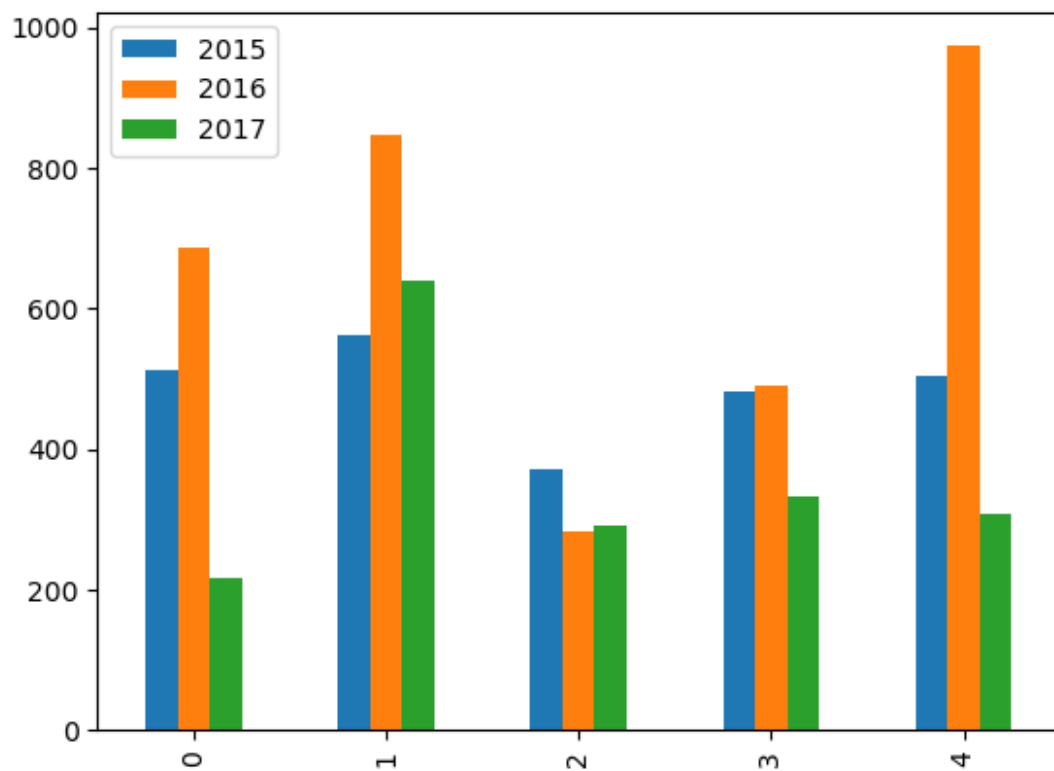
```
# kisi particular ke lye
temp['2015'].plot(kind='bar')
```

```
<Axes: >
```

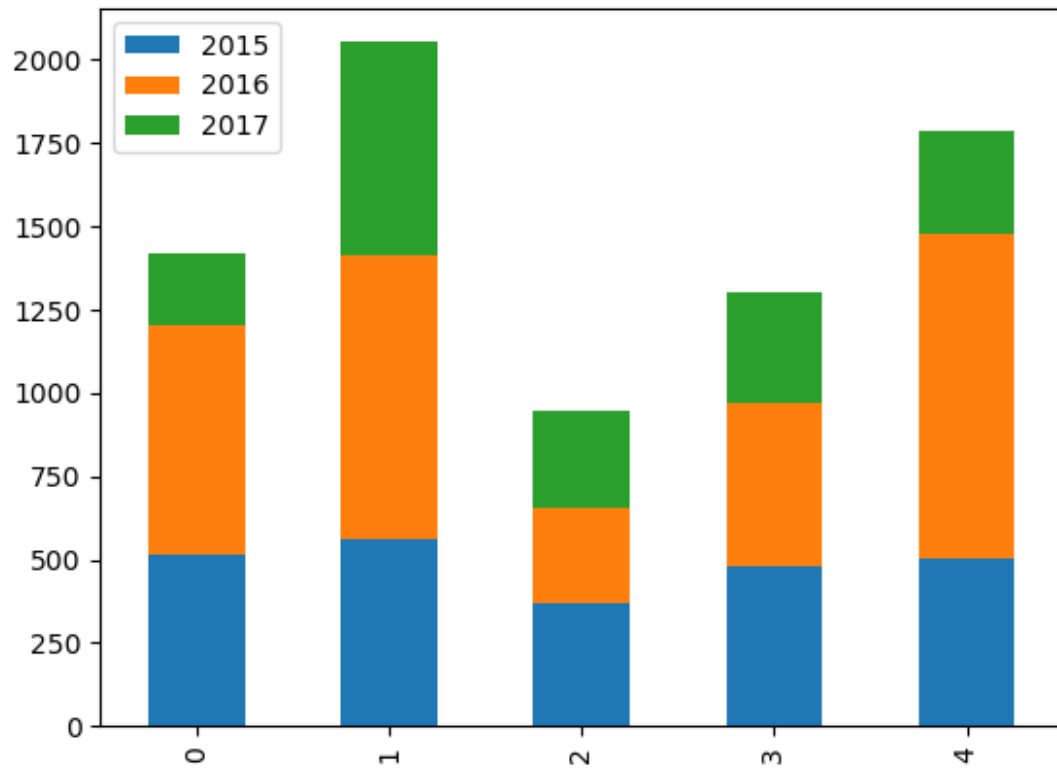


```
temp.plot(kind='bar') # yaad karo last class me aaisa karne me bahut mehnat karna pada tha but yaha pe hum easily kar sakte hain
```

<Axes: >

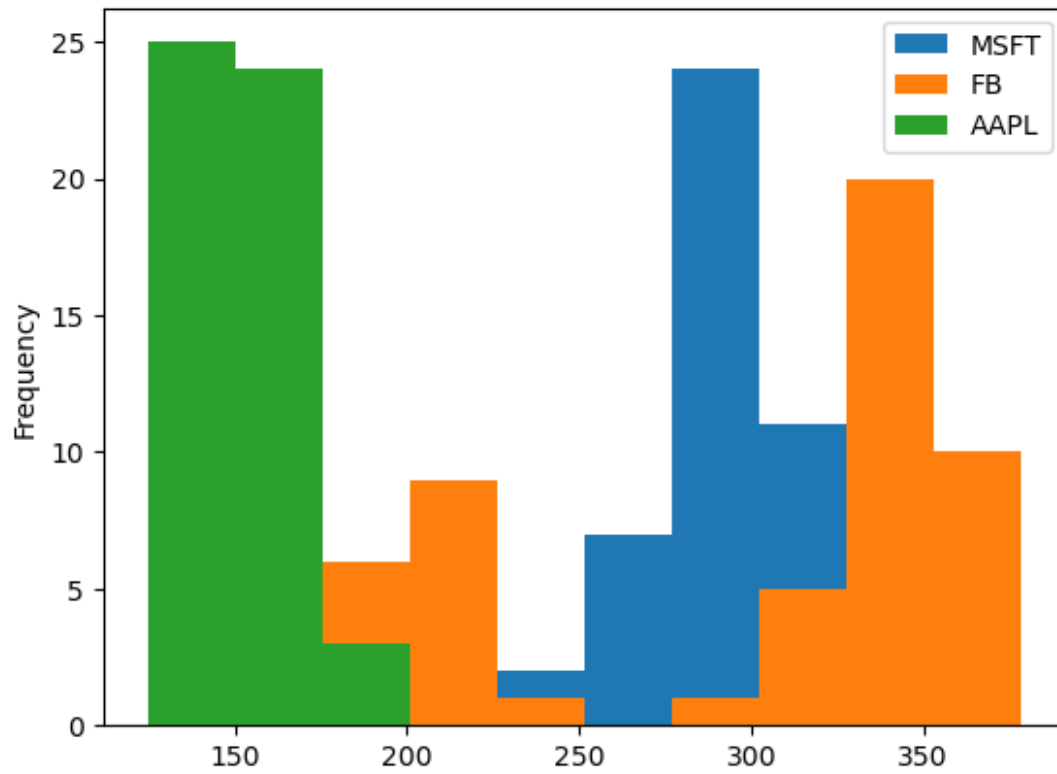


```
# stacked bar chart
temp.plot(kind='bar', stacked=True)
<Axes: >
```

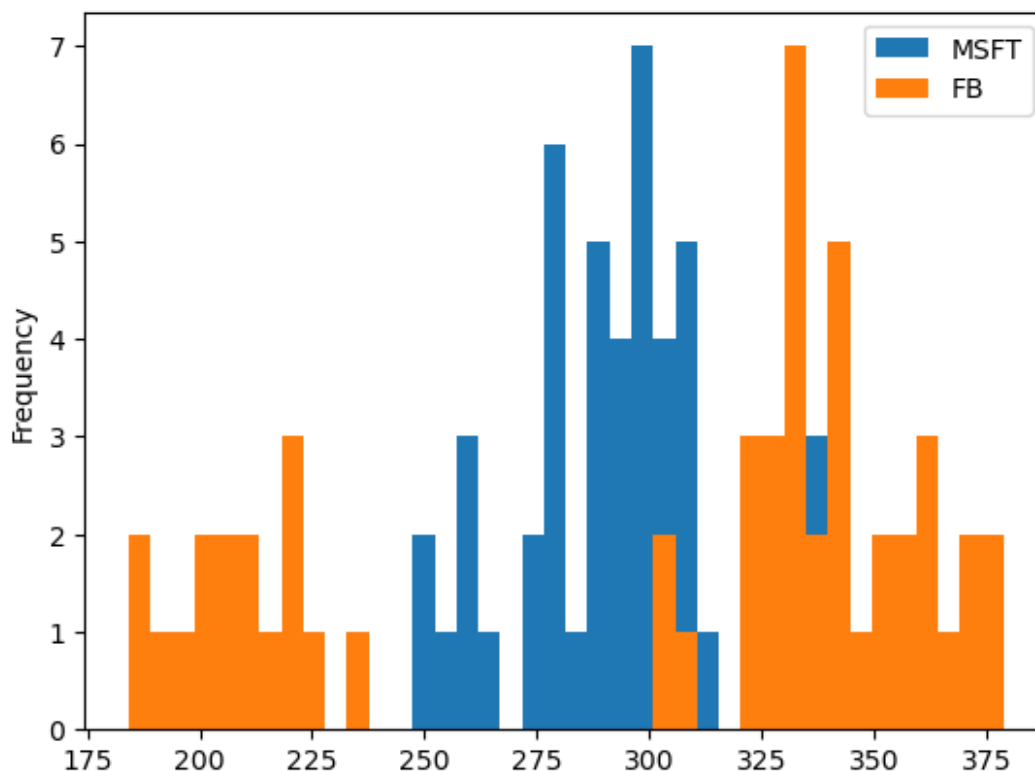


```
# histogram
# using stocks

stocks.plot(kind='hist')
<Axes: ylabel='Frequency'>
```

```
# kisi do ka hi graph banana ho to  
stocks[['MSFT','FB']].plot(kind='hist',bins=40)  
<Axes: ylabel='Frequency'>
```



pie -> single and multiple

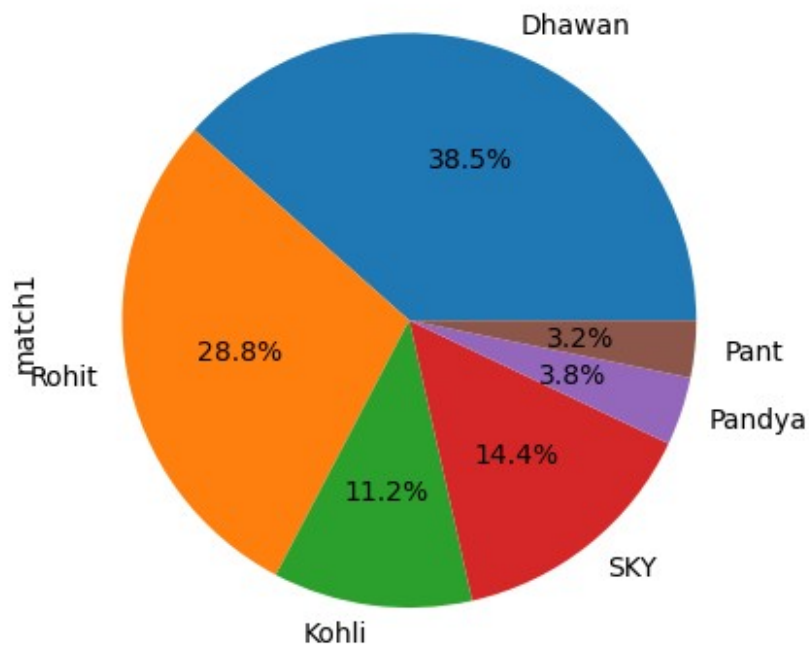
```
df = pd.DataFrame(
    {
        'batsman': ['Dhawan', 'Rohit', 'Kohli', 'SKY', 'Pandya', 'Pant'],
        'match1': [120, 90, 35, 45, 12, 10],
        'match2': [0, 1, 123, 130, 34, 45],
        'match3': [50, 24, 145, 45, 10, 90]
    }
)
```

```
df.head()
```

	batsman	match1	match2	match3
0	Dhawan	120	0	50
1	Rohit	90	1	24
2	Kohli	35	123	145
3	SKY	45	130	45
4	Pandya	12	34	10

```
df['match1'].plot(kind='pie', labels=df['batsman'].values, autopct='%0.1f%%')
```

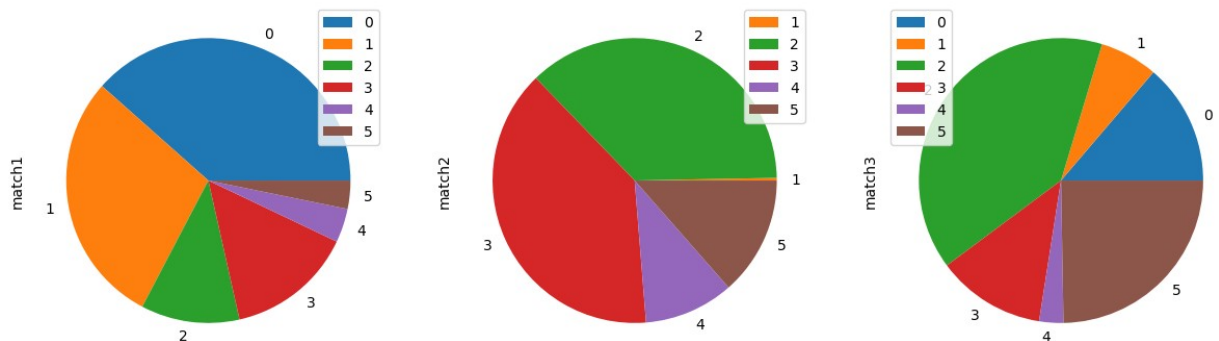
```
<Axes: ylabel='match1'>
```



multiple pie charts

```
df[['match1', 'match2', 'match3']].plot(kind='pie', subplots=True, figsize=(15, 8))

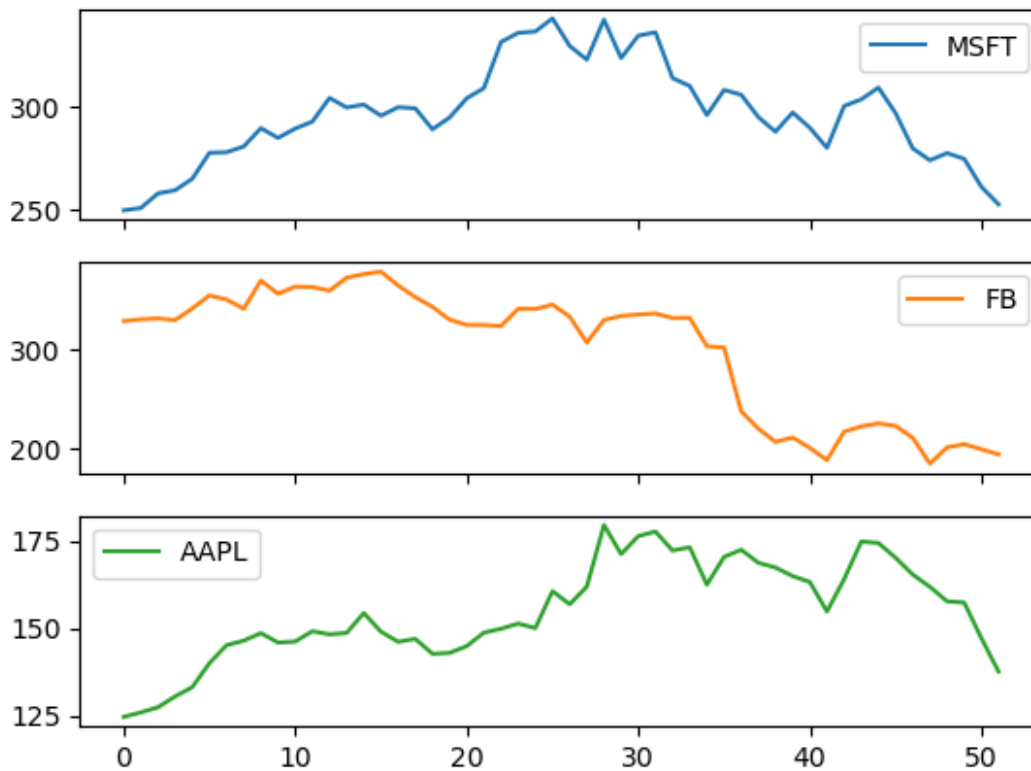
array([<Axes: ylabel='match1'>, <Axes: ylabel='match2'>,
       <Axes: ylabel='match3'>], dtype=object)
```



multiple separate graphs together
using stocks

```
stocks.plot(kind='line', subplots=True)
```

```
array([<Axes: >, <Axes: >, <Axes: >], dtype=object)
```



```
# on multiindex dataframes
# using tips
tips
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	200
1	10.34	1.66	Male	No	Sun	Dinner	300
2	21.01	3.50	Male	No	Sun	Dinner	300
3	23.68	3.31	Male	No	Sun	Dinner	200
4	24.59	3.61	Female	No	Sun	Dinner	400
...
239	29.03	5.92	Male	No	Sat	Dinner	300
240	27.18	2.00	Female	Yes	Sat	Dinner	200
241	22.67	2.00	Male	Yes	Sat	Dinner	200
242	17.82	1.75	Male	No	Sat	Dinner	200
243	18.78	3.00	Female	No	Thur	Dinner	200

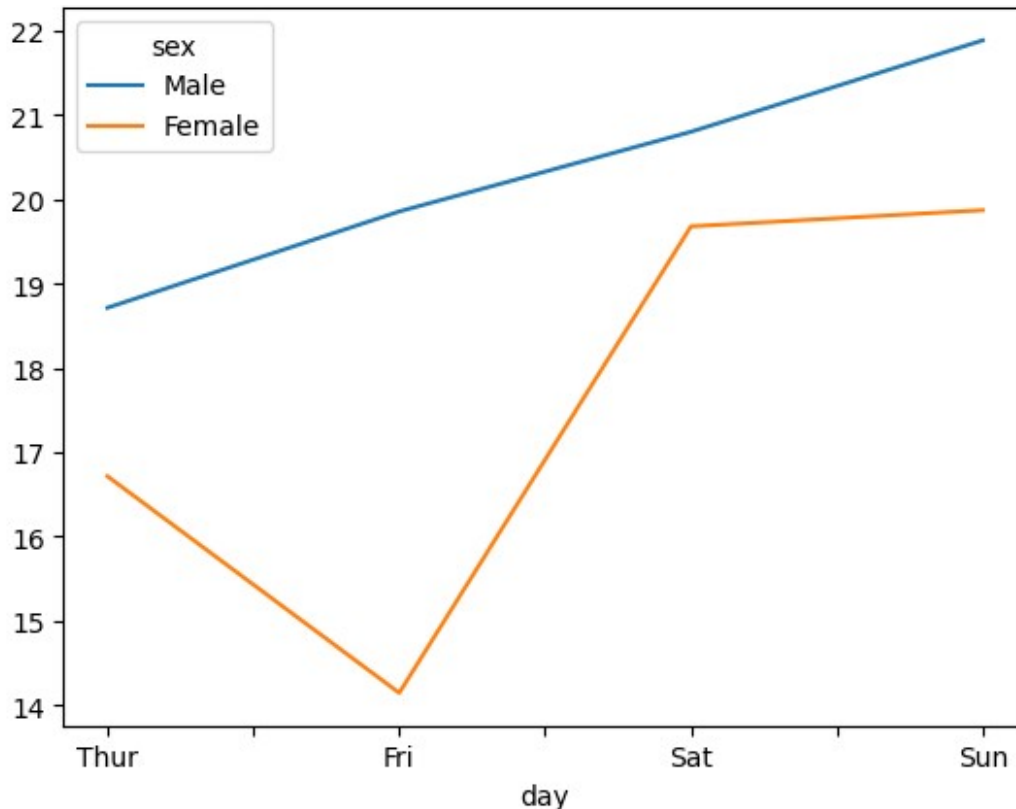
```
[244 rows x 7 columns]
```

```
tips.pivot_table(index='day',columns='sex',values='total_bill',aggfunc='mean').plot(kind='line')
```

```
C:\Users\jayra\AppData\Local\Temp\ipykernel_20392\2011089068.py:1:
FutureWarning: The default value of observed=False is deprecated and
will change to observed=True in a future version of pandas. Specify
observed=False to silence this warning and retain the current behavior
```

```
tips.pivot_table(index='day',columns='sex',values='total_bill',aggfunc=
'mean').plot(kind='line')
```

```
<Axes: xlabel='day'>
```

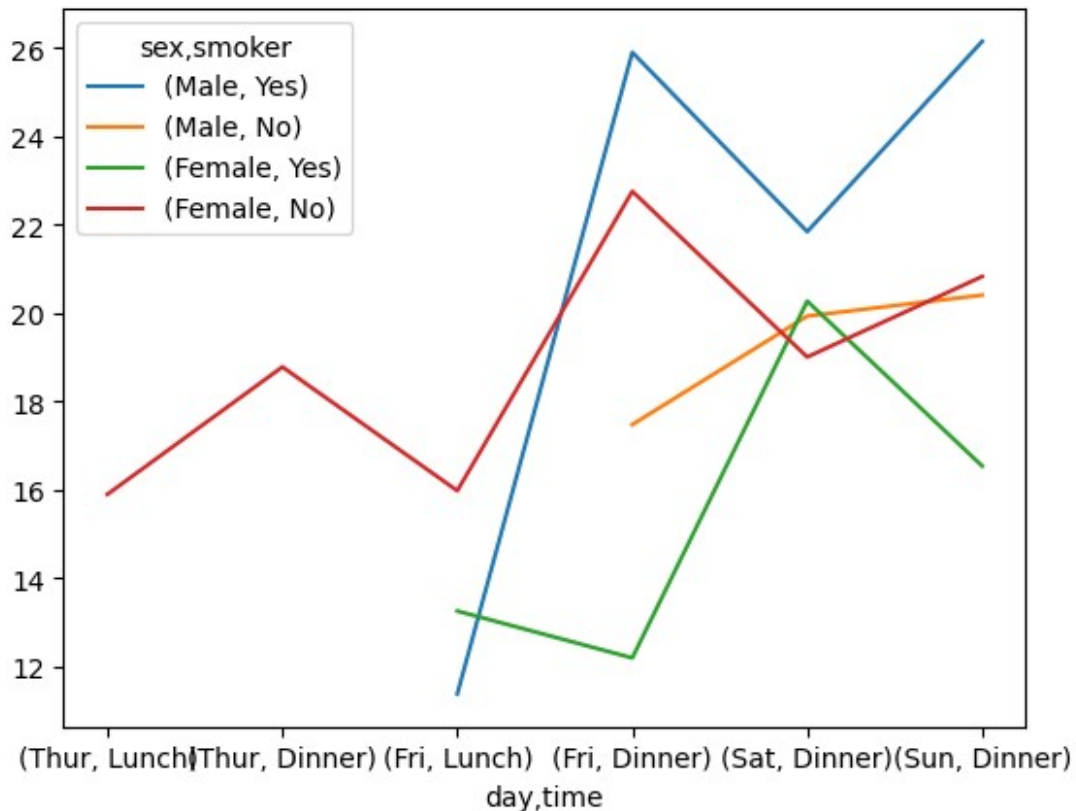


```
tips.pivot_table(index=['day','time'],columns=['sex','smoker'],values=
'total_bill',aggfunc='mean').plot(kind='line')
```

```
C:\Users\jayra\AppData\Local\Temp\ipykernel_20392\1122339873.py:1:
FutureWarning: The default value of observed=False is deprecated and
will change to observed=True in a future version of pandas. Specify
observed=False to silence this warning and retain the current behavior
```

```
tips.pivot_table(index=['day','time'],columns=['sex','smoker'],values=
'total_bill',aggfunc='mean').plot(kind='line')
```

```
<Axes: xlabel='day,time'>
```

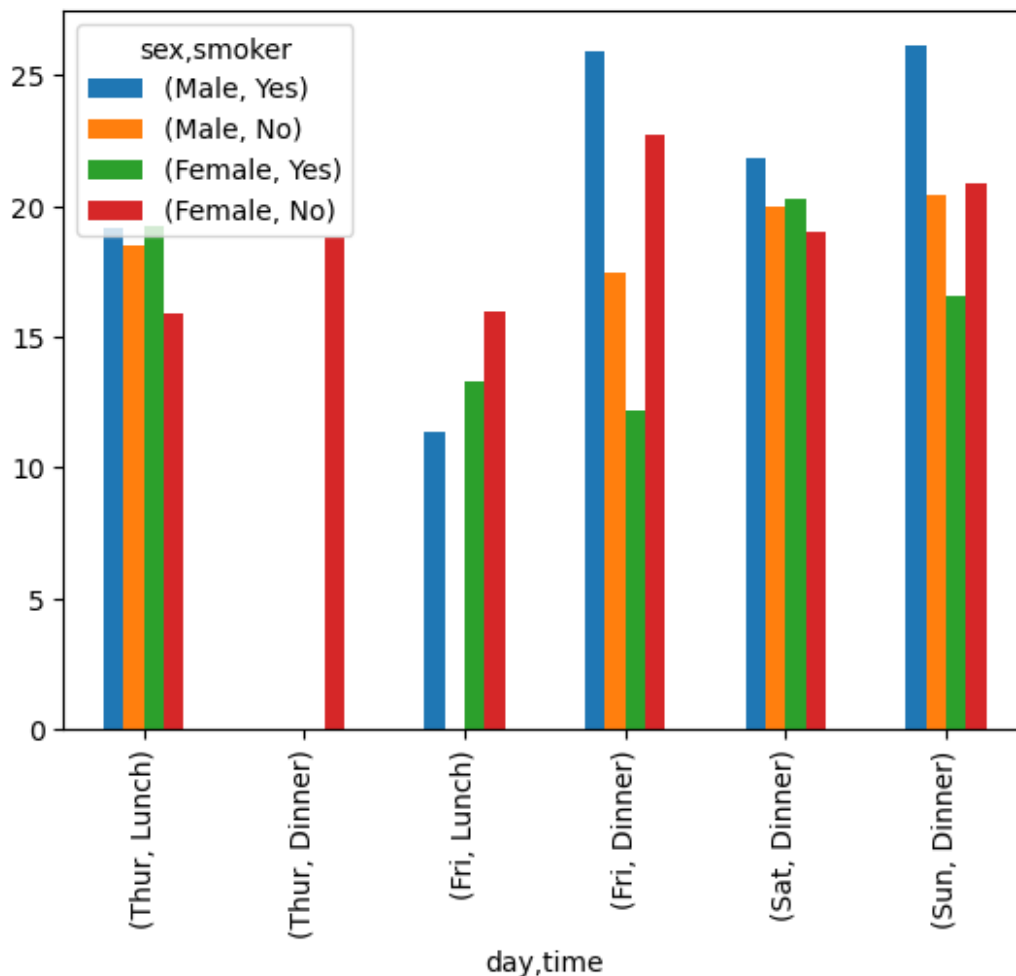


```
tips.pivot_table(index=['day', 'time'], columns=['sex', 'smoker'], values=
'total_bill', aggfunc='mean').plot(kind='bar')
```

C:\Users\jayra\AppData\Local\Temp\ipykernel_20392\3307320567.py:1:
FutureWarning: The default value of observed=False is deprecated and
will change to observed=True in a future version of pandas. Specify
observed=False to silence this warning and retain the current behavior

```
tips.pivot_table(index=['day', 'time'], columns=['sex', 'smoker'], values=
'total_bill', aggfunc='mean').plot(kind='bar')
```

<Axes: xlabel='day,time'>



```
tips.pivot_table(index=['day','time'],columns=['sex','smoker'],values=
'total_bill',aggfunc='mean').plot(kind='pie')
```

C:\Users\jayra\AppData\Local\Temp\ipykernel_20392\3946931005.py:1:
FutureWarning: The default value of observed=False is deprecated and
will change to observed=True in a future version of pandas. Specify
observed=False to silence this warning and retain the current behavior

```
tips.pivot_table(index=['day','time'],columns=['sex','smoker'],values=
'total_bill',aggfunc='mean').plot(kind='pie')
```

```
-----
-----
ValueError                                Traceback (most recent call
last)
Cell In[365], line 1
----> 1
tips.pivot_table(index=['day','time'],columns=['sex','smoker'],values=
'total_bill',aggfunc='mean').plot(kind='pie')
```

```
File ~\anaconda3\Lib\site-packages\pandas\plotting\_core.py:981, in
PlotAccessor.__call__(self, *args, **kwargs)
    979 if isinstance(data, ABCDataFrame):
    980     if y is None and kwargs.get("subplots") is False:
--> 981         raise ValueError(
    982             f"{kind} requires either y column or
'subplots=True'"
    983         )
    984     if y is not None:
    985         if is_integer(y) and not
data.columns._holds_integer():
```

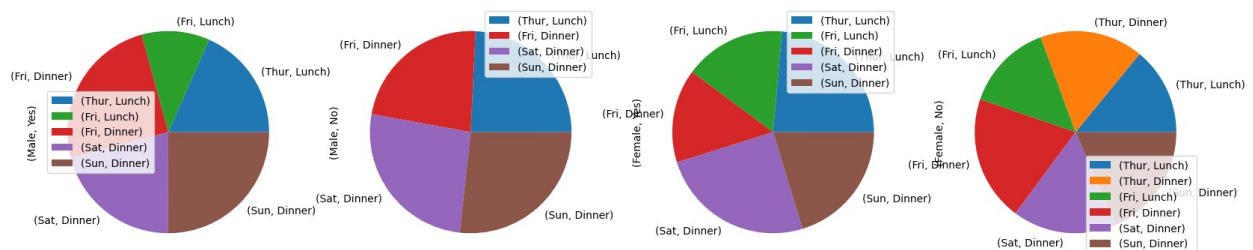
ValueError: pie requires either y column or 'subplots=True'

```
# subplots=True karna hoga multiple pie ke lye
tips.pivot_table(index=['day', 'time'], columns=['sex', 'smoker'], values=
'total_bill', aggfunc='mean').plot(kind='pie', subplots=True, figsize=(20
,10))
```

C:\Users\jayra\AppData\Local\Temp\ipykernel_20392\2408203065.py:2:
FutureWarning: The default value of observed=False is deprecated and
will change to observed=True in a future version of pandas. Specify
observed=False to silence this warning and retain the current behavior

```
tips.pivot_table(index=['day', 'time'], columns=['sex', 'smoker'], values=
'total_bill', aggfunc='mean').plot(kind='pie', subplots=True, figsize=(20
,10))
```

```
array([<Axes: ylabel='(Male, Yes)'>, <Axes: ylabel='(Male, No)'>,
      <Axes: ylabel='(Female, Yes)'>, <Axes: ylabel='(Female, No)'>],
      dtype=object)
```



```
# jab sab kuch pandas me kar sakte hain to matplotlib q --> islye qki
hum sirf series aur dataframe ke sath hi pandas better tarike se kaam
kar sakte hain
# but suppose ki humare pass numpy array hai to hume jyada
customization karna padega pandas me
```