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

Colored Scatterplots

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

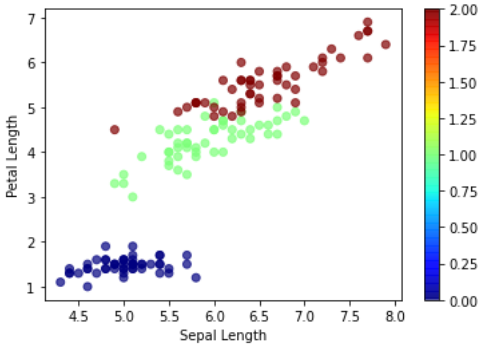
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
9	10	4.9	3.1	1.5	0.1	Iris-setosa
73	74	6.1	2.8	4.7	1.2	Iris-versicolor
44	45	5.1	3.8	1.9	0.4	Iris-setosa
51	52	6.4	3.2	4.5	1.5	Iris-versicolor
104	105	6.5	3.0	5.8	2.2	Iris-virginica

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

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
87	88	6.3	2.3	4.4	1.3	1
20	21	5.4	3.4	1.7	0.2	0
56	57	6.3	3.3	4.7	1.6	1
140	141	6.7	3.1	5.6	2.4	2
141	142	6.9	3.1	5.1	2.3	2

```
plt.scatter(iris['SepalLengthCm'],iris['PetalLengthCm'],c=iris['Species'],cmap='jet',alpha=0.7)
plt.xlabel('Sepal Length')
plt.ylabel('Petal Length')
plt.colorbar()
```

```
<matplotlib.colorbar.Colorbar at 0x7f5e17170bb0>
```

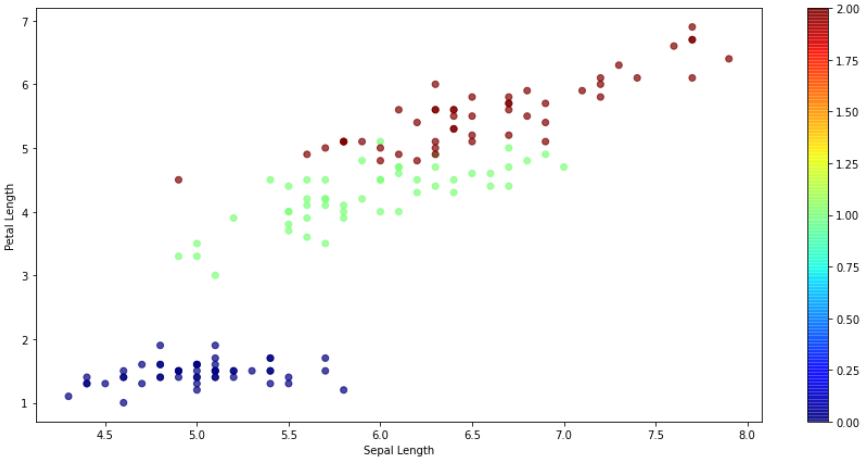



```
# cmap and alpha
```

Plot size

```
plt.figure(figsize=(15,7))

plt.scatter(iris['SepalLengthCm'],iris['PetalLengthCm'],c=iris['Species'],cmap='jet',alpha=0.7)
plt.xlabel('Sepal Length')
plt.ylabel('Petal Length')
plt.colorbar()
```

```
<matplotlib.colorbar.Colorbar at 0x7f5e16ee4430>
```



Annotations

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

```
batters.shape
```

```
(605, 4)
```

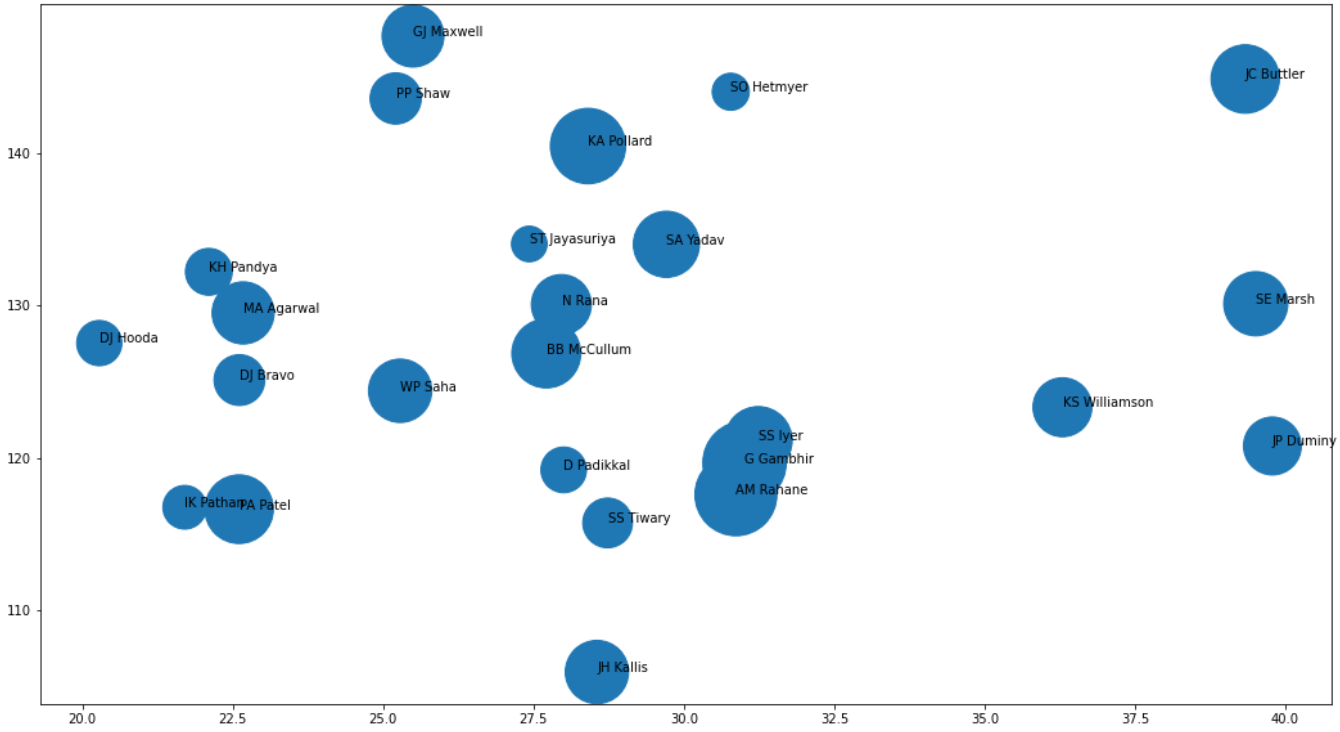
```
sample_df = df.head(100).sample(25,random_state=5)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-137-839dfd0bcf32> in <module>
----> 1 sample_df = df.head(100).sample(25,random_state=5)

/usr/local/lib/python3.8/dist-packages/pandas/core/generic.py in sample(self,
n, frac, replace, weights, random_state, axis, ignore_index)
   5363             )
   5364
-> 5365         locs = rs.choice(axis_length, size=n, replace=replace,
p=weights)
   5366         result = self.take(locs, axis=axis)
   5367         if ignore_index:

mtrand.pyx in numpy.random.mtrand.RandomState.choice()

ValueError: Cannot take a larger sample than population when 'replace=False'

SEARCH STACK OVERFLOW
```

```
sample_df
```

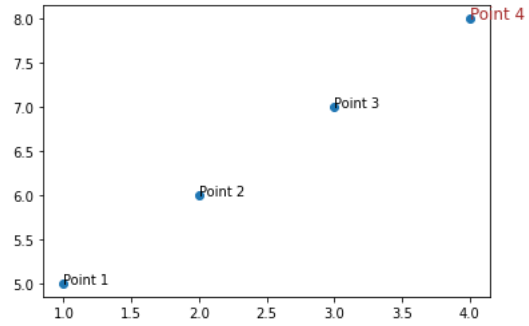
	batter	runs	avg	strike_rate
66	KH Pandya	1326	22.100000	132.203390
32	SE Marsh	2489	39.507937	130.109775
46	JP Duminy	2029	39.784314	120.773810
28	SA Yadav	2644	29.707865	134.009123
74	IK Pathan	1150	21.698113	116.751269
23	JC Buttler	2832	39.333333	144.859335
10	G Gambhir	4217	31.007353	119.665153
20	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

```
plt.figure(figsize=(18,10))
plt.scatter(sample_df['avg'],sample_df['strike_rate'],s=sample_df['runs'])

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])
```



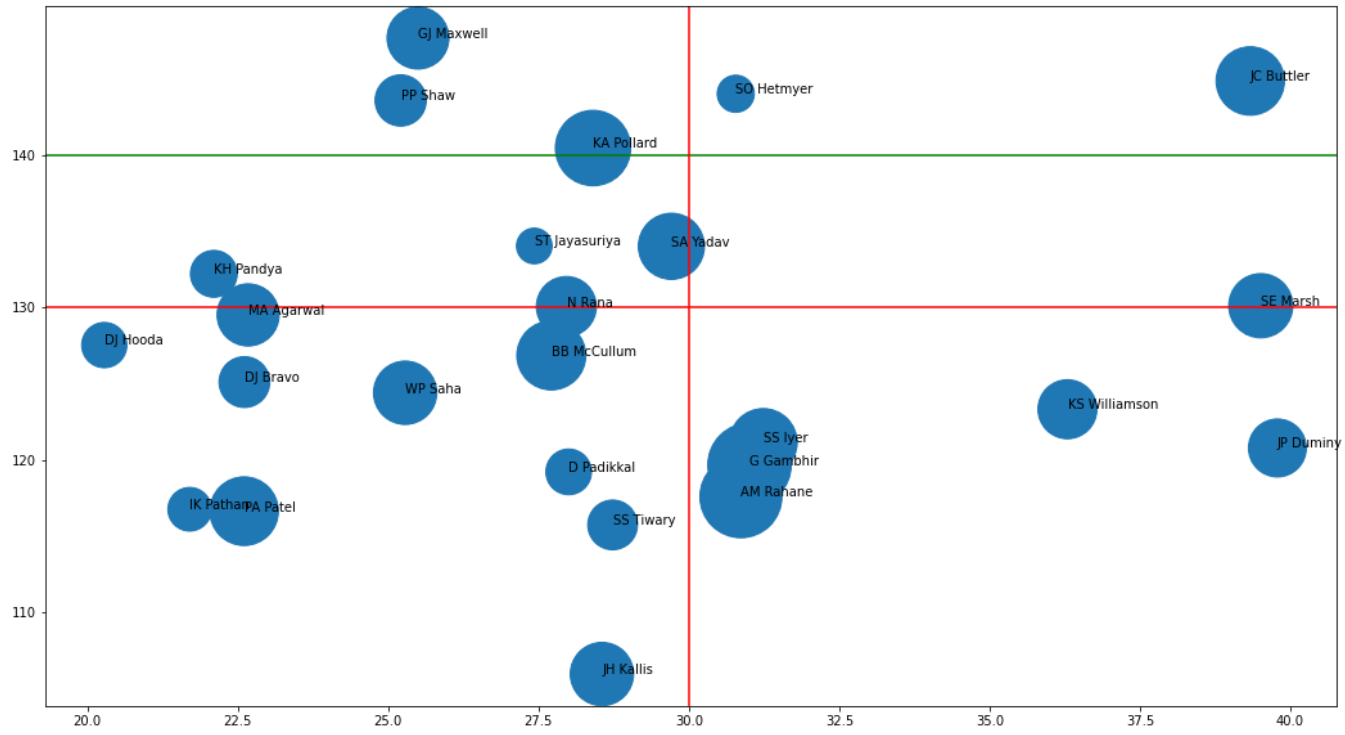
```
x = [1,2,3,4]
y = [5,6,7,8]

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

```
plt.figure(figsize=(18,10))
plt.scatter(sample_df['avg'],sample_df['strike_rate'],s=sample_df['runs'])

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

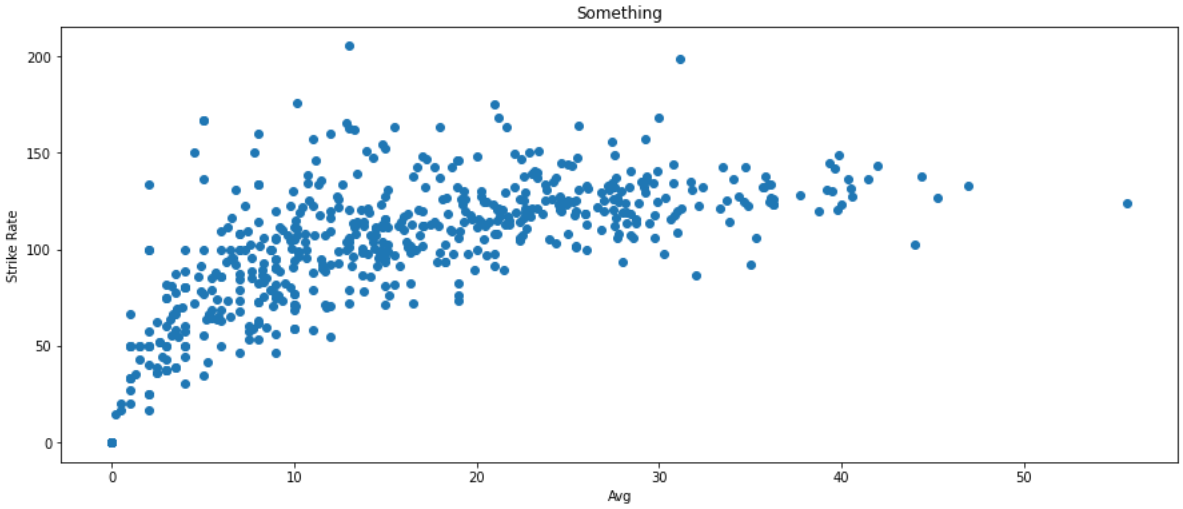


Subplots

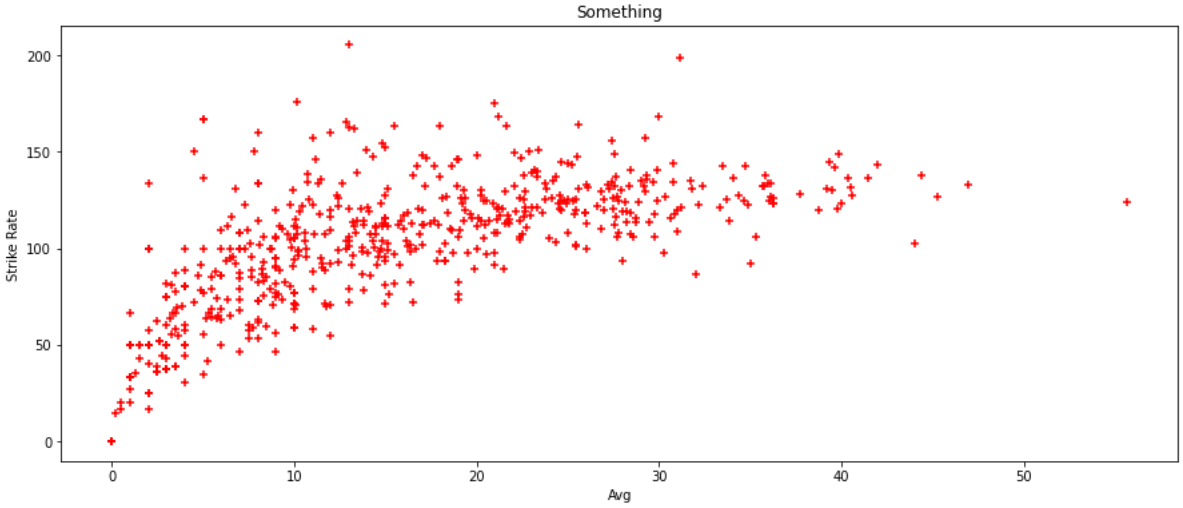
```
# A diff way to plot graphs
batters.head()
```

	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.figure(figsize=(15,6))
plt.scatter(batters['avg'],batters['strike_rate'])
plt.title('Something')
plt.xlabel('Avg')
plt.ylabel('Strike Rate')

plt.show()
```




```
fig,ax = plt.subplots(figsize=(15,6))

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

fig.show()
```



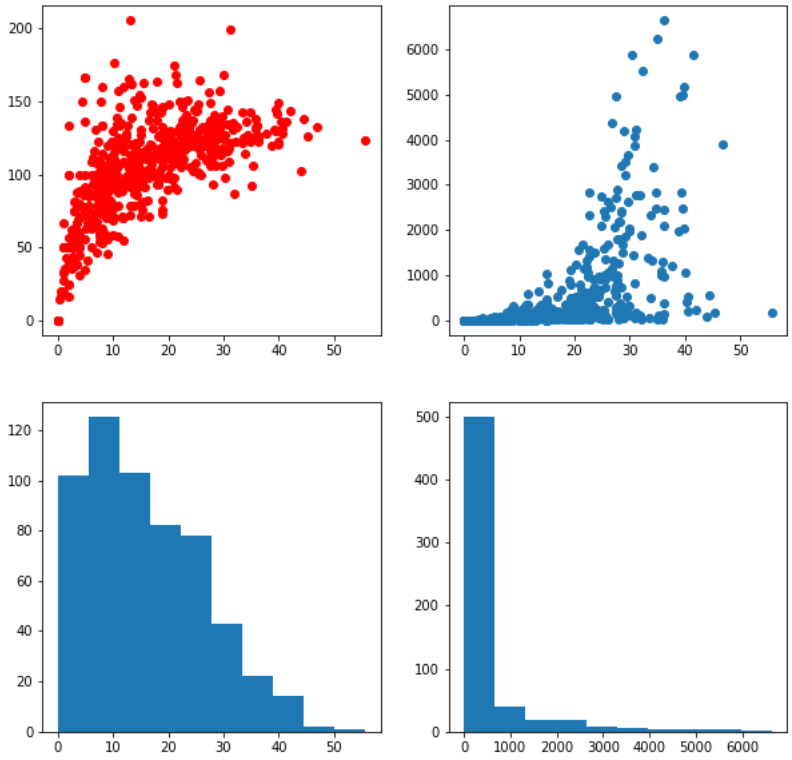

```
# batter dataset
```

```
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[0].set_ylabel('Strike Rate')


ax[1].set_title('Avg Vs Runs')
ax[1].set_ylabel('Runs')
ax[1].set_xlabel('Avg')
```

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

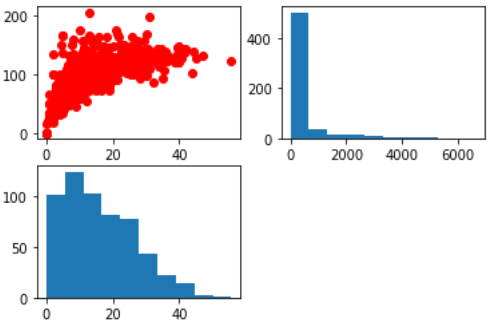

```
fig, ax = plt.subplots(nrows=2,ncols=2,figsize=(10,10))

ax[0,0].
ax[0,1].scatter(batters['avg'],batters['runs'])
ax[1,0].hist(batters['avg'])
ax[1,1].hist(batters['runs'])
```

```
(array([499.,  40.,  19.,  19.,   9.,   6.,   4.,   4.,   3.,   2.]),
 array([   0. ,  663.4, 1326.8, 1990.2, 2653.6, 3317. , 3980.4, 4643.8,
        5307.2, 5970.6, 6634. ]),
 <a list of 10 Patch objects>)
```



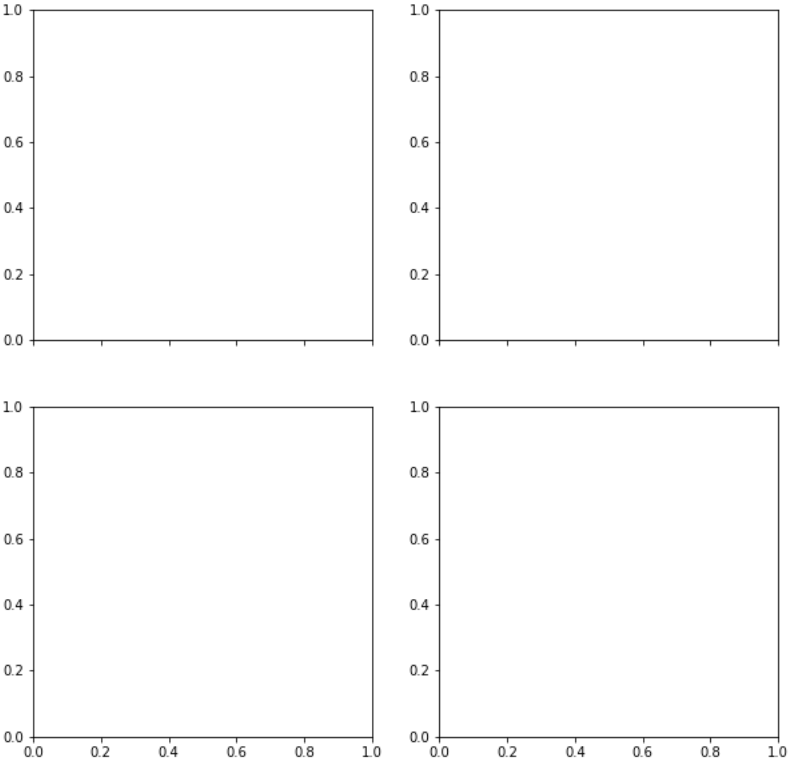
```
fig = plt.figure()

ax1 = fig.add_subplot(2,2,1)
ax1.scatter(batters['avg'],batters['strike_rate'],color='red')

ax2 = fig.add_subplot(2,2,2)
ax2.hist(batters['runs'])

ax3 = fig.add_subplot(2,2,3)
ax3.hist(batters['avg'])
```

```
(array([102., 125., 103.,  82.,  78.,  43.,  22.,  14.,   2.,   1.]),
 array([ 0.        ,  5.56666667, 11.13333333, 16.7       , 22.26666667,
        27.83333333, 33.4       , 38.96666667, 44.53333333, 50.1       ,
        55.66666667]),
 <a list of 10 Patch objects>)
```




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

ax[1,1]
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5e15913c10>
```

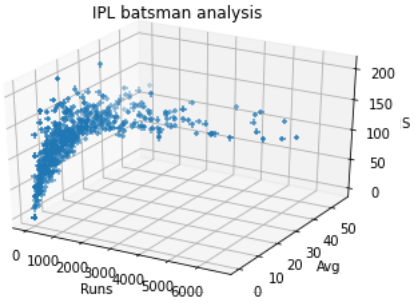


3D Scatter Plots

```
batters
```

```
fig = plt.figure()

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

ax.scatter3D(batters['runs'],batters['avg'],batters['strike_rate'],marker='+')
ax.set_title('IPL batsman analysis')

ax.set_xlabel('Runs')
ax.set_ylabel('Avg')
ax.set_zlabel('SR')
```

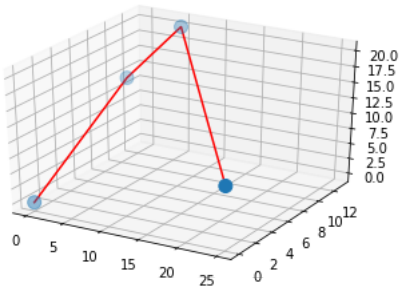
```
Text(0.5, 0, 'SR')
```



3D Line Plot

```
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 0x7f5e14d13f10>]
```



3D Surface Plots

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

```
xx, yy = np.meshgrid(x,y)
```

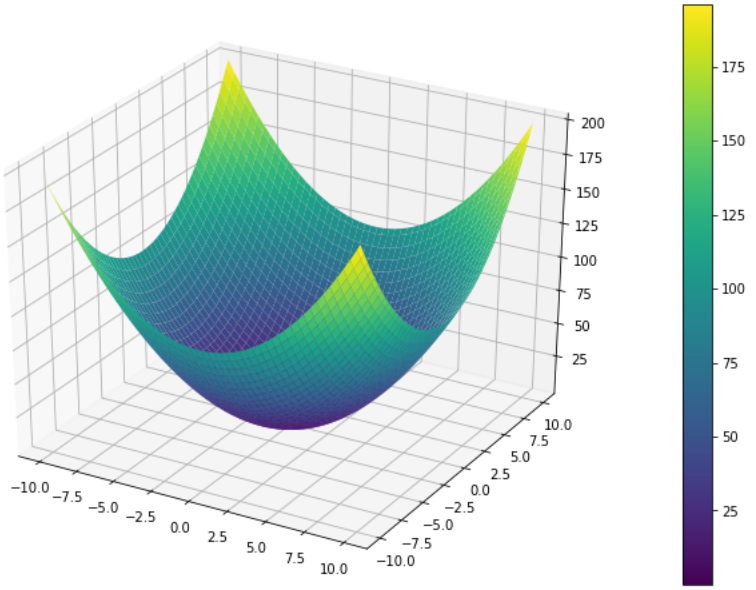
```
(100, 100)
```

```
z = xx**2 + yy**2
z.shape
```

```
(100, 100)
```

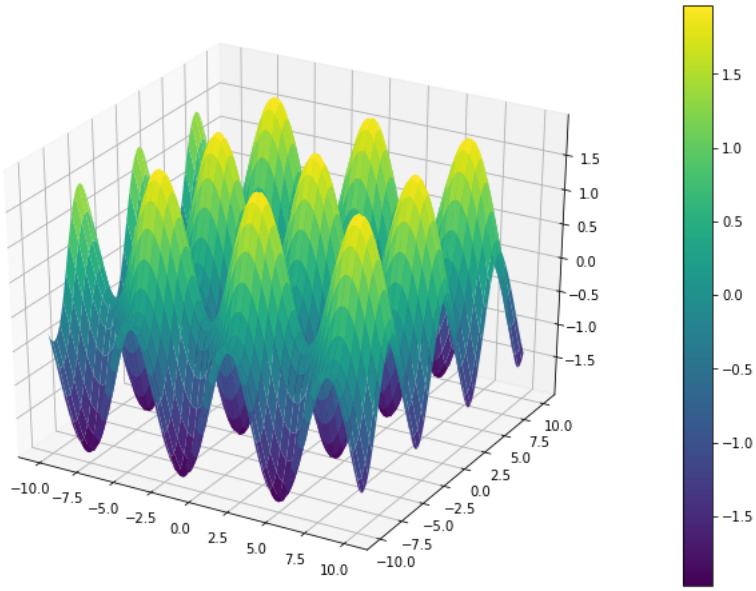
```
fig = plt.figure(figsize=(12,8))

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

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

```
<matplotlib.colorbar.Colorbar at 0x7f5e141ac970>
```



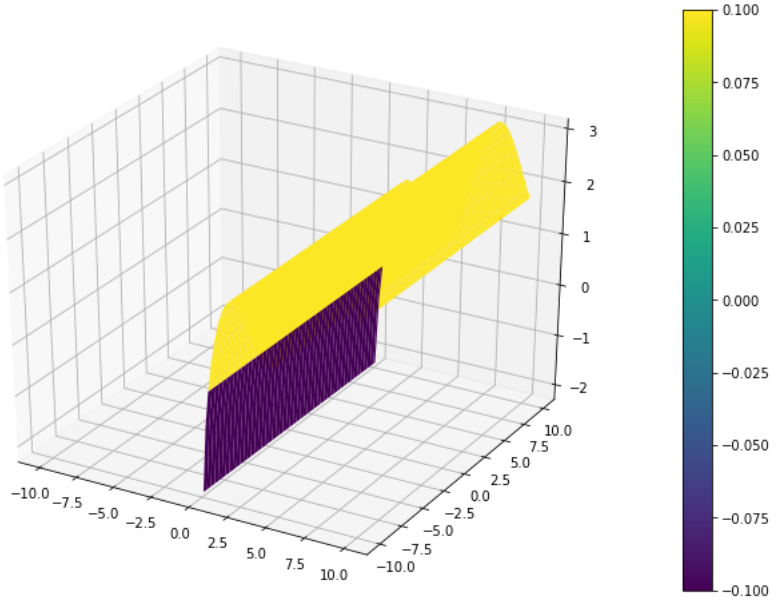

```
z = np.sin(xx) + np.cos(yy)

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 0x7f5e14076be0>
```



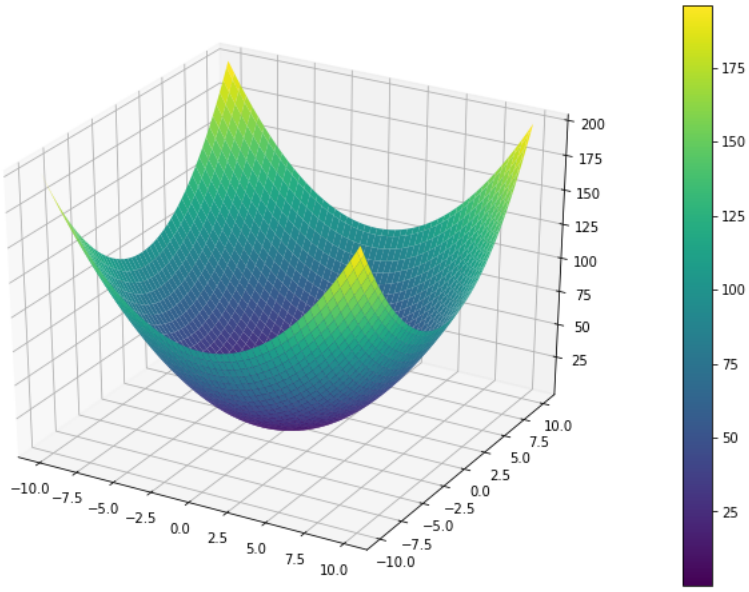

```
z = np.sin(xx) + np.log(xx)

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

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

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

```
<ipython-input-229-bbcd37ea4152>:1: RuntimeWarning: invalid value encountered
  z = np.sin(xx) + np.log(xx)
<ipython-input-229-bbcd37ea4152>:7: UserWarning: Z contains NaN values. This r
  p = ax.plot_surface(xx,yy,z,cmap='viridis')
<matplotlib.colorbar.Colorbar at 0x7f5e139a4a00>
```



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

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

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

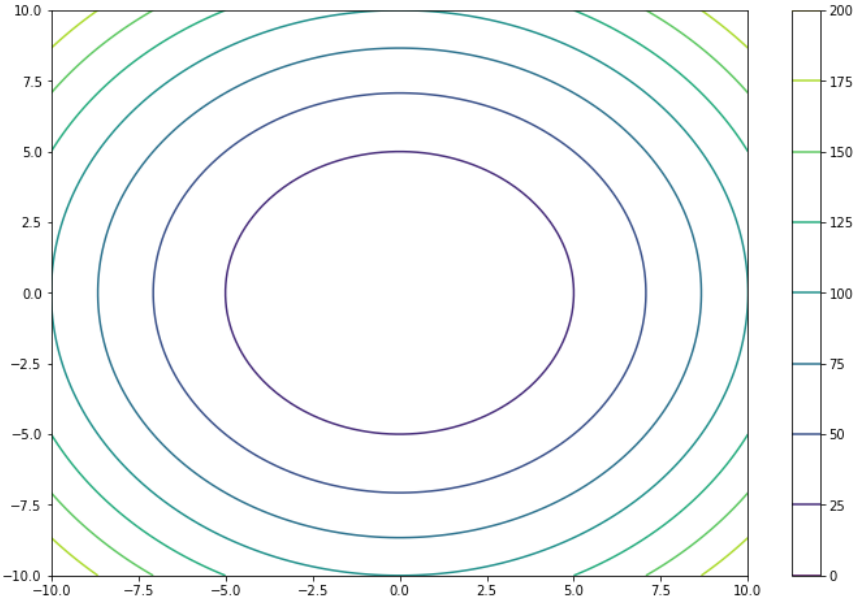
```
<matplotlib.colorbar.Colorbar at 0x7f5e136f8970>
```



Contour Plots

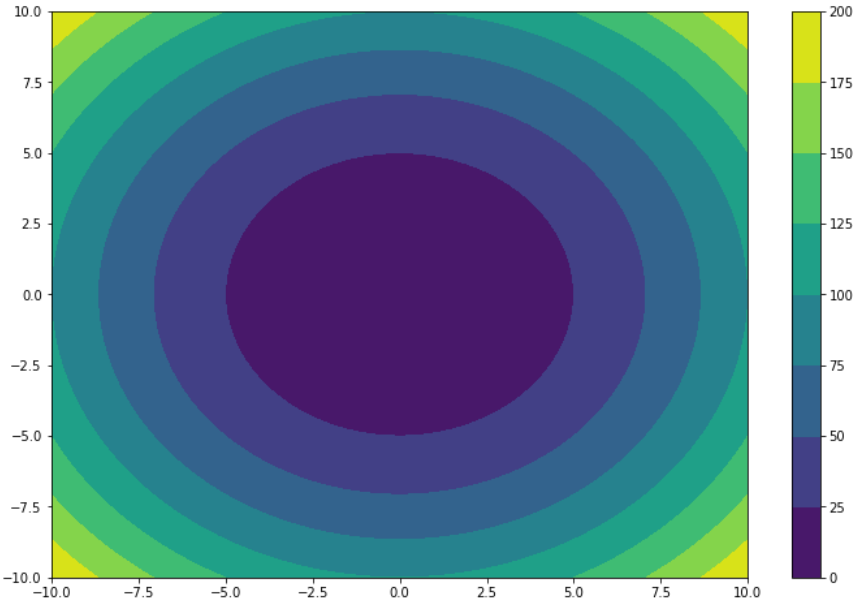
```
fig = plt.figure(figsize=(12,8))

ax = plt.subplot()

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

```
<matplotlib.colorbar.Colorbar at 0x7f5e13580a30>
```



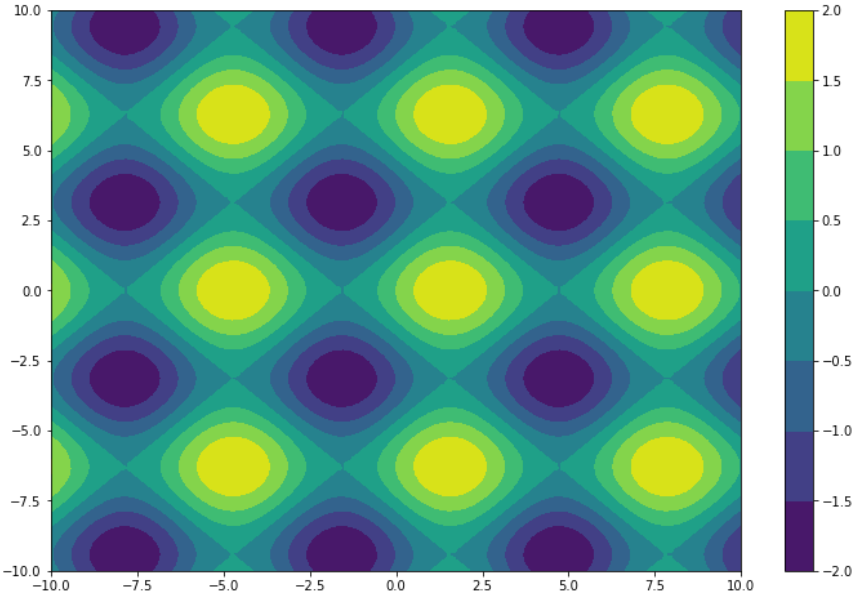

```
fig = plt.figure(figsize=(12,8))

ax = plt.subplot()

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

```
<matplotlib.colorbar.Colorbar at 0x7f5e14f202b0>
```




```
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 0x7f5e14d5a2e0>
```



Heatmap

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

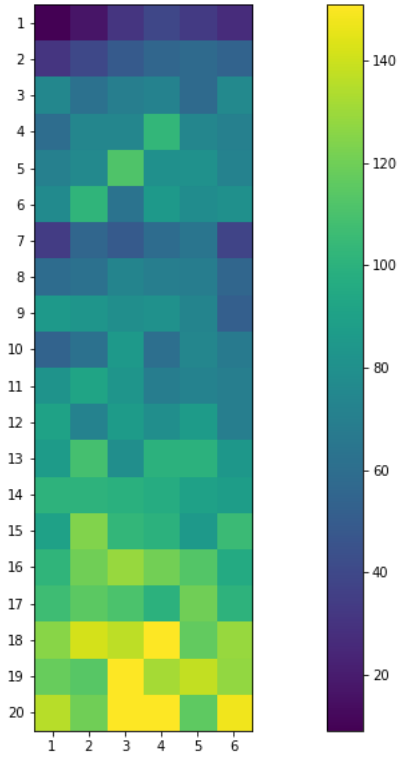
	ID	innings	overs	ballnumber	batter	bowler	non-striker	extra_type	I
0	1312200	1	0	1	YBK Jaiswal	Mohammed Shami	JC Buttler	NaN	
1	1312200	1	0	2	YBK Jaiswal	Mohammed Shami	JC Buttler	legbyes	
2	1312200	1	0	3	JC Buttler	Mohammed Shami	YBK Jaiswal	NaN	
3	1312200	1	0	4	YBK Jaiswal	Mohammed Shami	JC Buttler	NaN	
4	1312200	1	0	5	YBK Jaiswal	Mohammed Shami	JC Buttler	NaN	

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

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

```
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()
```

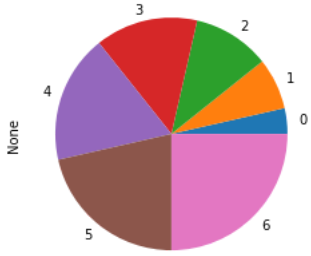
```
<matplotlib.colorbar.Colorbar at 0x7f5e12f98cd0>
```



Pandas Plot()

```
# on a series

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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5e12f0a070>
```

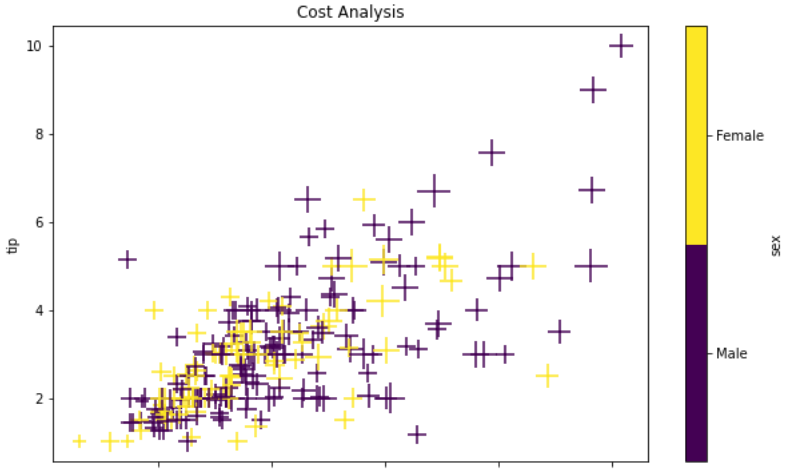



```
# can be used on a dataframe as well

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

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

```
# Scatter plot -> labels -> markers -> figsize -> color -> cmap
tips.plot(kind='scatter',x='total_bill',y='tip',title='Cost Analysis',marker='+',figsize=(10,6),s='size',c='sex',cmap='viric
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5e12b4d760>
```

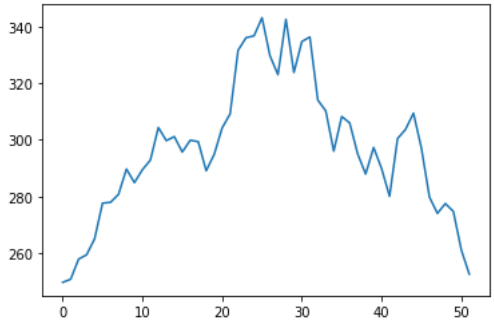



```
# 2d plot
# dataset = 'https://raw.githubusercontent.com/m-mehdi/pandas_tutorials/main/weekly_stocks.csv'

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

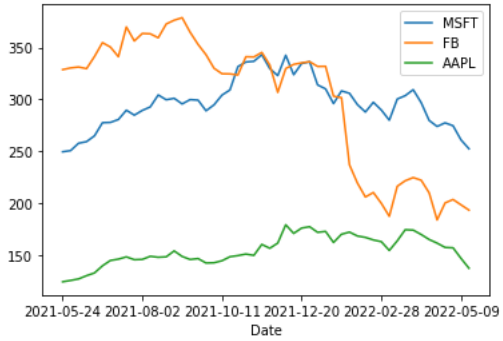
```
# line plot
stocks['MSFT'].plot(kind='line')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5e12a55730>
```



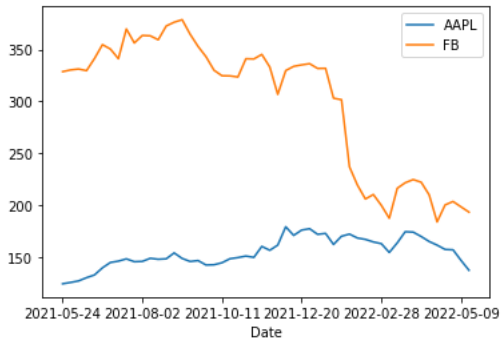

```
stocks.plot(kind='line',x='Date')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5e129f15e0>
```




```
stocks[['Date','AAPL','FB']].plot(kind='line',x='Date')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5e12950fa0>
```

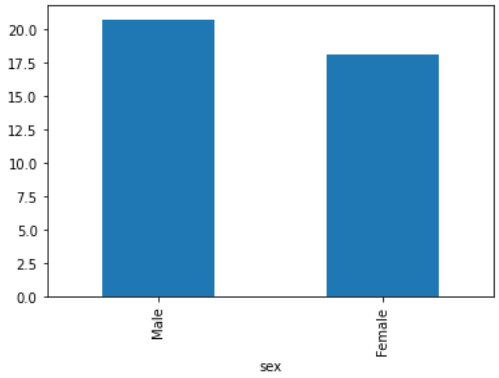



```
# bar chart -> single -> horizontal -> multiple
# using tips
temp = pd.read_csv('/content/batsman_season_record.csv')
temp.head()
```

	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

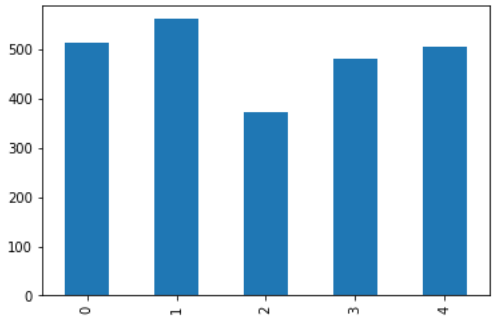
```
tips.groupby('sex')['total_bill'].mean().plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5e12350550>
```



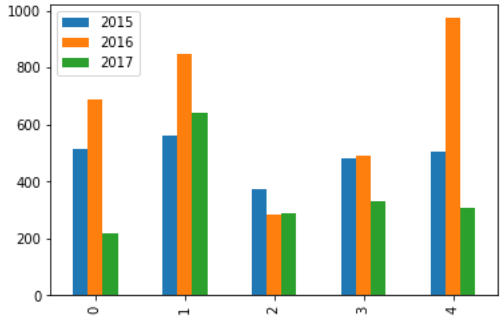

```
temp['2015'].plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5e123ceaf0>
```



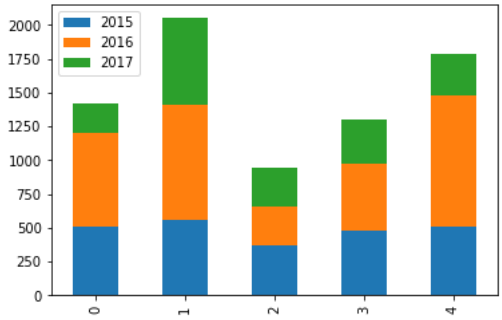

```
temp.plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5e1228fac0>
```



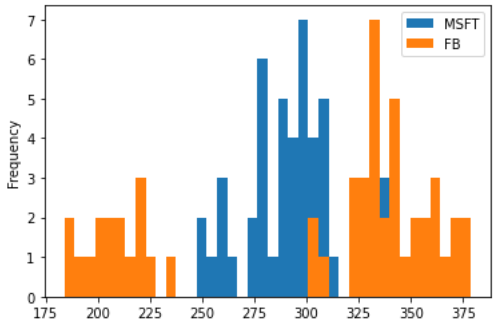

```
# stacked bar chart
temp.plot(kind='bar',stacked=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5e12216e50>
```




```
# histogram
# using stocks

stocks[['MSFT','FB']].plot(kind='hist',bins=40)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5e150247f0>
```



```
# 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()
```

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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5e11e50790>
```