

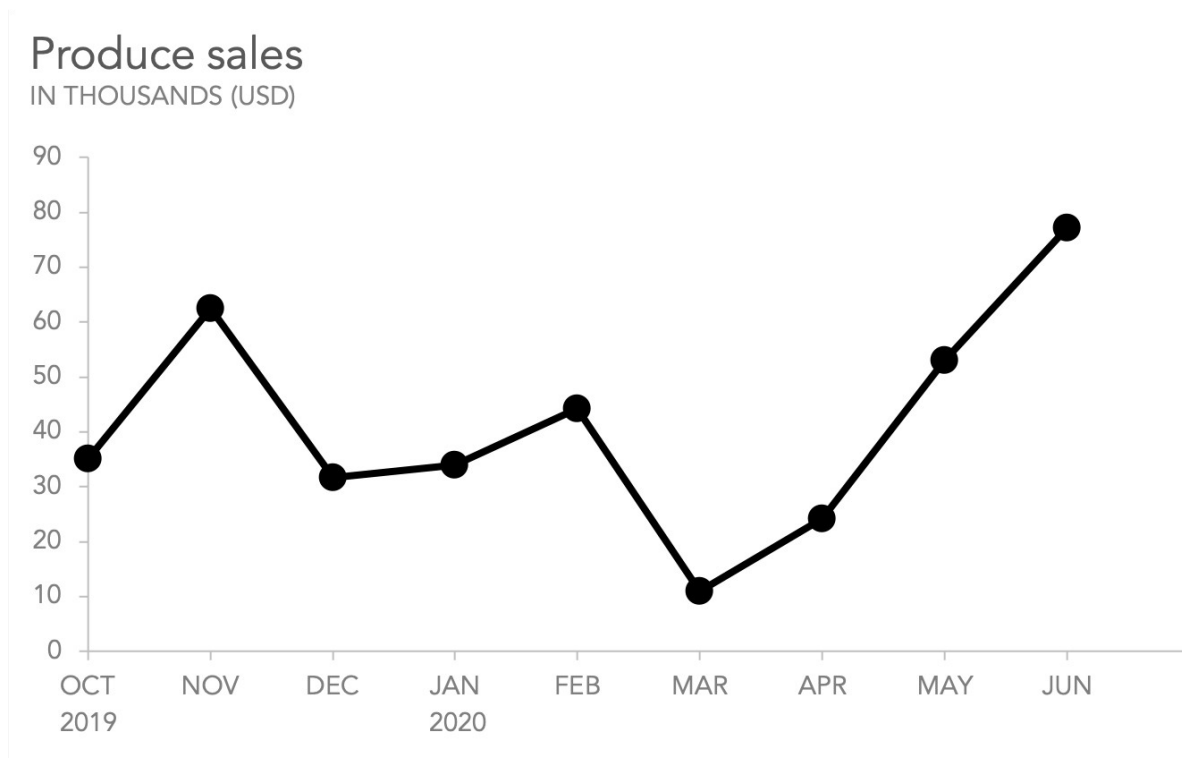
DATA VISUALIZATION

Types of Data

- (row facts and figure is called data)
- Numerical Data (roll-no,age ,etc)
- Categorical Data (jaha pe group involve hota hai jaise ki gender,mobile-compony)

```
# yadi aap single column ya single piece of data pe graph draw karte  
to usse --> univariate analysis bola jata hai  
#yadi aap do column ke upar graph draw karte ho to usse --->  
bivariate analysis bola jata hai  
# yadi aap do se jyada column ke upar graph draw karte hain to usse  
----> multivariate analysis bolte hain  
  
# import the library  
  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns
```

2D Line plot



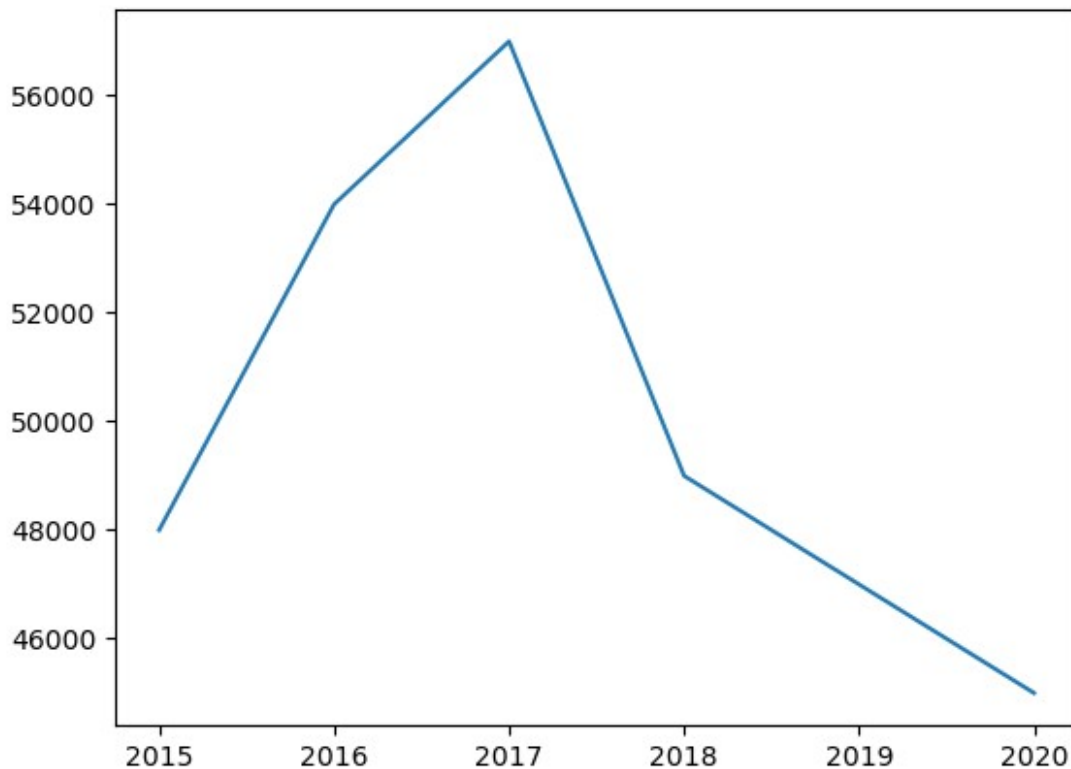
where to use 2D Line Plot

- Bivariate Analysis
- categorical vs numerical and numerical vs numerical data ke bich
- Use case - generally Time series data ko 2D line plot ke help se analyse karte hain

```
# plotting a simple function
price=[48000,54000,57000,49000,47000,45000]
year=[2015,2016,2017,2018,2019,2020]

plt.plot(year,price)  #plt.plot(X-axis,Y-axis) => X-axis pe
generally categorical data and Y-axis pe generally Numerical data pass
karte hain

[<matplotlib.lines.Line2D at 0x1cb24989df0>]
```

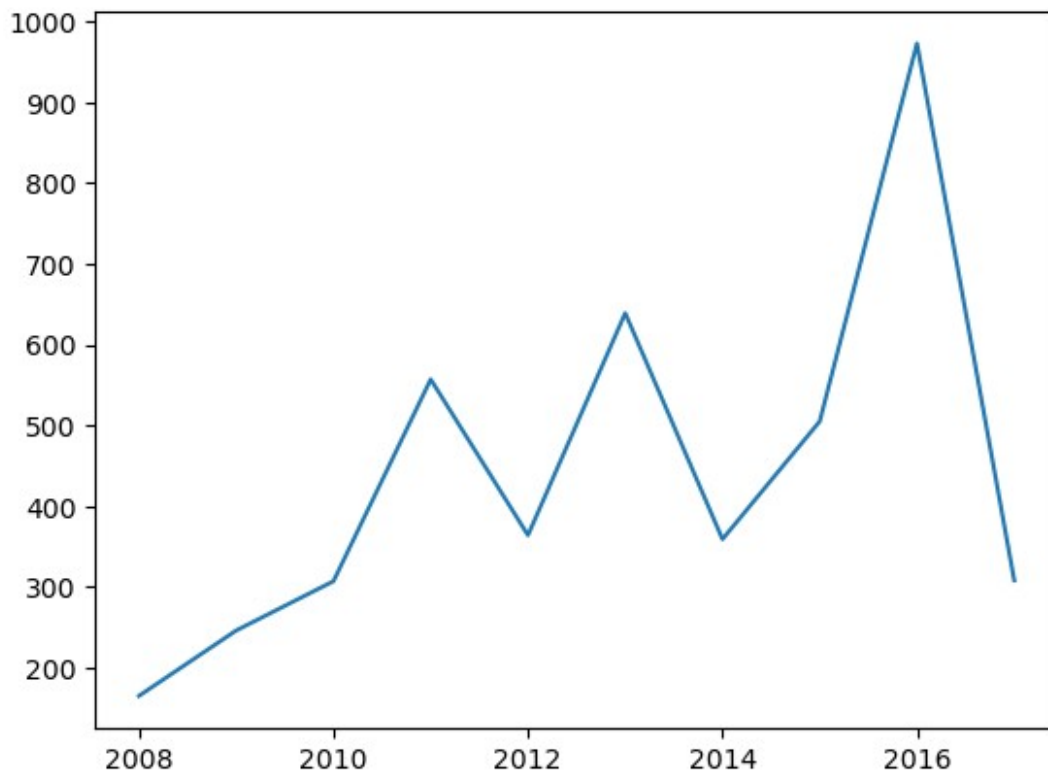


```
# from a pandas dataframe
batsman = pd.read_csv('sharma-kohli.csv')
batsman
```

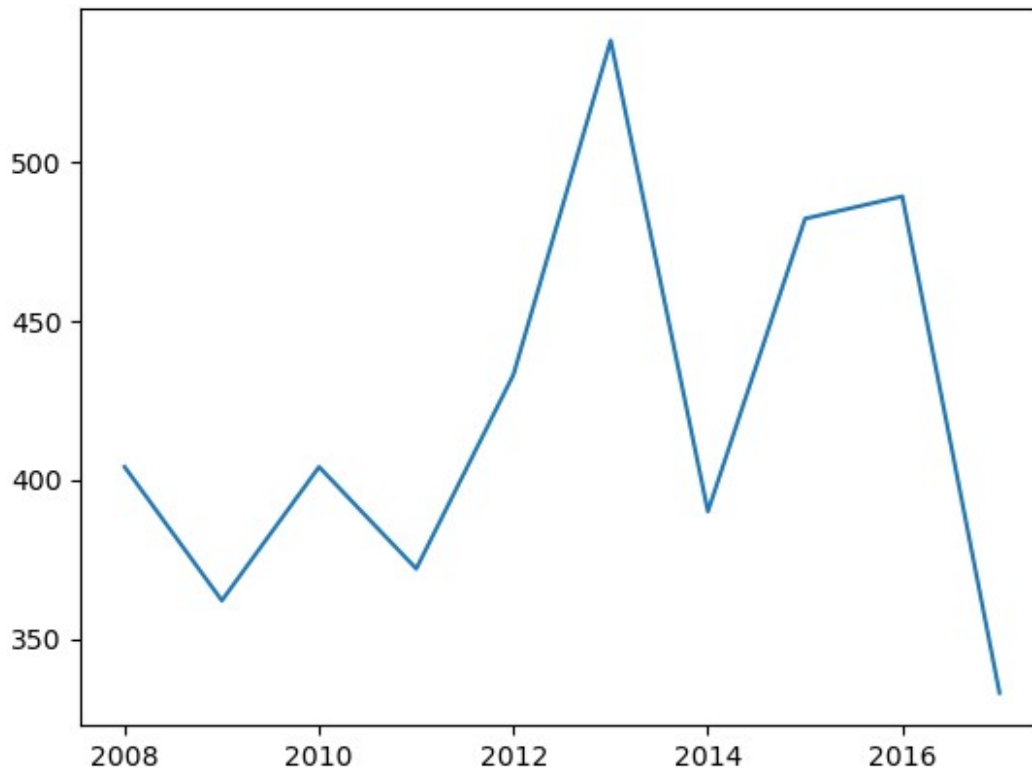
	index	RG Sharma	V Kohli
0	2008	404	165
1	2009	362	246
2	2010	404	307
3	2011	372	557
4	2012	433	364

5	2013	538	639
6	2014	390	359
7	2015	482	505
8	2016	489	973
9	2017	333	308

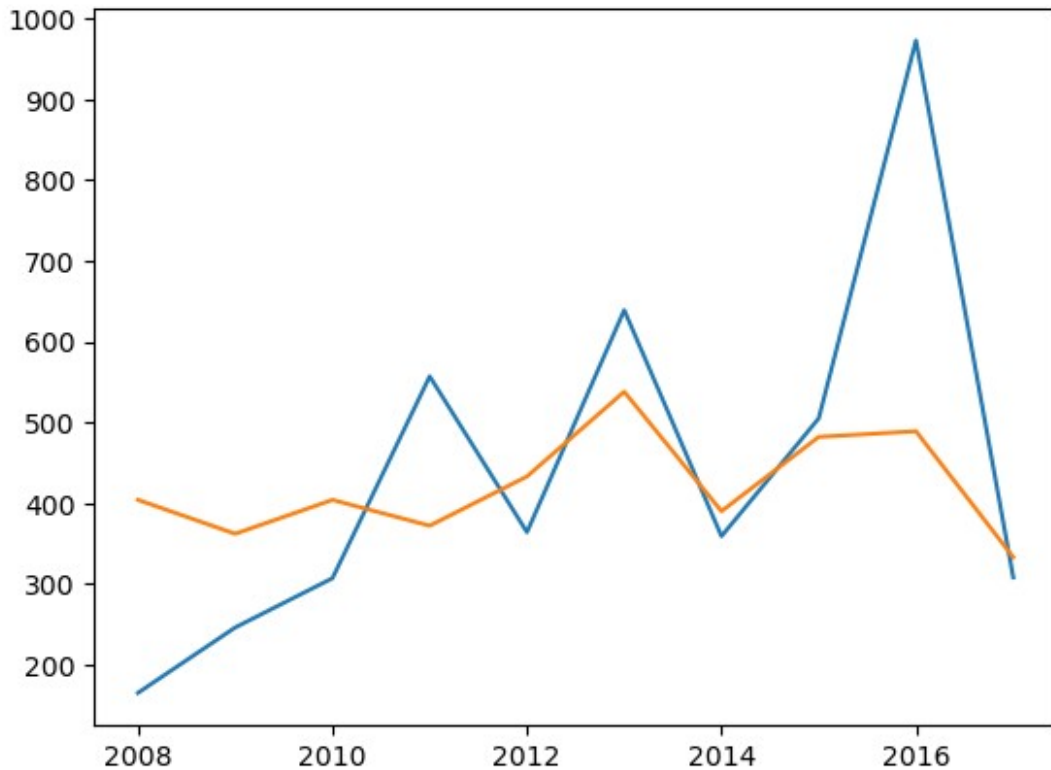
```
plt.plot(batsman['index'],batsman['V Kohli'])  
[<matplotlib.lines.Line2D at 0x1cb251c2390>]
```



```
plt.plot(batsman['index'],batsman['RG Sharma'])  
[<matplotlib.lines.Line2D at 0x1cb249dec90>]
```



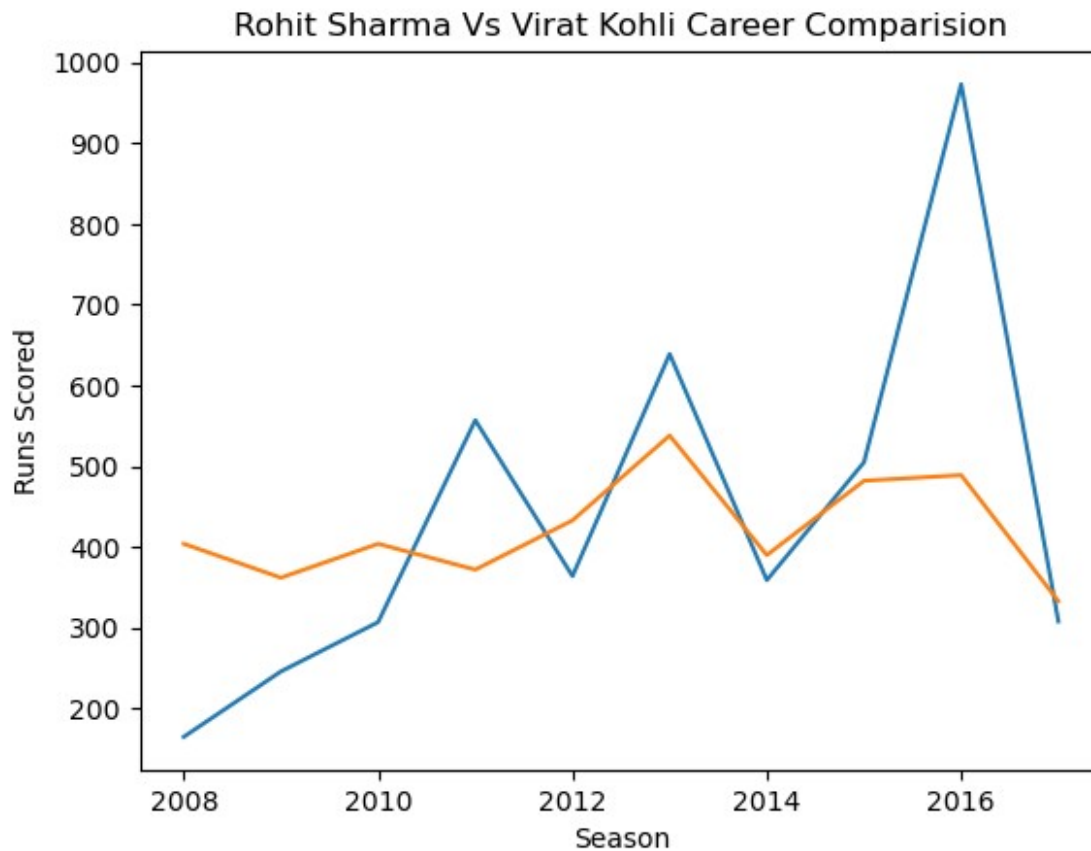
```
# plotting multiple plots    => dono ko ek ke baad ek likh do bass  
Abb dono ka graph ek hi jagah pe hai  
plt.plot(batsman['index'],batsman['V Kohli'])  
plt.plot(batsman['index'],batsman['RG Sharma'])  
  
[<matplotlib.lines.Line2D at 0x1cb24a497c0>]
```



abhi upar bale graph se kuch khas samajh nahi aa raha hai ki kon sa graph kohli ka and kon sa rohit ka hai x-axis and y-axis pe kya hai etc.

```
# labels title => isme ye sab kaam karenge
plt.plot(batsman['index'],batsman['V Kohli'])
plt.plot(batsman['index'],batsman['RG Sharma'])

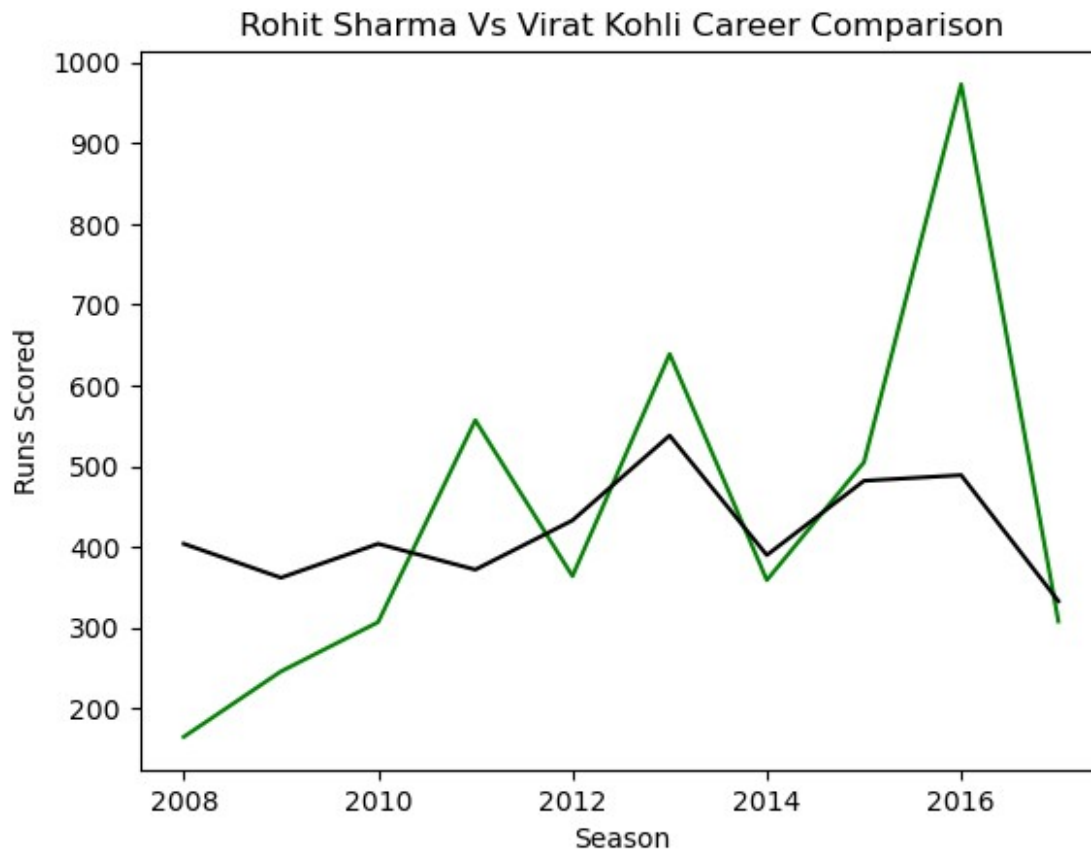
plt.title('Rohit Sharma Vs Virat Kohli Career Comparision')
plt.xlabel('Season')
plt.ylabel('Runs Scored')
Text(0, 0.5, 'Runs Scored')
```



```
# colors(hex) and line(width and style) and marker(size)
plt.plot(batsman['index'],batsman['V Kohli'],color='green')
plt.plot(batsman['index'],batsman['RG Sharma'],color='black')    # hum
hex code bhi provide kar sakte hain

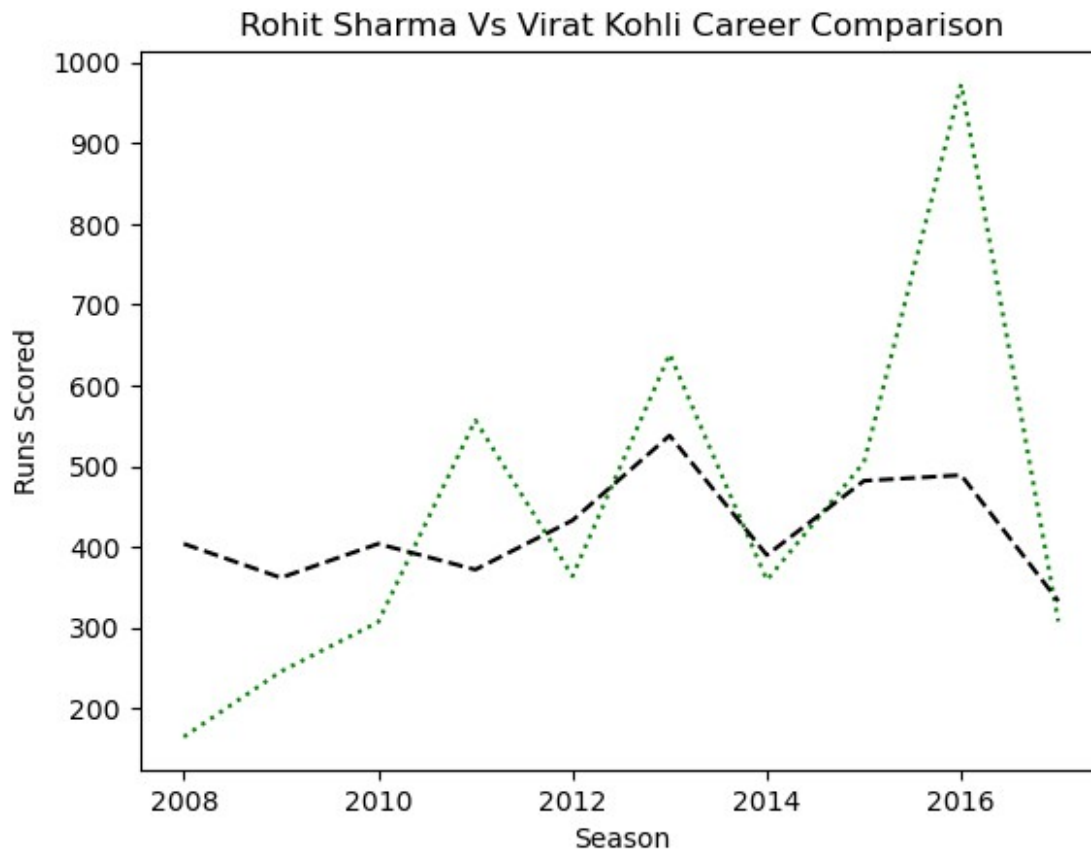
plt.title('Rohit Sharma Vs Virat Kohli Career Comparision')
plt.xlabel('Season')
plt.ylabel('Runs Scored')

Text(0, 0.5, 'Runs Scored')
```



```
# what is hume solid line nahi chahaiye
plt.plot(batsman['index'],batsman['V
Kohli'],color='green',linestyle='dotted') #
solid,dotted,dashed,dashdot -> ye sare option available hai
plt.plot(batsman['index'],batsman['RG
Sharma'],color='black',linestyle='dashed')

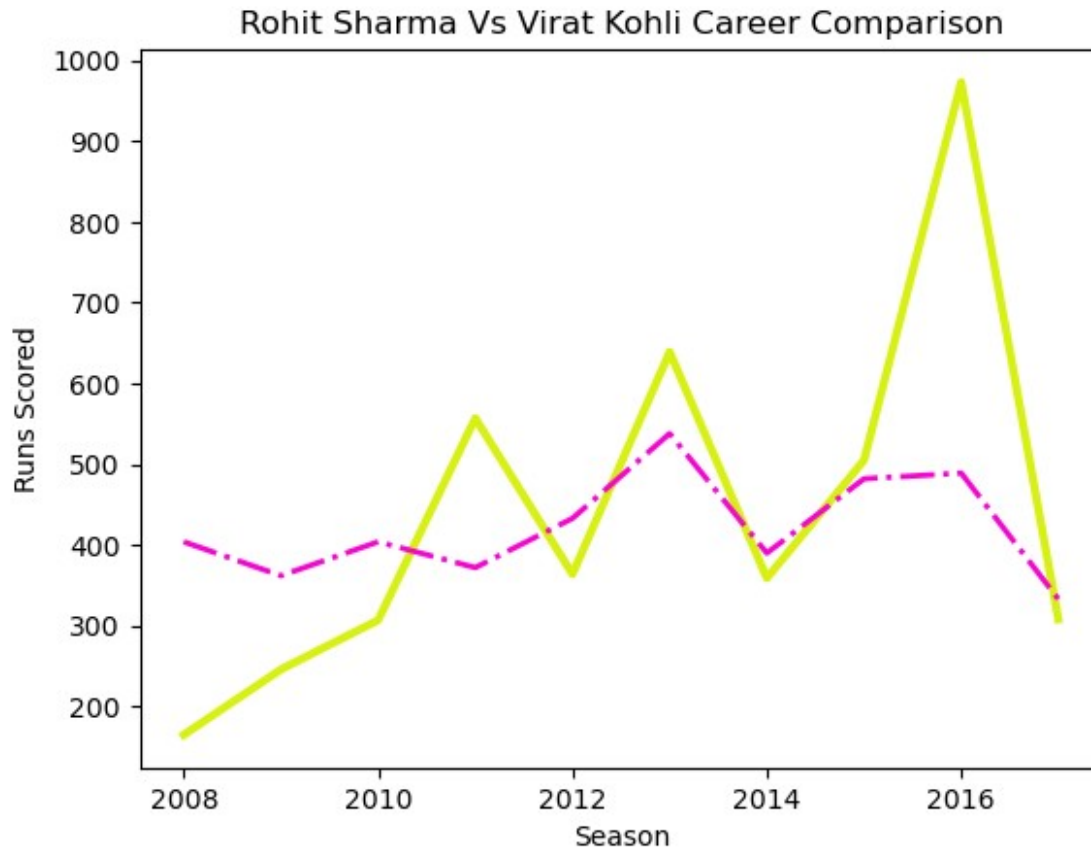
plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
plt.xlabel('Season')
plt.ylabel('Runs Scored')
Text(0, 0.5, 'Runs Scored')
```



```
# line ka width bhi change kar sakte hain
plt.plot(batsman['index'],batsman['V
Kohli'],color='#D9F10F',linestyle='solid',linewidth=3)
plt.plot(batsman['index'],batsman['RG
Sharma'],color='#FC00D6',linestyle='dashdot',linewidth=2)

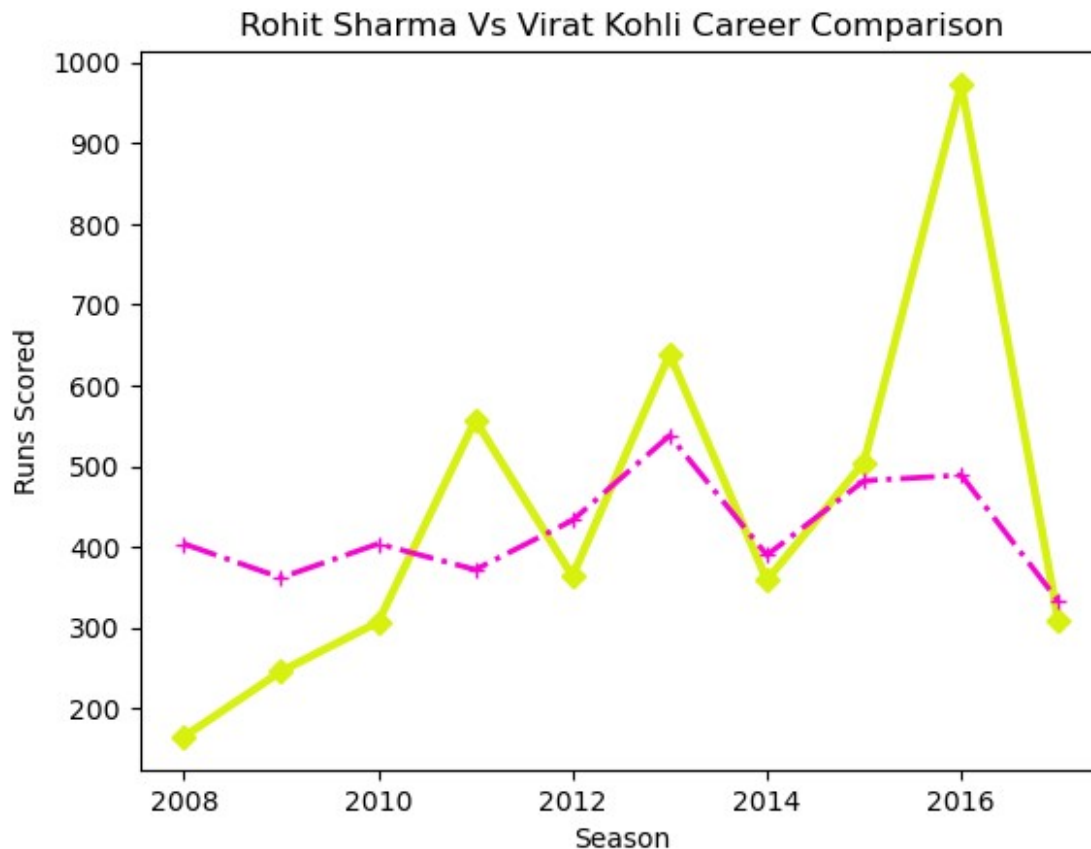
plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
plt.xlabel('Season')
plt.ylabel('Runs Scored')

Text(0, 0.5, 'Runs Scored')
```

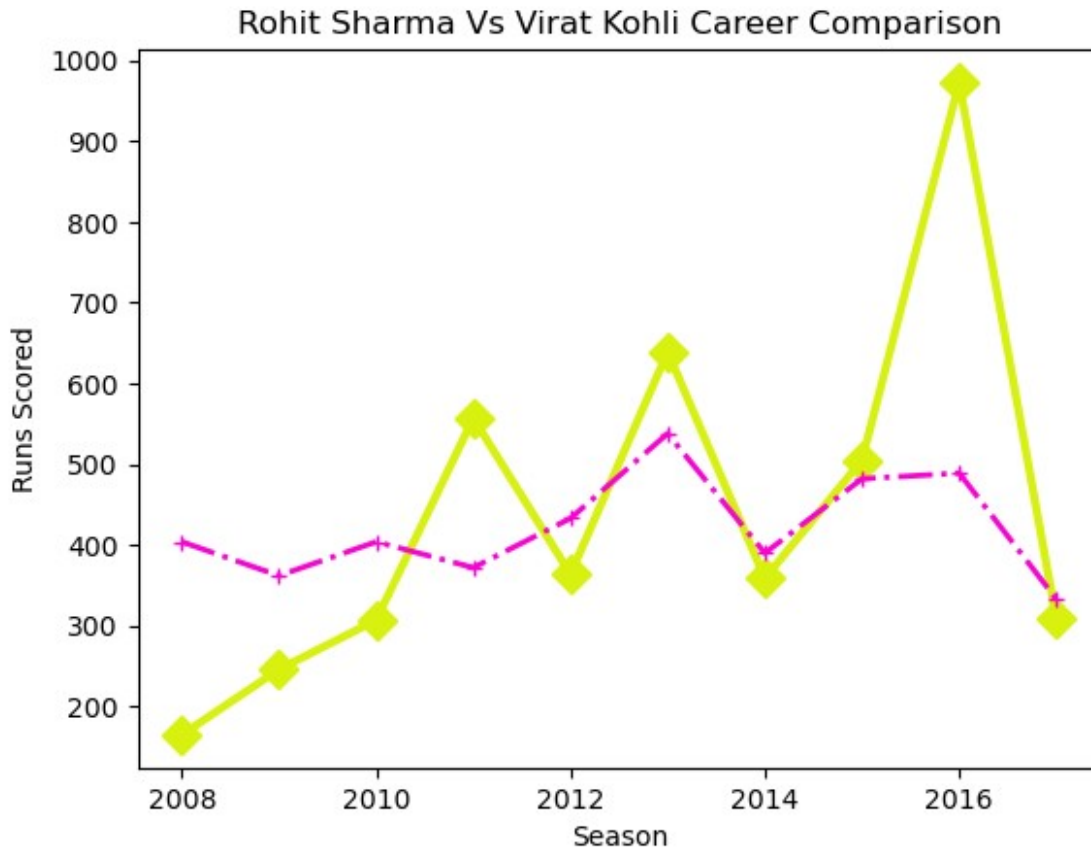
```
# if we want all the joining points(markers)
plt.plot(batsman['index'],batsman['V
Kohli'],color='#D9F10F',linestyle='solid',linewidth=3,marker='D') #
D-> diamond, + , . , > ,< ,o,
plt.plot(batsman['index'],batsman['RG
Sharma'],color='#FC00D6',linestyle='dashdot',linewidth=2,marker='+')

plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
plt.xlabel('Season')
plt.ylabel('Runs Scored')
Text(0, 0.5, 'Runs Scored')
```



```
plt.plot(batsman['index'],batsman['V
Kohli'],color='#D9F10F',linestyle='solid',linewidth=3,marker='D',marke
rsize=10)
plt.plot(batsman['index'],batsman['RG
Sharma'],color='#FC00D6',linestyle='dashdot',linewidth=2,marker='+')

plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
plt.xlabel('Season')
plt.ylabel('Runs Scored')
Text(0, 0.5, 'Runs Scored')
```

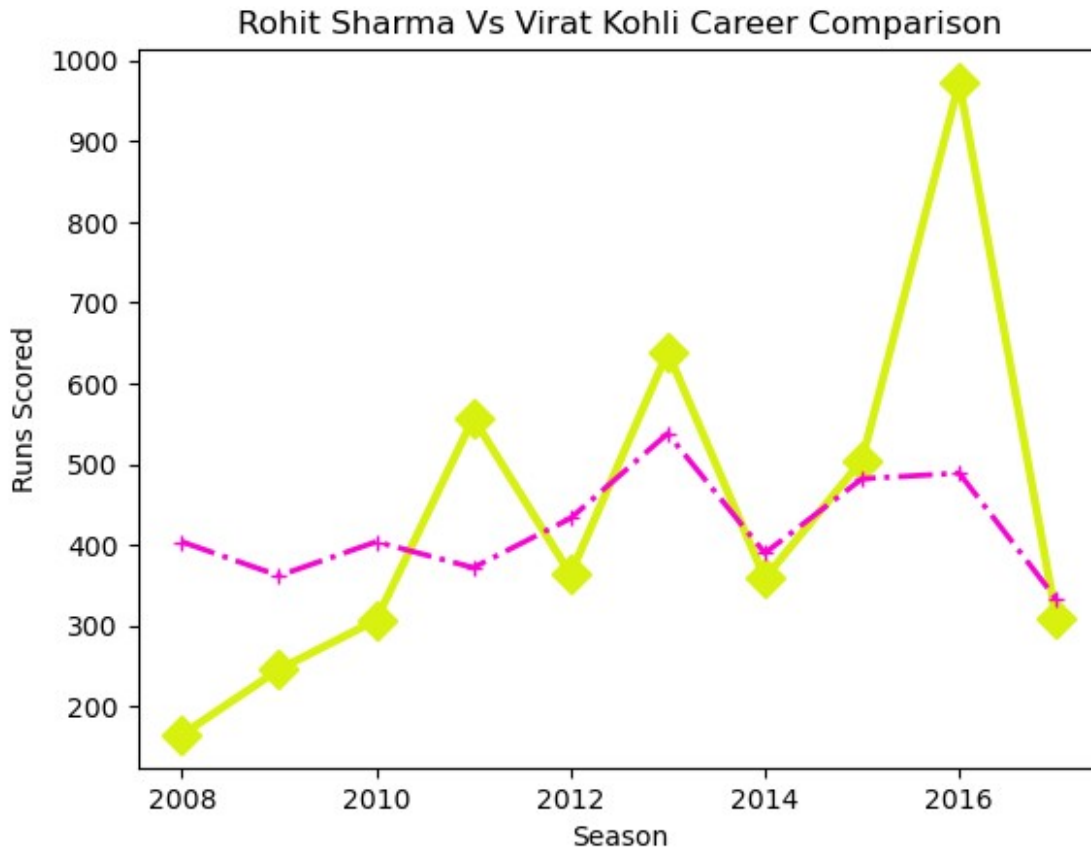


```
# har line ko name de sakte hain
plt.plot(batsman['index'],batsman['V
Kohli'],color='#D9F10F',linestyle='solid',linewidth=3,marker='D',marke
rsize=10,label='Virat')
plt.plot(batsman['index'],batsman['RG
Sharma'],color='#FC00D6',linestyle='dashdot',linewidth=2,marker='+',la
bel='Rohit')
```

abhi to humne label lagaya hai Virat and Rohit name se but wo dikh nahi raha hai graph me to iske lye ek fuunction i.e legend() ko call karna hota hai

```
plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
plt.xlabel('Season')
plt.ylabel('Runs Scored')
```

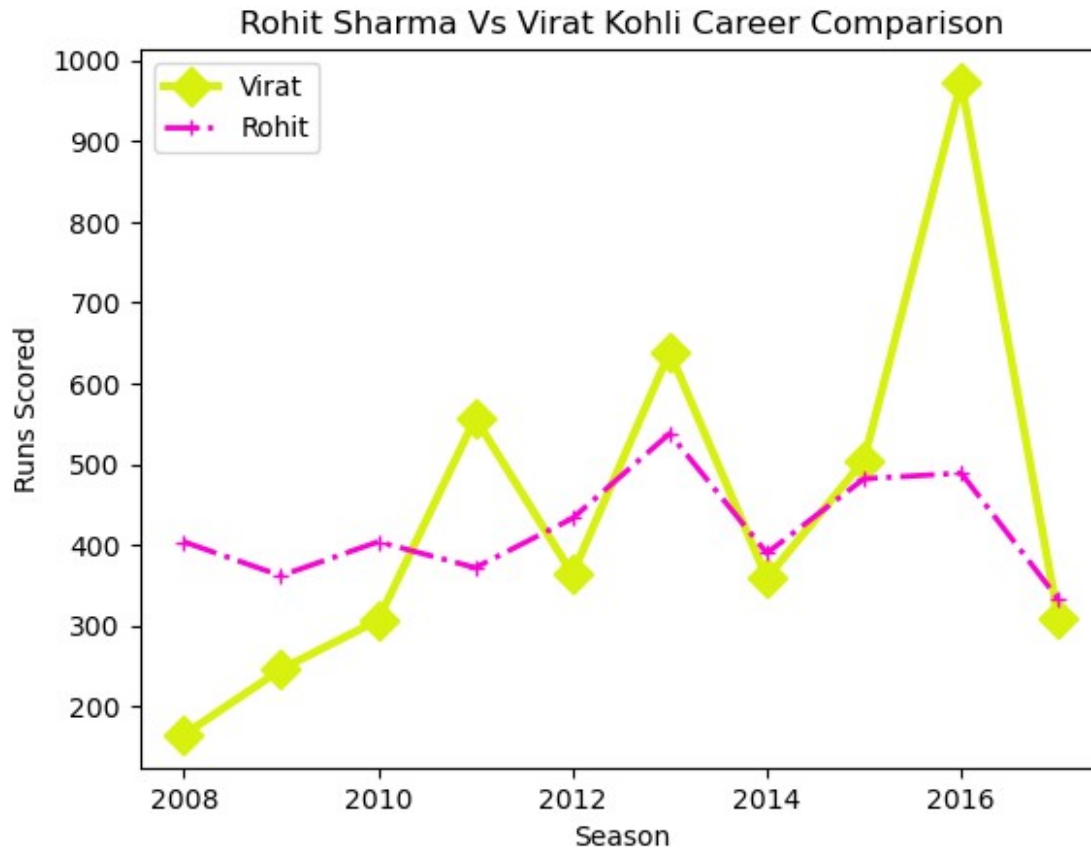
```
Text(0, 0.5, 'Runs Scored')
```



```
plt.plot(batsman['index'],batsman['Virat Kohli'],color='#D9F10F',linestyle='solid',linewidth=3,marker='D',marker_size=10,label='Virat')
plt.plot(batsman['index'],batsman['Rohit Sharma'],color='#FC00D6',linestyle='dashdot',linewidth=2,marker='+',label='Rohit')

plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
plt.xlabel('Season')
plt.ylabel('Runs Scored')
plt.legend() #now you can see ek box ban gya graph me aap iss box ke location ko bhi adjust kar sakte ho by default location best hota hai wo apne aap # best position dhund leta hai

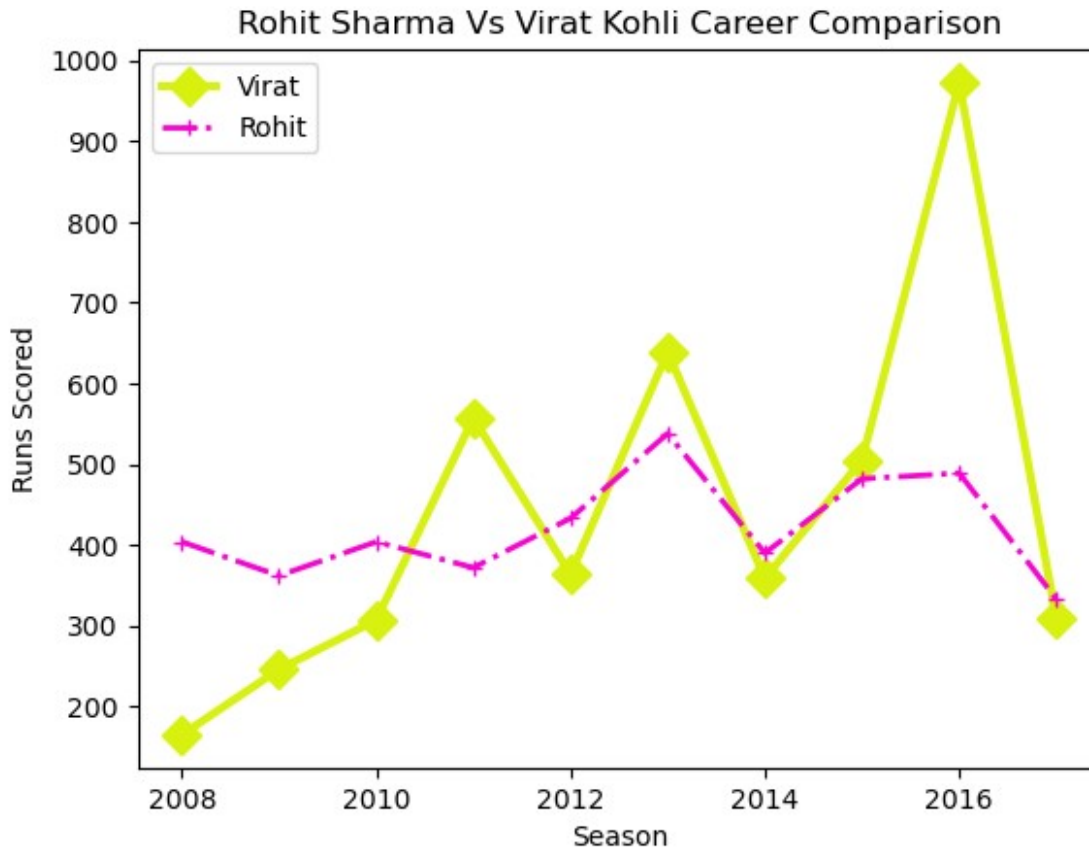
<matplotlib.legend.Legend at 0x1cb2568f170>
```



```
plt.plot(batsman['index'],batsman['V
Kohli'],color='#D9F10F',linestyle='solid',linewidth=3,marker='D',marke
rsize=10,label='Virat')
plt.plot(batsman['index'],batsman['RG
Sharma'],color='#FC00D6',linestyle='dashdot',linewidth=2,marker='+',la
bel='Rohit')

plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
plt.xlabel('Season')
plt.ylabel('Runs Scored')
plt.legend(loc='best')

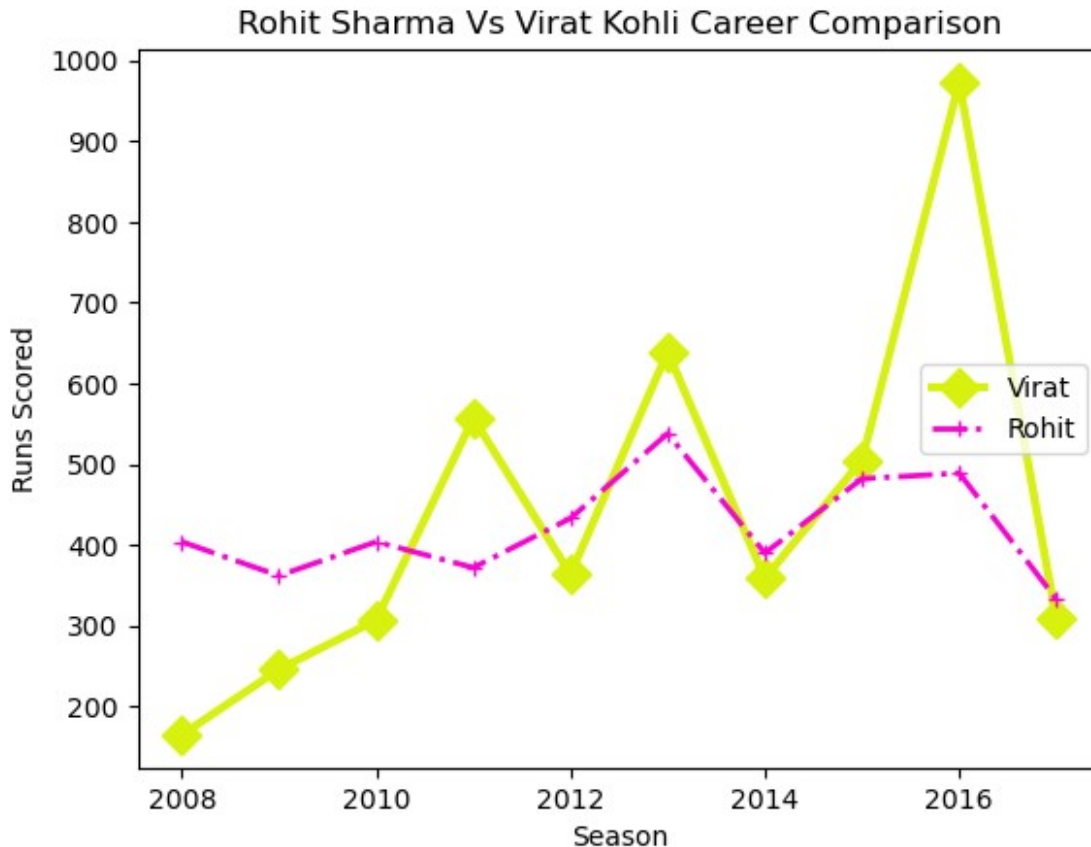
<matplotlib.legend.Legend at 0x1cb26085f70>
```



```
plt.plot(batsman['index'],batsman['V
Kohli'],color='#D9F10F',linestyle='solid',linewidth=3,marker='D',marke
rsize=10,label='Virat')
plt.plot(batsman['index'],batsman['RG
Sharma'],color='#FC00D6',linestyle='dashdot',linewidth=2,marker='+',la
bel='Rohit')

plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
plt.xlabel('Season')
plt.ylabel('Runs Scored')
plt.legend(loc='right')

<matplotlib.legend.Legend at 0x1cb25996000>
```



all options for adjusting label box

- best
- upper right
- upper left
- lower left
- lower right
- right
- center left
- center right
- lower center

`## limiting axes`

`price=[48000,54000,57000,49000,47000,4500000]` *# isme outliers hai qki ek phone ki cost bahut jyada hai compare to other to iss situation me another*

value ke lye curve flat ho jayega to isse hum trim kar sakte hain ye aaisie seenario me kaam aa

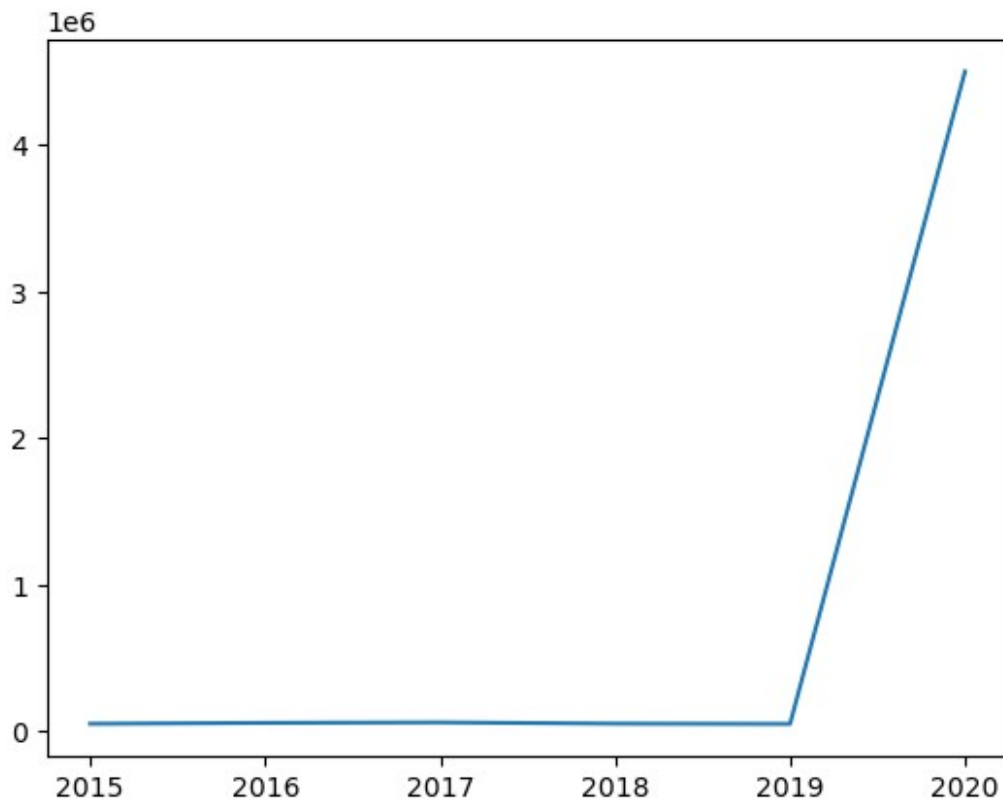
sakta hai suppose aap daily apne video ke views ko rack kar rahe ho but achanak se ek video viral

```
# ho gya to agar aap  
views ka graph banaoge to usme outliers ke karan graph sahi nahi ban  
payega
```

```
year=[2015,2016,2017,2018,2019,2020]
```

```
plt.plot(year,price)
```

```
[<matplotlib.lines.Line2D at 0x1cb274faf30>]
```



```
price=[48000,54000,57000,49000,47000,45000]
```

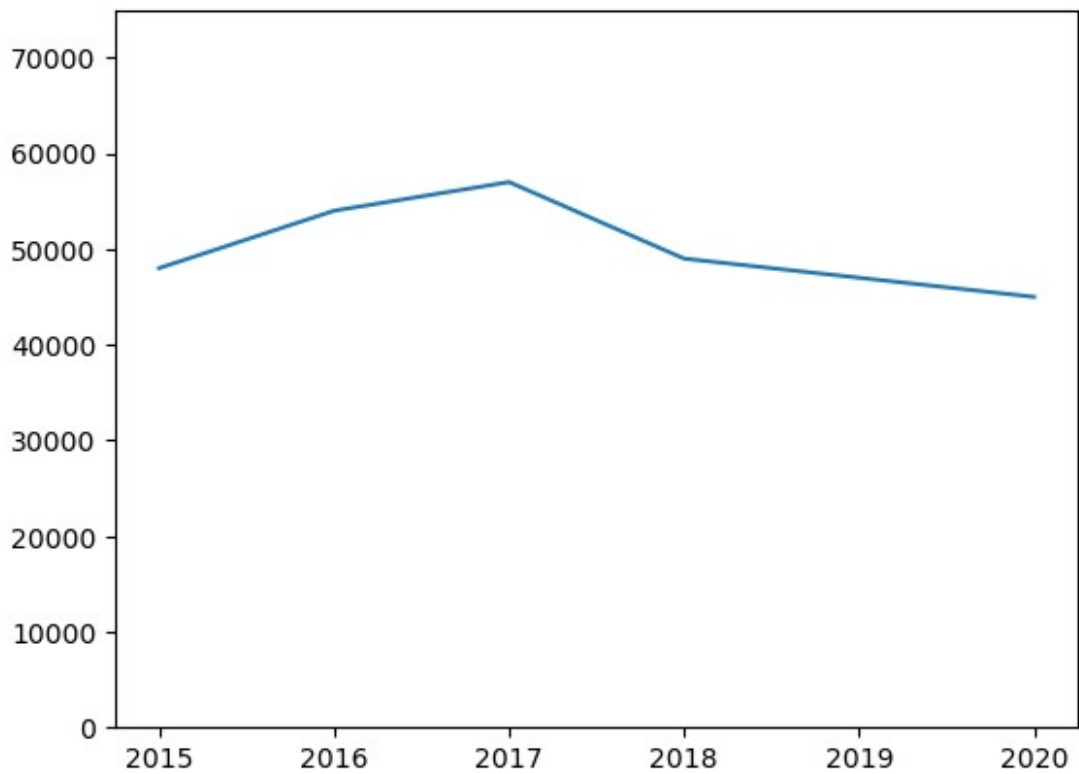
```
year=[2015,2016,2017,2018,2019,2020]
```

```
plt.plot(year,price)
```

```
# trimming
```

```
plt.ylim(0,75000) # abb isi range ko mainly focus karega
```

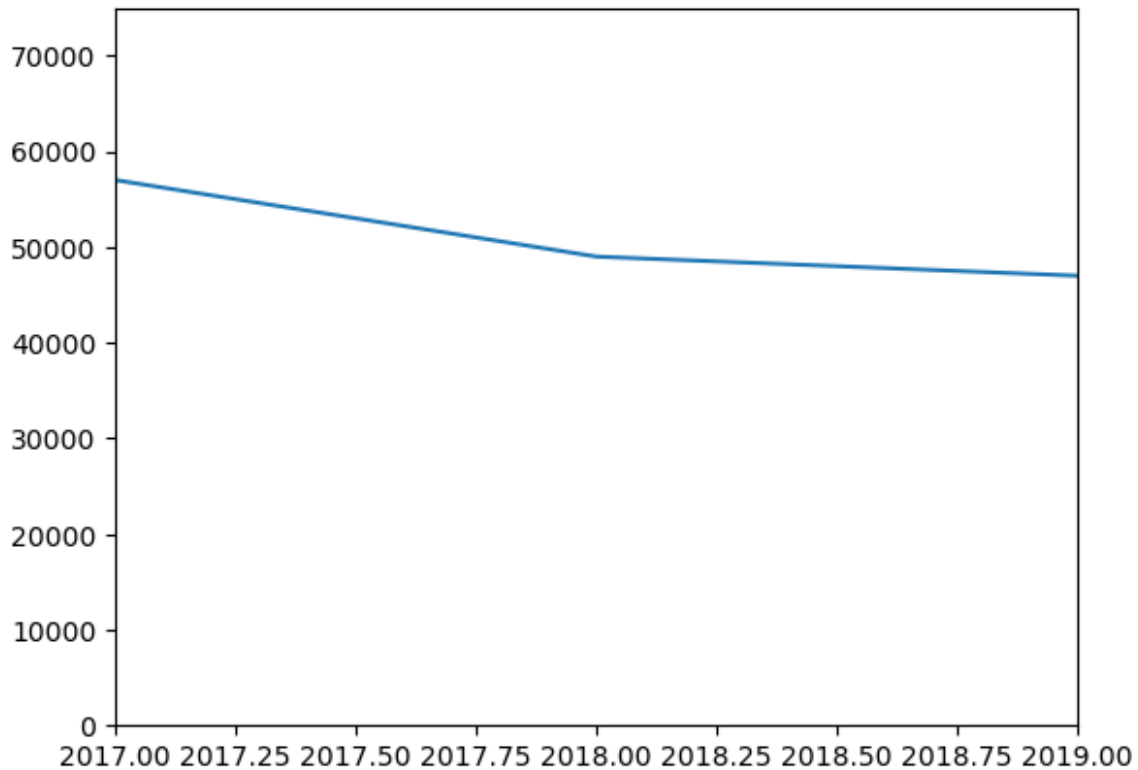
```
(0.0, 75000.0)
```

```
# similar x- axis pe bhi trim kar sakte hain
price=[48000,54000,57000,49000,47000,45000]
year=[2015,2016,2017,2018,2019,2020]

plt.plot(year,price)
# trimming
plt.ylim(0,75000)
plt.xlim(2017,2019)

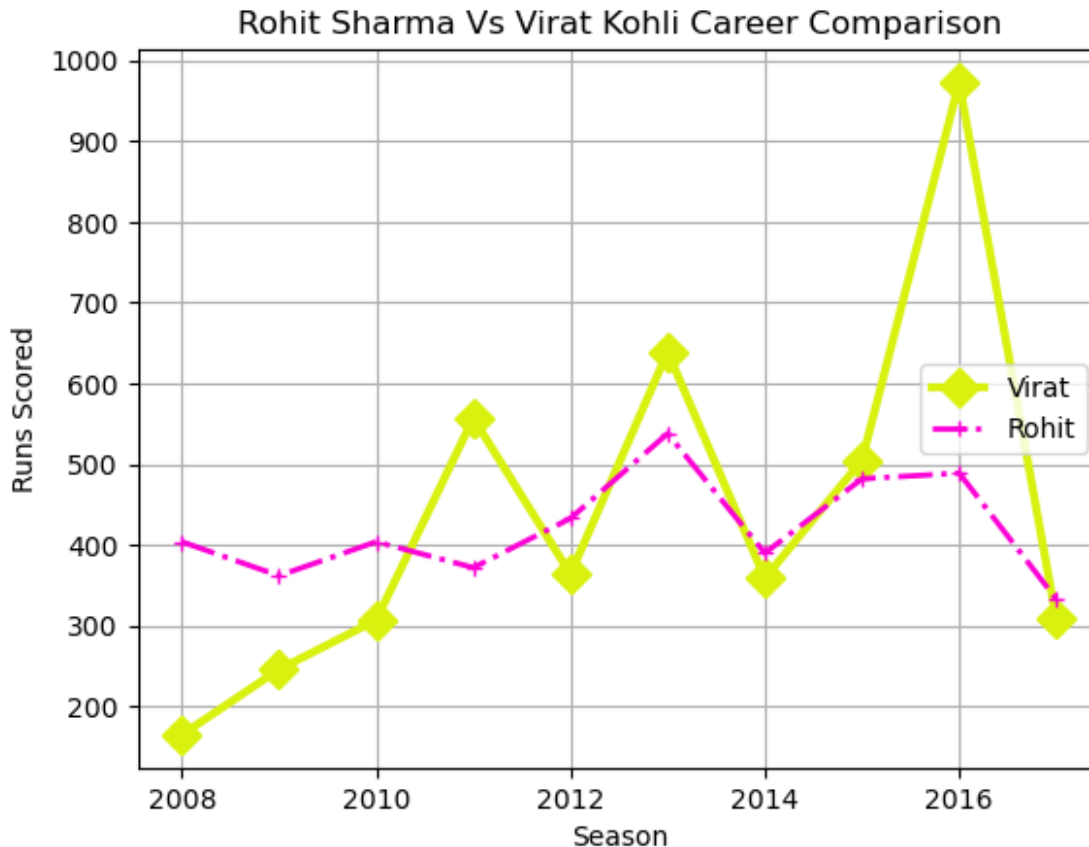
(2017.0, 2019.0)
```



```
## grid
plt.plot(batsman['index'],batsman['V
Kohli'],color='#D9F10F',linestyle='solid',linewidth=3,marker='D',marke
rsize=10,label='Virat')
plt.plot(batsman['index'],batsman['RG
Sharma'],color='#FC00D6',linestyle='dashdot',linewidth=2,marker='+',la
bel='Rohit')

plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
plt.xlabel('Season')
plt.ylabel('Runs Scored')
plt.legend(loc='right')

plt.grid()
```



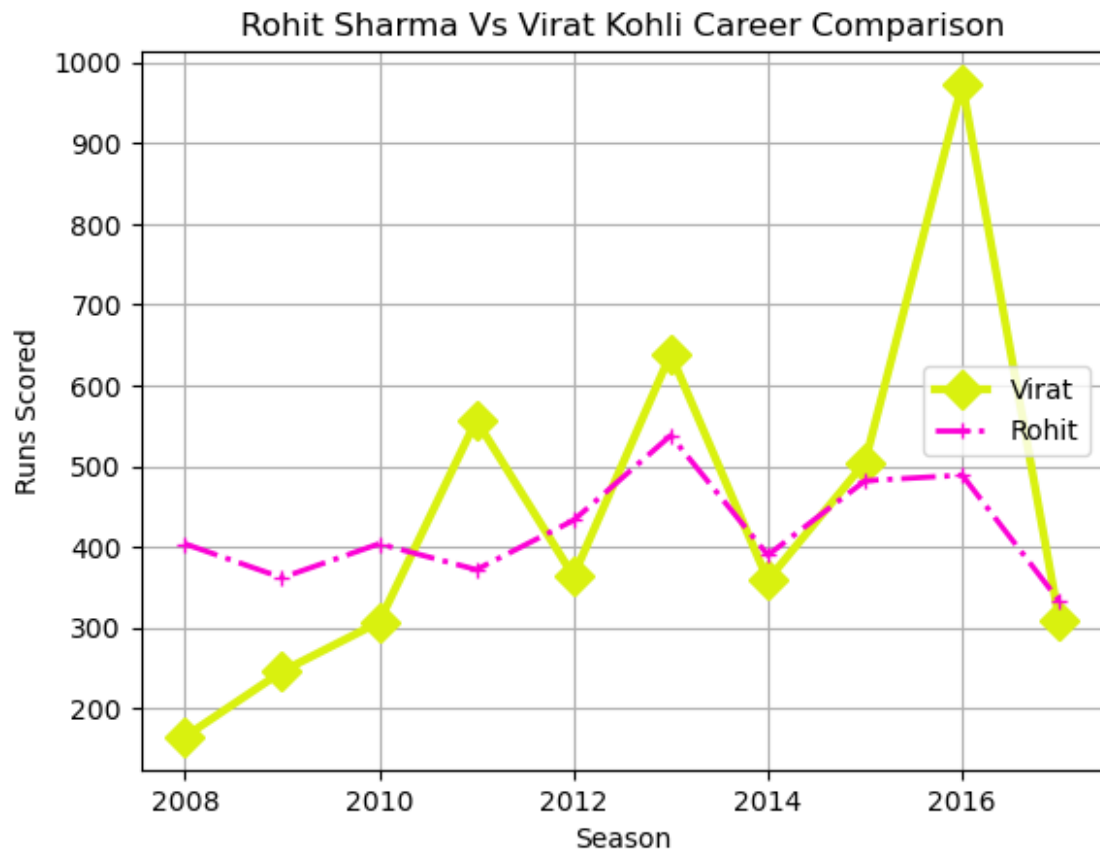
```
# show
plt.plot(batsman['index'],batsman['V
Kohli'],color='#D9F10F',linestyle='solid',linewidth=3,marker='D',marke
rsize=10,label='Virat')
plt.plot(batsman['index'],batsman['RG
Sharma'],color='#FC00D6',linestyle='dashdot',linewidth=2,marker='+',la
bel='Rohit')

plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
plt.xlabel('Season')
plt.ylabel('Runs Scored')
plt.legend(loc='right')

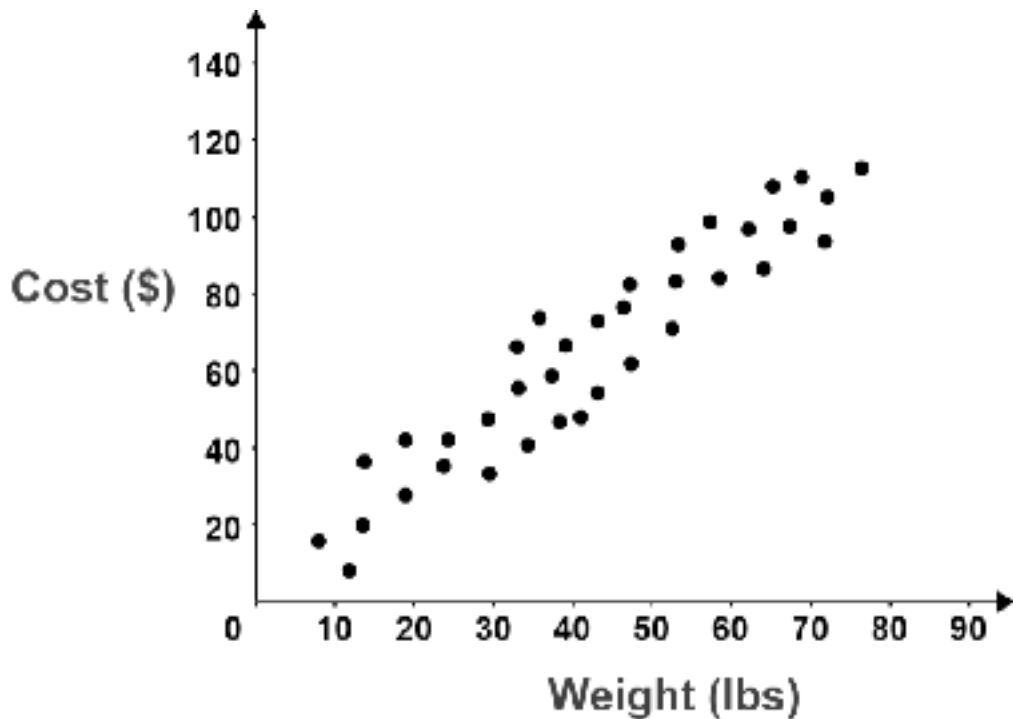
plt.grid()

plt.show() # ye main function hai graph ko show karane ke lye jab
aap pycharm ya vs code me graph banaoge to iss function ko call karna
hoga qki wo

# console base software hai
```



Scatter Plots



where to use

- Bivariate Analysis
- numerical vs numerical (mainly although categorical ke sath bhi kar sakte hain but koi sanse nahi banta hai)
- Use case - Finding correlation (jab do quantity ke bich me relation find karna hota hai)

*## Actually 2D line plot and scatter plot me koi difference nahi hai
scatter plot me jo point aate hain usi ko line se milane pe 2D plot
banta hai*

plt.scatter simple function

```
x = np.linspace(-10,10,50)  
y = 10*x + 3 + np.random.randint(0,300,50)
```

x

```
array([-10.        , -9.59183673, -9.18367347, -8.7755102 ,  
       -8.36734694, -7.95918367, -7.55102041, -7.14285714,  
       -6.73469388, -6.32653061, -5.91836735, -5.51020408,  
       -5.10204082, -4.69387755, -4.28571429, -3.87755102,  
       -3.46938776, -3.06122449, -2.65306122, -2.24489796,  
       -1.83673469, -1.42857143, -1.02040816, -0.6122449 ,  
       -0.20408163,  0.20408163,  0.6122449 ,  1.02040816,  
        1.42857143,  1.83673469,  2.24489796,  2.65306122,
```

```

3.06122449, 3.46938776, 3.87755102, 4.28571429,
4.69387755, 5.10204082, 5.51020408, 5.91836735,
6.32653061, 6.73469388, 7.14285714, 7.55102041,
7.95918367, 8.36734694, 8.7755102 , 9.18367347,
9.59183673, 10. ])
```

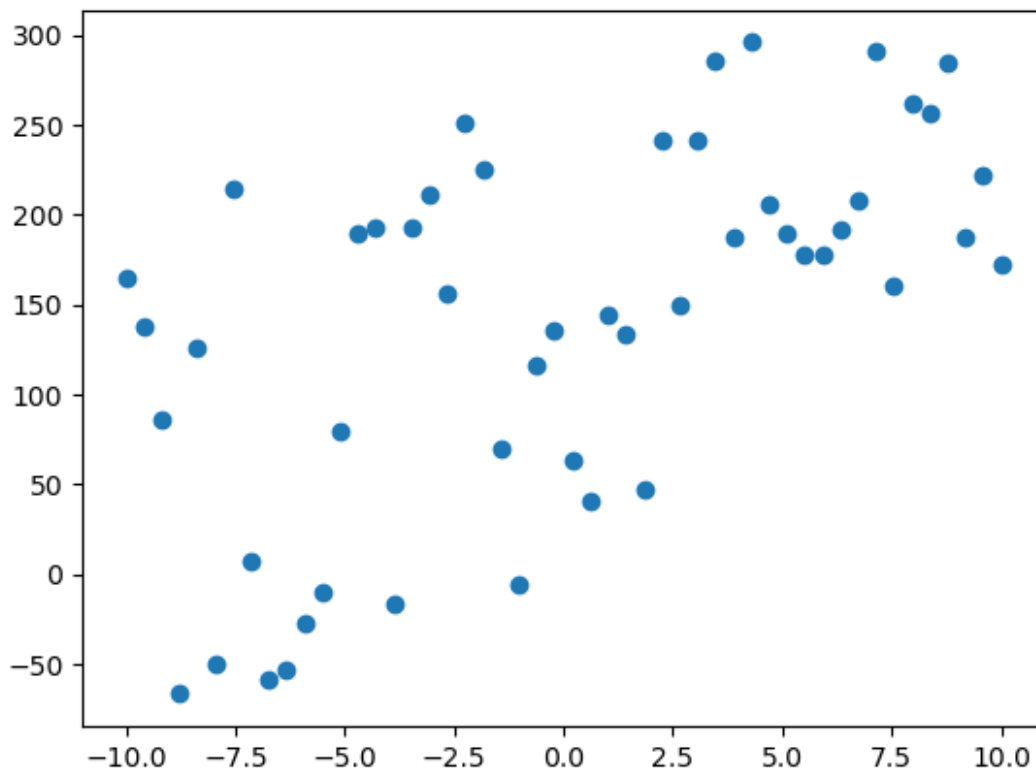
y

```

array([165.      , 138.08163265, 86.16326531, -66.75510204,
 126.32653061, -49.59183673, 214.48979592,  7.57142857,
 -58.34693878, -53.26530612, -27.18367347, -10.10204082,
 79.97959184, 190.06122449, 193.14285714, -16.7755102 ,
 192.30612245, 211.3877551 , 156.46938776, 251.55102041,
 225.63265306,  69.71428571,  -6.20408163, 115.87755102,
 135.95918367,  63.04081633,  41.12244898, 144.20408163,
 133.28571429,  47.36734694, 241.44897959, 149.53061224,
 241.6122449 , 285.69387755, 187.7755102 , 295.85714286,
 205.93877551, 189.02040816, 178.10204082, 177.18367347,
 191.26530612, 207.34693878, 291.42857143, 160.51020408,
 261.59183673, 256.67346939, 284.75510204, 187.83673469,
 221.91836735, 172.      ])
```

```
plt.scatter(x,y)
```

```
<matplotlib.collections.PathCollection at 0x1cb26143b90>
```



```
# plt.scatter on pandas dataframe
```

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

	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]
```

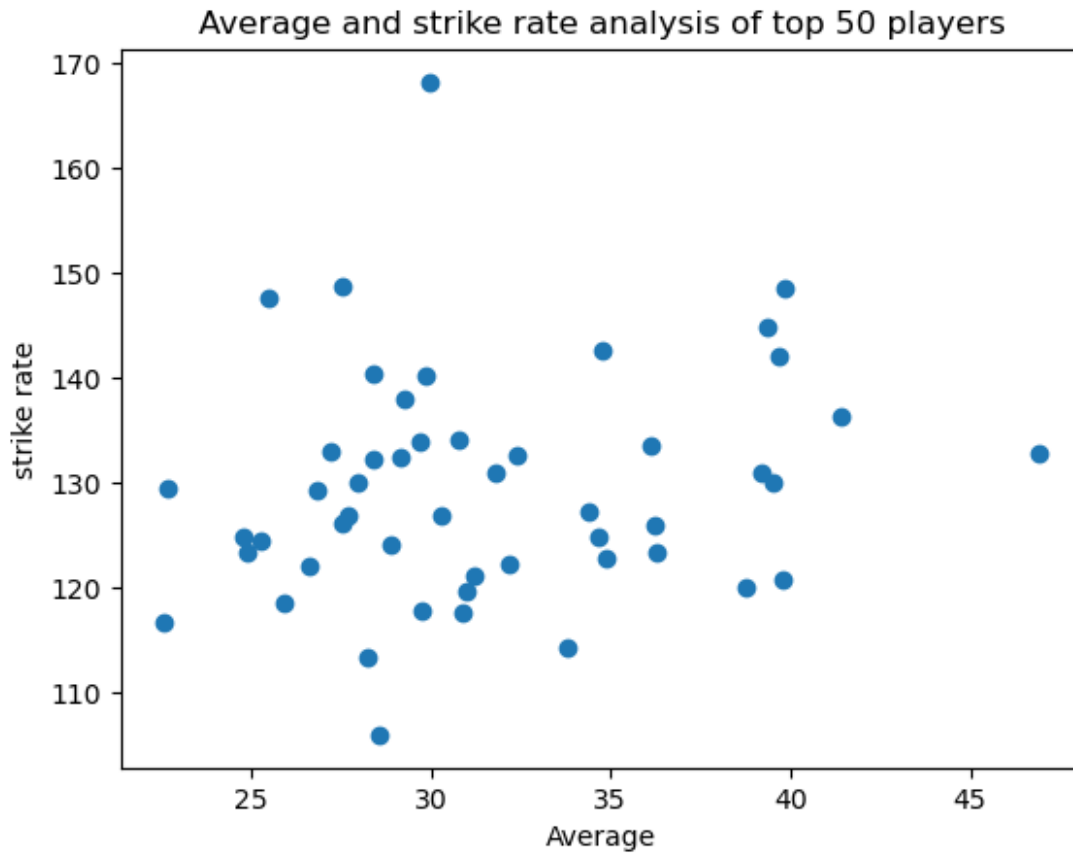
```
# our task ki as a ipl team honour avg and strike rate ke basis pe  
kiss player ko kharidna chahaiye aue kisse nahi
```

```
df=df.head(50)  
df
```

	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
5	AB de Villiers	5181	39.853846	148.580442
6	CH Gayle	4997	39.658730	142.121729
7	MS Dhoni	4978	39.196850	130.931089
8	RV Uthappa	4954	27.522222	126.152279
9	KD Karthik	4377	26.852761	129.267572
10	G Gambhir	4217	31.007353	119.665153
11	AT Rayudu	4190	28.896552	124.148148
12	AM Rahane	4074	30.863636	117.575758
13	KL Rahul	3895	46.927711	132.799182
14	SR Watson	3880	30.793651	134.163209
15	MK Pandey	3657	29.731707	117.739858
16	SV Samson	3526	29.140496	132.407060
17	KA Pollard	3437	28.404959	140.457703
18	F du Plessis	3403	34.373737	127.167414
19	YK Pathan	3222	29.290909	138.046272
20	BB McCullum	2882	27.711538	126.848592
21	RR Pant	2851	34.768293	142.550000
22	PA Patel	2848	22.603175	116.625717
23	JC Buttler	2832	39.333333	144.859335

24	SS Iyer	2780	31.235955	121.132898
25	Q de Kock	2767	31.804598	130.951254
26	Yuvraj Singh	2754	24.810811	124.784776
27	V Sehwag	2728	27.555556	148.827059
28	SA Yadav	2644	29.707865	134.009123
29	M Vijay	2619	25.930693	118.614130
30	RA Jadeja	2502	26.617021	122.108346
31	SPD Smith	2495	34.652778	124.812406
32	SE Marsh	2489	39.507937	130.109775
33	DA Miller	2455	36.102941	133.569097
34	JH Kallis	2427	28.552941	105.936272
35	WP Saha	2427	25.281250	124.397745
36	DR Smith	2385	28.392857	132.279534
37	MA Agarwal	2335	22.669903	129.506378
38	SR Tendulkar	2334	33.826087	114.187867
39	GJ Maxwell	2320	25.494505	147.676639
40	N Rana	2181	27.961538	130.053667
41	R Dravid	2174	28.233766	113.347237
42	KS Williamson	2105	36.293103	123.315759
43	AJ Finch	2092	24.904762	123.349057
44	AC Gilchrist	2069	27.223684	133.054662
45	AD Russell	2039	29.985294	168.234323
46	JP Duminy	2029	39.784314	120.773810
47	MEK Hussey	1977	38.764706	119.963592
48	HH Pandya	1972	29.878788	140.256046
49	Shubman Gill	1900	32.203390	122.186495

```
plt.scatter(df['avg'],df['strike_rate'])
plt.title('Average and strike rate analysis of top 50 players')
plt.xlabel('Average')
plt.ylabel('strike rate')
Text(0, 0.5, 'strike rate')
```

```
plt.scatter(df['avg'],df['strike_rate'],color='red')
plt.title('Average and strike rate analysis of top 50 players')
plt.xlabel('Average')
plt.ylabel('strike rate')
Text(0, 0.5, 'strike rate')
```

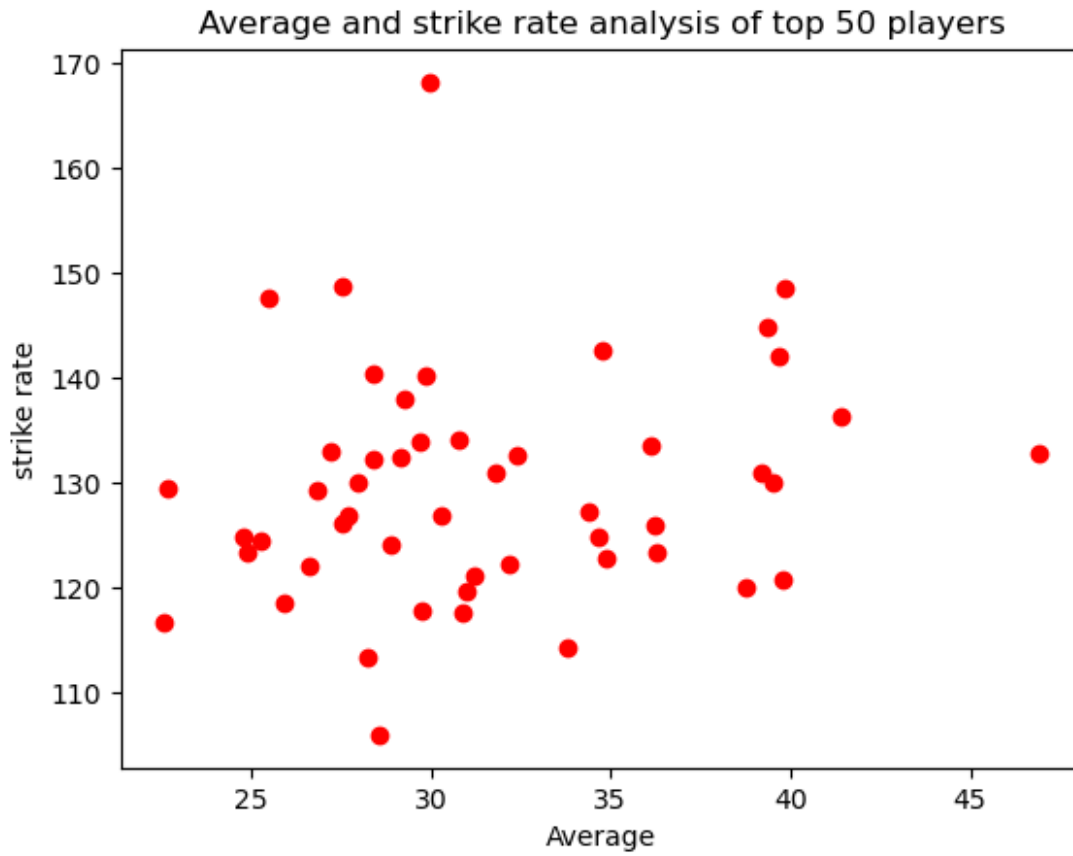


abb jo bhi upar sikhe ho 2D line graph me colour wagera wo sab yaha bhi apply kar sakte ho

marker

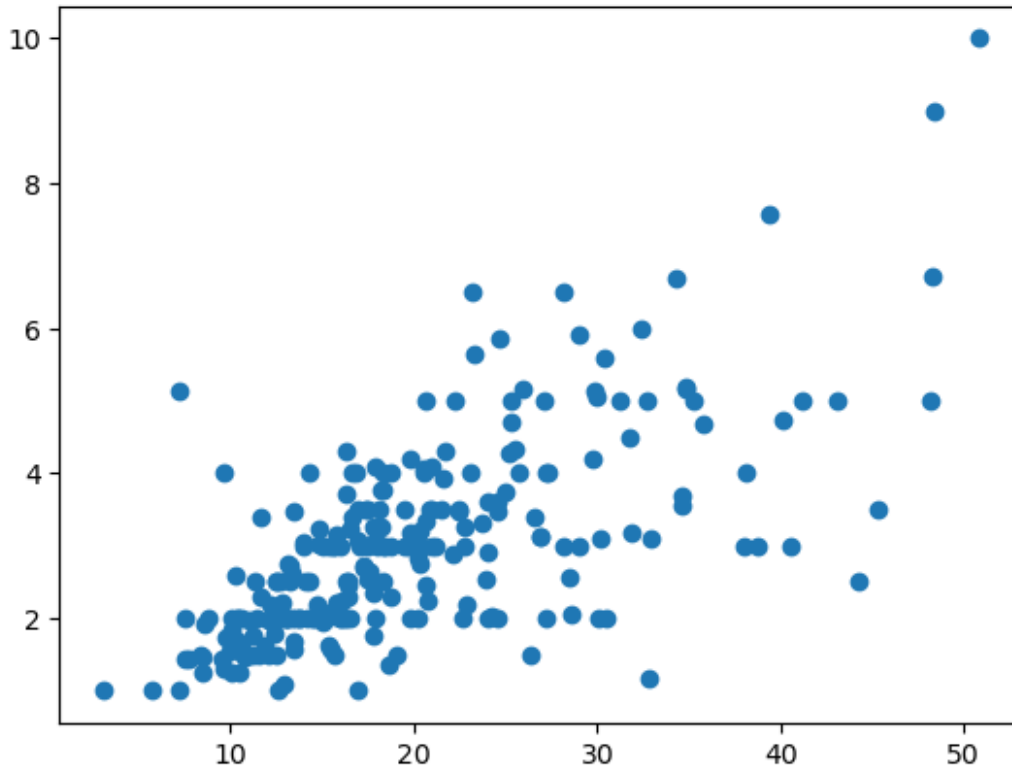
size

```
tips = sns.load_dataset('tips')
tips
```

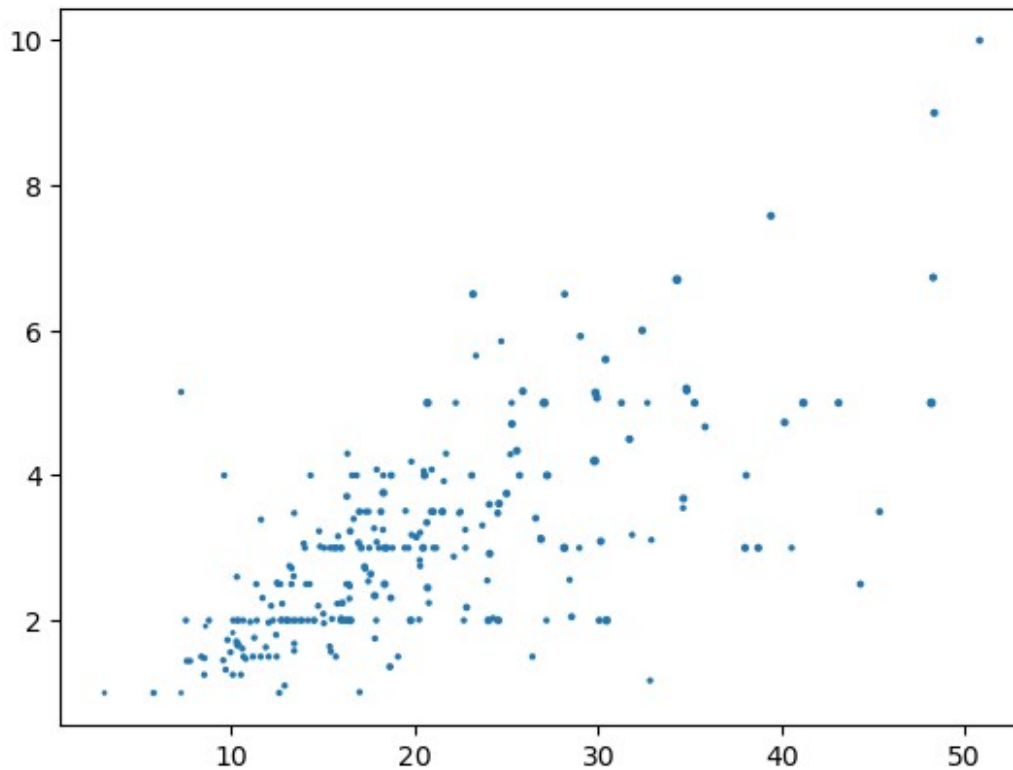
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67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

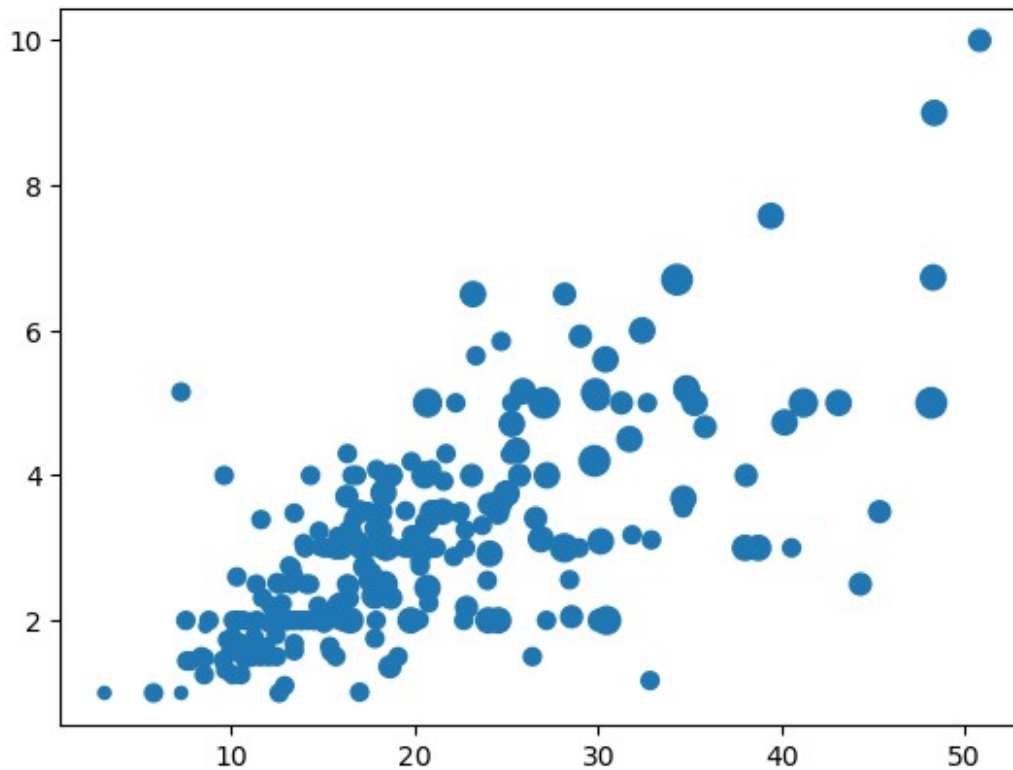
```
# just check ki total bill me and tip me kya relation hai
plt.scatter(tips['total_bill'],tips['tip'])
<matplotlib.collections.PathCollection at 0x1cb27c28830>
```



```
# yadi hum chahate hain ki group size ko bhi involve kare to simply ek
parameter aur pass kar sakte hain
plt.scatter(tips['total_bill'],tips['tip'],s=tips['size']) # to jiss
group me jyada member ya jiss group ka size bada hoga uska bubble bada
and jiska
# group
size chota hoga uska bubble chota dikhega abhi sab group me 2,3 log
hai so bubble
# sahi se
nahi dikh raha hai iss column ko 20 se multiply kar lete hain
<matplotlib.collections.PathCollection at 0x1cb279d1f40>
```

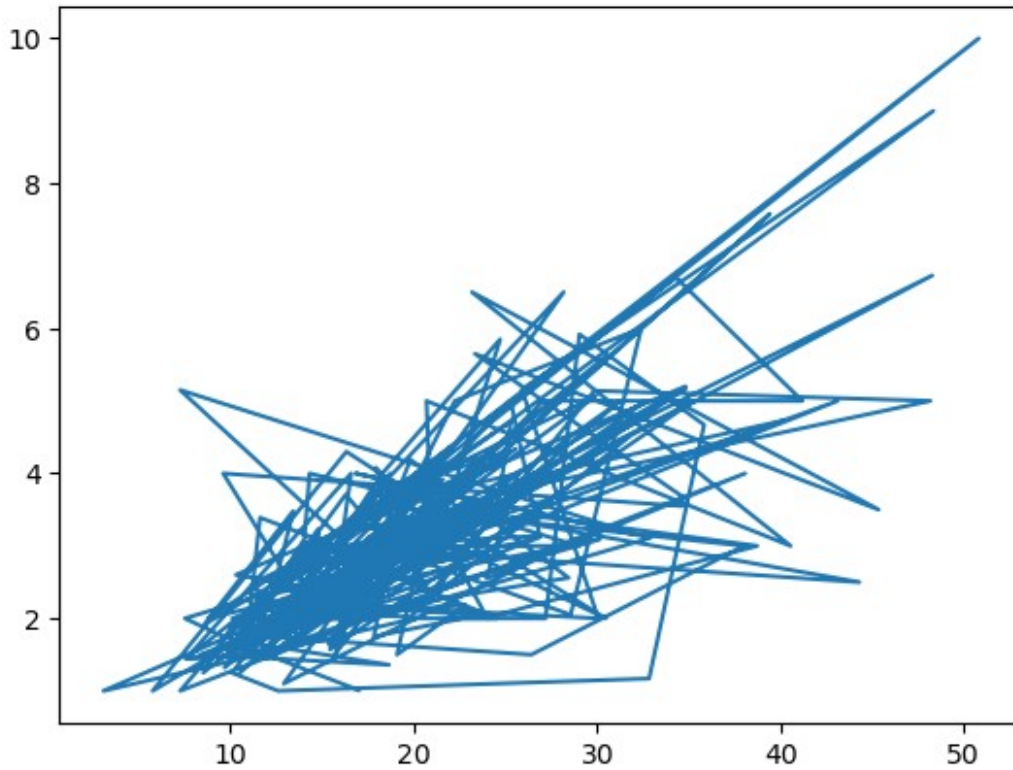


```
plt.scatter(tips['total_bill'],tips['tip'],s=tips['size']*20) # ye  
abb 3D me information provide karta hai ki total_bill kya tha , tip  
kitna dya  
# and team ka  
size kitna hai  
<matplotlib.collections.PathCollection at 0x1cb260b4470>
```

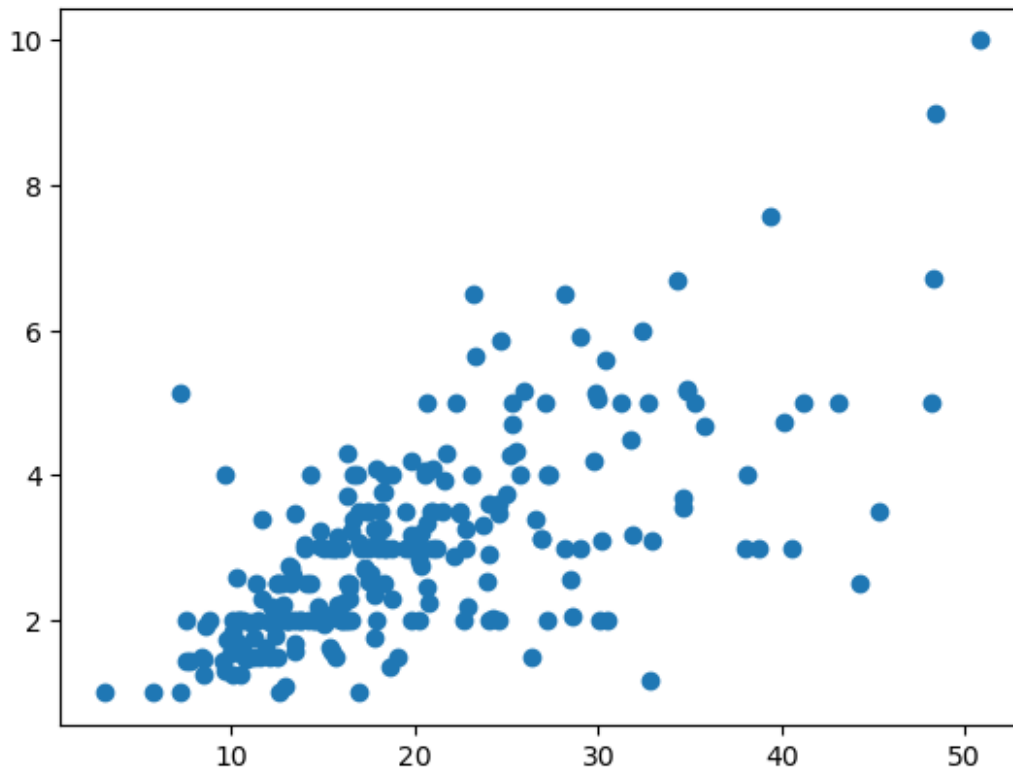


```
# scatterplot using plt.plot
plt.plot(tips['total_bill'],tips['tip']) # ye abhi 2D plot hai
simply sare point ko line se mila dya hai

[<matplotlib.lines.Line2D at 0x1cb2796b0b0>]
```



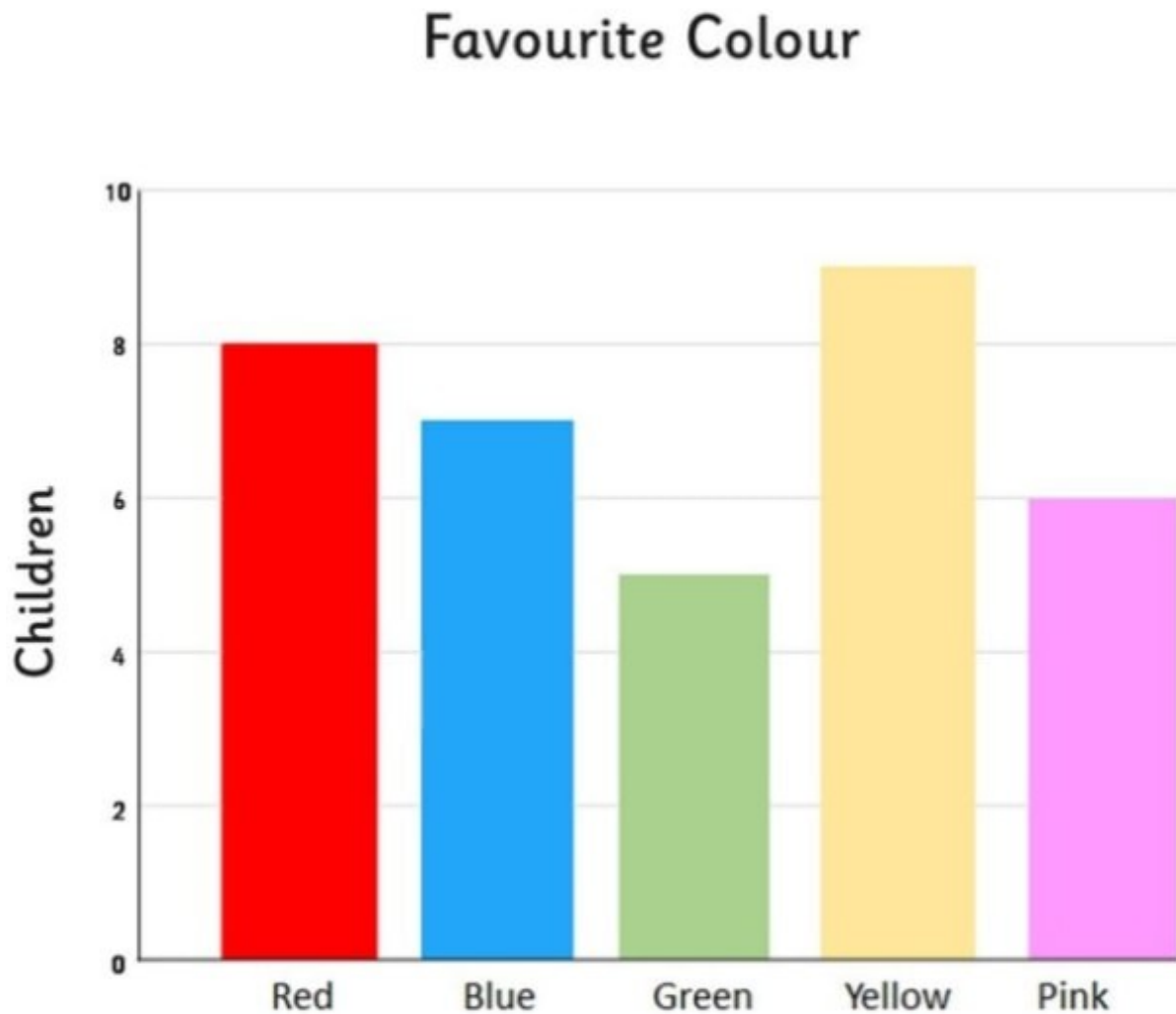
```
# passion addition parameter as 'o' , but plt,plot() se hum size bale  
ke sath kaam nahi kar sakte hai and color sab change karne me bhi  
problem hota hai  
plt.plot(tips['total_bill'],tips['tip'],'o')  
[<matplotlib.lines.Line2D at 0x1cb274ba690>]
```



plt.plot vs plt.scatter

plt.plot() --> faster hota hai & scatter function ke help se jo graph banate hain wo slower hota hai

Bar chart

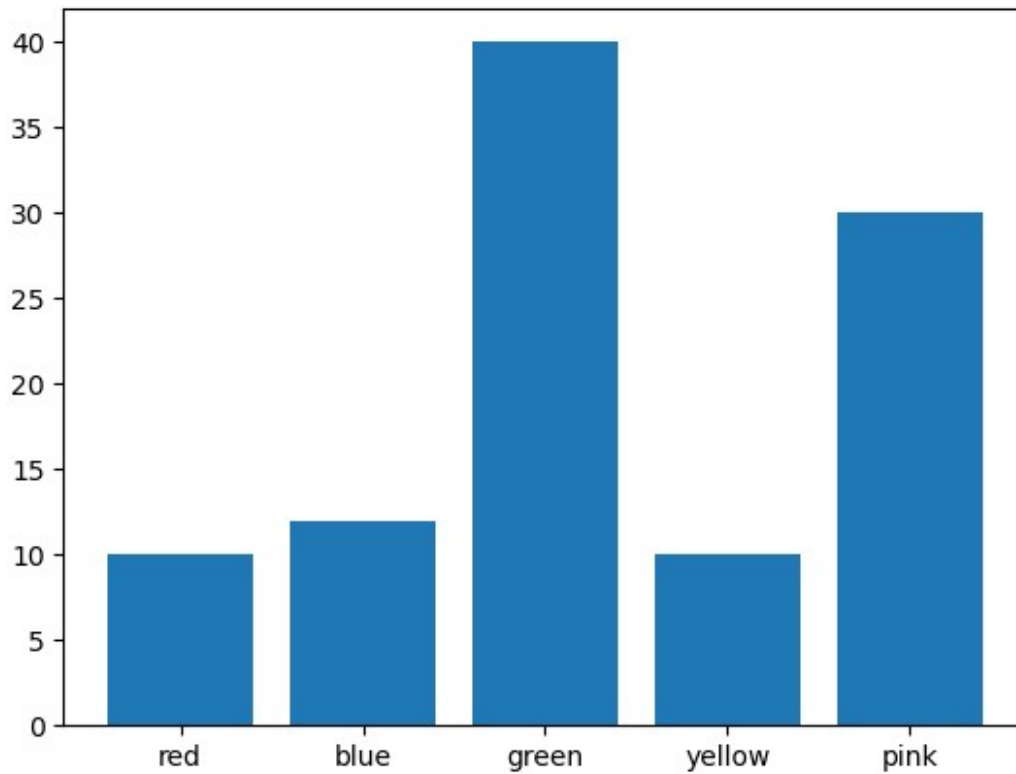


- Bivariate Analysis
- Numerical vs Categorical
- Use case - Aggregate analysis of groups

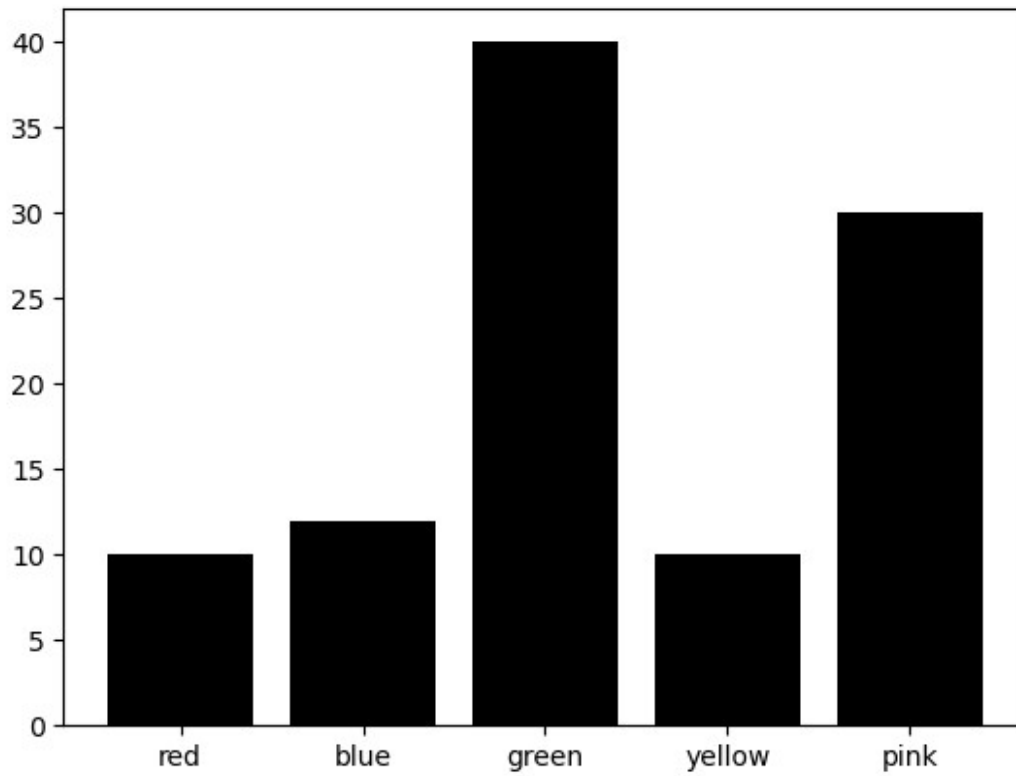
```
# simple bar chart
children = [10,12,40,10,30]
colors = ['red','blue','green','yellow','pink']

plt.bar(colors,children)

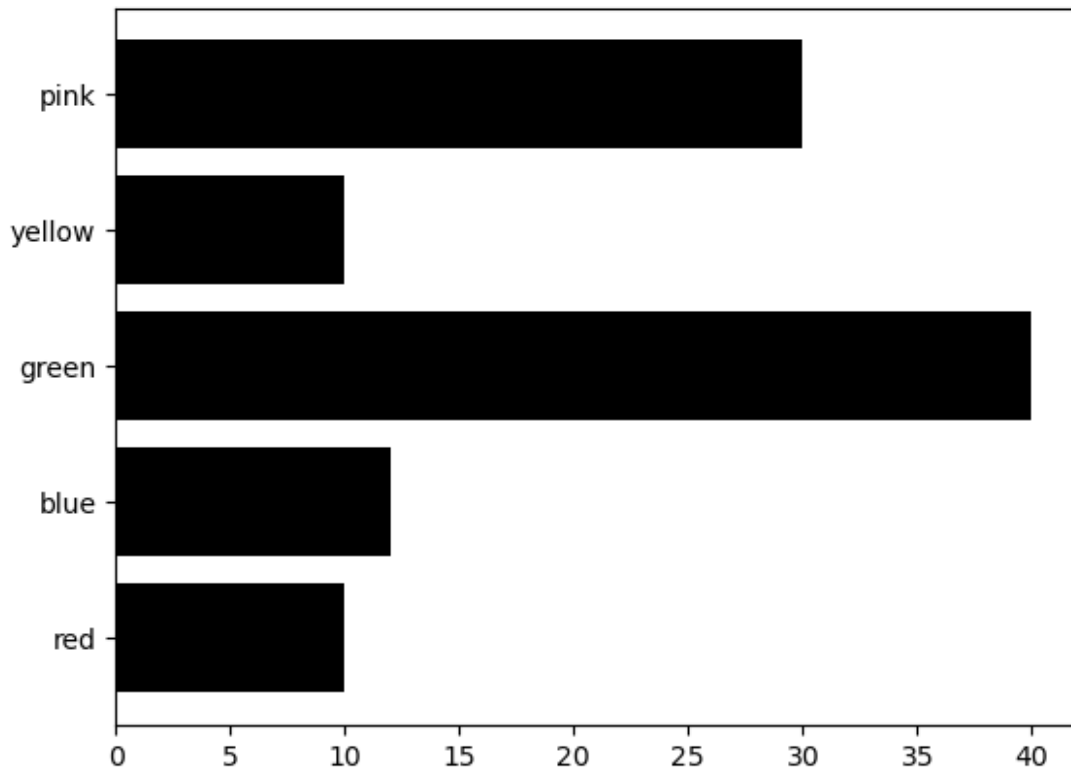
<BarContainer object of 5 artists>
```

```
plt.bar(colors,children,color='black') # sare changes jo upar kye  
hain wo yaha bhi applicable hai  
<BarContainer object of 5 artists>
```



```
# horizontal bar chart  
plt.barh(colors,children,color='black')  
<BarContainer object of 5 artists>
```



jab number of categories badh jaye tab hum horizontal bar graph banana prefer karte hain

colors and labels

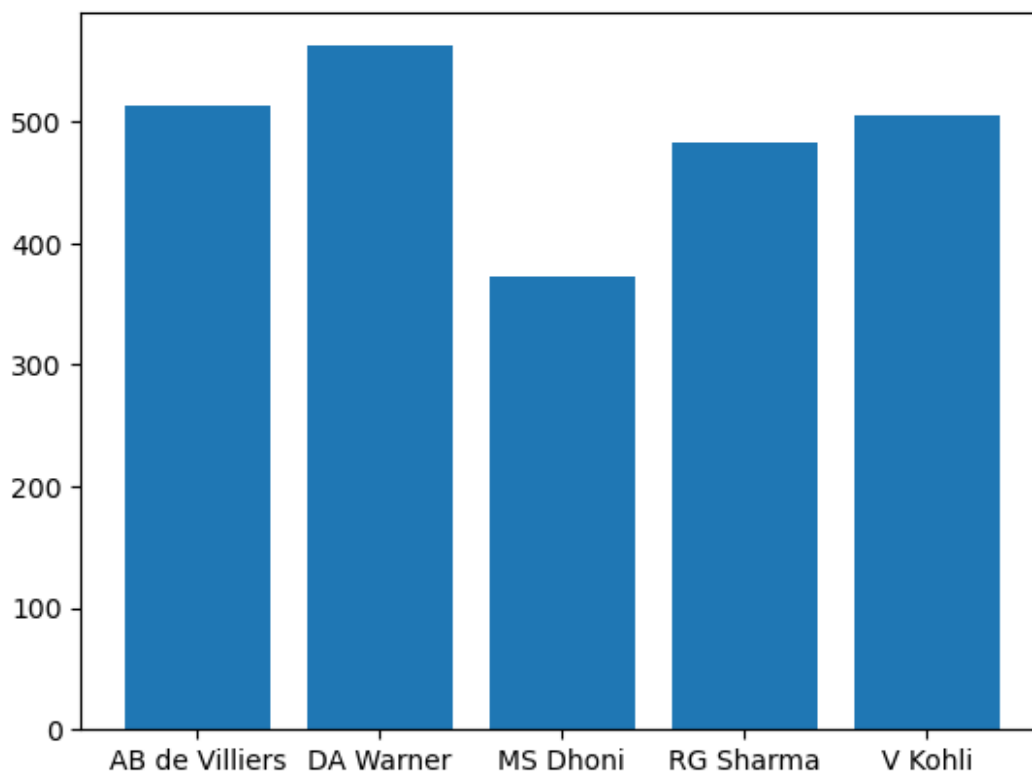
```
df=pd.read_csv('batsman_season_record.csv')
```

```
df
```

	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

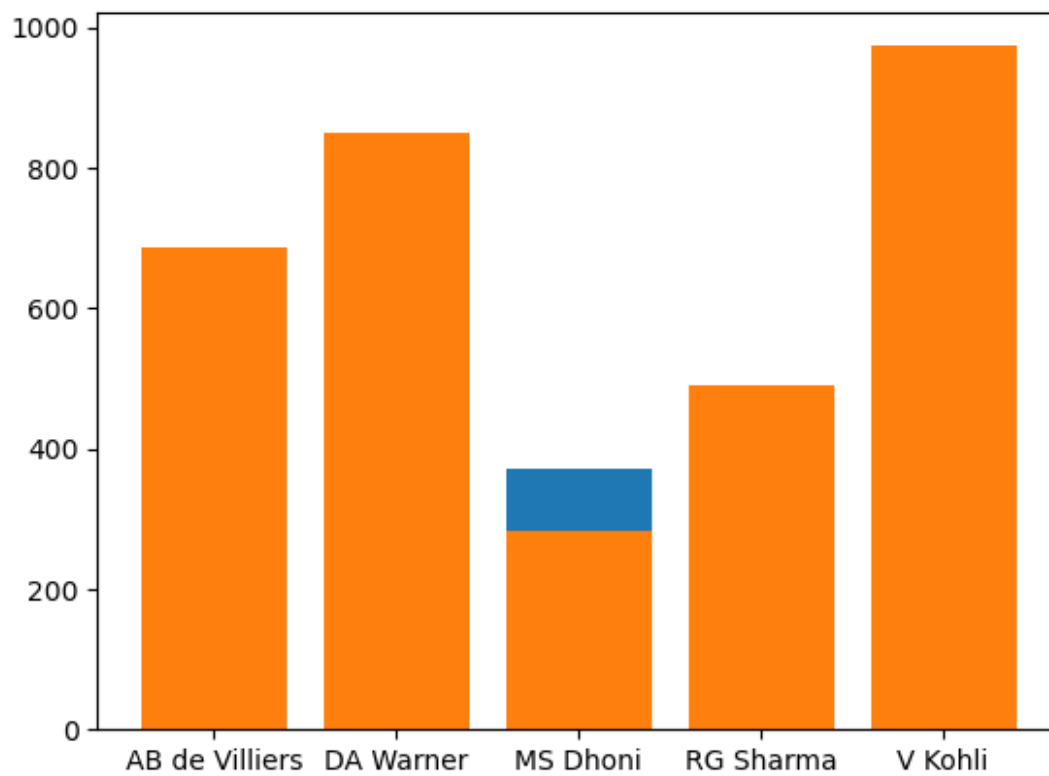
```
plt.bar(df['batsman'],df['2015'])
```

```
<BarContainer object of 5 artists>
```

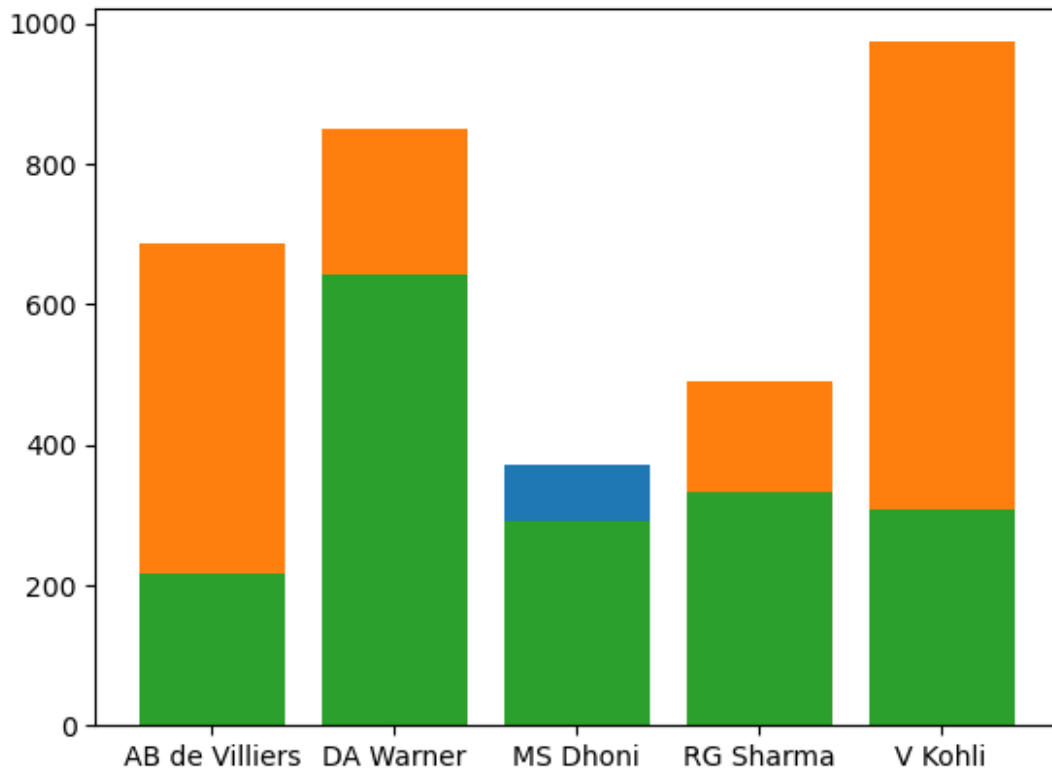


```
# abb suppose sath hi sath me hume 2016 ke bhi recordd dikhana hai yo  
abhi tak ke learning ke according ek ke baad ke likh do let's see  
plt.bar(df['batsman'],df['2015'])  
plt.bar(df['batsman'],df['2016'])    # nn kuch gadbad hai
```

```
<BarContainer object of 5 artists>
```

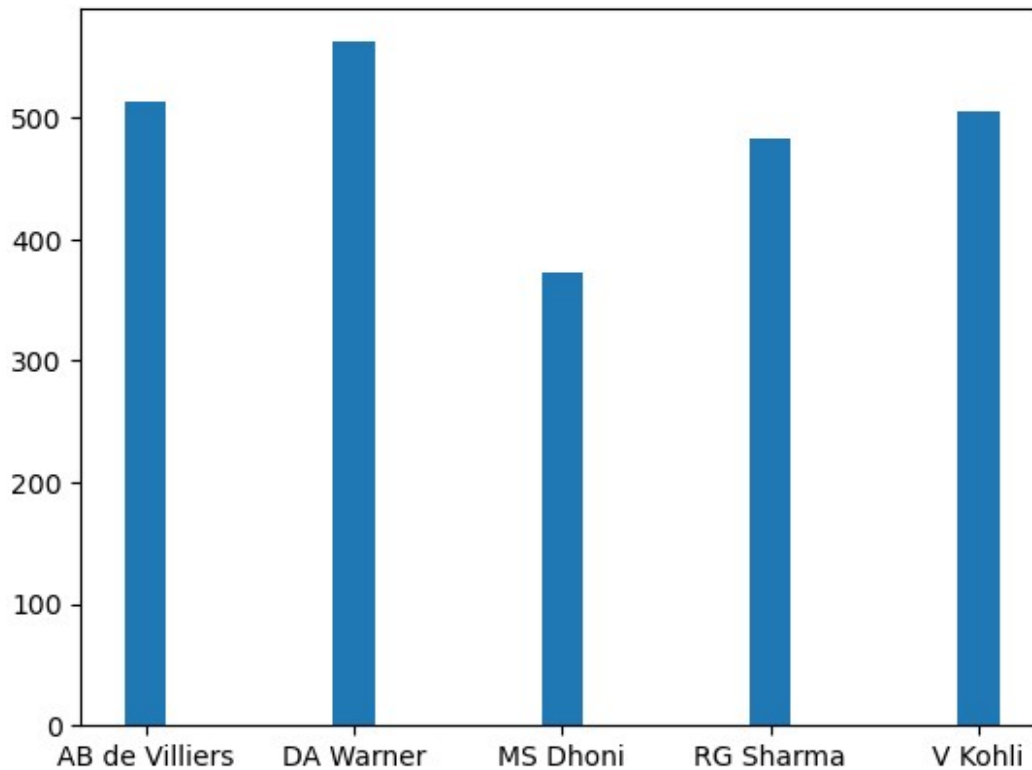


```
plt.bar(df['batsman'],df['2015'])  
plt.bar(df['batsman'],df['2016'])  
plt.bar(df['batsman'],df['2017'])  
<BarContainer object of 5 artists>
```



```
## hum sare graph ko side by side dikhana chahate hain  
plt.bar(df['batsman'],df['2015'],width=0.2) # hum bar ka width handle  
kar sakte hain
```

```
<BarContainer object of 5 artists>
```



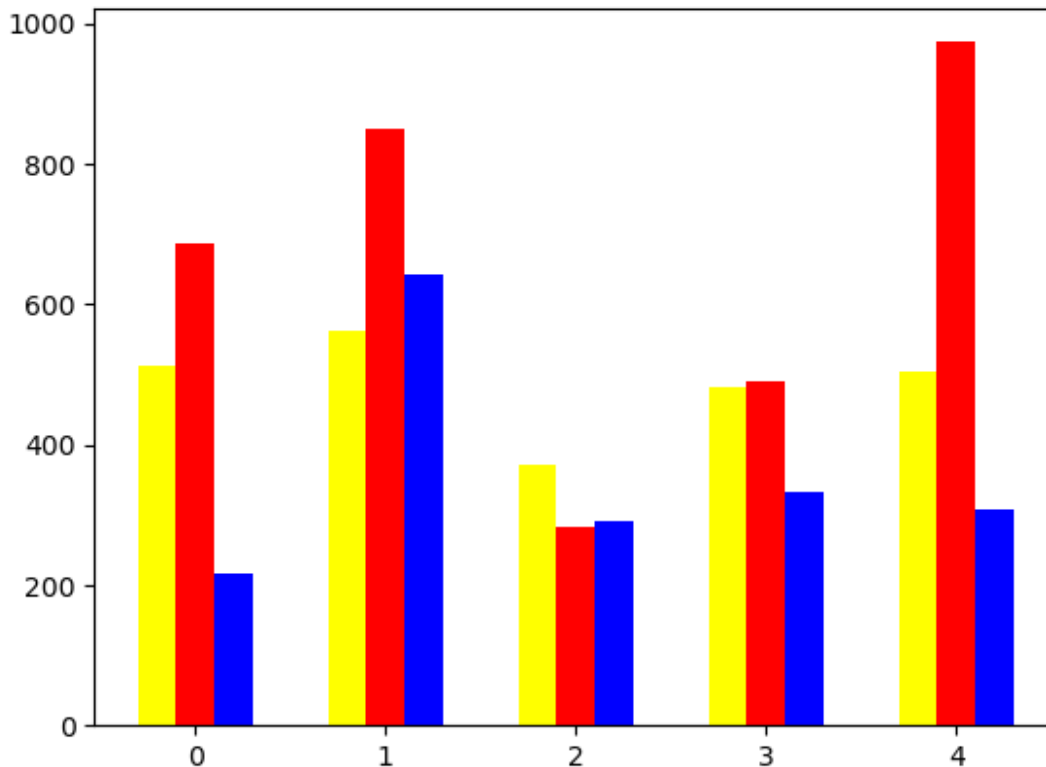
```
np.arange(df.shape[0])
```

```
array([0, 1, 2, 3, 4])
```

```
# basically humara approach rahega ki hum 2015 bale graph ko thoda  
side hata denge
```

```
plt.bar(np.arange(df.shape[0]) -  
0.2,df['2015'],width=0.2,color='yellow')  
plt.bar(np.arange(df.shape[0]),df['2016'],width=0.2,color='red')  
plt.bar(np.arange(df.shape[0]) +  
0.2,df['2017'],width=0.2,color='blue')
```

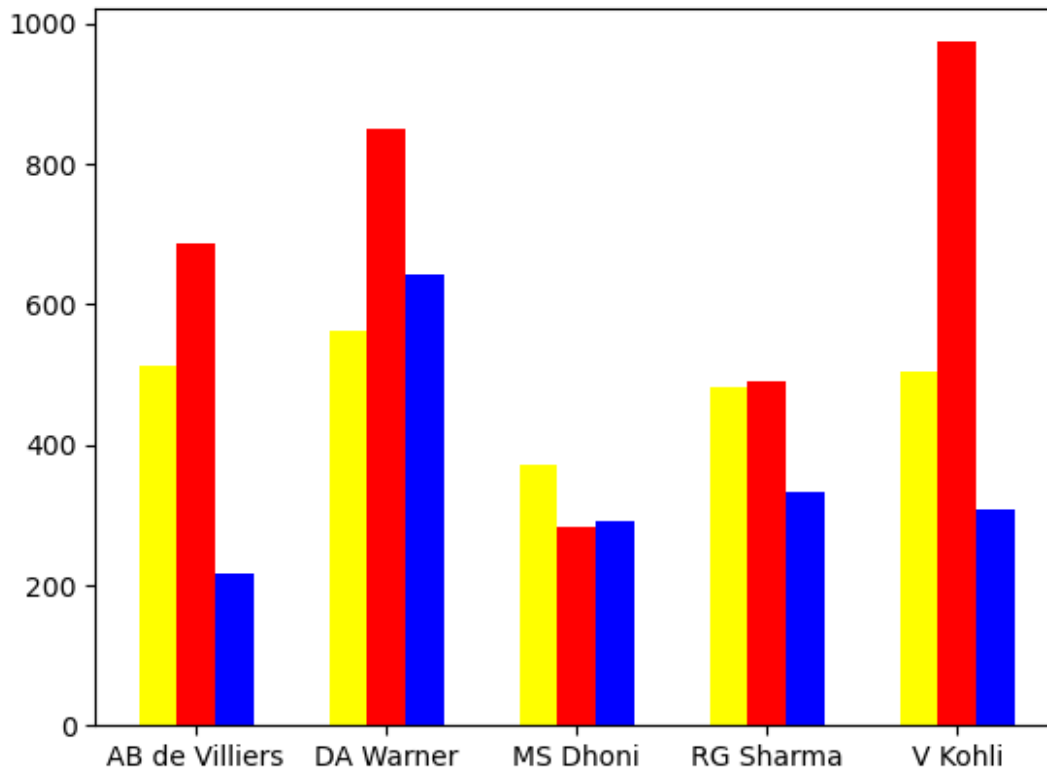
```
<BarContainer object of 5 artists>
```



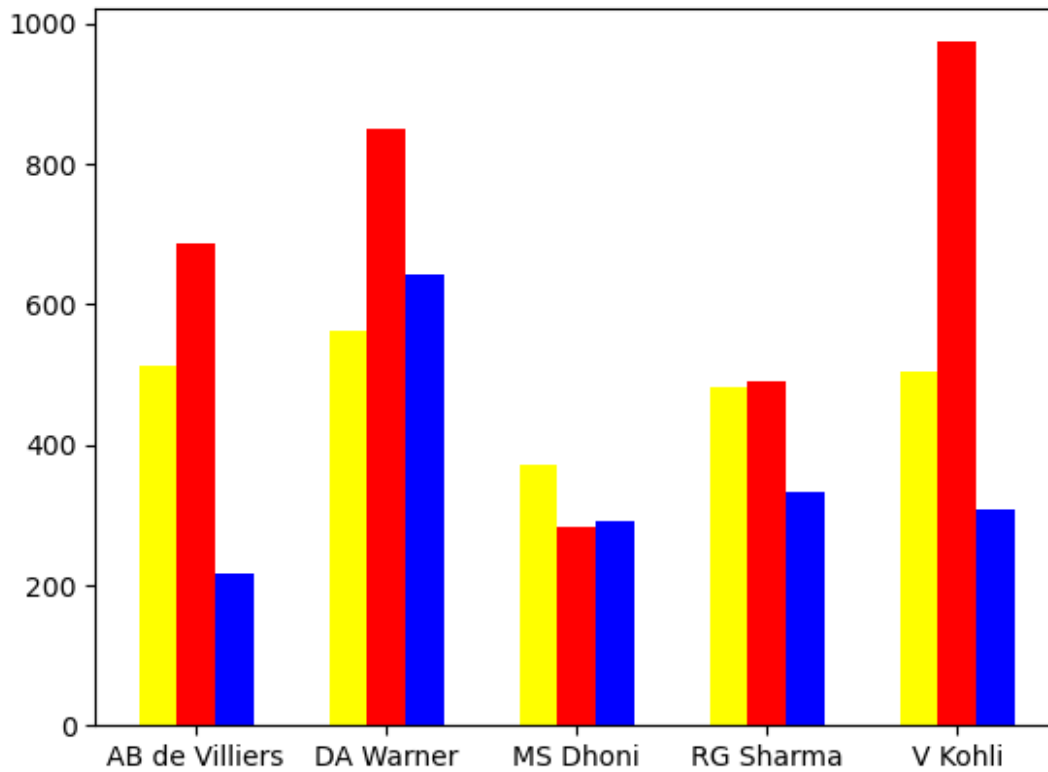
```
plt.bar(np.arange(df.shape[0]) -
0.2,df['2015'],width=0.2,color='yellow')
plt.bar(np.arange(df.shape[0]),df['2016'],width=0.2,color='red')
plt.bar(np.arange(df.shape[0]) +
0.2,df['2017'],width=0.2,color='blue')

plt.xticks(np.arange(df.shape[0]), df['batsman']) # x axis pe name
add kar dega

([<matplotlib.axis.XTick at 0x20be6703770>,
 <matplotlib.axis.XTick at 0x20be6708bc0>,
 <matplotlib.axis.XTick at 0x20be55a5580>,
 <matplotlib.axis.XTick at 0x20be4dec380>,
 <matplotlib.axis.XTick at 0x20be6702090>],
 [Text(0, 0, 'AB de Villiers'),
 Text(1, 0, 'DA Warner'),
 Text(2, 0, 'MS Dhoni'),
 Text(3, 0, 'RG Sharma'),
 Text(4, 0, 'V Kohli')])
```

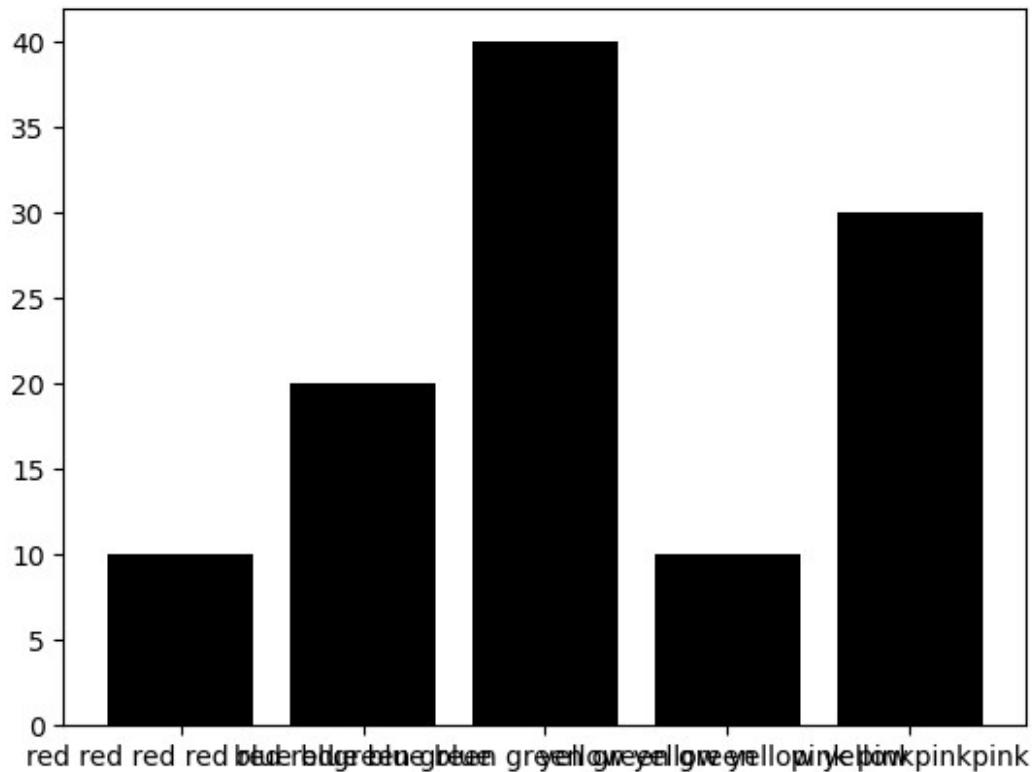
```
plt.bar(np.arange(df.shape[0]) -  
0.2,df['2015'],width=0.2,color='yellow')  
plt.bar(np.arange(df.shape[0]),df['2016'],width=0.2,color='red')  
plt.bar(np.arange(df.shape[0]) +  
0.2,df['2017'],width=0.2,color='blue')  
  
plt.xticks(np.arange(df.shape[0]), df['batsman']) #  
plt.show()
```



```
# a problem
children = [10,20,40,10,30]
colors = ['red red red red red red','blue blue blue blue','green green
green green green','yellow yellow yellow yellow ','pink pinkpinkpink']

plt.bar(colors,children,color='black')

# yaha category ka name bahut bada jiske karan overlap ho jaa raha hai
<BarContainer object of 5 artists>
```



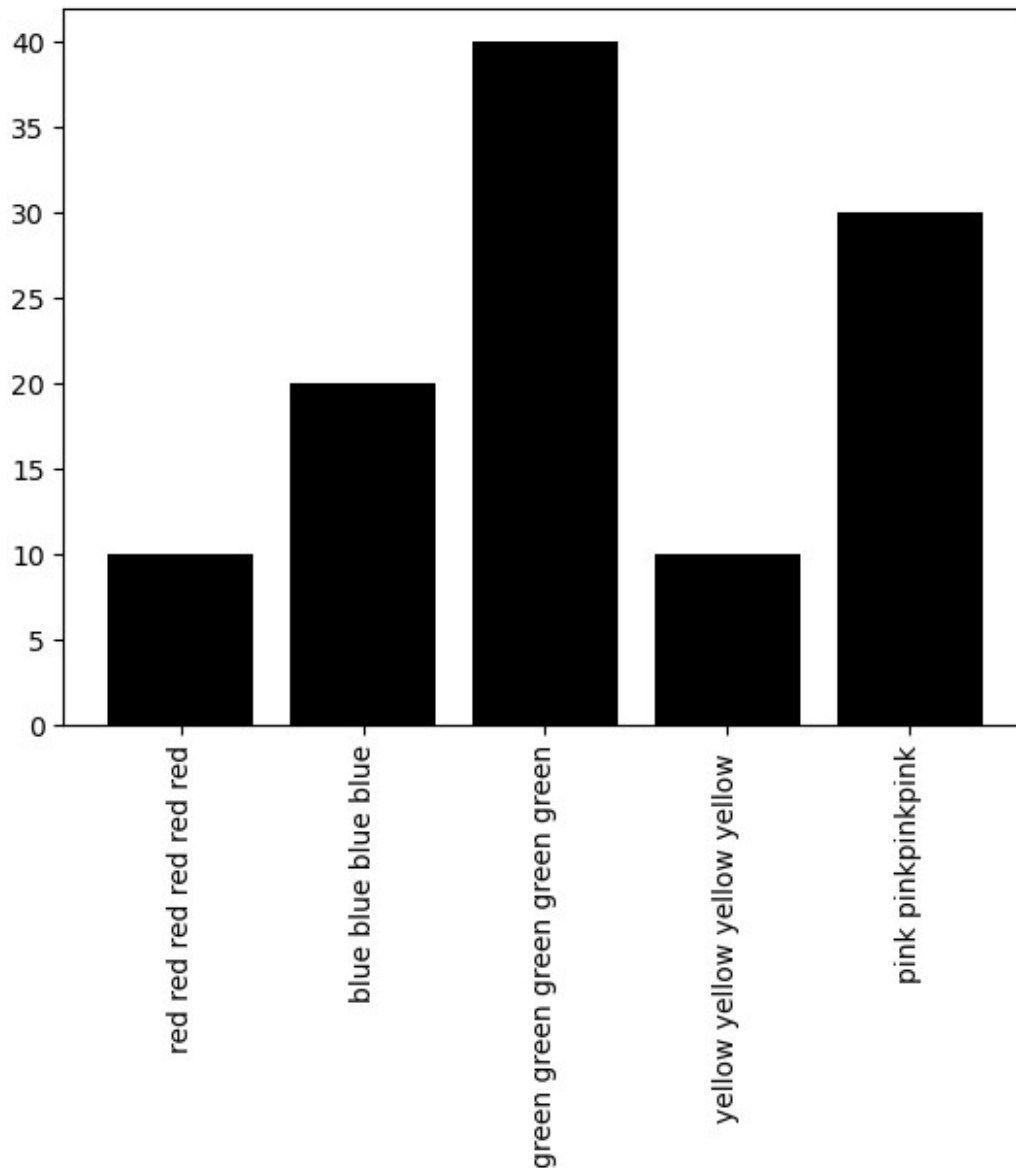
```

children = [10,20,40,10,30]
colors = ['red red red red red red red','blue blue blue blue','green green
green green green','yellow yellow yellow yellow ','pink pinkpinkpink']

plt.bar(colors,children,color='black')
plt.xticks(rotation='vertical') # ye name ko vertically print kar
dega and readability inhance ho jayegi

([0, 1, 2, 3, 4],
 [Text(0, 0, 'red red red red red red red'),
  Text(1, 0, 'blue blue blue blue'),
  Text(2, 0, 'green green green green green'),
  Text(3, 0, 'yellow yellow yellow yellow '),
  Text(4, 0, 'pink pinkpinkpink')])

```



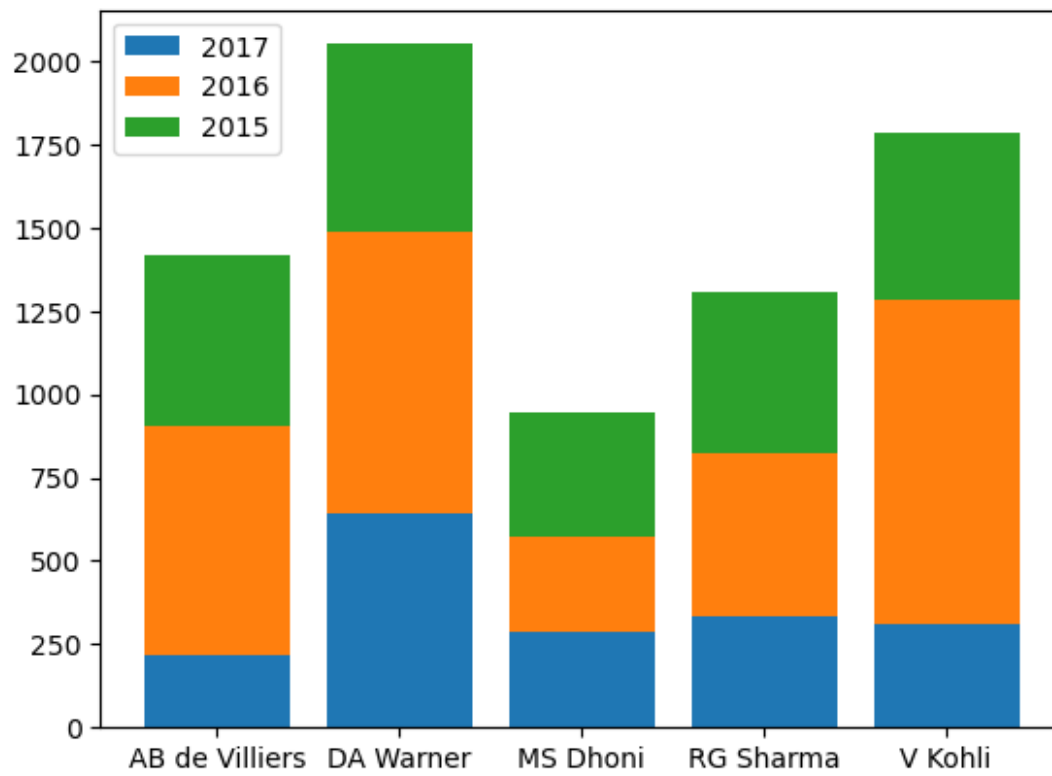
stacked bar chart

df

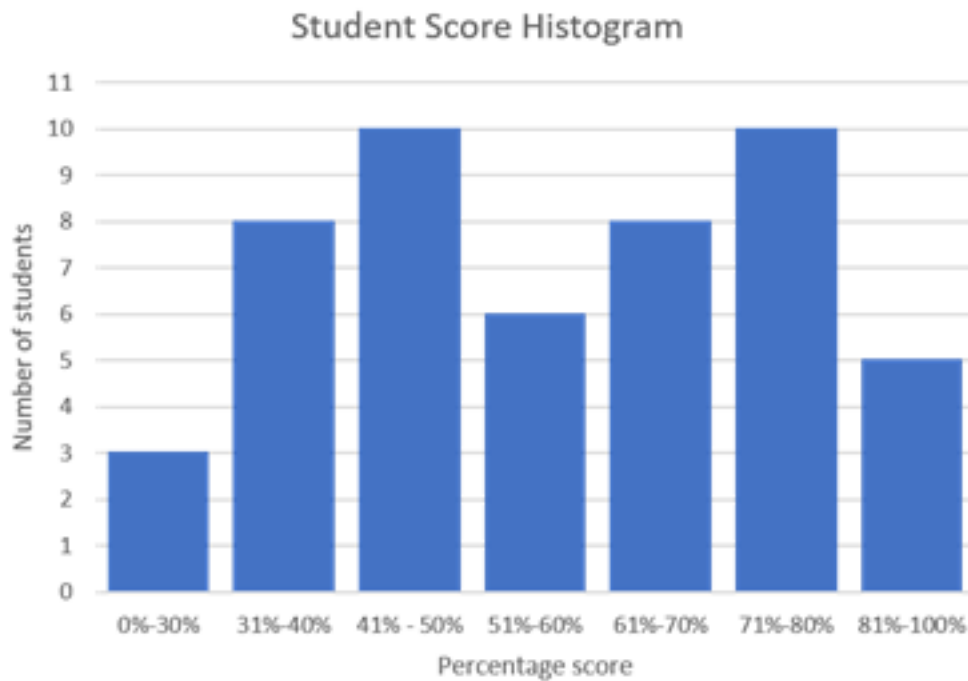
	batsman	2015	2016	2017
0	AB de Villiers	513	687	216
1	DA Warner	562	848	641
2	MS Dhoni	372	284	290
3	RG Sharma	482	489	333
4	V Kohli	505	973	308

```
plt.bar(df['batsman'],df['2017'],label='2017')
plt.bar(df['batsman'],df['2016'],bottom=df['2017'],label='2016')
plt.bar(df['batsman'],df['2015'],bottom=(df['2017']
+df['2016']),label='2015')
```

```
plt.legend()  
plt.show()
```



Histogram



- Univariate Analysis
- Numerical col
- Use case - Frequency Count

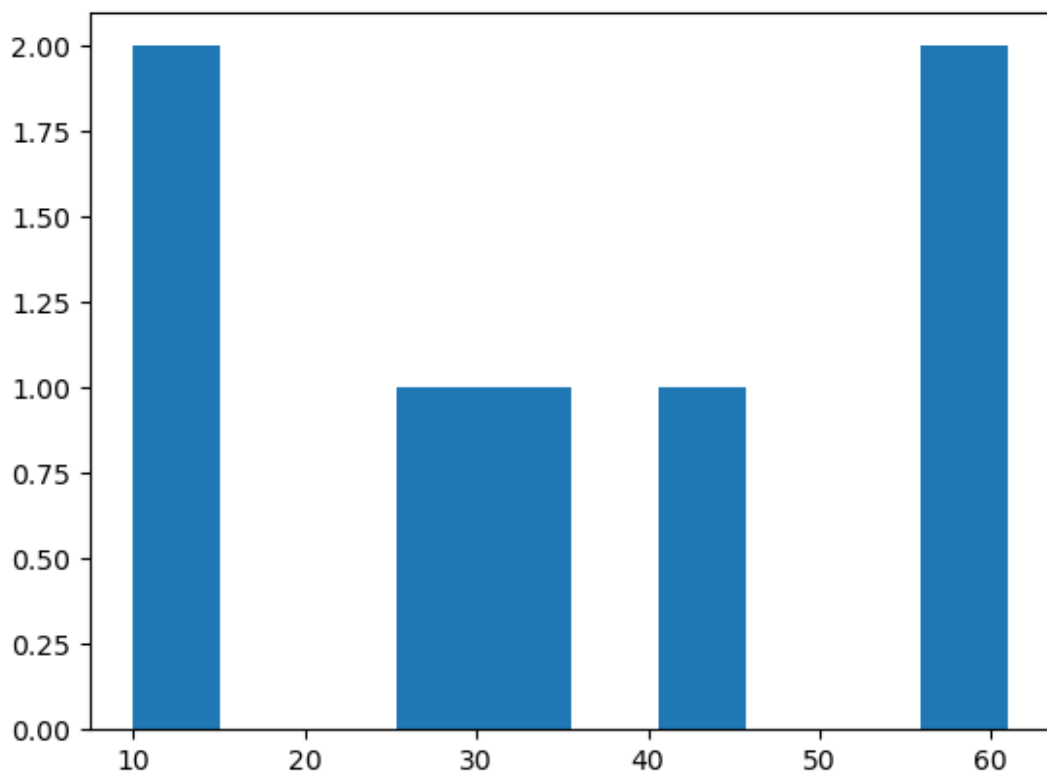
```
# range me jab graph banana hota hai
```

```
# simple data
```

```
data=[32,45,56,10,15,27,61]
```

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

```
(array([2., 0., 0., 1., 1., 0., 1., 0., 0., 2.]),  
 array([10. , 15.1, 20.2, 25.3, 30.4, 35.5, 40.6, 45.7, 50.8, 55.9,  
 61. ]),  
 <BarContainer object of 10 artists>)
```

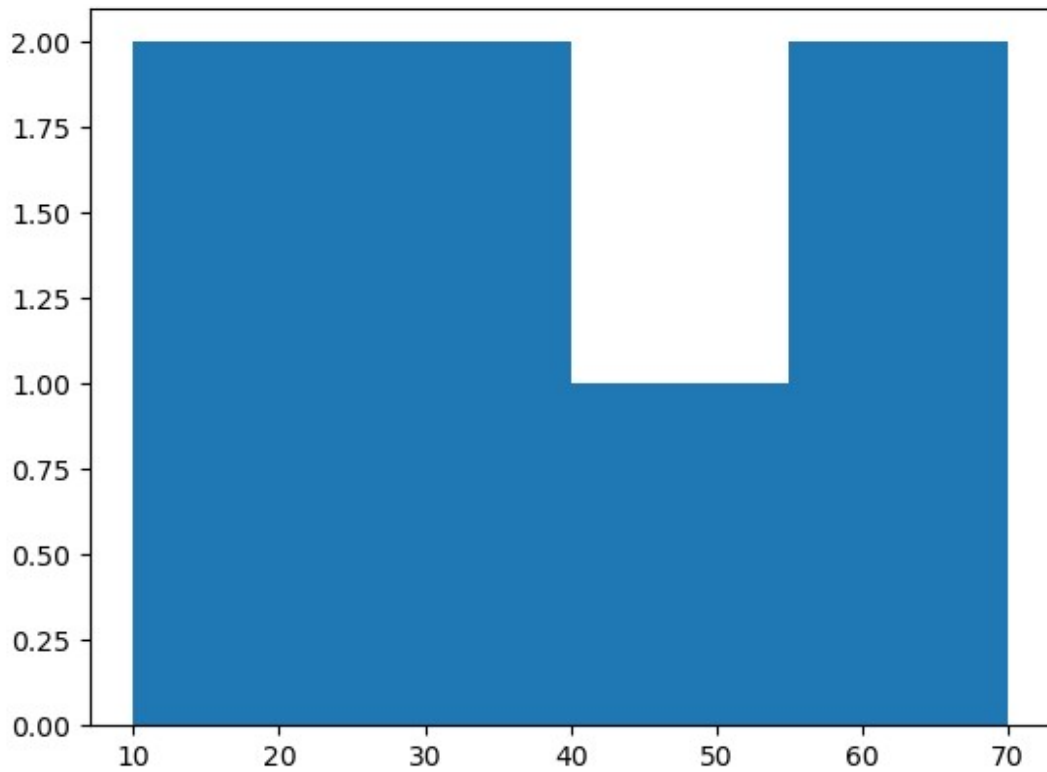


yaha bin sizee apne aap decide ho jaa raha hai aap khud se bhi kar sakte ho

```
data=[32,45,56,10,15,27,61]
```

```
plt.hist(data,bins=[10,25,40,55,70])
```

```
(array([2., 2., 1., 2.]),  
 array([10., 25., 40., 55., 70.]),  
 <BarContainer object of 4 artists>)
```



on some data

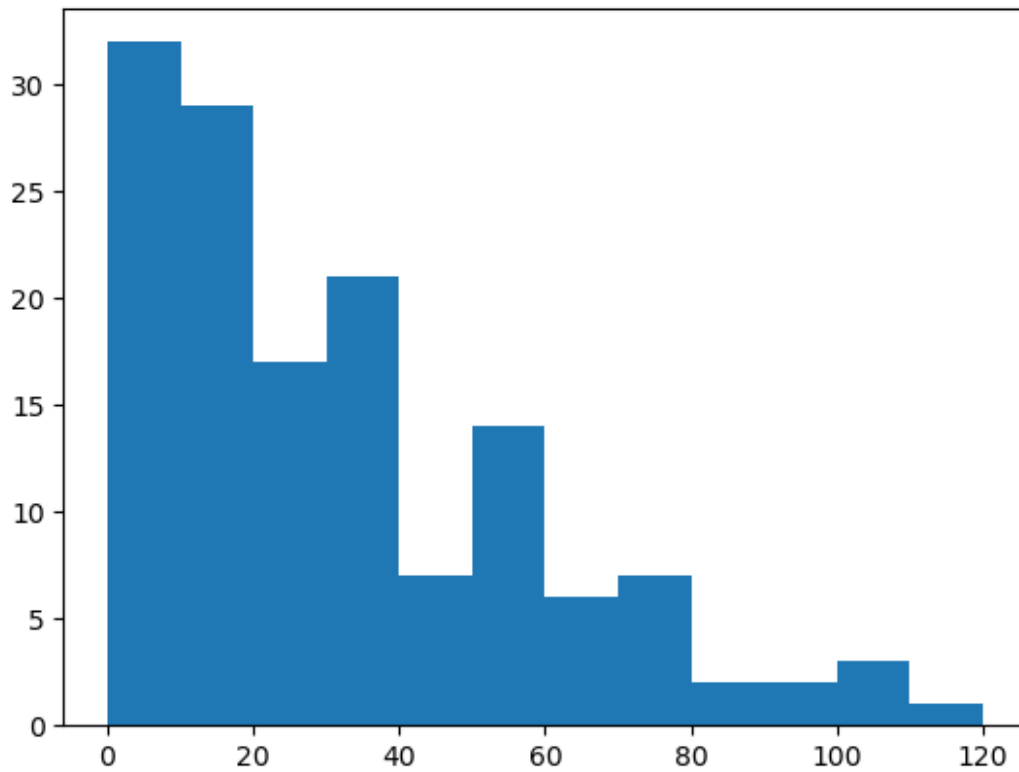
```
df=pd.read_csv('vk.csv')
```

```
df
```

	match_id	batsman_runs
0	12	62
1	17	28
2	20	64
3	27	0
4	30	10
...
136	624	75
137	626	113
138	632	54
139	633	0
140	636	54

```
[141 rows x 2 columns]
```

```
plt.hist(df['batsman_runs'],bins=[0,10,20,30,40,50,60,70,80,90,100,110,120])  
plt.show()
```

```
# logarithmic scale
```

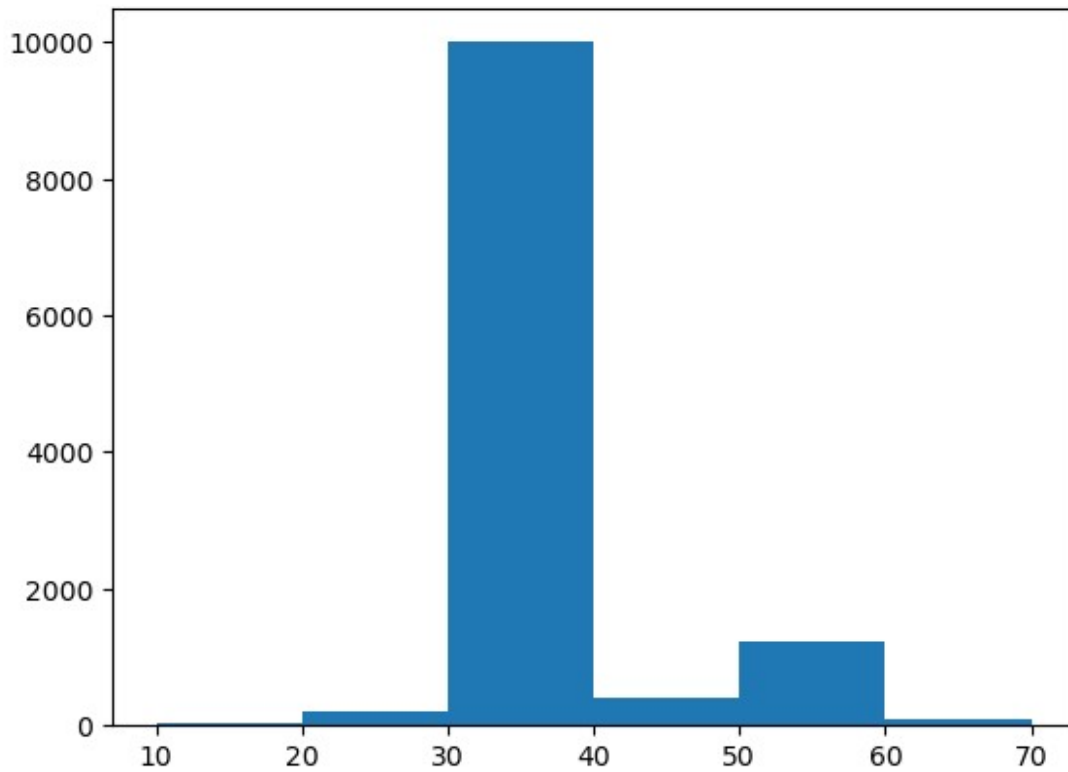
```
arr = np.load('big-array.npy')  
arr
```

```
array([33, 39, 37, ..., 33, 30, 39], dtype=int64)
```

```
arr.shape
```

```
(11949,)
```

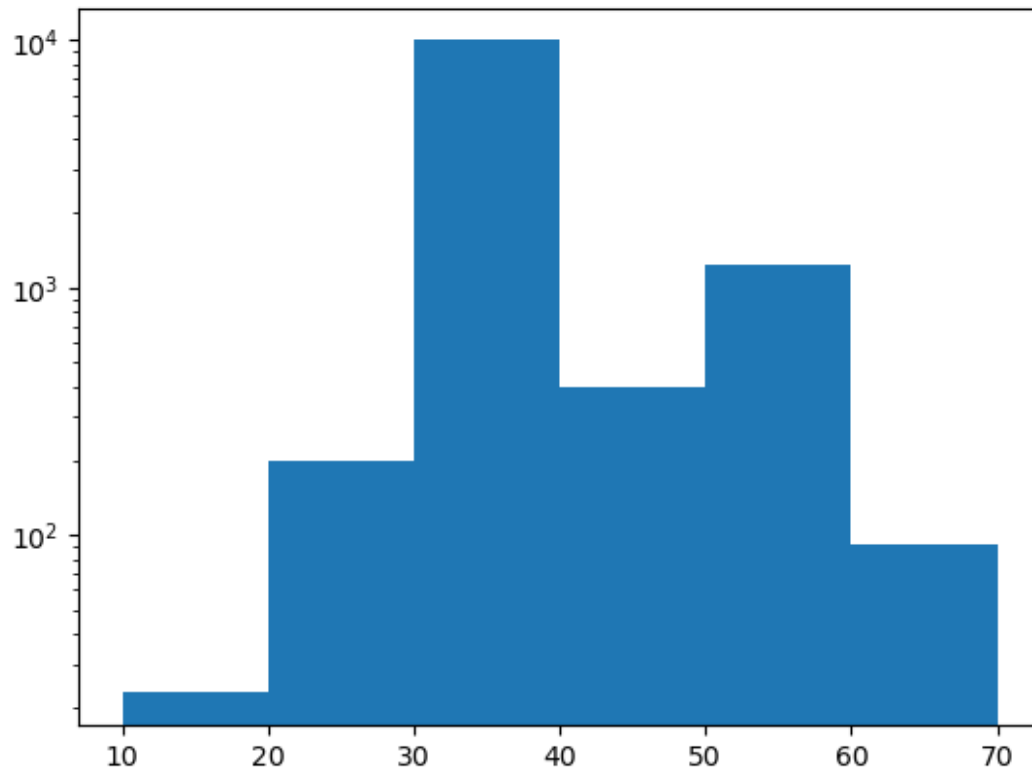
```
plt.hist(arr, bins=[10, 20, 30, 40, 50, 60, 70])  
plt.show()
```



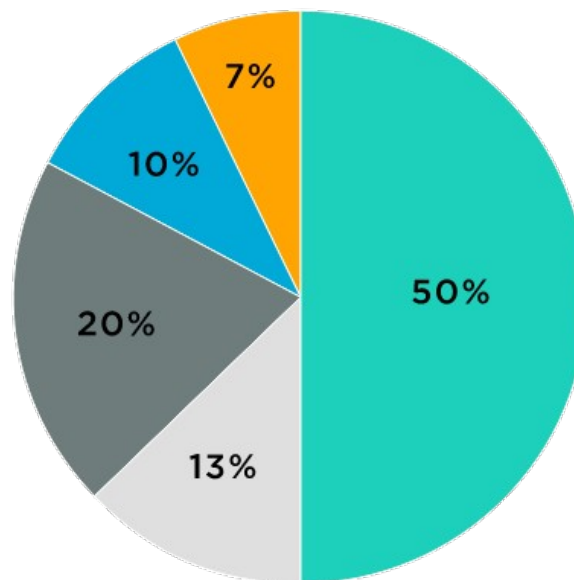
iss graph me ye problem ho raha hai ki 30-40 ke bich me itna data hai ki other data sahi se dikh hi nahi paa raha hai

toh iss scenario me hum logirathemic scale ka use karte hain

```
plt.hist(arr,bins=[10,20,30,40,50,60,70],log=True)  
plt.show()
```



Pie Chart

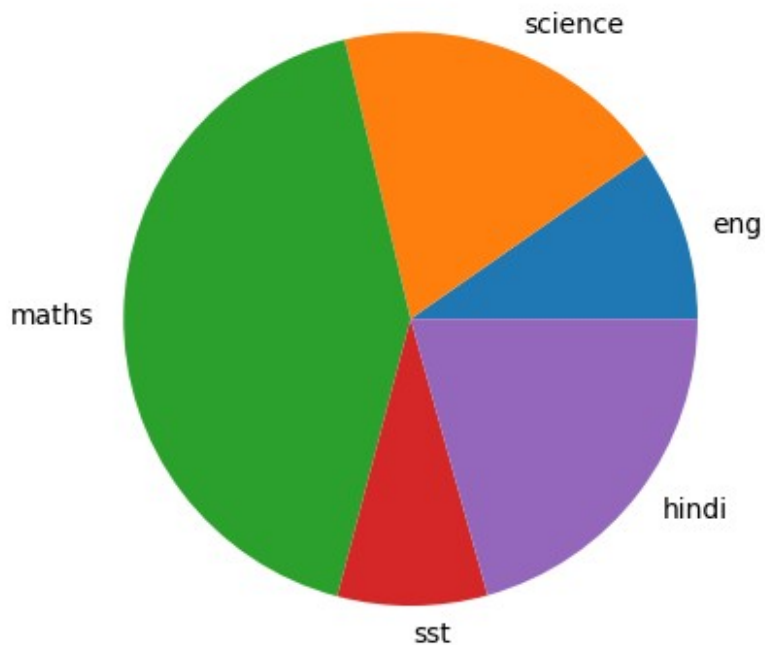


- Univariate/Bivariate Analysis
- Categorical vs numerical

- Use case - To find contribution on a standard scale

```
# simple data
data = [23,45,100,20,49]
subjects = ['eng','science','maths','sst','hindi']
plt.pie(data,labels=subjects)

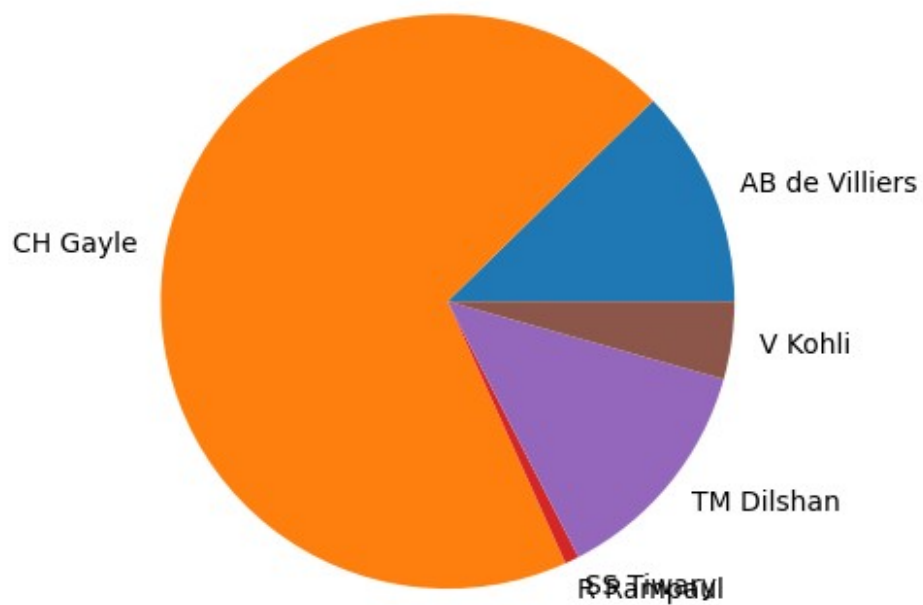
plt.show()
```



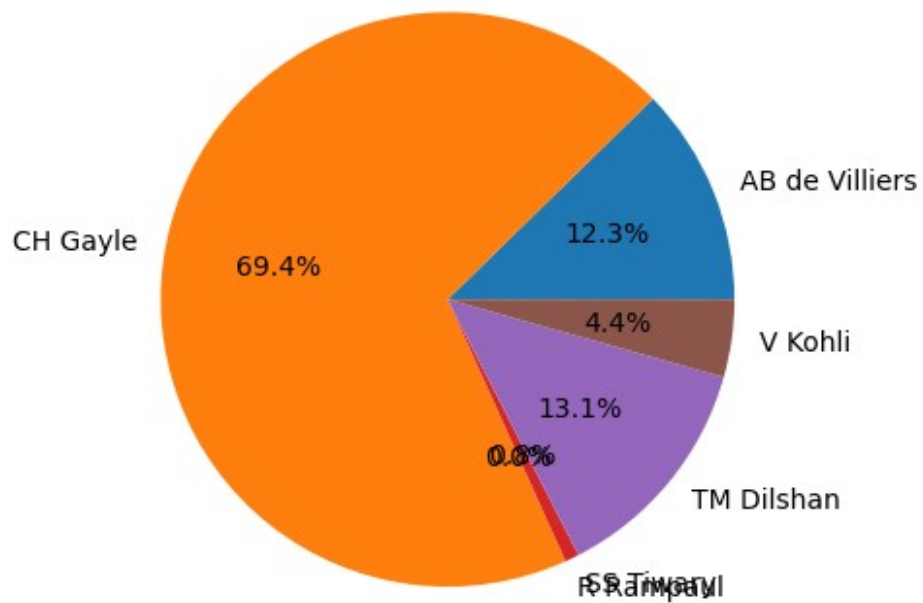
```
#dataset
df=pd.read_csv('gayle-175.csv')
df
```

	batsman	batsman_runs
0	AB de Villiers	31
1	CH Gayle	175
2	R Rampaul	0
3	SS Tiwary	2
4	TM Dilshan	33
5	V Kohli	11

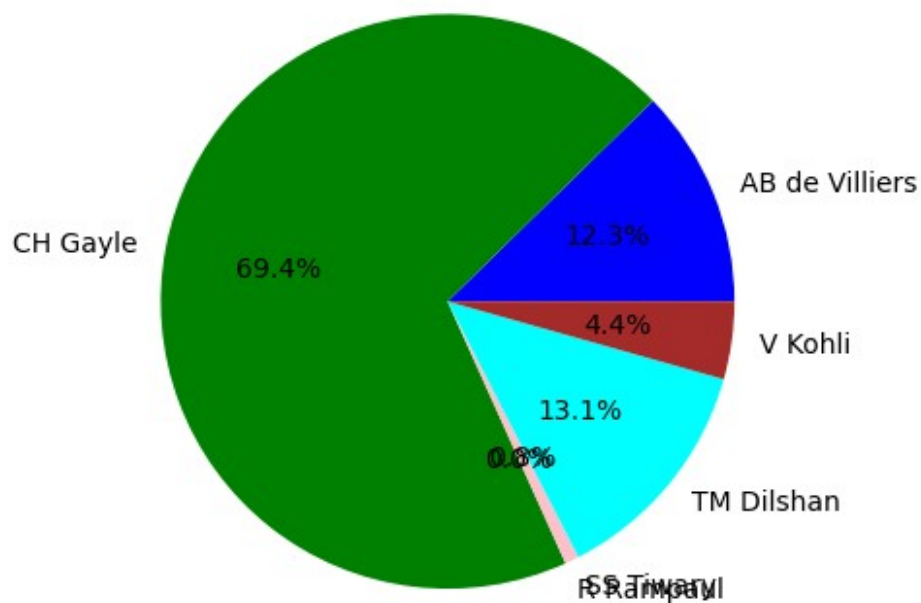
```
plt.pie(df['batsman_runs'],labels=df['batsman'])
plt.show()
```



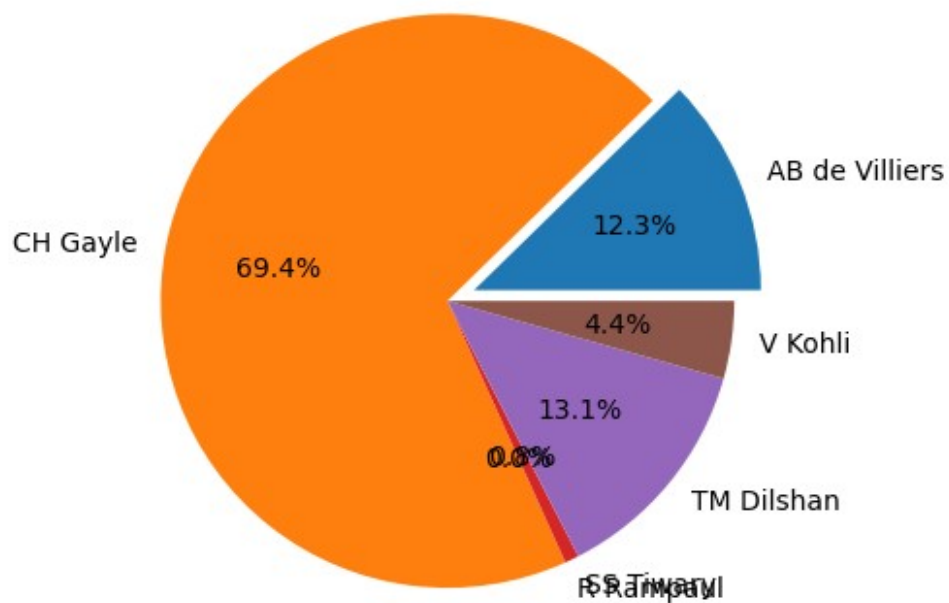
```
# suppose hume percentage bhi dikhana hai ki sare players ka  
contribution kitna hai  
plt.pie(df['batsman_runs'],labels=df['batsman'],autopct='%0.1f%%') #  
autopct --> autopercnt  
plt.show()
```



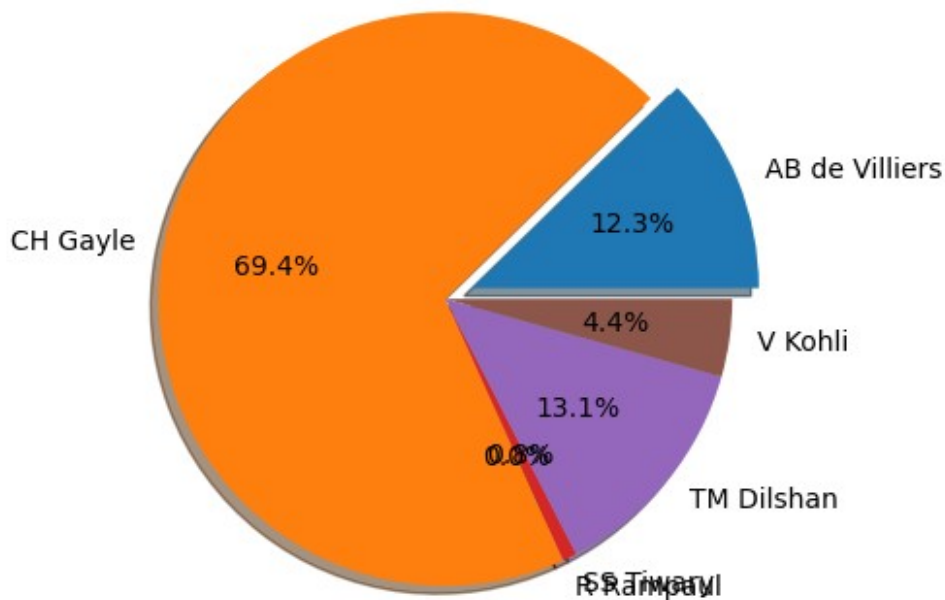
```
# you can specify colors
plt.pie(df['batsman_runs'], labels=df['batsman'], autopct='%0.1f%%', colors=['blue', 'green', 'yellow', 'pink', 'cyan', 'brown']) # autopct
--> autopercnt
plt.show()
```



```
# aap kisi bhi slice(pie) ko bahar nikal sakte ho
plt.pie(df['batsman_runs'], labels=df['batsman'], autopct='%0.1f%%', explode=[0.1,0,0,0,0,0]) # autopct --> autopercen
plt.show()
```



```
# shadow
plt.pie(df['batsman_runs'], labels=df['batsman'], autopct='%0.1f%%', explode=[0.1,0,0,0,0,0], shadow=True) # autopct --> autopercen
plt.show()
```



Changing styles

```
# humare pass bahut tarah ke style hota hai graph ko show karne ka
they are following
plt.style.available
```

```
['Solarize_Light2',
 '_classic_test_patch',
 '_mpl-gallery',
 '_mpl-gallery-nogrid',
 'bmh',
 'classic',
 'dark_background',
 'fast',
 'fivethirtyeight',
 'ggplot',
 'grayscale',
 'seaborn-v0_8',
 'seaborn-v0_8-bright',
 'seaborn-v0_8-colorblind',
 'seaborn-v0_8-dark',
 'seaborn-v0_8-dark-palette',
```

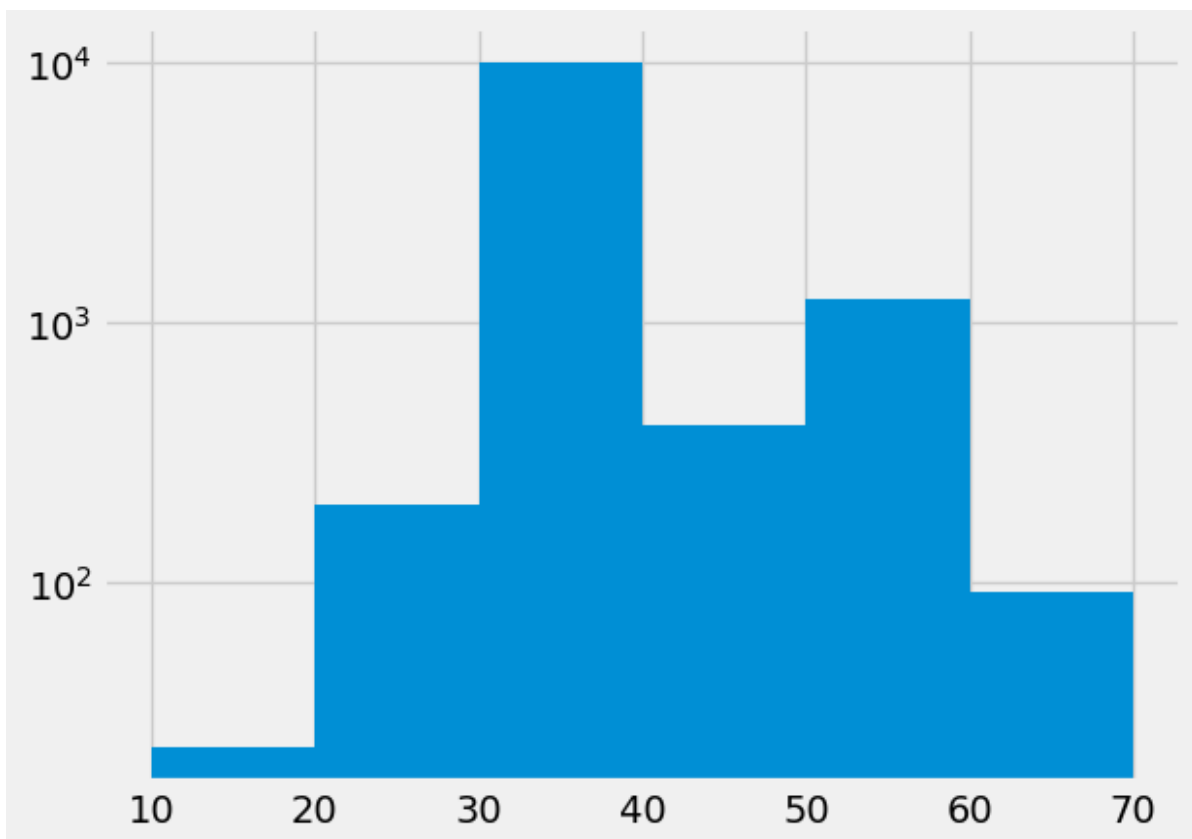


```
'seaborn-v0_8-darkgrid',  
'seaborn-v0_8-deep',  
'seaborn-v0_8-muted',  
'seaborn-v0_8-notebook',  
'seaborn-v0_8-paper',  
'seaborn-v0_8-pastel',  
'seaborn-v0_8-poster',  
'seaborn-v0_8-talk',  
'seaborn-v0_8-ticks',  
'seaborn-v0_8-white',  
'seaborn-v0_8-whitegrid',  
'tableau-colorblind10']
```

```
plt.style.use('fivethirtyeight')
```

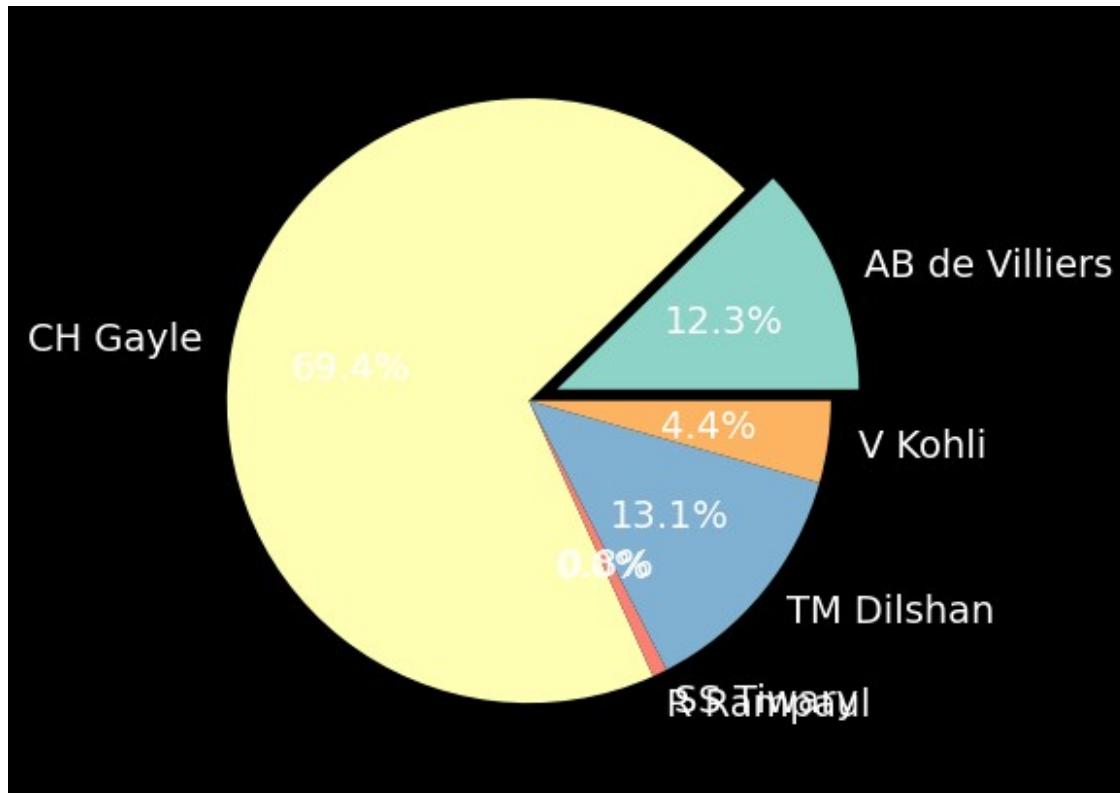
```
plt.hist(arr,bins=[10,20,30,40,50,60,70],log=True)  
plt.show()
```

ye graph pahle normal dikh raha tha



```
plt.style.use('dark_background') # abb jo pahle already run ho chuka  
hai usse agar phir se run karenge to new style jo recent bala use kye  
#honge uske according change ho  
jayega
```

```
plt.pie(df['batsman_runs'], labels=df['batsman'], autopct='%0.1f%%', explode=[0.1,0,0,0,0,0]) # autopct --> autopercnt
plt.show()
```



Save figure

- suppose hum jo bhi9 graph banaye hain usse save karke kahi send karna hai

```
plt.pie(df['batsman_runs'], labels=df['batsman'], autopct='%0.1f%%', explode=[0.1,0,0,0,0,0]) # autopct --> autopercnt
```

```
plt.savefig('sample.png')
```

dhyan rahe yaha pe plt.show() nahi karna hai jab file ko save kar rahe ho

