

day-5-matplotlib-part-1

January 12, 2023

0.0.1 Importing important library

```
[173]: import matplotlib.pyplot as plt
```

```
[174]: import pandas as pd
```

```
[175]: import seaborn as sns
```

```
[176]: import numpy as np
```

0.0.2 Import Dataset

```
[177]: df=pd.read_csv("D:\\Drivers\\matplot\\sharma-kohli.csv")
df
```

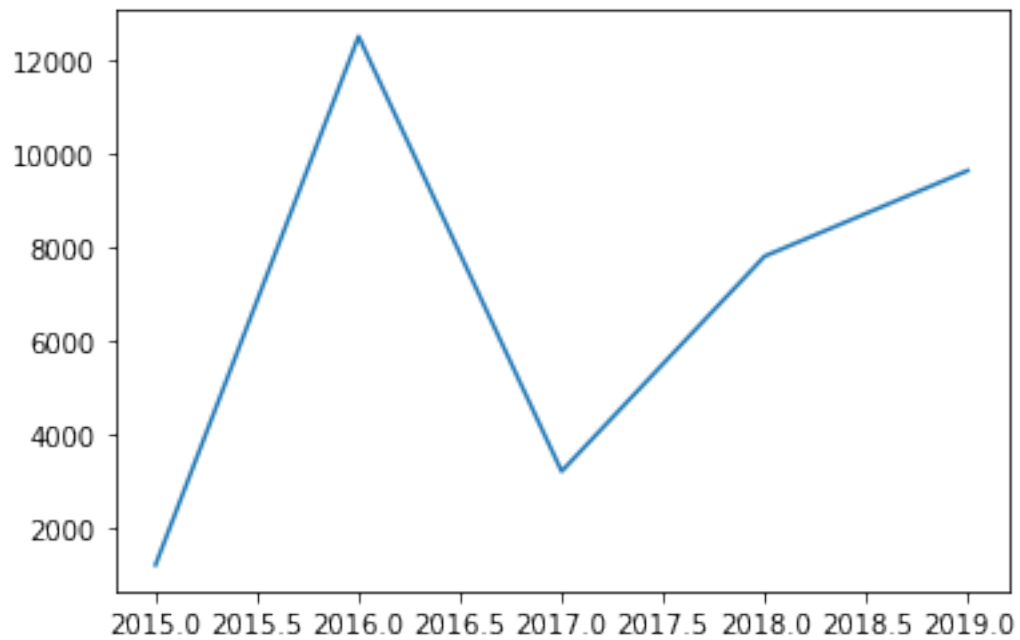
```
[177]:
```

	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

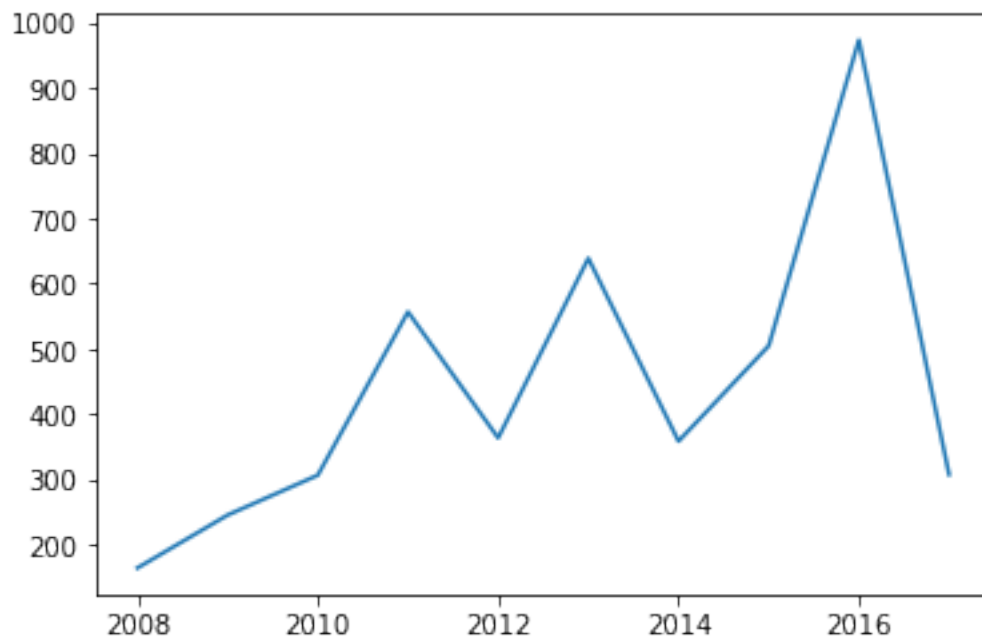
First Plot the simple plot

```
[178]: price=[1200,12500,3200,7800,9631]
year=[2015,2016,2017,2018,2019]
plt.plot(year,price)
```

```
[178]: [<matplotlib.lines.Line2D at 0x207e4eb3a60>]
```

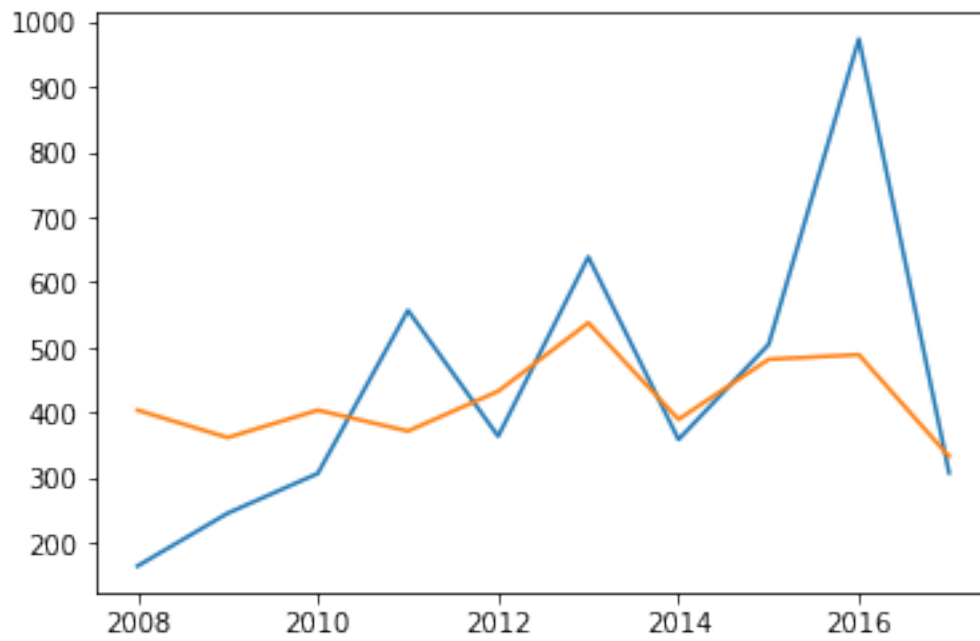


```
[179]: plt.plot(df["index"],df["V Kohli"])  
plt.show()
```



1 Multiple lines in one plot

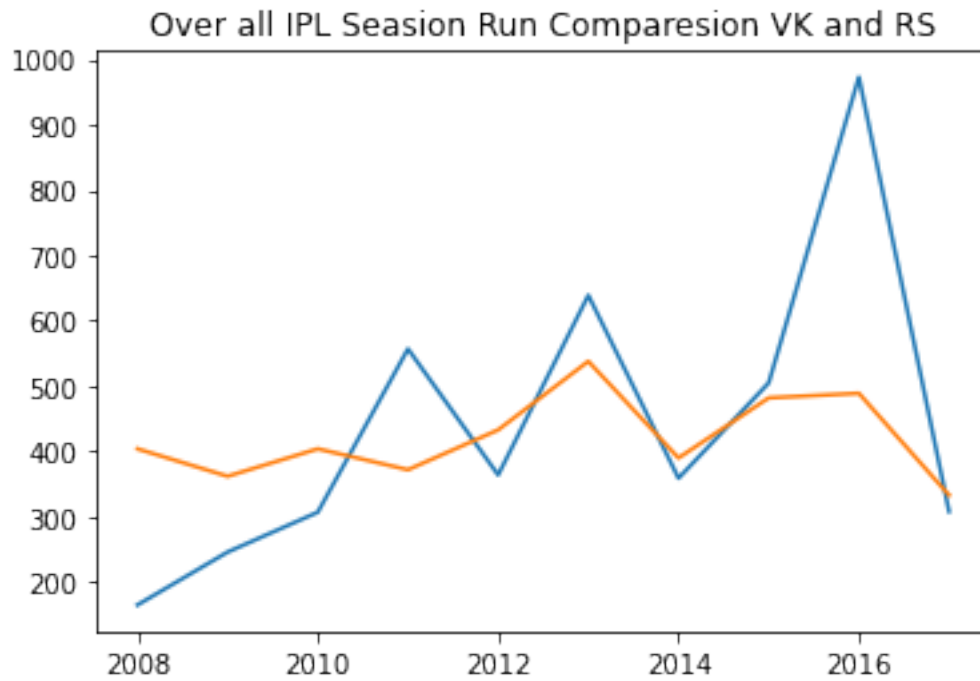
```
[180]: plt.plot(df["index"],df["V Kohli"])
plt.plot(df["index"],df["RG Sharma"])
plt.show()
```



Give the title of the graph

```
[181]: plt.plot(df["index"],df["V Kohli"])
plt.plot(df["index"],df["RG Sharma"])
plt.title("Over all IPL Seasion Run Comparesion VK and RS")

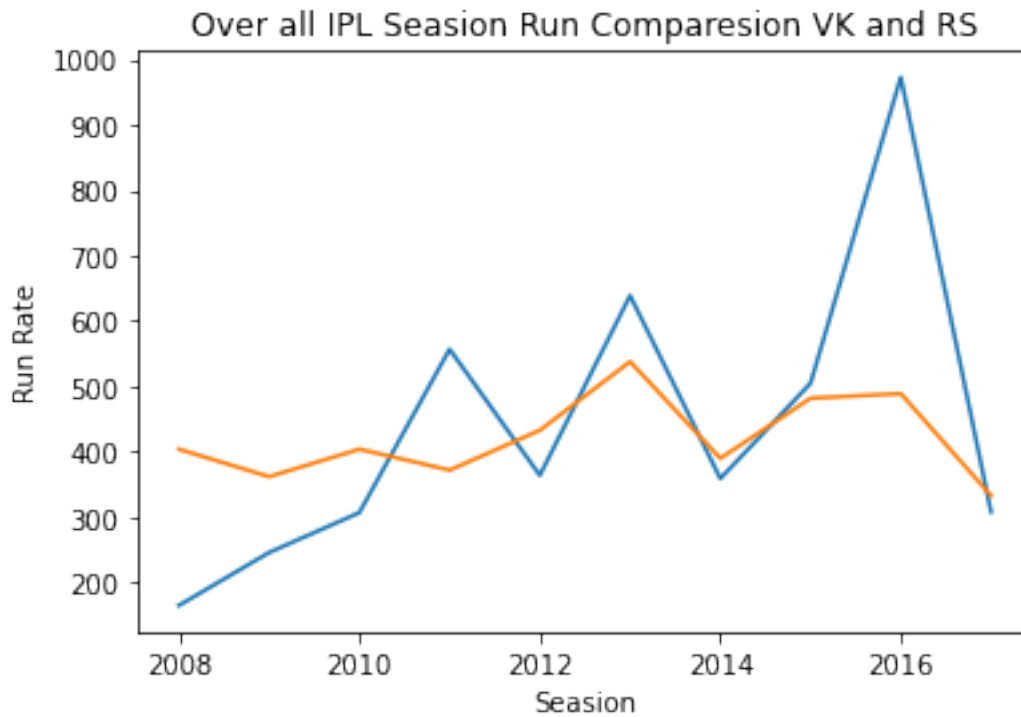
plt.show()
```



Give the xlabel and ylabel of the plot

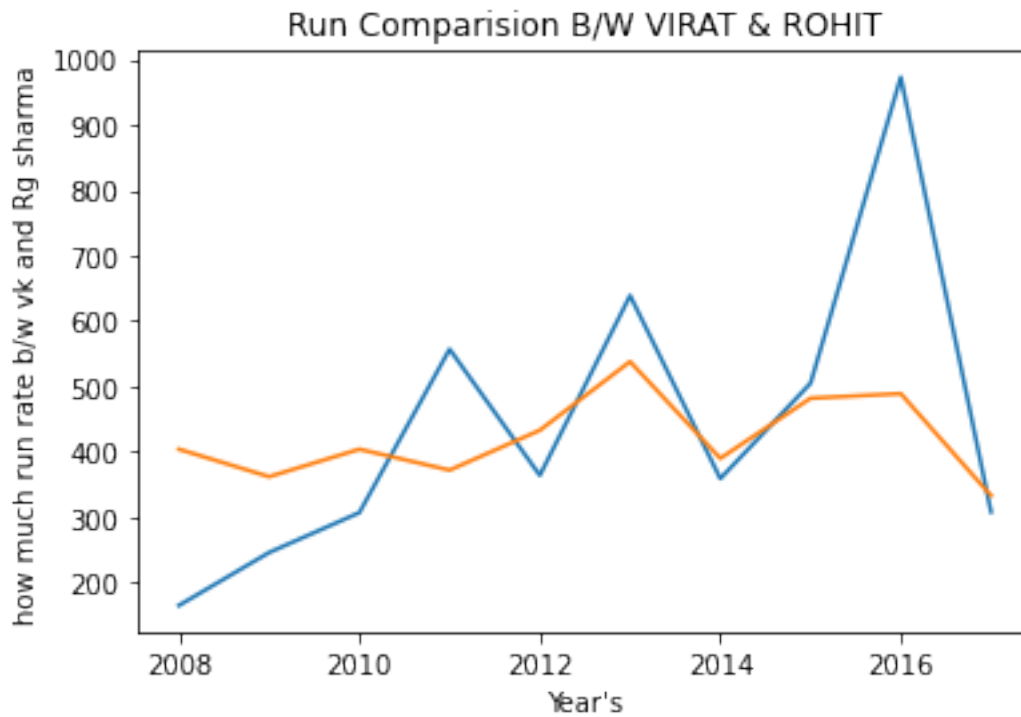
```
[182]: plt.plot(df["index"],df["V Kohli"])
plt.plot(df["index"],df["RG Sharma"])
plt.title("Over all IPL Seasion Run Comparesion VK and RS")
plt.xlabel("Seasion")
plt.ylabel("Run Rate")

plt.show()
```



```
[183]: plt.plot(df["index"],df["V Kohli"])
plt.plot(df["index"],df["RG Sharma"])
plt.title("Run Comparision B/W VIRAT & ROHIT")
plt.xlabel("Year's")
plt.ylabel("how much run rate b/w vk and Rg sharma")
plt.plot()
```

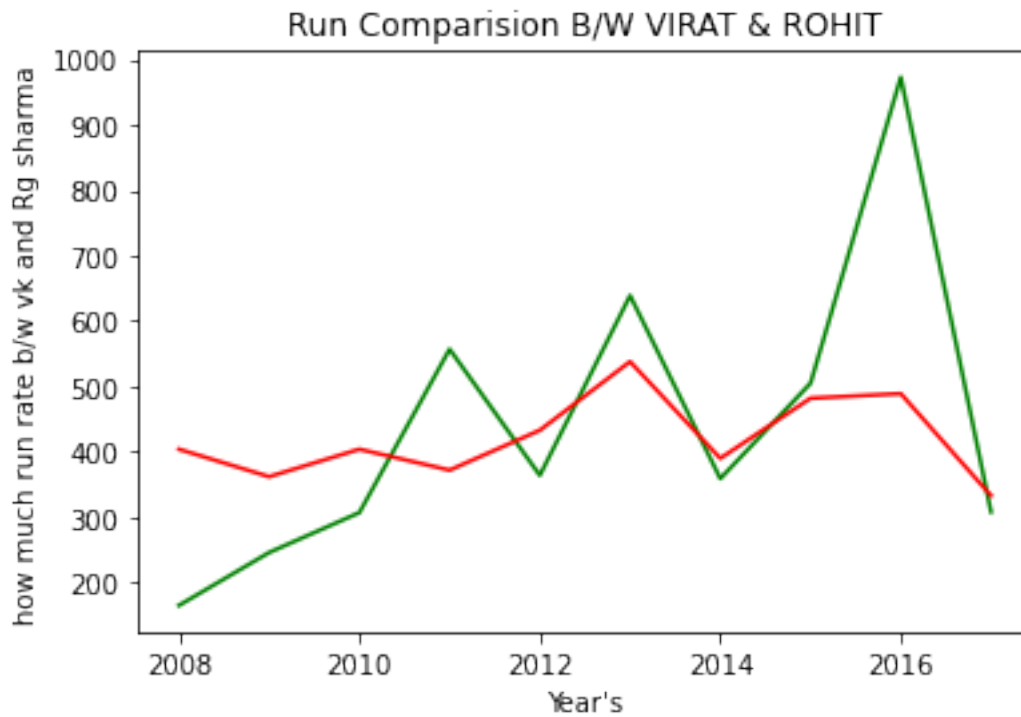
[183]: []



Give Specific line colour

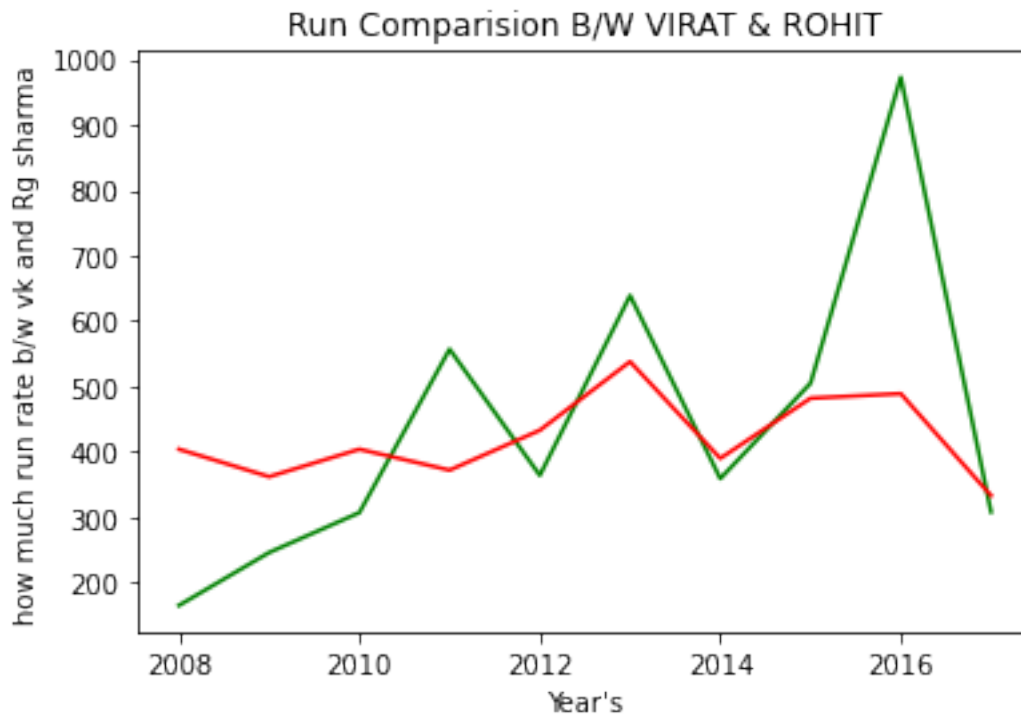
```
[184]: plt.plot(df["index"],df["V Kohli"],color="green")
plt.plot(df["index"],df["RG Sharma"],color="red")
plt.title("Run Comparision B/W VIRAT & ROHIT")
plt.xlabel("Year's")
plt.ylabel("how much run rate b/w vk and Rg sharma")
plt.plot()
```

[184]: []



```
[185]: plt.plot(df["index"],df["V Kohli"],color="green")
plt.plot(df["index"],df["RG Sharma"],color="red")
plt.title("Run Comparision B/W VIRAT & ROHIT")
plt.xlabel("Year's")
plt.ylabel("how much run rate b/w vk and Rg sharma")
plt.plot()
```

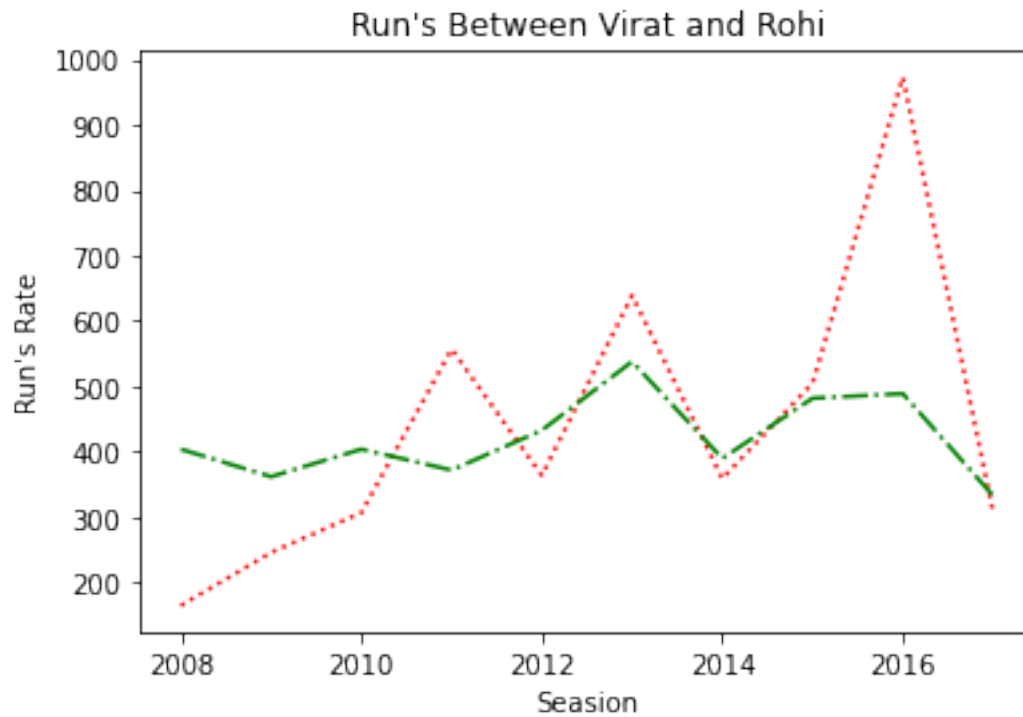
[185]: []



Specify Line style

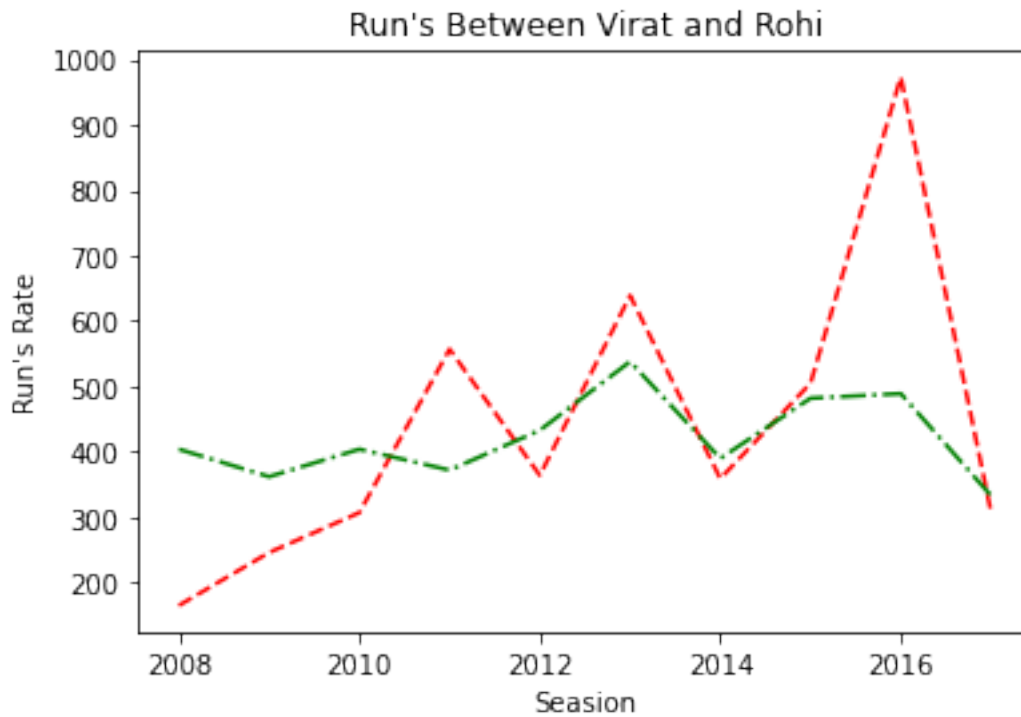
```
[186]: plt.plot(df["index"],df["V Kohli"],color="red",linestyle=":")
plt.plot(df["index"],df["RG Sharma"],color="green",linestyle="-.")
plt.title("Run's Between Virat and Rohi")
plt.xlabel("Seasion")
plt.ylabel("Run's Rate")
plt.plot()
```

[186]: []



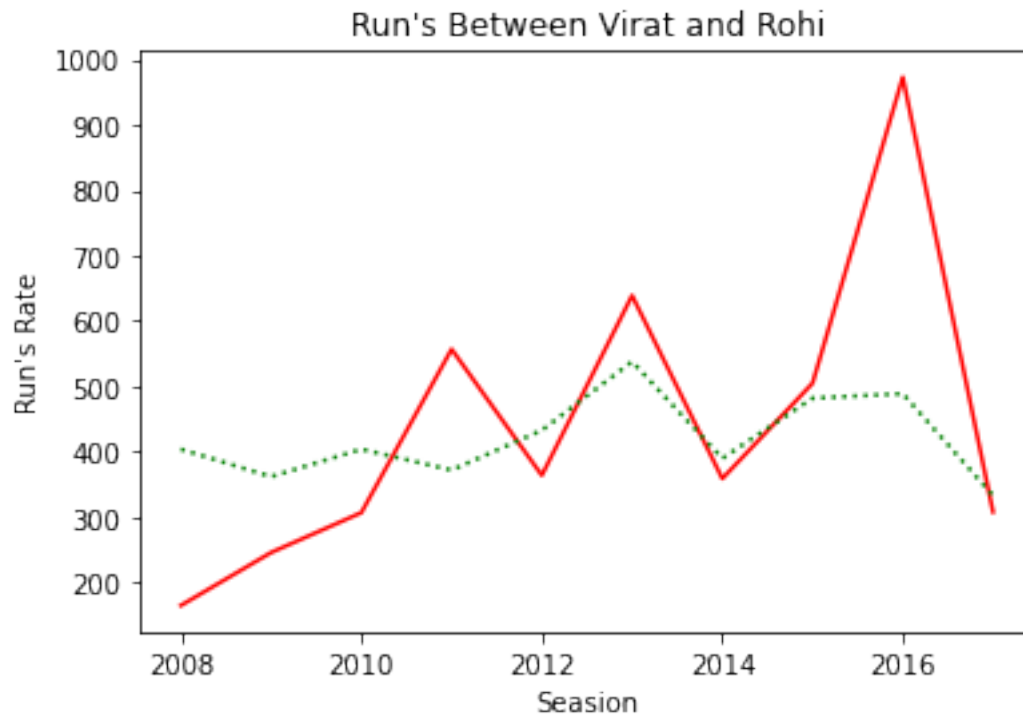
```
[187]: plt.plot(df["index"],df["V Kohli"],color="red",linestyle="--")
plt.plot(df["index"],df["RG Sharma"],color="green",linestyle="-.")
plt.title("Run's Between Virat and Rohi")
plt.xlabel("Seasion")
plt.ylabel("Run's Rate")
plt.plot()
```

[187]: []



```
[188]: plt.plot(df["index"],df["V Kohli"],color="red",linestyle="--")
plt.plot(df["index"],df["RG Sharma"],color="green",linestyle=":")
plt.title("Run's Between Virat and Rohi")
plt.xlabel("Seasion")
plt.ylabel("Run's Rate")
plt.plot()
```

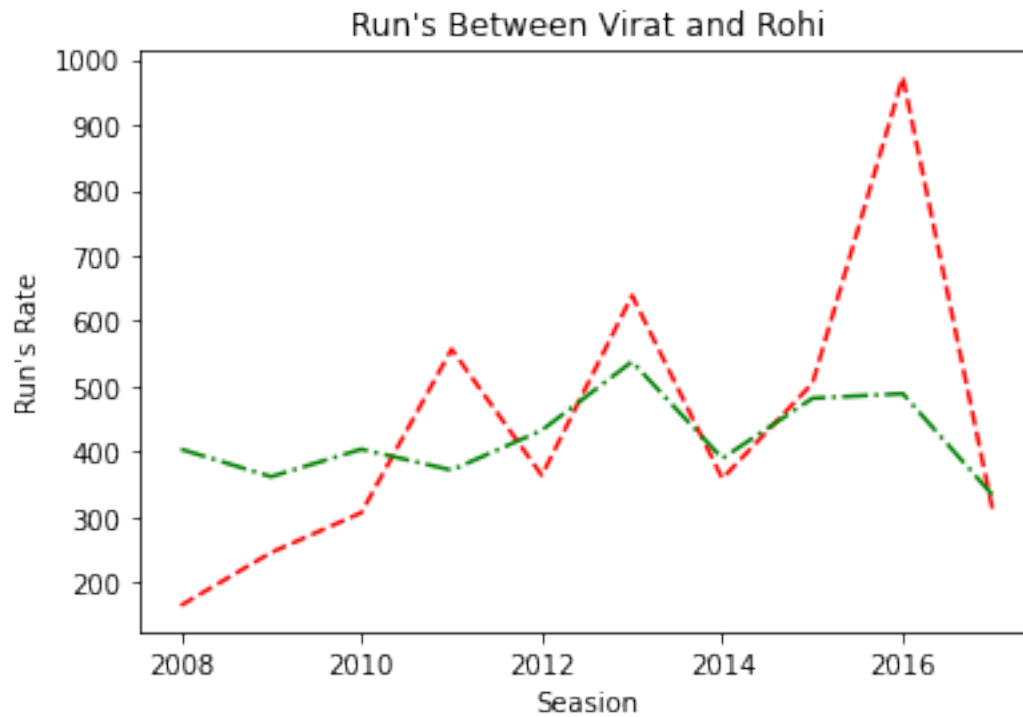
[188]: []



[]:

```
[189]: plt.plot(df["index"],df["V Kohli"],color="red",linestyle='--')
plt.plot(df["index"],df["RG Sharma"],color="green",linestyle="-.",)
plt.title("Run's Between Virat and Rohi")
plt.xlabel("Seasion")
plt.ylabel("Run's Rate")
plt.plot()
```

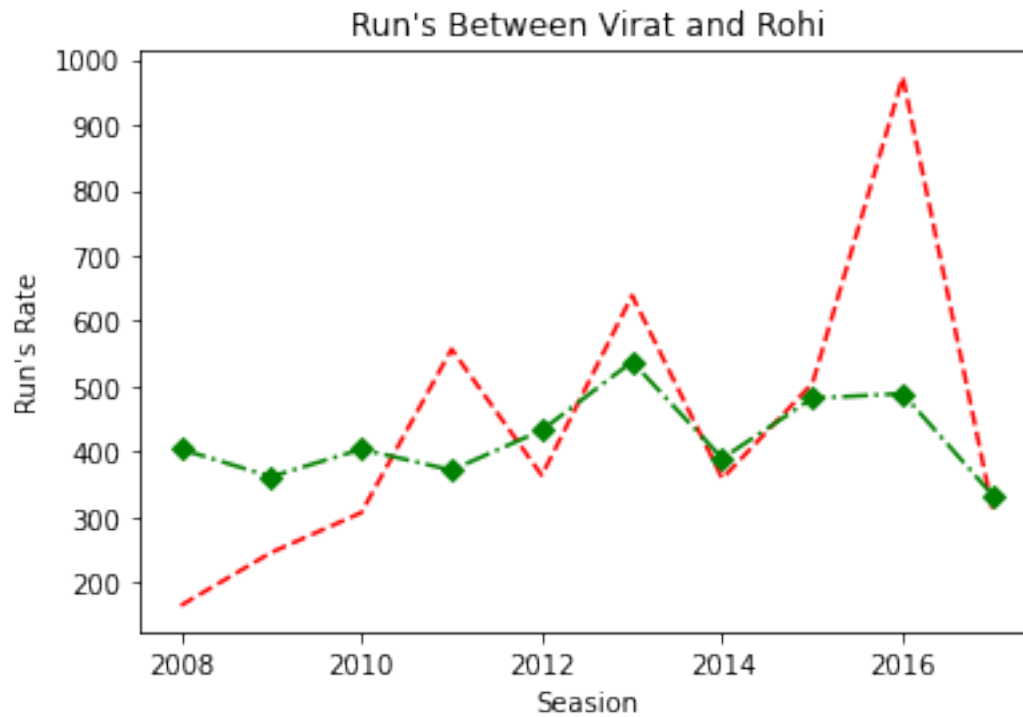
[189]: []



Give the marker

```
[190]: plt.plot(df["index"],df["V Kohli"],color="red",linestyle='--')
plt.plot(df["index"],df["RG Sharma"],color="green",linestyle="-.",marker="D")
plt.title("Run's Between Virat and Rohi")
plt.xlabel("Seasion")
plt.ylabel("Run's Rate")
plt.plot()
```

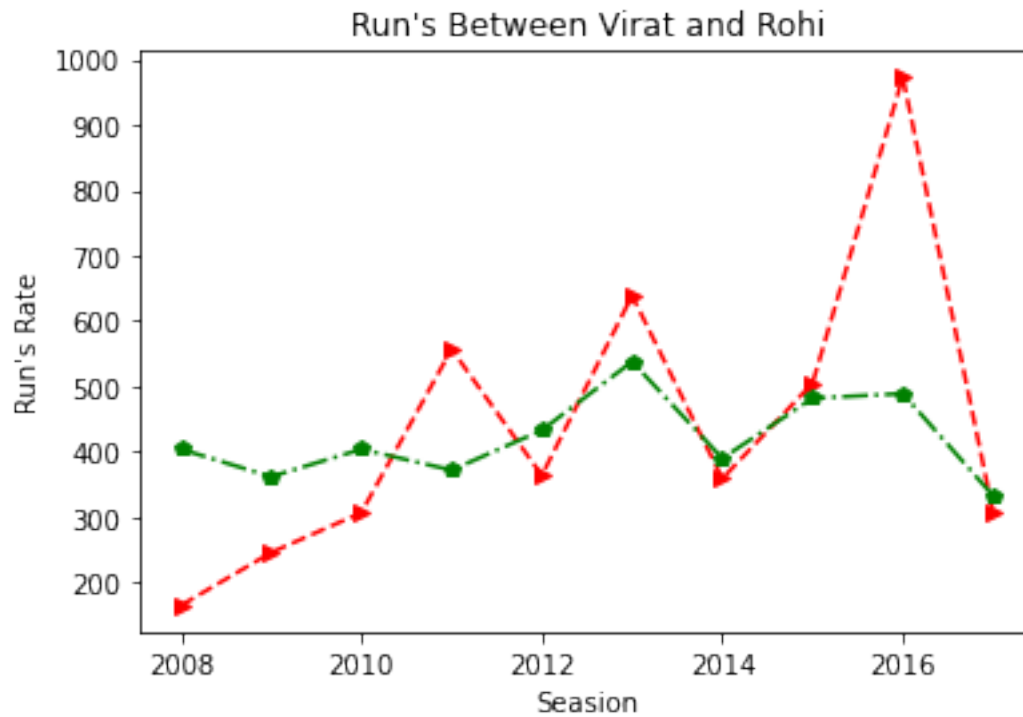
[190]: []



Diffrent marker point

```
[191]: plt.plot(df["index"],df["V Kohli"],color="red",linestyle='--',marker=">")
plt.plot(df["index"],df["RG Sharma"],color="green",linestyle="-.",marker="p")
plt.title("Run's Between Virat and Rohi")
plt.xlabel("Seasion")
plt.ylabel("Run's Rate")
plt.plot()
```

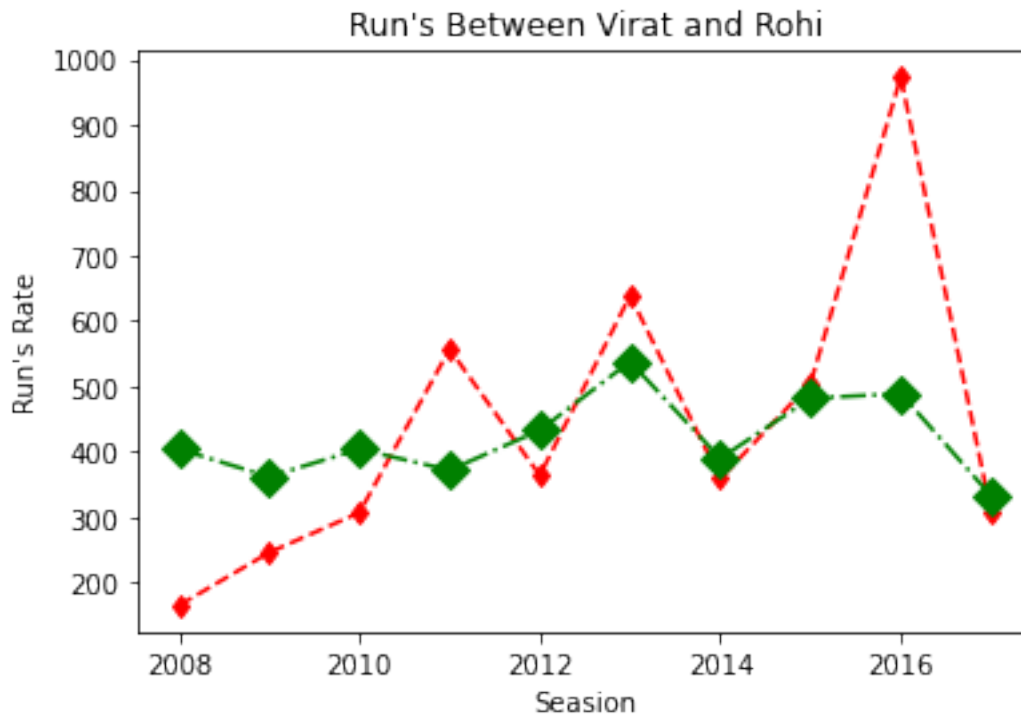
[191]: []



Marker size

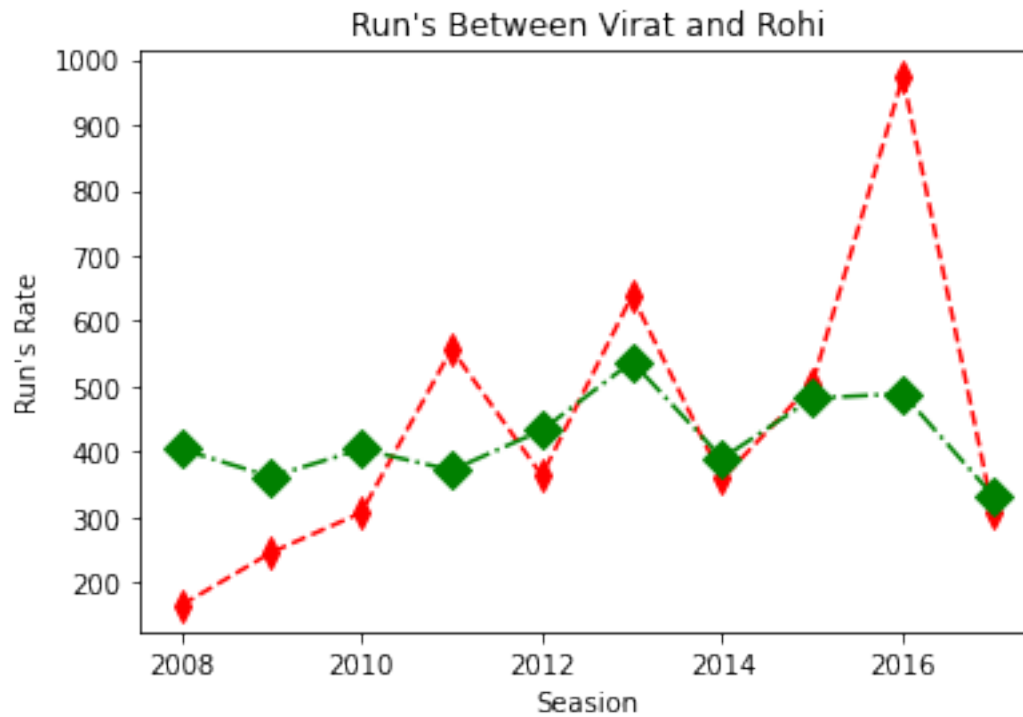
```
[192]: plt.plot(df["index"],df["V Kohli"],color="red",linestyle='--',marker="d")
plt.plot(df["index"],df["RG Sharma"],color="green",linestyle="-.
↵",marker="D",markersize=10)
plt.title("Run's Between Virat and Rohi")
plt.xlabel("Seasion")
plt.ylabel("Run's Rate")
plt.plot()
```

[192]: []



```
[193]: plt.plot(df["index"],df["Virat Kohli"],color="red",linestyle='--',marker="d",markersize=8)
plt.plot(df["index"],df["RG Sharma"],color="green",linestyle="-.",marker="D",markersize=10)
plt.title("Run's Between Virat and Rohi")
plt.xlabel("Seasion")
plt.ylabel("Run's Rate")
plt.plot()
```

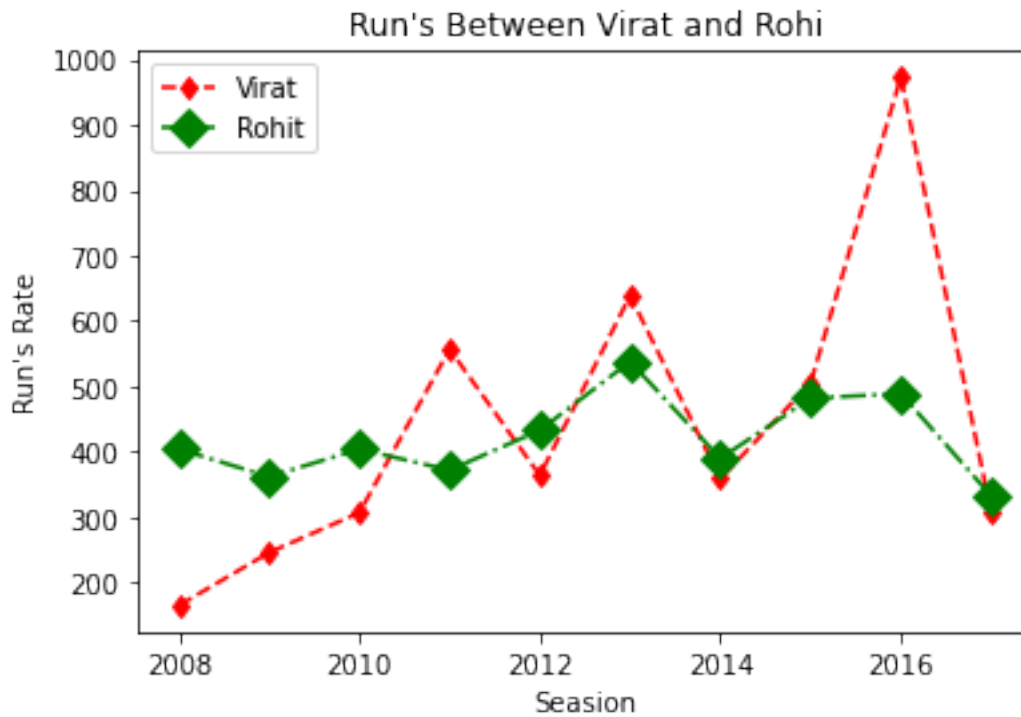
[193]: []



Give the label

```
[194]: plt.plot(df["index"],df["Virat Kohli"],color="red",linestyle='--',marker="d",label="Virat")
plt.plot(df["index"],df["RG Sharma"],color="green",linestyle="-.",marker="D",markersize=10,label="Rohit")
plt.title("Run's Between Virat and Rohi")
plt.xlabel("Seasion")
plt.ylabel("Run's Rate")
plt.legend()
plt.plot()
```

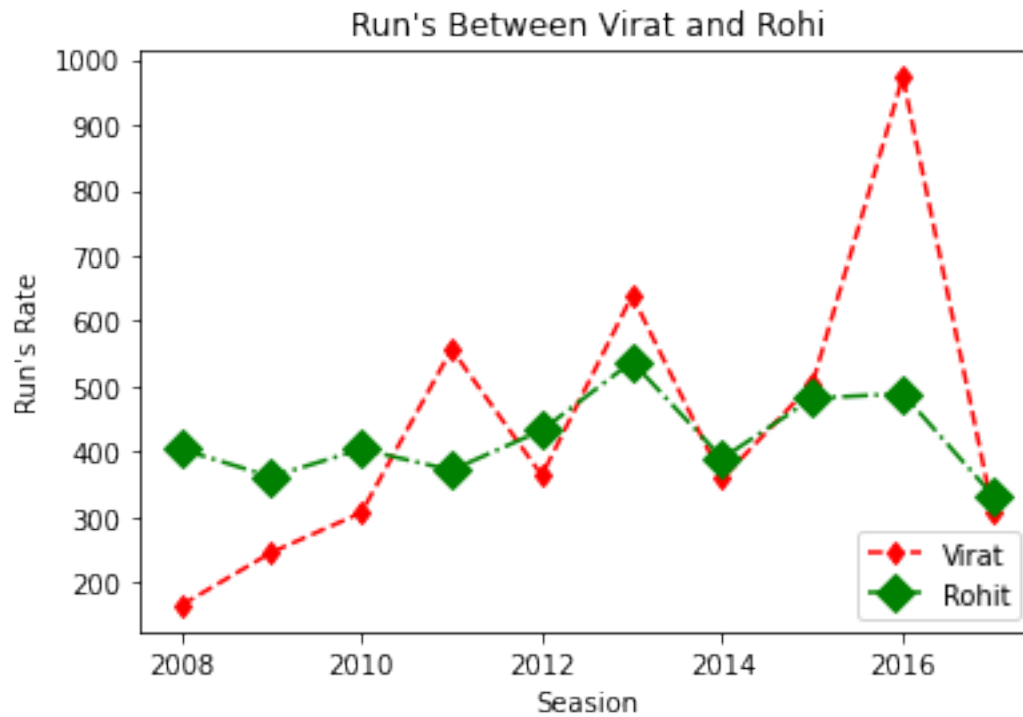
[194]: []



Specify label position by our requirement

```
[195]: plt.plot(df["index"],df["Virat Kohli"],color="red",linestyle='--',marker="d",label="Virat")
plt.plot(df["index"],df["RG Sharma"],color="green",linestyle="-.",marker="D",markersize=10,label="Rohit")
plt.title("Run's Between Virat and Rohi")
plt.xlabel("Seasion")
plt.ylabel("Run's Rate")
plt.legend(loc="lower right")
plt.plot()
```

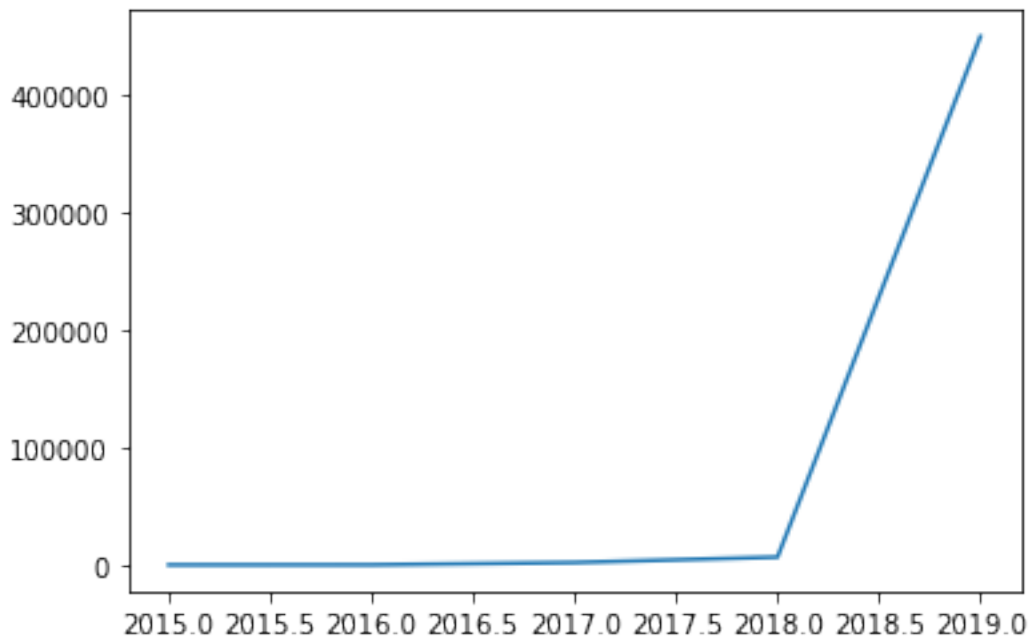
[195]: []



Trim the plot by our requirement

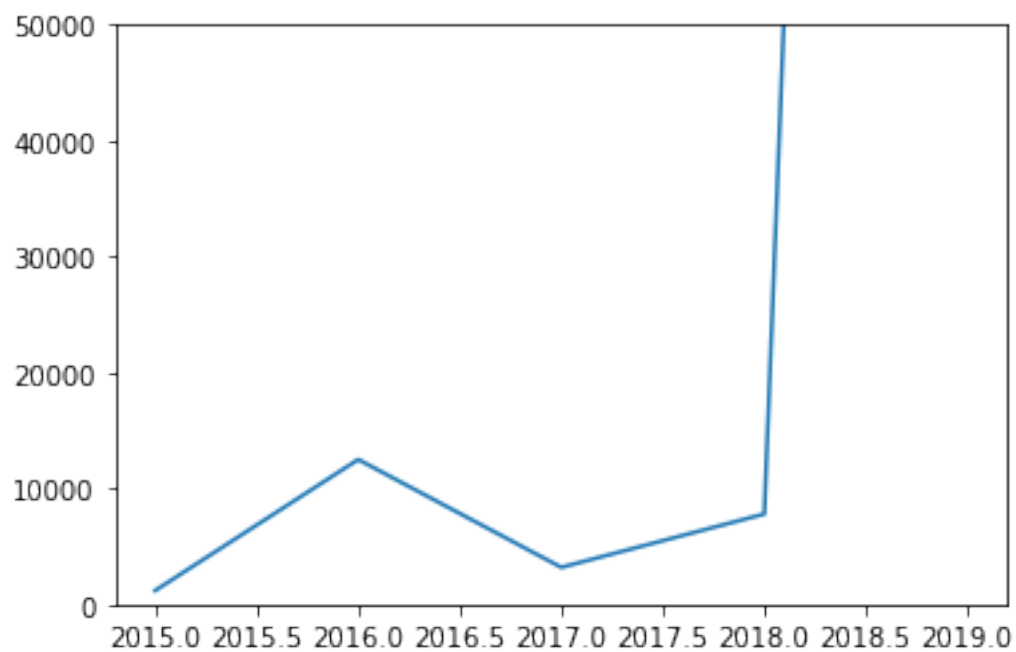
```
[196]: price=[1200,1250,3200,7800,450000]
      year=[2015,2016,2017,2018,2019]
      plt.plot(year,price)
```

```
[196]: [<matplotlib.lines.Line2D at 0x207e2903c40>]
```



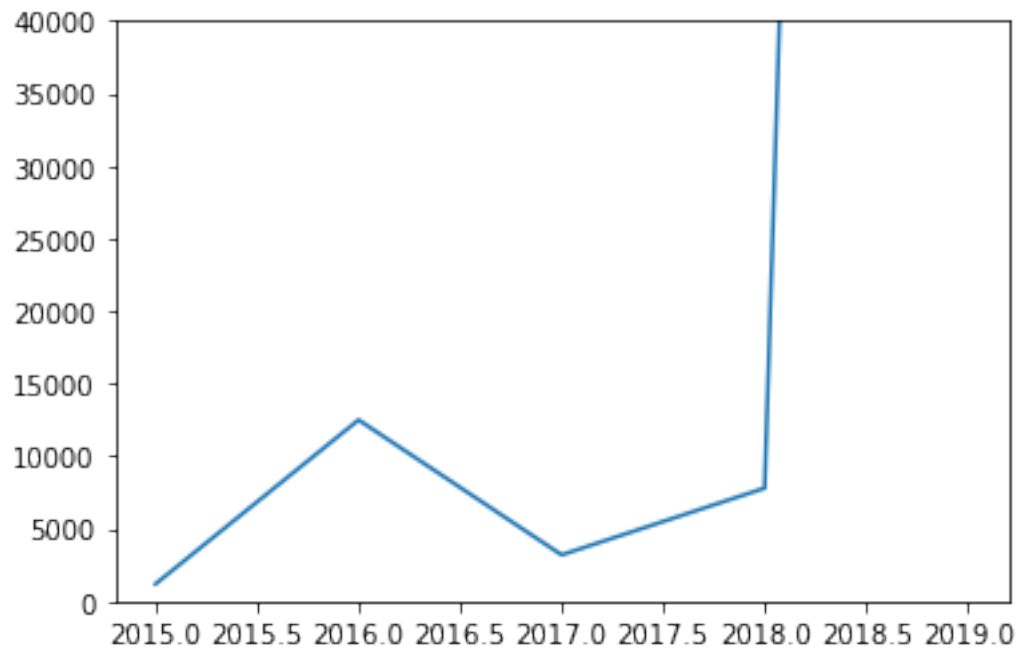
```
[197]: price=[1200,12500,3200,7800,450000]  
year=[2015,2016,2017,2018,2019]  
plt.plot(year,price)  
plt.ylim(0,50000)
```

[197]: (0.0, 50000.0)



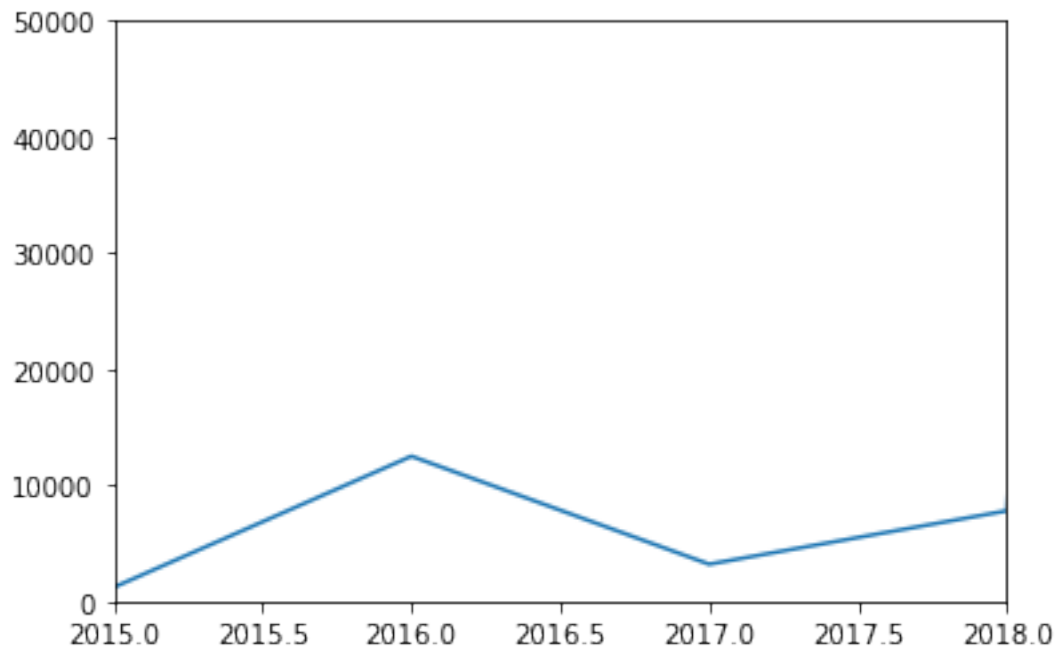
```
[198]: price=[1200,12500,3200,7800,450000]  
year=[2015,2016,2017,2018,2019]  
plt.plot(year,price)  
plt.ylim(0,40000)
```

[198]: (0.0, 40000.0)



```
[199]: price=[1200,12500,3200,7800,450000]  
year=[2015,2016,2017,2018,2019]  
plt.plot(year,price)  
plt.ylim(0,50000)  
plt.xlim(2015,2018)
```

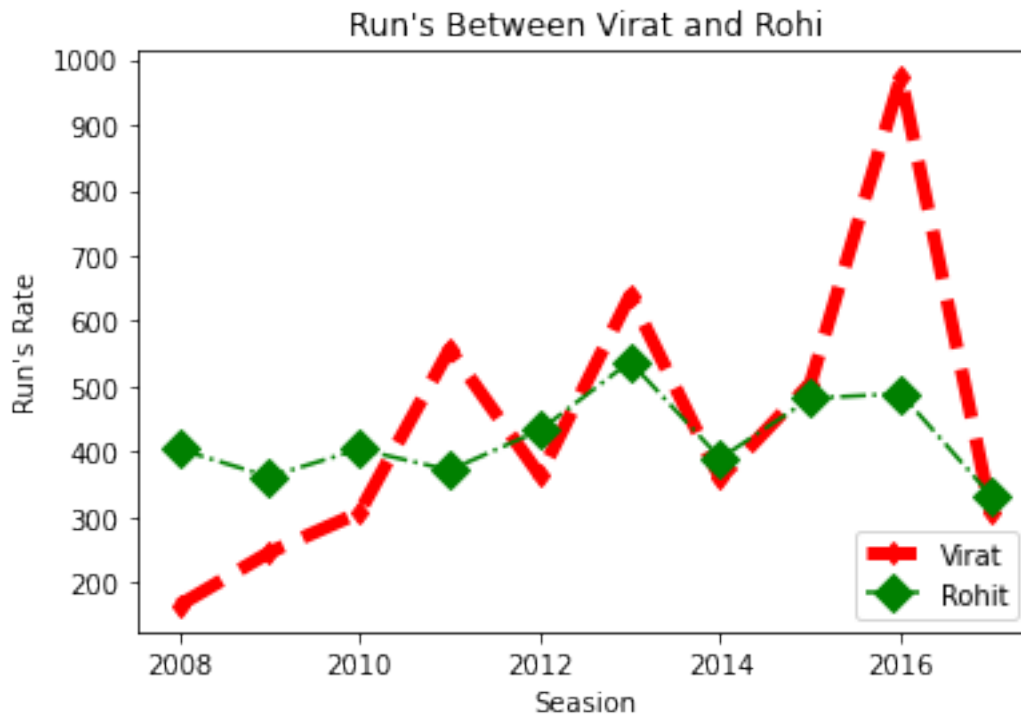
[199]: (2015.0, 2018.0)



Give the line width

```
[200]: plt.plot(df["index"],df["V_
↳Kohli"],color="red",linestyle='--',marker="d",label="Virat",linewidth=5)
plt.plot(df["index"],df["RG Sharma"],color="green",linestyle="-.
↳",marker="D",markersize=10,label="Rohit")
plt.title("Run's Between Virat and Rohi")
plt.xlabel("Seasion")
plt.ylabel("Run's Rate")
plt.legend(loc="lower right")
plt.plot()
```

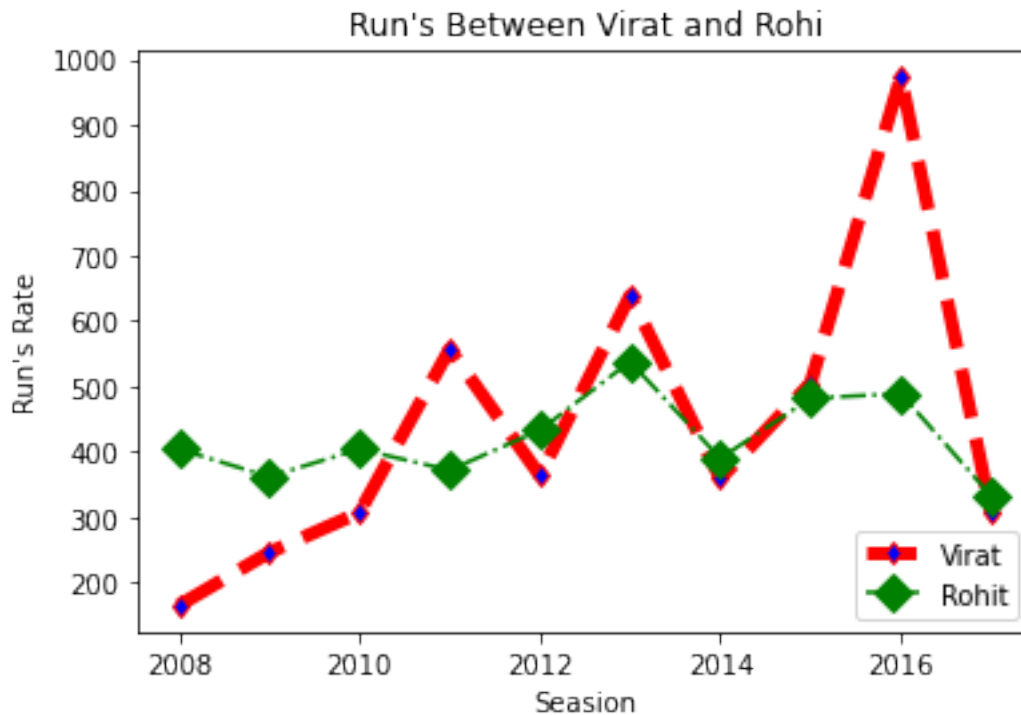
[200]: []



Give the marker face color

```
[201]: plt.plot(df["index"],df["Virat Kohli"],color="red",linestyle='--',marker="d",label="Virat",linewidth=5,mfc="b")
plt.plot(df["index"],df["RG Sharma"],color="green",linestyle="-.",marker="D",markersize=10,label="Rohit")
plt.title("Run's Between Virat and Rohi")
plt.xlabel("Seasion")
plt.ylabel("Run's Rate")
plt.legend(loc="lower right")
plt.plot()
```

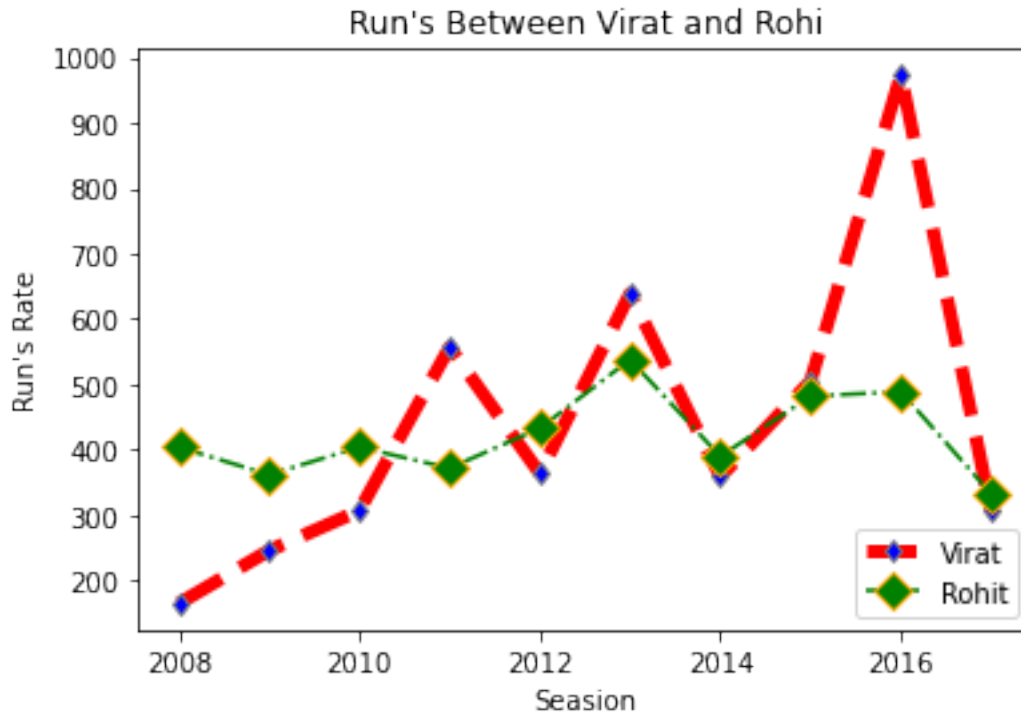
[201]: []



Give the markeredgecolor

```
[202]: plt.plot(df["index"],df["Virat"],color="red",linestyle='--',marker="d",label="Virat",linewidth=5,mfc="b",
↳markeredgecolor="grey")
plt.plot(df["index"],df["Rohit"],color="green",linestyle="-.",marker="D",markersize=10,label="Rohit", markeredgecolor="orange")
plt.title("Run's Between Virat and Rohi")
plt.xlabel("Seasion")
plt.ylabel("Run's Rate")
plt.legend(loc="lower right")
plt.plot()
```

[202]: []



Some Marker feature

```
[203]: #   markeredgecolor or mec: color
#   markeredgewidth or mew: float
#   markerfacecolor or mfc: color
#   markerfacecoloralt or mfcalt: color
```

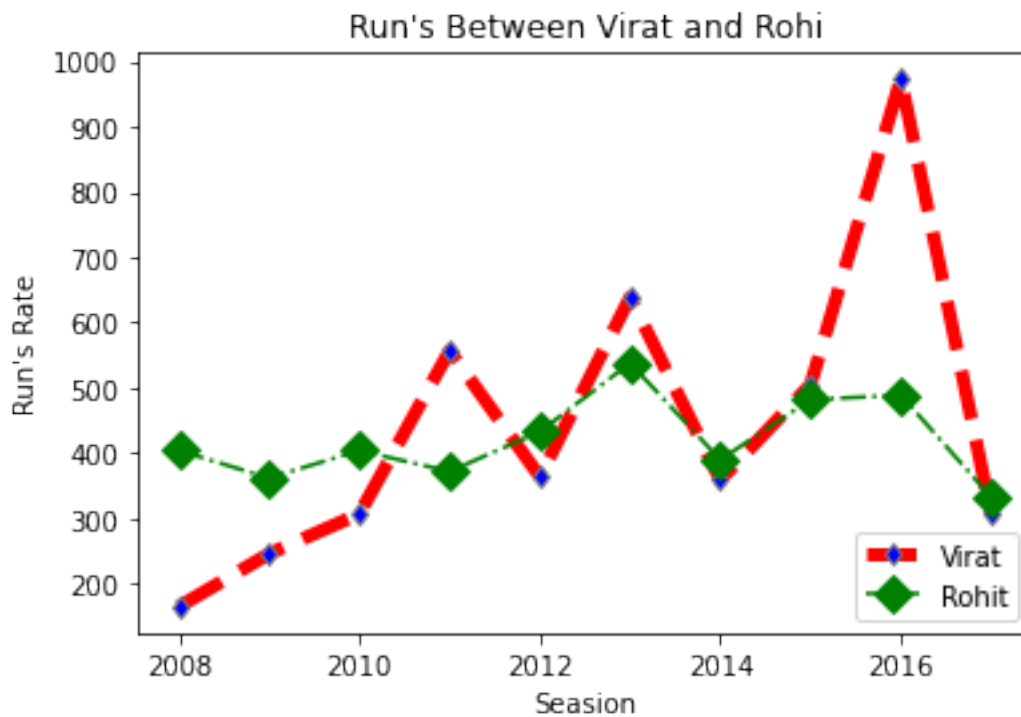
Input In [203]

```
markeredgecolor or mec: color
```

SyntaxError: illegal target for annotation

```
[204]: plt.plot(df["index"],df["V_Kohli"],color="red",linestyle='--',marker="d",label="Virat",linewidth=5,mfc="b",
           markeredgecolor="grey")
plt.plot(df["index"],df["RG Sharma"],color="green",linestyle="-.
           ",marker="D",markersize=10,label="Rohit", markerfacecoloralt='orange')
plt.title("Run's Between Virat and Rohi")
plt.xlabel("Seasion")
plt.ylabel("Run's Rate")
plt.legend(loc="lower right")
plt.plot()
```


[204]: []



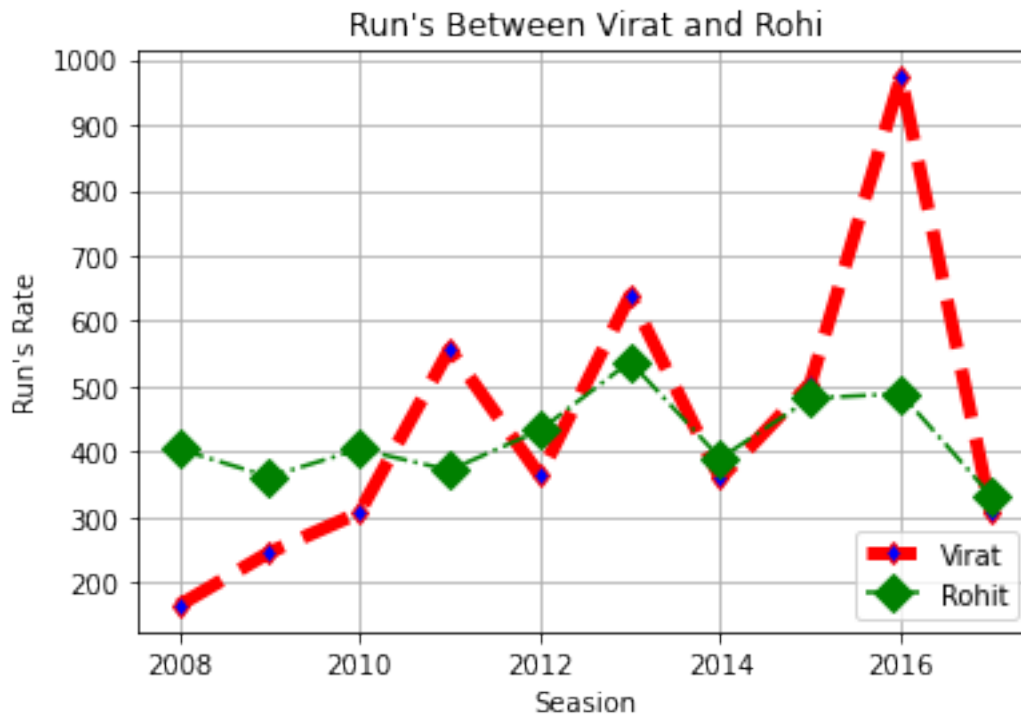
Enable Grid in the plot

```
[205]: plt.plot(df["index"],df["Virat"],color="red",linestyle='--',marker="d",label="Virat",linewidth=5,mfc="b")
plt.plot(df["index"],df["Rohit"],color="green",linestyle="-.",marker="D",markersize=10,label="Rohit")
plt.title("Run's Between Virat and Rohi")
plt.xlabel("Seasion")
plt.ylabel("Run's Rate")
plt.legend(loc="lower right")

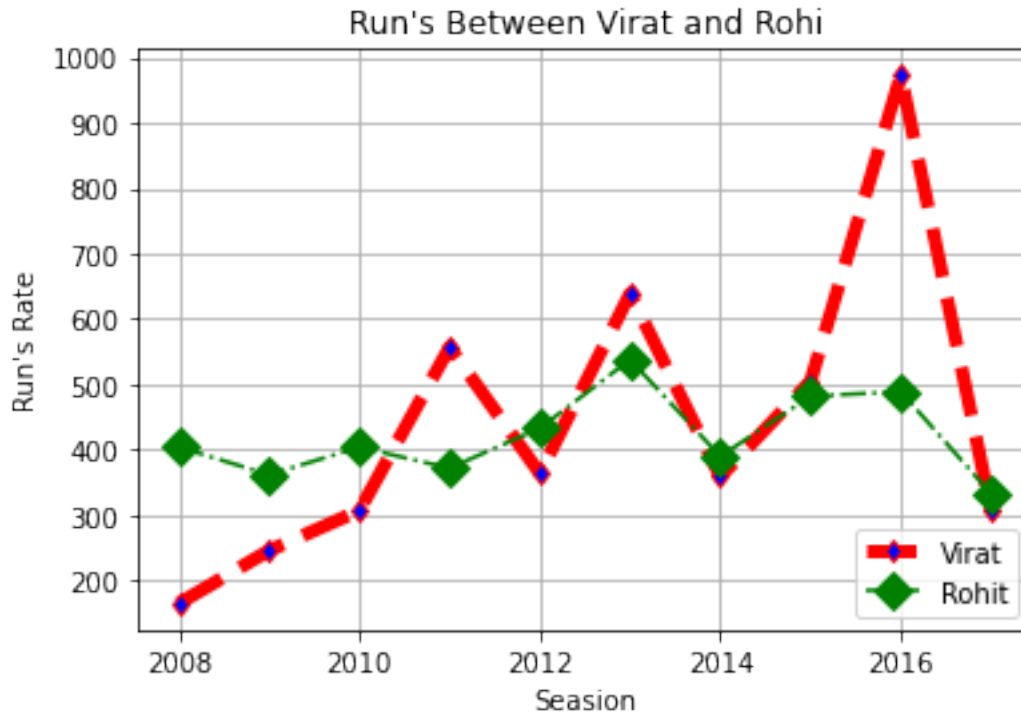
plt.grid()

plt.plot()
```

[205]: []



```
[206]: plt.plot(df["index"],df["V_Kohli"],color="red",linestyle='--',marker="d",label="Virat",linewidth=5,mfc="b")
plt.plot(df["index"],df["RG Sharma"],color="green",linestyle="-.",marker="D",markersize=10,label="Rohit")
plt.title("Run's Between Virat and Rohi")
plt.xlabel("Seasion")
plt.ylabel("Run's Rate")
plt.legend(loc="lower right")
plt.plot()
plt.grid()
```



2 Scatter Plot

Import Dataset

```
[207]: df=pd.read_csv("D:\\Drivers\\matplotlib\\batter.csv")
better
```

```
[207]:
```

	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]

```
[208]: df=df.head(50)
df
```

```
[208]:
```

	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

```
[239]: plt.scatter(df["avg"],df["strike_rate"])
plt.plot()
```

```
-----
KeyError                                Traceback (most recent call last)
File ~\python3.10\lib\site-packages\pandas\core\indexes\base.py:3621, in Index.
    get_loc(self, key, method, tolerance)
    3620 try:
-> 3621     return self._engine.get_loc(casted_key)
    3622 except KeyError as err:

File ~\python3.10\lib\site-packages\pandas\_libs\index.pyx:136, in pandas._libs
    index.IndexEngine.get_loc()

File ~\python3.10\lib\site-packages\pandas\_libs\index.pyx:163, in pandas._libs
    index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:5198, in pandas._libs.hashtable.
    PyObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:5206, in pandas._libs.hashtable.
    PyObjectHashTable.get_item()

KeyError: 'avg'
```

The above exception was the direct cause of the following exception:

```
KeyError                                Traceback (most recent call last)
Input In [239], in <module>
----> 1 plt.scatter(df["avg"],df["strike_rate"])
      2 plt.plot()

File ~\python3.10\lib\site-packages\pandas\core\frame.py:3505, in DataFrame.
    __getitem__(self, key)
    3503 if self.columns.nlevels > 1:
    3504     return self._getitem_multilevel(key)
-> 3505 indexer = self.columns.get_loc(key)
    3506 if is_integer(indexer):
    3507     indexer = [indexer]
```

```

File ~\python3.10\lib\site-packages\pandas\core\indexes\base.py:3623, in Index.
    ↪get_loc(self, key, method, tolerance)
    3621     return self._engine.get_loc(casted_key)
    3622 except KeyError as err:
-> 3623     raise KeyError(key) from err
    3624 except TypeError:
    3625     # If we have a listlike key, _check_indexing_error will raise
    3626     # InvalidIndexError. Otherwise we fall through and re-raise
    3627     # the TypeError.
    3628     self._check_indexing_error(key)

KeyError: 'avg'

```

```

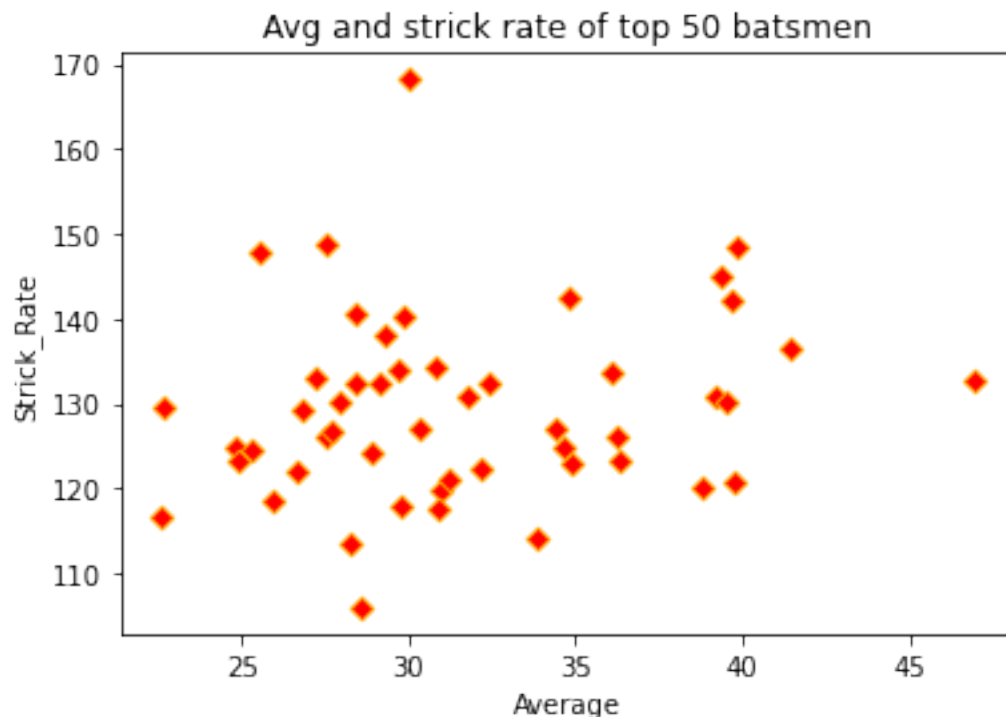
[210]: plt.
    ↪scatter(df["avg"],df["strike_rate"],color="red",marker="D",edgecolors="orange")
plt.title("Avg and strick rate of top 50 batsmen")
plt.xlabel("Average")
plt.ylabel("Strick_Rate")

```

```

[210]: Text(0, 0.5, 'Strick_Rate')

```



Load Some dataset from seaborn

```
[211]: tips=sns.load_dataset("tips")
```

```
[212]: tips
```

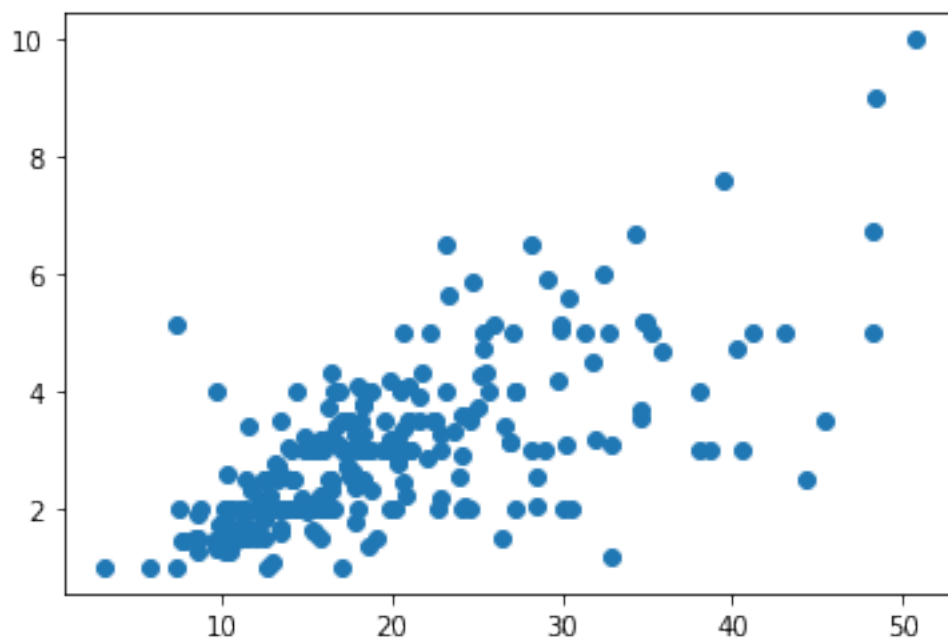
```
[212]:
```

	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]
```

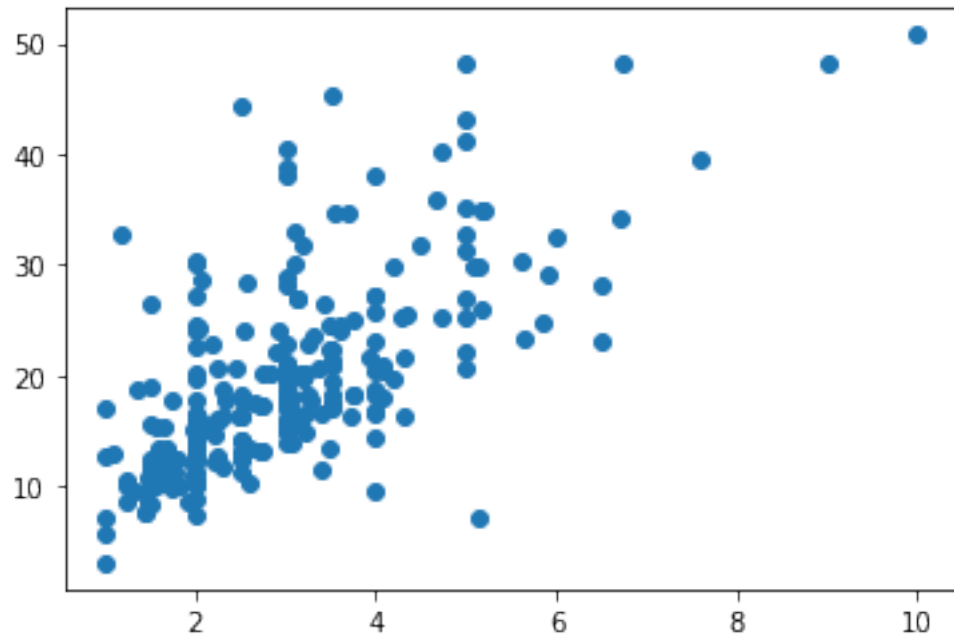
```
[213]: plt.scatter(tips["total_bill"],tips["tip"])
```

```
[213]: <matplotlib.collections.PathCollection at 0x207e6294520>
```



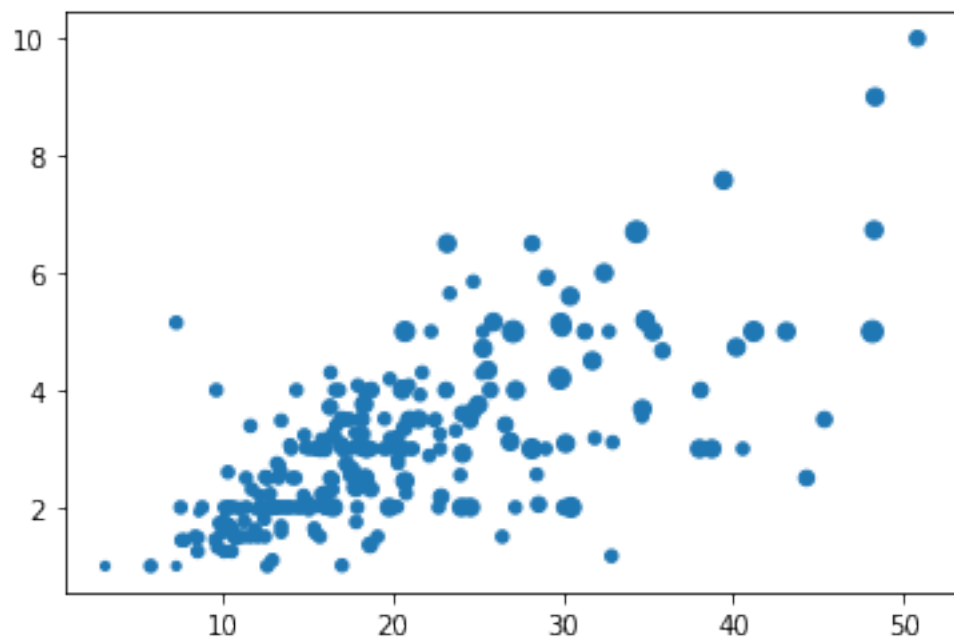
```
[215]: plt.scatter(tips["tip"],tips["total_bill"])
```

```
[215]: <matplotlib.collections.PathCollection at 0x207e63de0e0>
```



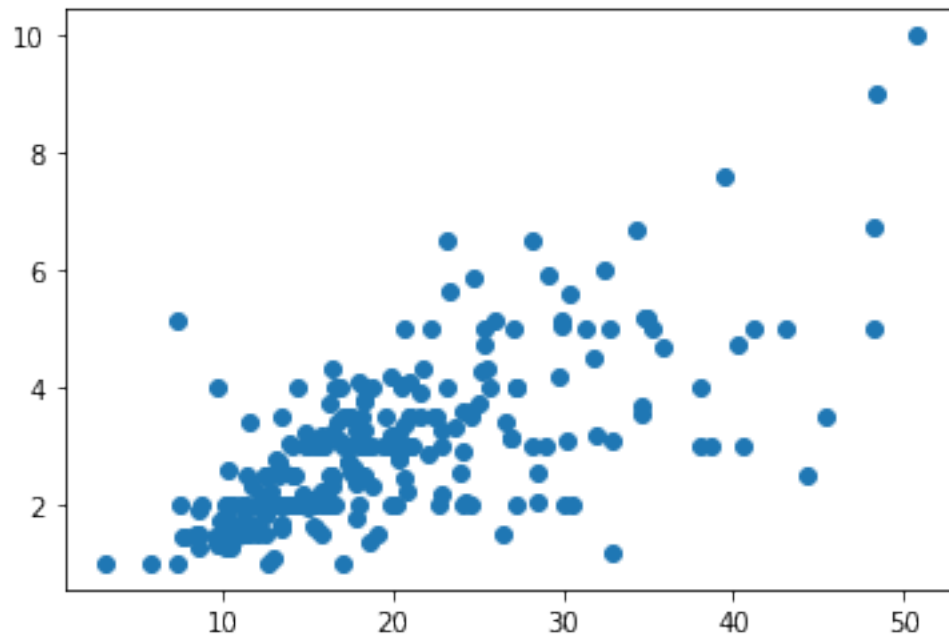
```
[216]: plt.scatter(tips["total_bill"],tips["tip"],s=tips["size"]*10)
```

```
[216]: <matplotlib.collections.PathCollection at 0x207e6359ba0>
```



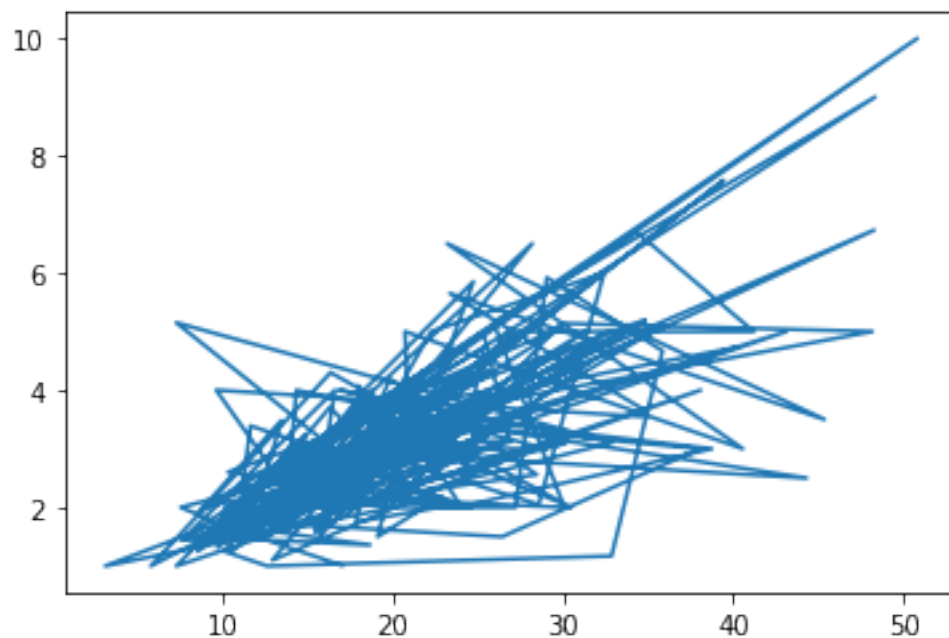

```
[217]: plt.plot(tips["total_bill"],tips["tip"],"o")
```

```
[217]: [<matplotlib.lines.Line2D at 0x207e648ebc0>]
```



```
[218]: plt.plot(tips["total_bill"],tips["tip"])
```

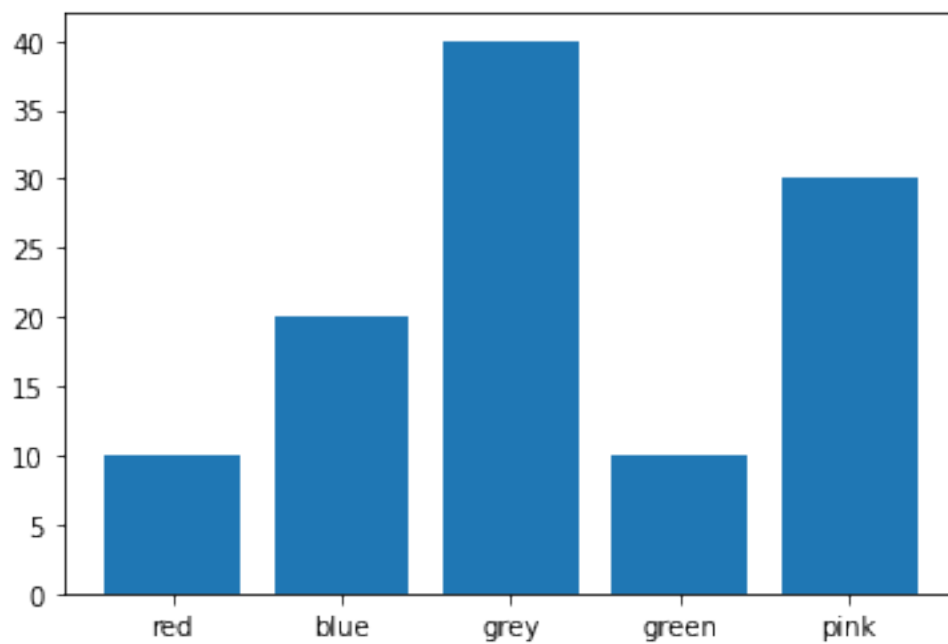
```
[218]: [<matplotlib.lines.Line2D at 0x207e6504d00>]
```



3 Bar Chart

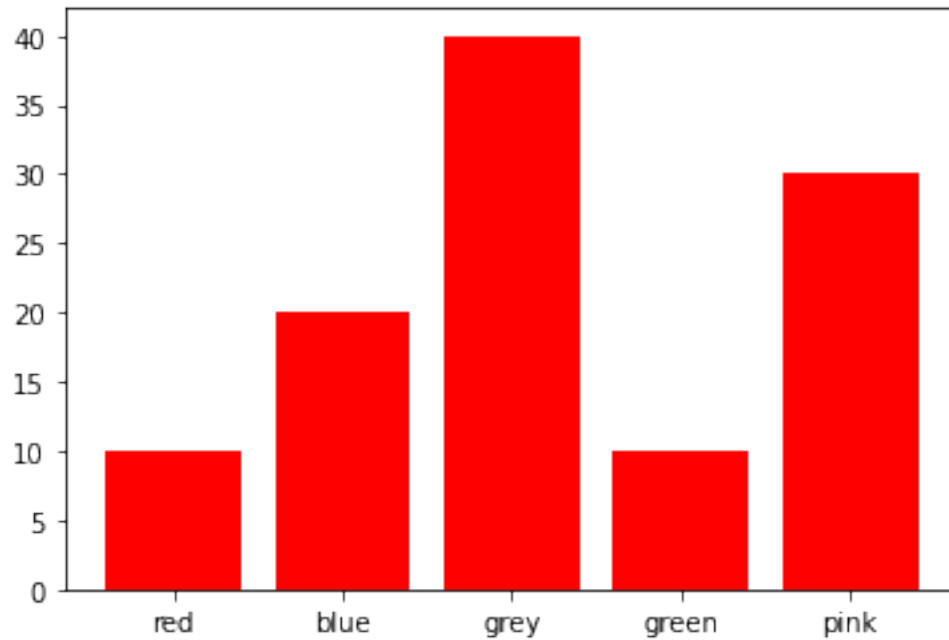
```
[219]: child=[10,20,40,10,30]  
color=['red','blue','grey','green','pink']  
plt.bar(color,child)
```

[219]: <BarContainer object of 5 artists>



```
[220]: child=[10,20,40,10,30]  
color=['red','blue','grey','green','pink']  
plt.bar(color,child,color='red')
```

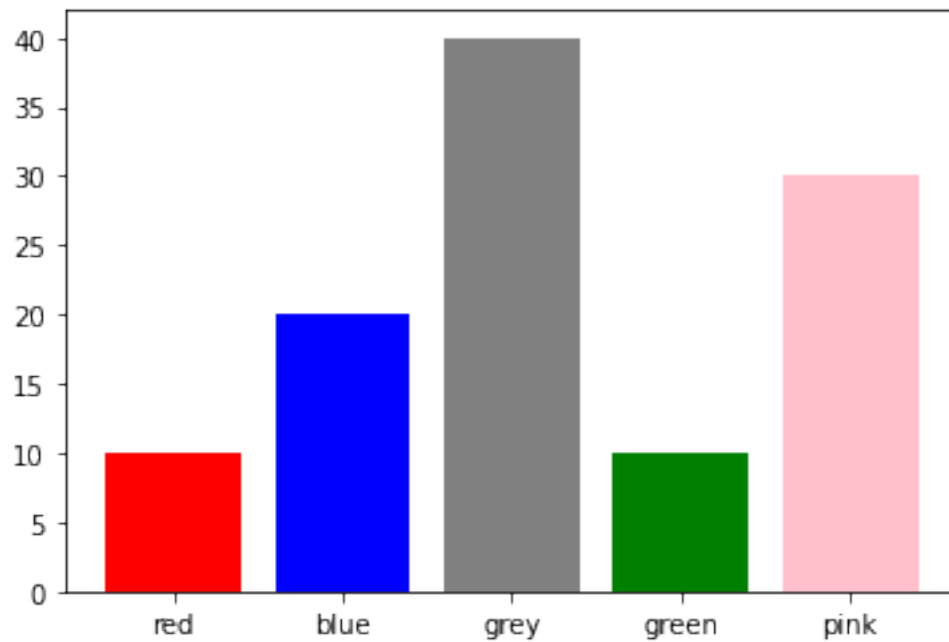
[220]: <BarContainer object of 5 artists>



Give the specific color for particular bar

```
[221]: child=[10,20,40,10,30]  
color=['red','blue','grey','green','pink']  
plt.bar(color,child,color=['red','blue','grey','green','pink'])
```

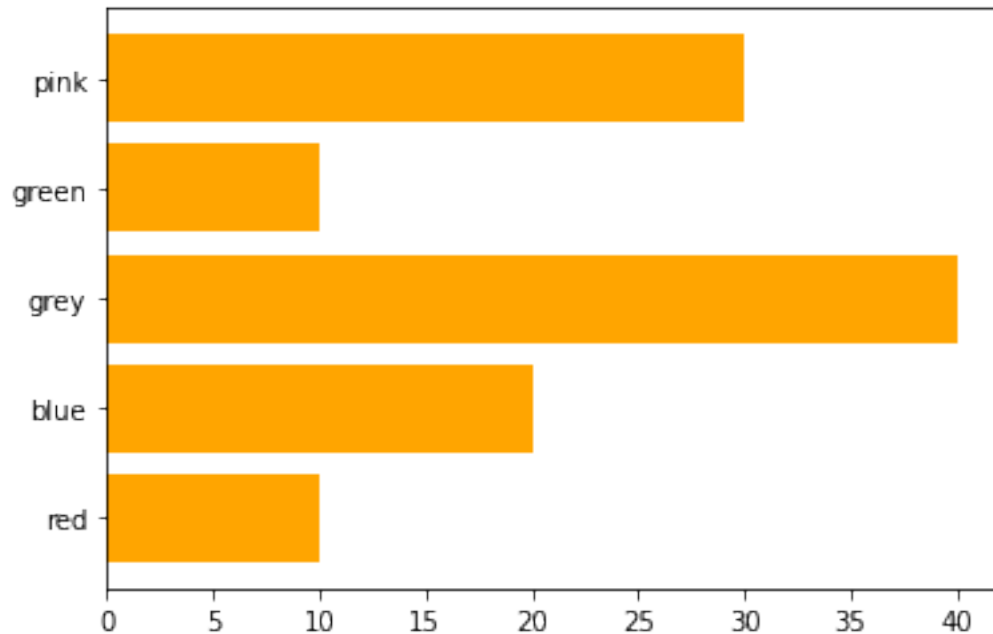
```
[221]: <BarContainer object of 5 artists>
```



Horizontal Bar chart

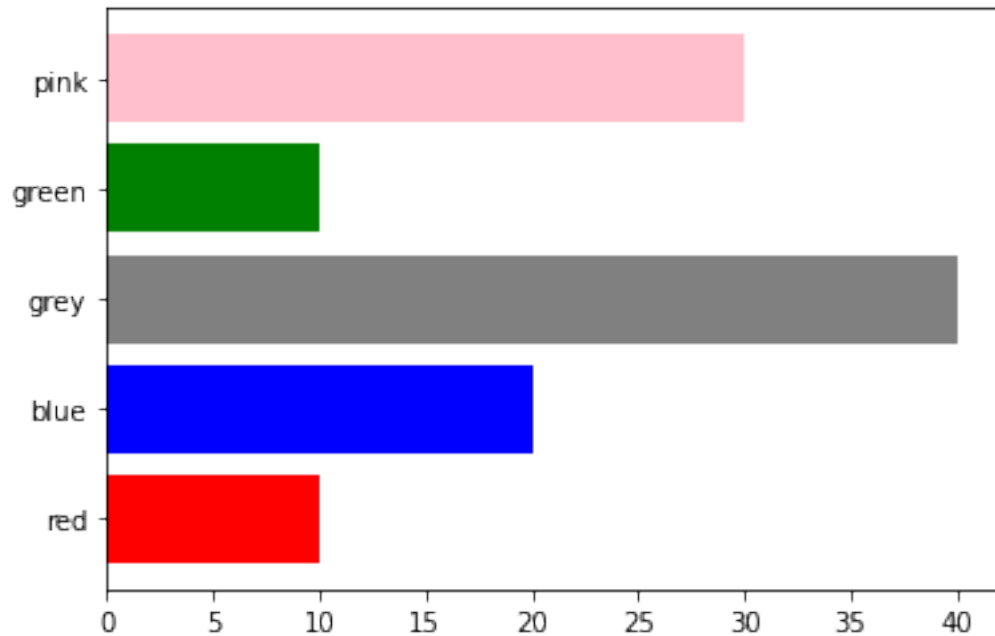
```
[222]: child=[10,20,40,10,30]  
color=['red','blue','grey','green','pink']  
plt.barh(color,child,color="orange")
```

[222]: <BarContainer object of 5 artists>



```
[223]: child=[10,20,40,10,30]  
color=['red','blue','grey','green','pink']  
plt.barh(color,child,color=['red','blue','grey','green','pink'])
```

[223]: <BarContainer object of 5 artists>



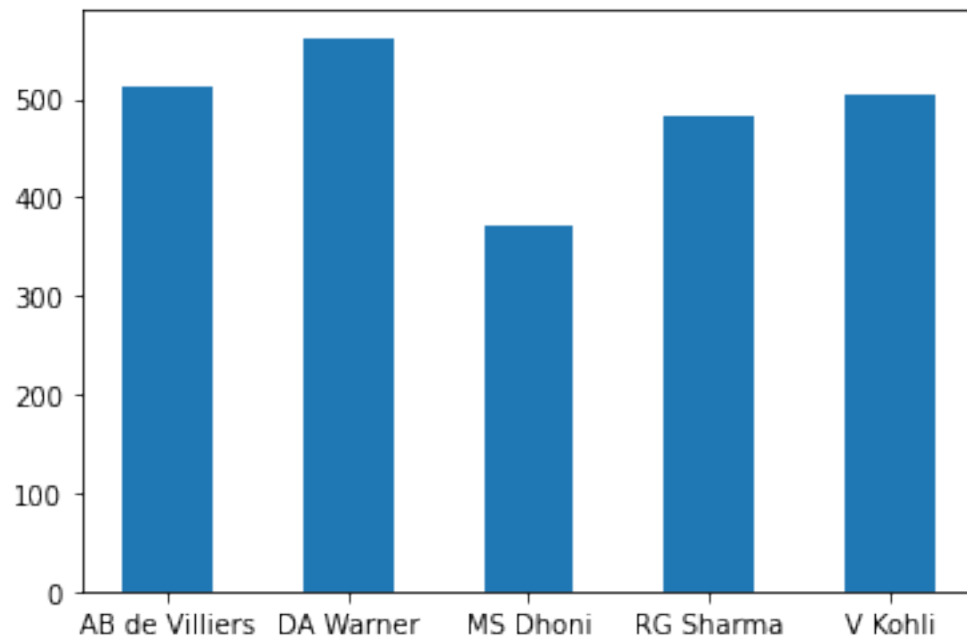
```
[224]: df=pd.read_csv("D:\\Drivers\\matplotlib\\batsman_season_record.csv")
df
```

```
[224]:
```

	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

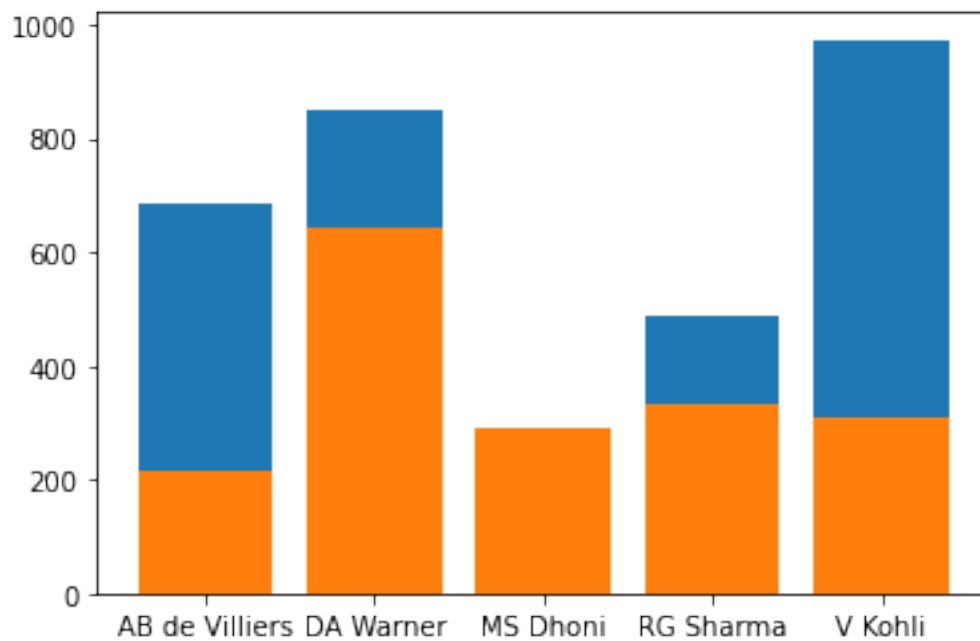
```
[225]: plt.bar(df["batsman"],df["2015"],width=0.5)
```

```
[225]: <BarContainer object of 5 artists>
```



```
[226]: plt.bar(df["batsman"],df["2016"])  
plt.bar(df["batsman"],df["2017"])
```

[226]: <BarContainer object of 5 artists>



```
[227]: np.array(df.shape[0])
```

```
[227]: array(5)
```

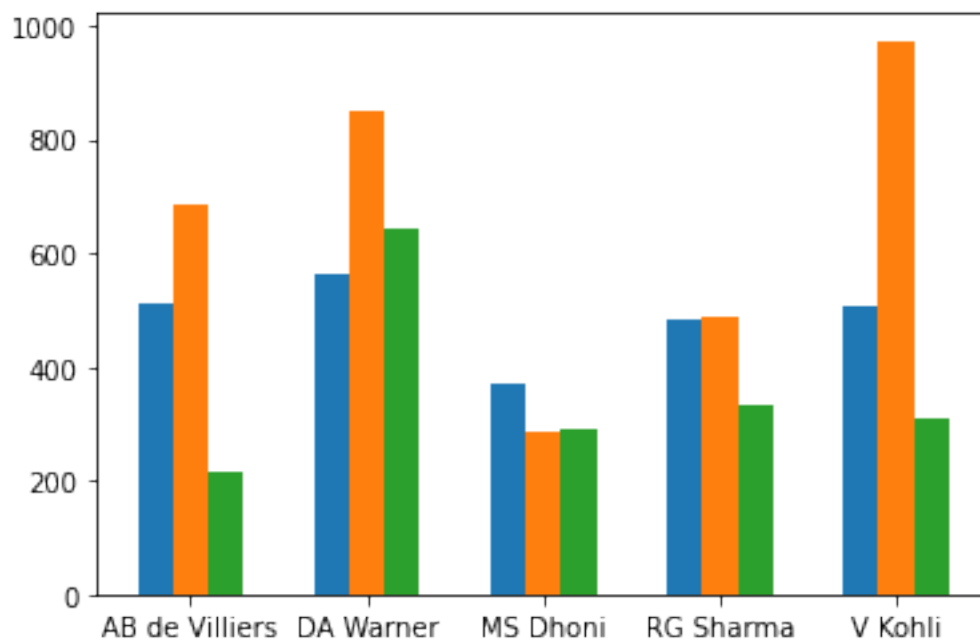
```
[228]: np.arange(df.shape[0])
```

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

4 Multiple Bar Chart's

```
[229]: plt.bar(np.arange(df.shape[0])-0.2,df['2015'],width=0.2)
plt.bar(np.arange(df.shape[0]),df['2016'],width=0.2)
plt.bar(np.arange(df.shape[0])+0.2,df['2017'],width=0.2)

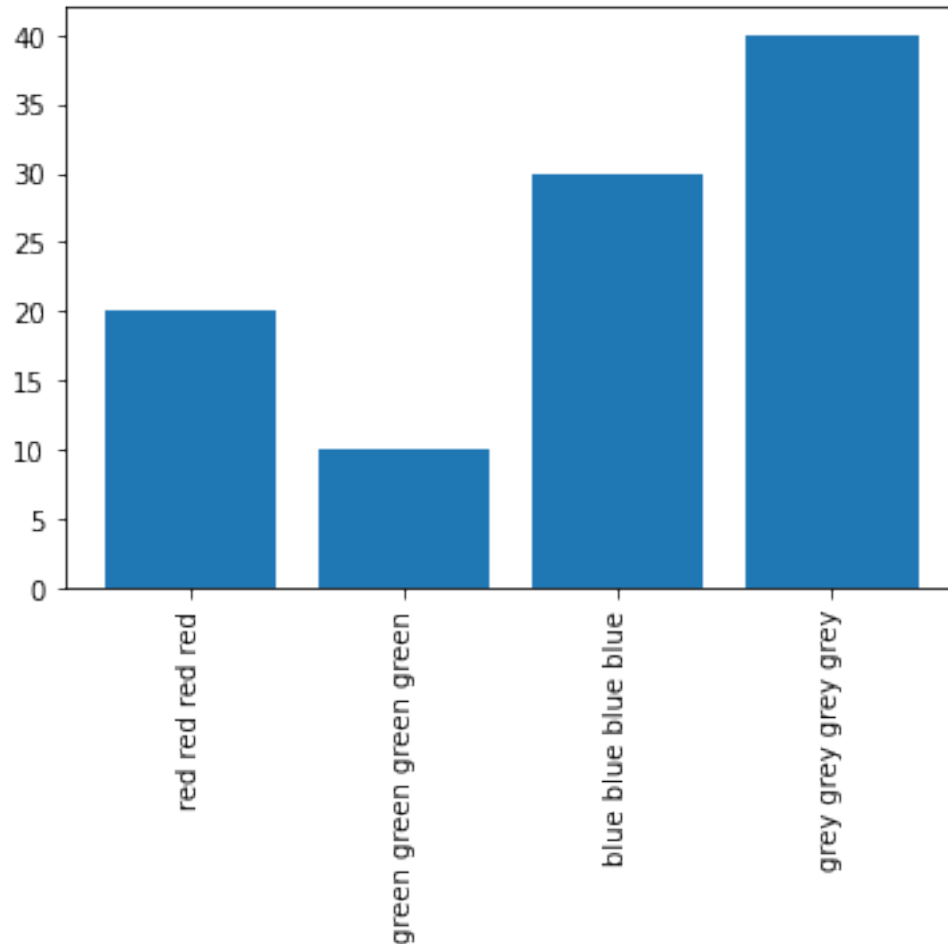
plt.xticks(np.arange(df.shape[0]),df["batsman"])
plt.show()
```



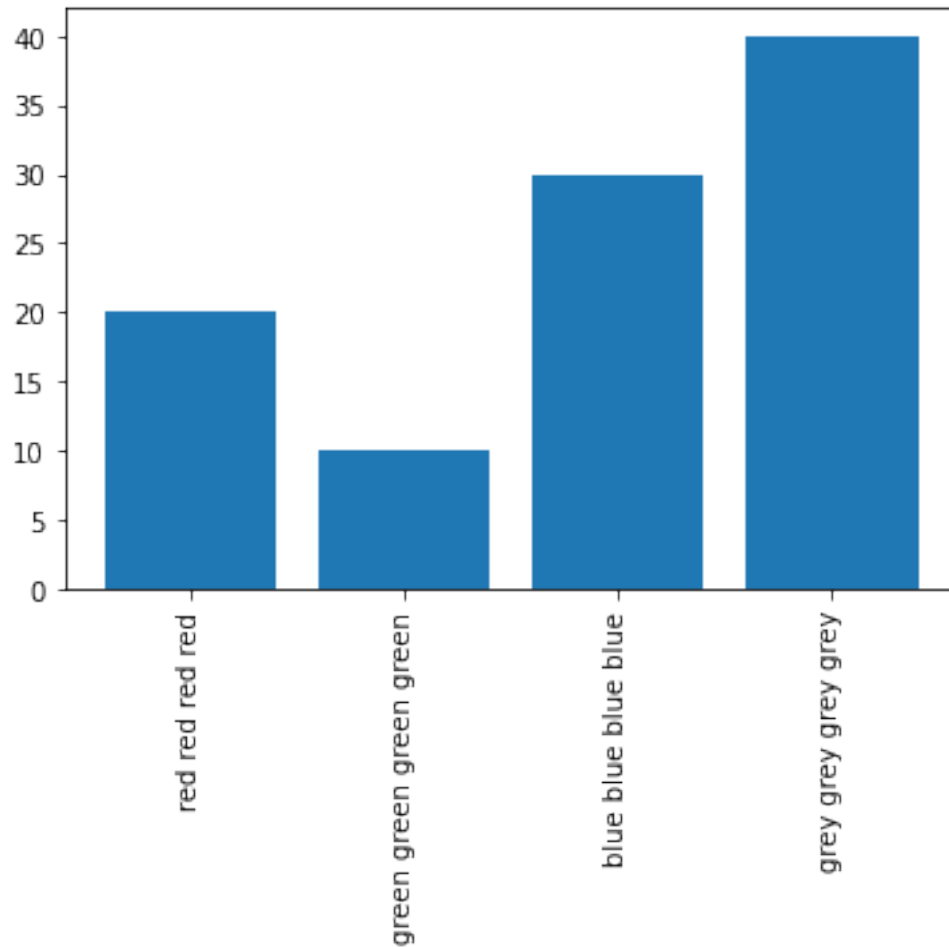
5 Overlapping names problem

```
[230]: names=[20,10,30,40]
color=["red red red red","green green green green","blue blue blue blue","grey_
↪grey grey grey"]
plt.bar(color,names)
plt.xticks(rotation="vertical")
```

```
[230]: ([0, 1, 2, 3],
[Text(0, 0, ''), Text(0, 0, ''), Text(0, 0, ''), Text(0, 0, '')])
```



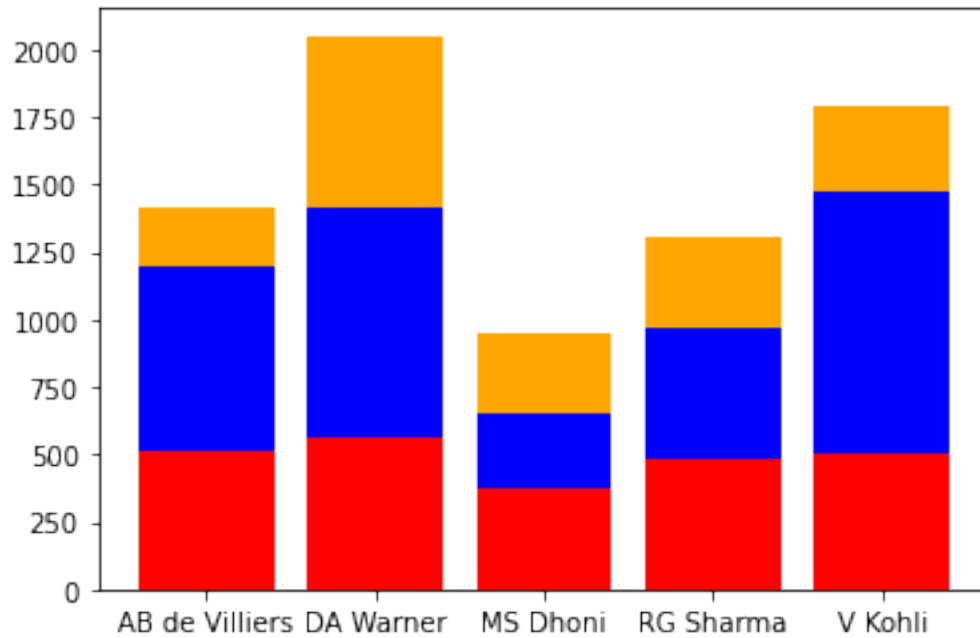
```
[231]: names=[20,10,30,40]
color=["red red red red","green green green green","blue blue blue blue","grey_
↪grey grey grey"]
plt.bar(color,names)
plt.xticks(rotation="vertical")
plt.show()
```

6 Stacked Bar Chart

```
[232]: plt.bar(df["batsman"],df["2015"],color='r')  
plt.bar(df["batsman"],df["2016"],bottom=df["2015"],color='blue')  
plt.bar(df["batsman"],df["2017"],bottom=df["2016"]+df["2015"],color='orange')
```

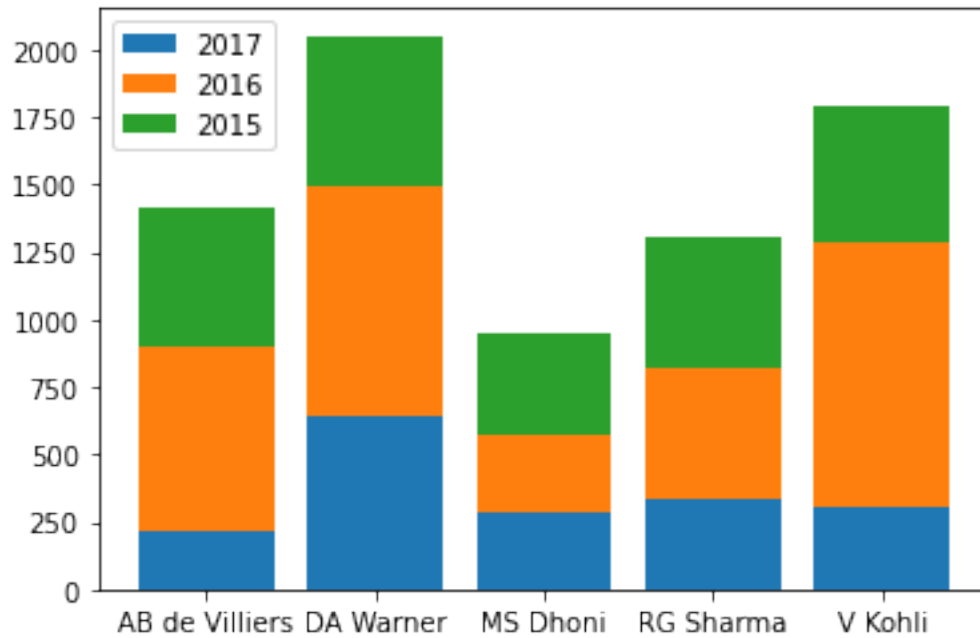
```
[232]: <BarContainer object of 5 artists>
```



7 Labeling the charts

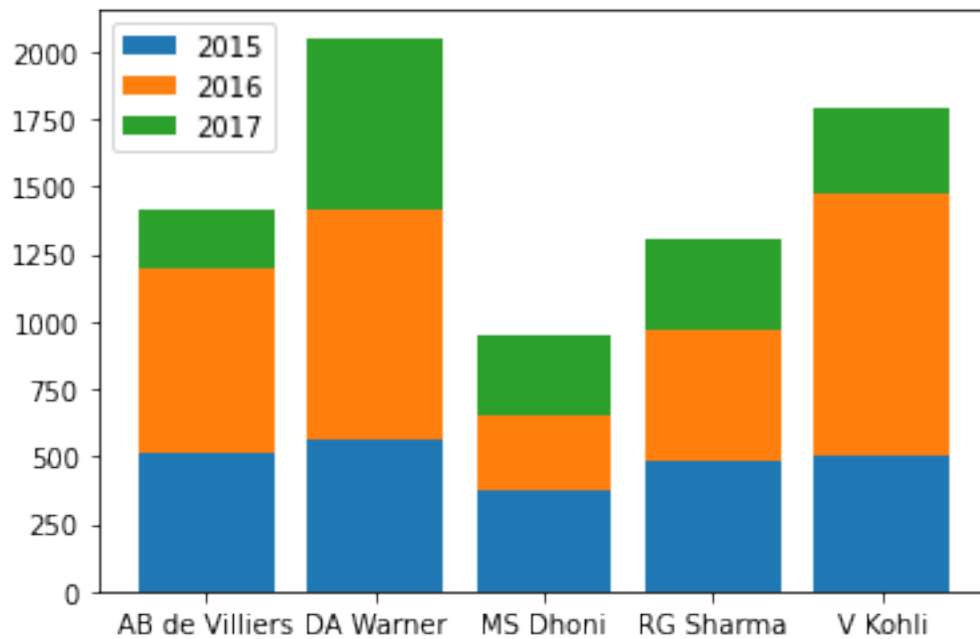
```
[233]: plt.bar(df['batsman'],df['2017'],label="2017")
plt.bar(df['batsman'],df['2016'],bottom=df['2017'],label="2016")
plt.bar(df['batsman'],df['2015'],bottom=df['2016']+df["2017"],label="2015")

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



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

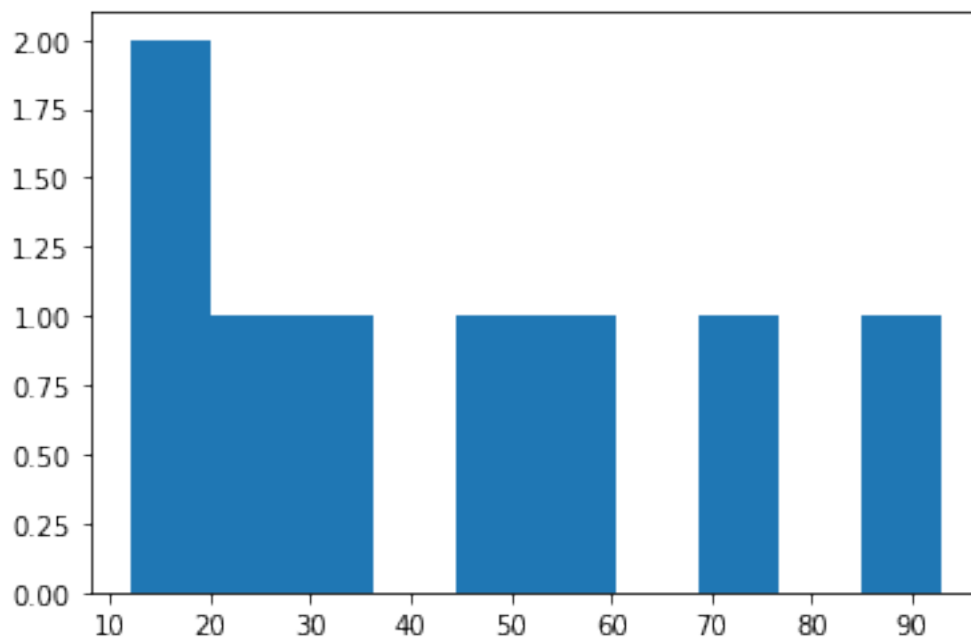
plt.legend()
plt.show()
```



8 Histogram Chart

```
[236]: number=[12,36,54,25,12,48,93,69]
plt.hist(number)
```

```
[236]: (array([2., 1., 1., 0., 1., 1., 0., 1., 0., 1.]),
array([12. , 20.1, 28.2, 36.3, 44.4, 52.5, 60.6, 68.7, 76.8, 84.9, 93. ]),
<BarContainer object of 10 artists>)
```



```
[238]: vk=pd.read_csv("D:\\Drivers\\matplotlib\\vk.csv")
vk
```

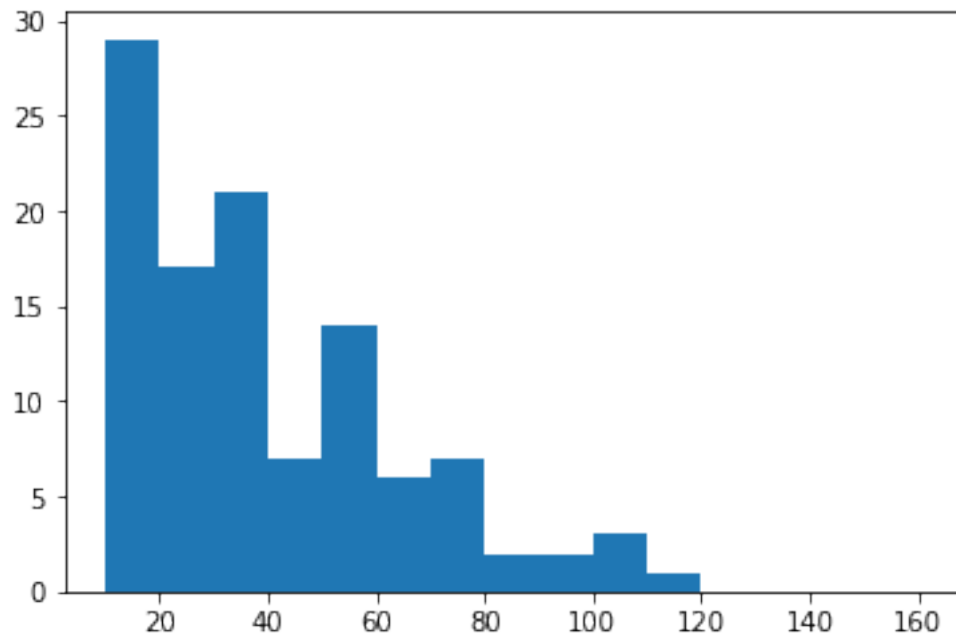
```
[238]:
```

	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]
```

```
[242]: plt.
        ↪hist(vk["batsman_runs"],bins=[10,20,30,40,50,60,70,80,90,100,110,120,140,160])
        plt.show()
```

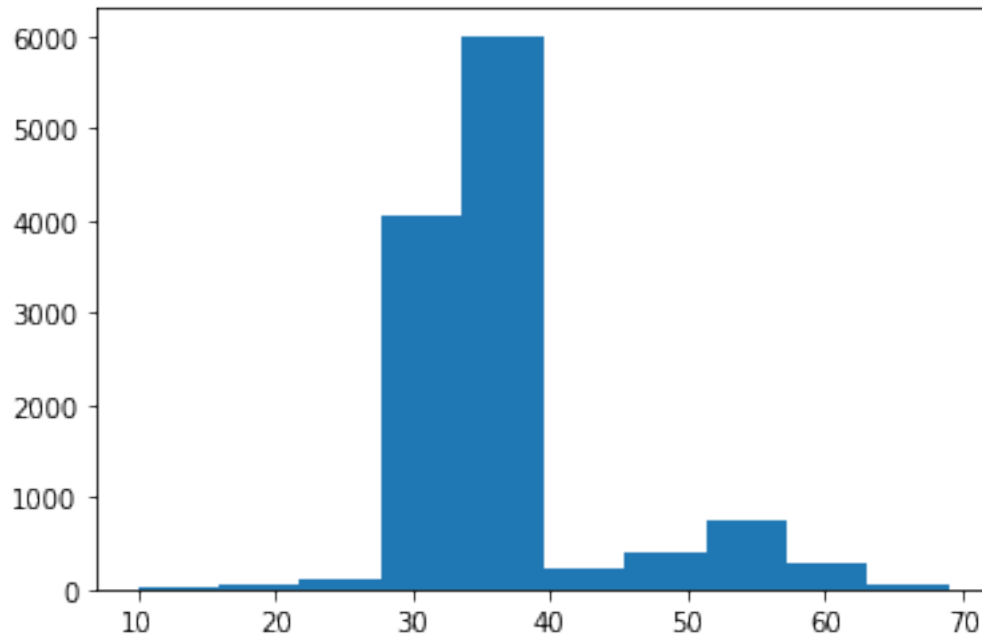


```
[244]: arr=np.load("D:\\Drivers\\matplotlib\\big-array.npy")
```

```
[245]: arr.shape
```

```
[245]: (11949,)
```

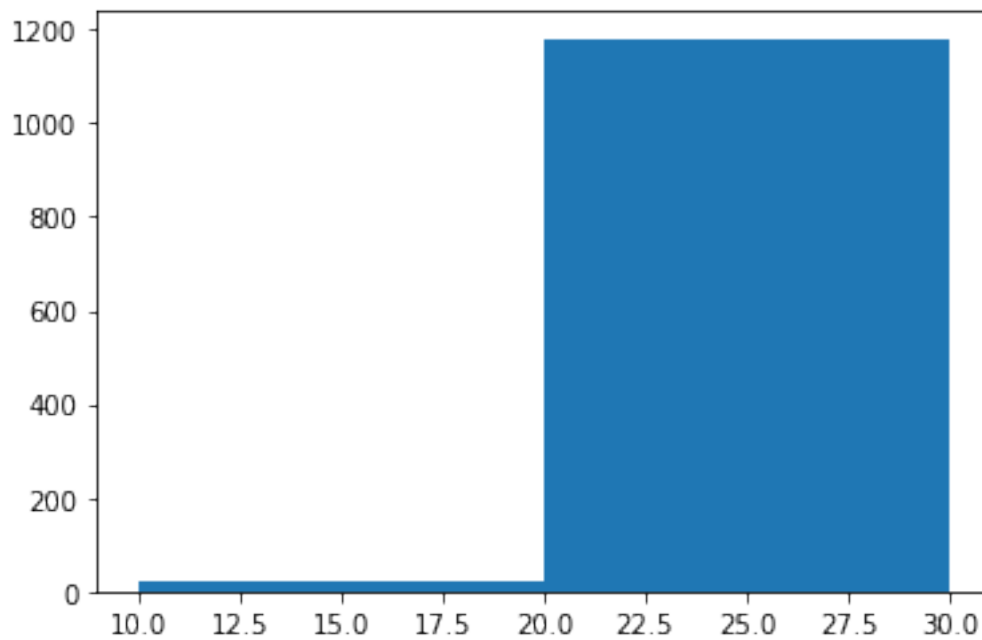
```
[247]: plt.hist(arr)
        plt.show()
```



We Can Trim the histogram

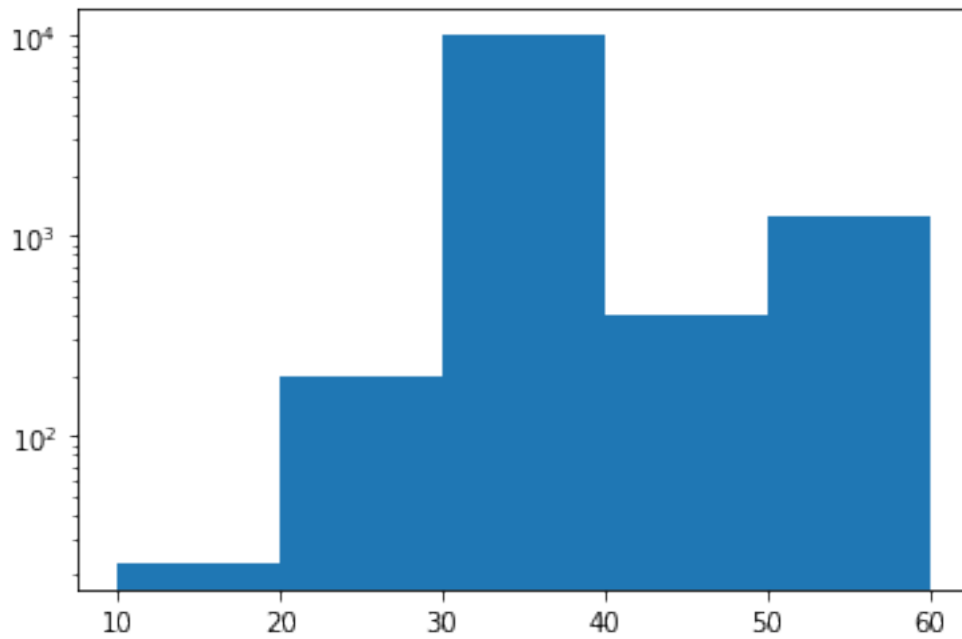
```
[250]: plt.hist(arr, bins=[10,20,30])
```

```
[250]: (array([ 23., 1180.]),
        array([10, 20, 30]),
        <BarContainer object of 2 artists>)
```



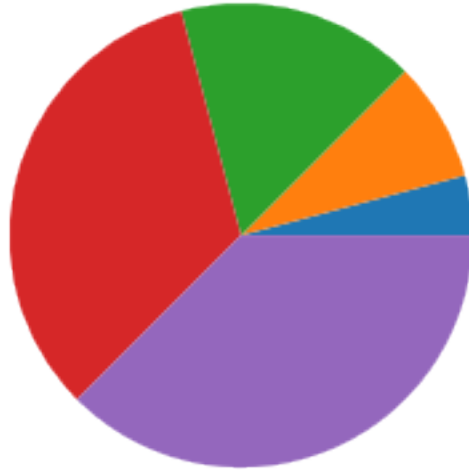
Logarithmic Scale

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



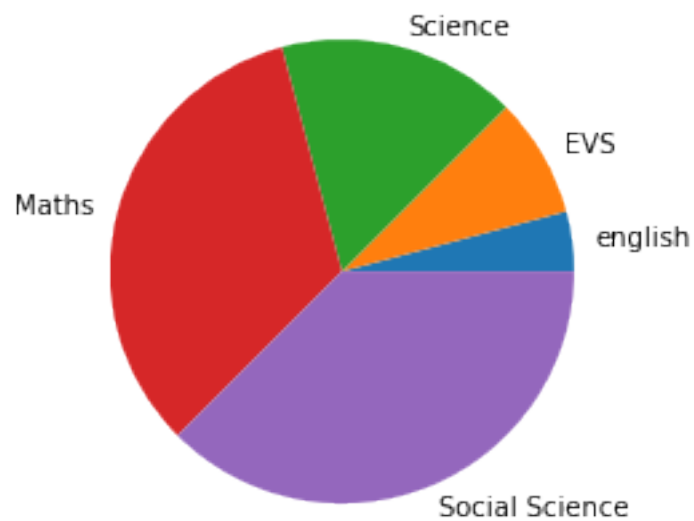
9 Pie Chart

```
[257]: data=[10,20,40,80,90]  
plt.pie(data)  
plt.show()
```



Give the labels for perticular each part

```
[259]: data=[10,20,40,80,90]  
subjects=["english","EVS","Science","Maths","Social Science"]  
plt.pie(data,labels=subjects)  
plt.show()
```

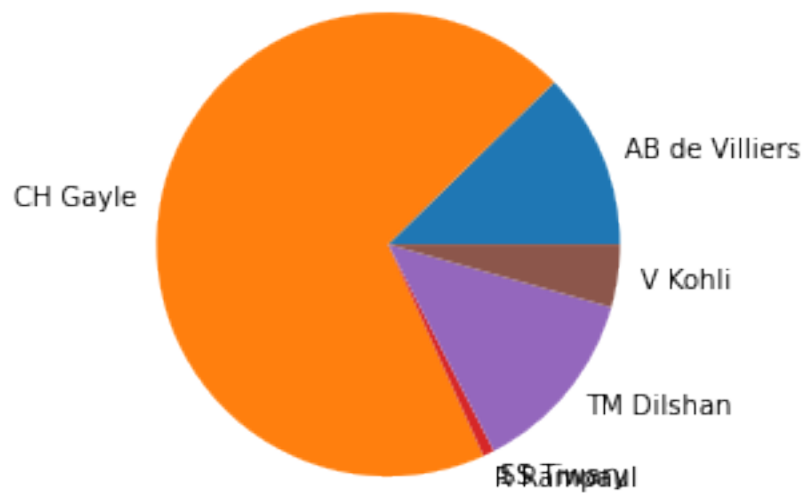



```
[261]: gell=pd.read_csv("D:\\Drivers\\matplotlib\\gayle-175.csv")
gell
```

```
[261]:
```

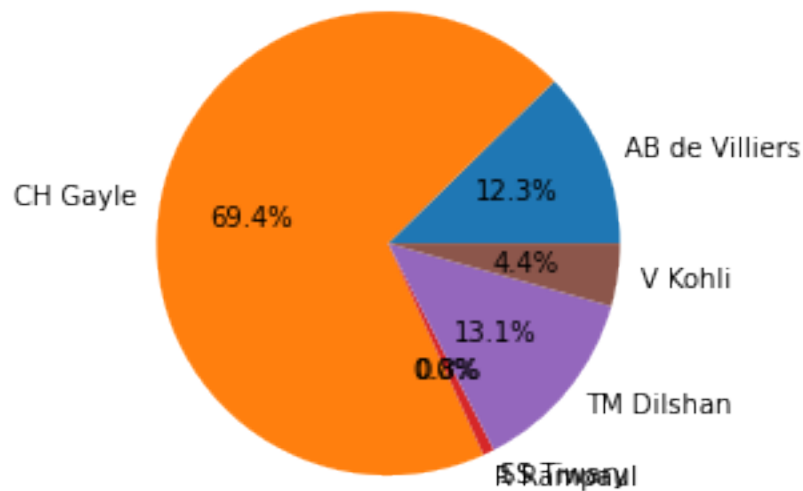
	batsman	batsman_runs
0	AB de Villiers	31
1	CH Gayle	175
2	R Rampaul	0
3	SS Tiwary	2
4	TM Dilshan	33
5	V Kohli	11

```
[264]: plt.pie(gell["batsman_runs"],labels=gell["batsman"])
plt.show()
```

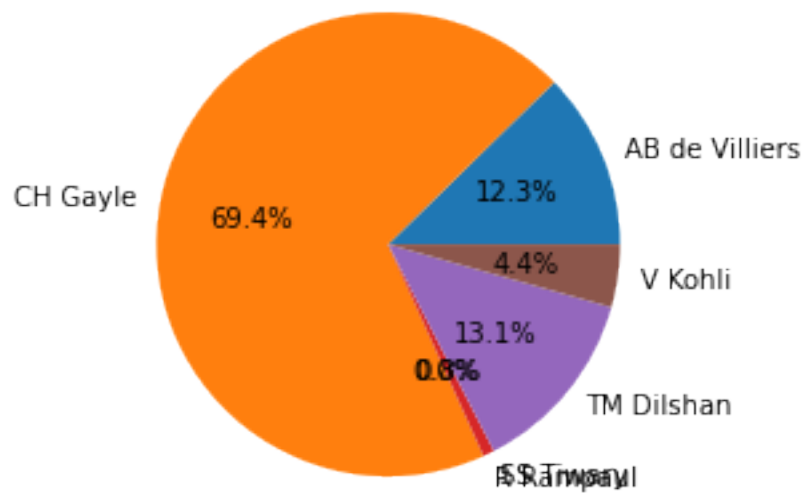


Autopct feature of the pie charts

```
[268]: plt.pie(gell["batsman_runs"],labels=gell["batsman"],autopct="%0.1f%%")
plt.show()
```

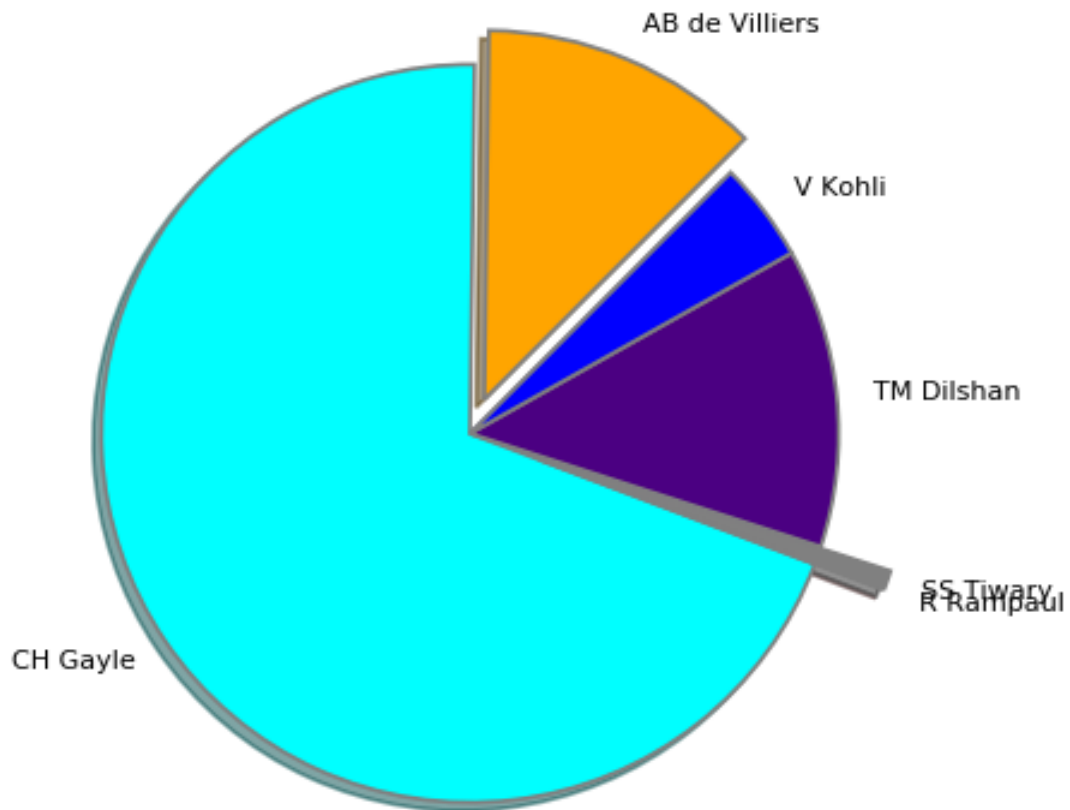


```
[269]: plt.pie(gell["batsman_runs"], labels=gell["batsman"], autopct="%0.1f%%")
plt.show()
```

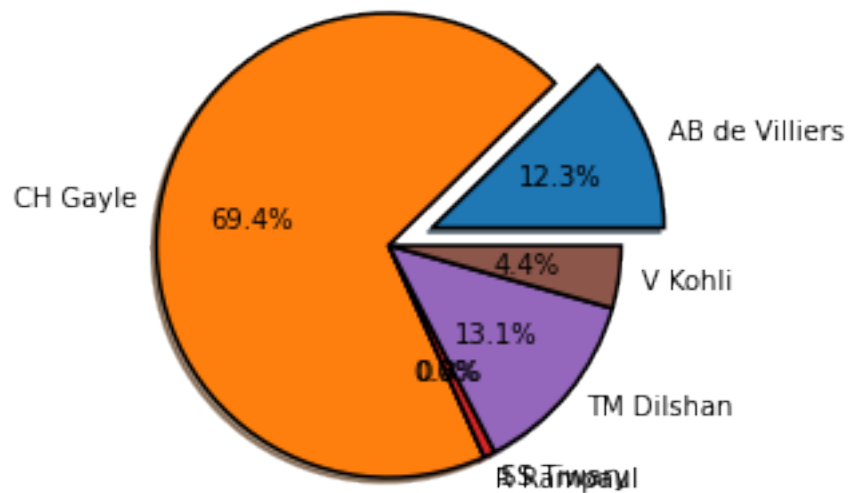


```
[328]: wp = { 'linewidth' : 1.5, 'edgecolor' : "grey" }
explode = (0.1, 0.0, 0.2, 0.2, 0.0, 0.0)
colors = ( "orange", "cyan", "brown",
           "grey", "indigo", "blue")
```

```
plt.pie(gell["batsman_runs"],labels = gell["batsman"],explode = explode,shadow_
    ↪ = True,colors = colors,
        startangle = 45,
        wedgeprops = wp,
        pctdistance=0.85)
plt.show()
```



```
[297]: plt.pie(gell["batsman_runs"],labels=gell["batsman"],autopct="%0.
    ↪ 1f%%",shadow=True,
        wedgeprops={"linewidth":1.5,"edgecolor":"black"},explode=[0.
    ↪ 2,0,0,0,0,0])
plt.show()
```



10 Changing Style of the plots

```
[305]: plt.style.available
```

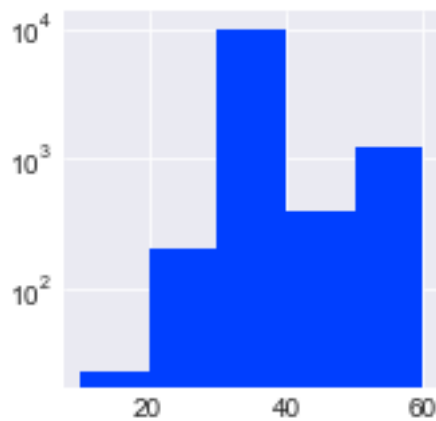
```
[305]: ['Solarize_Light2',
        '_classic_test_patch',
        '_mpl-gallery',
        '_mpl-gallery-nogrid',
        'bmh',
        'classic',
        'dark_background',
        'fast',
        'fivethirtyeight',
        'ggplot',
        'grayscale',
        'seaborn',
        'seaborn-bright',
        'seaborn-colorblind',
        'seaborn-dark',
        'seaborn-dark-palette',
        'seaborn-darkgrid',
        'seaborn-deep',
        'seaborn-muted',
        'seaborn-notebook',
        'seaborn-paper',
        'seaborn-pastel',
```

```
'seaborn-poster',  
'seaborn-talk',  
'seaborn-ticks',  
'seaborn-white',  
'seaborn-whitegrid',  
'tableau-colorblind10']
```

```
[309]: plt.style.use('tableau-colorblind10')
```

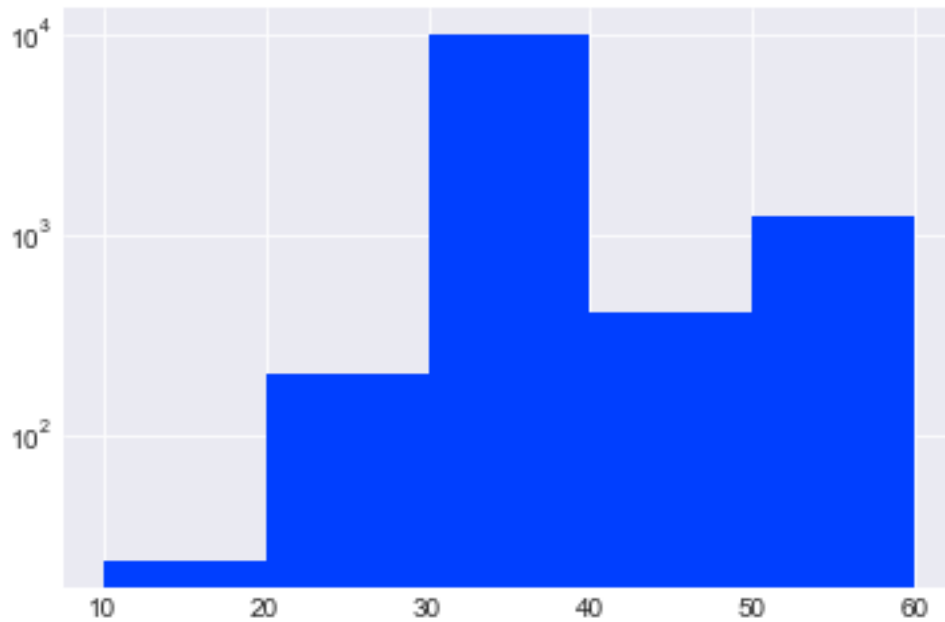
```
[321]: plt.style.use("seaborn-darkgrid")
```

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



```
[313]: plt.style.use("seaborn-bright")
```

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



```
[323]: plt.style.use("deafault")
```

```
-----
FileNotFoundError                                Traceback (most recent call last)
File ~\python3.10\lib\site-packages\matplotlib\style\core.py:127, in use(style)
    126 try:
--> 127     rc = rc_params_from_file(style, use_default_template=False)
    128     _apply_style(rc)

File ~\python3.10\lib\site-packages\matplotlib\__init__.py:852, in _rc_params_from_file(fname, fail_on_error, use_default_template)
    838 """
    839 Construct a `RcParams` from file *fname*.
    840
    (...)
    850     parameters specified in the file. (Useful for updating dicts.)
    851 """
--> 852 config_from_file = _rc_params_in_file(fname, fail_on_error=fail_on_error)
    854 if not use_default_template:

File ~\python3.10\lib\site-packages\matplotlib\__init__.py:778, in _rc_params_in_file(fname, transform, fail_on_error)
    777 rc_temp = {}
--> 778 with open_file_or_url(fname) as fd:
    779     try:
```

```
File ~\python3.10\lib\contextlib.py:135, in _GeneratorContextManager.
```

```
    ↪ __enter__(self)
      134 try:
--> 135     return next(self.gen)
      136 except StopIteration:
```

```
File ~\python3.10\lib\site-packages\matplotlib\__init__.py:755, in _
```

```
    ↪ _open_file_or_url(fname)
      754     encoding = "utf-8"
--> 755 with open(fname, encoding=encoding) as f:
      756     yield f
```

```
FileNotFoundError: [Errno 2] No such file or directory: 'deafault'
```

The above exception was the direct cause of the following exception:

```
OSError                                Traceback (most recent call last)
```

```
Input In [323], in <module>
```

```
----> 1 plt.style.use("deafault")
```

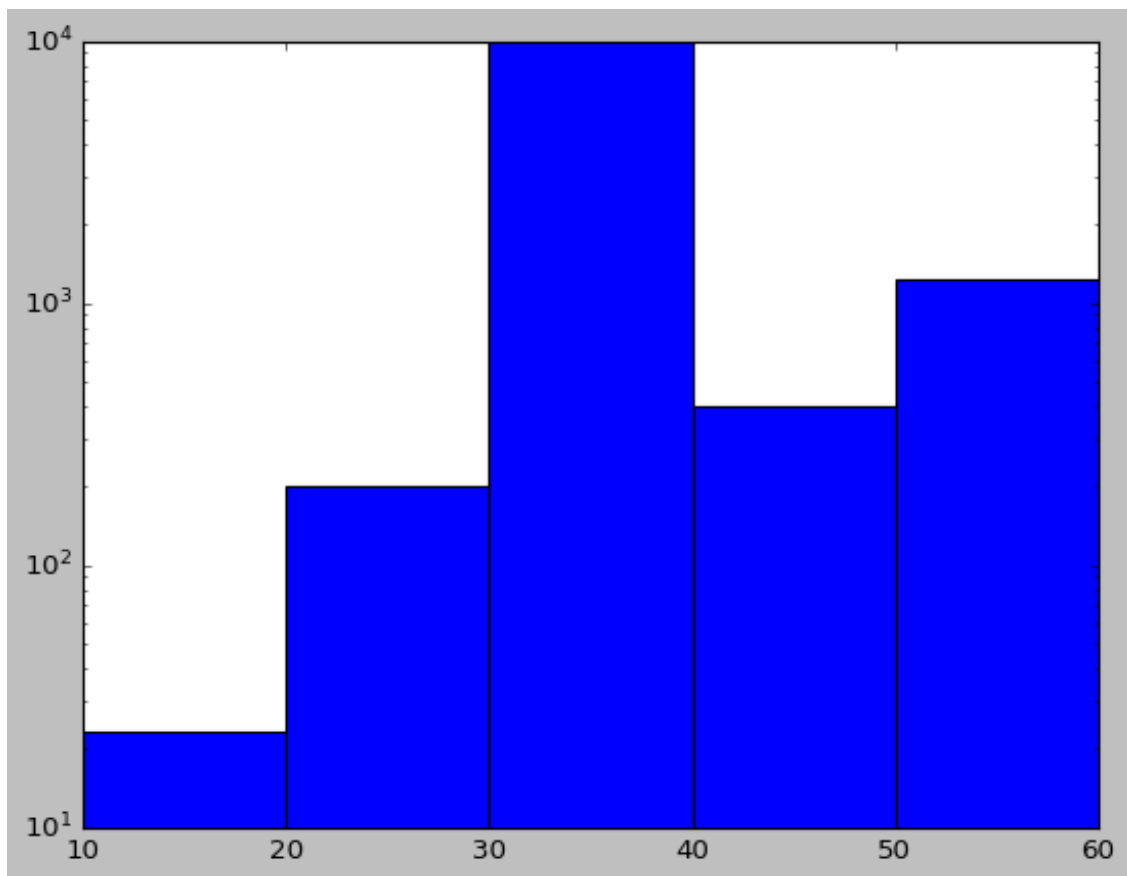
```
File ~\python3.10\lib\site-packages\matplotlib\style\core.py:130, in use(style)
```

```
    128     _apply_style(rc)
    129 except IOError as err:
--> 130     raise IOError(
    131         "{!r} not found in the style library and input is not a "
    132         "valid URL or path; see `style.available` for list of "
    133         "available styles".format(style)) from err
```

```
OSError: 'deafault' not found in the style library and input is not a valid URL,
    ↪ or path; see `style.available` for list of available styles
```

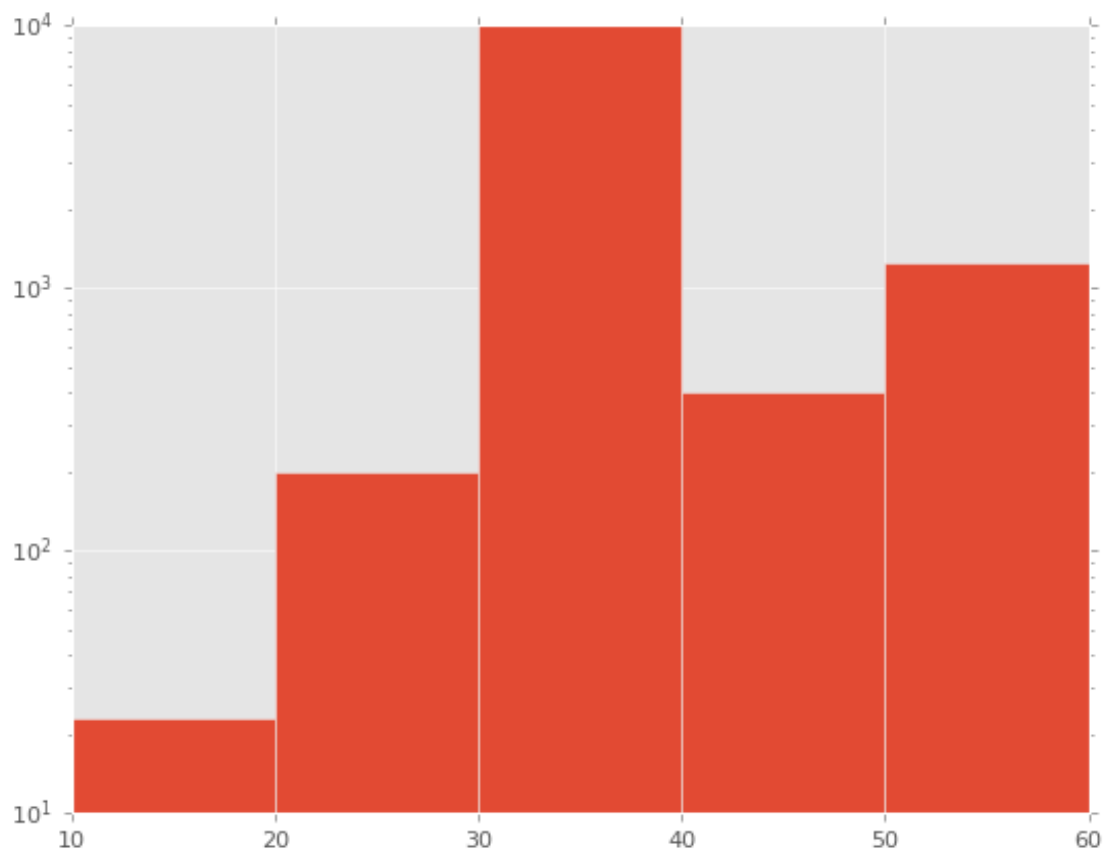
```
[345]: plt.style.use('classic')
```

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



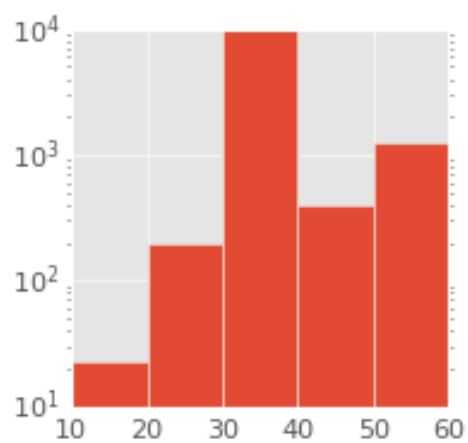
```
[335]: plt.style.use('ggplot')
```

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

```
[337]: plt.style.use('_mpl-gallery')
```

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

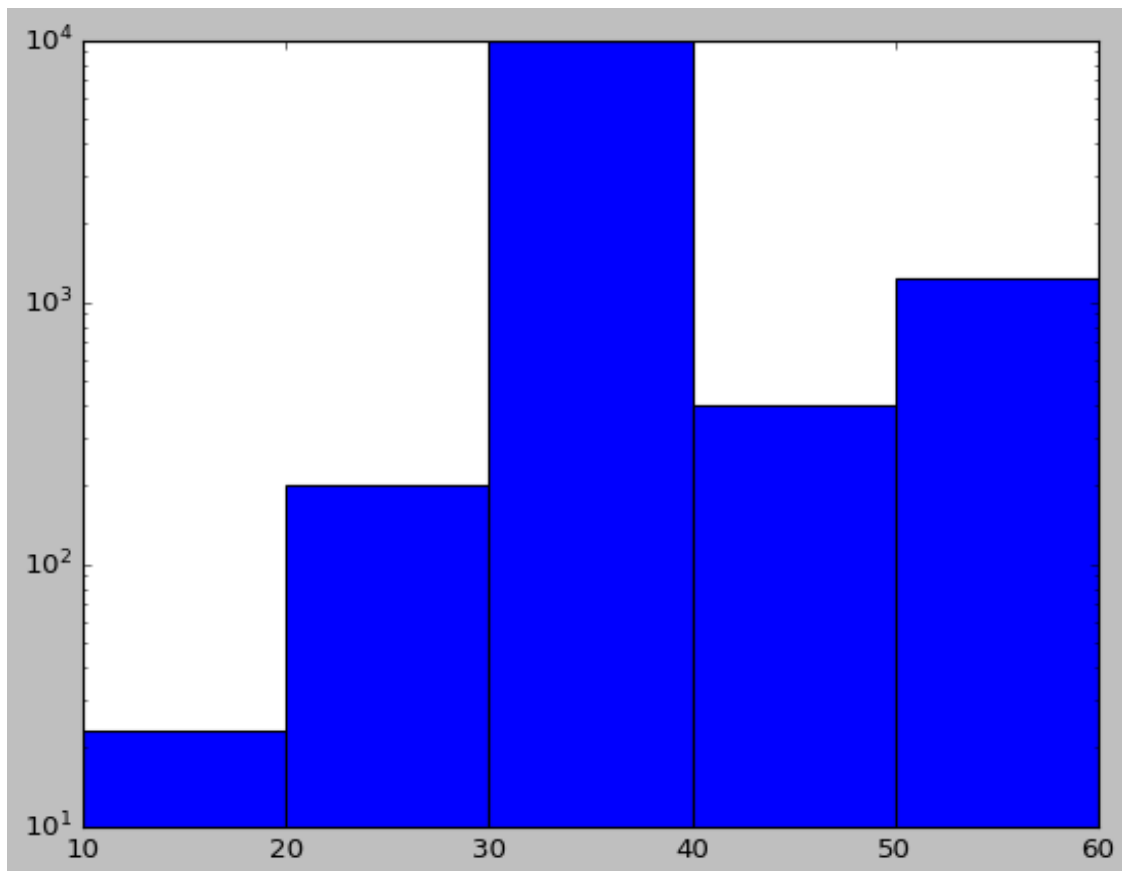


11 Save and share the plot

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

#this function save this plots in your current dict
#but condistion don't use "plt.show()" function

plt.savefig("plot.png")
```



```
[ ]:
```