Matplotlib

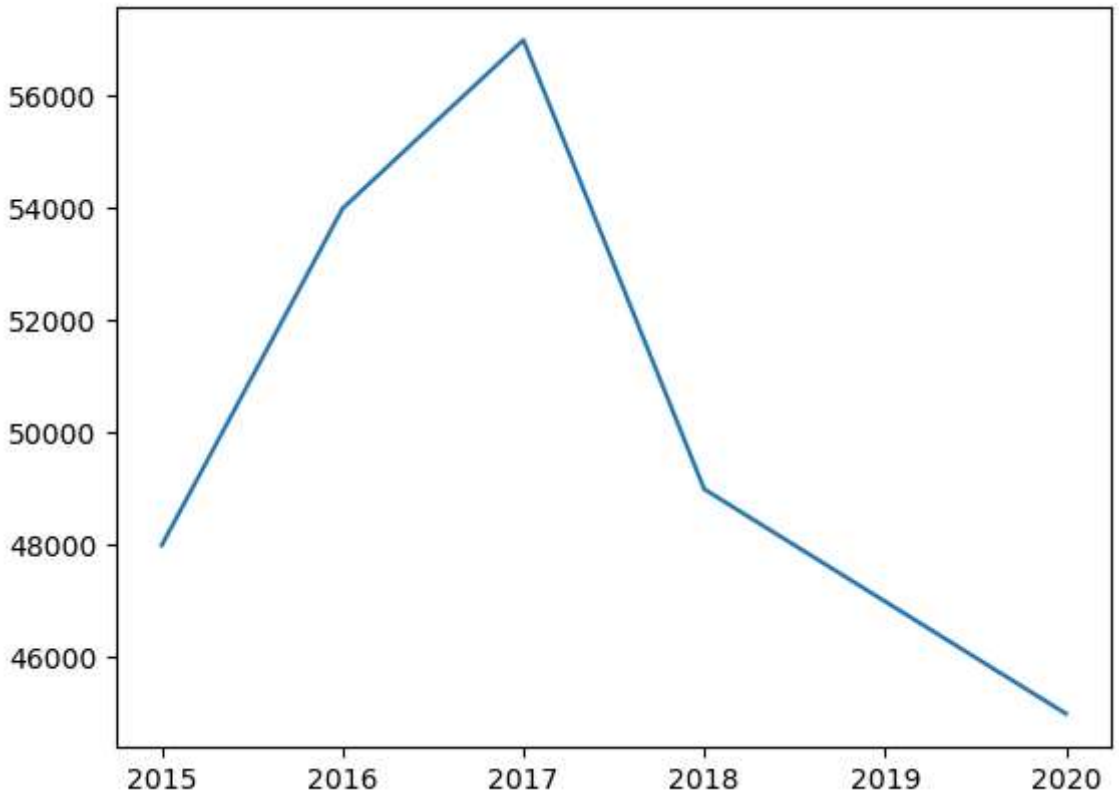
In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
plt.style.use('default')
```

2D Line plot

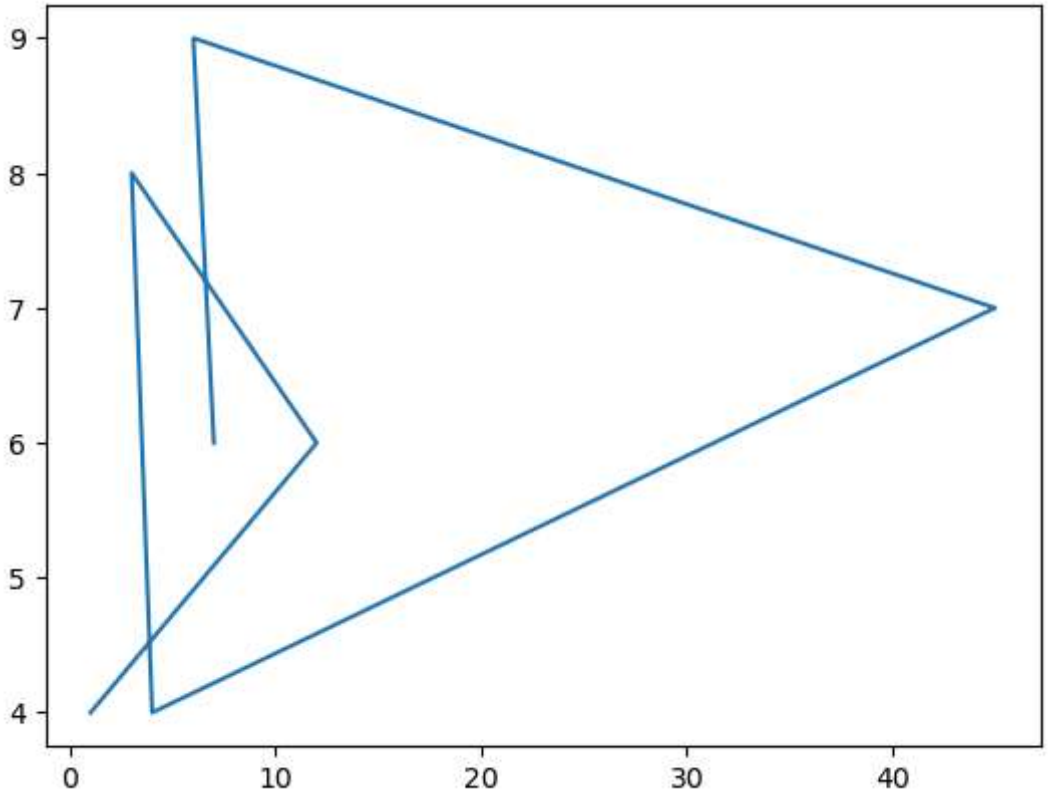
In [ ]:
```
price = [48000,54000,57000,49000,47000,45000]
year = [2015,2016,2017,2018,2019,2020]

plt.plot(year,price)
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x17f3553fe10>]
```



In [ ]:
```
x=[1,12,3,4,45,6,7]
y=[4,6,8,4,7,9,6]
plt.plot(x,y)
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x17f3aa14c90>]
```



In [ ]:
```
batsman = pd.read_csv('C:/Users/Mohan/Downloads/Matplotlib/sharma-kohli.csv')
batsman
```

Out[ ]:

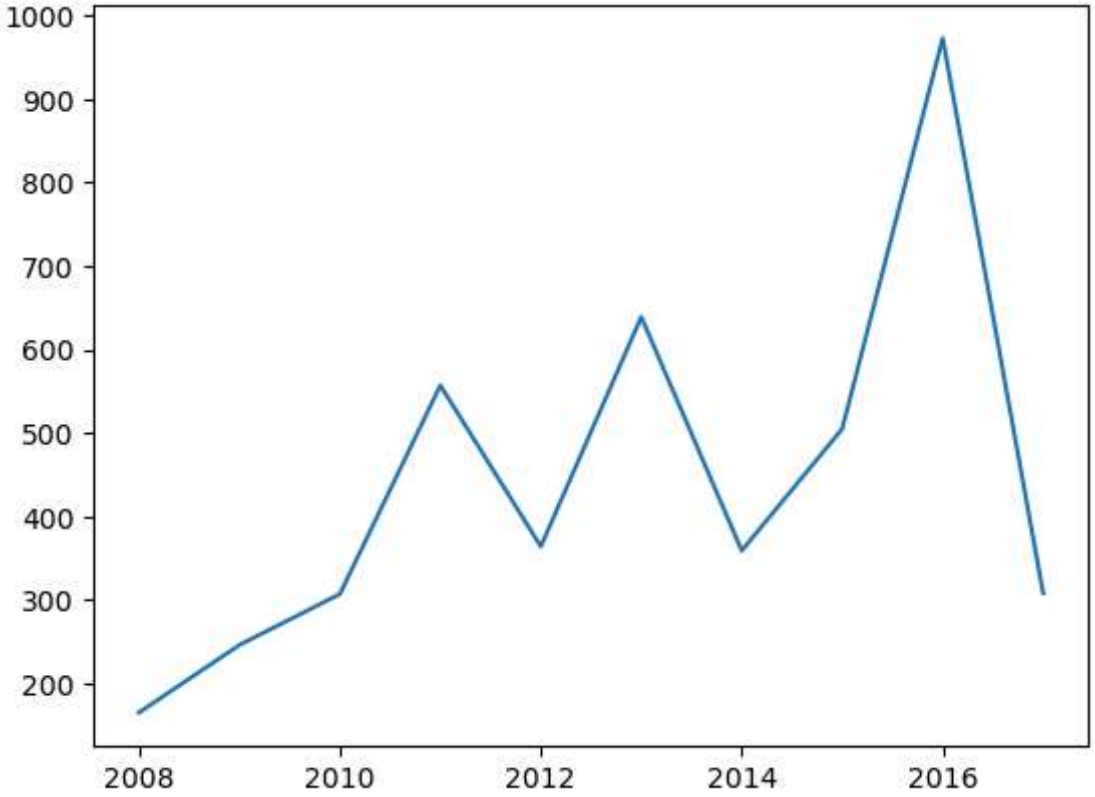
	index	RG Sharma	V Kohli
0	2008	404	165
1	2009	362	246
2	2010	404	307
3	2011	372	557
4	2012	433	364
5	2013	538	639
6	2014	390	359
7	2015	482	505
8	2016	489	973
9	2017	333	308

making graph from a csv file

pass column name to x and y

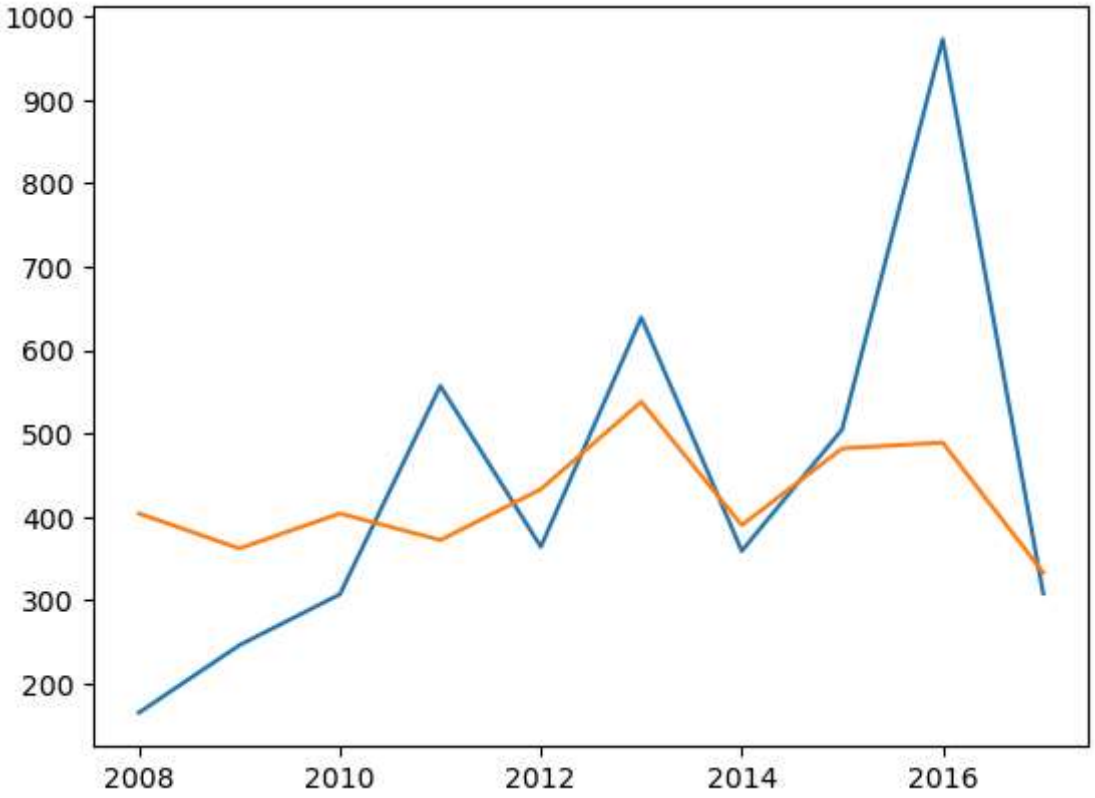
In [ ]:
```
plt.plot(batsman['index'],batsman['V Kohli'])
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x17f353b7d50>]
```



In [ ]:
```
plt.plot(batsman['index'],batsman['V Kohli'])
plt.plot(batsman['index'],batsman['RG Sharma'])
```

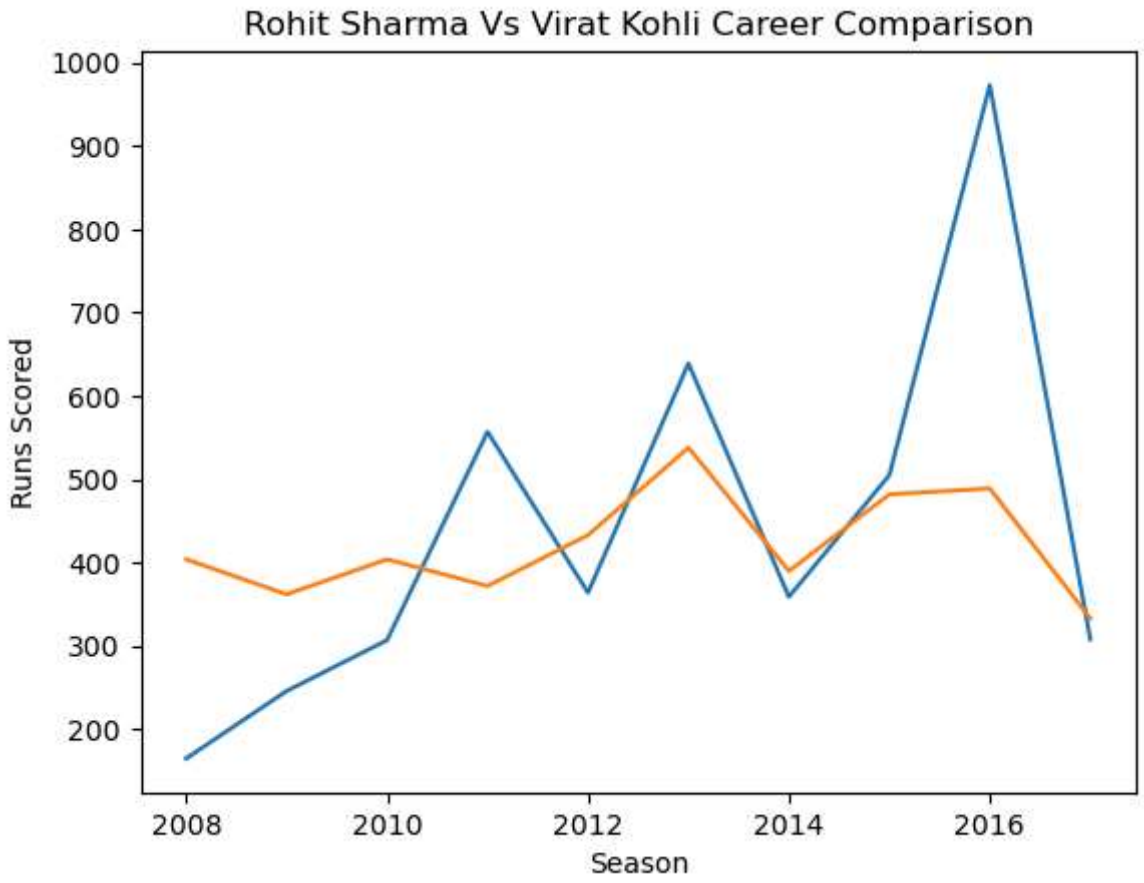
Out[ ]:
```
[<matplotlib.lines.Line2D at 0x17f3aada190>]
```



In [ ]:
```
plt.plot(batsman['index'],batsman['V Kohli'])
plt.plot(batsman['index'],batsman['RG Sharma'])

plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
```

```
plt.xlabel('Season')
plt.ylabel('Runs Scored')
```

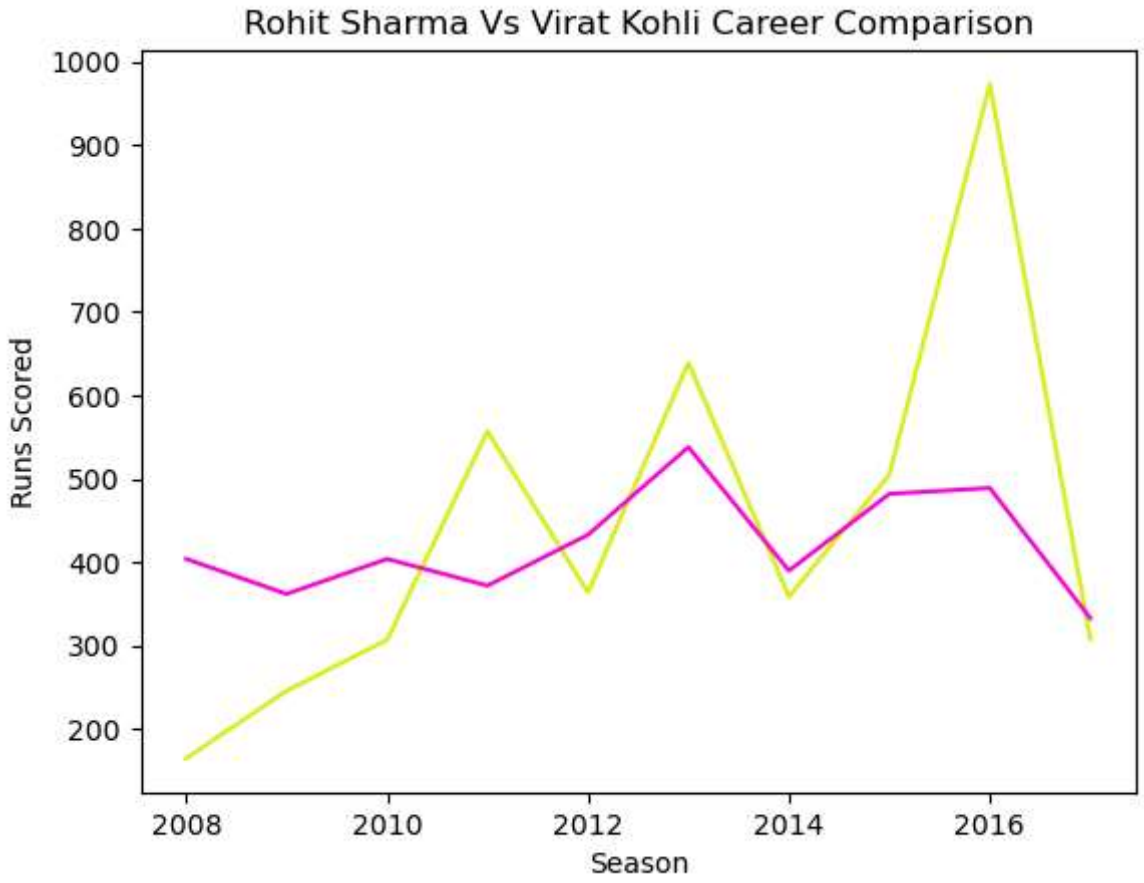
Out[ ]: Text(0, 0.5, 'Runs Scored')



lets change the color for the two lines

In [ ]:
```
# colors(hex) and line(width and style) and marker(size)
plt.plot(batsman['index'],batsman['V Kohli'],color='#D9F10F')
plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6')

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

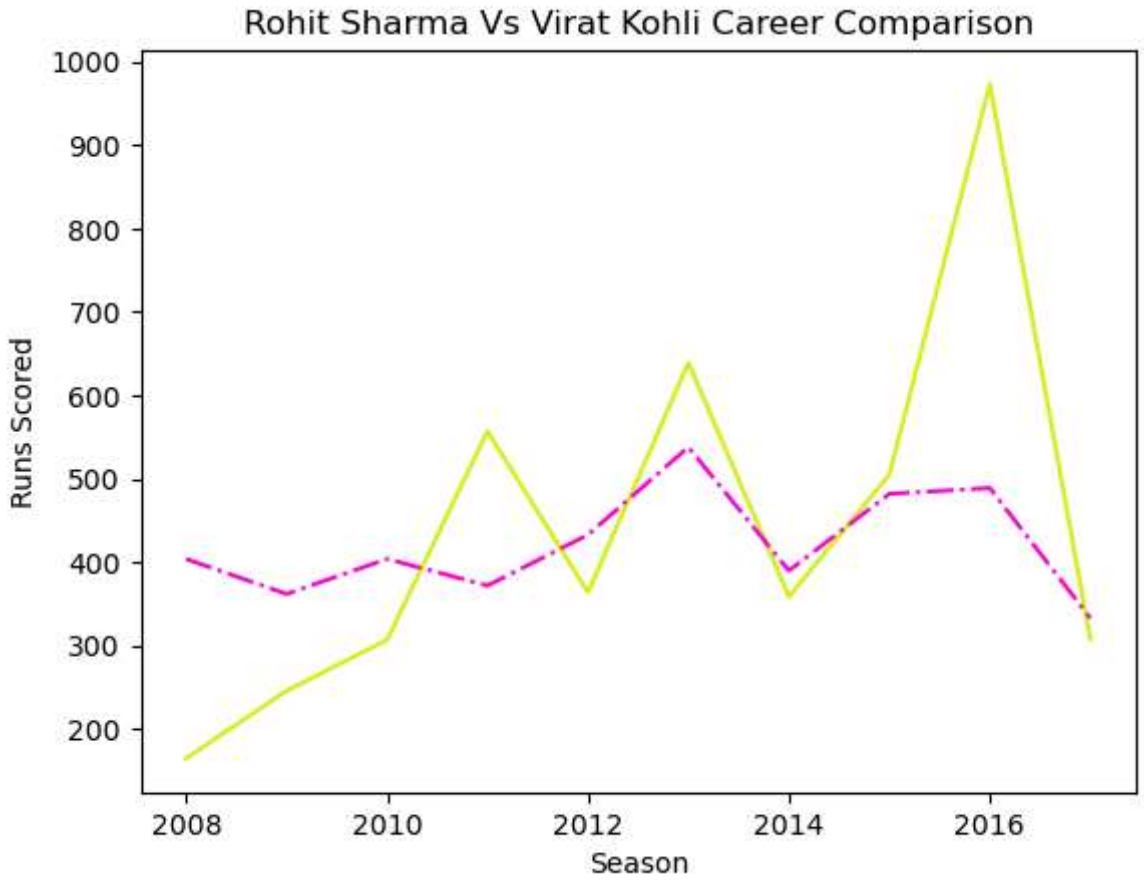
Out[ ]: Text(0, 0.5, 'Runs Scored')



lets change the line style

In [ ]:
```
plt.plot(batsman['index'],batsman['V Kohli'],color='#D9F10F',linestyle='solid')
plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6',linestyle='dashdot'

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

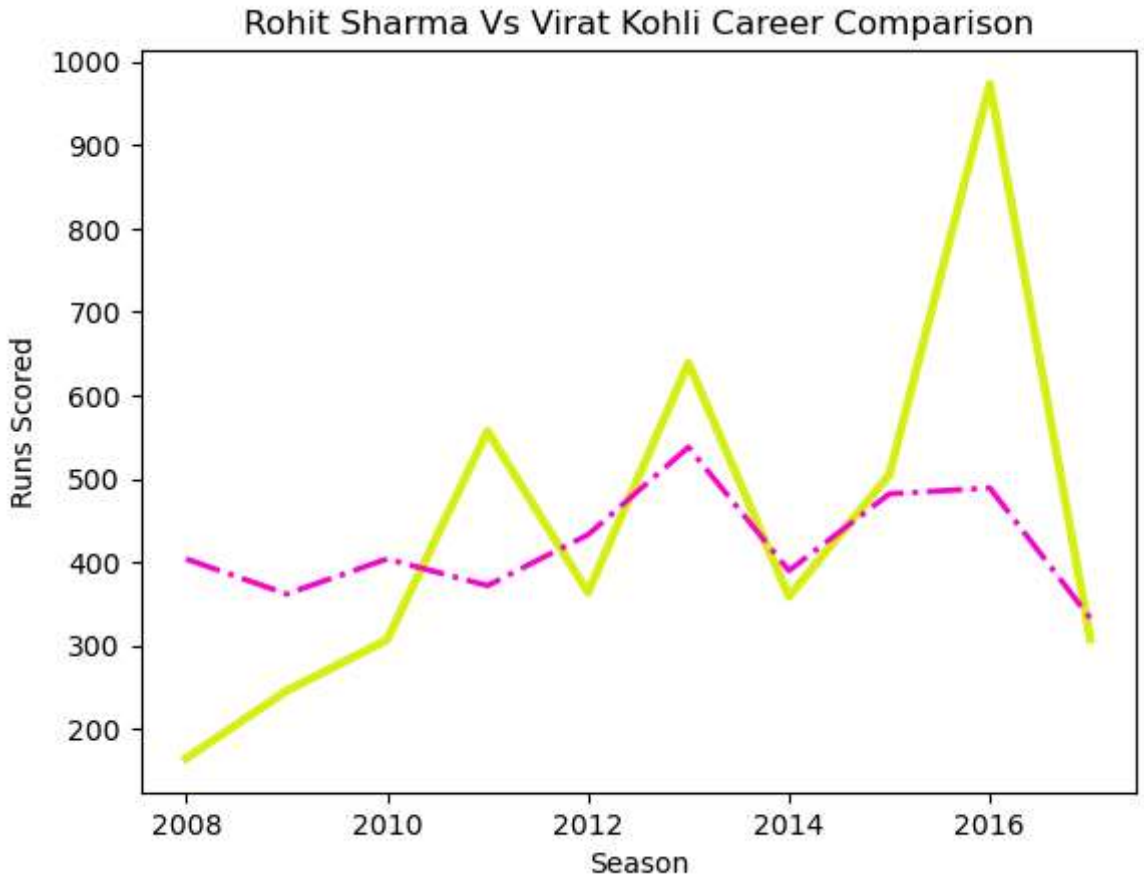
Out[ ]:
```
Text(0, 0.5, 'Runs Scored')
```



lets change line width

In [ ]:
```
plt.plot(batsman['index'],batsman['V Kohli'],color='#D9F10F',linestyle='solid',lir
plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6',linestyle='dashdot'

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

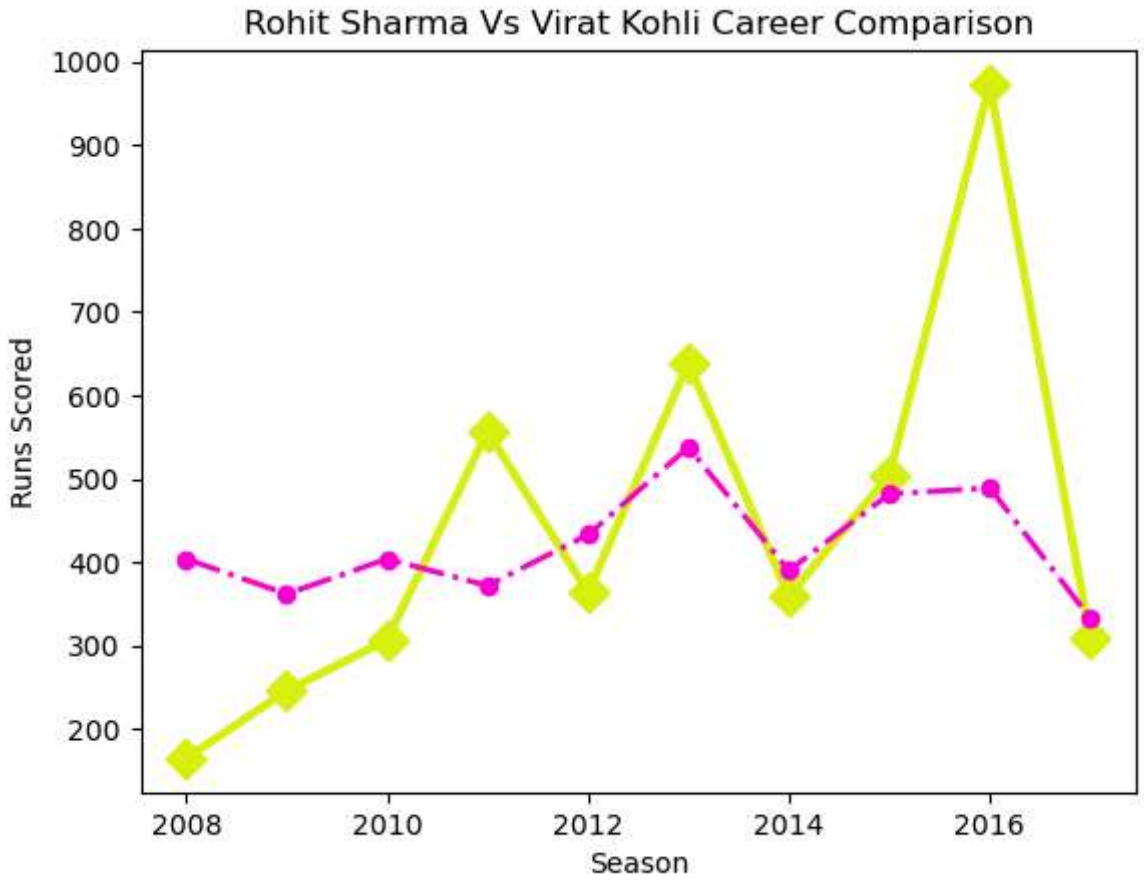
Out[ ]:
```
Text(0, 0.5, 'Runs Scored')
```



marks the coordinates

In [ ]:
```
plt.plot(batsman['index'],batsman['V Kohli'],color='#D9F10F',linestyle='solid',lir
plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6',linestyle='dashdot'

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

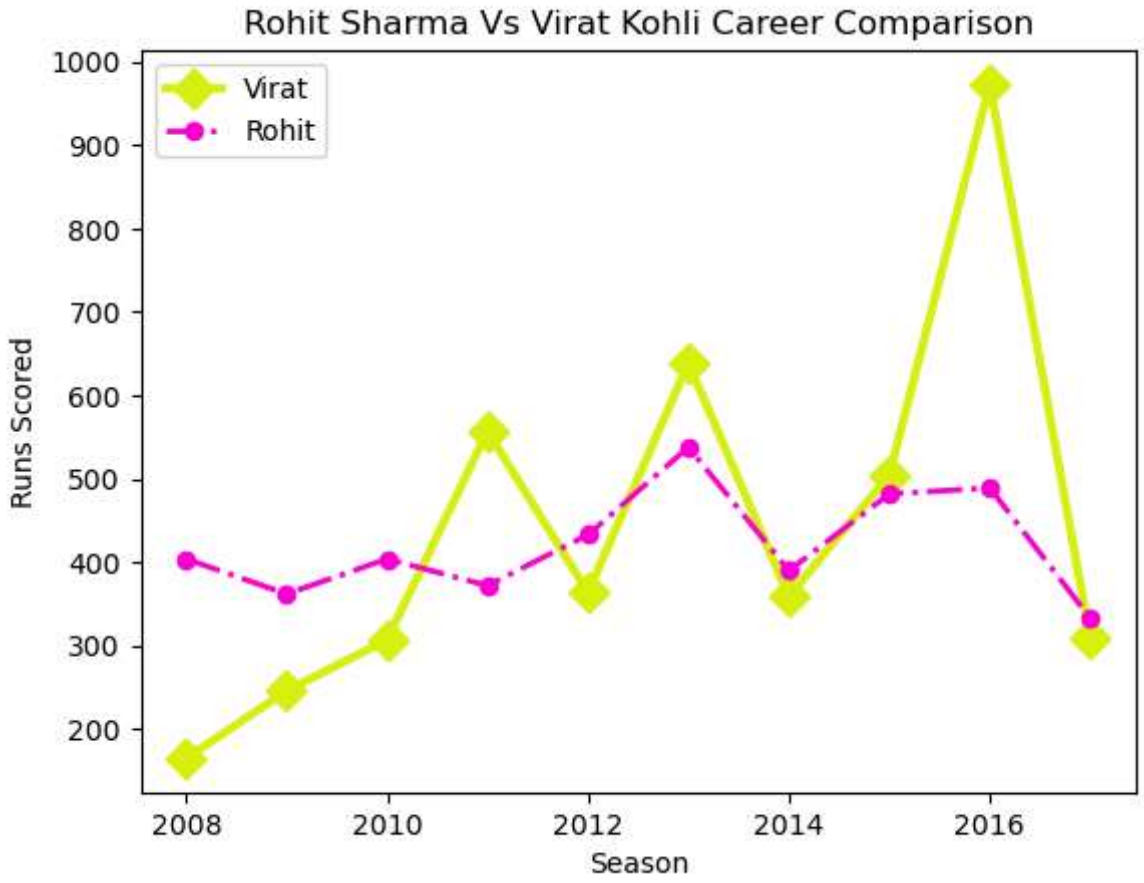
Out[ ]:
```
Text(0, 0.5, 'Runs Scored')
```



legend is use to differentiate the two lines

In [ ]:
```
# legend -> location
plt.plot(batsman['index'],batsman['V Kohli'],color='#D9F10F',linestyle='solid',lir
plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6',linestyle='dashdot

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

plt.legend(loc='upper left')#position of labels if we did not pass it automaticall
```

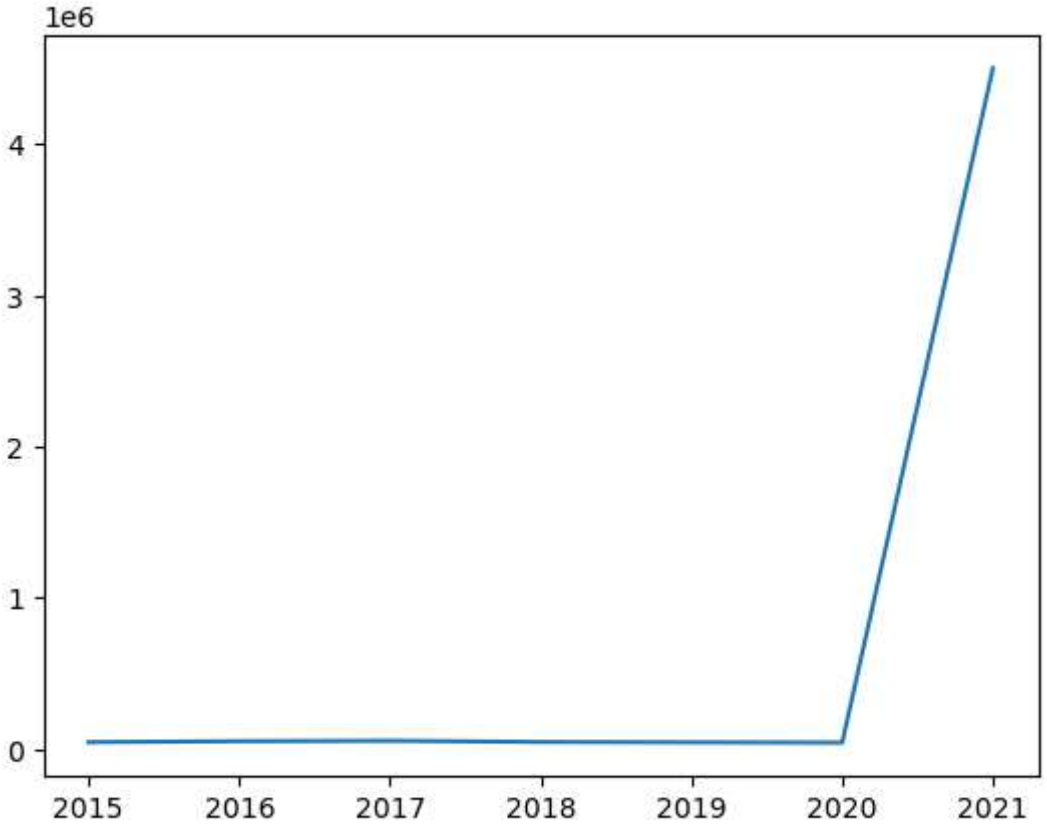
Out[ ]:
```
<matplotlib.legend.Legend at 0x17f3ac07bd0>
```



now we set the range of xaxis and yaxis

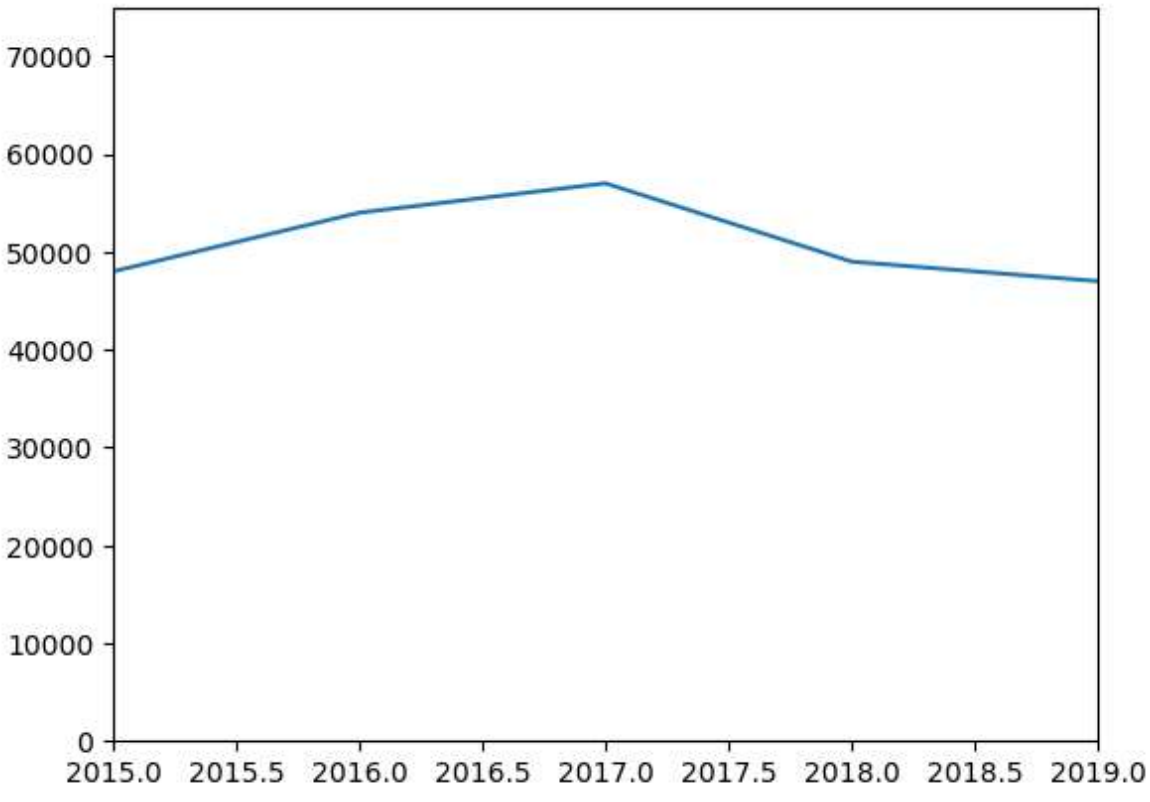
In [ ]:
```
# limiting axes
price = [48000,54000,57000,49000,47000,45000,4500000]
year = [2015,2016,2017,2018,2019,2020,2021]

plt.plot(year,price)
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x17f3b1b0710>]
```



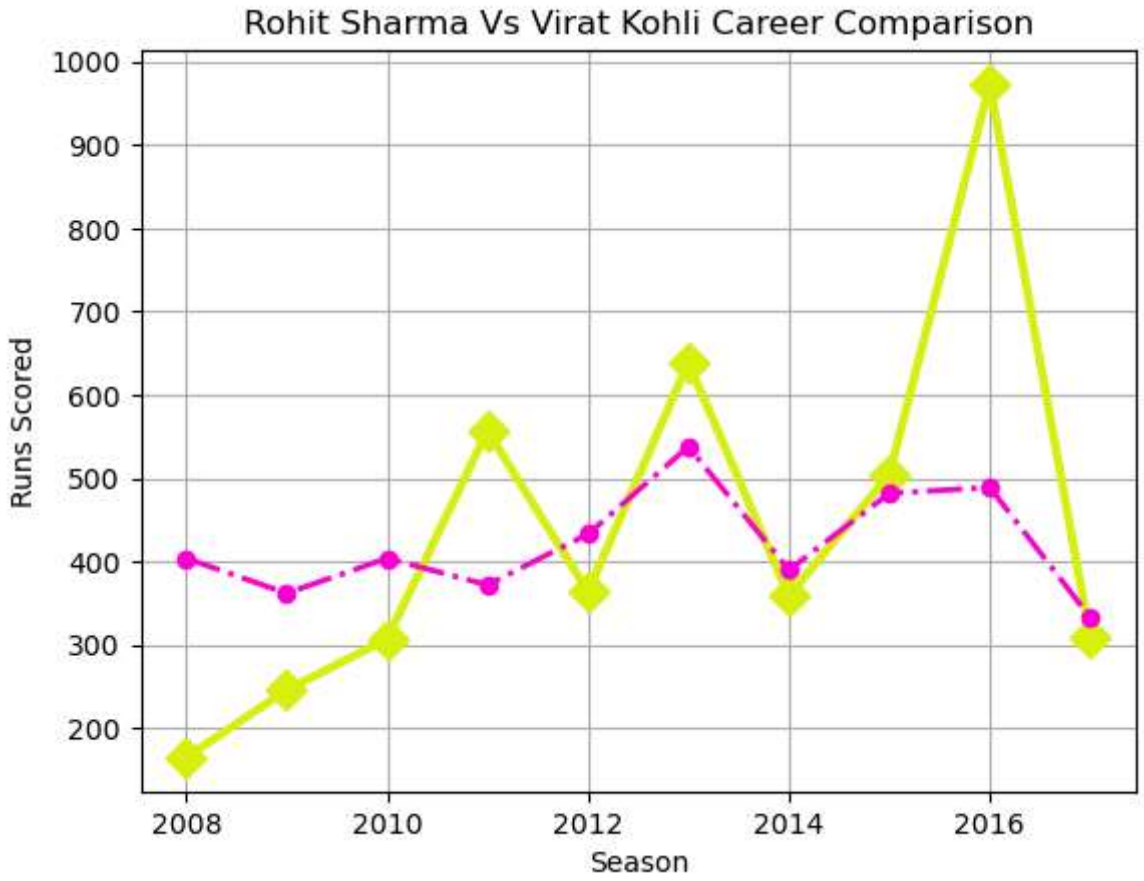
In [ ]:
```
price = [48000,54000,57000,49000,47000,45000,4500000]
year = [2015,2016,2017,2018,2019,2020,2021]

plt.plot(year,price)
plt.ylim(0,75000)
plt.xlim(2015,2019)
```

Out[ ]: (2015.0, 2019.0)

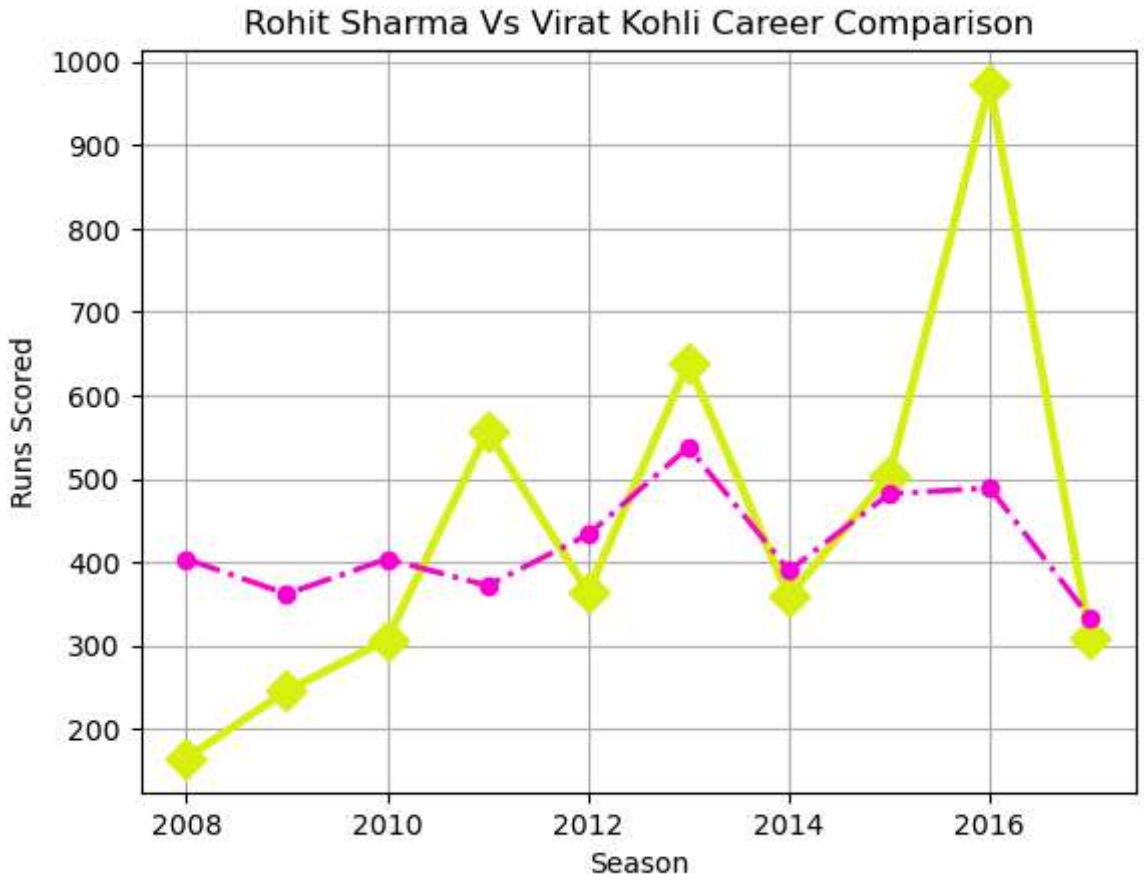


now we will add grid

In [ ]:
```
plt.plot(batsman['index'],batsman['V Kohli'],color='#D9F10F',linestyle='solid',lir
plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6',linestyle='dashdot

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

plt.grid()
plt.show()#in some code editor if graphs does not visible
```



In [ ]:
```
# show
plt.plot(batsman['index'],batsman['V Kohli'],color='#D9F10F',linestyle='solid',lir
plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6',linestyle='dashdot

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

plt.grid()

plt.show()
```



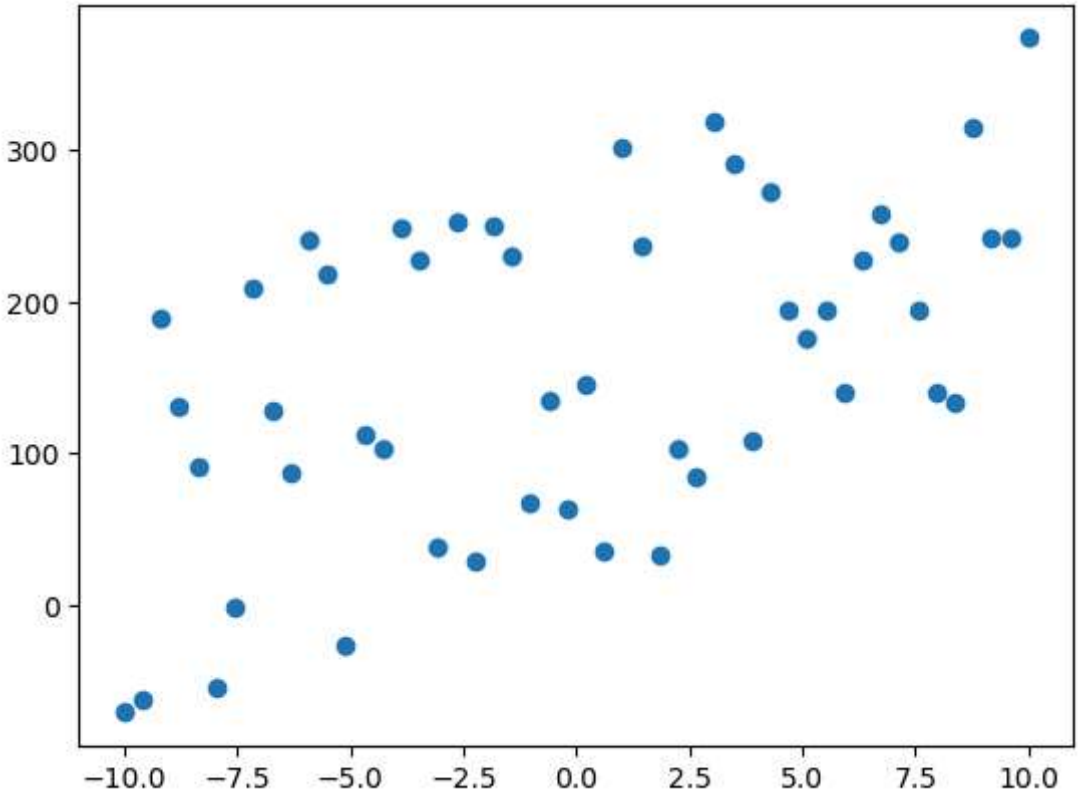
Scatter Plots

In [ ]:
```
# plt.scatter simple function
x = np.linspace(-10,10,50)

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

Out[ ]:
```
array([-71.        , -61.91836735, 189.16326531, 131.24489796,
        91.32653061, -54.59183673,  -1.51020408, 208.57142857,
       128.65306122,  87.73469388, 240.81632653, 218.89795918,
       -27.02040816, 112.06122449, 103.14285714, 248.2244898 ,
       227.30612245,  38.3877551 , 252.46938776,  28.55102041,
       250.63265306, 229.71428571,  67.79591837, 134.87755102,
        62.95918367, 146.04081633,  35.12244898, 301.20408163,
       236.28571429,  33.36734694, 103.44897959,  84.53061224,
       319.6122449 , 291.69387755, 107.7755102 , 272.85714286,
       194.93877551, 176.02040816, 194.10204082, 140.18367347,
       228.26530612, 258.34693878, 239.42857143, 194.51020408,
       139.59183673, 133.67346939, 314.75510204, 241.83673469,
       241.91836735, 374.        ])
```

In [ ]:
```
plt.scatter(x,y)
```

Out[ ]:
```
<matplotlib.collections.PathCollection at 0x17f3ae95290>
```

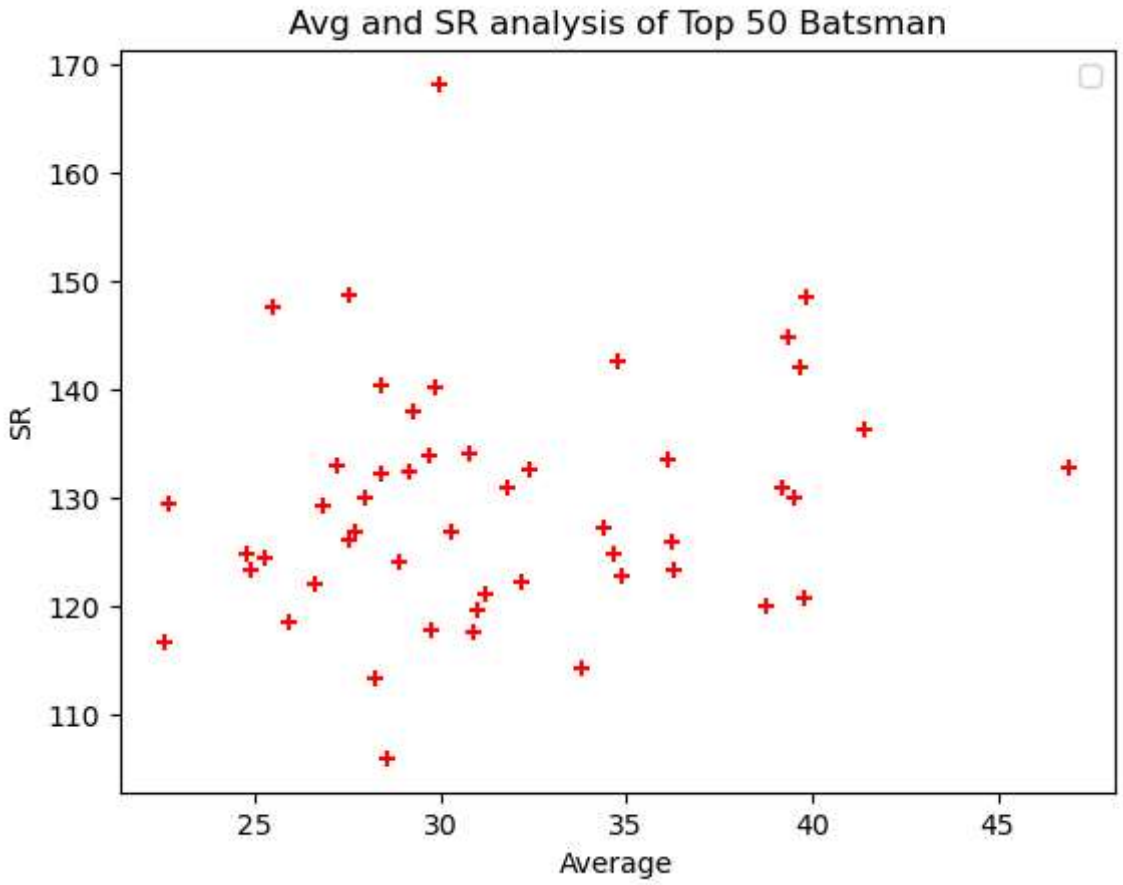


In [ ]:
```
# plt.scatter on pandas data
df = pd.read_csv('C:/Users/Mohan/Downloads/Matplotlib/batter.csv')
df = df.head(50)
```

In [ ]:
```
plt.scatter(df['avg'],df['strike_rate'],color='red',marker='+')
plt.title('Avg and SR analysis of Top 50 Batsman')
plt.xlabel('Average')
plt.ylabel('SR')
plt.legend()
```

```
No artists with labels found to put in legend.  Note that artists whose label start
with an underscore are ignored when legend() is called with no argument.
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x17f3af55410>
```



In [ ]:
```
tips = sns.load_dataset('tips')
```

In [ ]:
```
tips
```

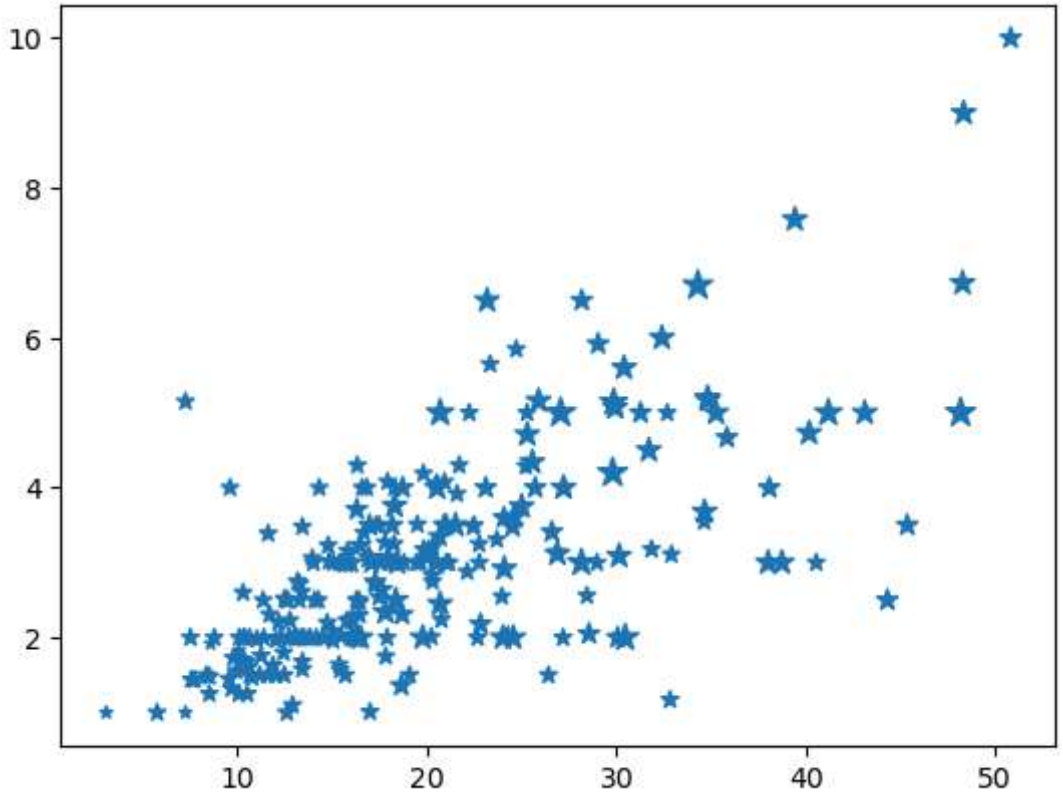
Out[ ]:

	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 × 7 columns

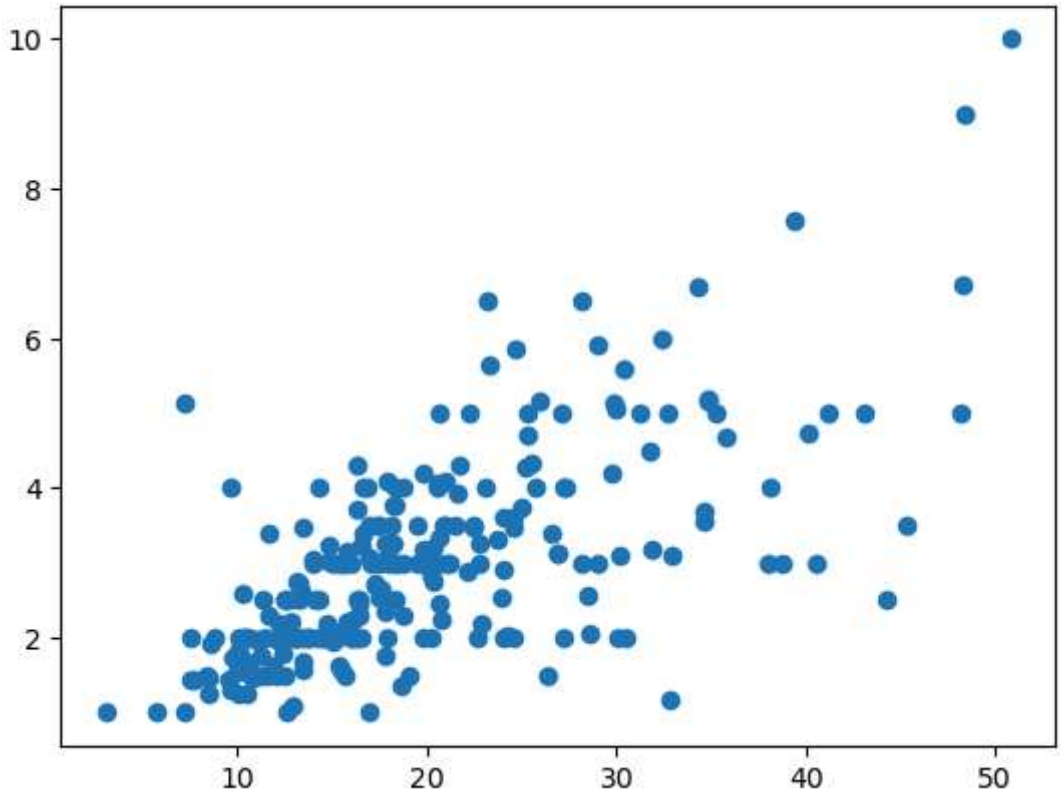
In [ ]:
```
plt.scatter(tips['total_bill'],tips['tip'],s=tips['size']*20,marker='*')
```

Out[ ]:
```
<matplotlib.collections.PathCollection at 0x17f3b5b5090>
```



In [ ]:
```
plt.plot(tips['total_bill'],tips['tip'],'o')
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x17f3b80c650>]
```

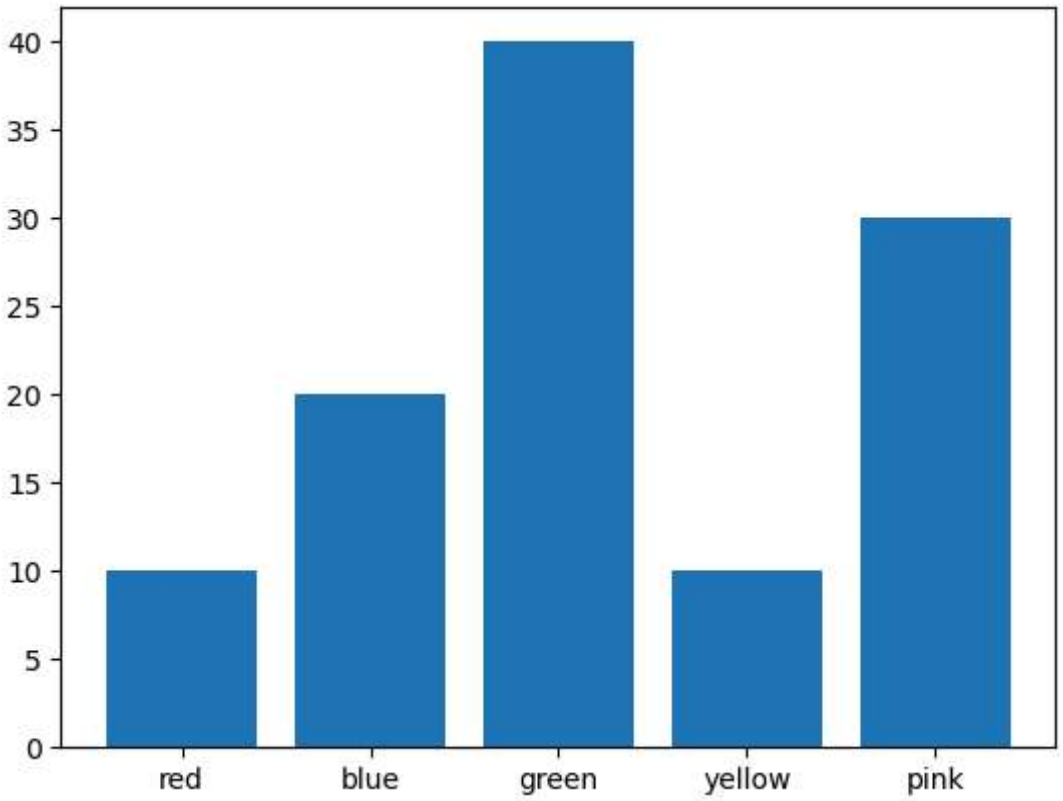


Bar chart

In [ ]:
```
# simple bar chart
children = [10,20,40,10,30]
```

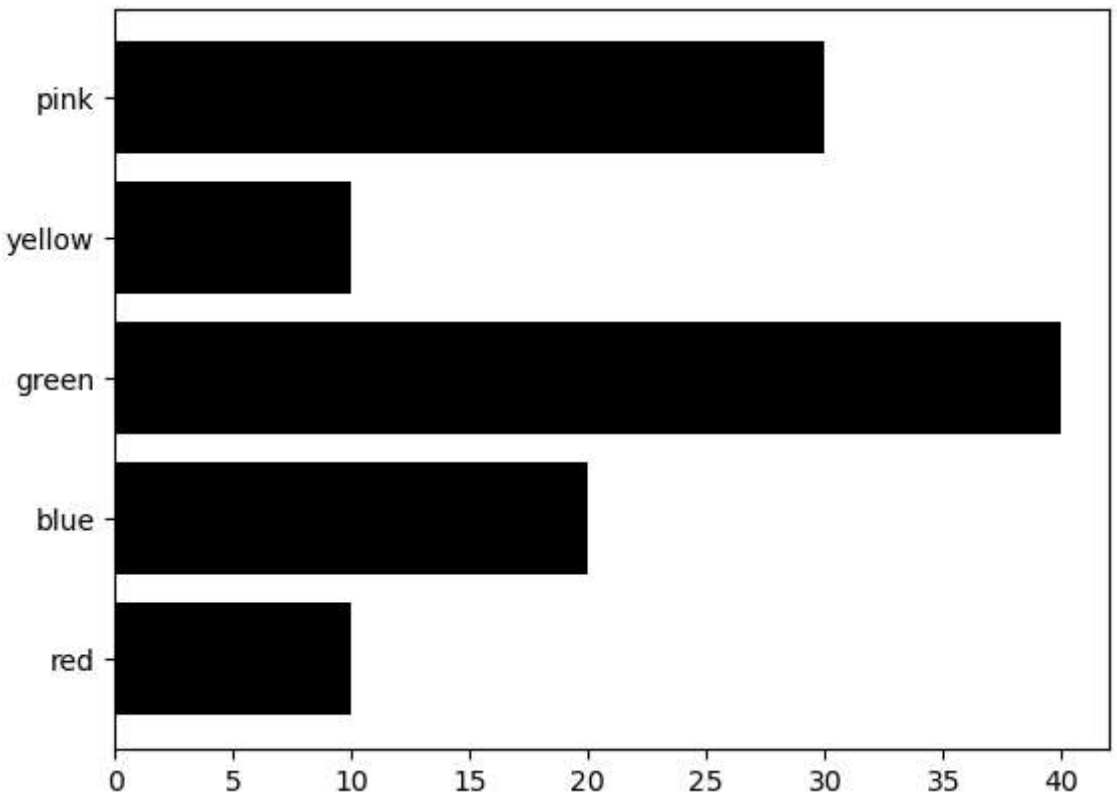
```
colors = ['red','blue','green','yellow','pink']

plt.bar(colors,children)
```

Out[ ]: <BarContainer object of 5 artists>



In [ ]:
```
# horizontal bar chart
plt.barh(colors,children,color='black')
```

Out[ ]: <BarContainer object of 5 artists>

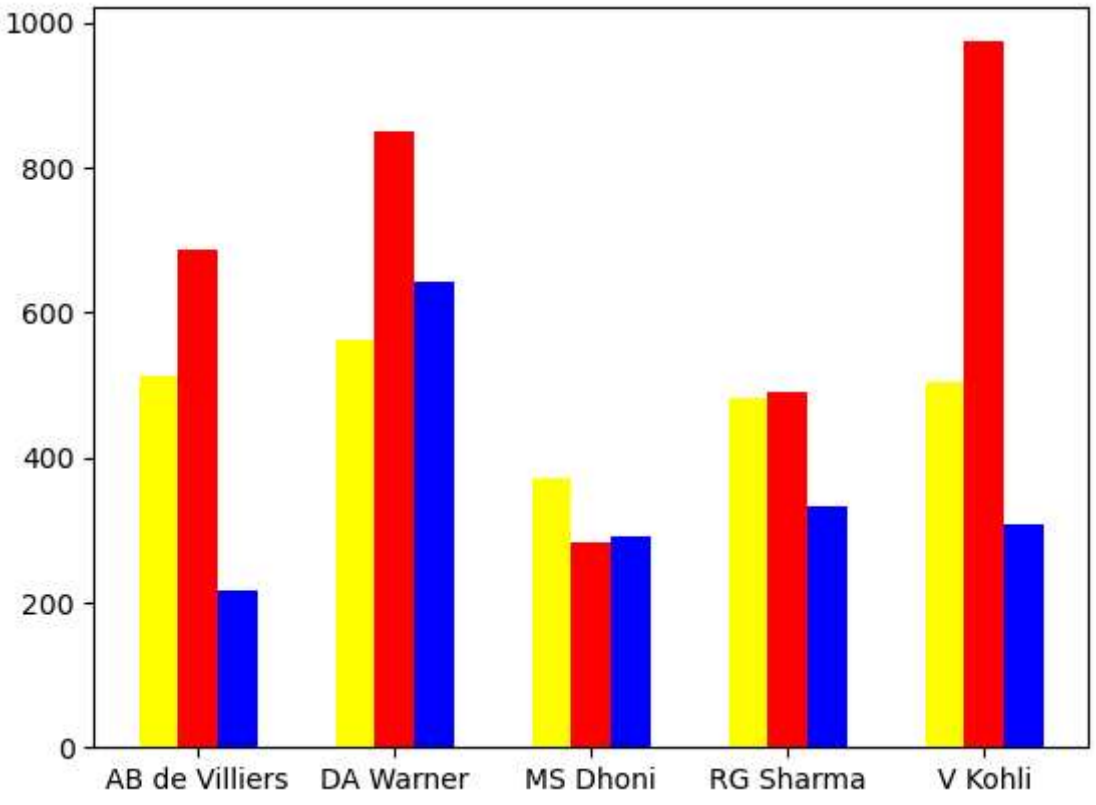


In [ ]:
```
# color and label
df = pd.read_csv('C:/Users/Mohan/Downloads/Matplotlib/batsman_season_record.csv')
df
```

Out[ ]:

	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

In [ ]:
```
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()
```



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

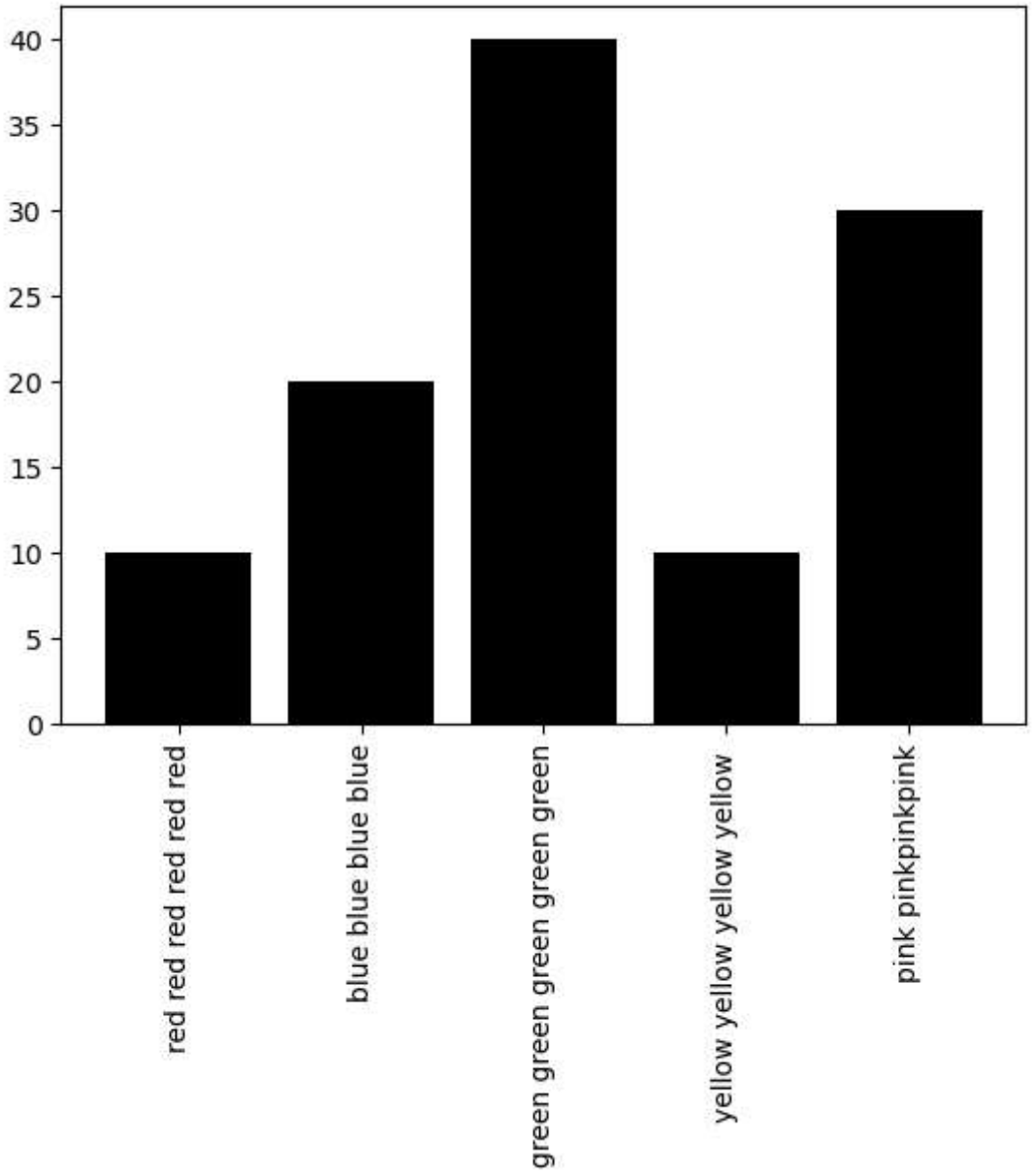
Out[ ]:
```
array([0, 1, 2, 3, 4])
```

In [ ]:
```
# Multiple Bar charts
```

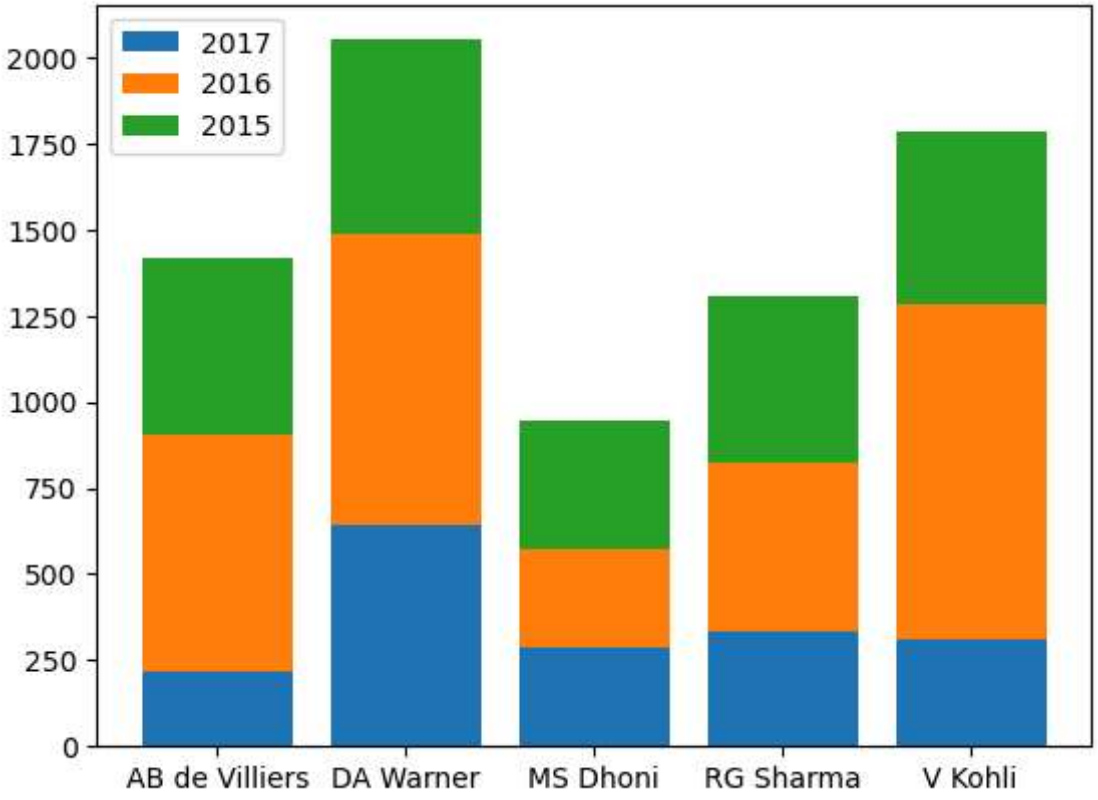
In [ ]:
```
# xticks
```

In [ ]:
```
# a problem
children = [10,20,40,10,30]
colors = ['red red red red red red','blue blue blue blue','green green green gree

plt.bar(colors,children,color='black')
plt.xticks(rotation='vertical')
```

Out[ ]:
```
([0, 1, 2, 3, 4],
 [Text(0, 0, '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')])
```



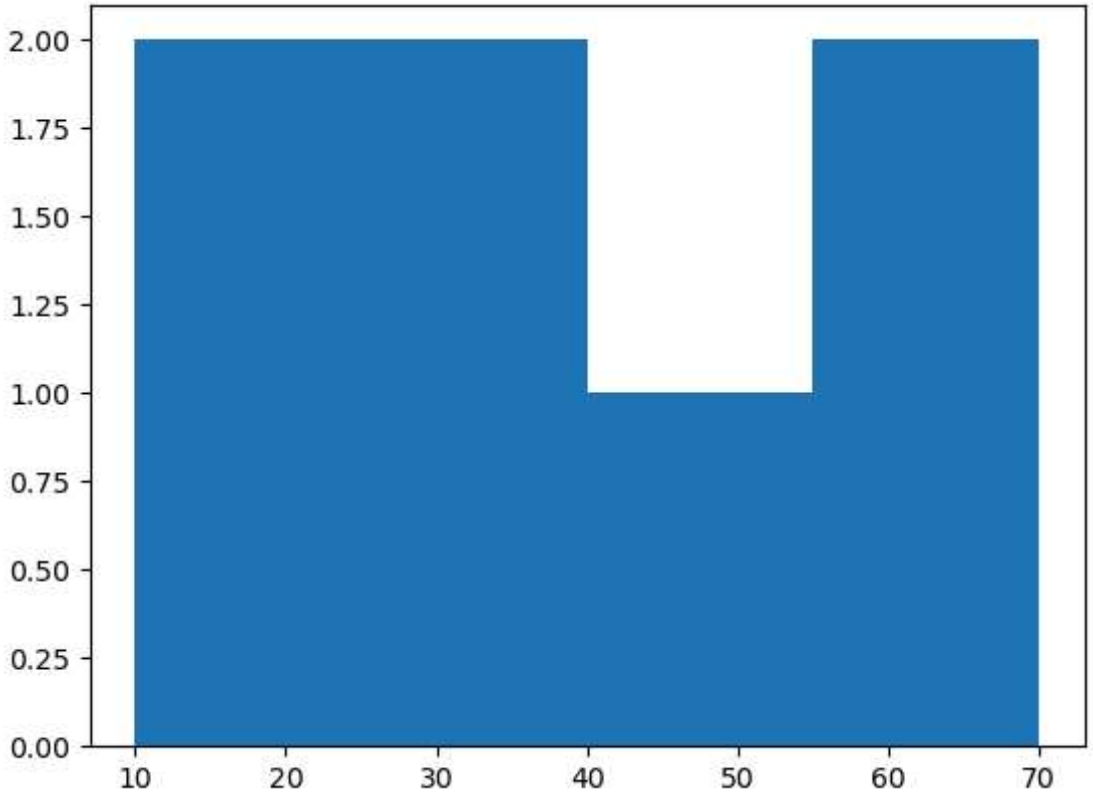
In [ ]:
```
# Stacked Bar chart
plt.bar(df['batsman'],df['2017'],label='2017')
plt.bar(df['batsman'],df['2016'],bottom=df['2017'],label='2016')
plt.bar(df['batsman'],df['2015'],bottom=(df['2016'] + df['2017']),label='2015')

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



Histogram

In [ ]:
```
# simple data

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

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

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



In [ ]:
```
# on some data
df = pd.read_csv('C:/Users/Mohan/Downloads/Matplotlib/vk.csv')
df
```

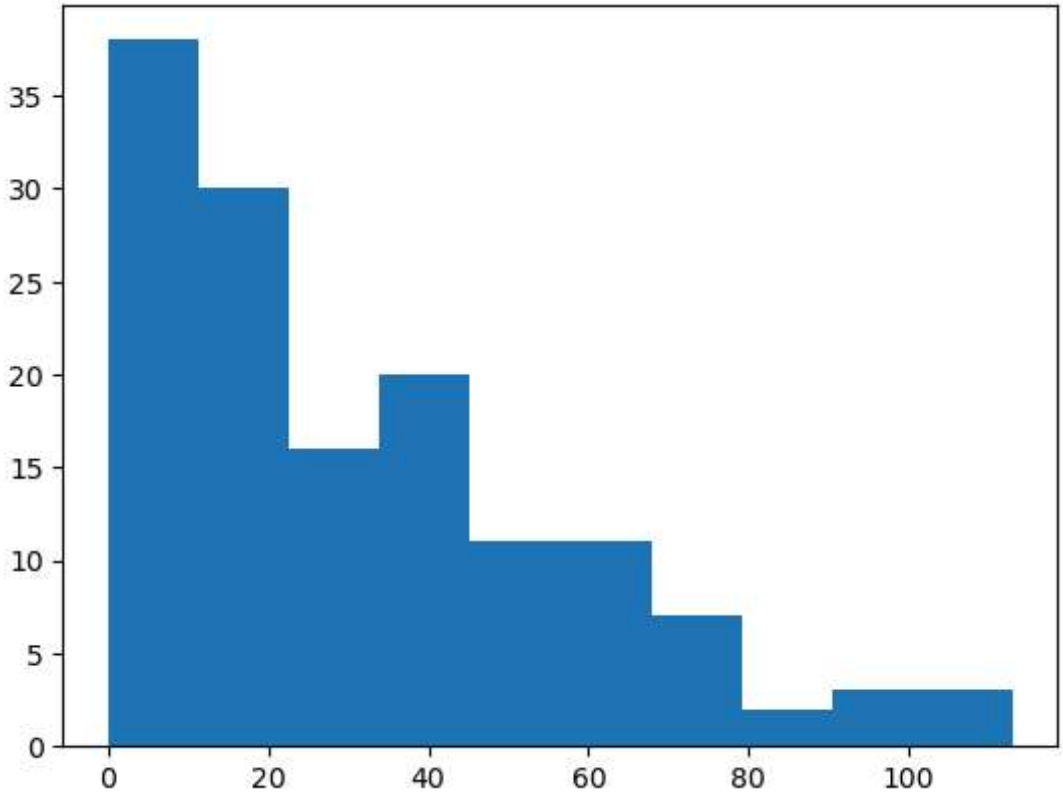
Out[ ]:

	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

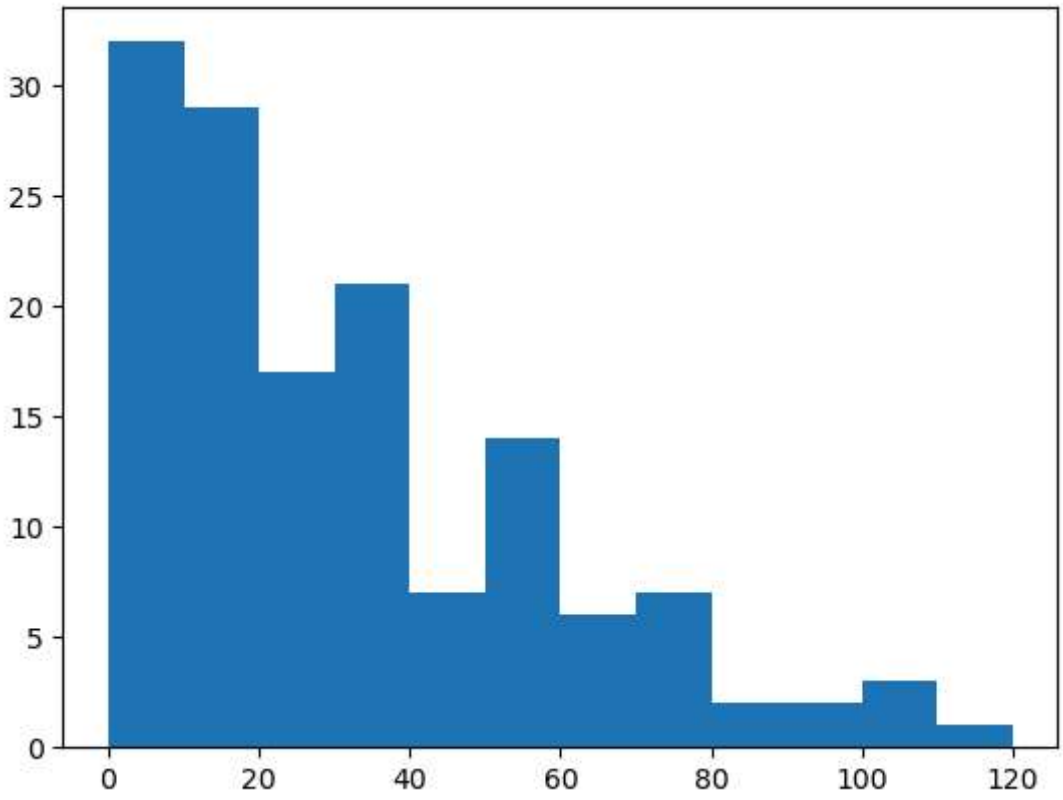
141 rows × 2 columns

In [ ]:
```
plt.hist(df['batsman_runs'])
```

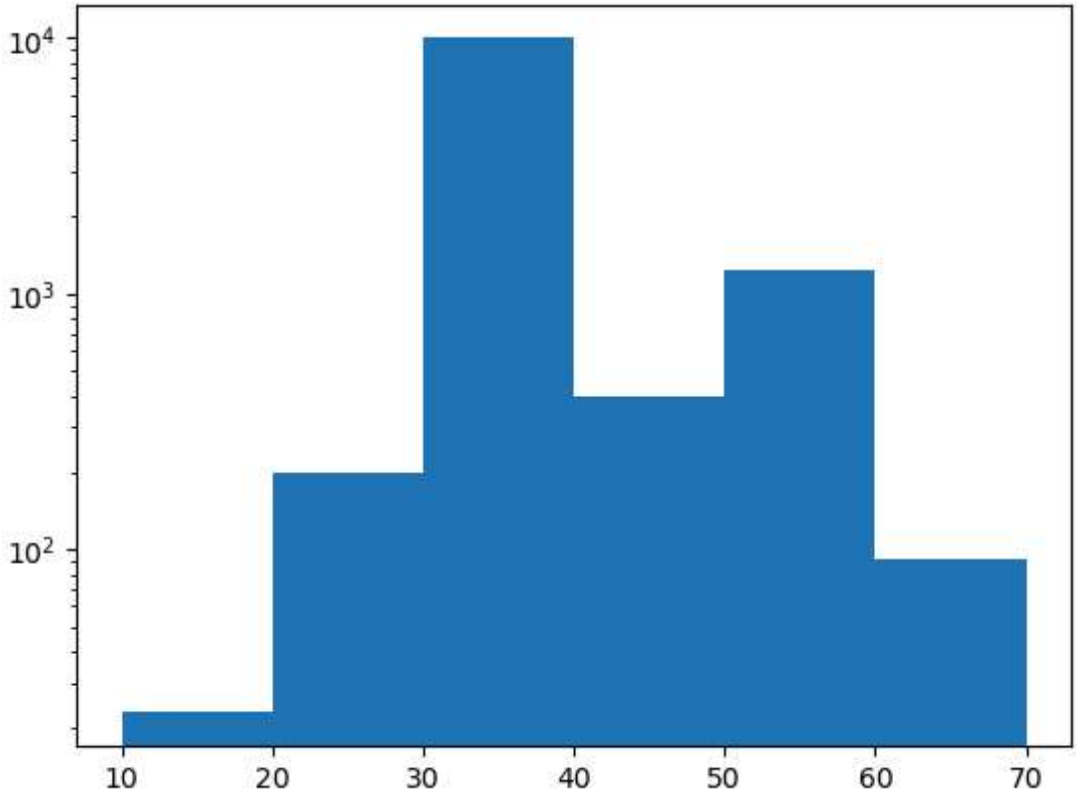
Out[ ]:
```
(array([38., 30., 16., 20., 11., 11.,  7.,  2.,  3.,  3.]),
 array([  0. ,  11.3,  22.6,  33.9,  45.2,  56.5,  67.8,  79.1,  90.4,
        101.7, 113. ]),
 <BarContainer object of 10 artists>)
```



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

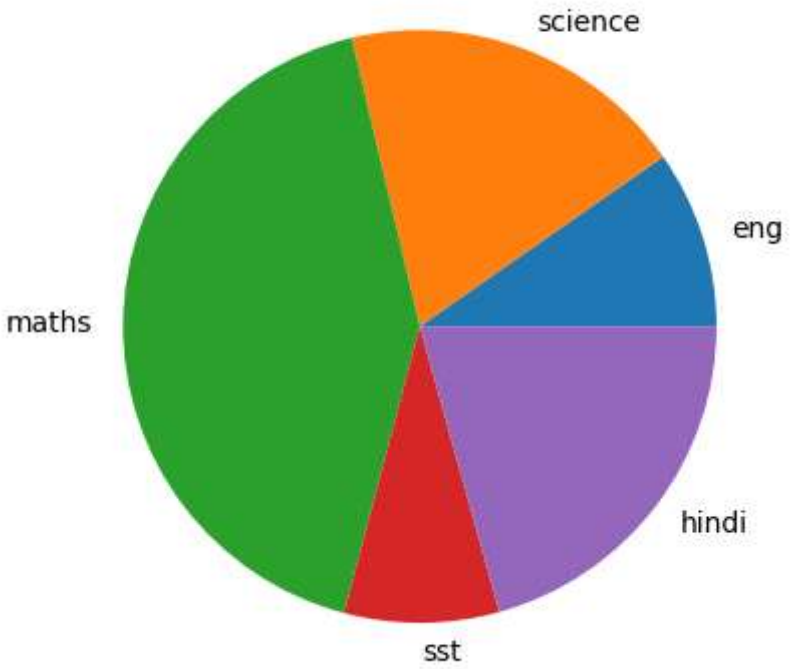


In [ ]:
```
# logarithmic scale
arr = np.load('C:/Users/Mohan/Downloads/Matplotlib/big-array.npy')
plt.hist(arr,bins=[10,20,30,40,50,60,70],log=True)
plt.show()
```



Pie Chart

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

plt.show()
```

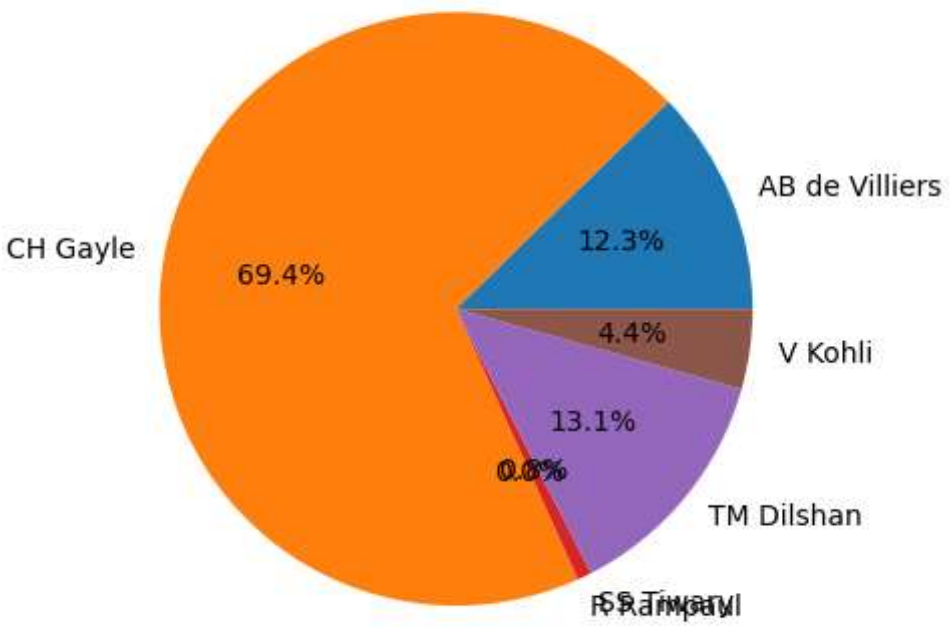


In [ ]:
```
# dataset
df = pd.read_csv('C:/Users/Mohan/Downloads/Matplotlib/gayle-175.csv')
df
```

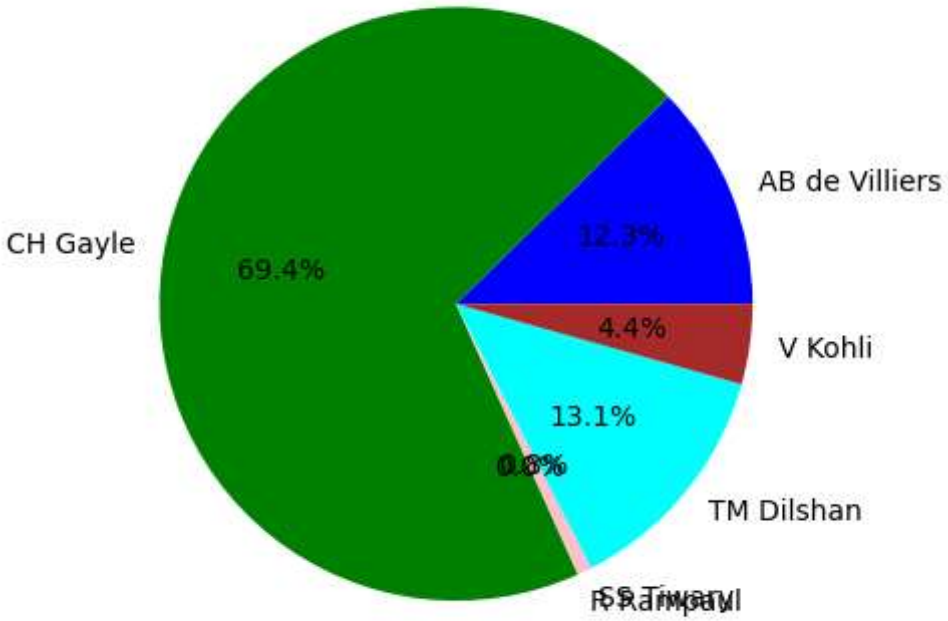
Out[ ]:

	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

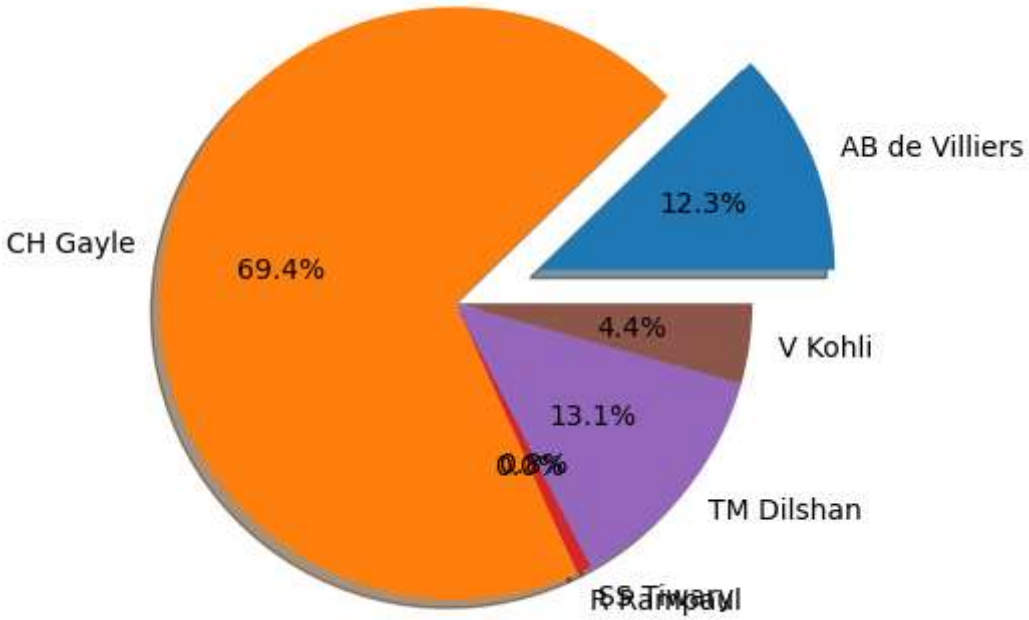
In [ ]:
```
plt.pie(df['batsman_runs'],labels=df['batsman'],autopct='%0.1f%%')
plt.show()
```



In [ ]:
```
# percentage and colors
plt.pie(df['batsman_runs'],labels=df['batsman'],autopct='%0.1f%%',colors=['blue',
plt.show()
```



In [ ]:
```
# explode shadow
plt.pie(df['batsman_runs'],labels=df['batsman'],autopct='%0.1f%%',explode=[0.3,0,0
plt.show()
```



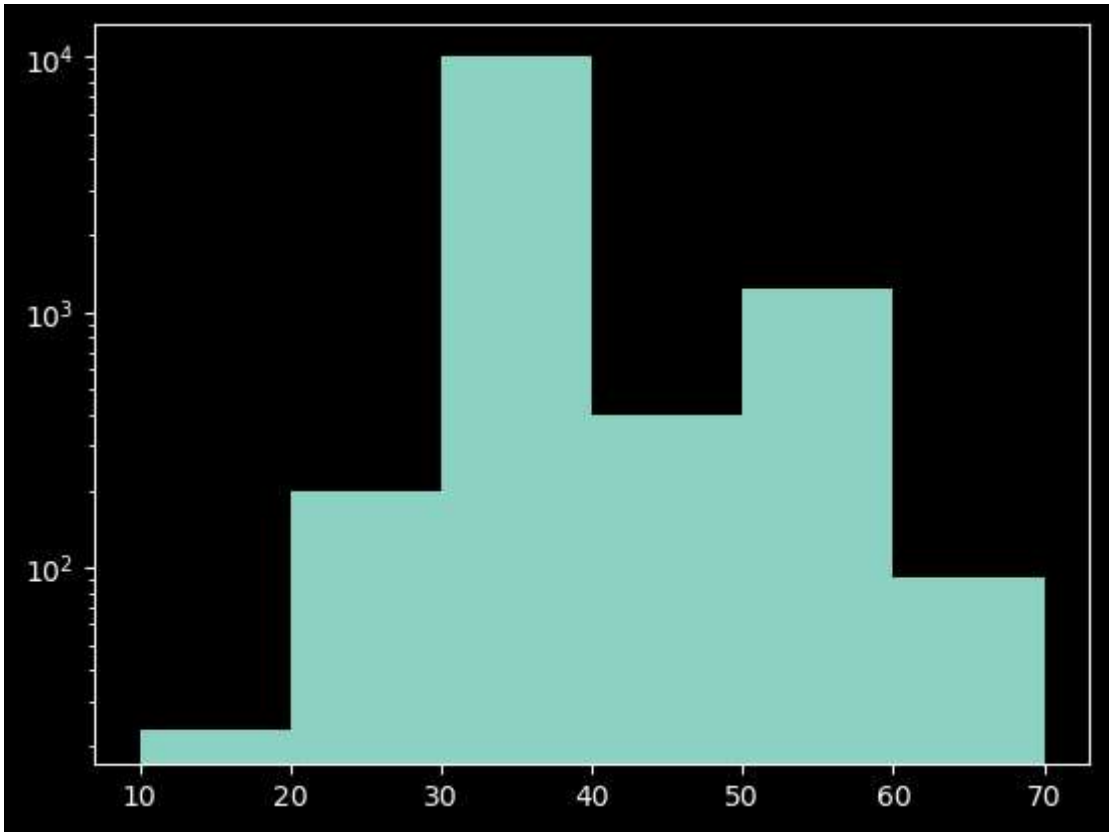
Changing styles

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

Out[ ]:
```
['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']
```

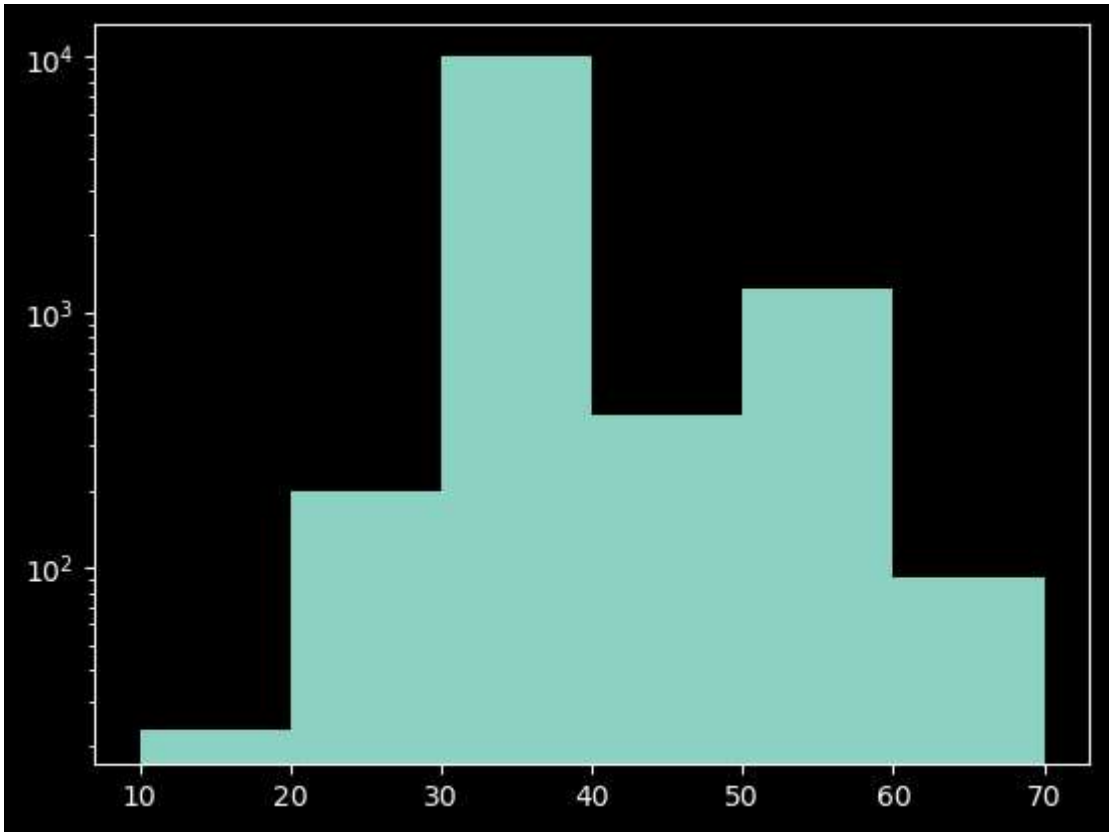
In [ ]:
```
plt.style.use('dark_background')
```

In [ ]:
```
arr = np.load('C:/Users/Mohan/Downloads/Matplotlib/big-array.npy')
plt.hist(arr,bins=[10,20,30,40,50,60,70],log=True)
plt.show()
```



Save figure

In [ ]:
```
arr = np.load('C:/Users/Mohan/Downloads/Matplotlib/big-array.npy')
plt.hist(arr,bins=[10,20,30,40,50,60,70],log=True)

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



Colored Scatterplot

In [ ]:
```
iris = pd.read_csv('C:/Users/Mohan/Downloads/Matplotlib/iris.csv')
iris.sample(5)
```

Out[ ]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
149	150	5.9	3.0	5.1	1.8	Iris-virginica
133	134	6.3	2.8	5.1	1.5	Iris-virginica
33	34	5.5	4.2	1.4	0.2	Iris-setosa
117	118	7.7	3.8	6.7	2.2	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica

Converting categorical column to numerical column

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

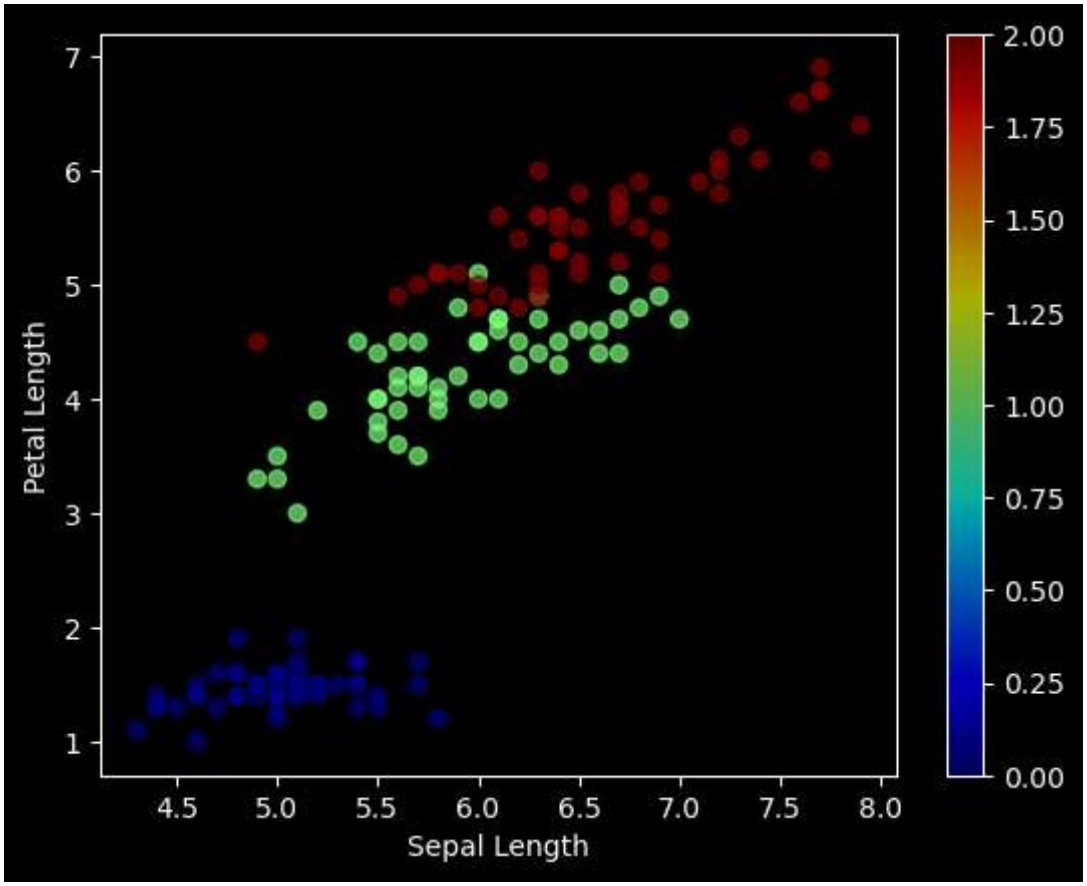
```
C:\Users\Mohan\AppData\Local\Temp\ipykernel_19140\855819524.py:1: FutureWarning: Do
wncasting behavior in `replace` is deprecated and will be removed in a future versi
on. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`.
To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcastin
g', True)`
  iris['Species'] = iris['Species'].replace({'Iris-setosa':0,'Iris-versicolor':1,'I
ris-virginica':2})
```

Out[ ]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
148	149	6.2	3.4	5.4	2.3	2
44	45	5.1	3.8	1.9	0.4	0
59	60	5.2	2.7	3.9	1.4	1
131	132	7.9	3.8	6.4	2.0	2
100	101	6.3	3.3	6.0	2.5	2

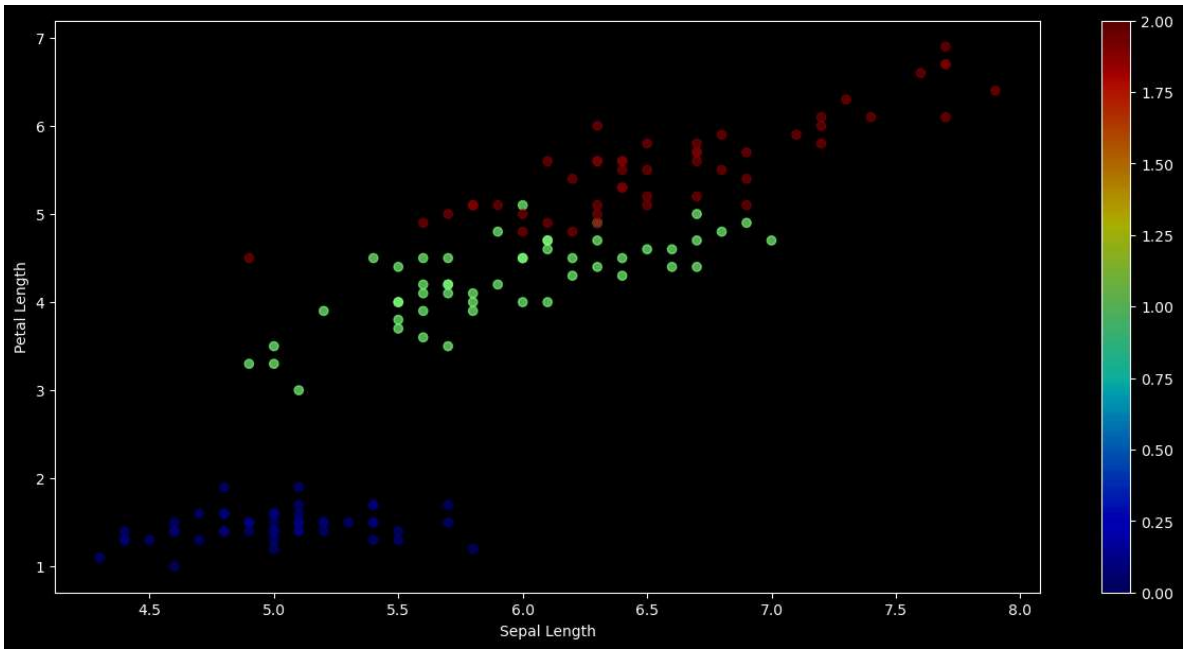
In [ ]:
```
plt.scatter(iris['SepalLengthCm'],iris['PetalLengthCm'],c=iris['Species'],cmap='je
plt.xlabel('Sepal Length')
plt.ylabel('Petal Length')
plt.colorbar()
```

Out[ ]:
```
<matplotlib.colorbar.Colorbar at 0x17f3e9ec050>
```



In [ ]:
```
#plot size
plt.figure(figsize=(15,7))

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

Out[ ]:
```
<matplotlib.colorbar.Colorbar at 0x17f3edc6e90>
```



In [ ]:
```
batters = pd.read_csv('C:/Users/Mohan/Downloads/Matplotlib/batter.csv')
```

In [ ]:
```
batters
```

Out[ ]:

	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 × 4 columns

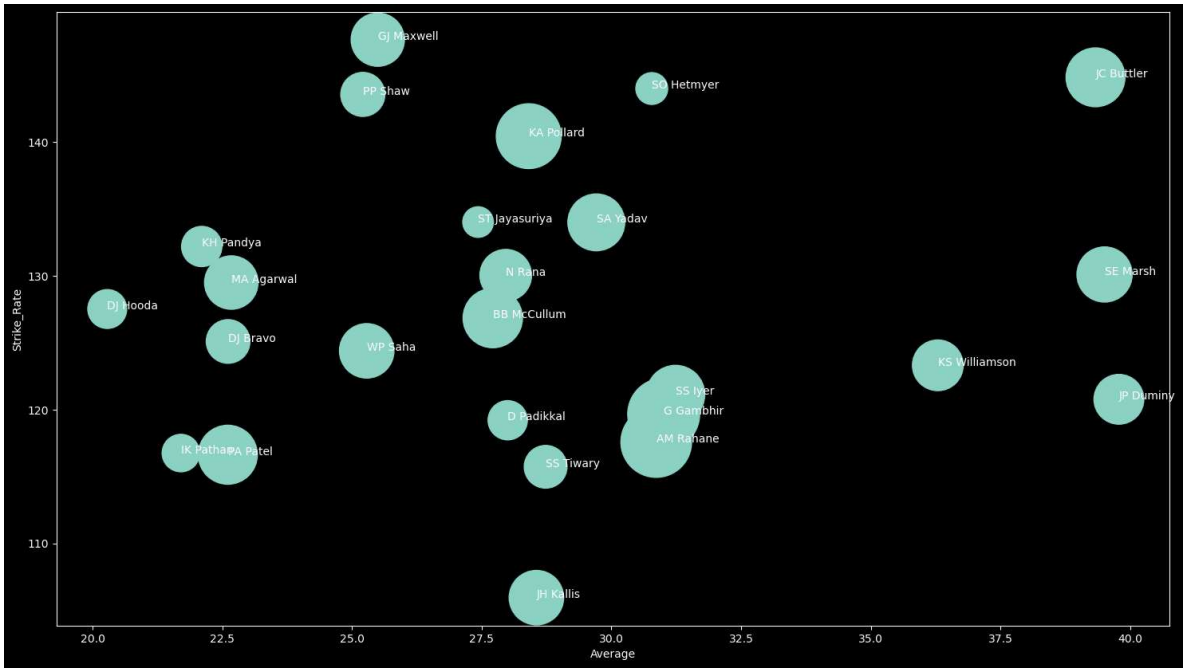
In [ ]:
```
sample_df = batters.head(100).sample(25,random_state=5)
```

In [ ]:
```
sample_df
```

Out[ ]:

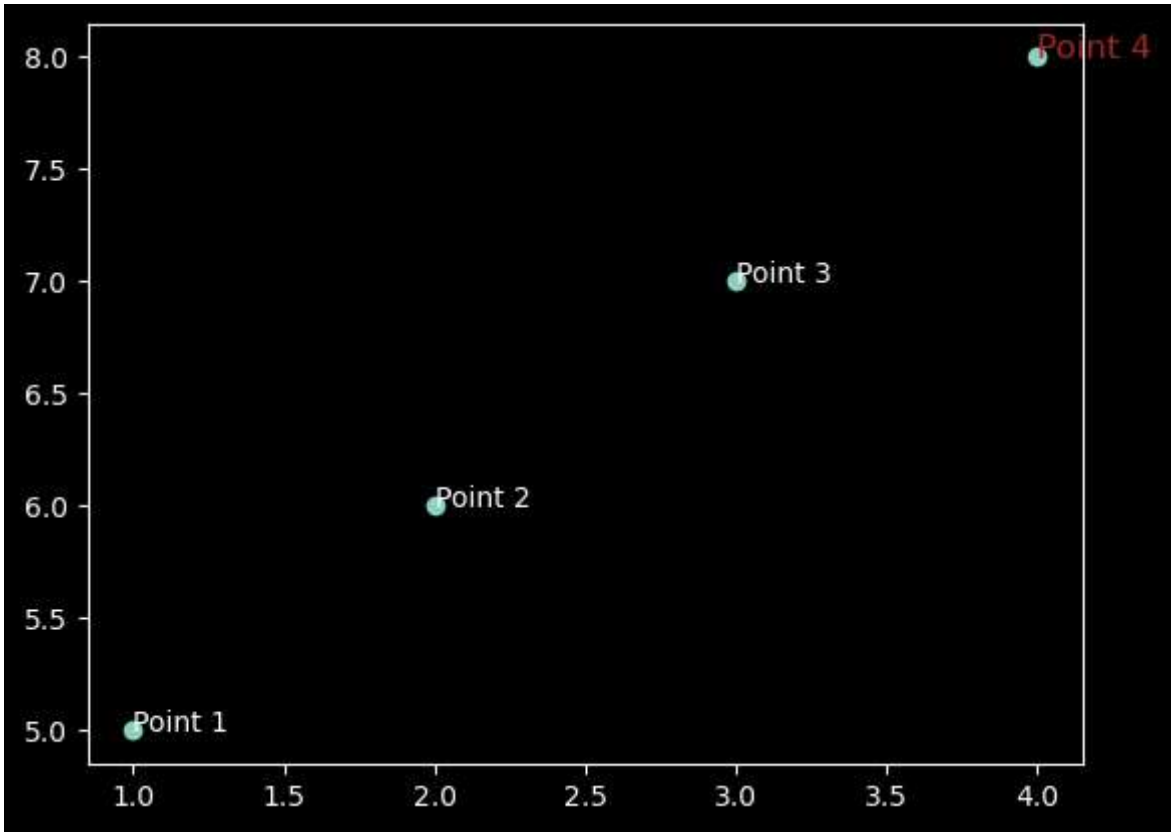
	batter	runs	avg	strike_rate
66	KH Pandya	1326	22.100000	132.203390
32	SE Marsh	2489	39.507937	130.109775
46	JP Duminy	2029	39.784314	120.773810
28	SA Yadav	2644	29.707865	134.009123
74	IK Pathan	1150	21.698113	116.751269
23	JC Buttler	2832	39.333333	144.859335
10	G Gambhir	4217	31.007353	119.665153
20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

In [ ]:
```
plt.figure(figsize=(18,10))
plt.scatter(sample_df['avg'],sample_df['strike_rate'],s=sample_df['runs'])
plt.xlabel("Average")
plt.ylabel("Strike_Rate")

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



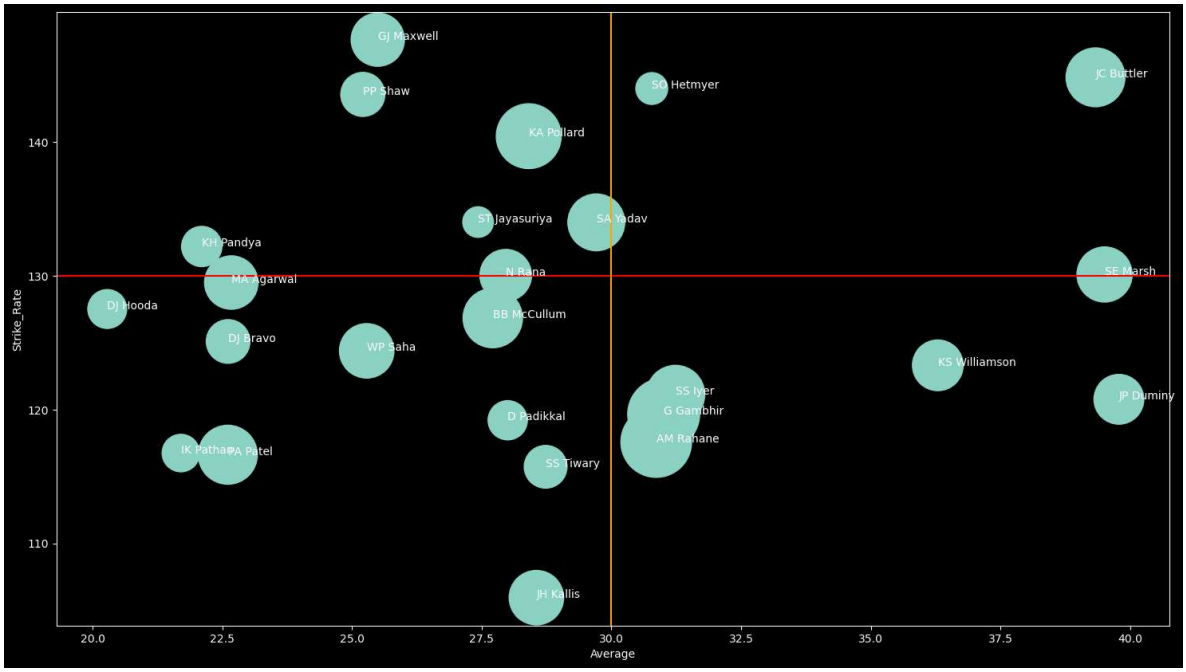
In [ ]:
```
x = [1,2,3,4]
y = [5,6,7,8]

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'})
```

Out[ ]: Text(4, 8, 'Point 4')



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

```
plt.ylabel("Strike_Rate")
plt.axhline(130,color='red')
plt.axvline(30,color='orange')

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



Subplots

In [ ]:
```
batters
```

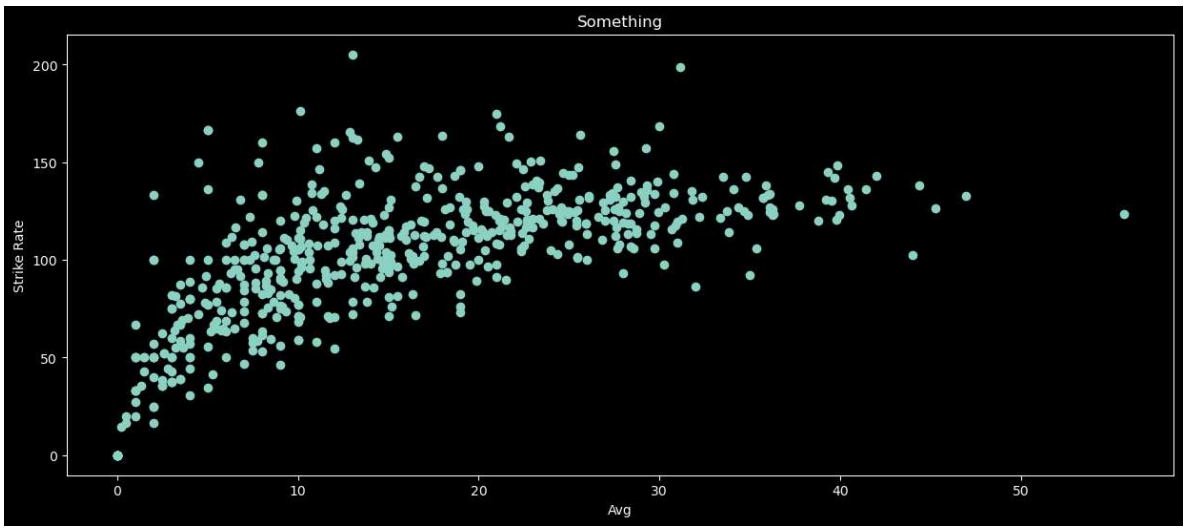
Out[ ]:

	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 × 4 columns

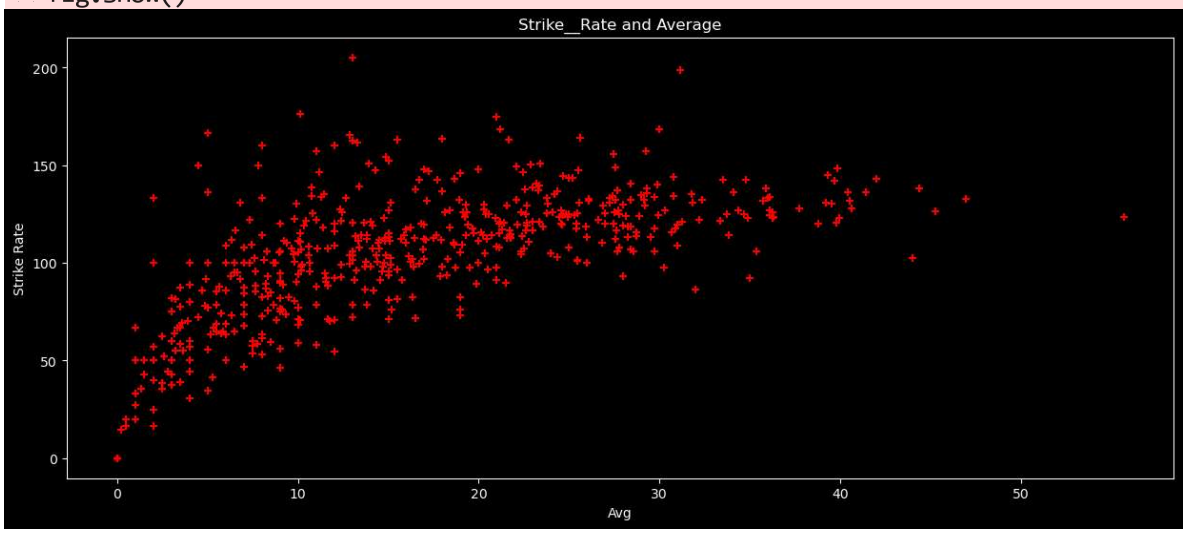
In [ ]:
```
plt.figure(figsize=(15,6))
plt.scatter(batters['avg'],batters['strike_rate'])
```

```
plt.title('Something')
plt.xlabel('Avg')
plt.ylabel('Strike Rate')

plt.show()
```



In [ ]:
```
fig,ax = plt.subplots(figsize=(15,6))

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

fig.show()
```

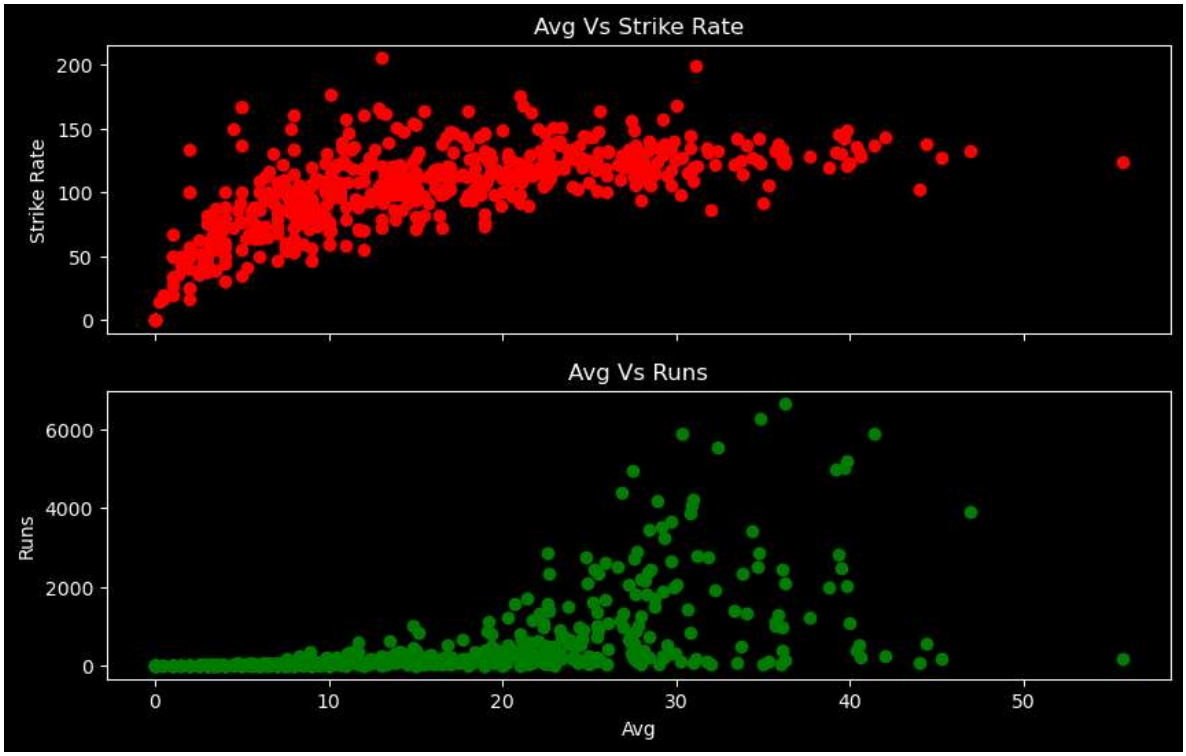
```
C:\Users\Mohan\AppData\Local\Temp\ipykernel_19140\2945378197.py:8: UserWarning: Fig
ureCanvasAgg is non-interactive, and thus cannot be shown
  fig.show()
```



In [ ]:
```
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'],color='green')

ax[0].set_title('Avg Vs Strike Rate')
ax[0].set_ylabel('Strike Rate')


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

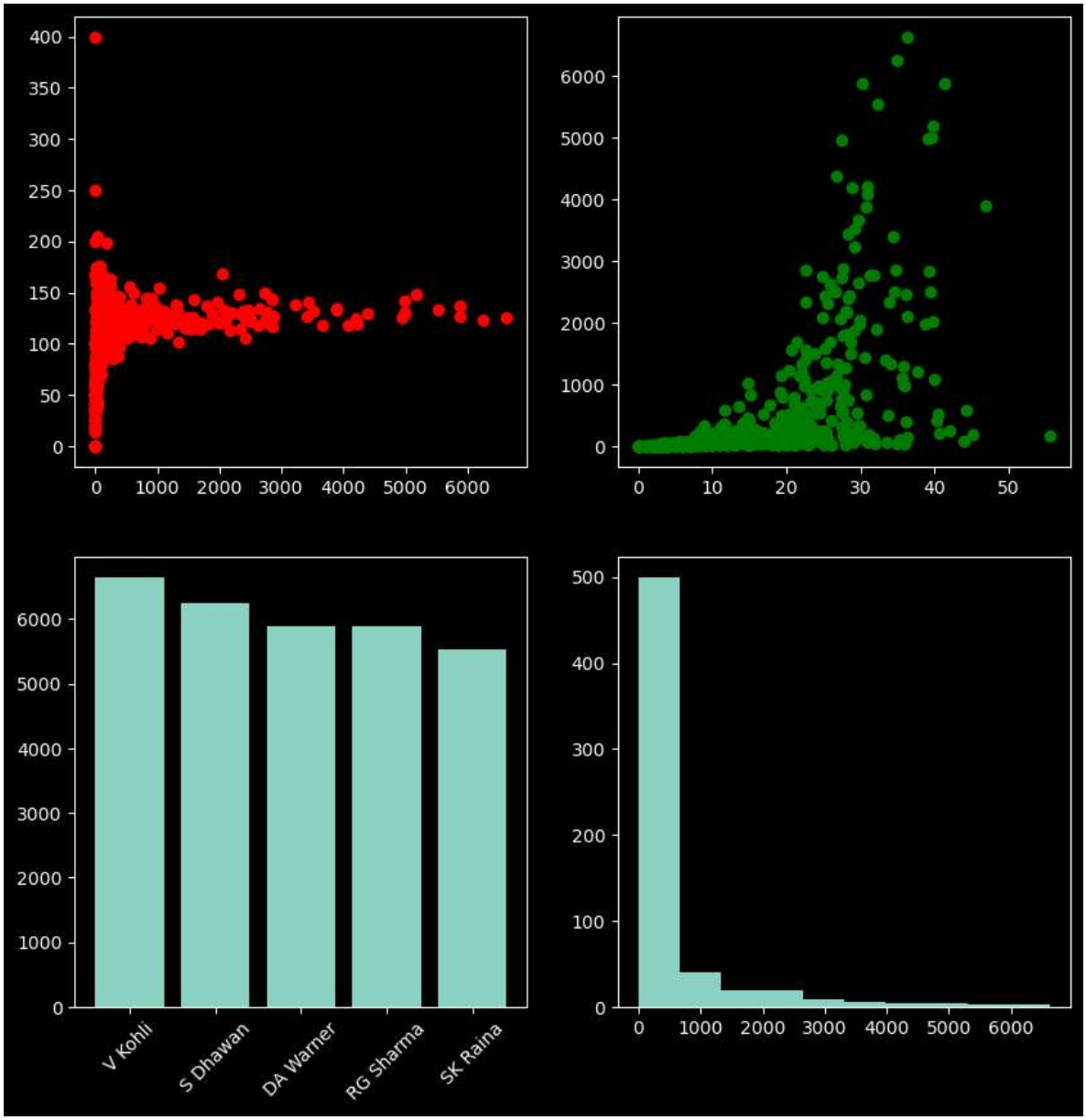
```
ax[1].set_ylabel('Runs')
ax[1].set_xlabel('Avg')
```

Out[ ]: Text(0.5, 0, 'Avg')



In [ ]:
```
fig, ax = plt.subplots(nrows=2,ncols=2,figsize=(10,10))

ax[0,0].scatter(batters['runs'],batters['strike_rate'],color='red')
ax[0,1].scatter(batters['avg'],batters['runs'],color='green')
ax[1,0].bar(batters['batter'].head(),batters['runs'].head())
ax[1, 0].tick_params(axis='x', rotation=45)
ax[1,1].hist(batters['runs'])
```

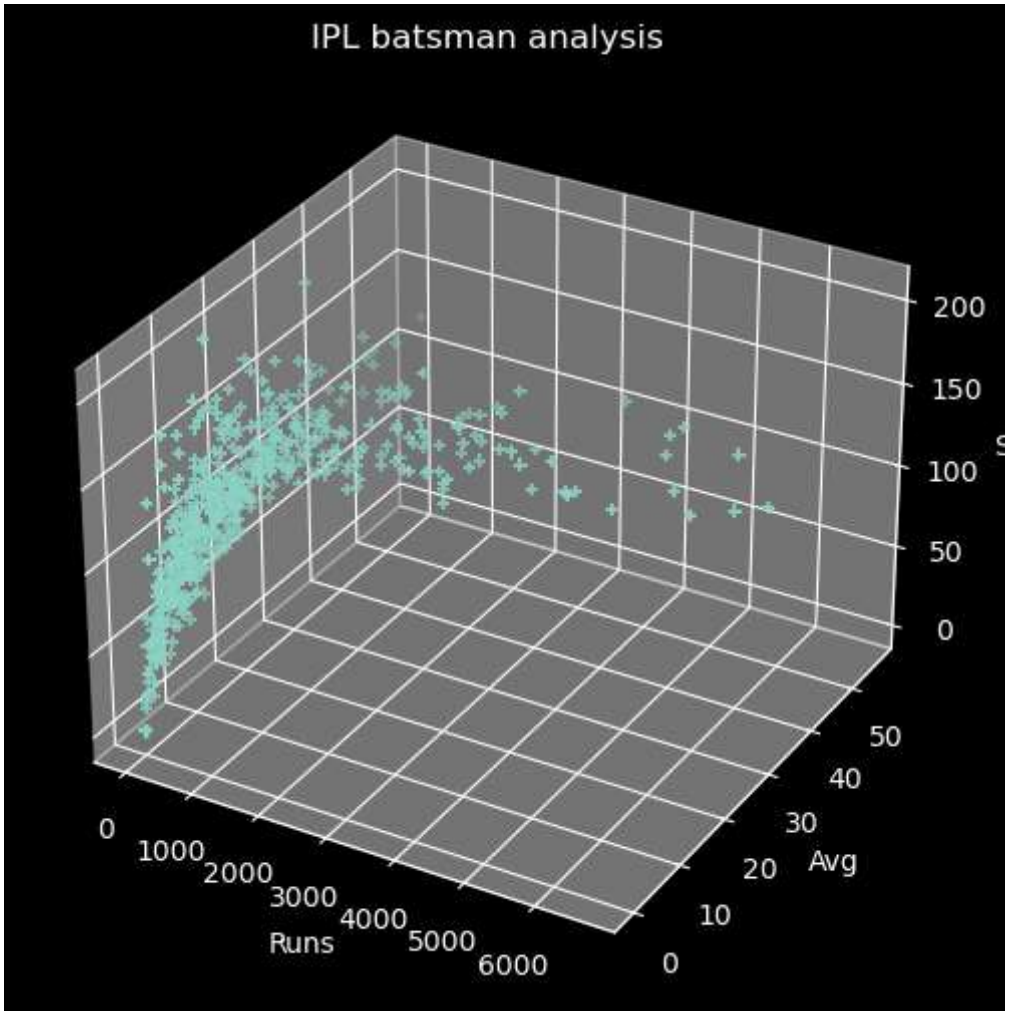
Out[ ]:
```
(array([499.,  40.,  19.,  19.,   9.,   6.,   4.,   4.,   3.,   2.]),
 array([   0. ,  663.4, 1326.8, 1990.2, 2653.6, 3317. , 3980.4, 4643.8,
        5307.2, 5970.6, 6634. ]),
 <BarContainer object of 10 artists>)
```



3D Scatter Plots

In [ ]:
```
fig = plt.figure(figsize=(6,6))

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')
```

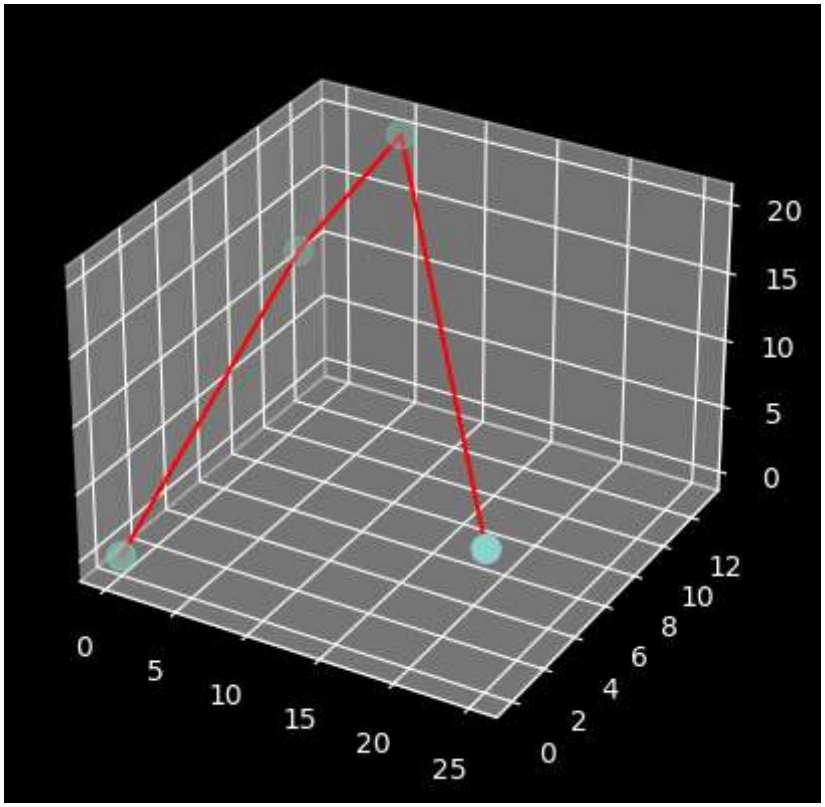
Out[ ]:
```
Text(0.5, 0, 'SR')
```



3D Line Plot

In [ ]:
```
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')
```

Out[ ]:
```
[<mpl_toolkits.mplot3d.art3d.Line3D at 0x17f3c915dd0>]
```



3D Surface Plots

In [ ]:
```
x = np.linspace(-10,10,100)
y = np.linspace(-10,10,100)
```

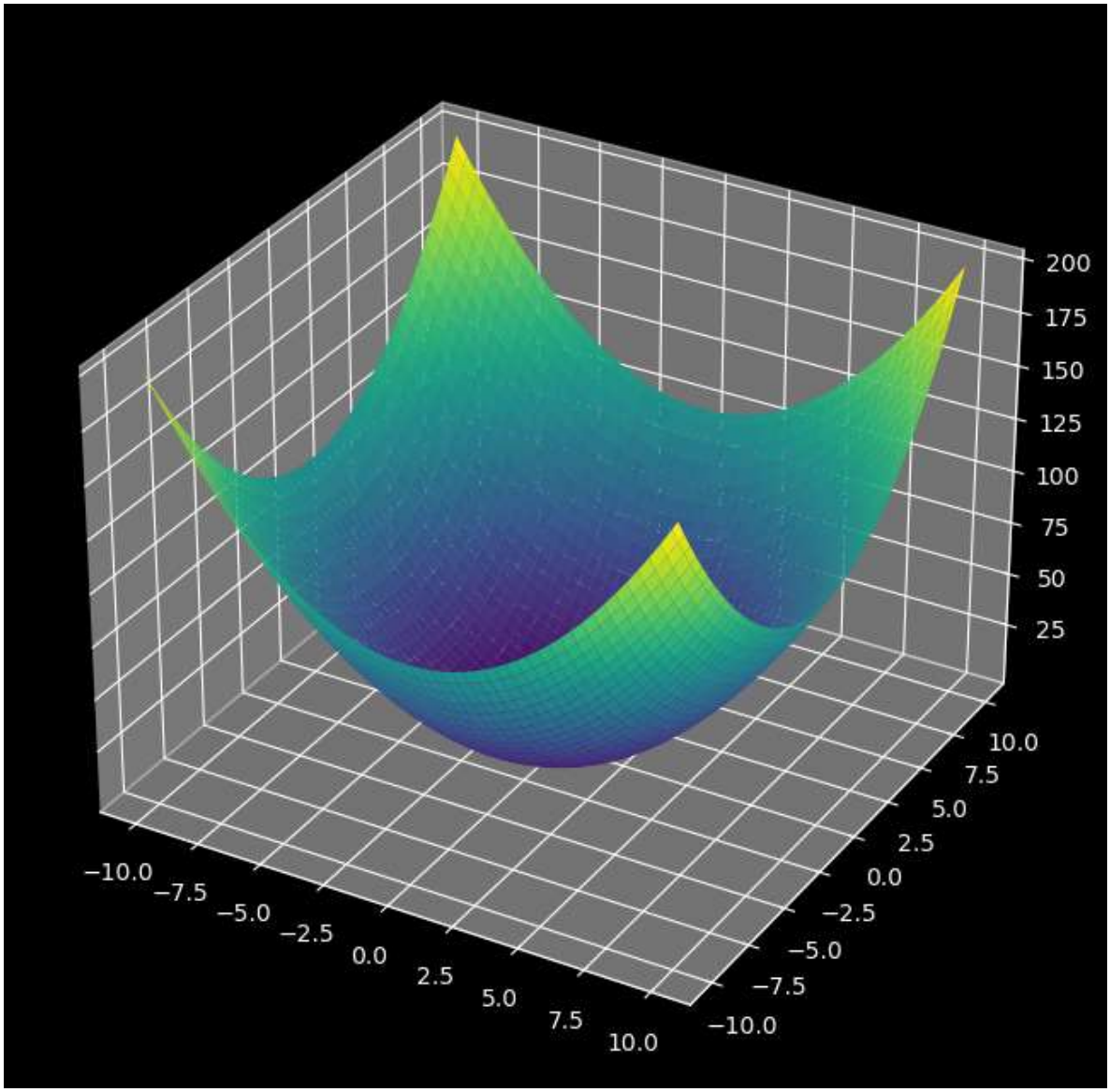
In [ ]:
```
xx,yy=np.meshgrid(x,y)
```

In [ ]:
```
z = xx**2 + yy**2
z.shape
```

Out[ ]:
```
(100, 100)
```

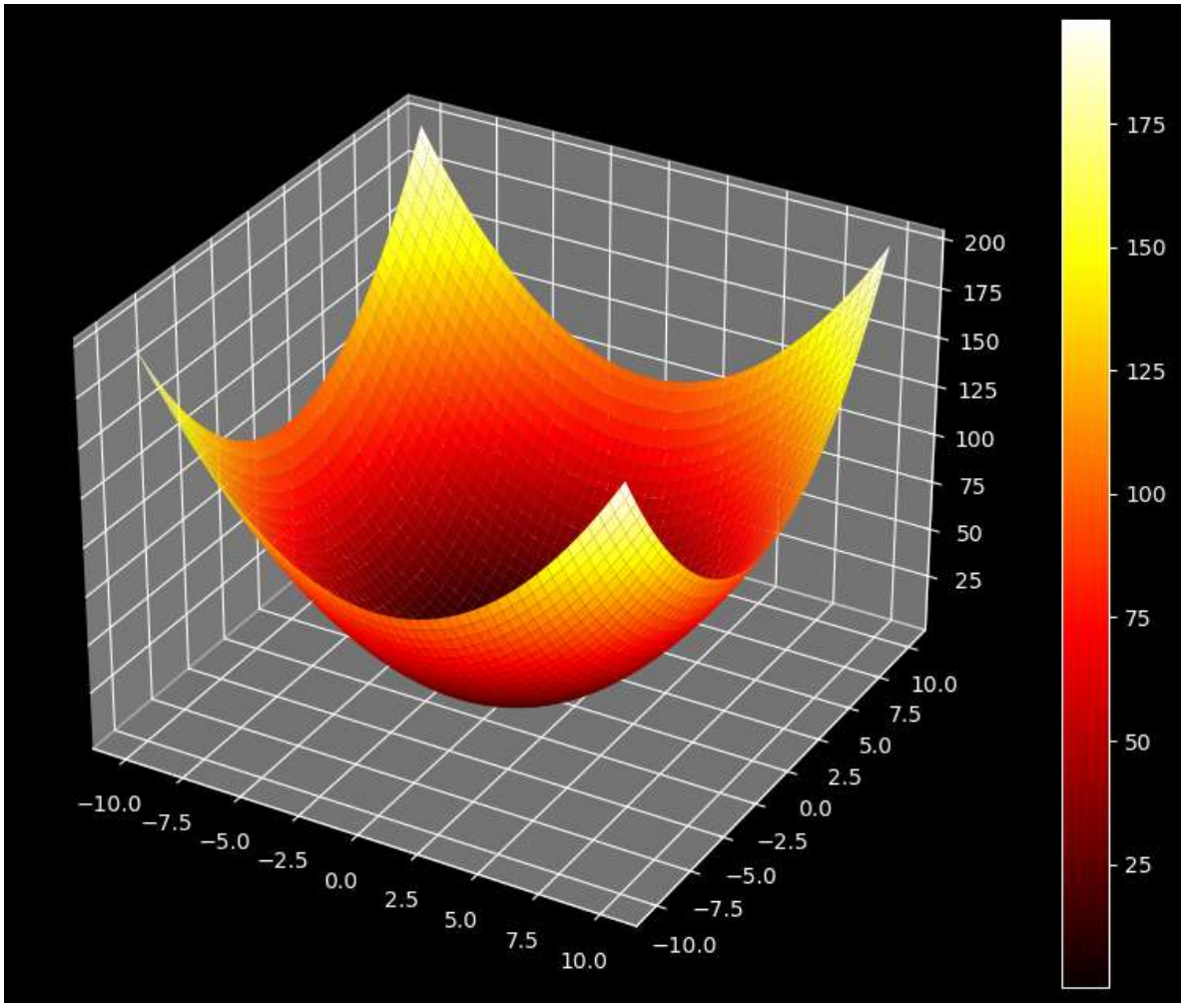
In [ ]:
```
fig = plt.figure(figsize=(12,8))

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

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

Out[ ]:
```
<mpl_toolkits.mplot3d.art3d.Poly3DCollection at 0x17f3ec60d50>
```



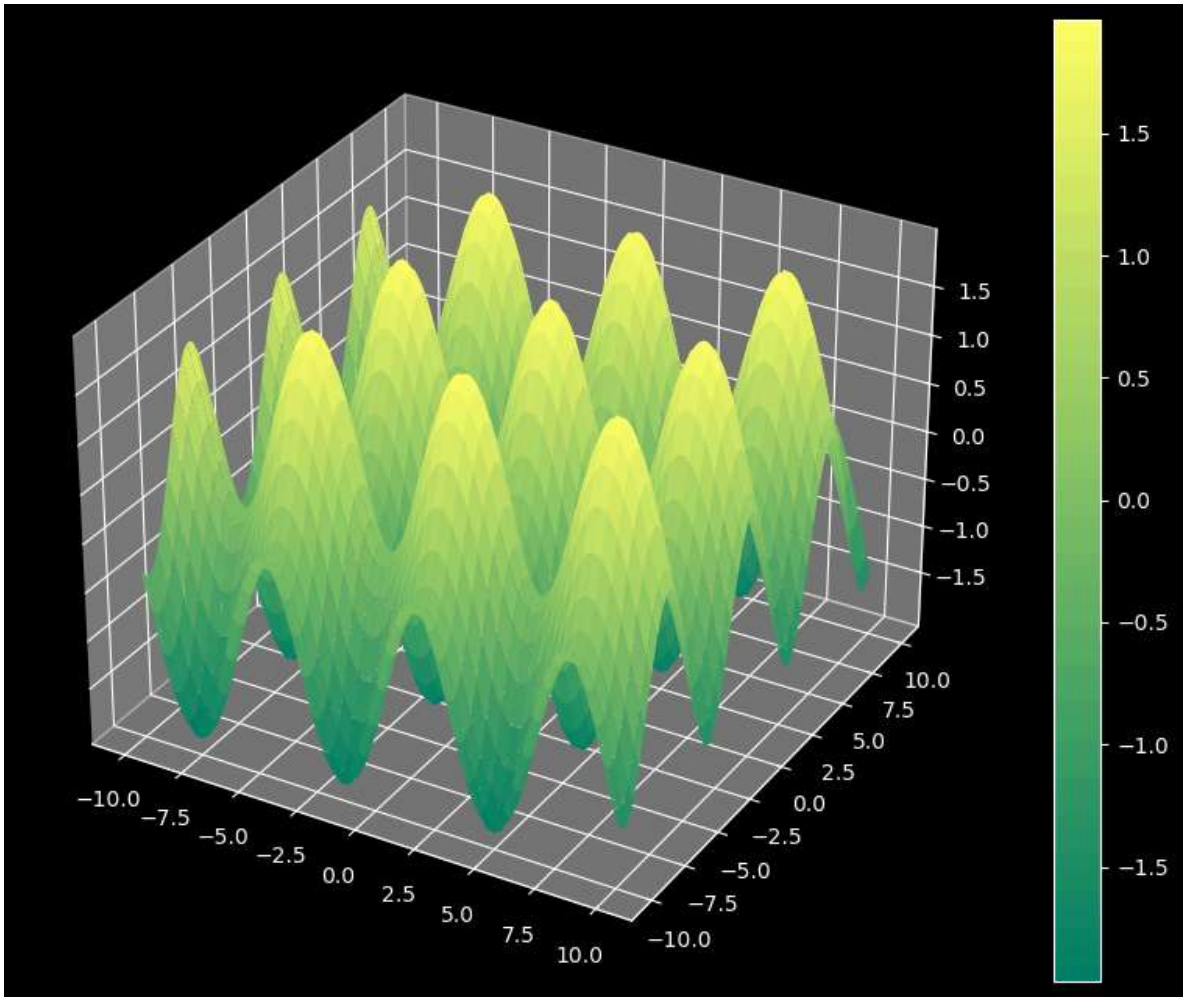
In [ ]:
```
fig = plt.figure(figsize=(12,8))

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

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

Out[ ]:
```
<matplotlib.colorbar.Colorbar at 0x17f40517b10>
```



In [ ]:
```
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='summer')
fig.colorbar(p)
```

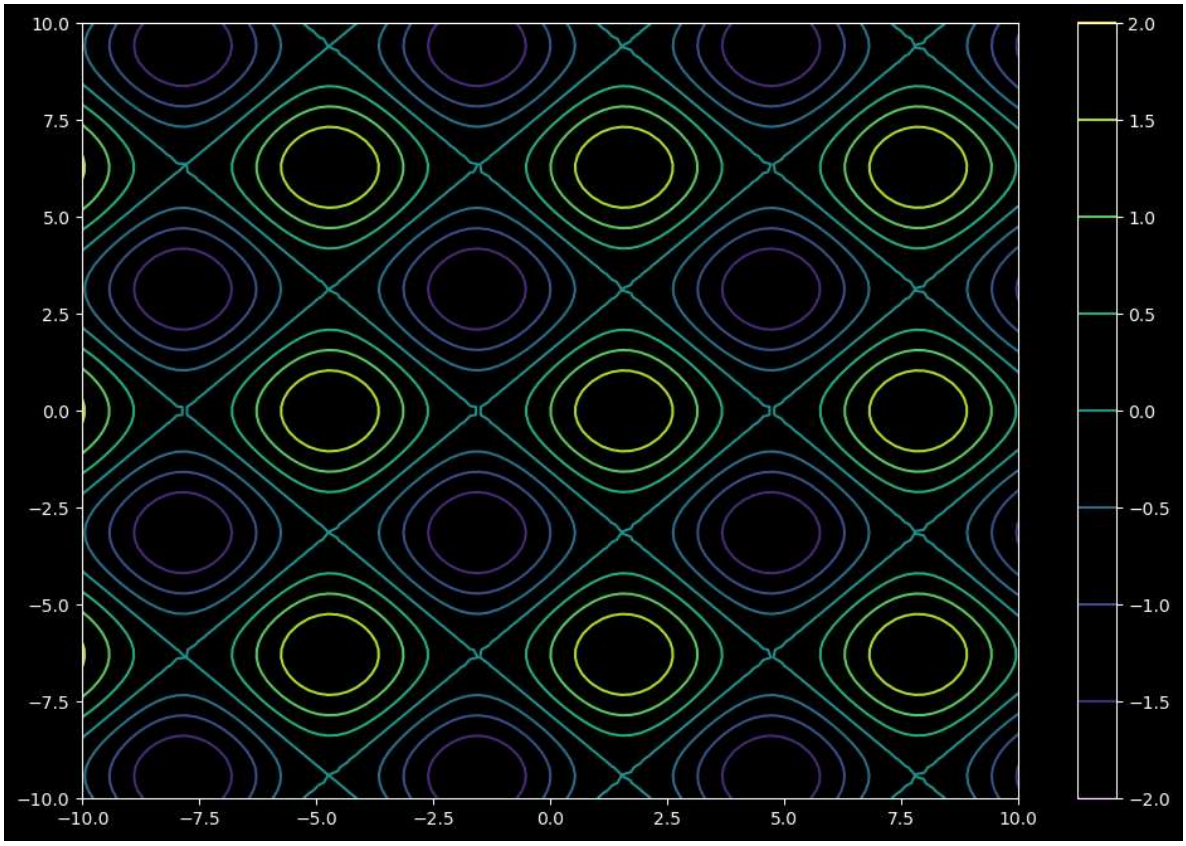
Out[ ]:
```
<matplotlib.colorbar.Colorbar at 0x17f40531310>
```



Contour Plots 3d-2d

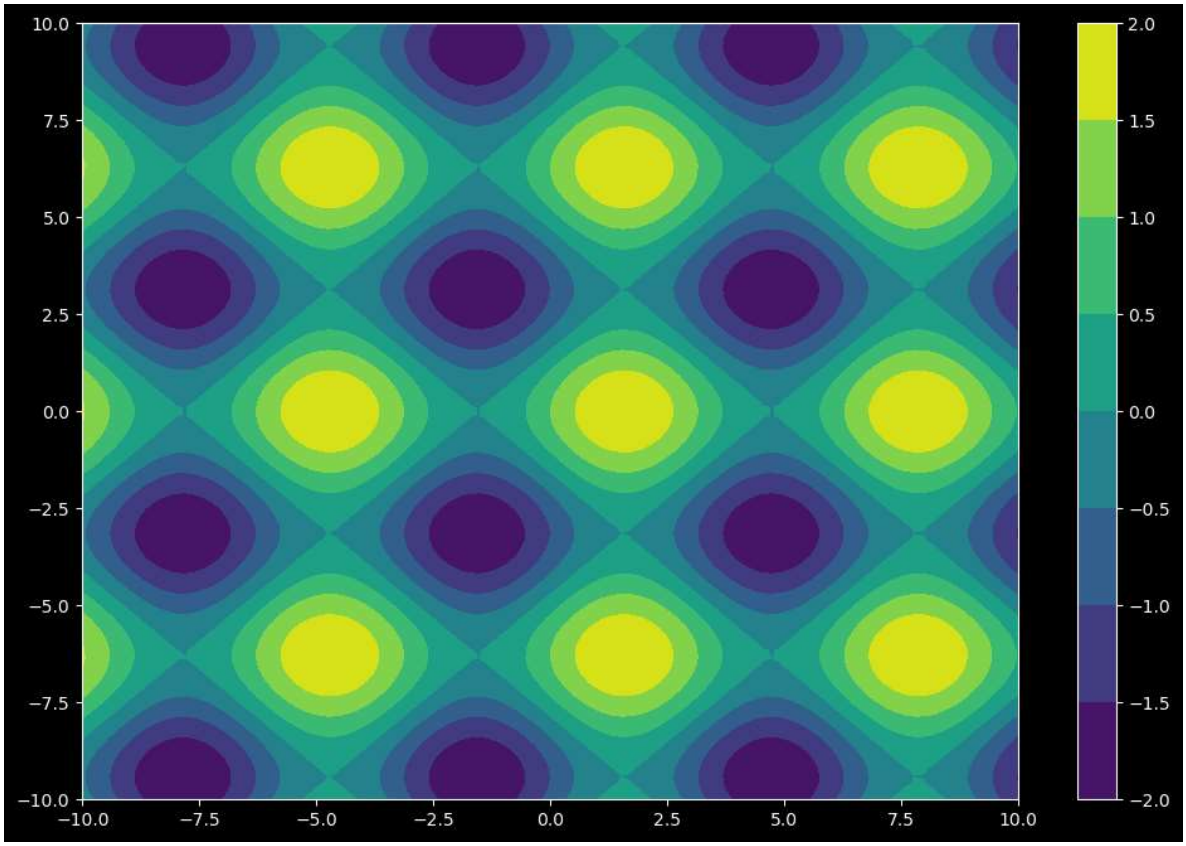
In [ ]:
```
fig = plt.figure(figsize=(12,8))

ax = plt.subplot()

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

Out[ ]:
```
<matplotlib.colorbar.Colorbar at 0x17f3f8da210>
```



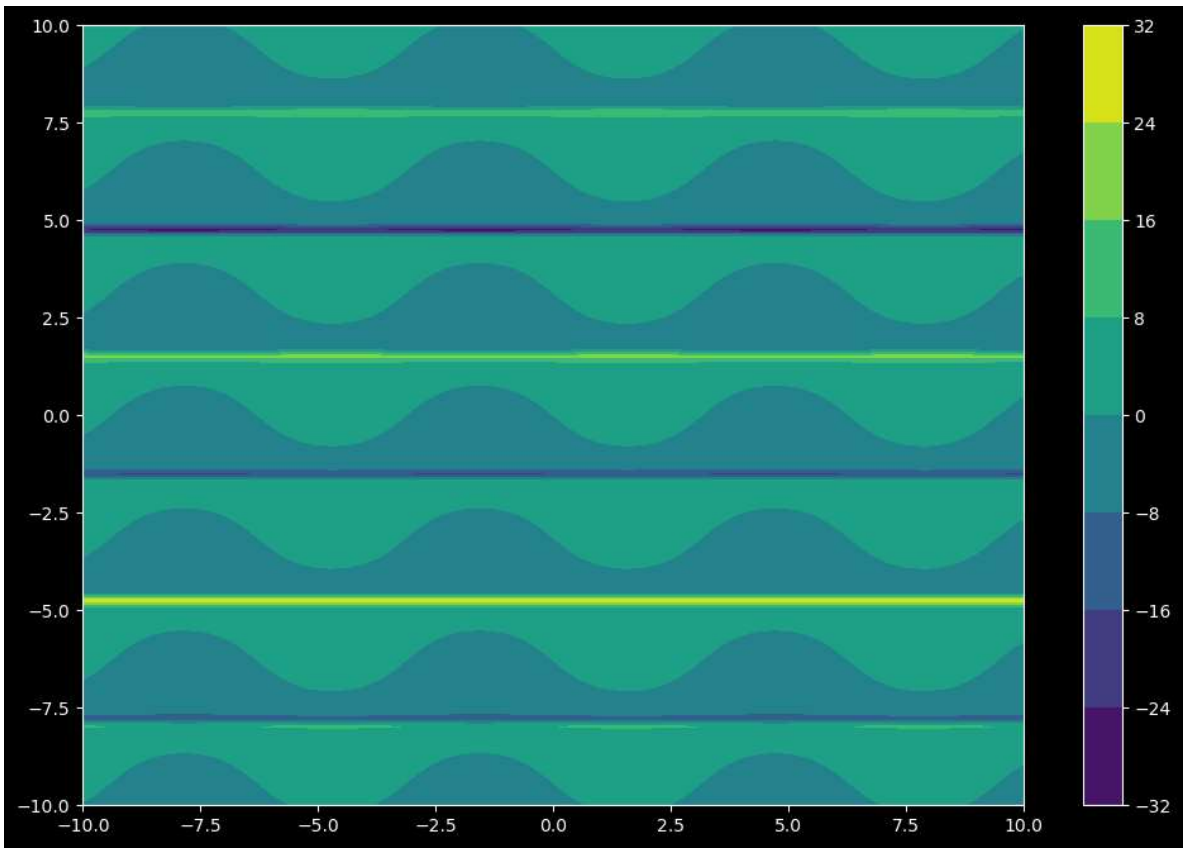
In [ ]:
```
fig = plt.figure(figsize=(12,8))

ax = plt.subplot()

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

Out[ ]: <matplotlib.colorbar.Colorbar at 0x17f401e9350>



In [ ]:
```
z = np.sin(xx) + np.tan(yy)

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

ax = plt.subplot()

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

Out[ ]:
```
<matplotlib.colorbar.Colorbar at 0x17f3f5aa890>
```



In [ ]:
```
delivery = pd.read_csv('C:/Users/Mohan/Downloads/Matplotlib/IPL_Ball_by_Ball_2008_
delivery.head()
```

Out[ ]:

	ID	innings	overs	ballnumber	batter	bowler	non-striker	extra_type	batsm
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	

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

In [ ]:
```
grouped_temp_df = temp_df.groupby('batter').size().reset_index(name='count')
```

In [ ]:
```
grouped_temp_df.sort_values(by='count',ascending=False)
```

Out[ ]:

	batter	count
71	CH Gayle	337
13	AB de Villiers	241
288	RG Sharma	234
226	MS Dhoni	217
376	V Kohli	215
...	...	...
181	L Balaji	1
182	L Ronchi	1
184	LH Ferguson	1
321	SD Lad	1
198	M Rawat	1

396 rows × 2 columns

Heatmap

In [ ]:
```
grid = temp_df.pivot_table(index='overs',columns='ballnumber',values='batsman_run
```

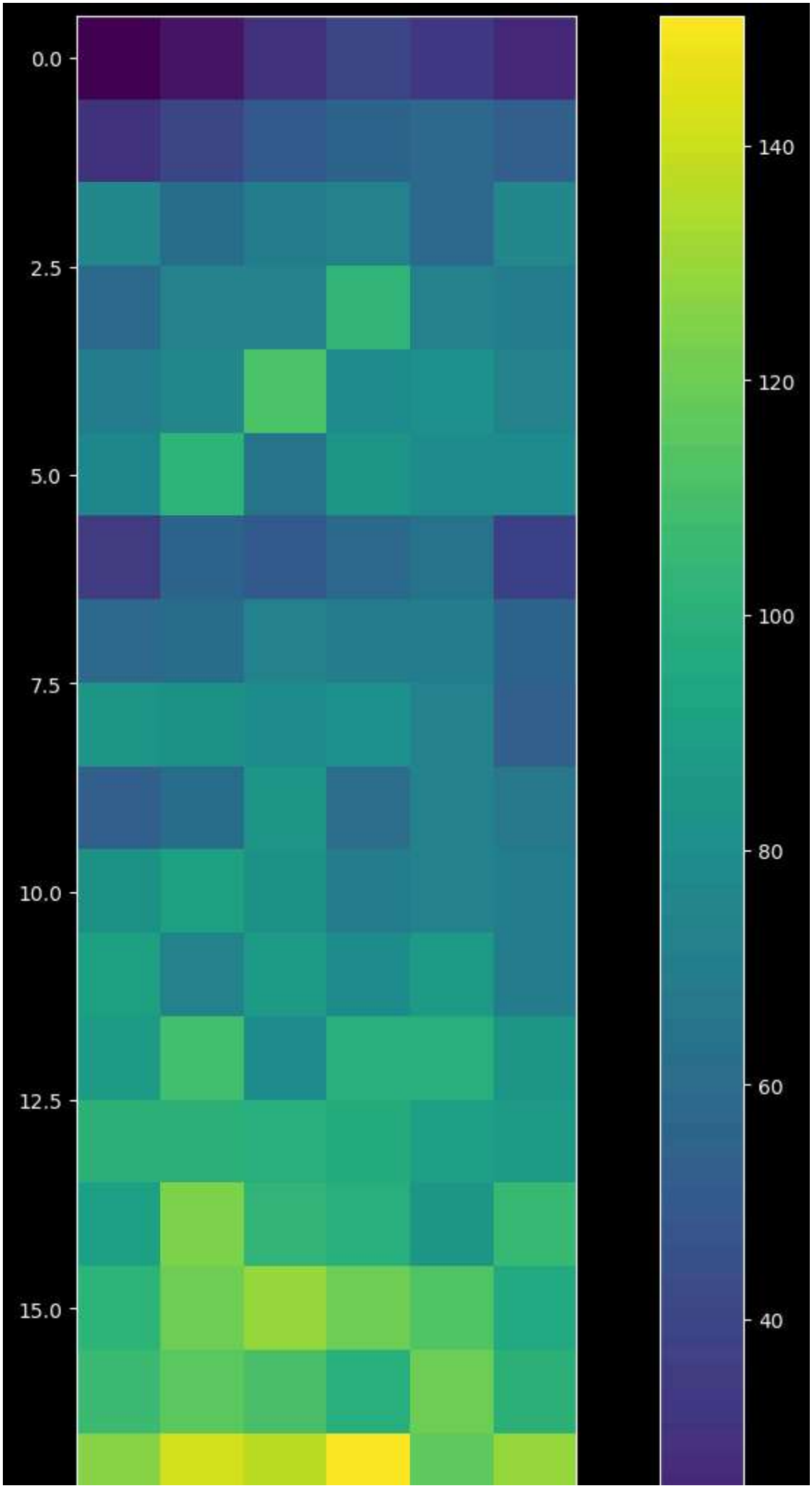
In [ ]:
```
grid
```

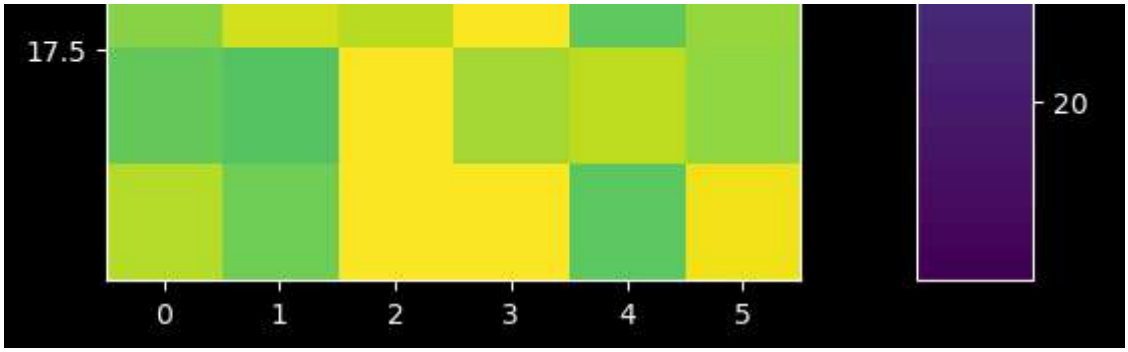
Out[ ]:

ballnumber overs	1	2	3	4	5	6
0	9	17	31	39	33	27
1	31	40	49	56	58	54
2	75	62	70	72	58	76
3	60	74	74	103	74	71
4	71	76	112	80	81	72
5	77	102	63	86	78	80
6	34	56	49	59	64	38
7	59	62	73	70	69	56
8	86	83	79	81	73	52
9	54	62	86	61	74	67
10	82	92	83	69	72	70
11	91	72	87	79	87	70
12	87	109	79	100	100	84
13	101	101	99	97	90	88
14	90	124	103	100	86	106
15	102	120	129	121	113	96
16	107	115	111	100	120	101
17	126	142	137	151	117	129
18	118	114	151	132	138	128
19	136	120	151	151	116	148

In [ ]:
```
plt.figure(figsize=(15,15))
plt.imshow(grid)
plt.colorbar()
```

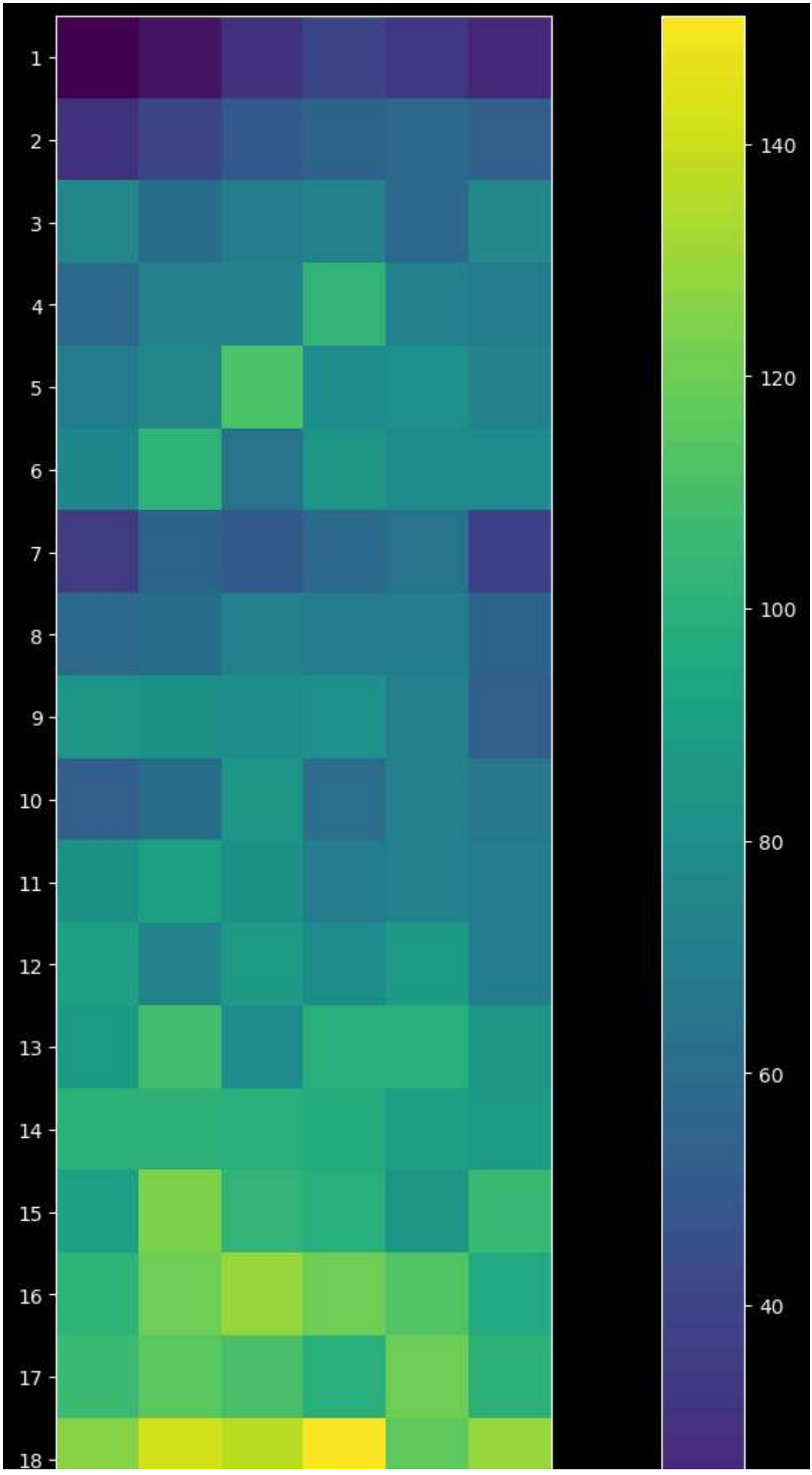
Out[ ]:
```
<matplotlib.colorbar.Colorbar at 0x17f41f60950>
```

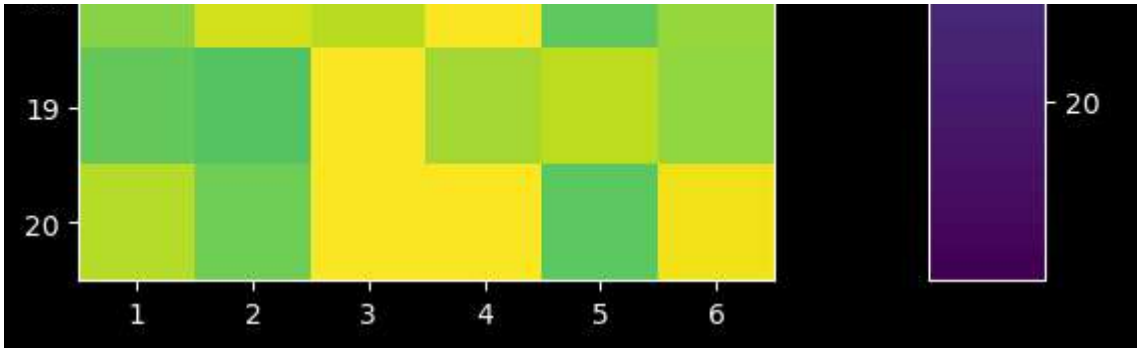

0.0
2.5
5.0
7.5
10.0
12.5
15.0
140
120
100
80
60
40



In [ ]:
```
plt.figure(figsize=(20,15))
plt.imshow(grid)
plt.yticks(delivery['overs'].unique(), list(range(1,21)))
plt.xticks(np.arange(0,6), list(range(1,7)))
plt.colorbar()
```

Out[ ]: <matplotlib.colorbar.Colorbar at 0x17f42055310>


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
140
120
100
80
60
40



In [ ]:

In [ ]: