Matplotlib: Visualization with Python

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

In [1]:
```
# import the library

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

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

Types of Data

- Numerical Data
- Categorical Data

Certainly! Here's a shorter version of data:

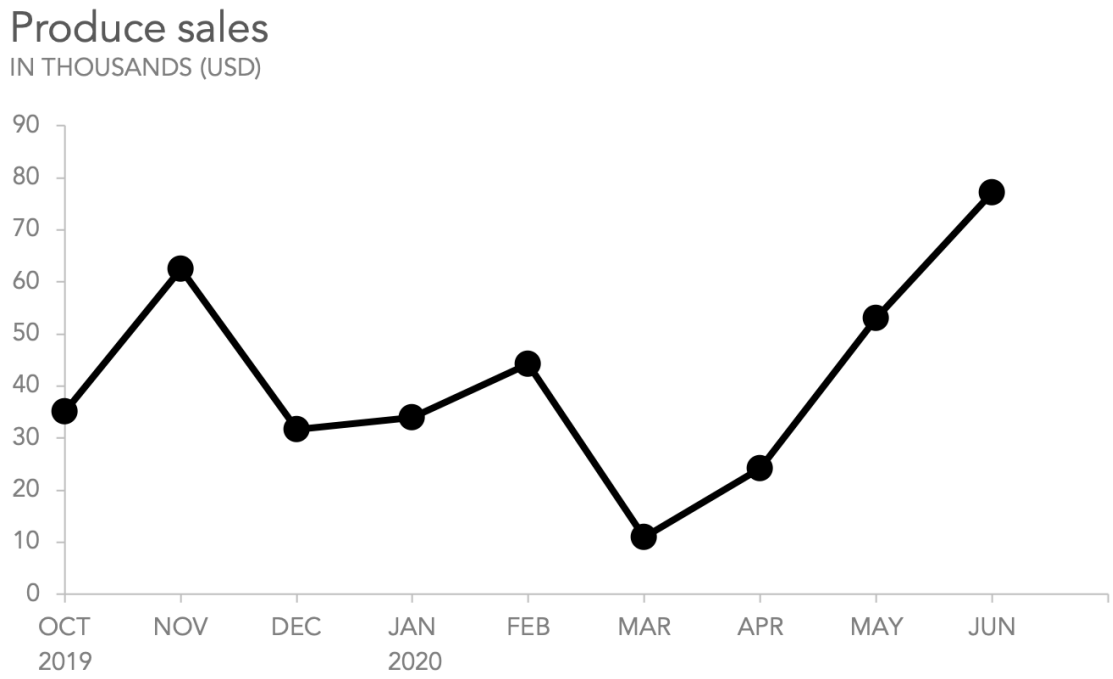
Single column: Univariate Analysis (Analyzing a single variable)

Two columns: Bivariate Analysis (Analyzing the relationship between two variables)

Two or more columns: Multivariate Analysis (Analyzing multiple variables simultaneously)

* 2D line Plot

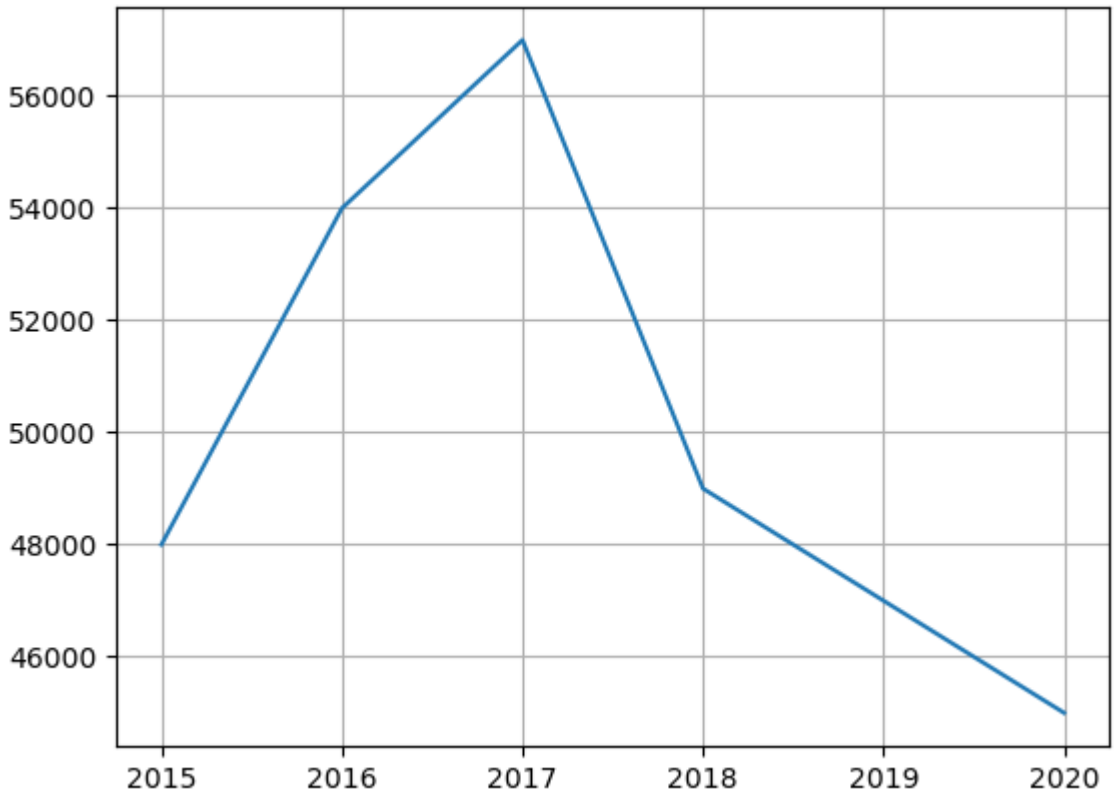
A 2D line plot in matplotlib is a graphical representation of data points connected by **straight lines on a two-dimensional coordinate system**. It is a basic type of plot used to visualize the relationship between **two continuous variables.**



2D line PLot can be used on

- Bivariate Analysis
- categorical -> numerical and numerical -> numerical
- Use case - Time series data

In [2]:
```
# plotting a simple function

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

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



In [3]:
```
# from a pandas dataframe

batsman = pd.read_csv("sharma-kohli.csv")
```

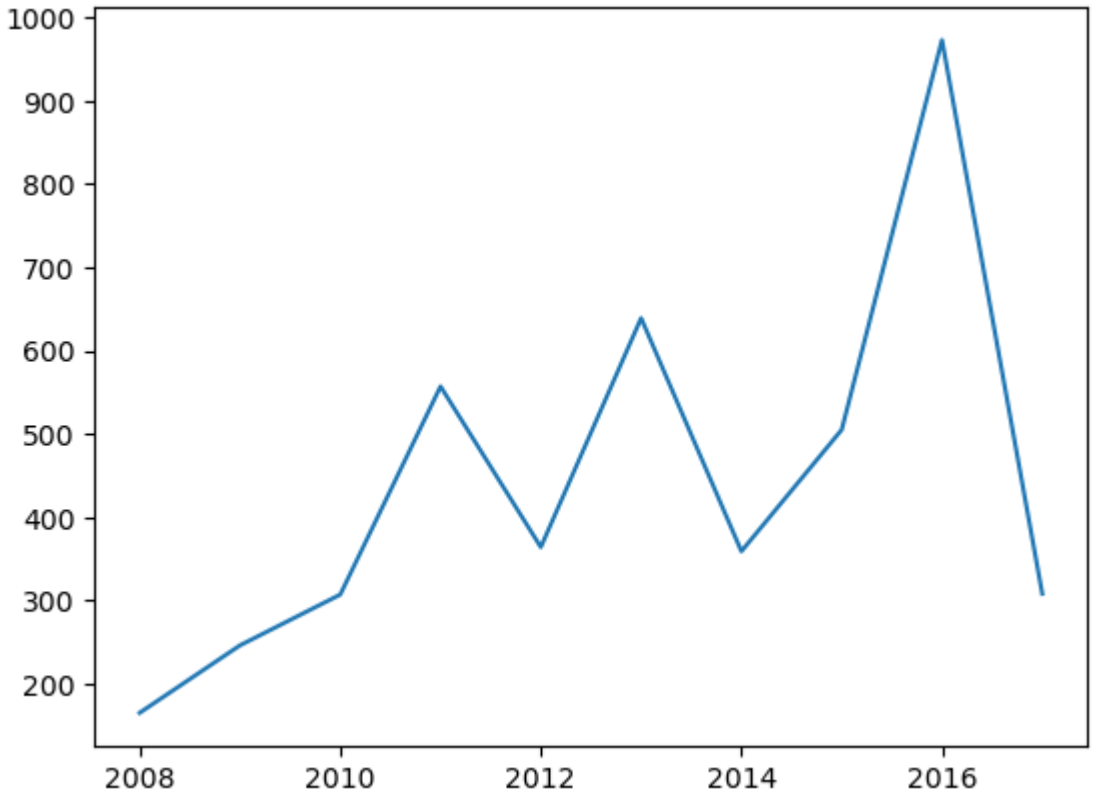
In [4]:
```
batsman
```

Out[4]:

	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

In [5]:
```
plt.plot(batsman['index'],batsman['V Kohli'])
```

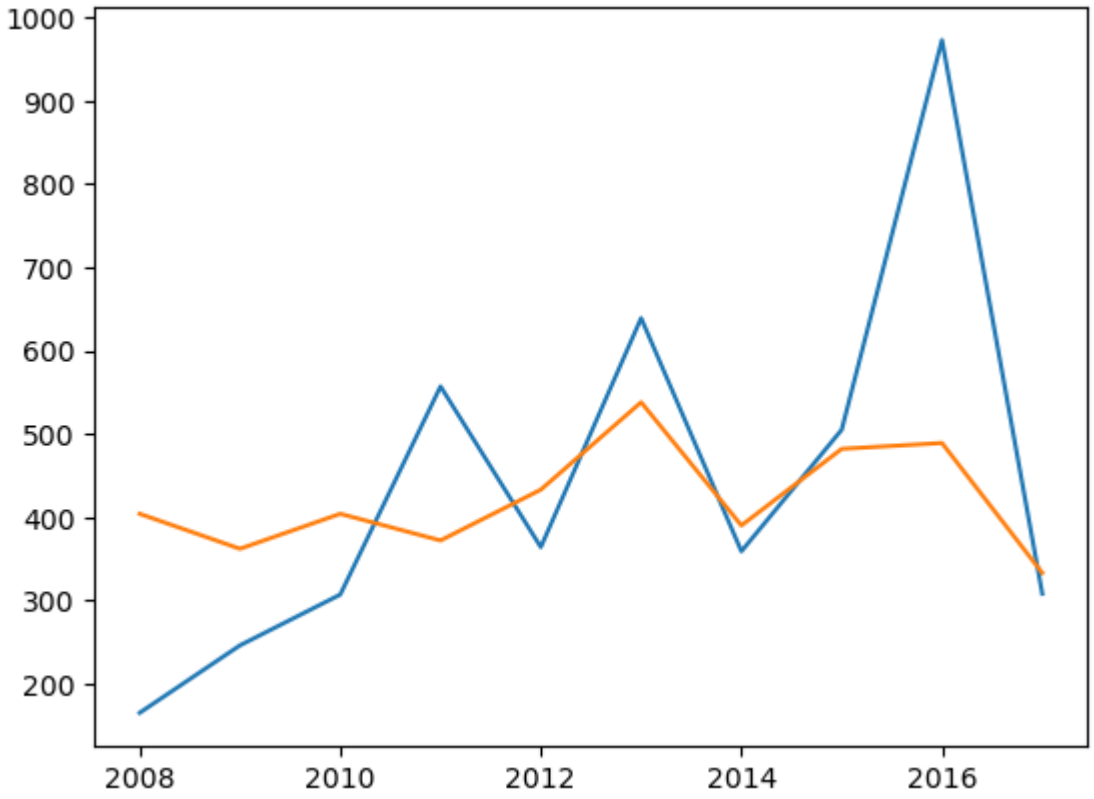
Out[5]:
```
[<matplotlib.lines.Line2D at 0x1ab0d70c790>]
```



plotting multiple plots

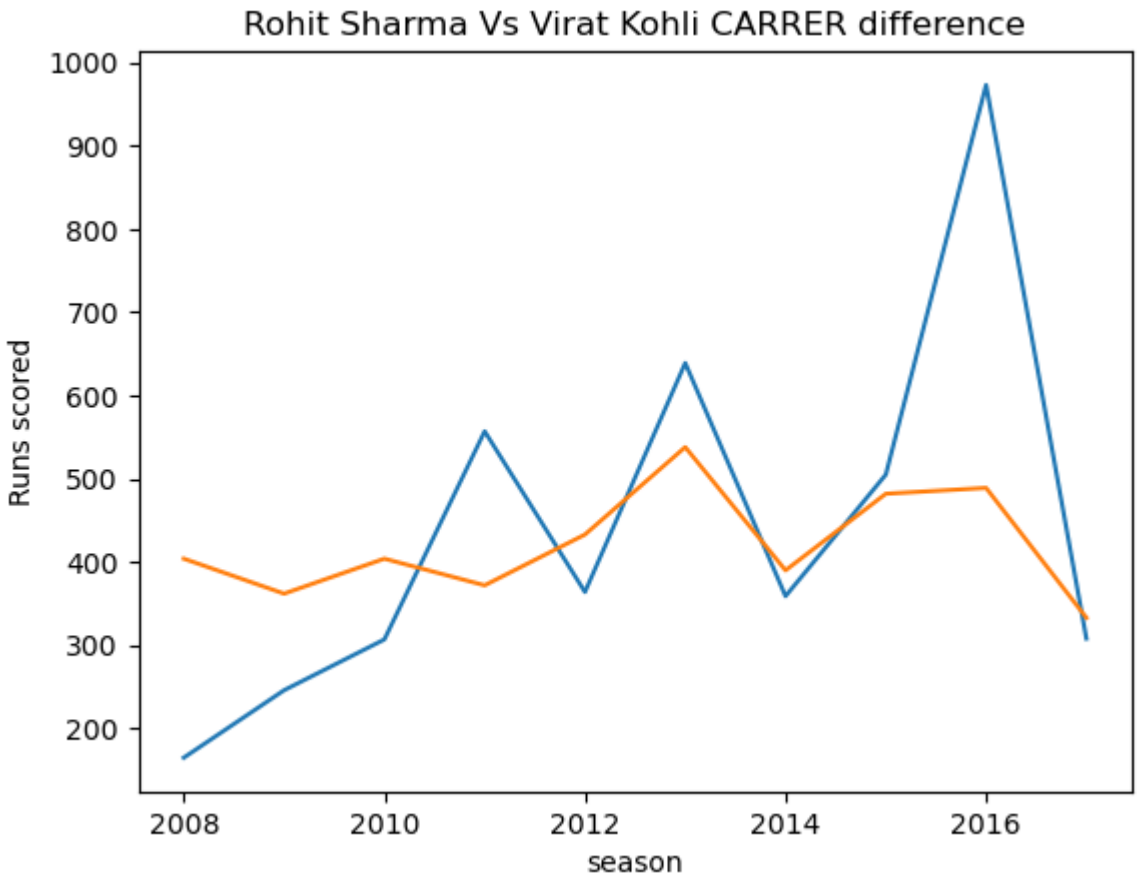
In [6]:
```
plt.plot(batsman['index'],batsman['V Kohli'])

plt.plot(batsman['index'],batsman['RG Sharma'])
```

Out[6]:
```
[<matplotlib.lines.Line2D at 0x1ab0df11b80>]
```



In [7]:
```
# labels title

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

plt.title('Rohit Sharma Vs Virat Kohli CARRER difference')

plt.xlabel('season')

plt.ylabel('Runs scored')
```

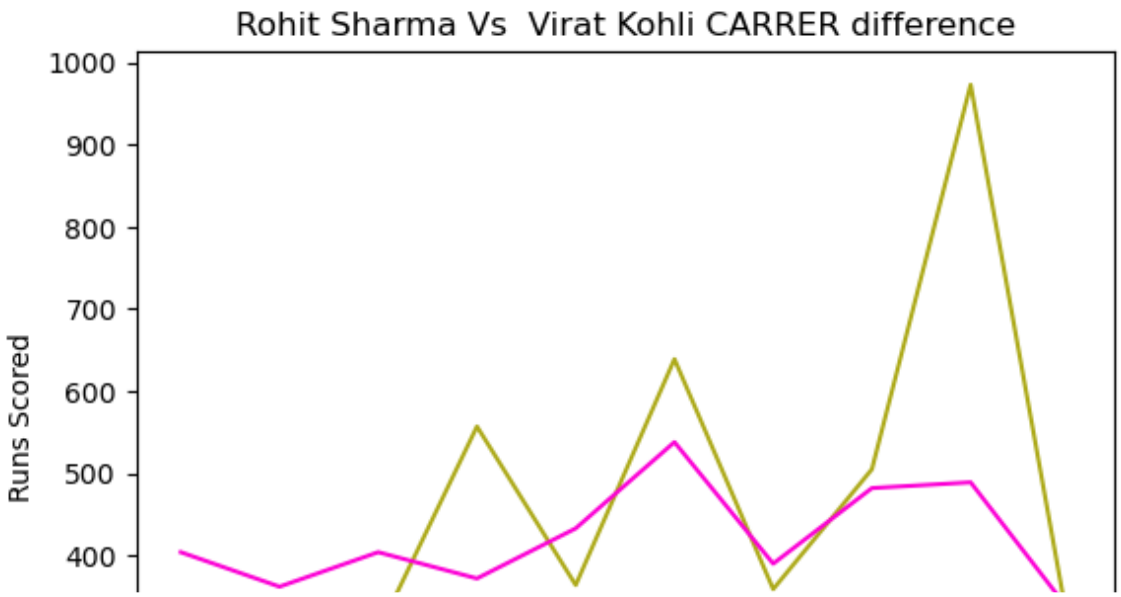
Out[7]:
```
Text(0, 0.5, 'Runs scored')
```



colors(hex)

In [8]:
```
# colors(hex)   # Hex codes

plt.plot(batsman['index'],batsman['V Kohli'],color='#ADAA19') # Yellow
plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6') # pink

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

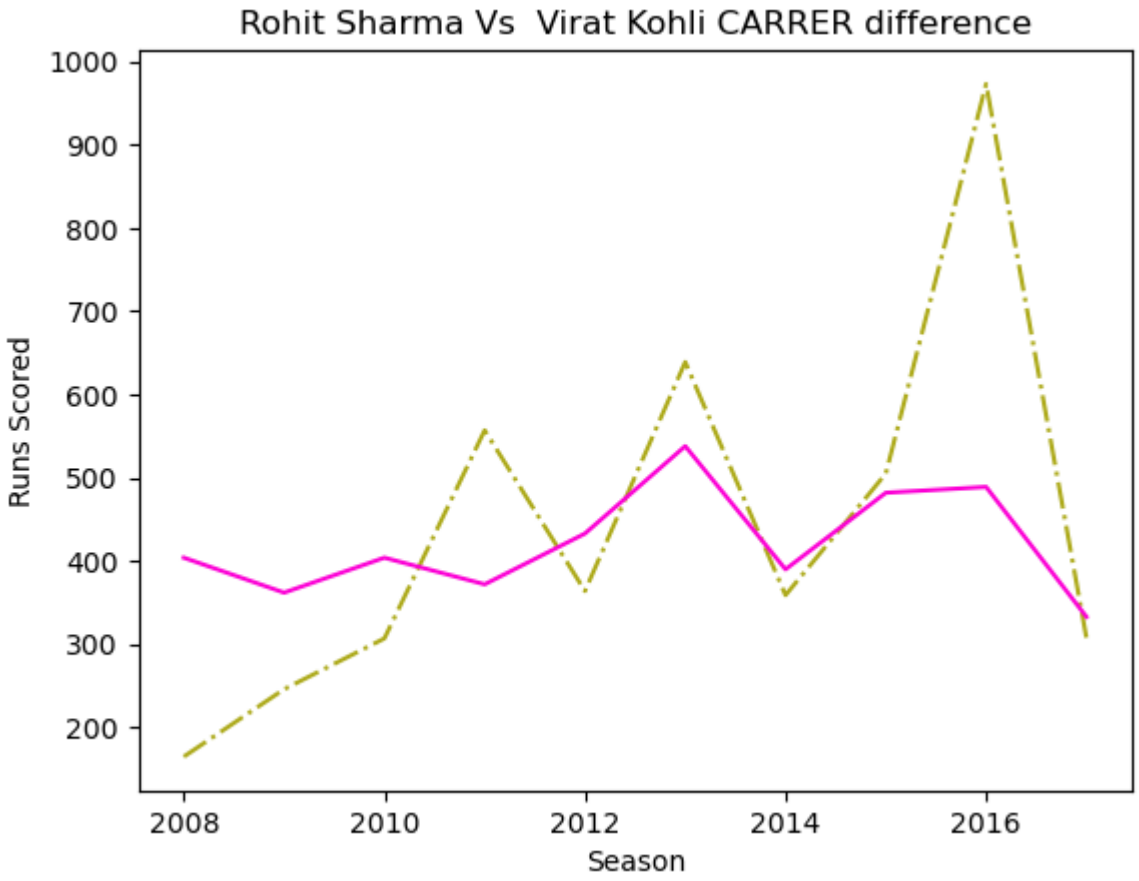
Out[8]:
```
Text(0, 0.5, 'Runs Scored')
```



Dashed Lines

In [9]:
```
plt.plot(batsman['index'],batsman['V Kohli'],color='#ADAA19',
         linestyle = 'dashdot') # dashdot

plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6')

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

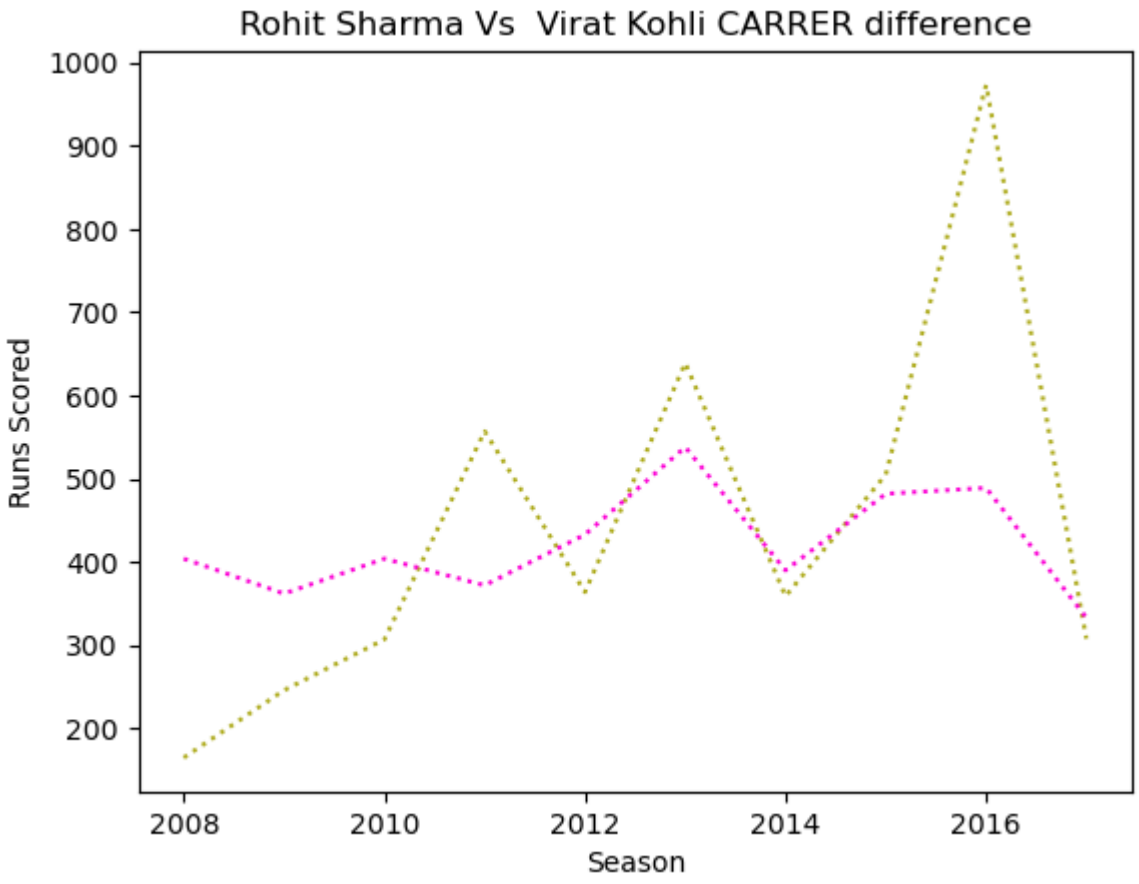
Out[9]: Text(0, 0.5, 'Runs Scored')



dotted lines

In [10]:
```
plt.plot(batsman['index'],batsman['V Kohli'],color='#ADAA19',
         linestyle = 'dotted') # dotted


plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6',
         linestyle ='dotted') #  dotted

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

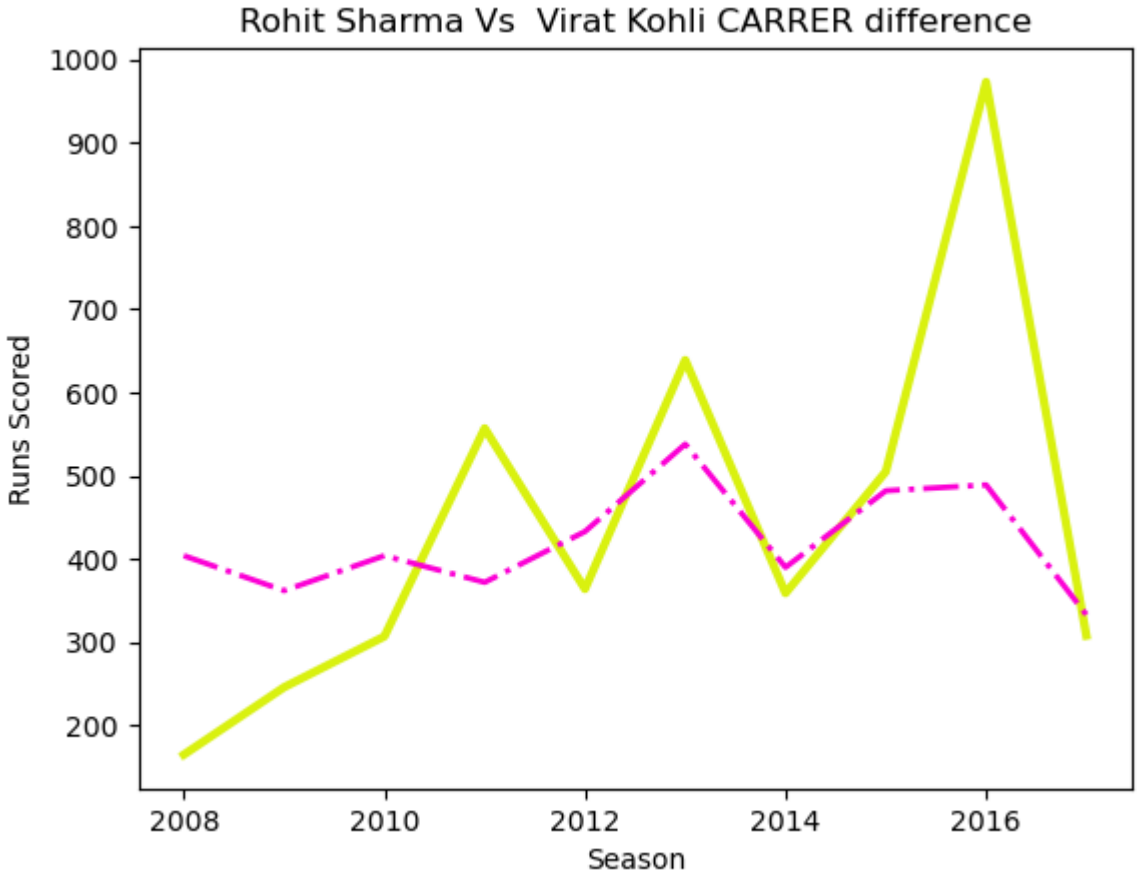
Out[10]: Text(0, 0.5, 'Runs Scored')



line width

In [12]:
```
plt.plot(batsman['index'],batsman['V Kohli'],color='#D9F10F'
         ,linestyle='solid',linewidth=3) # linewidth

plt.plot(batsman['index'],batsman['RG Sharma'],
         color='#FC00D6',linestyle='dashdot',linewidth=2)

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

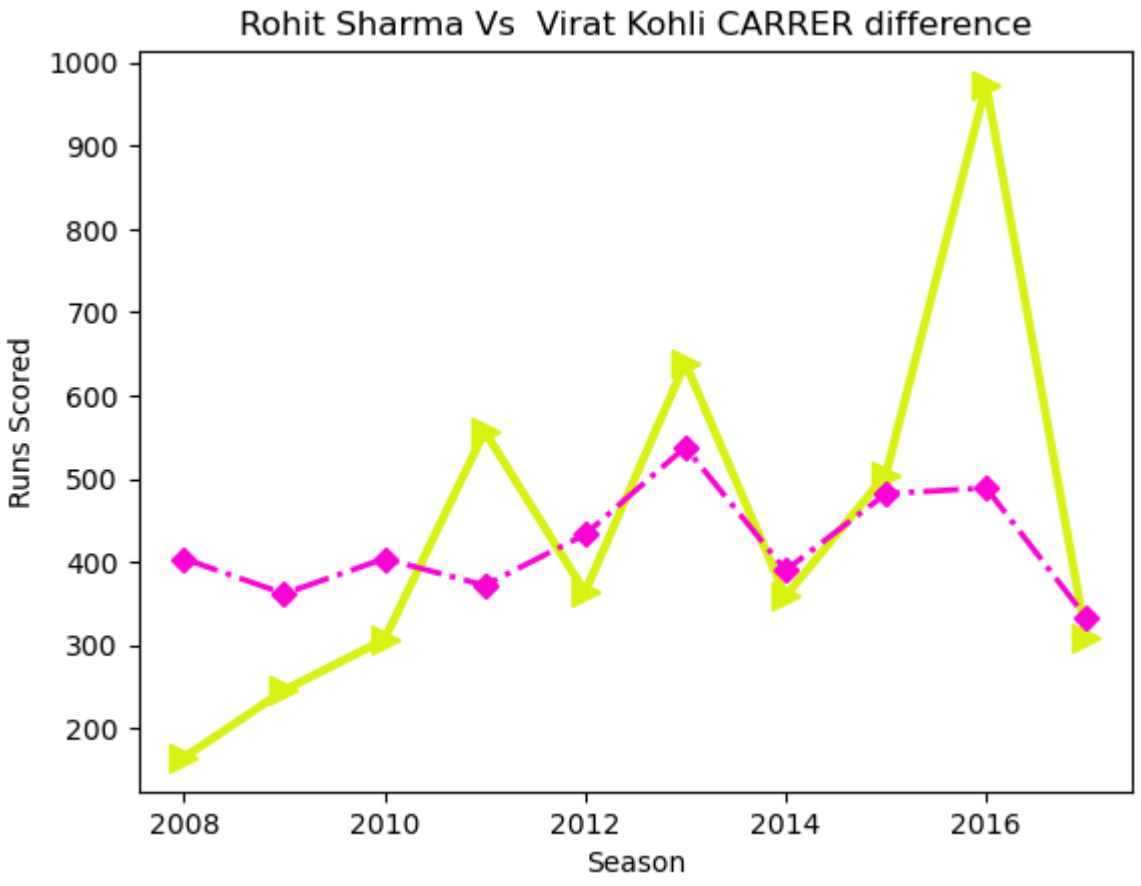
Out[12]: Text(0, 0.5, 'Runs Scored')



marker(size)

In [16]:
```
plt.plot(batsman['index'],batsman['V Kohli'],color='#D9F10F'
         ,linestyle='solid',linewidth=3,
         marker='>' , markersize =10) # Marker + marker size

plt.plot(batsman['index'],batsman['RG Sharma'],
         color='#FC00D6',linestyle='dashdot',
         linewidth=2 ,marker='D') # Marker ...> D- Diamond

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

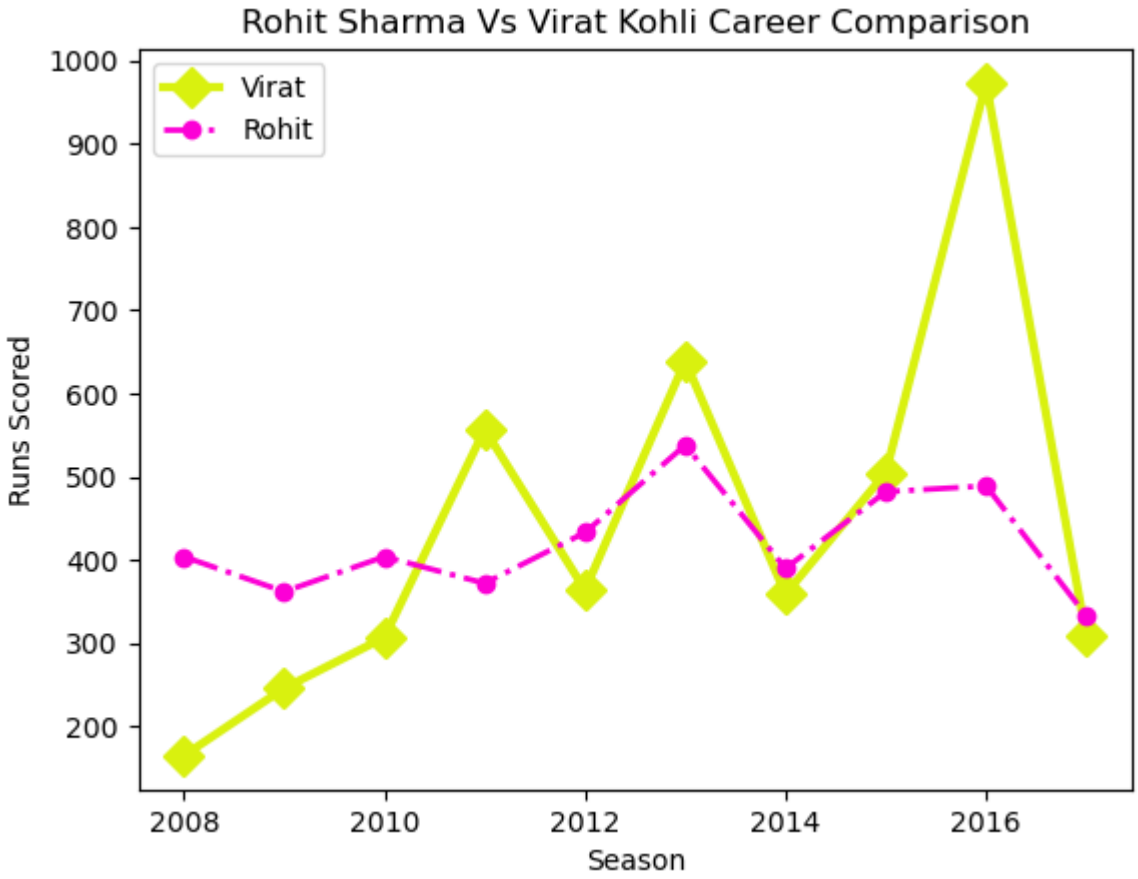
Out[16]: Text(0, 0.5, 'Runs Scored')



Legend

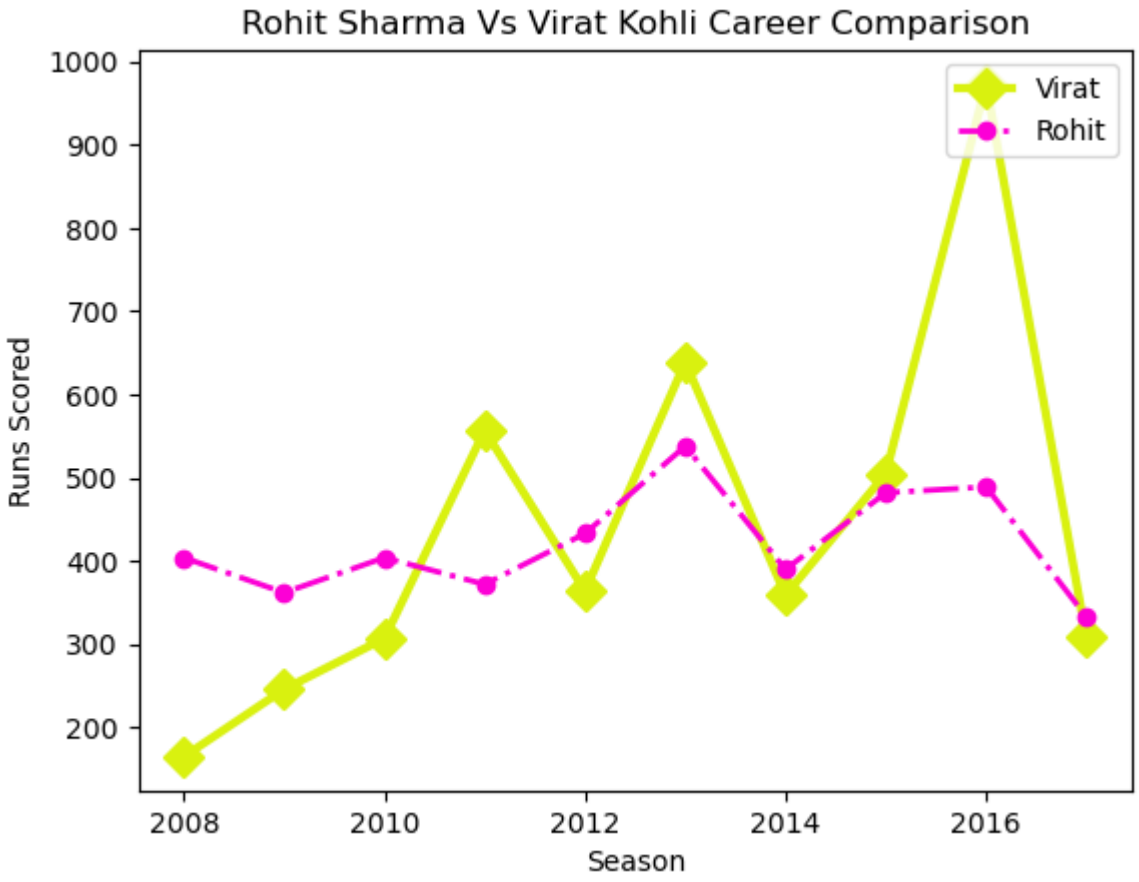
In [17]:
```
# Legend -

plt.plot(batsman['index'],batsman['V Kohli'],
         color='#D9F10F',linestyle='solid',
         linewidth=3,marker='D',markersize=10,label='Virat')

plt.plot(batsman['index'],batsman['RG Sharma'],
         color='#FC00D6',linestyle='dashdot',
         linewidth=2,marker='o',label='Rohit')

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

plt.legend() # loc = best
```

Out[17]: <matplotlib.legend.Legend at 0x1ab0ffa5ca0>



In [18]:
```
# Legend -> Location

plt.plot(batsman['index'],batsman['V Kohli'],
         color='#D9F10F',linestyle='solid',
         linewidth=3,marker='D',markersize=10,label='Virat')

plt.plot(batsman['index'],batsman['RG Sharma'],
         color='#FC00D6',linestyle='dashdot',linewidth=2,
         marker='o',label='Rohit')

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

plt.legend(loc='upper right')
```

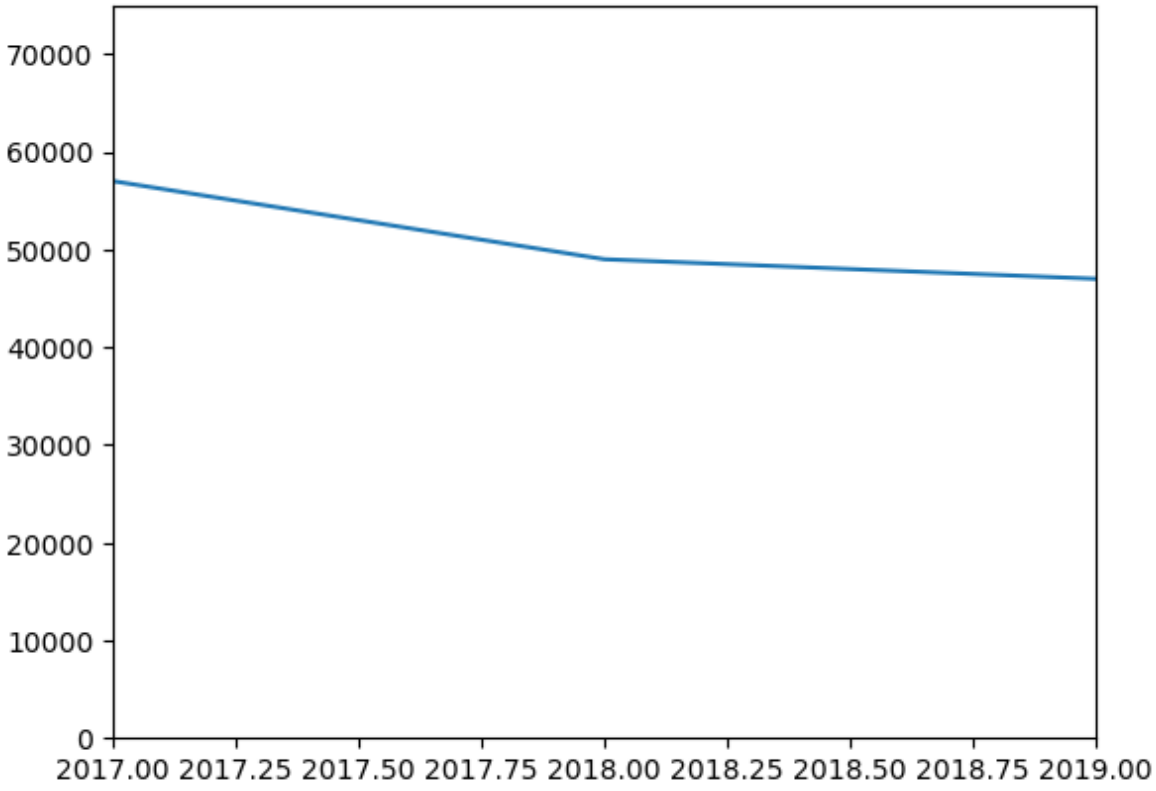
Out[18]: <matplotlib.legend.Legend at 0x1ab0ff27910>



limiting axes

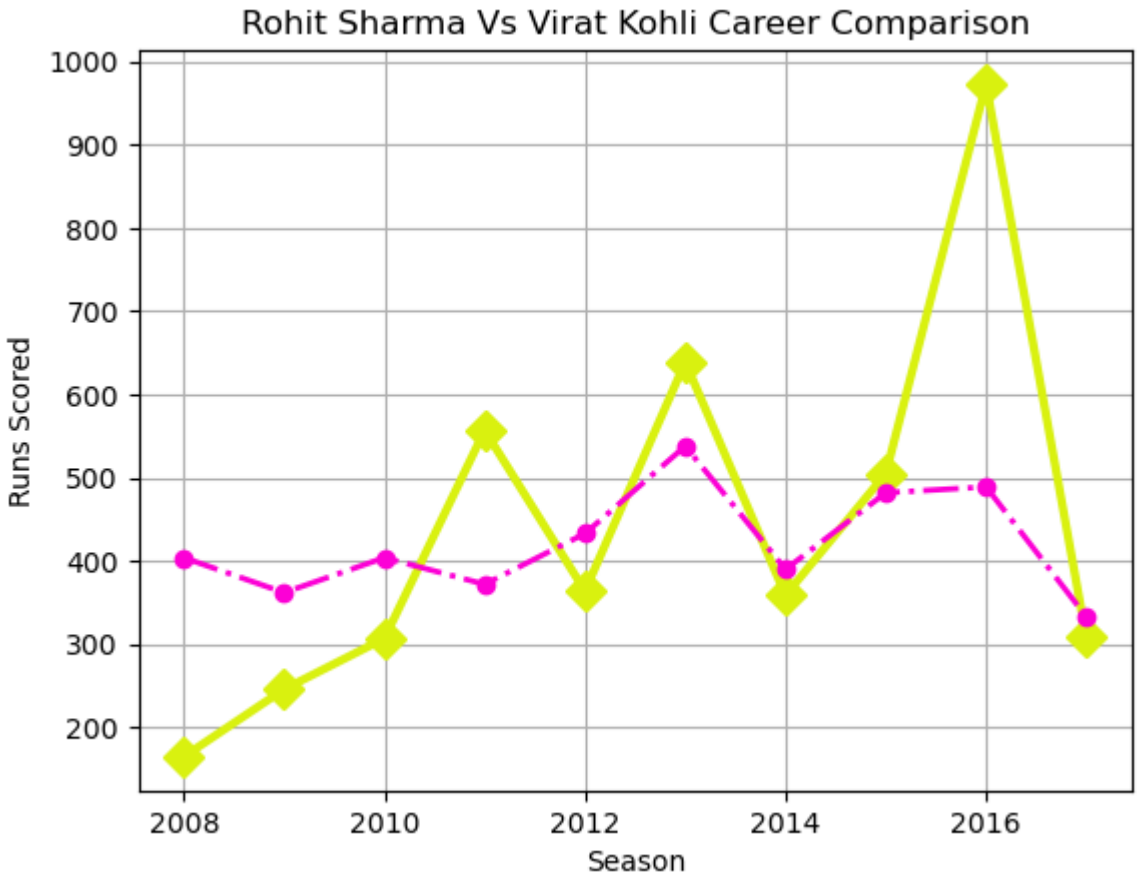
In [19]:
```
# Because of Outliers

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(2017,2019)
```

Out[19]: (2017.0, 2019.0)



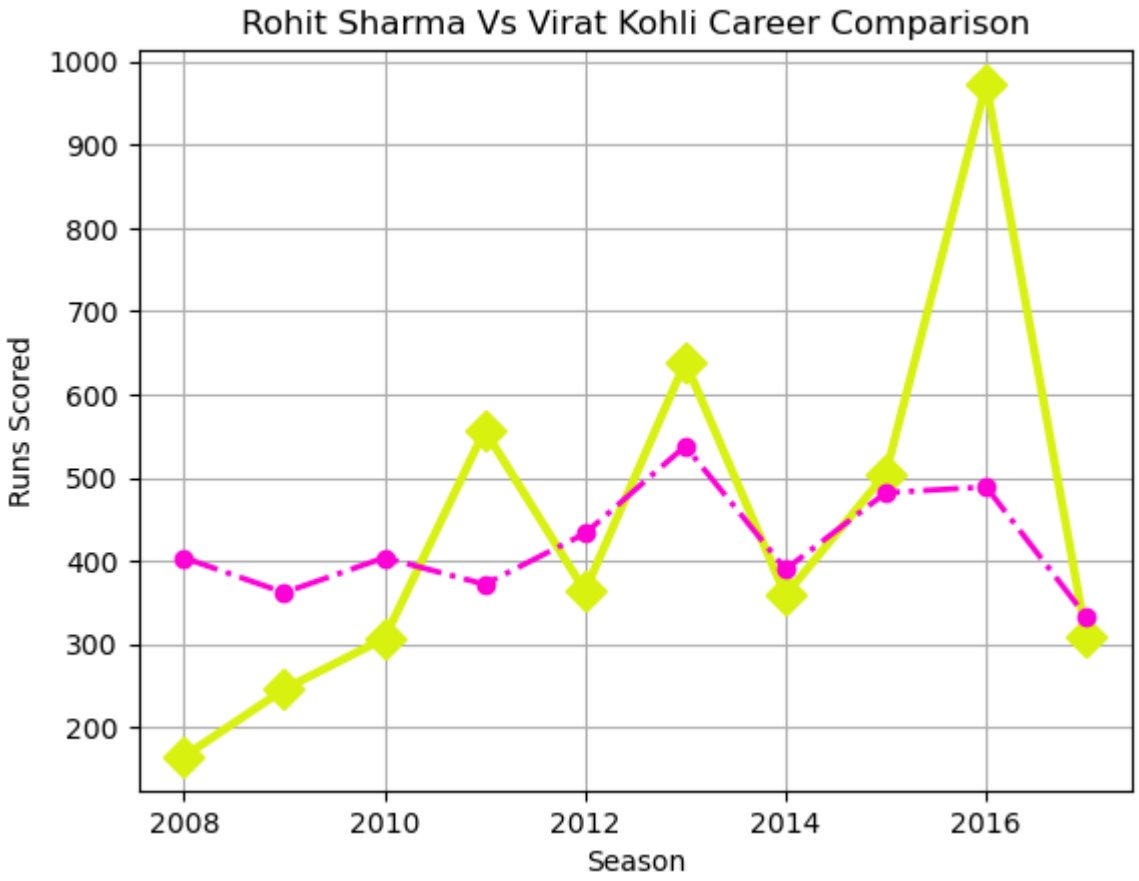
grid

In [20]:
```
plt.plot(batsman['index'],batsman['V Kohli'],
         color='#D9F10F',linestyle='solid',
         linewidth=3,marker='D',markersize=10)

plt.plot(batsman['index'],batsman['RG Sharma'],
         color='#FC00D6',linestyle='dashdot',
         linewidth=2,marker='o')

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

plt.grid()
```



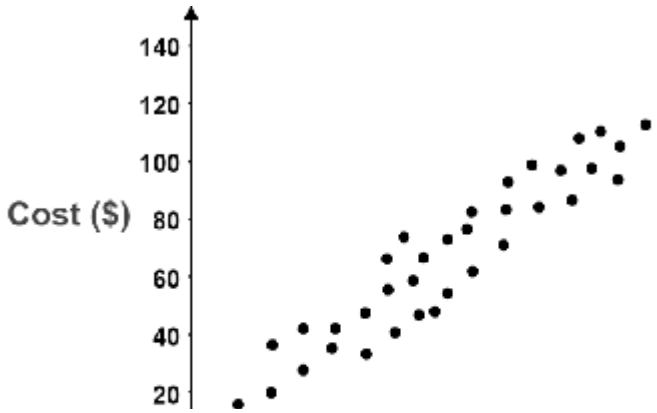
show

In [21]:
```
plt.plot(batsman['index'],batsman['V Kohli'],color='#D9F10F',linestyle='solid'
plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6',linestyle='dash

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

plt.grid()

plt.show()
```



Scatter plot

A scatter plot in matplotlib is a type of plot used to visualize the relationship between two continuous variables. It displays individual data points as markers on a two-dimensional coordinate system, with one variable represented on the x-axis and the other variable represented on the y-axis.



- Bivariate Analysis
- numerical vs numerical
- Use case - Finding correlation

In [22]:
```
# plt.scatter simple function
x = np.linspace(-10,10,50)
x
```

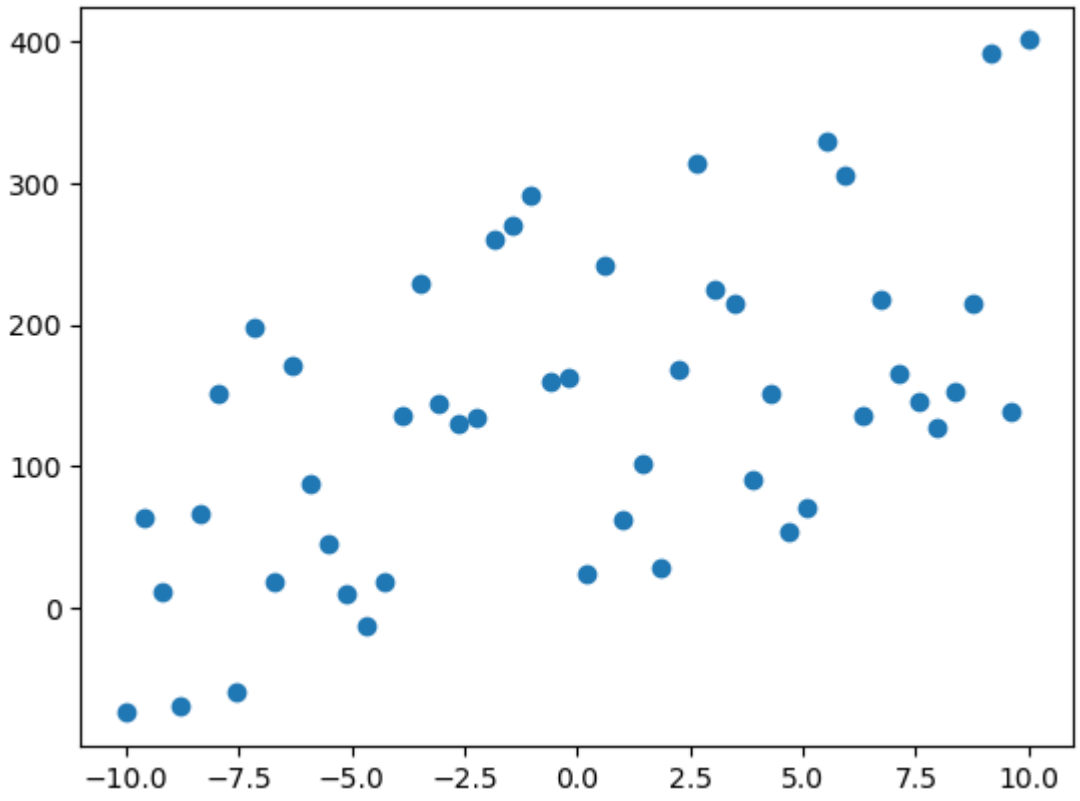
Out[22]:
```
array([-10.        ,  -9.59183673,  -9.18367347,  -8.7755102 ,
        -8.36734694,  -7.95918367,  -7.55102041,  -7.14285714,
        -6.73469388,  -6.32653061,  -5.91836735,  -5.51020408,
        -5.10204082,  -4.69387755,  -4.28571429,  -3.87755102,
        -3.46938776,  -3.06122449,  -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.        ])
```

In [23]:
```
y = 10*x + 3 + np.random.randint(0,300,50)
y
```

Out[23]:
```
array([-74.        ,  64.08163265,  11.16326531, -68.75510204,
        66.32653061, 151.40816327, -59.51020408, 198.57142857,
        18.65306122, 170.73469388,  87.81632653,  45.89795918,
         9.97959184, -12.93877551,  19.14285714, 135.2244898 ,
       229.30612245, 144.3877551 , 130.46938776, 134.55102041,
       260.63265306, 269.71428571, 291.79591837, 159.87755102,
       162.95918367,  24.04081633, 242.12244898,  62.20408163,
       101.28571429,  28.36734694, 168.44897959, 314.53061224,
       225.6122449 , 215.69387755,  90.7755102 , 151.85714286,
        53.93877551,  71.02040816, 330.10204082, 305.18367347,
       136.26530612, 217.34693878, 165.42857143, 145.51020408,
       127.59183673, 152.67346939, 214.75510204, 391.83673469,
       138.91836735, 401.        ])
```

In [24]:
```
plt.scatter(x,y)
```

Out[24]:
```
<matplotlib.collections.PathCollection at 0x1ab1039dd60>
```



In [26]:
```
# plt.scatter on pandas data

df = pd.read_csv("batter.csv")
df = df.head(50)
df
```

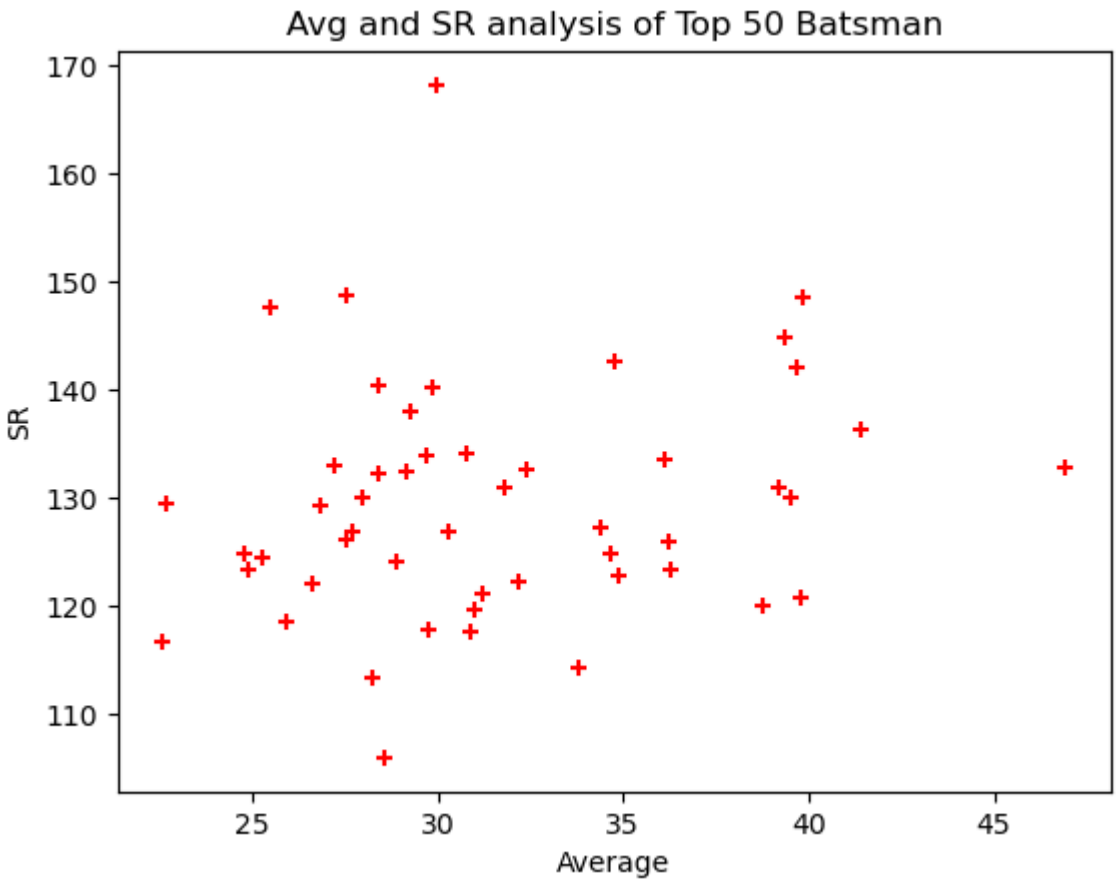
Out[26]:

	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

	batter	runs	avg	strike_rate
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

In [27]:
```
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')
```

Out[27]:
```
Text(0, 0.5, 'SR')
```

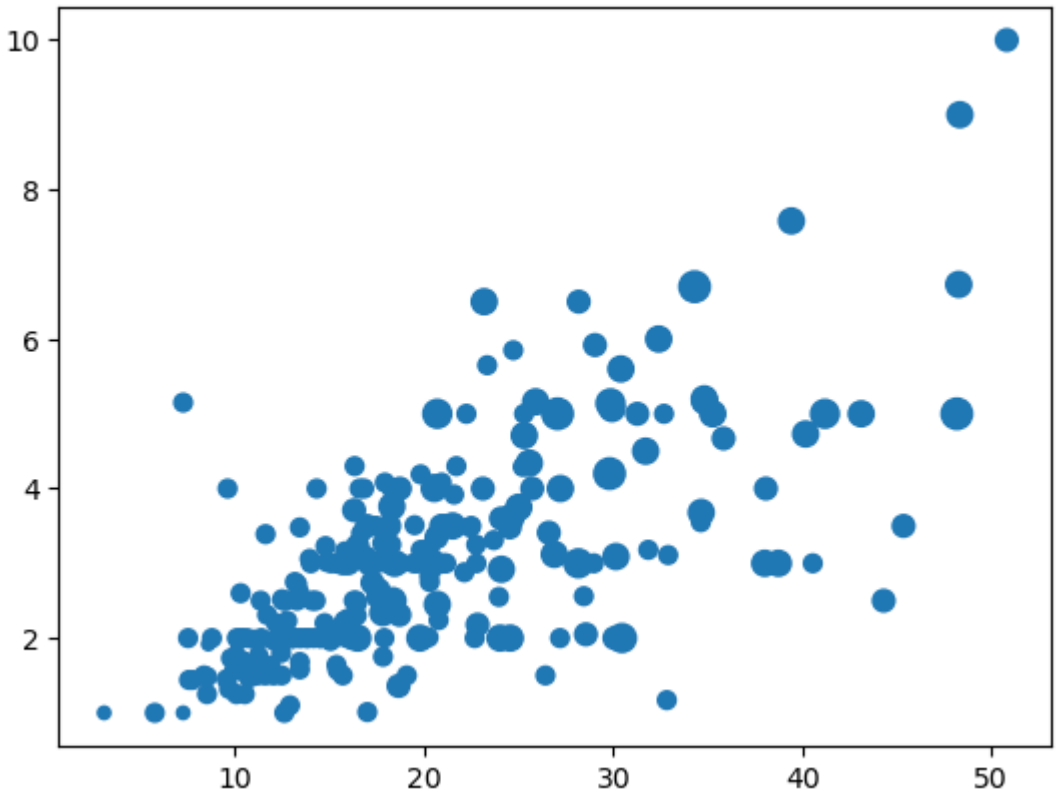


Size

In [31]:
```
sns.load_dataset('tips')

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

In [34]:
```
plt.scatter(tips['total_bill'], tips['tip'] , s = tips ['size']*20)
```

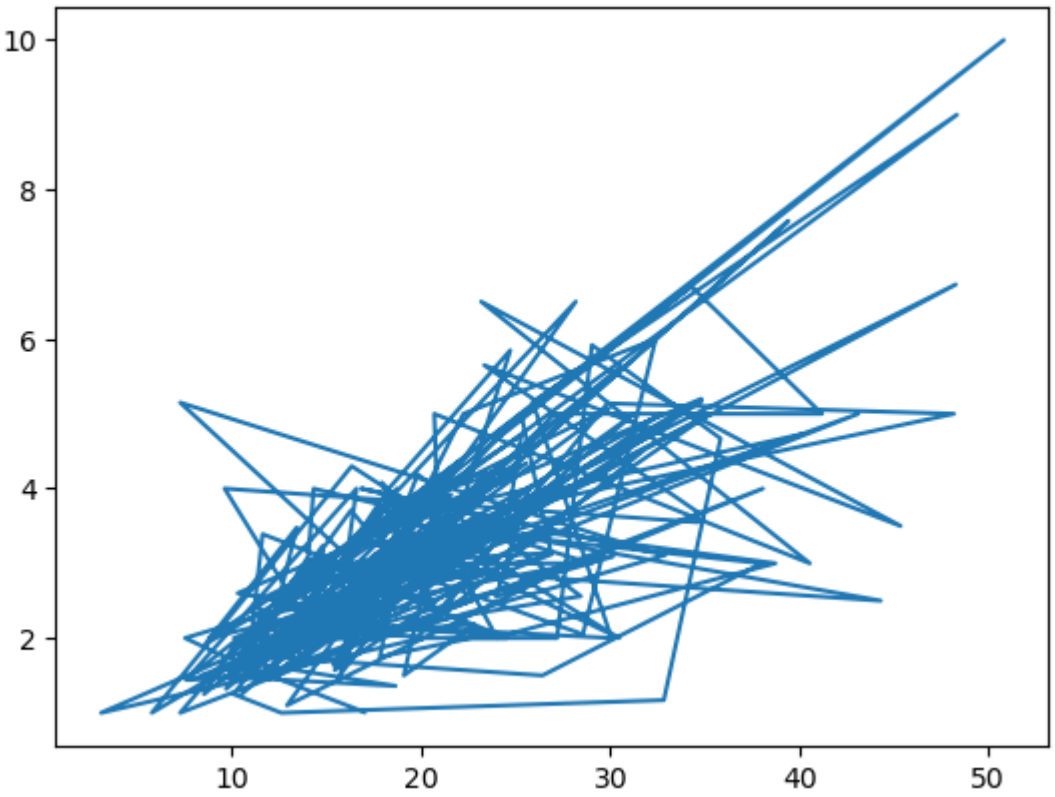
Out[34]:
```
<matplotlib.collections.PathCollection at 0x1ab10783220>
```



scatterplot using plt.plot

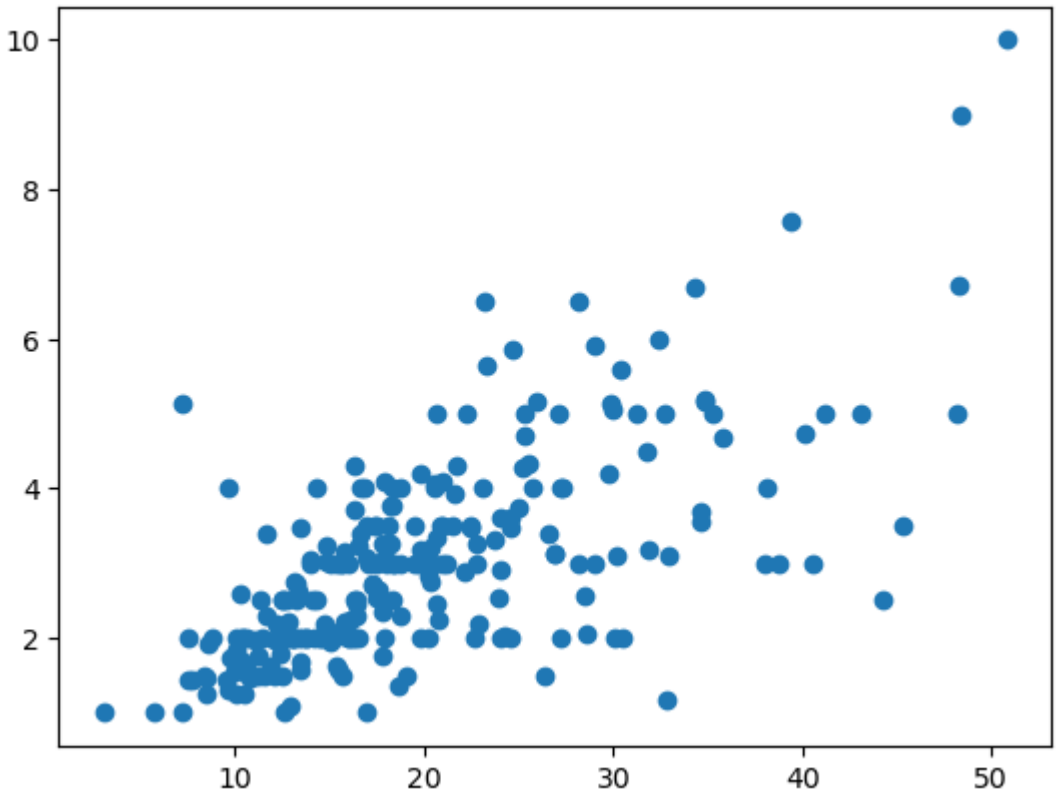
In [38]:
```
plt.plot(tips['total_bill'],tips['tip']) # Faster Technique
```

Out[38]:
```
[<matplotlib.lines.Line2D at 0x1ab11c21100>]
```

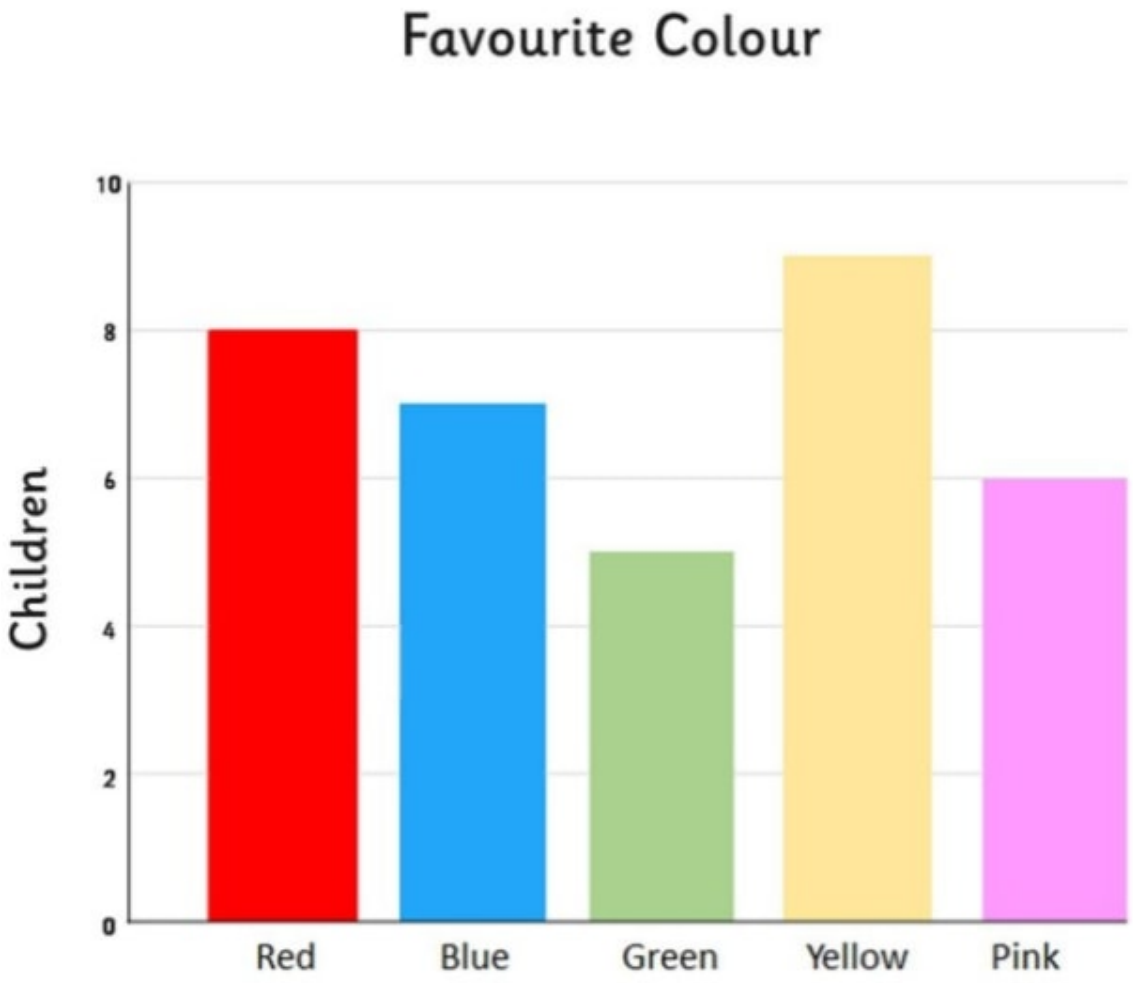


In [37]:
```
plt.plot(tips['total_bill'],tips['tip'],'o')
```

Out[37]:
```
[<matplotlib.lines.Line2D at 0x1ab10aee880>]
```



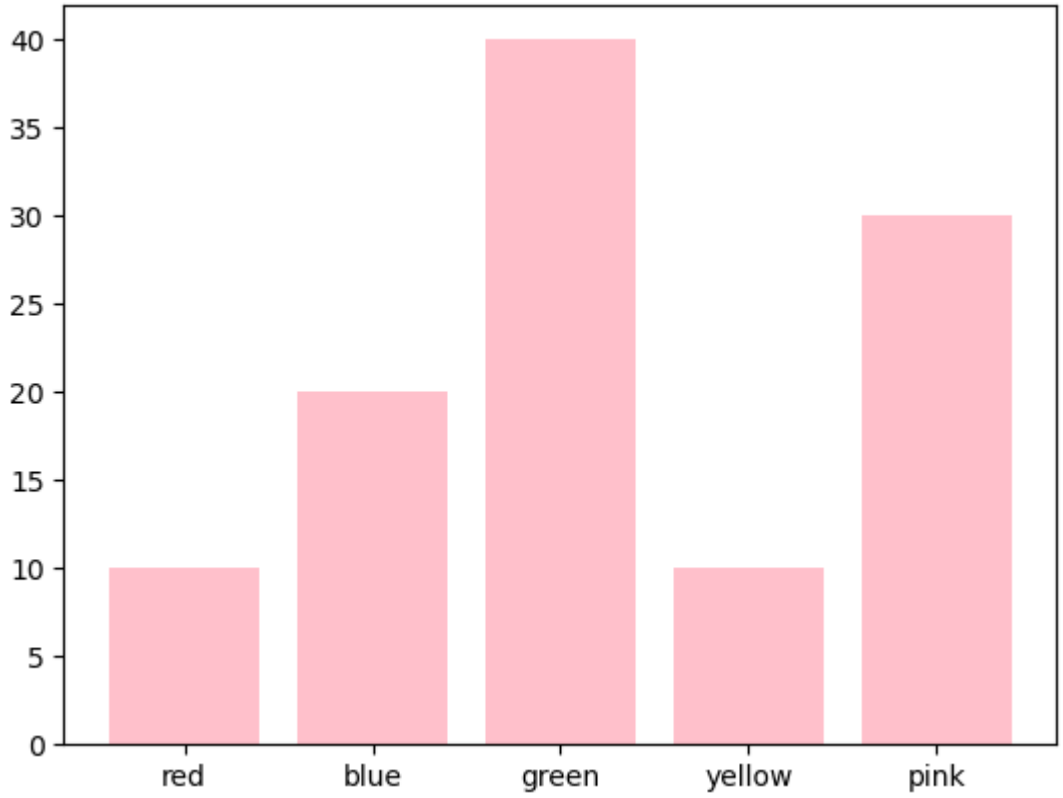
Bar chart



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

In [39]:
```
# simple bar chart
children = [10,20,40,10,30]
colors = ['red','blue','green','yellow','pink']

plt.bar(colors,children,color='pink')
```

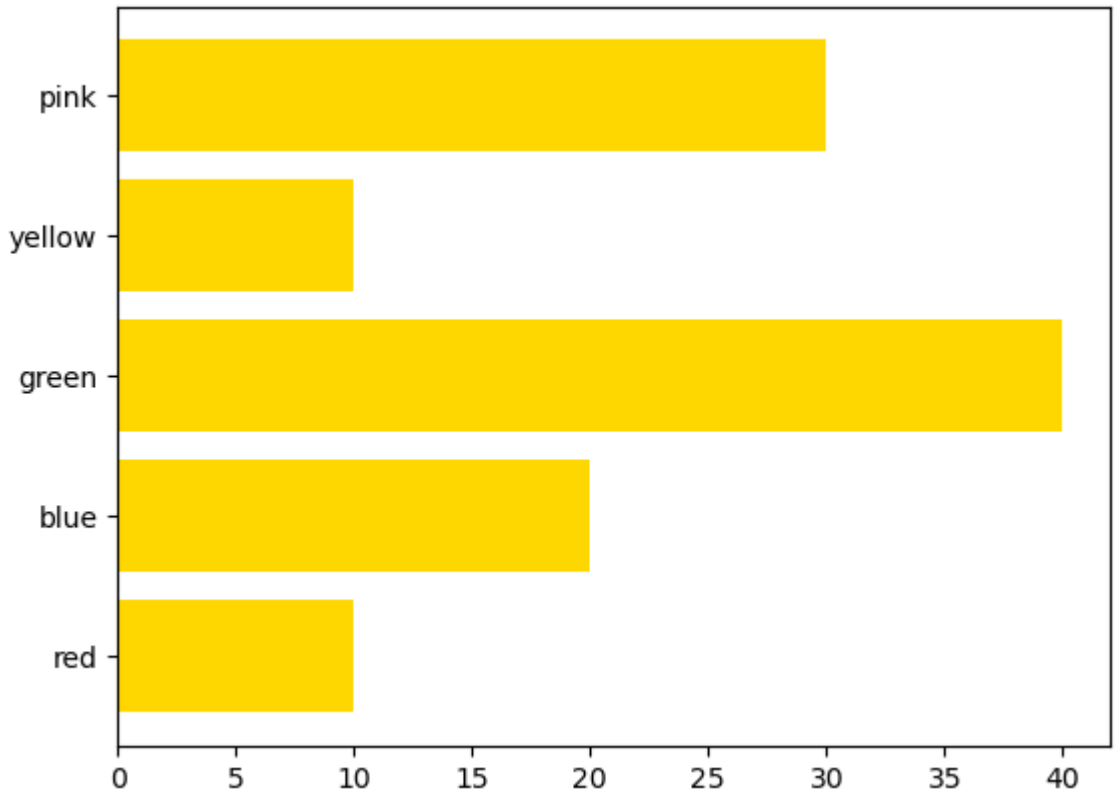
Out[39]: <BarContainer object of 5 artists>



horizontal bar chart

In [41]:
```
plt.barh(colors,children,color='gold')
```

Out[41]: <BarContainer object of 5 artists>



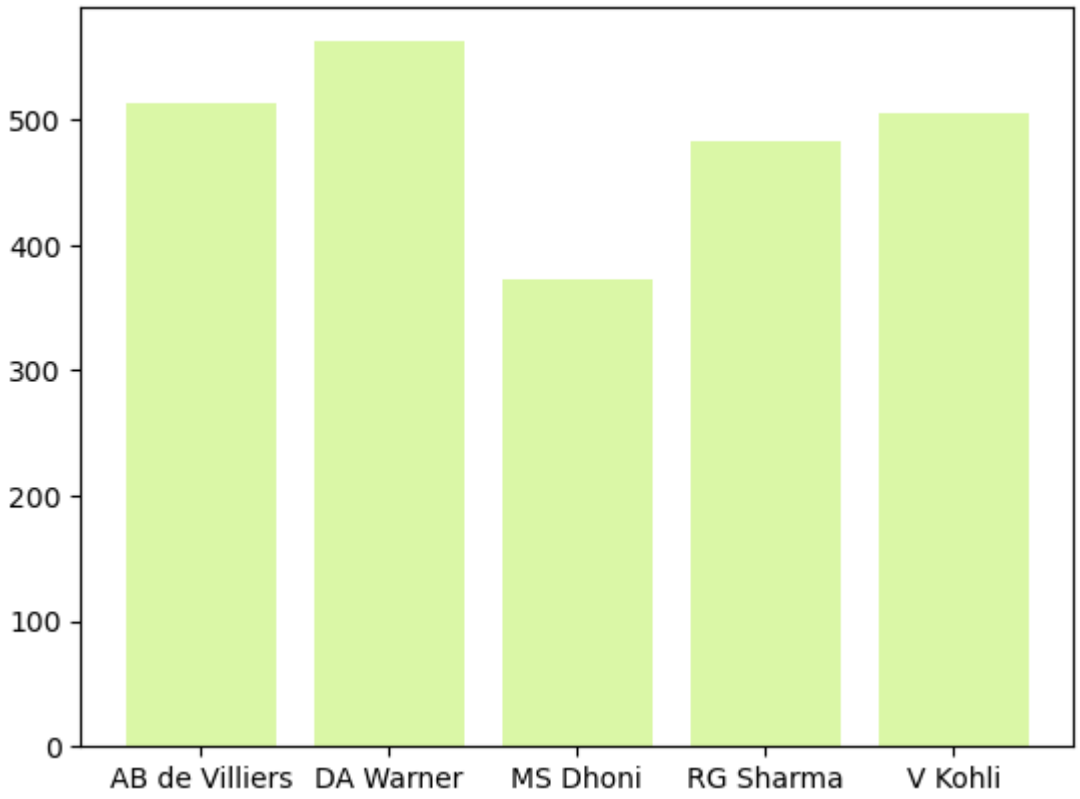
In [45]:
```
#Color and label

df= pd.read_csv("batsman_season_record.csv")

df
```

Out[45]:

	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 [55]:
```
plt.bar(df['batsman'] , df['2015'] , color ='#DAF7A6')
```

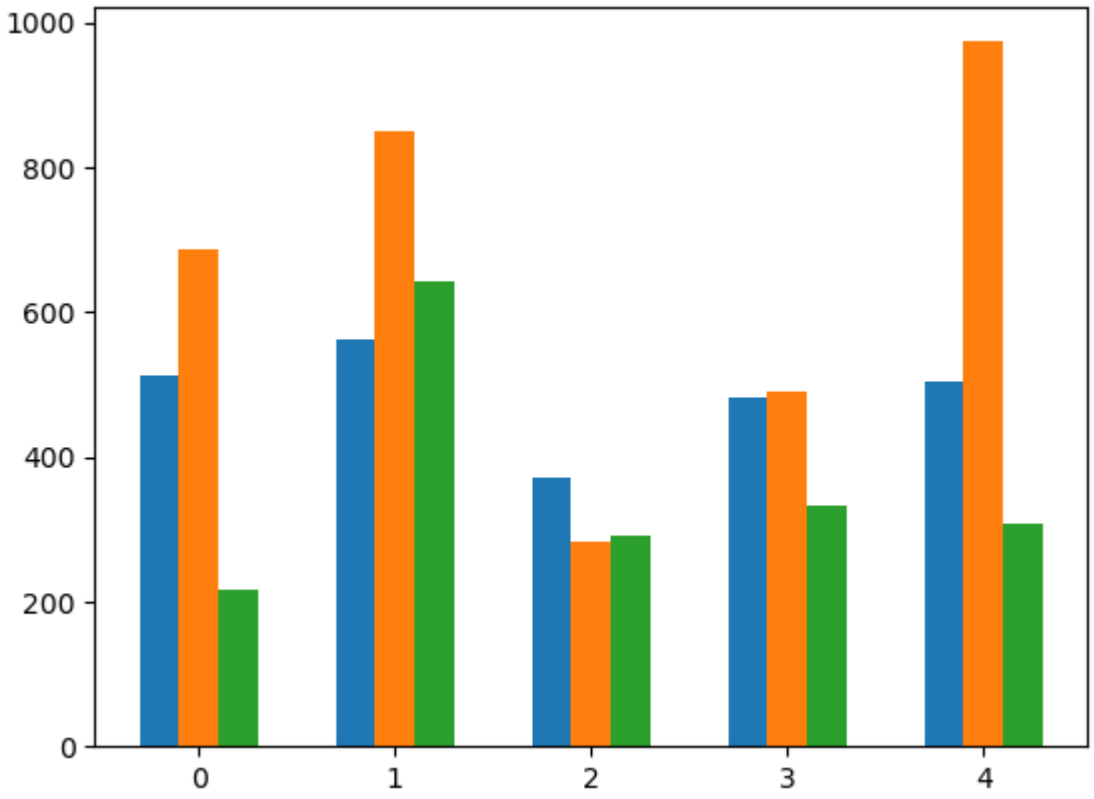
Out[55]: <BarContainer object of 5 artists>



Mutiple bar plots

In [69]:
```
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)
```

Out[69]:
```
<BarContainer object of 5 artists>
```

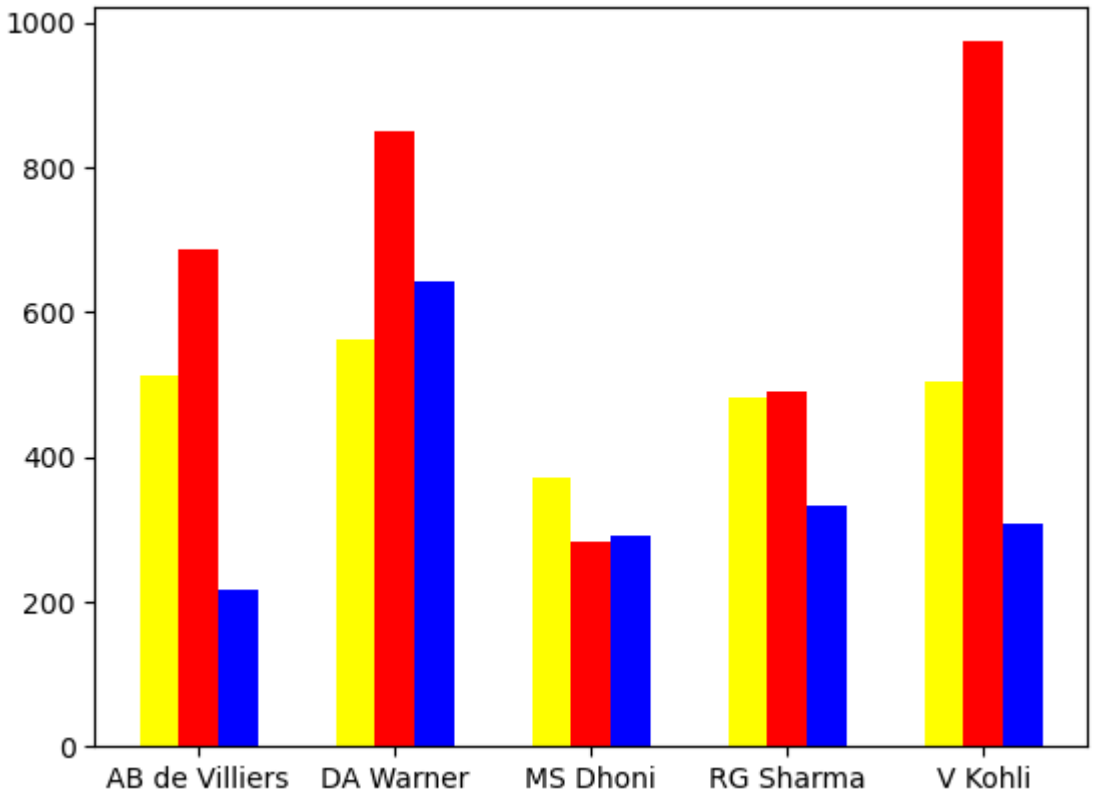


In [68]:
```
df.shape[0]  # batsman
```

Out[68]:
```
5
```

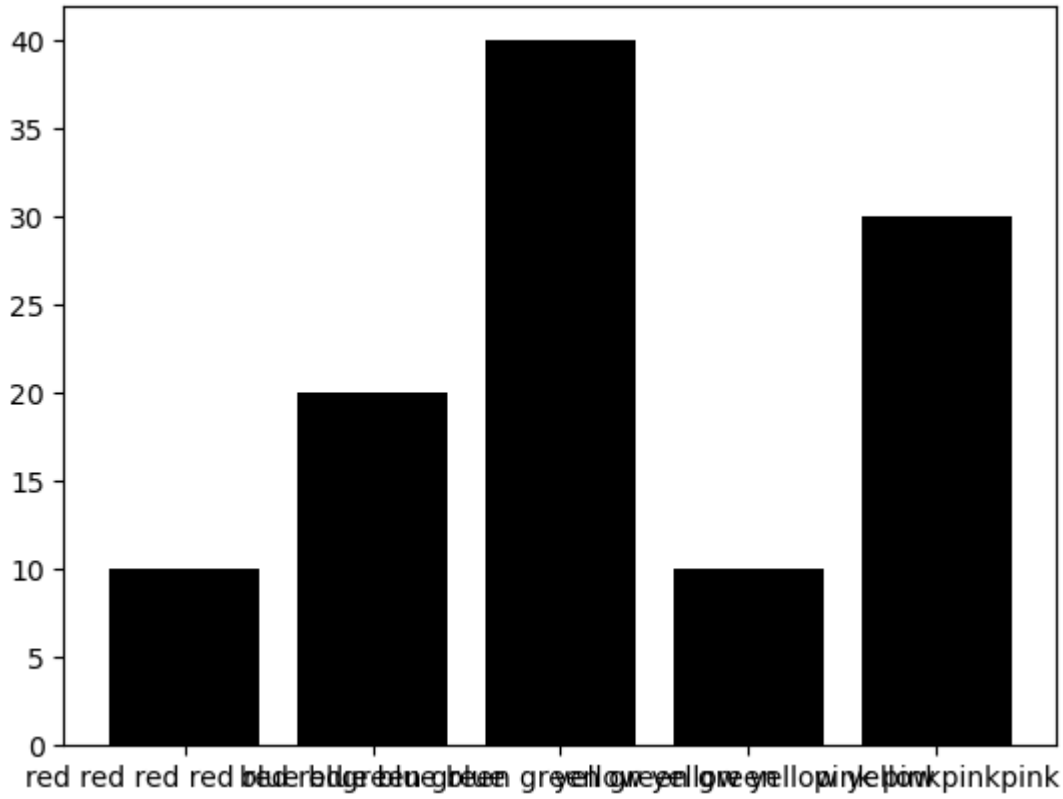
Colors

In [62]:
```
plt.bar(np.arange(df.shape[0] )- 0.2 , df['2015'] , width = 0.2 ,color ='yellc
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()
```

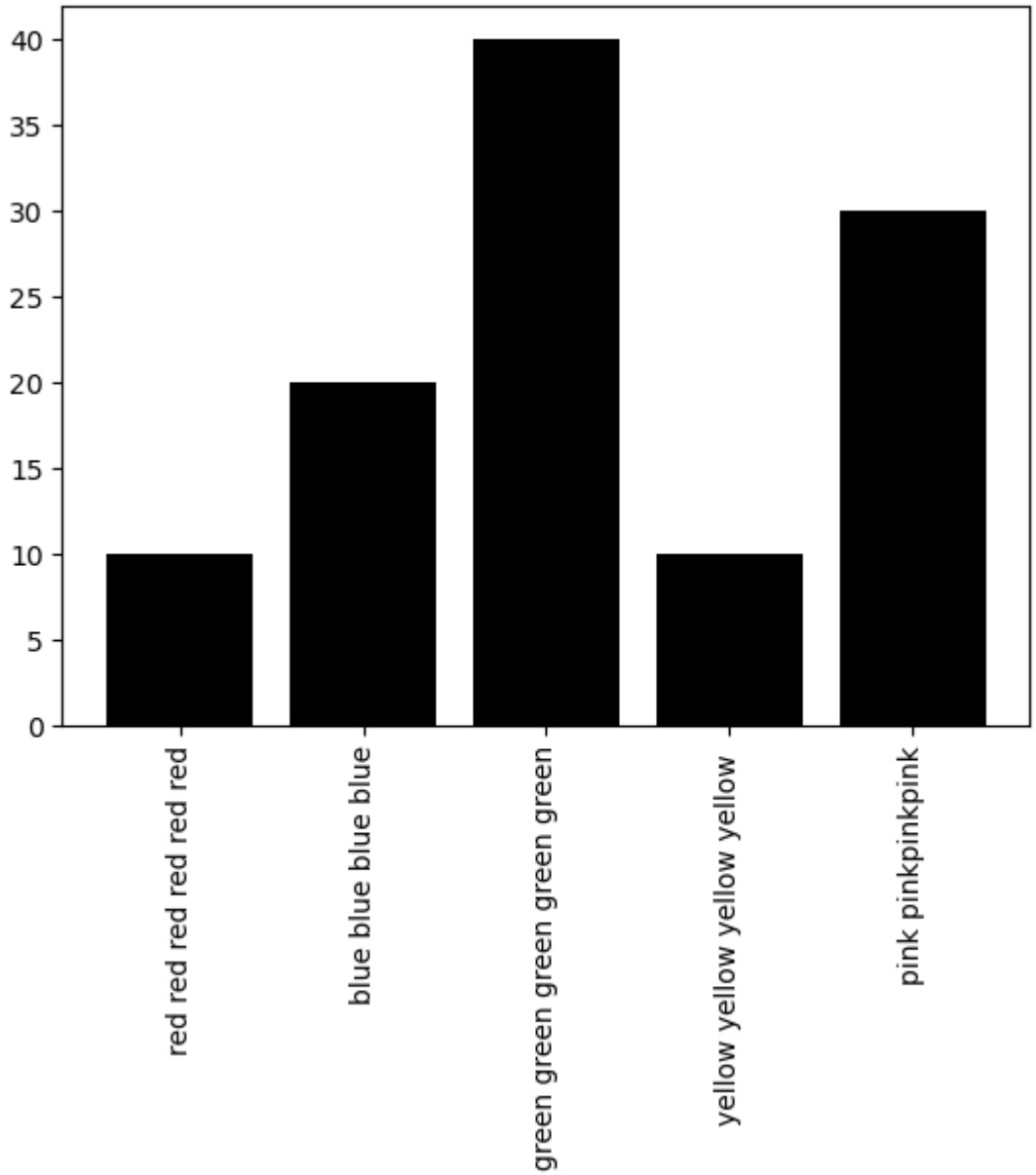


Overlapping problem

In [72]:
```
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')


plt.show()
```

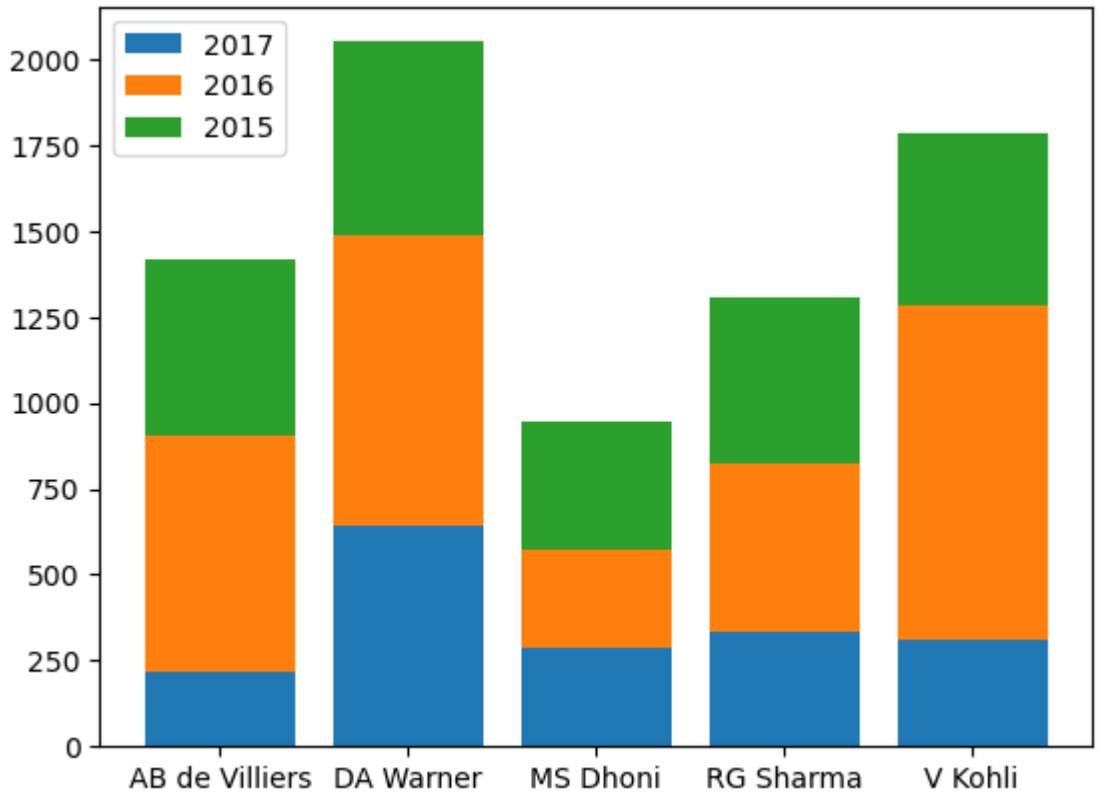


In [73]:
```
### Solution

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')

plt.xticks(rotation='vertical') # Use vertical

plt.show()
```

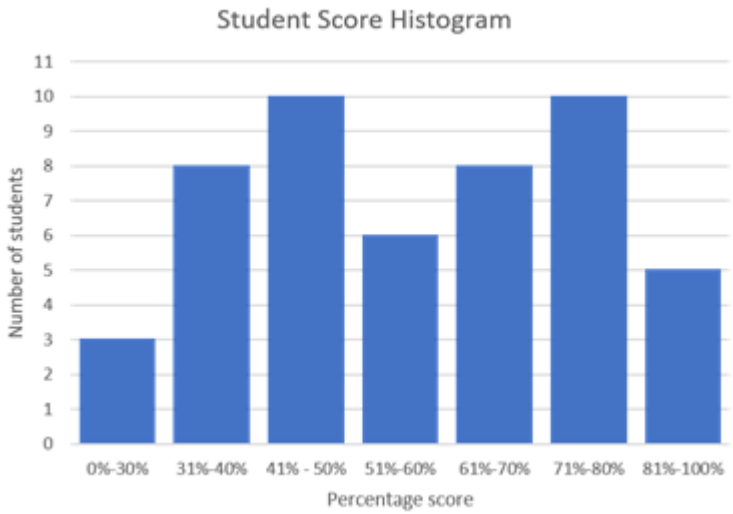


Stacked bar chart

In [75]:
```
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()
```



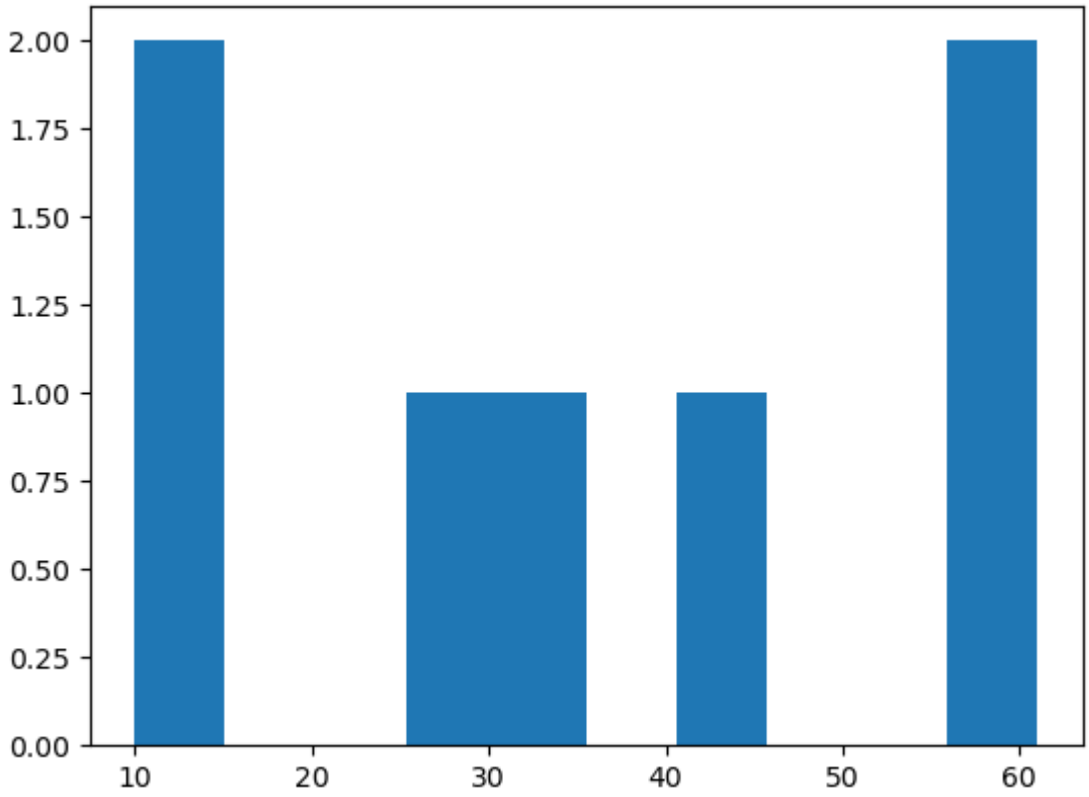
Histogram



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

In [77]:
```
# simple data

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

plt.hist(data)
```

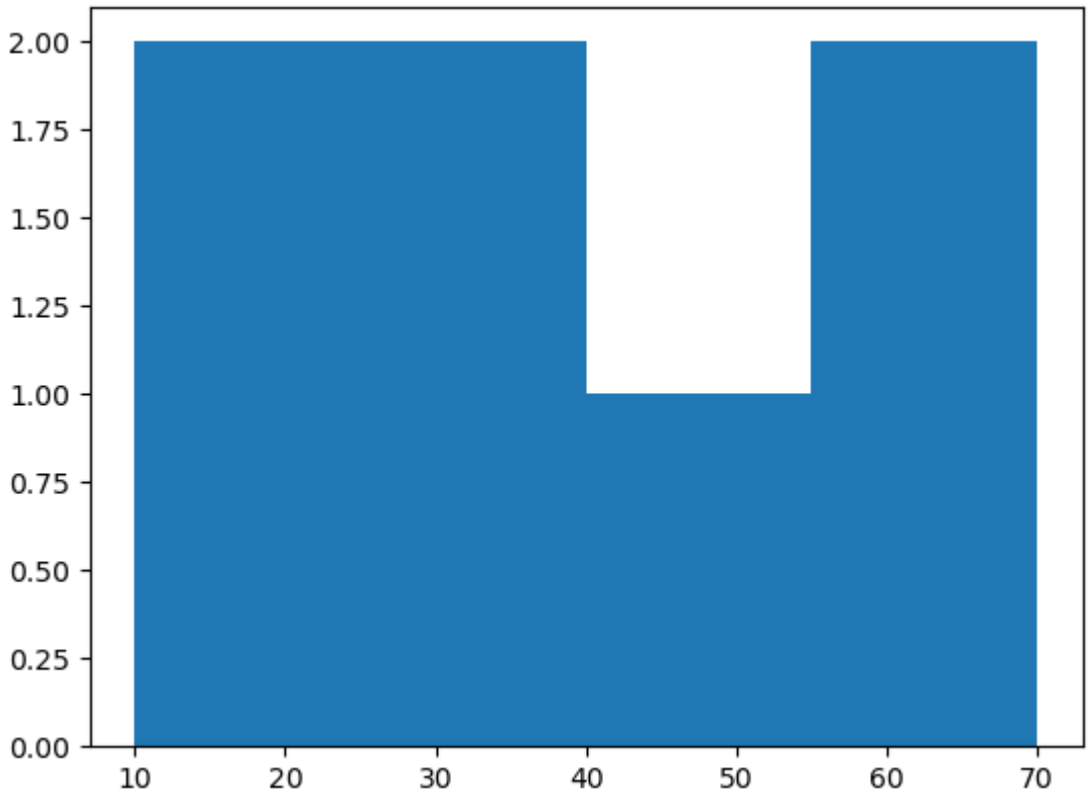
Out[77]:
```
(array([2., 0., 0., 1., 1., 0., 1., 0., 0., 2.]),
 array([10. , 15.1, 20.2, 25.3, 30.4, 35.5, 40.6, 45.7, 50.8, 55.9, 61. ]),
 <BarContainer object of 10 artists>)
```



bins

In [78]:
```
data = [32,45,56,10,15,27,61]

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

Out[78]:
```
(array([2., 2., 1., 2.]),
 array([10, 25, 40, 55, 70]),
 <BarContainer object of 4 artists>)
```



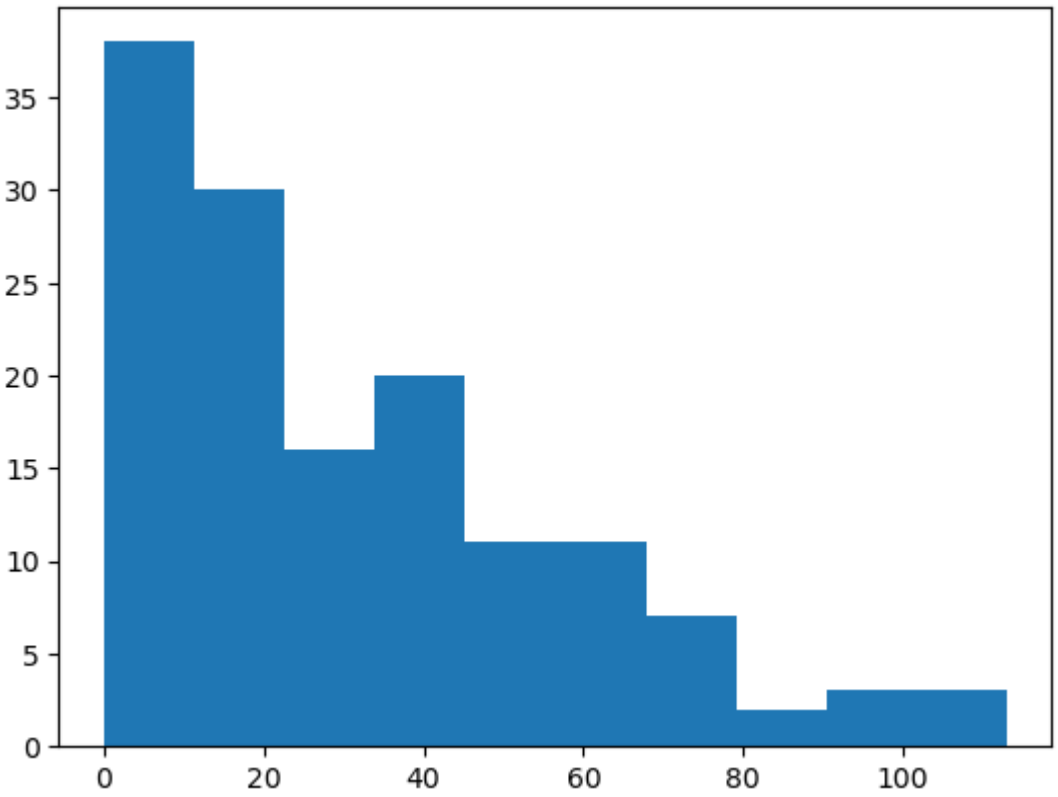
In [80]:
```
# on Data

df = pd.read_csv("vk.csv")

df
```

Out[80]:

	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 [85]:
```
plt.hist(df['batsman_runs'])
plt.show()
```

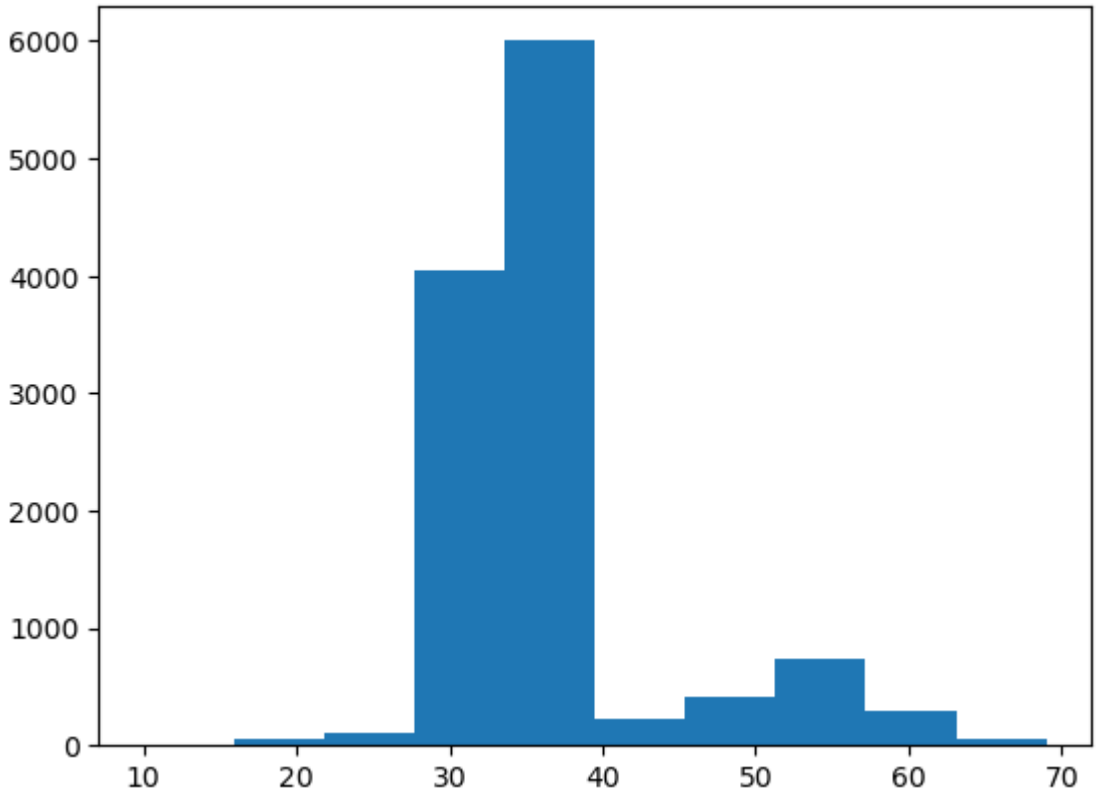


Logarithmic scale

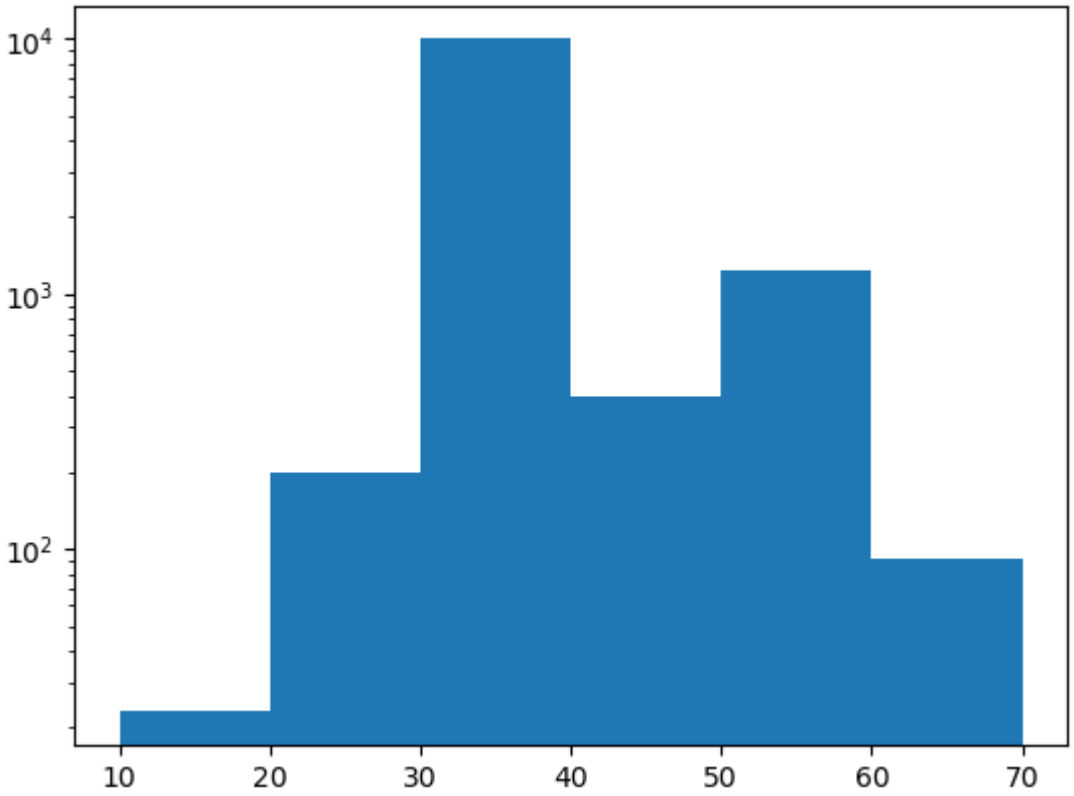
In [91]:
```
arr = np.load("big-array.npy")
```

In [92]:
```
plt.hist(arr) # problem
```

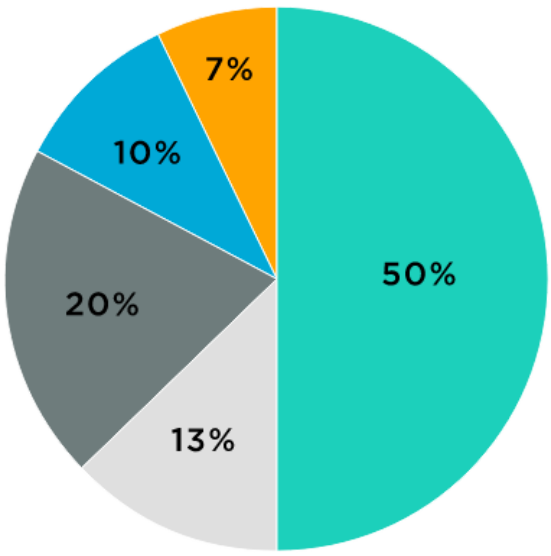
Out[92]:
```
(array([  12.,   60.,  109., 4039., 6003.,  230.,  410.,  744.,  291.,
         51.]),
 array([10. , 15.9, 21.8, 27.7, 33.6, 39.5, 45.4, 51.3, 57.2, 63.1, 69. ]),
 <BarContainer object of 10 artists>)
```



In [93]:
```
# Solution

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

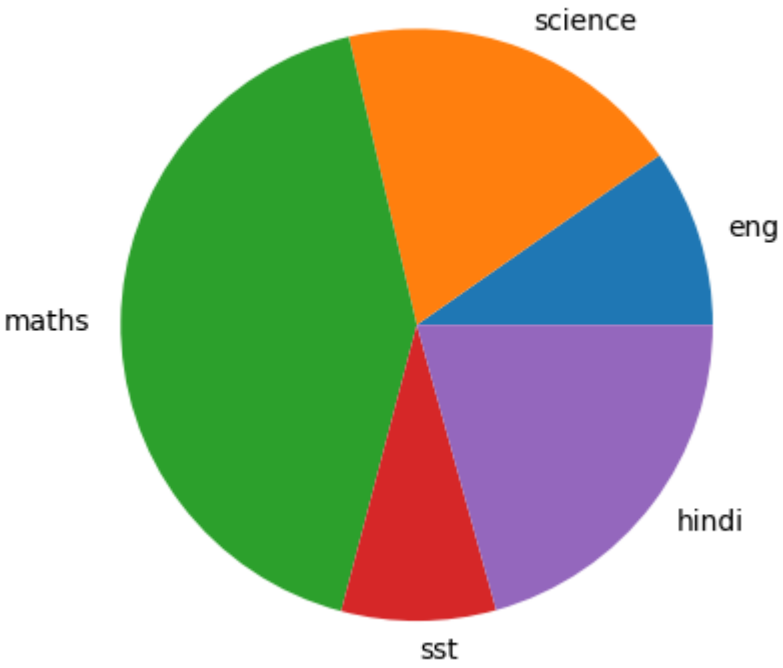


Pie Chart



- Univariate/Bivariate Analysis
- Categorical vs numerical

In [94]:
```
# simple data

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

plt.show()
```

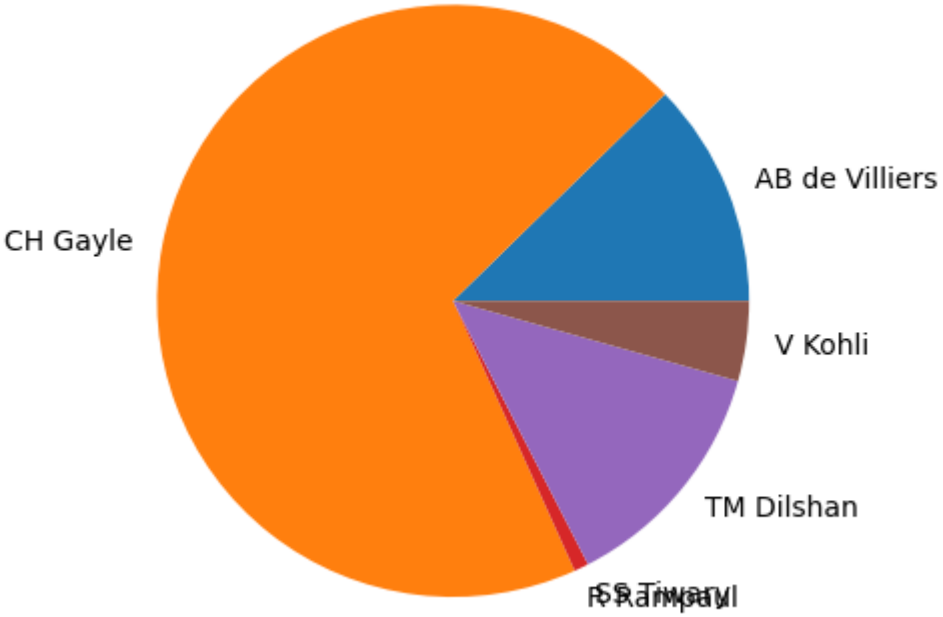


In [95]:
```
# On data

df =pd.read_csv("gayle-175.csv")

df
```

Out[95]:

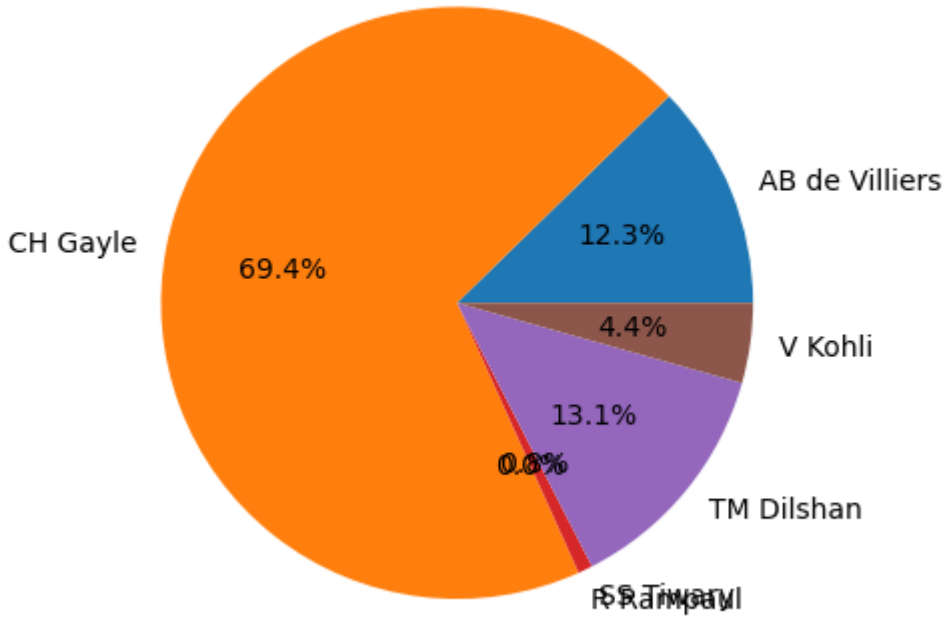
	batsman	batsman_runs
0	AB de Villiers	31
1	CH Gayle	175
2	R Rampaul	0
3	SS Tiwary	2
4	TM Dilshan	33
5	V Kohli	11

In [97]:
```
plt.pie(df['batsman_runs'],labels=df['batsman'])

plt.show()
```



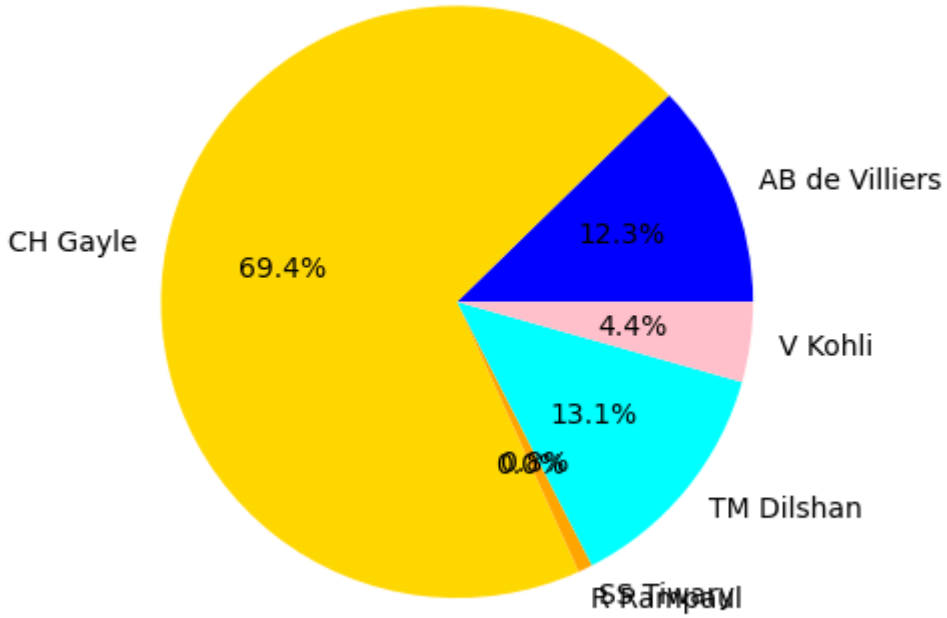
Percentages

In [98]:
```
#autopct = auto percentage

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



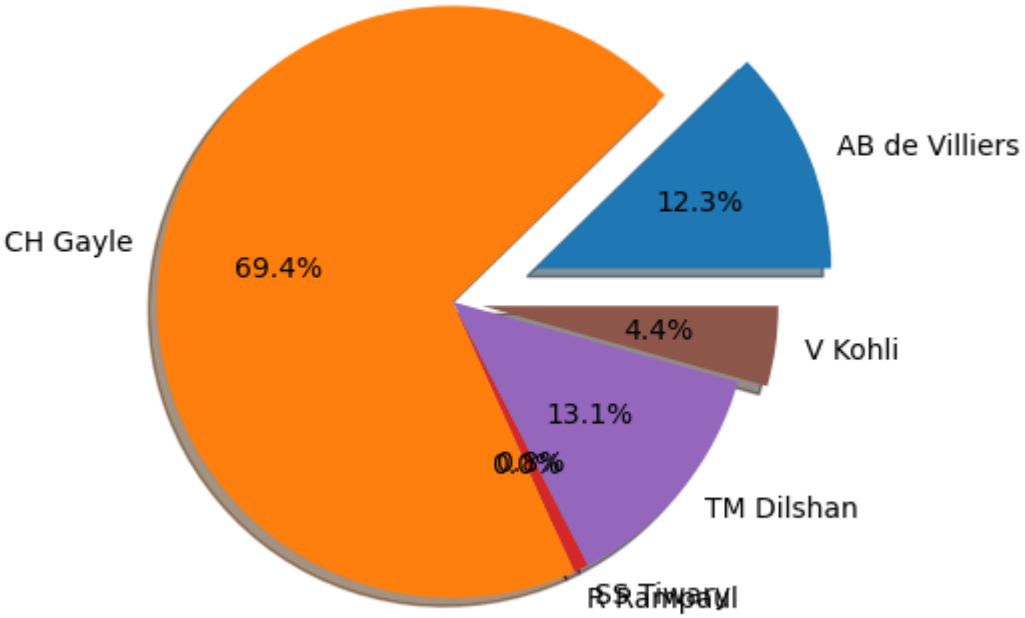
Colours

In [101]:
```
plt.pie(df['batsman_runs'],labels=df['batsman'],
        autopct='%0.1f%%',
        colors=['blue','gold','green','orange','cyan','pink'])


plt.show()
```



Explode shadow

In [107]:
```
plt.pie(df['batsman_runs'],labels=df['batsman'],
        autopct='%0.1f%%',
        explode=[0.3,0,0,0,0,0.1],shadow=True)


plt.show()
```



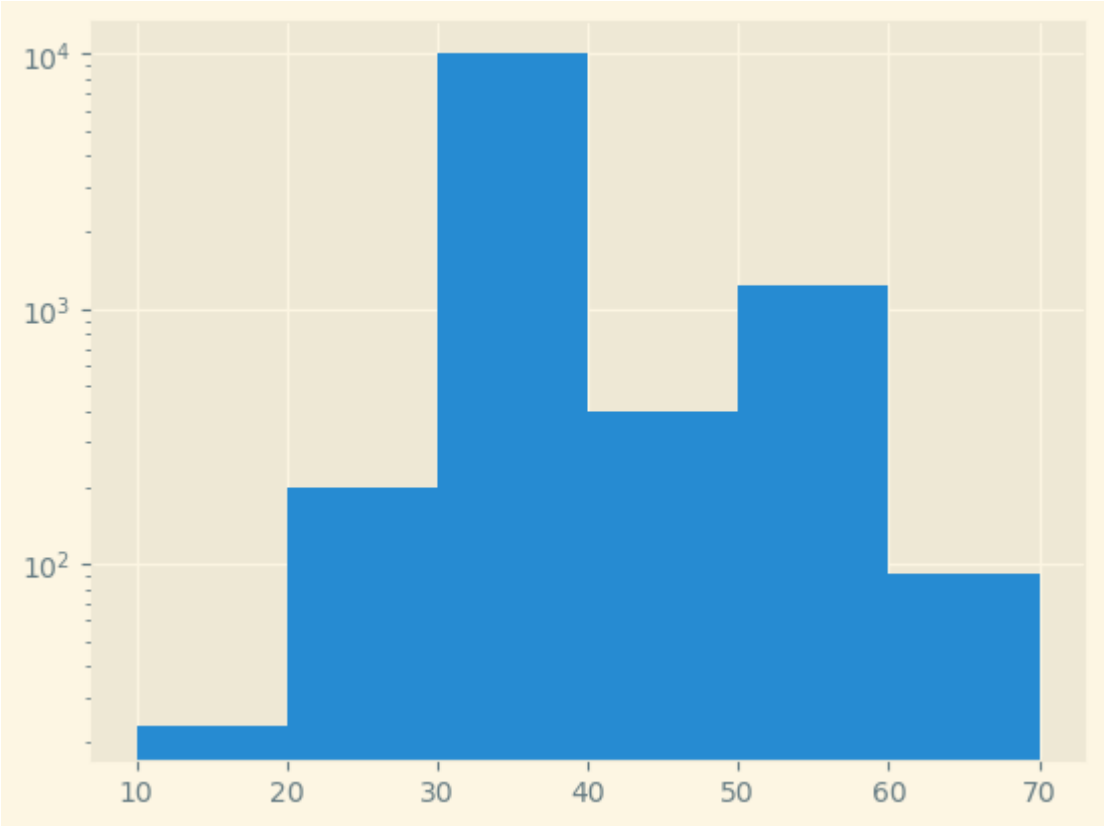
Changing Styles

In [108]:
```
plt.style.available
```

Out[108]:
```
['Solarize_Light2',
 '_classic_test_patch',
 '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']
```

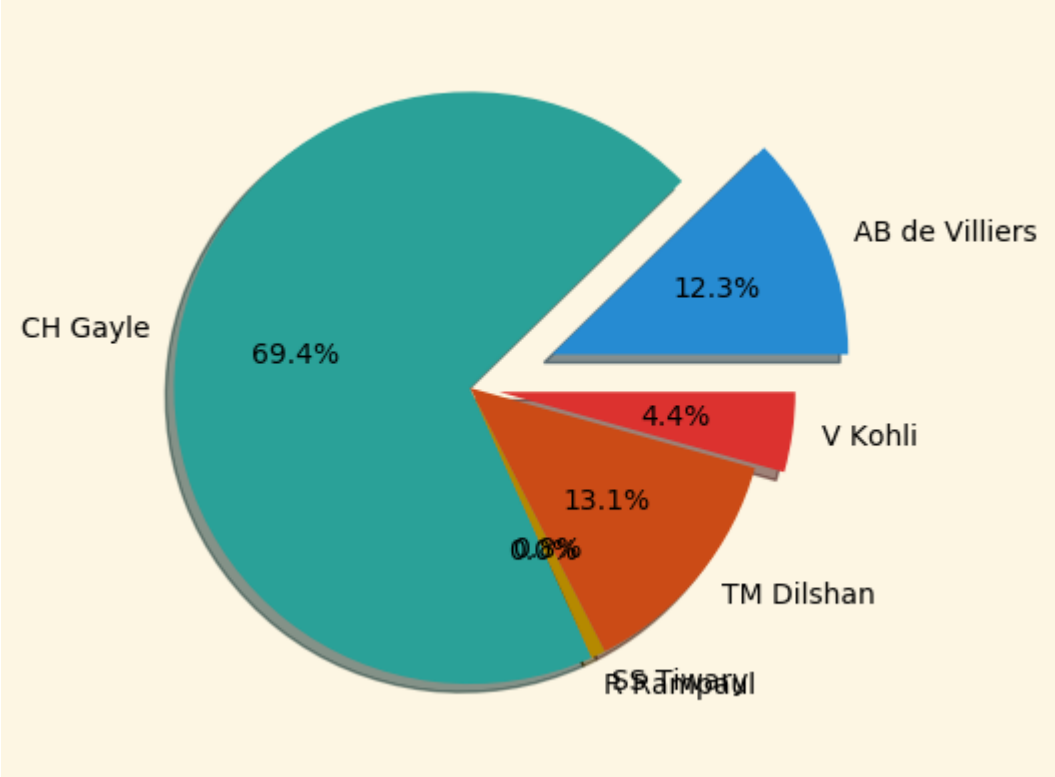
In [109]:
```
# style

plt.style.use('Solarize_Light2')
```

In [110]:
```
# Example

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



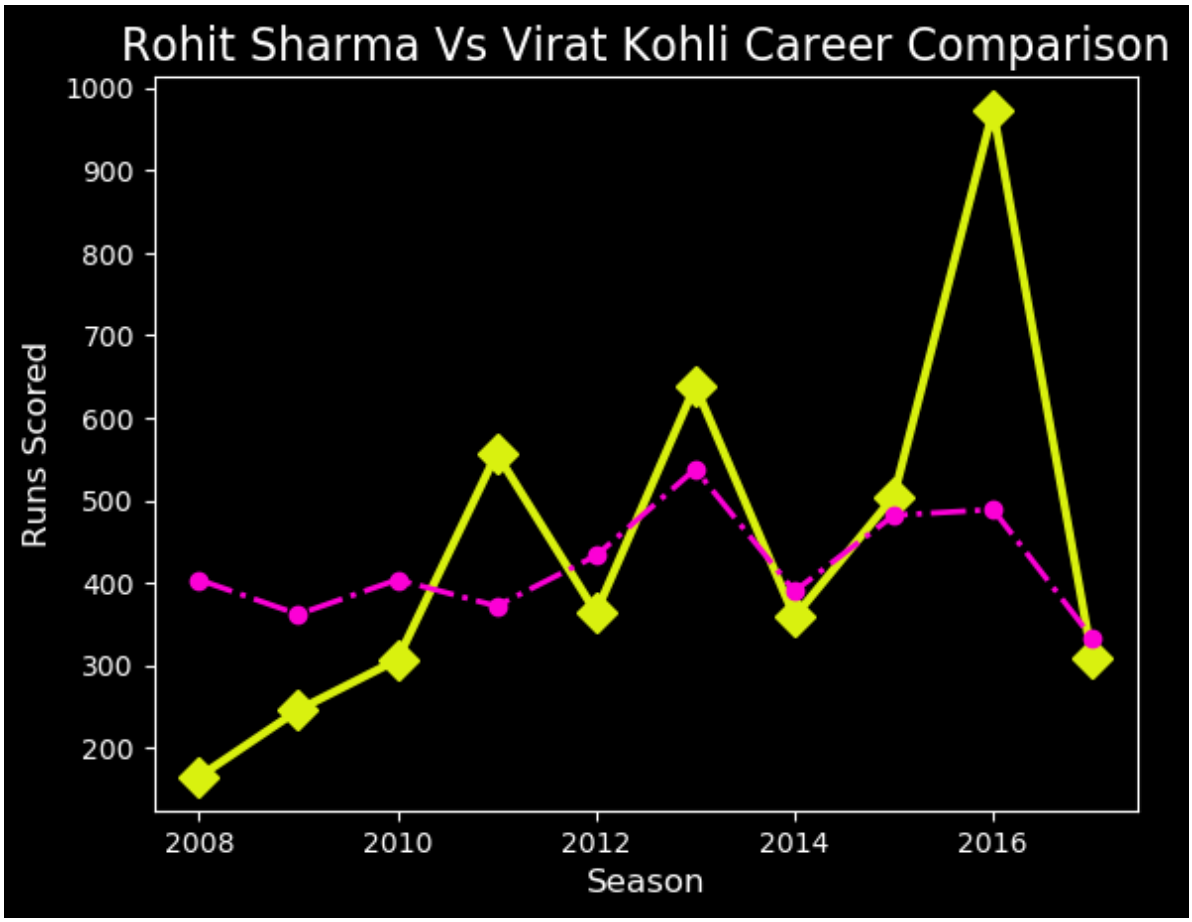
In [111]:
```
# Style 2

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

In [113]:
```
plt.pie(df['batsman_runs'],labels=df['batsman'],
        autopct='%0.1f%%',
        explode=[0.3,0,0,0,0,0.1],shadow=True)


plt.show()
```



In [114]:
```
# Style 3

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

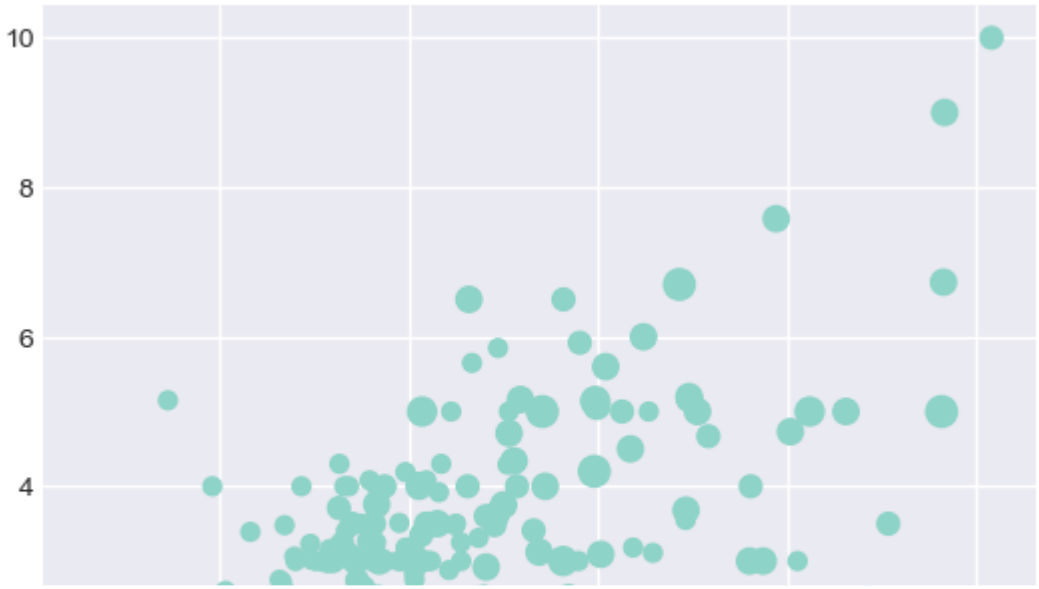
In [115]:
```
# Example

plt.plot(batsman['index'],batsman['V Kohli'],color='#D9F10F',linestyle='solid'
plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6',linestyle='dash

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

plt.grid()

plt.show()
```



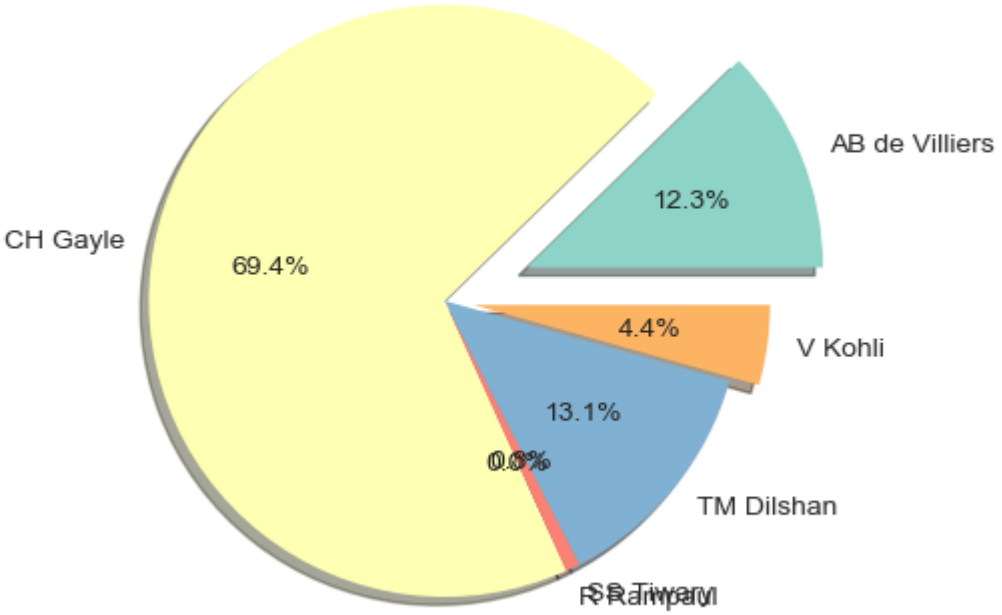
In [116]:
```
# Style 4

plt.style.use('seaborn-darkgrid')
```

In [117]:
```
plt.scatter(tips['total_bill'], tips['tip'] , s = tips ['size']*20)
```

Out[117]:
```
<matplotlib.collections.PathCollection at 0x1ab19271b80>
```



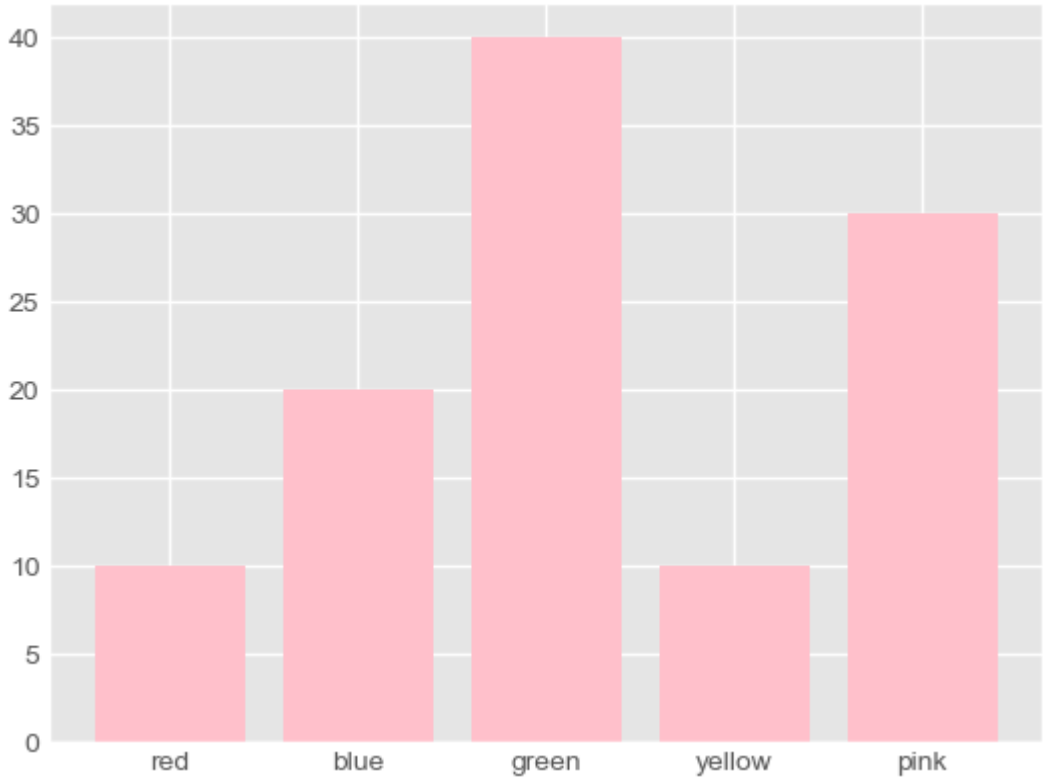
In [120]:
```
plt.pie(df['batsman_runs'],labels=df['batsman'],
        autopct='%0.1f%%',
        explode=[0.3,0,0,0,0,0.1],shadow=True)


plt.show()
```



In [121]:
```
# Style 5

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

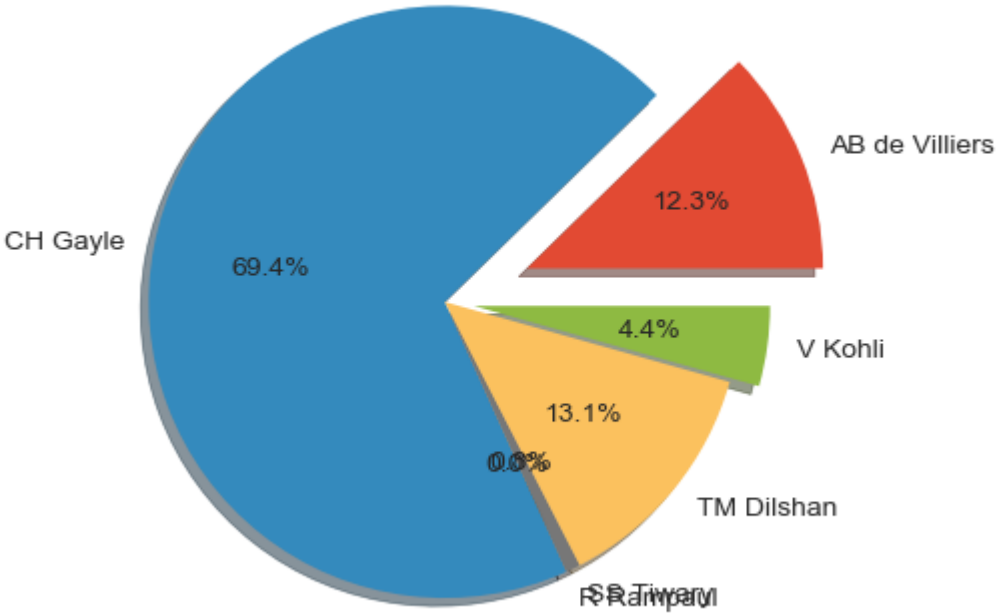
In [122]:
```
# simple bar chart
children = [10,20,40,10,30]
colors = ['red','blue','green','yellow','pink']

plt.bar(colors,children,color='pink')
```

Out[122]:
```
<BarContainer object of 5 artists>
```



Save Figure

In [124]:
```
# Dont apply plt.show() , when using the save

plt.pie(df['batsman_runs'],labels=df['batsman'],
        autopct='%0.1f%%',
        explode=[0.3,0,0,0,0,0.1],shadow=True)

plt.savefig('sample.png') # For saving
```



In [ ]:
```

```