

EDA PROJECT ON IPL DATASET

Q1. What is the maximum number of matches played by an individual player in a season?

Q2. What is the highest average of a player in a season?

Q3. What is the average number of matches played by a player in a season?

Q4. Please check the correlation between all the features.

Q5. Check the list of players who has an average greater than 50 as well strike rate above 120.

Q6. Please check the list of players who has an average greater than 40 and balls faced above 100.

Q7. Please check the list of players who scored more than 300 runs with an average greater than 45 & strike rate above 130.

Q8. Players who scored atleast one century in this season.

Q9. Players who scored atleast 4 half centuries in this season.

Q10. Check the list of players who hit more than 45 boundaries and more than 10 sixes in this season.

Q11. Plot a histogram of number of matches played in a season by players.

Q12. Plot the histogram of balls faced by players.

Q13. Top 30 players with most runs in a season v/s matches played.

Q14. Bar chart for top 20 players with best strike rate in a season.

Q15. Top 30 players with highest number of sixes.

Q16. Scatter plot of runs scored by a player v/s balls faced in a season.

Q17. KDE plot for averages on a season.

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
In [2]: df=pd.read_csv('Ipl_Dataset.csv')
```

```
In [3]: df.head()
```

	POS	Player	Mat	Inns	NO	Runs	HS	Avg	BF	SR	100	50	4s	6s
0	1	KL Rahul	14	14	2	670	132*	55.83	518	129.34	1	5	58	23
1	2	Shikhar Dhawan	17	17	3	618	106*	44.14	427	144.73	2	4	67	12
2	3	David Warner	16	16	2	548	85*	39.14	407	134.64	0	4	52	14
3	4	Shreyas Iyer	17	17	2	519	88*	34.60	421	123.27	0	3	40	16
4	5	Ishan Kishan	14	13	4	516	99	57.33	354	145.76	0	4	36	30

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 133 entries, 0 to 132
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
--- 
 0   POS        133 non-null   int64  
 1   Player     133 non-null   object  
 2   Mat        133 non-null   int64  
 3   Inns       133 non-null   int64  
 4   NO         133 non-null   int64  
 5   Runs       133 non-null   int64  
 6   HS         133 non-null   object  
 7   Avg         133 non-null   float64 
 8   BF         133 non-null   int64  
 9   SR         133 non-null   float64 
 10  100        133 non-null   int64  
 11  50         133 non-null   int64  
 12  4s         133 non-null   int64  
 13  6s         133 non-null   int64  
dtypes: float64(2), int64(10), object(2)
memory usage: 14.7+ KB
```

```
In [5]: df.describe()
```

	POS	Mat	Inns	NO	Runs	Avg	BF
count	133.000000	133.000000	133.000000	133.000000	133.000000	133.000000	133.000000
mean	67.000000	9.631579	6.631579	1.616541	139.157895	19.366241	105.714286
std	38.53786	4.893523	5.030925	1.550766	167.293103	18.053343	122.253870
min	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000	1.000000
25%	34.000000	5.000000	2.000000	0.000000	10.000000	6.000000	12.000000
50%	67.000000	10.000000	5.000000	1.000000	59.000000	15.000000	53.000000
75%	100.000000	14.000000	11.000000	2.000000	232.000000	29.900000	169.000000
max	133.000000	17.000000	17.000000	7.000000	670.000000	101.000000	191.420000

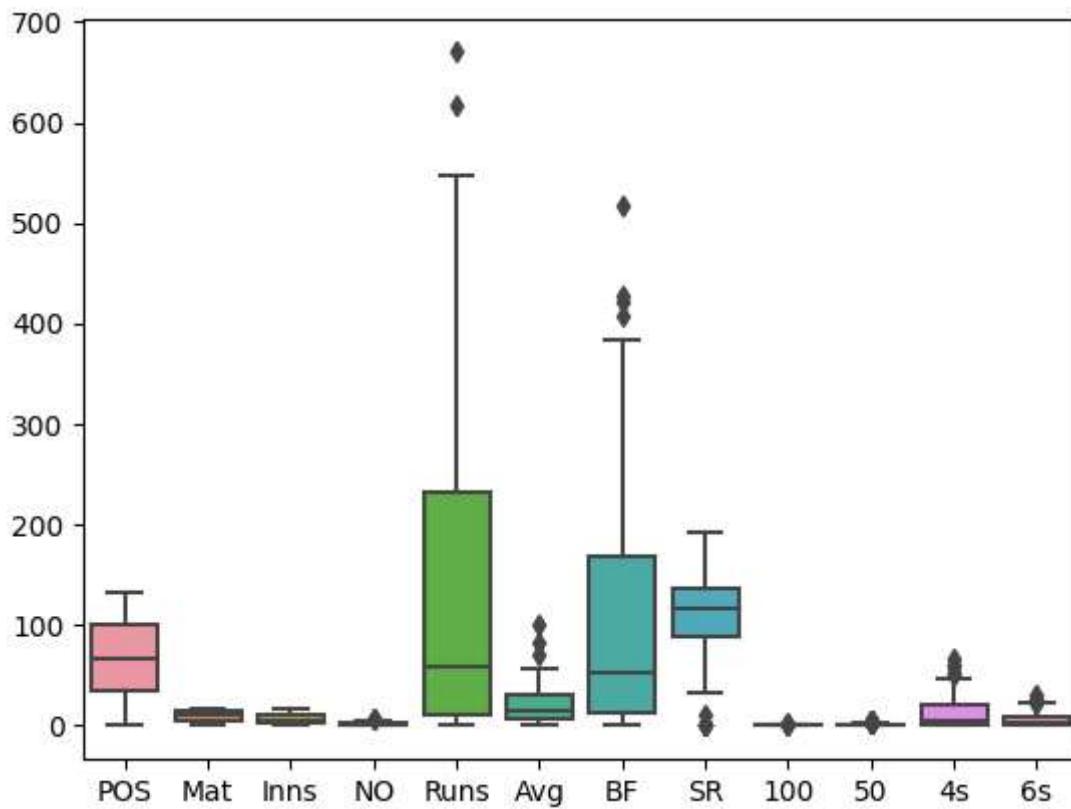


```
In [6]: df.isnull().sum()
```

```
Out[6]: POS      0  
Player    0  
Mat       0  
Inns      0  
NO        0  
Runs      0  
HS        0  
Avg       0  
BF        0  
SR        0  
100       0  
50        0  
4s        0  
6s        0  
dtype: int64
```

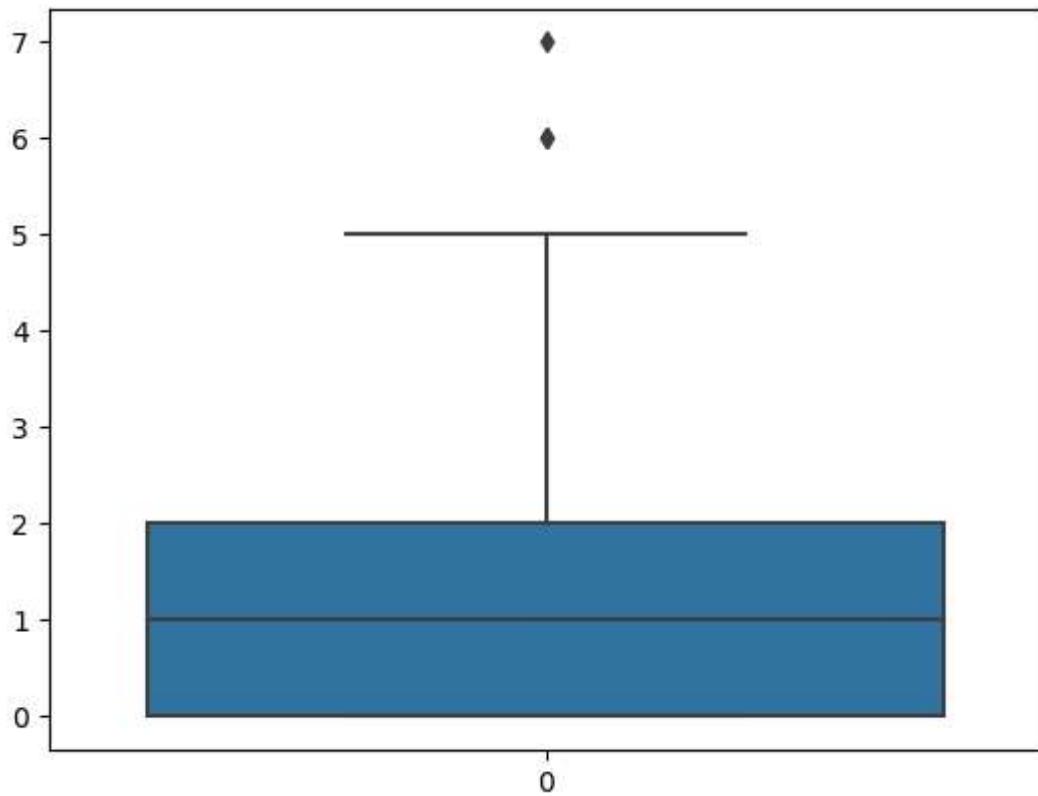
```
In [7]: sns.boxplot(df)
```

```
Out[7]: <AxesSubplot: >
```



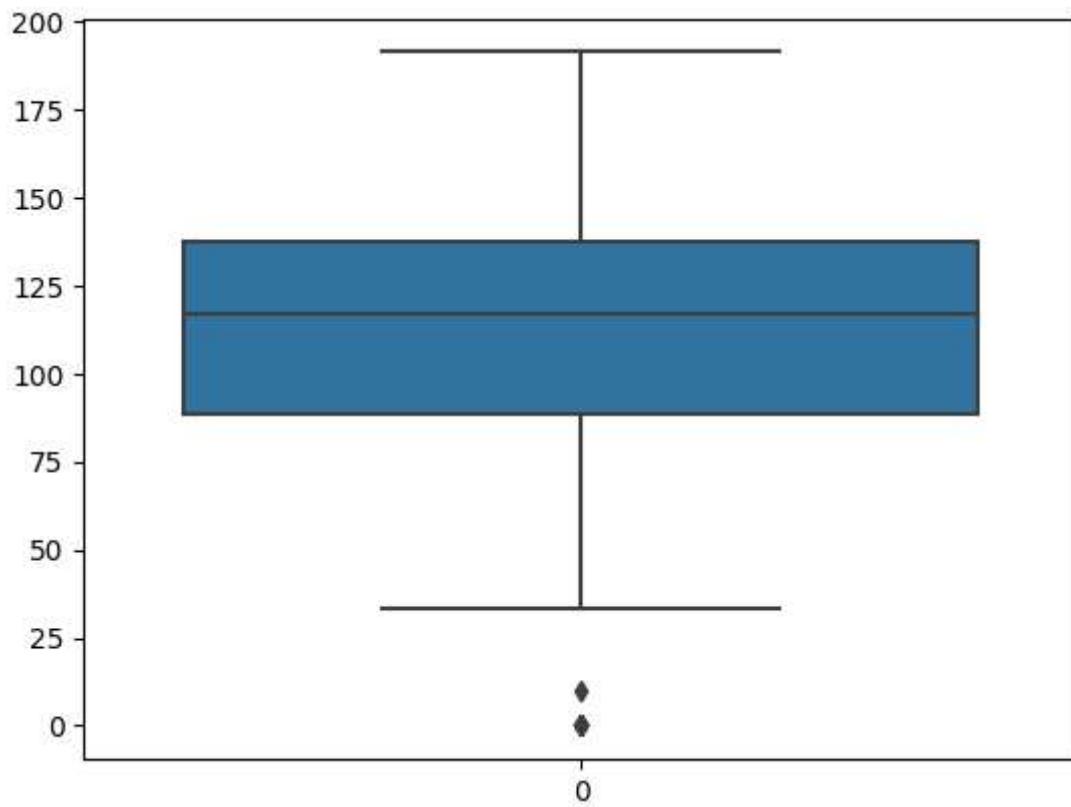
```
In [8]: sns.boxplot(df['NO'])
```

```
Out[8]: <AxesSubplot: >
```



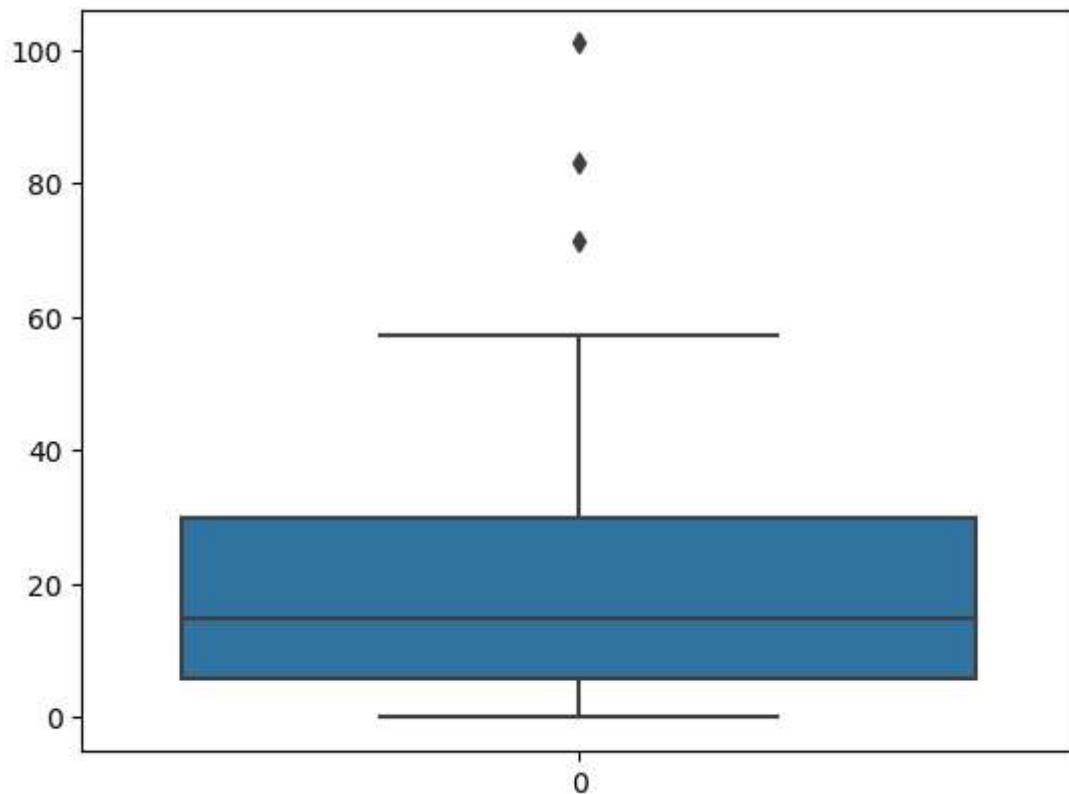
```
In [9]: sns.boxplot(df['SR'])
```

```
Out[9]: <AxesSubplot: >
```



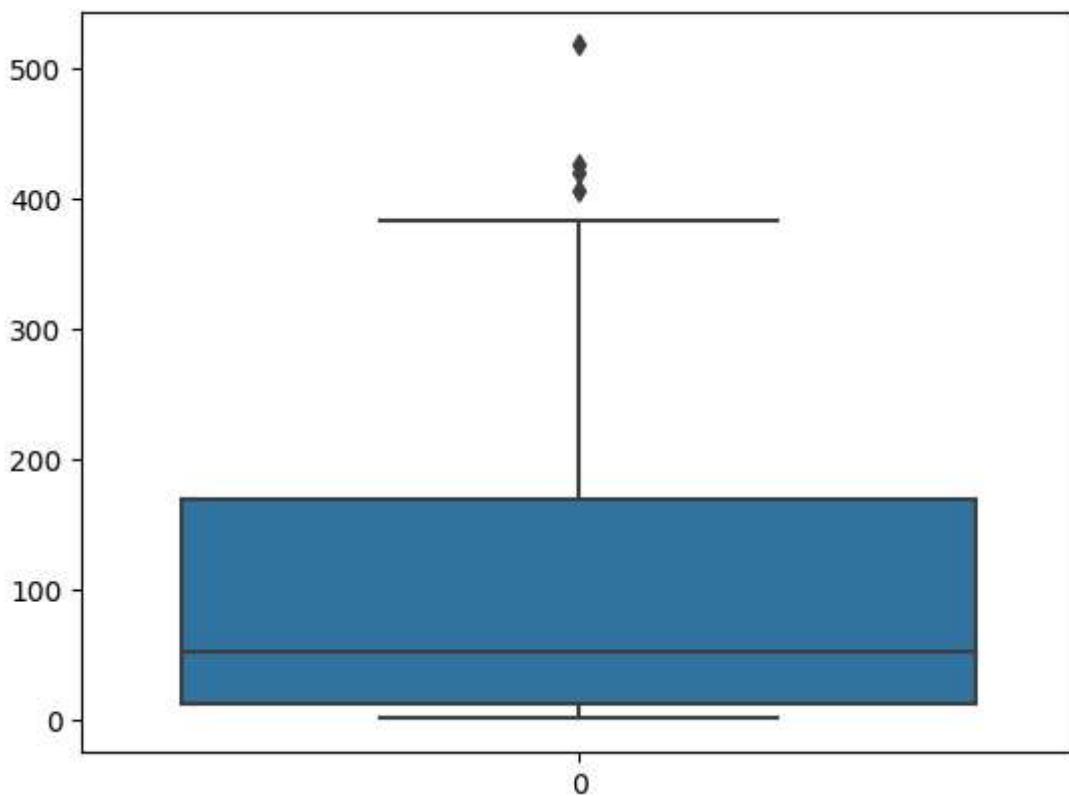
```
In [10]: sns.boxplot(df['Avg'])
```

```
Out[10]: <AxesSubplot: >
```



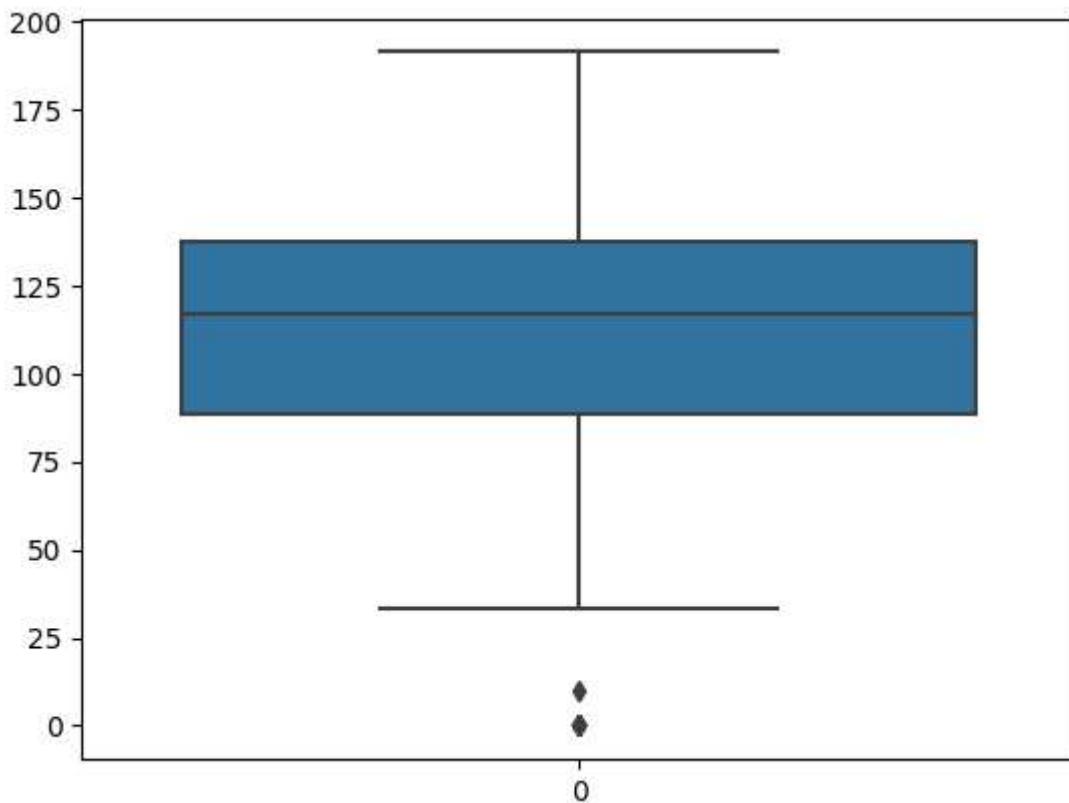
```
In [11]: sns.boxplot(df['BF'])
```

```
Out[11]: <AxesSubplot: >
```



```
In [12]: sns.boxplot(df['SR'])
```

```
Out[12]: <AxesSubplot: >
```



What is the maximum number of matches played by an individual player in a season?

```
In [13]: df['Mat'].max()
```

```
Out[13]: 17
```

What is the highest average of a player in a season?

```
In [14]: df['Avg'].max()
```

```
Out[14]: 101.0
```

What is the average number of matches played by a player in a season?

```
In [15]: df['Mat'].mean()
```

```
Out[15]: 9.631578947368421
```

Please check the correlation between all the features.

```
In [16]: df.corr()
```

C:\Users\acer\AppData\Local\Temp\ipykernel_3188\1134722465.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
df.corr()

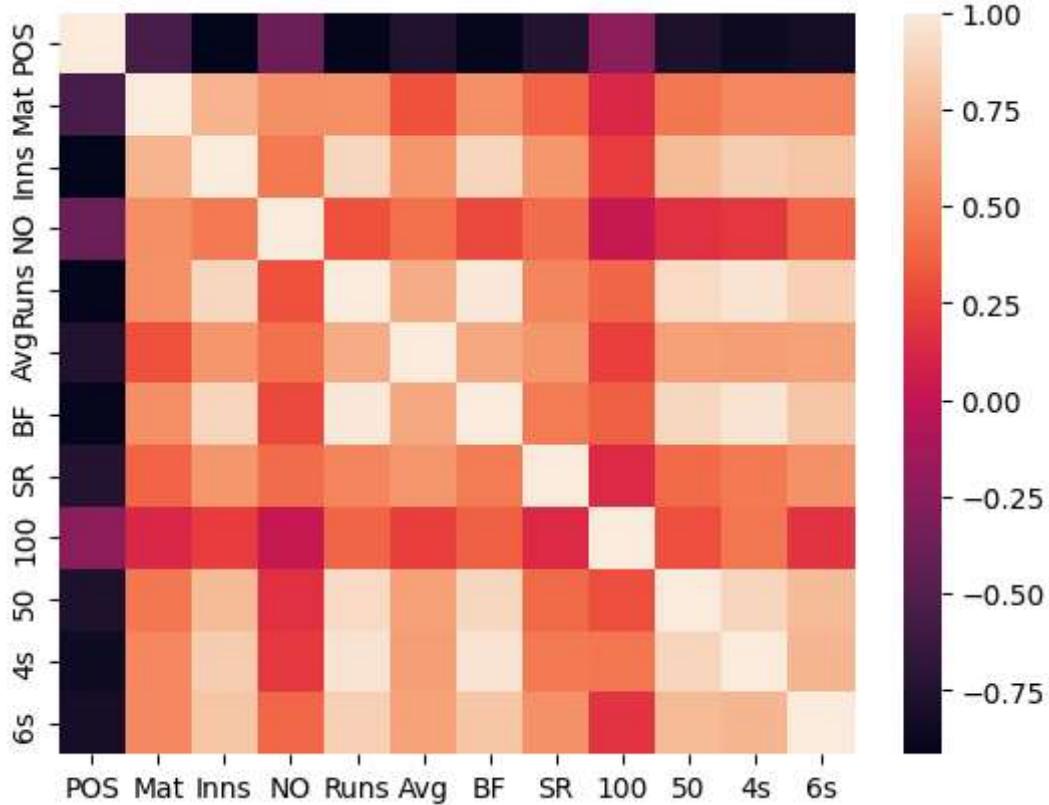
```
Out[16]:
```

	POS	Mat	Inns	NO	Runs	Avg	BF	SR	
POS	1.000000	-0.556695	-0.916174	-0.386247	-0.896664	-0.746510	-0.891969	-0.730489	-0.2481
Mat	-0.556695	1.000000	0.728358	0.555261	0.565237	0.306128	0.555102	0.371328	0.1284
Inns	-0.916174	0.728358	1.000000	0.464356	0.895914	0.585148	0.889753	0.581468	0.2241
NO	-0.386247	0.555261	0.464356	1.000000	0.308747	0.429888	0.271021	0.415060	0.0197
Runs	-0.896664	0.565237	0.895914	0.308747	1.000000	0.687475	0.989948	0.517678	0.3821
Avg	-0.746510	0.306128	0.585148	0.429888	0.687475	1.000000	0.664173	0.591306	0.2331
BF	-0.891969	0.555102	0.889753	0.271021	0.989948	0.664173	1.000000	0.474226	0.3581
SR	-0.730489	0.371328	0.581468	0.415060	0.517678	0.591306	0.474226	1.000000	0.1351
100	-0.248353	0.128405	0.224328	0.019726	0.382577	0.233726	0.358550	0.135331	1.0000
50	-0.769577	0.455437	0.755027	0.174184	0.912264	0.638326	0.900547	0.406029	0.2928
4s	-0.846965	0.522628	0.846749	0.204779	0.959576	0.623643	0.955994	0.465132	0.4560
6s	-0.808756	0.523367	0.810922	0.391472	0.861166	0.646224	0.810916	0.566317	0.1861

```
In [17]: sns.heatmap(df.corr())
```

C:\Users\acer\AppData\Local\Temp\ipykernel_3188\58359773.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
sns.heatmap(df.corr())

```
Out[17]: <AxesSubplot: >
```



Check the list of players who has an average greater than 50 as well strike rate above 120

```
In [18]: df_Avg_SR = df[(df['Avg'] > 50) & (df['SR'] > 120)]
```

```
In [19]: df_Avg_SR['Player']
```

```
Out[19]: 0      KL Rahul
        4      Ishan Kishan
       31      Kieron Pollard
       36      Wriddhiman Saha
       37      Ruturaj Gaikwad
       57      Deepak Hooda
       60      Tom Curran
Name: Player, dtype: object
```

Please check the list of players who has an average greater than 40 and balls faced above 100.

```
In [20]: df_Avg_BF = df[(df['Avg'] > 40) & (df['BF'] > 100)]
df_Avg_BF['Player']
```

```
Out[20]: 0      KL Rahul
         1      Shikhar Dhawan
         4      Ishan Kishan
         8      Virat Kohli
         9      AB de Villiers
        10      Faf du Plessis
        14      Eoin Morgan
        24      Kane Williamson
        27      Chris Gayle
        28      Ben Stokes
        31      Kieron Pollard
        32      Rahul Tewatia
        33      Ravindra Jadeja
        36      Wriddhiman Saha
        37      Ruturaj Gaikwad
Name: Player, dtype: object
```

Please check the list of players who scored more than 300 runs with an average greater than 45 & strike rate above 130.

```
In [21]: df_Avg_R_A_S = df[(df['Runs'] > 300) & (df['Avg'] > 45) & (df['SR'] > 130)]
df_Avg_R_A_S['Player']
```

```
Out[21]: 4      Ishan Kishan
         9      AB de Villiers
        24      Kane Williamson
Name: Player, dtype: object
```

Players who scored atleast one century in this season.

```
In [22]: dd_100=(df[(df['100']==1)])
```

```
In [23]: dd_100['Player']
```

```
Out[23]: 0      KL Rahul
         13     Mayank Agarwal
         28     Ben Stokes
Name: Player, dtype: object
```

Q9. Players who scored atleast 4 half centuries in this season.

```
In [24]: dd_50=(df[(df['50']==1)])
dd_50['Player']
```

```
Out[24]: 14      Eoin Morgan
16      Ambati Rayudu
21      Rishabh Pant
28      Ben Stokes
29      Hardik Pandya
30      Aaron Finch
31      Kieron Pollard
32      Rahul Tewatia
33      Ravindra Jadeja
34      Rahul Tripathi
40      Sam Curran
42      Dinesh Karthik
43      Pat Cummins
44      Priyam Garg
45      Mandeep Singh
47      Sunil Narine
51      Ajinkya Rahane
57      Deepak Hooda
58      Vijay Shankar
60      Tom Curran
Name: Player, dtype: object
```

Q10. Check the list of players who hit more than 45 boundaries and more than 10 sixes in this season.

```
In [25]: df_Avg_B_S_ = df[(df['4s'] > 45) & (df['6s'] > 10)]
df_Avg_B_S_['Player']
```

```
Out[25]: 0      KL Rahul
1      Shikhar Dhawan
2      David Warner
5      Quinton de Kock
6      Suryakumar Yadav
Name: Player, dtype: object
```

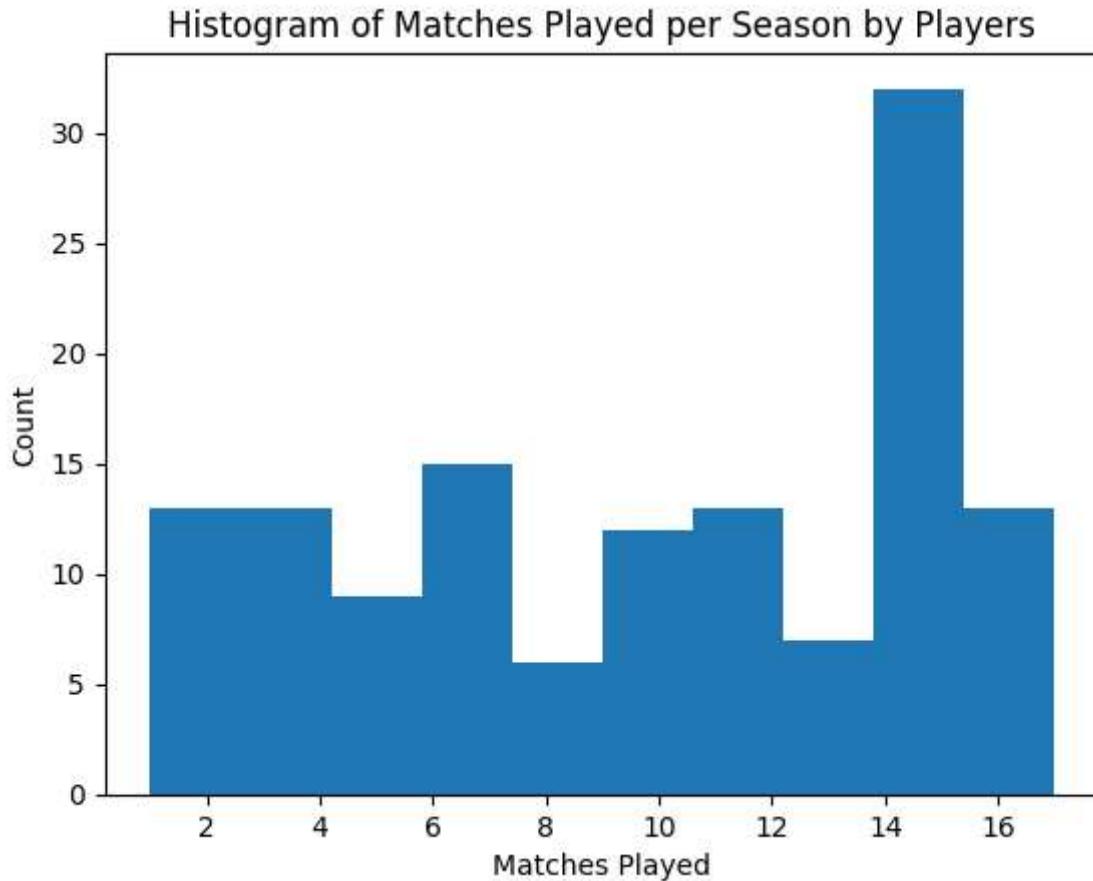
Q11. Plot a histogram of number of matches played in a season by players.

```
In [26]: matches_per_season = df.groupby('Player')['Mat'].sum()

# plot a histogram of matches played per season
plt.hist(matches_per_season, bins=10)

# add Labels and a title
plt.xlabel('Matches Played')
plt.ylabel('Count')
plt.title('Histogram of Matches Played per Season by Players')
```

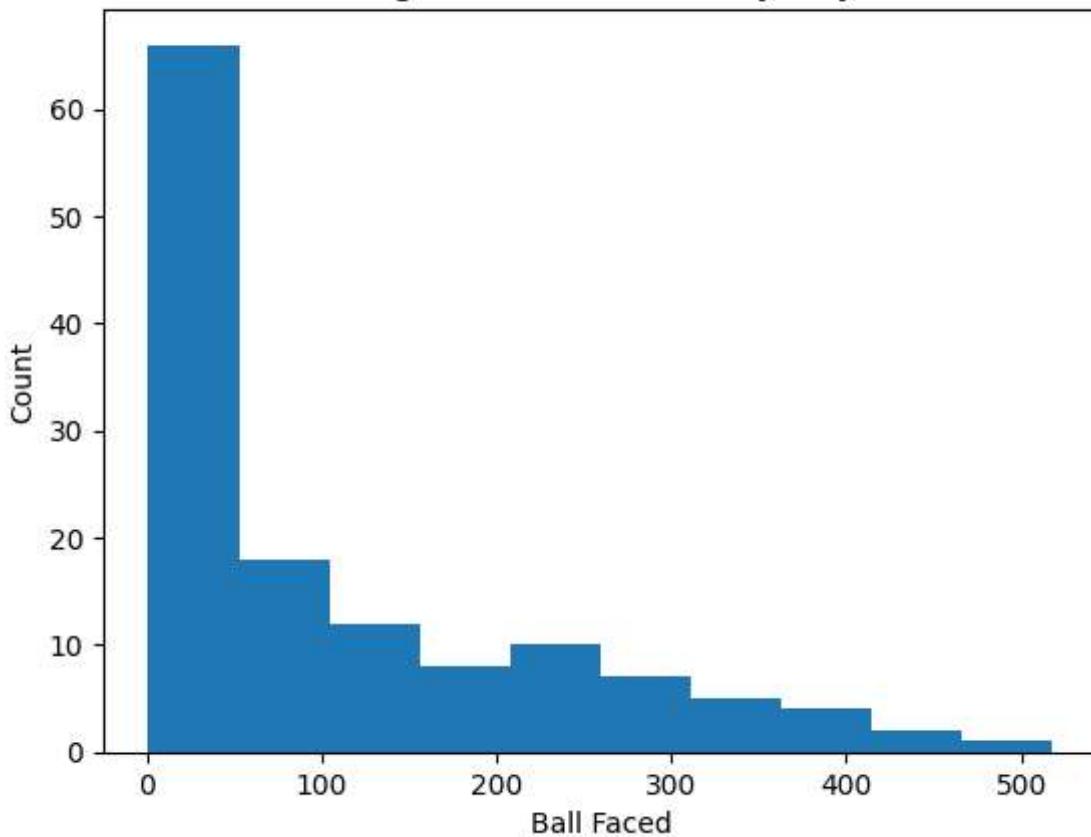
```
# show the plot  
plt.show()
```



Q12. Plot the histogram of balls faced by players.

```
In [27]: plt.hist(df['BF'], bins=10)  
  
# add Labels and a title  
plt.xlabel('Ball Faced')  
plt.ylabel('Count')  
plt.title('Histogram of Ball Faced Buy Playes')  
  
# show the plot  
plt.show()
```

Histogram of Ball Faced Buy Playes



Q13. Top 30 players with most runs in a season v/s matches played.

```
In [42]: Top_MatchPlayed_Playes=df.sort_values(by=['Mat','Runs'],ascending=False)
```

```
In [52]: Top_30_MatchPlayed_Playes=Top_MatchPlayed_Playes.iloc[:31,:]
```

```
In [59]: T_M_P=Top_30_MatchPlayed_Playes['Player']  
T_M_P
```

```
Out[59]: 1          Shikhar Dhawan
         3          Shreyas Iyer
        19         Marcus Stoinis
       67          Kagiso Rabada
         2          David Warner
         5         Quinton de Kock
         6        Suryakumar Yadav
        12         Manish Pandey
        31         Kieron Pollard
       54         Krunal Pandya
       73         Rashid Khan
      106        Anrich Nortje
     114         T Natarajan
        7        Devdutt Padikkal
         8         Virat Kohli
         9        AB de Villiers
        49         Axar Patel
       52        Washington Sundar
       72    Ravichandran Ashwin
      110        Jasprit Bumrah
      116        Rahul Chahar
     123    Yuzvendra Chahal
     132        Trent Boult
         0         KL Rahul
         4         Ishan Kishan
        11         Shubman Gill
        14         Eoin Morgan
        15         Sanju Samson
        17        Nicholas Pooran
        18         Nitish Rana
        21         Rishabh Pant
Name: Player, dtype: object
```

```
In [54]: Top_Run_Players=df.sort_values(by=['Runs','Mat'],ascending=False)
```

```
In [55]: Top_30_Run_Players=Top_Run_Players.iloc[:31,:]
```

```
In [60]: T_R_P=Top_30_Run_Players['Player']
T_R_P
```

```
Out[60]: 0      KL Rahul
1      Shikhar Dhawan
2      David Warner
3      Shreyas Iyer
4      Ishan Kishan
5      Quinton de Kock
6      Suryakumar Yadav
7      Devdutt Padikkal
8      Virat Kohli
9      AB de Villiers
10     Faf du Plessis
11     Shubman Gill
12     Manish Pandey
13     Mayank Agarwal
14     Eoin Morgan
15     Sanju Samson
16     Ambati Rayudu
17     Nicholas Pooran
19     Marcus Stoinis
18     Nitish Rana
20     Jonny Bairstow
21     Rishabh Pant
22     Rohit Sharma
23     Jos Buttler
24     Kane Williamson
25     Steve Smith
26     Shane Watson
27     Chris Gayle
28     Ben Stokes
29     Hardik Pandya
31     Kieron Pollard
Name: Player, dtype: object
```

```
In [62]: df1 = pd.DataFrame({'Top_30_Match_Players': T_M_P, 'Top_30_Run_Players': T_R_P})
df1
```

Out[62]:

	Top_30_Match_Players	Top_30_Run_Players
0	KL Rahul	KL Rahul
1	Shikhar Dhawan	Shikhar Dhawan
2	David Warner	David Warner
3	Shreyas Iyer	Shreyas Iyer
4	Ishan Kishan	Ishan Kishan
5	Quinton de Kock	Quinton de Kock
6	Suryakumar Yadav	Suryakumar Yadav
7	Devdutt Padikkal	Devdutt Padikkal
8	Virat Kohli	Virat Kohli
9	AB de Villiers	AB de Villiers
10	NaN	Faf du Plessis
11	Shubman Gill	Shubman Gill
12	Manish Pandey	Manish Pandey
13	NaN	Mayank Agarwal
14	Eoin Morgan	Eoin Morgan
15	Sanju Samson	Sanju Samson
16	NaN	Ambati Rayudu
17	Nicholas Pooran	Nicholas Pooran
18	Nitish Rana	Nitish Rana
19	Marcus Stoinis	Marcus Stoinis
20	NaN	Jonny Bairstow
21	Rishabh Pant	Rishabh Pant
22	NaN	Rohit Sharma
23	NaN	Jos Buttler
24	NaN	Kane Williamson
25	NaN	Steve Smith
26	NaN	Shane Watson
27	NaN	Chris Gayle
28	NaN	Ben Stokes
29	NaN	Hardik Pandya
31	Kieron Pollard	Kieron Pollard
49	Axar Patel	NaN
52	Washington Sundar	NaN

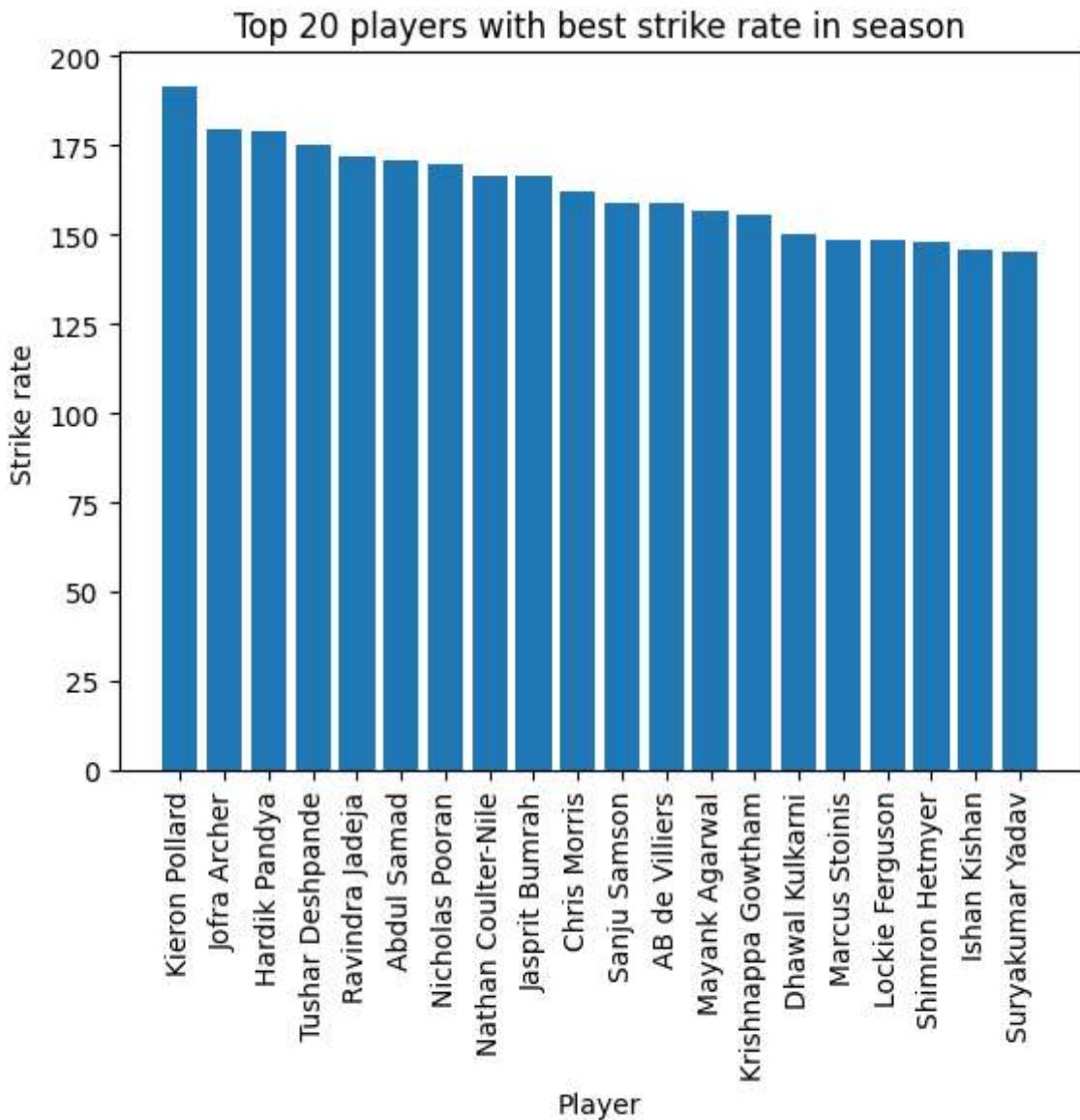
	Top_30_Match_Players	Top_30_Run_Players
54	Krunal Pandya	NaN
67	Kagiso Rabada	NaN
72	Ravichandran Ashwin	NaN
73	Rashid Khan	NaN
106	Anrich Nortje	NaN
110	Jasprit Bumrah	NaN
114	T Natarajan	NaN
116	Rahul Chahar	NaN
123	Yuzvendra Chahal	NaN
132	Trent Boult	NaN

Q14. Bar chart for top 20 players with best strike rate in a season.

```
In [66]: df['SR']

# Sort by strike rate in descending order and select top 20 players
top_20 = df.sort_values(by='SR', ascending=False).head(20)

# Create the bar chart
plt.bar(top_20['Player'], top_20['SR'])
plt.xticks(rotation=90)
plt.xlabel('Player')
plt.ylabel('Strike rate')
plt.title('Top 20 players with best strike rate in season')
plt.show()
```



Q15. Top 30 players with highest number of sixes.

```
In [72]: Top_30_6s=df.sort_values(by=['6s'],ascending=False).head(30)
Top_30_6s
```

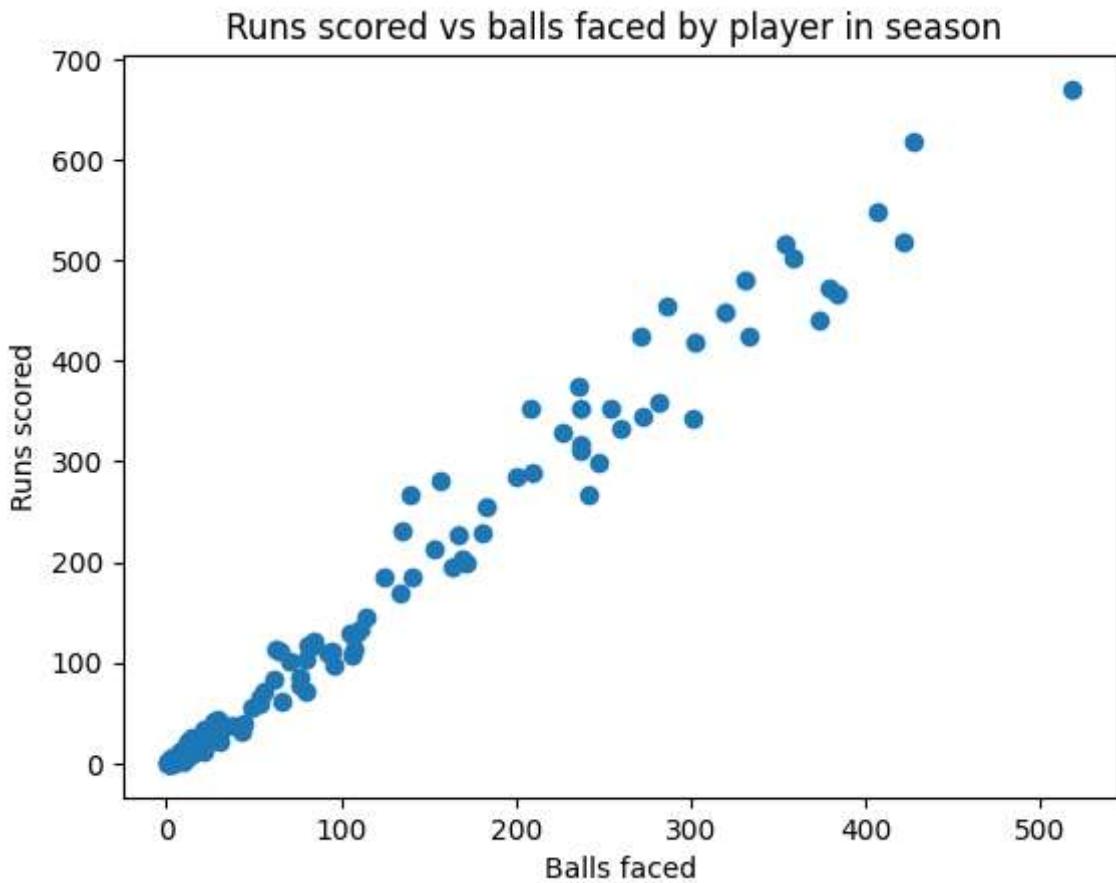
Out[72]:	POS	Player	Mat	Inns	NO	Runs	HS	Avg	BF	SR	100	50	4s	6s
	4	5 Ishan Kishan	14	13	4	516	99	57.33	354	145.76	0	4	36	30
	15	16 Sanju Samson	14	14	1	375	85	28.84	236	158.89	0	3	21	26
	29	30 Hardik Pandya	14	13	5	281	60*	35.12	157	178.98	0	1	14	25
	17	18 Nicholas Pooran	14	14	4	353	77	35.30	208	169.71	0	2	23	25
	14	15 Eoin Morgan	14	14	4	418	68*	41.80	302	138.41	0	1	32	24
	9	10 AB de Villiers	15	14	4	454	73*	45.40	286	158.74	0	5	33	23
	27	28 Chris Gayle	7	7	0	288	99	41.14	210	137.14	0	3	15	23
	0	1 KL Rahul	14	14	2	670	132*	55.83	518	129.34	1	5	58	23
	31	32 Kieron Pollard	16	12	7	268	60*	53.60	140	191.42	0	1	15	22
	5	6 Quinton de Kock	16	16	2	503	78*	35.92	358	140.50	0	4	46	22
	22	23 Rohit Sharma	12	12	0	332	80	27.66	260	127.69	0	3	27	19
	12	13 Manish Pandey	16	15	2	425	83*	32.69	333	127.62	0	3	35	18
	32	33 Rahul Tewatia	14	11	5	255	53	42.50	183	139.34	0	1	13	17
	3	4 Shreyas Iyer	17	17	2	519	88*	34.60	421	123.27	0	3	40	16
	23	24 Jos Buttler	13	12	2	328	70*	32.80	227	144.49	0	2	27	16
	19	20 Marcus Stoinis	17	17	3	352	65	25.14	237	148.52	0	3	31	16
	13	14 Mayank Agarwal	11	11	0	424	106	38.54	271	156.45	1	2	44	15
	10	11 Faf du Plessis	13	13	2	449	87*	40.81	319	140.75	0	4	42	14
	2	3 David Warner	16	16	2	548	85*	39.14	407	134.64	0	4	52	14
	20	21 Jonny Bairstow	11	11	0	345	97	31.36	272	126.83	0	3	31	13
	26	27 Shane Watson	11	11	1	299	83*	29.90	247	121.05	0	2	33	13
	40	41 Sam Curran	14	11	3	186	52	23.25	141	131.91	0	1	12	12
	18	19 Nitish Rana	14	14	0	352	87	25.14	254	138.58	0	3	43	12
	16	17 Ambati Rayudu	12	11	2	359	71	39.88	282	127.30	0	1	30	12
	41	42 Shimron Hetmyer	12	11	3	185	45	23.12	125	148.00	0	0	11	12
	1	2 Shikhar Dhawan	17	17	3	618	106*	44.14	427	144.73	2	4	67	12
	8	9 Virat Kohli	15	15	4	466	90*	42.36	384	121.35	0	3	23	11
	33	34 Ravindra Jadeja	14	11	6	232	50	46.40	135	171.85	0	1	22	11
	6	7 Suryakumar Yadav	16	15	3	480	79*	40.00	331	145.01	0	4	61	11
	24	25 Kane Williamson	12	11	4	317	67	45.28	237	133.75	0	3	26	10

In [73]: Top_30_6s['Player']

```
Out[73]: 4      Ishan Kishan
15      Sanju Samson
29      Hardik Pandya
17      Nicholas Pooran
14      Eoin Morgan
9       AB de Villiers
27      Chris Gayle
0       KL Rahul
31      Kieron Pollard
5       Quinton de Kock
22      Rohit Sharma
12      Manish Pandey
32      Rahul Tewatia
3       Shreyas Iyer
23      Jos Buttler
19      Marcus Stoinis
13      Mayank Agarwal
10      Faf du Plessis
2       David Warner
20      Jonny Bairstow
26      Shane Watson
40      Sam Curran
18      Nitish Rana
16      Ambati Rayudu
41      Shimron Hetmyer
1       Shikhar Dhawan
8       Virat Kohli
33      Ravindra Jadeja
6       Suryakumar Yadav
24      Kane Williamson
Name: Player, dtype: object
```

```
In [74]: # Q16. Scatter plot of runs scored by a player v/s balls faced in a season.
```

```
In [76]: # Create the scatter plot
plt.scatter(df['BF'], df['Runs'])
plt.xlabel('Balls faced')
plt.ylabel('Runs scored')
plt.title('Runs scored vs balls faced by player in season ')
plt.show()
```



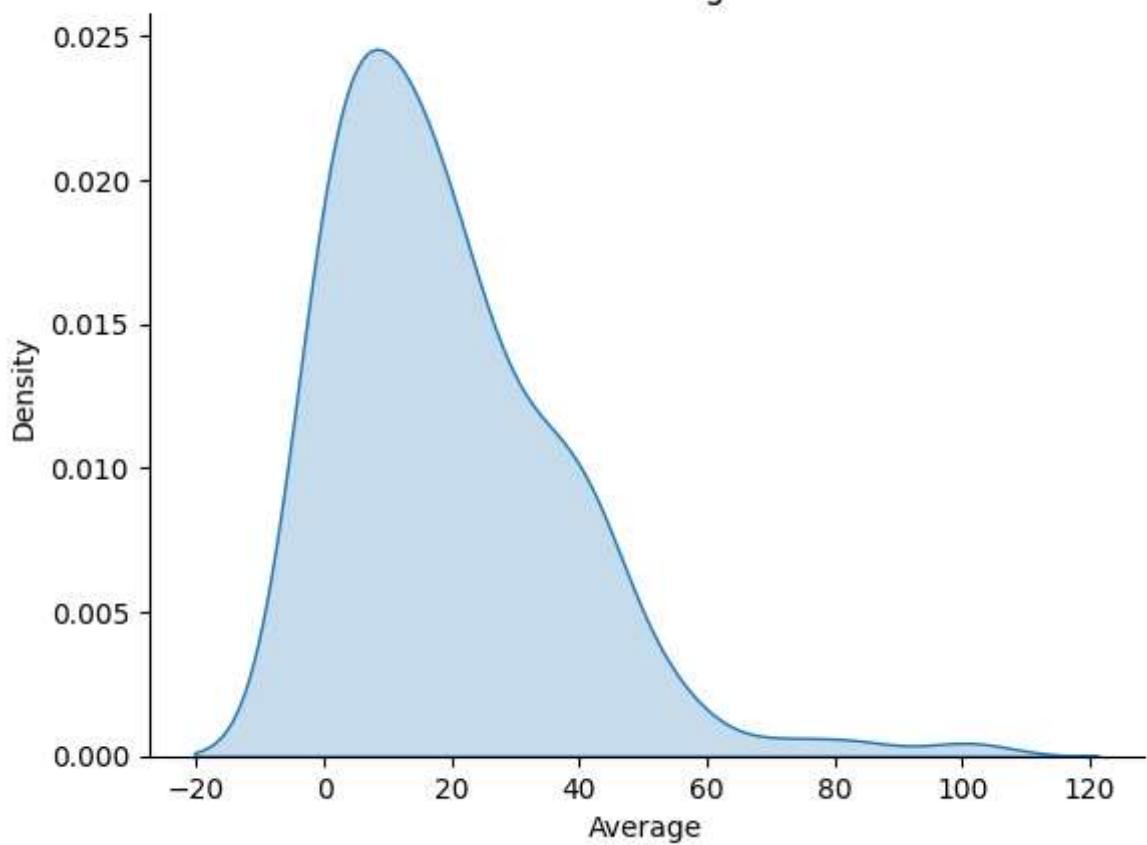
Q17. KDE plot for averages in a season.

```
In [80]: sns.kdeplot(df['Avg'], shade=True)
sns.despine()
plt.xlabel('Average')
plt.ylabel('Density')
plt.title('Distribution of averages in season')
plt.show()
```

C:\Users\acer\AppData\Local\Temp\ipykernel_3188\3556694676.py:1: FutureWarning:
`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

```
sns.kdeplot(df['Avg'], shade=True)
```

Distribution of averages in season



In []: