MINI PROJECT OF IPL DATA ANALYSES AND VISUALIZATION

-- Making the team of best 11 -based on ICC World Cup

In [1]:

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

Loading the data

To import the dataset we will use the pandas package from python programming language.

In [2]:

```
data =  pd.read_csv("E:\python\IPLdata.csv")
```

In [3]:

```
#first look of the data ,head() returns the first 5 rows of dataset.
data.head()
```

Out[3]:

	Player Name	Team	Nationality	Player_Type	Capped	Matches_Played	Runs	Average	Stri
0	Shikhar Dhawan	Punjab	Indian	Batter	1	192.0	5783.0	34.63	
1	Shreyas Iyer	Kolkata	Indian	Batter	1	87.0	2375.0	31.67	
2	Faf Du Plessis	Bangalore	Overseas	Batter	1	100.0	2935.0	34.94	
3	Manish Pandey	Lucknow	Indian	Batter	1	154.0	3560.0	30.69	
4	Shimron Hetmyer	Rajasthan	Overseas	Batter	1	31.0	517.0	25.85	

In [4]:

```
data
```

Out[4]:

	Player Name	Team	Nationality	Player_Type	Capped	Matches_Played	Runs	Average	Strike_Rate	W
0	Shikhar Dhawan	Punjab	Indian	Batter	1	192.0	5783.0	34.63	126.60	
1	Shreyas Iyer	Kolkata	Indian	Batter	1	87.0	2375.0	31.67	123.96	
2	Faf Du Plessis	Bangalore	Overseas	Batter	1	100.0	2935.0	34.94	131.09	
3	Manish Pandey	Lucknow	Indian	Batter	1	154.0	3560.0	30.69	121.83	
4	Shimron Hetmyer	Rajasthan	Overseas	Batter	1	31.0	517.0	25.85	151.17	
...	...	...	...	...	...	...	...	...	...	

In [5]:

```
#we can check the number of null values using isna().sum() method.
data.isna().sum()
```

Out[5]:

```
Player Name            0
Team                   0
Nationality            0
Player_Type            0
Capped                 0
Matches_Played        20
Runs                  70
Average               74
Strike_Rate           72
Wickets               95
Bowling_average      100
Economy               92
Bowling_Strike_Rate  116
Catches              208
Run_outs             208
Stumps               208
dtype: int64
```

In [6]:

```
# info() method will return the information
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 235 entries, 0 to 234
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Player Name          235 non-null    object 
 1   Team                 235 non-null    object 
 2   Nationality          235 non-null    object 
 3   Player_Type          235 non-null    object 
 4   Capped               235 non-null    int64  
 5   Matches_Played       215 non-null    float64
 6   Runs                 165 non-null    float64
 7   Average              161 non-null    float64
 8   Strike_Rate          163 non-null    float64
 9   Wickets              140 non-null    float64
 10  Bowling_average      135 non-null    float64
 11  Economy              143 non-null    float64
 12  Bowling_Strike_Rate  119 non-null    float64
 13  Catches              27 non-null     float64
 14  Run_outs             27 non-null     float64
 15  Stumps               27 non-null     float64
dtypes: float64(11), int64(1), object(4)
memory usage: 29.5+ KB
```

Cleaning the data

The players in the data can be intially segragated based on following parameters

1. Capped-Batters,Bowlers,Allrounder,Wicket keeper
2. UnCapped-Batters,Bowlers,Allrounder,Wicket keeper Uncapped playing for the first time in IPL

In [7]:

```
#segregating Data -Capped batters
batters = data.loc[(data["Player_Type"] == "Batter")]

batters_new = batters.loc[(batters["Capped"] == 1)]
Capped_Batters = batters_new[['Player Name','Team','Nationality','Matches_Played','Runs
```

In [8]:

```
Capped_Batters.head()
```

Out[8]:

	Player Name	Team	Nationality	Matches_Played	Runs	Average	Strike_Rate
0	Shikhar Dhawan	Punjab	Indian	192.0	5783.0	34.63	126.60
1	Shreyas Iyer	Kolkata	Indian	87.0	2375.0	31.67	123.96
2	Faf Du Plessis	Bangalore	Overseas	100.0	2935.0	34.94	131.09
3	Manish Pandey	Lucknow	Indian	154.0	3560.0	30.69	121.83
4	Shimron Hetmyer	Rajasthan	Overseas	31.0	517.0	25.85	151.17

In [9]:

```
#segregating Data - Capped Bowlers
bowlers = data.loc[(data["Player_Type"] == "Bowler ")]

bowlers_new = bowlers.loc[(bowlers["Capped"] == 1)]
Capped_Bowlers = bowlers_new[['Player Name','Team','Nationality','Matches_Played','Wic
```

In [10]:

```
Capped_Bowlers.head()
```

Out[10]:

	Player Name	Team	Nationality	Matches_Played	Wickets	Bowling_average	Economy	I
36	Kagiso Rabada	Punjab	Overseas	50.0	76.0	20.53	8.21	
37	Trent Boult	Rajasthan	Overseas	62.0	76.0	26.09	8.40	
38	Mohammad Shami	Gujarat	Indian	77.0	79.0	30.41	8.63	
39	T Natarajan	Hyderabad	Indian	24.0	20.0	34.40	8.24	
40	Deepak Chahar	Chennai	Indian	63.0	59.0	29.19	7.80	

In [11]:

```python
#segregating Data- Capped Keepers
Keepers = data.loc[(data["Player_Type"] == "Keeper")]
Keepers_new = Keepers.loc[(Keepers["Capped"] == 1)]
Capped_Keepers = Keepers_new[['Player Name',
                              'Team',
                              'Nationality',
                              'Matches_Played',
                              'Runs',
                              'Strike_Rate',
                              'Average',
                              'Catches',
                              'Run_outs',
                             'Stumps']]
```

In [12]:

```python
Capped_Keepers.head()
```

Out[12]:

	Player Name	Team	Nationality	Matches_Played	Runs	Strike_Rate	Average	Catches
105	Quinton De Kock	Lucknow	Overseas	77.0	2256.0	130.9	31.3	53.0
106	Ambati Rayudu	Chennai	Indian	175.0	3916.0	127.5	29.4	58.0
107	Ishan Kishan	Mumbai	Indian	61.0	1452.0	136.3	28.5	19.0
108	Jonny Bairstow	Punjab	Overseas	28.0	1038.0	142.2	41.5	18.0
109	Dinesh Karthik	Bangalore	Indian	213.0	4046.0	129.7	25.8	123.0

In [13]:

```python
#segragating Data - Capped Allrounders
Allrounders = data.loc[(data["Player_Type"] == "Allrounder")]
Allrounders_new = Allrounders.loc[(Allrounders["Capped"] == 1)]
Capped_Allrounders = Allrounders_new[['Player Name',
                              'Team',
                              'Nationality',
                              'Matches_Played',
                              'Runs',
                              'Average',
                              'Strike_Rate',
                              'Wickets',
                             'Bowling_average',
                             'Economy',
                            'Bowling_Strike_Rate']]
```

In [14]:

```
Capped_Allrounders.head()
```

Out[14]:

	Player Name	Team	Nationality	Matches_Played	Runs	Average	Strike_Rate	Wicket
127	Ravichandran Ashwin	Rajasthan	Indian	167.0	456.0	11.12	109.88	145.
128	Pat Cummins	Kolkata	Overseas	37.0	316.0	19.75	140.44	38.
129	Dwayne Bravo	Chennai	Overseas	151.0	1537.0	22.94	130.25	167.
130	Nitish Rana	Kolkata	Indian	77.0	1820.0	28.00	132.46	7.
131	Jason Holder	Lucknow	Overseas	26.0	189.0	14.54	121.15	35.

In [15]:

```
#Cleaning the data by the Null or Nan Values 0.

Capped_Battters = Capped_Batters.fillna(0)
Capped_Bowlers = Capped_Bowlers.fillna(0)
Capped_Keepers = Capped_Keepers.fillna(0)
Capped_Allrounders = Capped_Allrounders.fillna(0)
```

In [16]:

```python
#Checking null values in the data

print(Capped_Batters.isna().sum())
print(Capped_Bowlers.isna().sum())
print(Capped_Allrounders.isna().sum())
print(Capped_Keepers.isna().sum())
```

```
Player Name       0
Team              0
Nationality       0
Matches_Played    0
Runs              0
Average           0
Strike_Rate       0
dtype: int64
Player Name            0
Team                   0
Nationality            0
Matches_Played         0
Wickets                0
Bowling_average        0
Economy                0
Bowling_Strike_Rate    0
dtype: int64
Player Name            0
Team                   0
Nationality            0
Matches_Played         0
Runs                   0
Average                0
Strike_Rate            0
Wickets                0
Bowling_average        0
Economy                0
Bowling_Strike_Rate    0
dtype: int64
Player Name       0
Team              0
Nationality       0
Matches_Played    0
Runs              0
Strike_Rate       0
Average           0
Catches           0
Run_outs          0
Stumps            0
dtype: int64
```

Initial Analysis

In this section we will analyze the data for Batters,Keepers.Bowlers,Allrounder.

In [17]:

```
#Analyzing the Batters data

top_batters = Capped_Batters.loc[(Capped_Batters["Average"] >= 32.0)]

top_batters_average = top_batters.sort_values('Average',ascending= False)
top_batters_strike_rate = top_batters.sort_values('Strike_Rate',ascending=False)
top_batters_run = top_batters.sort_values('Runs',ascending=False)
top_batters_matches = top_batters.sort_values('Matches_Played',ascending=False)
```

In [18]:

```
# The data of Batters in descending order of batting averages.
top_batters_average
```

Out[18]:

	Player Name	Team	Nationality	Matches_Played	Runs	Average	Strike_Rate
212	KL Rahul	Lucknow	Indian	94.0	3273.0	47.43	136.38
231	Ruturaj Gaikwad	Chennai	Indian	22.0	839.0	46.61	132.13
19	David Warner	Delhi	Overseas	150.0	5449.0	41.60	139.97
207	Kane Williamson	Hyderabad	Overseas	63.0	1885.0	40.11	131.27
208	Virat Kohli	Bangalore	Indian	207.0	6283.0	37.40	129.95
2	Faf Du Plessis	Bangalore	Overseas	100.0	2935.0	34.94	131.09
0	Shikhar Dhawan	Punjab	Indian	192.0	5783.0	34.63	126.60
26	David Miller	Gujarat	Overseas	89.0	1974.0	32.90	136.51

In [19]:

```
# The data of Batters in descending order of Strike rate.
top_batters_strike_rate
```

Out[19]:

	Player Name	Team	Nationality	Matches_Played	Runs	Average	Strike_Rate
19	David Warner	Delhi	Overseas	150.0	5449.0	41.60	139.97
26	David Miller	Gujarat	Overseas	89.0	1974.0	32.90	136.51
212	KL Rahul	Lucknow	Indian	94.0	3273.0	47.43	136.38
231	Ruturaj Gaikwad	Chennai	Indian	22.0	839.0	46.61	132.13
207	Kane Williamson	Hyderabad	Overseas	63.0	1885.0	40.11	131.27
2	Faf Du Plessis	Bangalore	Overseas	100.0	2935.0	34.94	131.09
208	Virat Kohli	Bangalore	Indian	207.0	6283.0	37.40	129.95
0	Shikhar Dhawan	Punjab	Indian	192.0	5783.0	34.63	126.60

In [20]:

```
# The data of Batters in descending order of Runs.
top_batters_run
```

Out[20]:

	Player Name	Team	Nationality	Matches_Played	Runs	Average	Strike_Rate
208	Virat Kohli	Bangalore	Indian	207.0	6283.0	37.40	129.95
0	Shikhar Dhawan	Punjab	Indian	192.0	5783.0	34.63	126.60
19	David Warner	Delhi	Overseas	150.0	5449.0	41.60	139.97
212	KL Rahul	Lucknow	Indian	94.0	3273.0	47.43	136.38
2	Faf Du Plessis	Bangalore	Overseas	100.0	2935.0	34.94	131.09
26	David Miller	Gujarat	Overseas	89.0	1974.0	32.90	136.51
207	Kane Williamson	Hyderabad	Overseas	63.0	1885.0	40.11	131.27
231	Ruturaj Gaikwad	Chennai	Indian	22.0	839.0	46.61	132.13

In [21]:

```
# The data of Batters in descending order of Matches.
top_batters_matches
```

Out[21]:

	Player Name	Team	Nationality	Matches_Played	Runs	Average	Strike_Rate
208	Virat Kohli	Bangalore	Indian	207.0	6283.0	37.40	129.95
0	Shikhar Dhawan	Punjab	Indian	192.0	5783.0	34.63	126.60
19	David Warner	Delhi	Overseas	150.0	5449.0	41.60	139.97
2	Faf Du Plessis	Bangalore	Overseas	100.0	2935.0	34.94	131.09
212	KL Rahul	Lucknow	Indian	94.0	3273.0	47.43	136.38
26	David Miller	Gujarat	Overseas	89.0	1974.0	32.90	136.51
207	Kane Williamson	Hyderabad	Overseas	63.0	1885.0	40.11	131.27
231	Ruturaj Gaikwad	Chennai	Indian	22.0	839.0	46.61	132.13

From our analyses,if we rank 0-10, the top three batters are:

1. David Warner
2. KL Rahul
3. Virat Kohli

In [22]:

```
#Analyzing the Bowlers Data
top_bowlers = Capped_Bowlers.loc[(Capped_Bowlers["Bowling_average"] <= 24.0)]

top_bowlers_average = top_bowlers.sort_values('Bowling_average')
top_bowlers_strike_rate = top_bowlers.sort_values('Bowling_Strike_Rate')
top_bowlers_Wickets = top_bowlers.sort_values('Wickets',ascending=False)
top_bowlers_economy = top_bowlers.sort_values('Economy')
top_bowlers_matches = top_bowlers.sort_values('Matches_Played',ascending=False)
```

In [23]:

```
# This Dataframe the bowling average in ascending order.
top_bowlers_average
```

Out[23]:

	Player Name	Team	Nationality	Matches_Played	Wickets	Bowling_average	Economy
83	Rasikh Dar	Kolkata	Indian	1.0	1.0	0.00	10.50
87	Sean Abbott	Hyderabad	Overseas	2.0	0.0	0.00	11.40
101	Kuldip Yadav	Rajasthan	Indian	1.0	0.0	0.00	8.00
88	Alzarri Joseph	Gujarat	Overseas	3.0	6.0	14.50	10.04
93	Lungi Ngidi	Delhi	Overseas	14.0	25.0	17.92	8.30
36	Kagiso Rabada	Punjab	Overseas	50.0	76.0	20.53	8.21
234	Anrich Nortje	Delhi	Overseas	24.0	34.0	20.56	7.65
103	Nathan Coulter-Nile	Rajasthan	Overseas	38.0	48.0	21.92	7.52
50	Yuzvendra Chahal	Rajasthan	Indian	114.0	139.0	22.28	7.59
210	Arshdeep Singh	Punjab	Indian	23.0	30.0	22.30	8.78
215	Jasprit Bumrah	Mumbai	Indian	106.0	130.0	23.05	7.42
214	Varun Chakravarthy	Kolkata	Indian	31.0	36.0	23.31	6.82

In [24]:

```
# This Dataframe consist Strike Rate in ascending order.
top_bowlers_strike_rate
```

Out[24]:

	Player Name	Team	Nationality	Matches_Played	Wickets	Bowling_average	Economy
83	Rasikh Dar	Kolkata	Indian	1.0	1.0	0.00	10.50
87	Sean Abbott	Hyderabad	Overseas	2.0	0.0	0.00	11.40
101	Kuldip Yadav	Rajasthan	Indian	1.0	0.0	0.00	8.00
88	Alzarri Joseph	Gujarat	Overseas	3.0	6.0	14.50	10.04
93	Lungi Ngidi	Delhi	Overseas	14.0	25.0	17.92	8.30
36	Kagiso Rabada	Punjab	Overseas	50.0	76.0	20.53	8.21
210	Arshdeep Singh	Punjab	Indian	23.0	30.0	22.30	8.78
234	Anrich Nortje	Delhi	Overseas	24.0	34.0	20.56	7.65
103	Nathan Coulter-Nile	Rajasthan	Overseas	38.0	48.0	21.92	7.52
50	Yuzvendra Chahal	Rajasthan	Indian	114.0	139.0	22.28	7.59
215	Jasprit Bumrah	Mumbai	Indian	106.0	130.0	23.05	7.42
214	Varun Chakravarthy	Kolkata	Indian	31.0	36.0	23.31	6.82

In [25]:

```
# This Dataframe consist economy in ascending order.
top_bowlers_economy
```

Out[25]:

	Player Name	Team	Nationality	Matches_Played	Wickets	Bowling_average	Economy
214	Varun Chakravarthy	Kolkata	Indian	31.0	36.0	23.31	6.82
215	Jasprit Bumrah	Mumbai	Indian	106.0	130.0	23.05	7.42
103	Nathan Coulter-Nile	Rajasthan	Overseas	38.0	48.0	21.92	7.52
50	Yuzvendra Chahal	Rajasthan	Indian	114.0	139.0	22.28	7.59
234	Anrich Nortje	Delhi	Overseas	24.0	34.0	20.56	7.65
101	Kuldip Yadav	Rajasthan	Indian	1.0	0.0	0.00	8.00
36	Kagiso Rabada	Punjab	Overseas	50.0	76.0	20.53	8.21
93	Lungi Ngidi	Delhi	Overseas	14.0	25.0	17.92	8.30
210	Arshdeep Singh	Punjab	Indian	23.0	30.0	22.30	8.78
88	Alzarri Joseph	Gujarat	Overseas	3.0	6.0	14.50	10.04
83	Rasikh Dar	Kolkata	Indian	1.0	1.0	0.00	10.50
87	Sean Abbott	Hyderabad	Overseas	2.0	0.0	0.00	11.40

In [26]:

```
# This Dataframe consist Wickets in descending order.
top_bowlers_Wickets
```

Out[26]:

	Player Name	Team	Nationality	Matches_Played	Wickets	Bowling_average	Economy
50	Yuzvendra Chahal	Rajasthan	Indian	114.0	139.0	22.28	7.59
215	Jasprit Bumrah	Mumbai	Indian	106.0	130.0	23.05	7.42
36	Kagiso Rabada	Punjab	Overseas	50.0	76.0	20.53	8.21
103	Nathan Coulter-Nile	Rajasthan	Overseas	38.0	48.0	21.92	7.52
214	Varun Chakravarthy	Kolkata	Indian	31.0	36.0	23.31	6.82
234	Anrich Nortje	Delhi	Overseas	24.0	34.0	20.56	7.65
210	Arshdeep Singh	Punjab	Indian	23.0	30.0	22.30	8.78
93	Lungi Ngidi	Delhi	Overseas	14.0	25.0	17.92	8.30
88	Alzarri Joseph	Gujarat	Overseas	3.0	6.0	14.50	10.04
83	Rasikh Dar	Kolkata	Indian	1.0	1.0	0.00	10.50
87	Sean Abbott	Hyderabad	Overseas	2.0	0.0	0.00	11.40
101	Kuldip Yadav	Rajasthan	Indian	1.0	0.0	0.00	8.00

In [27]:

```
# This Dataframe consist Matches in descending order.
top_bowlers_matches
```

Out[27]:

	Player Name	Team	Nationality	Matches_Played	Wickets	Bowling_average	Economy
50	Yuzvendra Chahal	Rajasthan	Indian	114.0	139.0	22.28	7.59
215	Jasprit Bumrah	Mumbai	Indian	106.0	130.0	23.05	7.42
36	Kagiso Rabada	Punjab	Overseas	50.0	76.0	20.53	8.21
103	Nathan Coulter-Nile	Rajasthan	Overseas	38.0	48.0	21.92	7.52
214	Varun Chakravarthy	Kolkata	Indian	31.0	36.0	23.31	6.82
234	Anrich Nortje	Delhi	Overseas	24.0	34.0	20.56	7.65
210	Arshdeep Singh	Punjab	Indian	23.0	30.0	22.30	8.78
93	Lungi Ngidi	Delhi	Overseas	14.0	25.0	17.92	8.30
88	Alzarri Joseph	Gujarat	Overseas	3.0	6.0	14.50	10.04
87	Sean Abbott	Hyderabad	Overseas	2.0	0.0	0.00	11.40
83	Rasikh Dar	Kolkata	Indian	1.0	1.0	0.00	10.50
101	Kuldip Yadav	Rajasthan	Indian	1.0	0.0	0.00	8.00

From our analyses,if we rank 0-10, the top three bowlers are:

1. Kagiso Rabada
2. Jasprit Bumrah
3. Yuzvendra Chahal
4. Nathan Coulter-Nile

In [28]:

```
#Analyzing the All rounders Data
top_allrounders = Capped_Allrounders.loc[(Capped_Allrounders["Strike_Rate"] >=140)]

top_allrounders_average = top_allrounders.sort_values('Average',ascending = False)
top_allrounders_strike_Rate = top_allrounders.sort_values('Strike_Rate',ascending = Fal
top_allrounders_runs= top_allrounders.sort_values('Runs',ascending = False)
top_allrounders_matches = top_allrounders.sort_values('Matches_Played',ascending = Fals
top_allrounders_bowling_strike_rate = top_allrounders.sort_values('Bowling_Strike_Rate'
top_allrounders_bowling_average = top_allrounders.sort_values('Bowling_average')
top_allrounders_wickets = top_allrounders.sort_values('Wickets',ascending = False)
top_allrounders_economy = top_allrounders.sort_values('Economy')
```

In [29]:

```
#the dataframe consist average for top Allrounder in  Descending Order .
top_allrounders_average
```

Out[29]:

	Player Name	Team	Nationality	Matches_Played	Runs	Average	Strike_Rate	Wickets
233	Kieron Pollard	Mumbai	Overseas	178.0	3268.0	29.98	149.77	65.0
204	Andre Russell	Kolkata	Overseas	84.0	1700.0	29.31	178.57	72.0
211	Hardik Pandya	Gujarat	Indian	92.0	1476.0	27.33	153.91	42.0
218	Glen Maxwell	Bangalore	Overseas	97.0	2018.0	25.23	151.84	22.0
223	Moeen Ali	Chennai	Overseas	34.0	666.0	22.97	146.37	16.0
128	Pat Cummins	Kolkata	Overseas	37.0	316.0	19.75	140.44	38.0
232	Sunil Narine	Kolkata	Overseas	134.0	954.0	15.64	161.69	143.0
165	Jofra Archer	Mumbai	Overseas	35.0	195.0	15.00	157.26	46.0
195	Mohammad Nabi	Kolkata	Overseas	17.0	180.0	15.00	151.26	13.0
154	K Gowtham	Lucknow	Indian	24.0	186.0	14.31	169.09	13.0
202	Aman Khan	Kolkata	Indian	5.0	40.0	13.30	148.10	0.0

In [30]:

```
#the dataframe consist Runs for top Allrounder in  Descending Order .
top_allrounders_runs
```

Out[30]:

	Player Name	Team	Nationality	Matches_Played	Runs	Average	Strike_Rate	Wickets
233	Kieron Pollard	Mumbai	Overseas	178.0	3268.0	29.98	149.77	65.0
218	Glen Maxwell	Bangalore	Overseas	97.0	2018.0	25.23	151.84	22.0
204	Andre Russell	Kolkata	Overseas	84.0	1700.0	29.31	178.57	72.0
211	Hardik Pandya	Gujarat	Indian	92.0	1476.0	27.33	153.91	42.0
232	Sunil Narine	Kolkata	Overseas	134.0	954.0	15.64	161.69	143.0
223	Moeen Ali	Chennai	Overseas	34.0	666.0	22.97	146.37	16.0
128	Pat Cummins	Kolkata	Overseas	37.0	316.0	19.75	140.44	38.0
165	Jofra Archer	Mumbai	Overseas	35.0	195.0	15.00	157.26	46.0
154	K Gowtham	Lucknow	Indian	24.0	186.0	14.31	169.09	13.0
195	Mohammad Nabi	Kolkata	Overseas	17.0	180.0	15.00	151.26	13.0
202	Aman Khan	Kolkata	Indian	5.0	40.0	13.30	148.10	0.0

In [31]:

```
#the dataframe consist strike rate for top Allrounder in  ascending Order .
top_allrounders_strike_Rate
```

Out[31]:

	Player Name	Team	Nationality	Matches_Played	Runs	Average	Strike_Rate	Wickets
204	Andre Russell	Kolkata	Overseas	84.0	1700.0	29.31	178.57	72.0
154	K Gowtham	Lucknow	Indian	24.0	186.0	14.31	169.09	13.0
232	Sunil Narine	Kolkata	Overseas	134.0	954.0	15.64	161.69	143.0
165	Jofra Archer	Mumbai	Overseas	35.0	195.0	15.00	157.26	46.0
211	Hardik Pandya	Gujarat	Indian	92.0	1476.0	27.33	153.91	42.0
218	Glen Maxwell	Bangalore	Overseas	97.0	2018.0	25.23	151.84	22.0
195	Mohammad Nabi	Kolkata	Overseas	17.0	180.0	15.00	151.26	13.0
233	Kieron Pollard	Mumbai	Overseas	178.0	3268.0	29.98	149.77	65.0
202	Aman Khan	Kolkata	Indian	5.0	40.0	13.30	148.10	0.0
223	Moeen Ali	Chennai	Overseas	34.0	666.0	22.97	146.37	16.0
128	Pat Cummins	Kolkata	Overseas	37.0	316.0	19.75	140.44	38.0

In [32]:

```
#the dataframe Matches average for top Allrounder in  Descending Order .
top_allrounders_matches
```

Out[32]:

	Player Name	Team	Nationality	Matches_Played	Runs	Average	Strike_Rate	Wickets
233	Kieron Pollard	Mumbai	Overseas	178.0	3268.0	29.98	149.77	65.0
232	Sunil Narine	Kolkata	Overseas	134.0	954.0	15.64	161.69	143.0
218	Glen Maxwell	Bangalore	Overseas	97.0	2018.0	25.23	151.84	22.0
211	Hardik Pandya	Gujarat	Indian	92.0	1476.0	27.33	153.91	42.0
204	Andre Russell	Kolkata	Overseas	84.0	1700.0	29.31	178.57	72.0
128	Pat Cummins	Kolkata	Overseas	37.0	316.0	19.75	140.44	38.0
165	Jofra Archer	Mumbai	Overseas	35.0	195.0	15.00	157.26	46.0
223	Moeen Ali	Chennai	Overseas	34.0	666.0	22.97	146.37	16.0
154	K Gowtham	Lucknow	Indian	24.0	186.0	14.31	169.09	13.0
195	Mohammad Nabi	Kolkata	Overseas	17.0	180.0	15.00	151.26	13.0
202	Aman Khan	Kolkata	Indian	5.0	40.0	13.30	148.10	0.0

In [33]:

```
#the dataframe consist bowling average for top Allrounder in  Descending Order .
top_allrounders_bowling_average
```

Out[33]:

	Player Name	Team	Nationality	Matches_Played	Runs	Average	Strike_Rate	Wickets
202	Aman Khan	Kolkata	Indian	5.0	40.0	13.30	148.10	0.0
165	Jofra Archer	Mumbai	Overseas	35.0	195.0	15.00	157.26	46.0
232	Sunil Narine	Kolkata	Overseas	134.0	954.0	15.64	161.69	143.0
204	Andre Russell	Kolkata	Overseas	84.0	1700.0	29.31	178.57	72.0
223	Moeen Ali	Chennai	Overseas	34.0	666.0	22.97	146.37	16.0
128	Pat Cummins	Kolkata	Overseas	37.0	316.0	19.75	140.44	38.0
211	Hardik Pandya	Gujarat	Indian	92.0	1476.0	27.33	153.91	42.0
195	Mohammad Nabi	Kolkata	Overseas	17.0	180.0	15.00	151.26	13.0
233	Kieron Pollard	Mumbai	Overseas	178.0	3268.0	29.98	149.77	65.0
218	Glen Maxwell	Bangalore	Overseas	97.0	2018.0	25.23	151.84	22.0
154	K Gowtham	Lucknow	Indian	24.0	186.0	14.31	169.09	13.0

In [34]:

```
#the dataframe consist Wickets for top Allrounder in  Descending Order .
top_allrounders_wickets
```

Out[34]:

	Player Name	Team	Nationality	Matches_Played	Runs	Average	Strike_Rate	Wickets
232	Sunil Narine	Kolkata	Overseas	134.0	954.0	15.64	161.69	143.0
204	Andre Russell	Kolkata	Overseas	84.0	1700.0	29.31	178.57	72.0
233	Kieron Pollard	Mumbai	Overseas	178.0	3268.0	29.98	149.77	65.0
165	Jofra Archer	Mumbai	Overseas	35.0	195.0	15.00	157.26	46.0
211	Hardik Pandya	Gujarat	Indian	92.0	1476.0	27.33	153.91	42.0
128	Pat Cummins	Kolkata	Overseas	37.0	316.0	19.75	140.44	38.0
218	Glen Maxwell	Bangalore	Overseas	97.0	2018.0	25.23	151.84	22.0
223	Moeen Ali	Chennai	Overseas	34.0	666.0	22.97	146.37	16.0
154	K Gowtham	Lucknow	Indian	24.0	186.0	14.31	169.09	13.0
195	Mohammad Nabi	Kolkata	Overseas	17.0	180.0	15.00	151.26	13.0
202	Aman Khan	Kolkata	Indian	5.0	40.0	13.30	148.10	0.0

In [35]:

```
#the dataframe consist Economy for top Allrounder in  ascending Order .
top_allrounders_economy
```

Out[35]:

	Player Name	Team	Nationality	Matches_Played	Runs	Average	Strike_Rate	Wickets
232	Sunil Narine	Kolkata	Overseas	134.0	954.0	15.64	161.69	143.0
223	Moeen Ali	Chennai	Overseas	34.0	666.0	22.97	146.37	16.0
202	Aman Khan	Kolkata	Indian	5.0	40.0	13.30	148.10	0.0
165	Jofra Archer	Mumbai	Overseas	35.0	195.0	15.00	157.26	46.0
195	Mohammad Nabi	Kolkata	Overseas	17.0	180.0	15.00	151.26	13.0
128	Pat Cummins	Kolkata	Overseas	37.0	316.0	19.75	140.44	38.0
154	K Gowtham	Lucknow	Indian	24.0	186.0	14.31	169.09	13.0
218	Glen Maxwell	Bangalore	Overseas	97.0	2018.0	25.23	151.84	22.0
233	Kieron Pollard	Mumbai	Overseas	178.0	3268.0	29.98	149.77	65.0
204	Andre Russell	Kolkata	Overseas	84.0	1700.0	29.31	178.57	72.0
211	Hardik Pandya	Gujarat	Indian	92.0	1476.0	27.33	153.91	42.0

In [36]:

```
#the dataframe consist BOwling strike rate for top Allrounder in  Descending Order .
top_allrounders_bowling_strike_rate
```

Out[36]:

	Player Name	Team	Nationality	Matches_Played	Runs	Average	Strike_Rate	Wickets
202	Aman Khan	Kolkata	Indian	5.0	40.0	13.30	148.10	0.0
204	Andre Russell	Kolkata	Overseas	84.0	1700.0	29.31	178.57	72.0
165	Jofra Archer	Mumbai	Overseas	35.0	195.0	15.00	157.26	46.0
211	Hardik Pandya	Gujarat	Indian	92.0	1476.0	27.33	153.91	42.0
233	Kieron Pollard	Mumbai	Overseas	178.0	3268.0	29.98	149.77	65.0
232	Sunil Narine	Kolkata	Overseas	134.0	954.0	15.64	161.69	143.0
128	Pat Cummins	Kolkata	Overseas	37.0	316.0	19.75	140.44	38.0
223	Moeen Ali	Chennai	Overseas	34.0	666.0	22.97	146.37	16.0
195	Mohammad Nabi	Kolkata	Overseas	17.0	180.0	15.00	151.26	13.0
218	Glen Maxwell	Bangalore	Overseas	97.0	2018.0	25.23	151.84	22.0
154	K Gowtham	Lucknow	Indian	24.0	186.0	14.31	169.09	13.0

From our analyses,if we rank 0-10, the top three Allrounders are:

1. Andre Russell
2. Sunil Narine
3. Hardik Pandya
4. Jpfra Archer

In [37]:

```
# Analyzing The Keeper Data
top_keepers = Capped_Keepers.loc[(Capped_Keepers["Average"] >=25.0)]


top_keepers_average = top_keepers.sort_values('Average',ascending = False)
top_keepers_strike_rate = top_keepers.sort_values('Strike_Rate',ascending = False)
top_keepers_runs= top_keepers.sort_values('Runs',ascending = False)
top_keepers_matches = top_keepers.sort_values('Matches_Played',ascending = False)
top_keepers_catches = top_keepers.sort_values('Catches',ascending = False)
top_keepers_runouts = top_keepers.sort_values('Run_outs',ascending = False)
top_keepers_stumps = top_keepers.sort_values('Stumps',ascending = False)

```

In [38]:

```
#the dataframe consist Average of Keepers in Descending order.
top_keepers_average
```

Out[38]:

	Player Name	Team	Nationality	Matches_Played	Runs	Strike_Rate	Average	Catches	R
108	Jonny Bairstow	Punjab	Overseas	28.0	1038.0	142.20	41.50	18.0	
213	MS Dhoni	Chennai	Indian	220.0	4746.0	135.80	39.50	126.0	
111	KS Bharat	Delhi	Indian	8.0	191.0	122.40	38.20	4.0	
206	Rishabh Pant	Delhi	Indian	84.0	2498.0	147.46	35.18	56.0	
219	Jos Butler	Rajasthan	Overseas	65.0	1968.0	150.00	35.14	34.0	
105	Quinton De Kock	Lucknow	Overseas	77.0	2256.0	130.90	31.30	53.0	
106	Ambati Rayudu	Chennai	Indian	175.0	3916.0	127.50	29.40	58.0	
209	Sanju Samson	Rajasthan	Indian	121.0	3068.0	134.21	29.22	59.0	
107	Ishan Kishan	Mumbai	Indian	61.0	1452.0	136.30	28.50	19.0	
109	Dinesh Karthik	Bangalore	Indian	213.0	4046.0	129.70	25.80	123.0	

In [39]:

```
#the dataframe consist runs of Keepers in Descending order.
top_keepers_runs
```

Out[39]:

	Player Name	Team	Nationality	Matches_Played	Runs	Strike_Rate	Average	Catches	R
213	MS Dhoni	Chennai	Indian	220.0	4746.0	135.80	39.50	126.0	
109	Dinesh Karthik	Bangalore	Indian	213.0	4046.0	129.70	25.80	123.0	
106	Ambati Rayudu	Chennai	Indian	175.0	3916.0	127.50	29.40	58.0	
209	Sanju Samson	Rajasthan	Indian	121.0	3068.0	134.21	29.22	59.0	
206	Rishabh Pant	Delhi	Indian	84.0	2498.0	147.46	35.18	56.0	
105	Quinton De Kock	Lucknow	Overseas	77.0	2256.0	130.90	31.30	53.0	
219	Jos Butler	Rajasthan	Overseas	65.0	1968.0	150.00	35.14	34.0	
107	Ishan Kishan	Mumbai	Indian	61.0	1452.0	136.30	28.50	19.0	
108	Jonny Bairstow	Punjab	Overseas	28.0	1038.0	142.20	41.50	18.0	
111	KS Bharat	Delhi	Indian	8.0	191.0	122.40	38.20	4.0	

In [40]:

```
#the dataframe consist Strike Rate of Keepers in Descending order.
top_keepers_strike_rate
```

Out[40]:

	Player Name	Team	Nationality	Matches_Played	Runs	Strike_Rate	Average	Catches	F
219	Jos Butler	Rajasthan	Overseas	65.0	1968.0	150.00	35.14	34.0	
206	Rishabh Pant	Delhi	Indian	84.0	2498.0	147.46	35.18	56.0	
108	Jonny Bairstow	Punjab	Overseas	28.0	1038.0	142.20	41.50	18.0	
107	Ishan Kishan	Mumbai	Indian	61.0	1452.0	136.30	28.50	19.0	
213	MS Dhoni	Chennai	Indian	220.0	4746.0	135.80	39.50	126.0	
209	Sanju Samson	Rajasthan	Indian	121.0	3068.0	134.21	29.22	59.0	
105	Quinton De Kock	Lucknow	Overseas	77.0	2256.0	130.90	31.30	53.0	
109	Dinesh Karthik	Bangalore	Indian	213.0	4046.0	129.70	25.80	123.0	
106	Ambati Rayudu	Chennai	Indian	175.0	3916.0	127.50	29.40	58.0	
111	KS Bharat	Delhi	Indian	8.0	191.0	122.40	38.20	4.0	

In [41]:

```
#the dataframe consist Matches of Keepers in Descending order.
top_keepers_matches
```

Out[41]:

	Player Name	Team	Nationality	Matches_Played	Runs	Strike_Rate	Average	Catches	F
213	MS Dhoni	Chennai	Indian	220.0	4746.0	135.80	39.50	126.0	
109	Dinesh Karthik	Bangalore	Indian	213.0	4046.0	129.70	25.80	123.0	
106	Ambati Rayudu	Chennai	Indian	175.0	3916.0	127.50	29.40	58.0	
209	Sanju Samson	Rajasthan	Indian	121.0	3068.0	134.21	29.22	59.0	
206	Rishabh Pant	Delhi	Indian	84.0	2498.0	147.46	35.18	56.0	
105	Quinton De Kock	Lucknow	Overseas	77.0	2256.0	130.90	31.30	53.0	
219	Jos Butler	Rajasthan	Overseas	65.0	1968.0	150.00	35.14	34.0	
107	Ishan Kishan	Mumbai	Indian	61.0	1452.0	136.30	28.50	19.0	
108	Jonny Bairstow	Punjab	Overseas	28.0	1038.0	142.20	41.50	18.0	
111	KS Bharat	Delhi	Indian	8.0	191.0	122.40	38.20	4.0	

In [42]:

```
#the dataframe consist Catches of Keepers in Descending order.
top_keepers_catches
```

Out[42]:

	Player Name	Team	Nationality	Matches_Played	Runs	Strike_Rate	Average	Catches	F
213	MS Dhoni	Chennai	Indian	220.0	4746.0	135.80	39.50	126.0	
109	Dinesh Karthik	Bangalore	Indian	213.0	4046.0	129.70	25.80	123.0	
209	Sanju Samson	Rajasthan	Indian	121.0	3068.0	134.21	29.22	59.0	
106	Ambati Rayudu	Chennai	Indian	175.0	3916.0	127.50	29.40	58.0	
206	Rishabh Pant	Delhi	Indian	84.0	2498.0	147.46	35.18	56.0	
105	Quinton De Kock	Lucknow	Overseas	77.0	2256.0	130.90	31.30	53.0	
219	Jos Butler	Rajasthan	Overseas	65.0	1968.0	150.00	35.14	34.0	
107	Ishan Kishan	Mumbai	Indian	61.0	1452.0	136.30	28.50	19.0	
108	Jonny Bairstow	Punjab	Overseas	28.0	1038.0	142.20	41.50	18.0	
111	KS Bharat	Delhi	Indian	8.0	191.0	122.40	38.20	4.0	

In [43]:

```
#the dataframe consist Stumps of Keepers in Descending order.
top_keepers_stumps
```

Out[43]:

	Player Name	Team	Nationality	Matches_Played	Runs	Strike_Rate	Average	Catches	F
213	MS Dhoni	Chennai	Indian	220.0	4746.0	135.80	39.50	126.0	
109	Dinesh Karthik	Bangalore	Indian	213.0	4046.0	129.70	25.80	123.0	
105	Quinton De Kock	Lucknow	Overseas	77.0	2256.0	130.90	31.30	53.0	
206	Rishabh Pant	Delhi	Indian	84.0	2498.0	147.46	35.18	56.0	
209	Sanju Samson	Rajasthan	Indian	121.0	3068.0	134.21	29.22	59.0	
108	Jonny Bairstow	Punjab	Overseas	28.0	1038.0	142.20	41.50	18.0	
106	Ambati Rayudu	Chennai	Indian	175.0	3916.0	127.50	29.40	58.0	
107	Ishan Kishan	Mumbai	Indian	61.0	1452.0	136.30	28.50	19.0	
111	KS Bharat	Delhi	Indian	8.0	191.0	122.40	38.20	4.0	
219	Jos Butler	Rajasthan	Overseas	65.0	1968.0	150.00	35.14	34.0	

If we rank the keepers in order of 1- 10 on the abpbe parameters. The top 3 keepers will be

1. MS Dhoni
2. Dinesh Kartik
3. Rishabh Pant

Visualization For Enhanced Analysis

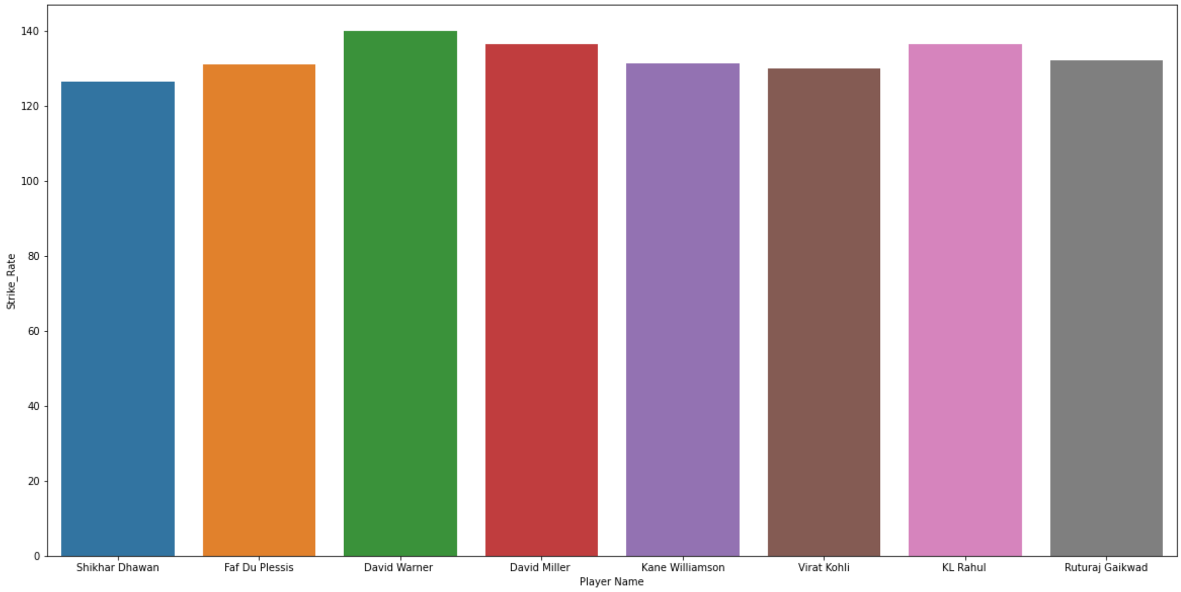
Now ,the initial analysis that has been done on the segregated data. You can create a visual representations of the same to get better analysis of the data.

In [44]:

```
# Visualization  for Batters Data
# the plot shows each batters strike rate

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Strike_Rate', data= top_batters)

```

Out[44]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Strike_Rate'>
```

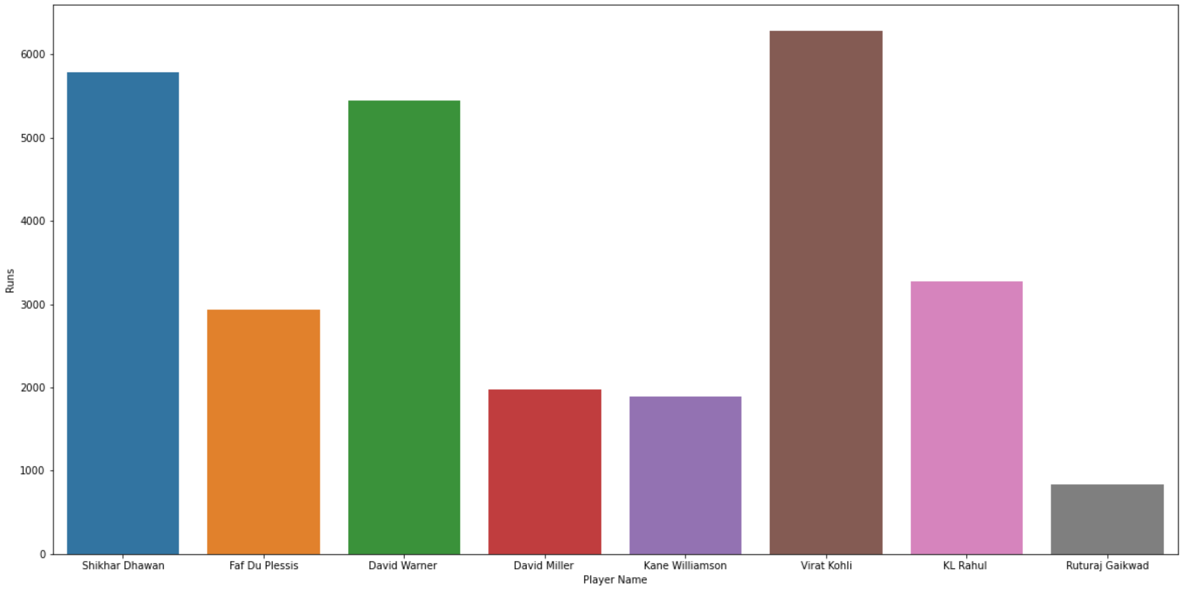


In [45]:

```python
# the plot shows each batters runs

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Runs', data= top_batters)
```

Out[45]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Runs'>
```

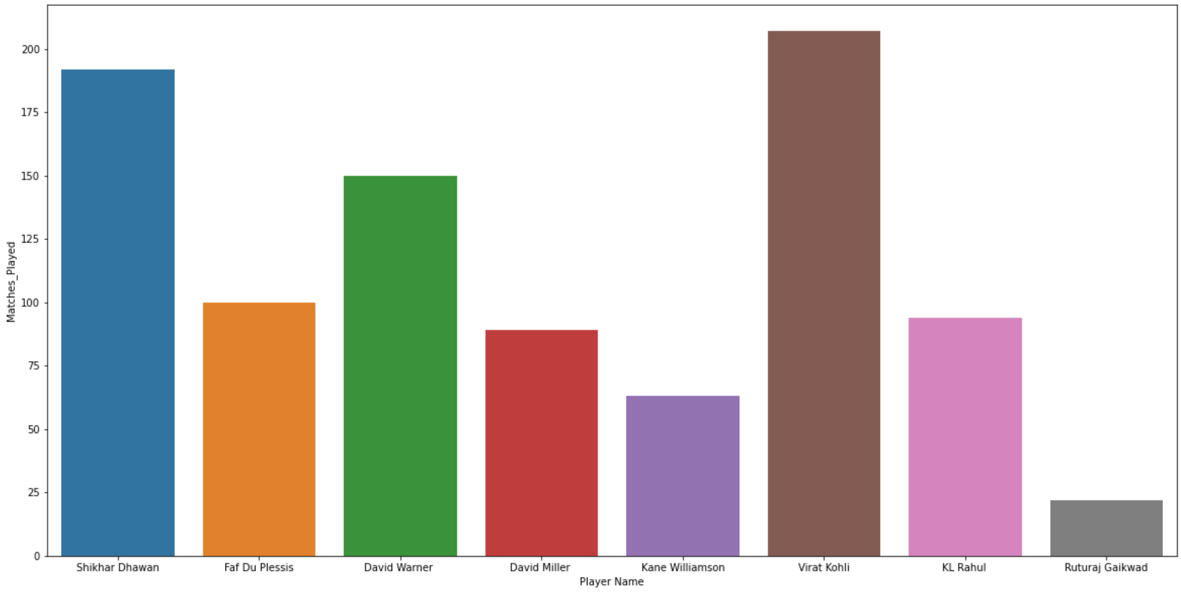


In [46]:

```python
# the plot shows each batters matches played

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Matches_Played', data= top_batters)
```

Out[46]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Matches_Played'>
```

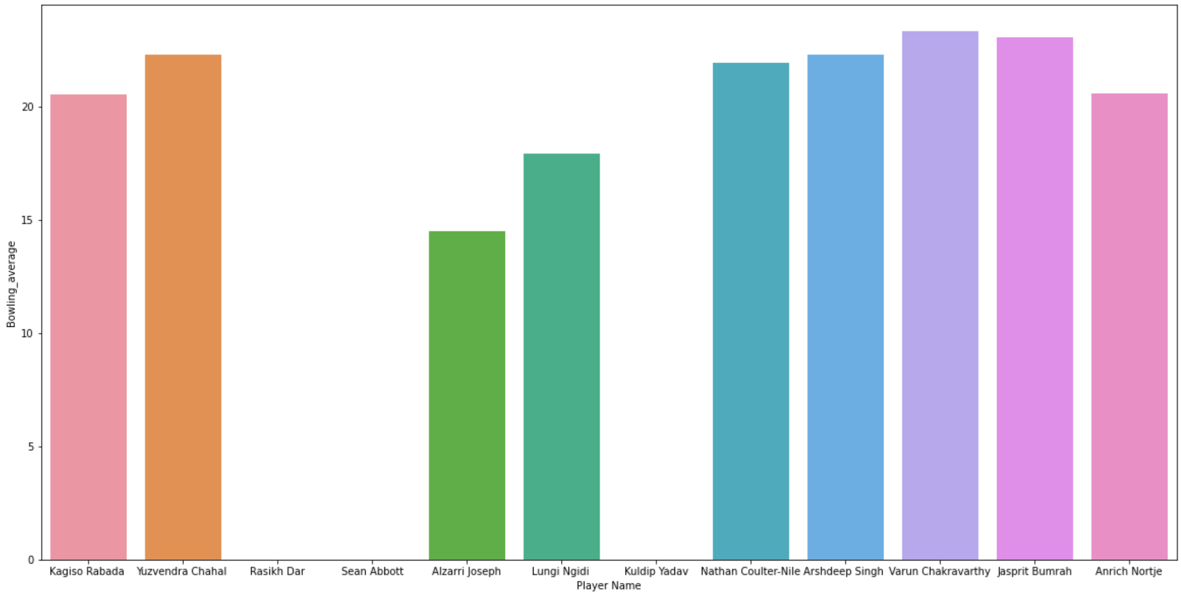


In [47]:

```
# Visualization  for Bowlers Data
# the plot shows bowling average of each of the two bowlers

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Bowling_average', data= top_bowlers)
```

Out[47]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Bowling_average'>
```

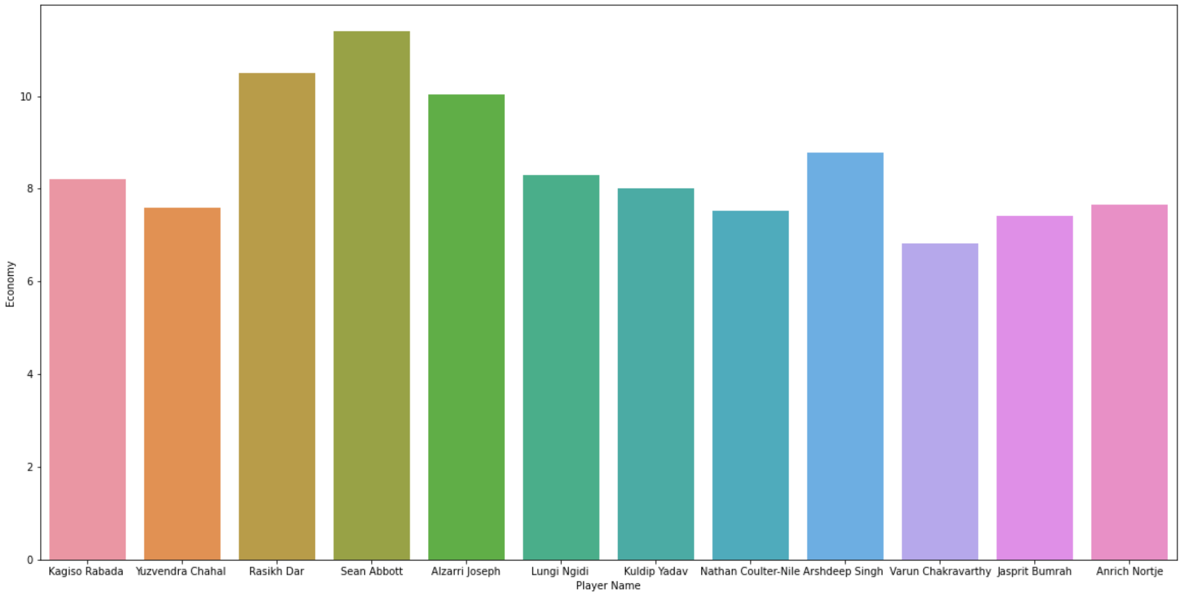


In [48]:

```
# the plot shows Economy of each of the two bowlers

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Economy', data= top_bowlers)
```

Out[48]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Economy'>
```

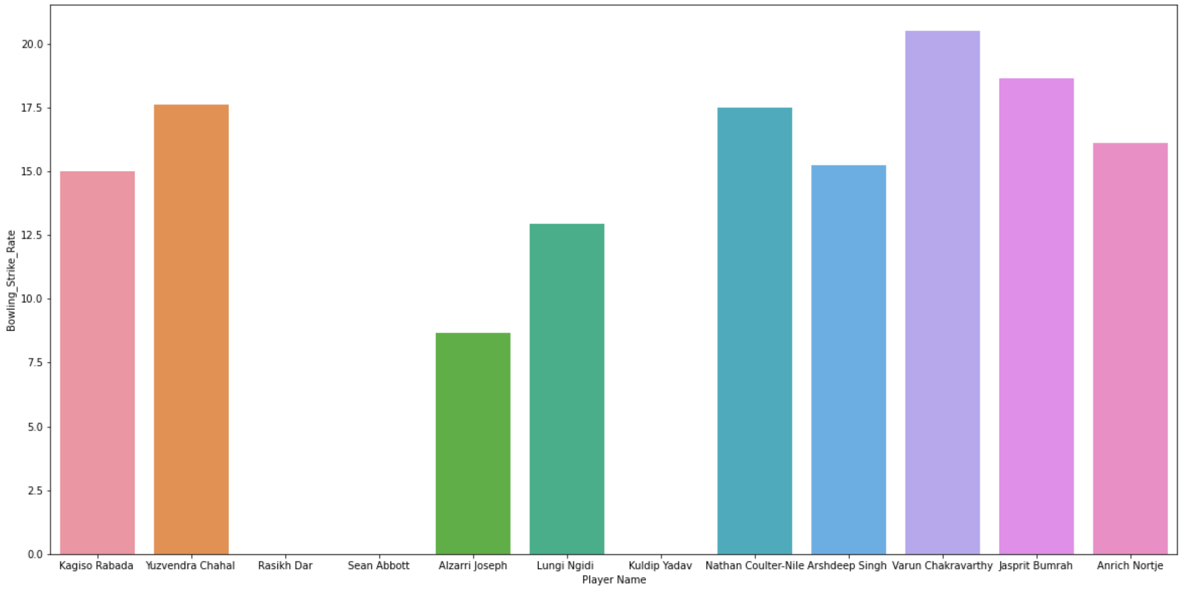


In [49]:

```
# the plot shows bowlingstrike of each of the two bowlers

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Bowling_Strike_Rate', data= top_bowlers)
```

Out[49]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Bowling_Strike_Rate'>
```

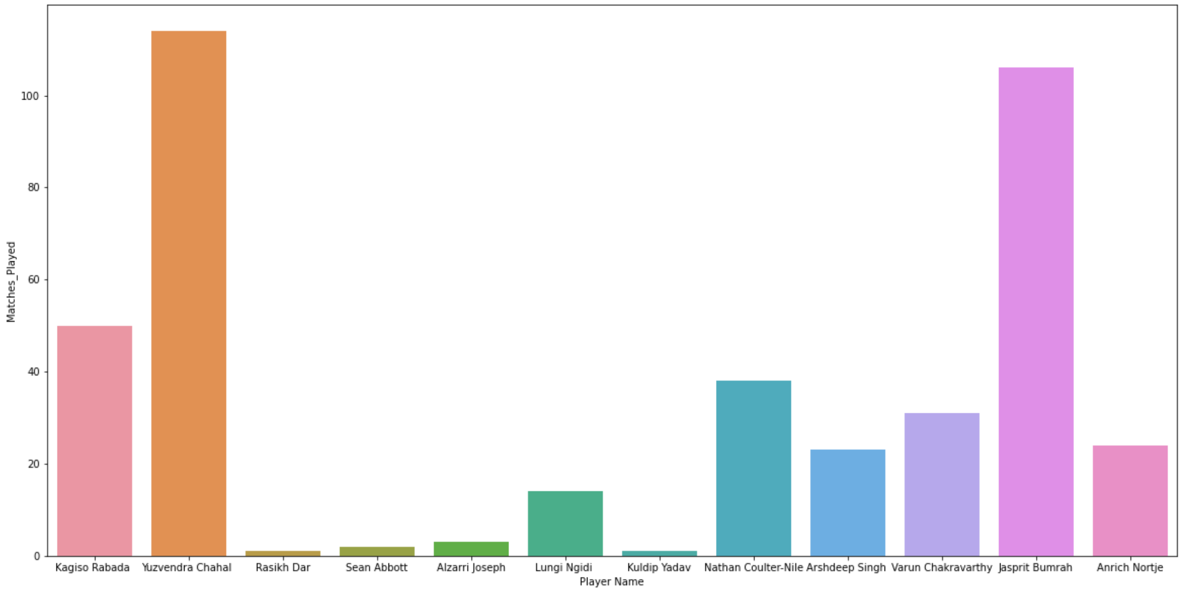


In [50]:

```
# the plot shows matches played of each of the two bowlers

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Matches_Played', data= top_bowlers)
```

Out[50]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Matches_Played'>
```

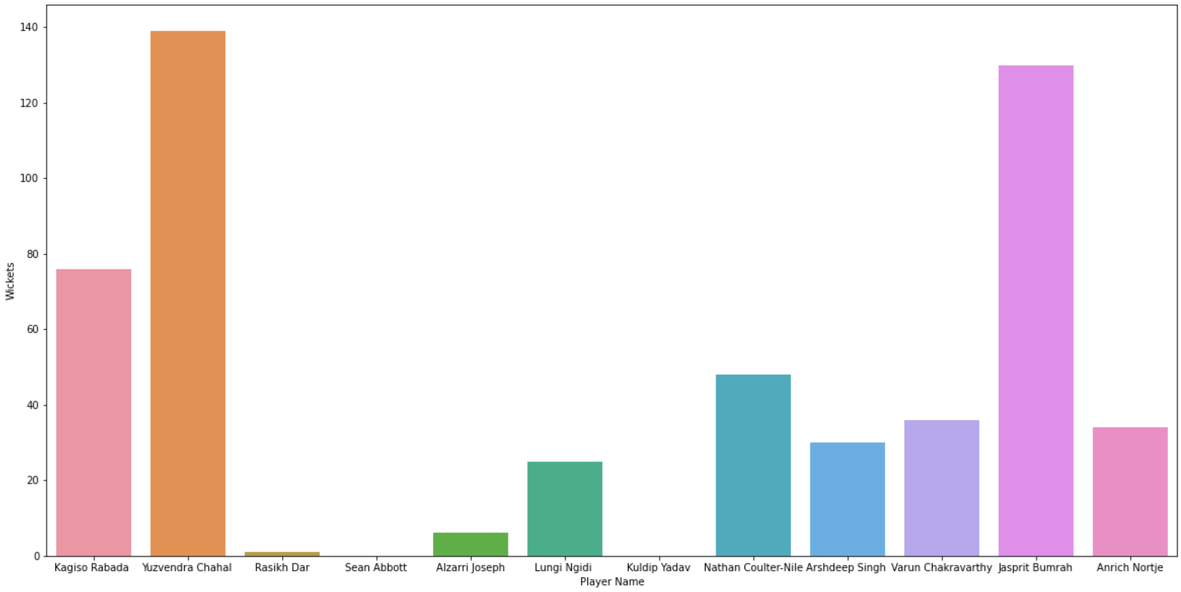


In [51]:

```
# the plot shows wickets of each of the two bowlers

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Wickets', data= top_bowlers)
```

Out[51]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Wickets'>
```

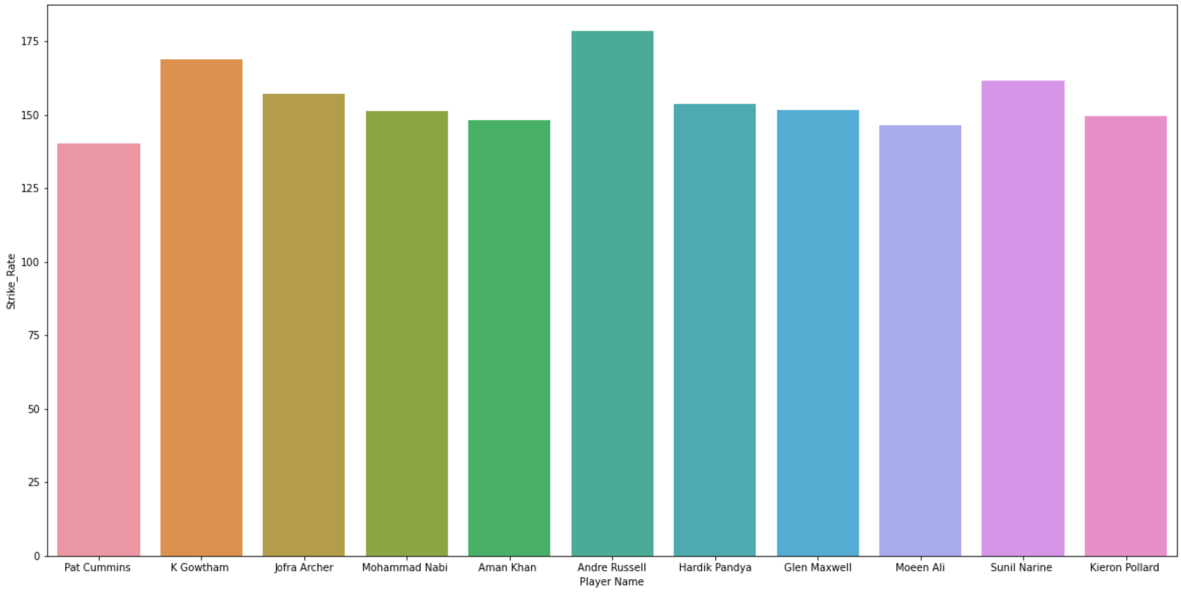


In [52]:

```
# Visualization  for the All rounders Data
# the plot shows Strike Rate of each of the two bowlers

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Strike_Rate', data= top_allrounders)
```

Out[52]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Strike_Rate'>
```

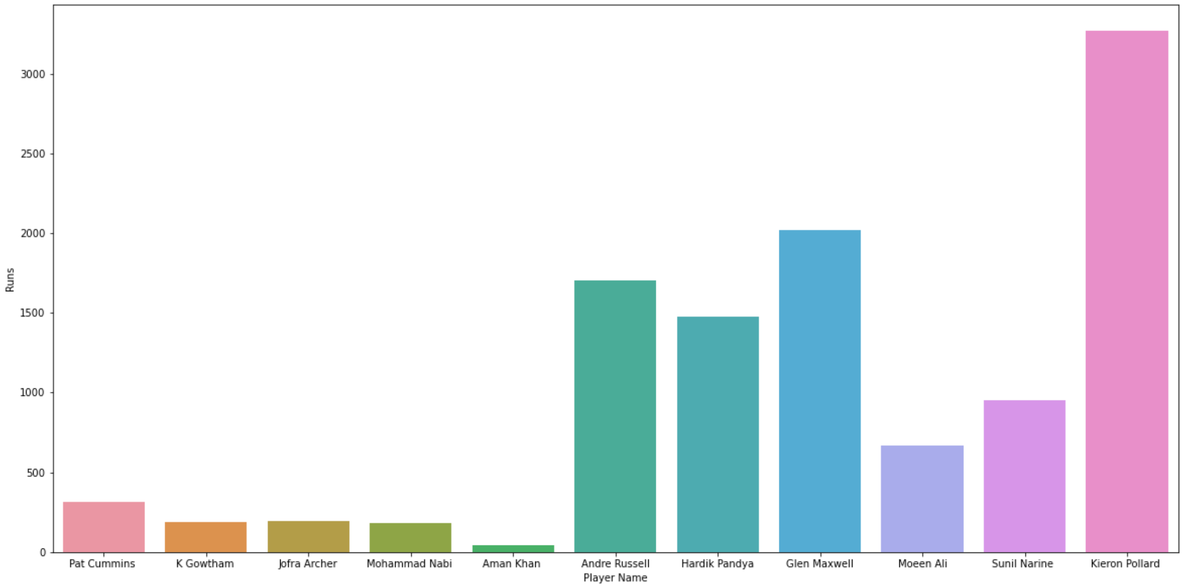


In [53]:

```

# the plot shows bowling average of each of the two bowlers

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Runs', data= top_allrounders)
```

Out[53]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Runs'>
```

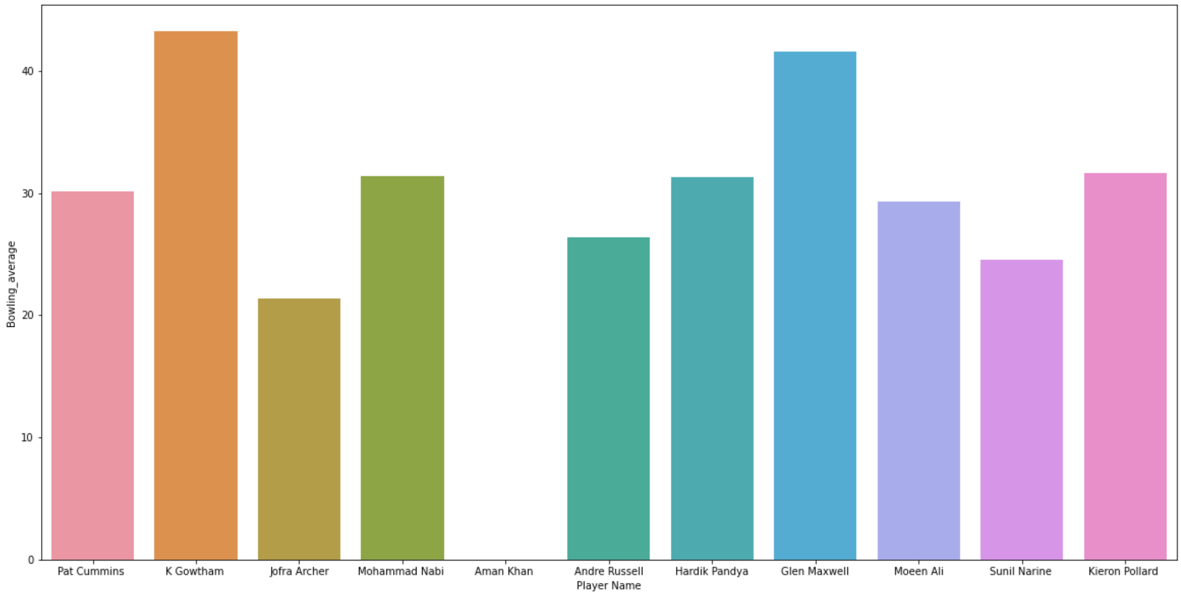


In [54]:

```
# the plot shows bowling average of the top Allrounders

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Bowling_average', data= top_allrounders)
```

Out[54]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Bowling_average'>
```

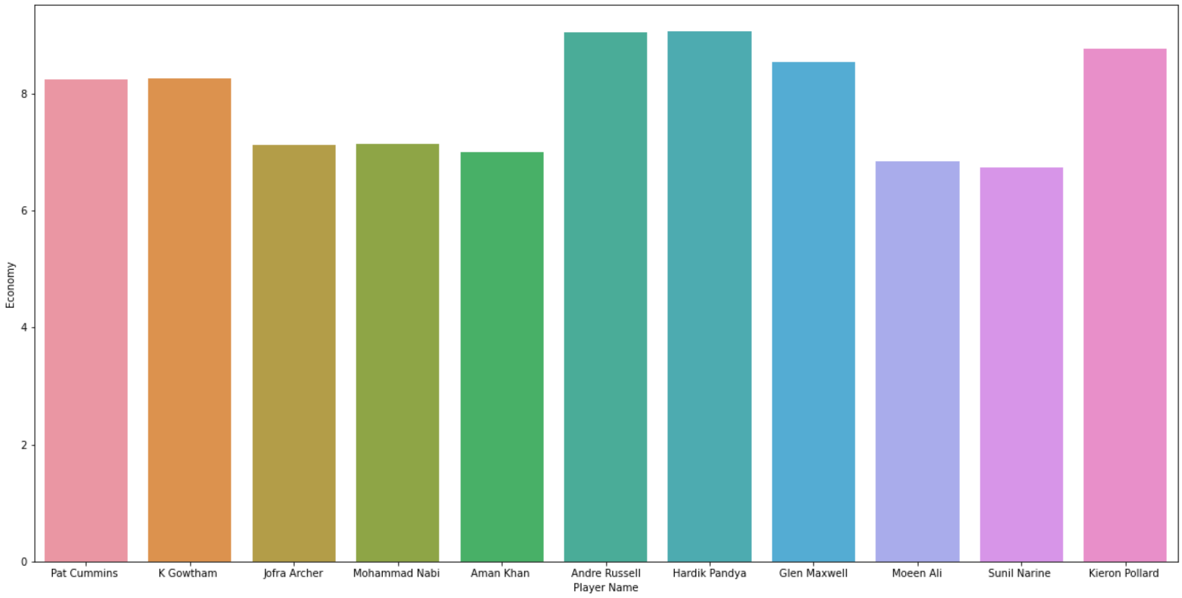


In [55]:

```
# the plot shows Economy of the top Allrounders

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Economy', data= top_allrounders)
```

Out[55]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Economy'>
```

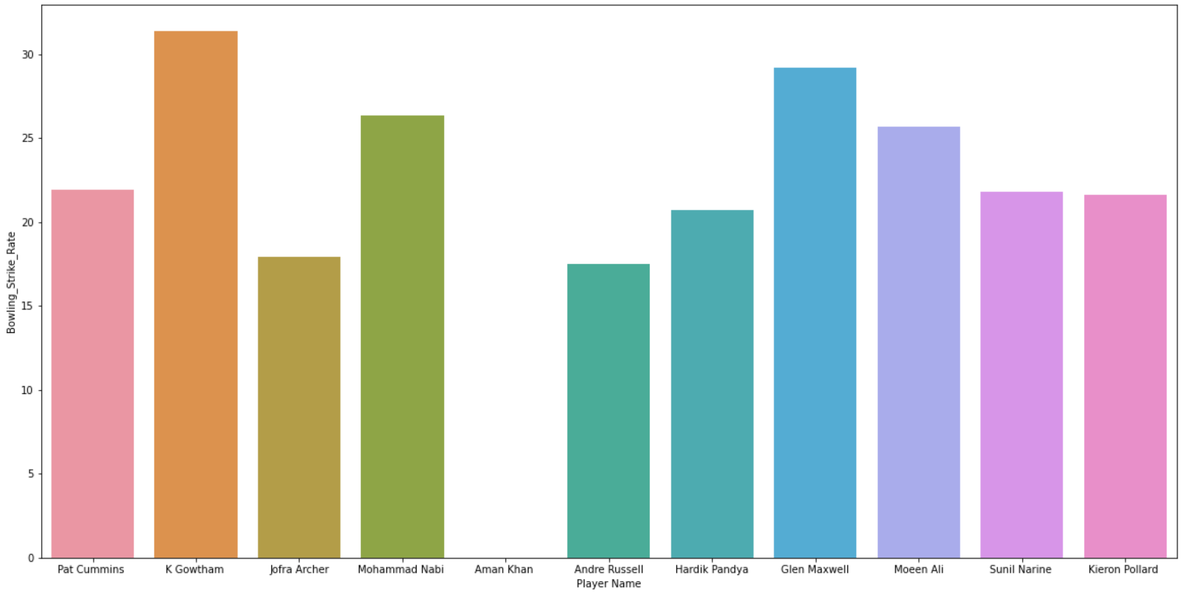


In [56]:

```
# the plot shows bowling Strike rate of the top Allrounders

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Bowling_Strike_Rate', data= top_allrounders)
```

Out[56]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Bowling_Strike_Rate'>
```

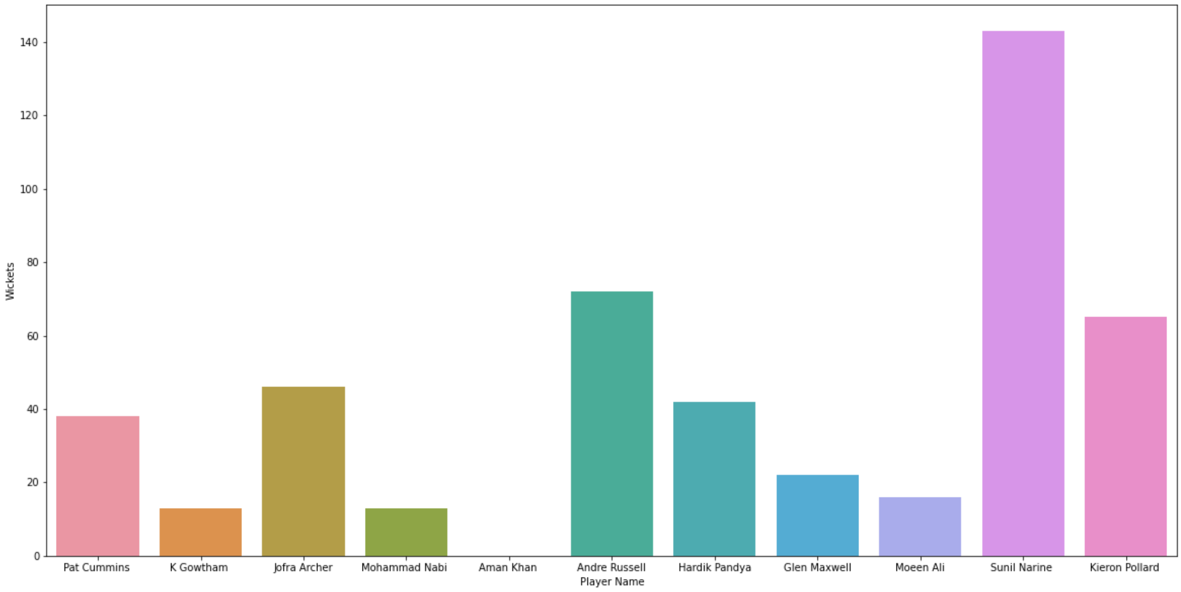


In [57]:

```
# the plot shows Wickets of the top Allrounders

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Wickets', data= top_allrounders)
```

Out[57]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Wickets'>
```

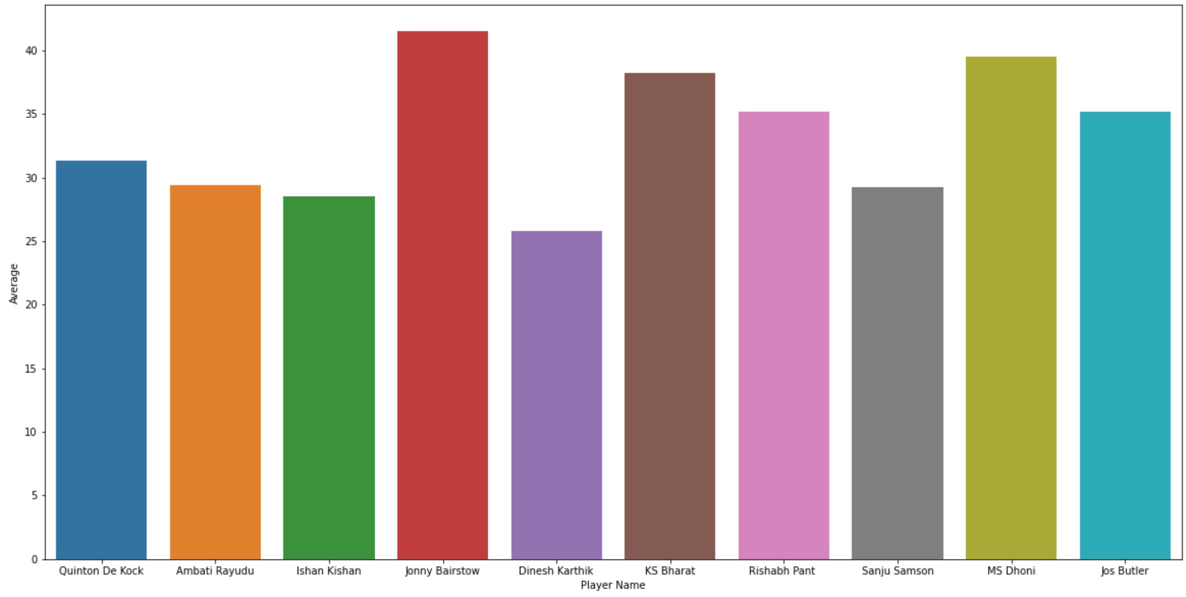


In [58]:

```
# Visualization  for the Keepers
# the plot shows Average of each of the of top keeper

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Average', data= top_keepers)
```

Out[58]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Average'>
```

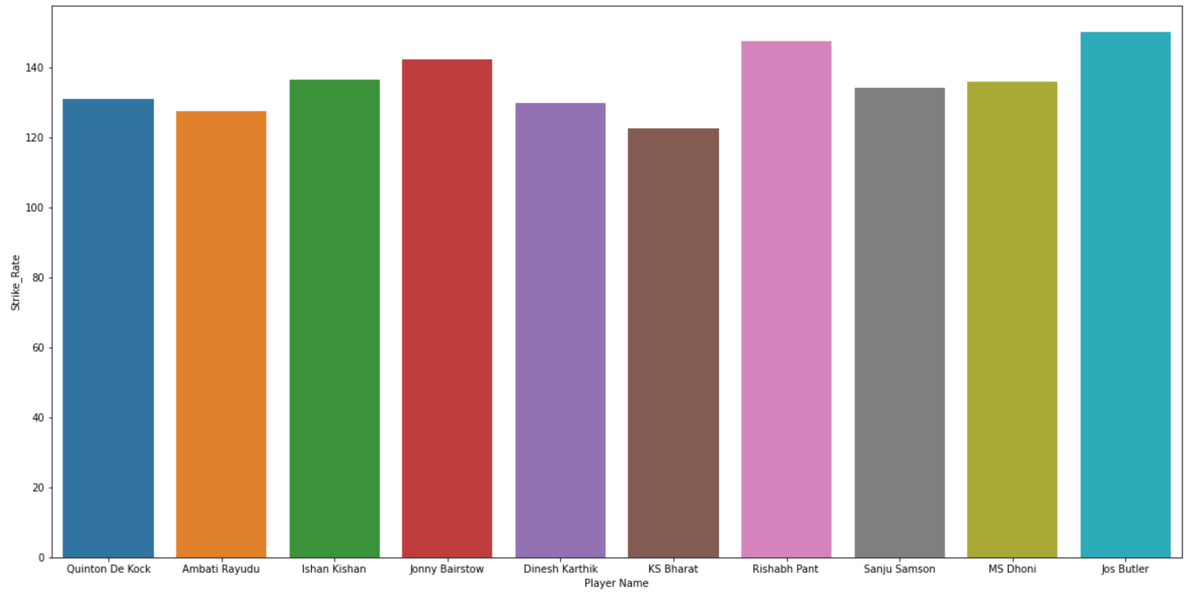


In [59]:

```

# the plot shows Strike Rate of each of the two bowlers

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Strike_Rate', data= top_keepers)
```

Out[59]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Strike_Rate'>
```

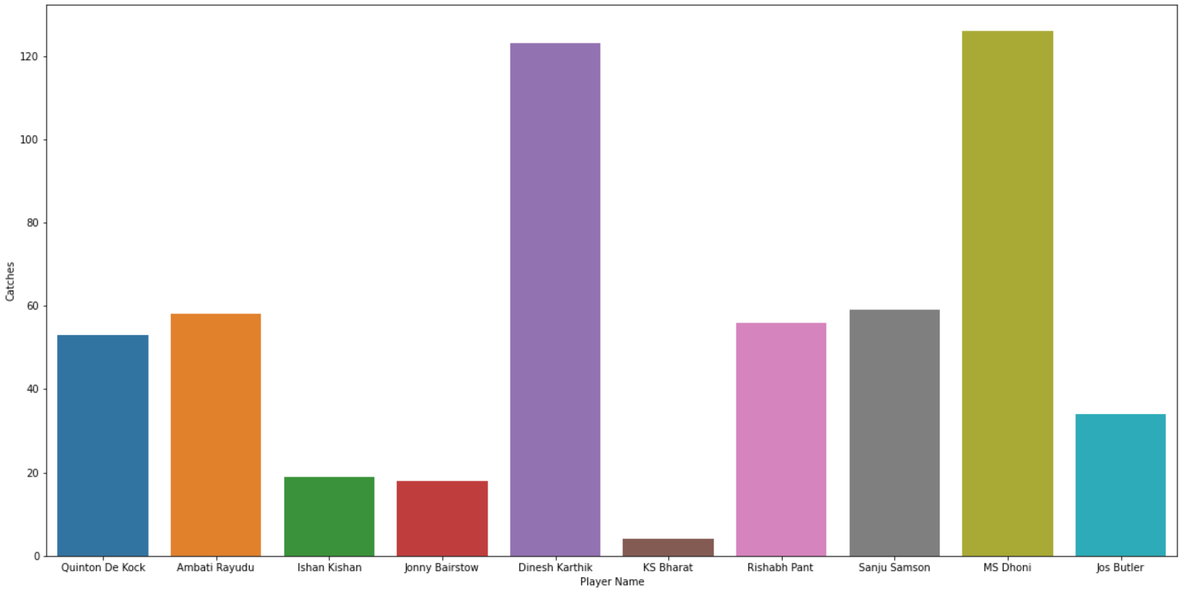


In [60]:

```

# the plot shows Strike Rate of each of the two bowlers

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Catches', data= top_keepers)
```

Out[60]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Catches'>
```

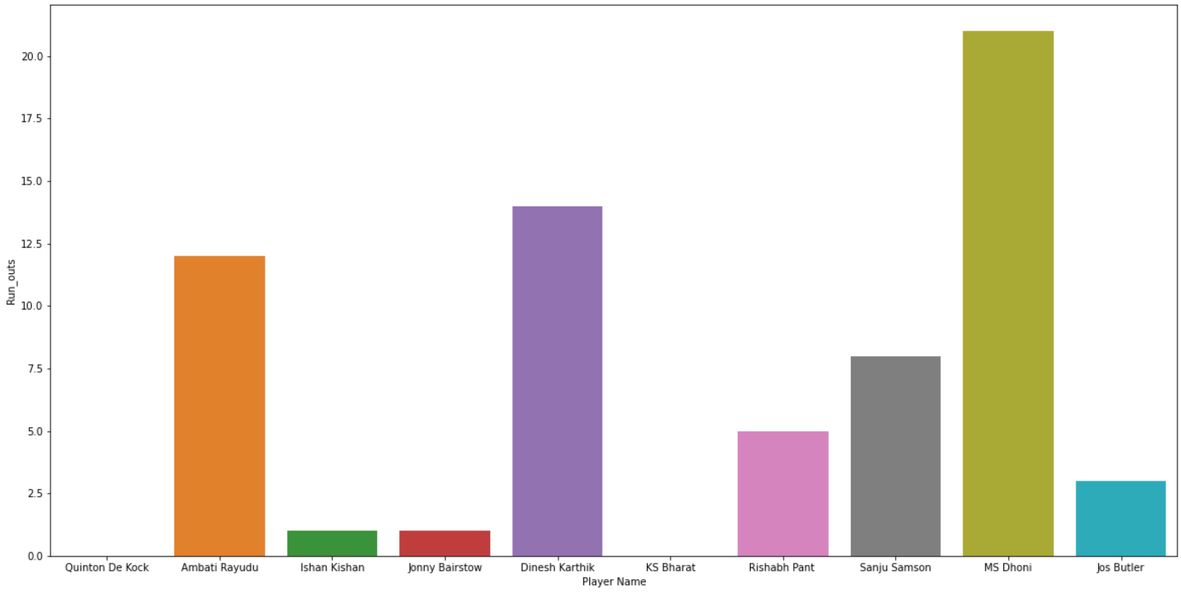


In [61]:

```

# the plot shows Runouts of each of the two bowlers

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Run_outs', data= top_keepers)
```

Out[61]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Run_outs'>
```

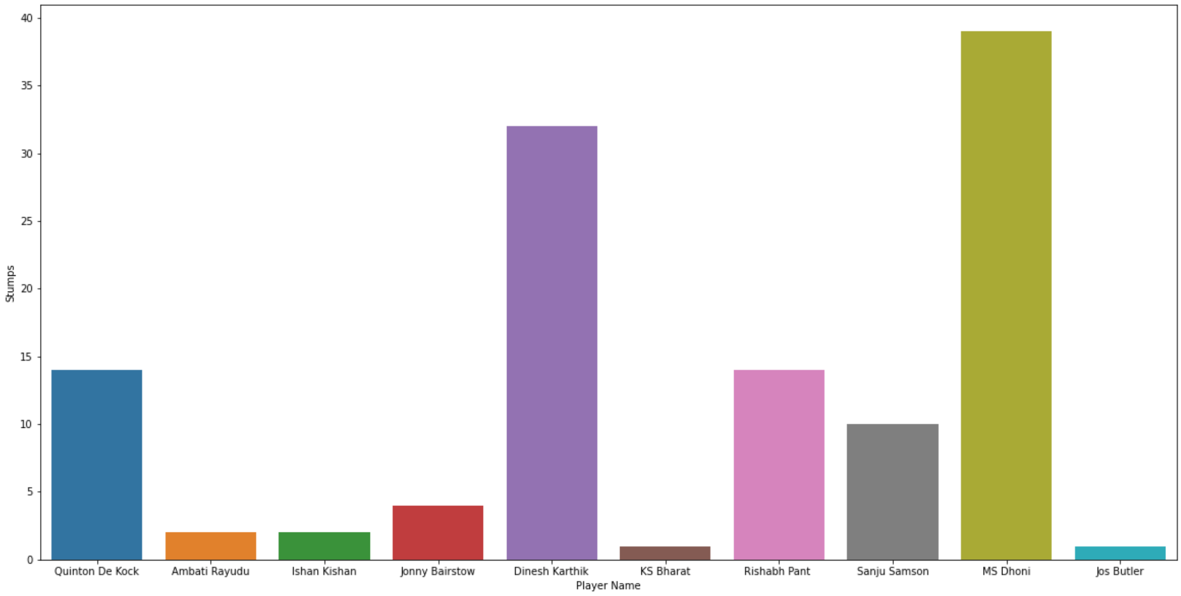


In [62]:

```

# the plot shows Stumps done by the keepers

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Stumps', data= top_keepers)
```

Out[62]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Stumps'>
```

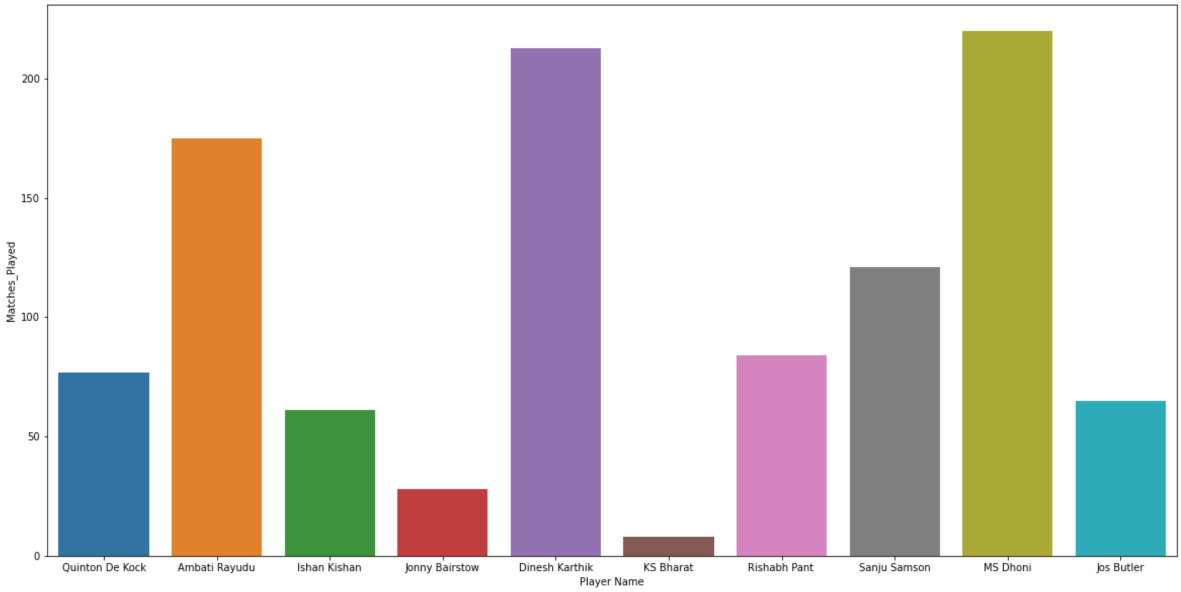


In [63]:

```

# the plot showsMatches played by the top keepers

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name',y='Matches_Played', data= top_keepers)
```

Out[63]:

```
<AxesSubplot:xlabel='Player Name', ylabel='Matches_Played'>
```



Forming our best 11 for campaign based above analysis

1. We will consider the number of players from each category that the t20 world cup wining and the last year's IPL wining team played in their final matches.

2. The Australia squad consisted of -3batters,3Allrounders,4bowlers with 1 spin option and 1 wicket keeper.

3. The chennai Squad consisted of -4batters,3Allrounders,3Bowler and 1 Wicket keeper.

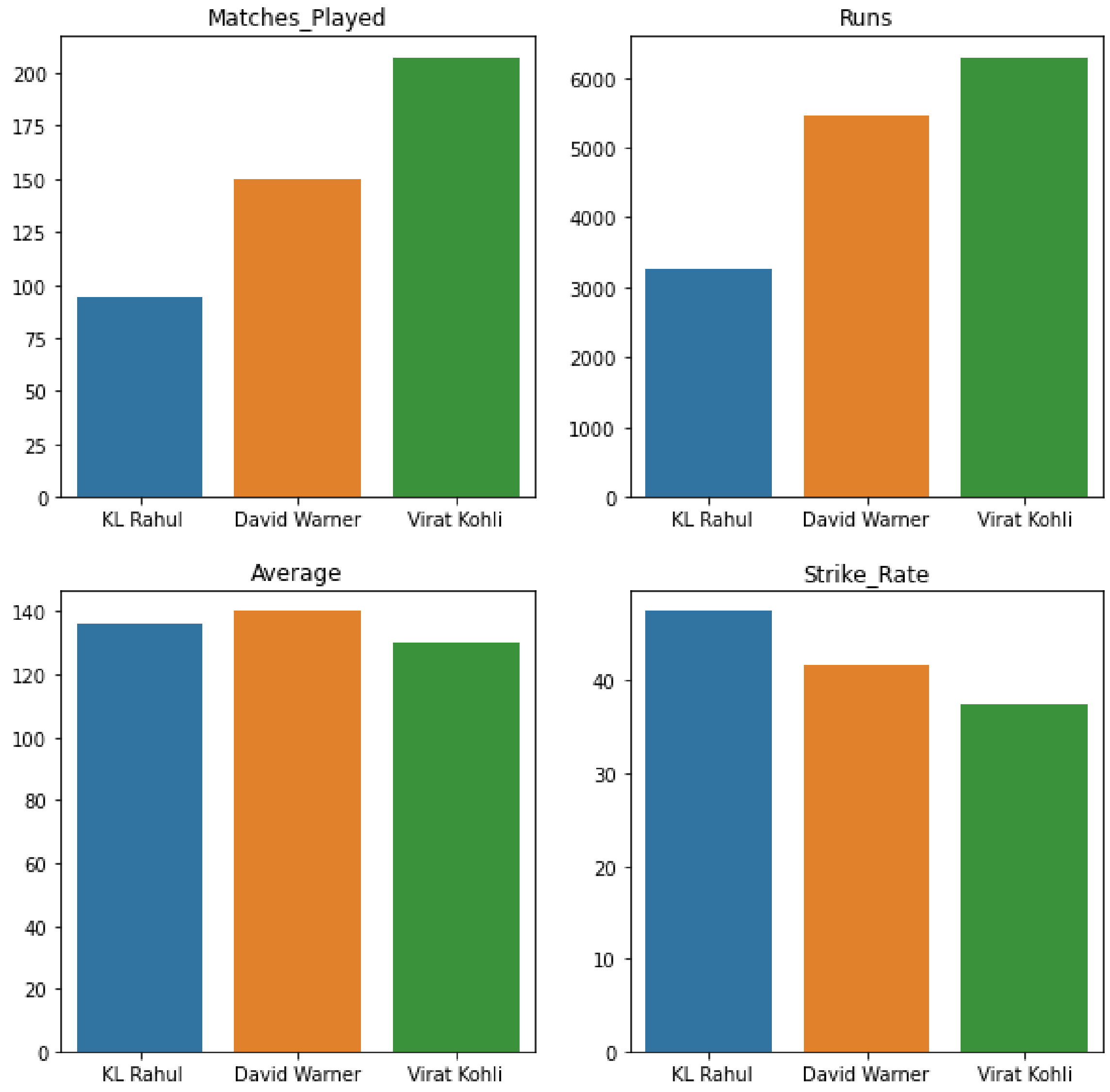
4. For our final analysis we will consider the ratio of players in the best 11 as follow:
   A. 3Batters
   B. 3Allrounders
   C. 4Bowlers with 2Spin Options
   D. 1Wickets Keeper

In [64]:

```
# batters for the final 11 -KL rahul,Virat kohli, David Warner
#here,we arestroring the values of each player in a separate dataframe to use for displ

top_batters.reset_index(drop=True)
matches_values = [top_batters.iloc[6]['Matches_Played'],top_batters.iloc[2]['Matches_Pl
run_values = [top_batters.iloc[6]['Runs'],top_batters.iloc[2]['Runs'],top_batters.iloc[
average_values = [top_batters.iloc[6]['Average'],top_batters.iloc[2]['Average'],top_bat
Strike_rate_values = [top_batters.iloc[6]['Strike_Rate'],top_batters.iloc[2]['Strike_Ra
labels = [' KL Rahul','David Warner','Virat Kohli']

fig,axes = plt.subplots(2,2 ,figsize=(10,10))
axes[0][0].set_title("Matches_Played")
axes[0][1].set_title("Runs")
axes[1][0].set_title("Average")
axes[1][1].set_title("Strike_Rate")

sns.barplot(x=labels,y=matches_values, ax=axes[0][0])
sns.barplot(x=labels,y=run_values, ax=axes[0][1])
sns.barplot(x=labels,y=Strike_rate_values , ax=axes[1][0])
sns.barplot(x=labels,y=average_values, ax=axes[1][1])


```

Out[64]:

```
<AxesSubplot:title={'center':'Strike_Rate'}>
```

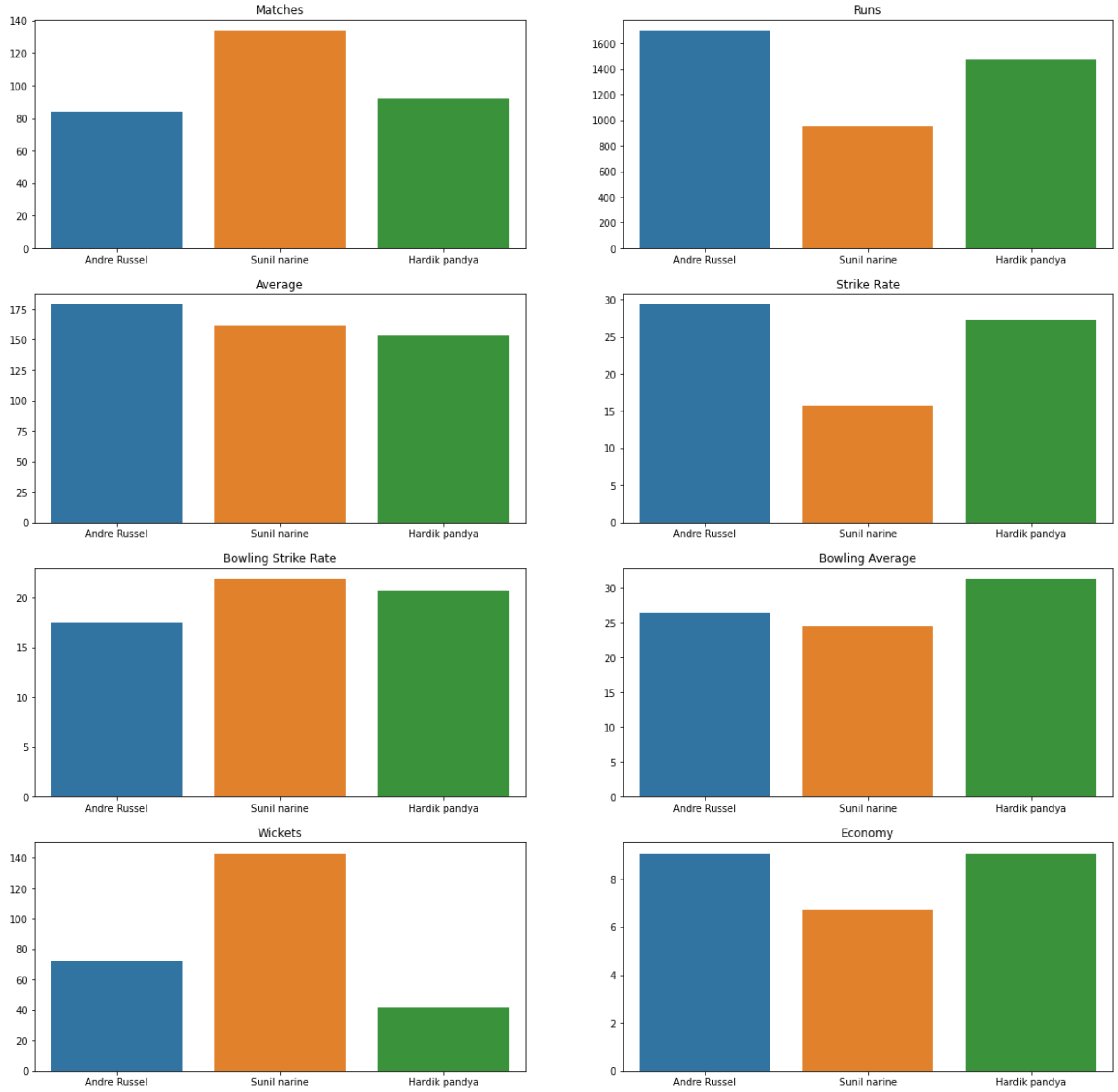

Matches_Played
200
175
150
125
100
75
50
25
0
KL Rahul
David Warner
Virat Kohli
Runs
6000
5000
4000
3000
2000
1000
0
KL Rahul
David Warner
Virat Kohli
Average
140
120
100
80
60
40
20
0
KL Rahul
David Warner
Virat Kohli
Strike_Rate
40
30
20
10
0
KL Rahul
David Warner
Virat Kohli

In [65]:

```
top_allrounders.reset_index(drop=True)
matches_values = [top_allrounders.iloc[5]['Matches_Played'],top_allrounders.iloc[9]['Ma
run_values = [top_allrounders.iloc[5]['Runs'],top_allrounders.iloc[9]['Runs'],top_allro
average_values = [top_allrounders.iloc[5]['Average'],top_allrounders.iloc[9]['Average']
Strike_rate_values = [top_allrounders.iloc[5]['Strike_Rate'],top_allrounders.iloc[9]['S
bowling_Strike_rate_values = [top_allrounders.iloc[5]['Bowling_Strike_Rate'],top_allrou
bowling_average_values = [top_allrounders.iloc[5]['Bowling_average'],top_allrounders.il
wickets_values = [top_allrounders.iloc[5]['Wickets'],top_allrounders.iloc[9]['Wickets']
economy_values = [top_allrounders.iloc[5]['Economy'],top_allrounders.iloc[9]['Economy']

labels = ['Andre Russel','Sunil narine','Hardik pandya']

fig,axes = plt.subplots(4,2 ,figsize=(20,20))
axes[0][0].set_title("Matches")
axes[0][1].set_title("Runs")
axes[1][0].set_title("Average")
axes[1][1].set_title("Strike Rate")
axes[2][0].set_title("Bowling Strike Rate")
axes[2][1].set_title("Bowling Average")
axes[3][0].set_title("Wickets")
axes[3][1].set_title("Economy")

sns.barplot(x=labels,y=matches_values, ax=axes[0][0])
sns.barplot(x=labels,y=run_values, ax=axes[0][1])
sns.barplot(x=labels,y=Strike_rate_values , ax=axes[1][0])
sns.barplot(x=labels,y=average_values, ax=axes[1][1])
sns.barplot(x=labels,y=bowling_Strike_rate_values, ax=axes[2][0])
sns.barplot(x=labels,y=bowling_average_values, ax=axes[2][1])
sns.barplot(x=labels,y=wickets_values, ax=axes[3][0])
sns.barplot(x=labels,y=economy_values, ax=axes[3][1])

```

Out[65]:

```
<AxesSubplot:title={'center':'Economy'}>
```

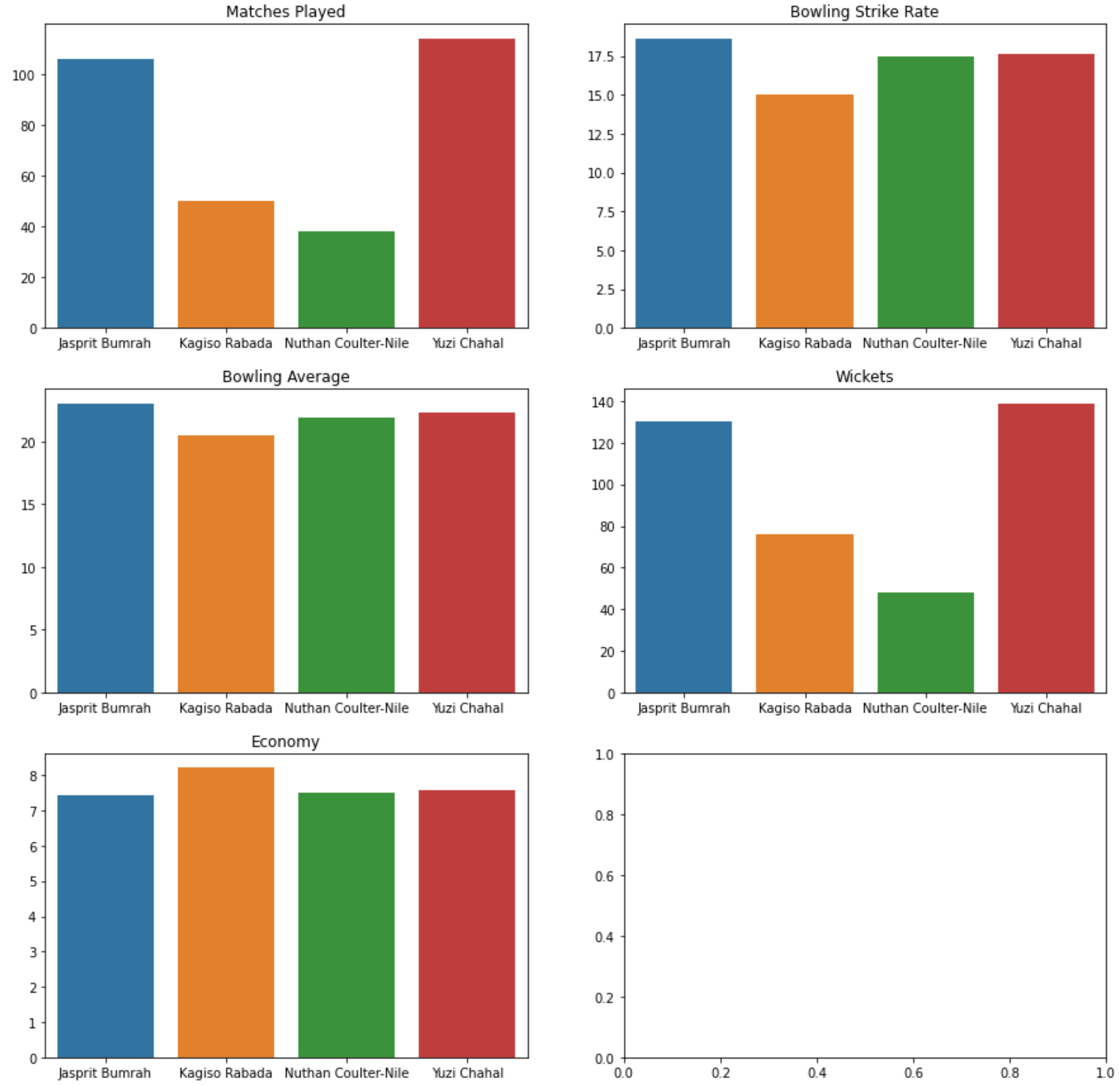

Matches
0
20
40
60
80
100
120
140
Andre Russel
Sunil narine
Hardik pandya
Runs
0
200
400
600
800
1000
1200
1400
1600
Andre Russel
Sunil narine
Hardik pandya
Average
0
25
50
75
100
125
150
175
Andre Russel
Sunil narine
Hardik pandya
Strike Rate
0
5
10
15
20
25
30
Andre Russel
Sunil narine
Hardik pandya
Bowling Strike Rate
0
5
10
15
20
Andre Russel
Sunil narine
Hardik pandya
Bowling Average
0
5
10
15
20
25
30
Andre Russel
Sunil narine
Hardik pandya
Wickets
0
20
40
60
80
100
120
140
Andre Russel
Sunil narine
Hardik pandya
Economy
0
2
4
6
8
Andre Russel
Sunil narine
Hardik pandya

In [66]:

```
top_bowlers.reset_index(drop=True)
matches_values = [top_bowlers.iloc[10]['Matches_Played'],top_bowlers.iloc[0]['Matches_P

bowling_Strike_rate_values = [top_bowlers.iloc[10]['Bowling_Strike_Rate'],top_bowlers.i
bowling_average_values = [top_bowlers.iloc[10]['Bowling_average'],top_bowlers.iloc[0]['
wickets_values = [top_bowlers.iloc[10]['Wickets'],top_bowlers.iloc[0]['Wickets'],top_bo
economy_values = [top_bowlers.iloc[10]['Economy'],top_bowlers.iloc[0]['Economy'],top_bo

labels = ['Jasprit Bumrah','Kagiso Rabada','Nuthan Coulter-Nile','Yuzi Chahal']

fig,axes = plt.subplots(3,2 ,figsize=(15,15))
axes[0][0].set_title("Matches Played")
axes[0][1].set_title("Bowling Strike Rate")
axes[1][0].set_title("Bowling Average")
axes[1][1].set_title("Wickets")
axes[2][0].set_title("Economy")

sns.barplot(x=labels,y=matches_values, ax=axes[0][0])
sns.barplot(x=labels,y=bowling_Strike_rate_values, ax=axes[0][1])
sns.barplot(x=labels,y=bowling_average_values, ax=axes[1][0])
sns.barplot(x=labels,y=wickets_values, ax=axes[1][1])
sns.barplot(x=labels,y=economy_values, ax=axes[2][0])
```

Out[66]:

```
<AxesSubplot:title={'center':'Economy'}>
```

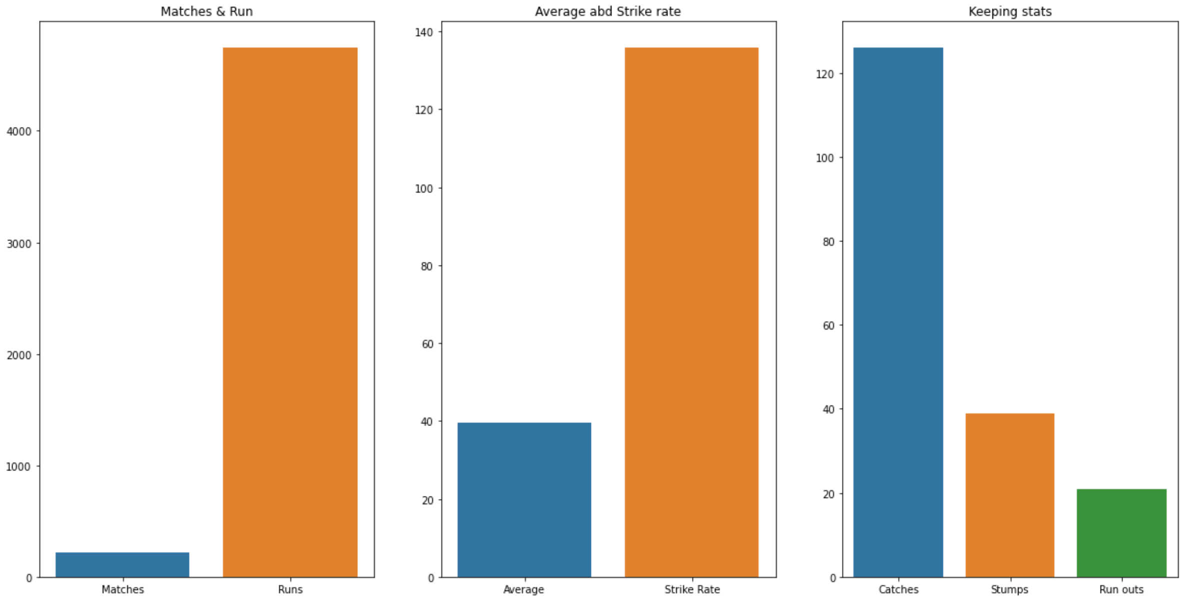

Matches Played
0
20
40
60
80
100
Jasprit Bumrah
Kagiso Rabada
Nuthan Coulter-Nile
Yuzi Chahal
Bowling Strike Rate
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Jasprit Bumrah
Kagiso Rabada
Nuthan Coulter-Nile
Yuzi Chahal
Bowling Average
0
5
10
15
20
Jasprit Bumrah
Kagiso Rabada
Nuthan Coulter-Nile
Yuzi Chahal
Wickets
0
20
40
60
80
100
120
140
Jasprit Bumrah
Kagiso Rabada
Nuthan Coulter-Nile
Yuzi Chahal
Economy
0
1
2
3
4
5
6
7
8
Jasprit Bumrah
Kagiso Rabada
Nuthan Coulter-Nile
Yuzi Chahal
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

In [67]:

```
 #wicket keeper for the final 11 - MS Dhoni
matches_values = [top_keepers.iloc[8]['Matches_Played'],top_keepers.iloc[8]['Runs']]
keeping_value = [top_keepers.iloc[8]['Average'],top_keepers.iloc[8]['Strike_Rate']]
average_values = [top_keepers.iloc[8]['Catches'],top_keepers.iloc[8]['Stumps'],top_keep

label1 = ['Matches','Runs']
label2 = ['Average','Strike Rate']
label3 = ['Catches','Stumps','Run outs']

fig,axes = plt.subplots(1,3 ,figsize=(20,10))
axes[0].set_title("Matches & Run")
axes[1].set_title("Average abd Strike rate")
axes[2].set_title("Keeping stats")


sns.barplot(x=label1,y=matches_values, ax=axes[0])
sns.barplot(x=label2,y=keeping_value, ax=axes[1])
sns.barplot(x=label3,y=average_values , ax=axes[2])


```

Out[67]:

```
<AxesSubplot:title={'center':'Keeping stats'}>
```



FINAL 11 FOR THE IPL CAMPAIGN

In [68]:

```
batter1 = top_batters.loc[(top_batters["Player Name"] == 'KL Rahul ')]
batter2 = top_batters.loc[(top_batters["Player Name"] == 'David Warner ')]
batter3 = top_batters.loc[(top_batters["Player Name"] == 'Virat Kohli')]

bowler1 = top_bowlers.loc[(top_bowlers["Player Name"] == 'Yuzvendra Chahal ')]
bowler2 = top_bowlers.loc[(top_bowlers["Player Name"] == 'Jasprit Bumrah')]
bowler3 = top_bowlers.loc[(top_bowlers["Player Name"] == 'Nathan Coulter-Nile')]
bowler4 = top_bowlers.loc[(top_bowlers["Player Name"] == 'Kagiso Rabada ')]

allrounder1 = top_allrounders.loc[(top_allrounders["Player Name"] == 'Andre Russel')]
allrounder2 = top_allrounders.loc[(top_allrounders["Player Name"] == 'Sunil Narine ')]
allrounder3 = top_allrounders.loc[(top_allrounders["Player Name"] == 'Hardik Pandya')]

keeper = top_keepers.loc[(top_keepers["Player Name"] == 'MS Dhoni')]
```

In [69]:

```
final = [batter1 ,batter2 ,batter3 ,allrounder1 , allrounder2, allrounder3 ,keeper,bowl
final_team = pd.concat(final)
final_team = final_team.drop(labels = ['Matches_Played','Runs','Average','Strike_Rate',
                                       ,'Bowling_Strike_Rate','Economy',
                                     'Catches','Run_outs','Stumps'],axis=1)
```

In [70]:

```
# showing the name players in finall 11 IPL Campaign
final_team.reset_index(drop=True)
```

Out[70]:

	Player Name	Team	Nationality
0	KL Rahul	Lucknow	Indian
1	David Warner	Delhi	Overseas
2	Virat Kohli	Bangalore	Indian
3	Sunil Narine	Kolkata	Overseas
4	Hardik Pandya	Gujarat	Indian
5	MS Dhoni	Chennai	Indian
6	Yuzvendra Chahal	Rajasthan	Indian
7	Jasprit Bumrah	Mumbai	Indian
8	Nathan Coulter-Nile	Rajasthan	Overseas
9	Kagiso Rabada	Punjab	Overseas

In [ ]:

```

```

In [ ]:

```

```