In [105]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [106]:
```python
df = pd.read_csv('matches.csv')
```

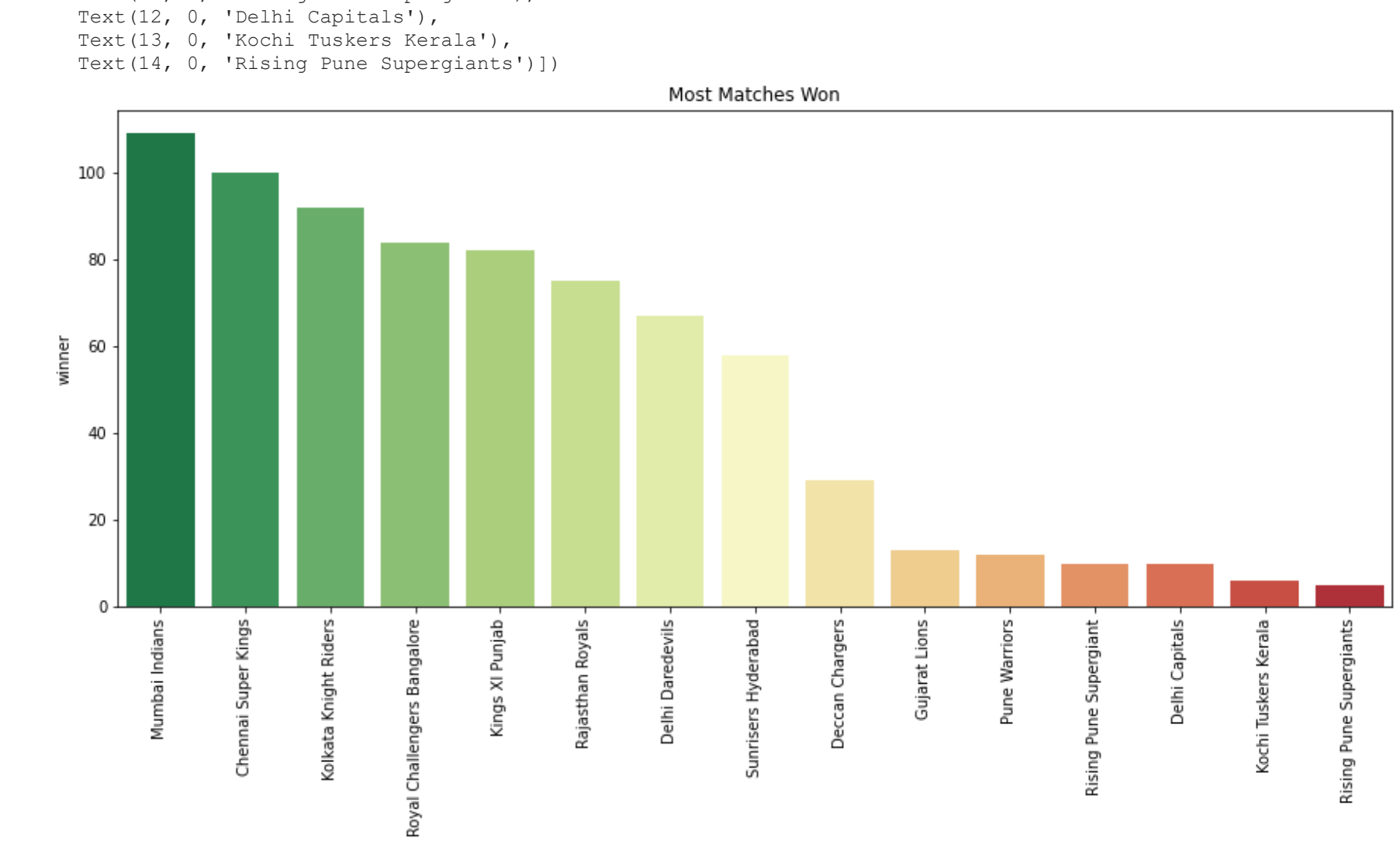
In [107]:
```python
df.head()
```

Out[107]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied	winner	win_by_runs
0	1	2017	Hyderabad	2017-04-05	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0	Sunrisers Hyderabad	35
1	2	2017	Pune	2017-04-06	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	0	Rising Pune Supergiant	0
2	3	2017	Rajkot	2017-04-07	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	0	Kolkata Knight Riders	0
3	4	2017	Indore	2017-04-08	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	normal	0	Kings XI Punjab	0
4	5	2017	Bangalore	2017-04-08	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	normal	0	Royal Challengers Bangalore	15

In [202]:
```python
plt.figure(figsize=(15,6))
win = df.winner.value_counts().reset_index()
sns.barplot(x=win['index'],y=win['winner'],palette='RdYlGn_r')
plt.title('Most Matches Won')
plt.xticks(rotation=90)
```

Out[202]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14]),
 [Text(0, 0, 'Mumbai Indians'),
  Text(1, 0, 'Chennai Super Kings'),
  Text(2, 0, 'Kolkata Knight Riders'),
  Text(3, 0, 'Royal Challengers Bangalore'),
  Text(4, 0, 'Kings XI Punjab'),
  Text(5, 0, 'Rajasthan Royals'),
  Text(6, 0, 'Delhi Daredevils'),
  Text(7, 0, 'Sunrisers Hyderabad'),
  Text(8, 0, 'Deccan Chargers'),
  Text(9, 0, 'Gujarat Lions'),
  Text(10, 0, 'Pune Warriors'),
  Text(11, 0, 'Rising Pune Supergiant'),
  Text(12, 0, 'Delhi Capitals'),
  Text(13, 0, 'Kochi Tuskers Kerala'),
  Text(14, 0, 'Rising Pune Supergiants')])
```



According to the chart above most successful teams are as follows

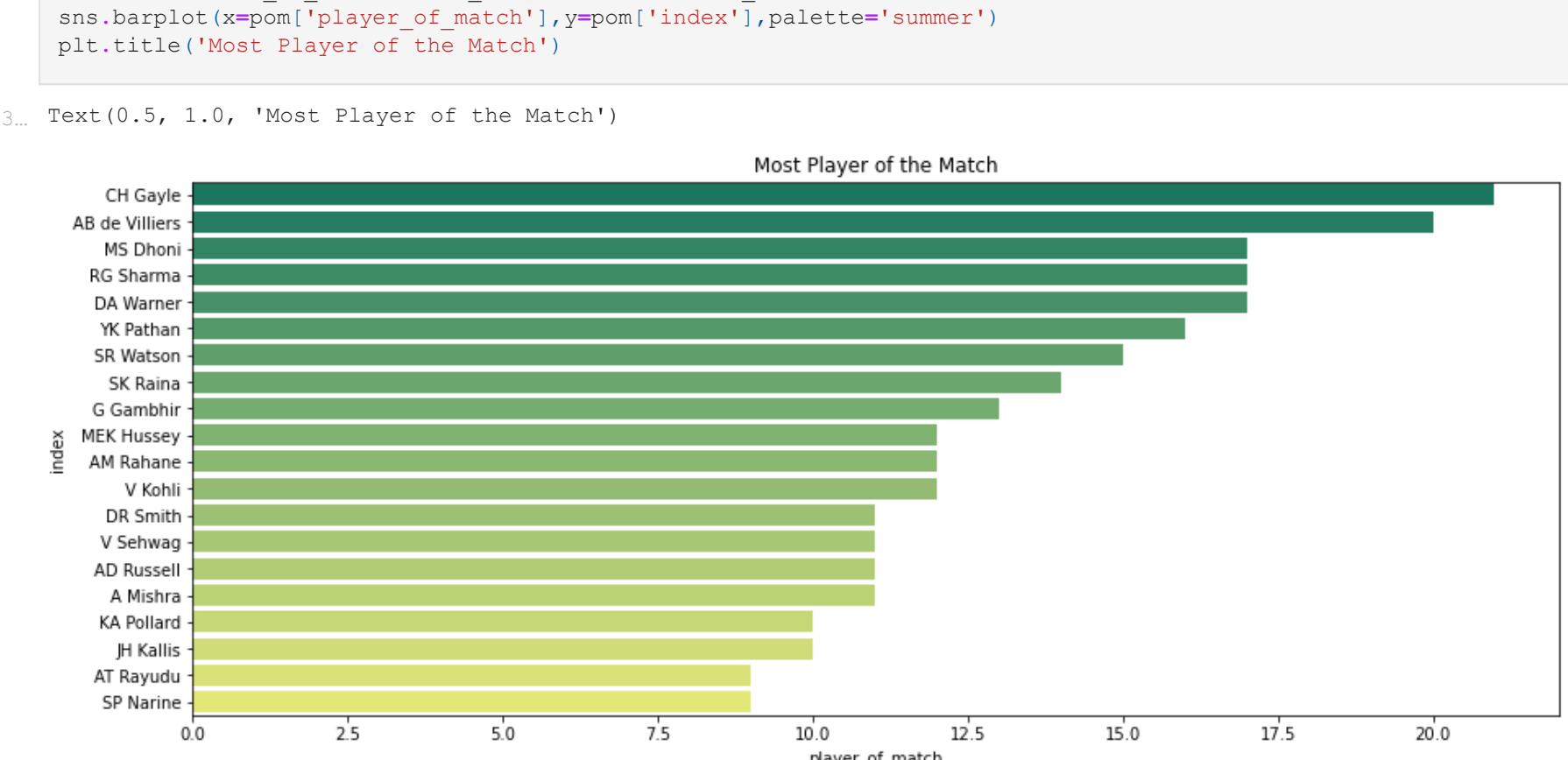
1) Mumbai Indians

2) Chennai Super Kings

3) Kolkata Knight Riders

In [203]:
```python
plt.figure(figsize=(15,6))
pom = df.player_of_match.value_counts()[:20].reset_index()
sns.barplot(x=pom['player_of_match'],y=pom['index'],palette='summer')
plt.title('Most Player of the Match')
```

Out[203]: Text(0.5, 1.0, 'Most Player of the Match')



Players with most man of the match for their outstand performances

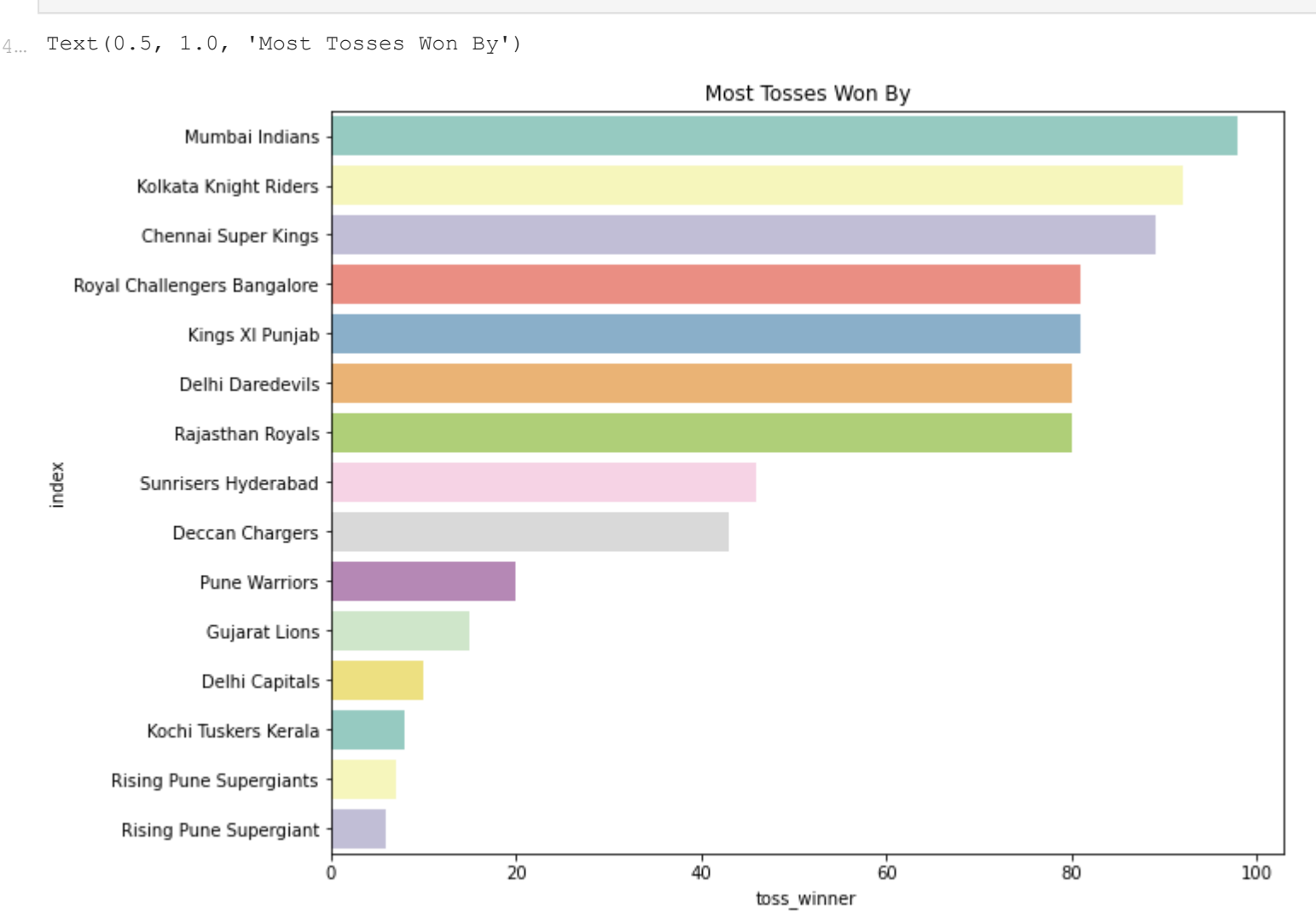
1) CH Gayle

2) AB de Villiers

3) MS Dhoni

In [204]:
```python
plt.figure(figsize=(10,8))
tos = df.toss_winner.value_counts().reset_index()
sns.barplot(x=tos['toss_winner'],y=tos['index'],palette='Set3')
plt.title('Most Tosses Won By')
```

Out[204]: Text(0.5, 1.0, 'Most Tosses Won By')



Team that have won the toss most numbers of time

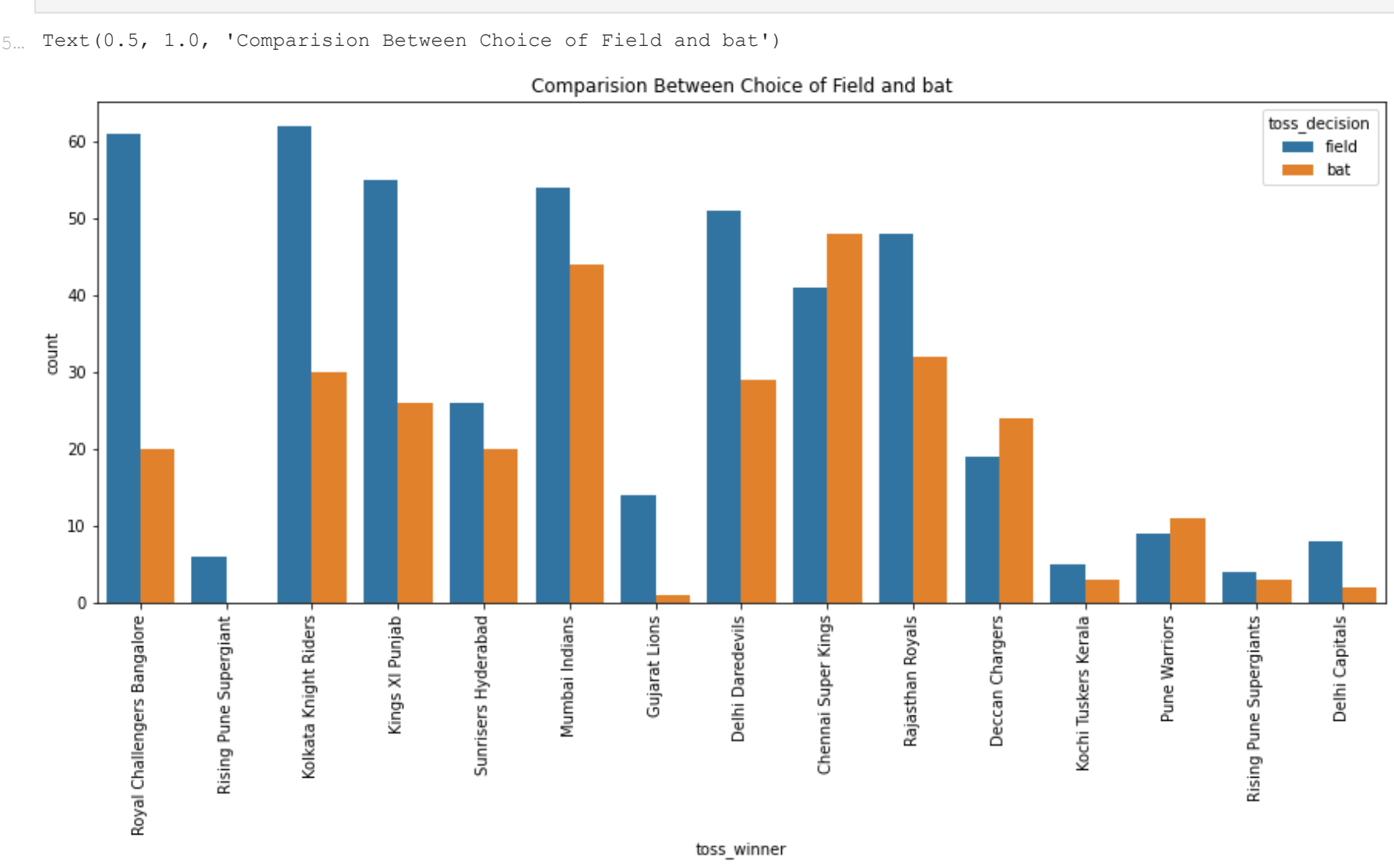
1) Mumbai Indians

2) Kolkata Knight Riders

3) Chennai Super Kings

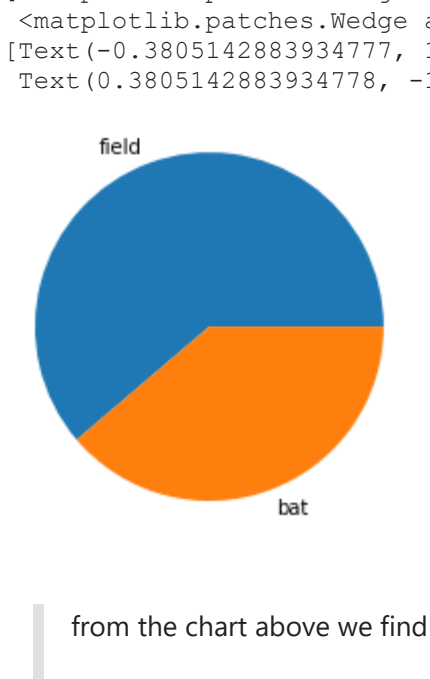
In [205]:
```python
plt.figure(figsize=(15,6))
sns.countplot(x=df['toss_winner'],hue=df['toss_decision'])
plt.xticks(rotation=90)
plt.title('Comparision Between Choice of Field and bat')
```

Out[205]: Text(0.5, 1.0, 'Comparision Between Choice of Field and bat')



In [112]:
```python
tos = df.toss_decision.value_counts().reset_index()
plt.pie(tos['toss_decision'],labels=tos['index'])
```

Out[112]:
```
([<matplotlib.patches.Wedge at 0x18628a42880>,
  <matplotlib.patches.Wedge at 0x18628a42d60>],
 [Text(-0.3805142883934777, 1.032089567977705, 'field'),
  Text(0.3805142883934778, -1.032089567977705, 'bat')])
```



> from the chart above we find that most of the teams tend to choose Field after winning the toss and try to chase the total
>
> Chennai super kings is the exception in this case the tend to choose to bat and the defend the total

In [113]:
```python
df1 = df[df['win_by_runs']!=0]
print("Number of times Match was won team defending the target",len(df1))
```

Number of times Match was won team defending the target 337

Wins While Defending = 337

In [114]:
```python
df2 = df[df['win_by_wickets']!=0]
print("Number of times match was won by the team chasing down the total",len(df2))
```

Number of times match was won by the team chasing down the total 406

Wins while chasing = 406

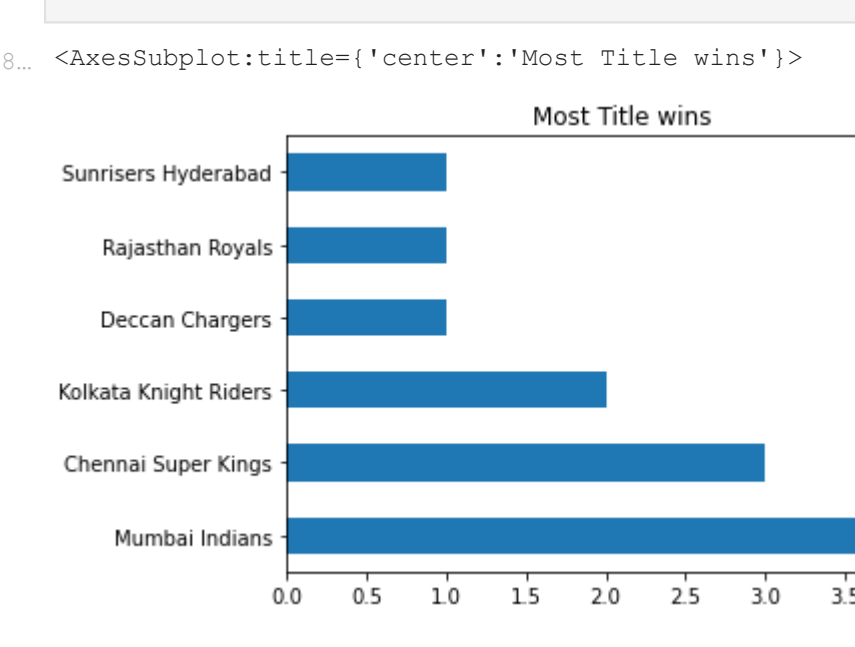
In [115]:
```python
print("Number of times match was tied",len(df)-len(df2)-len(df1))
```

Number of times match was tied 69

ties = 69

In [198]:
```python
season_winner = df.drop_duplicates(subset=['season'],keep='last')[['season', 'winner']].reset_index(drop=True)
season_winner.winner.value_counts().plot(kind='barh',orientation='horizontal',title='Most Title wins')
```

Out[198]: <AxesSubplot:title={'center':'Most Title wins'}>



Mumbai Indians is most successful team with **4 ipl titles** followed by **Chennai super Kings** with **3 titles**

Player Records

In [163]:
```python
dff = pd.read_csv('deliveries.csv')
```

In [164]:
```python
dff.head()
```

Out[164]:

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler	is_super_over	...	bye_runs	legbye_runs
0	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	1	DA Warner	S Dhawan	TS Mills	0	...	0	0
1	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	2	DA Warner	S Dhawan	TS Mills	0	...	0	0
2	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	3	DA Warner	S Dhawan	TS Mills	0	...	0	0
3	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	4	DA Warner	S Dhawan	TS Mills	0	...	0	0
4	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	5	DA Warner	S Dhawan	TS Mills	0	...	0	0

5 rows × 21 columns

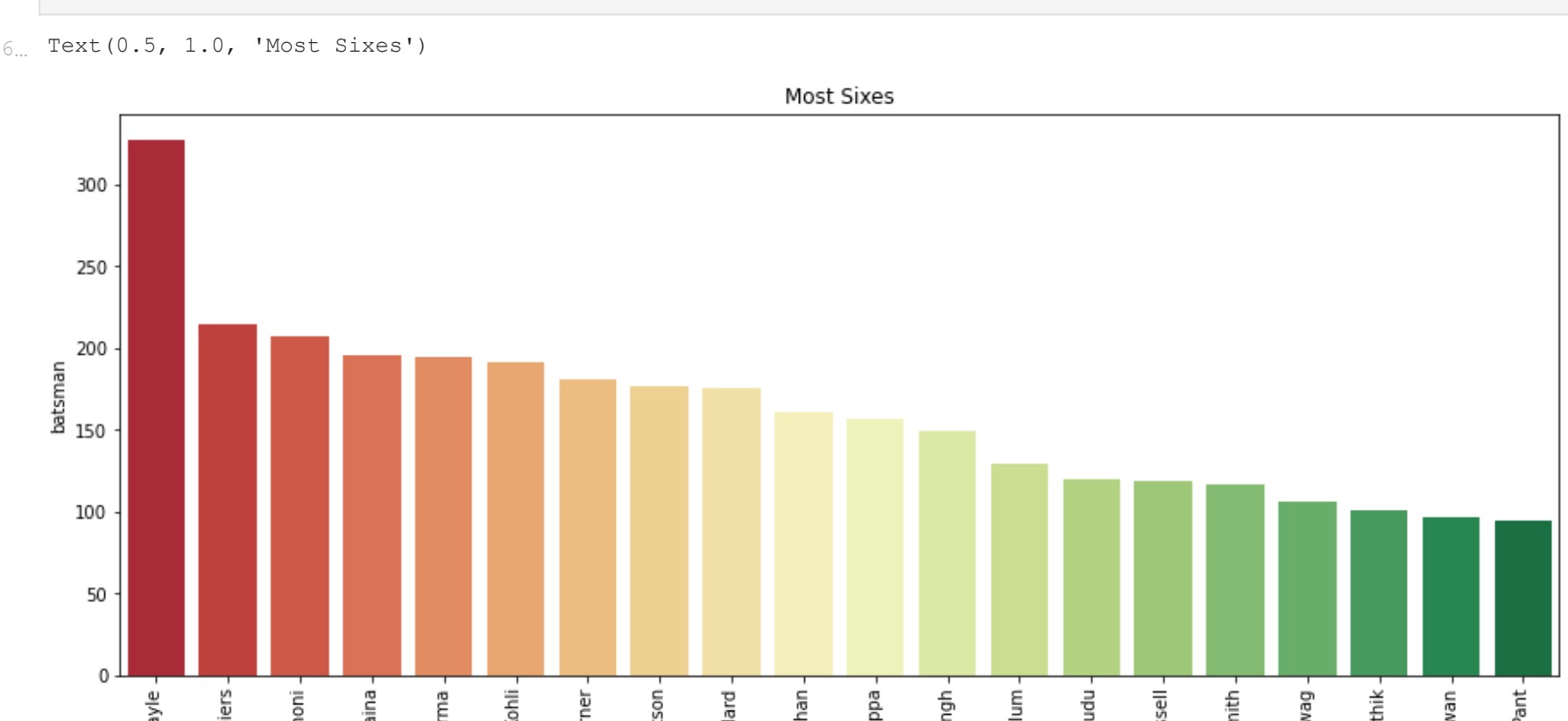
In [165]:
```python
dff.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 179078 entries, 0 to 179077
Data columns (total 21 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   match_id          179078 non-null  int64
 1   inning            179078 non-null  int64
 2   batting_team      179078 non-null  object
 3   bowling_team      179078 non-null  object
 4   over              179078 non-null  int64
 5   ball              179078 non-null  int64
 6   batsman           179078 non-null  object
 7   non_striker       179078 non-null  object
 8   bowler            179078 non-null  object
 9   is_super_over     179078 non-null  int64
 10  wide_runs         179078 non-null  int64
 11  bye_runs          179078 non-null  int64
 12  legbye_runs       179078 non-null  int64
 13  noball_runs       179078 non-null  int64
 14  penalty_runs      179078 non-null  int64
 15  batsman_runs      179078 non-null  int64
 16  extra_runs        179078 non-null  int64
 17  total_runs        179078 non-null  int64
 18  player_dismissed  8834 non-null    object
 19  dismissal_kind    8834 non-null    object
 20  fielder           6448 non-null    object
dtypes: int64(13), object(8)
memory usage: 28.7+ MB
```

Batting Records

In [206]:
```python
plt.figure(figsize=(15,6))
dff2 = dff[dff['batsman_runs']==6]
bat = dff2.batsman.value_counts()[:20].reset_index()
sns.barplot(x=bat['index'],y=bat['batsman'],palette='RdYlGn')
plt.xticks(rotation=90)
plt.title('Most Sixes')
```

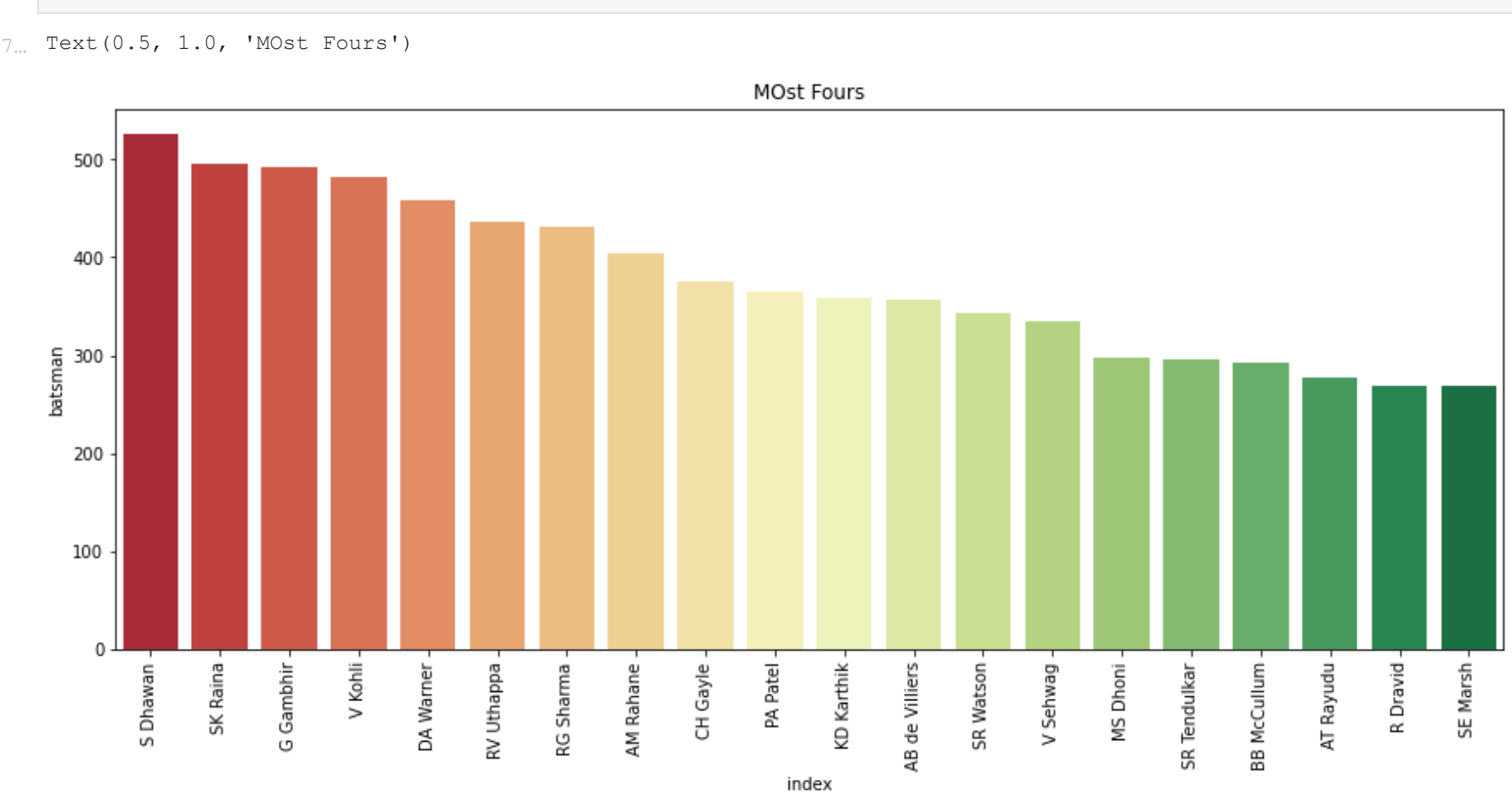
Out[206]: Text(0.5, 1.0, 'Most Sixes')



CH Gayle is most explosive batsman with over **300 6's** followed by **Ab de Villers** with over **200 6's** followed bt **MS Dhoni** with just a little more then **200 6's**

In [207]:
```python
plt.figure(figsize=(15,6))
dff2 = dff[dff['batsman_runs']==4]
bat = dff2.batsman.value_counts()[:20].reset_index()
sns.barplot(x=bat['index'],y=bat['batsman'],palette='RdYlGn')
plt.xticks(rotation=90)
plt.title('MOst Fours')
```

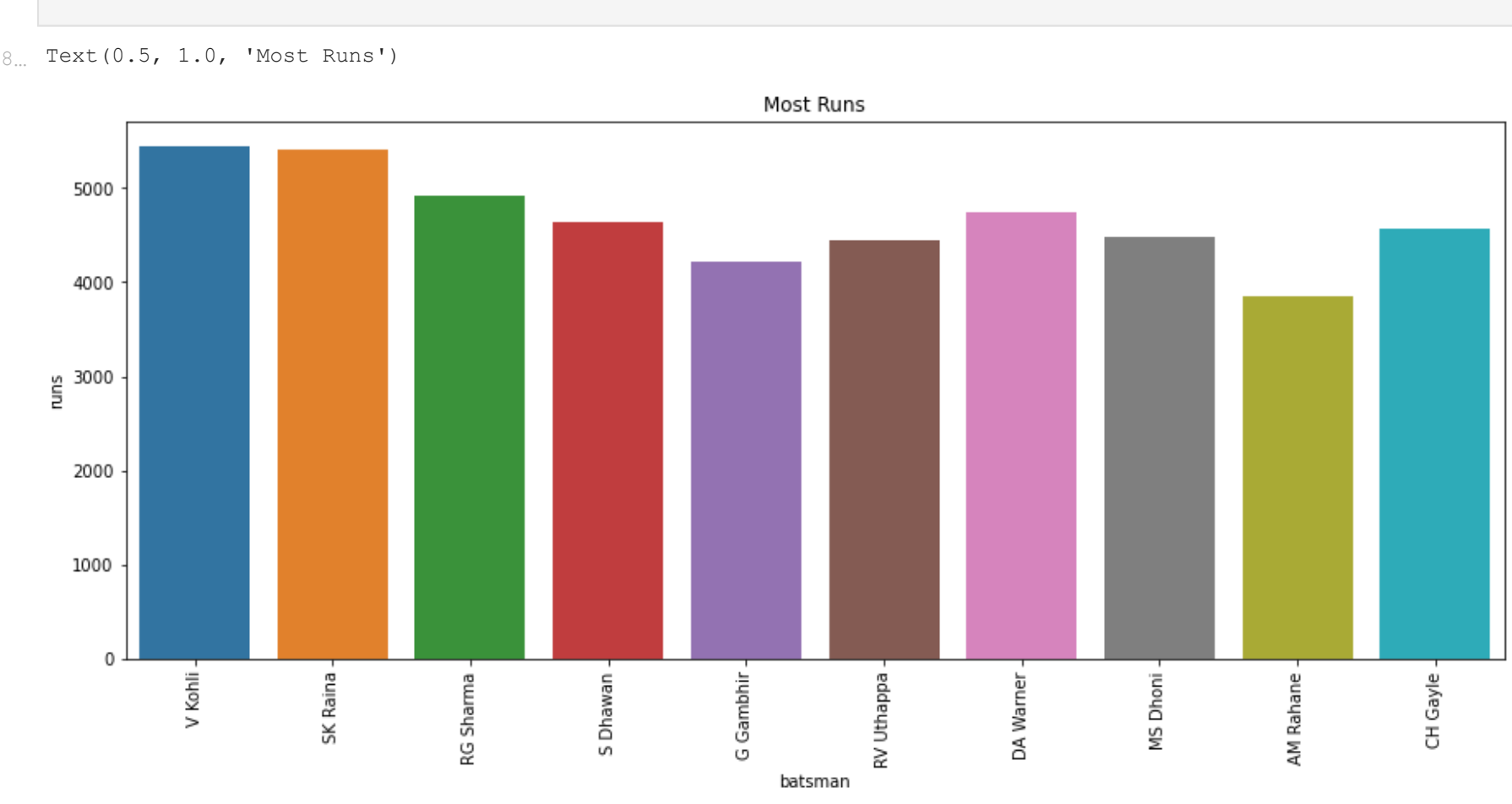
Out[207]: Text(0.5, 1.0, 'MOst Fours')



Talking about 4's **S Dhawan** Tops the list with over **500 4's** followed by **Sk Raina and G.Gambhir** with almost **500 4's**

In [208]:
```python
batsmen = dff.groupby("batsman").agg({'ball': 'count','batsman_runs': 'sum'})
batsmen.rename(columns={'ball':'balls', 'batsman_runs': 'runs'}, inplace=True)
batsmen = batsmen_summary.sort_values(['balls','runs'], ascending=False)[:10]
plt.figure(figsize=(15,6))
sns.barplot(x=batsmen.index,y=batsmen['runs'])
plt.xticks(rotation=90)
plt.title('Most Runs')
```

Out[208]: Text(0.5, 1.0, 'Most Runs')



V Kohil is the highest scorer in the whole tournament with above **5000 runs**

Sk Raina is next in line with V Kohil with just few runs short of him

Rg Sharma is next in line with almost **5000 runs**

In [209]:
```python
batsmen1 = batsmen
batsmen1['batting_strike_rate'] = batsmen1['runs']/batsmen1['balls'] * 100
batsmen1['batting_strike_rate'] = batsmen1['batting_strike_rate'].round(2)
batsmen1 = batsmen1[:10]
plt.figure(figsize=(15,6))
sns.barplot(x=batsmen1.index,y=batsmen['batting_strike_rate'])
plt.xticks(rotation=90)
plt.title('Best Strike Rate')
```

Out[209]: Text(0.5, 1.0, 'Best Strike Rate')

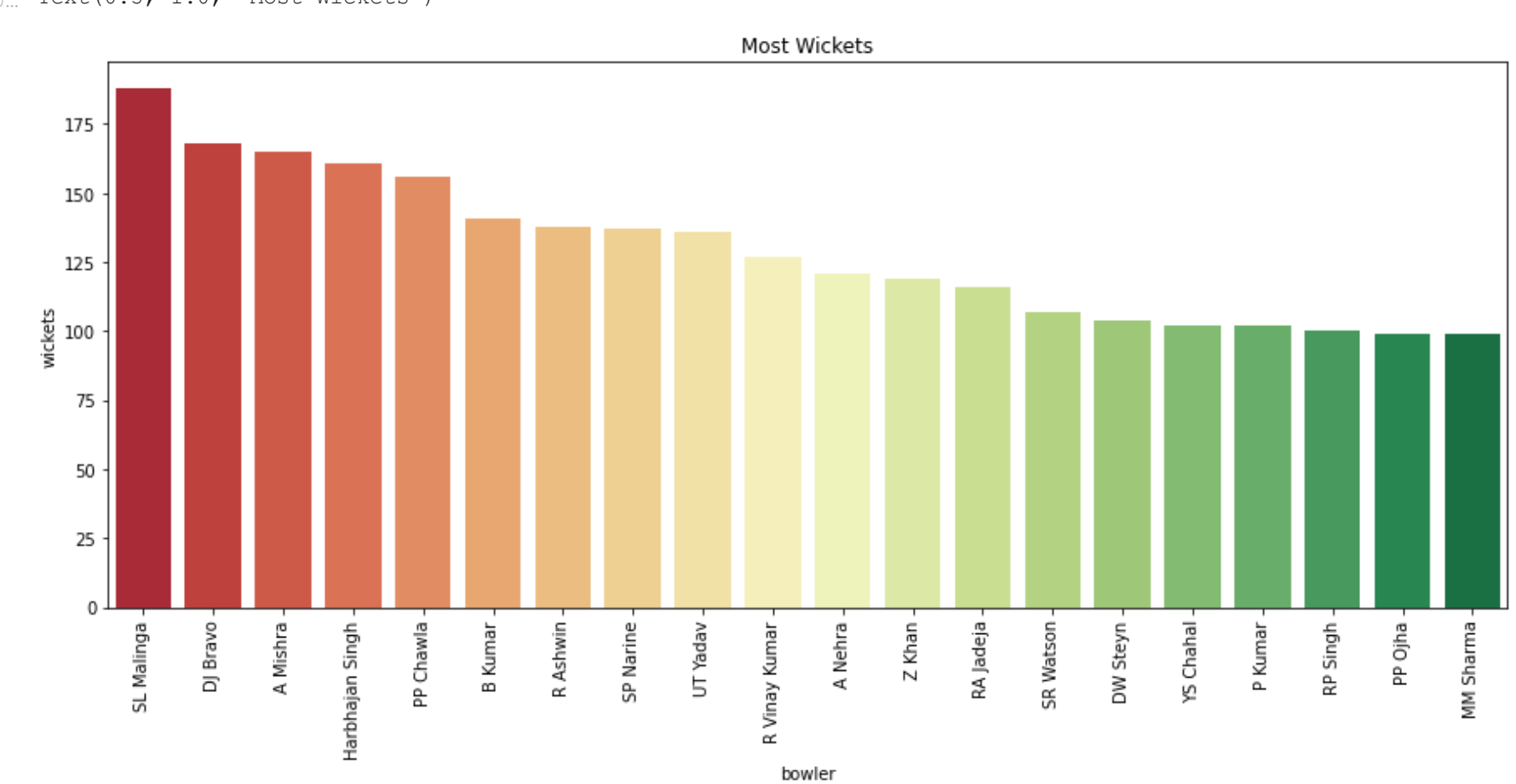
In T20 format strike rate is very important part of the game

CH Gayle, DA Warner and MS Dhoni top the list with strike rate of around **140**

Bowling Records

In [210]:
```python
bowler = dff.groupby('bowler').agg({'ball':'count','total_runs':'sum','player_dismissed':'count'})
bowler.rename(columns = {'ball':'balls','total_runs':'runs','player_dismissed':'wickets'},inplace=True)
bowler=bowler.sort_values(['wickets'],ascending=False)[:20]
plt.figure(figsize=(15,6))
sns.barplot(x=bowler.index,y=bowler['wickets'],palette='RdYlGn')
plt.xticks(rotation=90)
plt.title('Most Wickets')
```

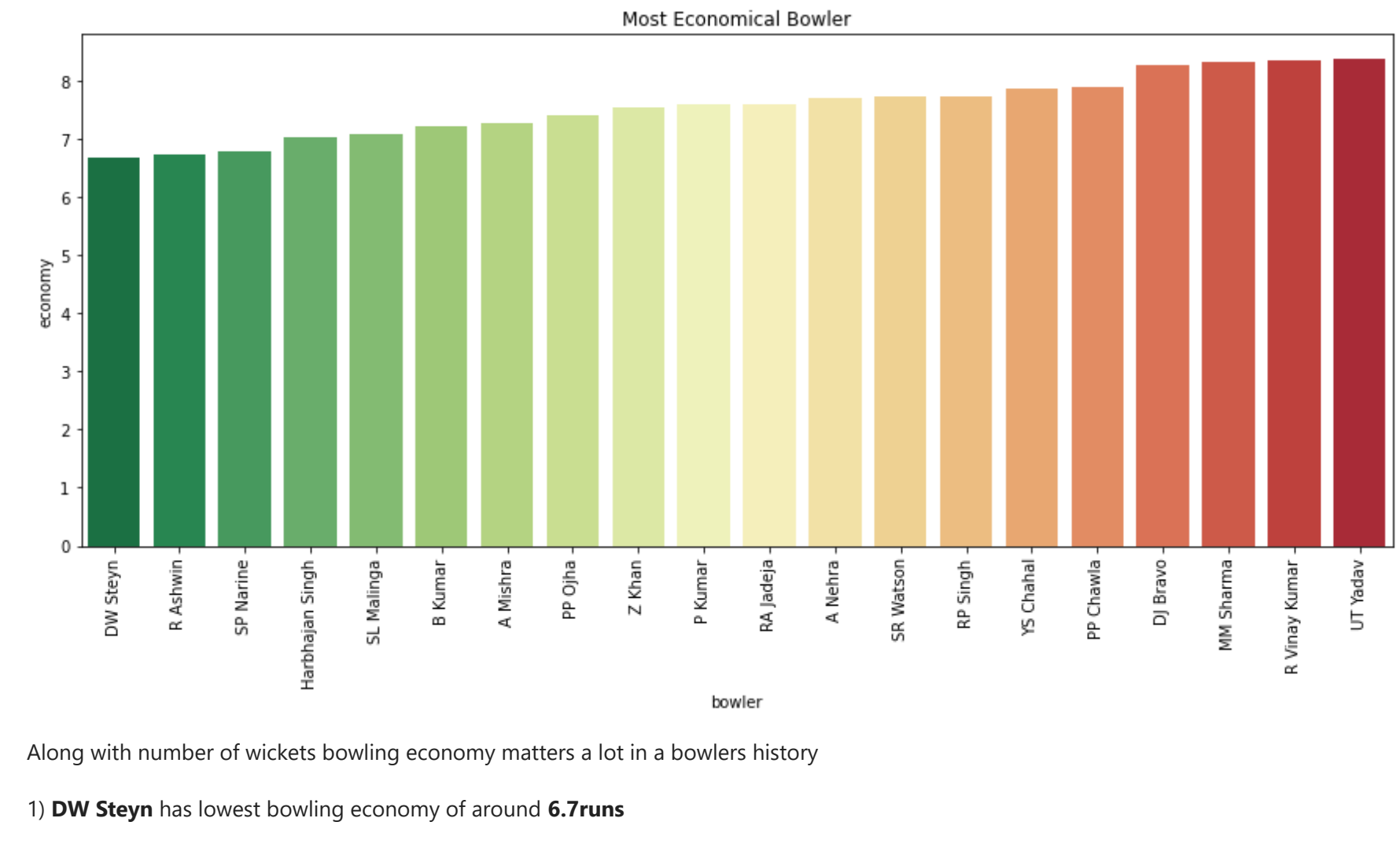
Out[210]: Text(0.5, 1.0, 'Most Wickets')



> **SL Malinga** is most destructive bowler with **over 175 wickets**
>
> Next is **Dj Bravo** with **almost 170 wickets** in his pocket
>
> **A Mishra** is next in line with **almost 170 wickets** but few wickets short to beat Dj Bravo

In [211]:
```python
bowler['economy'] = bowler['runs']/(bowler['balls']/6)
bowler = bowler.sort_values(['economy'],ascending=True)
plt.figure(figsize=(15,6))
sns.barplot(x=bowler.index,y=bowler['economy'],palette='RdYlGn_r')
plt.xticks(rotation=90)
plt.title('Most Economical Bowler')
```

Out[211]: Text(0.5, 1.0, 'Most Economical Bowler')



Along with number of wickets bowling economy matters a lot in a bowlers history

1) **DW Steyn** has lowest bowling economy of around **6.7runs**

2) Next is **R Ashwin** with bowling economy of around **6.8runs**

3) next is **SP Narine** with economy of approx **7.0runs**

After Detail analysis on the data we have found that

1) **Mumbai Indians** is Most Successful team in the history of tournament

2) **V Kohil** is most destructive batsmen

3) **Sl Malinga** is most fearced Bowler**

According to my analysis i will suggest that 1) **Mumbai Indians, Chennai Super Kings and Kolkata Knight Riders** are the top notch team to endorse

2) **V Kolhi, RG Sharma, CH Gayle, AB de Villers** are some of the best batsman to endorse

3) **SL Malinga, DW Steyn, DJ Bravo, A Mishra** are few best Bowlers to endorse

In [ ]: