

EDA on IPL Dataset

Importing required libraries

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

Loading the dataset

```
pip install pandas
```

```
➦ Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.22.4 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
```

```
pip install chardet
```

```
➦ Requirement already satisfied: chardet in /usr/local/lib/python3.10/dist-packages (5.2.0)
```

```
# Import necessary library
import pandas as pd
```

```
# Load the dataset
data = pd.read_csv(r"/content/matches (1).csv")
```

```
data
```

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied	winner	win_by_runs	win_b
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	
...
631	632	2016	Raipur	2016-05-22	Delhi Daredevils	Royal Challengers Bangalore	Royal Challengers Bangalore		field normal	0	Royal Challengers Bangalore	0	
632	633	2016	Bangalore	2016-05-24	Gujarat Lions	Royal Challengers Bangalore	Royal Challengers Bangalore		field normal	0	Royal Challengers Bangalore	0	
633	634	2016	Delhi	2016-05-25	Sunrisers Hyderabad	Kolkata Knight Riders	Kolkata Knight Riders		field normal	0	Sunrisers Hyderabad	22	
634	635	2016	Delhi	2016-05-27	Gujarat Lions	Sunrisers Hyderabad	Sunrisers Hyderabad		field normal	0	Sunrisers Hyderabad	0	
635	636	2016	Bangalore	2016-05-29	Sunrisers Hyderabad	Royal Challengers Bangalore	Sunrisers Hyderabad		bat normal	0	Sunrisers Hyderabad	8	

636 rows × 18 columns

data.shape

(636, 18)

data.info()

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 636 entries, 0 to 635
Data columns (total 18 columns):
#   Column                Non-Null Count  Dtype
---  ---                ---
0   id                    636 non-null   int64
1   season                636 non-null   int64
2   city                  629 non-null   object
3   date                  636 non-null   object
4   team1                 636 non-null   object
5   team2                 636 non-null   object
6   toss_winner           636 non-null   object
7   toss_decision         636 non-null   object
8   result                636 non-null   object
9   dl_applied            636 non-null   int64
10  winner                 633 non-null   object
11  win_by_runs            636 non-null   int64
12  win_by_wickets         636 non-null   int64
13  player_of_match        633 non-null   object
14  venue                  636 non-null   object
15  umpire1                635 non-null   object
16  umpire2                635 non-null   object
17  umpire3                0 non-null     float64
dtypes: float64(1), int64(5), object(12)
memory usage: 89.6+ KB

```

```
data.columns
```

```
Index(['id', 'season', 'city', 'date', 'team1', 'team2', 'toss_winner',
      'toss_decision', 'result', 'dl_applied', 'winner', 'win_by_runs',
      'win_by_wickets', 'player_of_match', 'venue', 'umpire1', 'umpire2',
      'umpire3'],
      dtype='object')
```

```
data.isnull()
```

```

      id  season  city  date  team1  team2  toss_winner  toss_decision  result  dl_applied  winner  win_by_runs  win_by_wickets  player
0  False  False  False  False  False  False  False      False      False  False      False      False      False      False
1  False  False  False  False  False  False  False      False      False  False      False      False      False      False
2  False  False  False  False  False  False  False      False      False  False      False      False      False      False
3  False  False  False  False  False  False  False      False      False  False      False      False      False      False
4  False  False  False  False  False  False  False      False      False  False      False      False      False      False
...    ...    ...    ...    ...    ...    ...      ...      ...    ...      ...      ...      ...      ...
631  False  False  False  False  False  False  False      False      False  False      False      False      False      False
632  False  False  False  False  False  False  False      False      False  False      False      False      False      False
633  False  False  False  False  False  False  False      False      False  False      False      False      False      False
634  False  False  False  False  False  False  False      False      False  False      False      False      False      False
635  False  False  False  False  False  False  False      False      False  False      False      False      False      False

```

636 rows × 18 columns

```
# data preprocessing. finding NaN values
data.isna().any()
```

```

      id  season  city  date  team1  team2  toss_winner  toss_decision  result  dl_applied  winner  win_by_runs  win_by_wickets  player_of_match  venue  umpire1  umpire2  umpire3
0  False  False  True  False  False  False  False      False      False  False      True      False      False      True      False      True      True      True

```

dtype: bool

```
# statistical description of dataset
data.describe()
```



	id	season	dl_applied	win_by_runs	win_by_wickets	umpire3
count	636.000000	636.000000	636.000000	636.000000	636.000000	0.0
mean	318.500000	2012.490566	0.025157	13.682390	3.372642	NaN
std	183.741666	2.773026	0.156726	23.908877	3.420338	NaN
min	1.000000	2008.000000	0.000000	0.000000	0.000000	NaN
25%	159.750000	2010.000000	0.000000	0.000000	0.000000	NaN
50%	318.500000	2012.000000	0.000000	0.000000	4.000000	NaN
75%	477.250000	2015.000000	0.000000	20.000000	7.000000	NaN
max	636.000000	2017.000000	1.000000	146.000000	10.000000	NaN

Q1) How many matches were played (in total) according to the dataset
`data['id'].count()`



636

Q2) How many IPL seasons are we using for analyse
`data['season'].unique()`



146

Q3) Which IPL team won by scoring maximum runs
`data['win_by_runs'].max()`
`data.iloc[data['win_by_runs'].idxmax()]`



	43
id	44
season	2017
city	Delhi
date	2017-05-06
team1	Mumbai Indians
team2	Delhi Daredevils
toss_winner	Delhi Daredevils
toss_decision	field
result	normal
dl_applied	0
winner	Mumbai Indians
win_by_runs	146
win_by_wickets	0
player_of_match	LMP Simmons
venue	Feroz Shah Kotla
umpire1	Nitin Menon
umpire2	CK Nandan
umpire3	NaN

Q4) Which IPL team won by maximum wickets
`data.iloc[data['win_by_wickets'].idxmax()]`



	2
id	3
season	2017
city	Rajkot
date	2017-04-07
team1	Gujarat Lions
team2	Kolkata Knight Riders
toss_winner	Kolkata Knight Riders
toss_decision	field
result	normal
dl_applied	0
winner	Kolkata Knight Riders
win_by_runs	0
win_by_wickets	10
player_of_match	CA Lynn
venue	Saurashtra Cricket Association Stadium
umpire1	Nitin Menon
umpire2	CK Nandan
umpire3	NaN

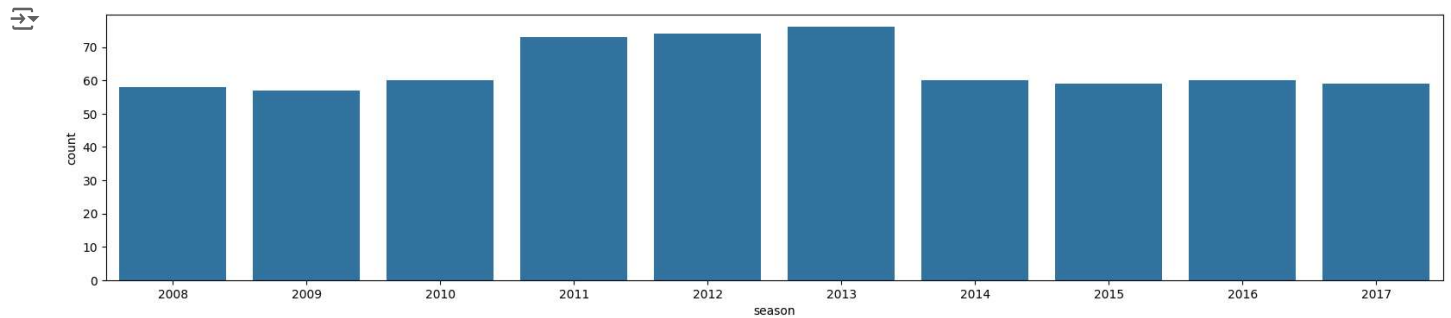
```
# Q5) which IPL team won by minimum wickets
data.iloc[data['win_by_wickets'].idxmin()]
```



	0
id	1
season	2017
city	Hyderabad
date	2017-04-05
team1	Sunrisers Hyderabad
team2	Royal Challengers Bangalore
toss_winner	Royal Challengers Bangalore
toss_decision	field
result	normal
dl_applied	0
winner	Sunrisers Hyderabad
win_by_runs	35
win_by_wickets	0
player_of_match	Yuvraj Singh
venue	Rajiv Gandhi International Stadium, Uppal
umpire1	AY Dandekar
umpire2	NJ Llong
umpire3	NaN

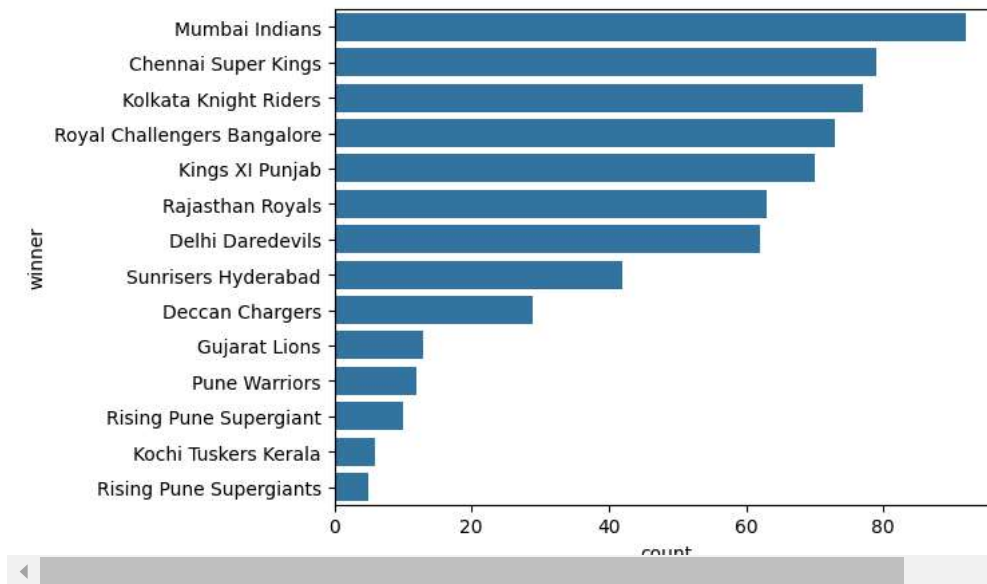
```
# Q6) which season consisted at highest number of matches played
import numpy as np
import pandas as pd
```

```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
fig_dims = (20, 4)
fig, ax = plt.subplots(figsize=fig_dims)
sns.countplot(x='season', ax=ax, data=data)
plt.show()
```




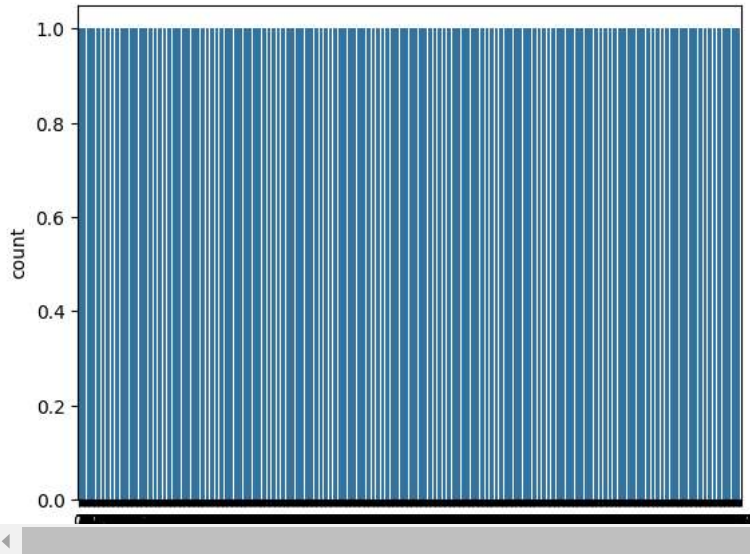
```
# 7) which is the most successful team
data1 = data.winner.value_counts()
sns.barplot(y=data1.index, x= data1)
```

<Axes: xlabel='count', ylabel='winner'>




```
# 8 ) probability of winning matches who won the toss
probability_of_win= data['toss_winner'] == data['winner']
probability_of_win.groupby(probability_of_win).size()
sns.countplot(probability_of_win)
```

 <Axes: ylabel='count'>



```
pd.set_option('display.max_rows', 99999)
pd.set_option('display.max_colwidth', 400)
```

```
pd.describe_option('max_colwidth')
```

 `display.max_colwidth` : int or None
 The maximum width in characters of a column in the repr of a pandas data structure. When the column overflows, a "..."
 placeholder is embedded in the output. A 'None' value means unlimited.
 [default: 50] [currently: 400]

```
# 9) highest matches won by team per season
data.groupby('season')['winner'].value_counts()
```



season	winner	count
2008	Rajasthan Royals	13
	Kings XI Punjab	10
	Chennai Super Kings	9
	Delhi Daredevils	7
	Mumbai Indians	7
	Kolkata Knight Riders	6
	Royal Challengers Bangalore	4
	Deccan Chargers	2
2009	Delhi Daredevils	10
	Deccan Chargers	9
	Royal Challengers Bangalore	9
	Chennai Super Kings	8
	Kings XI Punjab	7
	Rajasthan Royals	6
	Mumbai Indians	5
	Kolkata Knight Riders	3
2010	Mumbai Indians	11
	Chennai Super Kings	9
	Deccan Chargers	8
	Royal Challengers Bangalore	8
	Delhi Daredevils	7
	Kolkata Knight Riders	7
	Rajasthan Royals	6
	Kings XI Punjab	4
2011	Chennai Super Kings	11
	Mumbai Indians	10
	Royal Challengers Bangalore	10
	Kolkata Knight Riders	8
	Kings XI Punjab	7
	Deccan Chargers	6
	Kochi Tuskers Kerala	6
	Rajasthan Royals	6
	Delhi Daredevils	4
2012	Pune Warriors	4
	Kolkata Knight Riders	12
	Delhi Daredevils	11
	Chennai Super Kings	10
	Mumbai Indians	10
	Kings XI Punjab	8
	Royal Challengers Bangalore	8
	Rajasthan Royals	7
	Deccan Chargers	4
2013	Pune Warriors	4
	Mumbai Indians	13
	Chennai Super Kinas	12

	Rajasthan Royals	11
	Sunrisers Hyderabad	10
	Royal Challengers Bangalore	9
	Kings XI Punjab	8
	Kolkata Knight Riders	6
	Pune Warriors	4
	Delhi Daredevils	3
2014	Kings XI Punjab	12
	Kolkata Knight Riders	11
	Chennai Super Kings	10
	Mumbai Indians	7
	Rajasthan Royals	7
	Sunrisers Hyderabad	6
	Royal Challengers Bangalore	5
	Delhi Daredevils	2
2015	Chennai Super Kings	10
	Mumbai Indians	10
	Royal Challengers Bangalore	8
	Kolkata Knight Riders	7
	Rajasthan Royals	7
	Sunrisers Hyderabad	7
	Delhi Daredevils	5
	Kings XI Punjab	3
2016	Sunrisers Hyderabad	11
	Gujarat Lions	9
	Royal Challengers Bangalore	9
	Kolkata Knight Riders	8
	Delhi Daredevils	7
	Mumbai Indians	7
	Rising Pune Supergiants	5
	Kings XI Punjab	4
2017	Mumbai Indians	12
	Rising Pune Supergiant	10
	Kolkata Knight Riders	9
	Sunrisers Hyderabad	8
	Kings XI Punjab	7
	Delhi Daredevils	6
	Gujarat Lions	4
	Royal Challengers Bangalore	3