

Loading Data

```
In [1]: #IPL Dataset scrape
import pandas as pd
import numpy as np
df_ipl = pd.read_csv('df_ipl.csv', index_col=False)
```

```
In [2]: df_ipl
```

```
Out[2]:
```

	Unnamed: 0	season	city	date	team1	team2	toss_winner	toss_decision
0	0	2008	Bangalore	2008-04-18	Kolkata Knight Riders	Royal Challengers Bangalore	Royal Challengers Bangalore	field r
1	1	2008	Chandigarh	2008-04-19	Chennai Super Kings	Kings XI Punjab	Chennai Super Kings	bat r
2	2	2008	Delhi	2008-04-19	Rajasthan Royals	Delhi Daredevils	Rajasthan Royals	bat r
3	3	2008	Mumbai	2008-04-20	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat r
4	4	2008	Kolkata	2008-04-20	Deccan Chargers	Kolkata Knight Riders	Deccan Chargers	bat r
...
453	572	2016	Raipur	2016-05-22	Delhi Daredevils	Royal Challengers Bangalore	Royal Challengers Bangalore	field r
454	573	2016	Bangalore	2016-05-24	Gujarat Lions	Royal Challengers Bangalore	Royal Challengers Bangalore	field r
455	574	2016	Delhi	2016-05-25	Sunrisers Hyderabad	Kolkata Knight Riders	Kolkata Knight Riders	field r
456	575	2016	Delhi	2016-05-27	Gujarat Lions	Sunrisers Hyderabad	Sunrisers Hyderabad	field r
457	576	2016	Bangalore	2016-05-29	Sunrisers Hyderabad	Royal Challengers Bangalore	Sunrisers Hyderabad	bat r

458 rows × 17 columns

```
In [3]: del df_ip1['Unnamed: 0']
```

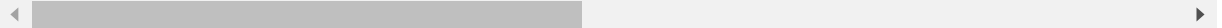
```
In [4]: df_ip1['season']=df_ip1['season'].astype('int').astype('object')
df_ip1['date']=pd.to_datetime(df_ip1['date'])
df_ip1['dl_applied'] = df_ip1['dl_applied'].astype('category')
df_ip1['toss_decision'] = df_ip1['toss_decision'].astype('category')
df_ip1['result']= df_ip1['result'].astype('category')
```

```
In [5]: df_ip1
```

```
Out[5]:
```

	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_a
0	2008	Bangalore	2008-04-18	Kolkata Knight Riders	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	
1	2008	Chandigarh	2008-04-19	Chennai Super Kings	Kings XI Punjab	Chennai Super Kings	bat	normal	
2	2008	Delhi	2008-04-19	Rajasthan Royals	Delhi Daredevils	Rajasthan Royals	bat	normal	
3	2008	Mumbai	2008-04-20	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	normal	
4	2008	Kolkata	2008-04-20	Deccan Chargers	Kolkata Knight Riders	Deccan Chargers	bat	normal	
...
453	2016	Raipur	2016-05-22	Delhi Daredevils	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	
454	2016	Bangalore	2016-05-24	Gujarat Lions	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	
455	2016	Delhi	2016-05-25	Sunrisers Hyderabad	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	
456	2016	Delhi	2016-05-27	Gujarat Lions	Sunrisers Hyderabad	Sunrisers Hyderabad	field	normal	
457	2016	Bangalore	2016-05-29	Sunrisers Hyderabad	Royal Challengers Bangalore	Sunrisers Hyderabad	bat	normal	

458 rows × 16 columns



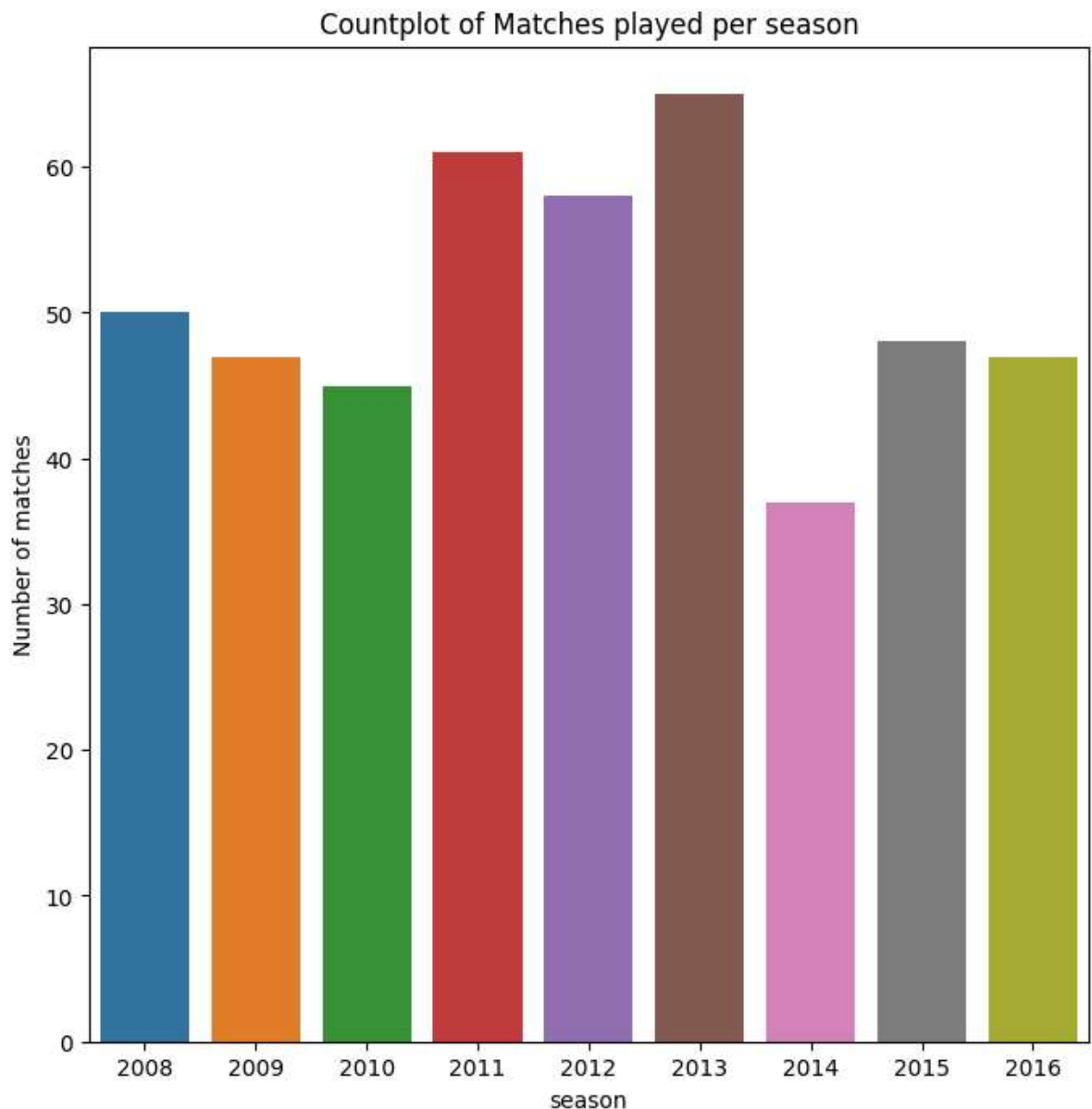
Data Visualization

```
In [6]: import matplotlib.pyplot as plt  
import seaborn as sns
```

1. How many matches are played in each season?

```
In [7]: fig,ax = plt.subplots(figsize=(8,8))  
sns.countplot(x='season',data=df_ipl)  
plt.ylabel('Number of matches')  
plt.title('Countplot of Matches played per season')
```

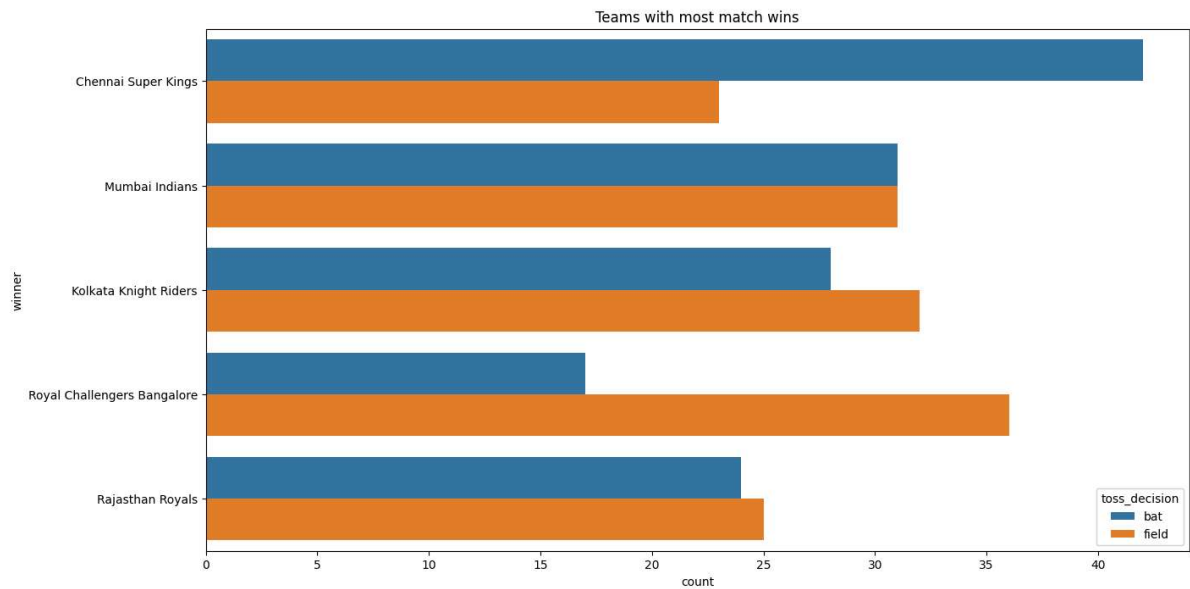
Out[7]: Text(0.5, 1.0, 'Countplot of Matches played per season')



2. Identify the teams with most wins based on their toss_choice.

```
In [8]: # Teams with highest wins and their toss decisions
fig,ax = plt.subplots(figsize=(15,8))
sns.countplot(y='winner',hue = 'toss_decision',data=df_ipl,order=df_ipl['winner'].value_counts().index)
plt.title('Teams with most match wins')
```

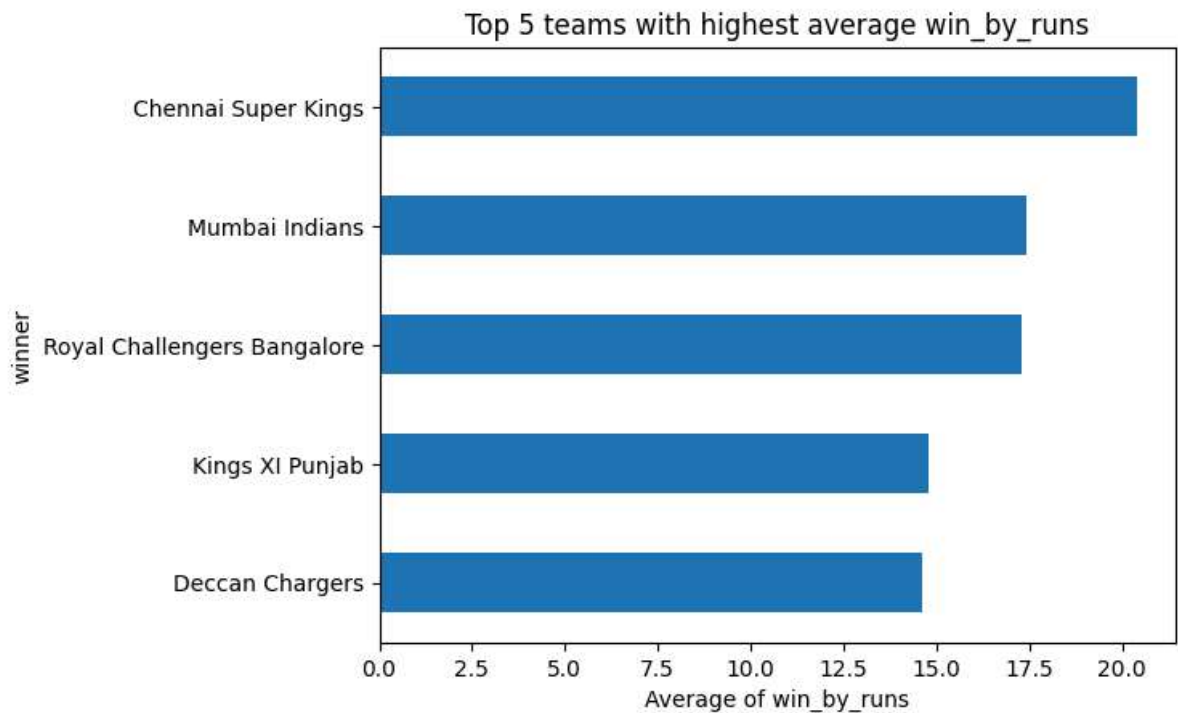
```
Out[8]: Text(0.5, 1.0, 'Teams with most match wins')
```



3. Which are the top 5 bowling teams? (Highest average win_by_runs)

```
In [9]: ax = df_ipl.groupby('winner')['win_by_runs'].mean().sort_values(ascending=False)
ax.invert_yaxis()
ax.set_xlabel('Average of win_by_runs')
```

```
Out[9]: Text(0.5, 0, 'Average of win_by_runs')
```

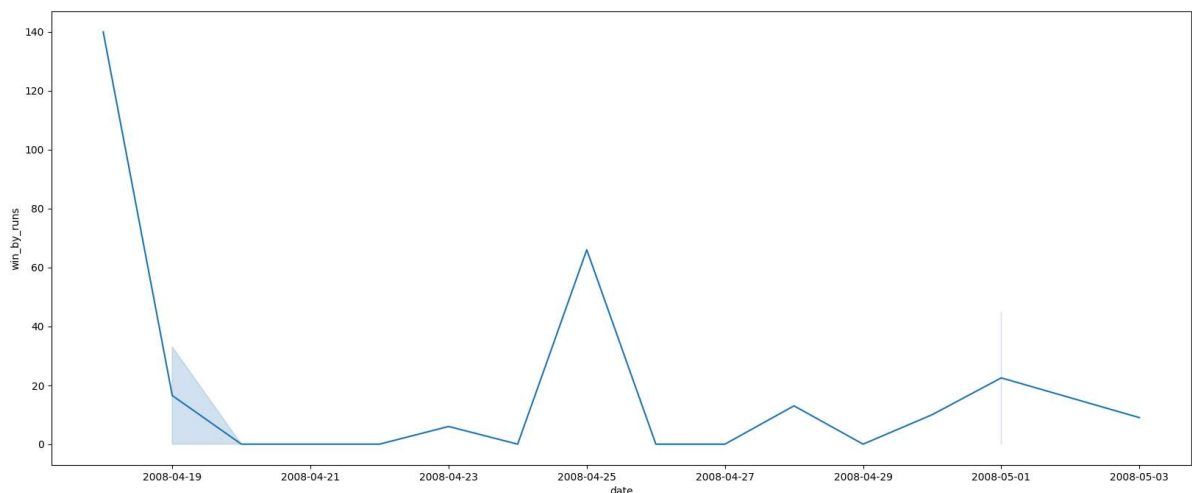


4. What is the trend of wins_by_runs over the duration of the years 2008 and 2016?

```
In [10]: #Line plot of wins_by_runs over time: Time Series analysis
fig,ax = plt.subplots(figsize=(20,8))

sns.lineplot(x=df_ipl['date'][:20],y='win_by_runs',data=df_ipl[:100])
```

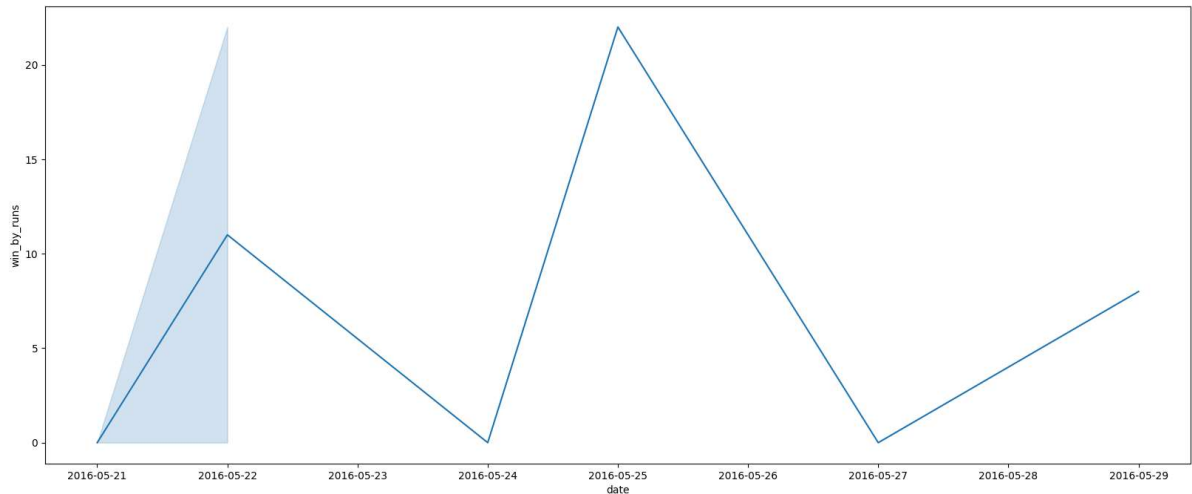
```
Out[10]: <Axes: xlabel='date', ylabel='win_by_runs'>
```



```
In [11]: #Line plot of wins_by_runs over time: Time Series analysis
fig,ax = plt.subplots(figsize=(20,8))

sns.lineplot(x=df_ipl['date'][450:],y='win_by_runs',data=df_ipl[400:])
```

Out[11]: <Axes: xlabel='date', ylabel='win_by_runs'>

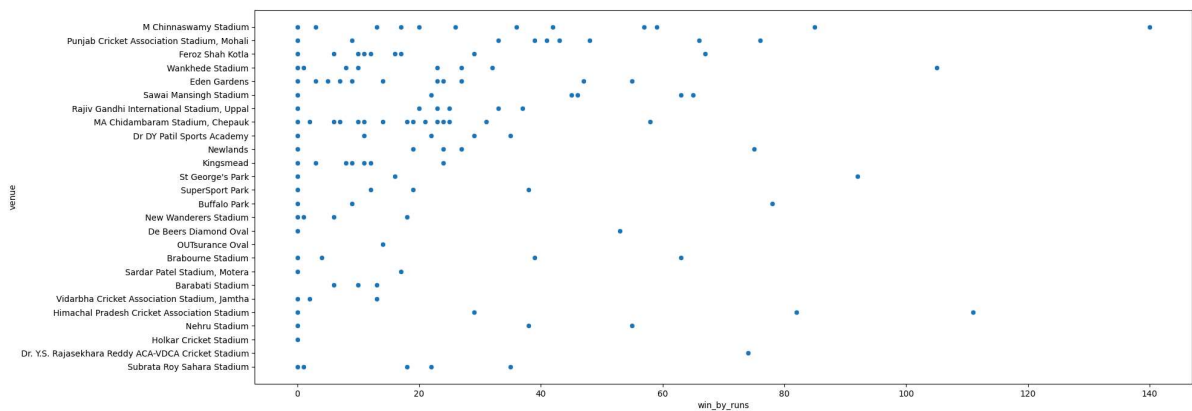


5. Which stadium witnessed the highest and lowest margins of win_by_runs?

```
In [12]: # Scatter plot of
plt.subplots(figsize=(20,8))

sns.scatterplot(y=df_ipl['venue'][:250],x='win_by_runs',data=df_ipl)
```

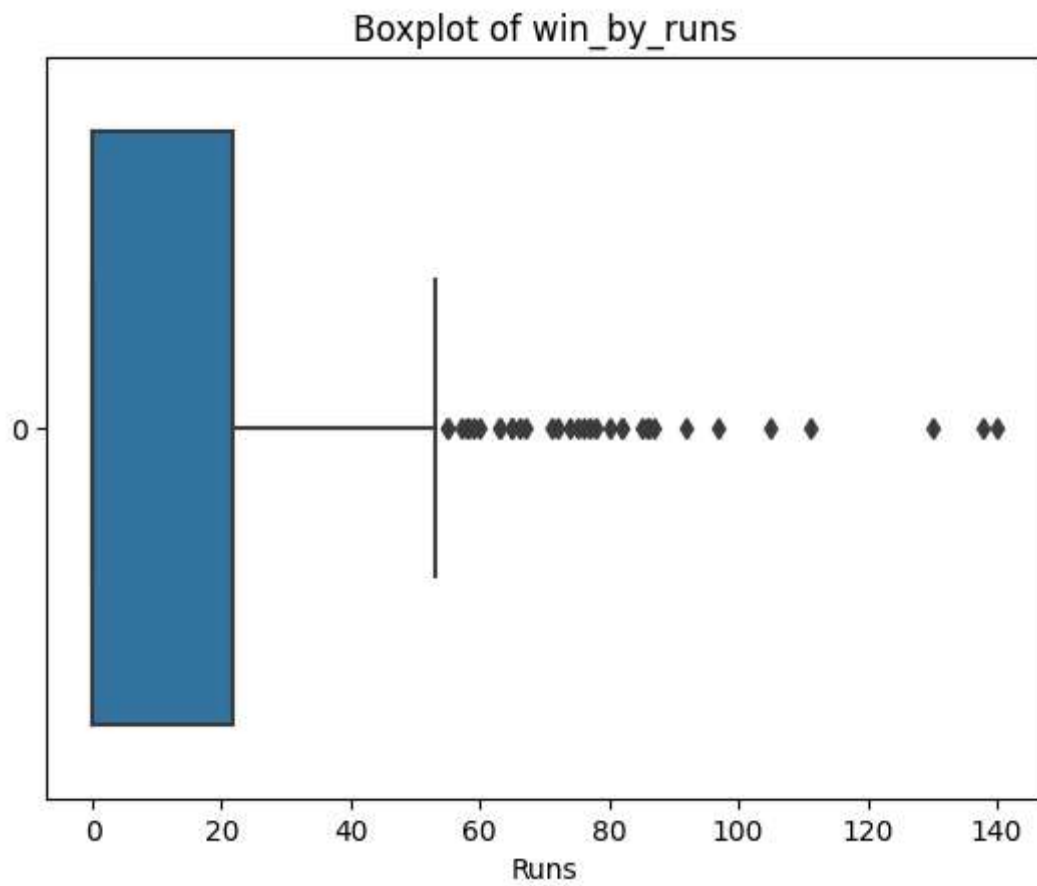
Out[12]: <Axes: xlabel='win_by_runs', ylabel='venue'>



6. Are there any outlier data in the win_by_runs, win_by_wickets columns? Identify the data distribution.

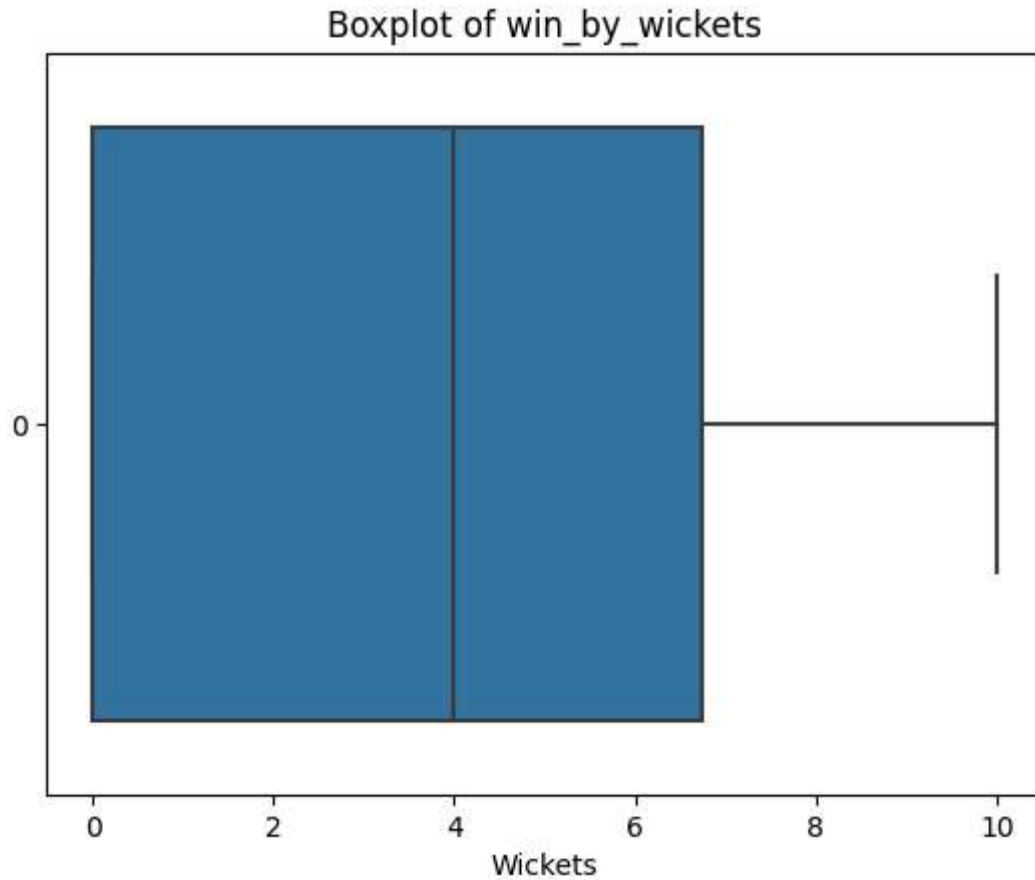
```
In [13]: ax = sns.boxplot(df_ip1['win_by_runs'],orient='h')  
ax.set_title('Boxplot of win_by_runs')  
ax.set_xlabel('Runs')
```

Out[13]: Text(0.5, 0, 'Runs')



```
In [14]: ax = sns.boxplot(df_ipl['win_by_wickets'],orient='h')
ax.set_title('Boxplot of win_by_wickets')
ax.set_xlabel('Wickets')
```

Out[14]: Text(0.5, 0, 'Wickets')



```
In [15]: df_ipl.describe()
```

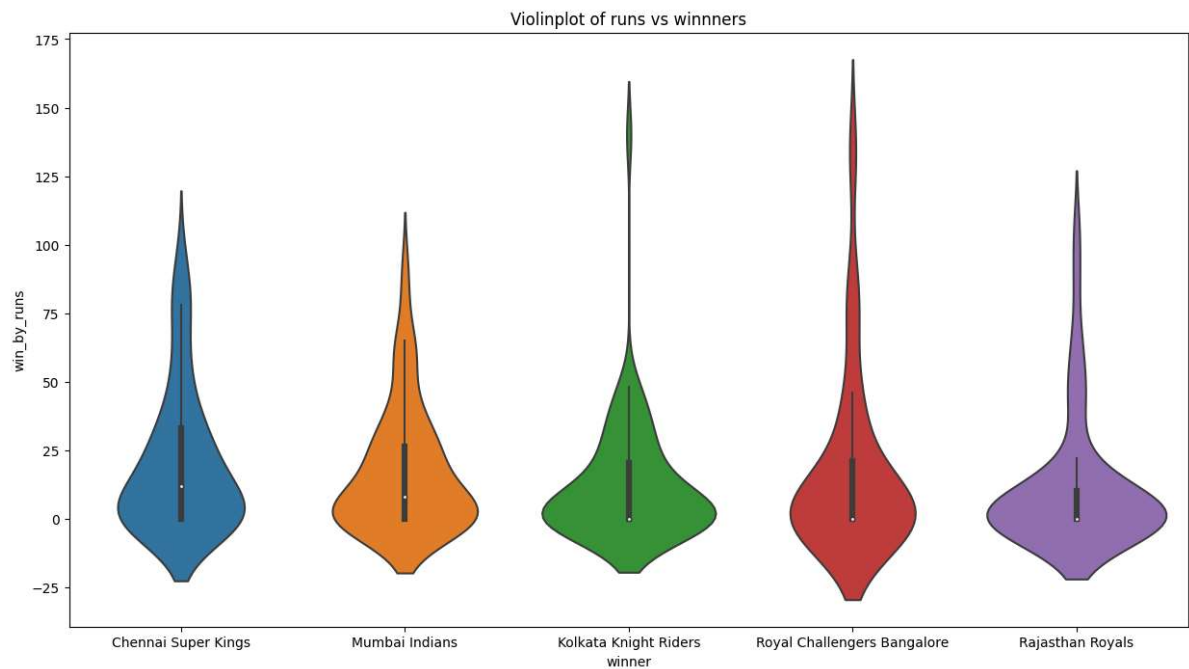
Out[15]:

	date	win_by_runs	win_by_wickets
count	458	458.000000	458.000000
mean	2012-04-12 19:29:36.419214080	13.875546	3.401747
min	2008-04-18 00:00:00	0.000000	0.000000
25%	2010-03-30 06:00:00	0.000000	0.000000
50%	2012-05-01 12:00:00	0.000000	4.000000
75%	2014-05-12 12:00:00	21.750000	6.750000
max	2016-05-29 00:00:00	140.000000	10.000000
std	NaN	23.381617	3.431760

7. Plot a violinplot showing the density of data distribution


```
In [16]: fig,ax=plt.subplots(figsize=(15,8))
sns.violinplot(x=df_ipl['winner'],y='win_by_runs',data=df_ipl,order=df_ipl['wi
ax.set_title('Violinplot of runs vs winnners')
```

```
Out[16]: Text(0.5, 1.0, 'Violinplot of runs vs winnners')
```



```
In [17]: df_ipl['winner'].value_counts()[:5]
```

```
Out[17]: winner
Chennai Super Kings    65
Mumbai Indians         62
Kolkata Knight Riders  60
Royal Challengers Bangalore  53
Rajasthan Royals       49
Name: count, dtype: int64
```

```
In [ ]:
```