Data Cleaning

In [1]:
```
# IPL Dataset scrape
import numpy as np
import pandas as pd
import requests
from bs4 import BeautifulSoup
req = requests.get("http://selfish-branch.surge.sh/",verify=False)
soup = BeautifulSoup(req.content, 'lxml')
table = soup.find_all('table')[0]
df_ipl = pd.read_html(str(table),index_col=None)[0]
```

In [2]:
```
df_ipl.head()
```

Out[2]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_
0	1.0	2008.0	Bangalore	2008-04-18	Kolkata Knight Riders	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	
1	2.0	2008.0	Chandigarh	2008-04-19	Chennai Super Kings	Kings XI Punjab	Chennai Super Kings	bat	normal	
2	3.0	2008.0	Delhi	2008-04-19	Rajasthan Royals	Delhi Daredevils	Rajasthan Royals	bat	normal	
3	4.0	2008.0	Mumbai	2008-04-20	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	normal	
4	5.0	2008.0	Kolkata	2008-04-20	Deccan Chargers	Kolkata Knight Riders	Deccan Chargers	bat	normal	

In [3]:
```
df_ipl.shape
```

Out[3]:
```
(577, 18)
```

Checking basic info about data

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 577 entries, 0 to 576
Data columns (total 18 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   id               512 non-null    float64
 1   season           524 non-null    float64
 2   city             570 non-null    object 
 3   date             577 non-null    object 
 4   team1            551 non-null    object 
 5   team2            538 non-null    object 
 6   toss_winner      577 non-null    object 
 7   toss_decision    577 non-null    object 
 8   result           577 non-null    object 
 9   dl_applied       577 non-null    int64  
 10  winner           574 non-null    object 
 11  win_by_runs      577 non-null    int64  
 12  win_by_wickets   577 non-null    int64  
 13  player_of_match  574 non-null    object 
 14  venue            577 non-null    object 
 15  umpire1          577 non-null    object 
 16  umpire2          577 non-null    object 
 17  umpire3          0 non-null      float64
dtypes: float64(3), int64(3), object(12)
memory usage: 81.3+ KB
```

Checking for missing values and duplicates

In [5]:
```
# Checking for missing values
df_ipl.umpire3.isna().sum()
```

Out[5]: 577

In [6]:
```
df_ipl.duplicated().sum()
```

Out[6]: 0

In [7]:
```
del df_ipl['umpire3']
```

In [8]:
```
del df_ipl['id']
```

In [9]:
```
df_ipl.isna().sum()
```

Out[9]:
```
season             53
city                7
date                0
team1              26
team2              39
toss_winner         0
toss_decision       0
result              0
dl_applied          0
winner              3
win_by_runs         0
win_by_wickets      0
player_of_match     3
venue               0
umpire1             0
umpire2             0
dtype: int64
```

Dropping data having missing values

In [10]:
```
df_ipl.dropna(inplace=True)
```

In [11]:
```
df_ipl.isna().sum()
```

Out[11]:
```
season             0
city               0
date               0
team1              0
team2              0
toss_winner        0
toss_decision      0
result             0
dl_applied         0
winner             0
win_by_runs        0
win_by_wickets     0
player_of_match    0
venue              0
umpire1            0
umpire2            0
dtype: int64
```

In [12]:
```
df_ipl.shape
```

Out[12]:
```
(458, 16)
```

Performing type convertion

In [13]:
```
df_ipl['season']=df_ipl['season'].astype('int').astype('object')
df_ipl['date']=pd.to_datetime(df_ipl['date'])
df_ipl['dl_applied'] = df_ipl['dl_applied'].astype('category')
df_ipl['toss_decision'] = df_ipl['toss_decision'].astype('category')
df_ipl['result']= df_ipl['result'].astype('category')
```

In [14]:
```
df_ipl.dtypes
```

Out[14]:
```
season                  object
city                    object
date              datetime64[ns]
team1                   object
team2                   object
toss_winner             object
toss_decision         category
result                category
dl_applied            category
winner                  object
win_by_runs              int64
win_by_wickets           int64
player_of_match         object
venue                   object
umpire1                 object
umpire2                 object
dtype: object
```

Cleaned dataset

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

Out[15]:

	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_appl
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	

Basic EDA

In [16]:
```
# Basic Statistical Info: 5 Number Summary
df_ipl.describe()
```

Out[16]:

	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

In [17]:
```
import numpy as np
Q3 = np.percentile(df_ipl['win_by_runs'],75)
```

In [18]:
```
Q3
```

Out[18]: 21.75

In [19]:
```
len(list(df_ipl.season.value_counts().index))
```

Out[19]: 9

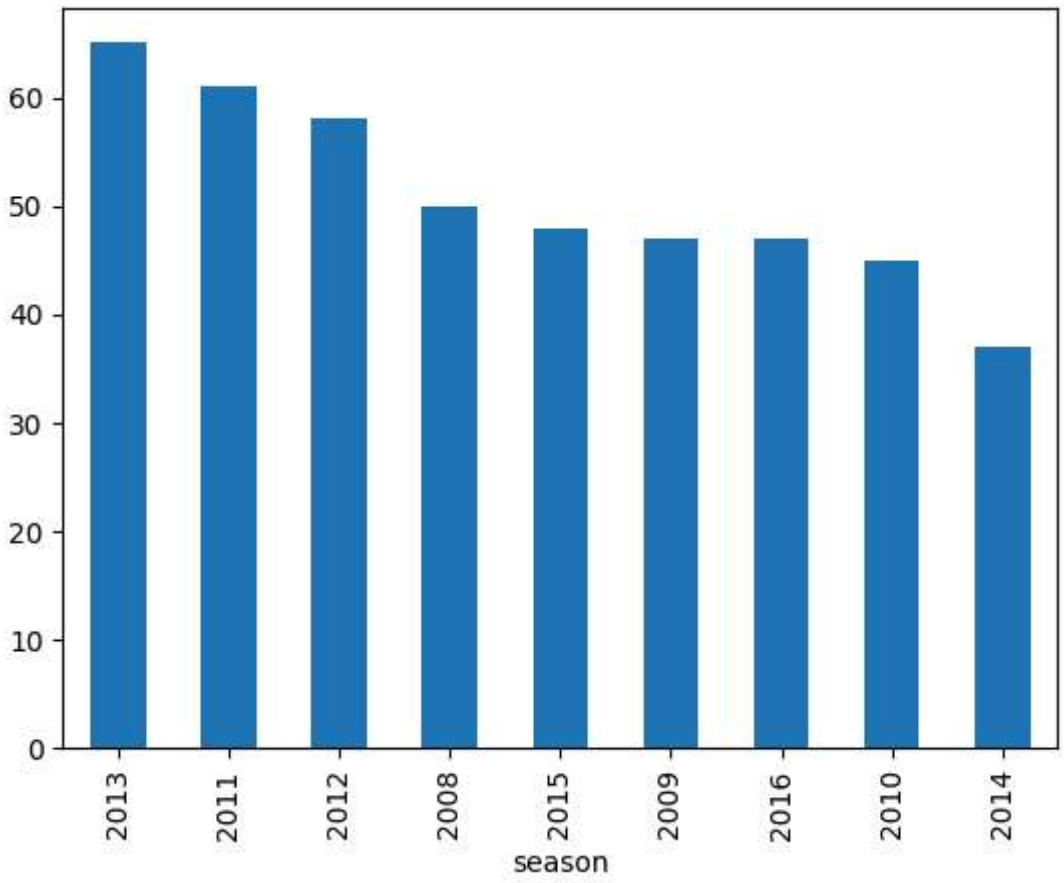
In [20]:
```
df_ipl.city.value_counts()
```

Out[20]:
```
city
Mumbai            64
Kolkata           48
Bangalore         46
Delhi             40
Chennai           39
Chandigarh        33
Hyderabad         30
Jaipur            28
Pune              19
Centurion         11
Durban            10
Dharamsala         9
Ahmedabad          9
Visakhapatnam      8
Johannesburg       8
Cape Town          6
Cuttack            5
Rajkot             5
Port Elizabeth     5
Raipur             5
Ranchi             5
Sharjah            4
Kochi              4
Abu Dhabi          4
Kimberley          3
Nagpur             3
East London        3
Kanpur             2
Indore             1
Bloemfontein       1
Name: count, dtype: int64
```

1. How many games have happened across all the seasons so far?

In [21]:
```
df_ipl.season.value_counts().plot(kind='bar')
```

Out[21]:
```
<Axes: xlabel='season'>
```



2. How many seasons has IPL seen so far?

In [22]:
```
df_ipl.season.value_counts().count()
```

Out[22]:
```
9
```

3. What are the top three teams that have won the most number of matches played and sum up the those matches from these top three teams.

In [23]:
```
df_ipl.winner.value_counts()[:3]
```

Out[23]:
```
winner
Chennai Super Kings      65
Mumbai Indians           62
Kolkata Knight Riders    60
Name: count, dtype: int64
```

In [24]:
```
df_ipl.winner.value_counts()[:3].sum()
```

Out[24]:
```
187
```

4. Identify the teams that have registered at least a single win in IPL so far. How many such teams are there since the inception of IPL?

In [25]:
```
df_ipl.winner.value_counts()
```

Out[25]:
```
winner
Chennai Super Kings            65
Mumbai Indians                 62
Kolkata Knight Riders          60
Royal Challengers Bangalore    53
Rajasthan Royals               49
Kings XI Punjab                48
Delhi Daredevils               44
Deccan Chargers                28
Sunrisers Hyderabad            24
Pune Warriors                   9
Gujarat Lions                   9
Rising Pune Supergiants         5
Kochi Tuskers Kerala            2
Name: count, dtype: int64
```

In [26]:
```
x = df_ipl.winner.value_counts()>=1
print(x)
len(x)
```

```
winner
Chennai Super Kings            True
Mumbai Indians                 True
Kolkata Knight Riders          True
Royal Challengers Bangalore    True
Rajasthan Royals               True
Kings XI Punjab                True
Delhi Daredevils               True
Deccan Chargers                True
Sunrisers Hyderabad            True
Pune Warriors                  True
Gujarat Lions                  True
Rising Pune Supergiants        True
Kochi Tuskers Kerala           True
Name: count, dtype: bool
```

Out[26]:
```
13
```

5. Which stadium has hosted the most number of IPL matches till date and get the count of those matches?

In [27]:
```
df_ipl.venue.value_counts()[0]
```

Out[27]:
```
48
```

In [ ]:

In [ ]:

In [28]:
```
round(df_ipl.groupby('venue')['win_by_wickets'].mean().sort_values(ascending=F
```

Out[28]:
```
venue
Saurashtra Cricket Association Stadium                 6.20
Green Park                                             6.00
Holkar Cricket Stadium                                 6.00
Punjab Cricket Association IS Bindra Stadium, Mohali   6.00
Sawai Mansingh Stadium                                 4.75
SuperSport Park                                        4.45
Dr DY Patil Sports Academy                             4.38
New Wanderers Stadium                                  4.38
Rajiv Gandhi International Stadium, Uppal              4.27
Sardar Patel Stadium, Motera                           4.22
Eden Gardens                                           3.96
Shaheed Veer Narayan Singh International Stadium       3.80
Feroz Shah Kotla                                       3.60
Sharjah Cricket Stadium                                3.50
De Beers Diamond Oval                                  3.33
Maharashtra Cricket Association Stadium                3.33
M Chinnaswamy Stadium                                  3.30
Nehru Stadium                                          3.25
Punjab Cricket Association Stadium, Mohali             3.21
St George's Park                                       3.20
JSCA International Stadium Complex                     3.20
Sheikh Zayed Stadium                                   3.00
Barabati Stadium                                       3.00
Newlands                                               3.00
Wankhede Stadium                                       2.78
Himachal Pradesh Cricket Association Stadium           2.56
Kingsmead                                              2.40
Buffalo Park                                           2.33
Brabourne Stadium                                      2.30
MA Chidambaram Stadium, Chepauk                        2.05
Subrata Roy Sahara Stadium                             2.00
Vidarbha Cricket Association Stadium, Jamtha           2.00
Dr. Y.S. Rajasekhara Reddy ACA-VDCA Cricket Stadium    1.25
OUTsurance Oval                                        0.00
Name: win_by_wickets, dtype: float64
```

10. Identify the top venues which are good for chasing. Which has highest average win_by_wickets values?

In [29]:
```
round(df_ipl.groupby('winner')['win_by_runs'].mean().sort_values(ascending=Fal
```

Out[29]:
```
winner
Chennai Super Kings            20.400
Mumbai Indians                 17.419
Royal Challengers Bangalore    17.302
Kings XI Punjab                14.771
Deccan Chargers                14.607
Kolkata Knight Riders          12.367
Rajasthan Royals               11.367
Rising Pune Supergiants        10.600
Pune Warriors                  10.111
Sunrisers Hyderabad             7.458
Delhi Daredevils                6.614
Gujarat Lions                   0.111
Kochi Tuskers Kerala            0.000
Name: win_by_runs, dtype: float64
```

In [30]:
```
round(df_ipl.groupby('winner')['win_by_wickets'].mean().sort_values(ascending=
```

Out[30]:
```
winner
Kochi Tuskers Kerala           7.500
Delhi Daredevils               4.614
Gujarat Lions                  4.556
Rajasthan Royals               4.184
Rising Pune Supergiants        4.000
Royal Challengers Bangalore    3.943
Kolkata Knight Riders          3.533
Sunrisers Hyderabad            3.458
Pune Warriors                  3.444
Kings XI Punjab                3.000
Mumbai Indians                 2.629
Deccan Chargers                2.571
Chennai Super Kings            2.462
Name: win_by_wickets, dtype: float64
```

In [31]:
```
df_ipl.groupby('winner')['win_by_runs'].mean()
```

Out[31]:
```
winner
Chennai Super Kings            20.400000
Deccan Chargers                14.607143
Delhi Daredevils                6.613636
Gujarat Lions                   0.111111
Kings XI Punjab                14.770833
Kochi Tuskers Kerala            0.000000
Kolkata Knight Riders          12.366667
Mumbai Indians                 17.419355
Pune Warriors                  10.111111
Rajasthan Royals               11.367347
Rising Pune Supergiants        10.600000
Royal Challengers Bangalore    17.301887
Sunrisers Hyderabad             7.458333
Name: win_by_runs, dtype: float64
```

In [32]:
```
df_ipl
```

Out[32]:

	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	
...	...	...	...	...	...	...	...	...	
572	2016	Raipur	2016-05-22	Delhi Daredevils	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	
573	2016	Bangalore	2016-05-24	Gujarat Lions	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	
574	2016	Delhi	2016-05-25	Sunrisers Hyderabad	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	
575	2016	Delhi	2016-05-27	Gujarat Lions	Sunrisers Hyderabad	Sunrisers Hyderabad	field	normal	
576	2016	Bangalore	2016-05-29	Sunrisers Hyderabad	Royal Challengers Bangalore	Sunrisers Hyderabad	bat	normal	

458 rows × 16 columns

9. Identify the top bowling teams. What is the average runs by which teams won games? Which team has the highest average?

In [33]:
```
x=(df_ipl.venue.value_counts()>=1)
print(x)
len(x)
```

```
venue
Eden Gardens                                        True
M Chinnaswamy Stadium                               True
Wankhede Stadium                                    True
Feroz Shah Kotla                                    True
MA Chidambaram Stadium, Chepauk                     True
Rajiv Gandhi International Stadium, Uppal           True
Punjab Cricket Association Stadium, Mohali          True
Sawai Mansingh Stadium                              True
Dr DY Patil Sports Academy                          True
Subrata Roy Sahara Stadium                          True
SuperSport Park                                     True
Brabourne Stadium                                   True
Kingsmead                                           True
Himachal Pradesh Cricket Association Stadium        True
Sardar Patel Stadium, Motera                        True
New Wanderers Stadium                               True
Dr. Y.S. Rajasekhara Reddy ACA-VDCA Cricket Stadium True
Newlands                                            True
Maharashtra Cricket Association Stadium             True
Barabati Stadium                                    True
Saurashtra Cricket Association Stadium              True
Shaheed Veer Narayan Singh International Stadium    True
JSCA International Stadium Complex                  True
St George's Park                                    True
Punjab Cricket Association IS Bindra Stadium, Mohali True
Nehru Stadium                                       True
Sheikh Zayed Stadium                                True
Sharjah Cricket Stadium                             True
Vidarbha Cricket Association Stadium, Jamtha        True
Buffalo Park                                        True
De Beers Diamond Oval                               True
Green Park                                          True
OUTsurance Oval                                     True
Holkar Cricket Stadium                              True
Name: count, dtype: bool
```

Out[33]: 34

8. How many stadiums have hosted atleast one IPL game?

In [34]:
```
win_per(d1,d2,d3)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[34], line 1
----> 1 win_per(d1,d2,d3)

NameError: name 'win_per' is not defined
```

In [ ]:
```
def win_per(d1,d2,d3):
    # Getting sorted list of wins and games player per team
    wins_dict = dict(sorted(d3.items()))
    total_games_dict1 = dict(sorted(d1.items()))
    total_games_dict2 = dict(sorted(d2.items()))
    total_games_dict = total_games_dict1

    # Creating final total games dictionary sorted alphabetically
    lst = [(i+j) for i,j in zip(total_games_dict1.values(),total_games_dict2.v
    for i,j in enumerate(total_games_dict):
        total_games_dict[j]=lst[i]

    #Finding percentage of wins and extracting successful teams
    success_list = []
    for i,j in zip(wins_dict.items(),total_games_dict.items()):
        if(round((i[1]/j[1])*100))>=50:
            success_list.append(i[0])
    print(success_list)
    return len(success_list)
```

6. Get the count of teams having win percentage greater than or equal to 50%.

In [ ]:
```
total_wins = mi_wins+sh_wins+kp_wins
total_games = mi_total+sh_total+kp_total
round((total_wins/total_games)*100)
```

In [ ]:
```
mi_per = round((mi_wins/mi_total)*100)
sh_per = round((sh_wins/sh_total)*100)
kp_per = round((kp_wins/kp_total)*100)
print("--Win Percentages--")
print(f'Mumbai Indians: {mi_per}% \nSunrisers Hyderabad: {sh_per}%\nKings XI P
```

In [ ]:
```
kp_wins
```

In [ ]:
```
mi_wins = d3['Mumbai Indians']
sh_wins = d3['Sunrisers Hyderabad']
kp_wins = d3['Kings XI Punjab']
```

In [ ]:
```
kp_total
```