IPL DATA ANALAYSIS

📚 IMPORTING LIBRARIES.

```
In [1]:  import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as sns
         %matplotlib inline
```

```
In [2]:  dataset=pd.read_csv("IPL_DATASET.csv")
         dataset.head()
```

Out[2]:

	id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	win
0	335982	Bangalore	2008-04-18	BB McCullum	M Chinnaswamy Stadium	0	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore	field	Koll Kn Ric
1	335983	Chandigarh	2008-04-19	MEK Hussey	Punjab Cricket Association Stadium, Mohali	0	Kings XI Punjab	Chennai Super Kings	Chennai Super Kings	bat	Cher Su K
2	335984	Delhi	2008-04-19	MF Maharoof	Feroz Shah Kotla	0	Delhi Daredevils	Rajasthan Royals	Rajasthan Royals	bat	D Darede
3	335985	Mumbai	2008-04-20	MV Boucher	Wankhede Stadium	0	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	R Challeng Banga
4	335986	Kolkata	2008-04-20	DJ Hussey	Eden Gardens	0	Kolkata Knight Riders	Deccan Chargers	Deccan Chargers	bat	Koll Kn Ric

PRE-PROCESSING.

INFORMATION ABOUT DATASET

In [3]: dataset.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 816 entries, 0 to 815
Data columns (total 17 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              816 non-null    int64
 1   city            803 non-null    object
 2   date            816 non-null    object
 3   player_of_match 812 non-null    object
 4   venue           816 non-null    object
 5   neutral_venue   816 non-null    int64
 6   team1           816 non-null    object
 7   team2           816 non-null    object
 8   toss_winner     816 non-null    object
 9   toss_decision   816 non-null    object
 10  winner          812 non-null    object
 11  result          812 non-null    object
 12  result_margin   799 non-null    float64
 13  eliminator      812 non-null    object
 14  method          19 non-null     object
 15  umpire1         816 non-null    object
 16  umpire2         816 non-null    object
dtypes: float64(1), int64(2), object(14)
memory usage: 108.5+ KB
```

Removing unwanted columns from the dataset

In [4]: dataset.drop(["id","neutral_venue","method"],axis=1,inplace=True)

In [5]: `dataset.head()`

Out[5]:

	city	date	player_of_match	venue	team1	team2	toss_winner	toss_decision	winner	result	result_mar
0	Bangalore	2008-04-18	BB McCullum	M Chinnaswamy Stadium	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore	field	Kolkata Knight Riders	runs	14
1	Chandigarh	2008-04-19	MEK Hussey	Punjab Cricket Association Stadium, Mohali	Kings XI Punjab	Chennai Super Kings	Chennai Super Kings	bat	Chennai Super Kings	runs	3
2	Delhi	2008-04-19	MF Maharoof	Feroz Shah Kotla	Delhi Daredevils	Rajasthan Royals	Rajasthan Royals	bat	Delhi Daredevils	wickets	
3	Mumbai	2008-04-20	MV Boucher	Wankhede Stadium	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	Royal Challengers Bangalore	wickets	
4	Kolkata	2008-04-20	DJ Hussey	Eden Gardens	Kolkata Knight Riders	Deccan Chargers	Deccan Chargers	bat	Kolkata Knight Riders	wickets	

In [6]: *# REMOVING NULL VALUES FROM THE DATASET*

In [7]:
```
#null values
#print(dataset.isnull().sum())
df=dataset
```

Filling null values in 4 columns as None since the empty rows in 'result' column indicating that the match was called off and hence no winner

```
In [8]:  df['result']=df['result'].fillna('None')
         df['eliminator']=df['eliminator'].fillna('None')
         df['player_of_match']=df['player_of_match'].fillna('None')
         df['winner']=df['winner'].fillna('None')
         df.isnull().sum()
```

```
Out[8]:  city                13
         date                 0
         player_of_match      0
         venue                0
         team1                0
         team2                0
         toss_winner          0
         toss_decision        0
         winner               0
         result               0
         result_margin       17
         eliminator           0
         umpire1              0
         umpire2              0
         dtype: int64
```

```
In [9]: #Rows having 'result_margin' column as null
        #The corresponding mathes were tied
        df[df['result_margin'].isnull()]
```

Out[9]:

	city	date	player_of_match	venue	team1	team2	toss_winner	toss_decision	winner	result	result_
66	Cape Town	2009-04-23	YK Pathan	Newlands	Kolkata Knight Riders	Rajasthan Royals	Kolkata Knight Riders	field	Rajasthan Royals	tie	
130	Chennai	2010-03-21	J Theron	MA Chidambaram Stadium, Chepauk	Chennai Super Kings	Kings XI Punjab	Chennai Super Kings	field	Kings XI Punjab	tie	
241	Delhi	2011-05-21	None	Feroz Shah Kotla	Delhi Daredevils	Pune Warriors	Delhi Daredevils	bat	None	None	
328	Hyderabad	2013-04-07	GH Vihari	Rajiv Gandhi International Stadium, Uppal	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	bat	Sunrisers Hyderabad	tie	
341	Bangalore	2013-04-16	V Kohli	M Chinnaswamy Stadium	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	field	Royal Challengers Bangalore	tie	
416	Abu Dhabi	2014-04-29	JP Faulkner	Sheikh Zayed Stadium	Kolkata Knight Riders	Rajasthan Royals	Rajasthan Royals	bat	Rajasthan Royals	tie	
476	Ahmedabad	2015-04-21	SE Marsh	Sardar Patel Stadium, Motera	Rajasthan Royals	Kings XI Punjab	Kings XI Punjab	field	Kings XI Punjab	tie	
486	Bangalore	2015-04-29	None	M Chinnaswamy Stadium	Royal Challengers Bangalore	Rajasthan Royals	Rajasthan Royals	field	None	None	
511	Bangalore	2015-05-17	None	M Chinnaswamy Stadium	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	field	None	None	
610	Rajkot	2017-04-29	KH Pandya	Saurashtra Cricket Association Stadium	Gujarat Lions	Mumbai Indians	Gujarat Lions	bat	Mumbai Indians	tie	
705	Delhi	2019-03-30	PP Shaw	Feroz Shah Kotla	Delhi Capitals	Kolkata Knight Riders	Delhi Capitals	field	Delhi Capitals	tie	

	city	date	player_of_match	venue	team1	team2	toss_winner	toss_decision	winner	result	result_
744	Bengaluru	2019-04-30	None	M.Chinnaswamy Stadium	Royal Challengers Bangalore	Rajasthan Royals	Rajasthan Royals	field	None	None	
746	Mumbai	2019-05-02	JJ Bumrah	Wankhede Stadium	Mumbai Indians	Sunrisers Hyderabad	Mumbai Indians	bat	Mumbai Indians	tie	
757	Dubai	2020-09-20	MP Stoinis	Dubai International Cricket Stadium	Delhi Capitals	Kings XI Punjab	Kings XI Punjab	field	Delhi Capitals	tie	
776	Abu Dhabi	2020-10-18	LH Ferguson	Sheikh Zayed Stadium	Kolkata Knight Riders	Sunrisers Hyderabad	Sunrisers Hyderabad	field	Kolkata Knight Riders	tie	
781	Dubai	2020-10-18	KL Rahul	Dubai International Cricket Stadium	Mumbai Indians	Kings XI Punjab	Mumbai Indians	bat	Kings XI Punjab	tie	
811	Dubai	2020-09-28	AB de Villiers	Dubai International Cricket Stadium	Royal Challengers Bangalore	Mumbai Indians	Mumbai Indians	field	Royal Challengers Bangalore	tie	

Rows having 'result_margin' column as null indicating that the corresponding mathes were tied

```
In [10]:   # Filling null value in 'result_margin' with 0.0
           df['result_margin']=df['result_margin'].fillna('0.0')
           df.isnull().sum()
```

Out[10]:   city   13
           date   0
           player_of_match   0
           venue   0
           team1   0
           team2   0
           toss_winner   0
           toss_decision   0
           winner   0
           result   0
           result_margin   0
           eliminator   0
           umpire1   0
           umpire2   0
           dtype: int64

Rows having city null indicating that Matches held in stadiums of Sharjah and Dubai having city column empty

```
In [11]: df[df['city'].isnull()]
```

Out[11]:

	city	date	player_of_match	venue	team1	team2	toss_winner	toss_decision	winner	result	result_margin
399	NaN	2014-04-17	YS Chahal	Sharjah Cricket Stadium	Delhi Daredevils	Royal Challengers Bangalore	Royal Challengers Bangalore	field	Royal Challengers Bangalore	wickets	8
402	NaN	2014-04-19	PA Patel	Dubai International Cricket Stadium	Royal Challengers Bangalore	Mumbai Indians	Royal Challengers Bangalore	field	Royal Challengers Bangalore	wickets	7
403	NaN	2014-04-19	JP Duminy	Dubai International Cricket Stadium	Kolkata Knight Riders	Delhi Daredevils	Kolkata Knight Riders	bat	Delhi Daredevils	wickets	4
404	NaN	2014-04-20	GJ Maxwell	Sharjah Cricket Stadium	Rajasthan Royals	Kings XI Punjab	Kings XI Punjab	field	Kings XI Punjab	wickets	7
406	NaN	2014-04-22	GJ Maxwell	Sharjah Cricket Stadium	Kings XI Punjab	Sunrisers Hyderabad	Sunrisers Hyderabad	field	Kings XI Punjab	runs	72
407	NaN	2014-04-23	RA Jadeja	Dubai International Cricket Stadium	Rajasthan Royals	Chennai Super Kings	Rajasthan Royals	field	Chennai Super Kings	runs	7
408	NaN	2014-04-24	CA Lynn	Sharjah Cricket Stadium	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore	field	Kolkata Knight Riders	runs	2
409	NaN	2014-04-25	AJ Finch	Dubai International Cricket Stadium	Sunrisers Hyderabad	Delhi Daredevils	Sunrisers Hyderabad	bat	Sunrisers Hyderabad	runs	4
410	NaN	2014-04-25	MM Sharma	Dubai International Cricket Stadium	Chennai Super Kings	Mumbai Indians	Mumbai Indians	bat	Chennai Super Kings	wickets	7
413	NaN	2014-04-27	M Vijay	Sharjah Cricket Stadium	Delhi Daredevils	Mumbai Indians	Mumbai Indians	bat	Delhi Daredevils	wickets	6

	city	date	player_of_match	venue	team1	team2	toss_winner	toss_decision	winner	result	result_margin
414	NaN	2014-04-27	DR Smith	Sharjah Cricket Stadium	Sunrisers Hyderabad	Chennai Super Kings	Sunrisers Hyderabad	bat	Chennai Super Kings	wickets	5
415	NaN	2014-04-28	Sandeep Sharma	Dubai International Cricket Stadium	Kings XI Punjab	Royal Challengers Bangalore	Kings XI Punjab	field	Kings XI Punjab	wickets	5
417	NaN	2014-04-30	B Kumar	Dubai International Cricket Stadium	Mumbai Indians	Sunrisers Hyderabad	Mumbai Indians	field	Sunrisers Hyderabad	runs	15

Replacing null value with Sharjah and Dubai according to the match venue

In [12]:
```python
df['city'][df['venue'] == 'Sharjah Cricket Stadium'] = df['city'].fillna('Sharjah')
df['city'][df['venue'] == 'Dubai International Cricket Stadium'] = df['city'].fillna('Dubai')
```

In [13]:
```python
df.isnull().sum()
# finally dataset is ready to play...!!!!!!
actual_name=['Sunrisers Hyderabad', 'Mumbai Indians', 'Gujarat Lions',
    'Rising Pune Supergiant', 'Royal Challengers Bangalore',
    'Kolkata Knight Riders', 'Delhi Daredevils', 'Kings XI Punjab',
    'Chennai Super Kings', 'Rajasthan Royals', 'Deccan Chargers',
    'Kochi Tuskers Kerala', 'Pune Warriors', 'Rising Pune Supergiants', 'Delhi Capitals']

reknowed_names = ['SRH','MI','GL','PUNE','RCB','KKR','DC','KXIP','CSK','RR','SRH','KERALA','PUNE','PUNE','DC']

df.replace(actual_name,reknowed_names,inplace = True)
```

FINALLY DATASET IS READY TO USE 🎉🎉🧨💥🧨🎉🎉

```
In [14]: #IPL
         ipl=df
         ipl.head()
```

Out[14]:

	city	date	player_of_match	venue	team1	team2	toss_winner	toss_decision	winner	result	result_margin	eliminato
0	Bangalore	2008-04-18	BB McCullum	M Chinnaswamy Stadium	RCB	KKR	RCB	field	KKR	runs	140	N
1	Chandigarh	2008-04-19	MEK Hussey	Punjab Cricket Association Stadium, Mohali	KXIP	CSK	CSK	bat	CSK	runs	33	N
2	Delhi	2008-04-19	MF Maharoof	Feroz Shah Kotla	DC	RR	RR	bat	DC	wickets	9	N
3	Mumbai	2008-04-20	MV Boucher	Wankhede Stadium	MI	RCB	MI	bat	RCB	wickets	5	N
4	Kolkata	2008-04-20	DJ Hussey	Eden Gardens	KKR	SRH	SRH	bat	KKR	wickets	5	N

```
In [43]: ipl.to_csv("IPL_CLEANED_DATASET.csv",index=False)
```

DATA VISUALIZATION

MOST WINS

```
In [15]: labels=ipl.winner.value_counts().index.tolist()

         values= ipl.winner.value_counts().tolist()
         colors={"MI":'#004BA0',"CSK":"yellow","RCB":"red","RR":"pink","DC":"#2561AE","KXIP":"#ed1f27","SRH":"orange","KK
                 "PUNE":"#9E3495","GL":"#F97324","KERALA":"#7E3F98","None":"k"}
         fig,ax=plt.subplots(figsize=(10,10))
         ax.pie(values,autopct="%.2f%%",labels=labels,pctdistance=0.85,colors=[colors[key] for key in labels])
         #drawing circle
         centre_circle=plt.Circle((0,0),radius=0.70,fc="white")
         fig=plt.gcf()
         fig.gca().add_artist(centre_circle)
         plt.title("Most wins in IPL",fontsize=17)

         plt.show()
```

Most wins in IPL

CSK 12.99%

KKR 12.13%

SRH 11.64%

RCB 11.15%

KXIP 10.78%

DC 10.54%

RR 9.93%

PUNE 3.31%

GL 1.59%

KERALA 0.74%

None

MI 14.71%
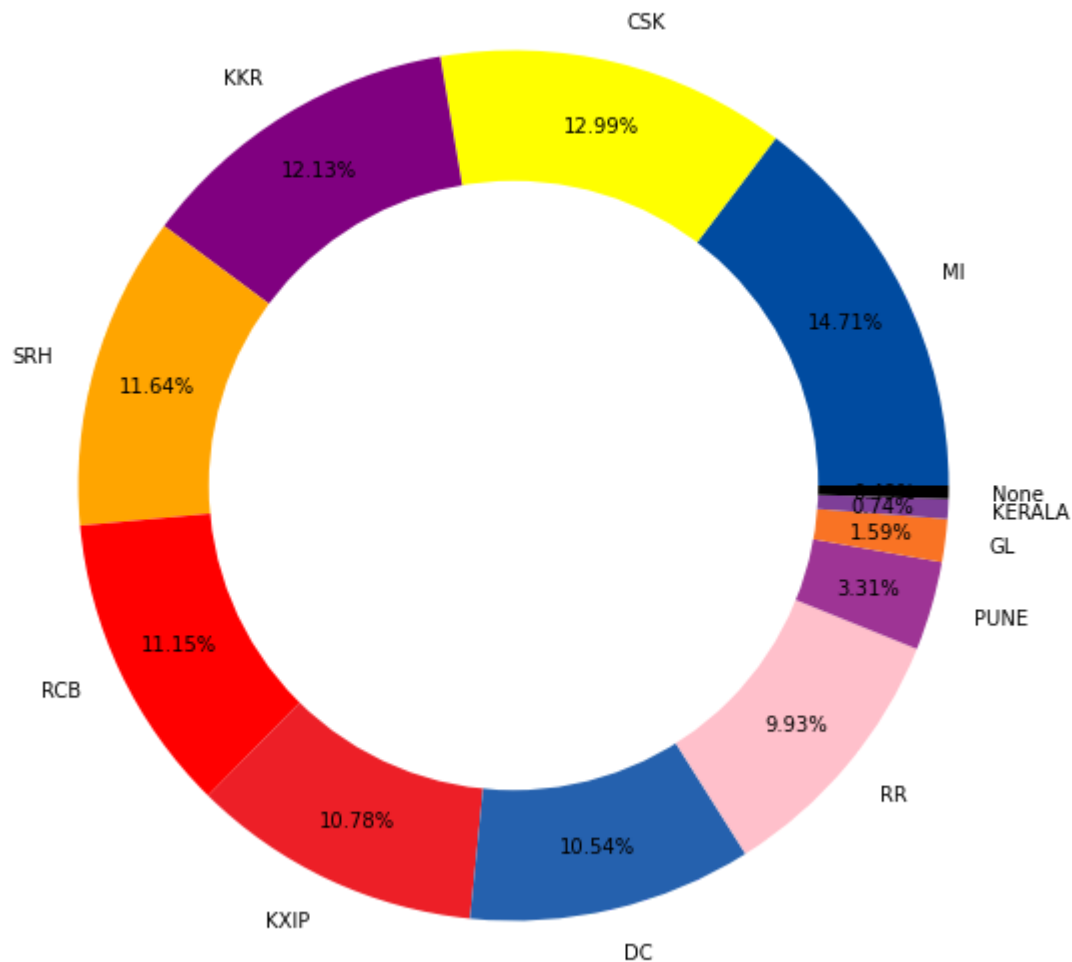
CONCLUSION:- The Mumbai Indians (MI) have registered 120 wins in 207 completed matches with 5 wins in the Finals. The team with 3 IPL trophies, Chennai Super Kings (CSK), has won 108 games out of 182 completed games. CSK did not participate in a couple of seasons and has managed to reach the three-figure mark in just 10 seasons.

Representation Using Bar Chart

```
In [16]: colors={"MI":'#004BA0',"CSK":"yellow","RCB":"red","RR":"pink","DC":"#2561AE","KXIP":"#ed1f27","SRH":"orange","KK
         "PUNE":"#9E3495","GL":"#F97324","KERALA":"#7E3F98","None":"k"}
#function to add value labels
def addlabels(labels,values):
    for i in range(len(labels)):
        plt.text(i,values[i],values[i])
plt.bar(labels,values,width=0.5,color=[colors[keys] for keys in labels])
plt.grid(False)
#calling function to add value labels
addlabels(labels,values)
#plt.set_facecolor('white')

plt.title("Most wins in IPL")
#bar[0].set_color('g')
#bar[1].set_color('c')

plt.gcf().tight_layout
plt.show()
```
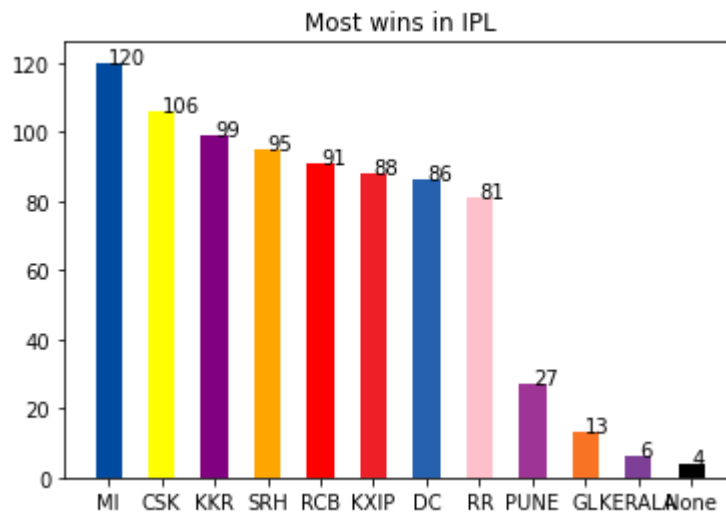

MOST WINS IN ELIMINATOR

```
In [17]: #ipl[ipl["eliminator"]=="Y"]["winner"].value_counts()
         data= ipl['winner'][ipl['eliminator']=='Y'].value_counts()
         teams=data.index
         count=data.values

         colors={"MI":'#004BA0',"CSK":"yellow","RCB":"red","RR":"pink","DC":"#2561AE","KXIP":"#ed1f27","SRH":"orange","KK
                 "PUNE":"#9E3495","GL":"#F97324","KERALA":"#7E3F98","None":"k"}
         plt.bar(teams,count,color=[colors[keys] for keys in teams])
         plt.title("Most wins in Eliminator",fontsize=22)
         plt.show()
```
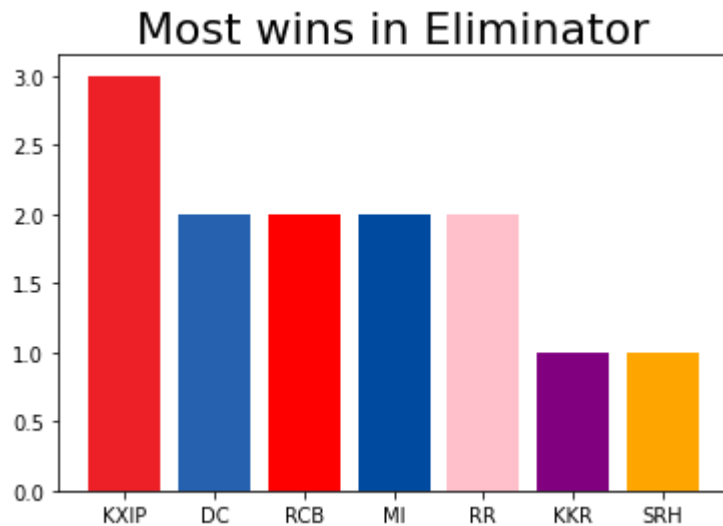

CONCLUSION:-Kings IX Punjab has won most matches(30) in Eliminator Round.

Same analysis but using seaborn countplot

```
In [18]: sns.countplot(ipl['winner'][ipl['eliminator']=='Y'],data=ipl)
         plt.title("Most wins in Eleminator",fontsize=17)
         plt.xticks(rotation=90)
         plt.show()
```

Most wins in Eleminator
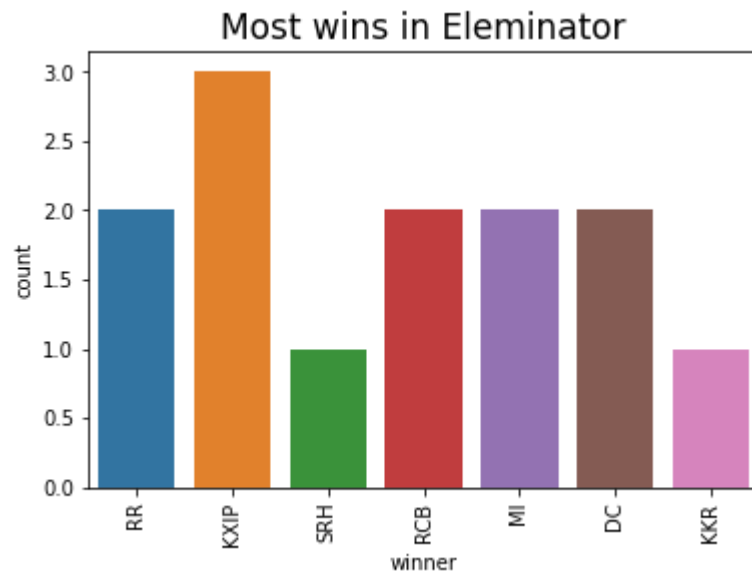
Top 10 famous venues

```
In [19]: ipl.columns
```

```
Out[19]: Index(['city', 'date', 'player_of_match', 'venue', 'team1', 'team2',
                'toss_winner', 'toss_decision', 'winner', 'result', 'result_margin',
                'eliminator', 'umpire1', 'umpire2'],
               dtype='object')
```

```
In [20]: ipl.venue.value_counts()[:10]
```

```
Out[20]: Eden Gardens                                    77
         Feroz Shah Kotla                                74
         Wankhede Stadium                                73
         M Chinnaswamy Stadium                           65
         Rajiv Gandhi International Stadium, Uppal       64
         MA Chidambaram Stadium, Chepauk                 57
         Sawai Mansingh Stadium                          47
         Punjab Cricket Association Stadium, Mohali      35
         Dubai International Cricket Stadium             33
         Sheikh Zayed Stadium                            29
         Name: venue, dtype: int64
```

```
In [21]: sns.barplot(x=ipl['venue'].value_counts().head(10).values,y=ipl['venue'].value_counts().head(10).index,data=ipl)
         plt.title("Famous Venues")
         plt.xlabel("Venue Count")
         plt.ylabel("Venue")
         plt.show()
```
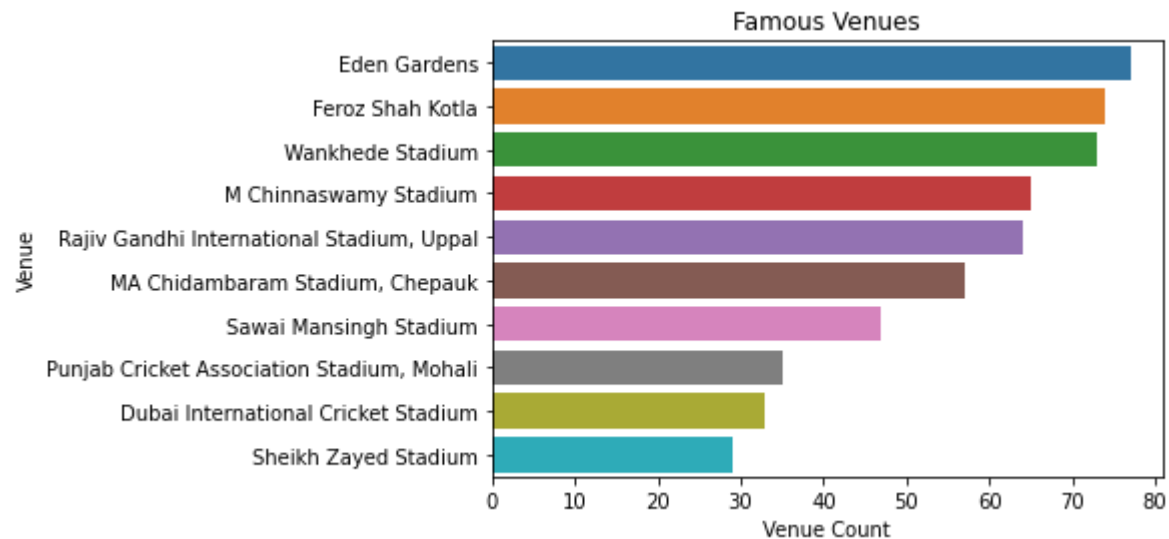

Location: Kolkata, West Bengal, India; Home club: Bengal; Establishment 1864; 157 years ago; Capacity : 80,000; Tenants : India national cricket team, Bengal cricket team, Kolkata Knight Riders,

Top 5 Umpires

```
sns.barplot(x=ipl['umpire1'].value_counts().head().index,
            y=ipl['umpire1'].value_counts().head().values,
            data=ipl)
plt.xticks(rotation=45)
plt.xlabel=("umpire 1")
plt.ylabel("Match Count")
plt.title('Top 5 Umpires')
plt.show()
```
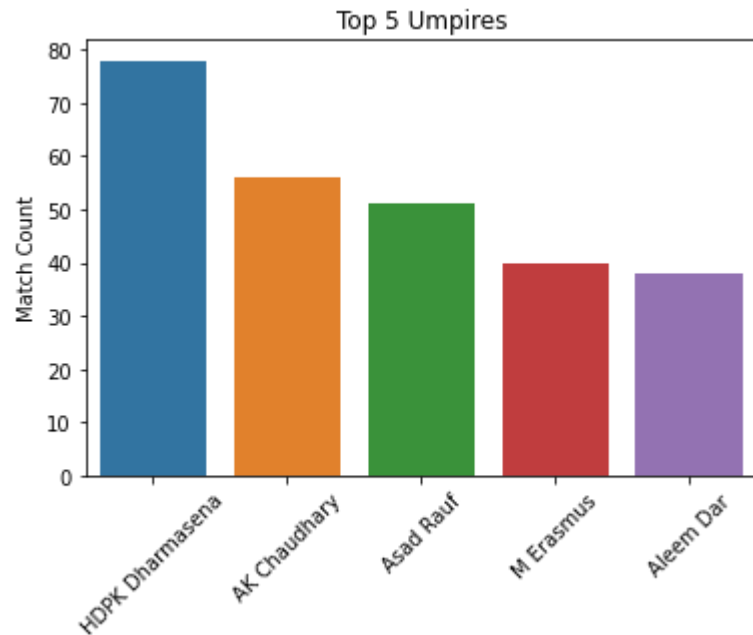

Most MOM award winning player

```
In [23]: ipl["player_of_match"].value_counts()
```

```
Out[23]: AB de Villiers    23
         CH Gayle          22
         RG Sharma         18
         DA Warner         17
         MS Dhoni          17
                           ..
         S Anirudha         1
         AP Tare            1
         A Singh            1
         AC Voges           1
         RJ Harris          1
         Name: player_of_match, Length: 234, dtype: int64
```

```python
#Top 10 players with most number of MOM award
most_mom=ipl['player_of_match'].value_counts()
most_mom=most_mom[:10,]
plt.figure(figsize=(10,7))
sns.set_style("darkgrid")
sns.barplot(most_mom.index,most_mom.values)
plt.title("Players with most MOM (top 10)")
#plt.xlabel("Players")
plt.ylabel("Number of times")
plt.xticks(rotation=90)
plt.show()
#AB de Villiers has the most Man of the Match awards.
```
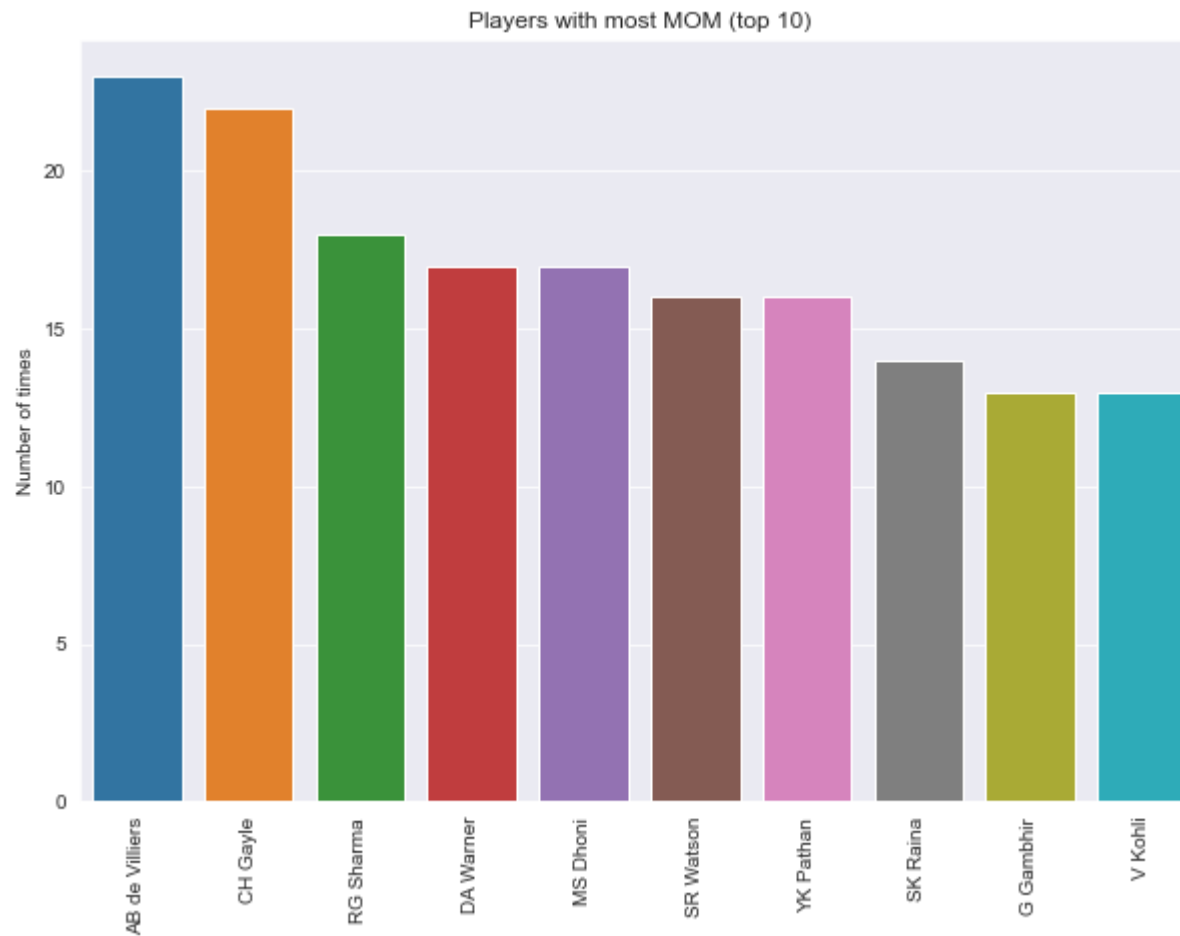
AB De Villiers: 23

Often referred to as the "Superman" of cricket, AB De Villiers has won the most number of Man of the Match awards in the league's history. He has won 23 MoM awards ever since he started playing in IPL. ABD began his IPL career with the Delhi Daredevils (now Delhi Capitals) and later on went on to represent RCB. ABD has been synonymous with RCB, as he has won a lot of matches for them. He won a few Man of the Match awards in IPL 2020 itself. This shows how impactful ABD has been for RCB.***
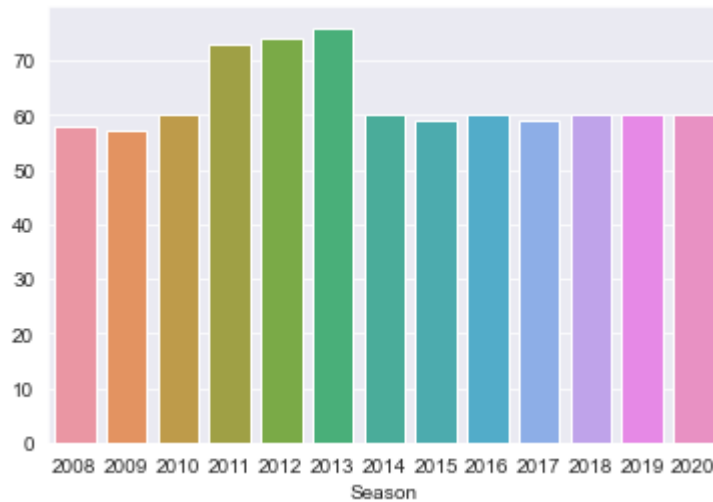
Number of matches played in various seasons :

```
In [25]: ipl['Season'] = pd.DatetimeIndex(ipl['date']).year
```

```
In [26]: #ipl.groupby("Season").size().plot(kind="bar")
         #value=new_group["Season"].count().tolist()
         #new_group
```

In [27]:
```
result=ipl.groupby("Season").size()
sns.barplot(result.index,result.values)
```

Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x12bd2d97880>

Has toss winning helped in match winning

In [28]:
```
#ipl.head(2)
```

```
In [29]:  toss_match_winners=ipl[ipl["toss_winner"]==ipl["winner"]]
          count_of_toss_match_winners=toss_match_winners.shape[0]
          count_of_toss_match_lossers=ipl.shape[0]-count_of_toss_match_winners
```

```
In [30]:  toss=plt.pie([count_of_toss_match_winners,count_of_toss_match_lossers],labels=["Won  the toss and the  match","w
                        ,autopct="%.1f%%",colors=["green","red"])
          plt.title("Has toss winning helped in match winning ??",fontsize=20)
          plt.show()
```

Has toss winning helped in match winning ??

Won the toss and the match

51.2%

48.8%

won the toss but lose the match
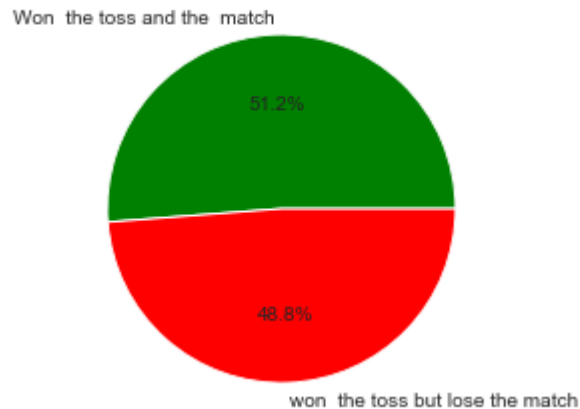
From the above plot we can derive an insight that winning the toss gives significant advantage in winning the matches. Lets check which decision makes more likely to win the matches

Overall Toss Decision (Bat or Ball)

```
#Toss decision in all the matches
plt.pie(ipl['toss_decision'].value_counts() , autopct = '%1.1f%%' , labels = ipl['toss_decision'].value_counts()
    )
plt.legend(fontsize = 10)
plt.title('Toss decision all the teams' , fontsize = 25)
plt.show()
```

Toss decision all the teams
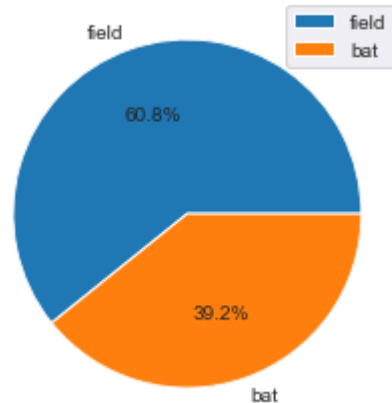
From this graph ,it is clear that 60.7 % of the times toss winning team choose to filed first and only 39.3% times choose to bat first

venues played from most to least

```
In [32]: plt.subplots(figsize=(15,8))
         ax = ipl['venue'].value_counts().sort_values(ascending=False).plot.bar(width=.9,color=sns.color_palette('infernc
         ax.set_xlabel('Grounds')
         ax.set_ylabel('count')
         plt.title("Venues played (from most to least)")
         plt.show()
```

Venues played (from most to least)
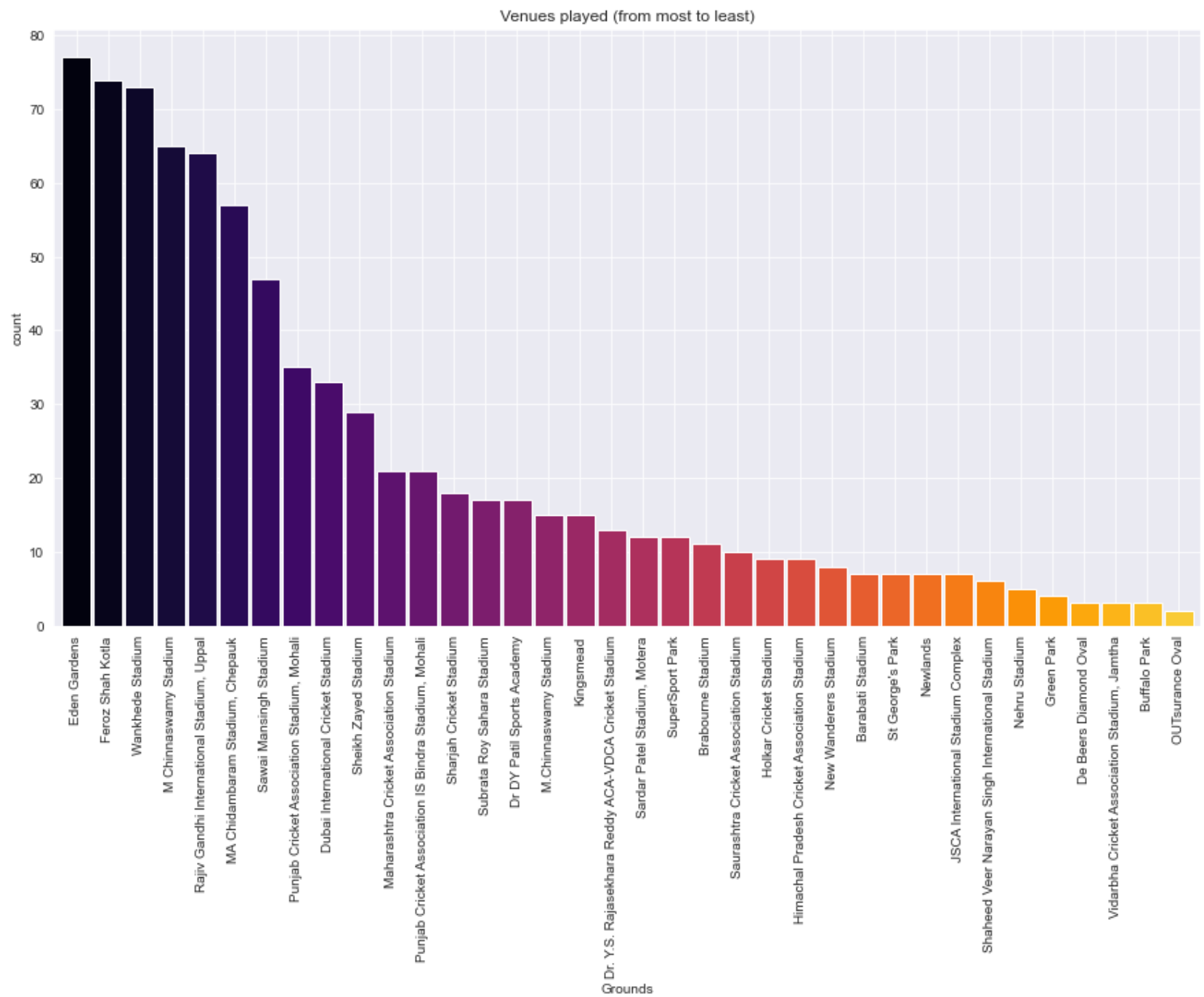
which team had won maximum matches in a single season

In [33]:
```python
#max_matches=ipl.groupby("Season")
#final=max_matches["winner"].value_counts()
#value=final.values
#print(value)
#lable=final.index
#print(lable[0][1])
#plt.bar(lable,value)
#res=max_matches[max_matches["winner"].value_counts().max()]
#res = res.reset_index()
#res
```

In [34]:
```python
pip install bar_chart_race
```

```
Requirement already satisfied: bar_chart_race in c:\users\krishna\anaconda3\lib\site-packages (0.1.0)Note: you
may need to restart the kernel to use updated packages.
Requirement already satisfied: matplotlib>=3.1 in c:\users\krishna\anaconda3\lib\site-packages (from bar_chart
_race) (3.2.2)
Requirement already satisfied: pandas>=0.24 in c:\users\krishna\anaconda3\lib\site-packages (from bar_chart_ra
ce) (1.0.5)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\users\krishna\anaconda3\lib\site
-packages (from matplotlib>=3.1->bar_chart_race) (2.4.7)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\krishna\anaconda3\lib\site-packages (from matp
lotlib>=3.1->bar_chart_race) (2.8.1)
Requirement already satisfied: cycler>=0.10 in c:\users\krishna\anaconda3\lib\site-packages (from matplotlib>=
3.1->bar_chart_race) (0.10.0)
Requirement already satisfied: numpy>=1.11 in c:\users\krishna\anaconda3\lib\site-packages (from matplotlib>=
3.1->bar_chart_race) (1.18.5)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\krishna\anaconda3\lib\site-packages (from matplot
lib>=3.1->bar_chart_race) (1.2.0)
Requirement already satisfied: pytz>=2017.2 in c:\users\krishna\anaconda3\lib\site-packages (from pandas>=0.24
->bar_chart_race) (2020.1)
Requirement already satisfied: six>=1.5 in c:\users\krishna\anaconda3\lib\site-packages (from python-dateutil>
=2.1->matplotlib>=3.1->bar_chart_race) (1.15.0)
```

`In [35]:` `ipl.head()`

`Out[35]:`

	city	date	player_of_match	venue	team1	team2	toss_winner	toss_decision	winner	result	result_margin	eliminator
0	Bangalore	2008-04-18	BB McCullum	M Chinnaswamy Stadium	RCB	KKR	RCB	field	KKR	runs	140	N
1	Chandigarh	2008-04-19	MEK Hussey	Punjab Cricket Association Stadium, Mohali	KXIP	CSK	CSK	bat	CSK	runs	33	N
2	Delhi	2008-04-19	MF Maharoof	Feroz Shah Kotla	DC	RR	RR	bat	DC	wickets	9	N
3	Mumbai	2008-04-20	MV Boucher	Wankhede Stadium	MI	RCB	MI	bat	RCB	wickets	5	N
4	Kolkata	2008-04-20	DJ Hussey	Eden Gardens	KKR	SRH	SRH	bat	KKR	wickets	5	N

```
In [37]: ipl.date=pd.to_datetime(ipl["date"],format="%Y-%m-%d")
         #ipl.winner=ipl['winner'].astype(str)
         ipl.dtypes
```

```
Out[37]: city                    object
         date              datetime64[ns]
         player_of_match         object
         venue                   object
         team1                   object
         team2                   object
         toss_winner             object
         toss_decision           object
         winner                  object
         result                  object
         result_margin           object
         eliminator              object
         umpire1                 object
         umpire2                 object
         Season                   int64
         dtype: object
```

```
In [38]: date = ipl.date.tolist()
         date.append(0)
         winners=ipl.winner

         teams={"MI":[0],"RCB":[0],"CSK":[0],"KKR":[0],"SRH":[0],"DC":[0],"KXIP":[0],"RR":[0],"PUNE":[0],"GL":[0],"KERALA
         print(len(date))
```

```
817
```

```
In [39]: for i in winners:
             for j in teams.keys():
                 if i==j:
                     teams[i].append(teams[i][-1] +1)
                     continue
                 teams[j].append(teams[j][-1] +0)

         #print(teams)
```

```
In [40]:  running_df = pd.DataFrame.from_dict(teams)
          #running_df.tail()
          #running_df.index
          running_df.shape
          running_df["date"]=date
          running_df.head()
          mostwins=running_df.set_index('date')
```

```
In [44]:  !pip install fsspec
```

Requirement already satisfied: fsspec in c:\users\krishna\anaconda3\lib\site-packages (0.7.4)

```
In [46]:  import bar_chart_race as bcr
          #bcr.bar_chart_race(df=mostwins,title="most wins")
```

OUTPUT IS AVAILABLE IN BELOW LINK ✓✓:

https://colab.research.google.com/drive/1LZhgw7ccMD9Lu7e4d5Q9_ltXuM4ykYTn?usp=sharing
(https://colab.research.google.com/drive/1LZhgw7ccMD9Lu7e4d5Q9_ltXuM4ykYTn?usp=sharing)

```
In [ ]:
```

```
In [ ]:  #THE END
```

```
In [ ]:
```