

Assignment4

October 10, 2020

1 Assignment 4

Before working on this assignment please read these instructions fully. In the submission area, you will notice that you can click the link to **Preview the Grading** for each step of the assignment. This is the criteria that will be used for peer grading. Please familiarize yourself with the criteria before beginning the assignment.

This assignment requires that you to find **at least** two datasets on the web which are related, and that you visualize these datasets to answer a question with the broad topic of **economic activity or measures** (see below) for the region of **Ann Arbor, Michigan, United States**, or **United States** more broadly.

You can merge these datasets with data from different regions if you like! For instance, you might want to compare **Ann Arbor, Michigan, United States** to Ann Arbor, USA. In that case at least one source file must be about **Ann Arbor, Michigan, United States**.

You are welcome to choose datasets at your discretion, but keep in mind **they will be shared with your peers**, so choose appropriate datasets. Sensitive, confidential, illicit, and proprietary materials are not good choices for datasets for this assignment. You are welcome to upload datasets of your own as well, and link to them using a third party repository such as github, bitbucket, pastebin, etc. Please be aware of the Coursera terms of service with respect to intellectual property.

Also, you are welcome to preserve data in its original language, but for the purposes of grading you should provide english translations. You are welcome to provide multiple visuals in different languages if you would like!

As this assignment is for the whole course, you must incorporate principles discussed in the first week, such as having as high data-ink ratio (Tufte) and aligning with Cairo's principles of truth, beauty, function, and insight.

Here are the assignment instructions:

- State the region and the domain category that your data sets are about (e.g., **Ann Arbor, Michigan, United States** and **economic activity or measures**).
- You must state a question about the domain category and region that you identified as being interesting.
- You must provide at least two links to available datasets. These could be links to files such as CSV or Excel files, or links to websites which might have data in tabular form, such as Wikipedia pages.
- You must upload an image which addresses the research question you stated. In addition to addressing the question, this visual should follow Cairo's principles of truthfulness, functionality, beauty, and insightfulness.

- You must contribute a short (1-2 paragraph) written justification of how your visualization addresses your stated research question.

What do we mean by **economic activity or measures**? For this category you might look at the inputs or outputs to the given economy, or major changes in the economy compared to other regions.

1.1 Tips

- Wikipedia is an excellent source of data, and I strongly encourage you to explore it for new data sources.
- Many governments run open data initiatives at the city, region, and country levels, and these are wonderful resources for localized data sources.
- Several international agencies, such as the [United Nations](#), the [World Bank](#), the [Global Open Data Index](#) are other great places to look for data.
- This assignment requires you to convert and clean datafiles. Check out the discussion forums for tips on how to do this from various sources, and share your successes with your fellow students!

1.2 Example

Looking for an example? Here's what our course assistant put together for the **Ann Arbor, MI, USA** area using **sports and athletics** as the topic. [Example Solution File](#)

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.ticker as ticker
```

```
%matplotlib inline
```

```
In [2]: df =pd.read_csv('ipl_dataset.csv')
df.head()
```

```
Out[2]:
```

	id	season	city	date	team1 \
0	1	2017	Hyderabad	2017-04-05	Sunrisers Hyderabad
1	2	2017	Pune	2017-04-06	Mumbai Indians
2	3	2017	Rajkot	2017-04-07	Gujarat Lions
3	4	2017	Indore	2017-04-08	Rising Pune Supergiant
4	5	2017	Bangalore	2017-04-08	Royal Challengers Bangalore

	team2	toss_winner	toss_decision
0	Royal Challengers Bangalore	Royal Challengers Bangalore	field
1	Rising Pune Supergiant	Rising Pune Supergiant	field
2	Kolkata Knight Riders	Kolkata Knight Riders	field
3	Kings XI Punjab	Kings XI Punjab	field
4	Delhi Daredevils	Royal Challengers Bangalore	bat

	result	dl_applied	winner	win_by_runs	\
0	normal	0	Sunrisers Hyderabad	35	
1	normal	0	Rising Pune Supergiant	0	
2	normal	0	Kolkata Knight Riders	0	
3	normal	0	Kings XI Punjab	0	
4	normal	0	Royal Challengers Bangalore	15	

	win_by_wickets	player_of_match	venue
0	0	Yuvraj Singh	Rajiv Gandhi International Stadium, Uppala
1	7	SPD Smith	Maharashtra Cricket Association Stadium
2	10	CA Lynn	Saurashtra Cricket Association Stadium
3	6	GJ Maxwell	Holkar Cricket Stadium
4	0	KM Jadhav	M Chinnaswamy Stadium

	umpire1	umpire2	umpire3
0	AY Dandekar	NJ Llong	NaN
1	A Nand Kishore	S Ravi	NaN
2	Nitin Menon	CK Nandan	NaN
3	AK Chaudhary	C Shamshuddin	NaN
4	NaN	NaN	NaN

```
In [5]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 636 entries, 0 to 635
Data columns (total 18 columns):
id                636 non-null int64
season            636 non-null int64
city              629 non-null object
date              636 non-null object
team1             636 non-null object
team2             636 non-null object
toss_winner       636 non-null object
toss_decision     636 non-null object
result            636 non-null object
dl_applied        636 non-null int64
winner            633 non-null object
win_by_runs       636 non-null int64
win_by_wickets    636 non-null int64
player_of_match   633 non-null object
venue             636 non-null object
umpire1           635 non-null object
umpire2           635 non-null object
umpire3           0 non-null float64
dtypes: float64(1), int64(5), object(12)
memory usage: 89.5+ KB
```

```
In [6]: df['venue'].value_counts()
```

```

Out [6]: M Chinnaswamy Stadium 66
         Eden Gardens 61
         Feroz Shah Kotla 60
         Wankhede Stadium 57
         Rajiv Gandhi International Stadium, Uppal 49
         MA Chidambaram Stadium, Chepauk 48
         Punjab Cricket Association Stadium, Mohali 35
         Sawai Mansingh Stadium 33
         Subrata Roy Sahara Stadium 17
         Dr DY Patil Sports Academy 17
         Kingsmead 15
         Maharashtra Cricket Association Stadium 15
         SuperSport Park 12
         Sardar Patel Stadium, Motera 12
         Punjab Cricket Association IS Bindra Stadium, Mohali 11
         Dr. Y.S. Rajasekhara Reddy ACA-VDCA Cricket Stadium 11
         Brabourne Stadium 11
         Saurashtra Cricket Association Stadium 10
         Himachal Pradesh Cricket Association Stadium 9
         New Wanderers Stadium 8
         Newlands 7
         St George's Park 7
         Sheikh Zayed Stadium 7
         Barabati Stadium 7
         Dubai International Cricket Stadium 7
         JSCA International Stadium Complex 7
         Shaheed Veer Narayan Singh International Stadium 6
         Sharjah Cricket Stadium 6
         Holkar Cricket Stadium 5
         Nehru Stadium 5
         Green Park 4
         Buffalo Park 3
         De Beers Diamond Oval 3
         Vidarbha Cricket Association Stadium, Jamtha 3
         OUTsurance Oval 2
         Name: venue, dtype: int64

In [7]: df['team1'].value_counts()
         df['team1']=df['team1'].replace('Rising Pune Supergiant','Rising Pune Super

         df['team2'].value_counts()
         df['team2']=df['team2'].replace('Rising Pune Supergiant','Rising Pune Super

         df['winner'].value_counts()
         df['winner']=df['winner'].replace('Rising Pune Supergiant','Rising Pune Sup

In [8]: df2=df.copy()

         df2=df2.sort(['season'],ascending=True)

```

```
/opt/conda/lib/python3.6/site-packages/ipykernel/__main__.py:3: FutureWarning: sort
app.launch_new_instance()
```

```
In [9]: df2['season'].value_counts()
df2.tail()
```

```
Out[9]:
```

	id	season	city	date	team1 \
34	35	2017	Chandigarh	2017-04-30	Delhi Daredevils
33	34	2017	Rajkot	2017-04-29	Gujarat Lions
32	33	2017	Pune	2017-04-29	Rising Pune Supergiants
44	45	2017	Bangalore	2017-05-07	Royal Challengers Bangalore
0	1	2017	Hyderabad	2017-04-05	Sunrisers Hyderabad

	team2	toss_winner	toss_decision
34	Kings XI Punjab	Kings XI Punjab	field
33	Mumbai Indians	Gujarat Lions	bat
32	Royal Challengers Bangalore	Royal Challengers Bangalore	field
44	Kolkata Knight Riders	Kolkata Knight Riders	field
0	Royal Challengers Bangalore	Royal Challengers Bangalore	field

	result	dl_applied	winner	win_by_runs	win_by_wicket
34	normal	0	Kings XI Punjab	0	1
33	tie	0	Mumbai Indians	0	
32	normal	0	Rising Pune Supergiants	61	
44	normal	0	Kolkata Knight Riders	0	
0	normal	0	Sunrisers Hyderabad	35	

	player_of_match	venue \
34	Sandeep Sharma	Punjab Cricket Association IS Bindra Stadium, ...
33	KH Pandya	Saurashtra Cricket Association Stadium
32	LH Ferguson	Maharashtra Cricket Association Stadium
44	SP Narine	M Chinnaswamy Stadium
0	Yuvraj Singh	Rajiv Gandhi International Stadium, Uppal

	umpire1	umpire2	umpire3
34	YC Barde	CK Nandan	NaN
33	AK Chaudhary	CB Gaffaney	NaN
32	KN Ananthapadmanabhan	M Erasmus	NaN
44	AY Dandekar	C Shamshuddin	NaN
0	AY Dandekar	NJ Llong	NaN

```
In [10]: dl=df2[df2['dl_applied']==1][['team1','team2','winner','toss_decision']]
dl
```

```
Out[10]:
```

	team1	team2 \
102	Kolkata Knight Riders	Chennai Super Kings
99	Delhi Daredevils	Kings XI Punjab
148	Chennai Super Kings	Kings XI Punjab

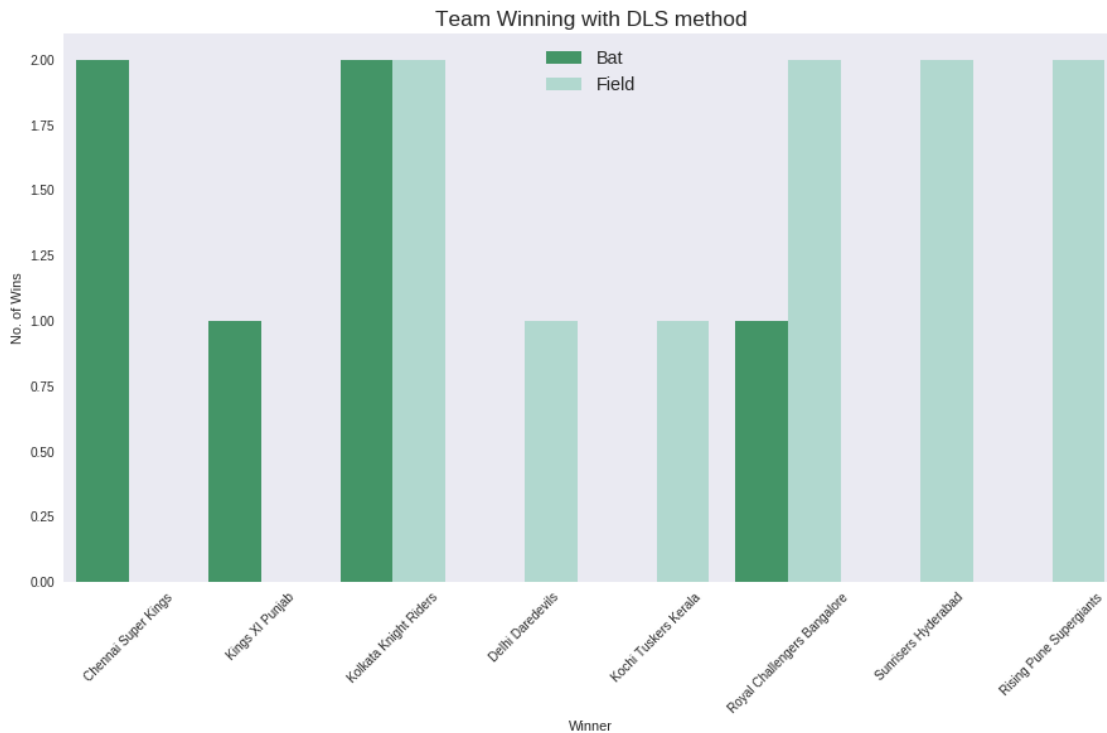
122	Kings XI Punjab	Kolkata Knight Riders
119	Kings XI Punjab	Delhi Daredevils
251	Chennai Super Kings	Kochi Tuskers Kerala
290	Kolkata Knight Riders	Royal Challengers Bangalore
280	Chennai Super Kings	Kolkata Knight Riders
488	Delhi Daredevils	Sunrisers Hyderabad
536	Sunrisers Hyderabad	Kolkata Knight Riders
567	Sunrisers Hyderabad	Royal Challengers Bangalore
597	Sunrisers Hyderabad	Rising Pune Supergiants
625	Royal Challengers Bangalore	Kings XI Punjab
624	Delhi Daredevils	Rising Pune Supergiants
620	Rising Pune Supergiants	Kolkata Knight Riders
56	Sunrisers Hyderabad	Kolkata Knight Riders

	winner	toss_decision
102	Chennai Super Kings	bat
99	Kings XI Punjab	bat
148	Chennai Super Kings	bat
122	Kolkata Knight Riders	field
119	Delhi Daredevils	field
251	Kochi Tuskers Kerala	field
290	Royal Challengers Bangalore	field
280	Kolkata Knight Riders	bat
488	Sunrisers Hyderabad	field
536	Sunrisers Hyderabad	field
567	Royal Challengers Bangalore	bat
597	Rising Pune Supergiants	field
625	Royal Challengers Bangalore	field
624	Rising Pune Supergiants	field
620	Kolkata Knight Riders	bat
56	Kolkata Knight Riders	field

```
In [11]: sns.set_style('dark')
plt.figure(figsize=(15,8))
grid=sns.countplot(x='winner',hue='toss_decision',data=dl,palette='BuGn_r')
sns.set_context("poster", font_scale = 0.9)
plt.xlabel('Winner')
plt.ylabel('No. of Wins')
plt.title('Team Winning with DLS method')
x=plt.gca().xaxis
for item in x.get_ticklabels():
    item.set_rotation(45)

plt.legend(['Bat','Field'])

plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
```



: # Kolkata Knight riders won the maximum number of times with DLS method

1.3 2-Batting First

1.4 2-Bowling First

```
In [12]: df3=df[['season','team1','team2' , 'city','winner','venue','toss_decision']]
          df3.head()
          df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 636 entries, 0 to 635
Data columns (total 7 columns):
season          636 non-null int64
team1           636 non-null object
team2           636 non-null object
city            629 non-null object
winner          633 non-null object
venue           636 non-null object
toss_decision   636 non-null object
dtypes: int64(1), object(6)
memory usage: 34.9+ KB
```

```

In [13]: sns.set_style('dark')
plt.figure(figsize=(15,8))
ax=sns.countplot(x='winner',hue='toss_decision',data=df3,palette='GnBu_d',
sns.set_context("poster", font_scale = 0.6)
plt.xlabel('Winner')
plt.ylabel('Number of wins')
plt.title('Maximum No of winning teams')
x=plt.gca().xaxis
for item in x.get_ticklabels():
    item.set_rotation(45)

plt.legend(['Bat','Field'])

plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)

# Make twin axis
#ax2=ax.twinx()

# Switch so count axis is on right, frequency on left
#ax2.yaxis.tick_left()
ax.yaxis.tick_left()

# Also switch the labels over
ax.yaxis.set_label_position('left')
#ax2.yaxis.set_label_position('left')

#ax2.set_ylabel('Frequency [%]')
ncount = 60
for p in ax.patches:
    x=p.get_bbox().get_points()[:,0]
    y=p.get_bbox().get_points()[1,1]
    ax.annotate('{:.0f}'.format(y),(x.mean(), y),ha='center', va='bottom')

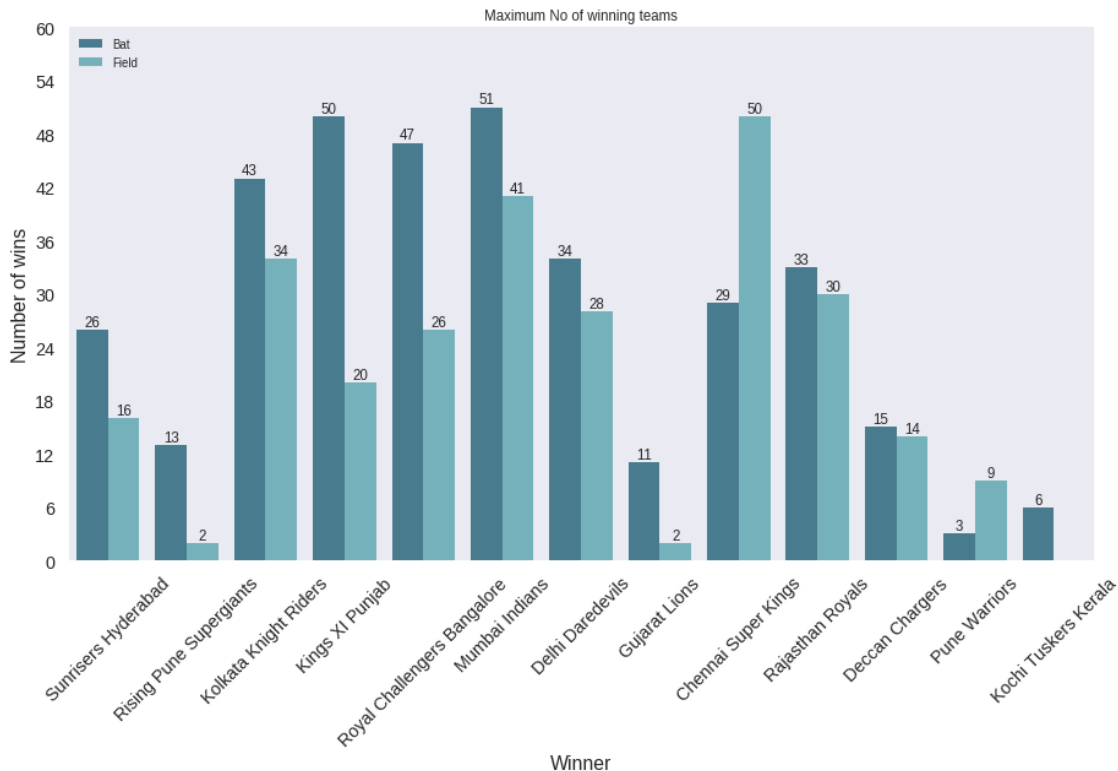
# Use a LinearLocator to ensure the correct number of ticks
ax.yaxis.set_major_locator(ticker.LinearLocator(11))

# Fix the frequency range to 0-100
#ax2.set_ylim(0,50)
ax.set_ylim(0,ncount)

# And use a MultipleLocator to ensure a tick spacing of 10
#ax2.yaxis.set_major_locator(ticker.MultipleLocator(10))

```

Out[13]: (0, 60)



2 Mumbai Indians has Won Maximum Number of matches (91 matches) overall in IPL

```
In [14]: df4=df[['umpire1','umpire2','winner']]
```

```
In [15]: df4['umpire1'].value_counts().head()
```

```
Out[15]: HDPK Dharmasena      73
         Asad Rauf             51
         AK Chaudhary          43
         Aleem Dar             38
         BF Bowden             37
         Name: umpire1, dtype: int64
```

```
In [16]: sns.set_style('dark')
plt.figure(figsize=(15,8))
ax=sns.countplot(x='umpire1',data=df4,palette=('cubehelix'),order=df['umpire1'].value_counts().index)
sns.set_context("poster", font_scale = 0.9)
plt.xlabel('Umpire1 ')
plt.ylabel('Number of time featured in matches ')
plt.title('Total Umpire Count for matches')
x=plt.gca().xaxis
```

```

for item in x.get_ticklabels():
    item.set_rotation(45)

#plt.legend(['Bat', 'Field'])

plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)

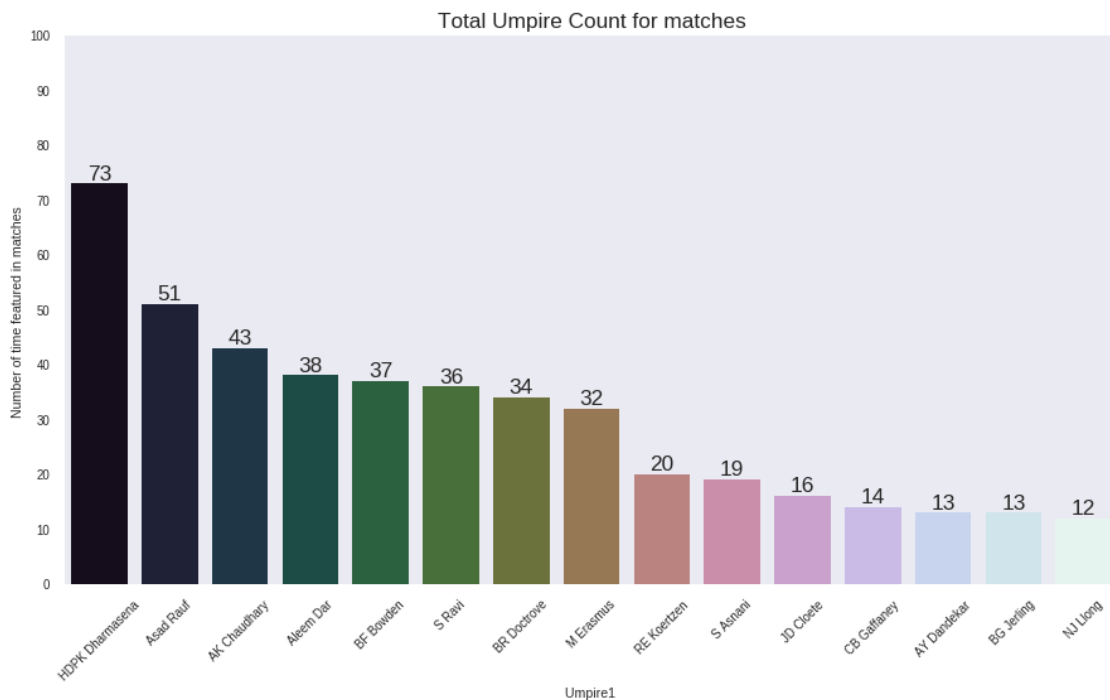
ax.yaxis.tick_left()

ax.yaxis.set_label_position('left')
ncount = 100
for p in ax.patches:
    x=p.get_bbox().get_points()[:,0]
    y=p.get_bbox().get_points()[1,1]
    ax.annotate('{:.0f}'.format(y), (x.mean(), y), ha='center', va='bottom')

ax.yaxis.set_major_locator(ticker.LinearLocator(11))
ax.set_ylim(0,ncount)

```

Out[16]: (0, 100)



In [17]: df4['umpire2'].value_counts().head()

```
Out[17]: SJA Taufel      54
         C Shamshuddin  50
         S Ravi         49
         RJ Tucker      38
         CK Nandan      36
         Name: umpire2, dtype: int64
```

```
In [18]: sns.set_style('dark')
plt.figure(figsize=(15,8))
ax=sns.countplot(x='umpire2',data=df4,palette=('cubehelix'),order=df['umpi
sns.set_context("poster", font_scale = 0.9)
plt.xlabel('Umpire2 position')
plt.ylabel('Number of time featured in matches ')
plt.title('Total Umpire Count for matches')
x=plt.gca().xaxis
for item in x.get_ticklabels():
    item.set_rotation(45)

#plt.legend(['Bat','Field'])

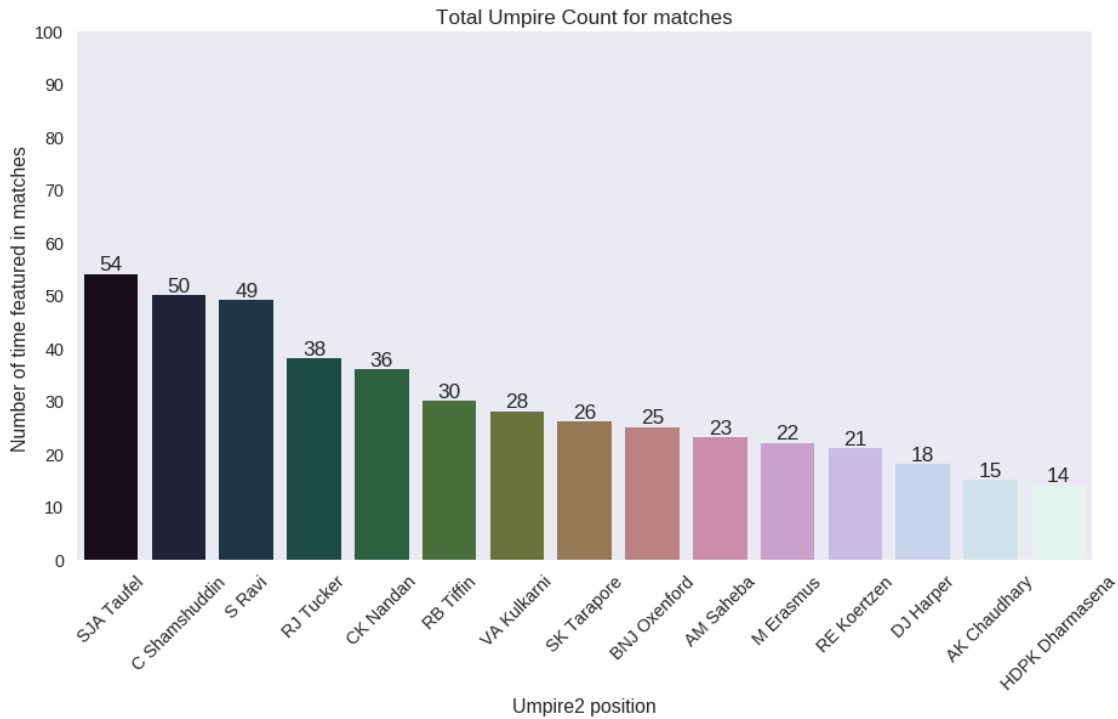
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)

ax.yaxis.tick_left()

ax.yaxis.set_label_position('left')
ncount = 100
for p in ax.patches:
    x=p.get_bbox().get_points()[0,0]
    y=p.get_bbox().get_points()[1,1]
    ax.annotate('{:.0f}'.format(y),(x.mean(),y),ha='center',va='bottom')

ax.yaxis.set_major_locator(ticker.LinearLocator(11))
ax.set_ylim(0,ncount)
```

```
Out[18]: (0, 100)
```



```
In [19]: u1=df4[(df4['umpire1']=='HDPK Dharmasena') | (df4['umpire2']=='HDPK Dhar
df_ump=df4['winner'].iloc[u1]
print(len(df_ump))
```

87

```
In [20]: df_ump.value_counts()
```

```
Out[20]: Mumbai Indians      17
Kings XI Punjab             14
Royal Challengers Bangalore  11
Chennai Super Kings         10
Sunrisers Hyderabad         8
Delhi Daredevils            7
Rajasthan Royals            6
Deccan Chargers            5
Kolkata Knight Riders       5
Rising Pune Supergiants     2
Gujarat Lions              1
Name: winner, dtype: int64
```

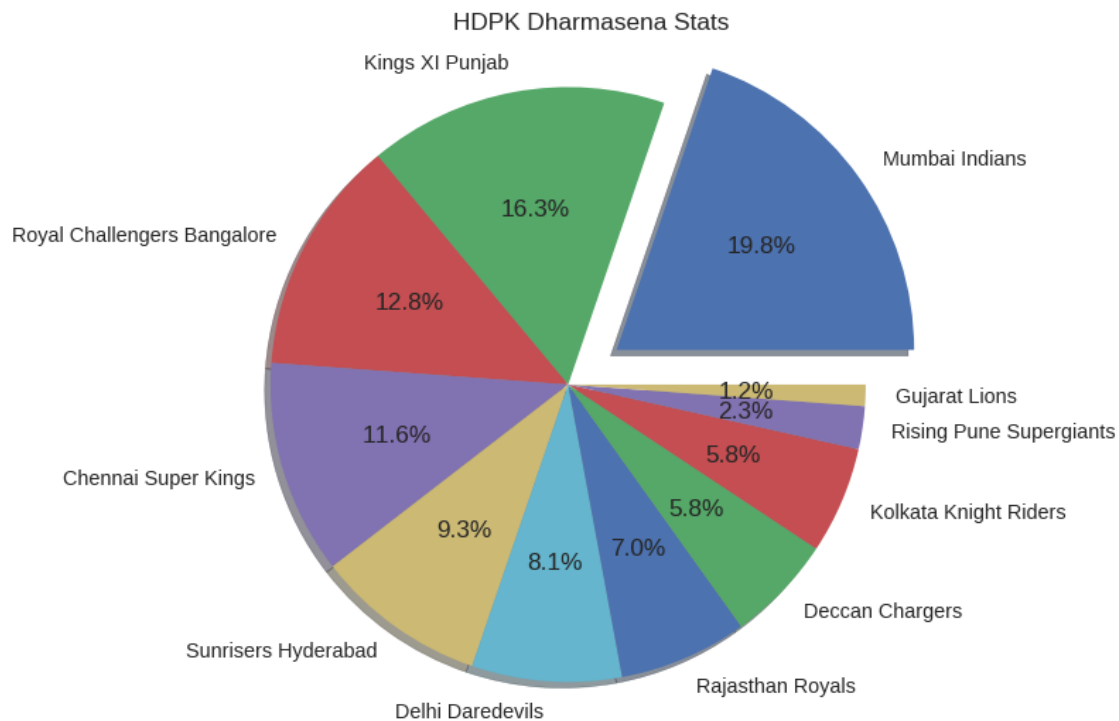
```
In [21]: labels = ('Mumbai Indians', 'Kings XI Punjab', 'Royal Challengers Bangalor
sizes = [17, 14,11,10,8,7,6,5,5,2,1]
```

```

#colors = ['lightskyblue', 'yellowgreen', 'lightcoral', 'lightskyblue']
explode = (0.2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) # explode 1st slice ie how mu

# Plot
plt.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%', shadow=True)
plt.axis('equal')
plt.title('HDPK Dharmasena Stats')
plt.show()

```



3 HDPK Dharmasena when featured as umpire1 or umpire2 it's seen that Mumbai Indians has one the maximum number of times following Kings XI Punjab

```

In [22]: u2=df4[(df4['umpire2']=='SJA Taufel') | (df4['umpire2']=='SJA Taufel')]

df_ump=df4['winner'].iloc[u2]
print(len(df_ump))

```

54

```

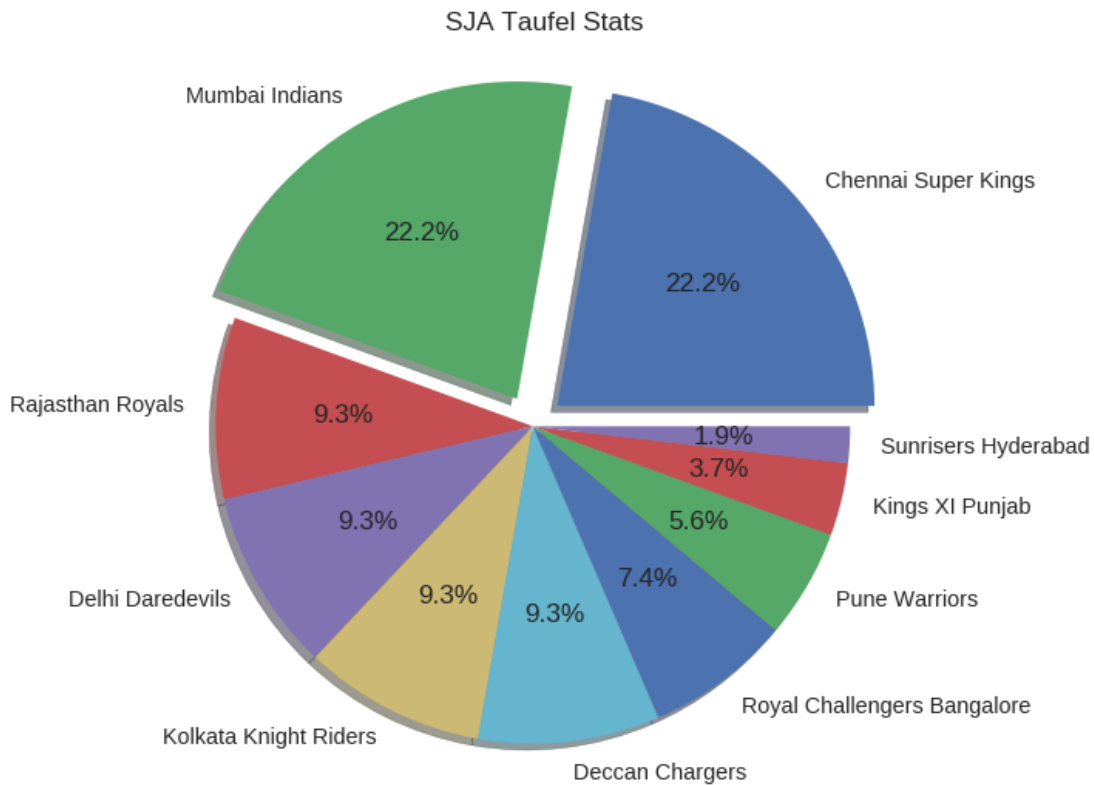
In [23]: df_ump.value_counts()

```

```
Out[23]: Mumbai Indians          12
Chennai Super Kings             12
Delhi Daredevils                 5
Rajasthan Royals                 5
Deccan Chargers                 5
Kolkata Knight Riders            5
Royal Challengers Bangalore      4
Pune Warriors                    3
Kings XI Punjab                 2
Sunrisers Hyderabad             1
Name: winner, dtype: int64
```

```
In [24]: labels = ('Chennai Super Kings', 'Mumbai Indians', 'Rajasthan Royals', 'De
sizes = [12,12,5,5,5,5,4,3,2,1]
explode = (0.1, 0.1,0,0,0, 0, 0, 0, 0, 0) # explode 1st slice ie how much

# Plot
plt.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%', shadow=True)
plt.axis('equal')
plt.title('SJA Taufel Stats')
plt.show()
```



4 SJA Taufel when featured as umpire1 or umpire2 it's seen that Mumbai Indians and Chennai Super Kings has won the maximum number of times.

```
In [25]: df5=df[['city','toss_decision','winner']]
         df5.head()
```

```
Out[25]:
```

	city	toss_decision	winner
0	Hyderabad	field	Sunrisers Hyderabad
1	Pune	field	Rising Pune Supergiants
2	Rajkot	field	Kolkata Knight Riders
3	Indore	field	Kings XI Punjab
4	Bangalore	bat	Royal Challengers Bangalore

```
In [26]: df['city'].value_counts()
```

```
Out[26]:
```

Mumbai	85
Bangalore	66
Kolkata	61
Delhi	60
Hyderabad	49
Chennai	48
Chandigarh	46
Jaipur	33
Pune	32
Durban	15
Centurion	12
Ahmedabad	12
Visakhapatnam	11
Rajkot	10
Dharamsala	9
Johannesburg	8
Ranchi	7
Abu Dhabi	7
Cape Town	7
Port Elizabeth	7
Cuttack	7
Raipur	6
Sharjah	6
Indore	5
Kochi	5
Kanpur	4
Nagpur	3
East London	3
Kimberley	3
Bloemfontein	2

Name: city, dtype: int64

```
In [27]: x = df5.groupby('city').count()
x.reset_index()
```

```
Out[27]:
```

	city	toss_decision	winner
0	Abu Dhabi	7	7
1	Ahmedabad	12	12
2	Bangalore	66	64
3	Bloemfontein	2	2
4	Cape Town	7	7
5	Centurion	12	12
6	Chandigarh	46	46
7	Chennai	48	48
8	Cuttack	7	7
9	Delhi	60	59
10	Dharamsala	9	9
11	Durban	15	15
12	East London	3	3
13	Hyderabad	49	49
14	Indore	5	5
15	Jaipur	33	33
16	Johannesburg	8	8
17	Kanpur	4	4
18	Kimberley	3	3
19	Kochi	5	5
20	Kolkata	61	61
21	Mumbai	85	85
22	Nagpur	3	3
23	Port Elizabeth	7	7
24	Pune	32	32
25	Raipur	6	6
26	Rajkot	10	10
27	Ranchi	7	7
28	Sharjah	6	6
29	Visakhapatnam	11	11

```
In [28]: x=x.where(x['toss_decision']>10).sort_values('toss_decision',axis=0,ascending=False)
x=x.reset_index()
x=x.dropna()
x
```

```
Out[28]:
```

	city	toss_decision	winner
0	Mumbai	85.0	85.0
1	Bangalore	66.0	64.0
2	Kolkata	61.0	61.0
3	Delhi	60.0	59.0
4	Hyderabad	49.0	49.0
5	Chennai	48.0	48.0
6	Chandigarh	46.0	46.0

7	Jaipur	33.0	33.0
8	Pune	32.0	32.0
9	Durban	15.0	15.0
10	Ahmedabad	12.0	12.0
11	Centurion	12.0	12.0
12	Visakhapatnam	11.0	11.0

```
In [29]: sns.set_style('dark')
ax=sns.barplot(x='city',y='toss_decision',data=x,palette='BuGn_r')
plt.xlabel('Cities')
plt.ylabel('Total number of matches played')
plt.title('Matches played in cities(Minimum 10)')
sns.set_context("poster", font_scale = 1)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)

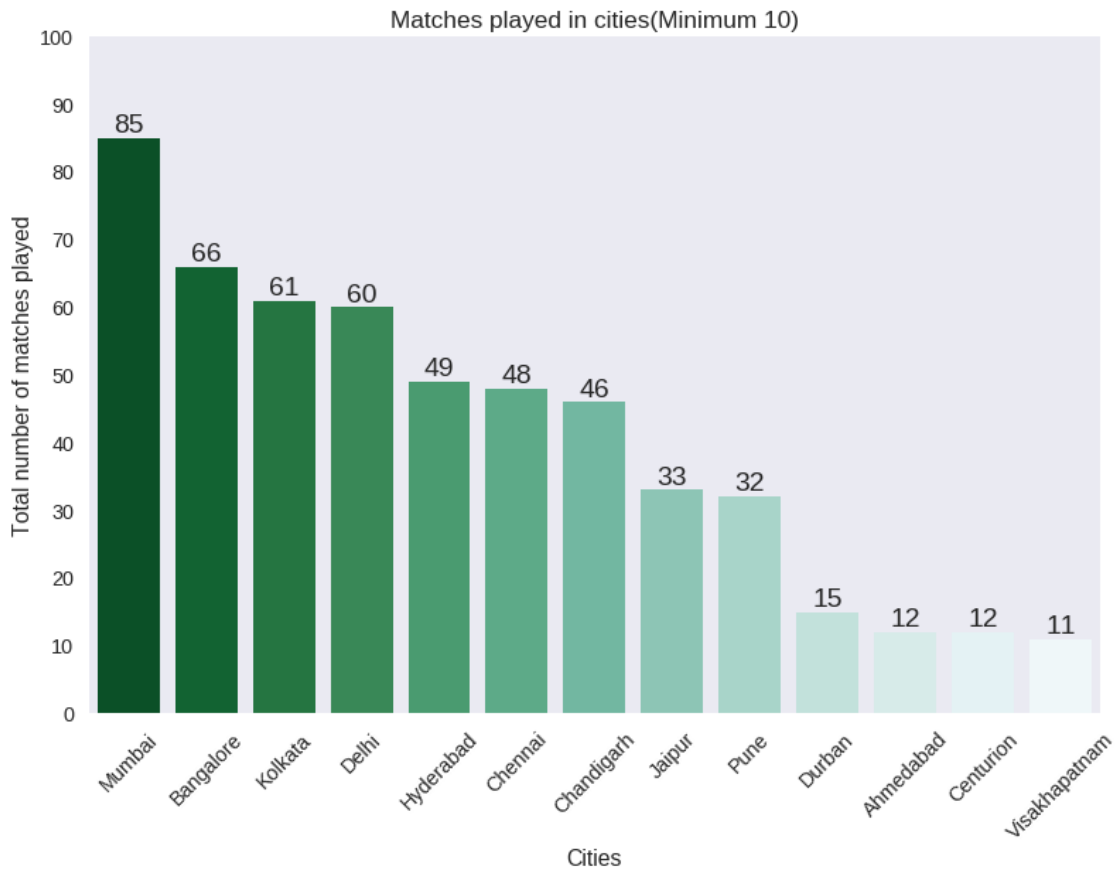
xl=plt.gca().xaxis
for item in xl.get_ticklabels():
    item.set_rotation(45)

ax.yaxis.tick_left()

ax.yaxis.set_label_position('left')
ncount = 100
for p in ax.patches:
    x_a=p.get_bbox().get_points()[:,0]
    y_a=p.get_bbox().get_points()[1,1]
    ax.annotate('{:.0f}'.format(y_a),(x_a.mean(), y_a),ha='center', va='bottom')

ax.yaxis.set_major_locator(ticker.LinearLocator(11))
ax.set_ylim(0,ncount)
```

```
Out[29]: (0, 100)
```



5 Mumbai is the city where maximum number of matches have been played

6 in stadiums:

7 1.Wankhede Stadium -57

8 2.Dr DY Patil Sports Academy -17

9 3.Brabourne Stadium -11

```
In [30]: df6=df[['venue','city','winner','player_of_match']]
          df6.head()
```

```
Out[30]:
```

	venue	city
0	Rajiv Gandhi International Stadium, Uppal	Hyderabad
1	Maharashtra Cricket Association Stadium	Pune
2	Saurashtra Cricket Association Stadium	Rajkot

```

3          Holkar Cricket Stadium      Indore
4          M Chinnaswamy Stadium  Bangalore

```

```

          winner player_of_match
0      Sunrisers Hyderabad      Yuvraj Singh
1      Rising Pune Supergiants      SPD Smith
2      Kolkata Knight Riders      CA Lynn
3      Kings XI Punjab      GJ Maxwell
4      Royal Challengers Bangalore      KM Jadhav

```

```

In [31]: x = df6.groupby('player_of_match').count()
         x.reset_index()

```

```

Out[31]:
   player_of_match  venue  city  winner
0      A Chandila      1     1        1
1      A Kumble      3     3        3
2      A Mishra     9     9        9
3      A Nehra      6     6        6
4      A Singh      1     1        1
5      A Symonds     3     3        3
6      A Zampa      1     1        1
7      AA Jhunjunwala  1     1        1
8      AB Dinda      2     2        2
9      AB de Villiers  15    15       15
10     AC Gilchrist     7     7        7
11     AC Voges        1     1        1
12     AD Mascarenhas  1     1        1
13     AD Mathews      1     1        1
14     AD Russell      6     6        6
15     AJ Finch        5     4        5
16     AJ Tye          2     2        2
17     AM Rahane     12    12       12
18     AP Tare        1     1        1
19     AR Patel       4     4        4
20     AT Rayudu      7     7        7
21     Azhar Mahmood   2     2        2
22     B Kumar        5     4        5
23     B Lee          2     2        2
24     BA Bhatt       1     1        1
25     BA Stokes      3     3        3
26     BB McCullum     5     5        5
27     BCJ Cutting    1     1        1
28     BJ Hodge       6     6        6
29     BW Hilfenhaus   2     2        2
..          ...      ...      ...      ...
171     SL Malinga     5     5        5
172     SM Katich      1     1        1
173     SM Pollock     2     2        2

```

174	SP Goswami	1	1	1
175	SP Narine	6	6	6
176	SPD Smith	4	4	4
177	SR Tendulkar	8	8	8
178	SR Watson	10	10	10
179	SS Iyer	2	2	2
180	ST Jayasuriya	2	2	2
181	SV Samson	3	3	3
182	Sandeep Sharma	5	4	5
183	Shakib Al Hasan	2	2	2
184	Shoaib Akhtar	1	1	1
185	Sohail Tanvir	2	2	2
186	TA Boult	1	1	1
187	TL Suman	1	1	1
188	TM Dilshan	1	1	1
189	UT Yadav	6	6	6
190	Umar Gul	1	1	1
191	V Kohli	11	11	11
192	V Sehwag	11	11	11
193	VR Aaron	1	1	1
194	WP Saha	3	3	3
195	WPUJC Vaas	1	1	1
196	Washington Sundar	1	1	1
197	YK Pathan	16	16	16
198	YS Chahal	1	1	1
199	Yuvraj Singh	5	5	5
200	Z Khan	1	1	1

[201 rows x 4 columns]

```
In [32]: x=x.where(x['winner']>=10).sort_values('winner',axis=0,ascending=False)
x=x.reset_index()
x=x.drop(['venue','city'],axis=1)
x=x.dropna()
x
```

```
Out[32]:
```

	player_of_match	winner
0	CH Gayle	18.0
1	YK Pathan	16.0
2	AB de Villiers	15.0
3	DA Warner	15.0
4	RG Sharma	14.0
5	SK Raina	14.0
6	G Gambhir	13.0
7	MS Dhoni	13.0
8	AM Rahane	12.0
9	MEK Hussey	12.0
10	DR Smith	11.0

11	V Kohli	11.0
12	V Sehwag	11.0
13	JH Kallis	10.0
14	SR Watson	10.0

```
In [33]: sns.set_style('dark')
ax=sns.barplot(x='player_of_match',y='winner',data=x,palette='GnBu_d')
plt.xlabel('Player')
plt.ylabel(' Number of times Player of the Match ')
plt.title('Total Player of the match award received by a Player ')
sns.set_context("poster", font_scale = 1)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)

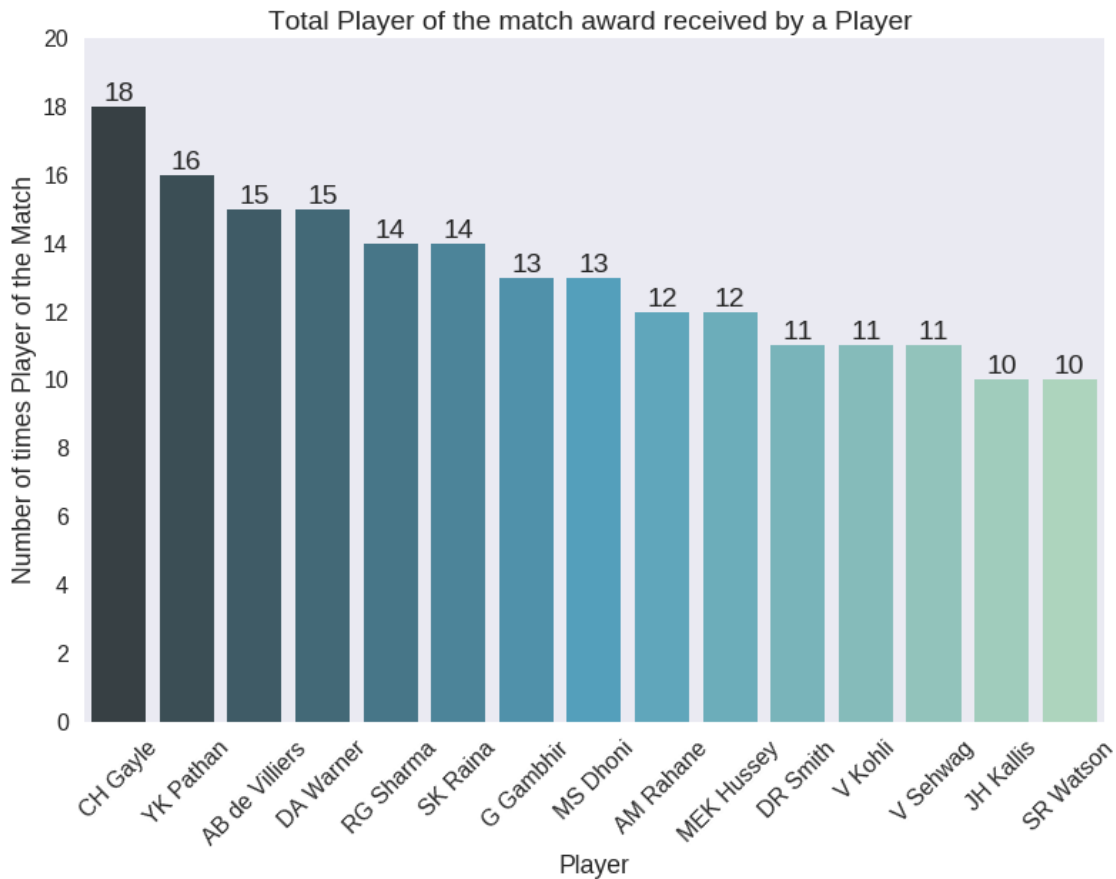
xl=plt.gca().xaxis
for item in xl.get_ticklabels():
    item.set_rotation(45)

ax.yaxis.tick_left()

ax.yaxis.set_label_position('left')
ncount = 20
for p in ax.patches:
    x_a=p.get_bbox().get_points()[:,0]
    y_a=p.get_bbox().get_points()[1,1]
    ax.annotate('{:.0f}'.format(y_a),(x_a.mean(), y_a),ha='center', va='bottom')

ax.yaxis.set_major_locator(ticker.LinearLocator(11))
ax.set_ylim(0,ncount)
```

Out[33]: (0, 20)



10 CH Gayle has maximum number of player of the match award in IPL

```
In [34]: df7=df2[['win_by_runs','win_by_wickets','winner']]
df7.head()
```

```
Out[34]:
```

	win_by_runs	win_by_wickets	winner
116	0	3	Rajasthan Royals
82	0	8	Rajasthan Royals
81	29	0	Mumbai Indians
80	9	0	Kings XI Punjab
79	0	5	Royal Challengers Bangalore

```
In [35]: df7=df7[df7["win_by_wickets"]==10]
```

```
In [36]: sns.set_style('dark')
ax=sns.countplot(x='winner',data=df7,palette='cubehelix')
plt.xlabel('Winning Teams')
plt.ylabel('Count')
```

```

plt.title('10 wicket win')
sns.set_context("poster", font_scale = 0.8)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)

xl=plt.gca().xaxis
for item in xl.get_ticklabels():
    item.set_rotation(45)

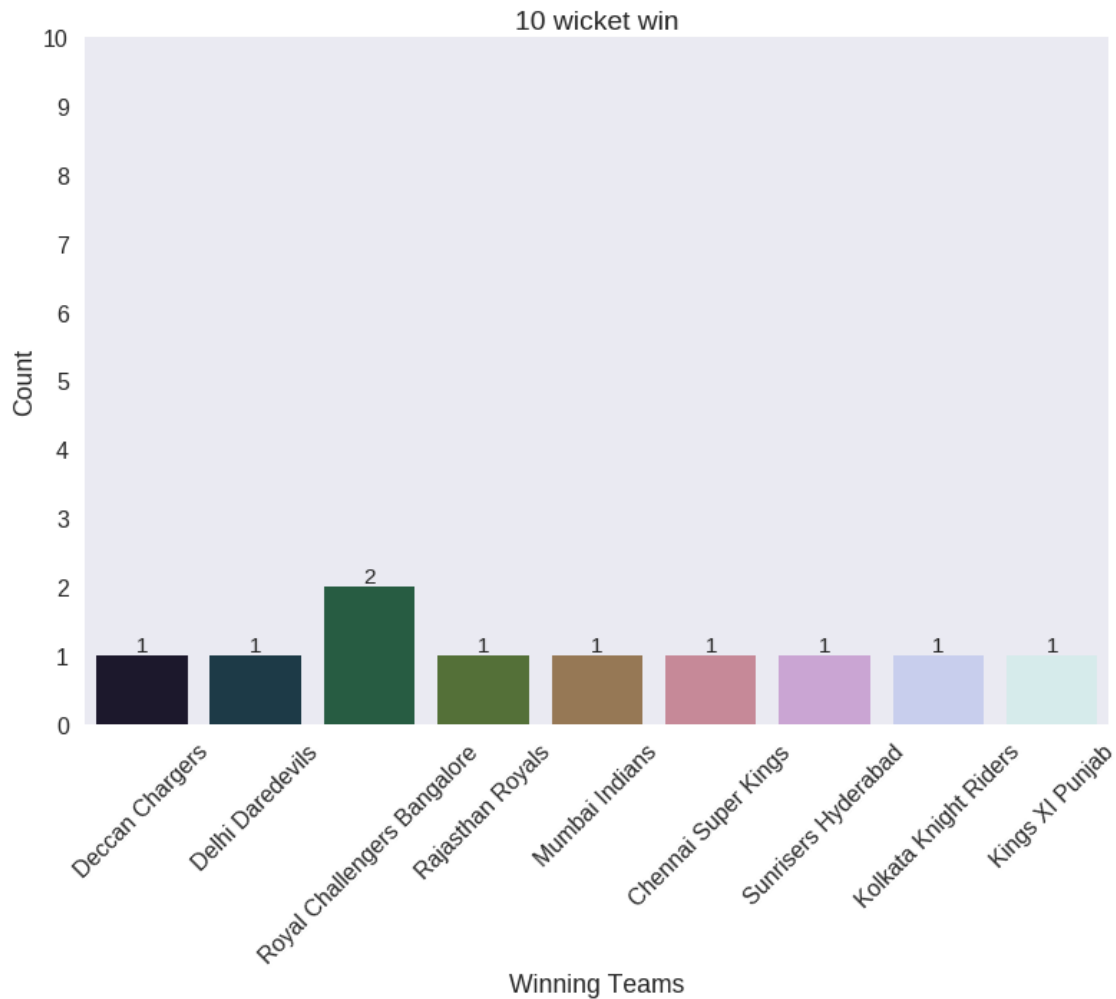
ax.yaxis.tick_left()

ax.yaxis.set_label_position('left')
ncount = 10
for p in ax.patches:
    x_a=p.get_bbox().get_points()[0,0]
    y_a=p.get_bbox().get_points()[1,1]
    ax.annotate('{:.0f}'.format(y_a), (x_a.mean(), y_a), ha='center', va='bottom')

ax.yaxis.set_major_locator(ticker.LinearLocator(11))
ax.set_ylim(0,ncount)

```

Out[36]: (0, 10)



11 Royal Challengers Bangalore has won 10 Wicket win Two times

```
In [37]: df8=df2[['win_by_runs','win_by_wickets','winner']]
         df8=df8[df8["win_by_runs"]>=100]

In [38]: sns.set_style('dark')
         ax=sns.countplot(x='winner',data=df8,palette='cubehelix')
         plt.xlabel('Teams Winning')
         plt.ylabel('Count')
         plt.title('Teams Winning by Runs(minimum 100)')
         sns.set_context("poster", font_scale = 1)
         plt.gca().spines['top'].set_visible(False)
         plt.gca().spines['right'].set_visible(False)

         xl=plt.gca().xaxis
         for item in xl.get_ticklabels():
```



```

        item.set_rotation(45)

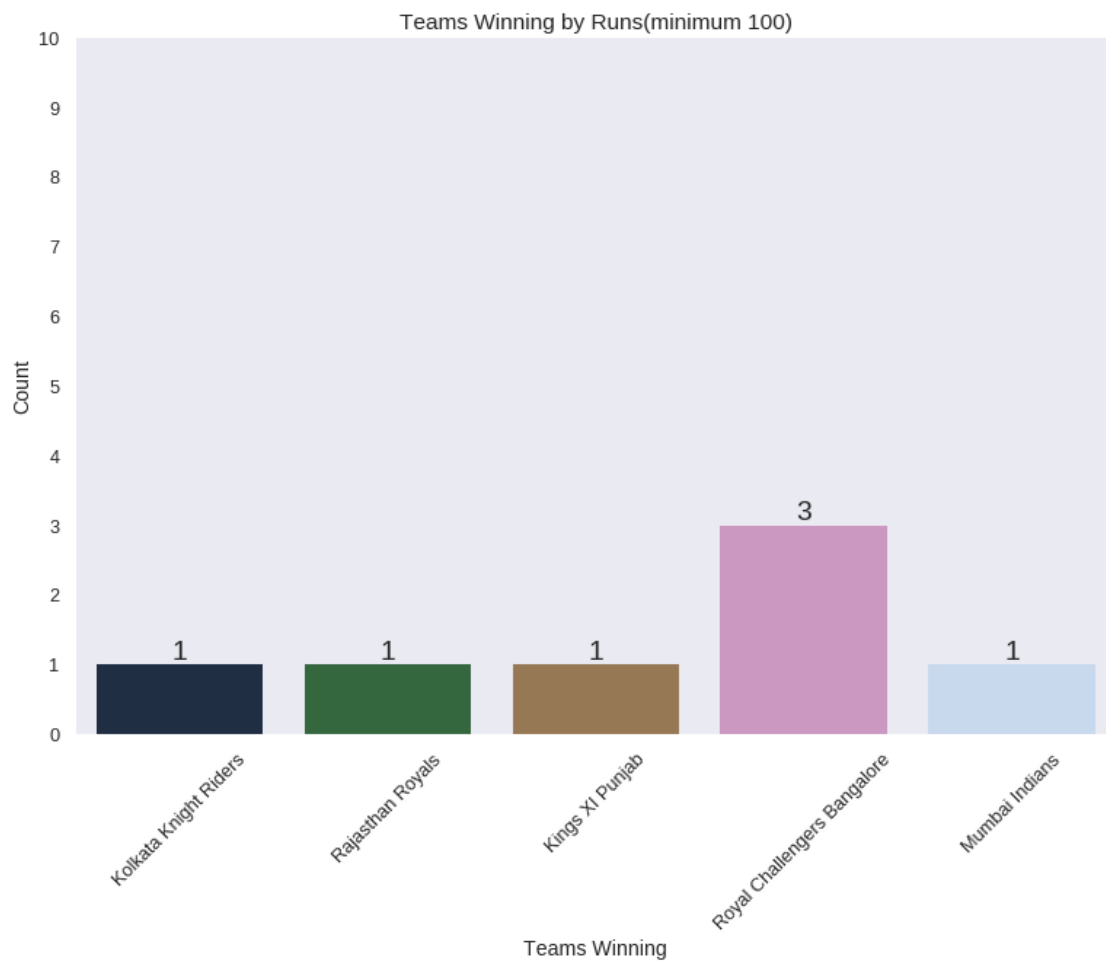
ax.yaxis.tick_left()

ax.yaxis.set_label_position('left')
ncount = 10
for p in ax.patches:
    x_a=p.get_bbox().get_points()[0,0]
    y_a=p.get_bbox().get_points()[1,1]
    ax.annotate('{:.0f}'.format(y_a), (x_a.mean(), y_a), ha='center', va='bottom')

ax.yaxis.set_major_locator(ticker.LinearLocator(11))
ax.set_ylim(0,ncount)

```

Out[38]: (0, 10)



```
In [39]: df2.head()
```

```
Out[39]:
```

	id	season	city	date	team1	\	team2	toss_winner	toss_decision	result
116	117	2008	Mumbai	2008-06-01	Chennai Super Kings		Rajasthan Royals	Rajasthan Royals	field	normal
82	83	2008	Jaipur	2008-05-04	Chennai Super Kings		Rajasthan Royals	Chennai Super Kings	bat	normal
81	82	2008	Mumbai	2008-05-04	Mumbai Indians		Delhi Daredevils	Delhi Daredevils	field	normal
80	81	2008	Chandigarh	2008-05-03	Kings XI Punjab		Kolkata Knight Riders	Kings XI Punjab	bat	normal
79	80	2008	Hyderabad	2008-05-25	Deccan Chargers		Royal Challengers Bangalore	Deccan Chargers	bat	normal

	dl_applied	winner	win_by_runs	win_by_wickets
116	0	Rajasthan Royals	0	3
82	0	Rajasthan Royals	0	8
81	0	Mumbai Indians	29	0
80	0	Kings XI Punjab	9	0
79	0	Royal Challengers Bangalore	0	5

	player_of_match	venue	umpire1
116	YK Pathan	Dr DY Patil Sports Academy	BF Bowden
82	Sohail Tanvir	Sawai Mansingh Stadium	Asad Rauf
81	SM Pollock	Dr DY Patil Sports Academy	IL Howell
80	IK Pathan	Punjab Cricket Association Stadium, Mohali	DJ Harper
79	R Vinay Kumar	Rajiv Gandhi International Stadium, Uppal	Asad Rauf

	umpire2	umpire3
116	RE Koertzen	NaN
82	AV Jayaprakash	NaN
81	RE Koertzen	NaN
80	I Shivram	NaN
79	RE Koertzen	NaN

```
In [40]: final=df2.copy()
```

```
In [49]: final['season'].unique()
```

```
Out[49]: array([2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017])
```

```
In [42]: fff=final.groupby(['season','winner']).count()
```

```
In [43]: fff.columns
```

```
Out[43]: Index(['id', 'city', 'date', 'team1', 'team2', 'toss_winner', 'toss_decision', 'result', 'dl_applied', 'win_by_runs', 'win_by_wickets', 'player_of_match', 'venue', 'umpire1', 'umpire2', 'umpire3'], dtype='object')
```

```
In [44]: dp=['id','city', 'date', 'team1', 'team2', 'toss_winner', 'toss_decision',
```

```
In [45]: fff=fff.drop(dp,axis=1)
```

```
In [46]: fff=fff.reset_index()
```

```
In [47]: fff.head()
```

```
Out[47]:
```

	season	winner	result
0	2008	Chennai Super Kings	9
1	2008	Deccan Chargers	2
2	2008	Delhi Daredevils	7
3	2008	Kings XI Punjab	10
4	2008	Kolkata Knight Riders	6

```
In [ ]:
```

```
In [ ]:
```