In [48]:

```
import pandas as pd
import matplotlib
import numpy as np
import seaborn as sb
from matplotlib import pyplot as plt
import random as rnd
```

In [47]:

```
df=pd.read_csv("IPL Matches 2008-2020.csv")
df
```

Out[47]:

	id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	
0	335982	Bangalore	2008-04-18	BB McCullum	M Chinnaswamy Stadium	0	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore	field	
1	335983	Chandigarh	2008-04-19	MEK Hussey	Punjab Cricket Association Stadium, Mohali	0	Kings XI Punjab	Chennai Super Kings	Chennai Super Kings	bat	
2	335984	Delhi	2008-04-19	MF Maharoof	Feroz Shah Kotla	0	Delhi Daredevils	Rajasthan Royals	Rajasthan Royals	bat	D
3	335985	Mumbai	2008-04-20	MV Boucher	Wankhede Stadium	0	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	Cl
4	335986	Kolkata	2008-04-20	DJ Hussey	Eden Gardens	0	Kolkata Knight Riders	Deccan Chargers	Deccan Chargers	bat	
...	...	...	...	...	...	...	...	...	...	...	
811	1216547	Dubai	2020-09-28	AB de Villiers	Dubai International Cricket Stadium	0	Royal Challengers Bangalore	Mumbai Indians	Mumbai Indians	field	Cl
812	1237177	Dubai	2020-11-05	JJ Bumrah	Dubai International Cricket Stadium	0	Mumbai Indians	Delhi Capitals	Delhi Capitals	field	
813	1237178	Abu Dhabi	2020-11-06	KS Williamson	Sheikh Zayed Stadium	0	Royal Challengers Bangalore	Sunrisers Hyderabad	Sunrisers Hyderabad	field	H
814	1237180	Abu Dhabi	2020-11-08	MP Stoinis	Sheikh Zayed Stadium	0	Delhi Capitals	Sunrisers Hyderabad	Delhi Capitals	bat	
815	1237181	Dubai	2020-11-10	TA Boult	Dubai International Cricket Stadium	0	Delhi Capitals	Mumbai Indians	Delhi Capitals	bat	

816 rows × 17 columns

column description

id - A unique id assigned to each match. city - The name of the city where the match was played. date - The date on which the match was played. player of match - The name of the player who was awarded the "Player of the Match" award for the match. venue - The name of the cricket stadium where the match was played. neutral venue - A binary column indicating whether the match was played in a neutral venue or not. team1 - The name of the first team playing in the match. team2 - The name of the second team playing in the match. toss winner - The name of the team that won the toss. toss decision - The decision made by the team that won the toss (batting or bowling). winner - The name of the team that won the match. result - The result of the match (win/loss/tie/no result). result_margin - The margin of victory (in runs or wickets) for the winning team. eliminator - a column indicating whether the match was decided by the eliminator method or not. method - The method by which the match was decided. umpire1 - The name of the first umpire . umpire2 - The name of the second umpire.

In [6]:

```
df.columns
```

Out[6]:

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

In [5]:

```
# which team has won the most matches
w=df.winner.value_counts()
print(w.index[0])
```

```
Mumbai Indians
```

In [8]:

```
# how many times mumbai indians wins the toss
w=df[(df.toss_winner=="Mumbai Indians")]
len(w)
```

Out[8]:

```
106
```

In [18]:

```
df.neutral_venue.unique()
```

Out[18]:

```
array([0, 1], dtype=int64)
```

In [32]:

```
# how many matches have been played in neutral venue
nv=df.loc[df["neutral_venue"]==1]
print("winning matches in neutral venue: ",len(nv))
```

```
winning matches in neutral venue:  77
```

In [37]:

```
#no of matches decides by eliminator
l=df.loc[df.eliminator=="Y"]
len(l)
```

Out[37]:

```
13
```

In [41]:

```
#teams that choose batting first after winning toss
w=df.loc[df.toss_decision=="bat"]["toss_winner"]
print(len(w)," teams")
w
```

```
320  teams
```

Out[41]:

```
1              Chennai Super Kings
2                 Rajasthan Royals
3                   Mumbai Indians
4                  Deccan Chargers
5                  Kings XI Punjab
                   ...
808    Royal Challengers Bangalore
809            Sunrisers Hyderabad
810                 Delhi Capitals
814                 Delhi Capitals
815                 Delhi Capitals
Name: toss_winner, Length: 320, dtype: object
```

In [43]:

```
#city in which most of the matches are played
city=df.city.value_counts()
print("city : ",city.index[0])
```

```
city :  Mumbai
```

In [45]:

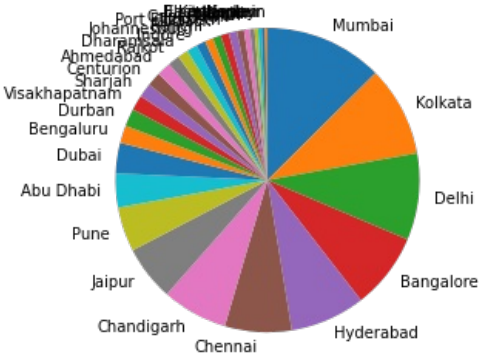
```
# which player become most of the time player of the match
l=df.player_of_match.value_counts()
print(l.index[0])
```

```
AB de Villiers
```

matplotlib

In [49]:

```
city=df.city.value_counts()
plt.pie(city,labels=city.index,startangle=90,counterclock=False);
plt.axis('square');
```

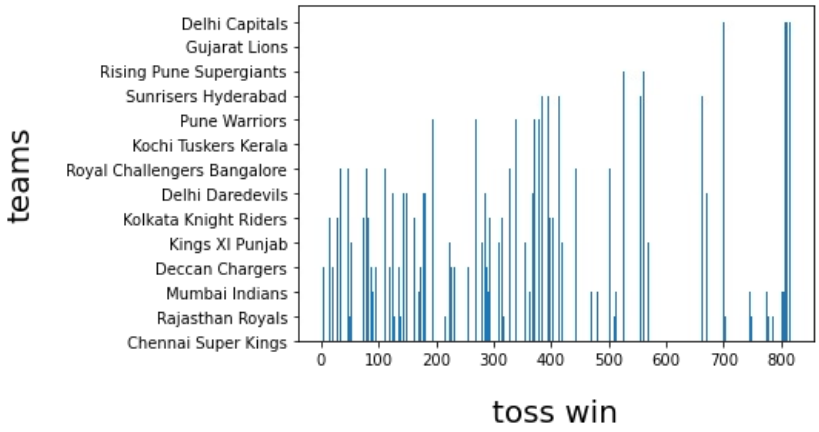


In [51]:

```
w=df.loc[df.toss_decision=="bat"]["toss_winner"]
w
w1=w.index
plt.xlabel("toss win",fontsize=20,labelpad=20)
plt.ylabel("teams",fontsize=20,labelpad=20)
plt.bar(w1,w)
```

Out[51]:

```
<BarContainer object of 320 artists>
```

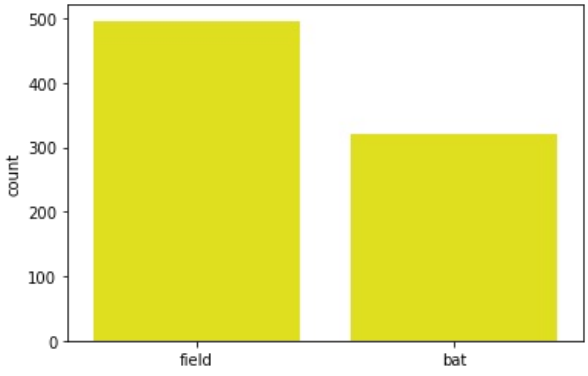


In [53]:

```
sb.countplot(data=df,x="toss_decision",color= "yellow")
plt.xlabel("")
```

Out[53]:

```
Text(0.5, 0, '')
```



In [ ]: