GAME PLAN FOR WINNING THE MATCHES IN BBL

1. Importing all the libriaries to perform analysis.

In [ ]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings("ignore")
```

In [ ]:

```
df = pd.read_table("BBL Matches 2011-2019.csv" , sep= ",")
df
```

Out[5]:

	id	city	date	player_of_match	venue	neutral_venue	team1	
0	524915	Sydney	2011-12-16	BJ Haddin	Sydney Cricket Ground	0	Sydney Sixers	E
1	524916	Melbourne	2011-12-17	DA Warner	Melbourne Cricket Ground	0	Melbourne Stars	-
2	524917	Adelaide	2011-12-18	M Klinger	Adelaide Oval	0	Adelaide Strikers	Me Rer
3	524918	Perth	2011-12-18	BW Hilfenhaus	Western Australia Cricket Association Ground	0	Perth Scorchers	Hu
4	524919	Brisbane	2011-12-20	RJ Quiney	Brisbane Cricket Ground, Woolloongabba	0	Brisbane Heat	Me
...	...	...	...	...	...	...	...	
348	1195629	Hobart	2020-01-30	JD Cook	Bellerive Oval	0	Hobart Hurricanes	-
349	1195630	Melbourne	2020-01-31	SNJ O'Keefe	Melbourne Cricket Ground	0	Melbourne Stars	
350	1195631	Adelaide	2020-02-01	AD Hales	Adelaide Oval	0	Adelaide Strikers	-
351	1195632	Melbourne	2020-02-06	NCR Larkin	Melbourne Cricket Ground	0	Melbourne Stars	-
352	1195633	Sydney	2020-02-08	JR Philippe	Sydney Cricket Ground	0	Sydney Sixers	Me

353 rows × 17 columns

2. To display the first five rows from the dataset.

In [ ]:

```
df.head()
```

Out[6]:

	id	city	date	player_of_match	venue	neutral_venue	team1	tea
0	524915	Sydney	2011-12-16	BJ Haddin	Sydney Cricket Ground	0	Sydney Sixers	Brisb H
1	524916	Melbourne	2011-12-17	DA Warner	Melbourne Cricket Ground	0	Melbourne Stars	Syd Thur
2	524917	Adelaide	2011-12-18	M Klinger	Adelaide Oval	0	Adelaide Strikers	Melbou Renega
3	524918	Perth	2011-12-18	BW Hilfenhaus	Western Australia Cricket Association Ground	0	Perth Scorchers	Ho Hurrica
4	524919	Brisbane	2011-12-20	RJ Quiney	Brisbane Cricket Ground, Woolloongabba	0	Brisbane Heat	Melbou S

3. To display last five rows from the dataset.

In [ ]:

```
df.tail()
```

Out[7]:

	id	city	date	player_of_match	venue	neutral_venue	team1	tean
348	1195629	Hobart	2020-01-30	JD Cook	Bellerive Oval	0	Hobart Hurricanes	Sydne Thund
349	1195630	Melbourne	2020-01-31	SNJ O'Keefe	Melbourne Cricket Ground	0	Melbourne Stars	Sydne Sixe
350	1195631	Adelaide	2020-02-01	AD Hales	Adelaide Oval	0	Adelaide Strikers	Sydne Thund
351	1195632	Melbourne	2020-02-06	NCR Larkin	Melbourne Cricket Ground	0	Melbourne Stars	Sydne Thund
352	1195633	Sydney	2020-02-08	JR Philippe	Sydney Cricket Ground	0	Sydney Sixers	Melbourn Sta

4. To display the datatypes of the entries.

In [ ]:

```
df.dtypes
```

Out[8]:

```
id                    int64
city                 object
date                 object
player_of_match      object
venue                object
neutral_venue         int64
team1                object
team2                object
toss_winner          object
toss_decision        object
winner               object
result               object
result_margin       float64
eliminator           object
method               object
umpire1              object
umpire2              object
dtype: object
```

5. To display the number of entries and information in the dataset.

In [ ]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 353 entries, 0 to 352
Data columns (total 17 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   id               353 non-null    int64  
 1   city             353 non-null    object 
 2   date             353 non-null    object 
 3   player_of_match  348 non-null    object 
 4   venue            353 non-null    object 
 5   neutral_venue    353 non-null    int64  
 6   team1            353 non-null    object 
 7   team2            353 non-null    object 
 8   toss_winner      353 non-null    object 
 9   toss_decision    353 non-null    object 
 10  winner           348 non-null    object 
 11  result           348 non-null    object 
 12  result_margin    344 non-null    float64
 13  eliminator       348 non-null    object 
 14  method           18 non-null     object 
 15  umpire1          353 non-null    object 
 16  umpire2          353 non-null    object 
dtypes: float64(1), int64(2), object(14)
memory usage: 47.0+ KB
```

6. To display the detailed information and description of the dataset.

In [ ]:

```
df.describe()
```

Out[10]:

	id	neutral_venue	result_margin
count	3.530000e+02	353.0	344.000000
mean	9.392681e+05	0.0	17.941860
std	2.421939e+05	0.0	21.320875
min	5.249150e+05	0.0	1.000000
25%	6.540990e+05	0.0	6.000000
50%	1.023623e+06	0.0	8.000000
75%	1.152542e+06	0.0	20.250000
max	1.195633e+06	0.0	112.000000

7. To remove unwanted column.

In [ ]:

```
df.drop(["neutral_venue","method"], axis=1 , inplace = True)
df
```

Out[11]:

	id	city	date	player_of_match	venue	team1	team2	toss_w
0	524915	Sydney	2011-12-16	BJ Haddin	Sydney Cricket Ground	Sydney Sixers	Brisbane Heat	Bri
1	524916	Melbourne	2011-12-17	DA Warner	Melbourne Cricket Ground	Melbourne Stars	Sydney Thunder	S Th
2	524917	Adelaide	2011-12-18	M Klinger	Adelaide Oval	Adelaide Strikers	Melbourne Renegades	Melb Rene
3	524918	Perth	2011-12-18	BW Hilfenhaus	Western Australia Cricket Association Ground	Perth Scorchers	Hobart Hurricanes	H Hurri
4	524919	Brisbane	2011-12-20	RJ Quiney	Brisbane Cricket Ground, Woolloongabba	Brisbane Heat	Melbourne Stars	Melb
...	...	...	...	...	...	...	...	
348	1195629	Hobart	2020-01-30	JD Cook	Bellerive Oval	Hobart Hurricanes	Sydney Thunder	S Th
349	1195630	Melbourne	2020-01-31	SNJ O'Keefe	Melbourne Cricket Ground	Melbourne Stars	Sydney Sixers	Melb
350	1195631	Adelaide	2020-02-01	AD Hales	Adelaide Oval	Adelaide Strikers	Sydney Thunder	Ad St
351	1195632	Melbourne	2020-02-06	NCR Larkin	Melbourne Cricket Ground	Melbourne Stars	Sydney Thunder	Melb
352	1195633	Sydney	2020-02-08	JR Philippe	Sydney Cricket Ground	Sydney Sixers	Melbourne Stars	Melb

353 rows × 15 columns

8. To display the NaN values.

In [ ]:

```
df.isna().sum()
```

Out[12]:

```
id                 0
city               0
date               0
player_of_match    5
venue              0
team1              0
team2              0
toss_winner        0
toss_decision      0
winner             5
result             5
result_margin      9
eliminator         5
umpire1            0
umpire2            0
dtype: int64
```

9. Filling the NaN values.

In [ ]:

```
df = df.fillna(method = "backfill")
```

In [ ]:

```
df.isna().sum()
```

Out[14]:

```
id                 0
city               0
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
```

10. The team to win most number of matches.

In [ ]:

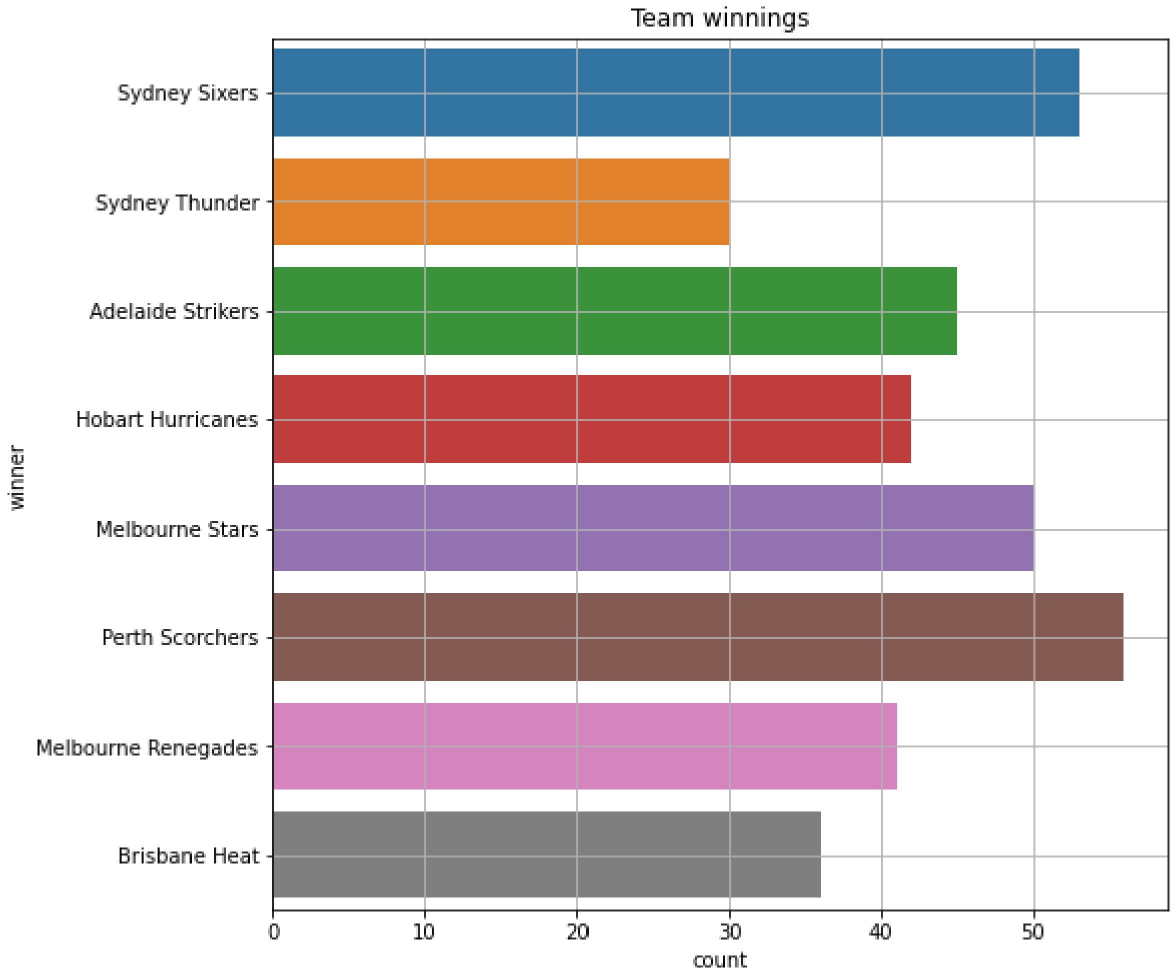
```
#Analytical

df.winner.value_counts()
```

Out[15]:

```
Perth Scorchers       56
Sydney Sixers         53
Melbourne Stars       50
Adelaide Strikers     45
Hobart Hurricanes     42
Melbourne Renegades   41
Brisbane Heat         36
Sydney Thunder        30
Name: winner, dtype: int64
```

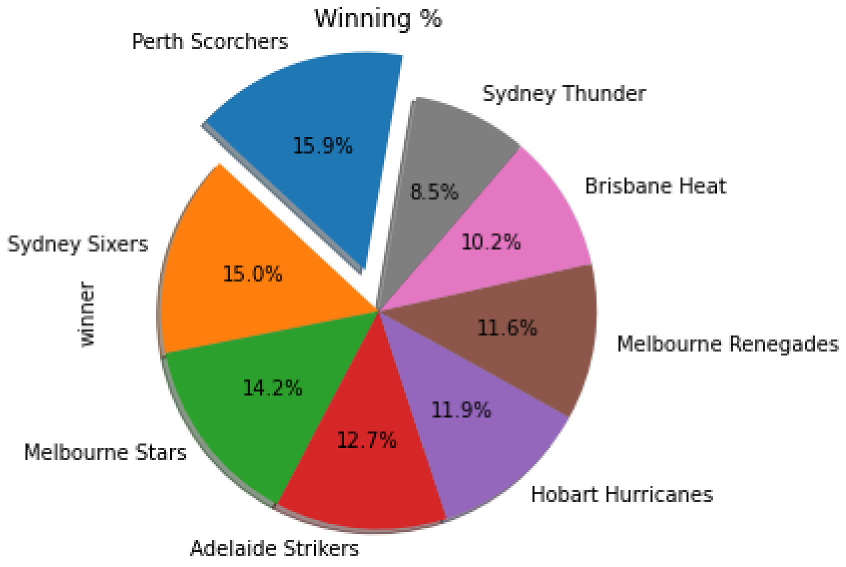
In [ ]:

```
#Graphical

plt.figure(figsize=(8,8))

sns.countplot(data=df, y="winner")

plt.title("Team winnings")
plt.grid(True)
plt.show()
```



In [ ]:

```
#Pie plot

plt.figure(figsize=(5,5))

df['winner'].value_counts().plot.pie(startangle=80, shadow=True, autopct="%1.1f%%",explode=

plt.title("Winning %")

plt.show()
```



From the above data it is observed that the team "Perth Scorchers" has won most number of matches having winning% of 15.9.

11. Date wise winning of the most successfull team.

In [ ]:

```
df[df['winner']=='Perth Scorchers']['date']
```

Out[18]:

```
6      2011-12-22
10     2011-12-29
14     2012-01-04
16     2012-01-08
24     2012-01-21
35     2012-12-16
36     2012-12-18
44     2012-12-29
47     2013-01-04
51     2013-01-10
53     2013-02-05
59     2013-12-26
67     2014-01-03
70     2014-01-07
73     2014-01-10
78     2014-01-16
87     2014-02-05
88     2014-02-07
95     2014-12-26
98     2014-12-29
106    2015-01-06
107    2015-01-08
110    2015-01-11
121    2015-01-25
122    2015-01-28
129    2015-12-26
133    2015-12-30
137    2016-01-02
142    2016-01-07
145    2016-01-10
159    2016-12-23
163    2016-12-29
167    2017-01-01
177    2017-01-11
185    2017-01-21
187    2017-01-24
189    2017-01-28
194    2017-12-23
196    2017-12-26
199    2017-12-29
203    2018-01-01
211    2018-01-08
216    2018-01-13
223    2018-01-20
227    2018-01-25
241    2018-12-26
257    2019-01-09
262    2019-01-13
283    2019-02-03
297    2019-12-21
298    2019-12-21
309    2020-01-01
317    2020-01-07
323    2020-01-11
326    2020-01-13
```

```
337    2020-01-20
Name: date, dtype: object
```

12. The team to win most number of toss.

In [ ]:

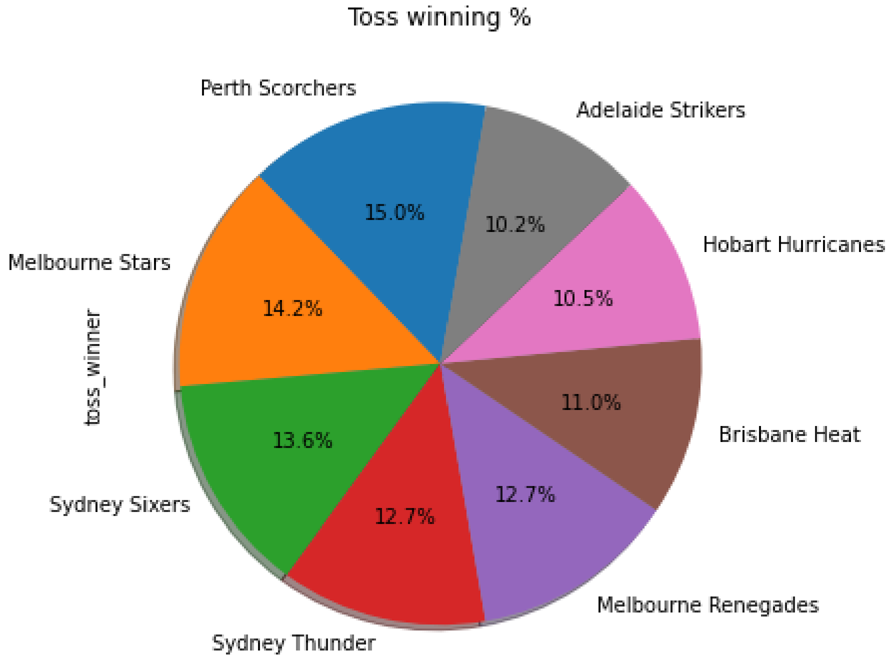
```
#Analytical

df["toss_winner"].value_counts().idxmax()
```

Out[19]:

```
'Perth Scorchers'
```

In [ ]:

```
#Pie Plot

plt.figure(figsize=(6,6))
df["toss_winner"].value_counts().plot.pie(startangle=80, shadow=True, autopct="%1.1f%%")

plt.title("Toss winning %")

plt.show()
```

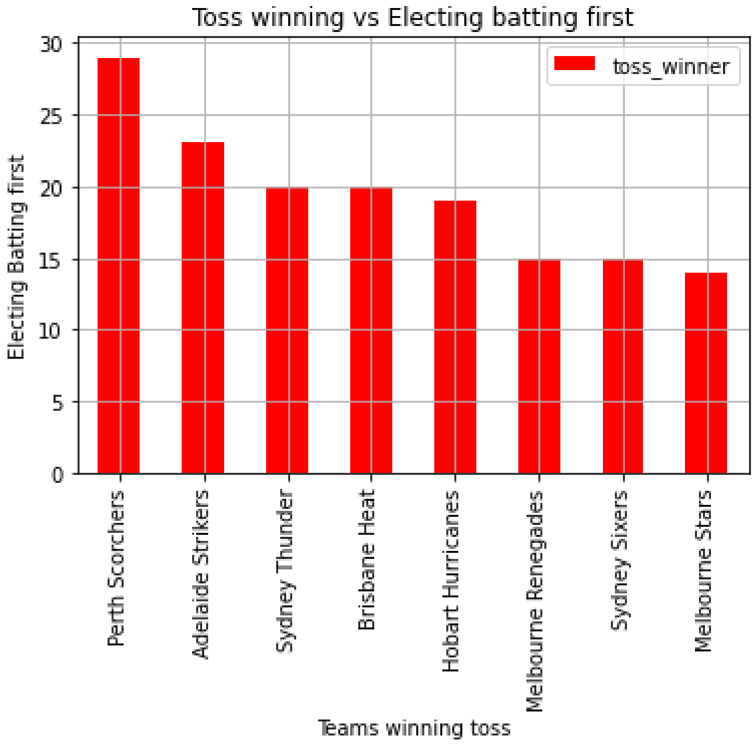


From the above data it is obsereved that the team "Perth Scorchers" has won most number of toss.

13. The teams to elect batting first after winning the toss.

In [ ]:

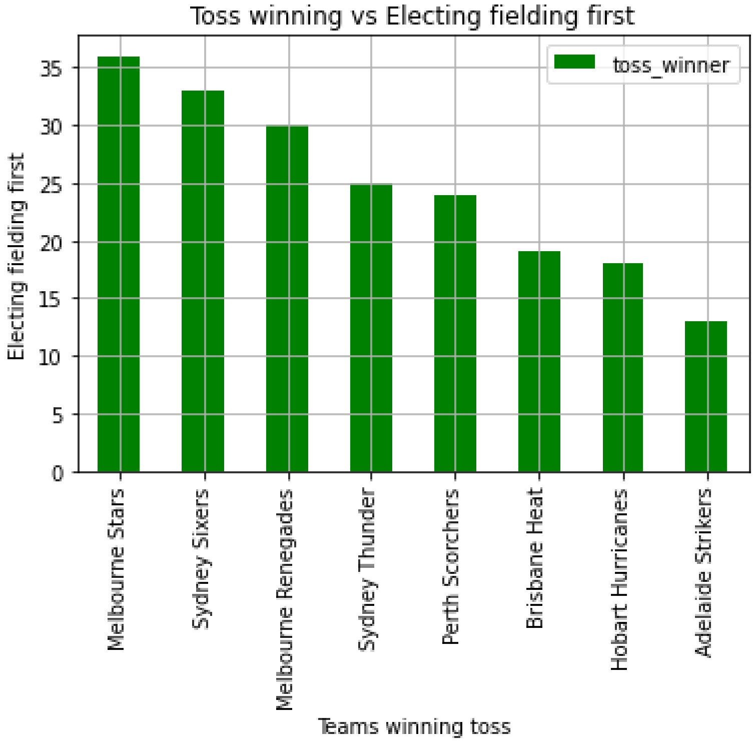
```
#Graphical

df[df.toss_decision=="bat"]['toss_winner'].value_counts().plot(kind=("bar") , color ="r")

plt.xlabel("Teams winning toss")
plt.ylabel("Electing Batting first")

plt.title("Toss winning vs Electing batting first")
plt.grid(True)
plt.legend()

plt.show()
```



14. The teams to elect fielding first after winning the toss.

In [ ]:

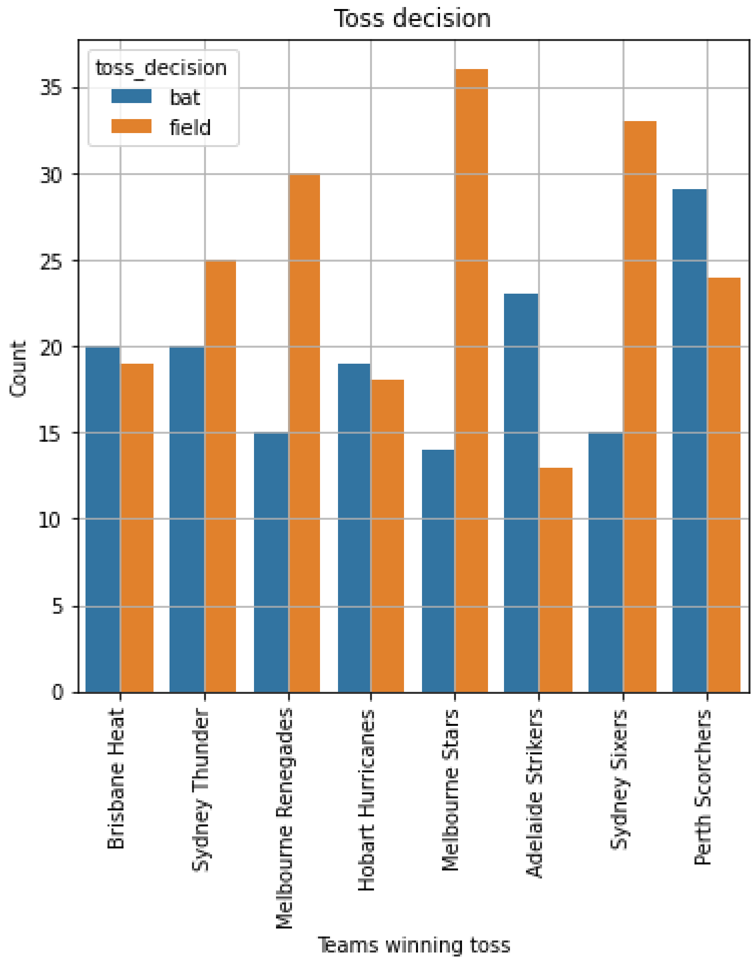
```
#Graphical

df[df.toss_decision=="field"]['toss_winner'].value_counts().plot(kind=("bar") , color ="g")

plt.xlabel("Teams winning toss")
plt.ylabel("Electing fielding first")

plt.title("Toss winning vs Electing fielding first")
plt.grid(True)
plt.legend()

plt.show()
```



15. Teamwise toss decision.

In [ ]:

```
plt.figure(figsize=(6,6))

sns.countplot(data=df, x="toss_winner", hue="toss_decision")

plt.title("Toss decision")

plt.xlabel("Teams winning toss")
plt.ylabel("Count")

plt.xticks(rotation=90)
plt.grid(True)
plt.show()
```



From above data it is observed that after winning the toss 50% of teams likes to bat first and 50% of teams like to field first.

16. The teams winning after electing to bat first.

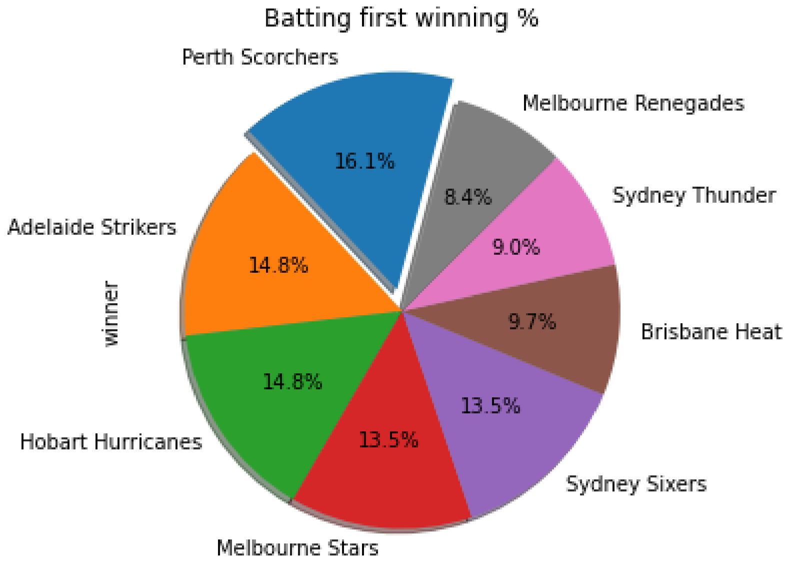
```
In [ ]:
```

```
#Analytical

df[df.toss_decision=="bat"]['winner'].value_counts()
```

```
Out[24]:

Perth Scorchers        25
Adelaide Strikers      23
Hobart Hurricanes      23
Melbourne Stars        21
Sydney Sixers          21
Brisbane Heat          15
Sydney Thunder         14
Melbourne Renegades    13
Name: winner, dtype: int64
```

```
In [ ]:
```

```
#Pie plot

plt.figure(figsize=(5,5))

df[df.toss_decision=="bat"]['winner'].value_counts().plot.pie(startangle=75, shadow=True, a
explode =(0.1,0,0,0,0,0,0,0))

plt.title("Batting first winning %")

plt.show()
```



From the above data it is observed that after winning the toss and electing to bat first the team "Perth Scorchers" has highest number of winnings and winning% i.e 25 winnings and 16.1 winning% respectievely.

17. Winning average after batting first.

```
In [ ]:
```

```
df[df.toss_decision=="bat"]['winner'].value_counts().mean()
```

```
Out[26]:

19.375
```

From the above data it is observed that the team batting first has an winning average of 19.375

18. The teams winning after electing to field first.

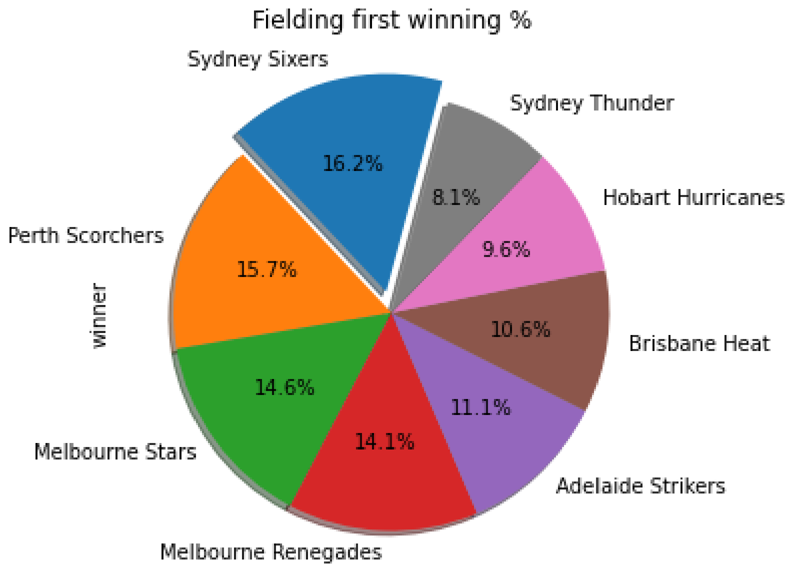
```
In [ ]:
```

```
#Analytical

df[df.toss_decision=="field"]['winner'].value_counts()
```

```
Out[27]:

Sydney Sixers          32
Perth Scorchers        31
Melbourne Stars        29
Melbourne Renegades    28
Adelaide Strikers      22
Brisbane Heat          21
Hobart Hurricanes      19
Sydney Thunder         16
Name: winner, dtype: int64
```

```
In [ ]:
```

```
#Pie plot

plt.figure(figsize=(5,5))

df[df.toss_decision=="field"]['winner'].value_counts().plot.pie(startangle=75, shadow=True,
explode =(0.1,0,0,0,0,0,0,0))

plt.title("Fielding first winning %")

plt.show()
```



From the above data it is observed that after winning the toss and electing to field first the team "Sydney Sixers" has highest number of winnings and winning% i.e 32 winnings and 16.2 winning% respectievely

19. Winning average after feilding first.

```
In [ ]:
```

```
df[df.toss_decision=="field"]['winner'].value_counts().mean()
```

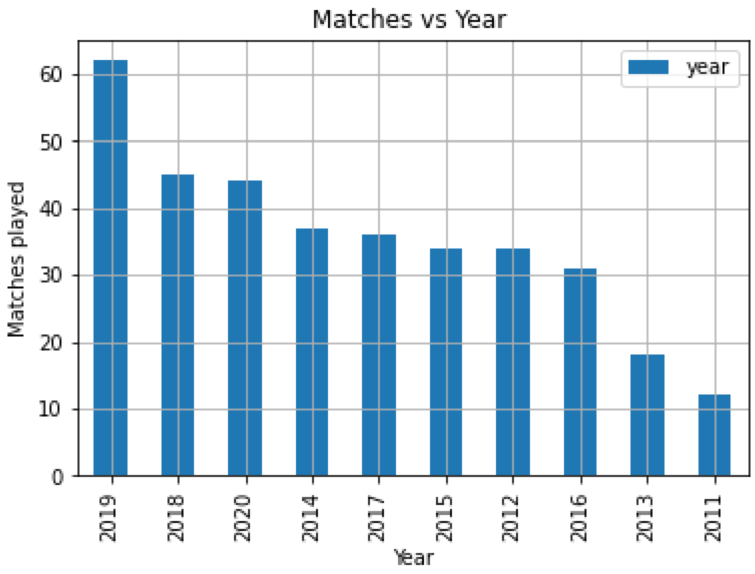
```
Out[29]:

24.75
```

From the above data it is observed that the team fielding first has an winning average of 24.75

20. Matches played each year.

In [ ]:

```
df["year"] = df["date"].str[:4].astype(int)
df.year.value_counts().plot(kind = 'bar')

plt.title("Matches vs Year ")

plt.xlabel("Year")
plt.ylabel("Matches played")

plt.grid(True)
plt.legend()
plt.show()
```



21. Average matches played per year.

In [ ]:

```
df["year"] = df["date"].str[:4].astype(int)
df.year.value_counts().mean()
```

Out[31]:

```
35.3
```

22. Total number of venues.

In [ ]:

```
tv = df.venue.unique()
c= pd.DataFrame(tv, columns=["total venues"])
c
```

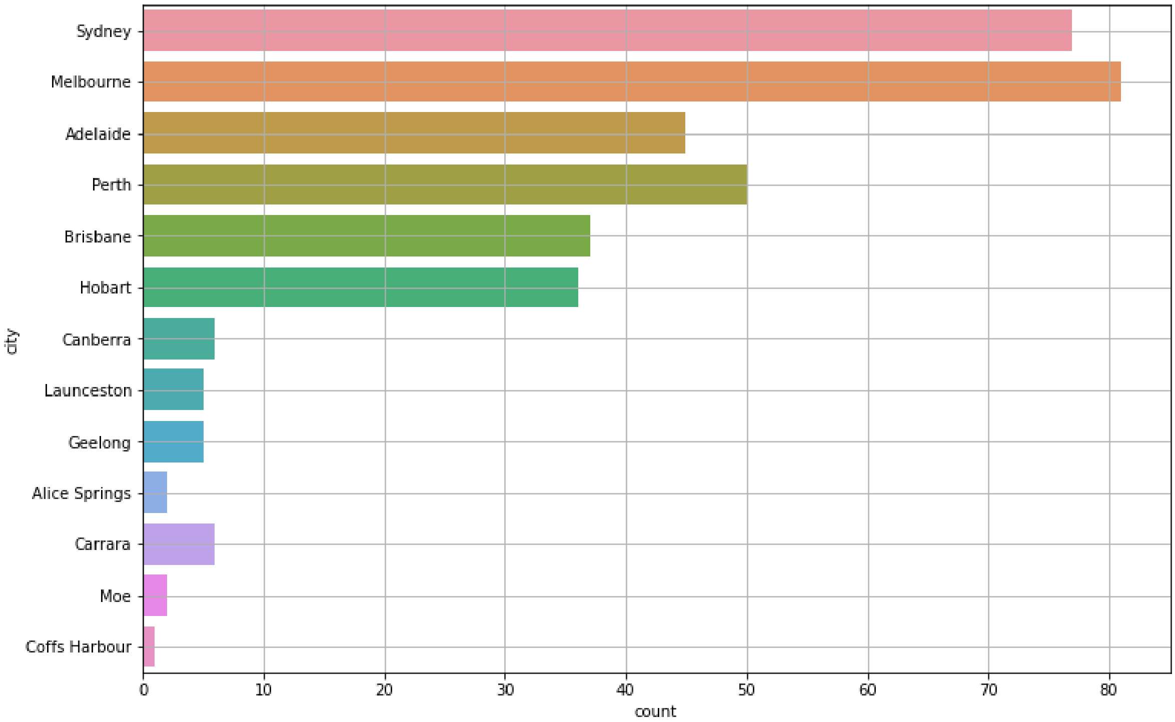
Out[32]:

```
                                     total venues
 0                           Sydney Cricket Ground
 1                        Melbourne Cricket Ground
 2                                    Adelaide Oval
 3   Western Australia Cricket Association Ground
 4          Brisbane Cricket Ground, Woolloongabba
 5                                   Bellerive Oval
 6                               Docklands Stadium
 7                               Stadium Australia
 8                       Sydney Showground Stadium
 9                                      Manuka Oval
10                                 W.A.C.A. Ground
11                                  Aurora Stadium
12       Simonds Stadium, South Geelong, Victoria
13                                    Traeger Park
14                                   Perth Stadium
15                                    Carrara Oval
16                         Geelong Cricket Ground
17                          Ted Summerton Reserve
18                    International Sports Stadium
```

23. City wise match count.

In [ ]:

```
plt.figure(figsize=(12,8))
sns.countplot(data=df, y="city")
plt.grid(True)
```

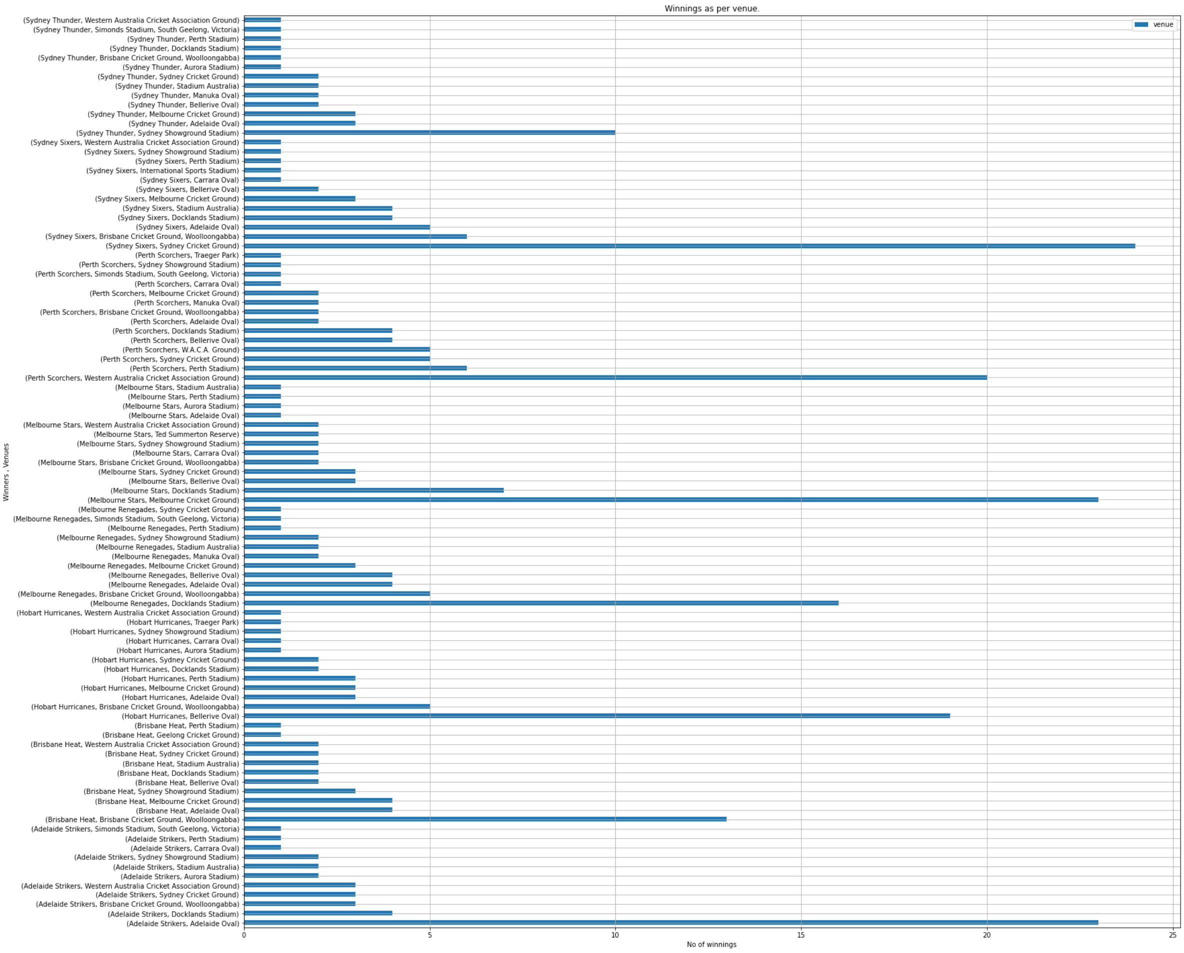


From the above data it is observed that most number of matches are played in "Melbourne" city.

24. Winning team according to the venue.

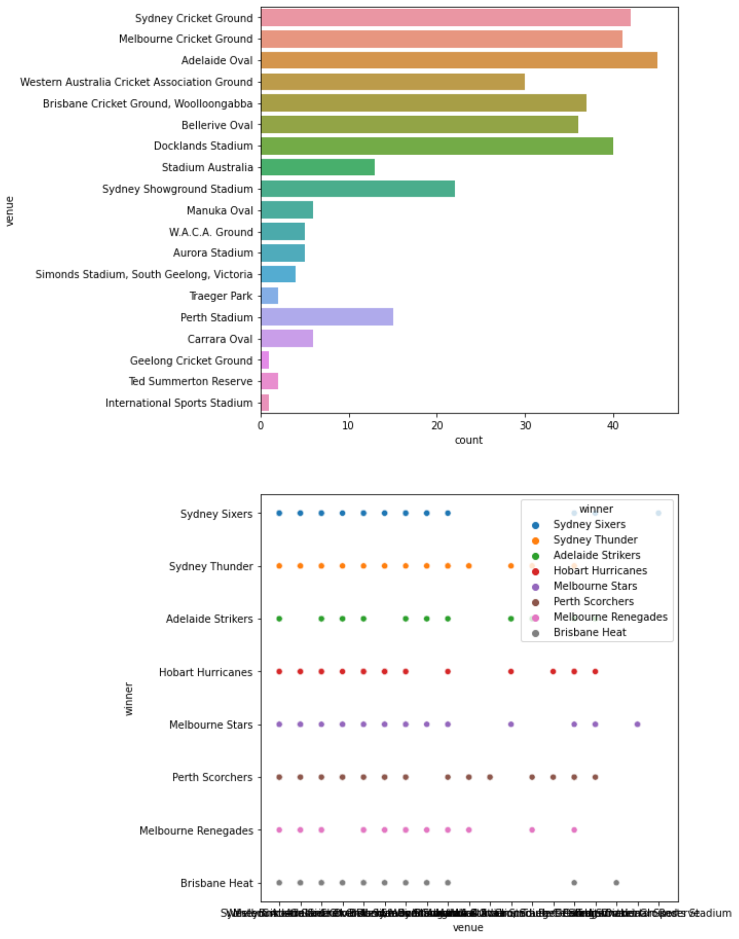
In [ ]:

```
plt.figure(figsize=(25,25))

df.groupby("winner")["venue"].value_counts().plot(kind="barh")

plt.title("Winnings as per venue.")

plt.xlabel("No of winnings")
plt.ylabel("Winners , Venues")

plt.grid(True)
plt.legend()

plt.show()
```



In [ ]:

```
fig,axes = plt.subplots(2,2,figsize=(16,16))

sns.countplot(data=df,y="venue", ax= axes[0,0])
sns.boxplot(data=df, x=df["winner"].value_counts() , ax= axes[0,1])
sns.scatterplot(data=df, x="venue", y="winner", hue="winner", ax=axes[1,0])
sns.boxplot(data=df, x=df["venue"].value_counts() , ax= axes[1,1])

plt.show()
```



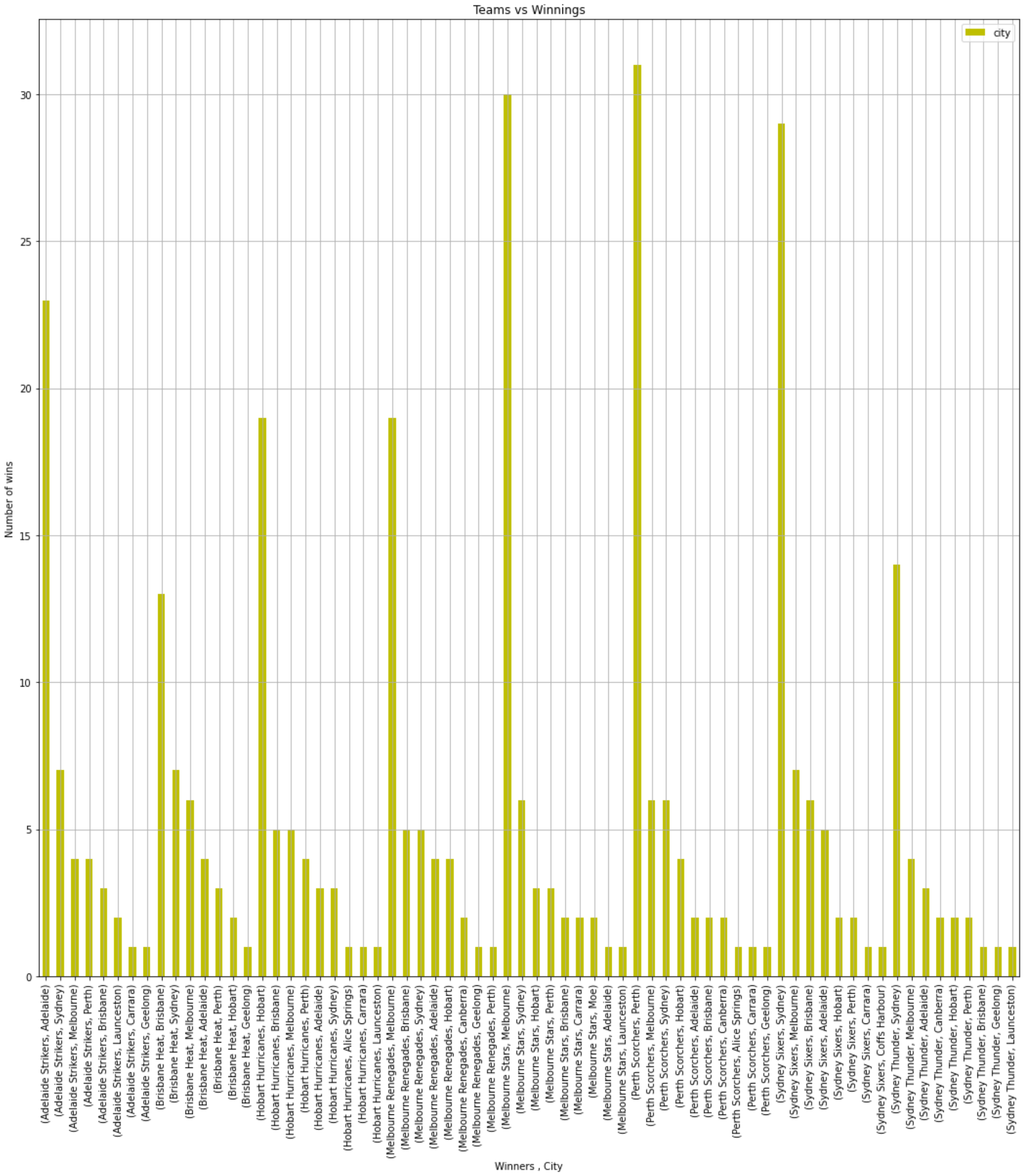
25. Home ground performance.

In [ ]:

```
plt.figure(figsize=(18,18))
df.groupby("winner")["city"].value_counts().plot(kind="bar" , color="y")

plt.title("Teams vs Winnings")
plt.xlabel("Winners , City")
plt.ylabel("Number of wins")
plt.grid(True)
plt.legend()
```

Out[36]:

```
<matplotlib.legend.Legend at 0x7fb7d86935d0>
```



From the above data it is obsereved that the number of wins are maximum when the team plays in its home ground.

26. Teams having a good record of run chase.

```
In [ ]:
```

```
df[df.result=="wickets"]['winner'].value_counts()
```

```
Out[37]:
```

```
Sydney Sixers          33
Melbourne Stars        29
Perth Scorchers        27
Melbourne Renegades    23
Hobart Hurricanes      19
Adelaide Strikers      17
Brisbane Heat          17
Sydney Thunder         16
Name: winner, dtype: int64
```

From the above data it is obsereved that the team "Sydney Sixers" is best at chasing the target.

27. Average of winning with respect to wickets.

```
In [ ]:
```

```
df[df.result=="wickets"]['winner'].value_counts().mean()
```

```
Out[38]:
```

```
22.625
```

28. Teams having good record of defending the target.

In [ ]:

```
df[df.result=="runs"]['winner'].value_counts()
```

Out[39]:

```
Adelaide Strikers      28
Perth Scorchers        28
Hobart Hurricanes      23
Melbourne Stars        20
Brisbane Heat          19
Melbourne Renegades    18
Sydney Sixers          18
Sydney Thunder         14
Name: winner, dtype: int64
```

From the above data it is obsereved that the team "Adelaide Strikers" is best at defending the target.

29. Average of winning with respect to runs.

In [ ]:

```
df[df.result=="runs"]['winner'].value_counts().mean()
```

Out[40]:

```
21.0
```

30. Top 4 winning teams of the league.

In [ ]:

```
df.winner.value_counts().head(4)
```

Out[41]:

```
Perth Scorchers      56
Sydney Sixers        53
Melbourne Stars      50
Adelaide Strikers    45
Name: winner, dtype: int64
```

31. Team with least number of wins.

In [ ]:

```
#Analytical

df.winner.value_counts().idxmin()
```

Out[42]:

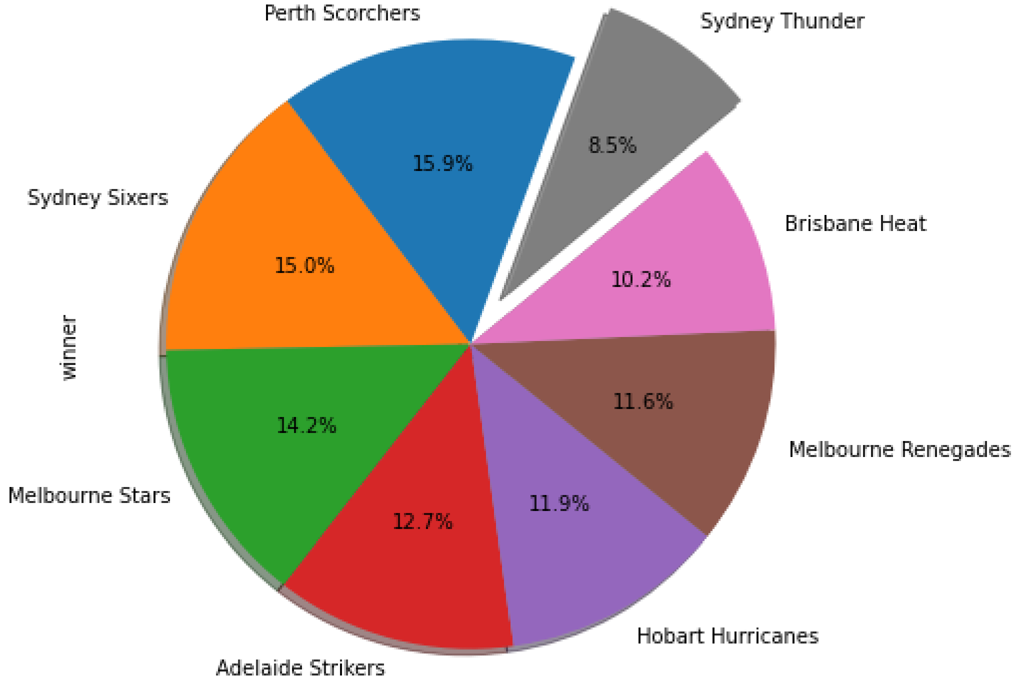
```
'Sydney Thunder'
```

In [ ]:

```
#Pie plot

plt.figure(figsize = (7,7))
df.winner.value_counts().plot.pie(startangle=70, shadow=True, autopct="%1.1f%%",explode =(0
```

Out[43]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb7d8764990>
```



From the above data it is obsereved that the team having least number of wins is "Sydney Thunder".

32. List of players winning "Player of the match" title.

In [ ]:

```
df["player_of_match"]
```

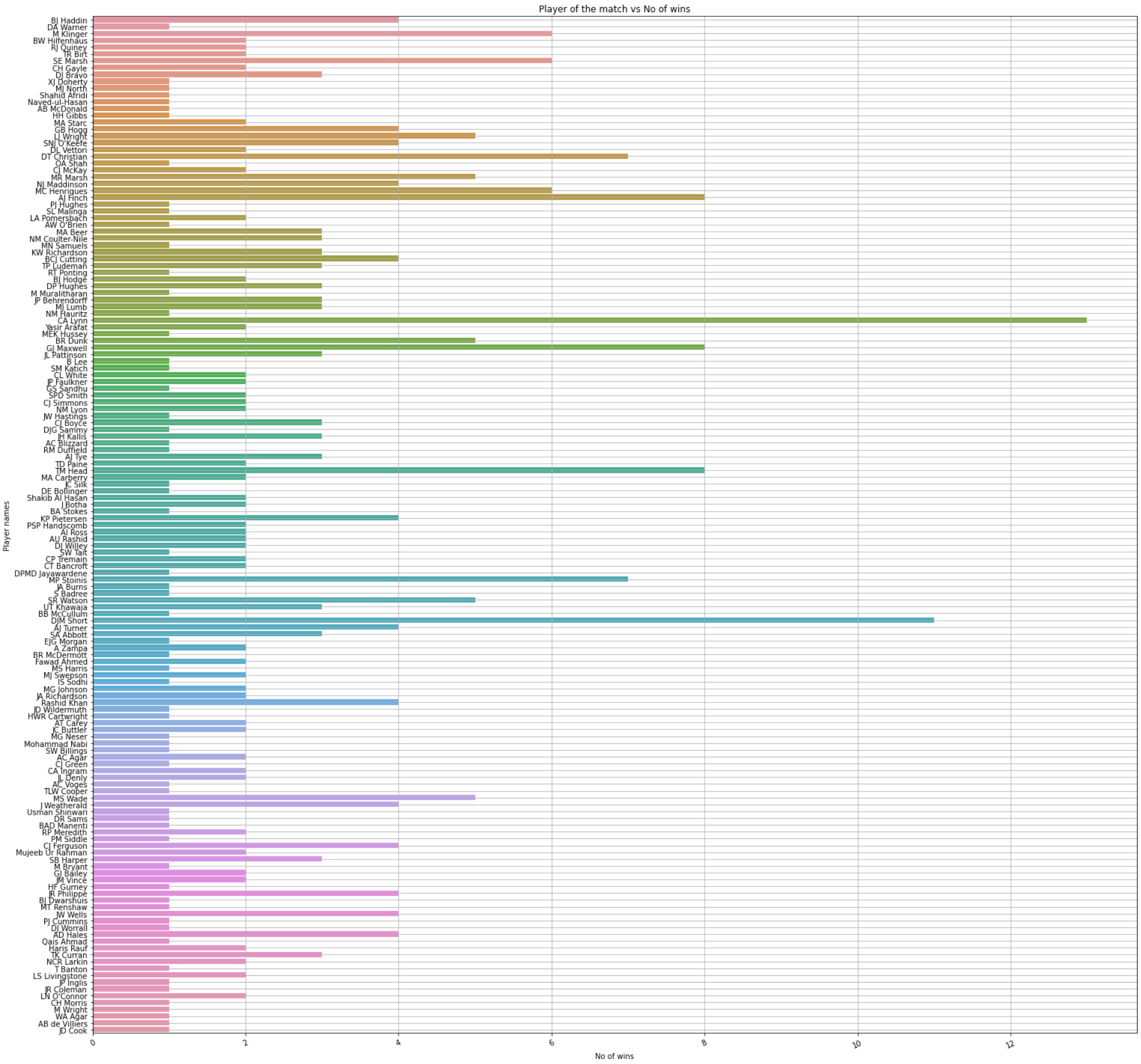
Out[44]:

```
0          BJ Haddin
1          DA Warner
2          M Klinger
3      BW Hilfenhaus
4          RJ Quiney
           ...
348          JD Cook
349      SNJ O'Keefe
350         AD Hales
351       NCR Larkin
352      JR Philippe
Name: player_of_match, Length: 353, dtype: object
```

33. The player to win most number of "Player Of The Match" title.

In [ ]:

```
plt.figure(figsize=(25,25))

sns.countplot(data=df , y="player_of_match" )

plt.title("Player of the match vs No of wins")

plt.xlabel("No of wins")
plt.ylabel("Player names")

plt.xticks(rotation=25)
plt.yticks(rotation=0)

plt.grid(True)

plt.show()
```

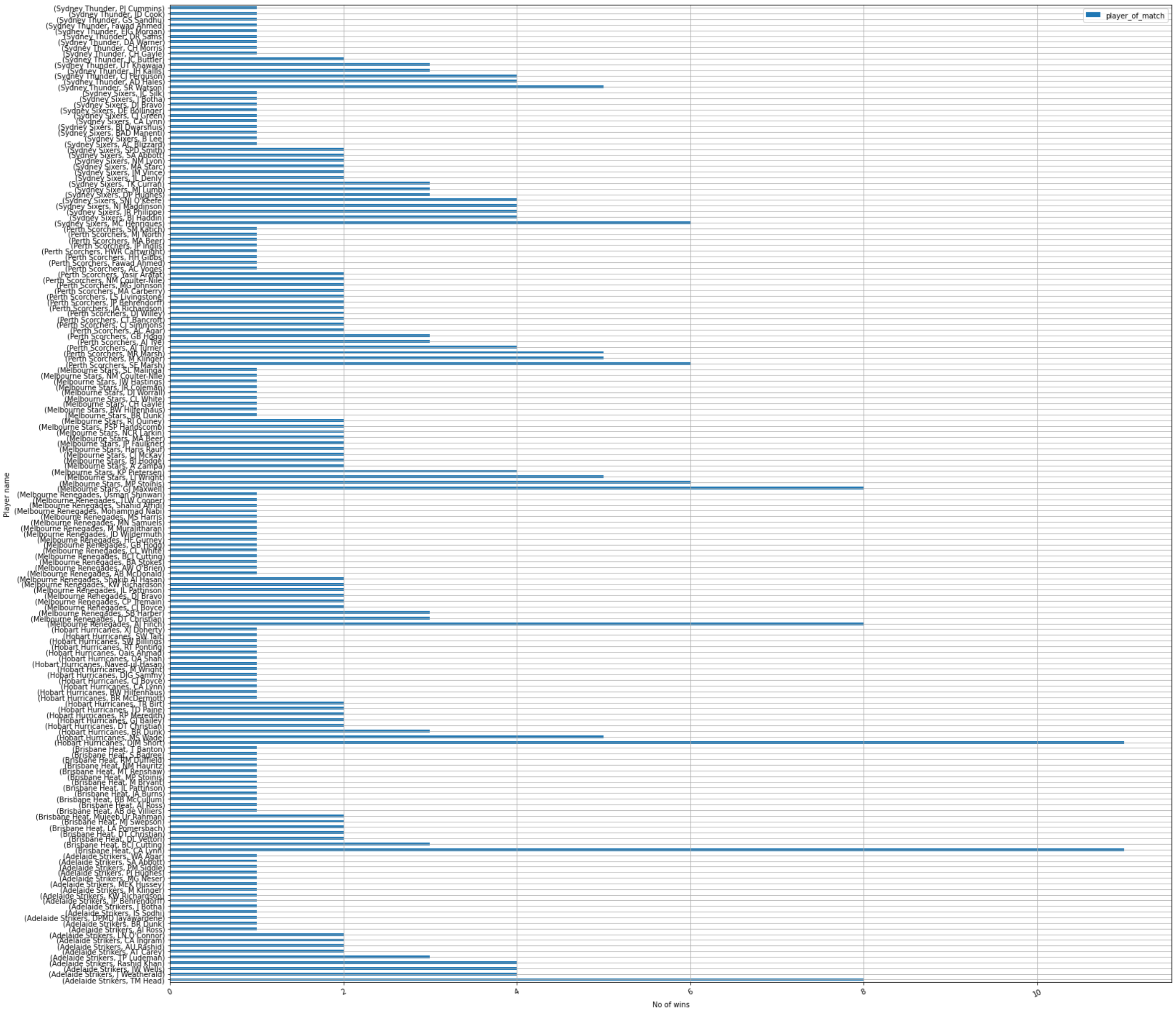


From above data it is obsereved that "CA Lynn" has won the most number of "Player of the match" title.

34. Contribution of Player of the match in winning the games.

In [ ]:

```
plt.figure(figsize=(25,25))

df.groupby("winner").player_of_match.value_counts().plot(kind=("barh"))

plt.xlabel("No of wins")
plt.ylabel("Player name")
plt.legend()

plt.xticks(rotation=25)
plt.yticks(rotation=0)

plt.grid(True)

plt.show()
```



From the above data it is observed that whenever the best player of the team i.e "Player of the match" performs well the chances of winning is higher.

35. Top 5 players.

In [ ]:

```
df["player_of_match"].value_counts().head()
```

Out[47]:

```
CA Lynn       13
DJM Short     11
AJ Finch       8
TM Head        8
GJ Maxwell     8
Name: player_of_match, dtype: int64
```

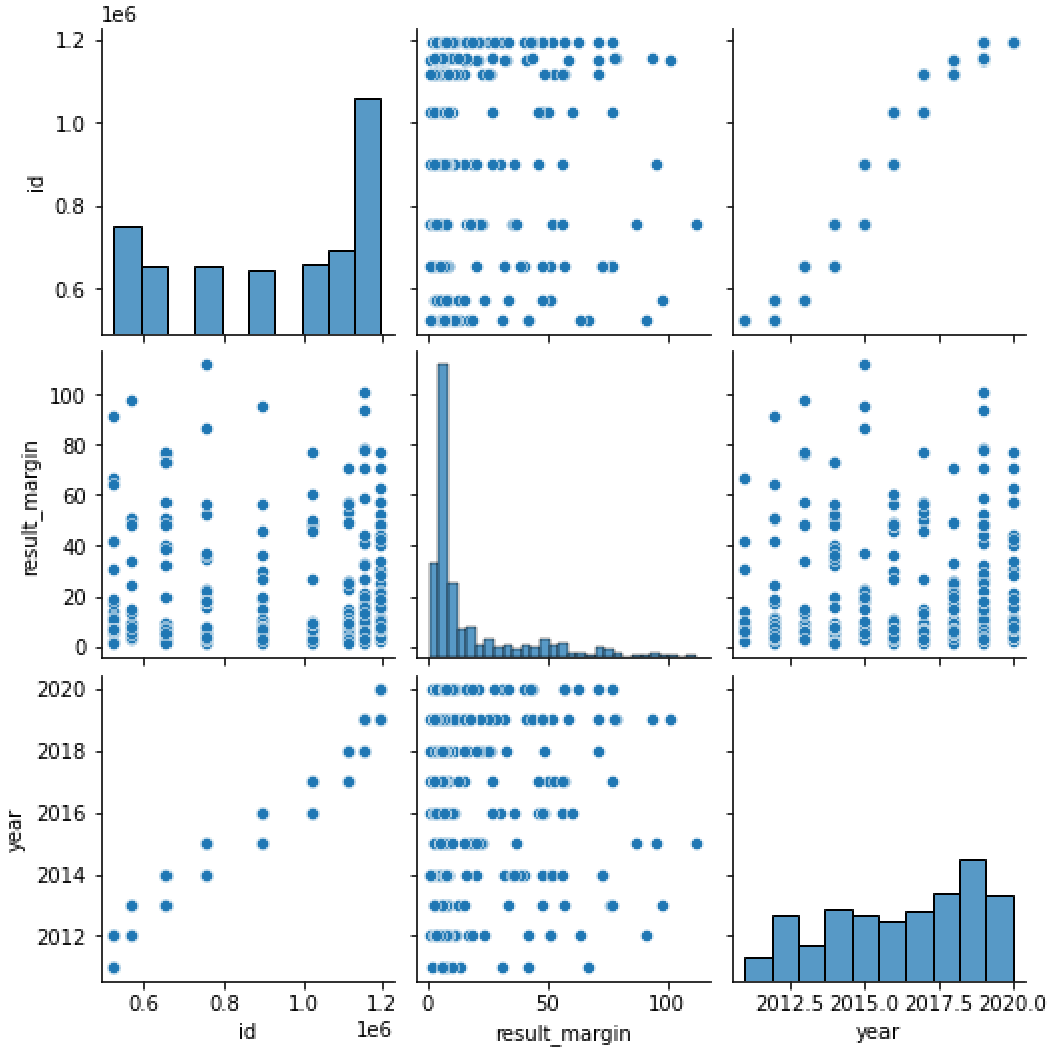
36. Pairplot representation.

In [ ]:

```
sns.pairplot(df)
```

Out[48]:

```
<seaborn.axisgrid.PairGrid at 0x7fb7d66b2510>
```



37. Correlation of data.

In [ ]:

```
df.corr()
```

Out[49]:

	id	result_margin	year
id	1.000000	-0.022588	0.973979
result_margin	-0.022588	1.000000	-0.008880
year	0.973979	-0.008880	1.000000

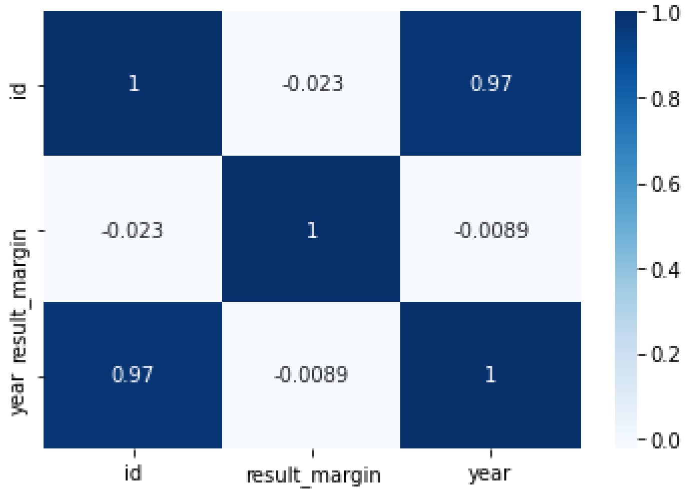
38. Heatmap.

In [ ]:

```
sns.heatmap(df.corr(), annot=True , cmap= "Blues")
```

Out[133]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb7c5556450>
```



As the correlation ranges from -1 to +1 and data have values between this range it can be said that there is a correlation.

1. There is a +ve correlation between "id" and "year" , hence it can be said that they are proportional to each other.
2. There is a -ve correlation between "id" and "result_margin" , hence it can be said that they are inversely proportional to each other.
3. There is a -ve correlation between "result_margin" and "year" , hence it can be said that they are inversely proportional to each other.

Conclusion

1. The team to win most number of matches is "Perth Scrochers" (From analysis performed in row no. 10).
2. The percentage of electing to bat first and electing to field first after winning the toss is 50%-50% (From analysis performed in row no. 15).

3. The winning percentage of teams electing to bat first is 16.1% (From analysis performed in row no. 16).
4. The winning percentage of teams electing to field first is 16.2% (From analysis performed in row no. 18).
5. The average number of matches played each year is 35 (From analysis performed in row no. 21).
6. Most number of matches are played in Melbourne city. (From analysis performed in row no. 23).
7. Teams while playing in their home ground perform better and have high winnings as compared to away grounds (From analysis performed in row no. 25).
8. The team to win least number of matches is "Sydney thunders" (From analysis performed in row no. 31).
9. The player to win most number of "Player of the match" title is "CA Lynn" (From analysis performed in row no. 33).

Solution

From the above analysis following points should be considered while deciding the game plan for winning the games -

1. As it is seen that the player of the match has a good contirbution in winning the games so the player winning most player of the match titles for the team must be put on priority.
2. As far as the decision based on toss is considered it is seen that the winning percentage of electing to field first after winning the toss is better than electing to bat first. So the team must elect to field first after winning the toss.
3. If in case the team loses the toss and is put to bat first then a good target is required to set on the board and picking wickets is must so that they can avoid the opponent to chase the target as the average of winning with respect to wickets is more than average of winning with respect to runs.
4. Whenever the team is performing in its home ground the winning rate is high. So while performing on the home ground the team should not implement major changes and play their normal game.

```
In [ ]:
```