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

In [3]:
```
df = pd.read_csv("IPL Matches 2008-2020.csv")
df.head()
```

Out[3]:

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

In [4]:
```
df.shape
```

Out[4]:
```
(816, 17)
```

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

Out[5]:

	id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	r

	id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	r
811	1216547	Dubai	2020-09-28	AB de Villiers	Dubai International Cricket Stadium	0	Royal Challengers Bangalore	Mumbai Indians	Mumbai Indians	field	Royal Challengers Bangalore	tie	
812	1237177	Dubai	2020-11-05	JJ Bumrah	Dubai International Cricket Stadium	0	Mumbai Indians	Delhi Capitals	Delhi Capitals	field	Mumbai Indians	runs	
813	1237178	Abu Dhabi	2020-11-06	KS Williamson	Sheikh Zayed Stadium	0	Royal Challengers Bangalore	Sunrisers Hyderabad	Sunrisers Hyderabad	field	Sunrisers Hyderabad	wickets	
814	1237180	Abu Dhabi	2020-11-08	MP Stoinis	Sheikh Zayed Stadium	0	Delhi Capitals	Sunrisers Hyderabad	Delhi Capitals	bat	Delhi Capitals	runs	
815	1237181	Dubai	2020-11-10	TA Boult	Dubai International Cricket Stadium	0	Delhi Capitals	Mumbai Indians	Delhi Capitals	bat	Mumbai Indians	wickets	

In [8]:
```
df.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
```

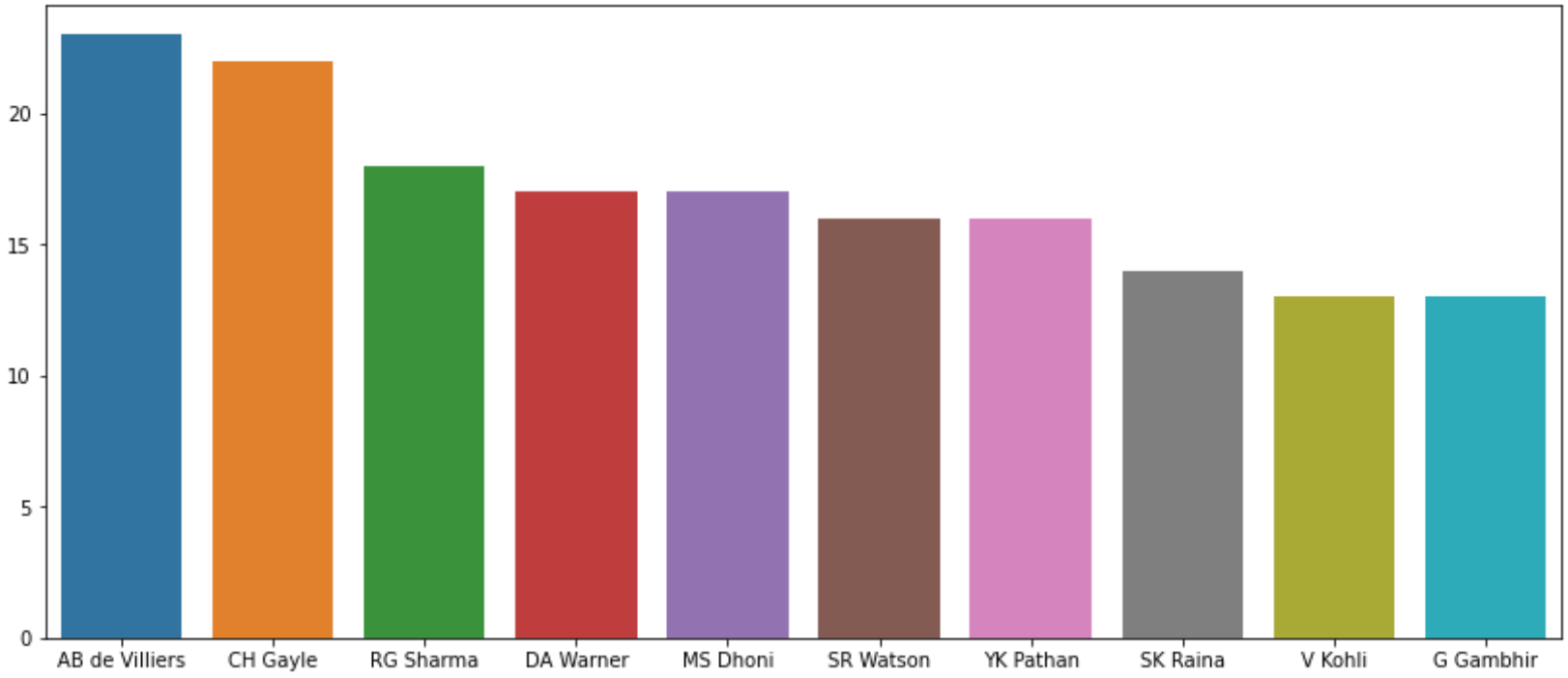
In [19]:
```
df["player_of_match"].value_counts()
```

Out[19]:
```
AB de Villiers     23
CH Gayle           22
RG Sharma          18
DA Warner          17
MS Dhoni           17
                   ..
GD McGrath          1
WPUJC Vaas          1
KMDN Kulasekara     1
R McLaren           1
BA Bhatt            1
Name: player_of_match, Length: 233, dtype: int64
```

In [67]:
```
plt.figure(figsize=(14,6))
sns.barplot(list(df["player_of_match"].value_counts()[0:10].keys()),list(df["player_of_match"].value_counts()[0:10]))
```

```
C:\Users\U.R Computer\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
es as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[67]:
```
<AxesSubplot:>
```



In [54]:
```
df.team1.unique()
```

Out[54]:
```
array(['Royal Challengers Bangalore', 'Kings XI Punjab',
       'Delhi Daredevils', 'Mumbai Indians', 'Kolkata Knight Riders',
       'Rajasthan Royals', 'Deccan Chargers', 'Chennai Super Kings',
       'Kochi Tuskers Kerala', 'Pune Warriors', 'Sunrisers Hyderabad',
       'Gujarat Lions', 'Rising Pune Supergiants',
       'Rising Pune Supergiant', 'Delhi Capitals'], dtype=object)
```

In [68]:
```
df.isnull().sum()
```

Out[68]:
```
id                 0
city              13
date               0
player_of_match    4
venue              0
neutral_venue      0
team1              0
team2              0
```

```
toss_winner          0
toss_decision        0
winner               4
result               4
result_margin       17
eliminator           4
method             797
umpire1              0
umpire2              0
dtype: int64
```

In [70]:
```
df_useful = df.drop(["umpire1","umpire2","method","id"],axis=1)
df_useful = df_useful.dropna()
df_useful.isnull().sum()
```

Out[70]:
```
city              0
date              0
player_of_match   0
venue             0
neutral_venue     0
team1             0
team2             0
toss_winner       0
toss_decision     0
winner            0
result            0
result_margin     0
eliminator        0
dtype: int64
```

In [72]:
```
df_useful.shape
```

Out[72]:
```
(786, 13)
```

In [73]:
```
from datetime import datetime
df_useful.date = pd.to_datetime(df_useful["date"])
df_useful.sample(3)
```

Out[73]:

	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	resu

	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	resu
630	Pune	2017-05-14	JD Unadkat	Maharashtra Cricket Association Stadium	0	Rising Pune Supergiant	Kings XI Punjab	Rising Pune Supergiant	field	Rising Pune Supergiant	wickets	
471	Visakhapatnam	2015-04-18	JP Duminy	Dr. Y.S. Rajasekhara Reddy ACA-VDCA Cricket St...	0	Sunrisers Hyderabad	Delhi Daredevils	Delhi Daredevils	bat	Delhi Daredevils	runs	
258	Delhi	2012-04-10	M Morkel	Feroz Shah Kotla	0	Delhi Daredevils	Chennai Super Kings	Delhi Daredevils	field	Delhi Daredevils	wickets	

In [142…
```
df_useful = df_useful.replace({'Sunrisers Hyderabad':"SRH", 'Mumbai Indians': "MI", 'Gujarat Lions':"GL",
       'Rising Pune Supergiant': "Pune", 'Royal Challengers Bangalore': "RCB",
       'Kolkata Knight Riders': "KKR", 'Delhi Daredevils': "Delhi", 'Kings XI Punjab':"Punjab",
       'Chennai Super Kings':"CSK", 'Rajasthan Royals': "RR", 'Deccan Chargers':"SRH",
       'Kochi Tuskers Kerala':"KTK", 'Pune Warriors':"Pune", 'Delhi Capitals':"Delhi"})
```

In [143…
```
df_useful.sample(3)
```

Out[143…

	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	result_margin	eli
769	Abu Dhabi	2020-11-02	A Nortje	Sheikh Zayed Stadium	0	RCB	Delhi	Delhi	field	Delhi	wickets	6.0	
32	Hyderabad	2008-05-11	SC Ganguly	Rajiv Gandhi International Stadium, Uppal	0	SRH	KKR	KKR	bat	KKR	runs	23.0	
7	Chennai	2008-04-23	ML Hayden	MA Chidambaram Stadium, Chepauk	0	CSK	MI	MI	field	CSK	runs	6.0	

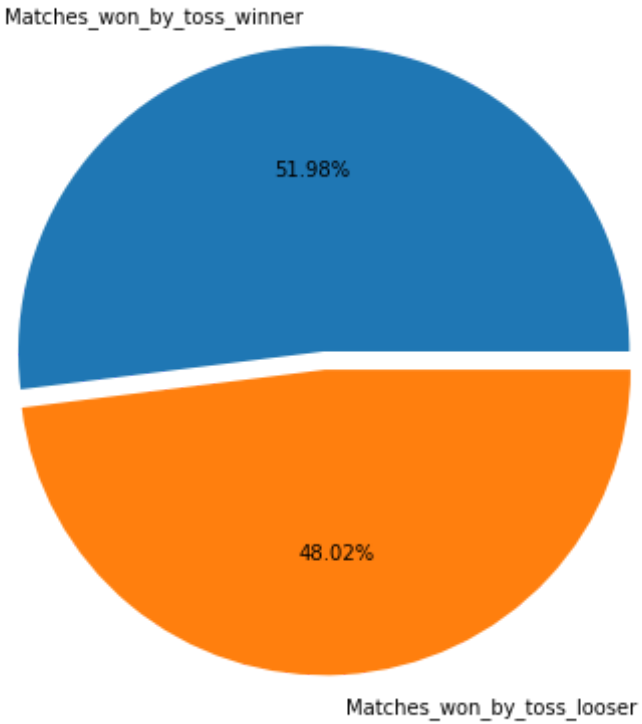
In [144…
```
df["winner"].value_counts()
```

Out[144…
```
Mumbai Indians                 120
Chennai Super Kings            106
Kolkata Knight Riders           99
Royal Challengers Bangalore     91
Kings XI Punjab                 88
Rajasthan Royals                81
Delhi Daredevils                67
Sunrisers Hyderabad             66
Deccan Chargers                 29
Delhi Capitals                  19
Gujarat Lions                   13
Pune Warriors                   12
Rising Pune Supergiant          10
Kochi Tuskers Kerala             6
Rising Pune Supergiants          5
Name: winner, dtype: int64
```

In [145…
```
toss_factor = df_useful.toss_winner == df_useful.winner
toss_factor.value_counts()
```

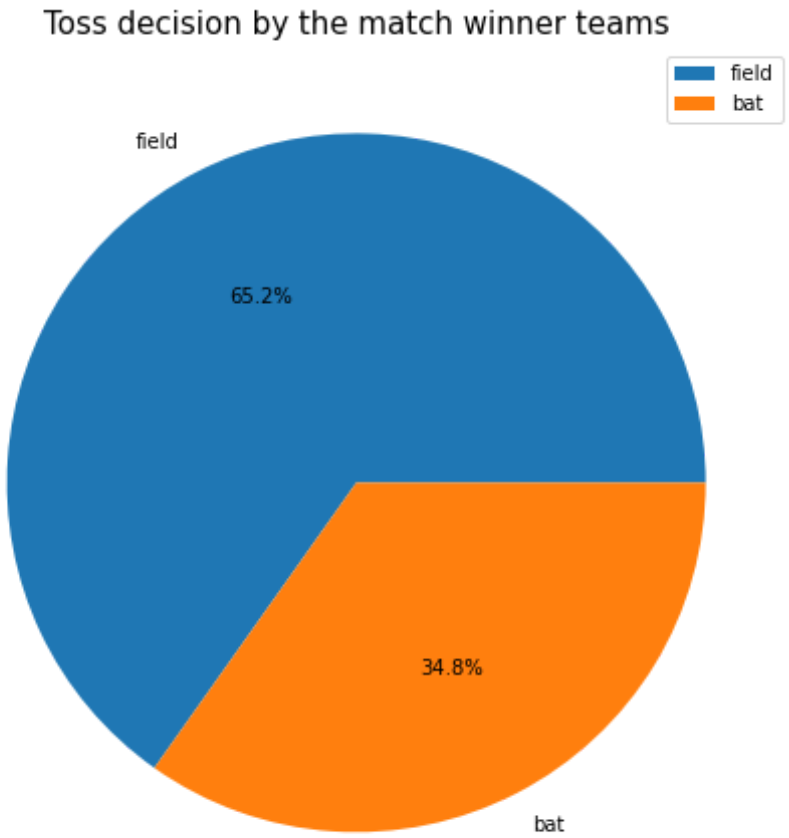
Out[145…
```
True     408
False    378
dtype: int64
```

In [146…
```
toss_data = { "Matches_won_by_toss_winner" : 393,
              "Matches_won_by_toss_looser" : 363}
toss_data = pd.Series(toss_data)
plt.figure(figsize=(7,7))
plt.pie(toss_data,autopct="%.2f%%",explode=[0.03]*2,labels=toss_data.index);
```



In [147…

```
#Toss decision by winner team
plt.figure(figsize=(10,8))
plt.pie(df_useful['toss_decision'][df_useful['toss_winner'] == df_useful['winner']].value_counts() , autopct = '%1.1f
plt.title('Toss decision by the match winner teams' , fontsize = 15)
plt.legend(fontsize = 10)
plt.show()
```



In [148…
```
batting_first = df_useful[df_useful["result"]=="runs"]
batting_first.sample(3)
```

Out[148…

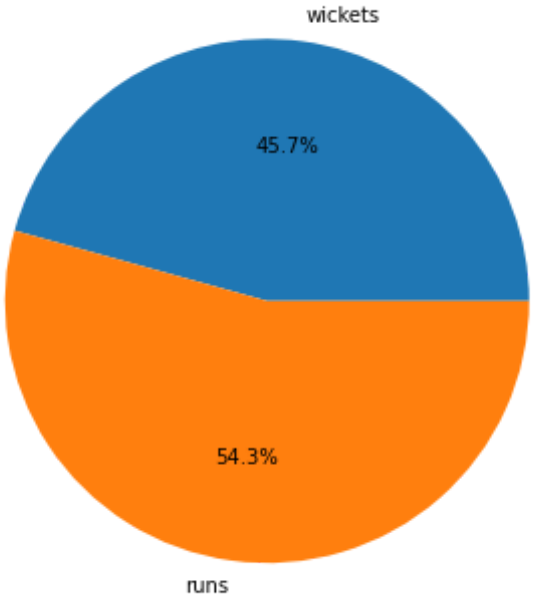
	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	result_margin	elimin
652	Pune	2018-04-20	SR Watson	Maharashtra Cricket Association Stadium	0	CSK	RR	RR	field	CSK	runs	64.0	
729	Delhi	2019-04-18	HH Pandya	Feroz Shah Kotla	0	Delhi	MI	MI	bat	MI	runs	40.0	

	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	result_margin	elimin
119	Chennai	2010-03-14	WPUJC Vaas	MA Chidambaram Stadium, Chepauk	0	CSK	SRH	SRH	bat	SRH	runs	31.0	

In [149…

```
batting_sec = df_useful[df_useful["result"]=="wickets"]
winner_data = { "Matches_won_by_bat_first" :len(batting_first),
                "Matches_won_by_bat_sec" : len(batting_sec)}
winner_data = pd.Series(winner_data)
print("Matches_won_by_bat_first :",len(batting_first),"and","Matches_won_by_bat_sec:",len(batting_sec))
plt.figure(figsize=(10,6))
plt.pie(winner_data,labels = df_useful['result'].value_counts().index,autopct = '%1.1f%%')
plt.show()
```

```
Matches_won_by_bat_first : 359 and Matches_won_by_bat_sec: 427
```

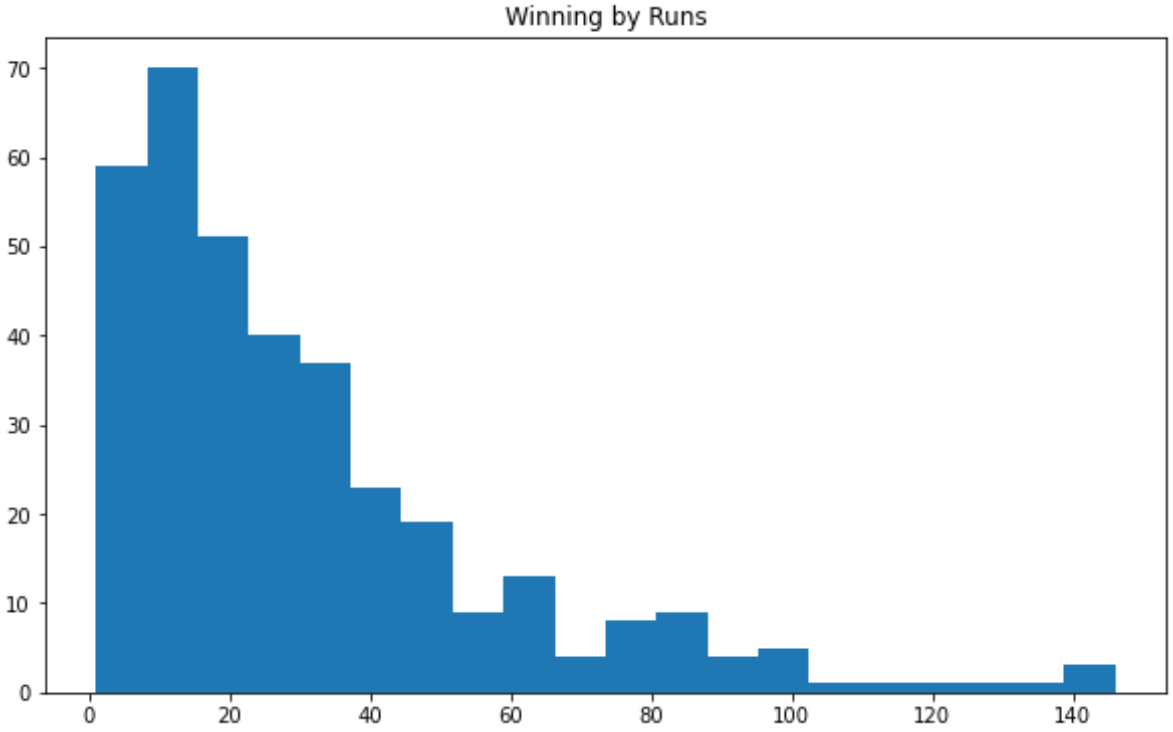


In [174…

```
plt.figure(figsize=(10,6))
```

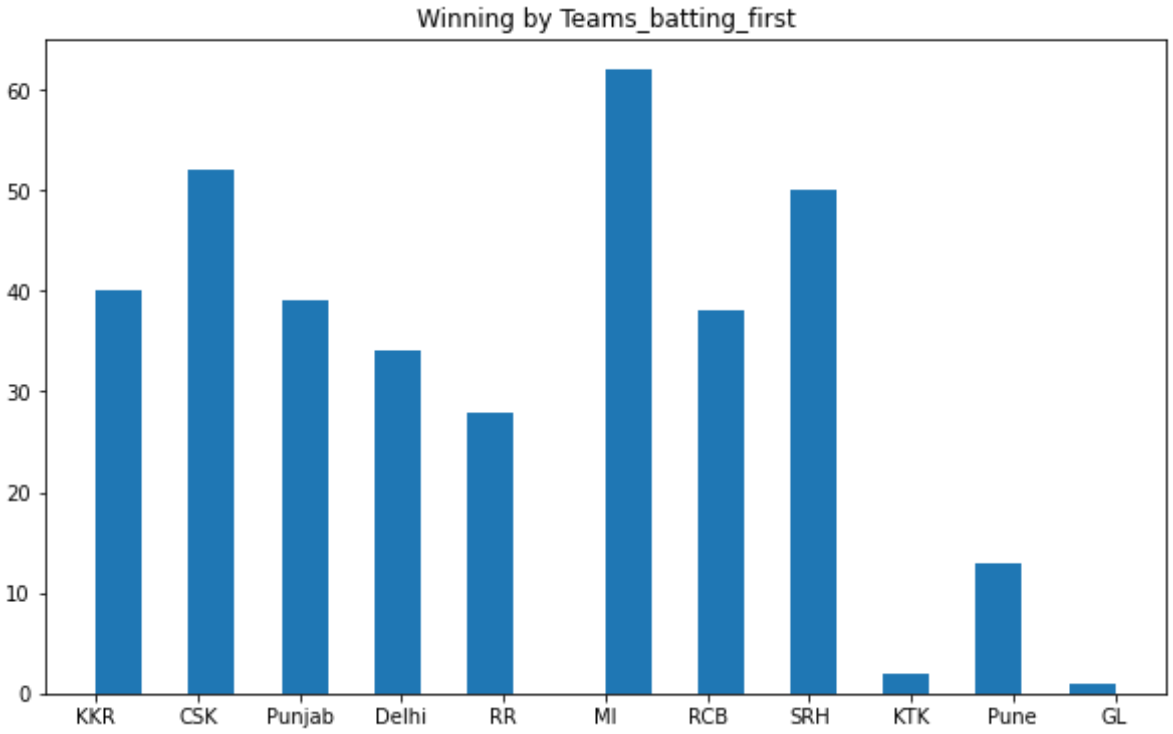
```
plt.hist(batting_first["result_margin"],bins=20)
plt.title("Winning by Runs")
```

Out[174… Text(0.5, 1.0, 'Winning by Runs')



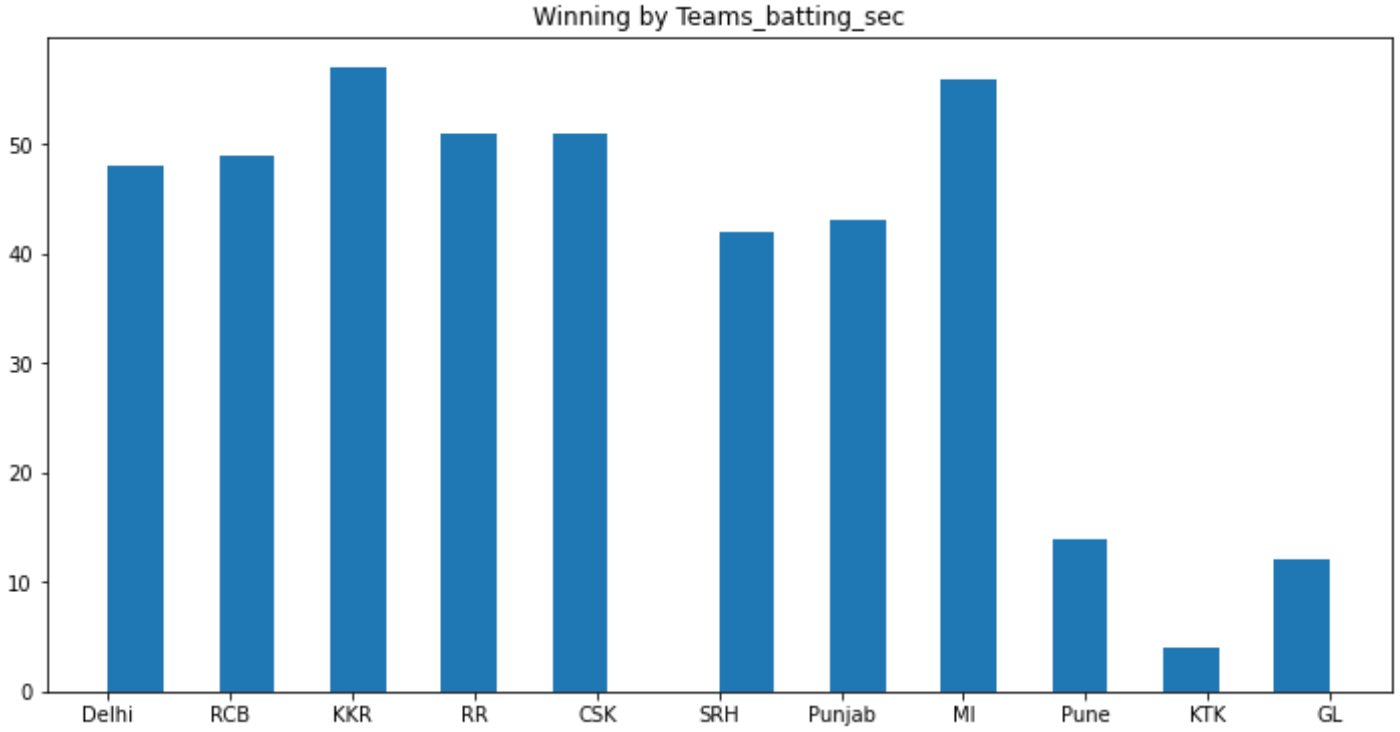
In [170…
```
plt.figure(figsize=(10,6))
plt.hist(batting_first["winner"],bins=22)
plt.title("Winning by Teams_batting_first")
```

Out[170… Text(0.5, 1.0, 'Winning by Teams_batting_first')



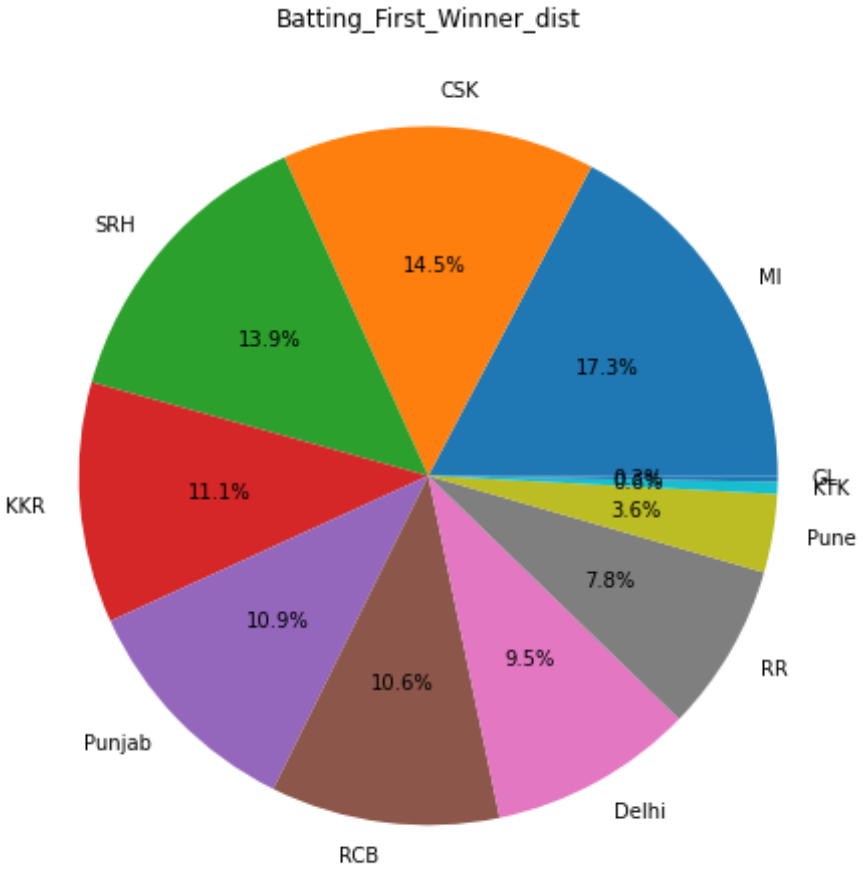
In [167…
```
plt.figure(figsize=(12,6))
plt.hist(batting_sec["winner"],bins=22)
plt.title("Winning by Teams_batting_sec")
```

Out[167…
```
Text(0.5, 1.0, 'Winning by Teams_batting_sec')
```



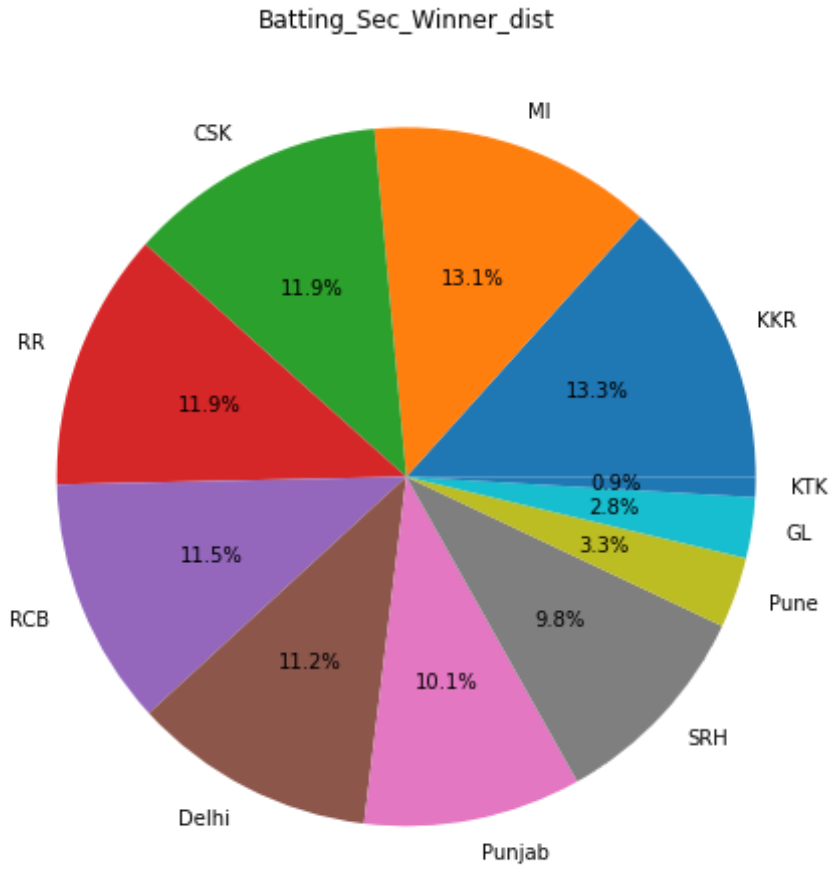
In [164…

```
plt.figure(figsize=(12,8))
plt.pie(list(batting_first["winner"].value_counts()),labels =list(batting_first["winner"].value_counts().keys()),auto
plt.title("Batting_First_Winner_dist")
plt.show()
```



In [165…

```
plt.figure(figsize=(12,8))
plt.pie(list(batting_sec["winner"].value_counts()),labels =list(batting_sec["winner"].value_counts().keys()),autopct
plt.title("Batting_Sec_Winner_dist")
plt.show()
```



In [175…
```
df_useful["city"].value_counts()
```

Out[175…
```
Mumbai           100
Kolkata           77
Delhi             72
Hyderabad         63
Bangalore         62
Chennai           56
Chandigarh        56
Jaipur            47
Pune              38
Abu Dhabi         27
```

```
Dubai              23
Durban             15
Bengaluru          14
Visakhapatnam      13
Centurion          12
Sharjah            12
Ahmedabad          11
Indore              9
Rajkot              9
Dharamsala          9
Johannesburg        8
Port Elizabeth      7
Cuttack             7
Ranchi              7
Cape Town           6
Raipur              6
Kochi               5
Kanpur              4
Kimberley           3
Nagpur              3
East London         3
Bloemfontein        2
Name: city, dtype: int64
```

In [181…

```
#building predictive model
from sklearn.preprocessing import LabelEncoder
var_mod = ['city','toss_decision','venue']
le = LabelEncoder()
for i in var_mod:
    df_useful[i] = le.fit_transform(df_useful[i])
df_useful.dtypes
```

Out[181…

```
city                        int32
date                datetime64[ns]
player_of_match             object
venue                        int32
neutral_venue                int64
team1                       object
team2                       object
toss_winner                 object
toss_decision                int32
winner                      object
result                      object
result_margin              float64
```

```
eliminator                object
dtype: object
```

In [185…
```
df_useful_model = df_useful.drop(["city","date","player_of_match","neutral_venue","result_margin","result","eliminato
df_useful_model.head()
```

Out[185…

	venue	team1	team2	toss_winner	toss_decision	winner
0	14	RCB	KKR	RCB	1	KKR
1	23	Punjab	CSK	CSK	0	CSK
2	8	Delhi	RR	RR	0	Delhi
3	35	MI	RCB	MI	0	RCB
4	7	KKR	SRH	SRH	0	KKR

In [357…
```
encode = {'team1': {'MI':1,'KKR':2,'RCB':3,'Delhi':4,'CSK':5,'RR':6,'GL':8,'Punjab':9,'SRH':10,'Pune':11,'KTK':12,'PW
          'team2': {'MI':1,'KKR':2,'RCB':3,'Delhi':4,'CSK':5,'RR':6,'GL':8,'Punjab':9,'SRH':10,'Pune':11,'KTK':12,'PW
          'toss_winner': {'MI':1,'KKR':2,'RCB':3,'Delhi':4,'CSK':5,'RR':6,'GL':8,'Punjab':9,'SRH':10,'Pune':11,'KTK':
          'winner': {'MI':1,'KKR':2,'RCB':3,'Delhi':4,'CSK':5,'RR':6,'GL':8,'Punjab':9,'SRH':10,'Pune':11,'KTK':12,'P
df_processed = df_useful_model.replace(encode)
df_processed.sample(5)
```

Out[357…

	venue	team1	team2	toss_winner	toss_decision	winner
347	14	3	6	3	1	3
282	8	4	1	1	1	4
672	35	1	2	2	1	1
196	23	9	6	6	1	9
365	16	5	9	5	0	5

In [214…
```
dicVal = encode['winner']
i =dicVal['CSK'] #key value
print(i)
print(list(dicVal.keys())[list(dicVal.values()).index(i)]) #find key by value search
```

```
5
CSK
```

In [219…
```
df_processed.dtypes
```

Out[219…
```
venue            int32
team1            int64
team2            int64
toss_winner      int64
toss_decision    int32
winner           int64
dtype: object
```

In [308…
```
#Import models from scikit learn module:
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import KFold   #For K-fold cross validation
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn import metrics

#Generic function for making a classification model and accessing performance:
def classification_model(model, data, predictors, outcome):
    model.fit(data[predictors],data[outcome])
    predictions = model.predict(data[predictors])
    accuracy = metrics.accuracy_score(predictions,data[outcome])
    print('Accuracy : %s' % '{0:.3%}'.format(accuracy))
    kf = KFold(n_splits=7,random_state=1,shuffle=True)
    error = []
    for train,test in kf.split(data):
        train_predictors = (data[predictors].iloc[train,:])
        train_target = data[outcome].iloc[train]
        model.fit(train_predictors, train_target)
        error.append(model.score(data[predictors].iloc[test,:], data[outcome].iloc[test]))

    print('Cross-Validation Score : %s' % '{0:.3%}'.format(np.mean(error)))

    model.fit(data[predictors],data[outcome])
```

In [309…
```
from sklearn.ensemble import RandomForestRegressor
outcome_var=['winner']
predictor_var = ['team1','team2','toss_winner']
model = LogisticRegression()
classification_model(model, df_processed,predictor_var,outcome_var)
```

```
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vect
```

```
or y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel
().
  return f(*args, **kwargs)
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs fai
led to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vect
or y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel
().
  return f(*args, **kwargs)
```

```
Accuracy : 22.137%
```

```
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs fai
led to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vect
or y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel
().
  return f(*args, **kwargs)
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs fai
led to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vect
or y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel
().
  return f(*args, **kwargs)
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs fai
led to converge (status=1):
```

```
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vect
or y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel
().
  return f(*args, **kwargs)
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs fai
led to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vect
or y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel
().
  return f(*args, **kwargs)
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs fai
led to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vect
or y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel
().
  return f(*args, **kwargs)
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs fai
led to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
```

```
  n_iter_i = _check_optimize_result(
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vect
or y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel
().
  return f(*args, **kwargs)
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs fai
led to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vect
or y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel
().
  return f(*args, **kwargs)
Cross-Validation Score : 21.379%
C:\Users\U.R Computer\anaconda\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs fai
led to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

In [310…

```
model = RandomForestClassifier(n_estimators=100)
outcome_var = ['winner']
predictor_var = ['team1', 'team2', 'venue', 'toss_winner','venue','toss_decision']
classification_model(model, df_processed,predictor_var,outcome_var)
```

```
<ipython-input-308-5d90c3fa959a>:10: DataConversionWarning: A column-vector y was passed when a 1d array was expecte
d. Please change the shape of y to (n_samples,), for example using ravel().
  model.fit(data[predictors],data[outcome])
Accuracy : 86.260%
<ipython-input-308-5d90c3fa959a>:19: DataConversionWarning: A column-vector y was passed when a 1d array was expecte
d. Please change the shape of y to (n_samples,), for example using ravel().
  model.fit(train_predictors, train_target)
<ipython-input-308-5d90c3fa959a>:19: DataConversionWarning: A column-vector y was passed when a 1d array was expecte
d. Please change the shape of y to (n_samples,), for example using ravel().
  model.fit(train_predictors, train_target)
```

```
<ipython-input-308-5d90c3fa959a>:19: DataConversionWarning: A column-vector y was passed when a 1d array was expecte
d. Please change the shape of y to (n_samples,), for example using ravel().
  model.fit(train_predictors, train_target)
<ipython-input-308-5d90c3fa959a>:19: DataConversionWarning: A column-vector y was passed when a 1d array was expecte
d. Please change the shape of y to (n_samples,), for example using ravel().
  model.fit(train_predictors, train_target)
<ipython-input-308-5d90c3fa959a>:19: DataConversionWarning: A column-vector y was passed when a 1d array was expecte
d. Please change the shape of y to (n_samples,), for example using ravel().
  model.fit(train_predictors, train_target)
<ipython-input-308-5d90c3fa959a>:19: DataConversionWarning: A column-vector y was passed when a 1d array was expecte
d. Please change the shape of y to (n_samples,), for example using ravel().
  model.fit(train_predictors, train_target)
<ipython-input-308-5d90c3fa959a>:19: DataConversionWarning: A column-vector y was passed when a 1d array was expecte
d. Please change the shape of y to (n_samples,), for example using ravel().
  model.fit(train_predictors, train_target)
Cross-Validation Score : 49.491%
<ipython-input-308-5d90c3fa959a>:24: DataConversionWarning: A column-vector y was passed when a 1d array was expecte
d. Please change the shape of y to (n_samples,), for example using ravel().
  model.fit(data[predictors],data[outcome])
```

In [346…

```
#'team1', 'team2', 'venue', 'toss_winner','city','toss_decision'
team1='RCB'
team2='KKR'
toss_winner='RCB'
input=[dicVal[team1],dicVal[team2],'14',dicVal[toss_winner],'2','1']
input = np.array(input).reshape((1, -1))
output=model.predict(input)
print(list(dicVal.keys())[list(dicVal.values()).index(output)]) #find key by value search output
```

```
RCB
```

In [350…

```
#'team1', 'team2', 'venue', 'toss_winner','city','toss_decision'
team1='MI'
team2='CSK'
toss_winner='CSK'
input=[dicVal[team1],dicVal[team2],'23',dicVal[toss_winner],'14','0']
input_1 = np.array(input).reshape((1, -1))
output=model.predict(input_1)
print(list(dicVal.keys())[list(dicVal.values()).index(output)]) #find key by value search output
```

```
MI
```

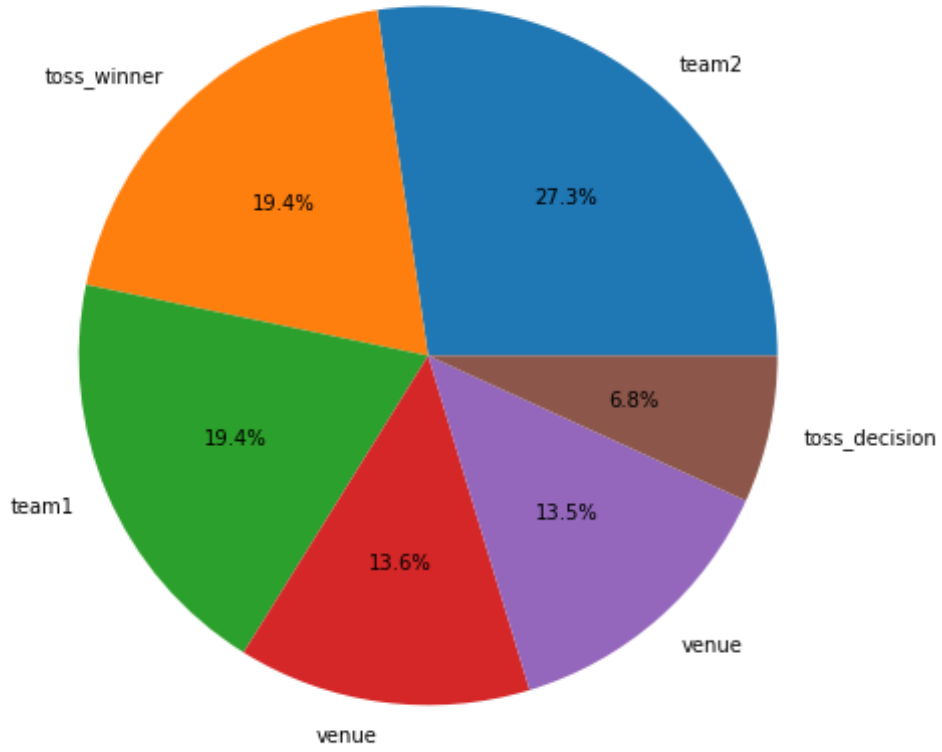
In [351…

```
imp_input = pd.Series(model.feature_importances_, index=predictor_var).sort_values(ascending=False)
print(imp_input)
```

```
team2              0.273235
toss_winner        0.194201
team1              0.193970
venue              0.135526
venue              0.134951
toss_decision      0.068117
dtype: float64
```

In [356…

```
plt.figure(figsize=(10,8))
plt.pie(imp_input,labels=imp_input.index,autopct = '%1.1f%%')
plt.show()
```



In [ ]: