Assignment3

July 20, 2024

[ ]:
```
Name:Krupanidhi Jena
Reg No:2141010032
Sec:P(16)
Celebal Technologies
```

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

[ ]:
```
3rd Assignment add on:
1: Upload the data set of your choice ?
```

[3]:
```
df=pd.read_csv('ipl2024 Matches.csv')
```

[5]:
```
df
```

[5]:

	id	date	team1	team2	toss_winner	decision	first_score	\
0	1	March 22,2024	Banglore	Chennai	Banglore	Bat	173	
1	2	March 23,2024	Delhi	Punjab	Punjab	Field	174	
2	3	March 23,2024	Kolkata	Hyderabad	Hyderabad	Field	208	
3	4	March 24,2024	Rajasthan	Lucknow	Rajasthan	Bat	193	
4	5	March 24,2024	Gujarat	Mumbai	Mumbai	Field	168	
..	..	…	…	…	…	…	…	
69	70	May 19,2024	Rajasthan	Kolkata	Kolkata	NaN	0	
70	71	May 21,2024	Hyderabad	Kolkata	Hyderabad	Bat	159	
71	72	May 22,2024	Banglore	Rajasthan	Rajasthan	Field	172	
72	73	May 24,2024	Hyderabad	Rajasthan	Rajasthan	Field	175	
73	74	May 26,2024	Hyderabad	Kolkata	Hyderabad	Bat	113	

	first_wkts	second_score	second_wkts	winner	player_of_the_match	\
0	6	176	4	Chennai	Mustafizur Rahman	
1	9	177	6	Punjab	Sam Curran	
2	7	204	7	Kolkata	Andre Russell	
3	4	173	6	Rajasthan	Sanju Samson	
4	6	162	9	Gujarat	Sai Sudharsan	
..	…	…	…	…	…	
69	0	0	0	Abandoned	NaN	
70	10	164	2	Kolkata	Mitchell Starc	

```
71           8          174           6  Rajasthan  Ravichandran Ashwin
72           9          139           7  Hyderabad        Shahbaz Ahmed
73          10          114           2    Kolkata       Mitchell Starc

           most_runs          most_wkts
0         Anuj Rawat  Mustafizur Rahman
1         Sam Curran      Kuldeep Yadav
2      Andre Russell        T Natarajan
3       Sanju Samson        Trent Boult
4      Dewald Brevis     Jasprit Bumrah
..               …                …
69               NaN              NaN
70      Shreyas Iyer     Mitchell Starc
71  Yashasvi Jaiswal         Avesh Khan
72      Dhruv Jurel       Shahbaz Ahmed
73    Venkatesh Iyer      Andre Russell

[74 rows x 14 columns]
```

[ ]:
```
2: Find the shape(number of rows (data points), columns (features))
```

[7]:
```
df.shape
```

[7]:
```
(74, 14)
```

[ ]:
```
3: find if any null values present ?
```

[9]:
```
null_values = df.isnull().sum()
print(null_values)
```

```
id                    0
date                  0
team1                 0
team2                 0
toss_winner           0
decision              3
first_score           0
first_wkts            0
second_score          0
second_wkts           0
winner                0
player_of_the_match   3
most_runs             3
most_wkts             3
dtype: int64
```

[ ]:
```
4: find the data types of each column ?
```

[11]:
```
data_types = df.dtypes
print(data_types)
```

```
id                      int64
date                   object
team1                  object
team2                  object
toss_winner            object
decision               object
first_score             int64
first_wkts              int64
second_score            int64
second_wkts             int64
winner                 object
player_of_the_match    object
most_runs              object
most_wkts              object
dtype: object
```

[ ]:
```
5: find the statistical information for each numerical column ?
```

[13]:
```
stats = df.describe()
print(stats)
```

```
              id  first_score  first_wkts  second_score  second_wkts
count  74.000000    74.000000   74.000000     74.000000    74.000000
mean   37.500000   180.554054    6.148649    169.054054     5.783784
std    21.505813    51.855474    2.469998     47.651386     2.934305
min     1.000000     0.000000    0.000000      0.000000     0.000000
25%    19.250000   162.250000    4.250000    145.250000     3.250000
50%    37.500000   182.500000    6.000000    173.500000     6.000000
75%    55.750000   208.000000    8.000000    198.250000     8.000000
max    74.000000   277.000000   10.000000    262.000000    10.000000
```

[ ]:
```
6: if there is any categorical column, convert them to numerical column ?
```

[21]:
```
cat=df['team1'].unique()
cat
```

[21]:
```
array(['Banglore', 'Delhi', 'Kolkata', 'Rajasthan', 'Gujarat', 'Punjab',
       'Chennai', 'Hyderabad', 'Lucknow', 'Mumbai'], dtype=object)
```

[23]:
```
df['team1'] = df['team1'].map({'Banglore': 0,'Delhi': 1, 'Kolkata':␣
 ↪2,'Rajasthan': 4,'Gujarat': 5,'Punjab': 6,'Chennai': 7,'Hyderabad':␣
 ↪8,'Lucknow': 9,'Mumbai': 10})
df['team2'] = df['team2'].map({'Banglore': 0,'Delhi': 1, 'Kolkata':␣
 ↪2,'Rajasthan': 4,'Gujarat': 5,'Punjab': 6,'Chennai': 7,'Hyderabad':␣
 ↪8,'Lucknow': 9,'Mumbai': 10})
```

```
df['toss_winner'] = df['toss_winner'].map({'Banglore': 0,'Delhi': 1, 'Kolkata':␣
 ↪2,'Rajasthan': 4,'Gujarat': 5,'Punjab': 6,'Chennai': 7,'Hyderabad':␣
 ↪8,'Lucknow': 9,'Mumbai': 10})
df['winner'] = df['winner'].map({'Banglore': 0,'Delhi': 1, 'Kolkata':␣
 ↪2,'Rajasthan': 4,'Gujarat': 5,'Punjab': 6,'Chennai': 7,'Hyderabad':␣
 ↪8,'Lucknow': 9,'Mumbai': 10})
df
```

[23]:

	id	date	team1	team2	toss_winner	decision	first_score	\
0	1	March 22,2024	0	7	0.0	Bat	173	
1	2	March 23,2024	1	6	6.0	Field	174	
2	3	March 23,2024	2	8	8.0	Field	208	
3	4	March 24,2024	4	9	4.0	Bat	193	
4	5	March 24,2024	5	10	10.0	Field	168	
..	..	…	…	…	…	…	…	
69	70	May 19,2024	4	2	2.0	NaN	0	
70	71	May 21,2024	8	2	8.0	Bat	159	
71	72	May 22,2024	0	4	4.0	Field	172	
72	73	May 24,2024	8	4	4.0	Field	175	
73	74	May 26,2024	8	2	8.0	Bat	113	

	first_wkts	second_score	second_wkts	winner	player_of_the_match	\
0	6	176	4	7.0	Mustafizur Rahman	
1	9	177	6	6.0	Sam Curran	
2	7	204	7	2.0	Andre Russell	
3	4	173	6	4.0	Sanju Samson	
4	6	162	9	5.0	Sai Sudharsan	
..	…	…	…	…	…	
69	0	0	0	NaN	NaN	
70	10	164	2	2.0	Mitchell Starc	
71	8	174	6	4.0	Ravichandran Ashwin	
72	9	139	7	8.0	Shahbaz Ahmed	
73	10	114	2	2.0	Mitchell Starc	

	most_runs	most_wkts
0	Anuj Rawat	Mustafizur Rahman
1	Sam Curran	Kuldeep Yadav
2	Andre Russell	T Natarajan
3	Sanju Samson	Trent Boult
4	Dewald Brevis	Jasprit Bumrah
..	…	…
69	NaN	NaN
70	Shreyas Iyer	Mitchell Starc
71	Yashasvi Jaiswal	Avesh Khan
72	Dhruv Jurel	Shahbaz Ahmed
73	Venkatesh Iyer	Andre Russell

[74 rows x 14 columns]

[ ]:
```
7: find the correlation matrix
```

[25]:
```
selected_columns = df[['first_score',␣
 ↪'first_wkts','second_score','second_wkts']]
print("Selected columns:\n", selected_columns)
```

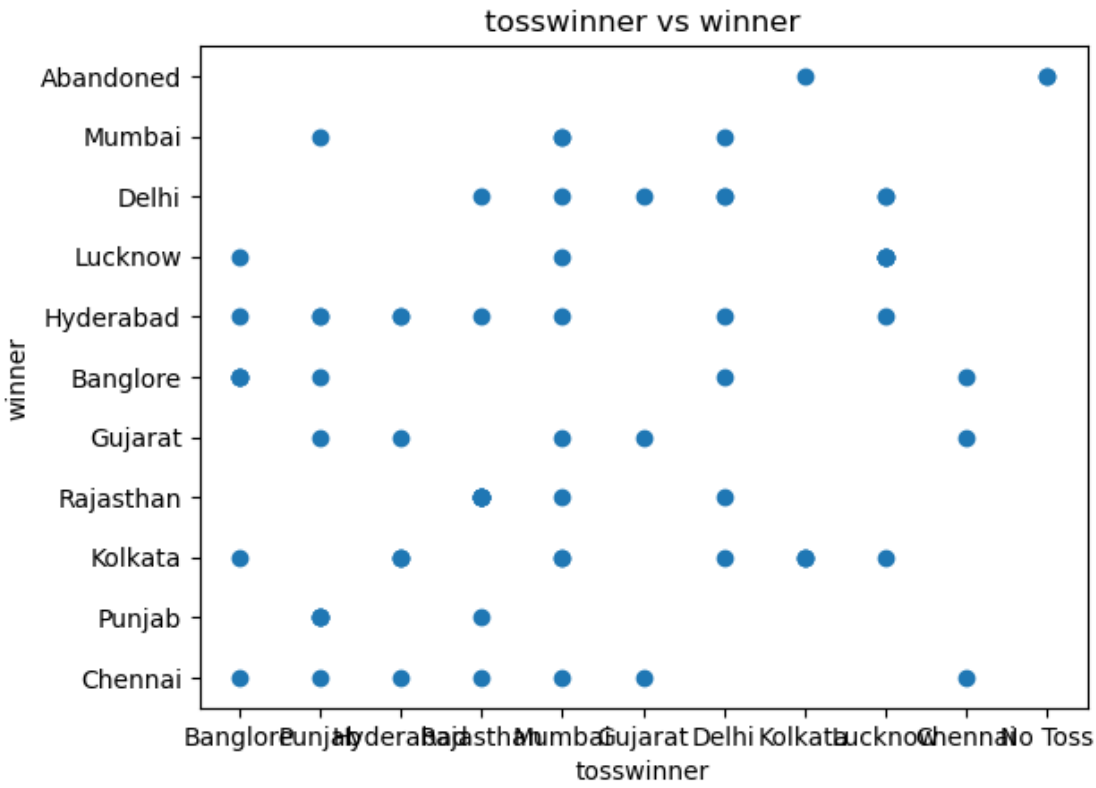
```
Selected columns:
     first_score  first_wkts  second_score  second_wkts
0            173           6           176            4
1            174           9           177            6
2            208           7           204            7
3            193           4           173            6
4            168           6           162            9
..           …           …           …           …
69             0           0             0            0
70           159          10           164            2
71           172           8           174            6
72           175           9           139            7
73           113          10           114            2

[74 rows x 4 columns]
```

[27]:
```
correlation_matrix = selected_columns.corr()
print("Correlation matrix:\n", correlation_matrix)
```

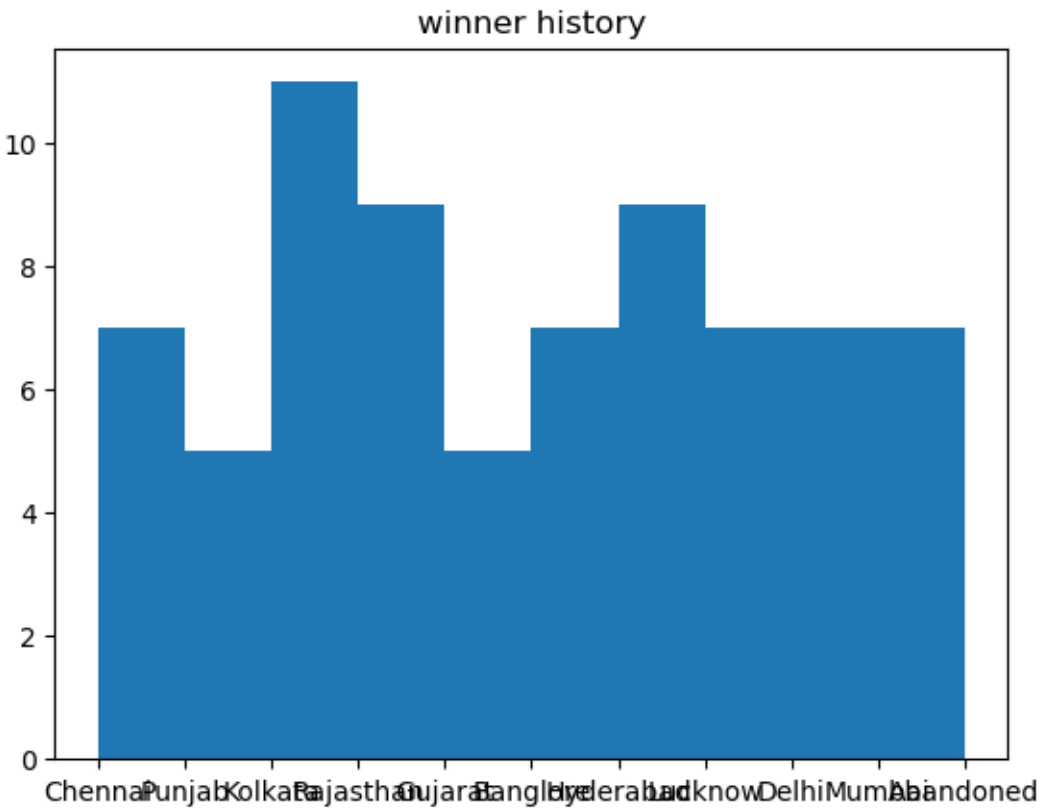
```
Correlation matrix:
              first_score  first_wkts  second_score  second_wkts
first_score      1.000000    0.064375      0.863921     0.549878
first_wkts       0.064375    1.000000      0.060219     0.116009
second_score     0.863921    0.060219      1.000000     0.291352
second_wkts      0.549878    0.116009      0.291352     1.000000
```

[ ]:
```
8: Do data visualization using matplot library ?
```

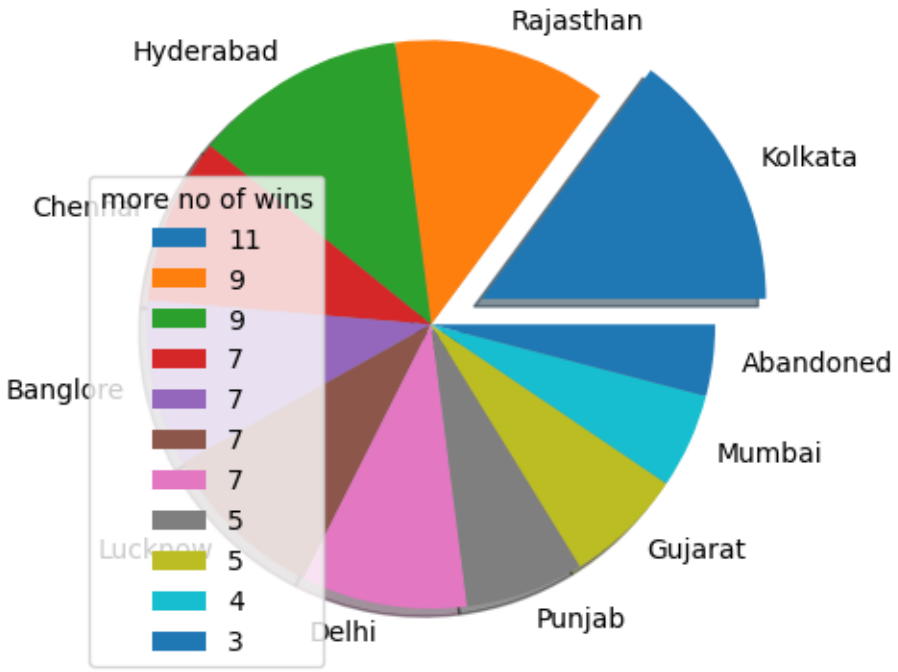
[15]:
```
plt.scatter(df['toss_winner'],df['winner'])
plt.title('tosswinner vs winner')
plt.xlabel('tosswinner')
plt.ylabel('winner')
plt.show()
```



[17]:
```
plt.hist(df['winner'])
plt.title('winner history')
plt.show()
```



[19]:
```
ser=df['winner'].value_counts()
myexplode = [0.2, 0, 0, 0,0,0,0,0,0,0,0]
mylabels =␣
 ↪['Kolkata','Rajasthan','Hyderabad','Chennai','Banglore','Lucknow','Delhi','Punjab','Gujarat
plt.pie(ser,labels = mylabels,explode = myexplode,shadow = True)
plt.legend(ser,title='more no of wins',loc='lower left')
plt.show()
ser
```



[19]:
```
winner
Kolkata      11
Rajasthan     9
Hyderabad     9
Chennai       7
Banglore      7
Lucknow       7
Delhi         7
Punjab        5
Gujarat       5
Mumbai        4
Abandoned     3
Name: count, dtype: int64
```

[ ]: