SCHOOL OF COMPUTER SCIENCE

MACHINE LEARNING

BACHELOR OF COMPUTER SCIENCE ENGINEERING WITH SPECIALIZATION IN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

BATCH: B3 (Hons.)

SEMESTER: IV

SUBMITTED TO: PROF. GOPAL S PHARTIYAL

SUBMITTED BY: HARSHIT RAHEJA and HARSH GOYAL

SAP ID: 500086226, 500086197

ROLL NO: R2142201556, R2142201617

ASSIGNMENT 2: PROJECT ANALYSIS OF IPL MATCHES

Procedure:

Step1: Download the zip File.

Step 2: Change the path of the csv files to the path you have stored the csv files.

Step 3: Analysis is done on the basis of various graphs

Code:

```
#%%
# importing all the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
#%%
# calling the dataset
warnings.filterwarnings('ignore')
data = pd.read_csv("D:\ML project\matches.csv") #change the path
data.head(3)


#%%
#calling the dataset
Data = pd.read_csv("D:\ML project\deliveries.csv")  #change the path
Data.head(3)
#%%
season_data=data[['id','season','winner']]
complete_data=Data.merge(season_data,how='inner',left_on='match_id',right_on='id')
data.columns.values
```

```
#%%
data = data.drop(columns=["umpire3"],axis=1)
data.head(3)
#%%
winner_per_season = data.groupby("season")["winner"].value_counts()
winner_per_season
#%%
#number of matches played per IPL season
plt.figure(figsize = (20,15))
sns.countplot('season',data=data,palette="tab10")
plt.title("Number of Matches played per IPL Season",fontsize=30)
plt.xlabel("Season",fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.ylabel('Matches',fontsize=20)
plt.show()
#%%
#Match wins by team
plt.figure(figsize = (20,15))
sns.countplot(x='winner',data=data, palette='crest')
plt.title("Match wins by team ",fontsize=30)
plt.xticks(fontsize=15,rotation=90)
plt.yticks(fontsize=15)
plt.xlabel("Teams",fontsize=20)
plt.ylabel("No of wins",fontsize=20)
plt.show()
#%%
data['win_by']=np.where(data['win_by_runs']>0,'Bat first','Bowl first')
#%%
```

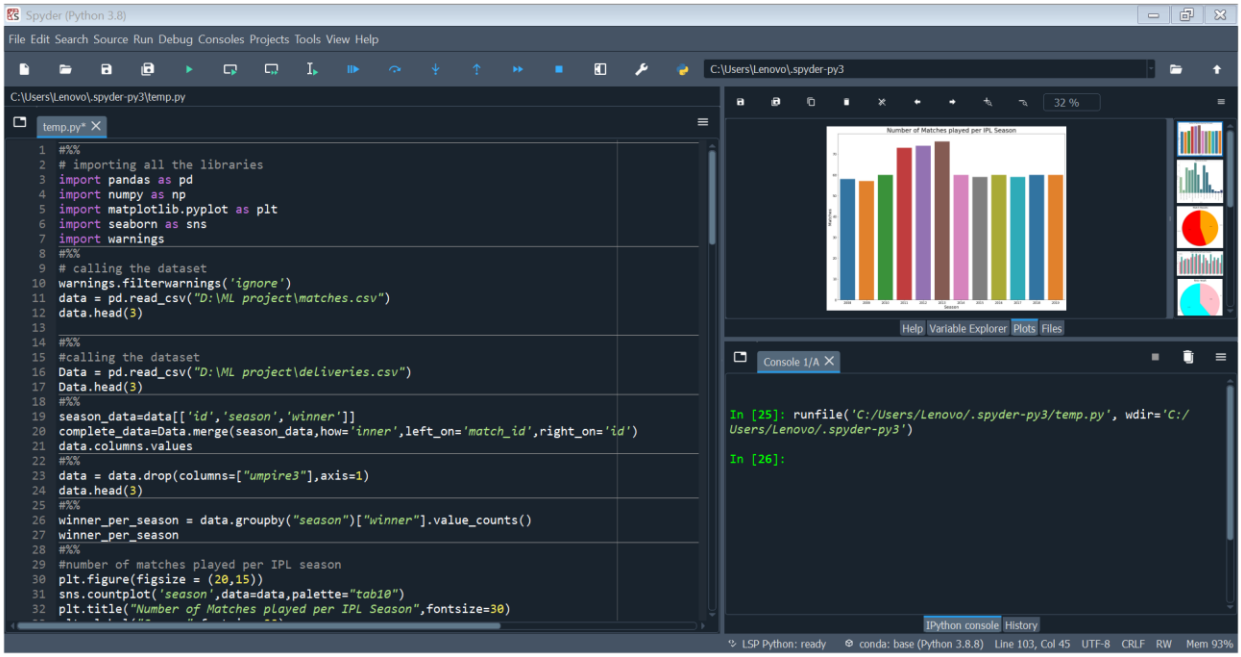
```
#match results who bowl first and win the match
Win=data.win_by.value_counts()
labels=np.array(Win.index)
sizes = Win.values
colors = ['red', 'orange']
plt.figure(figsize = (10,10))
plt.pie(sizes, labels=labels, colors=colors,autopct='%1.1f%%', shadow=True,startangle=90)
plt.title('Match Results',fontsize=30)
plt.axis('equal')
plt.show()
#%%
#Match wins by batting and bowling
plt.figure(figsize = (20,10))
sns.countplot('season',hue='win_by',data=data,palette='husl')
plt.title("Match wins by batting and bowling ",fontsize=30)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.xlabel("Season",fontsize=20)
plt.ylabel("Count",fontsize=20)
plt.show()
#%%
#Toss result
Toss=data.toss_decision.value_counts()
labels=np.array(Toss.index)
sizes = Toss.values
colors = ['cyan', 'pink']
plt.figure(figsize = (10,10))
plt.pie(sizes, labels=labels, colors=colors,autopct='%1.1f%%', shadow=True,startangle=90)
plt.title('Toss result',fontsize=30)
```

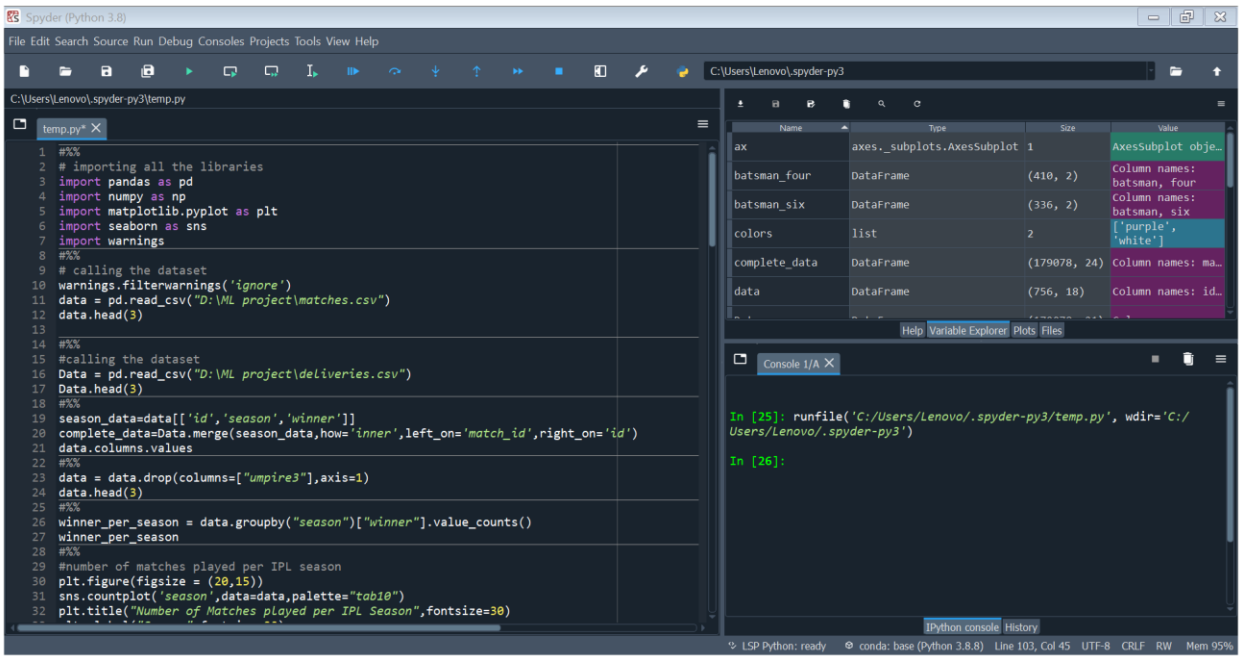
```
plt.axis('equal')
plt.show()
#%%
# match win by toss result
plt.figure(figsize = (20,10))
sns.countplot('season',hue='toss_decision',data=data,palette='rocket')
plt.title("Match win by Toss result ",fontsize=30)
plt.xlabel("Season",fontsize=20)
plt.ylabel("Count",fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
#%%
#final matches
final_matches=data.drop_duplicates(subset=['season'], keep='last')
final_matches[['season','winner']].reset_index(drop=True).sort_values('season')
#%%
#match result
match = final_matches.win_by.value_counts()
labels=np.array(Toss.index)
sizes = match.values
colors = ['purple', 'white']
plt.figure(figsize = (10,10))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', shadow=True,startangle=90)
plt.title('Match Result',fontsize=30)
plt.axis('equal')
plt.show()
#%%
#top player of the match winners
```

```
top_players = data.player_of_match.value_counts()[:15]
fig, ax = plt.subplots()
ax.set_ylim([0,25])
ax.set_xlim([0,20])
ax.set_ylabel("Count",fontsize=20)
ax.set_xlabel("Player_Name",fontsize=20)
ax.set_title("Top player of the match winners",fontsize=30)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
top_players.plot.bar(figsize = (20,15))
sns.barplot(x = top_players.index, y = top_players, orient='v', palette="Spectral");
plt.show()
#%%
# fours hits by players
four_data=complete_data[complete_data['batsman_runs']==4]
four_data.groupby('batting_team')['batsman_runs'].agg([('runs by fours','sum'),('fours','count')])
batsman_four=four_data.groupby('batsman')['batsman_runs'].agg([('four','count')]).reset_index().
sort_values('four',ascending=0)
ax=batsman_four.iloc[:15,:].plot('batsman','four',kind='bar',color='gold',figsize = (20,15))
ax.set_title("Fours hit by playes ",fontsize=30)
plt.xticks(rotation=90)
plt.xlabel("Player name",fontsize=20)
plt.ylabel("No of fours",fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
#%%
#six hits by players
six_data=complete_data[complete_data['batsman_runs']==6]
```

```
six_data.groupby('batting_team')['batsman_runs'].agg([('runs by six','sum'),('sixes','count')])

batsman_six=six_data.groupby('batsman')['batsman_runs'].agg([('six','count')]).reset_index().sort
_values('six',ascending=0)

ax=batsman_six.iloc[:15,:].plot('batsman','six',kind='bar',color='lime',figsize = (20,15))

plt.title("Six hit by playes ",fontsize=30)

plt.xticks(fontsize=15)

plt.yticks(fontsize=15)

plt.xlabel("Player name",fontsize=20)

plt.ylabel("No of six",fontsize=20)

plt.show()

#%%

#Dismissals

plt.figure(figsize=(20,15))

ax=sns.countplot(Data.dismissal_kind,palette="terrain")

plt.title("Dismissals",fontsize=30)

plt.xlabel("Dismissals type",fontsize=20)

plt.ylabel("count",fontsize=20)

plt.xticks(fontsize=15)

plt.yticks(fontsize=15)

plt.show()
```

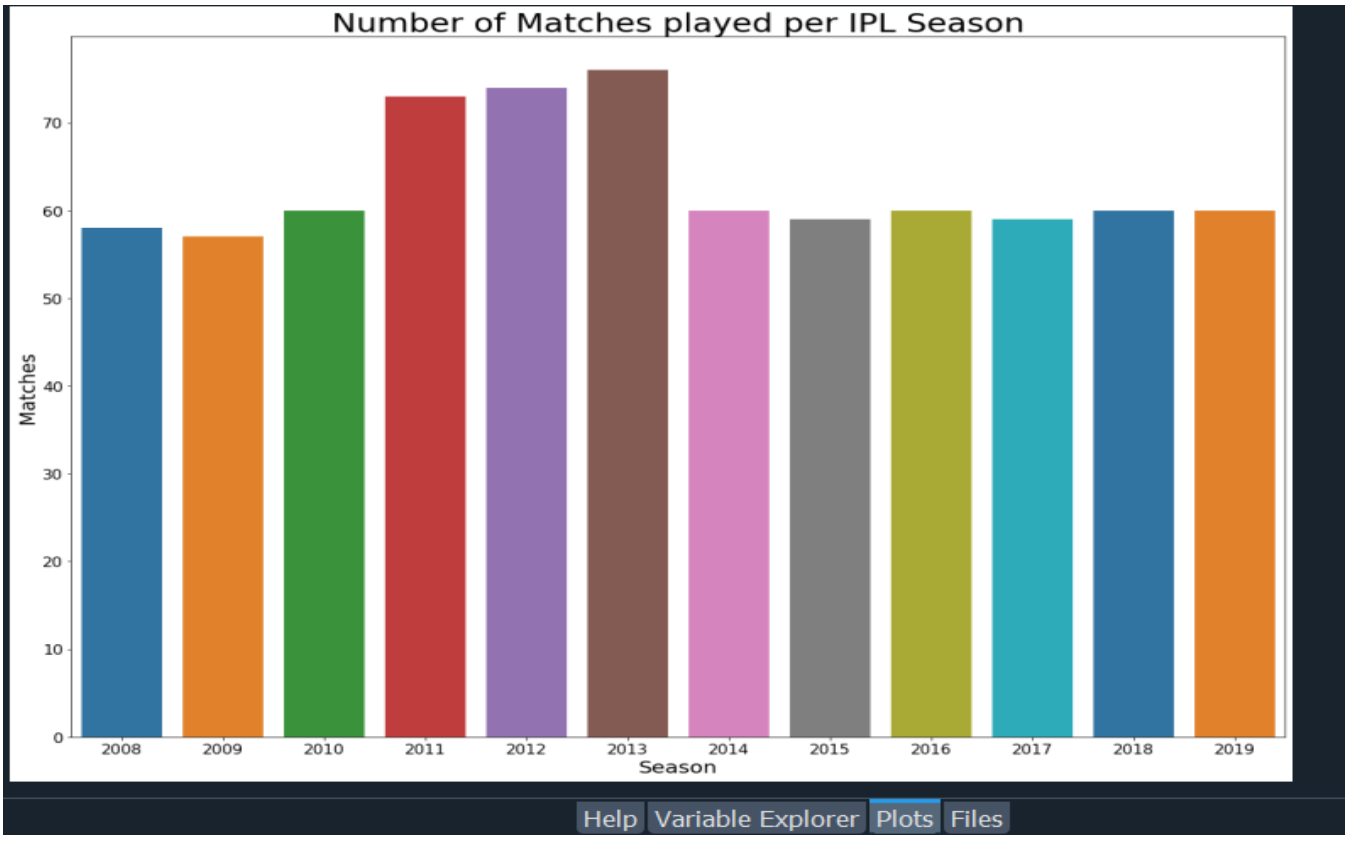
Output:


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Lenovo\.spyder-py3
C:\Users\Lenovo\.spyder-py3\temp.py
temp.py*
#%%
# importing all the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
#%%
# calling the dataset
warnings.filterwarnings('ignore')
data = pd.read_csv("D:\ML project\matches.csv")
data.head(3)
#%%
#calling the dataset
Data = pd.read_csv("D:\ML project\deliveries.csv")
Data.head(3)
#%%
season_data=data[['id','season','winner']]
complete_data=Data.merge(season_data,how='inner',left_on='match_id',right_on='id')
data.columns.values
#%%
data = data.drop(columns=["umpire3"],axis=1)
data.head(3)
#%%
winner_per_season = data.groupby("season")["winner"].value_counts()
winner_per_season
#%%
#number of matches played per IPL season
plt.figure(figsize = (20,15))
sns.countplot('season',data=data,palette="tab10")
plt.title("Number of Matches played per IPL Season",fontsize=30)
Number of Matches played per IPL Season
Help Variable Explorer Plots Files
Console 1/A
In [25]: runfile('C:/Users/Lenovo/.spyder-py3/temp.py', wdir='C:/Users/Lenovo/.spyder-py3')
In [26]:
IPython console History
LSP Python: ready
conda: base (Python 3.8.8)
Line 103, Col 45
UTF-8
CRLF
RW
Mem 93%

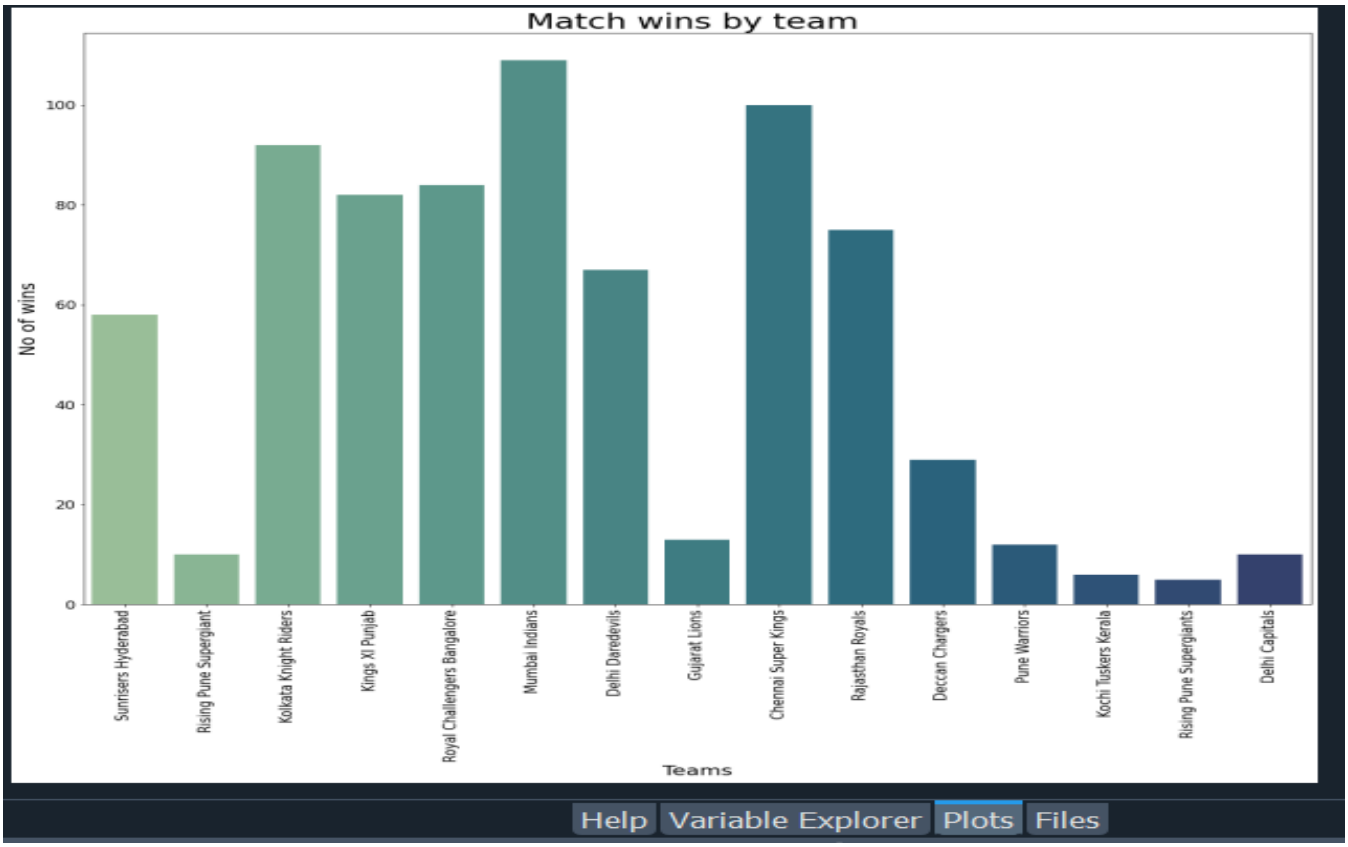

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Lenovo\.spyder-py3
C:\Users\Lenovo\.spyder-py3\temp.py
temp.py*
Name
Type
Size
Value
ax
axes._subplots.AxesSubplot
1
AxesSubplot obje…
batsman_four
DataFrame
(410, 2)
Column names: batsman, four
batsman_six
DataFrame
(336, 2)
Column names: batsman, six
colors
list
2
['purple', 'white']
complete_data
DataFrame
(179078, 24)
Column names: ma…
data
DataFrame
(756, 18)
Column names: id…
Help Variable Explorer Plots Files
Console 1/A
In [25]: runfile('C:/Users/Lenovo/.spyder-py3/temp.py', wdir='C:/Users/Lenovo/.spyder-py3')
In [26]:
IPython console History
LSP Python: ready
conda: base (Python 3.8.8)
Line 103, Col 45
UTF-8
CRLF
RW
Mem 95%

Output of Graphs and codes:

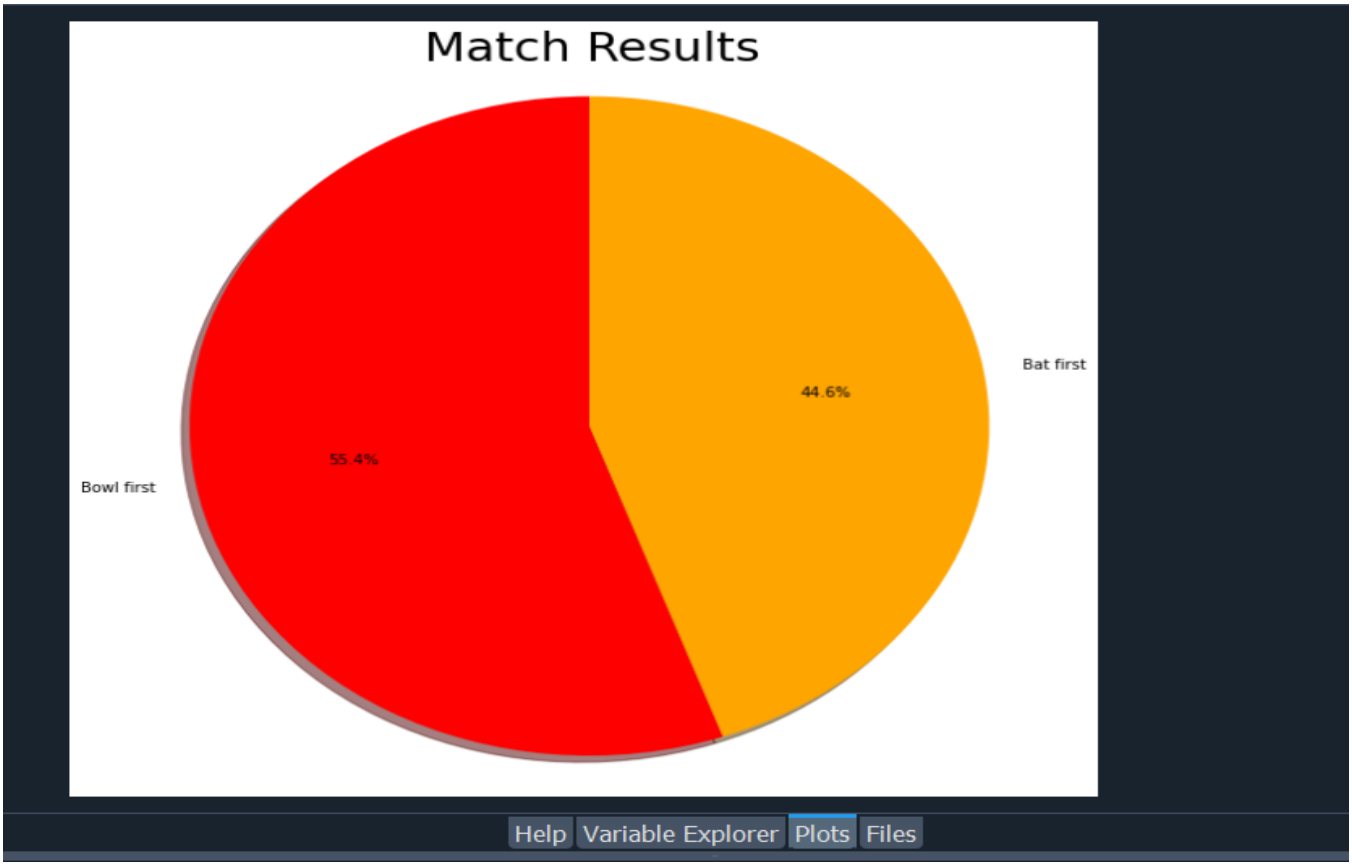
```
28  #%%
29  #number of matches played per IPL season
30  plt.figure(figsize = (20,15))
31  sns.countplot('season',data=data,palette="tab10")
32  plt.title("Number of Matches played per IPL Season",fontsize=30)
33  plt.xlabel("Season",fontsize=20)
34  plt.xticks(fontsize=15)
35  plt.yticks(fontsize=15)
36  plt.ylabel('Matches',fontsize=20)
37  plt.show()
```



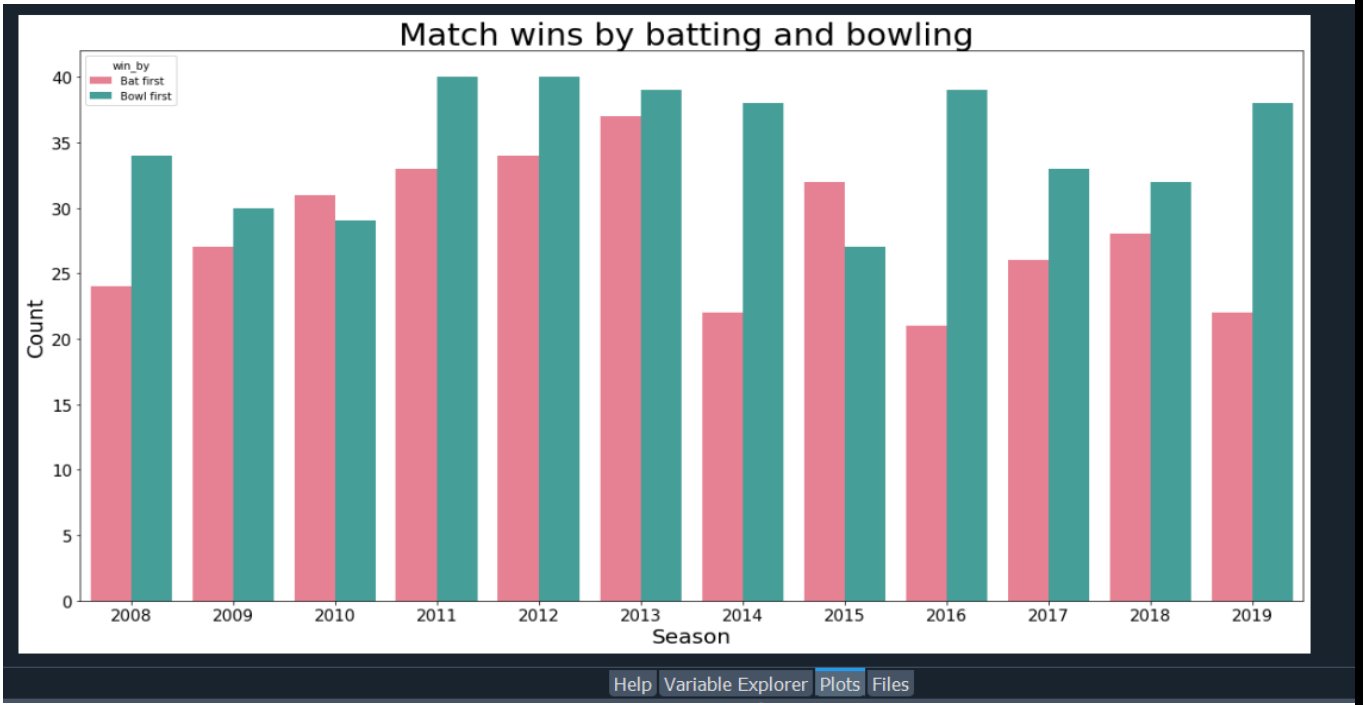
```
#%%
#Match wins by team
plt.figure(figsize = (20,15))
sns.countplot(x='winner',data=data, palette='crest')
plt.title("Match wins by team ",fontsize=30)
plt.xticks(fontsize=15,rotation=90)
plt.yticks(fontsize=15)
plt.xlabel("Teams",fontsize=20)
plt.ylabel("No of wins",fontsize=20)
plt.show()
```



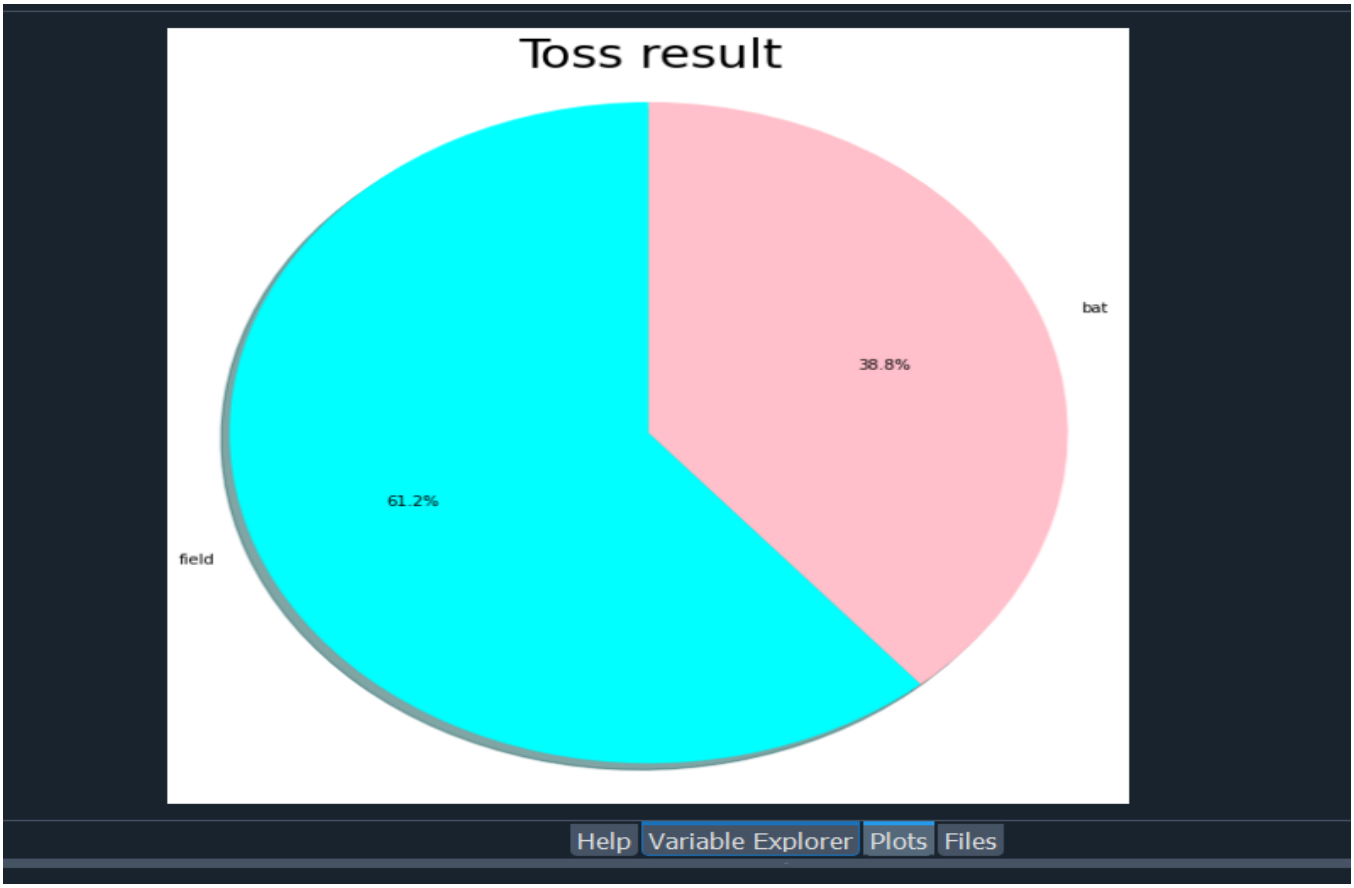
```
#%%
#match results who bowl first and win the match
Win=data.win_by.value_counts()
labels=np.array(Win.index)
sizes = Win.values
colors = ['red', 'orange']
plt.figure(figsize = (10,10))
plt.pie(sizes, labels=labels, colors=colors,autopct='%1.1f%%', shadow=True,startangle=90)
plt.title('Match Results',fontsize=30)
plt.axis('equal')
plt.show()
```



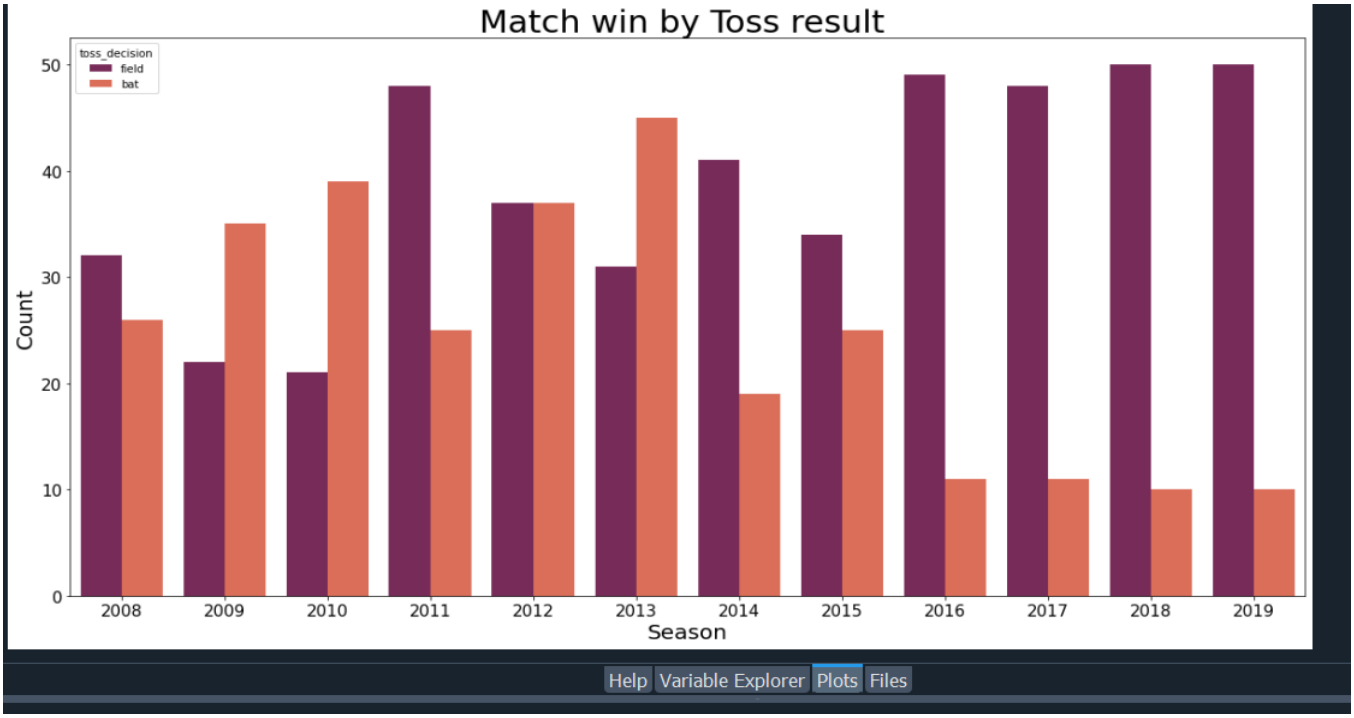
```
61  #%%
62  #Match wins by batting and bowling
63  plt.figure(figsize = (20,10))
64  sns.countplot('season',hue='win_by',data=data,palette='husl')
65  plt.title("Match wins by batting and bowling ",fontsize=30)
66  plt.xticks(fontsize=15)
67  plt.yticks(fontsize=15)
68  plt.xlabel("Season",fontsize=20)
69  plt.ylabel("Count",fontsize=20)
70  plt.show()
```



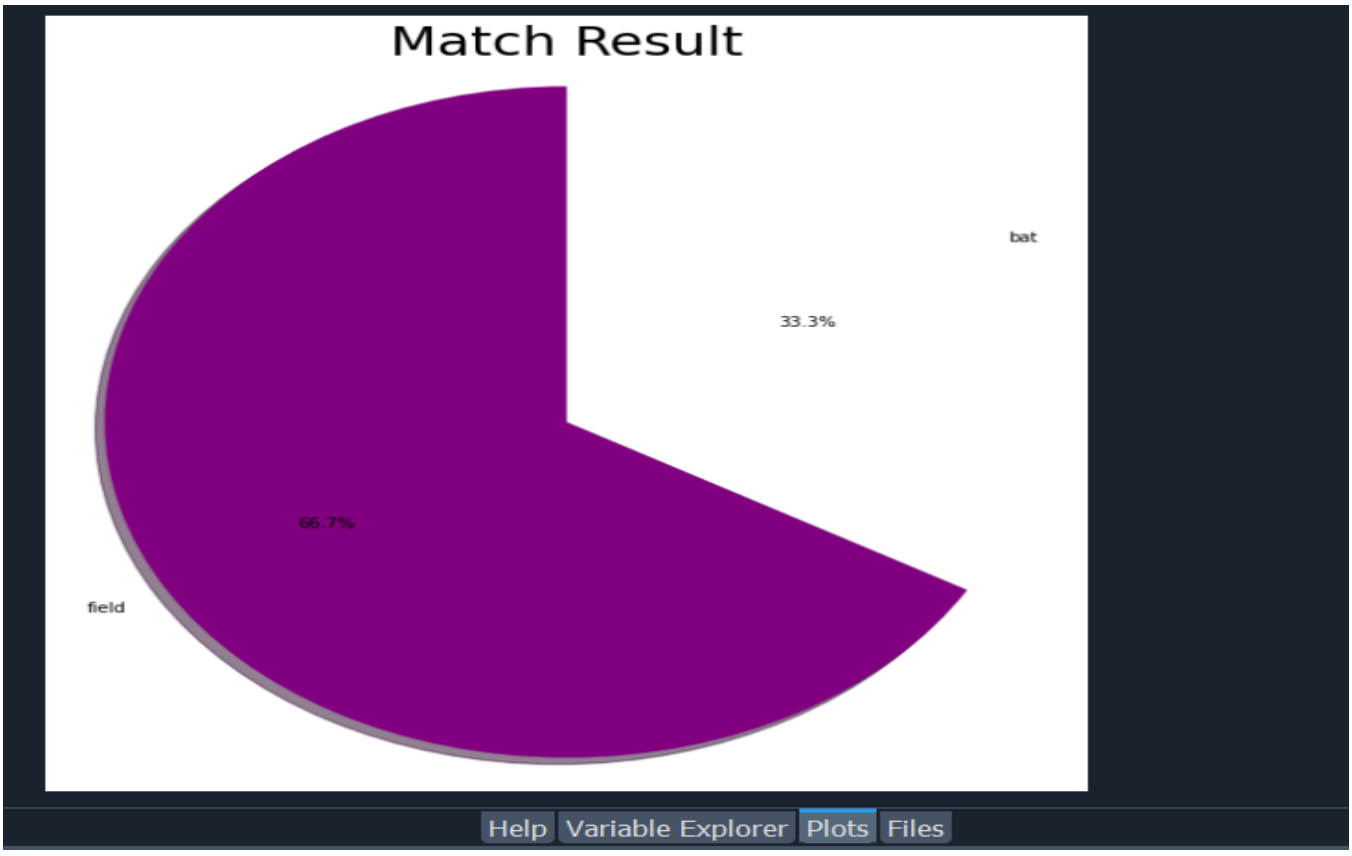
```
#%%
#Toss result
Toss=data.toss_decision.value_counts()
labels=np.array(Toss.index)
sizes = Toss.values
colors = ['cyan', 'pink']
plt.figure(figsize = (10,10))
plt.pie(sizes, labels=labels, colors=colors,autopct='%1.1f%%', shadow=True,startangle=90)
plt.title('Toss result',fontsize=30)
plt.axis('equal')
plt.show()
```



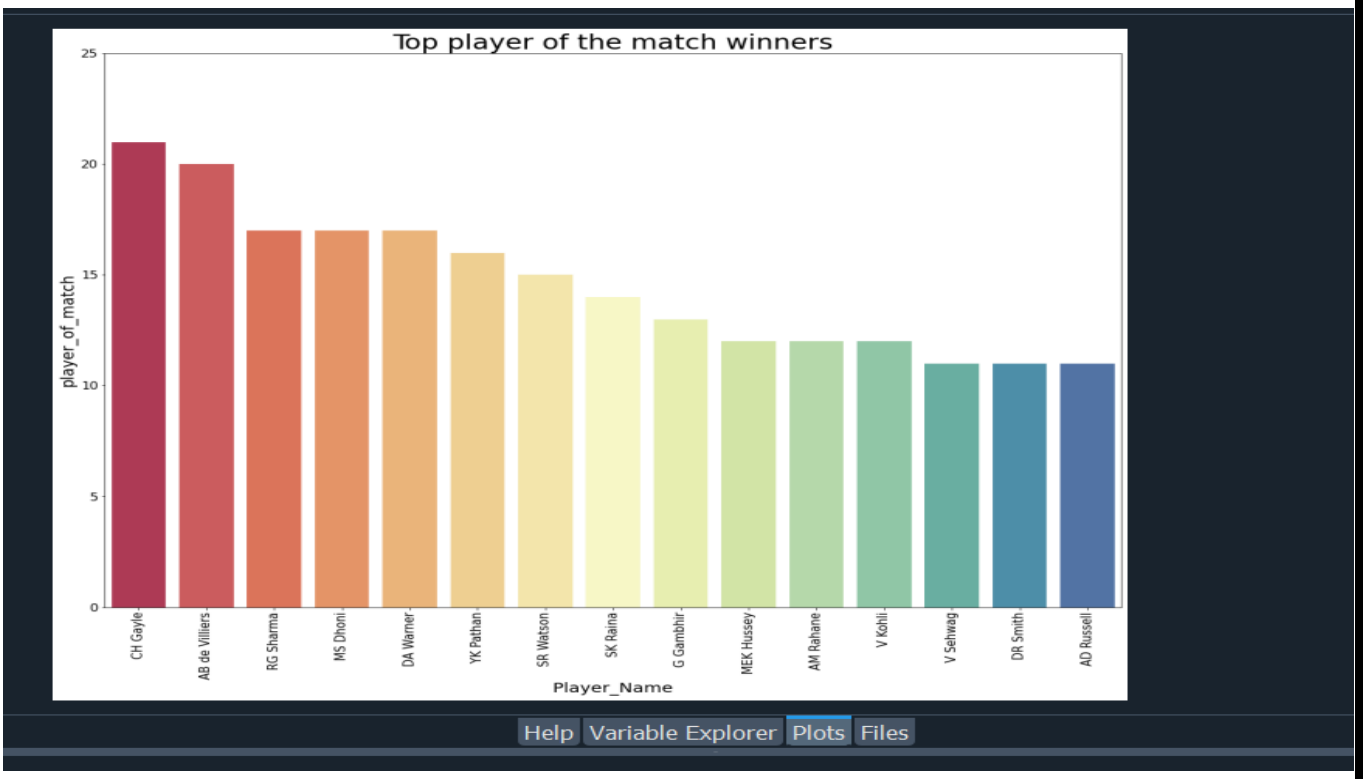
```
#%%
# match win by toss result
plt.figure(figsize = (20,10))
sns.countplot('season',hue='toss_decision',data=data,palette='rocket')
plt.title("Match win by Toss result ",fontsize=30)
plt.xlabel("Season",fontsize=20)
plt.ylabel("Count",fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```



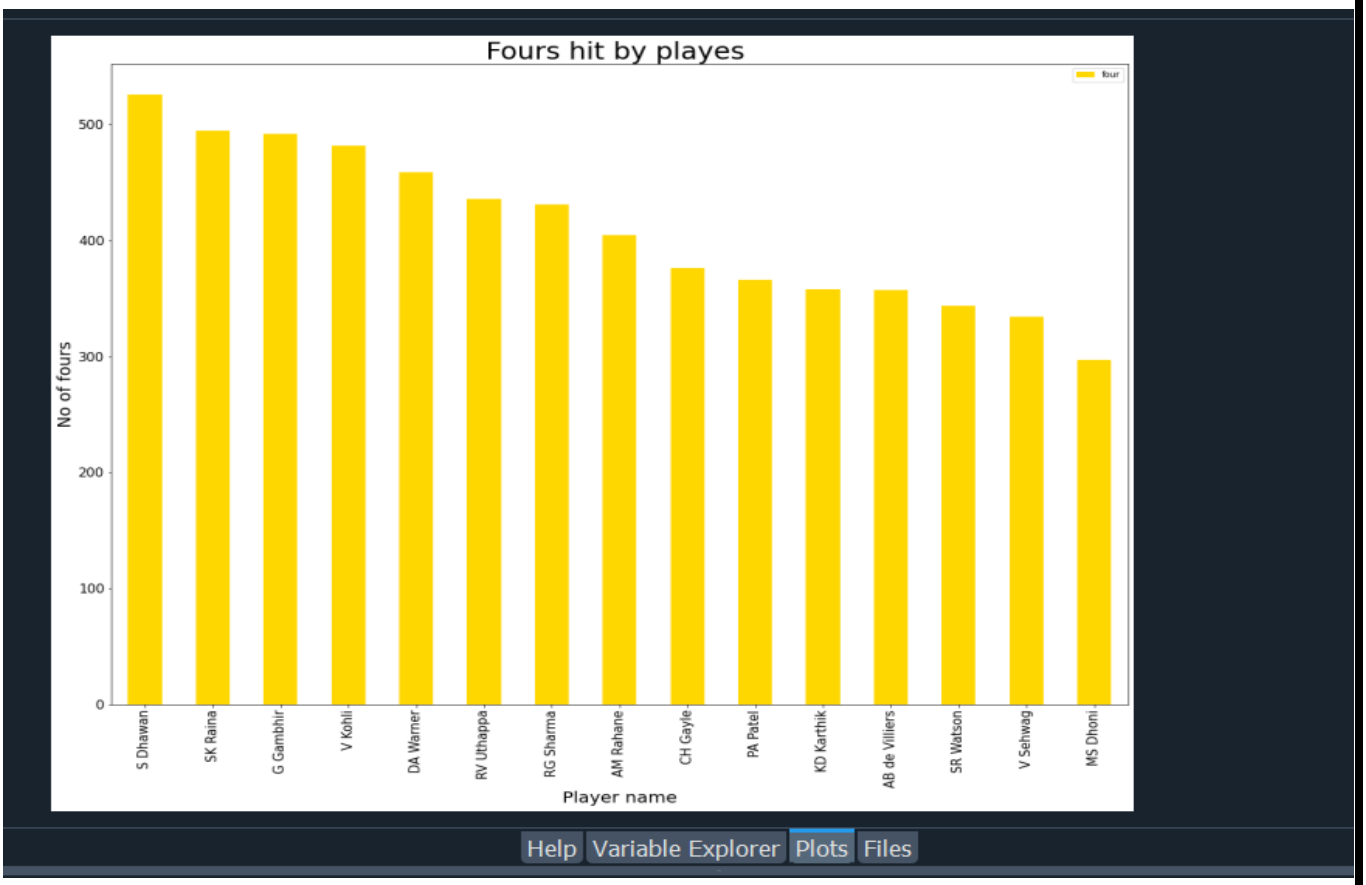
```
#%%
#final matches
final_matches=data.drop_duplicates(subset=['season'], keep='last')
final_matches[['season','winner']].reset_index(drop=True).sort_values('season')
#%%
#match result
match = final_matches.win_by.value_counts()
labels=np.array(Toss.index)
sizes = match.values
colors = ['purple', 'white']
plt.figure(figsize = (10,10))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', shadow=True,startangle=90)
plt.title('Match Result',fontsize=30)
plt.axis('equal')
plt.show()
```



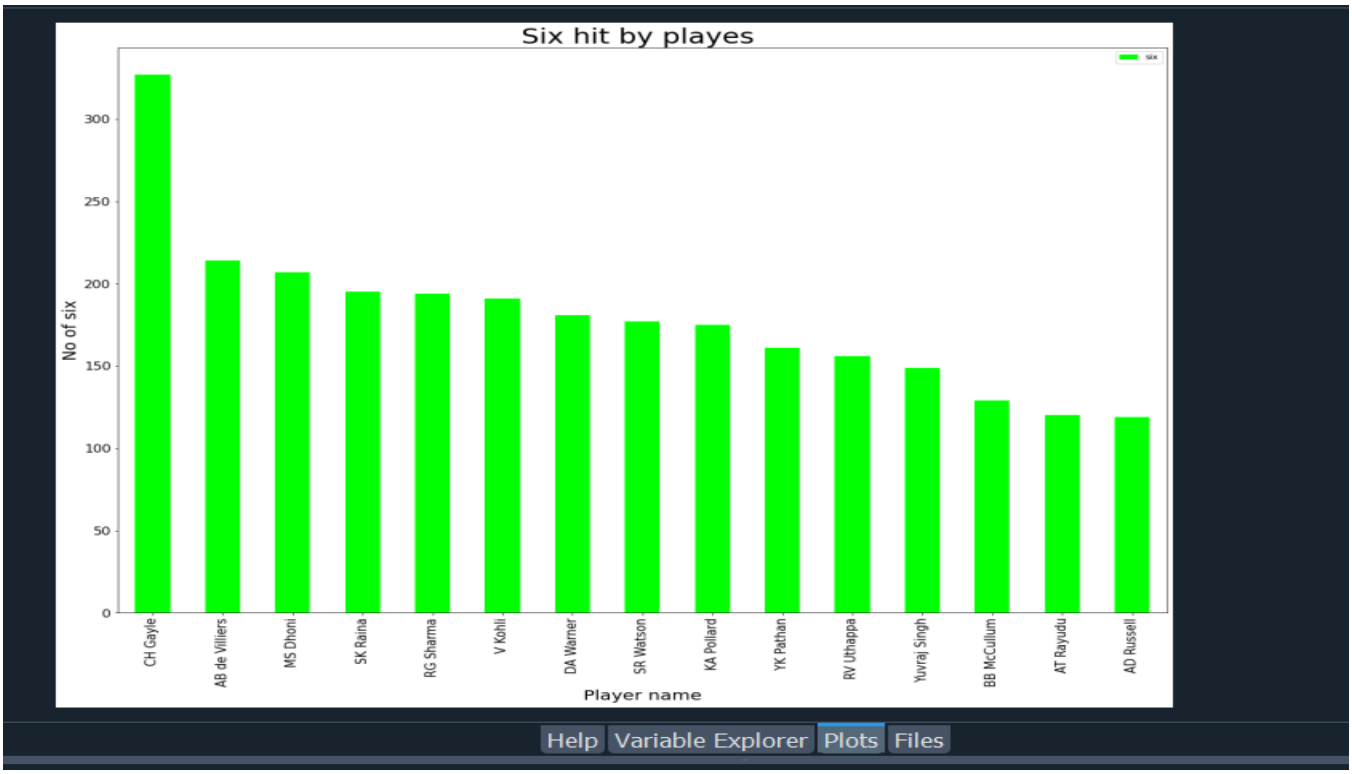
```
#%%
#top player of the match winners
top_players = data.player_of_match.value_counts()[:15]
fig, ax = plt.subplots()
ax.set_ylim([0,25])
ax.set_xlim([0,20])
ax.set_ylabel("Count",fontsize=20)
ax.set_xlabel("Player_Name",fontsize=20)
ax.set_title("Top player of the match winners",fontsize=30)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
top_players.plot.bar(figsize = (20,15))
sns.barplot(x = top_players.index, y = top_players, orient='v', palette="Spectral");
plt.show()
```



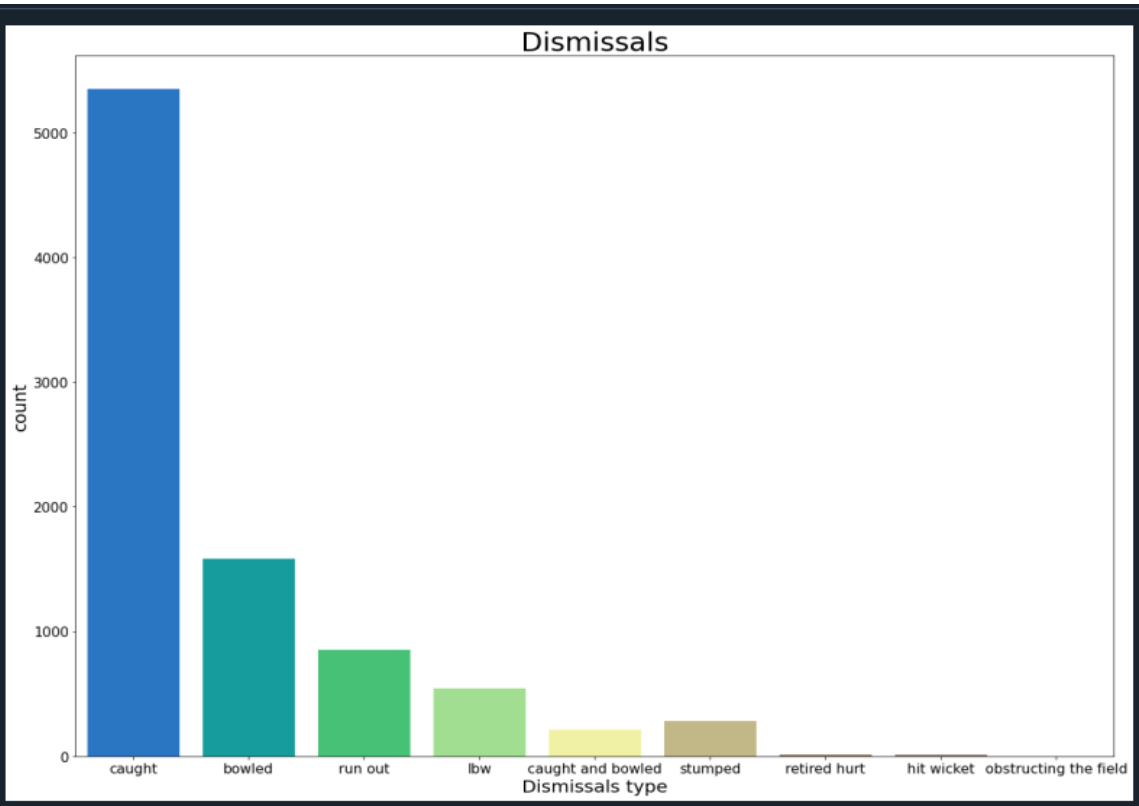
```
#%%
# fours hits by players
four_data=complete_data[complete_data['batsman_runs']==4]
four_data.groupby('batting_team')['batsman_runs'].agg([('runs by fours','sum'),('fours','count')])
batsman_four=four_data.groupby('batsman')['batsman_runs'].agg([('four','count')]).reset_index().sort_values('four',ascending=0)
ax=batsman_four.iloc[:15,:].plot('batsman','four',kind='bar',color='gold',figsize = (20,15))
ax.set_title("Fours hit by playes ",fontsize=30)
plt.xticks(rotation=90)
plt.xlabel("Player name",fontsize=20)
plt.ylabel("No of fours",fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```



```
#%%
#six hits by players
six_data=complete_data[complete_data['batsman_runs']==6]
six_data.groupby('batting_team')['batsman_runs'].agg([('runs by six','sum'),('sixes','count')])
batsman_six=six_data.groupby('batsman')['batsman_runs'].agg([('six','count')]).reset_index().sort_values('six',ascending=0)
ax=batsman_six.iloc[:15,:].plot('batsman','six',kind='bar',color='lime',figsize = (20,15))
plt.title("Six hit by playes ",fontsize=30)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.xlabel("PLayer name",fontsize=20)
plt.ylabel("No of six",fontsize=20)
plt.show()
```



```
146 #%%
147 #Dismissals
148 plt.figure(figsize=(20,15))
149 ax=sns.countplot(Data.dismissal_kind,palette="terrain")
150 plt.title("Dismissals",fontsize=30)
151 plt.xlabel("Dismissals type",fontsize=20)
152 plt.ylabel("count",fontsize=20)
153 plt.xticks(fontsize=15)
154 plt.yticks(fontsize=15)
155 plt.show()
156
```



Help | Variable Explorer | Plots | Files