**Logo, company name

Description automatically generated**

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)

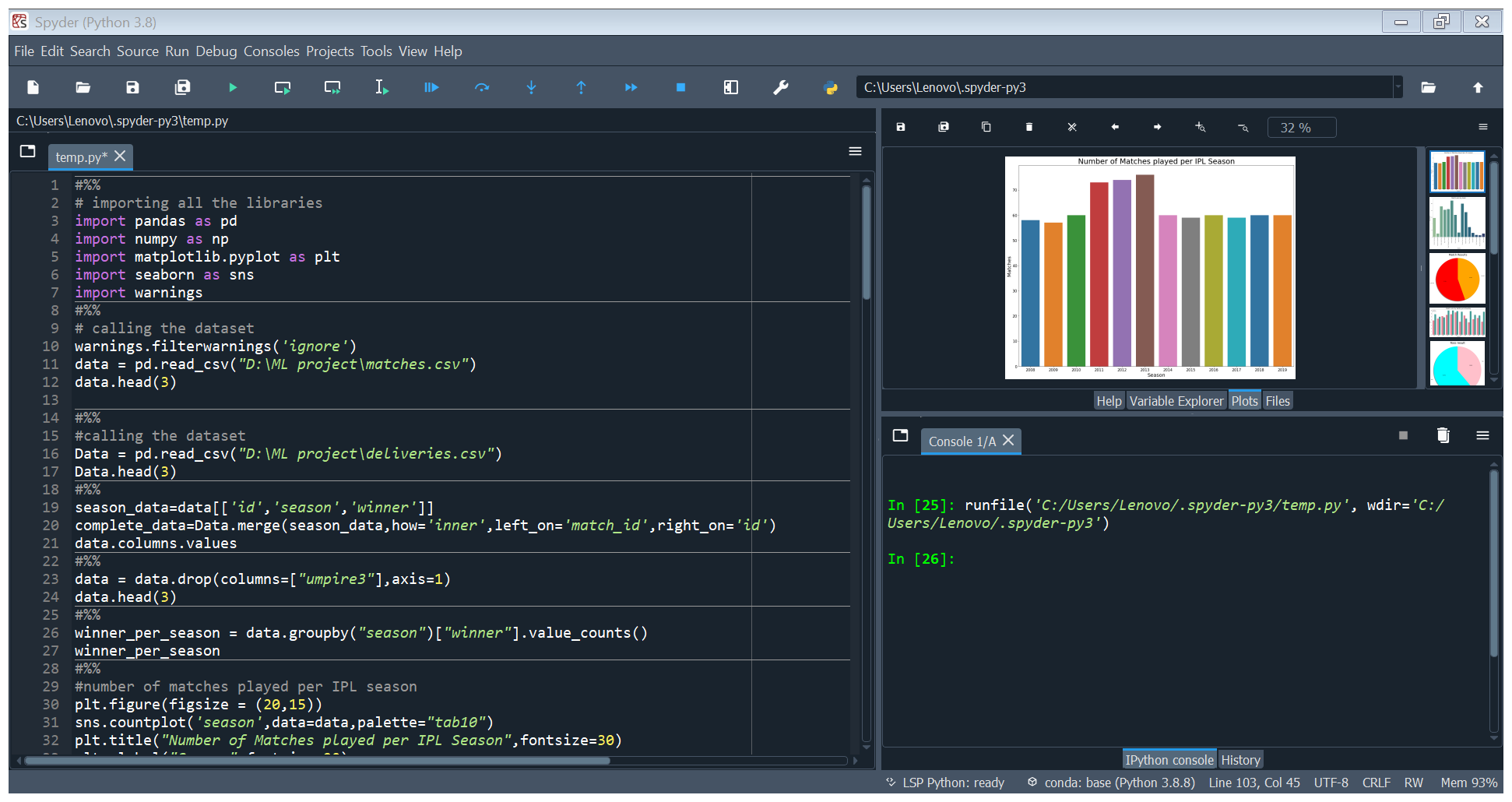
plt.ylabel("count",fontsize=20)

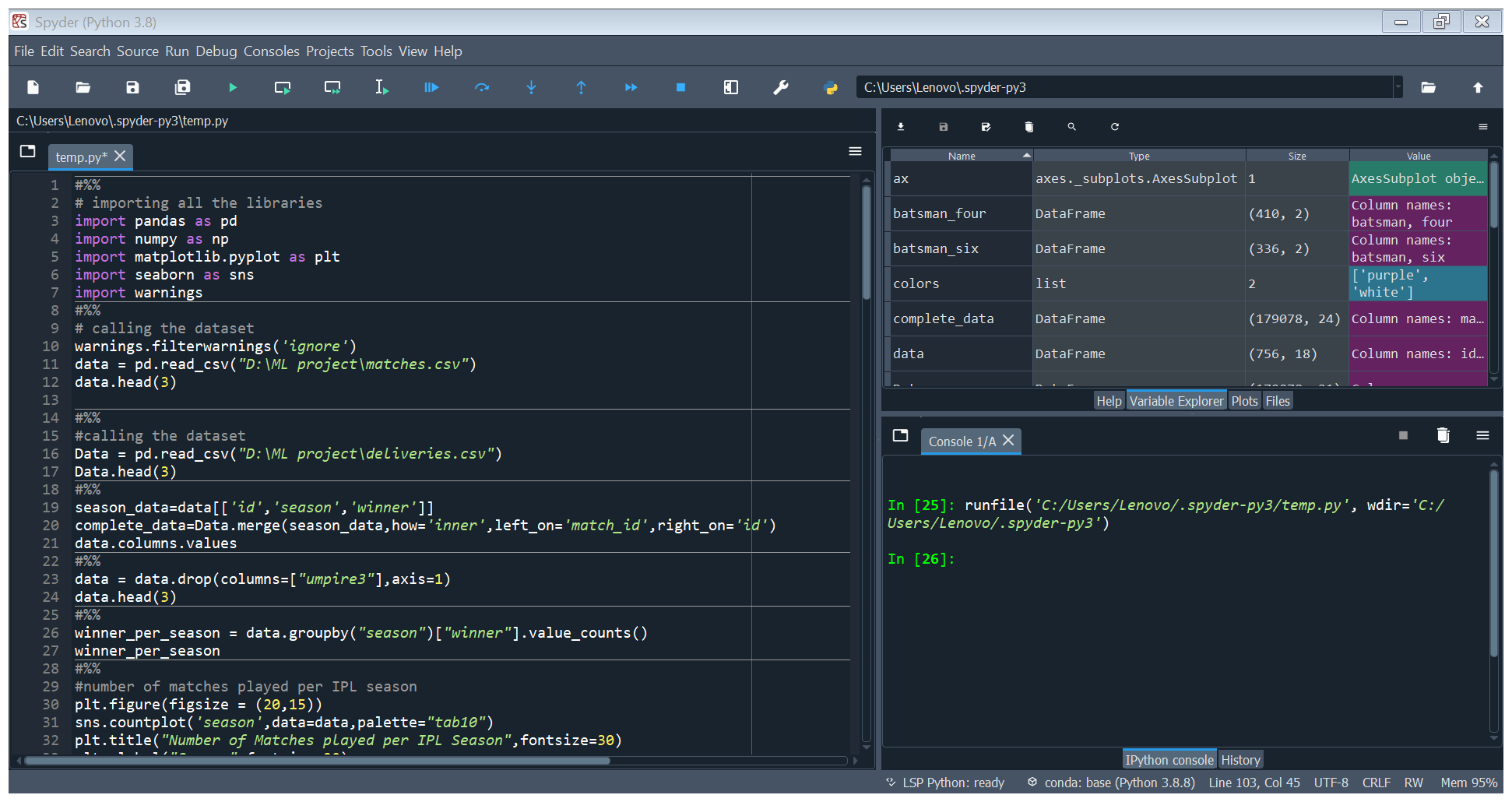
plt.xticks(fontsize=15)

plt.yticks(fontsize=15)

plt.show()

**Output:**





**Output of Graphs and codes:**

Text

Description automatically generated

Chart, bar chart

Description automatically generated

Text

Description automatically generated

Chart, bar chart, histogram

Description automatically generated

Text

Description automatically generated

Chart, pie chart

Description automatically generated

Text

Description automatically generated

Chart, bar chart

Description automatically generated

Text

Description automatically generated

Chart, pie chart

Description automatically generated

Text

Description automatically generated

Chart, bar chart

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Chart, pie chart

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Chart, bar chart

Description automatically generated

A computer screen capture

Description automatically generated with low confidence

Chart, bar chart

Description automatically generated

A computer screen capture

Description automatically generated with medium confidence

Chart, bar chart

Description automatically generated

Text

Description automatically generated

Chart, histogram

Description automatically generated