DATA SCIENCE

pcacs

UNIVERSITY OF MUMBAI

PROJECT ENTITLED

“DATA SCIENCE”

SUBMITTED BY

6510 Prasad Belote

6511 Viraj Bhagat

UNDER THE GUIDANCE OF

Mrs. Sanjana Bhangale Mam

PILLAI COLLEGE OF ARTS, COMMERCE &

SCIENCE, NEW PANVEL

2020-2021

Tel.: 2745 6100 / 1700
Fax: 2748 3208
website: www.pcac.ac.in
www.mes.ac.in

MAHATMA EDUCATION SOCIETY'S

Pillai's College of Arts, Commerce & Science

Dr. K. M. Vasudevan Pillai's Campus, 10, Sector 16, New Panvel 410206.

(NAAC ACCREDITED 'A' GRADE)

Certificate

This is to certify that the project entitled "**Data Science"** is successfully completed by **Mr. Prasad Belote and Mr.Viraj Bhagat** as per the syllabus and in partial fulfilment for the completion of BSc. degree in Computer Science of University of Mumbai, it is also to certify that this is the original work of the candidate done during the academic year 2020 – 2021.

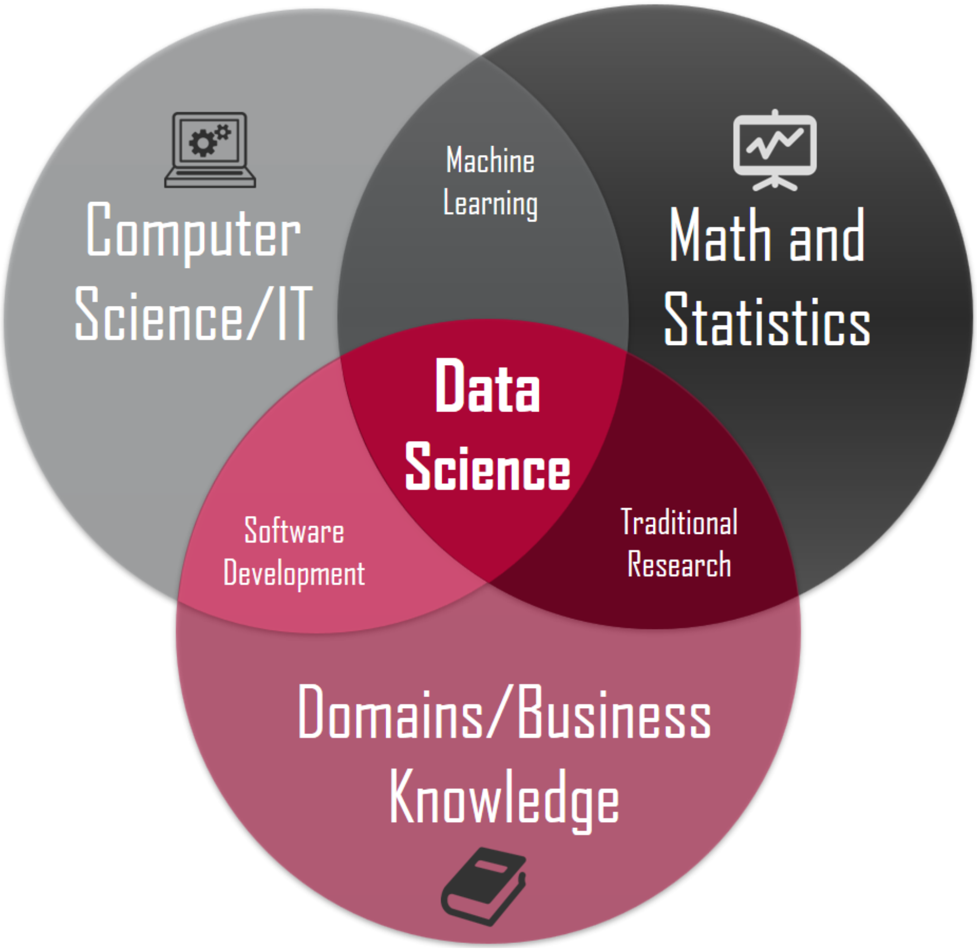
M. E. S.

pcacs

INTRODUCTION

Data science is a multi-disciplinary field that scientific uses methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data.

Data Science is the science which uses computer science, statistics and machine learning, visualization and human-computer interactions to **collect, clean, integrate, analyze, visualize, interact** With **data** to **create data products**.



EXPLORATORY DATA ANALYSIS (EDA)

Exploratory Data Analysis or (EDA) is understanding the data sets by summarizing their main characteristics often plotting them visually. This step is very important especially when we arrive at modeling the data in order to apply Machine learning. Plotting in EDA consists of Histograms, Box plot, Scatter plot and many more. It often takes much time to explore the data.

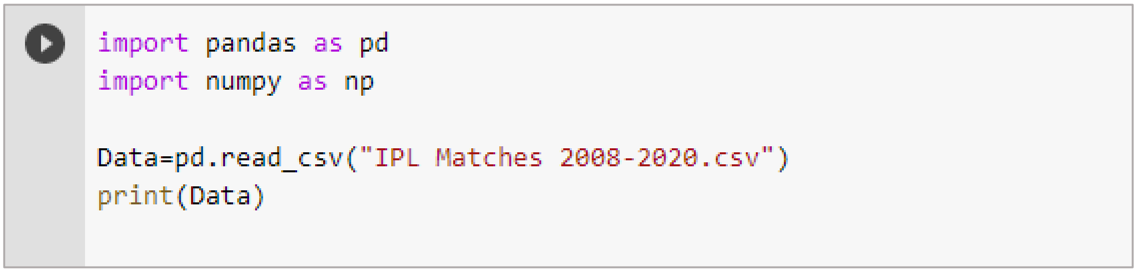
Since We are huge fan of IPL , We got a very beautiful data-set of IPL from Kaggle. To give a piece of brief information about the data set this data contains more of 500 rows and more than 10 columns which contains information about IPL from 2008 to 2020 .

We have total 17 columns from that we considered only 14 columns remaining 3 columns we exclude from our dataset. The 14 columns include such as match city , match date ,match venue, Neutral Venue means whether teams are playing on their home ground or both the teams are away from home, Man of the Match, team 1 ,team 2,toss decision ,toss winner then who was the winner of that match , match result means whether that team won by wickets or runs , result margin then eliminator and method .

So in this project , we will explore the data and make it ready for modeling.

1. Importing Dataset and required Libraries

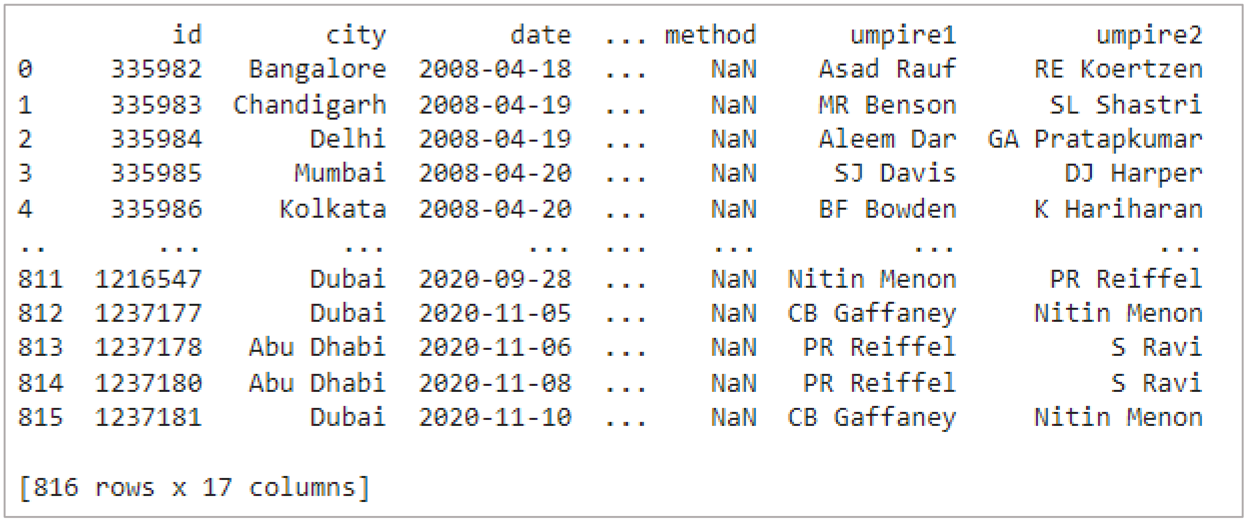
Code :



```
import pandas as pd
import numpy as np

Data=pd.read_csv("IPL Matches 2008-2020.csv")
print(Data)
```

Output:

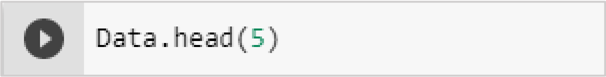


```
          id        city        date  ... method       umpire1         umpire2
0     335982   Bangalore  2008-04-18  ...    NaN     Asad Rauf     RE Koertzen
1     335983  Chandigarh  2008-04-19  ...    NaN     MR Benson      SL Shastri
2     335984       Delhi  2008-04-19  ...    NaN     Aleem Dar  GA Pratapkumar
3     335985      Mumbai  2008-04-20  ...    NaN      SJ Davis       DJ Harper
4     335986     Kolkata  2008-04-20  ...    NaN     BF Bowden     K Hariharan
..       ...         ...         ...  ...    ...           ...             ...
811  1216547       Dubai  2020-09-28  ...    NaN   Nitin Menon      PR Reiffel
812  1237177       Dubai  2020-11-05  ...    NaN   CB Gaffaney     Nitin Menon
813  1237178   Abu Dhabi  2020-11-06  ...    NaN    PR Reiffel          S Ravi
814  1237180   Abu Dhabi  2020-11-08  ...    NaN    PR Reiffel          S Ravi
815  1237181       Dubai  2020-11-10  ...    NaN   CB Gaffaney     Nitin Menon

[816 rows x 17 columns]
```

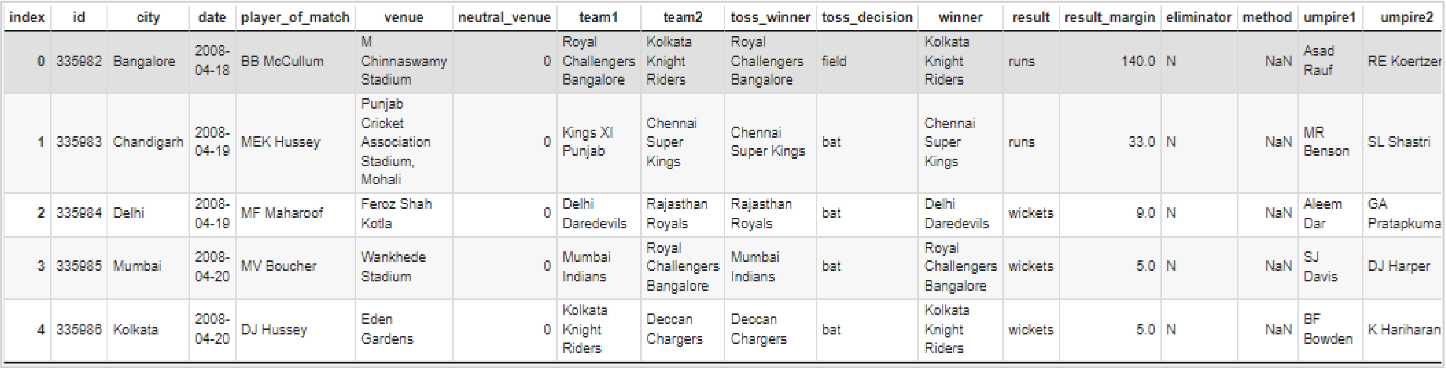
2.Loading Data of First 5 Rows

Code :



```
Data.head(5)
```

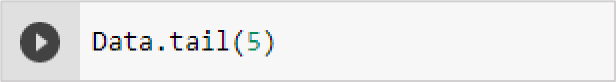
Output:



index	id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	result_margin	eliminator	method	umpire1	umpire2
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	runs	140.0	N	NaN	Asad Rauf	RE Koertzen
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	runs	33.0	N	NaN	MR Benson	SL Shastri
2	335984	Delhi	2008-04-19	MF Maharoof	Feroz Shah Kotla	0	Delhi Daredevils	Rajasthan Royals	Rajasthan Royals	bat	Delhi Daredevils	wickets	9.0	N	NaN	Aleem Dar	GA Pratapkuma
3	335985	Mumbai	2008-04-20	MV Boucher	Wankhede Stadium	0	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	Royal Challengers Bangalore	wickets	5.0	N	NaN	SJ Davis	DJ Harper
4	335986	Kolkata	2008-04-20	DJ Hussey	Eden Gardens	0	Kolkata Knight Riders	Deccan Chargers	Deccan Chargers	bat	Kolkata Knight Riders	wickets	5.0	N	NaN	BF Bowden	K Hariharan

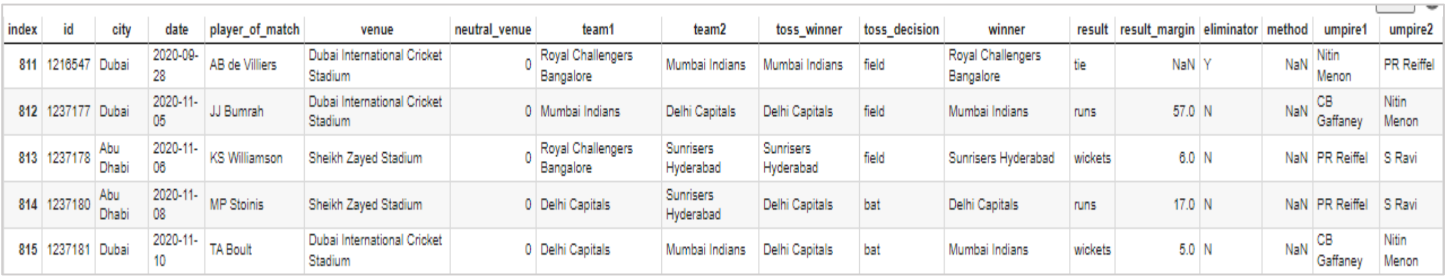
3.Loading Data of last 5 Rows

Code :



```
Data.tail(5)
```

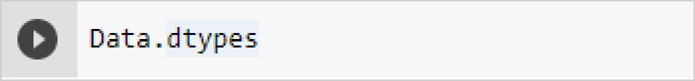
Output:



index	id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	result_margin	eliminator	method	umpire1	umpire2
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	NaN	Y	NaN	Nitin Menon	PR Reiffel
812	1237177	Dubai	2020-11-05	JJ Bumrah	Dubai International Cricket Stadium	0	Mumbai Indians	Delhi Capitals	Delhi Capitals	field	Mumbai Indians	runs	57.0	N	NaN	CB Gaffaney	Nitin Menon
813	1237178	Abu Dhabi	2020-11-06	KS Williamson	Sheikh Zayed Stadium	0	Royal Challengers Bangalore	Sunrisers Hyderabad	Sunrisers Hyderabad	field	Sunrisers Hyderabad	wickets	6.0	N	NaN	PR Reiffel	S Ravi
814	1237180	Abu Dhabi	2020-11-08	MP Stoinis	Sheikh Zayed Stadium	0	Delhi Capitals	Sunrisers Hyderabad	Delhi Capitals	bat	Delhi Capitals	runs	17.0	N	NaN	PR Reiffel	S Ravi
815	1237181	Dubai	2020-11-10	TA Boult	Dubai International Cricket Stadium	0	Delhi Capitals	Mumbai Indians	Delhi Capitals	bat	Mumbai Indians	wickets	5.0	N	NaN	CB Gaffaney	Nitin Menon

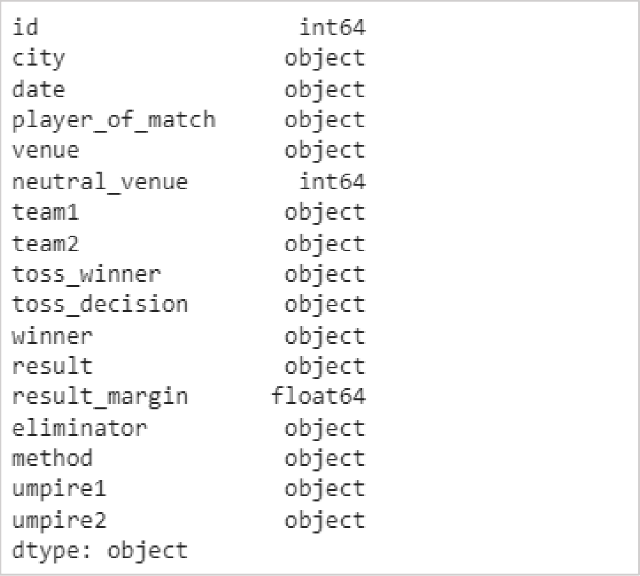
4. Checking Datatypes of Columns

Code :



```
Data.dtypes
```

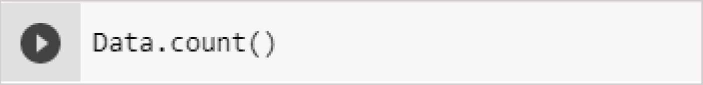
Output:



```
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
```

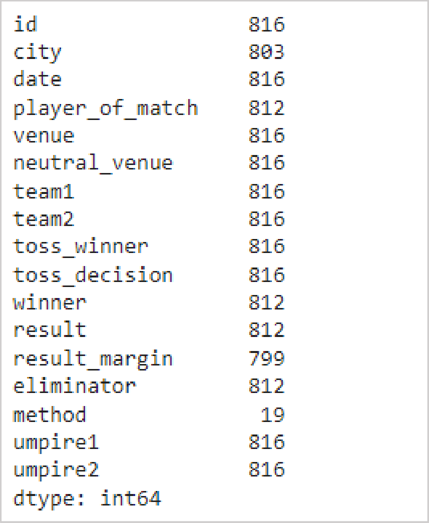
5.Counting Each Rows Data

Code :



```
Data.count()
```

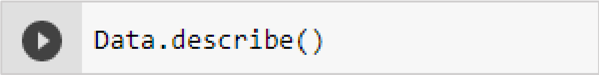
Output:



```
id                 816
city               803
date               816
player_of_match    812
venue              816
neutral_venue      816
team1              816
team2              816
toss_winner        816
toss_decision      816
winner             812
result             812
result_margin      799
eliminator         812
method              19
umpire1            816
umpire2            816
dtype: int64
```

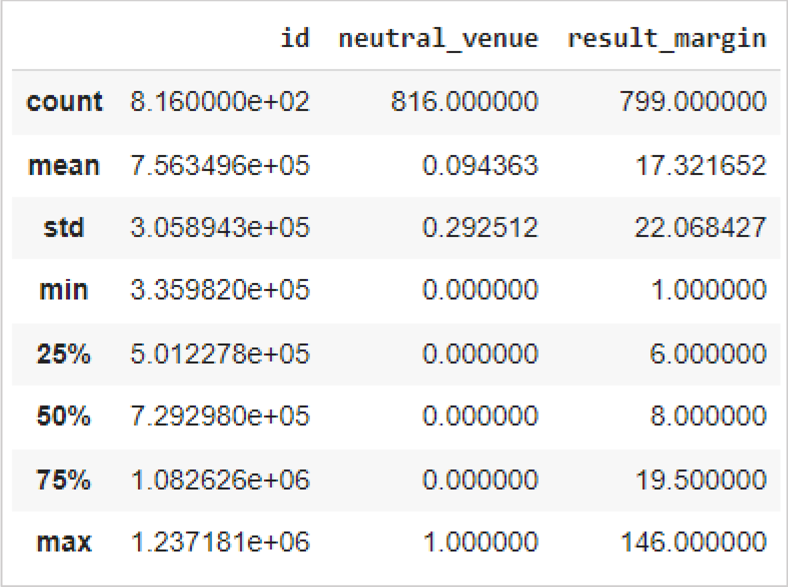
6. Describing Our Dataset

Code :



```
Data.describe()
```

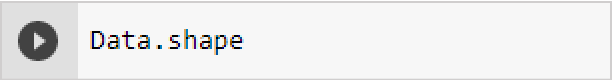
Output:



	id	neutral_venue	result_margin
count	8.160000e+02	816.000000	799.000000
mean	7.563496e+05	0.094363	17.321652
std	3.058943e+05	0.292512	22.068427
min	3.359820e+05	0.000000	1.000000
25%	5.012278e+05	0.000000	6.000000
50%	7.292980e+05	0.000000	8.000000
75%	1.082626e+06	0.000000	19.500000
max	1.237181e+06	1.000000	146.000000

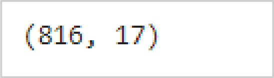
7. Checking No. of rows and columns present in dataset

Code :



```
Data.shape
```

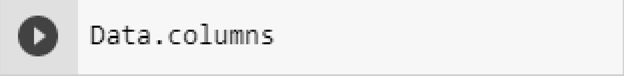
Output:



```
(816, 17)
```

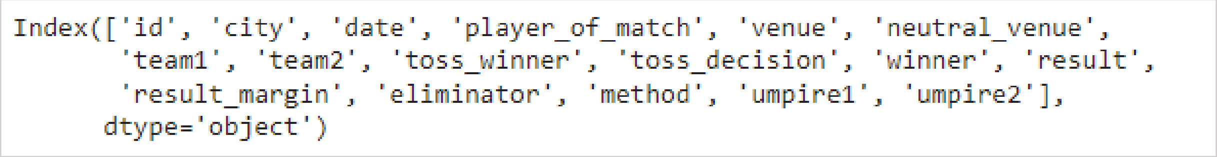
8.Printing all the column Names of the dataset

Code :



```
Data.columns
```

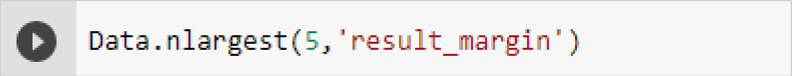
Output:



```
Index(['id', 'city', 'date', 'player_of_match', 'venue', 'neutral_venue',
       'team1', 'team2', 'toss_winner', 'toss_decision', 'winner', 'result',
       'result_margin', 'eliminator', 'method', 'umpire1', 'umpire2'],
      dtype='object')
```

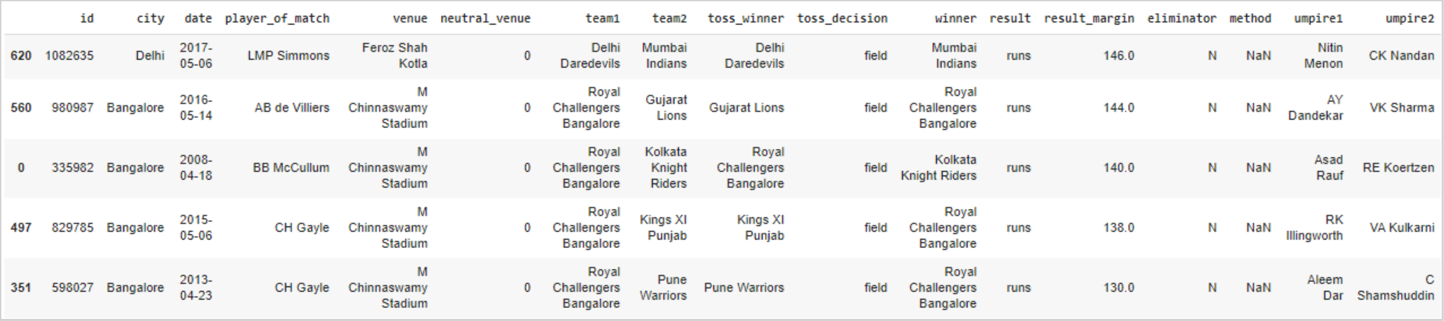
9.Finding top largest values from a particular column

Code :



```
Data.nlargest(5,'result_margin')
```

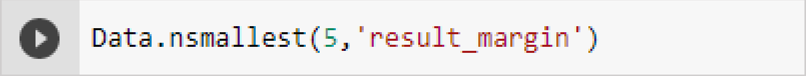
Output:



	id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	result_margin	eliminator	method	umpire1	umpire2
620	1082635	Delhi	2017-05-06	LMP Simmons	Feroz Shah Kotla	0	Delhi Daredevils	Mumbai Indians	Delhi Daredevils	field	Mumbai Indians	runs	146.0	N	NaN	Nitin Menon	CK Nandan
560	980987	Bangalore	2016-05-14	AB de Villiers	M Chinnaswamy Stadium	0	Royal Challengers Bangalore	Gujarat Lions	Gujarat Lions	field	Royal Challengers Bangalore	runs	144.0	N	NaN	AY Dandekar	VK Sharma
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	runs	140.0	N	NaN	Asad Rauf	RE Koertzen
497	829785	Bangalore	2015-05-06	CH Gayle	M Chinnaswamy Stadium	0	Royal Challengers Bangalore	Kings XI Punjab	Kings XI Punjab	field	Royal Challengers Bangalore	runs	138.0	N	NaN	RK Illingworth	VA Kulkarni
351	598027	Bangalore	2013-04-23	CH Gayle	M Chinnaswamy Stadium	0	Royal Challengers Bangalore	Pune Warriors	Pune Warriors	field	Royal Challengers Bangalore	runs	130.0	N	NaN	Aleem Dar	C Shamshuddin

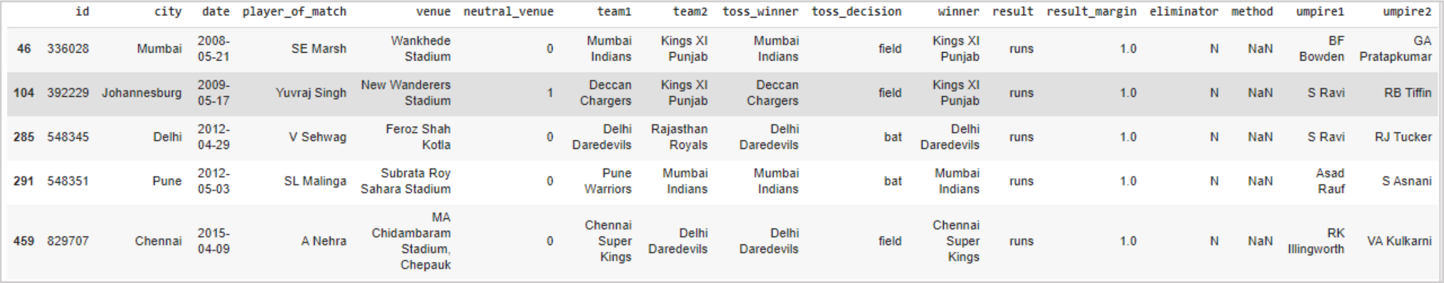
10.Finding smallest values from a particular column

Code :



```
Data.nsmallest(5,'result_margin')
```

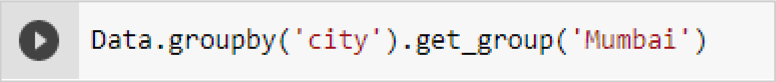
Output:



	id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	result_margin	eliminator	method	umpire1	umpire2
46	336028	Mumbai	2008-05-21	SE Marsh	Wankhede Stadium	0	Mumbai Indians	Kings XI Punjab	Mumbai Indians	field	Kings XI Punjab	runs	1.0	N	NaN	BF Bowden	GA Pratapkumar
104	392229	Johannesburg	2009-05-17	Yuvraj Singh	New Wanderers Stadium	1	Deccan Chargers	Kings XI Punjab	Deccan Chargers	field	Kings XI Punjab	runs	1.0	N	NaN	S Ravi	RB Tiffin
285	548345	Delhi	2012-04-29	V Sehwag	Feroz Shah Kotla	0	Delhi Daredevils	Rajasthan Royals	Delhi Daredevils	bat	Delhi Daredevils	runs	1.0	N	NaN	S Ravi	RJ Tucker
291	548351	Pune	2012-05-03	SL Malinga	Subrata Roy Sahara Stadium	0	Pune Warriors	Mumbai Indians	Mumbai Indians	bat	Mumbai Indians	runs	1.0	N	NaN	Asad Rauf	S Asnani
459	829707	Chennai	2015-04-09	A Nehra	MA Chidambaram Stadium, Chepauk	0	Chennai Super Kings	Delhi Daredevils	Delhi Daredevils	field	Chennai Super Kings	runs	1.0	N	NaN	RK Illingworth	VA Kulkarni

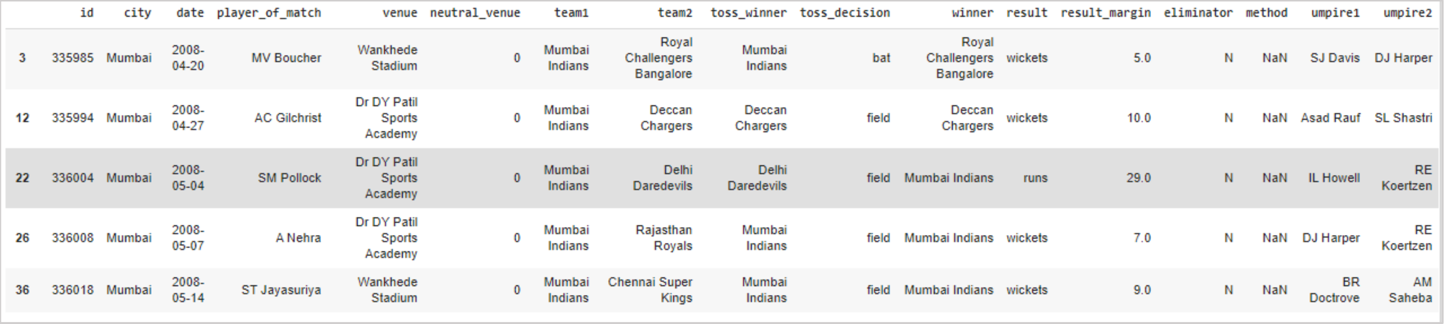
11. Splitting the data into groups

Code :



```
Data.groupby('city').get_group('Mumbai')
```

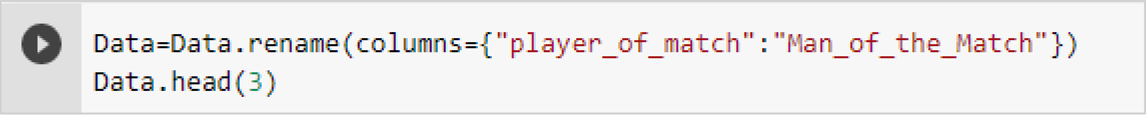
Output:



	id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	result_margin	eliminator	method	umpire1	umpire2
3	335985	Mumbai	2008-04-20	MV Boucher	Wankhede Stadium	0	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	Royal Challengers Bangalore	wickets	5.0	N	NaN	SJ Davis	DJ Harper
12	335994	Mumbai	2008-04-27	AC Gilchrist	Dr DY Patil Sports Academy	0	Mumbai Indians	Deccan Chargers	Deccan Chargers	field	Deccan Chargers	wickets	10.0	N	NaN	Asad Rauf	SL Shastri
22	336004	Mumbai	2008-05-04	SM Pollock	Dr DY Patil Sports Academy	0	Mumbai Indians	Delhi Daredevils	Delhi Daredevils	field	Mumbai Indians	runs	29.0	N	NaN	IL Howell	RE Koertzen
26	336008	Mumbai	2008-05-07	A Nehra	Dr DY Patil Sports Academy	0	Mumbai Indians	Rajasthan Royals	Mumbai Indians	field	Mumbai Indians	wickets	7.0	N	NaN	DJ Harper	RE Koertzen
36	336018	Mumbai	2008-05-14	ST Jayasuriya	Wankhede Stadium	0	Mumbai Indians	Chennai Super Kings	Mumbai Indians	field	Mumbai Indians	wickets	9.0	N	NaN	BR Doctrove	AM Saheba

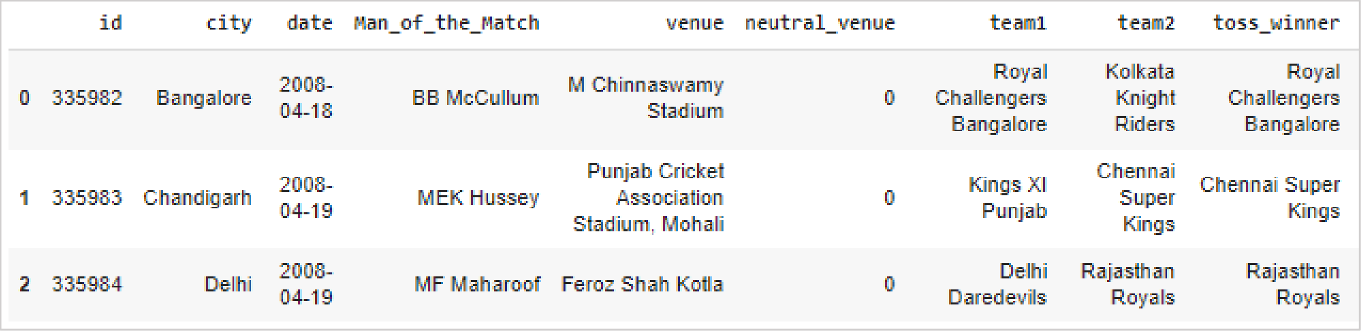
12.Renaming the column

Code :



```
Data=Data.rename(columns={"player_of_match":"Man_of_the_Match"})
Data.head(3)
```

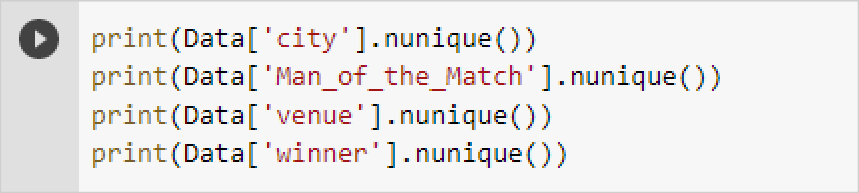
Output:



	id	city	date	Man_of_the_Match	venue	neutral_venue	team1	team2	toss_winner
0	335982	Bangalore	2008-04-18	BB McCullum	M Chinnaswamy Stadium	0	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore
1	335983	Chandigarh	2008-04-19	MEK Hussey	Punjab Cricket Association Stadium, Mohali	0	Kings XI Punjab	Chennai Super Kings	Chennai Super Kings
2	335984	Delhi	2008-04-19	MF Maharoof	Feroz Shah Kotla	0	Delhi Daredevils	Rajasthan Royals	Rajasthan Royals

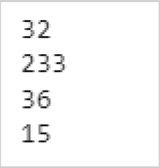
13. Printing Unique Values of rows

Code :



```
print(Data['city'].nunique())
print(Data['Man_of_the_Match'].nunique())
print(Data['venue'].nunique())
print(Data['winner'].nunique())
```

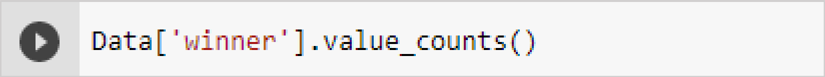
Output:



```
32
233
36
15
```

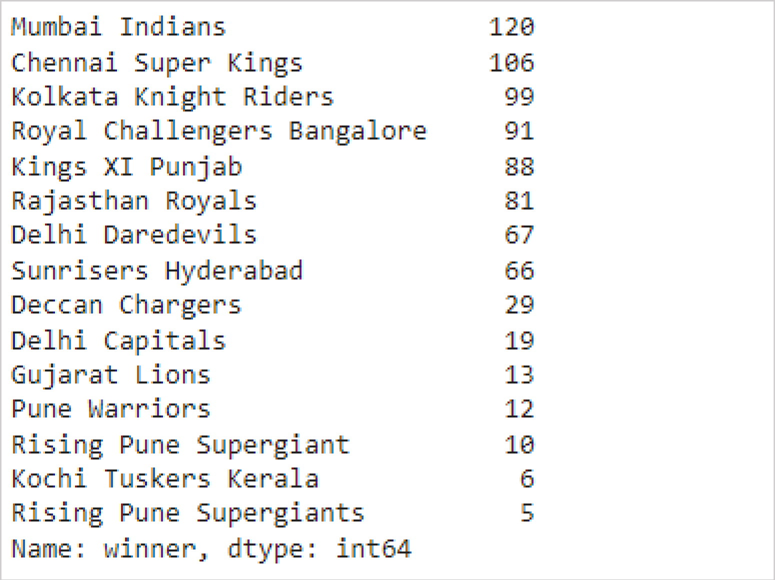
14. Printing Count of unique values by considering one particular column

Code :



```
Data['winner'].value_counts()
```

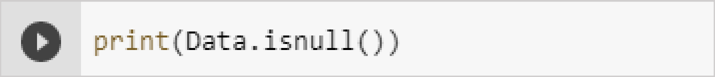
Output:



```
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
```

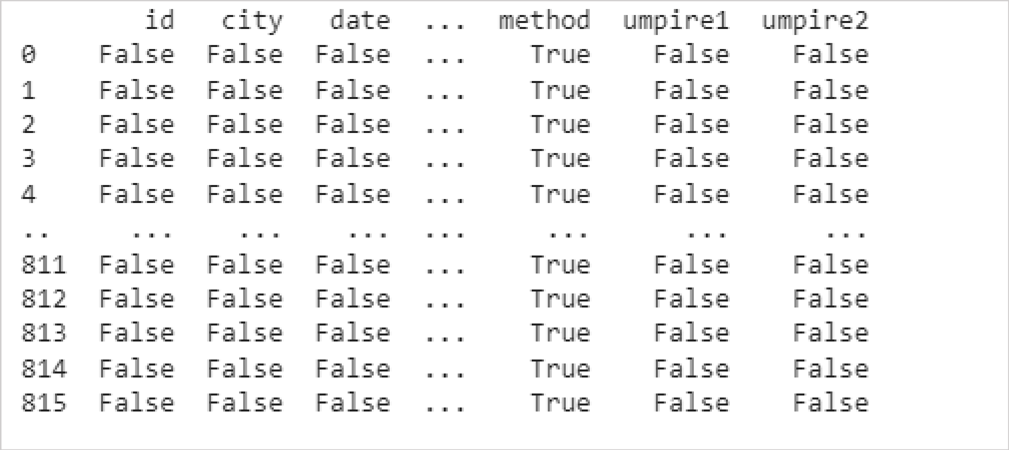
15.Checking for null values

Code :



```
print(Data.isnull())
```

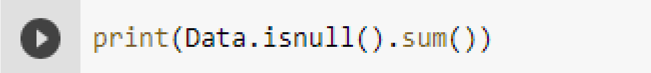
Output:



```
        id   city   date  ...  method  umpire1  umpire2
0    False  False  False  ...    True    False    False
1    False  False  False  ...    True    False    False
2    False  False  False  ...    True    False    False
3    False  False  False  ...    True    False    False
4    False  False  False  ...    True    False    False
..     ...    ...    ...  ...     ...      ...      ...
811  False  False  False  ...    True    False    False
812  False  False  False  ...    True    False    False
813  False  False  False  ...    True    False    False
814  False  False  False  ...    True    False    False
815  False  False  False  ...    True    False    False
```

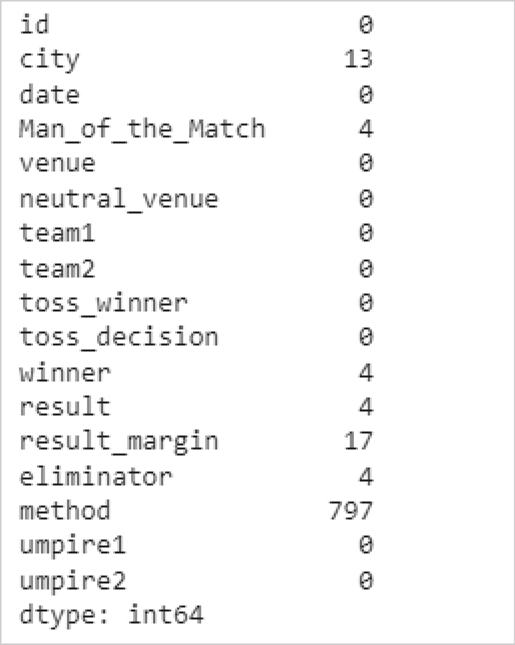
16.Counting Total Null Values

Code :



```
print(Data.isnull().sum())
```

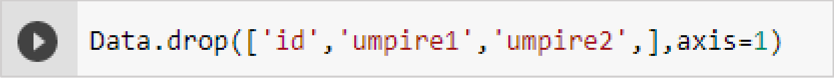
Output:



```
id                   0
city                13
date                 0
Man_of_the_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
```

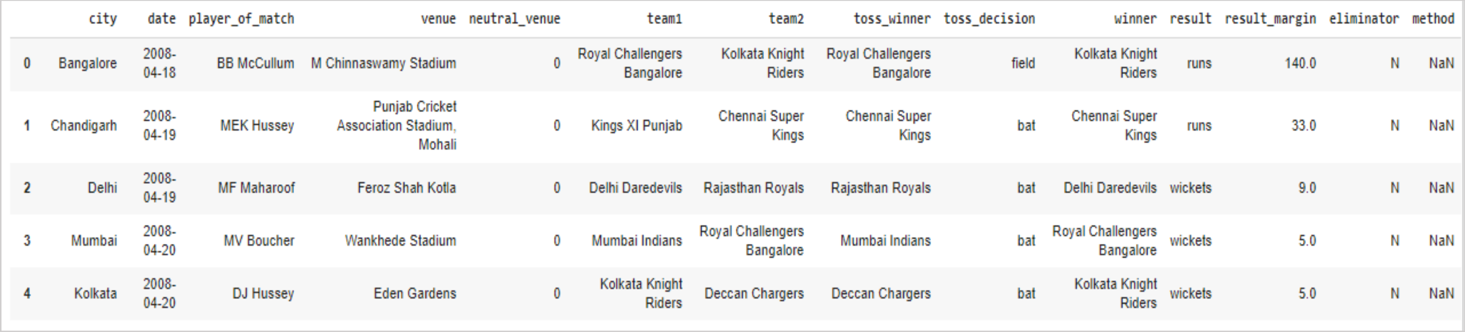
17.Dropping Irrelevent Columns

Code:



```
Data.drop(['id','umpire1','umpire2',],axis=1)
```

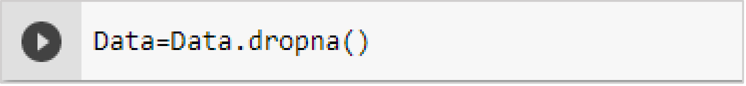
Output:



	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	result_margin	eliminator	method
0	Bangalore	2008-04-18	BB McCullum	M Chinnaswamy Stadium	0	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore	field	Kolkata Knight Riders	runs	140.0	N	NaN
1	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	runs	33.0	N	NaN
2	Delhi	2008-04-19	MF Maharoof	Feroz Shah Kotla	0	Delhi Daredevils	Rajasthan Royals	Rajasthan Royals	bat	Delhi Daredevils	wickets	9.0	N	NaN
3	Mumbai	2008-04-20	MV Boucher	Wankhede Stadium	0	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	Royal Challengers Bangalore	wickets	5.0	N	NaN
4	Kolkata	2008-04-20	DJ Hussey	Eden Gardens	0	Kolkata Knight Riders	Deccan Chargers	Deccan Chargers	bat	Kolkata Knight Riders	wickets	5.0	N	NaN

18.Removing Null Values

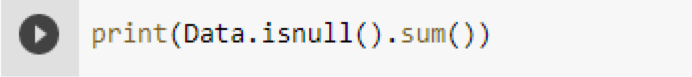
Code:



```
Data=Data.dropna()
```

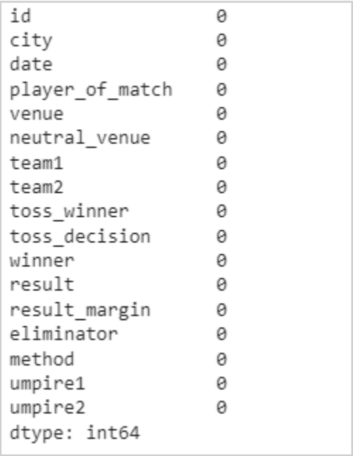
19. Verifying Is there any null value remain in the dataset or not

Code:



```
print(Data.isnull().sum())
```

Output:



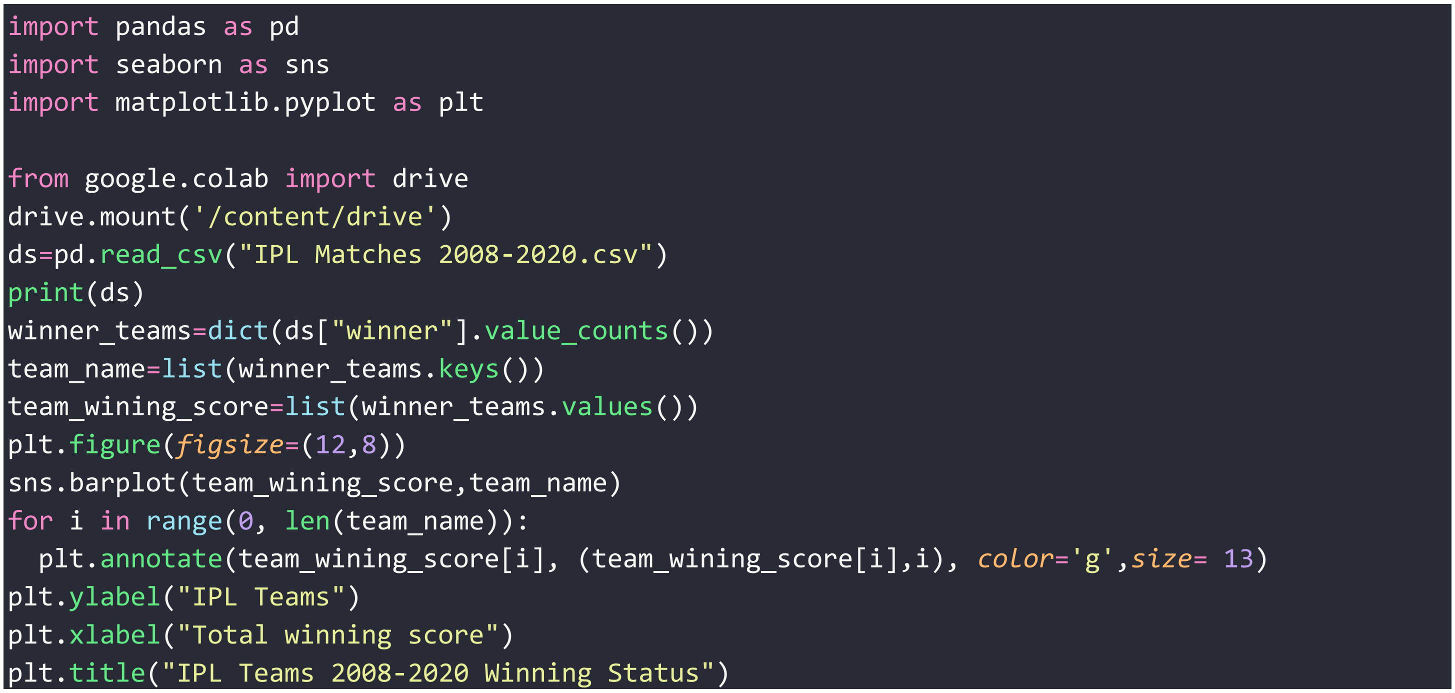
```
id                 0
city               0
date               0
player_of_match    0
venue              0
neutral_venue      0
team1              0
team2              0
toss_winner        0
toss_decision      0
winner             0
result             0
result_margin      0
eliminator         0
method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

PLOTTING GRAPHS ON THE BASIS OF DATASET

1) BARCHART(USING SEABORN)

A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. A bar chart describes the comparisons between the discrete categories.

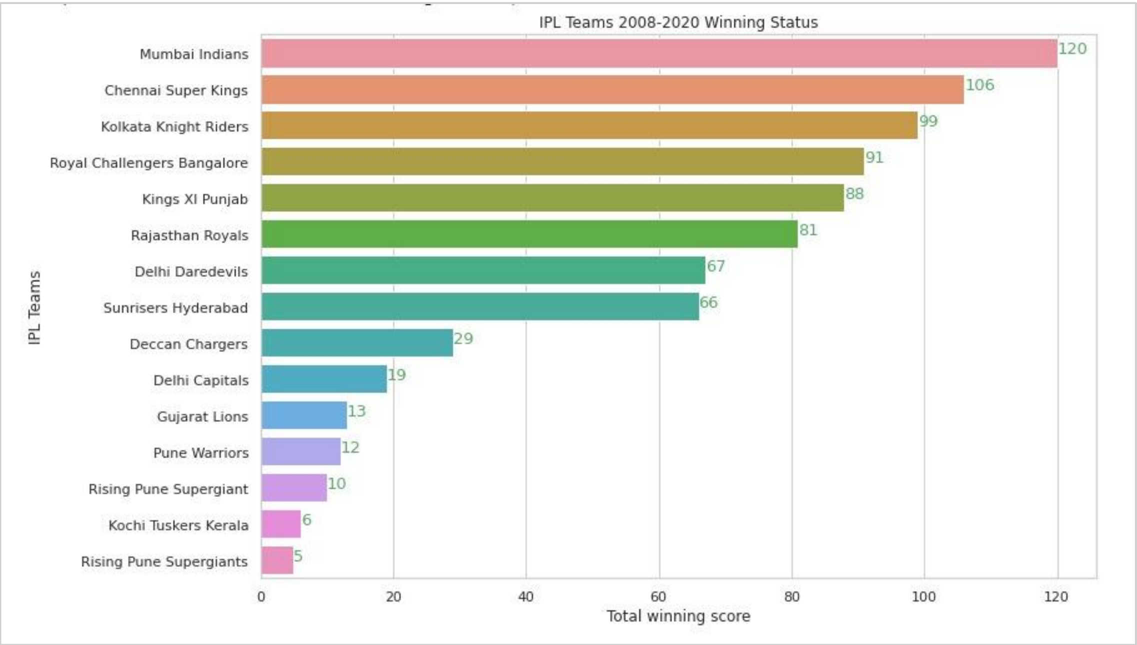
Code:



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

from google.colab import drive
drive.mount('/content/drive')
ds=pd.read_csv("IPL Matches 2008-2020.csv")
print(ds)
winner_teams=dict(ds["winner"].value_counts())
team_name=list(winner_teams.keys())
team_wining_score=list(winner_teams.values())
plt.figure(figsize=(12,8))
sns.barplot(team_wining_score,team_name)
for i in range(0, len(team_name)):
  plt.annotate(team_wining_score[i], (team_wining_score[i],i), color='g',size= 13)
plt.ylabel("IPL Teams")
plt.xlabel("Total winning score")
plt.title("IPL Teams 2008-2020 Winning Status")
```

Output :



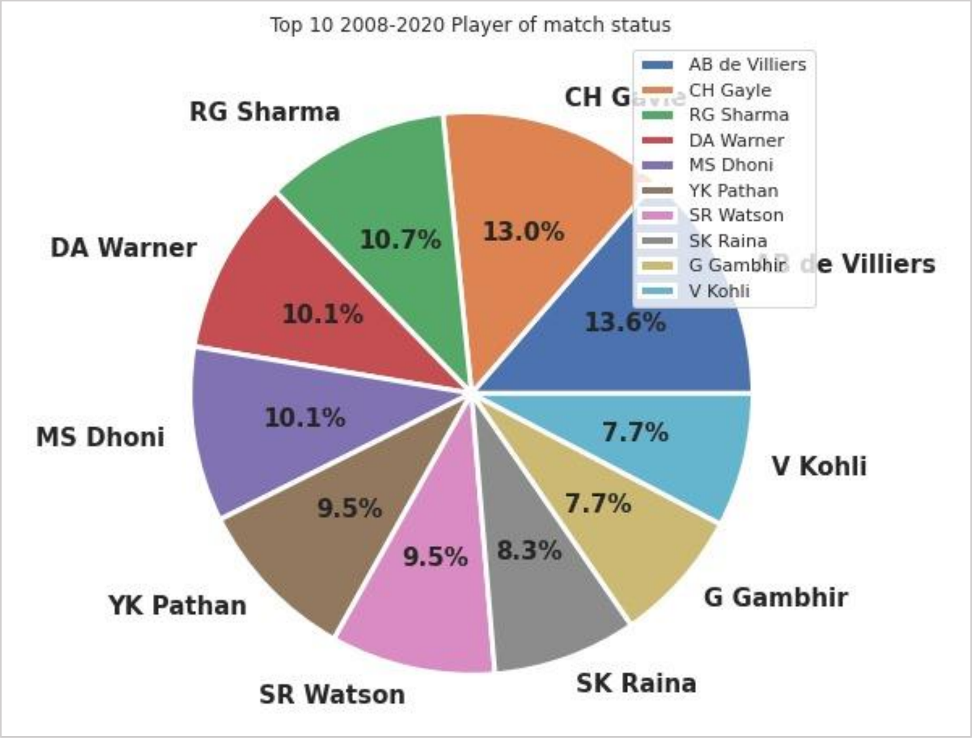
2) PIEPLOT

A Pie Chart is a circular statistical plot that can display only one series of data. The area of the chart is the total percentage of the given data. The area of slices of the pie represents the percentage of the parts of the data. The slices of pie are called wedges.

Code:

```
player_of_matches=dict(ds['player_of_match'].value_counts().head(10))
plt.figure(figsize=(12,8))
player_name=list(player_of_matches.keys())
player_man_of_MatchesScore=list(player_of_matches.values())
plt.pie(player_man_of_MatchesScore,labels=player_name,
textprops={'fontweight':'bold','fontsize':15}, wedgeprops={'linewidth':
3,'edgecolor':'white'}, autopct="%2.1f%%")
plt.title("Top 10 2008-2020 Player of match status")
plt.legend()
plt.show()
```

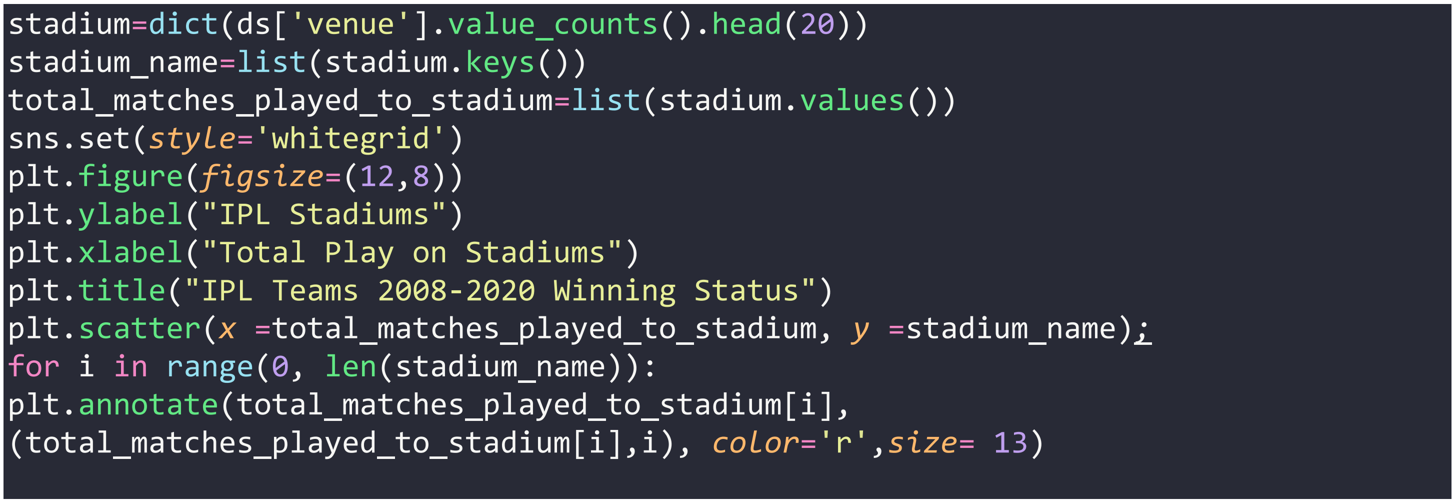
Output:



3) SCATTERPLOT

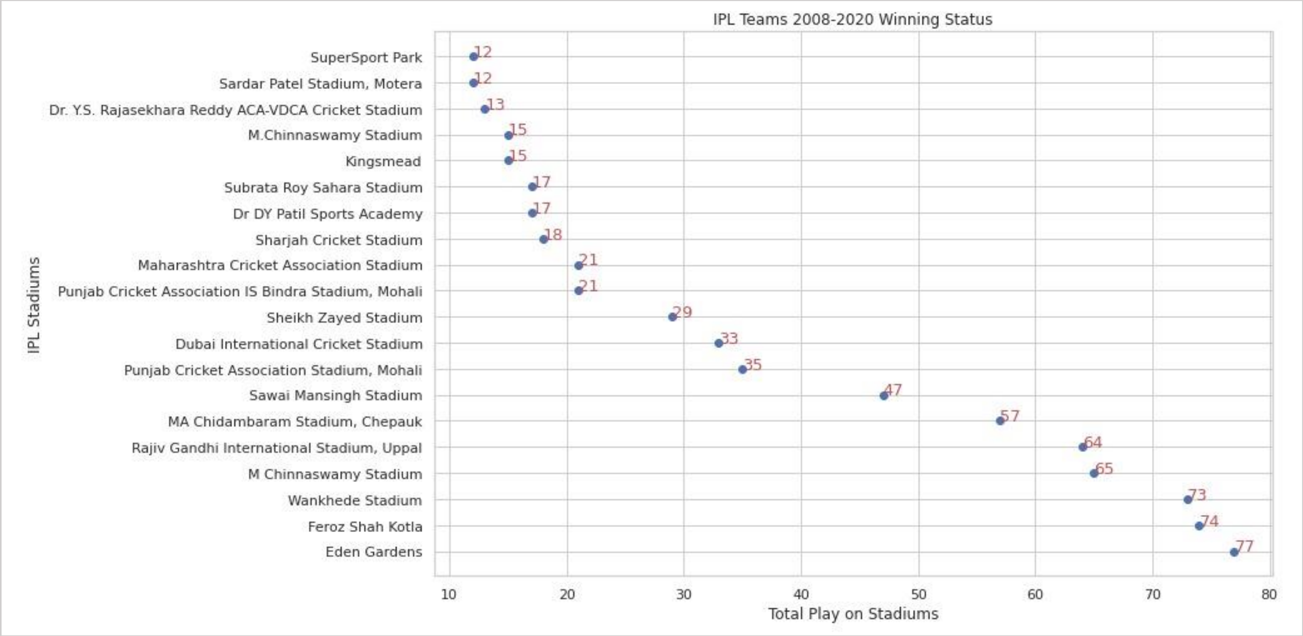
Scatter plots are used to observe relationship between variables and uses dots to represent the relationship between them. Scatter plots are widely used to represent relation among variables and how change in one affects the other.

Code :



```
stadium=dict(ds['venue'].value_counts().head(20))
stadium_name=list(stadium.keys())
total_matches_played_to_stadium=list(stadium.values())
sns.set(style='whitegrid')
plt.figure(figsize=(12,8))
plt.ylabel("IPL Stadiums")
plt.xlabel("Total Play on Stadiums")
plt.title("IPL Teams 2008-2020 Winning Status")
plt.scatter(x =total_matches_played_to_stadium, y =stadium_name);
for i in range(0, len(stadium_name)):
plt.annotate(total_matches_played_to_stadium[i],
(total_matches_played_to_stadium[i],i), color='r',size= 13)
```

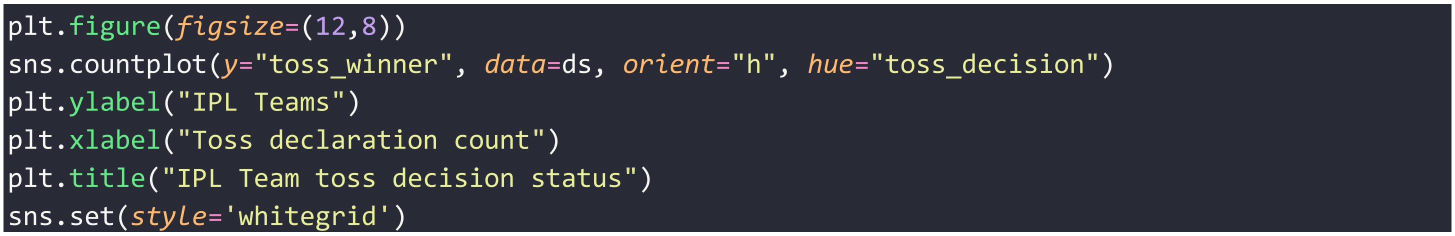
Output :



4) COUNTPLOT

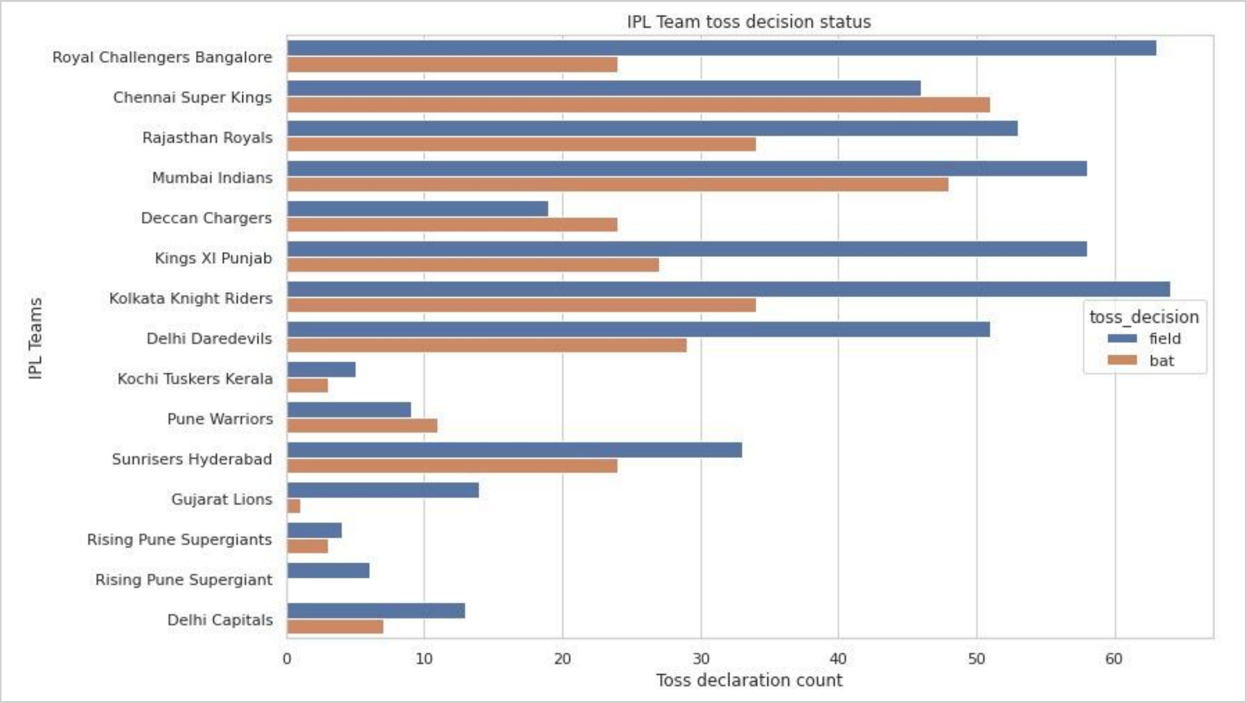
seaborn.countplot() is used to Show the counts of observations in each categorical bin using bars.

Code:



```
plt.figure(figsize=(12,8))
sns.countplot(y="toss_winner", data=ds, orient="h", hue="toss_decision")
plt.ylabel("IPL Teams")
plt.xlabel("Toss declaration count")
plt.title("IPL Team toss decision status")
sns.set(style='whitegrid')
```

Output:

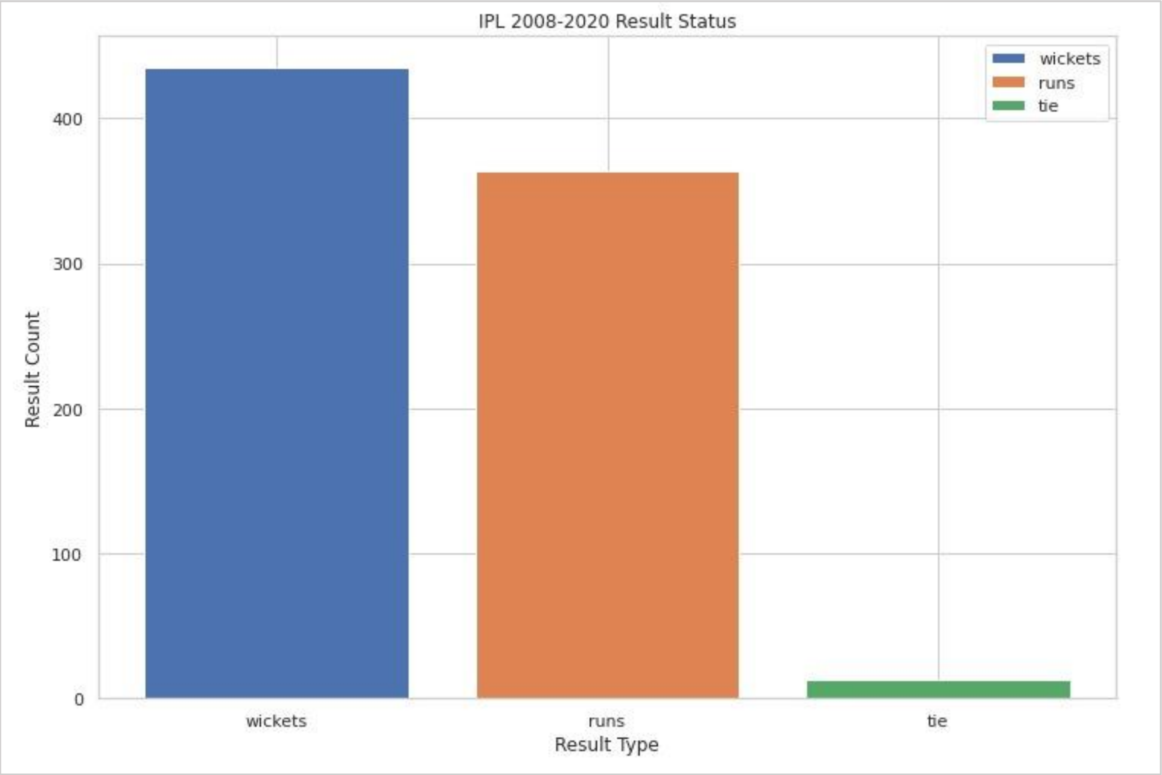


5) BARPLOT(USING MATPLOTLIB)

Code :

```
result=dict(ds['result'].value_counts().head(20))
plt.figure(figsize=(12,8))
plt.ylabel("Result Count")
plt.xlabel("Result Type")
plt.title("IPL 2008-2020 Result Status")
result_x=list(result.keys())
result_y=list(result.values())
for i in range(0, len(result_x)):
plt.bar(result_x[i],result_y[i], label=result_x[i])
plt.legend()
```

Output :



THANK YOU

Pillai