Sports Data Analysis

07 March 2024 22:56 PM

T20I Men's Cricket Match Data (2005 - 2023)

I have a sports-related dataset on Kaggle for the see some data understanding .Kaggle, and I plan to start a project using tools like Python for data cleaning and EDA.

Need to first install the below library

This is the code for making a report - Summary Below are the HTML -- Report_report

```
pip install ydata_profiling
from ydata_profiling import ProfileReport
profile = ProfileReport(dataset, title="Profiling Report")
profile.to_file("your_report.html")
```

Help of this library we create a good EDA Workbook where I can the all analysis

I will open this data and I will check all column once by one

your_report ← **This is the Report link, You can chick here to see the report**

There are a total of 96 teams that participated in T20 matches, and we have the list from 2003 to 2023.

Below are the some question's will find the solution :

1. I will check how many times the 'Bat First' team has won and how many times the 'Bat Second' team has won.
2. I will create a line plot for the total matches year by year.
3. Who is the player with the highest runs or the leading wicket-taker?
4. Who is the player who has played the most matches, and what is the total score?
5. Maximum sixes and fours.
6. Strike rate.
7. What are the top 10 venues where the maximum number of matches were played?
8. Top 10 highest-scoring batsmen.
9. Top 10 wicket-taking bowlers.
10. Top 10 batsmen (how many times they scored >=100).
11. Top 10 batsmen (how many times they scored >=50).
12. In which month of the year are the maximum number of matches played?
13. Which are the top 10 teams that have won the maximum number of matches?
14. Will make a function for every player and team where if you give this function some input it will give you the some statistical analysis like(

    Mean, Median, Min, Max , 25th percentile , 50 percentile, 75 percentile like this)

EDA

Columns name with description:

Match ID	Unique identifier for each match.
Date	Date of the match.
Venue	Location where the match was played.
Bat First	Team batting first.
Bat Second	Team batting second.
Innings	Inning number.
Over	Over number in the inning.

Ball	Ball number in the over.
Batter	Batsman facing the ball.
Non Striker	Batsman at the non-striker's end.
Bowler	Bowler bowling the ball.
Batter Runs	Runs scored by the batsman.
Extra Runs	Extra runs conceded.
Runs From Ball	Total runs from the ball.
Ball Rebowled	Indicates if the ball is rebowled.
Extra Type	Type of extra (e.g., wide, no-ball).
Wicket	Wicket status (e.g., not out, caught).
Method	**Method of dismissing the batsman.**
Player Out	**Player who got out.**
Innings Runs	Total runs in the inning.
Innings Wickets	Wickets fallen in the inning.
Target Score	Target score for the chasing team.
Runs to Get	**Runs required to win.**
Balls Remaining	Remaining balls to be bowled.
Winner	Winning team.
Chased Successfully	Indicator if the chase was successful.
Total Batter Runs	Total runs scored by all batsmen.
Total Non-Striker Runs	Total runs scored by non-strikers.
Batter Balls Faced	Total balls faced by batsmen.
Non Striker Balls Faced	Total balls faced by non-strikers.
Player Out Runs	**Runs scored by the player who got out.**
Player Out Balls Faced	**Balls faced by the player who got out.**
Bowler Runs Conceded	Runs conceded by the bowler.
Valid Ball	Indicator if the ball is valid for play.

First I will understand the data :

Total Data shape:
(Total rows: 113631, Columns :35)

Dataset info:

There are a total of 35 columns, where 11 are object columns, 21 are integers, and 3 are floats. Upon reviewing this data.
I noticed that some columns have missing values. We need to decide whether to remove or handle these missing values in those columns

```
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   Unnamed: 0            113631 non-null  int64
 1   Match ID              113631 non-null  int64
 2   Date                  113631 non-null  object
 3   Venue                 113631 non-null  object
 4   Bat First             113631 non-null  object
 5   Bat Second            113631 non-null  object
 6   Innings               113631 non-null  int64
 7   Over                  113631 non-null  int64
 8   Ball                  113631 non-null  int64
 9   Batter                113631 non-null  object
 10  Non Striker           113631 non-null  object
 11  Bowler                113631 non-null  object
 12  Batter Runs           113631 non-null  int64
 13  Extra Runs            113631 non-null  int64
 14  Runs From Ball        113631 non-null  int64
 15  Ball Rebowled         113631 non-null  int64
 16  Extra Type            113631 non-null  object
 17  Wicket                113631 non-null  int64
 18  Method                6265 non-null    object
 19  Player Out            6265 non-null    object
 20  Innings Runs          113631 non-null  int64
 21  Innings Wickets       113631 non-null  int64
 22  Target Score          113631 non-null  int64
 23  Runs to Get           53267 non-null   float64
 24  Balls Remaining       113631 non-null  int64
 25  Winner                113631 non-null  object
```

Dataset Describe:

Describe function provides statistical insights into data, summarizing central tendencies, spread, and distribution for each numeric column

	Unnamed: 0	Match ID	Innings	Over	Ball	Batter Runs	Extra Runs	[illegible] Ball	Ball Rebowled	Wicket	...
count	113631.000000	1.136310e+05	113631.000000	113631.000000	113631.000000	113631.000000	113631.000000	113631.000000	113631.000000	113631.000000	...
mean	56815.000000	1.084905e+06	1.468772	9.945631	3.488150	1.137603	0.076467	1.214070	0.040632	0.055135	...
std	32802.588556	3.312175e+05	0.499026	5.642274	1.708698	1.545731	0.364279	1.535963	0.197436	0.228244	...
min	0.000000	2.110280e+05	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
25%	28407.500000	9.513590e+05	1.000000	5.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
50%	56815.000000	1.251577e+06	1.000000	10.000000	3.000000	1.000000	0.000000	1.000000	0.000000	0.000000	...
75%	85222.500000	1.320977e+06	2.000000	15.000000	5.000000	1.000000	0.000000	1.000000	0.000000	0.000000	...
max	113630.000000	1.391337e+06	2.000000	20.000000	7.000000	6.000000	5.000000	7.000000	1.000000	1.000000	...

Checking null values:

I will not drop any col

```
Match ID                  0.000000
Date                      0.000000
Venue                     0.000000
Bat First                 0.000000
Bat Second                0.000000
Innings                   0.000000
Over                      0.000000
Ball                      0.000000
Batter                    0.000000
Non Striker               0.000000
Bowler                    0.000000
Batter Runs               0.000000
Extra Runs                0.000000
Runs From Ball            0.000000
Ball Rebowled             0.000000
Extra Type                0.000000
Wicket                    0.000000
Method                   94.486540
Player Out               94.486540
Innings Runs              0.000000
Innings Wickets           0.000000
Target Score              0.000000
Runs to Get              53.122827
Balls Remaining           0.000000
Winner                    0.000000
Chased Successfully       0.000000
Total Batter Runs         0.000000
Total Non Striker Runs    0.000000
Batter Balls Faced        0.000000
Non Striker Balls Faced   0.000000
Player Out Runs          94.486540
Player Out Balls Faced   94.486540
```

I will solve each problem we discussed above one by one:

1. **I will check how many times the 'Bat First' team has won and how many times the 'Bat Second' team has won.**

We have a total of 236 matches played from 2003 to 2023, where the majority of matches were won by teams batting second, with a winning percentage of 53%, while teams batting first achieved victories in 47% of the matches

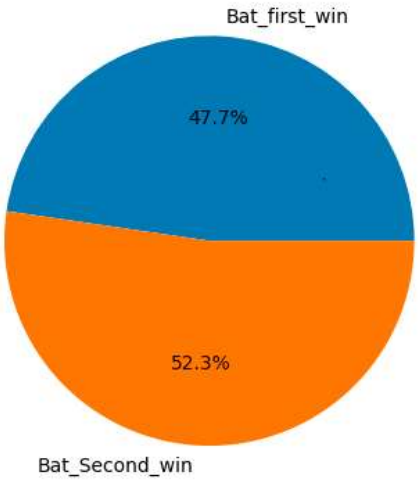
Below are the code and Graph:

```
Bat_first_win=dataset[dataset['Bat First']==dataset['Winner']]['Match ID'].nunique()
Bat_second_win=dataset[dataset['Bat Second']==dataset['Winner']]['Match ID'].nunique()
print('Bat first win time:', Bat_first_win, ' ', 'Bat second win time', Bat_second_win)
# Bat first  %
print('Bat first %',round((Bat_first_win/dataset['Match ID'].nunique())*100,2))
# Bat second %
print('Bat Second %',round((Bat_second_win/dataset['Match ID'].nunique())*100,2))
# plot some graph
```

```
plt.pie(x=[Bat_first_win,Bat_second_win],autopct='%1.1f%%',labels=['Bat_first_win','Bat_Second_win'])
plt.show()
```

Output:

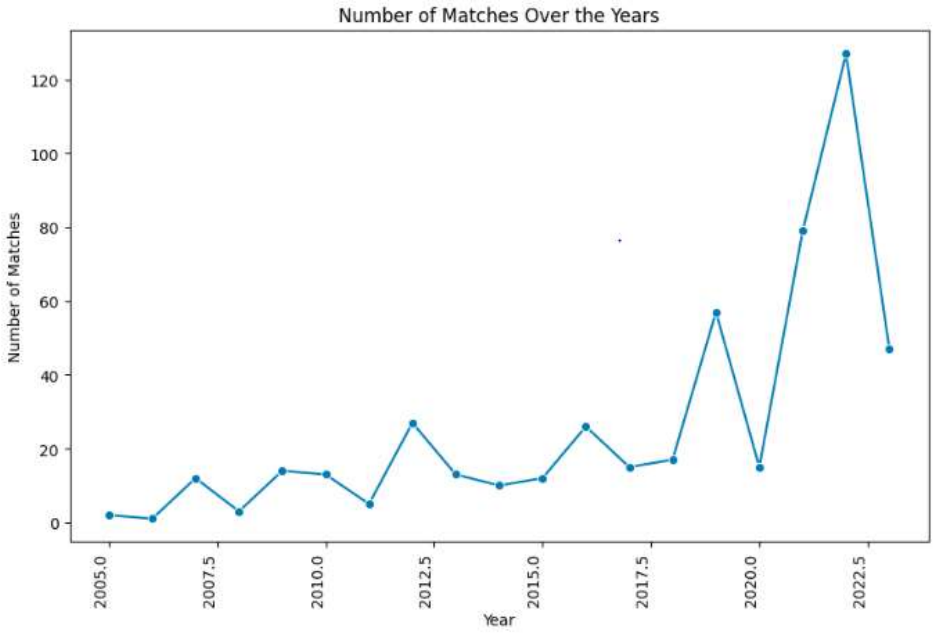
Total Mach : 236
Bat first win time: 236 Bat second win time 259
Bat first % :47.68
Bat Second % :52.32



2- I will create a line plot for the total matches year by year.

We can see the in the below graph the maximum number of matches are played in 2021 and 2022

```
YoY_match=dataset[['Match ID','Date']].drop_duplicates(keep='first')
YoY_match['Date'] = pd.to_datetime(YoY_match['Date'])
YoY_match['Year']=YoY_match['Date'].dt.year
plt.figure(figsize=(10, 6))
sn.lineplot(data=YoY_match['Year'].value_counts(),x=YoY_match['Year'].value_counts().index,y=YoY_match['Year'].value_counts().values, marker='o')
plt.xticks(rotation=90, ha='right')
plt.xlabel('Year')
plt.ylabel('Number of Matches')
plt.title('Number of Matches Over the Years')
plt.show()
```



3- Who is the player with the highest runs or the leading wicket-taker?

- **The Highest Run batsmen is : Babar Azam with 1274 runs**
- **The Highest Wicket taker is : TG Southee with 38 Wicket**

Note:- I am not sure this data is correct bcs I am taking this data from Kaggle

```
# Highest Run
# Highest Wicket
# Highest run player
Highest_bestmen=dataset.groupby('Batter')['Batter Runs'].sum().sort_values(ascending=False).head(1)
Highest_wicket=dataset.groupby('Bowler')['Wicket'].sum().sort_values(ascending=False).head(1)
print('Highest Run Bestmen',Highest_bestmen,' ',' Highest Wicket Bowler',Highest_wicket)
```

4- Who is the player who has played the most matches, and what is the total score?

The player who has played the most matches is 'Mahmudullah', with a total of 650 runs.

```
# Trems of bestmen
temp_df=dataset[['Match ID','Batter']].drop_duplicates(keep='first')
Better=temp_df['Batter'].value_counts().head(1).index[0]
total_runs=dataset[dataset['Batter']==temp_df['Batter'].value_counts().head(1).index[0]]['Batter Runs'].sum()
print('most matched payed Bestmen is : ',Better)
print(" ")
print('The total runs',Better,':',total_runs)
```

5- Maximum sixes and fours.

- **The player with the highest number of sixes is N Pooran.**
- **The player with the highest number of fours is Babar Azam.**

```
temp_df=dataset[(dataset['Batter Runs']==6)]
highest_sixes=temp_df['Batter'].value_counts().head(1).index[0]
temp_df=dataset[(dataset['Batter Runs']==4)]
highest_four=temp_df['Batter'].value_counts().head(1).index[0]
print('highest sixes :',highest_sixes)
print()
print('Highest Four :', highest_four)
```

6- Strike rate.

10 player highest strike rate:

$$\text{Strike Rate} = \left(\frac{\text{Total Runs}}{\text{Total Balls Faced}}\right) \times 100$$

 # Make function for stick rate

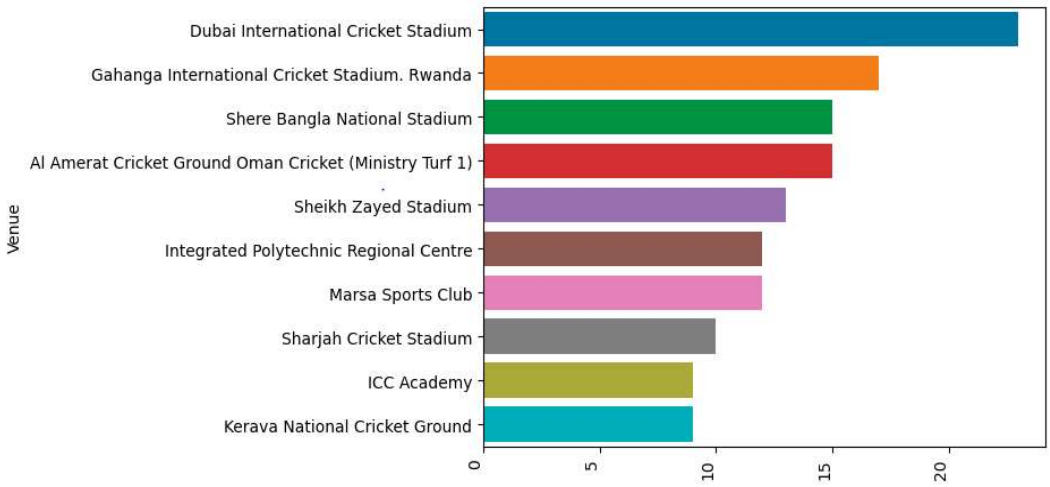
```
def Strike_rate(boll,runs):
    return (boll/runs)*100


temp_df=dataset[['Batter','Batter Runs']]
temp_df=pd.pivot_table(data=temp_df, index='Batter', aggfunc={'Batter Runs': ['count', 'sum']})
temp_df.columns = [temp_df.columns[0], 'Total_bqll']
temp_df.columns = [temp_df.columns[1], 'Total_runs']
temp_df['Strike_rate']=Strike_rate(temp_df['Total_ball'],temp_df['Total_runs'])
temp_df.sort_values('Strike_rate',ascending=False).head(10)
```

Batter	Total_ball	Total_runs	Strike_rate
PM Nevill	3	13	433.333333
NW Bracken	1	4	400.000000
Aziz Sualley	1	4	400.000000
Ashiqullah Said	1	4	400.000000
Mohammad Asif	1	4	400.000000
SSB Magala	5	18	360.000000
LC Le Tissier	9	29	322.222222
Aamir Lal	12	34	283.333333
M Yunusu Issa	4	11	275.000000
GC Galanis	7	19	271.428571

7- What are the top 10 venues where the maximum number of matches were played?

Below are the top 10 Venues.

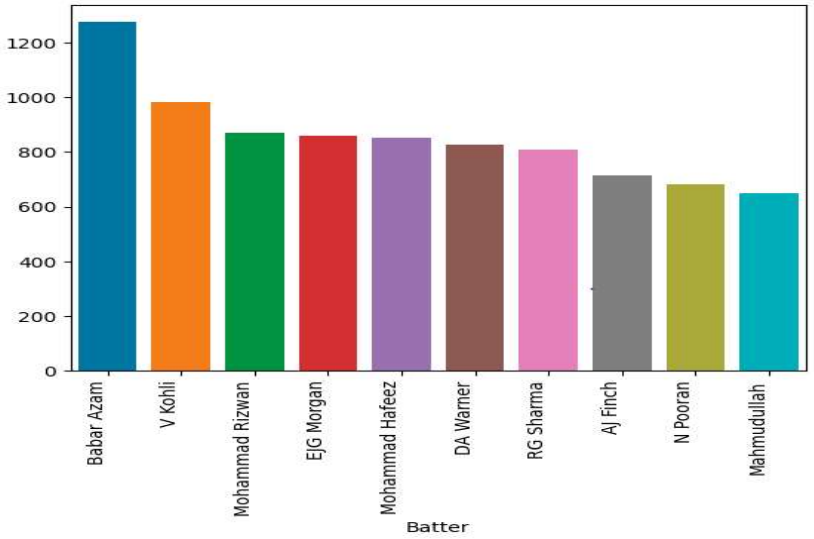
```
Venue
Dubai International Cricket Stadium                        23
Gahanga International Cricket Stadium. Rwanda              17
Shere Bangla National Stadium                              15
Al Amerat Cricket Ground Oman Cricket (Ministry Turf 1)    15
Sheikh Zayed Stadium                                       13
Integrated Polytechnic Regional Centre                     12
Marsa Sports Club                                          12
Sharjah Cricket Stadium                                    10
ICC Academy                                                 9
Kerava National Cricket Ground                              9
```



8- **Top 10 highest-scoring batsmen**.

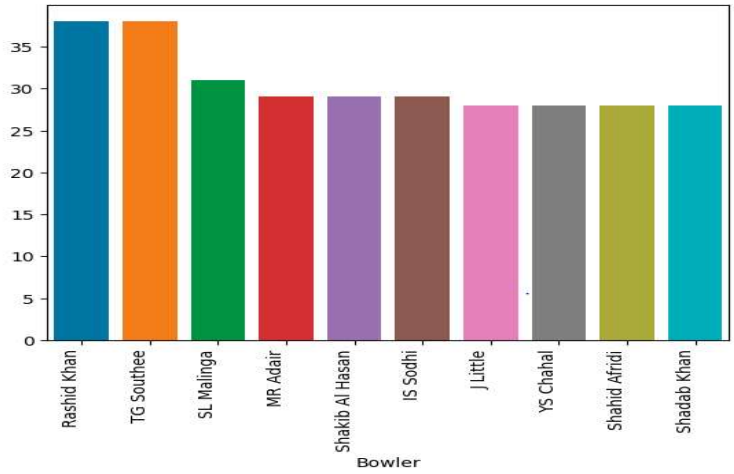
This correction ensures clarity and corrects the hyphenation and grammar.

Batter	
Babar Azam	1274
V Kohli	981
Mohammad Rizwan	871
EJG Morgan	858
Mohammad Hafeez	851
DA Warner	828
RG Sharma	809
AJ Finch	714
N Pooran	681
Mahmudullah	650



9- **Top 10 wicket-taking bowlers.**

Bowler	
Rashid Khan	38
TG Southee	38
SL Malinga	31
MR Adair	29
Shakib Al Hasan	29
IS Sodhi	29
J Little	28
YS Chahal	28
Shahid Afridi	28
Shadab Khan	28

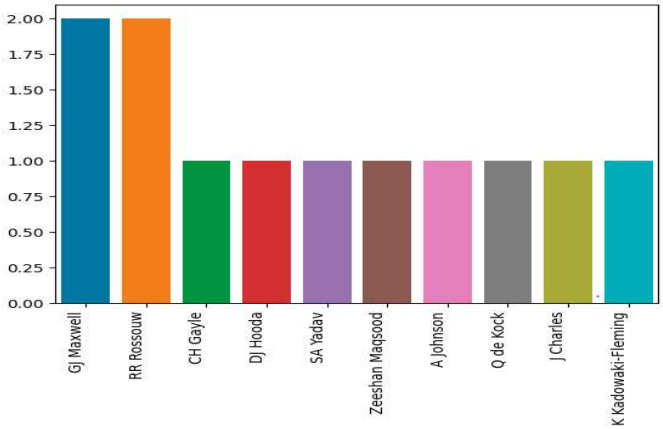


10- **Top 10 batsmen (how many times they scored >=100).**

Below are the graph and summary

```
temp_df=dataset[['Match ID','Batter','Batter Runs']]
temp_df=temp_df.groupby(['Match ID','Batter'])['Batter Runs'].sum()
temp_df=temp_df.reset_index()
temp_df=temp_df[temp_df['Batter Runs']>=100]
temp_df=temp_df['Batter'].value_counts().head(10)
# ploting grah
sn.barplot(x=temp_df.index,y=temp_df.values)
plt.xticks(rotation=90, ha='right')
plt.savefig('new.png')
plt.show()
temp_df
```

```
Batter
GJ Maxwell              2
RR Rossouw              2
CH Gayle                1
DJ Hooda                1
SA Yadav                1
Zeeshan Maqsood         1
A Johnson               1
Q de Kock               1
J Charles               1
K Kadowaki-Fleming      1
```

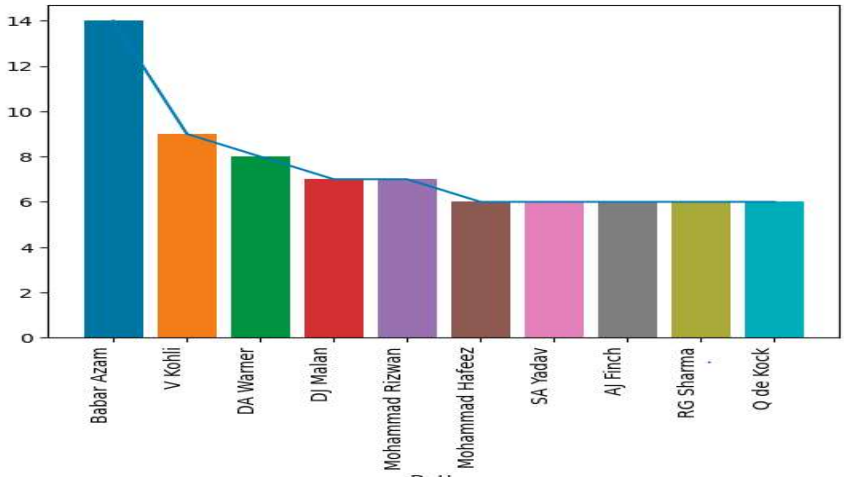


11- Top 10 batsmen (how many times they scored >=50).

The top 10 scorers include Babar Azam as the first, and the second is Virat Kohli, who has crossed the >50 score in the matches.

```
temp_df=dataset[['Match ID','Batter','Batter Runs']]
temp_df=temp_df.groupby(['Match ID','Batter'])['Batter Runs'].sum()
temp_df=temp_df.reset_index()
temp_df=temp_df[temp_df['Batter Runs']>=50]
temp_df=temp_df['Batter'].value_counts().head(10)
# ploting grah
sn.barplot(x=temp_df.index,y=temp_df.values)
plt.xticks(rotation=90, ha='right')
plt.savefig('new.png')
plt.show()
temp_df
```

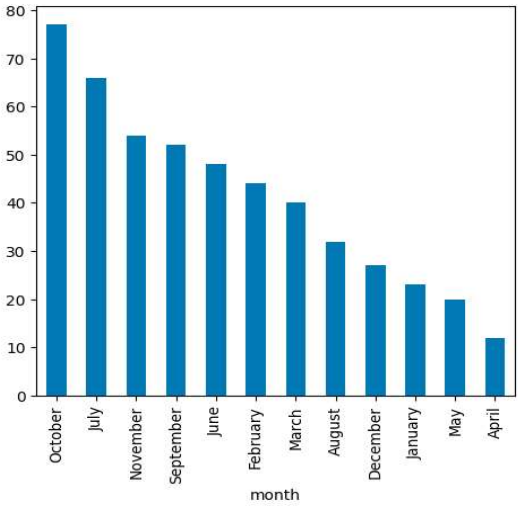
```
Batter
Babar Azam          14
V Kohli              9
DA Warner            8
DJ Malan             7
Mohammad Rizwan      7
Mohammad Hafeez      6
SA Yadav             6
AJ Finch             6
RG Sharma            6
Q de Kock            6
```

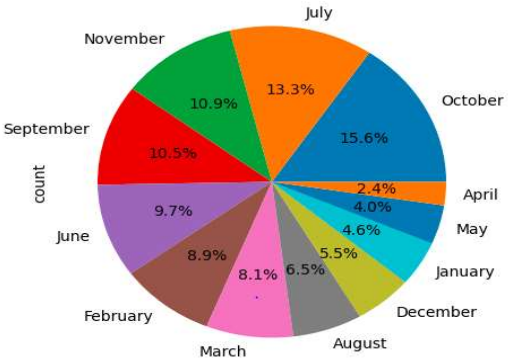


12- In which month of the year are the maximum number of matches played?

I will plot a graph where the x-axis represents the month and the y-axis represents the total number of matches

We can see that in the month of October, 15% of matches are played by the team





13- Which are the top 10 teams that have won the maximum number of matches?

According to the Kaggle data, Pakistan has the highest number of matches, and India holds the second position

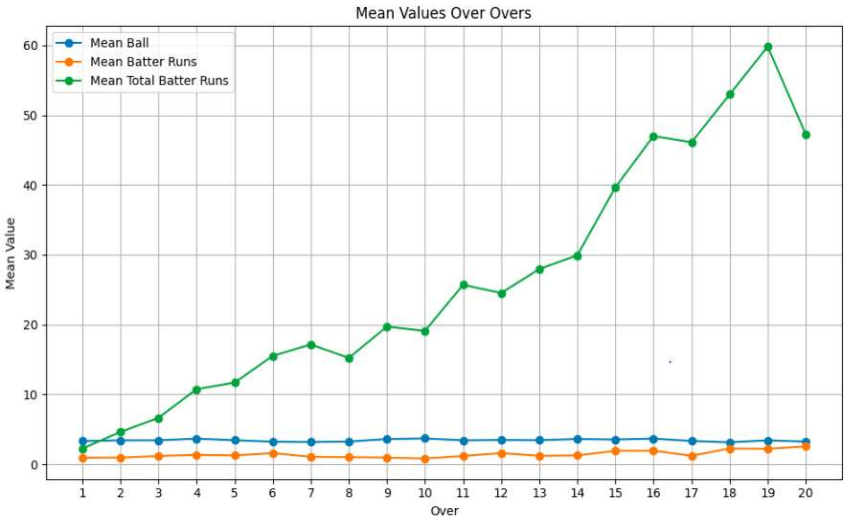
Below are the Plotly visualization :

14- Will make a function for every player and team where if you give this function some input it will give you the some statistical analysis like(Mean, Median, Min, Max like this)

Below are the function output and graph:

<u>This graph and table for selected better player</u>

	Ball		Batter Runs				Total Batter Runs		
	mean	count	mean	sum	max	min	mean	sum	max
Over									
1	3.312500	16	0.937500	15	4	0	2.250000	36	7
2	3.433333	30	0.966667	29	4	0	4.633333	139	17
3	3.432432	37	1.189189	44	4	0	6.648649	246	23
4	3.673913	46	1.347826	62	6	0	10.760870	495	28
5	3.442308	52	1.269231	66	4	0	11.692308	608	34
6	3.238095	42	1.595238	67	6	0	15.523810	652	38
7	3.188679	53	1.075472	57	6	0	17.150943	909	43
8	3.256410	39	1.025641	40	4	0	15.230769	594	45
9	3.612245	49	0.959184	47	6	0	19.714286	966	52
10	3.695652	46	0.826087	38	4	0	19.108696	879	56



Conclusion:

From 2005 to 2023, teams batting second prevailed in 53% of matches, indicating a trend favoring chasing. Peak match volumes were observed in 2021-2022. Star performers include Babar Azam (highest run-scorer), TG Southee (leading wicket-taker), and Mahmudullah (most matches played). Pakistan leads in match count, followed by India

I am uploading this notebook in Kaggle and GitHub

Thanks