# PYTHON PROJECT

## Data analysis -Sports

# Introduction

Data science is the study of data to extract knowledge and insights from the data and apply knowledge and actionable insights. In this tutorial, we will work on IPL Data Analysis and Visualization Project using Python where we will explore interesting insights from the data of IPL matches like most run by a player, most wicket taken by a player, and much more from IPL season 2008-2020.

## Importing Libraries

# 1.Loding Data

```
: import pandas as pd
  import numpy as np
  import matplotlib.pyplot as plt
  import seaborn as sns
```

```
: mat=pd.read_csv("matches.csv")
```

```
: dev=pd.read_csv("deliveries.csv")
```

# Importing IPL Dataset

```
mat.head()
```

| | id | season | city | date | team1 | team2 | toss_winner | toss_decision | result | dl_applied | winner | win_by_runs | win_by_wickets | player_of_n |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2017 | Hyderabad | 2017-04-05 | Sunrisers Hyderabad | Royal Challengers Bangalore | Royal Challengers Bangalore | field | normal | 0 | Sunrisers Hyderabad | 35 | 0 | Yuvraj |
| 1 | 2 | 2017 | Pune | 2017-04-06 | Mumbai Indians | Rising Pune Supergiant | Rising Pune Supergiant | field | normal | 0 | Rising Pune Supergiant | 0 | 7 | SPD |
| 2 | 3 | 2017 | Rajkot | 2017-04-07 | Gujarat Lions | Kolkata Knight Riders | Kolkata Knight Riders | field | normal | 0 | Kolkata Knight Riders | 0 | 10 | CA |
| 3 | 4 | 2017 | Indore | 2017-04-08 | Rising Pune Supergiant | Kings XI Punjab | Kings XI Punjab | field | normal | 0 | Kings XI Punjab | 0 | 6 | GJ Ma |
| 4 | 5 | 2017 | Bangalore | 2017-04-08 | Royal Challengers Bangalore | Delhi Daredevils | Royal Challengers Bangalore | bat | normal | 0 | Royal Challengers Bangalore | 15 | 0 | KM Ja |

```
dev.head()
```

| | match_id | inning | batting_team | bowling_team | over | ball | batsman | non_striker | bowler | is_super_over | ... | bye_runs | legbye_runs | noball_runs | penalty_runs |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | Sunrisers Hyderabad | Royal Challengers Bangalore | 1 | 1 | DA Warner | S Dhawan | TS Mills | 0 | ... | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | Sunrisers Hyderabad | Royal Challengers Bangalore | 1 | 2 | DA Warner | S Dhawan | TS Mills | 0 | ... | 0 | 0 | 0 | 0 |
| 2 | 1 | 1 | Sunrisers Hyderabad | Royal Challengers Bangalore | 1 | 3 | DA Warner | S Dhawan | TS Mills | 0 | ... | 0 | 0 | 0 | 0 |
| 3 | 1 | 1 | Sunrisers Hyderabad | Royal Challengers Bangalore | 1 | 4 | DA Warner | S Dhawan | TS Mills | 0 | ... | 0 | 0 | 0 | 0 |

# IPL Data Analysis and Visualization with Python

```
mat.shape
```

```
(756, 18)
```

```
mat.info()
```

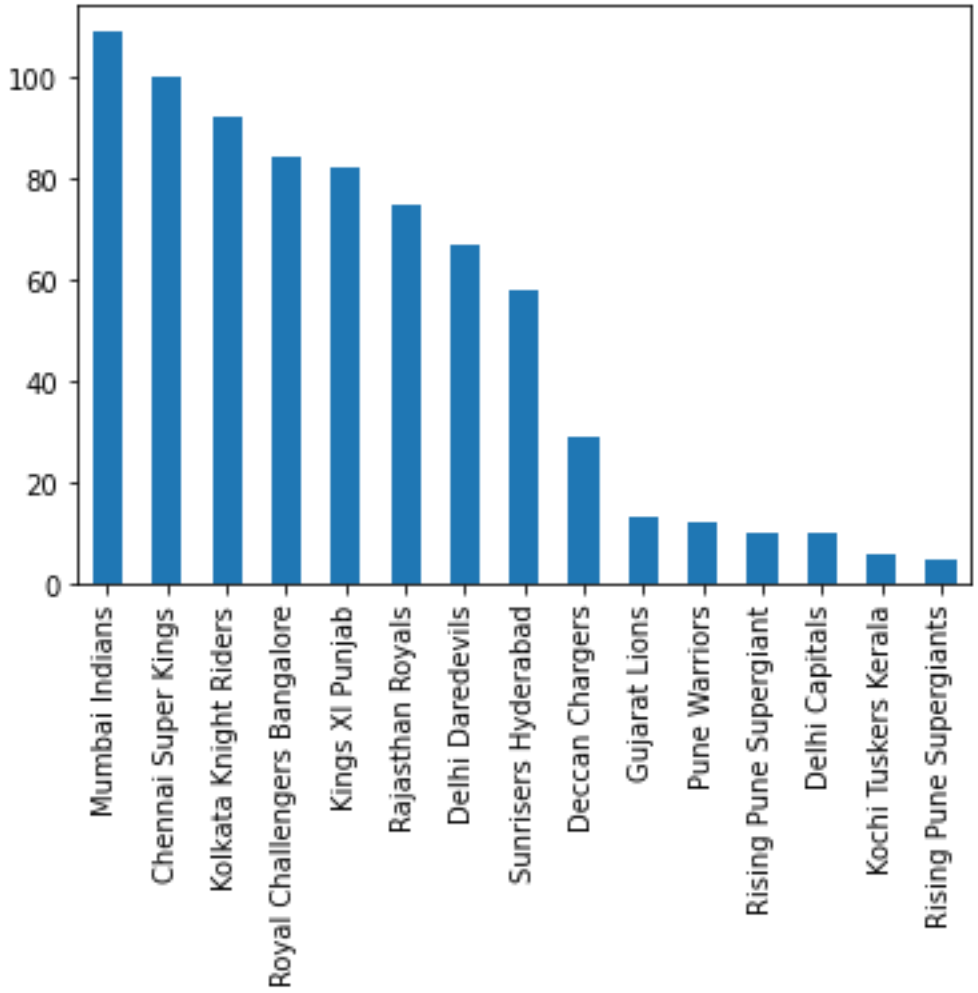
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 756 entries, 0 to 755
Data columns (total 18 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   id               756 non-null    int64
 1   season           756 non-null    int64
 2   city             749 non-null    object
 3   date             756 non-null    object
 4   team1            756 non-null    object
 5   team2            756 non-null    object
 6   toss_winner      756 non-null    object
 7   toss_decision    756 non-null    object
 8   result           756 non-null    object
 9   dl_applied       756 non-null    int64
 10  winner           752 non-null    object
 11  win_by_runs      756 non-null    int64
 12  win_by_wickets   756 non-null    int64
 13  player_of_match  752 non-null    object
 14  venue            756 non-null    object
 15  umpire1          754 non-null    object
 16  umpire2          754 non-null    object
 17  umpire3          119 non-null    object
dtypes: int64(5), object(13)
memory usage: 106.4+ KB
```

```
mat.describe()
```

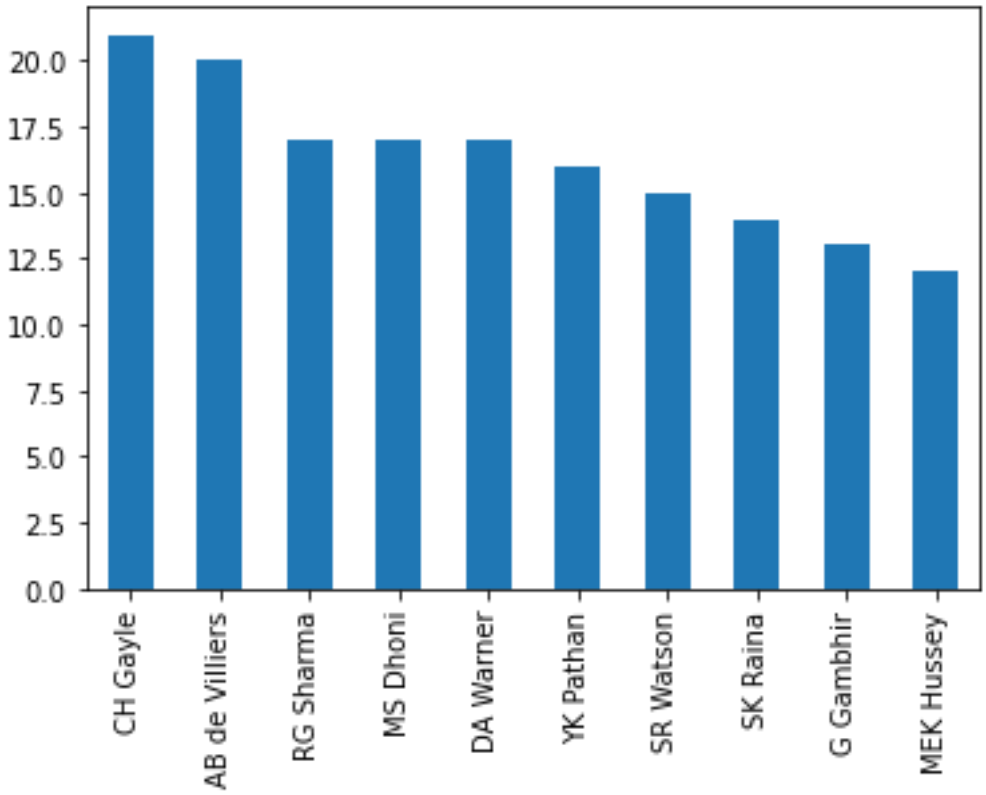
| | match_id | season | dl_applied | win_by_runs | win_by_wickets |
|---|---|---|---|---|---|
| count | 756.000000 | 756.000000 | 756.000000 | 756.000000 | 756.000000 |
| mean | 1792.178571 | 2013.444444 | 0.025132 | 13.283069 | 3.350529 |

# visualizing the data

1.Total match played by all team

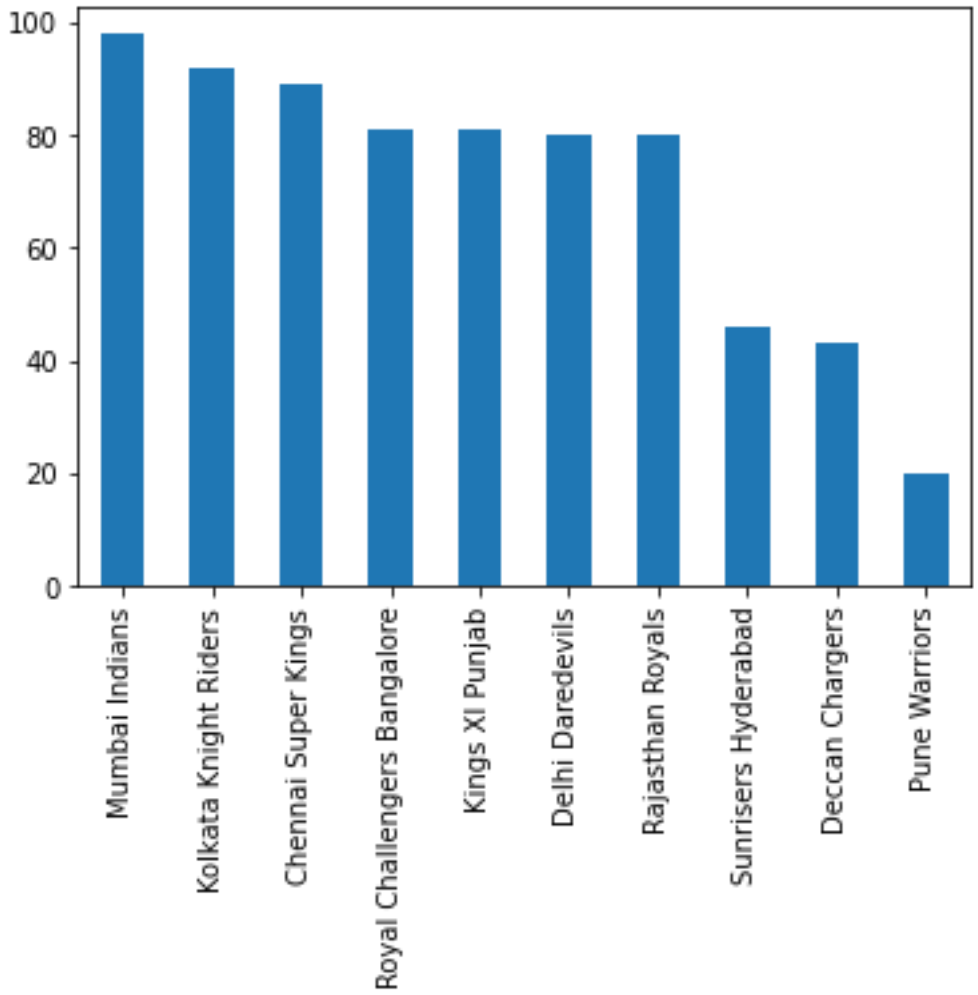



2. maximum time won the player of the match




2.

3.toss win by perticuler team




## Team that won most matches by batting first

```
mat.groupby(["toss_winner"]).agg({"winner":["count"]}).sort_values(ascending=False,by=("winner","count")).head(1)
```

| | winner |
|---|---|
| | count |
| toss_winner | |
| Mumbai Indians | 98 |

# 4.Data Analysis:

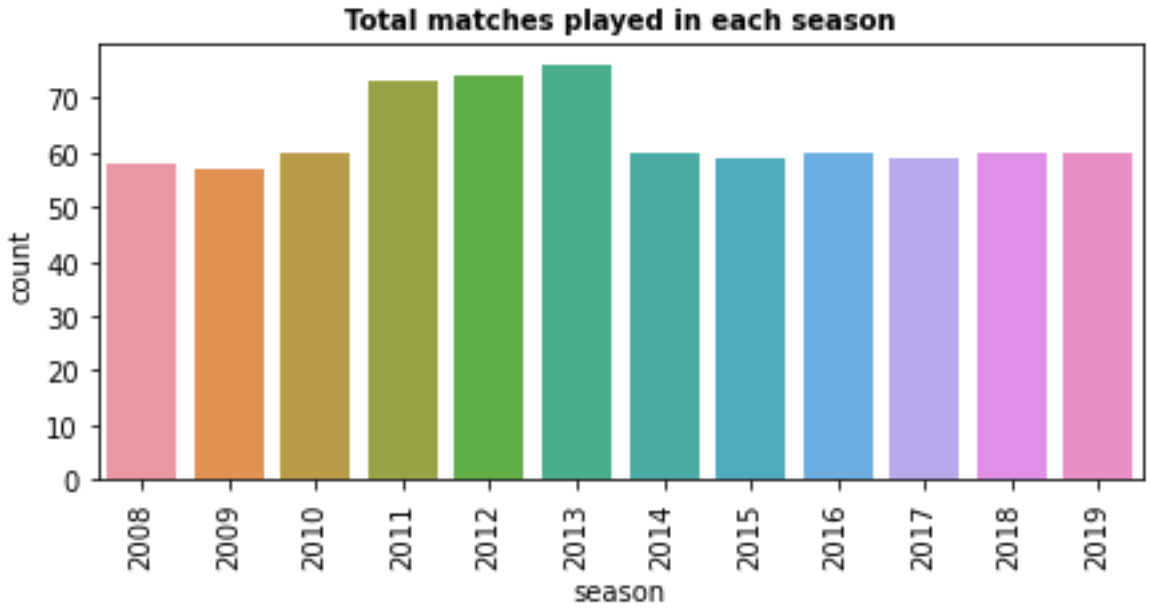
# Merging the two Datasets into a new datasets and read it

```
df=pd.merge(mat,dev,on="match_id",how="left")
```

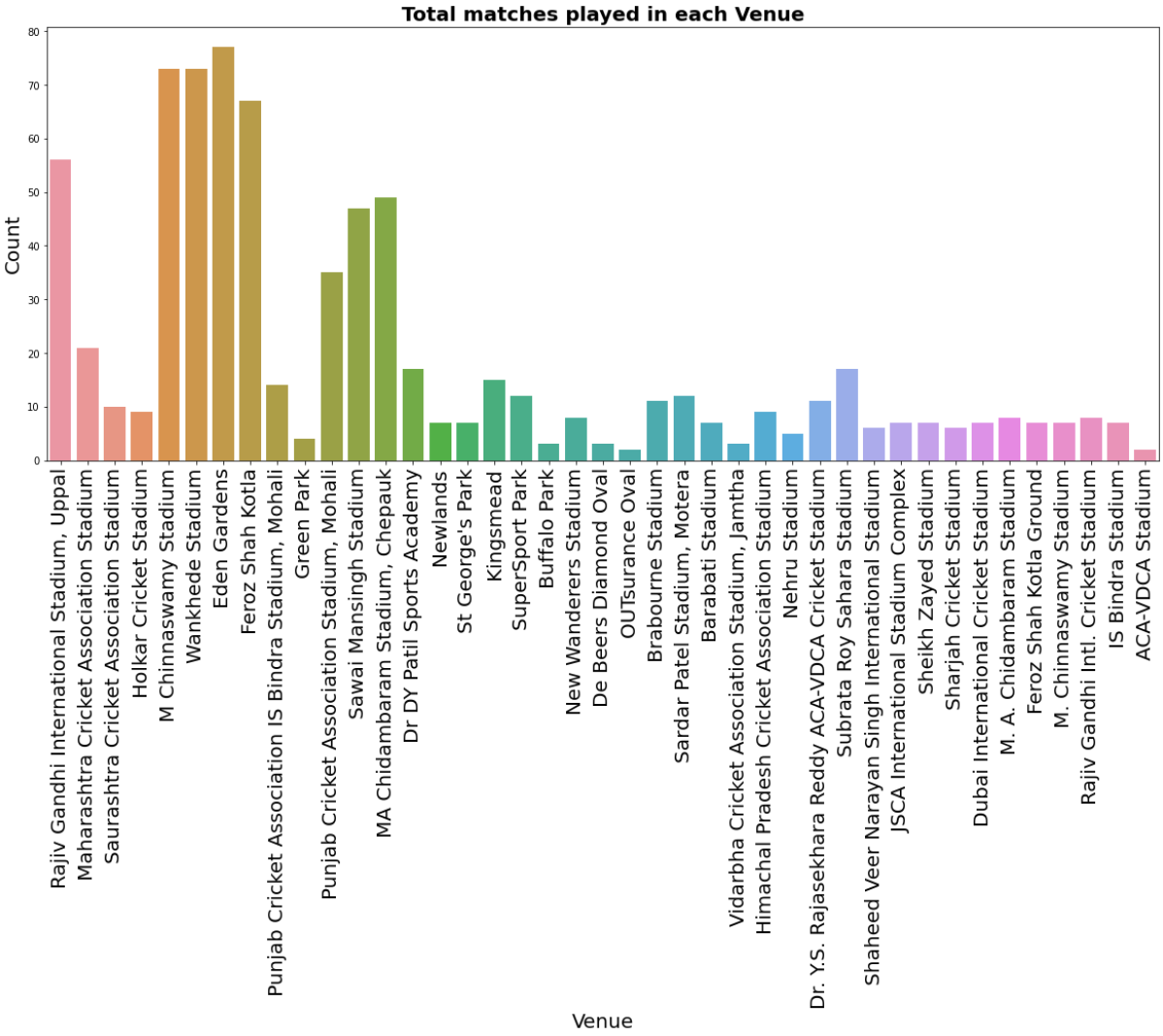
```
df.columns
```

```
Index(['match_id', 'city', 'date', 'team1', 'team2', 'toss_winner',
       'toss_decision', 'result', 'dl_applied', 'winner', 'win_by_runs',
       'win_by_wickets', 'player_of_match', 'venue', 'umpire1', 'umpire2',
       'umpire3', 'inning', 'batting_team', 'bowling_team', 'over', 'ball',
       'batsman', 'non_striker', 'bowler', 'is_super_over', 'wide_runs',
       'bye_runs', 'legbye_runs', 'noball_runs', 'penalty_runs',
       'batsman_runs', 'extra_runs', 'total_runs', 'player_dismissed',
       'dismissal_kind', 'fielder', 'Runs scored per match'],
      dtype='object')
```

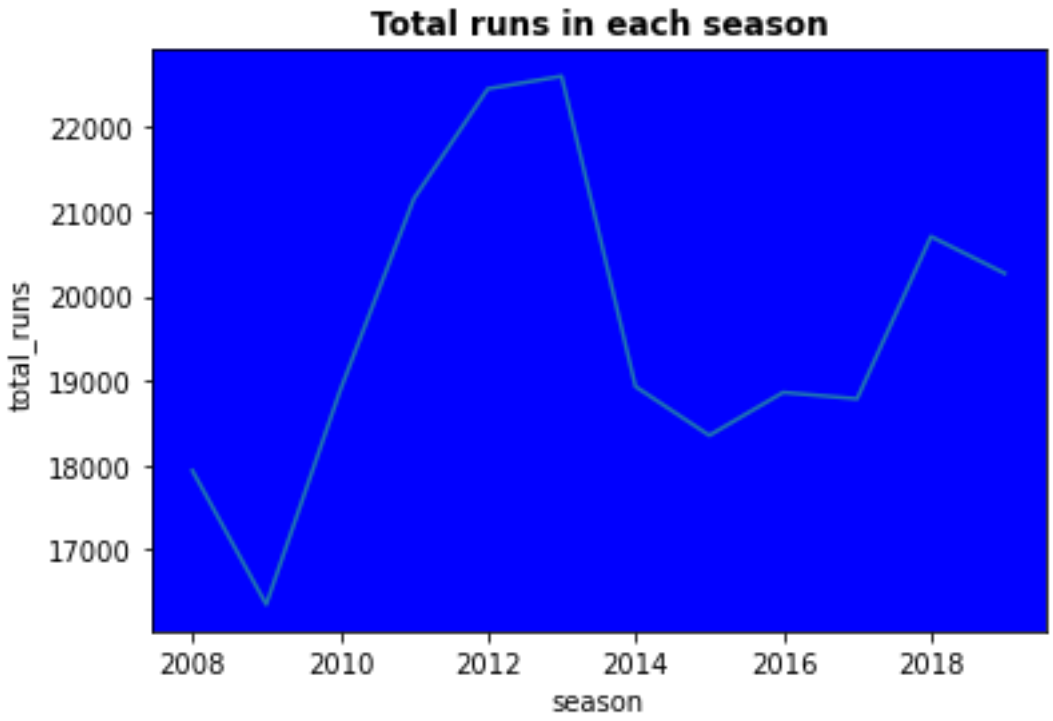
## no.of matches in particular years




## No.of Matches Played in Each Stadium


Total matches played in each Venue
Count
Venue
0
10
20
30
40
50
60
70
80
Rajiv Gandhi International Stadium, Uppal
Maharashtra Cricket Association Stadium
Saurashtra Cricket Association Stadium
Holkar Cricket Stadium
M Chinnaswamy Stadium
Wankhede Stadium
Eden Gardens
Feroz Shah Kotla
Punjab Cricket Association IS Bindra Stadium, Mohali
Green Park
Punjab Cricket Association Stadium, Mohali
Sawai Mansingh Stadium
MA Chidambaram Stadium, Chepauk
Dr DY Patil Sports Academy
Newlands
St George's Park
Kingsmead
SuperSport Park
Buffalo Park
New Wanderers Stadium
De Beers Diamond Oval
OUTsurance Oval
Brabourne Stadium
Sardar Patel Stadium, Motera
Barabati Stadium
Vidarbha Cricket Association Stadium, Jamtha
Himachal Pradesh Cricket Association Stadium
Nehru Stadium
Dr. Y.S. Rajasekhara Reddy ACA-VDCA Cricket Stadium
Subrata Roy Sahara Stadium
Shaheed Veer Narayan Singh International Stadium
JSCA International Stadium Complex
Sheikh Zayed Stadium
Sharjah Cricket Stadium
Dubai International Cricket Stadium
M. A. Chidambaram Stadium
Feroz Shah Kotla Ground
M. Chinnaswamy Stadium
Rajiv Gandhi Intl. Cricket Stadium
IS Bindra Stadium
ACA-VDCA Stadium

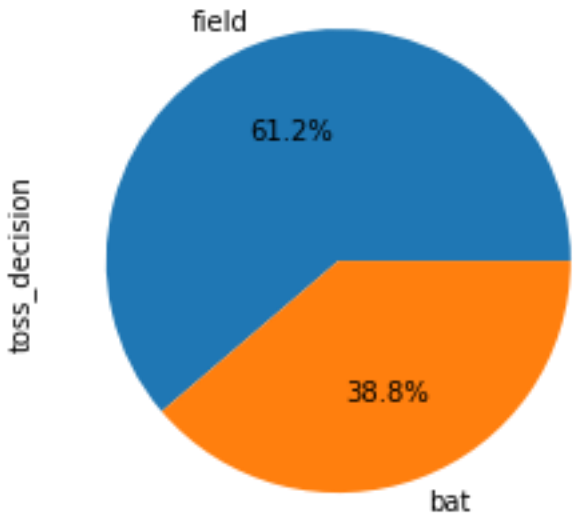
# Total run in each season



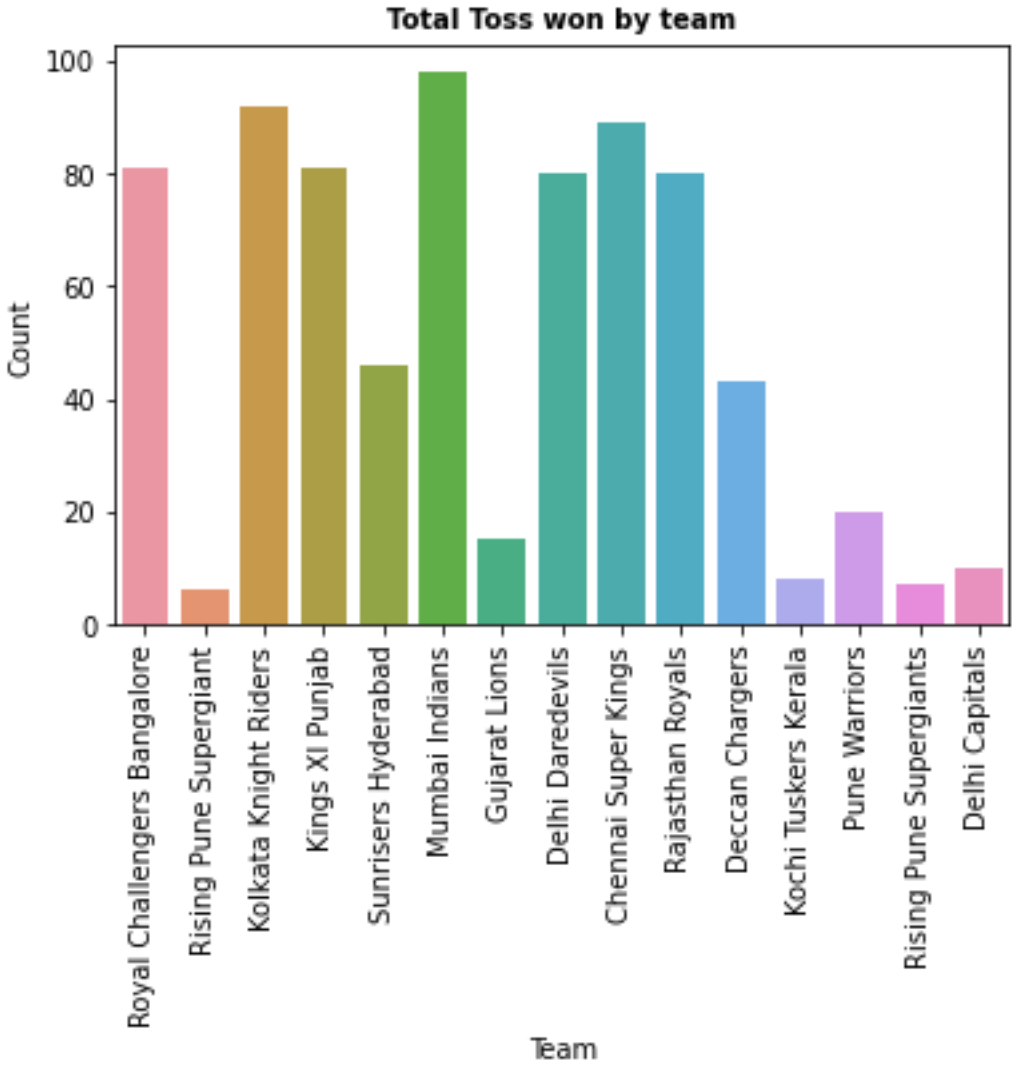
# Total run by each team  in total year

| | batting_team | total_runs |
|---|---|---|
| 0 | Mumbai Indians | 29809 |
| 1 | Royal Challengers Bangalore | 28126 |
| 2 | Kings XI Punjab | 27893 |
| 3 | Kolkata Knight Riders | 27419 |
| 4 | Chennai Super Kings | 26418 |
| 5 | Delhi Daredevils | 24388 |
| 6 | Rajasthan Royals | 22431 |
| 7 | Sunrisers Hyderabad | 17059 |
| 8 | Deccan Chargers | 11463 |
| 9 | Pune Warriors | 6358 |
| 10 | Gujarat Lions | 4862 |
| 11 | Delhi Capitals | 2630 |
| 12 | Rising Pune Supergiant | 2470 |
| 13 | Rising Pune Supergiants | 2063 |
| 14 | Kochi Tuskers Kerala | 1901 |

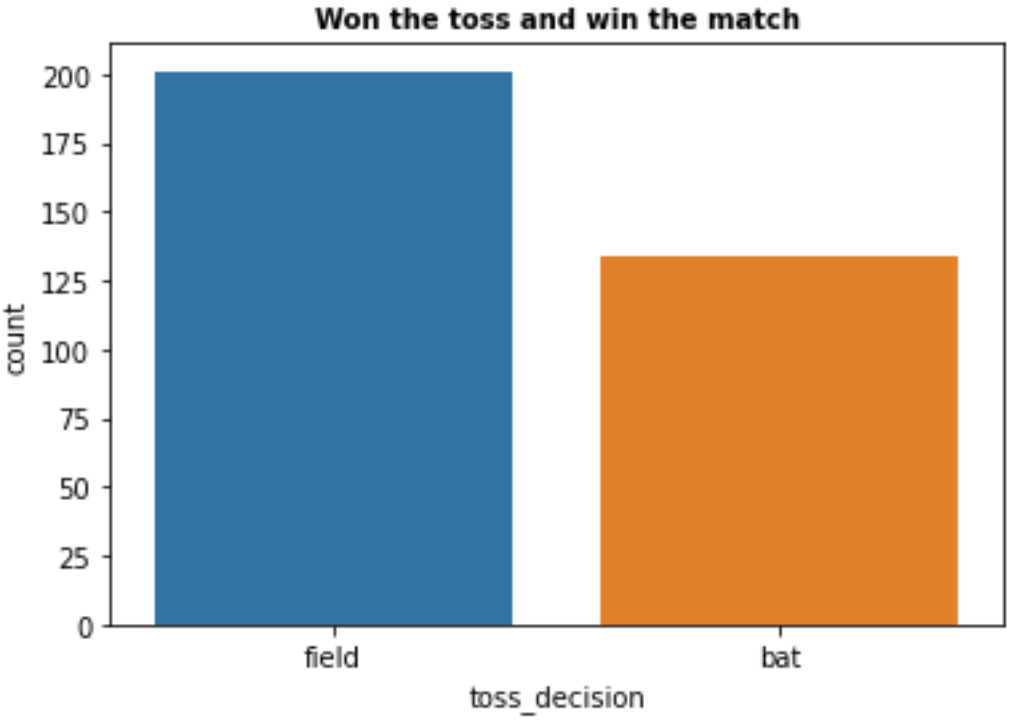
## Won the toss the choose



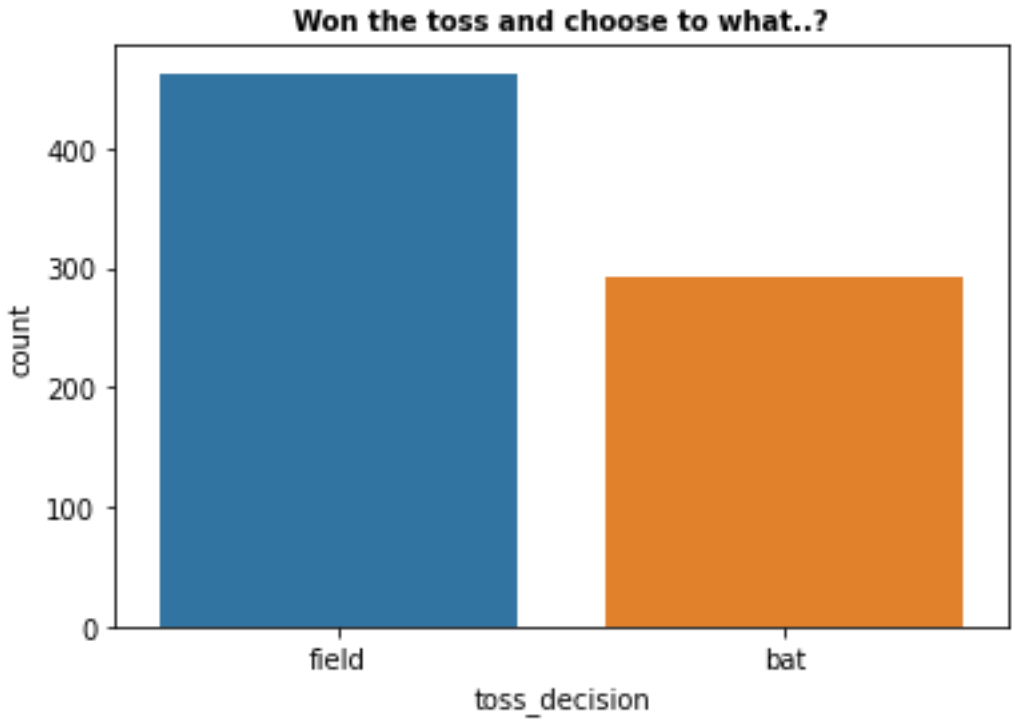

## Maximum toss won




## won the toss and win the match


Won the toss and win the match
200
175
150
125
100
75
50
25
0
count
field
bat
toss_decision

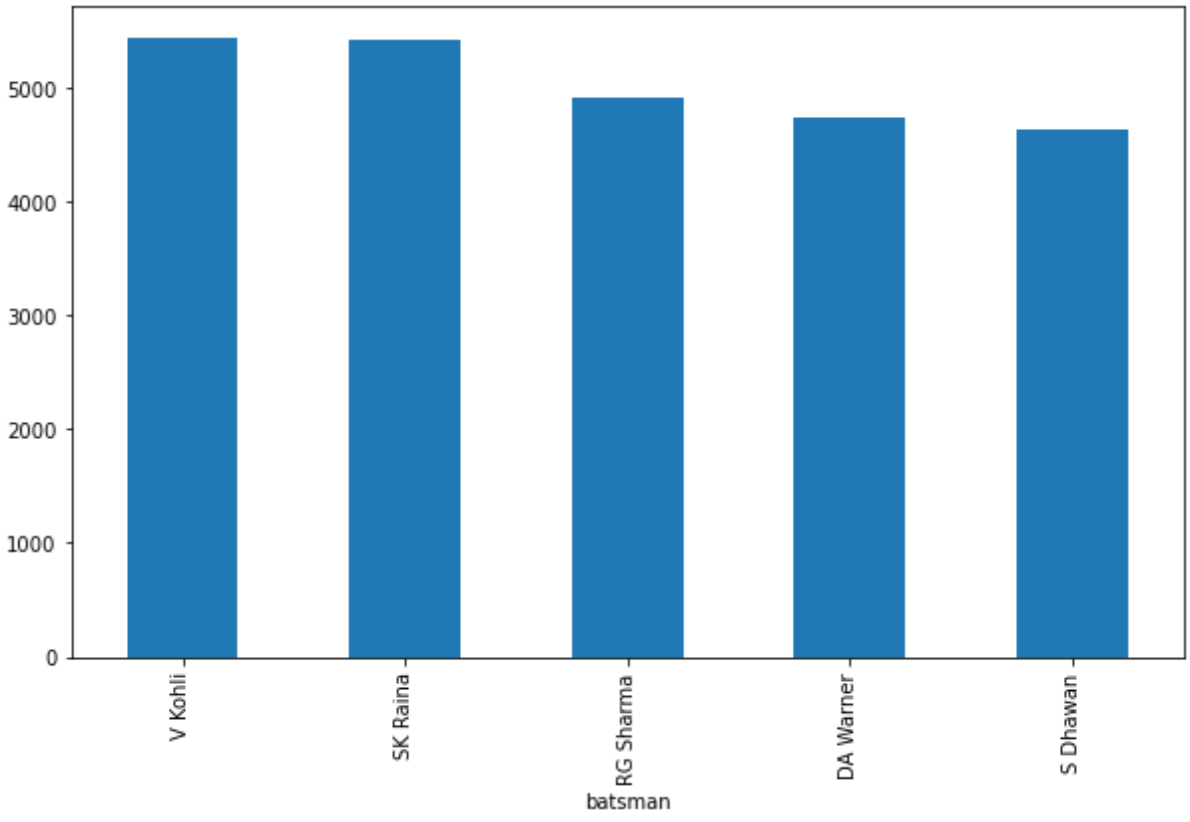
# Won the toss then deside



# betsman overview

## highest run total by betsman

```
batsman
V Kohli       5434
SK Raina      5415
RG Sharma     4914
DA Warner     4741
S Dhawan      4632
Name: batsman_runs, dtype: int64
```



# Total overvier in perticuler player

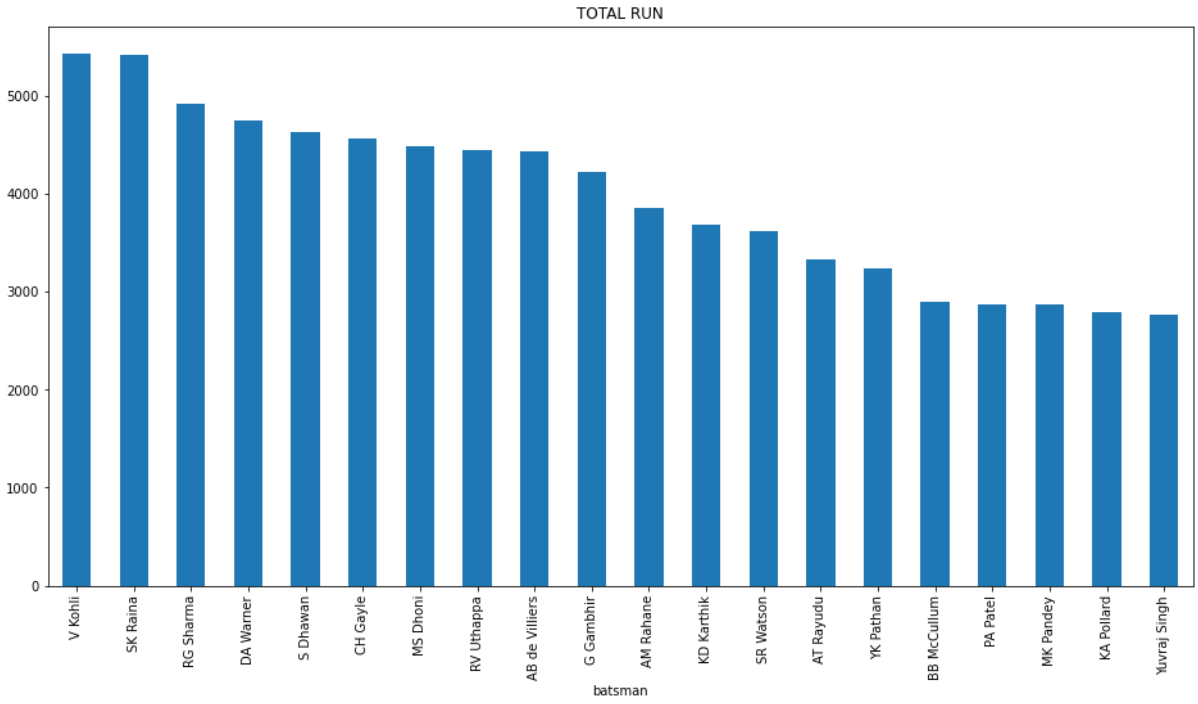
```
player = (dev['batsman']=='V Kohli')
kohli =dev[player]
```

```
def count(kohli,runs):
    return len(kohli[kohli['batsman_runs']==runs])*runs
```

```
print("Runs scored from 1's :",count(df_raina,1))
print("Runs scored from 2's :",count(df_raina,2))
print("Runs scored from 3's :",count(df_raina,3))
print("Runs scored from 4's :",count(df_raina,4))
print("Runs scored from 6's :",count(df_raina,6))
```

```
Runs scored from 1's : 1741
Runs scored from 2's : 586
Runs scored from 3's : 33
Runs scored from 4's : 1928
Runs scored from 6's : 1146
```

# Total run by betsman



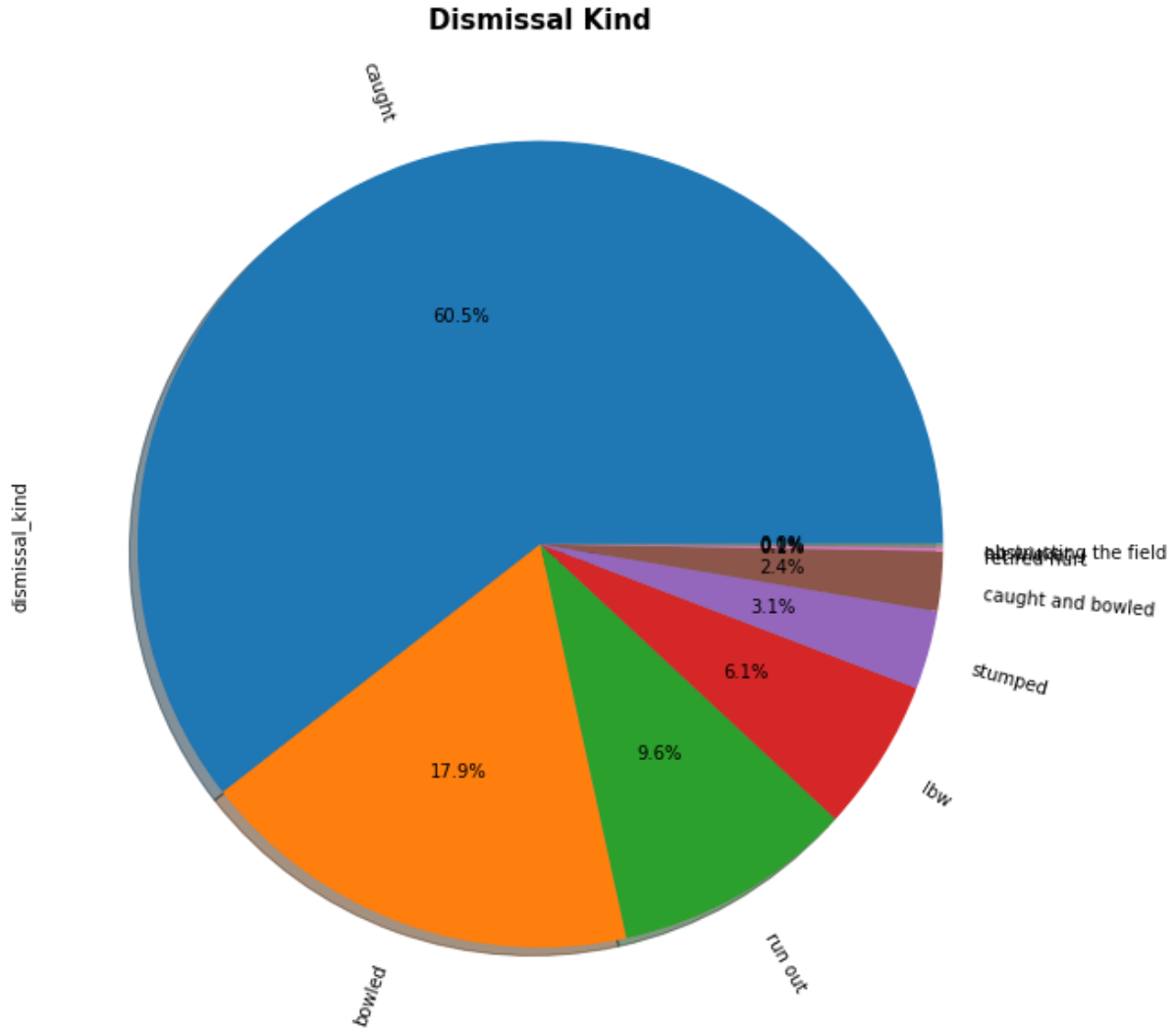
## Top player

| batsman | ball | batsman_runs | strike_rate |
|---|---|---|---|
| V Kohli | 4211 | 5434 | 129.042983 |
| SK Raina | 4044 | 5415 | 133.902077 |
| RG Sharma | 3816 | 4914 | 128.773585 |
| DA Warner | 3398 | 4741 | 139.523249 |
| S Dhawan | 3776 | 4632 | 122.669492 |
| CH Gayle | 3131 | 4560 | 145.640370 |
| MS Dhoni | 3318 | 4477 | 134.930681 |
| RV Uthappa | 3492 | 4446 | 127.319588 |
| AB de Villiers | 2977 | 4428 | 148.740343 |
| G Gambhir | 3524 | 4223 | 119.835414 |

## Each year maximum run made by player

```
batsman          season
V Kohli          2016      973
DA Warner        2016      848
KS Williamson    2018      747
MEK Hussey       2013      733
CH Gayle         2012      733
DA Warner        2019      727
CH Gayle         2013      720
RR Pant          2018      717
AB de Villiers   2016      687
KL Rahul         2018      678
Name: batsman_runs, dtype: int64
```

# Dismissal kind of player

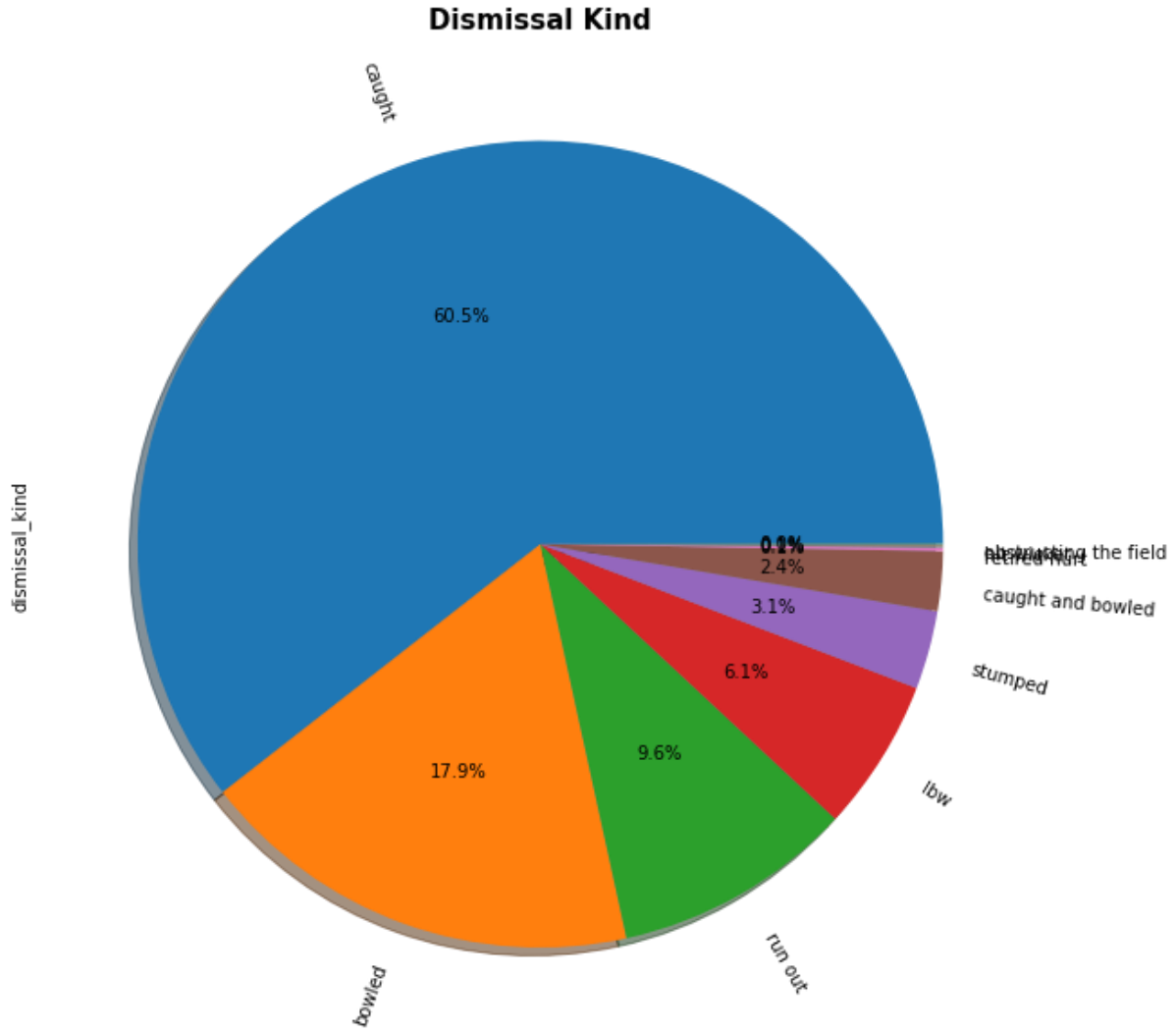


# BOLLING OVERVIEW

## Economy with maximum bolling

| bowler | batsman_runs | ball | economy |
|---|---|---|---|
| Harbhajan Singh | 3880 | 3451 | 6.745871 |
| A Mishra | 3727 | 3172 | 7.049811 |
| PP Chawla | 4022 | 3157 | 7.643966 |
| R Ashwin | 3224 | 3016 | 6.413793 |
| SL Malinga | 3218 | 2974 | 6.492266 |
| DJ Bravo | 3532 | 2711 | 7.817042 |
| B Kumar | 3067 | 2707 | 6.797931 |
| P Kumar | 3106 | 2637 | 7.067122 |
| UT Yadav | 3421 | 2605 | 7.879463 |
| SP Narine | 2825 | 2600 | 6.519231 |

# Player dismissed by bowler



# Conclusion

Hope you liked our project on IPL Data analysis and Visualization using Python. We just listed some basics to medium-advanced analysis over here, to give you an idea of how to use the data set. You can come up with your own data analysis of IPL data with Python libraries and projects.