

1. Loading data

In [2]:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
```

In [3]:

```
#read_ the csv file using read_csv command
deli = pd.read_csv("deliveries.csv")
matches = pd.read_csv("matches.csv")
```

2. Familiarizing with Data

In [4]:

```
#to show to 5 rows
deli.head(5)
```

Out[4]:

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler	is_super_over	...	bye_runs	legbye_runs	noball_runs	penalty
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
4	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	5	DA Warner	S Dhawan	TS Mills	0	...	0	0	0	0

5 rows × 21 columns

Anaysing Deliveries dataset

```
In [5]: #shape of dataframe  
deli.shape
```

```
Out[5]: (179078, 21)
```

```
In [6]: # to check the infromationkof the data set  
deli.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 179078 entries, 0 to 179077  
Data columns (total 21 columns):  
 #   Column           Non-Null Count  Dtype     
 ---    
 0   match_id         179078 non-null  int64    
 1   inning           179078 non-null  int64    
 2   batting_team     179078 non-null  object    
 3   bowling_team     179078 non-null  object    
 4   over             179078 non-null  int64    
 5   ball              179078 non-null  int64    
 6   batsman          179078 non-null  object    
 7   non_striker      179078 non-null  object    
 8   bowler            179078 non-null  object    
 9   is_super_over    179078 non-null  int64    
 10  wide_runs        179078 non-null  int64    
 11  bye_runs         179078 non-null  int64    
 12  legbye_runs      179078 non-null  int64    
 13  noball_runs      179078 non-null  int64    
 14  penalty_runs     179078 non-null  int64    
 15  batsman_runs     179078 non-null  int64    
 16  extra_runs       179078 non-null  int64    
 17  total_runs       179078 non-null  int64    
 18  player_dismissed 8834 non-null   object    
 19  dismissal_kind   8834 non-null   object    
 20  fielder           6448 non-null   object    
 dtypes: int64(13), object(8)  
 memory usage: 28.7+ MB
```

In [7]:

```
# to check all the null values
deli.isnull()
```

Out[7]:

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler	is_super_over	...	bye_runs	legbye_runs	noball_runs
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False
...
179073	False	False	False	False	False	False	False	False	False	False	...	False	False	False
179074	False	False	False	False	False	False	False	False	False	False	...	False	False	False
179075	False	False	False	False	False	False	False	False	False	False	...	False	False	False
179076	False	False	False	False	False	False	False	False	False	False	...	False	False	False
179077	False	False	False	False	False	False	False	False	False	False	...	False	False	False

179078 rows × 21 columns

◀	▶
---	---

In [8]:

```
# describe teh data sets
deli.describe()
```

Out[8]:

	match_id	inning	over	ball	is_super_over	wide_runs	bye_runs	legbye_runs	noball_runs	penalty
count	179078.000000	179078.000000	179078.000000	179078.000000	179078.000000	179078.000000	179078.000000	179078.000000	179078.000000	179078.000000
mean	1802.252957	1.482952	10.162488	3.615587	0.000452	0.036721	0.004936	0.021136	0.004183	0.00
std	3472.322805	0.502074	5.677684	1.806966	0.021263	0.251161	0.116480	0.194908	0.070492	0.0
min	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
25%	190.000000	1.000000	5.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00

	match_id	inning	over	ball	is_super_over	wide_runs	bye_runs	legbye_runs	noball_runs	penalty
50%	379.000000	1.000000	10.000000	4.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	567.000000	2.000000	15.000000	5.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	11415.000000	5.000000	20.000000	9.000000	1.000000	5.000000	4.000000	5.000000	5.000000	5.000000



In [9]:

```
deli.columns.unique()
```

Out[9]:

```
Index(['match_id', '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'],
      dtype='object')
```

Analysing Matches DataSet

In [10]:

```
#to show to 5 rows
matches.head(5)
```

Out[10]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied	winner	win_by_runs	win_by_wickets	player_
0	1	2017	Hyderabad	2017-04-05	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0	Sunrisers Hyderabad	35	0	Yuvra
1	2	2017	Pune	2017-04-06	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	0	Rising Pune Supergiant	0	7	S
2	3	2017	Rajkot	2017-04-07	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	0	Kolkata Knight Riders	0	10	

<u>id</u>	<u>season</u>	<u>city</u>	<u>date</u>	<u>team1</u>	<u>team2</u>	<u>toss_winner</u>	<u>toss_decision</u>	<u>result</u>	<u>dl_applied</u>	<u>winner</u>	<u>win_by_runs</u>	<u>win_by_wickets</u>	<u>player_of_match</u>	
3	4	2017	Indore	2017-04-08	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	normal	0	Kings XI Punjab	0	6	G
4	5	2017	Bangalore	2017-04-08	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	normal	0	Royal Challengers Bangalore	15	0	K



In [11]:

```
# shape of the dataframe
matches.shape
```

Out[11]:

(756, 18)

In [12]:

```
#information of the dataset
matches.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

```

In [13]:

```
# to check the null values
matches.isnull()
```

Out[13]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied	winner	win_by_runs	win_by_wickets	player_of_match	venue
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...
751	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
752	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
753	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
754	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
755	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False

756 rows × 18 columns



In [14]:

```
#description of the dataset
matches.describe()
```

Out[14]:

	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

	id	season	dl_applied	win_by_runs	win_by_wickets
std	3464.478148	3.366895	0.156630	23.471144	3.387963
min	1.000000	2008.000000	0.000000	0.000000	0.000000
25%	189.750000	2011.000000	0.000000	0.000000	0.000000
50%	378.500000	2013.000000	0.000000	0.000000	4.000000
75%	567.250000	2016.000000	0.000000	19.000000	6.000000
max	11415.000000	2019.000000	1.000000	146.000000	10.000000

In [15]:

`matches.head(2)`

Out[15]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied	winner	win_by_runs	win_by_wickets	player_o
0	1	2017	Hyderabad	2017-04-05	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0	Sunrisers Hyderabad	35	0	Yuv
1	2	2017	Pune	2017-04-06	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	0	Rising Pune Supergiant	0	7	SF

◀ ▶

In [16]:

```
#fill null values
matches = matches.fillna({'city': "no city", 'winner':'no data Available', 'player_of_match':'no data', 'umpire1':np.NaN, 'umpire2':np.NaN})
```

In [17]:

`matches.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            756 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         756 non-null   object
11 win_by_runs    756 non-null   int64
12 win_by_wickets 756 non-null   int64
13 player_of_match 756 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

In [18]:
matches = matches.drop_duplicates()

In [19]:
matches.shape

Out[19]:
(756, 18)

In [20]:
matches.head(2)

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied	winner	win_by_runs	win_by_wickets	player_o
0	1	2017	Hyderabad	2017-04-05	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0	Sunrisers Hyderabad	35	0	Yuv
1	2	2017	Pune	2017-04-06	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	0	Rising Pune Supergiant	0	7	SF

3. Visualizing the data

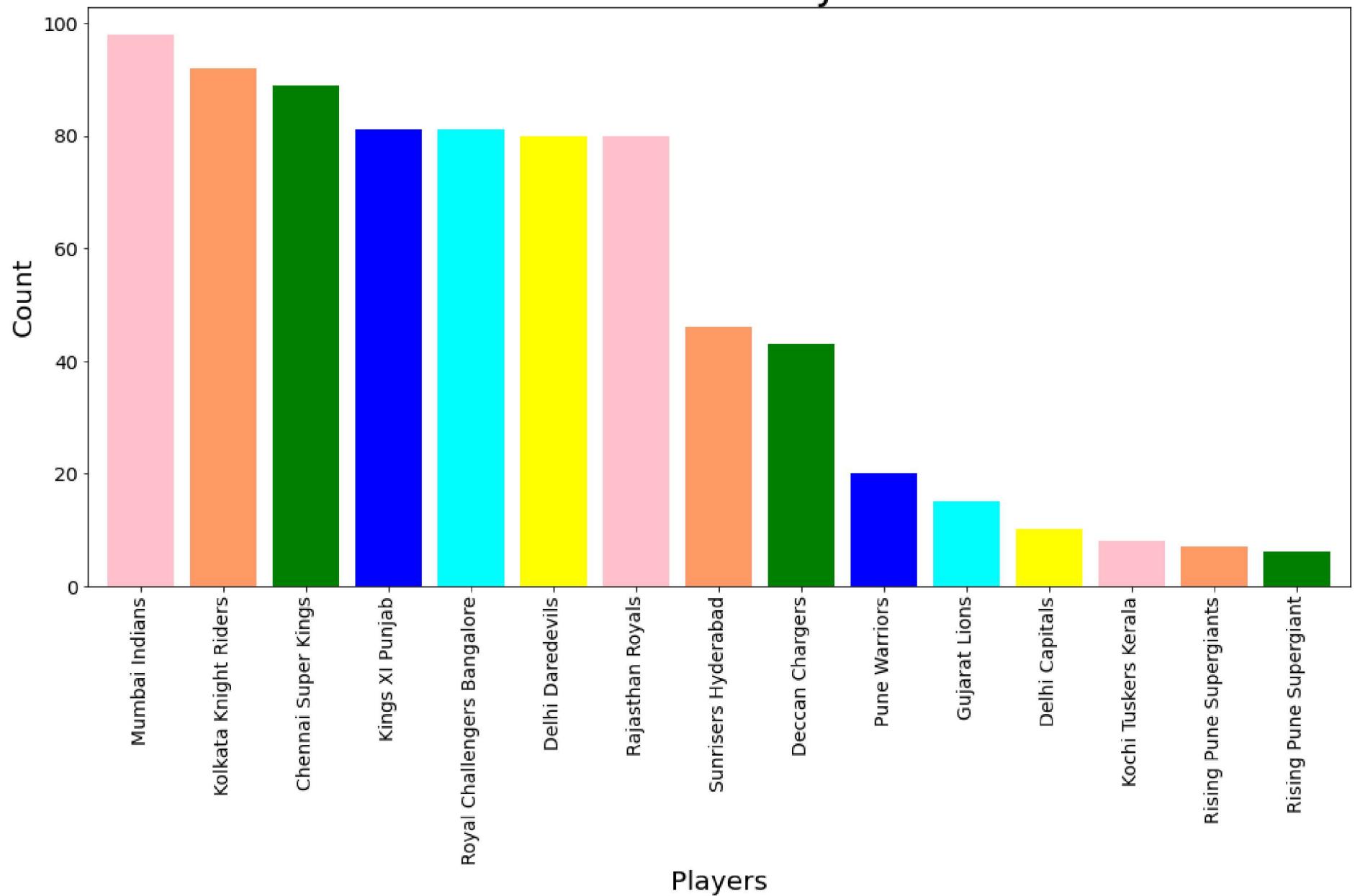
```
In [21]: matches.head(2)
```

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied	winner	win_by_runs	win_by_wickets	player_o
0	1	2017	Hyderabad	2017-04-05	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0	Sunrisers Hyderabad	35	0	Yuv
1	2	2017	Pune	2017-04-06	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	0	Rising Pune Supergiant	0	7	SF

Finding Top Teams and Players

```
In [22]: a = matches.groupby('toss_winner')[ "toss_winner"].count()
a1 = a.sort_values(ascending = False)
a1.plot.bar(figsize=(17,8),fontsize = 14,width = 0.8,color=['pink', '#ff9966', 'green', 'blue', 'cyan','Yellow'])
plt.title("Toss win Count by Team",fontsize = 30)
plt.xlabel("Players",fontsize = 20)
plt.ylabel("Count",fontsize = 20)
plt.show()
```

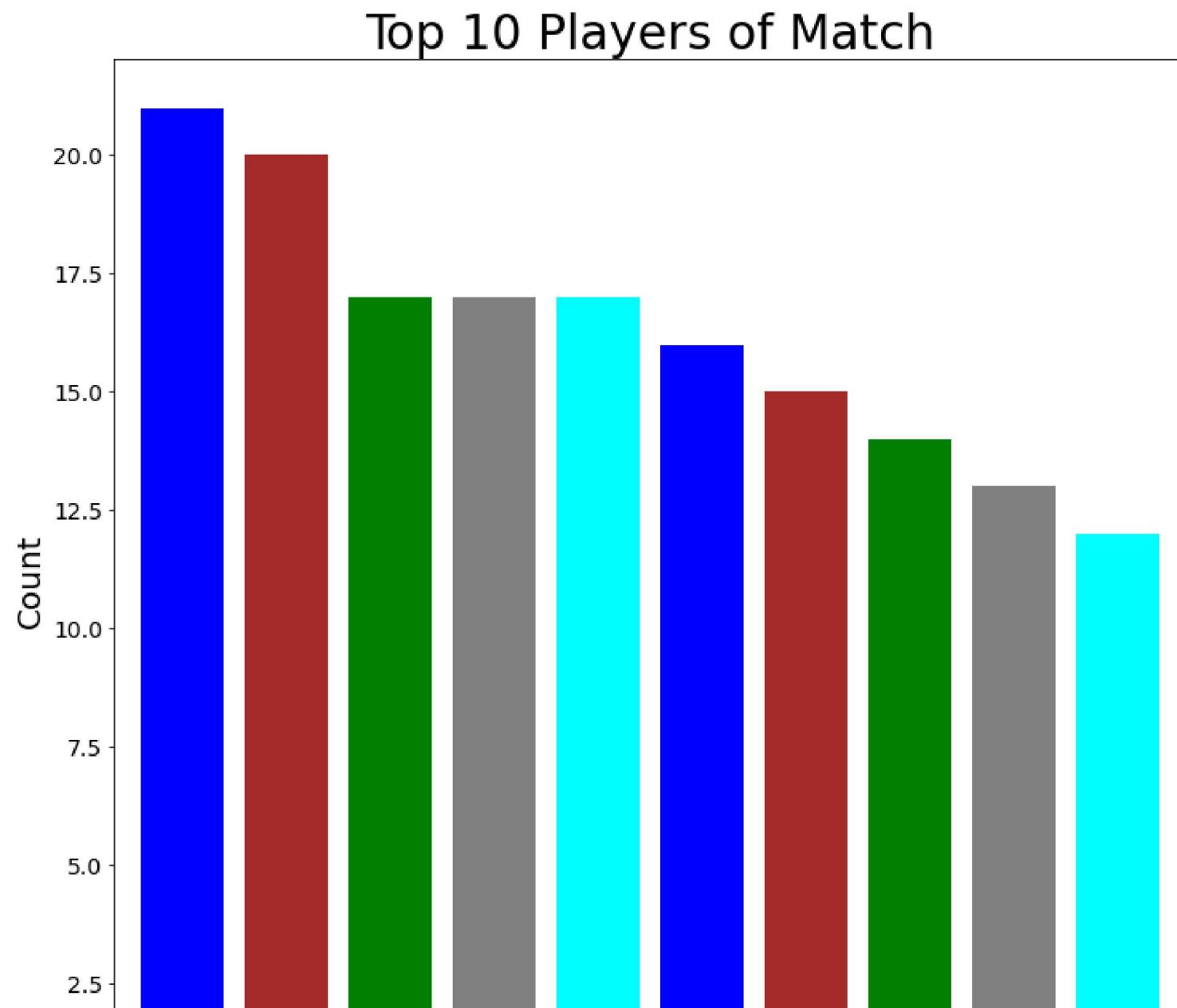
Toss win Count by Team

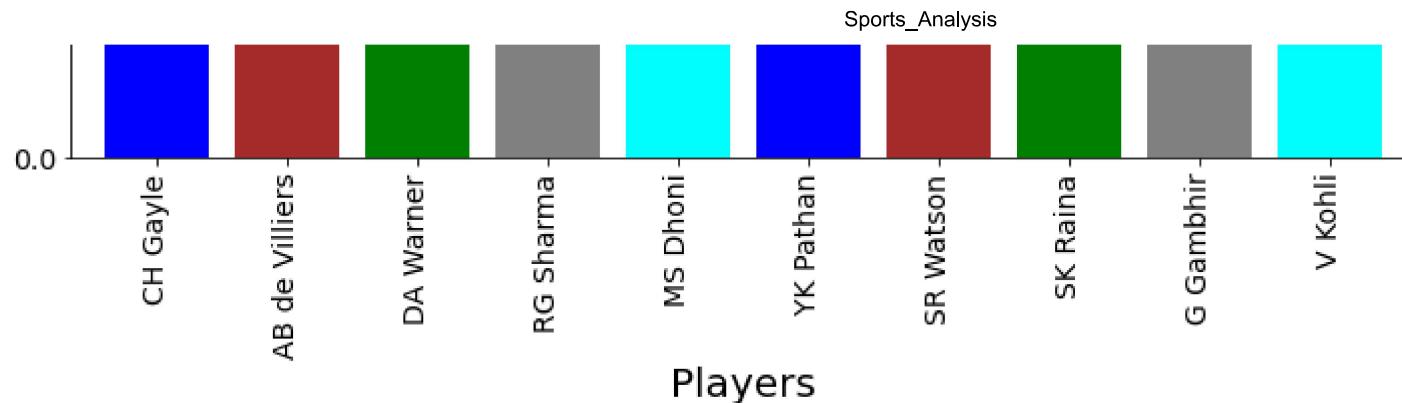


In [23]:

```
b = matches.groupby('player_of_match')['player_of_match'].count().sort_values(ascending = False).head(10)
b.plot.bar(figsize=(12,12), fontsize = 14, width = 0.8, color=['blue', 'brown', 'green', 'gray', 'cyan'])
```

```
plt.title("Top 10 Players of Match", fontsize = 30)
plt.xlabel("Players", fontsize = 20)
plt.ylabel("Count", fontsize = 20)
plt.show()
```





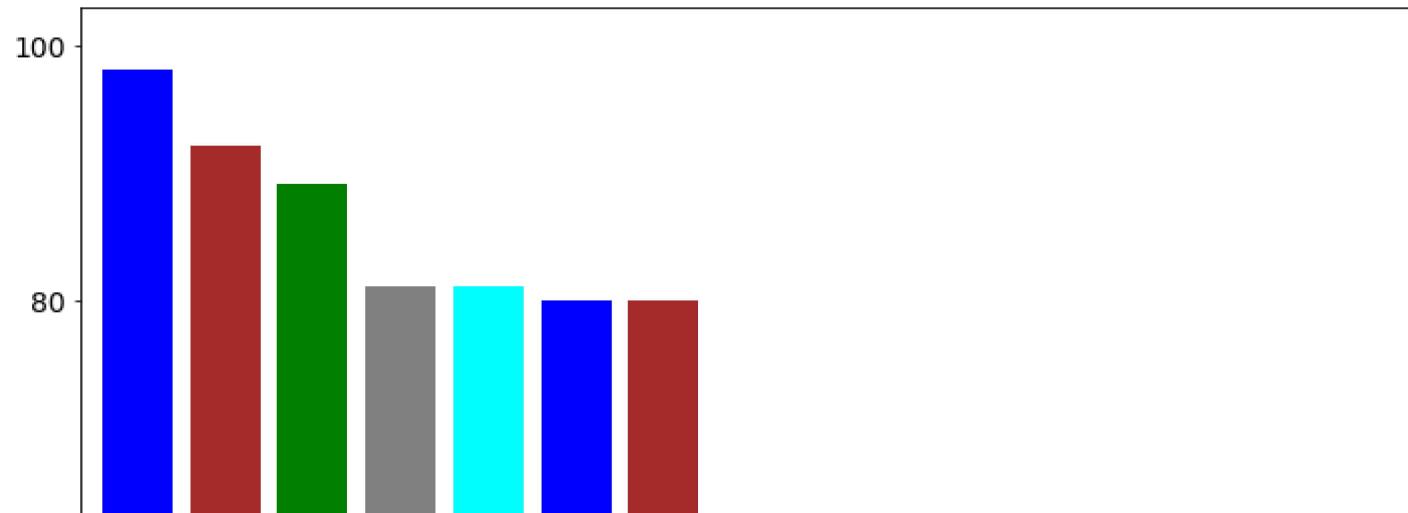
Finding the factors Affect the Victory

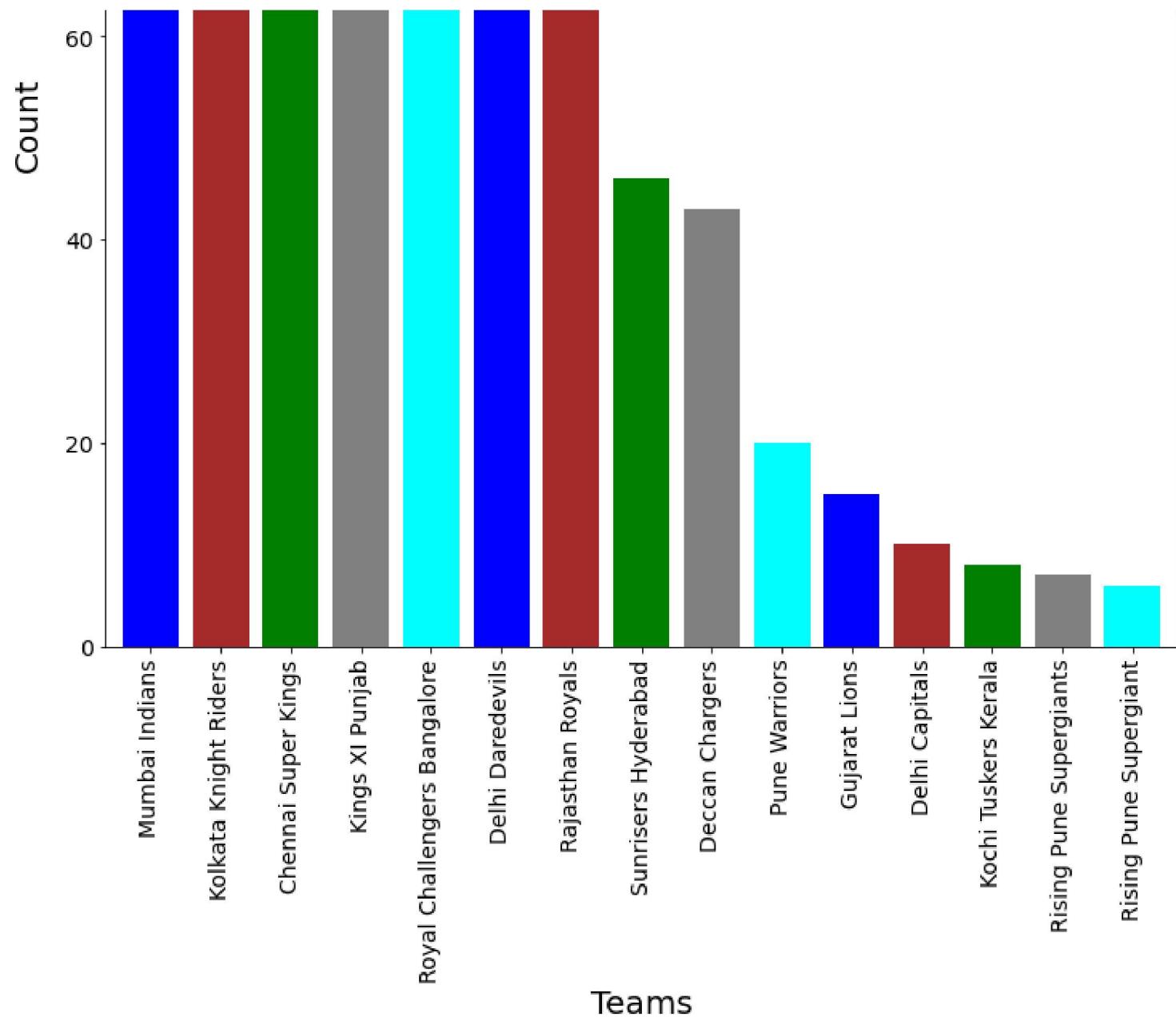
In []:

In [26]:

```
c = matches.groupby('toss_winner')[“toss_winner”].count().sort_values(ascending = False)
c.plot.bar(figsize=(12,12),fontsize = 14,width = 0.8,color=[‘blue’, ‘brown’, ‘green’, ‘gray’, ‘cyan’])
plt.title(“Team that won the toss”,fontsize = 30)
plt.xlabel(“Teams”,fontsize = 20)
plt.ylabel(“Count”,fontsize = 20)
plt.show()
```

Team that won the toss





In [27]:

```
bat_f = matches[matches['toss_decision'] == "bat"]
bat_f[bat_f["winner"] == bat_f["winner"].max()]
```

```
# bat_f.groupby("winner")["winner"].count()
# bat_f
```

Out[27]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied	winner	win_by_runs	win_by_wickets	player_of_match
300	301	2011	Delhi	2011-05-21	Delhi Daredevils	Pune Warriors	Delhi Daredevils	bat	no result	0	no data Available	0	0	no data

4.Data Analysis:

4.1 Merging the two Datasets into a new Dataset and Reading it(join on match-id)

In [28]:

```
merge = pd.merge(deli, matches, left_on = 'match_id', right_on = 'id', how ="left")
merge= merge.reset_index()
merge.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 179078 entries, 0 to 179077
Data columns (total 40 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   index            179078 non-null   int64  
 1   match_id         179078 non-null   int64  
 2   inning           179078 non-null   int64  
 3   batting_team     179078 non-null   object  
 4   bowling_team     179078 non-null   object  
 5   over             179078 non-null   int64  
 6   ball              179078 non-null   int64  
 7   batsman          179078 non-null   object  
 8   non_striker      179078 non-null   object  
 9   bowler            179078 non-null   object  
 10  is_super_over    179078 non-null   int64  
 11  wide_runs        179078 non-null   int64  
 12  bye_runs         179078 non-null   int64  
 13  legbye_runs      179078 non-null   int64  
 14  noball_runs      179078 non-null   int64  
 15  penalty_runs     179078 non-null   int64  
 16  batsman_runs     179078 non-null   int64
```

```
17 extra_runs      179078 non-null  int64
18 total_runs      179078 non-null  int64
19 player_dismissed 8834 non-null   object
20 dismissal_kind  8834 non-null   object
21 fielder         6448 non-null   object
22 id              179078 non-null  int64
23 season          179078 non-null  int64
24 city            179078 non-null  object
25 date            179078 non-null  object
26 team1           179078 non-null  object
27 team2           179078 non-null  object
28 toss_winner      179078 non-null  object
29 toss_decision    179078 non-null  object
30 result           179078 non-null  object
31 dl_applied       179078 non-null  int64
32 winner           179078 non-null  object
33 win_by_runs      179078 non-null  int64
34 win_by_wickets   179078 non-null  int64
35 player_of_match  179078 non-null  object
36 venue            179078 non-null  object
37 umpire1          178578 non-null  object
38 umpire2          178578 non-null  object
39 umpire3          28366 non-null   object
dtypes: int64(19), object(21)
memory usage: 54.7+ MB
```

In [29]: `merge.shape`

Out[29]: `(179078, 40)`

In [30]: `merge.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 179078 entries, 0 to 179077
Data columns (total 40 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   index            179078 non-null  int64  
 1   match_id         179078 non-null  int64  
 2   inning           179078 non-null  int64  
 3   batting_team     179078 non-null  object 
```

```

4  bowling_team      179078 non-null  object
5  over              179078 non-null  int64
6  ball              179078 non-null  int64
7  batsman           179078 non-null  object
8  non_striker       179078 non-null  object
9  bowler             179078 non-null  object
10 is_super_over     179078 non-null  int64
11 wide_runs         179078 non-null  int64
12 bye_runs          179078 non-null  int64
13 legbye_runs       179078 non-null  int64
14 noball_runs        179078 non-null  int64
15 penalty_runs       179078 non-null  int64
16 batsman_runs       179078 non-null  int64
17 extra_runs         179078 non-null  int64
18 total_runs         179078 non-null  int64
19 player_dismissed  8834 non-null   object
20 dismissal_kind     8834 non-null   object
21 fielder            6448 non-null   object
22 id                 179078 non-null  int64
23 season             179078 non-null  int64
24 city               179078 non-null  object
25 date               179078 non-null  object
26 team1              179078 non-null  object
27 team2              179078 non-null  object
28 toss_winner         179078 non-null  object
29 toss_decision       179078 non-null  object
30 result              179078 non-null  object
31 dl_applied          179078 non-null  int64
32 winner              179078 non-null  object
33 win_by_runs         179078 non-null  int64
34 win_by_wickets      179078 non-null  int64
35 player_of_match     179078 non-null  object
36 venue               179078 non-null  object
37 umpire1             178578 non-null  object
38 umpire2             178578 non-null  object
39 umpire3             28366 non-null   object

```

dtypes: int64(19), object(21)

memory usage: 54.7+ MB

In [31]:

```
merge.head(5)
```

Out[31]:

	index	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler	...	result	dl_applied	winner	win_by_runs	win_l
--	-------	----------	--------	--------------	--------------	------	------	---------	-------------	--------	-----	--------	------------	--------	-------------	-------

index	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler	...	result	dl_applied	winner	win_by_runs	win_l
0	0	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	1	DA Warner	S Dhawan	TS Mills	...	normal	0	Sunrisers Hyderabad	35
1	1	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	2	DA Warner	S Dhawan	TS Mills	...	normal	0	Sunrisers Hyderabad	35
2	2	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	3	DA Warner	S Dhawan	TS Mills	...	normal	0	Sunrisers Hyderabad	35
3	3	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	4	DA Warner	S Dhawan	TS Mills	...	normal	0	Sunrisers Hyderabad	35
4	4	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	5	DA Warner	S Dhawan	TS Mills	...	normal	0	Sunrisers Hyderabad	35

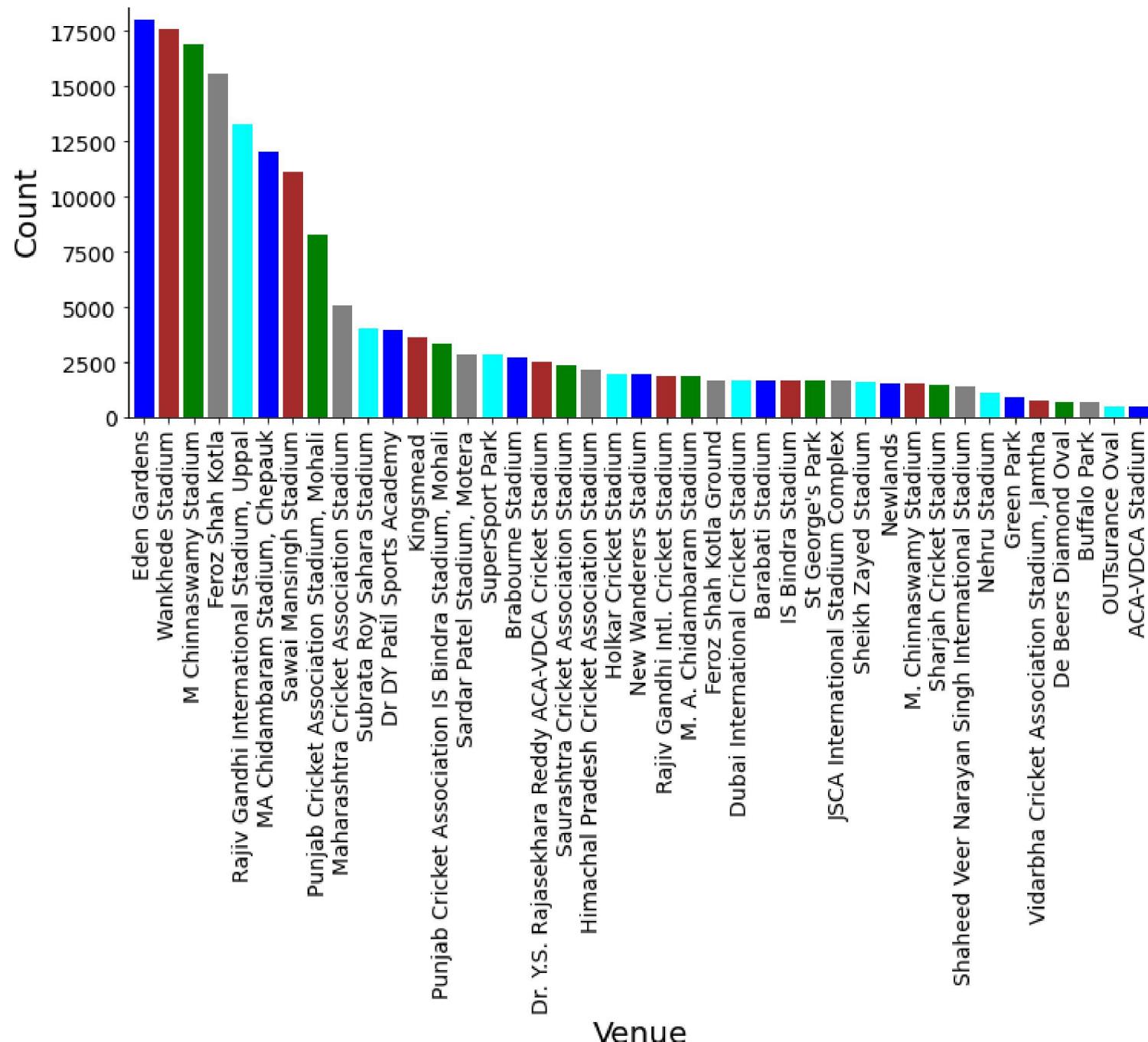
5 rows × 40 columns

Number of Matches Played in Each Stadium

In [32]:

```
m1 = merge.groupby('venue')['venue'].count().sort_values(ascending = False)
m1.plot.bar(figsize=(12,5), fontsize = 14, width = 0.8, color=['blue', 'brown', 'green', 'gray', 'cyan'])
plt.title("Number of Match in each Stadium", fontsize = 30)
# plt.tick_params(axis=)
plt.xlabel("Venue", fontsize = 20)
plt.ylabel("Count", fontsize = 20)
plt.show()
```

Number of Match in each Stadium



```
In [33]: merge.columns
```

```
Out[33]: Index(['index', 'match_id', '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', 'id', 'season', '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'],
  dtype='object')
```

```
In [34]: merge = merge.drop_duplicates()
merge.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 179078 entries, 0 to 179077
Data columns (total 40 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   index            179078 non-null   int64  
 1   match_id         179078 non-null   int64  
 2   inning            179078 non-null   int64  
 3   batting_team     179078 non-null   object  
 4   bowling_team     179078 non-null   object  
 5   over              179078 non-null   int64  
 6   ball              179078 non-null   int64  
 7   batsman           179078 non-null   object  
 8   non_striker      179078 non-null   object  
 9   bowler             179078 non-null   object  
 10  is_super_over    179078 non-null   int64  
 11  wide_runs         179078 non-null   int64  
 12  bye_runs          179078 non-null   int64  
 13  legbye_runs       179078 non-null   int64  
 14  noball_runs       179078 non-null   int64  
 15  penalty_runs      179078 non-null   int64  
 16  batsman_runs      179078 non-null   int64  
 17  extra_runs        179078 non-null   int64  
 18  total_runs        179078 non-null   int64  
 19  player_dismissed 8834 non-null    object  
 20  dismissal_kind    8834 non-null    object  
 21  fielder            6448 non-null    object  
 22  id                 179078 non-null   int64
```

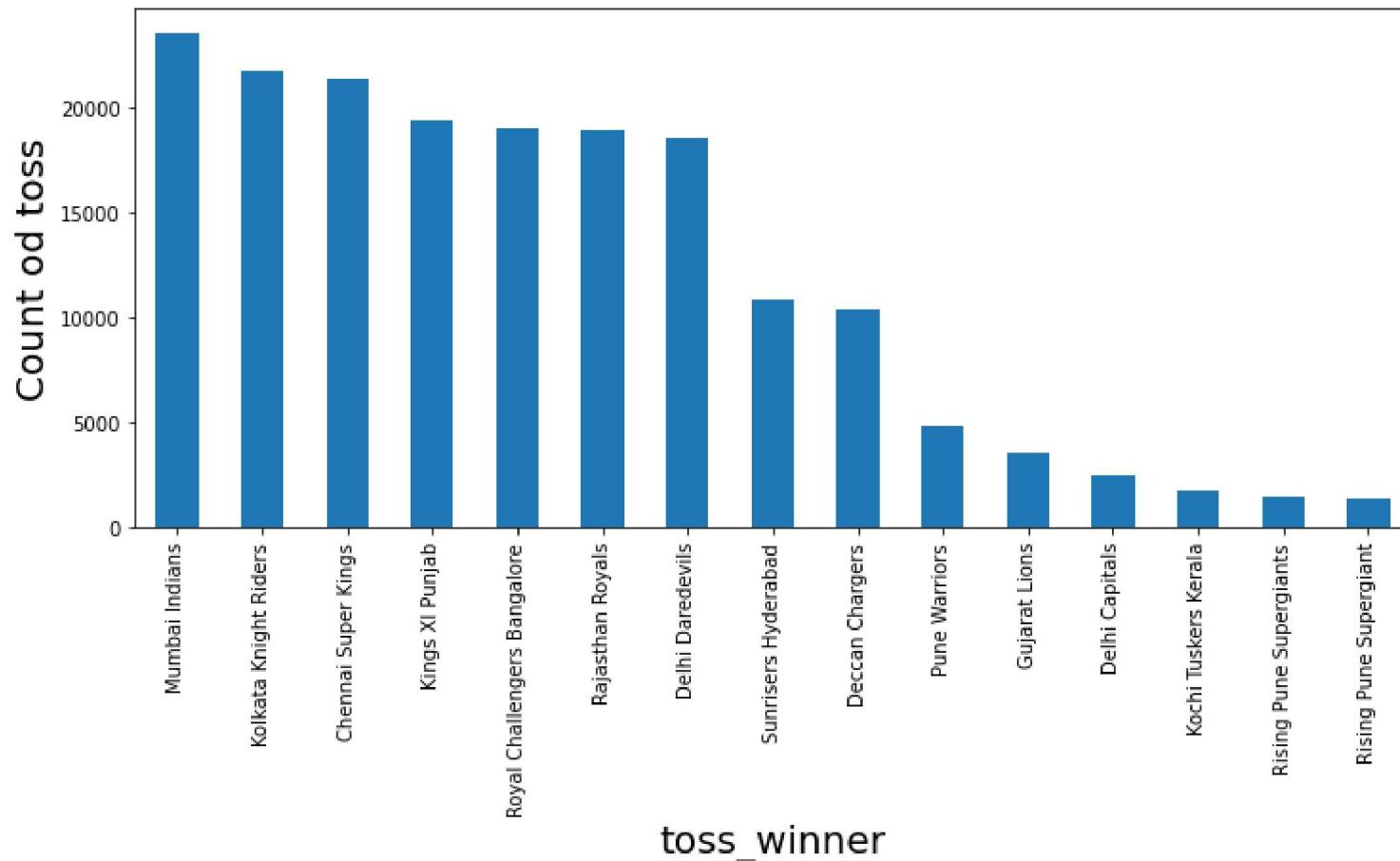
```
23 season          179078 non-null  int64
24 city            179078 non-null  object
25 date            179078 non-null  object
26 team1           179078 non-null  object
27 team2           179078 non-null  object
28 toss_winner      179078 non-null  object
29 toss_decision    179078 non-null  object
30 result           179078 non-null  object
31 dl_applied       179078 non-null  int64
32 winner           179078 non-null  object
33 win_by_runs      179078 non-null  int64
34 win_by_wickets   179078 non-null  int64
35 player_of_match  179078 non-null  object
36 venue            179078 non-null  object
37 umpire1          178578 non-null  object
38 umpire2          178578 non-null  object
39 umpire3          28366 non-null   object
dtypes: int64(19), object(21)
memory usage: 56.0+ MB
```

Maximum Toss Won

In [35]:

```
a = merge.groupby("toss_winner")["toss_winner"].count().sort_values(ascending = False)
a.plot.bar(figsize = (12,5))
plt.title("Maximum Toss Won", fontsize = 40)
plt.xlabel("toss_winner", fontsize = 20)
plt.ylabel("Count od toss", fontsize = 20)
plt.show()
```

Maximum Toss Won



In [36]:

```
m.toss_winner
```

```
NameError
```

```
Traceback (most recent call last)
```

```
~\AppData\Local\Temp\ipykernel_22192\3145332987.py in <module>
```

```
----> 1 m.toss_winner
```

```
NameError: name 'm' is not defined
```

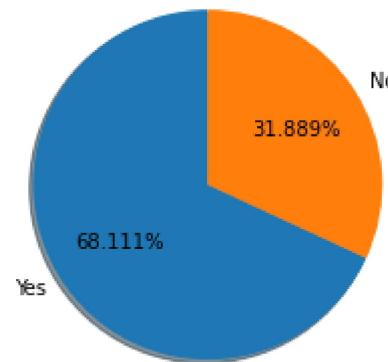
In []:

```
m.columns
```

```
In [ ]: matches.info()
```

```
In [47]: #obtain toss winners in the match
winner = matches[matches['toss_winner']==matches['winner']]
lab=['Yes', 'No']
plt.pie([len(winner),(577-len(winner))],labels=lab, autopct='%.1f%%', shadow=True, startangle=90)
plt.title("Teams who had won Toss and won the match")
plt.show()
```

Teams who had won Toss and won the match



Decidning whether to Bat or Field after Winning the Toss

```
In [ ]: toss_decision = matches.groupby("toss_decision")["toss_decision"].count().sort_values(ascending = False)
toss_decision.plot.bar()
plt.show()
```

Relation between Winning toss and victory

```
In [ ]: matches[matches["toss_decision"] == "toss_decision"]
```

```
In [ ]: win = matches[['winner', 'toss_winner']].count()
```

```
win.plot.bar()  
# toss_decision.plot.bar()  
  
# k.plot.bar()
```

```
In [ ]:  
batsman_overview = deli.groupby("batsman")[[ "ball", "batsman_runs" ]].agg({ "ball": "count", "batsman_runs": "sum" }).sort_values(by =  
batsman_overview[ "batting_strike_rate" ] = ((batsman_overview[ "batsman_runs" ]/batsman_overview[ "ball" ])*100)  
batsman_overview.head(20)
```

```
In [ ]:  
merge.columns
```

```
In [ ]:  
deli.groupby([ "batsman_runs", "batsman" ])["batsman_runs"].count().sort_values(ascending = False).T  
# batsman_overview = deli.groupby([ "batsman" ])[[ "ball", "batsman_runs" ]].agg({ "ball": "count", "batsman_runs": "sum" }).sort_values(b  
# batsman_overview[ "batting_strike_rate" ] = ((batsman_overview[ "batsman_runs" ]/batsman_overview[ "ball" ])*100)  
# batsman_overview.head(20)
```

Batsmen overview

Total runs by each batsman

```
In [ ]:  
batsman_overview = deli.groupby("batsman")["batsman_runs"].sum().sort_values(ascending = False).head(20)  
batsman_overview.plot.bar(figsize=(15,5), fontsize = 14, width = 0.5 )  
plt.title("Total Runs", fontsize = 30)  
# plt.tick_params(axis=  
plt.xlabel("Batsman", fontsize = 20)  
plt.ylabel("Runs", fontsize = 20)  
plt.show()
```

```
In [ ]:  
matches.head()
```

```
In [ ]:
```

```
In [ ]:
```

In []:

Each batsman strike rate

In [37]:

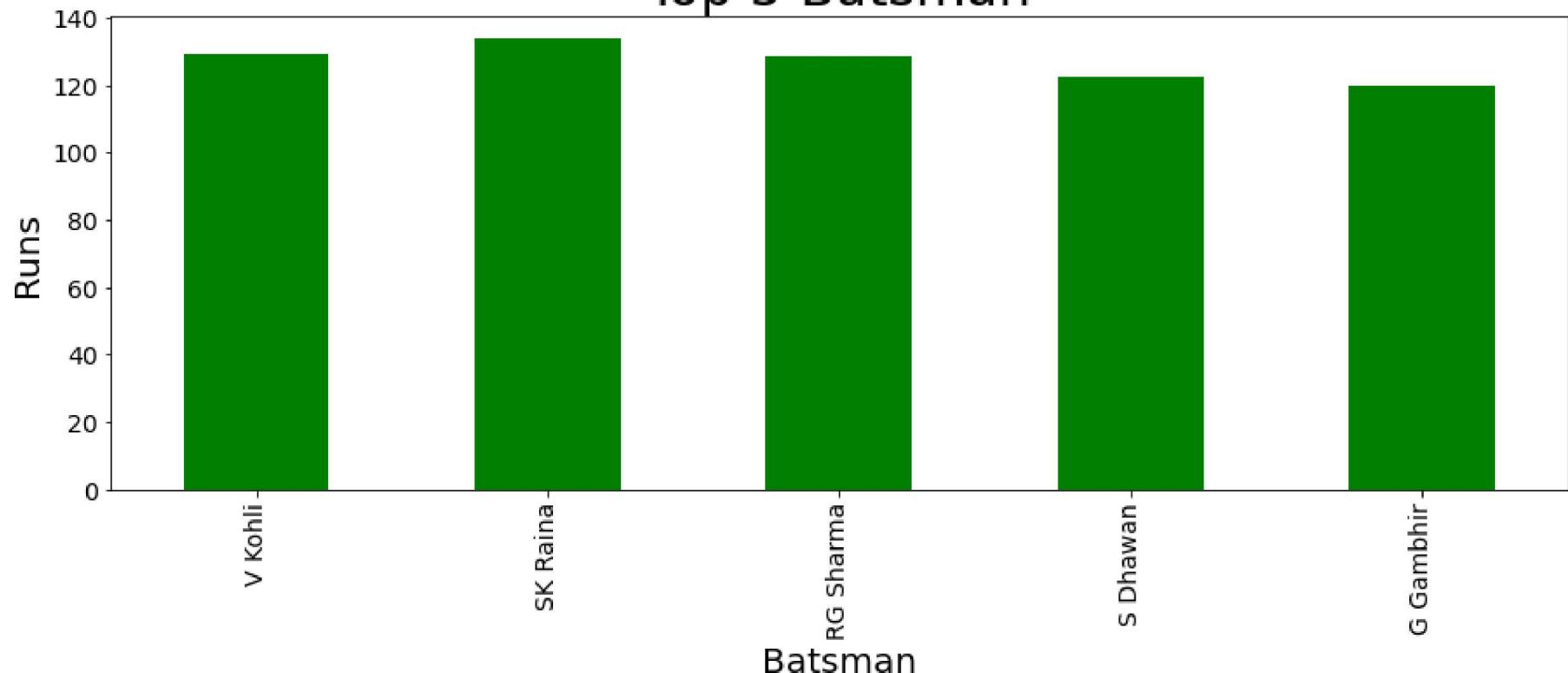
```
batsman_overview = deli.groupby("batsman")[["ball","batsman_runs"]].agg({"ball":"count", "batsman_runs": "sum"}).sort_values(by = batsman_overview["batting_strike_rate"] = ((batsman_overview["batsman_runs"]/batsman_overview["ball"])*100)
strike_rate = batsman_overview["batting_strike_rate"]
```

Top 5 batsmen

In [38]:

```
strike_rate.head(5).plot.bar(figsize=(15,5),fontsize = 14,width = 0.5,color = "g" )
plt.title("Top 5 Batsman",fontsize = 30)
# plt.tick_params(axis=
plt.xlabel("Batsman",fontsize = 20)
plt.ylabel("Runs",fontsize = 20)
plt.show()
```

Top 5 Batsman



In []:

In [39]:

merge.columns

```
Out[39]: Index(['index', 'match_id', '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', 'id', 'season', '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'],
      dtype='object')
```

Bowler information

In [40]:

```
bowler_overview = merge.groupby("bowler")[[ "total_runs", "ball", "win_by_wickets"]].agg({ "total_runs": "sum", "ball": "count", "win_b  
bowler_overview.head(20)
```

Out[40]:

bowler	total_runs	ball	win_by_wickets
Harbhajan Singh	4050	3451	3451
A Mishra	3850	3172	3172
PP Chawla	4153	3157	3157
R Ashwin	3391	3016	3016
SL Malinga	3511	2974	2974
DJ Bravo	3733	2711	2711
B Kumar	3264	2707	2707
P Kumar	3342	2637	2637
UT Yadav	3640	2605	2605
SP Narine	2939	2600	2600
RA Jadeja	3221	2541	2541
Z Khan	2860	2276	2276
DW Steyn	2454	2207	2207
R Vinay Kumar	3043	2186	2186
SR Watson	2751	2137	2137
IK Pathan	2711	2113	2113
I Sharma	2682	1999	1999
A Nehra	2537	1974	1974
PP Ojha	2399	1945	1945
RP Singh	2417	1874	1874

In [41]: `merge.columns`

```
Out[41]: Index(['index', 'match_id', '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', 'id', 'season', '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'],
      dtype='object')
```

In [42]: `bowler_overview = merge.groupby("bowler")
[["total_runs", "ball", "win_by_wickets"]].agg({ "total_runs": "sum", "ball": "count", "win_b
bowler_overview['six'] = ((bowler_overview["total_runs"])/6)
bowler_overview["economy"] = ((bowler_overview["ball"])/(bowler_overview["total_runs"])/6))
bowler_overview`

Out[42]:

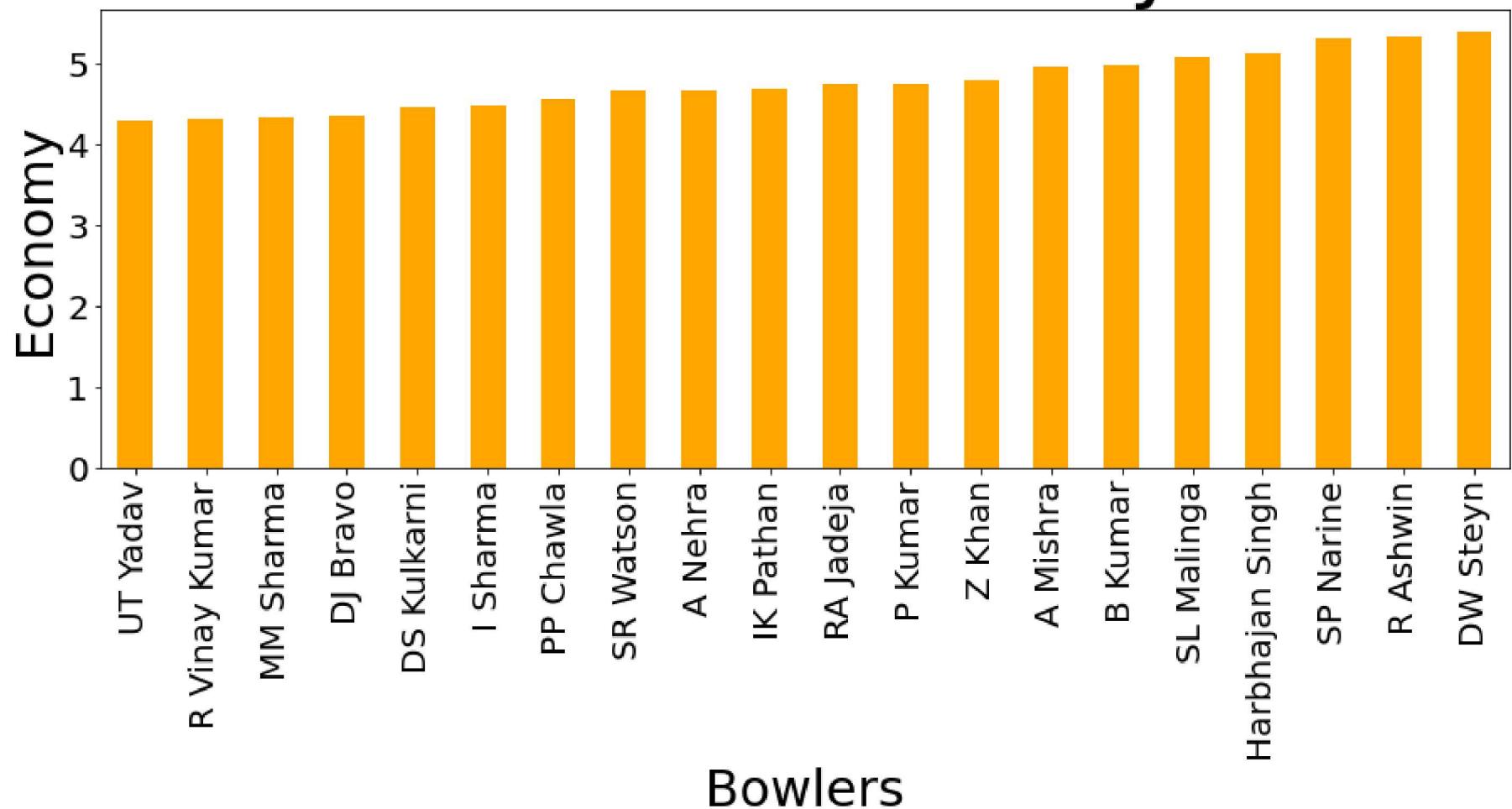
bowler	total_runs	ball	win_by_wickets	economy
Harbhajan Singh	4050	3451	3451	5.112593
A Mishra	3850	3172	3172	4.943377
PP Chawla	4153	3157	3157	4.561040
R Ashwin	3391	3016	3016	5.336479
SL Malinga	3511	2974	2974	5.082313
...
P Prasanth	18	6	6	2.000000
RA Shaikh	11	6	6	3.272727
SPD Smith	5	2	2	2.400000
SN Khan	6	2	2	2.000000
AC Gilchrist	0	1	1	inf

405 rows × 4 columns

In [43]:

```
bowler_overview = merge.groupby("bowler")[[ "total_runs", "ball", "win_by_wickets"]].agg({ "total_runs": "sum", "ball": "count", "win_b  
# bowler_overview['six'] = ((bowler_overview["total_runs"])/6)  
bowler_overview[ "economy" ] = ((bowler_overview[ "ball" ])/((bowler_overview[ "total_runs" ])/6))  
bowler_graph = bowler_overview[ "economy" ].head(20).sort_values(ascending = True)  
bowler_graph.plot.bar(figsize=(15,5), color = 'orange', fontsize = 20)  
plt.title("Bowlers Economy", fontsize = 50)  
plt.xlabel("Bowlers", fontsize = 30)  
plt.ylabel("Economy", fontsize = 30)  
plt.show()
```

Bowlers Economy



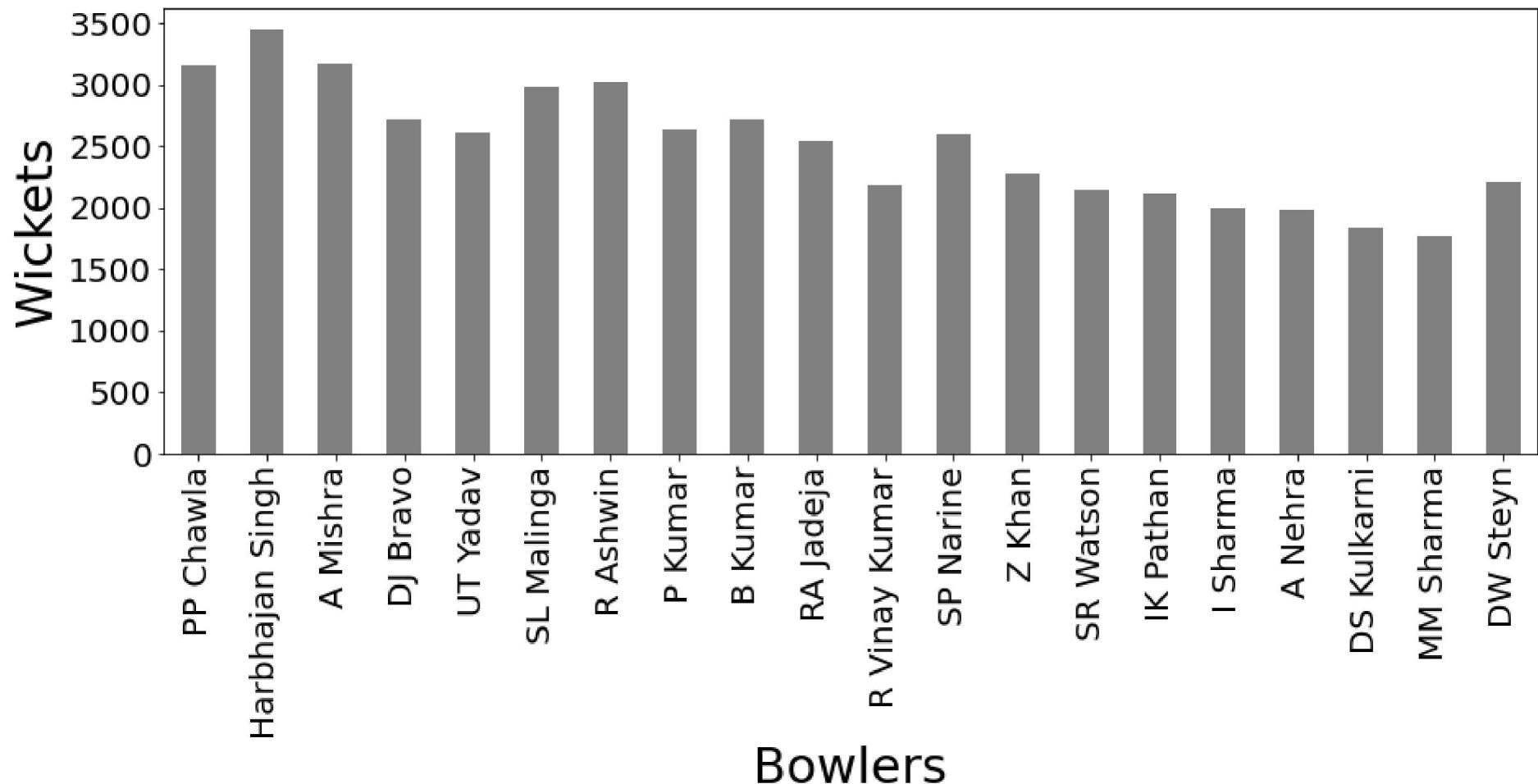
Wickets taken by a bowler

In [44]:

```
bowler_overview = merge.groupby("bowler")#[["total_runs","ball","win_by_wickets"]].agg({"total_runs":"sum", "ball": "count", "win_b  
bowler_graph_wickets = bowler_overview["win_by_wickets"].head(20)  
bowler_graph_wickets.plot.bar(figsize=(15,5), color = 'gray', fontsize = 20)  
plt.title("Bowlers Wickets", fontsize = 50)  
plt.xlabel("Bowlers", fontsize = 30)
```

```
plt.ylabel("Wickets", fontsize = 30)  
plt.show()
```

Bowlers Wickets

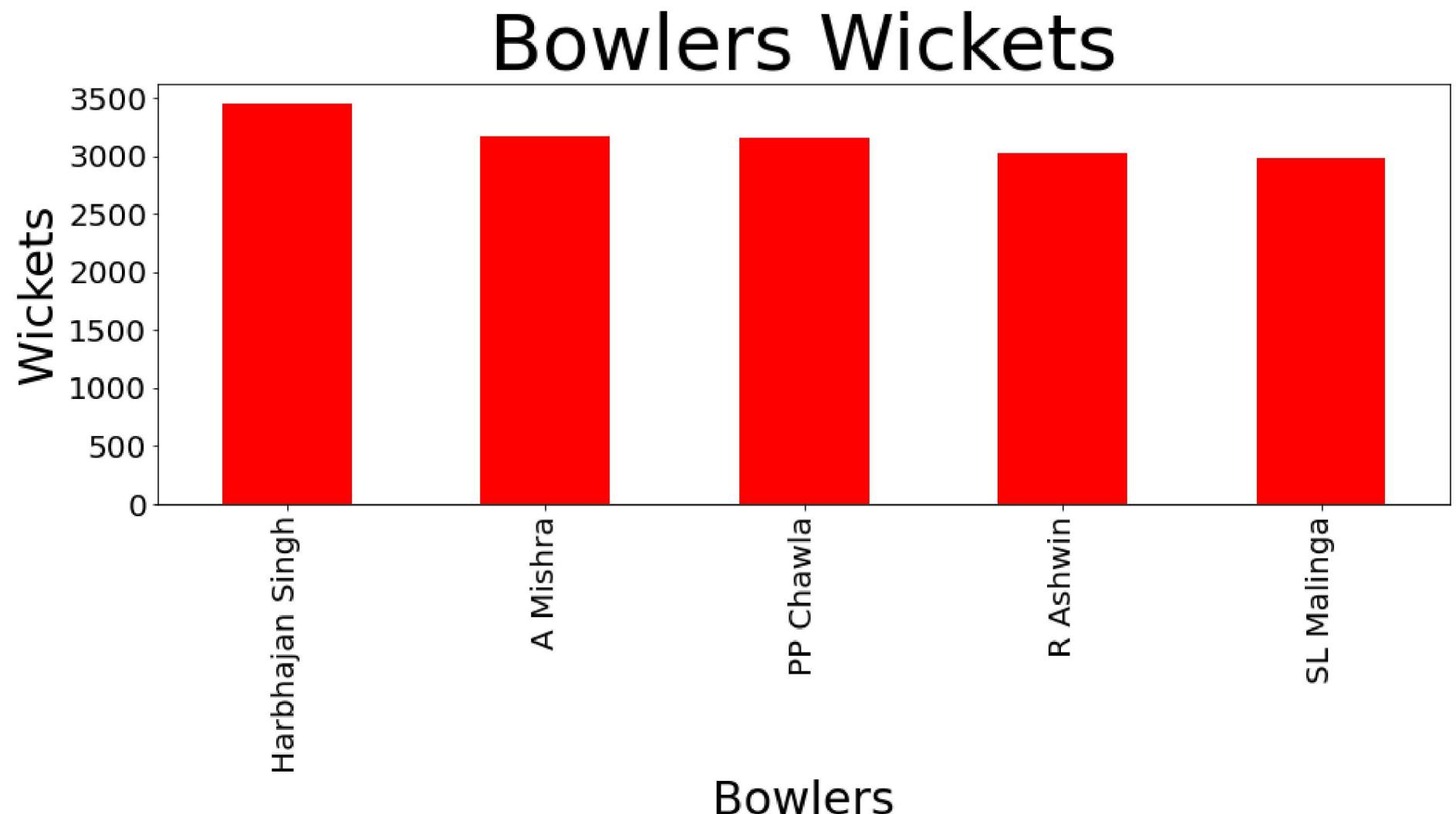


Top 5 Bowlers

In [45]:

```
bowler_overview = merge.groupby("bowler")[["total_runs", "ball", "win_by_wickets"]].agg({"total_runs": "sum", "ball": "count", "win_b  
bowler_graph_top5 = bowler_overview["win_by_wickets"].head(5).sort_values( ascending = False)  
bowler_graph_top5.plot.bar(figsize=(15,5), color = 'red', fontsize = 20)
```

```
plt.title("Bowlers Wickets", fontsize = 50)
plt.xlabel("Bowlers", fontsize = 30)
plt.ylabel("Wickets", fontsize = 30)
plt.show()
```



In [46]:
bowler_graph_top5.head(5)

Out[46]:
bowler

```
Harbhajan Singh      3451
A Mishra            3172
PP Chawla          3157
R Ashwin           3016
SL Malinga          2974
Name: win_by_wickets, dtype: int64
```

5. Conclusion:

write conclusion here

here , In this project (Ipl Data Analysis) where we learn about data handling (from read the data , familirising, data wrangling, and visualization)

cricket is a very intresting Game , from young generation to older generation all of them will take this game very seriously, from village to city (It is an Emotion of indians).

here , while analysing the data i ahve seen lots of factors that can affect the game's result. we can predict the probability of each over, strike rate and more. we have seen lots of insights while doing data cleaning, and visualization. from this we can predict which batsmat , or bowler, or overall team work better