In [1]:
```
import numpy as np
import pandas as pd
import os
print(os.getcwd())
!ls /home/swarnim/Downloads
```

```
/home/swarnim/Extra/jupyter
'Arcane S02E09 The Dirt Under Your Nails 1080p NF WEB-DL DD 5 1 Atmos H 264-
playWEB[TGx]'
 Jamboore.ipynb
 Jamboree___Linear_Regression.docx___Google_Docs.pdf
 ScalerResume.pdf
'Task 2 (Data Science).pdf'
 Venom.Let.There.Be.Carnage.2021.1080p.AMZN.WEB-DL.DDP5.1.H.264-alfaHD
 delhiivery.ipynb
 deliveries.csv
 deliveries.csv.zip
 docker-desktop-amd64.deb
 jamboore.pdf
 matches.csv
 pdf2doc
 pdf2doc.zip
 pinterest
 rld.pdf
'say.nothing.s01e01.1080p.web.h264-successfulcrab[EZTVx.to].mkv'
'say.nothing.s01e02.1080p.web.h264-successfulcrab[EZTVx.to].mkv'
 vertopal.com_Jamboore.pdf
 vertopal.com_delhiivery.pdf
```

In [2]:
```
matches = pd.read_csv('/home/swarnim/Downloads/matches.csv')
```

In [3]:
```
matches.head()
```

Out[3]:

	id	season	city	date	match_type	player_of_match	ven
0	335982	2007/08	Bangalore	2008-04-18	League	BB McCullum	Chinnaswa Stadi
1	335983	2007/08	Chandigarh	2008-04-19	League	MEK Hussey	Pun Cric Associat Stadiu Moh
2	335984	2007/08	Delhi	2008-04-19	League	MF Maharoof	Feroz Sh Ko
3	335985	2007/08	Mumbai	2008-04-20	League	MV Boucher	Wankhe Stadi
4	335986	2007/08	Kolkata	2008-04-20	League	DJ Hussey	Ec Garde

In [4]:
```
balls = pd.read_csv('/home/swarnim/Downloads/deliveries.csv')
```

In [5]:
```
balls.head()
```

Out[5]:

	match_id	inning	batting_team	bowling_team	over	ball	batter	bowle
0	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	1	SC Ganguly	I Kuma
1	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	2	BB McCullum	I Kuma
2	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	3	BB McCullum	I Kuma
3	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	4	BB McCullum	I Kuma
4	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	5	BB McCullum	I Kuma

In [6]:
```
print(matches['team1'].unique())
```

```
['Royal Challengers Bangalore' 'Kings XI Punjab' 'Delhi Daredevils'
 'Mumbai Indians' 'Kolkata Knight Riders' 'Rajasthan Royals'
 'Deccan Chargers' 'Chennai Super Kings' 'Kochi Tuskers Kerala'
 'Pune Warriors' 'Sunrisers Hyderabad' 'Gujarat Lions'
 'Rising Pune Supergiants' 'Rising Pune Supergiant' 'Delhi Capitals'
 'Punjab Kings' 'Lucknow Super Giants' 'Gujarat Titans'
 'Royal Challengers Bengaluru']
```

In [7]:
```
# Create a mapping dictionary for old team names to new names
team_name_mapping = {
    'Kings XI Punjab': 'Punjab Kings',
    'Delhi Daredevils': 'Delhi Capitals',
    'Rising Pune Supergiants': 'Rising Pune Supergiant',
    'Royal Challengers Bengaluru': 'Royal Challengers Bangalore'
    # Add more mappings if needed
}

# Replace old team names with new names in 'team1', 'team2', and 'winner' co
matches['team1'] = matches['team1'].replace(team_name_mapping)
matches['team2'] = matches['team2'].replace(team_name_mapping)
matches['winner'] = matches['winner'].replace(team_name_mapping)

# Verify the changes
print(matches[['team1', 'team2', 'winner']].head())
```

```
                         team1                        team2  \
0  Royal Challengers Bangalore        Kolkata Knight Riders
1                Punjab Kings          Chennai Super Kings
2              Delhi Capitals             Rajasthan Royals
3              Mumbai Indians  Royal Challengers Bangalore
4        Kolkata Knight Riders              Deccan Chargers

                       winner
0        Kolkata Knight Riders
1          Chennai Super Kings
2                Delhi Capitals
3  Royal Challengers Bangalore
4        Kolkata Knight Riders
```

In [8]:
```
print(matches['team1'].unique())
```

```
['Royal Challengers Bangalore' 'Punjab Kings' 'Delhi Capitals'
 'Mumbai Indians' 'Kolkata Knight Riders' 'Rajasthan Royals'
 'Deccan Chargers' 'Chennai Super Kings' 'Kochi Tuskers Kerala'
 'Pune Warriors' 'Sunrisers Hyderabad' 'Gujarat Lions'
 'Rising Pune Supergiant' 'Lucknow Super Giants' 'Gujarat Titans']
```

In [9]:
```
print(matches.info())
print(matches.shape)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1095 entries, 0 to 1094
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   id               1095 non-null   int64
 1   season           1095 non-null   object
 2   city             1044 non-null   object
 3   date             1095 non-null   object
 4   match_type       1095 non-null   object
 5   player_of_match  1090 non-null   object
 6   venue            1095 non-null   object
 7   team1            1095 non-null   object
 8   team2            1095 non-null   object
 9   toss_winner      1095 non-null   object
 10  toss_decision    1095 non-null   object
 11  winner           1090 non-null   object
 12  result           1095 non-null   object
 13  result_margin    1076 non-null   float64
 14  target_runs      1092 non-null   float64
 15  target_overs     1092 non-null   float64
 16  super_over       1095 non-null   object
 17  method           21 non-null     object
 18  umpire1          1095 non-null   object
 19  umpire2          1095 non-null   object
dtypes: float64(3), int64(1), object(16)
memory usage: 171.2+ KB
None
(1095, 20)
```

In [10]:
```
# lets see some basic observations
# matches playd by each team
```

```
print(matches['team1'].value_counts())
print(matches['team2'].value_counts())
```

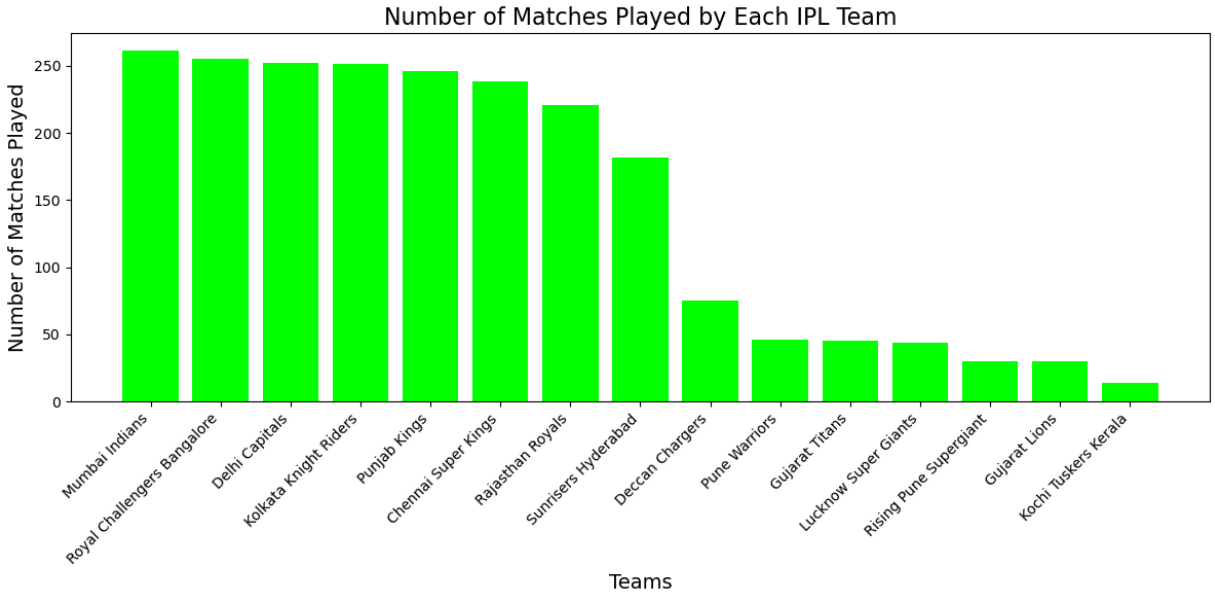
```
team1
Royal Challengers Bangalore    144
Chennai Super Kings            128
Delhi Capitals                 126
Mumbai Indians                 123
Punjab Kings                   123
Kolkata Knight Riders          121
Rajasthan Royals               101
Sunrisers Hyderabad             86
Deccan Chargers                 39
Pune Warriors                   23
Lucknow Super Giants            23
Gujarat Titans                  21
Gujarat Lions                   16
Rising Pune Supergiant          14
Kochi Tuskers Kerala             7
Name: count, dtype: int64
team2
Mumbai Indians                 138
Kolkata Knight Riders          130
Delhi Capitals                 126
Punjab Kings                   123
Rajasthan Royals               120
Royal Challengers Bangalore    111
Chennai Super Kings            110
Sunrisers Hyderabad             96
Deccan Chargers                 36
Gujarat Titans                  24
Pune Warriors                   23
Lucknow Super Giants            21
Rising Pune Supergiant          16
Gujarat Lions                   14
Kochi Tuskers Kerala             7
Name: count, dtype: int64
```

In [11]:
```
#lets combine this
combined_team  = pd.concat( [matches['team1'],matches['team2']])
print(combined_team)
print(" ")
matches_by_each_team = combined_team.value_counts()
print(matches_by_each_team)
```

```
0       Royal Challengers Bangalore
1                     Punjab Kings
2                   Delhi Capitals
3                   Mumbai Indians
4             Kolkata Knight Riders
                    ...
1090            Sunrisers Hyderabad
1091          Kolkata Knight Riders
1092               Rajasthan Royals
1093               Rajasthan Royals
1094          Kolkata Knight Riders
Length: 2190, dtype: object

Mumbai Indians                 261
Royal Challengers Bangalore    255
Delhi Capitals                 252
Kolkata Knight Riders          251
Punjab Kings                   246
Chennai Super Kings            238
Rajasthan Royals               221
Sunrisers Hyderabad            182
Deccan Chargers                 75
Pune Warriors                   46
Gujarat Titans                  45
Lucknow Super Giants            44
Rising Pune Supergiant          30
Gujarat Lions                   30
Kochi Tuskers Kerala            14
Name: count, dtype: int64
```

In [12]:

```python
import matplotlib.pyplot as plt
plt.figure(figsize=(12, 6))
plt.bar(matches_by_each_team.index, matches_by_each_team.values, color='lime
plt.xlabel('Teams', fontsize=14)
plt.ylabel('Number of Matches Played', fontsize=14)
plt.title('Number of Matches Played by Each IPL Team', fontsize=16)
plt.xticks(rotation=45, ha='right')

plt.tight_layout()
plt.show()
```



In [13]:
```
matches.head()
```

Out[13]:

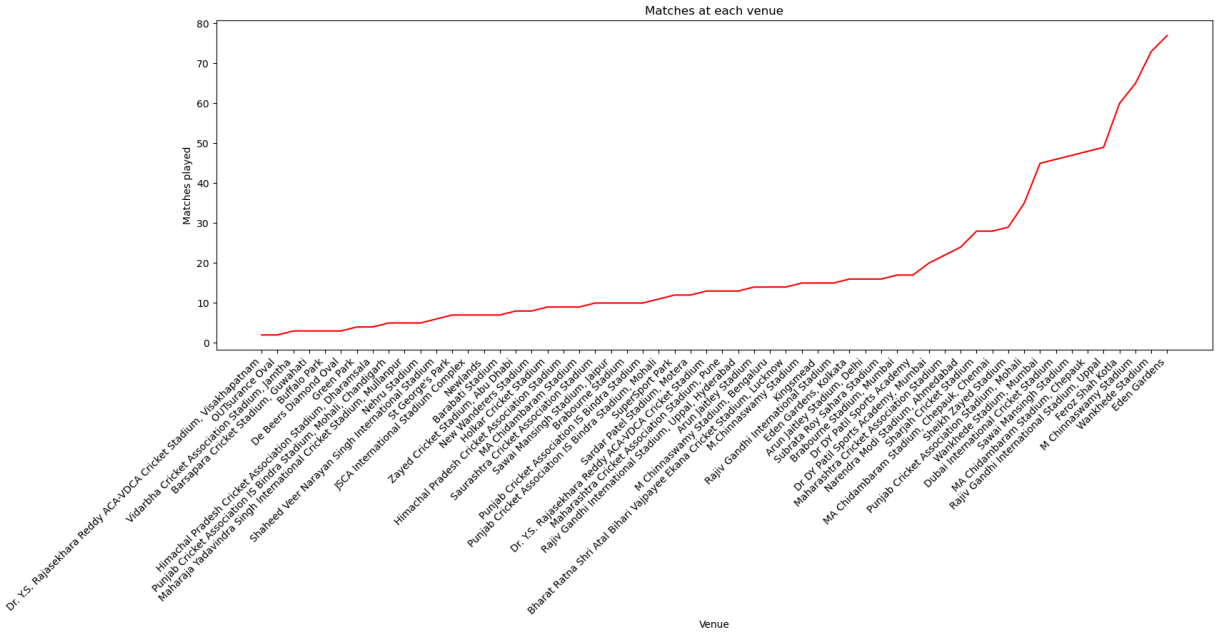
	id	season	city	date	match_type	player_of_match	ven
0	335982	2007/08	Bangalore	2008-04-18	League	BB McCullum	Chinnaswa Stadi
1	335983	2007/08	Chandigarh	2008-04-19	League	MEK Hussey	Pun Cric Associat Stadiu Moh
2	335984	2007/08	Delhi	2008-04-19	League	MF Maharoof	Feroz Sh Ko
3	335985	2007/08	Mumbai	2008-04-20	League	MV Boucher	Wankhe Stadi
4	335986	2007/08	Kolkata	2008-04-20	League	DJ Hussey	Ec Garde

In [14]:
```
venue_count = matches['venue'].value_counts()
venue_count = venue_count.sort_values(ascending = True)
print(venue_count)
```

venue	
Dr. Y.S. Rajasekhara Reddy ACA-VDCA Cricket Stadium, Visakhapatnam	2
OUTsurance Oval	2
Vidarbha Cricket Association Stadium, Jamtha	3
Barsapara Cricket Stadium, Guwahati	3
Buffalo Park	3
De Beers Diamond Oval	3
Green Park	4
Himachal Pradesh Cricket Association Stadium, Dharamsala	4
Punjab Cricket Association IS Bindra Stadium, Mohali, Chandigarh	5
Maharaja Yadavindra Singh International Cricket Stadium, Mullanpur	5
Nehru Stadium	5
Shaheed Veer Narayan Singh International Stadium	6
St George's Park	7
JSCA International Stadium Complex	7
Newlands	7
Barabati Stadium	7
Zayed Cricket Stadium, Abu Dhabi	8
New Wanderers Stadium	8
Holkar Cricket Stadium	9
Himachal Pradesh Cricket Association Stadium	9
MA Chidambaram Stadium	9
Saurashtra Cricket Association Stadium	10
Sawai Mansingh Stadium, Jaipur	10
Brabourne Stadium	10
Punjab Cricket Association IS Bindra Stadium	10
Punjab Cricket Association IS Bindra Stadium, Mohali	11
SuperSport Park	12
Sardar Patel Stadium, Motera	12
Dr. Y.S. Rajasekhara Reddy ACA-VDCA Cricket Stadium	13
Maharashtra Cricket Association Stadium, Pune	13
Rajiv Gandhi International Stadium, Uppal, Hyderabad	13
Arun Jaitley Stadium	14
M Chinnaswamy Stadium, Bengaluru	14
Bharat Ratna Shri Atal Bihari Vajpayee Ekana Cricket Stadium, Lucknow	14
M.Chinnaswamy Stadium	15
Kingsmead	15
Rajiv Gandhi International Stadium	15
Eden Gardens, Kolkata	16
Arun Jaitley Stadium, Delhi	16
Subrata Roy Sahara Stadium	16
Brabourne Stadium, Mumbai	17
Dr DY Patil Sports Academy	17
Dr DY Patil Sports Academy, Mumbai	20
Maharashtra Cricket Association Stadium	22
Narendra Modi Stadium, Ahmedabad	24
Sharjah Cricket Stadium	28
MA Chidambaram Stadium, Chepauk, Chennai	28
Sheikh Zayed Stadium	29
Punjab Cricket Association Stadium, Mohali	35
Wankhede Stadium, Mumbai	45
Dubai International Cricket Stadium	46
Sawai Mansingh Stadium	47
MA Chidambaram Stadium, Chepauk	48
Rajiv Gandhi International Stadium, Uppal	49
Feroz Shah Kotla	60

```
M Chinnaswamy Stadium                                                  65
Wankhede Stadium                                                       73
Eden Gardens                                                           77
Name: count, dtype: int64
```

In [15]:
```
plt.figure(figsize=(18, 6))  # Set the figure size using 'figsize'
plt.plot(venue_count.index, venue_count.values, linestyle="-", color="red")
plt.title("Matches at each venue")
plt.xlabel("Venue")
plt.ylabel("Matches played")
plt.xticks(rotation =45 , ha ='right')
plt.show()
```

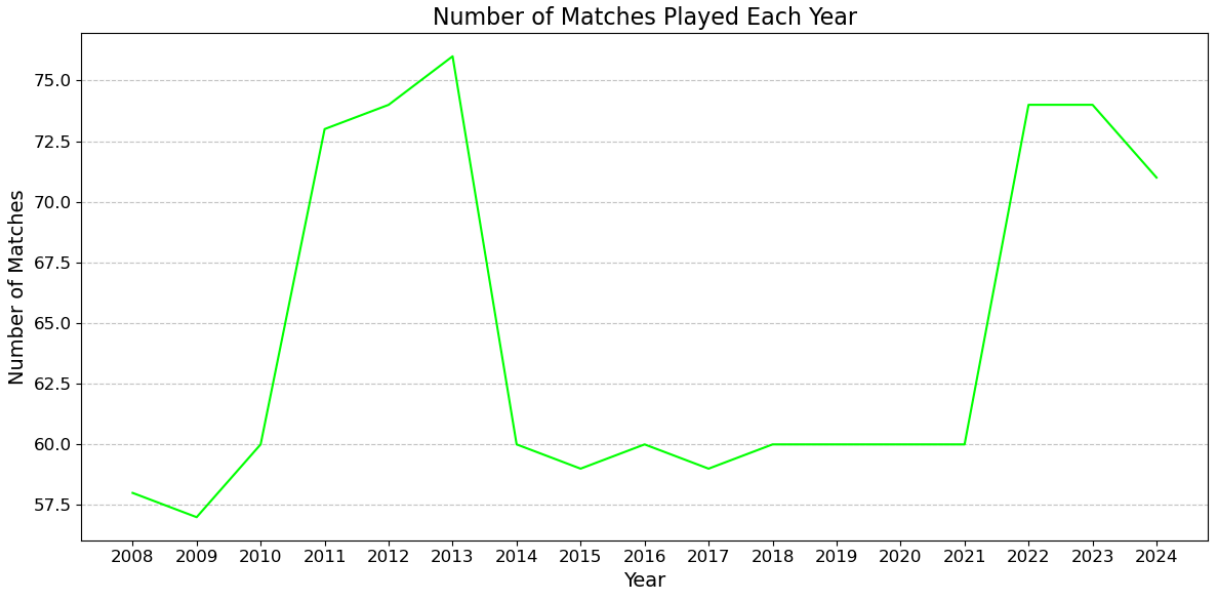


In [16]:
```
# Ensure the 'date' column is in datetime format
matches['date'] = pd.to_datetime(matches['date'])

# Extract the year from the 'date' column
matches['year'] = matches['date'].dt.year

# Count the number of matches played each year
matches_per_year = matches['year'].value_counts().sort_index()

# Print the result
print(matches_per_year)
```

```
year
2008    58
2009    57
2010    60
2011    73
2012    74
2013    76
2014    60
2015    59
2016    60
2017    59
2018    60
2019    60
2020    60
2021    60
2022    74
2023    74
2024    71
Name: count, dtype: int64
```

In [40]:

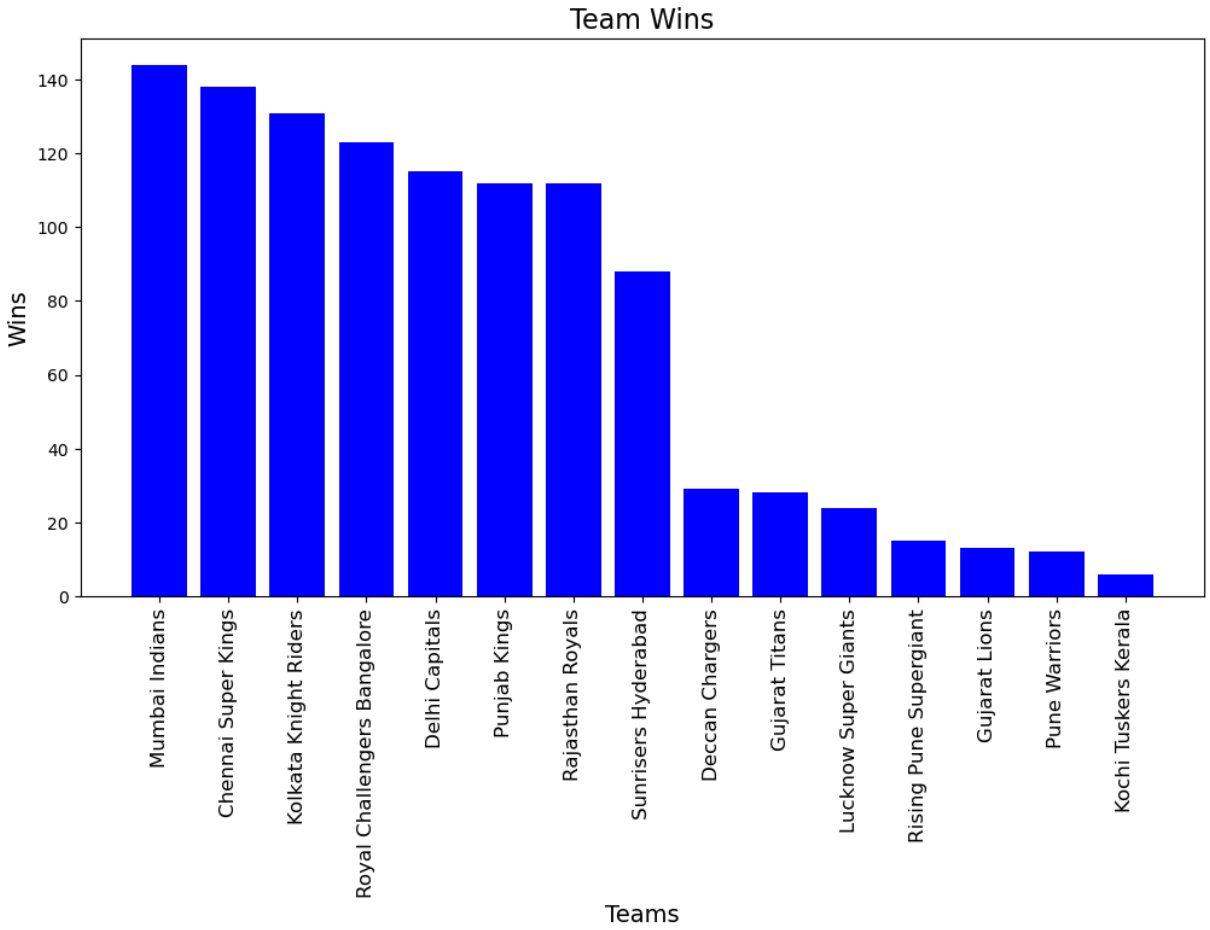
```
plt.figure(figsize=(12, 6))
plt.plot(matches_per_year.index, matches_per_year.values, color="lime")

# Add title and labels
plt.title("Number of Matches Played Each Year", fontsize=16)
plt.xlabel("Year", fontsize=14)
plt.ylabel("Number of Matches", fontsize=14)

plt.xticks(matches_per_year.index, fontsize=12)  # Use the years as x-axis l
plt.yticks(fontsize=12)

# Add grid for better readability (optional)
plt.grid(axis="y", linestyle="--", alpha=0.7)

# Adjust layout to prevent clipping
plt.tight_layout()

# Display the plot
plt.show()
```



In [18]:
```
team_wins = matches['winner'].value_counts()
print(team_wins)
```

```
winner
Mumbai Indians                144
Chennai Super Kings           138
Kolkata Knight Riders         131
Royal Challengers Bangalore   123
Delhi Capitals                115
Punjab Kings                  112
Rajasthan Royals              112
Sunrisers Hyderabad            88
Deccan Chargers                29
Gujarat Titans                 28
Lucknow Super Giants           24
Rising Pune Supergiant         15
Gujarat Lions                  13
Pune Warriors                  12
Kochi Tuskers Kerala            6
Name: count, dtype: int64
```

In [44]:
```
# Create a figure with a specified size
plt.figure(figsize=(12, 6))

# Create a bar plot
plt.bar(team_wins.index, team_wins.values, color="blue")

# Add title and labels
plt.title("Team Wins", fontsize=16)
plt.xlabel("Teams", fontsize=14)
plt.ylabel("Wins", fontsize=14)

# Set x-axis tick labels vertically
plt.xticks(rotation=90, fontsize=12)

# Display the plot
plt.show()
```



In [20]:
```
matches.head()
```

Out[20]:

	id	season	city	date	match_type	player_of_match	ven
0	335982	2007/08	Bangalore	2008-04-18	League	BB McCullum	Chinnaswa Stadi
1	335983	2007/08	Chandigarh	2008-04-19	League	MEK Hussey	Pun Cric Associat Stadiu Moh
2	335984	2007/08	Delhi	2008-04-19	League	MF Maharoof	Feroz Sh Ko
3	335985	2007/08	Mumbai	2008-04-20	League	MV Boucher	Wankhe Stadi
4	335986	2007/08	Kolkata	2008-04-20	League	DJ Hussey	Ec Garde

5 rows × 21 columns

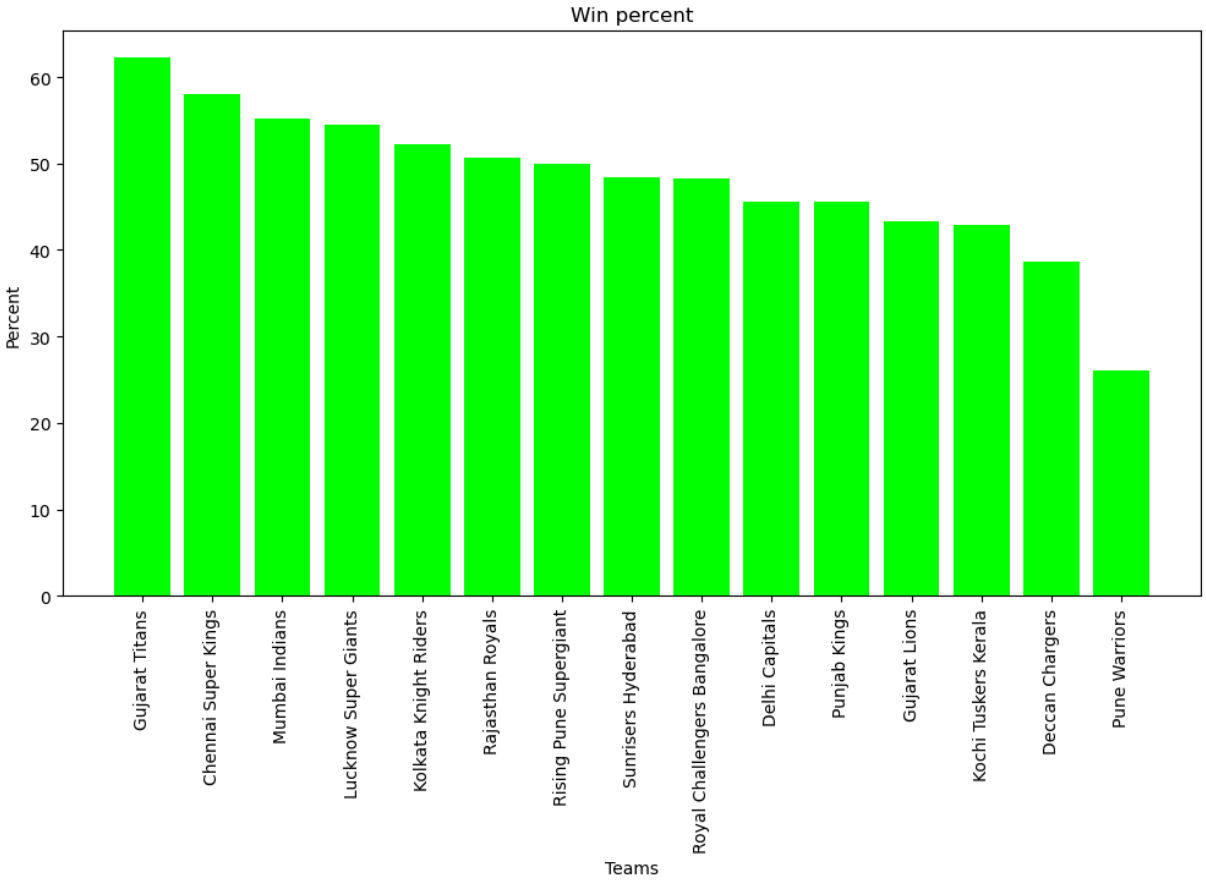
In [19]:
```
# Calculate the win percentage for each team
win_percentage = (team_wins / matches_by_each_team) * 100
```

```
win_percentage = win_percentage.sort_values(ascending=False)

# Print the win percentages
print(win_percentage)
```

```
Gujarat Titans                 62.222222
Chennai Super Kings            57.983193
Mumbai Indians                 55.172414
Lucknow Super Giants           54.545455
Kolkata Knight Riders          52.191235
Rajasthan Royals               50.678733
Rising Pune Supergiant         50.000000
Sunrisers Hyderabad            48.351648
Royal Challengers Bangalore    48.235294
Delhi Capitals                 45.634921
Punjab Kings                   45.528455
Gujarat Lions                  43.333333
Kochi Tuskers Kerala           42.857143
Deccan Chargers                38.666667
Pune Warriors                  26.086957
Name: count, dtype: float64
```

In [48]:
```
plt.figure(figsize=(12,6))
plt.bar(win_percentage.index , win_percentage.values , color="lime")
plt.title("Win percent")
plt.xlabel("Teams")
plt.ylabel("Percent")
plt.xticks(rotation=90 )
```

Out[48]:
```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14],
 [Text(0, 0, 'Gujarat Titans'),
  Text(1, 0, 'Chennai Super Kings'),
  Text(2, 0, 'Mumbai Indians'),
  Text(3, 0, 'Lucknow Super Giants'),
  Text(4, 0, 'Kolkata Knight Riders'),
  Text(5, 0, 'Rajasthan Royals'),
  Text(6, 0, 'Rising Pune Supergiant'),
  Text(7, 0, 'Sunrisers Hyderabad'),
  Text(8, 0, 'Royal Challengers Bangalore'),
  Text(9, 0, 'Delhi Capitals'),
  Text(10, 0, 'Punjab Kings'),
  Text(11, 0, 'Gujarat Lions'),
  Text(12, 0, 'Kochi Tuskers Kerala'),
  Text(13, 0, 'Deccan Chargers'),
  Text(14, 0, 'Pune Warriors')])
```



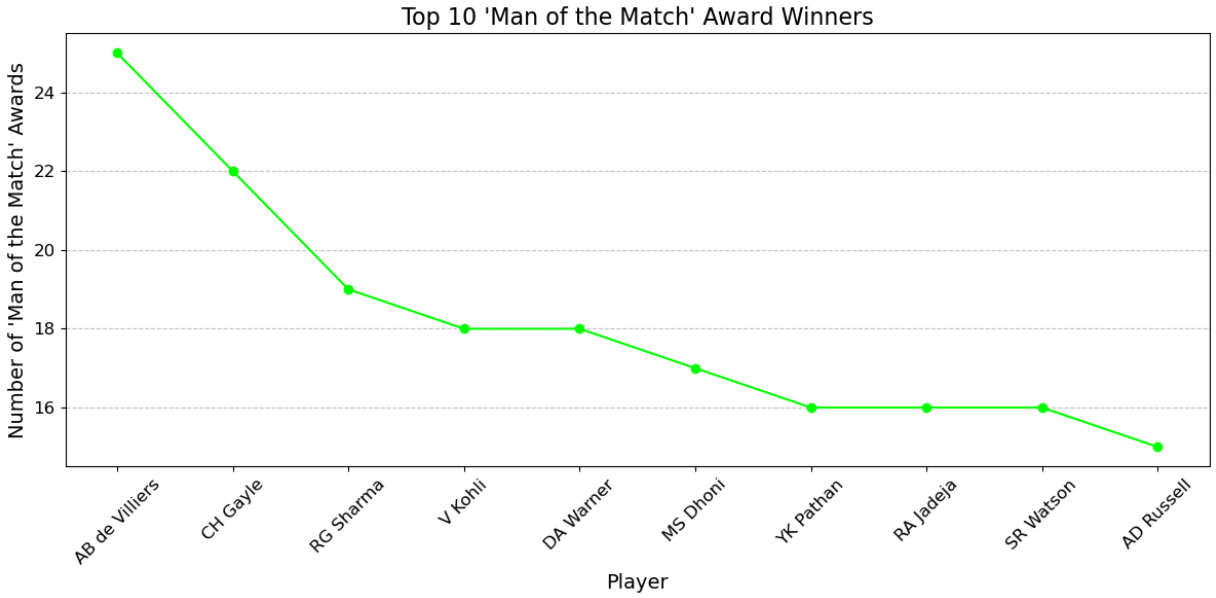
In [ ]:

In [21]:
```
man_of_the_match = matches['player_of_match'].value_counts(ascending=False)
print(man_of_the_match)
```

```
player_of_match
AB de Villiers    25
CH Gayle          22
RG Sharma         19
V Kohli           18
DA Warner         18
                  ..
R Shepherd         1
Akash Madhwal      1
Shashank Singh     1
WD Parnell         1
RR Rossouw         1
Name: count, Length: 291, dtype: int64
```

In [50]:
```
#Lets see the graph of top ten men of the matches
#fiitering
top_10_mom = matches['player_of_match'].value_counts(ascending=False).head(1

plt.figure(figsize=(12, 6))
plt.plot(top_10_mom.index, top_10_mom.values, marker='o', color='lime')

plt.title("Top 10 'Man of the Match' Award Winners", fontsize=16)
plt.xlabel("Player", fontsize=14)
plt.ylabel("Number of 'Man of the Match' Awards", fontsize=14)
```

```
plt.xticks(rotation=45, fontsize=12)
plt.yticks(fontsize=12)

plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.tight_layout()
plt.show()
```



In [22]:
```
balls.head()
```

Out[22]:

	match_id	inning	batting_team	bowling_team	over	ball	batter	bowle
0	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	1	SC Ganguly	I Kuma
1	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	2	BB McCullum	I Kuma
2	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	3	BB McCullum	I Kuma
3	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	4	BB McCullum	I Kuma
4	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	5	BB McCullum	I Kuma

In [23]:
```
balls.shape
```

Out[23]: (260920, 17)

In [24]:
```
balls.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 260920 entries, 0 to 260919
Data columns (total 17 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   match_id          260920 non-null  int64
 1   inning            260920 non-null  int64
 2   batting_team      260920 non-null  object
 3   bowling_team      260920 non-null  object
 4   over              260920 non-null  int64
 5   ball              260920 non-null  int64
 6   batter            260920 non-null  object
 7   bowler            260920 non-null  object
 8   non_striker       260920 non-null  object
 9   batsman_runs      260920 non-null  int64
 10  extra_runs        260920 non-null  int64
 11  total_runs        260920 non-null  int64
 12  extras_type       14125 non-null   object
 13  is_wicket         260920 non-null  int64
 14  player_dismissed  12950 non-null   object
 15  dismissal_kind    12950 non-null   object
 16  fielder           9354 non-null    object
dtypes: int64(8), object(9)
memory usage: 33.8+ MB
```

In [25]:
```
balls_bowled = balls.groupby('bowler')['ball'].count().sort_values(ascending
print(balls_bowled)
```

```
bowler
R Ashwin         4679
SP Narine        4146
B Kumar          4060
PP Chawla        3895
RA Jadeja        3895
                 ...
DA Warner           2
SN Khan             2
YBK Jaiswal         1
Atharva Taide       1
AC Gilchrist        1
Name: ball, Length: 530, dtype: int64
```

In [26]:
```
runs_given = balls.groupby('bowler')['total_runs'].sum().sort_values(ascendi
print(runs_given)
```

```
bowler
R Ashwin         5435
PP Chawla        5179
B Kumar          5051
RA Jadeja        4917
YS Chahal        4681
                 ...
SPD Smith           5
AM Rahane           5
Atharva Taide       4
DA Warner           2
AC Gilchrist        0
Name: total_runs, Length: 530, dtype: int64
```

In [27]:
```
wickets_taken = balls.groupby('bowler')['is_wicket'].sum().sort_values(ascer
print(wickets_taken)
```

```
bowler
YS Chahal          213
DJ Bravo           207
PP Chawla          201
SP Narine          200
R Ashwin           198
                  ...
Atharva Taide        0
Arshad Khan (2)      0
AUK Pathan           0
VS Yeligati          0
Vivrant Sharma       0
Name: is_wicket, Length: 530, dtype: int64
```

In [28]:
```
most_runs = balls.groupby('batter')['batsman_runs'].sum().sort_values(ascenc
print(most_runs)
```

```
batter
V Kohli           8014
S Dhawan          6769
RG Sharma         6630
DA Warner         6567
SK Raina          5536
                  ...
JL Denly             0
ND Doshi             0
C Nanda              0
Yash Dayal           0
V Pratap Singh       0
Name: batsman_runs, Length: 673, dtype: int64
```

In [30]:
```
balls_played = balls.groupby('batter')['ball'].count().sort_values(ascending
print(balls_played)
```

```
batter
V Kohli          6236
S Dhawan         5483
RG Sharma        5183
DA Warner        4849
SK Raina         4177
                 ...
Abdul Basith        1
Sunny Gupta         1
JL Denly            1
Mayank Dagar        1
DP Vijaykumar       1
Name: ball, Length: 673, dtype: int64
```

In [34]:
```python
# Filter batters who have played more than 50 balls
filtered_batters = balls_played[balls_played > 50]

# Calculate strike rate only for these batters
strike_rate = (most_runs[filtered_batters.index] / filtered_batters) * 100

# Sort the strike rate in descending order
strike_rate = strike_rate.sort_values(ascending=False)

print(strike_rate)
```

```
batter
J Fraser-McGurk    220.000000
WG Jacks           172.932331
PD Salt            169.610390
T Stubbs           169.456067
R Shepherd         169.117647
                         ...
Ravi Bishnoi        61.538462
S Sreesanth         59.649123
NS Naik             58.490566
S Nadeem            44.318182
YS Chahal           41.573034
Length: 382, dtype: float64
```

In [36]:
```python
career_stats = pd.DataFrame({
    'Total Runs': most_runs[filtered_batters.index],
    'Balls Faced': filtered_batters,
    'Strike Rate': strike_rate
}).sort_values(by='Total Runs', ascending=False)

print(career_stats)
```

```
              Total Runs  Balls Faced  Strike Rate
batter
V Kohli             8014         6236   128.511867
S Dhawan            6769         5483   123.454313
RG Sharma           6630         5183   127.918194
DA Warner           6567         4849   135.429986
SK Raina            5536         4177   132.535312
...                  ...          ...          ...
DJ Thornely           39           54    72.222222
YS Chahal             37           89    41.573034
S Sreesanth           34           57    59.649123
Ravi Bishnoi          32           52    61.538462
NS Naik               31           53    58.490566

[382 rows x 3 columns]
```

In [38]:
```
# Lets find the average of each batsmen
balls.head()
```

Out[38]:

	match_id	inning	batting_team	bowling_team	over	ball	batter	bowle
0	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	1	SC Ganguly	I Kuma
1	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	2	BB McCullum	I Kuma
2	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	3	BB McCullum	I Kuma
3	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	4	BB McCullum	I Kuma
4	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	5	BB McCullum	I Kuma

In [ ]:
```

```