Exploratory Data Analysis - Sports

Perform ‘Exploratory Data Analysis’ on dataset ‘Indian Premier League’

As a sports analysts, find out the most successful teams, players and factors contributing win or loss of a team.

Shital More

loading the required libraries

In [1]:
```
import os  #Standard imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
ipl=pd.read_csv('C:\\Users\\Shri Krupa\\Downloads\\matches.csv') #Loading the ip
```

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

Out[3]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_ap
0	1	2008	Bangalore	2008-04-18	Kolkata Knight Riders	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	
1	2	2008	Chandigarh	2008-04-19	Chennai Super Kings	Kings XI Punjab	Chennai Super Kings	bat	normal	
2	3	2008	Delhi	2008-04-19	Rajasthan Royals	Delhi Daredevils	Rajasthan Royals	bat	normal	
3	4	2008	Mumbai	2008-04-20	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	normal	
4	5	2008	Kolkata	2008-04-20	Deccan Chargers	Kolkata Knight Riders	Deccan Chargers	bat	normal	

In [4]:
```
ipl.shape #looking at the no. of rows and columns in the dataset
```

Out[4]:
```
(577, 18)
```

In [5]:
```
ipl['player_of_match'].value_counts() #Getting thr frequency of most man of the
```

Out[5]:
```
CH Gayle           17
YK Pathan          16
AB de Villiers     15
DA Warner          14
SK Raina           13
RG Sharma          13
MS Dhoni           12
AM Rahane          12
MEK Hussey         12
G Gambhir          12
V Sehwag           11
V Kohli            11
SR Watson          10
JH Kallis          10
DR Smith           10
SE Marsh            9
A Mishra            9
SR Tendulkar        8
KA Pollard          8
AC Gilchrist        7
RA Jadeja           7
AT Rayudu           6
UT Yadav            6
Harbhajan Singh     6
BJ Hodge            6
AD Russell          6
A Nehra             6
M Vijay             6
KC Sangakkara       5
DW Steyn            5
                   ..
MM Sharma           1
BA Bhatt            1
SP Goswami          1
AD Mascarenhas      1
MA Agarwal          1
RE Levi             1
MR Marsh            1
A Chandila          1
EJG Morgan          1
DP Nannes           1
JD Ryder            1
J Botha             1
M Kartik            1
LJ Wright           1
RS Bopara           1
SS Iyer             1
JDP Oram            1
CA Lynn             1
CL White            1
CRD Fernando        1
SB Jakati           1
RR Pant             1
TM Dilshan          1
MF Maharoof         1
```

```
SA Yadav           1
DJG Sammy          1
NM Coulter-Nile    1
SA Asnodkar        1
TL Suman           1
S Sohal            1
Name: player_of_match, Length: 187, dtype: int64
```

In [6]:
```
ipl['player_of_match'].value_counts()[0:5] #Getting the top 5 players with most
```

Out[6]:
```
CH Gayle          17
YK Pathan         16
AB de Villiers    15
DA Warner         14
SK Raina          13
Name: player_of_match, dtype: int64
```

In [7]:
```
list(ipl['player_of_match'].value_counts().keys())
```

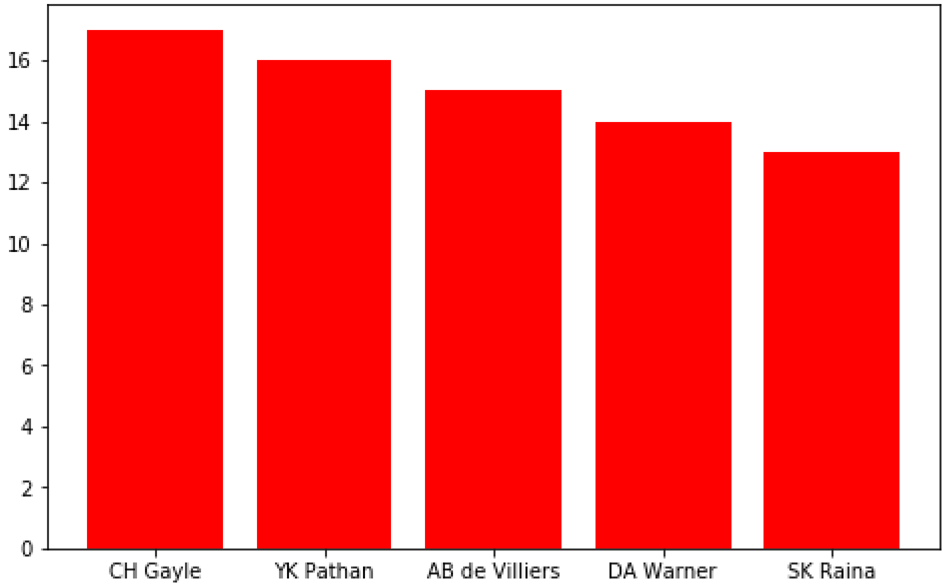
Out[7]:
```
['CH Gayle',
 'YK Pathan',
 'AB de Villiers',
 'DA Warner',
 'SK Raina',
 'RG Sharma',
 'MS Dhoni',
 'AM Rahane',
 'MEK Hussey',
 'G Gambhir',
 'V Sehwag',
 'V Kohli',
 'SR Watson',
 'JH Kallis',
 'DR Smith',
 'SE Marsh',
 'A Mishra',
 'SR Tendulkar',
 'KA Pollard',
 'AC Gilchrist',
```

In [8]:
```
list(ipl['player_of_match'].value_counts()[0:5].keys())
```

Out[8]:
```
['CH Gayle', 'YK Pathan', 'AB de Villiers', 'DA Warner', 'SK Raina']
```

In [ ]:

In [9]:
```
plt.figure(figsize=(8,5))
plt.bar(list(ipl['player_of_match'].value_counts()[0:5].keys()),list(ipl['player_
plt.show()
```



In [10]:
```
#making a bar plot for the top 5 players with most man of the match awards
```

In [11]:
```
ipl['result'].value_counts()  #getting the frequency of Result col
```

Out[11]:
```
normal       568
tie            6
no result      3
Name: result, dtype: int64
```

In [12]:
```
ipl['toss_winner'].value_counts()  #finding out the no. of toss wins w.r.t. each
```

Out[12]:
```
Mumbai Indians                 74
Kolkata Knight Riders          69
Chennai Super Kings            66
Delhi Daredevils               64
Kings XI Punjab                64
Rajasthan Royals               63
Royal Challengers Bangalore    61
Deccan Chargers                43
Sunrisers Hyderabad            30
Pune Warriors                  20
Gujarat Lions                   8
Kochi Tuskers Kerala            8
Rising Pune Supergiants         7
Name: toss_winner, dtype: int64
```

In [13]:
```
batting_first=ipl[ipl['win_by_runs']!=0] #extracting the records where a team wo
```

In [14]:
```
batting_first.head()   #looking at the head
```

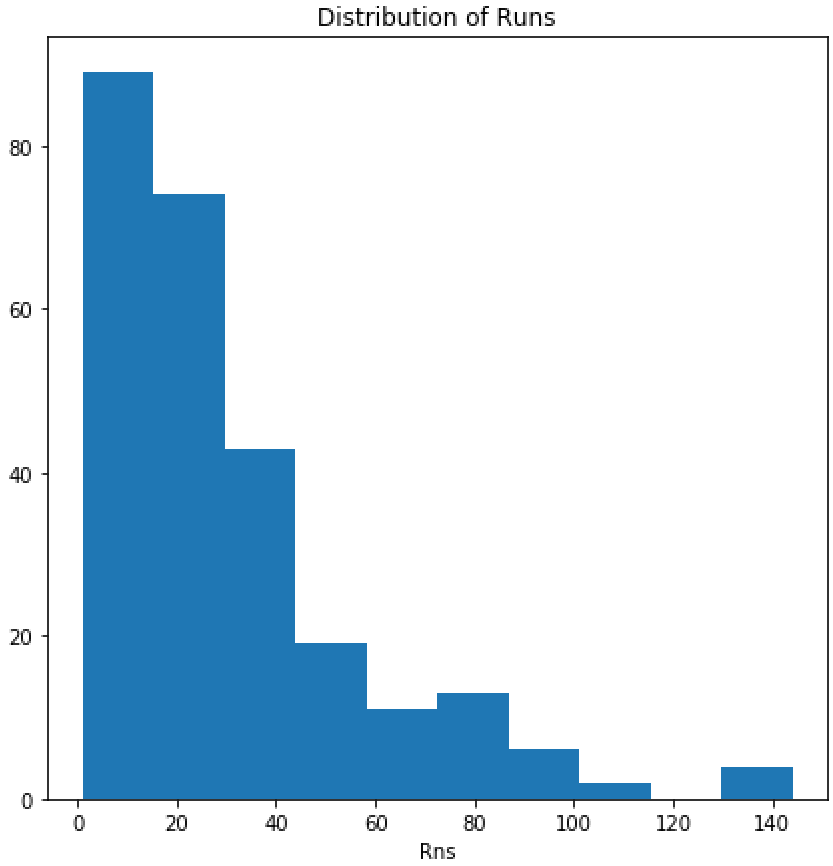
Out[14]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_ap
0	1	2008	Bangalore	2008-04-18	Kolkata Knight Riders	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	
1	2	2008	Chandigarh	2008-04-19	Chennai Super Kings	Kings XI Punjab	Chennai Super Kings	bat	normal	
7	8	2008	Chennai	2008-04-23	Chennai Super Kings	Mumbai Indians	Mumbai Indians	field	normal	
9	10	2008	Chandigarh	2008-04-25	Kings XI Punjab	Mumbai Indians	Mumbai Indians	field	normal	
14	15	2008	Bangalore	2008-04-28	Chennai Super Kings	Royal Challengers Bangalore	Chennai Super Kings	bat	normal	

Making a histogram

In [15]:

```
plt.figure(figsize=(7,7))
plt.hist(batting_first['win_by_runs'])
plt.title("Distribution of Runs")
plt.xlabel("Rns")
plt.show()
```



In [16]:

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

Out[16]:

```
Chennai Super Kings            46
Mumbai Indians                 43
Kolkata Knight Riders          29
Kings XI Punjab                28
Royal Challengers Bangalore    27
Rajasthan Royals               23
Delhi Daredevils               18
Deccan Chargers                18
Sunrisers Hyderabad            18
Pune Warriors                   6
Rising Pune Supergiants         2
Kochi Tuskers Kerala            2
Gujarat Lions                   1
Name: winner, dtype: int64
```

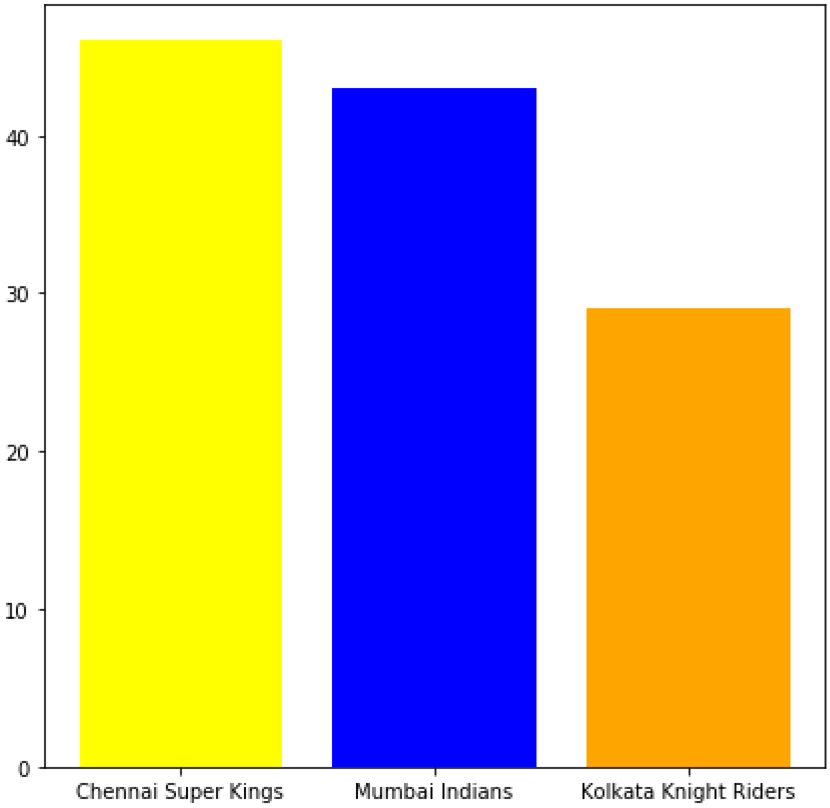
In [17]:
```
batting_first['winner'].value_counts().keys()  #finding out the no. of wins w.r.
```

Out[17]:
```
Index(['Chennai Super Kings', 'Mumbai Indians', 'Kolkata Knight Riders',
       'Kings XI Punjab', 'Royal Challengers Bangalore', 'Rajasthan Royals',
       'Delhi Daredevils', 'Deccan Chargers', 'Sunrisers Hyderabad',
       'Pune Warriors', 'Rising Pune Supergiants', 'Kochi Tuskers Kerala',
       'Gujarat Lions'],
      dtype='object')
```

In [18]:
```
batting_first['winner'].value_counts()[0:3].keys()
```

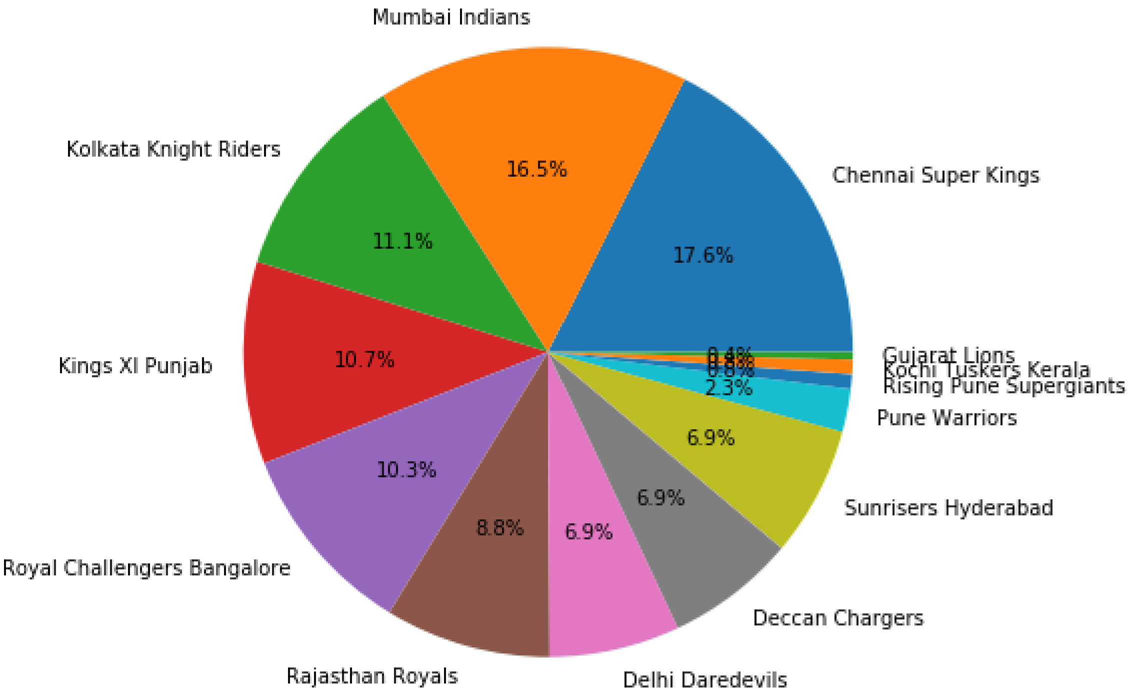
Out[18]:
```
Index(['Chennai Super Kings', 'Mumbai Indians', 'Kolkata Knight Riders'], dtype
='object')
```

In [19]:
```
#making a bar-plot for top 3 team with most wins after batting 1st
plt.figure(figsize=(7,7))
plt.bar(list(batting_first['winner'].value_counts()[0:3].keys()),list(batting_fi
plt.show()
```



Making a pie chart

In [20]:
```
plt.figure(figsize=(7,7))
plt.pie(list(batting_first['winner'].value_counts()),labels=list(batting_first['
plt.show()
```



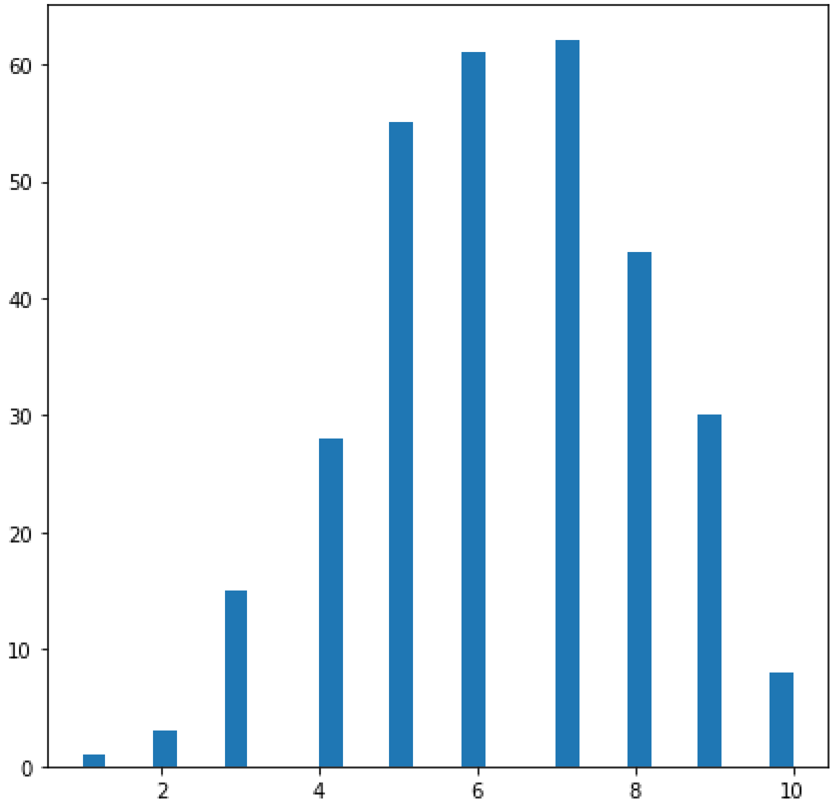
In [21]:
```
batting_second=ipl[ipl['win_by_wickets']!=0]  #extracting those records where a 
```

In [22]:
```
batting_second.head()  # looking at the head
```

Out[22]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_ap
2	3	2008	Delhi	2008-04-19	Rajasthan Royals	Delhi Daredevils	Rajasthan Royals	bat	normal	
3	4	2008	Mumbai	2008-04-20	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	normal	
4	5	2008	Kolkata	2008-04-20	Deccan Chargers	Kolkata Knight Riders	Deccan Chargers	bat	normal	
5	6	2008	Jaipur	2008-04-21	Kings XI Punjab	Rajasthan Royals	Kings XI Punjab	bat	normal	
6	7	2008	Hyderabad	2008-04-22	Deccan Chargers	Delhi Daredevils	Deccan Chargers	bat	normal	

In [23]:
```
plt.figure(figsize=(7,7))  #making a histogram for frequency of wins w,r.t no of
plt.hist(batting_second['win_by_wickets'],bins=30)
plt.show()
```

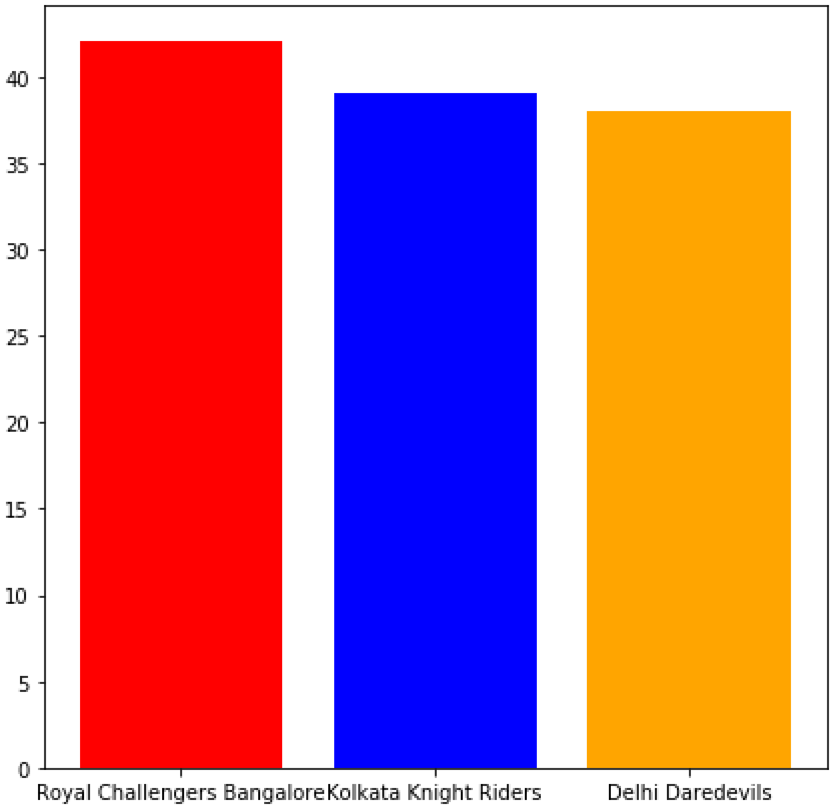


In [24]:
```
batting_second['winner'].value_counts() #finding out the freqency of wins w.r.t 
```

Out[24]:
```
Royal Challengers Bangalore    42
Kolkata Knight Riders          39
Delhi Daredevils               38
Rajasthan Royals               38
Mumbai Indians                 37
Kings XI Punjab                33
Chennai Super Kings            33
Sunrisers Hyderabad            15
Deccan Chargers                11
Gujarat Lions                   8
Pune Warriors                   6
Kochi Tuskers Kerala            4
Rising Pune Supergiants         3
Name: winner, dtype: int64
```

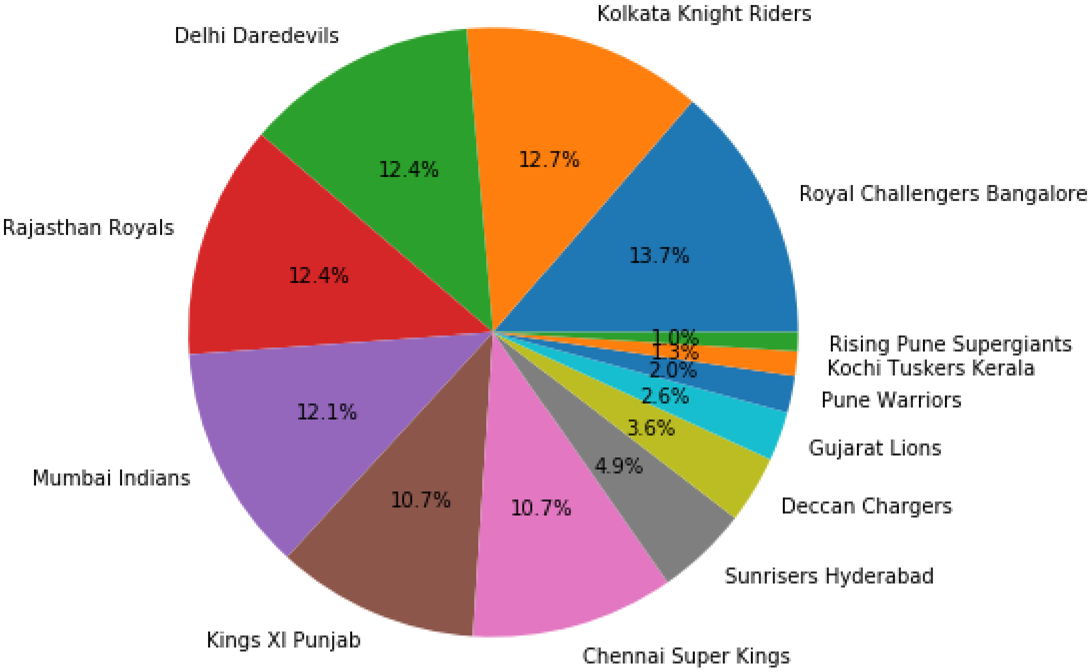
In [25]:
```
#making a bar plot for top 3 teams with most wins after batting 2nd
```

In [26]:
```
plt.figure(figsize=(7,7))
plt.bar(list(batting_second['winner'].value_counts()[0:3].keys()),list(batting_s
plt.show()
```



In [27]:
```
#making pie chart
```

In [28]:
```
plt.figure(figsize=(7,7))
plt.pie(list(batting_second['winner'].value_counts()),labels=list(batting_second
plt.show()
```



In [29]:
```
ipl['season'].value_counts()  #looking at the number  of matches played each sea
```

Out[29]:
```
2013    76
2012    74
2011    73
2016    60
2014    60
2010    60
2015    59
2008    58
2009    57
Name: season, dtype: int64
```

```
In [30]: ipl['city'].value_counts() # looking the no of matches played in each city
```

```
Out[30]: Mumbai            77
         Bangalore         58
         Kolkata           54
         Delhi             53
         Chennai           48
         Chandigarh        42
         Hyderabad         41
         Jaipur            33
         Pune              25
         Durban            15
         Centurion         12
         Ahmedabad         12
         Visakhapatnam     11
         Dharamsala         9
         Johannesburg       8
         Ranchi             7
         Cuttack            7
         Abu Dhabi          7
         Cape Town          7
         Port Elizabeth     7
         Raipur             6
         Sharjah            6
         Rajkot             5
         Kochi              5
         Nagpur             3
         Kimberley          3
         East London        3
         Bloemfontein       2
         Kanpur             2
         Indore             2
         Name: city, dtype: int64
```

```
In [31]: import numpy as np  #finding  out how many times a team has won the match after
         np.sum(ipl['toss_winner']==ipl['winner'])
```

```
Out[31]: 291
```

```
In [32]: 291/577
```

```
Out[32]: 0.5043327556325823
```

```
In [ ]:
```

conculsion:

There is no such advantage or not a clr advantage which you can decipher from a team winning the toss and going ahead to win the match

In [ ]:

```
```

In [44]:

```
deliveries=pd.read_csv('C:\\Users\\Shri Krupa\\Downloads\\deliveries.csv')
```

In [45]:

```
deliveries.head()
```

Out[45]:

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler	is_sup
0	1	1	Kolkata Knight Riders	Royal Challengers Bangalore	1	1	SC Ganguly	BB McCullum	P Kumar	
1	1	1	Kolkata Knight Riders	Royal Challengers Bangalore	1	2	BB McCullum	SC Ganguly	P Kumar	
2	1	1	Kolkata Knight Riders	Royal Challengers Bangalore	1	3	BB McCullum	SC Ganguly	P Kumar	
3	1	1	Kolkata Knight Riders	Royal Challengers Bangalore	1	4	BB McCullum	SC Ganguly	P Kumar	
4	1	1	Kolkata Knight Riders	Royal Challengers Bangalore	1	5	BB McCullum	SC Ganguly	P Kumar	

5 rows × 21 columns

In [50]:
```
deliveries['match_id'].unique()  #577 unique id matches
```

Out[50]:
```
array([  1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,  13,
        14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,  26,
        27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,  38,  39,
        40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,  52,
        53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,  65,
        66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,  78,
        79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,  91,
        92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 103, 104,
       105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117,
       118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130,
       131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143,
       144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156,
       157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169,
       170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182,
       183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195,
       196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208,
       209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221,
       222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234,
       235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247,
       248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260,
       261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273,
       274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286,
       287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299,
       300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312,
       313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325,
       326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338,
       339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351,
       352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364,
       365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377,
       378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390,
       391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403,
       404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416,
       417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429,
       430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442,
       443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455,
       456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468,
       469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481,
       482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494,
       495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507,
       508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520,
       521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533,
       534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546,
       547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559,
       560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572,
       573, 574, 575, 576, 577], dtype=int64)
```

In [54]:
```
match_1=deliveries[deliveries['match_id']==1] # particular  analysis match betwe
```

In [55]:
```
match_1.head()
```

Out[55]:

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler	is_sup
0	1	1	Kolkata Knight Riders	Royal Challengers Bangalore	1	1	SC Ganguly	BB McCullum	P Kumar	
1	1	1	Kolkata Knight Riders	Royal Challengers Bangalore	1	2	BB McCullum	SC Ganguly	P Kumar	
2	1	1	Kolkata Knight Riders	Royal Challengers Bangalore	1	3	BB McCullum	SC Ganguly	P Kumar	
3	1	1	Kolkata Knight Riders	Royal Challengers Bangalore	1	4	BB McCullum	SC Ganguly	P Kumar	
4	1	1	Kolkata Knight Riders	Royal Challengers Bangalore	1	5	BB McCullum	SC Ganguly	P Kumar	

5 rows × 21 columns

In [56]:
```
match_1.shape   #6*20=120+120=240
```

Out[56]:
```
(225, 21)
```

In [57]:
```
kkr=match_1[match_1['inning']==1]  # store in this object called kkr
```

In [58]:
```
kkr['batsman_runs'].value_counts() #how this batsman have scored run
```

Out[58]:
```
0    45
1    39
4    15
6    14
2    11
Name: batsman_runs, dtype: int64
```

In [59]:
```
kkr['dismissal_kind'].value_counts()  # only 3 wickets of kkr have fallen down
```

Out[59]:
```
caught    3
Name: dismissal_kind, dtype: int64
```

In [60]:
```
rcb=match_1[match_1['inning']==2]
```

In [61]:
```
rcb['batsman_runs'].value_counts()
```

Out[61]:
```
0    65
1    27
6     3
4     3
2     3
Name: batsman_runs, dtype: int64
```

In [63]:
```
rcb['dismissal_kind'].value_counts() #only 10 wickets  of rcb have falen down
```

Out[63]:
```
caught     6
bowled     3
run out    1
Name: dismissal_kind, dtype: int64
```

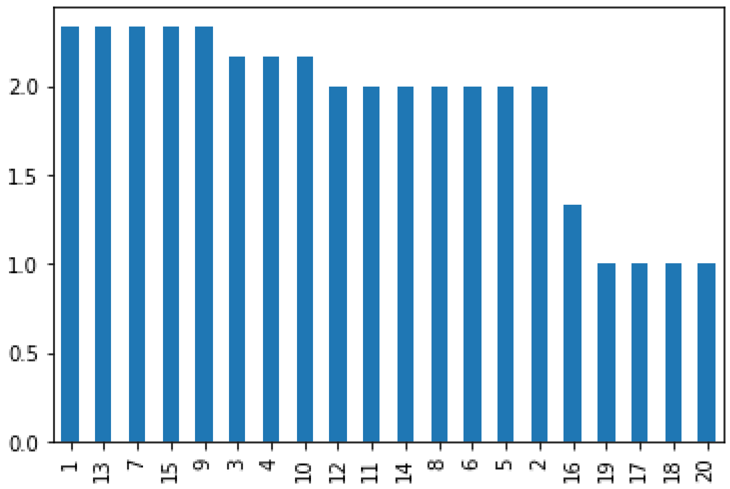
In [ ]:
```
#1st inning how much run made
```

In [73]:
```
over=match_1[match_1['match_id']==1]['over']  # all col
run=match_1[ (match_1['match_id']==1) & (match_1['over']==1)]['total_runs'].sum(
run
```

Out[73]:
```
7
```

In [74]:
```
(match_1['over'].value_counts()/6).plot(kind='bar')
```

Out[74]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f8888c1d68>
```



In [ ]:
```
```

In [ ]:
```
```