world-cup-project

February 24, 2024

IMPORT DATA

```
[1]: import  numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     import warnings
     warnings.filterwarnings('ignore')
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_25196\4090077741.py:2:
DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of
pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better
interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd
```

```
[2]: Match_data = pd.read_csv('matchSchedule.csv')
```

OBSERVE THE DATA

```
[3]: Match_data.head()
```

```
[3]:    Match_no    Date       Venue         Team1         Team2        Winner  \
     0         1  Oct-05   Ahmedabad       England   New Zealand   New Zealand
     1         2  Oct-06   Hyderabad      Pakistan   Netherlands      Pakistan
     2         3  Oct-07  Dharamsala    Bangladesh   Afghanistan    Bangladesh
     3         4  Oct-07       Delhi  South Africa     Sri Lanka  South Africa
     4         5  Oct-08     Chennai         India     Australia         India

                                           Scorecard URL  Unnamed: 7  Unnamed: 8  \
     0  https://www.cricketwa.com/scorecard/18020/engl…         NaN         NaN
     1  https://www.cricketwa.com/scorecard/18021/paki…         NaN         NaN
     2  https://www.cricketwa.com/scorecard/23008/bang…         NaN         NaN
     3  https://www.cricketwa.com/scorecard/23009/sout…         NaN         NaN
```

```
4  https://www.cricketwa.com/scorecard/23010/indi…        NaN        NaN

   Unnamed: 9  Unnamed: 10  Unnamed: 11  Unnamed: 12  Unnamed: 13  \
0         NaN          NaN          NaN          NaN          NaN
1         NaN          NaN          NaN          NaN          NaN
2         NaN          NaN          NaN          NaN          NaN
3         NaN          NaN          NaN          NaN          NaN
4         NaN          NaN          NaN          NaN          NaN

   Unnamed: 14  Unnamed: 15  Unnamed: 16  Unnamed: 17     Margin
0          NaN          NaN          NaN          NaN  9 wickets
1          NaN          NaN          NaN          NaN    81 runs
2          NaN          NaN          NaN          NaN  6 wickets
3          NaN          NaN          NaN          NaN   102 runs
4          NaN          NaN          NaN          NaN  6 wickets
```

REMOVE THE UNWANTED DATA

[4]:
```
columns_to_remove = ['Unnamed: 7', 'Unnamed: 8', 'Unnamed: 9', 'Unnamed: 10',
                     'Unnamed: 11', 'Unnamed: 12', 'Unnamed: 13', 'Unnamed: 14',
                     'Unnamed: 15', 'Unnamed: 16', 'Unnamed: 17']
```

[5]:
```
Match_data.drop(columns=columns_to_remove, inplace=True)
```

[6]:
```
Match_data.head()
```

[6]:
```
   Match_no    Date       Venue         Team1         Team2        Winner  \
0         1  Oct-05   Ahmedabad       England   New Zealand   New Zealand
1         2  Oct-06   Hyderabad      Pakistan   Netherlands      Pakistan
2         3  Oct-07  Dharamsala    Bangladesh   Afghanistan    Bangladesh
3         4  Oct-07       Delhi  South Africa     Sri Lanka  South Africa
4         5  Oct-08     Chennai         India     Australia         India

                                    Scorecard URL     Margin
0  https://www.cricketwa.com/scorecard/18020/engl…  9 wickets
1  https://www.cricketwa.com/scorecard/18021/paki…    81 runs
2  https://www.cricketwa.com/scorecard/23008/bang…  6 wickets
3  https://www.cricketwa.com/scorecard/23009/sout…   102 runs
4  https://www.cricketwa.com/scorecard/23010/indi…  6 wickets
```

TOTAL NUMBERS OF ROWS AND COLUMN

[7]:
```
print("Number Of Rows",Match_data.shape[0])
print("Number Of Columns",Match_data.shape[1])
```

```
Number Of Rows 48
Number Of Columns 8
```

TYPES OF DATA

```
[8]: Match_data.dtypes
```

```
[8]: Match_no          int64
     Date             object
     Venue            object
     Team1            object
     Team2            object
     Winner           object
     Scorecard URL    object
     Margin           object
     dtype: object
```

CONVERT DATA

```
[9]: Match_data['Date'] = pd.to_datetime(Match_data['Date'], format='%b-%d')
```

```
[10]: Match_data
```

```
[10]:     Match_no       Date                                              Venue  \
      0          1 1900-10-05                                          Ahmedabad
      1          2 1900-10-06                                          Hyderabad
      2          3 1900-10-07                                         Dharamsala
      3          4 1900-10-07                                              Delhi
      4          5 1900-10-08                                            Chennai
      5          6 1900-10-09                                          Hyderabad
      6          7 1900-10-10                                         Dharamsala
      7          8 1900-10-10                                          Hyderabad
      8          9 1900-10-11                                              Delhi
      9         10 1900-10-12                      Ekana Cricket Stadium Lucknow
      10        11 1900-10-13                                            Chennai
      11        12 1900-10-14                                          Ahmedabad
      12        13 1900-10-15                                              Delhi
      13        14 1900-10-16                      Ekana Cricket Stadium Lucknow
      14        15 1900-10-17                                         Dharamsala
      15        16 1900-10-18                                            Chennai
      16        17 1900-10-19                                               Pune
      17        18 1900-10-20                                          Bengaluru
      18        19 1900-10-21                      Ekana Cricket Stadium Lucknow
      19        20 1900-10-21                                             Mumbai
      20        21 1900-10-22  Himachal Pradesh Cricket Association Stadium, …
      21        22 1900-10-23                                            Chennai
      22        23 1900-10-24                                             Mumbai
      23        24 1900-10-25                                              Delhi
      24        25 1900-10-26                                          Bengaluru
      25        26 1900-10-27                                            Chennai
      26        27 1900-10-28                                         Dharamsala
      27        28 1900-10-28                                            Kolkata
```

```
28        29 1900-10-29                      Ekana Cricket Stadium Lucknow
29        30 1900-10-30                                               Pune
30        31 1900-10-31                                            Kolkata
31        32 1900-11-01                                               Pune
32        33 1900-11-02                                             Mumbai
33        34 1900-11-03                      Ekana Cricket Stadium Lucknow
34        35 1900-11-04                                          Bengaluru
35        36 1900-11-04                                          Ahmedabad
36        37 1900-11-05                                            Kolkata
37        38 1900-11-06                                              Delhi
38        39 1900-11-07                                             Mumbai
39        40 1900-11-08                                               Pune
40        41 1900-11-09                                          Bengaluru
41        42 1900-11-10                                          Ahmedabad
42        43 1900-11-11                                               Pune
43        44 1900-11-11                                            Kolkata
44        45 1900-11-12                                          Bengaluru
45        46 1900-11-15                                             Mumbai
46        47 1900-11-16                                            Kolkata
47        48 1900-11-19                                          Ahmedabad

            Team1          Team2        Winner  \
0        England     New Zealand   New Zealand
1       Pakistan     Netherlands      Pakistan
2     Bangladesh     Afghanistan    Bangladesh
3   South Africa       Sri Lanka  South Africa
4          India       Australia         India
5    New Zealand     Netherlands   New Zealand
6        England     Bangladesh        England
7       Pakistan       Sri Lanka      Pakistan
8          India     Afghanistan         India
9      Australia    South Africa  South Africa
10   New Zealand      Bangladesh   New Zealand
11         India        Pakistan         India
12       England     Afghanistan   Afghanistan
13     Australia       Sri Lanka     Australia
14  South Africa     Netherlands   Netherlands
15   New Zealand     Afghanistan   New Zealand
16         India      Bangladesh         India
17     Australia        Pakistan     Australia
18   Netherlands       Sri Lanka     Sri Lanka
19       England    South Africa  South Africa
20         India     New Zealand         India
21      Pakistan     Afghanistan   Afghanistan
22  South Africa      Bangladesh  South Africa
23     Australia     Netherlands     Australia
24       England       Sri Lanka     Sri Lanka
```

```
25      Pakistan    South Africa  South Africa
26     Australia     New Zealand     Australia
27   Netherlands      Bangladesh   Netherlands
28         India         England         India
29   Afghanistan       Sri Lanka   Afghanistan
30      Pakistan      Bangladesh      Pakistan
31   New Zealand    South Africa  South Africa
32         India       Sri Lanka     Sri Lanka
33   Netherlands     Afghanistan   Afghanistan
34   New Zealand        Pakistan      Pakistan
35       England       Australia     Australia
36         India    South Africa         India
37    Bangladesh       Sri Lanka    Bangladesh
38     Australia     Afghanistan     Australia
39       England     Netherlands       England
40   New Zealand       Sri Lanka   New Zealand
41  South Africa     Afghanistan  South Africa
42     Australia      Bangladesh     Australia
43       England        Pakistan       England
44         India     Netherlands         India
45         India     New Zealand         India
46     Australia    South Africa     Australia
47          India      Australia     Australia

                                        Scorecard URL     Margin
0   https://www.cricketwa.com/scorecard/18020/engl…  9 wickets
1   https://www.cricketwa.com/scorecard/18021/paki…    81 runs
2   https://www.cricketwa.com/scorecard/23008/bang…  6 wickets
3   https://www.cricketwa.com/scorecard/23009/sout…   102 runs
4   https://www.cricketwa.com/scorecard/23010/indi…  6 wickets
5   https://www.cricketwa.com/scorecard/23011/new-…    99 runs
6   https://www.cricketwa.com/scorecard/23012/engl…   137 runs
7   https://www.cricketwa.com/scorecard/23014/paki…  6 wickets
8   https://www.cricketwa.com/scorecard/23013/indi…  8 wickets
9   https://www.cricketwa.com/scorecard/23015/aust…   134 runs
10  https://www.cricketwa.com/scorecard/23016/new-…  8 wickets
11  https://www.cricketwa.com/scorecard/23018/indi…  7 wickets
12  https://www.cricketwa.com/scorecard/23017/engl…    69 runs
13  https://www.cricketwa.com/scorecard/23019/aust…  5 wickets
14  https://www.cricketwa.com/scorecard/23020/sout…    38 runs
15  https://www.cricketwa.com/scorecard/23021/new-…   149 runs
16  https://www.cricketwa.com/scorecard/23022/indi…  7 wickets
17  https://www.cricketwa.com/scorecard/23023/aust…    62 runs
18  https://www.cricketwa.com/scorecard/23024/sri-…  5 wickets
19  https://www.cricketwa.com/scorecard/23025/engl…   229 runs
20  https://www.cricketwa.com/scorecard/23026/indi…  4 wickets
21  https://www.cricketwa.com/scorecard/23027/paki…  8 wickets
```

```
22  https://www.cricketwa.com/scorecard/23028/sout…   149 runs
23  https://www.cricketwa.com/scorecard/23029/aust…   309 runs
24  https://www.cricketwa.com/scorecard/23030/engl…  8 wickets
25  https://www.cricketwa.com/scorecard/23031/paki…   1 wicket
26  https://www.cricketwa.com/scorecard/23032/aust…     5 runs
27  https://www.cricketwa.com/scorecard/23033/bang…    87 runs
28  https://www.cricketwa.com/scorecard/23034/indi…   100 runs
29  https://www.cricketwa.com/scorecard/23035/afgh…  7 wickets
30  https://www.cricketwa.com/scorecard/23036/paki…  7 wickets
31  https://www.cricketwa.com/scorecard/23037/new-…   190 runs
32  https://www.cricketwa.com/scorecard/23038/indi…   302 runs
33  https://www.cricketwa.com/scorecard/23039/neth…  7 wickets
34  https://www.cricketwa.com/scorecard/23040/new-…    21 runs
35  https://www.cricketwa.com/scorecard/23041/engl…    33 runs
36  https://www.cricketwa.com/scorecard/23042/indi…   243 runs
37  https://www.cricketwa.com/scorecard/23043/bang…  3 wickets
38  https://www.cricketwa.com/scorecard/23044/aust…  3 wickets
39  https://www.cricketwa.com/scorecard/23045/engl…   160 runs
40  https://www.cricketwa.com/scorecard/23046/new-…  5 wickets
41  https://www.cricketwa.com/scorecard/23047/sout…  5 wickets
42  https://www.cricketwa.com/scorecard/23049/aust…  8 wickets
43  https://www.cricketwa.com/scorecard/23050/engl…    93 runs
44  https://www.cricketwa.com/scorecard/23048/indi…   160 runs
45  https://www.cricketwa.com/scorecard/23051/indi…    70 runs
46  https://www.cricketwa.com/scorecard/23052/aust…  3 wickets
47  https://www.cricketwa.com/scorecard/23053/indi…  6 wickets
```

[11]:
```
Match_data['Date'] = Match_data['Date'].dt.strftime('2023-%m-%d')
```

[12]:
```
Match_data.head()
```

[12]:
```
   Match_no        Date       Venue         Team1         Team2  \
0         1  2023-10-05   Ahmedabad       England   New Zealand
1         2  2023-10-06   Hyderabad      Pakistan   Netherlands
2         3  2023-10-07  Dharamsala    Bangladesh   Afghanistan
3         4  2023-10-07       Delhi  South Africa     Sri Lanka
4         5  2023-10-08     Chennai         India     Australia

         Winner                                    Scorecard URL     Margin
0   New Zealand  https://www.cricketwa.com/scorecard/18020/engl…  9 wickets
1      Pakistan  https://www.cricketwa.com/scorecard/18021/paki…    81 runs
2    Bangladesh  https://www.cricketwa.com/scorecard/23008/bang…  6 wickets
3  South Africa  https://www.cricketwa.com/scorecard/23009/sout…   102 runs
4         India  https://www.cricketwa.com/scorecard/23010/indi…  6 wickets
```

[13]:
```
Match_data.tail()
```

[13]:

```
    Match_no        Date      Venue      Team1          Team2     Winner  \
43        44  2023-11-11    Kolkata    England       Pakistan    England   
44        45  2023-11-12  Bengaluru      India    Netherlands      India   
45        46  2023-11-15     Mumbai      India    New Zealand      India   
46        47  2023-11-16    Kolkata  Australia   South Africa  Australia   
47        48  2023-11-19  Ahmedabad       India      Australia  Australia   

                                        Scorecard URL     Margin  
43  https://www.cricketwa.com/scorecard/23050/engl…    93 runs  
44  https://www.cricketwa.com/scorecard/23048/indi…   160 runs  
45  https://www.cricketwa.com/scorecard/23051/indi…    70 runs  
46  https://www.cricketwa.com/scorecard/23052/aust…  3 wickets  
47  https://www.cricketwa.com/scorecard/23053/indi…  6 wickets  
```

[14]:

```
Match_data['Venue'] = Match_data['Venue'].astype('string')
Match_data['Team1'] = Match_data['Team1'].astype('string')
Match_data['Team2'] = Match_data['Team2'].astype('string')
Match_data['Winner'] = Match_data['Winner'].astype('string')
```

[15]:

```
Match_data.head()
```

[15]:

```
   Match_no        Date       Venue         Team1          Team2  \
0         1  2023-10-05   Ahmedabad       England    New Zealand   
1         2  2023-10-06   Hyderabad      Pakistan    Netherlands   
2         3  2023-10-07  Dharamsala    Bangladesh    Afghanistan   
3         4  2023-10-07       Delhi  South Africa      Sri Lanka   
4         5  2023-10-08     Chennai         India      Australia   

         Winner                                      Scorecard URL     Margin  
0   New Zealand  https://www.cricketwa.com/scorecard/18020/engl…  9 wickets  
1      Pakistan  https://www.cricketwa.com/scorecard/18021/paki…    81 runs  
2    Bangladesh  https://www.cricketwa.com/scorecard/23008/bang…  6 wickets  
3  South Africa  https://www.cricketwa.com/scorecard/23009/sout…   102 runs  
4         India  https://www.cricketwa.com/scorecard/23010/indi…  6 wickets  
```

[16]:

```
Match_data.dtypes
```

[16]:

```
Match_no                  int64
Date                     object
Venue            string[python]
Team1            string[python]
Team2            string[python]
Winner           string[python]
Scorecard URL            object
Margin                   object
dtype: object
```

CHECK NULL

```
[17]: Match_data.isna().sum()
```

```
[17]: Match_no         0
      Date             0
      Venue            0
      Team1            0
      Team2            0
      Winner           0
      Scorecard URL    0
      Margin           0
      dtype: int64
```

```
[18]: Match_data.isnull().mean()*100
```

```
[18]: Match_no         0.0
      Date             0.0
      Venue            0.0
      Team1            0.0
      Team2            0.0
      Winner           0.0
      Scorecard URL    0.0
      Margin           0.0
      dtype: float64
```

REMOVE SCORECARD URL AS IT IS NOT NEEDED

```
[19]: Match_data.drop('Scorecard URL',axis=1,inplace=True)
```

```
[20]: Match_data.head()
```

```
[20]:    Match_no        Date       Venue         Team1        Team2  \
      0         1  2023-10-05   Ahmedabad       England  New Zealand
      1         2  2023-10-06   Hyderabad      Pakistan  Netherlands
      2         3  2023-10-07  Dharamsala    Bangladesh  Afghanistan
      3         4  2023-10-07       Delhi  South Africa    Sri Lanka
      4         5  2023-10-08     Chennai         India    Australia

               Winner     Margin
      0   New Zealand  9 wickets
      1      Pakistan    81 runs
      2    Bangladesh  6 wickets
      3  South Africa   102 runs
      4         India  6 wickets
```

Match Analysis(EDA)

TOTAL MATCH

[21]:

```
Match = Match_data['Match_no'].nunique()
print(f"The Number Of Matches Played: {Match}")
```

```
The Number Of Matches Played: 48
```

Venue Where Matches Were Held

[22]:

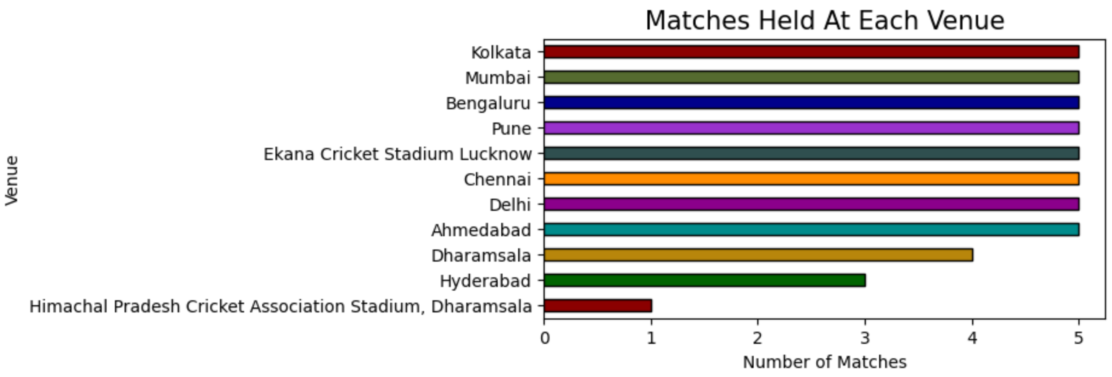
```
venues = Match_data['Venue'].unique()
print('Different Venues Where Matches Were Held:')
for venue in venues:
    print(venue)
```

```
Different Venues Where Matches Were Held:
Ahmedabad
Hyderabad
Dharamsala
Delhi
Chennai
Ekana Cricket Stadium Lucknow
Pune
Bengaluru
Mumbai
Himachal Pradesh Cricket Association Stadium, Dharamsala
Kolkata
```

Matches Held at Each Venue

[23]:

```
venue = Match_data['Venue'].value_counts().sort_values(ascending=True)


colors = ['darkred', 'darkgreen', 'darkgoldenrod', 'darkcyan', 'darkmagenta',
          'darkorange', 'darkslategray', 'darkorchid', 'darkblue',␣
 ↪'darkolivegreen']


plt.figure(figsize=(6, 3))
venue.plot(kind='barh', color=colors, edgecolor='black')
plt.title('Matches Held At Each Venue', color='black', size=15)
plt.ylabel('Venue', color='black', size=10)
plt.xlabel('Number of Matches', color='black', size=10)
plt.show()
```



Teams Participated

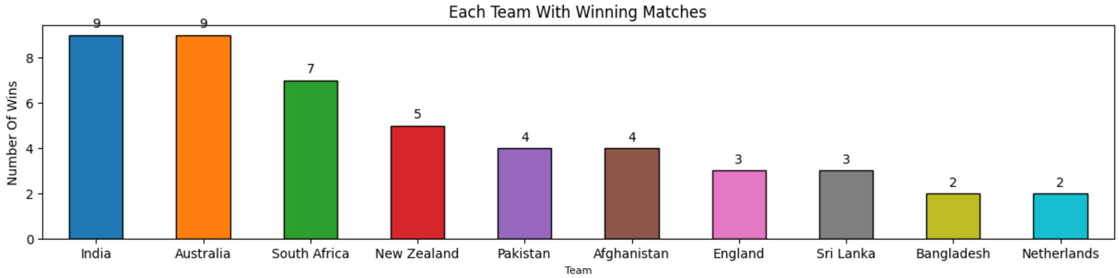
[24]:
```
Teams = Match_data['Winner'].unique()
print('Teams Participated in WorldCup :')
for Team in Teams:
    print(Team)
```

```
Teams Participated in WorldCup :
New Zealand
Pakistan
Bangladesh
South Africa
India
England
Afghanistan
Australia
Netherlands
Sri Lanka
```

Winning Matches

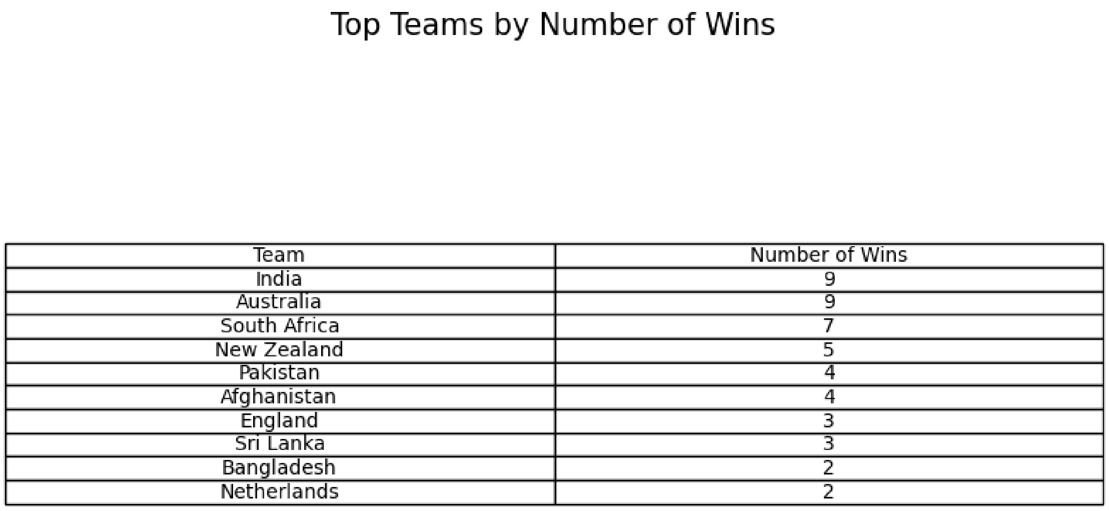
[25]:
```
win_counts = Match_data[Match_data['Winner'] != 'Draw']['Winner'].value_counts()



dark_colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',␣
 ↪'#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf']



plt.figure(figsize=(15, 3))
win_counts.plot(kind='bar', color=dark_colors, edgecolor='black')
plt.title('Each Team With Winning Matches', color='black', size=12)
plt.xlabel('Team', color='black', size=8)
plt.ylabel('Number Of Wins', color='black', size=10)
plt.xticks(rotation=360, color='black')
```

```
for i, v in enumerate(win_counts):
    plt.text(i, v + 0.2, str(v), ha='center', va='bottom', size=10)
plt.show()
```



[112]:

```
fig, ax = plt.subplots(figsize=(10, 6))

ax.axis('off')

table_data = win_counts_table.head(10)

ax.table(cellText=table_data.values, colLabels=table_data.columns,␣
 ↪loc='center', cellLoc='center', fontsize=10)

plt.title('Top Teams by Number of Wins', size=15)

plt.show()
```



Top Teams by Number of Wins

Team	Number of Wins
India	9
Australia	9
South Africa	7
New Zealand	5
Pakistan	4
Afghanistan	4
England	3
Sri Lanka	3
Bangladesh	2
Netherlands	2

[40]:

```
pip install wordcloud
```

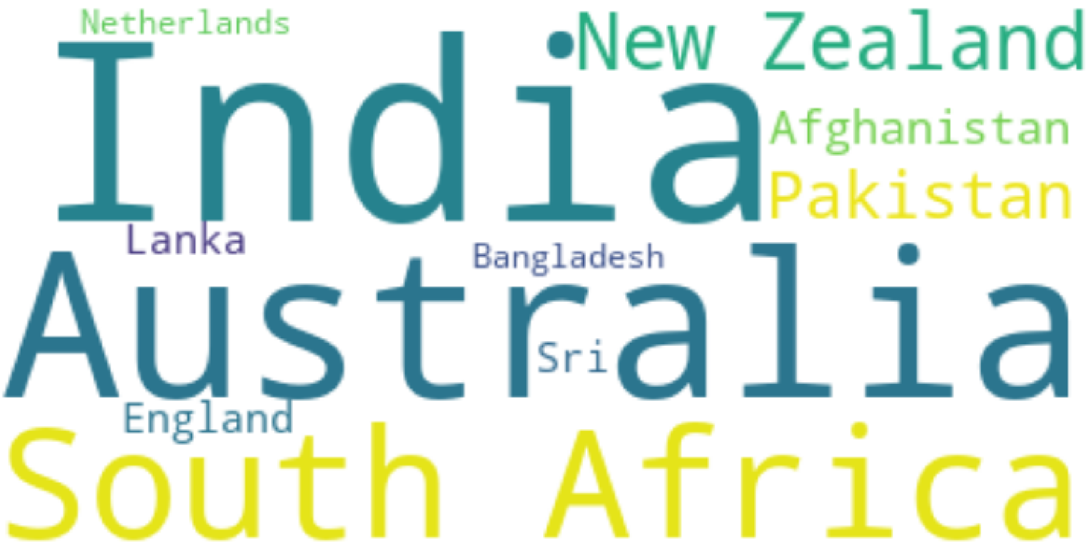
```
Requirement already satisfied: wordcloud in c:\python310\lib\site-packages
(1.9.3)Note: you may need to restart the kernel to use updated packages.
```

```
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\python310\lib\site-packages)
```

```
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\python310\lib\site-packages)

[notice] A new release of pip available: 22.2.2 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip


Requirement already satisfied: pillow in c:\python310\lib\site-packages (from
wordcloud) (10.2.0)
Requirement already satisfied: numpy>=1.6.1 in c:\python310\lib\site-packages
(from wordcloud) (1.26.4)
Requirement already satisfied: matplotlib in c:\python310\lib\site-packages
(from wordcloud) (3.8.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\python310\lib\site-
packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\python310\lib\site-
packages (from matplotlib->wordcloud) (1.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\python310\lib\site-
packages (from matplotlib->wordcloud) (3.1.1)
Requirement already satisfied: packaging>=20.0 in c:\python310\lib\site-packages
(from matplotlib->wordcloud) (23.2)
Requirement already satisfied: fonttools>=4.22.0 in c:\python310\lib\site-
packages (from matplotlib->wordcloud) (4.48.1)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\python310\lib\site-
packages (from matplotlib->wordcloud) (1.4.5)
Requirement already satisfied: cycler>=0.10 in c:\python310\lib\site-packages
(from matplotlib->wordcloud) (0.12.1)
Requirement already satisfied: six>=1.5 in c:\python310\lib\site-packages (from
python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
```

[41]:

```python
from wordcloud import WordCloud
text_data=' '.join(Match_data['Winner'])
wordcloud=WordCloud(contour_color = 'steelblue',background_color='white').
 ↪generate(text_data)
plt.figure(figsize=(10,8))
plt.imshow(wordcloud,interpolation='bilinear')
plt.axis('off')
plt.show()
```



BATTING ANALYSIS

IMPORT DATA

```
[42]: batting_data=pd.read_csv("batting_summary.csv")
```

```
[43]: batting_data.head()
```

```
[43]:    Match_no            Match_Between Team_Innings    Batsman_Name  \
      0         1  England vs New Zealand      England  Jonny Bairstow
      1         1  England vs New Zealand      England     Dawid Malan
      2         1  England vs New Zealand      England        Joe Root
      3         1  England vs New Zealand      England     Harry Brook
      4         1  England vs New Zealand      England       Moeen Ali

         Batting_Position                            Dismissal  Runs  Balls  4s  6s  \
      0                 1  c Daryl Mitchell b Mitchell Santner    33     35   4   1
      1                 2            c Tom Latham b Matt Henry    14     24   2   0
      2                 3                     b Glenn Phillips    77     86   4   1
      3                 4     c Devon Conway b Rachin Ravindra    25     16   4   1
      4                 5                     b Glenn Phillips    11     17   1   0

        Strike_Rate
      0      94.300
      1      58.300
      2      89.500
      3     156.300
      4      64.700
```

CHECK THE NUMBER OF ROWS AND COLUMN

[44]:
```
print("Number of Rows",batting_data.shape[0])
print("Number of Columns",batting_data.shape[1])
```

```
Number of Rows 916
Number of Columns 11
```

CHECK TYPES OF DATA

[45]:
```
batting_data.dtypes
```

[45]:
```
Match_no              int64
Match_Between        object
Team_Innings         object
Batsman_Name         object
Batting_Position      int64
Dismissal            object
Runs                  int64
Balls                 int64
4s                    int64
6s                    int64
Strike_Rate          object
dtype: object
```

CHECK NULL DATA

[46]:
```
batting_data.isnull().sum()
```

[46]:
```
Match_no             0
Match_Between        0
Team_Innings         0
Batsman_Name         0
Batting_Position     0
Dismissal            2
Runs                 0
Balls                0
4s                   0
6s                   0
Strike_Rate          0
dtype: int64
```

Descriptive Statistics for Batting Data

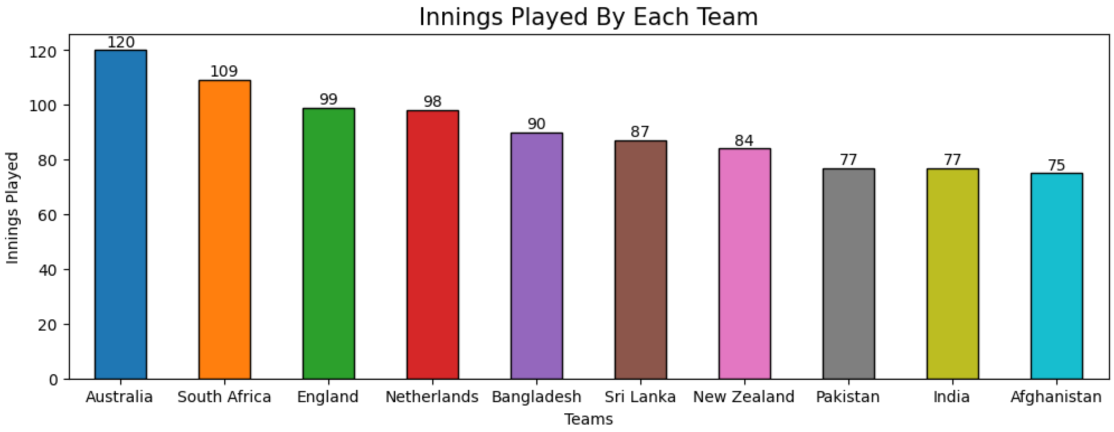
[49]:
```
descriptive_stats = batting_data.describe().T


styled_descriptive_stats = descriptive_stats.style.
  ↪background_gradient(cmap='cool')
```

```
styled_descriptive_stats
```

[49]:
```
<pandas.io.formats.style.Styler at 0x207edcb1000>
```

Innings Played By Each Team

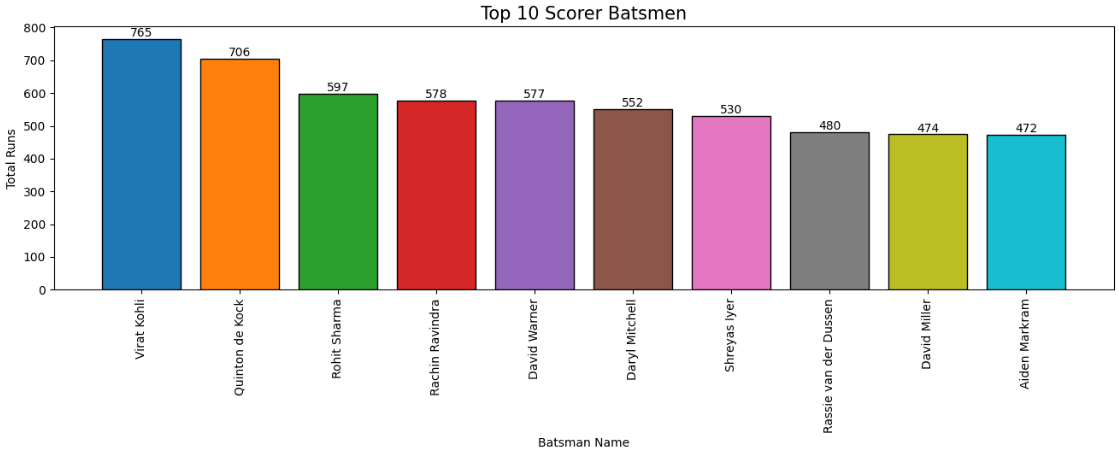
[52]:
```
dark_colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',␣
  ↪'#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf']

innings=batting_data['Team_Innings'].value_counts()
plt.figure(figsize=(12, 4))
innings.plot(kind='bar', color=dark_colors, edgecolor='black')
plt.title('Innings Played By Each Team', color='black', size=15)
plt.xlabel('Teams', size=10, color='black')
plt.ylabel('Innings Played', size=10, color='black')
plt.xticks(rotation=360, color='black')


for i, v in enumerate(innings):
    plt.text(i, v + 0.2, str(v), ha='center', va='bottom', size=10)

plt.show()
```



Top Run-Scorers In The WorldCup

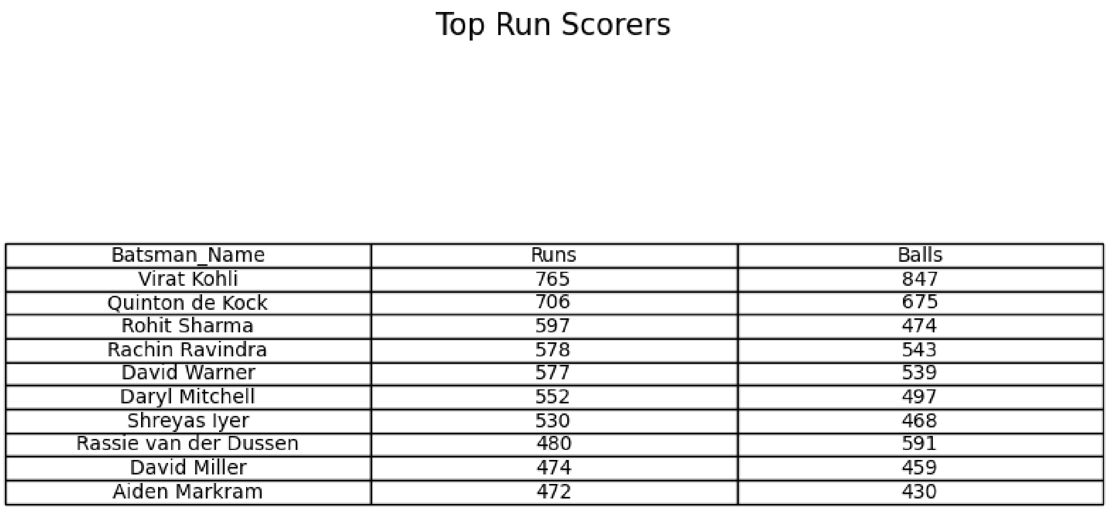
[55]:
```
# Define 10 different dark colors for 10 bars
dark_colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
               '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf']
```

```
top_scorers = batting_data.groupby('Batsman_Name')['Runs'].sum().reset_index().
 ↪sort_values(by='Runs', ascending=False)
plt.figure(figsize=(16, 4))
plt.bar(top_scorers['Batsman_Name'].head(10), top_scorers['Runs'].head(10),
        color=dark_colors, edgecolor='black')
plt.title('Top 10 Scorer Batsmen', color='black', size=15)
plt.xlabel('Batsman Name', size=10, color='black')
plt.ylabel('Total Runs', size=10, color='black')
plt.xticks(rotation=90, color='black')

# Add text annotations on top of each bar
for i, v in enumerate(top_scorers['Runs'].head(10)):
    plt.text(i, v + 0.2, str(v), ha='center', va='bottom', fontsize=10)

plt.show()
```



[111]:

```
fig, ax = plt.subplots(figsize=(10, 6))

ax.axis('off')

table_data = top_run_scorers.round(2)

ax.table(cellText=table_data.values, colLabels=table_data.columns,␣
 ↪loc='center', cellLoc='center', fontsize=10)

plt.title('Top Run Scorers', size=15)

plt.show()

print(top_run_scorers)
```



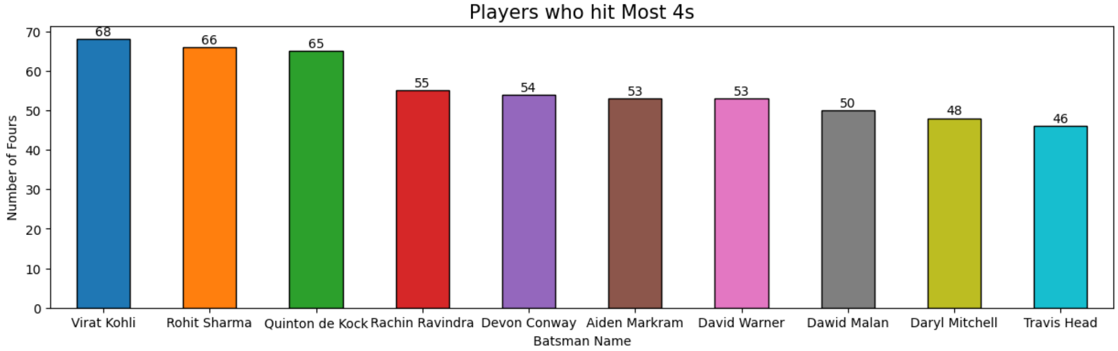
Top Run Scorers

Batsman_Name	Runs	Balls
Virat Kohli	765	847
Quinton de Kock	706	675
Rohit Sharma	597	474
Rachin Ravindra	578	543
David Warner	577	539
Daryl Mitchell	552	497
Shreyas Iyer	530	468
Rassie van der Dussen	480	591
David Miller	474	459
Aiden Markram	472	430

```
              Batsman_Name  Runs  Balls
143            Virat Kohli   765    847
103        Quinton de Kock   706    675
113           Rohit Sharma   597    474
104        Rachin Ravindra   578    543
21            David Warner   577    539
18          Daryl Mitchell   552    497
125           Shreyas Iyer   530    468
108  Rassie van der Dussen   480    591
20            David Miller   474    459
4            Aiden Markram   472    430
```

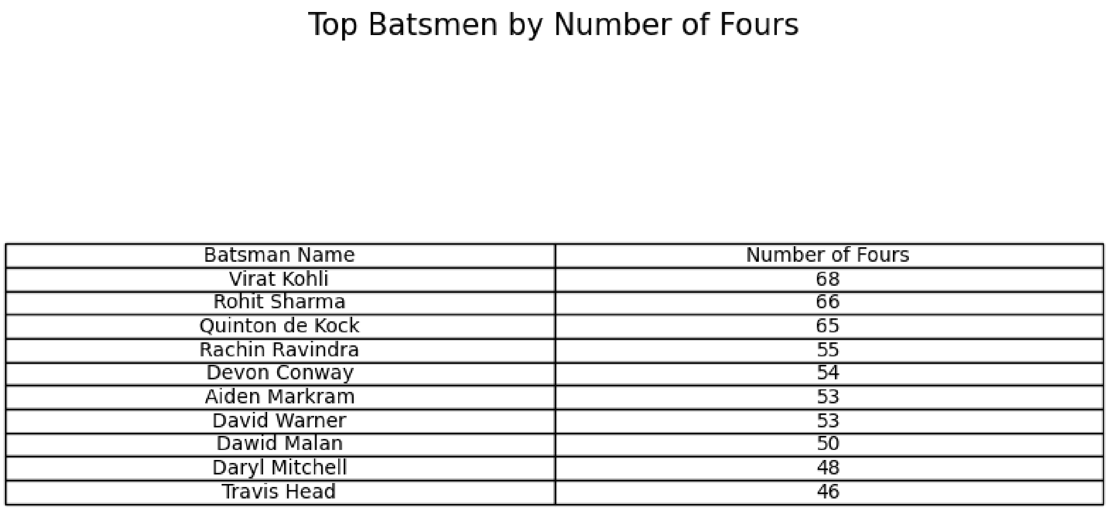
Most 4s By Players

[61]:
```
dark_colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
               '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf']

fours = batting_data[batting_data['4s'] > 0]
total_4s = fours.groupby('Batsman_Name')['4s'].sum().
  ↪sort_values(ascending=False)
plt.figure(figsize=(15, 4))
total_4s.head(10).plot(kind='bar', color=dark_colors, edgecolor='black')
plt.title('Players who hit Most 4s', size=15, color='black')
plt.xlabel('Batsman Name', size=10, color='black')
plt.ylabel('Number of Fours', size=10, color='black')
plt.xticks(rotation=360, color='black')


for i, v in enumerate(total_4s.head(10)):
    plt.text(i, v + 0.2, str(v), ha='center', va='bottom', fontsize=10)

plt.show()
```



[109]:

```
fig, ax = plt.subplots(figsize=(10, 6))

ax.axis('off')

table_data = top_4s_players.round(2)  # Round to 2 decimal places

ax.table(cellText=table_data.values, colLabels=table_data.columns,␣
 ↪loc='center', cellLoc='center', fontsize=10)

plt.title('Top Batsmen by Number of Fours', size=15)

plt.show()

display(top_4s_players)
```



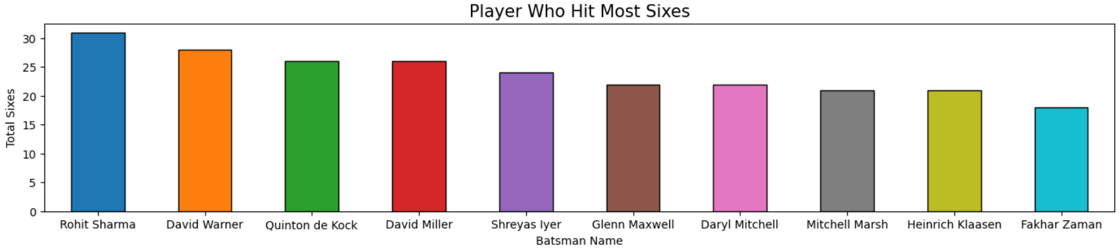
Top Batsmen by Number of Fours

Batsman Name	Number of Fours
Virat Kohli	68
Rohit Sharma	66
Quinton de Kock	65
Rachin Ravindra	55
Devon Conway	54
Aiden Markram	53
David Warner	53
Dawid Malan	50
Daryl Mitchell	48
Travis Head	46

```
      Batsman Name  Number of Fours
0      Virat Kohli               68
1     Rohit Sharma               66
2  Quinton de Kock               65
3  Rachin Ravindra               55
4     Devon Conway               54
5    Aiden Markram               53
6     David Warner               53
7      Dawid Malan               50
8   Daryl Mitchell               48
9      Travis Head               46
```

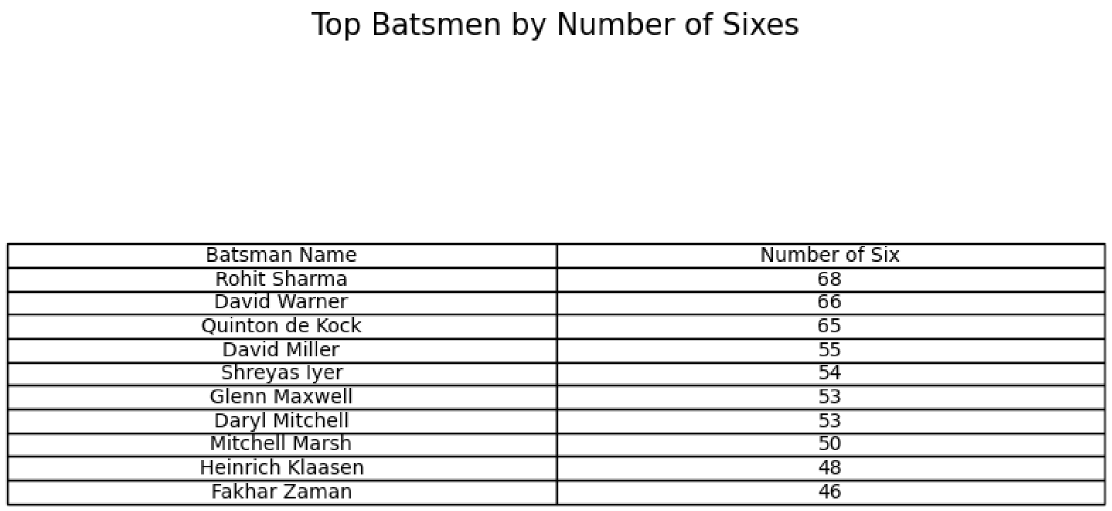
Most 6s By Players

[63]:
```
dark_colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
               '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf']

sixs=batting_data[batting_data['6s']>0]
total_6s=sixs.groupby('Batsman_Name')['6s'].sum().sort_values(ascending=False)
plt.figure(figsize=(17,3))
total_6s.head(10).plot(kind='bar',color=dark_colors,edgecolor='black')
plt.title('Player Who Hit Most Sixes',size=15,color='black')
plt.xlabel('Batsman Name',size=10,color='black')
plt.ylabel('Total Sixes',size=10,color='black')
plt.xticks(rotation=360,color='black')
plt.show()
```



[108]:

```
fig, ax = plt.subplots(figsize=(10, 6))

ax.axis('off')

table_data = top_6s_players.round(2)  # Round to 2 decimal places

ax.table(cellText=table_data.values, colLabels=table_data.columns, loc='center', cellLoc='center', fontsize=10)

plt.title('Top Batsmen by Number of Sixes', size=15)

plt.show()

display(top_6s_players)
```



Top Batsmen by Number of Sixes

Batsman Name	Number of Six
Rohit Sharma	68
David Warner	66
Quinton de Kock	65
David Miller	55
Shreyas Iyer	54
Glenn Maxwell	53
Daryl Mitchell	53
Mitchell Marsh	50
Heinrich Klaasen	48
Fakhar Zaman	46

```
      Batsman Name  Number of Six
0     Rohit Sharma             68
1     David Warner             66
2  Quinton de Kock             65
3      David Miller            55
4      Shreyas Iyer            54
5     Glenn Maxwell            53
6    Daryl Mitchell            53
7    Mitchell Marsh            50
8  Heinrich Klaasen            48
9      Fakhar Zaman            46
```

BOWLING ANALYSIS

IMPORT DATA

[66]:
```
bowling_data=pd.read_csv('bowling_summary.csv')
```

[67]:
```
bowling_data.head(5)
```

[67]:
```
   Match_no          Match_Between Bowling_Team       Bowler_Name  Overs  \
0         1  England vs New Zealand  New Zealand       Trent Boult   10.0
1         1  England vs New Zealand  New Zealand        Matt Henry   10.0
2         1  England vs New Zealand  New Zealand  Mitchell Santner   10.0
3         1  England vs New Zealand  New Zealand     Jimmy Neesham    7.0
4         1  England vs New Zealand  New Zealand   Rachin Ravindra   10.0
```

```
   Maidens  Runs  Wickets  Economy
0        1    48        1      4.8
1        1    48        3      4.8
2        0    37        2      3.7
3        0    56        0      8.0
4        0    76        1      7.6
```

[68]:
```
bowling_data.sample(5)
```

[68]:
```
     Match_no                Match_Between  Bowling_Team     Bowler_Name  \
288        24     Australia vs Netherlands     Australia     Adam Zampa
532        45         India vs Netherlands         India  Mohammed Shami
382        32  South Africa vs New Zealand  South Africa  Gerald Coetzee
272        23   South Africa vs Bangladesh  South Africa    Marco Jansen
69          6   New Zealand vs Netherlands   New Zealand      Matt Henry

     Overs  Maidens  Runs  Wickets  Economy
288    3.0        0     8        4    2.667
532    6.0        0    41        0    6.833
382    6.3        0    41        2    6.308
272    8.0        0    39        2    4.875
69     8.3        0    40        3    4.706
```

TOTAL NUMBER OF ROWS AND COLUMNS

[69]:
```
print('Number Of Rows',bowling_data.shape[0])
print('Number of Columns',bowling_data.shape[1])
```

```
Number Of Rows 574
Number of Columns 9
```

TYPES OF DATA

[70]:
```
bowling_data.dtypes
```

[70]:
```
Match_no           int64
Match_Between     object
Bowling_Team      object
Bowler_Name       object
Overs            float64
Maidens            int64
Runs               int64
Wickets            int64
Economy          float64
dtype: object
```

FIND NUMBER OF NULL

[71]:

```
bowling_data.isna().sum()
```

[71]:

```
Match_no         0
Match_Between    0
Bowling_Team     0
Bowler_Name      0
Overs            0
Maidens          0
Runs             0
Wickets          0
Economy          0
dtype: int64
```

[72]:

```
bowling_data.isnull().mean()*100
```

[72]:

```
Match_no         0.0
Match_Between    0.0
Bowling_Team     0.0
Bowler_Name      0.0
Overs            0.0
Maidens          0.0
Runs             0.0
Wickets          0.0
Economy          0.0
dtype: float64
```

Descriptive Statistics for Bowling Data

[73]:

```
descriptive_stats = bowling_data.describe().T



styled_descriptive_stats = descriptive_stats.style.
 ↪background_gradient(cmap='cool')



styled_descriptive_stats
```

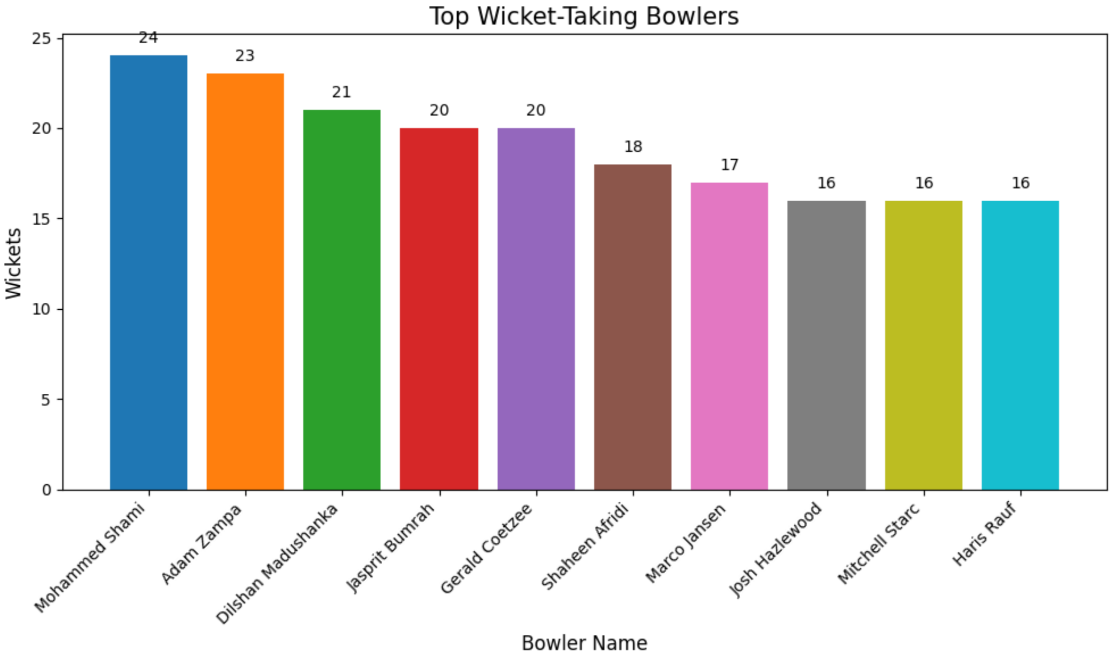
[73]:

```
<pandas.io.formats.style.Styler at 0x207eeec86d0>
```

Most Wicket Takens

[75]:

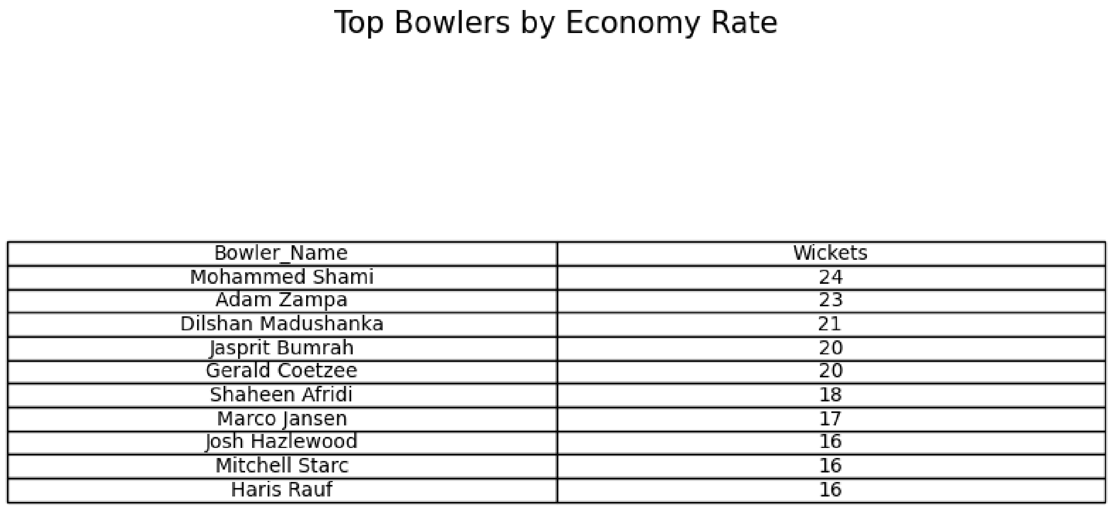
```
dark_colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
               '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf']

top_wickets=bowling_data.groupby('Bowler_Name')['Wickets'].sum().reset_index().
 ↪sort_values(by='Wickets', ascending=False)
plt.figure(figsize=(10, 6))
plt.bar(top_wickets['Bowler_Name'].head(10), top_wickets['Wickets'].head(10),␣
 ↪color=dark_colors)
plt.title('Top Wicket-Taking Bowlers', size=15)
plt.xlabel('Bowler Name', size=12)
plt.ylabel('Wickets', size=12)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

for i, v in enumerate(top_wickets['Wickets'].head(10)):
    plt.text(i, v + 0.5, str(v), ha='center', va='bottom', fontsize=10)

plt.show()
```



[102]:

```
fig, ax = plt.subplots(figsize=(10, 6))


ax.axis('off')


table_data = top_wickets.head(10).round(2)


ax.table(cellText=table_data.values, colLabels=table_data.columns,␣
 ↪loc='center', cellLoc='center', fontsize=10)
```

```python
plt.title('Top Bowlers by Economy Rate', size=15)

plt.show()
```

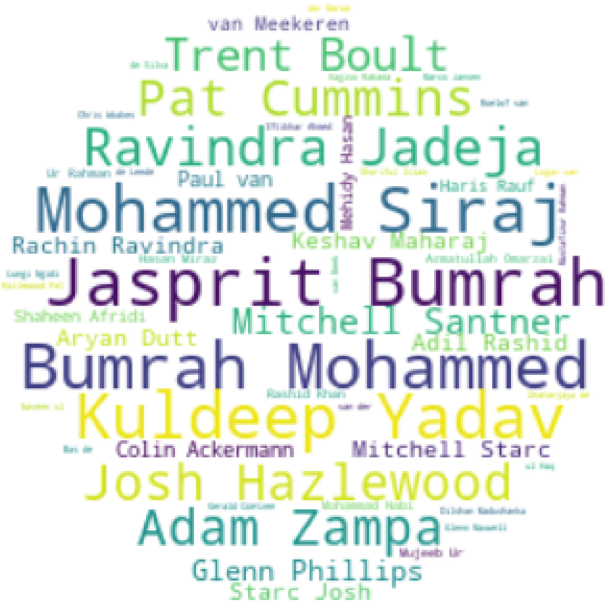


Top Bowlers by Economy Rate

Bowler_Name	Wickets
Mohammed Shami	24
Adam Zampa	23
Dilshan Madushanka	21
Jasprit Bumrah	20
Gerald Coetzee	20
Shaheen Afridi	18
Marco Jansen	17
Josh Hazlewood	16
Mitchell Starc	16
Haris Rauf	16

[89]:

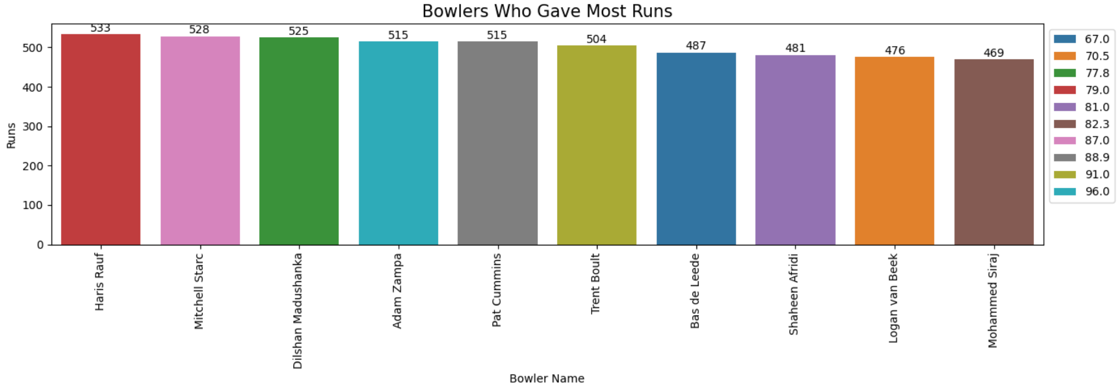
```python
from wordcloud import WordCloud


text_data = ' '.join(bowling_data['Bowler_Name'])

# Circle mask
x, y = np.ogrid[:300, :300]
mask = (x - 150) ** 2 + (y - 150) ** 2 > 130 ** 2
mask = 255 * mask.astype(int)

wc = WordCloud(background_color="white", repeat=True, mask=mask)
wc.generate(text_data)

plt.axis("off")

# Set the image extent to center the word cloud
plt.imshow(wc, interpolation="bilinear", extent=[-150, 150, -150, 150])

plt.show()
```



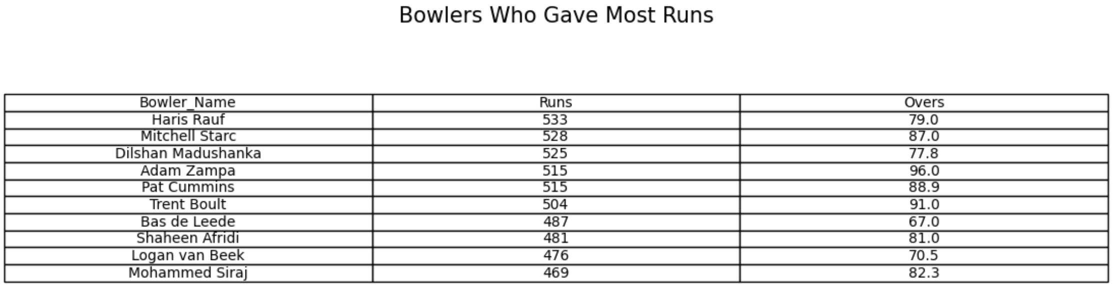
BOWLERS WHO GAVE MOST RUNS

```
[90]: dark_colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
                     '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf']

      most_run = bowling_data.groupby('Bowler_Name')['Runs'].sum().reset_index().
       ↪sort_values(by='Runs', ascending=False)
      total_overs = bowling_data.groupby('Bowler_Name')['Overs'].sum().reset_index()

      merge_data = pd.merge(most_run, total_overs, on='Bowler_Name')

      plt.figure(figsize=(14, 5))
      sns.barplot(x='Bowler_Name', y='Runs', data=merge_data.head(10), hue='Overs',␣
       ↪dodge=False, palette=dark_colors)

      plt.legend(bbox_to_anchor=(1, 1), loc='upper left')
      plt.title('Bowlers Who Gave Most Runs', size=15, color='black')
      plt.xlabel('Bowler Name', size=10, color='black')

      for i, v in enumerate(merge_data['Runs'].head(10)):
          plt.text(i, v + 0.5, str(v), ha='center', va='bottom', fontsize=10)

      plt.xticks(rotation=90, color='black')
      plt.tight_layout()
```

```python
plt.show()
```



```python
[91]: fig, ax = plt.subplots(figsize=(14, 4))


ax.axis('off')


table_data = merge_data.head(10)


ax.table(cellText=table_data.values, colLabels=table_data.columns,
 ↪loc='center', cellLoc='center', fontsize=10)


plt.title('Bowlers Who Gave Most Runs', size=15, color='black')

plt.show()
```



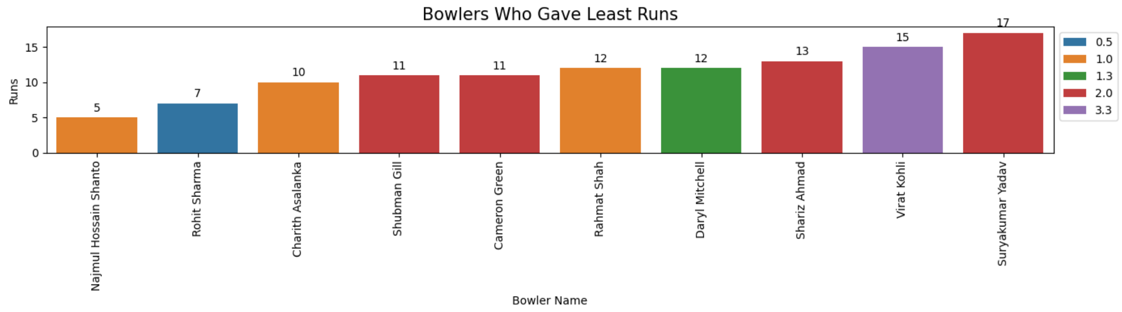
Bowlers Who Gave Most Runs

Bowler_Name	Runs	Overs
Haris Rauf	533	79.0
Mitchell Starc	528	87.0
Dilshan Madushanka	525	77.8
Adam Zampa	515	96.0
Pat Cummins	515	88.9
Trent Boult	504	91.0
Bas de Leede	487	67.0
Shaheen Afridi	481	81.0
Logan van Beek	476	70.5
Mohammed Siraj	469	82.3

Bowler Who Gave Least Wicket

[92]:

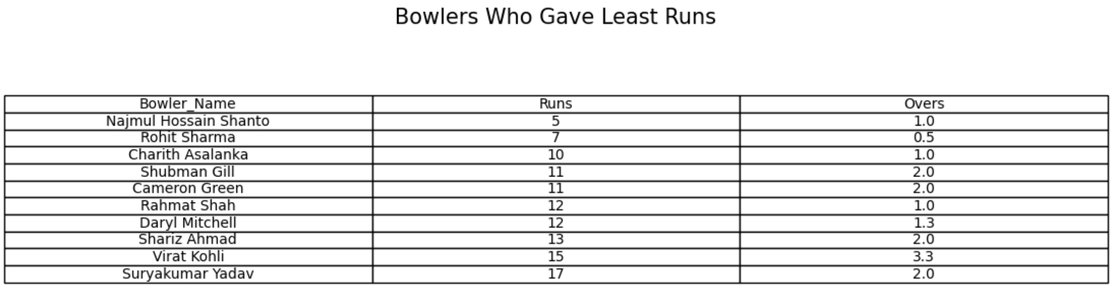
```
dark_colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
               '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf']

less_run = bowling_data.groupby('Bowler_Name')['Runs'].sum().reset_index().
 ↪sort_values(by='Runs', ascending=True)
total_overs = bowling_data.groupby('Bowler_Name')['Overs'].sum().reset_index()

merge_data = pd.merge(less_run, total_overs, on='Bowler_Name')

plt.figure(figsize=(14, 4))
sns.barplot(x='Bowler_Name', y='Runs', data=merge_data.head(10), hue='Overs',␣
 ↪dodge=False, palette=dark_colors)

plt.legend(bbox_to_anchor=(1, 1), loc='upper left')
plt.title('Bowlers Who Gave Least Runs', size=15, color='black')  # Updated␣
 ↪title
plt.xlabel('Bowler Name', size=10, color='black')

for i, v in enumerate(merge_data['Runs'].head(10)):
    plt.text(i, v + 0.5, str(v), ha='center', va='bottom', fontsize=10)

plt.xticks(rotation=90, color='black')
plt.tight_layout()
plt.show()
```



[94]:

```
fig, ax = plt.subplots(figsize=(14, 4))

# Hide the axes
ax.axis('off')

# Create the table data
table_data = merge_data.head(10)

# Plot the table
```

```
ax.table(cellText=table_data.values, colLabels=table_data.columns,␣
↪loc='center', cellLoc='center', fontsize=10)

# Add a title to the table
plt.title('Bowlers Who Gave Least Runs', size=15, color='black')

plt.show()
```

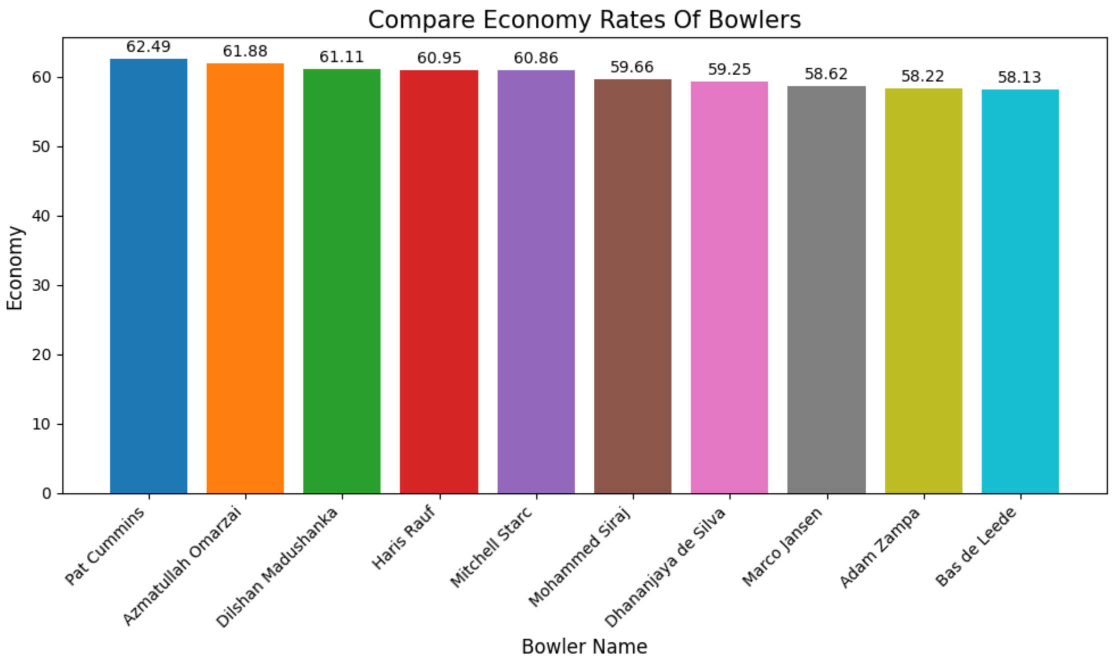


Bowlers Who Gave Least Runs

Bowler_Name	Runs	Overs
Najmul Hossain Shanto	5	1.0
Rohit Sharma	7	0.5
Charith Asalanka	10	1.0
Shubman Gill	11	2.0
Cameron Green	11	2.0
Rahmat Shah	12	1.0
Daryl Mitchell	12	1.3
Shariz Ahmad	13	2.0
Virat Kohli	15	3.3
Suryakumar Yadav	17	2.0

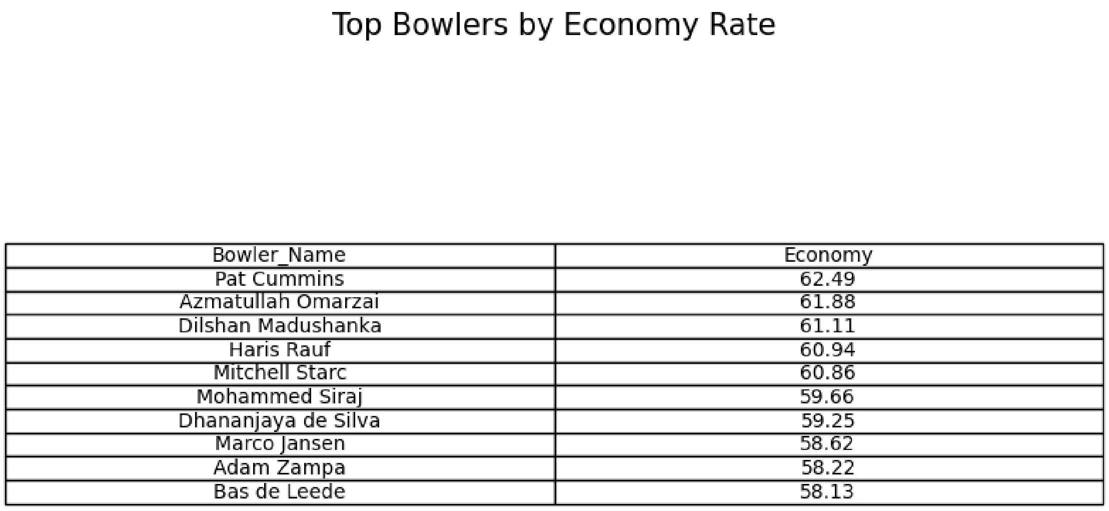
ECONOMY BOWLERS

```
[95]: dark_colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
                     '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf']



      economy_bowler = bowling_data.groupby('Bowler_Name')['Economy'].sum().
       ↪reset_index().sort_values(by='Economy', ascending=False)
      plt.figure(figsize=(10, 6))
      plt.bar(economy_bowler['Bowler_Name'].head(10), economy_bowler['Economy'].
       ↪head(10), color=dark_colors)
      plt.title('Compare Economy Rates Of Bowlers', size=15)
      plt.xlabel('Bowler Name', size=12)
      plt.ylabel('Economy', size=12)
      plt.xticks(rotation=45, ha='right')
      plt.tight_layout()

      # Add text annotations on top of each bar
      for i, v in enumerate(economy_bowler['Economy'].head(10)):
          plt.text(i, v + 0.5, f'{v:.2f}', ha='center', va='bottom', fontsize=10)

      plt.show()
```



[100]:

```
fig, ax = plt.subplots(figsize=(10, 6))


ax.axis('off')


table_data = economy_bowler.head(10).round(2)


ax.table(cellText=table_data.values, colLabels=table_data.columns, loc='center', cellLoc='center', fontsize=10)


plt.title('Top Bowlers by Economy Rate', size=15)

plt.show()
```

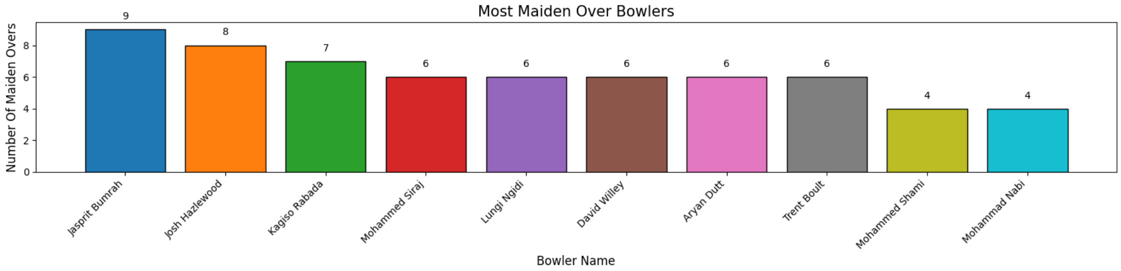


Top Bowlers by Economy Rate

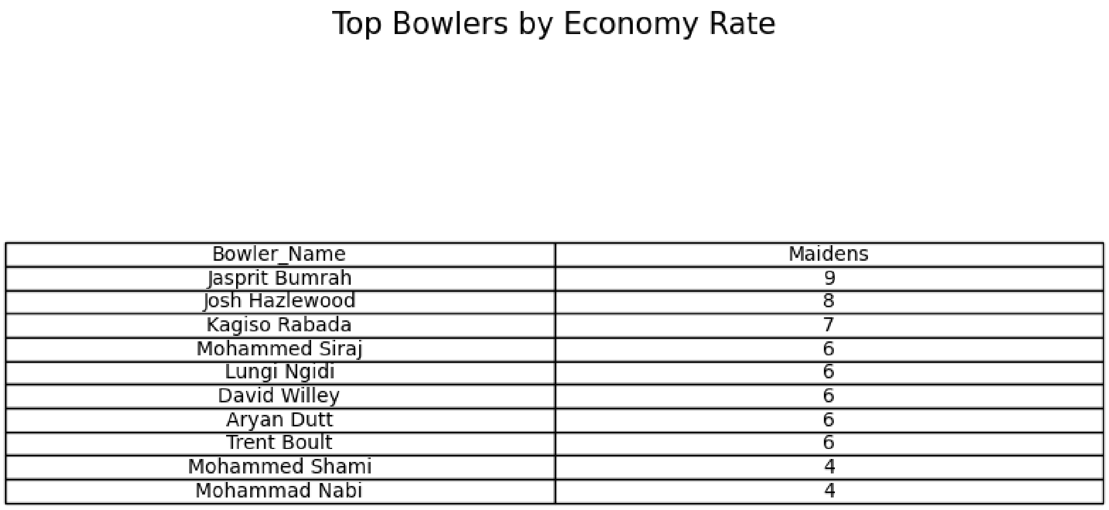
Bowler_Name	Economy
Pat Cummins	62.49
Azmatullah Omarzai	61.88
Dilshan Madushanka	61.11
Haris Rauf	60.94
Mitchell Starc	60.86
Mohammed Siraj	59.66
Dhananjaya de Silva	59.25
Marco Jansen	58.62
Adam Zampa	58.22
Bas de Leede	58.13

Maidens By Bowlers

[98]:
```
dark_colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
               '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf']

maidens=bowling_data.groupby('Bowler_Name')['Maidens'].sum().reset_index().
  ↪sort_values(by='Maidens',ascending=False)
plt.figure(figsize=(16,4))
plt.bar(maidens['Bowler_Name'].head(10),maidens['Maidens'].
  ↪head(10),color=dark_colors,edgecolor='black')
plt.title('Most Maiden Over Bowlers', size=15)
plt.xlabel('Bowler Name', size=12)
plt.ylabel('Number Of Maiden Overs', size=12)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

# Add text annotations on top of each bar
for i, v in enumerate(maidens['Maidens'].head(10)):
    plt.text(i, v + 0.5, str(v), ha='center', va='bottom', fontsize=10)

plt.show()
```



[101]:
```
fig, ax = plt.subplots(figsize=(10, 6))


ax.axis('off')


table_data = maidens.head(10).round(2)


ax.table(cellText=table_data.values, colLabels=table_data.columns, loc='center', cellLoc='center', fontsize=10)


plt.title('Top Bowlers by Economy Rate', size=15)

plt.show()
```



Top Bowlers by Economy Rate

Bowler_Name	Maidens
Jasprit Bumrah	9
Josh Hazlewood	8
Kagiso Rabada	7
Mohammed Siraj	6
Lungi Ngidi	6
David Willey	6
Aryan Dutt	6
Trent Boult	6
Mohammed Shami	4
Mohammad Nabi	4

[ ]:

```
```