PROGRAM 6: PANDAS IN PYTHON

Requirement:

```
    After the death of Mr. Jagmohan Dalmia BCCI is worried about the performance of Indian team in the worldcup
 as they didn’t qualify for the finals in the last season.They believe that legendary cricketer, Prince of Kolka
ta, Mr. Sourav Ganguly (Dada) can do some magic to bring the worldcup back in India. BCCI appointed him as the35
th Chairman. Dada feels that there will be a cut throat competition in coming worldcup. He wants you, the future
data analysts to help him perform criclytics by analyzing the data and come up with patterns which will help us
 in bring the cup to home this time. I am sure you will do the needful. ALL THE BEST!
```

Note: You need to make use of Pandas to perform the CRICLYTICS on the given cricket dataset and come up with potential patterns.

CRICKET WORLD CUP 2023

The 2023 Men's ICC Cricket World Cup will be the 13th edition of the ICC Men's Cricket World Cup, scheduled to be hosted by India during October and November 2023.

This will be the first time the competition is held completely in India.

NOTE: The analysis is done on previous year's 2019 dataset cosidering the venue of Cricket World Cup 2023

In [1]:

```python
import pandas as pd
import plotly.graph_objects as go
import plotly.express as px
import plotly.io as pio
import seaborn as sns
import matplotlib.pyplot as plt
```

IMPORTING DATASETS

In [2]:

```python
batsman_data      = pd.read_csv('Batsman_Data.csv')
bowlers_data      = pd.read_csv('Bowler_data.csv')
ground_avg        = pd.read_csv('Ground_Averages.csv')
odi_match_results = pd.read_csv('ODI_Match_Results.csv')
odi_match_totals  = pd.read_csv('ODI_Match_Totals.csv')
wc_players        = pd.read_csv('WC_players.csv')
```

LIST OF INDIAN PLAYERS

In [3]:

```python
df = wc_players[(wc_players['Country']=='India')]
df
```

Out[3]:

	Player	ID	Country
61	Virat Kohli (c)	253802	India
62	Rohit Sharma (vc)	34102	India
63	Shikhar Dhawan	28235	India
64	K. L. Rahul	422108	India
65	Vijay Shankar	477021	India
66	MS Dhoni (wk)	28081	India
67	Kedar Jadhav	290716	India
68	Dinesh Karthik	30045	India
69	Yuzvendra Chahal	430246	India
70	Kuldeep Yadav	559235	India
71	Bhuvneshwar Kumar	326016	India
72	Jasprit Bumrah	625383	India
73	Hardik Pandya	625371	India
74	Ravindra Jadeja	234675	India
75	Mohammed Shami	481896	India

BOWLING PERFORMANCE

In [4]:
```
IndianPlayers_ID = [253802, 34102, 28235, 422108, 477021, 28081, 290716,
                    30045, 430246,559235, 326016, 625383, 625371, 234675, 481896]
indianbowler_id = bowlers_data[(bowlers_data['Player_ID'].isin(IndianPlayers_ID))]

IndianGround = ['Nagpur','Kolkata','Delhi','Dharamsala','Visakhapatnam','Guwahati','Thiruvananthapuram','Chennai',
                'Cuttack','Ahmedabad','Indore','Kochi','Pune','Mumbai (BS)','Kanpur','Rajkot','Jaipur','Mumbai',
                'Hyderabad (Deccan)','Ranchi','Hyderabad (Sind)','Bengaluru','Chandigarh','Dehradun']
indianground = bowlers_data[(bowlers_data['Ground'].isin(IndianGround))]

overs_notnull = bowlers_data[(bowlers_data['Overs'] != '-')]

opp_india = bowlers_data[(bowlers_data['Opposition'] == 'v India')]

wkts_notzero = bowlers_data[(bowlers_data['Wkts']!= 0)]
```

Indian Bolwers

In [5]:
```
b1 = indianbowler_id.iloc[:]
b2 = indianground.iloc[:]
b3 = overs_notnull.iloc[:]

BO1 = pd.merge(b1,b2,how='inner')
Indian_Bowler_Stats = pd.merge(BO1,b3,how='inner')

Indian_Bowler_Stats
```

Out[5]:

	Unnamed: 0	Overs	Mdns	Runs	Wkts	Econ	Ave	SR	Opposition	Ground	Start Date	Match_ID	Bowler	Player_ID
0	5321	9.0	4	23	1	2.55	23.00	54.0	v Pakistan	Delhi	6 Jan 2013	ODI # 3316	Mohammed Shami	481896
1	5322	4.0	1	24	1	6.00	24.00	24.0	v England	Kochi	15 Jan 2013	ODI # 3320	Mohammed Shami	481896
2	5323	8.0	0	23	1	2.87	23.00	48.0	v England	Ranchi	19 Jan 2013	ODI # 3322	Mohammed Shami	481896
3	5325	9.0	1	46	1	5.11	46.00	54.0	v England	Dharamsala	27 Jan 2013	ODI # 3329	Mohammed Shami	481896
4	5332	8.0	1	42	3	5.25	14.00	16.0	v Australia	Ranchi	23 Oct 2013	ODI # 3422	Mohammed Shami	481896
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
200	6806	2.0	0	20	0	10.00	-	-	v West Indies	Delhi	11 Oct 2014	ODI # 3533	Virat Kohli	253802
201	6807	1.0	0	14	0	14.00	-	-	v West Indies	Dharamsala	17 Oct 2014	ODI # 3535	Virat Kohli	253802
202	6808	0.5	0	6	0	7.20	-	-	v Sri Lanka	Cuttack	2 Nov 2014	ODI # 3539	Virat Kohli	253802
203	6828	0.2	0	1	0	3.00	-	-	v South Africa	Kanpur	11 Oct 2015	ODI # 3689	Virat Kohli	253802
204	6832	2.0	0	14	0	7.00	-	-	v South Africa	Mumbai	25 Oct 2015	ODI # 3700	Virat Kohli	253802

205 rows × 14 columns

In [6]:

```
fig = go.Figure(go.Bar(
    x = [Indian_Bowler_Stats['Opposition'],Indian_Bowler_Stats['Wkts']],
    y = Indian_Bowler_Stats['Bowler']
))

fig.update_yaxes(
    title = 'INDIAN BOWLERS & GROUND',
    showgrid=True,
)
fig.update_layout(
    title = 'WICKETS OVER INDIAN GROUND'
)
fig.show()
```

In [7]:

```
fig = px.line(Indian_Bowler_Stats,
              x = 'Econ',
              y = 'Bowler',
              title='INDIAN BOLWER ECONOMIC RATE')
fig.show()
```

Bowlers Against Indian Team :Overseas Bowlers

In [8]:

```
d1 = opp_india.iloc[:]
d2 = overs_notnull.iloc[:]
d3 = wkts_notzero.iloc[:]
d4 = indianground.iloc[:]

D1 = pd.merge(d1,d2,how='inner')
D2 = pd.merge(D1,d3,how='inner')
Overseas_Bowler_Stats = pd.merge(D2,d4,how='inner')

Overseas_Bowler_Stats
```

Out[8]:

	Unnamed: 0	Overs	Mdns	Runs	Wkts	Econ	Ave	SR	Opposition	Ground	Start Date	Match_ID	Bowler	Player_ID
0	1	8.0	0	57	0	7.12	-	-	v India	Nagpur	18 Dec 2009	ODI # 2933	Suranga Lakmal	49619
1	2	10.0	0	55	2	5.50	27.50	30.0	v India	Kolkata	24 Dec 2009	ODI # 2935	Suranga Lakmal	49619
2	68	10.0	4	13	4	1.30	3.25	15.0	v India	Dharamsala	10 Dec 2017	ODI # 3939	Suranga Lakmal	49619
3	70	5.0	2	20	0	4.00	-	-	v India	Visakhapatnam	17 Dec 2017	ODI # 3942	Suranga Lakmal	49619
4	82	9.0	0	83	1	9.22	83.00	54.0	v India	Guwahati	21 Oct 2018	ODI # 4056	Oshane Thomas	914567
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
226	10185	0.4	0	2	1	3.00	2.00	4.0	v India	Pune	13 Oct 2013	ODI # 3419	Aaron Finch	5334
227	10189	4.0	0	20	1	5.00	20.00	24.0	v India	Nagpur	30 Oct 2013	ODI # 3424	Aaron Finch	5334
228	10190	1.0	0	2	0	2.00	-	-	v India	Bengaluru	2 Nov 2013	ODI # 3428	Aaron Finch	5334
229	10256	0.1	0	1	0	6.00	-	-	v India	Bengaluru	28 Sep 2017	ODI # 3917	Aaron Finch	5334
230	10257	3.0	0	17	0	5.66	-	-	v India	Nagpur	1 Oct 2017	ODI # 3919	Aaron Finch	5334

231 rows × 14 columns

In [9]:

```
fig = go.Figure(go.Bar(
    x = Overseas_Bowler_Stats['Bowler'],
    y = [Overseas_Bowler_Stats['Wkts'],Overseas_Bowler_Stats['Ground']],
))

fig.update_xaxes( title = 'Overseas Bowler')
fig.update_yaxes( title = 'Wickets Per Match')
fig.update_layout( title = '<b>OVERSEAS BOWLERS WICKETS RATE TOWARDS INDIAN GROUND<b>')
fig.show()
```

In [10]:

```
fig = go.Figure()
fig.add_trace(
    go.Scatter(
        x=Overseas_Bowler_Stats['Bowler'],
        y=Overseas_Bowler_Stats['Mdns'],
        name="Maiden Over"
    ))

fig.add_trace(
    go.Bar(
        x=Overseas_Bowler_Stats['Bowler'],
        y=Overseas_Bowler_Stats['Wkts'],
        name="Wickets"
    ))
fig.update_layout(
    title = 'DIFF ON WICKETS & MAIDEN'
)
fig.update_yaxes(
    title = 'WICKETS & MAIDEN'
)
fig.show()
```

BATTING PERFORMANCE

In [11]:

```
IndianPlayers_ID = [253802, 34102, 28235, 422108, 477021, 28081, 290716, 30045,
                    430246, 559235, 326016, 625383, 625371, 234675, 481896]
indianbatter_id = batsman_data[(batsman_data['Player_ID'].isin(IndianPlayers_ID))]

IndianGround = ['Nagpur','Kolkata','Delhi','Dharamsala','Visakhapatnam','Guwahati','Thiruvananthapuram','Chennai',
                'Cuttack','Ahmedabad','Indore','Kochi','Pune','Mumbai (BS)','Kanpur','Rajkot','Jaipur','Mumbai',
                'Hyderabad (Deccan)','Ranchi','Hyderabad (Sind)','Bengaluru','Chandigarh','Dehradun']
indianground = batsman_data[(batsman_data['Ground'].isin(IndianGround))]

Bat1=['DNB','TDNB','absent','sub']
bat1_notnull = batsman_data[(~batsman_data['Bat1'].isin(Bat1))]

opp_india = batsman_data[(batsman_data['Opposition'] == 'v India')]
```

Indian Batsman

In [12]:

```
ba1 = indianbatter_id.iloc[:]
ba2 = indianground.iloc[:]
ba3 = bat1_notnull.iloc[:]

BA1 = pd.merge(ba1,ba2,how='inner')
Batsman_indian = pd.merge(BA1,ba3,how='inner')

Batsman_indian
```

Out[12]:

	Unnamed: 0	Bat1	Runs	BF	SR	4s	6s	Opposition	Ground	Start Date	Match_ID	Batsman	Player_ID
0	5331	0*	0	5	0.00	0	0	v Pakistan	Delhi	6 Jan 2013	ODI # 3316	Mohammed Shami	481896
1	5335	1	1	6	16.66	0	0	v England	Dharamsala	27 Jan 2013	ODI # 3329	Mohammed Shami	481896
2	5364	19	19	17	111.76	2	1	v West Indies	Kochi	8 Oct 2014	ODI # 3531	Mohammed Shami	481896
3	5365	1*	1	1	100.00	0	0	v West Indies	Delhi	11 Oct 2014	ODI # 3533	Mohammed Shami	481896
4	5380	6*	6	6	100.00	1	0	v Australia	Bengaluru	28 Sep 2017	ODI # 3917	Mohammed Shami	481896
...	...	...	...	...	...	...	...	...	...	...	...	...	...
390	6892	33*	33	29	113.79	6	0	v West Indies	Thiruvananthapuram	1 Nov 2018	ODI # 4064	Virat Kohli	253802
391	6899	44	44	45	97.77	6	1	v Australia	Hyderabad (Deccan)	2 Mar 2019	ODI # 4102	Virat Kohli	253802
392	6900	116	116	120	96.66	10	0	v Australia	Nagpur	5 Mar 2019	ODI # 4106	Virat Kohli	253802
393	6901	123	123	95	129.47	16	1	v Australia	Ranchi	8 Mar 2019	ODI # 4109	Virat Kohli	253802
394	6903	20	20	22	90.90	2	0	v Australia	Delhi	13 Mar 2019	ODI # 4113	Virat Kohli	253802

395 rows × 13 columns

In [13]:

```
fig = go.Figure(go.Bar(
    x = Batsman_indian['Batsman'],
    y = [Batsman_indian['Opposition'],Batsman_indian['SR']],
))

fig.update_xaxes( title = 'Indian Players')
fig.update_yaxes( title = 'Opposition Teams')
fig.update_layout(title = '<b>INDIAN BATSMAN STRIKE RATE WITH OPPOSITION TEAMS<b>')
fig.show()
```

In [14]:

```
subject = Batsman_indian["Batsman"]
score = Batsman_indian["Runs"]
aggs = ["count","sum","avg","median","stddev","min","max"]

agg = []
agg_func = []
for i in range(0, len(aggs)):
    agg = dict(
        args=['transforms[0].aggregations[0].func', aggs[i]],
        label=aggs[i],
        method='restyle'
    )
    agg_func.append(agg)

data = [dict(
  type = 'bar',
  x = subject,
  y = score,
  mode = 'markers',
  marker=dict(size=[40, 60, 80, 100],
              color=[0,1,2,3]),
  transforms = [dict(
    type = 'aggregate',
    groups = subject,
    aggregations = [dict(
        target = 'y', func = 'sum', enabled = True)
    ]
  )]
)]

layout = dict(
  title = '<b>INDIAN BATSMAN STATISTICAL RECORDS</b><br>Descriptive statistical dropdown menu',
  xaxis = dict(title = 'Indian players'),
  yaxis = dict(title = 'Score statistics', range = [0,190]),
  updatemenus = [dict(
        x = 0.85,
        y = 1.15,
        xref = 'paper',
        yref = 'paper',
        yanchor = 'top',
        active = 1,
        showactive = True,
        buttons = agg_func
  )]
)

fig_dict = dict(data=data, layout=layout)

pio.show(fig_dict, validate=False)
```

In [15]:

```
import plotly.express as px

fig = px.bar(Batsman_indian, y='Opposition', x="Batsman", color='Opposition', orientation="v", hover_name="SR",
             #color_discrete_map={
                #"v Australia": "red",
                #"v Pakistan": "green",
                #"v Bangladesh": "blue"},

             title="Indian Batsmans Strike Rate [SR]"
            )

fig.show()
```

Batsman against Indian team : Overseas Batsman

In [16]:

```
da1 = opp_india.iloc[:]
da2 = indianground.iloc[:]
da3 = bat1_notnull.iloc[:]

DA1 = pd.merge(da1,da2,how='inner')
Batsman_Against_India = pd.merge(DA1,da3,how='inner')

Batsman_Against_India
```

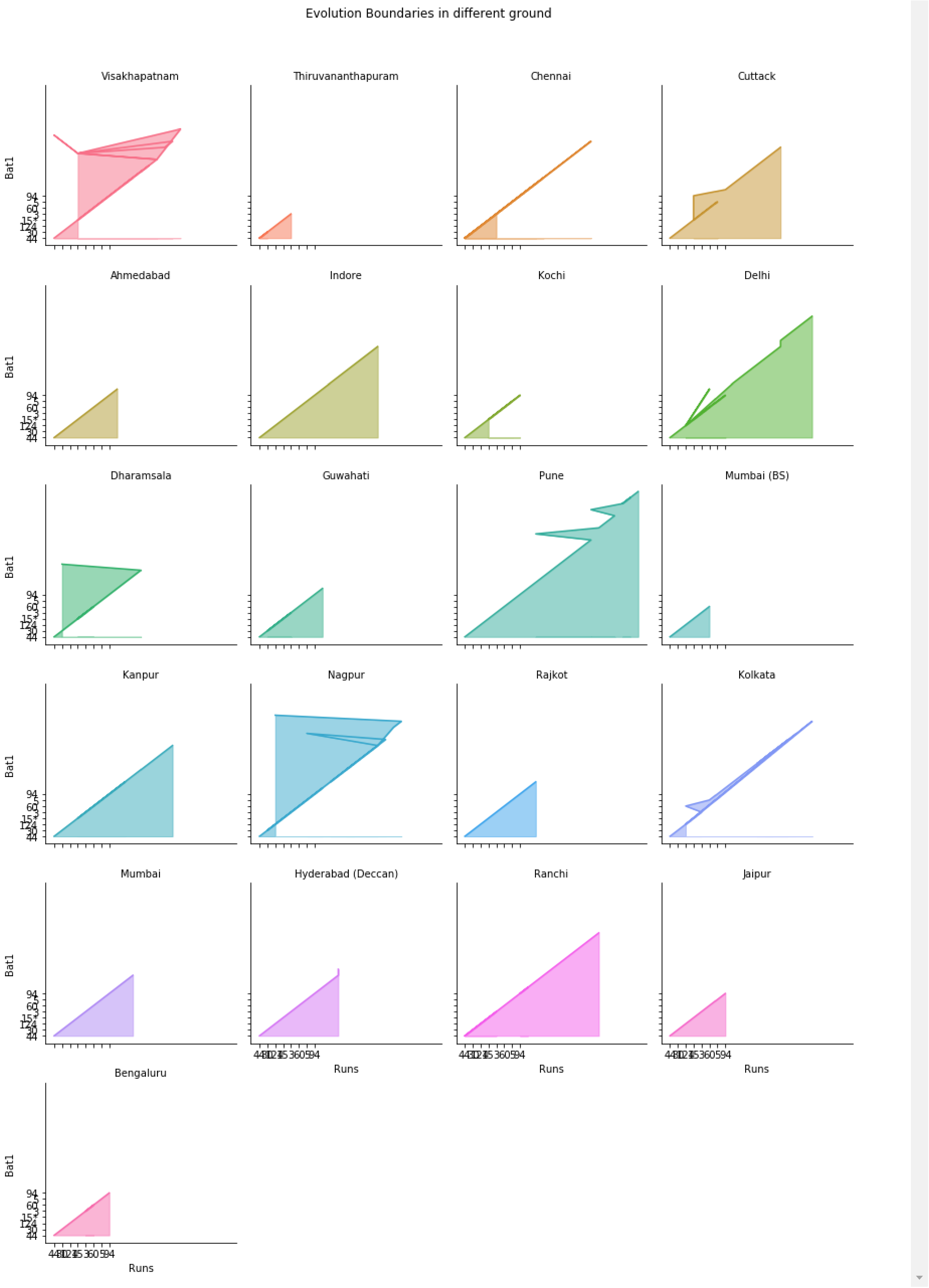
Out[16]:

	Unnamed: 0	Bat1	Runs	BF	SR	4s	6s	Opposition	Ground	Start Date	Match_ID	Batsman	Player_ID
0	70	1	1	2	50.00	0	0	v India	Visakhapatnam	17 Dec 2017	ODI # 3942	Oshane Thomas	49619
1	83	0	0	2	0.00	0	0	v India	Thiruvananthapuram	1 Nov 2018	ODI # 4064	Oshane Thomas	914567
2	93	0	0	5	0.00	0	0	v India	Chennai	20 Mar 2011	ODI # 3141	Andre Russell	276298
3	102	22	22	20	110.00	2	1	v India	Cuttack	29 Nov 2011	ODI # 3217	Andre Russell	276298
4	103	11	11	13	84.61	2	0	v India	Visakhapatnam	2 Dec 2011	ODI # 3219	Andre Russell	276298
...	...	...	...	...	...	...	...	...	...	...	...	...	...
324	10277	32	32	36	88.88	6	0	v India	Nagpur	1 Oct 2017	ODI # 3919	Aaron Finch	5334
325	10292	0	0	3	0.00	0	0	v India	Hyderabad (Deccan)	2 Mar 2019	ODI # 4102	Aaron Finch	5334
326	10293	37	37	53	69.81	5	1	v India	Nagpur	5 Mar 2019	ODI # 4106	Aaron Finch	5334
327	10294	93	93	99	93.93	10	3	v India	Ranchi	8 Mar 2019	ODI # 4109	Aaron Finch	5334
328	10296	27	27	43	62.79	4	0	v India	Delhi	13 Mar 2019	ODI # 4113	Aaron Finch	5334

329 rows × 13 columns

In [17]:

```
my_count=Batsman_Against_India["Ground"]
df = pd.DataFrame({
                "Ground": Batsman_Against_India["Ground"],
                "Runs": Batsman_Against_India["Runs"],
                "Bat1": Batsman_Against_India["Bat1"]
                })

# Create a grid : initialize it
g = sns.FacetGrid(df, col='Ground', hue='Ground', col_wrap=4, )

# Add the line over the area with the plot function
g = g.map(plt.plot, 'Runs', 'Bat1')

# Fill the area with fill_between
g = g.map(plt.fill_between, 'Runs', 'Bat1', alpha=0.5).set_titles("{col_name} Ground")

# Control the title of each facet
g = g.set_titles("{col_name}")

# Add a title for the whole plo
plt.subplots_adjust(top=0.92)
g = g.fig.suptitle('Evolution Boundaries in different ground')
```



GROUND PERFORMANCE

In [18]:

```
ground_avg
```

Out[18]:

	Ground	Span	Mat	Won	Tied	NR	Runs	Wkts	Balls	Ave	RPO
0	Eden Gardens Kolkata-India	2013-2017	4	4	0	0	2161	72	2297	30.01	5.64
1	Feroz Shah Kotla Delhi-India	2013-2019	4	4	0	0	1789	75	2331	23.85	4.60
2	Melbourne Cricket Ground - Australia	2013-2019	15	15	0	0	7656	217	8482	35.28	5.41
3	Saurashtra Cricket Association Stadium Rajkot-...	2013-2015	2	2	0	0	1163	26	1200	44.73	5.81
4	Adelaide Oval - Australia	2013-2019	10	10	0	0	4863	157	5645	30.97	5.16
...	...	...	...	...	...	...	...	...	...	...	...
101	Greenfield International Stadium Thiruvanantha...	2018-2018	1	1	0	0	209	11	280	19.00	4.47
102	Sylhet International Cricket Stadium - Bangladesh	2018-2018	1	1	0	0	400	11	531	36.36	4.51
103	Rajiv Gandhi International Cricket Stadium Deh...	2019-2019	5	4	0	1	1864	67	2524	27.82	4.43
104	Wanderers Cricket Ground, Windhoek - Namibia	2019-2019	1	1	0	0	307	17	474	18.05	3.88
105	Affies Park, Windhoek - Namibia	2019-2019	1	1	0	0	329	15	460	21.93	4.29

106 rows × 11 columns

In [19]:

```
fig = go.Figure(go.Scatter(
    x = ground_avg['Ground'],
    y = ground_avg['Runs'],
))

fig.update_layout(
    title = 'RUNS OVER DIFFERENT GROUND',
    #xaxis_tickformat = '%d %B (%a)<br>%Y'
)

fig.show()
```

In [20]:

```
fig = go.Figure(go.Bar(
    x = ground_avg['Ground'],
    y = ground_avg['Wkts'],
))

fig.update_layout(
    title = 'WICKETS OVER DIFFERENT GROUND',

)

fig.show()
```

In [21]:

```
fig = go.Figure()

fig.add_trace(
    go.Scatter(
        x=ground_avg['Ground'],
        y=ground_avg['RPO'],
        name="Runs Per Over"
    ))

fig.add_trace(
    go.Bar(
        x=ground_avg['Ground'],
        y=ground_avg['Ave'],
        name="Runs Per Wickets"
    ))

fig.update_layout(
    title = 'Runs Per Over Vs Runs Per Wickets in Different Grounds',

)

fig.show()
```

In [22]:

```
fig = go.Figure()

fig.add_trace(
    go.Scatter(
        x=ground_avg['Span'],
        y=ground_avg['RPO'],
        name="Runs Per Over"
    ))

fig.add_trace(
    go.Bar(
        x=ground_avg['Span'],
        y=ground_avg['Ave'],
        name="Runs Per Wickets"
    ))

fig.update_layout(
    title = 'Runs Per Over Vs Runs Per Wickets in Different Spans',

)

fig.show()
```

In [23]:

```
df=ground_avg
fig = px.scatter(ground_avg,
                 x=ground_avg['Mat'],
                 y=ground_avg['Won'],
                 size="Tied",
                 color="Won",
           hover_name="Span", log_x=True, size_max=60)

fig.update_layout(
    title = 'Year wise Number of matches and winning & Tied matches count',

)
fig.show()
```

In [24]:

```
df = ground_avg
fig = px.scatter(df, x="Mat", y="Runs", animation_frame="Span", animation_group="Runs",
                     size="Runs",color="Runs", hover_name="Won")
fig.show()
```

In [25]:

```
fig = px.bar_polar(ground_avg, r="Mat", theta="Runs", color="Span",
                   color_discrete_sequence= px.colors.sequential.Plasma_r,
                   title="Part of a continuous color scale used as a discrete sequence"
                  )
fig.show()
```

FINAL TEAM SELECTION TO PLAY ON FIELD BASED ON ALL FACTORS

In [26]:

```
bowler = pd.read_excel('data/Bowler_data.xlsx')
player= pd.read_csv('data/WC_players.csv')
batsman=pd.read_csv('data/Batsman_data.csv')
newteam=pd.read_csv('data/Players.csv')
```

In [27]:

```
indian=player.loc[player['Country'] =='India']
indian
```

Out[27]:

	Player	ID	Country
61	Virat Kohli (c)	253802	India
62	Rohit Sharma (vc)	34102	India
63	Shikhar Dhawan	28235	India
64	K. L. Rahul	422108	India
65	Vijay Shankar	477021	India
66	MS Dhoni (wk)	28081	India
67	Kedar Jadhav	290716	India
68	Dinesh Karthik	30045	India
69	Yuzvendra Chahal	430246	India
70	Kuldeep Yadav	559235	India
71	Bhuvneshwar Kumar	326016	India
72	Jasprit Bumrah	625383	India
73	Hardik Pandya	625371	India
74	Ravindra Jadeja	234675	India
75	Mohammed Shami	481896	India

BATSMEN PERFORMANCE

In [28]:

```
cricket_batsman={}
```

In [29]:

```
# Removing null values
bowler=bowler.dropna()
batsman=batsman.dropna()
```

In [30]:

```
# Adding details to dict about batsman
for i in indian['ID']:
    for j in batsman['Player_ID']:
        if i==j:
            a=batsman.loc[batsman['Player_ID'] == i]
            run=a['Runs'].mean()
            fours=a['4s'].sum()
            sixs=a['6s'].sum()
            sr=a['SR'].mean()
            cricket_batsman[i]=[run,fours,sixs,sr]
```

In [31]:

```
cricket_batsman
```

Out[31]:

```
{253802: [49.73853211009175, 1019.0, 116.0, 79.88307339449541],
 34102: [40.05, 699.0, 218.0, 69.27959999999999],
 28235: [42.16535433070866, 666.0, 67.0, 83.80551181102362],
 422108: [26.384615384615383, 26.0, 5.0, 72.08846153846153],
 477021: [33.0, 14.0, 4.0, 107.73600000000002],
 28081: [36.332179930795846, 806.0, 224.0, 87.69823529411768],
 290716: [29.35, 121.0, 20.0, 98.75874999999999],
 30045: [22.571428571428573, 174.0, 15.0, 69.50688311688312],
 430246: [4.857142857142857, 5.0, 0.0, 36.94857142857143],
 559235: [7.142857142857143, 8.0, 0.0, 61.995],
 326016: [11.369565217391305, 44.0, 8.0, 69.7573913043478],
 625383: [2.0, 2.0, 1.0, 89.19666666666666],
 625371: [25.20689655172414, 48.0, 36.0, 113.72999999999999],
 234675: [20.76530612244898, 155.0, 37.0, 86.16061224489796],
 481896: [4.517241379310345, 10.0, 6.0, 88.79758620689655]}
```

In [32]:

```
bats=pd.DataFrame.from_dict(cricket_batsman).T
```

In [33]:

```
bats=bats.rename(columns={0:"Runs",1:"4s",2:"6s",3:"SR"})
bats
```

Out[33]:

	Runs	4s	6s	SR
253802	49.738532	1019.0	116.0	79.883073
34102	40.050000	699.0	218.0	69.279600
28235	42.165354	666.0	67.0	83.805512
422108	26.384615	26.0	5.0	72.088462
477021	33.000000	14.0	4.0	107.736000
28081	36.332180	806.0	224.0	87.698235
290716	29.350000	121.0	20.0	98.758750
30045	22.571429	174.0	15.0	69.506883
430246	4.857143	5.0	0.0	36.948571
559235	7.142857	8.0	0.0	61.995000
326016	11.369565	44.0	8.0	69.757391
625383	2.000000	2.0	1.0	89.196667
625371	25.206897	48.0	36.0	113.730000
234675	20.765306	155.0	37.0	86.160612
481896	4.517241	10.0	6.0	88.797586

BOWLERS PERFORMANCE

In [34]:

```
#empty dict for bowlers
cricket_bowler={}
```

In [35]:

```
#addding details of bowlers to dict
for i in indian['ID']:
    for j in bowler['Player_ID']:
        if i==j:
            a=bowler.loc[bowler['Player_ID'] == i]
            over=a['Overs'].sum()
            mdns=a['Mdns'].sum()
            runs=a['Runs'].sum()
            wkts=a['Wkts'].sum()
            eco=a['Econ'].mean()
            ave=a['Ave'].mean()
            sr=a['SR'].mean()
            cricket_bowler[i]=[over,mdns,runs,wkts,eco,ave,sr]
balls=pd.DataFrame.from_dict(cricket_bowler).T
balls=balls.rename(columns={0:"Overs",1:"MDNS",2:"Runs",3:"WKTS",4:"ECO",5:"AVE",6:"SR"})
balls
```

Out[35]:

	Overs	MDNS	Runs	WKTS	ECO	AVE	SR
253802	18.0	0.0	115.0	4.0	6.657500	28.750000	27.000000
34102	23.0	0.0	108.0	8.0	5.468333	13.250000	15.500000
477021	1.3	0.0	15.0	2.0	10.000000	7.500000	4.500000
28081	2.0	0.0	14.0	1.0	7.000000	14.000000	12.000000
290716	104.1	1.0	473.0	27.0	4.445263	20.569474	26.315789
430246	303.9	12.0	1471.0	72.0	4.918182	27.566970	32.863636
559235	323.6	9.0	1612.0	87.0	5.020286	23.822571	27.688571
326016	572.7	52.0	2757.0	118.0	4.717681	30.915942	36.502899
625383	349.1	23.0	1555.0	85.0	4.365610	22.220732	29.002439
625371	219.2	3.0	1143.0	44.0	5.395000	30.303214	34.357143
234675	853.0	35.0	3919.0	174.0	4.634687	29.469479	37.750000
481896	455.5	30.0	2484.0	113.0	5.427736	27.533774	30.990566

SELECTION

In [36]:

```
# Selecting Top 6 batsman
bats=bats.sort_values(['6s','SR'],ascending=False)
bats=bats.head(6)
bats
```

Out[36]:

	Runs	4s	6s	SR
28081	36.332180	806.0	224.0	87.698235
34102	40.050000	699.0	218.0	69.279600
253802	49.738532	1019.0	116.0	79.883073
28235	42.165354	666.0	67.0	83.805512
234675	20.765306	155.0	37.0	86.160612
625371	25.206897	48.0	36.0	113.730000

In [37]:

```
# Selecting Top 6 bowlers
balls=balls.sort_values(['ECO'],ascending=True)
balls=balls.head(6)
balls
```

Out[37]:

	Overs	MDNS	Runs	WKTS	ECO	AVE	SR
625383	349.1	23.0	1555.0	85.0	4.365610	22.220732	29.002439
290716	104.1	1.0	473.0	27.0	4.445263	20.569474	26.315789
234675	853.0	35.0	3919.0	174.0	4.634687	29.469479	37.750000
326016	572.7	52.0	2757.0	118.0	4.717681	30.915942	36.502899
430246	303.9	12.0	1471.0	72.0	4.918182	27.566970	32.863636
559235	323.6	9.0	1612.0	87.0	5.020286	23.822571	27.688571

In [38]:

```
# Common players
playerid_bats=bats.index
bats_list=set(playerid_bats)
playerid_balls=balls.index
balls_list=set(playerid_balls)
bats_list.intersection(balls_list)
```

Out[38]:

```
{234675}
```

In [39]:

```python
# Making list of playres
players=[]
bats_list=list(playerid_bats)
balls_list=list(playerid_balls)
for i in bats_list:
    if i not in players:
        players.append(i)
for i in balls_list:
    if i not in players:
        players.append(i)
```

LIST OF PLAYERS ON FIELD FOR 2023 WORLD CUP

In [40]:

```python
f_l=[]
for i in players:
    for j in list(indian['ID']):
        if i==j:
            a=indian.loc[indian['ID'] == i]
            a=a.drop(['ID','Country'],axis=1)
            k= a.to_string(index=False,header=False)
            f_l.append(k)
```

In [41]:

```python
import numpy as np
S=pd.DataFrame(np.array(f_l).reshape(len(f_l)))
S.columns=['Player']
S
#pd.DataFrame(np.array(f_l).reshape(-1,len(f_l)))
```

Out[41]:

	Player
0	MS Dhoni (wk)
1	Rohit Sharma (vc)
2	Virat Kohli (c)
3	Shikhar Dhawan
4	Ravindra Jadeja
5	Hardik Pandya
6	Jasprit Bumrah
7	Kedar Jadhav
8	Bhuvneshwar Kumar
9	Yuzvendra Chahal
10	Kuldeep Yadav

INFERENCES:

Based on the performance of the team members we need a change in the team.

- MS Dhoni has taken retirement
- Dinesh Kartik and Vijay Shankar can be removed
- New young members can be included like Shreyas Iyer, Prithvi Shaw, Washington Sundar, Navdeep Saini, Devdutt Padikalal,Varun Chakravorthy
- Virat Kohli(Captain)
- Rohit Sharma(Vice Captain)

In [42]:

```python
TEAM_2023=pd.DataFrame(newteam)
TEAM_2023
```

Out[42]:

	PLAYER	GAME
0	Virat Kohli (c)	Batsman
1	Rohit Sharma (vc)	Batsman
2	Shikhar Dhawan	Batsman
3	K. L. Rahul	Batsman, W.K
4	Dinesh Karthik	Batsman, W.K
5	Yuzvendra Chahal	Bowler
6	Kuldeep Yadav	Bowler
7	Bhuvneshwar Kumar	Bowler
8	Jasprit Bumrah	Bowler
9	Hardik Pandya	All Rounder
10	Ravindra Jadeja	All Rounder
11	Mohammed Shami	Bowler