World Cup 2023 Analysis

Mayank Pujara | mayankpujara25@gmail.com

The World Cup 2023 Data Analysis project aims to explore and derive insights from a comprehensive dataset that encapsulates the performance metrics of cricket teams and players during the prestigious tournament. The primary goal is to conduct an in-depth Exploratory Data Analysis (EDA) to reveal patterns, trends, and significant statistics that contribute to a nuanced understanding of team and player performance in both batting and bowling aspects

Importing Necessary Libraries

In [1]:

```python
import numpy as np
import pandas as pd
import plotly.express as px
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from matplotlib.dates import DateFormatter
import plotly.graph_objs as go
from plotly.subplots import make_subplots
```

```
C:\Python310\lib\site-packages\scipy\__init__.py:169: UserWarning: A NumPy version >=1.18.5 and
<1.26.0 is required for this version of SciPy (detected version 1.26.3
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

Loading the Dataset

In [2]:

```python
data = pd.read_csv(r"CWC23_all_innings.csv")
```

In [3]:

```python
data
```

Out[3]:

	team	player	bat_or_bowl	bb_bf	runs	wkts	wicketball_prob	runs_per_ball	opposition	ground	s
0	PAK	Shaheen Shah Afridi (PAK)	bowl	60	45	3.0	0.05	0.750000	v South Africa	Chennai	2
1	ENG	DJ Willey (ENG)	bowl	60	45	3.0	0.05	0.750000	v India	Lucknow	2
2	NZ	MJ Henry (NZ)	bowl	60	48	3.0	0.05	0.800000	v England	Ahmedabad	
3	NZ	LH Ferguson (NZ)	bowl	60	49	3.0	0.05	0.816667	v Bangladesh	Chennai	1
4	AFG	Noor Ahmad (AFG)	bowl	60	49	3.0	0.05	0.816667	v Pakistan	Chennai	2
...	...	...	...	...	...	...	...	...	...	...	
1403	NZ	MJ Santner (NZ)	bowl	60	51	0.0	0.00	0.850000	v India	Wankhede	
1404	AUS	A Zampa (AUS)	bowl	42	55	0.0	0.00	1.309524	v South Africa	Eden Gardens	
1405	NZ	R Ravindra (NZ)	bowl	42	60	0.0	0.00	1.428571	v India	Wankhede	
1406	IND	RA Jadeja (IND)	bowl	60	63	0.0	0.00	1.050000	v New Zealand	Wankhede	
1407	NZ	LH Ferguson (NZ)	bowl	48	65	0.0	0.00	1.354167	v India	Wankhede	

1408 rows × 20 columns

Data Preprocessing & Cleaning

In [4]:

```
data.shape
```

Out[4]:

```
(1408, 20)
```

In [5]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1408 entries, 0 to 1407
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   team             1408 non-null   object 
 1   player           1408 non-null   object 
 2   bat_or_bowl      1408 non-null   object 
 3   bb_bf            1408 non-null   int64  
 4   runs             1408 non-null   int64  
 5   wkts             562 non-null    float64
 6   wicketball_prob  1408 non-null   float64
 7   runs_per_ball    1408 non-null   float64
 8   opposition       1408 non-null   object 
 9   ground           1408 non-null   object 
 10  start_date       1408 non-null   object 
 11  overs            562 non-null    float64
 12  mdns             562 non-null    float64
 13  econ             562 non-null    float64
 14  inns             1408 non-null   int64  
 15  4s               846 non-null    float64
 16  6s               846 non-null    float64
 17  sr               846 non-null    float64
 18  not_out          846 non-null    float64
 19  mins             846 non-null    float64
dtypes: float64(11), int64(3), object(6)
memory usage: 220.1+ KB
```

In [6]:

```python
data.isnull().sum()
```

Out[6]:

```
team                 0
player               0
bat_or_bowl          0
bb_bf                0
runs                 0
wkts               846
wicketball_prob      0
runs_per_ball        0
opposition           0
ground               0
start_date           0
overs              846
mdns               846
econ               846
inns                 0
4s                 562
6s                 562
sr                 562
not_out            562
mins               562
dtype: int64
```

Replacing the null values with 0

In [7]:

```python
data.fillna(0, inplace = True)
data.head()
```

Out[7]:

	team	player	bat_or_bowl	bb_bf	runs	wkts	wicketball_prob	runs_per_ball	opposition	ground	start
0	PAK	Shaheen Shah Afridi (PAK)	bowl	60	45	3.0	0.05	0.750000	v South Africa	Chennai	27-C
1	ENG	DJ Willey (ENG)	bowl	60	45	3.0	0.05	0.750000	v India	Lucknow	29-C
2	NZ	MJ Henry (NZ)	bowl	60	48	3.0	0.05	0.800000	v England	Ahmedabad	5-C
3	NZ	LH Ferguson (NZ)	bowl	60	49	3.0	0.05	0.816667	v Bangladesh	Chennai	13-C
4	AFG	Noor Ahmad (AFG)	bowl	60	49	3.0	0.05	0.816667	v Pakistan	Chennai	23-C

Team wise Bowling Performance

In [8]:

```
bowlersData = data[data['bat_or_bowl'] == 'bowl'].groupby('team').agg({
'bb_bf':'sum',
'runs':'sum',
'wkts':'sum',
'wicketball_prob':'mean',
'runs_per_ball':'mean',
'overs':'sum',
'mdns':'sum',
'econ':'mean',
'4s':'sum',
'6s':'sum','sr':'mean','not_out':'sum','mins':'sum'})
bowlersData.reset_index(inplace=True)
bowlersData
```

Out[8]:

	team	bb_bf	runs	wkts	wicketball_prob	runs_per_ball	overs	mdns	econ	4s	6s	sr	not_out	mins
0	AFG	2403	2144	53.0	0.020289	0.960929	399.1	13.0	5.763818	0.0	0.0	0.0	0.0	0.0
1	AUS	2695	2512	77.0	0.028144	0.943872	448.3	13.0	5.661129	0.0	0.0	0.0	0.0	0.0
2	BAN	2390	2414	51.0	0.019142	1.016412	397.0	13.0	6.095769	0.0	0.0	0.0	0.0	0.0
3	ENG	2343	2261	65.0	0.024686	1.011806	388.7	14.0	6.068302	0.0	0.0	0.0	0.0	0.0
4	IND	2506	1973	94.0	0.042093	0.827580	415.4	23.0	4.963793	0.0	0.0	0.0	0.0	0.0
5	NED	2502	2540	63.0	0.022379	1.028768	416.2	11.0	6.170339	0.0	0.0	0.0	0.0	0.0
6	NZ	2704	2664	71.0	0.024635	1.039779	449.2	17.0	6.237143	0.0	0.0	0.0	0.0	0.0
7	PAK	2478	2566	65.0	0.024247	1.048316	412.6	8.0	6.286923	0.0	0.0	0.0	0.0	0.0
8	SA	2561	2324	88.0	0.035004	0.947549	424.5	20.0	5.682778	0.0	0.0	0.0	0.0	0.0
9	SL	2259	2424	50.0	0.019058	1.101300	375.5	12.0	6.606111	0.0	0.0	0.0	0.0	0.0

In [9]:

```
bowlersData.rename(columns={'wkts': 'wickets'}, inplace=True)
bowlersData.rename(columns={'mdns': 'maidens'}, inplace=True)
bowlersData.rename(columns={'econ': 'economy'}, inplace=True)
```

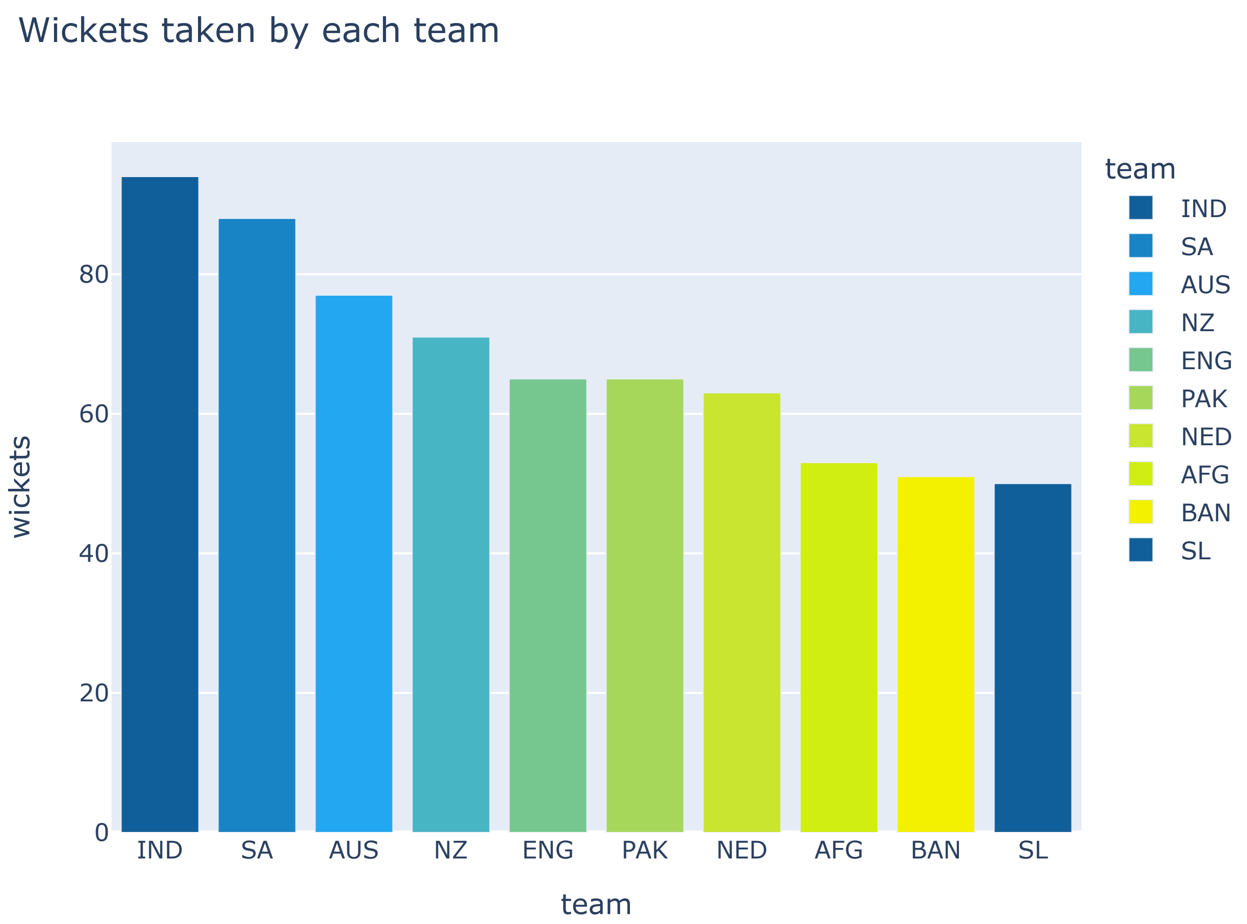
In [10]:

```
wicketsSorted = bowlersData.sort_values(['wickets'], ascending = False)
oversSorted = bowlersData.sort_values(['overs'], ascending = False)
maidensSorted = bowlersData.sort_values(['maidens'], ascending = False)
economySorted = bowlersData.sort_values(['economy'], ascending = True)

print(wicketsSorted [['team', 'overs', 'maidens', 'wickets', 'economy']])
```

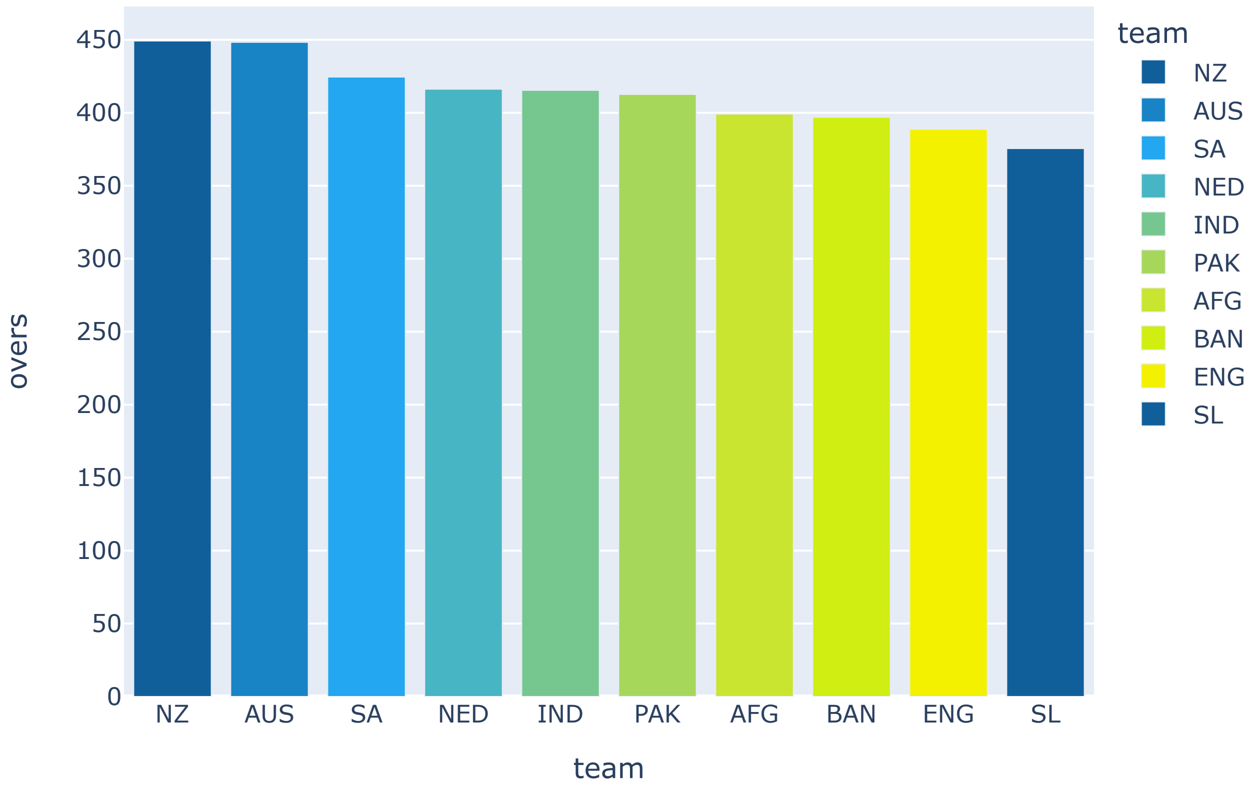
```
  team  overs  maidens  wickets   economy
4  IND  415.4     23.0     94.0  4.963793
8   SA  424.5     20.0     88.0  5.682778
1  AUS  448.3     13.0     77.0  5.661129
6   NZ  449.2     17.0     71.0  6.237143
3  ENG  388.7     14.0     65.0  6.068302
7  PAK  412.6      8.0     65.0  6.286923
5  NED  416.2     11.0     63.0  6.170339
0  AFG  399.1     13.0     53.0  5.763818
2  BAN  397.0     13.0     51.0  6.095769
9   SL  375.5     12.0     50.0  6.606111
```

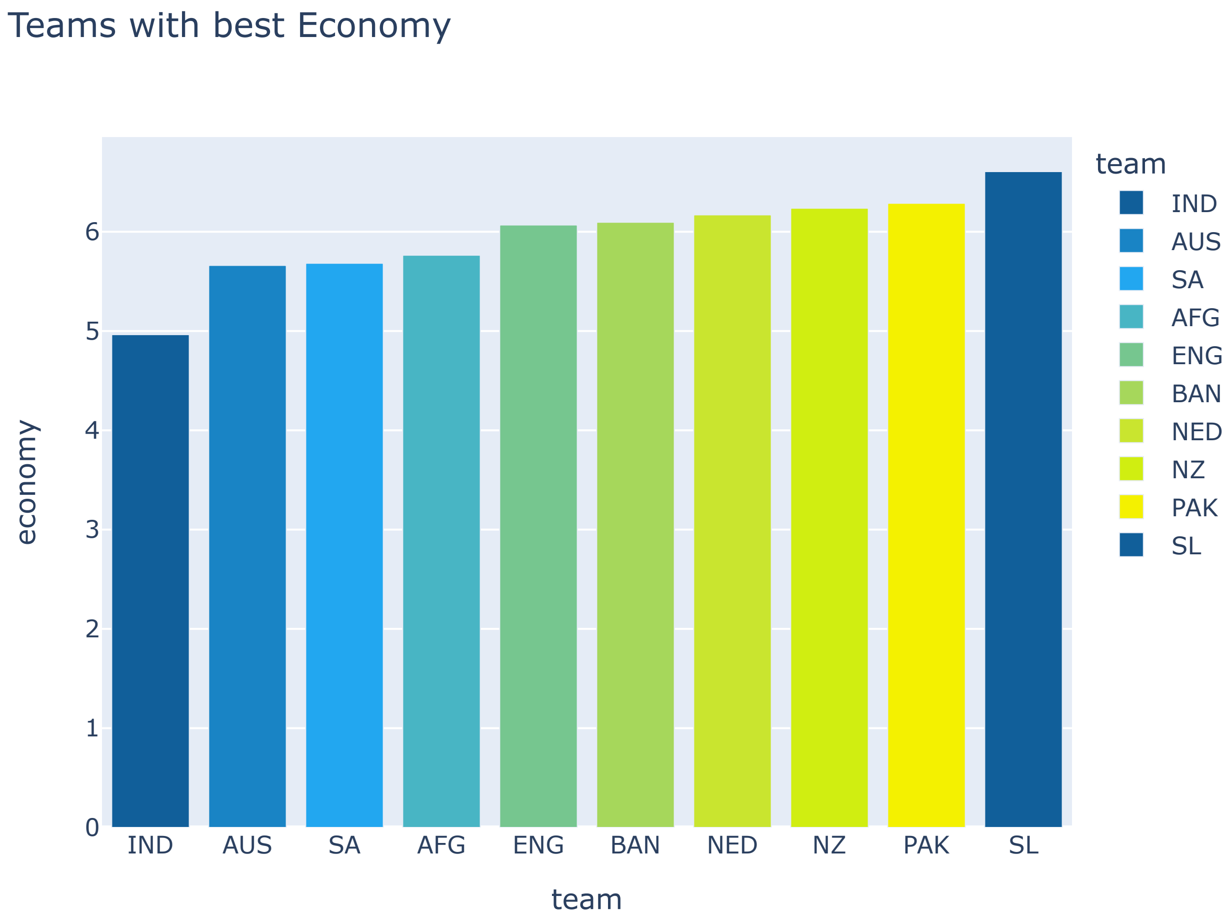
In [11]:

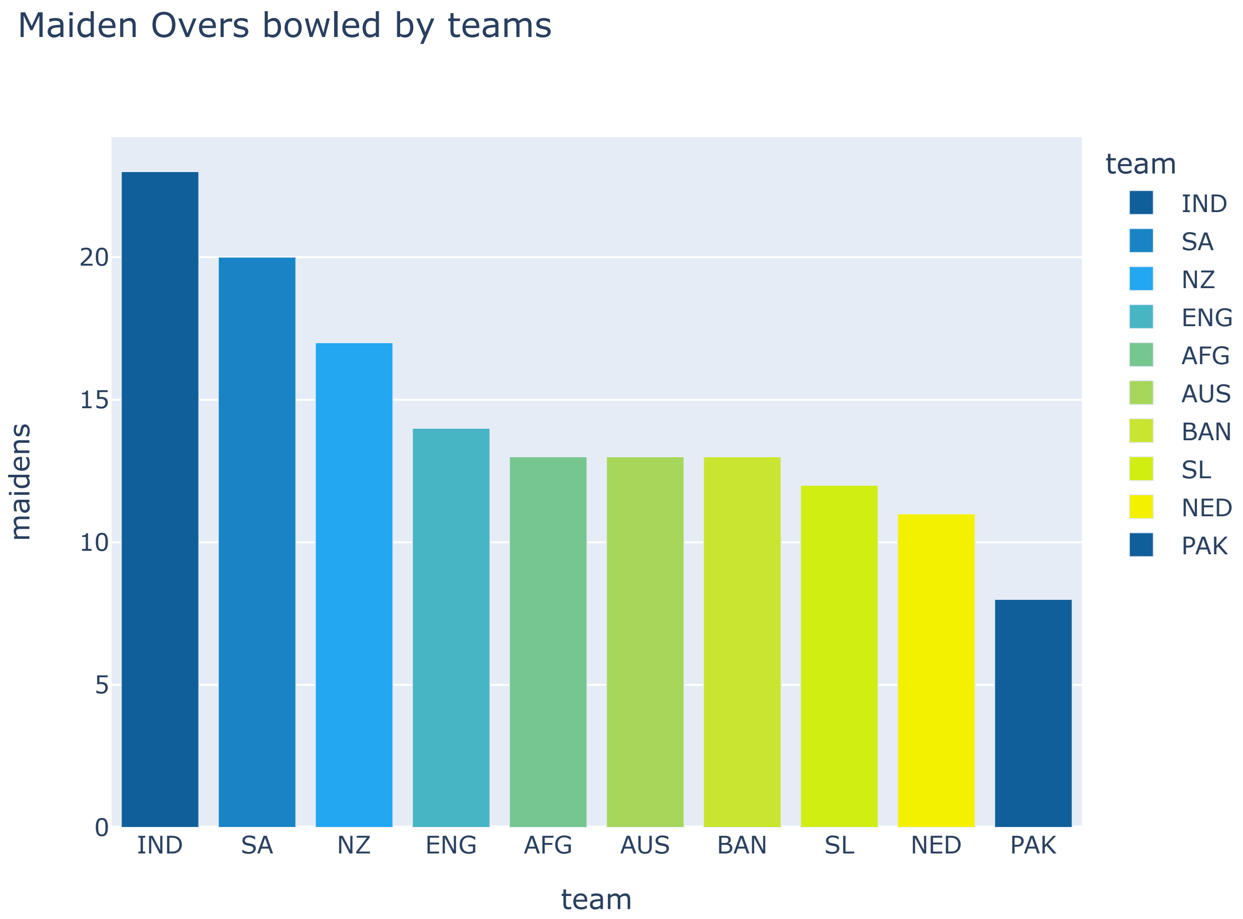
```
colorScheme = ["#115f9a", "#1984c5", "#22a7f0", "#48b5c4", "#76c68f", "#a6d75b", "#c9e52f", "#d0

fig = px.bar(wicketsSorted, x='team', y='wickets', title='Wickets taken by each team', color='te
fig.show()

fig = px.bar(oversSorted, x='team', y='overs', title='Overs delivered by each team', color='team
fig.show()

fig = px.bar(economySorted, x='team', y='economy', title='Teams with best Economy', color='team'
fig.show()

fig = px.bar(maidensSorted, x='team', y='maidens', title='Maiden Overs bowled by teams', color='
fig.show()
```


Wickets taken by each team
wickets
0
20
40
60
80
team
IND
SA
AUS
NZ
ENG
PAK
NED
AFG
BAN
SL
team
IND
SA
AUS
NZ
ENG
PAK
NED
AFG
BAN
SL

Overs delivered by each team




Teams with best Economy
economy
0
1
2
3
4
5
6
team
IND
AUS
SA
AFG
ENG
BAN
NED
NZ
PAK
SL
team
IND
AUS
SA
AFG
ENG
BAN
NED
NZ
PAK
SL



Maiden Overs bowled by teams

Bowling Analysis

Wickets

- India (94 wickets), South Africa (88 wickets) and Australia (77 wickets) are the top 3 teams with the most number of wickets.
- Sri Lanka is the team with least number of wickets, with 50 wickets.

Overs Bowled

- New Zealand and Australia delivered more number of overs (499 and 448 respecitvely).
- Sri Lanka bowled the least number of overs (375).

Economy and Maidens

- India had the best economy among all the teams (4.9), followed by Australia.
- Sri Lanka and Pakistan had the worst economy among all teams.
- Indian bowling unit bowled the most number of maidens in the tournament

Overall, The Indian Team had the best bowling unit in the World Cup.

Team wise Batting Performance

In [12]:

```
batsmanData = data[data['bat_or_bowl'] == 'bat'].groupby('team').agg({
'bb_bf':'sum',
'runs':'sum',
'wkts':'sum',
'wicketball_prob':'mean',
'runs_per_ball':'mean',
'overs':'sum',
'mdns':'sum',
'econ':'mean',
'4s':'sum',
'6s':'sum','sr':'mean','not_out':'sum','mins':'sum'})
batsmanData.reset_index(inplace=True)
batsmanData
```

Out[12]:

	team	bb_bf	runs	wkts	wicketball_prob	runs_per_ball	overs	mdns	econ	4s	6s	sr	not_out
0	AFG	2392	1990	0.0	0.079171	0.776484	0.0	0.0	0.0	178.0	42.0	77.648378	12.0
1	AUS	2787	2722	0.0	0.116286	0.855715	0.0	0.0	0.0	265.0	92.0	85.571290	15.0
2	BAN	2459	1944	0.0	0.092788	0.738124	0.0	0.0	0.0	188.0	43.0	73.812022	13.0
3	ENG	2277	2135	0.0	0.122657	0.894078	0.0	0.0	0.0	216.0	51.0	89.407551	12.0
4	IND	2685	2810	0.0	0.070361	1.002598	0.0	0.0	0.0	265.0	89.0	100.259545	18.0
5	NED	2300	1728	0.0	0.122535	0.677484	0.0	0.0	0.0	163.0	33.0	67.748061	10.0
6	NZ	2616	2712	0.0	0.086853	0.970416	0.0	0.0	0.0	265.0	82.0	97.041605	15.0
7	PAK	2309	2220	0.0	0.080997	0.854799	0.0	0.0	0.0	220.0	60.0	85.479342	10.0
8	SA	2787	2773	0.0	0.071214	0.984432	0.0	0.0	0.0	243.0	99.0	98.443218	15.0
9	SL	2257	1942	0.0	0.139508	0.667990	0.0	0.0	0.0	201.0	45.0	66.799048	8.0

In [13]:

```
batsmanData.rename(columns={'sr': 'strike rate'}, inplace=True)
batsmanData.rename(columns={'mins': 'minutes batted'}, inplace=True)
```

In [14]:

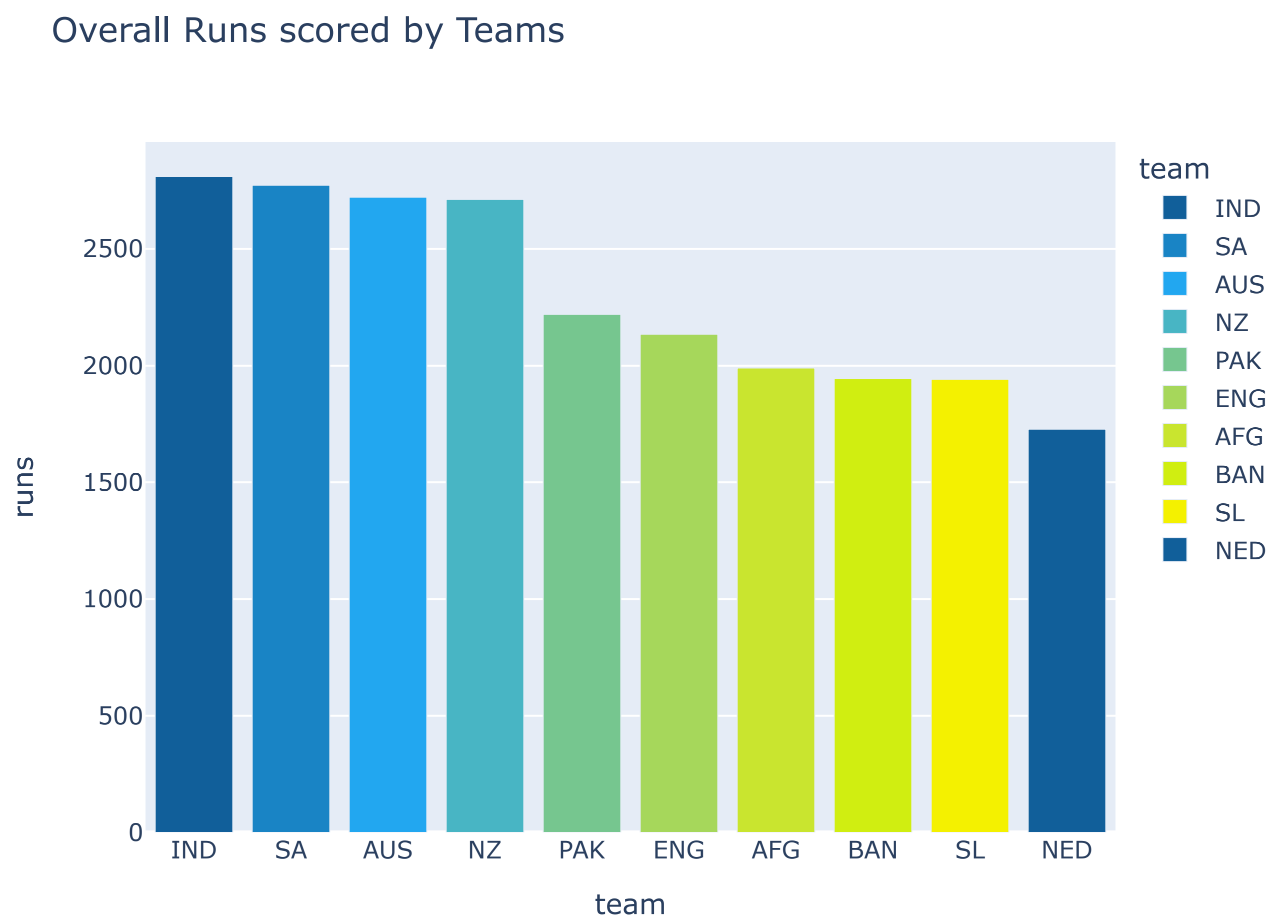
```
runsSorted = batsmanData.sort_values(['runs'], ascending = False)
strikeRateSorted = batsmanData.sort_values(['strike rate'], ascending = False)
maximum4s = batsmanData.sort_values(['4s'], ascending = False)
maximum6s = batsmanData.sort_values(['6s'], ascending = False)
minutesBatted = batsmanData.sort_values(['minutes batted'], ascending = False)

batsmanData[['team', 'runs', 'strike rate', '4s', '6s', 'minutes batted']]
```

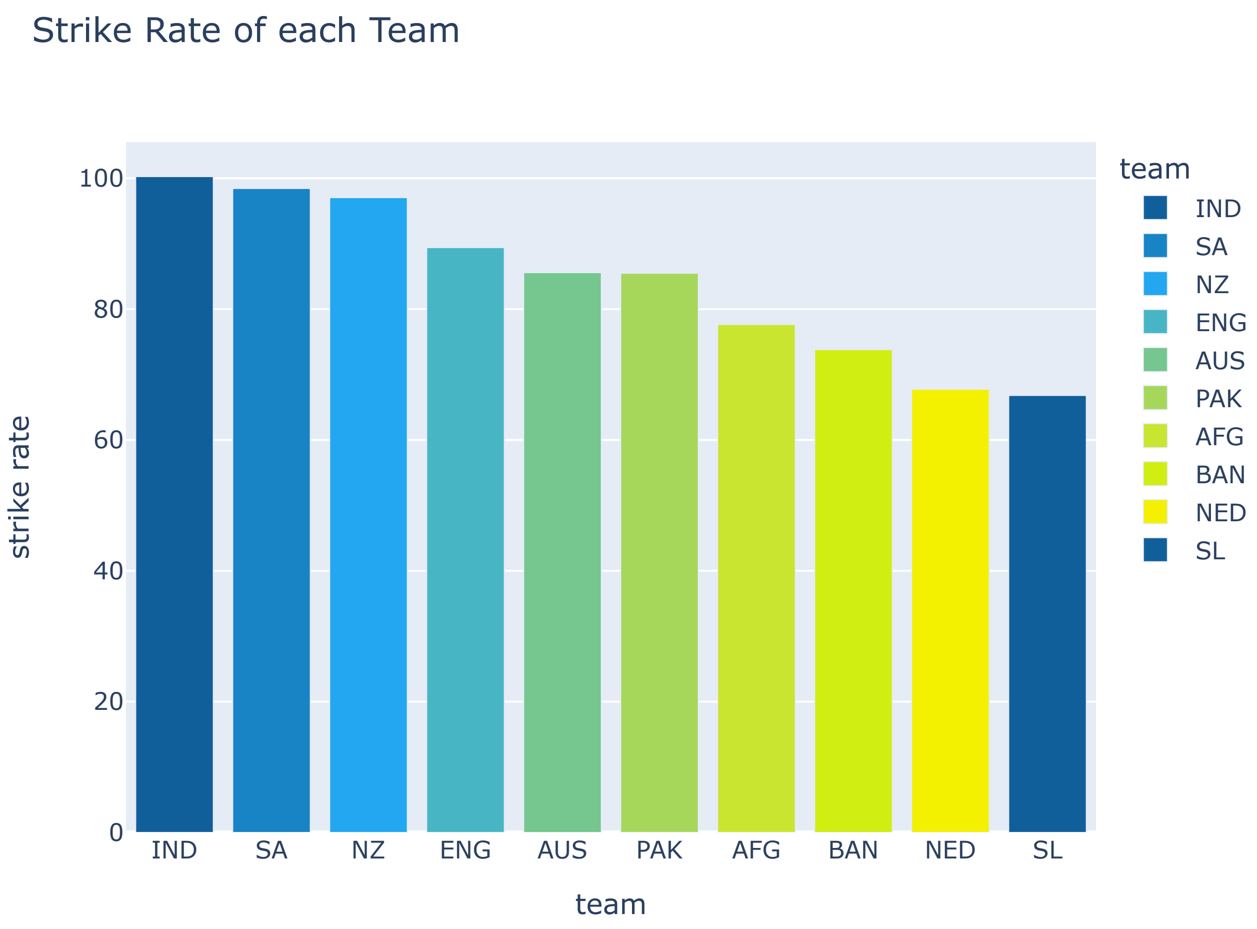
Out[14]:

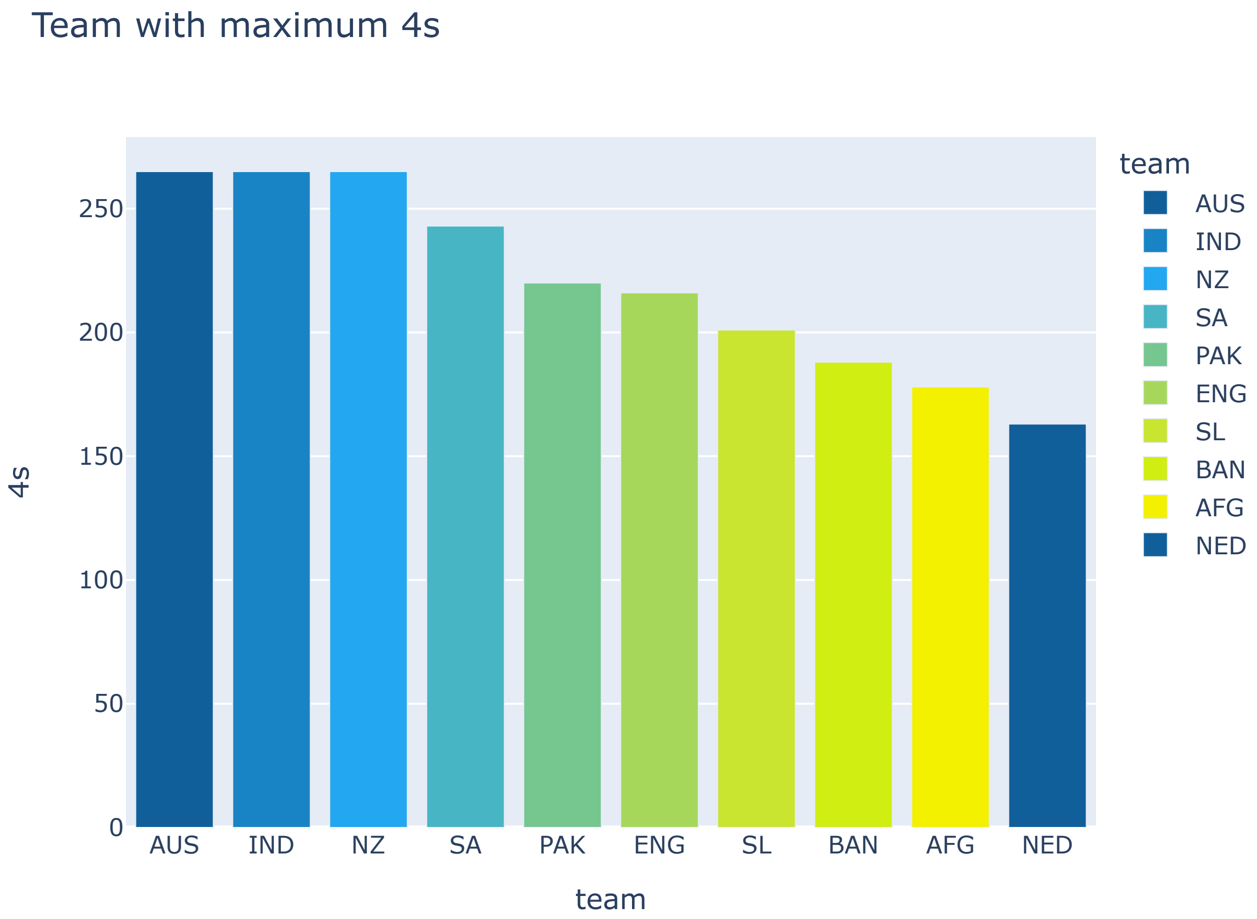
	team	runs	strike rate	4s	6s	minutes batted
0	AFG	1990	77.648378	178.0	42.0	3462.0
1	AUS	2722	85.571290	265.0	92.0	4090.0
2	BAN	1944	73.812022	188.0	43.0	3599.0
3	ENG	2135	89.407551	216.0	51.0	3301.0
4	IND	2810	100.259545	265.0	89.0	3881.0
5	NED	1728	67.748061	163.0	33.0	3167.0
6	NZ	2712	97.041605	265.0	82.0	3895.0
7	PAK	2220	85.479342	220.0	60.0	3342.0
8	SA	2773	98.443218	243.0	99.0	4083.0
9	SL	1942	66.799048	201.0	45.0	3322.0

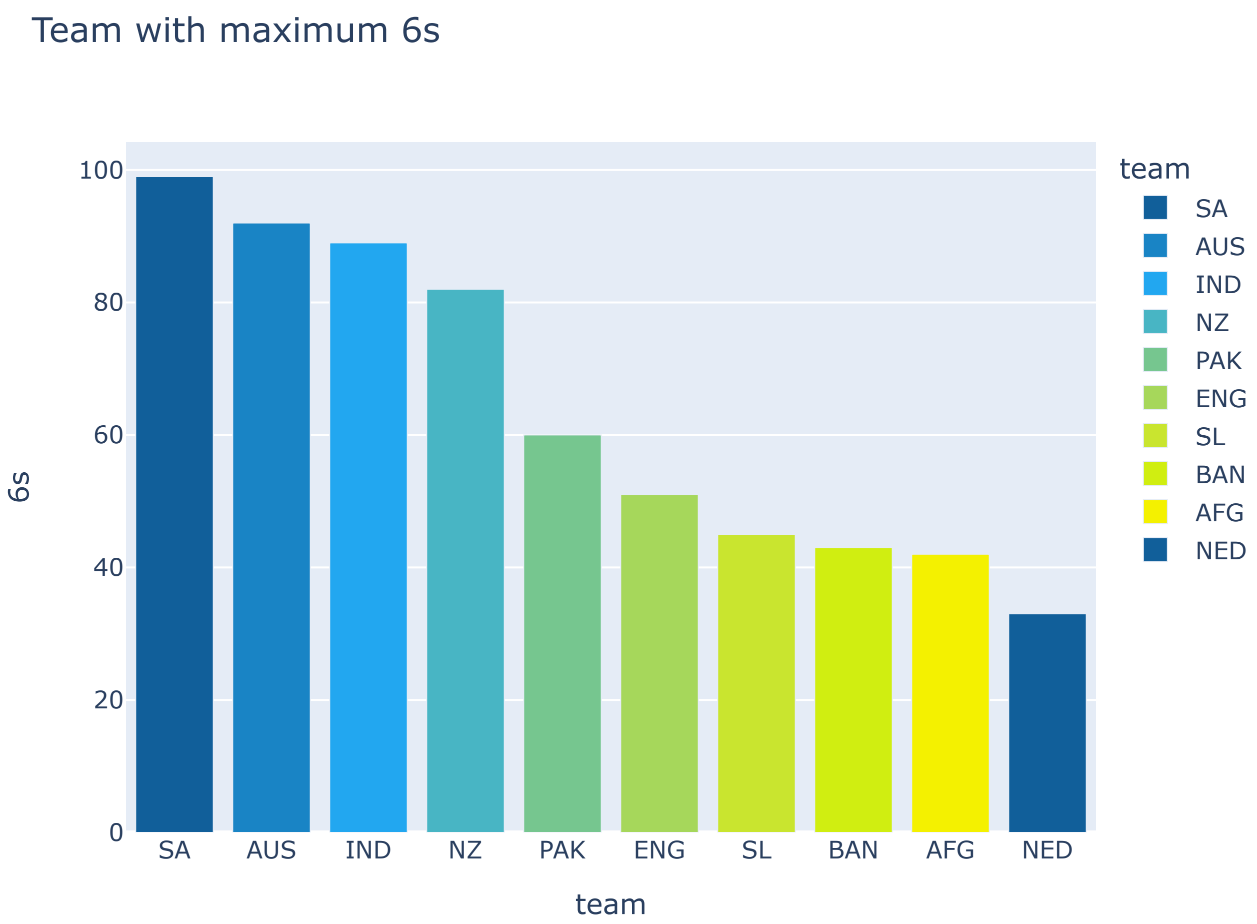
In [15]:

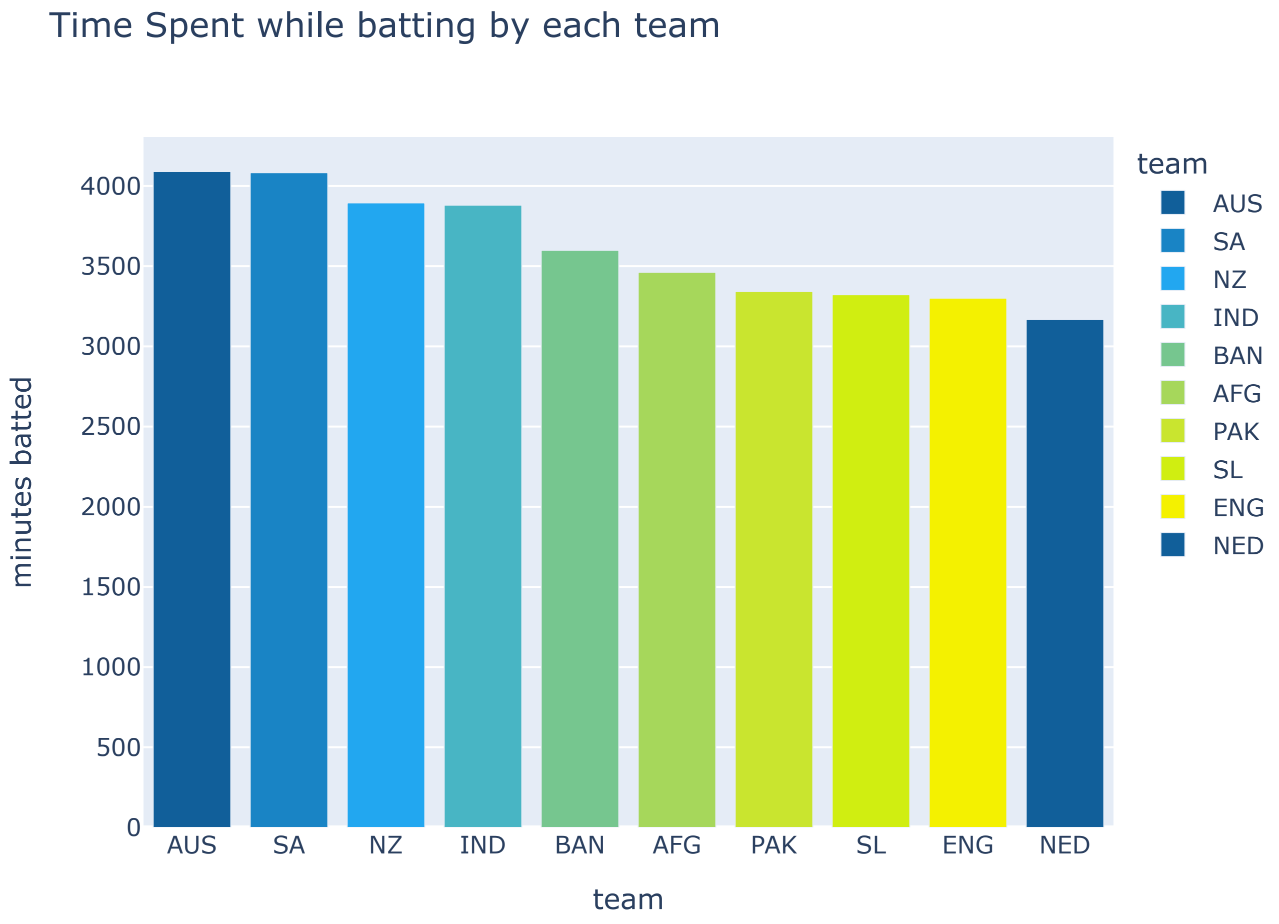
```
colorScheme = ["#115f9a", "#1984c5", "#22a7f0", "#48b5c4", "#76c68f", "#a6d75b", "#c9e52f", "#d0

fig = px.bar(runsSorted, x='team', y='runs', title='Overall Runs scored by Teams', color='team',
fig.show()

fig = px.bar(strikeRateSorted, x='team', y='strike rate', title='Strike Rate of each Team', colo
fig.show()

fig = px.bar(maximum4s, x='team', y='4s', title='Team with maximum 4s', color='team', color_disc
fig.show()

fig = px.bar(maximum6s, x='team', y='6s', title='Team with maximum 6s', color='team', color_disc
fig.show()

fig = px.bar(minutesBatted, x='team', y='minutes batted', title='Time Spent while batting by eac
fig.show()
```


Overall Runs scored by Teams
runs
2500
2000
1500
1000
500
0
IND
SA
AUS
NZ
PAK
ENG
AFG
BAN
SL
NED
team
team
IND
SA
AUS
NZ
PAK
ENG
AFG
BAN
SL
NED


Strike Rate of each Team
strike rate
100
80
60
40
20
0
IND
SA
NZ
ENG
AUS
PAK
AFG
BAN
NED
SL
team
team
IND
SA
NZ
ENG
AUS
PAK
AFG
BAN
NED
SL


Team with maximum 4s
4s
250
200
150
100
50
0
AUS
IND
NZ
SA
PAK
ENG
SL
BAN
AFG
NED
team
team
AUS
IND
NZ
SA
PAK
ENG
SL
BAN
AFG
NED


Team with maximum 6s
6s
100
80
60
40
20
0
SA
AUS
IND
NZ
PAK
ENG
SL
BAN
AFG
NED
team
team
SA
AUS
IND
NZ
PAK
ENG
SL
BAN
AFG
NED



Batting Analysis

Runs Scored

- India (2810 runs), South Africa (2773 runs) and Australia (2772 runs) are the top 3 teams with the most number of runs scored in the tournament.
- Netherlands is the team with least number of runs, 1728 runs.

Strike Rate and Boundries Scored

- India, South Africa and New Zealand have the best Strike Rate among all the teams, with 100, 98, 97 respectively.
- Overall, Australian scored the most number of boundries (4s and 6s combined) as well as have spent most time on the ground while batting followed by South Africa.

Overall, Both The Indian Team and the South African team were the pick of the teams while batting.

Player wise Bowling Performance

In [16]:
```
playerBowlingPerformance = data[data['bat_or_bowl'] == 'bowl'].groupby('player').agg({
'bb_bf':'sum',
'runs':'sum',
'wkts':'sum',
'wicketball_prob':'mean',
'runs_per_ball':'mean',
'overs':'sum',
'mdns':'sum',
'econ':'mean'
})

playerBowlingPerformance.sort_values('wkts',ascending=False,inplace=True)
playerBowlingPerformance.reset_index(inplace=True)
playerBowlingPerformance.head(10)
```

Out[16]:

	player	bb_bf	runs	wkts	wicketball_prob	runs_per_ball	overs	mdns	econ
0	Mohammed Shami (IND)	251	210	23.0	0.091203	0.813133	41.5	3.0	4.876667
1	A Zampa (AUS)	516	471	22.0	0.053889	0.897063	86.0	1.0	5.380000
2	D Madushanka (SL)	470	525	21.0	0.040438	1.131685	77.8	4.0	6.786667
3	G Coetzee (SA)	381	396	20.0	0.056749	1.074176	63.3	1.0	6.442500
4	JJ Bumrah (IND)	497	330	18.0	0.035783	0.631065	82.5	7.0	3.785000
5	Shaheen Shah Afridi (PAK)	486	481	18.0	0.037419	0.982863	81.0	3.0	5.895556
6	M Jansen (SA)	414	450	17.0	0.040694	1.085521	68.6	3.0	6.508889
7	MJ Santner (NZ)	556	449	16.0	0.028188	0.826449	92.4	4.0	4.958000
8	Haris Rauf (PAK)	474	533	16.0	0.032562	1.128601	79.0	1.0	6.768889
9	RA Jadeja (IND)	501	355	16.0	0.052067	0.728521	82.9	4.0	4.369000

In [17]:

```
playerBowlingPerformanceEconomy = playerBowlingPerformance[playerBowlingPerformance['overs'] > 1
playerBowlingPerformanceOvers = playerBowlingPerformance.sort_values('overs',ascending=False)
playerBowlingPerformanceMaidens = playerBowlingPerformance.sort_values('mdns',ascending=False)
playerBowlingPerformanceEconomy[['player','wkts','overs','econ','mdns']].head(10)
```

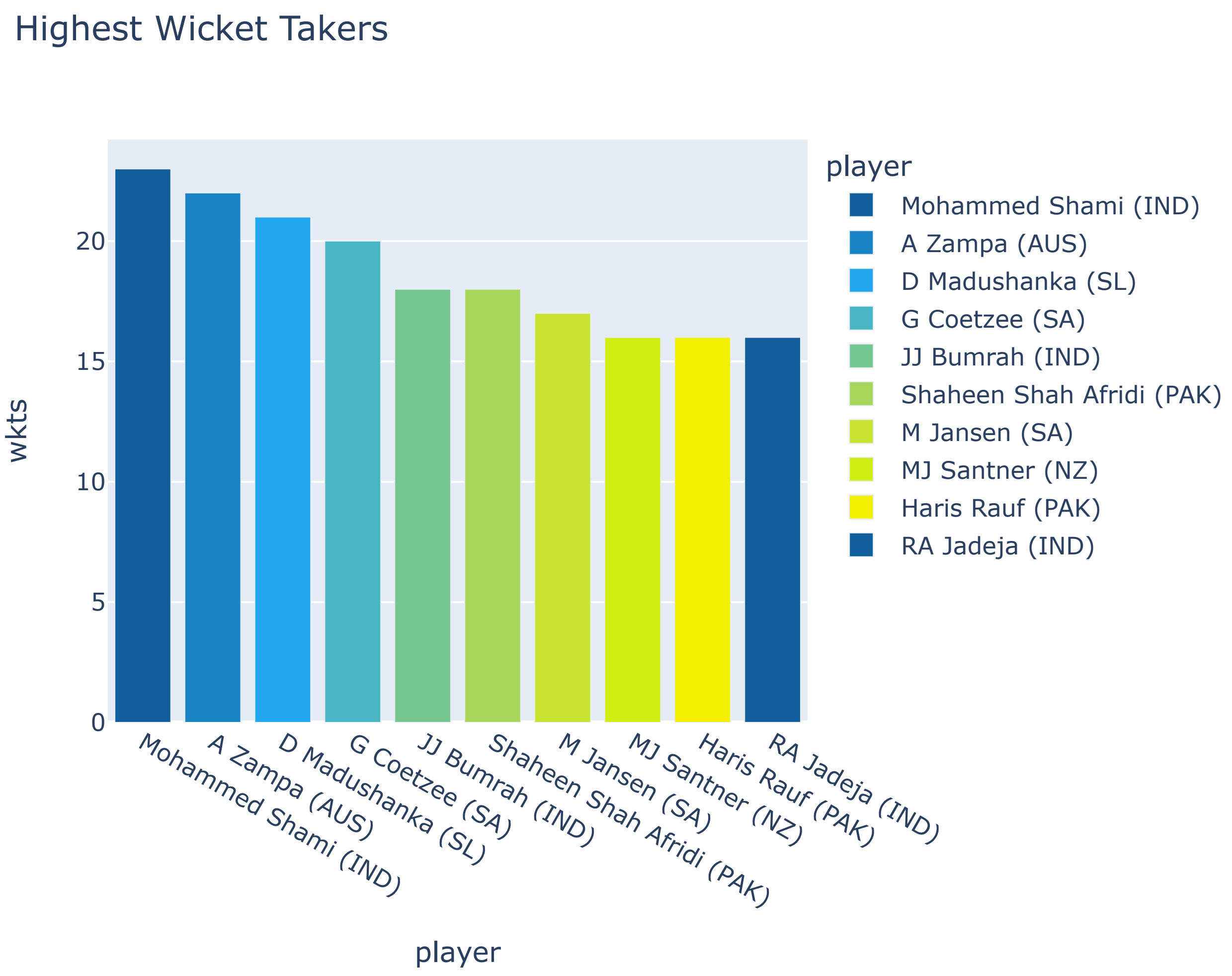
Out[17]:

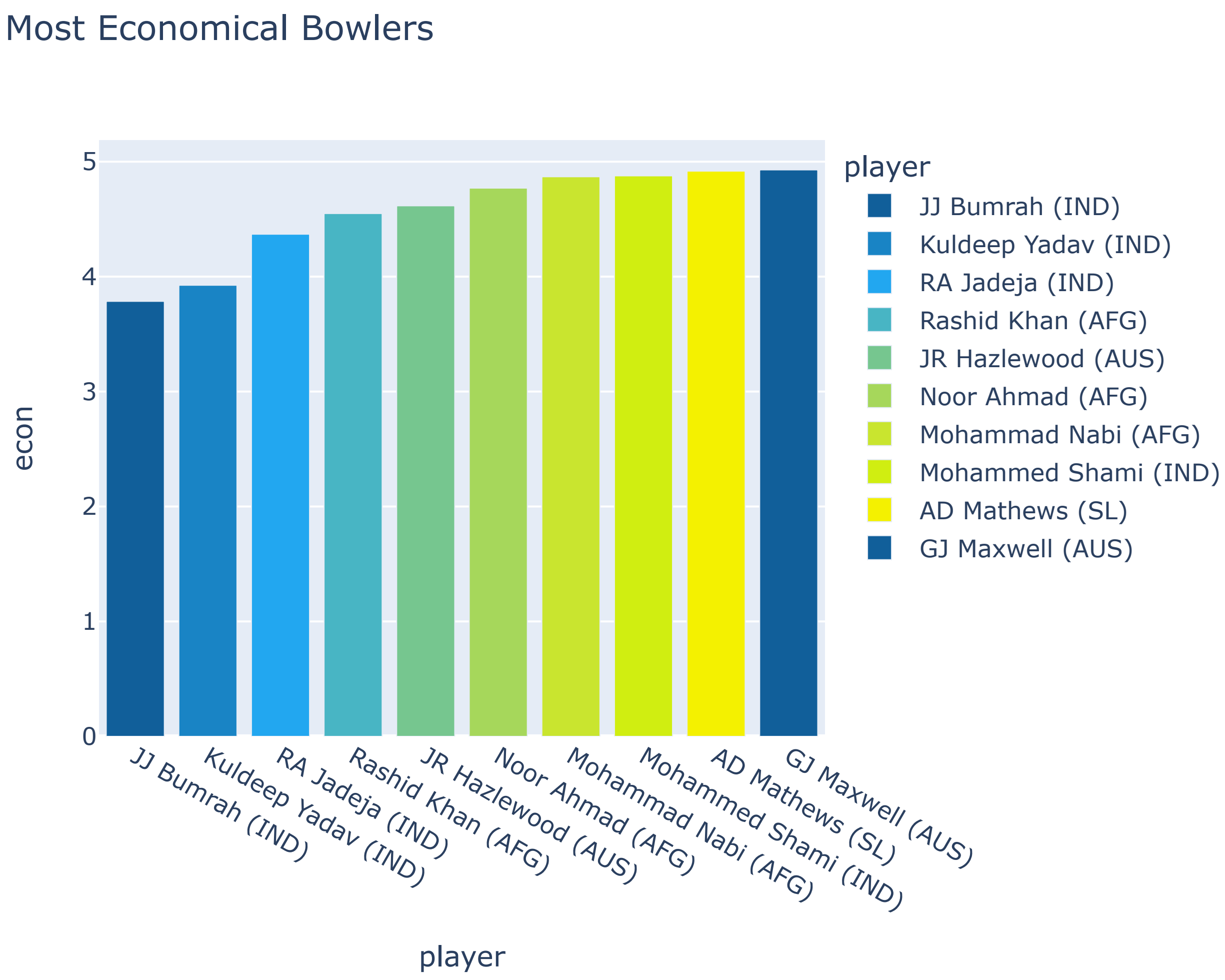
	player	wkts	overs	econ	mdns
4	JJ Bumrah (IND)	18.0	82.5	3.785000	7.0
12	Kuldeep Yadav (IND)	15.0	85.1	3.925000	2.0
9	RA Jadeja (IND)	16.0	82.9	4.369000	4.0
23	Rashid Khan (AFG)	11.0	86.3	4.548889	2.0
14	JR Hazlewood (AUS)	14.0	83.1	4.616000	8.0
55	Noor Ahmad (AFG)	5.0	38.0	4.770000	1.0
38	Mohammad Nabi (AFG)	8.0	61.3	4.868889	4.0
0	Mohammed Shami (IND)	23.0	41.5	4.876667	3.0
43	AD Mathews (SL)	6.0	22.1	4.918000	2.0
52	GJ Maxwell (AUS)	5.0	62.3	4.928571	1.0

In [18]:

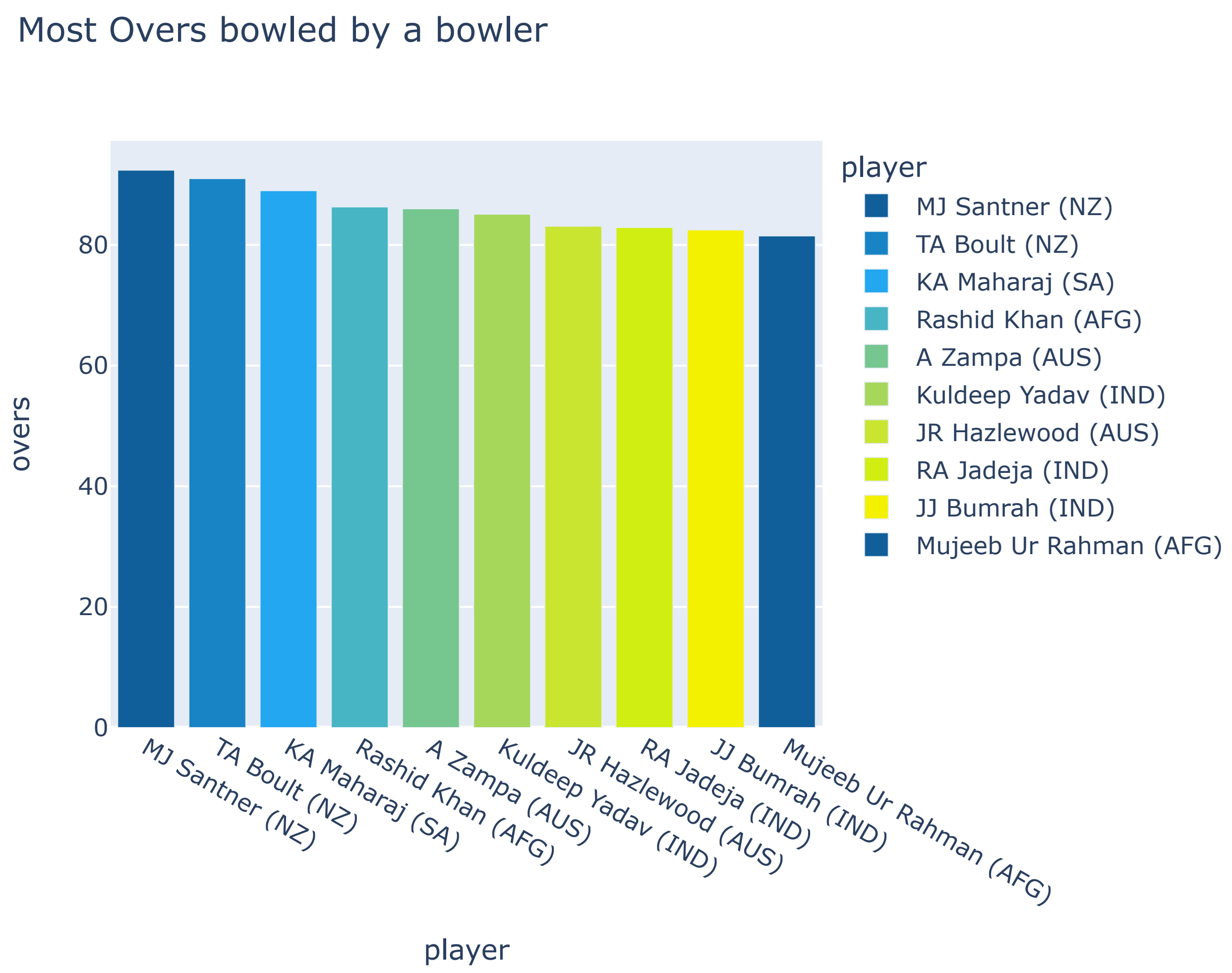
```
colorScheme = ["#115f9a", "#1984c5", "#22a7f0", "#48b5c4", "#76c68f", "#a6d75b", "#c9e52f", "#d0

fig = px.bar(playerBowlingPerformance.head(10), x='player', y='wkts', title='Highest Wicket Taker
fig.show()

fig = px.bar(playerBowlingPerformanceEconomy.head(10), x='player', y='econ', title='Most Economi
fig.show()
```

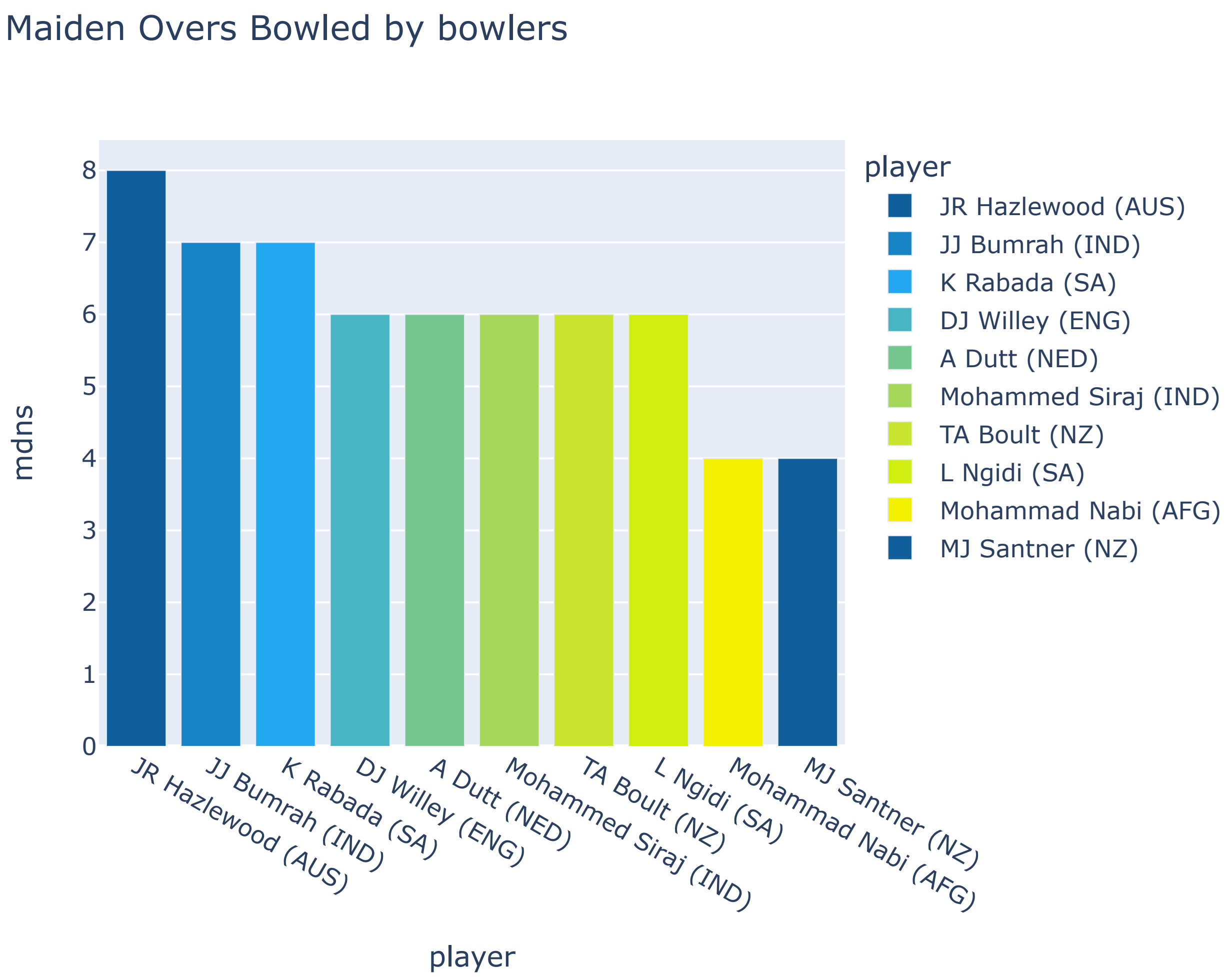
```
fig = px.bar(playerBowlingPerformanceOvers.head(10), x='player', y='overs', title='Most Overs bow
fig.show()

fig = px.bar(playerBowlingPerformanceMaidens.head(10), x='player', y='mdns', title='Maiden Overs
fig.show()
```





Most Economical Bowlers


Most Overs bowled by a bowler
overs
0
20
40
60
80
player
MJ Santner (NZ)
TA Boult (NZ)
KA Maharaj (SA)
Rashid Khan (AFG)
A Zampa (AUS)
Kuldeep Yadav (IND)
JR Hazlewood (AUS)
RA Jadeja (IND)
JJ Bumrah (IND)
Mujeeb Ur Rahman (AFG)
player



Bowling Analysis for Individual Player

Wickets

- **Mohammed Shami** was the the top wickets taker with *23 wickets* followed by Adam Zampa with 22 wickets and Dilshan Madushanka with 21 wickets.

Overs Bowled and Economy

- Mitchell Santner bowled 92 overs followed by Trent Boult (91 overs), both represent the New Zealand Team.
- Jasprit Bumrah (3.7) and Kuldeep Yadav (3.9) had the best economy among all the bowlers.

Overall, Mohammad Shami was the bowler of the tournament with 23 wickets at an economy of 4.8.

Player wise Batting Performance

In [19]:
```
playerBattingPerformance = data[data['bat_or_bowl'] == 'bat'].groupby('player').agg({
'bb_bf':'sum',
'runs':'sum',
'runs_per_ball':'mean',
'4s':'sum',
'6s':'sum',
'sr':'mean',
'not_out':'sum',
'mins':'sum'
})

playerBattingPerformance.sort_values('runs',ascending=False,inplace=True)
playerBattingPerformance.reset_index(inplace=True)
playerBattingPerformance.head(10)
```

Out[19]:

	player	bb_bf	runs	runs_per_ball	4s	6s	sr	not_out	mins
0	V Kohli (IND)	784	711	0.829561	64.0	9.0	82.956000	3.0	1097.0
1	Q de Kock (SA)	555	594	1.065390	57.0	21.0	106.539000	0.0	799.0
2	R Ravindra (NZ)	543	578	0.945670	55.0	17.0	94.567000	1.0	872.0
3	DJ Mitchell (NZ)	497	552	1.055422	48.0	22.0	105.542222	1.0	724.0
4	RG Sharma (IND)	443	550	1.255263	62.0	28.0	125.526000	0.0	617.0
5	DA Warner (AUS)	491	528	1.081995	49.0	24.0	108.199000	0.0	758.0
6	SS Iyer (IND)	465	526	0.930050	36.0	24.0	93.005000	3.0	648.0
7	HE van der Dussen (SA)	530	448	0.661140	39.0	8.0	66.114000	1.0	794.0
8	MR Marsh (AUS)	395	426	0.826112	42.0	20.0	82.611111	1.0	528.0
9	AK Markram (SA)	366	406	1.515250	44.0	9.0	151.525000	1.0	510.0

In [20]:

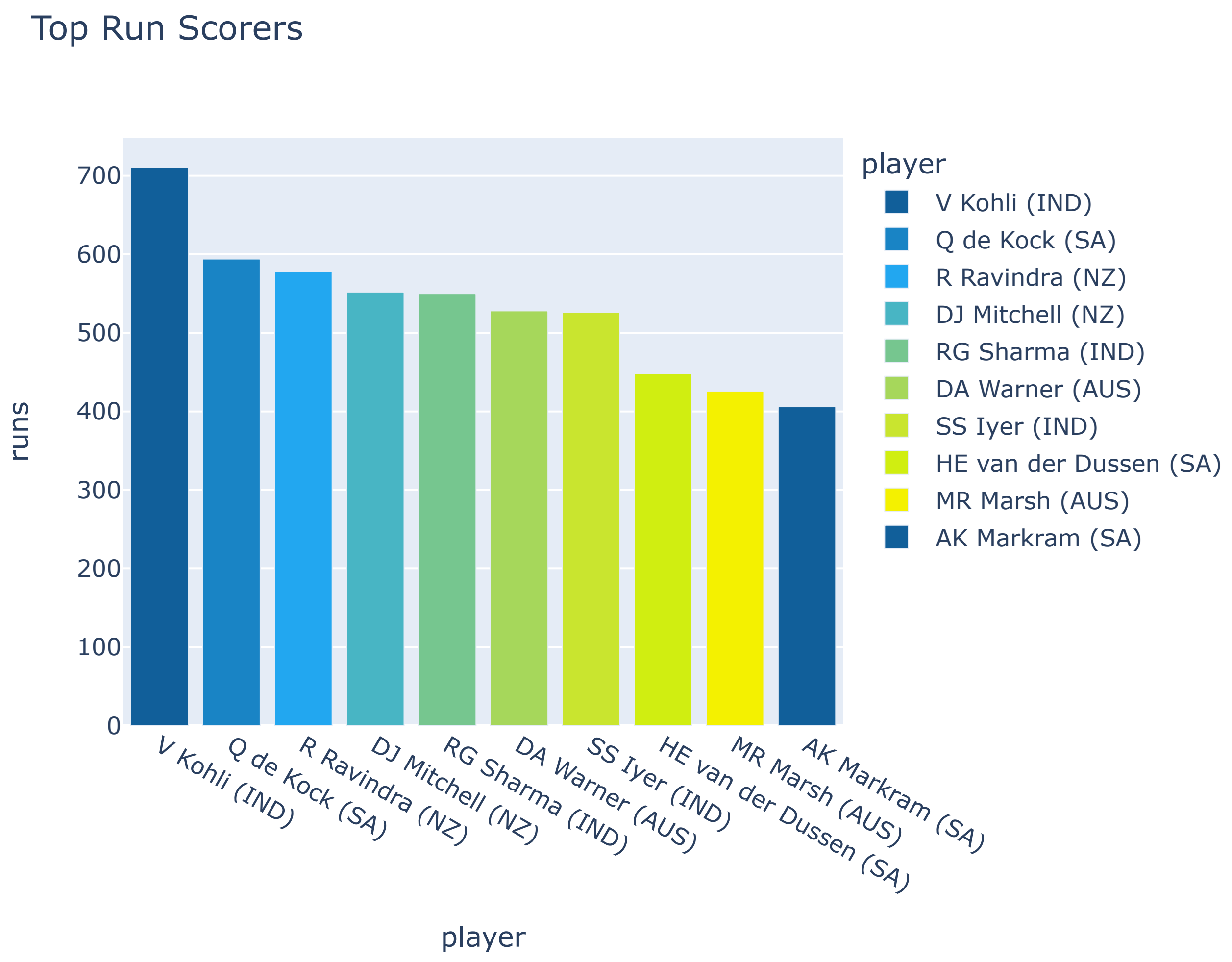
```
playerMaximum6s = playerBattingPerformance.sort_values('6s',ascending=False)
playerMaximum4s = playerBattingPerformance.sort_values('4s',ascending=False)
playerStrikeRate = playerBattingPerformance[playerBattingPerformance['bb_bf'] > 50].sort_values(
playerNotOutInnings = playerBattingPerformance[playerBattingPerformance['runs']>= 300].sort_valu
playerBattingPerformance[['player','runs','sr','4s','6s']].head(10)
```

Out[20]:

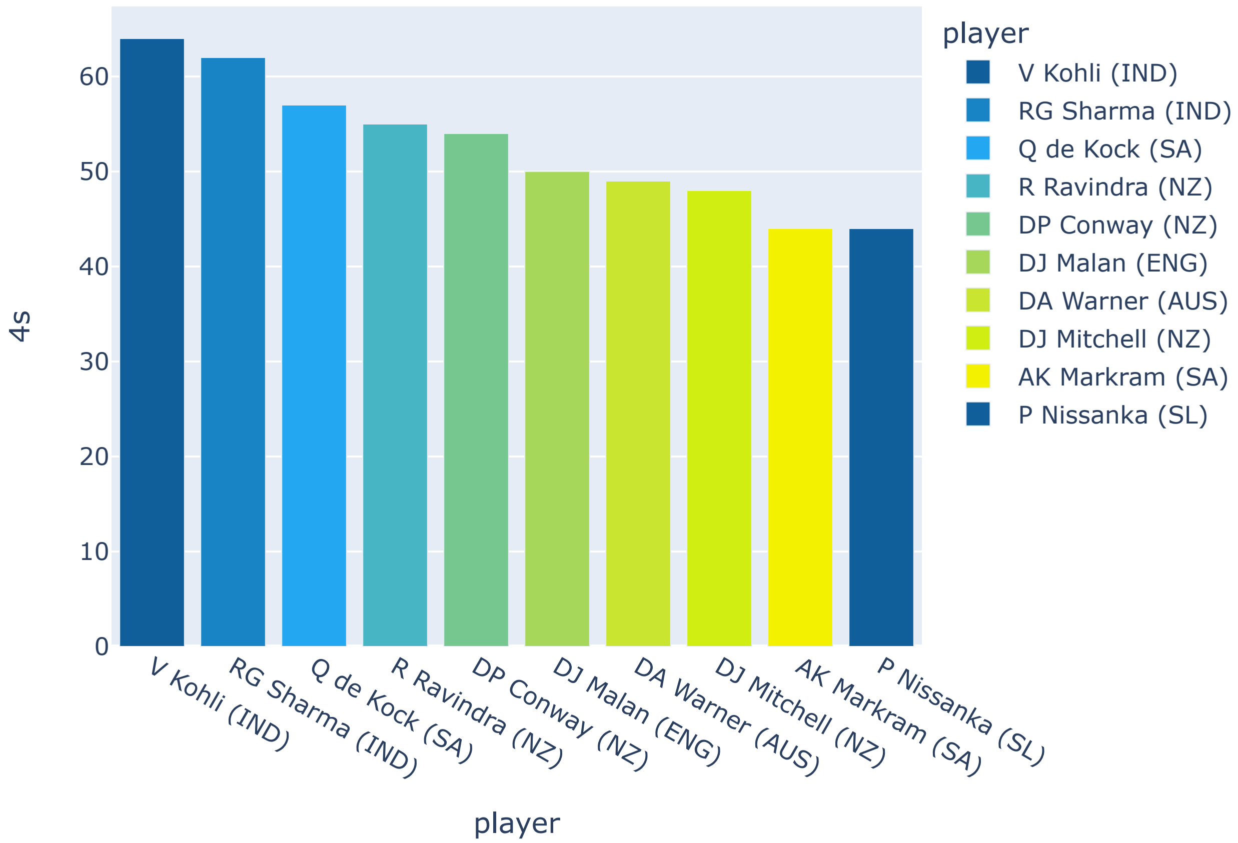
	player	runs	sr	4s	6s
0	V Kohli (IND)	711	82.956000	64.0	9.0
1	Q de Kock (SA)	594	106.539000	57.0	21.0
2	R Ravindra (NZ)	578	94.567000	55.0	17.0
3	DJ Mitchell (NZ)	552	105.542222	48.0	22.0
4	RG Sharma (IND)	550	125.526000	62.0	28.0
5	DA Warner (AUS)	528	108.199000	49.0	24.0
6	SS Iyer (IND)	526	93.005000	36.0	24.0
7	HE van der Dussen (SA)	448	66.114000	39.0	8.0
8	MR Marsh (AUS)	426	82.611111	42.0	20.0
9	AK Markram (SA)	406	151.525000	44.0	9.0

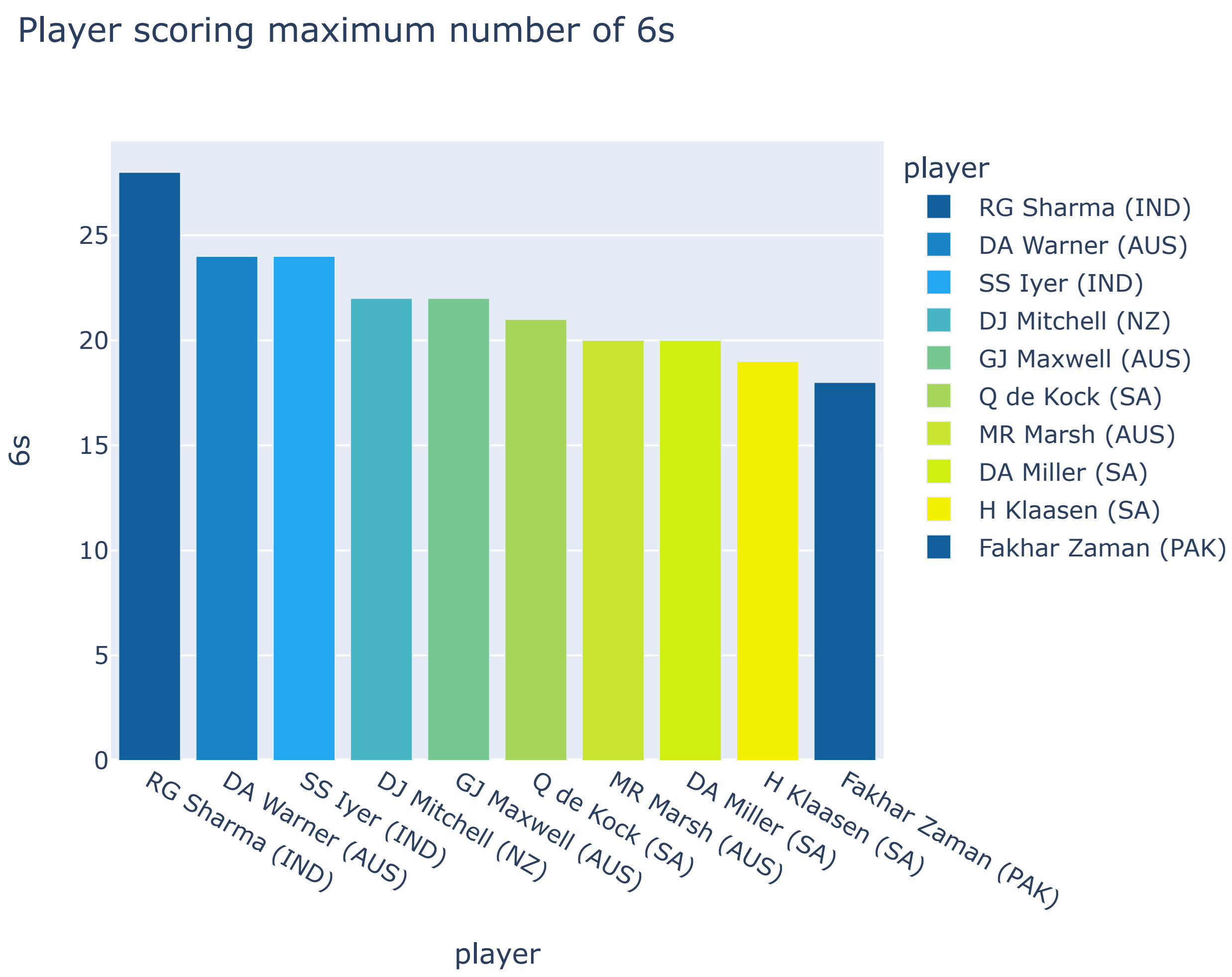
In [21]:

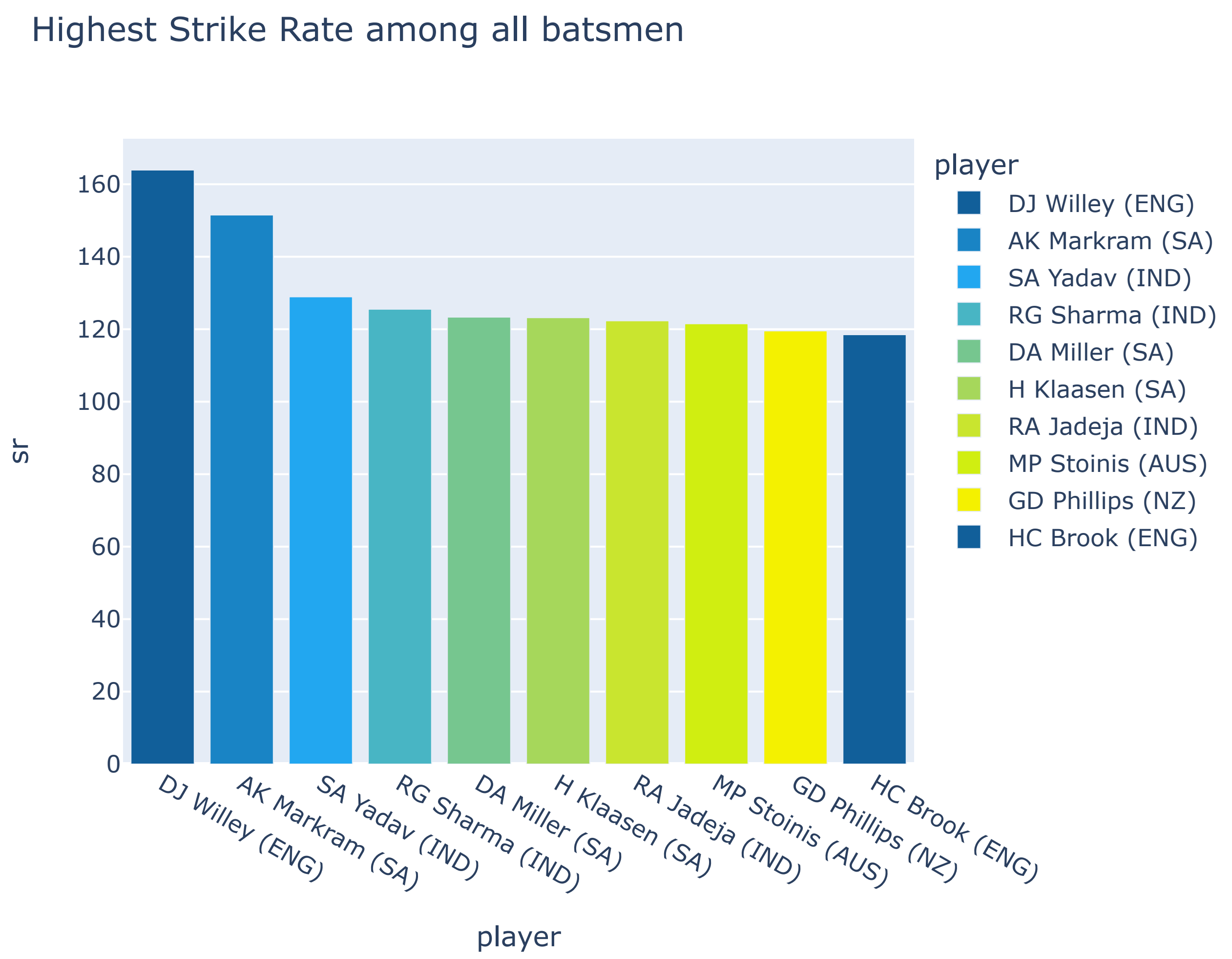
```
colorScheme = ["#115f9a", "#1984c5", "#22a7f0", "#48b5c4", "#76c68f", "#a6d75b", "#c9e52f", "#d0

fig = px.bar(playerBattingPerformance.head(10), x='player', y='runs', title='Top Run Scorers', co
fig.show()

fig = px.bar(playerMaximum4s.head(10), x='player', y='4s', title='Player scoring maximum number o
fig.show()

fig = px.bar(playerMaximum6s.head(10), x='player', y='6s', title='Player scoring maximum number o
fig.show()

fig = px.bar(playerStrikeRate.head(10), x='player', y='sr', title='Highest Strike Rate among all
fig.show()
```



Player scoring maximum number of 4s




Player scoring maximum number of 6s
6s
0
5
10
15
20
25
player
RG Sharma (IND)
DA Warner (AUS)
SS Iyer (IND)
DJ Mitchell (NZ)
GJ Maxwell (AUS)
Q de Kock (SA)
MR Marsh (AUS)
DA Miller (SA)
H Klaasen (SA)
Fakhar Zaman (PAK)
player
RG Sharma (IND)
DA Warner (AUS)
SS Iyer (IND)
DJ Mitchell (NZ)
GJ Maxwell (AUS)
Q de Kock (SA)
MR Marsh (AUS)
DA Miller (SA)
H Klaasen (SA)
Fakhar Zaman (PAK)



Batting Analysis for Individual Player

Runs Scored

- **Virat Kohli (India)** was the the top runs scorer with *711 runs* followed by Quinton de Kock with 594 runs and Rachin Ravindra with 578 runs.

Strike Rate and Boundries

- David Willey (England) had the best Strike Rate followed by Aiden Markram
- Virat Kohli (India) and Rohit Sharma(India) scored the maximum number of 4s and 6s respectively

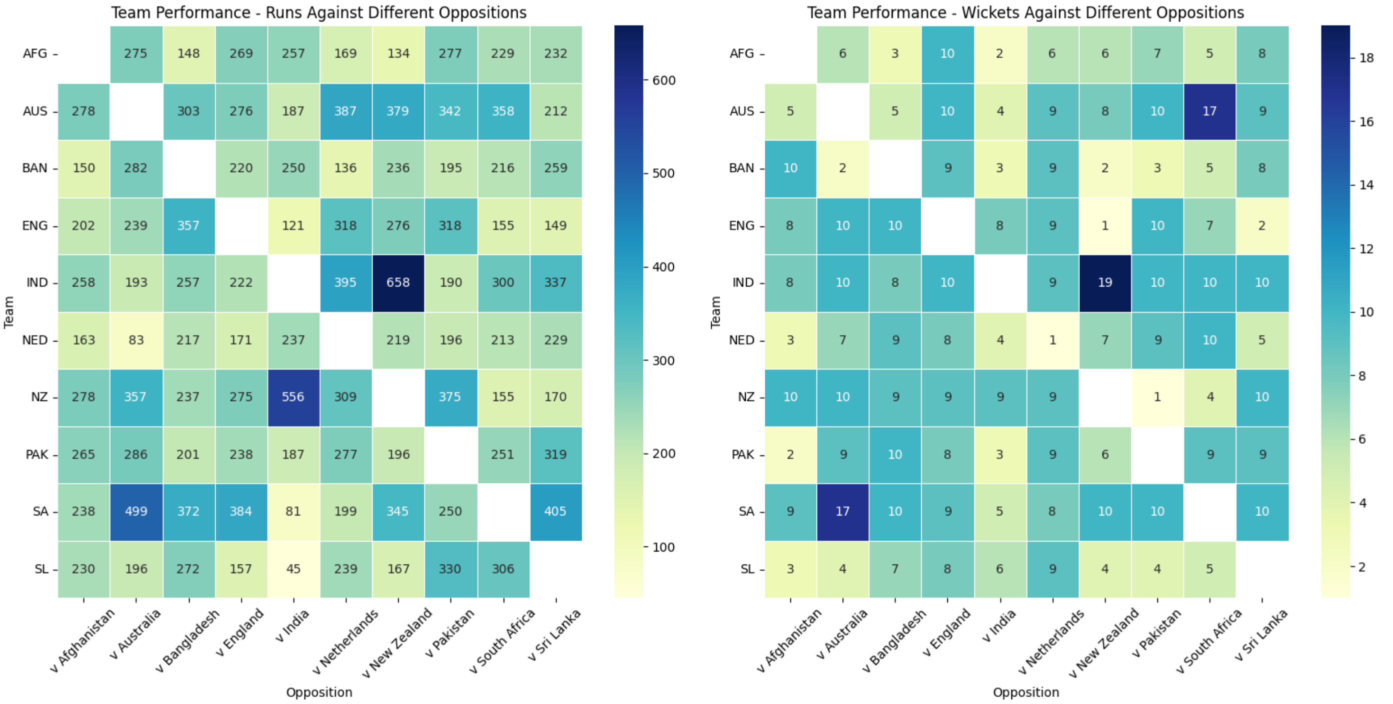
Overall, Virat Kohli was the Player of the Tournament with 711 runs and scoring maximum number of 4s

Opposition and Ground Analysis

Team wise Performance Against Opposition

In [22]:

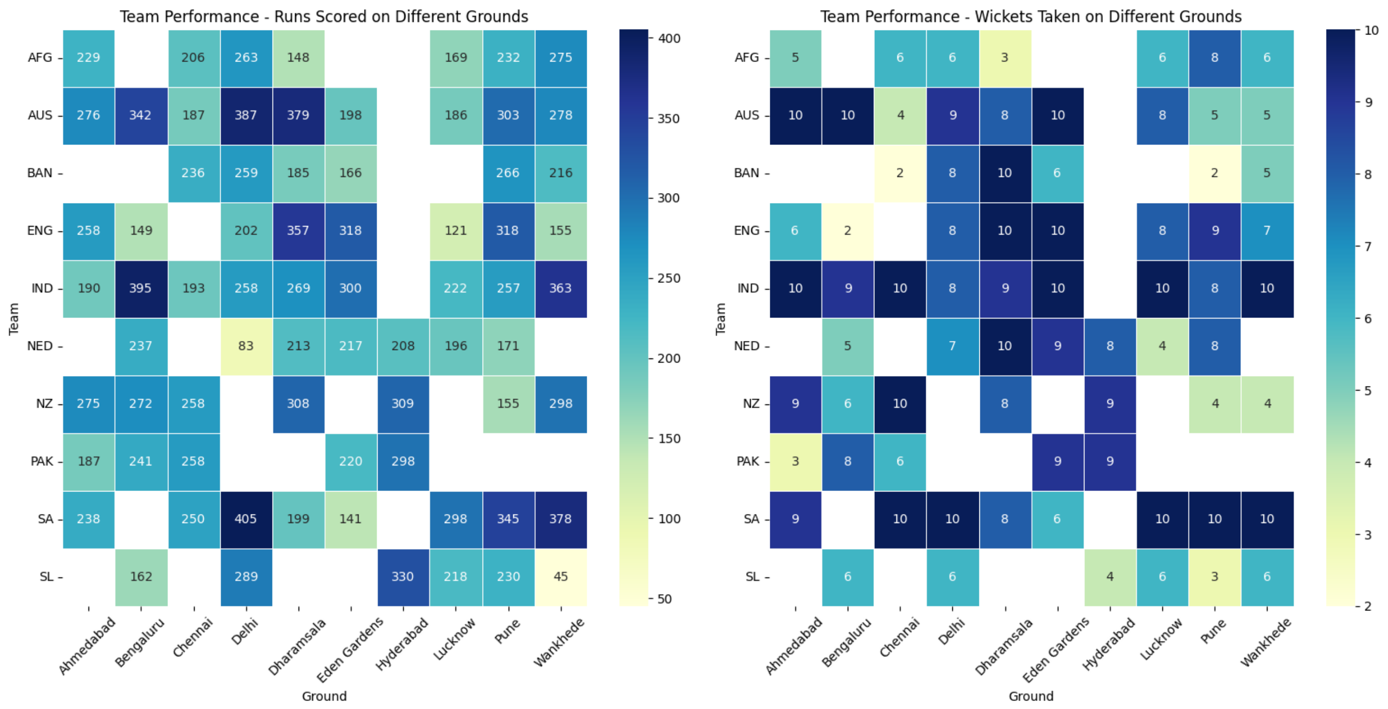
```
teamRuns = data[data['bat_or_bowl'] == 'bat'].groupby(['team', 'opposition']).agg({'runs': 'sum'
teamWickets = data[data['bat_or_bowl'] == 'bowl'].groupby(['team', 'opposition']).agg({'wkts': '
runsHeatmapData = teamRuns.pivot(index='team', columns='opposition', values='runs')
wicketsHeatmapData = teamWickets.pivot(index='team', columns='opposition', values='wkts')

fig, axes = plt.subplots(1, 2, figsize=(16, 8))

sns.heatmap(runsHeatmapData, cmap='YlGnBu', annot=True, fmt='g', linewidths=.5, ax=axes[0])
axes[0].set_title('Team Performance - Runs Against Different Oppositions')
axes[0].set_xlabel('Opposition')
axes[0].set_ylabel('Team')
axes[0].tick_params(axis='x', rotation=45)
axes[0].tick_params(axis='y', rotation=0)

sns.heatmap(wicketsHeatmapData, cmap='YlGnBu', annot=True, fmt='g', linewidths=.5, ax=axes[1])
axes[1].set_title('Team Performance - Wickets Against Different Oppositions')
axes[1].set_xlabel('Opposition')
axes[1].set_ylabel('Team')
axes[1].tick_params(axis='x', rotation=45)
axes[1].tick_params(axis='y', rotation=0)

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



Team wise Performance in each Cricket Ground Venues

In [23]:
```
teamRunsPerGround = data[data['bat_or_bowl'] == 'bat'].groupby(['team', 'ground', 'start_date'])
teamWicketsPerGround = data[data['bat_or_bowl'] == 'bowl'].groupby(['team', 'ground', 'start_date

teamRunsPerGround = teamRunsPerGround.groupby(['team', 'ground'])[['runs']].mean().reset_index()
teamRunsPerGround['runs'] = round(teamRunsPerGround['runs'])
teamWicketsPerGround = teamWicketsPerGround.groupby(['team', 'ground'])[['wkts']].mean().reset_i
teamWicketsPerGround['wkts'] = round(teamWicketsPerGround['wkts'])

runsPerGroundHeatmap = teamRunsPerGround.pivot(index='team', columns='ground', values='runs')
wicketsPerGroundHeatmap = teamWicketsPerGround.pivot(index='team', columns='ground', values='wkt

fig, axes = plt.subplots(1, 2, figsize=(16, 8))
```

```
sns.heatmap(runsPerGroundHeatmap, cmap='YlGnBu', annot=True, fmt='g', linewidths=.5, ax=axes[0])
axes[0].set_title('Team Performance - Runs Scored on Different Grounds')
axes[0].set_xlabel('Ground')
axes[0].set_ylabel('Team')
axes[0].tick_params(axis='x', rotation=45)
axes[0].tick_params(axis='y', rotation=0)

sns.heatmap(wicketsPerGroundHeatmap, cmap='YlGnBu', annot=True, fmt='g', linewidths=.5, ax=axes[
axes[1].set_title('Team Performance - Wickets Taken on Different Grounds')
axes[1].set_xlabel('Ground')
axes[1].set_ylabel('Team')
axes[1].tick_params(axis='x', rotation=45)
axes[1].tick_params(axis='y', rotation=0)

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

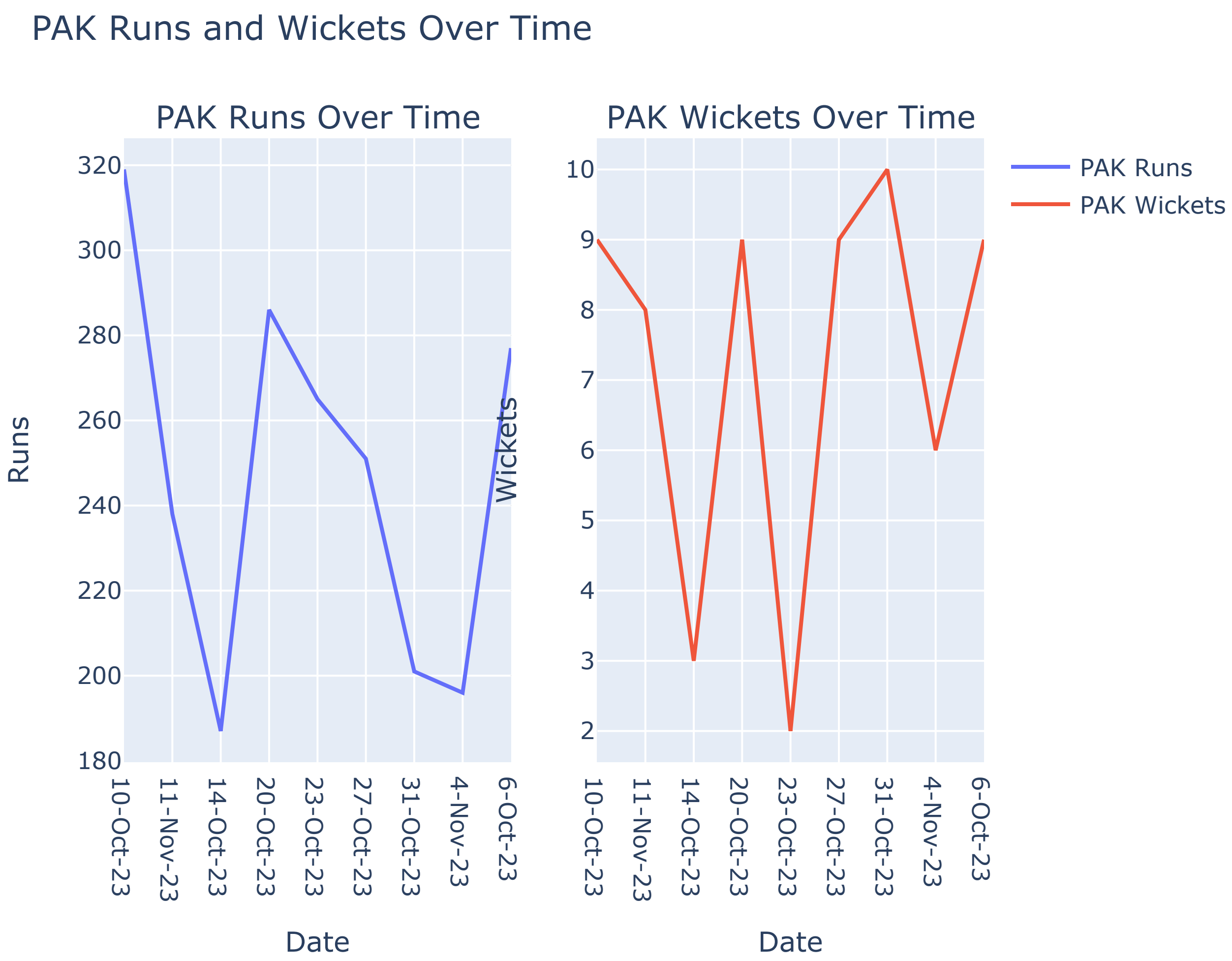


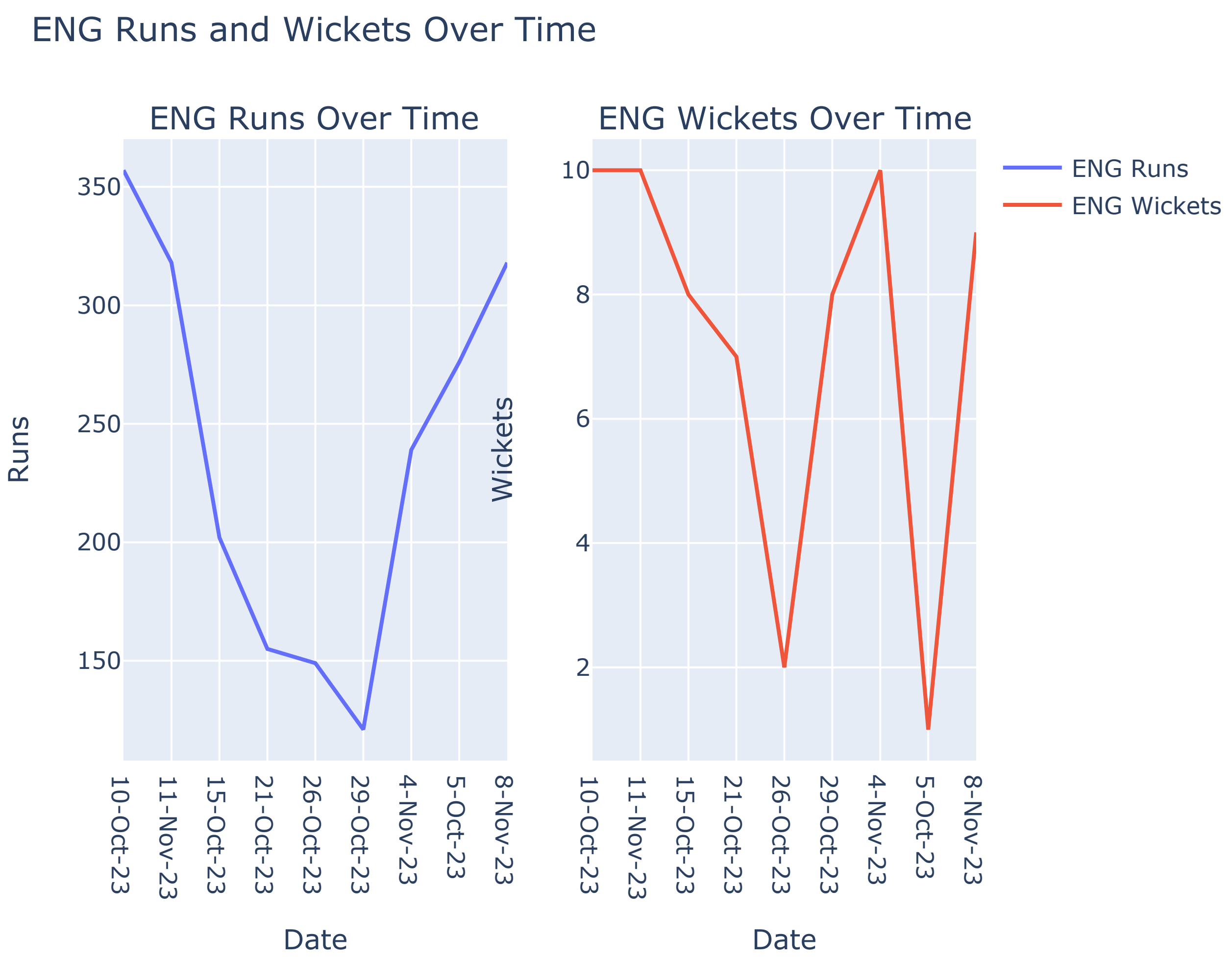
Performance trends of each team in terms of runs scored and wickets taken over the entire tournament's duration

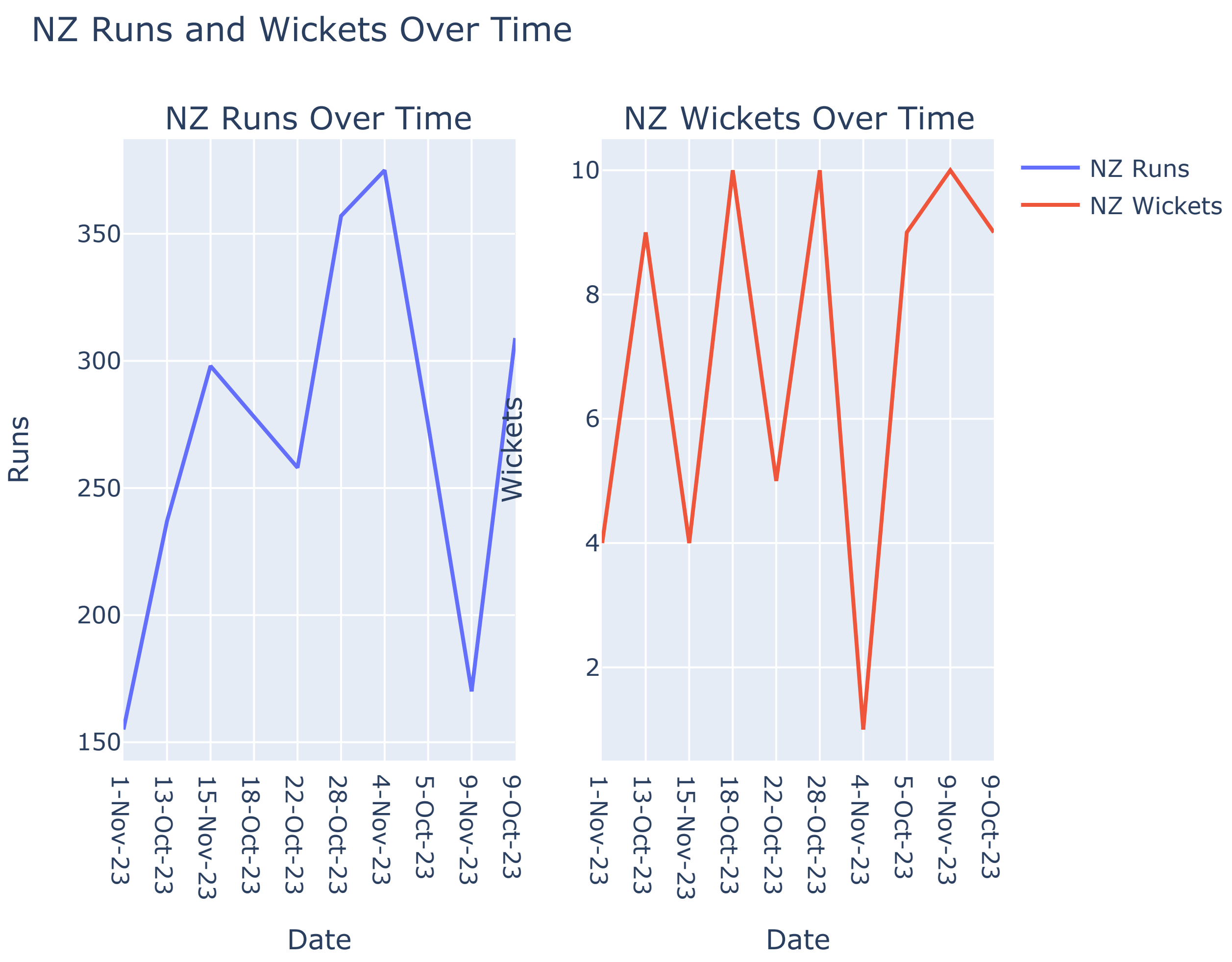
In [24]:

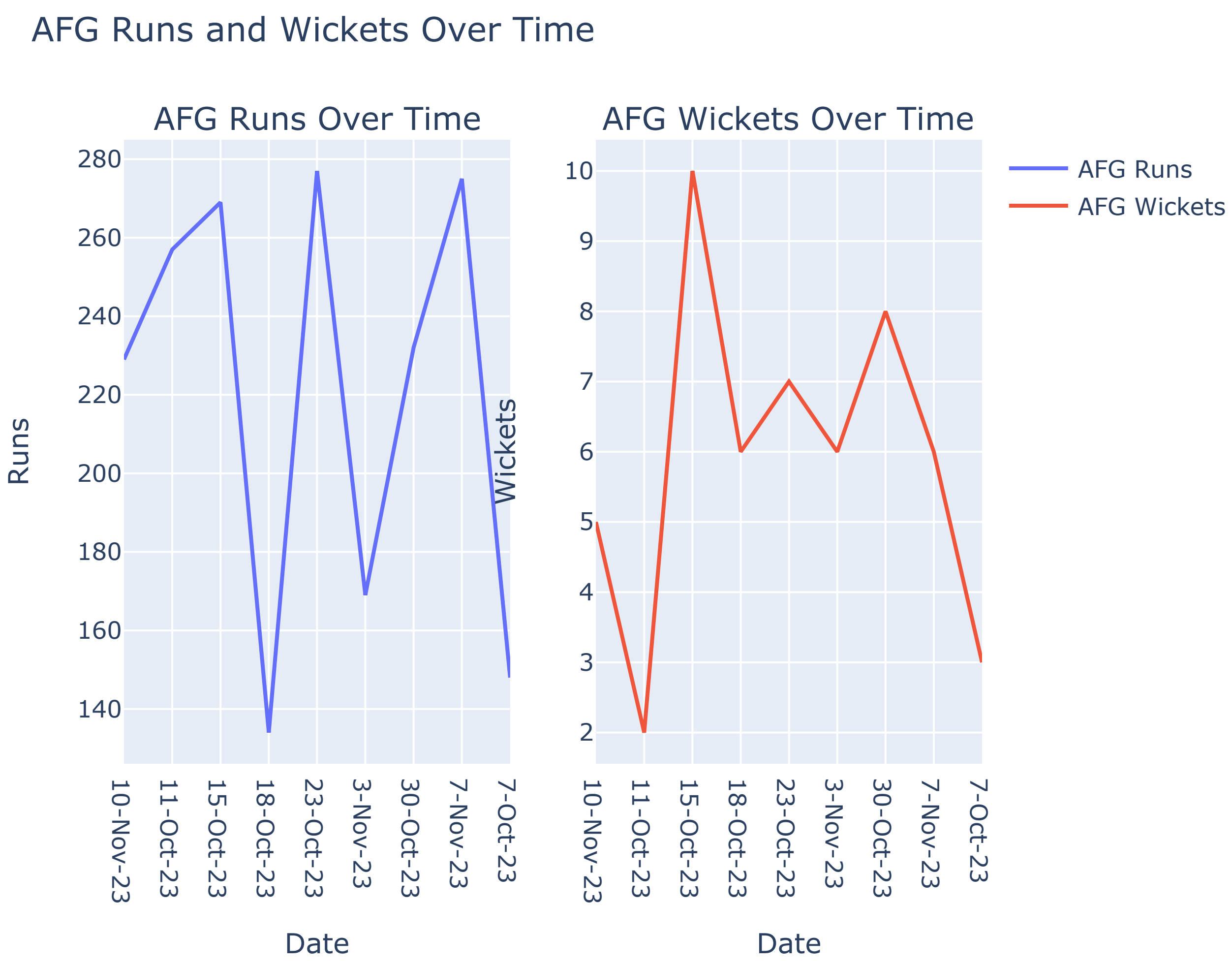
```
def plotPerformance(teams, battingData, bowlingData):
    fig = make_subplots(rows=1, cols=2, subplot_titles=(f'{teams} Runs Over Time', f'{teams} Wick
    fig.add_trace(go.Scatter(x=battingData['start_date'], y=battingData['runs'], mode='lines', na
    fig.add_trace(go.Scatter(x=bowlingData['start_date'], y=bowlingData['wkts'], mode='lines', na

    fig.update_layout(title=f'{teams} Runs and Wickets Over Time')
    fig.update_xaxes(title_text='Date', row=1, col=1)
    fig.update_xaxes(title_text='Date', row=1, col=2)
    fig.update_yaxes(title_text='Runs', row=1, col=1)
    fig.update_yaxes(title_text='Wickets', row=1, col=2)

    fig.show()

def teamPerformance(data):
  battingData = data[data['bat_or_bowl'] == 'bat'].groupby(['start_date'])[['runs']].sum().reset_
  bowlingData = data[data['bat_or_bowl'] == 'bowl'].groupby(['start_date'])[['wkts']].sum().rese

  return battingData, bowlingData
```

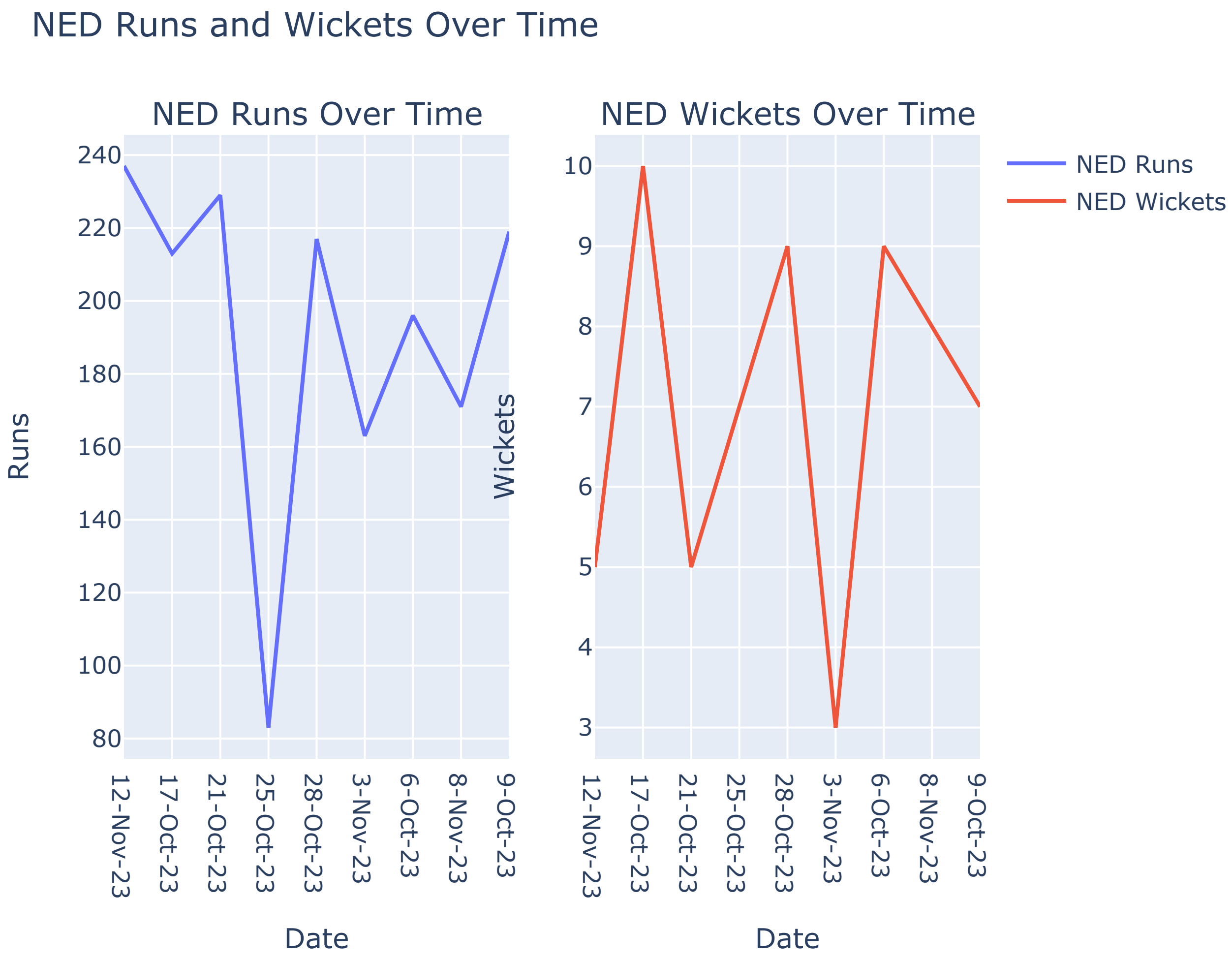
```
uniqueTeams = data['team'].unique()
for teams in uniqueTeams:
    teamData = data[data['team'] == teams]
    batting, bowling = teamPerformance(teamData)
    plotPerformance(teams, batting, bowling)
```

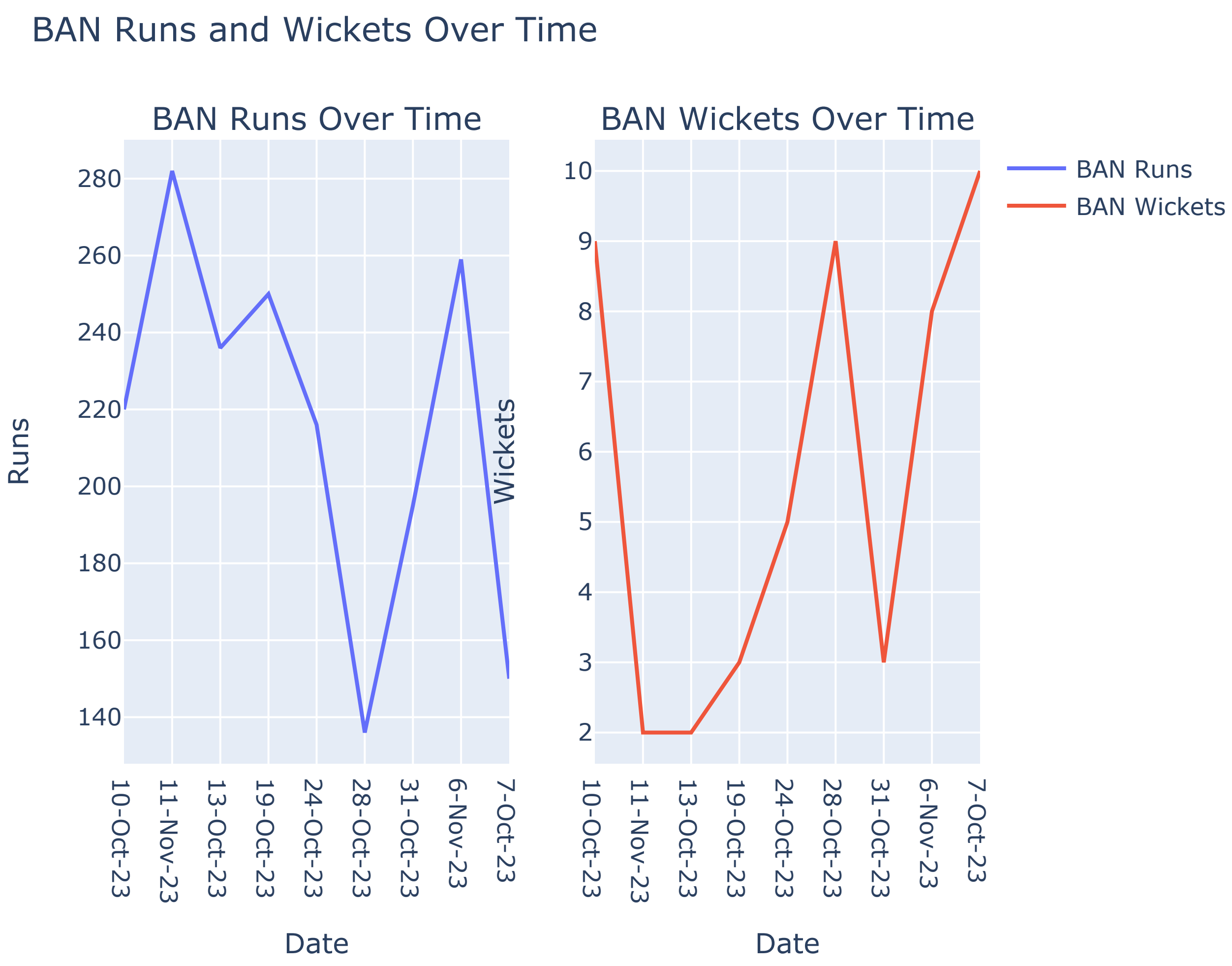


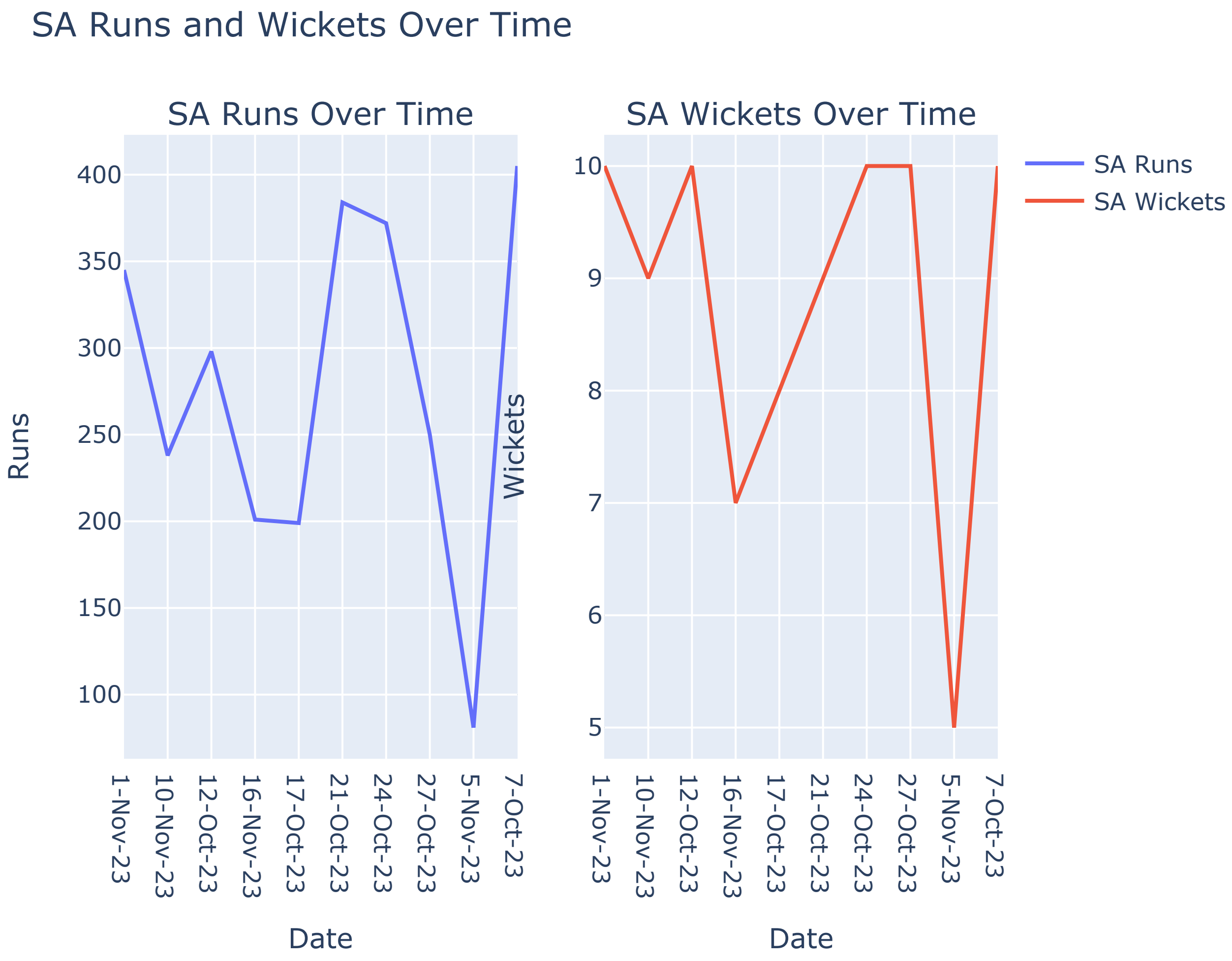

ENG Runs and Wickets Over Time
ENG Runs Over Time
ENG Wickets Over Time
Runs
350
300
250
200
150
Wickets
10
8
6
4
2
10-Oct-23
11-Nov-23
15-Oct-23
21-Oct-23
26-Oct-23
29-Oct-23
4-Nov-23
5-Oct-23
8-Nov-23
Date
ENG Runs
ENG Wickets

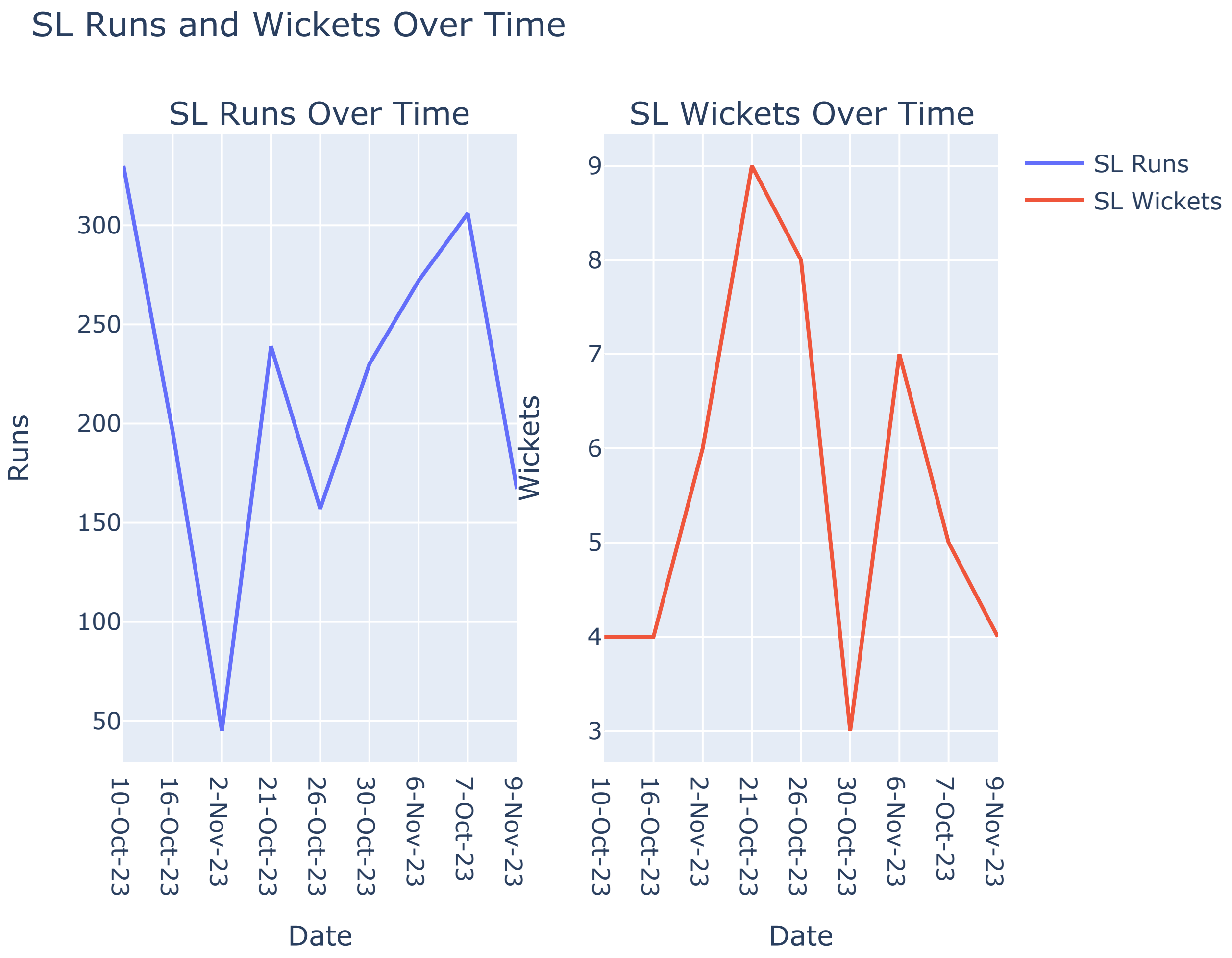

NZ Runs and Wickets Over Time
NZ Runs Over Time
NZ Wickets Over Time
Runs
Wickets
350
300
250
200
150
10
8
6
4
2
1-Nov-23
13-Oct-23
15-Nov-23
18-Oct-23
22-Oct-23
28-Oct-23
4-Nov-23
5-Oct-23
9-Nov-23
9-Oct-23
Date
NZ Runs
NZ Wickets

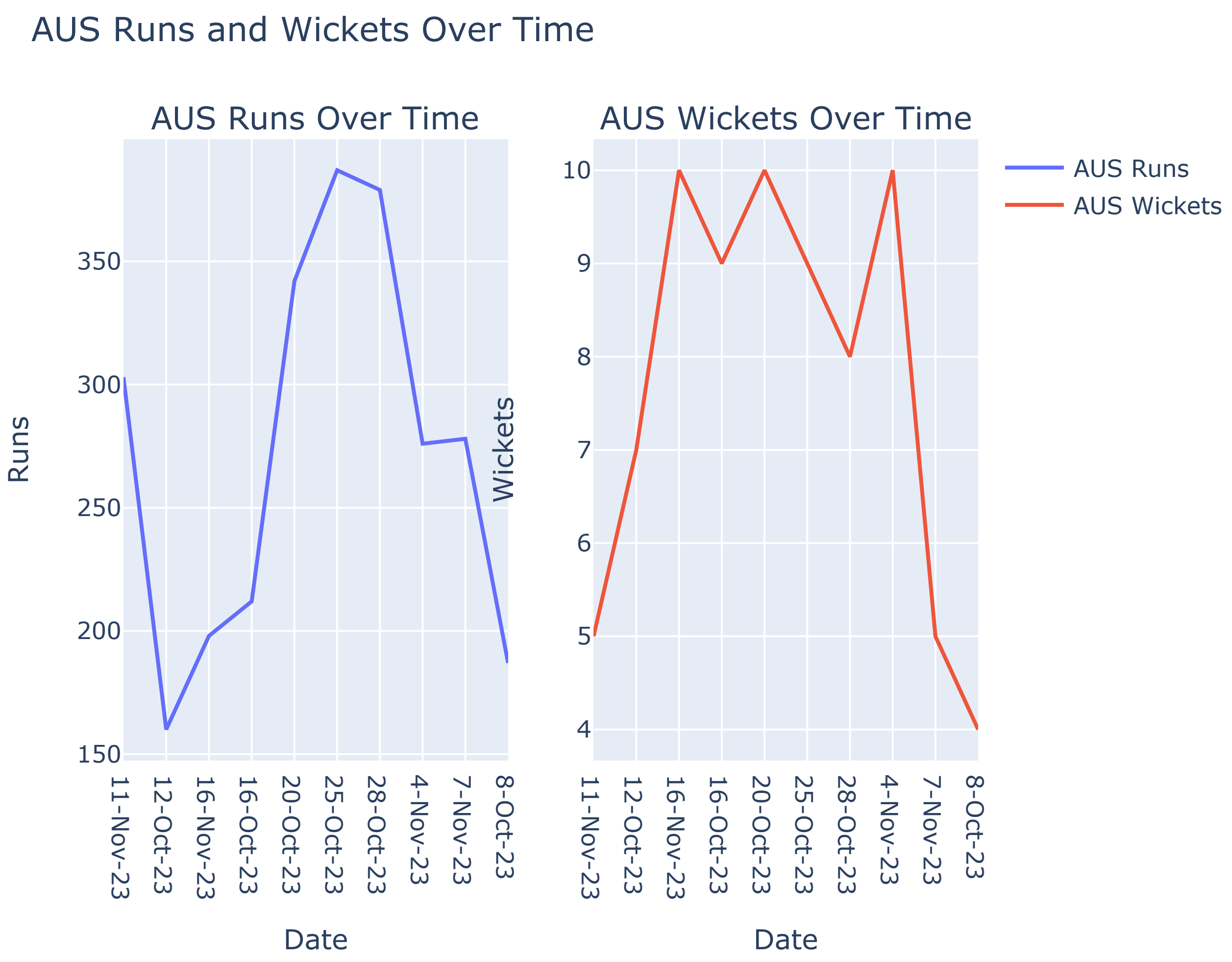

AFG Runs and Wickets Over Time
AFG Runs Over Time
AFG Wickets Over Time
AFG Runs
AFG Wickets
Runs
280
260
240
220
200
180
160
140
Wickets
10
9
8
7
6
5
4
3
2
10-Nov-23
11-Oct-23
15-Oct-23
18-Oct-23
23-Oct-23
3-Nov-23
30-Oct-23
7-Nov-23
7-Oct-23
Date
10-Nov-23
11-Oct-23
15-Oct-23
18-Oct-23
23-Oct-23
3-Nov-23
30-Oct-23
7-Nov-23
7-Oct-23
Date

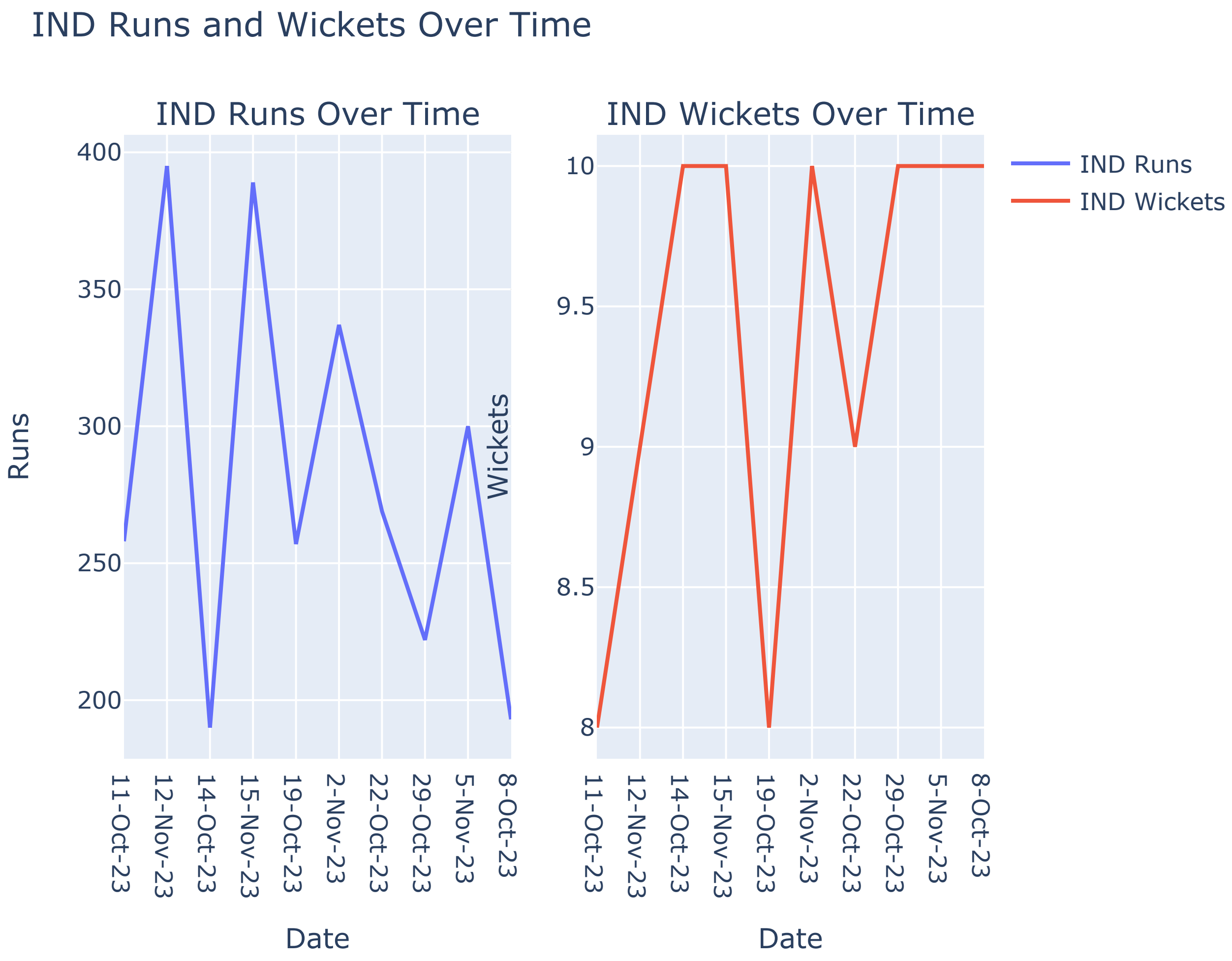

NED Runs and Wickets Over Time
NED Runs Over Time
NED Wickets Over Time
Runs
240
220
200
180
160
140
120
100
80
Wickets
10
9
8
7
6
5
4
3
12-Nov-23
17-Oct-23
21-Oct-23
25-Oct-23
28-Oct-23
3-Nov-23
6-Oct-23
8-Nov-23
9-Oct-23
Date
NED Runs
NED Wickets


BAN Runs and Wickets Over Time
BAN Runs Over Time
BAN Wickets Over Time
Runs
280
260
240
220
200
180
160
140
Wickets
10
9
8
7
6
5
4
3
2
10-Oct-23
11-Nov-23
13-Oct-23
19-Oct-23
24-Oct-23
28-Oct-23
31-Oct-23
6-Nov-23
7-Oct-23
Date
BAN Runs
BAN Wickets


SA Runs and Wickets Over Time
SA Runs Over Time
SA Wickets Over Time
Runs
400
350
300
250
200
150
100
Wickets
10
9
8
7
6
5
1-Nov-23
10-Nov-23
12-Oct-23
16-Nov-23
17-Oct-23
21-Oct-23
24-Oct-23
27-Oct-23
5-Nov-23
7-Oct-23
Date
Date
SA Runs
SA Wickets


SL Runs and Wickets Over Time
SL Runs Over Time
SL Wickets Over Time
SL Runs
SL Wickets
Runs
300
250
200
150
100
50
Wickets
9
8
7
6
5
4
3
10-Oct-23
16-Oct-23
2-Nov-23
21-Oct-23
26-Oct-23
30-Oct-23
6-Nov-23
7-Oct-23
9-Nov-23
Date


AUS Runs and Wickets Over Time
AUS Runs Over Time
AUS Wickets Over Time
Runs
350
300
250
200
150
Wickets
10
9
8
7
6
5
4
11-Nov-23
12-Oct-23
16-Nov-23
16-Oct-23
20-Oct-23
25-Oct-23
28-Oct-23
4-Nov-23
7-Nov-23
8-Oct-23
Date
AUS Runs
AUS Wickets



Overall, we can conclude that the top 4 consistent teams were **India, Australia, South Africa and New Zealand**, wherein the team performed well in the batting as well as the bowling departments

__Australia__ **were the winner of the 2023 Cricket World Cup and the runner ups were** *__India.__*