

Sachin Tendulkar - 100 Centuries EDA (Exploratory Data Analysis)

About Data Set

Sachin Ramesh Tendulkar is an Indian former international cricketer who captained the Indian national team. He is regarded as one of the greatest batsmen in the history of cricket. He was a right-handed top-order batsman in the Indian Cricket Team. He is known for his batting skills, technique, vision, and game reading. He is the all-time most runscorer in both ODI and Test Format with more than 18000 runs and 15000 runs respectively in total. He also holds the record for receiving the most Man-of-the-Match awards in international Cricket in all forms combined. He is the only batsman in the world to score 100 international centuries. This dataset consists of all the 100s scored by Sachin during his international career.

The dataset description is as follows:

S.No: International Century Number

Score: Number of runs scored by Sachin in the match

Dismissed: Whether Sachin is dismissed or not?

Captain: Is Sachin the captain of the match?

POTM: Did Sachin win the player of the match for his performance?

Against: The opposition team

Position: Batting position of Sachin in that match

Innings: In which innings of the match did Sachin score the century?

Test: Test match number in the series

Strike Rate: Number of runs scored per 100 balls

Venue: The stadium where the match is played

City: The city in which the stadium is present

H-A: Is the game played at home or away or at a neutral venue?

Date: The date on which Sachin scored that century

Result: Result of the match

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
In [3]: df = pd.read_csv("Sachin Tendulkar - 100 Centuries(1).csv")
df.head(10)
```

91	92	106	Yes	No	No	South Africa	4	2	2.0	NaN	Eden Gardens	Kolkata	Home	2/14/2010	Won
92	93	200	No	No	Yes	South Africa	2	1	NaN	136.05	Roop Singh Stadium	Gwalior	Home	2/24/2010	Won
93	94	203	Yes	No	No	Sri Lanka	4	2	2.0	NaN	Sinhalese Sports Club Ground	Colombo	Away	7/26/2010	Drawn
94	95	214	Yes	No	Yes	Australia	4	2	2.0	NaN	M. Chinnaswamy Stadium	Bangalore	Home	10/9/2010	Won
95	96	111	No	No	No	South Africa	5	3	1.0	NaN	SuperSport Park	Centurion	Away	12/16/2010	Lost
96	97	146	Yes	No	No	South Africa	4	2	3.0	NaN	Newlands Cricket Ground	Cape Town	Away	1/2/2011	Drawn
97	98	120	Yes	No	No	England	2	1	NaN	104.34	M. Chinnaswamy Stadium	Bangalore	Home	2/27/2011	Tied
98	99	111	Yes	No	No	South Africa	2	1	NaN	109.90	VCA Stadium	Nagpur	Home	3/12/2011	Lost
99	100	114	Yes	No	No	Bangladesh	2	1	NaN	77.55	Sher-e-Bangla National Stadium	Mirpur	Away	3/16/2012	Lost

In [5]: df.info()

<class 'pandas.core.frame.DataFrame'>

```
In [4]: df.tail(10)
```

9

Strike Rate

49 non-null

float64

10

Venue

100 non-null

object

11

City

100 non-null

object

12

H-A

100 non-null

object

13

Date

100 non-null

object

14

Result

100 non-null

object

dtypes: float64(2), int64(4), object(9)

memory usage: 11.8+ KB

In [7]:

df.columns

Out[7]:

Index(['S.No.', 'Score', 'Dismissed', 'Captain', 'POTM', 'Against', 'Position', 'Innings', 'Test', 'Strike Rate', 'Venue', 'City', 'H-A', 'Date', 'Result'], dtype='object')

In [8]:

df.describe()

Out[8]:

	S.No.	Score	Position	Innings	Test	Strike Rate
count	100.000000	100.000000	100.000000	100.000000	51.000000	49.000000
mean	50.500000	135.57000	3.140000	1.630000	2.000000	100.810000
std	29.011492	33.33038	1.206213	0.786952	0.979796	16.102412
min	1.000000	100.00000	1.000000	1.000000	1.000000	72.140000
25%	25.750000	111.00000	2.000000	1.000000	1.000000	88.490000
50%	50.500000	124.00000	4.000000	1.000000	2.000000	97.700000
75%	75.250000	149.00000	4.000000	2.000000	2.500000	110.150000
max	100.000000	248.00000	6.000000	4.000000	5.000000	138.610000

```
In [5]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype  
---  --
0   S.No.                 100 non-null    int64  
1   Score                 100 non-null    int64  
2   Dismissed             100 non-null    object  
3   Captain               100 non-null    object  
4   POTM                  100 non-null    object  
5   Against               100 non-null    object  
6   Position              100 non-null    int64  
7   Innings               100 non-null    int64  
8   Test                  51 non-null     float64 
9   Strike Rate           49 non-null     float64 
10  Venue                 100 non-null    object  
11  City                  100 non-null    object  
12  H-A                   100 non-null    object  
13  Date                  100 non-null    object  
14  Result                100 non-null    object  
dtypes: float64(2), int64(4), object(9)
memory usage: 11.8+ KB
```

```
In [7]: df.columns

Out[7]: Index(['S.No.', 'Score', 'Dismissed', 'Captain', 'POTM', 'Against', 'Position', 'Innings', 'Test', 'Strike Rate', 'Venue', 'City', 'H-A', 'Date', 'Result', 'dtype': 'object'])
```

```
In [8]: df.describe()
```

Out[8]:	S.No.	Score	Position	Innings	Test	Strike Rate
	count	100.000000	100.000000	100.000000	51.000000	49.000000
	mean	50.500000	135.570000	3.140000	1.630000	2.000000
	std	29.011492	33.33038	1.206213	0.786952	0.979796
	min	1.000000	100.00000	1.000000	1.000000	72.140000
	25%	25.750000	111.00000	2.000000	1.000000	88.490000
	50%	50.500000	124.00000	4.000000	1.000000	2.000000
	75%	75.250000	149.00000	4.000000	2.000000	2.500000
	max	100.000000	248.00000	6.000000	4.000000	5.000000

Observation:

1. His average score has been 135 and average strike-rate has been 100.81
2. His highest score has been 248 and highest strike-rate was 138.61

In Data Analysis What All Things We Do

1. Missing values.
2. Explore about the Numerical Variables.
3. Explore about Categorical Variables.
4. Finding Relationship between features.

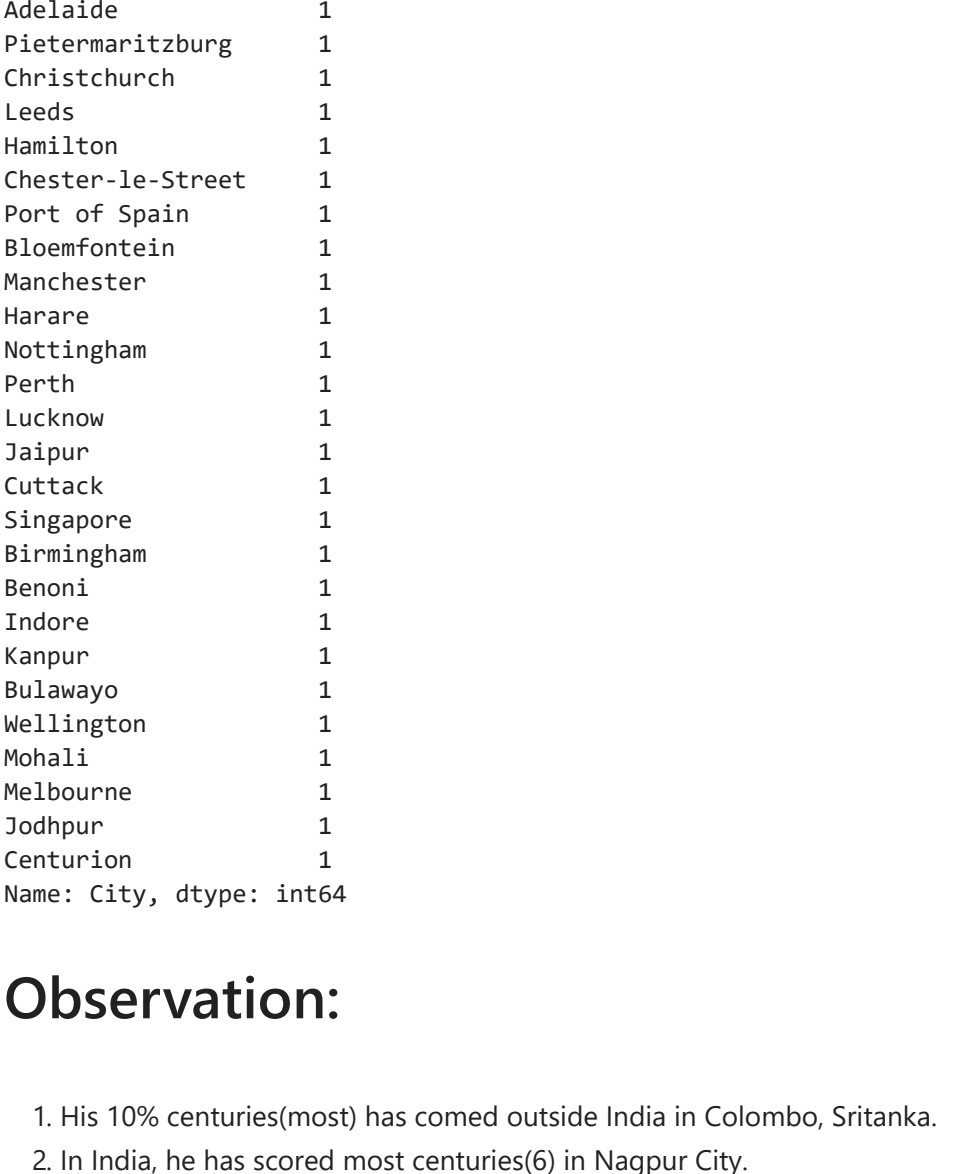
```
In [9]: df.isnull().sum()

Out[9]: S.No.      0
Score      0
Dismissed   0
Captain     0
POTM        0
Against     0
Position    0
Innings     0
Test        49
Strike Rate  51
Venue       0
City        0
H-A         0
Date        0
Result      0
dtype: int64
```

Heat Map of above Data

```
In [10]: import matplotlib
sns.heatmap(df.isnull(),yticklabels=False,cbar=True, cmap="viridis")
```

```
Out[10]: <AxesSubplot: >
```



Observation:

1. In 49 innings, his test match no. was not available.
2. In 51 innings, his strike rate was not available.

```
In [11]: df.City.value_counts()
```

Out[11]:	Colombo	10
	Sharjah	7
	Nagpur	6
	Chennai	5
	Ahmedabad	4
	Bangalore	4
	Hyderabad	3
	Vadodara	3
	Mirpur	3
	New Delhi	3
	Kolkata	3
	Bristol	2
	Cape Town	2
	Dhaka	2
	Mumbai	2
	Chittagong	2
	Johannesburg	2
	Paarl	1
	Multan	1
	Peshawar	1
	Rawalpindi	1
	Kuala Lumpur	1
	Pietermaritzburg	1
	Christchurch	1
	Leeds	1
	Hamilton	1
	Chester-le-Street	1
	Port of Spain	1
	Bloemfontein	1
	Manchester	1
	Harare	1
	Nottingham	1
	Perth	1
	Lucknow	1
	Jaipur	1
	Cuttack	1
	Singapore	1
	Birmingham	1
	Benoni	1
	Indore	1
	Kanpur	1
	BuLawayo	1
	Wellington	1
	Mohali	1
	Melbourne	1
	Jodhpur	1
	Centurion	1
	Name: City, dtype: int64	

Observation:

1. His 10% centuries(most) has come outside India in Colombo, Sri Lanka.
2. In India, he has scored most centuries(5) in Nagpur City.
3. He has scored only 2 centuries at his hometown i.e. Mumbai.

```
In [12]: df.Venue.value_counts()
```

Observation:

1. Sachin has scored most centuries against team Australia i.e. '20'.
2. He was able to score 1 century against team Namibia.
3. He has scored 50% centuries against the SENA countries (South Africa England,New Zealand Australia).

```
[14]: Result_name = df.Result.value_counts().index
      Result_val = df.Result.value_counts().values
      plt.pie(Result_val, labels=Result_name, autopct = "%1.1f%%")

Out[14]: ([<matplotlib.patches.Wedge at 0x1871f4b6e7b0>,
<matplotlib.patches.Wedge at 0x1871f4b6cab0>,
<matplotlib.patches.Wedge at 0x1871f4cfa3b0>,
<matplotlib.patches.Wedge at 0x1871f4cfa3b0>,
<matplotlib.patches.Wedge at 0x1871f4d6d100>),
Text(' - 10.8351904621929765', 1.095118176367859, 'Won'),
Text(' - 6.18281971840073', -0.908788250557988, 'Lost'),
Text(' - 0.8018653618515086', -0.7530019515959862, 'Drawn'),
Text(' - 1.0951181044308164', -0.10355132257919566, 'W/R'),
Text(' - 1.099462718765344', -0.83455204155547046, 'Tied'),
Text(' - 0.0564693430143508', 0.59737138387014, '53%'),
Text(' - 0.33725011591854936', -0.4962482839395266, '25%'),
Text(' - 0.43738110646445755', -0.41072833808726517, '20%'),
Text(' - 0.505731610018000', -0.01666598460456555, '2%')]
```

Observation:

1. Sharjah cricket stadium has been very good for Sachin as he has scored 7 centuries(most)
2. In India, he has scored most centuries(5) at Vidarbha Cricket Association ground.

```
In [13]: df.Against.value_counts()
```

Lost

Observation:

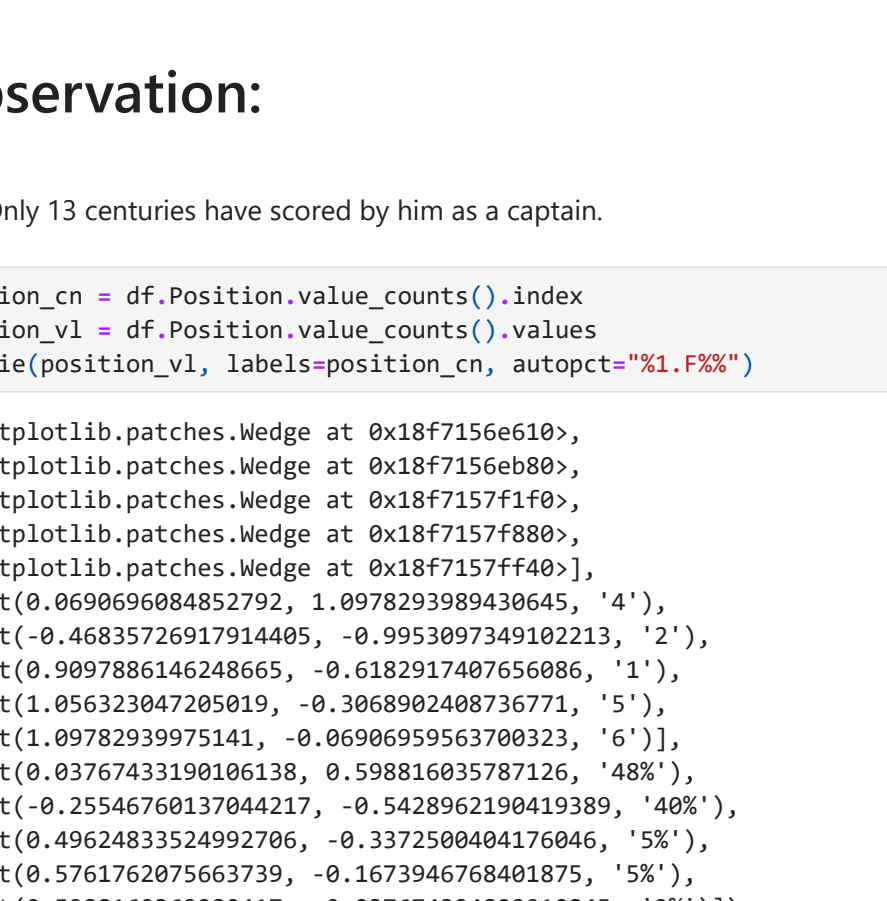
1. Indian team didn't won 25 times, when he scored a century.
2. His 53 centuries contributed in team India's win.

Observation:

1. Sachin has scored most centuries against team Australia i.e. '20'.
2. He was able to score 1 century against team Namibia.
3. He has scored 50% centuries against the SENA countries (South Africa England,New Zealand Australia).

```
In [14]: Result_name = df.Result.value_counts().index
Result_val = df.Result.value_counts().values
plt.pie(Result_val, labels=Result_name, autopct="%1.1f%%")
```

```
Out[14]: ([<matplotlib.patches.Wedge at 0x18f714be670>,
<matplotlib.patches.Wedge at 0x18f714beca0>,
<matplotlib.patches.Wedge at 0x18f714cf3a0>,
<matplotlib.patches.Wedge at 0x18f714cf3a0>],
[Text(-0.18951904621929765, 1.095118170367859, 'Won'),
Text(-0.6182918791840073, -0.9097886520557988, 'Lost'),
Text(-0.8018653618515086, -0.75308019513599862, 'Drawn'),
Text(-0.0951181443201643, -0.10351932177501965, 'N/A'),
Text(-1.099457210765344, -0.034552014356474946, 'Tied')],
[Text(-0.05646493430143508, 0.5973737183837014, '53%'),
Text(-0.33725011591854936, -0.49624828393959266, '25%'),
Text(-0.43738110646445755, -0.41672833808726517, '20%'),
Text(-0.59737371096291804, -0.05646508646455617, '1%'),
Text(-0.599783933144733, -0.018846553285349968, '1%')])
```

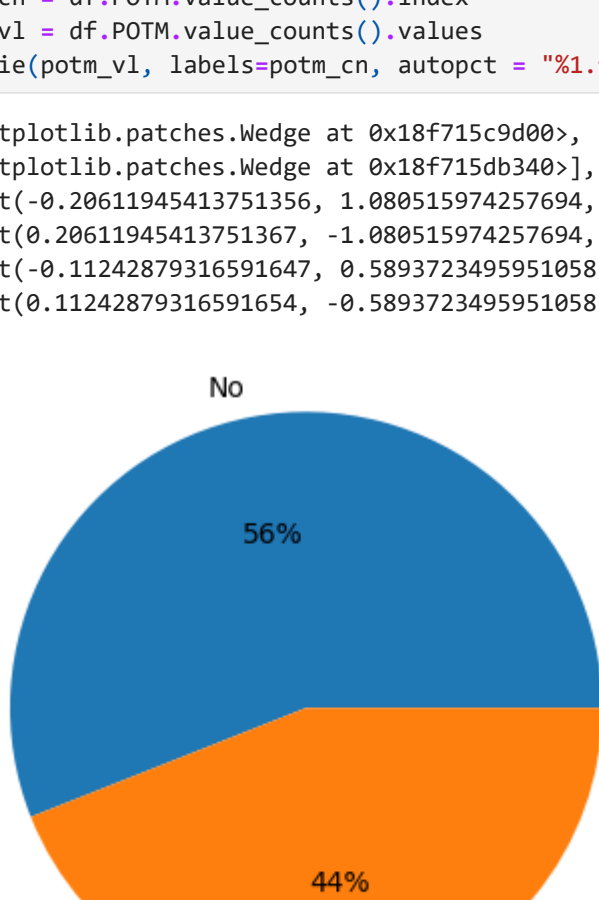


Observation:

1. Indian team didn't won 25 times, when he scored a century.
2. His 53 centuries contributed in team India's win.

```
In [15]: dismissed_cn = df.Dismissed.value_counts().index
dismissed_vl = df.Dismissed.value_counts().values
plt.pie(dismissed_vl, labels=dismissed_cn, autopct="%1.1f%%")
```

```
Out[15]: ([<matplotlib.patches.Wedge at 0x18f71523760>,
<matplotlib.patches.Wedge at 0x18f71523d90>],
[Text(-0.6182917888228987, 0.9097886363311, 'Yes'),
Text(-0.8018653618515086, -0.909788636331101, 'No'),
Text(-0.33725002299438834, -0.49624834709078725, '69%'),
Text(-0.3372500229943883, -0.4962483470907873, '31%')])
```

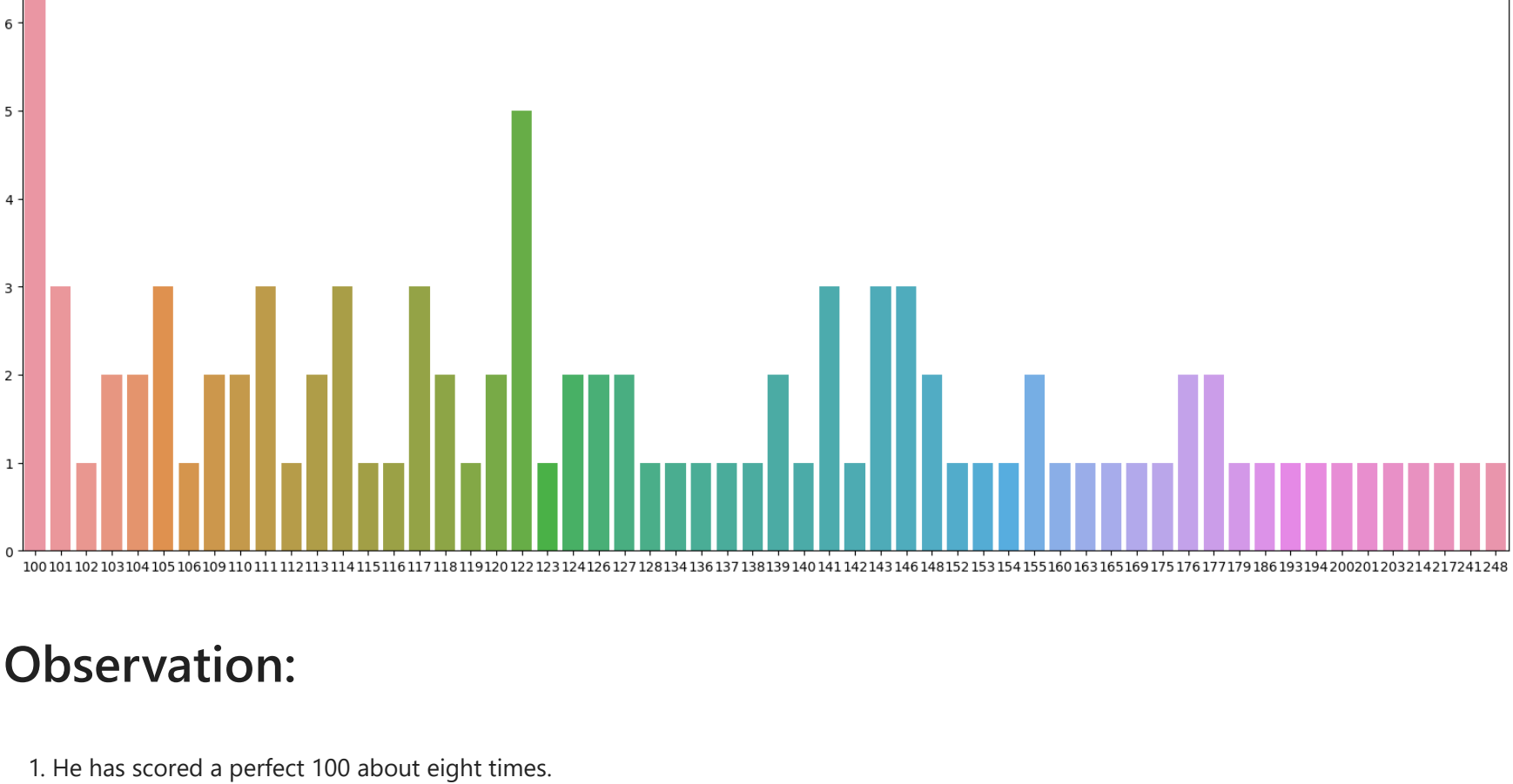


Observation:

1. He has been awarded Player of the Match for 44 times when he scored a century.

```
In [20]: Score_ct = df.Score.value_counts().index
Score_vl = df.Score.value_counts().values
matplotlib.rcParams['figure.figsize'] = (20,10)
sns.barplot(x = Score_ct, y = Score_vl)
```

```
Out[20]: <AxesSubplot: >
```



Observation:

1. He has scored a perfect 100 about eight times.
2. Next most frequent score scored is 122 about 5 times.

Reference:

<https://www.kaggle.com/datasets/gadigevishals/sachin-tendulkar-100-centuries>

```
In [ ]:
```