In [1]:
```
import pandas as pd
```

In [2]:
```
df=pd.read_csv("CricketTestMatchData.csv",encoding="utf-8")
df.head()
```

Out[2]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	BF	SR	100	50	0	4s	6s
0	DG Bradman (AUS)	1928-1948	52	80	10	6996	334	99.94	9800+	58.60	29	13	7	626+	6
1	HC Brook (ENG)	2022-2023	12	20	1	1181	186	62.15	1287	91.76	4	7	1	141	23
2	AC Voges (AUS)	2015-2016	20	31	7	1485	269*	61.87	2667	55.68	5	4	2	186	5
3	RG Pollock (SA)	1963-1970	23	41	4	2256	274	60.97	1707+	54.48	7	11	1	246+	11
4	GA Headley (WI)	1930-1954	22	40	4	2190	270*	60.83	416+	56.00	10	5	2	104+	1

In [3]:
```
df.columns
```

Out[3]:
```
Index(['Player', 'Span', 'Mat', 'Inns', 'NO', 'Runs', 'HS', 'Ave', 'BF',
'SR',
       '100', '50', '0', '4s', '6s'],
      dtype='object')
```

Rename multiple columns in list

In [4]:
```python
df.rename(columns={'Mat':'Matches','NO':'Not_Outs','HS':'Highest_Inns_Score
```

Out[4]:

	Player	Span	Matches	Inns	Not_Outs	Runs	Highest_Inns_Score	Ave	Balls_Faced
0	DG Bradman (AUS)	1928-1948	52	80	10	6996	334	99.94	9800+
1	HC Brook (ENG)	2022-2023	12	20	1	1181	186	62.15	1287
2	AC Voges (AUS)	2015-2016	20	31	7	1485	269*	61.87	2667
3	RG Pollock (SA)	1963-1970	23	41	4	2256	274	60.97	1707+
4	GA Headley (WI)	1930-1954	22	40	4	2190	270*	60.83	416+
...	...	...	...	...	...	...	...	...	...
62	GC Smith (ICC/SA)	2002-2014	117	205	13	9265	277	48.25	15525
63	WH Ponsford (AUS)	1924-1934	29	48	4	2122	266	48.22	3118+
64	SJ McCabe (AUS)	1930-1938	39	62	5	2748	232	48.21	3217+
65	DR Jardine (ENG)	1928-1934	22	33	6	1296	127	48.00	2110+
66	V Kohli (IND)	2011-2023	111	187	11	8676	254*	49.29	15708

67 rows × 15 columns

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

Out[5]:

	Player	Span	Mat	Inns	NO	Runs	HS	Ave	BF	SR	100	50	0	4s	6s
0	DG Bradman (AUS)	1928-1948	52	80	10	6996	334	99.94	9800+	58.60	29	13	7	626+	6
1	HC Brook (ENG)	2022-2023	12	20	1	1181	186	62.15	1287	91.76	4	7	1	141	23
2	AC Voges (AUS)	2015-2016	20	31	7	1485	269*	61.87	2667	55.68	5	4	2	186	5
3	RG Pollock (SA)	1963-1970	23	41	4	2256	274	60.97	1707+	54.48	7	11	1	246+	11
4	GA Headley (WI)	1930-1954	22	40	4	2190	270*	60.83	416+	56.00	10	5	2	104+	1

In [6]:
```
#rename multiple columns in list
df=df.rename(columns={'Mat':'Matches','NO':'Not_Outs','HS':'Highest_Inns_Sc
df.head()
```

Out[6]:

	Player	Span	Matches	Inns	Not_Outs	Runs	Highest_Inns_Score	Ave	Balls_Faced
0	DG Bradman (AUS)	1928-1948	52	80	10	6996	334	99.94	9800+
1	HC Brook (ENG)	2022-2023	12	20	1	1181	186	62.15	1287
2	AC Voges (AUS)	2015-2016	20	31	7	1485	269*	61.87	2667
3	RG Pollock (SA)	1963-1970	23	41	4	2256	274	60.97	1707+
4	GA Headley (WI)	1930-1954	22	40	4	2190	270*	60.83	416+

Check null values

- df.isnull() creates a DataFrame of the same shape as df, where each cell contains a boolean value indicating whether the corresponding cell in df is null (True) or not null (False).
- .any() is called on the resulting DataFrame to check if there are any True values in each column.

In [7]:
```
df.isnull().any()
```

Out[7]:
```
Player                   False
Span                     False
Matches                  False
Inns                     False
Not_Outs                 False
Runs                     False
Highest_Inns_Score       False
Ave                      False
Balls_Faced               True
Batting_Strike_Rate       True
100                      False
50                       False
0                        False
4s                       False
6s                       False
dtype: bool
```

In [8]:
```
pd.set_option('display.max_rows', None)
```

df['Balls_Faced'].isna() == 0 further compares each element in the resulting Boolean Series to 0, which essentially checks if the corresponding element in the 'Balls_Faced' column is not missing. If it's not missing, it will be True (since 0 is equivalent to False in a Boolean context), and if it's missing, it will be False (since 1 is equivalent to True in a Boolean context).

In [9]:
```
df['Balls_Faced'].isna()==1
```

Out[9]:
```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8      True
9     False
10    False
11    False
12    False
13    False
14    False
15    False
16     True
17    False
18    False
19    False
20    False
21    False
22    False
23    False
24    False
25    False
26    False
27    False
28    False
29    False
30    False
31    False
32    False
33    False
34    False
35    False
36    False
37    False
38    False
39    False
40    False
41    False
42    False
43    False
44    False
45    False
46    False
47    False
48    False
49    False
50    False
51    False
52    False
53    False
54    False
55    False
56    False
57    False
58     True
59    False
60    False
```

```
61    False
62    False
63    False
64    False
65    False
66    False
Name: Balls_Faced, dtype: bool
```

In [70]:
```
df[df['Balls_Faced'].isna()==1]
```

Out[70]:

	Player	Matches	Inns	Not_Outs	Runs	Highest_Inns_Score	Ave	Balls_Faced	Battin
8	ED Weekes	48	81	5	4455	207	58.61	NaN	
16	CL Walcott	44	74	7	3798	220	56.68	NaN	
58	Hon.FS Jackson	20	33	4	1415	144	48.79	NaN	

In [73]:
```
df['Balls_Faced']=df['Balls_Faced'].fillna(0)
```

In [74]:
```
df.loc[8]
```

Out[74]:
```
Player                   ED Weekes
Matches                         48
Inns                            81
Not_Outs                         5
Runs                          4455
Highest_Inns_Score             207
Ave                          58.61
Balls_Faced                      0
Batting_Strike_Rate            0.0
100                             15
50                              19
0                                6
4s                            258+
6s                               2
Start_Year                    1948
End_Year                      1958
Country                         WI
Name: 8, dtype: object
```

In [13]:
```
df['Balls_Faced']=df['Balls_Faced'].fillna(0)
df.loc[8]
```

Out[13]:
```
Player                      ED Weekes (WI)
Span                             1948-1958
Matches                                 48
Inns                                    81
Not_Outs                                 5
Runs                                  4455
Highest_Inns_Score                     207
Ave                                  58.61
Balls_Faced                              0
Batting_Strike_Rate                    NaN
100                                     15
50                                      19
0                                        6
4s                                    258+
6s                                       2
Name: 8, dtype: object
```

In [14]:
```
df['Batting_Strike_Rate']=df['Batting_Strike_Rate'].fillna(0)
```

In [15]:
```
df[df['Player']=='ED Weekes (WI)']
```

Out[15]:

	Player	Span	Matches	Inns	Not_Outs	Runs	Highest_Inns_Score	Ave	Balls_Faced
8	ED Weekes (WI)	1948-1958	48	81	5	4455	207	58.61	0

Drop duplicates

In [16]:
```
df.duplicated()
```

Out[16]:
```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
11    False
12     True
13    False
14    False
15    False
16    False
17    False
18    False
19    False
20    False
21    False
22    False
23    False
24    False
25    False
26    False
27    False
28    False
29     True
30    False
31    False
32    False
33    False
34    False
35    False
36    False
37    False
38    False
39    False
40    False
41    False
42    False
43    False
44    False
45    False
46    False
47    False
48    False
49    False
50    False
51    False
52    False
53    False
54    False
55    False
56    False
57    False
58    False
59    False
60    False
```

```
61    False
62    False
63    False
64    False
65    False
66     True
dtype: bool
```

In [17]:
```
df[df['Player'].duplicated()==1]
```

Out[17]:

	Player	Span	Matches	Inns	Not_Outs	Runs	Highest_Inns_Score	Ave	Balls_Faced
12	GS Sobers (WI)	1954-1974	93	160	21	8032	365*	57.78	4063+
29	Javed Miandad (PAK)	1976-1993	124	189	21	8832	280*	52.57	15164+
66	V Kohli (IND)	2011-2023	111	187	11	8676	254*	49.29	15708

In [18]:
```
df[df['Player']=='Javed Miandad (PAK)']
```

Out[18]:

	Player	Span	Matches	Inns	Not_Outs	Runs	Highest_Inns_Score	Ave	Balls_Faced
28	Javed Miandad (PAK)	1976-1993	124	189	21	8832	280*	52.57	15164+
29	Javed Miandad (PAK)	1976-1993	124	189	21	8832	280*	52.57	15164+

In [19]:
```
df[df['Player'].isin(['GS Sobers (WI)','Javed Miandad (PAK)','V Kohli (IND)
```

Out[19]:

	Player	Span	Matches	Inns	Not_Outs	Runs	Highest_Inns_Score	Ave	Balls_Faced
11	GS Sobers (WI)	1954-1974	93	160	21	8032	365*	57.78	4063+
12	GS Sobers (WI)	1954-1974	93	160	21	8032	365*	57.78	4063+
28	Javed Miandad (PAK)	1976-1993	124	189	21	8832	280*	52.57	15164+
29	Javed Miandad (PAK)	1976-1993	124	189	21	8832	280*	52.57	15164+
53	V Kohli (IND)	2011-2023	111	187	11	8676	254*	49.29	15708
66	V Kohli (IND)	2011-2023	111	187	11	8676	254*	49.29	15708

In [20]:
```
df.shape[0]-1
```
Out[20]:
```
66
```

In [21]:
```
df=df.drop_duplicates()
```

In [22]:
```
df[df['Player'].isin(['GS Sobers (WI)','Javed Miandad (PAK)','V Kohli (IND)
```
Out[22]:

	Player	Span	Matches	Inns	Not_Outs	Runs	Highest_Inns_Score	Ave	Balls_Faced
11	GS Sobers (WI)	1954-1974	93	160	21	8032	365*	57.78	4063+
28	Javed Miandad (PAK)	1976-1993	124	189	21	8832	280*	52.57	15164+
53	V Kohli (IND)	2011-2023	111	187	11	8676	254*	49.29	15708

Split up span into start and end date

In [23]:
```
"ayesha_khan".split('_')
```
Out[23]:
```
['ayesha', 'khan']
```

In [24]:
```
df['Span'].str.split(pat='-')
```

Out[24]:
```
0     [1928, 1948]
1     [2022, 2023]
2     [2015, 2016]
3     [1963, 1970]
4     [1930, 1954]
5     [1924, 1935]
6     [1931, 1939]
7     [1955, 1968]
8     [1948, 1958]
9     [2010, 2023]
10    [1927, 1947]
11    [1954, 1974]
13    [2000, 2015]
14    [2019, 2023]
15    [1908, 1930]
16    [1948, 1960]
17    [1937, 1955]
18    [1995, 2013]
19    [1921, 1929]
20    [2010, 2023]
21    [1968, 1973]
22    [1993, 1995]
23    [1970, 1984]
24    [1935, 1951]
25    [1989, 2013]
26    [2018, 2023]
27    [1990, 2006]
28    [1976, 1993]
30    [1996, 2012]
31    [1998, 2010]
32    [2000, 2017]
33    [1995, 2012]
34    [1920, 1929]
35    [1992, 2002]
36    [2005, 2013]
37    [1994, 2015]
38    [1971, 1987]
39    [1985, 2004]
40    [2021, 2023]
41    [1994, 2009]
42    [2004, 2018]
43    [1978, 1994]
44    [2012, 2023]
45    [1974, 1991]
46    [2021, 2023]
47    [1937, 1957]
48    [1997, 2014]
49    [1992, 2007]
50    [1948, 1963]
51    [1911, 1928]
52    [2001, 2013]
53    [2011, 2023]
54    [2019, 2021]
55    [2004, 2015]
56    [1961, 1966]
57    [1929, 1949]
58    [1893, 1905]
59    [2001, 2013]
60    [1948, 1963]
61    [1965, 1981]
62    [2002, 2014]
```

```
63    [1924, 1934]
64    [1930, 1938]
65    [1928, 1934]
Name: Span, dtype: object
```

In [27]:
```
pd.reset_option('display.max_rows')
```

In [28]:
```
df['Start_Year']=df['Span'].str.split(pat='-').str[0]
df['Start_Year']
```

Out[28]:
```
0     1928
1     2022
2     2015
3     1963
4     1930
      ...
61    1965
62    2002
63    1924
64    1930
65    1928
Name: Start_Year, Length: 64, dtype: object
```

In [31]:
```
df['End_Year']=df['Span'].str.split(pat='-').str[1]
df['End_Year']
```

Out[31]:
```
0     1948
1     2023
2     2016
3     1970
4     1954
      ...
61    1981
62    2014
63    1934
64    1938
65    1934
Name: End_Year, Length: 64, dtype: object
```

In [32]:
```
df.head(3)
```

Out[32]:

	Player	Span	Matches	Inns	Not_Outs	Runs	Highest_Inns_Score	Ave	Balls_Faced
0	DG Bradman (AUS)	1928-1948	52	80	10	6996	334	99.94	9800+
1	HC Brook (ENG)	2022-2023	12	20	1	1181	186	62.15	1287
2	AC Voges (AUS)	2015-2016	20	31	7	1485	269*	61.87	2667

In [33]:
```
df.drop(['Span'])
```

```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call las
t)
Input In [33], in <cell line: 1>()
----> 1 df.drop(['Span'])

File ~\anaconda3\lib\site-packages\pandas\util\_decorators.py:311, in depr
ecate_nonkeyword_arguments.<locals>.decorate.<locals>.wrapper(*args, **kwa
rgs)
    305 if len(args) > num_allow_args:
    306     warnings.warn(
    307         msg.format(arguments=arguments),
    308         FutureWarning,
    309         stacklevel=stacklevel,
    310     )
--> 311 return func(*args, **kwargs)

File ~\anaconda3\lib\site-packages\pandas\core\frame.py:4954, in DataFram
e.drop(self, labels, axis, index, columns, level, inplace, errors)
   4806 @deprecate_nonkeyword_arguments(version=None, allowed_args=["sel
f", "labels"])
   4807 def drop(
   4808     self,
   (...)
   4815     errors: str = "raise",
   4816 ):
   4817     """
   4818     Drop specified labels from rows or columns.
   4819
   (...)
   4952             weight  1.0     0.8
   4953     """
-> 4954     return super().drop(
   4955         labels=labels,
   4956         axis=axis,
   4957         index=index,
   4958         columns=columns,
   4959         level=level,
   4960         inplace=inplace,
   4961         errors=errors,
   4962     )

File ~\anaconda3\lib\site-packages\pandas\core\generic.py:4267, in NDFram
e.drop(self, labels, axis, index, columns, level, inplace, errors)
   4265 for axis, labels in axes.items():
   4266     if labels is not None:
-> 4267         obj = obj._drop_axis(labels, axis, level=level, errors=err
ors)
   4269 if inplace:
   4270     self._update_inplace(obj)

File ~\anaconda3\lib\site-packages\pandas\core\generic.py:4311, in NDFram
e._drop_axis(self, labels, axis, level, errors, consolidate, only_slice)
   4309         new_axis = axis.drop(labels, level=level, errors=errors)
   4310     else:
-> 4311         new_axis = axis.drop(labels, errors=errors)
   4312     indexer = axis.get_indexer(new_axis)
   4314 # Case for non-unique axis
   4315 else:
```

```
File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:6644, in In
dex.drop(self, labels, errors)
   6642 if mask.any():
   6643     if errors != "ignore":
-> 6644         raise KeyError(f"{list(labels[mask])} not found in axis")
   6645     indexer = indexer[~mask]
   6646 return self.delete(indexer)

KeyError: "['Span'] not found in axis"
```

In [34]:
```
df.drop(['Span'],axis=1)
```

Out[34]:

	Player	Matches	Inns	Not_Outs	Runs	Highest_Inns_Score	Ave	Balls_Faced	Battii
0	DG Bradman (AUS)	52	80	10	6996	334	99.94	9800+	
1	HC Brook (ENG)	12	20	1	1181	186	62.15	1287	
2	AC Voges (AUS)	20	31	7	1485	269*	61.87	2667	
3	RG Pollock (SA)	23	41	4	2256	274	60.97	1707+	
4	GA Headley (WI)	22	40	4	2190	270*	60.83	416+	
...	...	...	...	...	...	...	...	...	
61	KD Walters (AUS)	74	125	14	5357	250	48.26	8662+	
62	GC Smith (ICC/SA)	117	205	13	9265	277	48.25	15525	
63	WH Ponsford (AUS)	29	48	4	2122	266	48.22	3118+	
64	SJ McCabe (AUS)	39	62	5	2748	232	48.21	3217+	
65	DR Jardine (ENG)	22	33	6	1296	127	48.00	2110+	

64 rows × 16 columns

In [35]:
```
df.head()
```

Out[35]:

	Player	Span	Matches	Inns	Not_Outs	Runs	Highest_Inns_Score	Ave	Balls_Faced
0	DG Bradman (AUS)	1928-1948	52	80	10	6996	334	99.94	9800+
1	HC Brook (ENG)	2022-2023	12	20	1	1181	186	62.15	1287
2	AC Voges (AUS)	2015-2016	20	31	7	1485	269*	61.87	2667
3	RG Pollock (SA)	1963-1970	23	41	4	2256	274	60.97	1707+
4	GA Headley (WI)	1930-1954	22	40	4	2190	270*	60.83	416+

In [36]:
```
df.drop(['Span'],axis=1,inplace=True)
```

In [37]:
```
df.head(3)
```

Out[37]:

	Player	Matches	Inns	Not_Outs	Runs	Highest_Inns_Score	Ave	Balls_Faced	Batting
0	DG Bradman (AUS)	52	80	10	6996	334	99.94	9800+	
1	HC Brook (ENG)	12	20	1	1181	186	62.15	1287	
2	AC Voges (AUS)	20	31	7	1485	269*	61.87	2667	

Split country from the player column

In [38]:
```
df['Player'].str.split(pat='(')
```

Out[38]:
```
0      [DG Bradman , AUS)]
1        [HC Brook , ENG)]
2        [AC Voges , AUS)]
3       [RG Pollock , SA)]
4       [GA Headley , WI)]
              ...
61     [KD Walters , AUS)]
62    [GC Smith , ICC/SA)]
63    [WH Ponsford , AUS)]
64       [SJ McCabe , AUS)]
65     [DR Jardine , ENG)]
Name: Player, Length: 64, dtype: object
```

In [39]:
```
df['Country']=df['Player'].str.split(pat='(').str[1]
```

In [40]:
```
df['Country']
```

Out[40]:
```
0         AUS)
1         ENG)
2         AUS)
3          SA)
4          WI)
        ...
61        AUS)
62     ICC/SA)
63        AUS)
64        AUS)
65        ENG)
Name: Country, Length: 64, dtype: object
```

In [43]:
```
df['Country']=df['Country'].str.split(pat=')').str[0]
```

In [44]:
```
df['Country']
```

Out[44]:
```
0         AUS
1         ENG
2         AUS
3          SA
4          WI
        ...
61        AUS
62     ICC/SA
63        AUS
64        AUS
65        ENG
Name: Country, Length: 64, dtype: object
```

In [45]:
```
 df['Player']
```

Out[45]:
```
0       DG Bradman (AUS)
1         HC Brook (ENG)
2         AC Voges (AUS)
3        RG Pollock (SA)
4        GA Headley (WI)
             ...
61      KD Walters (AUS)
62     GC Smith (ICC/SA)
63     WH Ponsford (AUS)
64       SJ McCabe (AUS)
65      DR Jardine (ENG)
Name: Player, Length: 64, dtype: object
```

In [46]:
```
df['Player']=df['Player'].str.split(pat='(').str[0]
```

In [47]:
```
df['Player']
```

Out[47]:
```
0      DG Bradman
1        HC Brook
2        AC Voges
3      RG Pollock
4      GA Headley
          ...
61     KD Walters
62       GC Smith
63    WH Ponsford
64      SJ McCabe
65     DR Jardine
Name: Player, Length: 64, dtype: object
```

In [48]:
```
df.head(3)
```

Out[48]:

	Player	Matches	Inns	Not_Outs	Runs	Highest_Inns_Score	Ave	Balls_Faced	Battin
0	DG Bradman	52	80	10	6996	334	99.94	9800+	
1	HC Brook	12	20	1	1181	186	62.15	1287	
2	AC Voges	20	31	7	1485	269*	61.87	2667	

change Datatypes

In [49]:
```
df.dtypes
```

Out[49]:
```
Player                  object
Matches                  int64
Inns                     int64
Not_Outs                 int64
Runs                     int64
Highest_Inns_Score      object
Ave                    float64
Balls_Faced             object
Batting_Strike_Rate    float64
100                      int64
50                       int64
0                        int64
4s                      object
6s                      object
Start_Year              object
End_Year                object
Country                 object
dtype: object
```

In [51]:
```
text = "***Important Text***"
stripped_text = text.strip('*')
print(stripped_text)
```

```
Important Text
```

In [52]:
```
df['Highest_Inns_Score']
```

Out[52]:
```
0      334
1      186
2     269*
3      274
4     270*
      ...
61     250
62     277
63     266
64     232
65     127
Name: Highest_Inns_Score, Length: 64, dtype: object
```

In [54]:
```
df['Highest_Inns_Score']=df['Highest_Inns_Score'].str.strip('*')
```

In [55]:
```
df['Highest_Inns_Score']
```

Out[55]:
```
0     334
1     186
2     269
3     274
4     270
     ...
61    250
62    277
63    266
64    232
65    127
Name: Highest_Inns_Score, Length: 64, dtype: object
```

In [60]:
```
df['Highest_Inns_Score']=df['Highest_Inns_Score'].astype('int')
```

In [61]:
```
df.dtypes
```

Out[61]:
```
Player                    object
Matches                    int64
Inns                       int64
Not_Outs                   int64
Runs                       int64
Highest_Inns_Score         int32
Ave                      float64
Balls_Faced               object
Batting_Strike_Rate      float64
100                        int64
50                         int64
0                          int64
4s                        object
6s                        object
Start_Year                object
End_Year                  object
Country                   object
dtype: object
```

In [63]:
```
df=df.astype({'Start_Year':'int','End_Year':'int'})
```

In [65]:
```
df.dtypes
```

Out[65]:
```
Player                   object
Matches                   int64
Inns                      int64
Not_Outs                  int64
Runs                      int64
Highest_Inns_Score        int32
Ave                     float64
Balls_Faced              object
Batting_Strike_Rate     float64
100                       int64
50                        int64
0                         int64
4s                       object
6s                       object
Start_Year                int32
End_Year                  int32
Country                  object
dtype: object
```

In [66]:
```
df['Balls_Faced']
```

Out[66]:
```
0      9800+
1       1287
2       2667
3      1707+
4       416+
       ...
61     8662+
62     15525
63     3118+
64     3217+
65     2110+
Name: Balls_Faced, Length: 64, dtype: object
```

In [67]:
```
df['Balls_Faced']=df['Balls_Faced'].str.strip('+')
```

In [76]:
```
df['Balls_Faced']=df['Balls_Faced'].astype('int')
```

In [78]:
```
df.dtypes
```

Out[78]:
```
Player                  object
Matches                  int64
Inns                     int64
Not_Outs                 int64
Runs                     int64
Highest_Inns_Score       int32
Ave                    float64
Balls_Faced              int32
Batting_Strike_Rate    float64
100                      int64
50                       int64
0                        int64
4s                      object
6s                      object
Start_Year               int32
End_Year                 int32
Country                 object
dtype: object
```

append career_length column

In [80]:
```
df['Career_Length']=df['End_Year']-df['Start_Year']
```

In [91]:
```
df.head(3)
```

Out[91]:

	Player	Matches	Inns	Not_Outs	Runs	Highest_Inns_Score	Ave	Balls_Faced	Battin
0	DG Bradman	52	80	10	6996	334	99.94	9800	
1	HC Brook	12	20	1	1181	186	62.15	1287	
2	AC Voges	20	31	7	1485	269	61.87	2667	

Analysis

In [82]:
```
#1. What is the avg career length
df['Career_Length'].mean()
```

Out[82]:
```
12.75
```

In [83]:
```
#2. what is the avg batting strike rate for cricketers who played over 10 y
df[df['Career_Length']>10]['Batting_Strike_Rate'].mean()
```

Out[83]:
```
47.95454545454545
```

In [86]:
```
#3. find no. of cricketers who played before 1960
df[df['Start_Year']<1960]['Player'].count()
```

Out[86]:
```
23
```

In [92]:
```
# Find Max highest inns score by Country
df.groupby('Country')['Highest_Inns_Score'].max().to_frame('Highinncountry'
reset_index().sort_values('Highinncountry',ascending=False)
```

Out[92]:

	Country	Highinncountry
5	ICC/WI	400
0	AUS	380
10	SL	374
11	WI	365
1	ENG	364
3	ICC/PAK	329
2	ICC/IND	319
8	PAK	313
9	SA	278
4	ICC/SA	277
6	IND	254
7	NZ	251
12	ZIM	232

In [90]:
```
# Find Max highest inns scoore by Country
round(df.groupby('Country')[['100','50','0']].mean(),2)
```

Out[90]:

	100	50	0
Country			
AUS	20.62	28.00	8.50
ENG	12.31	20.77	4.31
ICC/IND	29.50	47.50	12.00
ICC/PAK	25.00	46.00	15.00
ICC/SA	36.00	48.00	13.50
ICC/WI	34.00	48.00	17.00
IND	29.50	36.25	10.75
NZ	12.33	16.33	4.33
PAK	17.80	23.20	8.00
SA	9.80	20.20	3.40
SL	28.67	44.00	12.33
WI	16.62	25.62	7.25
ZIM	12.00	27.00	5.00