Final project

July 21, 2020

1 Title: Term Project Data Preparation

1.0.1 Author: Sanjay Jaras

1.1 Import Libraries

[2]:
```
from datetime import datetime

import Levenshtein as lv
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import yaml

%matplotlib inline
from IPython.display import set_matplotlib_formats

from test_match_helpers import websiteutils as wu
from test_match_helpers import yamlutils as yu
from test_match_helpers import ziputils as zp
from test_match_helpers import web_profile_id_utis as wuapi
from test_match_helpers import pid_api_utils as apiu
from test_match_helpers import profile_api_utils as papiu


"""
The above modules in test_match_helpers are written to load yaml file into␣
 ↪dataframe
"""

# test_match_helpers: module for classes that used in converting per match yaml␣
 ↪files to consolidated data frame
```

[2]:
```
'\nThe above modules in test_match_helpers are written to load yaml file into
dataframe\n'
```

1.2 Configurations

[3]:
```
set_matplotlib_formats("png", "pdf")
plt.style.use(    "seaborn-darkgrid"
)  # fivethirtyeight,ggplot,seaborn-darkgrid,seaborn-whitegrid
plt.rcParams["figure.figsize"] = [24, 12]
```

1.3 Test Reading Yaml file

[4]:
```
yamlIn = open("291352.yaml", "r")
yamlFile = yaml.load(yamlIn, Loader=yaml.FullLoader)
tempDf = yu.readYamlToDataFrame(1, yamlFile)
tempDf.head()
# tempGroupByInnings = tempDf.groupby(by=["MatchId", "InningNo"])
# tempTeamsTotalRuns = tempGroupByInnings["TotalRuns"].sum()
```

[4]:
```
   MatchId        Date  City                Venue      Team1  Team2  \
0        1  2008-01-02   NaN  Sydney Cricket Ground  Australia  India
1        1  2008-01-02   NaN  Sydney Cricket Ground  Australia  India
2        1  2008-01-02   NaN  Sydney Cricket Ground  Australia  India
3        1  2008-01-02   NaN  Sydney Cricket Ground  Australia  India
4        1  2008-01-02   NaN  Sydney Cricket Ground  Australia  India

  TossWinner TossDecision ManOfTheMatch     Winner  …  BattingTeam  \
0  Australia          bat     A Symonds  Australia  …    Australia
1  Australia          bat     A Symonds  Australia  …    Australia
2  Australia          bat     A Symonds  Australia  …    Australia
3  Australia          bat     A Symonds  Australia  …    Australia
4  Australia          bat     A Symonds  Australia  …    Australia

     Opener1    Opener2 BallNo    Batsman    Bowler NonStriker RunsBat  \
0  PA Jaques  ML Hayden    0.1  PA Jaques  RP Singh  ML Hayden       0
1  PA Jaques  ML Hayden    0.2  PA Jaques  RP Singh  ML Hayden       0
2  PA Jaques  ML Hayden    0.3  PA Jaques  RP Singh  ML Hayden       0
3  PA Jaques  ML Hayden    0.4  PA Jaques  RP Singh  ML Hayden       0
4  PA Jaques  ML Hayden    0.5  PA Jaques  RP Singh  ML Hayden       0

  RunsExtras TotalRuns
0          0         0
1          0         0
2          0         0
3          0         0
4          0         0

[5 rows x 23 columns]
```

1.4 Extract Zip file to yaml file and convert yaml data into data frame

1.4.1 Please find ziputils, yamlutils and testmatches classes modules from test_match_helpers folder

[5]:

```
df = zp.extractZipAndProcess("tests.zip", 50)
df.head()
```

```
Done processing in 56.74766776000001 seconds
```

[5]:

```
   MatchId        Date City                 Venue      Team1  Team2  \
0        1  2008-01-02  NaN  Sydney Cricket Ground  Australia  India
1        1  2008-01-02  NaN  Sydney Cricket Ground  Australia  India
2        1  2008-01-02  NaN  Sydney Cricket Ground  Australia  India
3        1  2008-01-02  NaN  Sydney Cricket Ground  Australia  India
4        1  2008-01-02  NaN  Sydney Cricket Ground  Australia  India

  TossWinner TossDecision ManOfTheMatch     Winner  … BattingTeam  \
0  Australia          bat     A Symonds  Australia  …   Australia
1  Australia          bat     A Symonds  Australia  …   Australia
2  Australia          bat     A Symonds  Australia  …   Australia
3  Australia          bat     A Symonds  Australia  …   Australia
4  Australia          bat     A Symonds  Australia  …   Australia

    Opener1    Opener2 BallNo    Batsman    Bowler  NonStriker RunsBat  \
0  PA Jaques  ML Hayden    0.1  PA Jaques  RP Singh   ML Hayden       0
1  PA Jaques  ML Hayden    0.2  PA Jaques  RP Singh   ML Hayden       0
2  PA Jaques  ML Hayden    0.3  PA Jaques  RP Singh   ML Hayden       0
3  PA Jaques  ML Hayden    0.4  PA Jaques  RP Singh   ML Hayden       0
4  PA Jaques  ML Hayden    0.5  PA Jaques  RP Singh   ML Hayden       0

  RunsExtras TotalRuns
0          0         0
1          0         0
2          0         0
3          0         0
4          0         0

[5 rows x 23 columns]
```

[6]:

```
df.shape
```

[6]:

```
(100503, 23)
```

1.5 Convert Data frame to csv for future reference

Read 5 lines from CSV

[7]:

```python
df.to_csv("all-records.csv", index=False)
count = 0
with open("all-records.csv") as f:
    line = f.readline()
    while line != "":
        count += 1
        print(line, end="")
        if count > 5:
            break
        print(f.readline())
```

```
MatchId,Date,City,Venue,Team1,Team2,TossWinner,TossDecision,ManOfTheMatch,Winner
,WonByRuns,WonByWickets,InningNo,BattingTeam,Opener1,Opener2,BallNo,Batsman,Bowl
er,NonStriker,RunsBat,RunsExtras,TotalRuns
1,2008-01-02,,Sydney Cricket Ground,Australia,India,Australia,bat,A
Symonds,Australia,122,0,1,Australia,PA Jaques,ML Hayden,0.1,PA Jaques,RP
Singh,ML Hayden,0,0,0

MatchId,Date,City,Venue,Team1,Team2,TossWinner,TossDecision,ManOfTheMatch,Winner
,WonByRuns,WonByWickets,InningNo,BattingTeam,Opener1,Opener2,BallNo,Batsman,Bowl
er,NonStriker,RunsBat,RunsExtras,TotalRuns
1,2008-01-02,,Sydney Cricket Ground,Australia,India,Australia,bat,A
Symonds,Australia,122,0,1,Australia,PA Jaques,ML Hayden,0.2,PA Jaques,RP
Singh,ML Hayden,0,0,0

MatchId,Date,City,Venue,Team1,Team2,TossWinner,TossDecision,ManOfTheMatch,Winner
,WonByRuns,WonByWickets,InningNo,BattingTeam,Opener1,Opener2,BallNo,Batsman,Bowl
er,NonStriker,RunsBat,RunsExtras,TotalRuns
1,2008-01-02,,Sydney Cricket Ground,Australia,India,Australia,bat,A
Symonds,Australia,122,0,1,Australia,PA Jaques,ML Hayden,0.3,PA Jaques,RP
Singh,ML Hayden,0,0,0

MatchId,Date,City,Venue,Team1,Team2,TossWinner,TossDecision,ManOfTheMatch,Winner
,WonByRuns,WonByWickets,InningNo,BattingTeam,Opener1,Opener2,BallNo,Batsman,Bowl
er,NonStriker,RunsBat,RunsExtras,TotalRuns
1,2008-01-02,,Sydney Cricket Ground,Australia,India,Australia,bat,A
Symonds,Australia,122,0,1,Australia,PA Jaques,ML Hayden,0.4,PA Jaques,RP
Singh,ML Hayden,0,0,0

MatchId,Date,City,Venue,Team1,Team2,TossWinner,TossDecision,ManOfTheMatch,Winner
,WonByRuns,WonByWickets,InningNo,BattingTeam,Opener1,Opener2,BallNo,Batsman,Bowl
er,NonStriker,RunsBat,RunsExtras,TotalRuns
1,2008-01-02,,Sydney Cricket Ground,Australia,India,Australia,bat,A
Symonds,Australia,122,0,1,Australia,PA Jaques,ML Hayden,0.5,PA Jaques,RP
Singh,ML Hayden,0,0,0

MatchId,Date,City,Venue,Team1,Team2,TossWinner,TossDecision,ManOfTheMatch,Winner
```

```
,WonByRuns,WonByWickets,InningNo,BattingTeam,Opener1,Opener2,BallNo,Batsman,Bowl
er,NonStriker,RunsBat,RunsExtras,TotalRuns
```

1.6 Check for Values in City column

```
[8]: df.City.unique()
```

```
[8]: array([nan, 'Cape Town', 'Dunedin', 'Durban', 'Wellington', 'Perth',
            'Mirpur', 'Hamilton', 'Guyana', 'Napier', 'Chennai', 'Ahmedabad',
            'Trinidad', 'Kanpur', 'London', 'Jamaica', 'Manchester', 'Antigua',
            'Nottingham', 'Barbados', 'Leeds', 'Colombo', 'Birmingham',
            'Bangalore', 'Chandigarh', 'Delhi', 'Nagpur', 'Bloemfontein',
            'Brisbane', 'Centurion', 'Karachi'], dtype=object)
```

```
[9]: any(df.City.isna())
```

```
[9]: True
```

1.6.1 City column contains NaN values so let's drop this column

```
[10]: df.drop(labels=["City"], axis=1, inplace=True)
      df.head()
```

```
[10]:    MatchId        Date                 Venue      Team1  Team2 TossWinner  \
      0        1  2008-01-02  Sydney Cricket Ground  Australia  India  Australia
      1        1  2008-01-02  Sydney Cricket Ground  Australia  India  Australia
      2        1  2008-01-02  Sydney Cricket Ground  Australia  India  Australia
      3        1  2008-01-02  Sydney Cricket Ground  Australia  India  Australia
      4        1  2008-01-02  Sydney Cricket Ground  Australia  India  Australia

        TossDecision ManOfTheMatch     Winner  WonByRuns  …  BattingTeam  \
      0          bat     A Symonds  Australia        122  …    Australia
      1          bat     A Symonds  Australia        122  …    Australia
      2          bat     A Symonds  Australia        122  …    Australia
      3          bat     A Symonds  Australia        122  …    Australia
      4          bat     A Symonds  Australia        122  …    Australia

           Opener1    Opener2 BallNo    Batsman    Bowler NonStriker RunsBat  \
      0  PA Jaques  ML Hayden    0.1  PA Jaques  RP Singh  ML Hayden       0
      1  PA Jaques  ML Hayden    0.2  PA Jaques  RP Singh  ML Hayden       0
      2  PA Jaques  ML Hayden    0.3  PA Jaques  RP Singh  ML Hayden       0
      3  PA Jaques  ML Hayden    0.4  PA Jaques  RP Singh  ML Hayden       0
      4  PA Jaques  ML Hayden    0.5  PA Jaques  RP Singh  ML Hayden       0

        RunsExtras  TotalRuns
      0          0          0
      1          0          0
```

```
2          0          0
3          0          0
4          0          0

[5 rows x 22 columns]
```

1.6.2 Add columns for boundries

[11]:

```python
df["Fours"] = df["RunsBat"] == 4
df["Sixes"] = df["RunsBat"] == 6
```

1.6.3 Check for duplicates

[12]:

```python
duplicates = df[df.duplicated(["MatchId", "InningNo", "BallNo"], keep=False)]
duplicates.head()
```

[12]:

```
      MatchId        Date                                         Venue  \
9686        6  2008-01-16  Western Australia Cricket Association Ground
9695        6  2008-01-16  Western Australia Cricket Association Ground
9715        6  2008-01-16  Western Australia Cricket Association Ground
9724        6  2008-01-16  Western Australia Cricket Association Ground

          Team1  Team2 TossWinner TossDecision ManOfTheMatch Winner  \
9686  Australia  India      India          bat     IK Pathan  India
9695  Australia  India      India          bat     IK Pathan  India
9715  Australia  India      India          bat     IK Pathan  India
9724  Australia  India      India          bat     IK Pathan  India

      WonByRuns  …   Opener2  BallNo    Batsman   Bowler NonStriker  \
9686         72  …  V Sehwag     6.1   W Jaffer    B Lee   V Sehwag
9695         72  …  V Sehwag     6.1   W Jaffer    B Lee   V Sehwag
9715         72  …  V Sehwag    10.1   V Sehwag  SW Tait  IK Pathan
9724         72  …  V Sehwag    10.1  IK Pathan  SW Tait   V Sehwag

      RunsBat RunsExtras TotalRuns  Fours  Sixes
9686        1          1         2  False  False
9695        0          0         0  False  False
9715        1          0         1  False  False
9724        0          0         0  False  False

[4 rows x 24 columns]
```

[13]:

```python
print(f"We found {len(duplicates)/2} records duplicated.")
```

```
We found 2.0 records duplicated.
```

1.7 Drop Duplcates

```
[14]: df.drop_duplicates(["MatchId", "InningNo", "BallNo"], keep="first",␣
       ↪inplace=True)
      duplicates = df[df.duplicated(["MatchId", "InningNo", "BallNo"], keep=False)]
      duplicates.head()
```

```
[14]: Empty DataFrame
      Columns: [MatchId, Date, Venue, Team1, Team2, TossWinner, TossDecision,
      ManOfTheMatch, Winner, WonByRuns, WonByWickets, InningNo, BattingTeam, Opener1,
      Opener2, BallNo, Batsman, Bowler, NonStriker, RunsBat, RunsExtras, TotalRuns,
      Fours, Sixes]
      Index: []

      [0 rows x 24 columns]
```

Now there are no duplicates

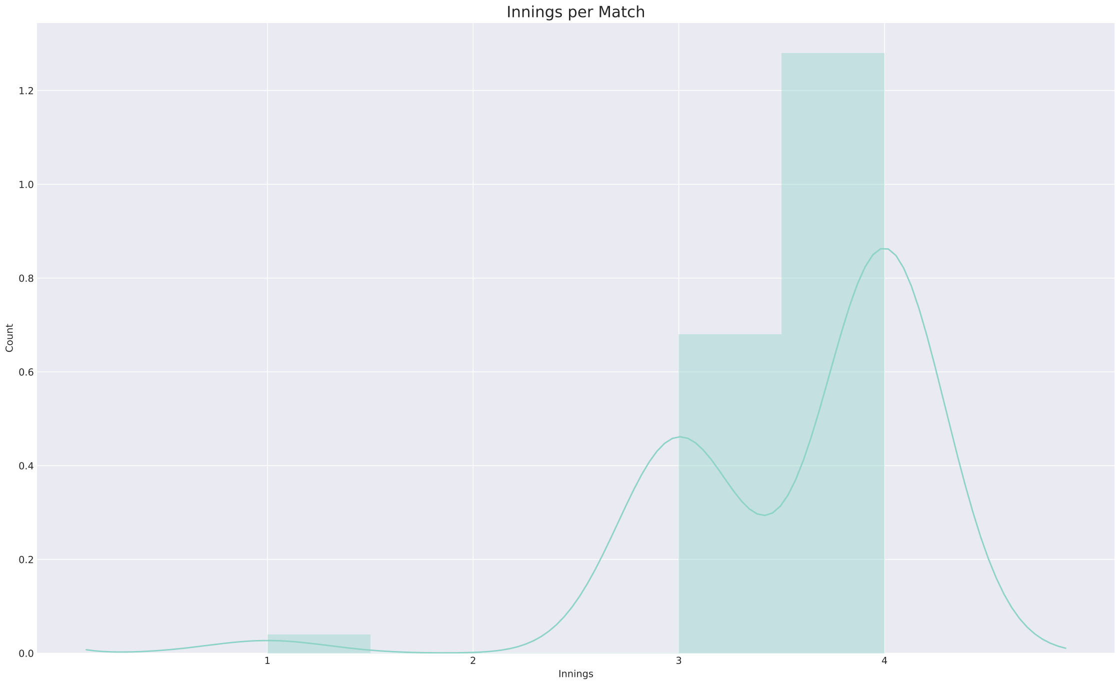
```
[15]: matchinfo = df.groupby(by=["MatchId"])
      dfMatchInfo = matchinfo.head(n=1)
      dfMatchInfo.shape
```

```
[15]: (50, 24)
```

```
[16]: inningsPerMatch = matchinfo.agg({"InningNo": "max"})

      plt.figure(figsize=(20, 12))
      g = sns.distplot(inningsPerMatch.InningNo)
      g.set(xlabel="Innings", ylabel="Count")
      g.axes.set_title("Innings per Match", fontsize=16)
      g.set_xticks(range(1, 5, 1))
```

```
[16]: [<matplotlib.axis.XTick at 0x7efd8be128d0>,
       <matplotlib.axis.XTick at 0x7efd8bb5b3d0>,
       <matplotlib.axis.XTick at 0x7efd8bb5e790>,
       <matplotlib.axis.XTick at 0x7efd8ba6fb10>]
```



1.7.1 Matches with inning 1 & 2 are outliers as we say there are not completed matches, so we can remove them

1.8 Drop outliers

[17]:

```
df.dropna(inplace=True)

inningsPerMatch = inningsPerMatch[inningsPerMatch.InningNo < 3]

df.drop(df[df.MatchId.isin(inningsPerMatch)], errors="ignore", inplace=True)
```

[18]:

```
df.shape
```

[18]:

```
(100491, 24)
```

[19]:

```
groupByInnings = df.groupby(by=["MatchId", "InningNo"])
teamsTotalRuns = groupByInnings["TotalRuns"].sum()
boundriesPerInnings = groupByInnings["Fours"].sum() + groupByInnings["Sixes"].
 ↪sum()
```

[20]:

```
df.MatchId.unique()
test = df[df.MatchId == 47]
```

1.9 Conduct Fuzzy Matching

1.9.1 Do fuzzy matching to check problems Team names

[21]:

```
distance_to_check = 2

teams = np.concatenate((df.Team1.unique(), df.Team2.unique()))
teams = pd.Series(teams)
teams = teams.unique()
df_dist = pd.DataFrame()
for team in teams:
    dist = []
    for target in teams:
        d = lv.distance(team, target)
        dist.append(d)
        if d > 0 and d <= distance_to_check:
            print(f"Close Match found for Team Names:{team} and {target}")
    df_dist[team] = dist
```

[22]:

```
df_dist
```

[22]:

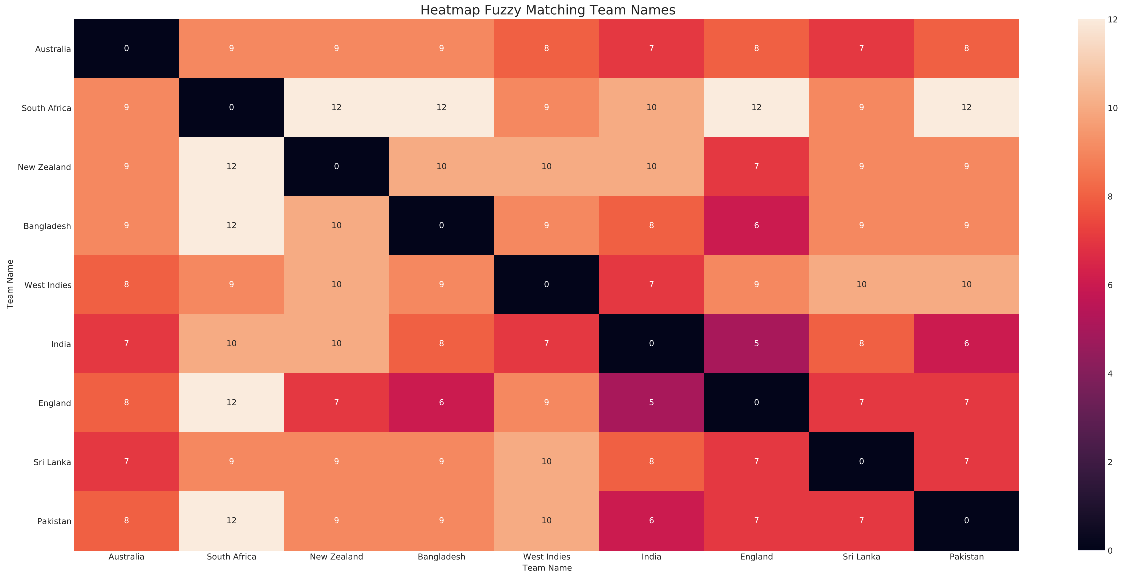
```
   Australia  South Africa  New Zealand  Bangladesh  West Indies  India  \
0          0             9            9           9            8      7
1          9             0           12          12            9     10
2          9            12            0          10           10     10
3          9            12           10           0            9      8
4          8             9           10           9            0      7
5          7            10           10           8            7      0
6          8            12            7           6            9      5
7          7             9            9           9           10      8
8          8            12            9           9           10      6

   England  Sri Lanka  Pakistan
0        8          7         8
1       12          9        12
2        7          9         9
3        6          9         9
4        9         10        10
5        5          8         6
6        0          7         7
7        7          0         7
8        7          7         0
```

[23]:

```
plt.figure(figsize=(26, 12))
g = sns.heatmap(df_dist, annot=True)
g.set(xlabel="Team Name", ylabel="Team Name")
g.axes.set_title("Heatmap Fuzzy Matching Team Names", fontsize=16)
g.set_yticklabels(teams, rotation=0)
```

```
[23]: [Text(0, 0.5, 'Australia'),
       Text(0, 1.5, 'South Africa'),
       Text(0, 2.5, 'New Zealand'),
       Text(0, 3.5, 'Bangladesh'),
       Text(0, 4.5, 'West Indies'),
       Text(0, 5.5, 'India'),
       Text(0, 6.5, 'England'),
       Text(0, 7.5, 'Sri Lanka'),
       Text(0, 8.5, 'Pakistan')]
```



1.9.2 Do fuzzy matching to check problems Player names

```
[24]: distance_to_check = 2

players = np.concatenate((df.Batsman.unique(), df.Bowler.unique()))
players = pd.Series(players)
players = players.unique()
df_players = pd.DataFrame()
for player in players:
    playerCol = []
    for target in players:
        d = lv.distance(player, target)
        playerCol.append(d)
        if d > 0 and d <= distance_to_check:
            print(f"Close Match found for Player Names:{player} and {target}")
    df_players[player] = playerCol
# df_dist.values[[np.arange(df_dist.shape[0])] * 2] = 99
```

```
Close Match found for Player Names:GC Smith and DS Smith
Close Match found for Player Names:MD Bell and IR Bell
Close Match found for Player Names:JS Patel and PA Patel
Close Match found for Player Names:IR Bell and MD Bell
Close Match found for Player Names:DS Smith and GC Smith
Close Match found for Player Names:PA Patel and JS Patel
```

These player names seems close however all are valid names. So nothing to correct

```
[25]: df_players
```

```
[25]:      PA Jaques  ML Hayden  RT Ponting  MEK Hussey  MJ Clarke  A Symonds  \
      0            0          6           9           8          7          7
      1            6          0           8           6          6          8
      2            9          8           0          10          9          9
      3            8          6          10           0          8         10
      4            7          6           9           8          0          9
      ..         …        …         …         …        …        …
      171          8          8           7           9          7          8
      172          9          9           7          10          8          9
      173          8          9          10           9          9          9
      174         11         10          11          11         10         11
      175         12         12          12          13         12         13

           AC Gilchrist  GB Hogg  B Lee  MG Johnson  …  Shoaib Malik  \
      0              10        8      7           8  …            11
      1              11        7      7           7  …            11
      2              10        7      9           7  …            12
      3              12        8      8           8  …            12
      4               9        8      7           8  …            11
      ..            …      …    …         …  …           …
      171            10        7      8           8  …            11
      172            10        8      9           8  …            10
      173            11        8      7          10  …             9
      174            10       10     10          10  …             8
      175            13       13     12          12  …            11

           Misbah-ul-Haq  Faisal Iqbal  Kamran Akmal  Yasir Arafat  RS Bopara  \
      0               11            10            11            11          8
      1               11            11            11            11          8
      2               13            12            11            12          7
      3               11            12            12            11          9
      4               11            11            11            10          7
      ..             …           …           …           …        …
      171             12            11            11            10          0
      172             12            11            12            12          8
      173             11             9             8            10          9
      174             11             9            10            10         10
```

```
175              12             11             12             11          11

     SM Pollock  Umar Gul  Sohail Khan  Danish Kaneria
0             9         8           11              12
1             9         9           10              12
2             7        10           11              12
3            10         9           11              13
4             8         9           10              12
..          …        …          …             …
171           8         9           10              11
172           0        10            9              13
173          10         0            9              13
174           9         9            0              11
175          13        13           11               0

[176 rows x 176 columns]
```

1.10 Cleaning/Formatting Website Data

[26]:

```python
df_bat_ws = pd.DataFrame()
df_bow_ws = pd.DataFrame()
first_match_summary = None
for i in range(len(dfMatchInfo)):
    matchInfo = dfMatchInfo.iloc[i, :]
    dt = datetime.strptime(matchInfo["Date"], "%Y-%m-%d")
    first_match_summary = wu.get_match_summary(
        dt, matchInfo["Team1"], matchInfo["Team2"]
    )
    batting, bowling = wu.get_scorecard(first_match_summary[2])
    df_batting = pd.DataFrame(
        batting,
        columns=[
            "Innings",
            "Batsman",
            "Dismissal",
            "Runs",
            "Balls",
            "4s",
            "6s",
            "SR",
            "PercOfTotal",
        ],
    )
    df_batting["MatchId"] = matchInfo["MatchId"]
    df_bowling = pd.DataFrame(
        bowling,
        columns=[
            "Innings",
            "Bowler",
            "Overs",
            "Middens",
            "Runs",
            "Wickets",
            "ER",
            "PercOfWickets",
        ],
    )
    df_bowling["MatchId"] = matchInfo["MatchId"]
    df_bat_ws = df_bat_ws.append(df_batting)
    df_bow_ws = df_bow_ws.append(df_bowling)
```

```
Error occured for Match url: http://www.howstat.com/cricket/Statistics/Matches/M
atchScorecard.asp?MatchCode=1899 list index out of range
Error occured for Match url: http://www.howstat.com/cricket/Statistics/Matches/M
atchScorecard.asp?MatchCode=1909 list index out of range
Error occured for Match url: http://www.howstat.com/cricket/Statistics/Matches/M
atchScorecard.asp?MatchCode=1925 list index out of range
```

[27]:

```
df_bat_ws.head()
df_bat_ws[df_bat_ws.MatchId==12]
```

[27]:

```
    Innings               Batsman           Dismissal Runs Balls  4s 6s  \
0         1           M G Vandort        lbw b  Taylor   52   117   8  0
1         1       B S M Warnapura  c †Ramdin b  Bravo  120   226  14  0
2         1        K C Sangakkara  c  Smith b  Taylor   50   114   3  0
3         1  D P M D Jayawardene*         lbw b  Gayle  136   234  13  0
4         1        T T Samaraweera      c sub b  Taylor    0     2   0  0
5         1           T M Dilshan        lbw b  Taylor   20    39   4  0
6         1  H A P W Jayawardene†            b  Powell   21    83   2  0
7         1         W P U J C Vaas           not out   54   142   4  0
8         1            T Thushara       c sub b  Gayle    0     9   0  0
9         1      H M R K B Herath           not out   13     8   3  0
10        1        M Muralitharan

        SR PercOfTotal  MatchId
0    44.44      10.92%       12
1    53.10      25.21%       12
2    43.86      10.50%       12
3    58.12      28.57%       12
4     0.00                   12
5    51.28       4.20%       12
6    25.30       4.41%       12
7    38.03      11.34%       12
8     0.00                   12
```

```
9   162.50       2.73%       12
10                           12
```

[28]:
```
df_bow_ws.head()
```

[28]:
```
   Innings           Bowler Overs Middens Runs Wickets    ER PercOfWickets  \
0        1         R P Singh  26.0       3  124       4  4.77        40.00%
1        1          I Sharma  23.0       3   87       0  3.78
2        1       S C Ganguly   6.0       1   13       0  2.17
3        1   Harbhajan Singh  27.0       3  108       2  4.00        20.00%
4        1          A Kumble  25.3       0  106       4  4.16        40.00%

   MatchId
0        1
1        1
2        1
3        1
4        1
```

1.10.1 Fill blank values with appropriate values For Batsman Stats

[29]:
```
df_bat_ws["Dismissal"].replace("", "not out", inplace=True)
for col in ["Runs", "Balls", "4s", "6s", "SR", "PercOfTotal"]:
    df_bat_ws[col].replace("", 0, inplace=True)
```

[30]:
```
df_bat_ws["PercOfTotal"] = (
    df_bat_ws["PercOfTotal"].astype("str").apply(lambda x: x.replace("%", ""))
)
```

1.10.2 Fill blank values with appropriate values For Bowler Stats

[31]:
```
df_bow_ws["PercOfWickets"] = (
    df_bow_ws["PercOfWickets"].astype("str").apply(lambda x: x.replace("%", ""))
)
df_bow_ws["PercOfWickets"].replace("", "0", inplace=True)
```

[32]:
```
dup = df_bat_ws.duplicated(["MatchId", "Innings", "Batsman"]).sum()
print(f"There are {dup} duplicate records")
```

```
There are 0 duplicate records
```

1.10.3 Find duplicates for Bowler Stas

[33]:
```
dup = df_bow_ws.duplicated(["MatchId", "Innings", "Bowler"]).sum()
print(f"There are {dup} duplicate records")
```

```
There are 0 duplicate records
```

1.10.4 Replace * and † from Batsman And Dismissal columns

[34]:
```
df_bat_ws["Batsman"] = df_bat_ws["Batsman"].apply(lambda x: x.replace("*", ""))
df_bat_ws["Batsman"] = df_bat_ws["Batsman"].apply(lambda x: x.replace("†", ""))
df_bat_ws["Dismissal"] = df_bat_ws["Dismissal"].apply(lambda x: x.replace("†",␣
 ↪""))
```

1.10.5 Find closest matching player names with Fuzzy matching

find all player names from Datset downloaded

[35]:
```
player_dataset = df.Batsman.unique()
player_website = set(df_bat_ws.Batsman.unique()).union(df_bow_ws.Bowler.
 ↪unique())
```

[36]:
```
player_not_matched = {}
for player in player_dataset:
    mini = 99
    closest_match = ""
    for target in player_website:
        dist = lv.distance(player.upper(), target.upper())
        if dist < mini:
            mini = dist
            closest_match = target
        if mini == 0:
            break
    if mini > 0:
        player_not_matched[closest_match] = player
```

[37]:
```
player_not_matched
```

[37]:
```
{'P A Jaques': 'PA Jaques',
 'M L Hayden': 'ML Hayden',
 'R T Ponting': 'RT Ponting',
 'M E K Hussey': 'MEK Hussey',
 'M J Clarke': 'MJ Clarke',
 'A C Gilchrist': 'AC Gilchrist',
 'G B Hogg': 'GB Hogg',
 'M G Johnson': 'MG Johnson',
 'S R Clark': 'SR Clark',
 'V V S Laxman': 'VVS Laxman',
 'S R Tendulkar': 'SR Tendulkar',
 'S C Ganguly': 'SC Ganguly',
 'M S Dhoni': 'MS Dhoni',
 'R P Singh': 'RP Singh',
 'C H Gayle': 'CH Gayle',
 'R S Morton': 'RS Morton',
 'M N Samuels': 'MN Samuels',
 'D J J Bravo': 'DJ Bravo',
 'R N Lewis': 'RN Lewis',
 'J E Taylor': 'JE Taylor',
 'D B Powell': 'DBL Powell',
 'F H Edwards': 'FH Edwards',
 'G C Smith': 'GC Smith',
 'N D McKenzie': 'ND McKenzie',
 'H M Amla': 'HM Amla',
 'J H Kallis': 'JH Kallis',
 'A G Prince': 'AG Prince',
 'A B de Villiers': 'AB de Villiers',
 'M V Boucher': 'MV Boucher',
 'P L Harris': 'PL Harris',
 'D W Steyn': 'DW Steyn',
 'Enamul Haque': 'Enamul Haque jnr',
 'Sajidul Islam': 'Sajedul Islam',
 'C D Cumming': 'CD Cumming',
 'M D Bell': 'MD Bell',
 'P G Fulton': 'PG Fulton',
 'S P Fleming': 'SP Fleming',
 'M S Sinclair': 'MS Sinclair',
 'J D P Oram': 'JDP Oram',
 'B B McCullum': 'BB McCullum',
 'D L Vettori': 'DL Vettori',
 'K D Mills': 'KD Mills',
 "I E O'Brien": "IE O'Brien",
 'C S Martin': 'CS Martin',
 'B A Parchment': 'BA Parchment',
 'D J G Sammy': 'DJG Sammy',
 'H H Gibbs': 'HH Gibbs',
 'I K Pathan': 'IK Pathan',
 'C J L Rogers': 'CJL Rogers',
 'S W Tait': 'SW Tait',
 'R J Peterson': 'RJ Peterson',
 'J M How': 'JM How',
 'L R P L Taylor': 'LRPL Taylor',
 'J S Patel': 'JS Patel',
 'A N Cook': 'AN Cook',
 'M P Vaughan': 'MP Vaughan',
 'M J Hoggard': 'MJ Hoggard',
 'A J Strauss': 'AJ Strauss',
 'K P Pietersen': 'KP Pietersen',
 'I R Bell': 'IR Bell',
 'P D Collingwood': 'PD Collingwood',
 'T R Ambrose': 'TR Ambrose',
 'R J Sidebottom': 'RJ Sidebottom',
 'S J Harmison': 'SJ Harmison',
 'M S Panesar': 'MS Panesar',
 'S C J Broad': 'SCJ Broad',
 'M R Gillespie': 'MR Gillespie',
 'J M Anderson': 'JM Anderson',
 'M G Vandort': 'MG Vandort',
 'B S M Warnapura': 'SM Warnapura',
 'K C Sangakkara': 'KC Sangakkara',
 'D P M D Jayawardene': 'DPMD Jayawardene',
 'T T Samaraweera': 'TT Samaraweera',
 'T M Dilshan': 'TM Dilshan',
 'H A P W Jayawardene': 'HAPW Jayawardene',
 'W P U J C Vaas': 'WPUJC Vaas',
 'H M R K B Herath': 'HMRKB Herath',
 'D S Smith': 'DS Smith',
 'R R Sarwan': 'RR Sarwan',
 'R O Hinds': 'RO Hinds',
 'S J Benn': 'SJ Benn',
 'G D Elliott': 'GD Elliott',
 'T G Southee': 'TG Southee',
 'L P C Silva': 'LPC Silva',
 'M K D Amerasinghe': 'MKDI Amerasinghe',
 'P P Chawla': 'PP Chawla',
 'A J Redmond': 'AJ Redmond',
 'J A H Marshall': 'JAH Marshall',
 'D R Flynn': 'DR Flynn',
 'S M Katich': 'SM Katich',
 'B J Hodge': 'BJ Hodge',
 'B J Haddin': 'BJ Haddin',
 'S C G MacGill': 'SCG MacGill',
 'A S Jaggernauth': 'AS Jaggernauth',
 'X M Marshall': 'XM Marshall',
 'G J Hopkins': 'GJ Hopkins',
 'D J Pattinson': 'DJ Pattinson',
 'K D Karthik': 'KD Karthik',
 'K M D N Kulasekara': 'KMDN Kulasekara',
 'B A W Mendis': 'BAW Mendis',
 'P A Patel': 'PA Patel',
 'K T G D Prasad': 'KTGD Prasad',
 'S R Watson': 'SR Watson',
 'C L White': 'CL White',
 'P M Siddle': 'PM Siddle',
 'J D Ryder': 'JD Ryder',
 'Mehrab Hossain': 'Mehrab Hossain jnr',
 'J J Krejza': 'JJ Krejza',
 'N M Hauritz': 'NM Hauritz',
 'M J Prior': 'MJ Prior',
 'G P Swann': 'GP Swann',
 'J-P Duminy': 'JP Duminy',
 'A B McDonald': 'AB McDonald',
 'D E Bollinger': 'DE Bollinger',
 'C K Kapugedera': 'CK Kapugedera',
 'C R D Fernando': 'CRD Fernando',
 'B P Nash': 'BP Nash',
 'O A Shah': 'OA Shah',
 'N T Paranavitana': 'NT Paranavitana',
 'R S Bopara': 'RS Bopara'}
```

[38]:

```
def replace_player_name(x):
    if x in player_not_matched:
        return player_not_matched[x]
    else:
        return x


def replace_player_name_in_dismissal(x):
    for key,    value in player_not_matched.items():
        if key in x:
            x = x.replace(key, value)
    return x


def get_dismissed_by_bowler(x):
    if " b " in x:
        splts = x.split(" b ")
        return splts[-1].strip()
    elif x.startswith("b "):
        splts = x.split("b ")
        return splts[-1].strip()
    return ""
```

[39]:

```
df_bat_ws["Batsman"] = df_bat_ws["Batsman"].apply(lambda x:␣
 ↪replace_player_name(x))
df_bow_ws["Bowler"] = df_bow_ws["Bowler"].apply(lambda x:␣
 ↪replace_player_name(x))
df_bat_ws["Dismissal"] = df_bat_ws["Dismissal"].apply(
    lambda x: replace_player_name_in_dismissal(x)
)
```

1.10.6 Correct Batsman Name

[40]:

```
df_bat_ws.head()
df_bat_ws.Batsman.str.replace("SM Warnapura", "BSM Warnapura")
```

```
[40]: 0          PA Jaques
      1          ML Hayden
      2         RT Ponting
      3         MEK Hussey
      4          MJ Clarke
                 …
      28        TR Ambrose
      29         SCJ Broad
      30       JM Anderson
      31          GP Swann
      32     RJ Sidebottom
      Name: Batsman, Length: 1892, dtype: object
```

```
[41]: df_bow_ws.head()
```

```
[41]:   Innings           Bowler Overs Middens Runs Wickets    ER PercOfWickets  \
      0       1         RP Singh  26.0       3  124       4  4.77         40.00
      1       1         I Sharma  23.0       3   87       0  3.78             0
      2       1       SC Ganguly   6.0       1   13       0  2.17             0
      3       1  Harbhajan Singh  27.0       3  108       2  4.00         20.00
      4       1         A Kumble  25.3       0  106       4  4.16         40.00

         MatchId
      0        1
      1        1
      2        1
      3        1
      4        1
```

1.10.7 Find bowler name from Dismissal column

```
[42]: df_bat_ws["Bowler"] = df_bat_ws["Dismissal"].apply(lambda x:␣
       ↪get_dismissed_by_bowler(x))
```

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

```
[43]:    Innings     Batsman                Dismissal Runs Balls 4s 6s     SR  \
      0        1   PA Jaques       c Dhoni b RP Singh    0     9  0  0   0.00
      1        1   ML Hayden  c  Tendulkar b RP Singh   13    26  2  0  50.00
      2        1  RT Ponting   lbw b  Harbhajan Singh   55    69  9  0  79.71
      3        1  MEK Hussey  c  Tendulkar b RP Singh   41    79  3  0  51.90
      4        1   MJ Clarke   lbw b  Harbhajan Singh    1     4  0  0  25.00

        PercOfTotal  MatchId           Bowler
      0           0        1         RP Singh
      1        2.81        1         RP Singh
      2       11.88        1  Harbhajan Singh
```

```
3        8.86        1         RP Singh
4        0.22        1  Harbhajan Singh
```

[44]:
```
player_website = set(df_bat_ws.Batsman.unique()).union(df_bow_ws.Bowler.
 ↪unique())
```

1.11 API Datasets

1.11.1 Get Profile-Ids By using API

1. Tried to find player by exact match first

2. If not found in above step, searched with last-name

3. Used Fuzzy Matching for name matching if not found with exact match

4. Helper Modules

  1. `pid_api_utils`
  2. `profile_api_utils`
  3. `web_profile_id_utis`

[ ]:
```
df_player_profiles = pd.DataFrame(columns=["PlayerName", "Profile-Id"])
i = 0
for player in player_website:
    #print(player)
    df_player_profiles = df_player_profiles.append(
        {"PlayerName": player, "Profile-Id": apiu.get_profile_id(player)[1]},
        ignore_index=True,
    )
df_player_profiles.to_csv("Player-Profile.csv", index = False)
```

1.11.2 As API has limit on Daily requests(100 Request/Day), written this backup approach with web-scrapping for Profile-Ids

[ ]:
```
df_player_profiles = pd.DataFrame(columns=["PlayerName", "Profile-Id"])
for player in player_website:
    print(player)
    df_player_profiles = df_player_profiles.append(
        {"PlayerName": player, "Profile-Id": wuapi.get_profile_id(player)[1]},
        ignore_index=True,
    )
df_player_profiles.to_csv("Player-Profile.csv", index = False)
```

1.11.3 Read Profile-Ids from file

[82]:

```
df_player_profiles = pd.read_csv("Player-Profile.csv")
```

1.11.4 Correct Profile-Ids as API has some issues with Search functionality

[95]:

```
players_need_update = {"Aftab Ahmed":56266, "Junaid Siddique":55946, "CL White":
 ↪8291, "Yasir Arafat":43654, "DPMD Jayawardene":49289, "SM Warnapura":50874,␣
 ↪"Naeem Islam":56054, "Shahadat Hossain":56149 ,"Sohail Khan":317252}
for key, val in players_need_update.items():
    df_player_profiles.at[df_player_profiles[df_player_profiles.
 ↪PlayerName==key].index[0], "Profile-Id"] = val
```

1.11.5 Read Profile with Statastics from API

[83]:

```
from test_match_helpers import profile_api_utils as papiu
df_profile_bat = pd.DataFrame()
df_profile_bowl = pd.DataFrame()
for pid in df_player_profiles["Profile-Id"]:
    batting, bowling = papiu.get_profile(str(pid))
    batting["Profile-Id"] = pid
    bowling["Profile-Id"] = pid
    df_profile_bat=df_profile_bat.append(batting, ignore_index=True)
    df_profile_bowl=df_profile_bowl.append(bowling, ignore_index=True)

df_profile_bat.to_csv("Player-Profile-Bat.csv", index = False)
df_profile_bowl.to_csv("Player-Profile-Bowl.csv", index = False)
```

1.11.6 Read Profile Stats from Files

[84]:

```
df_profile_bat = pd.read_csv("Player-Profile-Bat.csv")
df_profile_bowl = pd.read_csv("Player-Profile-Bowl.csv")
```

1.11.7 Batting Stats

[85]:

```
df_profile_bat.head()
```

[85]:

```
   100      4s  50  6s    Ave       BF   Ct    HS  Inns  Mat  NO  Profile-Id  \
0   22   919.0  46  39  42.69  15622.0  100   235   205  118  24      9062.0
1    6   372.0  12  14  30.79   4196.0  184  158*    92   53   6     41028.0
2    1    55.0   3   4  29.80    963.0   31   102    16   11   1      8845.0
3    0    48.0   0   2   6.56   1397.0   10    30    88   55  28     51782.0
4   27  1165.0  38  24  48.25  15525.0  169   277   205  117  13     47270.0

   Runs     SR  St
0  7727  49.46   0
```

```
1   2648  63.10   22
2    447  46.41    0
3    394  28.20    0
4   9265  59.67    0
```

1.11.8 Bowling Stats

[86]:
```python
df_profile_bowl.head()
```

[86]:
```
   10  4w   5w     Ave    BBI    BBM  Balls  Econ  Inns  Mat  Profile-Id  Runs  \
0   0   0    0   76.00   1/33   1/33    108  4.22     6  118      9062.0    76
1   -   -    -       -      -      -      -     -     -   53     41028.0     -
2   -   -    -       -      -      -      -     -     -   11      8845.0     -
3   0   3   12   37.87   7/87  8/132   9602  3.90    97   55     51782.0  6249
4   0   0    0  110.62  2/145  2/145   1418  3.74    37  117     47270.0   885

      SR  Wkts
0  108.0     1
1      -     -
2      -     -
3   58.1   165
4  177.2     8
```

1.11.9 Rename Columns names Before merging

[87]:
```python
df_profile_bat.columns = "bat_" + df_profile_bat.columns.values
df_profile_bowl.columns = "bowl_" + df_profile_bowl.columns.values
```

1.11.10 Join profile-Ids, batting, and bowling sataframe into one dataframe by using key as Profile-Id

[88]:
```python
df_profiles_all = pd.merge(df_player_profiles, df_profile_bat, how="outer",␣
 ↪left_on="Profile-Id", right_on="bat_Profile-Id")
df_profiles_all = pd.merge(df_profiles_all, df_profile_bowl, how="outer",␣
 ↪left_on="Profile-Id", right_on="bowl_Profile-Id")
```

1.11.11 Drop duplicate columns

[89]:
```python
df_profiles_all.drop(columns=["bat_Profile-Id", "bowl_Profile-Id"],␣
 ↪inplace=True)
```

1.11.12 Replace "-" with pd.np.nan, Intentionally keeping it as NaN to indcate the stats is not applicable for that player

[90]:
```
df_profiles_all = df_profiles_all.apply(lambda x: x.replace("-", pd.np.nan))
```

1.11.13 Missing Values

[91]:
```
df_profiles_all[df_profiles_all.bat_BF.isna()]
```

[91]:
```
       PlayerName  Profile-Id  bat_100  bat_4s  bat_50  bat_6s bat_Ave  bat_BF  \
63    IE O'Brien        5327        0     NaN       0       1    9.00     NaN
66  SR Tendulkar       35320       51     NaN      68      69   53.78     NaN

   bat_Ct bat_HS  …  bowl_Ave  bowl_BBI  bowl_BBM  bowl_Balls  bowl_Econ  \
63      0    16*  …     37.38     6/110    10/239        3093       3.04
66    115   248*  …     54.17      3/10      3/14        4240       3.52

   bowl_Inns bowl_Mat bowl_Runs bowl_SR bowl_Wkts
63        18       10      1570    73.6        42
66       145      200      2492    92.1        46

[2 rows x 29 columns]
```

1.11.14 Find invalid records or records with all missing values

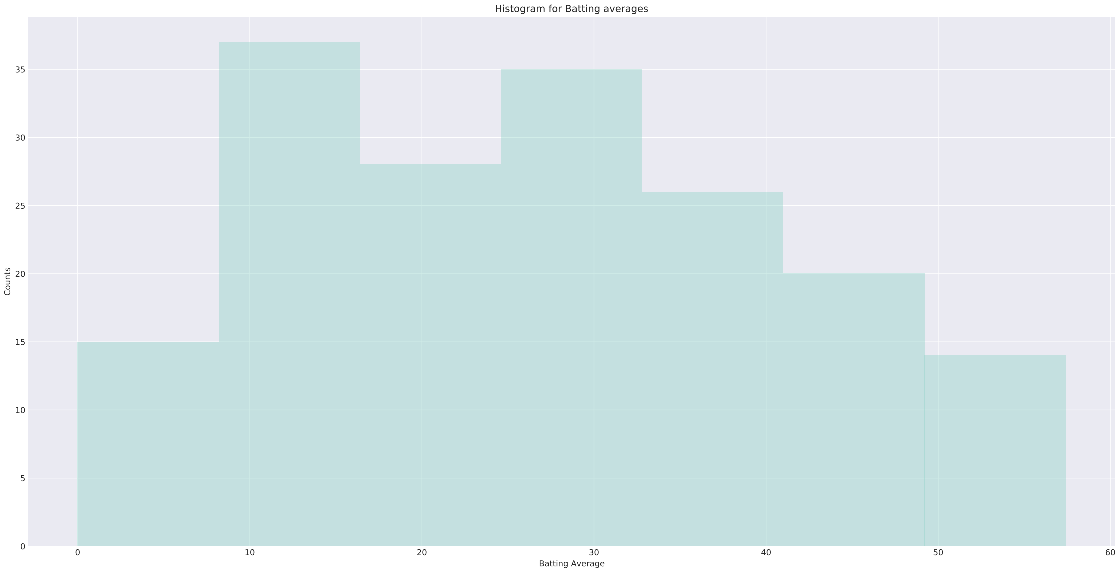
[98]:
```
df_profiles_all[df_profiles_all.bat_HS.isna()]
```

[98]:
```
Empty DataFrame
Columns: [PlayerName, Profile-Id, bat_100, bat_4s, bat_50, bat_6s, bat_Ave,
bat_BF, bat_Ct, bat_HS, bat_Inns, bat_Mat, bat_NO, bat_Runs, bat_SR, bat_St,
bowl_10, bowl_4w, bowl_5w, bowl_Ave, bowl_BBI, bowl_BBM, bowl_Balls, bowl_Econ,
bowl_Inns, bowl_Mat, bowl_Runs, bowl_SR, bowl_Wkts]
Index: []

[0 rows x 29 columns]
```

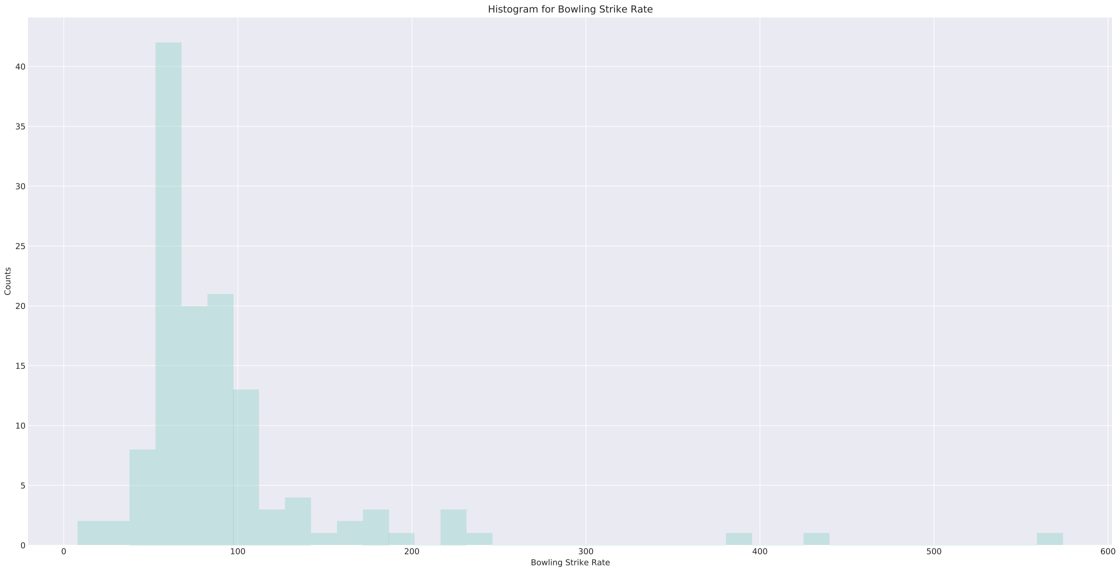
1.11.15 Histogram for batting averages to find outliers

[92]:
```
ax = sns.distplot(df_profiles_all.bat_Ave, kde=False)
ax.set(xlabel="Batting Average", ylabel="Counts", title="Histogram for Batting␣
 ↪averages")
plt.show()
```



1.11.16 Histogram for bowling strike rates to find outliers

```
[93]: ax = sns.distplot(df_profiles_all.bowl_SR, kde=False)
      ax.set(xlabel="Bowling Strike Rate", ylabel="Counts", title="Histogram for␣
       ↪Bowling Strike Rate")
      plt.show()
```



```
[94]: df_profiles_all[df_profiles_all.bowl_SR.astype(float) < 15]
```

```
[94]:      PlayerName  Profile-Id  bat_100  bat_4s  bat_50  bat_6s bat_Ave   bat_BF  \
      146  MV Boucher       44111        5   656.0      35      20   30.30  11005.0

           bat_Ct bat_HS  …  bowl_Ave  bowl_BBI  bowl_BBM  bowl_Balls  bowl_Econ  \
      146     532    125  …      6.00       1/6       1/6           8       4.50

           bowl_Inns bowl_Mat bowl_Runs bowl_SR bowl_Wkts
      146          1      147         6     8.0         1

      [1 rows x 29 columns]
```

As this bowler “Mark Boucher” only bowled 8 balls, this stats is valid