# 22110089-es114-dn3

May 12, 2023

### 0.0.1 Setting-Up and Importing Libraries

**Files will automatically download from Google Drive once this whole cell runs. No Need for re-uploading each time**

[ ]:
```
!pip install -U -q PyDrive

from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials

auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)
```

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

links = ["https://drive.google.com/file/d/1hIc6Pg7DGNr4j6nLk8ZuSgrlAbejxO9L/
 ↪view?usp=sharing", "https://drive.google.com/file/d/
 ↪1av5MvvJ4s9PrcwETDooRVds3u5Mnz2WU/view?usp=share_link", "https://drive.
 ↪google.com/file/d/1395XRX9gOep5YsY2YoivxaviK8jqcWJG/view?usp=share_link",␣
 ↪"https://drive.google.com/file/d/16wnCOMY_grOMEuH9FDH1fsqd_PXpW41-/view?
 ↪usp=share_link", "https://drive.google.com/file/d/
 ↪1Pz946xTifVAtg7PxAn8s5A6ilIdrBxPr/view?usp=share_link", "https://drive.
 ↪google.com/file/d/1iHr5hsxJAM8X-Q0Np206_PayM9Qs8pUx/view?usp=share_link",␣
 ↪"https://drive.google.com/file/d/1wrPlAeTMV7ra11AE11hdNxhxlQx8PSdr/view?
 ↪usp=share_link", "https://drive.google.com/file/d/
 ↪1hrAMvDNRn4fgOvmqCTsTw7yhNzPIjh2g/view?usp=share_link"]
names = ["AusOpen-men-2013.csv", "AusOpen-women-2013.csv", "FrenchOpen-men-2013.
 ↪csv", "FrenchOpen-women-2013.csv", "USOpen-men-2013.csv", "USOpen-women-2013.
 ↪csv", "Wimbledon-men-2013.csv", "Wimbledon-women-2013.csv"]
for i in range(8):
  id = links[i].split("/")[-2]
```

```
  downloaded = drive.CreateFile({"id" : id})
  downloaded.GetContentFile(names[i])

ausM = pd.read_csv("AusOpen-men-2013.csv")
ausW = pd.read_csv("AusOpen-women-2013.csv")
fraM = pd.read_csv("FrenchOpen-men-2013.csv")
fraW = pd.read_csv("FrenchOpen-women-2013.csv")
usM = pd.read_csv("USOpen-men-2013.csv")
usW = pd.read_csv("USOpen-women-2013.csv")
wimM = pd.read_csv("Wimbledon-men-2013.csv")
wimW = pd.read_csv("Wimbledon-women-2013.csv")
ausM = ausM.fillna(0)
ausW = ausW.fillna(0)
fraM = fraM.fillna(0)
fraW = fraW.fillna(0)
usM = usM.fillna(0)
usW = usW.fillna(0)
wimM = wimM.fillna(0)
wimW = wimW.fillna(0)
```

[ ]:
```
%matplotlib inline
```

### 0.0.2 Libraries

[ ]:
```
!pip install kaleido
!pip install plotly>=4.0.0
!wget https://github.com/plotly/orca/releases/download/v1.2.1/orca-1.2.1-x86_64.
  ↪AppImage -O /usr/local/bin/orca
!chmod +x /usr/local/bin/orca
!apt-get install xvfb libgtk2.0-0 libgconf-2-4
import plotly.express as px
import plotly.figure_factory as ff
import os
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split, cross_val_score,␣
  ↪LeaveOneOut
from sklearn.metrics import mean_squared_error, accuracy_score
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Requirement already satisfied: kaleido in /usr/local/lib/python3.10/dist-
packages (0.2.1)
/usr/local/bin/orca: Text file busy
Reading package lists… Done
```

```
Building dependency tree
Reading state information… Done
libgtk2.0-0 is already the newest version (2.24.32-4ubuntu4).
libgconf-2-4 is already the newest version (3.2.6-6ubuntu1).
xvfb is already the newest version (2:1.20.13-1ubuntu1~20.04.8).
0 upgraded, 0 newly installed, 0 to remove and 24 not upgraded.
```

[ ]:
```
import plotly.graph_objects as go
```

[ ]:
```
print(ausM.shape, ausW.shape, fraM.shape, fraW.shape, usM.shape, usW.shape,␣
↪wimW.shape, wimW.shape)
```

```
(126, 42) (127, 42) (125, 42) (127, 42) (126, 42) (76, 42) (122, 42) (122, 42)
```

**The Column Names**

[ ]:
```
# Player 1              Name of Player 1
# Player 2              Name of Player 2
# Result                Result of the match (0/1) - Referenced on Player 1 is␣
↪Result = 1 if Player 1 wins (FNL.1>FNL.2)
# FSP.1                 First Serve Percentage for player 1 (Real Number)
# FSW.1                 First Serve Won by player 1 (Real Number)
# SSP.1                 Second Serve Percentage for player 1 (Real Number)
# SSW.1                 Second Serve Won by player 1 (Real Number)
# ACE.1                 Aces won by player 1 (Numeric-Integer)
# DBF.1                 Double Faults committed by player 1 (Numeric-Integer)
# WNR.1                 Winners earned by player 1 (Numeric)
# UFE.1                 Unforced Errors committed by player 1 (Numeric)
# BPC.1                 Break Points Created by player 1   (Numeric)
# BPW.1                 Break Points Won by player 1    (Numeric)
# NPA.1                 Net Points Attempted by player 1 (Numeric)
# NPW.1                 Net Points Won by player 1  (Numeric)
# TPW.1                 Total Points Won by player 1 (Numeric)
# ST1.1                 Set 1 result for Player 1 (Numeric-Integer)
# ST2.1                 Set 2 Result for Player 1 (Numeric-Integer)
# ST3.1                 Set 3 Result for Player 1 (Numeric-Integer)
# ST4.1                 Set 4 Result for Player 1 (Numeric-Integer)
# ST5.1                 Set 5 Result for Player 1 (Numeric-Integer)
# FNL.1                 Final Number of Games Won by Player 1 (Numeric-Integer)
# FSP.2                 First Serve Percentage for player 2 (Real Number)
# FSW.2                 First Serve Won by player 2 (Real Number)
# SSP.2                 Second Serve Percentage for player 2 (Real Number)
# SSW.2                 Second Serve Won by player 2 (Real Number)
# ACE.2                 Aces won by player 2 (Numeric-Integer)
# DBF.2                 Double Faults committed by player 2 (Numeric-Integer)
# WNR.2                 Winners earned by player 2 (Numeric)
# UFE.2                 Unforced Errors committed by player 2 (Numeric)
# BPC.2                 Break Points Created by player 2   (Numeric)
```

```
# BPW.2                 Break Points Won by player 2    (Numeric)
# NPA.2                 Net Points Attempted by player 2 (Numeric)
# NPW.2                 Net Points Won by player 2  (Numeric)
# TPW.2                 Total Points Won by player 2 (Numeric)
# ST1.2                 Set 1 result for Player 2 (Numeric-Integer)
# ST2.2                 Set 2 Result for Player 2 (Numeric-Integer)
# ST3.2                 Set 3 Result for Player 2 (Numeric-Integer)
# ST4.2                 Set 4 Result for Player 2 (Numeric-Integer)
# ST5.2                 Set 5 Result for Player 2 (Numeric-Integer)
# FNL.2                 Final Number of Games Won by Player 2 (Numeric-Integer)
# Round                 Round of the tournament at which game is played␣
 ↪(Numeric-Integer)
```

### 0.0.3 Australia Open Men's Tournament

[ ]:
```
ausM
```

[ ]:
```
              Player1             Player2  Round  Result  FNL1  FNL2  FSP.1  \
0         Lukas Lacko      Novak Djokovic      1       0     0     3     61
1      Leonardo Mayer     Albert Montanes      1       1     3     0     61
2    Marcos Baghdatis       Denis Istomin      1       0     0     3     52
3     Dmitry Tursunov     Michael Russell      1       1     3     0     53
4          Juan Monaco      Ernests Gulbis      1       0     1     3     76
..                 …                 …    …      …   …   …     …
121       Andy Murray       Roger Federer      5       0     1     3     61
122      Rafael Nadal     Grigor Dimitrov      5       1     3     1     73
123     Tomas Berdych  Stanislas Wawrinka      6       0     1     3     62
124      Rafael Nadal       Roger Federer      6       1     3     0     65
125      Rafael Nadal  Stanislas Wawrinka      7       0     1     3     78

     FSW.1  SSP.1  SSW.1  …  BPC.2  BPW.2  NPA.2  NPW.2  TPW.2  ST1.2  \
0       35     39     18  …      4      8    8.0    9.0    101      6
1       31     39     13  …      0      0    0.0    0.0     42      1
2       53     48     20  …      4     13   12.0   16.0    126      6
3       39     47     24  …      1      7    0.0    0.0     79      2
4       63     24     12  …      3      5   16.0   28.0    127      1
..     …    …    …  …    …    …    …    …    …    …
121     60     39     28  …      4     17   49.0   66.0    147      6
122     66     27     22  …      3      6   28.0   41.0    132      6
123     71     38     30  …      1      4   14.0   18.0    143      6
124     41     35     22  …      1      2   23.0   42.0     86      6
125     50     22     10  …      5     15   11.0   12.0    116      6

     ST2.2  ST3.2  ST4.2  ST5.2
0      7.0    6.0    0.0    0.0
1      3.0    1.0    0.0    0.0
2      7.0    6.0    0.0    0.0
```

```
3      2.0    3.0    0.0    0.0
4      6.0    7.0    6.0    0.0
..     …      …      …      …
121    6.0    6.0    6.0    0.0
122    6.0    6.0    2.0    0.0
123    6.0    7.0    7.0    0.0
124    3.0    3.0    0.0    0.0
125    6.0    3.0    6.0    0.0

[126 rows x 42 columns]
```

**Distributions of First-Serve Percentages for Player 1 and Player 2: They have no difference, both follow similar distribution**

[ ]:
```
hist_data = [ausM["FSP.1"], ausM["FSP.2"]]
group_labels = ["P1: First-Serve %", "P2: First-Serve %"]
fig = ff.create_distplot(hist_data, group_labels, bin_size = 1.5, colors =␣
↪["#37AA9C", "purple"])
fig.show()
fig.write_image("fig1_0.png", engine = "orca")
```

**Correlation between First-Serve Percentage and First-Serve Won by Player 1**

[ ]:
```
fig = px.scatter(ausM, x = "FSP.1", y = "FSW.1", color = "Player1", trendline =␣
↪"ols", trendline_scope = "overall", trendline_color_override = "deeppink")
fig.show()
fig.write_image("fig1_1.png", engine = "orca")
```

**Correlation Factor**

[ ]:
```
print("Correlation Factor:")
print(ausM["FSW.1"].corr(ausM["FSP.1"]))
```

```
Correlation Factor:
0.18869064127952748
```

[ ]:
```
fig = px.scatter(ausM, x = "FSP.2", y = "FSW.2", color = "Player2", trendline =␣
↪"ols", trendline_scope = "overall", trendline_color_override = "deeppink")
fig.show()
fig.write_image("fig1_2.png", engine = "orca")
```

[ ]:
```
print("Correlation Factor:")
print(ausM["FSW.2"].corr(ausM["FSP.2"]))
```

```
Correlation Factor:
0.2671214679032754
```

### 0.0.4 Australia Open Women's Tournament

[ ]:
```
ausW
```

[ ]:
```
                Player1             Player2  Round  Result  FNL1  FNL2  \
0        Serena Williams       Ashleigh Barty      1       1   2.0   0.0
1           Vesna Dolonc    Lara Arruabarrena      1       1   2.0   1.0
2     Pauline Parmentier    Karolina Pliskova      1       0   0.0   2.0
3          Heather Watson   Daniela Hantuchova      1       0   1.0   2.0
4        Samantha Stosur     Klara Zakopalova      1       1   2.0   0.0
..                   …                   …    …      …   …   …
122         Simona Halep   Dominika Cibulkova      5       0   0.0   2.0
123  Agnieszka Radwanska    Victoria Azarenka      5       1   2.0   1.0
124     Eugenie Bouchard                Na Li      6       0   0.0   2.0
125   Dominika Cibulkova  Agnieszka Radwanska      6       1   2.0   0.0
126                Na Li   Dominika Cibulkova      7       1   2.0   0.0

     FSP.1  FSW.1  SSP.1  SSW.1  …  BPC.2  BPW.2  NPA.2  NPW.2  TPW.2  \
0       59     20     41      8  …    0.0    0.0    2.0    4.0     31
1       65     33     35     10  …    4.0    7.0    0.0    0.0     74
2       63     16     37      4  …    5.0   14.0    0.0    0.0     64
3       61     41     39     19  …    5.0   13.0    5.0    8.0    102
4       65     28     35     11  …    4.0   14.0   10.0   15.0     60
..     …    …    …    …  …    …    …    …    …    …
122     67     13     33      6  …    5.0    9.0    3.0    4.0     54
123     59     33     41     16  …    2.0    5.0   20.0   34.0     74
124     45     13     55      5  …    6.0   10.0   11.0   14.0     71
125     64     22     36     10  …    1.0    9.0    4.0    9.0     40
126     60     21     40     15  …    2.0    3.0    3.0    4.0     58

     ST1.2  ST2.2  ST3.2  ST4.2  ST5.2
0      2.0    1.0    0.0    0.0    0.0
1      6.0    2.0    4.0    0.0    0.0
2      6.0    6.0    0.0    0.0    0.0
3      7.0    3.0    6.0    0.0    0.0
4      3.0    4.0    0.0    0.0    0.0
..     …    …    …    …    …
122    6.0    6.0    0.0    0.0    0.0
123    1.0    7.0    0.0    0.0    0.0
124    6.0    6.0    0.0    0.0    0.0
125    1.0    2.0    0.0    0.0    0.0
126    6.0    0.0    0.0    0.0    0.0

[127 rows x 42 columns]
```

[ ]:
```
df = ausW.iloc[:,4:19]
cov = df.corr()
```

```
cov = cov.fillna(0)
```

**Which Features have high correlation factors**

[ ]:
```
fig = px.imshow(cov, text_auto = ".2f")
fig.layout.height = 800
fig.layout.width = 800
fig.show()
fig.write_image("fig2_0.png", engine = "orca")
```

[ ]:
```
fig = px.scatter(ausW, x = "FSP.1", y = "SSP.1", color = "Player1", trendline =␣
↪"ols", trendline_scope = "overall", trendline_color_override = "deeppink")
fig.show()
fig.write_image("fig2_1.png", engine = "orca")
```

[ ]:
```
fig = px.scatter(ausW, x = "NPA.1", y = "NPW.1", color = "Player1", trendline =␣
↪"ols", trendline_scope = "overall", trendline_color_override = "lime")
fig.show()
fig.write_image("fig2_1.png", engine = "orca")
```

### 0.0.5 French Open Men's Tournament

[ ]:
```
fraM
```

[ ]:

| | Player1 | Player2 | Round | Result | FNL.1 | \ |
|---|---|---|---|---|---|---|
| 0 | Pablo Carreno-Busta | Roger Federer | 1 | 0 | 0 | |
| 1 | Somdev Devvarman | Daniel Munoz-De La Nava | 1 | 1 | 3 | |
| 2 | Tobias Kamke | Paolo Lorenzi | 1 | 1 | 3 | |
| 3 | Julien Benneteau | Ricardas Berankis | 1 | 1 | 3 | |
| 4 | Lukas Lacko | Sam Querrey | 1 | 0 | 0 | |
| .. | … | … | … | … | … | |
| 120 | Rafael Nadal | Stanislas Wawrinka | 5 | 1 | 3 | |
| 121 | Novak Djokovic | Tommy Haas | 5 | 1 | 3 | |
| 122 | David Ferrer | Jo-Wilfried Tsonga | 6 | 1 | 3 | |
| 123 | Novak Djokovic | Rafael Nadal | 6 | 0 | 2 | |
| 124 | Rafael Nadal | David Ferrer | 7 | 1 | 3 | |

| | FNL.2 | FSP.1 | FSW.1 | SSP.1 | SSW.1 | … | BPC.2 | BPW.2 | NPA.2 | NPW.2 | \ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 3 | 62 | 27 | 38 | 11 | … | 7 | 7 | 14 | 18 | |
| 1 | 0 | 62 | 54 | 38 | 22 | … | 1 | 16 | 22 | 25 | |
| 2 | 2 | 62 | 53 | 38 | 15 | … | 10 | 18 | 19 | 27 | |
| 3 | 1 | 72 | 87 | 28 | 19 | … | 4 | 13 | 33 | 43 | |
| 4 | 3 | 52 | 31 | 48 | 22 | … | 4 | 7 | 12 | 13 | |
| .. | … | … | … | … | … | … | … | … | … | … | |
| 120 | 0 | 75 | 40 | 25 | 11 | … | 1 | 5 | 16 | 30 | |
| 121 | 0 | 64 | 41 | 36 | 22 | … | 2 | 2 | 2 | 17 | |
| 122 | 0 | 60 | 35 | 40 | 23 | … | 2 | 5 | 7 | 16 | |

```
123      3     67     76     33     30  …      8     16     15     26
124      0     70     43     30     11  …      3     12     10     14

     TPW.2  ST1.2  ST2.2  ST3.2  ST4.2  ST5.2
0       88      6      6    6.0    0.0    0.0
1      106      3      3    5.0    0.0    0.0
2      139      3      3    6.0    6.0    3.0
3      149      6      3    7.0    6.0    0.0
4       93      6      6    6.0    0.0    0.0
..     …    …    …    …    …    …
120     64      2      3    1.0    0.0    0.0
121     84      3      6    5.0    0.0    0.0
122     84      1      6    2.0    0.0    0.0
123    177      6      3    6.0    6.0    9.0
124     72      3      2    3.0    0.0    0.0

[125 rows x 42 columns]
```

**Breaking Points created and Won by the Winner/Loser**

[ ]:
```
fig = px.box(fraM, x = "Result", y = "BPC.1", color = "Result", notched = True)
fig.show()
fig.write_image("fig3_0.png", engine = "orca")
```

[ ]:
```
fig = px.box(fraM, x = "Result", y = "BPW.1", color = "Result", notched = True)
fig.show()
fig.write_image("fig3_1.png", engine = "orca")
```

[ ]:
```
fig = px.box(fraM, x = "Result", y = "BPC.2", color = "Result", notched = True)
fig.show()
fig.write_image("fig3_2.png", engine = "orca")
```

[ ]:
```
fig = px.box(fraM, x = "Result", y = "BPW.2", color = "Result", notched = True)
fig.show()
fig.write_image("fig3_3.png", engine = "orca")
```

### 0.0.6 French Open Women's Tournament

[ ]:
```
fraW
```

[ ]:
```
                Player1             Player2  Round  Result  FNL.1  FNL.2  \
0          Su-Wei Hsieh     Maria Sharapova      1       0      0      2
1      Eugenie Bouchard  Tsvetana Pironkova      1       1      2      0
2             Jie Zheng       Vesna Dolonc      1       1      2      0
3         Tamira Paszek       Melanie Oudin      1       0      0      2
4           Karin Knapp     Sloane Stephens      1       0      0      2
..                  …                  …    …      …    …    …
```

```
122  Agnieszka Radwanska          Sara Errani      5       0       0       2
123      Serena Williams  Svetlana Kuznetsova      5       1       2       1
124    Victoria Azarenka      Maria Sharapova      6       0       1       2
125      Serena Williams          Sara Errani      6       1       2       0
126      Serena Williams      Maria Sharapova      7       1       2       0

     FSP.1  FSW.1  SSP.1  SSW.1  …  BPC.2  BPW.2  NPA.2  NPW.2  TPW.2  \
0       62     18     38      5  …      4      6      3      5     57
1       57     23     43     17  …      1      3      4      8     48
2       76     30     24      5  …      0      4     14     20     56
3       59     16     41      8  …      8     13      5      8     78
4       57     18     43     13  …      5      7      1      4     61
..     …    …    …    …  …    …    …    …    …    …
122     70     28     30      5  …      6      7     16     24     80
123     66     42     34     12  …      4      9      3      6     75
124     72     28     28      8  …      6     10      2      6     87
125     52     14     48     14  …      0      0      2      2     16
126     69     27     31      8  …      2      2      3      4     56

     ST1.2  ST2.2  ST3.2  ST4.2  ST5.2
0        6    6.0    0.0    0.0    0.0
1        1    6.0    0.0    0.0    0.0
2        4    1.0    0.0    0.0    0.0
3        6    6.0    0.0    0.0    0.0
4        6    7.0    0.0    0.0    0.0
..     …    …    …    …    …
122      6    7.0    0.0    0.0    0.0
123      1    6.0    3.0    0.0    0.0
124      6    2.0    6.0    0.0    0.0
125      0    1.0    0.0    0.0    0.0
126      4    4.0    0.0    0.0    0.0

[127 rows x 42 columns]
```

### Segregating Data into Features and Target Array

[ ]:
```
features = fraW[["FSP.1", "FSW.1", "SSP.1",        "SSW.1", "ACE.1", "DBF.1",␣
 ↪"WNR.1", "UFE.1", "BPC.1", "BPW.1", "NPA.1", "TPW.1", "FSP.2", "FSW.2", "SSP.
 ↪2",        "SSW.2", "ACE.2", "DBF.2", "WNR.2", "UFE.2", "BPC.2", "BPW.2",␣
 ↪"NPA.2", "TPW.2"]].to_numpy()
target = fraW["Result"].to_numpy()
print(features)
print()
print(target)
```

```
[[62. 18. 38. …  6.  3. 57.]
 [57. 23. 43. …  3.  4. 48.]
 [76. 30. 24. …  4. 14. 56.]
```

```
…
[72. 28. 28. … 10.  2. 87.]
[52. 14. 48. …  0.  2. 16.]
[69. 27. 31. …  2.  3. 56.]]

[0 1 1 0 0 1 1 1 0 1 0 1 1 1 1 1 0 0 0 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 1 1 0
 0 1 0 0 0 1 0 1 1 1 1 0 1 0 1 1 0 0 1 0 1 1 1 0 0 0 1 0 0 0 0 0 1 0 1 0 1
 1 1 0 1 1 0 0 1 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0
 1 0 0 0 0 0 1 1 1 0 0 0 1 0 1 1]
```

**For classification into Win/Loss, GaussianNB model is trained**

[ ]:
```
model = GaussianNB()
Xtrain, Ytrain = features[:26], target[:26]
Xtest, Ytest = features[26:], target[26:]
model.fit(Xtrain, Ytrain)
y_model = model.predict(Xtest)
print(accuracy_score(y_model, Ytest))
```

```
0.7722772277227723
```

**Cross Validation Score**

[ ]:
```
model = GaussianNB()
scores = cross_val_score(model, features, target, cv = 5)
print(scores)
```

```
[1.         0.88461538 0.92       0.84       0.88       ]
```

**Average Score**

[ ]:
```
print(scores.mean())
```

```
0.9049230769230769
```

### 0.0.7 US Open Men's Tournament

[ ]:
```
usM
```

[ ]:
```
                Player1             Player2  Round  Result  FNL1  FNL2  FSP.1  \
0       Richard Gasquet     Michael Russell      1       1     3     0     63
1       Stephane Robert     Albano Olivetti      1       1     3     0     61
2    Jan-Lennard Struff     Guillaume Rufin      1       0     2     3     55
3          Aljaz Bedene     Dmitry Tursunov      1       0     1     3     52
4       Feliciano Lopez       Florent Serra      1       1     3     1     58
..                  …                 …    …      …     …   …    …
121      Novak Djokovic     Mikhail Youzhny      1       1     3     1     68
122         Andy Murray  Stanislas Wawrinka      1       0     0     3     63
123      Novak Djokovic  Stanislas Wawrinka      1       1     3     2     67
124     Richard Gasquet        Rafael Nadal      1       0     0     3     64
```

```
125      Novak Djokovic        Rafael Nadal      1       0     1     3     68

     FSW.1  SSP.1  SSW.1  …  BPC.2  BPW.2  NPA.2  NPW.2  TPW.2  ST1.2  \
0       45     37     16  …      1      3   30.0   40.0     83      3
1       44     39     19  …      0      1    0.0    0.0     71      3
2       61     45     32  …      5     15    0.0    0.0    149      7
3       41     48     19  …      6      9    0.0    0.0    121      7
4       54     42     30  …      0      3    0.0    0.0    123      7
..     …    …    …  …    …    …    …    …    …    …
121     49     32     19  …      2     10   10.0   21.0     87      3
122     37     37     22  …      4     11   31.0   42.0    107      6
123     64     33     25  …      5      9   26.0   41.0    165      6
124     41     36     15  …      4      4   22.0   28.0    102      6
125     40     32     16  …      7     12   17.0   23.0    121      6

     ST2.2  ST3.2  ST4.2  ST5.2
0        4    2.0    0.0    0.0
1        3    4.0    0.0    0.0
2        6    2.0    2.0    6.0
3        4    6.0    6.0    0.0
4        2    3.0    3.0    0.0
..     …    …    …    …
121      2    6.0    0.0    0.0
122      6    6.0    0.0    0.0
123      6    6.0    3.0    4.0
124      7    6.0    0.0    0.0
125      3    6.0    6.0    0.0

[126 rows x 42 columns]
```

**Can we again predict the Win Classification 0/1 by using 3 features?**

[ ]:
```
features = usM[["FSP.1", "FSW.1", "SSP.1",        "SSW.1", "ACE.1", "DBF.1",␣
 ↪"WNR.1", "UFE.1", "BPC.1", "BPW.1", "NPA.1", "TPW.1", "FSP.2", "FSW.2", "SSP.
 ↪2",        "SSW.2", "ACE.2", "DBF.2", "WNR.2", "UFE.2", "BPC.2", "BPW.2",␣
 ↪"NPA.2", "TPW.2"]].to_numpy()
target = usM["Result"].to_numpy()
scaler = StandardScaler()
X_scaled = scaler.fit_transform(features)
model = PCA(n_components = 3)
model.fit(X_scaled)
X_3D = model.transform(X_scaled)
df = pd.DataFrame({"Result" : target, "X" : X_3D[:,0], "Y" : X_3D[:,1], "Z" :␣
 ↪X_3D[:,2]})
df
```

[ ]:
```
     Result         X         Y         Z
0         1 -1.237340  1.244708 -1.446330
1         1 -1.884589 -0.552136 -2.267845
2         0  3.043515 -1.746605  0.094681
3         0  0.027489 -1.866698  1.224241
4         1  0.956425  0.981605 -1.298955
..      …       …       …       …
121       1 -1.327483  0.961892 -2.357617
122       0 -1.874227 -0.136433  0.912463
123       1  3.741905  0.508184 -2.115019
124       0 -2.204719  2.781073  1.526635
125       0 -1.026591  1.672345  0.495803

[126 rows x 4 columns]
```

**Perform PCA-3 and See the new 3D points**

[ ]:
```
fig = px.scatter_3d(df, x = "X", y = "Y", z = "Z", color = "Result")
fig.show()
fig.write_image("fig5_0.png", engine = "orca")
```

**Do the classification using the KMeans Model and see accuracy of prediction**

[ ]:
```
from sklearn.cluster import KMeans
km = KMeans(n_clusters = 2, init = "random", n_init = "auto", max_iter = 300)
kmeans = km.fit(X_3D)
labelK = kmeans.labels_
print(labelK)
print(accuracy_score(labelK, target))
```

```
[1 1 0 1 0 0 0 1 0 1 1 1 0 0 0 1 0 0 1 1 0 1 0 1 1 1 1 0 1 1 0 0 1 1 1 1 1
 1 0 0 0 1 1 1 1 0 0 1 0 1 1 1 1 1 0 0 1 0 0 0 1 0 1 0 1 1 0 1 1 1 1 1 1 0
 1 0 1 1 1 1 1 0 0 0 1 1 0 0 1 0 0 1 1 1 1 0 1 0 0 0 1 0 1 1 1 0 1 1 1 0 0
 1 0 0 1 1 0 1 1 0 1 1 1 0 1 1]
0.5555555555555556
```

**Plot the Clusters with their centers**

[ ]:
```
import plotly.graph_objs as go
center = kmeans.cluster_centers_
fig = go.Figure()
fig.add_trace(go.Scatter3d(x = X_3D[:,0], y = X_3D[:,1], z = X_3D[:,2], mode =␣
 ↪"markers", marker = dict(size = 6, color = labelK, opacity = 0.8),␣
 ↪showlegend = True, name = "Cluster Points"))
fig.add_trace(go.Scatter3d(x = center[:,0], y = center[:,1], z = center[:,2],␣
 ↪mode = "markers", marker = dict(size = 15, opacity = 1, symbol = "cross"),␣
 ↪showlegend = True, name = "Centers"))
fig.show()
fig.write_image("fig5_1.png", engine = "orca")
```

### 0.0.8 US Open Women's Tournament

[ ]:
```
usW
```

[ ]:
```
        Player 1            Player 2  ROUND  Result  FNL.1  FNL.2  FSP.1  \
0     S Williams          V Azarenka      7       1      2      1     57
1     F Pennetta          V Azarenka      6       0      0      2     44
2     S Williams               N Li       6       1      2      0     63
3        R Vinci          F Pennetta      5       0      0      2     60
4   D Hantuchova          V Azarenka      5       0      0      2     58
..           …                   …    …      …    …    …    …
71   P Ormaechea        K Date-Krumm      1       1      2      0     59
72    K Pliskova          E Bouchard      1       0      1      2     53
73    L Hradecka            A Kerber      1       0      0      2     49
74       L Davis    C Suarez Navarro      1       0      0      2     63
75  K Mladenovic  A Medina Garrigues      1       1      2      1     51

    FSW.1  SSP.1  SSW.1  …  BPC.2  BPW.2  NPA.2  NPW.2  TPW.2  ST2.1.1  \
0      44     43     20  …      8      4   15.0   10.0    0.0        5
1      12     56      7  …     13      8   30.0   20.0    0.0        6
2      26     37      9  …      4      1   19.0   13.0    0.0        0
3      21     40      7  …     12      6   14.0    7.0    0.0        6
4      14     42      5  …     11      7   13.0   12.0    0.0        6
..    …    …    …  …    …    …    …    …    …      …
71     32     41     10  …      9      4   14.0    9.0    0.0        3
72     48     47     21  …     13      3   13.0   10.0    0.0        4
73     17     51      4  …      8      5    6.0    5.0    0.0        6
74     12     37      3  …     12      6   10.0    8.0    0.0        6
75     29     49     15  …     13      4   14.0    7.0    0.0        1

    ST2.2  ST3.2  ST4.2  ST5.2
0       7    1.0    0.0    0.0
1       6    0.0    0.0    0.0
2       3    0.0    0.0    0.0
3       6    0.0    0.0    0.0
4       6    0.0    0.0    0.0
..    …    …    …    …
71      6    0.0    0.0    0.0
72      6    7.0    0.0    0.0
73      6    0.0    0.0    0.0
74      6    0.0    0.0    0.0
75      6    1.0    0.0    0.0

[76 rows x 42 columns]
```

**Serena Williams vs Rest Winners in terms of Breaking Points Won**

[ ]:
```
df = usW[usW["Player 1"] == "S Williams"]
df
```

[ ]:

| | Player 1 | Player 2 | ROUND | Result | FNL.1 | FNL.2 | FSP.1 | FSW.1 | \ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | S Williams | V Azarenka | 7 | 1 | 2 | 1 | 57 | 44 | |
| 2 | S Williams | N Li | 6 | 1 | 2 | 0 | 63 | 26 | |
| 10 | S Williams | S Stephens | 4 | 1 | 2 | 0 | 62 | 26 | |
| 23 | S Williams | Y Shvedova | 3 | 1 | 2 | 0 | 66 | 28 | |
| 30 | S Williams | G Voskoboeva | 2 | 1 | 2 | 0 | 64 | 21 | |
| 64 | S Williams | F Schiavone | 1 | 1 | 2 | 0 | 51 | 13 | |

| | SSP.1 | SSW.1 | … | BPC.2 | BPW.2 | NPA.2 | NPW.2 | TPW.2 | ST2.1.1 | ST2.2 | \ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 43 | 20 | … | 8 | 4 | 15.0 | 10.0 | 0.0 | 5 | 7 | |
| 2 | 37 | 9 | … | 4 | 1 | 19.0 | 13.0 | 0.0 | 0 | 3 | |
| 10 | 38 | 11 | … | 2 | 1 | 12.0 | 6.0 | 0.0 | 4 | 1 | |
| 23 | 44 | 7 | … | 1 | 0 | 13.0 | 4.0 | 0.0 | 3 | 1 | |
| 30 | 36 | 11 | … | 1 | 0 | 10.0 | 5.0 | 0.0 | 3 | 0 | |
| 64 | 49 | 12 | … | 0 | 0 | 6.0 | 3.0 | 0.0 | 0 | 1 | |

| | ST3.2 | ST4.2 | ST5.2 |
|---|---|---|---|
| 0 | 1.0 | 0.0 | 0.0 |
| 2 | 0.0 | 0.0 | 0.0 |
| 10 | 0.0 | 0.0 | 0.0 |
| 23 | 0.0 | 0.0 | 0.0 |
| 30 | 0.0 | 0.0 | 0.0 |
| 64 | 0.0 | 0.0 | 0.0 |

[6 rows x 42 columns]

### Breaking Points Won by Serena Williams

[ ]:
```
print(df["BPW.1"].mean())
```

5.166666666666667

[ ]:
```
fig = px.bar(df, x = "ROUND", y = "BPW.1", hover_data = ["BPW.1"], color = "BPW.
↪1", labels={"BPW.1" : "Break Points Won"}, height=400)
fig.show()
fig.write_image("fig6_0.png", engine = "orca")
```

### Breaking Points Won by the Respective Winners of each match

[ ]:
```
print(usW[usW["Result"] == 1]["BPW.1"].mean())
```

5.027777777777778

[ ]:

```
fig = px.bar(usW[usW["Result"] == 1], x = "ROUND", y = "BPW.1", hover_data =␣
↪["BPW.1"], color = "BPW.1", labels={"BPW.1" : "Break Points Won"},␣
↪height=400)
fig.show()
fig.write_image("fig6_1.png", engine = "orca")
```

### 0.0.9 Wimbledon Men's Tournament

[ ]:
```
wimM
```

[ ]:
```
        Player1          Player2  Round  Result  FNL.1  FNL.2  FSP.1  FSW.1  \
0      B.Becker         A.Murray      1       0      0      3     59     29
1        J.Ward           Y-H.Lu      1       0      1      3     62     77
2       N.Mahut          J.Hajek      1       1      3      0     72     44
3     T.Robredo  A.Bogomolov Jr.      1       1      3      0     77     40
4       R.Haase        M.Youzhny      1       0      0      3     68     61
..          …              …    …     …    …    …    …    …
109    D.Ferrer      J.Del Potro      5       0      0      3     68     45
110  N.Djokovic        T.Berdych      5       1      3      0     61     42
111  J.Janowicz         A.Murray      6       0      1      3     55     54
112  N.Djokovic      J.Del Potro      6       1      3      2     69    102
113  N.Djokovic         A.Murray      7       0      0      3     65     40

     SSP.1  SSW.1  …  BPC.2  BPW.2  NPA.2  NPW.2  TPW.2  ST1.2  ST2.2  \
0       41     14  …     10      5     23     17    0.0      6      6
1       38     35  …     15      2     46     39    0.0      6      6
2       28     10  …      1      0     19     12    0.0      2      4
3       23     12  …      0      0     22     13    0.0      2      2
4       32     15  …     21      3     44     30    0.0      6      7
..     …    …  …    …    …    …    …    …    …    …
109     32     17  …      8      3     21     17    0.0      6      6
110     39     21  …      2      2     31     21    0.0      6      4
111     45     27  …     13      5     36     22    0.0      6      6
112     31     21  …      7      2     37     25    0.0      5      6
113     35     15  …     17      7     37     26    0.0      6      7

     ST3.2  ST4.2  ST5.2
0        6    0.0    0.0
1        7    7.0    0.0
2        3    0.0    0.0
3        4    0.0    0.0
4        7    0.0    0.0
..     …    …    …
109      7    0.0    0.0
110      3    0.0    0.0
111      6    6.0    0.0
112      6    7.0    3.0
```

```
113      6    0.0    0.0

[114 rows x 42 columns]
```

### Distinct players in the tournament

[ ]:
```
players = set(wimM["Player1"].unique()) | set(wimM["Player2"].unique())
print(players)
```

```
{'B.Paire', 'J-W.Tsonga', 'S.Stakhovsky', 'T.Haas', 'M.Baghdatis', 'A.Haider-
Maurer', 'J.Nieminen', 'D.Istomin', 'A.Murray', 'O.Rochus', 'E.Gulbis',
'B.Tomic', 'V.Hanescu', 'I.Dodig', 'A.Montanes', 'J.Chardy', 'D.Ferrer',
'K.Anderson', 'A.Bedene', 'A.Mannarino', 'R.Stepanek', 'G.Zemlja', 'A.Seppi',
'M.Przysiezny', 'M.Matosevic', 'L.Kubot', 'S.Bolelli', 'J.Struff', 'G.Simon',
'W.Odesnik', 'P.Kohlschreiber', 'S.Wawrinka', 'M.Alund', 'J.Monaco', 'J.Levine',
'M.Klizan', 'J.Ward', 'G.Garcia-Lopez', 'T.Berdych', 'R.Gasquet', 'T.Kamke',
'B.Reynolds', 'V.Pospisil', 'A.Ungur', 'M.Raonic', 'R.Bautista Agut',
'H.Zeballos', 'J.Duckworth', 'P.Petzschner', 'M.Reid', 'A.Kuznetsov',
'D.Tursunov', 'G.Rufin', 'R.Dutra Silva', 'R.Nadal', 'L.Lacko', 'P.Lorenzi',
'L.Hewitt', 'S.Darcis', 'K.Edmund', 'R.Ram', 'E.Donskoy', 'M.Granollers',
'J.Janowicz', 'G.Elias', 'D.Brown', 'A.Falla', 'B.Becker', 'I.Andreev',
'V.Troicki', 'A.Ramos', 'R.Haase', 'R.Federer', 'S.Giraldo', 'D.Goffin',
'D.Kudla', 'B.Kavcic', 'M.Youzhny', 'B.Knittel', 'C.Berlocq', 'M.Ebden',
'G.Pella', 'X.Malisse', 'R.Harrison', 'M.Russell', 'J.Del Potro',
'A.Dolgopolov', 'T.Robredo', 'P-H.Mathieu', 'D.Brands', 'G.Dimitrov', 'L.Mayer',
'K.De Schepper', 'N.Djokovic', 'L.Rosol', 'J.Benneteau', 'S.Johnson', 'J.Isner',
'I.Sijsling', 'F.Lopez', 'N.Mahut', 'N.Almagro', 'S.Robert', 'E.Roger-Vasselin',
'J.Hajek', 'Y-T.Wang', 'T.De Bakker', 'P.Andujar', 'A.Bogomolov Jr.', 'Y-H.Lu',
'J.Reister', 'J.Zopp', 'S.Querrey', 'R.Berankis', 'T.Gabashvili', 'F.Verdasco',
'K.Nishikori', 'D.Gimeno-Traver', 'G.Soeda', 'F.Fognini', 'M.Cilic',
'M.Gicquel', 'F.Mayer', 'M.Llodra', 'J.Tipsarevic', 'J.Blake', 'J.Melzer'}
```

### List of Matches won by each player during the wimbledon

[ ]:
```
dfnew = pd.DataFrame({"Player" : list(players), "Wins" : 0})
for _, row in wimM.iterrows():
    if row["Result"] == 1:
        dfnew.loc[dfnew["Player"] == row["Player1"], "Wins"] += 1
    else:
        dfnew.loc[dfnew["Player"] == row["Player2"], "Wins"] += 1
dfnew = dfnew.sort_values("Wins", ascending = False)
dfnew.reset_index(inplace = True)
dfnew.drop("index", axis = 1, inplace = True)
dfnew[:20]
```

[ ]:
```
          Player  Wins
0       A.Murray     7
1     N.Djokovic     6
```

```
2    J.Del Potro     5
3     J.Janowicz     4
4      T.Berdych     4
5        B.Tomic     3
6     F.Verdasco     3
7       D.Ferrer     3
8        I.Dodig     3
9       J.Melzer     3
10      E.Gulbis     2
11       L.Kubot     2
12   K.Nishikori     2
13  S.Stakhovsky     2
14     N.Almagro     2
15    I.Sijsling     2
16        T.Haas     2
17     T.Robredo     2
18     M.Youzhny     2
19     V.Troicki     2
```

### 0.0.10 Wimbledon Women's Tournament

[ ]:
```
wimW
```

[ ]:
```
           Player1      Player2  Round  Result  FNL.1  FNL.2  FSP.1  FSW.1  \
0       M.Koehler   V.Azarenka      1       0      0      2     60     21
1      E.Baltacha   F.Pennetta      1       0      0      2     69     23
2       S-W.Hsieh      T.Maria      1       1      2      0     63     17
3        A.Cornet       V.King      1       1      2      1     57     36
4    Y.Putintseva   K.Flipkens      1       0      0      2     73     34
..            …           …    …      …    …    …    …    …
117   A.Radwanska         N.Li      5       1      2      1     77     52
118     S.Lisicki     K.Kanepi      5       1      2      0     59     26
119     M.Bartoli   K.Flipkens      6       1      2      0     61     21
120     S.Lisicki  A.Radwanska      6       1      2      1     63     53
121     S.Lisicki    M.Bartoli      7       0      0      2     65     22

     SSP.1  SSW.1  …  BPC.2  BPW.2  NPA.2  NPW.2  TPW.2  ST1.1.1  ST2.2  \
0       40      8  …     16      6      8      4    0.0        6      6
1       31      6  …      6      5     14     11    0.0        6      6
2       37     10  …      1      0      8      2    0.0        1      0
3       43     21  …      4      1     48     32    0.0        6      3
4       27     12  …      9      3     35     24    0.0        7      6
..     …    …  …    …    …    …    …    …    …        …      …
117     23      9  …     10      4     71     48    0.0        6      6
118     41     10  …      2      1     19      9    0.0        3      3
119     39     10  …      2      1     21      8    0.0        1      2
120     37     19  …     14      6     31     16    0.0        4      6
```

```
121     35      9  …     13      5     11      9    0.0        6      6

     ST3.2  ST4.2  ST5.2
0      0.0    0.0    0.0
1      0.0    0.0    0.0
2      0.0    0.0    0.0
3      1.0    0.0    0.0
4      0.0    0.0    0.0
..     …    …    …
117    2.0    0.0    0.0
118    0.0    0.0    0.0
119    0.0    0.0    0.0
120    7.0    0.0    0.0
121    0.0    0.0    0.0

[122 rows x 42 columns]
```

[ ]:
```
players = set(wimW["Player1"].unique()) | set(wimW["Player2"].unique())
dfnew = pd.DataFrame({"Player" : list(players), "Wins" : 0})
for _, row in wimW.iterrows():
    if row["Result"] == 1:
        dfnew.loc[dfnew["Player"] == row["Player1"], "Wins"] += 1
    else:
        dfnew.loc[dfnew["Player"] == row["Player2"], "Wins"] += 1
dfnew = dfnew.sort_values("Wins", ascending = False)
dfnew.reset_index(inplace = True)
dfnew.drop("index", axis = 1, inplace = True)
display(dfnew[:20])
newdf = pd.DataFrame({"Player" : [dfnew["Player"][0]]*2})
display(newdf)
```

```
                Player  Wins
0            M.Bartoli     7
1            S.Lisicki     6
2          A.Radwanska     5
3           K.Flipkens     5
4                 N.Li     4
5              R.Vinci     3
6          E.Birnerova     3
7           S.Stephens     3
8           E.Makarova     3
9          T.Pironkova     3
10            K.Kanepi     3
11             K.Knapp     3
12          S.Williams     3
13         P.Cetkovska     3
14   C.Suarez Navarro     3
```

```
15        K.Date-Krumm     2
16  M.Larcher De Brito     2
17             A.Cornet     2
18             C.Giorgi     2
19         D.Cibulkova      2
```

```
      Player
0  M.Bartoli
1  M.Bartoli
```

[ ]:
```
FNL, FSP, FSW, SSP, SSW, ACE, DBF, WNR, UFE, BPC, BPW, NPA, NPW = [], [], [],␣
↪[], [], [], [], [], [], [], [], [], []
for i in range(len(wimW.index)):
  row = wimW.iloc[i]
  if ((newdf["Player"][0] == row["Player1"]) and (row["Result"] == 1)):
    FNL.append(row["FNL.1"])
    FSP.append(row["FSP.1"])
    FSW.append(row["FSW.1"])
    SSP.append(row["SSP.1"])
    SSW.append(row["SSW.1"])
    ACE.append(row["ACE.1"])
    DBF.append(row["DBF.1"])
    WNR.append(row["WNR.1"])
    UFE.append(row["UFE.1"])
    BPC.append(row["BPC.1"])
    BPW.append(row["BPW.1"])
    NPA.append(row["NPA.1"])
    NPW.append(row["NPW.1"])

  if ((newdf["Player"][0] == row["Player2"]) and (row["Result"] == 0)):
    FNL.append(row["FNL.2"])
    FSP.append(row["FSP.2"])
    FSW.append(row["FSW.2"])
    SSP.append(row["SSP.2"])
    SSW.append(row["SSW.2"])
    ACE.append(row["ACE.2"])
    DBF.append(row["DBF.2"])
    WNR.append(row["WNR.2"])
    UFE.append(row["UFE.2"])
    BPC.append(row["BPC.2"])
    BPW.append(row["BPW.2"])
    NPA.append(row["NPA.2"])
    NPW.append(row["NPW.2"])
```

[ ]:
```
print(FNL, FSP, FSW, SSP, SSW, ACE, DBF, WNR, UFE, BPC, BPW, NPA, NPW)
```

```
[2, 2, 2, 2, 2, 2, 2] [58, 64, 62, 61, 58, 61, 67] [29, 38, 26, 26, 26, 21, 31]
[42, 36, 38, 39, 42, 39, 33] [16, 11, 10, 7, 15, 10, 7] [4.0, 1.0, 1.0, 1.0,
```

```
0.0, 5.0, 2.0] [3, 6, 5, 7, 3, 3, 6] [30, 22, 20, 12, 6, 23, 15] [25, 17, 12,
13, 12, 10, 14] [7, 12, 8, 7, 12, 7, 13] [3, 5, 6, 5, 6, 5, 5] [18, 16, 8, 2,
10, 11, 11] [13, 12, 7, 2, 4, 11, 9]
```

[ ]:
```
newdf["FNL"] = [sum(FNL[:-1])/6, FNL[-1]]
newdf["FSP"] = [sum(FSP[:-1])/6, FSP[-1]]
newdf["FSW"] = [sum(FSW[:-1])/6, FSW[-1]]
newdf["SSP"] = [sum(SSP[:-1])/6, SSP[-1]]
newdf["SSW"] = [sum(SSW[:-1])/6, SSW[-1]]
newdf["ACE"] = [sum(ACE[:-1])/6, ACE[-1]]
newdf["DBF"] = [sum(DBF[:-1])/6, DBF[-1]]
newdf["WNR"] = [sum(WNR[:-1])/6, WNR[-1]]
newdf["UFE"] = [sum(UFE[:-1])/6, UFE[-1]]
newdf["BPC"] = [sum(BPC[:-1])/6, BPC[-1]]
newdf["BPW"] = [sum(BPW[:-1])/6, BPW[-1]]
newdf["NPA"] = [sum(NPA[:-1])/6, NPA[-1]]
newdf["NPW"] = [sum(NPW[:-1])/6, NPW[-1]]
newdf
```

[ ]:

| | Player | FNL | FSP | FSW | SSP | SSW | ACE | DBF | WNR | \ |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | M.Bartoli | 2.0 | 60.666667 | 27.666667 | 39.333333 | 11.5 | 2.0 | 4.5 | 18.833333 | |
| 1 | M.Bartoli | 2.0 | 67.000000 | 31.000000 | 33.000000 | 7.0 | 2.0 | 6.0 | 15.000000 | |

| | UFE | BPC | BPW | NPA | NPW |
|---|---|---|---|---|---|
| 0 | 14.833333 | 8.833333 | 5.0 | 10.833333 | 8.166667 |
| 1 | 14.000000 | 13.000000 | 5.0 | 11.000000 | 9.000000 |

[ ]:
```
d = list(newdf.columns[1:])
d1 = newdf.iloc[0].values[1:]
d2 = newdf.iloc[1].values[1:]
```

**How does the performance of the Winner vary when comapared to Final and Previous Rounds**

[ ]:
```
fig = go.Figure()
fig.add_trace(go.Scatter(x = d, y = d1, name = "Avg Stats for Round 1 - 6"))
fig.add_trace(go.Scatter(x = d, y = d2, name = "Stats for Round 7"))
fig.show()
fig.write_image("fig7_0.png", engine = "orca")
```