CS 241

# 12 Prove Milestone - Data Analysis

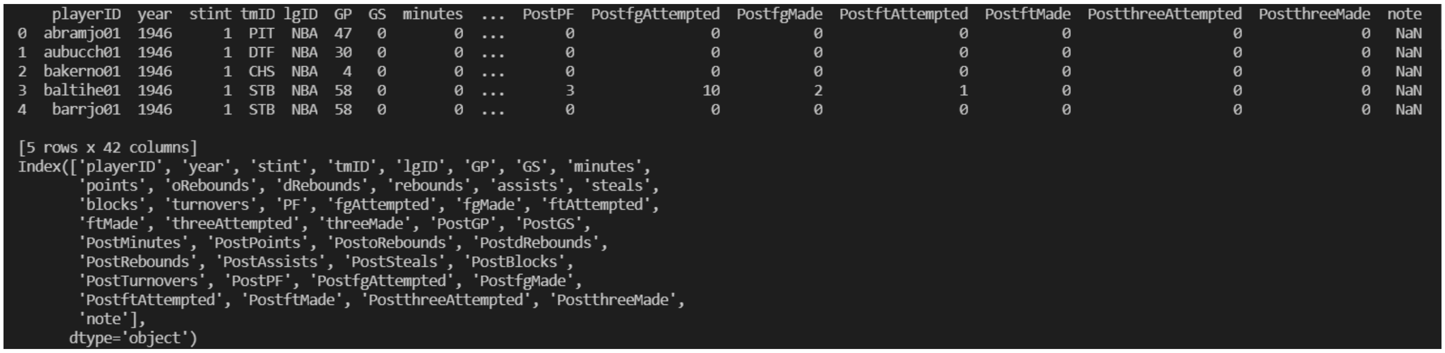
## PART I

*Jose Israel Carmona Morales*

To initialize the libraries and load the data:

```
import pandas as pd #The library that manipulates our data
import seaborn as sns #Used to plotting and graphing
import matplotlib.pyplot as plt #If we need any low level methods

bestb_players = pd.read_csv("basketball_players.csv")
print(bestb_players.head())
print(bestb_players.columns)
```

## OUTPUT



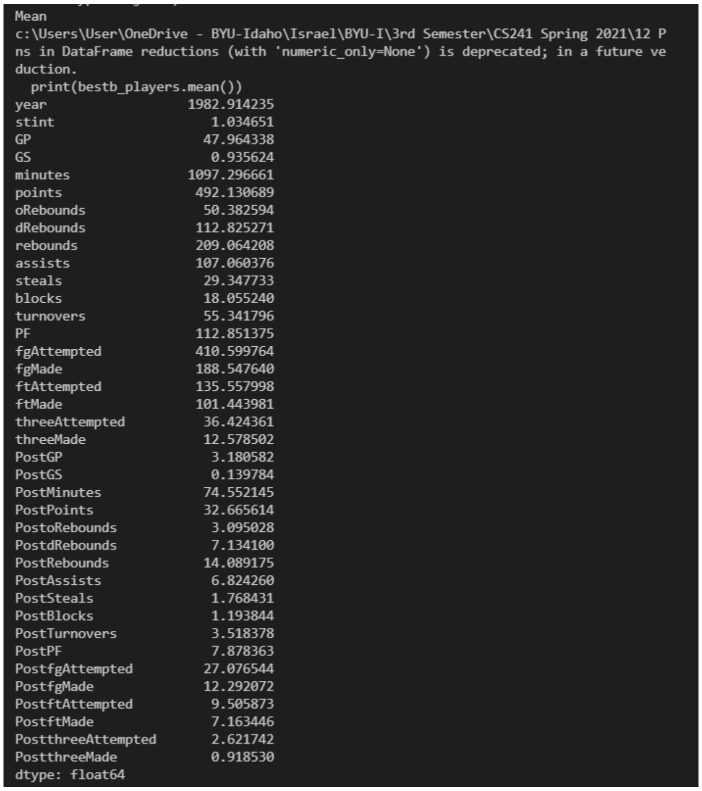
```
    playerID  year  stint tmID lgID  GP  GS  minutes  ...  PostPF  PostfgAttempted  PostfgMade  PostftAttempted  PostftMade  PostthreeAttempted  PostthreeMade  note
0  abramjo01  1946      1  PIT  NBA  47   0        0  ...       0                0           0                0           0                   0              0   NaN
1  aubucch01  1946      1  DTF  NBA  30   0        0  ...       0                0           0                0           0                   0              0   NaN
2  bakerno01  1946      1  CHS  NBA   4   0        0  ...       0                0           0                0           0                   0              0   NaN
3  baltihe01  1946      1  STB  NBA  58   0        0  ...       3               10           2                1           0                   0              0   NaN
4   barrjo01  1946      1  STB  NBA  58   0        0  ...       0                0           0                0           0                   0              0   NaN

[5 rows x 42 columns]
Index(['playerID', 'year', 'stint', 'tmID', 'lgID', 'GP', 'GS', 'minutes',
       'points', 'oRebounds', 'dRebounds', 'rebounds', 'assists', 'steals',
       'blocks', 'turnovers', 'PF', 'fgAttempted', 'fgMade', 'ftAttempted',
       'ftMade', 'threeAttempted', 'threeMade', 'PostGP', 'PostGS',
       'PostMinutes', 'PostPoints', 'PostoRebounds', 'PostdRebounds',
       'PostRebounds', 'PostAssists', 'PostSteals', 'PostBlocks',
       'PostTurnovers', 'PostPF', 'PostfgAttempted', 'PostfgMade',
       'PostftAttempted', 'PostftMade', 'PostthreeAttempted', 'PostthreeMade',
       'note'],
      dtype='object')
```

# REQUIREMENT 01.

It finds the mean and the median of numbers of points scored.

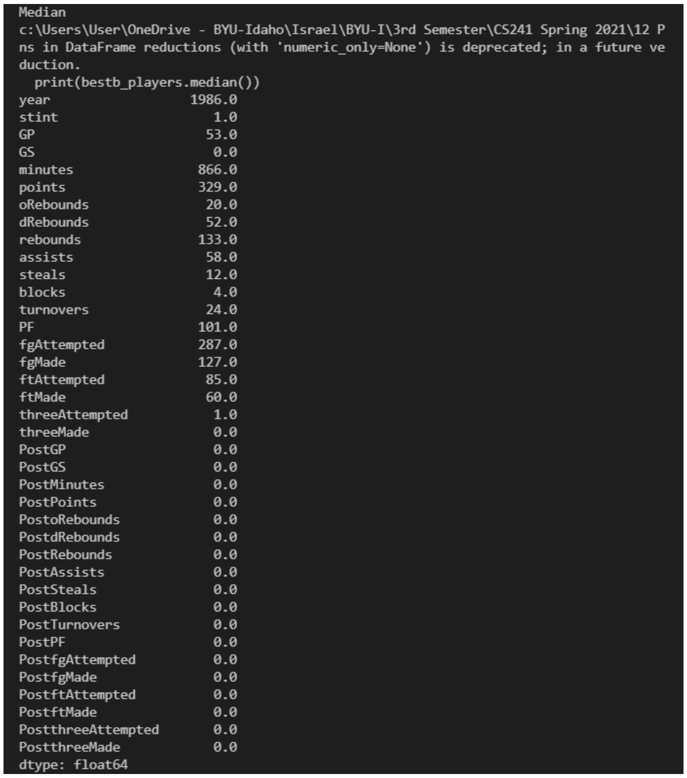
```
print("Mean")
print(bestb_players.mean())
print("Median")
print(bestb_players.median())
```

## MEAN



```
Mean
c:\Users\User\OneDrive - BYU-Idaho\Israel\BYU-I\3rd Semester\CS241 Spring 2021\12 P
ns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future ve
duction.
  print(bestb_players.mean())
year                  1982.914235
stint                    1.034651
GP                      47.964338
GS                       0.935624
minutes               1097.296661
points                 492.130689
oRebounds               50.382594
dRebounds              112.825271
rebounds               209.064208
assists                107.060376
steals                  29.347733
blocks                  18.055240
turnovers               55.341796
PF                     112.851375
fgAttempted            410.599764
fgMade                 188.547640
ftAttempted            135.557998
ftMade                 101.443981
threeAttempted          36.424361
threeMade               12.578502
PostGP                   3.180582
PostGS                   0.139784
PostMinutes             74.552145
PostPoints              32.665614
PostoRebounds            3.095028
PostdRebounds            7.134100
PostRebounds            14.089175
PostAssists              6.824260
PostSteals               1.768431
PostBlocks               1.193844
PostTurnovers            3.518378
PostPF                   7.878363
PostfgAttempted         27.076544
PostfgMade              12.292072
PostftAttempted          9.505873
PostftMade               7.163446
PostthreeAttempted       2.621742
PostthreeMade            0.918530
dtype: float64
```

## MEDIAN



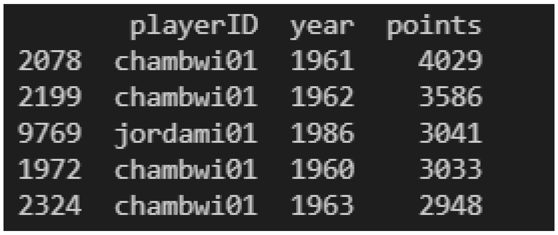
```
Median
c:\Users\User\OneDrive - BYU-Idaho\Israel\BYU-I\3rd Semester\CS241 Spring 2021\12 P
ns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future ve
duction.
  print(bestb_players.median())
year                  1986.0
stint                    1.0
GP                      53.0
GS                       0.0
minutes                866.0
points                 329.0
oRebounds               20.0
dRebounds               52.0
rebounds               133.0
assists                 58.0
steals                  12.0
blocks                   4.0
turnovers               24.0
PF                     101.0
fgAttempted            287.0
fgMade                 127.0
ftAttempted             85.0
ftMade                  60.0
threeAttempted           1.0
threeMade                0.0
PostGP                   0.0
PostGS                   0.0
PostMinutes              0.0
PostPoints               0.0
PostoRebounds            0.0
PostdRebounds            0.0
PostRebounds             0.0
PostAssists              0.0
PostSteals               0.0
PostBlocks               0.0
PostTurnovers            0.0
PostPF                   0.0
PostfgAttempted          0.0
PostfgMade               0.0
PostftAttempted          0.0
PostftMade               0.0
PostthreeAttempted       0.0
PostthreeMade            0.0
dtype: float64
```

# REQUIREMENT 02.

It finds the highest number of points per season, sorted the data by points, year and shows the highest 5.

```
print(bestb_players[["playerID", "year", "points"]].sort_values("points", ascending = False).head(5))
```

## OUTPUT



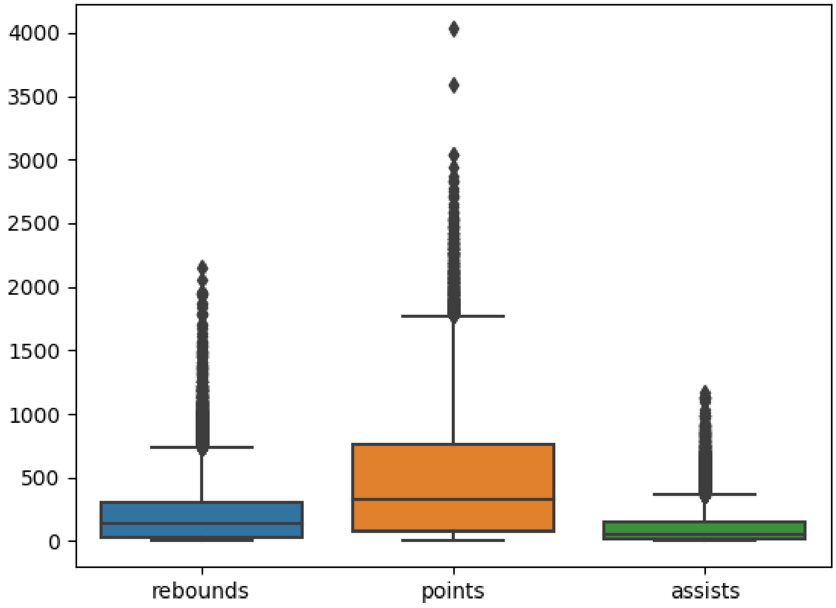
```
        playerID  year  points
2078  chambwi01  1961    4029
2199  chambwi01  1962    3586
9769  jordami01  1986    3041
1972  chambwi01  1960    3033
2324  chambwi01  1963    2948
```

# REQUIREMENT 03.

Produces a boxplot that shows the distribution of total points, total assists, and total rebounds.

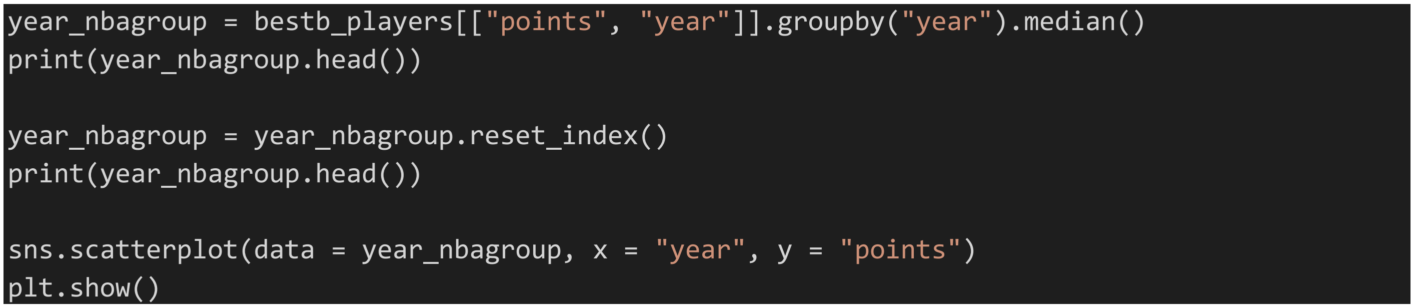
```
sns.boxplot(data = bestb_players[["rebounds", "points", "assists"]])
plt.show()
```

## OUTPUT



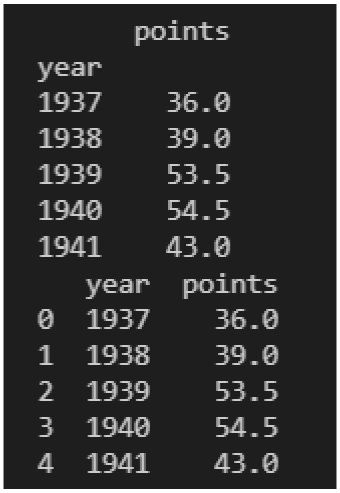
# REQUIREMENT 04.

Produces a boxplot that shows the distribution of total points, total assists, and total rebounds.

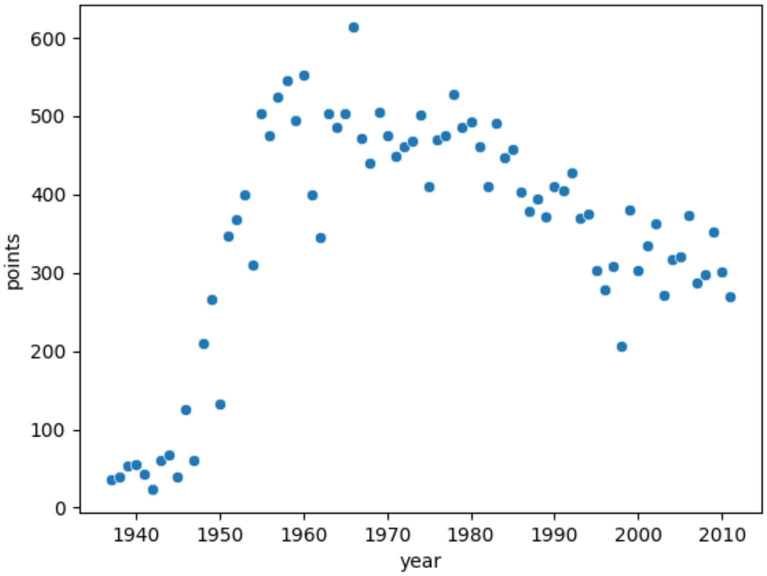


```
year_nbagroup = bestb_players[["points", "year"]].groupby("year").median()
print(year_nbagroup.head())

year_nbagroup = year_nbagroup.reset_index()
print(year_nbagroup.head())

sns.scatterplot(data = year_nbagroup, x = "year", y = "points")
plt.show()
```

## OUTPUT



```
      points
year
1937    36.0
1938    39.0
1939    53.5
1940    54.5
1941    43.0
   year  points
0  1937    36.0
1  1938    39.0
2  1939    53.5
3  1940    54.5
4  1941    43.0
```


points
year