Sunday Onwuchekwa

Curtis Mellor

CS 241 – 05

7 April 2020

Prove: Data Analysis

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import os

# Change directory
os.chdir("/Users/osunday/Documents/School/BYUI/Winter 2020/CS 241 Survey Object Oriented
Programming and Data Structure/Week 13/data-analysis/nba_basketball_data/")

# Load CSV files
players = pd.read_csv("basketball_players.csv")
master = pd.read_csv("basketball_master.csv")

# Merge both files
merged_data = pd.merge(players, master, how="left", left_on="playerID", right_on="bioID")
```

PART I - SPECIFIC ASSIGNMENTS

1. Calculate the mean and median number of points scored. (In other words, each row is the amount of points a player scored during a particular season. Calculate the median of these values. The result of this is that we have the median number of points players score each season.)

```
mean = merged_data["points"].mean()
median = merged_data["points"].median()

print("The mean of scored points is {:.2f}\n".format(mean))
print("The median of scored points is {:.2f}".format(median))
```

```
a. The mean of scored points is 492.13

b. The median of scored points is 329.00
```

2. Determine the highest number of points recorded in a single season. Identify who scored those points and the year they did so.

```
highest_point = merged_data["points"].max()

highest_pointer = merged_data[merged_data["points"] == highest_point][["firstName",
"middleName", "lastName", "year", "points"]]

print("The highest number of points in a single season is {}\n".format(highest_point))
print("The details of the Highest pointer are as follows:\n\n{}".format(highest_pointer))
```

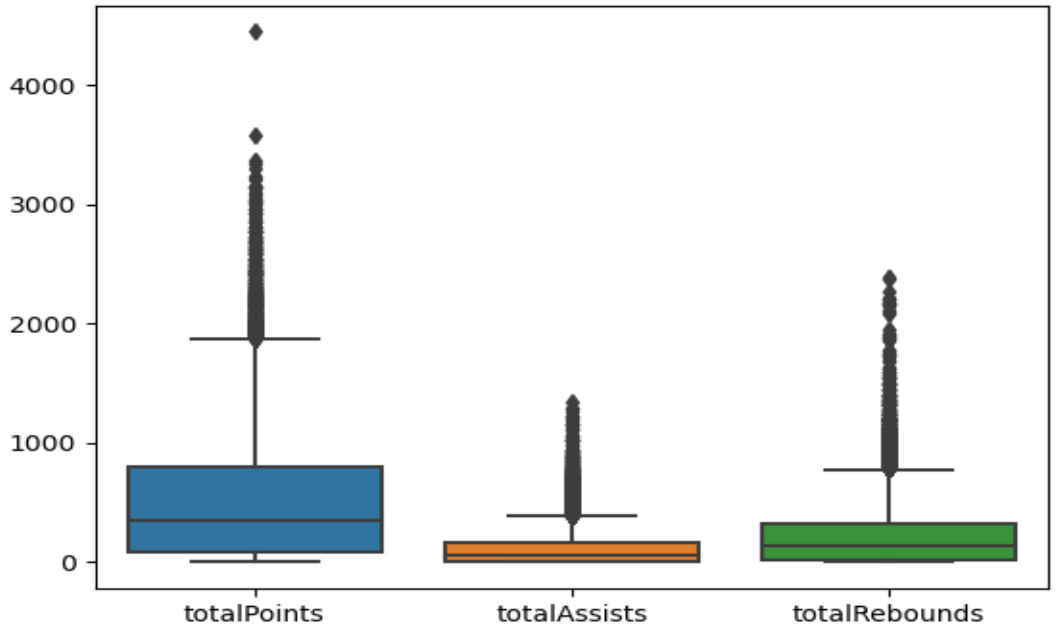
a. The highest number of points in a single season is 4029

b. The details of the Highest pointer are as follows:

	firstName	middleName	lastName	year	points
2078	Wilton	Norman	Chamberlain	1961	4029

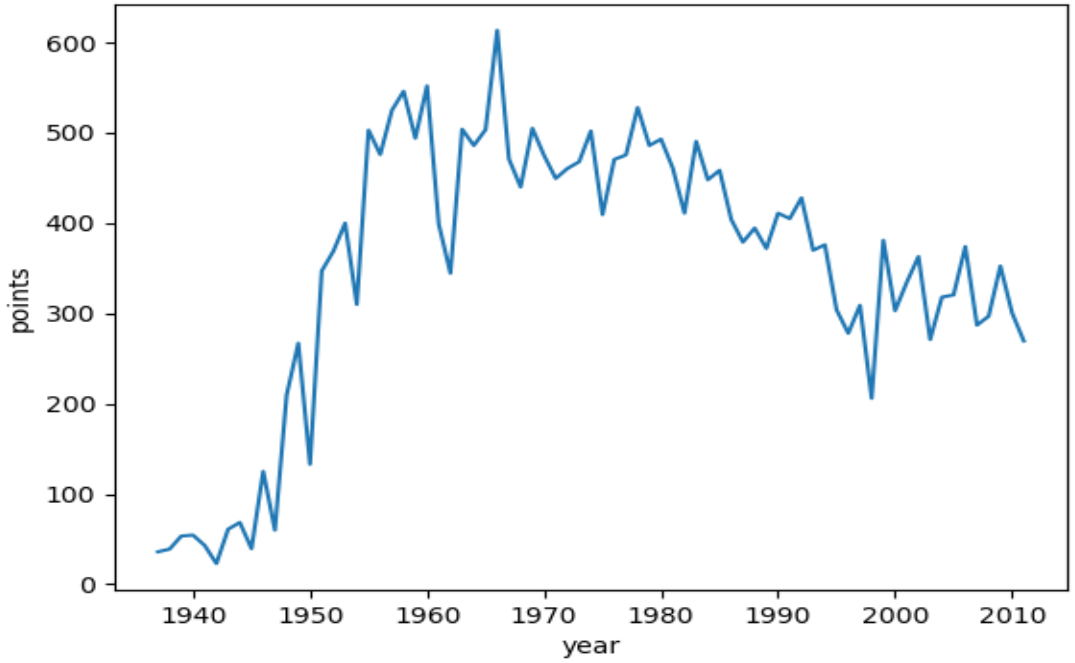
3. Produce a boxplot that shows the distribution of total points, total assists, and total rebounds (each of these three is a separate box plot, but they can be on the same scale and in the same graphic).

```
merged_data["totalPoints"] = merged_data["points"] + merged_data["PostPoints"]
merged_data["totalAssists"] = merged_data["assists"] + merged_data["PostAssists"]
merged_data["totalRebounds"] = merged_data["rebounds"] + merged_data["PostRebounds"]

sns.boxplot(data=merged_data[["totalPoints", "totalAssists", "totalRebounds"]])
plt.show()
```



4. Produce a plot that shows how the number of points scored has changed over time by showing the median of points scored per year, over time. The x-axis is the year and the y-axis is the median number of points among all players for that year.

```
pointsByYear = merged_data[["year", "points"]].groupby("year").median()
pointsByYear = pointsByYear.reset_index()
pointsByYear = pointsByYear[pointsByYear["points"] > 0]

sns.lineplot(data=pointsByYear, x="year", y="points")
plt.show()
```



PART II - COME UP WITH SUPPORTING EVIDENCE

1. Some players score a lot of points because they attempt a lot of shots. Among players that have scored a lot of points, are there some that are much more efficient (points per attempt) than others?

```
merged_data["points_per_attempt"] = merged_data["points"] / (merged_data["fgAttempted"]
+ merged_data["ftAttempted"])

player_attempted_points = merged_data[["playerID",
"points_per_attempt"]].groupby("playerID").mean()

player_attempted_points =
player_attempted_points[player_attempted_points["points_per_attempt"] != np.inf]
```

```
top5_attempted_points =
player_attempted_points.nlargest(5,"points_per_attempt")["points_per_attempt"]

top5_attempted_points_merged = pd.merge(top5_attempted_points, master, how = "left",
left_on = "playerID", right_on = "bioID")[["firstName", "lastName","points_per_attempt"]]

print("Below are top five players who score many points because they attempt many
shots:\n\n{}".format(top5 attempted points merged))
```

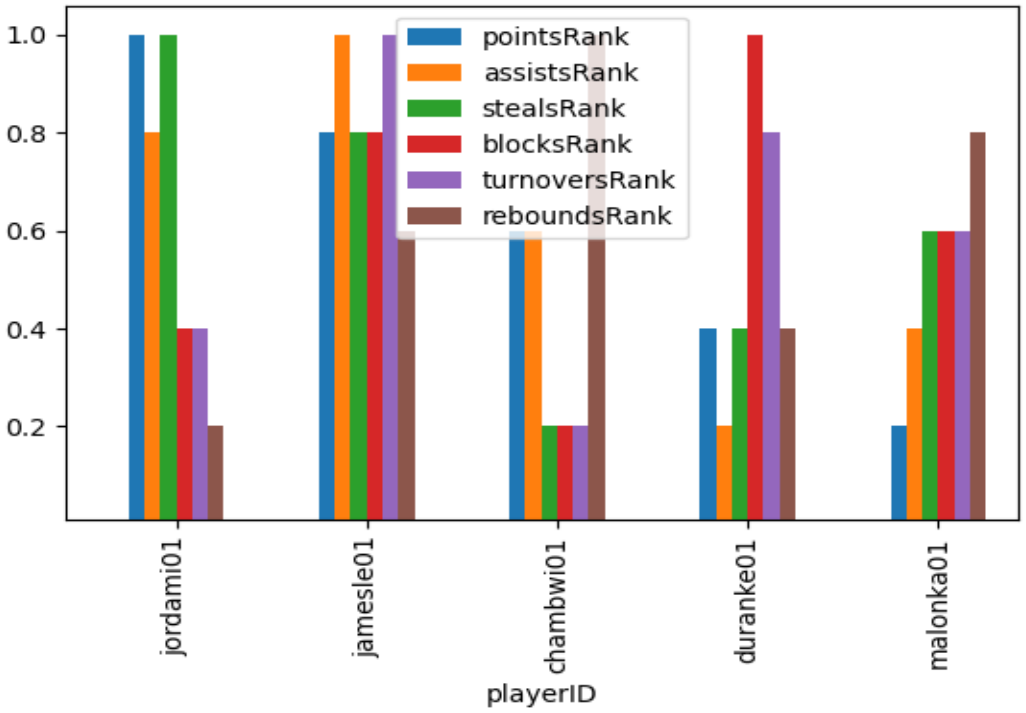
```
Below are top five players who score many points because they attempt many

shots:

       firstName     lastName     points_per_attempt

0       Norm          Rosen        11.222222

1       Robert        Skarda       10.000000

2       Harry         Johnson      7.666667

3       Harold        Lambert      7.666667

4       Paul          Napolitano   7.380952
```

2. It seems like some players may excel in one statistical category, but produce very little in other areas. Are there any players that are exceptional across many categories?

```
all_round_stats = ["points", "assists", "steals", "blocks", "turnovers", "rebounds"]

all_round_player = merged_data[all_round_stats +
["playerID"]].groupby("playerID").mean().nlargest(5, all_round_stats)

for i in all_round_stats:
    all_round_player[i + "Rank"] = all_round_player[i].rank(ascending = True, pct =  True)

all_round_player_merged = pd.merge(all_round_player[all_round_player.columns[-6:]],
master, how = "left", left_on = "playerID", right_on = "bioID")[["bioID",
"firstName","middleName","lastName"]]

print(all_round_player_merged)
```

```
       bioID        firstName    middleName    lastName

0      jordami01    Michael      Jeffrey       Jordan

1      jamesle01    LeBron       Raymone       James

2      chambwi01    Wilton       Norman        Chamberlain

3      duranke01     Kevin       Wayne         Durant

4      malonka01    Karl         Anthony        Malone
```

```
all_round_player[all_round_player.columns[-6:]].plot(kind = "bar")
plt.show()
```

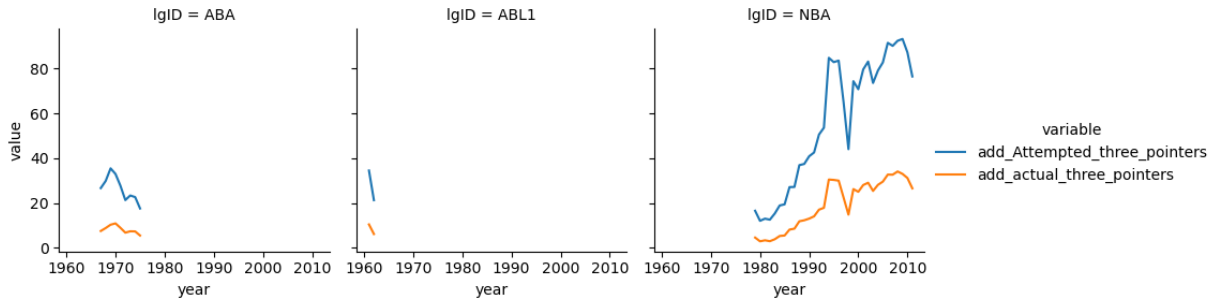


3. Much has been said about the rise of the three-point shot in recent years. It seems that players are shooting and making more three-point shots than ever. Recognizing that this dataset doesn't contain the very most recent data, do you see a trend of more three-point shots either across the league or among certain groups of players? Is there a point at which popularity increased dramatically?

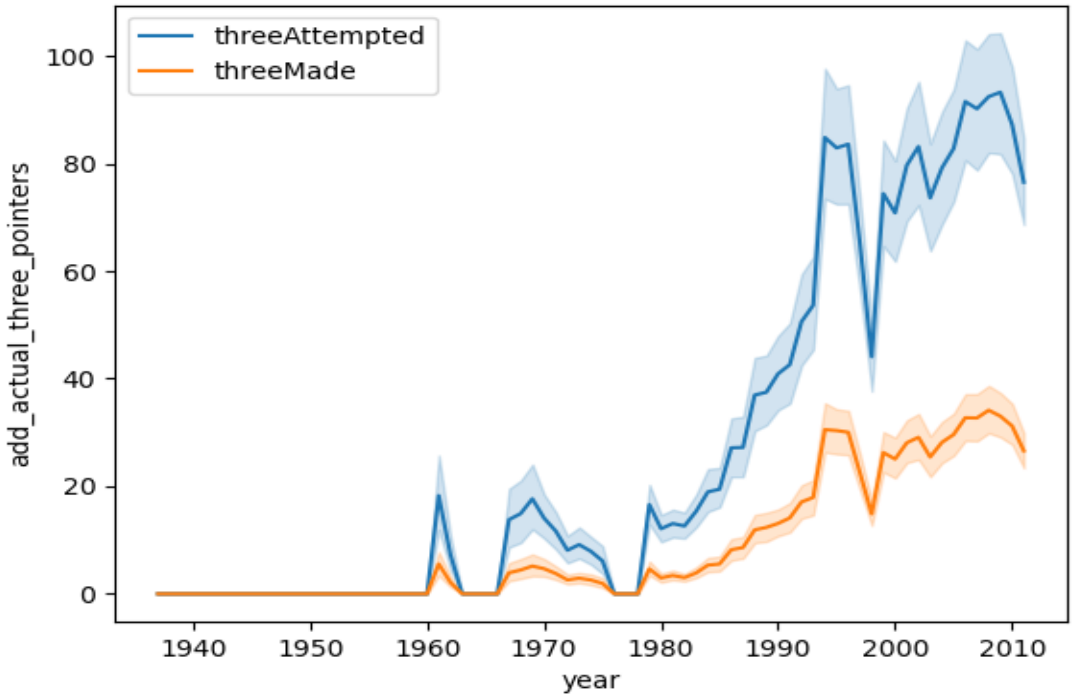
```
merged_data["add_Attempted_three_pointers"] = merged_data["threeAttempted"] +
merged_data["PostthreeAttempted"]

merged_data["add_actual_three_pointers"] = merged_data["threeMade"] +
merged_data["PostthreeMade"]

threePointData =
merged_data[["lgID","year","add_Attempted_three_pointers","add_actual_three_pointers"]].
groupby(["lgID","year"]).mean()

threePointData = threePointData.reset_index()
```

```
threePointData = threePointData[(threePointData["add_Attempted_three_pointers"] > 0) &
(threePointData["add_actual_three_pointers"] > 0)]

threePointData = threePointData.melt(["lgID","year"])
```

```
grid = sns.FacetGrid(threePointData, col = "lgID", hue= "variable")
grid.map(sns.lineplot, "year", "value").add_legend()
plt.show()
```




```
line1 = sns.lineplot(x = "year", y = "add_Attempted_three_pointers", data = merged_data)

line2 = sns.lineplot(x = "year", y = "add_actual_three_pointers", data = merged_data)

plt.legend(['threeAttempted','threeMade'])
plt.show()
```



PART III - SHOW CREATIVITY

1. Many sports analysts argue about which player is the GOAT (the Greatest Of All Time). Based on this data, who would you say is the GOAT? Provide evidence to back up your decision.

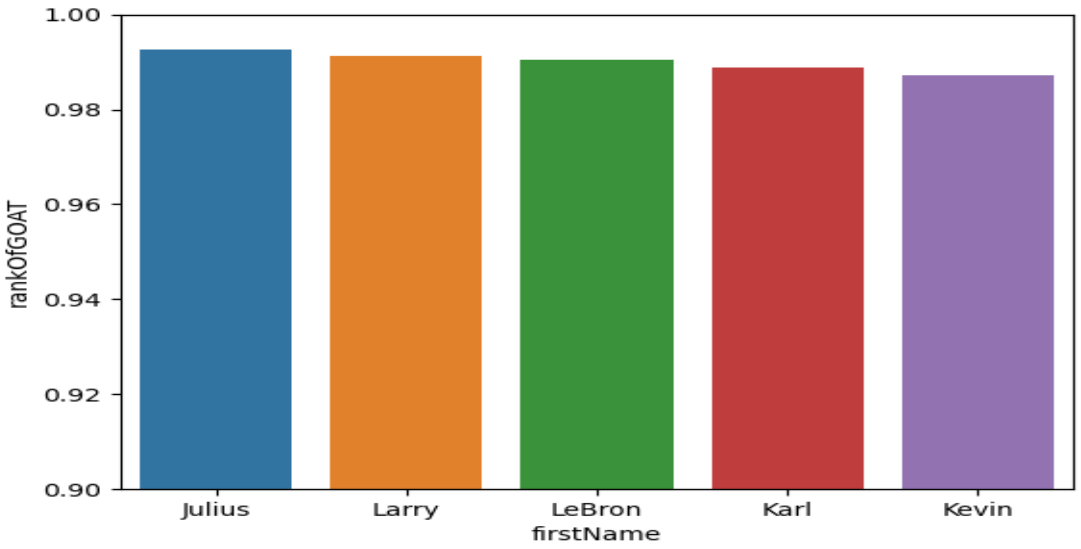
```
statistics = ["points","rebounds","assists","steals","blocks","turnovers"]

player_stats = merged_data[["playerID"] + statistics].groupby("playerID").mean()

for i in statistics:
    player_stats[i + "Rank"] = player_stats[i].rank(ascending = True, pct =  True)

rank = player_stats.iloc[:,[x for x in range(6,12)]]

rank_goat = rank * [0.35, 0.2, 0.15, 0.1, 0.1, 0.1]

rank_goat["rankOfGOAT"] = rank_goat.sum(axis = 1)

top_5_goats = rank_goat.nlargest(5,"rankOfGOAT")
```

I generated a GOAT Rank that sums rank of points, rebounds, assists, steals, blocks, and turnovers for each player based on a percentage format and then allot a proportion of 35% to points, 20% to rebounds, 15% to assists, and 10% to rest of stats. Below is the list of top five players who can be referred as GOAT:

```
top_5_goats = pd.merge(top_5_goats["rankOfGOAT"], master, how="inner",
left_on="playerID", right_on="bioID")[["firstName","middleName","lastName",
"rankOfGOAT"]]

print(top_5_goats)
```

	firstName	middleName	lastName	rankOfGOAT
0	Julius	Winfield	Erving	0.992617
1	Larry	Joe	Bird	0.991158
2	LeBron	Raymone	James	0.990455
3	Karl	Anthony	Malone	0.988803
4	Kevin	Maurice	Garnett	0.987100

```
sns.barplot(x="firstName", y="rankOfGOAT", data=top_5_goats)

plt.ylim(0.9, 1)
plt.show()
```

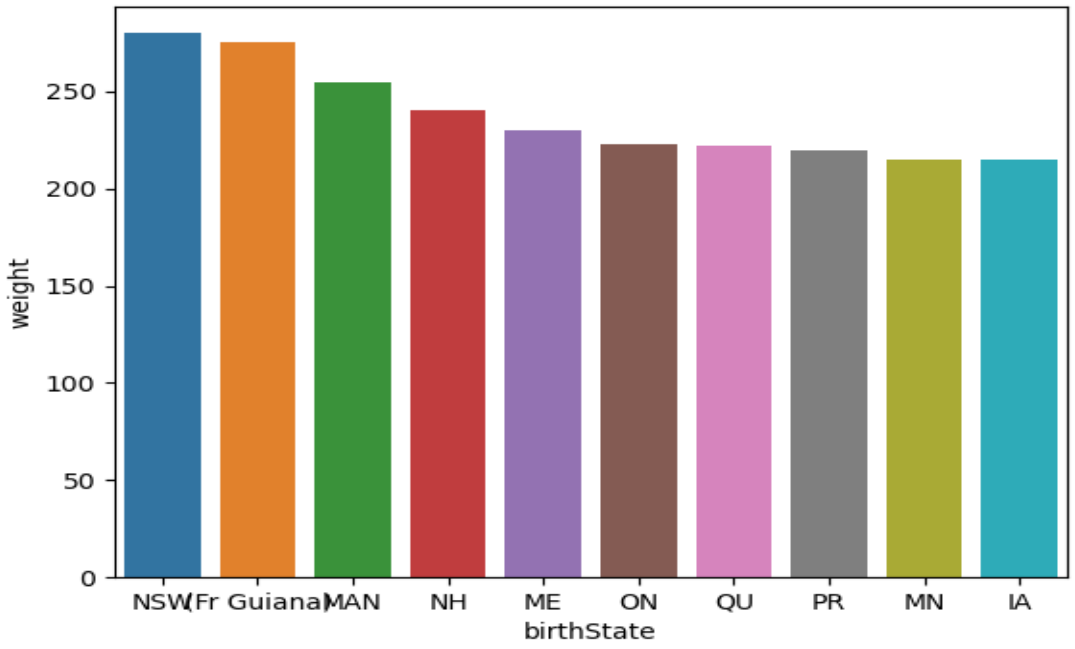


2. The biographical data in this dataset contains information about home towns, home states, and home countries for these players. Can you find anything interesting about players who came from a similar location?

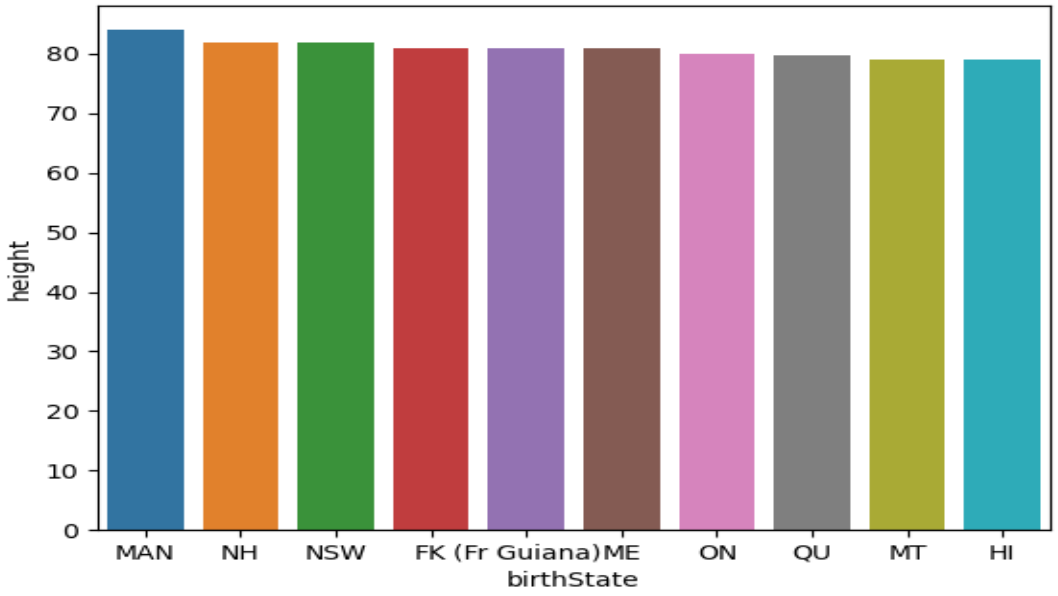
```
playerHeight = master[["birthState",
"height"]].groupby("birthState").mean()["height"].sort_values(ascending =
False).nlargest(10)

playerWeight = master[["birthState",
"weight"]].groupby("birthState").mean()["weight"].sort_values(ascending =
False).nlargest(10)
```

The first bar charts shows the mean weight of players based on their birth state. MAN, Fr. Guiana and NH states shows the high rate of weight compared to other states. The same can be said of the player's height. The data showed that players from MAN tends to be the tallest of all the states.

```
playerHeight = playerHeight.reset_index()

sns.barplot(data=playerHeight, x="birthState", y="height")
plt.show()
```




```
playerWeight = playerWeight.reset_index()

sns.barplot(data=playerWeight, x="birthState", y="weight")
plt.show()
```



3. Find something else in this dataset that you consider interesting. Produce a graph to communicate your insight.

How much height have been changed over the years per position

```
changeInHeight = merged_data[["pos","year","height","weight"]].replace([np.inf, -
np.inf], np.nan).dropna()[merged_data["height"] > 0][merged_data["weight"] > 0]

changeInHeight = changeInHeight.reset_index()
sns.lineplot(x="year", y="height", data=changeInHeight, hue="pos")

plt.show()
```