13 Prove: Data Analysis

PART I

1. Mean and median of points scored.

```
mean = players['points'].mean()
median = players['points'].median()
print(f'Mean: {mean:.2f}, Median: {median:.2f}')
```

```
Mean: 492.13, Median: 329.00
```

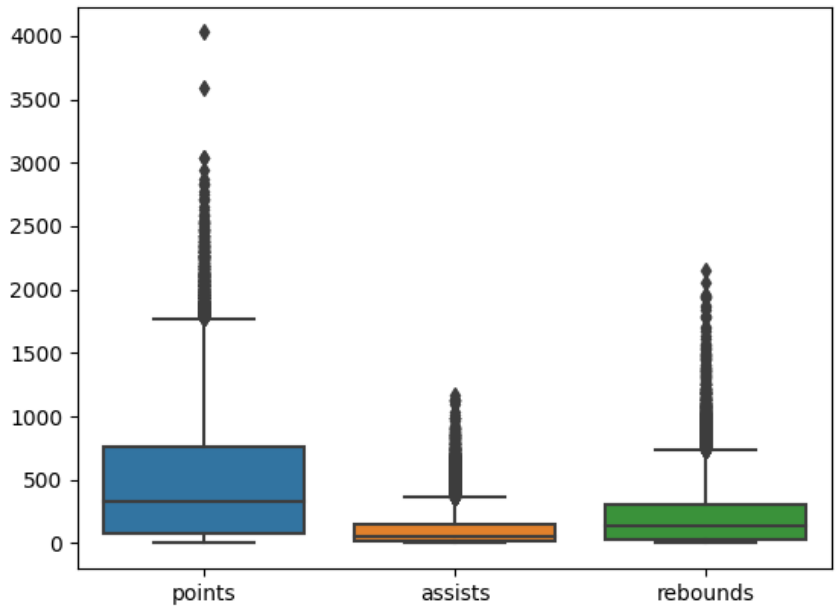
2. Highest number of points recorded in a season

```
print(players[['playerID', 'year', 'points']].sort_values(
    'points', ascending=False).head(1))
```

```
  playerID  year  points
chambwi01  1961    4029
```

3. Distribution of total points, assists, and rebounds.

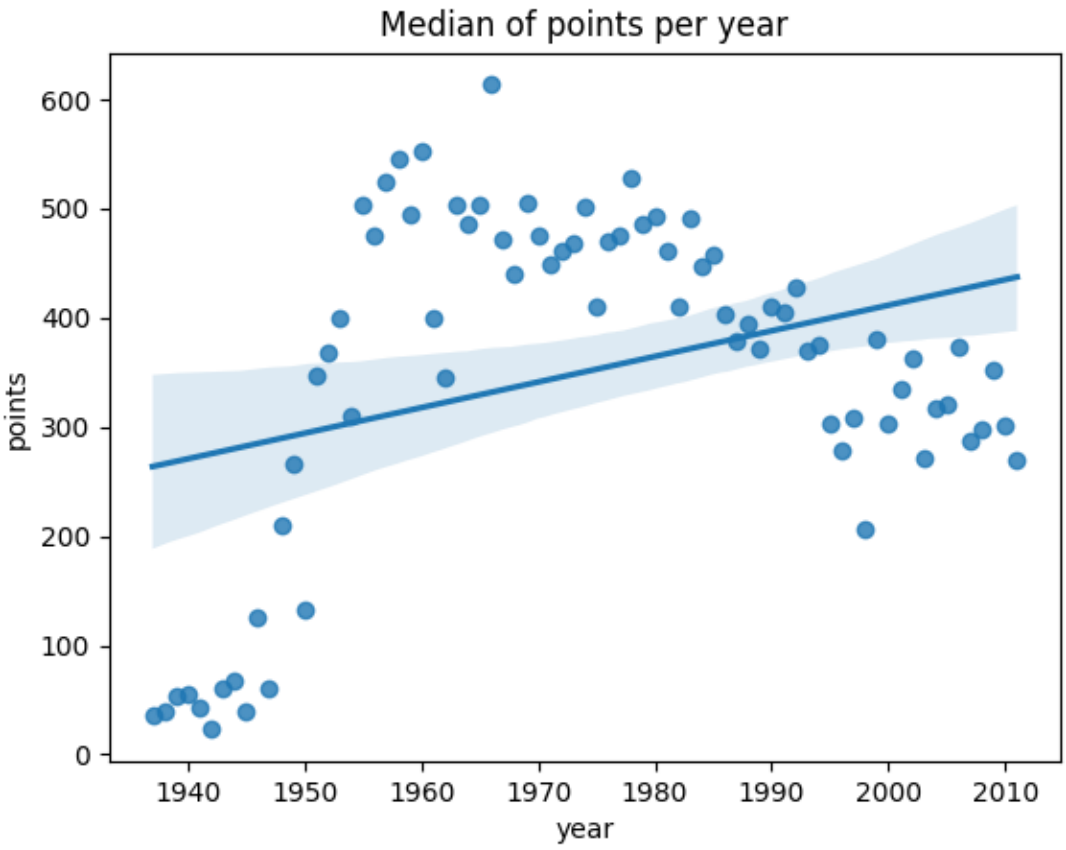
```
sns.boxplot(data=players[['points', 'assists', 'rebounds']])
plt.show()
plt.savefig("boxplot_rebounds.png")
```



4. Median of points scored per year, over time.

```
years_median = players[['points', 'year']].groupby('year').median()

years_median = years_median.reset_index()
sns.regplot(data=years_median, x='year', y='points').set_title('Medium of points per
year')
plt.show()
plt.savefig("boxplot_pointYears.png")
```



PART II

1. Players that scored more compared to attempts.

```
sum_attempts = players['fgAttempted'] + players['ftAttempted']
players['totalAttempts'] = sum_attempts
print(players[['playerID', 'totalAttempts', 'points']].sort_values('totalAttempts',
ascending=False).head(10))
```

```
  playerID  totalAttempts  points
0 chambwi01           4522    4029
1 chambwi01           3883    3586
2 chambwi01           3511    3033
3 chambwi01           3314    2948
4 chambwi01           3302    2707
5 jordami01           3251    3041
6 barryri01           3092    2775
7 bayloel01           3063    2719
8 bayloel01           3029    2538
9 bryanko01           2992    2832
```

There is one player who was more efficient at scoring. The player with ID ‘chambwi01’ scored more points per attempt.

2. Exceptional players across many categories.

```
print(players[['playerID', 'points']].sort_values(
    'points', ascending=False).head(10))

print(players[['playerID', 'rebounds']].sort_values(
    'rebounds', ascending=False).head(10))

print(players[['playerID', 'assists']].sort_values(
    'assists', ascending=False).head(10))

print(players[['playerID', 'steals']].sort_values(
    'steals', ascending=False).head(10))

print(players[['playerID', 'blocks']].sort_values(
    'blocks', ascending=False).head(10))

print(players[['playerID', 'turnovers']].sort_values(
    'turnovers', ascending=False).head(10))

print(players[['playerID', 'fgAttempted', 'fgMade']
             ].sort_values('fgMade', ascending=False).head(10))

print(players[['playerID', 'ftAttempted', 'ftMade']
             ].sort_values('ftMade', ascending=False).head(10))
```

playerID	points
chambwi01	4029
chambwi01	3586
jordami01	3041
chambwi01	3033
chambwi01	2948
jordami01	2868
bryanko01	2832
mcadobo01	2831
abdulka01	2822
barryri01	2775

playerID	rebounds
chambwi01	2149
chambwi01	2052
chambwi01	1957
chambwi01	1952
chambwi01	1946
chambwi01	1943
chambwi01	1941
russebi01	1930
russebi01	1878
russebi01	1868

playerID	assists
stockjo01	1164
stockjo01	1134
stockjo01	1128
stockjo01	1126
thomais01	1123
stockjo01	1118
porteke01	1099
stockjo01	1031
stockjo01	1011
johnske02	991

playerID	steals
conlemi01	354
busedo01	346
roberal01	301
busedo01	281
richami01	265
stockjo01	263
wattssl01	261
roberal01	260
jordami01	259
mcclate01	250

playerID	blocks
eatonma01	456
gilmoar01	422
bolma01	397
smithel01	393
olajuha01	376
eatonma01	369
eatonma01	351
bolma01	345
rollitr01	343
olajuha01	342

playerID	turnovers
mcginge01	422
mcginge01	401
mcginge01	393
cunnibi01	381
gilmoar01	366
simpsra01	360
richami01	359
calvima01	356
brownla01	356
boonero01	355

playerID	fgAttempted	fgMade
chambwi01	3159	1597
chambwi01	2770	1463
chambwi01	2457	1251
chambwi01	2298	1204
abdulka01	2019	1159
jordami01	2279	1098
mcadobo01	2138	1095
chambwi01	1990	1074
jordami01	1998	1069
chambwi01	2311	1065

playerID	ftAttempted	ftMade
westje01	977	840
chambwi01	1363	835
jordami01	972	833
dantlad01	946	813
roberos01	938	800
duranke01	840	756
barryri01	852	753
roberos01	881	742
malonmo01	904	737
roberos01	843	736

The player that repeats in more than one category, as top 10 is chambwi01. He outstands on points, rebounds, field goals, and free-throws categories.

3. Three-point shots.

```
print(players[['playerID', 'threeMade', 'year', 'tmID', 'lgID']].sort_values(
    'threeMade', ascending=False).head(10))
```

playerID	threeMade	year	tmID	lgID
allenra02	269	2005	SEA	NBA
scottde01	267	1995	ORL	NBA
mccloge01	257	1995	DAL	NBA
richaja01	243	2007	CHR	NBA
stojape01	240	2003	SAC	NBA
blaylmo01	231	1995	ATL	NBA
stojape01	231	2007	NOH	NBA
millere01	229	1996	IND	NBA
allenra02	229	2001	MIL	NBA
korveky01	226	2004	PHI	NBA

The top 10 players that scored more three-point shootings played in the NBA league. The popularity increased in 2005.

PART III

1. The GOAT player.

```
print(players[['playerID', 'points']].sort_values(
'points', ascending=False).head(10))
print(players[['playerID', 'rebounds']].sort_values(
    'rebounds', ascending=False).head(10)
print(players[['playerID', 'fgAttempted', 'fgMade']
].sort_values('fgMade', ascending=False).head(10))
print(players[['playerID', 'ftAttempted', 'ftMade']
].sort_values('ftMade', ascending=False).head(10))
```

playerID	points
chambwi01	4029
chambwi01	3586
jordami01	3041
chambwi01	3033
chambwi01	2948
jordami01	2868
bryanko01	2832
mcadobo01	2831
abdulka01	2822
barryri01	2775

playerID	rebounds
chambwi01	2149
chambwi01	2052
chambwi01	1957
chambwi01	1952
chambwi01	1946
chambwi01	1943
chambwi01	1941
russebi01	1930
russebi01	1878
russebi01	1868

playerID	fgAttempted	fgMade
chambwi01	3159	1597
chambwi01	2770	1463
chambwi01	2457	1251
chambwi01	2298	1204
abdulka01	2019	1159
jordami01	2279	1098
mcadobo01	2138	1095
chambwi01	1990	1074
jordami01	1998	1069
chambwi01	2311	1065

playerID	ftAttempted	ftMade
westje01	977	840
chambwi01	1363	835
jordami01	972	833
dantlad01	946	813
roberos01	938	800
duranke01	840	756
barryri01	852	753
roberos01	881	742
malonmo01	904	737
roberos01	843	736

I think chambwi01 is the greatest of all time because he has the highest score in points, rebounds, and field goals. Only in free-throws he has second place.

2. Something interesting about players from same State.

```
player_bio = pd.read_csv("basketball_master.csv")
player_data = pd.merge(players, player_bio, how="left",
                       left_on="playerID", right_on="bioID")

print(player_data[['playerID', 'birthCity', 'birthState', 'birthCountry',
'steals']].sort_values('steals', ascending=False).head(10))

print(player_data[['playerID', 'birthCity', 'birthState', 'birthCountry',
'turnovers']].sort_values('turnovers', ascending=False).head(10))
```

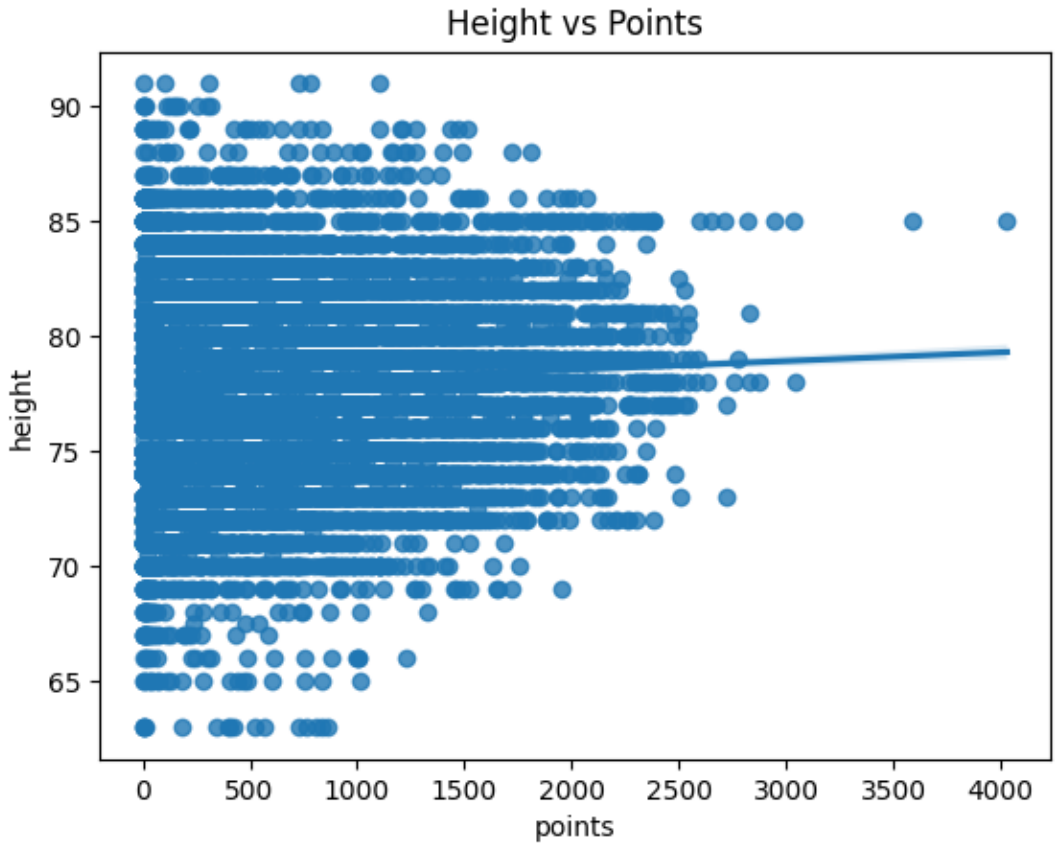
```
  playerID     birthCity birthState birthCountry  steals
 conlemi01  Indianapolis         IN          USA     354
  busedo01   Huntingburg         IN          USA     346
 roberal01     Barberton         OH          USA     301
  busedo01   Huntingburg         IN          USA     281
 richami01       Lubbock         TX          USA     265
 stockjo01       Spokane         WA          USA     263
 wattssl01  Rolling Fork         MS          USA     261
 roberal01     Barberton         OH          USA     260
 jordami01      Brooklyn         NY          USA     259
 mcclate01     Nashville         TN          USA     250

 playerID     birthCity birthState birthCountry  turnovers
mcginge01   Indianapolis         IN          USA        422
mcginge01   Indianapolis         IN          USA        401
mcginge01   Indianapolis         IN          USA        393
cunnibi01       Brooklyn         NY          USA        381
gilmoar01        Chipley         FL          USA        366
simpsra01        Detroit         MI          USA        360
richami01        Lubbock         TX          USA        359
calvima01     Fort Worth         TX          USA        356
brownla01        Brooklyn         NY          USA        356
boonero01  Oklahoma City         OK          USA        355
```

Players who were born in the state of Indiana, like conlemi01, busedo01, and mcginge01, are in the top 10 of more steals and turnovers.

3. Something interesting in the dataset.

```
print(player_data[['height', 'points']].sort_values(
    'height', ascending=False).head(10))
player_data = player_data[player_data["height"] > 0]
sns.regplot(data=player_data, x='points', y='height').set_title(
    'Height vs Points')
plt.show()
plt.savefig("boxplot_height.png")
```



I wanted to know if a player's height would be directly related to the points scored to see if someone had a physical advantage over other players who scored less. But looking at the graph, I realized that even the tallest players had not scored any points. Nevertheless, the shortest one did not score over 2000 points. The player who had the highest points has around 85 inches of height.