DATA ANALISIS – NBA BASKETBALL DATA.

Student: **Alegre Bustos, Cristian Nahuel**

Point 1)

In first place we read both files that we need for this assignment.

```
players= pd.read_csv("players.csv", low_memory=False)

master=pd.read_csv("master.csv", low_memory=False)
```

Then we calculate the mean and median points from the columns *points* with this code. I get that the *mean point* is equal to **492.130**, and the *median points* is **329.**

```
#Get the mean and the median points.
mean = players.points.mean()
median=players.points.median()
```

Point 2)

To get the highest number of point in a single season I did:

First, Merge the players and master .csv file.

Second, I did a DataSet called *"HigherPoint_Row"* selecting the columns called ***year*** or season, ***useFirst*** or first name***, lastName*** and ***points.*** Sorting the values to ***points*** I used the. head *()* function to get the highest goal scorer.

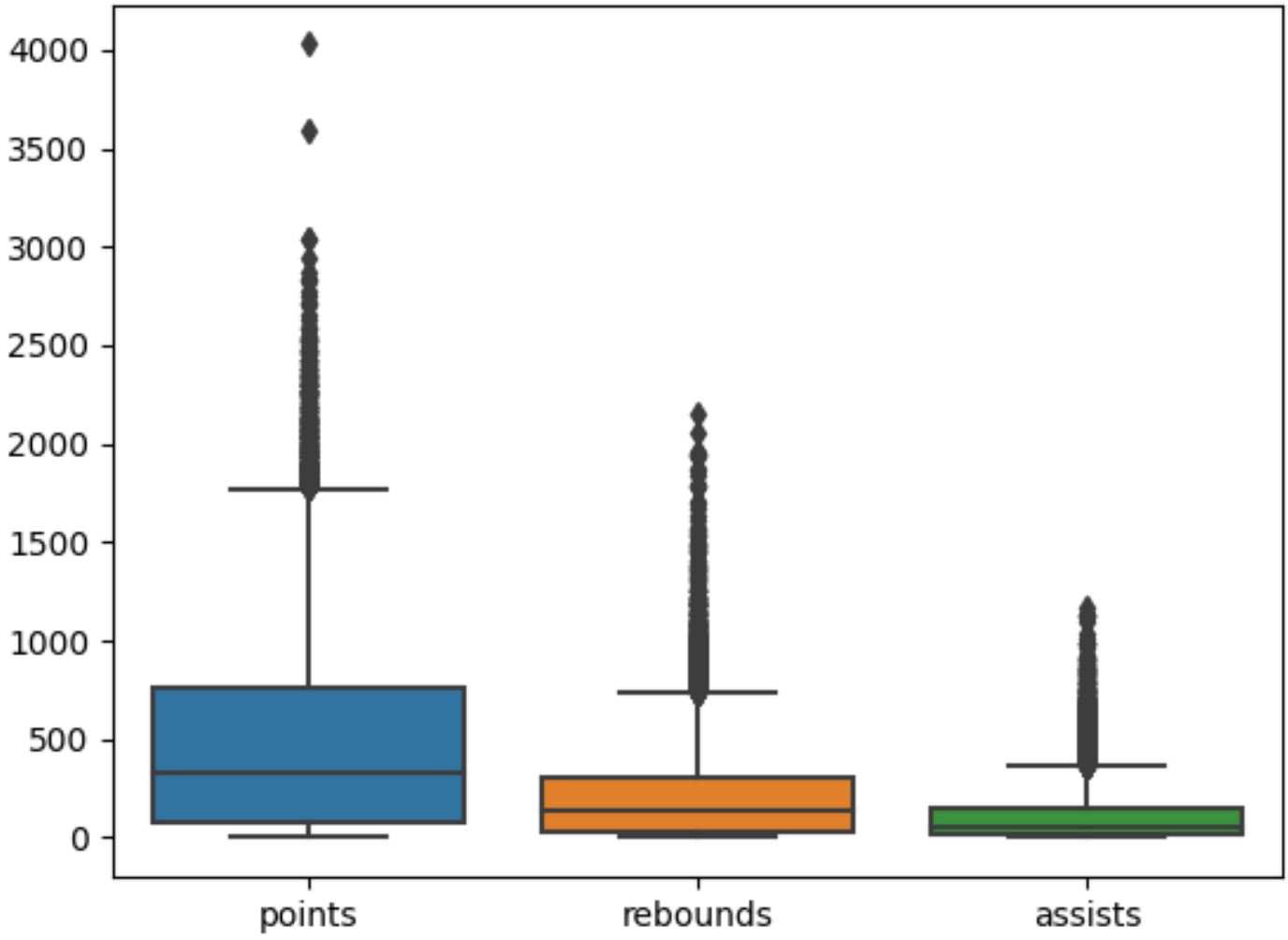
```
nba= pd.merge(players,master, how="left", left_on="playerID", right_on="bioID")

HigherPoint_Row= nba[["year", "useFirst", "lastName", "points"]].sort_values("points", ascending =False).head(1)
```

Point 3)

To produce a Box plot I did the next code, making reference to the columns: ***points, rebounds and assists.***

```
#boxplot for the points, rebunds and assists

Boxplot= sns.boxplot(data=nba[["points", "rebounds", "assists"]])

plt.show()
```

In the next page you can see the Box Plot:



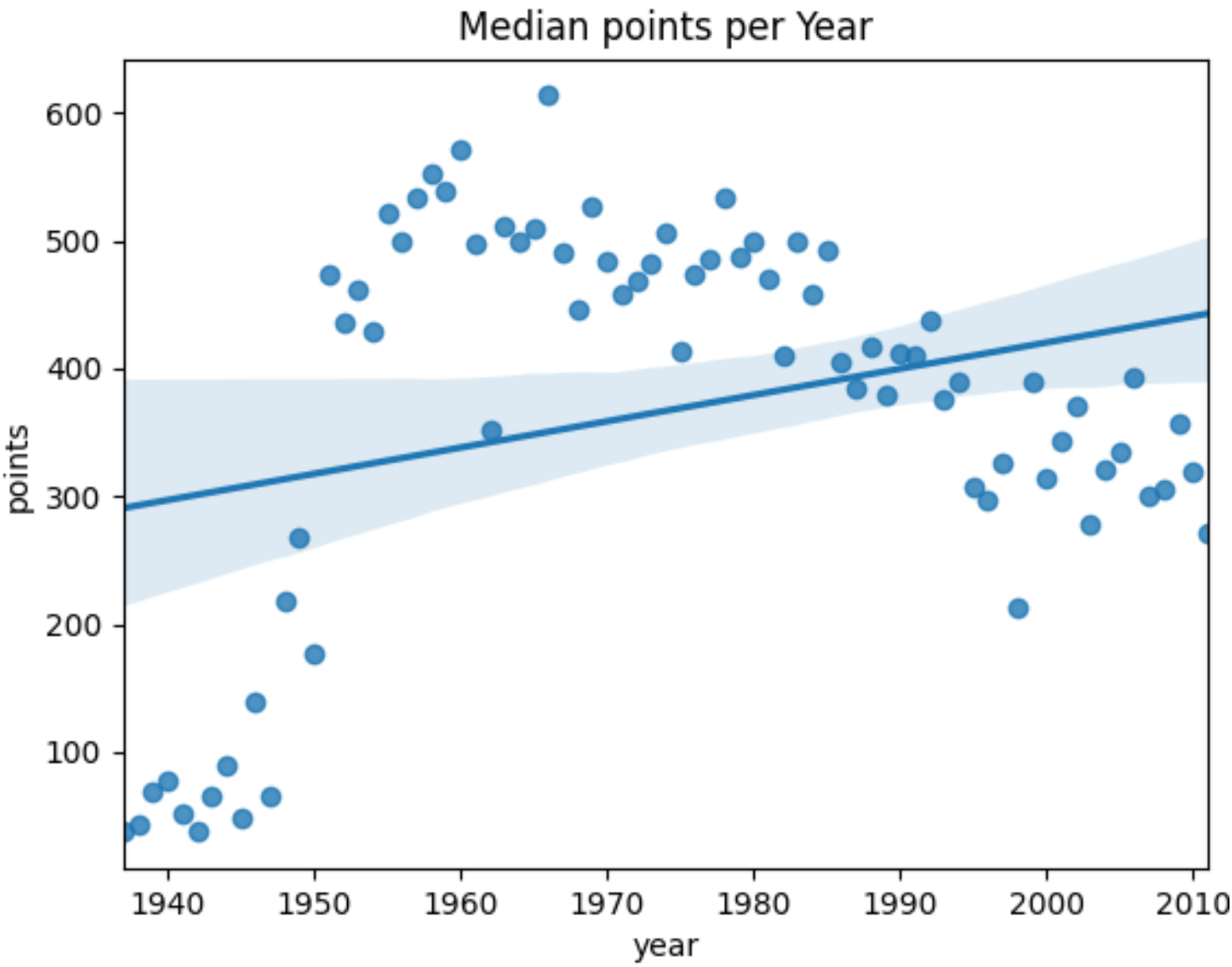
Point 4)

To show how the number of points scored has changed over time I grouped the ***year*** and ***points*** in a same table and I grouped the ***points*** by every ***year.***

```
yearsVSpoint= nba[["year", "points"]][nba.points >0].groupby("year",as_index=False).median()

print (yearsVSpoint)

yearsVSpoint = yearsVSpoint.reset_index()

sns.regplot(data=yearsVSpoint, x="year", y="points").set_title("Median points per Year")

plt.show()
```

Even taught in the plot we can see that the years with more points are between 1957 and 1980. Even still we can see a general upward trend, we can see the median points per year is coming down after 1985.

In the next page is the plot.



Part 2

To this part of the project I had to learn a little about the different kind of goals in Basket. So I did a comparison between the efficiency of every player with every kind of throw.

First, I did a new column called ***NameComplete*** and concatenate two columns to get a full name of the players.

```
#Concatenate two columns to get a full name of the players
nba.insert(nba.columns.get_loc("playerID")+1, "NameComplete", nba["useFirst"] +" "+ nba["lastName"]  )
```

Second, I did and inserted a new column for every kind of throw: ***FGEfficiency*** (Field Goal Efficiency) , ***FTEfficiency*** (Free Goal Efficiency) , ***ThreeEfficiency*** (Three Goal or Three throw Efficiency).

I calculate every ***efficiency*** using the other columns made and Attempted. Usign : *made/Attempted *100*

Here is the code:

```
#Make a column with the efficiency % of the Field Goal. Usign : made/Attempted *100
nba.insert(nba.columns.get_loc("fgMade") + 1 , "FGEfficiency",nba["fgMade"] / nba["fgAttempted"] * 100 )

#Make a column with the efficiency of the Free Throw. Usign : made/Attempted *100
nba.insert(nba.columns.get_loc("ftMade") + 1 , "FTEfficiency",nba["ftMade"] / nba["ftAttempted"] * 100 )

#Make a column with the efficiency of the Three Throw. Usign : made/Attempted *100
nba.insert(nba.columns.get_loc("threeMade") + 1 , "ThreeEfficiency",nba["threeMade"] / nba["threeAttempted"] * 100 )
```

Then, I inserted a new column called ***AverageEfficiency*** where I got the average of the three kind of throw.

```
#Make a column with the average of  efficiency. So we summ the differents Throw or goal Efficiency  and divide per three

nba.insert(nba.columns.get_loc("ThreeEfficiency")+1, "AverageEfficiency", (nba["FGEfficiency"] + nba["FTEfficiency"] + nba["ThreeEfficiency"]) /3
```

After of that I set the number of player that I will to compare. `N_Player=5`

And then I make a new DataSet called ***playerEfficiency*** with the date above mentioned.

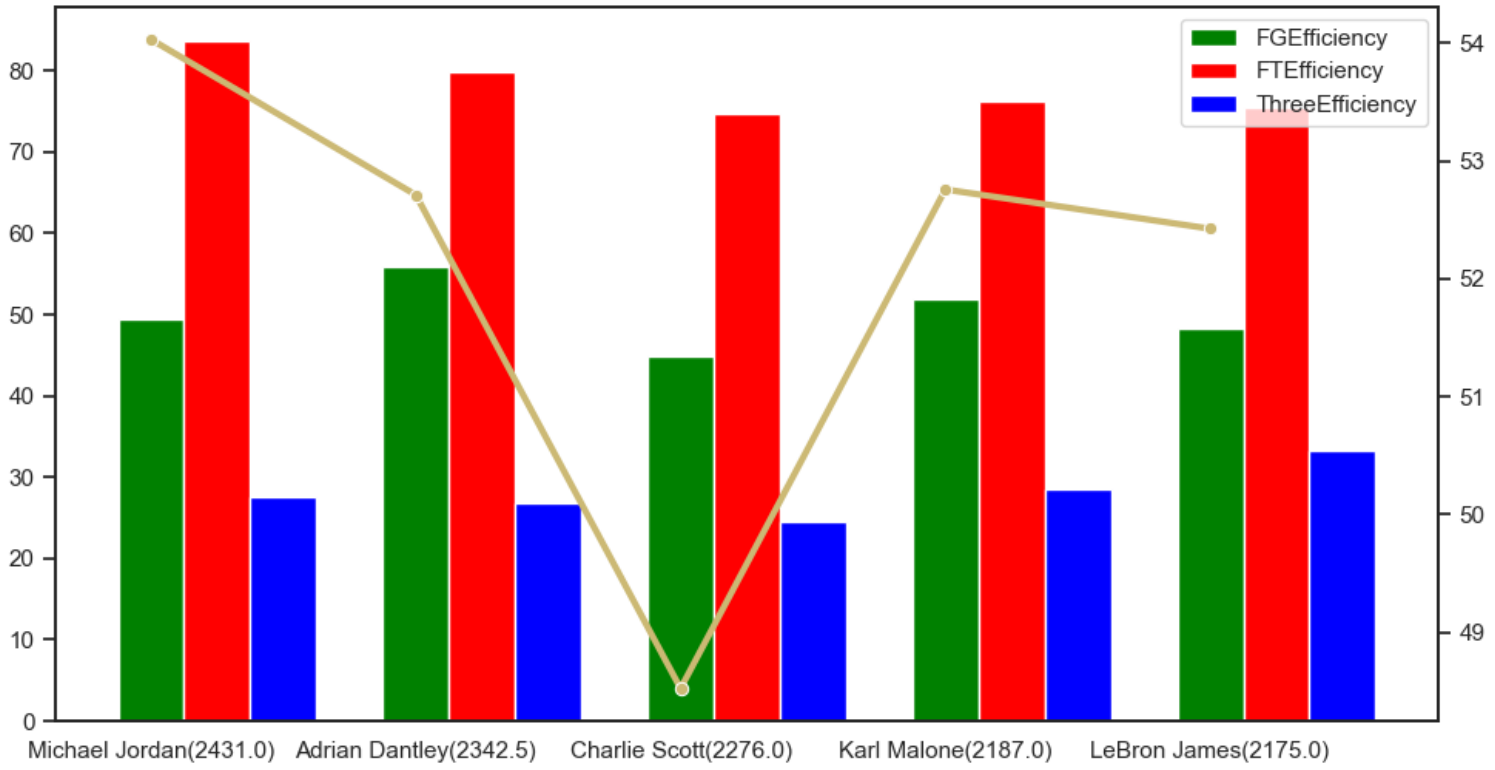
```
playerEfficiency=nba[["NameComplete","points","FGEfficiency","FTEfficiency","ThreeEfficiency"
, "AverageEfficiency"]][nba.threeMade >0 ] [nba.threeAttempted>0] .groupby("NameComplete",as_
index=False).median().sort_values("points", ascending=False).head(N_Player)
```

In the next table we see a comparison between the five principal goal scorer:

NameComplete	points	FGEfficiency	FTEfficiency	ThreeEfficiency	AverageEfficiency
Michael Jordan	2431.0	49.513514	83.655536	27.551020	54.030175
Adrian Dantley	2342.5	55.820802	79.914924	26.785714	52.711586
Charlie Scott	2276.0	44.936131	74.631751	24.615385	48.524940
Karl Malone	2187.0	51.907895	76.328502	28.571429	52.757528
LeBron James	2175.0	48.355664	75.434783	33.333333	52.424654

In this table we can see that Michael Jordan is the goal scorer with more points and with more Free Throw Efficiency. This Efficiency put him with the Average Efficiency, and set him in the first place. Also we can see that there are others players better than Michael Jordan with the other kind of throws.

I did a bar graphic when we can see better this analysis. In the same graphic I did a line graphic that show the **AverageEfficiency.**



This is the code that I used to plot this bar chart:

```
Eficiency_list= ["FGEfficiency","FTEfficiency","ThreeEfficiency"]
colors = ['green', 'red', 'blue']
numerical = [[x for x in playerEfficiency.FGEfficiency],[x for x in playerEfficiency.FTEffici
ency],[x for x in playerEfficiency.ThreeEfficiency] ]
average= [x for x in playerEfficiency.AverageEfficiency]
points = [x for x in playerEfficiency.points]
Better_Names = [x for x in playerEfficiency.NameComplete]
points_name= zip(points, Better_Names)
final_list= ["{}({})".format(y,str(x)) for x,y in points_name]
```

```
sns.set(style="white", rc={"lines.linewidth": 3})
number_groups = len(Eficiency_list)
bin_width = 1.0/(number_groups+1)
fig, ax1 = plt.subplots(figsize=(6,6))
ax2 = ax1.twinx()
for i in range(number_groups):
    ax1.bar(x=np.arange(N_Player) + i*bin_width,
            height=numerical[i],
            width=bin_width,
            color=colors[i],
            align='center')
ax1.set_xticks(np.arange(len(playerEfficiency.NameComplete)) + number_groups/(2*(number_groups+1)))
# number_groups/(2*(number_groups+1)): offset of xticklabel

ax1.set_xticklabels(playerEfficiency.NameComplete)
ax1.legend(Eficiency_list, facecolor='w')
```

```
sns.lineplot(x=final_list,
             y=average,
             color='y',
             marker="o",
             ax=ax2)

plt.show()
```

Point 3)

To get the Trend of the Three-Points I did a DataSet ***yearsVSThree*** where I sort the Three points made by years.

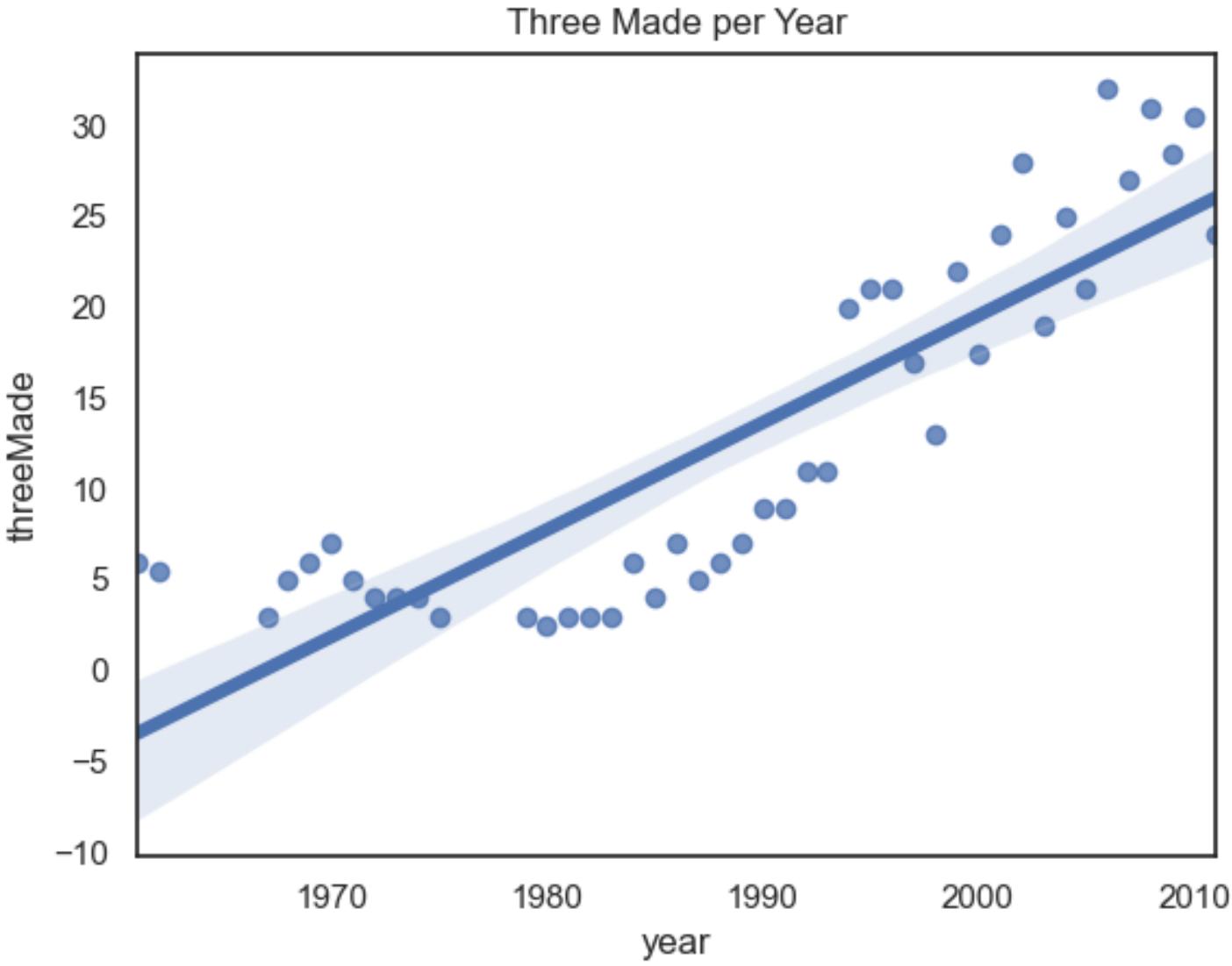
```
yearsVSThree= nba[["year", "threeMade"]][nba.threeMade>0].groupby("year",as_index=False).median()
```

I got the next table that I split in two part.

year	threeMade
1961	6.0
1962	5.5
1967	3.0
1968	5.0
1969	6.0
1970	7.0
1971	5.0
1972	4.0
1973	4.0
1974	4.0
1975	3.0
1979	3.0
1980	2.5
1981	3.0
1982	3.0
1983	3.0
1984	6.0
1985	4.0
1986	7.0
1987	5.0
1988	6.0
1989	7.0

1989	7.0
1990	9.0
1991	9.0
1992	11.0
1993	11.0
1994	20.0
1995	21.0
1996	21.0
1997	17.0
1998	13.0
1999	22.0
2000	17.5
2001	24.0
2002	28.0
2003	19.0
2004	25.0
2005	21.0
2006	32.0
2007	27.0
2008	31.0
2009	28.5
2010	30.5
2011	24.0

Just for watch this table we can see that the trend is upward, but we will watch the next char to confirm this theory.



This chart show that after of 1970 the points made by the Three Throw started to increase and not will stop beyond of the year 2010.

Part 3

For to know the GOAT player I did a DataSet to compare ***points, assists, steals, rebounds and blocks*** of the best 5 players.

I wrote the next code:

```
GOAT= nba[["NameComplete","points","assists","steals","rebounds", "blocks"]].groupby("NameCom
plete",as_index=False).median().sort_values("points", ascending=False).head(N_Player)
```

This code is to add a column that concatenate the ***NameComplete*** and the ***Points.*** This helped me after to plot the line chart.

And I got the next table:

NameComplete	NamePoint	points	assists	steals	rebounds	blocks
Michael Jordan	Michael Jordan - 2431.0	2431.0	377.0	182.0	492.0	54.0
LeBron James	LeBron James - 2175.0	2175.0	539.0	125.0	556.0	58.0
Karl Malone	Karl Malone - 2106.0	2106.0	285.0	113.0	834.0	59.0
Oscar Robertson	Oscar Robertson - 2060.0	2060.0	724.0	0.0	494.0	0.0
George Gervin	George Gervin - 1965.0	1965.0	202.0	88.0	400.0	67.0

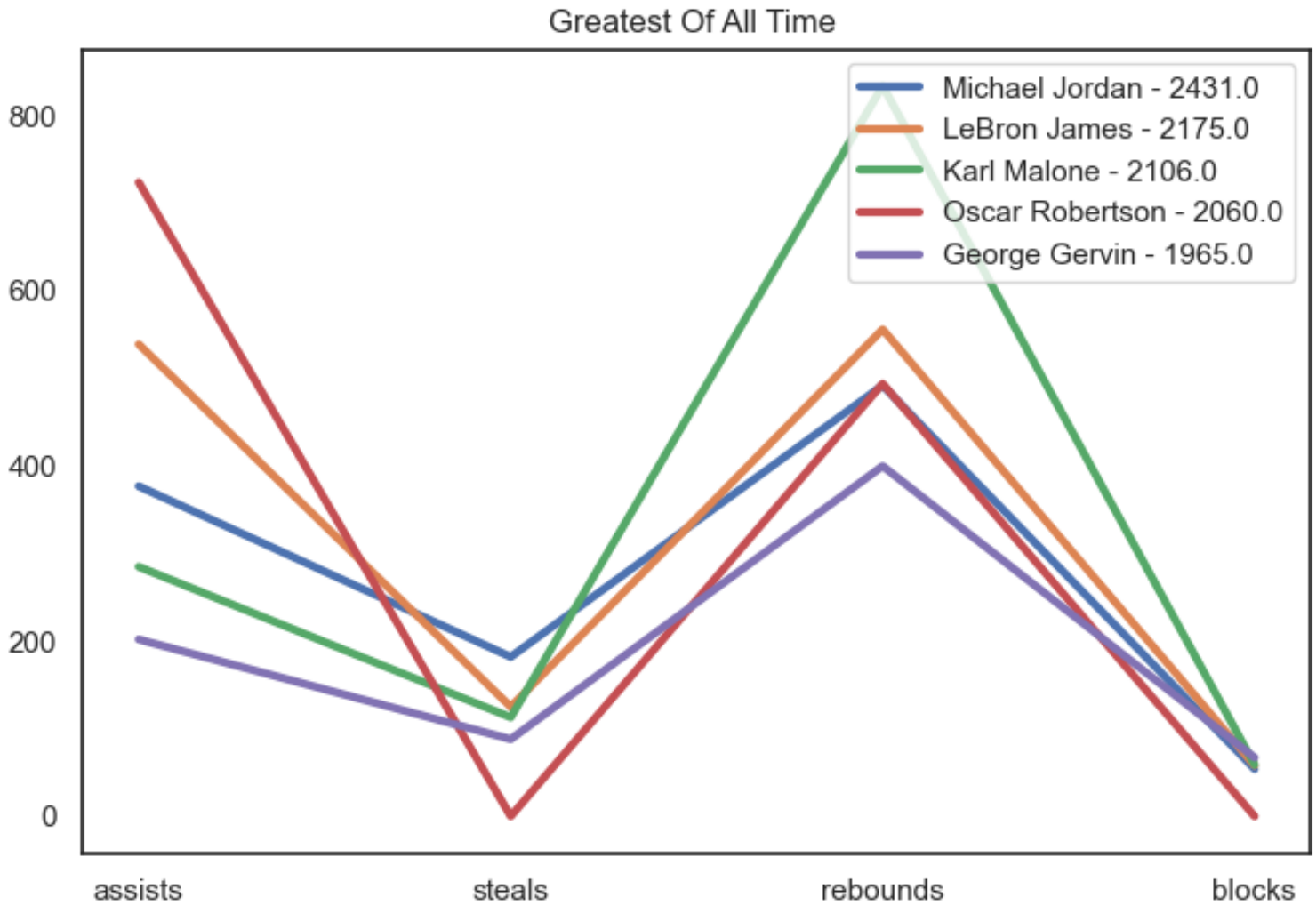
To get a multiply line chart I did the next code:

```
Row_dict={}
GOAT_list= ["assists","steals","rebounds", "blocks"]


for index, rows in GOAT.iterrows():
   Row_dict[rows.NamePoint] = [rows.assists,rows.steals, rows.rebounds, rows.blocks]


fig, ax1 = plt.subplots(figsize=(8.2, 5.4))
for key in Row_dict:
        ax1.plot(GOAT_list, Row_dict[key], label=key)


ax1.legend()
#ax1.set_ylabel('Percentage (%)')
ax1.set_title('Greatest Of All Time ')
plt.show()
```

The chart and table confirm that Michael Jordan is the GOAT player. I already show his great efficiency making points, but also he is in the first five positions in the others important characteristics.

The others players only could to overcome to Michael Jordan in only one aspect but did not show the same balanced than Michael.

May be we can wait that Michael Jordan would be the best in every category but he was a balanced player in assists, steals, rebounds and blocks. Also, he was the first player making goals with a highest Efficiency of more than % 54. This is very important. After of all the matches are won by scoring.


Greatest Of All Time
Michael Jordan - 2431.0
LeBron James - 2175.0
Karl Malone - 2106.0
Oscar Robertson - 2060.0
George Gervin - 1965.0
800
600
400
200
0
assists
steals
rebounds
blocks