Jae Wook Jung 151342596

Part 1

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.)

```
#1
points_mean = player.points.mean()
print("1) Points mean: {}".format(points_mean))
print()

points_median = player.points.median()
print("1) Points median: {}".format(points_median))
print()
```

```
1) Points mean: 492.1306892341375

1) Points median: 329.0
```

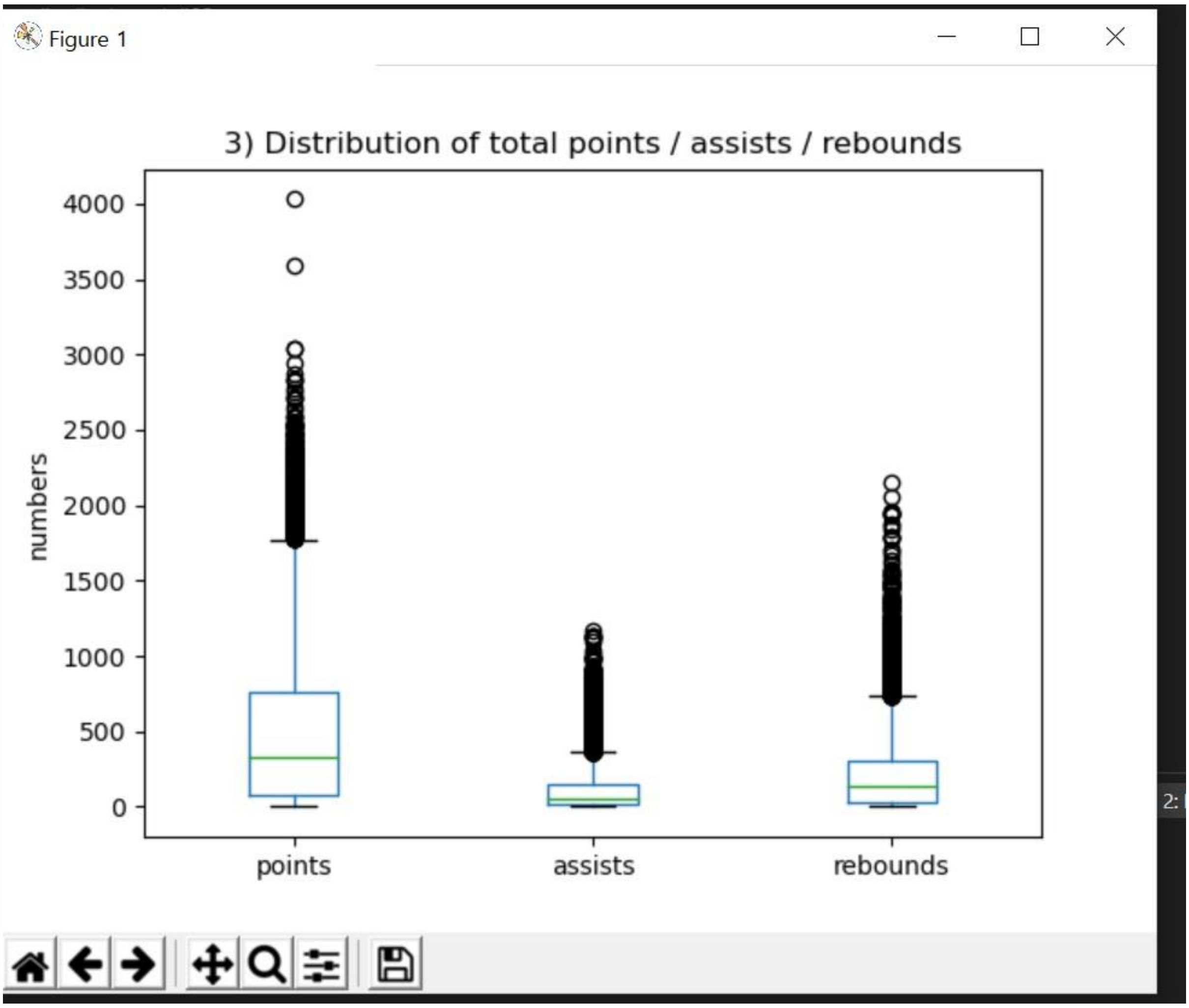
2. Determine the highest number of points recorded in a single season. Identify who scored those points and the year they did so.

```
#2
points_highest = player.sort_values(by = "points", ascending = False)
#print(player_master[player_master.playerID == points_highest.iloc[0][0]])
print()
name = player_master[player_master.playerID == points_highest.iloc[0][0]].iloc[0]["useFirst"] ## question
surname = player_master[player_master.playerID == points_highest.iloc[0][0]].iloc[0]["lastName"]
print()
print("2) Point: {}, name: {} {}, playerID: {}, year: {}".format(points_highest.iloc[0][8], name, surname, points_highest.iloc[0][0], points_highest.iloc[0][1]))
print()
```

```
2) Point: 4029, name: Wilt Chamberlain, playerID: chambwi01, year: 1961
```

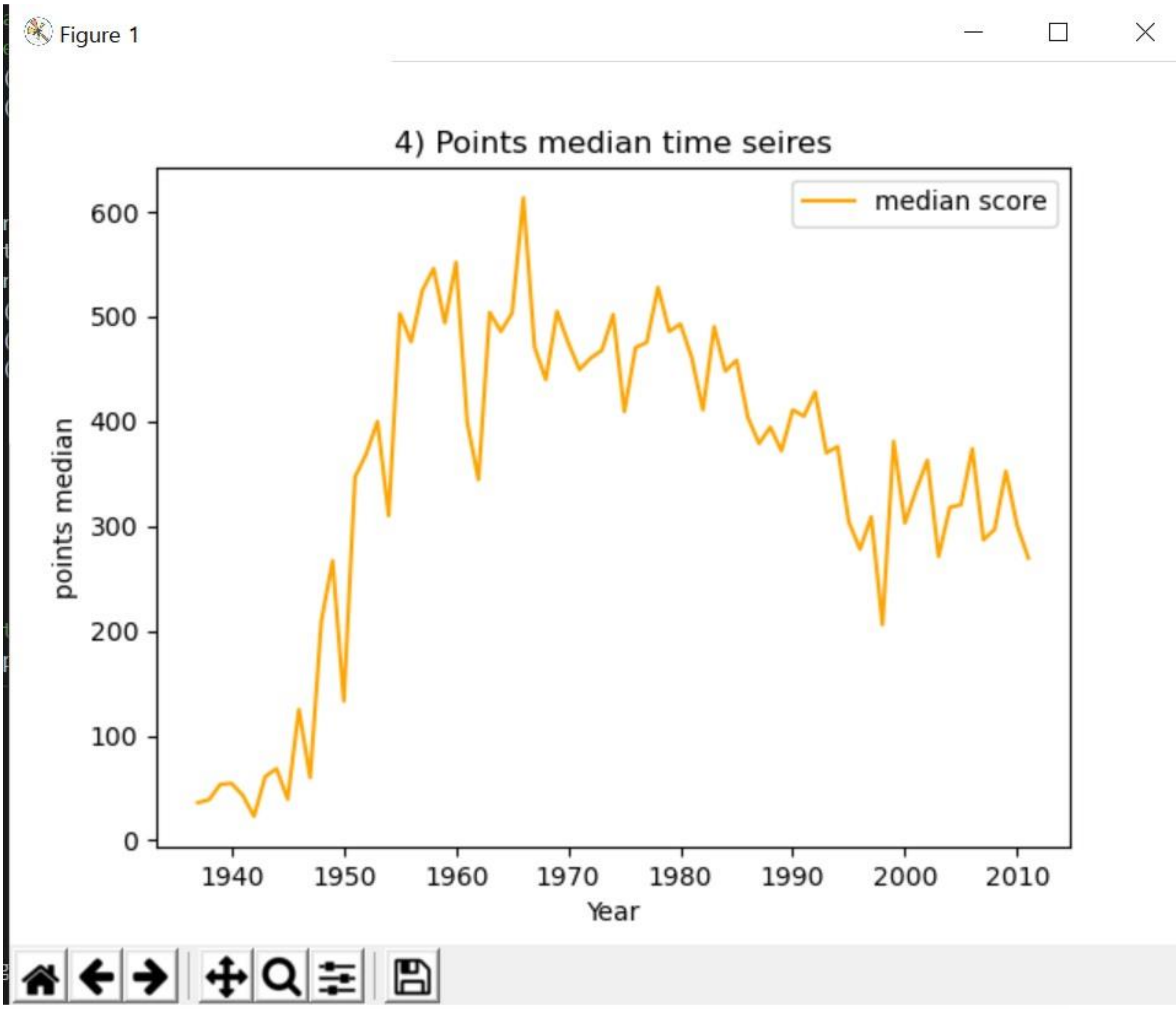
3. Produce a boxplot that shows the distribution of total points, total assists, and total rebounds (each of these three is a separate box plot, but they can be on the same scale and in the same graphic).

```
#3
three = player[["points", "assists", "rebounds"]]
#three_p = three.points.sum()
#three_a = three.assists.sum()
#three_r = three.rebounds.sum()
#three_total = [[three_p, three_a, three_r]]
#print(three_total)
three.plot(kind = "box", title = "3) Distribution of total points / assists / rebounds")
plt.ylabel("numbers")
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.

```
#4
points_change = player.groupby("year").points.median()
print(points_change)
points_change.plot(kind = "line", c = "orange", title = "4) Points median time seires", label = "median score")
plt.legend()
plt.xlabel("Year")
plt.ylabel("points median")
plt.show()
```



Part 2

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?

```
# egAttempted doesn't include ftAttempted
player1 = player[(player.fgAttempted > 0) & (player.ftAttempted)]
player1["points_eff"] = player1.points / (player1.fgAttempted + player1.ftAttempted)
player1 = player1[["playerID", "year", "points_eff", "fgAttempted", "ftAttempted"]]
points_efficient = player1.sort_values(by = "points_eff", ascending  = False).head(10)
print(points_efficient)
print()
print("{} is the most efficient goal maker.".format(points_efficient.iloc[0]["playerID"]))
print()
```

```
  player1["points_eff"] = player1.points / (player1.fgAttempted + player1.ftAttempted)
         playerID  year  points_eff  fgAttempted  ftAttempted
19828  conlemi01  2008    5.341151          146          323
19295  conlemi01  2007    5.040146           85          189
16625  slatere01  2001    1.666667            2            1
12481  brownch01  1993    1.500000            1            1
15518   langan02  1999    1.500000            1            1
9634   berrywa01  1986    1.444444            8            1
6980   colemec01  1978    1.375000            7            1
13303  youngda01  1994    1.333333           17            1
16668  vardara01  2001    1.250000            3            1
11579  wrighho02  1990    1.250000            3            1

conlemi01 is the most efficient goal maker.
```

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?

```
player["PPG"] = player.points / player.GP
player["RPG"] = player.rebounds / player.GP
player["APG"] = player.assists / player.GP
player["SPG"] = player.steals / player.GP

player2 = player[player.GP > 0]
#player2_in_order = player2.sort_values(by = "SPG", ascending = False)
#player2_in_order = player2_in_order.sort_values(by = "APG", ascending = False)
#player2_in_order = player2_in_order.sort_values(by = "RPG", ascending = False)
#player2_in_order = player2_in_order.sort_values(by = "PPG", ascending = False)
player2 = player2[["playerID", "year", "PPG", "RPG", "APG", "SPG"]]

player2["PPGRank"] = player2.PPG.rank(pct = True)
player2["RPGRank"] = player2.RPG.rank(pct = True)
player2["APGRank"] = player2.APG.rank(pct = True)
player2["SPGRank"] = player2.SPG.rank(pct = True)

print(player2[(player2.PPGRank > 0.95) & (player2.RPGRank > 0.95) & (player2.APGRank > 0.95) & (player2.SPGRank > 0.95)])
```

	playerID	year	PPG	RPG	APG	SPG	PPGRank	RPGRank	APGRank	SPGRank
4777	cunnibi01	1972	24.142857	12.047619	6.309524	2.571429	0.980676	0.981778	0.968938	0.996716
5590	ervinju01	1974	27.892857	10.880952	5.500000	2.214286	0.993601	0.970294	0.953004	0.991228
5708	mcginge01	1974	29.784810	14.253165	6.265823	2.607595	0.996568	0.993389	0.968281	0.996991
7571	birdla01	1980	21.231707	10.914634	5.500000	1.963415	0.957412	0.970591	0.953004	0.982583
7907	birdla01	1981	22.870130	10.870130	5.805195	1.857143	0.973303	0.970040	0.960505	0.978070
8254	birdla01	1982	23.632911	11.012658	5.797468	1.873418	0.978155	0.971989	0.960208	0.978791
8606	birdla01	1983	24.151899	10.075949	6.582278	1.822785	0.980803	0.958153	0.972625	0.976078
8934	birdla01	1984	28.687500	10.525000	6.637500	1.612500	0.995127	0.965484	0.973133	0.959170
9278	birdla01	1985	25.792683	9.817073	6.792683	2.024390	0.987457	0.954000	0.975061	0.985401
17124	webbech01	2002	23.014925	10.507463	5.432836	1.582090	0.974574	0.965209	0.950970	0.955674

Cunnibi01 was good in many categories for one year, but birdla01 was good in many categories for many years.

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?

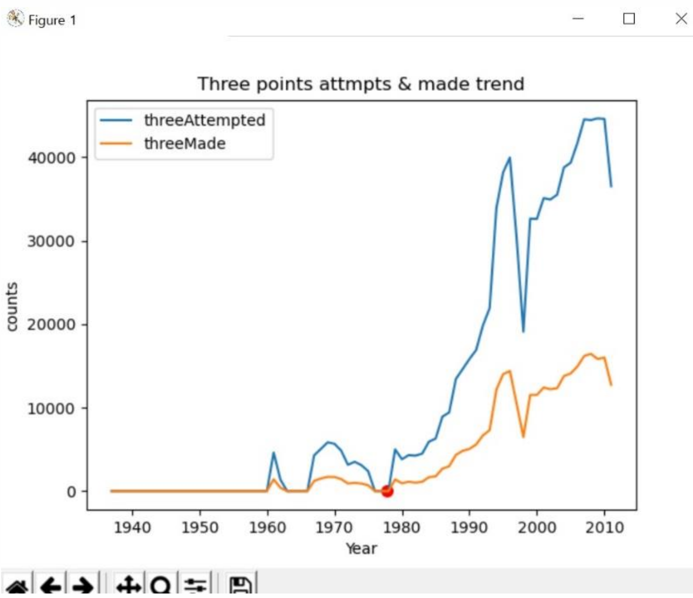
```python
three_points1 = player[["year", "threeAttempted", "threeMade"]]
print(three_points1)
three_points1 = three_points1.groupby("year").sum()
three_points1.plot(kind  = "line")
#plt.show()

#three_points[three_points.index > 1970].plot(kind = "line")
plt.scatter(1977.8, 53, s = 50, c = "red")
plt.title("Three points attmpts & made trend")
plt.xlabel("Year")
plt.ylabel("counts")
plt.show()

# add who threw the most three attempt
most_three_player = player.groupby("playerID").threeAttempted.sum()
most_three_player = most_three_player.sort_values(ascending = False)
print(most_three_player.head(3))
print()
print("The three players threw that most three points shots are: ")
for i in range(3):
    print("{} {} - {} attempts".format(player_master[player_master.playerID == most_three_player.index[i]].iloc[0]["useFirst"],
    player_master[player_master.playerID == most_three_player.index[i]].iloc[0]["lastName"], most_three_player[i]))
print()
```

```
playerID
allenra02    6788
millere01    6486
kiddja01     5376
Name: threeAttempted, dtype: int64

The three players threw that most three points shots are:
Ray Allen - 6788 attempts
Reggie Miller - 6486 attempts
Jason Kidd - 5376 attempts
```



The three points shots attempts were keep increasing since around 1778. Since the shot attempts increased, three points shots made increased too, but not so much as attempts increased.

Part 3

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. This question requires you to do additional analysis beyond the 2nd question in Part II above.

```
player["PPG"] = player.points / player.GP
player["RPG"] = player.rebounds / player.GP
player["APG"] = player.assists / player.GP
player["SPG"] = player.steals / player.GP

player2 = player[player.GP > 0]
#player2_in_order = player2.sort_values(by = "SPG", ascending = False)
#player2_in_order = player2_in_order.sort_values(by = "APG", ascending = False)
#player2_in_order = player2_in_order.sort_values(by = "RPG", ascending = False)
#player2_in_order = player2_in_order.sort_values(by = "PPG", ascending = False)
player2 = player2[["playerID", "year", "PPG", "RPG", "APG", "SPG"]]

player2["PPGRank"] = player2.PPG.rank(pct = True)
player2["RPGRank"] = player2.RPG.rank(pct = True)
player2["APGRank"] = player2.APG.rank(pct = True)
player2["SPGRank"] = player2.SPG.rank(pct = True)

player_goat = player2[(player2.PPGRank > 0.90) & (player2.RPGRank > 0.90) & (player2.APGRank > 0.90) & (player2.SPGRank > 0.90)]
player_goat = player_goat.playerID.value_counts().head(5)
print(player_goat)
```

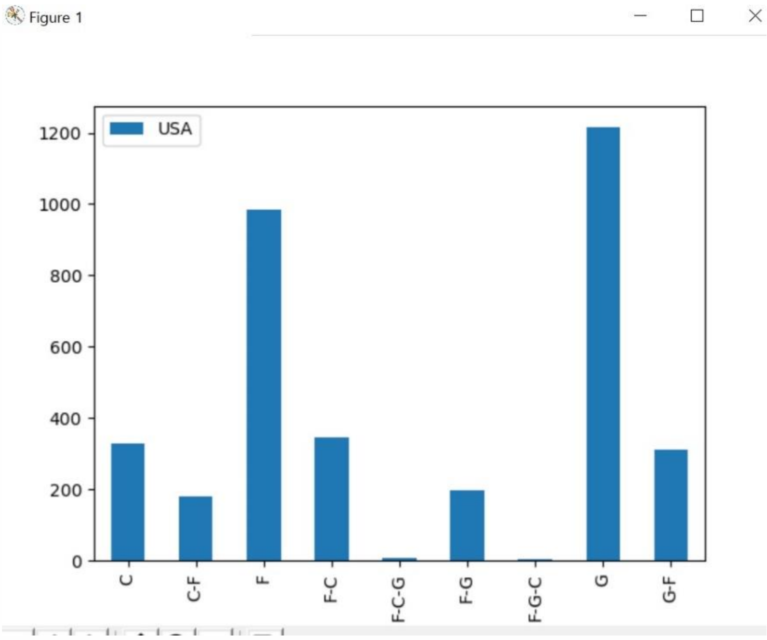
```
birdla01      11
webbech01      8
garneke01      7
barklch01      5
ervinju01      5
Name: playerID, dtype: int64
PS C:\Users\Jae\Desktop\CS241\W13>
```

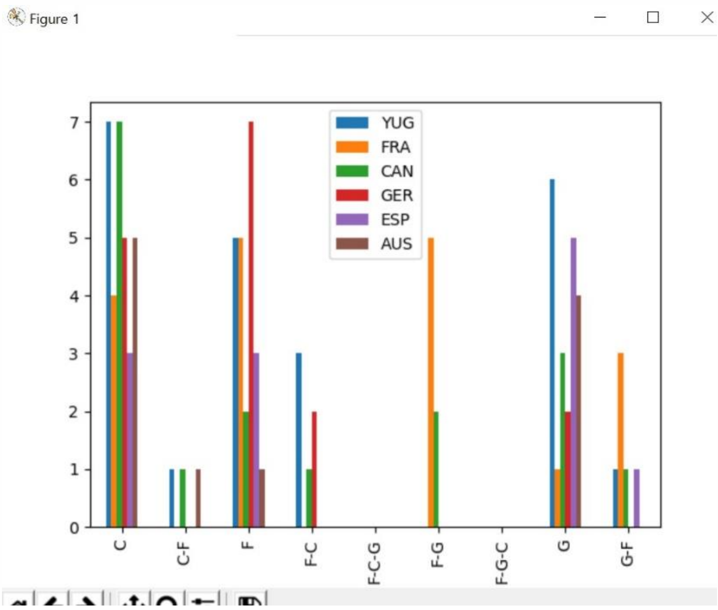
Birdla01 was above 90 percent in many categories for 11 years. Comparing to the average NBA basketball career, 4.8 years, it is a high number.

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?

```
country = []
for i in master.birthCountry:
    if i not in country:
        country.append(i)
print(country)
print()
country_positions = {}
for i in country:
    country_positions[i] = master[master.birthCountry == i].pos.value_counts()
print(country_positions)
df = pd.DataFrame(country_positions)
df = df.fillna(0)
df1 = df.sum().sort_values(ascending = False)
df1 = df1[df1>=10]
subset_ithers = df[["YUG", "FRA", "CAN", "GER", "ESP", "AUS"]].drop(["pos", " G"])
subset_USA = df[["USA"]].drop(["pos"," G"])
subset_USA.plot(kind = "bar")
plt.show()
subset_others.plot(kind = "bar")
plt.legend(loc = 'upper center')
plt.show()
```

NBA is mostly filled with Americans. Many of them are G and F. There are some players from other countries. I sorted the countries out that have greater and equal to 10 people with any position. Compare to other countries YUG and CAN nationality players are C in NBA. And many players from F were from GER. However, the numbers from the other country players are prominently small comparing to the USA players. Within USA players, there are more F and G.





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

   Which position throws the most three points shots.

   Position G throws the most three points shots.

```python
positions = []
for i in player_master.pos:
    if i not in positions:
        positions.append(i)
print(positions)
print()
player_master1 = player_master[["pos", "threeAttempted"]]
print(player_master1)
three_positions = player_master1.groupby("pos").threeAttempted.sum()
print(three_positions)
three_positions.plot(kind = "bar")
plt.title("Three points throws by positions")
plt.xlabel("Positions")
plt.ylabel("Throws")
plt.show()
```