INFO 2201 Final Project

Michael Patel

How can we visualize the NBA?

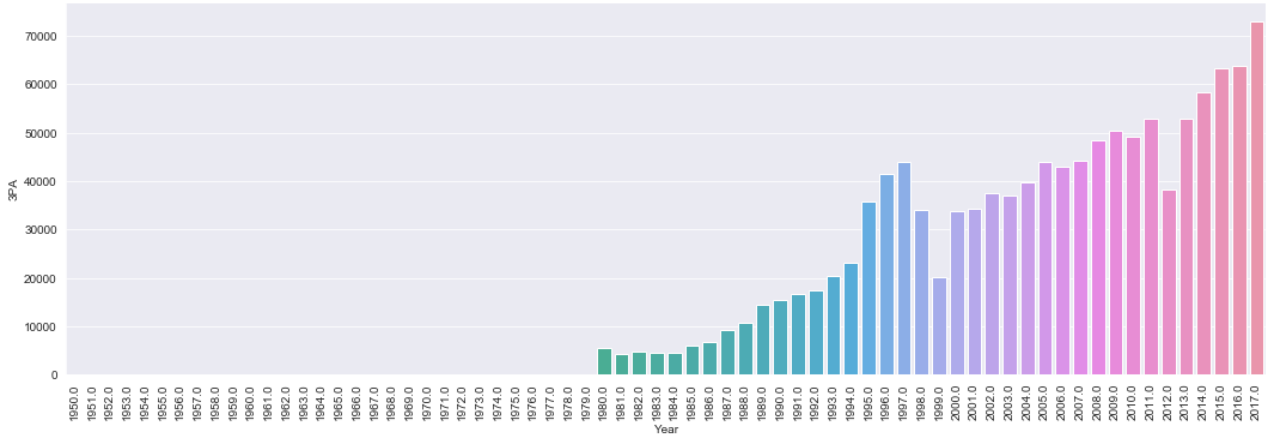
The NBA has been around since 1946. Since then, rules have changed, and we have seen players who revolutionize the game. Certain positions and types of players have seen their roles rise and fall over time. For example, the dunk was banned in the 1970s, as officials saw it as an unfair advantage. The three point field goal didn't exist until 1979, meaning players like Wilt Chamberlain never had an opportunity to shoot it from deep. Today, the game is as fast paced as it has ever been before. Basketball has continued to sustain high popularity, which has allowed us to witness some of the best athletes the sport has ever seen. With all of this, I want to examine just how the game has changed, and look at some players in particular.

I want to begin by visualizing how the game has changed. First, let's look at how the three point field goal has changed since its inception. To do this, we will look at the total number of Three-Point *attempts* per year.

In [295…

```
from IPython.display import Image
Image(filename="threePointByYear.png")
```

Out[295…



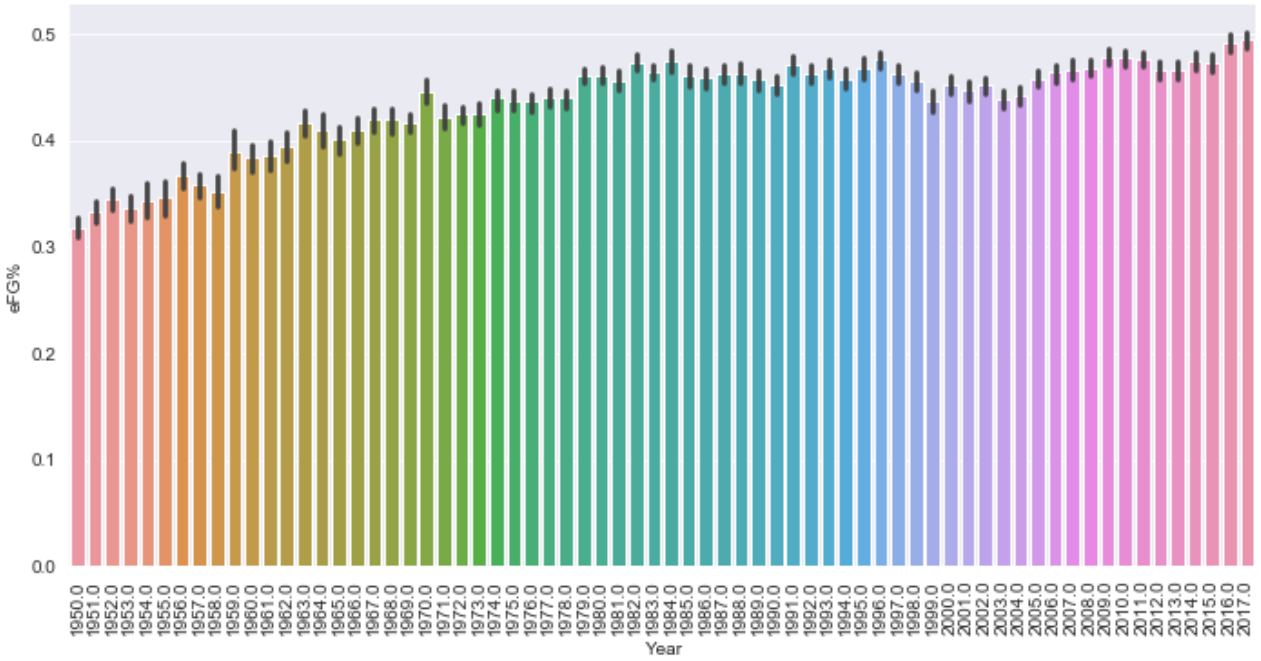
Of course, as mentioned earlier, the 3 point line was not added to the NBA until 1980, which is why we see the first attempt during that season. More importantly, we see a very obvious trend upwards since the inception of the 3 point line, with a few years that break out of the trend. Still, this is a very big implication in how the strategy of teams has changed over the years. It can be observed that at the beginning of the 3 point line, still not many players opted to shoot from deep. In fact, in 2017, over 70,000 threes were attempted, whereas in 1980, only about 5,500 threes were shot. Obviously, 1980 was the first year teams could shoot from deep, and so players were not accustomed to the change. Also, it should be noted that there were 23 teams in 1980, compared to the 30 in 2017. Still though, we have seen a huge jump in the amount of deep shooting in the NBA.

Now, let's look at a metric in our data that can help illustrate shooting effiency, eFG%. eFG stands for Effective Field Goal Percentage. This metric accounts for both three point field goals and total field goal attempts, which is important, because it accounts for threes. With this metric, a high-volume shooter who only shoots twos would have a higher eFG% than a shooter who scores the same amount, but only shoots threes. This will help normalize the data from before the three-point line existed.

Since this is a percentage, we can't just sum up the data by year, since having a percentage above 1 doesn't make sense. Instead, we'll take the average to see which year had the highest average eFG%.

In [296…

```
from IPython.display import Image
Image(filename="efgPlot.png")
```



This plot is a little bit messy, but it does show an overall trend upwards. In the early part of the NBA, we saw an average eFG% of around 35%. In 2017, however, we see the average eFG% was around 50%. Of course, the increase has been gradual, as can be seen in the trend above, but we can see just how much more efficient players have become.

Let's move on to using the api data, from https://www.balldontlie.io/api/v1/

This data is especially important because it let's us see each individual game since 1979. We have data on each season, and we have game data since 2003, but now we have a lot of game data with this api.

In [ ]:

```
import requests
#LeBron James' ID is 237. Let's look through his stats
lebronQuery = requests.get("https://www.balldontlie.io/api/v1/stats?&per_page=10
# we have a bit of an issue, as the api has pages with a limit of 100 per page.
# so to fix that, we need a list to hold all of the responses
responseList = []
for i in range(0,17):
```

```
    # 0 to 17, because range is up to not including
    allGameQuery = requests.get("https://www.balldontlie.io/api/v1/stats?&per_pa
    # use .format to use the i, since it increases by 1 in each iteration
    # and add the response to the list
    responseList.append(allGameQuery.json())
```

In [297…

```
# Let's get some of LeBron's key career totals
totalPoints = 0
totalRebounds = 0
totalAssists = 0

data = {}
for game in responseList:
    data = game["data"]
    for i in data:
        if i["pts"] == None:
            # some games return as None, so we will mark those as 0 to avoid get
            totalPoints += 0
        elif i["pts"] != None:
            totalPoints += i["pts"]
        if i["ast"] == None:
            totalAssists += 0
        elif i["ast"] != None:
            totalAssists += i["ast"]
        if i["reb"] == None:
            totalRebounds += 0
        elif i["reb"] != None:
            totalRebounds += i["reb"]
print("PTS: " +str(totalPoints))
print("AST: "+ str(totalAssists))
print("REB: "+str(totalRebounds))
# Note: this data includes playof games too
```

```
PTS: 43898
AST: 11813
REB: 12317
```

In [ ]:

```
mikeQuery = requests.get("https://www.balldontlie.io/api/v1/players?search=Micha
# Michael Jordan's ID is 2931, let's go ahead and compare the two players that a
# on top of their GOAT list.
mikeList = []
for i in range(0,17):
    # 0 to 17, because range is up to not including
    mikeGameQuery = requests.get("https://www.balldontlie.io/api/v1/stats?&per_p
    # use .format to use the i, since it increases by 1 in each iteration
    # and add the response to the list
    mikeList.append(mikeGameQuery.json())
```

In [298…

```
mikePoints = 0
mikeRebounds = 0
mikeAssists = 0
mikeData = {}
for game in mikeList:
    mikeData = game["data"]
    for i in mikeData:
        if i["pts"] == None:
            # some games return as None, so we will mark those as 0 to avoid get
            mikePoints += 0
```

```
        elif i["pts"] != None:
            mikePoints += i["pts"]
        if i["ast"] == None:
            mikeAssists += 0
        elif i["ast"] != None:
            mikeAssists += i["ast"]
        if i["reb"] == None:
            mikeRebounds += 0
        elif i["reb"] != None:
            mikeRebounds += i["reb"]
print("PTS: " +str(mikePoints))
print("AST: "+ str(mikeAssists))
print("REB: "+str(mikeRebounds))
```

```
PTS: 40336
AST: 7100
REB: 8369
```

Of course, it is hard to quantify greatness. But as we can see, LeBron leads Jordan in all of points, rebounds, and assists. He has played more seasons than Jordan, so that is certainly a big part of it.

Now let's look through another csv file. This one contains stats for each player from each individual game from about 2003 through 2019. This time frame is especially interesting for the NBA. A lot of players of note were drafted in the 2003 NBA Draft, including; LeBron James, Carmelo Anthony, Dwyane Wade, and Chris Bosh.

In [68]:
```
nbaGameDetails = pd.read_csv("games_details.csv")
```

In [69]:
```
# I am going to create a "Triple Double" column. Since we have seen so many trip
# especially recently, in NBA history

# first, let's define the conditions. To make things more simple, I will only be
# points, assists, and rebounds are in double-digits.
def tripDubConditions(seasonsStatsDF):
    if seasonsStatsDF["PTS"] >= 10 and seasonsStatsDF["AST"] >= 10 and seasonsSt
        return 1
    else:
        return 0
# now we will use the apply method
nbaGameDetails["Triple Double"] = nbaGameDetails.apply(tripDubConditions,1)
```

In [90]:
```
tripleDoubleDF = nbaGameDetails.groupby("PLAYER_NAME").agg({"Triple Double":sum}
tripleDoubleDF.columns = ["PLAYER_NAME","Triple Double"]

topFifteenTripDubDF = tripleDoubleDF.head(15)
topFifteenTripDubDF
```

Out[90]:

	PLAYER_NAME	Triple Double
0	Russell Westbrook	157
1	LeBron James	117
2	Jason Kidd	62

	PLAYER_NAME	Triple Double
3	James Harden	48
4	Nikola Jokic	44
5	Rajon Rondo	42
6	Draymond Green	33
7	Ben Simmons	30
8	Luka Doncic	21
9	Giannis Antetokounmpo	19
10	Elfrid Payton	17
11	Chris Paul	17
12	Kyle Lowry	17
13	Blake Griffin	13
14	Kobe Bryant	13

The table above shows the top 15 players in Triple Doubles. We can see that both Russell Westbrook and LeBron James have many more than anyone else on this list. A few things are interesting here. First of all, Westbrook debuted in 2008, so LeBron has about 4 years on him. Also, Nikola Jokic is the only primary Center on the list, and he comes in at 5th.

Let's see how the top 5 players in triple doubles have performed over their games in this time period

In [92]:
```
topFiveTripDubs = nbaGameDetails.query("PLAYER_NAME == 'Russell Westbrook' or PL
# we have 4971 games from these 5 players!
```

Out[92]:

	GAME_ID	TEAM_ID	TEAM_ABBREVIATION	TEAM_CITY	PLAYER_ID	PLAYER_NAME	S
89	21900898	1610612743	DEN	Denver	203999	Nikola Jokic	
124	21900900	1610612747	LAL	Los Angeles	2544	LeBron James	
270	21900891	1610612747	LAL	Los Angeles	2544	LeBron James	
299	21900892	1610612745	HOU	Houston	201935	James Harden	
300	21900892	1610612745	HOU	Houston	201566	Russell Westbrook	
...	...	...	...	...	...	...	
576341	11200025	1610612752	NYK	New York	467	Jason Kidd	
576366	11200028	1610612748	MIA	Miami	2544	LeBron James	
576420	11200019	1610612760	OKC	Oklahoma City	201935	James Harden	
576426	11200019	1610612760	OKC	Oklahoma City	201566	Russell Westbrook	
576690	11200008	1610612748	MIA	Miami	2544	LeBron James	

4971 rows × 29 columns

In [293…

```
# Let's see which of these players is the best scorer, passer, rebounder
# first, let's create a dataframe sorted by each player's points per game (avera
tripDubScorers = topFiveTripDubs.groupby("PLAYER_NAME").agg({"PTS":np.mean}).sor
tripDubScorers.columns = ["PLAYER_NAME","PTS"]
tripDubPassers = topFiveTripDubs.groupby("PLAYER_NAME").agg({"AST":np.mean}).sor
tripDubPassers.columns = ["PLAYER_NAME","AST"]
tripDubRebounders = topFiveTripDubs.groupby("PLAYER_NAME").agg({"REB":np.mean}).
tripDubRebounders.columns = ["PLAYER_NAME","REB"]
print(tripDubScorers)
print("*********")
print(tripDubPassers)
print("*********")
print(tripDubRebounders)
```

```
         PLAYER_NAME        PTS
0       LeBron James  26.854766
1       James Harden  24.463710
2  Russell Westbrook  22.985207
3       Nikola Jokic  16.955774
4         Jason Kidd  10.400460
*********
         PLAYER_NAME       AST
0  Russell Westbrook  8.188363
1         Jason Kidd  7.905639
2       LeBron James  7.193218
3       James Harden  6.143145
4       Nikola Jokic  5.326781
*********
         PLAYER_NAME       REB
0       Nikola Jokic  9.552826
1       LeBron James  7.502239
2  Russell Westbrook  6.899408
3         Jason Kidd  6.156502
4       James Harden  5.259073
```

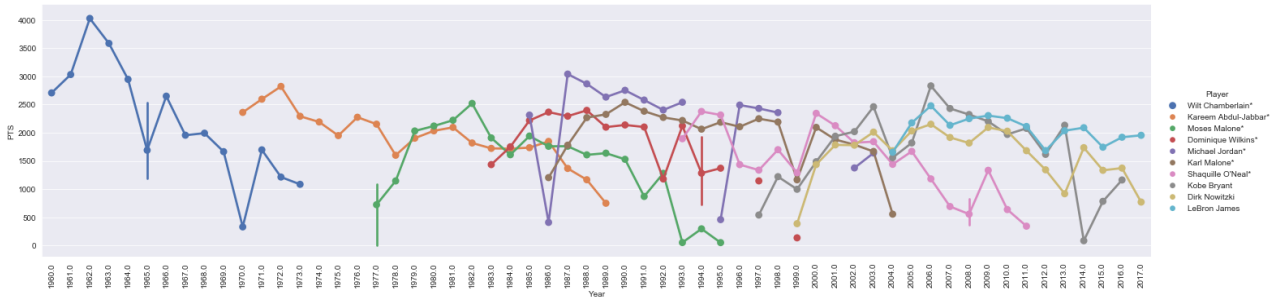
These results above don't really go against what we might expect. Jokic, the Center, leads in rebounds, and Kidd, the least aggressive of shooters, comes in last in points.

Who is the best scorer of all time? Best passer and best rebounder? Maybe some of our triple double guys will show up.

In [292…

```
from IPython.display import Image
Image(filename="topCareerScorers.png")
```

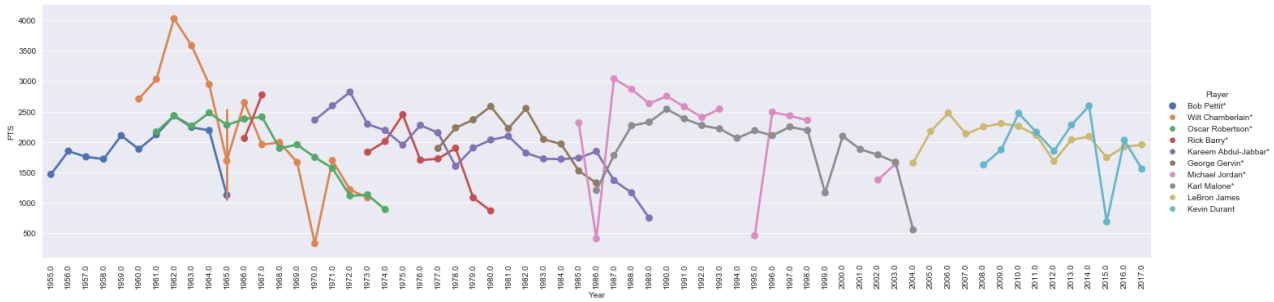
Out[292…



The above plot is a good representation of how individual scoring has changed over time. Of course, this is only the top 10 career scorers of all time, but it does show how each of those

players compare to each other over time. And as we can see, Wilt Chamberlain dominated in his time playing. His peak is much higher than anyone elses. Now, let's compare these top 10 to the top 10 points per game players.

In [290…
```
from IPython.display import Image
Image(filename="topAvgScorers.png")
```

Out[290…


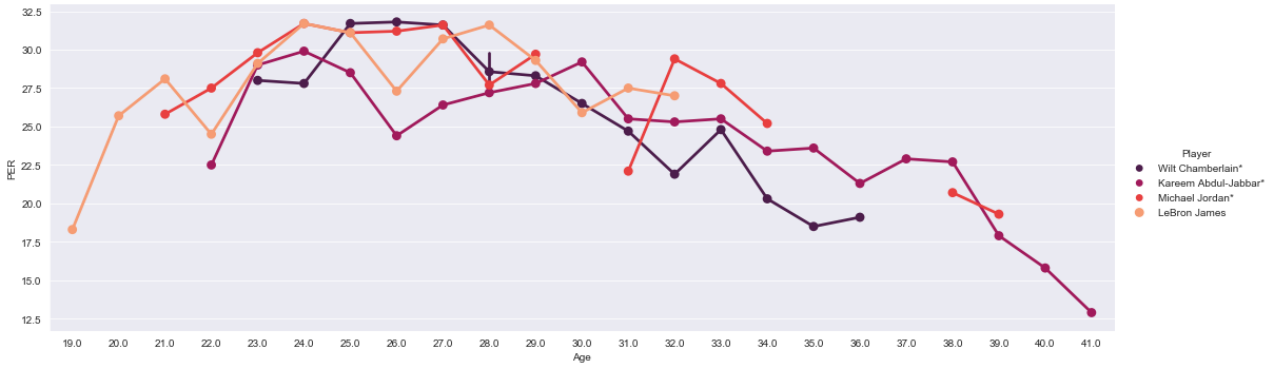
We can see the lists do not match up, in terms of players, but we do see some of the same names. Among those; Wilt, Kareem, Michael Jordan, and LeBron James. It's very interesting to see how certain players we clearly ahead of others during their specific era. For example, Wilt Chamberlain's huge peak is observed in both graphs and simply towers over other players in any era.

In [325…
```
topFourScorers = seasonsStatsDF.query("Player == 'Michael Jordan*' or Player ==
pd.options.display.max_columns = 200
sb.catplot(x="Age",y="WS",hue="Player",palette="rocket",aspect=3,kind="point",da
# have to remember to use the * next to players who are in the hall of fame
# So, these are our presumed potential GOATs, now lets put them all on one graph
# First, Win-Share
# Let's use Age as the x-axis, so that we can compare each player's physical pea
```

Out[325… `<seaborn.axisgrid.FacetGrid at 0x7fc6f5ba1f50>`

In [326…
```
# Player Efficiency Rating
sb.catplot(x="Age",y="PER",hue="Player",palette="rocket",aspect=3,kind="point",d
```

Out[326… `<seaborn.axisgrid.FacetGrid at 0x7fc6f69addd0>`



In [327…

```
# Points
sb.catplot(x="Age",y="PTS",hue="Player",palette="rocket",aspect=3,kind="point",d
```

Out[327… <seaborn.axisgrid.FacetGrid at 0x7fc6f69cee10>

The three above graphs are all measuring different things. The first shows each players Win-Share. This stat is key in showing how valuable each player is to their team. Sometimes, we see what we call 'stat-padders', who get easy rebounds or assists during blowout games, for example. Win-Share quantifies each players contributions to their teams.

The second graph shows Player Efficiency Rating. From Basketball-Reference.com: The Player Efficiency Rating (PER) is a per-minute rating developed by ESPN.com columnist John Hollinger. In John's words, "The PER sums up all a player's positive accomplishments, subtracts the negative accomplishments, and returns a per-minute rating of a player's performance." This one incorporates everything, scoring and beyond.

The third graph shows points, which are always flashy. You can see how each of these players had a rise and fall in their scoring volume.

All of these graphs share one thing in common, and that is that each of these players have a peak around age 24-26. We all hear about players 'prime' and all of these graphs, despite all representing completely different statistics, show these players general 'prime' is around 24-26. Even LeBron, who began playing 2 years prior to any other player here, shows the same trend.

So, of these four, who is the GOAT? Well, LeBron hasn't scored as consistently high as the other three, and his Win Share has been typically lower than the others - at least to this point - so we'll sort of disqualify him from being number one. We can see how good the longevity of Kareem Abdul-Jabbar was, as he played into his 40s with consistent numbers. It's tough to

analyze Michael Jordan, as he retired three times during his career, but we can only grade on when he played. Michael Jordan appears to be the best player around his early 30s, and he is right at the top or near it during each players prime. Wilt Chamberlain had the highest highs of any of these players, but also had some of the lowest lows and dropped off quite a bit. LeBron is still playing, and may pass over everyone. Still, the GOAT conversation includes more than stats, like championships and leadership ability. Kareem and Michael both have 6 championships, and since I'm the one doing the analyzing, I think we can go ahead and crown Michael Jordan as the GOAT, at least for now.

Overall, it's hard to say who is the greatest player of all time. Greatness is measured beyond statistics, but it has been interesting to look at how the NBA has changed over time and how players have changed their strategies. We saw how the introduction of a three-point line changed scoring, and how shooters have gotten better over time. We also were able to look at how a couple of players have been dominate in their own eras, like Michael Jordan and LeBron James.