Data Analysis Project

Michael EJIGU - Nhat Luan TRUONG 5ISS B1
(Academic year: 2020-2021)

Data has today become one of the most expensive resources. It provides answers to questions, and can help predict the future. Today we decide to use data to settle the biggest debate of our generation : Who between **Lebron James**, **Kobe Bryant** and **Michael Jordan** can be considered the best player of our generation?

To achieve our goal, we use R for the Data Analysis and Kaggle for our dataset (https://www.kaggle.com/xvivancos/michael-jordan-kobe-bryant-and-lebron-james-stats) This archive contains many datasets, but the one we focus on is the one with information about all the games of the players (allgames_stat.csv). This gives us Jordan's stats from 1984 to 2003, Bryant's stats from 1996 to 2006 and James' stats from 2003 to 2019. For each game we have information about the minutes played, the points scored, assists, rebounds etc..

To download the archive don't forget to create an account on Kaggle. It otherwise returns an error.

To run our script, all that needs to be changed is the location of the working directory (The absolute path to the RFiles folder). If you are on a non Windows System, please also change the \\ into /.
Also the libraries dplyr and ggplot2 are used.

---

1. Rookie year

First we compare the players performances in their first year of the NBA

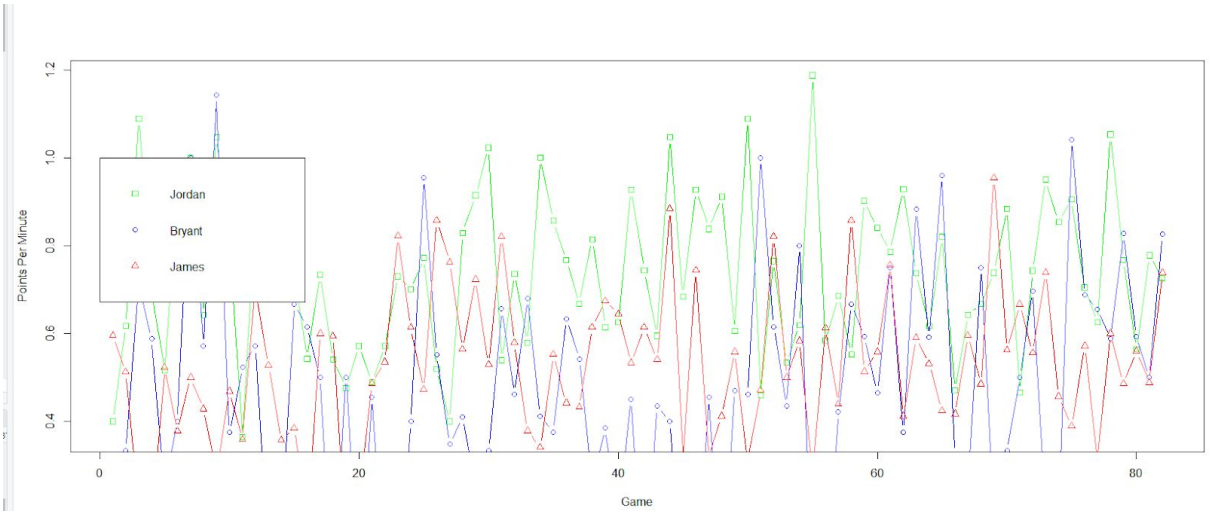

Fig 1 : Rookie year point per minute per game for each player

We plot the number of points per minute for each player for each game. The minutes need to be taken into account because they didn't get the same amount of play time in their rookie year.
Jordan (which is the green points) was overall the most performant. For all players, there is no consistency nor progress through the games of their rookie year.

```
[1] "Jordan Average Minute Per Game Rookie Year :"
> mean(jordanRookieGames$PTS/jordanMins)
[1] 0.7346455
> print("Bryant Average Minute Per Game Rookie Year :")
[1] "Bryant Average Minute Per Game Rookie Year :"
> mean(kobeRookieGames$PTS/kobeMins)
[1] 0.4592723
> print("James Average Minute Per Game Rookie Year :")
[1] "James Average Minute Per Game Rookie Year :"
> mean(jamesRookieGames$PTS/jamesMins)
[1] 0.5255443
```

Fig 2 : Average point per minute for each player

Jordan also had the highest mean.
Points : Jordan (3 pts), Bryant (1 pt), James (2 pts)

2. Progression Across the Years

We then look at how their scoring abilities improve through their first 15 years.

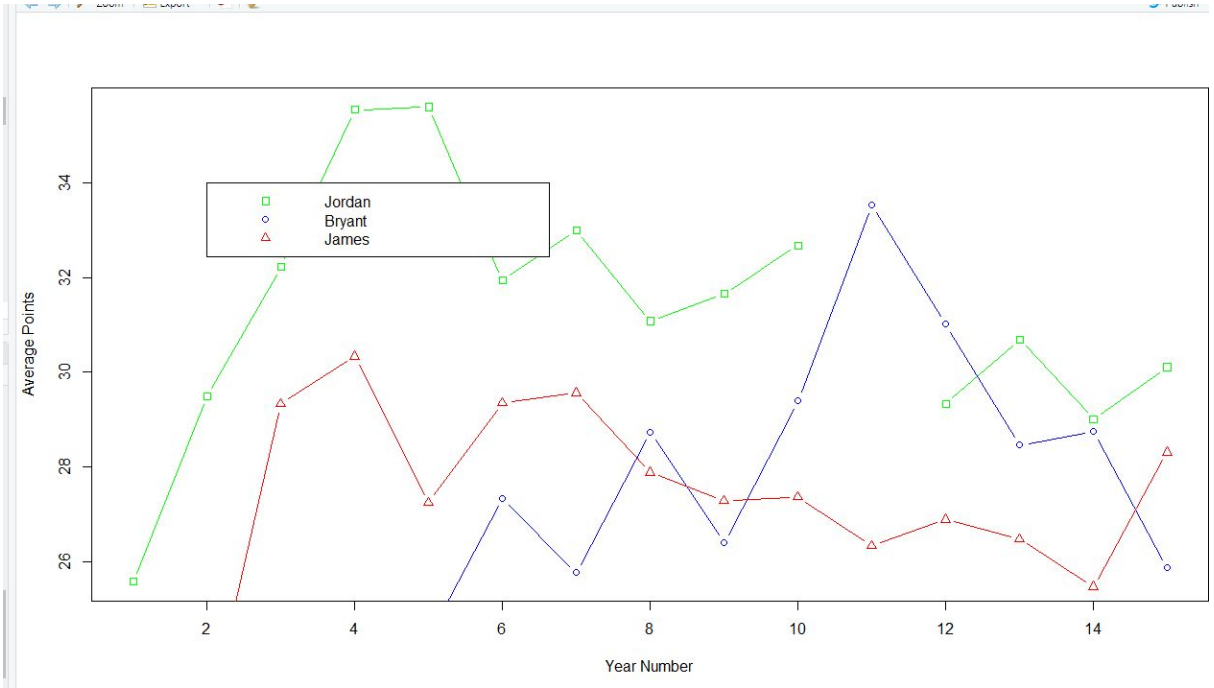

Fig 3 : Points per game average through the first 15 years for each player

Jordan here in green reaches his maximum potential in the first 5 years. Bryant takes more time to reach his maximum potential but still has a more than decent scoring ability. He also had less than 26 points per game in his first 5 years. James finally has a good scoring ability but does not compare to Jordan and Bryant in this category

Points : Jordan (3 pts), Bryant (2 pt), James (1 pts)

3. Adaptability to today's game: 3 point shot

Today the game of basketball has however changed. Big men used to be the stars of the NBA until teams realised the importance and efficiency of having a nice 3 point shot.
We look at how efficient the three point shots of our players are.

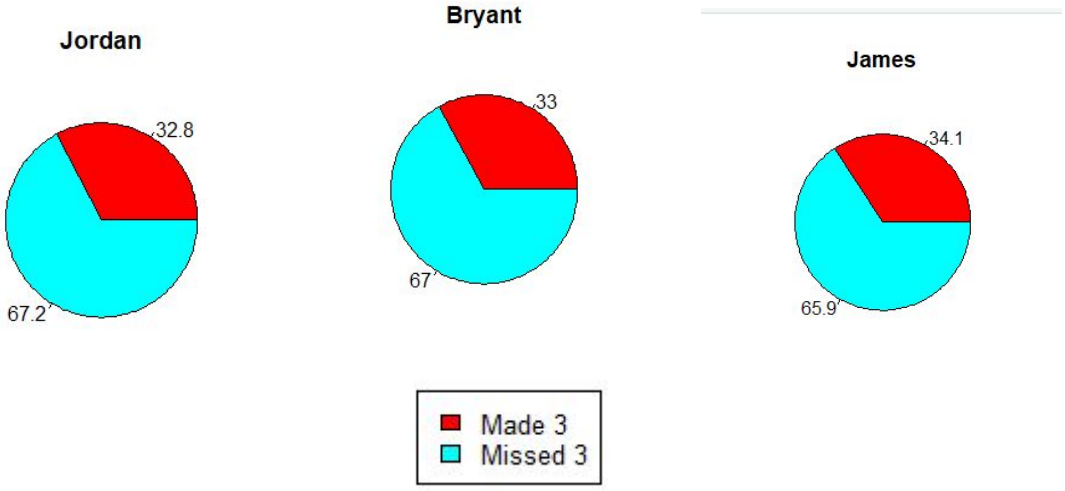

Fig 4 : Career 3 point success rate each player

James has the most efficient 3 point shot between the three. That is understandable since he had to adapt to today's game. 33 percent efficiency is still an excellent 3 points success rate. Coaches would happily tell any player with this efficiency to keep shooting the ball.

Points : Jordan (1 pt), Bryant (2 pts), James (3 pts)

4. Ability to make others better

There is no I in Team (even if there is one in Win). We then look at how these players made others around them better. Let's imagine player A makes a pass to player B, and player B scores. A has made an assist. We look at how the 3 players' assists progressed in their first 15 years.

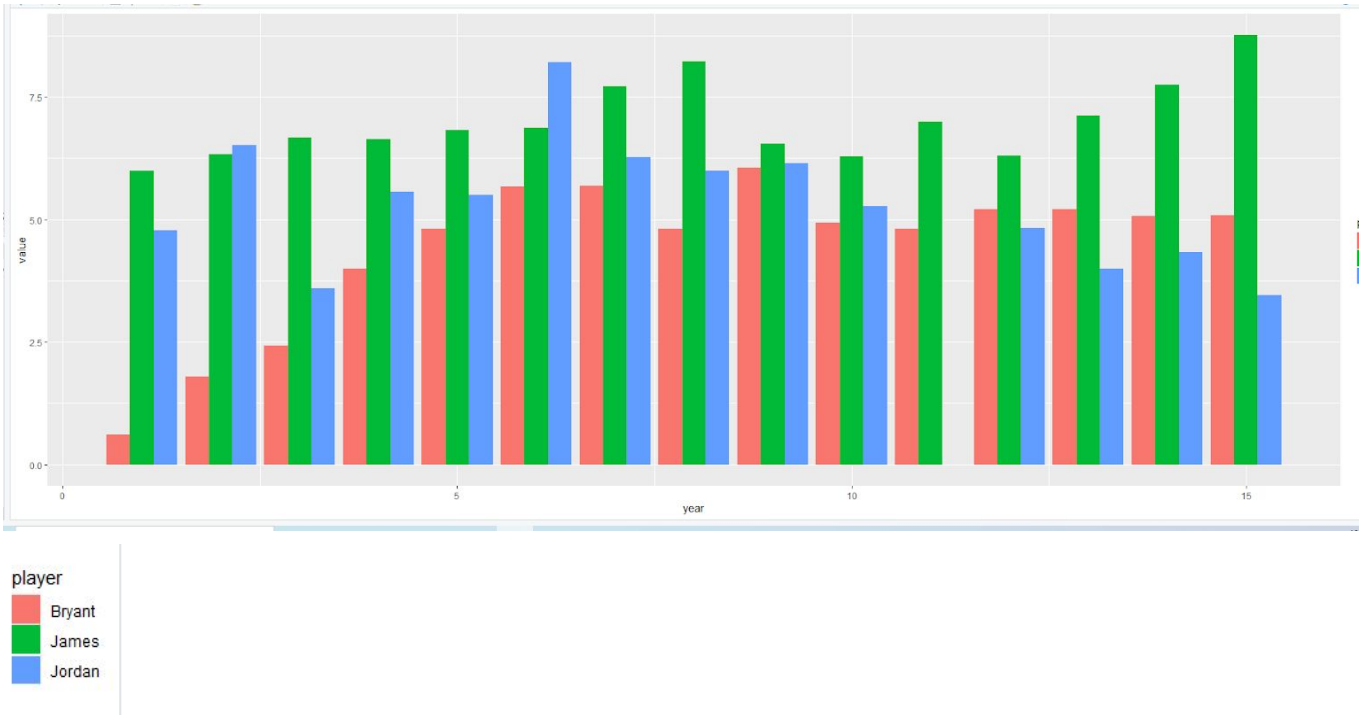

Fig 5 : Assist per game average through the first 15 years for each player

Bryant was never known for his passing. In fact he is actually known for never passing. He makes progress through his years but passing never really was his priority.
Jordan has decent assist statistics. However James has since his entry in the NBA been an excellent facilitator. He without a doubt has the highest game IQ and is the most rounded up player (good in all fields).
Points : Jordan (2 pts), Bryant (1 pt), James (3 pts)

5. **12th year rebounding**

When a player is as good as any of the three, there is a big risk that that player will lose his work ethic and put less effort into the game. We therefore look at how they rebounded (get the ball back after a shot) in their 12th year (well after everyone knew they were excellent)

```
> print( Jordan 12th year rebound :
[1] "Jordan 12th year rebound :"
> jordanReboundGames=select(filter(
> jordanReboundLastYear=mean(as.num
> jordanReboundLastYear
[1] 5.048387
> print("Bryant 12th year rebound :
[1] "Bryant 12th year rebound :"
> kobeReboundGames=select(filter(ga
> kobeReboundLastYear=mean(as.numer
> kobeReboundLastYear
[1] 4.714286
> print("James 12th year rebound :"
[1] "James 12th year rebound :"
> jamesReboundGames=select(filter(g
> jamesReboundLastYear=mean(as.nume
> jamesReboundLastYear
[1] 8.794118
>
```

Fig 6 : Average rebounding in 12th year for each player

James has the highest rebound average in his 12th year. This not only again proves he is the most rounded up player, but that his work ethic has remained excellent.
Points : Jordan (2 pts), Bryant (1 pt), James (3 pts)

Total :

- **Jordan: 11 points**
- **Bryant: 7 points**
- **James: 12 points**

To summarize, James comes out ahead in this statistical analysis of these 3 players. Even though each of these players had a unique style of play and comparing them should not be allowed as they were each the best at their peak, data is a useful tool to make some observations (especially with large amounts of data). R is a very intuitive language, which made this task easier. However it is evident that optimization can be done on our code.