Luke Maschoff
INFO 2201

Final Writeup

In this project, the overarching question I was trying to answer is how many NBA players have won MVP, an NBA championship, placed Top 20 in one of 3 popular basketball statistics in either the playoffs, or regular season (all time), and then also have been one of the Top 5 paid players in a given season? I really wanted to see if these players have been paid accordingly for how good they have played. This is a ton of data comparison, so I had to break it up into multiple data frames, functions, and loops to get me to my end answer. To start, I first read in a CSV with all the MVPs from the NBA (https://www.basketball-reference.com/awards/mvp.html), and created a Pandas dataframe with a few of the more important statistics about the player including the name, age, games played in his NBA season, team, and year that he won the MVP award. With this data I was able to mess around and see information such as the oldest, and youngest players to win MVP. I also visualized this data in multiple ways, like printing out the DF, and also creating a graph of the games played per age of MVP:
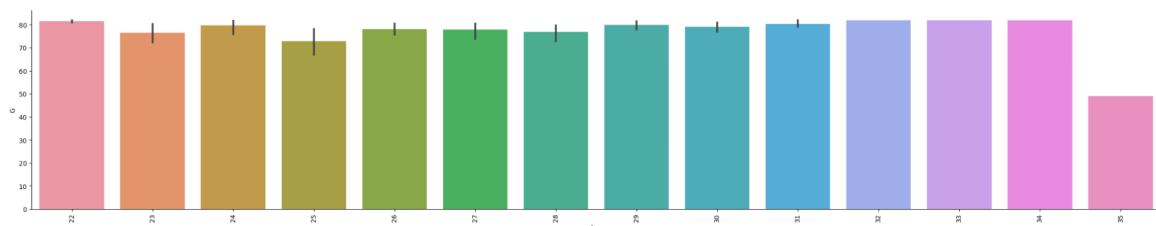
After this, I created another table with the MVP's in game statistics such as MP, PTS, TRB, AST, and FG%. From this table I was able to calculate the average for all these statistics, across all MVPs and found out that the average MVP will average 38.9 minutes per game, 26.1 points per game, 12.2 total rebounds per game, 5.6 assists per game, shooting 50.8% from the floor, and the average age of an MVP is 27. This was pretty fascinating to learn and I hope to use this information next NBA season to see how close the next MVP came to having these statistics.

After learning about the MVPs, I loaded in another CSV (https://www.basketball-reference.com/leagues), but this time with every NBA championship winning team, and the year they won and created another pandas DF. A hiccup I ran into here was that in my MVP CSV, the team names were abbreviated, but in the championship winning CSV, the names were not. To fix this, I replaced every abbreviation in the MVP DF with the actual team name so it would match up with the championship CSV. After I did this, I was able to use pd.merge to merge my MVP table and my championship table to see what players won an MVP, and a championship in the same year. It was pretty fascinating to see that only 23 players have ever accomplished this feat, but I wasn't done just yet.

I created a function that allows a user to have 2 inputs. These two inputs are whether the user wants to search for the Top 20 players in a given statistic all time in the playoffs, or all time in the regular season. The second input is the statistic the user wants to search for. The function formulates a URL for the NBA.com API. Once the user enters these, it will display a table with

the Top 20 players in the statistic, and also display a graph with the amount of games each player on the table played to hit the given statistic, and also a table with the players who are on the Top 20, who have also won an MVP, and an NBA championship in the same year. After this, it was pretty clear these players are the cream of the crop when it comes to royalty in the NBA, but have they been compensated for their greatness?

To find this out, I created a while loop to search through a website called HoopsHype (https://hoopshype.com/salaries/players/), which has every NBA players salary dating back to 1990. I was able to use BeautifulSoup and Requests to figure out a theme on the website to be able to scrape the Top 5 players paid in each season up until the 2021-2022 season. I also took the Top 5 adjusted salaries from each season, and the Top 5 salaries from each season to create graphs and help visualize the data:


```
In [29]:    1
            2   # this displays a chart of all the adjusted salaries, and their correlating year
            3   sb.catplot(x="Salary", y="Year", aspect=5, height=7, data=salaryOnlyDF, kind="bar").set_xticklabels(rotation=90)
            4
            5
            6
Out[29]: <seaborn.axisgrid.FacetGrid at 0x7fa0d87c7a00>
```


With this information, I was able to see what players who met all the criteria from earlier were on this list, and the results were pretty shocking. There were very few players who met all the criteria, who were also once in the Top 5 paid players at one point. The data varied based on what stat the user searches for, but there were only about a maximum of 5 players. There are two different things this data could tell us. 1: These players we see in our final dataframe are the best basketball players of all time, in their given category whether that is scoring, assisting, or shooting from beyond the arc. 2: There are dozens of players who have accomplished winning an MVP and a championship in the same year while also being dominant in their given statistical category while not being compensated accordingly.

If I were to continue working on this project, I think I would try to dive more into the salaries, and maybe try to find data on what caused an increase in average adjusted salaries per season. To do this, I would try to find data on viewership for the NBA per season, and cross reference this to what seasons had higher average adjusted salaries.

Much of this project differed from my initial project write up. The similarities include gathering NBA player data, statistics, salaries, and using similar data types, and sources. The differences include finding MVPs, not players of different height, and also not finding the ideal NBA player, but finding the best NBA players. I changed the project up a bit because I was unable to find a way to gather different NBA players heights without manually downloading many different CSV files which defeated the purpose of using code.

I had a great time working on this project, especially with the ongoing NBA playoffs and really learned a ton and it was super awesome being able to pick whatever I wanted to do to work on, as it made it so interesting, and also super fun to work on. If I had one recommendation for this class is to keep the same exact format for the final project. I truly appreciated having so much freedom, and genuinely looked forward to working on this project every time I opened my laptop. Thank you for the great year in INFO 2201, and I hope you have a great end to the semester, and an even better summer.