NBA Player Salary Analysis:
SI 330 Final Project Report

Motivation

For my final project, I decided to look at data from the NBA 2017-2018 season. Specifically, I look at two data set that contained information about each player's statistics from that season, and another that had information about players salaries. I also used an extra third data set that held information on the number of wins and losses each team had in the season to answer one of my research questions.

I was motivated to analyze NBA data because I have a passion for sports. I have always been interested in data analytics and the impact that it can have on sports at every level. I searched for data on a few of the sports that I'm most interested in (soccer, football, basketball) and ultimately found the data for the NBA the quickest.

Once I knew exactly which data sets I planned on working with (player statistics and player salaries), I came up with related research questions that I was genuinely curious about answering.

Research Questions:

1. Do NBA players that receive higher salaries have better statistics during a season?
2. Question 2: How does a players age impact how much they make, on average?
3. Does a teams success during a season correlate to the money that the team spends on its players?

Data Sources

I used three datasets for my project. Two of them I got using the pandas read HTML function, and the other I downloaded a CSV file and added to my working directory.
The primary dataset I used I found on Kaggle and can be found at this hyperlink. I downloaded the data from this source as a CSV file. This dataset had information on player statistics from the 2017-2018 season. This dataset included about 30 different columns, but the ones that were important to my analysis include points per game, rebounds per game, assists per game, blocks per game, and steals per game. These statistics were easy to analyze and manipulate because they were all numeric values.

The second data set I used was from a website called “Basketball Reference” and can be found at this hyperlink. I retrieved this data using pandas to read the HTML of the website and locate the table that I needed to analyze the data. The important values in this dataset were the player names because that is what I merged the two tables on, and the player's salaries because that statistic is what all of my research questions are based on.

The third extra dataset I used was necessary to answer research question #3. I needed information on how successful each NBA team was during the 2017-2018 season. I also used the information on “Basketball Reference” and that information can be found at this hyperlink. I retrieved the data from this source using the pandas read HTML function and then locating the information I needed. For this source, I had to get information from two tables on the website and concatenate them. The important variables in this dataset were the player name because I used this to merge the information, and the win column so that I had a metric on how successful a team was during the season.

Data Manipulation Method

In order to make the python code work with accurate findings, I had to manipulate all three of the datasets. To explain what data I manipulated in this section, I will first talk about what was required to merge the two primary datasets, and then how I merged the data in my third research question.

Merging Primary Datasets

When first joining the player statistics table with the player salary information, the main problem I encountered was that there were player names that were repeated in both datasets. Players that were traded during the season had multiple columns that represented their statistics with each particular team. I solved this by dropping duplicate rows and keeping the first row for each player; this row usually possessed data for the team that the player spent the most games playing for. After this, I merged that salary table with the player information table using an inner merge. This was because I wanted to make sure all players matched with statistical data otherwise my analysis wouldn’t work. I also dropped columns that were irrelevant because I only wanted to have access to the columns that would aid my analytical endeavors. The code for this process can be found in the appendix.

Question 3

To answer my final question, I used a third dataset with information on each team’s season standings. I faced a couple of obstacles when manipulating this data. First, the tables of information I retrieved were separated by conference (eastern and western). To

concatenate these two tables I had to change the column names containing the team name to just say “team” and then put them together. The next challenge I faced was being able to merge this third dataset with my first one. The team names were in a different format and I had to use regex to clean the data in that column to allow for a clean merge. I looked up and got information on how to do that from here and here. I then had to use a groupby on my player statistics dataset. I grouped by the team to get information on the mean salary for each team during the 2017-2018 season. My next and final step was to make the team names the indexes of the two columns to merge them easily. The code for this process can be found here.

Analysis and Visualization

Once I had compiled and manipulated the data, the last step was to analyze it and create visualizations that showed my findings.

Question 1

To answer the first question, the data was already cleaned but I had to create a new column to use the pandas groupby function on. I created a function that allowed me to classify the amount of money players make as either a small, medium or large salary. This analysis allowed me to get the answers I needed for my first research question. The code is in the appendix.

I opted to create a bar graph that showed how the amount of money a player makes, on average, impacts how well they perform statistically in a game. When creating this visualization I basically wanted to see if the money teams spend on players was worth it. To do this I used matplotlib and took the values from the pandas data frame and converted each row representing a statistic for a category of player salary. I learned how to do this here. To make the bar graph look appropriate, I had to create a title, set an x and y label, and xstick labels for the visualization to make the most sense. As a result of my analysis, I found that higher paid players do actually have better statistics than the medium and lower paid categories of players. In fact, the amount of money a player made directly correlated with how many statistics they put up, on average, for all 5 categories that I used. The visualization is in the appendix as well.

Question 2

The process for answering the second question was similar to that of the first. I had to create a helper function that would classify each player into a particular age group. I then had to use groupby on those columns to create a data frame that I could answer my research question with. The code for this can be found here.

I was looking at how a players age impacted how much money they make on average. I wanted to see if players in their prime, veterans, or young players made the most money in the league. I hypothesized that it would be players in their prime because they usually sign big contracts. However, I discovered that veteran players make the most followed by players in their prime and then young players. Although I had set the ages to make their resulting groups equal in size, I tried various age ranges and the distribution remained the same. For this graph, I also decided to create a bar chart where the bars represent how much each category's salary is. I created it using the pandas function ".plot" which works with matplotlib and allowed me to easily create and label the bar graph.

Question 3

In question 3, most of the data analysis was performed before I merged the datasets and that process can be found in the Data Manipulation section. I wanted to determine if how much a team spent on their players impacted their performance. I was looking for a correlation between a teams success during a season which I measured by how many wins they received, and the average salary that teams spend on their players. To create the scatter plot for this section I used the pandas .iterrows() function to extract the information that I needed to create the plot. I also decided to use numpy to show the correlation coefficient which would help show if the correlation was significant. It turned out that there was essentially 0 correlation between the amount of money teams spend on their players and a teams success during that season. The correlation coefficient was **.014,** and the plot clearly appears scattered and random.

Appendix

Merged Data Code

```
## Function to merge the two datasets I am using
def mergedfs(link1, link2):
    df1 = pd.read_csv(l1)
    df2 = pd.read_html(l2)
    df2 = df2[0]
    ## There are some players that got traded so the data tracked will be with the team
    ## a player was on for the longest time
    df2 = df2.drop_duplicates(subset='Rk', keep='first')
    comp = df2.merge(df1, how='inner', left_on='Player', right_on='NAME')
    comp = comp.drop(['Rk', 'POSITION', 'NAME'], axis = 1)
    comp = comp.drop_duplicates(subset = 'Player', keep='first')
    return comp
```

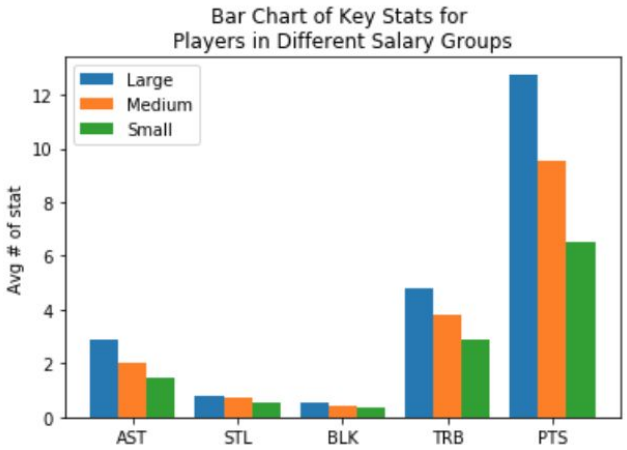
Question 1 Code

Question 1: Do NBA players that receive higher salaries have better statistics during a season?

```
## Set up numbers trying to make each group size as even as possible
def q1help(df):
    if df > 8350000:
        return 'Large'
    elif df > 2700000:
        return 'Medium'
    else:
        return 'Small'
```

```
q1 = comp.copy()
# Create new column that catagorizes player salary as small/medium/large
q1['Sal Size'] = q1['SALARY'].apply(q1help)
# Get only relevant information
q1info = q1[['AST', 'STL', 'BLK', 'TRB', 'PTS', 'Sal Size']]
# Make columns numbers so they can be manipulated
q1info[['AST', 'STL', 'BLK', 'TRB', 'PTS']] = q1info[['AST', 'STL', 'BLK', 'TRB', 'PTS']].apply(pd.to_numeric)
# Group by thte new column to get information on each salary range
inf = q1info.groupby('Sal Size').mean()
inf
```

Question 1 Visualization

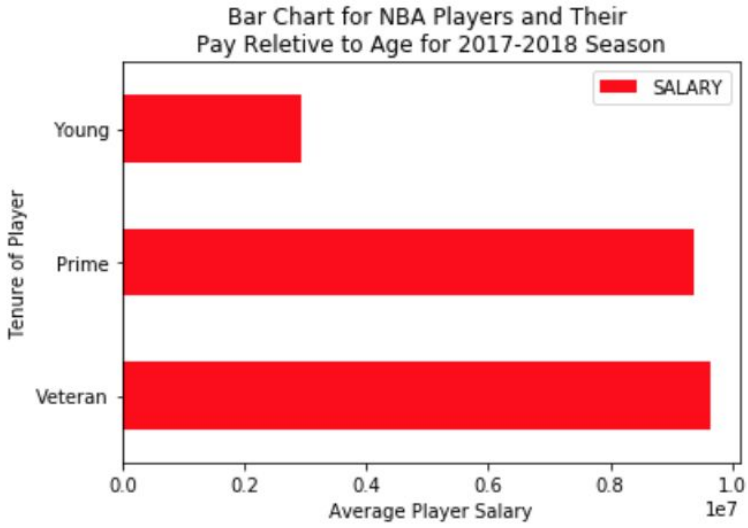


Question 2 Code

```
def q2help(df):
    if int(df) > 33:
        return 'Veteran'
    elif int(df) > 24:
        return 'Prime'
    else:
        return 'Young'
```

```
q2 = comp.copy()
# Apply age catagorizing function to create new column
q2['Age Group'] = q2['Age'].apply(q2help)
# Only get relevant information
q2info = q2[['Age Group', 'SALARY']]
# Group by the newly created column
q2df = q2info.groupby('Age Group').mean()
# Change column type to make it manipulatable
q2df['SALARY'] = q2df['SALARY'].astype('int64')
# Sort the salaries to find the top ones
q2ans = q2df.sort_values('SALARY', ascending = False)
```

Question 2 Visualization



Question 3 Code

```
l3 = 'https://www.basketball-reference.com/leagues/NBA_2018_standings.html'
# Get correct tables and change column names to easily concat
df1 = pd.read_html(l3)[0]
df1 = df1.rename(index=str, columns = {'Eastern Conference': 'Team'})
df2 = pd.read_html(l3)[1]
df2 = df2.rename(index=str, columns = {'Western Conference': 'Team'})
stand = pd.concat([df1, df2])
# Only get relevant information
stand = stand[['Team', 'W', 'L']]
# Sort by number of wins
sorted_standings = stand.sort_values(by = 'W', ascending = False).reset_index()
```

```
# Change column to have no digits and extraneous characters
sorted_standings['Team'] = sorted_standings['Team'].apply(lambda x: re.sub(r'\W+', '', x))\
                                                   .apply(lambda y: re.sub("\d+", "", y))
```

```
# Get only relevant informaiton from original df
q3 = comp[['TEAM', 'SALARY']]
# Clean data from original df
q3['TEAM'] = q3['TEAM'].apply(lambda x: re.sub(r'\W+', '', x))
# Group by newly cleaned column
q3sal = q3.groupby('TEAM').mean().sort_values('SALARY', ascending = True)
ss2 = q3sal.sort_values('TEAM').reset_index()
# Reset index for visual purposes and drop extraneous columns
ss = sorted_standings.sort_values('Team').reset_index().drop(columns=['level_0', 'index'])
# Merge the two dataframes, drop extraneous columns, and reset index again
fin = pd.merge(ss, ss2, left_index=True, right_index=True).drop(columns=['TEAM', 'L'])\
                .sort_values('W', ascending=False).reset_index().drop(columns=['index'])
```

Question 3 Visualization

```
The correlation coefficient is 0.014466382420785287
```