Final Project Report
ANLY511
Due: Dec 6, 2021
Jack Piccione, Matt Harding, Matt Young, & Scott Johnson

# Introduction

Sports are an integral part of popular culture in the United States, where fans emotionally invest in the performance of their favorite teams and players. One of the most popular sports leagues in the United States is the National Basketball Association (NBA). The NBA is a professional basketball league based in North America composed of 30 teams. The league was founded in 1946 with only 11 teams, where the league encountered financial troubles. Over time, the league grew in popularity and became a majorly profitable business; with the total revenue of the NBA being upwards of $8 billion. Moreover, players are making up to $43 million annually. Because of the extensive US and global popularity among fans and economic implications due to immense revenue streams, the goal of this study is to understand the factors influential in increasing the winning percentage of teams in the NBA. To accomplish this goal, an analysis on the specific factors of in-game actions, salaries, fan attendance, and outcome of previous seasons will be conducted.

The first factor of interest is whether or not a higher three-point score percentage leads to more wins for teams. The three-point shot was not always popular when the league first began. However, over time players started to perfect their shot and had great success in shooting three pointers. Due to this success, the league decided to move the three-point line backwards. Fans love three-point shots, as they are likely the deciding factor at the end of a close game. If a team is trailing by two points at the end of a game, they will try to score a three-pointer with only a few seconds remaining. Additionally, three-point shots are a great way to reduce the gap in score between teams. If the leading team is making two-point goals in every play, the gap will be reduced by one every time the trailing team makes a three-point goal. On the other hand, three-point shots are much more difficult to score, and therefore, relying too heavily on three-point scores could be detrimental to a team’s success. Because of this dichotomy between difficulty in score and higher point yield, the analysis of team success and three-point percentage, the results could be useful in determining future strategies for player in-game actions.

Balancing the salaries across players of a given team is one of the most difficult but important things a team has to do each and every season. Teams are attempting to pay their best players enough money to keep them from leaving to play for another team while also trying to leave enough money to pay other players to surround them and create a team that has as many strengths as possible. Teams have to abide by a salary cap in the NBA. This means that there is a maximum amount of money a team can spend in a given season. While this amount changes from season to season, it is the same for each team during a given season. However, there is a way around the salary cap to spend more money. By paying a tax to the league based on how much over the salary cap a team spent, a team may surpass the limit set by the salary cap. In general, some teams tend to spend over the salary cap in a given season while others may spend significantly less than the allotted amount. An example of spending during a given season can be seen in *Figure 1*. Given how salaries are distributed in the NBA, it would be useful both from a business perspective and a team success standpoint to understand if and how salary distribution on a given team affects the overall success of that team.

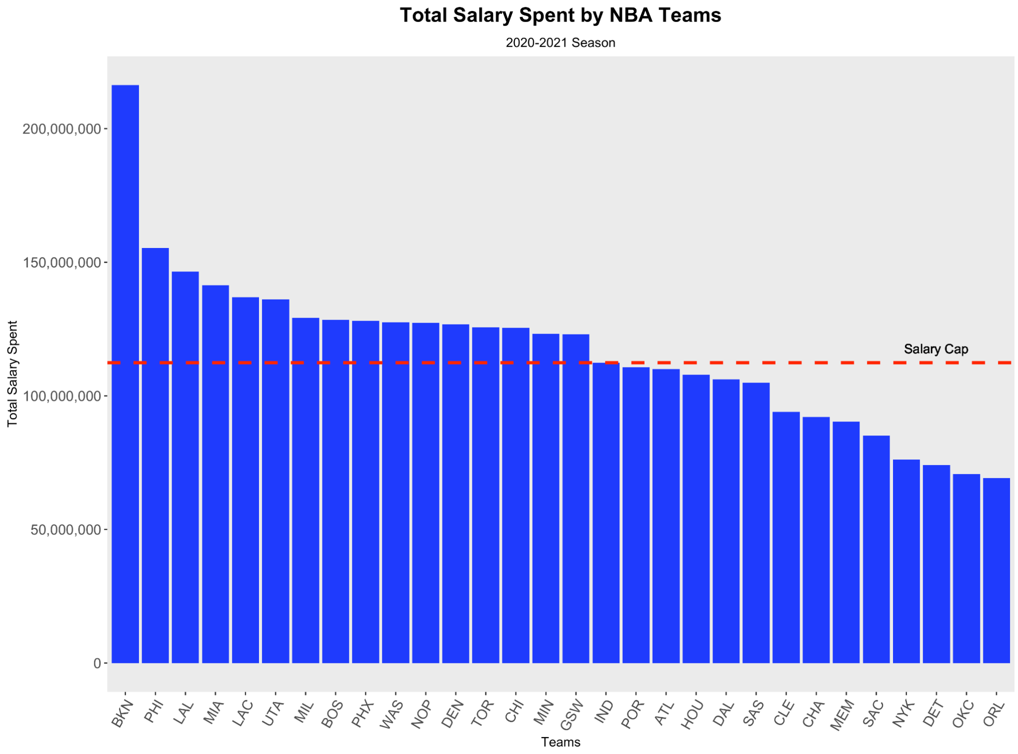


*Figure 1: Total money spent by teams in the 2020-2021 season*

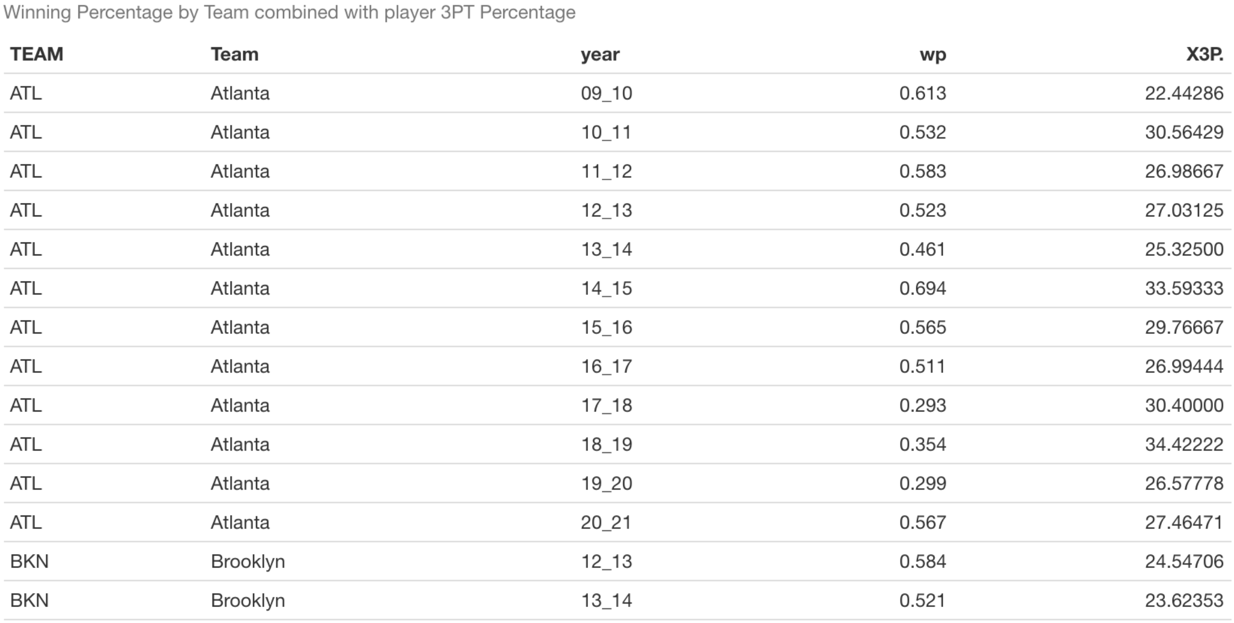
Fans are crucial to making an exciting and electric atmosphere but do they make a difference when it comes to the outcome of the game in the regular season? Playing on the road during a long NBA season can be a drag. Oftentimes players are in a different time zone and have spent hours on a plane when they have to play an away game. These factors can result in a worse performance on the road.  Not to mention there are thousands of opposing fans cheering their team on. Does the number of fans at a game lead to more wins for the home team? By answering this question, the impact of the number of fans at a game will be quantified.

The final factor of interest is how the outcome of past seasons influences the winning percentage of the seasons following. To address this factor of the outcome of previous seasons, the following data science question was posed: Does winning an NBA championship have an effect on the winning percentage of the following season? Today, in the NBA, teams play a total of 82 regular season games spanning from October to April. However, if a team is fortunate enough to qualify for the playoffs and make it to the NBA finals, this extends their season into the month of June; two months after the majority of other teams have concluded their season. Because of this, we sought to determine if this extended season had any affect on the following season for a team, as they have less time to recover and prepare for the upcoming season.

# Methodology

## In-Game Actions

Player three-point percentage data was manually copied from www.nba.com/stats/players/traditional/ into an excel workbook. Attributes of the data include player, team, age, games played, wins, losses, average minutes played, average points, field goals made, field goals attempted, field goal percentage, three-pointers made, three-pointers attempted, and three-pointer percentage. The data was then merged and reduced to only represent the player, games played, three-pointers made, three-pointers attempted, and three-pointer percentage. Next, using a dataset from www.teamrankings.com/nba/stat/win-pct-all-games, the win percentages from each team in a given season were matched to a corresponding team.



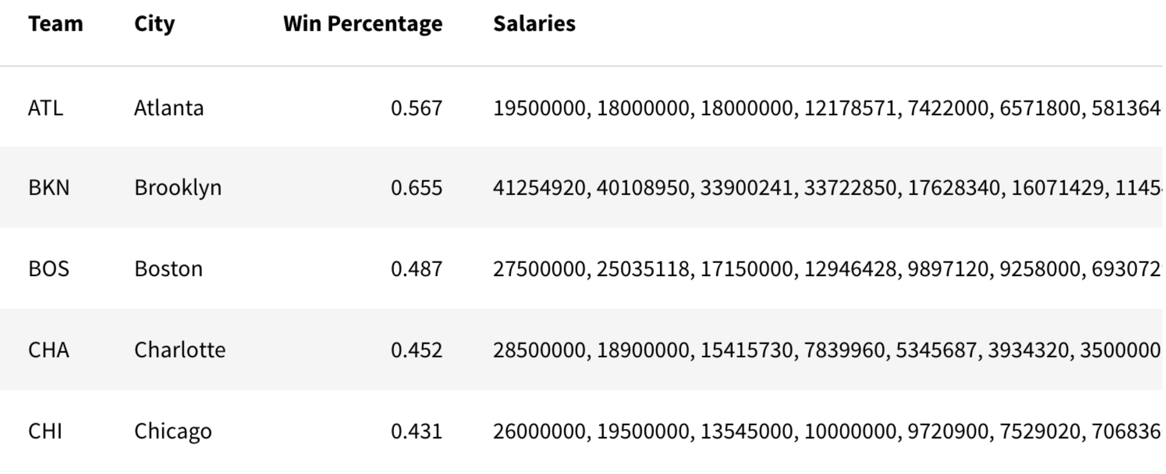
Winning Percentage by Team combined with player 3PT Percentage

| TEAM | Team | year | wp | X3P. |
|---|---|---|---|---|
| ATL | Atlanta | 09_10 | 0.613 | 22.44286 |
| ATL | Atlanta | 10_11 | 0.532 | 30.56429 |
| ATL | Atlanta | 11_12 | 0.583 | 26.98667 |
| ATL | Atlanta | 12_13 | 0.523 | 27.03125 |
| ATL | Atlanta | 13_14 | 0.461 | 25.32500 |
| ATL | Atlanta | 14_15 | 0.694 | 33.59333 |
| ATL | Atlanta | 15_16 | 0.565 | 29.76667 |
| ATL | Atlanta | 16_17 | 0.511 | 26.99444 |
| ATL | Atlanta | 17_18 | 0.293 | 30.40000 |
| ATL | Atlanta | 18_19 | 0.354 | 34.42222 |
| ATL | Atlanta | 19_20 | 0.299 | 26.57778 |
| ATL | Atlanta | 20_21 | 0.567 | 27.46471 |
| BKN | Brooklyn | 12_13 | 0.584 | 24.54706 |
| BKN | Brooklyn | 13_14 | 0.521 | 23.62353 |

*Table 1: Cleaned Player 3PT Data*

## Salaries

Player salary data was manually copied from https://hoopshype.com/salaries/players/ into an excel workbook. This data consisted of the player, the team, and the player's salary for a given season. The data was then aggregated in such a way that it combined each player on a team in a given season into a row that kept the player's salaries in a list in the data frame along with the team and the season. Then using a dataset from https://www.teamrankings.com/nba/stat/win-pct-all-games, the win percentages from each team in a given season were matched to the corresponding team. The data gathered for in-game actions analysis was found at https://www.nba.com/stats/players/traditional/ where it was cleaned and aggregated in the same way. A sample of what the cleaned data looks like can be found in *Table 2*.



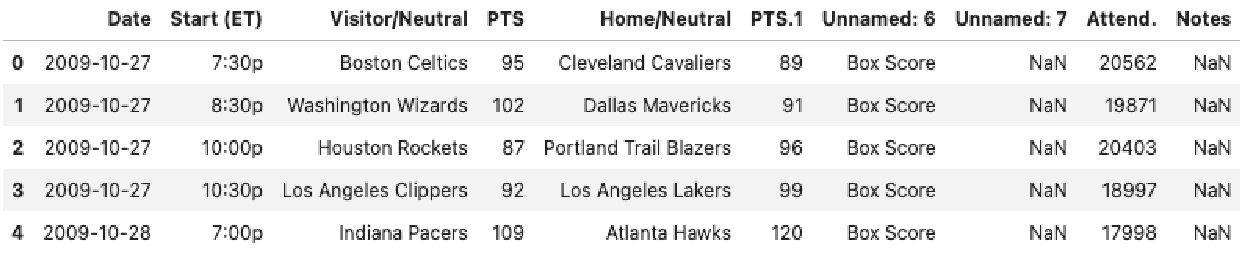
| Team | City | Win Percentage | Salaries |
|---|---|---|---|
| ATL | Atlanta | 0.567 | 19500000, 18000000, 18000000, 12178571, 7422000, 6571800, 581364 |
| BKN | Brooklyn | 0.655 | 41254920, 40108950, 33900241, 33722850, 17628340, 16071429, 1145 |
| BOS | Boston | 0.487 | 27500000, 25035118, 17150000, 12946428, 9897120, 9258000, 693072 |
| CHA | Charlotte | 0.452 | 28500000, 18900000, 15415730, 7839960, 5345687, 3934320, 3500000 |
| CHI | Chicago | 0.431 | 26000000, 19500000, 13545000, 10000000, 9720900, 7529020, 706836 |

*Table 2: Cleaned Player Salary Data*

## Fan Attendance

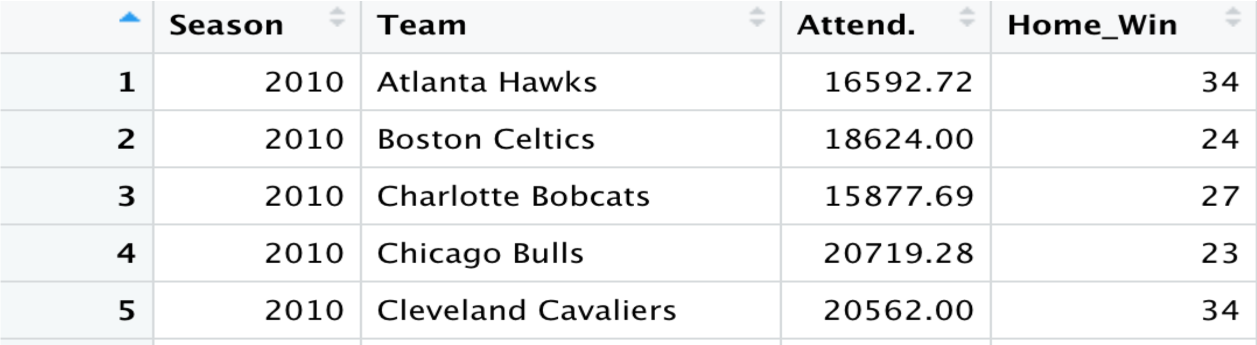
### *Data Gathering and Cleaning*

NBA attendance data was scraped from basketball-reference.com using Python. The dataset initially consisted of box scores from every NBA game from 2009-2019. After processing the data, the average attendance and total number of home wins were calculated for each team and grouped by season. The 2012 season was dropped from the dataset because there was a reduced number of games due to the NBA lockout. It is also important to note that the dataset only includes regular season games.



| | Date | Start (ET) | Visitor/Neutral | PTS | Home/Neutral | PTS.1 | Unnamed: 6 | Unnamed: 7 | Attend. | Notes |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2009-10-27 | 7:30p | Boston Celtics | 95 | Cleveland Cavaliers | 89 | Box Score | NaN | 20562 | NaN |
| 1 | 2009-10-27 | 8:30p | Washington Wizards | 102 | Dallas Mavericks | 91 | Box Score | NaN | 19871 | NaN |
| 2 | 2009-10-27 | 10:00p | Houston Rockets | 87 | Portland Trail Blazers | 96 | Box Score | NaN | 20403 | NaN |
| 3 | 2009-10-27 | 10:30p | Los Angeles Clippers | 92 | Los Angeles Lakers | 99 | Box Score | NaN | 18997 | NaN |
| 4 | 2009-10-28 | 7:00p | Indiana Pacers | 109 | Atlanta Hawks | 120 | Box Score | NaN | 17998 | NaN |

*Table 3: Raw attendance data*



|  | Season | Team | Attend. | Home_Win |
|---|---|---|---|---|
| 1 | 2010 | Atlanta Hawks | 16592.72 | 34 |
| 2 | 2010 | Boston Celtics | 18624.00 | 24 |
| 3 | 2010 | Charlotte Bobcats | 15877.69 | 27 |
| 4 | 2010 | Chicago Bulls | 20719.28 | 23 |
| 5 | 2010 | Cleveland Cavaliers | 20562.00 | 34 |

*Table 4: Cleaned attendance data*

## *Data Analysis*

The two variables subject to analysis in this dataset are Attend. and Home_Win. Before seeing if the two variables have a relationship it is necessary to see how each is distributed, by plotting a histogram for each. Attendance does not follow a normal distribution and has a negative skew. The red line in the figure below is the mean attendance across 10 seasons.

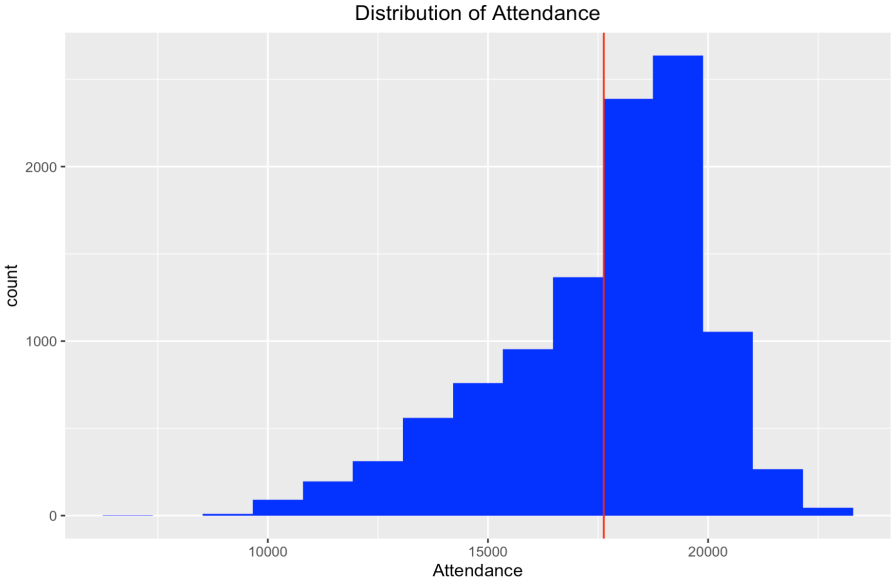


*Figure 2: Distribution of Attendance*

Upon first inspection Home_Win appears to be normal; however, according to the Shapiro-Wilks test it does not follow a normal distribution. The red line in the figure below is the mean home wins across 10 seasons.

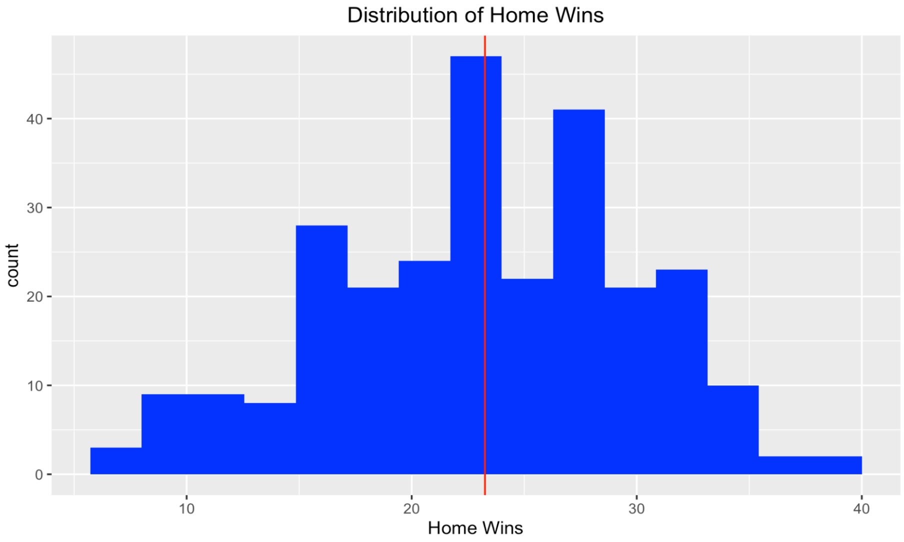


*Figure 3: Distribution of Home Wins*

The trends of each variable was also considered before testing correlation. From 2009-2019 fan attendance has increased by around 750 fans on average. The figure below shows a sharp rise in attendance but it is important to note that the x-axis only ranges from 17,000 to 18,000. 750 fans over the course of a decade is growth but not rapid growth. Fan attendance peaked in 2018 at just under 18,000 fans per game.

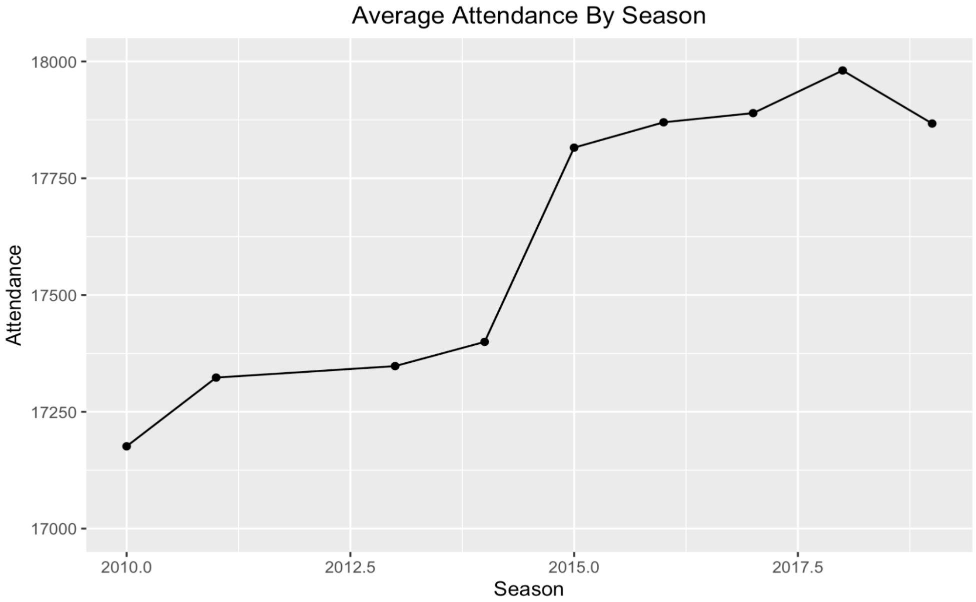


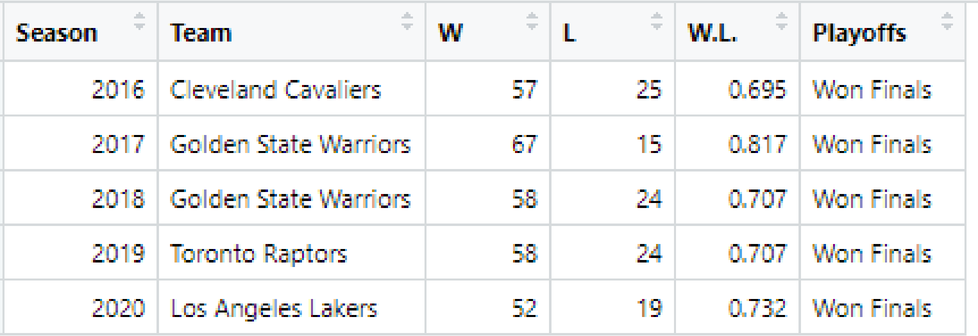
*Figure 4: Average Attendance By Season*

Unlike attendance, home wins, over the course of a decade, remained relatively stagnant. This is somewhat predictable because there is no reason to see a positive or negative trend in home wins. The amount of games has remained stable during the decade. Although, given that attendance has increased, there would be an increase in home wins if there was a positive correlation between the two variables.  Home wins peaked during the 2013 season, at 24 wins, and reached a decade low during the 2015 season, with an average of  22.5 home wins per team.

## Outcome of Previous Season

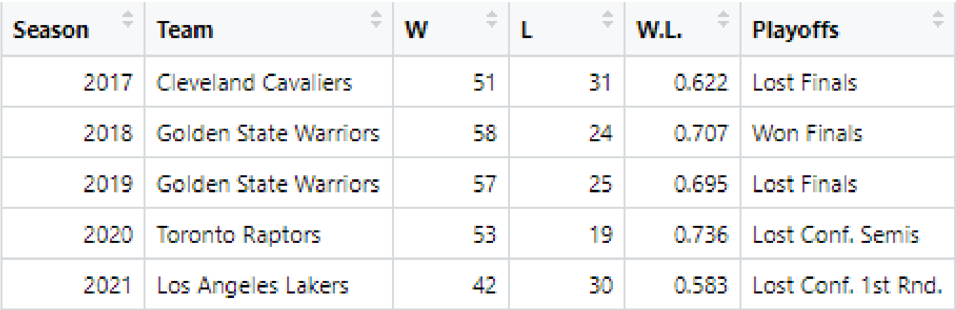
### *Data Gathering and Cleaning*

Similar to the attendance factor data, the data utilized to address the outcome of previous season factor was data accessed and subsequently web scraped using python from basketball-reference.com. This data spanned from the first NBA championship in 1951 to the most recently concluded NBA season in summer of 2021. Firstly, the team that won the NBA championship title each year was identified and saved to a dataframe. Afterwards, now knowing which team won the championship each season, the following season data for each team was identified and saved to its own dataframe. Each data frame included various variables relating to the overall season stats for each team observation, such as total number of wins and losses, win percentage, and when their season ended (for the year after dataframe). Other variables present on basketball-reference.com, such as each teams' Coach, each teams' conference finish, and other unnecessary features were removed from the dataframe. Additional data cleaning removed a * marker off of each of the team names (this was utilized on basketball-reference.com to denote if the team made the playoffs that year). Moreover, the East and West denotation for conference playoff losses were removed and the loss was only marked as a conference loss; with no note of the division the team played in. *Table 5* offers a display of the season statistics for the past five years NBA champions. While *Table 6* displays the subsequent season statistics for the season that came after each team was crowned champion.



| Season | Team | W | L | W.L. | Playoffs |
|---|---|---|---|---|---|
| 2016 | Cleveland Cavaliers | 57 | 25 | 0.695 | Won Finals |
| 2017 | Golden State Warriors | 67 | 15 | 0.817 | Won Finals |
| 2018 | Golden State Warriors | 58 | 24 | 0.707 | Won Finals |
| 2019 | Toronto Raptors | 58 | 24 | 0.707 | Won Finals |
| 2020 | Los Angeles Lakers | 52 | 19 | 0.732 | Won Finals |

*Table 5: NBA Champion Team Season Statistics*



| Season | Team | W | L | W.L. | Playoffs |
|---|---|---|---|---|---|
| 2017 | Cleveland Cavaliers | 51 | 31 | 0.622 | Lost Finals |
| 2018 | Golden State Warriors | 58 | 24 | 0.707 | Won Finals |
| 2019 | Golden State Warriors | 57 | 25 | 0.695 | Lost Finals |
| 2020 | Toronto Raptors | 53 | 19 | 0.736 | Lost Conf. Semis |
| 2021 | Los Angeles Lakers | 42 | 30 | 0.583 | Lost Conf. 1st Rnd. |

*Table 6: NBA Champion Team Following Season Statistics*

### *Data Analysis*

After successfully gathering and processing the data, statistical analysis was performed to analyze the effect that winning an NBA championship had on a team the following season. Specifically, the analysis conducted was a hypothesis test and subsequent two-tailed t-test. In this hypothesis test, the null hypothesis was defined as follows: For teams that won an NBA championship, the mean winning percentage is equal between the season that the team won the championship and the following season. While the alternative hypothesis was defined as follows: For teams that won an NBA  championship, the mean winning percentage is not equal between the season that the team won the championship and the following season. *Figure 5* displays these null and alternative hypotheses in mathematical notation. Subsequently, a two-tailed t-test was conducted utilizing these two hypotheses to determine the relationship between the mean winning percentages between the two seasons. The results of this test are further discussed in the Results section that will follow.

$$H_0\text{: } \mu_{\textit{Championship Season}} = \mu_{\textit{Following Season}}$$

$$H_A\text{: } \mu_{\textit{Championship Season}} \neq \mu_{\textit{Following Season}}$$

(Where $\mu$ equals the mean winning percentage for teams that won an NBA championship)

*Figure 5: Outcome of Previous Season Factor Hypothesis Test*

# Results

## In-Game Actions

In order to see whether winning teams score a higher three-point percentage than losing teams, a variety of statistical methods were used. These tests are important because there are a multitude of factors that go into winning games. To first understand the accuracy of three-point shots, *Figure 6* shows the difference between three-point shots attempted compared to three-point shots made successfully. The distribution shifts to the left and approaches zero as it shifts from points attempted to points made. The gap in these distributions is known as the three-point percentage, where further tests and analysis will be conducted below.

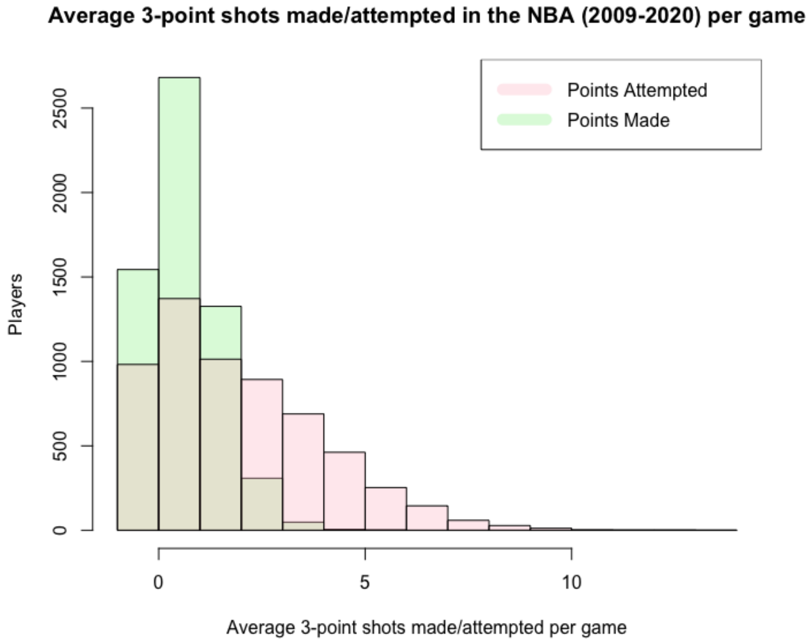


*Figure 6: Distribution of 3PT shots attempted and made.*

Next, to see how winning and losing teams compare to each other in recent history, the data was subset into losing teams and winning teams and split at where teams with a record higher than 50% wins out of total games are considered winning teams. Conversely, teams with a record less than or equal to 50% are considered losing teams. From there the three-point percentages of players were averaged by team. A potential drawback to this method could include certain players who play infrequently. Additionally, teams with smaller rosters may experience a larger standard deviation of three-point percentage differences among players. Nevertheless, looking at *Figure 7,* the New York Knicks (NYK) have the lowest average three-point percentage. This could be due to the high number of bench players who missed all of the three-point shots that they attempted. Therefore, looking at additional years will help provide a more accurate representation of the distribution of three-point percentages by team.

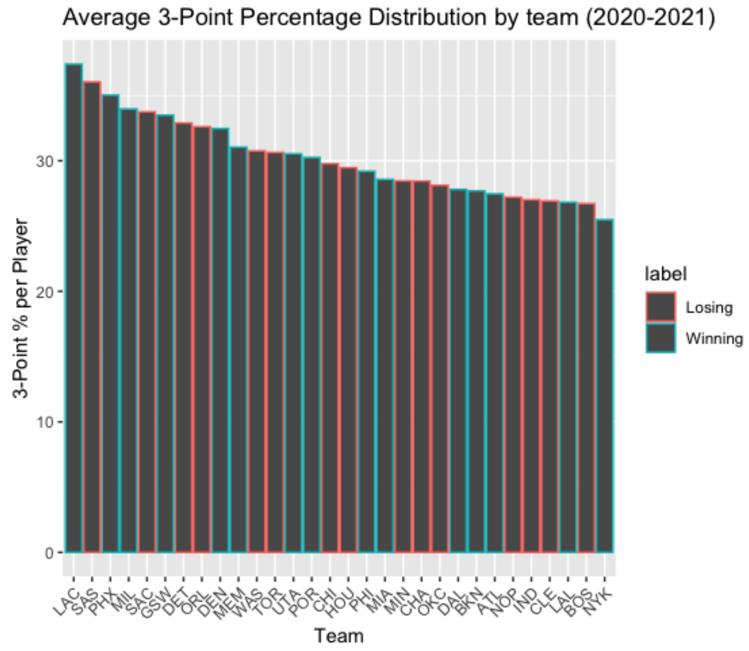


*Figure 7: Distribution of 3PT% by team in a single season (2020-2021).*

After expanding the scope of play to span from the 2009-2010 season to 2019-2020 season, the data was subset into winning and losing teams using the same criteria as mentioned in the previous paragraph. Looking at *Figure 8*, although there appears to be only a slight difference in winning and losing teams, the winning team distributions are predominately larger on the right side of the chart, and the losing team distributions are predominately larger in the left side of the chart. Similarly, the winning mean is higher than the losing mean. However, this is not enough evidence to say that winning teams have higher three-point percentages than losing teams.

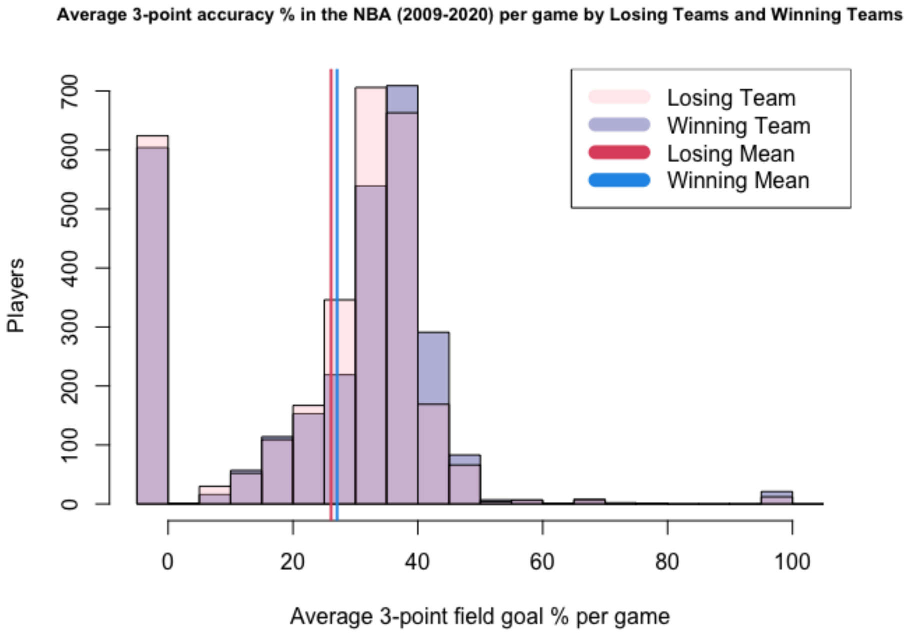


*Figure 8: Distribution of 3PT% by winning and losing teams from 2009-2020.*

Due to the close proximity of winning and losing means in both *Figure 8* above and *Figure 9* below, additional testing is necessary to determine whether winning teams have a higher three-point percentage than losing teams.To get a closer look, hypothesis tests were conducted. The null hypothesis states that for teams that had a winning record, the mean three-pointer percentage is less than or equal to the teams that had a losing record. Inversely, the alternative hypothesis states that for teams that had a winning record, the mean three-pointer percentage is greater than the teams that had a losing record.

In *Table 7*, the p-value was .014, meaning that the null hypothesis is rejected. With 95% confidence it can be concluded that the true mean three-point percentage of winning teams is at least .243% higher than losing teams. Additionally the mean three-point percentage of winning teams is 27.06%, and the mean three-point percentage of losing teams is 26.11%. In *Table 8*, the test is insignificant at a p-value of .312, meaning that we fail to reject the null hypothesis. However, the mean of winning teams is 30.48% and the mean of losing teams is 29.9%. Therefore, the three-point percentage is higher in winning teams for the 2020-21 season, but it is not statistically significant.

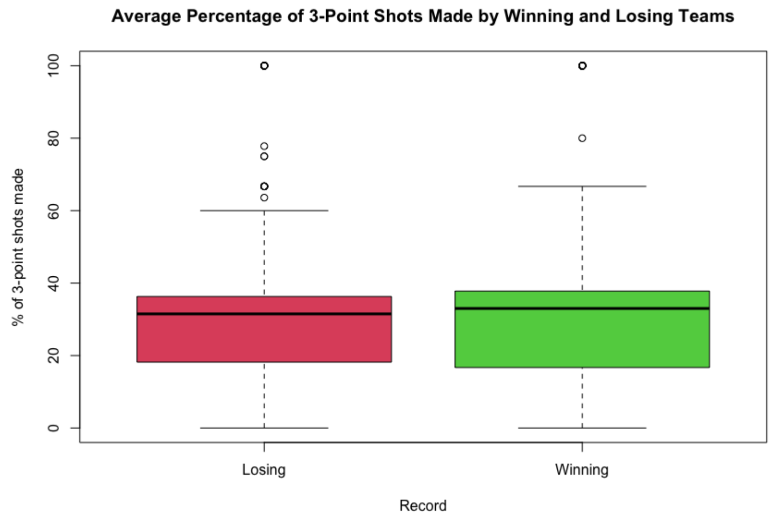


*Figure 9: Box Plots of 3PT % shots by winning and losing teams from 2009-2020.*

| P-Value | 95% Confidence Interval: Upper Bound | Mean 3PT % for Winning Teams | Mean 3PT % for Losing Teams |
|---|---|---|---|
| 0.014 | 0.243% | 27.06% | 26.11% |

*Table 7: One-sided T-Test for the difference in the distribution of winning and losing teams based on 3PT% in 2009-2020.*

| P-Value | 95% Confidence Interval: Upper Bound | Mean 3PT % for Winning Teams | Mean 3PT % for Losing Teams |
|---|---|---|---|
| 0.312 | -1.359% | 30.48% | 29.90% |

*Table 8: One-sided T-Test for the difference in the distribution of winning and losing teams based on 3PT% in 2020-21.*

## Salaries

The purpose of studying the way teams spend their money is to understand whether or not the successful teams spend their money differently from unsuccessful teams. In order to do this the data was first labeled by calling teams with a winning percentage greater than 50% a winning team and those with a winning percentage of less than 50% a losing team. Then to understand the distributions of the data better, new columns in the data were created to specify the mean, median, standard deviation, and total salary for each team respectively. Before any analysis was done boxplots for each of the mentioned statistics were created to inspect for significant differences between winning and losing teams. *Figure 10* shows each of these boxplots. From these plots, it appears that the winning team tends to be larger in each distribution statistic. In an attempt to confirm this, a 95% confidence interval was created for the difference between winning and losing teams for each distribution statistic using bootstrapping. Bootstrapping was utilized because of the relatively small sample size for each of the winning and losing teams.

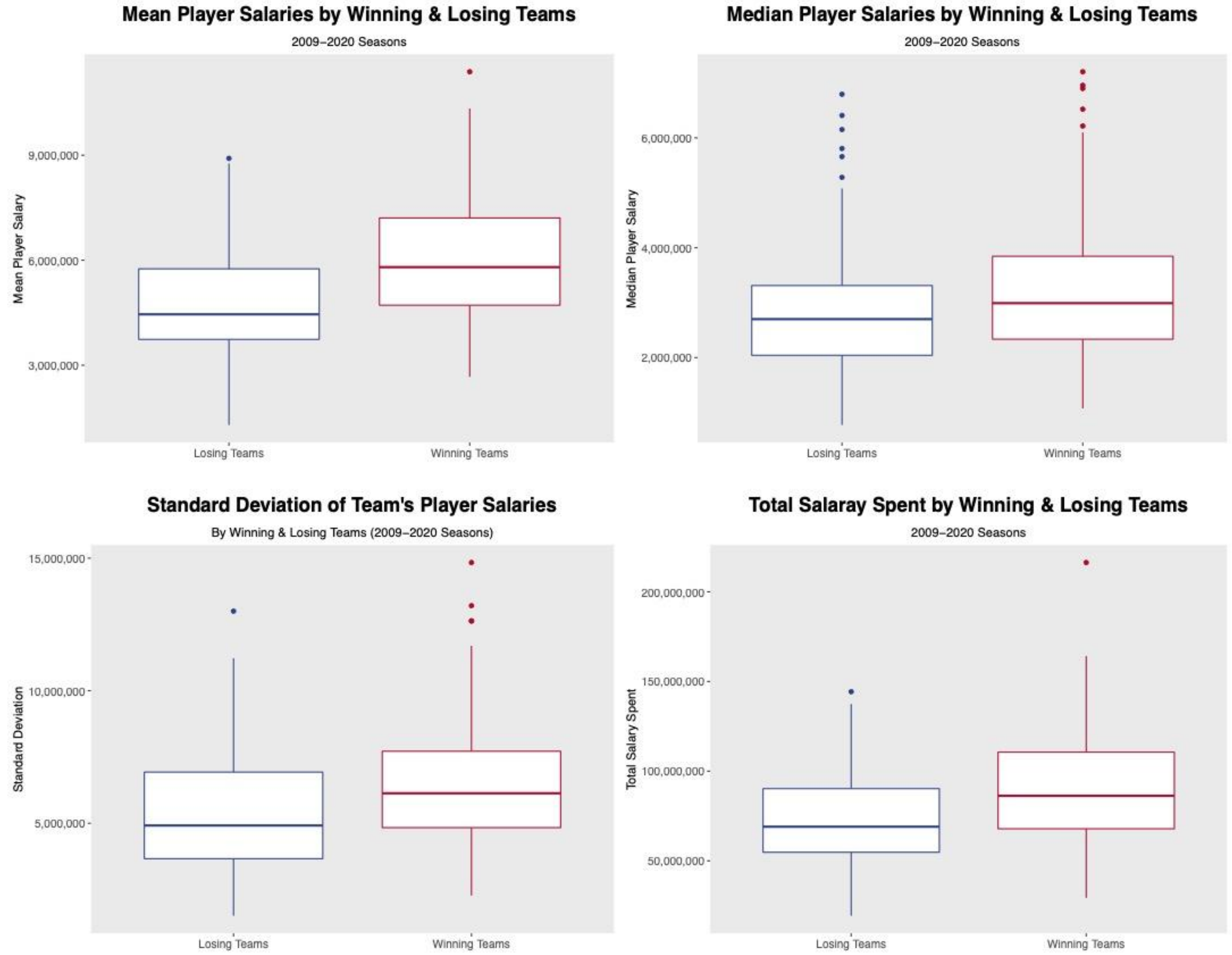
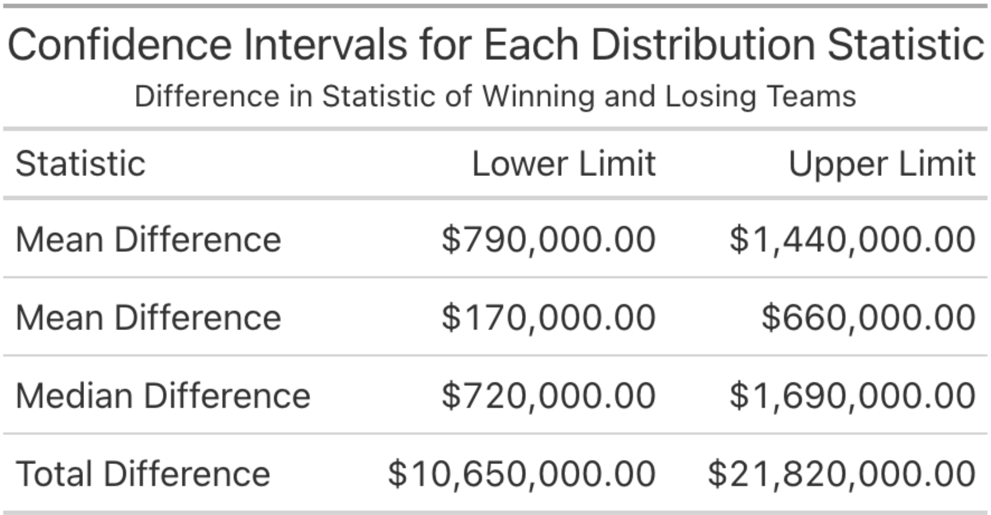


*Figure 10: Differences in salary statistics between winning & losing teams*

For each distribution statistic, the difference in the mean of the statistic for winning and losing teams of each of 100,000 bootstrapping samples was compiled. 95% confidence intervals for the difference in winning and losing teams were then calculated yielding the results shown in *Table 9*. From the resulting table, it can be seen that each confidence interval is strictly positive, meaning that for every distribution statistic, the winning teams have larger values than the losing teams.



Confidence Intervals for Each Distribution Statistic

Difference in Statistic of Winning and Losing Teams

| Statistic | Lower Limit | Upper Limit |
|---|---:|---:|
| Mean Difference | $790,000.00 | $1,440,000.00 |
| Mean Difference | $170,000.00 | $660,000.00 |
| Median Difference | $720,000.00 | $1,690,000.00 |
| Total Difference | $10,650,000.00 | $21,820,000.00 |

*Table 9: Confidence intervals for difference in distribution statistics of winning and losing teams (Rounded to nearest $10,000)*

## Fan Attendance

A scatter plot with a regression line was created to see if there is a relationship between attendance and home wins. There is a weak positive correlation between attendance and home wins because the correlation coefficient is 0.4. The grey area surrounding the regression line represents the 95% confidence interval.

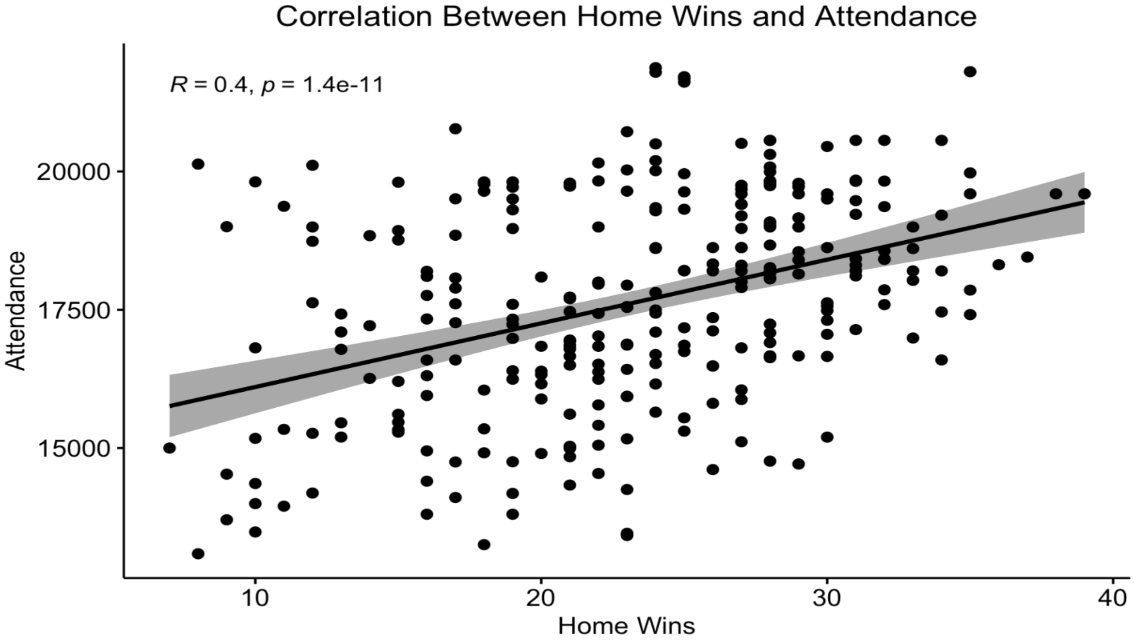


*Figure 11: Correlation Between Home Wins and Attendance*

## Outcome of Previous Season

As previously described, to analyze the effect that winning an NBA championship had on a team the following season a hypothesis test, and subsequent two-tailed t-test were employed (see *Figure 5* in Methodology section for Null and Alternative hypothesis). The conducted two-tailed t-test resulted in a p-value equal to 0.00959. Thus, utilizing the widely accepted significance level of 0.05, because this p-value was less than the significance level of 0.05 the null hypothesis was rejected. Moreover, we then adopted the alternative hypothesis that, for teams that won an NBA championship, the mean winning percentage is not equal between the season that the team won and the following season. Furthermore, the t-test provided a 95% confidence interval of (0.00946, 0.0671). Thus also supporting the conclusion of rejecting the null hypothesis because this confidence interval did not include the value of 0 in its span. Additionally, the mean winning percentage for teams that won the NBA championship was calculated to be 71.17%. While the mean winning percentage for teams season after winning the NBA championship was less, 67.34% to be exact. All these results can be easily observed in *Table 10* below.

| P-Value | 95% Confidence Interval: Lower Bound | 95% Confidence Interval: Upper Bound | Mean winning percentage for teams that won the NBA championship | Mean winning percentage for teams season after winning the NBA championship |
|---|---|---|---|---|
| 0.00959 | 0.00946 | 0.0671 | 71.17% | 67.34% |

*Table 10: Outcome of Previous Season T-Test Results*

To provide additional insight into the relationship between the winning percentage of NBA title seasons and those following, the overlapping histograms in *Figure 12* was created. In the histogram figure the distribution of winning percentage for championship title teams is displayed in light red, while the distribution of winning percentage for the season following a championship title is displayed in light blue. Furthermore, the solid red line marks the mean winning percentage of teams championship seasons, and the solid blue line marks the mean winning percentage for teams the year after winning. As can be observed, there is a large amount of overlap between the two histogram distributions; however, there is a difference in the actual means for each.

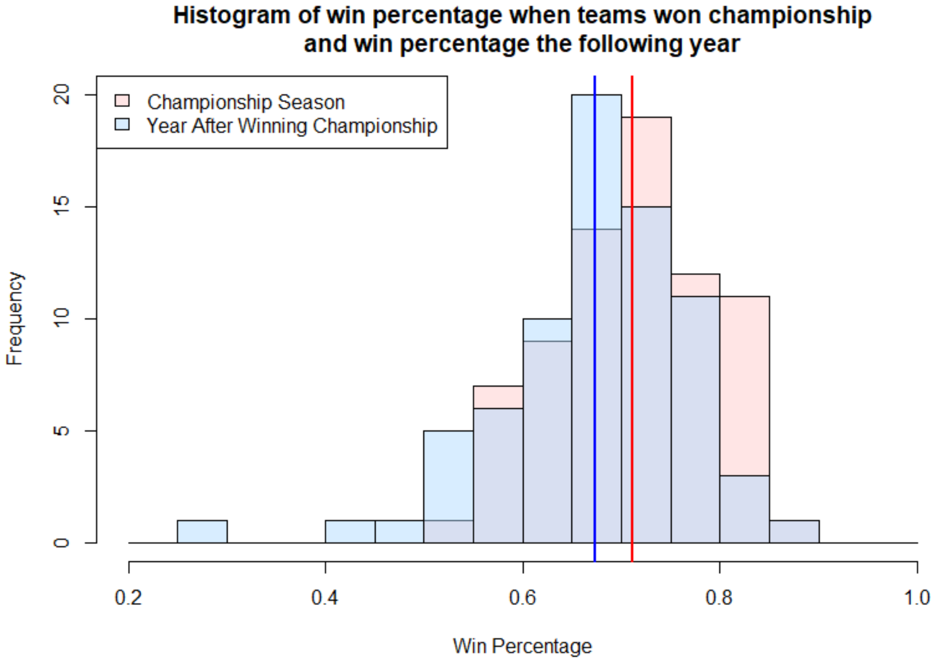


*Figure 12: Histogram of Win Percentage when Teams Won Championship and Win Percentage the Following Year*

Although there is not only an observable difference between the mean winning percentage of a championship season and the following season (as seen in the figure above), as well as a proven difference (through the earlier discussed hypothesis test), it should be noted that oftentimes the season following a championship are still relatively successful. This is best observed in *Figure 13*. This figure is a Pie Chart that displays the proportion of season outcomes the year following a NBA championship. As can be seen, the smallest percentage is the proportion of teams that miss the playoffs the year following an NBA championship title (2.703%). This means that 97% of teams that win an NBA championship at least make it to the playoffs the following season. Following that, a total of 50% of teams then lose at some point in the Conference bracket, with 9.459% of champion teams losing in the first round of Conference play, 14.865% of champion teams losing in the Conference Semifinals, and 25.676% losing in the Conference Finals. Furthermore, that leaves a whopping 47.297% of champion teams making the NBA championship Final game the following year, with 31.081% of champion teams actually winning the NBA championship finals again. Thus, although there does seem to be a difference between the average winning percentage of a season where a team wins an NBA championship and the season following, the majority of teams don’t merely make the playoffs but instead continue to compete to a high level throughout the postseason.

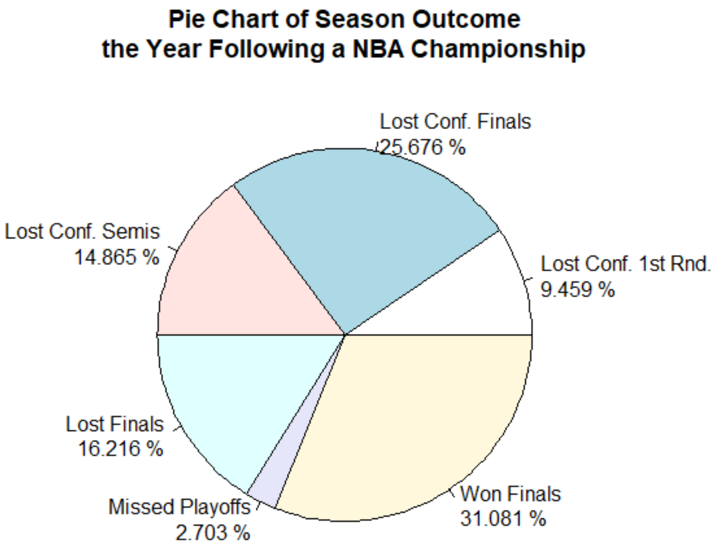


*Figure 13: Pie Chart of Season Outcome the Year Following a NBA Championship*

# Conclusions and Discussion

## In-game Actions

Looking collectively at 2009-2021 seasons of play, winning teams have a greater three-point percentage than losing teams. However, this is not true for all seasons. Particularly in the 2020-21 season, we cannot conclude that the mean three-point percentage is higher for winning teams than losing teams. Even though the New York Knicks had a winning record in the 2020-21 season, they had the lowest team average three-point percentage.

These findings can help both players and coaches realize the overall importance of scoring three-point baskets. Even the players who shoot three-point shots less frequently will help their team win if they successfully score. However, recent seasons show that being successful in just three-point shooting will not help them win a basketball game. With the increased competitiveness of the NBA, the team likely has to have success in a multitude of areas in order to win.

In addition, there are likely other in-game metrics that could be more successful in determining whether a team would be a winning team or losing team. Some other in-game factors that might be more predictive of a winning team could include free-throw percentage, field goal percentage, or blocks. Additionally, looking at a combination of these factors together could create some insightful conclusions. Furthermore, some of the external factors mentioned below significantly impact whether a team will win or lose.

# Salary

The results show that, in general, winning teams tend to spend more money than losing teams. This is not a groundbreaking conclusion, however it does confirm the idea that spending more money does lead to winning more games. This information would also be useful to the teams who, as seen on the right side of *Figure 1* , spent well under the allotted salary cap. Regardless of their success that season, this information would imply that spending more money than they did would have given them a better chance to have more success than they did.

The more insightful conclusion from this data comes from the fact that the differences in player salary mean and team salary standard deviation are much larger than that of the median. This would suggest that winning teams are more likely to spend more money on their best players than losing teams. *Figures 14 & 15* illustrate this using the real data. *Figure 14* shows the salary distribution of each team in the dataset with the highest paid player being on the left and lowest paid player being on the right. It can be seen clearly on the left side of the graph that winning teams are spending more money than losing teams on these top players. *Figure 15* again proves this point in another way. It shows a histogram of salaries of players who made more than $20 million in a given season and is separated by winning and losing teams. Again, this plot is clearly dominated by winning teams. Given these results, an NBA team that wants to be more successful should consider signing a star player as the results back up that star players contribute to more wins and are worth spending the extra money when team success is the variable being solved for.

In reality however, some teams are just not willing to spend the large amounts of money that others are. In this case, this analysis would not be all that useful to them and they will likely not have as much success as other teams who are willing to spend the money. Though, a similar analysis could be done in order to be more useful to these more frugal teams. Instead of doing an analysis on the actual salary of each player, the data could be normalized in a way by using the percentage of total salary a team spent on each given player. This type of analysis could lead to some different results as it could be seen clearly in *Figure 1* that some teams spent significantly less money than other teams did.

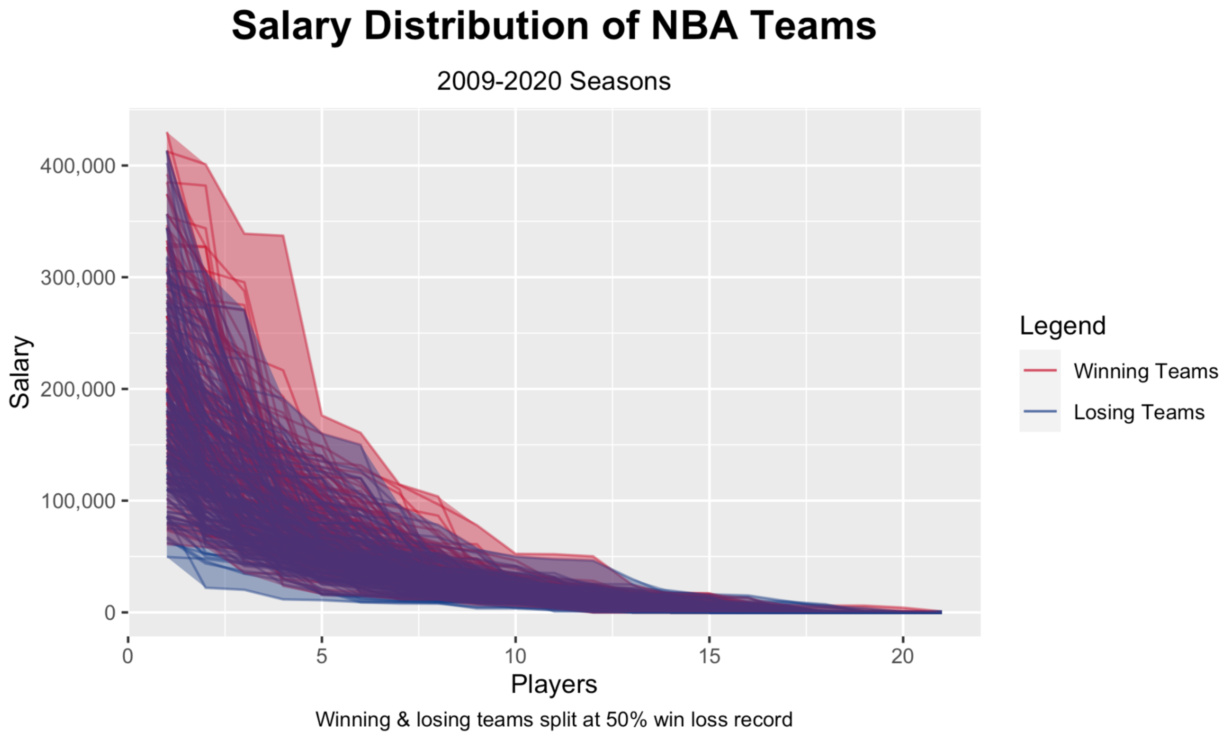


*Figure 14: Salary distribution of each team in the dataset from highest to lowest paid player*

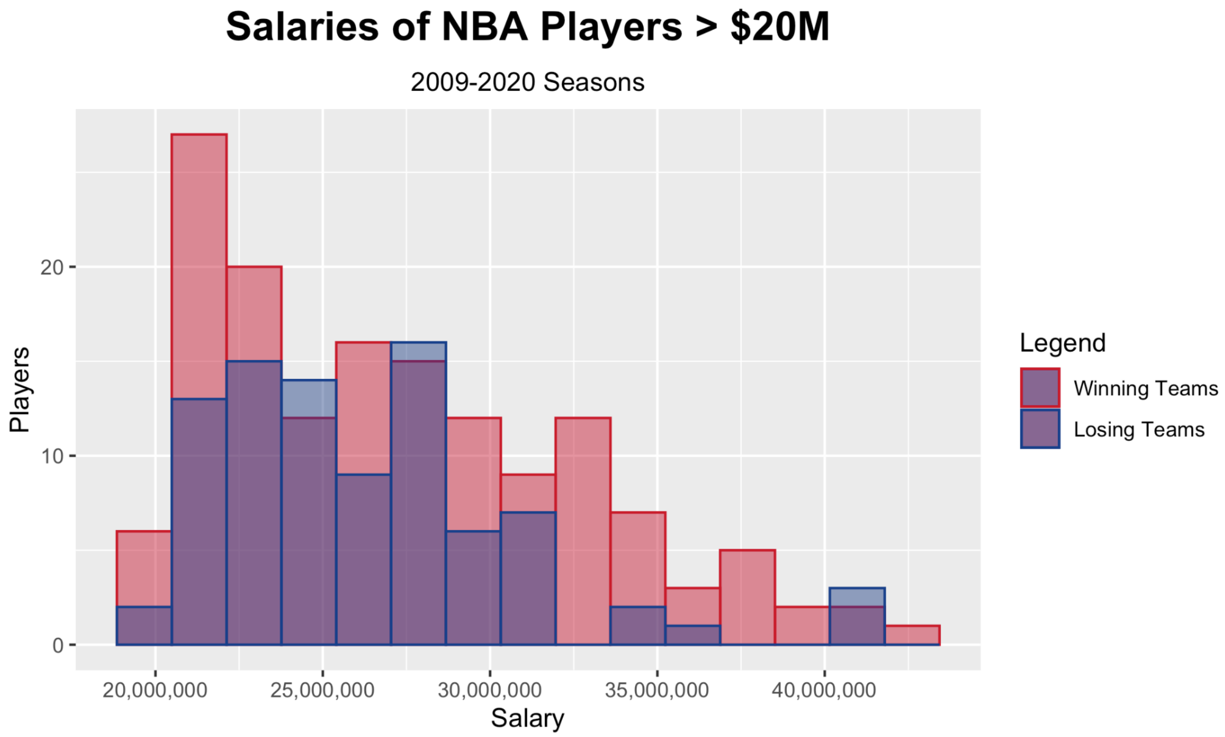


*Figure 15: Salaries of players who made more than $20 million in a given season*

## Fan Attendance

The number of fans at a game do not make a significant difference when it comes to the home team winning a game during the NBA regular season. The main reason fan attendance does not have a bigger impact on winning at home is because teams in larger markets (New York, Chicago, Los Angeles) still draw large crowds even when their teams are not good. Over the past decade, the teams in the aforementioned cities have not been good; therefore, they have had a low number of home wins. Teams in these cities are still able to draw big crowds, even when the teams are not good, due to tourism. Since good teams and bad teams both have high fan attendance, there is not much of a relationship between attendance and home wins. If the playoffs were also considered in the analysis, there may have been a stronger relationship between the variables since home court advantage is very important in the playoffs.

## Outcome of Previous Seasons

In conclusion, when investigating the factor of outcome of previous season for NBA teams, an observable difference between the overall season winning percentage of teams that win an NBA championship title and the overall winning percentage for the season following was identified. Specifically, with a team's winning percentage decreasing the season after winning a championship. Even so, it is important to note that although this difference may be present, the vast majority of NBA championship title teams still at least make the playoffs the following season. Moreover, a notable percentage of them are successful in the playoffs and at least make the NBA championship finals game.

This revelation provides insight into the initial concern mentioned in the introduction regarding the effects that the extended length of a season in the playoffs might have on a team, as they would have less time to recover and prepare for the upcoming season. Although there may be a decrease in the overall winning percentage, there is still a majority of teams that continue to compete at a high level in the regular season and after in the playoffs. Thus, players, coaches, and managers should not worry about an extended season in the playoffs, for fear of it affecting the following season.

Furthermore, knowing that the NBA and the teams within the association make a profit off of the various revenue streams that the sport brings and that winning an NBA championship title does not affect the following season, additional analysis would be beneficial in determining the actual dollar value increase in profit/revenue that teams within the NBA experience when making the playoffs, and especially when winning an NBA championship title. This investigation might provide insight into the relationship between the success of NBA teams and the financial impact that has for a team and the association as a whole.

# Appendix:

```
import urllib
from urllib.request import urlopen
import pandas as pd
import numpy as np


# In[8]:


def make_soup(url):
    req = urllib.request.Request(url,
    headers={'User-Agent': 'Mozilla/5.0'})
    webpage = urlopen(req)
    main_doc = webpage.read()
    return main_doc


# In[69]:


months=["october","november","december","january","february","march","april"]
df_list=[]
for year in range(2010,2020):
    for month in months:
        try:
            soup= make_soup("https://www.basketball-reference.com/leagues/NBA_"+str(year)+"_games-"+str(month)
+".html")
            df_list.append(pd.read_html(soup)[0])
        except Exception as e:
            pass


# In[70]:


#add to dataframe
attendanceDF = pd.concat(df_list)
attendanceDF.head()


# In[71]:


#drop row with playoffs
attendanceDF.drop(attendanceDF.loc[attendanceDF['Date']=="Playoffs"].index, inplace=True)


# In[72]:


#convert Date col to datetime type
from datetime import datetime
attendanceDF['Date']= pd.to_datetime(attendanceDF['Date'])
attendanceDF.head()
```

```
# In[73]:


#keep only regular season
start_date = "2009-10-27"
end_date = "2010-04-14"

after_start_date = attendanceDF["Date"] >= start_date
before_end_date = attendanceDF["Date"] <= end_date
between_two_dates = after_start_date & before_end_date
attendanceDF10 = attendanceDF.loc[between_two_dates]
attendanceDF10["Season"] = 2010


# In[74]:


#keep only regular season
start_date = "2010-10-26"
end_date = "2011-04-13"

after_start_date = attendanceDF["Date"] >= start_date
before_end_date = attendanceDF["Date"] <= end_date
between_two_dates = after_start_date & before_end_date
attendanceDF11 = attendanceDF.loc[between_two_dates]
attendanceDF11["Season"] = 2011


# In[75]:


#keep only regular season
start_date = "2011-12-25"
end_date = "2012-04-26"

after_start_date = attendanceDF["Date"] >= start_date
before_end_date = attendanceDF["Date"] <= end_date
between_two_dates = after_start_date & before_end_date
attendanceDF12 = attendanceDF.loc[between_two_dates]
attendanceDF12["Season"] = 2012


# In[76]:


#keep only regular season
start_date = "2012-10-30"
end_date = "2013-04-17"

after_start_date = attendanceDF["Date"] >= start_date
before_end_date = attendanceDF["Date"] <= end_date
between_two_dates = after_start_date & before_end_date
attendanceDF13 = attendanceDF.loc[between_two_dates]
attendanceDF13["Season"] = 2013
```

```
# In[77]:


#keep only regular season
start_date = "2013-10-29"
end_date = "2014-04-16"

after_start_date = attendanceDF["Date"] >= start_date
before_end_date = attendanceDF["Date"] <= end_date
between_two_dates = after_start_date & before_end_date
attendanceDF14 = attendanceDF.loc[between_two_dates]
attendanceDF14["Season"] = 2014


# In[78]:


#keep only regular season
start_date = "2014-10-28"
end_date = "2015-04-15"

after_start_date = attendanceDF["Date"] >= start_date
before_end_date = attendanceDF["Date"] <= end_date
between_two_dates = after_start_date & before_end_date
attendanceDF15 = attendanceDF.loc[between_two_dates]
attendanceDF15["Season"] = 2015


# In[79]:


#keep only regular season
start_date = "2015-10-27"
end_date = "2016-04-13"

after_start_date = attendanceDF["Date"] >= start_date
before_end_date = attendanceDF["Date"] <= end_date
between_two_dates = after_start_date & before_end_date
attendanceDF16 = attendanceDF.loc[between_two_dates]
attendanceDF16["Season"] = 2016


# In[80]:


#keep only regular season
start_date = "2016-10-25"
end_date = "2017-04-12"

after_start_date = attendanceDF["Date"] >= start_date
before_end_date = attendanceDF["Date"] <= end_date
between_two_dates = after_start_date & before_end_date
attendanceDF17 = attendanceDF.loc[between_two_dates]
```

```
attendanceDF17["Season"] = 2017


# In[81]:


#keep only regular season
start_date = "2017-10-17"
end_date = "2018-04-11"

after_start_date = attendanceDF["Date"] >= start_date
before_end_date = attendanceDF["Date"] <= end_date
between_two_dates = after_start_date & before_end_date
attendanceDF18 = attendanceDF.loc[between_two_dates]
attendanceDF18["Season"] = 2018


# In[82]:


#keep only regular season
start_date = "2018-10-16"
end_date = "2019-04-10"

after_start_date = attendanceDF["Date"] >= start_date
before_end_date = attendanceDF["Date"] <= end_date
between_two_dates = after_start_date & before_end_date
attendanceDF19 = attendanceDF.loc[between_two_dates]
attendanceDF19["Season"] = 2019


# In[91]:


frames = [attendanceDF10, attendanceDF11,attendanceDF12,attendanceDF13,attendanceDF14,
attendanceDF15,attendanceDF16,attendanceDF17,attendanceDF18,attendanceDF19]

attendanceDF = pd.concat(frames)


# In[92]:


attendanceDF.drop(columns=["Start (ET)","Unnamed: 6","Unnamed: 7","Notes"],inplace=True)
attendanceDF.head()


# In[41]:


celtics = attendanceDF.loc[attendanceDF['Home/Neutral']=="Boston Celtics"]


# In[42]:
```

```
celtics['Attend.']=celtics['Attend.'].astype(int)


# In[94]:


import matplotlib.pyplot as plt
attendanceDF['Attend.']=attendanceDF['Attend.'].astype(int)
plt.hist(attendanceDF["Attend."])
plt.show()


# In[95]:


attendanceDF['PTS']=attendanceDF['PTS'].astype(int)
attendanceDF['PTS.1']=attendanceDF['PTS.1'].astype(int)


# In[96]:


attendanceDF['Home_Win'] = np.where(attendanceDF['PTS.1']>attendanceDF['PTS'], True, False)
attendanceDF.head()


# In[102]:


attendanceDF.to_csv("attendanceData.csv",index=False)


# In[97]:


teams=attendanceDF["Home/Neutral"].unique()


# In[99]:


dfList=[]
dfWins=[]
dfTeams=[]
#calculate avg attendance for each team
for team in teams:
    dfTeams.append(team)
    means=attendanceDF.loc[attendanceDF['Home/Neutral']==team,"Attend."].mean()
    wins = sum(attendanceDF.loc[((attendanceDF['Home/Neutral'] ==team) & (attendanceDF['Home_Win']==True))]
.value_counts())
    dfList.append(means)
    dfWins.append(wins)
```

```
# In[100]:


attd = pd.DataFrame(
    {'Team': dfTeams,
     'avg_attendance': dfList,
     'home_wins': dfWins
    })


# In[101]:


attd.head()


# In[131]:


attd.describe()


# In[134]:


attd.to_csv("aggregateData.csv",index=False)


# In[150]:


attd.sort_values(by="avg_attendance")


# In[ ]:
```

```
import urllib
from urllib.request import urlopen
import pandas as pd
import numpy as np


# In[31]:


def make_soup(url):
    req = urllib.request.Request(url,
    headers={'User-Agent': 'Mozilla/5.0'})
    webpage = urlopen(req)
    main_doc = webpage.read()
    return main_doc


# In[32]:



teams = ["ATL", "BOS", "BRK", "CHI", "CHO",
 "CLE", "DAL", "DEN", "DET", "GSW" ,"HOU", "IND",
 "LAC", "LAL", "MEM", "MIA", "MIL", "MIN", "NOP", "NYK", "OKC", "ORL", "PHI", "PHO", "POR" ,"SAC",
"SAS", "TOR", "UTA", "WAS"]
df_list=[]

for team in teams:
    #print(team)
    try:
        soup= make_soup("https://www.basketball-reference.com/teams/"+str(team)+"/")
        df_list.append(pd.read_html(soup)[0])
    except Exception as e:
            pass



# In[67]:


df = pd.concat(df_list)


# In[68]:


df.head()


# In[69]:


df.drop([ 'Lg','Unnamed: 8',
       'Unnamed: 15','Coaches',
    'Top WS',
     "Finish", "SRS", "Pace", "Rel Pace", "ORtg", "Rel ORtg", "DRtg", "Rel DRtg"],axis=1,inplace=True)


# In[70]:


df.loc[df.Playoffs.str.startswith('Won')==True, 'Playoffs'] = "Won Finals"


# In[71]:


df.loc[(df.Playoffs.isna()==True),'Playoffs']='Missed Playoffs'
df.head(10)


# In[72]:


seasons = df['Season'].str.split('-', expand=True)
df.Season = seasons[0]


# In[73]:


df.reset_index(inplace=True)


# In[74]:


df.head(10)


# In[75]:


champions = df.loc[(df.Playoffs == "Won Finals")].index
yearAfter = df.loc[(df.Playoffs == "Won Finals")].index-1


# In[76]:


champsDF=df.iloc[champions]


# In[77]:


champsDF.head()
```

```
# In[78]:


yearAfterDF=df.iloc[yearAfter]
yearAfterDF.head()


# In[79]:


frames = [champsDF, yearAfterDF]
df = pd.concat(frames)


# In[80]:


df.head()


# In[81]:


df.drop_duplicates(inplace=True)


# In[82]:


df=df.sort_values(by=['Season'])
champsDF = champsDF.sort_values(by = ['Season'])
yearAfterDF = yearAfterDF.sort_values(by = ['Season'])


# In[83]:


df.drop(df[df.Season == "2021"].index, inplace=True)
yearAfterDF.drop(yearAfterDF[yearAfterDF.Season == "2021"].index, inplace=True)


# In[84]:


df.tail()


# In[85]:


df.Season=df.Season.astype(int)
champsDF.Season=champsDF.Season.astype(int)
yearAfterDF.Season=yearAfterDF.Season.astype(int)
```

```
# In[86]:


df.Season = df.Season+1
champsDF.Season = champsDF.Season+1
yearAfterDF.Season = yearAfterDF.Season+1


# In[87]:


df.tail()


# In[88]:


df.drop("index",inplace=True,axis=1)
champsDF.drop("index",inplace=True,axis=1)
yearAfterDF.drop("index",inplace=True,axis=1)


# In[89]:


df.dropna(inplace=True)
champsDF.dropna(inplace=True)
yearAfterDF.dropna(inplace=True)


# In[92]:


df.tail(10)


# In[93]:


df.to_csv("TitleEffect.csv",index=False)
champsDF.to_csv("Champs.csv", index = False)
yearAfterDF.to_csv("yearAfter.csv", index = False)


# In[ ]:
```

# R Notebook

Notes about output:

Upon output - there will be a single data set for each season spanning from the 09-10 season to the 20-21 season. Each dataset contains the player, his team, his salary, and the number of games he played that season. My thoughts for using this for analysis would be to label teams based on win percentage as a good or bad team and then we could do some hypothesis testing on whether the good teams have a higher or lower median salary, top salary, Q3 salary etc.

### Data URLs

https://hoopshype.com/salaries/players/2020-2021/ https://www.nba.com/stats/players/traditional/?sort=PTS&dir=-1&Season=2020-21&SeasonType=Regular%20Season

## Libraries

```
library(ggplot2)
library(XLConnect)
```

```
## XLConnect 1.0.5 by Mirai Solutions GmbH [aut],
##   Martin Studer [cre],
##   The Apache Software Foundation [ctb, cph] (Apache POI),
##   Graph Builder [ctb, cph] (Curvesapi Java library),
##   Brett Woolridge [ctb, cph] (SparseBitSet Java library)
```

```
## https://mirai-solutions.ch
## https://github.com/miraisolutions/xlconnect
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.1.0     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggimage)
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'

## The following object is masked from 'package:purrr':
##
##     flatten
```

```
library(httr)
library(rvest)
```

```
##
## Attaching package: 'rvest'

## The following object is masked from 'package:readr':
##
##     guess_encoding
```

```
library(gghighlight)
`%!in%` <- Negate(`%in%`)
options(java.parameters = "- Xmx1024m")
```

## Importing Datasets

```
l_sals = c()
years = c('09_10',"10_11","11_12","12_13","13_14","14_15","15_16","16_17","17_18","18_19","19_20","20_2

l_stats = c()
for(i in 1:12){
  stats = readWorksheetFromFile("raw_player+team_data.xlsx",sheet = i)
  d_var_2 = paste0('stats_',years[i])
  stats = stats[,c(2:3,5,13:15)]
  assign(d_var_2, stats)
  xlcFreeMemory()
}
```

## Join Datasets

```
threes = list(stats_09_10,stats_10_11,stats_11_12,stats_12_13,stats_13_14,
            stats_14_15,stats_15_16,stats_16_17,stats_17_18,stats_18_19,
            stats_19_20,stats_20_21)

stat_group = function(stat){
  stat = stat %>%
    drop_na() %>%
```

```r
    arrange(TEAM)
  return(stat)
}

df = data.frame(player=character(),
                TEAM=character(),
                GP=numeric(),
                X3PM = numeric(),
                X3PA = numeric(),
                X3p. = numeric(),
                year=character(),
                stringsAsFactors=FALSE)

for(i in 1:12){
  x = threes[[i]]
  x = stat_group(x)
  fn = paste0('threes_',years[i],'.csv')
  write_csv(x,fn)

  x$year = years[i]
  df = rbind(df,x)

}

write_csv(df,'all_seasons_threes.csv')
```

# R Notebook

Notes about output:

Upon output - there will be a single data set for each season spanning from the 09-10 season to the 20-21 season. Each dataset contains the player, his team, his salary, and the number of games he played that season. My thoughts for using this for analysis would be to label teams based on win percentage as a good or bad team and then we could do some hypothesis testing on whether the good teams have a higher or lower median salary, top salary, Q3 salary etc.

### Data URLs

https://hoopshype.com/salaries/players/2020-2021/ https://www.nba.com/stats/players/traditional/?sort=PTS&dir=-1&Season=2020-21&SeasonType=Regular%20Season

## Libraries

```
library(ggplot2)
library(XLConnect)
```

```
## XLConnect 1.0.5 by Mirai Solutions GmbH [aut],
##   Martin Studer [cre],
##   The Apache Software Foundation [ctb, cph] (Apache POI),
##   Graph Builder [ctb, cph] (Curvesapi Java library),
##   Brett Woolridge [ctb, cph] (SparseBitSet Java library)
```

```
## https://mirai-solutions.ch
## https://github.com/miraisolutions/xlconnect
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.1.0     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(ggimage)
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'

## The following object is masked from 'package:purrr':
##
##     flatten
```

```r
library(httr)
library(rvest)
```

```
##
## Attaching package: 'rvest'

## The following object is masked from 'package:readr':
##
##     guess_encoding
```

```r
library(gghighlight)
`%!in%` <- Negate(`%in%`)
options(java.parameters = "- Xmx1024m")
```

## Importing Datasets

```r
l_sals = c()
years = c('09_10',"10_11","11_12","12_13","13_14","14_15","15_16","16_17","17_18","18_19","19_20","20_2
for(i in 1:12){
  sals = readWorksheetFromFile("raw_salary_data.xlsx",sheet = i,header = FALSE)
  d_var_1 = paste0('sals_',years[i])
  sals = sals[,2:3]
  colnames(sals) = c('player','salary')
  assign(d_var_1, sals)
  xlcFreeMemory()


}
l_stats = c()
for(i in 1:12){
  stats = readWorksheetFromFile("raw_player+team_data.xlsx",sheet = i)
  d_var_2 = paste0('stats_',years[i])
  stats = stats[,c(2:3,5)]
  assign(d_var_2, stats)
  xlcFreeMemory()
}
```

## Join Datasets

```r
sal_join = function(sal,stat){
  sal = sal %>%
    left_join(stat, by = c('player' = 'PLAYER')) %>%
    drop_na() %>%
    arrange(TEAM)
  return(sal)
}
sals_09_10 = sal_join(sals_09_10,stats_09_10)
sals_10_11 = sal_join(sals_10_11,stats_10_11)
sals_11_12 = sal_join(sals_11_12,stats_11_12)
sals_12_13 = sal_join(sals_12_13,stats_12_13)
sals_13_14 = sal_join(sals_13_14,stats_13_14)
sals_14_15 = sal_join(sals_14_15,stats_14_15)
sals_15_16 = sal_join(sals_15_16,stats_15_16)
sals_16_17 = sal_join(sals_16_17,stats_16_17)
sals_17_18 = sal_join(sals_17_18,stats_17_18)
sals_18_19 = sal_join(sals_18_19,stats_18_19)
sals_19_20 = sal_join(sals_19_20,stats_19_20)
sals_20_21 = sal_join(sals_20_21,stats_20_21)
sals = list(sals_09_10,sals_10_11,sals_11_12,sals_12_13,sals_13_14,
            sals_14_15,sals_15_16,sals_16_17,sals_17_18,sals_18_19,
            sals_19_20,sals_20_21)
```

### Write to csvs

```r
for(i in 1:12){
  fn = paste0('salaries_',years[i],'.csv')
  write_csv(sals[[i]],fn)
}
```

```r
df = data.frame(player=character(),
                salary=numeric(),
                TEAM=character(),
                GP=numeric(),
                year=character(),
                stringsAsFactors=FALSE)

for(i in 1:12){
  x = sals[[i]]
  x$year = years[i]

  df = rbind(df,x)

}
#write_csv(df,'all_seasons_salaries.csv')
```

# Final Project R Markdown

Matt Young

11/28/2021

Read in data

```
df<-read.csv("all_seasons_threes.csv", stringsAsFactors = TRUE)
head(df)
```

```
##            PLAYER TEAM GP X3PM X3PA  X3P.  year
## 1     Joe Johnson  ATL 76  1.7  4.6  36.9 09_10
## 2  Jamal Crawford  ATL 79  2.1  5.4  38.2 09_10
## 3      Josh Smith  ATL 81  0.0  0.1   0.0 09_10
## 4      Al Horford  ATL 81  0.0  0.0 100.0 09_10
## 5 Marvin Williams  ATL 81  0.5  1.6  30.3 09_10
## 6      Mike Bibby  ATL 80  1.6  4.1  38.9 09_10
```

```
str(df)
```

```
## ’data.frame’:    5916 obs. of  7 variables:
##  $ PLAYER: Factor w/ 1446 levels "Aaron Brooks",..: 684 578 755 18 956 992 969 1439 650 685 ...
##  $ TEAM  : Factor w/ 32 levels "ATL","BKN","BOS",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ GP    : int  76 79 81 81 81 80 79 78 71 64 ...
##  $ X3PM  : num  1.7 2.1 0 0 0.5 1.6 0.7 0 0.1 0 ...
##  $ X3PA  : num  4.6 5.4 0.1 0 1.6 4.1 2.1 0.1 0.5 0.1 ...
##  $ X3P.  : num  36.9 38.2 0 100 30.3 38.9 33.7 0 21.9 14.3 ...
##  $ year  : Factor w/ 12 levels "09_10","10_11",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Sort by 3 point percentage in descending order

```
newdata<-df[order(-df$X3P.),]
```

Require at least 1 3-point attempt

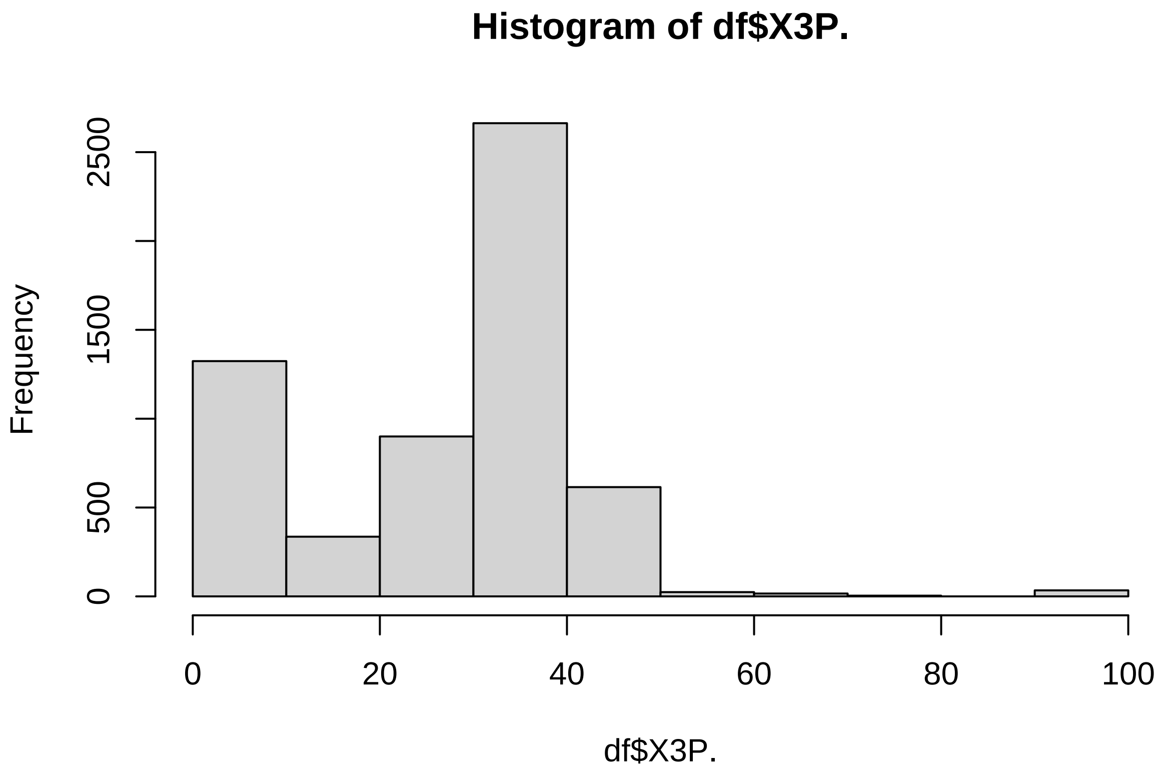
```
newdata<-newdata[newdata$X3PA > 0.0,]
```

Require at least 25% of games played

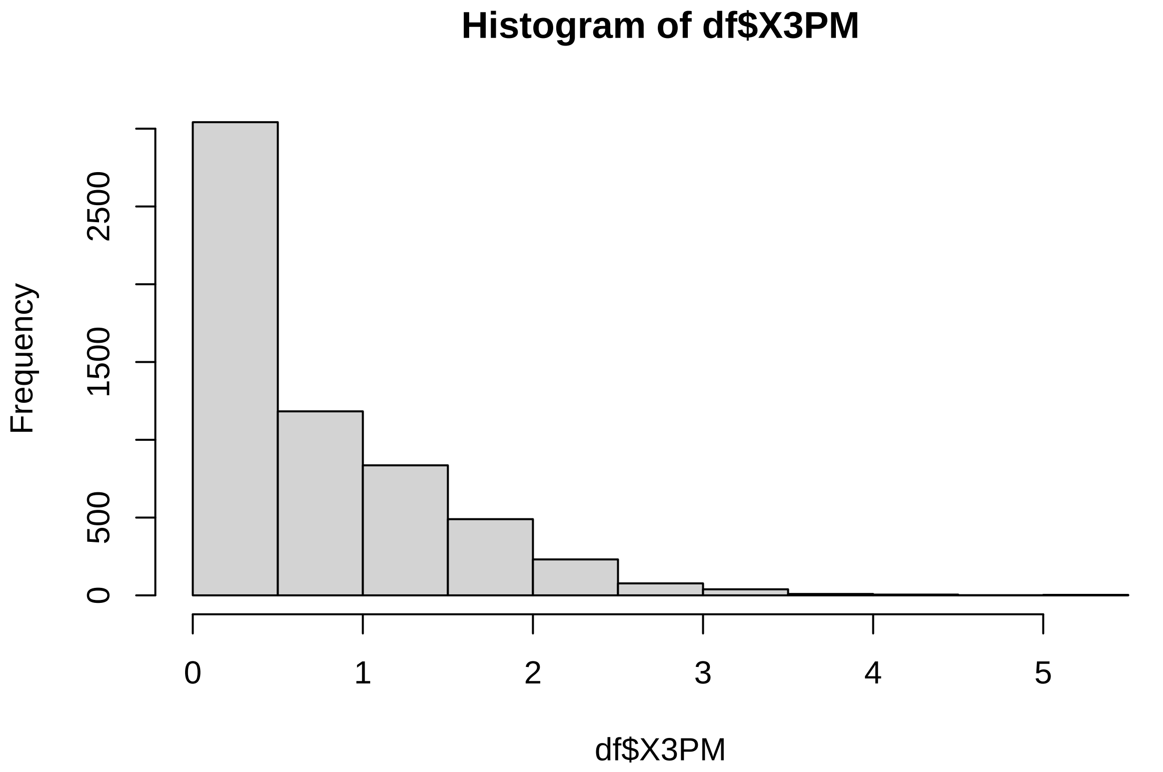
```
newdata<-newdata[newdata$GP > 20,]
```

Histogram of Original Data

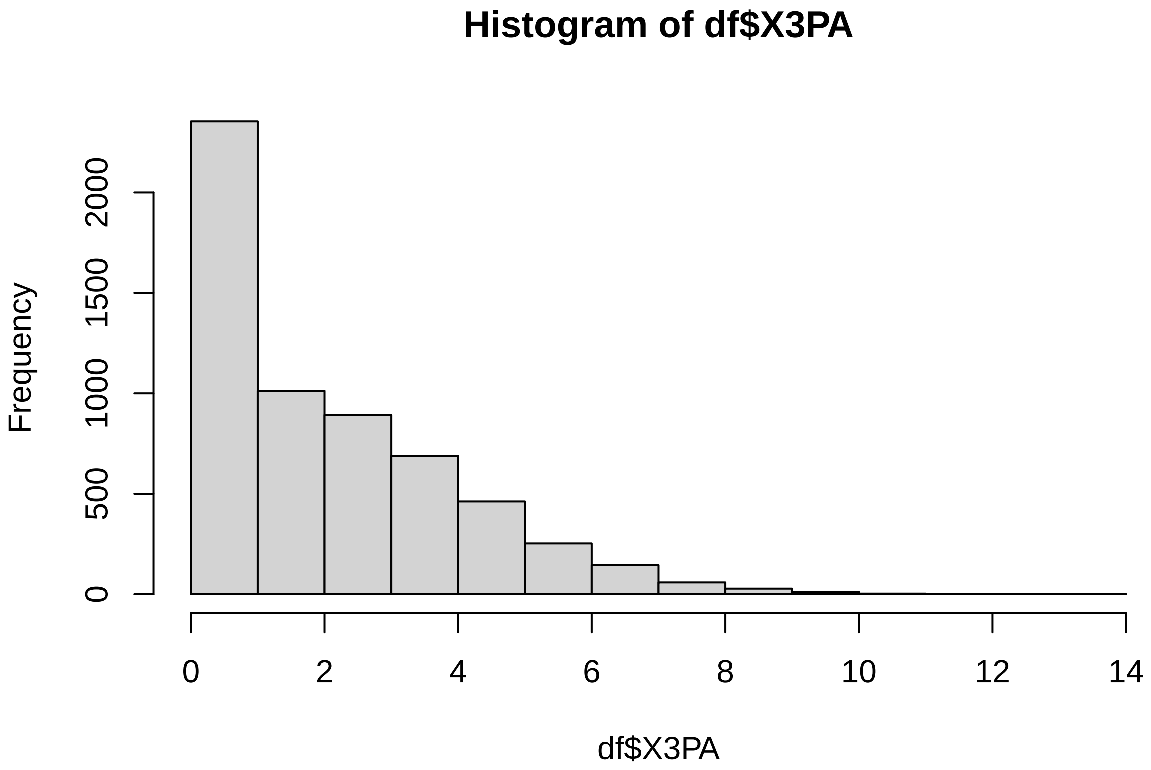
```
hist(df$X3P.)
```



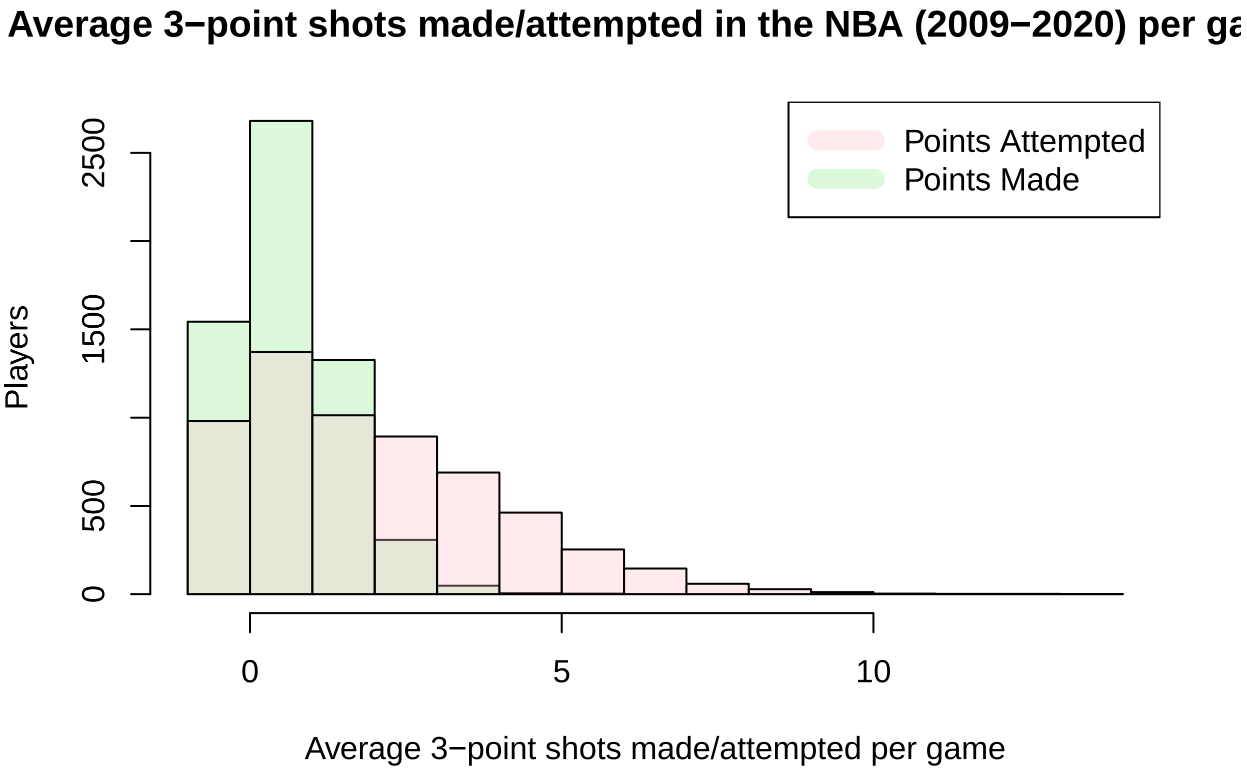
```
hist(df$X3PM)
```



```r
hist(df$X3PA)
```

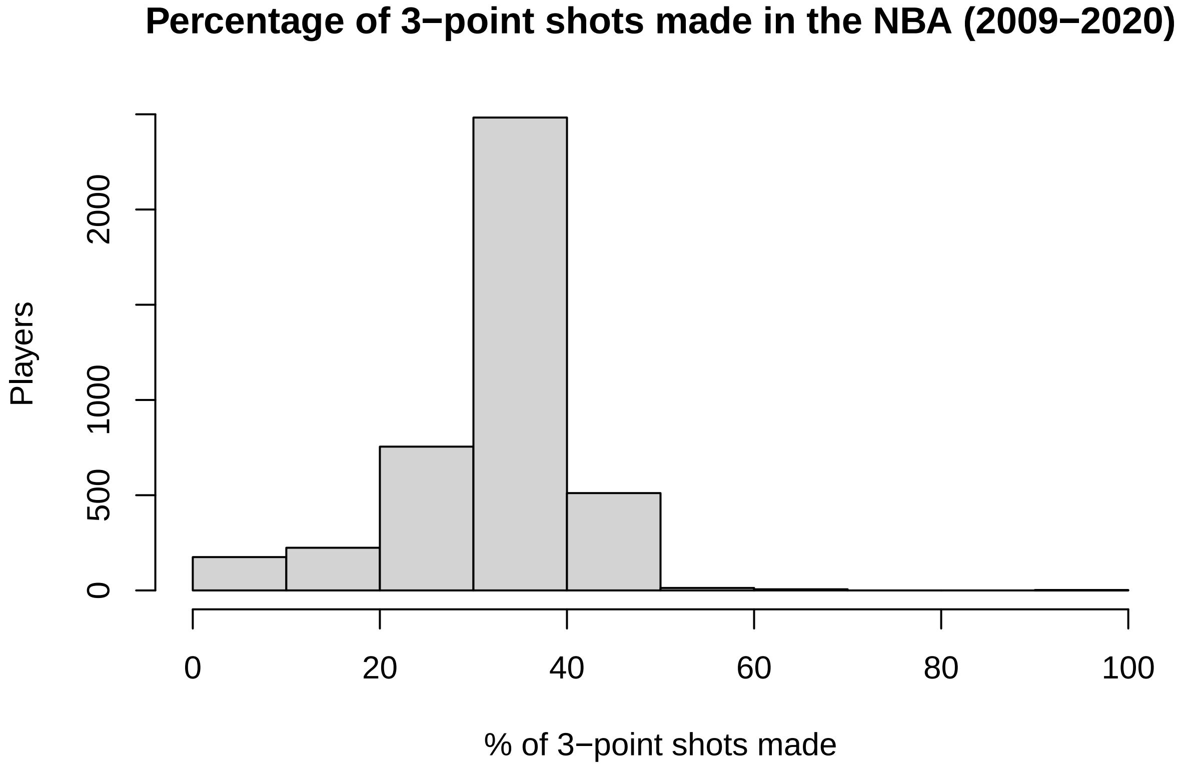


```r
c1 <- rgb(144,238,144,max = 255, alpha = 80, names = "lt.green")
c2 <- rgb(255,192,203, max = 255, alpha = 80, names = "lt.pink")

b <- min(c(df$X3PM,df$X3PA)) - 1
e <- max(c(df$X3PM,df$X3PA)) + 1
ax <- pretty(b:e, n=15)
hgM<-hist(df$X3PM, breaks=ax, plot=FALSE)
hgA<-hist(df$X3PA, breaks=ax, plot=FALSE)

plot(hgM, col=c1, main = "Average 3-point shots made/attempted in the NBA (2009-2020) per game", xlab="
plot(hgA, col=c2, add=TRUE)
legend("topright", c("Points Attempted","Points Made"), col=c(c2,c1), lwd=10)
```

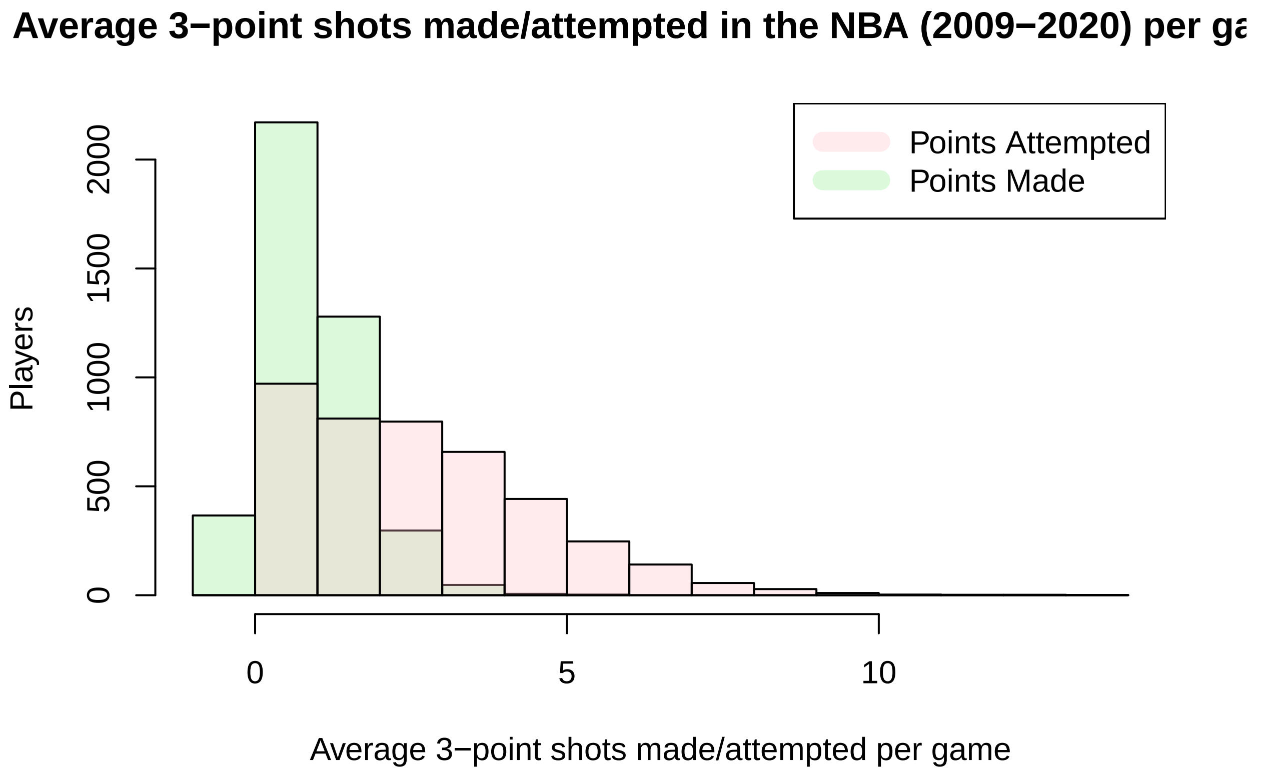


Histogram of those who attempted threes and played at least one-quarter of season

```
hist(newdata$X3P., main = "Percentage of 3-point shots made in the NBA (2009-2020)", xlab="% of 3-point
```



```
c1 <- rgb(144,238,144,max = 255, alpha = 80, names = "lt.green")
c2 <- rgb(255,192,203, max = 255, alpha = 80, names = "lt.pink")

b <- min(c(newdata$X3PM,newdata$X3PA)) - 1
e <- max(c(newdata$X3PM,newdata$X3PA)) + 1
ax <- pretty(b:e, n=15)
hgM<-hist(newdata$X3PM, breaks=ax, plot=FALSE)
hgA<-hist(newdata$X3PA, breaks=ax, plot=FALSE)

plot(hgM, col=c1, main = "Average 3-point shots made/attempted in the NBA (2009-2020) per game", xlab="
plot(hgA, col=c2, add=TRUE)
legend("topright", c("Points Attempted","Points Made"), col=c(c2,c1), lwd=10)
```



Read in data for just a single season

```
df2<-read.csv("threes_20_21.csv", stringsAsFactors = TRUE)
head(df2)
```

```
##              PLAYER TEAM GP X3PM X3PA X3P.
## 1        Trae Young  ATL 63  2.2  6.3 34.3
## 2      John Collins  ATL 63  1.3  3.3 39.9
## 3 Bogdan Bogdanovic  ATL 44  3.3  7.6 43.8
## 4      Clint Capela  ATL 63  0.0  0.0  0.0
## 5     De'Andre Hunter  ATL 23  1.3  4.1 32.6
## 6  Danilo Gallinari  ATL 51  2.0  5.0 40.6
```

Read in winning data

```r
wins<-read.csv("wins.csv", stringsAsFactors=TRUE)
head(wins)
```

```
##            Team    wp  year
## 1          Utah 0.699 20_21
## 2       Phoenix 0.692 20_21
## 3  Philadelphia 0.667 20_21
## 4      Brooklyn 0.655 20_21
## 5     Milwaukee 0.653 20_21
## 6   LA Clippers 0.626 20_21
```

```r
wins_20_21<-wins[wins$year == "20_21",]
wins_20_21$TEAM<-c("UTA","PHX","PHI","BKN","MIL","LAC","DEN","LAL","DAL","ATL","POR","NYK","GSW","MIA",
wins_20_21<-wins_20_21[-c(3)]
head(wins_20_21)
```

```
##            Team    wp TEAM
## 1          Utah 0.699  UTA
## 2       Phoenix 0.692  PHX
## 3  Philadelphia 0.667  PHI
## 4      Brooklyn 0.655  BKN
## 5     Milwaukee 0.653  MIL
## 6   LA Clippers 0.626  LAC
```

```r
key<-wins_20_21[-c(2)] #create a key to use as a common table to merge each subset of data with team na
key<-rbind(key,c("New Jersey","NJN")) #Add in teams that were later disbanded.
year_list<-unique(wins$year)
year_list<-as.character(year_list)

dfnew<-subset(df,year=="09_10", select=PLAYER:year)
wins_year<-wins[wins$year == "09_10",]
merge_ready<-merge(x=wins_year,y=key,all=TRUE)
merge_complete1<-merge(x=merge_ready,y=dfnew)

dfnew<-subset(df,year=="10_11", select=PLAYER:year)
wins_year<-wins[wins$year == "10_11",]
merge_ready<-merge(x=wins_year,y=key,all=TRUE)
merge_complete2<-merge(x=merge_ready,y=dfnew)

dfnew<-subset(df,year=="11_12", select=PLAYER:year)
wins_year<-wins[wins$year == "11_12",]
merge_ready<-merge(x=wins_year,y=key,all=TRUE)
merge_complete3<-merge(x=merge_ready,y=dfnew)

dfnew<-subset(df,year=="12_13", select=PLAYER:year)
wins_year<-wins[wins$year == "12_13",]
merge_ready<-merge(x=wins_year,y=key,all=TRUE)
merge_complete4<-merge(x=merge_ready,y=dfnew)

dfnew<-subset(df,year=="13_14", select=PLAYER:year)
wins_year<-wins[wins$year == "13_14",]
merge_ready<-merge(x=wins_year,y=key,all=TRUE)
```

```r
merge_complete5<-merge(x=merge_ready,y=dfnew)

dfnew<-subset(df,year=="14_15", select=PLAYER:year)
wins_year<-wins[wins$year == "14_15",]
merge_ready<-merge(x=wins_year,y=key,all=TRUE)
merge_complete6<-merge(x=merge_ready,y=dfnew)

dfnew<-subset(df,year=="15_16", select=PLAYER:year)
wins_year<-wins[wins$year == "15_16",]
merge_ready<-merge(x=wins_year,y=key,all=TRUE)
merge_complete7<-merge(x=merge_ready,y=dfnew)

dfnew<-subset(df,year=="16_17", select=PLAYER:year)
wins_year<-wins[wins$year == "16_17",]
merge_ready<-merge(x=wins_year,y=key,all=TRUE)
merge_complete8<-merge(x=merge_ready,y=dfnew)

dfnew<-subset(df,year=="17_18", select=PLAYER:year)
wins_year<-wins[wins$year == "17_18",]
merge_ready<-merge(x=wins_year,y=key,all=TRUE)
merge_complete9<-merge(x=merge_ready,y=dfnew)

dfnew<-subset(df,year=="18_19", select=PLAYER:year)
wins_year<-wins[wins$year == "18_19",]
merge_ready<-merge(x=wins_year,y=key,all=TRUE)
merge_complete10<-merge(x=merge_ready,y=dfnew)

dfnew<-subset(df,year=="19_20", select=PLAYER:year)
wins_year<-wins[wins$year == "19_20",]
merge_ready<-merge(x=wins_year,y=key,all=TRUE)
merge_complete11<-merge(x=merge_ready,y=dfnew)

dfnew<-subset(df,year=="20_21", select=PLAYER:year)
wins_year<-wins[wins$year == "20_21",]
merge_ready<-merge(x=wins_year,y=key,all=TRUE)
merge_complete12<-merge(x=merge_ready,y=dfnew)

complete_cases<-rbind(merge_complete1,merge_complete2,merge_complete3,merge_complete4,merge_complete5,me

threes_and_wins<-merge(x=df2, y=wins_20_21, by="TEAM", all=TRUE)
#survivors <- subset(Titanic, select=Age, subset=Survived=="1", drop=T)
```

```r
#install.packages("lemon")
library(lemon)
knit_print.data.frame <- lemon_print
```

Group By Team and Print Pretty

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
by_team<- threes_and_wins %>% group_by(TEAM)
head(by_team)
```

Table 1: Winning Percentage by Team combined with player 3PT Percentage

| TEAM | PLAYER | GP | X3PM | X3PA | X3P. | Team | wp |
|---|---|---|---|---|---|---|---|
| ATL | Trae Young | 63 | 2.2 | 6.3 | 34.3 | Atlanta | 0.567 |
| ATL | John Collins | 63 | 1.3 | 3.3 | 39.9 | Atlanta | 0.567 |
| ATL | Bogdan Bogdanovic | 44 | 3.3 | 7.6 | 43.8 | Atlanta | 0.567 |
| ATL | Clint Capela | 63 | 0.0 | 0.0 | 0.0 | Atlanta | 0.567 |
| ATL | De'Andre Hunter | 23 | 1.3 | 4.1 | 32.6 | Atlanta | 0.567 |
| ATL | Danilo Gallinari | 51 | 2.0 | 5.0 | 40.6 | Atlanta | 0.567 |

```
by_team_and_year<- complete_cases %>%
  group_by(TEAM, Team, year, wp) %>%
  summarize_at(vars(X3P.),list(X3P.=mean))
head(by_team_and_year)
```

Table 2: Winning Percentage by Team combined with player 3PT Percentage

| TEAM | Team | year | wp | X3P. |
|---|---|---|---|---|
| ATL | Atlanta | 09_10 | 0.613 | 22.44286 |
| ATL | Atlanta | 10_11 | 0.532 | 30.56429 |
| ATL | Atlanta | 11_12 | 0.583 | 26.98667 |
| ATL | Atlanta | 12_13 | 0.523 | 27.03125 |
| ATL | Atlanta | 13_14 | 0.461 | 25.32500 |
| ATL | Atlanta | 14_15 | 0.694 | 33.59333 |

```
by_team<-tapply(threes_and_wins$X3P.,threes_and_wins$TEAM,mean)

newdf<-as.data.frame(by_team, row.names=NULL)
newdf$TEAM<-rownames(by_team)
rownames(newdf)<-NULL

head(by_team_and_year)
```

Table 3: Winning Percentage by Team combined with player 3PT Percentage

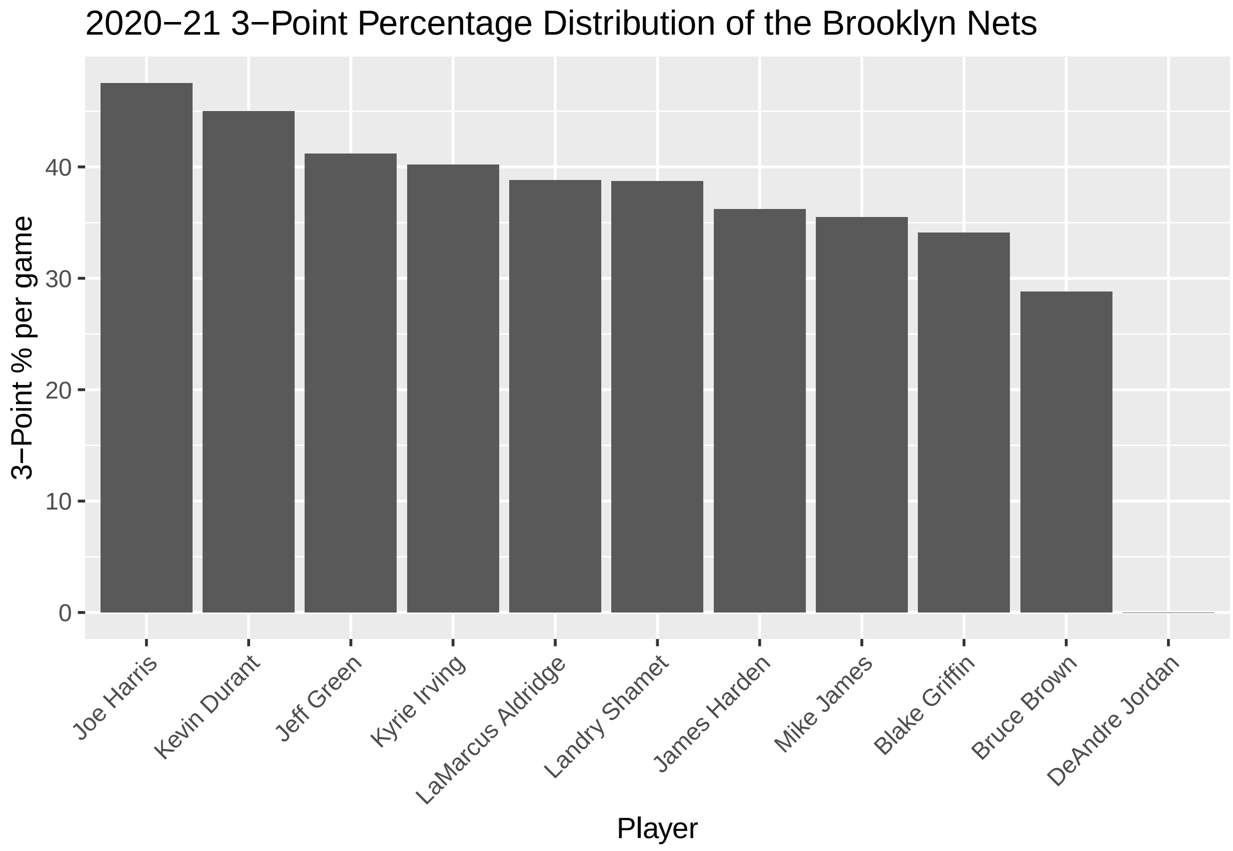
| TEAM | Team | year | wp | X3P. |
|---|---|---|---:|---:|
| ATL | Atlanta | 09_10 | 0.613 | 22.44286 |
| ATL | Atlanta | 10_11 | 0.532 | 30.56429 |
| ATL | Atlanta | 11_12 | 0.583 | 26.98667 |
| ATL | Atlanta | 12_13 | 0.523 | 27.03125 |
| ATL | Atlanta | 13_14 | 0.461 | 25.32500 |
| ATL | Atlanta | 14_15 | 0.694 | 33.59333 |

Subset to Brooklyn Nets

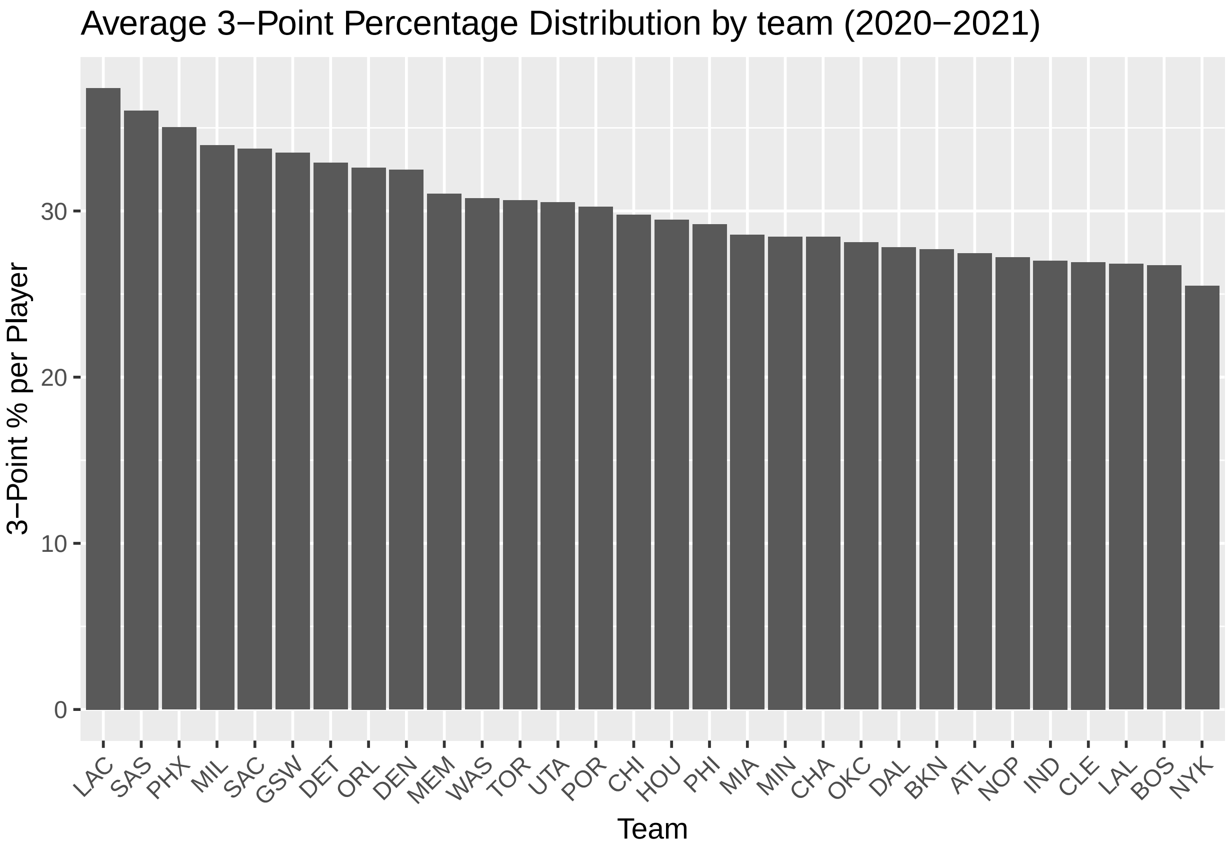
```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:lemon':
##
##     CoordCartesian, element_render
```

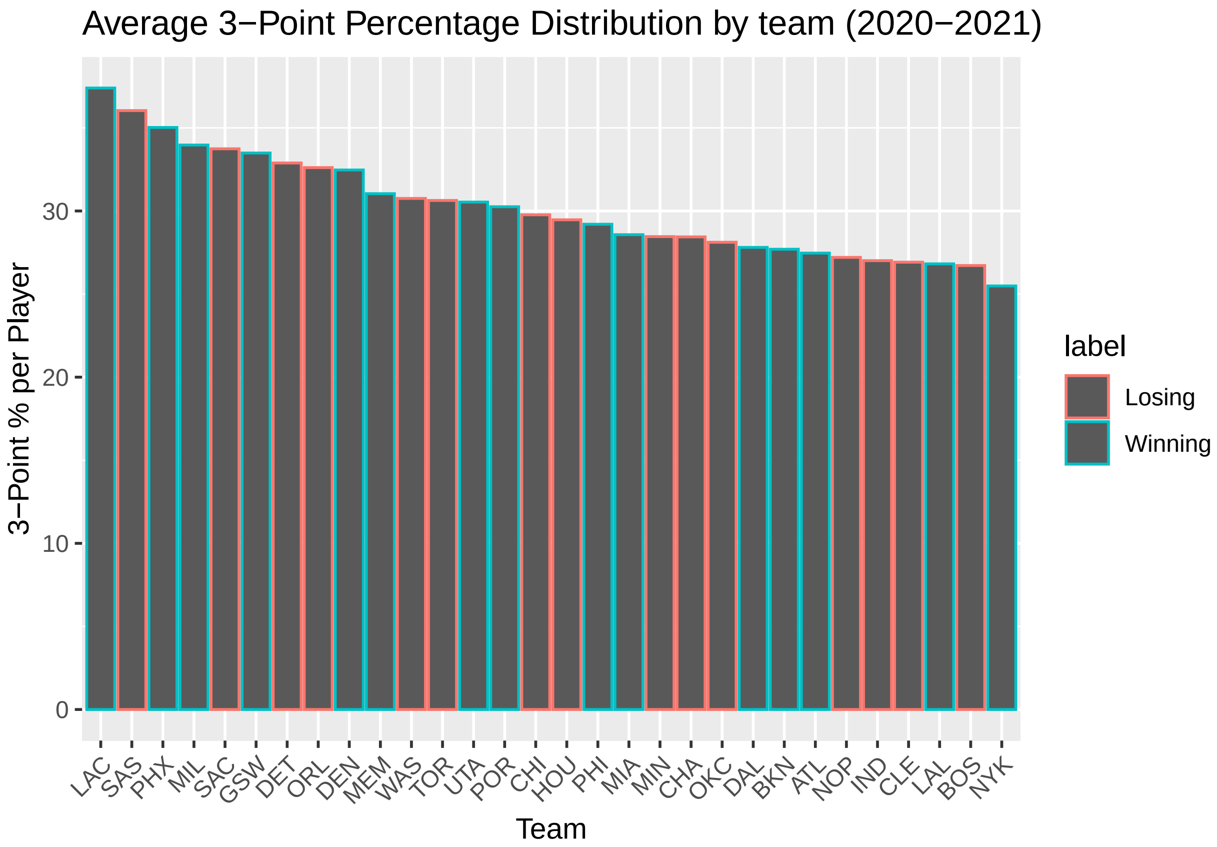
```
nets<-subset(threes_and_wins, select=c(PLAYER,X3P.),subset=TEAM=="BKN", drop=T)
nets<-head(nets,11)
nets$PLAYER<-as.factor(as.character(nets$PLAYER))
#barplot(nets$X3P.~nets$PLAYER, col="maroon", cex.names=0.5, las=2)
ggplot(nets, aes(x=reorder(`PLAYER`,-`X3P.`),y=`X3P.`)) + geom_bar(stat="identity") + theme(axis.text.x
```



```
newdf2<-merge(newdf,wins_20_21,all=TRUE)
newdf2$label<-as.factor(ifelse(newdf2$wp>.5,"Winning","Losing"))
#Plot without winning/losing labels
ggplot(newdf, aes(x=reorder(`TEAM`,-`by_team`),y=`by_team`)) + geom_bar(stat="identity") + theme(axis.te
```



```
#Plot with winning/losing labels
ggplot(newdf2, aes(x=reorder(`TEAM`,-`by_team`),y=`by_team`,color=`label`)) + geom_bar(stat="identity")
```

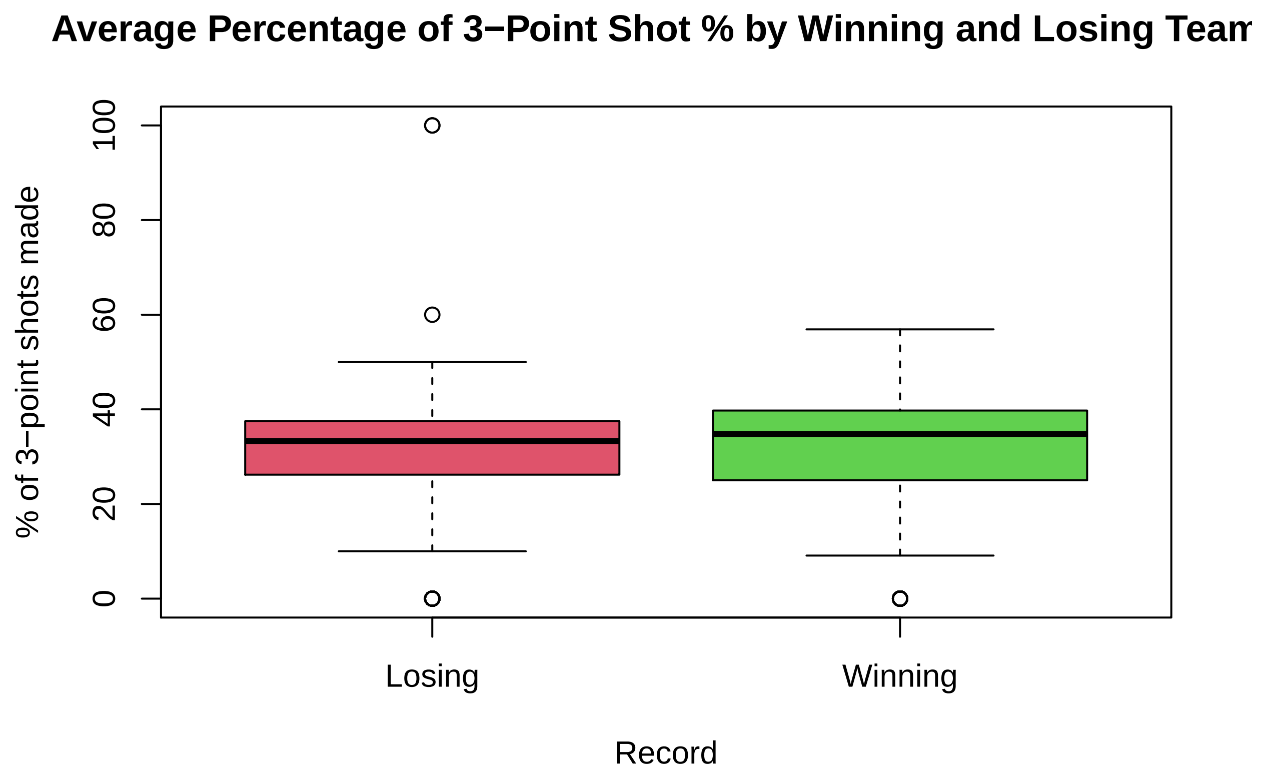


Subset to winners and losers for hypothesis testing

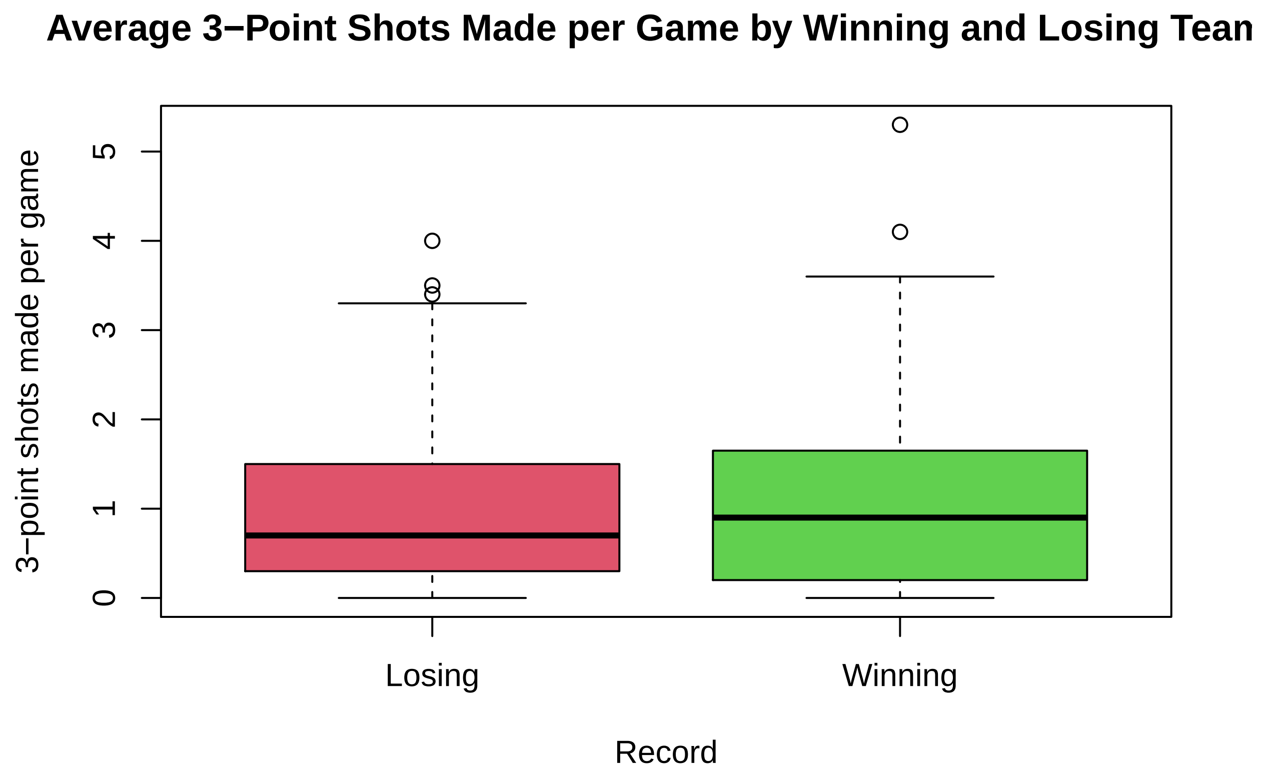
```
threes_and_wins$label<-as.factor(ifelse(threes_and_wins$wp>.5,"Winning","Losing"))
winning_team<- subset(threes_and_wins, select=X3P., subset=label=="Winning", drop=T)
losing_team<- subset(threes_and_wins, select=X3P., subset=label=="Losing", drop=T)

complete_cases$label<-as.factor(ifelse(complete_cases$wp>.5,"Winning","Losing"))
winning_team_full<- subset(complete_cases, select=X3P., subset=label=="Winning", drop=T)
losing_team_full<- subset(complete_cases, select=X3P., subset=label=="Losing", drop=T)
```

Boxplot

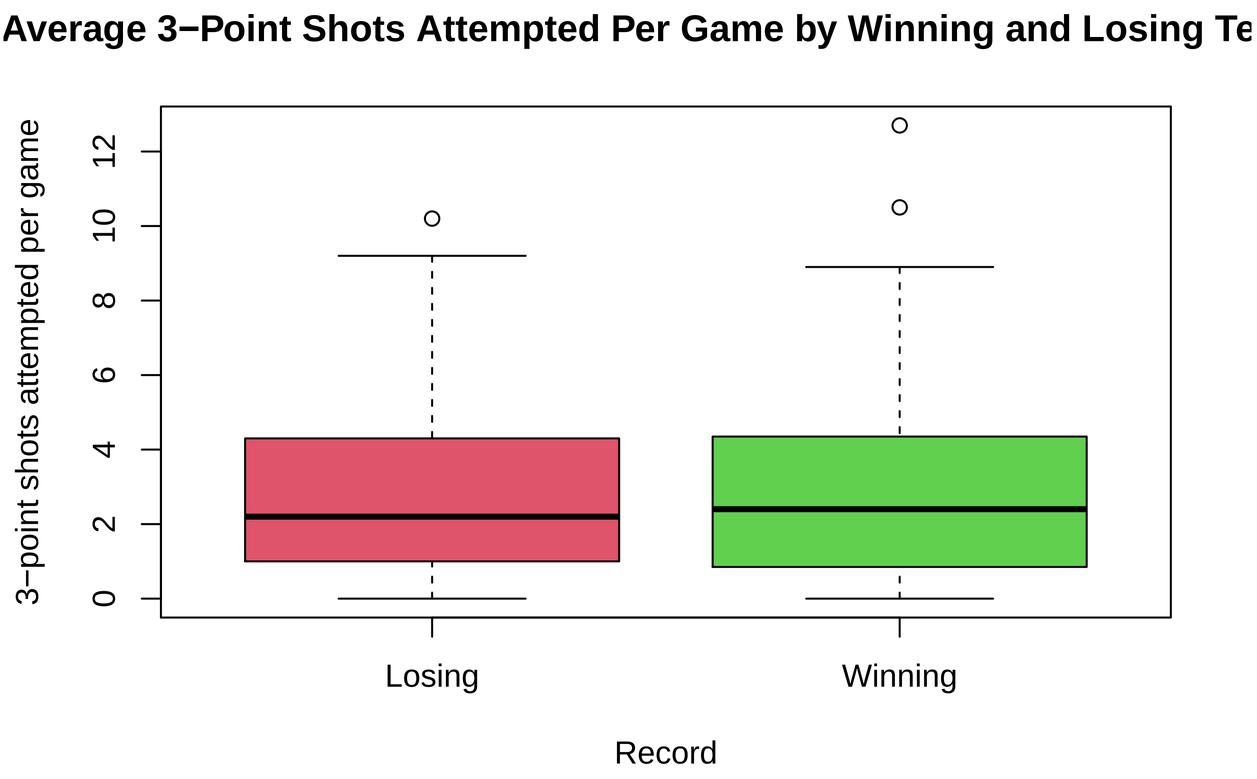
```
boxplot(X3P.~label, data=threes_and_wins, col=c(2,3), main = "Average Percentage of 3-Point Shot % by Wi
```



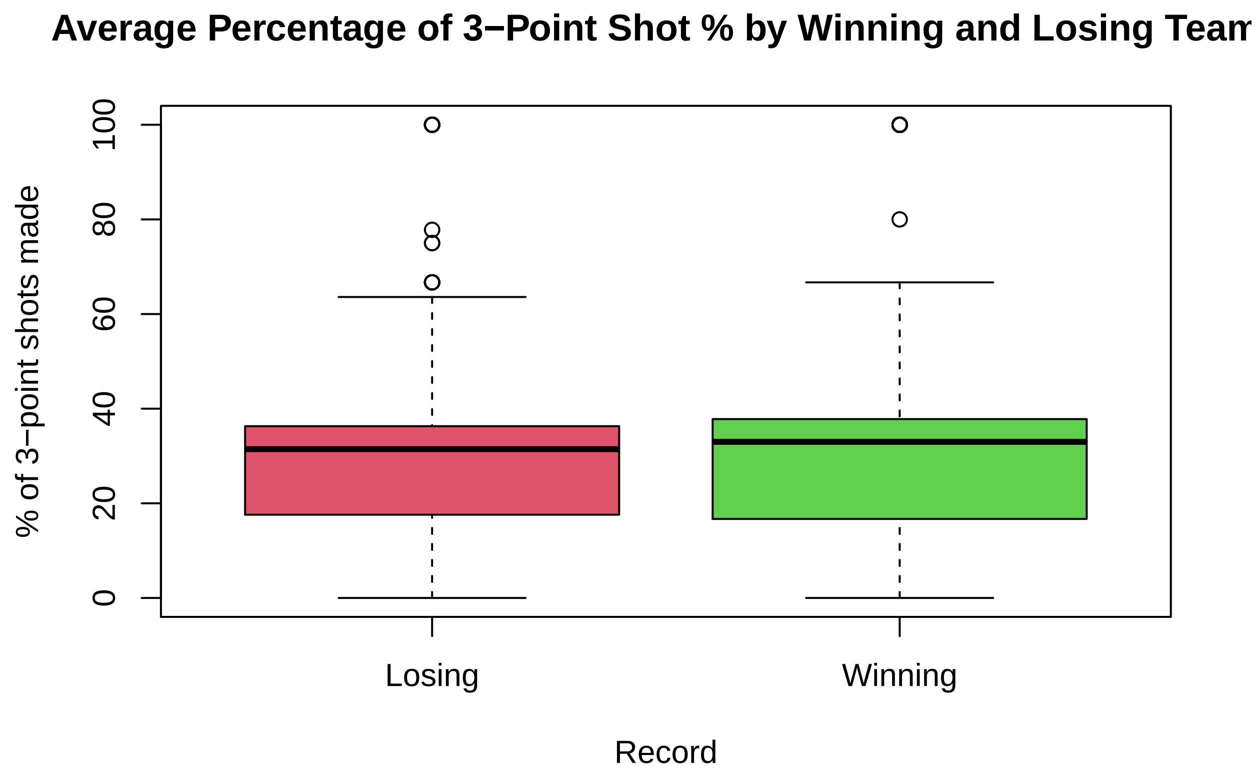
```
boxplot(X3PM~label, data=threes_and_wins, col=c(2,3), main = "Average 3-Point Shots Made per Game by Wi
```



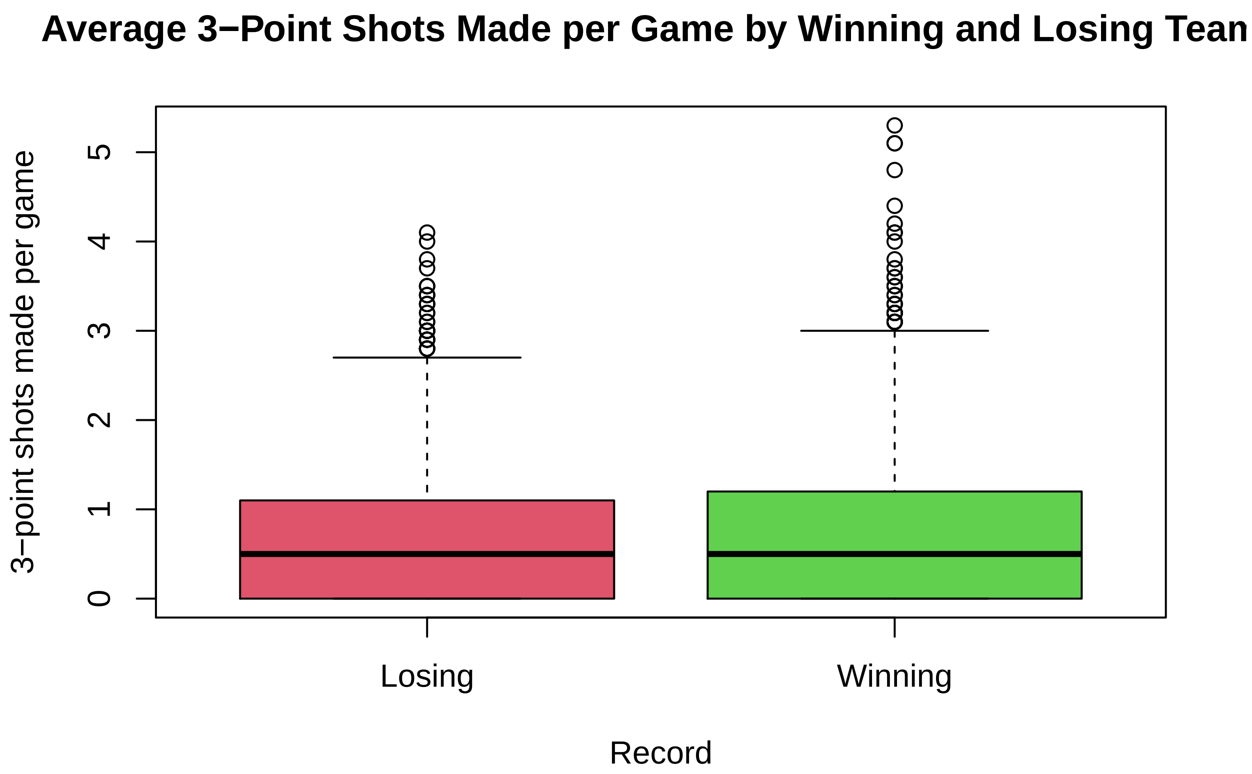
```
boxplot(X3PA~label, data=threes_and_wins, col=c(2,3), main = "Average 3-Point Shots Attempted Per Game
```



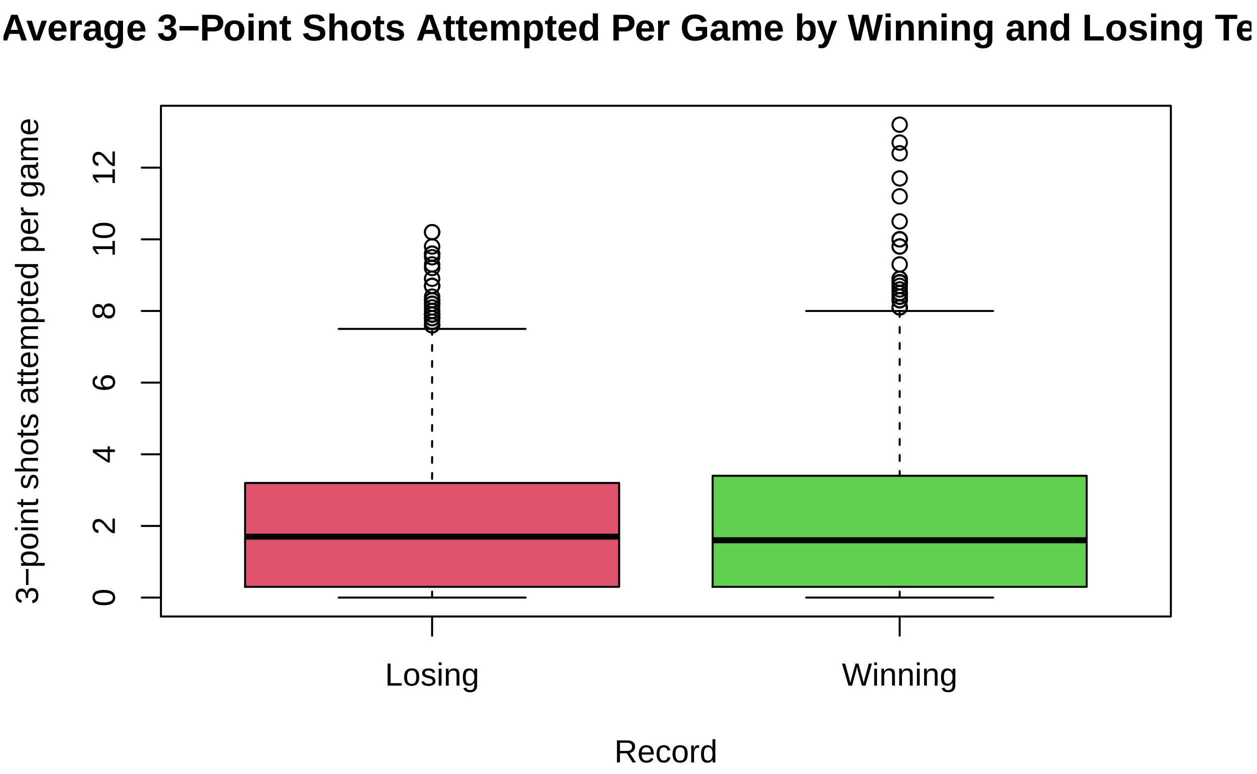
```
boxplot(X3P.~label, data=complete_cases, col=c(2,3), main = "Average Percentage of 3-Point Shot % by Wi
```



```
boxplot(X3PM~label, data=complete_cases, col=c(2,3), main = "Average 3-Point Shots Made per Game by Winn
```



```
boxplot(X3PA~label, data=complete_cases, col=c(2,3), main = "Average 3-Point Shots Attempted Per Game by
```



Perform t-test

```
t.test(winning_team, losing_team, alt="greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  winning_team and losing_team
## t = 0.4919, df = 537.82, p-value = 0.3115
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -1.358588       Inf
## sample estimates:
## mean of x mean of y
##  30.48327  29.90505
```

```
t.test(winning_team_full, losing_team_full, alt="greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  winning_team_full and losing_team_full
## t = 2.4127, df = 5759.5, p-value = 0.007934
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.3298677       Inf
## sample estimates:
## mean of x mean of y
##  27.05827  26.02138
```

Require at least 1 3-point attempt

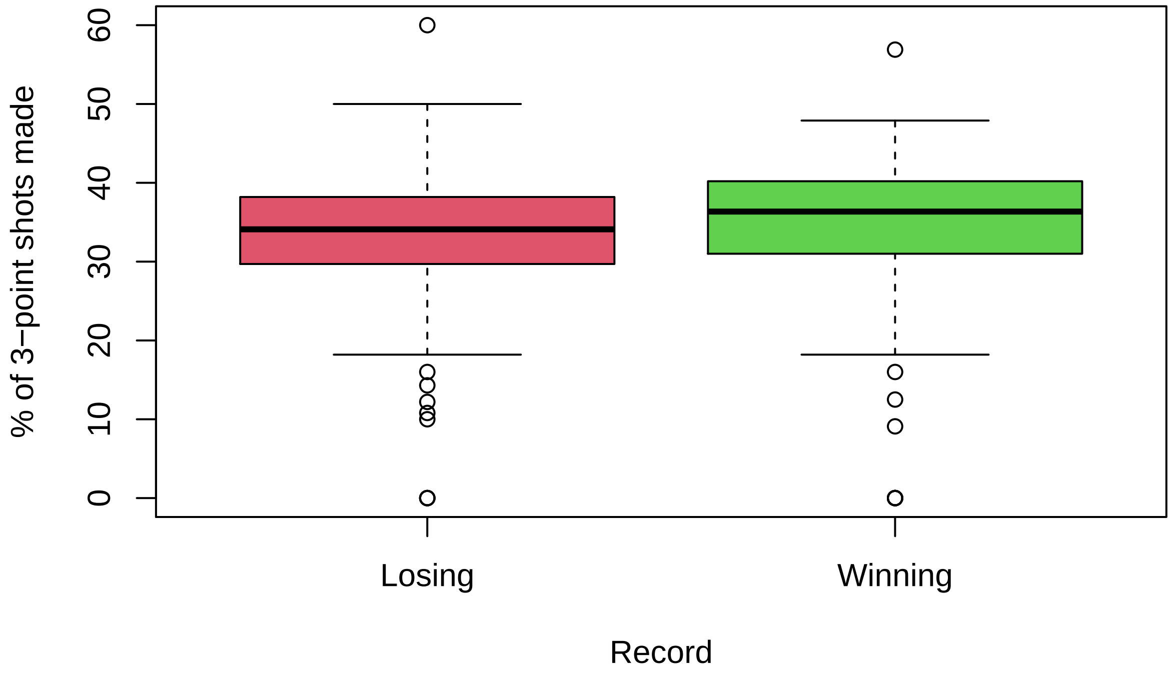
```
newdata2<-threes_and_wins[threes_and_wins$X3PA > 0.0,]
```

Require at least 25% of games played

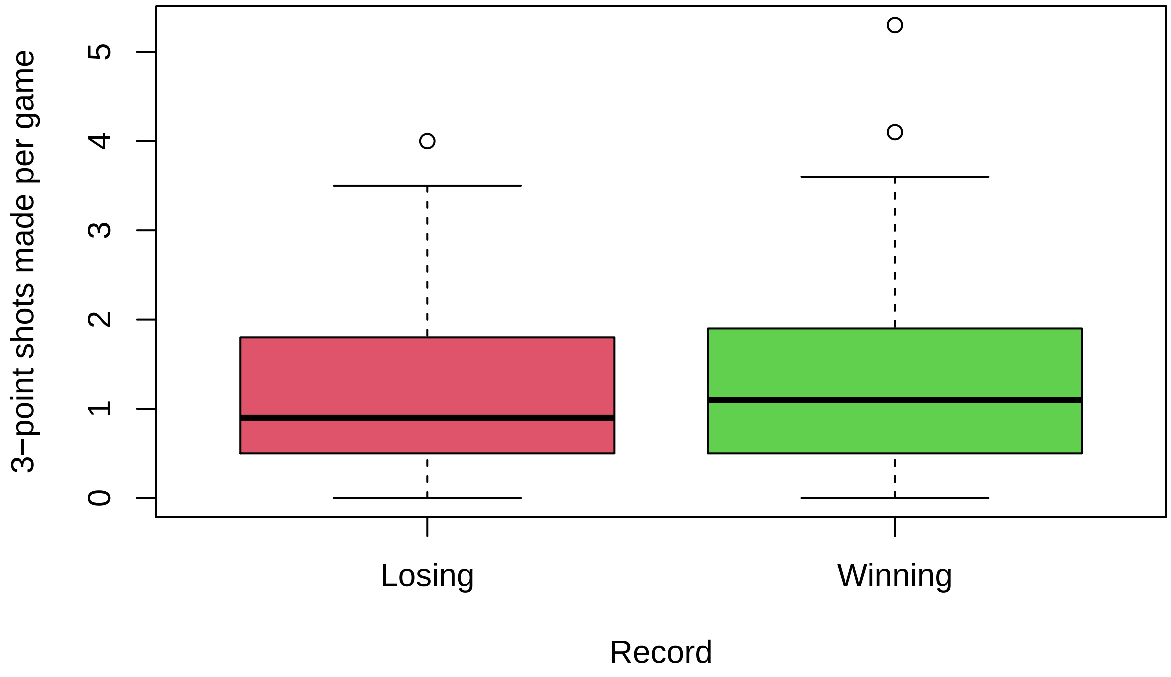
```
newdata2<-newdata2[newdata2$GP > 20,]
```

Boxplot

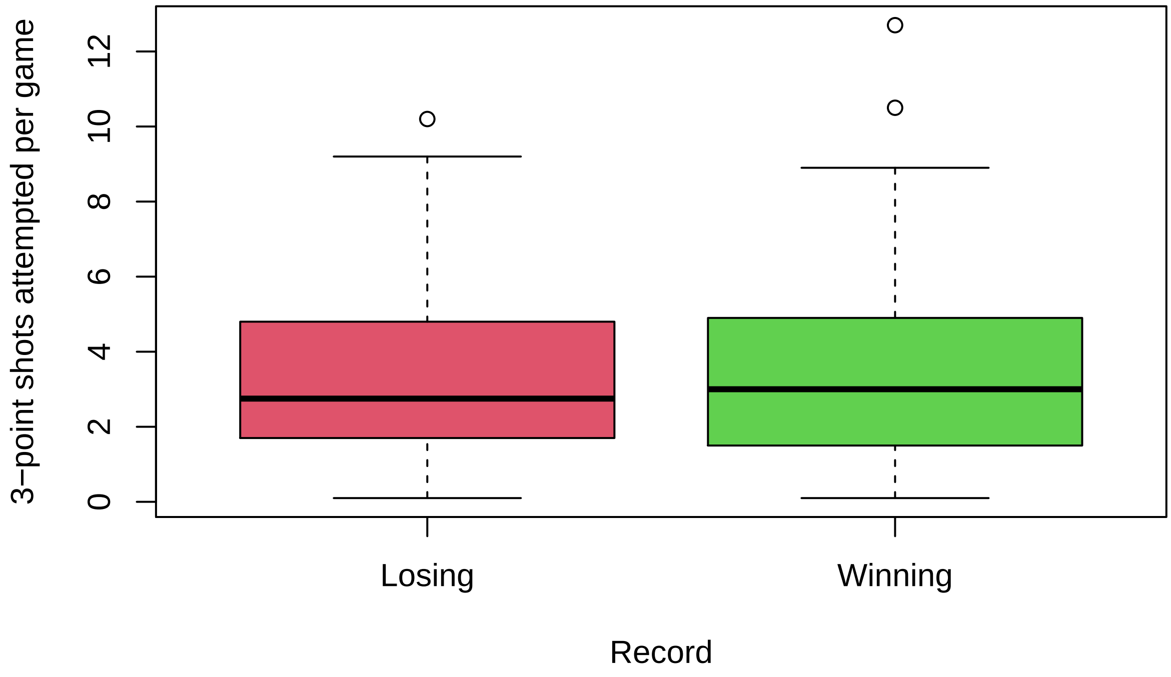
```
boxplot(X3P.~label, data=newdata2, col=c(2,3), xlab="Record", ylab="% of 3-point shots made")
```



```
boxplot(X3PM~label, data=newdata2, col=c(2,3), xlab="Record", ylab="3-point shots made per game")
```



```
boxplot(X3PA~label, data=newdata2, col=c(2,3), xlab="Record", ylab="3-point shots attempted per game")
```



Bootstrapping

```
library(boot)
n <- length(complete_cases)
N <- 10000

percentage.mean <- numeric(N)
for (i in 1:N)
{
  x <- sample(complete_cases$X3P., n, replace = TRUE)
  percentage.mean[i] <- mean(x)/100 #bootstrap sample mean
}

(interval<-quantile(percentage.mean, c(0.025, 0.975))) #95% confidence interval
```

```
##      2.5%     97.5%
## 0.1630975 0.3647025
```

```
boot.mean.percentage <- replicate(10000, mean(sample(complete_cases$X3P., #All teams boot mean
                                                     replace = T)))

quantile(boot.mean.percentage, c(.025, .975))
```

```
##     2.5%    97.5%
## 26.10022 26.95098
```

```
boot.mean.winning <- replicate(10000, mean(sample(winning_team_full, #Winning teams boot mean
                                                     replace = T)))

quantile(boot.mean.winning, c(.025, .975))
```

```
##     2.5%    97.5%
## 26.43964 27.67943
```

```
boot.mean.losing <- replicate(10000, mean(sample(losing_team_full, #Losing teams boot mean
                                                   replace = T)))

quantile(boot.mean.losing, c(.025, .975))
```

```
##     2.5%    97.5%
## 25.44449 26.58489
```

```
mean(boot.mean.winning)-mean(boot.mean.losing)
```

```
## [1] 1.044207
```

```
(boot.sd=sd(boot.mean.percentage)) # bootstrap SE
```

```
## [1] 0.2166645
```

```
# Let's run the bootstrap
set.seed(1000)
## but when you repeat it for many times, estimate get closer to the original
sd.fn=function(data,ind){sd(data[ind])}#specify the index

boot.out= boot(complete_cases$X3P.,sd.fn,R=1000) #repeat this 1000 times
boot.out
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = complete_cases$X3P., statistic = sd.fn, R = 1000)
##
##
## Bootstrap Statistics :
##     original        bias    std. error
## t1* 16.40203 -0.0007200233   0.1925833
```

```
(boot.mean= mean(boot.mean.percentage)) # bootstrap mean
```

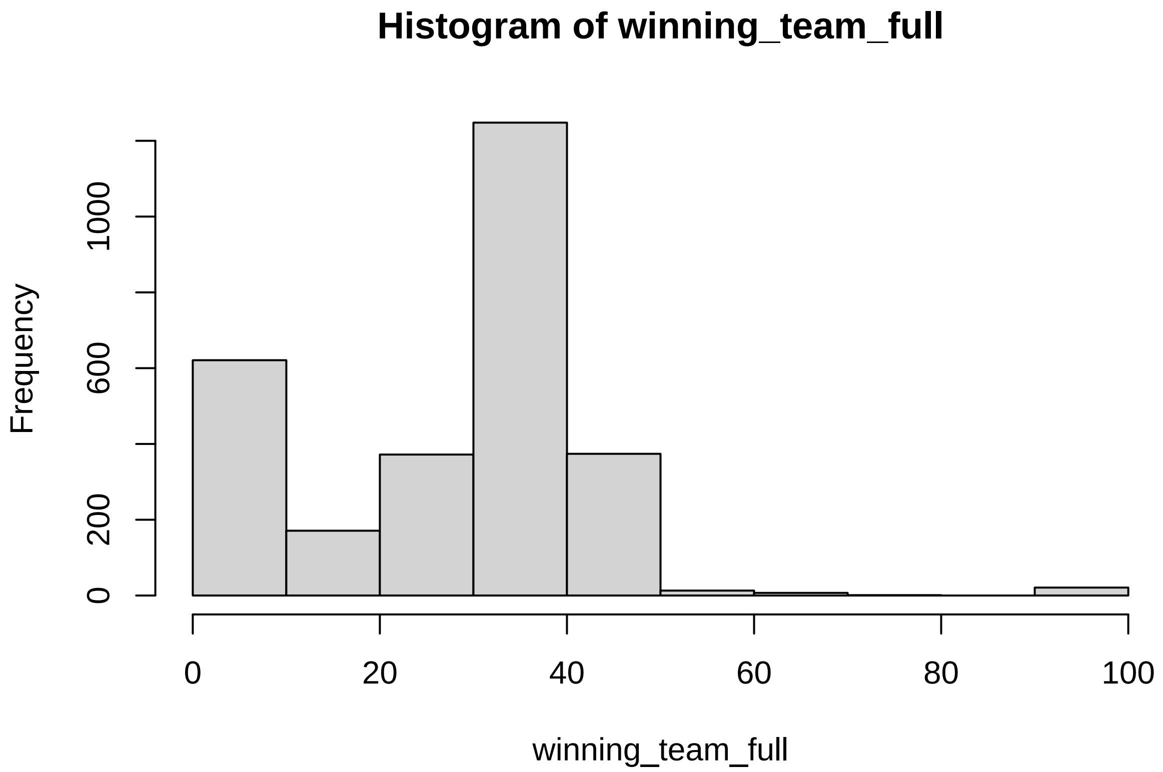
```
## [1] 26.52294
```

```
(bias= mean(boot.mean.percentage)-mean(complete_cases$X3P.)) # estimated bias
```

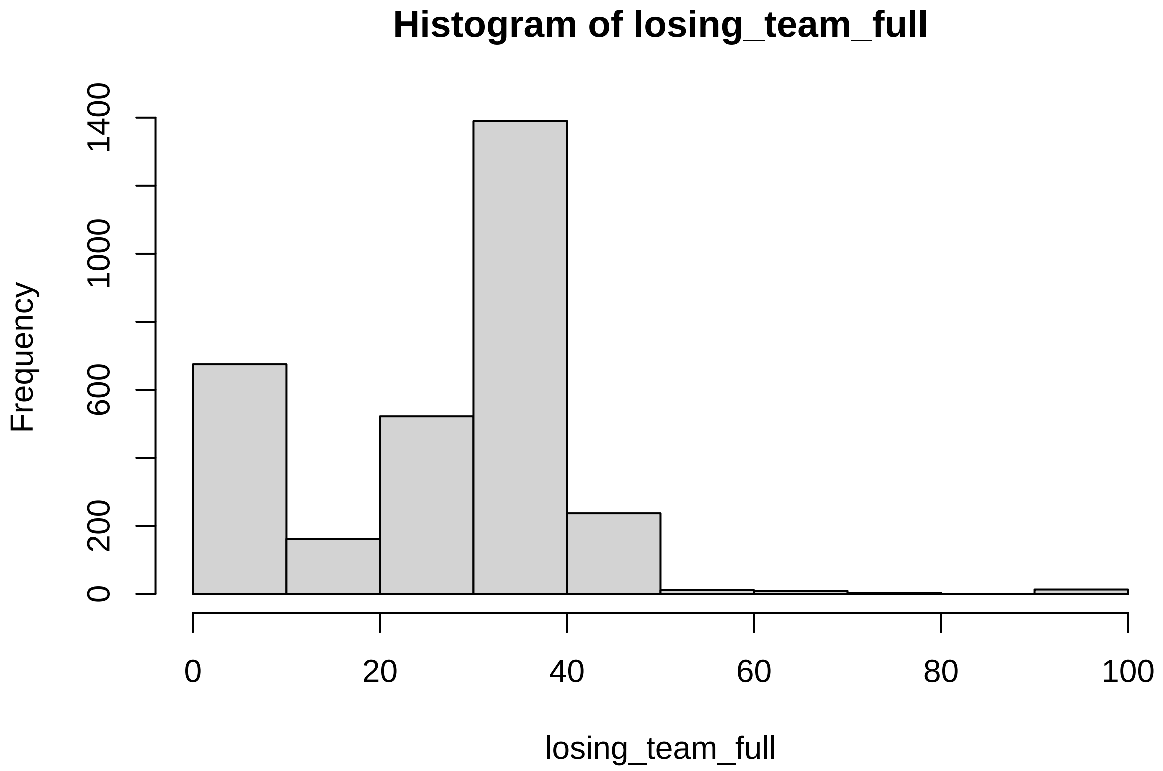
```
## [1] 0.0003116376
```

Winning and Losing Histogramss

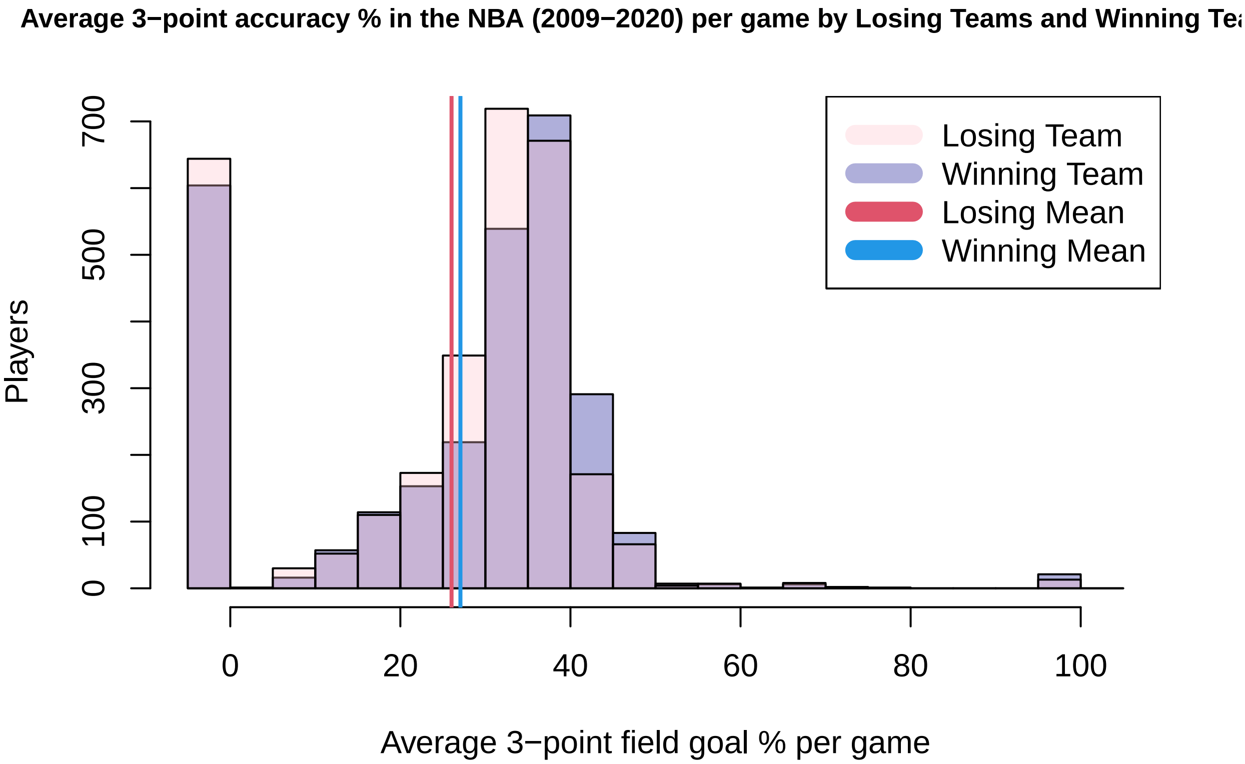
```
hist(winning_team_full)
```



```
hist(losing_team_full)
```

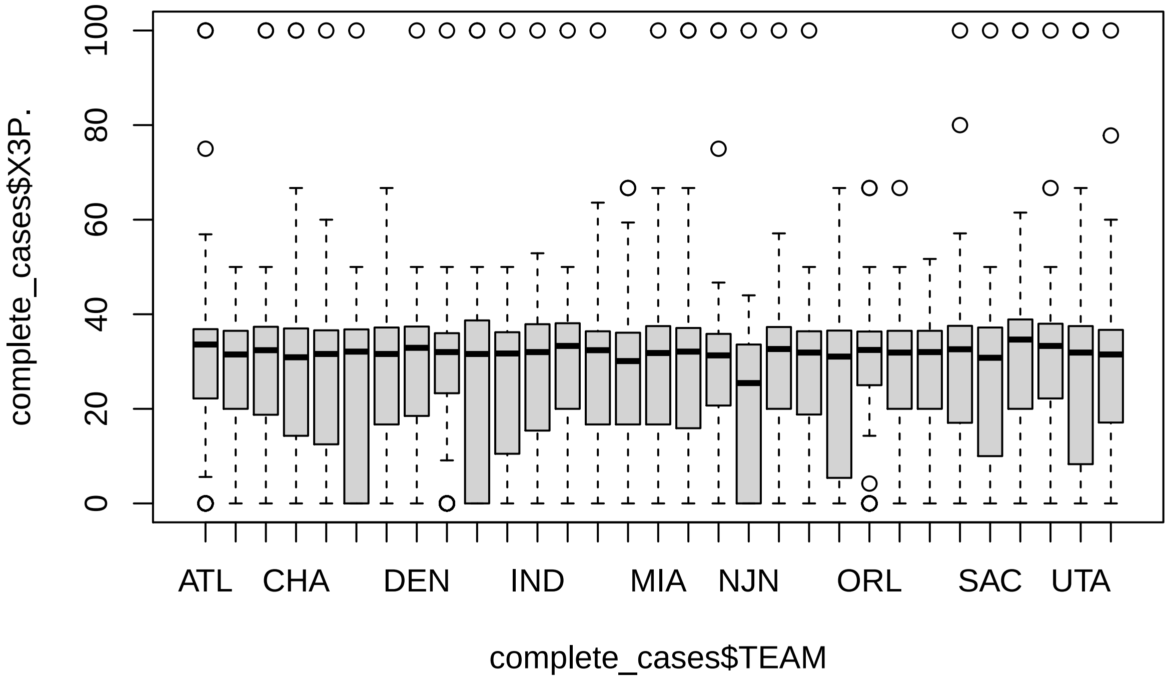


```
c1 <- rgb(0,0,139,max = 255, alpha = 80, names = "blue")
c2 <- rgb(255,192,203, max = 255, alpha = 80, names = "lt.pink")

b <- min(c(winning_team_full,losing_team_full)) - 1
e <- max(c(winning_team_full,losing_team_full)) + 1
ax <- pretty(b:e, n=15)
hgW<-hist(winning_team_full, breaks=ax, plot=FALSE)
hgL<-hist(losing_team_full, breaks=ax, plot=FALSE)

plot(hgW, col=c1, main = "Average 3-point accuracy % in the NBA (2009-2020) per game by Losing Teams and
plot(hgL, col=c2, add=TRUE)
abline(v = mean(boot.mean.winning), col = 4, lwd = 2)
abline(v = mean(boot.mean.losing), col = 2, lwd = 2)
legend("topright", c("Losing Team","Winning Team", "Losing Mean", "Winning Mean"), col=c(c2,c1,2,4), lwd
```

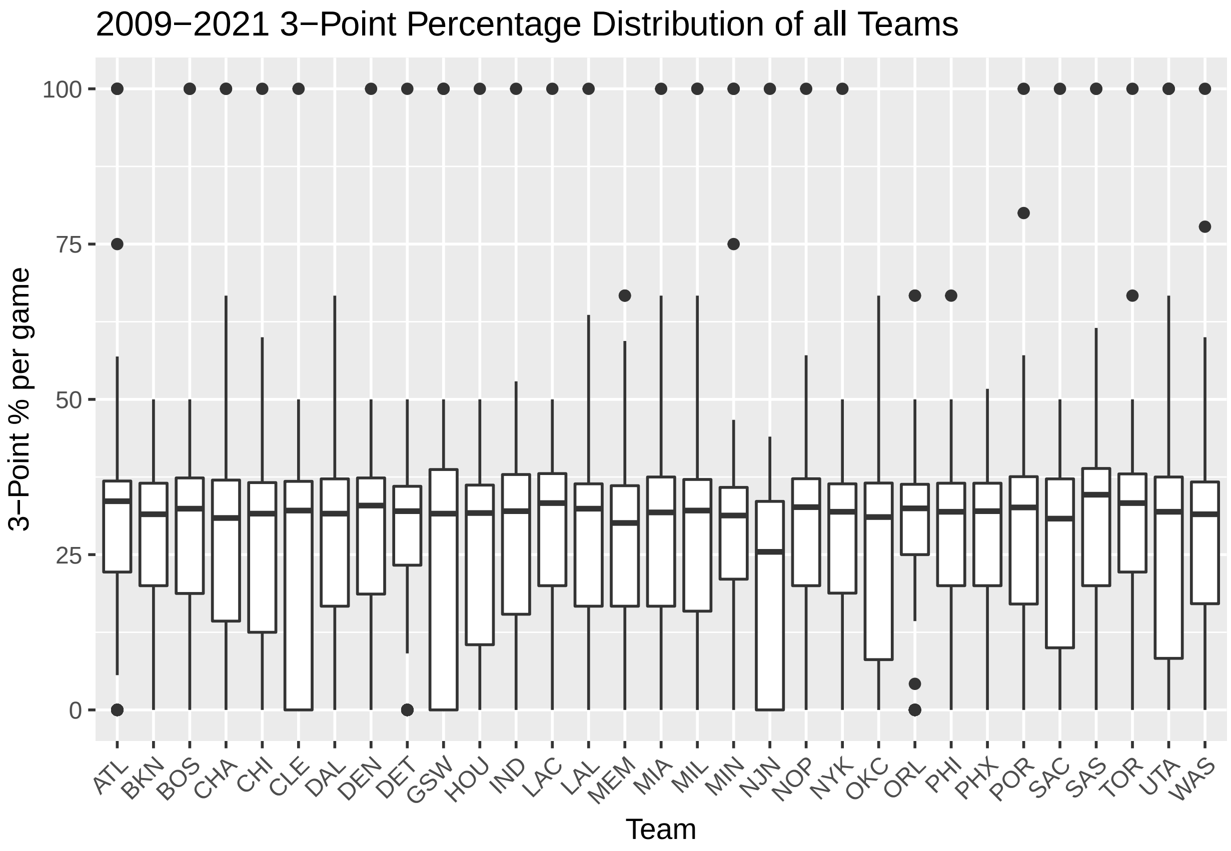


Team Comparisons

```
complete_cases$TEAM<-as.factor(complete_cases$TEAM)
plot(complete_cases$X3P.~complete_cases$TEAM)
```



```
ggplot(complete_cases, aes(x=`TEAM`,y=`X3P.`)) + geom_boxplot() + theme(axis.text.x = element_text(angl
```



Pretty Print

```
library(lemon)
knit_print.data.frame <- lemon_print
```

```
head(complete_cases)
```

Table 4: Winning Percentage by Team combined with player 3PT Percentage

| year | TEAM | Team | wp | PLAYER | GP | X3PM | X3PA | X3P. | label |
|---|---|---|---:|---|---:|---:|---:|---:|---|
| 09_10 | ATL | Atlanta | 0.613 | Joe Johnson | 76 | 1.7 | 4.6 | 36.9 | Winning |
| 09_10 | ATL | Atlanta | 0.613 | Jamal Crawford | 79 | 2.1 | 5.4 | 38.2 | Winning |
| 09_10 | ATL | Atlanta | 0.613 | Josh Smith | 81 | 0.0 | 0.1 | 0.0 | Winning |
| 09_10 | ATL | Atlanta | 0.613 | Al Horford | 81 | 0.0 | 0.0 | 100.0 | Winning |
| 09_10 | ATL | Atlanta | 0.613 | Marvin Williams | 81 | 0.5 | 1.6 | 30.3 | Winning |
| 09_10 | ATL | Atlanta | 0.613 | Mike Bibby | 80 | 1.6 | 4.1 | 38.9 | Winning |

# R Notebook

## Libraries

```
library(ggplot2)
library(XLConnect)
```

```
## XLConnect 1.0.5 by Mirai Solutions GmbH [aut],
##   Martin Studer [cre],
##   The Apache Software Foundation [ctb, cph] (Apache POI),
##   Graph Builder [ctb, cph] (Curvesapi Java library),
##   Brett Woolridge [ctb, cph] (SparseBitSet Java library)
```

```
## https://mirai-solutions.ch
## https://github.com/miraisolutions/xlconnect
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.1.0     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggimage)
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'
```

```
## The following object is masked from 'package:purrr':
##
##     flatten
```

```
library(httr)
library(rvest)
```

```
##
## Attaching package: 'rvest'

## The following object is masked from 'package:readr':
##
##     guess_encoding
```

```
library(gghighlight)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
`%!in%` <- Negate(`%in%`)
options(java.parameters = "- Xmx1024m")
```

## Import cleaned salary dataset

```
sal = read_csv('all_seasons_salaries.csv')
```

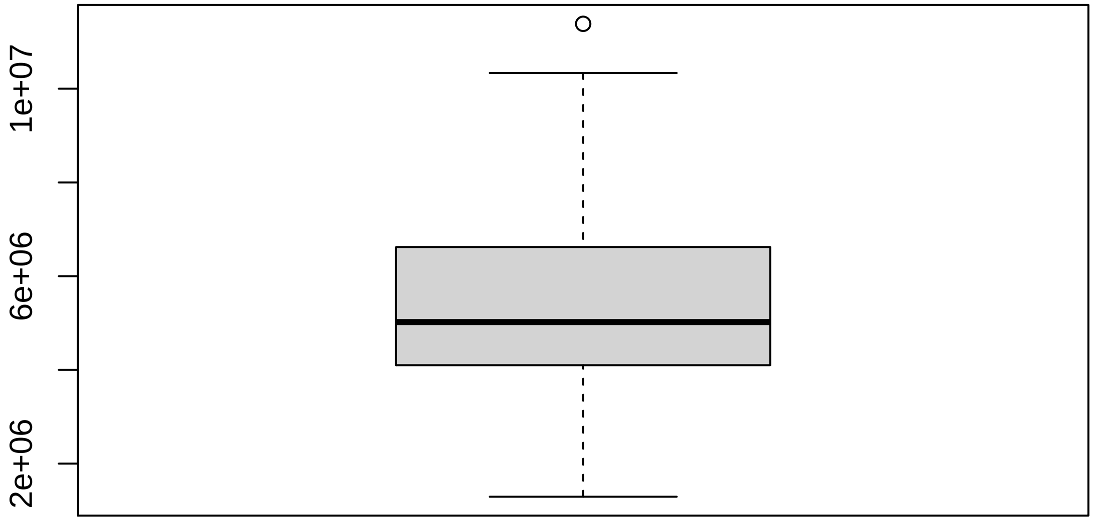
```
##
## -- Column specification --------------------------------------------------------
## cols(
##   player = col_character(),
##   salary = col_double(),
##   TEAM = col_character(),
##   GP = col_double(),
##   year = col_character()
## )
```

## Match Players to teams by season & Add statistics columns

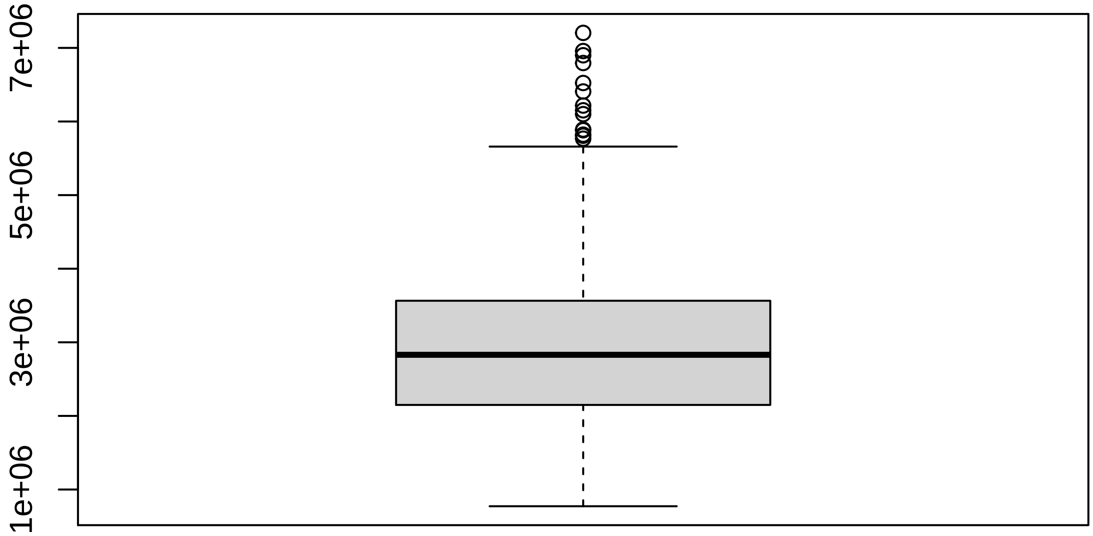
```r
# Amount of rows in data set
seasons = 12
teams = 30
l = seasons*teams

# Copy dataset
copy_sal = sal

# Create empty new dataframe for output
sal_dist = data.frame('Team'=character(l),
                      'Season'=character(l),
                      'Mean_Salary'=numeric(l),
                      'Median-Salary'=numeric(l),
                      'Max'=numeric(l),
                      'Min'=numeric(l),
                      'Standard_deviation'=numeric(l),
                      'Total' = numeric(l),
                      'salaries' = numeric(l))

# For counting players on teams
i = 1

# For loop to transform data to new dataframe
for(z in 1:l){

  # Get team and season
  team = copy_sal$TEAM[1]
  yr = copy_sal$year[1]

  # Extract salaries by team and season
  sals = copy_sal$salary[copy_sal$TEAM == team & copy_sal$year == yr & !is.na(copy_sal$salary)]

  # Place team, season, & salary statistics in data frame
  sal_dist$Team[z] = team
  sal_dist$Season[z] = yr
  sal_dist$Mean_Salary[z] = mean(sals)
  sal_dist$Median.Salary[z] = median(sals)
  sal_dist$Max[z] = max(sals)
  sal_dist$Min[z] = min(sals)
  sal_dist$Standard_deviation[z] = sd(sals)
  sal_dist$Total[z] = sum(sals)
  sal_dist$salaries[z] = list(sals)

  # Remove data from copied data frame
  c_len = length(copy_sal$salary)
  len = length(sals)
  copy_sal = copy_sal[1+len:c_len,]

}
```

## Explore Data using box plots

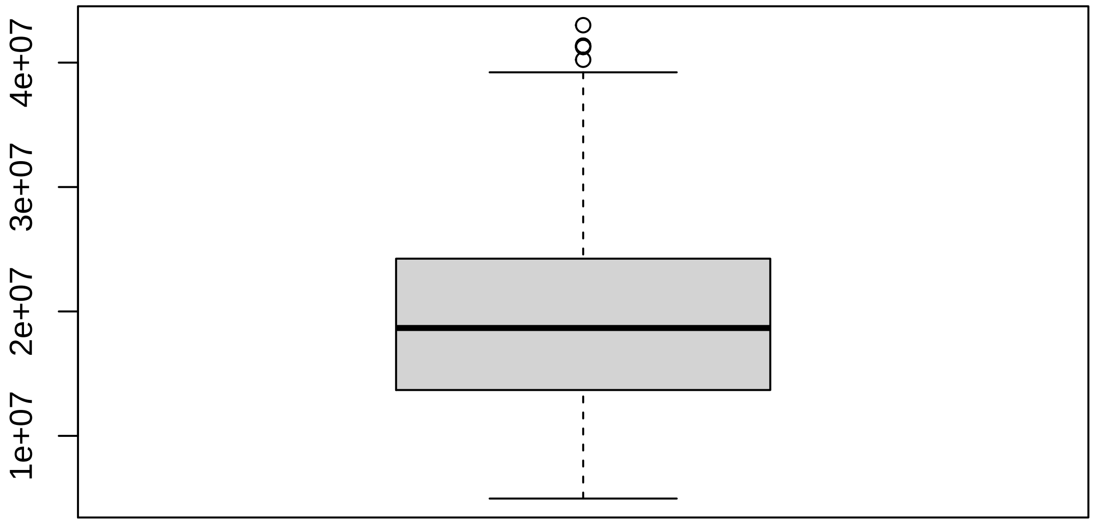
```
boxplot(sal_dist$Mean_Salary)
```



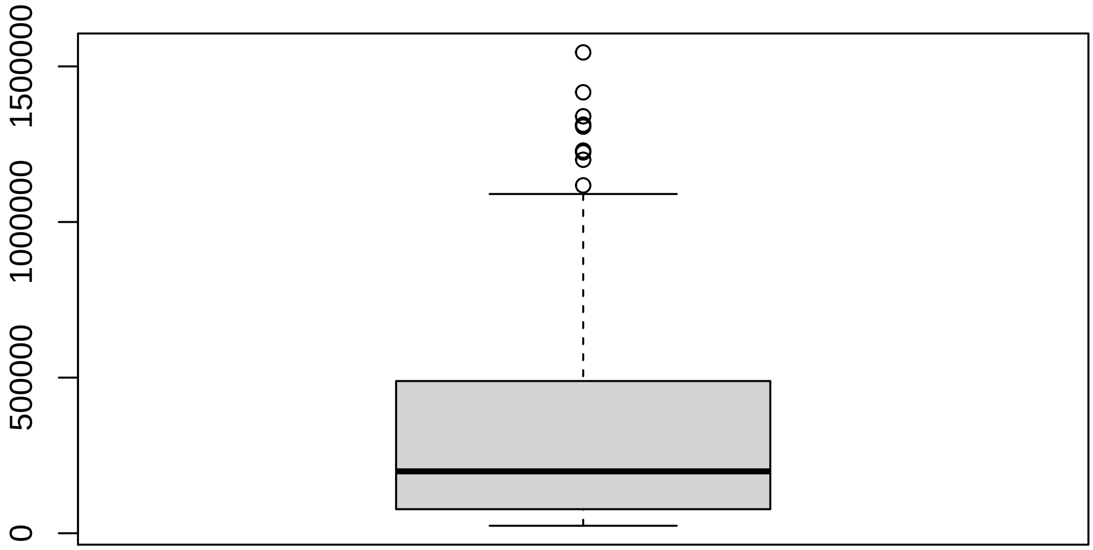
```
boxplot(sal_dist$Median.Salary)
```



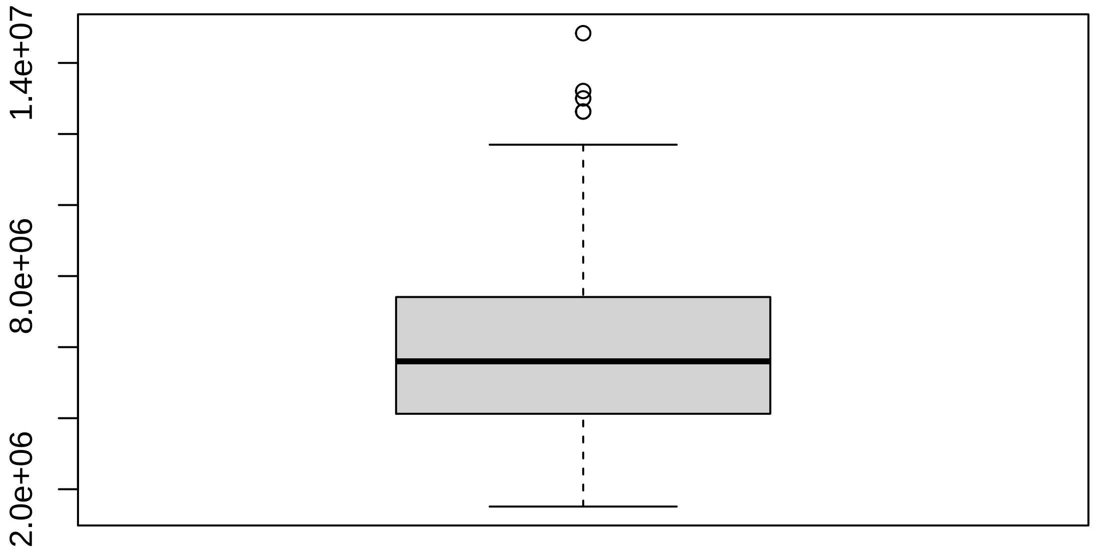
```
boxplot(sal_dist$Max)
```



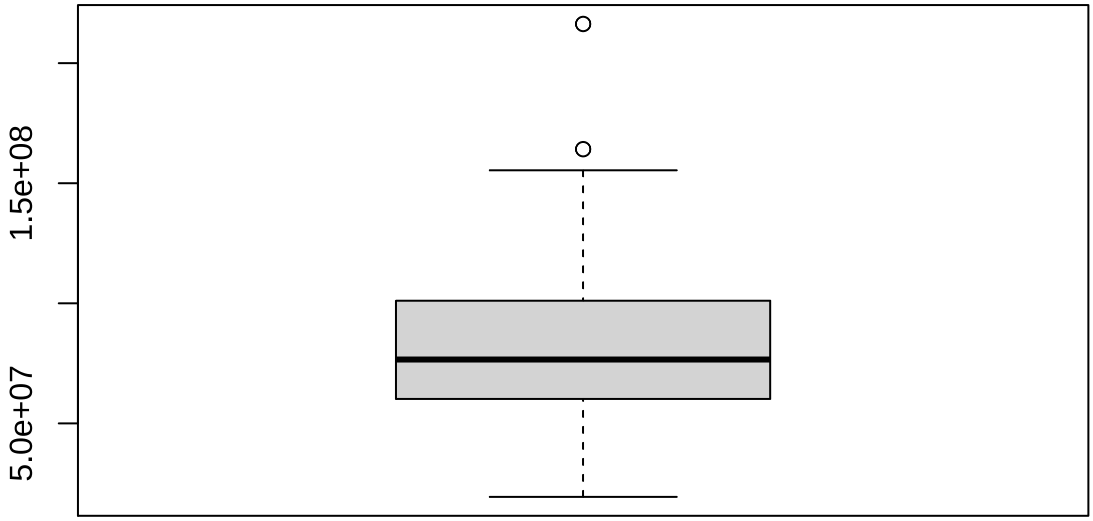
```
boxplot(sal_dist$Min)
```



```
boxplot(sal_dist$Standard_deviation)
```



```
boxplot(sal_dist$Total)
```



## Match abbreviation to city to match dataset with win dataset

```
for(i in 1:length(sal_dist$Team)){
  if(sal_dist$Team[i] == "ATL")
    sal_dist$nickname[i] = "Atlanta"
  else if(sal_dist$Team[i] == "BOS")
    sal_dist$nickname[i] = "Boston"
  else if(sal_dist$Team[i] == "CHA")
    sal_dist$nickname[i] = "Charlotte"
  else if(sal_dist$Team[i] == "CHI")
    sal_dist$nickname[i] = "Chicago"
  else if(sal_dist$Team[i] == "CLE")
    sal_dist$nickname[i] = "Cleveland"
  else if(sal_dist$Team[i] == "DAL")
    sal_dist$nickname[i] = "Dallas"
  else if(sal_dist$Team[i] == "DEN")
    sal_dist$nickname[i] = "Denver"
  else if(sal_dist$Team[i] == "DET")
    sal_dist$nickname[i] = "Detroit"
  else if(sal_dist$Team[i] == "GSW")
    sal_dist$nickname[i] = "Golden State"
  else if(sal_dist$Team[i] == "HOU")
    sal_dist$nickname[i] = "Houston"
  else if(sal_dist$Team[i] == "IND")
    sal_dist$nickname[i] = "Indiana"
  else if(sal_dist$Team[i] == "LAC")
    sal_dist$nickname[i] = "LA Clippers"
  else if(sal_dist$Team[i] == "LAL")
    sal_dist$nickname[i] = "LA Lakers"
  else if(sal_dist$Team[i] == "MEM")
    sal_dist$nickname[i] = "Memphis"
  else if(sal_dist$Team[i] == "MIA")
    sal_dist$nickname[i] = "Miami"
  else if(sal_dist$Team[i] == "MIL")
    sal_dist$nickname[i] = "Milwaukee"
  else if(sal_dist$Team[i] == "MIN")
    sal_dist$nickname[i] = "Minnesota"
  else if(sal_dist$Team[i] == "NJN")
    sal_dist$nickname[i] = "New Jersey"
  else if(sal_dist$Team[i] == "NOH")
    sal_dist$nickname[i] = "New Orleans"
  else if(sal_dist$Team[i] == "NYK")
    sal_dist$nickname[i] = "New York"
  else if(sal_dist$Team[i] == "OKC")
    sal_dist$nickname[i] = "Okla City"
  else if(sal_dist$Team[i] == "ORL")
    sal_dist$nickname[i] = "Orlando"
  else if(sal_dist$Team[i] == "PHI")
    sal_dist$nickname[i] = "Philadelphia"
  else if(sal_dist$Team[i] == "PHX")
    sal_dist$nickname[i] = "Phoenix"
  else if(sal_dist$Team[i] == "POR")
    sal_dist$nickname[i] = "Portland"
  else if(sal_dist$Team[i] == "SAC")
    sal_dist$nickname[i] = "Sacramento"
  else if(sal_dist$Team[i] == "SAS")
    sal_dist$nickname[i] = "San Antonio"
  else if(sal_dist$Team[i] == "TOR")
    sal_dist$nickname[i] = "Toronto"
  else if(sal_dist$Team[i] == "UTA")
    sal_dist$nickname[i] = "Utah"
  else if(sal_dist$Team[i] == "WAS")
    sal_dist$nickname[i] = "Washington"
  else if(sal_dist$Team[i] == "BKN")
    sal_dist$nickname[i] = "Brooklyn"
  else if(sal_dist$Team[i] == "NOP")
    sal_dist$nickname[i] = "New Orleans"

}
```

## Clean up naming & column order of new dataset

```
names(sal_dist)[10] = "City"
sal_dist = sal_dist[,c("Team","City","Season","Mean_Salary","Median.Salary",
                      "Max","Min","Standard_deviation","Total","salaries")]
```

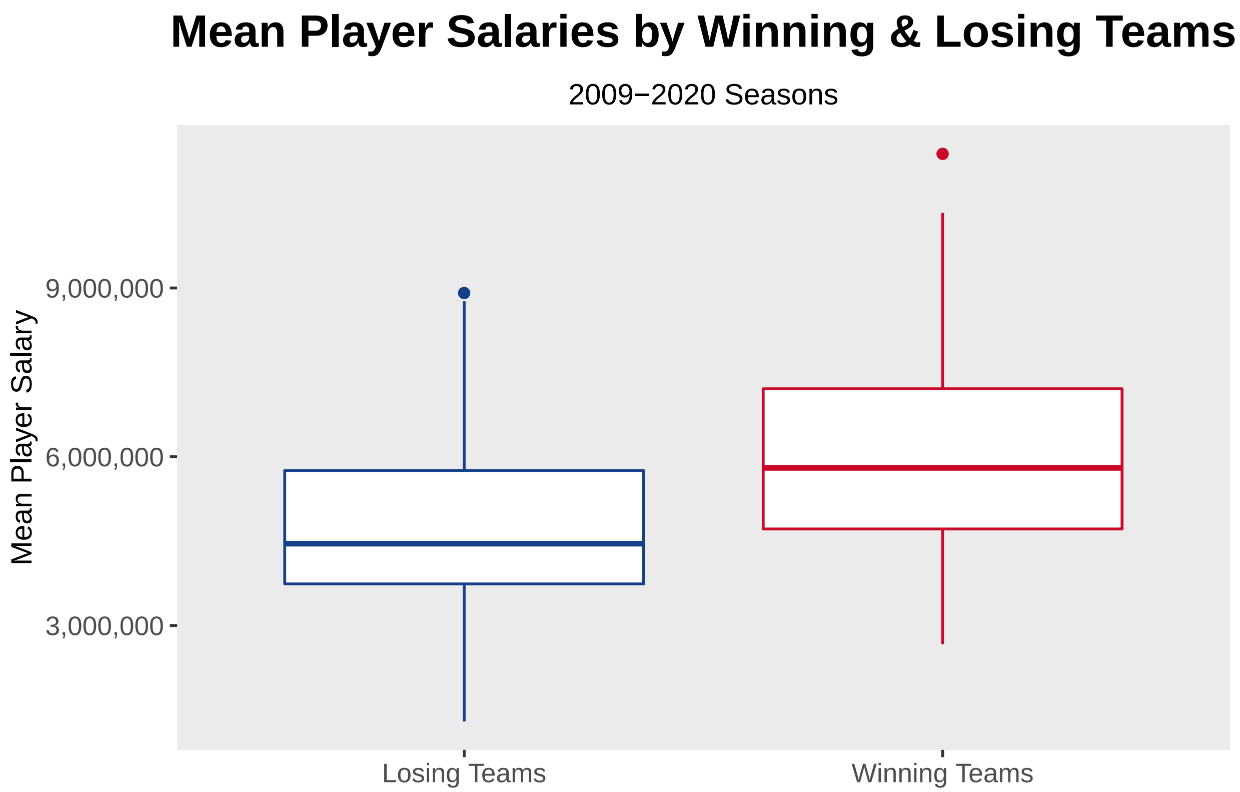
## Combine dataset with Wins dataset

```
wins = readWorksheetFromFile("wins.xlsx",sheet = 1,header = T)
sal_dist = sal_dist %>%
  left_join(wins, by = c('City' = 'Team','Season'='year'))
#sum(is.na(sal_dist$wp))
```

```
# Save in case needed for later usage
#write_csv(sal_dist,'sal_dist.csv')
```

## View Mean differences between winning and losing teams.

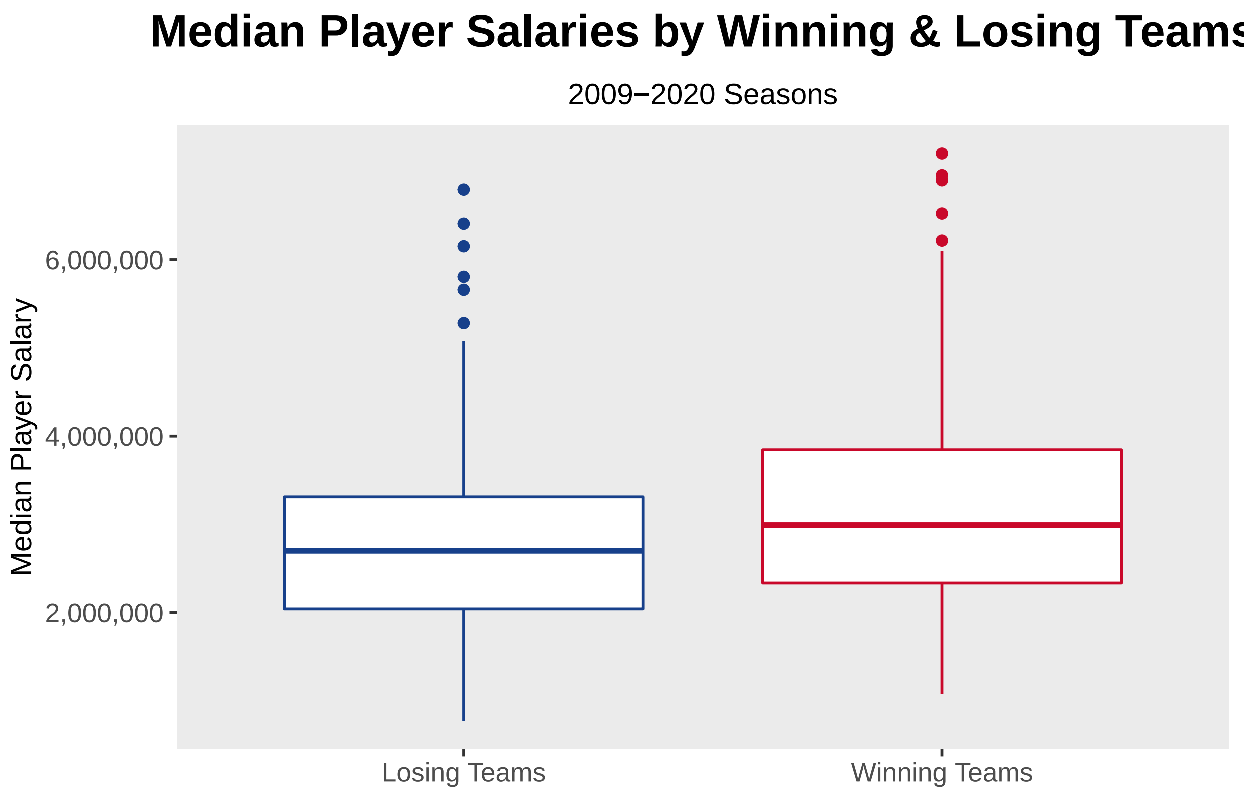
```
# Split winning and losing team labels
sal_dist$WinTeam = "Winning Teams"
sal_dist$WinTeam[sal_dist$wp < .5] = "Losing Teams"

# Save as variables
w = sal_dist$WinTeam[sal_dist$WinTeam == "Winning Teams"]
l = sal_dist$WinTeam[sal_dist$WinTeam == "Losing Teams"]

# NBA colors
logocolors = c('#17408B',"#C9082A")

# Boxplot
ggplot(data = sal_dist)+
  geom_boxplot(aes(x = WinTeam,y = Mean_Salary),color = logocolors)+
  labs(x = "",
       y = "Mean Player Salary",
       title = "Mean Player Salaries by Winning & Losing Teams",
       subtitle = "2009-2020 Seasons"
         ) +
  theme(plot.title = element_text(size=17, face="bold",
    margin = margin(10, 0, 10, 0),hjust =.5),
    plot.subtitle = element_text(hjust =.5),
    axis.text=element_text(size=10),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank())+
  scale_y_continuous(labels = scales::comma)
```



```
ggsave('mean_box.png',width = 12, height = 9, dpi = 'retina')
```

### View Median differences between winning and losing teams.

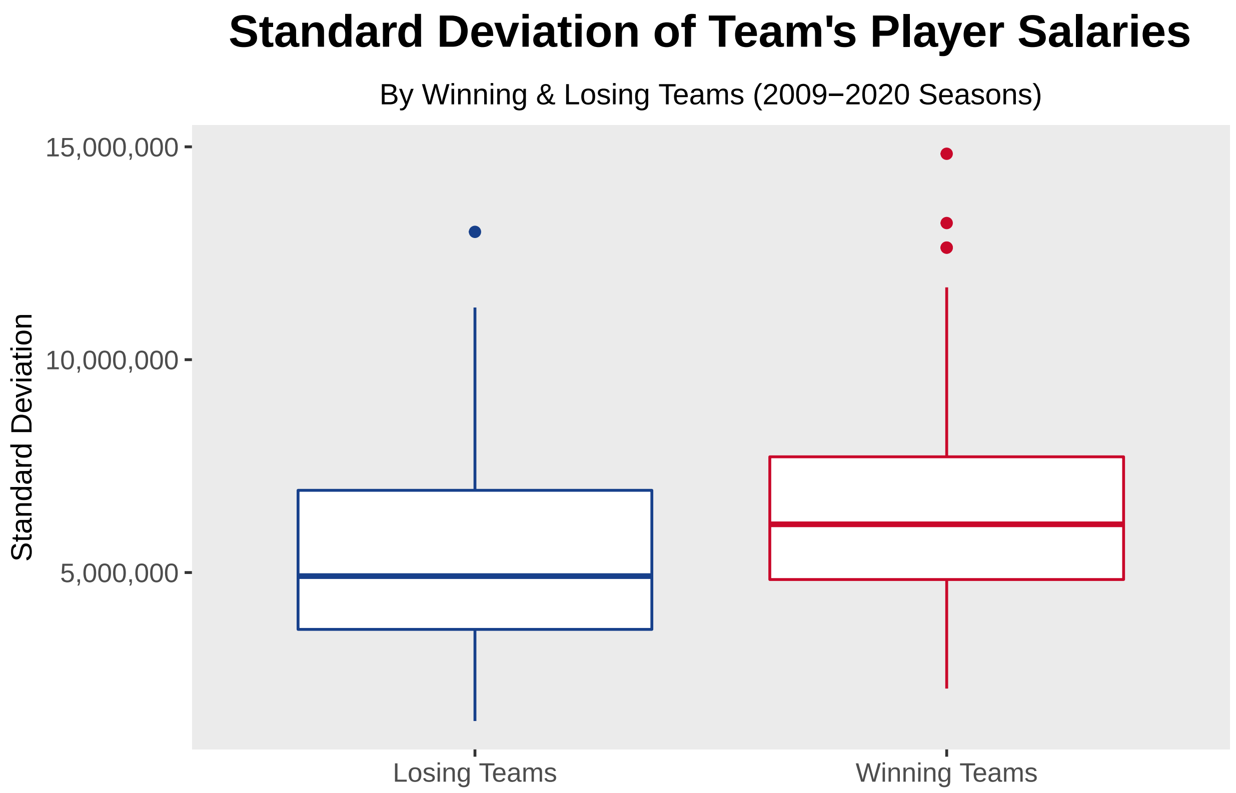
```
ggplot(data = sal_dist)+
  geom_boxplot(aes(x = WinTeam,y = Median.Salary),color = logocolors)+
  labs(x = "",
       y = "Median Player Salary",
       title = "Median Player Salaries by Winning & Losing Teams",
       subtitle = "2009-2020 Seasons"
         ) +
  theme(plot.title = element_text(size=17, face="bold",
    margin = margin(10, 0, 10, 0),hjust =.5),
    plot.subtitle = element_text(hjust =.5),
    axis.text=element_text(size=10),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank())+
  scale_y_continuous(labels = scales::comma)
```



```
ggsave('median_box.png',width = 12, height = 9, dpi = 'retina')
```

**View Standard deviation differences between winning and losing teams.**

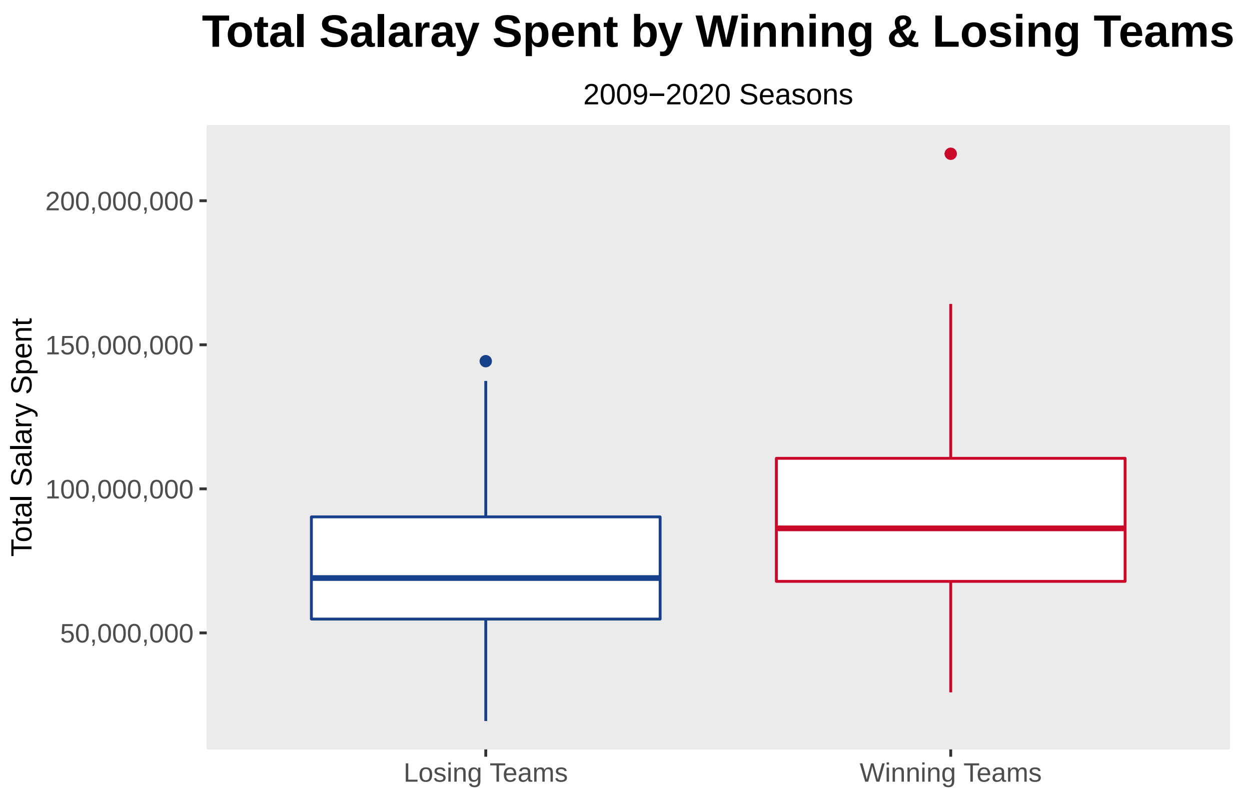
```
ggplot(data = sal_dist)+
  geom_boxplot(aes(x = WinTeam,y = Standard_deviation),color = logocolors)+
  labs(x = "",
       y = "Standard Deviation",
       title = "Standard Deviation of Team's Player Salaries",
       subtitle = "By Winning & Losing Teams (2009-2020 Seasons)"
         ) +
  theme(plot.title = element_text(size=17, face="bold",
    margin = margin(10, 0, 10, 0),hjust =.5),
    plot.subtitle = element_text(hjust =.5),
    axis.text=element_text(size=10),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank())+
  scale_y_continuous(labels = scales::comma)
```



```
ggsave('sd_box.png',width = 12, height = 9, dpi = 'retina')
```

### View Total Spend differences between winning and losing teams.

```
ggplot(data = sal_dist)+
  geom_boxplot(aes(x = WinTeam,y = Total),color = logocolors)+
  labs(x = "",
       y = "Total Salary Spent",
       title = "Total Salaray Spent by Winning & Losing Teams",
       subtitle = "2009-2020 Seasons"
         ) +
  theme(plot.title = element_text(size=17, face="bold",
    margin = margin(10, 0, 10, 0),hjust =.5),
    plot.subtitle = element_text(hjust =.5),
    axis.text=element_text(size=10),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank())+
  scale_y_continuous(labels = scales::comma)
```



```
ggsave('total_box.png',width = 12, height = 9, dpi = 'retina')
```

### Run bootstrapping on mean differences to find 95% confidence intervals

```
set.seed(1234)

# 100000 bootstrapping samples
N = 100000

x = numeric(N)

## Run bootstrapping samples to find difference between winning and losing teams
for (i in 1:N){
  win <- mean(sample(sal_dist$Mean_Salary[sal_dist$WinTeam == "Winning Teams"], length(w), replace = T)
  loss <- mean(sample(sal_dist$Mean_Salary[sal_dist$WinTeam == 'Losing Teams'],length(l), replace = T))
  x[i] = win-loss
}

# Develop Confidence Intervals
std = sd(x)
LL=mean(x)-1.96*std
UL=mean(x)+1.96*std
ci = c(LL,UL)
cat('95% Confidence Interval: [',ci[1],',',ci[2],']')
```

```
## 95% Confidence Interval: [ 791790.9 , 1441503 ]
```

### Run bootstrapping on median differences to find 95% confidence intervals

```
set.seed(1234)

# 100000 bootstrapping samples
N = 100000

x = numeric(N)

## Run bootstrapping samples to find difference between winning and losing teams
for (i in 1:N){
  win <- mean(sample(sal_dist$Median.Salary[sal_dist$WinTeam == "Winning Teams"], length(w), replace =
  loss <- mean(sample(sal_dist$Median.Salary[sal_dist$WinTeam == 'Losing Teams'],length(l), replace = T
  x[i] = win-loss
}

# Develop Confidence Intervals
std = sd(x)
LL=mean(x)-1.96*std
UL=mean(x)+1.96*std
ci = c(LL,UL)
cat('95% Confidence Interval: [',ci[1],',',ci[2],']')
```

```
## 95% Confidence Interval: [ 170840.4 , 655284.5 ]
```

```
set.seed(1234)

# 100000 bootstrapping samples
N = 100000

x = numeric(N)

## Run bootstrapping samples to find difference between winning and losing teams
for (i in 1:N){
  win <- mean(sample(sal_dist$Standard_deviation[sal_dist$WinTeam == 'Winning Teams'], length(w), repla
  loss <- mean(sample(sal_dist$Standard_deviation[sal_dist$WinTeam == 'Losing Teams'],length(l), replac
  x[i] = win-loss
}

# Develop Confidence Intervals
std = sd(x)
LL=mean(x)-1.96*std
UL=mean(x)+1.96*std
ci = c(LL,UL)
cat('95% Confidence Interval: [',ci[1],',',ci[2],']')
```

```
## 95% Confidence Interval: [ 716973.9 , 1690814 ]
```

```
set.seed(1234)

# 100000 bootstrapping samples
```

```
N = 100000

x = numeric(N)

## Run bootstrapping samples to find difference between winning and losing teams
for (i in 1:N){
  win <- mean(sample(sal_dist$Total[sal_dist$WinTeam == 'Winning Teams'], length(w), replace = T))
  loss <- mean(sample(sal_dist$Total[sal_dist$WinTeam == 'Losing Teams'],length(l), replace = T))
  x[i] = win-loss
}

# Develop Confidence Intervals
std = sd(x)
LL=mean(x)-1.96*std
UL=mean(x)+1.96*std
ci = c(LL,UL)
cat('95% Confidence Interval: [',ci[1],',',ci[2],']')
```

```
## 95% Confidence Interval: [ 10652061 , 21822391 ]
```

### Melt Winning dataframe in order to plot each individual team as its own line

```
# Create new data frame
df_w = data.frame(x=1:21)

# Copy salaries as new variable
sals = sal_dist$salaries[sal_dist$WinTeam == 'Winning Teams']

# Get salaries from each team and make it a new column in df
for (i in 1:length(w)){

  a = replicate(21,0)
  b = sort(sals[[i]],decreasing =T)
  a[1:length(b)] = b

  nam = paste("col", i, sep = "")
  df_w = cbind(df_w,a)
  colnames(df_w)[ncol(df_w)] = nam


}

# melt dataframe to make x's repeat & y's be salaries of each team
df_w = melt(df_w, id = "x")
```

### Melt Losing dataframe in order to plot each individual team as its own line

```
# Create new data frame
df_l = data.frame(x=1:21)
```

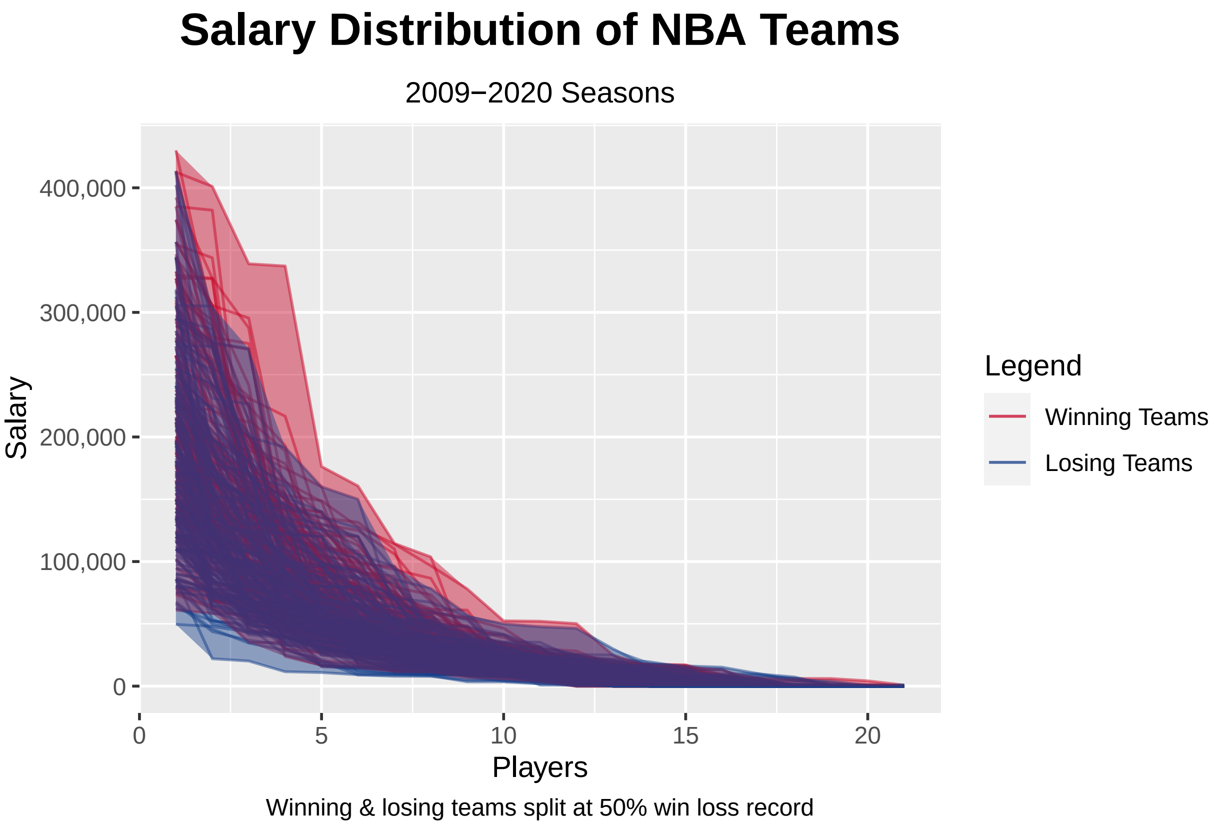
```r
# Copy salaries as new variable
sals = sal_dist$salaries[sal_dist$WinTeam == 'Losing Teams']

# Get salaries from each team and make it a new column in df
for (i in 1:length(l)){

  a = replicate(21,0)
  b = sort(sals[[i]],decreasing = T)
  a[1:length(b)] = b
  nam = paste("col", i, sep = "")

  df_l = cbind(df_l,a)
  colnames(df_l)[ncol(df_l)] = nam


}

# melt dataframe to make x's repeat & y's be salaries of each team
df_l = melt(df_l, id = "x")
```

### Find mins and maxes for plotting ribbon

```r
win_max = c()
win_min = c()
lose_max = c()
lose_min = c()
for (i in 1:21){
  d = df_w$value[df_w$x == i]
  e = df_l$value[df_l$x == i]
  win_max = append(win_max,max(d))
  win_min = append(win_min,min(d))
  lose_max = append(lose_max,max(e))
  lose_min = append(lose_min,min(e))
}
```

### Plot each teams salary distribution from dataset. Separating winning and losing teams

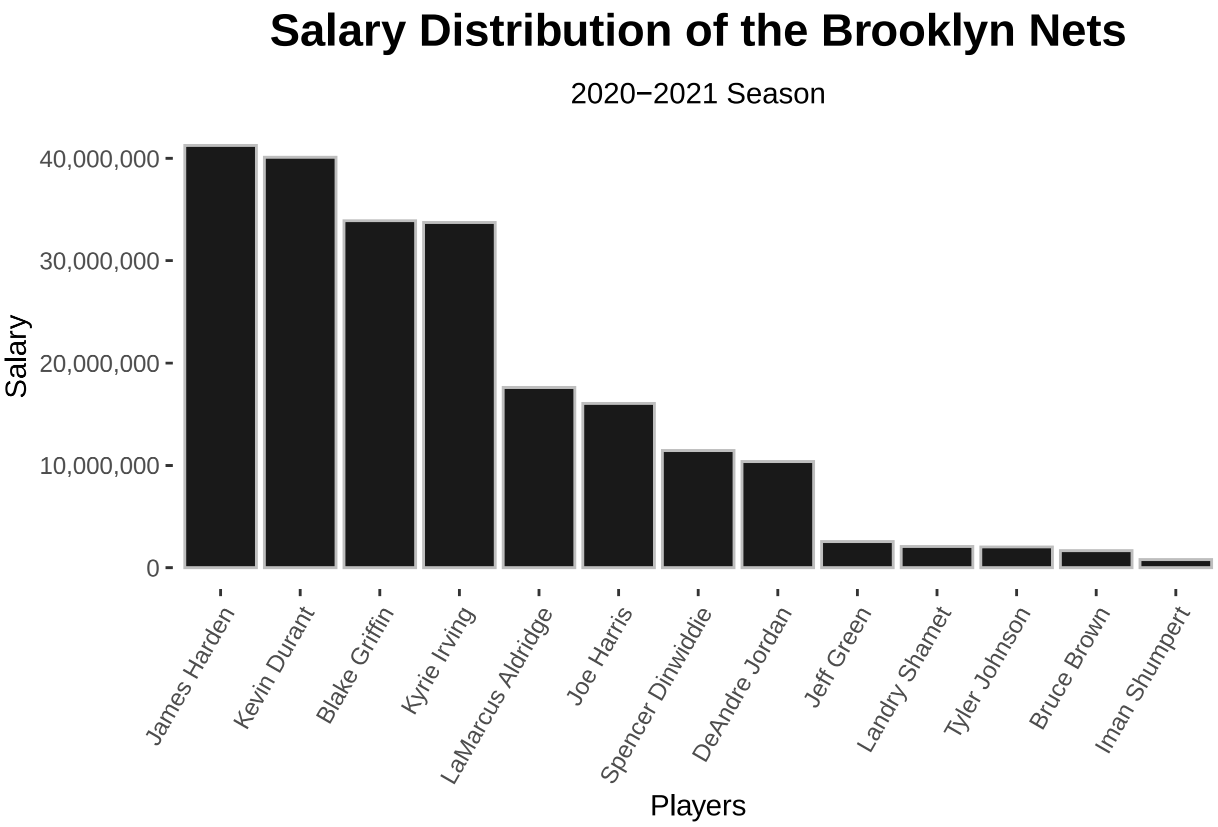
```r
colors <- c("Winning Teams" = logocolors[2], "Losing Teams" = logocolors[1])

plt <- ggplot() +
  geom_line(data = df_w,aes(group = variable, x = x,y = value/100,color = "Winning Teams"),alpha = .5)+
  geom_line(data = df_l,aes(group = variable, x = x,y = value/100,color = "Losing Teams"),alpha = .5)+
  geom_ribbon(aes(x = 1:21,ymin = win_min/100, ymax = win_max/100), fill = logocolors[2],alpha = .45 )+
  geom_ribbon(aes(x = 1:21,ymin = lose_min/100, ymax = lose_max/100), fill = logocolors[1],alpha = .45)
  labs(x = "Players",
       y = "Salary",
       color = "Legend",
       title = "Salary Distribution of NBA Teams",
        subtitle = "2009-2020 Seasons",
        caption = "Winning & losing teams split at 50% win loss record"
          ) +
  scale_color_manual(values = colors)+
  theme(plot.title = element_text(size=17, face="bold",
    margin = margin(10, 0, 10, 0),hjust =.5),
    plot.subtitle = element_text(hjust =.5),
    plot.caption = element_text(hjust=0.5))+
  scale_y_continuous(labels = scales::comma)
plt
```



```
#ggsave('Team Salary Distributions.png',plot = plt)
```

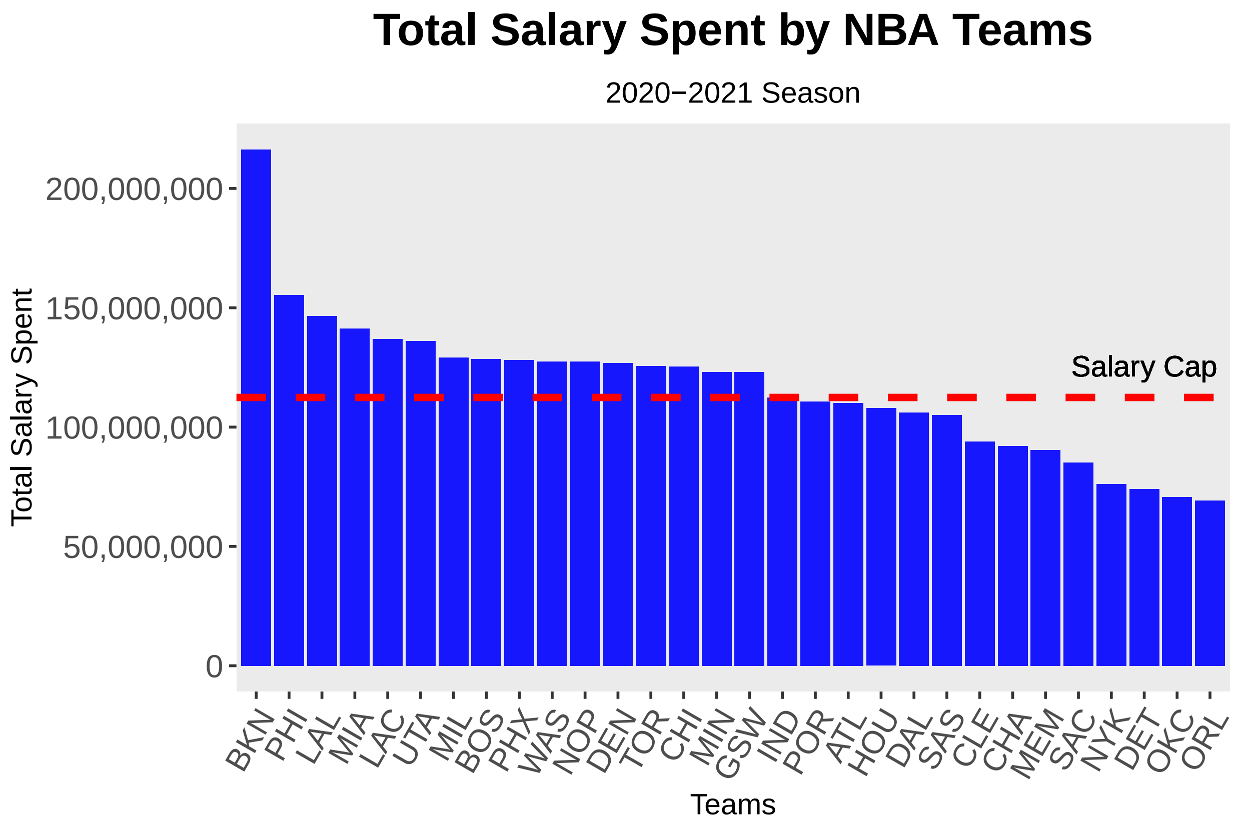
### EDA for presentation

```
# Salaries Example
ex = sal[sal$TEAM == 'BKN' & sal$year == '20_21',]
ex = ex[1:13,]

plt <- ggplot(data = ex,aes(x = fct_reorder(player,-salary), y = salary))+
  geom_bar(fill = 'black',color = 'grey', stat = "identity", alpha = 0.9)+
  labs(x = "Players",
        y = "Salary",
        title = "Salary Distribution of the Brooklyn Nets",
        subtitle = "2020-2021 Season"
          ) +
  theme(plot.title = element_text(size=17, face="bold",
    margin = margin(10, 0, 10, 0),hjust =.5),
    plot.subtitle = element_text(hjust =.5),
    axis.text.x = element_text(angle = 60, vjust = 0.99, hjust=1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.background = element_blank())+
  scale_y_continuous(labels = scales::comma)

plt
```



```
#ggsave('Nets_20_21.png',plot = plt,width = 12, height = 9, dpi = 'retina')
```

```
# Salary Cap Example
cap = sal_dist[sal_dist$Season == '20_21',]
plt <- ggplot(data = cap,aes(x = fct_reorder(Team,-Total), y = Total))+
  geom_bar(fill = 'blue', stat = "identity", alpha = 0.9)+
  geom_hline(yintercept=112414000, linetype="dashed", color = "red",size=1.3)+
  geom_text(aes(28, 112414000, label = 'Salary Cap', vjust = - 1))+
  labs(x = "Teams",
       y = "Total Salary Spent",
       title = "Total Salary Spent by NBA Teams",
       subtitle = "2020-2021 Season"
         ) +
  theme(plot.title = element_text(size=17, face="bold",
    margin = margin(10, 0, 10, 0),hjust =.5),
    plot.subtitle = element_text(hjust =.5),
    axis.text.x = element_text(angle = 60, vjust = 0.99, hjust=1),
    axis.text=element_text(size=12),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank())+
  scale_y_continuous(labels = scales::comma)

plt
```



```
#ggsave('sal_cap_20_21.png',plot = plt,width = 12, height = 9, dpi = 'retina')
```

```
# Data table
dt = cap[1:7,c(1,2,11,10)]
colnames(dt)[3] = 'Win Percentage'
colnames(dt)[4] = 'Salaries'

dt %>%
  kbl() %>%
  kable_material(c("striped", "hover"))
```

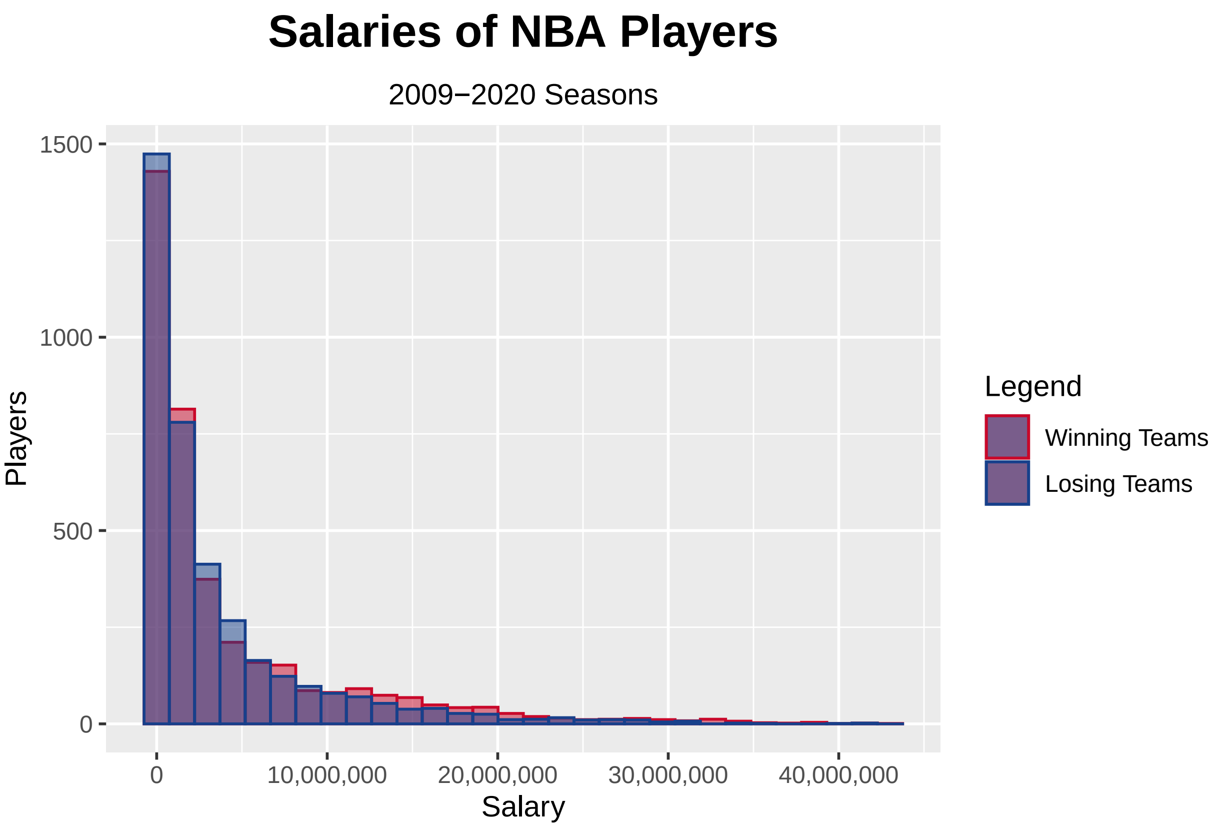
```
# Histogram  of player salaries on winning and losing teams
colors <- c("Winning Teams" = logocolors[2], "Losing Teams" = logocolors[1])

plt <- ggplot() +
```

|  | Team | City | Win Percentage | Salaries |
|---|---|---|---|---|
| 331 | ATL | Atlanta | 0.567 | 19500000, 18000000, 18000000, 12178571, 7422000, 6571800, 5813640, 476700 |
| 332 | BKN | Brooklyn | 0.655 | 41254920, 40108950, 33900241, 33722850, 17628340, 16071429, 11454048, 103 |
| 333 | BOS | Boston | 0.487 | 27500000, 25035118, 17150000, 12946428, 9897120, 9258000, 6930729, 363120 |
| 334 | CHA | Charlotte | 0.452 | 28500000, 18900000, 15415730, 7839960, 5345687, 3934320, 3500000, 2250000 |
| 335 | CHI | Chicago | 0.431 | 26000000, 19500000, 13545000, 10000000, 9720900, 7529020, 7068360, 673150 |
| 336 | CLE | Cleveland | 0.306 | 31258256, 12250000, 8840580, 6720720, 6400920, 4991880, 3909902, 2432353, |
| 337 | DAL | Dallas | 0.570 | 29467800, 13013700, 11080125, 10865952, 8475000, 8049360, 4100000, 400000 |

```
  geom_histogram(data = df_w,aes(x = value,color = "Winning Teams"), fill = logocolors[2], bins = 30,alp
  geom_histogram(data = df_l,aes(x = value,color = "Losing Teams"), fill = logocolors[1], bins = 30, alp
  labs(x = "Salary",
       y = "Players",
       color = "Legend",
       title = "Salaries of NBA Players",
       subtitle = "2009-2020 Seasons"
         ) +
  scale_color_manual(values = colors)+
  theme(plot.title = element_text(size=17, face="bold",
    margin = margin(10, 0, 10, 0),hjust =.5),
    plot.subtitle = element_text(hjust =.5))+
  scale_x_continuous(labels = scales::comma)+
  scale_color_manual(values = colors)
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```

```
plt
```



```
#ggsave('player_sals.png',plot = plt)
```

# TitleEffect_Analysis

Jack Piccione

12/6/2021

```
#### Loading necessary libraries


###########################################################
#### Accessing the data and saving csv file as a dataframe
title_df = read.csv("TitleEffect.csv")
champs_df = read.csv("Champs.csv")
year_after_df = read.csv("yearAfter.csv")

#### Verification of correct data upload
head(title_df)
```

```
##   Season                Team  W  L  W.L.   Finish  SRS Pace Rel.Pace ORtg
## 1   1951   Rochester Royals* 41 27 0.603 2nd of 5 2.54 92.3      6.8 89.3
## 2   1951 Minneapolis Lakers* 44 24 0.647 1st of 5 4.79 94.8      9.3 86.4
## 3   1952 Minneapolis Lakers* 40 26 0.606 2nd of 5 5.28 97.2     11.4 85.3
## 4   1952   Rochester Royals* 41 25 0.621 1st of 5 2.92 92.7      6.9 91.6
## 5   1953 Minneapolis Lakers* 48 22 0.686 1st of 5 5.54 93.7     13.3 90.1
## 6   1954 Minneapolis Lakers* 46 26 0.639 1st of 4 2.71 93.5      9.9 86.8
##   Rel.ORtg DRtg Rel.DRtg            Playoffs
## 1      4.2 86.2      1.1                   1
## 2      1.3 80.7     -4.4 Lost W. Div. Finals
## 3     -1.6 79.3     -7.6                   1
## 4      4.7 88.1      1.2 Lost W. Div. Finals
## 5      2.1 83.6     -4.1                   1
## 6     -0.7 83.5     -4.0                   1
```

```
#str(title_df)
head(champs_df)
```

```
##   Season                   Team  W  L  W.L.   Playoffs
## 1   1947 Philadelphia Warriors* 35 25 0.583 Won Finals
## 2   1949    Minneapolis Lakers* 44 16 0.733 Won Finals
## 3   1950    Minneapolis Lakers* 51 17 0.750 Won Finals
## 4   1951      Rochester Royals* 41 27 0.603 Won Finals
## 5   1952    Minneapolis Lakers* 40 26 0.606 Won Finals
## 6   1953    Minneapolis Lakers* 48 22 0.686 Won Finals
```

```
#str(champs_df)
head(year_after_df)
```

```
##   Season                   Team  W  L  W.L.           Playoffs
## 1   1948 Philadelphia Warriors* 27 21 0.563        Lost Finals
## 2   1950    Minneapolis Lakers* 51 17 0.750         Won Finals
## 3   1951    Minneapolis Lakers* 44 24 0.647 Lost W. Div. Finals
## 4   1952      Rochester Royals* 41 25 0.621 Lost W. Div. Finals
## 5   1953    Minneapolis Lakers* 48 22 0.686         Won Finals
## 6   1954    Minneapolis Lakers* 46 26 0.639         Won Finals
```

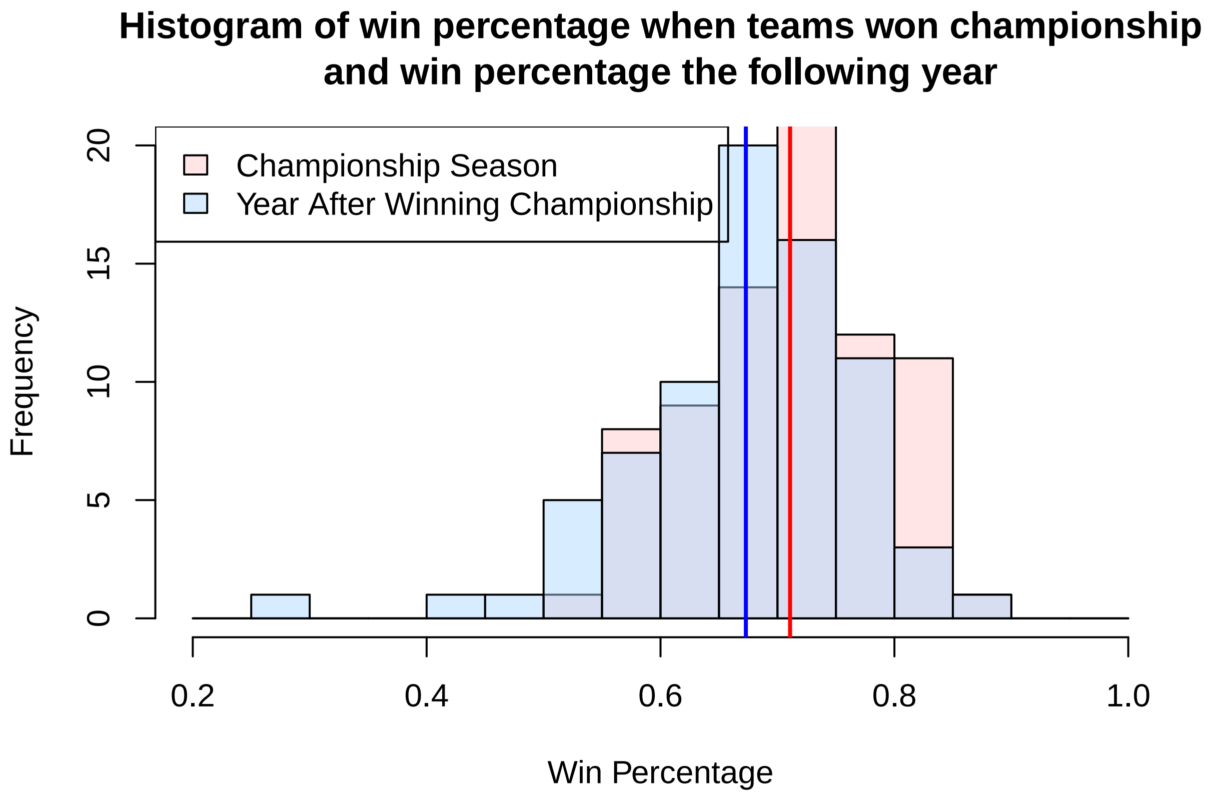
```
#str(year_after_df)

############### Extra Data Cleaning: #############
#### Removing the * at the end of each team name
## title_df
for (i in 1:nrow(title_df)) {
  team_name = gsub(".$", "", title_df$Team[i])
  title_df$Team[i] = team_name
}
## champs_df
for (i in 1:nrow(champs_df)) {
  team_name = gsub(".$", "", champs_df$Team[i])
  champs_df$Team[i] = team_name
}
## year_after_df
for (i in 1:nrow(year_after_df)) {
  team_name = gsub(".$", "", year_after_df$Team[i])
  year_after_df$Team[i] = team_name
  ## Changing value of winning finals from 1 to Won Finals
  if (year_after_df$Playoffs[i] == "1") {
  year_after_df$Playoffs[i] = "Won Finals"
  }
}

## Removing E. and W. from Playoffs column in year_after_df
# This will help with later visualization
year_after_df$Playoffs = gsub("E. ", "", as.character(year_after_df$Playoffs))
year_after_df$Playoffs = gsub("W. ", "", as.character(year_after_df$Playoffs))
## Changing Div. to Conf. in Playoffs column in year_after_df
# This will help with later visualization
year_after_df$Playoffs = gsub("Div.", "Conf.", as.character(year_after_df$Playoffs))



#################################################
#### EDA
## Setting color variables
lightred = rgb(255, 200, 200, max = 255, alpha = 120, names = "lightred")
lightblue = rgb(173, 216, 255, max = 255, alpha = 120, names = "lightblue")
## Setting break points
break_points = pretty(2:10, n = 16) / 10

#### Distribution of win percentage for teams that won the championships and their win percentage the f
## Histogram for championship season
hist(champs_df$W.L., col = lightred, breaks = break_points, ylim = c(0,20),
     main = "Histogram of win percentage when teams won championship\nand win percentage the following
     xlab = "Win Percentage")
## Histogram for season after champs
hist(year_after_df$W.L.,
     col = lightblue,
     breaks = break_points,
     ylim = c(0,20),
     add = TRUE)
## Mean line for championship season
abline(v = mean(champs_df$W.L.), col = "red", lwd = 2, add = TRUE)
```

```
## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "add" is
## not a graphical parameter
```

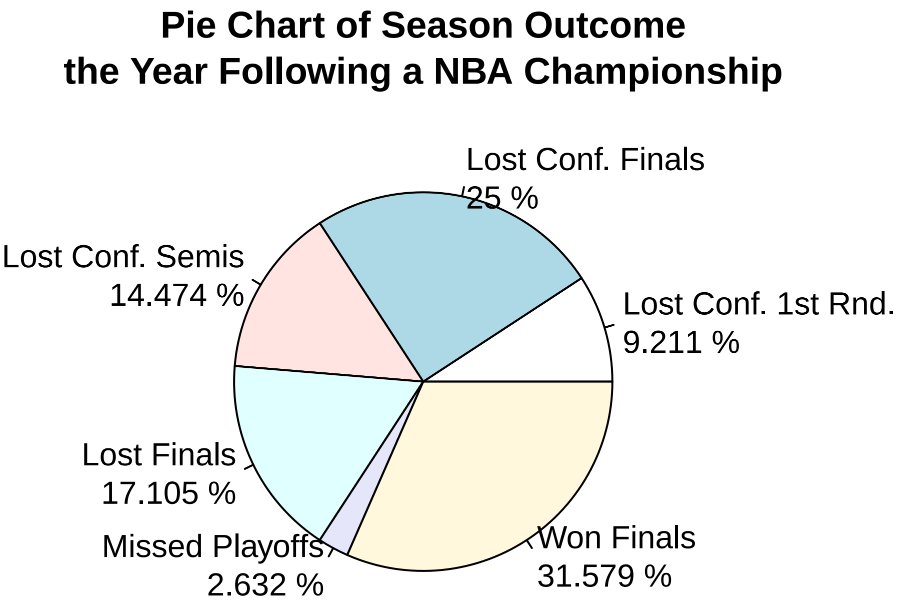
```
## Mean line for season after champs
abline(v = mean(year_after_df$W.L.), col = "blue", lwd = 2, add = TRUE)
```

```
## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "add" is
## not a graphical parameter
```

```
## Adding legend
legend("topleft", legend = c("Championship Season", "Year After Winning Championship"), fill = c(lightr
```



```
#### Creating pie chart
season_results = table(year_after_df$Playoffs)
labels = paste(names(season_results), "\n", paste(round((season_results/nrow(year_after_df)) * 100, dig
pie(season_results, labels = labels, main = "Pie Chart of Season Outcome\nthe Year Following a NBA Cham
```



```
################################################
#### Creating summary statistics to help tell story
## Proportion of teams that missed the playoffs after winning the year prior
missed_playoffs = round(sum(year_after_df$Playoffs == "Missed Playoffs") / nrow(year_after_df), digits =

## Proportion of teams that lost in the 1st round of the conference bracket after winning the year prio
lost_conf_first_round = round(sum(year_after_df$Playoffs == "Lost Conf. 1st Rnd.") /
                          nrow(year_after_df), digits = 4)

## Proportion of teams that lost in the div/conf semi finals after winning the year prior
lost_conf_semis = round(sum(year_after_df$Playoffs == "Lost Conf. Semis") /
                     nrow(year_after_df), digits = 4)

## Proportion of teams that lost in the div/conf finals after winning the year prior
lost_conf_finals = round(sum(year_after_df$Playoffs == "Lost Conf. Finals") /
                      nrow(year_after_df), digits = 4)

## Proportion of teams that lost in the finals after winning the year prior
lost_finals = round(sum(year_after_df$Playoffs == "Lost Finals") / nrow(year_after_df), digits = 4)

## Proportion of teams that won the championships after winning the year prior
won_finals = round(sum(year_after_df$Playoffs == "Won Finals") / nrow(year_after_df), digits = 4)

## Verification that all options are accounted for (should equal 1)
missed_playoffs + lost_conf_first_round + lost_conf_semis + lost_conf_finals + lost_finals + won_finals
```

```
## [1] 1
```

```
## Display in consol
paste("Proportion of teams that missed the playoffs after winning the year prior", missed_playoffs)
```

```
## [1] "Proportion of teams that missed the playoffs after winning the year prior 0.0263"
```

```r
paste("Proportion of teams that lost in the 1st round of the conference bracket after winning the year p
```

```
## [1] "Proportion of teams that lost in the 1st round of the conference bracket after winning the year
```

```r
paste("Proportion of teams that lost in the div/conf semi finals after winning the year prior", lost_co
```

```
## [1] "Proportion of teams that lost in the div/conf semi finals after winning the year prior 0.1447"
```

```r
paste('Proportion of teams that lost in the div/conf finals after winning the year prior', lost_conf_fi
```

```
## [1] "Proportion of teams that lost in the div/conf finals after winning the year prior 0.25"
```

```r
paste("Proportion of teams that lost in the finals after winning the year prior", lost_finals)
```

```
## [1] "Proportion of teams that lost in the finals after winning the year prior 0.1711"
```

```r
paste("Proportion of teams that won the championships after winning the year prior", won_finals)
```

```
## [1] "Proportion of teams that won the championships after winning the year prior 0.3158"
```

```r
###########################################################################
#### Hypothesis Testing
### Null Hypothesis: For teams that won an NBA championship,
# The mean winning percentage is equal between the year that the team won and the year after the team w
### Alternative Hypothesis: For teams that won an NBA championship,
# The mean winning percentage is not equal between the year that the team won and the year after the te

#### Observing actual means for reference
mean_win_percent_champions = mean(champs_df$W.L.)
mean_win_percent_year_after = mean(year_after_df$W.L.)
mean_win_percent_champions ## Display
```

```
## [1] 0.7107792
```

```r
mean_win_percent_year_after ## Display
```

```
## [1] 0.6729737
```

```r
## Conducting T-Test
ttest = t.test(champs_df$W.L., year_after_df$W.L., alternative = "two.sided")
ttest # Display
```

```
##
##  Welch Two Sample t-test
##
## data:  champs_df$W.L. and year_after_df$W.L.
## t = 2.6385, df = 143.55, p-value = 0.009247
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.009483367 0.066127707
## sample estimates:
## mean of x mean of y
## 0.7107792 0.6729737
```

# NBAttendance

Jack Piccione

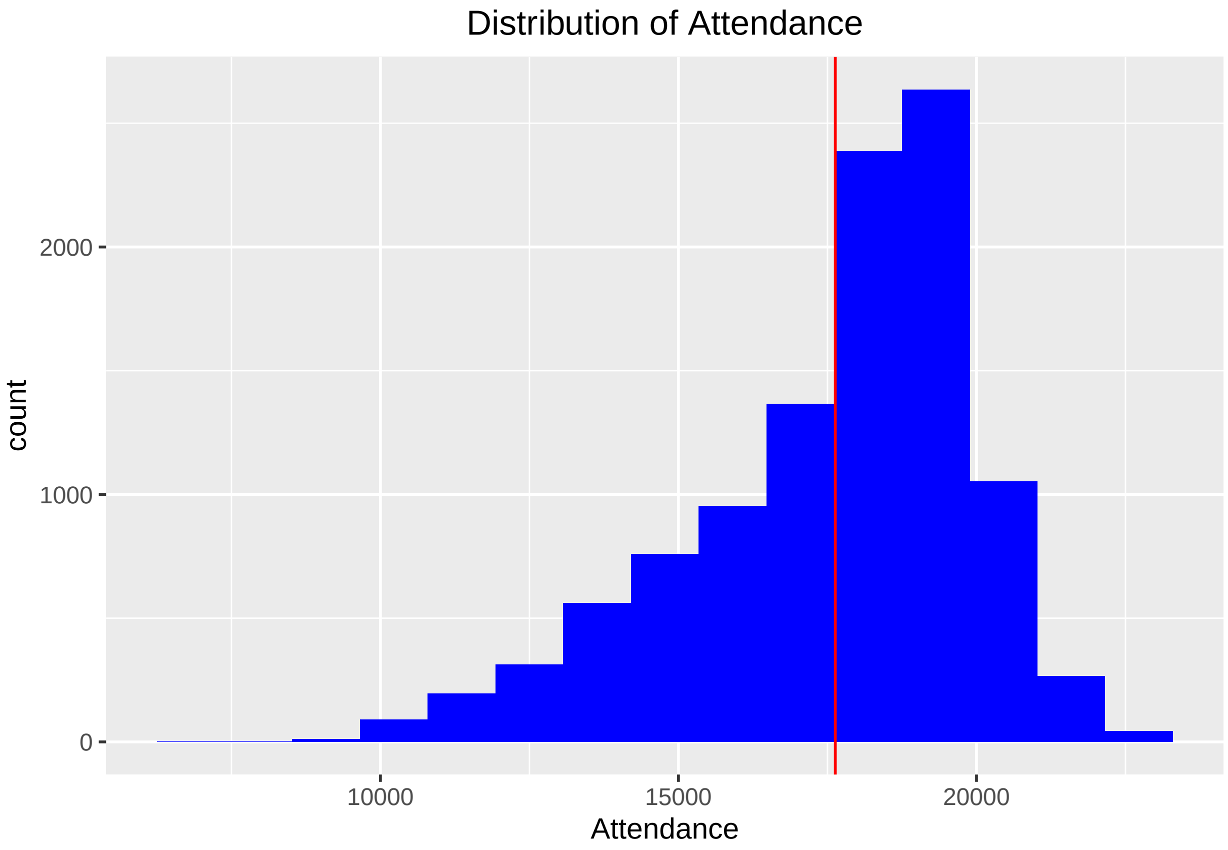
11/11/2021

```
library(ggplot2)
library("ggpubr")
```

```
df<-read.csv("attendanceData.csv")
str(df)
```

```
## ’data.frame’:    11591 obs. of  8 variables:
##  $ Date           : chr  "2009-10-27" "2009-10-27" "2009-10-27" "2009-10-27" ...
##  $ Visitor.Neutral: chr  "Boston Celtics" "Washington Wizards" "Houston Rockets" "Los Angeles Clipper
##  $ PTS            : int  95 102 87 92 109 91 106 59 93 89 ...
##  $ Home.Neutral   : chr  "Cleveland Cavaliers" "Dallas Mavericks" "Portland Trail Blazers" "Los Angel
##  $ PTS.1          : int  89 91 96 99 120 101 120 92 115 102 ...
##  $ Attend.        : int  20562 19871 20403 18997 17998 20152 17461 18624 19600 18203 ...
##  $ Season         : int  2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
##  $ Home_Win       : chr  "False" "False" "True" "True" ...
```

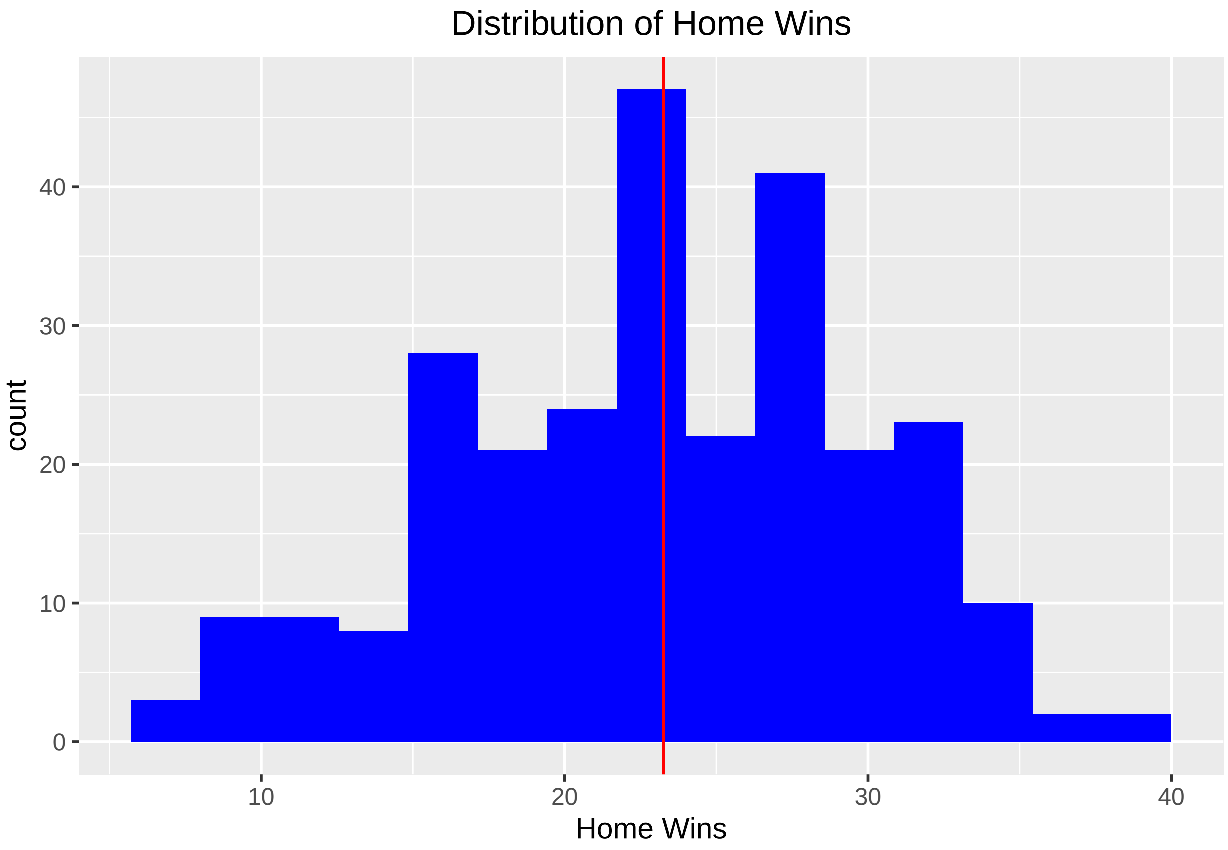
```
df$Home_Win <- as.logical(df$Home_Win)

df<-subset(df, Season != "2012")
names(df)[4] <- 'Team'

muA <- mean(df$Attend.)

#dist of wins
ggplot(df, aes(x=Attend.)) + geom_histogram(bins=15,fill="blue")+
  ggtitle("Distribution of Attendance")+
  theme(plot.title = element_text(hjust = 0.5))+
  geom_vline(data=df, aes(xintercept = muA), colour="red") +
  xlab("Attendance")
```



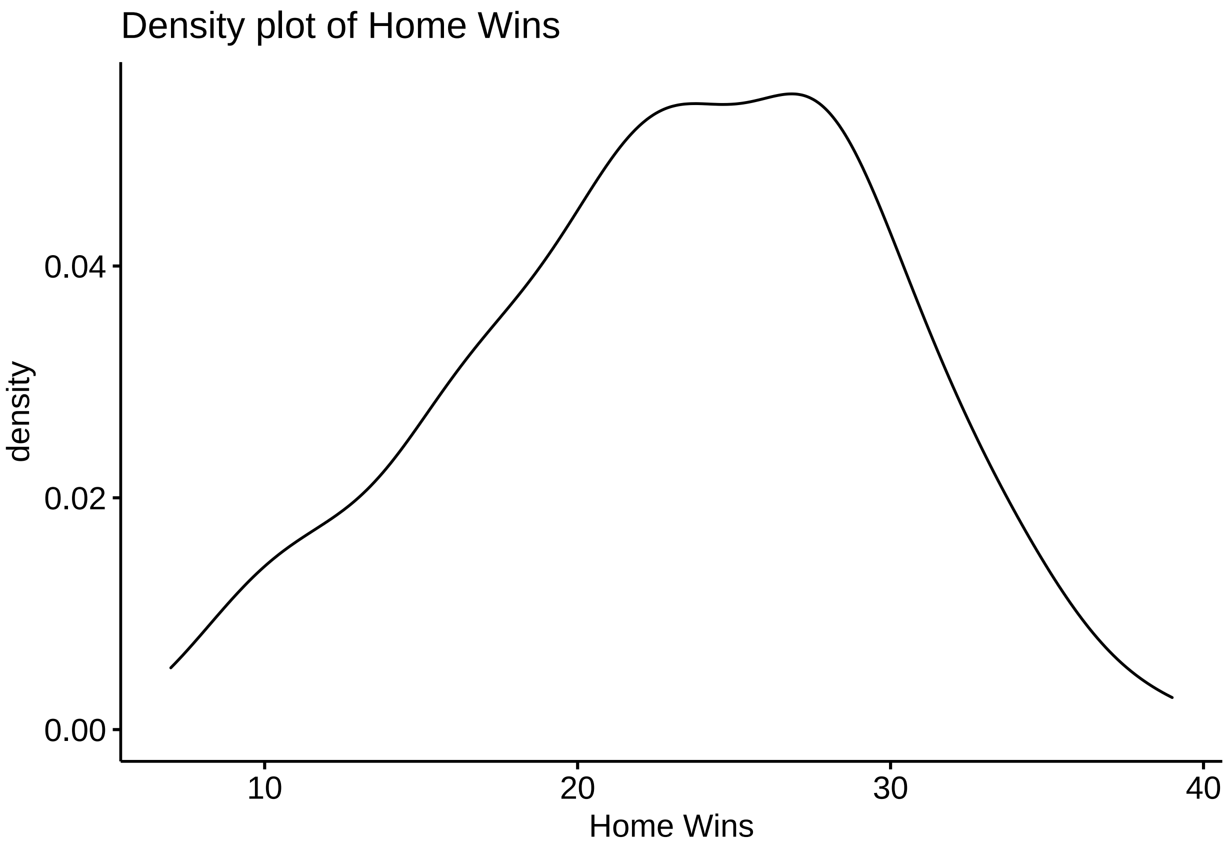
```
#aggregate(df['Attend.'], by=df['Season'], mean)

df3<-aggregate(Attend. ~ Season + Team, data = df, FUN = mean, na.rm = TRUE)

df2<-aggregate(Home_Win ~ Season + Team, data = df, FUN = sum, na.rm = TRUE)

attendDF <- merge(df3,df2,by=c("Season","Team"))

cor(attendDF$Attend.,attendDF$Home_Win)
```

```
## [1] 0.3961002
```

```
muHW <- mean(attendDF$Home_Win)

#dist of home wins
ggplot(attendDF, aes(x=Home_Win)) + geom_histogram(bins=15,fill="blue")+
  ggtitle("Distribution of Home Wins")+
  theme(plot.title = element_text(hjust = 0.5))+
  geom_vline(data=attendDF, aes(xintercept = muHW), colour="red") +
  xlab("Home Wins")
```



```
ggdensity(attendDF$Home_Win,
          main = "Density plot of Home Wins",
          xlab = "Home Wins")
```



```
#Home_Win not normally distributed
shapiro.test(attendDF$Home_Win)
```

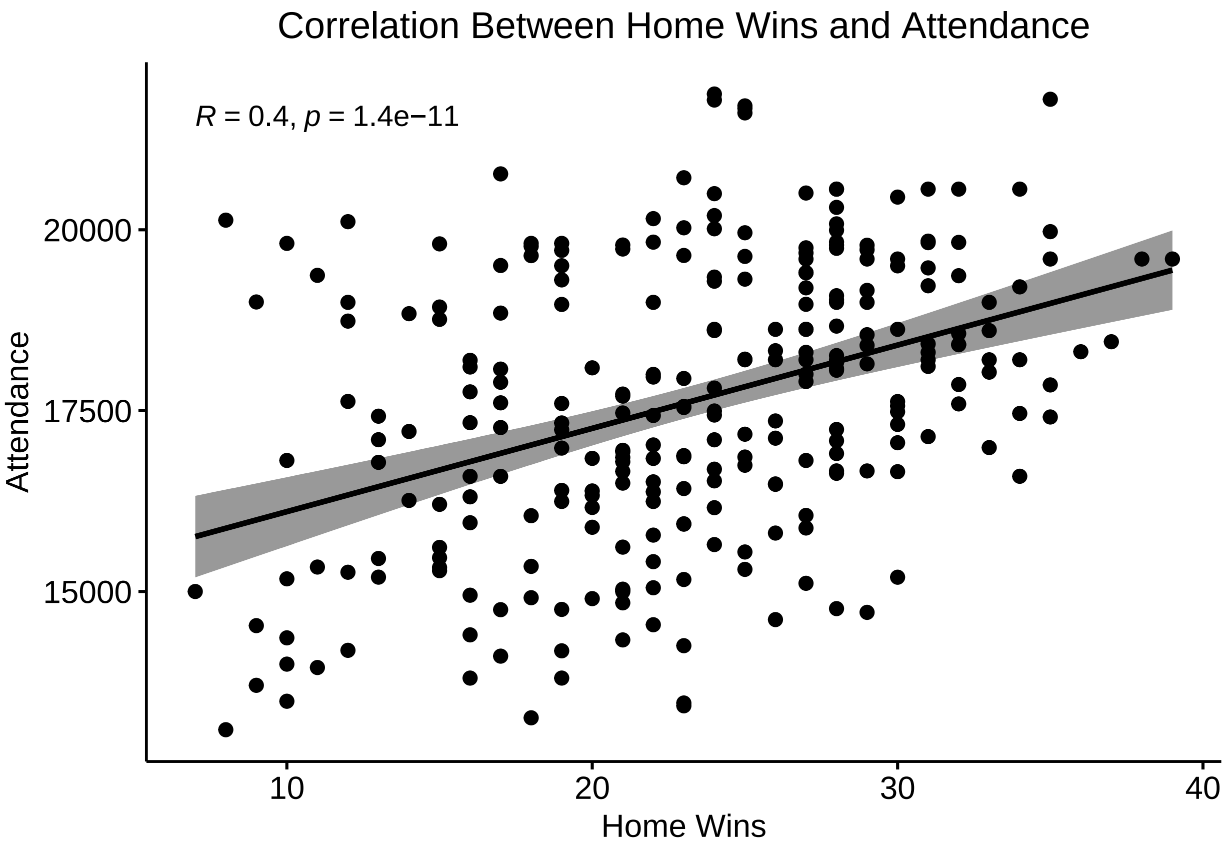
```
##
##  Shapiro-Wilk normality test
##
## data:  attendDF$Home_Win
## W = 0.98577, p-value = 0.008845
```

```
#attendance not normal
shapiro.test(attendDF$Attend.)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  attendDF$Attend.
## W = 0.98365, p-value = 0.003504
```

```
ggscatter(attendDF, x = "Home_Win", y = "Attend.",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "Home Wins", ylab = "Attendance",
          title = "Correlation Between Home Wins and Attendance")+
          theme(plot.title = element_text(hjust = 0.5))
```

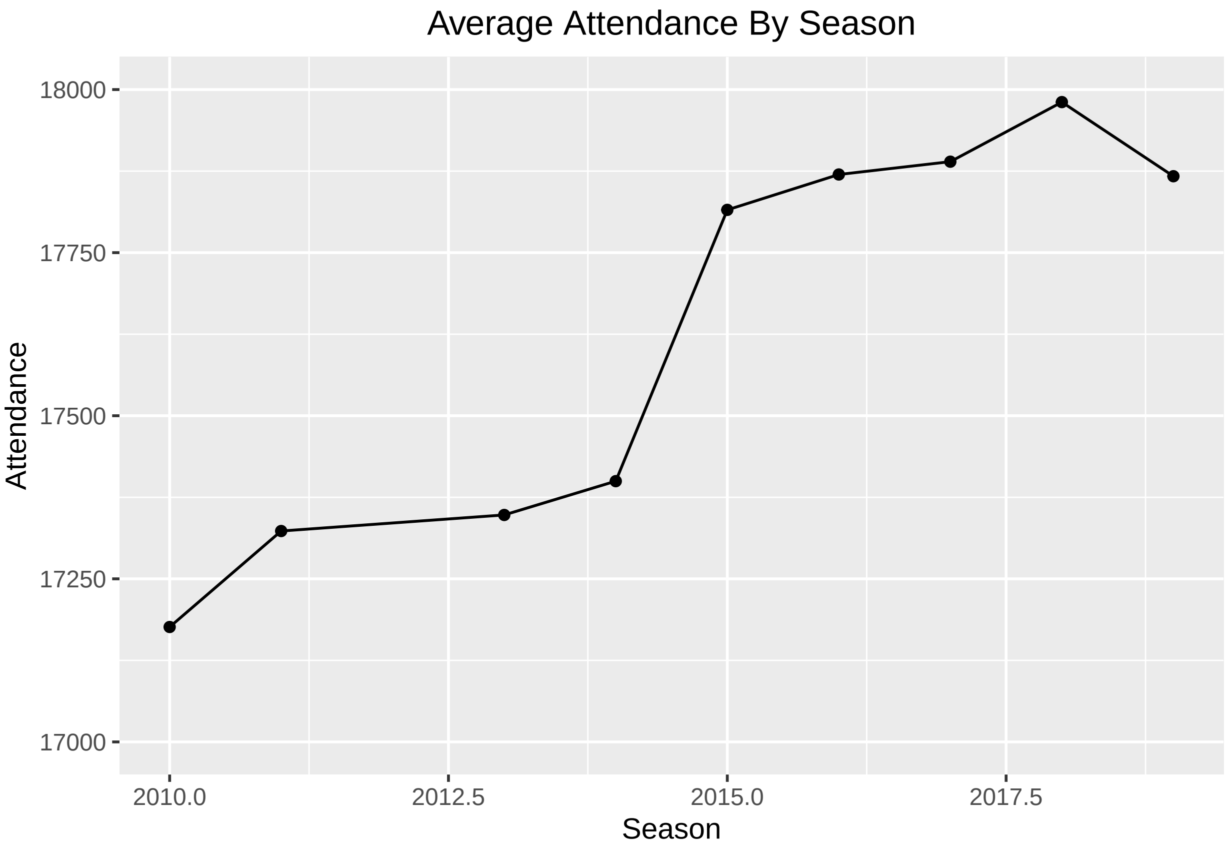
```
## 'geom_smooth()' using formula 'y ~ x'
```



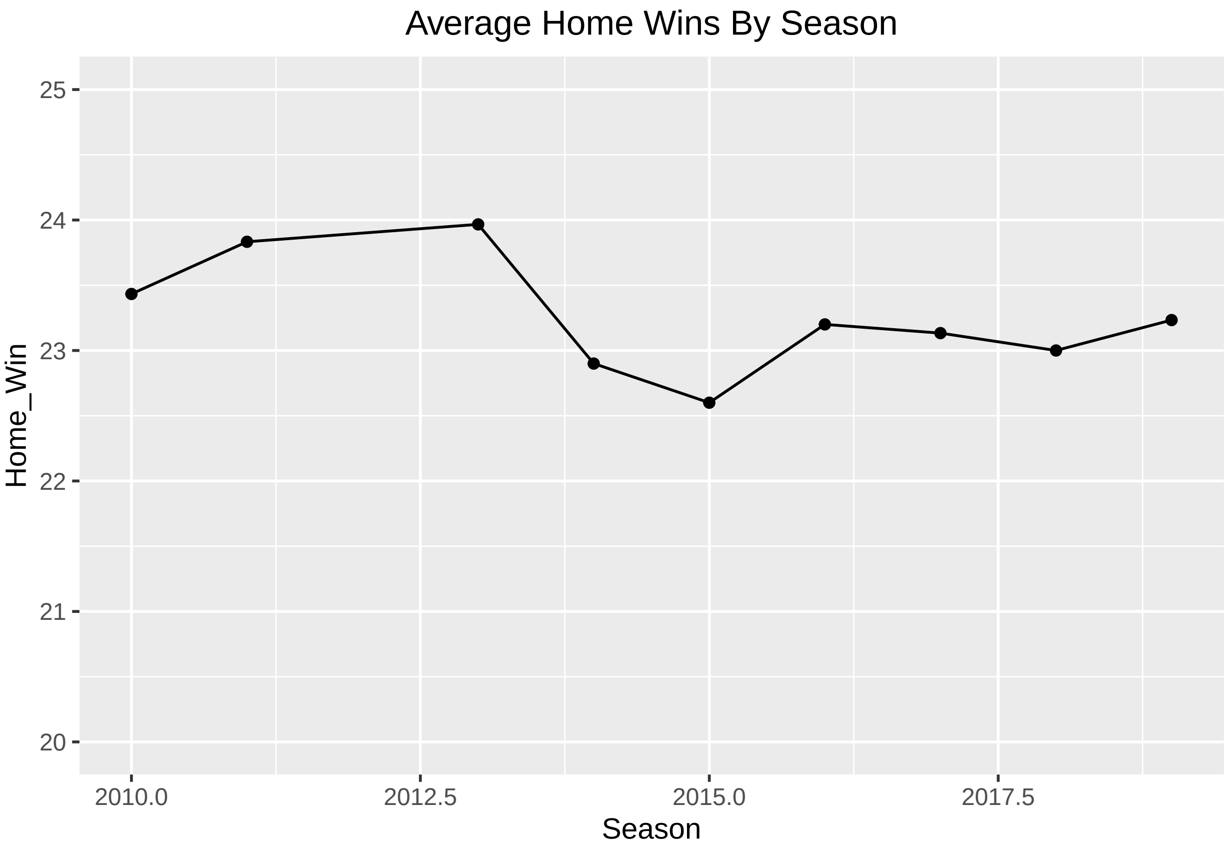
```
          #+geom_text(aes(label=Home.Neutral), size=3)

#still weak correlation when adjusting normality
cor(log(attendDF$Attend.),log(attendDF$Home_Win))
```

```
## [1] 0.3959053
```

```
avg_attd<-aggregate(attendDF['Attend.'], by=attendDF['Season'], mean)

ggplot(data=avg_attd, aes(x=Season, y=Attend., group=1)) +
  geom_line()+
  geom_point()+
  ylim(17000,18000)+
  ggtitle("Average Attendance By Season")+
  ylab("Attendance")+
  theme(plot.title = element_text(hjust = 0.5))
```



```
avg_W<-aggregate(attendDF['Home_Win'], by=attendDF['Season'], mean)

ggplot(data=avg_W, aes(x=Season, y=Home_Win, group=1)) +
  geom_line()+
  geom_point()+
  ylim(20,25)+
  ggtitle("Average Home Wins By Season")+
  theme(plot.title = element_text(hjust = 0.5))
```


Average Home Wins By Season
25
24
23
22
21
20
Home_Win
2010.0
2012.5
2015.0
2017.5
Season