An Analytics Project on Basketball

Sophia Huang

Introduction:

I grew up watching a lot basketball with my dad whenever I could. When I was younger, the game seemed simple: one team was better than the other (for me, that was always either the Lakers or the Celtics), and the team that was better would win. As I grew older, I learned that there is a variety of both measurable and implicit factors that go into the game of basketball. So, I wanted to do this project on analyzing a portion of the NBA with some math and data manipulation.

I will be working with data that details the Dallas Maverick's 2021-2022 regular season in the NBA. The Dallas Mavericks are part of the NBA's Western Conference, and I chose this team by using a random NBA team generator. Specifically, I will be looking at their traditional box scores, advanced box scores, and their overall team performance in each game. The box scores tell us how each player on the team performed in each game and gives an overall summary of the game. Originally, I planned on just grabbing a CSV from Kaggle, but then I decided I wanted to have some fun scraping the NBA website.

Data Source:

I gathered my data in VSCode using Python with the amazing NBA API created by Swar Patel. You can access the NBA API here: https://github.com/swar/nba_api (https://github.com/swar/nba_api)! This API is extremely useful because you can scrape almost any data you want directly from the NBA website.

I gathered the box scores datasets by using Swar's endpoints documentations. You can access the endpoints by going to the link provided above then going to the following directories:

`nba_api/docs/nba_api/stats/endpoints/boxscoreadvancedv2.md`

`https://github.com/swar/nba_api/blob/master/docs/nba_api/stats/endpoints/boxscoretraditionalv2.md`

To pull data on the Mavericks, first I pulled all of the teams. This is in the statics folder and you can reach it by following this directory:

`nba_api/docs/nba_api/stats/static/teams.md`

Data Import:

```
traditional_box_scores <- read_csv("mavs_box_scores21_22.csv")
games <- read_csv("mavs.csv")
advanced_box_scores <- read_csv("mavsBoxScoresAdvanced21_22.csv")
```

What this code is doing: Importing in my three CSV files from the same folder as this project.

Data Exploration:

First, I just want to do a bit of data cleaning to make the data easier for me to read.

```
#1
mavs_tradboxscore <- traditional_box_scores %>% filter(TEAM_ABBREVIATION == "DAL")
#2
colnames(mavs_tradboxscore) <- paste("PLAYER",colnames(traditional_box_scores),sep="_")
#3
player_bxsc <- mavs_tradboxscore %>% rename("PLAYER_ID" = "PLAYER_PLAYER_ID") %>%
  rename("PLAYER_NAME" = "PLAYER_PLAYER_NAME")%>%
  rename("GAME_ID" = "PLAYER_GAME_ID", "RN" = "PLAYER_...1")

player_bxsc
```

What this code is doing:

Step 1: Filtering the traditional box scores CSV for only Mavericks data.

Step 2: Changing all the column names in the traditional box scores to add a prefix of “PLAYER_”.

Step 3: Because one column is named PLAYER_ID, step 2 turns this into PLAYER_PLAYER_ID. I renamed two columns where this occured in to PLAYER_ID and PLAYER_NAME, respectively. I also renamed the PLAYER_GAME_ID column to GAME_ID so it would be easier to join with another dataset later.

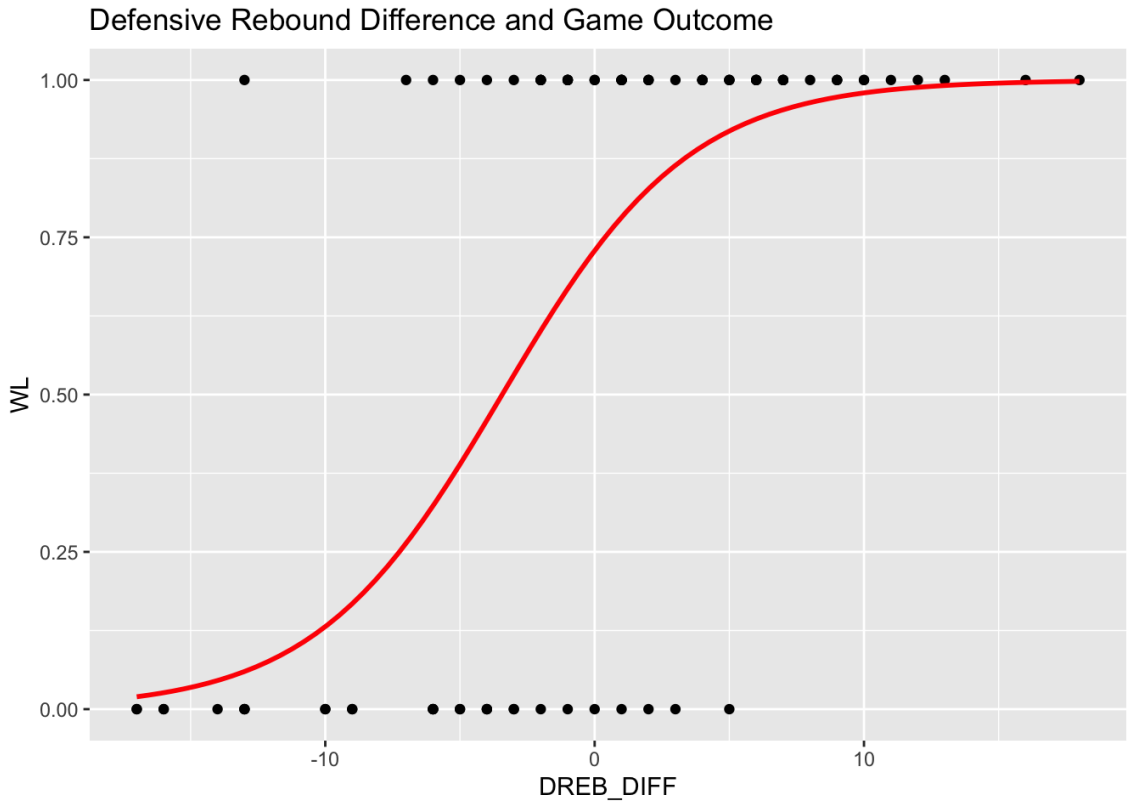
First Analysis Question: How can defensive rebounds affect the overall team performance?

Rebound: When a player gains possession of the ball after a missed shot.

Offensive rebound - possession of ball does not change teams. A player on the shooter’s team gains the rebound.

Defensive rebound - possession of ball changes teams. An opponent on the shooter’s team gains the rebound.

```
#1
dal_bs <- traditional_box_scores %>% filter(TEAM_ABBREVIATION == "DAL")
dal_dreb <-dal_bs%>%group_by(GAME_ID)%>% summarize(DREB = sum(DREB, na.rm = TRUE))

#2
opp_bs <- anti_join(traditional_box_scores, dal_bs, by = "TEAM_ABBREVIATION")
opp_dreb <- opp_bs%>%group_by(GAME_ID)%>% summarize(OPP_DREB = sum(DREB, na.rm = TRUE))

#3
dreb_diff <-left_join(dal_dreb, opp_dreb) %>% mutate(DREB_DIFF = DREB - OPP_DREB)

#4
dreb <- left_join(dreb_diff, games, by = "GAME_ID") %>% mutate(WL = ifelse(WL =="W", 1, 0))

#5
model <- glm(WL ~ DREB_DIFF, data = dreb, family = binomial)
#summary(model)
dreb %>% ggplot() +
  geom_point(aes(x=`DREB_DIFF`, y = WL)) +
  stat_smooth(aes(x=`DREB_DIFF`, y = WL), color = "red", method="glm", se=FALSE, method.args = list(famil
y=binomial)) +
  ggtitle("Defensive Rebound Difference and Game Outcome")
```



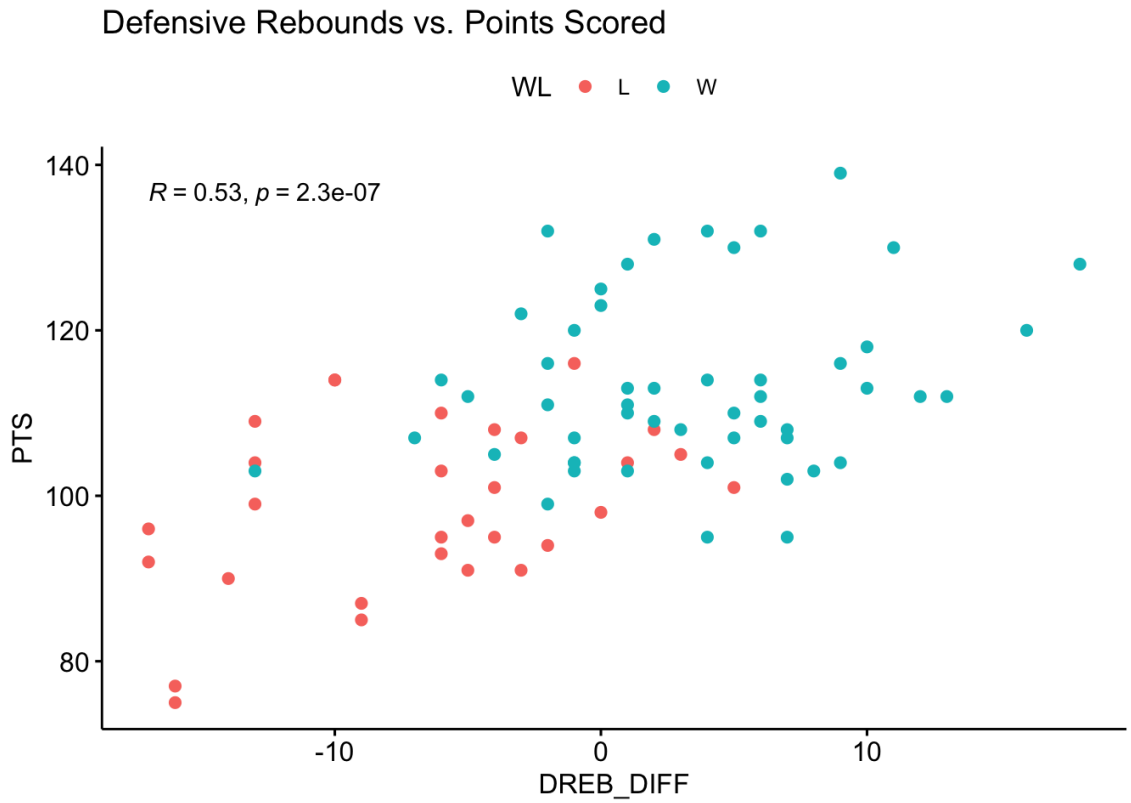
What this code is doing:

1. Creating a new tibble called dal_bs that filters for all of the box scores for the Mavs. I didn't use the one I created above was because it was already stored with all of the column names renamed. Then, I'm creating a tibble called dal_dreb that takes in the new tibble I created and summing up the defensive rebounds for each game.
2. Creating a tibble called opp_bs that looks at the opponent box scores. I did an anti join to keep all team abbreviations in traditional_box_scores that did not appear in dal_bs. Then, because I know that each game id only contains one opponent, I summed the amount of defensive rebounds to obtain the number of defensive rebounds the Mavs' opponents had.
3. Joined the two tibbles I created to keep all observations from both tibbles. Because they have the same game ids, I just decided to do a left join. I then wanted to calculated the defensive rebound difference between the Mavericks. The DREB difference is calculated by subtracting the number of rebounds the opponent had from the number of rebounds the Mavs had. Originally, I only looked at the Mavs' rebounds, but I realized this wasn't useful because the same number of rebounds in two games can't tell us how well they did compared to their opponent. Because you're playing against an opponent in basketball, everything one team does is relative to the other teams' performances.
4. Joining with the games tibble because this actually tells us whether or not the Mavericks won or lost. Then, I'm changing the WL column to be binary, with "1" representing a win and "0" a loss.
5. Plotting a logistic regression model to see how the defensive rebound difference affects the outcome of a game.

Observations: From this graph, we can see that as the defensive rebound difference increases (Mavs defensive rebounds - opponent defensive rebounds), the WL probability is closer to 1, which denotes a win.

To examine defensive rebounds further, I want to look at the correlation between the defensive rebound difference and points scored.

```
dreb %>% mutate(WL = ifelse(WL ==1, "W", "L")) %>%
  ggscatter(x="DREB_DIFF", y = "PTS", color = "WL", title = "Defensive Rebounds vs. Points Scored")+
  stat_cor(method="pearson")
```



What this code is doing:

Changing the win loss column back into "W" and "L" so we can see on the scatterplot which games the Maverick's won/lost. Plotting defensive rebound difference against points then adding a Pearson correlation coefficient.

Observations:

As we can see from these graphs, the difference in defensive rebounds can influence a game's outcome significantly. Our linear model has an adjusted R^2 of 0.5174 and a p-value of $1.629 \cdot 10^{-14}$. As the defensive rebound difference (Mavs rebounds - opponent rebounds) increases, the Maverick's have a higher chance of winning. Furthermore, we can see that as the defensive rebound difference increases, the majority of the points are blue, indicating a Mavericks win. The points closer to the y-axis are red, indicating that the Mavericks lost.

I also want to create a linear model using the difference in defensive rebounds to predict the difference in points.

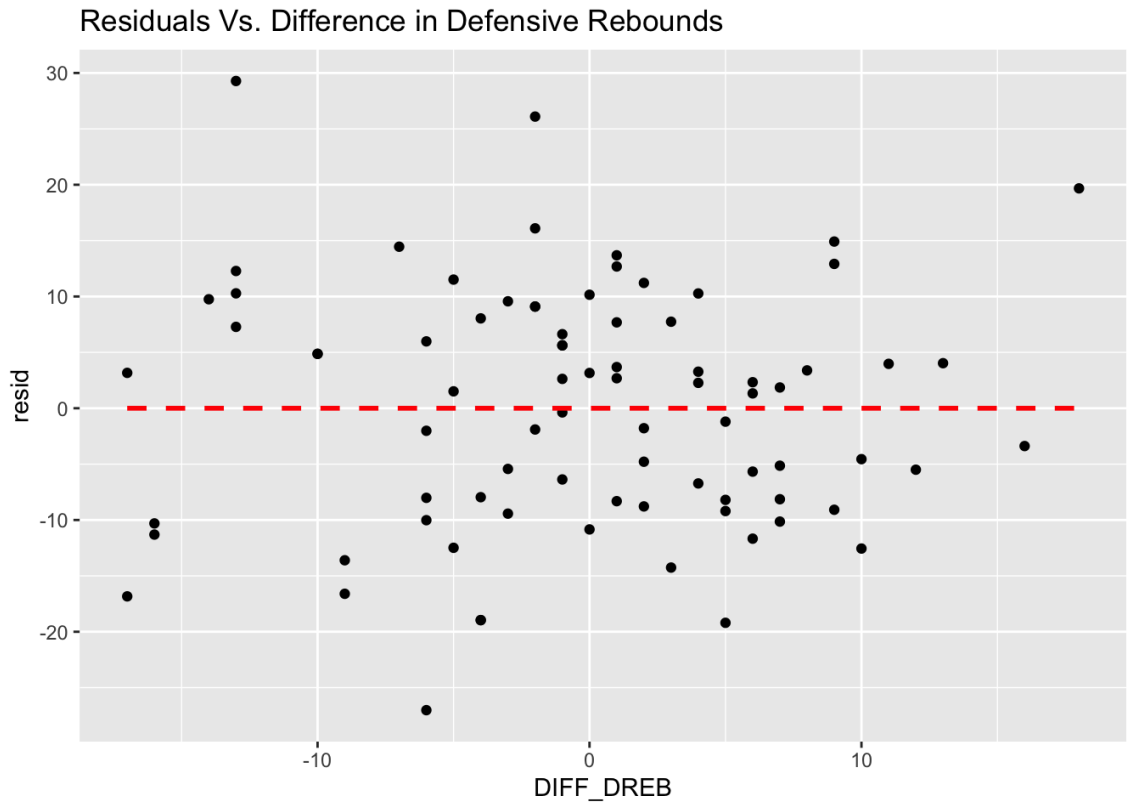
```
dal_point_dreb <- traditional_box_scores %>% filter(TEAM_ABBREVIATION == "DAL") %>% group_by(GAME_ID)%>%
  summarize(DREB = sum(DREB, na.rm = TRUE), DAL_PTS = sum(PTS, na.rm = TRUE)) %>% arrange(desc(DREB)) %>%
mutate("TEAM_ABBREVIATION" = "DAL")
# Looking at total Dallas points and defensive rebounds per game

opp_point_dreb <- anti_join(traditional_box_scores, dal_point_dreb, by = "TEAM_ABBREVIATION") %>%
  group_by(GAME_ID) %>% summarize(OPP_DREB = sum(DREB, na.rm = TRUE), OPP_PTS = sum(PTS, na.rm = TRUE))
# Looking at total Opponent points and defensive rebounds per game

point_dreb <- inner_join(dal_point_dreb, opp_point_dreb, by = "GAME_ID") %>% mutate(DIFF_DREB = DREB - OP
P_DREB, DIFF_PTS = DAL_PTS - OPP_PTS) %>%
  select(GAME_ID, DREB, OPP_DREB, DIFF_DREB, DAL_PTS, OPP_PTS, DIFF_PTS)
#Taking the difference in defensive rebounds and points scored

model <- lm(DIFF_PTS ~ DIFF_DREB, data = point_dreb) # Creating linear model to see if difference in defe
nsive rebounds can help predict difference in points
model
```

```
##
## Call:
## lm(formula = DIFF_PTS ~ DIFF_DREB, data = point_dreb)
##
## Coefficients:
## (Intercept)    DIFF_DREB
##       3.843        1.471
```

```
predictions <- point_dreb %>% add_predictions(model) %>% add_residuals(model)  # Adding residuals and pre
dictions to out original dataset
predictions
```

```
## # A tibble: 82 × 9
##    GAME_ID     DREB OPP_DREB DIFF_DREB DAL_PTS OPP_PTS DIFF_PTS  pred  resid
##    <chr>      <dbl>    <dbl>     <dbl>   <dbl>   <dbl>    <dbl> <dbl>  <dbl>
##  1 0022100400    44       36         8     103      84       19 15.6    3.39
##  2 0022100586    44       33        11     130     106       24 20.0    3.97
##  3 0022100639    44       31        13     112      85       27 23.0    4.03
##  4 0022100420    43       42         1     104     107       -3  5.31  -8.31
##  5 0022101209    42       24        18     128      78       50 30.3   19.7
##  6 0022100116    41       35         6     109     108        1 12.7  -11.7
##  7 0022100222    41       41         0      98     105       -7  3.84 -10.8
##  8 0022100550    41       34         7      95      86        9 14.1   -5.14
##  9 0022100603    41       31        10     113      99       14 18.6   -4.55
## 10 0022100820    41       32         9     116      86       30 17.1   12.9
## # … with 72 more rows
```

```
predictions%>% ggplot()+
  geom_point(aes(x=DIFF_DREB, y = resid))+
  geom_smooth(mapping=aes(x=DIFF_DREB, y=resid), method='lm', se = FALSE, linetype = "dashed", color = "r
ed") +
  ggtitle("Residuals Vs. Difference in Defensive Rebounds")
```



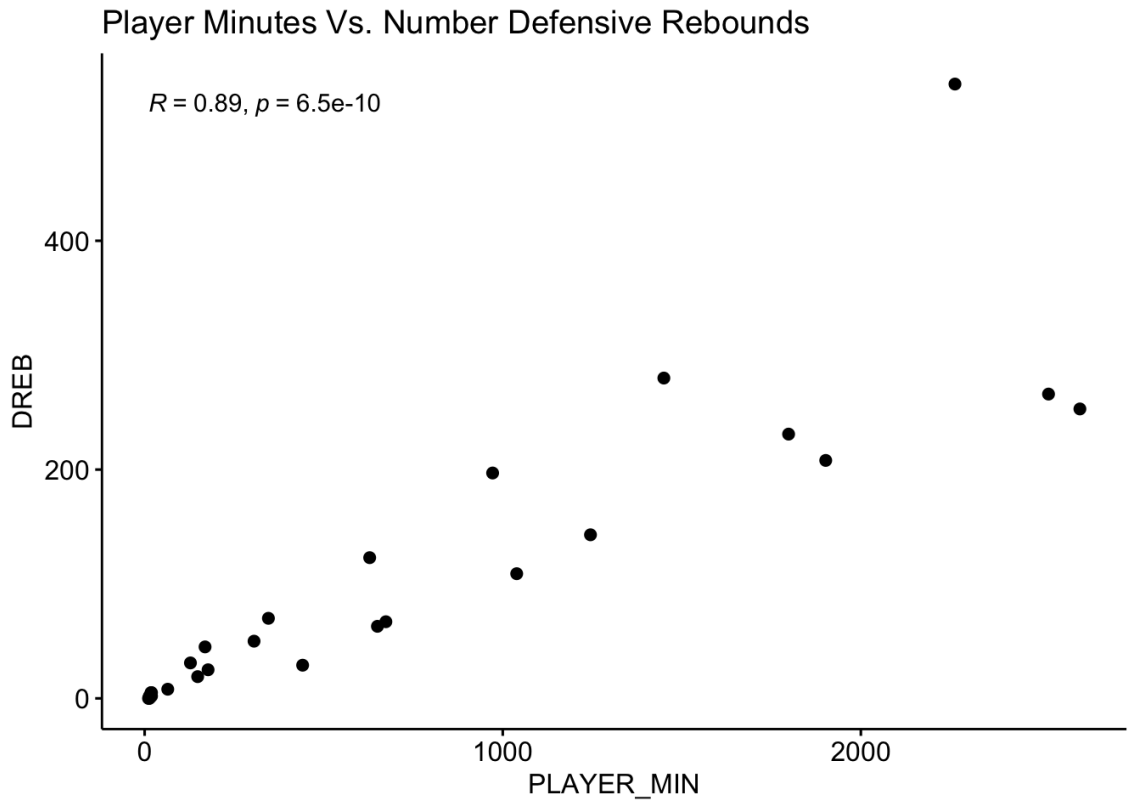
```
# Plotting residuals against our independent variable to determine whether this is a good model

summary(model)
```

```
##
## Call:
## lm(formula = DIFF_PTS ~ DIFF_DREB, data = point_dreb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.016  -8.285   1.686   7.729  29.282
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.843      1.202   3.197  0.00199 **
## DIFF_DREB      1.471      0.157   9.373 1.63e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.87 on 80 degrees of freedom
## Multiple R-squared:  0.5234, Adjusted R-squared:  0.5174
## F-statistic: 87.85 on 1 and 80 DF,  p-value: 1.629e-14
```

From our linear model, we get an adjusted R-squared value of 0.5714 and a p-value of $1.29 \cdot 10^{-14}$. We can see that not only is this statistically significant, but also that the difference in defensive rebounds can account for approximately 57.14% of the variation in difference in points. This tells us that defensive rebounds are actually quite important in determining the point difference between teams, which is related to a team's win/loss outcome.

For this question, I also want to look at if the number of defensive rebounds a player has is correlated with the amount of minutes they play to see if certain players are actually better at getting rebounds than others.

```
most_minutes <- player_bxsc %>%
  mutate(PLAYER_MIN = (period_to_seconds(hms(PLAYER_MIN))/3600))%>%
  group_by(PLAYER_NAME) %>%
  summarize(PLAYER_MIN = sum(PLAYER_MIN, na.rm=TRUE)) %>%
  arrange(desc(PLAYER_MIN))
# Looking at which players have the most playing time

player_dreb <- traditional_box_scores %>% filter(TEAM_ABBREVIATION == "DAL") %>% group_by(PLAYER_NAME)%>%
  summarize(DREB = sum(DREB, na.rm = TRUE)) %>% arrange(desc(DREB))
# Looking at each individual players number of defensive rebounds

inner_join(player_dreb, most_minutes, by = "PLAYER_NAME") %>%
  ggscatter(x = "PLAYER_MIN", y = "DREB")+
  stat_cor(method="pearson") +
  ggtitle("Player Minutes Vs. Number Defensive Rebounds")
```




```
# Keeping all players that appear in both of the created tibbles then plotting to see if there exists a r
elationship between a player's defensive rebounds and their minutes played in the season
```

Observations:

From this, we can see that there exists such a significant and strong correlation between defensive rebounds and player minutes. I wanted to explore this because I noticed some players were averaging many more defensive rebounds and wondered if that was because they were actually better at getting defensive rebounds, or if there was another variable involved.

From this graph, we can see that if a player isn't playing many minutes and doesn't have as many defensive rebounds, it does not mean that they are a bad player. However, let's say that a player played for over 2000 minutes and had fewer than 200 defensive rebounds. This could indicate to the coaches that this player might not be the best player might be playing weaker defensively than his teammates. Furthermore, it's impressive to see that there is one player who managed to get significantly more rebounds than his teammates.

Conclusion/How could this influence coaching decision?

From this, we can see that it is pretty important to get the defensive rebound. Not only does it prevent the other team from scoring again, but it also gives your team a chance to score. The opponent has now lost possession of the ball. Furthermore, this is important because defensive rebounds change the direction of the game by turning the defensive team to the offensive team and vice versa, which can impact a team's tempo.

Additionally, this could influence coaching decision because it would be beneficial for the coach to put in players with a high rebound percent. Putting in players who all have lower rebound percentages than the other team could be extremely harmful to the team. It also shows the importance in playing defense. If a team plays well offensively but much worse than the other team defensively, their chances of winning probably won't be extremely high. Or for example, there isn't much time on the shot clock and the other team has possession of the ball then misses their field goal. It could be advantageous for the Mavs if the coach positioned a strong defensive player who also averages many assists to pass the ball to another teammate to score a field goal. On the other hand, it's important to position players with high offensive rebound percentages near the net when the Mavs have possession of the ball to combat the other teams' defensive players.

With more time, I would like to dive in deeper to this question by looking at offensive rebounds and other defensive strategies as well.

Second Analysis Question: Are there any additional variables we can combine with defensive rebounds to help us create a better linear model to predict the difference in score (W/L)?

```
dallas <- advanced_box_scores %>% filter(TEAM_ABBREVIATION == "DAL") %>%
  group_by(GAME_ID)%>%
  summarize(DAL_TOV_PCT = mean(TM_TOV_PCT, na.rm = TRUE), DAL_EFG = mean(EFG_PCT, na.rm = TRUE), DAL_DREB
_PCT = mean(DREB_PCT, na.rm = TRUE)) %>% mutate("TEAM_ABBREVIATION" = "DAL")
dallas
```

```
## # A tibble: 82 × 5
##    GAME_ID    DAL_TOV_PCT DAL_EFG DAL_DREB_PCT TEAM_ABBREVIATION
##    <chr>            <dbl>   <dbl>        <dbl> <chr>
##  1 0022100014        9.42   0.385       0.129  DAL
##  2 0022100029        6.37   0.394       0.0905 DAL
##  3 0022100052       17.7    0.489       0.107  DAL
##  4 0022100069        5.87   0.540       0.0863 DAL
##  5 0022100075       10.4    0.351       0.141  DAL
##  6 0022100089        4.45   0.528       0.165  DAL
##  7 0022100104        7.89   0.379       0.166  DAL
##  8 0022100116       16.9    0.347       0.127  DAL
##  9 0022100136       12.0    0.582       0.107  DAL
## 10 0022100150        5.16   0.452       0.113  DAL
## # … with 72 more rows
```

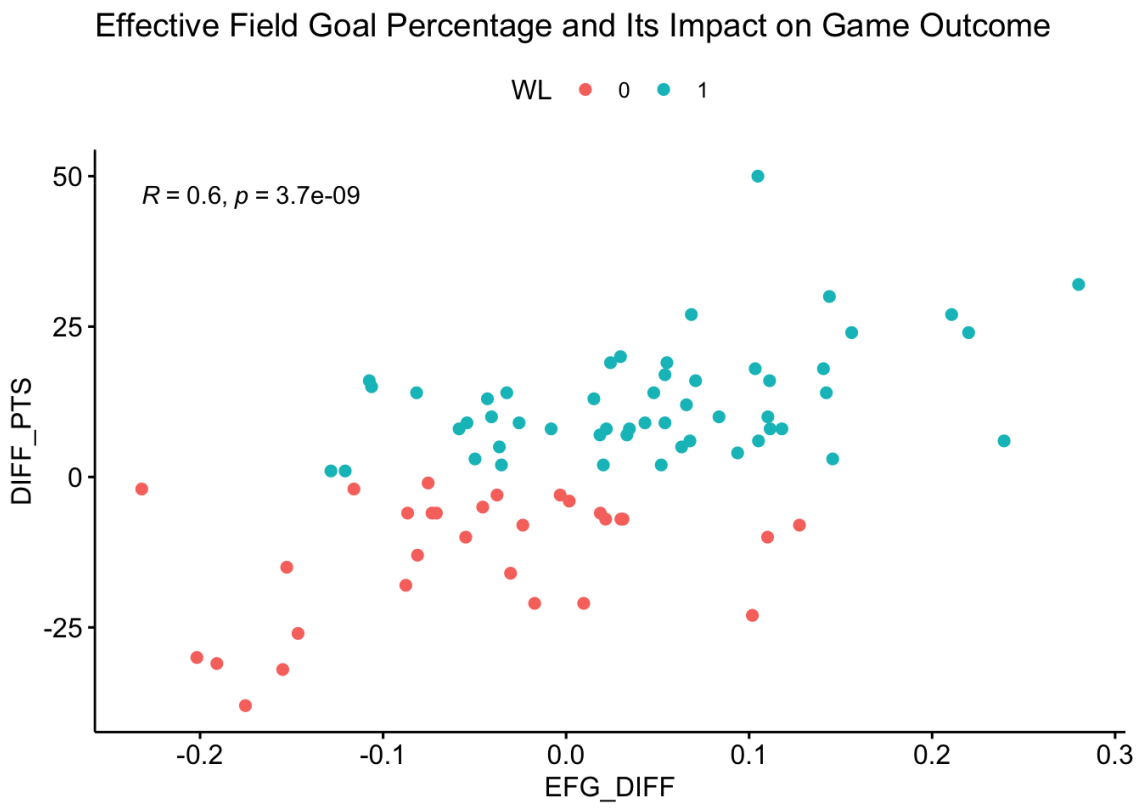
```
# Now, looking at percents: turnover percent, effective field goal percent, defensive

opponent <- anti_join(advanced_box_scores, dallas, by = "TEAM_ABBREVIATION") %>%
  group_by(GAME_ID) %>% summarize(OPP_TOV_PCT = mean(TM_TOV_PCT, na.rm = TRUE), OPP_EFG = mean(EFG_PCT, n
a.rm = TRUE), OPP_DREB_PCT = mean(DREB_PCT, na.rm = TRUE))
opponent
```

```
## # A tibble: 82 × 4
##    GAME_ID    OPP_TOV_PCT OPP_EFG OPP_DREB_PCT
##    <chr>            <dbl>   <dbl>        <dbl>
##  1 0022100014        5.44   0.532       0.174
##  2 0022100029       11.1    0.360       0.103
##  3 0022100052        9.46   0.529       0.128
##  4 0022100069       10.8    0.477       0.123
##  5 0022100075       15.2    0.542       0.151
##  6 0022100089        9.46   0.423       0.101
##  7 0022100104        4.74   0.532       0.190
##  8 0022100116       10.1    0.468       0.088
##  9 0022100136       13.3    0.436       0.0937
## 10 0022100150       10.8    0.340       0.0935
## # … with 72 more rows
```

```
advancedstats <- left_join(dallas, opponent) %>% mutate(TM_TOV_PCT_DIFF = DAL_TOV_PCT - OPP_TOV_PCT, EFG_
DIFF = DAL_EFG - OPP_EFG, DREB_PCT_DIFF = DAL_DREB_PCT - OPP_DREB_PCT) %>%
  select(GAME_ID:DAL_TOV_PCT, OPP_TOV_PCT, TM_TOV_PCT_DIFF, DAL_EFG, OPP_EFG, EFG_DIFF, DAL_DREB_PCT, OPP
_DREB_PCT, DREB_PCT_DIFF)
advancedstats
```

```
## # A tibble: 82 × 10
##    GAME_ID    DAL_TOV_PCT OPP_TOV_PCT TM_TOV_PCT_DIFF DAL_EFG OPP_EFG EFG_DIFF
##    <chr>            <dbl>       <dbl>           <dbl>   <dbl>   <dbl>    <dbl>
##  1 0022100014        9.42        5.44            3.98   0.385   0.532  -0.146
##  2 0022100029        6.37       11.1            -4.71   0.394   0.360   0.0346
##  3 0022100052       17.7         9.46            8.28   0.489   0.529  -0.0408
##  4 0022100069        5.87       10.8            -4.91   0.540   0.477   0.0631
##  5 0022100075       10.4        15.2            -4.82   0.351   0.542  -0.191
##  6 0022100089        4.45        9.46           -5.00   0.528   0.423   0.105
##  7 0022100104        7.89        4.74            3.14   0.379   0.532  -0.153
##  8 0022100116       16.9        10.1             6.85   0.347   0.468  -0.121
##  9 0022100136       12.0        13.3            -1.36   0.582   0.436   0.146
## 10 0022100150        5.16       10.8            -5.68   0.452   0.340   0.111
## # … with 72 more rows, and 3 more variables: DAL_DREB_PCT <dbl>,
## #   OPP_DREB_PCT <dbl>, DREB_PCT_DIFF <dbl>
```

```
game_and_win <- point_dreb %>% select(GAME_ID, DIFF_PTS)
advancedstats_joined <- inner_join(advancedstats, game_and_win, by = "GAME_ID")
logisticregression <- advancedstats_joined%>% mutate(WL = ifelse(DIFF_PTS<0, "0", "1")) %>% select(DIFF_P
TS, EFG_DIFF, DREB_PCT_DIFF, TM_TOV_PCT_DIFF, WL)

logisticregression %>%
  ggscatter(x = "EFG_DIFF", y = "DIFF_PTS", color = "WL")+
  stat_cor(method="pearson") +
  ggtitle("Effective Field Goal Percentage and Its Impact on Game Outcome")
```



```
lm_efg <- lm(DIFF_PTS ~ EFG_DIFF, data = logisticregression)
summary(lm_efg)
```

```
##
## Call:
## lm(formula = DIFF_PTS ~ EFG_DIFF, data = logisticregression)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.411  -7.208   1.799   7.652  38.319
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.294      1.406   1.631    0.107
## EFG_DIFF      89.569     13.525   6.623 3.73e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.66 on 80 degrees of freedom
## Multiple R-squared:  0.3541, Adjusted R-squared:  0.346
## F-statistic: 43.86 on 1 and 80 DF,  p-value: 3.731e-09
```

Creating a linear model for prediting difference in points from Effective Field Goal Percent gives us an adjusted R-squared of 0.346 and a p-value of $3.731 \cdot 10^{-9}$. This is quite statistically significant, but EFG difference only accounts for about 34.6% of the variation in data. It could be interesting to look at this component with defensive rebounds once we convert to a standard unit of measurement.

Conclusion/How could this impact the game?

This goes to show how important it is to not only score points, but also to play efficiently. Many times, just knowing how many points a player made in a game might not be helpful. We also have to know how many attempts they made to score and out of those attempts how many shots were successful. We want to visualize how this player is doing compared to their teammates as well as their opponents.

We care about effective field goal percent because we want to know whether a team is playing more efficient than their opponent. For example, one team may have more possessions than the other team, but if they're not maximizing their value by actually scoring at a higher ratio, then they're not being efficient with each possession they obtain. Additionally, the coach could make a decision to rest their star player while playing a team with a lower effective field goal percent per player. They can choose to substitute another player in who helps maintain their higher effective field goal percent over their opponent. The benefits from this are that the team is still putting themselves at a position where they can win the game, and they can rest their star player, utilizing them for when a matchup is more even. Furthermore, this is important in the case that a player becomes injured to choose a substitution.

Third Analysis Question: Is there an advantage for the Mavericks when certain players are playing?

First, let's see how many games the Mavericks played in the 2021-2022 regular season:

```
games <- read_csv("mavs.csv") %>% rename("id" = "...1")
games
```

```
## # A tibble: 82 × 29
##       id SEASON_ID TEAM_ID TEAM_ABBREVIATI… TEAM_NAME GAME_ID GAME_DATE  MATCHUP
##    <dbl>     <dbl>   <dbl> <chr>            <chr>     <chr>   <date>     <chr>
##  1     0     22021  1.61e9 DAL              Dallas M… 002210… 2022-04-10 DAL vs…
##  2     1     22021  1.61e9 DAL              Dallas M… 002210… 2022-04-08 DAL vs…
##  3     2     22021  1.61e9 DAL              Dallas M… 002210… 2022-04-06 DAL @ …
##  4     3     22021  1.61e9 DAL              Dallas M… 002210… 2022-04-03 DAL @ …
##  5     4     22021  1.61e9 DAL              Dallas M… 002210… 2022-04-01 DAL @ …
##  6     5     22021  1.61e9 DAL              Dallas M… 002210… 2022-03-30 DAL @ …
##  7     6     22021  1.61e9 DAL              Dallas M… 002210… 2022-03-29 DAL vs…
##  8     7     22021  1.61e9 DAL              Dallas M… 002210… 2022-03-27 DAL vs…
##  9     8     22021  1.61e9 DAL              Dallas M… 002210… 2022-03-25 DAL @ …
## 10     9     22021  1.61e9 DAL              Dallas M… 002210… 2022-03-23 DAL vs…
## # … with 72 more rows, and 21 more variables: WL <chr>, MIN <dbl>, PTS <dbl>,
## #   FGM <dbl>, FGA <dbl>, FG_PCT <dbl>, FG3M <dbl>, FG3A <dbl>, FG3_PCT <dbl>,
## #   FTM <dbl>, FTA <dbl>, FT_PCT <dbl>, OREB <dbl>, DREB <dbl>, REB <dbl>,
## #   AST <dbl>, STL <dbl>, BLK <dbl>, TOV <dbl>, PF <dbl>, PLUS_MINUS <dbl>
```

We can see that there were 82 total games played by the Mavs in the 2021-2022 regular season based on the 82 rows.

To start off with my analysis, I want to first look at which players are playing the most games and most minutes. These players will be my main areas of focus to answer my first question.

Players with Most Games Played

```
most_games <- player_bxsc %>% count(PLAYER_NAME) %>% arrange(desc(n))

top_10_most_games <- most_games %>% filter(n >= 62)
top_10_most_games
```

```
## # A tibble: 10 × 2
##    PLAYER_NAME             n
##    <chr>               <int>
##  1 Dwight Powell          82
##  2 Dorian Finney-Smith    81
##  3 Jalen Brunson          79
##  4 Josh Green             78
##  5 Boban Marjanovic       76
##  6 Frank Ntilikina        73
##  7 Reggie Bullock         68
##  8 Luka Doncic            65
##  9 Trey Burke             65
## 10 Maxi Kleber            62
```

What this code is doing:

Because we now are only looking at the data for the Mavericks, I want to count which players come up the most in the CSV file. Because this uses box score data, which records how each player performed in each game, I know that each player is only recorded once for each game. Therefore, by counting the players, I am also counting how many games they played in. I then want to arrange my tibble in descending order by number of games played in. (I compared this to the data on the NBA website for some of the players to ensure that the number of games they played in shown in this tibble matched the NBA website)

The reason I decided to filter for where the number of games was greater than or equal to 62 was because after I saw the output of the first tibble, I saw that the player with the 10th most games had 62 games. So, I wanted to create another tibble with just the top 10 players who played the most amount of games.

Players With Most Minutes Played

```
top_10_most_minutes <- most_minutes %>% filter(PLAYER_MIN >= 673.65000)
top_10_most_minutes
```

```
## # A tibble: 10 × 2
##    PLAYER_NAME          PLAYER_MIN
##    <chr>                     <dbl>
##  1 Dorian Finney-Smith      2612.
##  2 Jalen Brunson            2524.
##  3 Luka Doncic              2263.
##  4 Reggie Bullock           1902.
##  5 Dwight Powell            1798.
##  6 Maxi Kleber              1450.
##  7 Tim Hardaway Jr.         1245
##  8 Josh Green               1039.
##  9 Kristaps Porzingis        972.
## 10 Frank Ntilikina           674.
```

What this code is doing:

I also want to look at the players who played the most minutes. The reasoning behind this is that a player could play in a lot of games but only play for 5 minutes in each game, and we wouldn't be able to see this from the previous tibble.

To look at players who played the most minutes, first I wanted to convert the time to minutes. Because we are looking at box scores for each game, the player minutes recorded is only for that given game. This is why I had to sum the player minutes to get the overall minutes each player played in the 2021-2022 season. I then arranged this in descending order by player minutes and created a tibble with the top 10 players who played the most amount of games.

```
most_min_game <- inner_join(top_10_most_games, top_10_most_minutes, by = "PLAYER_NAME") %>%
  arrange(desc(PLAYER_MIN))
most_min_game
```

```
## # A tibble: 8 × 3
##   PLAYER_NAME             n PLAYER_MIN
##   <chr>               <int>      <dbl>
## 1 Dorian Finney-Smith    81      2612.
## 2 Jalen Brunson          79      2524.
## 3 Luka Doncic            65      2263.
## 4 Reggie Bullock         68      1902.
## 5 Dwight Powell          82      1798.
## 6 Maxi Kleber            62      1450.
## 7 Josh Green             78      1039.
## 8 Frank Ntilikina        73       674.
```

What this code is doing:

I am choosing to inner join the previous two tibbles because I want to know which players appeared in both the top 10 most games played and top 10 most minutes played on the Mavs.

Observations:

Finney-Smith, Brunson, Doncic, Bullock, Powell, and Kleber played the most minutes. Of the top 10 players who played the most minutes, 8 players were also in the top 10 for most games played. These are player I want to further analyze because I am assuming that the coaches are choosing to play these individuals because they are the top players on the team.

```
gamewins <- games %>% select(GAME_ID, WL)
player_winloss <- inner_join(gamewins, player_bxsc, by = "GAME_ID") %>% select(GAME_ID, WL, PLAYER_TEAM_A
BBREVIATION:PLAYER_PLUS_MINUS)
player_winloss
```

```
## # A tibble: 1,097 × 29
##    GAME_ID    WL    PLAYER_TEAM_ABBREV… PLAYER_TEAM_CITY PLAYER_ID PLAYER_NAME
##    <chr>      <chr> <chr>               <chr>                <dbl> <chr>
##  1 0022101219 W     DAL                 Dallas              203493 Reggie Bullo…
##  2 0022101219 W     DAL                 Dallas             1627827 Dorian Finne…
##  3 0022101219 W     DAL                 Dallas              203939 Dwight Powell
##  4 0022101219 W     DAL                 Dallas             1628973 Jalen Brunson
##  5 0022101219 W     DAL                 Dallas             1629029 Luka Doncic
##  6 0022101219 W     DAL                 Dallas              203915 Spencer Dinw…
##  7 0022101219 W     DAL                 Dallas             1630182 Josh Green
##  8 0022101219 W     DAL                 Dallas             1627737 Marquese Chr…
##  9 0022101219 W     DAL                 Dallas              202722 Davis Bertans
## 10 0022101219 W     DAL                 Dallas             1629033 Theo Pinson
## # … with 1,087 more rows, and 23 more variables: PLAYER_NICKNAME <chr>,
## #   PLAYER_START_POSITION <chr>, PLAYER_COMMENT <chr>, PLAYER_MIN <time>,
## #   PLAYER_FGM <dbl>, PLAYER_FGA <dbl>, PLAYER_FG_PCT <dbl>, PLAYER_FG3M <dbl>,
## #   PLAYER_FG3A <dbl>, PLAYER_FG3_PCT <dbl>, PLAYER_FTM <dbl>,
## #   PLAYER_FTA <dbl>, PLAYER_FT_PCT <dbl>, PLAYER_OREB <dbl>,
## #   PLAYER_DREB <dbl>, PLAYER_REB <dbl>, PLAYER_AST <dbl>, PLAYER_STL <dbl>,
## #   PLAYER_BLK <dbl>, PLAYER_TO <dbl>, PLAYER_PF <dbl>, PLAYER_PTS <dbl>, …
```

```
powell_games <- player_winloss %>% filter(PLAYER_NAME == "Dwight Powell")
powell_games%>%
  group_by(WL)%>%
  summarize(PowellNum=n())
```

```
## # A tibble: 2 × 2
##   WL    PowellNum
##   <chr>     <int>
## 1 L            30
## 2 W            52
```

```
finneysmith_games <- player_winloss %>% filter(PLAYER_NAME == "Dorian Finney-Smith")
finneysmith_games %>%
  group_by(WL)%>%
  summarize(FinneySmithNum=n())
```

```
## # A tibble: 2 × 2
##   WL    FinneySmithNum
##   <chr>          <int>
## 1 L                 29
## 2 W                 52
```

```
brunson_games <- player_winloss %>% filter(PLAYER_NAME == "Jalen Brunson")
brunson_games %>%
  group_by(WL)%>%
  summarize(BrunsonNum=n())
```

```
## # A tibble: 2 × 2
##   WL    BrunsonNum
##   <chr>      <int>
## 1 L             29
## 2 W             50
```

We can see that out of the 82 games Dwight Powell played in, 52 of these games were winning games. It's also interesting to note that Dwight Powell played in every game this season. Out of the 81 games Dorian Finney-Smith played in, 52 of these games were winning games. Out of the 79 games that Jalen Brunson played in, 50 of these games were winning games.

```
doncic_games <- player_winloss %>% filter(PLAYER_NAME == "Luka Doncic")
doncic_games%>%
  group_by(WL)%>%
  summarize(DoncicNum=n())
```

```
## # A tibble: 2 × 2
##   WL    DoncicNum
##   <chr>     <int>
## 1 L            21
## 2 W            44
```

Out of the 65 games Doncic played in, 44 of these games were winning games. So, around 67.69% of games Doncic played in were winning games. This means that Doncic didn't play in 17 games. Now, let's look at how many of the games Doncic didn't play in were winning games.

```
anti_join(player_winloss, doncic_games, by = "GAME_ID") %>%
  filter(WL == "W") %>%
  group_by(GAME_ID)
```

```
## # A tibble: 103 × 29
## # Groups:   GAME_ID [8]
##    GAME_ID    WL    PLAYER_TEAM_ABBREV… PLAYER_TEAM_CITY PLAYER_ID PLAYER_NAME
##    <chr>      <chr> <chr>               <chr>                <dbl> <chr>
##  1 0022101092 W     DAL                 Dallas              203493 Reggie Bullo…
##  2 0022101092 W     DAL                 Dallas             1627827 Dorian Finne…
##  3 0022101092 W     DAL                 Dallas              203939 Dwight Powell
##  4 0022101092 W     DAL                 Dallas              203915 Spencer Dinw…
##  5 0022101092 W     DAL                 Dallas             1628973 Jalen Brunson
##  6 0022101092 W     DAL                 Dallas             1630182 Josh Green
##  7 0022101092 W     DAL                 Dallas             1628467 Maxi Kleber
##  8 0022101092 W     DAL                 Dallas             1628373 Frank Ntilik…
##  9 0022101092 W     DAL                 Dallas             1628425 Sterling Bro…
## 10 0022101092 W     DAL                 Dallas             1630589 Moses Wright
## # … with 93 more rows, and 23 more variables: PLAYER_NICKNAME <chr>,
## #   PLAYER_START_POSITION <chr>, PLAYER_COMMENT <chr>, PLAYER_MIN <time>,
## #   PLAYER_FGM <dbl>, PLAYER_FGA <dbl>, PLAYER_FG_PCT <dbl>, PLAYER_FG3M <dbl>,
## #   PLAYER_FG3A <dbl>, PLAYER_FG3_PCT <dbl>, PLAYER_FTM <dbl>,
## #   PLAYER_FTA <dbl>, PLAYER_FT_PCT <dbl>, PLAYER_OREB <dbl>,
## #   PLAYER_DREB <dbl>, PLAYER_REB <dbl>, PLAYER_AST <dbl>, PLAYER_STL <dbl>,
## #   PLAYER_BLK <dbl>, PLAYER_TO <dbl>, PLAYER_PF <dbl>, PLAYER_PTS <dbl>, …
```

Out of the 17 games that Doncic did not play in, only 8 of these were winning games. This is only around 47.06%.

What does this tell us?

Percent of games with Doncic: 65/82 or **79.27%**

Percent of games without Doncic: 17/82 or **20.73%**

Percent of game wins given Doncic playing: 44/65 or **67.69%**

Percent of game wins given Doncic NOT playing: 8/17 or **47.06%**

Total percent of game wins: 52/82 or **63.41%**

How I checked this:

W = Win

DP = Doncic Played

Law of total probability: $\mathbb{P}(W) = \mathbb{P}(W|DP)\mathbb{P}(DP) + \mathbb{P}(W|\neg DP)\mathbb{P}(\neg DP) = 0.6769 * 0.7927 + 0.4706 * 0.2073 \approx 0.6341$

Percent of game losses given Doncic playing: 21/65 or **32.31%**

Percent of game losses given Doncic NOT playing: 9/17 or **52.94%**

Total percent of game losses: **36.59%**

Question: How did Dwight Powell play in games without Doncic?

```
anti_join(powell_games, doncic_games, by = "GAME_ID") %>% group_by(GAME_ID) %>%
  filter(WL == "W")
```

```
## # A tibble: 8 × 29
## # Groups:   GAME_ID [8]
##   GAME_ID    WL    PLAYER_TEAM_ABBREVIATION PLAYER_TEAM_CITY PLAYER_ID PLAYER_NAME
##   <chr>      <chr> <chr>                    <chr>                <dbl> <chr>
## 1 0022101092 W     DAL                      Dallas              203939 Dwight Pow…
## 2 0022100954 W     DAL                      Dallas              203939 Dwight Pow…
## 3 0022100586 W     DAL                      Dallas              203939 Dwight Pow…
## 4 0022100531 W     DAL                      Dallas              203939 Dwight Pow…
## 5 0022100506 W     DAL                      Dallas              203939 Dwight Pow…
## 6 0022100468 W     DAL                      Dallas              203939 Dwight Pow…
## 7 0022100410 W     DAL                      Dallas              203939 Dwight Pow…
## 8 0022100400 W     DAL                      Dallas              203939 Dwight Pow…
## # … with 23 more variables: PLAYER_NICKNAME <chr>, PLAYER_START_POSITION <chr>,
## #   PLAYER_COMMENT <chr>, PLAYER_MIN <time>, PLAYER_FGM <dbl>,
## #   PLAYER_FGA <dbl>, PLAYER_FG_PCT <dbl>, PLAYER_FG3M <dbl>,
## #   PLAYER_FG3A <dbl>, PLAYER_FG3_PCT <dbl>, PLAYER_FTM <dbl>,
## #   PLAYER_FTA <dbl>, PLAYER_FT_PCT <dbl>, PLAYER_OREB <dbl>,
## #   PLAYER_DREB <dbl>, PLAYER_REB <dbl>, PLAYER_AST <dbl>, PLAYER_STL <dbl>,
## #   PLAYER_BLK <dbl>, PLAYER_TO <dbl>, PLAYER_PF <dbl>, PLAYER_PTS <dbl>, …
```

Powell played in 17 games that Doncic did not play in. Of these 17 games, 8 were winning games.

Therefore, Powell's winning percentage is around 47.06% without Doncic, and 67.69% with Doncic.

Question: How did Jalen Brunson play in games with Doncic?

Now, I'm interested in looking at games that both Jalen Brunson and Doncic played in together, since both played in over half of the games and are top scorers.

```
doncic_brunson <- inner_join(doncic_games, brunson_games, by = "GAME_ID")
doncic_brunson
```

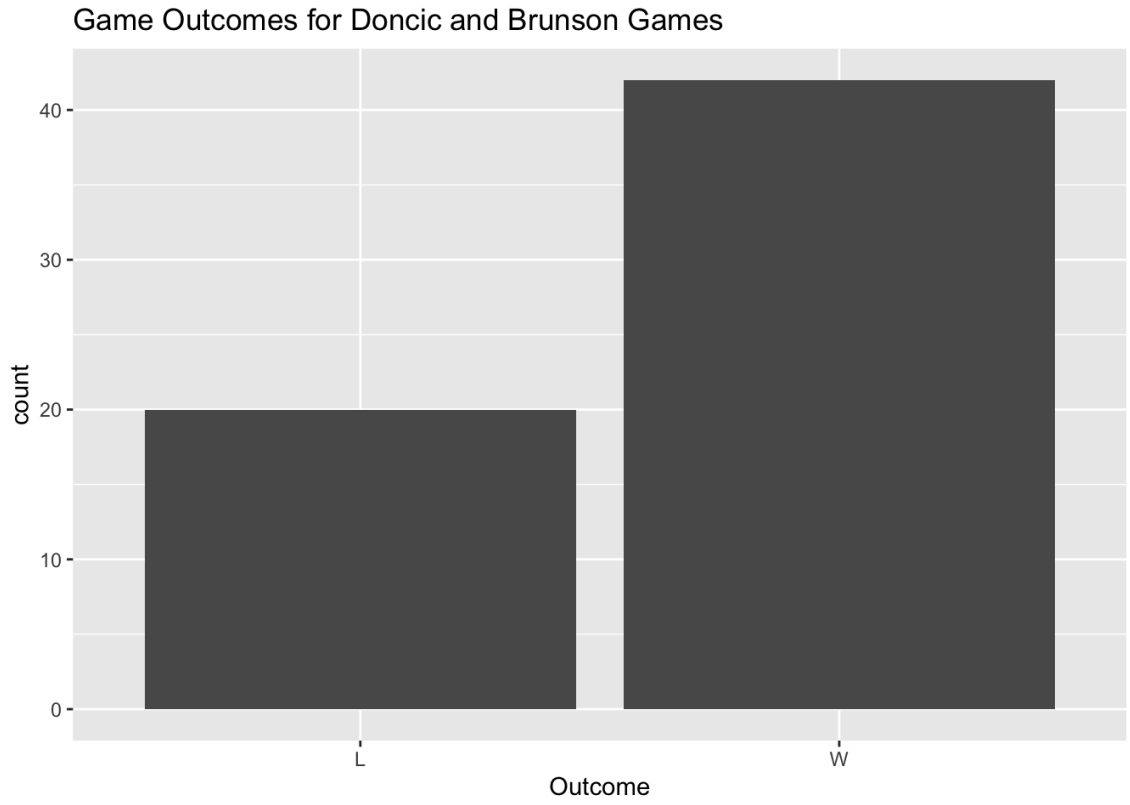
```
## # A tibble: 62 × 57
##    GAME_ID    WL.x  PLAYER_TEAM_ABBR… PLAYER_TEAM_CIT… PLAYER_ID.x PLAYER_NAME.x
##    <chr>      <chr> <chr>             <chr>                  <dbl> <chr>
##  1 0022101219 W     DAL               Dallas               1629029 Luka Doncic
##  2 0022101209 W     DAL               Dallas               1629029 Luka Doncic
##  3 0022101190 W     DAL               Dallas               1629029 Luka Doncic
##  4 0022101167 W     DAL               Dallas               1629029 Luka Doncic
##  5 0022101152 L     DAL               Dallas               1629029 Luka Doncic
##  6 0022101136 W     DAL               Dallas               1629029 Luka Doncic
##  7 0022101134 W     DAL               Dallas               1629029 Luka Doncic
##  8 0022101121 W     DAL               Dallas               1629029 Luka Doncic
##  9 0022101104 L     DAL               Dallas               1629029 Luka Doncic
## 10 0022101079 W     DAL               Dallas               1629029 Luka Doncic
## # … with 52 more rows, and 51 more variables: PLAYER_NICKNAME.x <chr>,
## #   PLAYER_START_POSITION.x <chr>, PLAYER_COMMENT.x <chr>, PLAYER_MIN.x <time>,
## #   PLAYER_FGM.x <dbl>, PLAYER_FGA.x <dbl>, PLAYER_FG_PCT.x <dbl>,
## #   PLAYER_FG3M.x <dbl>, PLAYER_FG3A.x <dbl>, PLAYER_FG3_PCT.x <dbl>,
## #   PLAYER_FTM.x <dbl>, PLAYER_FTA.x <dbl>, PLAYER_FT_PCT.x <dbl>,
## #   PLAYER_OREB.x <dbl>, PLAYER_DREB.x <dbl>, PLAYER_REB.x <dbl>,
## #   PLAYER_AST.x <dbl>, PLAYER_STL.x <dbl>, PLAYER_BLK.x <dbl>, …
```

There were 62 games that these two players both played in.

```
doncic_brunson %>% filter(WL.x == "W")
```

```
## # A tibble: 42 × 57
##    GAME_ID    WL.x  PLAYER_TEAM_ABBR… PLAYER_TEAM_CIT… PLAYER_ID.x PLAYER_NAME.x
##    <chr>      <chr> <chr>             <chr>                  <dbl> <chr>
##  1 0022101219 W     DAL               Dallas               1629029 Luka Doncic
##  2 0022101209 W     DAL               Dallas               1629029 Luka Doncic
##  3 0022101190 W     DAL               Dallas               1629029 Luka Doncic
##  4 0022101167 W     DAL               Dallas               1629029 Luka Doncic
##  5 0022101136 W     DAL               Dallas               1629029 Luka Doncic
##  6 0022101134 W     DAL               Dallas               1629029 Luka Doncic
##  7 0022101121 W     DAL               Dallas               1629029 Luka Doncic
##  8 0022101079 W     DAL               Dallas               1629029 Luka Doncic
##  9 0022101036 W     DAL               Dallas               1629029 Luka Doncic
## 10 0022101014 W     DAL               Dallas               1629029 Luka Doncic
## # … with 32 more rows, and 51 more variables: PLAYER_NICKNAME.x <chr>,
## #   PLAYER_START_POSITION.x <chr>, PLAYER_COMMENT.x <chr>, PLAYER_MIN.x <time>,
## #   PLAYER_FGM.x <dbl>, PLAYER_FGA.x <dbl>, PLAYER_FG_PCT.x <dbl>,
## #   PLAYER_FG3M.x <dbl>, PLAYER_FG3A.x <dbl>, PLAYER_FG3_PCT.x <dbl>,
## #   PLAYER_FTM.x <dbl>, PLAYER_FTA.x <dbl>, PLAYER_FT_PCT.x <dbl>,
## #   PLAYER_OREB.x <dbl>, PLAYER_DREB.x <dbl>, PLAYER_REB.x <dbl>,
## #   PLAYER_AST.x <dbl>, PLAYER_STL.x <dbl>, PLAYER_BLK.x <dbl>, …
```

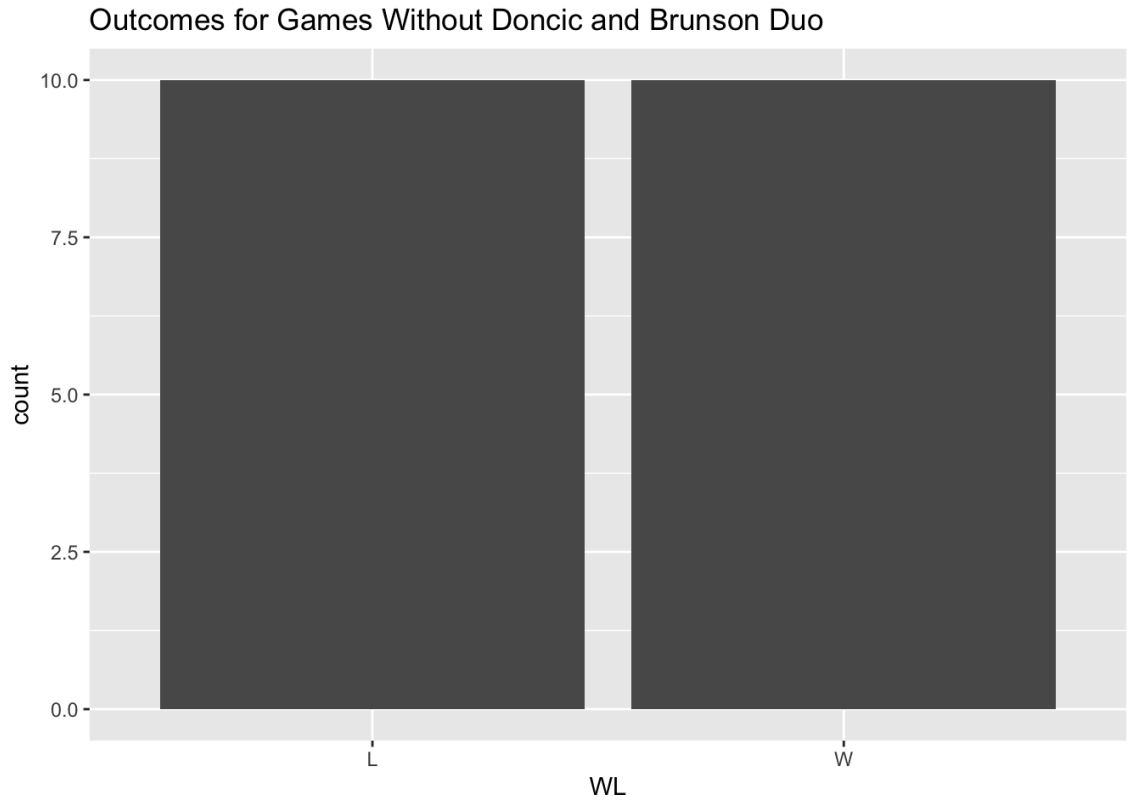
```
doncic_brunson %>% ggplot() +
  geom_bar(aes(x= WL.x)) +
  xlab("Outcome") +
  ggtitle("Game Outcomes for Doncic and Brunson Games")
```



Out of these 62 games that these two players played in, 42 of them were winning games. This means that when these two players played together, 67.74% of them were winning games.

Now, let's look at the games that Doncic and Brunson were not in together.

```
anti_join(player_winloss, doncic_brunson, by="GAME_ID") %>% group_by(GAME_ID) %>% summarize(WL) %>% disti
nct() %>%
  ggplot() +
  geom_bar(aes(x=WL)) +
  ggtitle("Outcomes for Games Without Doncic and Brunson Duo")
```



Out of the 82 games total, there were 20 games that both Doncic and Brunson did not play in. Out of these 20 games, only 10 were winning games. Therefore, without both Doncic and Brunson, 50% of the games the Mavs played were losing games.

Question: How did the team do when the top five players who played the most amount of minutes played in the same game?

Top 5 players: Dorian Finney-Smith, Jalen Brunson, Luka Doncic, Reggie Bullock, Dwight Powell

Because Powell played in all of the games, I won't be using him for the data analysis process.

```
bullock_games <- player_winloss %>% filter(PLAYER_NAME == "Reggie Bullock")
bullock_finney_games<- semi_join(bullock_games, finneysmith_games, by = "GAME_ID")

top_five <- semi_join(doncic_brunson, bullock_finney_games, by = "GAME_ID") %>%
  group_by(GAME_ID)
top_five # Looking at games the top five/six players played in together
```

```
## # A tibble: 54 × 57
## # Groups:   GAME_ID [54]
##    GAME_ID    WL.x  PLAYER_TEAM_ABBR… PLAYER_TEAM_CIT… PLAYER_ID.x PLAYER_NAME.x
##    <chr>      <chr> <chr>             <chr>                  <dbl> <chr>
##  1 0022101219 W     DAL               Dallas               1629029 Luka Doncic
##  2 0022101209 W     DAL               Dallas               1629029 Luka Doncic
##  3 0022101190 W     DAL               Dallas               1629029 Luka Doncic
##  4 0022101167 W     DAL               Dallas               1629029 Luka Doncic
##  5 0022101152 L     DAL               Dallas               1629029 Luka Doncic
##  6 0022101136 W     DAL               Dallas               1629029 Luka Doncic
##  7 0022101134 W     DAL               Dallas               1629029 Luka Doncic
##  8 0022101121 W     DAL               Dallas               1629029 Luka Doncic
##  9 0022101104 L     DAL               Dallas               1629029 Luka Doncic
## 10 0022101079 W     DAL               Dallas               1629029 Luka Doncic
## # … with 44 more rows, and 51 more variables: PLAYER_NICKNAME.x <chr>,
## #   PLAYER_START_POSITION.x <chr>, PLAYER_COMMENT.x <chr>, PLAYER_MIN.x <time>,
## #   PLAYER_FGM.x <dbl>, PLAYER_FGA.x <dbl>, PLAYER_FG_PCT.x <dbl>,
## #   PLAYER_FG3M.x <dbl>, PLAYER_FG3A.x <dbl>, PLAYER_FG3_PCT.x <dbl>,
## #   PLAYER_FTM.x <dbl>, PLAYER_FTA.x <dbl>, PLAYER_FT_PCT.x <dbl>,
## #   PLAYER_OREB.x <dbl>, PLAYER_DREB.x <dbl>, PLAYER_REB.x <dbl>,
## #   PLAYER_AST.x <dbl>, PLAYER_STL.x <dbl>, PLAYER_BLK.x <dbl>, …
```

```
anti_join(player_winloss, top_five, by="GAME_ID")  %>%
  filter(WL == "W") %>%
  group_by(GAME_ID, WL) %>%
  summarize()
```

```
## # A tibble: 16 × 2
## # Groups:   GAME_ID [16]
##    GAME_ID    WL
##    <chr>      <chr>
##  1 0022100266 W
##  2 0022100400 W
##  3 0022100410 W
##  4 0022100468 W
##  5 0022100506 W
##  6 0022100531 W
##  7 0022100586 W
##  8 0022100663 W
##  9 0022100868 W
## 10 0022100886 W
## 11 0022100954 W
## 12 0022100972 W
## 13 0022100999 W
## 14 0022101014 W
## 15 0022101036 W
## 16 0022101092 W
```

These five players played in 54 games together. Out of these 54 games, they won 36 games (~66.67%). There are 28 games the five players didn't play in together. 16/28 were winning games. (~57.14%)

Conclusion for Question 3:

There are definitely some notable advantages for the Mavericks when certain players are playing. For example, in games that Doncic and Brunson played together, ~67.74% of them were winning games. In games that Doncic and Brunson did not play in together, ~50% of them were losing games. We can also see that Doncic does have some sort of impact on the team's game outcome.

How could this impact coaching decision?

It's beneficial to have players with the most minutes played to continue playing because they bring the team an advantage. Knowing what each player brings to the table is important because coaches can choose the right substitute for if one of the top players gets injured.

A good pairing would be Doncic and Brunson, who could help lead the team to a win. But, the limitations of the analysis I did was that I did not analyze players who didn't have both the most minutes and most games played. If a star player like Doncic becomes injured, then the statistics for other players could look different without him which is something I didn't fully analyze. This could be something I do when I expand upon this project.

Limitations

From my analysis, I don't know if defensive rebounds can be applicable to all of the NBA teams as an indicator of measuring wins. The data I looked at pertained only to the Dallas Mavericks and only included the 2021-2022 regular season. I could analyze defensive rebounds for all NBA teams and compare their defensive rebounds to that of their opponents if I wanted a more generalizable model. Furthermore, it doesn't note when players gets traded (such as Porzingis) or injured. There could also be other reasons why some players don't show up as much in this dataset that aren't explicitly obvious while looking at this data.

There are also many many more factors that influence a player's value to their team such as their points per possession, their effective field goal percent, and their defensive style. These are all things that I want to look at more with more time.

Sources/Websites Used:

Basketball Terms Glossary: https://www.basketball-reference.com/about/glossary.html# (https://www.basketball-reference.com/about/glossary.html#):~:text=Poss%20%2D%20Possessions%20(available%20since%20the,FG)%20%2B%20Opp%20TOV

Swar's Github with the NBA API: https://github.com/swar/nba_api/blob/master/docs/nba_api/stats/examples.md (https://github.com/swar/nba_api/blob/master/docs/nba_api/stats/examples.md)

NBA Website: https://www.nba.com/ (https://www.nba.com/)

Stat Quest: https://www.youtube.com/watch?v=nk2CQITm_eo (https://www.youtube.com/watch?v=nk2CQITm_eo)

More Stat Quest: https://www.youtube.com/watch?v=zITIFTsivN8&t=79s (https://www.youtube.com/watch?v=zITIFTsivN8&t=79s)

And more Stat Quest: https://www.youtube.com/watch?v=C4N3_XJJ-jU&t=126s (https://www.youtube.com/watch?v=C4N3_XJJ-jU&t=126s)

Future Expansions of Project

Look at other factors we need to take into consideration for playing strategy that might not be explicitly reported as stats. How can we use data to measure discrepancy between individual player stats and team performance? For example, 2012 OKC had players with extremely impressive personal stats: Westbrook, Harden, Durant, yet still weren't able to win a ring. Create more of a comparative analysis rather than focusing on one team.

Create an NBA shot chart to visualize how players shot at each spot on the field. Do players have a weaker side on the court? Are there specific regions certain players are better at shooting from like the wings or the three point line?

Look for public data about player sentiment towards their coaches or their teammates. Do teams in which the players get along better perform stronger than teams where the players don't feel as strongly about their teammates/coaches?