```
---
title: "nba_project_rough"
output: html_document
date: "2025-05-28"
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

#Six Datasets: Lebron's Regular Season Stats, Lebron's Playoff Stats, Lebron's Play-By-Play Projections, Jordan's Regular Season Stats, Jordan's Playoff Stats, Jordan's Play-By-Play Projections. (note: we excluded shooting stats for Jordan because it is only two rows, but feel free to reference when comparing against Lebron). (note: Lebron's datsets have missing values, while Jordan's does not.)
```{r}
library(ggplot2)
```


A big part of Lebron's case is his longevity. We are going to plot both players total production throughout their careers to show why people argue for Lebron. We are going to add points, rebounds, assists, steals, and blocks.


```{r error = TRUE}


lebron_regular_season_master$Total_per_game <- lebron_regular_season_master$PTS +
                                               lebron_regular_season_master$TRB +
                                               lebron_regular_season_master$AST +
                                               lebron_regular_season_master$STL +
                                               lebron_regular_season_master$BLK

lebron_regular_season_master$Season <- as.factor(lebron_regular_season_master$Season)



ggplot(lebron_regular_season_master, aes(x = Season, y = Total_per_game, group = 1, color
= "red")) +
  geom_line(linewidth = 1) +
  labs(title = "LeBron's Per Game Stats Over Seasons",
       y = "Sum of PTS + TRB + AST + STL + BLK",
       x = "Season") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

```
```{r error = TRUE}
lebron_regular_season_master$Career_Year <- 1:nrow(lebron_regular_season_master)
jordan_master_stats$Career_Year <- 1:nrow(jordan_master_stats)


```

```{r error = TRUE}
lebron_clean <-
lebron_regular_season_master[!is.na(lebron_regular_season_master$Total_per_game), ]
jordan_clean <- jordan_master_stats[!is.na(jordan_master_stats$Total_per_game), ]


lebron_clean$Career_Year <- as.numeric(lebron_clean$Career_Year)
jordan_clean$Career_Year <- as.numeric(jordan_clean$Career_Year)
```
```

```
ggplot() +
  geom_line(data = lebron_clean,
            aes(x = Career_Year, y = Total_per_game, color = "LeBron"),
            linewidth = 1) +
  geom_line(data = jordan_clean,
            aes(x = Career_Year, y = Total_per_game, color = "Jordan"),
            linewidth = 1) +
  labs(title = "LeBron vs. Jordan: Sum of Per Game Stats Over Career Years",
       y = "PTS + TRB + AST + STL + BLK",
       x = "Career Year",
       color = "Player") +
  theme_minimal()
```

Graph conclusion: Jordan's flame burned extremely bright, but Lebron's flame burned 95% as bright but lasted a lot longer. This is one of the main arguments for Lebron

A big argument for Jordan, besides all of his championships and awards, was his immaculate scoring in his prime. Lets plot both players scoring for years 3-9. (I know Lebron's prime was much longer, but we are doing this to compare)

```{r error = TRUE}
lebron_prime <- lebron_regular_season_master[3:9, ]

jordan_prime <- jordan_master_stats[3:9, ]

```
```{r error = TRUE}

lebron_prime$Career_Year <- 3:9
jordan_prime$Career_Year <- 3:9

lebron_prime$PTS <- as.numeric(lebron_prime$PTS)
jordan_prime$PTS_pergame <- as.numeric(jordan_prime$PTS_pergame)


ggplot() +
  geom_line(data = lebron_prime,
            aes(x = Career_Year, y = PTS, color = "LeBron"),
            linewidth = 1) +

  geom_line(data = jordan_prime,
            aes(x = Career_Year, y = PTS_pergame, color = "Jordan"),
            linewidth = 1) +

  labs(title = "LeBron vs. Jordan Early Prime: Points Per Game (Years 3-9)",
       y = "Points Per Game",
       x = "Career Year (Early Prime)",
       color = "Player") +

  theme_minimal()
```
This clearly highlights just how much better of a scorer Jordan was over Lebron in the early prime. Lebron exploded onto the scene, but Jordan was even better in this stat.

```{r error = TRUE}
jordan_master_stats_clean <- jordan_master_stats[1:(nrow(jordan_master_stats) - 5), ]
```

```

```{r}
library(ggplot2)
library(dplyr)

plot_multiple_stats <- function(lebron_data, jordan_data, stat_list) {

  lebron_data$Career_Year <- 1:nrow(lebron_data)
  jordan_data$Career_Year <- 1:nrow(jordan_data)

  combined_list <- list()

  for (stat in stat_list) {

    lebron_stat <- lebron_data %>%
      filter(.data[[stat]] != "baseball") %>%
      mutate(Value = as.numeric(.data[[stat]]),
             Stat = stat,
             Player = "LeBron") %>%
      select(Career_Year, Stat, Player, Value)

    jordan_stat <- jordan_data %>%
      filter(.data[[stat]] != "baseball") %>%
      mutate(Value = as.numeric(.data[[stat]]),
             Stat = stat,
             Player = "Jordan") %>%
      select(Career_Year, Stat, Player, Value)

    combined_list[[stat]] <- rbind(lebron_stat, jordan_stat)
  }

  combined_data <- bind_rows(combined_list)

  ggplot(combined_data, aes(x = Career_Year, y = Value, color = Player)) +
    geom_line(linewidth = 1) +
    facet_wrap(~ Stat, scales = "free_y") +
    labs(title = "LeBron vs. Jordan: Stat Comparisons Over Career",
         x = "Career Year",
         y = "Value",
         color = "Player") +
    theme_minimal() +
    theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank())
}

```


```{r error = TRUE}
plot_multiple_stats(lebron_regular_season_master,
                    jordan_master_stats_clean,
                    c("TS%", "ORtg", "DRtg", "USG%"))
```
USG%: estimate of teams plays used by the player while they were on the floor. The graph shows how ready the bulls were for Jordan, as they made him the franchise player right after he was drafted. Lebron "carried" the Cavs too, but Jordan was given the keys. After early playoff failure by the Bulls, his usage rate dropped as they focused on adding key pieces to build the team. Around Lebron's Miami Heat years, he is at or above Jordan in USG%. Overall Jordan had more plays drawn for him as he was a dominant scorer. This shows Lebron's exellence in producing in all ways including scoring, while having a generally lower USG% than Jordan. Jordan's ability to stay efficient while having this high USG% shows is greatness, and Lebron being able to produce almost as much as Jordan with lower
```

USG% shows his greatness.

TS%: measures shooting efficiency while taking in 2P,3P, and FT. Lebron clearly wins here. He entered the league straight out of high school and that gives him more NBA experience, but a bigger pressure to adjust to the NBA game. right after about 6 years deep in the league, his TS% soared over Jordan's. But I will give credit to Jordan, as his TS% was very high right when he entered the League. His TS% was much lower than Lebron's during his dynasty, as the peaks of his TS% were during the Bull's championship building and hardships.

DRtg: estimate of points allowed per 100 possessions: Jordan generally had a higher defensive rating than Lebron during his prime. He excelled in quickness and deflections, leading to great on-ball defense and collected many steals. What is unique about this graph is that Lebron's DRtg improved durastically as he got older, this is suprising because getting older leads to being slower, but the eye test showed that he got smarter and more dedicated to defense as he matured. Its hard to tell who wins here.

ORtg: estimate of points produced per 100 possessions: Jordan was generally better here too. But during his first 3 championships, Jordan's ORtg was lower than Lebron's at that point in their respective careers. But during the last 3 championships, Jordan's ORtg was higher than Lebron's. I would say Jordan gets the edge here, but its close and Lebron's longevity is impressive.

```{r error = TRUE}
plot_multiple_stats(lebron_regular_season_master,
                    jordan_master_stats_clean,
                    c("PER", "BPM", "OBPM", "DBPM"))

```
BPM: box score estimate of the points per 100 possessions a player contributed above a league average player, translated to an average team. This one is so similar, with both players rising and falling below each other. Lebron's longevity is impressive here, showing is impact even late in his career, around 6 more points contributed per 100 possessions in 2024-2025. Jordan's early peak shows here, with his BPM soaring to complement his raw production. Lebron's max BPM is higher than Jordan's max BPM. But during the early primes, Jordan's was generally better, but during the middle primes, they alternated and were similar. Inconclusive here.

DBPM: box score estimate of the defensive points per 100 possessions a player contributed above a league average player. inconclusive here. some observations - Jordans max DBPM was higher, besides that they were so similar and need context, nuance, and qualitative things to bring substance

OBPM: box score estimate of the offensive points per 100 possessions a player contributed above a league average player. Its close, and these players had very different tragjectories in their career, but Lebron wins here. Jordan was higher during the early parts of his career, but Lebron figured it out and had a higher peak and better valleys than Jordan.

PER: player efficiency rating - per minute production standardized to the league average of 15.Lebron wins here. His PER in his very late stages of his career was better than Jordan's late 90s runs, and early 90s runs. He had two peaks which were as high as Jordans two peaks. But Jordan wins the early prime once again, but during these years, its key to point out that Lebron had much better playoff success during the early prime over Jordan.

```{r error = TRUE}
```

```
plot_multiple_stats(lebron_regular_season_master,
                    jordan_master_stats_clean,
                    "VORP")

```
VORP: Value over Replacement Player
A box score estimate of the points per 100 TEAM possessions that a player contributed above a replacement-level (-2.0) player, translated to an average team and prorated to an 82-game season.

Multiply by 2.70 to convert to wins over replacement.

I think Jordan wins here visually. Higher peak and Generally better during his prime, despite a low drop during the early 90s. Lebron rises above him some years but Jordan wins numerically visually. Context: The average player in the modern NBA is much better than the average player in Jordan's era.


```{r}
plot_multiple_stats(lebron_regular_season_master,
                    jordan_master_stats_clean,
                    "WS")
```
WS: estimate of number of wins contributed by a player.Its hard to tell with Lebron's longevity and Jordans major drop off during early 90s. along with the switing early on. If I had to guess based on higher highlights, I'd say Jordan wins here.


```{r}
plot_multiple_stats(lebron_regular_season_master,
                    jordan_master_stats_clean,
                    "WS/48")
```
WS/48: estimate of win shares by a player per 48 minutes, Jordan dominates early prime, but In their respective peaks, LeBron wins here. Highest peak and stays relatively above him throught important years. adding longevity, Lebron clears. He had higher WS/48 his years with the lakers which were meant to be his retirement send off on par with Jordans explosion early years and his early 90s years.


```{r}
jordan_master_stats
```

```{r}
lebron_prime_only <- lebron_regular_season_master[3:(nrow(lebron_regular_season_master) - 3), ]
lebron_prime_only
```

```{r}
jordan_prime_only <- jordan_master_stats[3:(nrow(jordan_master_stats) - 5), ]
jordan_prime_only
```



```{r}
lebron_prime_numeric <- lebron_prime_only %>%
  mutate(across(everything(), ~ as.numeric(.)))

jordan_prime_numeric <- jordan_prime_only %>%
  mutate(across(everything(), ~ as.numeric(.)))
```

```
lebron_prime_avg <- colMeans(lebron_prime_numeric, na.rm = TRUE)
jordan_prime_avg <- colMeans(jordan_prime_numeric, na.rm = TRUE)

lebron_avg_row <- as.data.frame(t(lebron_prime_avg))
jordan_avg_row <- as.data.frame(t(jordan_prime_avg))

box_stat_pairs <- list(
  "PTS" = "PTS-pergame",
  "AST" = "AST-pergame",
  "TRB" = "TRB-pergame",
  "STL" = "STL-pergame",
  "BLK" = "BLK-pergame",
  "FG%" = "FG%",
  "eFG%" = "eFG%",
  "TS%" = "TS%",
  "TOV" = "TOV"
)

adv_stat_pairs <- list(
  "ORtg" = "ORtg",
  "DRtg" = "DRtg",
  "PER" = "PER",


  "WS" = "WS",

  "OBPM" = "OBPM",
  "DBPM" = "DBPM",
  "BPM" = "BPM",
  "VORP" = "VORP",

)

lebron_score <- 0
jordan_score <- 0

for (pair in names(box_stat_pairs)) {
  lebron_col <- pair
  jordan_col <- box_stat_pairs[[pair]]

  if (lebron_col %in% names(lebron_avg_row) && jordan_col %in% names(jordan_avg_row)) {
    lebron_val <- as.numeric(lebron_avg_row[[lebron_col]])
    jordan_val <- as.numeric(jordan_avg_row[[jordan_col]])

    if (all(!is.na(c(lebron_val, jordan_val)))) {
      if (pair == "TOV") {
        if (lebron_val < jordan_val) {
          lebron_score <- lebron_score + 4
        } else if (jordan_val < lebron_val) {
          jordan_score <- jordan_score + 4
        }
      } else {
        if (lebron_val > jordan_val) {
          lebron_score <- lebron_score + 4
        } else if (jordan_val > lebron_val) {
          jordan_score <- jordan_score + 4
        }
      }
    }
  }
}

for (pair in names(adv_stat_pairs)) {
  lebron_col <- pair
```

```
  jordan_col <- adv_stat_pairs[[pair]]

  if (lebron_col %in% names(lebron_avg_row) && jordan_col %in% names(jordan_avg_row)) {
    lebron_val <- as.numeric(lebron_avg_row[[lebron_col]])
    jordan_val <- as.numeric(jordan_avg_row[[jordan_col]])

    if (all(!is.na(c(lebron_val, jordan_val)))) {
      if (lebron_val > jordan_val) {
        lebron_score <- lebron_score + 3
      } else if (jordan_val > lebron_val) {
        jordan_score <- jordan_score + 3
      }
    }
  }
}


```
LeBron's Total Weighted Score: 22
Jordan's Total Weighted Score: 24

I used a weighted scoring system to evaluate and compare each player. I used a mix of advanced stats and regular box score stats and weighted regular box score stats slightly more because they are used more in comparisions. I still used many advanced stats because I believe they aid great discovery. I tried to make the system as unbiased as possible, by not looking at the data to pick and choose stats that favor Lebron (my pick). I simply used stats thatI believe tell the best story in a player's ability.

Next I am going to perform a hypothesis testing procedure.
```{r}
lebron_obpm <- as.numeric(lebron_prime_only$OBPM)
jordan_obpm <- as.numeric(jordan_prime_only$OBPM)


lebron_obpm <- lebron_obpm[!is.na(lebron_obpm)]
jordan_obpm <- jordan_obpm[!is.na(jordan_obpm)]


lebron_avg_obpm <- mean(lebron_obpm)
jordan_avg_obpm <- mean(jordan_obpm)
```


I am going to use a permutation testing procedure to perform a hypothesis test on whether the difference between Jordan's prime average OBPM is statistically significantly better than LeBron's prime average OBPM. I am doing this because these stats are close numerically, and Jordan is known as the better offensive player because of his scoring numbers, and it shows in the numbers. But lets see if he truly is better in this statistic, or if it is just due to random chance.

LeBron's Prime Average OBPM: 7.317647
Jordan's Prime Average OBPM: 7.736364

Null hypothesis: Jordan's mean OBPM is less than or equal to Lebron's mean OBPM

Alternative hypothesis: Jordan's mean OBPM is greater than Lebron's mean OBPM


```{r}
set.seed(123)
```

```
lebron_obpm <- as.numeric(lebron_prime_only$OBPM)
jordan_obpm <- as.numeric(jordan_prime_only$OBPM)

lebron_obpm <- lebron_obpm[!is.na(lebron_obpm)]
jordan_obpm <- jordan_obpm[!is.na(jordan_obpm)]

obs_diff <- mean(jordan_obpm) - mean(lebron_obpm)

combined <- c(jordan_obpm, lebron_obpm)
n_jordan <- length(jordan_obpm)
n_perm <- 10000
perm_diffs <- numeric(n_perm)

for (i in 1:n_perm) {
  shuffled <- sample(combined)
  jordan_sample <- shuffled[1:n_jordan]
  lebron_sample <- shuffled[(n_jordan + 1):length(shuffled)]

  perm_diffs[i] <- mean(jordan_sample) - mean(lebron_sample)
}

p_value <- mean(perm_diffs >= obs_diff)



obs_diff
p_value
```


```{r}
hist(perm_diffs, breaks = 30, main = "Permutation Test: OBPM Mean Difference",
     xlab = "Mean Difference", col = "skyblue")
abline(v = obs_diff, col = "red", lwd = 2, lty = 2)

```
the observed difference is 0.4187
the p_value is 0.2287.

conclusion: with a limited sample size, different eras, and enough natural year-to-year variation, there is not evidence to reject the null. The p value is not statistically significant enough to claim that Jordans average OBPM in his prime was higher than Lebron's.


```{r}
lebron_playoffs_master
```
```{r}
jordan_master_playoff_stats
```



```{r}
lebron_playoffs_master$Career_Year <- 1:nrow(lebron_playoffs_master)
jordan_master_playoff_stats$Career_Year <- 1:nrow(jordan_master_playoff_stats)

jordan_clean <- jordan_master_playoff_stats[
  jordan_master_playoff_stats$PTS_pergame != "baseball" &
  !is.na(jordan_master_playoff_stats$PTS_pergame) &
  !is.na(jordan_master_playoff_stats$AST_pergame) &
```

```
  !is.na(jordan_master_playoff_stats$TRB_pergame) &
  !is.na(jordan_master_playoff_stats$STL_pergame) &
  !is.na(jordan_master_playoff_stats$BLK_pergame),
]

lebron_clean <- lebron_playoffs_master[
  !is.na(lebron_playoffs_master$PTS) &
  !is.na(lebron_playoffs_master$AST) &
  !is.na(lebron_playoffs_master$TRB) &
  !is.na(lebron_playoffs_master$STL) &
  !is.na(lebron_playoffs_master$BLK),
]

lebron_clean$Career_Year <- as.numeric(lebron_clean$Career_Year)
jordan_clean$Career_Year <- as.numeric(jordan_clean$Career_Year)

lebron_clean$Total_per_game <- lebron_clean$PTS + lebron_clean$AST + lebron_clean$TRB +
lebron_clean$STL + lebron_clean$BLK
jordan_clean$Total_per_game <- jordan_clean$PTS_pergame + jordan_clean$AST_pergame +
jordan_clean$TRB_pergame + jordan_clean$STL_pergame + jordan_clean$BLK_pergame


ggplot() +
  geom_line(data = lebron_clean,
            aes(x = Career_Year, y = Total_per_game, color = "LeBron"),
            linewidth = 1) +
  geom_line(data = jordan_clean,
            aes(x = Career_Year, y = Total_per_game, color = "Jordan"),
            linewidth = 1) +
  labs(title = "LeBron vs. Jordan: Sum of Per Game Playoff Stats Over Career Years",
       y = "PTS + TRB + AST + STL + BLK",
       x = "Career Year",
       color = "Player") +
  theme_minimal()

```
In their first years in the playoffs, Lebron (3rd season) had very similar totals to Jordan (rookie). Jordan mostly is over Lebron for the first around 9 playoff runs, with Lebron sneaking above Jordan for one of them. From 10 >, Lebron was over Jordan.

```{r}
library(ggplot2)
library(dplyr)

plot_multiple_stats <- function(lebron_data, jordan_data, stat_list) {

  lebron_data$Career_Year <- 1:nrow(lebron_data)
  jordan_data$Career_Year <- 1:nrow(jordan_data)

  combined_list <- list()

  for (stat in stat_list) {

    lebron_stat <- lebron_data %>%
      filter(.data[[stat]] != "baseball" & !is.na(.data[[stat]])) %>%
      mutate(Value = as.numeric(.data[[stat]]),
             Stat = stat,
             Player = "LeBron") %>%
      select(Career_Year, Stat, Player, Value)


    jordan_stat <- jordan_data %>%
      filter(.data[[stat]] != "baseball" & !is.na(.data[[stat]])) %>%
      mutate(Value = as.numeric(.data[[stat]]),
```

```
            Stat = stat,
            Player = "Jordan") %>%
      select(Career_Year, Stat, Player, Value)

    combined_list[[stat]] <- rbind(lebron_stat, jordan_stat)
  }

  combined_data <- bind_rows(combined_list)


  ggplot(combined_data, aes(x = Career_Year, y = Value, color = Player)) +
    geom_line(linewidth = 1) +
    facet_wrap(~ Stat, scales = "free_y") +
    labs(title = "LeBron vs. Jordan Playoff Stat Comparisons Over Career",
         x = "Career Year",
         y = "Value",
         color = "Player") +
    theme_minimal()
}

```


```{r}
plot_multiple_stats(lebron_playoffs_master,
                    jordan_master_playoff_stats,
                    c("TS%", "ORtg", "DRtg", "USG%"))

```


DRtg: Jordan wins overall here, especially early on in his career. but Lebron's longevity shows his improvement later in his career on the defensive end.

ORtg: Lebron has the highest peak here and the lowest valley. Between Jordans later seasons of his prime, Lebron was all over him. Overall very up and down between both and hard to pick one or the other visually

TS%: Lebron wins here clearly. highest peak and higher overall and longevity

USG%: Jordan is higher than Lebron here on almost every playoff run they share throughout their careers.


```{r}
plot_multiple_stats(lebron_playoffs_master,
                    jordan_master_playoff_stats,
                    c("PER", "BPM", "OBPM", "DBPM"))

```

BPM: suprisingly, Lebron dominates Jordan in BPM for the early prime, something we have not seen. Jordan is above him throughout the rest of the early prime, then its relatively even through the middle and late prime, with Lebron's ressurgence after MJ's retirement

DBPM: Lebron wins here

OBPM: Lebron has the highest peak by far and has the longevity, but Jordan is over Lebron in most of the graph and is more consistent. Jordan wins here

PER: Lebron wins here. Jordan is only above him in the vert beginning and a small part of the prime. Lebron has the highest peak here by far, stays with jordan throughout his 90s, and his second stint with the cavs had higher PER than late 90s dynasty Bulls.

```{r}
plot_multiple_stats(lebron_playoffs_master,
                    jordan_master_playoff_stats,
                    "VORP")

```

VORP: Lebron wins here: starts his playoff runs better, stays with Jordan throughout the prime, and Jordan had a deep drop during his prime, while lebrons drop was near then end of his career before the ressurgence.

```{r}
plot_multiple_stats(lebron_playoffs_master,
                    jordan_master_playoff_stats,
                    "WS")

```

WS: Lebron wins here. Starts higher, higher peak, Jordan had a steep drop off, and Lebron played longer


```{r}
plot_multiple_stats(lebron_playoffs_master,
                    jordan_master_playoff_stats,
                    "WS/48")

```
WS/48: Lebron wins here, higher peak, stays with Jordan for the most part during the primes, even though Jordan had highest peaks 2 and 3. Longevity again.

```{r}
lebron_numeric <- lebron_playoffs_master %>%
  slice(-((n() - 3):n())) %>%
  mutate(across(everything(), ~ as.numeric(.)))

jordan_numeric <- jordan_master_playoff_stats %>%
  slice(-c(1, 2, 3)) %>%
  mutate(across(everything(), ~ as.numeric(.)))

lebron_avg <- colMeans(lebron_numeric, na.rm = TRUE)
jordan_avg <- colMeans(jordan_numeric, na.rm = TRUE)

lebron_avg_row <- as.data.frame(t(lebron_avg))
jordan_avg_row <- as.data.frame(t(jordan_avg))

box_stat_pairs <- list(
  "PTS" = "PTS_pergame",
  "AST" = "AST_pergame",
  "TRB" = "TRB_pergame",
  "STL" = "STL_pergame",
  "BLK" = "BLK_pergame",
  "FG%" = "FG%",
  "eFG%" = "eFG%",
  "TS%" = "TS%",
  "TOV" = "TOV"
)

adv_stat_pairs <- list(
  "ORtg" = "ORtg",
  "DRtg" = "DRtg",
  "PER" = "PER",
  "WS" = "WS",
```

```
  "OBPM" = "OBPM",
  "DBPM" = "DBPM",
  "BPM" = "BPM",
  "VORP" = "VORP"
)


lebron_score <- 0
jordan_score <- 0

for (pair in names(box_stat_pairs)) {
  lebron_col <- pair
  jordan_col <- box_stat_pairs[[pair]]

  if (lebron_col %in% names(lebron_avg_row) & jordan_col %in% names(jordan_avg_row)) {
    lebron_val <- as.numeric(lebron_avg_row[[lebron_col]])
    jordan_val <- as.numeric(jordan_avg_row[[jordan_col]])

    if (all(!is.na(c(lebron_val, jordan_val)))) {
      if (pair == "TOV") {
        if (lebron_val < jordan_val) {
          lebron_score <- lebron_score + 4
        } else if (jordan_val < lebron_val) {
          jordan_score <- jordan_score + 4
        }
      } else {
        if (lebron_val > jordan_val) {
          lebron_score <- lebron_score + 4
        } else if (jordan_val > lebron_val) {
          jordan_score <- jordan_score + 4
        }
      }
    }
  }
}


for (pair in names(adv_stat_pairs)) {
  lebron_col <- pair
  jordan_col <- adv_stat_pairs[[pair]]

  if (lebron_col %in% names(lebron_avg_row) & jordan_col %in% names(jordan_avg_row)) {
    lebron_val <- as.numeric(lebron_avg_row[[lebron_col]])
    jordan_val <- as.numeric(jordan_avg_row[[jordan_col]])

    if (all(!is.na(c(lebron_val, jordan_val)))) {
      if (lebron_val > jordan_val) {
        lebron_score <- lebron_score + 3
      } else if (jordan_val > lebron_val) {
        jordan_score <- jordan_score + 3
      }
    }
  }
}

lebron_score
jordan_score

```


LeBron's Total Weighted Playoff Score: 37
Jordan's Total Weighted Playoff Score: 23

Although this weighted scoring system is not perfect, I made an effort to make it unbiased and correctly weigh these stats.

```{r}
wins <- c(
  35, 42, 50, 50, 45, 66, 61, 58, 46, 66,
  54, 53, 57, 51, 50, 37, 52, 42, 30, 43,
  47, 50
)

lebron_regular_model_master$Wins <- wins
```

```{r}
lebron_regular_model_master
```

```{r}
jordan_regular_model_master
```
```{r}
wins <- c(38, 30, 40, 50, 47, 55, 61, 67, 57, 72, 69, 62, 37, 37)

jordan_regular_model_master$Wins <- wins
```
```{r}
jordan_regular_model_master
```

Making a multiple regression model for Lebron James in the regular season:

```{r}
intercept_model <- lm(Wins ~ 1, data = lebron_regular_model_master)

full_model <- lm(Wins ~ PER + `eFG%` + `TS%` + TRB + AST + STL + BLK + `USG%` + `WS/48` + BPM + VORP + `eFG+` + `TS+` + PTS_per36 + AST_per36 + AST_per100 + PTS_per100 + DRtg + ORtg + PTS, data = lebron_regular_model_master)

stepwise_selection_model <- step(intercept_model, scope = formula(full_model), direction = "forward")

summary(stepwise_selection_model)

```
```{r}
lebron_regular_model_master
```
```{r}
selected_predictors <- lebron_regular_model_master[, c(
  "PER", "eFG%", "TS%", "TRB", "AST", "STL", "BLK",
  "USG%", "WS/48", "BPM", "VORP", "PTS_per100",
  "AST_per100", "PTS_per36", "AST_per36", "PTS",
  "DRtg", "ORtg"
)]

cor_matrix <- cor(selected_predictors, use = "complete.obs")
```

```
library(corrplot)
corrplot(cor_matrix, method = "color", tl.cex = 0.7, number.cex = 0.7)
```
notes:

dont use ast_pe36 and assists
ast_per100 and asssist
dont use assist per 100 and asssist per 36 (dont use multiple assists)

dont use efficienty percentages together

dont use stl and drating together

be cautious with USG% and points stats

WS/48 might not go along with PER, BPM, VORP

take 2:

```{r}
intercept_model2 <- lm(Wins ~ 1, data = lebron_regular_model_master)

full_model2 <- lm(Wins ~ PER + TRB + AST + STL + BLK + `USG%` + `WS/48` + BPM + VORP +
DRtg + ORtg + PTS, data = lebron_regular_model_master)

stepwise_selection_model2 <- step(intercept_model2, scope = formula(full_model2),
direction = "forward")

summary(stepwise_selection_model2)
```
```{r}
selected_predictors2 <- lebron_regular_model_master[, c(
  "PER", "TRB", "AST", "STL", "BLK", "USG%", "WS/48",
  "BPM", "VORP", "DRtg", "ORtg", "PTS"
)]

cor_matrix2 <- cor(selected_predictors2, use = "complete.obs")

corrplot(cor_matrix2, method = "color", tl.cex = 0.7, number.cex = 0.7)

```
-steals and DRtg cant go together

-WS/48, PER, BPM, VORP all correlated

take 3:

```{r}
intercept_model3 <- lm(Wins ~ 1, data = lebron_regular_model_master)

full_model3 <- lm(Wins ~ TRB + AST + STL + BLK + `USG%` + `WS/48` + ORtg + PTS, data =
lebron_regular_model_master)

stepwise_selection_model3 <- step(intercept_model3, scope = formula(full_model3),
direction = "forward")

summary(stepwise_selection_model3)
```
```{r}
lebron_stepwise_model3 <- lm(formula = Wins ~ `WS/48` + STL + TRB, data =
lebron_regular_model_master)
```

```
summary(lebron_stepwise_model3)


```
^^This is an intuitive model based on take three stepwise selection model. I deleted Points and Assists because they were not significant in the stepwise selection model. Multiple R^2 and adjusted R^2 shows that enough of the variation of the data is explained by the predictors, RSE is small. the F-statistic is significant, meaning that there is evidence at least one of the predictors does not have a coefficient of 0.

cross validation using the Leave Out One Cross Validation LOOCV method:

```{r}
formula <- Wins ~ `WS/48` + STL + TRB

n <- nrow(lebron_regular_model_master)

cv_results <- cv.glm(lebron_regular_model_master, glm(formula, data =
lebron_regular_model_master), K = n)

cat("LOOCV error:", cv_results$delta[1], "\n")

```


regularization:

```{r}
library(glmnet)
X <- model.matrix(Wins ~ `WS/48` + STL + TRB, data = lebron_regular_model_master)[, -1]
y <- lebron_regular_model_master$Wins
```

```{r}
ridge_cv <- cv.glmnet(X, y, alpha = 0)

best_lambda_ridge <- ridge_cv$lambda.min
cat("Best Ridge lambda:", best_lambda_ridge, "\n")

ridge_coefs <- coef(ridge_cv, s = best_lambda_ridge)
print(ridge_coefs)

```
```{r}
lasso_cv <- cv.glmnet(X, y, alpha = 1)

best_lambda_lasso <- lasso_cv$lambda.min
cat("Best Lasso lambda:", best_lambda_lasso, "\n")

lasso_coefs <- coef(lasso_cv, s = best_lambda_lasso)
print(lasso_coefs)

```

```{r}
cat("Ridge CV MSE:", min(ridge_cv$cvm), "\n")
cat("Lasso CV MSE:", min(lasso_cv$cvm), "\n")

```
Lasso method of regularization is better. It did not have to shrink any coefficients to zero, suggesting that the model we have will be great at using on un seen data. Its a great prediction model that is neither overfit or underfit.

model diagnostics:

```{r}
plot(lebron_stepwise_model3, which=1)

```

```{r}
plot(lebron_stepwise_model3, which=2)
```

3 assumptions of linear regression:

constant variance: looking at the red trendline in the residuals vs fitted plot. the error terms seem to be vertically spread the same amount throughout the plot as we move left to right. This assumption addresses homoscedasticity, and it appears to satisfy that the model has homoscedasticity

linearity: looking at the residuals vs fitted plot, the error terms seem to have a mean of zero

normality: the error terms seem to follow the normal distribution as we can see in the Q-Q residuals plot. the circles are supposed to follow the normal distribution. It is off at the tail end, but we only have 22 data points so as long as it generally fits the trend, it is fine.

```{r}
win_shares_model <- lm(formula = Wins ~ `WS/48`, data = lebron_regular_model_master)

summary(win_shares_model)
```
Win shares per 48 minutes is the MVP predictor, but I decided not to use it alone because the multiple R squared was only about 56%. only 56% of the variation in Wins is explained by WS/48. Very good predictor, but the model is not complex enough.

I built a linear model using regular season statistics on Lebron. I started with stepwise selection, which started with the null model and added predictors that minimizes the AIC better than the null model would. AIC balances how well the model fits the data with model complexity. a model that is too complex will cause overfitting, which makes it impossible to make predictions based on new data. An underfit model simply does not fit the model well, and over generalizes to new data. We made one final tweak by disposing two insignificant predictors. We ran ridge and lasso regularization and found that neither of these methods shrunk any predictors, although there were some coefficient adjustments. We got a cross validation error of 32, which means the model can generalize to unseen data pretty well. Lasso slightly out performed Ridge in terms of predictive performance. Lastly, all of the linear assumptions were confirmed using diagnostic plots. In conclusion, the combination of Win shares per 48 minutes, steals per game, and rebounds per game do a good job of predicting how many wins Lebron's team gets in a season.

Michael Jordan Regular Season Linear Model:

```{r}
jordan_regular_model_master
```
```{r}
jordan_prime_regular_model_master <- jordan_regular_model_master[1:
(nrow(jordan_regular_model_master) - 2), ]
```

```{r}
jordan_corr_data <- jordan_prime_regular_model_master[, c("DRtg", "ORtg", "PER", "WS/48",
"eFG%", "TS%", "VORP", "USG%",
                                                         "PTS_pergame", "TRB_pergame",
"STL_pergame", "BLK_pergame",
                                                         "AST_pergame", "BPM")]

jordan_corr_matrix <- cor(jordan_corr_data, use = "complete.obs")


corrplot(jordan_corr_matrix, method = "color", type = "upper",
         tl.col = "red", tl.srt = 45)

```
```{r}
jordan_intercept_model1 <- lm(Wins ~ 1, data = jordan_prime_regular_model_master)

jordan_full_model1 <- lm(Wins ~ PER + `WS/48` + VORP + `USG%` + ORtg +
                           PTS_pergame + TRB_pergame + STL_pergame + BLK_pergame +
AST_pergame,
                           data = jordan_prime_regular_model_master)

jordan_stepwise_model1 <- step(jordan_intercept_model1,
                               scope = formula(jordan_full_model1),
                               direction = "forward", steps = 6)

summary(jordan_stepwise_model1)

```


cross validation:

```{r}
formula <- Wins ~ `WS/48` + STL_pergame

cv_results <- cv.glm(jordan_prime_regular_model_master, glm(formula, data =
jordan_prime_regular_model_master), K = 12)
cat("LOOCV error:", cv_results$delta[1], "\n")
```


ridge regression:

```{r}
X <- model.matrix(formula, data = jordan_prime_regular_model_master)[, -1]
y <- jordan_prime_regular_model_master$Wins

ridge_cv <- cv.glmnet(X, y, alpha = 0)
best_lambda_ridge <- ridge_cv$lambda.min
cat("Best Ridge lambda:", best_lambda_ridge, "\n")
cat("Best Ridge MSE:", min(ridge_cv$cvm), "\n")
```
Lasso regression:

```{r}
lasso_cv <- cv.glmnet(X, y, alpha = 1)
best_lambda_lasso <- lasso_cv$lambda.min
cat("Best Lasso lambda:", best_lambda_lasso, "\n")
cat("Best Lasso MSE:", min(lasso_cv$cvm), "\n")
```
Lasso is the preffered method because of a lower MSE, better ability to use the training

data on unseen data for prediction

diagnostic plots:

```{r}
plot(jordan_stepwise_model1, which = 1)
```
```{r}
plot(jordan_stepwise_model1, which = 2)
```

All three assumptions are violated, but its expected since there are only 12 data points. I tried using polynomial terms and interaction terms and it did not make it better. This is the best model we could make, but we must be careful in using it since there are only 9 degrees of freedom. The mild heteroscedasicity, non linearity of residuals, and non-normality of the residuals question whether this is a great prediction model.
'

We started by making a correlation matrix for the predictors to avoid multicollinearity. Then we used forward stepwise selection to arrive at WS/48 and Steals per game. Both of these predictors led to a low AIC, which fits the model enough but avoids over complexity. We had a significant F stat, both predictors were significant due to the p value being lower than a 0.05 significance level, and a low RSE. This model fits the data well and will perform well in predicting unseen data. a disclaimer is that the three assumptions for linear regression were slightly off due to the degree of freedom being only 9.

Logistic Regression:

```{r}
lebron_james_logistic <- read.csv("lebron_regular_model_master.csv")

win_pct <- c(
  0.418,
  0.513,
  0.595,
  0.603,
  0.600,
  0.815,
  0.789,
  0.722,
  0.726,
  0.803,
  0.675,
  0.725,
  0.737,
  0.689,
  0.610,
  0.509,
  0.746,
  0.667,
  0.446,
  0.545,
  0.577,
  0.629
)
```

```
# Add as new column
lebron_james_logistic$Win_Percentage <- win_pct
```

```{r}
lebron_james_logistic
```

```{r}
library(dplyr)
lebron_james_logistic <- lebron_james_logistic %>%
  mutate(Win60 = ifelse(Win_Percentage > 0.60, 1, 0))

model <- glm(
  Win60 ~ PTS + AST + TRB + `STL.` + `BLK.` + `TS.` + `eFG..1` +
    OnCourt + `On.Off` + OBPM + DBPM + OWS + DWS + `FG..1` +
    `X3P..1` + `FT..1` + BPM + WS + `WS.48` + VORP + PER + `USG.`,
  data = lebron_james_logistic,
  family = "binomial"
)

summary(model)
```
```{r}
model2 <- glm(
  Win60 ~ 1 + PER + PTS,
  data = lebron_james_logistic,
  family = "binomial"
)

# Step 5: View model summary
summary(model2)
```

```{r}
library(ggplot2)

# Fix PTS at mean
mean_pts <- mean(lebron_james_logistic$PTS, na.rm = TRUE)

# Create a dataset of PER values while holding PTS constant
plot_data <- data.frame(
  PER = seq(min(lebron_james_logistic$PER), max(lebron_james_logistic$PER), length.out =
100),
  PTS = mean_pts
)

# Predict probabilities from the model
plot_data$Win60_Prob <- predict(model2, newdata = plot_data, type = "response")

# Raw data for plotting points
lebron_points <- lebron_james_logistic %>%
  select(PER, Win60)

# Plot
ggplot() +
  geom_point(data = lebron_points, aes(x = PER, y = Win60)) +
  geom_line(data = plot_data, aes(x = PER, y = Win60_Prob), color = "blue", size = 1) +
  labs(
    x = "PER",
    y = "P(Win% > 60%)",
    title = "Logistic Curve: Predicting Win60 from PER (PTS fixed)"
  ) +
```

```
  theme_minimal()

```
```{r}
mean_per <- mean(lebron_james_logistic$PER, na.rm = TRUE)

plot_data2 <- data.frame(
  PTS = seq(min(lebron_james_logistic$PTS), max(lebron_james_logistic$PTS), length.out =
100),
  PER = mean_per
)

plot_data2$Win60_Prob <- predict(model2, newdata = plot_data2, type = "response")

lebron_points2 <- lebron_james_logistic %>%
  select(PTS, Win60)

ggplot() +
  geom_point(data = lebron_points2, aes(x = PTS, y = Win60)) +
  geom_line(data = plot_data2, aes(x = PTS, y = Win60_Prob), color = "blue", size = 1) +
  labs(
    x = "PTS",
    y = "P(Win% > 60%)",
    title = "Logistic Curve: Predicting Win60 from PTS (PER fixed)"
  ) +
  theme_minimal()

```



```{r}
lebron_james_logistic$pred_prob <- predict(model2, type = "response")

lebron_james_logistic$pred_class <- ifelse(lebron_james_logistic$pred_prob >= 0.5, 1, 0)

```



```{r}
table(Predicted = lebron_james_logistic$pred_class, Actual = lebron_james_logistic$Win60)

```
```{r}
TP <- sum(lebron_james_logistic$pred_class == 1 & lebron_james_logistic$Win60 == 1)
TN <- sum(lebron_james_logistic$pred_class == 0 & lebron_james_logistic$Win60 == 0)
FP <- sum(lebron_james_logistic$pred_class == 1 & lebron_james_logistic$Win60 == 0)
FN <- sum(lebron_james_logistic$pred_class == 0 & lebron_james_logistic$Win60 == 1)

Accuracy <- (TP + TN) / (TP + TN + FP + FN)
TPR <- TP / (TP + FN)
TNR <- TN / (TN + FP)
FPR <- FP / (FP + TN)
FNR <- FN / (FN + TP)

cat("Accuracy:", Accuracy, "\n")
cat("TPR (Sensititivy):", TPR, "\n")
cat("TNR (Specificity):", TNR, "\n")
cat("FPR (False Positive Rate:", FPR, "\n")
cat("FNR (False Negative Rate):", FNR, "\n")

```
```{r}
plot(
  roc_obj,
```

```
  col = "blue",
  lwd = 2,
  main = "ROC Curve for PER + PTS Model",
  legacy.axes = TRUE,
  xaxs = "i",
  yaxs = "i"
)
abline(a = 0, b = 1, lty = 2, col = "gray")

```

I created a logistic regression model to evaluate how PER (Player Efficiency Rating) and PTS (Points per Game) predict the probability that a given LeBron James season had a win percentage above 60%.

Surprisingly, the model showed that higher scoring (PTS) was associated with a lower likelihood of a high-win season, while higher PER was positively associated with team success. This suggests that in seasons where Lebron scored more, it may have reflected a weaker supporting cast, requiring him to carry more of the offensive load, which did not translate into team wins. In contrast, a high PER likely reflects efficient, well-rounded play, which may indicate a better-fitting system or stronger teammates allowing him to maximize his strengths.

The model achieved high accuracy, with a Type I error rate (false positive) of 0 and a low Type II error rate (false negative). The True Positive Rate (Sensitivity) was also strong, showing that the model is effective at identifying high-win seasons when they actually occur. Finally, the ROC curve had a high AUC, indicating strong discriminatory power. The curve hugged the top-left corner, which reflects a high-quality binary classifier.

Michael Jordan Logistic Model:

```{r}
jordan_logistic <- read.csv("jordan_regular_model_master.csv")

win_pct <- c(0.463, 0.500, 0.488, 0.610, 0.580, 0.671,
             0.744, 0.838, 0.718, 0.878, 0.841, 0.756,
             0.500, 0.451)

jordan_logistic$Win_Percentage <- win_pct

```

```{r}
jordan_logistic$Win60 <- ifelse(jordan_logistic$Win_Percentage > 0.60, 1, 0)

jordan_model_full <- glm(
  Win60 ~ PTS_pergame + AST_pergame + TRB_pergame + STL_pergame + BLK_pergame +
    TS..1 + eFG..1 + OBPM + DBPM + OWS + DWS + FG..1 + X3P..1 + FT..1 +
    BPM + WS + WS.48 + VORP + PER + USG.,
  data = jordan_logistic,
  family = "binomial"
)

summary(jordan_model_full)

```

```{r}
jordan_model2 <- glm(
  Win60 ~ DRtg,
  data = jordan_logistic,
```

```
  family = "binomial"
)


summary(jordan_model2)
```

```{r}
jordan_logistic$pred_prob <- predict(jordan_model2, type = "response")

jordan_logistic$pred_class <- ifelse(jordan_logistic$pred_prob >= 0.5, 1, 0)


```



```{r}
table(Predicted = jordan_logistic$pred_class, Actual = jordan_logistic$Win60)

TP <- sum(jordan_logistic$pred_class == 1 & jordan_logistic$Win60 == 1)
TN <- sum(jordan_logistic$pred_class == 0 & jordan_logistic$Win60 == 0)
FP <- sum(jordan_logistic$pred_class == 1 & jordan_logistic$Win60 == 0)
FN <- sum(jordan_logistic$pred_class == 0 & jordan_logistic$Win60 == 1)

Accuracy <- (TP + TN) / (TP + TN + FP + FN)
TPR <- TP / (TP + FN)
TNR <- TN / (TN + FP)
FPR <- FP / (FP + TN)
FNR <- FN / (FN + TP)

cat("Accuracy:", Accuracy, "\n")
cat("TPR (Sensitivity):", TPR, "\n")
cat("TNR (Specificity):", TNR, "\n")
cat("FPR (False Positive Rate):", FPR, "\n")
cat("FNR (False Negative Rate):", FNR, "\n")

```
```{r}
jordan_logistic$Win55 <- ifelse(jordan_logistic$Win_Percentage > 0.55, 1, 0)

jordan_model3 <- glm(Win55 ~ DRtg, data = jordan_logistic, family = "binomial")

summary(jordan_model3)
```
```{r}
jordan_logistic$pred_prob <- predict(jordan_model3, type = "response")

jordan_logistic$pred_class <- ifelse(jordan_logistic$pred_prob >= 0.5, 1, 0)
```



```{r}
table(Predicted = jordan_logistic$pred_class, Actual = jordan_logistic$Win55)

TP <- sum(jordan_logistic$pred_class == 1 & jordan_logistic$Win55 == 1)
TN <- sum(jordan_logistic$pred_class == 0 & jordan_logistic$Win55 == 0)
FP <- sum(jordan_logistic$pred_class == 1 & jordan_logistic$Win55 == 0)
FN <- sum(jordan_logistic$pred_class == 0 & jordan_logistic$Win55 == 1)

Accuracy <- (TP + TN) / (TP + TN + FP + FN)
TPR <- TP / (TP + FN)
TNR <- TN / (TN + FP)
```

```
FPR <- FP / (FP + TN)
FNR <- FN / (FN + TP)

cat("Accuracy:", Accuracy, "\n")
cat("TPR (Sensitivity):", TPR, "\n")
cat("TNR (Specificity):", TNR, "\n")
cat("FPR (False Positive Rate):", FPR, "\n")
cat("FNR (False Negative Rate):", FNR, "\n")
```

We'll use the 55% win percentage threshold because the increase in power (True Positive Rate) outweighs the slight rise in the Type I error rate, making the model more sensitive to correctly identifying strong seasons without sacrificing too much specificity. Also, the predictor Defensive Rating officially becomes significant with a p-value below 0.05.

```{r}
plot_data <- data.frame(
  DRtg = seq(min(jordan_logistic$DRtg, na.rm = TRUE),
             max(jordan_logistic$DRtg, na.rm = TRUE),
             length.out = 100)
)

plot_data$Win55_Prob <- predict(jordan_model3, newdata = plot_data, type = "response")

jordan_points <- jordan_logistic %>%
  select(DRtg, Win55)

ggplot() +
  geom_point(data = jordan_points, aes(x = DRtg, y = Win55)) +
  geom_line(data = plot_data, aes(x = DRtg, y = Win55_Prob), color = "blue", size = 1) +
  labs(
    x = "Defensive Rating (DRtg)",
    y = "P(Win% > 55%)",
    title = "Logistic Curve: Predicting Win55 from DRtg"
  ) +
  theme_minimal()

```

```{r}
roc_obj <- roc(jordan_logistic$Win55, jordan_logistic$pred_prob)

plot(
  roc_obj,
  col = "blue",
  lwd = 2,
  main = "ROC Curve for Jordan's DRtg Model",
  legacy.axes = TRUE,
  xaxs = "i",
  yaxs = "i"
)
abline(a = 0, b = 1, lty = 2, col = "gray")

```

I created a logistic model to evaluate whether Michael Jordan’s Defensive Rating (DRtg) could predict the probability that a given season had a win percentage above 55%. DRtg was statistically significant in the model, with lower DRtg (better defense) corresponding to a higher probability of team success. We adjusted our win threshold to 55% to improve

model power (true positive rate), accepting a slight increase in the false positive rate to better detect seasons where Jordan’s teams were strong. The logistic curve showed a clear negative relationship, and the ROC curve demonstrated solid classification ability, with the curve hugging the upper left, indicating decent predictive power for the one-predictor model