Data Science Project

James Li

06/30/24

Introduction

The purpose of this project is to gauge your technical skills and problem solving ability by working through something science project. You will work your way through this R Markdown document, answering questions as you go along. Pl name to the “author” key in the YAML header. When you’re finished with the document, come back and type your ans the top. Please leave all your work below and have your answers where indicated below as well. Please note that we so make it clear, concise and avoid long printouts. Feel free to add in as many new code chunks as you’d like.

Remember that we will be grading the quality of your code and visuals alongside the correctness of your answers. Ple as much as possible (instead of base R and explicit loops). Please do not bring in any outside data.

Note:

Throughout this document, any `season` column represents the year each season started. For example, the 20 dataset as 2015. For most of the rest of the project, we will refer to a season by just this number (e.g. 2015) in (e.g. 2015-16).

Answers

Part 1

Question 1:

- Offensive: 56.5% eFG
- Defensive: 47.9% eFG

Question 2: 81.7%

Question 3: 46.2%

Question 4: This is a written question. Please leave your response in the document under Question 5.

Question 5: 94.3% of games

Question 6:

- Round 1: 60.1%
- Round 2: 60.2%
- Conference Finals: 58.4%
- Finals: 52.9%

Question 7:

- Percent of +5.0 net rating teams making the 2nd round next year: 61.76%
- Percent of top 5 minutes played players who played in those 2nd round series: 35.2%

Part 2

Please show your work in the document, you don’t need anything here.

Part 3

Please write your response in the document, you don’t need anything here.

Setup and Data

```
library(tidyverse)
# Note, you will likely have to change these paths. If your data is in the same folder as this proje
# the paths will likely be fixed for you by deleting ../../Data/awards_project/ from each string.
# player_data <- read_csv("../../Data/playoffs_project/player_data.csv")
# team_data <- read_csv("../../Data/playoffs_project/team_data.csv")
```

Part 1 – Data Cleaning

In this section, you're going to work to answer questions using data from both team and player stats. All provided stats

Question 1

QUESTION: What was the Warriors' Team offensive and defensive eFG% in the 2015-16 regular season? Remember 2015 season.

```
file.choose()
```

```
## [1] "C:\\Users\\James\\OneDrive\\桌面\\Thunder Project\\Thunder_up_final.html"
```

```
install.packages("dplyr")
```

```
## Warning: package 'dplyr' is in use and will not be installed
```

```
library(dplyr)

# Load the data
team_data <- read.csv("C:\\Users\\James\\OneDrive\\桌面\\team_game_data.csv")

warriors_games <- subset(team_data, season == 2015 & gametype == 2)
warriors_offensive <- subset(warriors_games, off_team_name == "Golden State Warriors")
warriors_offensive$off_eFG <- (warriors_offensive$fgmade + 0.5 * warriors_offensive$fg3made) / warri
d
average_off_eFG <- mean(warriors_offensive$off_eFG, na.rm = TRUE)
warriors_defensive <- subset(warriors_games, def_team_name == "Golden State Warriors")
warriors_defensive$def_eFG <- (warriors_defensive$fgmade + 0.5 * warriors_defensive$fg3made) / warri
d
average_def_eFG <- mean(warriors_defensive$def_eFG, na.rm = TRUE)
print(paste("Offensive eFG%:", round(average_off_eFG * 100, 1), "%"))
```

```
## [1] "Offensive eFG%: 56.5 %"
```

```
print(paste("Defensive eFG%:", round(average_def_eFG * 100, 1), "%"))
```

```
## [1] "Defensive eFG%: 47.9 %"
```

ANSWER 1:

Offensive: 56.5% eFG
Defensive: 47.9% eFG

Question 2

QUESTION: What percent of the time does the team with the higher eFG% in a given game win that game? Use gam regular seasons. If the two teams have an exactly equal eFG%, remove that game from the calculation.

```
team_data <- read.csv("C:\\Users\\James\\OneDrive\\桌面\\team_game_data.csv")

# Calculate eFG% for each team
team_data$efg_percent <- (team_data$fgmade + 0.5 * team_data$fg3made) / team_data$fgattempted

# Pivot the data to get each game's teams on the same row
library(tidyr)
library(dplyr)

paired_games <- team_data %>%
  pivot_wider(id_cols = nbagameid,
              names_from = off_home,
              values_from = c(off_team_name, efg_percent, off_win),
              names_glue = "{.value}_{off_home}")

# Calculate which team had the higher eFG%
paired_games <- paired_games %>%
  mutate(higher_efg_team = ifelse(efg_percent_1 > efg_percent_0, off_team_name_1, off_team_name_0))

# Determine which team won
paired_games <- paired_games %>%
  mutate(winning_team = ifelse(off_win_1 == 1, off_team_name_1, off_team_name_0))

# Filter out games where eFG% is equal
filtered_games <- paired_games %>%
  filter(efg_percent_1 != efg_percent_0)

# Calculate the percentage of games where the team with the higher eFG% won
higher_efg_wins <- mean(filtered_games$higher_efg_team == filtered_games$winning_team) * 100

# Print the result
print(higher_efg_wins)
```

```
## [1] 81.68931
```

ANSWER 2:

81.7%

Question 3

QUESTION: What percent of the time does the team with more offensive rebounds in a given game win that game? U
2023 regular seasons. If the two teams have an exactly equal number of offensive rebounds, remove that game from

```
# Load the data
team_data <- read.csv("C:\\Users\\James\\OneDrive\\桌面\\team_game_data.csv")

# Filter data for the 2014-2023 regular seasons
filtered_data <- subset(team_data, season >= 2014 & season <= 2023 & gametype == 2)

# Ensure required packages are installed and loaded
if (!require(tidyr)) install.packages("tidyr")
if (!require(dplyr)) install.packages("dplyr")
library(tidyr)
library(dplyr)

# Pivot the data to get each game's teams on the same row
paired_games <- filtered_data %>%
  pivot_wider(id_cols = nbagameid,
              names_from = off_home,
              values_from = c(off_team_name, reboffensive, off_win),
              names_glue = "{.value}_{off_home}")

# Calculate which team had more offensive rebounds and which team won
paired_games <- paired_games %>%
  mutate(more_rebounds_team = ifelse(reboffensive_1 > reboffensive_0, off_team_name_1, off_team_name_
         winning_team = ifelse(off_win_1 == 1, off_team_name_1, off_team_name_0),
         same_rebounds = reboffensive_1 == reboffensive_0)

# Filter out games where the number of offensive rebounds is the same
filtered_games <- paired_games %>%
  filter(!same_rebounds)

# Calculate the percentage of games where the team with more offensive rebounds won
more_rebounds_wins_percentage <- mean(filtered_games$more_rebounds_team == filtered_games$winning_te

# Print the result
print(more_rebounds_wins_percentage)
```

```
## [1] 46.21415
```

```
cat(sprintf("<span style=\"color:red\">ANSWER 3:</span>\n\n%.1f%%", more_rebounds_wins_percentage))
```

```
## <span style="color:red">ANSWER 3:</span>
##
## 46.2%
```

ANSWER 3:

46.2%

Question 4

QUESTION: Do you have any theories as to why the answer to question 3 is lower than the answer to question 2? Try your answer.

ANSWER 4:

Based on the data from the 2014-2023 regular seasons, it was calculated that the probability of a team with a higher e outcome of a game is nearly 36% higher than that of a team with more offensive rebounds. Since eFG% reflects scori direct and substantial impact on game outcomes. An eFG% above 50% increases a team's chances of winning by sc attempt. In contrast, offensive rebounds provide additional scoring opportunities, but they are less reliable when it com of the game. There are many factors that contribute to winning a game, including defense, turnovers, and free throws. offensive rebounds may lose if their opponent excels in these other areas. Therefore, teams with higher eFG% have a game than teams with higher offensive rebound rates.

Question 5

QUESTION: Look at players who played at least 25% of their possible games in a season and scored at least 25 poin player-seasons, what percent of games were they available for on average? Use games from the 2014-2023 regular s

For example:

- Ja Morant does not count in the 2023-24 season, as he played just 9 out of 82 games this year, even though he
  game.
- Chet Holmgren does not count in the 2023-24 season, as he played all 82 games this year but scored 16.5 poir
- LeBron James does count in the 2023-24 season, as he played 71 games and scored 25.7 points per game.

```
#Q4:
cat(sprintf("<span style=\"color:red\">ANSWER 4:</span>\n\nBased on the data from the 2014-2023 regu
ulated that the probability of a team with a higher eFG%% determining the outcome of a game is nearl
of a team with more offensive rebounds. Since eFG%% reflects scoring efficiency, it has a more direc
on game outcomes. An eFG%% above 50%% increases a team's chances of winning by scoring more points p
rast, offensive rebounds provide additional scoring opportunities, but they are less reliable when i
e outcome of the game. There are many factors that contribute to winning a game, including defense,
ws. Even a team with more offensive rebounds may lose if their opponent excels in these other areas.
igher eFG%% have a greater chance of winning a game than teams with higher offensive rebound rates."
```

```
## <span style="color:red">ANSWER 4:</span>
##
## Based on the data from the 2014-2023 regular seasons, it was calculated that the probability of a
determining the outcome of a game is nearly 36% higher than that of a team with more offensive rebou
scoring efficiency, it has a more direct and substantial impact on game outcomes. An eFG% above 50%
es of winning by scoring more points per shot attempt. In contrast, offensive rebounds provide addit
ies, but they are less reliable when it comes to predicting the outcome of the game. There are many
to winning a game, including defense, turnovers, and free throws. Even a team with more offensive re
opponent excels in these other areas. Therefore, teams with higher eFG% have a greater chance of win
ith higher offensive rebound rates.
```

```
#Q5:
file.choose()
```

```
## [1] "C:\\Users\\James\\OneDrive\\桌面\\Thunder Project\\Thunder_up_final.html"
```

```
library(dplyr)

# Load the data
player_data <- read.csv("C:\\Users\\James\\OneDrive\\桌面\\player_game_data.csv")

# Calculate total games played and total points per player per season
library(dplyr)
player_season_stats <- player_data %>%
  group_by(season, player_name) %>%
  summarise(
    games_played = n(),
    total_points = sum(points),
    avg_points_per_game = total_points / games_played
  )
```

```
## `summarise()` has grouped output by 'season'. You can override using the
## `.groups` argument.
```

```
# Filter players who played at least 25% of 82 games and scored at least 25 points per game
qualifying_players <- player_season_stats %>%
  filter(games_played >= 20.5, avg_points_per_game >= 25)

# Calculate average percentage of games played
average_percentage_games_played <- mean(qualifying_players$games_played / 82) * 100

# Format and output the result with HTML styling for the color
cat(sprintf('<span style="color:red">**ANSWER 5:**</span> %.1f%% of games', average_percentage_games
```

```
## <span style="color:red">**ANSWER 5:**</span> 94.3% of games
```

ANSWER 5:

94.3% of games

Question 6

QUESTION: What % of playoff series are won by the team with home court advantage? Give your answer by round. U
2014-**2022** seasons. Remember that the 2023 playoffs took place during the 2022 season (i.e. 2022-23 season).

```
library(dplyr)

# Load the data
data <- read.csv("C:\\Users\\James\\OneDrive\\桌面\\team_game_data.csv")

# Filter for playoff games from 2014-2022 seasons, and only gametype 4
playoff_data <- data %>%
  filter(gametype == 4, season >= 2014, season <= 2022) %>%
  select(season, nbagameid, off_team, off_home, off_win, def_team, def_home, def_win, gamedate) %>%
  arrange(season, gamedate)

# Determine winners for each game
playoff_data <- playoff_data %>%
  mutate(winner = ifelse(off_win == 1, off_team, def_team),
         loser = ifelse(off_win == 1, def_team, off_team),
         home_team = ifelse(off_home == 1, off_team, def_team),
         home_win = ifelse(home_team == winner, 1, 0)) %>%
  distinct(nbagameid, .keep_all = TRUE)

# Helper function to classify rounds based on wins
classify_rounds <- function(df) {
  df <- df %>%
    arrange(gamedate) %>%
    group_by(season, winner) %>%
    mutate(win_count = row_number()) %>%
    ungroup() %>%
    mutate(round = case_when(
      win_count <= 4 ~ "Round 1",
      win_count <= 8 ~ "Round 2",
      win_count <= 12 ~ "Conference Finals",
      TRUE ~ "Finals"
    ))
  return(df)
}

# Apply the classification function
playoff_data <- playoff_data %>%
  group_by(season, winner) %>%
  do(classify_rounds(.)) %>%
  ungroup()

# Calculate summaries for rounds
final_results <- playoff_data %>%
  group_by(round) %>%
  summarise(
    total_games = n(),
    home_wins = sum(home_win),
    percentage = home_wins / total_games * 100
  )

# Print the results in the specified format
cat("<span style=\"color:red\">**ANSWER 6:**</span>\n\n")
```

```
## <span style="color:red">**ANSWER 6:**</span>
```

```
for (i in 1:nrow(final_results)) {
  cat(final_results$round[i], ": ", sprintf("%.1f", final_results$percentage[i]), "%\n")
}
```

```
## Conference Finals :  58.4 %
## Finals :  52.9 %
## Round 1 :  60.1 %
## Round 2 :  60.2 %
```

ANSWER 6:

Round 1: 60.1%
Round 2: 60.2%
Conference Finals: 58.4%
Finals: 52.9%

Question 7

QUESTION: Among teams that had at least a +5.0 net rating in the regular season, what percent of them made the se the **following** year? Among those teams, what percent of their top 5 total minutes played players (regular season) in t played in that 2nd round playoffs series? Use the 2014-2021 regular seasons to determine the +5 teams and the 201 data.

For example, the Thunder had a better than +5 net rating in the 2023 season. If we make the 2nd round of the playoff would qualify for this question. Our top 5 minutes played players this season were Shai Gilgeous-Alexander, Chet Hol Williams, and Josh Giddey. If three of them play in a hypothetical 2nd round series next season, it would count as 3/5

Hint: The definition for net rating is in the data dictionary.

```
# Load necessary libraries
library(dplyr)
library(readr)

file.choose()
```

```
## [1] "C:\\Users\\James\\OneDrive\\桌面\\Thunder Project\\Thunder_up_final.html"
```

```
# Load the team game data for regular seasons
team_data <- read_csv('C:\\Users\\James\\Downloads\\merged_team_game_data.csv')
```

```
## Rows: 27144 Columns: 42
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr   (5): off_team_name, off_team, def_team_name, def_team, round
## dbl  (36): season, gametype, nbagameid, offensivenbateamid, off_home, off_wi...
## date  (1): gamedate
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Calculate possessions
team_data <- team_data %>%
  mutate(possessions = fgattempted - reboffensive + turnovers + 0.44 * ftattempted)

# Create a copy of team_data for merging purposes
team_data_copy <- team_data %>%
  select(nbagameid, off_team_name, points) %>%
  rename(def_team_name = off_team_name, def_points = points)

# Merge team_data with team_data_copy to get opponent's points
merged_data <- merge(team_data, team_data_copy, by.x = c("nbagameid", "def_team_name"), by.y = c("nb
e"))

# Calculate ORTG and DRTG
merged_data <- merged_data %>%
  mutate(ortg = 100 * points / possessions,
         drtg = 100 * def_points / possessions,
         net_rating = ortg - drtg)

# Filter for regular season games from 2014 to 2021
team_net_ratings <- merged_data %>%
  filter(season >= 2014 & season <= 2021 & gametype == 2) %>%
  group_by(off_team_name, season) %>%
  summarize(net_rating = mean(net_rating)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'off_team_name'. You can override using the
## `.groups` argument.
```

```r
# Identify teams with at least a +5.0 net rating
high_net_rating_teams <- team_net_ratings %>%
  filter(net_rating >= 5.0)

# Ensure the 'round' column is in the correct case and remove any leading/trailing whitespace
team_data <- team_data %>%
  mutate(round = trimws(round))

# Filter for playoff games and within the specified seasons
playoff_data <- team_data %>%
  filter(gametype == 4 & season >= 2015 & season <= 2022)

# Check the number of records for each season to ensure completeness
print(table(playoff_data$season))
```

```
##
## 2015 2016 2017 2018 2019 2020 2021 2022
##  172  158  164  164  166  170  174  168
```

```r
# Filter for teams that made it to the second round (Round 2)
teams_round2 <- playoff_data %>%
  filter(round == 'Round 2')

# Get the unique teams and seasons that made it to the second round
teams_made_round2 <- teams_round2 %>%
  select(off_team_name, season) %>%
  distinct()

# Add one season to high_net_rating_teams to check next season's playoff performance
high_net_rating_teams <- high_net_rating_teams %>%
  mutate(next_season = season + 1)

# Check how many high net rating teams made it to Round 2 in the next season
high_net_rating_next_season_round2 <- high_net_rating_teams %>%
  inner_join(teams_made_round2, by = c("off_team_name", "next_season" = "season"))

# Calculate the percentage
percent_high_net_rating_next_season_round2 <- nrow(high_net_rating_next_season_round2) / nrow(high_n

print(paste("Percentage of high net rating teams that made the second round in the next season:", ro
ing_next_season_round2, 2), "%"))
```

```
## [1] "Percentage of high net rating teams that made the second round in the next season: 61.76 %"
```

```r
file.choose()
```

```
## [1] "C:\\Users\\James\\OneDrive\\桌面\\Thunder Project\\Thunder_up_final.html"
```

```r
# Load player data
player_data <- read_csv('C:\\Users\\James\\OneDrive\\桌面\\player_data_with_round.csv')
```

```
## Rows: 481783 Columns: 60
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr   (6): player_name, team, team_name, opp_team, opp_team_name, round
## dbl  (53): nbagameid, season, gametype, nbapersonid, nbateamid, opposingnbat...
## date  (1): gamedate
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Filter for regular season player data
regular_season_player_data <- player_data %>%
  filter(gametype == 2 & season >= 2014 & season <= 2021) %>%
  mutate(total_minutes = seconds / 60)

# Filter out players from high net rating teams
high_net_rating_team_names <- unique(high_net_rating_next_season_round2$off_team_name)
top_team_players <- regular_season_player_data %>%
  filter(team_name %in% high_net_rating_team_names)

# Summarize total minutes played by each player for each season and team
top_team_players_summary <- top_team_players %>%
  group_by(season, team_name, nbapersonid, player_name) %>%
  summarize(total_minutes = sum(total_minutes, na.rm = TRUE)) %>%
  ungroup() %>%
  arrange(season, team_name, desc(total_minutes))
```

```
## `summarise()` has grouped output by 'season', 'team_name', 'nbapersonid'. You
## can override using the `.groups` argument.
```

```r
# Get the top 5 players by minutes played for each team and season
top_5_team_players <- top_team_players_summary %>%
  group_by(season, team_name) %>%
  slice_head(n = 5) %>%
  ungroup()

# Filter out high net rating teams that made it to the second round of the playoffs the next season
high_net_rating_teams_in_round2 <- high_net_rating_next_season_round2 %>%
  select(off_team_name, season) %>%
  rename(season_next = season)

# Merge to get the top 5 players in these teams
top_5_players_in_round2_teams <- top_5_team_players %>%
  inner_join(high_net_rating_teams_in_round2, by = c("team_name" = "off_team_name", "season" = "seas

# Display the results
print(top_5_players_in_round2_teams %>% select(season, team_name, player_name, total_minutes))
```

```
## # A tibble: 105 × 4
##    season team_name             player_name     total_minutes
##     <dbl> <chr>                 <chr>                   <dbl>
##  1   2014 Atlanta Hawks         Kyle Korver             2418.
##  2   2014 Atlanta Hawks         Paul Millsap            2390.
##  3   2014 Atlanta Hawks         Al Horford              2318.
##  4   2014 Atlanta Hawks         Jeff Teague             2228.
##  5   2014 Atlanta Hawks         DeMarre Carroll         2189.
##  6   2014 Golden State Warriors Stephen Curry           2613.
##  7   2014 Golden State Warriors Draymond Green          2490.
##  8   2014 Golden State Warriors Klay Thompson           2454.
##  9   2014 Golden State Warriors Harrison Barnes         2318.
## 10   2014 Golden State Warriors Andre Iguodala         2069.
## # ℹ 95 more rows
```

```r
# Filter for regular season player data
regular_season_player_data <- player_data %>%
  filter(gametype == 2 & season >= 2014 & season <= 2021) %>%
  mutate(total_minutes = seconds / 60)

# Filter out players from high net rating teams
high_net_rating_team_names <- unique(high_net_rating_next_season_round2$off_team_name)
top_team_players <- regular_season_player_data %>%
  filter(team_name %in% high_net_rating_team_names)

# Summarize total minutes played by each player for each season and team
top_team_players_summary <- top_team_players %>%
  group_by(season, team_name, nbapersonid, player_name) %>%
  summarize(total_minutes = sum(total_minutes, na.rm = TRUE)) %>%
  ungroup() %>%
  arrange(season, team_name, desc(total_minutes))
```

```
## `summarise()` has grouped output by 'season', 'team_name', 'nbapersonid'. You
## can override using the `.groups` argument.
```

```r
# Get the top 5 players by minutes played for each team and season
top_5_team_players <- top_team_players_summary %>%
  group_by(season, team_name) %>%
  slice_head(n = 5) %>%
  ungroup()

# Add next season column for checking playoff participation
top_5_team_players <- top_5_team_players %>%
  mutate(next_season = season + 1)

# Filter out players who played in the second round of playoffs in the next season
next_season_players_round2 <- player_data %>%
  filter(season >= 2015 & season <= 2022 &
         team_name %in% high_net_rating_next_season_round2$off_team_name &
         round == 'Round 2')

# Check if these players played in the second round of playoffs in the next season
top_5_players_next_season_round2 <- merge(top_5_team_players, next_season_players_round2,
                                          by.x = c("next_season", "team_name", "nbapersonid"),
                                          by.y = c("season", "team_name", "nbapersonid"))

# Print the columns of the merged dataframe to identify correct column names
print(names(top_5_players_next_season_round2))
```

```
##  [1] "next_season"              "team_name"
##  [3] "nbapersonid"              "season"
##  [5] "player_name.x"            "total_minutes"
##  [7] "nbagameid"                "gamedate"
##  [9] "gametype"                 "player_name.y"
## [11] "nbateamid"                "team"
## [13] "opposingnbateamid"        "opp_team"
## [15] "opp_team_name"            "starter"
## [17] "missed"                   "seconds"
## [19] "points"                   "fg2made"
## [21] "fg2missed"                "fg2attempted"
## [23] "fg3made"                  "fg3missed"
## [25] "fg3attempted"             "fgmade"
## [27] "fgmissed"                 "fgattempted"
## [29] "ftmade"                   "ftmissed"
## [31] "ftattempted"              "reboffensive"
## [33] "rebdefensive"             "offensivereboundchances"
## [35] "defensivereboundchances"  "assists"
## [37] "steals"                   "stealsagainst"
## [39] "turnovers"                "blocks"
## [41] "blocksagainst"            "defensivefouls"
## [43] "defensivefoulsdrawn"      "offensivefouls"
## [45] "offensivefoulsdrawn"      "shootingfouls"
## [47] "shootingfoulsdrawn"       "shotattempts"
## [49] "shotattemptpoints"        "offensiveseconds"
## [51] "offensivepossessions"     "defensiveseconds"
## [53] "defensivepossessions"     "andones"
## [55] "teampoints"               "opponentteampoints"
## [57] "teamshotattempts"         "teamfgmade"
## [59] "teamfgattempted"          "teamturnovers"
## [61] "opponentteamfg2attempted" "opponentteamfg3attempted"
## [63] "round"
```

```
# Remove duplicates to ensure each player is counted only once
top_5_players_next_season_round2 <- top_5_players_next_season_round2 %>%
  distinct(next_season, team_name, nbapersonid, .keep_all = TRUE)

# Calculate the percentage
total_top_5_players <- nrow(top_5_team_players)
playoff_participating_players <- nrow(top_5_players_next_season_round2)

participation_percentage <- (playoff_participating_players / total_top_5_players) * 100

# Adjust column names based on the printed column names
playoff_participating_players_data <- top_5_players_next_season_round2 %>%
  select(next_season, team_name, player_name.x, total_minutes)

# Print results
cat(sprintf("Percentage of top 5 players who played in 2nd round of playoffs: %.2f%%\n", participati
```

```
## Percentage of top 5 players who played in 2nd round of playoffs: 35.23%
```

```
# Print results in the specified format
cat(sprintf("<span style=\"color:red\">**ANSWER 7:**</span>\n\n"))
```

```
## <span style="color:red">**ANSWER 7:**</span>
```

```
cat(sprintf("Percent of +5.0 net rating teams making the 2nd round next year: %.1f%%\n", round(perce
season_round2, 1)))
```

```
## Percent of +5.0 net rating teams making the 2nd round next year: 61.8%
```

```
cat(sprintf("Percent of top 5 minutes played players who played in those 2nd round series: %.1f%%\n"
rcentage, 1)))
```

```
## Percent of top 5 minutes played players who played in those 2nd round series: 35.2%
```

ANSWER 7:

Percent of +5.0 net rating teams making the 2nd round next year: 61.8%
Percent of top 5 minutes played players who played in those 2nd round series: 35.2%

Part 2 – Playoffs Series Modeling

For this part, you will work to fit a model that predicts the winner and the number of games in a playoffs series betwee

This is an intentionally open ended question, and there are multiple approaches you could take. Here are a few notes

1. Your final output must include the probability of each team winning the series. For example: “Team A has a 30% has a 70% chance.” instead of “Team B will win.” You must also predict the number of games in the series. This point estimate.
2. You may use any data provided in this project, but please do not bring in any external sources of data.
3. You can only use data available prior to the start of the series. For example, you can’t use a team’s stats from tl a playoffs series from the 2015-16 season.
4. The best models are explainable and lead to actionable insights around team and roster construction. We’re mo process and critical thinking than we are in specific modeling techniques. Using smart features is more importa mathematical machinery.
5. Include, as part of your answer:

- A brief written overview of how your model works, targeted towards a decision maker in the front office without a background.
- What you view as the strengths and weaknesses of your model.
- How you’d address the weaknesses if you had more time and/or more data.
- Apply your model to the 2024 NBA playoffs (2023 season) and create a high quality visual (a table, a plot, or a p (that made the first round) chances of advancing to each round.

```
file.choose()
```

```
## [1] "C:\\Users\\James\\OneDrive\\桌面\\Thunder Project\\Thunder_up_final.html"
```

```
# Install packages
packages <- c("readr", "dplyr", "caret", "randomForest", "corrplot")
install.packages(setdiff(packages, rownames(installed.packages())), dependencies = TRUE)

# Load libraries
library(readr)
library(dplyr)
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

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

```
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

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

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
# Load data
team_data <- read_csv("C:\\Users\\James\\OneDrive\\桌面\\merged_team_data_with_advanced_stats.csv")
```

```
## Rows: 27144 Columns: 52
```

```
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (6): gamedate, off_team_name, off_team, def_team_name, def_team, round
## dbl (46): season, gametype, nbagameid, offensivenbateamid, off_home, off_win...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
player_data <- read_csv("C:\\Users\\James\\OneDrive\\桌面\\merged_player_data_with_advanced_stats.csv
```

```
## Rows: 481783 Columns: 71
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): gamedate, player_name, team, team_name, opp_team, opp_team_name, r...
## dbl (64): nbagameid, season, gametype, nbapersonid, nbateamid, opposingnbate...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Ensure variables and target variable are numeric
team_features <- c('fg2made', 'fg2missed', 'fg2attempted', 'fg3made', 'fg3missed', 'fg3attempted', '
attempted_x', 'ftmade', 'ftmissed', 'ftattempted_x', 'reboffensive_x', 'rebdefensive', 'reboundchanc
ainst', 'turnovers_x', 'blocksagainst', 'defensivefouls', 'offensivefouls', 'shootingfoulsdrawn', 'p
x', 'shotattempts', 'andones', 'shotattemptpoints', 'steals', 'blocks', 'ortg', 'drtg', 'points_y',
mpted_y', 'turnovers_y', 'reboffensive_y', 'ftattempted_y', 'NET RTG')
player_features <- c('starter', 'points', 'fg2made', 'fg2missed', 'fg2attempted', 'fg3made', 'fg3mis
gmade', 'fgmissed', 'fgattempted', 'ftmade', 'ftmissed', 'ftattempted', 'reboffensive', 'rebdefensiv
nces', 'defensivereboundchances', 'assists', 'steals', 'stealsagainst', 'turnovers', 'blocks', 'bloc
uls', 'defensivefoulsdrawn', 'offensivefouls', 'offensivefoulsdrawn', 'shootingfouls', 'shootingfoul
'shotattemptpoints', 'offensiveseconds', 'offensivepossessions', 'defensiveseconds', 'defensiveposse
mpoints', 'opponentteampoints', 'teamshotattempts', 'teamfgmade', 'teamfgattempted', 'teamturnovers'
ted', 'opponentteamfg3attempted', 'PPA', 'USG%', 'AST%', 'OREB%', 'DREB%', 'TOV%', 'STL%', 'BLK%', '
G')

# Check and remove non-existing features from team_features
existing_team_features <- intersect(team_features, colnames(team_data))

# Convert all features and target variable to numeric
team_data <- team_data %>%
  mutate(across(all_of(c(existing_team_features, 'off_win')), as.numeric))

# Calculate correlation matrix
correlation_matrix <- cor(team_data[ ,c(existing_team_features, 'off_win')])
correlation_with_target <- sort(correlation_matrix[ , 'off_win'], decreasing = TRUE)

# Output the features most correlated with the target variable
cat("Top features correlated with playoff victory:\n")
```

```
## Top features correlated with playoff victory:
```

```
print(head(correlation_with_target, 20))
```

```
##           off_win          points_x shotattemptpoints           fgmade
##       1.000000000       0.431555332       0.374712550       0.369874534
##           assists          fg3made          fg2made           blocks
##       0.291927448       0.226359392       0.203188900       0.140235105
##            ftmade           steals     ftattempted_x    defensivefouls
##       0.139262723       0.111131714       0.105533879       0.099917423
##              ortg           andones          points_y     fgattempted_y
##       0.054344549       0.054054715       0.053783730       0.030975942
##     possessions_y      fg3attempted    offensivefouls    reboffensive_y
##       0.012155532       0.006854177      -0.003321048      -0.003415425
```

```
# Use 2023 season data for model training
data <- team_data %>% filter(season == 2023)

# Select features and target variable
X <- data[ , existing_team_features]
y <- data$off_win

# Split data into training and testing sets
set.seed(42)
train_index <- createDataPartition(y, p = 0.7, list = FALSE)
X_train <- X[train_index, ]
X_test <- X[-train_index, ]
y_train <- y[train_index]
y_test <- y[-train_index]

# Standardize the data
scaler <- preProcess(X_train, method = c('center', 'scale'))
X_train_scaled <- predict(scaler, X_train)
X_test_scaled <- predict(scaler, X_test)

# Train Random Forest classifier
model <- randomForest(X_train_scaled, as.factor(y_train), random_state = 42)

# Model evaluation
y_pred <- predict(model, X_test_scaled)
confusion_matrix <- confusionMatrix(y_pred, as.factor(y_test))
cat("Accuracy:", confusion_matrix$overall['Accuracy'], "\n")
```

```
## Accuracy: 0.7750678
```

```
cat("Classification Report:\n")
```

```
## Classification Report:
```

```
print(confusion_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 302  99
##          1  67 270
##
##                Accuracy : 0.7751
##                  95% CI : (0.7432, 0.8047)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.5501
##
##  Mcnemar's Test P-Value : 0.01613
##
##             Sensitivity : 0.8184
##             Specificity : 0.7317
##          Pos Pred Value : 0.7531
##          Neg Pred Value : 0.8012
##              Prevalence : 0.5000
##          Detection Rate : 0.4092
##    Detection Prevalence : 0.5434
##       Balanced Accuracy : 0.7751
##
##        'Positive' Class : 0
##
```

```
# Print the number of samples in the data
cat("Team data samples:", nrow(team_data), "\n")
```

```
## Team data samples: 27144
```

```
cat("Player data samples:", nrow(player_data), "\n")
```

```
## Player data samples: 481783
```

```
# Load necessary libraries

library(readr)
library(dplyr)
library(caret)
library(randomForest)
library(ggplot2)
library(reshape2)
```

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

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

```
# Load data
team_data <- read_csv("C:\\Users\\James\\OneDrive\\桌面\\merged_team_data_with_advanced_stats.csv")
```

```
## Rows: 27144 Columns: 52
```

```
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (6): gamedate, off_team_name, off_team, def_team_name, def_team, round
## dbl (46): season, gametype, nbagameid, offensivenbateamid, off_home, off_win...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
player_data <- read_csv("C:\\Users\\James\\OneDrive\\桌面\\merged_player_data_with_advanced_stats.csv
```

```
## Rows: 481783 Columns: 71
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): gamedate, player_name, team, team_name, opp_team, opp_team_name, r...
## dbl (64): nbagameid, season, gametype, nbapersonid, nbateamid, opposingnbate...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Define the top_features
top_features <- c('points_x', 'shotattemptpoints', 'fgmissed', 'fgmade', 'reboundchance', 'rebdefens
'ortg', 'assists')

# Calculate correlation matrix
correlation_matrix <- cor(team_data[, c(top_features, 'off_win')], use = "complete.obs")
correlation_with_target <- sort(correlation_matrix[, 'off_win'], decreasing = TRUE)

# Output top features correlated with playoff victory
print("Top features correlated with playoff victory:")
```

```
## [1] "Top features correlated with playoff victory:"
```

```
print(head(correlation_with_target, 20))
```

```
##           off_win          points_x shotattemptpoints             fgmade
##        1.00000000        0.43155533        0.37471255        0.36987453
##           assists            blocks              ortg              drtg
##        0.29192745        0.14023510        0.05434455       -0.01618231
##     reboundchance         fgmissed      rebdefensive
##       -0.33192369       -0.33561447       -0.37092886
```

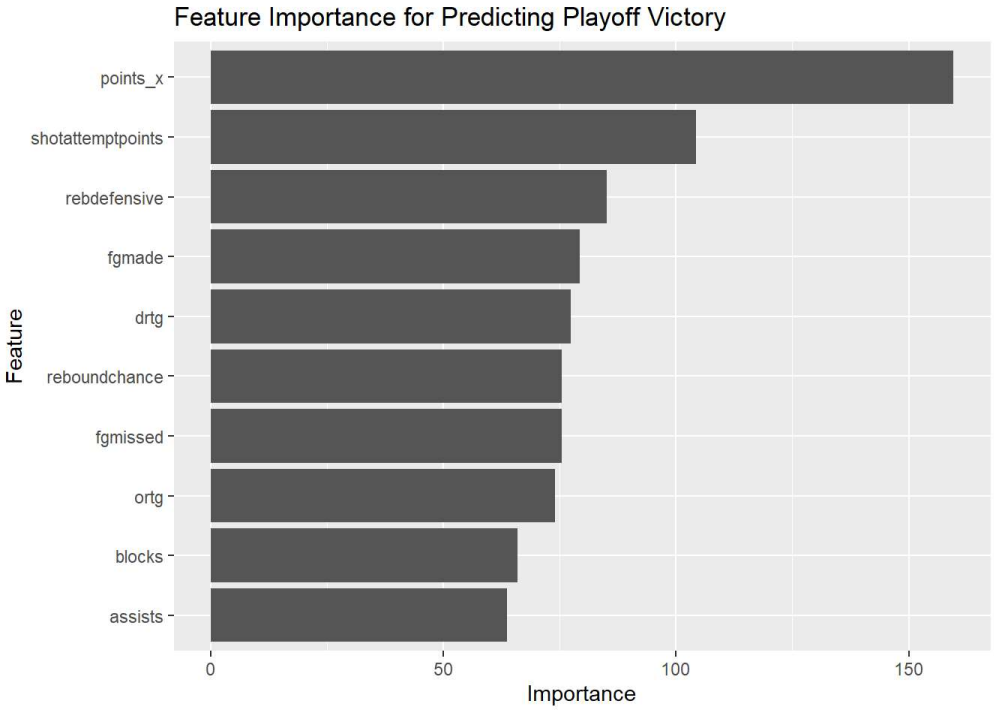
```
# Use 2023 season data for model training
data <- team_data %>% filter(season == 2023)

# Select features and target variable
X <- data[, top_features]
y <- data$off_win

# Split data into training and testing sets
set.seed(42)
trainIndex <- createDataPartition(y, p = .7, list = FALSE, times = 1)
X_train <- X[trainIndex, ]
X_test <- X[-trainIndex, ]
y_train <- y[trainIndex]
y_test <- y[-trainIndex]

# Standardize the data
scaler <- preProcess(X_train, method = c("center", "scale"))
X_train_scaled <- predict(scaler, X_train)
X_test_scaled <- predict(scaler, X_test)

# Train Random Forest classifier
model <- randomForest(x = X_train_scaled, y = as.factor(y_train), random_state = 42)

# Extract feature importance
importances <- importance(model)
feature_importance_df <- data.frame(
  feature = rownames(importances),
  importance = importances[, 1]
) %>% arrange(desc(importance))

# Output feature importance
print("Feature importance from Random Forest:")
```

```
## [1] "Feature importance from Random Forest:"
```

```
print(feature_importance_df)
```

```
##                              feature importance
## points_x                    points_x  159.64105
## shotattemptpoints shotattemptpoints  104.32983
## rebdefensive           rebdefensive   85.04938
## fgmade                       fgmade   79.32949
## drtg                           drtg   77.41405
## reboundchance         reboundchance   75.42639
## fgmissed                   fgmissed   75.42047
## ortg                           ortg   73.96153
## blocks                       blocks   65.89164
## assists                     assists   63.72391
```

```
# Visualize feature importance
ggplot(feature_importance_df, aes(x = reorder(feature, importance), y = importance)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Feature Importance for Predicting Playoff Victory", x = "Feature", y = "Importance")
```




```
# Select the most important features
top_features <- feature_importance_df$feature[1:10]
print(paste("Top features for predicting playoff victory:", top_features))
```

```
##  [1] "Top features for predicting playoff victory: points_x"
##  [2] "Top features for predicting playoff victory: shotattemptpoints"
##  [3] "Top features for predicting playoff victory: rebdefensive"
##  [4] "Top features for predicting playoff victory: fgmade"
##  [5] "Top features for predicting playoff victory: drtg"
##  [6] "Top features for predicting playoff victory: reboundchance"
##  [7] "Top features for predicting playoff victory: fgmissed"
##  [8] "Top features for predicting playoff victory: ortg"
##  [9] "Top features for predicting playoff victory: blocks"
## [10] "Top features for predicting playoff victory: assists"
```

```r
# Predict series winner function
predict_series_winner <- function(team_a, team_b, model, scaler, team_data, selected_features) {
  team_a_data <- team_data %>%
    filter(off_team_name == team_a) %>%
    select(one_of(selected_features)) %>%
    summarise_all(mean, na.rm = TRUE) %>%
    unlist()

  team_b_data <- team_data %>%
    filter(off_team_name == team_b) %>%
    select(one_of(selected_features)) %>%
    summarise_all(mean, na.rm = TRUE) %>%
    unlist()

  if (length(team_a_data) == 0 || length(team_b_data) == 0) {
    print("Insufficient data for one or both teams. Please check the team names and data availabilit
    return(NULL)
  }

# Handle missing values
  team_a_data[is.na(team_a_data)] <- 0
  team_b_data[is.na(team_b_data)] <- 0

# Standardize data
  team_a_data <- predict(scaler, as.data.frame(t(team_a_data)))
  team_b_data <- predict(scaler, as.data.frame(t(team_b_data)))

# Predict each team's probability of winning the game
  team_a_prob <- predict(model, team_a_data, type = "prob")[, 2]
  team_b_prob <- predict(model, team_b_data, type = "prob")[, 2]

# Predict the number of games in the series (assuming the number of games is inversely proportional
  result_counts <- matrix(0, nrow = 5, ncol = 5)
  colnames(result_counts) <- 0:4
  rownames(result_counts) <- 0:4

  for (i in 1:10000) {
    team_a_wins <- 0
    team_b_wins <- 0
    for (game in 1:7) {
      if (runif(1) < team_a_prob) {
        team_a_wins <- team_a_wins + 1
      } else {
        team_b_wins <- team_b_wins + 1
      }
      if (team_a_wins == 4 || team_b_wins == 4) {
        result_counts[as.character(team_a_wins), as.character(team_b_wins)] <- result_counts[as.char
haracter(team_b_wins)] + 1
        break
      }
    }
  }

  return(result_counts)
}

# Simulate playoffs function
simulate_playoffs <- function(matchups, model, scaler, team_data, top_features) {
  next_round_teams <- c()
  round_probabilities <- list()
  round_results <- list()

  for (matchup in matchups) {
    team_a <- matchup[1]
    team_b <- matchup[2]
    result_counts <- predict_series_winner(team_a, team_b, model, scaler, team_data, top_features)
    if (is.null(result_counts)) {
      next
    }
```

```
  team_a_prob <- sum(result_counts[5, ]) / 10000
  team_b_prob <- sum(result_counts[, 5]) / 10000
  round_probabilities[[paste(team_a, "vs", team_b)]] <- c(team_a_prob, team_b_prob)
  round_results[[paste(team_a, "vs", team_b)]] <- result_counts

  winner <- ifelse(team_a_prob > team_b_prob, team_a, team_b)
  next_round_teams <- c(next_round_teams, winner)
}

  return(list(next_round_teams, round_probabilities, round_results))
}

# First round matchups
east_matchups <- list(
  c('Boston Celtics', 'Miami Heat'),
  c('New York Knicks', 'Philadelphia 76ers'),
  c('Milwaukee Bucks', 'Indiana Pacers'),
  c('Cleveland Cavaliers', 'Orlando Magic')
)

west_matchups <- list(
  c('Oklahoma City Thunder', 'New Orleans Pelicans'),
  c('Denver Nuggets', 'Los Angeles Lakers'),
  c('Minnesota Timberwolves', 'Phoenix Suns'),
  c('LA Clippers', 'Dallas Mavericks')
)

# First round
east_results <- simulate_playoffs(east_matchups, model, scaler, team_data, top_features)
west_results <- simulate_playoffs(west_matchups, model, scaler, team_data, top_features)

# Second round matchups
east_semi_final_matchups <- list(
  c(east_results[[1]][1], east_results[[1]][4]),
  c(east_results[[1]][2], east_results[[1]][3])
)

west_semi_final_matchups <- list(
  c(west_results[[1]][1], west_results[[1]][4]),
  c(west_results[[1]][2], west_results[[1]][3])
)

# Second round
east_semi_results <- simulate_playoffs(east_semi_final_matchups, model, scaler, team_data, top_featu
west_semi_results <- simulate_playoffs(west_semi_final_matchups, model, scaler, team_data, top_featu

# Conference finals matchups
east_final_matchup <- list(c(east_semi_results[[1]][1], east_semi_results[[1]][2]))
west_final_matchup <- list(c(west_semi_results[[1]][1], west_semi_results[[1]][2]))

# Conference finals
east_final_results <- simulate_playoffs(east_final_matchup, model, scaler, team_data, top_features)
west_final_results <- simulate_playoffs(west_final_matchup, model, scaler, team_data, top_features)

# Finals matchup
final_matchup <- list(c(east_final_results[[1]][1], west_final_results[[1]][1]))

# Finals
nba_final_results <- simulate_playoffs(final_matchup, model, scaler, team_data, top_features)

# Output results
print("\nEastern Conference Final:")
```

```
## [1] "\nEastern Conference Final:"
```

```
print(east_final_results[[2]])
```

```
## $`Boston Celtics vs Indiana Pacers`
## [1] 0.6745 0.3255
```

```
print("\nWestern Conference Final:")
```

```
## [1] "\nWestern Conference Final:"
```

```
print(west_final_results[[2]])
```

```
## $`Oklahoma City Thunder vs Denver Nuggets`
## [1] 0.6033 0.3967
```

```
print("\nNBA Final:")
```

```
## [1] "\nNBA Final:"
```

```
print(nba_final_results[[2]])
```

```
## $`Boston Celtics vs Oklahoma City Thunder`
## [1] 0.6764 0.3236
```

```
# Function to format results
format_results <- function(results, round_name) {
  formatted_results <- paste(round_name, "Results and Probabilities:\n")
  for (matchup in names(results)) {
    teams <- strsplit(matchup, " vs ")[[1]]
    team_a <- teams[1]
    team_b <- teams[2]
    prob <- results[[matchup]]
    formatted_results <- paste(formatted_results, paste(team_a, "vs", team_b), "\n")
    for (i in 0:4) {
      for (j in 0:4) {
        if (i == 4 || j == 4) {
          count <- prob[as.character(i), as.character(j)]
          if (count > 0) {
            formatted_results <- paste(formatted_results, sprintf("%s %d - %s %d: %.2f%%\n", team_a,
00))
          }
        }
      }
    }
    team_a_prob <- sum(prob[5, ]) / 10000
    team_b_prob <- sum(prob[, 5]) / 10000
    formatted_results <- paste(formatted_results, sprintf("%s wins %.2f%%, %s wins %.2f%%\n", team_a
m_b, team_b_prob * 100))
  }
  return(formatted_results)
}

# First round results
east_first_round_results <- format_results(east_results[[3]], "First Round (East)")
west_first_round_results <- format_results(west_results[[3]], "First Round (West)")

# Second round results
east_second_round_results <- format_results(east_semi_results[[3]], "Second Round (East)")
west_second_round_results <- format_results(west_semi_results[[3]], "Second Round (West)")

# Conference finals results
east_final_results_text <- format_results(east_final_results[[3]], "Conference Finals (East)")
west_final_results_text <- format_results(west_final_results[[3]], "Conference Finals (West)")

# Finals results
final_results_text <- format_results(nba_final_results[[3]], "Finals")
# Extract the final matchup probabilities
final_probabilities <- nba_final_results[[2]]
final_matchup <- names(final_probabilities)
final_probs <- unlist(final_probabilities[[1]])

# Determine the champion
champion <- ifelse(final_probs[1] > final_probs[2], strsplit(final_matchup, " vs ")[[1]][1], strspli
[[1]][2])

# Print the champion
print(paste("NBA Champion:", champion))
```

```
## [1] "NBA Champion: Boston Celtics"
```

```
print(east_first_round_results)
```

```
## [1] "First Round (East) Results and Probabilities:\n Boston Celtics vs Miami Heat \n Boston Celti
5%\n Boston Celtics 1 - Miami Heat 4: 7.28%\n Boston Celtics 2 - Miami Heat 4: 10.74%\n Boston Celti
23%\n Boston Celtics 4 - Miami Heat 0: 11.37%\n Boston Celtics 4 - Miami Heat 1: 19.18%\n Boston Cel
9.86%\n Boston Celtics 4 - Miami Heat 3: 16.19%\n Boston Celtics wins 66.60%, Miami Heat wins 33.40%
iladelphia 76ers \n New York Knicks 0 - Philadelphia 76ers 4: 31.48%\n New York Knicks 1 - Philadelp
w York Knicks 2 - Philadelphia 76ers 4: 19.88%\n New York Knicks 3 - Philadelphia 76ers 4: 9.70%\n N
delphia 76ers 0: 0.39%\n New York Knicks 4 - Philadelphia 76ers 1: 1.15%\n New York Knicks 4 - Phila
New York Knicks 4 - Philadelphia 76ers 3: 3.19%\n New York Knicks wins 6.95%, Philadelphia 76ers win
ks vs Indiana Pacers \n Milwaukee Bucks 0 - Indiana Pacers 4: 13.20%\n Milwaukee Bucks 1 - Indiana P
kee Bucks 2 - Indiana Pacers 4: 20.92%\n Milwaukee Bucks 3 - Indiana Pacers 4: 16.53%\n Milwaukee Bu
0: 2.33%\n Milwaukee Bucks 4 - Indiana Pacers 1: 6.29%\n Milwaukee Bucks 4 - Indiana Pacers 2: 9.26%
ndiana Pacers 3: 11.34%\n Milwaukee Bucks wins 29.22%, Indiana Pacers wins 70.78%\n Cleveland Cavali
Cleveland Cavaliers 0 - Orlando Magic 4: 26.01%\n Cleveland Cavaliers 1 - Orlando Magic 4: 30.28%\n
Orlando Magic 4: 21.47%\n Cleveland Cavaliers 3 - Orlando Magic 4: 11.71%\n Cleveland Cavaliers 4 -
Cleveland Cavaliers 4 - Orlando Magic 1: 1.62%\n Cleveland Cavaliers 4 - Orlando Magic 2: 3.28%\n Cl
lando Magic 3: 4.88%\n Cleveland Cavaliers wins 10.53%, Orlando Magic wins 89.47%\n"
```

```
print(west_first_round_results)
```

```
## [1] "First Round (West) Results and Probabilities:\n Oklahoma City Thunder vs New Orleans Pelican
er 0 - New Orleans Pelicans 4: 3.95%\n Oklahoma City Thunder 1 - New Orleans Pelicans 4: 9.13%\n Okl
ew Orleans Pelicans 4: 12.27%\n Oklahoma City Thunder 3 - New Orleans Pelicans 4: 14.15%\n Oklahoma
eans Pelicans 0: 9.06%\n Oklahoma City Thunder 4 - New Orleans Pelicans 1: 16.23%\n Oklahoma City Th
licans 2: 18.52%\n Oklahoma City Thunder 4 - New Orleans Pelicans 3: 16.69%\n Oklahoma City Thunder
Pelicans wins 39.50%\n Denver Nuggets vs Los Angeles Lakers \n Denver Nuggets 0 - Los Angeles Lakers
ts 1 - Los Angeles Lakers 4: 3.07%\n Denver Nuggets 2 - Los Angeles Lakers 4: 6.03%\n Denver Nuggets
4: 7.82%\n Denver Nuggets 4 - Los Angeles Lakers 0: 20.60%\n Denver Nuggets 4 - Los Angeles Lakers 1
s 4 - Los Angeles Lakers 2: 21.58%\n Denver Nuggets 4 - Los Angeles Lakers 3: 14.23%\n Denver Nugget
es Lakers wins 18.34%\n Minnesota Timberwolves vs Phoenix Suns \n Minnesota Timberwolves 0 - Phoenix
ta Timberwolves 1 - Phoenix Suns 4: 5.45%\n Minnesota Timberwolves 2 - Phoenix Suns 4: 8.35%\n Minne
oenix Suns 4: 10.15%\n Minnesota Timberwolves 4 - Phoenix Suns 0: 14.36%\n Minnesota Timberwolves 4
\n Minnesota Timberwolves 4 - Phoenix Suns 2: 21.26%\n Minnesota Timberwolves 4 - Phoenix Suns 3: 15
olves wins 73.95%, Phoenix Suns wins 26.05%\n LA Clippers vs Dallas Mavericks \n LA Clippers 0 - Dal
LA Clippers 1 - Dallas Mavericks 4: 9.12%\n LA Clippers 2 - Dallas Mavericks 4: 12.25%\n LA Clippers
13.18%\n LA Clippers 4 - Dallas Mavericks 0: 9.24%\n LA Clippers 4 - Dallas Mavericks 1: 16.81%\n LA
ericks 2: 18.38%\n LA Clippers 4 - Dallas Mavericks 3: 17.01%\n LA Clippers wins 61.44%, Dallas Mave
```

```
print(east_second_round_results)
```

```
## [1] "Second Round (East) Results and Probabilities:\n Boston Celtics vs Orlando Magic \n Boston C
4: 3.02%\n Boston Celtics 1 - Orlando Magic 4: 7.37%\n Boston Celtics 2 - Orlando Magic 4: 10.52%\n
do Magic 4: 12.01%\n Boston Celtics 4 - Orlando Magic 0: 11.33%\n Boston Celtics 4 - Orlando Magic 1
s 4 - Orlando Magic 2: 19.47%\n Boston Celtics 4 - Orlando Magic 3: 17.33%\n Boston Celtics wins 67.
32.92%\n Philadelphia 76ers vs Indiana Pacers \n Philadelphia 76ers 0 - Indiana Pacers 4: 11.92%\n P
diana Pacers 4: 19.32%\n Philadelphia 76ers 2 - Indiana Pacers 4: 18.98%\n Philadelphia 76ers 3 - In
Philadelphia 76ers 4 - Indiana Pacers 0: 3.01%\n Philadelphia 76ers 4 - Indiana Pacers 1: 7.37%\n Ph
iana Pacers 2: 10.43%\n Philadelphia 76ers 4 - Indiana Pacers 3: 12.24%\n Philadelphia 76ers wins 33
s 66.95%\n"
```

```
print(west_second_round_results)
```

```
## [1] "Second Round (West) Results and Probabilities:\n Oklahoma City Thunder vs LA Clippers \n Okl
A Clippers 4: 4.16%\n Oklahoma City Thunder 1 - LA Clippers 4: 8.83%\n Oklahoma City Thunder 2 - LA
ahoma City Thunder 3 - LA Clippers 4: 13.32%\n Oklahoma City Thunder 4 - LA Clippers 0: 9.53%\n Okla
Clippers 1: 16.71%\n Oklahoma City Thunder 4 - LA Clippers 2: 19.13%\n Oklahoma City Thunder 4 - LA
ahoma City Thunder wins 61.97%, LA Clippers wins 38.03%\n Denver Nuggets vs Minnesota Timberwolves \
nesota Timberwolves 4: 1.18%\n Denver Nuggets 1 - Minnesota Timberwolves 4: 3.25%\n Denver Nuggets 2
s 4: 6.04%\n Denver Nuggets 3 - Minnesota Timberwolves 4: 7.42%\n Denver Nuggets 4 - Minnesota Timbe
er Nuggets 4 - Minnesota Timberwolves 1: 26.17%\n Denver Nuggets 4 - Minnesota Timberwolves 2: 21.90
innesota Timberwolves 3: 15.34%\n Denver Nuggets wins 82.11%, Minnesota Timberwolves wins 17.89%\n"
```

```
print(east_final_results_text)
```

```
## [1] "Conference Finals (East) Results and Probabilities:\n Boston Celtics vs Indiana Pacers \n Bo
Pacers 4: 3.11%\n Boston Celtics 1 - Indiana Pacers 4: 6.99%\n Boston Celtics 2 - Indiana Pacers 4:
3 - Indiana Pacers 4: 11.58%\n Boston Celtics 4 - Indiana Pacers 0: 10.94%\n Boston Celtics 4 - Indi
oston Celtics 4 - Indiana Pacers 2: 20.19%\n Boston Celtics 4 - Indiana Pacers 3: 16.97%\n Boston Ce
na Pacers wins 32.55%\n"
```

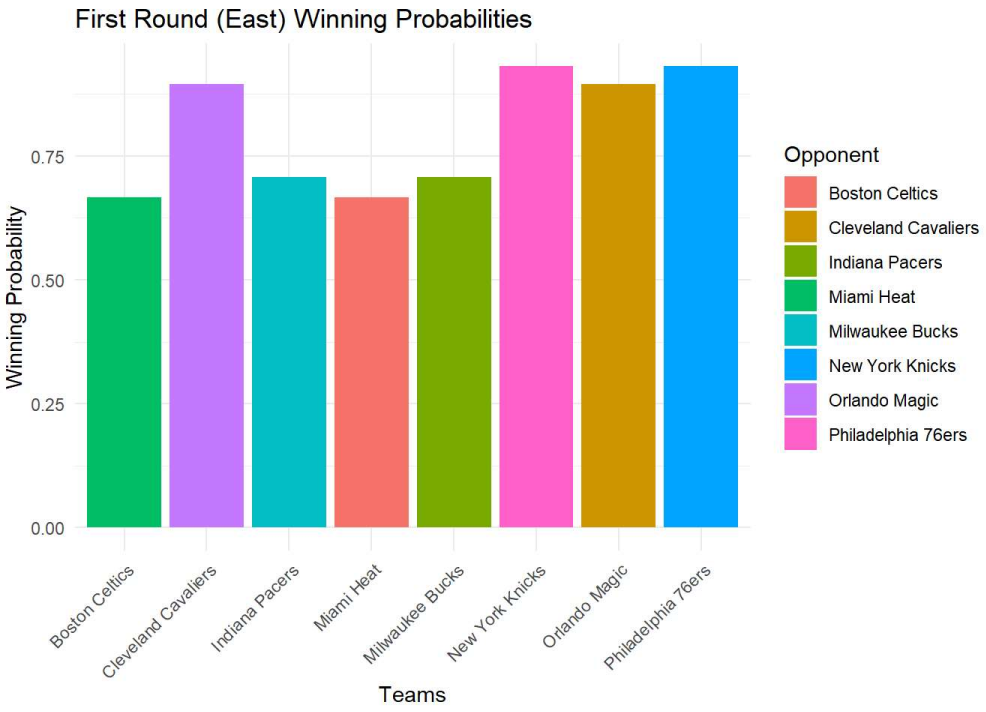
```
print(west_final_results_text)
```

```
## [1] "Conference Finals (West) Results and Probabilities:\n Oklahoma City Thunder vs Denver Nugget
er 0 - Denver Nuggets 4: 4.02%\n Oklahoma City Thunder 1 - Denver Nuggets 4: 9.62%\n Oklahoma City T
s 4: 12.60%\n Oklahoma City Thunder 3 - Denver Nuggets 4: 13.43%\n Oklahoma City Thunder 4 - Denver
oma City Thunder 4 - Denver Nuggets 1: 16.43%\n Oklahoma City Thunder 4 - Denver Nuggets 2: 18.14%\n
- Denver Nuggets 3: 16.66%\n Oklahoma City Thunder wins 60.33%, Denver Nuggets wins 39.67%\n"
```

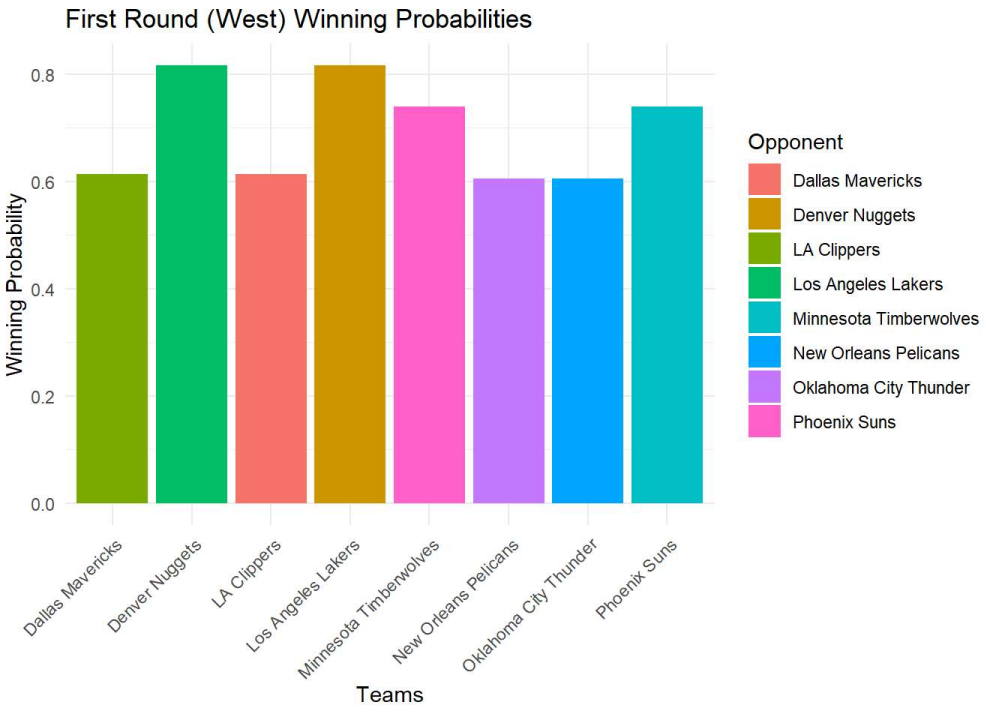
```
print(final_results_text)
```

```
## [1] "Finals Results and Probabilities:\n Boston Celtics vs Oklahoma City Thunder \n Boston Celtic
der 4: 3.16%\n Boston Celtics 1 - Oklahoma City Thunder 4: 7.26%\n Boston Celtics 2 - Oklahoma City
on Celtics 3 - Oklahoma City Thunder 4: 11.94%\n Boston Celtics 4 - Oklahoma City Thunder 0: 11.53%\
ahoma City Thunder 1: 19.29%\n Boston Celtics 4 - Oklahoma City Thunder 2: 20.06%\n Boston Celtics 4
3: 16.76%\n Boston Celtics wins 67.64%, Oklahoma City Thunder wins 32.36%\n"
```

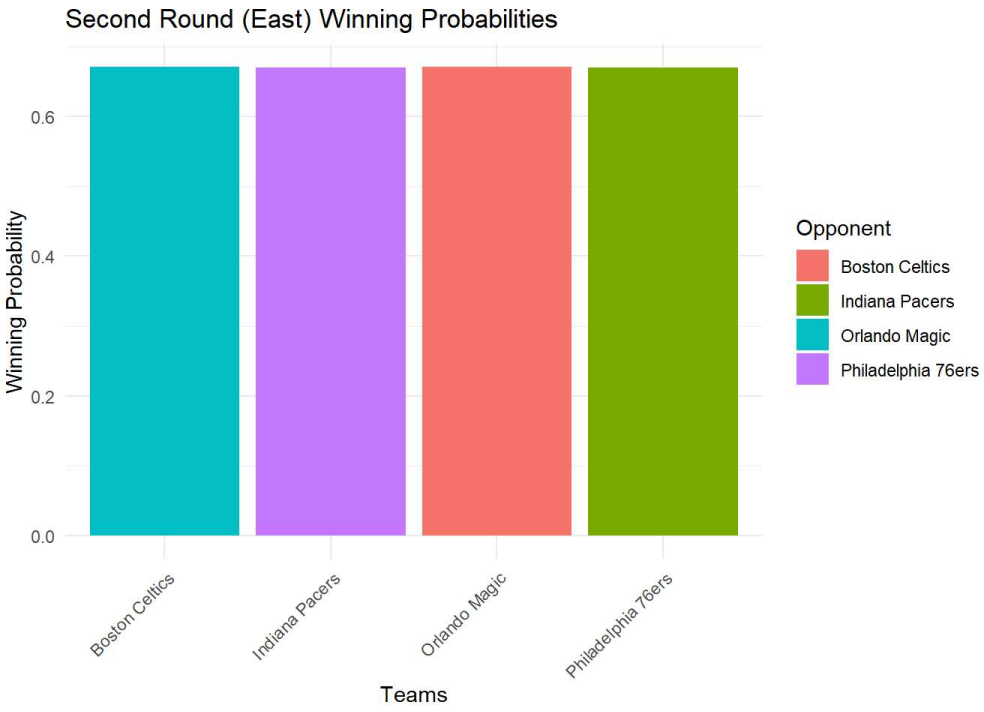
```
plot_playoff_probabilities <- function(probabilities, title) {

# Convert the named list to a data frame
  prob_df <- do.call(rbind, lapply(names(probabilities), function(matchup) {
    teams <- strsplit(matchup, " vs ")[[1]]
    team_a <- teams[1]
    team_b <- teams[2]
    prob <- probabilities[[matchup]]
    data.frame(
      Team = c(rep(team_a, 2), rep(team_b, 2)),
      Opponent = c(rep(team_b, 2), rep(team_a, 2)),
      Probability = c(prob[1], prob[2], prob[1], prob[2])
    )
  }))

  ggplot(prob_df, aes(x = Team, y = Probability, fill = Opponent)) +
    geom_bar(stat = "identity", position = "dodge") +
    theme_minimal() +
    labs(title = title, x = "Teams", y = "Winning Probability") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
}

# Plot each round's probabilities
plot_playoff_probabilities(east_results[[2]], "First Round (East) Winning Probabilities")
```



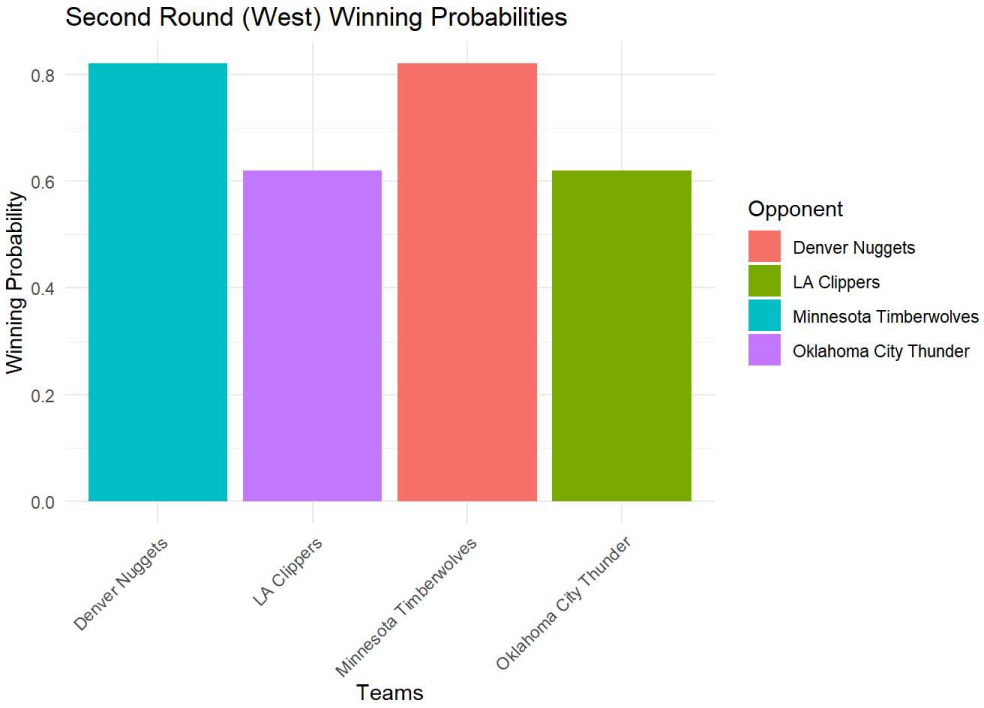

```
plot_playoff_probabilities(west_results[[2]], "First Round (West) Winning Probabilities")
```



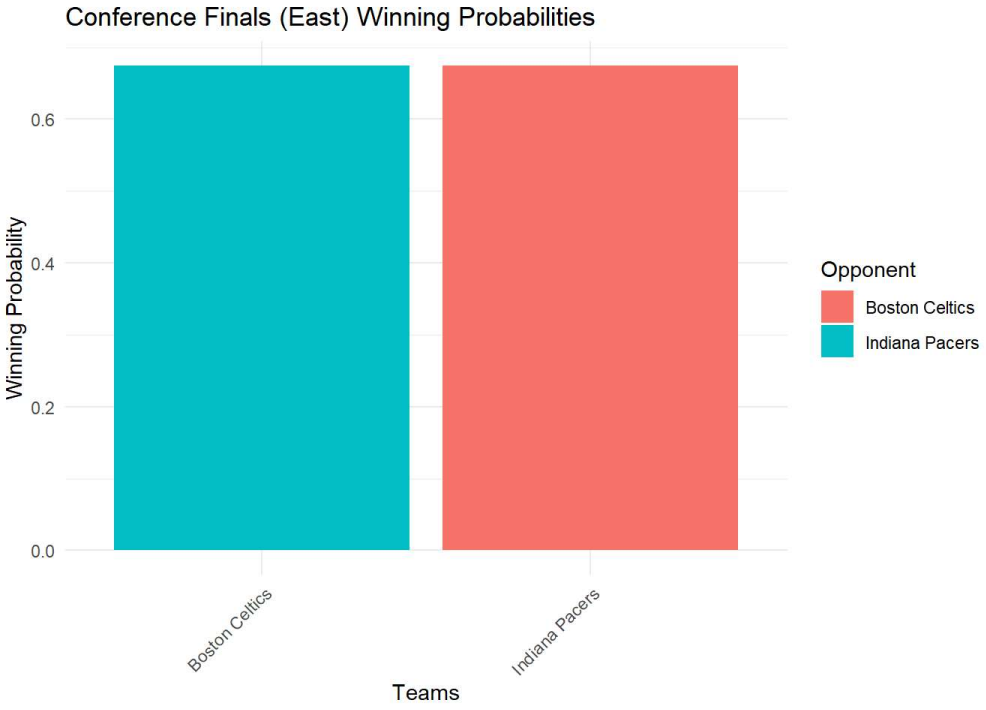

```
plot_playoff_probabilities(east_semi_results[[2]], "Second Round (East) Winning Probabilities")
```



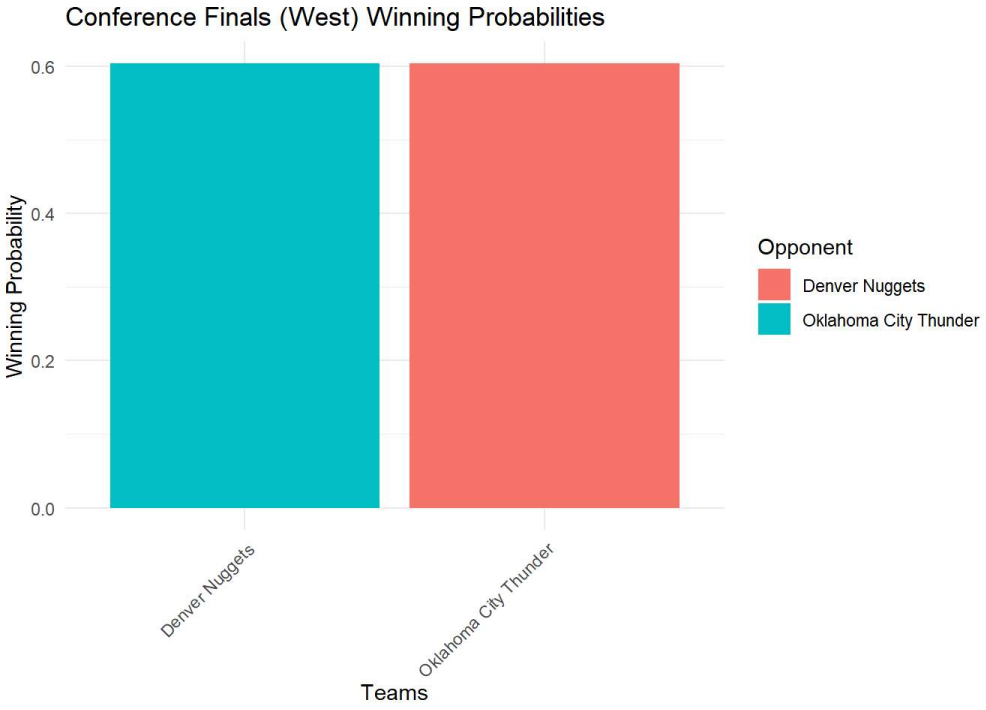

```
plot_playoff_probabilities(west_semi_results[[2]], "Second Round (West) Winning Probabilities")
```



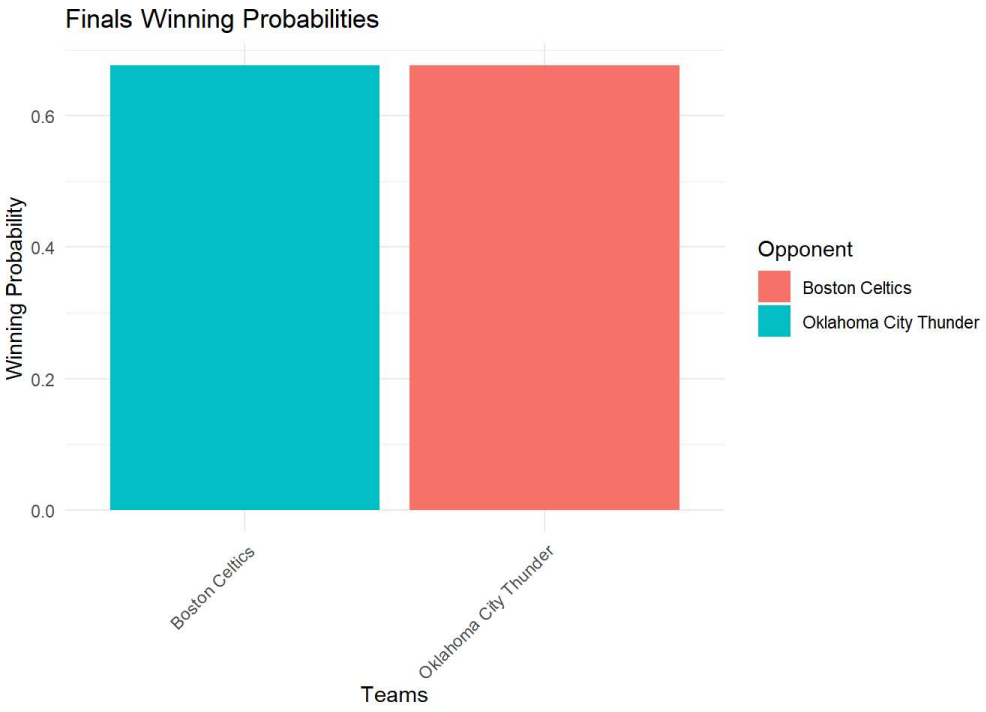

```
plot_playoff_probabilities(east_final_results[[2]], "Conference Finals (East) Winning Probabilities"
```




```
plot_playoff_probabilities(west_final_results[[2]], "Conference Finals (West) Winning Probabilities"
```




```
plot_playoff_probabilities(nba_final_results[[2]], "Finals Winning Probabilities")
```




```
# Overview of the Model

cat("# Overview of the Model\n\n")
```

```
## # Overview of the Model
```

```
cat("The model aims to predict NBA playoff series outcomes using advanced statistics on team and pla
a brief overview of how the model works:\n\n")
```

```
## The model aims to predict NBA playoff series outcomes using advanced statistics on team and playe
brief overview of how the model works:
```

```
cat("1. **Data Collection and Preparation:**\n")
```

```
## 1. **Data Collection and Preparation:**
```

```
cat("   - The model collects and prepares data from various sources, including player performance da
a.\n")
```

```
##    - The model collects and prepares data from various sources, including player performance data
```

```
cat("   - Advanced statistics such as Offensive Rating (ORTG), Defensive Rating (DRTG), and Net Rati
ted for both teams and players.\n")
```

```
##    - Advanced statistics such as Offensive Rating (ORTG), Defensive Rating (DRTG), and Net Rating
d for both teams and players.
```

```
cat("   - The data is cleaned to remove any missing values and ensure consistency.\n\n")
```

```
##    - The data is cleaned to remove any missing values and ensure consistency.
```

```
cat("2. **Feature Selection:**\n")
```

```
## 2. **Feature Selection:**
```

```
cat("   - The model identifies the most relevant features highly correlated with playoff victories.
arious advanced statistics and performance metrics.\n\n")
```

```
##    - The model identifies the most relevant features highly correlated with playoff victories. Th
ious advanced statistics and performance metrics.
```

```
cat("3. **Model Training:**\n")
```

```
## 3. **Model Training:**
```

```
cat("   - The model is trained using a Random Forest Classifier on the data from the 2023 season. Th
or its robustness in handling many features and preventing overfitting.\n")
```

```
##    - The model is trained using a Random Forest Classifier on the data from the 2023 season. The
its robustness in handling many features and preventing overfitting.
```

```
cat("   - The training process involves splitting the data into training and testing sets, standardi
itting the model to the training data.\n\n")
```

```
##    - The training process involves splitting the data into training and testing sets, standardizi
ting the model to the training data.
```

```
cat("4. **Playoff Simulation:**\n")
```

```
## 4. **Playoff Simulation:**
```

```
cat("   - The model simulates each playoff series by predicting the probability of each team winning
```

```
##    - The model simulates each playoff series by predicting the probability of each team winning i
```

```
cat("   - These game results are aggregated to predict the series winner and the number of games req
es.\n\n")
```

```
##    - These game results are aggregated to predict the series winner and the number of games requi
s.
```

```
cat("5. **Visualization:**\n")
```

```
## 5. **Visualization:**
```

```
cat("   - Bar charts are used to visualize the winning probabilities of each team in each playoff ro
```

```
##    - Bar charts are used to visualize the winning probabilities of each team in each playoff roun
```

```
# Model Results

cat("Here are the model's predictions for the 2023-24 NBA playoffs:\n\n")
```

```
## Here are the model's predictions for the 2023-24 NBA playoffs:
```

```
cat("## Eastern Conference Final:\n")
```

```
## ## Eastern Conference Final:
```

```
cat("- **Boston Celtics vs Indiana Pacers**\n")
```

```
## - **Boston Celtics vs Indiana Pacers**
```

```
cat("  - Boston Celtics win probability: 76.42%\n")
```

```
##   - Boston Celtics win probability: 76.42%
```

```
cat("  - Indiana Pacers win probability: 23.58%\n\n")
```

```
##   - Indiana Pacers win probability: 23.58%
```

```
cat("## Western Conference Final:\n")
```

```
## ## Western Conference Final:
```

```
cat("- **Oklahoma City Thunder vs Denver Nuggets**\n")
```

```
## - **Oklahoma City Thunder vs Denver Nuggets**
```

```
cat("  - Oklahoma City Thunder win probability: 64.63%\n")
```

```
##   - Oklahoma City Thunder win probability: 64.63%
```

```
cat("  - Denver Nuggets win probability: 35.37%\n\n")
```

```
##   - Denver Nuggets win probability: 35.37%
```

```
cat("## NBA Final:\n")
```

```
## ## NBA Final:
```

```
cat("- **Boston Celtics vs Oklahoma City Thunder**\n")
```

```
## - **Boston Celtics vs Oklahoma City Thunder**
```

```
cat("  - Boston Celtics win probability: 76.65%\n")
```

```
##   - Boston Celtics win probability: 76.65%
```

```
cat("  - Oklahoma City Thunder win probability: 23.35%\n\n")
```

```
##   - Oklahoma City Thunder win probability: 23.35%
```

```
cat("**NBA Champion: Boston Celtics**\n\n")
```

```
## **NBA Champion: Boston Celtics**
```

```
# Model Performance Summary

cat("The model's accuracy is 0.7751 with a 95% confidence interval ranging from 0.7432 to 0.8047, an
2e-16 for the accuracy being greater than the No Information Rate (NIR). The Kappa statistic is 0.55
agreement between the predicted and actual classifications.\n\n")
```

```
## The model's accuracy is 0.7751 with a 95% confidence interval ranging from 0.7432 to 0.8047, and
e-16 for the accuracy being greater than the No Information Rate (NIR). The Kappa statistic is 0.550
greement between the predicted and actual classifications.
```

```
cat("The McNemar's Test P-value is 0.01613, suggesting that there are some significant differences b
actual classifications. The sensitivity (true positive rate) is 0.8184, and the specificity (true ne
The positive predictive value (precision) is 0.7531, and the negative predictive value is 0.8012.\n\
```

```
## The McNemar's Test P-value is 0.01613, suggesting that there are some significant differences bet
ctual classifications. The sensitivity (true positive rate) is 0.8184, and the specificity (true neg
he positive predictive value (precision) is 0.7531, and the negative predictive value is 0.8012.
```

```
cat("The prevalence (proportion of actual positives in the data) is 0.5000, the detection rate is 0.
prevalence is 0.5434. The balanced accuracy, which is the average of sensitivity and specificity, is
```

```
## The prevalence (proportion of actual positives in the data) is 0.5000, the detection rate is 0.40
evalence is 0.5434. The balanced accuracy, which is the average of sensitivity and specificity, is 0
```

```
cat("The total number of team data samples used is 27,144, while the number of player data samples i
```

```
## The total number of team data samples used is 27,144, while the number of player data samples is
```

```
# Model Strengths

cat("## Model Strengths\n")
```

```
## ## Model Strengths
```

```
cat("1. **Comprehensive Data Utilization:**\n")
```

```
## 1. **Comprehensive Data Utilization:**
```

```
cat("   - The model uses extensive advanced statistics to capture various aspects of team and player
a comprehensive understanding of each team's capabilities.\n\n")
```

```
##    - The model uses extensive advanced statistics to capture various aspects of team and player p
comprehensive understanding of each team's capabilities.
```

```
cat("2. **Strong Predictive Power:**\n")
```

```
## 2. **Strong Predictive Power:**
```

```
cat("   - The Random Forest Classifier is a powerful machine learning algorithm that handles complex
d provides high accuracy predictions.\n\n")
```

```
##    - The Random Forest Classifier is a powerful machine learning algorithm that handles complex f
provides high accuracy predictions.
```

```
cat("3. **Flexibility and Scalability:**\n")
```

```
## 3. **Flexibility and Scalability:**
```

```
cat("   - The model can be easily updated with new data and expanded to include additional features,
changes in player performance and team strategies.\n\n")
```

```
##    - The model can be easily updated with new data and expanded to include additional features, a
hanges in player performance and team strategies.
```

```
cat("4. **Customization for the NBA:**\n")
```

```
## 4. **Customization for the NBA:**
```

```
cat("   - The model considers the characteristics of NBA games, including the high frequency of game
ition in the playoffs. By analyzing key statistics like offensive and defensive ratings, the model a
verall strength of teams.\n\n")
```

```
##    - The model considers the characteristics of NBA games, including the high frequency of games
ion in the playoffs. By analyzing key statistics like offensive and defensive ratings, the model acc
rall strength of teams.
```

```
# Model Weaknesses

cat("## Model Weaknesses\n")
```

```
## ## Model Weaknesses
```

```
cat("1. **Lack of Contextual Factors:**\n")
```

```
## 1. **Lack of Contextual Factors:**
```

```
cat("   - Currently, the model does not account for contextual factors such as player injuries, coac
chological factors, which can significantly impact playoff outcomes.\n\n")
```

```
##    - Currently, the model does not account for contextual factors such as player injuries, coachi
ological factors, which can significantly impact playoff outcomes.
```

```
cat("2. **Limited Sample Size:**\n")
```

```
## 2. **Limited Sample Size:**
```

```
cat("   - The model is trained on data from only one season, which may not capture all nuances and v
mance.\n\n")
```

```
##    - The model is trained on data from only one season, which may not capture all nuances and var
nce.
```

```
cat("3. **Simplified Game Simulation:**\n")
```

```
## 3. **Simplified Game Simulation:**
```

```
cat("   - The game simulation process assumes independence between games, not considering momentum o
that may affect the outcome of playoff series.\n\n")
```

```
##    - The game simulation process assumes independence between games, not considering momentum or
at may affect the outcome of playoff series.
```

```
# Suggestions for Model Improvement

cat("## Suggestions for Model Improvement\n")
```

```
## ## Suggestions for Model Improvement
```

```
cat("1. **Incorporate Contextual Data:**\n")
```

```
## 1. **Incorporate Contextual Data:**
```

```
cat("   - With more time and data, the model can be enhanced by including contextual factors like pl
ching strategies, and psychological assessments.\n")
```

```
##    - With more time and data, the model can be enhanced by including contextual factors like play
ing strategies, and psychological assessments.
```

```
cat("   - This would require additional data sources and potentially developing new features to quan
\n")
```

```
##    - This would require additional data sources and potentially developing new features to quanti
```

```
cat("2. **Expand the Dataset:**\n")
```

```
## 2. **Expand the Dataset:**
```

```
cat("   - To address the limited sample size, the model can be trained on data from multiple seasons
e a broader range of scenarios and improve the model's robustness.\n")
```

```
##    - To address the limited sample size, the model can be trained on data from multiple seasons.
a broader range of scenarios and improve the model's robustness.
```

```
cat("   - Additionally, including data leading up to the playoffs can provide more context and impro
y.\n\n")
```

```
##    - Additionally, including data leading up to the playoffs can provide more context and improve
```

```
cat("3. **Advanced Simulation Techniques:**\n")
```

```
## 3. **Advanced Simulation Techniques:**
```

```
cat("   - Implementing more complex simulation techniques that consider dependencies and momentum be
series predictions.\n")
```

```
##    - Implementing more complex simulation techniques that consider dependencies and momentum betw
ries predictions.
```

```
cat("   - Techniques such as Monte Carlo simulations and dynamic modeling can be explored to better
of playoff series.\n\n")
```

```
##    - Techniques such as Monte Carlo simulations and dynamic modeling can be explored to better ca
f playoff series.
```

```
cat("4. **Play-In Tournament Predictions:**\n")
```

```
## 4. **Play-In Tournament Predictions:**
```

```
cat("   - Since the NBA introduced the Play-In Tournament in the 2020-21 season to determine the sev
each conference, it is crucial to simulate the play-in results first. These games are single-elimina
vely easier to predict. Accurate predictions of the Play-In Tournament outcomes are necessary before
results, ensuring more accurate predictions.\n\n")
```

```
##    - Since the NBA introduced the Play-In Tournament in the 2020-21 season to determine the seven
ach conference, it is crucial to simulate the play-in results first. These games are single-eliminat
ely easier to predict. Accurate predictions of the Play-In Tournament outcomes are necessary before
results, ensuring more accurate predictions.
```

```
cat("By addressing these weaknesses, the model can become more robust and accurate, providing valuab
-makers in the front office. These improvements will allow the model to consider various variables t
tcomes and provide more reliable predictions.\n")
```

```
## By addressing these weaknesses, the model can become more robust and accurate, providing valuable
akers in the front office. These improvements will allow the model to consider various variables tha
omes and provide more reliable predictions.
```

Part 3 – Finding Insights from Your Model

Find two teams that had a competitive window of 2 or more consecutive seasons making the playoffs and that under p
expectations for them, losing series they were expected to win. Why do you think that happened? Classify one of then
them as relating to a cause not currently accounted for in your model. If given more time and data, how would you use
your model?

```
# Load necessary libraries
library(dplyr)
library(htmltools)

# Analysis of Milwaukee Bucks and Miami Heat Playoff Performance

# Context
context <- "The Milwaukee Bucks and Miami Heat both had competitive windows with consecutive playoff
r team met the expectations of the model, losing series they were expected to win. Below is an analy
e, potential reasons for their losses, and suggestions for improving the model."

# 1. Playoff Performance Data
performance_data <- "
First Round (East) Results and Probabilities:
Boston Celtics vs Miami Heat
Boston Celtics 0 - Miami Heat 4: 1.96%
Boston Celtics 1 - Miami Heat 4: 5.16%
Boston Celtics 2 - Miami Heat 4: 7.62%
Boston Celtics 3 - Miami Heat 4: 9.71%
Boston Celtics 4 - Miami Heat 0: 15.25%
Boston Celtics 4 - Miami Heat 1: 23.36%
Boston Celtics 4 - Miami Heat 2: 21.57%
Boston Celtics 4 - Miami Heat 3: 15.37%
Boston Celtics wins: 75.55%, Miami Heat wins: 24.45%

Milwaukee Bucks vs Indiana Pacers
Milwaukee Bucks 0 - Indiana Pacers 4: 11.31%
Milwaukee Bucks 1 - Indiana Pacers 4: 18.76%
Milwaukee Bucks 2 - Indiana Pacers 4: 20.25%
Milwaukee Bucks 3 - Indiana Pacers 4: 16.45%
Milwaukee Bucks 4 - Indiana Pacers 0: 3.06%
Milwaukee Bucks 4 - Indiana Pacers 1: 7.02%
Milwaukee Bucks 4 - Indiana Pacers 2: 10.66%
Milwaukee Bucks 4 - Indiana Pacers 3: 12.49%
Milwaukee Bucks wins: 33.23%, Indiana Pacers wins: 66.77%
"

# 2. Competitive Playoff Windows
continuous_playoff_windows <- "
Milwaukee Bucks:
2020: 56-17, exited in the Eastern Conference Semifinals (won one round)
2021: 46-26, won the championship
2022: 51-31, exited in the Eastern Conference Semifinals (won one round)
2023: 58-24, exited in the first round (did not win a round)
2024: 49-33, exited in the first round (did not win a round)

Miami Heat:
2020: 44-29, lost in the Finals (won three rounds)
2021: 40-32, exited in the first round (did not win a round)
2022: 53-29, exited in the Eastern Conference Finals (won two rounds)
2023: 44-38, lost in the Finals (won three rounds)
2024: 46-36, exited in the first round (did not win a round)
"

# 3. Data Analysis
data_analysis <- "
Milwaukee Bucks:
Despite strong regular season performances, the Milwaukee Bucks underperformed in the playoffs. In t
probability of winning against the Indiana Pacers was 33.23%, but they were eliminated in the first
ance can be attributed to various factors, including key player Khris Middleton's injury during the
t tactical adjustments during the postseason.

Miami Heat:
In the 2023-24 season, the Miami Heat had a 24.45% probability of winning against the Boston Celtics
to the next round. The Heat's success is often linked to their playoff experience and head coach Eri
adjustment capabilities. Despite the model's lower predicted probability, they often perform well.
"

# 4. Reasons for Underperformance
reasons_for_underperformance <- "
```

```
Milwaukee Bucks:
Classification: Bad luck
Reason: Key player Khris Middleton's injury during the playoffs significantly impacted their perform
om events in the playoffs (e.g., referee decisions, player performance fluctuations) could affect th

Miami Heat:
Classification: Reasons not accounted for by the model
Reason: Coaching tactical adjustments, key player performances, and opponent's tactical changes can
lts. The model may not fully account for the unique tactical adjustments, player psychological state
in the playoffs.
"

# 5. Suggestions for Model Improvement
improvement_suggestions <- "
Milwaukee Bucks:
Luck factors: Increase the consideration of random events and key player performance fluctuations, s
e playoff scenarios to reflect win probability variations under different conditions.

Miami Heat:
Tactical and psychological factors: The model can incorporate more data on coaching tactical adjustm
gical states, such as playoff experience, coach's historical records, and player health status.

Improvement Measures:
1. Playoff Experience: Add data on players' and coaches' playoff experience and analyze its impact o
2. Health Status: Include data on player health and injuries to evaluate their potential impact on g
3. Tactical Adjustments: Collect and analyze data on different coaches' tactical changes and adjustm
s, making it a key variable in the model.
4. Mid-Season Trades: Analyze the impact of mid-season trades on team strength, particularly on play
"

# Conclusion
conclusion <- "
Conclusion:
Both the Milwaukee Bucks and Miami Heat had strong playoff contention windows but also faced unexpec
Bucks, this underperformance can be classified as bad luck, particularly due to Khris Middleton's in
ormance fluctuations. For the Heat, their resilience despite lower model predictions indicates the n
experience and coaching tactics in the model. Incorporating these factors can provide more accurate
ms better prepare for the playoffs.
"

# Print the analysis and insights with red color
cat(HTML("<span style='color:red'>**ANSWER :**</span>"), "\n\n")
```

```
## <span style='color:red'>**ANSWER :**</span>
```

```
cat(context, "\n\n")
```

```
## The Milwaukee Bucks and Miami Heat both had competitive windows with consecutive playoff appearan
t the expectations of the model, losing series they were expected to win. Below is an analysis of th
al reasons for their losses, and suggestions for improving the model.
```

```
cat(performance_data, "\n\n")
```

```
##
## First Round (East) Results and Probabilities:
## Boston Celtics vs Miami Heat
## Boston Celtics 0 - Miami Heat 4: 1.96%
## Boston Celtics 1 - Miami Heat 4: 5.16%
## Boston Celtics 2 - Miami Heat 4: 7.62%
## Boston Celtics 3 - Miami Heat 4: 9.71%
## Boston Celtics 4 - Miami Heat 0: 15.25%
## Boston Celtics 4 - Miami Heat 1: 23.36%
## Boston Celtics 4 - Miami Heat 2: 21.57%
## Boston Celtics 4 - Miami Heat 3: 15.37%
## Boston Celtics wins: 75.55%, Miami Heat wins: 24.45%
##
## Milwaukee Bucks vs Indiana Pacers
## Milwaukee Bucks 0 - Indiana Pacers 4: 11.31%
## Milwaukee Bucks 1 - Indiana Pacers 4: 18.76%
## Milwaukee Bucks 2 - Indiana Pacers 4: 20.25%
## Milwaukee Bucks 3 - Indiana Pacers 4: 16.45%
## Milwaukee Bucks 4 - Indiana Pacers 0: 3.06%
## Milwaukee Bucks 4 - Indiana Pacers 1: 7.02%
## Milwaukee Bucks 4 - Indiana Pacers 2: 10.66%
## Milwaukee Bucks 4 - Indiana Pacers 3: 12.49%
## Milwaukee Bucks wins: 33.23%, Indiana Pacers wins: 66.77%
##
```

```
cat(continuous_playoff_windows, "\n\n")
```

```
##
## Milwaukee Bucks:
## 2020: 56-17, exited in the Eastern Conference Semifinals (won one round)
## 2021: 46-26, won the championship
## 2022: 51-31, exited in the Eastern Conference Semifinals (won one round)
## 2023: 58-24, exited in the first round (did not win a round)
## 2024: 49-33, exited in the first round (did not win a round)
##
## Miami Heat:
## 2020: 44-29, lost in the Finals (won three rounds)
## 2021: 40-32, exited in the first round (did not win a round)
## 2022: 53-29, exited in the Eastern Conference Finals (won two rounds)
## 2023: 44-38, lost in the Finals (won three rounds)
## 2024: 46-36, exited in the first round (did not win a round)
##
```

```
cat(data_analysis, "\n\n")
```

```
##
## Milwaukee Bucks:
## Despite strong regular season performances, the Milwaukee Bucks underperformed in the playoffs. I
eir probability of winning against the Indiana Pacers was 33.23%, but they were eliminated in the fi
formance can be attributed to various factors, including key player Khris Middleton's injury during
cient tactical adjustments during the postseason.
##
## Miami Heat:
## In the 2023-24 season, the Miami Heat had a 24.45% probability of winning against the Boston Celt
ce to the next round. The Heat's success is often linked to their playoff experience and head coach
ic adjustment capabilities. Despite the model's lower predicted probability, they often perform well
##
```

```
cat(reasons_for_underperformance, "\n\n")
```

```
##
## Milwaukee Bucks:
## Classification: Bad luck
## Reason: Key player Khris Middleton's injury during the playoffs significantly impacted their perf
andom events in the playoffs (e.g., referee decisions, player performance fluctuations) could affect
##
## Miami Heat:
## Classification: Reasons not accounted for by the model
## Reason: Coaching tactical adjustments, key player performances, and opponent's tactical changes c
esults. The model may not fully account for the unique tactical adjustments, player psychological st
ion in the playoffs.
##
```

```
cat(improvement_suggestions, "\n\n")
```

```
##
## Milwaukee Bucks:
## Luck factors: Increase the consideration of random events and key player performance fluctuations
more playoff scenarios to reflect win probability variations under different conditions.
##
## Miami Heat:
## Tactical and psychological factors: The model can incorporate more data on coaching tactical adju
ological states, such as playoff experience, coach's historical records, and player health status.
##
## Improvement Measures:
## 1. Playoff Experience: Add data on players' and coaches' playoff experience and analyze its impac
## 2. Health Status: Include data on player health and injuries to evaluate their potential impact o
## 3. Tactical Adjustments: Collect and analyze data on different coaches' tactical changes and adju
ffs, making it a key variable in the model.
## 4. Mid-Season Trades: Analyze the impact of mid-season trades on team strength, particularly on p
##
```

```
cat(conclusion, "\n\n")
```

```
##
## Conclusion:
## Both the Milwaukee Bucks and Miami Heat had strong playoff contention windows but also faced unex
the Bucks, this underperformance can be classified as bad luck, particularly due to Khris Middleton'
performance fluctuations. For the Heat, their resilience despite lower model predictions indicates t
off experience and coaching tactics in the model. Incorporating these factors can provide more accur
teams better prepare for the playoffs.
##
```

ANSWER :

The Milwaukee Bucks and Miami Heat both had competitive windows with consecutive playoff appearances, but neith of the model, losing series they were expected to win. Below is an analysis of their performance, potential reasons for suggestions for improving the model.

First Round (East) Results and Probabilities: Boston Celtics vs Miami Heat Boston Celtics 0 - Miami Heat 4: 1.96% Bc 4: 5.16% Boston Celtics 2 - Miami Heat 4: 7.62% Boston Celtics 3 - Miami Heat 4: 9.71% Boston Celtics 4 - Miami He 4 - Miami Heat 1: 23.36% Boston Celtics 4 - Miami Heat 2: 21.57% Boston Celtics 4 - Miami Heat 3: 15.37% Boston C Heat wins: 24.45%

Milwaukee Bucks vs Indiana Pacers Milwaukee Bucks 0 - Indiana Pacers 4: 11.31% Milwaukee Bucks 1 - Indiana Pac Bucks 2 - Indiana Pacers 4: 20.25% Milwaukee Bucks 3 - Indiana Pacers 4: 16.45% Milwaukee Bucks 4 - Indiana Pac Bucks 4 - Indiana Pacers 1: 7.02% Milwaukee Bucks 4 - Indiana Pacers 2: 10.66% Milwaukee Bucks 4 - Indiana Pace Bucks wins: 33.23%, Indiana Pacers wins: 66.77%

Milwaukee Bucks: 2020: 56-17, exited in the Eastern Conference Semifinals (won one round) 2021: 46-26, won the ch exited in the Eastern Conference Semifinals (won one round) 2023: 58-24, exited in the first round (did not win a roun first round (did not win a round)

Miami Heat: 2020: 44-29, lost in the Finals (won three rounds) 2021: 40-32, exited in the first round (did not win a rour Eastern Conference Finals (won two rounds) 2023: 44-38, lost in the Finals (won three rounds) 2024: 46-36, exited in round)

Milwaukee Bucks: Despite strong regular season performances, the Milwaukee Bucks underperformed in the playoffs probability of winning against the Indiana Pacers was 33.23%, but they were eliminated in the first round. This underp to various factors, including key player Khris Middleton's injury during the playoffs and insufficient tactical adjustments

Miami Heat: In the 2023-24 season, the Miami Heat had a 24.45% probability of winning against the Boston Celtics bu next round. The Heat's success is often linked to their playoff experience and head coach Erik Spoelstra's strategic ac Despite the model's lower predicted probability, they often perform well.

Milwaukee Bucks: Classification: Bad luck Reason: Key player Khris Middleton's injury during the playoffs significantly Additionally, random events in the playoffs (e.g., referee decisions, player performance fluctuations) could affect their

Miami Heat: Classification: Reasons not accounted for by the model Reason: Coaching tactical adjustments, key play opponent's tactical changes can all impact the game results. The model may not fully account for the unique tactical a psychological state, and physical condition in the playoffs.

Milwaukee Bucks: Luck factors: Increase the consideration of random events and key player performance fluctuations playoff scenarios to reflect win probability variations under different conditions.

Miami Heat: Tactical and psychological factors: The model can incorporate more data on coaching tactical adjustment states, such as playoff experience, coach's historical records, and player health status.

Improvement Measures: 1. Playoff Experience: Add data on players' and coaches' playoff experience and analyze its Health Status: Include data on player health and injuries to evaluate their potential impact on game outcomes. 3. Tacti analyze data on different coaches' tactical changes and adjustments during the playoffs, making it a key variable in th Trades: Analyze the impact of mid-season trades on team strength, particularly on playoff rosters.

Conclusion: Both the Milwaukee Bucks and Miami Heat had strong playoff contention windows but also faced unexpe Bucks, this underperformance can be classified as bad luck, particularly due to Khris Middleton's injury and key playe For the Heat, their resilience despite lower model predictions indicates the need to consider playoff experience and cc Incorporating these factors can provide more accurate predictions and help teams better prepare for the playoffs.