# NBA Playoff Prediction

AUTHOR
Ayush Batra

This notebook attempts to predict the results of NBA Playoff series using regular season performance, player talent, and previous playoff experience.

# Loading Dependencies

## Packages

Here are the packages that I use in this notebook.

```r
library(tidyverse)   # for data manipulation
library(tidymodels)  # for modeling
library(recipes)     # also for modeling
library(nbastatR)    # Load NBA data
library(ggimage)     # put images on plots
library(gridExtra)   # put plots side by side
library(knitr)       # get cleaner model output

# increase size of connection buffer to be able to load box score data
Sys.setenv("VROOM_CONNECTION_SIZE" = 2*131072)
```

Next, I just define a constant color to use for some plots along with a theme to make the graphs look nicer.

```r
# color to use for graphs
MY_COLOR = "#C6A664"

theme_bbs <- function() {
  font = "Arial Unicode MS"
  bg = "#E4DBD7"
  light_ln = "#A0A0A0"
  dark_ln = "#404040"
  theme_minimal() %+replace%
```

```
  theme(plot.background = element_rect(fill = bg,
                                       color = NA),
        panel.border = element_blank(),
        panel.background = element_blank(),
        legend.background = element_blank(),
        panel.grid = element_line(color = light_ln,
                                  linewidth = 0.25),
        panel.grid.minor = element_blank(),
        plot.title = element_text(family = font,
                                  size = 18,
                                  face = 'bold',
                                  hjust = 0.5,
                                  vjust = 2),
        plot.subtitle = element_text(family = font,
                                     size = 12,
                                     hjust = 0.5,
                                     vjust = 1),
        plot.caption = element_text(family = font,
                                    size = 9,
                                    hjust = 1,
                                    vjust = -5,
                                    color = dark_ln),
        axis.title.x = element_text(family = font,
                                    size = 15,
                                    vjust = -2,
                                    color = dark_ln),
        axis.title.y = element_text(family = font,
                                    size = 15,
                                    angle = 90,
                                    vjust = 3,
                                    color = dark_ln),
        axis.text = element_text(family = font,
                                 size = 13,
                                 color = dark_ln),
        legend.title = element_text(family = font,
                                    size = 13,
                                    color = dark_ln,
                                    face = 'bold',
                                    hjust = 0.5),
        legend.text = element_text(family = font,
                                   size = 12,
```

```
                                        color = dark_ln),
          legend.box.background = element_blank(),
          axis.ticks = element_line(color = light_ln),
          plot.margin = unit(c(1,1,1,1),"cm"))
}
```

## Load in NBA Data

First, we must load in the data. The relevant data includes team box scores for the regular season (to calculate team regular season stats) and team box scores for the playoffs (to gather the playoff series). This data can be loaded using the `nbastatR` package's `game_logs()` function, as done below.

```
# gather team box score stats
team_logs <- game_logs(seasons = c(2011:2023),
                       result_types = "team",
                       season_types = c("Regular Season", "Playoffs"))
```

```
Acquiring NBA basic team game logs for the 2010-11 Regular Season
Acquiring NBA basic team game logs for the 2010-11 Playoffs
Acquiring NBA basic team game logs for the 2011-12 Regular Season
Acquiring NBA basic team game logs for the 2011-12 Playoffs
Acquiring NBA basic team game logs for the 2012-13 Regular Season
Acquiring NBA basic team game logs for the 2012-13 Playoffs
Acquiring NBA basic team game logs for the 2013-14 Regular Season
Acquiring NBA basic team game logs for the 2013-14 Playoffs
Acquiring NBA basic team game logs for the 2014-15 Regular Season
Acquiring NBA basic team game logs for the 2014-15 Playoffs
Acquiring NBA basic team game logs for the 2015-16 Regular Season
Acquiring NBA basic team game logs for the 2015-16 Playoffs
Acquiring NBA basic team game logs for the 2016-17 Regular Season
Acquiring NBA basic team game logs for the 2016-17 Playoffs
Acquiring NBA basic team game logs for the 2017-18 Regular Season
Acquiring NBA basic team game logs for the 2017-18 Playoffs
Acquiring NBA basic team game logs for the 2018-19 Regular Season
Acquiring NBA basic team game logs for the 2018-19 Playoffs
Acquiring NBA basic team game logs for the 2019-20 Regular Season
Acquiring NBA basic team game logs for the 2019-20 Playoffs
Acquiring NBA basic team game logs for the 2020-21 Regular Season
```

```
Acquiring NBA basic team game logs for the 2020-21 Playoffs
Acquiring NBA basic team game logs for the 2021-22 Regular Season
Acquiring NBA basic team game logs for the 2021-22 Playoffs
Acquiring NBA basic team game logs for the 2022-23 Regular Season
Acquiring NBA basic team game logs for the 2022-23 Playoffs
```

```r
team_logs24 <- game_logs(seasons = c(2024),
                         result_types = "team",
                         season_types = c("Regular Season"))
```

```
Acquiring NBA basic team game logs for the 2023-24 Regular Season
```

```r
team_logs <- rbind(team_logs, team_logs24)
```

Another piece of useful data is All-NBA voting shares, which will be helpful for estimating player talent. I webscraped this data from Basketball Reference. The webscraping code can be seen in the webscraping python notebook. In addition, I aggregated NBA playoff experience stats for each season. The process for gathering the playoff experience stats can be seen in the `nba_playoff_experience.qmd` file.

```r
allnba <- read_csv("data/allnba.csv")
experience <- read_csv("data/nba_experience.csv")
```

# Data Aggregation

## Gathering Regular Season Stats

An important and obvious predictor of playoff success is regular season performance. There are lots of ways to measure regular season success, but I think using net rating (point differential per 100 possessions) is the simplest and most effective way.

First, we have to do some cleaning with the team names in the team box scores since some team names have changed over the years.

```
# get the ID and abbreviation (slug) associated with each team
team_ids <- team_logs %>%
  distinct(idTeam, nameTeam, slugTeam)

# vector of outdated team names
duplicates <- c("New Orleans Hornets", "LA Clippers", "New Jersey Nets"
                "Charlotte Bobcats")

# remove the outdated teams, which now has the correct number of teams
team_ids <- team_ids %>%
  filter(!(nameTeam %in% duplicates))
print(nrow(team_ids))
```

```
[1] 30
```

```
# change outdated team abbreviations
team_logs <- team_logs %>%
  mutate(slugTeam = case_when(
    slugTeam == "NOH" ~ "NOP",
    slugTeam == "NJN" ~ "BKN",
    .default = slugTeam
  )) %>%
  mutate(slugOpponent = case_when(
    slugOpponent == "NOH" ~ "NOP",
    slugOpponent == "NJN" ~ "BKN",
```

```
    .default = slugOpponent
  ))
```

Now that the team names are all fixed up, we can aggregate the box score statistics by team and season.

```
# There is probably a more efficient way of doing this but oh well

# gather offensive stats
team_stats <- team_logs %>%
  group_by(idTeam, nameTeam, yearSeason, typeSeason) %>%
  summarize(G = n(),
            W = sum(isWin),
            MIN = sum(minutesTeam),
            FGM = sum(fgmTeam),
            FGA = sum(fgaTeam),
            FG3M = sum(fg3mTeam),
            FG3A = sum(fg3aTeam),
            FTM = sum(ftmTeam),
            FTA = sum(ftaTeam),
            OREB = sum(orebTeam),
            DREB = sum(drebTeam),
            AST = sum(astTeam),
            STL = sum(stlTeam),
            BLK = sum(blkTeam),
            TOV = sum(tovTeam),
            PF = sum(pfTeam),
            PTS = sum(ptsTeam)) %>%
  ungroup()
```

```
`summarise()` has grouped output by 'idTeam', 'nameTeam', 'yearSeason'.
You can
override using the `.groups` argument.
```

```
# gather defensive stats
def_stats <- team_logs %>%
  group_by(slugOpponent, yearSeason, typeSeason) %>%
  summarize(G = n(),
            L = sum(isWin),
            MIN = sum(minutesTeam),
            FGM = sum(fgmTeam),
```

```
              FGA = sum(fgaTeam),
              FG3M = sum(fg3mTeam),
              FG3A = sum(fg3aTeam),
              FTM = sum(ftmTeam),
              FTA = sum(ftaTeam),
              OREB = sum(orebTeam),
              DREB = sum(drebTeam),
              AST = sum(astTeam),
              STL = sum(stlTeam),
              BLK = sum(blkTeam),
              TOV = sum(tovTeam),
              PF = sum(pfTeam),
              PTS = sum(ptsTeam)) %>%
    ungroup()
```

```
`summarise()` has grouped output by 'slugOpponent', 'yearSeason'. You
can
override using the `.groups` argument.
```

Next, we can feature engineer some common advanced box score statistics using the aggregated raw box score stats. Note that many of the stats created below don't end up getting used in the rest of the analysis, but it would be possible to create a different playoff prediction model using different predictor variables. While creating these stats, we use the formula $FGA + 0.44 * FTA + TOV - OREB$ to estimate the number of possessions from box score statistics. This shows up in the calculations for pace and offensive rating.

```
# put the stats we want to create into a function
createStats <- function(df) {
  df <- df %>%
    mutate(PACE = 240 * (FGA + TOV - OREB + 0.44 * FTA) / MIN,
           # Pace = estimate of possessions per 48 minutes
           ORTG = 100 * PTS / (FGA + TOV - OREB + 0.44 * FTA),
           # Offensive rating = points scored per 100 possessions
           TSP = 100 * PTS / (2 * (FGA + 0.44 * FTA)),
           # True Shooting %: takes 3-pointers and free throws into acc
           FG3P = 100 * FG3M / FG3A,
           # team 3-point percentage
           FTP = 100 * FTM / FTA,
           # team free throw percentage
           TPAr = 100 * FG3A / FGA,
```

```
            # three point attempt rate = % of field goal attempts from 3
            FTR = 100 * FTA / FGA,
            # free throw rate = proportion of free throw attempts to FGA
            AST_RATE = 100 * AST / FGM,
            # assist rate = percentage of made shots that were assisted
            TOV_RATE = 100 * TOV / (FGA + TOV + 0.44 * FTA))
            # turnover rate = estimate of % of plays ending in a turnove
  return(df)
}

# get the stats for both offense and defense
team_stats <- createStats(team_stats)
def_stats <- createStats(def_stats)
```

Now, we need to join the offensive and defensive stats dataframes together so we can use them in the future.

```
# join ID info to defensive stats
def_stats <- def_stats %>%
  left_join(team_ids, by = c("slugOpponent" = "slugTeam"),
            relationship = "many-to-one") %>%
  select(-slugOpponent, -nameTeam)

# join offensive and defensive stats together in one dataframe
all_stats <- team_stats %>%
  inner_join(def_stats, by = c("idTeam", "yearSeason", "typeSeason"),
             suffix = c(".off", ".def"),
             relationship = "one-to-one")

# create rebounding percentage for offense and defense, which requires
# a mix of team and opponent stats
all_stats <- all_stats %>%
  mutate(ORBP = 100 * OREB.off / (OREB.off + DREB.def),
         DRBP = 100 * DREB.off / (DREB.off + OREB.def))
```

This last step below filters the data to include include stats from the regular season and only from years before the current NBA season.

```
# filter stats to only include Regular Season stats before this year
rs_stats <- all_stats %>%
  filter(typeSeason == "Regular Season",
```

```
        yearSeason < 2024) %>%
  mutate(nameTeam = ifelse(nameTeam == "LA Clippers",
                           "Los Angeles Clippers",
                           nameTeam))
```

## Measuring Player Talent

In this project, I used All-NBA voting shares as a proxy for player talent. Every year at the end of the season, 100 people selected by the NBA vote for the All-NBA teams. Each voter selects a 1st team, 2nd team, and 3rd team. Players get 5 points for being voted on the 1st team, 3 points for the 2nd team, and 1 point for the 3rd team. The voting share for a given player (ranging from 0 to 1) is his total number of voting shares divided by the maximum possible number of voting shares.

There is a little bit to clean with the All-NBA data as well. Primarily, there are some players who played for multiple teams due to be traded mid season. For these players, I decided to manually enter the team they played for at the end of the year. Note that we only care about the players after 2011 because later we will only be modeling playoff series from 2011 onwards.

```
# filter for players that played for multiple teams in a season
unknowns <- allnba %>%
  filter(Season >= 2011, Tm == "TOT")

# manually change to team they played for last during the season
allnba <- allnba %>%
  mutate(Tm = case_when(
    Player == "Carmelo Anthony" & Season == 2011 ~ "NYK",
    Player == "Deron Williams" & Season == 2011 ~ "NJN",
    Player == "Kendrick Perkins" & Season == 2011 ~ "OKC",
    Player == "Gerald Wallace" & Season == 2011 ~ "POR",
    Player == "Monta Ellis" & Season == 2012 ~ "MIL",
    Player == "Rudy Gay" & Season == 2013 ~ "TOR",
    Player == "DeMarcus Cousins" & Season == 2017 ~ "NOP",
    Player == "Tobias Harris" & Season == 2019 ~ "PHI",
    Player == "Andre Drummond" & Season == 2020 ~ "CLE",
    Player == "James Harden" & Season == 2021 ~ "BRK",
    Player == "Nikola Vučević" & Season == 2021 ~ "CHI",
```

```
    Player == "Kevin Durant" & Season == 2023 ~ "PHO",
    Player == "Mikal Bridges" & Season == 2023 ~ "BRK",
    .default = Tm
  ))
```

Next, we aggregate the All-NBA voting shares by season and team. This number serves as a proxy for the “player talent” on the team.

```
# aggregate All-NBA voting shares
allnba_counts <- allnba %>%
  mutate(Made = ifelse(`# Tm` != "ORV", 1, 0)) %>%
  group_by(Season, Tm) %>%
  summarize(n = n(),
            total_shares = sum(Share),
            AllNBA = sum(Made)) %>%
  ungroup()
```

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

Lastly, we have to join this data onto our previous dataframe of regular season stats.

```
# manipulate team abbreviations so we can match with the team stats
slugs <- team_logs %>%
  distinct(nameTeam, slugTeam) %>%
  filter(nameTeam != "LA Clippers") %>%
  mutate(slugTeam = case_when(
    nameTeam == "Brooklyn Nets" ~ "BRK",
    nameTeam == "Charlotte Bobcats" ~ "CHH",
    nameTeam == "Charlotte Hornets" ~ "CHO",
    nameTeam == "New Jersey Nets" ~ "NJN",
    nameTeam == "New Orleans Hornets" ~ "NOH",
    nameTeam == "Phoenix Suns" ~ "PHO",
    .default = slugTeam
  )) %>%
  add_row(nameTeam = "Charlotte Bobcats", slugTeam = "CHA")

# filter to only include stats past 2011, keep relevant data
allnba_counts2 <- allnba_counts %>%
  filter(Season >= 2011) %>%
```

```
  left_join(slugs, by = c("Tm" = "slugTeam")) %>%
  select(nameTeam, Season, total_shares)

# verify that each team in the All-NBA voting shares data has a match i
# the regular season stats dataframe
# a result of a dataframe with 0 rows means we are all good
allnba_counts2 %>% anti_join(rs_stats,
                             by = c("nameTeam", "Season" = "yearSeason"
```

```
# A tibble: 0 × 3
# ℹ 3 variables: nameTeam <chr>, Season <dbl>, total_shares <dbl>
```

```
# join the data, fill in missing values with 0 (since they had no All-N
# voting shares)
rs_stats <- rs_stats %>%
  left_join(allnba_counts2, by = c("nameTeam", "yearSeason" = "Season")
  mutate(total_shares = ifelse(is.na(total_shares), 0, total_shares))
```

## Adding Playoff Experience

Now that we have added All-NBA voting shares stats, we can move on to adding the stats for playoff experience. I gathered the playoff experience stats in a different file (see `nba_playoff_experience.qmd`) and saved it to a csv, which we loaded at the beginning of the notebook. The numbers in the `experience` column are a weighted average of playoff minutes played before the given season, where the weights correspond to player minutes per game during the regular season. Since the data is already in a tidy form, we just have to make a few small changes with team abbreviations then join it to the other data.

```
# edit team abbreviations, join with team IDs
experience <- experience %>%
  mutate(slugTeam = case_when(
    slugTeam == "NJN" ~ "BKN",
    slugTeam == "NOH" ~ "NOP",
    .default = slugTeam
  )) %>%
  inner_join(team_ids, by = "slugTeam") %>%
  select(idTeam, season, experience)
```

```r
# join experience to existing stats data
rs_stats <- rs_stats %>%
  inner_join(experience, by = c("idTeam", "yearSeason" = "season"))
```

## Compile Playoff Series

Now that we have the relevant team statistics for each year, we must gather the playoff series so we can make our final dataframe that we can use for modeling. To get the playoff series, we start with the playoff team logs, which tell us which matchups occurred in the playoffs. Using that, we can manipulate the data so we have the year, round, series start date, higher-seeded team (plays first game of series at home), and lower-seeded team. In the series dataframe, the `Home` column denotes the higher-seeded team while the `Away` column denotes the lower-seeded team because the higher-seeded team always plays the first game at home.

```r
# get year, starting date, teams for each series
series <- team_logs %>%
  filter(typeSeason == "Playoffs") %>%
  group_by(yearSeason, slugTeam, slugOpponent) %>%
  mutate(game_num = c(1:n())) %>%
  ungroup() %>%
  filter(game_num == 1, locationGame == "H") %>%
  select(yearSeason, dateGame, slugTeam, slugOpponent)

# add IDs, other identifiers (like round) to series data
series <- series %>%
  group_by(yearSeason) %>%
  mutate(num = c(1:n())) %>%
  ungroup() %>%
  mutate(Round = case_when(
    num <= 8 ~ "Conf QF",
    num <= 12 ~ "Conf SF",
    num <= 14 ~ "Conf Finals",
    num == 15 ~ "NBA Finals"
  )) %>%
  mutate(id = c(1:n())) %>%
```

```
  select(id, yearSeason, Round, dateGame,
         Home = slugTeam, Away = slugOpponent)
```

Next, we want to identify which team won each playoff series. Since all playoff series since 2011 have been best of 7, we can find the team that wins 4 games and consider them to be the winner of the series.

```
# find which team won 4 games in the series
winners <- team_logs %>%
  filter(typeSeason == "Playoffs", outcomeGame == "W") %>%
  count(yearSeason, slugTeam, slugOpponent, outcomeGame) %>%
  filter(n == 4) %>%
  select(yearSeason, Winner = slugTeam, Loser = slugOpponent)

# get series id's for series where higher-seeded team (Home) won
home_winner_ids <- series %>%
  semi_join(winners, by = c("yearSeason",
                            "Home" = "Winner",
                            "Away" = "Loser")) %>%
  pull(id)

# assign win or loss to higher-seeded team (Home)
series <- series %>%
  mutate(Result = ifelse(id %in% home_winner_ids, "Win", "Loss")) %>%
  mutate(Neutral = ifelse(yearSeason == 2020, 1, 0))
```

Lastly, we can substitute the team abbreviations with full team names, just to make the dataframe look nicer.

```
# switch out team abbreviations for full team names
series <- series %>%
  mutate(Home = case_when(
    Home == "BKN" ~ "BRK",
    Home == "CHA" ~ "CHO",
    Home == "PHX" ~ "PHO",
    .default = Home
  )) %>%
  mutate(Away = case_when(
    Away == "BKN" ~ "BRK",
    Away == "CHA" ~ "CHO",
```

```
    Away == "PHX" ~ "PHO",
    .default = Away)) %>%
  inner_join(slugs, by = c("Home" = "slugTeam")) %>%
  inner_join(slugs, by = c("Away" = "slugTeam")) %>%
  select(id, yearSeason, Round, dateGame,
         Home = nameTeam.x, Away = nameTeam.y, Result, Neutral)
```

## Join Predictors and Outcomes

Now that we have our features in one dataframe and the series (along with the series outcomes) in a different dataframe, we can join them together so we are ready to model. First, we choose the relevant predictor variables (in this case, the relevant predictor variables include offensive/defensive rating, total All-NBA shares, and playoff experience; if you are editing this notebook, you can include different stats and produce a different playoff prediction model).

```
# select relevant predictors
relevant <- rs_stats %>%
  mutate(WP = W / G.off,
         PACE = (PACE.off + PACE.def)/2) %>%
  select(nameTeam, yearSeason, ORTG.off, ORTG.def, total_shares, experi

# join predictors to series data
series2 <- series %>%
  mutate(Away = case_when(
    (Away == "New Orleans Pelicans"
     & yearSeason == 2011) ~ "New Orleans Hornets",
    (Away == "Charlotte Hornets" &
       yearSeason == 2014) ~ "Charlotte Bobcats",
    .default = Away
  )) %>%
  left_join(relevant, by = c("yearSeason", "Home" = "nameTeam")) %>%
  left_join(relevant, by = c("yearSeason", "Away" = "nameTeam"),
            suffix = c(".h",".a"))
```

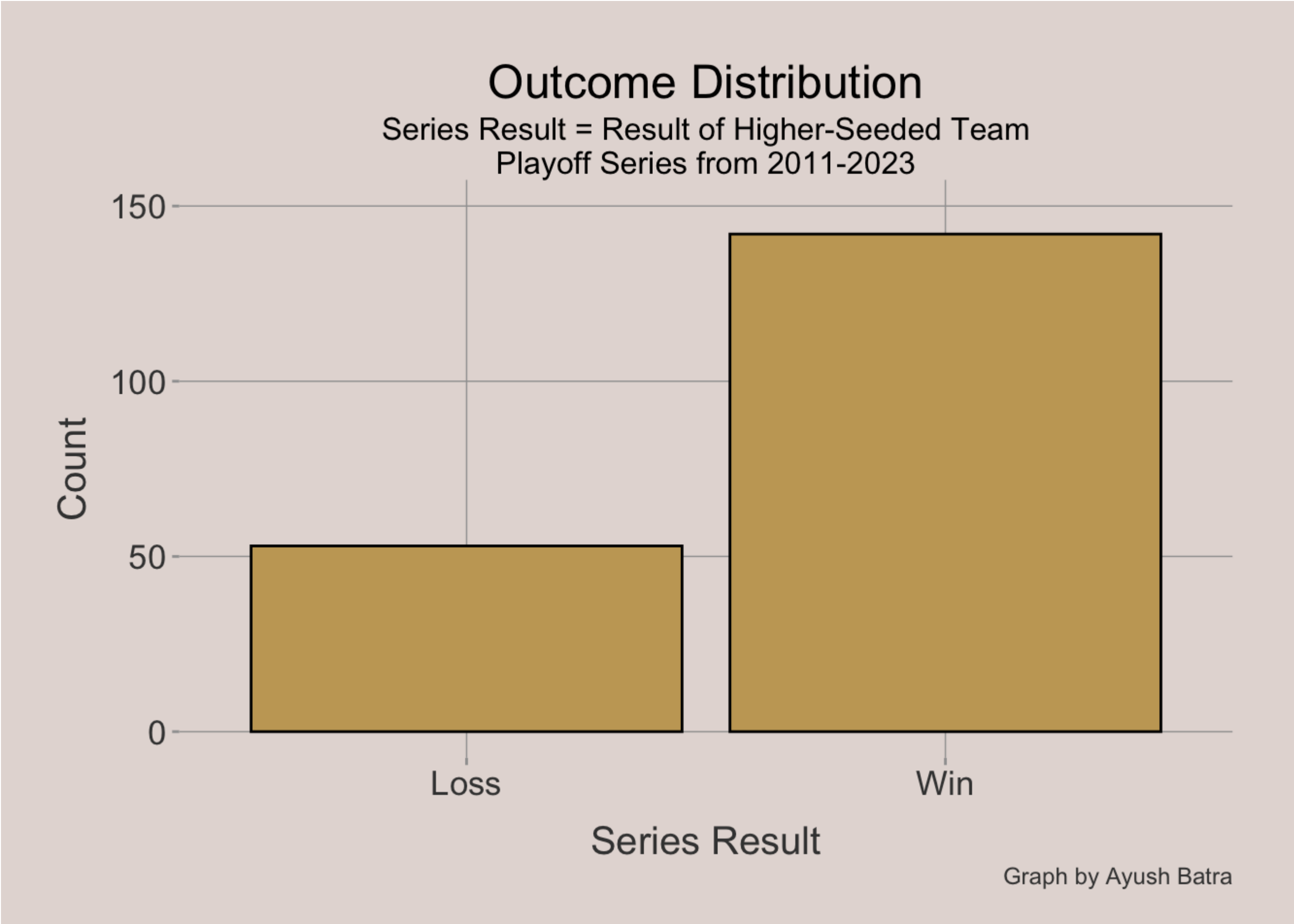
We are finally ready to do some analysis!

# Exploratory Data Analysis

## Response Variable

First, we should look at the distribution of the response/outcome variable, which is whether the higher-seeded team wins the playoff series. By looking at the plot, we see that around 70% of playoff series are won by the higher seeded team.

```
outcome_bar <- series2 %>%
  ggplot(aes(x = Result)) +
  geom_bar(color = 'black', fill = MY_COLOR) +
  labs(x = "Series Result",
       y = "Count",
       title = "Outcome Distribution",
       subtitle = "Series Result = Result of Higher-Seeded Team\nPlayof
       caption = "Graph by Ayush Batra") +
  scale_y_continuous(breaks = seq(0,150,50), limits = c(0, 150)) +
  theme_bbs()
outcome_bar
```




```
series2 %>%
  count(Result) %>%
  mutate(prob = n / sum(n))
```

```
# A tibble: 2 × 3
  Result     n  prob
  <chr>  <int> <dbl>
1 Loss      53 0.272
2 Win      142 0.728
```

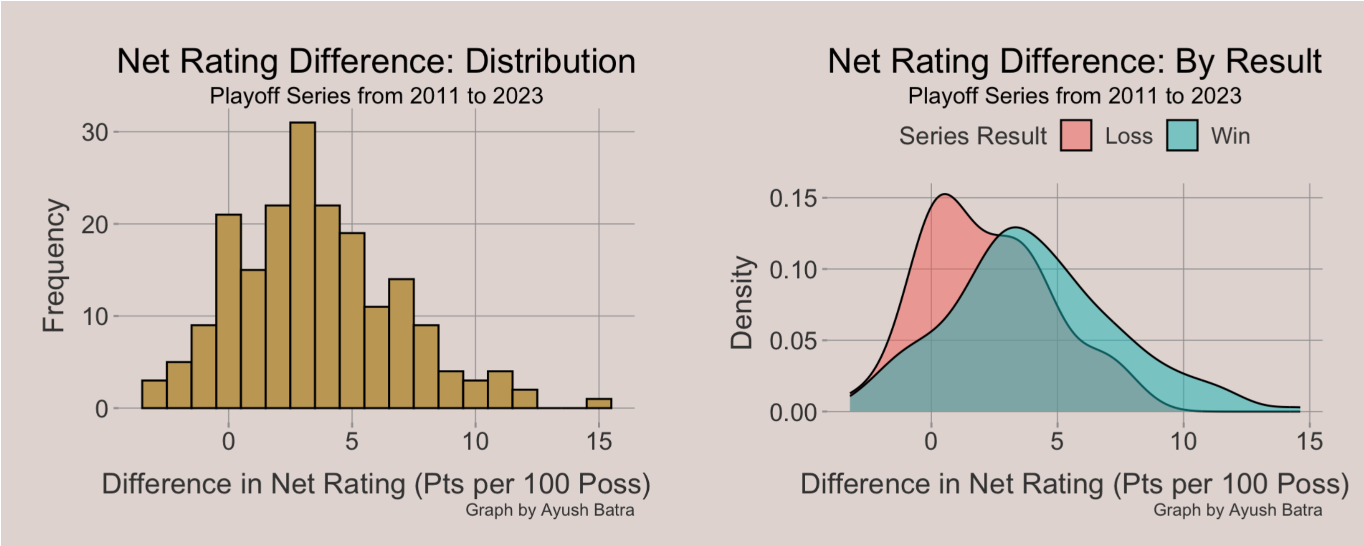
## Predictor Variables + Bivariate Relationships

We can also look at the distributions of the predictor variables and the relationship between the predictors and the outcome. First, we will look at how net rating impacts playoff series.

## Net Rating

To understand the distribution of net rating and its impact on winning playoff series, we can begin by looking at the distribution of net rating differences for the series. The net rating difference is the higher seeded team's net rating minus the lower seeded team's net rating. In addition, we can look at how the net rating difference distribution is for teams that won their series compared to teams that lost their series. It can be seen that higher seeded teams that win their playoff series have greater net rating differences than those that lose typically.

```
nrtg_dist <- series2 %>%
  # calculate net rating difference
  mutate(NRTG_h = ORTG.off.h - ORTG.def.h,
         NRTG_a = ORTG.off.a - ORTG.def.a,
         NRTG_diff = NRTG_h - NRTG_a) %>%
  ggplot(aes(x = NRTG_diff)) +
  geom_histogram(color = 'black', fill = MY_COLOR, binwidth = 1) +
  labs(x = "Difference in Net Rating (Pts per 100 Poss)",
       y = "Frequency",
       title = "Net Rating Difference: Distribution",
       subtitle = "Playoff Series from 2011 to 2023",
       caption = "Graph by Ayush Batra") +
  theme_bbs()

nrtg_res <- series2 %>%
  mutate(NRTG_h = ORTG.off.h - ORTG.def.h,
         NRTG_a = ORTG.off.a - ORTG.def.a,
         NRTG_diff = NRTG_h - NRTG_a) %>%
  ggplot(aes(x = NRTG_diff, fill = Result)) +
  geom_density(color = 'black', alpha = 0.5) +
  labs(x = "Difference in Net Rating (Pts per 100 Poss)",
       y = "Density",
       title = "Net Rating Difference: By Result",
       subtitle = "Playoff Series from 2011 to 2023",
       fill = "Series Result",
       caption = "Graph by Ayush Batra") +
  theme_bbs() +
  theme(legend.position = 'top')

grid.arrange(nrtg_dist, nrtg_res, nrow = 1)
```

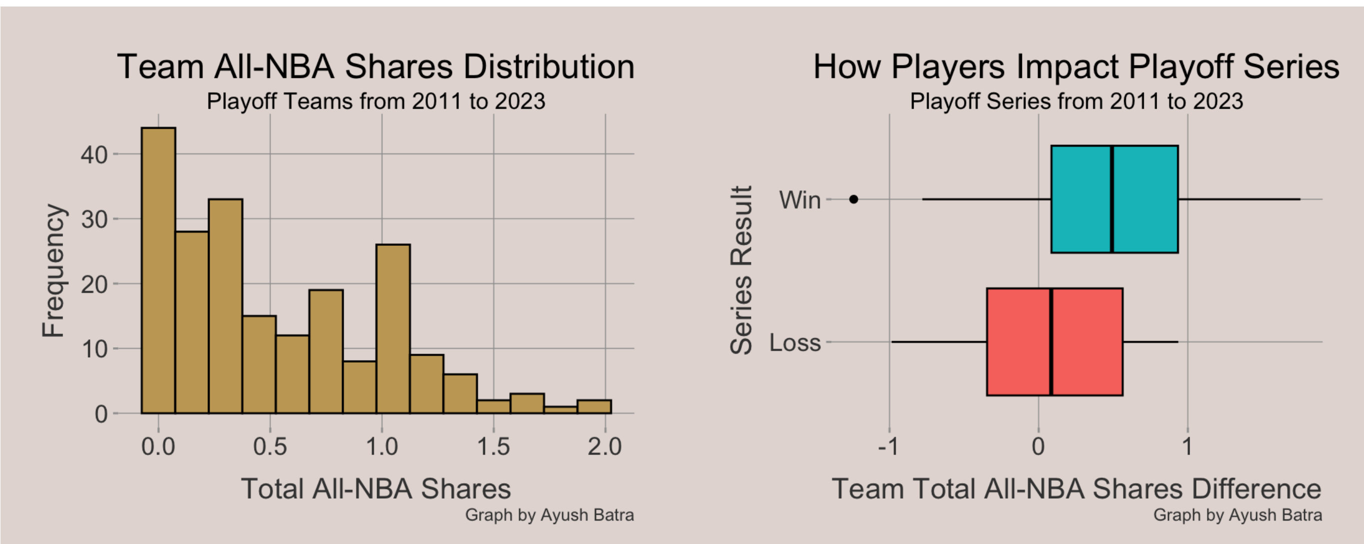



## All-NBA Shares

A more interesting variable is the total All-NBA shares per team. We can again look at its distribution and how it impacts winning playoff series. The distribution shown with the histogram below is a little different than the one from the section above. In this histogram, we look at the total All-NBA shares for every single playoff team since 2011, so each observation is a single team in a single season. In the histogram for net rating difference, we calculated the net rating difference between the higher seeded team and lower seeded team for each series and showed that distribution, so each observation was a series.

```
share_dist <- series2 %>%
  # get shares for every playoff team
  select(yearSeason, Tm = Home, total_shares = total_shares.h) %>%
  rbind(series2 %>% select(yearSeason,
                            Tm = Away,
                            total_shares = total_shares.a)) %>%
  distinct(yearSeason, Tm, total_shares) %>%
  # plot distribution of shares
  ggplot(aes(x = total_shares)) +
  geom_histogram(color = 'black', fill = MY_COLOR, binwidth = 0.15) +
  labs(x = "Total All-NBA Shares",
       y = "Frequency",
       title = "Team All-NBA Shares Distribution",
       subtitle = "Playoff Teams from 2011 to 2023",
       caption = "Graph by Ayush Batra") +
  theme_bbs()
```

```
# shares difference based on if team won or lost
share_res <- series2 %>%
  mutate(shares_diff = total_shares.h - total_shares.a) %>%
  ggplot(aes(x = shares_diff, y = Result, fill = Result)) +
  geom_boxplot(color = 'black', show.legend = F) +
  labs(x = "Team Total All-NBA Shares Difference",
       y = "Series Result",
       title = "How Players Impact Playoff Series",
       subtitle = "Playoff Series from 2011 to 2023",
       caption = "Graph by Ayush Batra") +
  theme_bbs()

grid.arrange(share_dist, share_res, nrow = 1)
```




The plot to the right of the distribution of All-NBA shares shows how the difference in total All-NBA shares between the higher seeded team and the lower seeded team impacts the chance of winning the playoff series. The median All-NBA shares difference for higher seeded teams that won their series was greater than the All-NBA shares difference for those who lost, which gives us some indication that player talent (as measured by All-NBA shares) is important.

## Playoff Experience

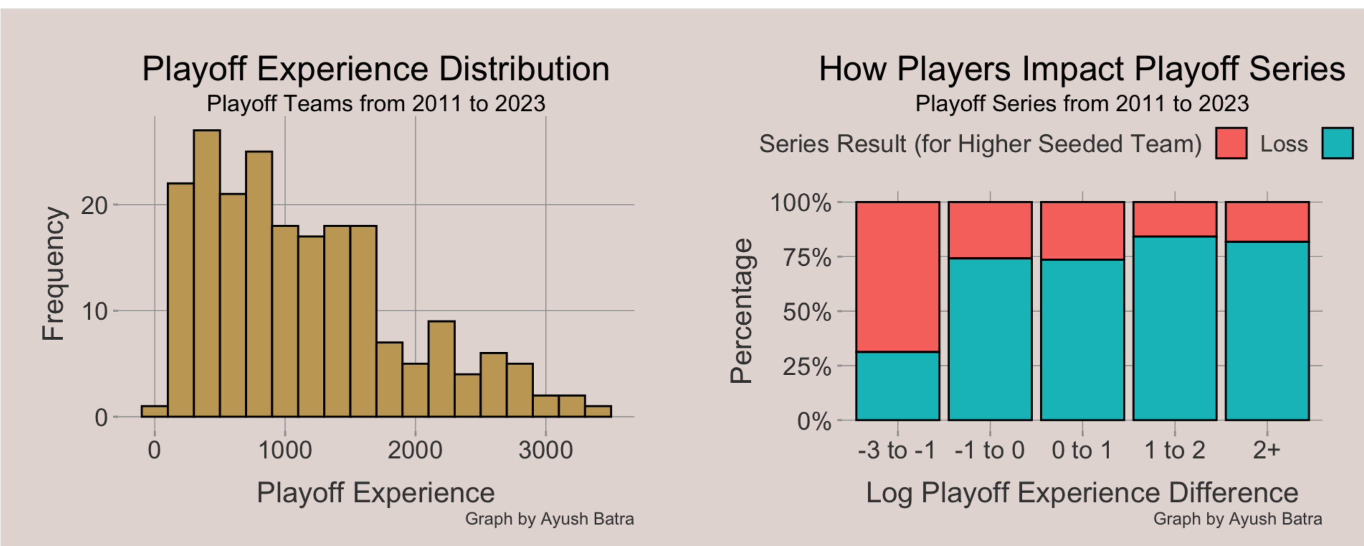
Lastly, we will look at playoff experience. Playoff experience measures the weighted average of playoff minutes played prior to the season in question, with the weights being proportional to minutes played per game during the regular season. See `NBA Playoff Experience.R` to see the exact code used to calculate it.

Once again, we will begin by showing the distribution of playoff experience. This is similar to the histogram from the All-NBA shares section, as it shows the playoff experience for each individual team. Next to it, we can see how playoff experience difference impacts series results. For this, I took the natural log of the playoff experience for both teams, then took the difference because the playoff experience is heavily skewed. Taking the log also causes an additional minute of playoff experience to be more important when a team has less experience rather than when a team has more experience.

Examining the plot on the right shows us that playoff experience is indeed valuable. In fact, there looks to be an interesting relationship as higher seeded teams that were heavily outmatched in terms of playoff experience won at a very low rate (about 25% of series), while teams with a close amount of playoff experience or more playoff experience all won at similar rates (around 75% of series).

```
exp_dist <- series2 %>%
  # get playoff experience for each team
  select(yearSeason, Tm = Home, exp = experience.h) %>%
  rbind(series2 %>% select(yearSeason,
                           Tm = Away,
                           exp = experience.a)) %>%
  distinct(yearSeason, Tm, exp) %>%
  # plot distribution
  ggplot(aes(x = exp)) +
  geom_histogram(color = 'black', fill = MY_COLOR, binwidth = 200) +
  labs(x = "Playoff Experience",
       y = "Frequency",
       title = "Playoff Experience Distribution",
       subtitle = "Playoff Teams from 2011 to 2023",
       caption = "Graph by Ayush Batra") +
  theme_bbs()

exp_res <- series2 %>%
  # note that we are using the natural log of playoff experience here
  # add 1 within the log just to avoid errors if a team has 0 experienc
  mutate(exp_diff = log(experience.h + 1) - log(experience.a + 1),
```

```
        group = cut(exp_diff, breaks = c(-3, seq(-1,2,1), 4))) %>%
  ggplot(aes(x = group, fill = Result)) +
  geom_bar(position = 'fill', color = 'black') +
  labs(x = "Log Playoff Experience Difference",
       y = "Percentage",
       fill = "Series Result (for Higher Seeded Team)",
       title = "How Players Impact Playoff Series",
       subtitle = "Playoff Series from 2011 to 2023",
       caption = "Graph by Ayush Batra") +
  scale_y_continuous(labels = label_percent()) +
  scale_x_discrete(labels = c("-3 to -1",
                              "-1 to 0",
                              "0 to 1",
                              "1 to 2",
                              "2+")) +
  theme_bbs() +
  theme(legend.position = 'top')

grid.arrange(exp_dist, exp_res, nrow = 1)
```




```
# shows the second graph in table form
series2 %>%
  mutate(exp_diff = log(experience.h + 1) - log(experience.a + 1),
         group = cut(exp_diff, breaks = c(-3, seq(-1,2,1), 4))) %>%
  group_by(group) %>%
  count(Result) %>%
  pivot_wider(id_cols = group, names_from = Result, values_from = n) %>
```

```
  select(group, Win, Loss) %>%
  mutate(win_pct = Win / (Win + Loss))
```

```
# A tibble: 5 × 4
# Groups:   group [5]
  group      Win  Loss win_pct
  <fct>    <int> <int>   <dbl>
1 (-3,-1]      5    11   0.312
2 (-1,0]      43    15   0.741
3 (0,1]       53    19   0.736
4 (1,2]       32     6   0.842
5 (2,4]        9     2   0.818
```

# Modeling

## Modeling Fitting

After some initial exploratory data analysis, we can begin modeling. This can allow us to separate the impact of each variable. First, we must create the relevant variables and split the data into a training set and testing set.

```
# create variables
series3 <- series2 %>%
  mutate(ORTG_diff = ORTG.off.h - ORTG.off.a,
         DRTG_diff = ORTG.def.h - ORTG.def.a,
         NRTG_diff = ORTG_diff - DRTG_diff,
         shares_diff = total_shares.h - total_shares.a,
         log_exp_diff = log(experience.h + 1) - log(experience.a + 1)) 
  select(id : Neutral, ORTG_diff : log_exp_diff) %>%
  mutate(Result = factor(Result, levels = c("Loss", "Win")))

# split data, put most in training set because limited data
set.seed(123)
series_split <- initial_split(series3, prop = 0.80)
series_train <- training(series_split)
series_test <- testing(series_split)
```

I want to keep the model simple, so I only included the three variables that I've written about previously, without any interactions or higher-order terms. Of course, there are many different models that can be made, but I just went with the simplest one.

```
# specify a logistic regression
series_spec <- logistic_reg() %>%
  set_engine("glm")

# regression formula
series_rec <- recipe(Result ~ NRTG_diff + shares_diff + log_exp_diff,
                     data = series3)

series_wflow <- workflow() %>%
  add_model(series_spec) %>%
  add_recipe(series_rec)

# fit the model
series_fit <- series_wflow %>%
  fit(series_train)

# display model coefficients
tidy(series_fit) %>%
  mutate(across(estimate : p.value, ~ round(.x, 4))) %>%
  kable()
```

| term | estimate | std.error | statistic | p.value |
|---|---|---|---|---|
| (Intercept) | 0.4089 | 0.2738 | 1.4932 | 0.1354 |
| NRTG_diff | 0.1217 | 0.0749 | 1.6241 | 0.1044 |
| shares_diff | 0.7741 | 0.3667 | 2.1106 | 0.0348 |
| log_exp_diff | 0.4532 | 0.2138 | 2.1198 | 0.0340 |

## Model Inference

The table above shows the model output. The p-values and coefficient estimates can give us a good idea about the importance of each variable. The most significant coefficient is the All-NBA shares difference, with a p-value of 0.0348. The coefficient of the shares difference variable indicates that the log odds of the higher-seeded team winning increases by 0.774 for each additional All-NBA share, holding net rating and experience constant. A more interpretable way to understand this is that the odds of the higher-seeded team winning the series increases by 116.9% for each additional All-NBA share. Similarly, we learn that each additional point in net rating difference increases the odds the higher-seeded team wins the series by 12.9%.

Overall, the final model has this equation:

$$\begin{aligned}\log\left(\frac{\hat{\pi}}{1-\hat{\pi}}\right) = \beta_0 &+ \beta_{net} \times NRTG\_diff \\ &+ \beta_{share} \times shares\_diff \\ &+ \beta_{exp} \times log\_exp\_diff\end{aligned}$$

Note that we can manipulate and expand this equation to better understand what each variable means.

$$\begin{aligned}\frac{\hat{\pi}}{1-\hat{\pi}} = \exp(\beta_0 &+ \beta_{net} \times (NRTG_{high} - NRTG_{low}) \\ &+ \beta_{share} \times (shares_{high} - shares_{low}) \\ &+ \beta_{exp} \times (\log(exp_{high}) - \log(exp_{low}))\end{aligned}$$

$$\begin{aligned}\frac{\hat{\pi}}{1-\hat{\pi}} = \exp(\beta_0) &\times \exp(\beta_{net}(NRTG_{high} - NRTG_{low})) \\ &\times \exp(\beta_{share}(shares_{high} - shares_{low})) \\ &\times \exp(\beta_{exp}(\log\left(\frac{exp_{high}}{exp_{low}}\right)))\end{aligned}$$

$$\begin{aligned}\frac{\hat{\pi}}{1-\hat{\pi}} = \exp(\beta_0) &\times \exp(\beta_{net}(NRTG_{high} - NRTG_{low})) \\ &\times \exp(\beta_{share}(shares_{high} - shares_{low})) \\ &\times \left(\frac{exp_{high}}{exp_{low}}\right)^{\beta_{exp}}\end{aligned}$$

After manipulating the equations, we can see how to interpret the coefficient for log experience difference. When the ratio of experience for the higher-seeded team to the lower-seeded team doubles, the odds of the higher-seeded team winning the series will increase by 36.9%. (at least I think that is the correct interpretation)

## Model Performance

To see how good our model is, we can evaluate its predictions on the training and testing set. One metric we can use is area under the ROC curve.

```
# get probablistic predictions for train and test sets
train_pred <- series_train %>%
  bind_cols(predict(series_fit, new_data = series_train, type = "prob"))
test_pred <- series_test %>%
  bind_cols(predict(series_fit, new_data = series_test, type = "prob"))

# calculate ROC area under curve for both sets
roc_auc(train_pred, Result, .pred_Win, event_level = "second")
```

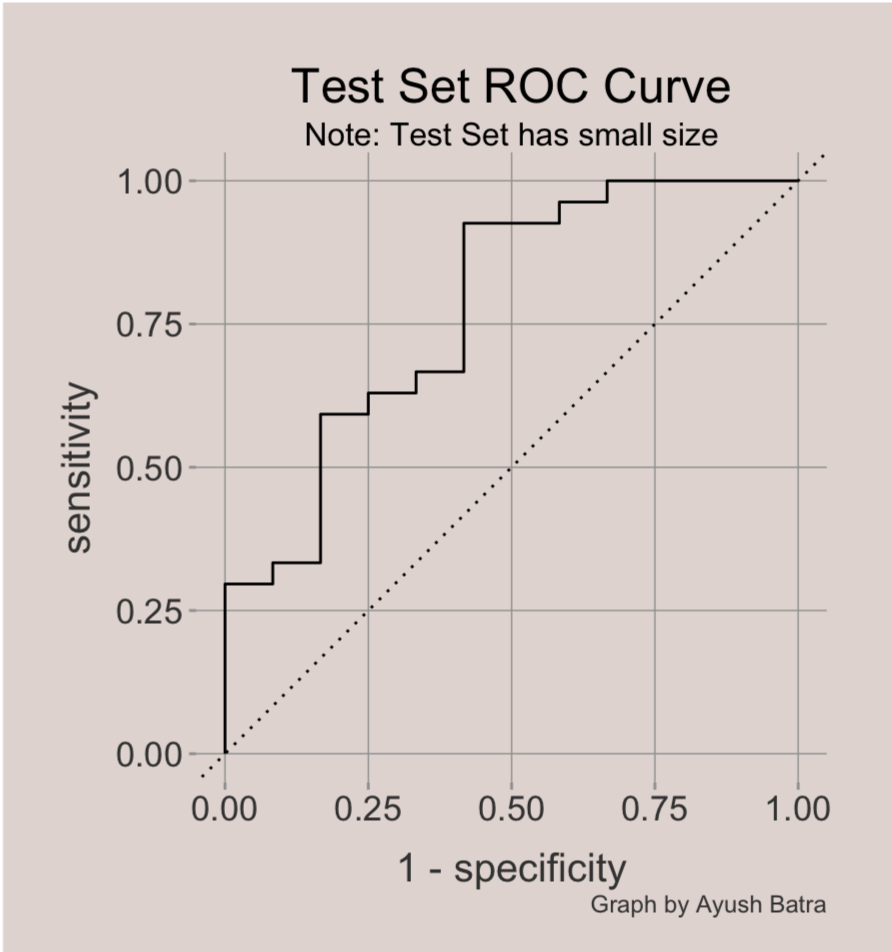
```
# A tibble: 1 × 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 roc_auc binary         0.720
```

```
roc_auc(test_pred, Result, .pred_Win, event_level = "second")
```

```
# A tibble: 1 × 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 roc_auc binary         0.778
```

```
# plot the ROC curve
roc_plot <- roc_curve(test_pred, Result,
                      .pred_Win,
                      event_level = "second") %>%
  autoplot() +
  labs(title = "Test Set ROC Curve",
       subtitle = "Note: Test Set has small size",
       caption = "Graph by Ayush Batra") +
```

```
  theme_bbs()
roc_plot
```




The test set ROC AUC (area under curve) of 0.742 is promising as it is a good bit better than the random guess diagonal line. Furthermore, the fact that there is little separation between the training AUC and testing AUC tells us that there the model isn't overfit, which was expected since we used a very simple model. To further evaluate the results, we can look at a confusion matrix of the test set predictions.

```
# create a binary prediction with a cutoff prob of 75%
# I chose 75% because the baseline win probability for the higher-seede
# team is about 73%
CUTOFF = 0.75
train_pred <- train_pred %>%
  mutate(pred_binary = ifelse(.pred_Win > CUTOFF, "Win", "Loss"))
test_pred <- test_pred %>%
  mutate(pred_binary = ifelse(.pred_Win > CUTOFF, "Win", "Loss"))

# display confusion matrix
```

```
test_pred %>%
  count(Result, pred_binary) %>%
  pivot_wider(id_cols = Result, names_from = pred_binary,
              values_from = n) %>%
  mutate(Result = ifelse(Result == "Win",
                         "Actual Win",
                         "Actual Loss")) %>%
  kable(col.names = c("", "Predicted Loss", "Predicted Win"))
```

| | Predicted Loss | Predicted Win |
|---|---|---|
| Actual Loss | 7 | 5 |
| Actual Win | 5 | 22 |

From the confusion matrix, we can gather some important metrics. The sensitivity (or 1 - false negative rate) is 81.5%, so 81.5% of series that were actually won by the higher-seeded team were predicted to be won by the higher-seeded team. The specificity (or 1 - false positive rate) is 58.3%. This means that 58.3% of the series that the higher seeded team lost were predicted to be lost. Lastly, the precision is 81.5, which means that 81.5% of our predicted wins were actually wins. Overall, these metrics are fairly promising and indicate that our model is not bad, especially for its simplicity. The only red flag is a low specificity as the model tends to produce false positives, which in this case means we tend to predict that a team will win a series when they actually lose.

I was curious to see what the model predicted for series in last year's NBA Playoffs, so I generated a table below that summarizes these results. Like most people, the model wasn't able to predict the Heat's surprise run. However, it correctly picked the lower seeded Warriors to win against the Kings in round 1 and the Heat to beat the Knicks in the Conference Semi Finals.

```
# get predictions for all series
final_pred <- series3 %>%
  bind_cols(predict(series_fit, new_data = series3, type = "prob"))

# display predictions from 2023 playoffs
final_pred %>%
  filter(yearSeason == 2023) %>%
```

```
  select(Round, dateGame, Home, Away, Result, .pred_Win, .pred_Loss) %>%
  mutate(prediction = ifelse(.pred_Win > .5, 'Win', 'Loss')) %>%
  kable(digits = 3)
```

| Round | dateGame | Home | Away | Result | .pred_Win | .pred_Loss | pre |
|---|---|---|---|---|---|---|---|
| Conf QF | 2023-04-15 | Cleveland Cavaliers | New York Knicks | Loss | 0.779 | 0.221 | Win |
| Conf QF | 2023-04-15 | Boston Celtics | Atlanta Hawks | Win | 0.943 | 0.057 | Win |
| Conf QF | 2023-04-15 | Sacramento Kings | Golden State Warriors | Loss | 0.472 | 0.528 | Loss |
| Conf QF | 2023-04-15 | Philadelphia 76ers | Brooklyn Nets | Win | 0.861 | 0.139 | Win |
| Conf QF | 2023-04-16 | Denver Nuggets | Minnesota Timberwolves | Win | 0.817 | 0.183 | Win |
| Conf QF | 2023-04-16 | Memphis Grizzlies | Los Angeles Lakers | Loss | 0.534 | 0.466 | Win |
| Conf QF | 2023-04-16 | Phoenix Suns | Los Angeles Clippers | Win | 0.644 | 0.356 | Win |
| Conf QF | 2023-04-16 | Milwaukee Bucks | Miami Heat | Loss | 0.810 | 0.190 | Win |
| Conf SF | 2023-04-29 | Denver Nuggets | Phoenix Suns | Win | 0.686 | 0.314 | Win |
| Conf SF | 2023-04-30 | New York Knicks | Miami Heat | Loss | 0.450 | 0.550 | Loss |
| Conf SF | 2023-05-01 | Boston Celtics | Philadelphia 76ers | Win | 0.725 | 0.275 | Win |
| Conf SF | 2023-05-02 | Golden State Warriors | Los Angeles Lakers | Loss | 0.686 | 0.314 | Win |

| Round | dateGame | Home | Away | Result | .pred_Win | .pred_Loss | pre |
|---|---|---|---|---|---|---|---|
| Conf Finals | 2023-05-16 | Denver Nuggets | Los Angeles Lakers | Win | 0.711 | 0.289 | Win |
| Conf Finals | 2023-05-17 | Boston Celtics | Miami Heat | Loss | 0.892 | 0.108 | Win |
| NBA Finals | 2023-06-01 | Denver Nuggets | Miami Heat | Win | 0.726 | 0.274 | Win |

## 2024 Predictions

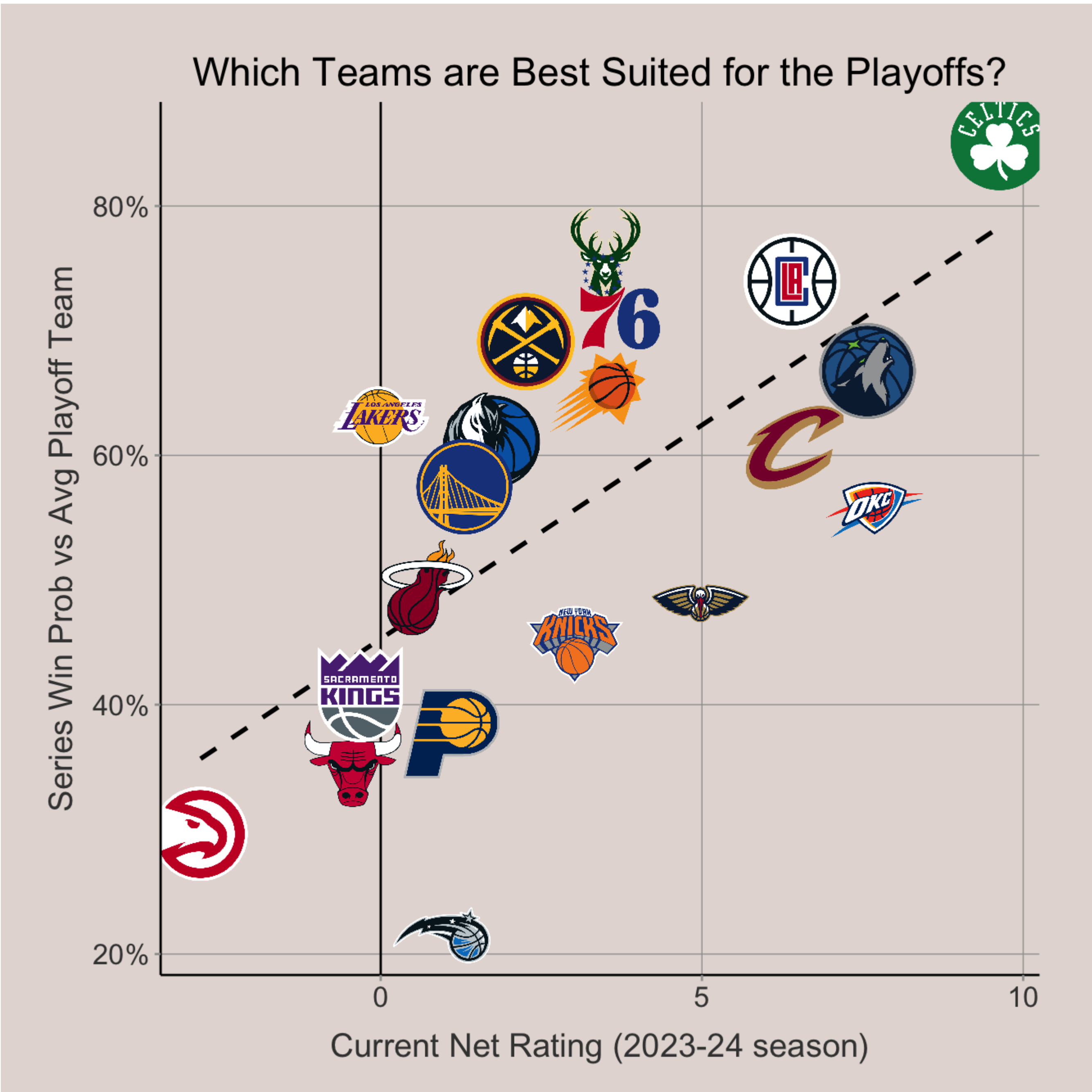
One last thing that I felt would be interesting is to see what the model's predictions would be for this season. Obviously we don't know the playoff matchups yet, so I ranked the teams by their series win probability if they were the higher-seeded team against the league average playoff contender this year.

To evaluate teams for this season, I made two plots. The first plot shows each team's probability of winning a series against the average of the teams (y-axis) vs their current net rating (x-axis). Teams above the dashed line (like the Bucks, Nuggets, 76ers, and Lakers) are expected to be better in the playoffs due to lots of experience and player talent. Meanwhile, teams below the dashed line (like the Magic, Cavs, Pelicans, and Rockets) are expected to be worse in the playoffs due to less experience and less high-level player talent.

```
library(hoopR)

# read data for 2024 predicted all-nba players and experience
allnba24 <- read_csv("data/allnba_2024.csv")
exp24 <- read_csv("data/nba_exp_2024.csv")

# teams with < 5% chance to make playoffs, according to ESPN BPI
no_playoffs <- c("DET", "WAS", "CHA", "SAS", "POR", "MEM",
                 "HOU", "TOR", "UTA", "BKN")

stats24 <- all_stats %>%
  # get team stats for 2024 regular season only
```

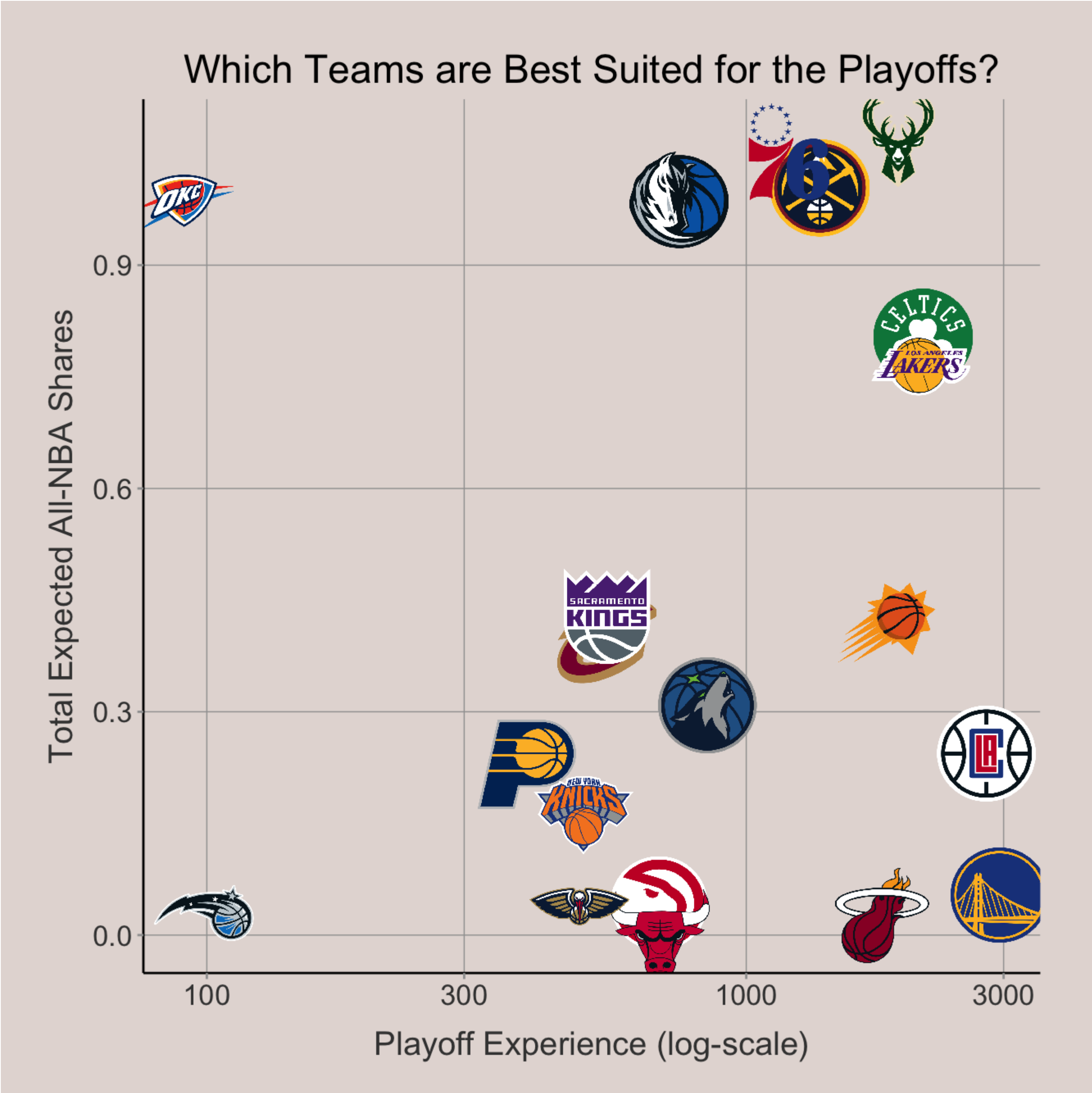
```r
  filter(yearSeason == 2024, typeSeason == "Regular Season") %>%
  # clean up names and abbreviations
  mutate(nameTeam = ifelse(nameTeam == "LA Clippers",
                           "Los Angeles Clippers",
                           nameTeam)) %>%
  left_join(slugs, by = c("nameTeam")) %>%
  mutate(slugTeam = case_when(
    slugTeam == "CHO" ~ "CHA",
    slugTeam == "PHO" ~ "PHX",
    slugTeam == "BRK" ~ "BKN",
    .default = slugTeam
  )) %>%
  # select relevant variables, add on player talent + experience
  select(nameTeam, slugTeam, ORTG.off, ORTG.def) %>%
  left_join(allnba24, by = c("slugTeam" = "Tm")) %>%
  left_join(exp24, by = c("slugTeam")) %>%
  # filter out non-playoff teams
  filter(slugTeam %in% no_playoffs == FALSE) %>%
  # calculate relevant statistics
  mutate(exp_Shares = round(exp_Shares, 4)) %>%
  mutate(NRTG = ORTG.off - ORTG.def,
         NRTG_diff = NRTG - mean(NRTG),
         shares_diff = exp_Shares - mean(exp_Shares),
         exp_diff = experience - mean(experience),
         log_exp_diff = log(experience + 1) - log(mean(experience) + 1)

# add series predictions
# note: each "series" is just the team against the average of all the t
stats24 <- stats24 %>%
  bind_cols(predict(series_fit, new_data = stats24, type = "prob")) %>%
  arrange(-.pred_Win)

# load in data for NBA logo images
nba_logos <- hoopR::nba_teams() %>%
  select(team_abbreviation, logo)

# plot current net rating vs probability of winning series vs avg team
scatter24 <- stats24 %>%
  left_join(nba_logos, by = c("slugTeam" = "team_abbreviation")) %>%
  ggplot(aes(x = NRTG, y = .pred_Win)) +
  geom_smooth(method = lm, se = FALSE,
```

```
                linetype = 'dashed', color = 'black') +
  geom_vline(xintercept = 0, linewidth = 0.5) +
  geom_image(aes(image = logo), size = 0.12) +
  labs(x = "Current Net Rating (2023-24 season)",
       y = "Series Win Prob vs Avg Playoff Team",
       title = "Which Teams are Best Suited for the Playoffs?") +
  theme_bbs() +
  theme(axis.line = element_line(color = 'black', linewidth = 0.5)) +
  scale_y_continuous(labels = label_percent())
scatter24
```



The second plot, which is shown below, displays each team's expected All-NBA shares and playoff experience. The playoff experience is shown on a log-scale. Teams in the top left corner have lots of playoff experience and are expected to have high All-NBA shares totals. In contrast, teams towards the bottom right have neither of these. We can see clearly from this that the Bucks, 76ers, Nuggets, and Celtics should be formidable in the playoffs, while the Magic might have a hard time finding success due to low experience and All-NBA shares. Meanwhile, the Thunder are the only team that is likely to have an All-NBA player (Shai Gilgeous-Alexander) but has nearly no playoff experience. On the other hand, the Clippers, Warriors, Heat, and Suns have lots of playoff experience but probably won't have an All-NBA player (according to the All-NBA model) this year despite the fact that they have great players like Steph Curry, Kawhi Leonard, and Jimmy Butler. It is important to note that the expected All-NBA shares does not take into account the new 65 game rule, so players that won't meet that criteria for the real All-NBA selections later this year will still have an expected All-NBA share greater than 0 in this model. This is why the 76ers have a high expected All-NBA shares despite Embiid missing the 65 game criteria.

```r
scatter24_2 <- stats24 %>%
  left_join(nba_logos, by = c("slugTeam" = "team_abbreviation")) %>%
  ggplot(aes(x = experience, y = exp_Shares)) +
  geom_image(aes(image = logo), size = 0.12) +
  labs(x = "Playoff Experience (log-scale)",
       y = "Total Expected All-NBA Shares",
       title = "Which Teams are Best Suited for the Playoffs?") +
  scale_x_log10() +
  theme_bbs() +
  theme(axis.line = element_line(color = 'black', linewidth = 0.5))
scatter24_2
```




Thank you for reading my notebook!