Mehra_Kai_Final-Project

Kai Mehra

2023-05-1

```
# Loading neccessary libraries
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.2.1     v dplyr   1.1.1
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(ggpubr)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(GPArotation)
```

```
##
## Attaching package: 'GPArotation'
##
## The following objects are masked from 'package:psych':
##
##     equamax, varimin
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(rms)
```

```
## Loading required package: Hmisc
## Loading required package: survival
##
## Attaching package: 'survival'
##
## The following object is masked from 'package:caret':
##
##     cluster
##
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
##
## The following object is masked from 'package:psych':
##
##     describe
##
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
##
## The following objects are masked from 'package:base':
##
##     format.pval, units
##
## Loading required package: SparseM
##
## Attaching package: 'SparseM'
##
## The following object is masked from 'package:base':
##
##     backsolve
```

Data

```
# Loading Data
teams_per_game <- read.csv("./Data/Team Stats Per Game.csv")
advanced_stats <- read.csv("./Data/Advanced.csv")
```

```
end_season_teams <- read.csv("./Data/End of Season Teams.csv")
all_stars <- read.csv("./Data/nbaallstargames.csv")
```

```
dim(teams_per_game)
```

```
## [1] 1814   28
```

```
dim(advanced_stats)
```

```
## [1] 31135    32
```

```
dim(end_season_teams)
```

```
## [1] 2120   11
```

```
dim(all_stars)
```

```
## [1] 1715    6
```

There are four data sets used in this analysis that were all downloaded from Kaggle. The `teams_per_game` dataset contains the basic team statistics and playoff status for every team from 1947 until 2023. The `advanced_stats` dataset contains data on the advanced statistics accumulated by individual players from 1947 through 2023. The `end_season_teams` dataset displays which players were voted onto the end of season award teams (see variables of interest for more informaiton). Finally, the `all_stars` dataset lists every player to play in the annual NBA All-Star Game from 1951 to 2021

Research Question

What team statistical and individual award factors contribute to NBA teams making the playoffs? For example, are NBA playoff teams better rebounders than non-playoff teams?

Hypotheses

I hypothesize that there will be a strong correlation between the total number of all stars and end of season award team players and whether or not a team made the playoffs. Generally, teams with better players win more games, and better players are more likely to get awards. Additionally, I think playoff teams will score more points, assists, and rebounds while turning the ball over less. I think playoff teams will also have better average advanced stat metrics.

Variables of Interest

Independent Variables

- Basic Statistics: All of the below statistics are the basic statistics that record the events that happen during each game. These stats are recorded on a team level and on a per season basis. More information about each stat can be found here: https://www.nba.com/stats/help/glossary

  - Points per game
  - Assists per game
  - Rebounds per game (Offensive and Defensive)
  - Blocks per game
  - Steals per game
  - Turnovers per game
  - Personal Fouls per game
  - Field Goals Attempted, Field Goals Made, Field Goal Percentage
  - Free Throws Attempted, Free Throws Made, Free Throw Percentage
  - 2 Pointers Attempted, 2 Pointers Made, 2 Point Percentage
  - 3 Pointers Attempted, 3 Pointers Made, 3 Point Percentage

- Advanced Statistics: These three advanced statistics are considered all-in-one metrics that assign one number to evaluate the value of a player (higher is better). They all use different formulas, but they follow similar principles of attempting to use one number to determine a player's contribution to winning. I then took the average of each advanced statistic for each player on each team per season to generate an average value for each team per season.

- Average Win Shares (WS)

- Average Box Plus Minus (BPM)

- Average Value Over Replacement Player (VORP)

- End of Season Award Teams

  - All NBA Teams: At the end of each season, three teams of five players are selected by the media as the best players in the league. The 1st team is comprised of the top 5 players of the that season, with the 2nd team has the 6-10th and the 3rd has the 11-15th best players that season.
  - All Defensive Teams: There are two five player All Defensive teams that are comprised of the top 10 defenders in the NBA that season.
  - All Rookie Teams: Like the All NBA and Defensive teams, but there are two five player teams for the 10 best rookies (first-year players) of that season.

Dependent Variable

- Playoff Status - At the end of every NBA season, 16 teams make it to the post-season tournament known as the playoffs based on their regular season record. Since there are 30 teams in the league, about 53% of the teams make it to the playoffs each season. The winner of the post-season tournament is crowned the NBA champion of that season.

Data Wrangling

```
advanced_stats <-
  advanced_stats %>%
  filter(case_when(season == 1999 ~ g >= 40,
                   season == 2012 ~ g >= 52,
                   season == 2020 ~ g >= 52,
                   season == 2021 ~ g >= 57,
                   !(season %in% c(1999, 2012, 2020, 2021)) ~ g >= 65)
         ) %>% # remove shortened seasons
  select(player, season, tm, ws, bpm, vorp) %>% # select vars of interest
```

```
  filter(season >= 1980) %>%
  filter(season < 2023) %>%
  filter(tm != "TOT") # remove duplicates
```

The 1999 and 2012 seasons were affected by a lockout, and the season was shortend from the normal 82 games. The 2020 and 2021 seasons were affected by the COVID-19 pandemic and were also shortened. I set a cutoff of at least 80% of games for players to be elligible in this dataset. The 80% cutoff is also used by the NBA for award consideration.

```
advanced_stats_team <-
  advanced_stats %>%
  group_by(season, tm) %>% # group by season and team
  summarise(avg_ws = round(mean(ws), 3),
            avg_bpm = round(mean(bpm), 3),
            avg_vorp = round(mean(vorp), 3)) %>%
  unite("team_season", c(tm, season)) # create a team_season var for joining
```

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

```
advanced_stats_team <-
  na.omit(advanced_stats_team) # remove nas
```

The advanced stat value of each player on a team was averaged to come up with an overall team value.

```
end_season_teams <-
  end_season_teams %>%
  select(player, season, tm, type, number_tm) %>% # vars of interest
  filter(season >= 1980) %>%
  filter(season < 2023) %>%
  filter(tm != "TOT") %>% # remove duplicates
  unite("Award_Team", type:number_tm, na.rm = TRUE, remove = TRUE) %>%
  pivot_wider(names_from = Award_Team, values_from = Award_Team, values_fn = ~1, values_fill = 0) # piv
# each award team has its own column
```

```
end_season_teams_team <-
  end_season_teams %>%
  group_by(season, tm) %>% # group by season and team
  summarise(
    All_Defense_1st_total = sum(`All-Defense_1st`),
    All_Defense_2nd_total = sum(`All-Defense_2nd`),
    All_NBA_1st_total = sum(`All-NBA_1st`),
    All_NBA_2nd_total = sum(`All-NBA_2nd`),
    All_NBA_3rd_total = sum(`All-NBA_3rd`),
    All_Rookie_1st_total = sum(`All-Rookie_1st`),
    All_Rookie_2nd_total = sum(`All-Rookie_2nd`)
  ) %>% # summarize data onto a team scael
  unite("team_season", c(tm, season)) # create team_season var for joining
```

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

```
end_season_teams_team <-
  na.omit(end_season_teams_team) # remove nas
```

I summarized the data to make a count of how many players on each team received each type of award team selection.

```
teams_per_game <-
  teams_per_game %>%
  rename("tm" = "abbreviation") %>% # rename vars
  select(-lg, -team, -g) %>% # vars of interest
  filter(season >= 1980) %>%
  filter(season < 2023) %>%
  unite("team_season", c(tm, season)) # create team_season var for joining

teams_per_game <-
  na.omit(teams_per_game) # remove nas
```

```
all_stars <-
  all_stars %>%
  rename("season" = "Year",
         "tm" = "Team.1",
         "player" = "Player") %>% # rename vars
  select(player, season, tm) %>% # vars of interest
  filter(season >= 1980) %>%
  filter(season < 2023)
```

```
all_stars_team <-
  all_stars %>%
  group_by(season, tm) %>% # group by season and team
  summarise(
    total_all_stars = n()
  ) %>%
  unite("team_season", c(tm, season)) # create team_season var for joining
```

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

```
all_stars_team <-
  na.omit(all_stars_team) # remove nas
```

I summaized the data to count how many players made the all star game from each team during each season.

```
nba_data <-
  advanced_stats_team %>%
  full_join(all_stars_team, by="team_season") %>%
  full_join(end_season_teams_team, by="team_season") %>%
  full_join(teams_per_game, by= "team_season") # join datasets on team_season

nba_data[c(5:12)][is.na(nba_data[c(5:12)])] <- 0 # replace na values with 0

nba_data <-
  na.omit(nba_data) %>% # remove nas
```

```
  separate(team_season, into = c("team", "season")) # sep team_season

nba_data$season <- as.numeric(nba_data$season) # convert year into numeric
```

I joined all of the discrete data sets into one master data set called `nba_data`.

```
nba_data <-
  nba_data %>%
  select(
    -mp_per_game,
    -fg_per_game,
    -x3p_per_game,
    -x2p_per_game,
    -ft_per_game
  ) # remove extraneous variables
```

Visualization

Basic Stats Visualization

```
season_averages <-
  nba_data %>%
  group_by(season) %>% # group by season
  select(-season, -team) %>%
  summarise_all(mean) # calculate league average of stats per year
```

```
## Adding missing grouping variables: `season`
```

```
season_averages <-
  season_averages %>%
  mutate_if(is.numeric,
            round,
            digits = 2) # round data

season_averages$season <- as.numeric(season_averages$season)
```

For the sake of comparison, I calculated the league average values of each stat in my dataset.

Points Plot:

```
points_plot <-
  ggplot() +
  geom_point( # scatter plot
    data = nba_data,
    aes(
      x = season,
      y = pts_per_game,
      color = playoffs
    ),
```

```
    alpha = 0.8 # add transparency
  ) +
  geom_line( # line plot of league average values
    data = season_averages,
    aes(
      x = season,
      y = pts_per_game
      ),
      color = "black"
  ) +
  geom_point( # scatter plot of league average values
    data = season_averages,
    aes(
      x = season,
      y = pts_per_game
      ),
      fill = "#ffffff",
    color = "black",
    shape = 21
  ) +
  scale_color_manual( # better colors and labels
    labels = c("Non-Playoff Team", "Playoff Team"),
    values = c("#C9082A", "#006BB8") # official NBA logo colors
  ) +
  scale_x_continuous( # better x scale
    limits = c(1980, 2021),
    breaks = seq(1980, 2020, 10)
  ) +
  scale_y_continuous( # better y scale
    limits = c(84, 128),
    breaks = seq(84, 128, 5)
  ) +
  labs( # better labels
    x = "Season",
    y = "Points Per Game",
    title = "Points Per Game",
    color = "Playoff Status"
  ) +
  theme_bw() +
  theme(
  plot.background=element_rect(fill = "white"),
  panel.background = element_rect(fill = "ghostwhite"),
  legend.background = element_rect(fill = "white"),
  legend.key = element_rect(fill = "ghostwhite"),
  legend.position = "bottom",
  plot.title = element_text(hjust = 0.5, size = 12)
  ) # theme adjustments for better visualizations
```

Assists Plot:

```
assists_plot <-
  ggplot() +
  geom_point( # scatterplot
```

```
  data = nba_data,
  aes(
    x = season,
    y = ast_per_game,
    color = playoffs
  ),
  alpha = 0.8 # transparency
) +
geom_line( # line plot of league avg
  data = season_averages,
  aes(
    x = season,
    y = ast_per_game
    ),
    color = "black"
) +
geom_point( # scatter plot of league avg
  data = season_averages,
  aes(
    x = season,
    y = ast_per_game
    ),
    fill = "#ffffff",
  color = "black",
  shape = 21
) +
scale_color_manual( # better colors and labels
  labels = c("Non-Playoff Team", "Playoff Team"),
  values = c("#C9082A", "#006BB8")
) +
scale_x_continuous(
  limits = c(1980, 2021),
  breaks = seq(1980, 2020, 10)
) +
scale_y_continuous(
  limits = c(15, 32),
  breaks = seq(15, 32, 2)
) + # better scaling
labs(
  x = "Season",
  y = "Assists Per Game",
  title = "Assits Per Game",
  color = "Playoff Status"
) + # better labels
theme_bw() +
theme(
plot.background=element_rect(fill = "white"),
panel.background = element_rect(fill = "ghostwhite"),
legend.background = element_rect(fill = "white"),
legend.key = element_rect(fill = "ghostwhite"),
legend.position = "bottom",
plot.title = element_text(hjust = 0.5, size = 12)
) # better theme
```

Rebounds Plot:

```
rebounds_plot <- # same process as previous plots
  ggplot() +
  geom_point(
    data = nba_data,
    aes(
      x = season,
      y = trb_per_game,
      color = playoffs
    ),
    alpha = 0.8
  ) +
  geom_line(
    data = season_averages,
    aes(
      x = season,
      y = trb_per_game
      ),
      color = "black"
  ) +
  geom_point(
    data = season_averages,
    aes(
      x = season,
      y = trb_per_game
      ),
      fill = "#ffffff",
    color = "black",
    shape = 21
  ) +
  scale_color_manual(
    labels = c("Non-Playoff Team", "Playoff Team"),
    values = c("#C9082A", "#006BB8")
  ) +
  scale_x_continuous(
    limits = c(1980, 2021),
    breaks = seq(1980, 2020, 10)
  ) +
  scale_y_continuous(
    limits = c(35, 52),
    breaks = seq(35, 52, 2)
  ) +
  labs(
    x = "Season",
    y = "Rebounds Per Game",
    title = "Rebounds Per Game",
    color = "Playoff Status"
  ) +
  theme_bw() +
  theme(
  plot.background=element_rect(fill = "white"),
  panel.background = element_rect(fill = "ghostwhite"),
  legend.background = element_rect(fill = "white"),
```

```
  legend.key = element_rect(fill = "ghostwhite"),
  legend.position = "bottom",
  plot.title = element_text(hjust = 0.5, size = 12)
  )
```

Blocks Plot:

```
blocks_plot <- # same process as previous plots
  ggplot() +
  geom_point(
    data = nba_data,
    aes(
      x = season,
      y = blk_per_game,
      color = playoffs
    ),
    alpha = 0.8
  ) +
  geom_line(
    data = season_averages,
    aes(
      x = season,
      y = blk_per_game
      ),
      color = "black"
  ) +
  geom_point(
    data = season_averages,
    aes(
      x = season,
      y = blk_per_game
      ),
      fill = "#ffffff",
    color = "black",
    shape = 21
  ) +
  scale_color_manual(
    labels = c("Non-Playoff Team", "Playoff Team"),
    values = c("#C9082A", "#006BB8")
  ) +
  scale_x_continuous(
    limits = c(1980, 2021),
    breaks = seq(1980, 2020, 10)
  ) +
  scale_y_continuous(
    limits = c(2, 9),
    breaks = seq(2, 9, 1)
  ) +
  labs(
    x = "Season",
    y = "Steals Per Game",
    title = "Blocks Per Game",
    color = "Playoff Status"
```

```
  ) +
  theme_bw() +
  theme(
  plot.background=element_rect(fill = "white"),
  panel.background = element_rect(fill = "ghostwhite"),
  legend.background = element_rect(fill = "white"),
  legend.key = element_rect(fill = "ghostwhite"),
  legend.position = "bottom",
  plot.title = element_text(hjust = 0.5, size = 12)
  )
```

Steals Plot:

```
steals_plot <- # same process as previous plots
  ggplot() +
  geom_point(
    data = nba_data,
    aes(
      x = season,
      y = stl_per_game,
      color = playoffs
    ),
    alpha = 0.8
  ) +
  geom_line(
    data = season_averages,
    aes(
      x = season,
      y = stl_per_game
      ),
      color = "black"
  ) +
  geom_point(
    data = season_averages,
    aes(
      x = season,
      y = stl_per_game
      ),
      fill = "#ffffff",
    color = "black",
    shape = 21
  ) +
  scale_color_manual(
    labels = c("Non-Playoff Team", "Playoff Team"),
    values = c("#C9082A", "#006BB8")
  ) +
  scale_x_continuous(
    limits = c(1980, 2021),
    breaks = seq(1980, 2020, 10)
  ) +
  scale_y_continuous(
    limits = c(5, 13),
    breaks = seq(5, 13, 1)
```

```
  ) +
  labs(
    x = "Season",
    y = "Steals Per Game",
    title = "Steals Per Game",
    color = "Playoff Status"
  ) +
  theme_bw() +
  theme(
  plot.background=element_rect(fill = "white"),
  panel.background = element_rect(fill = "ghostwhite"),
  legend.background = element_rect(fill = "white"),
  legend.key = element_rect(fill = "ghostwhite"),
  legend.position = "bottom",
  plot.title = element_text(hjust = 0.5, size = 12)
  )
```

Turnovers Plot:

```
turnovers_plot <- # same process as previous plots
  ggplot() +
  geom_point(
    data = nba_data,
    aes(
      x = season,
      y = tov_per_game,
      color = playoffs
    ),
    alpha = 0.8
  ) +
  geom_line(
    data = season_averages,
    aes(
      x = season,
      y = tov_per_game
      ),
      color = "black"
  ) +
  geom_point(
    data = season_averages,
    aes(
      x = season,
      y = tov_per_game
      ),
      fill = "#ffffff",
    color = "black",
    shape = 21
  ) +
  scale_color_manual(
    labels = c("Non-Playoff Team", "Playoff Team"),
    values = c("#C9082A", "#006BB8")
  ) +
  scale_x_continuous(
```

```
    limits = c(1980, 2021),
    breaks = seq(1980, 2020, 10)
  ) +
  scale_y_continuous(
    limits = c(11, 23),
    breaks = seq(11, 23, 2)
  ) +
  labs(
    x = "Season",
    y = "Turnovers Per Game",
    title = "Turnovers Per Game",
    color = "Playoff Status"
  ) +
  theme_bw() +
  theme(
  plot.background=element_rect(fill = "white"),
  panel.background = element_rect(fill = "ghostwhite"),
  legend.background = element_rect(fill = "white"),
  legend.key = element_rect(fill = "ghostwhite"),
  legend.position = "bottom",
  plot.title = element_text(hjust = 0.5, size = 12)
  )
```

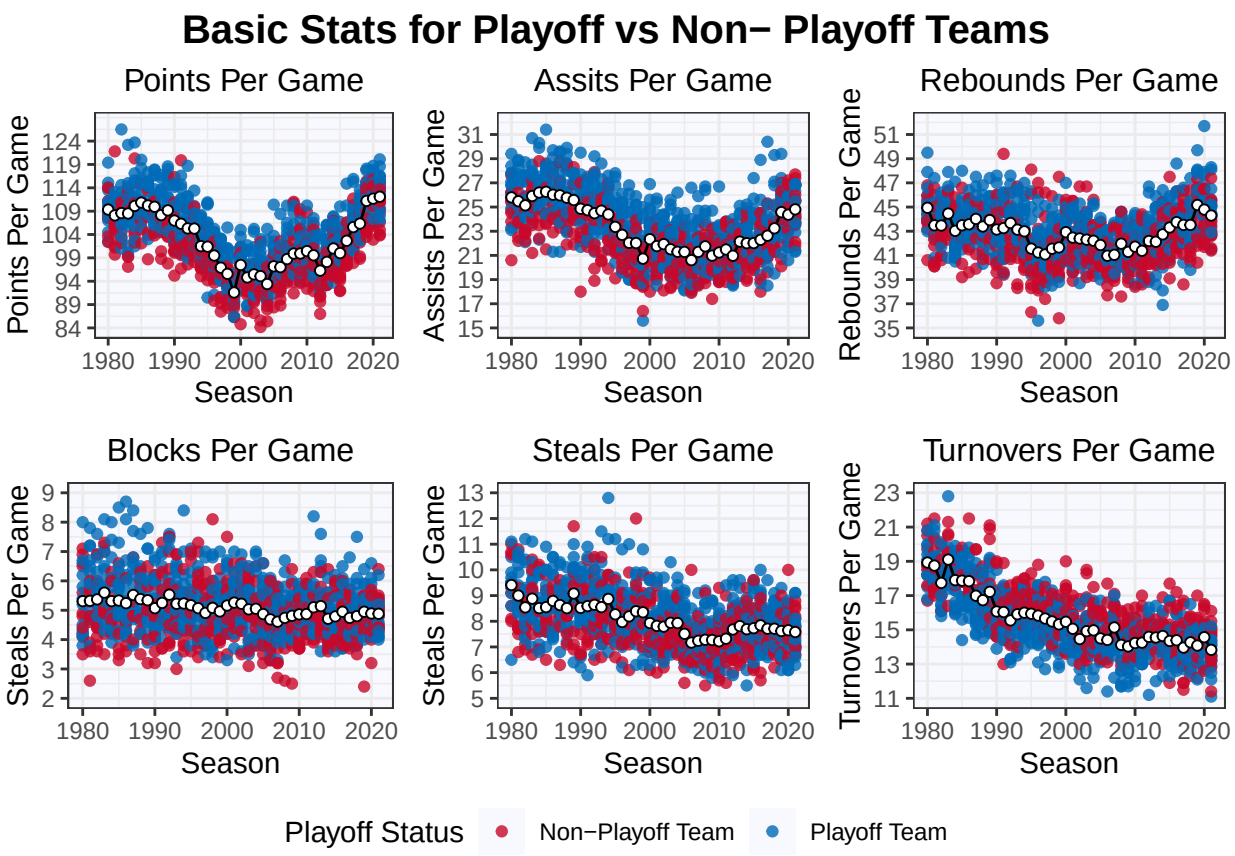
Basic Stats Multiplot:

```
basic_stats_multiplot <- # combining all plots into one
  ggarrange(points_plot,
          assists_plot,
          rebounds_plot,
          blocks_plot,
          steals_plot,
          turnovers_plot,
          ncol = 3,
          nrow = 2,
          common.legend = TRUE, # common legend
          legend = "bottom"
          )
```

```
## Warning: Removed 30 rows containing missing values (`geom_point()`).

## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 1 rows containing missing values (`geom_point()`).

## Warning: Removed 30 rows containing missing values (`geom_point()`).

## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 1 rows containing missing values (`geom_point()`).

## Warning: Removed 29 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 1 rows containing missing values (`geom_point()`).

## Warning: Removed 29 rows containing missing values (`geom_point()`).

## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 1 rows containing missing values (`geom_point()`).

## Warning: Removed 29 rows containing missing values (`geom_point()`).

## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 1 rows containing missing values (`geom_point()`).

## Warning: Removed 29 rows containing missing values (`geom_point()`).

## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 1 rows containing missing values (`geom_point()`).

## Warning: Removed 29 rows containing missing values (`geom_point()`).

## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 1 rows containing missing values (`geom_point()`).
```

```r
annotate_figure(
  basic_stats_multiplot,
  top = text_grob("Basic Stats for Playoff vs Non- Playoff Teams",
                  face = "bold",
                  size = 15)
) # add one overall title
```



Advanced Stats Visualization

Win Shares Plot:

```
ws_plot <- # same process as previous plots
  ggplot() +
  geom_point(
    data = nba_data,
    aes(
      x = season,
      y = avg_ws,
      color = playoffs
    ),
    alpha = 0.8
  ) +
  geom_line(
    data = season_averages,
    aes(
      x = season,
      y = avg_ws
      ),
      color = "black"
  ) +
  geom_point(
    data = season_averages,
```

```
    aes(
      x = season,
      y = avg_ws
      ),
      fill = "#ffffff",
    color = "black",
    shape = 21
  ) +
  scale_color_manual(
    labels = c("Non-Playoff Team", "Playoff Team"),
    values = c("#C9082A", "#006BB8")
  ) +
  scale_x_continuous(
    limits = c(1980, 2021),
    breaks = seq(1980, 2020, 10)
  ) +
  scale_y_continuous(
    limits = c(0, 10),
    breaks = seq(0, 10, 0.5)
  ) +
  labs(
    x = "Season",
    y = "Average Win Shares",
    title = "Average Win Shares",
    color = "Playoff Status"
  ) +
  theme_bw() +
  theme(
  plot.background=element_rect(fill = "white"),
  panel.background = element_rect(fill = "ghostwhite"),
  legend.background = element_rect(fill = "white"),
  legend.key = element_rect(fill = "ghostwhite"),
  legend.position = "bottom",
  plot.title = element_text(hjust = 0.5, size = 12)
  )
```

Box Plus Minus Plot:

```
bpm_plot <- # same process as previous plots
  ggplot() +
  geom_point(
    data = nba_data,
    aes(
      x = season,
      y = avg_bpm,
      color = playoffs
    ),
    alpha = 0.8
  ) +
  geom_line(
    data = season_averages,
    aes(
      x = season,
```

```
      y = avg_bpm
      ),
      color = "black"
  ) +
  geom_point(
    data = season_averages,
    aes(
      x = season,
      y = avg_bpm
      ),
      fill = "#ffffff",
    color = "black",
    shape = 21
  ) +
  scale_color_manual(
    labels = c("Non-Playoff Team", "Playoff Team"),
    values = c("#C9082A", "#006BB8")
  ) +
  scale_x_continuous(
    limits = c(1980, 2021),
    breaks = seq(1980, 2020, 10)
  ) +
  scale_y_continuous(
    limits = c(-5.2, 4.8),
    breaks = seq(-5.2, 4.8, 0.8)
  ) +
  labs(
    x = "Season",
    y = "Box Plus Minus",
    title = "Box Plus Minus",
    color = "Playoff Status"
  ) +
  theme_bw() +
  theme(
  plot.background=element_rect(fill = "white"),
  panel.background = element_rect(fill = "ghostwhite"),
  legend.background = element_rect(fill = "white"),
  legend.key = element_rect(fill = "ghostwhite"),
  legend.position = "bottom",
  plot.title = element_text(hjust = 0.5, size = 12)
  )
```

Value Over Replacement Player Plot:

```
vorp_plot <- # same process as previous plots
  ggplot() +
  geom_point(
    data = nba_data,
    aes(
      x = season,
      y = avg_vorp,
      color = playoffs
    ),
    alpha = 0.8
```

```
  ) +
  geom_line(
    data = season_averages,
    aes(
      x = season,
      y = avg_vorp
      ),
      color = "black"
  ) +
  geom_point(
    data = season_averages,
    aes(
      x = season,
      y = avg_vorp
      ),
      fill = "#ffffff",
    color = "black",
    shape = 21
  ) +
  scale_color_manual(
    labels = c("Non-Playoff Team", "Playoff Team"),
    values = c("#C9082A", "#006BB8")
  ) +
  scale_x_continuous(
    limits = c(1980, 2021),
    breaks = seq(1980, 2020, 10)
  ) +
  scale_y_continuous(
    limits = c(-1.5, 4),
    breaks = seq(-1.5, 4, 0.5)
  ) +
  labs(
    x = "Season",
    y = "Average VORP",
    title = "Average VORP",
    color = "Playoff Status"
  ) +
  theme_bw() +
  theme(
  plot.background=element_rect(fill = "white"),
  panel.background = element_rect(fill = "ghostwhite"),
  legend.background = element_rect(fill = "white"),
  legend.key = element_rect(fill = "ghostwhite"),
  legend.position = "bottom",
  plot.title = element_text(hjust = 0.5, size = 12)
  )
```

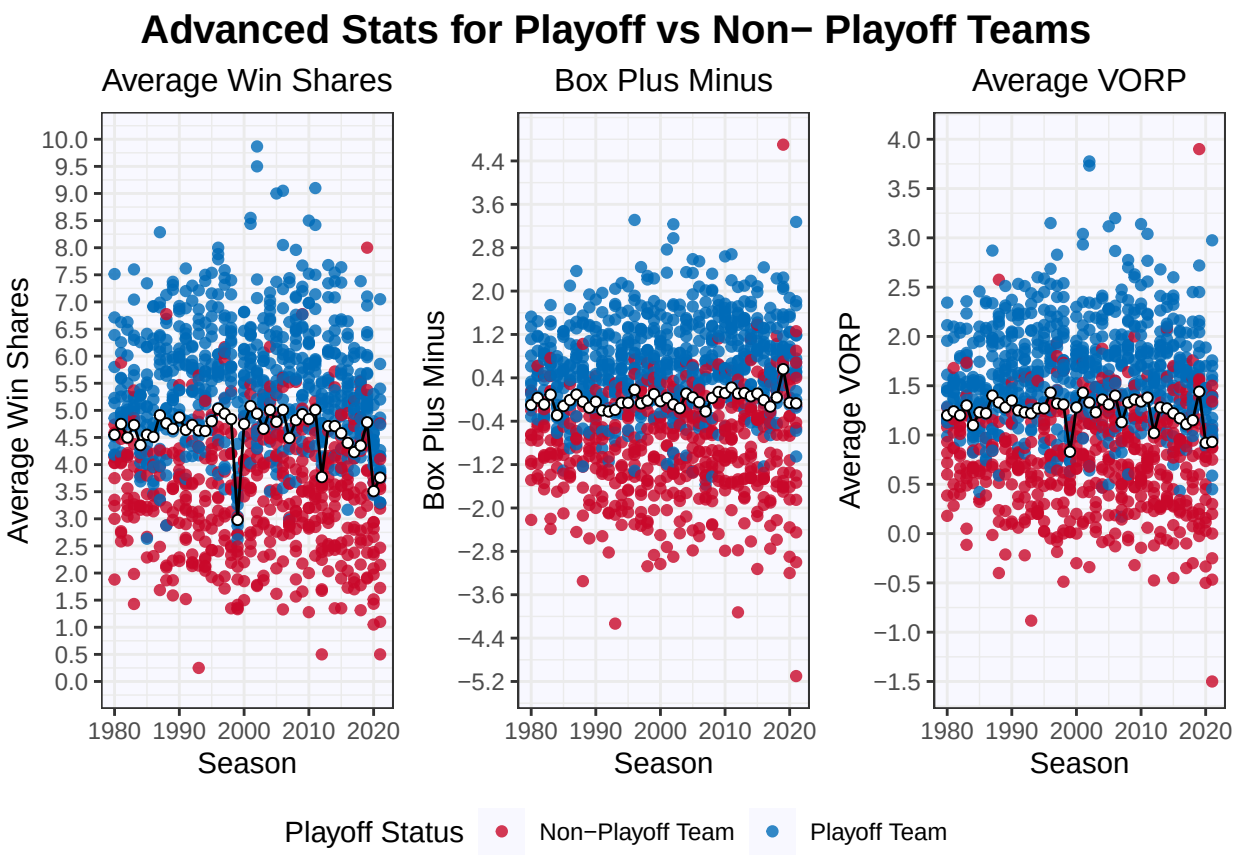
Advanced Stats Multiplot:

```
adv_stats_multiplot <- # same process as previous plots
  ggarrange(ws_plot,
            bpm_plot,
            vorp_plot,
```

```
          ncol = 3,
          common.legend = TRUE,
          legend = "bottom"
          )
```

```
## Warning: Removed 29 rows containing missing values (`geom_point()`).

## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 1 rows containing missing values (`geom_point()`).

## Warning: Removed 29 rows containing missing values (`geom_point()`).

## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 1 rows containing missing values (`geom_point()`).

## Warning: Removed 29 rows containing missing values (`geom_point()`).

## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 1 rows containing missing values (`geom_point()`).

## Warning: Removed 29 rows containing missing values (`geom_point()`).

## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 1 rows containing missing values (`geom_point()`).
```

```
annotate_figure(
  adv_stats_multiplot,
  top = text_grob("Advanced Stats for Playoff vs Non- Playoff Teams",
                  face = "bold",
                  size = 15)
) # overall title
```



Awards Visualization

```
awards_cummulative <-
  nba_data %>%
  group_by(playoffs) %>% # group by playoff status
  summarise(all_stars = sum(total_all_stars),
            All_NBA_1st = sum(All_NBA_1st_total),
            All_NBA_2nd = sum(All_NBA_2nd_total),
            All_NBA_3rd = sum(All_NBA_3rd_total),
            All_Defense_1st = sum(All_Defense_1st_total),
            All_Defense_2nd = sum(All_Defense_2nd_total),
            All_Rookie_1st = sum(All_Rookie_1st_total),
            All_Rookie_2nd = sum(All_Rookie_2nd_total))

awards_cummulative[["playoffs"]][1] <- "Non-Playoff Team" # better labels
awards_cummulative[["playoffs"]][2] <- "Playoff Team"
```

I created a new data frame which groups all of the award results by playoff status.

All Star Plot:

```
all_star_plot <-
  awards_cummulative %>%
  ggplot() +
```

```
  geom_bar( # bar plot
    aes(x = playoffs,
        y = all_stars,
        fill = playoffs),
    stat = "identity"
  ) +
  scale_fill_manual(
    values = c("#C9082A", "#006BB8") # NBA logo colors
  ) +
  labs(
    x = "Playoff Status",
    y = "",
    title = "All Stars"
  ) +
  theme_bw() +
  theme(
  plot.background=element_rect(fill = "white"),
  panel.background = element_rect(fill = "ghostwhite"),
  legend.position = "none",
  plot.title = element_text(hjust = 0.5, size = 12)
  ) # same theme elements
```

All NBA 1st Plot:

```
all_nba_1st_plot <-# same process as previous plots
  awards_cummulative %>%
  ggplot() +
  geom_bar(
    aes(x = playoffs,
        y = All_NBA_1st,
        fill = playoffs),
    stat = "identity"
  ) +
  scale_fill_manual(
    values = c("#C9082A", "#006BB8")
  ) +
  labs(
    x = "Playoff Status",
    y = "",
    title = "All NBA 1st Team"
  ) +
  theme_bw() +
  theme(
  plot.background=element_rect(fill = "white"),
  panel.background = element_rect(fill = "ghostwhite"),
  legend.position = "none",
  plot.title = element_text(hjust = 0.5, size = 12)
  )
```

All NBA 2nd Plot:

```
all_nba_2nd_plot <- # same process as previous plots
  awards_cummulative %>%
  ggplot() +
  geom_bar(
    aes(x = playoffs,
        y = All_NBA_2nd,
        fill = playoffs),
    stat = "identity"
  ) +
  scale_fill_manual(
    values = c("#C9082A", "#006BB8")
  ) +
  labs(
    x = "Playoff Status",
    y = "",
    title = "All NBA 2nd Team"
  ) +
  theme_bw() +
  theme(
  plot.background=element_rect(fill = "white"),
  panel.background = element_rect(fill = "ghostwhite"),
  legend.position = "none",
  plot.title = element_text(hjust = 0.5, size = 12)
  )
```

All NBA 3rd Plot:

```
all_nba_3rd_plot <-# same process as previous plots
  awards_cummulative %>%
  ggplot() +
  geom_bar(
    aes(x = playoffs,
        y = All_NBA_3rd,
        fill = playoffs),
    stat = "identity"
  ) +
  scale_fill_manual(
    values = c("#C9082A", "#006BB8")
  ) +
  labs(
    x = "Playoff Status",
    y = "",
    title = "All NBA 3rd Team"
  ) +
  theme_bw() +
  theme(
  plot.background=element_rect(fill = "white"),
  panel.background = element_rect(fill = "ghostwhite"),
  legend.position = "none",
  plot.title = element_text(hjust = 0.5, size = 12)
  )
```

All Defense 1st Plot:

```
all_def_1st_plot <- # same process as previous plots
  awards_cummulative %>%
  ggplot() +
  geom_bar(
    aes(x = playoffs,
        y = All_Defense_1st,
        fill = playoffs),
    stat = "identity"
  ) +
  scale_fill_manual(
    values = c("#C9082A", "#006BB8")
  ) +
  labs(
    x = "Playoff Status",
    y = "",
    title = "All Def 1st Team"
  ) +
  theme_bw() +
  theme(
  plot.background=element_rect(fill = "white"),
  panel.background = element_rect(fill = "ghostwhite"),
  legend.position = "none",
  plot.title = element_text(hjust = 0.5, size = 12)
  )
```

All Defense 2nd Plot:

```
all_def_2nd_plot <- # same process as previous plots
  awards_cummulative %>%
  ggplot() +
  geom_bar(
    aes(x = playoffs,
        y = All_Defense_2nd,
        fill = playoffs),
    stat = "identity"
  ) +
  scale_fill_manual(
    values = c("#C9082A", "#006BB8")
  ) +
  labs(
    x = "Playoff Status",
    y = "",
    title = "All Def 2nd Team"
  ) +
  theme_bw() +
  theme(
  plot.background=element_rect(fill = "white"),
  panel.background = element_rect(fill = "ghostwhite"),
  legend.position = "none",
  plot.title = element_text(hjust = 0.5, size = 12)
  )
```

All Rookie 1st Plot:

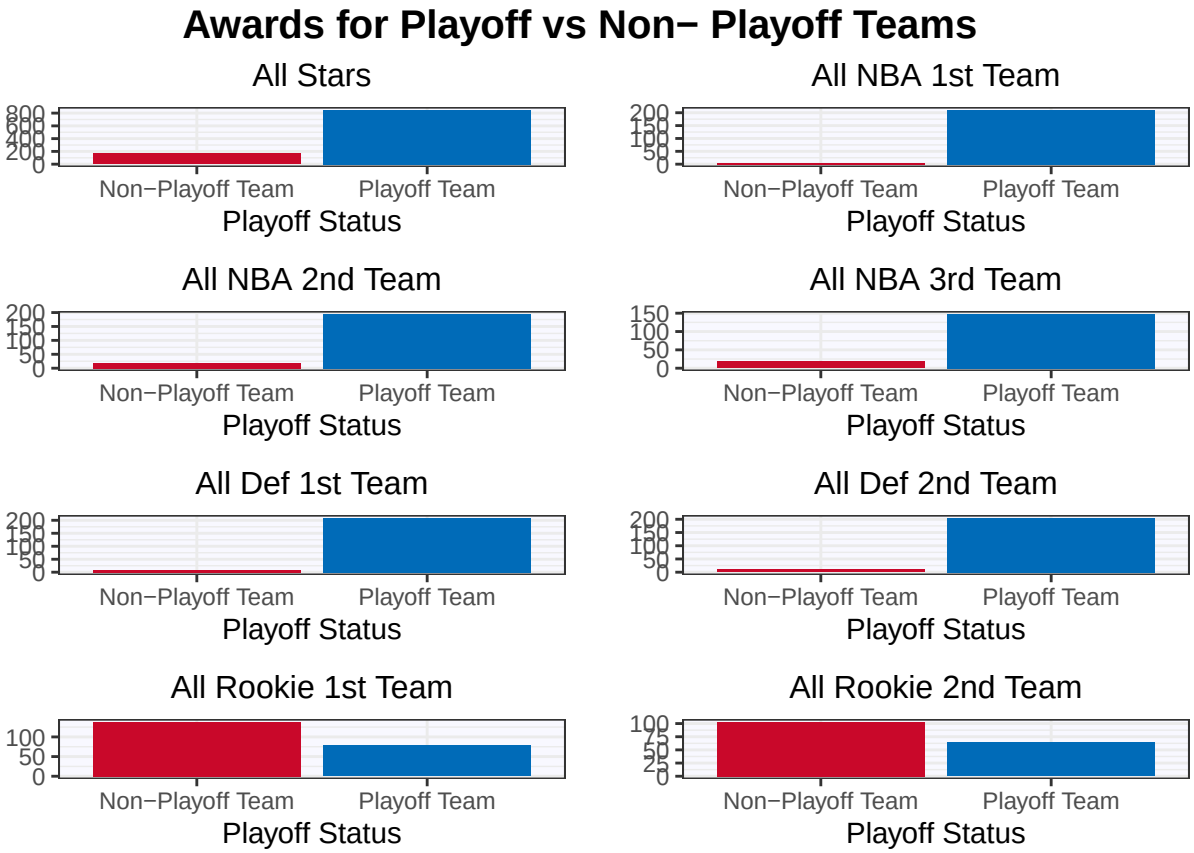
```
all_rook_1st_plot <- # same process as previous plots
  awards_cummulative %>%
  ggplot() +
  geom_bar(
    aes(x = playoffs,
        y = All_Rookie_1st,
        fill = playoffs),
    stat = "identity"
  ) +
  scale_fill_manual(
    values = c("#C9082A", "#006BB8")
  ) +
  labs(
    x = "Playoff Status",
    y = "",
    title = "All Rookie 1st Team"
  ) +
  theme_bw() +
  theme(
  plot.background=element_rect(fill = "white"),
  panel.background = element_rect(fill = "ghostwhite"),
  legend.position = "none",
  plot.title = element_text(hjust = 0.5, size = 12)
  )
```

All Rookie 2nd Plot:

```
all_rook_2nd_plot <- # same process as previous plots
  awards_cummulative %>%
  ggplot() +
  geom_bar(
    aes(x = playoffs,
        y = All_Rookie_2nd,
        fill = playoffs),
    stat = "identity"
  ) +
  scale_fill_manual(
    values = c("#C9082A", "#006BB8")
  ) +
  labs(
    x = "Playoff Status",
    y = "",
    title = "All Rookie 2nd Team"
  ) +
  theme_bw() +
  theme(
  plot.background=element_rect(fill = "white"),
  panel.background = element_rect(fill = "ghostwhite"),
  legend.position = "none",
  plot.title = element_text(hjust = 0.5, size = 12)
  )
```

Awards Multiplot

```
awards_multiplot <- # combining all award plots into one
  ggarrange(all_star_plot,
            all_nba_1st_plot,
            all_nba_2nd_plot,
            all_nba_3rd_plot,
            all_def_1st_plot,
            all_def_2nd_plot,
            all_rook_1st_plot,
            all_rook_2nd_plot,
          nrow = 4,
          ncol = 2
          )

annotate_figure(
  awards_multiplot,
  top = text_grob("Awards for Playoff vs Non- Playoff Teams",
                  face = "bold",
                  size = 15)
) # overall title
```



After seeing these visualizations and based on my own knowledge, I decided to remove awards as independent variables. There is a circular logic as players who play well often help their teams make the playoffs and get recognized with award team placement. However, award team voters also value team success when considering a player's candidacy. Therefore, the award teams are dominated by players who made the playoffs, except for the rookie teams. The rookie teams are generally made up of top picks from the previous draft. Teams who have top picks in the draft are generally bad, so they likely do not make the playoffs

anyways.

Correlation Plot

```
nba_data_num <- # numeric data only
  nba_data %>%
  filter(
    !(season %in% c(1999, 2012, 2020, 2021)) # removing outlier years
  ) %>%
  select(-team,
         -season,
         -playoffs,
         -total_all_stars,
         -All_NBA_1st_total,
         -All_NBA_2nd_total,
         -All_NBA_3rd_total,
         -All_Defense_1st_total,
         -All_Defense_2nd_total,
         -All_Rookie_1st_total,
         -All_Rookie_2nd_total) # removing non-numeric variables
```

Correlation Heatmap Function: Using the heatmap function from class.

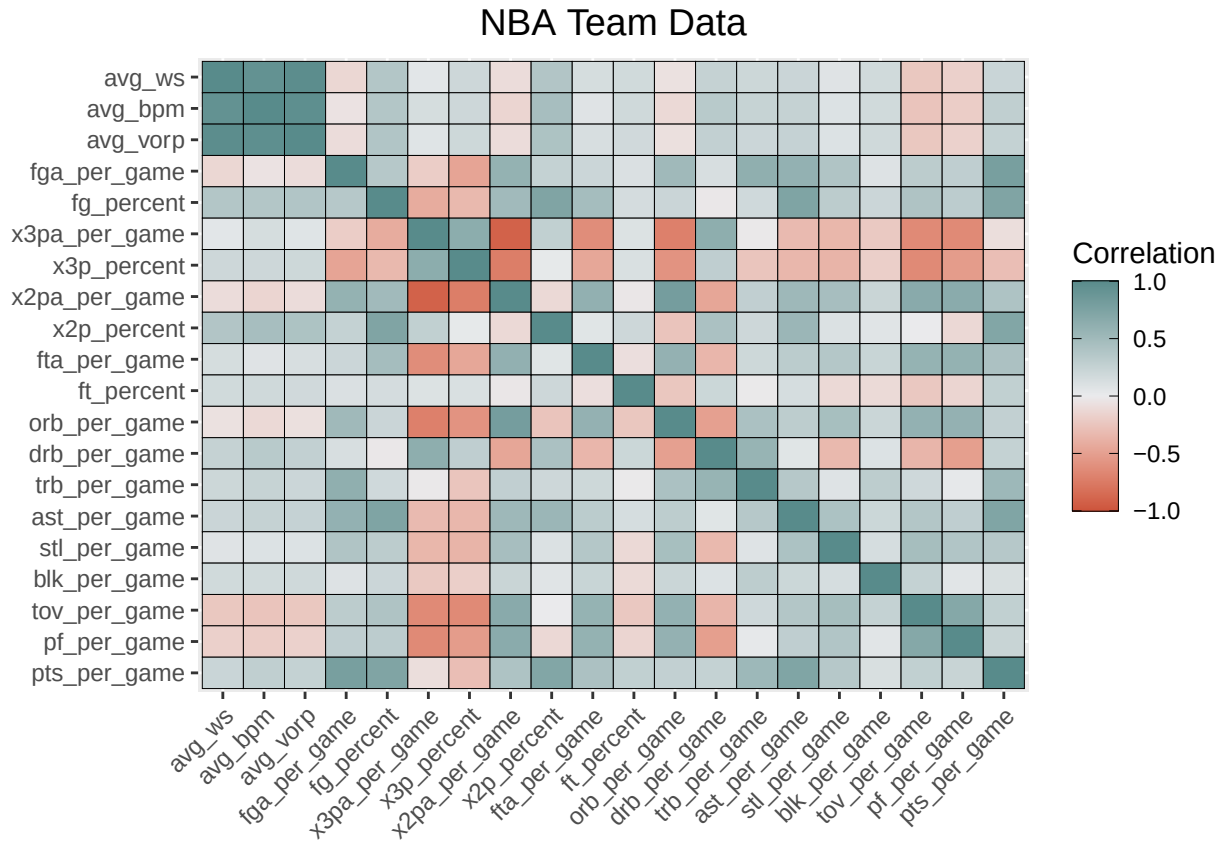
```
setup_ggplot2_heatmap <- function(
    correlation_matrix, # input for correlation matrix
    type = c("full", "lower", "upper")
    # whether matrix should depict the full, lower,
    # or upper matrix
)
{

  # Ensure correlation matrix is a `matrix` object
  corr_mat <- as.matrix(correlation_matrix)

  # Determine triangle
  if(type == "lower"){
    corr_mat[upper.tri(corr_mat)] <- NA
  }else if(type == "upper"){
    corr_mat[lower.tri(corr_mat)] <- NA
  }

  # Convert to long format
  corr_df <- data.frame(
    Var1 = rep(colnames(corr_mat), each = ncol(corr_mat)),
    Var2 = rep(colnames(corr_mat), times = ncol(corr_mat)),
    Correlation = as.vector(corr_mat)
  )

  # Set levels
  corr_df$Var1 <- factor(
    corr_df$Var1, levels = colnames(corr_mat)
  )
```

```
  corr_df$Var2 <- factor(
    corr_df$Var2, levels = rev(colnames(corr_mat))
  )
  corr_df$Correlation <- as.numeric(corr_df$Correlation)

  # Return data frame for plotting
  return(corr_df)

}
```

```
nba_lower <- setup_ggplot2_heatmap(
  cor(nba_data_num), type = "full"
) # creating correlation data
```

Correlation Heatmap Plot:

```
ggplot( # based on code from class
  data = nba_lower,
  aes(x = Var1, y = Var2, fill = Correlation)
) +
  geom_tile(color = "black") +
  scale_fill_gradient2(
    low = "#CD533B", mid = "#EAEBED",
    high = "#588B8B", limits = c(-1, 1),
    guide = guide_colorbar(
      frame.colour = "black",
      ticks.colour = "black"
    ) # fill based on correlation
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.title = element_blank(),
    plot.title = element_text(size = 14, hjust = 0.5)
  ) +
  labs(title = "NBA Team Data") # title
```



Based on the correlation plot, it is clear that there are variables that are related to eachother suggesting that PCA is viable. For example, the advanced stats are definitely a block of highly correlated variables.

Dimension Reduction - PCA

Check for Multicollinearity

```
# Obtain correlations
correlations <- cor(nba_data_num)
```

```
# Determine which variables are greater than 0.90
greater_than <- which(abs(correlations) >= 0.90, arr.ind = TRUE)
```

```
# Duplicate relationships happen because of symmetric matrix
# Also removes diagonal which equals 1
greater_than <- greater_than[
  greater_than[,"row"] < greater_than[,"col"],
]
```

```
# Replace indices with actual names
greater_than[,"row"] <- colnames(nba_data_num)[
  greater_than[,"row"]
]
```

```
greater_than[,"col"] <- colnames(nba_data_num)[
  as.numeric(greater_than[,"col"])
]
```

```
# Remove names for ease of interpretation
unname(greater_than)
```

```
##      [,1]            [,2]
## [1,] "avg_ws"        "avg_bpm"
## [2,] "avg_ws"        "avg_vorp"
## [3,] "avg_bpm"       "avg_vorp"
## [4,] "x3pa_per_game" "x2pa_per_game"
```

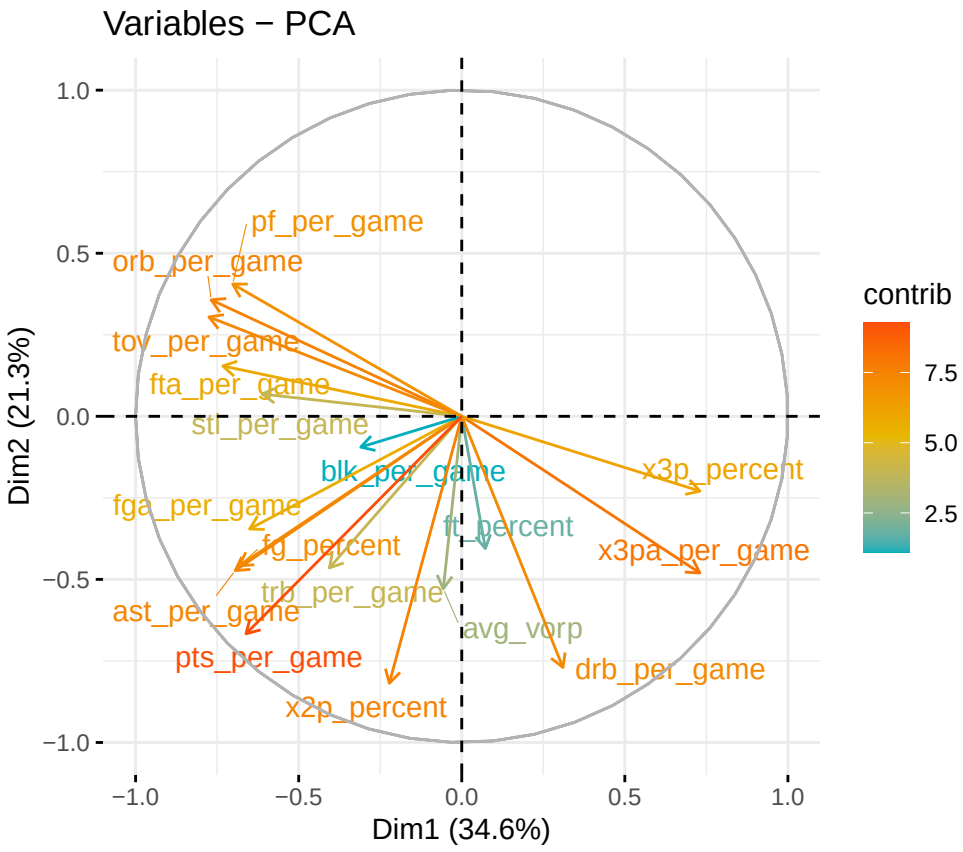
```
# remove multicollinear variables
nba_data_num_unique <-
  nba_data_num %>%
  select(
    -avg_ws,
    -avg_bpm,
    -x2pa_per_game
  )
```

Scale Variables and Visualize Data

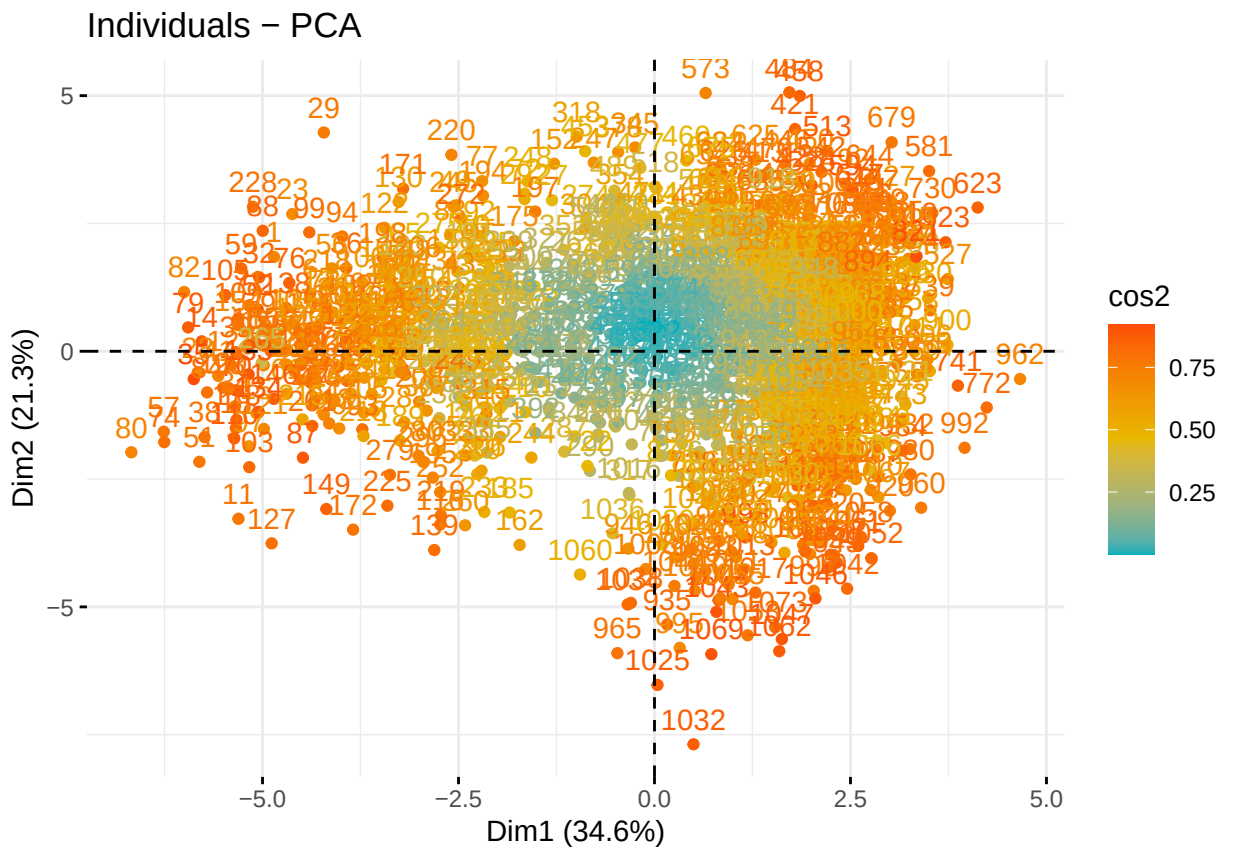
```
# run an initial pca
nba_pca <- prcomp(nba_data_num_unique, center = TRUE, scale. = TRUE)
summary(nba_pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.4258 1.9041 1.26797 1.12979 0.96193 0.90870 0.84035
## Proportion of Variance 0.3462 0.2133 0.09457 0.07508 0.05443 0.04857 0.04154
## Cumulative Proportion  0.3462 0.5594 0.65401 0.72909 0.78352 0.83209 0.87363
##                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.72173 0.66396 0.62171 0.54151 0.47779 0.39082 0.33358
## Proportion of Variance 0.03064 0.02593 0.02274 0.01725 0.01343 0.00898 0.00655
## Cumulative Proportion  0.90427 0.93020 0.95294 0.97019 0.98362 0.99260 0.99915
##                           PC15    PC16    PC17
## Standard deviation     0.11706 0.02407 0.01324
## Proportion of Variance 0.00081 0.00003 0.00001
## Cumulative Proportion  0.99996 0.99999 1.00000
```

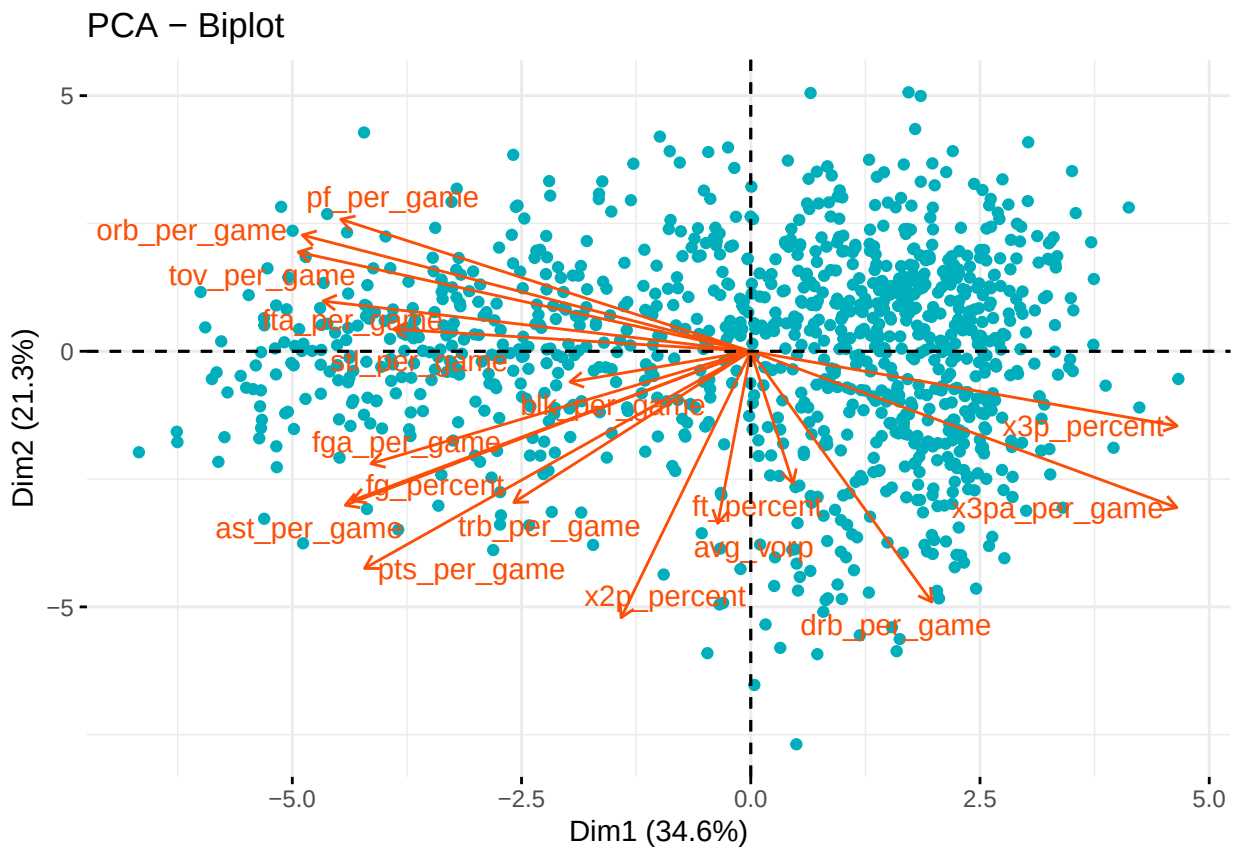
```
fviz_pca_var(
  nba_pca,
  col.var = "contrib", # Color by contributions to the PCA
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
  repel = TRUE #Avoid overlapping text if possible
)
```




```
fviz_pca_ind(
  nba_pca,
  c = "point", # Observations
  col.ind = "cos2", # Quality of representation
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
  repel = FALSE
)
```




```
# Biplot
fviz_pca_biplot(
  nba_pca, repel = TRUE,
  col.var = "#FC4E07", # Variables color
  col.ind = "#00AFBB",  # Individuals color
  label = "var" # Variables only
)
```



Bartlett's Test

```
cortest.bartlett(nba_data_num_unique)
```

```
## R was not square, finding R from data

## $chisq
## [1] 28189.08
##
## $p.value
## [1] 0
##
## $df
## [1] 136
```

Bartlett's test is significant showing that PCA is appropriate for this data. ## Conduct KMO

```
KMO(nba_data_num_unique)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = nba_data_num_unique)
## Overall MSA =  0.49
```

```
## MSA for each item =
##      avg_vorp  fga_per_game    fg_percent x3pa_per_game   x3p_percent
##          0.60          0.34          0.37          0.39          0.89
##   x2p_percent  fta_per_game    ft_percent  orb_per_game  drb_per_game
##          0.46          0.34          0.09          0.61          0.51
##  trb_per_game  ast_per_game  stl_per_game  blk_per_game  tov_per_game
##          0.45          0.97          0.82          0.92          0.81
##   pf_per_game  pts_per_game
##          0.94          0.40
```

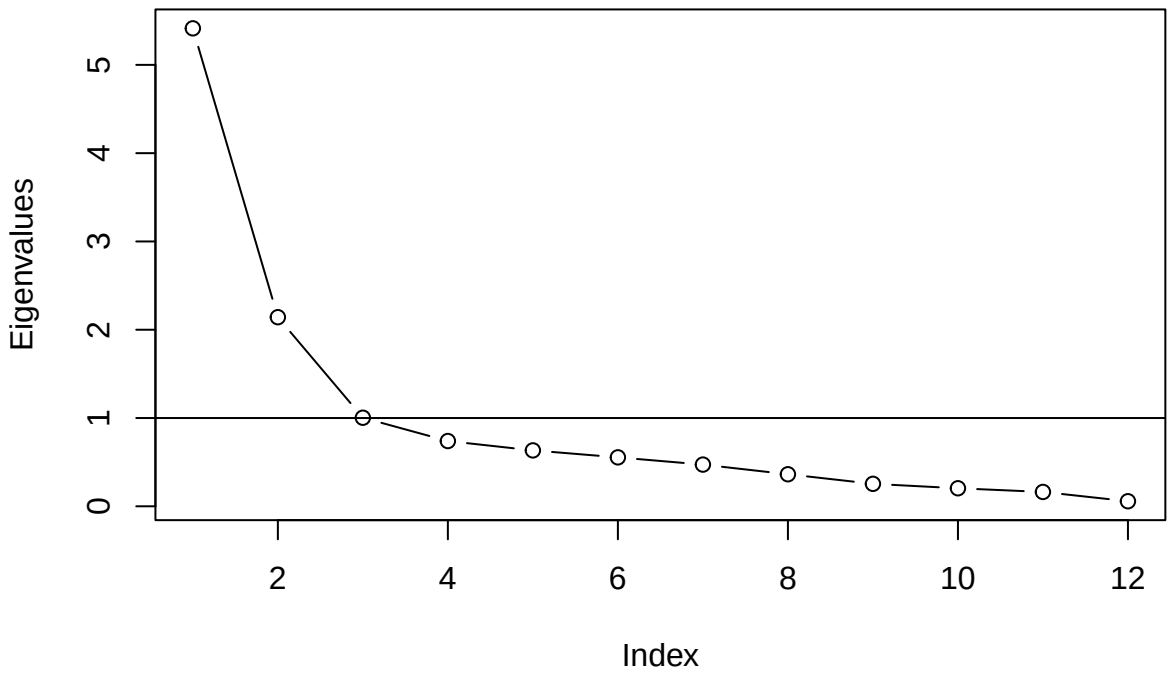
There are variables with MSA under 0.5, so I removed them one by one until only variables with MSA > 0.5 remained.

```
nba_data_num_unique <-
  nba_data_num_unique %>%
  select(
    -ft_percent,
    -trb_per_game,
    -avg_vorp,
    -fga_per_game,
    -x2p_percent
  ) # remove variables with low MSA
```

Parallel Analysis/Scree Plot

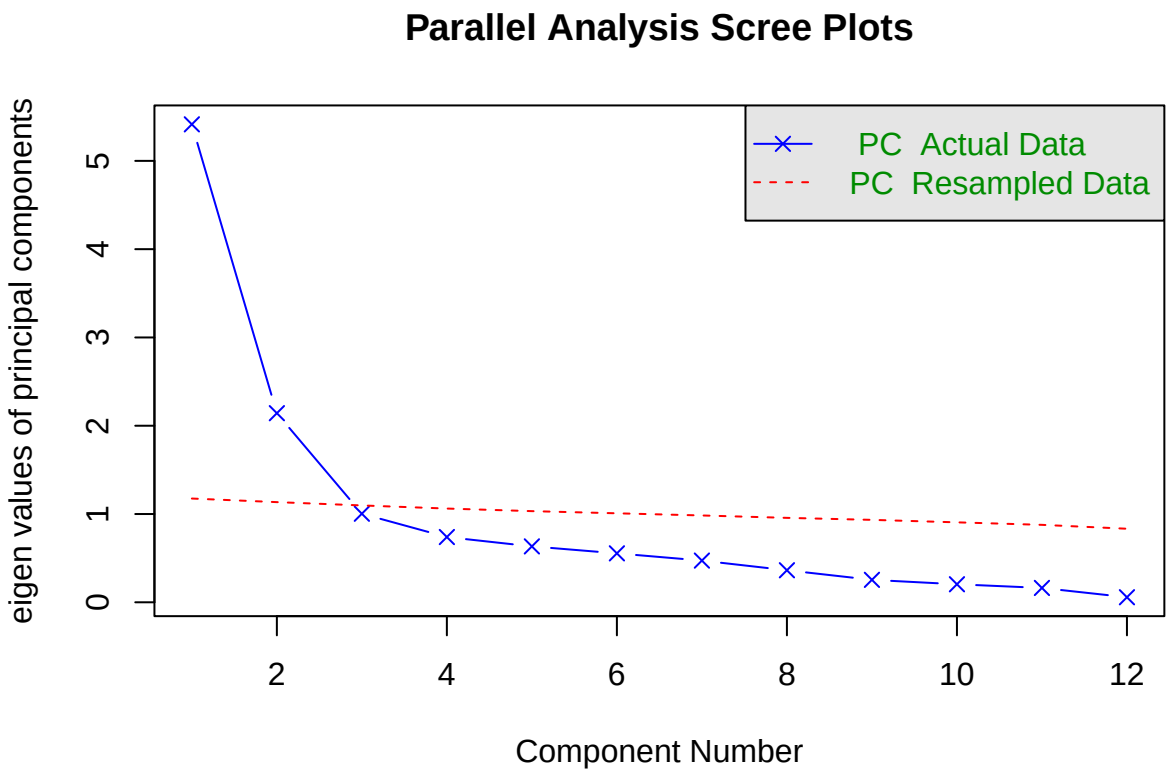
```
# compute an initial pca to check eigenvalues
initial_pca <- principal(nba_data_num_unique, nfactors = ncol(nba_data_num_unique), rotate = "oblimin")
```

```
# plot scree plot
plot(initial_pca$values, type = "b", ylab = "Eigenvalues"); abline(h = 1);
```



The scree plot shows that 2 principal components are best.

```
# PCA with {psych}
parallel_pca <- fa.parallel(
  x = nba_data_num_unique, fa = "pc",
  sim = FALSE # ensures resampling
)
```




```
## Parallel analysis suggests that the number of factors =  NA  and the number of components =  2
```

```
# PCA with {psych}
final_pca <- principal(
  r = nba_data_num_unique, nfactors = 2,
  rotate = "oblimin", # Correlated dimensions
  residuals = TRUE # Obtain residuals
)
```

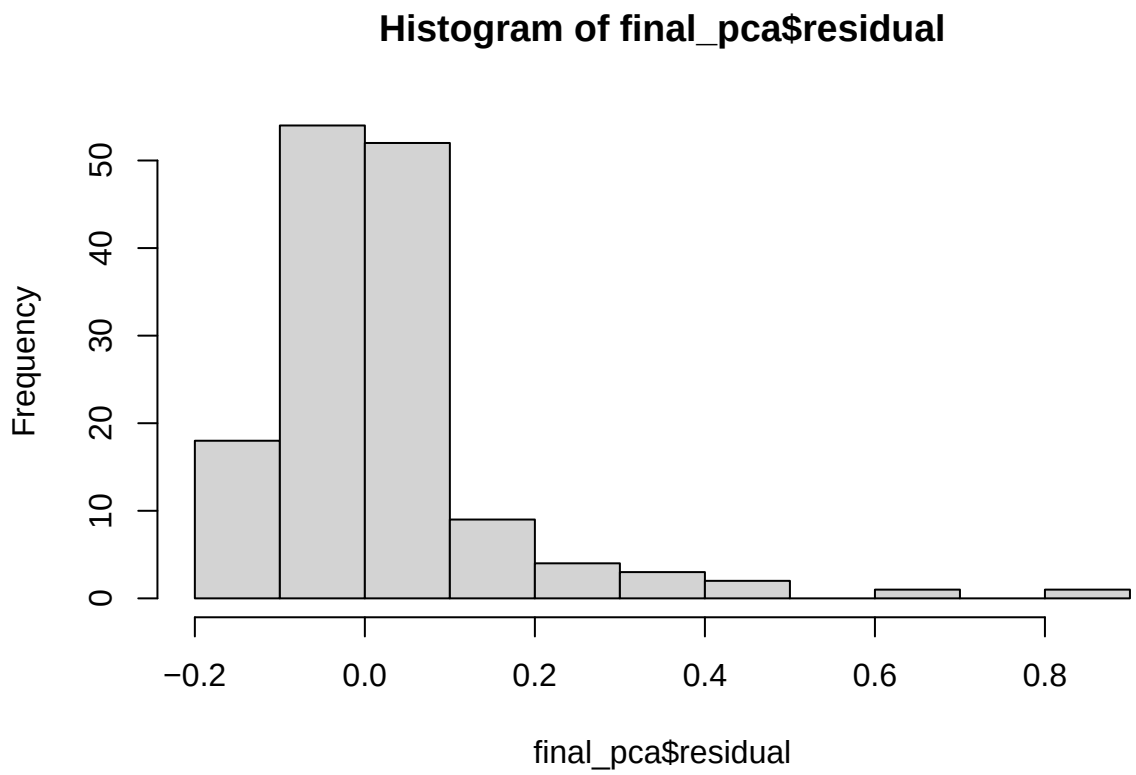
Parallel analysis also says that 2 components is best.

Check Residuals

```
# Shapiro-Wilk
shapiro.test(final_pca$residual)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  final_pca$residual
## W = 0.76218, p-value = 5.33e-14
```

```r
# Histogram
hist(final_pca$residual)
```



The shapiro test is significant meaning that the residuals are not normally distributed which is refelcted in the histogram.

```r
# Check loadings
loadings <- round(final_pca$loadings[,1:2], 3)

# For interpretation, set less than 0.30 to ""
loadings[abs(loadings) < 0.30] <- ""

# Print loadings
as.data.frame(loadings)
```

```
##                  TC1   TC2
## fg_percent           0.814
## x3pa_per_game -0.888
## x3p_percent   -0.674
## fta_per_game   0.654
## orb_per_game   0.814
## drb_per_game  -0.823 0.467
## ast_per_game         0.847
## stl_per_game   0.473
## blk_per_game
```

```
## tov_per_game    0.75
## pf_per_game    0.804
## pts_per_game         0.928
```

The loadings show which variables contribute (and don't contribute) most to each of the two principal components. I used ChatGPT to come up with names for the dimensions which I called "Aggressiveness" and "Offensive_Skill".

Obtain Scores

```
pca_scores <- final_pca$scores # obtaining scores
```

```
pca_scores <- as.data.frame(pca_scores) # convert to data frame
```

```
colnames(pca_scores) <-
  c("Aggressiveness", "Offensive_Skill") # rename dimensions
```

```
nba_data_remove_outliers <-
  nba_data %>%
  filter(
    !(season %in% c(1999, 2012, 2020, 2021)) # remove outlier seasons
  )

playoffs <- nba_data_remove_outliers$playoffs # playoff status
```

```
scores_final <-
  pca_scores %>%
  mutate(
    playoff_status = factor(playoffs, levels = c(FALSE, TRUE))
  ) # add playoff status to scores
```

Aggressiveness Plot:

```
aggress_plot <-
scores_final %>%
  ggplot() +
  geom_density( # density plot
    aes(
      x = Aggressiveness,
      fill = playoff_status,
    ),
    alpha = 0.65 # transparency
  ) +
  scale_fill_manual(
    labels = c("Non-Playoff", "Playoff"),
    values = c("#C9082A", "#006BB8") # NBA logo colors
  ) +
  labs(
    x = "Aggressiveness",
    y = "Density",
```

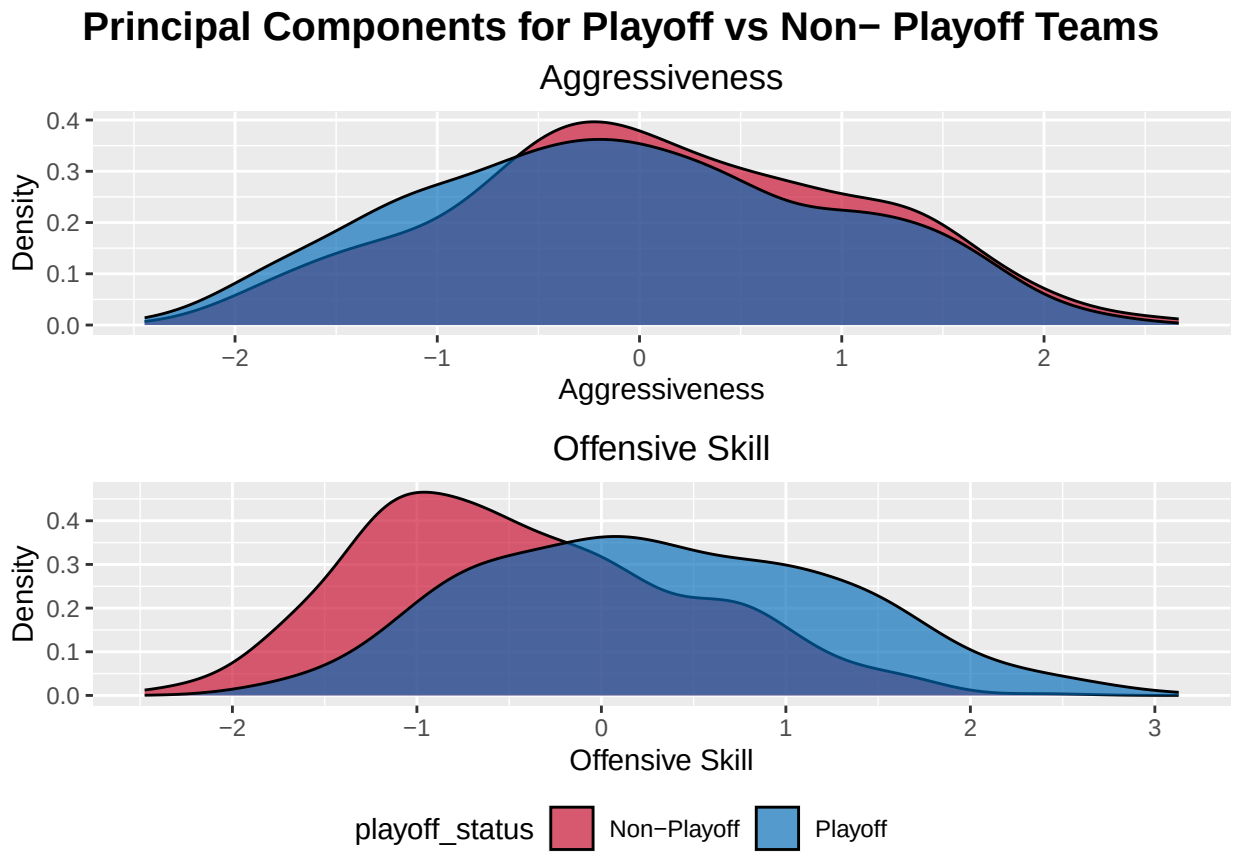
```
    title = "Aggressiveness"
  ) +
  theme(
    plot.title = element_text(hjust = 0.5)
  )
```

Offensive Skill Plot:

```
off_skill_plot <- # same as previous plot
scores_final %>%
  ggplot() +
  geom_density(
    aes(
      x = Offensive_Skill,
      fill = playoff_status,
    ),
    alpha = 0.65
  ) +
  scale_fill_manual(
    labels = c("Non-Playoff", "Playoff"),
    values = c("#C9082A", "#006BB8")
  ) +
  labs(
    x = "Offensive Skill",
    y = "Density",
    title = "Offensive Skill"
  ) +
  theme(
    plot.title = element_text(hjust = 0.5)
  )
```

Principal Components Multiplot:

```
pc_multiplot <- # combine plots into one
  ggarrange(
    aggress_plot,
    off_skill_plot,
    common.legend = TRUE, # common legend
    legend = "bottom",
    nrow = 2
  )

annotate_figure(
  pc_multiplot,
  top = text_grob("Principal Components for Playoff vs Non- Playoff Teams",
                  face = "bold",
                  size = 15)
) # common title
```



Logistic Regression

Check Multicollinearity

```
cor(scores_final[,1:2]) # check correlations
```

```
##                 Aggressiveness Offensive_Skill
## Aggressiveness       1.0000000       0.2423981
## Offensive_Skill      0.2423981       1.0000000
```

The principal components are not highly correlated.

Balance Playoff Status

```
table(scores_final$playoff_status) # check distribution of playoff status
```

```
##
## FALSE  TRUE
##   466   608
```

```r
set.seed(1234)
scores_final_balanced <-
  scores_final[
  c(
    # Keep high value category
    which(scores_final$playoff_status == FALSE),
    # Randomly sample low value category
    sample(
      which(scores_final$playoff_status == TRUE),
      466
    )
  ),
]
```

```r
set.seed(1234) # set seed

train_index <- sample(
  1:nrow(scores_final_balanced),
  round(nrow(scores_final_balanced) * 0.70)
) # training data indices

test_index <- setdiff(
  1:nrow(scores_final_balanced),
  train_index
) # testing data indicies
```

```r
table(scores_final_balanced[train_index,]$playoff_status)
```

```
##
## FALSE  TRUE
##   328   324
```

```r
table(scores_final_balanced[test_index,]$playoff_status)
```

```
##
## FALSE  TRUE
##   138   142
```

```r
# ensuring playoff status is still balanced in training and testin data
```

Run Logistic Regression

```r
nba_lrm <-
  lrm(
    playoff_status ~ Aggressiveness + Offensive_Skill,
    data = scores_final_balanced[train_index,]
  ) # logistic regression on training data
```

```
nba_lrm # results
```

```
## Logistic Regression Model
##
## lrm(formula = playoff_status ~ Aggressiveness + Offensive_Skill,
##     data = scores_final_balanced[train_index, ])
##
##                       Model Likelihood    Discrimination    Rank Discrim.
##                             Ratio Test           Indexes          Indexes
## Obs           652    LR chi2     132.39    R2       0.245    C        0.750
##  FALSE        328    d.f.             2    R2(2,652)0.181    Dxy      0.500
##  TRUE         324    Pr(> chi2) <0.0001    R2(2,489)0.234    gamma    0.500
## max |deriv| 1e-07                          Brier    0.203    tau-a    0.250
##
##                 Coef    S.E.   Wald Z Pr(>|Z|)
## Intercept        0.0559 0.0873  0.64  0.5225
## Aggressiveness  -0.3780 0.0937 -4.03  <0.0001
## Offensive_Skill  1.0761 0.1067 10.09  <0.0001
```

```
exp(nba_lrm$coefficients) # log of odds ratio
```

```
##       Intercept  Aggressiveness Offensive_Skill
##       1.0574418       0.6852472       2.9331613
```

```
# Regular logistic for predictions
nba_logm <- glm(
  playoff_status ~ Aggressiveness + Offensive_Skill,
  data = scores_final_balanced[train_index,],
  family = "binomial"
)
```

Predict Training and Testing Data

```
train_predicted <- factor(
  ifelse(predict(nba_logm) > 0, TRUE, FALSE)
) # predict data and classify
```

```
test_predicted <- factor(
  ifelse(predict(
    nba_logm,
    newdata = scores_final_balanced[test_index,]
  ) > 0, TRUE, FALSE)
) # predict testing data and classify
```

Comparing Predictions to Ground Truth Playoff Results

```r
confusionMatrix(data = train_predicted,  positive = "TRUE",
                reference = factor(scores_final_balanced$playoff_status[train_index]))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE   224  107
##      TRUE    104  217
##
##                Accuracy : 0.6764
##                  95% CI : (0.639, 0.7122)
##     No Information Rate : 0.5031
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.3527
##
##  Mcnemar's Test P-Value : 0.8905
##
##             Sensitivity : 0.6698
##             Specificity : 0.6829
##          Pos Pred Value : 0.6760
##          Neg Pred Value : 0.6767
##              Prevalence : 0.4969
##          Detection Rate : 0.3328
##    Detection Prevalence : 0.4923
##       Balanced Accuracy : 0.6763
##
##        'Positive' Class : TRUE
##
```

```r
# confusion matrix of training data
```

```r
confusionMatrix(data = test_predicted,  positive = "TRUE",
                reference = factor(scores_final_balanced$playoff_status[test_index]))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE    99   51
##      TRUE     39   91
##
##                Accuracy : 0.6786
##                  95% CI : (0.6204, 0.7329)
##     No Information Rate : 0.5071
##     P-Value [Acc > NIR] : 4.663e-09
##
##                   Kappa : 0.3578
##
##  Mcnemar's Test P-Value : 0.2463
##
```

```
##             Sensitivity : 0.6408
##             Specificity : 0.7174
##          Pos Pred Value : 0.7000
##          Neg Pred Value : 0.6600
##              Prevalence : 0.5071
##          Detection Rate : 0.3250
##    Detection Prevalence : 0.4643
##       Balanced Accuracy : 0.6791
##
##        'Positive' Class : TRUE
##
```

```
# confusion matrix of testing data
```

Discussion

Looking at the initial visualizations of the data, there are clear relationships between basic counting stats and playoff status. In the scatter plots, the black line indicates the league average for the season. In general, playoff teams scored more points and had more assists than the league average. Rebounds, blocks, and steals provided less clear relationships, but playoff teams generally turned the ball over less. For the advanced stats, it was more clear that playoff teams generally performed above average in these statistics. Finally, the All Star roster and All NBA and All Defensive teams were dominated by playoff teams. As I said before, this may be a chicken and egg problem where players are rewarded for making the playoffs, but also good players lead their teams to the playoffs. The rookie teams generally have players from bad teams, so it makes sense that these awards are dominated by non-playoff teams.

The correlation plot shows that there are strong postive and negative correlations between many of the variables. The advanced stats are particularly correlated, and many of the jump shooting related stats like field goal percent and 2 point field goal percent are also related. This plot shows that the data likely lends itself towards dimension reduction through PCA. The PCA plots showed that there were not many strongly orthogonal variables, and the individual PCA plot and biplot showed the strong variance in the data. Importantly, Bartlett's test was signifcant showing that the data was suitable for PCA. I used the KMO process to remove variables one by one with low MSA values. The scree plot and parallel analysis agreed that 2 dimensions are best to model the NBA data. 2 dimensions also accounts for over 50% of the variance. The residuals of the model were not regular, but the loadings showed interesting results for the principal components.

Dimension 1 was characterized by offensive rebounding, personal fouls, turnovers, free throw attempts, and steals. It was strongly not characterized by three point attempts and percent and defensive rebounding. Using ChatGPT and my intuition, I classified this dimension as aggression as teams who play aggressively can get lots of offensive rebounds and steals but also commit many fouls and turnovers. They are also likely teams who shoot the ball very well.

Dimension 2 was characterized by points per game, field goal percentage, assists per game, and defensive rebounds per game. Using ChatGPT and my intuition, I classified this dimension as Offensive Skill as teams who shoot, score, and assit well have to be highly skilled.

Using the results of the principal components, I created overlapping density plots. Playoff and non-Playoff teams do not differ strongly in aggressiveness, but playoff teams seem to be more offensively skilled than non-Playoff teams.

Finally, I conducted a logistic regression using the principal components to see if they can predict playoff status. I balanced the samples of the data and did a 70/30 training/testing split. From the regression, I learned that a one standard deviation in aggressiveness is related to about a 1.5 times less chance of making

the playoffs. A one standard deviation increase in offensive skill is correlated with nearly a 3 times higher chance of making the playoffs.

The confusion matrices for the the training and testing data set were similar with the model having an overall accuracy of about 68%. The model had similar specificity and sensitivity values, and it had a kappa of about 0.35. Overall, the model was moderately successful at predicting which teams would make the playoffs. I hypothesized that playoff teams would score more points and have more assists which was captured in the offensive skill dimension. However, I would not have predicted that aggressiveness, especially offensive rebounding and steals would not be correlated with making the playoffs.

My results are very interesting to me as a basketball, and they may be somewhat useful for NBA teams when constructing their rosters. Teams can look to find players who posses strong offensive skill, but it is important to consider other parts of the game such as height, weight, and age of players. In the future, I would like to conduct a similar analysis, but I would look to do it on a player scale rather than a team scale.