NBA Finals Exploratory Data Analysis: Championships vs Runner-Ups

Kesar Singh Sidhu

2023-1-15

Introduction

Data Source: The dataset used in this analysis has been obtained from Kaggle, with credit to Dave Rosenman as the dataset author. This dataset was sourced by scraping game-by-game statistics from basketball-reference.com. It contains detailed information on NBA Finals team statistics from the 1980-2018 seasons, segmented into separate CSV files for championship-winning teams (champsdata.csv) and runner-ups (runnerupsdata.csv).

Background: The NBA Finals represents the pinnacle of excitement in an NBA season, showcasing intense matchups between the top teams from the Western and Eastern Conferences battling through a potential seven-game series. Since the 1947 season, this event has captured the historical triumphs of championship-winning teams and the heartbreaks of those who fell short during the NBA Playoffs.

Goal: In this analysis, the primary focus is to analyze the evolving trends in play styles of championship-winning NBA Finals teams and examine the influence of home-court advantage on team performances. To grasp the project's bigger picture, we can ask ourselves smaller specific questions:

- In which statistical categories did NBA Championship teams consistently outperform the runner-ups in NBA Finals game?
- Which team had the most success, what team came up short the most. How do they compare?
- How does home-court advantage impact NBA Finals performance?
- Over-time, how has the play style of winning championships evolved, more offense centered or defensive minded?

Usefulness: This analysis provides valuable insights into why some NBA teams win championships while others fall short. It helps teams understand how the NBA has changed over time and what they need to focus on. This can include areas such as improving play strategies and reducing mistakes like personal fouls and turnovers. Coaches can use these insights to guide their teams, while analysts and fans get a better grasp of the game's evolution.

Preparing the Environment

To start this project, lets import the necessary libraries and read in the datasets crucial for our analysis.

We will be using:

- Tidyverse - *Dplyr, Tidyr, and Ggplot2*

- Forcats - *Manage and manipulate categorical variables*
- Grid Extra - *Construct and manipulate the layout of our visualizations*

```
# Import the libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(forcats)
library(gridExtra)
```

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

```
# Read in the datasets
nba_championships.df <- read_csv("/Users/kesarsidhu/R Studio/EDA/Data-Sets/NBA Championships/championsda

nba_runnerUps.df <- read_csv("/Users/kesarsidhu/R Studio/EDA/Data-Sets/NBA Championships/runnerupsdata.
```

Process

(Data Cleaning and Manipulation)

Steps:fix any missing or incorrect values, after cleaning merge data sets

After importing the datasets, it's essential to start by examining their summary statistics, such as variables, data types, column and row numbers, among other details.

Each dataframe contains 220 rows and 24 columns.

```
# Preview each dataframe
glimpse(nba_championships.df)
```

```
## Rows: 220
## Columns: 24
## $ Year <dbl> 1980, 1980, 1980, 1980, 1980, 1980, 1981, 1981, 1981, 1981, 1981,~
## $ Team <chr> "Lakers", "Lakers", "Lakers", "Lakers", "Lakers", "Lakers", "Celt~
## $ Game <dbl> 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4,~
## $ Win  <dbl> 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1,~
## $ Home <dbl> 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0,~
## $ MP   <dbl> 240, 240, 240, 240, 240, 240, 240, 240, 240, 240, 240, 240, 240, ~
## $ FG   <dbl> 48, 48, 44, 44, 41, 45, 41, 41, 40, 35, 41, 43, 49, 35, 50, 45, 4~
## $ FGA  <dbl> 89, 95, 92, 93, 91, 92, 95, 82, 89, 74, 94, 78, 93, 83, 91, 97, 1~
## $ FGP  <dbl> 0.539, 0.505, 0.478, 0.473, 0.451, 0.489, 0.432, 0.500, 0.449, 0.~
## $ TP   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,~
## $ TPA  <dbl> 0, 1, 1, 0, 0, 2, 1, 3, 3, 3, 3, 4, 0, 5, 1, 1, 2, 0, 0, 1, 1, 1,~
## $ TPP  <dbl> NA, 0.000, 0.000, NA, NA, 0.000, 0.000, 0.000, 0.667, 0.000, 0.00~
## $ FT   <dbl> 13, 8, 23, 14, 26, 33, 16, 8, 12, 16, 27, 15, 26, 24, 28, 21, 8, ~
## $ FTA  <dbl> 15, 12, 30, 19, 33, 35, 20, 13, 19, 24, 35, 18, 35, 37, 47, 29, 1~
## $ FTP  <dbl> 0.867, 0.667, 0.767, 0.737, 0.788, 0.943, 0.800, 0.615, 0.632, 0.~
## $ ORB  <dbl> 12, 15, 22, 18, 19, 17, 25, 14, 16, 17, 19, 9, 19, 17, 17, 16, 26~
## $ DRB  <dbl> 31, 37, 34, 31, 37, 35, 29, 34, 28, 30, 35, 28, 31, 22, 31, 33, 2~
## $ TRB  <dbl> 43, 52, 56, 49, 56, 52, 54, 48, 44, 47, 54, 37, 50, 39, 48, 49, 4~
## $ AST  <dbl> 30, 32, 20, 23, 28, 27, 23, 17, 24, 22, 25, 26, 34, 25, 30, 35, 3~
## $ STL  <dbl> 5, 12, 5, 12, 7, 14, 6, 6, 12, 5, 5, 6, 11, 11, 15, 10, 5, 12, 11~
## $ BLK  <dbl> 9, 7, 5, 6, 6, 4, 5, 7, 6, 6, 8, 0, 7, 6, 5, 4, 9, 11, 13, 6, 2, ~
## $ TOV  <dbl> 17, 26, 20, 19, 21, 17, 19, 22, 11, 22, 14, 13, 22, 18, 18, 12, 2~
## $ PF   <dbl> 24, 27, 25, 22, 27, 22, 21, 27, 25, 22, 23, 21, 26, 21, 30, 21, 2~
## $ PTS  <dbl> 109, 104, 111, 102, 108, 123, 98, 90, 94, 86, 109, 102, 124, 94, ~
```

```
glimpse(nba_runnerUps.df)
```

```
## Rows: 220
## Columns: 24
## $ Year <dbl> 1980, 1980, 1980, 1980, 1980, 1980, 1981, 1981, 1981, 1981, 1981,~
## $ Team <chr> "Sixers", "Sixers", "Sixers", "Sixers", "Sixers", "Sixers", "Rock~
## $ Game <dbl> 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4,~
## $ Win  <dbl> 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0,~
## $ Home <dbl> 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1,~
## $ MP   <dbl> 240, 240, 240, 240, 240, 240, 240, 240, 240, 240, 240, 240, 240, ~
## $ FG   <dbl> 40, 43, 45, 41, 42, 47, 42, 34, 24, 37, 30, 36, 48, 49, 39, 44, 5~
## $ FGA  <dbl> 90, 85, 93, 79, 94, 89, 99, 85, 79, 103, 84, 86, 98, 93, 88, 91, ~
## $ FGP  <dbl> 0.444, 0.506, 0.484, 0.519, 0.447, 0.528, 0.424, 0.400, 0.304, 0.~
## $ TP   <dbl> 0, 0, 1, 0, 0, 0, 0, 2, 0, 1, 0, 0, 3, 0, 1, 0, 0, 0, 0, 2, 0, 1,~
## $ TPA  <dbl> 2, 1, 4, 0, 3, 6, 2, 2, 1, 3, 1, 2, 6, 2, 1, 3, 0, 2, 2, 7, 3, 3,~
## $ TPP  <dbl> 0.000, 0.000, 0.250, NA, 0.000, 0.000, 0.000, 1.000, 0.000, 0.333~
## $ FT   <dbl> 22, 21, 10, 23, 19, 13, 11, 22, 23, 16, 20, 19, 18, 12, 29, 13, 2~
## $ FTA  <dbl> 28, 27, 17, 26, 24, 22, 14, 32, 31, 22, 33, 23, 23, 21, 40, 20, 3~
## $ FTP  <dbl> 0.786, 0.778, 0.588, 0.885, 0.792, 0.591, 0.786, 0.688, 0.742, 0.~
## $ ORB  <dbl> 14, 5, 13, 5, 13, 7, 19, 13, 19, 28, 15, 18, 18, 20, 14, 11, 13, ~
## $ DRB  <dbl> 26, 29, 24, 29, 29, 29, 23, 22, 29, 21, 26, 23, 23, 32, 29, 29, 2~
## $ TRB  <dbl> 40, 34, 37, 34, 42, 36, 42, 35, 48, 49, 41, 41, 41, 52, 43, 40, 3~
## $ AST  <dbl> 28, 34, 34, 31, 32, 27, 23, 16, 10, 22, 15, 22, 28, 28, 25, 32, 3~
## $ STL  <dbl> 12, 14, 12, 5, 9, 4, 15, 6, 6, 8, 8, 4, 11, 4, 10, 3, 14, 11, 7, ~
## $ BLK  <dbl> 13, 11, 8, 10, 7, 11, 3, 8, 10, 2, 9, 2, 7, 9, 8, 7, 13, 7, 8, 10~
## $ TOV  <dbl> 14, 20, 13, 14, 12, 18, 10, 9, 21, 10, 17, 12, 18, 21, 19, 16, 11~
## $ PF   <dbl> 17, 21, 25, 20, 25, 27, 20, 17, 19, 20, 24, 21, 26, 30, 36, 23, 1~
## $ PTS  <dbl> 102, 107, 101, 105, 103, 107, 95, 92, 71, 91, 80, 91, 117, 110, 1~
```

Now that we've got a clearer picture of how these datasets are organized, it's time to tackle the Data Cleaning and Manipulation phase! Given the similar structures of both datasets, it's a good idea to clean and prepare them separately before merging them later on.

We'll start by working on the **NBA Championship dataset!**

To begin our cleaning stage, we should first locate any NA values from the NBA Championship dataset.

```
# Filter throughout all rows of where any variable contains at least one piece of NA
nba_championships.df %>% filter_all(any_vars(is.na(.)))
```

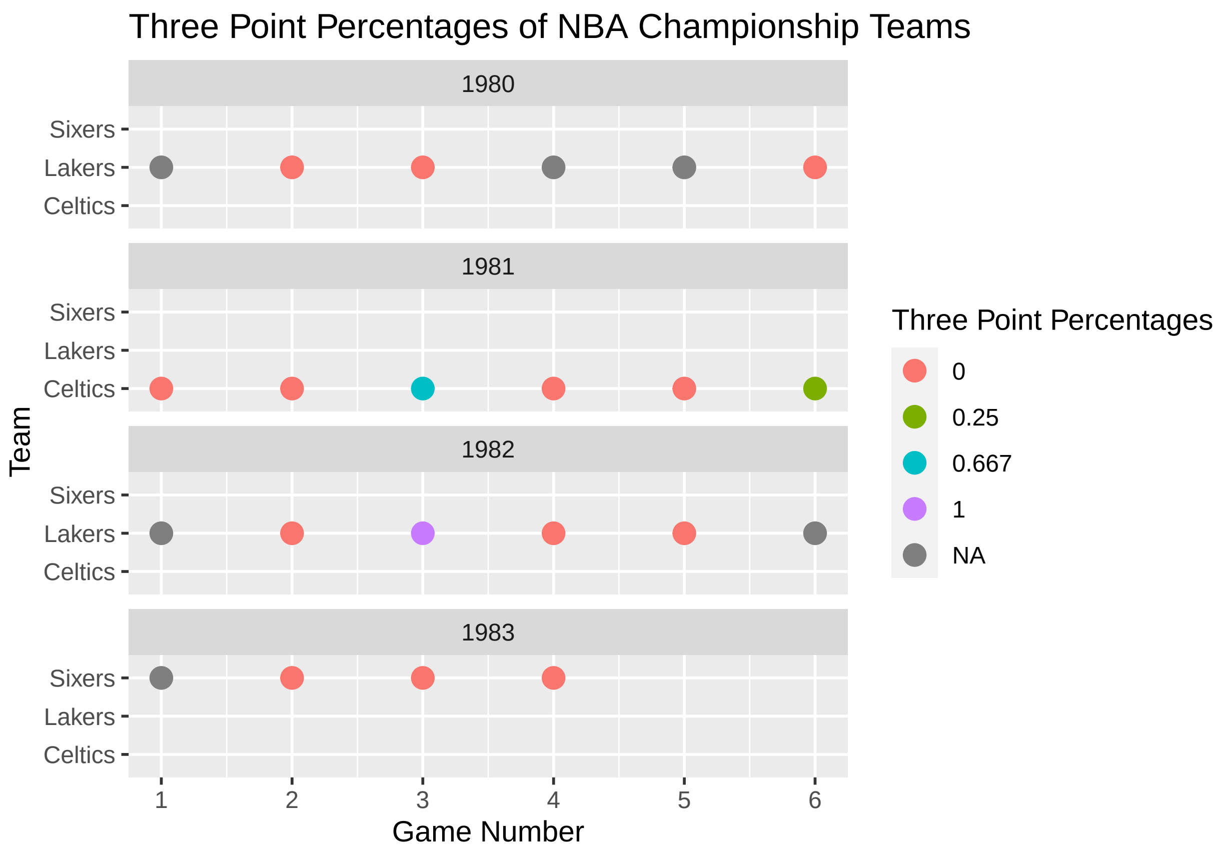
```
## # A tibble: 6 x 24
##    Year Team    Game   Win  Home    MP    FG   FGA   FGP    TP   TPA   TPP    FT
##   <dbl> <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  1980 Lakers     1     1     1   240    48    89 0.539     0     0    NA    13
## 2  1980 Lakers     4     0     0   240    44    93 0.473     0     0    NA    14
## 3  1980 Lakers     5     1     1   240    41    91 0.451     0     0    NA    26
## 4  1982 Lakers     1     1     0   240    49    93 0.527     0     0    NA    26
## 5  1982 Lakers     6     1     1   240    47    87 0.54      0     0    NA    20
## 6  1983 Sixers     1     1     1   240    45    96 0.469     0     0    NA    23
## # i 11 more variables: FTA <dbl>, FTP <dbl>, ORB <dbl>, DRB <dbl>, TRB <dbl>,
## #   AST <dbl>, STL <dbl>, BLK <dbl>, TOV <dbl>, PF <dbl>, PTS <dbl>
```

After filtering through the dataset, we discover there are 6 rows of missing data, all from the Three Point Percentages column (TPP):

- Lakers 1980 Finals Games 1,4,and 5
- Lakers 1982 Finals Games 1 and 6
- Lakers 1983 Finals Game 1

Creating a visualization would help display this.

```
# Creating visualization of Three Point Percentages between 1980 - 1983
nba_championships.df %>%
  filter(between(Year, 1980, 1983)) %>%
  ggplot(aes(Game, Team, color = factor(TPP))) +
  geom_point(na.rm = FALSE, size = 3.5)+
  facet_wrap(~Year, ncol = 1)+
  scale_x_continuous(breaks = c(1,2,3,4,5,6))+
  labs(
    x = "Game Number",
    color = "Three Point Percentages",
    title = "Three Point Percentages of NBA Championship Teams"
  )
```



Now to fix this, using the fill() function from the Tidyr package can help fill the NA values.

To decide what values to replace the NA's with, we can safely place a 0 for all of them since each row shows 0 Three Point Attempts taken.

In order to correctly fill each NA row with 0, using the direction "updown" will let R consider both up and down values of the NA value, and choose the closest non-missing value. Since each NA value has at least a 0 in either direction will replace the NA values with 0.

Lastly, we save these changes into a new data set named "nba_championships_cleaned.df" to avoid altering the original dataset.

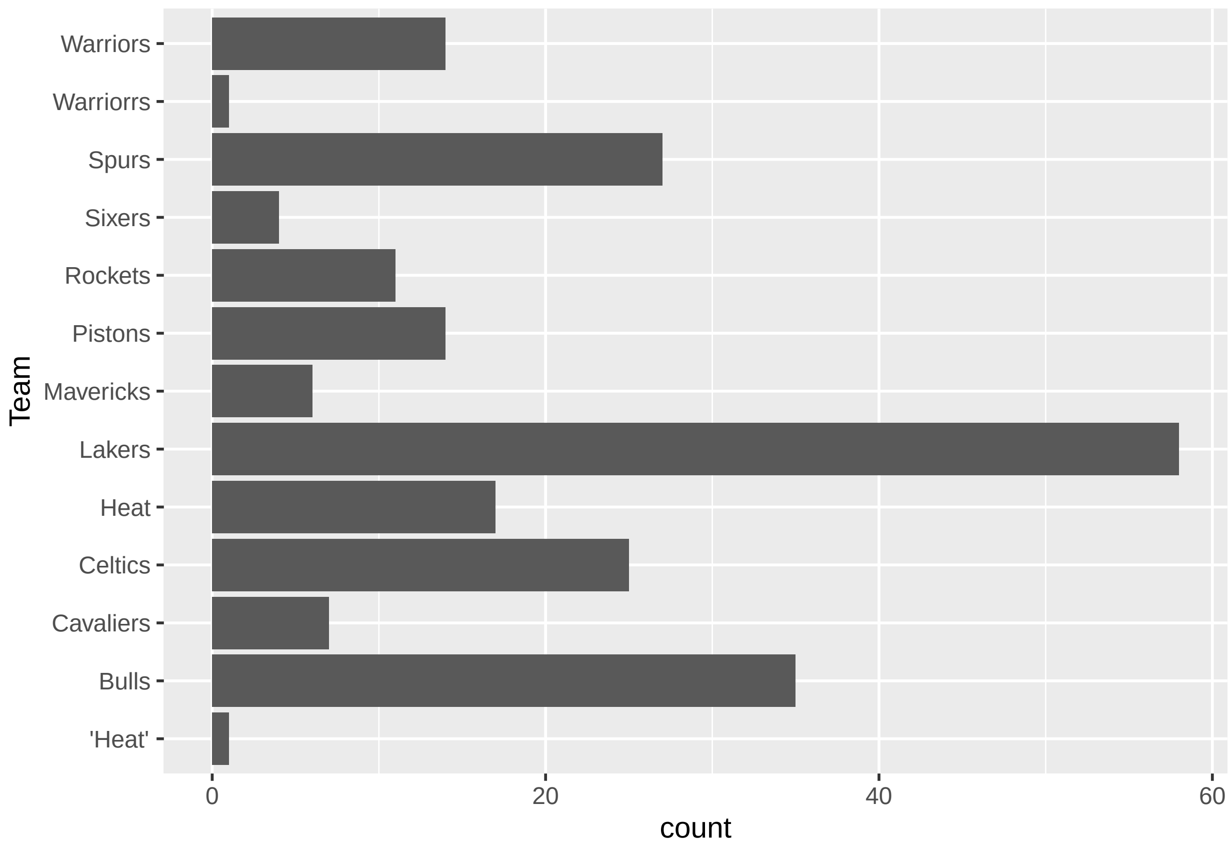
```
nba_championships_cleaned.df <-
  nba_championships.df %>%
  fill(TPP, .direction = "updown")


print(nba_championships_cleaned.df)
```

```
## # A tibble: 220 x 24
##     Year Team   Game   Win  Home    MP    FG   FGA   FGP    TP   TPA   TPP    FT
##    <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  1980 Lake~     1     1     1   240    48    89 0.539     0     0 0        13
##  2  1980 Lake~     2     0     1   240    48    95 0.505     0     1 0         8
##  3  1980 Lake~     3     1     0   240    44    92 0.478     0     1 0        23
##  4  1980 Lake~     4     0     0   240    44    93 0.473     0     0 0        14
##  5  1980 Lake~     5     1     1   240    41    91 0.451     0     0 0        26
##   6   1980 Lake~      6      1      0    240     45     92 0.489     0      2  0          33
##   7   1981 Celt~      1      1      1    240     41     95 0.432     0      1  0          16
##   8   1981 Celt~      2      0      1    240     41     82 0.5       0      3  0           8
##   9   1981 Celt~      3      1      0    240     40     89 0.449     2      3  0.667      12
##  10   1981 Celt~      4      0      0    240     35     74 0.473     0      3  0          16
## # i 210 more rows
## # i 11 more variables: FTA <dbl>, FTP <dbl>, ORB <dbl>, DRB <dbl>, TRB <dbl>,
## #   AST <dbl>, STL <dbl>, BLK <dbl>, TOV <dbl>, PF <dbl>, PTS <dbl>
```

Now that we have cleaner dataset after getting rid of the NA values, the next step in this data cleaning and manipulation stage is to find any duplications or errors.

If we create a simple visualization of how many times each NBA teams shows up in this dataset, we see there are two duplcates of the Warriors and Heat teams labeled as “Warriorrs and”‘Heat’“.

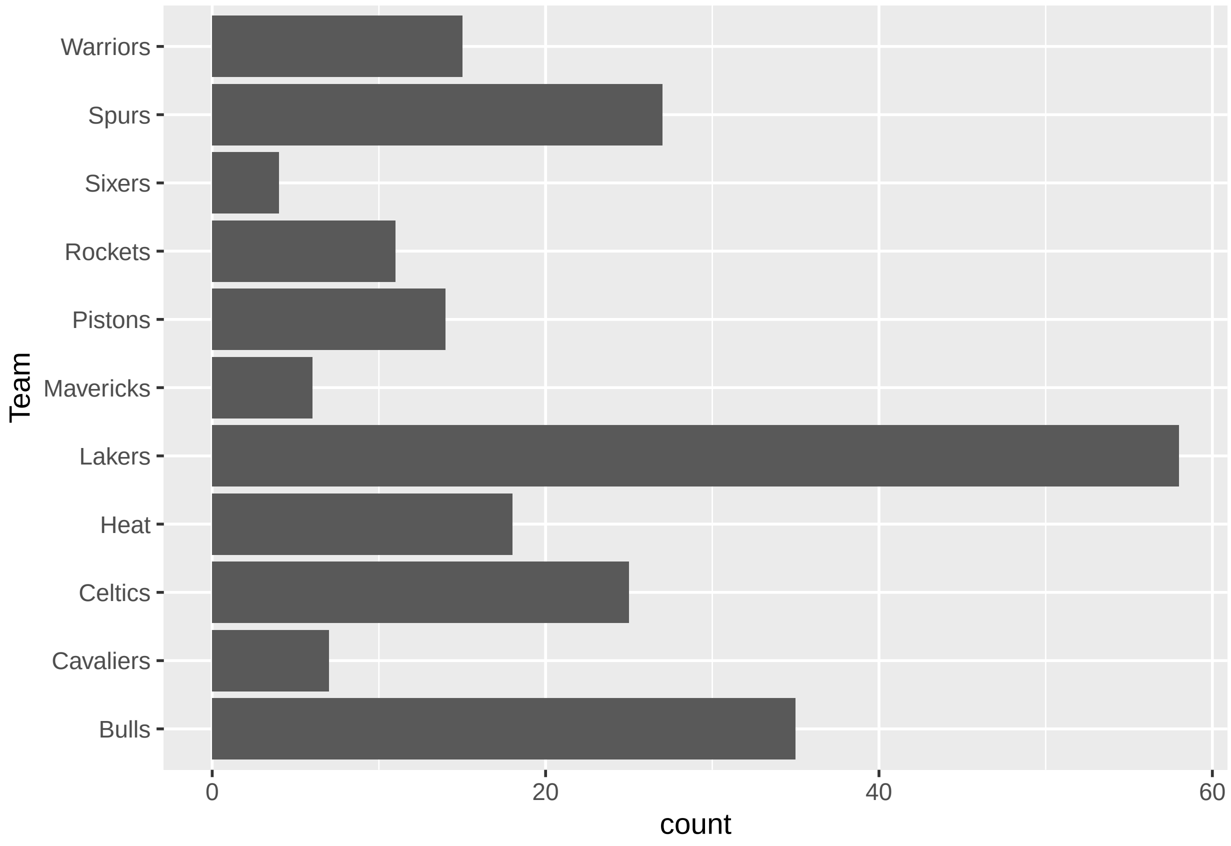
```
# Visulization of NBA teams
nba_championships.df %>%
  ggplot(aes(Team)) +
  geom_bar()+
  coord_flip()
```



An easy fix would be to use an “if-else” statement from the Team variable and create a condition that if Warriorrs or ‘Heat’ is found, then it is to be replaced with its correct NBA team (Warriors and Heat).

```
# Check the Team column and replace Warriorrs with Warriors and 'Heat' with Heat
nba_championships_cleaned.df <-
```

```
  nba_championships_cleaned.df %>%
  mutate(Team = if_else(Team == "Warriorrs", "Warriors", Team)) %>%
  mutate(Team = if_else(Team == "'Heat'", "Heat", Team))

# Create same visualization with updated team names
nba_championships_cleaned.df %>%
  ggplot(aes(Team)) +
  geom_bar()+
  coord_flip()
```



After fixing the duplicate team names, I spent a few hours digging and checking for any remaining common errors such as duplicate Team Points, or if two competing teams from a single game being declared both winners or losers.

I was able to pick up some common errors during my search in the Championship data-set.

These errors included: - 1984 Celtics, Game 1: "PTS" should be changed from 115 to 109 and "Win" should be set to 0 since they lost.

- 1984 Celtics, Game 2: "Win" should be set to 1 since they won.
- 1996 Bulls, Game 6: "Home" should be set to 1 since they were the home team.
- 2012 Heat Game 1 duplicates found. "Year" should be set to 2013.

```r
# 1984 Celtics Game 1
nba_championships_cleaned.df[23, 24] <- 109
nba_championships_cleaned.df[23,4] <- 0

# 1984 Celtics Game 2
nba_championships_cleaned.df[24,4] <- 1

# 1996 Bulls Game 6
nba_championships_cleaned.df[97,5] <- 1

# 2012 Heat Game 1
nba_championships_cleaned.df[187, 1] <- 2013
```

All NA values and incorrect pieces of information within the dataset have now been solved. A great idea would be to convert some variables represented by binary values into Categorical variables. This would make it easier to understand and communicate with the data.

The variables that can be changed to categorical variables could be Home and Win since they contain only 1's and 0's.

```r
# Converting Home and Win variables into factors (categorical variables)
nba_championships_cleaned.df <-
  nba_championships_cleaned.df %>%
  mutate(Home = as.factor(Home)) %>%
  mutate(Win = as.factor(Win))

nba_championships_cleaned.df %>%
  select(Win, Home)
```

```
## # A tibble: 220 x 2
##    Win   Home
##    <fct> <fct>
##  1 1     1
##  2 0     1
##  3 1     0
##  4 0     0
##  5 1     1
##  6 1     0
##  7 1     1
##  8 0     1
##  9 1     0
## 10 0     0
## # i 210 more rows
```

With the Home and Win variables being successfully converted into categorical variables, it's best to replace the 0's and 1's in each column with a label that's more readable.

For the Home column we can replace 0's with the label "Away Team" and 1's with "Home Team".

Then the Win column, 0's can be replaced with "Loss" and 1's with "Win".

From there, lets display a preview of how the columns look now. The presentation going from numeric values to actual character labels.

```
# Home Variable --> numeric (0 & 1) to character (Away Team & Home Team)
home_levels <- c("Away Team", "Home Team")
levels(nba_championships_cleaned.df$Home) <- home_levels

#Win Variable --> numeric (0 & 1) to character (Loss & Win)
win_levels <- c("Loss", "Win")
levels(nba_championships_cleaned.df$Win) <- win_levels


nba_championships_cleaned.df %>% head(5)
```

```
## # A tibble: 5 x 24
##    Year Team    Game Win   Home     MP    FG   FGA   FGP    TP   TPA   TPP    FT
##   <dbl> <chr>  <dbl> <fct> <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  1980 Lakers     1 Win   Home~   240    48    89 0.539     0     0     0    13
## 2  1980 Lakers     2 Loss  Home~   240    48    95 0.505     0     1     0     8
## 3  1980 Lakers     3 Win   Away~   240    44    92 0.478     0     1     0    23
## 4  1980 Lakers     4 Loss  Away~   240    44    93 0.473     0     0     0    14
## 5  1980 Lakers     5 Win   Home~   240    41    91 0.451     0     0     0    26
## # i 11 more variables: FTA <dbl>, FTP <dbl>, ORB <dbl>, DRB <dbl>, TRB <dbl>,
## #   AST <dbl>, STL <dbl>, BLK <dbl>, TOV <dbl>, PF <dbl>, PTS <dbl>
```

Alright, we've just finished the data cleaning and manipulation stage for our NBA Championship dataset! However we have one more dataset to go, our NBA Runner-Ups datatset. Since we saw earlier that both these datasets are similarly structured, we can carry out the same steps from the NBA Championships dataset.

Likewise with the NBA Championship data-set, let's first check the runner-ups dataset for any missing values.

```
nba_runnerUps.df %>% filter_all(any_vars(is.na(.)))
```

```
## # A tibble: 3 x 24
##    Year Team    Game   Win  Home    MP    FG   FGA   FGP    TP   TPA   TPP    FT
##   <dbl> <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  1980 Sixers     4     1     1   240    41    79 0.519     0     0    NA    23
## 2  1982 Sixers     5     1     1   240    56    94 0.596     0     0    NA    23
## 3  1984 Lakers     4     0     1   265    50    85 0.588     0     0    NA    25
## # i 11 more variables: FTA <dbl>, FTP <dbl>, ORB <dbl>, DRB <dbl>, TRB <dbl>,
## #   AST <dbl>, STL <dbl>, BLK <dbl>, TOV <dbl>, PF <dbl>, PTS <dbl>
```

After filtering all rows for any missing values, it appears that the same column Three Point Percentages (TPP) has missing values this time occurring for:

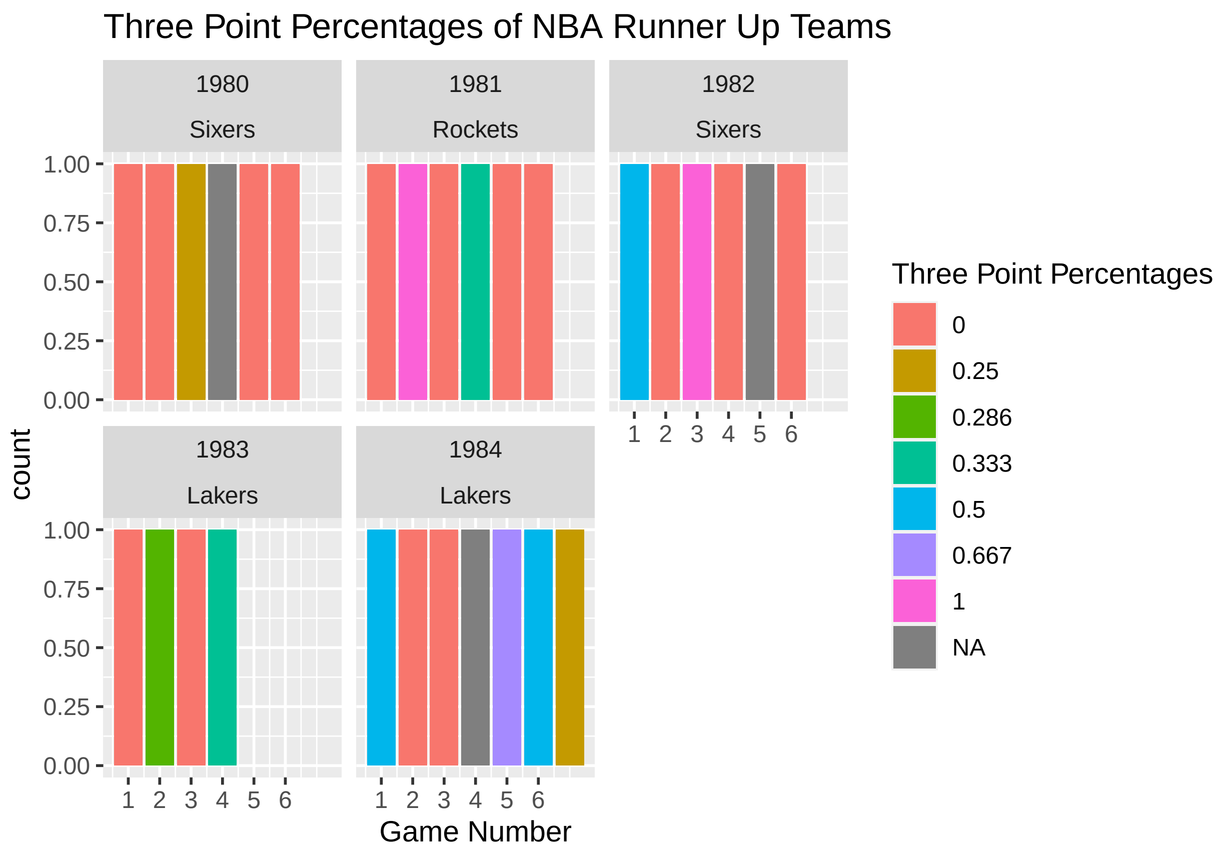
- Sixers 1980 Finals Game 4
- Sixers 1982 Finals Game 5
- Sixers 1984 Finals Game 4

To better illustrate this lets create a visualization.

```r
nba_runnerUps.df %>%
  filter(between(Year, 1980, 1984))
```

```
## # A tibble: 29 x 24
##     Year Team     Game   Win  Home    MP    FG   FGA   FGP    TP   TPA    TPP
##    <dbl> <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
##   1  1980 Sixers      1     0     0   240    40    90 0.444     0     2  0
##   2  1980 Sixers      2     1     0   240    43    85 0.506     0     1  0
##   3  1980 Sixers      3     0     1   240    45    93 0.484     1     4  0.25
##   4  1980 Sixers      4     1     1   240    41    79 0.519     0     0 NA
##   5  1980 Sixers      5     0     0   240    42    94 0.447     0     3  0
##   6  1980 Sixers      6     0     1   240    47    89 0.528     0     6  0
##   7  1981 Rockets     1     0     0   240    42    99 0.424     0     2  0
##   8  1981 Rockets     2     1     0   240    34    85 0.4       2     2  1
##   9  1981 Rockets     3     0     1   240    24    79 0.304     0     1  0
## 10  1981 Rockets     4     1     1   240    37   103 0.359     1     3  0.333
## # i 19 more rows
## # i 12 more variables: FT <dbl>, FTA <dbl>, FTP <dbl>, ORB <dbl>, DRB <dbl>,
## #   TRB <dbl>, AST <dbl>, STL <dbl>, BLK <dbl>, TOV <dbl>, PF <dbl>, PTS <dbl>
```

```r
nba_runnerUps.df %>%
  filter(between(Year, 1980, 1984)) %>%
  ggplot(aes(Game, fill = factor(TPP)))+
  geom_bar(position = "fill")+
  facet_wrap(Year~Team) +
  scale_x_continuous(breaks = c(1,2,3,4,5,6))+
  labs(
    x = "Game Number",
    fill = "Three Point Percentages",
    title = "Three Point Percentages of NBA Runner Up Teams"
  )
```



Now that we know where the NA values are located from the NBA runner ups data set, we can begin to replace each of those values with a 0 since each game likewise with the NBA Championships data set has Three Pointers Attempt of 0 (TPA).

However, unlike last time, we can't use the fill(TPP, .direction) code line since each NA value either has an actual three point percentage number above or below the NA value. So applying each NA value with a 0 in the same direction won't work.

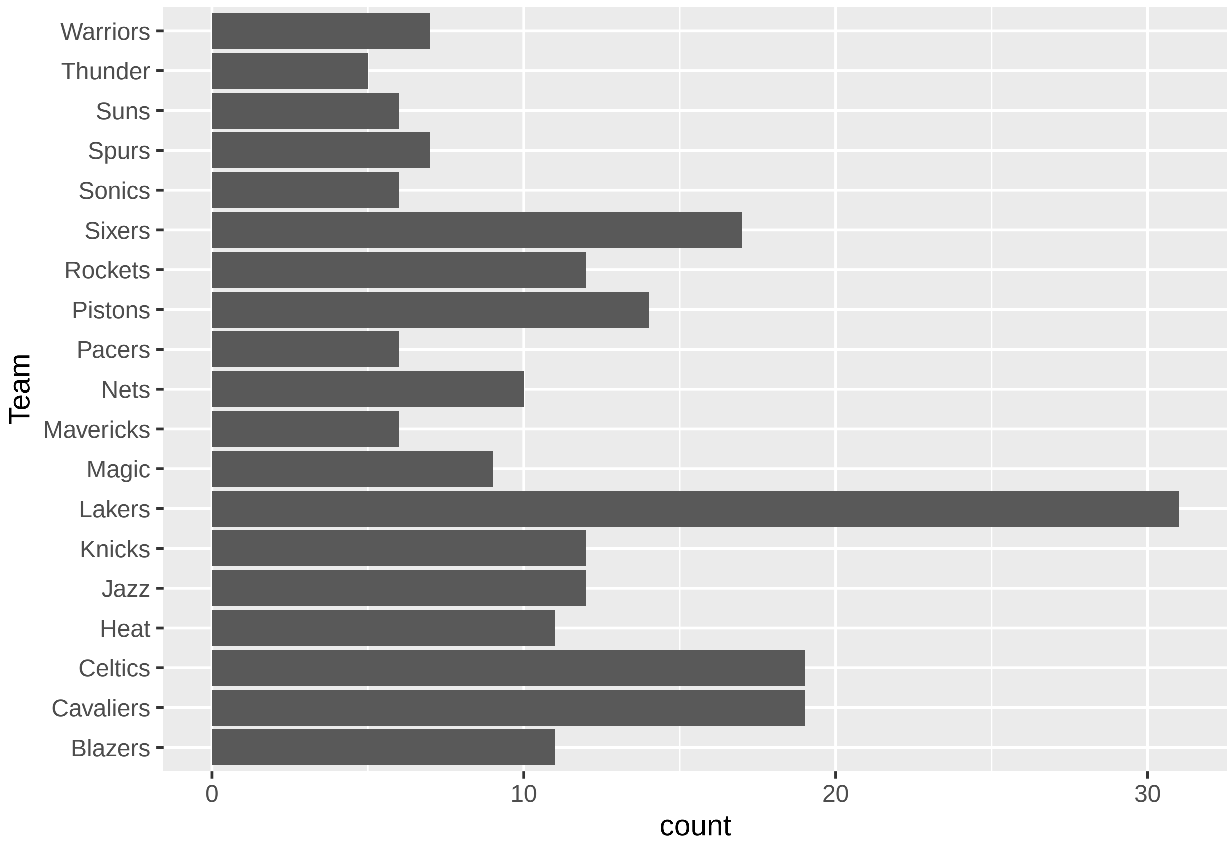
Instead, we can physically replace each NA value with 0 ourselves, since we know where the NA values are located.

```
nba_runnerUps_cleaned.df <- nba_runnerUps.df %>% replace(is.na(.), 0)
print(nba_runnerUps_cleaned.df)
```

```
## # A tibble: 220 x 24
##      Year Team   Game   Win  Home    MP    FG   FGA   FGP    TP   TPA   TPP    FT
##     <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  1980 Sixe~     1     0     0   240    40    90 0.444     0     2 0        22
##  2  1980 Sixe~     2     1     0   240    43    85 0.506     0     1 0        21
##  3  1980 Sixe~     3     0     1   240    45    93 0.484     1     4 0.25     10
##  4  1980 Sixe~     4     1     1   240    41    79 0.519     0     0 0        23
##  5  1980 Sixe~     5     0     0   240    42    94 0.447     0     3 0        19
##  6  1980 Sixe~     6     0     1   240    47    89 0.528     0     6 0        13
##  7  1981 Rock~     1     0     0   240    42    99 0.424     0     2 0        11
##  8  1981 Rock~     2     1     0   240    34    85 0.4       2     2 1        22
##  9  1981 Rock~     3     0     1   240    24    79 0.304     0     1 0        23
## 10  1981 Rock~     4     1     1   240    37   103 0.359     1     3 0.333     16
## # i 210 more rows
## # i 11 more variables: FTA <dbl>, FTP <dbl>, ORB <dbl>, DRB <dbl>, TRB <dbl>,
## #   AST <dbl>, STL <dbl>, BLK <dbl>, TOV <dbl>, PF <dbl>, PTS <dbl>
```

Lets run a simple visualization of every NBA team to make sure that each team is spelled correctly and there are no duplicates.

```
nba_runnerUps.df %>%
  ggplot(aes(Team)) +
  geom_bar()+
  coord_flip()
```



Furthermore, I'll spend some time looking around for remaining common errors like how I found from the NBA Championship dataset.

Errors I found: - 1982 Sixers, Game 2: "Home" should be set to 1 instead of 0 since they're the home team.

- 1984 Lakers, Game 1: "Win" should be set to 1 since they won and "Home" should be set to 0 since they're the away team.
- 1987 Celtics, Game 3: "MP" should be set to 240 instead of 40.
- 1998 Jazz, Game 5: "Home" should be set to 0 since they're the away team.

```
# 1982 Sixers Game 2
nba_runnerUps_cleaned.df[14,5] <- 1
```

```
# 1984 Lakers Game 1
nba_runnerUps_cleaned.df[23,4] <- 1
nba_runnerUps_cleaned.df[23,5] <- 0

# 1987 Celtics Game 3
nba_runnerUps_cleaned.df[44,6] <- 240

# 1998 Jazz Game 5
nba_runnerUps_cleaned.df[108,5] <- 0
```

Once all missing, duplicate, and incorrect data have been fixed, it's time to convert the same variables (Home and Win) into categorical variables.

```
# Converting Home and Win variables into factors (categorical variables)
nba_runnerUps_cleaned.df <-
  nba_runnerUps_cleaned.df %>%
  mutate(Home = as.factor(Home)) %>%
  mutate(Win = as.factor(Win))

nba_runnerUps_cleaned.df %>%
  select(Win, Home)
```

```
## # A tibble: 220 x 2
##    Win   Home
##    <fct> <fct>
##  1 0     0
##  2 1     0
##  3 0     1
##  4 1     1
##  5 0     0
##  6 0     1
##  7 0     0
##  8 1     0
##  9 0     1
## 10 1     1
## # i 210 more rows
```

Now these variables have been converted into categorical variables, we can apply the same labels per variables.

Home: 0's will be replaced by "Away Team" and 1's with "Home Team".

Win: 0's will be replaced by "Loss" and 1's with "Win".

```
# Home Variable --> numeric (0 & 1) to character (Away Team & Home Team)
runnerUps_home_levels <- c("Away Team", "Home Team")
levels(nba_runnerUps_cleaned.df$Home) <- runnerUps_home_levels

#Win Variable --> numeric (0 & 1) to character (Loss & Win)
runnerUps_win_levels <- c("Loss", "Win")
levels(nba_runnerUps_cleaned.df$Win) <- runnerUps_win_levels


nba_runnerUps_cleaned.df %>% head(5)
```

```
## # A tibble: 5 x 24
##    Year Team    Game Win   Home     MP    FG   FGA   FGP    TP   TPA   TPP    FT
##   <dbl> <chr>  <dbl> <fct> <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  1980 Sixers     1 Loss  Away~   240    40    90 0.444     0     2  0       22
## 2  1980 Sixers     2 Win   Away~   240    43    85 0.506     0     1  0       21
## 3  1980 Sixers     3 Loss  Home~   240    45    93 0.484     1     4  0.25    10
## 4  1980 Sixers     4 Win   Home~   240    41    79 0.519     0     0  0       23
## 5  1980 Sixers     5 Loss  Away~   240    42    94 0.447     0     3  0       19
## # i 11 more variables: FTA <dbl>, FTP <dbl>, ORB <dbl>, DRB <dbl>, TRB <dbl>,
## #   AST <dbl>, STL <dbl>, BLK <dbl>, TOV <dbl>, PF <dbl>, PTS <dbl>
```

We have now completed the Data Cleaning and Manipulation Stage!

Up till now our data has been: * Processed: datasets are read-in and saved into our global environment * Cleaned: fixed any missing, duplicate, and incorrect information from the datasets * Manipulated: changed variables into categorical variables and replaced information in the columns with something more readable.

We can now start to begin our Analysis stage, where we dive into our initial questions stated at the beginning of the project and unravel meaningful insights.

Analyze

(Exploratory Data Analysis & Visualization)

Finally, now that everything has been prepared and cleaned, we can dive into our Analysis process and explore the questions that we asked ourselves.

Our first question in our Analysis stage is:

In which statistical categories did NBA Championship teams consistently outperform the runner-ups in NBA Finals game?

Before we tackle this question, let's first list the different stats we can look at: * Field Goals * Threes * Free Throws * Assists * Team's total points * Rebounds * Steals * Blocks * Turnovers * Personal Fouls

Since there's a lot of info on these stats, we can split them into two groups: **Offense** and **Defense**.

Offensive Statistical Categories: Field Goals, Threes, Free Throws, Assists, Team's total points. **Defensive Statistical Categories**: Rebounds, Steals, Blocks, Turnovers, Personal fouls.

To make things simpler for our analysis, let's combine the stats from both datasets into one. I suggest adding a new column called "Outcome" to tell us if a team won the NBA Championship or was the Runner-Up. This way, we won't have to keep going back and forth between the datasets, making the whole analysis easier for all of us.

```r
nba_championships_cleaned.df$Outcome <- "Championships"
nba_runnerUps_cleaned.df$Outcome <- "Runner-Ups"

combined_data <- rbind(nba_championships_cleaned.df, nba_runnerUps_cleaned.df)
```

Offensive Let's kick off by looking at the offensive side of basketball.

For Field Goals, Threes, and Free Throws, we have subcategories like makes, attempts, and percentage. We can group them by outcome (Championships and Runner-Ups) and quickly calculate the average for each of these subcategories.

```
combined_data %>%
  group_by(Outcome) %>%
  summarise("Total Field Goals" = mean(FG), "Field Goal Attempts" = mean(FGA), "Field Goal Percentage" =
```

```
## # A tibble: 2 x 4
##   Outcome       ‘Total Field Goals‘ ‘Field Goal Attempts‘ Field Goal Percentag~1
##   <chr>                       <dbl>                 <dbl>                  <dbl>
## 1 Championships                37.8                  80.9                   46.7
## 2 Runner-Ups                   36.3                  81.8                   44.5
## # i abbreviated name: 1: ‘Field Goal Percentage‘
```

```
combined_data %>%
  group_by(Outcome) %>%
  summarise("Total Three Pointers" = mean(TP), "Three Point Attempts" = mean(TPA), "Three Point Percenta
```

```
## # A tibble: 2 x 4
##   Outcome   ‘Total Three Pointers‘ ‘Three Point Attempts‘ Three Point Percenta~1
##   <chr>                      <dbl>                  <dbl>                  <dbl>
## 1 Champion~                   5.35                   14.6                   33.3
## 2 Runner-U~                   4.75                   14.6                   29.3
## # i abbreviated name: 1: ‘Three Point Percentage‘
```

```
combined_data %>%
  group_by(Outcome) %>%
  summarise("Total Free Throws" = mean(FT), "Free Throw Attempts" = mean(FTA), "Free Throw Percentate" =
```

```
## # A tibble: 2 x 4
##   Outcome       ‘Total Free Throws‘ ‘Free Throw Attempts‘ Free Throw Percentat~1
##   <chr>                       <dbl>                 <dbl>                  <dbl>
## 1 Championships                19.9                  27.1                   73.6
## 2 Runner-Ups                   19.0                  25.4                   74.8
## # i abbreviated name: 1: ‘Free Throw Percentate‘
```

After the calculations, let’s check how Championships and Runner-Ups stack up.

Field Goals:

Championships made roughly 37.35, Runner-Ups made 36.35. Runner-Ups attempted more shots (81.791) than Championships (80.877). Championships had a higher shooting percentage (46.652%) than Runner-Ups (44.460%). **Outcome:** Championships outperformed in all three Field Goal subcategories.

Threes:

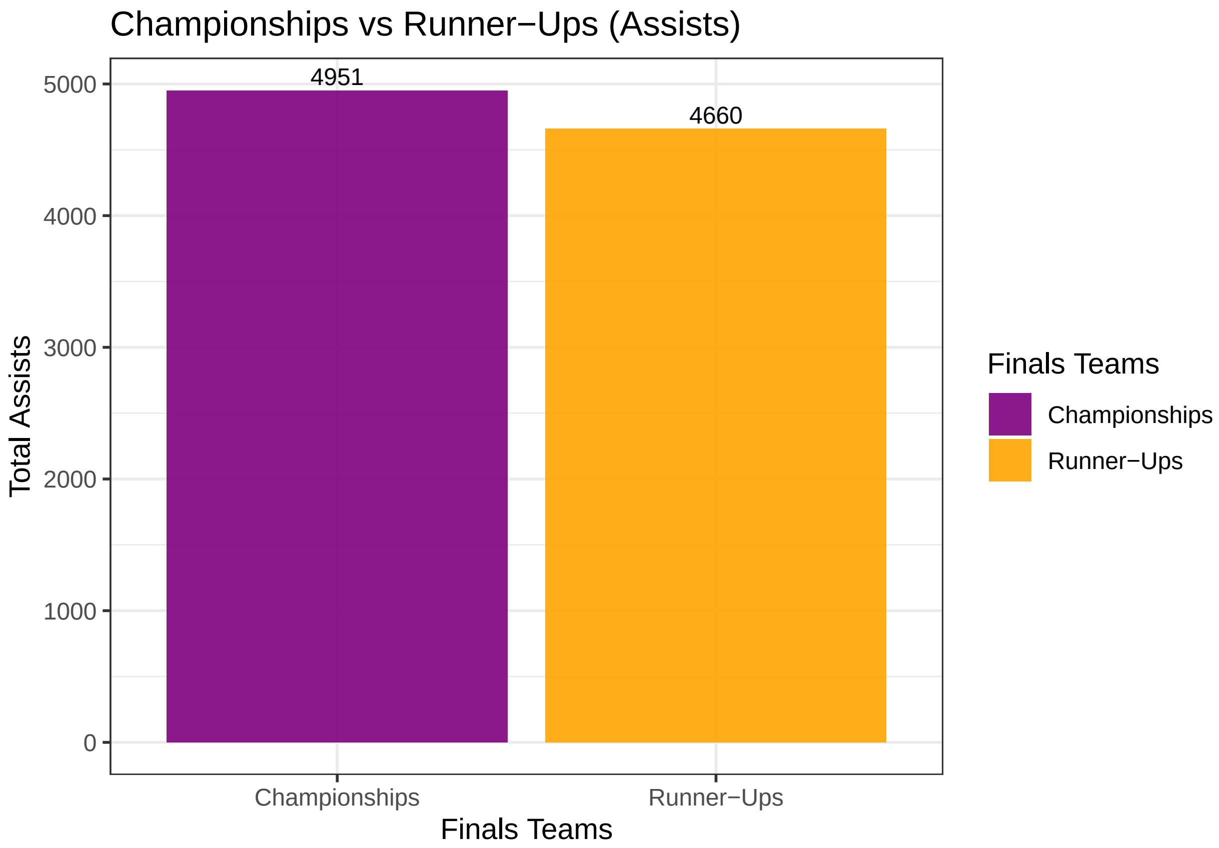
Championships made 5.35, Runner-Ups made 4.75. Championships attempted 14.60 threes, slightly more than Runner-Ups’ 14.56. Championships had a higher three-point shooting percentage (33.28%) compared to Runner-Ups (29.28%). **Outcome:** Championships outperformed in all three Threes subcategories.

Free Throws:

Championships made 19.93, Runner-Ups made 18.96. Runner-Ups attempted fewer free throws (25.35) compared to Championships (27.13). Runner-Ups had a higher free throw percentage (74.77%) than Championships (73.55%). **Outcome:** Championships outperformed in two out of three Free Throw subcategories.

Overall, across Field Goals, Threes, and Free Throws, Championships dominated in all three categories, showing better offensive performance in 3 out of the 5 categories compared to Runner-Ups. Next up, let’s visualize the assist comparison between Championships and Runner-Ups.

```
combined_data %>%
  group_by(Outcome) %>%
  summarise(total_Assists = sum(AST)) %>%
  ggplot(aes(Outcome, total_Assists, fill = Outcome)) +
  geom_bar(stat = "identity", alpha = 0.9) +
  geom_text(aes(label = total_Assists), vjust = -.3, size = 3) +
  theme_bw()+
  scale_fill_manual(values = c("#800080", "#FFA500")) +
  labs(
    fill = "Finals Teams",
    y = "Total Assists",
    x = "Finals Teams",
    title = "Championships vs Runner-Ups (Assists)"
  )
```



After visualizing the data, Championships accumulated 4951 assists, surpassing Runner-Ups with 4660 assists. This suggests that Championships possibly emphasized team-based play by moving the ball effectively to create better scoring opportunities.

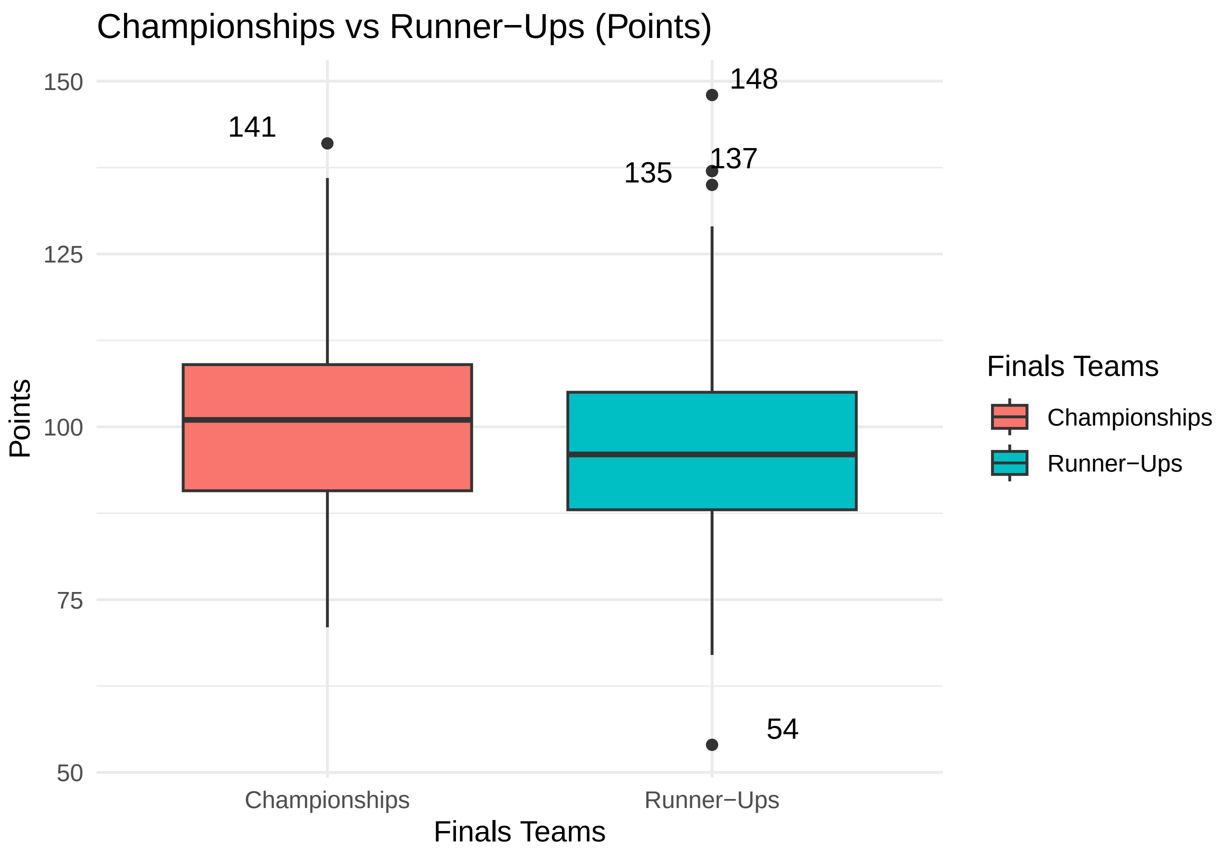
In the assist category, the **Championships did better**.

Moving on to our final offensive category, Team's Total Points, let's use a boxplot for visualization. This will help identify any outliers in both high and low-scoring games.

```
combined_data %>%
  group_by(Outcome) %>%
  summarise(
    Q1 = (quantile(PTS, .25)),
    Median = median(PTS),
    Q3 = quantile(PTS, .75),
    IQR = IQR(PTS),
    Max = max(PTS),
    Min = min(PTS)
  )
```

```
## # A tibble: 2 x 7
##   Outcome          Q1 Median    Q3   IQR   Max   Min
##   <chr>         <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Championships  90.8    101   109  18.2   141    71
## 2 Runner-Ups     88       96   105  17     148    54
```

```
find_Outlier <- function(x) {
  return(x < quantile(x, .25) - 1.5*IQR(x) | x > quantile(x, .75) + 1.5*IQR(x))
}

set.seed(14)
combined_data %>%
  group_by(Outcome) %>%
  mutate(outlier = ifelse(find_Outlier(PTS), PTS, NA)) %>%
  ggplot(aes(Outcome, PTS, fill = Outcome)) +
  geom_boxplot()+
  geom_text(check_overlap = TRUE, aes(label = outlier), na.rm = TRUE, position = position_jitter(), vju
  theme_minimal() +
  labs(
    fill = "Finals Teams",
    y = "Points",
    x = "Finals Teams",
    title = "Championships vs Runner-Ups (Points)"
  )
```



Upon creating this visualization, it's clear that Championships had more games with higher team points. Their median surpasses the 100-point mark, while Runner-Ups fall below it. Runner-Ups show a broader range, with the lowest scoring game at 54 and the highest at 148. Additionally, Runner-Ups exhibit more outliers on both ends. In contrast, Championships appear more consistent in their game scores.

Therefore, Championships outperformed in the Team's Total Points category.

This concludes our assessment of all offensive categories, with Championships scoring 5/5 compared to Runner-Ups' 0/5. It's evident that Championships were superior offensively.

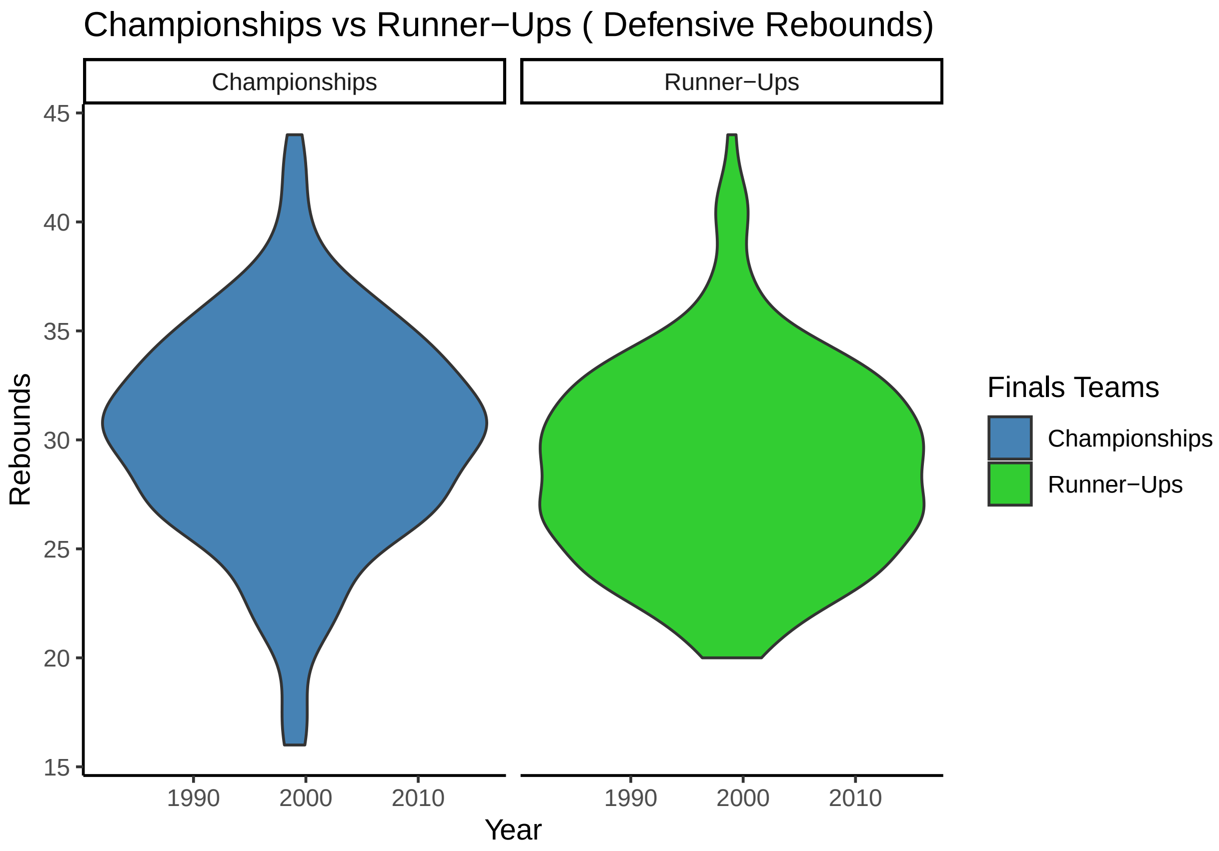
Now, let's explore their defensive capabilities!

Defense Starting with rebounds, specifically defensive rebounds. A visualization will better illustrate the difference between Runner-Ups and Championships.

```
#Rebounds

combined_data %>%
  group_by(Outcome) %>%
  summarise("Total Defensive Rebounds" = sum(DRB))
```

```
## # A tibble: 2 x 2
##   Outcome       `Total Defensive Rebounds`
##   <chr>                              <dbl>
## 1 Championships                       6644
## 2 Runner-Ups                          6327
```

```
combined_data %>%
  ggplot(aes(x = Year, y = DRB, fill = Outcome)) +
  geom_violin()+
  facet_wrap(~Outcome) +
  theme_classic()+
  scale_fill_manual(values = c("steelblue", "limegreen")) +
  labs(
    fill = "Finals Teams",
    y = "Rebounds",
    title = "Championships vs Runner-Ups ( Defensive Rebounds)"
  )
```



Before we jump into the violin plot for defensive rebounds, note that the Championship team got 6644 defensive rebounds, beating the Runner-Ups' 6327. It's a good sign for the Championships, but we need more details on consistency.

A violin plot will help show defensive rebounds for both teams over the years.

Starting with the Championships, in 2000, they had 15 to 20 defensive rebounds at the lower end. From 25 to 35, there's consistent rebounding spread from 1980 to 2018. After 35, high rebounds drop, mostly around 2000.

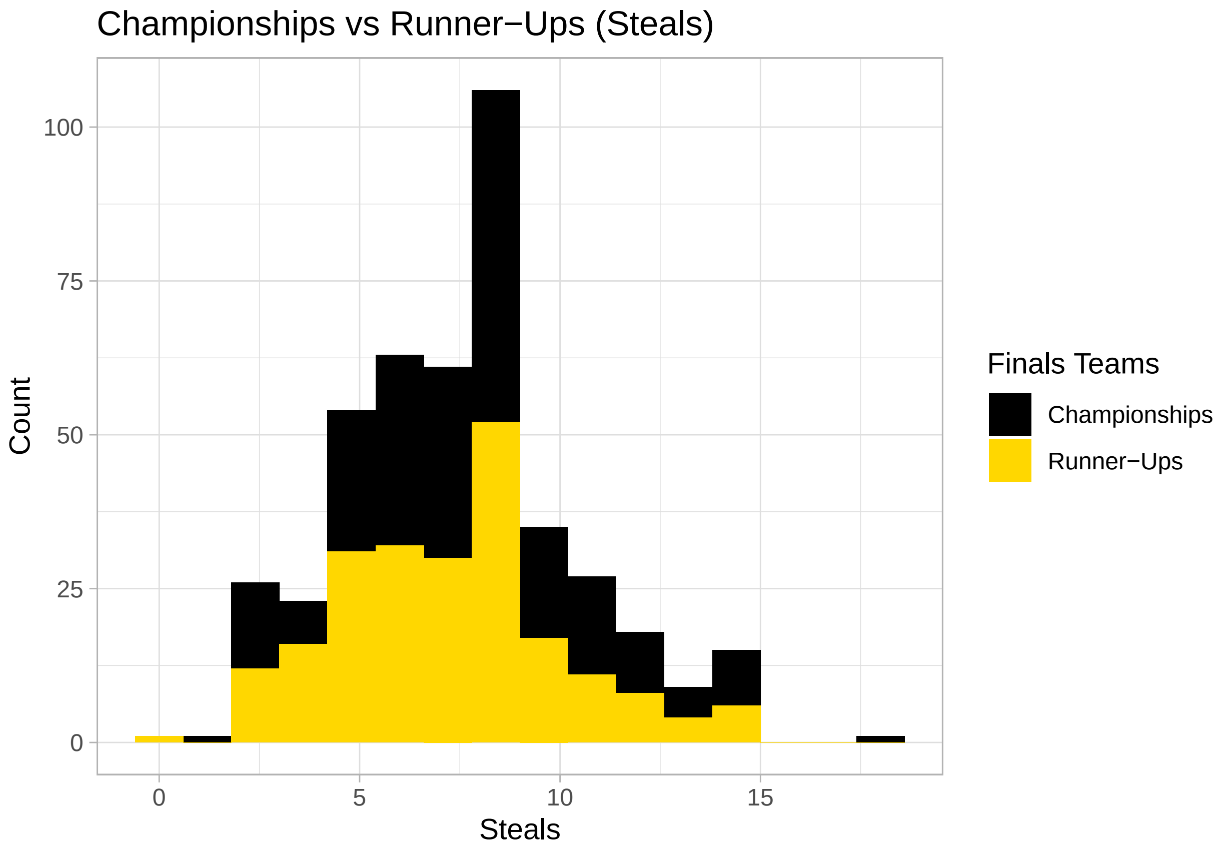
Now, the Runner-Ups: their rebounds start at 20, meaning few low rebound games. After 20, a wide range of consistent high rebounds up to 35. After 35, occasional very high rebounds, like the Championships.

Comparing, Championships show more consistency in low and high defensive rebounds, especially from 30 to 40. Runner-Ups are consistent in the 25-30 range.

Championships did better in defensive rebounds compared to Runner-Ups.

Moving to steals, let's make a histogram to show the spread and frequency of steals.

```
combined_data %>%
  ggplot(aes(STL, fill = Outcome))+
  geom_histogram(binwidth = 1.2)+
  theme_light()+
  scale_fill_manual(values = c("black", "gold")) +
  labs(
    fill = "Finals Teams",
    x = "Steals",
    y = "Count",
    title = "Championships vs Runner-Ups (Steals)"
  )
```



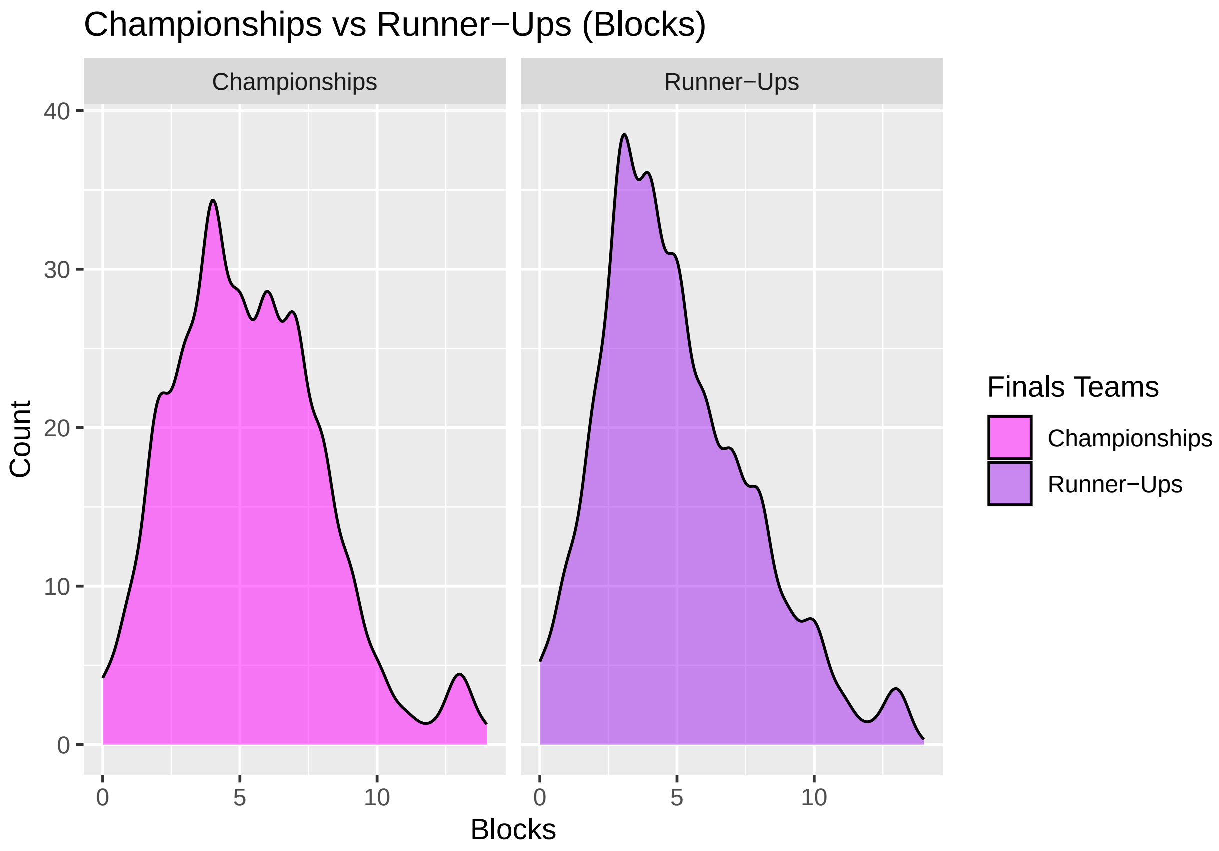
After making the histogram, where Championships are in black and Runner-Ups in yellow, we see some clear peaks in the Championship's distribution. Around 7-8 steals, Championships have over 100 occurrences, while Runner-Ups barely cross 50.

Looking at the chart, it's clear that **Championships performed better** in steals.

Now, let's move on to blocks. For this, we'll use a density plot.

```
#Blocks

combined_data %>%
  ggplot(aes(BLK, fill = Outcome))+
  geom_density(alpha = 0.5, aes(y = ..count..), adjust = 0.55)+
  facet_wrap(~Outcome)+
  theme_gray()+
  scale_fill_manual(values = c("magenta", "purple"))+
  labs(
    fill = "Finals Teams",
    x = "Blocks",
    y = "Count",
    title = "Championships vs Runner-Ups (Blocks)"
  )
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



With the density plot, it seems that Runner-Ups have higher frequencies in block numbers compared to Championships, who maintain a more consistent number of blocks throughout the chart. Additionally, Championships experience a significant decline in the number of recorded large blocks after their peak around the 7.5 blocks mark. In contrast, Runner-Ups occasionally have peaks in the large block region after the 7.5 margin.

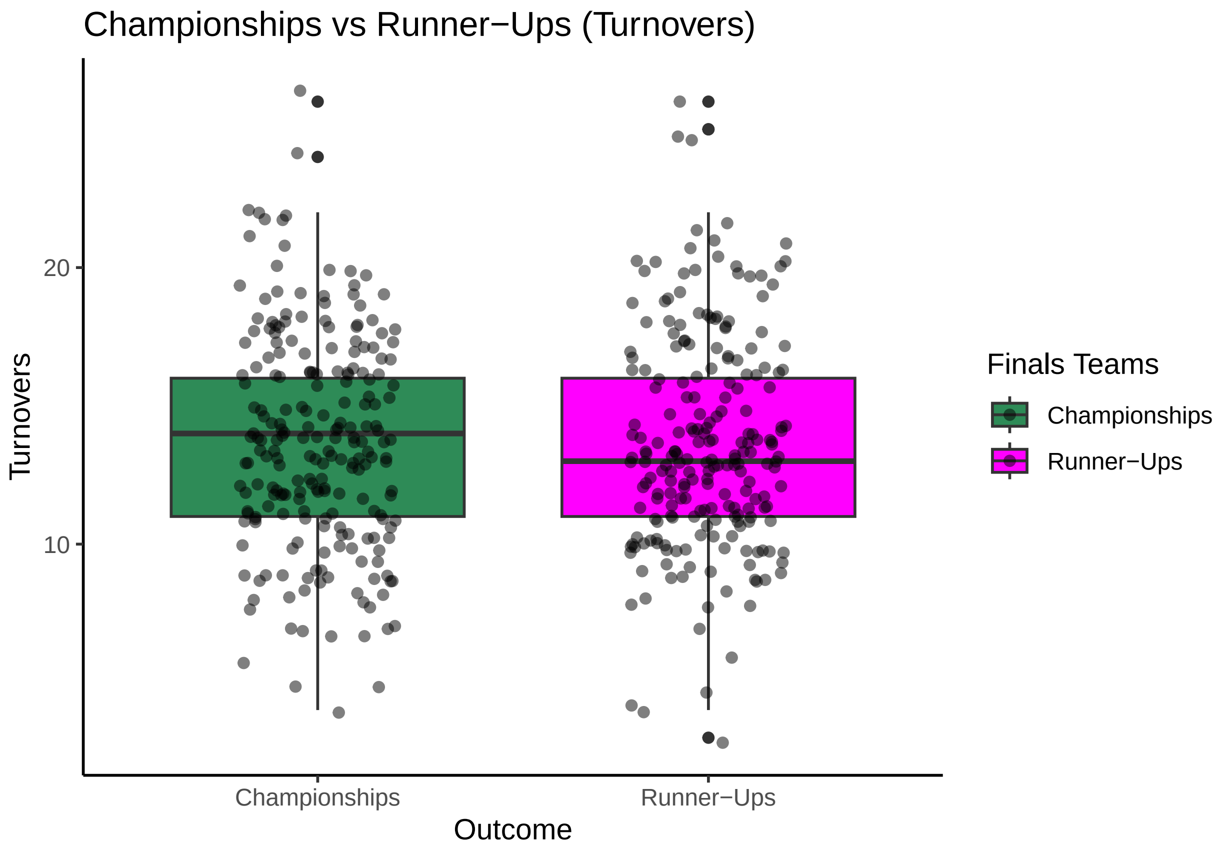
Considering this analysis, it appears **Runner-Ups performed better** in the blocks category.

Moving on to turnovers, we'll perform a quick calculation to see the difference between Championships and Runner-Ups. Additionally, we'll use another box plot chart, this time displaying all values with low transparency filtered by Game numbers *(Game 1, Game 2, etc.)*.

```
#Turnover
combined_data %>%
  group_by(Outcome) %>%
  summarise("Number of Turnovers" = sum(TOV))
```

```
## # A tibble: 2 x 2
##   Outcome       `Number of Turnovers`
##   <chr>                         <dbl>
## 1 Championships                  3016
## 2 Runner-Ups                     3001
```

```
combined_data %>%
  ggplot(aes(x = Outcome, y = TOV, fill = Outcome)) +
  geom_boxplot() +
  geom_jitter(position = position_jitter(width = 0.2), alpha = 0.5) +
  theme_classic() +
  scale_fill_manual(values = c("seagreen", "magenta")) +
  labs(
    fill = "Finals Teams",
    x = "Outcome",
    y = "Turnovers",
    title = "Championships vs Runner-Ups (Turnovers)"
  )
```



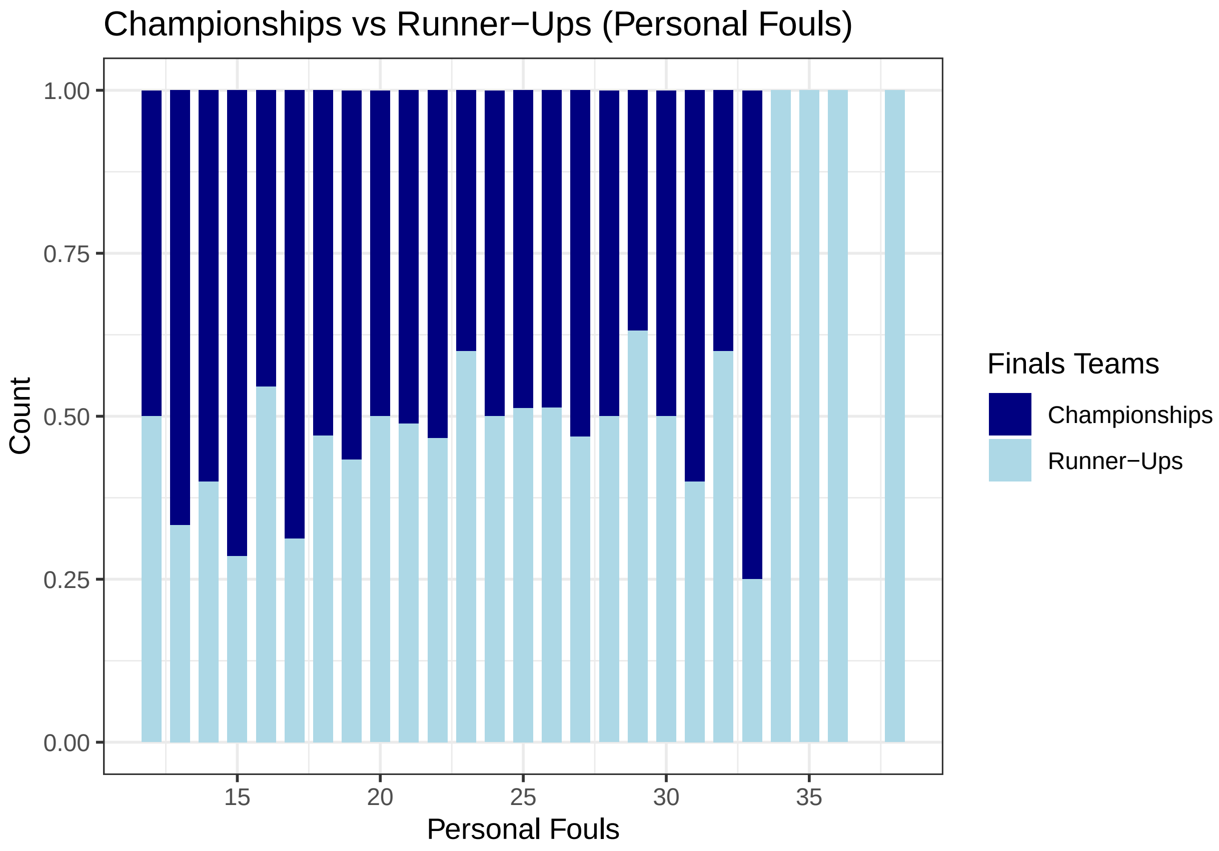
After analyzing both the calculation and visualization, it's clear that Championships have slightly more turnovers than Runner-Ups, with 3016 compared to 3001. Runner-Ups appear to prioritize defense.

In the visualization, Runner-Ups have more low-scored turnovers, indicating a defensive focus. Both sides have similar numbers for high-scored turnovers.

Overall,**Runner-Ups performed better in the turnovers category**, showing fewer recorded turnovers and a defensive mindset.

Now, let's examine personal fouls using a bar chart to compare proportions for both Championships and Runner-Ups.

```
#Personal Fouls

combined_data %>%
ggplot(aes(PF, fill = Outcome)) +
  geom_bar(width = 0.7, position = "fill") +
  scale_fill_manual(values = c("navy", "lightblue"))+
  scale_x_continuous(breaks = seq(from = 15, to = 35, by = 5))+
  theme_bw()+
  labs(
    fill = "Finals Teams",
    x = "Personal Fouls",
    y = "Count",
    title = "Championships vs Runner-Ups (Personal Fouls)"
  )
```



In the bar chart, the last four bars on the right side are all filled by Runner-Ups, showing they had high personal foul games. While some calls might have been mistakes, Championships generally did better in avoiding fouls. Most of the lower foul numbers are from Championships on the left side, with more high fouls from Runner-Ups on the right.

Out of the total 26 bars, 10 are under 50% for Runner-Ups, meaning those 10 are over 50% for Championships. Conversely, 4 bars are over 50% for Runner-Ups, showing those 4 are under 50% for Championships. The rest of the bars show similar proportions for both. Runner-Ups did well in avoiding turnovers and playing defense, but Championships showed they were better at avoiding turnovers.

In summary, Championships did better in avoiding turnovers, showing they focused more on defense. Tallying up the scores, Championships scored 5/5 on offense and 3/5 on defense, while Runner-Ups scored 0/5 and 2/5 respectively. **Championship teams performed better!**

Additionally, the strong defense shown by both Championships and Runner-Ups suggests that defense is key in NBA Playoffs and Finals runs. This shows how important defense is in achieving success in the playoffs.

Now lets move onto the second question in our analysis:

Which team had the most success, what team came up short the most. How do they compare?

To determine the team with the most success among Championship teams, we'll analyze which team has accumulated the most wins and championships.

```r
combined_data %>%
  filter(Outcome == "Championships", Win == "Win") %>%
  group_by(Team) %>%
  summarise(Wins = n()) %>%
  arrange(desc(Wins)) %>%
  mutate("Num. of Championships" = Wins/4)
```

```
## # A tibble: 11 x 3
##    Team       Wins ‘Num. of Championships‘
##    <chr>     <int>                   <dbl>
##  1 Lakers       40                      10
##  2 Bulls        24                       6
##  3 Spurs        20                       5
##  4 Celtics      16                       4
##  5 Heat         12                       3
##  6 Pistons      12                       3
##  7 Warriors     12                       3
##  8 Rockets       8                       2
##  9 Cavaliers     4                       1
## 10 Mavericks     4                       1
## 11 Sixers        4                       1
```

After filtering the Championship teams, we find that the Lakers have the most wins with 40 wins and 10 championships. We determine the number of championships by dividing each team's wins by 4. Additionally, the Cavaliers, Mavericks, and Sixers each have 4 wins, indicating they have one championship each.

Now, let's determine the team that has come up short the most among the Runner-Up teams. This means identifying the team with the most losses and missed championship opportunities. We'll analyze the Runner-Up teams by their losses and number of missed championships.

```r
combined_data %>%
  filter(Outcome == "Runner-Ups", Win == "Loss") %>%
  group_by(Team) %>%
  summarise(Loss = n()) %>%
  arrange(desc(Loss)) %>%
  mutate("Num. of Missed Championships" = Loss/4)
```

```
## # A tibble: 19 x 3
##    Team       Loss ‘Num. of Missed Championships‘
##    <chr>     <int>                          <dbl>
##  1 Lakers       24                              6
##  2 Cavaliers    16                              4
##  3 Celtics      12                              3
##  4 Sixers       12                              3
##  5 Blazers       8                              2
##  6 Heat          8                              2
##  7 Jazz          8                              2
##  8 Knicks        8                              2
##  9 Magic         8                              2
## 10 Nets          8                              2
## 11 Pistons       8                              2
## 12 Rockets       8                              2
## 13 Mavericks     4                              1
## 14 Pacers        4                              1
## 15 Sonics        4                              1
## 16 Spurs         4                              1
## 17 Suns          4                              1
## 18 Thunder       4                              1
## 19 Warriors      4                              1
```

We can see that the team that has came up short the most is the Lakers. Despite their 40 wins and 10 championships, they have also faced the most losses among Runner-Up teams, totaling 24 losses and missing out on 6 potential NBA championships. This highlights the Lakers’ consistent presence as strong contenders throughout the years.

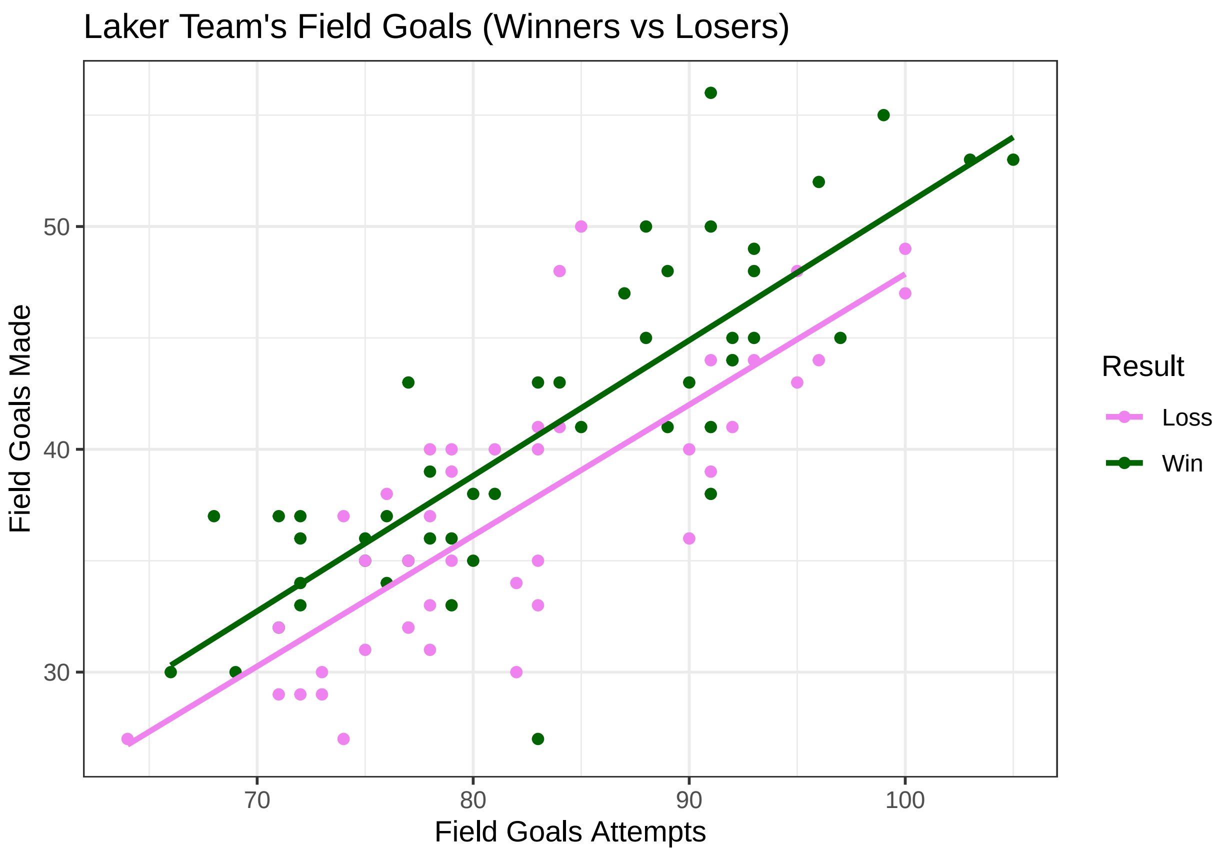
Now we know that the Lakers team has been on both opposite of the win/loss spectrum, it would be interesting to compare each of these Lakers versions and see how great they differed from one another.

Our first comparison will be their Field Goals Attempts vs. Field Goals Made. This comparison can help us understand if attempting more shots led to making more of them. We’ll also identify areas of efficiency (more field goals made with fewer attempts) and inefficiency (fewer field goals with more attempts).

To visualize this, we’ll create a plot graph and insert a linear line to identify any correlation between both variables.

```
# Field Goals Attempts vs Field Goals Made
combined_data %>%
  filter(Team == "Lakers") %>%
  ggplot(aes(x = FGA, y = FG, color = Win)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  scale_color_manual(values = c("violet", "darkgreen")) +
  theme_bw()+
  labs(
    color = "Result",
    x = "Field Goals Attempts",
    y = "Field Goals Made",
    title = "Laker Team's Field Goals (Winners vs Losers)"
  )
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



After creating the visualization we see various data points alongside outliers for both Lakers teams. The winning team, shown in green, made more shots with similar field goal attempts compared to the losing team in pink. This means they were more efficient at scoring.

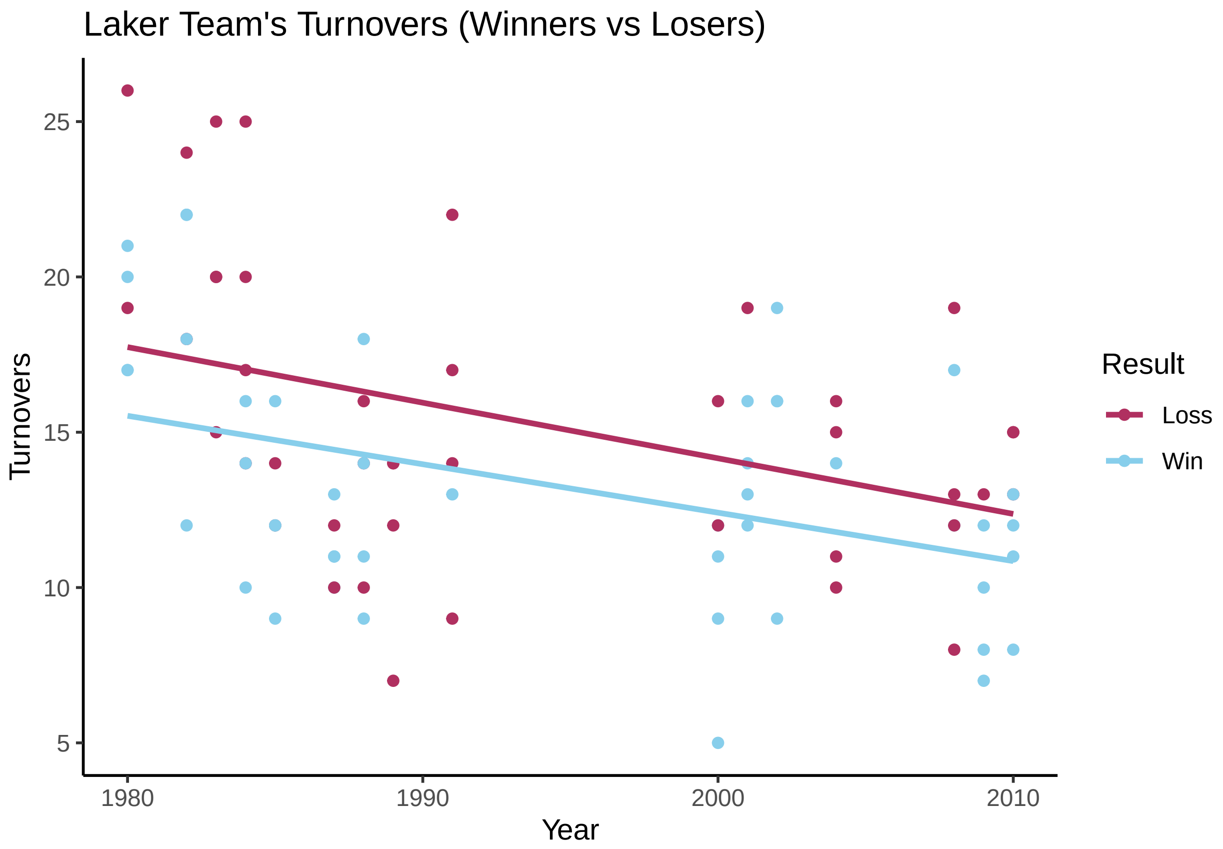
The green line for the winning team is higher, showing they generally made more shots with similar attempts. Meanwhile, the pink line for the losing team is lower, indicating they were less efficient at making shots.

From this information we can determine **the Winning sided Lakers were more efficient** from the field and have demonstrated a positive correlation with the Field Goals Made and Field Goals Attempted variables.

Now, let's look at their turnovers over the years. We'll use another graph to see how they compare in this aspect.

```r
combined_data %>%
  filter(Team == "Lakers") %>%
  ggplot(aes(x = Year, y = TOV, color = Win)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  scale_color_manual(values = c("maroon", "skyblue")) +
  theme_classic()+
  labs(
    color = "Result",
    x = "Year",
    y = "Turnovers",
    title = "Laker Team's Turnovers (Winners vs Losers)"
  )
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

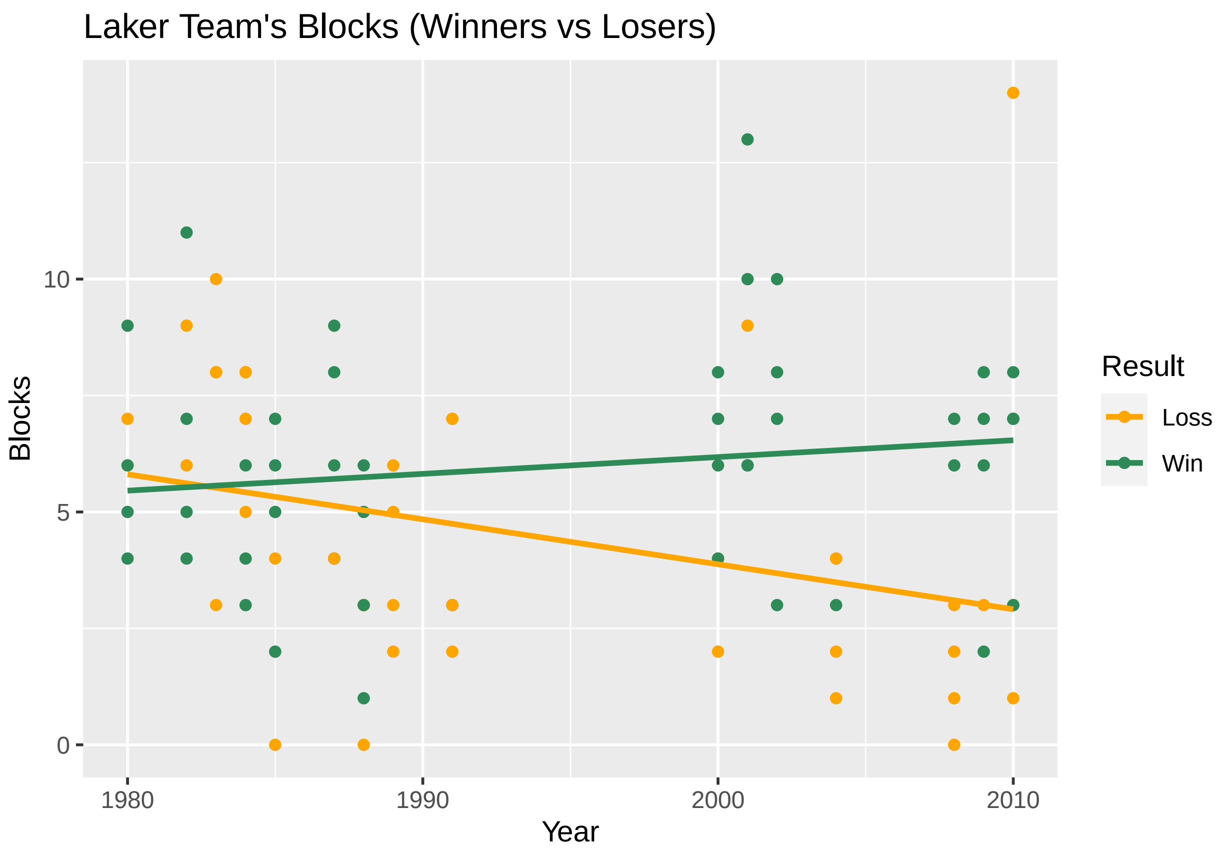


In the graph, we notice that both sides have decreased their turnovers over time, which is a positive sign as it means they're giving up fewer empty possessions. However, **the Winning side seems to perform better here.** While both sides have improved, the winning side has most of its values towards the bottom of the graph, indicating lower turnovers. Meanwhile, the losing team has most of its values higher up the graph, suggesting more turnovers.

For our last comparison, let's look at each side's blocks throughout the years. We'll use a similar approach to analyze this aspect.

```
combined_data %>%
  filter(Team == "Lakers") %>%
  ggplot(aes(x = Year, y = BLK, color = Win)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  scale_color_manual(values = c("orange", "seagreen")) +
  theme_grey()+
  labs(
    color = "Result",
    x = "Year",
    y = "Blocks",
    title = "Laker Team's Blocks (Winners vs Losers)"
  )
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



From the visualization, we notice that initially, the losing version of the Lakers had a lead for a year or two in block recordings, but then their numbers dropped over time. Meanwhile, the winning side showed significant improvement. Based on this, **the winning side** performed better in terms of blocks.

Overall, after analyzing the three categories comparing the winning and losing versions of the Lakers, **the winning side proved to be the better version of the Lakers.**

Now moving onto our third question:

How does home-court advantage impact NBA Finals performance?

We can answer this question in a variety of ways. To get as accurate to as possible we can see: * Who had a better win percentage, home or away team? * How did winning as the home team have an impact on NBA championship and Runner Up teams? * What were the categorical differences between home and away games?

Firstly, let's determine which team, home or away, had a better win percentage.

```
combined_data %>%
  filter(Win == "Win") %>%
  group_by(Team = Home) %>%
  summarise(Wins = n(), Percentage = (Wins/220)*100)
```

```
## # A tibble: 2 x 3
##   Team       Wins Percentage
##   <fct>     <int>      <dbl>
## 1 Away Team    86       39.1
## 2 Home Team   134       60.9
```

```
combined_data %>%
  filter(Win == "Loss") %>%
  group_by(Team = Home) %>%
  summarise(Losses = n(), Percentage = (Losses/220)*100)
```

```
## # A tibble: 2 x 3
##   Team      Losses Percentage
##   <fct>      <int>      <dbl>
## 1 Away Team    134       60.9
## 2 Home Team     86       39.1
```

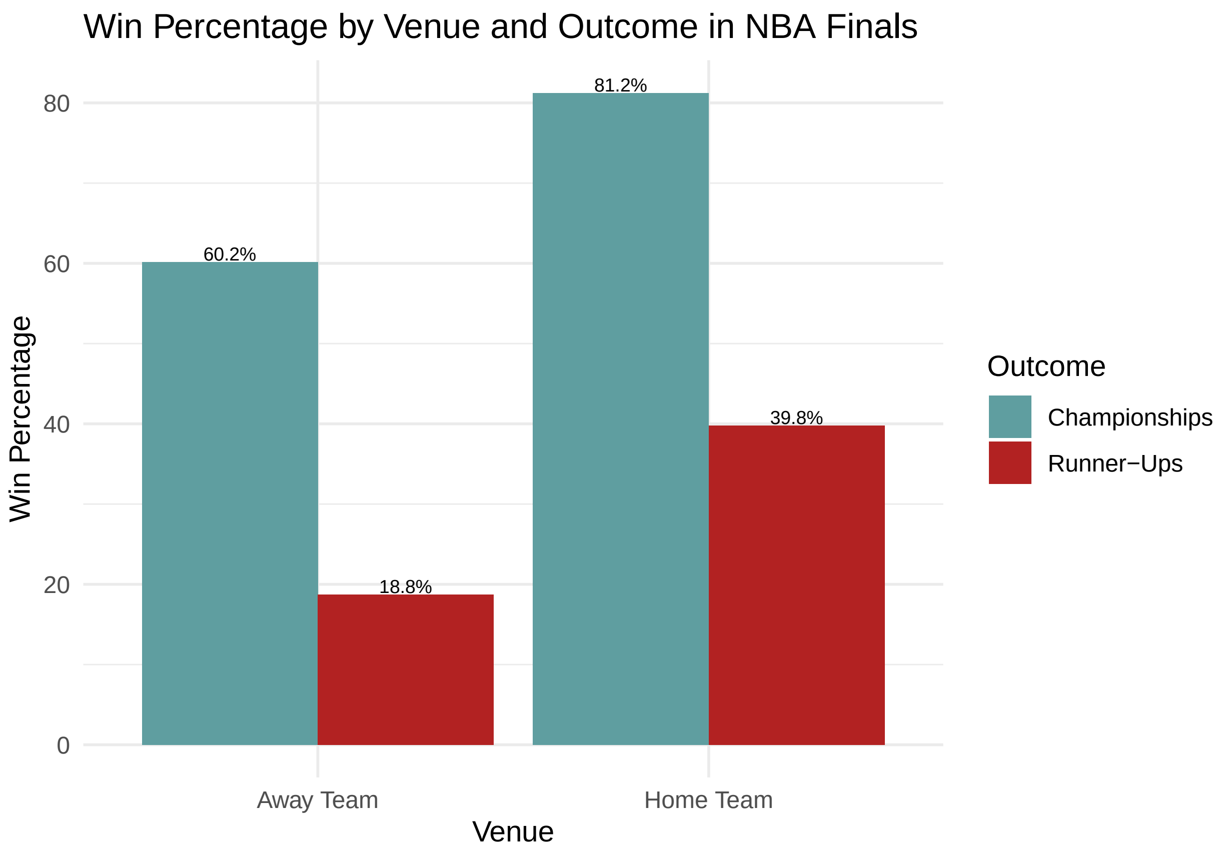
Performing a simple calculation, we find that the home team has won 134 games and lost 86, while the away team has won 86 games and lost 134. **This indicates that the home team has a 61% winning percentage compared to the away team's 39%.**

Next, we'll examine whether winning home games has an impact on Championship and Runner-Up teams. To do this, we'll create a bar chart that splits the home and away teams into two sides, each side representing either a Runner-Up or Championship team.

```
win_percentages <- combined_data %>%
  group_by(Home, Outcome) %>%
  summarise(Wins = sum(Win == "Win"), TotalGames = n()) %>%
  mutate(WinPercentage = Wins / TotalGames * 100)
```

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

```
win_percentages %>%
  ggplot(aes(x = Home, y = WinPercentage, fill = Outcome, label = sprintf("%.1f%%", WinPercentage))) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(position = position_dodge(width = 0.9), vjust = -0.1, size = 2.5) +
  scale_fill_manual(values = c("cadetblue", "firebrick")) +
  theme_minimal()+
  labs(
    fill = "Outcome",
    x = "Venue",
    y = "Win Percentage",
    title = "Win Percentage by Venue and Outcome in NBA Finals"
  )
```



Looking at the visualization, several insights emerge. Firstly, the Home team, both Championships and Runner-Ups, holds a slight edge in winning games, highlighting the impact of home games in basketball.

Furthermore, the highest win percentage of 81.2% is observed when a team is both playing at home and is built to win the NBA Championship.

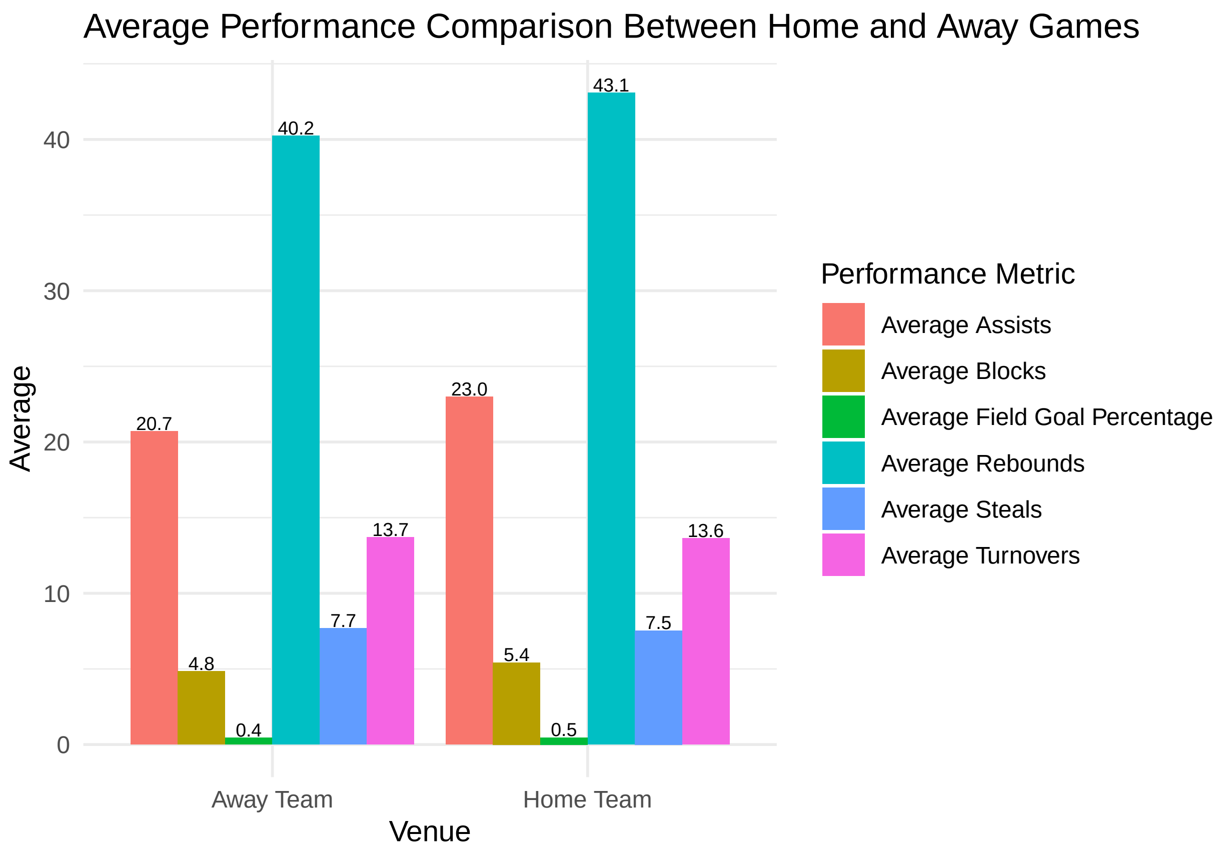
Interestingly, NBA Championship level teams perform well in both home and away games, with a winning percentage above 50%. Conversely, the worst combination is being an Away Team and a Runner-Up level team.

In summary, **teams that won the Championship and played at home had the greatest edge in winning games**, indicating the significance of home-court advantage in the NBA Finals. This suggests that having **home-court advantage increases the likelihood of winning the championship.**

Before determining the impact of being a home team, let's address the categorical differences between home and away games. To illustrate this, we'll use a stacked bar plot visualization that calculates the mean stats from both offensive and defensive categories and categorizes them based on venue type (home or away).

```r
performance_comparison <- combined_data %>%
  group_by(Home) %>%
  summarise(
    "Average Field Goal Percentage" = mean(FGP, na.rm = TRUE),
    "Average Rebounds" = mean(TRB, na.rm = TRUE),
    "Average Assists" = mean(AST, na.rm = TRUE),
    "Average Steals" = mean(STL, na.rm = TRUE),
    "Average Blocks" = mean(BLK, na.rm = TRUE),
    "Average Turnovers" = mean(TOV, na.rm = TRUE)
  )
```

```r
performance_comparison %>%
  pivot_longer(cols = -Home, names_to = "PerformanceMetric", values_to = "Average") %>%
  ggplot(aes(x = Home, y = Average, fill = PerformanceMetric, label = sprintf("%.1f", Average))) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(position = position_dodge(width = 0.9), vjust = -0.1, size = 2.5) +
  labs(x = "Venue", y = "Average", fill = "Performance Metric") +
  ggtitle("Average Performance Comparison Between Home and Away Games") +
  theme_minimal()
```



After creating the visualization, let's compare the stats in which the Home team performed better than the Away team.

One notable difference favoring the Home team is their Average Field Goal Percentage. This is logical as playing in their home arena allows players to become more accustomed to the court, environment, and rims, potentially leading to higher shooting accuracy. Additionally, the Home team excelled in sharing the basketball, recording an average of 23 assists compared to the Away team's 20.7. Moreover, the Home team performed well in rebounds, blocks, and turnovers (averaging slightly less with a difference of 0.1).

The only stat where the Away team outperformed the Home team was average steals. Perhaps the Away team was more defensively focused due to the pressure circumstances, knowing they needed to get stops to overcome the Championship team.

Overall, the category averages favor the Home team, with better scores in 5 out of 6 categories, while the Runner-Ups only maintained an advantage in 1 out of 6 categories.

Bringing this question to a close, circling back to our main inquiry of "How does home-court advantage impact NBA Finals performance?", we've answered it by analyzing sub-questions and gaining insight into the impact of playing on home court. From observing high win percentages to the advantages of holding home-court advantage in the NBA Finals, and now examining various categories elevated by being the home team, it's clear that **there is a significant impact when it comes to playing as the Home team in the NBA Finals.**

Now lastly, our fourth and final analysis question:

Over-time, how has the play style of winning championships evolved, more offense centered or defensive minded?

To answer the question of how the play style of winning championships has evolved over time, we can create time series graphs, such as line graphs, of various categories factored by Championships and Runner-Ups. Since we want to determine if the play style has become more offense-centered or defensive-minded, we'll split the statistical categories into Offensive and Defensive categories, *similar to what we did in the first analysis question.*

Offensive categories: Field Goals, Three-Pointers Made, Free Throws Made, Assists, Team Total Points
Defensive categories: Rebounds, Steals, Blocks, Turnovers, Personal Fouls

First, let's create a mini-dataframe that contains the averages of each category grouped by the Year and the team type (Championships or Runner-Ups). This will allow us to analyze trends over time.

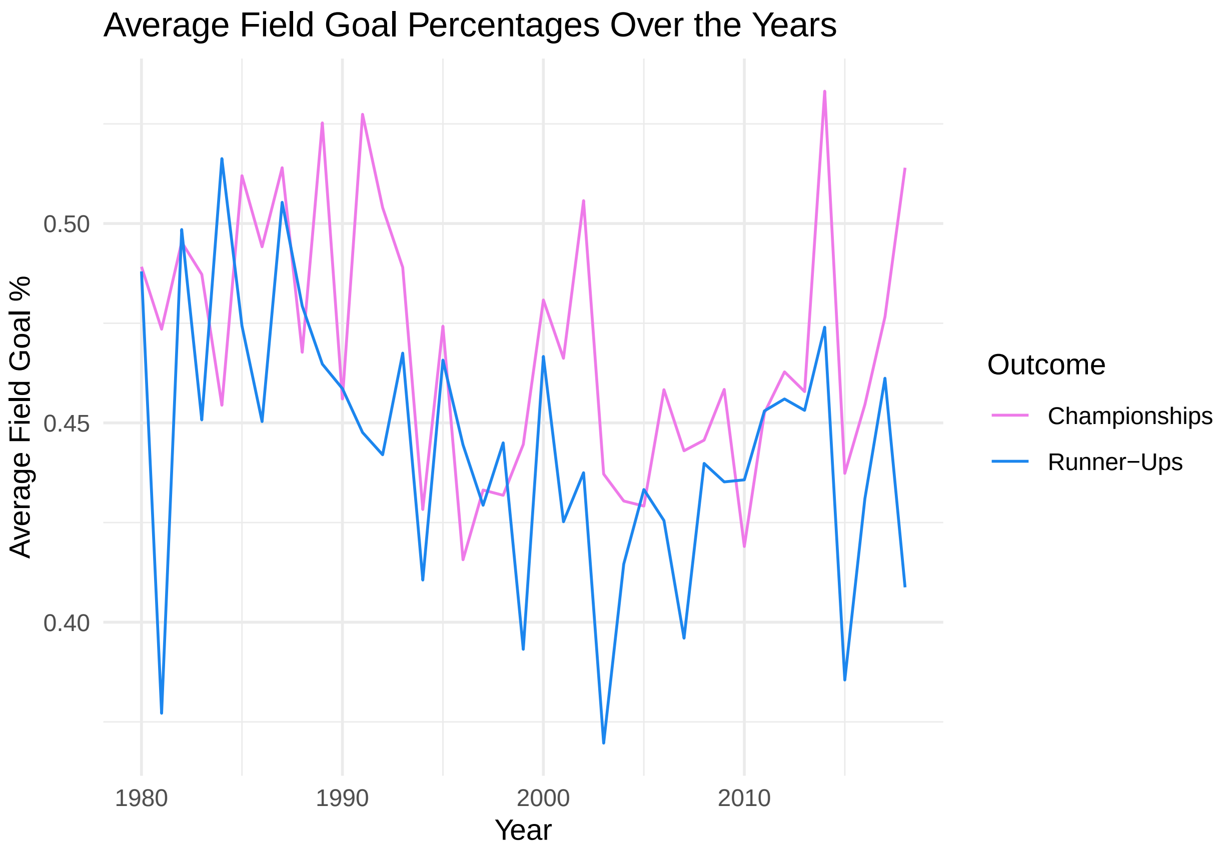
```r
performance_comparison_pt2 <-
  combined_data %>%
  group_by(Year, Outcome) %>%
  summarise(
    Avg_FG_Percent = mean(FGP), #Offense
    Avg_Three_Percent = mean(TPP), #Offense
    Avg_FT_Percent = mean(FTP), #Offense
    Avg_Defensive_Rebounds = mean(DRB), #Defense
    Avg_Assists = mean(AST), #Offense
    Avg_Steals = mean(STL), #Defense
    Avg_Blocks = mean(BLK), #Defense
    Avg_Turnovers = mean(TOV), #Defense
    Avg_Personal_Fouls = mean(PF), #Defense
    Avg_Points = mean(PTS) #Offense
  ) %>%
  ungroup()
```

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

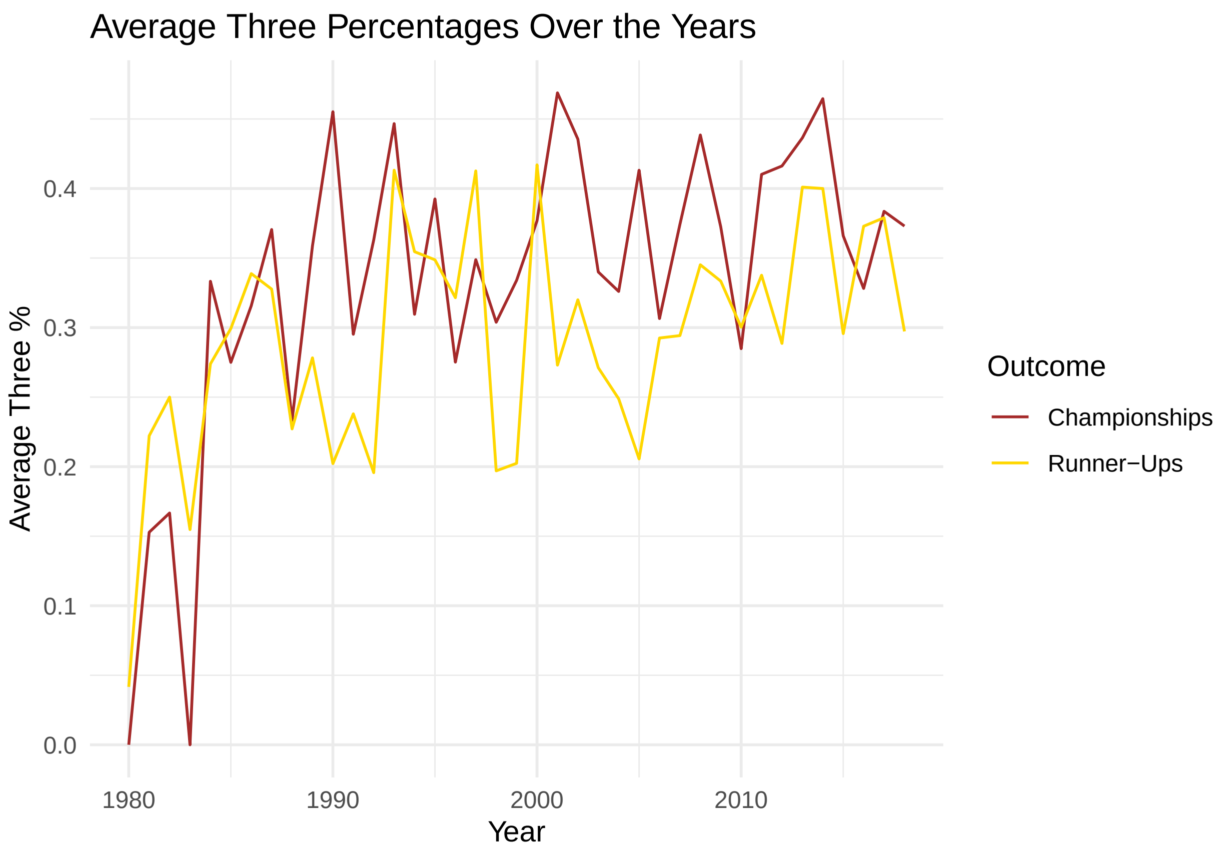
Once our dataset with averages is ready, we can start creating our line graphs. To keep things simple and easy to understand, we'll use the same style for all graphs. This way, it's straightforward to browse through each category.

First, let's focus on the Offensive categories:

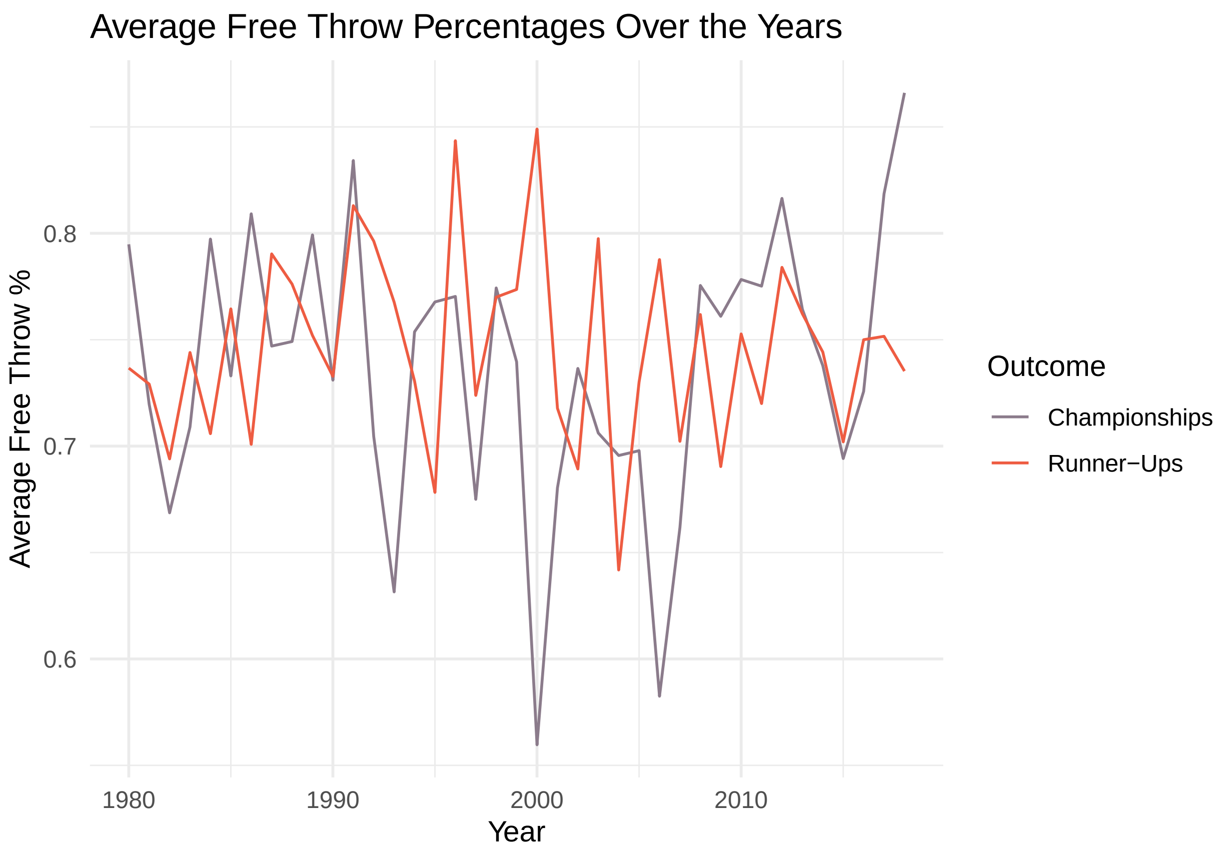
```r
#Offense Comparisons

performance_comparison_pt2 %>%
  ggplot(aes(x = Year, y = Avg_FG_Percent, color = Outcome))+
  geom_line()+
  scale_color_manual(values = c("orchid2", "dodgerblue2")) +
  labs(x = "Year", y = "Average Field Goal %", fill = "Outcome") +
  ggtitle("Average Field Goal Percentages Over the Years") +
  theme_minimal()
```



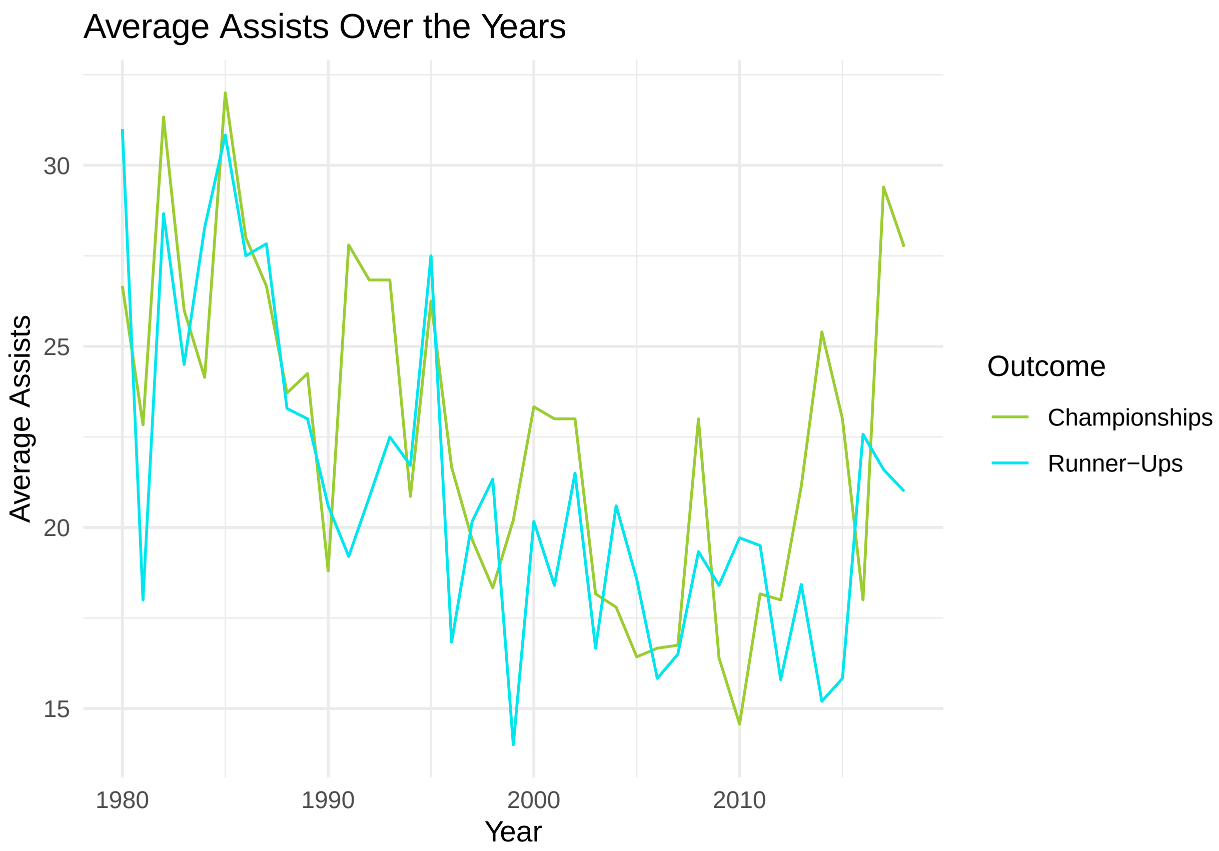
```
performance_comparison_pt2 %>%
  ggplot(aes(x = Year, y = Avg_Three_Percent, color = Outcome))+
  geom_line()+
  scale_color_manual(values = c("brown", "gold")) +
  labs(x = "Year", y = "Average Three %", fill = "Outcome") +
  ggtitle("Average Three Percentages Over the Years") +
  theme_minimal()
```



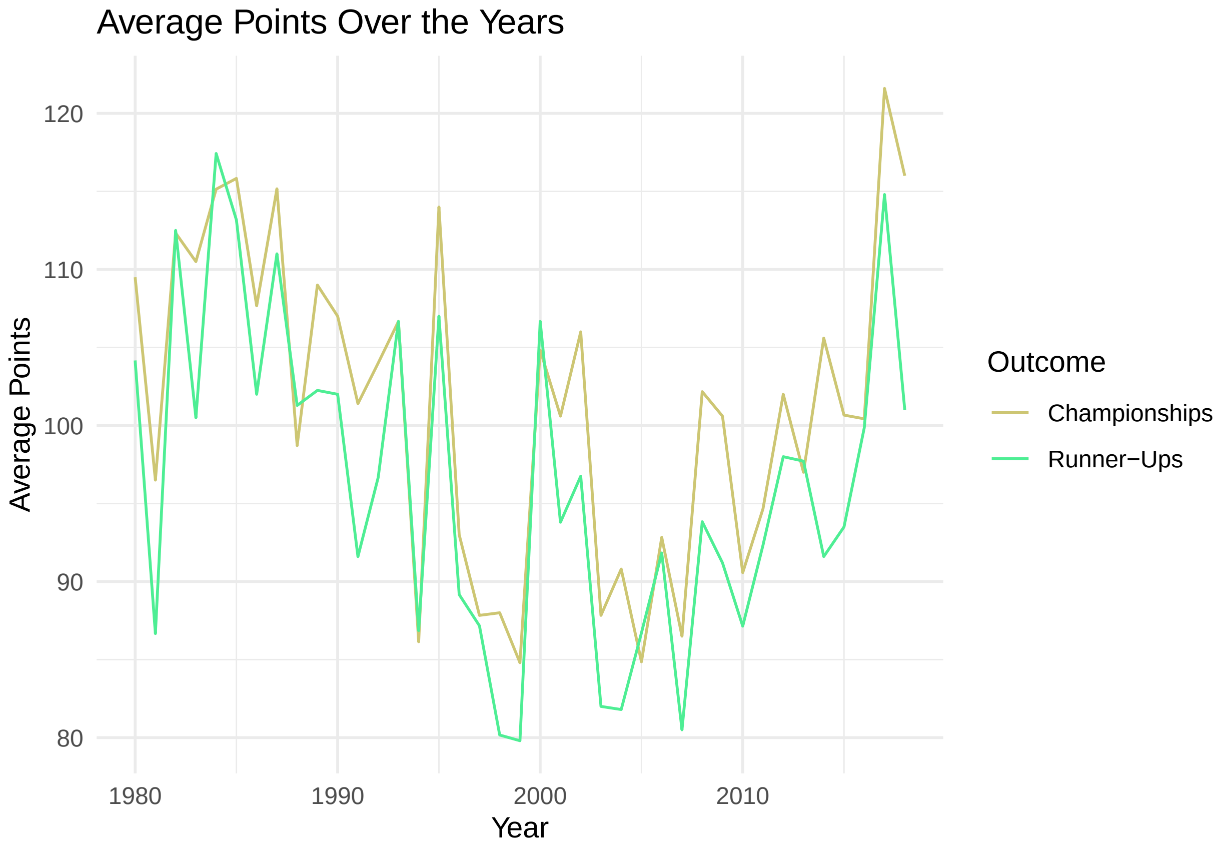
```
performance_comparison_pt2 %>%
  ggplot(aes(x = Year, y = Avg_FT_Percent, color = Outcome))+
  geom_line()+
  scale_color_manual(values = c("thistle4", "tomato2")) +
  labs(x = "Year", y = "Average Free Throw %", fill = "Outcome") +
  ggtitle("Average Free Throw Percentages Over the Years") +
  theme_minimal()
```



```
performance_comparison_pt2 %>%
  ggplot(aes(x = Year, y = Avg_Assists, color = Outcome))+
  geom_line()+
  scale_color_manual(values = c("yellowgreen", "turquoise2")) +
  labs(x = "Year", y = "Average Assists", fill = "Outcome") +
  ggtitle("Average Assists Over the Years") +
  theme_minimal()
```



```
performance_comparison_pt2 %>%
  ggplot(aes(x = Year, y = Avg_Points, color = Outcome))+
  geom_line()+
  scale_color_manual(values = c("khaki3", "seagreen2")) +
  labs(x = "Year", y = "Average Points", fill = "Outcome") +
  ggtitle("Average Points Over the Years") +
  theme_minimal()
```

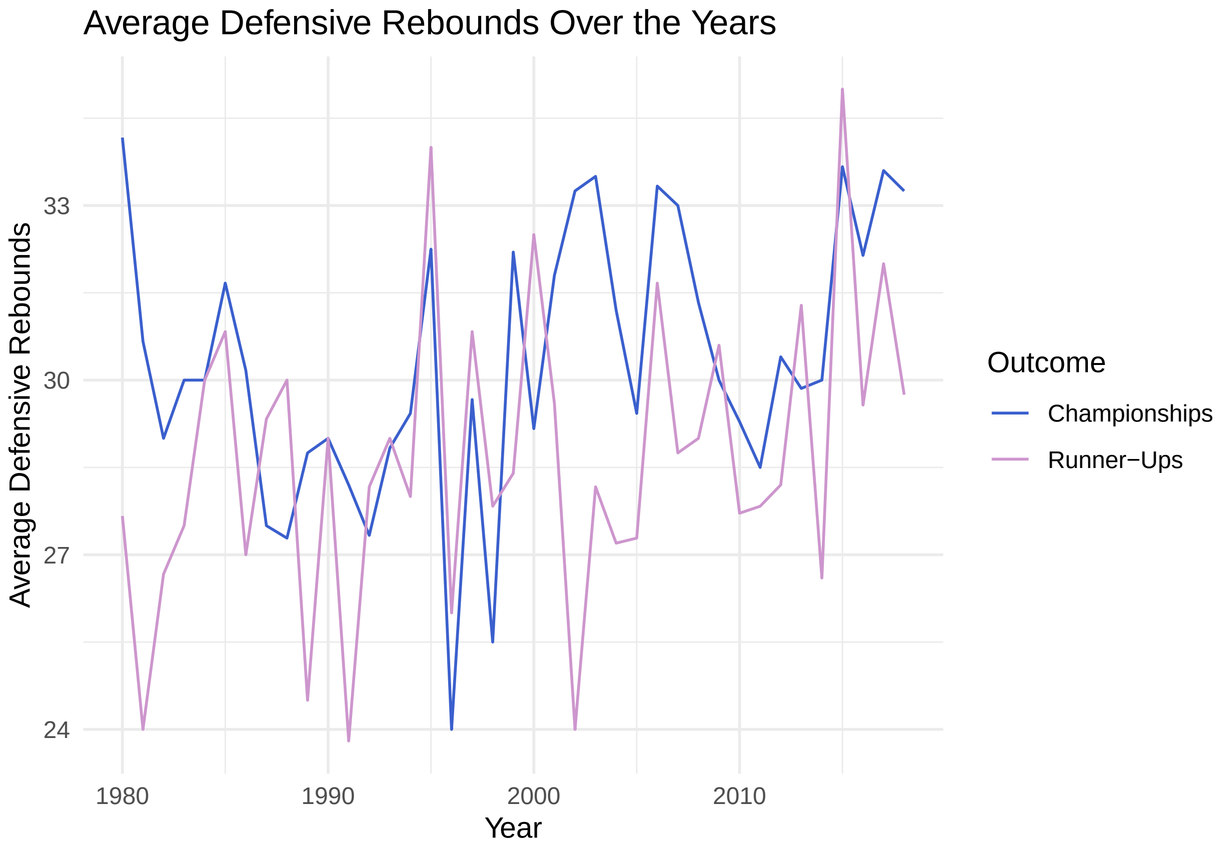


Analyzing Offensive Trends:

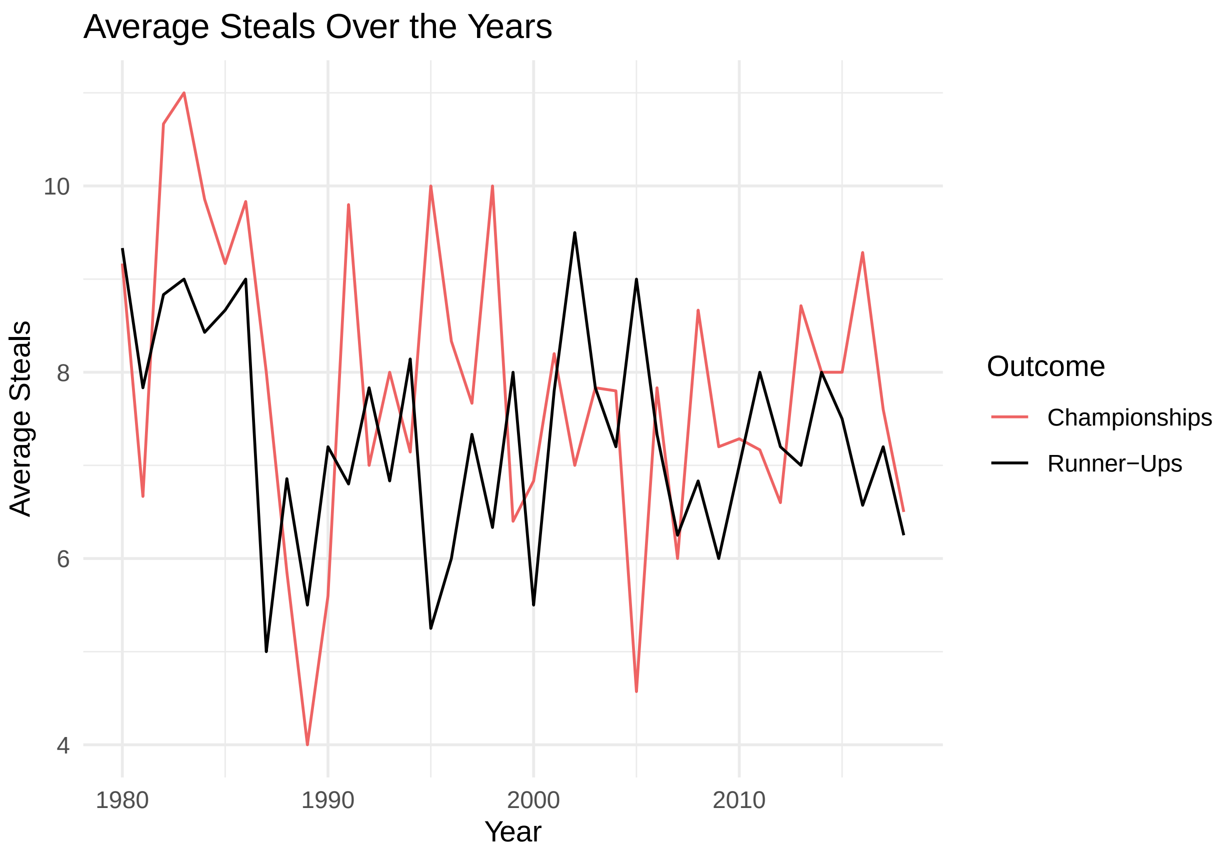
- **Field Goal Percentages:** shows a noticeable drop from 1990-2003, with championship teams maintaining somewhat higher percentages despite fluctuations. This could be due to increased shot attempts, like 2-for-1's between shot clocks, affecting percentages.
- **Three-Point Percentages:** displays a massive improvement of percentages after the years 1983 and 1984. Championship teams generally outperformed Runner-Ups, even though Runner-Ups dipped to 20% in 1990, 1999, 2005. Higher percentages of makes and attempts taken can be seen supported from trends on the graph. Teams predicate shooters around their players since having a three-point ability is a necessity in today's game.
- **Free Throw Percentages:** illustrates various highs and lows for both sides. Runner-Ups maintained around 70% from 1980 to 2018, while Championship teams faced drops as low as 55% except towards the end when they soared above 85%. Runner-Ups seemed more consistent, while Championships were more erratic. Recall we witnessed that the Lakers are a team that has reached the finals frequently, and knowing that they had superstar Shaquille O'Neal definitely impacted their team's percentage.
- **Average Assists:** for both sides declined over the years, then spiked after 2010, particularly for Championship teams. This might be because teams played more isolation basketball, reducing passing frequency. In the early years, players frequently passed the ball, evident in old footage where they took 3-4 dribbles before passing.
- **Average Points:** points scored showed a rise, then a decline, followed by an increase beyond the 2010s. The decline could be attributed to defensive-oriented teams like the San Antonio Spurs, Miami Heat, and Bad Boy Pistons. The subsequent rise mirrors today's more offensive-oriented game, where teams exploit foul calls and shoot more frequently.

After analyzing how the offensive categories has changed over the years, it's time to view the defensive end.

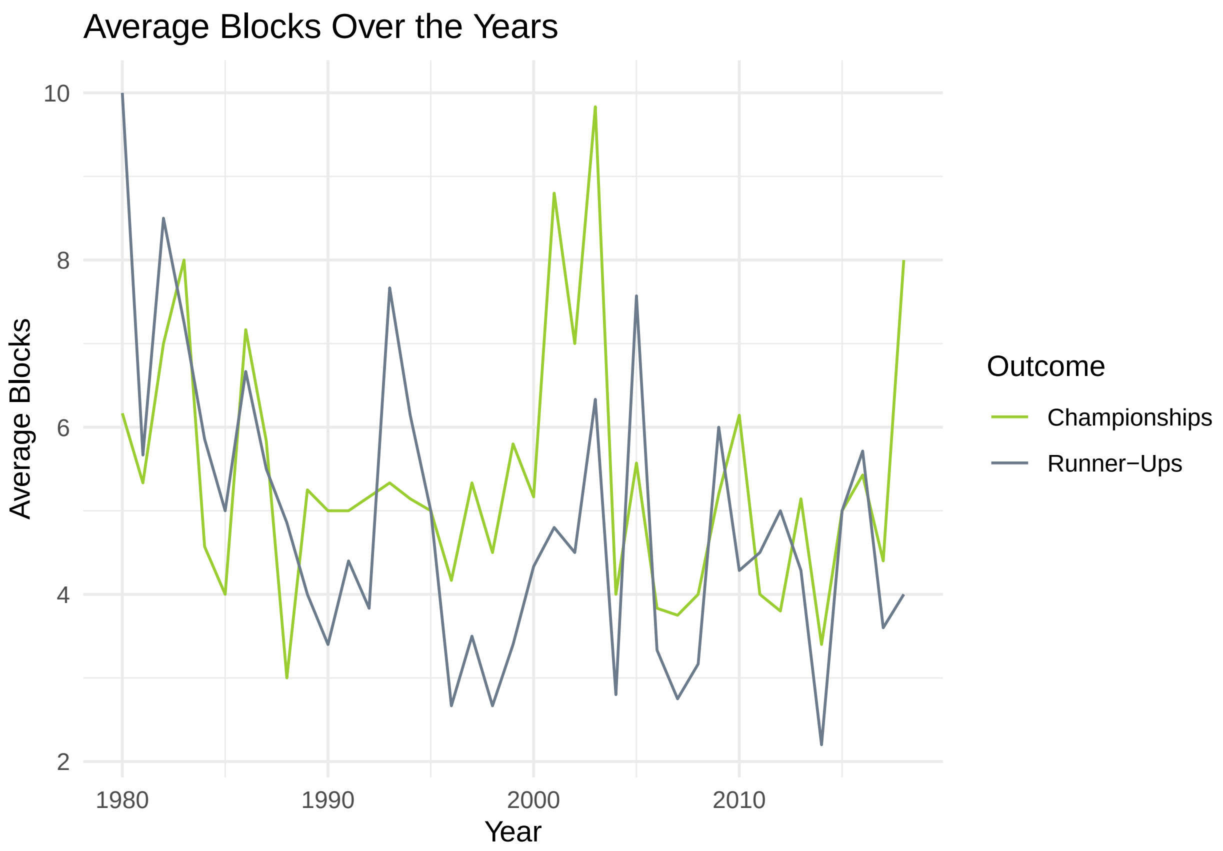
```
#Defensive Comparisons

performance_comparison_pt2 %>%
  ggplot(aes(x = Year, y = Avg_Defensive_Rebounds, color = Outcome))+
  geom_line()+
  scale_color_manual(values = c("royalblue3", "plum3")) +
  labs(x = "Year", y = "Average Defensive Rebounds", fill = "Outcome") +
  ggtitle("Average Defensive Rebounds Over the Years") +
  theme_minimal()
```



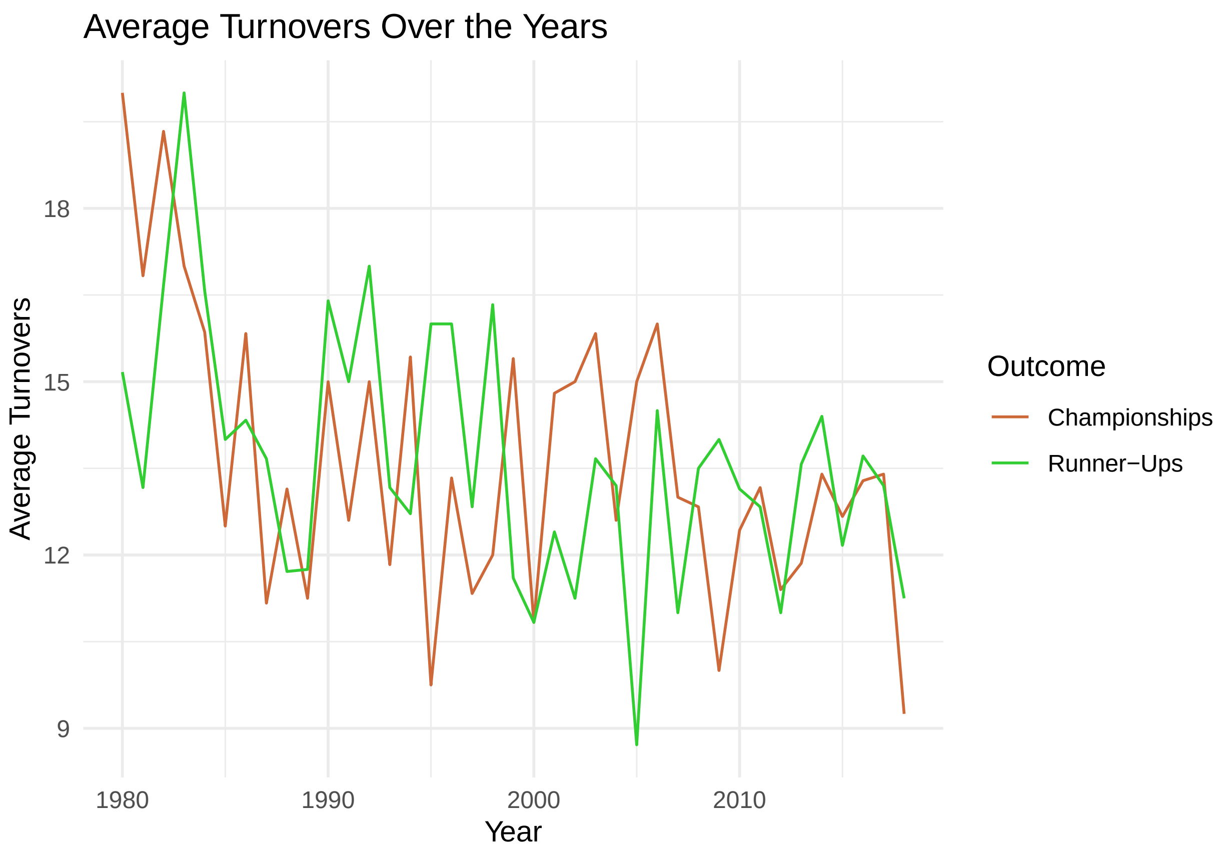
```
performance_comparison_pt2 %>%
  ggplot(aes(x = Year, y = Avg_Steals, color = Outcome))+
  geom_line()+
  scale_color_manual(values = c("indianred2", "black")) +
  labs(x = "Year", y = "Average Steals", fill = "Outcome") +
  ggtitle("Average Steals Over the Years") +
  theme_minimal()
```



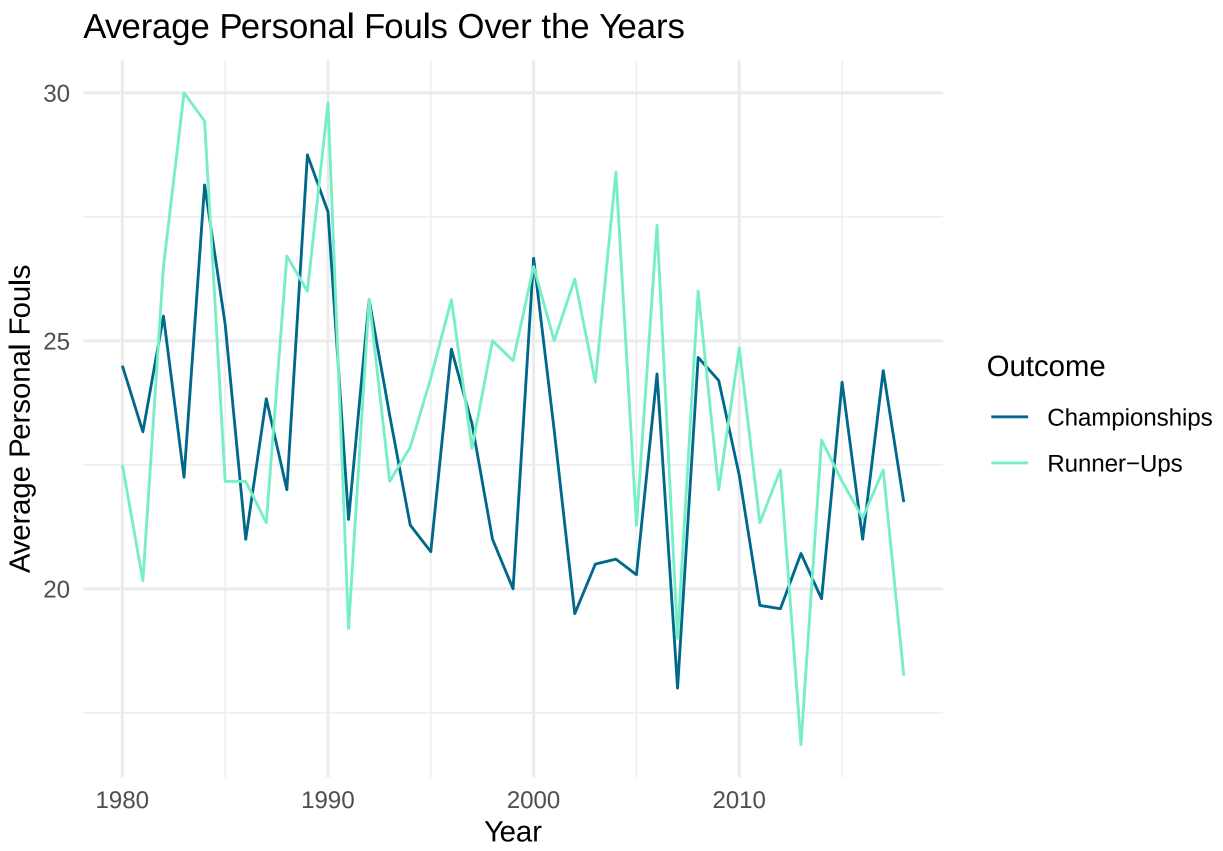
```
performance_comparison_pt2 %>%
  ggplot(aes(x = Year, y = Avg_Blocks, color = Outcome))+
  geom_line()+
  scale_color_manual(values = c("olivedrab3", "slategray4")) +
  labs(x = "Year", y = "Average Blocks", fill = "Outcome") +
  ggtitle("Average Blocks Over the Years") +
  theme_minimal()
```



```
performance_comparison_pt2 %>%
  ggplot(aes(x = Year, y = Avg_Turnovers, color = Outcome))+
  geom_line()+
  scale_color_manual(values = c("sienna3", "limegreen")) +
  labs(x = "Year", y = "Average Turnovers", fill = "Outcome") +
  ggtitle("Average Turnovers Over the Years") +
  theme_minimal()
```



```
performance_comparison_pt2 %>%
  ggplot(aes(x = Year, y = Avg_Personal_Fouls, color = Outcome))+
  geom_line()+
  scale_color_manual(values = c("deepskyblue4", "aquamarine2")) +
  labs(x = "Year", y = "Average Personal Fouls", fill = "Outcome") +
  ggtitle("Average Personal Fouls Over the Years") +
  theme_minimal()
```



Analyzing Defensive Trends:

- **Average Defensive Rebounds:** initially, Championship and Runner-Up teams showed opposite trends, but gradually became more similar over the years. Runner-Ups experienced consistent dips, particularly in the 1980s, 1990s, and 2003s, while Championships saw a major drop only in 1996. Fluctuations in defensive rebounds may reflect the defensive construction of teams over time.
- **Average Steals:** started strong in the 1980s for both sides but dropped significantly from 1987-1989. There's been a partial resurgence since then, though not reaching the initial 8-10 steals per game range, dropping to around 6 steals per game by 2018. Teams may be more conservative on steals due to the risk of fouls, exploiting referees' tendencies to favor the whistle.
- **Average Blocks:** Runner-Ups performed strongly, averaging 10 blocks, but by the end (2018 finals), this dropped to 4 blocks, allowing Championship teams to dominate. Championship teams also peaked in 2003-2004, averaging just above 9.5 blocks. Overall, there's been a drop-off in blocks over the years with occasional peaks, mirroring the Steals graph.
- **Average Turnovers:** trend in turnovers shows a positive shift in the game's evolution. Over the years, turnovers have decreased significantly, starting above 18 turnovers and dropping to as low as 9 turnovers, nearly halving their averages. This change reflects advancements in skill development. While there were skillful players in earlier years, today's NBA benefits from improved training camps and technology, leading to better ball control and handles for a majority of players.
- **Average Personal Fouls:** trend in personal fouls shows a gradual decline, with the lowest drop occurring just below 15 in 2013 and the highest reaching 30 in 1983. This trend is surprising considering the perceived increase in foul calls in today's game compared to the past, where more physicality was allowed. However, the graph indicates otherwise, with a decrease in foul calls over time. One explanation could be players' increased game IQ, leading to more conservative defensive play to avoid

fouls. Additionally, teams may intentionally foul players who struggle with free throws to gain an extra possession.

Over the years, basketball has undergone significant changes in play styles, marked by increases in points, three-point and free throw percentages, alongside decreases in turnovers, foul calls, and steals. This shift suggests a clear trend towards a more offense-focused game. While many analysts and players acknowledge that defense wins championships, the current emphasis on offensive firepower has somewhat overshadowed the importance of defensive tenacity on the court.

Summary and Conclusions

Summary

At the beginning of this project, I set a goal of wanting to explore the evolving trends. However since it was such a broad topic to explore, I knew I had to create smaller and more specific questions to provide me an overall idea of how the game has changed over the years, and what areas teams in today's game should focus in order to strive for championships.

Conclusions

Before we explore the main findings of this project, let's refer to the main project's goal: *"Analyze the evolving trends in play styles of championship-winning NBA Finals teams and examine the influence of home-court advantage on team performances."*

Main Findings of my Project:

Looking at the trends in play-styles:

- NBA Teams strategy of winning games pick up from their score-first mentality with a massive increase in three point shooting *(around 40% compared to 25-30% in the olden days).*
- Turnover averages have massively gone down. Teams before averaged close to 15 turnovers and even as high as 18+, however now teams surface the 9 average turnovers range. This indicates a variety of reasons, either players have become more skillful and IQ than players in the past with controlling the ball and avoid ill-advised passes OR the level of defense has gone down, since defensive stops can force teams to overturn the ball.
- Average Field Goal Percentage has dropped in between 1995 - 2010 with a slow increase towards 2018. Point and shot inflation has been a main difference in today's game where teams would throw up a higher number shots such as 2-for-1's especially with the shot-clock going down towards the end of the quarter. Another reason is the era of outside shooting taking over inside shot taking, and as we know inside scoring provides a higher percentage than shooting behind the arc.
- Recordings of personal fouls have gone starting at 30 on average to as low as 15 a game within the recent years. Emphasis on rule changes such as hand-checking and physical defense, allows offensive players to maneuver fluently resulting in defenders being more cautious about committing fouls.
- Defensive rebounds averages started out at its lowest point with 24 a game around 1982-1983 to 2018 being at it's average with 33-35. The increase in three-point and field goal attempts may have an effect, since it leads to longer rebounds towards the perimeter, where the defense often staggers. Resulting in more secured of rebounds by the defense. In addition some teams run with small-ball lineups in today's game in favor of spacing the floor and being more agile on defense. Therefore the other team may have a taller line-up running, leading to more defensive rebounds secured.

Examining home-court advantage:

*The winning percentage that both Championship and Runner-Up teams have both favor them when playing at home instead of away. This is due to home-court providing a familiar surrounding and crowd support resulting to better performance and higher confidence levels.

- Highest winning percentage of 81.2% is observed when a team is both playing at home and built to win a championship. Chances of success enhance with a combination of talent and depth alongside a supportive home environment.

*Being an away team, especially as a Runner-Up results in the lowest winning percentage of 18.8%. Teams that may be weaker than their opponents and well as facing them on the road can slim a teams chance towards a championship.

Further Ideas

Looking towards the future, I hope to return to this analysis project and expand it further by applying a Logistic Regression model such as creating a NBA Championship predictor. Then I could provide it statistics from previous NBA Championship teams, and train it in order to find the probability of success of winning a championship for current NBA teams.