

Predicting NBA Team Success - 2870 Final Project

Joey Gilmartin, Nick Kramer, Tyler Thompson

2024-12-10

```
#knitr
knitr::opts_chunk$set(echo = T,
                      warning = F,
                      message = F,
                      fig.align = "center")

#load packages
pacman::p_load(tidyverse, regclass, broom, GGally, rpart, rpart.plot, caret, class, FNN)

#update theme to be BW and title centered
theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = .5),
             plot.subtitle = element_text(hjust = .5))

colors <- read.csv('teamcolors_bigfour.csv') |>
  filter(league == 'nba') |>
  mutate( team = name ) |>
  dplyr::select(team, primary, secondary)

#load in a self made champions dataset to update champion column
champions <- read.csv('champions.csv') |>
  dplyr::select(-X) # loaded extra empty column named X, remove it
```

Introduction

Research Question: To what extent do basic NBA statistics predict the outcome of a teams NBA season? More specifically, are we able to predict the likelihood of a team making the NBA playoffs or winning the NBA finals?

Introduce Data

```

#load in data and take out the incomplete 2025 season and seasons before 1980
team_per_game <- read.csv("Team Stats Per Game.csv") |>
  filter(season != 2025,
         # remove league average rows
         team != "League Average",
         season > 1979) |>
# used the champions dataframe to add 1s for all champions in the column
left_join(
  y = champions,
  by = c('season', 'team')
) |>
mutate(
  #change playoffs to be 0 and 1 not F and T
  playoffs = as.numeric(playoffs),
  #manually add a championship column
  champion = if_else( is.na(champion), true = 0, false = 1)
) |>
#select some columns we may want to use
dplyr::select(season, team, abbreviation, playoffs, champion, fg_percent, x3p_percent, orb_per_game:pts_per_game, -pf_per_game)

opp_per_game <- read.csv("Opponent Stats Per Game.csv") |>
  #take out the incomplete 2025 season and seasons before 1980
  filter(season != 2025,
         team != "League Average",
         season > 1979) |>
  #select some columns we may want to use
  dplyr::select(season, team, opp_fg_percent, opp_x3p_percent, opp_trb_per_game, opp_ast_per_game, opp_tov_per_game, opp_pts_per_game)

team_summaries <- read.csv('Team Summaries.csv') |>
  #take out the incomplete 2025 season and seasons before 1980
  filter(season != 2025,
         team != "League Average",
         season > 1979) |>
  #select some columns we may want to use
  dplyr::select(season, team, o_rtg, d_rtg, n_rtg, ts_percent, e_fg_percent, opp_e_fg_percent)

#join all 3 data frames into one df
nba <- team_per_game |>
  left_join( y = opp_per_game,
            by = c('season', 'team')) |>
  left_join( y = team_summaries,
            by = c('season', 'team')) |>
  left_join( y = colors,
            by = 'team') |>
  # final decision on what to keep in nba dataset
  dplyr::select(season, team, abbreviation, playoffs, champion, n_rtg, pts_per_game, fg_percent, x3p_percent, trb_per_game, ast_per_game, stl_per_game, blk_per_game, tov_per_game, opp_pts_per_game, opp_fg_percent, primary, secondary)

#remove the old dataframes
rm(team_per_game, opp_per_game, team_summaries)

```

```
#skim to confirm all variables are full
skimr::skim(nba)
```

Data summary

Name	nba
Number of rows	1254
Number of columns	18
Column type frequency:	
character	4
numeric	14
Group variables	
None	

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
team	0	1.00	9	33	0	39	0
abbreviation	0	1.00	3	3	0	40	0
primary	118	0.91	7	7	0	22	0
secondary	130	0.90	7	7	0	20	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
season	0	1	2003.12	12.70	1980.0	1993.00	2004.00	2014.00	2024.00	
playoffs	0	1	0.56	0.50	0.0	0.00	1.00	1.00	1.00	
champion	0	1	0.04	0.19	0.0	0.00	0.00	0.00	1.00	
n_rtg	0	1	0.01	4.85	-15.2	-3.30	0.30	3.50	13.40	
pts_per_game	0	1	103.21	7.58	81.9	97.30	102.80	108.90	126.50	
fg_percent	0	1	0.46	0.02	0.4	0.45	0.46	0.48	0.54	
x3p_percent	0	1	0.34	0.05	0.1	0.32	0.35	0.36	0.43	
trb_per_game	0	1	42.86	2.13	35.6	41.40	42.80	44.30	51.70	
ast_per_game	0	1	23.32	2.64	15.6	21.30	23.20	25.20	31.40	
stl_per_game	0	1	8.00	1.07	5.5	7.20	7.90	8.67	12.80	
blk_per_game	0	1	5.04	0.94	2.4	4.40	4.90	5.60	8.70	
tov_per_game	0	1	15.35	1.83	11.1	14.10	15.10	16.40	22.80	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
opp_pts_per_game	0	1	103.21	7.61	83.4	97.40	103.20	108.50	130.80	
opp_fg_percent	0	1	0.46	0.02	0.4	0.45	0.46	0.48	0.54	

```
head(nba)
```

```
##      season      team abbreviation playoffs champion n_rtg pts_per_game
## 1   2024    Atlanta Hawks          ATL         0         0  -2.2      118.3
## 2   2024    Boston Celtics          BOS         1         1  11.6      120.6
## 3   2024    Brooklyn Nets          BRK         0         0  -2.9      110.4
## 4   2024    Chicago Bulls          CHI         0         0  -1.4      112.3
## 5   2024    Charlotte Hornets        CH0         0        0 -10.5      106.6
## 6   2024    Cleveland Cavaliers      CLE         1         0   2.5      112.6
##      fg_percent x3p_percent trb_per_game ast_per_game stl_per_game blk_per_game
## 1      0.465      0.364      44.7         26.6         7.5         4.5
## 2      0.487      0.388      46.3         26.9         6.8         6.6
## 3      0.456      0.362      44.1         25.6         6.8         5.2
## 4      0.470      0.358      43.8         25.0         7.8         4.8
## 5      0.460      0.355      40.3         24.8         6.9         4.5
## 6      0.479      0.367      43.3         28.0         7.4         4.6
##      tov_per_game opp_pts_per_game opp_fg_percent primary secondary
## 1          13.5         120.5         0.495 #e13a3e #c4d600
## 2          11.9         109.2         0.453 #008348 #bb9753
## 3          13.1         113.3         0.470 #061922 <NA>
## 4          12.2         113.7         0.473 #ce1141 #061922
## 5          13.8         116.8         0.494 #1d1160 #008ca8
## 6          13.6         110.2         0.463 #860038 #fdbb30
```

Description of Our Data

Our NBA data was found on Kaggle and compiled by Sumitro Datta. The data goes back to 1947 is updated to the present. Our data is not a sample, as it contains the entire population of NBA statistics from 1947 to modern day. The NBA is a competitive league with rules and regulations that are meant to be fair, therefore, we do not suspect any bias in sampling or our measurements. This was an observational study, as our data was recorded by the NBA after every game. This data interests us because we all are fans of the NBA, and enjoy the statistical side of the game. The class should find it interesting because statistics can give us a perspective on sports that is otherwise incomprehensible. The data came mostly full, however some advanced statistics were not tracked in the early NBA, so we will only use data from 1980 and on. We also removed data for the incomplete 2025 season. Additionally, we removed rows that contained the league average data. We also merged the data frame with a list of NBA Champions to create a column indicating if a team has won the championship. Lastly, since our data frame was extremely large, we got rid of many columns of statistics that we didn't need.

Data Visualizations and Investigation

NBA Championships By Franchise

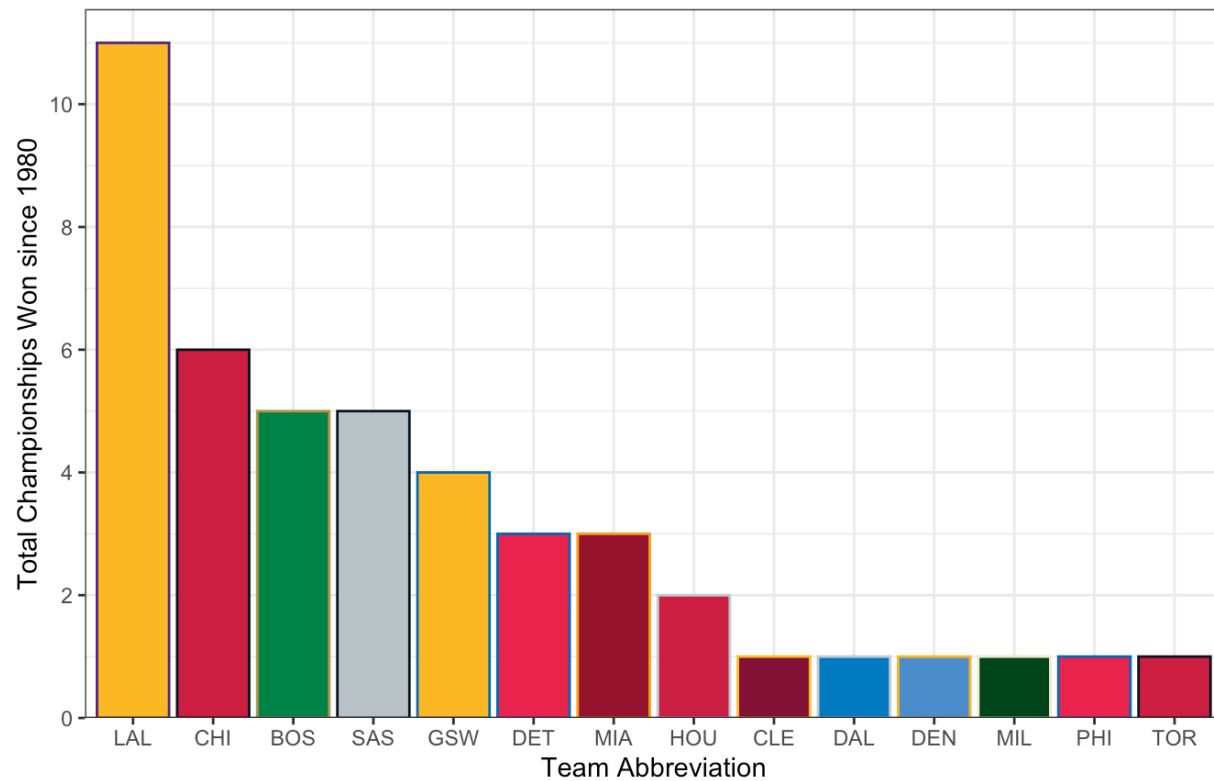
```
nba_champ_sum <- nba |>
  #trim columns for simplicity
  dplyr::select(team, abbreviation, champion, primary, secondary) |>
  #sum champions
  summarize(
    .by = c(team, abbreviation, primary, secondary),
    total_championships = sum(champion)
  ) |>
  #get rid of teams with 0
  filter( total_championships != 0)

gg_champs_by_franch <- ggplot(
  data = nba_champ_sum,
  mapping = aes(
    #order the bars
    x = fct(abbreviation) |> fct_infreq( total_championships ),
    y = total_championships,
    fill = primary,
    color = secondary
  )
) +
  #don't show legend
  geom_col(show.legend = F) +
  #add labels
  labs(
    x = "Team Abbreviation",
    y = "Total Championships Won since 1980",
    title = "NBA Championships Won By Franchise",
    subtitle = "Since 1980"
  ) +
  scale_fill_identity() +
  scale_color_identity() +

  scale_y_continuous(expand = c(0, 0, 0.05, 0),    #putting bars on the x axis
                     breaks=seq(0,12,by=2))      #tick marks count by 2

gg_champs_by_franch
```

NBA Championships Won By Franchise Since 1980

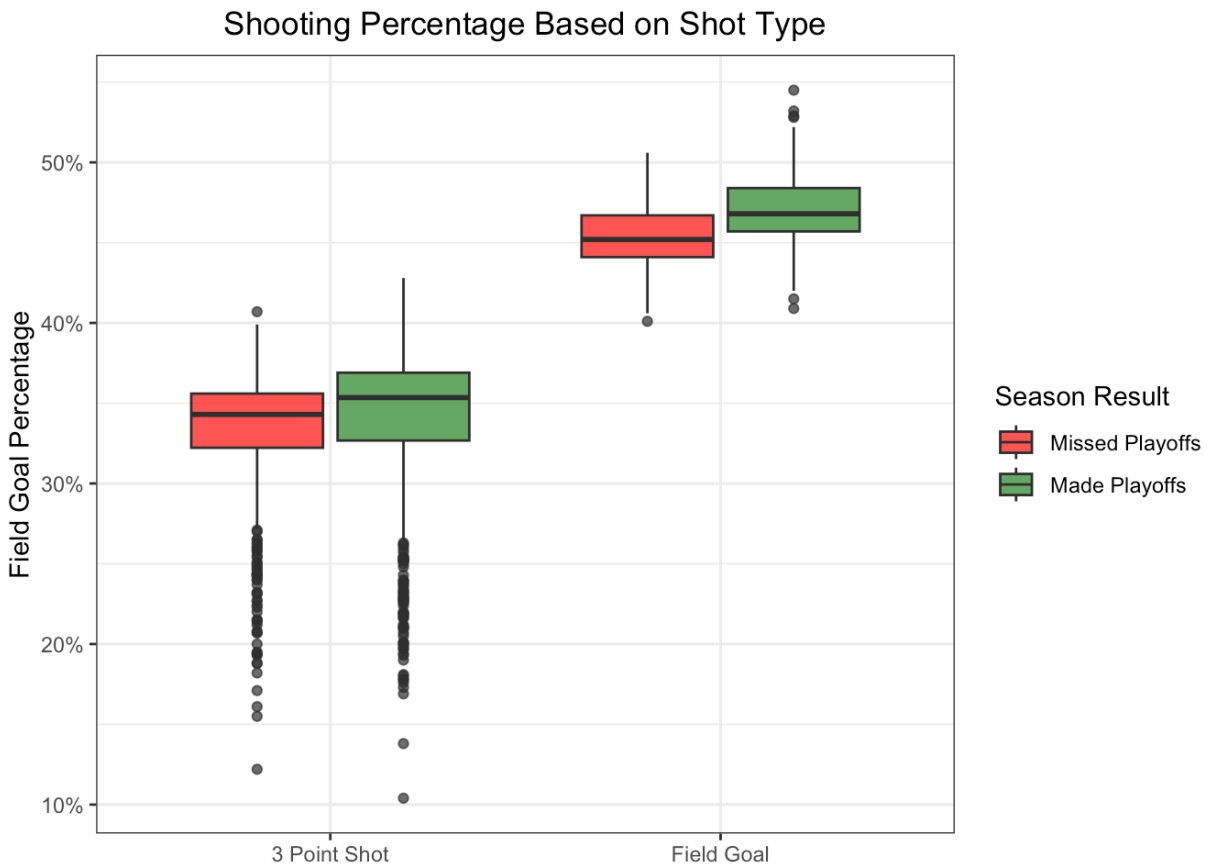


Before we dive deep into the statistics, we thought it would be a good idea to show a breakdown of the NBA Champions since 1980 by team. The Los Angeles Lakers have won 11 championships since 1980 which is almost double the next closest team, the Chicago Bulls with 6 championships. This is followed by the Boston Celtics and San Antonio Spurs who have both won 5 championships since 1980. Any team that has not won a championship is not shown. This doesn't really necessarily provide any insight to data analysis, but it can be helpful for context.

Shooting Percentage

```
# Data separating fg and 3 point shooting based on if team made playoffs
nba_shooting <- nba |>
  dplyr::select(team, abbreviation, playoffs, fg_percent, x3p_percent) |>
  pivot_longer(
    cols = c(fg_percent, x3p_percent),
    names_to = "shot_type",
    values_to = 'shot_percent'
  ) |>
  mutate(
    shot_type = if_else( condition = shot_type == "fg_percent",
                        true = "Field Goal",
                        false = "3 Point Shot"),
    playoffs = as.logical(playoffs)
  )

# Create boxplots of shooting percentages
ggplot(
  data = nba_shooting,
  mapping = aes(
    x = shot_type,
    y = shot_percent,
    fill = playoffs)) +
  geom_boxplot(alpha = .75) +
  labs(
    x = NULL,
    y = "Field Goal Percentage",
    title = "Shooting Percentage Based on Shot Type",
    fill = "Season Result"
  ) +
  scale_fill_manual(
    labels = c("Missed Playoffs", "Made Playoffs"),
    values = c("red", "forestgreen")) +
  scale_y_continuous(labels = scales::label_percent())
```



On average, teams are consistently more efficient in their overall field goal percentage than their 3 point field goal percentage. In the 3 point shot percentage data, there are large lower tails for both non playoff and playoff teams, indicating that it is not a very consistent measure and may not be good for predicting in our model. We will break down why this statistic has such a large variance in the next graph. In the field goal percentage box plots, there is a noticeable difference between playoff teams and non playoff teams. As expected, playoff teams shoot the ball slightly more efficiently and have more upper tail outliers than the non playoff teams.

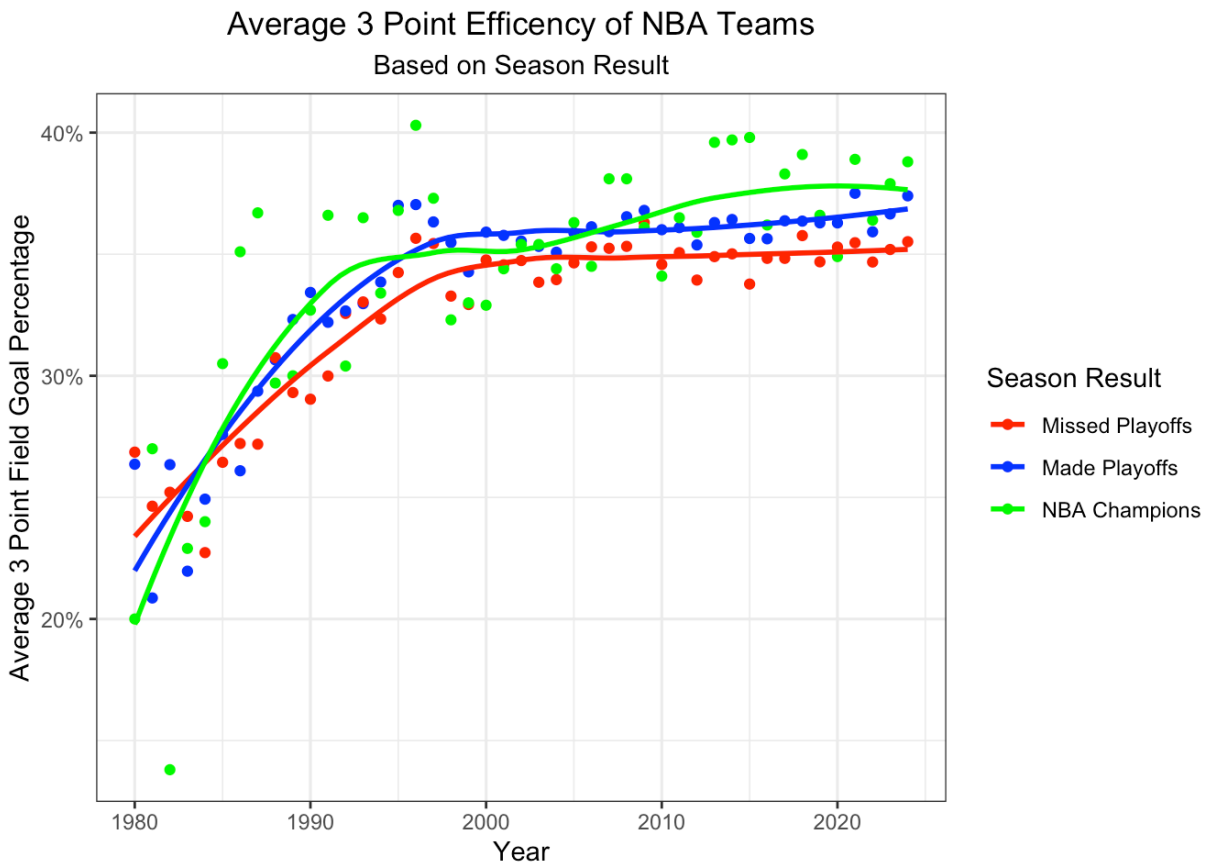
NOTE: Three point field goals are included in field goal percentage, however we are still interested in examining the statistic's predictive power on a team's success.

3 Point Percentage

```
# get avg 3 point fg percentage by season for each season_result
nba_3p <-
  nba |> dplyr::select(season, team, playoffs, champion, x3p_percent) |>
  mutate(
    season_result = playoffs + champion
  ) |>
  summarize(
    .by = c(season, season_result),
    avg_3p = mean(x3p_percent)
  )

#plot 3 point percentages of championship winning teams
x3p_championship <-
  ggplot(
    data = nba_3p,
    mapping = aes(
      x = season,
      y = avg_3p,
      group = factor(season_result),
      color = factor(season_result)
    )
  ) +
  geom_point() +
  # add the trend line to the graph
  geom_smooth(
    method = "loess",
    formula = y~x,
    se = F
  ) +
  #change labels
  labs(
    x = "Year",
    y = "Average 3 Point Field Goal Percentage",
    title = "Average 3 Point Efficiency of NBA Teams",
    subtitle = "Based on Season Result",
    color = "Season Result",
    group = NULL
  ) +
  #add percent to bar
  scale_y_continuous(labels = scales::label_percent() ) +
  #change labels in the legend
  scale_color_manual(
    labels = c('2' = "NBA Champions", '1' = "Made Playoffs", '0' = "Missed Playoffs"),
    values = c('red','blue','green'))

x3p_championship
```



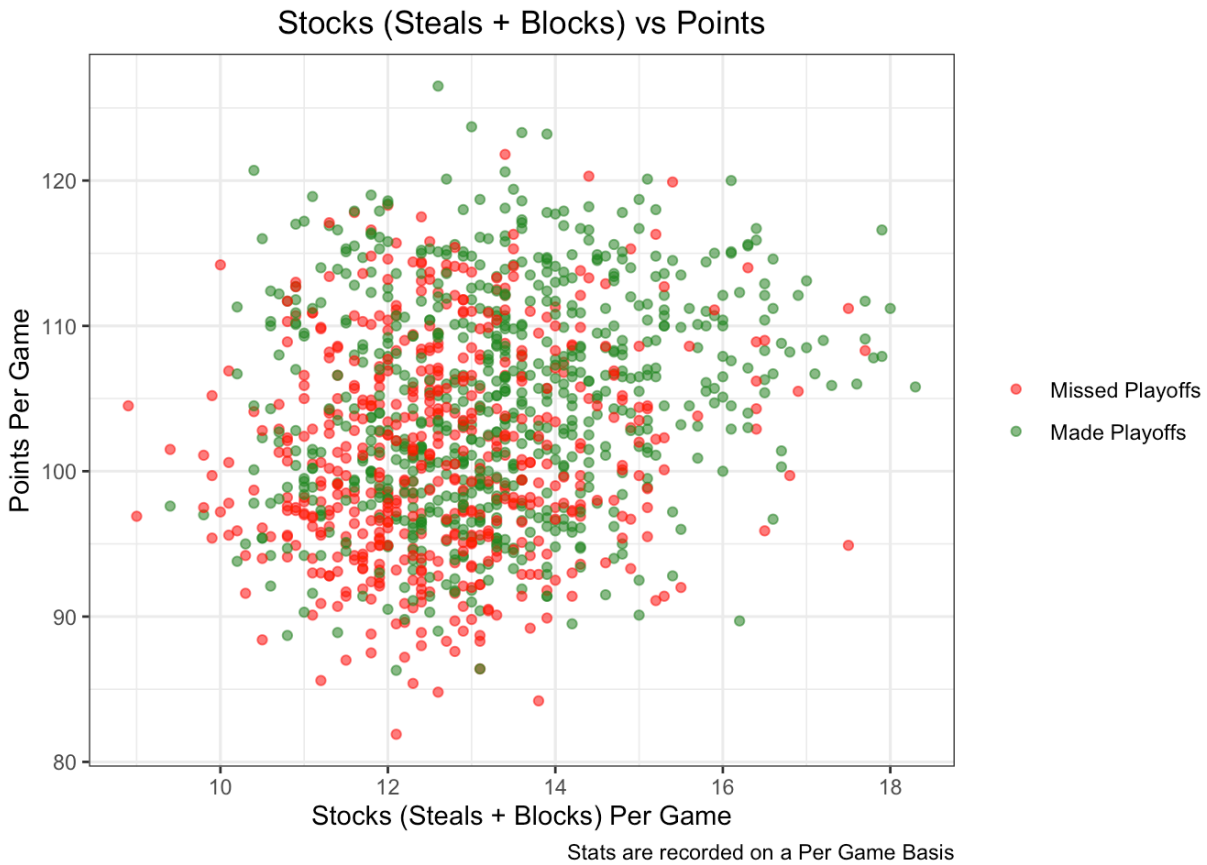
In this graph, we compare the 3 point percentages of every NBA Championship winning team since 1980. The graph shows an increase in percentage over time, which is explained by the context of NBA history. Since the 3 point line was added in 1979, teams in the 80's were not good at shooting threes yet. By the time we get to the 90's, winning teams have figured out how to shoot three's at a decent level. As we get to the 2010's the three point revolution occurs, and the best teams start shooting more three's at even higher efficiency. This causes the values in this graph to continue increasing into modern day, as teams like the Warriors and Celtics shot 3's at a very high level in their championship seasons.

Do Defense and Offense Correlate?

```
# Creating new data set with variables needed for graph
points_and_stocks <-
  nba |>
  dplyr::select(team, abbreviation, playoffs, champion, pts_per_game, stl_per_game, blk_per_game) |>
  mutate(stocks_per_game = stl_per_game + blk_per_game,
         playoffs = as.logical(playoffs))

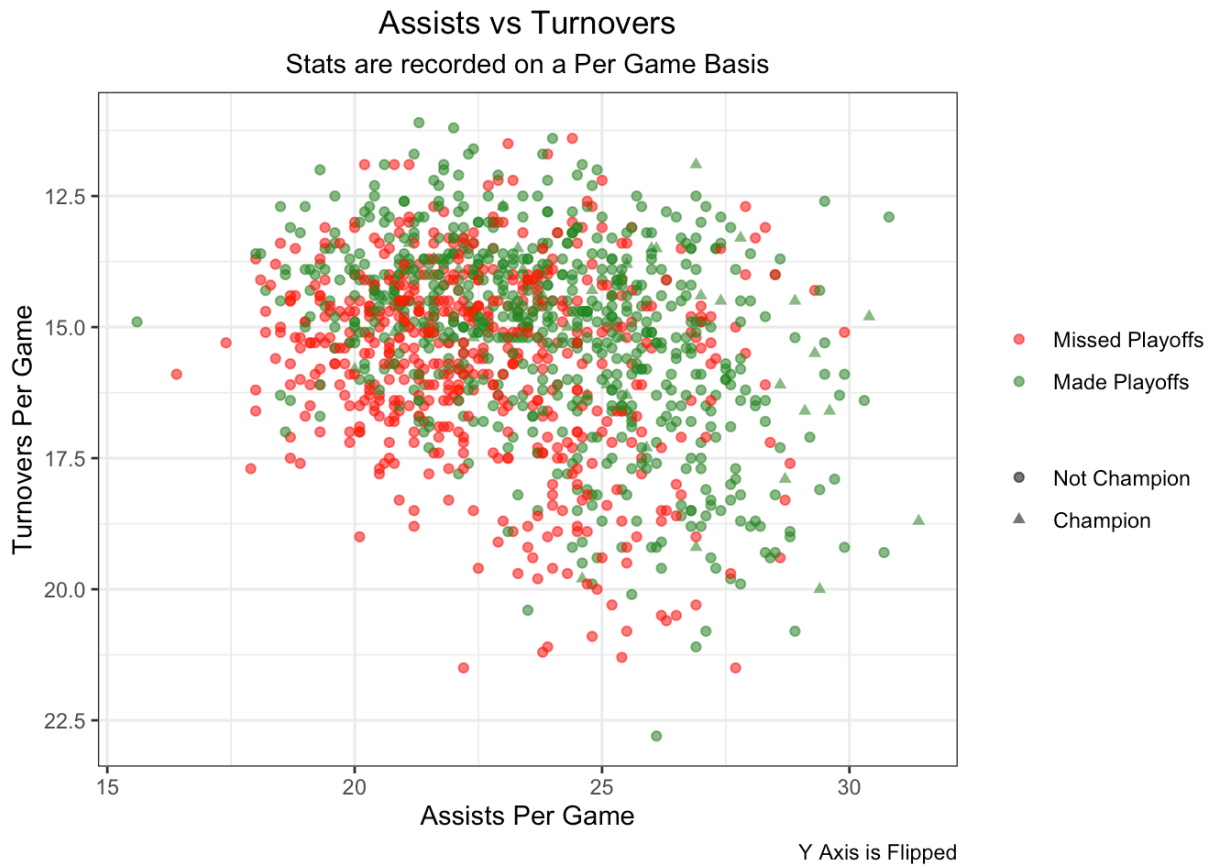
# Create scatterplot comparing points per game and stocks per game, showing if team made playoffs
gg_pts_vs_stocks <-
  ggplot(
    data = points_and_stocks,
    mapping = aes(
      x = stocks_per_game,
      y = pts_per_game
    )
  ) +
  geom_point(mapping = aes(
    color = factor(playoffs)),
    alpha = 0.6) +
  labs(
    x = "Stocks (Steals + Blocks) Per Game",
    y = "Points Per Game",
    title = "Stocks (Steals + Blocks) vs Points",
    caption = "Stats are recorded on a Per Game Basis",
    color = NULL
  ) +
  scale_color_manual(
    labels = c("FALSE" = "Missed Playoffs", "TRUE" = "Made Playoffs"),
    values = c("red", "forestgreen"))

gg_pts_vs_stocks
```



In this graph we compared the relationship between stocks per game, which is steals plus blocks, and points per game. We were looking to see if there was a relationship between those 2 variables and whether they have an impact on if a team makes the playoffs. From the graph, we can see most of the teams that had low stocks per game and low points per game missed the playoffs, while most of the teams with high stocks per game and points per game made the playoffs. This implies that having a really good offense (shown by more points per game) and having a really good defense (shown by more stocks per game) means a team is more likely to make the playoffs. This observation makes sense when it comes to the game of basketball.

```
to_v_ast <-  
  nba |>  
  dplyr::select(season, team, abbreviation, playoffs, champion, ast_per_game, tov_per_game) |>  
  mutate( playoffs = as.logical(playoffs),  
          champion = as.logical(champion) )  
  
gg_tov_ast <-  
  ggplot(  
    data = to_v_ast,  
    mapping = aes(  
      x = ast_per_game,  
      y = tov_per_game  
    )  
  ) +  
  #add points, change color if they made the playoffs, change shape if champions  
  geom_point(  
    mapping = aes(  
      color = playoffs,  
      shape = champion),  
    alpha = .6) +  
  #add labels  
  labs(  
    x = "Assists Per Game",  
    y = "Turnovers Per Game",  
    title = "Assists vs Turnovers",  
    subtitle = "Stats are recorded on a Per Game Basis",  
    color = NULL,  
    shape = NULL,  
    caption = "Y Axis is Flipped"  
  ) +  
  #change color and shape legend labels to reflect what they actually mean  
  scale_color_manual(  
    labels = c("FALSE" = "Missed Playoffs", "TRUE" = "Made Playoffs"),  
    values = c("red", "forestgreen")) +  
  scale_shape_manual(  
    labels = c("FALSE" = "Not Champion", "TRUE" = "Champion"),  
    values = c("circle", "triangle")  
  ) +  
  scale_y_reverse()  
  
gg_tov_ast
```



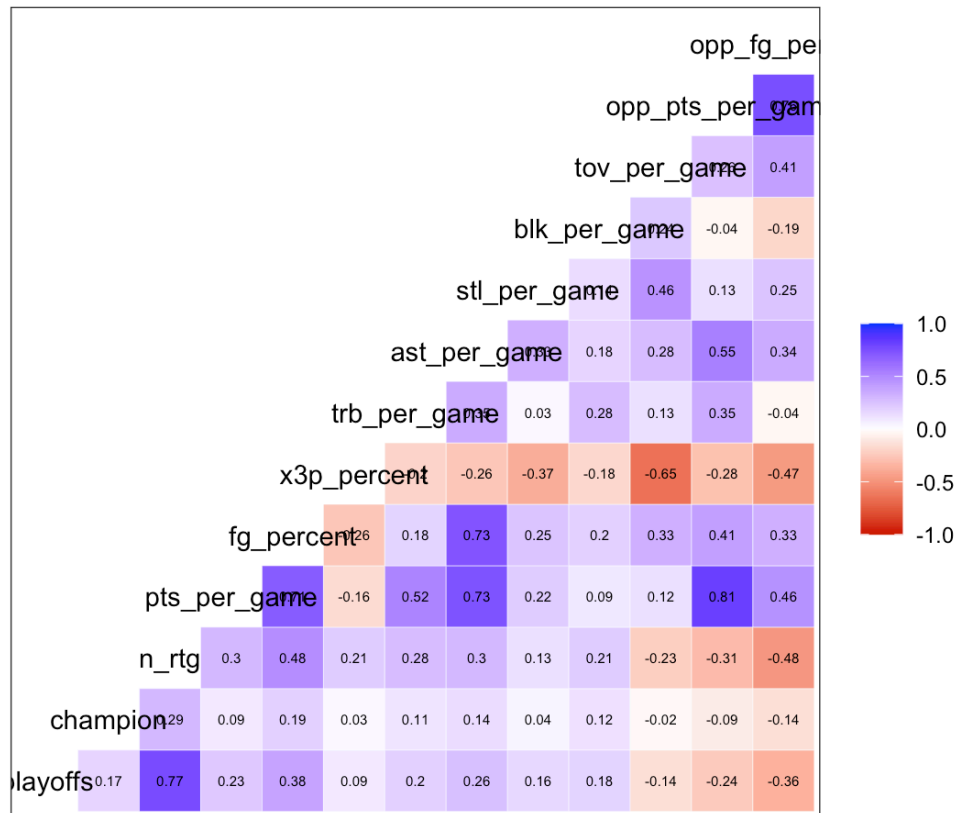
The assist to turnover ratio is a highly valued statistic among the basketball community. It is considered one of the best indicators of a good team and a bad one. We decided to take a look at these variables to see if there was any correlation between the stats. Additionally, we used the playoffs and champion classification variables to see if there were patterns between teams of each group. The results of the graph are pretty much what we expected to see. Most of the teams with high assist averages and low turnover averages made the playoffs. However, there are quite a few exceptions to this due to the large number of teams. Many of the teams on the far right side of this graph not only made the playoffs, but also won the championship.

Machine Learning

Before we create any models, we are going to examine a correlation plot to determine which predictors we would like to use, as well as any covariance between variables

```
nba |>
  dplyr::select(where(is.numeric) & !season) |>
  ggcorr(
    low = "red3", #low neg cor = red
    mid = "white",
    high = "blue", #high pos cor = blue
    label = T,
    label_size = 2,
    label_round = 2
  ) + labs( title = "Correlation Plot of All Possible Predictors")
```

Correlation Plot of All Possible Predictors



Examining this correlation plot, we are trying to find any variables that have a strong correlation with our champion variable.

kNN Regression

After examining the previous correlation plot, specifically different predictors' relationship with the champion row, we decided to use these 5 predictors for kNN:

- n_rtg (net rating)
- fg_percent (field goal percentage)
- ast_per_game (assists per game)
- stocks_per_game (steals + blocks per game)
- opp_fg_percentage (opponent field goal percentage)

Additionally, there is not any visible confounding effects between any of these variables.

Select Variables

```
#selecting the variables we want to use
nba_ml <- nba |>
  mutate(stocks_per_game = stl_per_game + blk_per_game) |>
  dplyr::select(season, team, abbreviation, playoffs, champion, n_rtg, fg_percent, ast_per_game,
    stocks_per_game, opp_fg_percent)
```

Setting up the data

```
#Normalize Function:
normalize <- function( x ){
  norm_x <- ( (x - min(x)) / ( max(x) - min(x) ))
  return( norm_x )
}

# Standardize function:
standardize <- function( x ){
  standard_x <- ( (x - mean(x)) / sd(x) )
  return( standard_x )
}

#normalizing our selected variables
nba_norm <- nba_ml |>
  mutate(
    across(
      .cols = c(n_rtg, fg_percent, ast_per_game, stocks_per_game, opp_fg_percent),
      .fns = normalize
    )
  )

#standardizing our selected variables
nba_stan <- nba_ml |>
  mutate(
    across(
      .cols = c(n_rtg, fg_percent, ast_per_game, stocks_per_game, opp_fg_percent),
      .fns = standardize
    )
  )

#skimr::skim(nba_norm)
#skimr::skim(nba_stan)
```

A Small Problem

```
#running with 10 as an example
example_problem_norm <- knn.cv(
  train = nba_norm |> dplyr::select(n_rtg:opp_fg_percent),
  # dont need test because LOO CV
  cl = nba$champion,
  k = 10
)
tibble(example_problem_norm)
```



```
## # A tibble: 1,254 × 1
##   example_problem_norm
##   <fct>
## 1 0
## 2 0
## 3 0
## 4 0
## 5 0
## 6 0
## 7 0
## 8 0
## 9 0
## 10 0
## # i 1,244 more rows
```

We are trying to predict an NBA Champion using kNN classification, however this results in a problem. kNN classification works by looking at the k nearest data points, 10 in this case, and making a guess based on those 10 points. In our case, these points represent teams. However, most points in our data are not classified as NBA champions only 1 out of every 30 teams is a champion. This results in every team being predicted as a non-champion because every team must be classified as either a champion or non-champion.

Instead, we are going to use kNN regression, predicting the champion variable as a numerical variable and not a classification variable. Instead of predicting each team's championship column as a 0 or 1, this method will result in a decimal average between 0 and 1 of the k nearest teams championship column. We are going to refer to this number as "championship score". Grouping by the year, the team with the highest championship score will be considered our predicted champion.

Due to these restraints and the circumstances of predicting an NBA Champion, we are going to change the way we view the accuracy of the model. The model should be able to predict most non-champions, so we are going to base the accuracy on predicting the correct champion for each year. Our best model will be the one that predicts the most correct champions using the last 44 years of data. We will also provide an R2 value for reference.

Grid Search

```

#set up vector to record how many correct champions each model predicted
knn_championships_correct <-
  data.frame(
    k = 3:200,
    normalized = -1,
    standardized = -1
  )

#set up vector to record R2 values
r2_knn <- data.frame(
  k = 3:200,
  norm = -1,
  stan = -1
)

for (i in 1:nrow(r2_knn)){

  # kNN for norm data
  loop_pred_norm <- knn.reg(
    train = nba_norm |> dplyr::select(n_rtg:opp_fg_percent),
    # if we do not give it a test vector, it does LOOCV by default
    y = nba_norm$champion,
    k = r2_knn$k[i]
  )

  #create a df with our predicted champions
  loop_pred_champs_norm <- nba_norm |>
    #add the predicted champ_score variable to each team
    mutate(
      champ_score = loop_pred_norm$pred
    ) |>
    #get the max for every season
    slice_max(
      by = season,
      order_by = champ_score,
      n = 1,
      # we are allowing ties in the case that there is one, there is no good way of breaking a t
      ie
      # if two teams have the same predicted champ_score
      with_ties = T
    )

  #correct champion prediction count for norm
  loop_champs_pred_norm <- sum(loop_pred_champs_norm$champion)
  knn_championships_correct[i, 2] <- loop_champs_pred_norm
  #r2 for norm
  r2_knn[i, 'norm'] <- loop_pred_norm$R2Pred

  # kNN for stan data
  loop_pred_stan <- knn.reg(
    train = nba_stan |> dplyr::select(n_rtg:opp_fg_percent),
    # if we do not give it a test vector, it does LOOCV by default
    y = nba_stan$champion,
    k = r2_knn$k[i]
  )
}

```

```

)

#create a df with our predicted champions
loop_pred_champs_stan <- nba_stan |>
  #add the predicted champ_score variable to each team
  mutate(
    champ_score = loop_pred_stan$pred
  ) |>
  #get the max for every season
  slice_max(
    by = season,
    order_by = champ_score,
    n = 1,
    # we are allowing ties in the case that there is one, there is no good way
    # of breaking a tie if two teams have the same predicted champ_score
    with_ties = T
  )

#correct champion prediction count for stan
loop_champs_pred_stan <- sum(loop_pred_champs_stan$champion)
knn_championships_correct[i, 3] <- loop_champs_pred_stan
#r2 for stan
r2_knn[i, 'stan'] <- loop_pred_stan$R2Pred
}

#show the correct champions df
head(knn_championships_correct)

```

```

##    k normalized standardized
## 1 3      24      22
## 2 4      27      26
## 3 5      23      23
## 4 6      20      22
## 5 7      19      18
## 6 8      16      15

```

```

#show the r2 df
head(r2_knn)

```

```

##    k      norm      stan
## 1 3 0.001194743 -0.01673253
## 2 4 0.047773642  0.03913014
## 3 5 0.076181969  0.05774249
## 4 6 0.082507735  0.09403241
## 5 7 0.104368033  0.11142396
## 6 8 0.108278191  0.11151951

```

Here you can see the first 6 rows of the data frames that are being used to access the accuracy of the model.

Finding the best model

```
#find max championships predicted
knn_championships_correct |>
  pivot_longer(
    cols = - k,
    names_to = "model",
    values_to = "correct_predictions"
  ) |>
  slice_max(
    by = model,
    order_by = correct_predictions
  )
```

```
## # A tibble: 2 × 3
##       k model      correct_predictions
##   <int> <chr>          <dbl>
## 1     4 normalized          27
## 2     4 standardized        26
```

```
#show the top predicted R2
r2_knn |> filter( k == 4)
```

```
##   k      norm      stan
## 1 4 0.04777364 0.03913014
```

```
#find max R2
r2_knn |>
  pivot_longer(
    cols = - k,
    names_to = "model",
    values_to = "r2"
  ) |>
  slice_max(
    order_by = r2
  )
```

```
## # A tibble: 1 × 3
##       k model    r2
##   <int> <chr> <dbl>
## 1    13 stan  0.124
```

Using $k = 4$ in our model produced the most number of correct champions predicted. Normalizing the data allowed us to predict 1 additional champion compared to standardizing the data while using $k = 4$. The best model predicted the correct champion for 27 out of 44 seasons, which is 61.36% accuracy. This is pretty impressive given that the no information rate (blindly guessing) of guessing an NBA champion in any given season is about $1/30$, or 3.33%, since there are 30 teams competing for the NBA championship.

However, when examining the R^2 values, we see that the models with $k = 4$ are not our most accurate overall. The best model in terms of R^2 is produced when we standardize the data and use $k = 13$. If we were more interested in the predicting a team's record or the NBA standings, it may be a better idea to use this model since it is more accurate for every team.

For this project we are most concerned about predicting NBA champions, so we consider the model with normalized data and $k = 4$ to be the best kNN model for predicting NBA champions based on `n_rtg`, `fg_percent`, `ast_per_game`, `stocks_per_game`, `opp_fg_percentage`.

Side Note: As mentioned prior, the accuracy of these models is incredibly low with the best one having an R^2 of about 0.13. This would be borderline unacceptable for most models, however when it comes to predicting a champion of a profession sports league, the odds of being correct are very low.

Classification Tree

```
# set seed
RNGversion("4.1.0")
set.seed(2870)

# create stocks per game variable (steals + blocks)
nba_class_tree <-
  nba |>
  mutate(stocks_per_game = stl_per_game + blk_per_game)

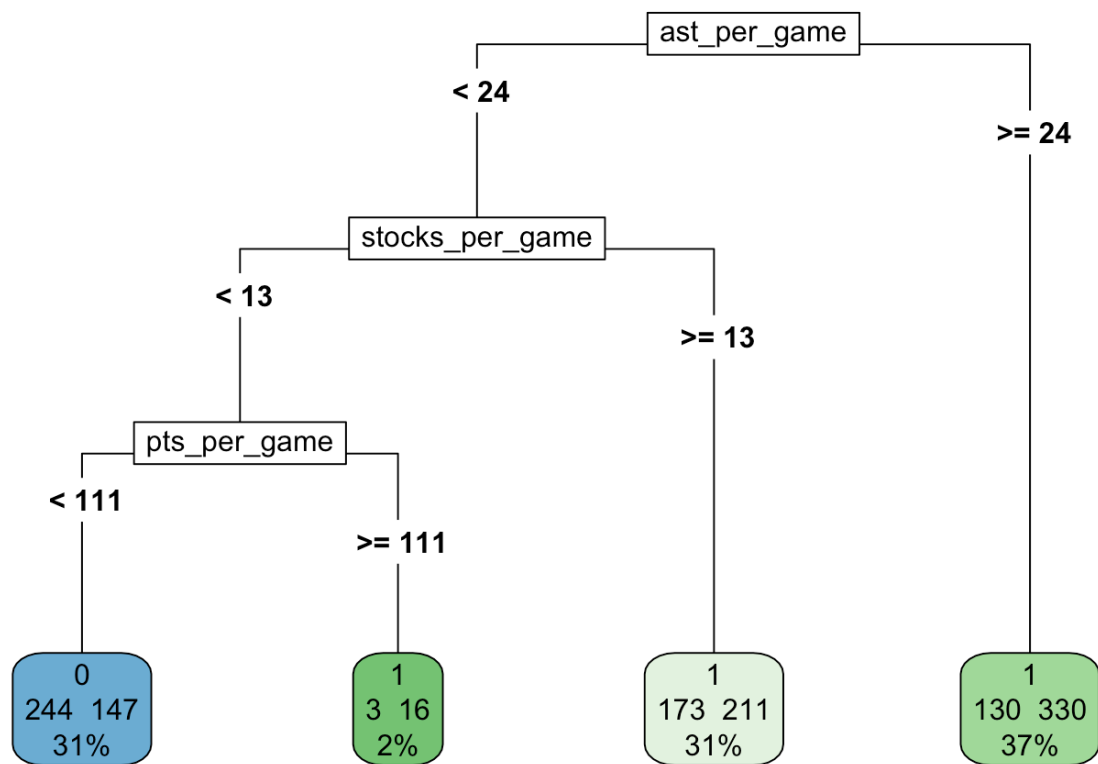
# Create the full classification tree
nba_full <-
  rpart(
    formula = playoffs ~ pts_per_game + ast_per_game + stocks_per_game,
    data = nba_class_tree,
    method = "class",
    parms = list(split = "information"),
    minsplit = 0,
    minbucket = 0,
    cp = -1
  )

# Find the xerror cut off: min(xerror) + xstd
xcutoff <-
  data.frame(nba_full$sctestable) %>%
  slice_min(order_by = xerror,
            n = 1,
            with_ties = F) %>%
  mutate(xcut = xerror + xstd) %>%
  pull(xcut)

cp_prune <-
  data.frame(nba_full$sctestable) %>%
  filter(xerror < xcutoff) %>%
  slice(1) %>%
  pull(CP)

# create pruned tree
nba_prune <-
  prune(
    tree = nba_full,
    cp = cp_prune
  )

# create plot
rpart.plot(
  x = nba_prune,
  type = 5,
  extra = 101
)
```



```

# use cross validation to predict accuracy
nba_predict <-
  predict(
    object = nba_prune,
    type = "class",
    newdata = nba_class_tree
  )

confusionMatrix(
  data = nba_predict,
  reference = factor(nba_class_tree$playoffs),
  positive = "1"
)

```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0    1
##           0 244 147
##           1 306 557
##
##           Accuracy : 0.6388
##           95% CI : (0.6115, 0.6654)
##           No Information Rate : 0.5614
##           P-Value [Acc > NIR] : 1.535e-08
##
##           Kappa : 0.2425
##
## Mcnemar's Test P-Value : 1.141e-13
##
##           Sensitivity : 0.7912
##           Specificity : 0.4436
##           Pos Pred Value : 0.6454
##           Neg Pred Value : 0.6240
##           Prevalence : 0.5614
##           Detection Rate : 0.4442
##           Detection Prevalence : 0.6882
##           Balanced Accuracy : 0.6174
##
##           'Positive' Class : 1
##
```

```
# determining variable importance
varImp(nba_prune)
```

```
##           Overall
## ast_per_game    44.28594
## pts_per_game    39.11472
## stocks_per_game 43.91603
```

We used a decision tree because it gives us a visual display of our prediction model. Since our target variable is categorical (playoffs), we will use a classification tree.

While we originally intended to champion as our response variable, the lack of champions in our data (1 per year) meant that the tree did not compile fully. Therefore, we switched to playoffs, as a way to measure a team's success. Our explanatory variables used are points, assists, and stocks (steals + blocks) per game. The variables net rating and field goal percentage were too strong of predictors, and were therefore not included in this model.

Our results show that, in order to be predicted as a playoff team in the NBA, your team must achieve at least one of the following: 24 assists per game, 13 stocks per game, or 111 points per game. If a team fails to achieve at least one of these criteria, our model will not predict them as a playoff team.

Our model has 63.88 percent accuracy, and is shown to be a significantly better predictor of a NBA team making the playoffs than the no information rate. Variable importance shows that assists per game and stocks per game are more important predictors than points per game. This shows that defense and offensive efficiency may be just as, if not more, important than raw points totals.

Using our models to predict a 2025 NBA Champions

Although not required, we thought it would be fun to use our models and the existing data for the 2025 NBA season. Since the 2025 season will be less than halfway done our prediction may not be very accurate come playoffs, but it should make a reasonable team.

Load in 2025 Data

```

#load in data and take out the incomplete 2025 season and seasons before 1980
team_per_game25 <- read.csv("Team Stats Per Game25.csv") |>
  filter(season == 2025,
         # remove league average rows
         team != "League Average") |>
# used the champions dataframe to add 1s for all champions in the column
left_join(
  y = champions,
  by = c('season', 'team')
) |>
mutate(
  #change playoffs to be 0 and 1 not F and T
  playoffs = as.numeric(playoffs),
  #manually add a championship column
  champion = if_else( is.na(champion), true = 0, false = 1)
) |>
#select some columns we may want to use
dplyr::select(season, team, abbreviation, playoffs, champion, fg_percent, x3p_percent, orb_per_game:pts_per_game, -pf_per_game)

opp_per_game25 <- read.csv("Opponent Stats Per Game25.csv") |>
#take out the incomplete 2025 season and seasons before 1980
filter(season == 2025,
       team != "League Average") |>
#select some columns we may want to use
dplyr::select(season, team, opp_fg_percent, opp_x3p_percent, opp_trb_per_game, opp_ast_per_game, opp_tov_per_game, opp_pts_per_game)

team_summaries25 <- read.csv('Team Summaries25.csv') |>
#take out the incomplete 2025 season and seasons before 1980
filter(season == 2025,
       team != "League Average") |>
#select some columns we may want to use
dplyr::select(season, team, o_rtg, d_rtg, n_rtg, ts_percent, e_fg_percent, opp_e_fg_percent)

#join all 3 data frames into one df

nba25 <- team_per_game25 |>
  left_join( y = opp_per_game25,
            by = c('season', 'team')) |>
  left_join( y = team_summaries25,
            by = c('season', 'team'))|>
# final decision on what to keep in nba dataset
dplyr::select(season, team, abbreviation, playoffs, champion, n_rtg, pts_per_game, fg_percent, x3p_percent, trb_per_game, ast_per_game, stl_per_game, blk_per_game, tov_per_game, opp_pts_per_game, opp_fg_percent)

#remove the old dataframes
rm(team_per_game25, opp_per_game25)

#skim to confirm all variables are full
skimr::skim(nba25)

```













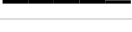

Data summary

Name	nba25
Number of rows	30
Number of columns	16
Column type frequency:	
character	2
numeric	14
Group variables	
None	

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
team	0	1	9	22	0	30	0
abbreviation	0	1	3	3	0	30	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
season	0	1	2025.00	0.00	2025.00	2025.00	2025.00	2025.00	2025.00	
playoffs	0	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
champion	0	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
n_rtg	0	1	-0.02	6.67	-14.70	-4.18	0.30	5.62	10.70	
pts_per_game	0	1	112.93	4.82	103.80	108.90	113.25	116.18	122.40	
fg_percent	0	1	0.46	0.02	0.42	0.45	0.46	0.48	0.51	
x3p_percent	0	1	0.36	0.02	0.32	0.34	0.36	0.38	0.40	
trb_per_game	0	1	43.89	2.62	38.30	42.23	43.65	45.50	50.00	
ast_per_game	0	1	26.02	2.45	21.20	24.35	25.55	27.85	30.40	
stl_per_game	0	1	8.29	1.21	5.90	7.40	8.10	9.17	11.90	
blk_per_game	0	1	5.20	0.91	3.60	4.62	5.10	5.57	7.30	
tov_per_game	0	1	14.74	1.66	11.60	13.48	14.60	16.17	18.20	
opp_pts_per_game	0	1	112.93	4.97	102.80	110.35	112.55	115.68	124.20	
opp_fg_percent	0	1	0.46	0.02	0.42	0.46	0.46	0.48	0.49	

```
#create df for mach learning
nba25_ml <- nba25 |>
  mutate(stocks_per_game = stl_per_game + blk_per_game) |>
  dplyr::select(season, team, abbreviation, playoffs, champion, n_rtg, fg_percent, ast_per_game,
stocks_per_game, opp_fg_percent)
head(nba25_ml)
```

```
##   season      team abbreviation playoffs  champion  n_rtg  fg_percent
## 1  2025   Atlanta Hawks         ATL      0        0   -3.1    0.463
## 2  2025   Boston Celtics         BOS      0        0   10.7    0.464
## 3  2025   Brooklyn Nets         BRK      0        0   -3.1    0.468
## 4  2025   Chicago Bulls         CHI      0        0   -5.5    0.475
## 5  2025  Charlotte Hornets         CHO      0        0   -4.6    0.424
## 6  2025  Cleveland Cavaliers         CLE      0        0    9.8    0.511
##   ast_per_game  stocks_per_game  opp_fg_percent
## 1          29.8           15.5         0.465
## 2          25.9           12.6         0.467
## 3          26.4           10.1         0.486
## 4          28.7           11.6         0.488
## 5          22.9           13.3         0.467
## 6          28.2           14.2         0.460
```

kNN Prediction

```
nba25_norm <- nba25_ml |>
  mutate(
    across(
      .cols = c(n_rtg, fg_percent, ast_per_game, stocks_per_game, opp_fg_percent),
      .fns = normalize
    )
  )

nba25_knn <- knn.reg(
  train = nba_norm |> select(n_rtg:opp_fg_percent),
  test = nba25_norm |> select(n_rtg:opp_fg_percent),
  y = nba_stan$champion,
  k = 4
)

pred_champ25 <- nba25 |>
  mutate(
    champ_score = nba25_knn$pred
  ) |>
  slice_max(
    order_by = champ_score
  ) |> dplyr::select( team, abbreviation)

pred_champ25
```

```
##           team abbreviation
## 1 Cleveland Cavaliers      CLE
```

Our kNN model predicts that the Cleveland Cavaliers will win the 2025 NBA Championship. If you are up to date with the NBA, this makes sense. The Cavaliers had one of the best starts to a season in NBA history, winning their first 15 games. As of 12/9/2024, the Cleveland Cavaliers sit at the top of the NBA's power rankings (Source: <https://www.nba.com/news/power-rankings-2024-25-week-8> (<https://www.nba.com/news/power-rankings-2024-25-week-8>)).

Conclusions

From our graphs in the data visualization section we can see that teams shoot 3 pointers better now than they did in the 1980's, and that the teams that shoot the best 3 point percentage tend to have more success in that season. We can also see that teams that have more assists per game tend to have more success and make the playoffs and win championships more.

For the kNN regression section, $k = 4$ predicted the most correct champions at 27/44, however $k = 13$ would be the best value of k for predicting wins or NBA standings as it has the highest R^2 value. Using the model with $k = 4$, we predict that the 2025 NBA Champion the Cleveland Cavaliers will be the 2025 NBA champions.

From the classification tree section, in order to be predicted as a playoff team in the NBA, your team must achieve at least one of the following: 24 assists per game, 13 steals per game, or 111 points per game. If a team fails to achieve at least one of these criteria, our model will not predict them as a playoff team. Both of these machine learning methods produce results that are better than the no information rate.

After completing our machine learning, we conclude that it is possible to predict a team's success using basic NBA statistics. Our classification model was about 63% accurate at predicting playoff teams and on top of that our kNN model predicted more than half of the NBA champions from the past 44 years correctly. Both of these are well above the no information guess rate.

Limitations

One limitation from our study is that we used data that is a summary of results, which can cause some predictive power to be lost since some teams may regress towards the mean. The data also doesn't factor in other effects such as coaching, playoff experience, home court advantage, injuries, etc which can have major implications on a team's season and playoff success.

Another limitation on our 2025 prediction is that the 2024-2025 season won't be complete, which may make our prediction inaccurate, even if the prediction that we got does seem reasonable right now.

However, the main limitation to this machine learning process is the difficulty of predicting a champion in general. Only 1 team in the NBA can finish as the champion each season, meaning that the other 29 are all marked as non-champions. This makes machine learning very difficult because championship teams are heavily outnumbered meaning most techniques will result in over predicting non-champions due to their prevalence. This issue was seen in our kNN section where we had to change from kNN classification to kNN regression. One solution to this limitation in a future study could be creating a variable that is better at measuring a team's season outcome. For example, a team that makes the conference finals would have a different factor level than a team that completely missed the playoffs.

An idea for future research would be to try using different variables that could be used to predict champions and see whether those might work better overall, not just at this part of the season. Another idea for future research could be to try to predict total wins or NBA standings for a given season, instead of the champion.