DATA621 Final Project, Does Tanking Work in the NBA?

Michael O’Donnell

December 8, 2020

## Load data and required libraries

# load required packages  
library(ggplot2)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(corrplot)

## corrplot 0.84 loaded

library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(caret)

## Loading required package: lattice

library(haven)  
library(QuantPsyc)

## Loading required package: boot

##   
## Attaching package: 'boot'

## The following object is masked from 'package:lattice':  
##   
## melanoma

##   
## Attaching package: 'QuantPsyc'

## The following object is masked from 'package:base':  
##   
## norm

# Loading the data  
git\_dir <- 'https://raw.githubusercontent.com/odonnell31/NBA-Team-Strategies/main/data'  
df = read.csv(paste(git\_dir, "/nba\_teams\_data\_1990\_2020.csv", sep=""))

## Exploratory data analysis

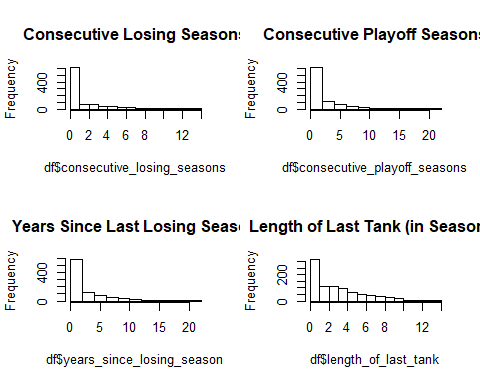
See a summary of each variable

summary(df)

## Year Team W L   
## Min. :1990 Atlanta Hawks : 31 Min. : 7.00 Min. : 9.00   
## 1st Qu.:1998 Boston Celtics : 31 1st Qu.:30.00 1st Qu.:30.00   
## Median :2005 Brooklyn Nets : 31 Median :41.00 Median :39.00   
## Mean :2005 Chicago Bulls : 31 Mean :40.03 Mean :40.03   
## 3rd Qu.:2013 Cleveland Cavaliers: 31 3rd Qu.:50.00 3rd Qu.:49.00   
## Max. :2020 Dallas Mavericks : 31 Max. :73.00 Max. :72.00   
## (Other) :717   
## W.L. PS.G PA.G SRS   
## Min. :0.1060 Min. : 81.9 Min. : 83.4 Min. :-14.680000   
## 1st Qu.:0.3780 1st Qu.: 95.8 1st Qu.: 95.9 1st Qu.: -3.175000   
## Median :0.5120 Median : 99.7 Median :100.2 Median : 0.170000   
## Mean :0.4998 Mean :100.4 Mean :100.4 Mean : -0.005637   
## 3rd Qu.:0.6200 3rd Qu.:104.3 3rd Qu.:104.7 3rd Qu.: 3.285000   
## Max. :0.8900 Max. :119.9 Max. :130.8 Max. : 11.800000   
##   
## Playoffs Losing\_season Finals\_Team   
## Min. :0.0000 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.00000   
## Median :1.0000 Median :0.0000 Median :0.00000   
## Mean :0.5493 Mean :0.4374 Mean :0.06866   
## 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.00000   
## Max. :1.0000 Max. :1.0000 Max. :1.00000   
##   
## consecutive\_losing\_seasons consecutive\_playoff\_seasons  
## Min. : 0.000 Min. : 0.00   
## 1st Qu.: 0.000 1st Qu.: 0.00   
## Median : 0.000 Median : 1.00   
## Mean : 1.493 Mean : 2.19   
## 3rd Qu.: 2.000 3rd Qu.: 3.00   
## Max. :14.000 Max. :22.00   
##   
## years\_since\_losing\_season length\_of\_last\_tank  
## Min. : 0.000 Min. : 0.000   
## 1st Qu.: 0.000 1st Qu.: 1.000   
## Median : 1.000 Median : 3.000   
## Mean : 2.533 Mean : 3.446   
## 3rd Qu.: 4.000 3rd Qu.: 5.000   
## Max. :22.000 Max. :14.000   
##

Look at histograms of important predictors

# setup 4 plots  
par(mfrow=c(2,2))  
  
# plot a histogram for each of the predictor variables  
hist(df$consecutive\_losing\_seasons, main = "Consecutive Losing Seasons")  
hist(df$consecutive\_playoff\_seasons, main = "Consecutive Playoff Seasons")  
hist(df$years\_since\_losing\_season, main = "Years Since Last Losing Season")  
hist(df$length\_of\_last\_tank, main = "Length of Last Tank (in Seasons)")



Subset the data for only possible predictors and response

keep\_vars <- c("Year", "Team", "Playoffs",  
 "Losing\_season", "Finals\_Team",  
 "consecutive\_losing\_seasons", "consecutive\_playoff\_seasons",  
 "years\_since\_losing\_season", "length\_of\_last\_tank")  
df <- df[keep\_vars]

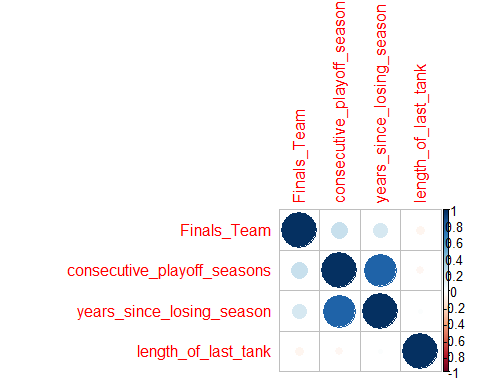
Check for NA’s

has\_NA = names(which(sapply(df, anyNA)))  
has\_NA

## character(0)

Explore correlations among important predictors

# look at correlations   
corr\_vars = c("Finals\_Team","consecutive\_playoff\_seasons",  
 "years\_since\_losing\_season", "length\_of\_last\_tank")  
cor\_train = cor(df[corr\_vars], use = "na.or.complete")  
corrplot(cor\_train)



## Creating Regression Model

Create a binary logicstic regression model with Finals\_Team as the response

# create Binary Logistic Regression model  
finals\_logistic\_model <- glm(Finals\_Team ~ consecutive\_playoff\_seasons +  
 length\_of\_last\_tank +  
 years\_since\_losing\_season,  
 data = df, family = binomial())  
  
summary(finals\_logistic\_model)

##   
## Call:  
## glm(formula = Finals\_Team ~ consecutive\_playoff\_seasons + length\_of\_last\_tank +   
## years\_since\_losing\_season, family = binomial(), data = df)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.5069 -0.3666 -0.3118 -0.2837 2.5271   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.934880 0.243357 -12.060 < 2e-16 \*\*\*  
## consecutive\_playoff\_seasons 0.164200 0.062332 2.634 0.00843 \*\*   
## length\_of\_last\_tank -0.064449 0.050723 -1.271 0.20387   
## years\_since\_losing\_season 0.006124 0.060776 0.101 0.91974   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 451.79 on 902 degrees of freedom  
## Residual deviance: 418.33 on 899 degrees of freedom  
## AIC: 426.33  
##   
## Number of Fisher Scoring iterations: 5

Calculate Odd Ratios based on Regression Coefficients

#Logistic Regression Coefficient  
finals\_summary.coeff0 = summary(finals\_logistic\_model)$coefficient  
  
#Calculating Odd Ratios  
FinalsOddRatio = exp(coef(finals\_logistic\_model))  
finals\_summary.coeff = cbind(Variable = row.names(finals\_summary.coeff0), FinalsOddRatio, finals\_summary.coeff0)  
row.names(finals\_summary.coeff0) = NULL

Create a function to standardize the regression coefficients

# function to standardize regression coefficients  
standardize\_coefficients <- function (bl\_model)   
{ b <- summary(bl\_model)$coef[-1,1]  
 sx <- sapply(bl\_model$model[-1], sd)  
 beta <-(3^(1/2))/pi \* sx \* b  
 return(beta)  
}

Create standardized regression coefficients with new function

# use above function to standardize regression coefficients from model  
std\_Coeff = data.frame(Standardized.Coeff = standardize\_coefficients(finals\_logistic\_model))  
std\_Coeff = cbind(Variable = row.names(std\_Coeff), std\_Coeff)  
row.names(std\_Coeff) = NULL

Merge the Odds Ratios and Coefficients to see all results

#Final Summary Report  
final\_report = merge(finals\_summary.coeff, std\_Coeff, by = "Variable", all.x = TRUE)  
  
final\_report

## Variable FinalsOddRatio Estimate  
## 1 (Intercept) 0.0531370984630938 -2.93487994188953  
## 2 consecutive\_playoff\_seasons 1.1784498693266 0.164199904794426  
## 3 length\_of\_last\_tank 0.93758354070881 -0.0644494150182418  
## 4 years\_since\_losing\_season 1.00614247843751 0.00612369031467352  
## Std. Error z value Pr(>|z|)  
## 1 0.243357195570939 -12.0599677975579 1.71850093581279e-33  
## 2 0.0623320180371948 2.63427865750222 0.00843162580819393  
## 3 0.0507231386563655 -1.27061173116411 0.203866811448084  
## 4 0.0607762041056168 0.100758025361896 0.919742548115072  
## Standardized.Coeff  
## 1 NA  
## 2 0.29170226  
## 3 -0.10264590  
## 4 0.01208479