IDA 5103 Group Project Initial Draft

Creighton DeKalb (cdekalb@ou.edu), Kevin Oberlag (koberlag@ou.edu), Mandy Chan (mandychan@ou.edu)

November 26, 2019

```
library(tidyverse)
library(caret)
```

Introduction to the problem

The problem we want to examine is the prediction of an NBA Finals berth for all 2019-2020 NBA teams.

Description of the data

Our data is the total season stats for every NBA team since the implementation of the 3-point line (the 1979-1980 season). We chose this because it gives us a sufficient amount of data, while also minimizing missingness, namely since the 3-point stats for each team before the 1979-1980 are NA's. Our intial data only contained in-game statistics and did not give us any data for the team from an overal season perspective; thus through additional research, we added columns pertaining to whether a given team for a given season won the championship. We also added a column noting whether a given team made the NBA finals for that season.

Exploratory data analysis

Preprocessing

```
# read in teams data
teams <- read_csv("teams.csv")

# change variable type for the factor variables
teams$Champion <- as.logical(teams$Champion)
teams$Finals <- as.logical(teams$Finals)
teams$Champion <- ifelse(teams$Champion == 1, "Yes", "No")
teams$Finals <- ifelse(teams$Finals == 1, "Yes", "No")

# add features that might be relevant by finding shot percentages and per game stats
teams$`FG%` <- teams$FG / teams$FGA
teams$`2P%` <- teams$`2P` / teams$`2PA`
teams$`3P%` <- teams$`3P` / teams$`3PA`
teams$`FT%` <- teams$FT / teams$FTA
teams$TRBperG <- teams$TRB / teams$G
teams$ASTperG <- teams$AST / teams$G
teams$STLperG <- teams$STL / teams$G
teams$BLKperG <- teams$BLK / teams$G
teams$PTSperG <- teams$PTS / teams$G
```

```r
head(teams)
```

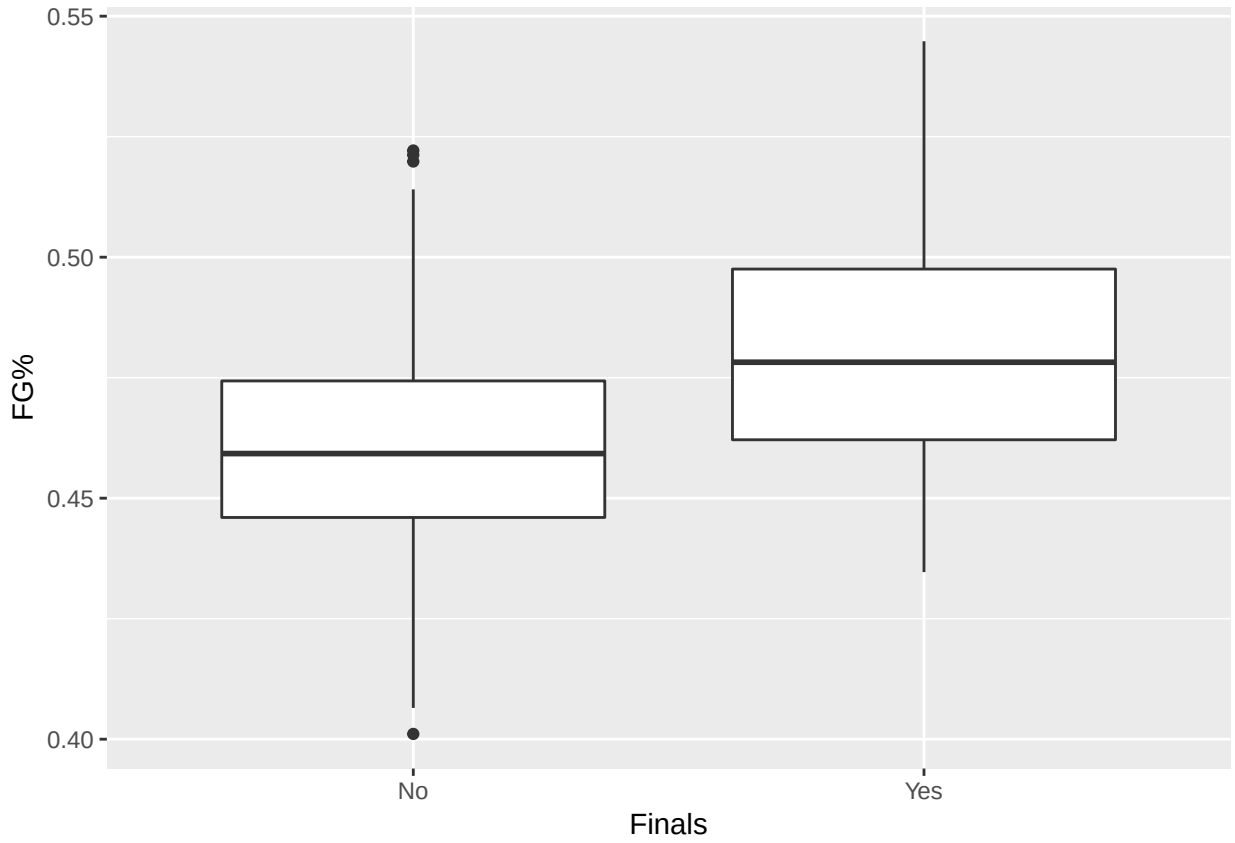
```
## # A tibble: 6 x 36
##   Season Tm    Champion Finals Lg        G     W     L `W/L%`    MP    FG
##   <chr>  <chr> <chr>    <chr>  <chr> <int> <int> <int>  <dbl> <int> <int>
## 1 1979-~ ATL   No       No     NBA      82    50    32  0.61  19780  3261
## 2 1980-~ ATL   No       No     NBA      82    31    51  0.378 19930  3291
## 3 1981-~ ATL   No       No     NBA      82    42    40  0.512 19880  3210
## 4 1982-~ ATL   No       No     NBA      82    43    39  0.524 19780  3352
## 5 1983-~ ATL   No       No     NBA      82    40    42  0.488 19855  3230
## 6 1984-~ ATL   No       No     NBA      82    34    48  0.415 19855  3444
## # ... with 25 more variables: FGA <int>, `2P` <int>, `2PA` <int>,
## #   `3P` <int>, `3PA` <int>, FT <int>, FTA <int>, ORB <int>, DRB <int>,
## #   TRB <int>, AST <int>, STL <int>, BLK <int>, TOV <int>, PF <int>,
## #   PTS <int>, `FG%` <dbl>, `2P%` <dbl>, `3P%` <dbl>, `FT%` <dbl>,
## #   TRBperG <dbl>, ASTperG <dbl>, STLperG <dbl>, BLKperG <dbl>,
## #   PTSperG <dbl>
```

```r
# select 70% of the teams data to form our training data and the remaining 30% to use as test data
noTrainRows <- floor(nrow(teams) * .7)
set.seed(4)
sampleRows <- sample(1:nrow(teams), noTrainRows, replace = F)
trainData <- teams[sampleRows,]
testData <- teams[-sampleRows,]
```

Plot exploration

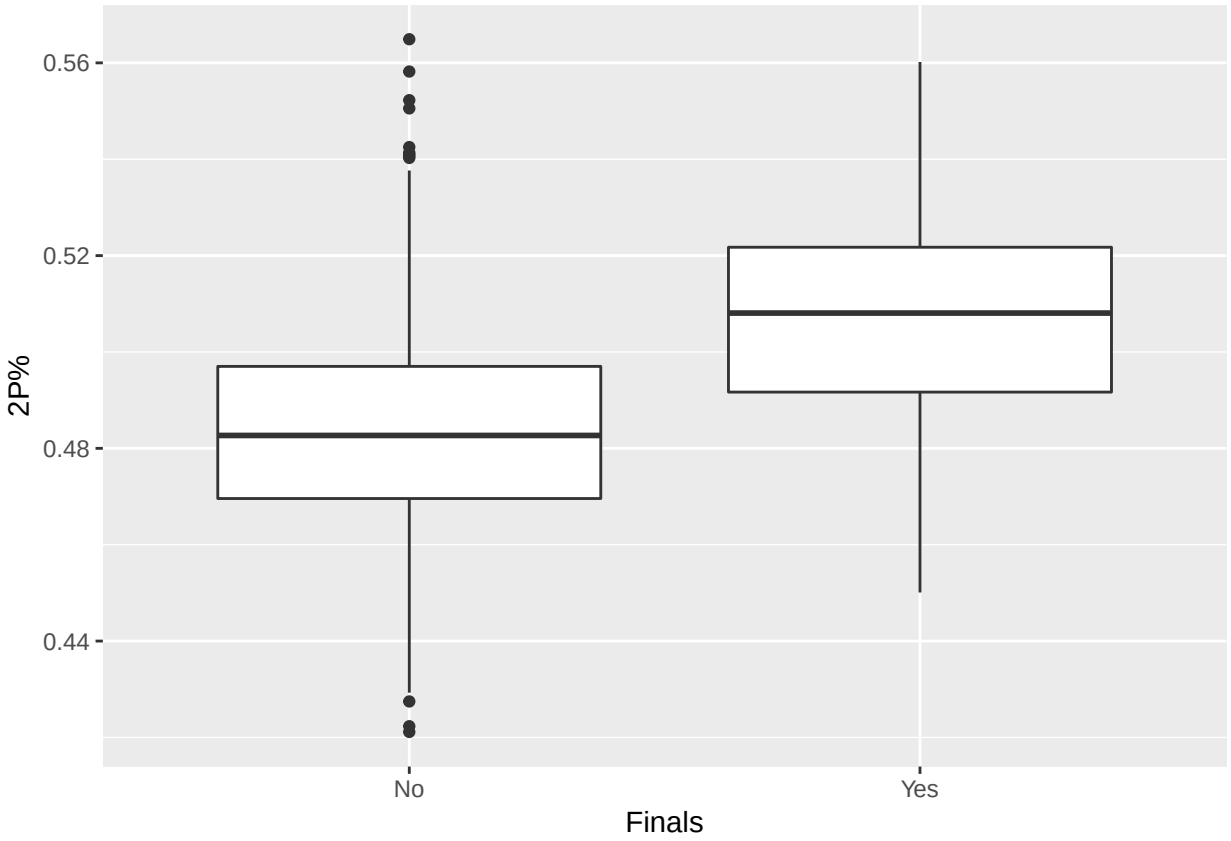
We have created boxplots to show the distributions of all teams in our data that did not make the Finals compared to teams that did for the statistics that we created.

```r
ggplot(data = teams, aes(x = teams$Finals, y = teams$`FG%`)) + geom_boxplot() + labs(x = "Finals", y =
```



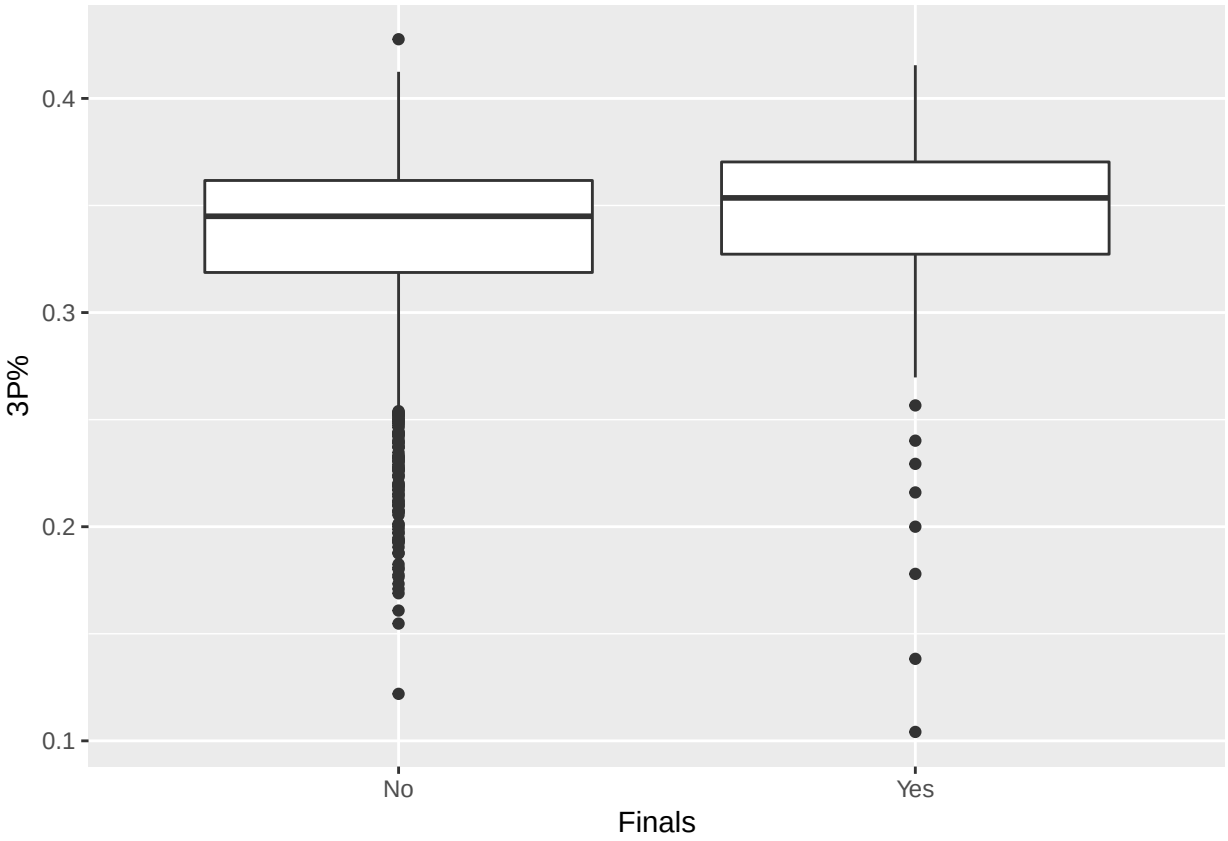
FG% appears to be a decent predictor of Finals.

```
ggplot(data = teams, aes(x = teams$Finals, y = teams$`2P%`)) + geom_boxplot() + labs(x = "Finals", y =
```



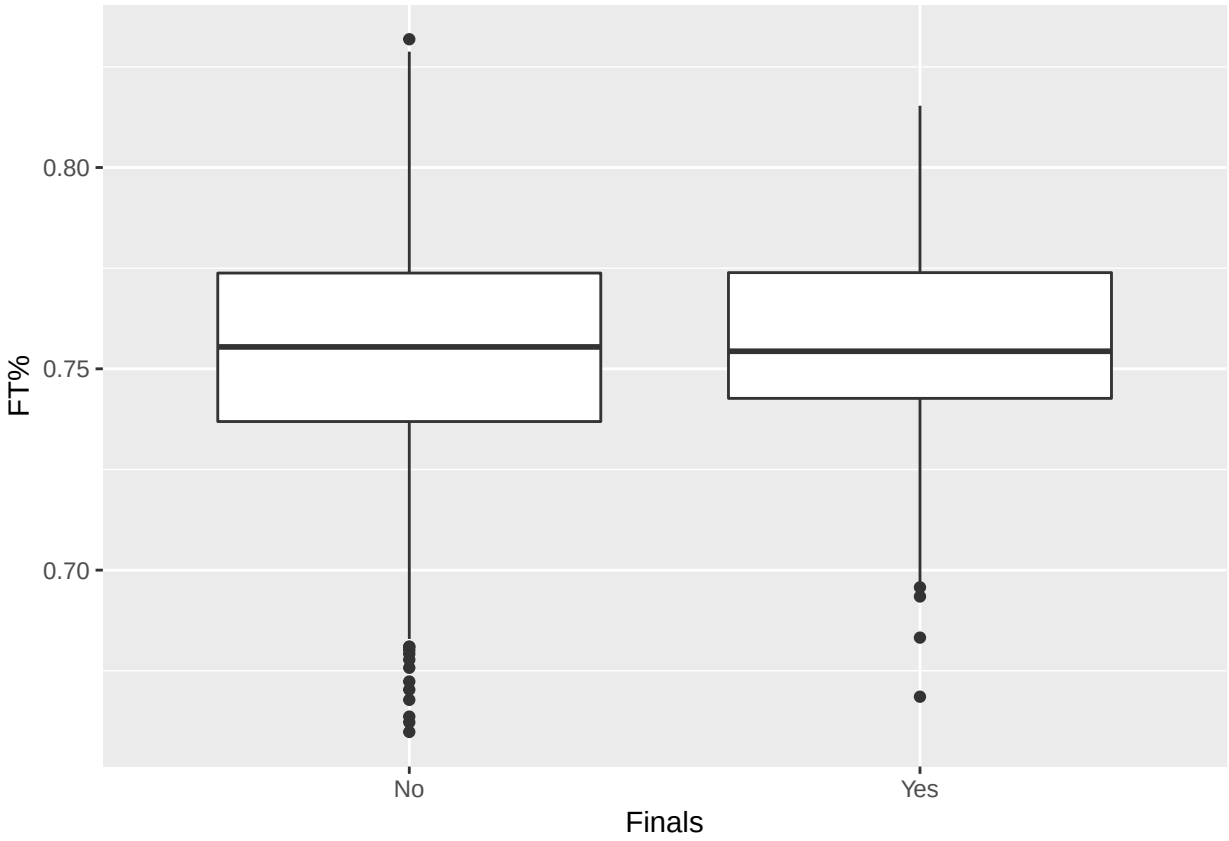
2P% appears to be a decent predictor of Finals.

```
ggplot(data = teams, aes(x = teams$Finals, y = teams$`3P%`)) + geom_boxplot() + labs(x = "Finals", y =
```



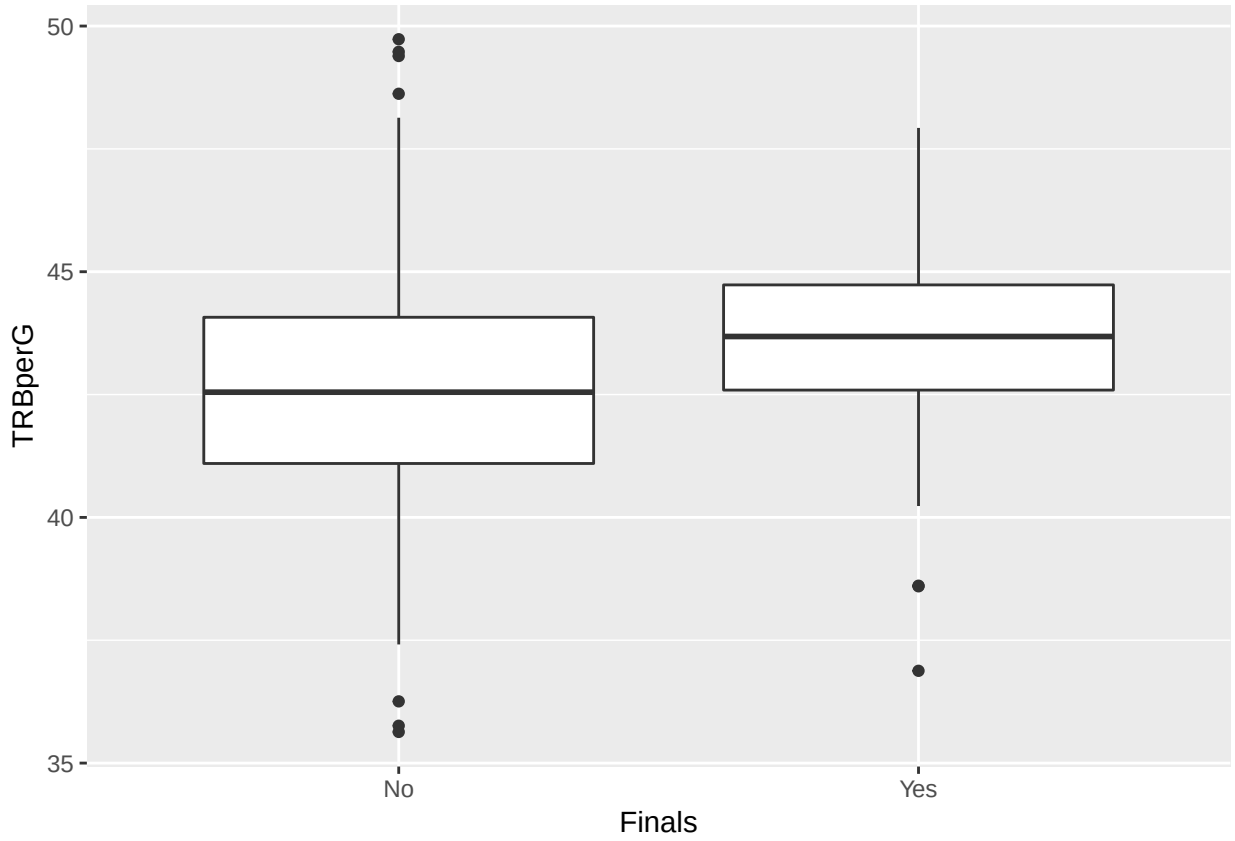
3P% appears to have little to no correlation with Finals.

```
ggplot(data = teams, aes(x = teams$Finals, y = teams$`FT%`)) + geom_boxplot() + labs(x = "Finals", y =
```



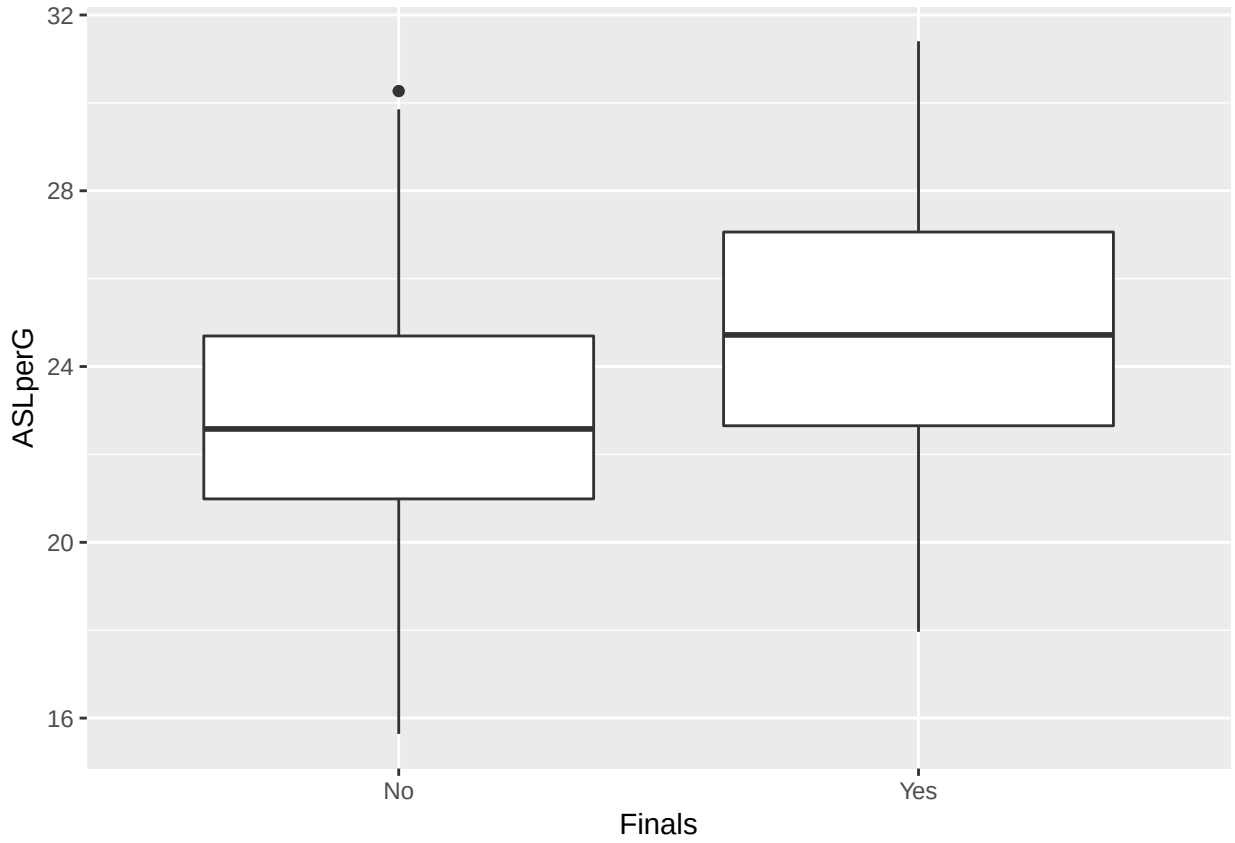
FT% appears to have little to no correlation with Finals.

```
ggplot(data = teams, aes(x = teams$Finals, y = teams$TRBperG)) + geom_boxplot() + labs(x = "Finals", y =
```



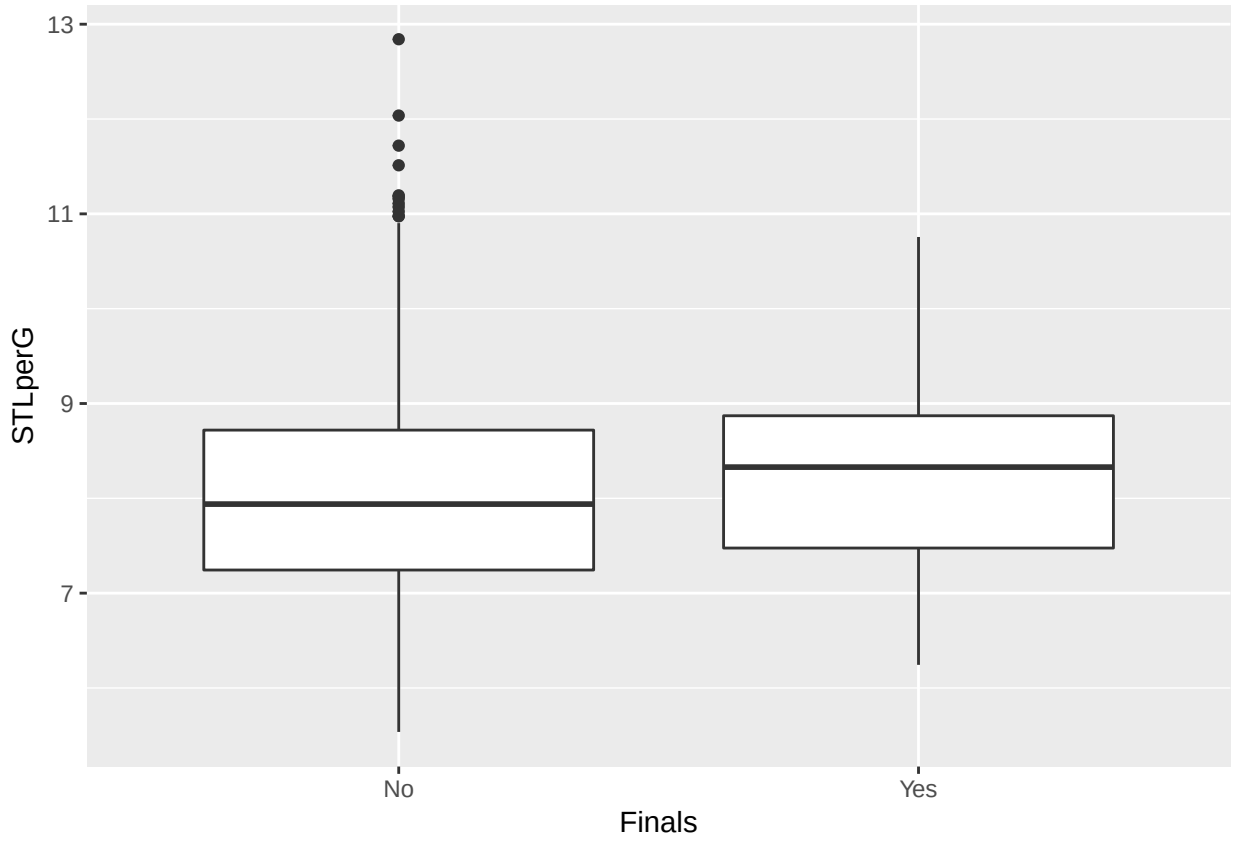
TRBperG appears to be a decent predictor of Finals.

```
ggplot(data = teams, aes(x = teams$Finals, y = teams$ASTperG)) + geom_boxplot() + labs(x = "Finals", y =
```



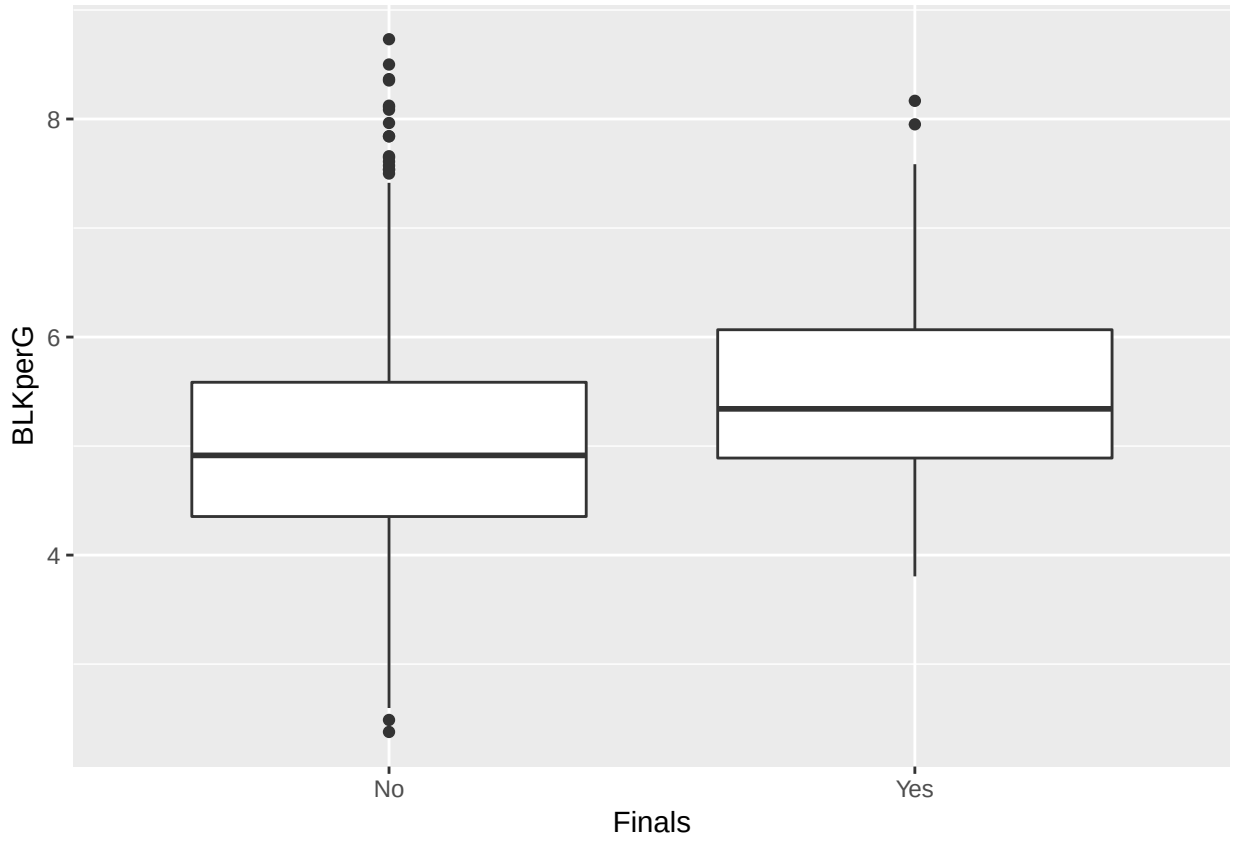
ASTperG appears to be a decent predictor of Finals.

```
ggplot(data = teams, aes(x = teams$Finals, y = teams$STLperG)) + geom_boxplot() + labs(x = "Finals", y =
```



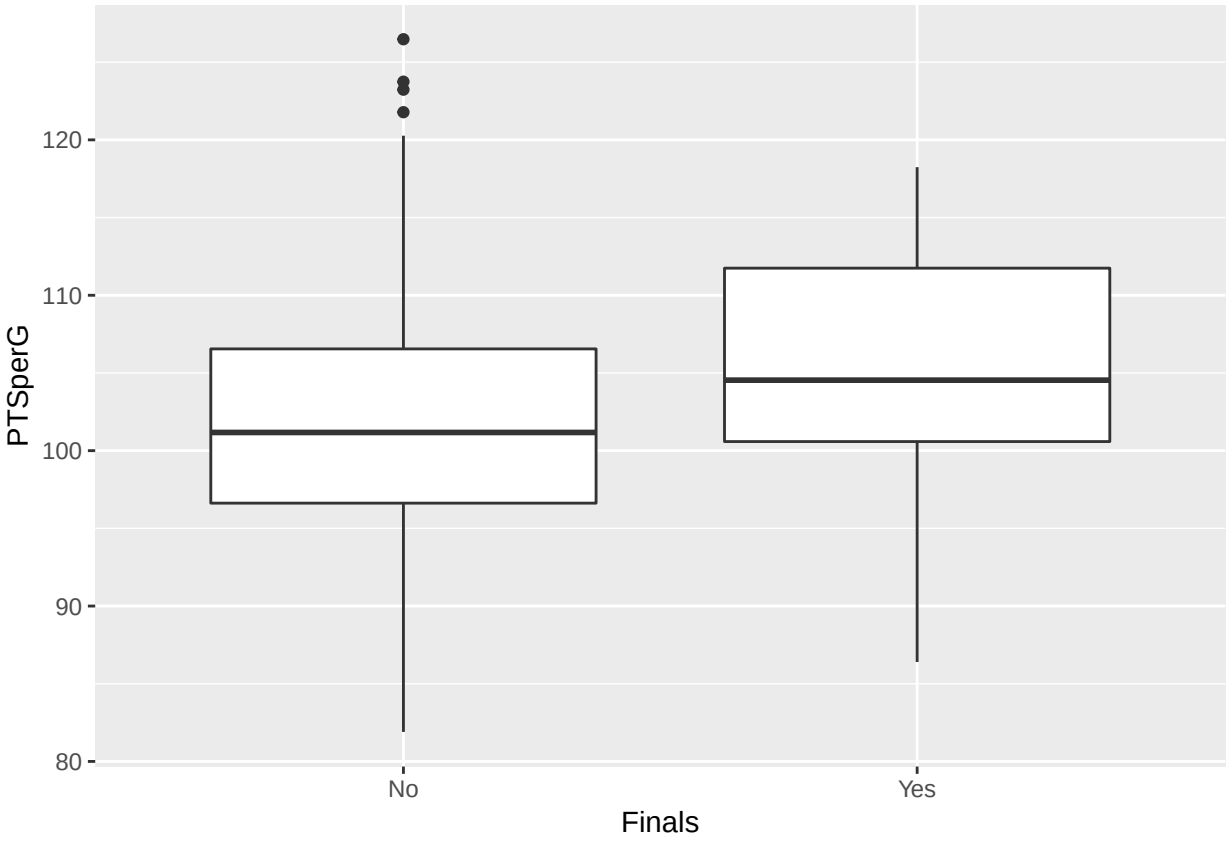
STLperG appears to have minimal correlation with Finals.

```
ggplot(data = teams, aes(x = teams$Finals, y = teams$BLKperG)) + geom_boxplot() + labs(x = "Finals", y =
```



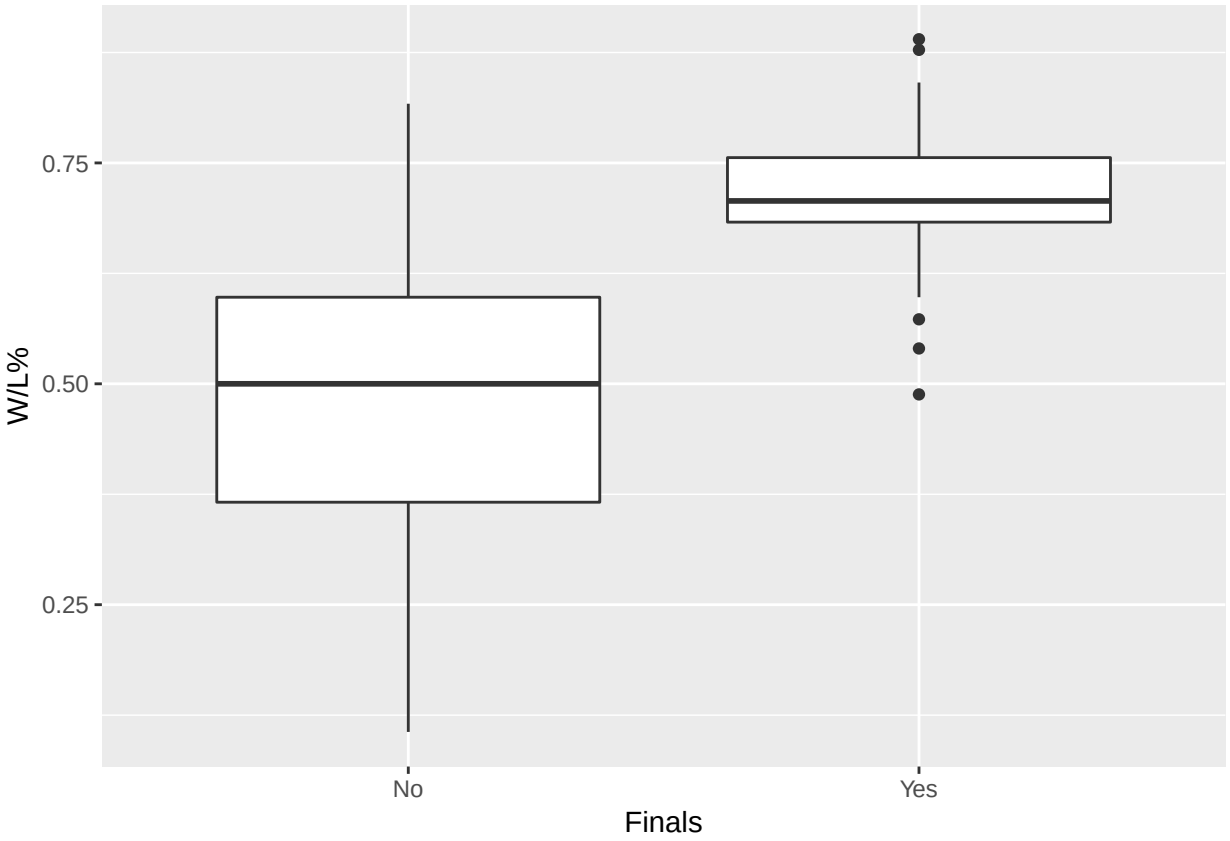
BLKperG appears to have minimal correlation with Finals.

```
ggplot(data = teams, aes(x = teams$Finals, y = teams$PTSperG)) + geom_boxplot() + labs(x = "Finals", y =
```



PTSperG appears to have minimal correlation with Finals.

```
ggplot(data = teams, aes(x = teams$Finals, y = teams$`W/L%`)) + geom_boxplot() + labs(x = "Finals", y =
```



W/L% appears to be a very good predictor of Finals.

Description of modeling approach

We used a generalized linear model on our training data to implement logistic regression, with logLoss as our accuracy measurement. Our intial choices for model features were the features that we created, in addition to W/L%. We chose these because they scale each team data so that team data can be compared across years. From our exploratory data analysis, we expected 3P%, FT%, STLperG, BLKperG, and PTSperG to not be significant predictors in our generalized linear model. However, after running our model we found that the set of features 2P%, 3P%, TRBperG, ASTperG, PTSperG, and W/L% provides the best performing model, as found by taking our original model and removing insignificant features until our model AIC was minimized.

```
control = trainControl(method = "repeatedcv",
                       number = 3,
                       classProbs = TRUE,
                       summaryFunction=mnLogLoss,
                       verboseIter = TRUE)

glm_fit = train(Finals ~ `2P%` + `3P%` + TRBperG + ASTperG + PTSperG + `W/L%`,
                data= trainData,
                method = "glm",
                family = binomial(),
                metric = "logLoss",
                trControl = control,
                preProcess = c("center", "scale")) #important to set metric to logLoss to tune for logl
```

```
## + Fold1.Rep1: parameter=none
## - Fold1.Rep1: parameter=none
## + Fold2.Rep1: parameter=none
## - Fold2.Rep1: parameter=none
## + Fold3.Rep1: parameter=none
## - Fold3.Rep1: parameter=none
## Aggregating results
## Fitting final model on full training set
```

```
summary(glm_fit)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.47707  -0.19906  -0.07275  -0.01816   3.09571
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -5.6836     0.5909  -9.619  < 2e-16 ***
## `\\`2P%\\``    1.1431     0.3427   3.336 0.000851 ***
## `\\`3P%\\``   -0.5788     0.2262  -2.558 0.010516 *
## TRBperG        0.4326     0.2639   1.639 0.101210
## ASTperG        0.5781     0.2733   2.115 0.034450 *
## PTSperG       -1.5019     0.4704  -3.193 0.001409 **
## `\\`W/L%\\``   2.7698     0.4355   6.360 2.02e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 343.13  on 771  degrees of freedom
## Residual deviance: 193.27  on 765  degrees of freedom
## AIC: 207.27
##
## Number of Fisher Scoring iterations: 8
```

```
pred <- as.numeric(glm_fit$finalModel$fitted.values>0.5)

teams$Finals <- ifelse(teams$Finals == "Yes", 1, 0)
trainData <- teams[sampleRows,]
testData <- teams[-sampleRows,]

actual <- as.numeric(trainData$Finals)

m <- confusionMatrix(as.factor(pred), as.factor(actual), positive="1")
m
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 725  31
##          1   2  14
```

```
##
##                Accuracy : 0.9573
##                  95% CI : (0.9405, 0.9704)
##     No Information Rate : 0.9417
##     P-Value [Acc > NIR] : 0.03421
##
##                   Kappa : 0.442
##
##  Mcnemar's Test P-Value : 1.093e-06
##
##             Sensitivity : 0.31111
##             Specificity : 0.99725
##          Pos Pred Value : 0.87500
##          Neg Pred Value : 0.95899
##              Prevalence : 0.05829
##          Detection Rate : 0.01813
##    Detection Prevalence : 0.02073
##       Balanced Accuracy : 0.65418
##
##        'Positive' Class : 1
##
```

Initial results

The logLoss value of our generalized linear model is 0.142 and our confusion matrix has an accuracy of 0.957. We are planning on using alternative modeling techniques to find the best possible method of logistic regression for our problem. Additionally, we are planning to evaluate our problem as a time series problem. Since this class does not cover any techniques regarding time series, we will likely have to do more independent research to best tackle our problem.