NBA Stats Lab Team

Derek Wales and Ravitashaw Bathla

9/23/2019

```
#Loading Data for NBA
setwd("C:\\Users\\derek\\Desktop\\MIDS Program\\.Mids 1st Semester\\.GIT Organization\\Stats Assignments
nba <- read.csv("nba_games_stats.csv",header = TRUE,sep = ",",stringsAsFactors = FALSE)

# Set factor variables
nba$Home <- factor(nba$Home)
nba$Team <- factor(nba$Team)
nba$WINorLOSS <- factor(nba$WINorLOSS)

# Convert date to the right format
nba$Date <- as.Date(nba$Date, "%Y-%m-%d")

# Also create a binary variable from WINorLOSS.
# This is not always necessary but can be useful for R functions that prefer numeric
# binary variables to the original factor variables
nba$Win <- rep(0,nrow(nba))
nba$Win[nba$WINorLOSS=="W"] <- 1

# I picked the Lakers!
nba_reduced <- nba[nba$Team == "LAL", ]

# Set aside the 2017/2018 season as your test data
nba_reduced_train <- nba_reduced[nba_reduced$Date < "2017-10-01",]
nba_reduced_test <- nba_reduced[nba_reduced$Date >= "2017-10-01",]
```

1. Make exploratory plots to explore the relationships between Win and the following variables: Home, TeamPoints, FieldGoals., Assists, Steals, Blocks and Turnovers. Don't include any of the plots, just briefly describe the relationships.

Note for all Questions we used the Lakers as our team.

The plots revealed the obvious (the more the TeamPoints/FieldGoals./Assists/Blocks, the more likely the team will win). There were several elements that were interesting

1. The plots of Turnovers to Win ratio was close (with the winning/losing team being within 3 Turnovers each) and the Home team wins nearly 60% of the time.
2. Home played a very important factor in determining the Winning of the game. Based upon the result of the chi-squared test, the p-value was also very significant, indicating the statistical significance of Home on winning (Win)

Refer to *Appendix Q1* for all the plots and chi-sqaured test result.

2. There are several combinations of variables we should not include as predictors in the logistic model. Identify at least two pairs and explain in at most two sentences, why we should not include them in the model at the same time.

FieldGoals. - *TotalPoints* and *Turnovers* - *DefensiveSteals* are both highly correlated because the vast majority of points scored in a basketball game are from field goals and many of the turnovers in basketball are

when the opponent steals the ball. Since the correlational coefficient is high (*Appendix Q2*), this results into the problem of multicollinearity and, therefore, we should not include them in the model at the same time.

3. Fit a logistic regression model for Win (or WinorLoss) using Home, TeamPoints, FieldGoals., Assists, Steals, Blocks and Turnovers. as your predictors. Using the vif function, are there are any concerns regarding multicollinearity in this model?

Based upon the benchmarks if the VIF value is 1 then its not correlated, 2-5 moderately correlated, and greater than 5 is highly correlated. Keeping that in mind Home and Blocks seems to be not correlated (values are very close to 1). Similarly, Assists, Steals and Turnovers are close to 1 but less than 2, so relatively,these can be considered as not correlated. FieldGoals. has the highest value and it can be categorized as moderately correlated than any of the other variables. However, the affect of multicolinearity is not that high, so it is safe to include these variables as predictor variables.

```
model_1 <- glm(Win ~ Home + TeamPoints + FieldGoals. +
               Assists + Steals + Blocks + Turnovers, data = nba_reduced_train,
               family=binomial(link=logit))

VIF(model_1)
```

```
##        Home  TeamPoints FieldGoals.     Assists      Steals      Blocks
##    1.063248    1.792945    2.093871    1.333389    1.114480    1.022820
##   Turnovers
##    1.224493
```

4. Present the output of the fitted model and interpret the significant coefficients in terms of the odds of your team winning an NBA game.

From the summary of the model we can see that Home, TeamPoints and Steals are statistically significant coefficients in predicting the odds of winning.

This means that if the game is a Home game (team playing in the hometown) the odds of winning increase by exp(0.789)=????

Similarly, for each TeamPoint scored, the odds of winning increase by 8 percent (exp(0.08044) -1 = 0.083764) Also, for each Steal, the odds of winning increase by 20 percent (exp(0.18793) -1 = 0.206749)

Note, this model is only representative of one team and doesn't consider how well the opponent is playing.

```
summary(model_1)
```

```
##
## Call:
## glm(formula = Win ~ Home + TeamPoints + FieldGoals. + Assists +
##     Steals + Blocks + Turnovers, family = binomial(link = logit),
##     data = nba_reduced_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0855  -0.6439  -0.3791   0.5837   2.3499
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -15.15218    2.24635  -6.745 1.53e-11 ***
## HomeHome      0.76259    0.35920   2.123 0.033751 *
## TeamPoints    0.08044    0.02383   3.375 0.000738 ***
## FieldGoals.   6.19207    5.49910   1.126 0.260159
## Assists       0.03864    0.04540   0.851 0.394744
```

```
## Steals        0.18793    0.06688   2.810 0.004953 **
## Blocks        0.04117    0.07395   0.557 0.577753
## Turnovers     0.01386    0.04669   0.297 0.766555
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 282.03  on 245  degrees of freedom
## Residual deviance: 209.33  on 238  degrees of freedom
## AIC: 225.33
##
## Number of Fisher Scoring iterations: 5
```

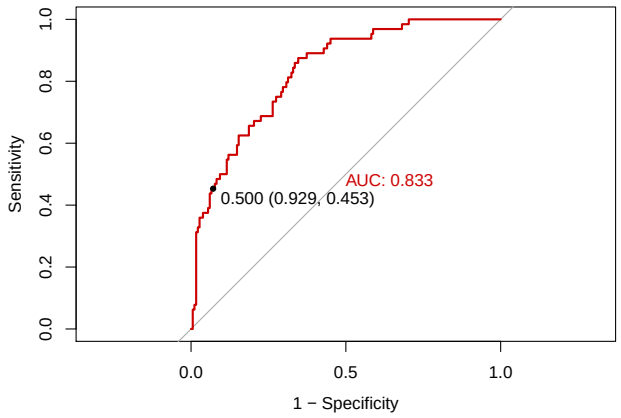
5. Using 0.5 as your cutoff for predicting wins or losses (1 vs 0) from the predicted probabilities, what is the accuracy of this model? Plot the roc curve for the fitted model. What is the AUC value?

With threshold of 0.5 on Wins or Losses prediction, the AUC Value is 0.833 (see code/graph below).

```
invisible(pROC::roc(nba_reduced_train$Win, fitted(model_1), plot=T, print.thres= 0.5,
                    legacy.axes=T,
                    print.auc =T,col="red3"))
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```



6. Now add Opp.FieldGoals. as a predictor to the previous model. Is the coefficient significant? If yes, interpret the coefficient in the context of the question.

Yes, Opp.FieldGoals.'s p value is very low, thus the coefficient is statistically significant. This means that every percent increase in FieldGoal. of the opponent, the odds of winning decrease by 42 percent (exp(-0.53) -1 = 0.58 - 1).

```
model_2 <- glm(Win ~ Home + TeamPoints + FieldGoals. +
                 + Assists + Steals + Blocks + Turnovers + Opp.FieldGoals.,
               data = nba_reduced_train, family=binomial(link=logit))

summary(model_2)
```

```
##
## Call:
## glm(formula = Win ~ Home + TeamPoints + FieldGoals. + +Assists +
##     Steals + Blocks + Turnovers + Opp.FieldGoals., family = binomial(link = logit),
##     data = nba_reduced_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.45434  -0.28990  -0.07669   0.07582   2.33277
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -7.33782    3.11984  -2.352 0.018673 *
## HomeHome          0.87779    0.51587   1.702 0.088837 .
## TeamPoints        0.20454    0.04526   4.519 6.22e-06 ***
## FieldGoals.      17.92151    8.22539   2.179 0.029346 *
## Assists          -0.05263    0.06433  -0.818 0.413312
## Steals            0.45401    0.11704   3.879 0.000105 ***
## Blocks           -0.15639    0.10440  -1.498 0.134146
## Turnovers        -0.05011    0.07057  -0.710 0.477652
## Opp.FieldGoals. -53.29401    8.34815  -6.384 1.73e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 282.03  on 245  degrees of freedom
## Residual deviance: 111.96  on 237  degrees of freedom
## AIC: 129.96
##
## Number of Fisher Scoring iterations: 7
```

7. What is the accuracy of this new model? Plot the roc curve for the fitted model. What is the new AUC value? Which model predicts the odds of winning better?

The accuracy went from 0.804 to 0.894, this means that adding the Opp.FieldGoals. (opponents field goals) in the model, increased the model's accuracy by 9%.

```
#Confusion Matrix from Model 1
conf_mat_1 <- confusionMatrix(as.factor(ifelse(fitted(model_1) >= 0.5, "1","0")),
                              as.factor(nba_reduced_train$Win),positive = "1")

conf_mat_1$overall["Accuracy"]
```

```
## Accuracy
## 0.804878
```

```
#Confusion Matrix from Model 2
conf_mat_2 <- confusionMatrix(as.factor(ifelse(fitted(model_2) >= 0.5, "1","0")),
                              as.factor(nba_reduced_train$Win),positive = "1")

conf_mat_2$overall["Accuracy"]
```

```
##  Accuracy
## 0.8943089
```

8. Using the results of the model with the better predictive ability, what suggestions do you have for the coach of your team trying to improve the odds of his team winning a regular season game?

Based upon the results from the model, the Coach should focus more on helping the team to improve defense in the game, as every percent increase in opponents field goal, decreases their odds of winning considerably.

Also, stealing the ball from opponent also helps in increasing the odds of winning, so the coach should focus on training the team tactics for stealing the balls.

9. Use this model to predict out-of-sample probabilities for the nba_reduced_test data. Using 0.5 as your cutoff for predicting wins or losses (1 vs 0) from the out-of-sample predicted probabilities, what is the out-of-sample accuracy? How well does your model do in predicting data for the 2017/2018 season?

The accuracy for the test data is 79 precent. For training data the accuracy was 89% so it has decreased by 10% on the test data. Therefore, the predictor is not a very good estimator of predicting out-of-sample probabilities.

```
conf_mat_3 <- confusionMatrix(as.factor(ifelse(predict(model_2,nba_reduced_test,
              type="response") >= 0.5, "1","0")), as.factor(nba_reduced_test$Win),
              positive = "1")

conf_mat_3$overall["Accuracy"]
```

```
##  Accuracy
## 0.7926829
```

10. Using the change in deviance test, test whether including Opp.Assists and Opp.Blocks in the model at the same time would improve the model. Is there any other variable in this dataset which we did not consider that you think might improve our model? Which one and why?

The p-value from the chi-sqaured test is very high. This implies that including Opp.Assists and Opp.Blocks in the model does not help in improving the prediction from the model.

```
model_3 <- glm(Win ~ Home + TeamPoints + FieldGoals. +
                Assists + Steals + Blocks + Turnovers + Opp.FieldGoals. + Opp.Assists
              + Opp.Blocks, data = nba_reduced_train, family=binomial(link=logit))

anova(model_2, model_3, test= "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Win ~ Home + TeamPoints + FieldGoals. + +Assists + Steals + Blocks +
##     Turnovers + Opp.FieldGoals.
## Model 2: Win ~ Home + TeamPoints + FieldGoals. + Assists + Steals + Blocks +
##     Turnovers + Opp.FieldGoals. + Opp.Assists + Opp.Blocks
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       237     111.96
## 2       235     111.15  2  0.80671   0.6681
```

Adding TotalFouls seems to be statistically significany and the p-value obtained from chi-squared test by including TotalFouls in one model is statistically significany. Therefore, adding TotalFouls might improve the accuracy in predicting the model.

```
model_4 <- glm(Win ~ Home + TeamPoints + FieldGoals. +
               Assists + Steals + Blocks + Turnovers + Opp.FieldGoals. + TotalFouls,
```

```
              data = nba_reduced_train, family=binomial(link=logit))

anova(model_2, model_4, test= "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Win ~ Home + TeamPoints + FieldGoals. + +Assists + Steals + Blocks +
##     Turnovers + Opp.FieldGoals.
## Model 2: Win ~ Home + TeamPoints + FieldGoals. + Assists + Steals + Blocks +
##     Turnovers + Opp.FieldGoals. + TotalFouls
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       237     111.96
## 2       236     104.19  1   7.7707  0.00531 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#AIC(model_2)
#AIC(model_3)
```

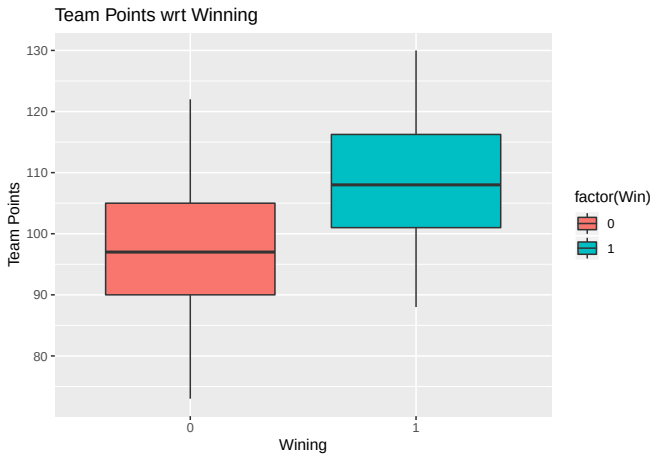
Appendix

Appendix Q1

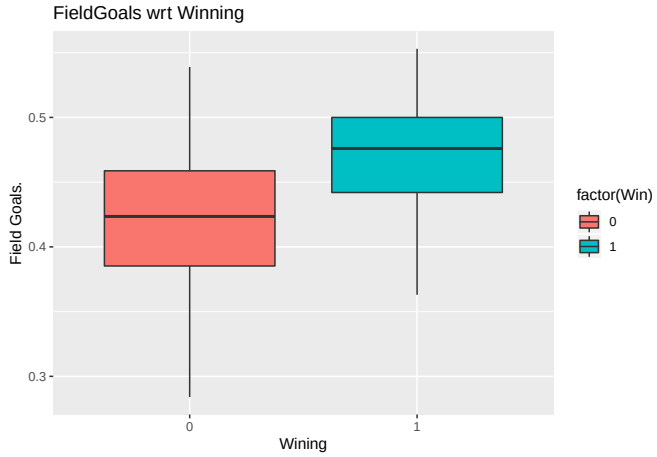
```
par(mfrow=c(3,2))
#ggplot(data=nba_reduced_train, aes(x=as.factor(Win), fill=Home, y=WINorLOSS))+
#  xlab('Wining') +  geom_bar(stat="identity")

#Chi-squared test for determinig the relation between binary variables
chisq.test(table(nba_reduced_train[,c("Win","Home")])) # Home influences Win
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(nba_reduced_train[, c("Win", "Home")])
## X-squared = 6.1035, df = 1, p-value = 0.01349
```

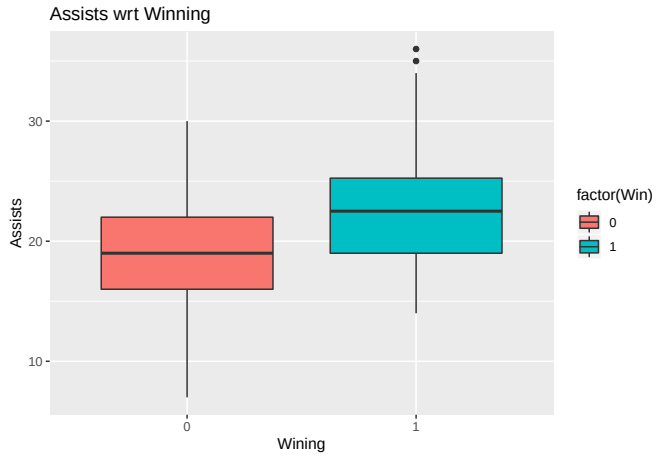
```
ggplot(nba_reduced_train, aes(y=TeamPoints, x=factor(Win), fill=factor(Win)))+
  geom_boxplot() + xlab('Wining')+ ylab('Team Points')+ ggtitle('Team Points wrt Winning')
```



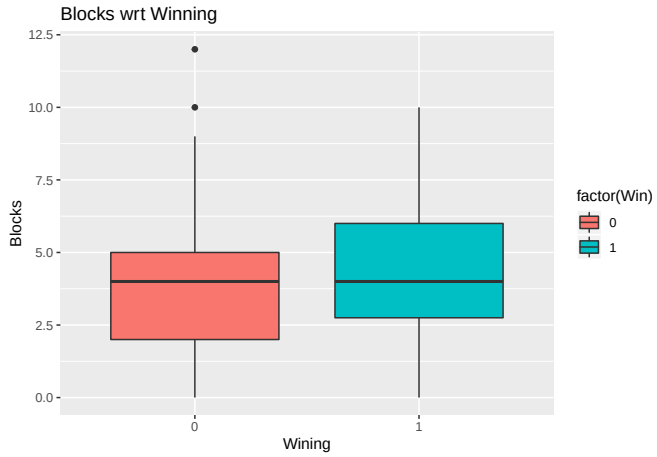
```
ggplot(nba_reduced_train, aes(y=FieldGoals., x=factor(Win), fill=factor(Win)))+
  geom_boxplot() + xlab('Wining')+ ylab('Field Goals.')+ ggtitle('FieldGoals wrt Winning')
```



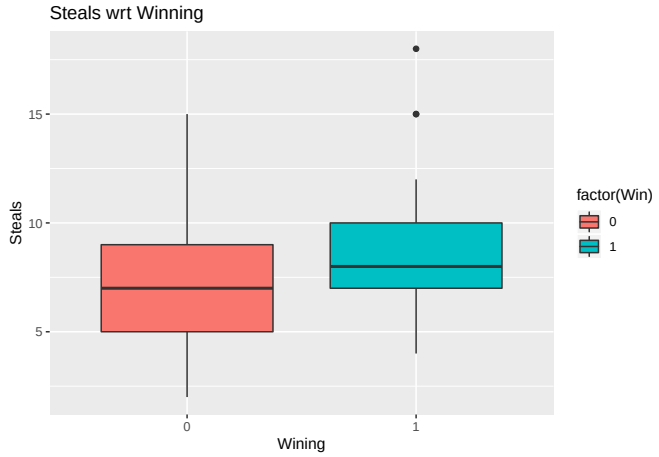

```
ggplot(nba_reduced_train, aes(y=Assists, x=factor(Win), fill=factor(Win)))+
  geom_boxplot() + xlab('Wining')+ ylab('Assists')+ ggtitle('Assists wrt Winning')
```



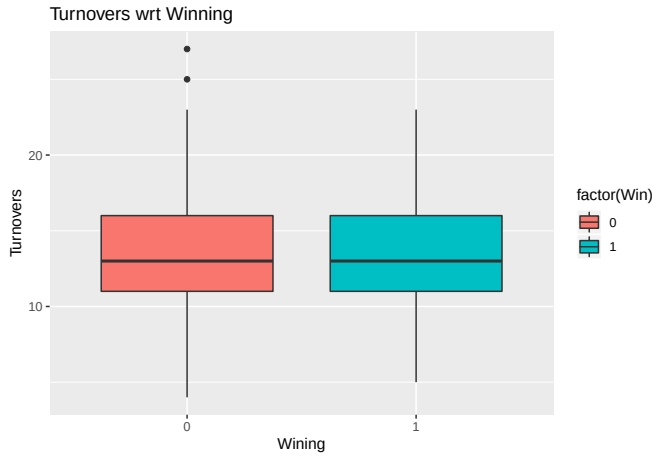

```
ggplot(nba_reduced_train, aes(y=Blocks, x=factor(Win), fill=factor(Win)))+
  geom_boxplot() + xlab('Wining')+ ylab('Blocks')+ ggtitle('Blocks wrt Winning')
```



```
ggplot(nba_reduced_train, aes(y=Steals, x=factor(Win), fill=factor(Win)))+
  geom_boxplot() + xlab('Wining')+ ylab('Steals')+ ggtitle('Steals wrt Winning')
```




```
ggplot(nba_reduced_train, aes(y=Turnovers, x=factor(Win), fill=factor(Win)))+
  geom_boxplot() + xlab('Wining')+ ylab('Turnovers')+ ggtitle('Turnovers wrt Winning')
```



Appendix Q2

```
cor(nba_reduced_train[8:ncol(nba_reduced_train)])
```