Comparison in Predictability of Winner in Soccer vs. Football

Jörg Duschmalé

7/28/2021

Introduction

On a global scale, soccer is most certainly the most followed sports in the world. In the United States of America, however, football, tends to be more popular. Albeit both sports historically are cousins (together with rugby, aussie rules football and the likes) they are distinctly different with respect rules, setup and the necessary skills to win a game.

In football scores are higher than in soccer and, additionally, the events scoring points tend to happen more often. This usually results in the better team winning the game. In stark contrast, as the scoring events in soccer happen much less often (many games actually end 0:0 or with a single goal margin), the winning team of a particular game is often not necessarily the better team as well. The significantly better team can dominate the game, push for the winning goal throughout and, yet, lose because of an individual mistake of their goalkeeper in the last minute. While for the passionate supporter of football this comes across as "unfair", for many soccer aficionado it is exactly this element of unpredictability that makes this game so irresistible (Tierney 2014).

Within this report, which is the capstone project of a HarvardX Professional Certificate in data science (Irizarry 2021), using the caret package (Kuhn 2009), several methods discussed in the course are used to predict soccer and football results based on game statistics datasets. Indeed, it is found that football results are predicted with a higher accuracy than soccer results.

The datasets used are readily available on Kaggle ("Kaggle: Your Home for Data Science," n.d.). For football the "NFL Game Stats" dataset was chosen (Sasser 2021). This dataset contains NFL games from seasons 2010 - 2019.
For soccer the selected dataset is called "English Premier League Match Events and Results 2001-2002 season onwards" (Mohr 2021) and contains data on games in the EPL between years 2001 and 2020.

Required Libraries

The following libraries are necessary for the computation.

```r
library(dslabs)
library(tidyverse)
library(caret)
library(randomForest)
library(MASS)
```

Loading the Datasets and Preparing the Data

The required datasets were downloaded from kaggle ("Kaggle: Your Home for Data Science," n.d.) and saved on the harddrive within the datascience\capstone_final folder. From there the data is loaded into R using the read.csv function.

```r
# loading the data from the harddrive
data<-read.csv("C:\\Users\\duschmaj\\Desktop\\datascience\\capstone_final\\matches.csv")

gs2010 <- read.csv("C:\\Users\\duschmaj\\Desktop\\datascience\\capstone_final\\game_stats_2010.csv")
gs2011 <- read.csv("C:\\Users\\duschmaj\\Desktop\\datascience\\capstone_final\\game_stats_2011.csv")
gs2012 <- read.csv("C:\\Users\\duschmaj\\Desktop\\datascience\\capstone_final\\game_stats_2012.csv")
gs2013 <- read.csv("C:\\Users\\duschmaj\\Desktop\\datascience\\capstone_final\\game_stats_2013.csv")
gs2014 <- read.csv("C:\\Users\\duschmaj\\Desktop\\datascience\\capstone_final\\game_stats_2014.csv")
gs2015 <- read.csv("C:\\Users\\duschmaj\\Desktop\\datascience\\capstone_final\\game_stats_2015.csv")
gs2016 <- read.csv("C:\\Users\\duschmaj\\Desktop\\datascience\\capstone_final\\game_stats_2016.csv")
gs2017 <- read.csv("C:\\Users\\duschmaj\\Desktop\\datascience\\capstone_final\\game_stats_2017.csv")
gs2018 <- read.csv("C:\\Users\\duschmaj\\Desktop\\datascience\\capstone_final\\game_stats_2018.csv")
gs2019 <- read.csv("C:\\Users\\duschmaj\\Desktop\\datascience\\capstone_final\\game_stats_2019.csv")
data_nfl <- rbind(gs2010,gs2011,gs2012,gs2013,gs2014,gs2015,gs2016,gs2017,gs2018,gs2019)

#select relevant predictors
data <- data %>% filter(game_status=="FT") %>% summarize(
  home_score,
  away_score,
  home_possessionPct,
  home_shotsSummary,
  away_shotsSummary,
  home_wonCorners,
  away_wonCorners,
  home_saves,
  away_saves)

data_nfl <- data_nfl %>% summarize(
  H.RushAtt,
  H.RushYards,
  H.PassYards,
  H.Turnover,
  H.Score,
  A.RushAtt,
  A.RushYards,
  A.PassYards,
  A.Turnover,
  A.Score,
  )
```

Table 1: The chosen predictors for each sport

Soccer	Football
Ball possession [%]	Rushing attempts
Total shots	Rushing Yards
Shots on goal	Passing Yards
Corners won	Turnover
Saves	

From the available datasets a set of statistics for each sport were chosen, that were considered to be predictive for match outcome. In doing so, care was taken not to use many more predictors with one over the other sport in order to keep a certain amount of comparability. The chosen predictors are given in Table 1

Additionally, the soccer dataset contains % characters in the possession column, which needs to be removed.

Also there is a single shotsSummary column in the format “shots (thereof on goal)” which needs to be split into two individual columns. Both is achieved using the code below.

```
#remove 0% possessionPct because this is indication that there is missing data.
data <- data %>% filter(home_possessionPct !="0%")

#make possession numeric / split shots and shots on target
data <- data %>% mutate(home_possessionPct=as.numeric(str_replace(data$home_possessionPct, "[%]", "")))

data <- data %>%
  mutate(home_shots=as.numeric(str_extract(home_shotsSummary,"[0-9]*"))) %>%
  mutate(away_shots=as.numeric(str_extract(away_shotsSummary,"[0-9]*"))) %>%
  mutate(home_shots_ontarget=str_remove(home_shotsSummary,"^[0-9]*\\s\\(")) %>%
  mutate(home_shots_ontarget=as.numeric(str_remove(home_shots_ontarget, "\\)"))) %>%
  mutate(away_shots_ontarget=str_remove(away_shotsSummary,"^[0-9]*\\s\\(")) %>%
  mutate(away_shots_ontarget=as.numeric(str_remove(away_shots_ontarget, "\\)")))
data <- data[,-which(names(data) %in% c("home_shotsSummary", "away_shotsSummary"))]
```

Next both in the football as well as the soccer dataset a column is introduced containing a factor if the hometeam (h) or the awayteam (a) won or if there was a draw (d). Scores are then removed from the dataset, because they should not be used for predictions (as they decide the outcome already). In the case of the football dataset there are only 8 entries with a draw (not shown) which is not enough instances for prediction. Thus these entries are removed. In the soccer dataset there are some instances with 0% ball possession. These entries are removed as well, as they are indicative of missing data. Finally entries containing Nas are removed as well. All of this is done with the following piece of code.

```
# decide if the hometeam (h) or the awayteam (a) won, or if there was a draw (d)
data <- data %>%
  mutate(winner=ifelse(home_score>away_score, "h", ifelse(home_score==away_score, "d", "a"))) %>%
  mutate(winner=as.factor(winner))

data_nfl <- data_nfl %>%
  mutate(winner=ifelse(H.Score>A.Score, "h", ifelse(H.Score==A.Score, "d", "a")))

# Remove scores as they should not be used as predictors
data <- data[,-which(names(data) %in% c("home_score", "away_score"))]
data_nfl <- data_nfl[,-which(names(data_nfl) %in% c("H.Score", "A.Score"))]

#remove 0% possessionPct because this is indication that there is missing data.
data <- data %>% filter(home_possessionPct !=0)

#remove rows with Na's
data <- na.omit(data)
data_nfl <- na.omit(data_nfl)

#remove draws as they are less than 0.3% of entries
table(data_nfl$winner)
```

```
##
##    a    d    h
## 1105    8 1447
```

```
table(data_nfl$winner)/nrow(data_nfl)
```

```
##
##         a         d         h
## 0.4316406 0.0031250 0.5652344
```

```
data_nfl <- data_nfl %>% filter(winner != "d") %>% mutate(winner=as.factor(winner))
```

Creation of final validation sets

The data is then split into a dataset for model building and a validation dataset only used for the final validation of the built model. This is done using the following code.

```
# Creating a validation set for final model validation
set.seed(1)
test_index <- createDataPartition(y = data$winner, times = 1, p = 0.1, list = FALSE)
soccer <- data[-test_index,]
soccer_validation <- data[test_index,]

set.seed(240289)
test_index <- createDataPartition(y = data_nfl$winner, times = 1, p = 0.1, list = FALSE)
football <- data_nfl[-test_index,]
football_validation <- data_nfl[test_index,]
```

Data analysis

A summary of the datasets thus achieved is given below.

```
# Creating a validation set for final model validation
summary(soccer)
```

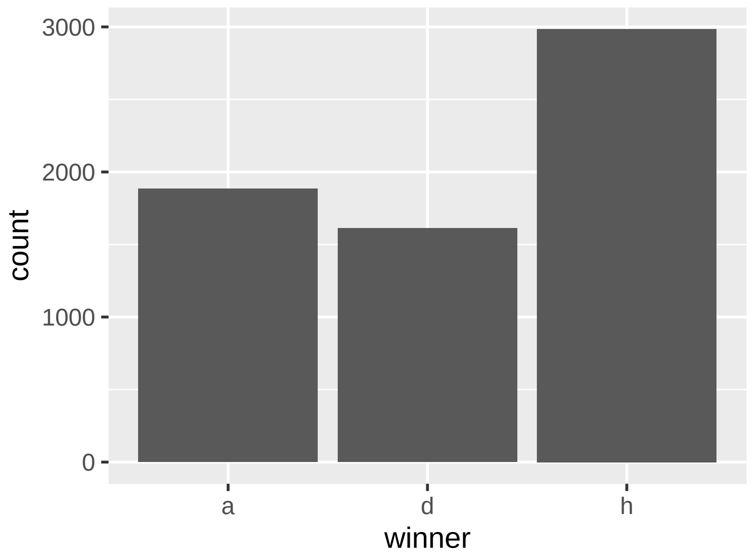
```
##  home_possessionPct home_wonCorners  away_wonCorners    home_saves
##  Min.   :18.0       Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:45.0       1st Qu.: 3.000   1st Qu.: 2.000   1st Qu.: 1.000
##  Median :51.0       Median : 5.000   Median : 4.000   Median : 3.000
##  Mean   :51.3       Mean   : 5.735   Mean   : 4.552   Mean   : 3.085
##  3rd Qu.:58.0       3rd Qu.: 8.000   3rd Qu.: 6.000   3rd Qu.: 4.000
##  Max.   :88.0       Max.   :20.000   Max.   :19.000   Max.   :18.000
##    away_saves       home_shots      away_shots    home_shots_ontarget
##  Min.   : 0.000   Min.   : 0.00   Min.   : 0.00   Min.   : 0.000
##  1st Qu.: 2.000   1st Qu.:10.00   1st Qu.: 8.00   1st Qu.: 3.000
##  Median : 3.000   Median :14.00   Median :11.00   Median : 5.000
##  Mean   : 3.856   Mean   :14.18   Mean   :11.26   Mean   : 5.478
##  3rd Qu.: 5.000   3rd Qu.:17.00   3rd Qu.:14.00   3rd Qu.: 7.000
##  Max.   :18.000   Max.   :43.00   Max.   :38.00   Max.   :21.000
##  away_shots_ontarget winner
##  Min.   : 0.000      a:1885
##  1st Qu.: 2.000      d:1613
##  Median : 4.000      h:2985
##  Mean   : 4.307
##  3rd Qu.: 6.000
##  Max.   :18.000
```

```r
summary(football)
```

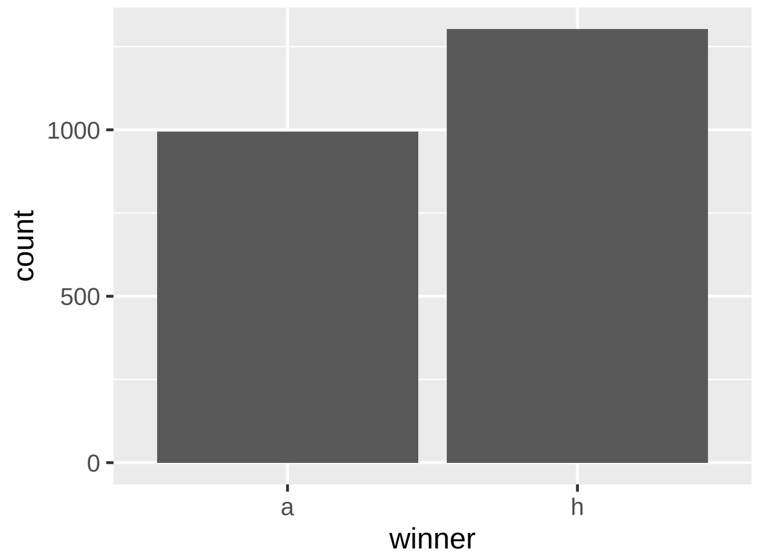
```
##    H.RushAtt       H.RushYards   H.PassYards      H.Turnover      A.RushAtt
##  Min.   : 6.00   Min.   :  7   Min.   : 57.0   Min.   :0.000   Min.   : 9.0
##  1st Qu.:22.00   1st Qu.: 79   1st Qu.:198.0   1st Qu.:0.000   1st Qu.:21.0
##  Median :27.00   Median :109   Median :249.0   Median :1.000   Median :25.0
##  Mean   :27.13   Mean   :115   Mean   :252.5   Mean   :1.461   Mean   :26.2
##  3rd Qu.:32.00   3rd Qu.:145   3rd Qu.:303.0   3rd Qu.:2.000   3rd Qu.:31.0
##  Max.   :54.00   Max.   :355   Max.   :527.0   Max.   :7.000   Max.   :55.0
##   A.RushYards      A.PassYards      A.Turnover     winner
##  Min.   :  1.0   Min.   : 39.0   Min.   :0.000   a: 994
##  1st Qu.: 73.0   1st Qu.:190.0   1st Qu.:1.000   h:1302
##  Median :103.0   Median :240.5   Median :1.000
##  Mean   :109.6   Mean   :245.7   Mean   :1.493
##  3rd Qu.:138.0   3rd Qu.:298.0   3rd Qu.:2.000
##  Max.   :328.0   Max.   :520.0   Max.   :8.000
```

When the outcome of the games is visualized as barplots, it becomes clear that in both sports the most likely outcome is a home victory, though in soccer the bias towards a home victory appears to be bigger. As a consequence there is more data available for the model to train to predict home victories. This has to be taken into account later (see below).

```r
soccer %>% ggplot(aes(winner)) + geom_bar()
```

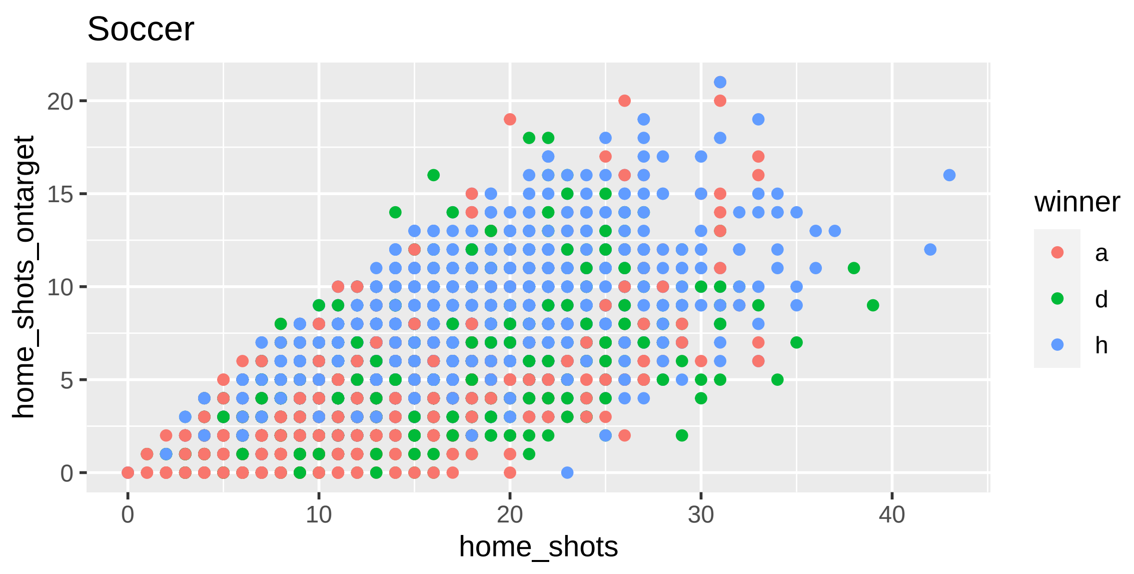


```r
football %>% ggplot(aes(winner)) + geom_bar()
```

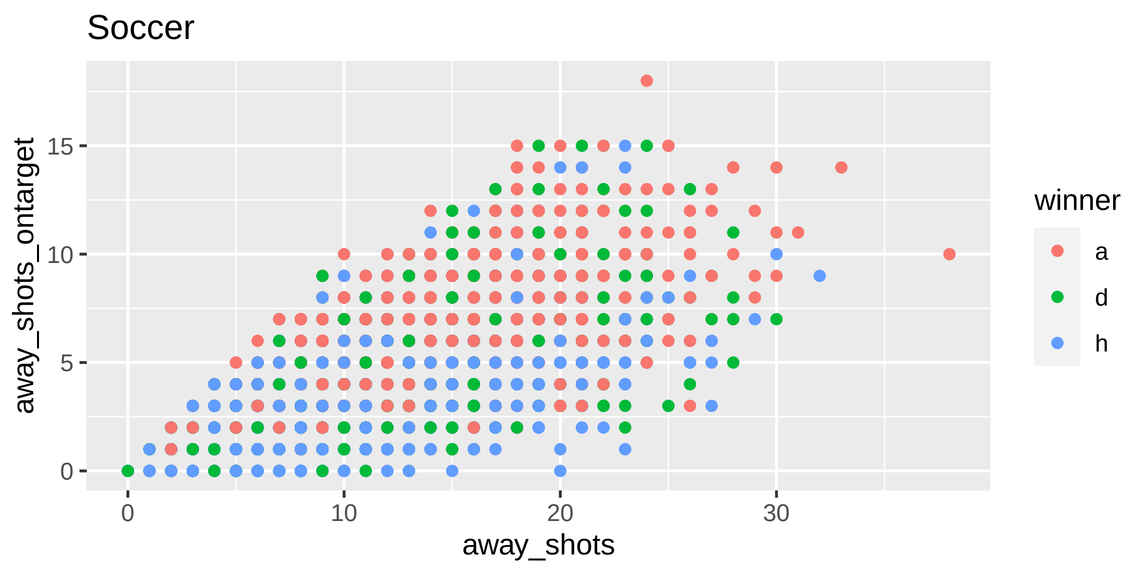


Related statistics such as shots and shots on target in soccer, as well as attempted rushings and rushing yards in football, are plotted against each other and colored by game outcome. Identical graphs are made for both, home team statistics as well as away team statistics. It can be seen that these predictors actually do appear to predict game outcome (same color tends to be in the same area of the plot). However, what becomes also immediately clear is that in football there appears to be a much clearer separation than in soccer. On the one hand, this might be due to the fact that there are no draws in football. Draws in general appear to be hard to predict, based of their distribution accross all of the shot-data. But on the other hand, this already gives a hint that the original hypothesis that football games are more predictable than soccer games might be true.

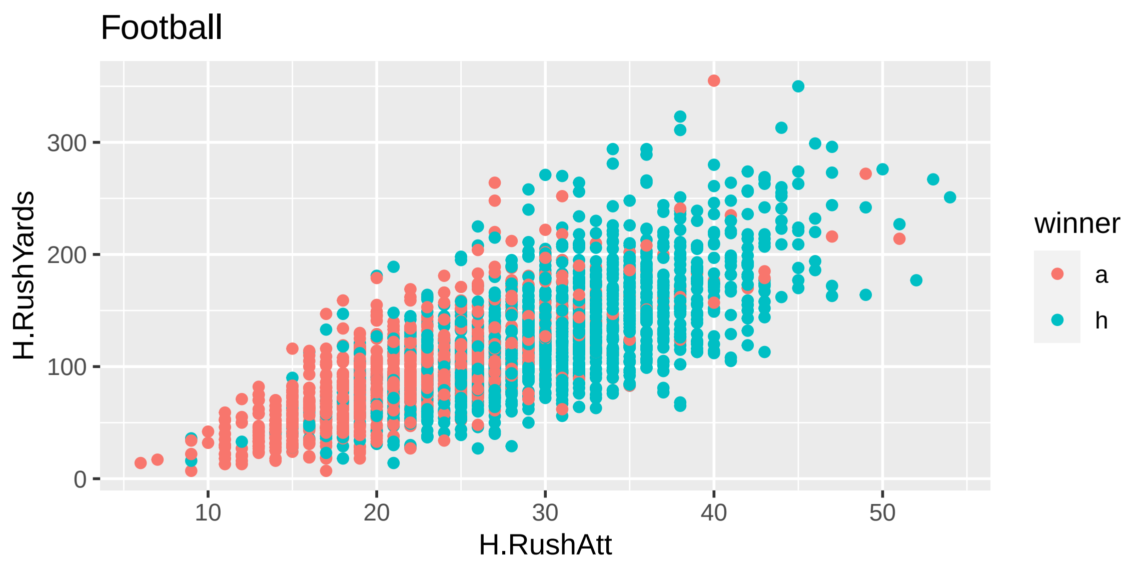
```
#soccer
soccer %>% ggplot(aes(home_shots, home_shots_ontarget, col=winner)) +
              geom_point() + ggtitle("Soccer")
```



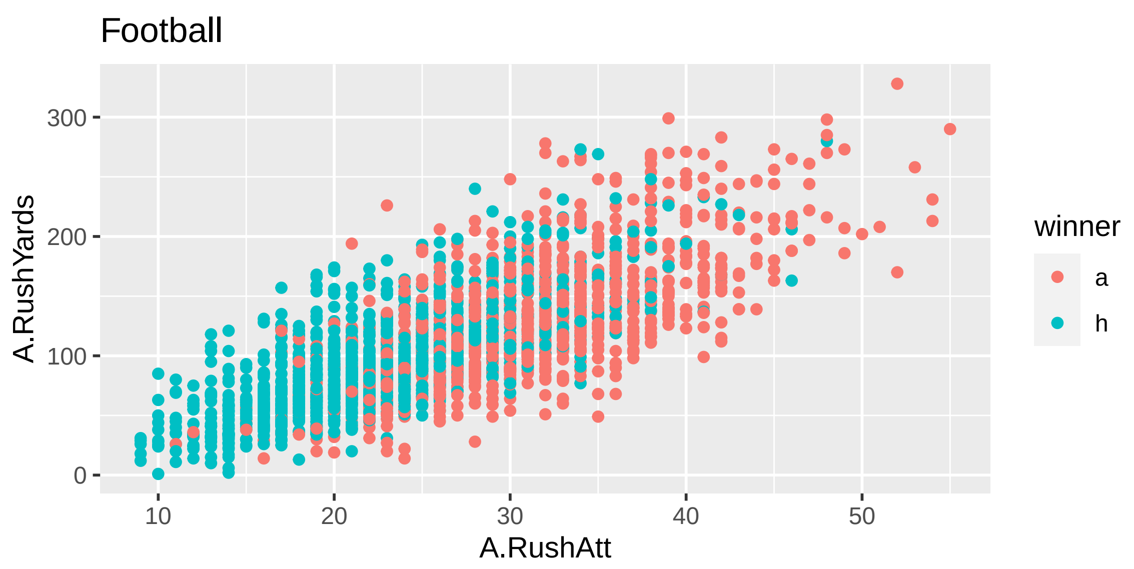
```r
soccer %>% ggplot(aes(away_shots, away_shots_ontarget, col=winner)) +
           geom_point() + ggtitle("Soccer")
```



```r
#football
football %>% ggplot(aes(H.RushAtt, H.RushYards, col=winner)) +
           geom_point() + ggtitle("Football")
```

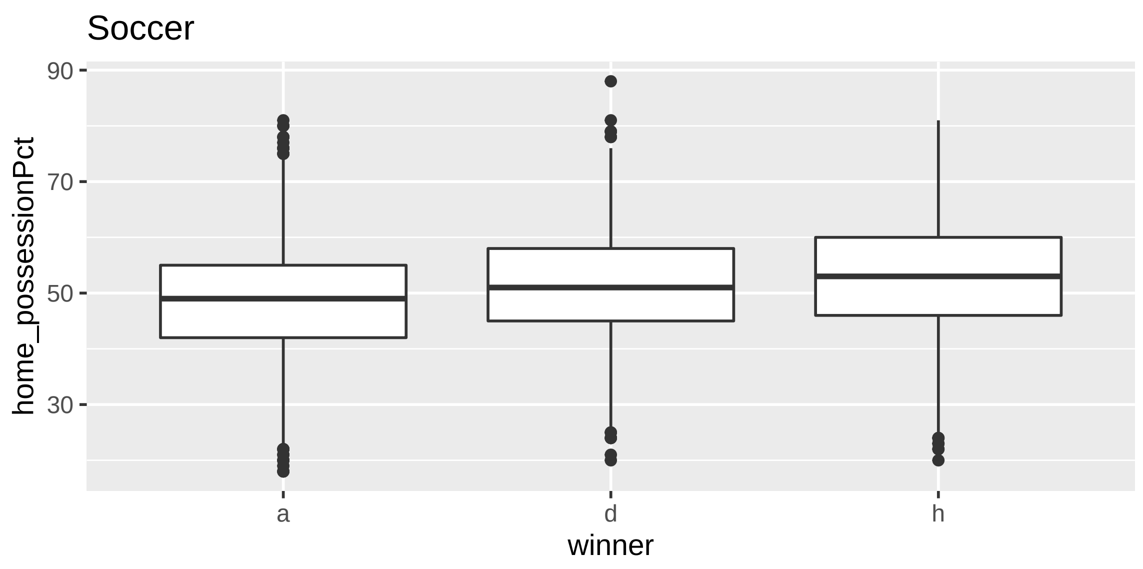


```r
football %>% ggplot(aes(A.RushAtt, A.RushYards, col=winner)) +
           geom_point() + ggtitle("Football")
```

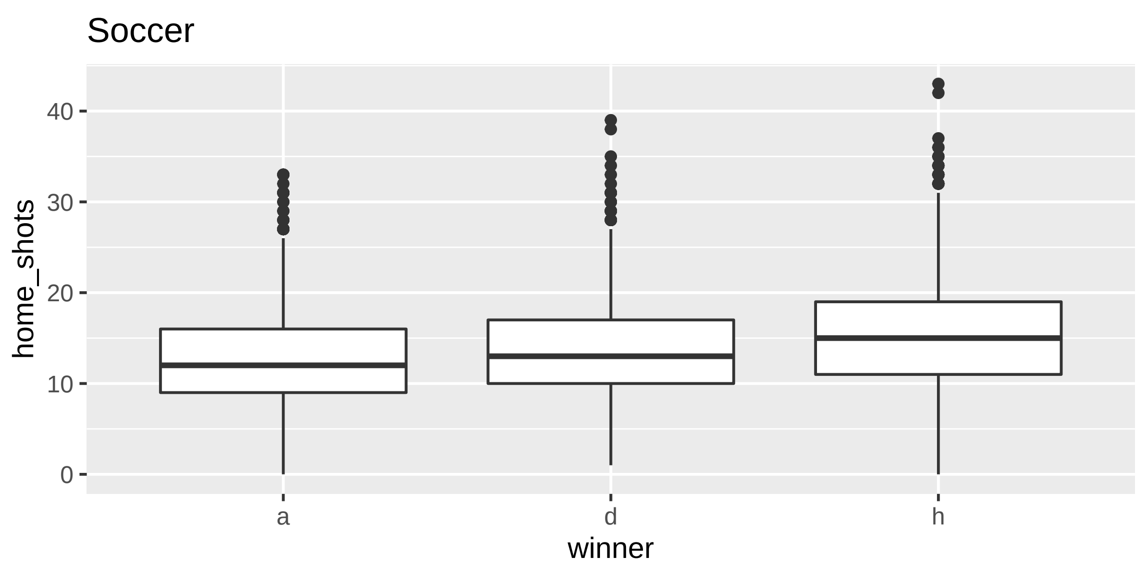


Boxplots for home-win and away-win (and draws in the case of football) for all of the chosen predictors support the same notion. While, at first sight, correlations between many of the chosen statistics with winning generally appear to be surprisingly small, there appears to be a tendency towards stronger correlations in football than in soccer (compare quartile separation within the boxplots below). Again, draws in soccer often appear to be highly similar to home wins with respect to many predictors and thus are expected to be very difficult to predict. In football, rushing appears to be much more predictive than passing, while turnovers obviously play an important role as well with a strong negative correlation to winning.

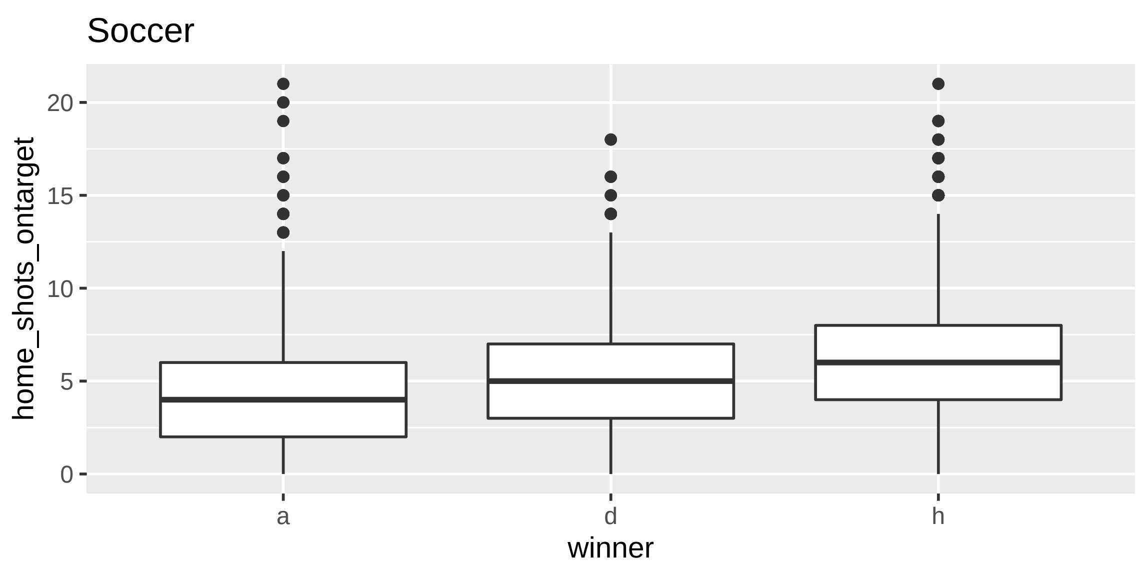
```
#soccer
soccer %>% ggplot(aes(winner, home_possessionPct)) + geom_boxplot() +
  ggtitle("Soccer")
```



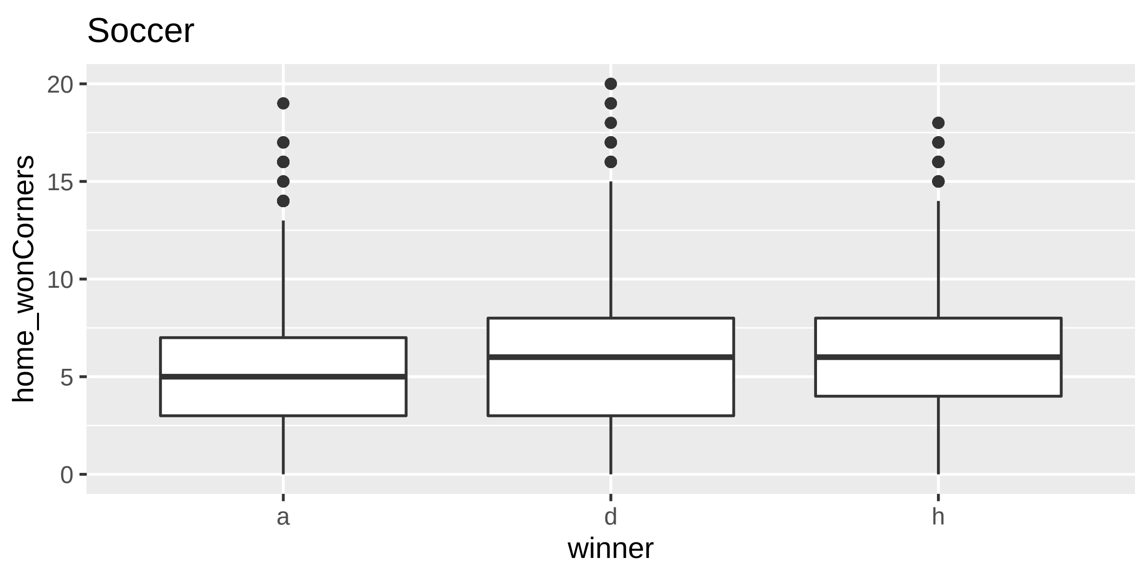
```
soccer %>% ggplot(aes(winner, home_shots)) + geom_boxplot() + ggtitle("Soccer")
```



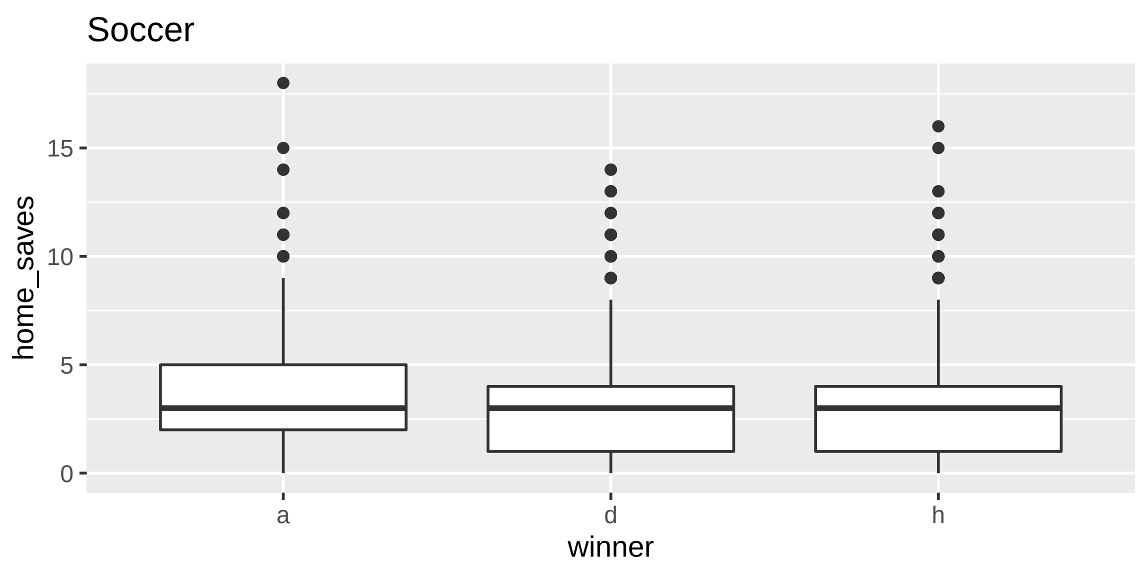
```r
soccer %>% ggplot(aes(winner, home_shots_ontarget)) + geom_boxplot() +
  ggtitle("Soccer")
```



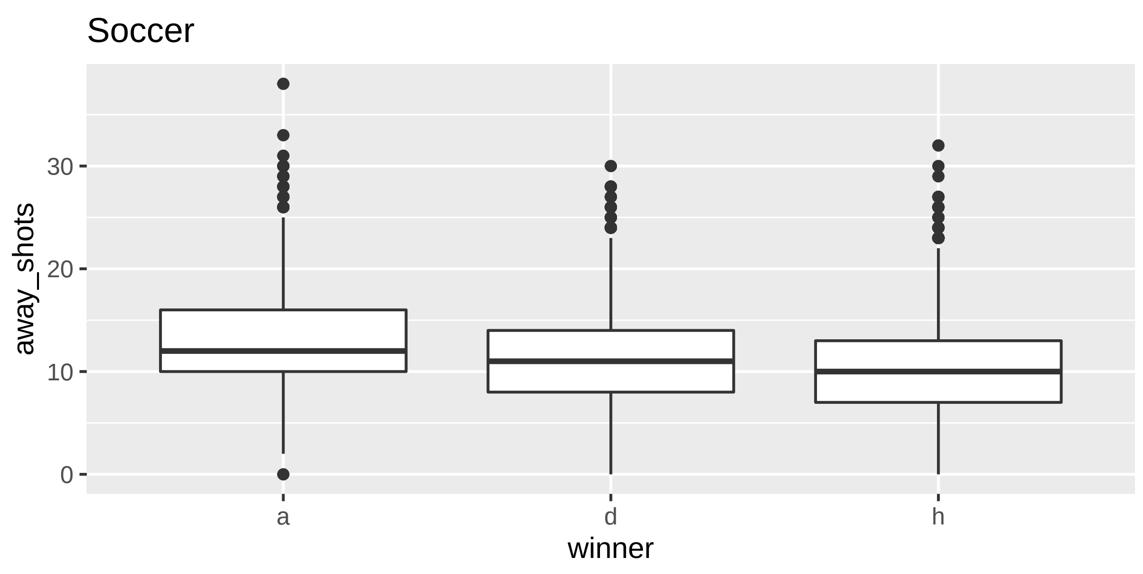
```r
soccer %>% ggplot(aes(winner, home_wonCorners)) + geom_boxplot() +
  ggtitle("Soccer")
```



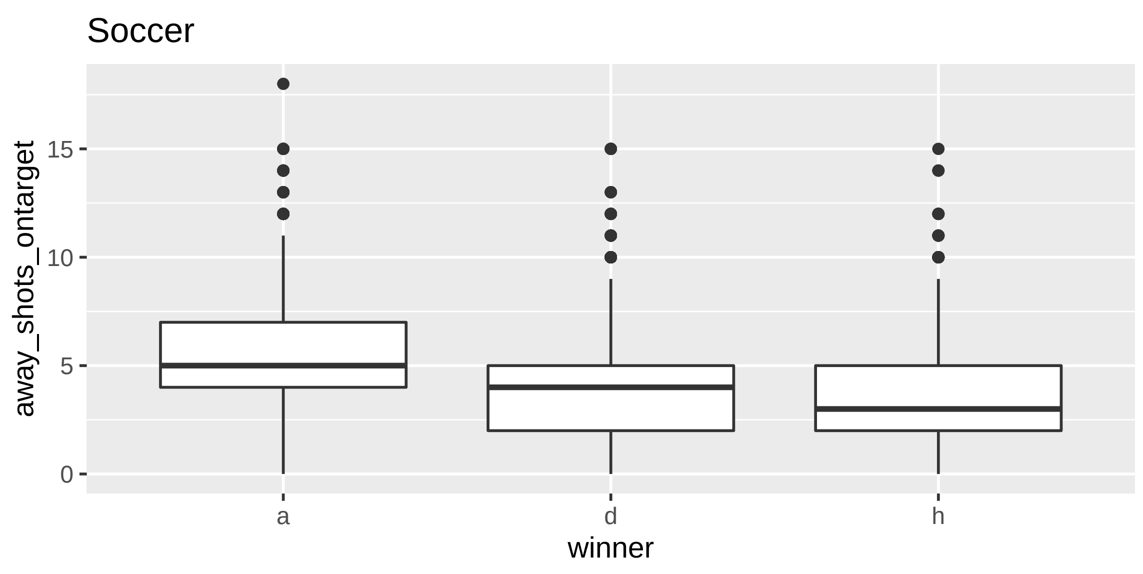
```
soccer %>% ggplot(aes(winner, home_saves)) + geom_boxplot() + ggtitle("Soccer")
```



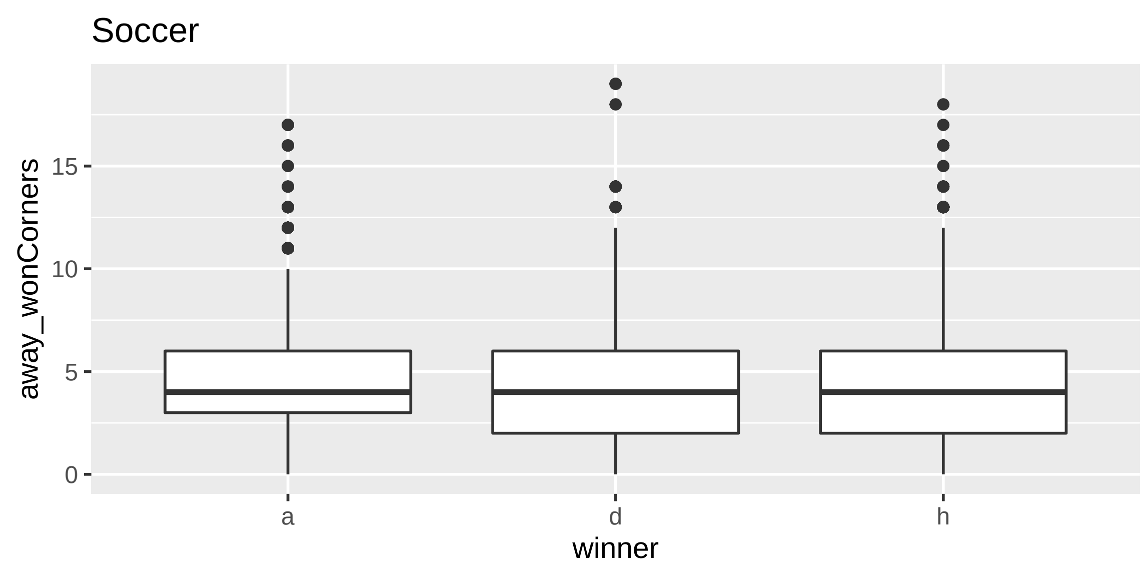
```
soccer %>% ggplot(aes(winner, away_shots)) + geom_boxplot() + ggtitle("Soccer")
```



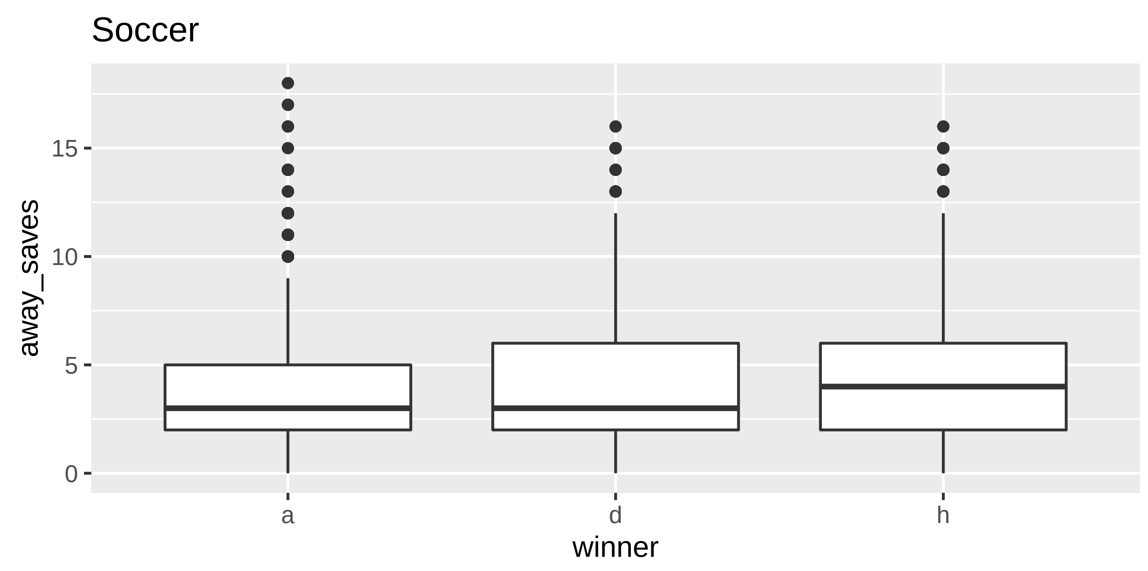
```
soccer %>% ggplot(aes(winner, away_shots_ontarget)) + geom_boxplot() +
  ggtitle("Soccer")
```



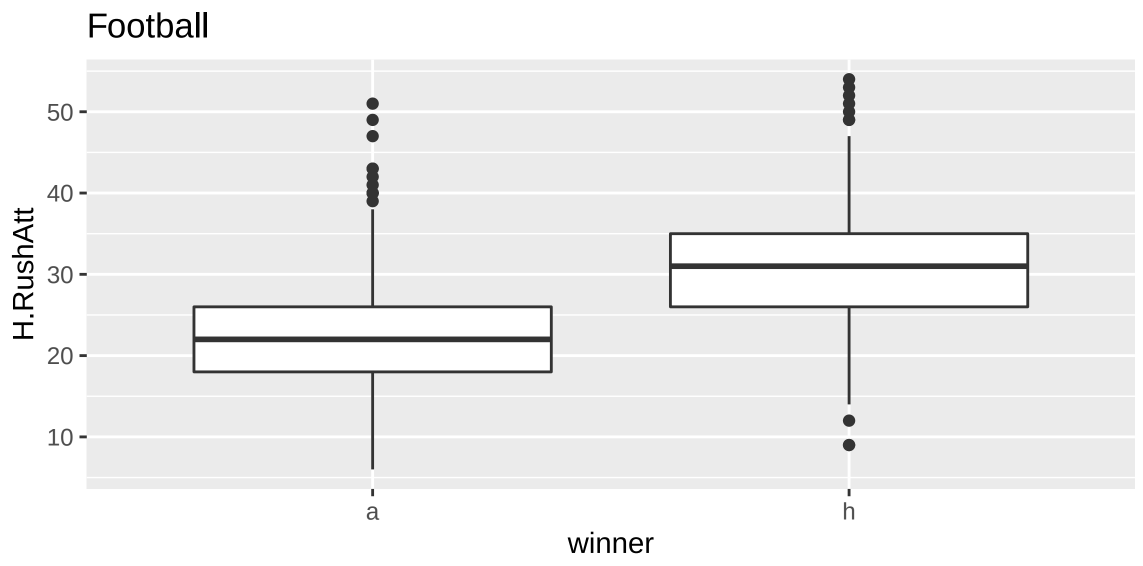
```
soccer %>% ggplot(aes(winner, away_wonCorners)) + geom_boxplot() +
  ggtitle("Soccer")
```



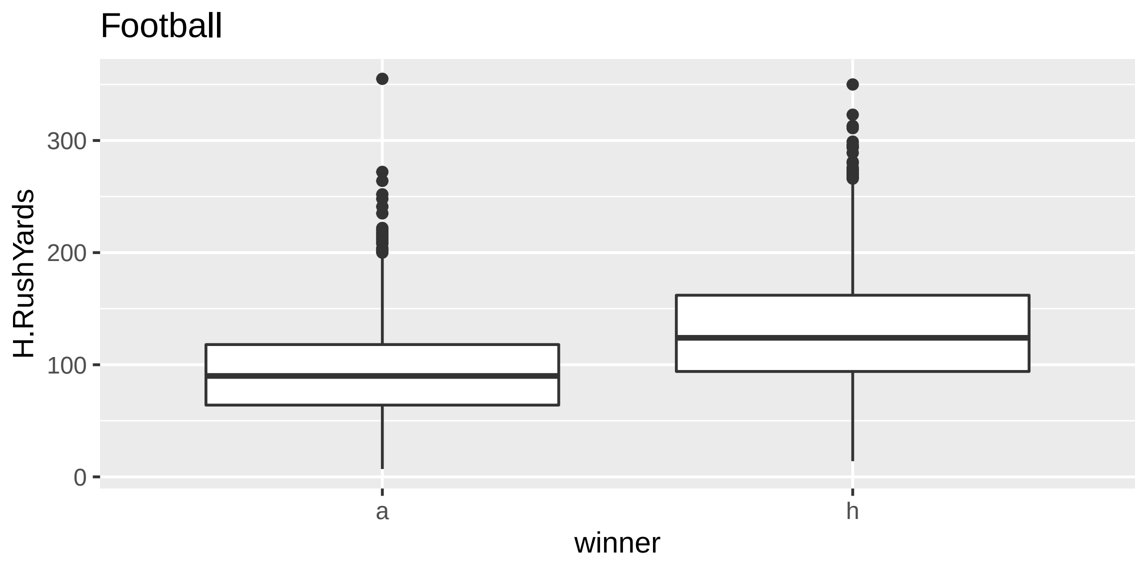
```r
soccer %>% ggplot(aes(winner, away_saves)) + geom_boxplot() + ggtitle("Soccer")
```



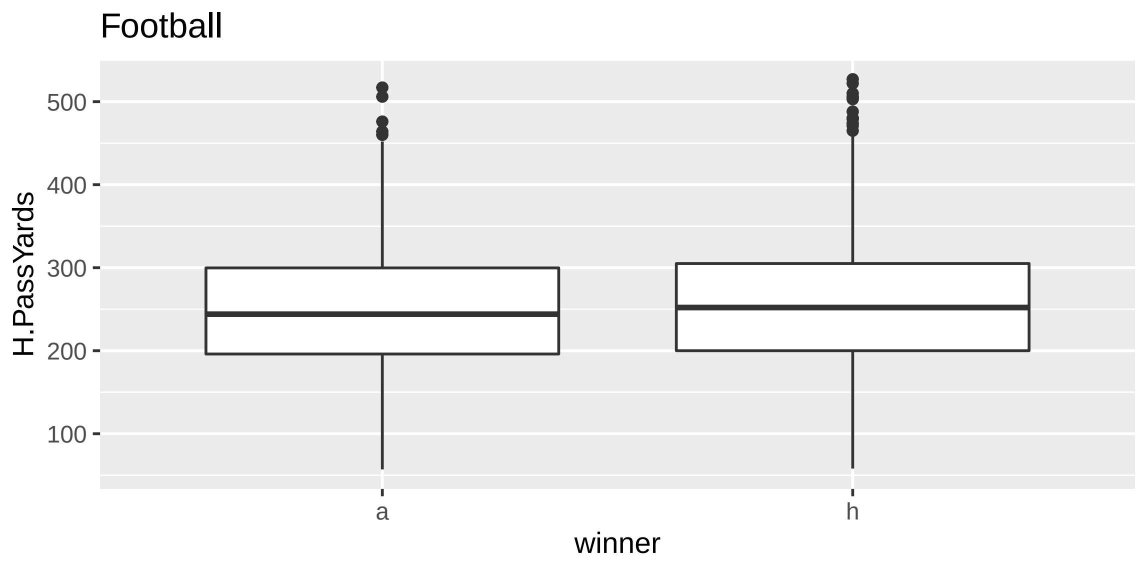
```r
#football
football %>% ggplot(aes(winner, H.RushAtt)) + geom_boxplot() +
  ggtitle("Football")
```



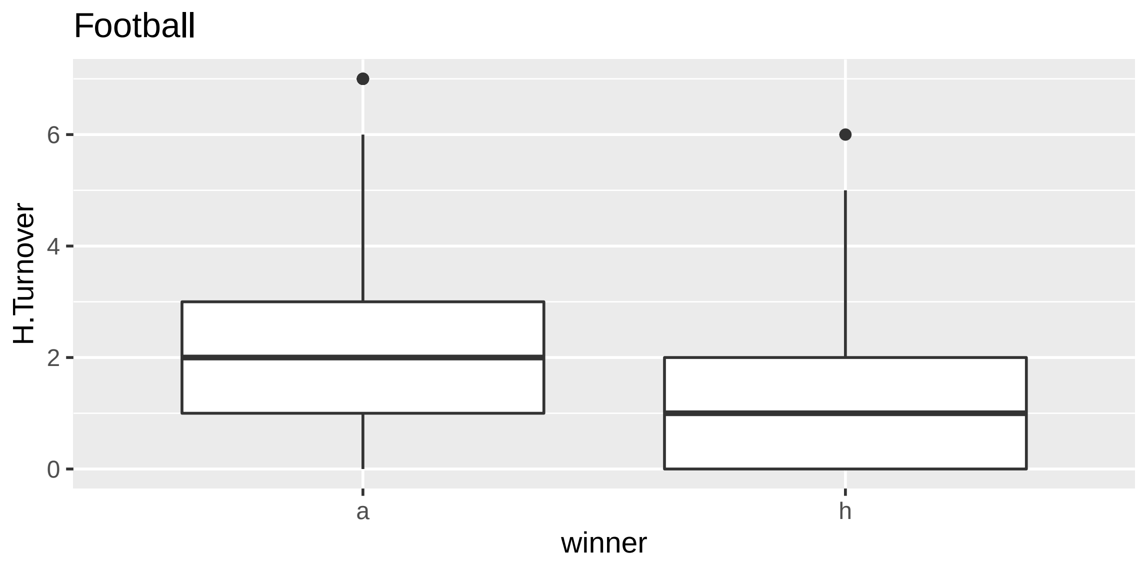
```
football %>% ggplot(aes(winner, H.RushYards)) + geom_boxplot() +
  ggtitle("Football")
```



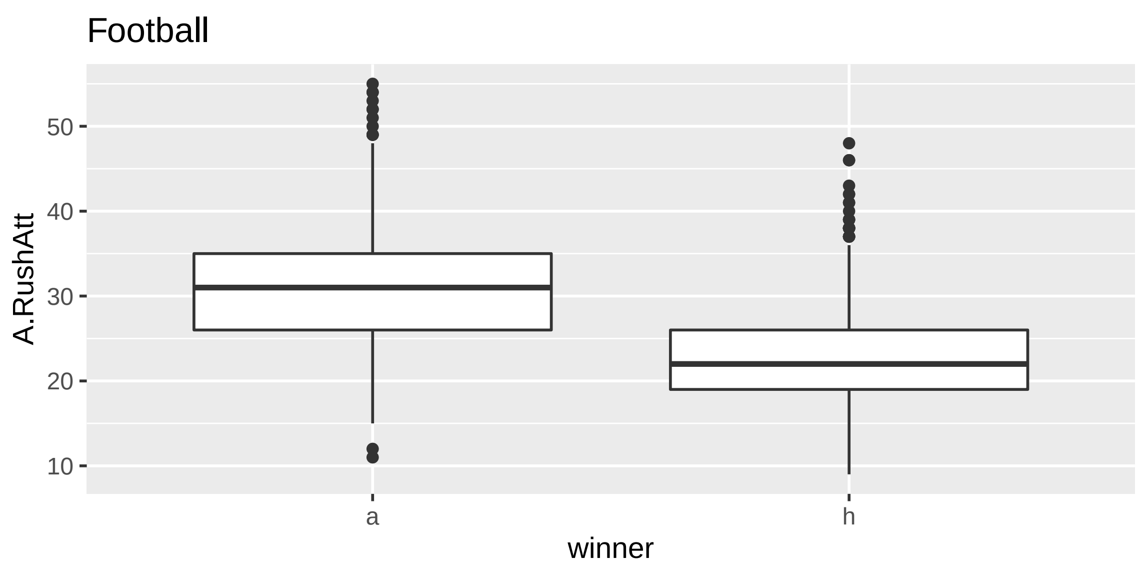
```
football %>% ggplot(aes(winner, H.PassYards)) + geom_boxplot() +
  ggtitle("Football")
```



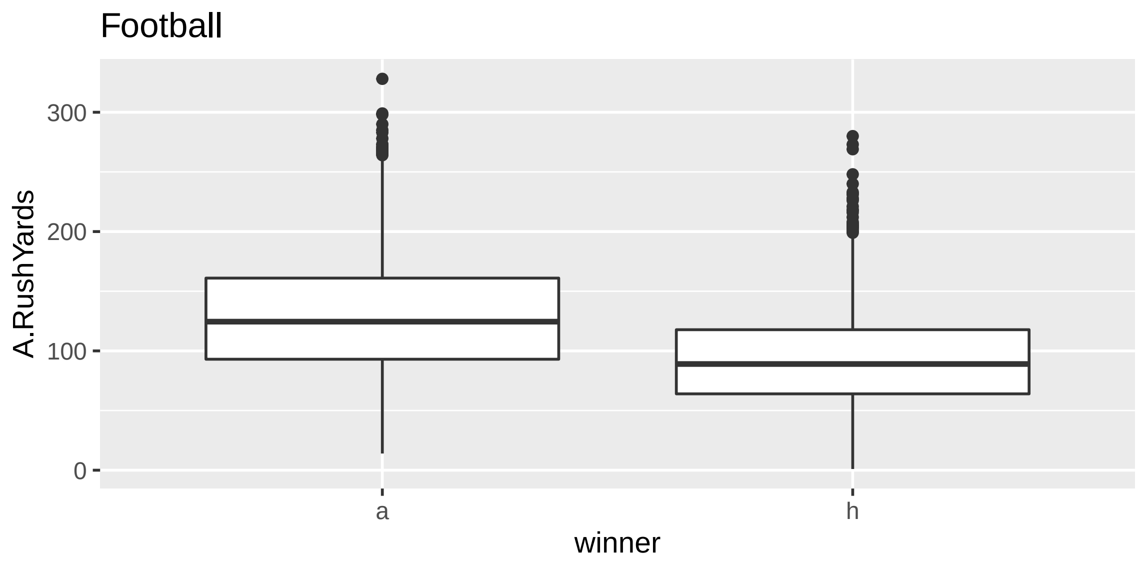
```
football %>% ggplot(aes(winner, H.Turnover)) + geom_boxplot() +
  ggtitle("Football")
```



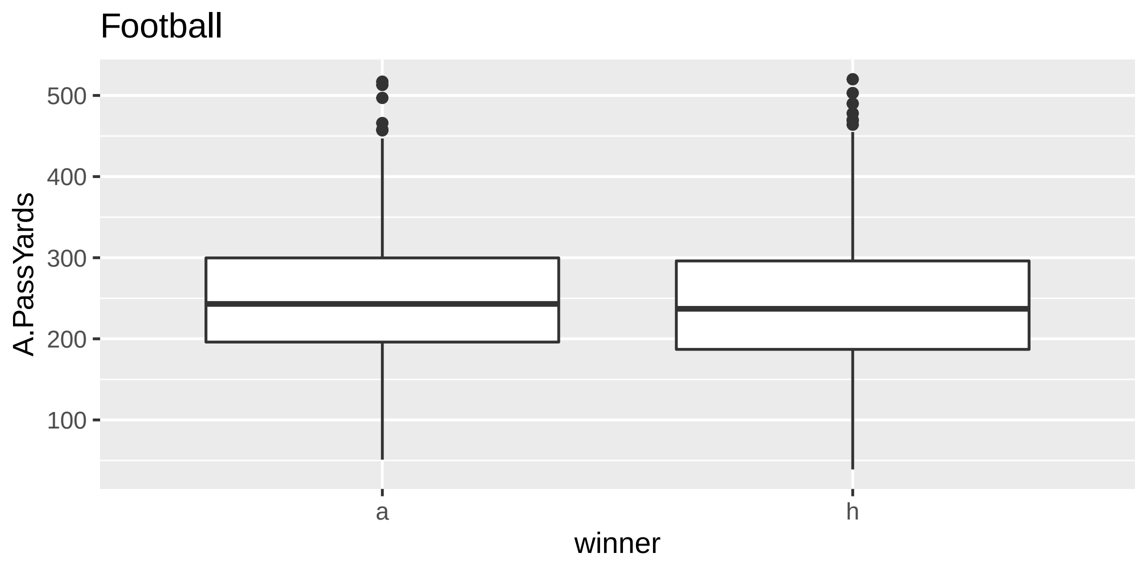
```
football %>% ggplot(aes(winner, A.RushAtt)) + geom_boxplot() +
  ggtitle("Football")
```



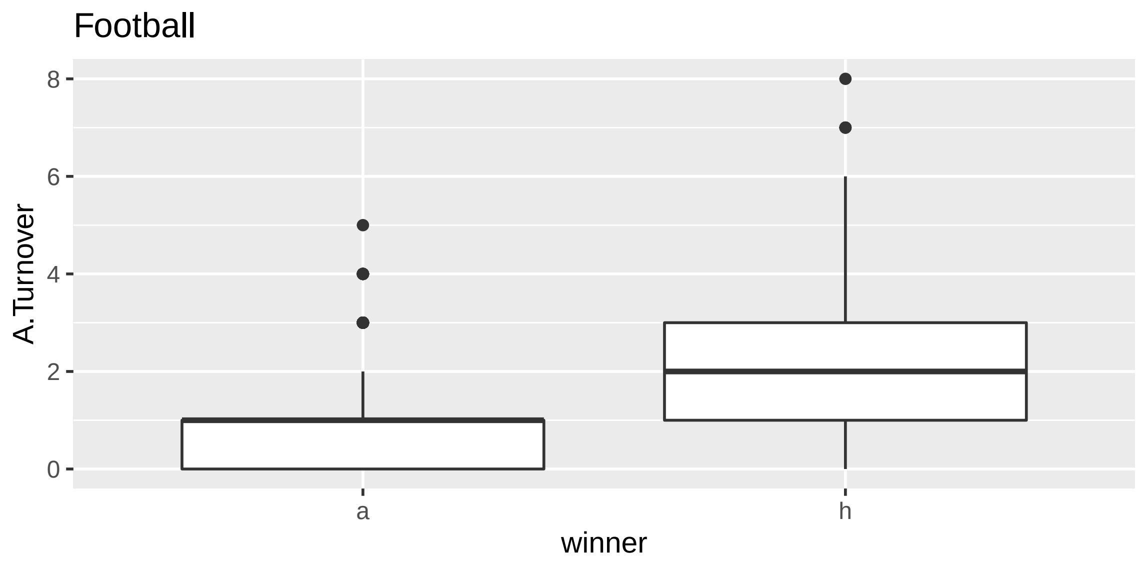
```
football %>% ggplot(aes(winner, A.RushYards)) + geom_boxplot() +
  ggtitle("Football")
```



```
football %>% ggplot(aes(winner, A.PassYards)) + geom_boxplot() +
  ggtitle("Football")
```



```
football %>% ggplot(aes(winner, A.Turnover)) + geom_boxplot() +
  ggtitle("Football")
```



Model Building

Soccer

For model building, the soccer dataset is again split into a train and a test set.

```
set.seed(13041984)
test_index <- createDataPartition(y = soccer$winner, times = 1, p = 0.1, list = FALSE)
train <- soccer[-test_index,]
test <- soccer[test_index,]
```

The resulting test set contains 189 away wins, 162 draws and 299 home wins. These numbers are important to keep in mind when interpreting confusion matrices later on.

```r
table(test$winner)
```

```
##
##   a   d   h
## 189 162 299
```

For model building accuracy of predictions was chosen as the decisive measure, because there is no preference whatsoever towards correctly predicting either of the possible game outcomes. For comparison's sake the accuracy of guessing a game outcome was calculated. Because the data (and the sport itself?) has a bias towards home wins, the accuracy was also calculated predicting every game as a home win. The results of these calculations are given below.

```r
# guess
random_result <- test %>% mutate(pred=sample(c("h","a","d"), nrow(test), replace=TRUE)) %>%
  summarize(accuracy=mean(winner==pred))

results <- data.frame(what="Soccer: random guessing", accuracy=random_result)

# all home wins
all_h <- test %>% mutate(pred=replicate(nrow(test), "h")) %>%
  summarize(accuracy=mean(winner==pred))
results <- bind_rows(results, data.frame(what="Soccer: All home wins", accuracs=all_h))
results
```

```
##                      what  accuracy
## 1 Soccer: random guessing 0.3492308
## 2   Soccer: All home wins 0.4600000
```

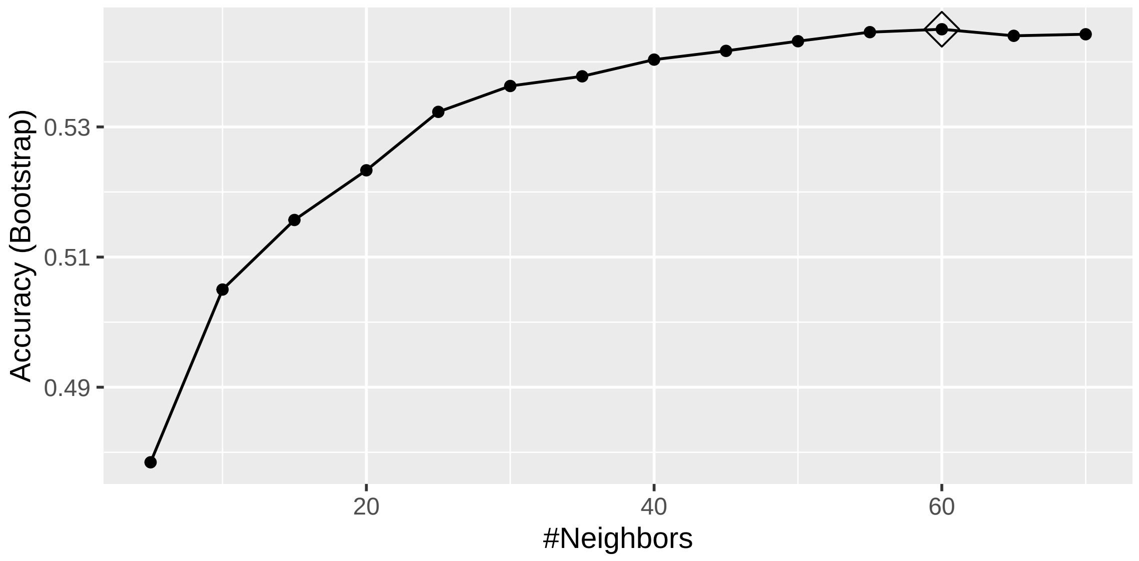
Next the caret package is used in order to apply several classification algorithms to the result prediction.

K Nearest Neighbours The first method tried is k nearest neighbours (KNN), where the outcome is decided based on the average properties of the neighboring data points. The number of neighbors used (k) is a tuning parameter. Here 9 values between 5 and 70 are examined. Model training is achieved using the caret package with the code below.

```r
fit_knn <- train(winner ~ .,  method = "knn",
                 tuneGrid = data.frame(k = seq(5, 70, 5)), data = train)
```

As can be seen in the following plot, an optimal accuracy is obtained when using a k value of 60.

```r
ggplot(fit_knn, highlight=TRUE)
```



When the obtained model is applied on the test set, the following accuracy and confusion matrix is obtained.

```
knn_result <- test %>% mutate(pred=predict(fit_knn, test)) %>% summarize(accuracy=mean(winner==pred))
results<- bind_rows(results, data.frame(what="Soccer: KNN", accuracy=knn_result))
results
```

```
##                       what  accuracy
## 1 Soccer: random guessing 0.3492308
## 2   Soccer: All home wins 0.4600000
## 3             Soccer: KNN 0.5276923
```

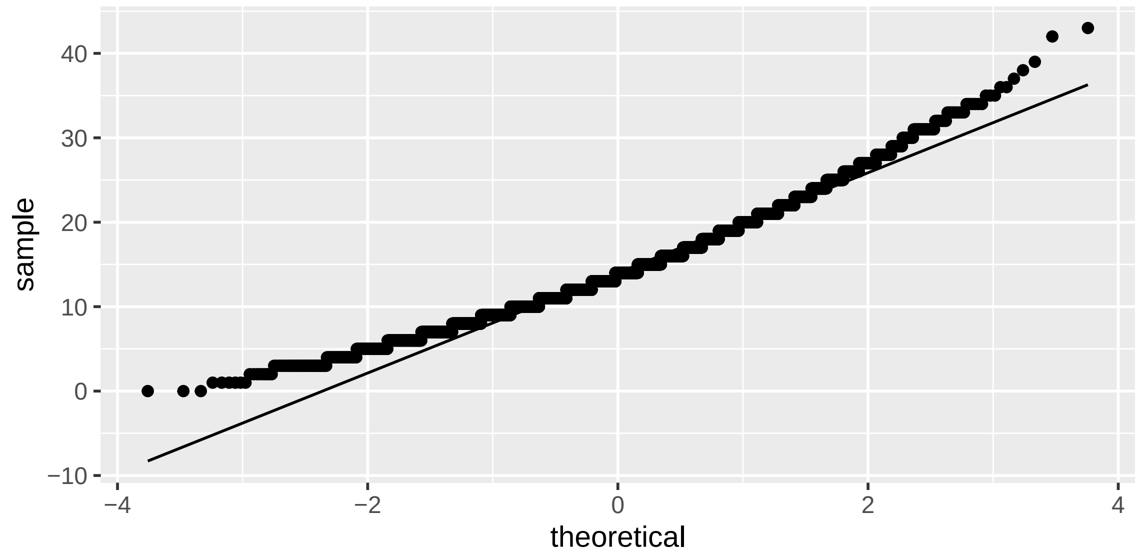
```
confusionMatrix(predict(fit_knn, test), test$winner)[2]
```

```
## $table
##           Reference
## Prediction   a   d   h
##          a  83  48  43
##          d   7   9   8
##          h  99 105 248
```

While the obtained accuracy is somewhat better than just guessing home-win all the time, the improvement is only small (from 46% to 53%). A look at the confusion matrix shows that the model predicted 452 times a home win. Unsurprisingly, 248 of the 299 home victories were predicted correctly. As expected, predicting draws turns out to be particularly challenging and the model only called 24 draws and only 9 of 162 correctly.

Linear and Quadratic Discriminant Analysis (LDA and QDA) Next LDA and QDA were applied to the problem of predicting game outcomes. With these methods boundaries between classifiers are calculated with allow to decide where a particular point belongs. Strictly speaking, for these methods to work, data needs to be normally distributed. A qq-plot on home shots demonstrates that while the normal distribution is not perfect, the data appears to be somewhat normally distributed.

```r
train %>% ggplot(aes(sample=home_shots)) + geom_qq() + geom_qq_line()
```



The model training and the obtained results are given below.

```r
#LDA
fit_lda <- train(winner ~ ., method = "lda", data = train)
lda_result <- test %>% mutate(pred=predict(fit_lda, test)) %>%
  summarize(accuracy=mean(winner==pred))
results <- bind_rows(results, data.frame(what="Soccer: LDA", accuracy=lda_result))
confusionMatrix(predict(fit_lda, test), test$winner)[2]
```

```
## $table
##           Reference
## Prediction   a   d   h
##          a 138  35  20
##          d  27  32  13
##          h  24  95 266
```

```r
#QDA
fit_qda <- train(winner ~ ., method = "qda", data = train)
qda_result <- test %>% mutate(pred=predict(fit_qda, test)) %>%
  summarize(accuracy=mean(winner==pred))
results <- bind_rows(results, data.frame(what="Soccer: QDA", accuracy=qda_result))
results
```

```
##                      what  accuracy
## 1 Soccer: random guessing 0.3492308
## 2   Soccer: All home wins 0.4600000
## 3             Soccer: KNN 0.5276923
## 4             Soccer: LDA 0.6707692
## 5             Soccer: QDA 0.6769231
```

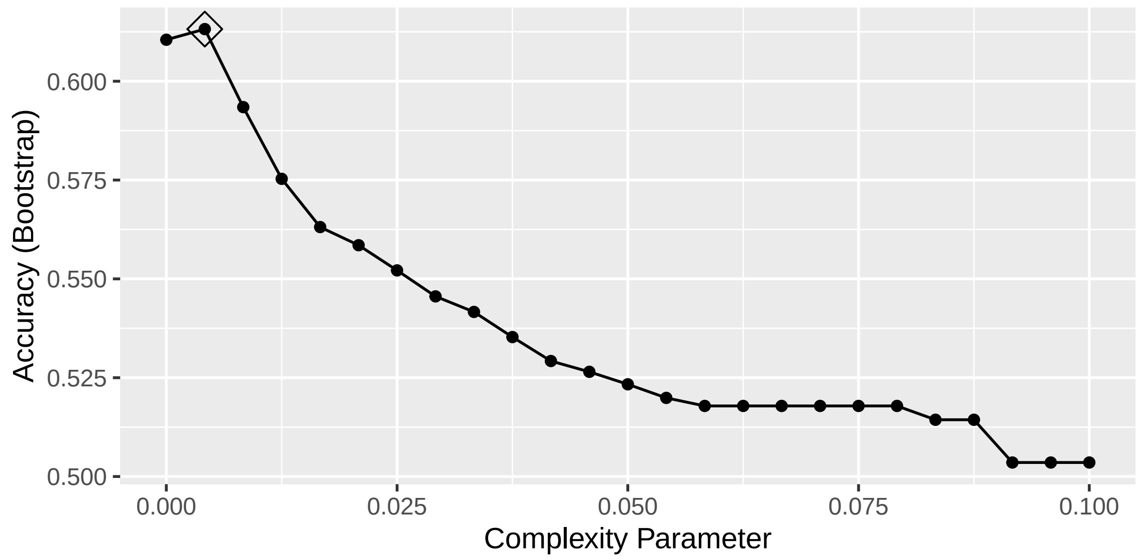
```r
confusionMatrix(predict(fit_qda, test), test$winner)[2]
```

```
## $table
##           Reference
## Prediction   a   d   h
##          a 130  25  19
##          d  31  53  23
##          h  28  84 257
```

Both LDA as well as QDA show a markedly improved accuracy of match outcome predictions (67% and 68% respectively). Interestingly compared to KNN they are both much better in correctly predicting away wins getting 138 and 130 right out of the 189. Still predicting a draw is challenging even though especially QDA does a much better job than KNN (predicting 58 of the 162 draws right).

Decision Tree Next an algorythm building a decision tree for classification is examined. Again there is a tuning parameter in this method, the complexity parameter which is the minimum improvement necessary for the model in order to create another decision node. The results of these calculations is given below.

```r
# rpart
fit_rpart <- train(winner ~ ., method = "rpart",
                     tuneGrid = data.frame(cp = seq(0.0, 0.1, len = 25)),
                     data = train)
ggplot(fit_rpart, highlight = TRUE)
```



```r
rpart_result <- test %>% mutate(pred=predict(fit_rpart, test)) %>%
  summarize(accuracy=mean(winner==pred))
results <- bind_rows(results, data.frame(what="Soccer: rpart", accuracy=rpart_result))
results
```

```
##                        what  accuracy
## 1 Soccer: random guessing 0.3492308
## 2   Soccer: All home wins 0.4600000
## 3             Soccer: KNN 0.5276923
## 4             Soccer: LDA 0.6707692
## 5             Soccer: QDA 0.6769231
## 6           Soccer: rpart 0.5892308
```

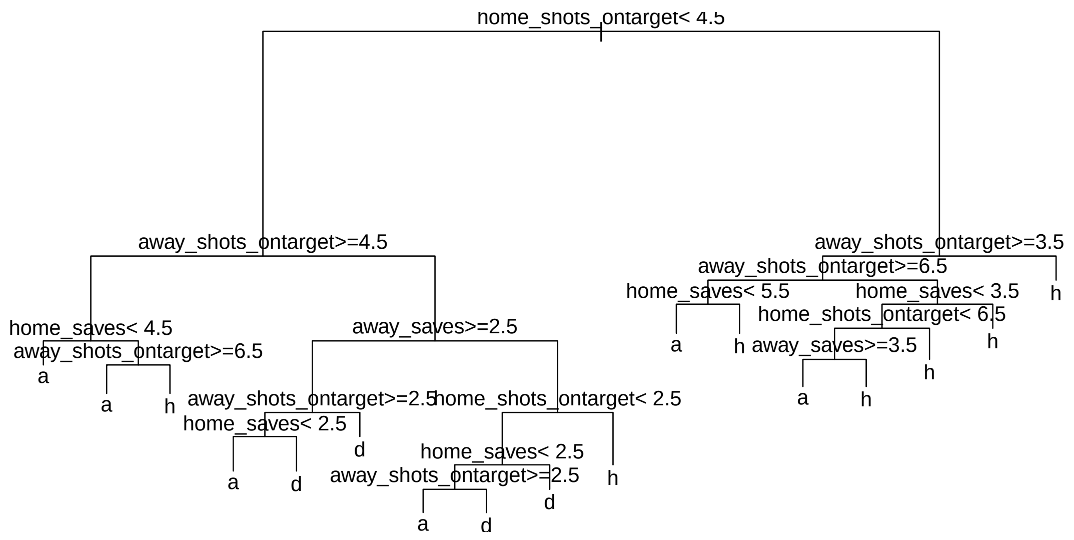
```
confusionMatrix(predict(fit_rpart, test), test$winner)[2]
```

```
## $table
##           Reference
## Prediction   a   d   h
##          a 107  32  31
##          d  11  34  26
##          h  71  96 242
```

The best model is obtained with a complexity parameter 0.0042 giving a model predicting the game outcome with an accuracy of 59%. This is better than KNN but distinctly lower than both LDA and QDA. Again the draws turn out to be particularly difficult to predict.

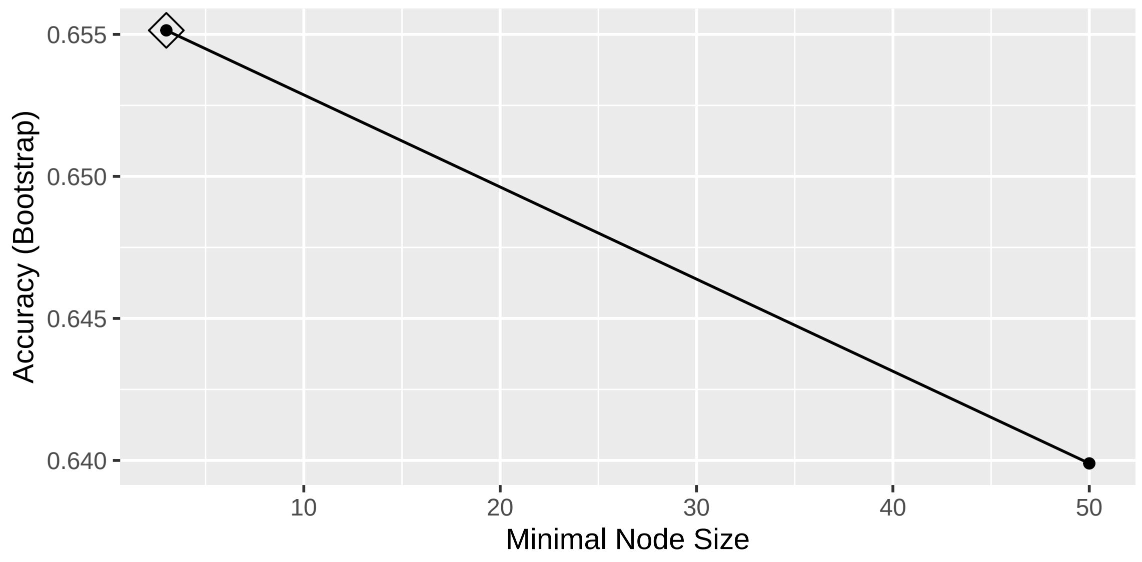
One advantage of a decision tree is that its outcome can be easily interpreted. The tree obtained above is depicted below. Highly interestingly, a game outcome appears to be mainly depending on which team manages to bring most shots on goal, followed by the performance of the goalkeeper (i.e. how many of those shots are saved).

```
plot(fit_rpart$finalModel)
text(fit_rpart$finalModel)
```



Random Forest Finally, a random forest algorythm is to be examined. Random forests are bundles of decision trees that are used to classify data points. Due to the long computation time of this method, only two minimal nodesizes are examined, a small one at 3 and a large one at 50, with the one at 3 leading to a better model (see plot below).

```r
#random forest
fit_randomforest <- train(winner ~ ., method = "Rborist",
                    tuneGrid = data.frame(predFixed = 2, minNode = c(3, 50)),
                    data = train)

ggplot(fit_randomforest, highlight=TRUE)
```



```r
randomforest_result <- test %>% mutate(pred=predict(fit_randomforest, test)) %>% summarize(accuracy=mean
results <- bind_rows(results, data.frame(what="Soccer: Random Forest", accuracy=randomforest_result))
results
```

```
##                        what  accuracy
## 1 Soccer: random guessing 0.3492308
## 2   Soccer: All home wins 0.4600000
## 3             Soccer: KNN 0.5276923
## 4             Soccer: LDA 0.6707692
## 5             Soccer: QDA 0.6769231
## 6           Soccer: rpart 0.5892308
## 7   Soccer: Random Forest 0.6723077
```

```r
confusionMatrix(predict(fit_randomforest, test), test$winner)[2]
```

```
## $table
##           Reference
## Prediction   a   d   h
##          a 135  43  23
##          d  28  50  24
##          h  26  69 252
```

Random forest gives a model predicting the game outcomes with an accuracy of 67%, peing on par with LDA and QDA. Also the confusion matrix is very similar to the one obtained using QDA, including a relatively good result in predicting draws.

Ensemble The combination of several methods for improving predictions is called ensembling. Here it is for example possible to cobine the predictions of the three best methods (LDA, QDA and RF) for deciding if a game should be classified as home win, away win or draw. The corresponding result is obtained with the code below.

```
# Ensemble
ensemble <- data.frame(QDA= predict(fit_qda, test),
                       LDA= predict(fit_lda, test),
                       RF = predict(fit_randomforest, test))

read_out <- function(x){names(which.max(table(t(ensemble[x,]))))}
ensemble_pred <- sapply(seq(1,nrow(ensemble)), read_out)
ensemble_result <- test %>% cbind(ensemble_pred) %>% summarize(accuracy=mean(winner==ensemble_pred))
results <- bind_rows(results, data.frame(what="Soccer: Ensemble", accuracy=ensemble_result))
results
```

```
##                       what  accuracy
## 1 Soccer: random guessing 0.3492308
## 2   Soccer: All home wins 0.4600000
## 3             Soccer: KNN 0.5276923
## 4             Soccer: LDA 0.6707692
## 5             Soccer: QDA 0.6769231
## 6           Soccer: rpart 0.5892308
## 7   Soccer: Random Forest 0.6723077
## 8        Soccer: Ensemble 0.6784615
```

```
confusionMatrix(as.factor(ensemble_pred), test$winner)[2]
```

```
## $table
##           Reference
## Prediction   a   d   h
##          a 142  37  22
##          d  24  36  14
##          h  23  89 263
```

Indeed, the resulting accuracy is slightly better than all of the combined ones - albeit only by a negligible margin. Additionally, while the predictions of home victories and away victories is good, the ensemble performs again much worse than QDA and Random Forest, when it comes to predicting draws.

Football

All the above steps are repeated for the football dataset.

```
# Creating train and test set for model creation
set.seed(020816)
test_index <- createDataPartition(y = football$winner, times = 1, p = 0.1, list = FALSE)
train_nfl <- football[-test_index,]
test_nfl <- football[test_index,]

#MODEL BUILDING
# guess
random_result_nfl <- test_nfl %>%
```

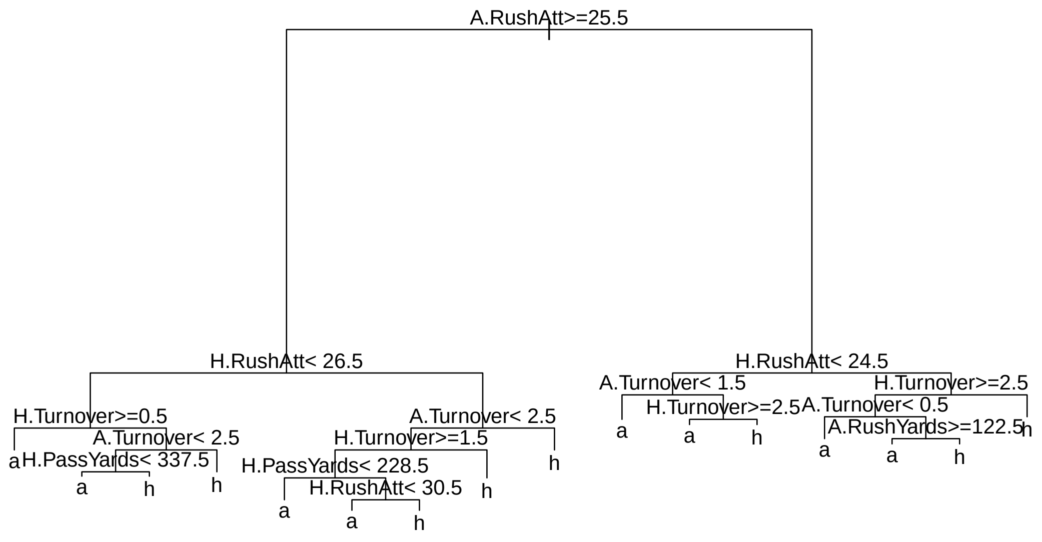
```r
  mutate(pred=sample(c("h","a","d"), nrow(test_nfl), replace=TRUE)) %>%
  summarize(accuracy=mean(winner==pred))
results <- bind_rows(results, data.frame(what="Footall: random guessing", accuracy=random_result_nfl))

# all home wins
all_h_nfl <- test_nfl %>% mutate(pred=replicate(nrow(test_nfl), "h")) %>%
  summarize(accuracy=mean(winner==pred))
results <- bind_rows(results, data.frame(what="Football: All home wins", accuracy=all_h_nfl))

# knn
fit_knn_nfl <- train(winner ~ .,  method = "knn",
                tuneGrid = data.frame(k = seq(1, 100, 5)),
                data = train_nfl)
knn_result_nfl <- test_nfl %>% mutate(pred=predict(fit_knn_nfl, test_nfl)) %>%
  summarize(accuracy=mean(winner==pred))
results<- bind_rows(results, data.frame(what="Football: KNN", accuracy=knn_result_nfl))

#LDA
fit_lda_nfl <- train(winner ~ ., method = "lda", data = train_nfl)
lda_result_nfl <- test_nfl %>% mutate(pred=predict(fit_lda_nfl, test_nfl)) %>%
  summarize(accuracy=mean(winner==pred))
results <- bind_rows(results, data.frame(what="Football: LDA", accuracy=lda_result_nfl))

#QDA
fit_qda_nfl <- train(winner ~ ., method = "qda", data = train_nfl)
qda_result_nfl <- test_nfl %>% mutate(pred=predict(fit_qda_nfl, test_nfl)) %>%
  summarize(accuracy=mean(winner==pred))
results <- bind_rows(results, data.frame(what="Football: QDA", accuracy=qda_result_nfl))

# rpart
fit_rpart_nfl <- train(winner ~ ., method = "rpart",
                  tuneGrid = data.frame(cp = seq(0.0, 0.1, len = 25)),
                  data = train_nfl)
rpart_result_nfl <- test_nfl %>% mutate(pred=predict(fit_rpart_nfl, test_nfl)) %>%
  summarize(accuracy=mean(winner==pred))
results <- bind_rows(results, data.frame(what="Football: rpart", accuracy=rpart_result_nfl))

# random forest

fit_randomforest_nfl <- train(winner ~ .,
                          method = "Rborist",
                          tuneGrid = data.frame(predFixed = 2, minNode = c(3, 50)),
                          data = train_nfl)
randomforest_result_nfl <- test_nfl %>%
  mutate(pred=predict(fit_randomforest_nfl, test_nfl)) %>%
  summarize(accuracy=mean(winner==pred))
results <- bind_rows(results, data.frame(what="Football: Random Forest", accuracy=randomforest_result_n

# Ensemble

ensemble_nfl <- data.frame(QDA= predict(fit_qda_nfl, test_nfl),
                           LDA= predict(fit_lda_nfl, test_nfl),
                           RF = predict(fit_randomforest_nfl, test_nfl))
```

```
read_out <- function(x){names(which.max(table(t(ensemble_nfl[x,]))))}
ensemble_pred_nfl <- sapply(seq(1,nrow(ensemble_nfl)), read_out)
ensemble_result_nfl <- test_nfl %>% cbind(ensemble_pred_nfl) %>%
  summarize(accuracy=mean(winner==ensemble_pred_nfl))
results <- bind_rows(results, data.frame(what="Football: Ensemble", accuracy=ensemble_result_nfl))
results
```

```
##                        what  accuracy
## 1   Soccer: random guessing 0.3492308
## 2     Soccer: All home wins 0.4600000
## 3               Soccer: KNN 0.5276923
## 4               Soccer: LDA 0.6707692
## 5               Soccer: QDA 0.6769231
## 6             Soccer: rpart 0.5892308
## 7     Soccer: Random Forest 0.6723077
## 8          Soccer: Ensemble 0.6784615
## 9  Footall: random guessing 0.2987013
## 10  Football: All home wins 0.5670996
## 11            Football: KNN 0.6969697
## 12            Football: LDA 0.8528139
## 13            Football: QDA 0.8398268
## 14          Football: rpart 0.8225108
## 15  Football: Random Forest 0.8398268
## 16       Football: Ensemble 0.8484848
```

First of all, all the examined methods score higher accuracy in predicting match outcomes in football compared to soccer. Again, the most successful methods are LDA, QDA and Random Forest, albeit the decision tree also performes very well here. The corresponding decision tree is given below. As hypothesized in the data analysis part (see above), results appear to be relatively well predicted by rushing attempts and turnovers.

```
plot(fit_rpart_nfl$finalModel)
text(fit_rpart_nfl$finalModel)
```



Interestingly, LDA alone performes slightly better than the Ensemble, albeit only by a small margin.

```
#confusion matrix KNN
confusionMatrix(predict(fit_knn_nfl, test_nfl), test_nfl$winner)[2]
```

```
## $table
##           Reference
## Prediction   a   h
##          a  60  31
##          h  40 100
```

```
#confusion matrix LDA
confusionMatrix(predict(fit_lda_nfl, test_nfl), test_nfl$winner)[2]
```

```
## $table
##           Reference
## Prediction   a   h
##          a  82  16
##          h  18 115
```

```
#confusion matrix QDA
confusionMatrix(predict(fit_qda_nfl, test_nfl), test_nfl$winner)[2]
```

```
## $table
##           Reference
## Prediction   a   h
##          a  81  18
##          h  19 113
```

```
#confusion matrix decision tree
confusionMatrix(predict(fit_rpart_nfl, test_nfl), test_nfl$winner)[2]
```

```
## $table
##           Reference
## Prediction   a   h
##          a  86  27
##          h  14 104
```

```
#confusion matrix random forest
confusionMatrix(predict(fit_randomforest_nfl, test_nfl), test_nfl$winner)[2]
```

```
## $table
##           Reference
## Prediction   a   h
##          a  85  22
##          h  15 109
```

```
#confusion matrix ensemble
confusionMatrix(as.factor(ensemble_pred_nfl), test_nfl$winner)[2]
```

```
## $table
##           Reference
## Prediction   a   h
##          a  83  18
##          h  17 113
```

Confusion matrices demonstrate that there is no bias towards predicting away win or home win with either method.

Tuning parameters are slightly different for football than for soccer with respect to the numbers of neighbors in KNN and the complexity parameter in the decision tree method.

```
bestTune <- data.frame(KNN_k=fit_knn_nfl$bestTune$k,
                       rpart_cp=fit_rpart_nfl$bestTune$cp,
                       RF_minNode=fit_randomforest_nfl$bestTune$minNode)
bestTune
```

```
##   KNN_k    rpart_cp RF_minNode
## 1    66 0.004166667          3
```

Results

After having generated models for both predicting football results as well as soccer results, in this section the final prediction results are obtained using the two validation data sets. Therefore, both datasets are used to predict the winner of soccer and football games using for each sport an ensemble of the LDA, QDA and Random Forest models created before. This is achieved using the code below.

```
ensemble_final <- data.frame(QDA= predict(fit_qda, soccer_validation),
                          LDA= predict(fit_lda, soccer_validation),
                          RF = predict(fit_randomforest, soccer_validation))
read_out <- function(x){names(which.max(table(t(ensemble_final[x,]))))}
ensemble_final_pred <- sapply(seq(1,nrow(ensemble_final)), read_out)

ensemble_final_result <- soccer_validation %>% cbind(ensemble_final_pred) %>%
  summarize(accuracy=mean(winner==ensemble_final_pred))

ensemble_final_nfl <- data.frame(QDA= predict(fit_qda_nfl, football_validation),
                              LDA= predict(fit_lda_nfl, football_validation),
                              RF = predict(fit_randomforest_nfl, football_validation))
read_out <- function(x){names(which.max(table(t(ensemble_final_nfl[x,]))))}
ensemble_final_pred_nfl <- sapply(seq(1,nrow(ensemble_final_nfl)), read_out)

ensemble_final_result_nfl <- football_validation %>% cbind(ensemble_final_pred_nfl) %>%
  summarize(accuracy=mean(winner==ensemble_final_pred_nfl))

data.frame(sport=c("soccer","football"), accuracy=c(ensemble_final_result$accuracy,
                                                    ensemble_final_result_nfl$accuracy))
```

```
##      sport  accuracy
## 1   soccer 0.6731302
## 2 football 0.8203125
```

For soccer, a final accuracy of match outcome prediction of 67% was achieved. This is well in line with what has been observed during model building. For football, on the other hand, the accuracy drops from 85% to 82% when going from the train/test dataset to the final validation dataset. This suggests that there might have been overtraining of the model to some degree.

Discussion

In summary, the caret package was used to apply different classification algorithms to game statistics data sets in soccer and football. After examining several algorithms (KNN, LDA, QDA, decision tree and random forest), it was finally decided to perform predictions using an ensemble of LDA, QDA and random forest. Using these methods the outcome of soccer games (home win, away win or draw) could be predicted with an accuracy of 67%. On the other hand, using this methodology football results (home win or away win) could be predicted with an accuracy of 82%. These findings are in agreement with the original hypothesis, that football has a lower element of chance in deciding game results and that usually the better team dominates the game and ultimately is the winner. In this respect the results support with data what is generally assumed (Tierney 2014).

Of course, this was a very rudimentary analysis with lots of caveats associated with it. While the origin of the data (NFL versus English Premier League) might still be comparable, as the two leagues are arguably the best ones in the world, the two data sets were not necessarily comparable in size, the football dataset being nearly three times as big. Another possible complication is that many soccer games end in draws, whereas draws are virtually absent in football. Comparing accuracy of classification with three classes (h, a, d) to classification with two classes (h, a) is certainly comparing apples with pears and should only be done with highest caution regarding the results. Additionally, more research would be necessary in order to find out which game statistics are most predictive for game outcome and should be used in an analysis like this one. Here, the main driver was data availability rather than well thought through decision. Finally, the classification algorithms used were the ones introduced in a online course on data science (Irizarry 2021).

Arguably, more research into suitable alternative methods combined with more diligent optimization of the tuning parameters could improve the prediction accuracy in both soccer as well as football.

Nonetheless, in conclusion, data science offers an intriguing possibility to examine generally assumed knowledge with scientific methods. The findings described above explain why so many people on this planet experience the joy, frustration and anger of the unpredictability of soccer (and why some football aficionados reject the game as unfair).

References

Irizarry, Rafael A. 2021. “Data Science: Capstone.”

“Kaggle: Your Home for Data Science.” n.d. www.kaggle.com.

Kuhn, Max. 2009. “The Caret Package.”

Mohr, Joseph. 2021. “English Premier League Match Events and Results 2001-2002 Season Onwards.” June 12, 2021. https://www.kaggle.com/josephvm/english-premier-league-game-events-and-results.

Sasser, David. 2021. “NFL Game Stats (2010-2019) Weekly Nfl Game Stats over the Past 10 Seasons.” January 25, 2021. https://www.kaggle.com/davidsasser/nfl-game-stats-20102019.

Tierney, John. 2014. “Soccer, a Beautyful Game of Chance.” *The New York Times*, July. https://www.nytimes.com/2014/07/08/science/soccer-a-beautiful-game-of-chance.html.