

Project Final

John Cho

11/6/2019

```
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(lubridate)

##
## Attaching package: 'lubridate'
##
## The following object is masked from 'package:base':
##
##   date
```

```
library(pscl)

## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```
library(stringr)
```

2012-2016 Seasons

Post Rule Change Wins: 2012-2016 Seasons

```
postWin <- read.csv(file="wonAppendedUPD.csv", header=TRUE, sep=",", stringsAsFactors = FALSE)
postWinKickoffs <- postWin[(postWin$Time == "15:00"), ]

##Adds the Home team variable
postHometeam <- read.csv(file="12-16home.csv", header=TRUE, sep=",", stringsAsFactors = FALSE)

postHometeam <- postHometeam %>%
  filter(At.Symbol == "@") %>%
  filter(!grepl("T", Result))
postHometeam$Date <- format(mdy(postHometeam$Date), "%Y-%m-%d")

postWinKickoffs <- postWinKickoffs %>%
```

```

mutate(Home = 0) %>%
mutate(TeamRating = Team1RatingDiff - Team2RatingDiff)

sizePostWinKickoffs = length(postWinKickoffs$Date)
sizePostHometeam = length(postHometeam$Date)

for(i in 1:sizePostWinKickoffs){
  for (j in 1:sizePostHometeam){
    if(postWinKickoffs$Date[i] == postHometeam$Date[j] & postWinKickoffs$Team[i] == postHometeam$Opp[j])
      postWinKickoffs$Home[i] = 1
  }
}

#postWinKickoffs <- postWinKickoffs %>%
# mutate(Win = 0)
#postWinKickoffs$Win <- ifelse(postWinKickoffs$Team == postWinKickoffs$Home, 1, 0)

```

Post Rule Change Losses: 2012-2016 Seasons

```

postLost <- read.csv(file="lostAppendedUPD.csv", header=TRUE, sep=",", stringsAsFactors = FALSE)
postLostKickoffs <- postLost[(postLost$Time == "15:00"), ]

##Adds the Home team variable
postLostKickoffs <- postLostKickoffs %>%
  mutate(Home = 0) %>%
  mutate(WinDiff = -WinDiff) %>%
  mutate(TeamRating = Team2RatingDiff - Team1RatingDiff)

sizePostLostKickoffs = length(postLostKickoffs$Date)

for(i in 1:sizePostLostKickoffs){
  for (j in 1:sizePostHometeam){
    if(postLostKickoffs$Date[i] == postHometeam$Date[j] & postLostKickoffs$Opp[i] == postHometeam$Opp[j])
      postLostKickoffs$Home[i] = 1
  }
}

#postLostKickoffs$Win <- ifelse(postLostKickoffs$Opp == postLostKickoffs$CoinToss, 1, 0)

```

Post Rule Changed Combined for Wins and Losses

```

postKickoffs <- rbind(postWinKickoffs, postLostKickoffs)
postKickoffs <- postKickoffs %>%
  arrange(desc(Date), Team) %>%
  mutate(Post = 1)

mean(postLostKickoffs$WinDiff)

```

```
## [1] 1.25
```

```
mean(postWinKickoffs$WinDiff)
```

```
## [1] 1.543478
```

```
postKickoffs <- postKickoffs %>%  
  mutate(CoinTossBinary = 0)
```

```
for(i in 1:sizePostHometeam){  
  for (j in 1:sizePostHometeam){  
    if(postKickoffs$Date[i] == postHometeam$Date[j] & postKickoffs$CoinToss[i] == postHometeam$Opp[j]){  
      postKickoffs$CoinTossBinary[i] = 1  
    }  
  }  
}  
}
```

2007-2011 Seasons

Pre Rule Change Wins: 2007-2011 Seasons

```
preWinKickoffs <- read.csv(file="07-11WonAppendedUPD.csv", header=TRUE, sep="," , stringsAsFactors = FALSE)
```

```
##Adds the Home team variable
```

```
preHometeam <- read.csv(file="7-11home.csv", header=TRUE, sep="," , stringsAsFactors = FALSE)
```

```
preHometeam <- preHometeam %>%  
  filter(At.Symbol == "@") %>%  
  filter(!grepl("T", Result))
```

```
preHometeam$Date <- format(mdy(preHometeam$Date), "%Y-%m-%d")
```

```
preWinKickoffs <- preWinKickoffs %>%  
  mutate(Home = 0) %>%  
  mutate(TeamRating = Team1RatingDiff - Team2RatingDiff)
```

```
sizePreWinKickoffs = length(preWinKickoffs$Date)  
sizePreHometeam = length(preHometeam$Date)
```

```
for(i in 1:sizePreWinKickoffs){  
  for (j in 1:sizePreHometeam){  
    if(preWinKickoffs$Date[i] == preHometeam$Date[j] & preWinKickoffs$Team[i] == preHometeam$Opp[j]){  
      preWinKickoffs$Home[i] = 1  
    }  
  }  
}  
}
```

Pre Rule Change Losses: 2007-2011 Seasons

```
preLostKickoffs <- read.csv(file="07-11LostAppendedUPD.csv", header=TRUE, sep="," , stringsAsFactors = FALSE)
```

```
##Adds the Home team variable
```

```
preLostKickoffs <- preLostKickoffs %>%  
  mutate(Home = 0) %>%  
  mutate(WinDiff = -WinDiff) %>%
```

```

mutate(TeamRating = Team2RatingDiff - Team1RatingDiff)

sizePreLostKickoffs = length(preLostKickoffs$Date)

for(i in 1:sizePreLostKickoffs){
  for (j in 1:sizePreHometeam){
    if(preLostKickoffs$Date[i] == preHometeam$Date[j] & preLostKickoffs$Opp[i] == preHometeam$Opp[j]){
      preLostKickoffs$Home[i] = 1
    }
  }
}

```

Pre Rule Changed Combined for Wins and Losses

```

preKickoffs <- rbind(preWinKickoffs, preLostKickoffs)
preKickoffs <- preKickoffs %>%
  arrange(desc(Date), Team) %>%
  mutate(Post = 0)

```

```
mean(preLostKickoffs$WinDiff)
```

```
## [1] 0.5555556
```

```
mean(preWinKickoffs$WinDiff)
```

```
## [1] 1.763158
```

```

preKickoffs <- preKickoffs %>%
  mutate(CoinTossBinary = 0)

for(i in 1:sizePreHometeam){
  for (j in 1:sizePreHometeam){
    if(preKickoffs$Date[i] == preHometeam$Date[j] & preKickoffs$CoinToss[i] == preHometeam$Opp[j]){
      preKickoffs$CoinTossBinary[i] = 1
    }
  }
}

```

```

Kickoffs <- rbind(preKickoffs, postKickoffs)
Kickoffs <- Kickoffs %>%
  arrange(Date, Team)

```

Models

Binary Logit Model

```

logit.windiff <- glm(Home ~ WinDiff + Post*WinDiff + CoinTossBinary,
  data=Kickoffs,
  family=binomial(link="logit"))

summary(logit.windiff)

```

```
##
```

```
## Call:
```

```
## glm(formula = Home ~ WinDiff + Post * WinDiff + CoinTossBinary,
##      family = binomial(link = "logit"), data = Kickoffs)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.5752  -1.2703   0.9726   1.0725   1.2169
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.27947    0.28366   0.985   0.325
## WinDiff       -0.06198    0.05962  -1.040   0.299
## Post          -0.13908    0.34792  -0.400   0.689
## CoinTossBinary 0.16161    0.33153   0.487   0.626
## WinDiff:Post    0.08709    0.08121   1.072   0.284
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 208.08  on 151  degrees of freedom
## Residual deviance: 206.59  on 147  degrees of freedom
## AIC: 216.59
##
## Number of Fisher Scoring iterations: 4
```

```
logit.cointoss <- glm(Home ~ WinDiff + Post*CoinTossBinary + CoinTossBinary,
                      data=Kickoffs,
                      family=binomial(link="logit"))

summary(logit.cointoss)
```

```
##
## Call:
## glm(formula = Home ~ WinDiff + Post * CoinTossBinary + CoinTossBinary,
##      family = binomial(link = "logit"), data = Kickoffs)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.388  -1.278   1.011   1.079   1.145
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.20422    0.31203   0.654   0.513
## WinDiff       -0.01560    0.04013  -0.389   0.697
## Post           0.02865    0.45079   0.064   0.949
## CoinTossBinary 0.20106    0.47643   0.422   0.673
## Post:CoinTossBinary -0.10830    0.66004  -0.164   0.870
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 208.08  on 151  degrees of freedom
## Residual deviance: 207.73  on 147  degrees of freedom
## AIC: 217.73
##
## Number of Fisher Scoring iterations: 4
```