Final\_CurryAdam

Adam Curry

8/5/2019

### Introduction

I chose to conduct analysis on three NFL Football datasets. The first was a spreadsheet of games from 1966 - 2017. It contained the pregame over/under, point spread, final scores, who the favorite team was to win the game, the wind speed, game temperature, and various other valuable data elements. The second dataset contained stadium information; the stadium type (dome, field, etc.), the typical weather type (cold, warm, moderate), and also the coordinates of the various teams. The third dataset was all the teams, their cities, team names, and team abbreviations.

I was drawn to the dataset primarily because football season is on the horizon, and I can hardly contain myself. In fact, I was hoping I could locate an NCAA dataset, as I’m an avid college football fan (Go Big Red!)… unfortunately, nothing was free. So, I settled on the next best league, the NFL. The dataset was relatively easy to read and mostly clean. I say mostly, because there were several anomalies I had to account for, like the fact that the Rams switched climate zones, the Vikings played at Gopher’s stadium for a season while US Bank arena was constructed, and various other things I remembered as I was inspecting the data. Also, there were several other variables I had to add to the dataset, in order to tell the full story.

### The problem statement you addressed.

Football is played all over the country, from Miami to Buffalo. We all know Miami and Buffalo have significantly different climates. In fact, as a football fan, I often hear that southern states suffer against northern states because of the weather. Since I have several historical game scores, point spreads, and weather variables, I thought this would be a great test. Do northern teams benefit from playing in colder states? Is the outcome of a game attributed to the weather? Can we predict if a northern home team will win, simply by looking at weather patterns?

### How you addressed this problem statement

I began by cleaning the data as illustrated in the prior assignment. However, during my analysis, I noticed several additional variables I needed to include. For example, I needed z-scores when the spread and the over/under were off in their pre-game predictions. If the outcome can’t be determined with the entire population, maybe we can predict the outlier games. I also calculated the windchill as an additional variable. Below are some steps I took to construct a final dataset:

setwd("C:\\Users\\adamp\\OneDrive\\Desktop\\a\_StatsR\\Final")  
df\_stadiums <- read.csv('nfl\_stadiums.csv')  
spreadspoke\_scores <- read.csv('spreadspoke\_scores.csv')  
df\_teams <- read.csv('nfl\_teams.csv')  
  
library(sqldf)

## Loading required package: gsubfn

## Loading required package: proto

## Loading required package: RSQLite

library(tidyverse)

## -- Attaching packages ---------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.0 v purrr 0.3.2  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 0.8.3 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(class)  
library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

library(pROC)

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

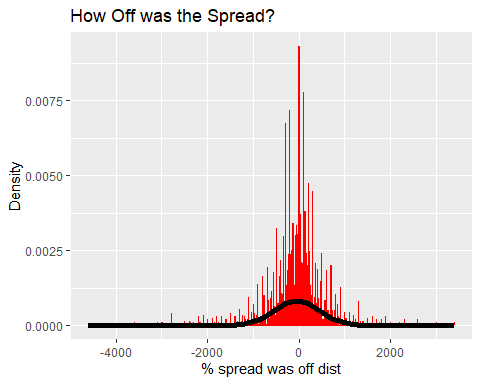
## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

# pull in stadium data  
df\_stadiums <- sqldf("SELECT DISTINCT  
 stadium\_name  
 ,stadium\_weather\_type  
 ,stadium\_open  
 FROM  
 df\_stadiums  
 ")  
# remove white space and special characters from stadium names to join further downstream   
df\_stadiums$stad\_pkey <- str\_replace\_all(df\_stadiums$stadium\_name, "[^[:alnum:]]", " ")  
df\_stadiums$stad\_pkey <- gsub("[[:space:]]", "", df\_stadiums$stad\_pkey)  
  
# designate an SKEY for a unique identifier further downstream  
df\_main <- sqldf("SELECT   
 schedule\_date||team\_home AS SKEY  
 ,schedule\_week  
 ,schedule\_date  
 ,team\_home  
 ,team\_away  
 ,spread\_favorite  
 ,over\_under\_line  
 ,team\_favorite\_id  
 ,score\_home  
 ,score\_away  
 ,over\_under\_line  
 ,weather\_temperature  
 ,weather\_wind\_mph  
 ,weather\_humidity  
 ,stadium  
 FROM  
 spreadspoke\_scores  
 WHERE  
 schedule\_season > 1980  
 /\*AND lower(schedule\_week) NOT IN ('wildcard','division','conference','superbowl')\*/  
 ")  
  
# remove white space and special characters from stadium names to join further downstream   
df\_main$stad\_fkey <- str\_replace\_all(df\_main$stadium, "[^[:alnum:]]", " ")  
df\_main$stad\_fkey <- gsub("[[:space:]]", "", df\_main$stad\_fkey)  
  
# add a foreign key for the teams table  
df\_main$fav\_fkey <- str\_replace\_all(df\_main$team\_favorite\_id, "[^[:alnum:]]", " ")  
df\_main$fav\_fkey <- gsub("[[:space:]]", "", df\_main$fav\_fkey)  
  
# pull in required fields from the team table  
df\_teams <- sqldf("SELECT  
 team\_id  
 ,team\_name  
 FROM   
 df\_teams  
 WHERE   
 team\_division IS NOT NULL AND team\_division <> ''  
 AND team\_name <> 'San Diego Chargers'  
 ")  
# add primary key to the teams table  
df\_teams$fav\_pkey <- str\_replace\_all(df\_teams$team\_id, "[^[:alnum:]]", " ")  
df\_teams$fav\_pkey <- gsub("[[:space:]]", "", df\_teams$fav\_pkey)  
  
#combine all three tables remove dome home games  
df\_final <- sqldf("SELECT  
 a.\*, b.\*,c.\*  
 FROM   
 df\_main a  
 LEFT JOIN   
 df\_stadiums b ON a.stad\_fkey = b.stad\_pkey  
 LEFT JOIN  
 df\_teams c ON a.fav\_fkey= c.fav\_pkey  
 where stadium\_weather\_type <>'dome'  
 ")  
# had to remove null divisions further upstream and had to remove San Diego Chargers and keep LA Chargers  
checkDups <- sqldf("SELECT  
 COUNT(SKEY) as CNT  
 ,SKEY  
 FROM  
 df\_final  
 GROUP BY SKEY  
 HAVING COUNT(SKEY) > 1")  
  
# calculate the actual over under, and identify the winning team, losing team, and their scores  
winningTeam <- sqldf(' SELECT  
 a.SKEY  
 ,a.schedule\_date  
 ,a.team\_home  
 ,a.team\_away  
 ,a.spread\_favorite  
 ,CASE WHEN winningTeam = favTeam THEN 1 ELSE 0 END AS favWon  
 ,favTeam,winningTeam,winningTeamScore,losingTeamScore  
 ,a.over\_under\_line  
 ,winningTeamScore + losingTeamScore AS actualOverUnder  
 FROM   
 df\_final a  
 INNER JOIN  
 (SELECT  
 SKEY  
 ,team\_home  
 ,schedule\_date  
 ,CASE WHEN score\_home > score\_away THEN team\_home   
 WHEN score\_away > score\_home THEN team\_away  
 ELSE "tie" END AS winningTeam  
 ,CASE WHEN score\_home > score\_away THEN score\_home   
 WHEN score\_away > score\_home THEN score\_away  
 ELSE 0 END AS winningTeamScore  
 ,CASE WHEN score\_home > score\_away THEN score\_away   
 WHEN score\_away > score\_home THEN score\_home  
 ELSE 0 END AS losingTeamScore  
 ,over\_under\_line  
 ,team\_name as favTeam  
 FROM   
 df\_final a  
 ) scores ON a.SKEY = scores.SKEY   
 ')  
   
# create a calculated field to show who won and the final point spread outcome  
spreadCorrect1 <- sqldf('  
 SELECT  
 SKEY  
 ,schedule\_date  
 ,team\_home  
 ,team\_away  
 ,favTeam  
 ,winningTeam  
 ,winningTeamScore  
 ,losingTeamScore  
 ,spread\_favorite  
 ,actualOverUnder  
 ,over\_under\_line  
 ,CASE WHEN favWon = 1 THEN losingTeamScore - winningTeamScore   
 WHEN favWon = 0 THEN winningTeamScore - losingTeamScore  
 ELSE spread\_favorite end as actualSpread  
 ,CASE WHEN favTeam = winningTeam THEN 1 ELSE 0 END AS favTeamWon  
 FROM   
 winningTeam  
 ')  
  
# create a calculated field to show how off was the Over Under and spread  
df\_final <- sqldf('SELECT  
 b.\*  
 ,a.favTeam  
 ,a.winningTeam  
 ,a.winningTeamScore  
 ,a.losingTeamScore  
 ,a.actualSpread  
 ,CASE WHEN a.actualSpread > 0 THEN 1 ELSE 0  
 END AS upsetInd  
 ,CASE WHEN a.favTeamWon = 1 THEN (abs(a.actualSpread) - abs(a.spread\_favorite))/abs(a.spread\_favorite)  
 ELSE (abs(a.actualSpread) - (a.spread\_favorite))/(a.spread\_favorite)  
 END\*100 prcntSprdOff  
 ,a.actualOverUnder  
 ,((abs(a.actualOverUnder) - abs(a.over\_under\_line))/abs(a.over\_under\_line))\*100 as prcntOvrUndrOff  
 FROM   
 spreadCorrect1 a  
 INNER JOIN df\_final b ON a.SKEY = b.SKEY  
 ')  
# caclulate the respective z scores  
df\_final$zSprdOff <- (df\_final$prcntSprdOff-mean(df\_final$prcntSprdOff, na.rm=TRUE))/sd(df\_final$prcntSprdOff, na.rm=TRUE)  
df\_final$zOvrUndOff <- (df\_final$prcntOvrUndrOff-mean(df\_final$prcntOvrUndrOff, na.rm=TRUE))/sd(df\_final$prcntOvrUndrOff, na.rm=TRUE)  
  
# bring in some weather variables to indicate the severity of the weather  
# cold season is defined as any games played in the last three weeks of the regular season  
# cold severity is binned accordingly  
# z score outliers are named  
df\_final <- sqldf("SELECT DISTINCT   
 1 as ind  
 ,1 as ind2  
 ,a.\*  
 ,CASE WHEN schedule\_week < 14 THEN 0 ELSE 1 END AS coldSeason   
 ,CASE WHEN weather\_temperature > 20 and weather\_temperature <= 32 THEN 'coldLVL1'  
 WHEN weather\_temperature > 10 and weather\_temperature < 20 THEN 'coldLVL2'  
 WHEN weather\_temperature <= 10 THEN 'coldLVL3'  
 ELSE 'coldLVL0'  
 END AS coldDayLvl  
 ,CASE WHEN favTeam = winningTeam THEN 1 ELSE 0 END AS favTeamWon  
 ,CASE WHEN team\_home = winningTeam THEN 1 ELSE 0 END AS homeTeamWon  
 ,CASE WHEN abs(zSprdOff) >= 1.96 THEN 1 ELSE 0 END AS zSprdOffIndicator  
 ,CASE WHEN abs(zOvrUndOff) >= 1.96 THEN 1 ELSE 0 END AS zOvrUndOffIndicator  
 FROM  
 df\_final a  
 ")  
  
# remove NA values  
df\_final <- na.omit(df\_final)  
  
# pivot the categorical variables  
df\_final <- df\_final %>% spread(stadium\_weather\_type, 1)  
df\_final <- df\_final %>% spread(coldDayLvl, 1)  
  
# replace the NA's in the new columns with 0's  
df\_final[is.na(df\_final)] <- 0  
  
df\_coldTeam2 <- sqldf("  
 SELECT  
 team\_home  
 ,count(SKEY) as gamesPlayed  
 ,sum(coldLvl1)AS sumCold1   
 ,sum(coldLvl2)AS sumCold2  
 ,sum(coldLvl3)AS sumCold3  
 FROM   
 df\_final  
 WHERE coldSeason = 1  
 GROUP BY 1")  
  
  
# cold teams are teams that have more cold level 1,2,3 games above the average in any category   
# played more than a single season  
df\_coldTeam1 <- sqldf("  
 SELECT DISTINCT   
 team\_home as coldTeamName  
 ,1 as coldTeamInd  
 FROM  
 df\_coldTeam2 a  
 ,(  
 SELECT   
 avg(sumCold1) AS avgCold1   
 ,avg(sumCold2) AS avgCold2  
 ,avg(sumCold3) AS avgCold3  
 FROM   
 df\_coldTeam2  
 ) c1  
 WHERE CASE WHEN sumCold1 > avgCold1 or sumCold2 > avgCold2 or sumCold3 > avgCold3 THEN 1 ELSE 0  
 END = 1 and gamesPlayed > 20   
 ")  
  
  
# indicator for the cold teams  
df\_final <- sqldf("  
 SELECT   
 a.\*  
 ,CASE WHEN coldTeamInd is null then 0 else 1 END as coldTeamHomeInd  
 FROM   
 df\_final a   
 left join   
 df\_coldTeam1 b on b.coldTeamName = a.team\_home  
 ")  
# indicator for the warm teams   
df\_warmTeams <- sqldf("  
   
 SELECT DISTINCT   
 team\_home as warmTeamName  
 ,1 as warmTeamInd  
 FROM   
 df\_final  
 WHERE team\_home not in (SELECT DISTINCT coldTeamName FROM df\_coldTeam1)  
 ")  
# combine the two  
df\_final <- sqldf("  
 SELECT   
 a.\*  
 ,CASE WHEN warmTeamInd is null then 0 else 1 END as warmTeamAwayInd  
 FROM   
 df\_final a   
 left join   
 df\_warmTeams b on warmTeamName = a.team\_away  
 ")  
  
# Calculate the windchill  
df\_final$windChill <- 35.74 + .6251 \* df\_final$weather\_temperature -   
 35.75 \* df\_final$weather\_wind\_mph^.16 + .4275 \*   
 df\_final$weather\_temperature \* df\_final$weather\_wind\_mph^.16  
  
df\_final <- sqldf("SELECT   
 1 as Kind  
 ,SKEY  
 ,schedule\_week  
 ,schedule\_date  
 ,team\_home  
 ,team\_away  
 ,team\_favorite\_id  
 ,favTeam  
 ,favTeamWon  
 ,winningTeam  
 ,homeTeamWon  
 ,coldSeason  
 ,coldLVL0  
 ,coldLVL1  
 ,coldLVL2  
 ,coldLVL3  
 ,coldTeamHomeInd  
 ,warmTeamAwayInd  
 ,cold  
 ,moderate  
 ,warm  
 ,zSprdOffIndicator  
 ,zOvrUndOffIndicator  
 ,score\_home  
 ,score\_away  
 ,stadium\_open  
 ,winningTeamScore  
 ,losingTeamScore  
 ,over\_under\_line  
 ,actualOverUnder  
 ,spread\_favorite  
 ,actualSpread  
 ,weather\_temperature  
 ,weather\_wind\_mph  
 ,windChill  
 ,weather\_humidity  
 ,prcntSprdOff  
 ,prcntOvrUndrOff  
 FROM   
 df\_final  
   
 ")

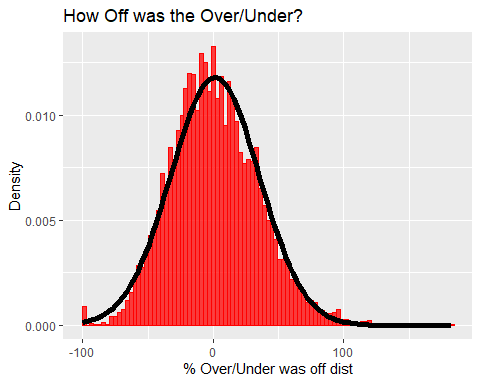
### Analysis

Next, I wanted to visualize the data distribution and initial correlation. I did this with my two variables “prcntSprdOff” and “prcntOvrUndrOff” in a histogram:

# view a distribution of how off the spread and over under were  
ggplot(df\_final, aes(prcntSprdOff)) +   
 geom\_histogram(aes(y = ..density..), colour = "red", fill = "red", alpha = 0.75,binwidth = 3 ) +   
 ggtitle("How Off was the Spread?") +  
 labs(x = "% spread was off dist", y = "Density") +   
 stat\_function(fun = dnorm, args =   
 list(mean = mean(df\_final$prcntSprdOff), sd =   
 sd(df\_final$prcntSprdOff)),colour = "black", size = 2)



ggplot(df\_final, aes(prcntOvrUndrOff)) +   
 geom\_histogram(aes(y = ..density..), colour = "red", fill = "red", alpha = 0.75,binwidth = 3 ) +   
 ggtitle("How Off was the Over/Under?") +  
 labs(x = "% Over/Under was off dist", y = "Density") +   
 stat\_function(fun = dnorm, args =   
 list(mean = mean(df\_final$prcntOvrUndrOff), sd =   
 sd(df\_final$prcntOvrUndrOff)),colour = "black", size = 2)



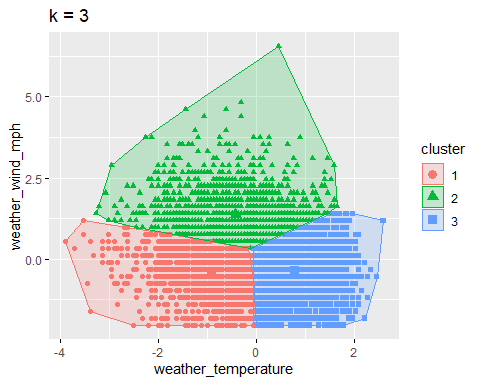
Based on the histograms, it seems Vegas is pretty good at what they do, and most of the games are around 0% for both point spreads and over/under predictions.

Next, I utilized the weather variables to determine the K means nearest neighbor, instead of focusing on “north” and “south”. These clusters may show a pattern in our scatter plots.

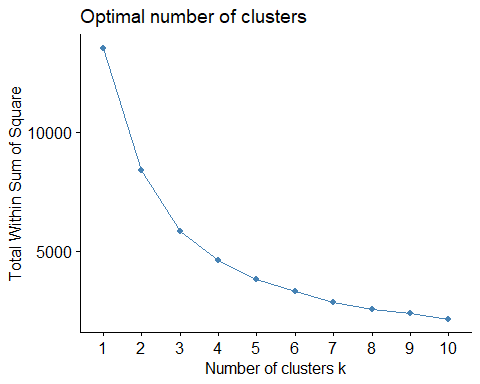
# k-means cluster variables based on weather patterns  
set.seed(123)  
weatherK <- df\_final[,33:34]  
library(cluster) # clustering algorithms  
library(factoextra) # clustering algorithms & visualization

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ

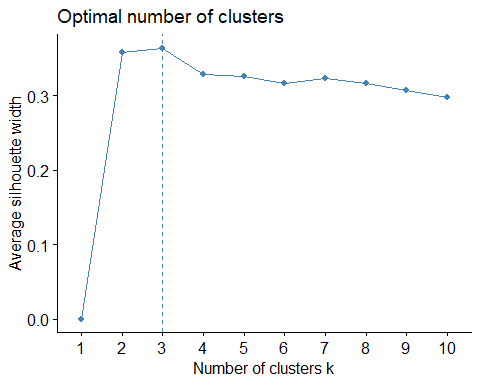
weatherK <- na.omit(weatherK)  
# convert all the columns to numeric  
weatherK <- sapply( weatherK, as.numeric)  
weatherK <- scale(weatherK)  
weatherK <- data.frame(weatherK)  
  
k2 <- kmeans(weatherK, centers = 3, nstart = 25)  
# plot  
fviz\_cluster(k2, geom = "point", data = weatherK) + ggtitle("k = 3")



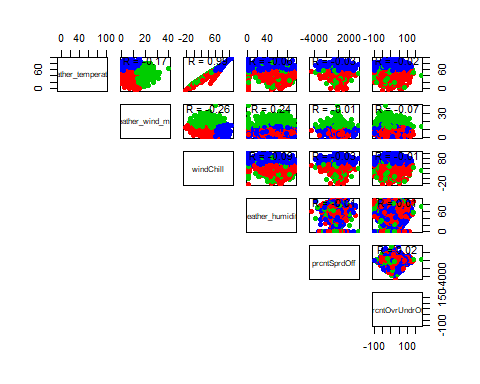
# find the optimal number of clusters for the K-means cluster algorithm above  
fviz\_nbclust(weatherK, kmeans, method = "wss")



fviz\_nbclust(weatherK, kmeans, method = "silhouette")



# add the clusters to the data frame  
df\_final$cluster <- k2$cluster  
  
# split apart clusters into separate columns  
df\_final <- df\_final %>% spread(cluster, 1)  
  
# replace the NA's in the new columns with 0's  
df\_final[is.na(df\_final)] <- 0  
df\_final$cluster <- k2$cluster  
#convert to numeric   
df\_final$`1` <- sapply(df\_final$`1`, as.numeric)  
df\_final$`2` <- sapply(df\_final$`2`, as.numeric)  
df\_final$`3` <- sapply(df\_final$`3`, as.numeric)  
  
upper.panel <- function(x, y){  
 points(x,y, pch = 19, col=c("red", "green3","blue")[df\_final$cluster])  
 r <- round(cor(x, y), digits = 2)  
 txt <- paste0("R = ", r)  
 usr <- par("usr"); on.exit(par(usr))  
 par(usr = c(0, 1, 0, 1))  
 text(0.5, 0.9, txt)  
}  
pairs(df\_final[,32:37], lower.panel = NULL,   
 upper.panel = upper.panel)  
pairs(df\_final[,32:37], lower.panel = NULL,   
 upper.panel = upper.panel)



### Implications

So far, the correlation between these variables appears minimal, at least based on the surface of the scatter plots above. Even breaking the teams up into clusters doesn’t appear to impact the correlation, as they appear evenly spread out within the plots.

Next, I conducted a multi-linear regression analysis.

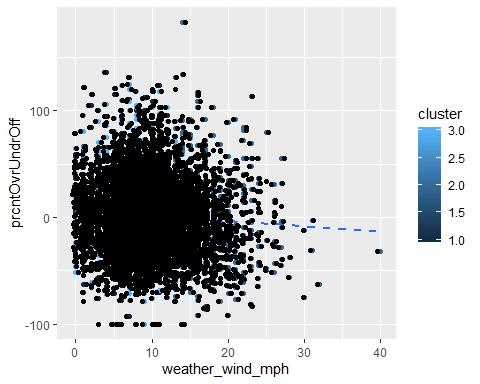
lm.spread <- lm(prcntSprdOff ~ weather\_wind\_mph + windChill, data = df\_final, na.action = na.exclude)  
summary(lm.spread)

##   
## Call:  
## lm(formula = prcntSprdOff ~ weather\_wind\_mph + windChill, data = df\_final,   
## na.action = na.exclude)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4564.4 -170.3 15.5 190.3 3458.2   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 29.0513 24.2142 1.200 0.23027   
## weather\_wind\_mph -2.4730 1.3232 -1.869 0.06168 .   
## windChill -0.8441 0.3111 -2.713 0.00668 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 488.5 on 6758 degrees of freedom  
## Multiple R-squared: 0.001299, Adjusted R-squared: 0.001004   
## F-statistic: 4.395 on 2 and 6758 DF, p-value: 0.01237

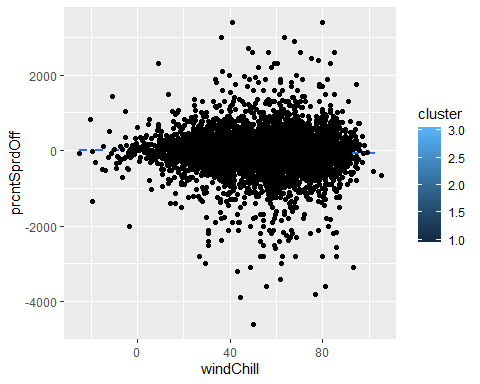
lm.overunder <- lm(prcntOvrUndrOff ~ weather\_wind\_mph + windChill, data = df\_final, na.action = na.exclude)  
summary(lm.overunder)

##   
## Call:  
## lm(formula = prcntOvrUndrOff ~ weather\_wind\_mph + windChill,   
## data = df\_final, na.action = na.exclude)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -105.014 -23.107 -2.335 21.589 182.535   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 9.43968 1.67321 5.642 1.75e-08 \*\*\*  
## weather\_wind\_mph -0.54081 0.09144 -5.915 3.49e-09 \*\*\*  
## windChill -0.05378 0.02150 -2.502 0.0124 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 33.75 on 6758 degrees of freedom  
## Multiple R-squared: 0.005289, Adjusted R-squared: 0.004995   
## F-statistic: 17.97 on 2 and 6758 DF, p-value: 1.65e-08

ggplot(data = df\_final, mapping=aes(x = weather\_wind\_mph, y = prcntOvrUndrOff)) +   
 geom\_point(aes(color = cluster)) + geom\_smooth(aes(group = 1), method = "lm", se = FALSE, linetype = 2) +  
 geom\_jitter()



ggplot(data = df\_final, mapping=aes(x = windChill, y = prcntSprdOff)) +   
 geom\_point(aes(color = cluster)) + geom\_smooth(aes(group = 1), method = "lm", se = FALSE, linetype = 2) +  
 geom\_jitter()



While there is a linear relationship, and the weather variables are a factor in the model’s prediction according the p-value’s significance, the overall R^2 is minimal .005, which means .5% of the variance in the over/under score is a result of the weather variables Windchill and weather\_wind\_mph. Even less is the spread where .1% of the variance is explained by the weather variables.

A final technique used was a logistic regression. Perhaps we could predict a binary variable’s outcome given the weather patterns.

Train <- createDataPartition(df\_final$homeTeamWon, p = 0.8, list = FALSE)  
training <- df\_final[ Train, ]  
testing <- df\_final[ -Train, ]  
  
homeTeamWon <- glm(formula = training$homeTeamWon ~ training$windChill  
 , family = binomial(), data = training)  
summary(homeTeamWon)

##   
## Call:  
## glm(formula = training$homeTeamWon ~ training$windChill, family = binomial(),   
## data = training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.445 -1.307 1.009 1.049 1.118   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.520199 0.080635 6.451 1.11e-10 \*\*\*  
## training$windChill -0.003612 0.001398 -2.583 0.00979 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 7357.6 on 5408 degrees of freedom  
## Residual deviance: 7350.9 on 5407 degrees of freedom  
## AIC: 7354.9  
##   
## Number of Fisher Scoring iterations: 4

res <- predict(homeTeamWon, testing, type = "response")  
res <- predict(homeTeamWon, training, type = "response")  
  
confmatrix <- table(Actual\_Value = training$homeTeamWon, Predicted\_Value = res > .5)  
confmatrix

## Predicted\_Value  
## Actual\_Value TRUE  
## 0 2269  
## 1 3140

roc\_obj <- roc(training$homeTeamWon, res)

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

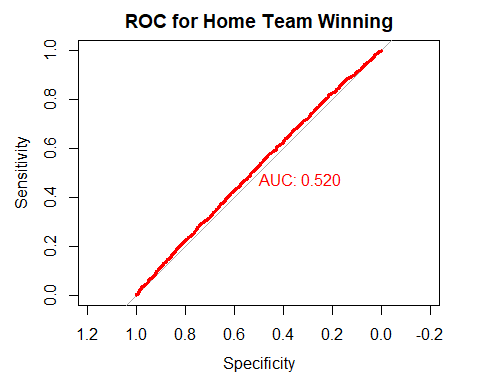
## Setting direction: controls < cases

auc(roc\_obj)

## Area under the curve: 0.5203

TPR = rev(roc\_obj$sensitivities)  
FPR = rev(1 - roc\_obj$specificities)  
labels = roc\_obj$homeTeamWon  
scores = roc\_obj$res  
  
plot(roc(training$homeTeamWon, res, direction = "<"), print.auc = TRUE,  
 col = "red", lwd = 3, main = "ROC for Home Team Winning")

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



Again, the findings are unyielding, and don’t indicate a significant impact from the weather. Also, the logistic regression is not a good model to predict the outcome of the northern team win.

### Limitations

There are several limitations in my analysis. Primarily, these findings are limited to the wind temp and wind speed, and do not account for other weather variables such as rain or snow. Also, the dataset doesn’t include other important variables, such as injuries, who the coach was, or how many Pro Bowlers were on the team.

### Concluding Remarks

At this point, I do not think we can reject the null hypothesis that weather significantly impacts southern teams. It may have an impact, but it is minimal. I even compared the outlier spreads and over/under predictions to see if the weather could’ve been a factor, but there was minimal improvement to the models.

If we think about what impacts a game, there are several other variables that would probably have a greater impact on pre-game spreads and over/under predictions. For example, midgame injuries can devastate a team, especially if the quarterback goes out. Furthermore, the NFL is filled with professional athletes who get paid a lot of money to play. They also have equipment, warming stations, and medical staff to aid in cold games. It’d be interesting to see if the NCAA or even lower level divisions yield similar results. However, for this analysis, the datasets were not enough to reject the null-hypothesis.