

NFL_Project

Eric Young

8/13/2019

Predicting the Longevity of NFL Players

The goal of this project is to predict which players from NCAA College Football will be drafted into the NFL. I will use data of the college players to predict which players will be drafted into the NFL. The success criteria will be based on TRUE or FALSE, TRUE if the player is drafted into the NFL, they were successful or FALSE the player was not drafted into the NFL or they were unsuccessful.

I will start the prediction from their Freshman year through their Senior year. Performance data from the NFL data set will not be used for predictions. I will be using data from the NCAA (via the CollegeballR package) to determine if a player will be drafted into the NFL.

Potential Clients

The client for this project would be the NFL teams, players, and agents. The information provided could help the NFL teams understand which teams generally have the most talent. It will also help them to see what teams may have hidden talent. This project could benefit players in High School deciding which schools can best help them get to the NFL. For current college players the project can help them know their likelihood of getting into the NFL. For agents the project will help them decide which college teams they should go after to get players drafted into the NFL.

Libraries used

Below are the libraries which I will be using in the capstone project. The collegeballR package is from GitHub and you will need to use devtools to install it.

```
library(devtools)
library(readr)
library(dplyr)
library(tidyr)
library(ggplot2)
library(xtable)
library(collegeballR)
library(knitr)
library(kableExtra)
library(ROCR)
library(pROC)
library(caTools)
library(texreg)
```

NFL Data Set Import

Set the working directory and read in the Basic Stats file

```
basic_stats <- read.csv("nflstatistics/basic_stats_clean.csv", header = TRUE, stringsAsFactors = FALSE)
```

Age	Name	College	Experience	Height..inches.	Weight..lbs.
NA	Evans, Fred	Notre Dame	3	71	185
NA	Raiff, Jim	Dayton	1	70	235
56	Fowler, Bobby	Louisiana Tech	1	74	230
30	Johnson, Quinn	LSU	5	73	255
25	Walton, L.T.	Central Michigan	3	77	305

NCAA Data Set Import

We need to import the collegedf data set which has all of our NCAA player and team information.

```
collegedf <- read.csv("./nflstatistics/college_data.csv", header = TRUE, stringsAsFactors = FALSE)
```

Player	team_name	Yr	Pos	GP	GS	G	Rush.Attempts	Rush.Net.Yards	Rush.YdsGained
Lacoste, Anthony	Air Force	Sr	DB	12	6	12	135	890	906
Husar, Jr., Michael	Air Force	Sr	OL	12	12	12	NA		
Adenji, Moshhood	Air Force	Sr	OL	12	12	12	NA		
Henry, Jerry	Air Force	Sr	OL	12	12	12	NA		
Baska, David	Air Force	Sr	K	12	0	12	1		

NFL & NCAA Joined Data Set Import

```
college_draft <- read.csv("./nflstatistics/college_draft.csv")
```

Player	Yr	Pos	GP	GS	GNS	Height..inches.	Weight..lbs.	was_drafted	team_name
Lacoste, Anthony	Sr	DB	12	6	6	NA	NA	FALSE	Air Force
Husar, Jr., Michael	Sr	OL	12	12	0	NA	NA	FALSE	Air Force
Adenji, Moshhood	Sr	OL	12	12	0	NA	NA	FALSE	Air Force
Henry, Jerry	Sr	OL	12	12	0	NA	NA	FALSE	Air Force
Baska, David	Sr	K	12	0	12	NA	NA	FALSE	Air Force

Source of data

There are two data sets which I will be using for this capstone project. The first source dataset is from Kaggle and is named NFL Statistics. The second source comes the NCAA (pulled using a package created by Meyappan called collegeballR).

When trying to tie team_stats to the data frame I found a bug in the team_stats.R where it was not looking up sport and giving an error. I had to fork the original package into my repository and edit the code.

The original size of the basic_stats data frame is 17,172 rows and 17 columns. Since the data from the collegeballR package is only available from 2014 - 2018. I will only be covering the years between 2014 and 2017 as the 2018 data was not available yet at the time of downloading.

The variables I will use to predict are:

- College Team
- Position
- Games Played - How many games the player played in the season
- Games Started - How many games a player started
- Games Not Started - Games Played - Games started = Games Not Started
- Rush Attempts
- Rush Net Yards

- Rush Yards Gained
- Height
- Weight
- Experience
- Year - Freshman, Sophomore, Junior, Senior

Deliverables

The deliverables for this project will be:

- NFL_Project.R - this is the file with all of the code
- Capstone-Project.Rmd - this file has the analysis
- Output of project file as a PDF
- Deck of slides with insights and plots

Clean Basic Stats (NFL Data)

Let's check the data and basic information on basic_stats - xtable turn summary into a table

```
kable(summary(basic_stats[ , 1:7])) %>%
  kable_styling(latex_options = c("striped", "scale_down"))
```

X	Age	Birth.Place	Birthday	College	Current.Status	Current.Team
Min. : 1	Min. : 19.00	Length:17172	Length:17172	Length:17172	Length:17172	Length:17172
1st Qu.: 4294	1st Qu.: 28.00	Class :character				
Median : 8586	Median : 39.00	Mode :character				
Mean : 8586	Mean : 43.84	NA	NA	NA	NA	NA
3rd Qu.: 12879	3rd Qu.: 55.00	NA	NA	NA	NA	NA
Max. :17172	Max. :2017.00	NA	NA	NA	NA	NA
NA	NA's:3668	NA	NA	NA	NA	NA

```
kable(summary(basic_stats[ , 8:14])) %>%
  kable_styling(latex_options = c("striped", "scale_down"))
```

Experience	Height..inches.	High.School	High.School.Location	Name	Number	Player.Id
Min. : 0.000	Min. :61.00	Length:17172	Length:17172	Length:17172	Min. : 1.00	Length:17172
1st Qu.: 1.000	1st Qu.:72.00	Class :character	Class :character	Class :character	1st Qu.:26.00	Class :character
Median : 3.000	Median :74.00	Mode :character	Mode :character	Mode :character	Median :52.00	Mode :character
Mean : 3.829	Mean :73.51	NA	NA	NA	Mean :51.77	NA
3rd Qu.: 6.000	3rd Qu.:75.00	NA	NA	NA	3rd Qu.:77.00	NA
Max. :25.000	Max. :82.00	NA	NA	NA	Max. :99.00	NA
NA	NA's:146	NA	NA	NA	NA's:15464	NA

```
kable(summary(basic_stats[ , 15:18])) %>%
  kable_styling(latex_options = c("striped"))
```

	Position	Weight..lbs.	start_year	end_year
Length:17172	Min. : 1.0	Min. :1920	Min. :1920	
Class :character	1st Qu.:195.0	1st Qu.:1961	1st Qu.:1966	
Mode :character	Median :220.0	Median :1986	Median :1987	
NA	Mean :229.2	Mean :1979	Mean :1982	
NA	3rd Qu.:255.0	3rd Qu.:2002	3rd Qu.:2006	
NA	Max. :375.0	Max. :2016	Max. :2016	
NA	NA's:51	NA's:3096	NA's:3096	

Clean College Ball R

The CollegeBallR package was available on Github but we had many issues using this data set. So we had to pull the data directly from the NCAA website and create our own data frame. The data frame had the below issues which needed to be fixed.

1. Pull all combinations for each team and season from the NCAA website.
2. Dropped all players which didn't have a position
 - The players didn't have a position because they didn't play that season.
3. Dropped all years that were N/A
4. Relabeled positions and statndardized them.
 - For example WILL is a type of Line Backer so I relabeled them as LB
5. Created csv to be be used later in the project.

Create and Clean College_draft

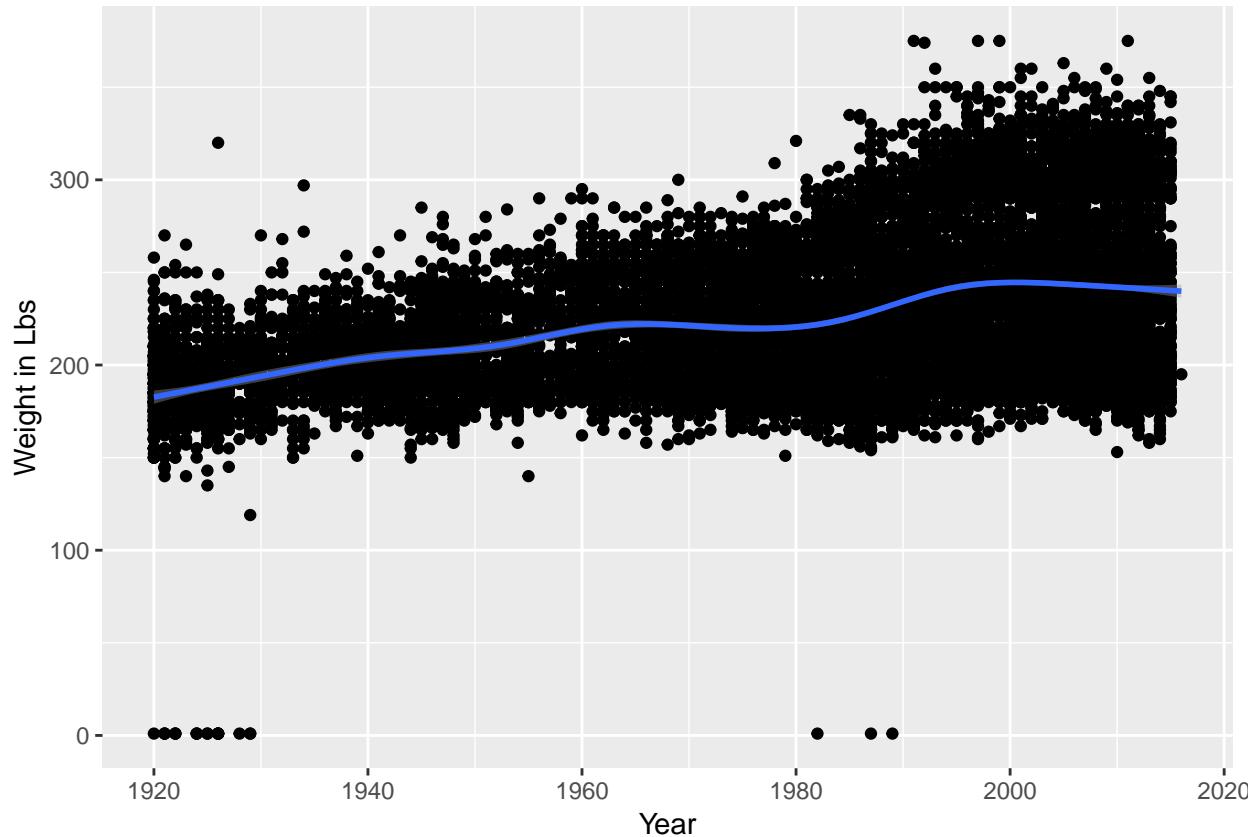
To create the data frame College_draft I did a left join on the NCAA and NFL data frames. By creating this data frame we were able to analyze how many players were drafted from college into the NFL and from which teams. However, there were some data problems that I needed to fix which I will list below.

1. Create logical variable of TRUE or FALSE if the player was drafted into the NFL.
2. Sanitized the column names and stripped out invalid characters.
3. Remove comma from rushing yards variables.
4. Change games played and games started variables to numeric.
5. Created a new variable games not started.
6. Dropped all NULL positions from the data set.
7. Created a cleaned version of the csv to be be used later in the project.

NFL Plots

One question that may be asked while reviewing NFL player data is are the players bigger now than they were when the NFL started. In the chart below we see that players now weigh aproximately 60 to 70 lbs more than in the 1920's. From 1920 to around 2000 there was a steady increase in the weight of the players. Since 2000 the average weight of players has stayed about the same. The below chart shows the mean weight of football players has increased from 1920 to 2018.

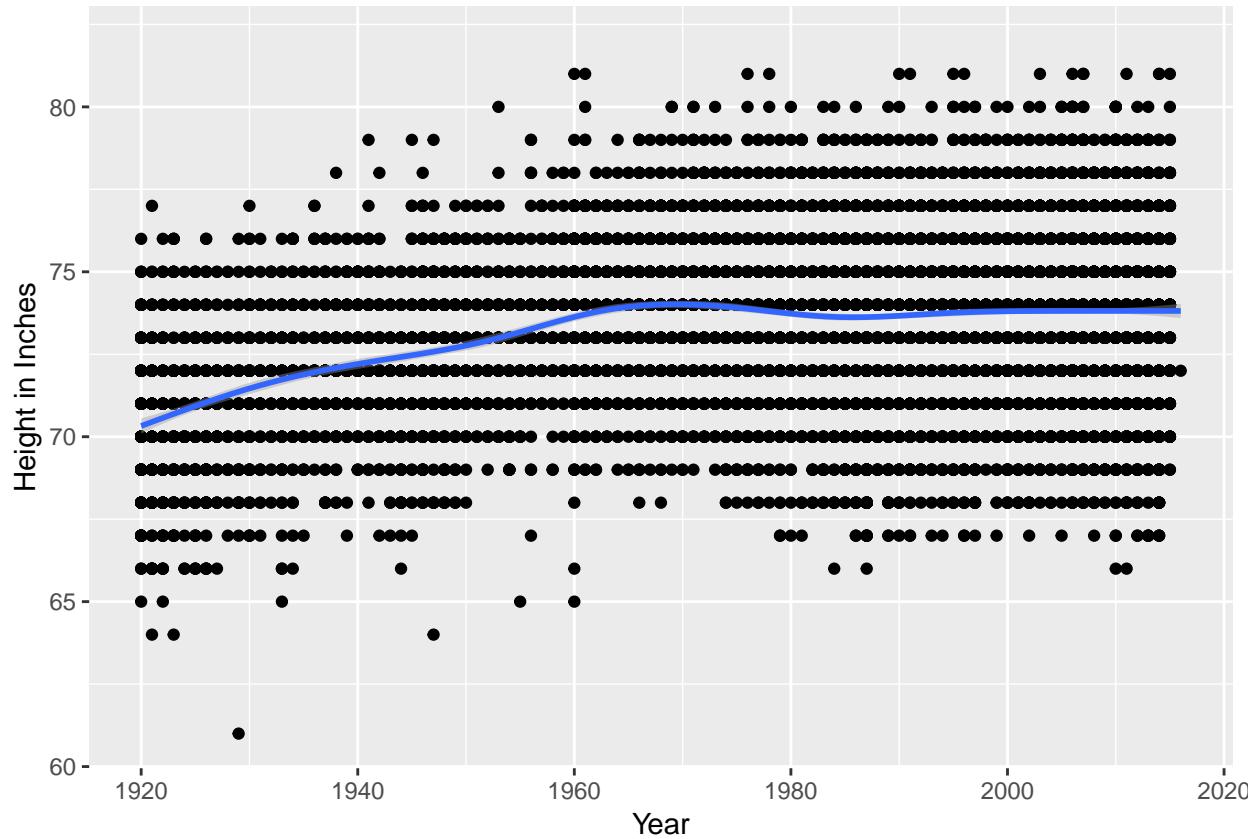
```
ggplot(basic_stats) +  
  aes(x = as.numeric(start_year), y = Weight..lbs.) +  
  geom_point() +  
  geom_smooth() +  
  labs(x = "Year", y = "Weight in Lbs")  
  
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'  
## Warning: Removed 3147 rows containing non-finite values (stat_smooth).  
## Warning: Removed 3147 rows containing missing values (geom_point).
```



Another question that may be asked is are player taller today then they were when the NFL started. The answer is yes, the average height in 1920 was about 70 inches and dramatically increased until about 1970 where players are an average height of 74 inches. This chart shows the average height in the NFL hit its peak in the 1970s and is close to the same today.

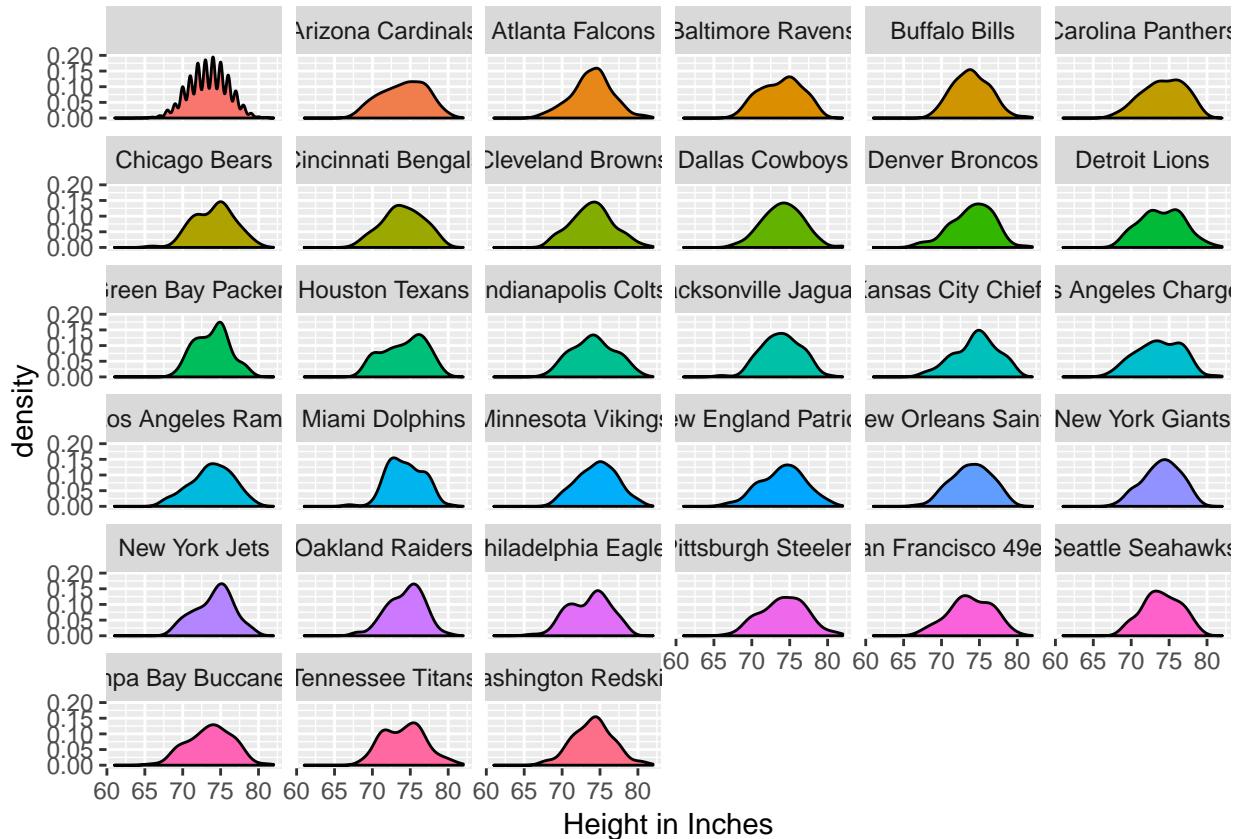
What is interesting about the chart below is all of the plots are evenly placed and staggered. The reason for this is when a team records the height of a player they don't measure in quarter or half of an inch they will round up to the nearest inch.

```
ggplot(basic_stats) +
  aes(x = as.numeric(start_year), y = Height..inches.) +
  geom_point() +
  geom_smooth() +
  labs(x = "Year", y = "Height in Inches")
```



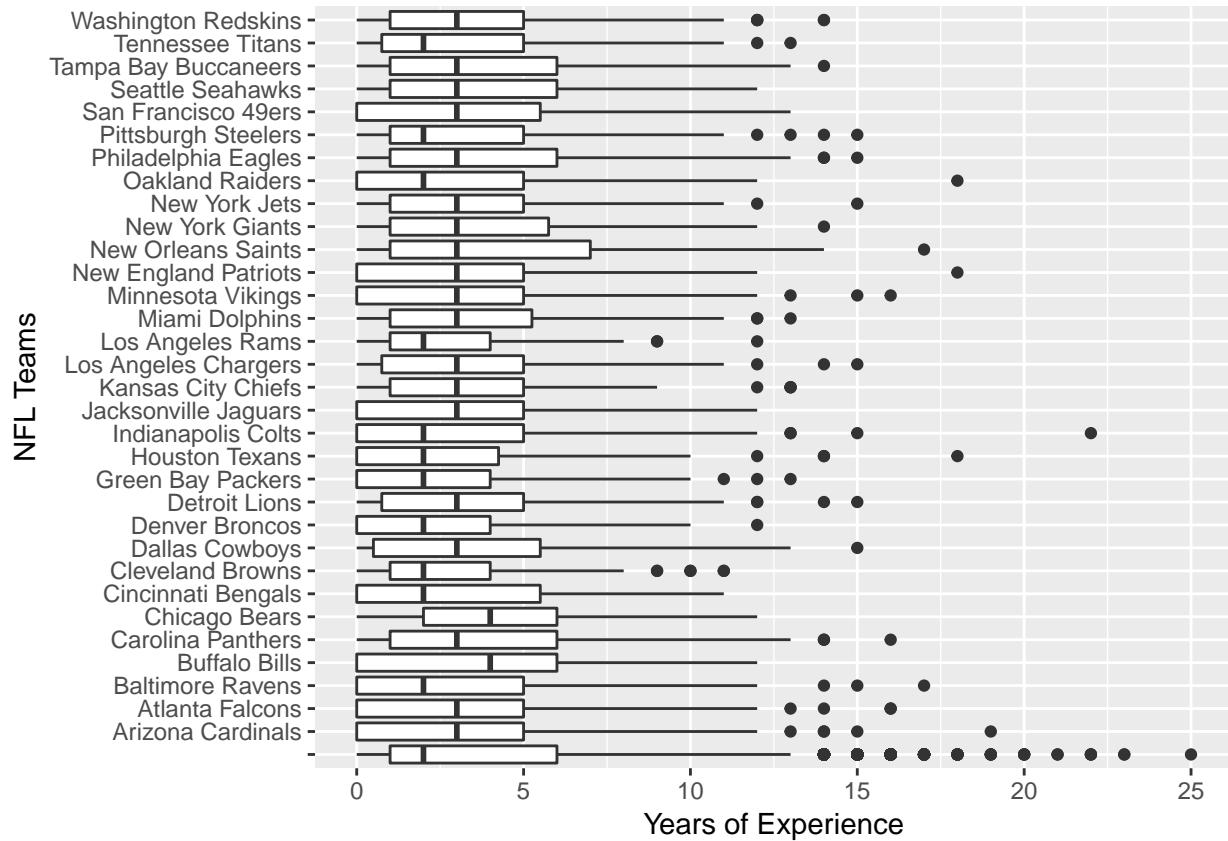
The density plot below shows the difference in height between all of the NFL teams. The most common height in the NFL seems to be around 75 inches. Players who are drafted and play in the NFL are usually between 70 and 77 inches in height. This means if you are below 70 inches it is still possible to be drafted but you'll have to be an excellent athlete to make it into the NFL. On the other hand there are a few players who are in the NFL at 80 inches or about 6 feet 8 inches.

```
ggplot(basic_stats, aes(Height..inches., fill = Current.Team)) +
  geom_density() +
  facet_wrap(~Current.Team) +
  guides(fill = "none") +
  labs(x = "Height in Inches")
```



The box and whisker plot below shows us the lowest observation, highest observation, the four quartiles, and average years of experience for each NFL team. For many of the teams the first quartile is 0 meaning 1 in 4 players are rookies. There are 19 teams with 3 years of median experience and 12 teams with 2 years of median experience. This means most players in the NFL have between 2 and 3 years of experience. It looks like 75% of players will retire between 5 and 7 years of playing in the NFL. The New Orleans Saints have the highest observation of years of experience and third quartile of player retiring. Almost every team has outliers and what is surprising is some players have over 15 years in the NFL.

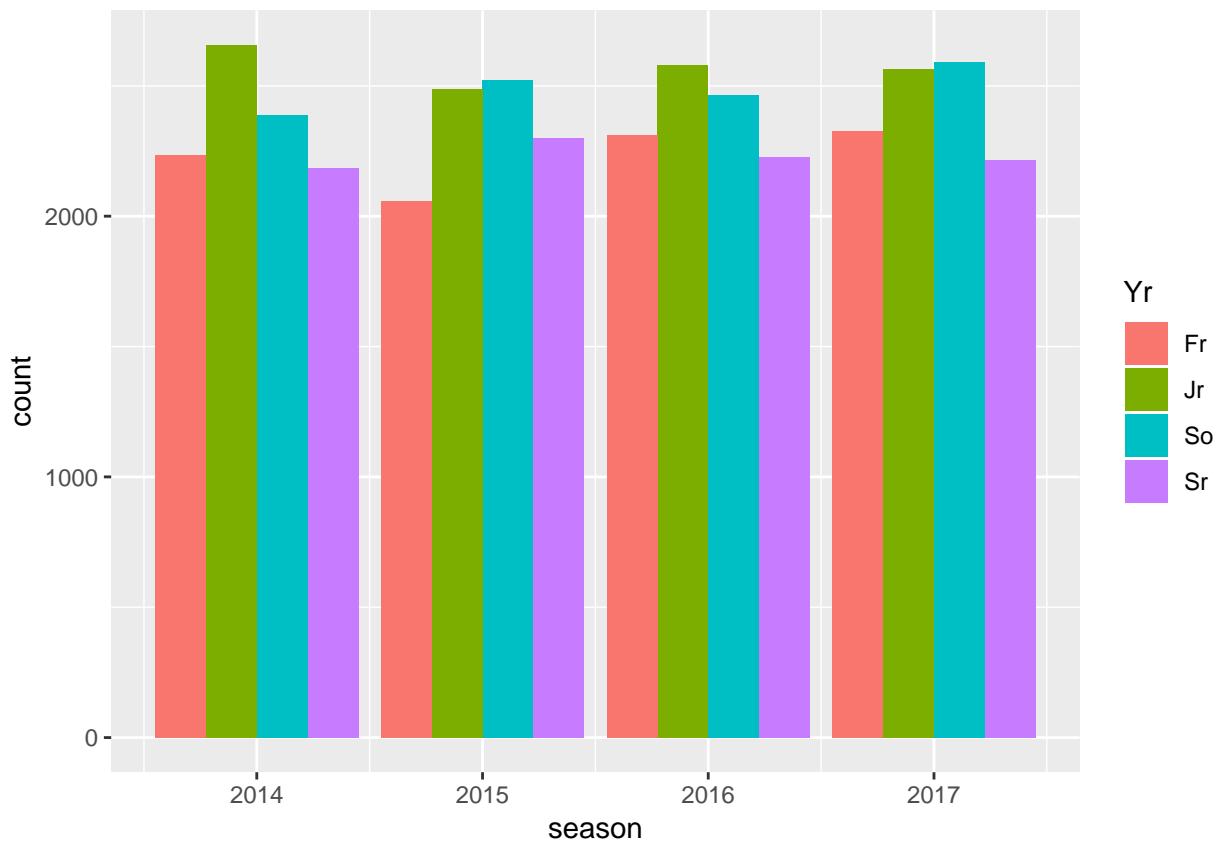
```
ggplot(basic_stats, aes(Current.Team, Experience)) +
  geom_boxplot() +
  labs(x = "NFL Teams", y = "Years of Experience") +
  coord_flip()
```



NCAA Plots

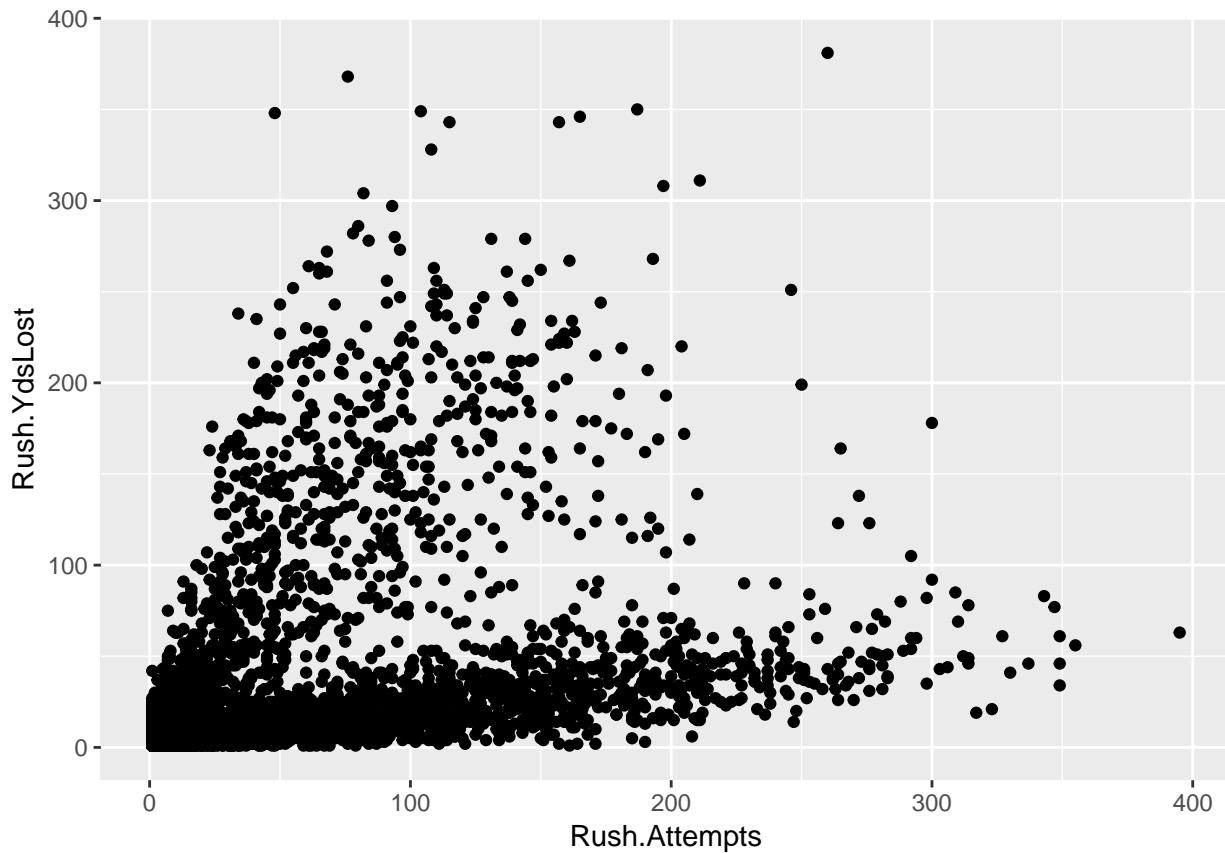
This bar plot shows the distribution of players by year, season and the progression of players from Freshman, Sophomore, Junior, and Senior by year. The amount of players progressing from one year to the next is never the same for any of the years. Senior year is always lower than the previous Junior year. This could be because of players moving into the NFL before completing their Senior year. The increase of players in the Sophomore and Junior years could be caused by players transferring from Junior Colleges.

```
ggplot(collegedf, aes(season, fill = Yr, group = Yr)) +
  geom_bar(position = "dodge")
```



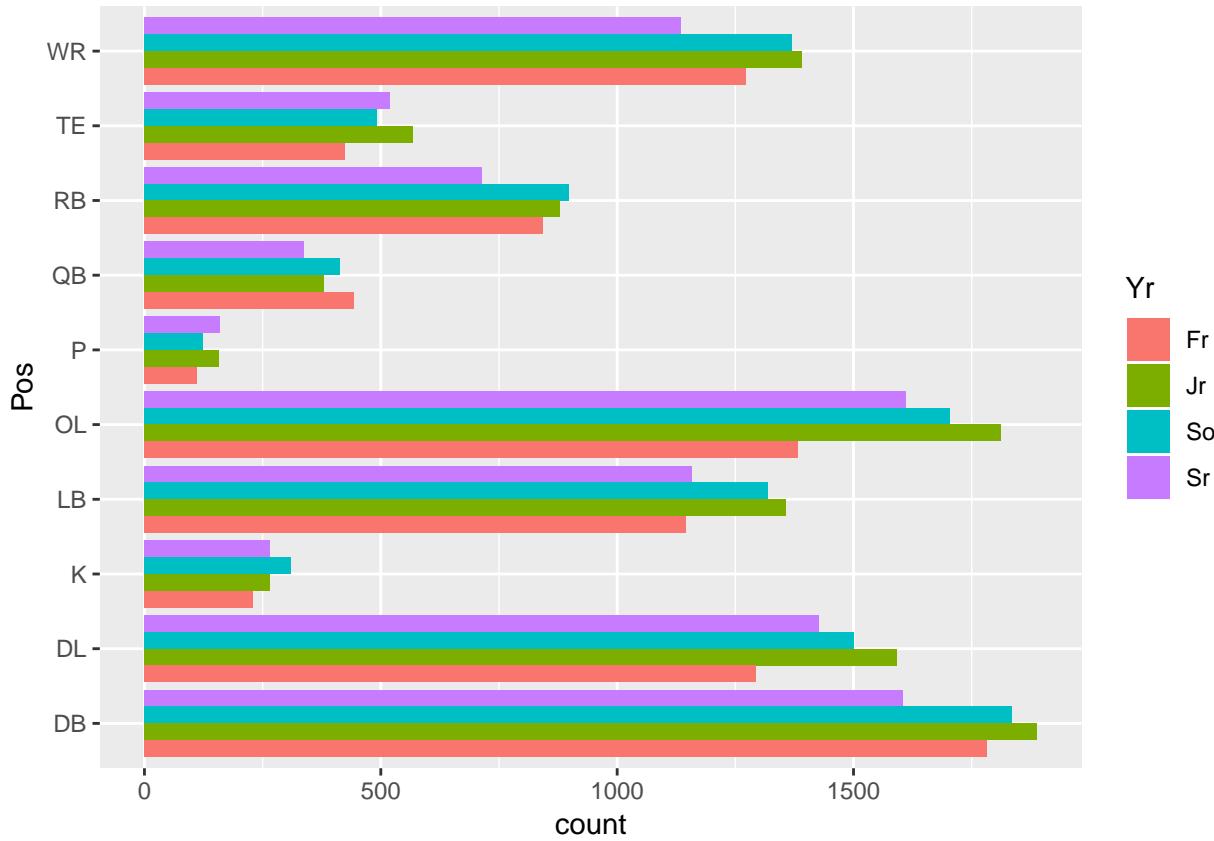
The below chart shows by the number of attempts how many yards are lost. The natural thought would be the more rushing attempts a player has the more likely they are to lose yardage. To an extent this is true. However we see in the scatter plot that this isn't exactly true. We can see players who have rushed nearly 400 times and have only lost around 50 yards. Then there is another group who have rushed far less times but have lost a lot more yardage.

```
ggplot(collegedf, aes(x = `Rush.Attempts`, y = `Rush.YdsLost`)) +
  geom_point()
```



The below chart shows by year how many players played each position and displays them by year. We can see that there is variation for every position from year to year. It is not uncommon for a player to switch positions, for example a wide receiver may switch to a Defensive Back from one year to the next. Another scenario is a player transfers from a junior college to a university and that will cause a difference in count from one year to the next. Players may also leave for the NFL draft after their Junior year and this can cause a drop in the number of players for the positions. Another thing we can look at is some positions like Punter and Kicker have far fewer players in these positions than Defensive Back or Offensive Lineman. These positions only require a few players per team as there is only one slot on the field for them and they are less likely to get injured. Whereas Defensive Backs and Offensive Linemen take up multiple spots on a field and are far more likely to get injured.

```
ggplot(collegedf, aes(Pos, fill = Yr, group = Yr)) +
  geom_bar(position = "dodge") +
  coord_flip()
```



Define college_teams data frame

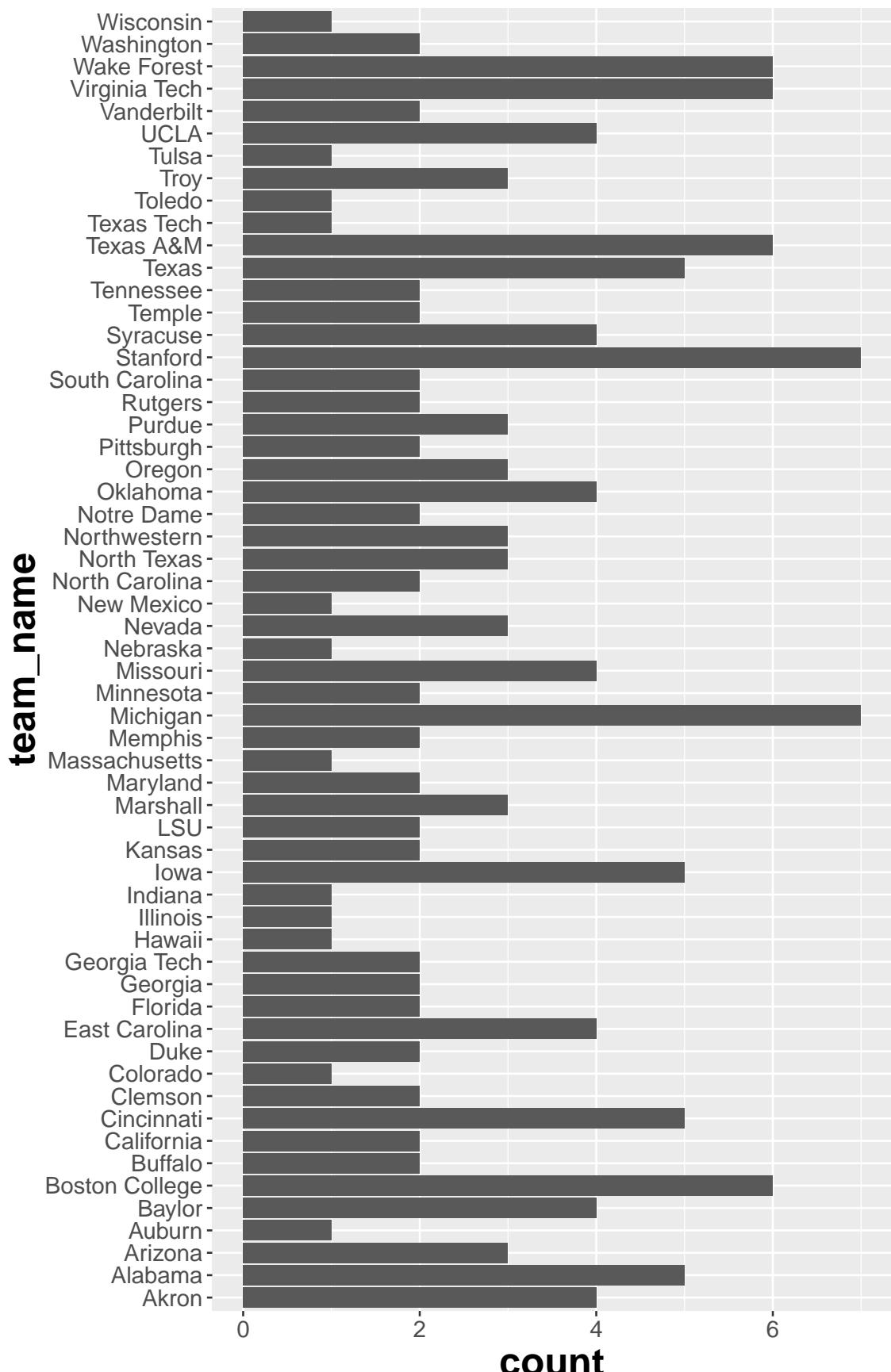
```
college_teams <- inner_join(collegedf, basic_stats, by = c("team_name" = "College", "Player" = "Name"))
```

The below chart shows us the colleges which have at least one or more players drafted into the NFL from 2014 onwards. The two teams with the highest number of drafted players are Michigan and Stanford tied with 7 players drafted into the NFL. This chart can give us an idea of which schools consistently send players to the NFL and can even give us an idea of how competitive they are in the NCAA. The idea to the last notion is the more players which are drafted would mean the teams have more highly skilled players which in turn could mean the team was more competitive. The total number of players drafted between 2014 and 2017 is 2735.

```
table(college_draft$was_drafted)

ggplot(filter(college_teams, start_year >= 2014), aes(x = team_name)) +
  geom_bar() +
  coord_cartesian(ylim=c(2,7)) +
  coord_flip() +
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=20,face="bold"))

## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```



-Include high level statistics and aggregate data not raw data

Linear and Logistic Regression

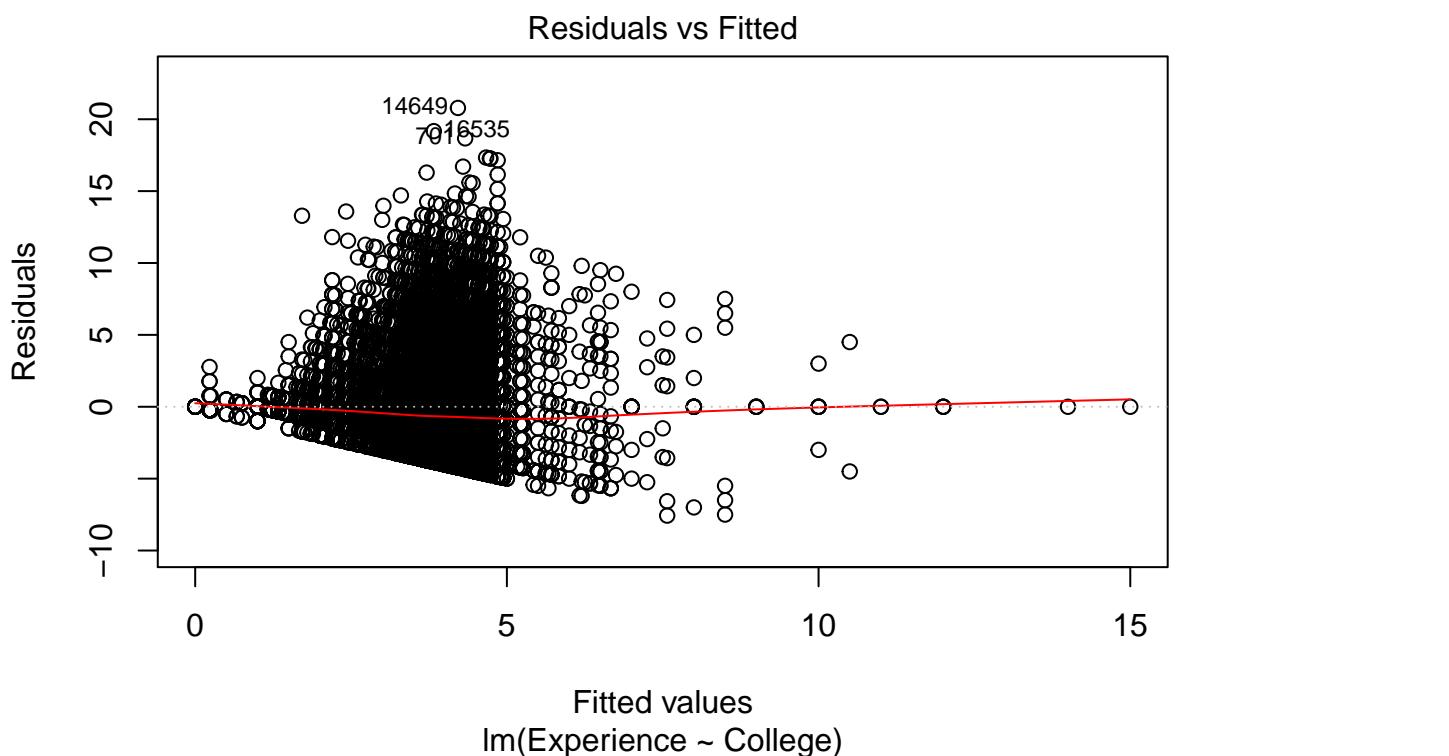
```
#Linear Model for experience
nfl_model <- lm(Experience ~ College, data = basic_stats)

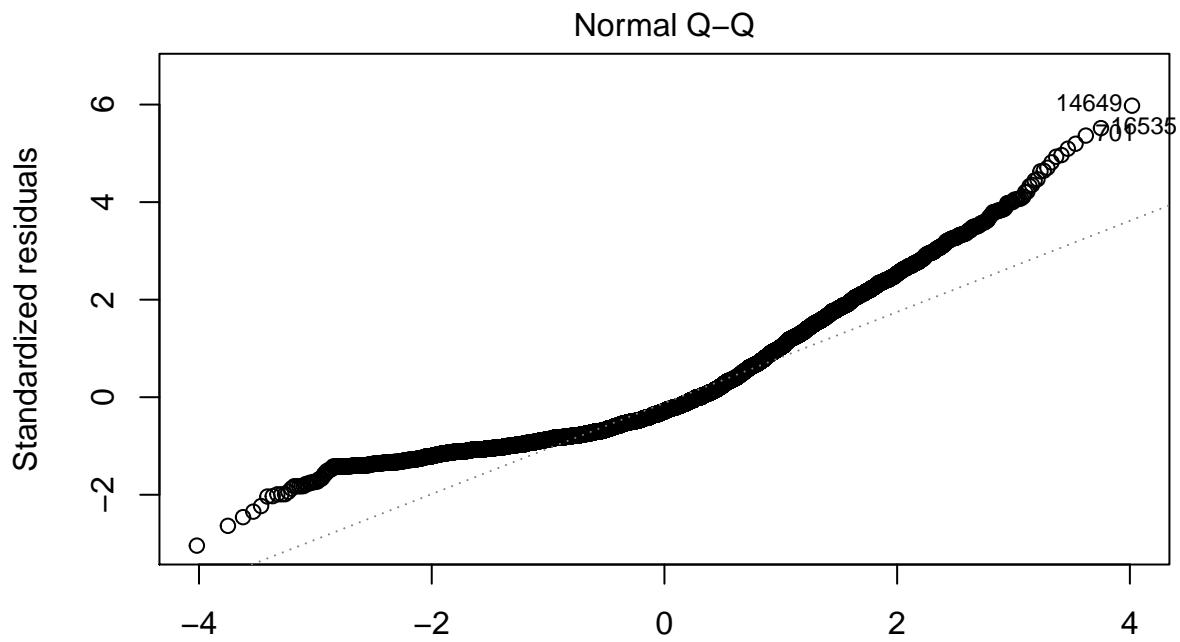
SSE <- sum(nfl_model$residuals^2)
SSE

## [1] 199823.1

Plot of nfl_model
plot(nfl_model)

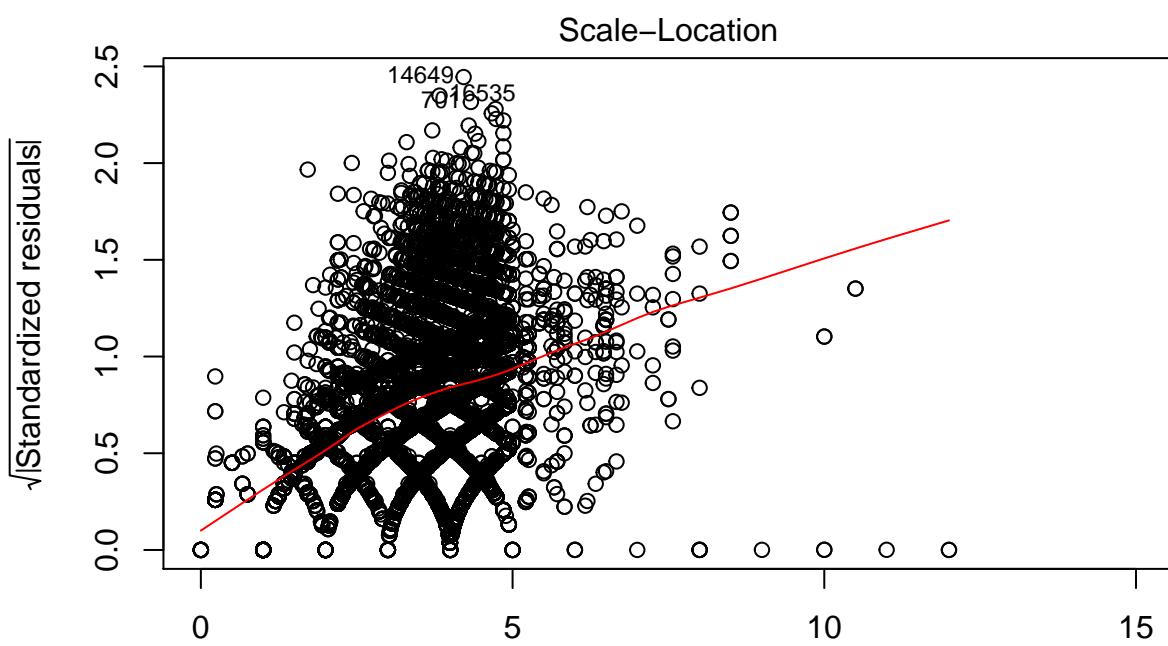
## Warning: not plotting observations with leverage one:
##   291, 468, 1060, 1190, 1305, 1339, 1392, 1595, 1608, 1756, 1928, 2536, 2630, 2647, 2701, 2715, 2805
```





```
## Warning: not plotting observations with leverage one:
```

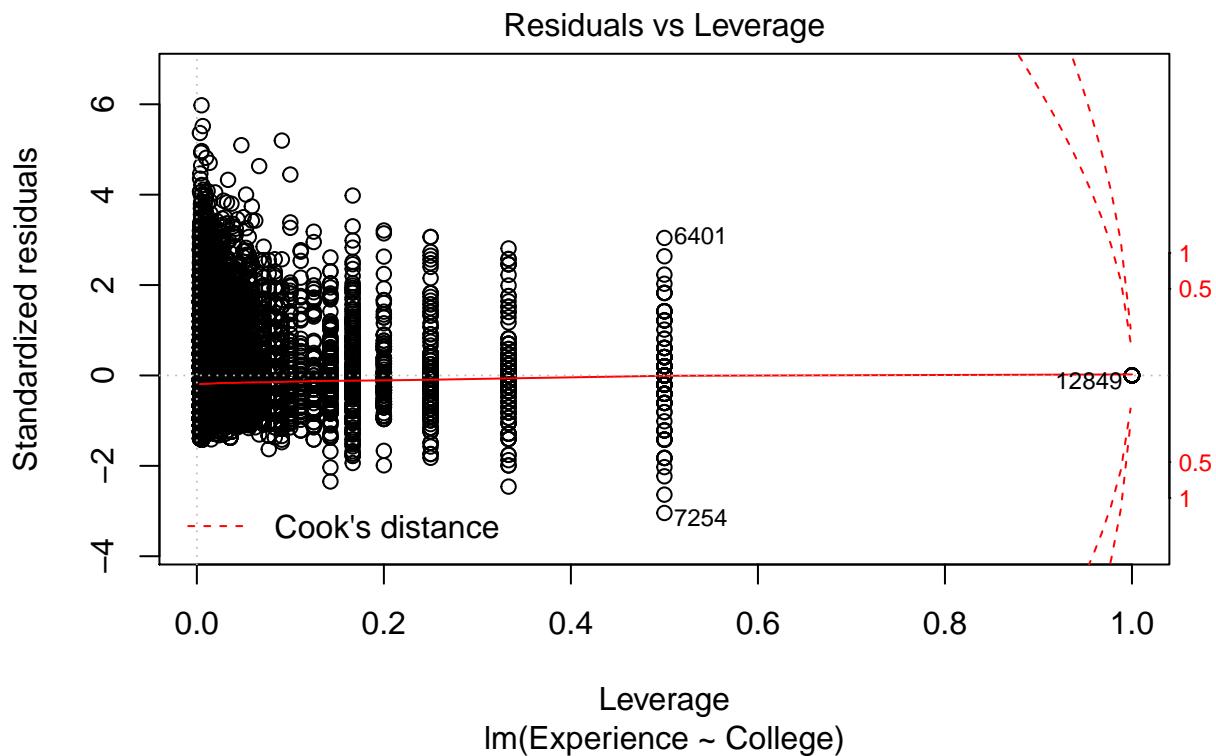
```
##   291, 468, 1060, 1190, 1305, 1339, 1392, 1595, 1608, 1756, 1928, 2536, 2630, 2647, 2701, 2715, 2805
```



Im(Experience ~ College)

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```



```
nfl_model2 <- lm(Experience ~ College + Weight..lbs., data = basic_stats)
```

```
SSE2 <- sum(nfl_model2$residuals^2)
SSE2
```

```
## [1] 198610.6
```

```
nfl_model3 <- lm(Experience ~ Height..inches. + Weight..lbs., data = basic_stats)
```

```
SSE3 <- sum(nfl_model3$residuals^2)
SSE3
```

```
## [1] 210394.3
```

Linear Model for drafting

```
draft1 <- lm(was_drafted ~ poly(Height..inches., 2) + Weight..lbs., data = na.omit(college_draft[c("Hei
summary(draft1)
```

```
##
## Call:
## lm(formula = was_drafted ~ poly(Height..inches., 2) + Weight..lbs.,
##     data = na.omit(college_draft[c("Height..inches.", "Weight..lbs.",
##     "was_drafted"))))
##
## Residuals:
##      Min        1Q    Median        3Q       Max
## -1.75e-12  2.10e-16  7.00e-16  1.10e-15  2.43e-15
##
## Coefficients:
```

```

##                               Estimate Std. Error   t value Pr(>|t|) 
## (Intercept)           1.000e+00  4.753e-15 2.104e+14 <2e-16 ***
## poly(Height..inches., 2)1 -4.770e-14  4.652e-14 -1.025e+00   0.305
## poly(Height..inches., 2)2  7.441e-15  3.359e-14  2.210e-01   0.825
## Weight..lbs.          1.406e-17  1.953e-17  7.200e-01   0.472
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 3.354e-14 on 2725 degrees of freedom
## Multiple R-squared:    0.5, Adjusted R-squared:  0.4994 
## F-statistic: 908.3 on 3 and 2725 DF, p-value: < 2.2e-16
draft2 <- lm(was_drafted ~ Yr + Pos + GP + Rush.Attempts + Rush.Net.Yards + Rush.YdsGained, data = college_draft)
summary(draft2)

##
## Call:
## lm(formula = was_drafted ~ Yr + Pos + GP + Rush.Attempts + Rush.Net.Yards +
##     Rush.YdsGained, data = college_draft)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.46034 -0.13633 -0.08064 -0.01741  1.02194 
## 
## Coefficients:
##                               Estimate Std. Error   t value Pr(>|t|) 
## (Intercept)        -0.0075320  0.0328669 -0.229  0.81875
## YrJr              0.0556609  0.0113342  4.911 9.33e-07 ***
## YrSo              0.0204057  0.0113117  1.804  0.07129 .
## YrSr              0.0752046  0.0117005  6.427 1.41e-10 ***
## PosDL             0.0081329  0.0576809  0.141  0.88788
## PosK              0.0325759  0.0549159  0.593  0.55307
## PosLB             0.0452583  0.0499812  0.906  0.36524
## PosOL             -0.0284778  0.0923126 -0.308  0.75772
## PosP              -0.0433417  0.0449517 -0.964  0.33500
## PosQB             -0.0446240  0.0320378 -1.393  0.16372
## PosRB             -0.0581409  0.0300279 -1.936  0.05289 .
## PosTE             0.0413863  0.0428876  0.965  0.33459
## PosWR             0.0301238  0.0302366  0.996  0.31916
## GP                0.0059415  0.0013824  4.298 1.75e-05 ***
## Rush.Attempts    -0.0003725  0.0003050 -1.221  0.22205
## Rush.Net.Yards   -0.0001894  0.0001258 -1.505  0.13232
## Rush.YdsGained   0.0004082  0.0001479  2.759  0.00581 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.2904 on 5410 degrees of freedom
## (32681 observations deleted due to missingness)
## Multiple R-squared:  0.06804, Adjusted R-squared:  0.06529 
## F-statistic: 24.69 on 16 and 5410 DF, p-value: < 2.2e-16
texreg(list(draft1,draft2), table = FALSE, use.packages = FALSE)

```

	Model 1	Model 2
(Intercept)	1.00*** (0.00)	-0.01 (0.03)
poly(Height..inches., 2)1	-0.00 (0.00)	
poly(Height..inches., 2)2	0.00 (0.00)	
Weight..lbs.	0.00 (0.00)	
YrJr		0.06*** (0.01)
YrSo		0.02 (0.01)
YrSr		0.08*** (0.01)
PosDL		0.01 (0.06)
PosK		0.03 (0.05)
PosLB		0.05 (0.05)
PosOL		-0.03 (0.09)
PosP		-0.04 (0.04)
PosQB		-0.04 (0.03)
PosRB		-0.06 (0.03)
PosTE		0.04 (0.04)
PosWR		0.03 (0.03)
GP		0.01*** (0.00)
Rush.Attempts		-0.00 (0.00)
Rush.Net.Yards		-0.00 (0.00)
Rush.YdsGained		0.00** (0.00)
R ²	0.50	0.07
Adj. R ²	0.50	0.07
Num. obs.	2729	5427
RMSE	0.00	0.29

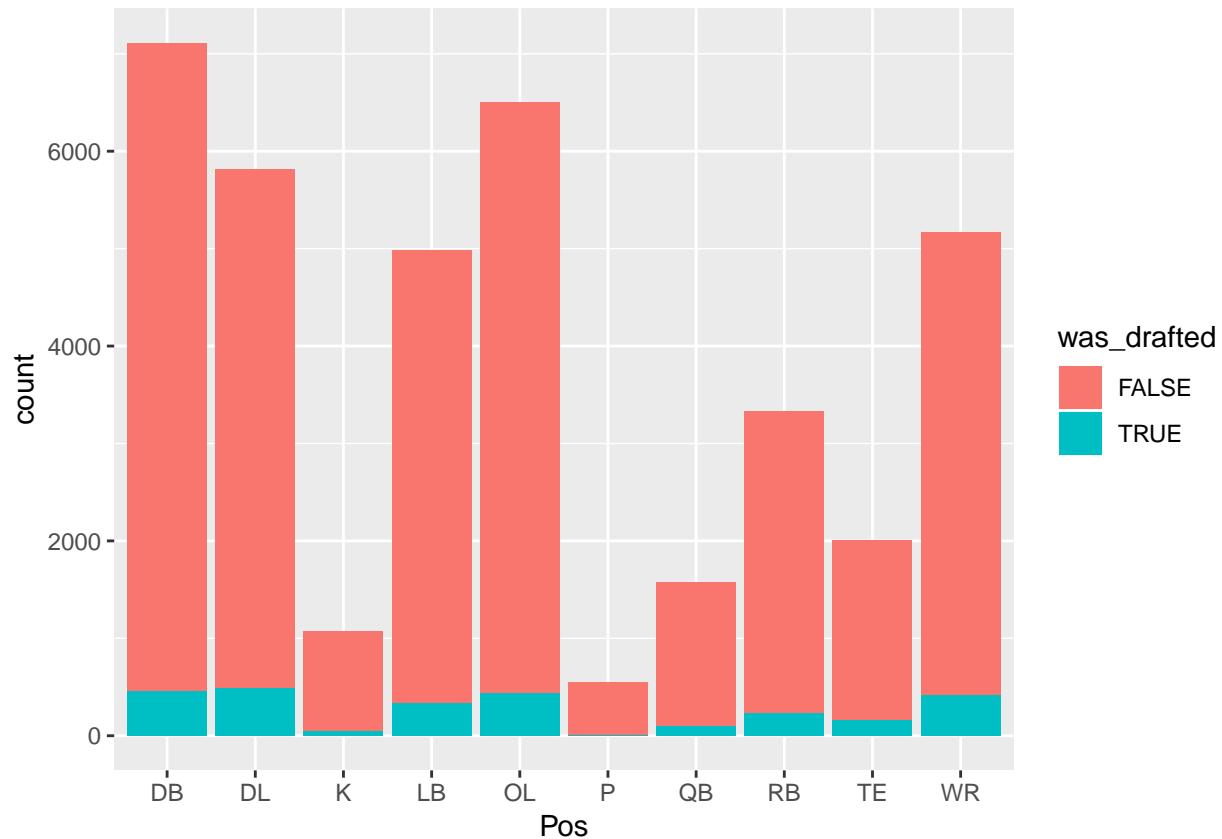
*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$

Differences in the drafting rates for position

The number of drafted players per position will come down to how many slots on the field utilize that position. Punters and kickers are one of the least drafted positions because there only needs to be one of them on the field and are not utilized very frequently. This will also mean punters and kickers will be less likely to get injured so they will have longer careers. So the need to draft a kicker or punter isn't as necessary as a

position like an Offensive Lineman or Defensive Lineman. These two positions will play every snap on offense or defense and they are always in the mix. For these positions injuries occur a lot more frequently and career length will be shorter.

```
ggplot(college_draft) + aes(x=Pos, fill = was_drafted) + geom_bar()
```



Logistic Regression

```
set.seed(122)
split <- sample.split(college_draft$was_drafted, SplitRatio = 0.75)
```

create training set

```
college_draftTrain <- subset(college_draft, split == TRUE)
nrow(college_draftTrain)
```

```
## [1] 28581
```

Run the model on Train

The subset has all colleges where at least one player was drafted

```
college_draftTrain <- subset(college_draftTrain, ave(college_draftTrain$was_drafted, college_draftTrain
```

Build the Logistic Regression Model

```
college_draftLog <- glm(was_drafted ~ GP + Pos + Yr + team_name, family = binomial(), college_draftTrain
```

Check why LS only has values for 2018

```
subset(college_draft, college_draft$Pos == "LS" & season == 2017)

## [1] X                 Jersey             Player
## [4] Yr                Pos               GP
## [7] GS                G                 Rush.Attempts
## [10] Rush.Net.Yards  Rush.YdsGained   Yds.Rush
## [13] Rush.YdsLost    RushTDs          Rush.Yds.G
## [16] Rush.Long       team_id          season
## [19] team_name        Age               Birth.Place
## [22] Birthday         Current.Status  Current.Team
## [25] Experience       Height..inches. High.School
## [28] High.School.Location Number  Player.Id
## [31] Position          Weight..lbs.   start_year
## [34] end_year          copyofname    was_drafted
## [37] GNS
## <0 rows> (or 0-length row.names)
```

Build prediction of college_draftLog

```
predict_college_draftTrain <- predict(college_draftLog, newdata = college_draftTrain, type = "response")
summary(predict_college_draftTrain)

##      Min.    1st Qu.     Median      Mean    3rd Qu.    Max.
## 0.0001443 0.0245986 0.0800930 0.1233058 0.1889975 0.6896595
```

average predicted probabilities

```
tapply(predict_college_draftTrain, college_draftTrain$was_drafted, FUN=mean)

##      FALSE      TRUE
## 0.1058062 0.2477268
```

To get the fn, fp, tn, tp

```

train_fnfp <- table(college_draftTrain$was_drafted, predict_college_draftTrain > mean(college_draftTrain))
kable(train_fnfp) %>%
  kable_styling(latex_options = "striped")

```

	FALSE	TRUE
FALSE	9885	4669
TRUE	456	1591

Sensitivity

Sensitivity is the correct number of predicted players drafted by by the total number of true positives and false negatives.

$\frac{1591}{2047}$

```
## [1] 0.777235
```

Specificity

$\frac{9885}{14554}$

```
## [1] 0.6791947
```

Precision and recall

Accuracy

$\frac{(9887 + 1616)}{(9887 + 4862 + 447 + 1616)}$

```
## [1] 0.6842137
```

ROC Curve

```
ROCRpred_Train <- prediction
```

create testing set

```

college_draftTest <- subset(college_draft, split == FALSE)
college_draftTest <- subset(college_draftTest, college_draftTest$team_name %in% college_draftTrain$team)
nrow(college_draftTest)

## [1] 5605

```

Test prediction on Test data set

```
predict_Test <- predict(college_draftLog, newdata = college_draftTest, type = "response" )
summary(predict_Test)

##      Min.   1st Qu.    Median     Mean   3rd Qu.   Max.
## 0.0003097 0.0252831 0.0850081 0.1260812 0.1941952 0.6620922

test_fnfp <- table(college_draftTest$was_drafted, predict_Test > mean(college_draftTrain$was_drafted))
kable(test_fnfp) %>%
  kable_styling(latex_options = "striped")
```

	FALSE	TRUE
FALSE	3242	1681
TRUE	133	549

Sensitivity

Sensitivity is the correct number of predicted players drafted by by the total number of true positives and false negatives.

549/682

```
## [1] 0.8049853
```

Specificity

3242/4923

```
## [1] 0.6585415
```

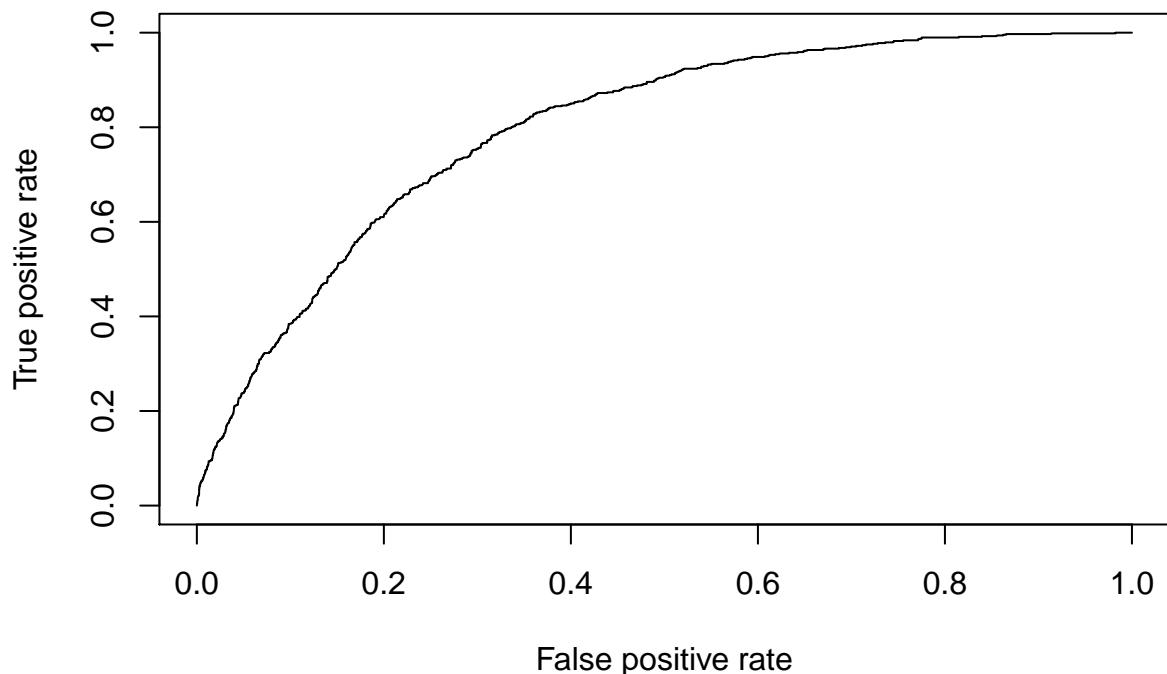
Accuracy

(3479 + 544)/(3479 + 1593 + 144 + 544)

```
## [1] 0.6984375
```

ROC Curve

```
ROCRpred_Test <- prediction(predict_Test, college_draftTest$was_drafted)
ROCRperf_Test <- performance(ROCRpred_Test, "tpr", "fpr")
plot(ROCRperf_Test)
```



Recall

```
ROCRperf_Test@x.values[[1]][1698]  
## [1] 0.3414585
```

Precision

```
ROCRperf_Test@y.values[[1]][1698]  
## [1] 0.8049853
```

Logistic Regression 2

create training set

```
college_draftTrain2 <- subset(college_draft, split == TRUE)  
nrow(college_draftTrain2)  
## [1] 28581
```

Run the model on Train2

the subset has all colleges where at least one player was drafted

```
college_draftTrain2 <- subset(college_draftTrain2, ave(college_draftTrain2$was_drafted, college_draftTrain2$team_name, FUN = mean) >= 1)
```

Build the Logistic Regression Model

```
college_draftLog2 <- glm(was_drafted ~ Pos + GP + GS + season + team_name, family = binomial(), college_draftTrain2)

## Call:
## glm(formula = was_drafted ~ Pos + GP + GS + season + team_name,
##      family = binomial(), data = college_draftTrain2)
##
## Deviance Residuals:
##    Min      1Q   Median      3Q     Max
## -2.1564  -0.4495  -0.2588  -0.1335   3.5069
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)               928.041827  51.068662 18.172 < 2e-16 ***
## PosDL                    0.456513   0.092302  4.946 7.58e-07 ***
## PosK                     0.844462   0.191915  4.400 1.08e-05 ***
## PosLB                    0.071575   0.101658  0.704 0.481384
## PosOL                   -0.117816   0.094860 -1.242 0.214237
## PosP                     -0.007299   0.314005 -0.023 0.981456
## PosQB                    0.176588   0.156437  1.129 0.258977
## PosRB                    0.561213   0.114058  4.920 8.64e-07 ***
## PosTE                    0.531944   0.130883  4.064 4.82e-05 ***
## PosWR                    0.546616   0.096253  5.679 1.36e-08 ***
## GP                       0.120579   0.011489 10.495 < 2e-16 ***
## GS                       0.186520   0.006470 28.828 < 2e-16 ***
## season                  -0.464256   0.025346 -18.317 < 2e-16 ***
## team_nameAkron           2.375305   0.766272  3.100 0.001936 **
## team_nameAlabama          4.016406   0.736351  5.454 4.91e-08 ***
## team_nameArizona          1.983391   0.774005  2.563 0.010392 *
## team_nameArkansas         3.678466   0.741445  4.961 7.01e-07 ***
## team_nameAuburn           3.759313   0.738338  5.092 3.55e-07 ***
## team_nameBaylor            3.032942   0.749407  4.047 5.19e-05 ***
## team_nameBoston College   2.825504   0.754317  3.746 0.000180 ***
## team_nameBuffalo           1.783595   0.801009  2.227 0.025968 *
## team_nameCalifornia        3.517790   0.744699  4.724 2.32e-06 ***
## team_nameCincinnati        2.682477   0.759603  3.531 0.000413 ***
## team_nameClemson           3.741679   0.737043  5.077 3.84e-07 ***
## team_nameColorado          3.201886   0.748853  4.276 1.91e-05 ***
## team_nameDuke              2.892268   0.749655  3.858 0.000114 ***
## team_nameEast Carolina     2.293801   0.764904  2.999 0.002710 **
## team_nameFlorida            4.180229   0.736486  5.676 1.38e-08 ***
```

## team_nameGeorgia	3.587902	0.741957	4.836	1.33e-06	***
## team_nameGeorgia Tech	3.081980	0.750600	4.106	4.03e-05	***
## team_nameHawaii	1.870704	0.797934	2.344	0.019056	*
## team_nameHouston	3.099576	0.750423	4.130	3.62e-05	***
## team_nameIdaho	2.058177	0.790028	2.605	0.009182	**
## team_nameIllinois	2.852853	0.754927	3.779	0.000157	***
## team_nameIndiana	2.555797	0.759687	3.364	0.000767	***
## team_nameIowa	3.460728	0.745428	4.643	3.44e-06	***
## team_nameKansas	1.809660	0.806538	2.244	0.024849	*
## team_nameKentucky	2.401452	0.769966	3.119	0.001815	**
## team_nameLouisiana Tech	2.744310	0.752310	3.648	0.000264	***
## team_nameLouisville	3.145232	0.749498	4.196	2.71e-05	***
## team_nameLSU	4.641660	0.736028	6.306	2.86e-10	***
## team_nameMarshall	1.296713	0.800222	1.620	0.105137	
## team_nameMaryland	2.955057	0.752181	3.929	8.54e-05	***
## team_nameMassachusetts	2.096793	0.784896	2.671	0.007553	**
## team_nameMemphis	2.817393	0.752942	3.742	0.000183	***
## team_nameMichigan	4.743257	0.733548	6.466	1.01e-10	***
## team_nameMinnesota	3.081855	0.748079	4.120	3.79e-05	***
## team_nameMissouri	3.329424	0.745558	4.466	7.98e-06	***
## team_nameNavy	-0.162570	1.012550	-0.161	0.872444	
## team_nameNebraska	3.684835	0.741432	4.970	6.70e-07	***
## team_nameNevada	1.706153	0.798604	2.136	0.032645	*
## team_nameNew Mexico	0.617348	0.926115	0.667	0.505028	
## team_nameNorth Carolina	3.161747	0.745645	4.240	2.23e-05	***
## team_nameNorth Texas	0.513691	0.928191	0.553	0.579968	
## team_nameNorthwestern	2.439441	0.761890	3.202	0.001366	**
## team_nameNotre Dame	3.506382	0.745461	4.704	2.56e-06	***
## team_nameOklahoma	3.295151	0.744138	4.428	9.50e-06	***
## team_nameOld Dominion	2.177165	0.785631	2.771	0.005584	**
## team_nameOregon	3.361690	0.744398	4.516	6.30e-06	***
## team_namePittsburgh	3.987388	0.741254	5.379	7.48e-08	***
## team_namePurdue	2.956836	0.755372	3.914	9.06e-05	***
## team_nameRice	1.133799	0.834672	1.358	0.174344	
## team_nameRutgers	3.067739	0.751751	4.081	4.49e-05	***
## team_nameSouth Alabama	1.590848	0.807874	1.969	0.048933	*
## team_nameSouth Carolina	3.080087	0.748850	4.113	3.90e-05	***
## team_nameStanford	3.350949	0.741628	4.518	6.23e-06	***
## team_nameSyracuse	2.062033	0.781811	2.638	0.008352	**
## team_nameTemple	3.259446	0.746223	4.368	1.25e-05	***
## team_nameTennessee	3.157560	0.749902	4.211	2.55e-05	***
## team_nameTexas	3.558541	0.744520	4.780	1.76e-06	***
## team_nameTexas A&M	3.846325	0.740794	5.192	2.08e-07	***
## team_nameTexas Tech	2.397640	0.765049	3.134	0.001725	**
## team_nameToledo	2.617545	0.760615	3.441	0.000579	***
## team_nameTroy	1.292371	0.816883	1.582	0.113632	
## team_nameTulane	2.458565	0.765203	3.213	0.001314	**
## team_nameTulsa	0.882984	0.880407	1.003	0.315896	
## team_nameUCLA	4.107585	0.737177	5.572	2.52e-08	***
## team_nameUtah	4.123134	0.738286	5.585	2.34e-08	***
## team_nameVanderbilt	3.168705	0.749819	4.226	2.38e-05	***
## team_nameVirginia	3.156757	0.750218	4.208	2.58e-05	***
## team_nameVirginia Tech	3.258518	0.746423	4.366	1.27e-05	***
## team_nameWake Forest	2.742856	0.755264	3.632	0.000282	***

```

## team_nameWashington      3.540347  0.741414  4.775 1.80e-06 ***
## team_nameWest Virginia   3.635624  0.749429  4.851 1.23e-06 ***
## team_nameWisconsin       3.296306  0.744286  4.429 9.48e-06 ***
## team_nameWyoming         2.878755  0.753894  3.819 0.000134 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 12399.1  on 16598  degrees of freedom
## Residual deviance: 9071.4  on 16513  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: 9243.4
##
## Number of Fisher Scoring iterations: 7

```

Build prediction of college_draftLog2

```

predict_college_draftTrain2 <- predict(college_draftLog2, newdata = college_draftTrain2, type = "response")
summary(predict_college_draftTrain2)

##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.    NA's
## 0.0001989 0.0189856 0.0526550 0.1233207 0.1574638 0.9022158      2

```

Get the fn, fp, tn, tp

```

table(college_draftTrain2$was_drafted, predict_college_draftTrain2 > mean(college_draftTrain2$was_drafted))

##
##          FALSE  TRUE
##    FALSE 11199 3353
##    TRUE   405 1642

```

Sensitivity

Sensitivity is the correct number of predicted players drafted by the total number of true positives and false negatives.

1642/2047

```
## [1] 0.8021495
```

Specificity

11199/14552

```
## [1] 0.7695849
```

average predicted probabilities

```
tapply(predict_college_draftTrain2, college_draftTrain2$was_drafted, FUN=mean)  
  
## FALSE TRUE  
## NA 0.3302933
```

Accuracy

```
(11291 + 1651)/(11291 + 3456 + 412 + 1651)  
  
## [1] 0.7698989
```

create testing set

```
college_draftTest2 <- subset(college_draft, split == FALSE)  
nrow(college_draftTest2)  
  
## [1] 9527
```

the subset has all colleges where at least one player was drafted in test

```
college_draftTest2 <- subset(college_draftTest2, ave(college_draftTest2$was_drafted, college_draftTest2$
```

Build test prediction of college_draftTestLog

```
predict_cdTest2 <- predict(college_draftLog2, newdata = college_draftTest2, type = "response")  
summary(predict_cdTest2)  
  
## Min. 1st Qu. Median Mean 3rd Qu. Max.  
## 0.0001989 0.0215763 0.0616054 0.1329174 0.1777606 0.9011971
```

To get the fn, fp, tn, tp

```
table(college_draftTest2$was_drafted, predict_cdTest2 > mean(college_draftTest2$was_drafted))  
  
##  
## FALSE TRUE  
## FALSE 3418 1120  
## TRUE 155 527
```

Sensitivity

Sensitivity is the correct number of predicted players drafted by by the total number of true positives and false negatives.

```
527/682
```

```
## [1] 0.7727273
```

Specificity

```
3418/4538
```

```
## [1] 0.7531952
```

average predicted probabilities on Test

```
tapply(predict_cdTest2, college_draftTest2$was_drafted, FUN=mean)
```

```
##      FALSE      TRUE  
## 0.1034471 0.3290114
```

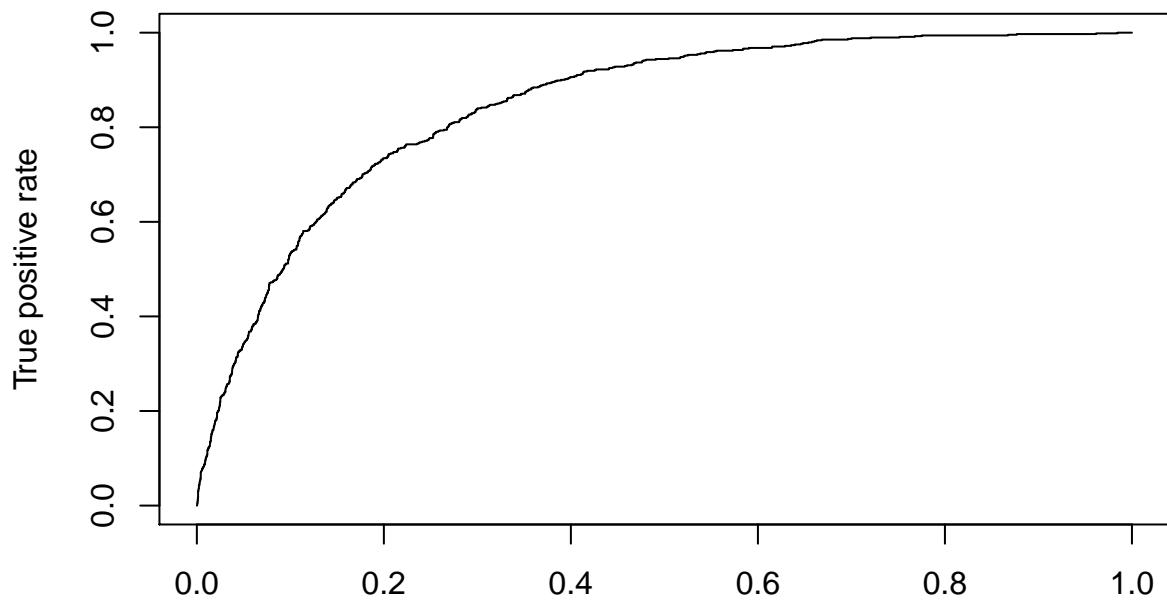
Accuracy

```
(3811 + 558)/(3811 + 1181 + 130 + 558)
```

```
## [1] 0.7691901
```

ROC Curve

```
ROCRpred_Test2 <- prediction(predict_cdTest2, college_draftTest2$was_drafted)  
ROCRperf_Test2 <- performance(ROCRpred_Test2, "tpr", "fpr")  
plot(ROCRperf_Test2)
```



```
##  
Recall  
ROCRperf_Test2@x.values[[1]] [1613]  
## [1] 0.2626708
```

Precision

```
ROCRperf_Test2@y.values[[1]] [1613]  
## [1] 0.7932551
```

Trying to stack ROC plots

```
plot(ROCRperf_Test, col = 552, lty = 1, main = "ROC")  
plot(ROCRperf_Test2, col = 24 , lty = 2, add = TRUE)
```

ROC

