NFL Project

Eric Young

11/19/2019

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 succesful 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 in to 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 tallent. 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 likelyhood of gettign 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)
library(margins)
```

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, Moshood	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, Moshood	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, Sophmore, 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

Basic Stats (NFL Data)

Let's check the data and basic information on basic_stats

```
basic_stats <- separate(basic_stats, Birth.Place, c("City", "State"), sep = ",")
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 12 rows [2808,
## 3526, 4383, 5358, 6752, 7919, 9005, 10143, 12049, 13314, 14579, 14760].
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 2799 rows
## [9, 13, 18, 30, 32, 46, 55, 57, 59, 68, 69, 72, 80, 85, 89, 93, 94, 96, 98,
## 124, ...].
```

-xtable turn summary into a table

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

	X	Age	City	State	Birthday	College	Current.Status
	Min. : 1	Min. : 19.00	Length:17172	Length:17172	Length:17172	Length:17172	Length:17172
	1st Qu.: 4294	1st Qu.: 28.00	Class :character	Class :character	Class :character	Class :character	Class :character
	Median : 8586	Median : 39.00	Mode :character	Mode :character	Mode :character	Mode :character	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"))
```

	Current.Team	Experience	Height..inches.	High.School	High.School.Location	Name	Number
	Length:17172	Min. : 0.000	Min. :61.00	Length:17172	Length:17172	Length:17172	Min. : 1.00
	Class :character	1st Qu.: 1.000	1st Qu.:72.00	Class :character	Class :character	Class :character	1st Qu.:26.00
	Mode :character	Median : 3.000	Median :74.00	Mode :character	Mode :character	Mode :character	Median :52.00
	NA	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

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

	Player.Id	Position	Weight..lbs.	start_year
	Length:17172	Length:17172	Min. : 1.0	Min. :1920
	Class :character	Class :character	1st Qu.:195.0	1st Qu.:1961
	Mode :character	Mode :character	Median :220.0	Median :1986
	NA	NA	Mean :229.2	Mean :1979
	NA	NA	3rd Qu.:255.0	3rd Qu.:2002
	NA	NA	Max. :375.0	Max. :2016
	NA	NA	NA's :51	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. Relabled positions and statndardized them.

- For example WILL is a type of Line Backer so I relabled 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.

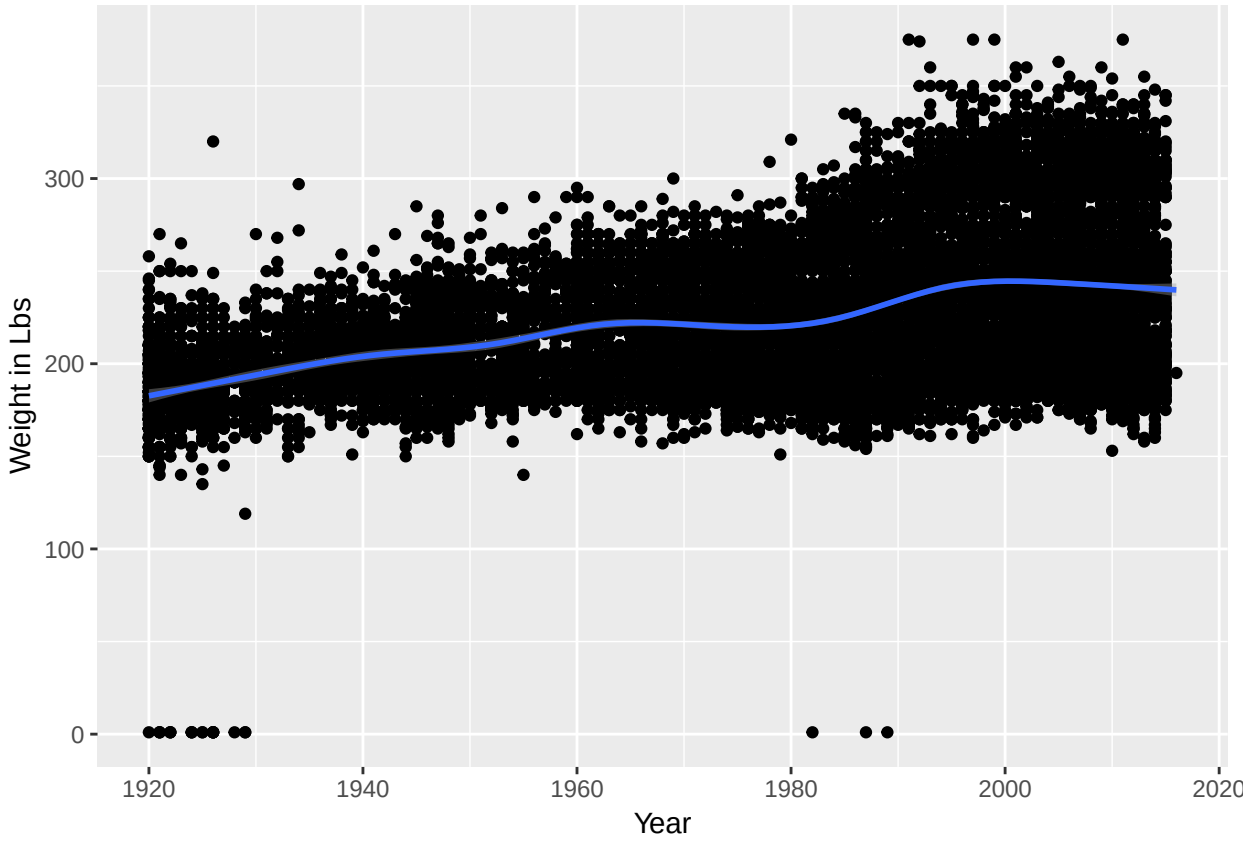
1. Create logical variable of TRUE or FALSE if the player was drafted into the NFL.
2. Sanitized the column names and stripped out invalid characters.
3. Remove comma from rushing yards variables.
4. Change games played and games started variables to numeric.
5. Created a new variable games not started.
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 200 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")
```

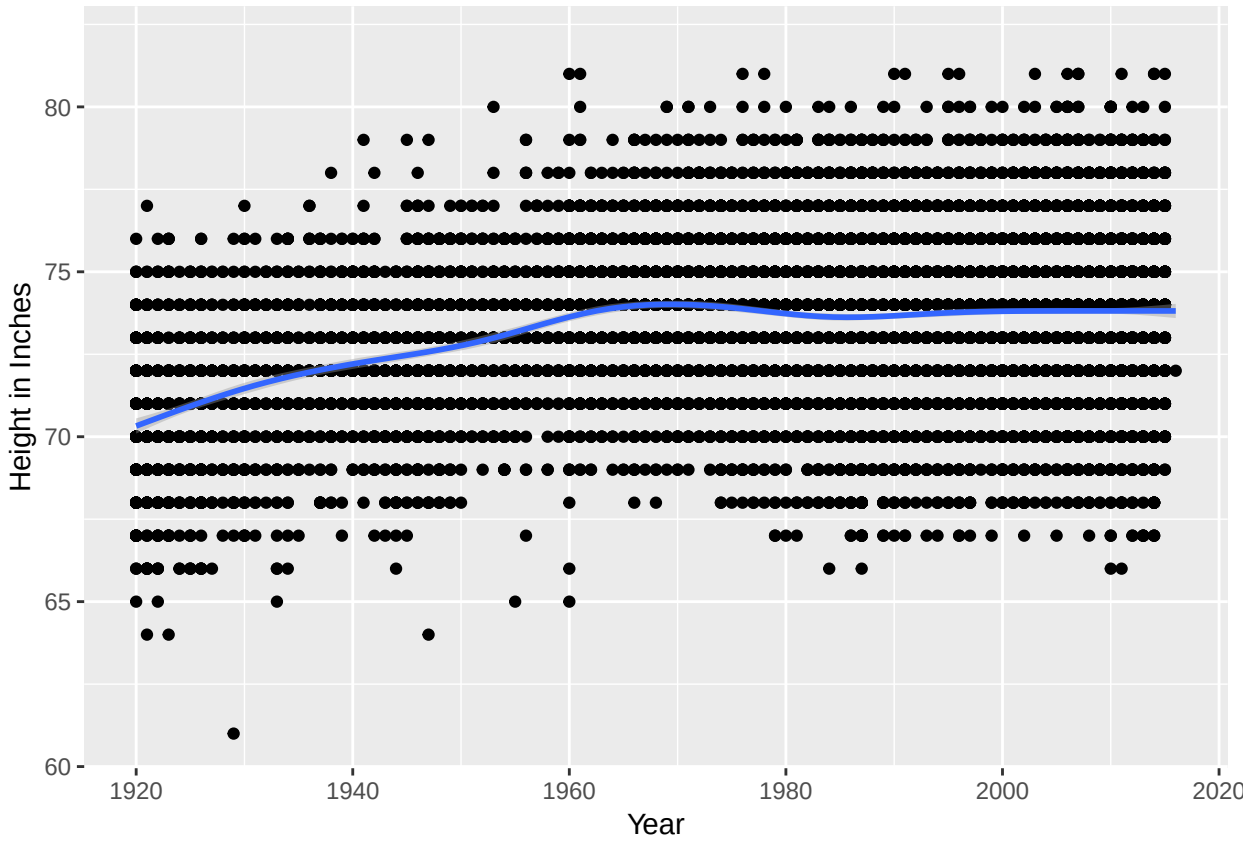
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## Warning: Removed 3147 rows containing non-finite values (stat_smooth).
## Warning: Removed 3147 rows containing missing values (geom_point).
```



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.

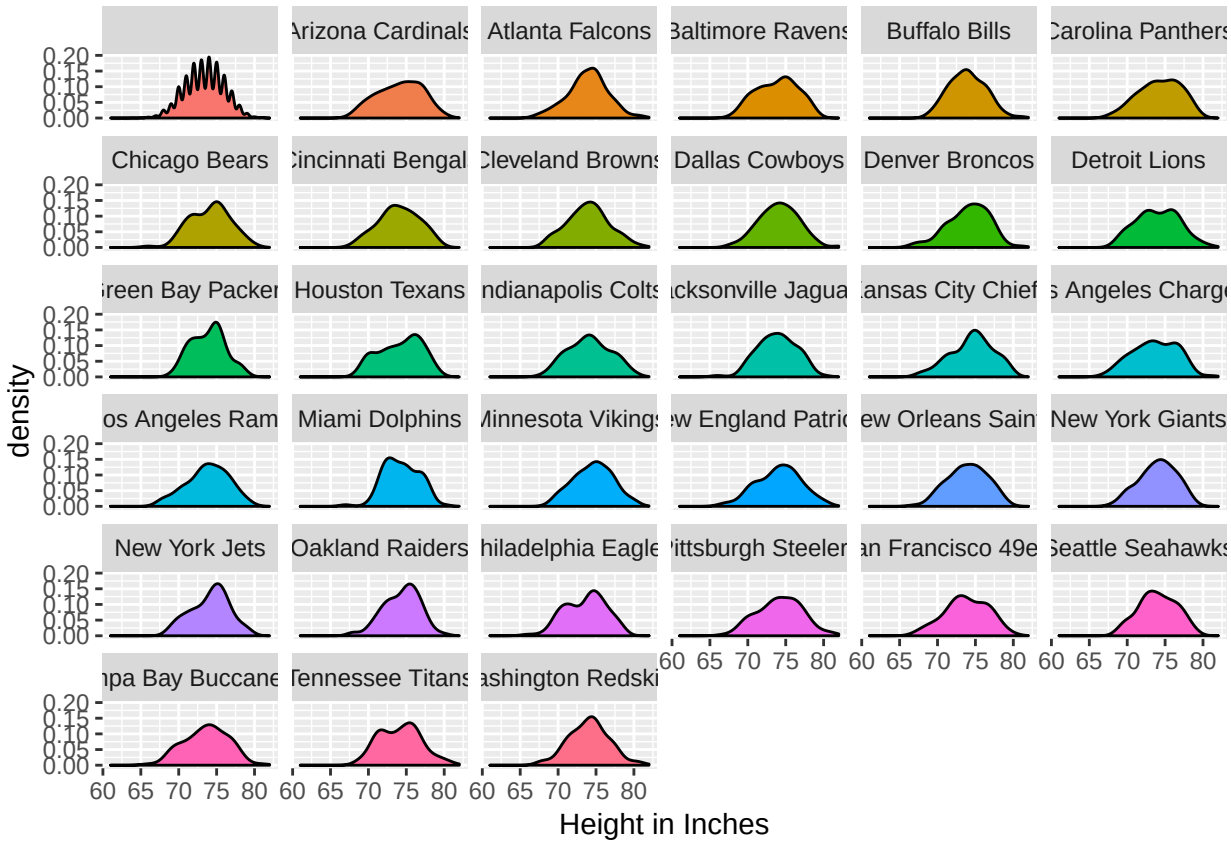
What is interesting about the chart below is all of the plots are evely placed and staggered. The reason for this is when a team records the height of a player they don't measue in quarter or half of an inch they will round up to the nearest inch.

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



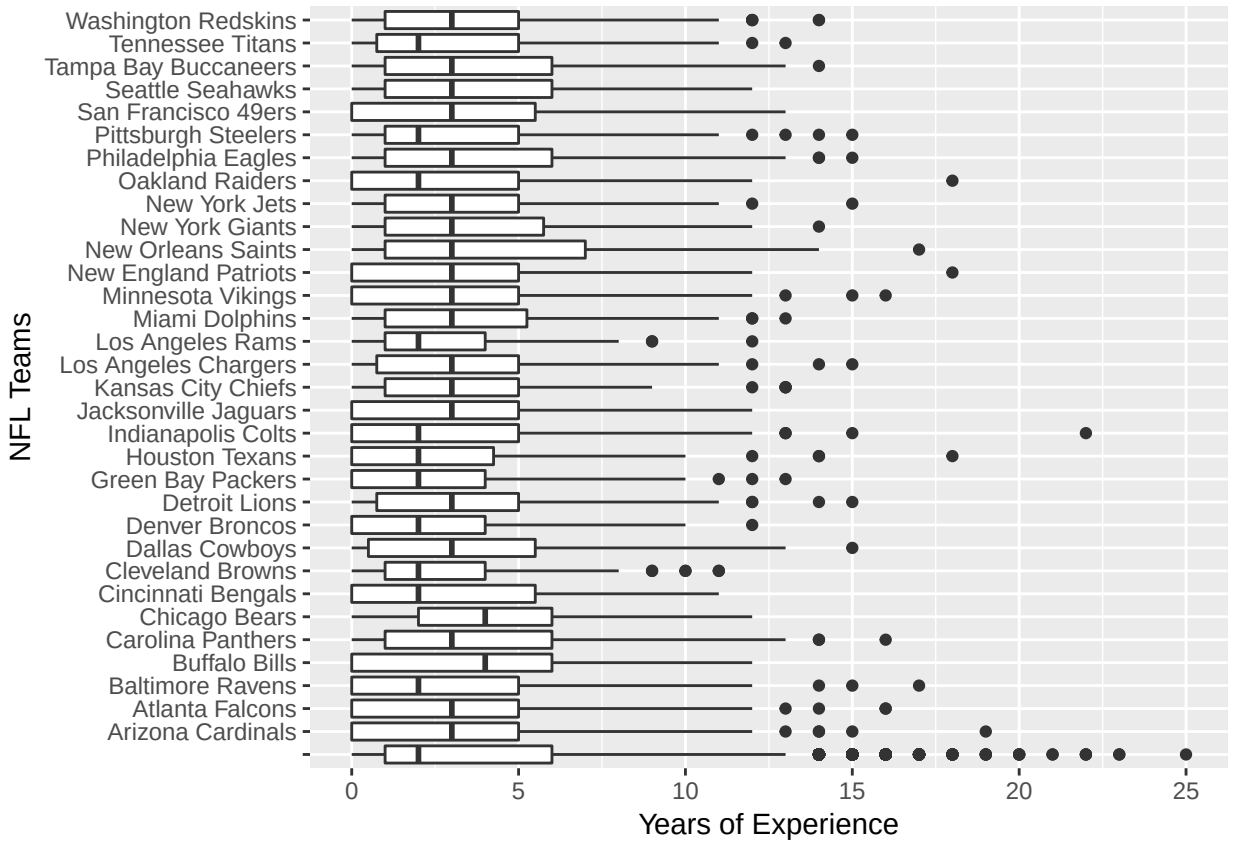
The density plot below shows the difference in height between all of the NFL teams. The most common height in the NFl seems to be around 75 inches. Players who are drafted and play in the NFL are usually between 70 and 77 inches in height. This means if you are below 70 inches it is still possible to 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 oservation, 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.

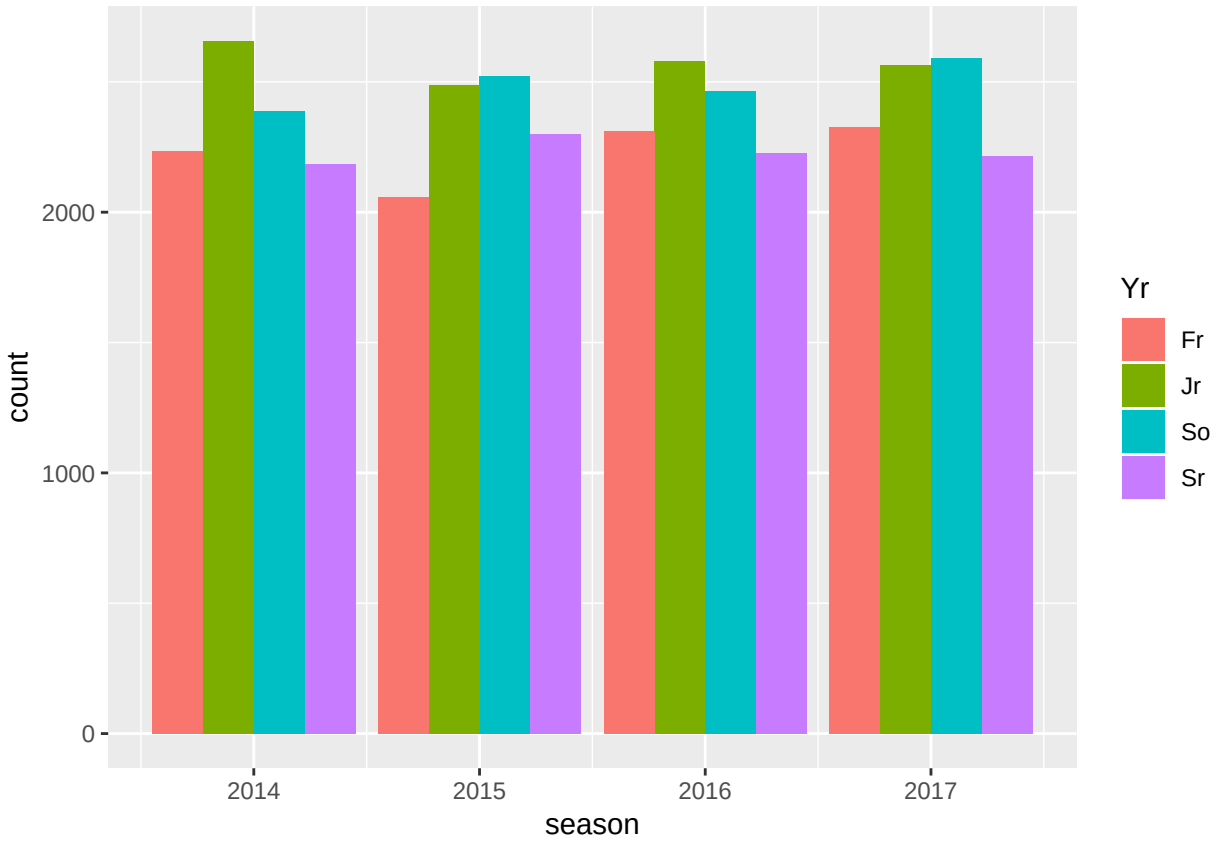
```
ggplot(basic_stats, aes(Current.Team, Experience)) +
  geom_boxplot() +
  labs(x = "NFL Teams", y = "Years of Experience") +
  coord_flip()
```



NCAA Plots

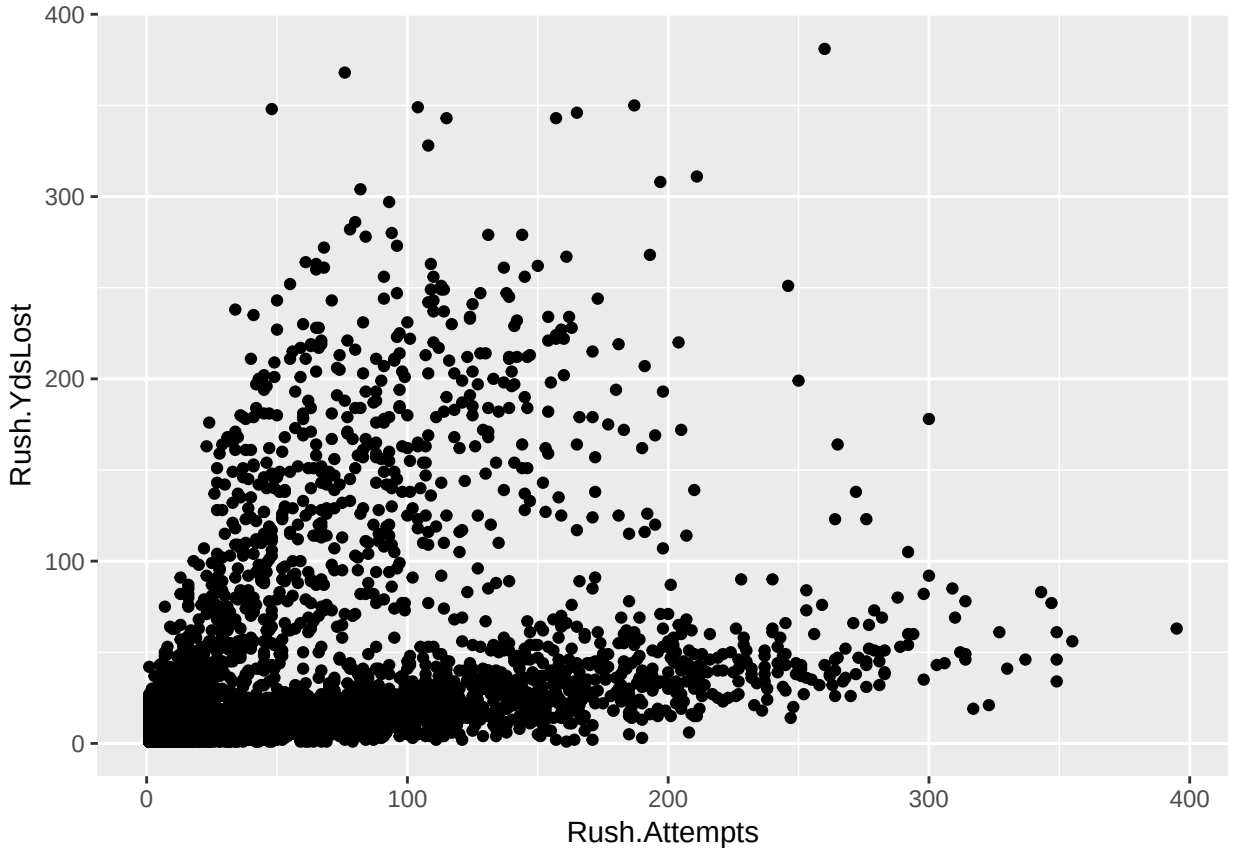
This bar plot shows the distribution of players by year, season and the progression of players from Freshman, Sophmore, Junior, and Senior by year. The amount of players progressing from one year to the next is never the same for any of the years. Senior year is always lower than the previous Junior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

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



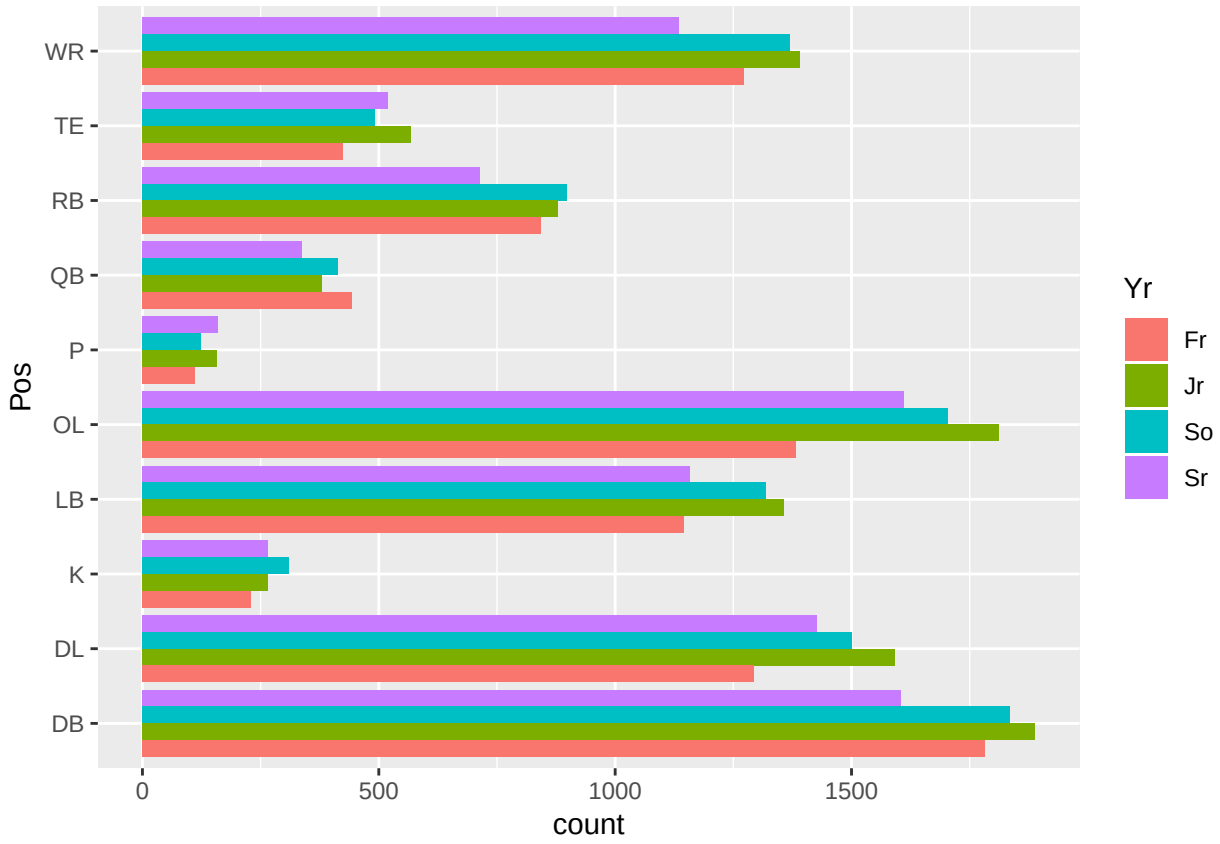
The below chart shows by the number of attempts how many yards are lost. The natural thought would be the more rushing attempts a player has the more likely they are to lose yardage. To an extent this is true. However we see in the scatter plot that this isn't exactly true. We can see players who have 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 senario 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. Where as 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

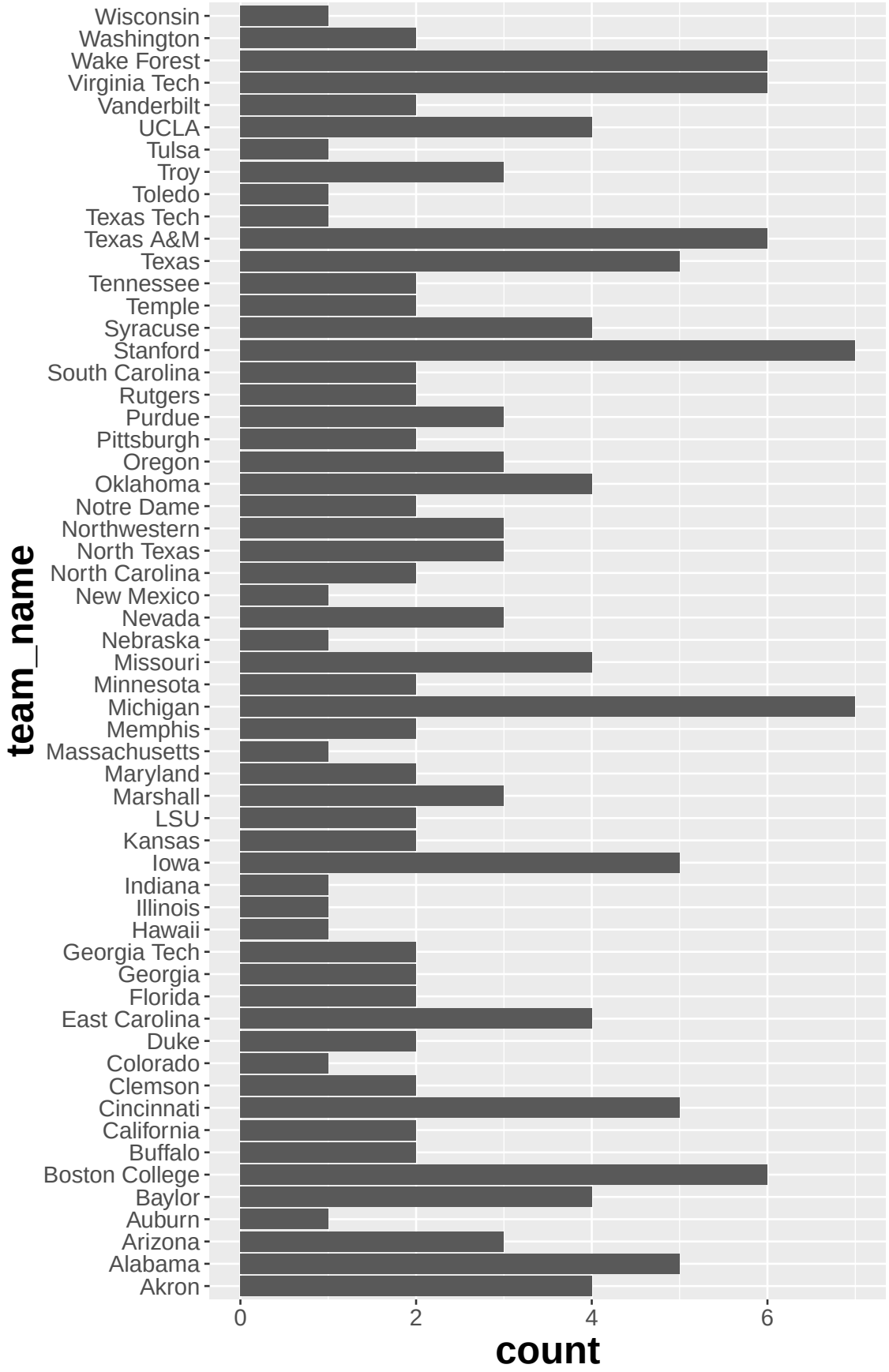
```
college_teams <- inner_join(collegedf, basic_stats, by = c("team_name" = "College", "Player" = "Name"))
```

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 toal 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
```


Wisconsin
Washington
Wake Forest
Virginia Tech
Vanderbilt
UCLA
Tulsa
Troy
Toledo
Texas Tech
Texas A&M
Texas
Tennessee
Temple
Syracuse
Stanford
South Carolina
Rutgers
Purdue
Pittsburgh
Oregon
Oklahoma
Notre Dame
Northwestern
North Texas
North Carolina
New Mexico
Nevada
Nebraska
Missouri
Minnesota
Michigan
Memphis
Massachusetts
Maryland
Marshall
LSU
Kansas
Iowa
Indiana
Illinois
Hawaii
Georgia Tech
Georgia
Florida
East Carolina
Duke
Colorado
Clemson
Cincinnati
California
Buffalo
Boston College
Baylor
Auburn
Arizona
Alabama
Akron
team_name
0
2
4
6
count

Linear Regression

Linear Regression Model for experience

Using a linear model I wanted to see if we could predict experience or the number of years a player would be in the NFL. What this model does is predicts the years of experience by college.

```
basic_stats <- mutate(basic_stats, College = sub("&", " ", College, ignore.case = TRUE))
nfl_model <- lm(Experience ~ College -1, data = basic_stats)
AIC(nfl_model)
```

```
## [1] 92332.73
```

The texreg of the model shows the relative experience by college.

```
broom::tidy(nfl_model) %>% top_n(5, estimate)
```

A tibble: 5 x 5

term estimate std.error statistic p.value 1 CollegeCentral Coll. (Iowa) 12 3.49 3.44 5.78e-4 2 CollegeCheyney 12 3.49 3.44 5.78e-4 3 CollegeSonoma State 15 3.49 4.30 1.69e-5 4 CollegeWalla Walla CC WA 12 3.49 3.44 5.78e-4 5 CollegeWestern Nebraska CC-Scottsb~ 14 3.49 4.02 5.94e-5

```
broom::tidy(nfl_model) %>% top_n(5, -estimate)
```

A tibble: 16 x 5

term estimate std.error statistic p.value 1 CollegeChattanooga State 0 3.49 0 1 2 CollegeConcordia (Quebec) 0 3.49 0 1 3 CollegeFaulkner University 0 3.49 0 1 4 CollegeGrand Rapids CC MI 0 3.49 0 1 5 CollegeGreenville College 0 3.49 0 1 6 CollegeKennesaw St. (GA) 0 3.49 0 1 7 CollegeKentucky Wesleyan 0 3.49 0 1 8 CollegeLake Erie College 0 3.49 0 1 9 CollegeLaval University 0 3.49 0 1 10 CollegeNorth Carolina-Charlotte 0 1.74 0 1 11 CollegeNortheast Mississippi CC 0 3.49 0 1 12 CollegeStetson 0 3.49 0 1 13 CollegeTexas-Permian Basin 0 3.49 0 1 14 CollegeVictor Valley 0 3.49 0 1 15 CollegeVirginia Commonwealth 0 2.46 0 1 16 CollegeWalsh 0 3.49 0 1

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

```
## [1] 199823.1
```

Second Regression Model for experience

The second linear model uses weight in lbs and college to predict the experince of a player.

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

```
## [1] 92009.56
```

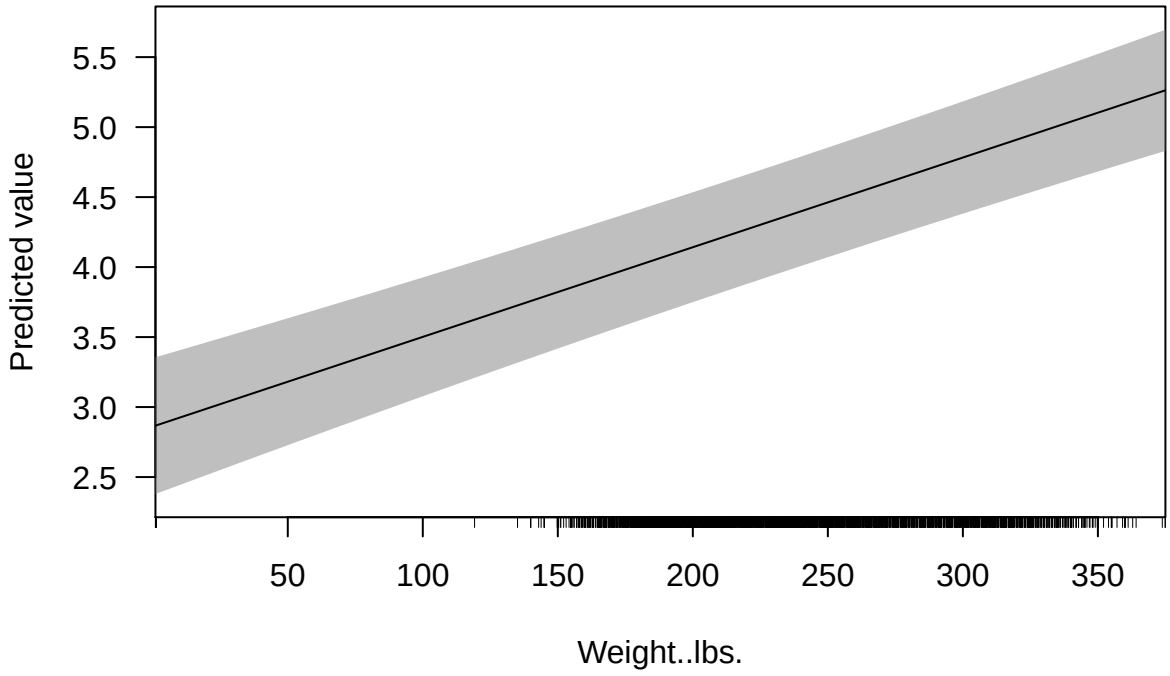
```
SSE2 <- sum(nfl_model2$residuals^2)
SSE2
```

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

From the plot below we see that this model is predicting the heavier the player the longer they will play in the NFL. This is not an accurate model for predicting experience for a player.

```
cplot(nfl_model2, "Weight..lbs.")
```

```
##        xvals    yvals    upper    lower
## 1    1.00000 2.867109 3.355368 2.378850
## 2   16.58333 2.966943 3.443487 2.490399
## 3   32.16667 3.066778 3.532169 2.601386
## 4   47.75000 3.166612 3.621455 2.711769
## 5   63.33333 3.266446 3.711388 2.821505
## 6   78.91667 3.366281 3.802012 2.930549
## 7   94.50000 3.466115 3.893373 3.038857
## 8  110.08333 3.565949 3.985513 3.146385
## 9  125.66667 3.665784 4.078478 3.253089
## 10 141.25000 3.765618 4.172309 3.358927
## 11 156.83333 3.865452 4.267045 3.463860
## 12 172.41667 3.965287 4.362720 3.567853
## 13 188.00000 4.065121 4.459365 3.670877
## 14 203.58333 4.164955 4.557002 3.772909
## 15 219.16667 4.264790 4.655649 3.873930
## 16 234.75000 4.364624 4.755316 3.973932
## 17 250.33333 4.464458 4.856002 4.072915
## 18 265.91667 4.564293 4.957702 4.170883
## 19 281.50000 4.664127 5.060401 4.267853
## 20 297.08333 4.763961 5.164079 4.363844
```



Third Regression Model for experience

The third linear model uses height in inches, weight in lbs, and I(Weight..lbs.^2) + Position to predict the experince of a player.

```
nfl_model3 <- lm(Experience ~ Height..inches. + Weight..lbs. + I(Weight..lbs.^2) + Position, data = bas
AIC(nfl_model3)
```

```
## [1] 90702.7
```

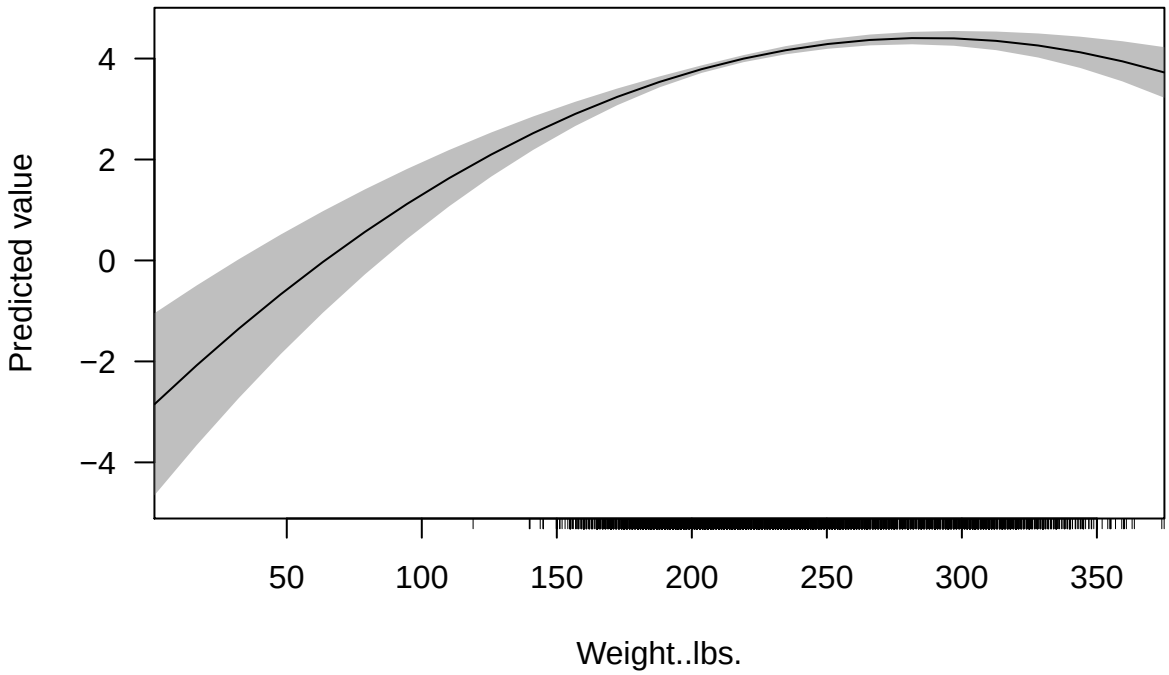
```
SSE3 <- sum(nfl_model3$residuals^2)
SSE3
```

```
## [1] 204675.8
```

From the weight plot below we see that this model is showing players who are between 250 and 300 lbs have the longest longevity in the NFL. Above 300 lbs the players eperience starts to fall. From the height plot below it shows the taller the player the more experince it is expected a player will have. This is not a practical assumption similar to model 2, it's not necesarily true the taller a player is the more years a player will have in the NFL.

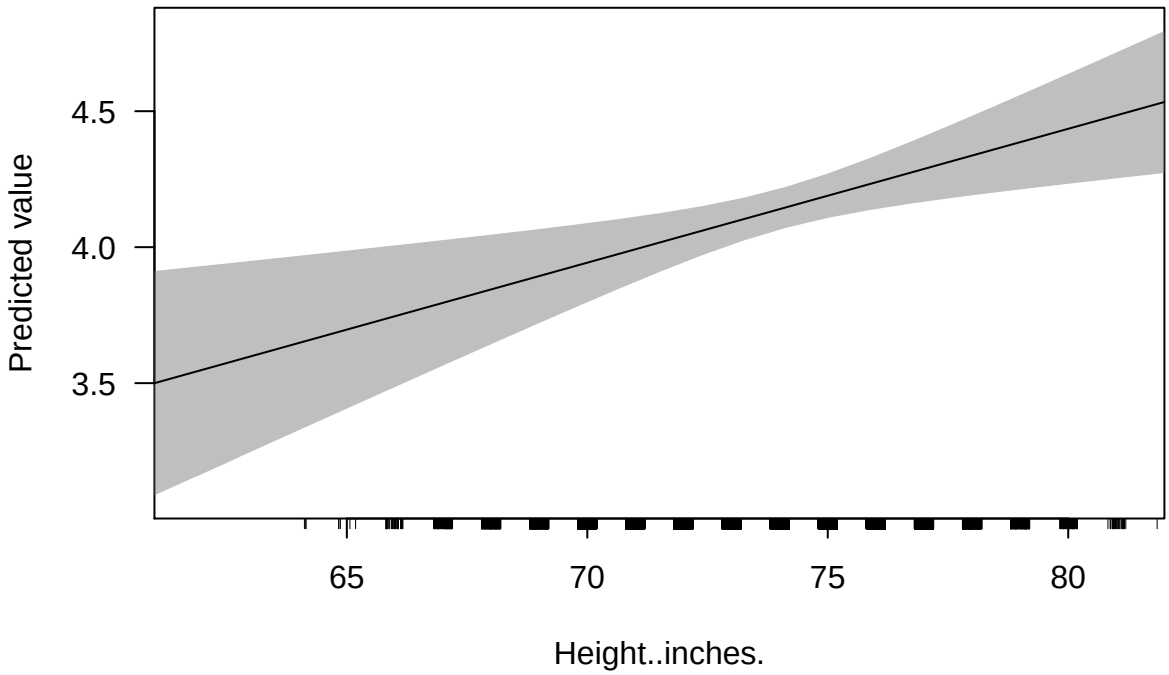
```
cplot(nfl_model3, "Weight..lbs.")
```

```
##        xvals       yvals       upper      lower
## 1    1.00000 -2.84908449 -1.04595582 -4.6522132
## 2   16.58333 -2.08016453 -0.49771549 -3.6626136
## 3   32.16667 -1.35429163  0.02174862 -2.7303319
## 4   47.75000 -0.67146579  0.51243587 -1.8553674
## 5   63.33333 -0.03168701  0.97434780 -1.0377218
## 6   78.91667  0.56504470  1.40749034 -0.2774009
## 7   94.50000  1.11872936  1.81187818  0.4255805
## 8  110.08333  1.62936695  2.18754388  1.0711900
## 9  125.66667  2.09695748  2.53455774  1.6593572
## 10 141.25000  2.52150095  2.85307401  2.1899279
## 11 156.83333  2.90299736  3.14344843  2.6625463
## 12 172.41667  3.24144671  3.40656595  3.0763275
## 13 188.00000  3.53684900  3.64479861  3.4288994
## 14 203.58333  3.78920422  3.86390139  3.7145070
## 15 219.16667  3.99851238  4.06828261  3.9287422
## 16 234.75000  4.16477348  4.24582183  4.0837251
## 17 250.33333  4.28798752  4.38264889  4.1933262
## 18 265.91667  4.36815450  4.47562149  4.2606875
## 19 281.50000  4.40527442  4.52802054  4.2825283
## 20 297.08333  4.39934727  4.54588388  4.2528107
```




```
cplot(nfl_model3, "Height..inches.")
```

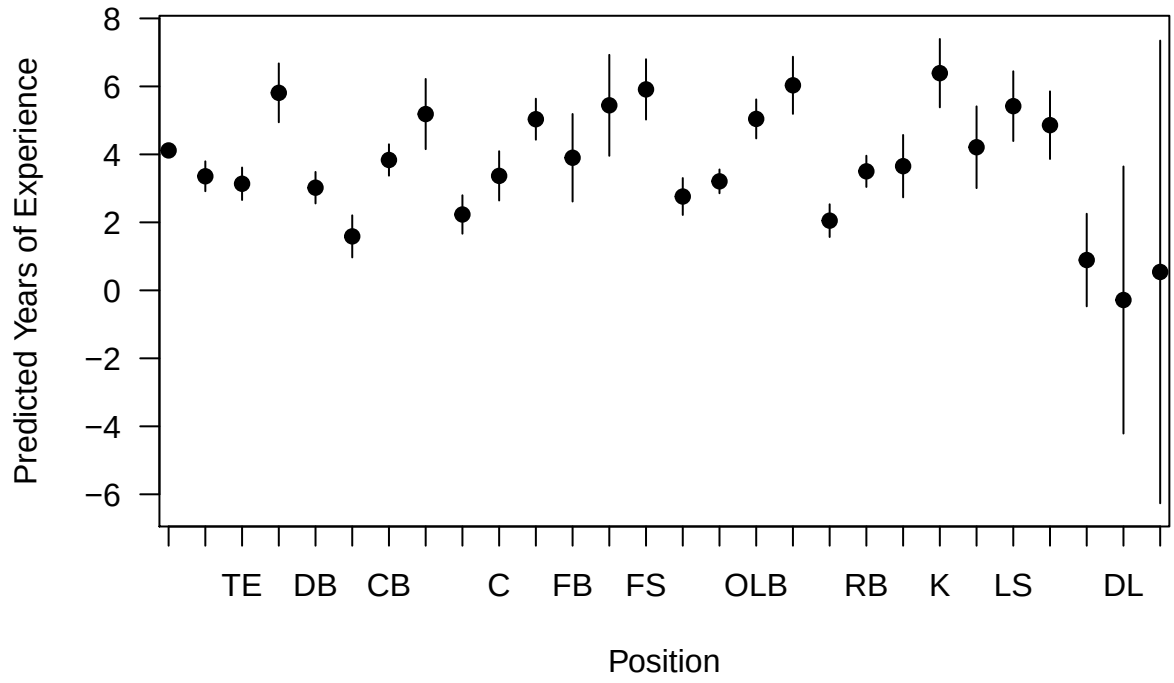
```
##     xvals    yvals    upper    lower
## 1  61.000 3.499920 3.912242 3.087598
## 2  61.875 3.542996 3.928496 3.157496
## 3  62.750 3.586073 3.944825 3.227320
## 4  63.625 3.629149 3.961246 3.297053
## 5  64.500 3.672226 3.977783 3.366669
## 6  65.375 3.715302 3.994469 3.436136
## 7  66.250 3.758379 4.011350 3.505408
## 8  67.125 3.801456 4.028496 3.574415
## 9  68.000 3.844532 4.046006 3.643058
## 10 68.875 3.887609 4.064042 3.711176
## 11 69.750 3.930685 4.082860 3.778510
## 12 70.625 3.973762 4.102904 3.844619
## 13 71.500 4.016838 4.124961 3.908716
## 14 72.375 4.059915 4.150441 3.969389
## 15 73.250 4.102991 4.181678 4.024305
## 16 74.125 4.146068 4.221434 4.070702
## 17 75.000 4.189144 4.270756 4.107533
## 18 75.875 4.232221 4.327787 4.136655
## 19 76.750 4.275298 4.389741 4.160854
## 20 77.625 4.318374 4.454586 4.182162
```



From the plot below we see the predicted years if experience for positions based on the weight in lbs. I should note that a predicted value cannot be negative as a player cannot play negative time. For the defensive lineman it shows a negative to a positive predicted years of experience and has a median predicted value at 0 years of experience. From this example we gather that a DL will most likely have a short career in the NFL.

```
cplot(nfl_model3, x = "Position", dx = "Weight..lbs.", ylab = "Predicted Years of Experience")
```

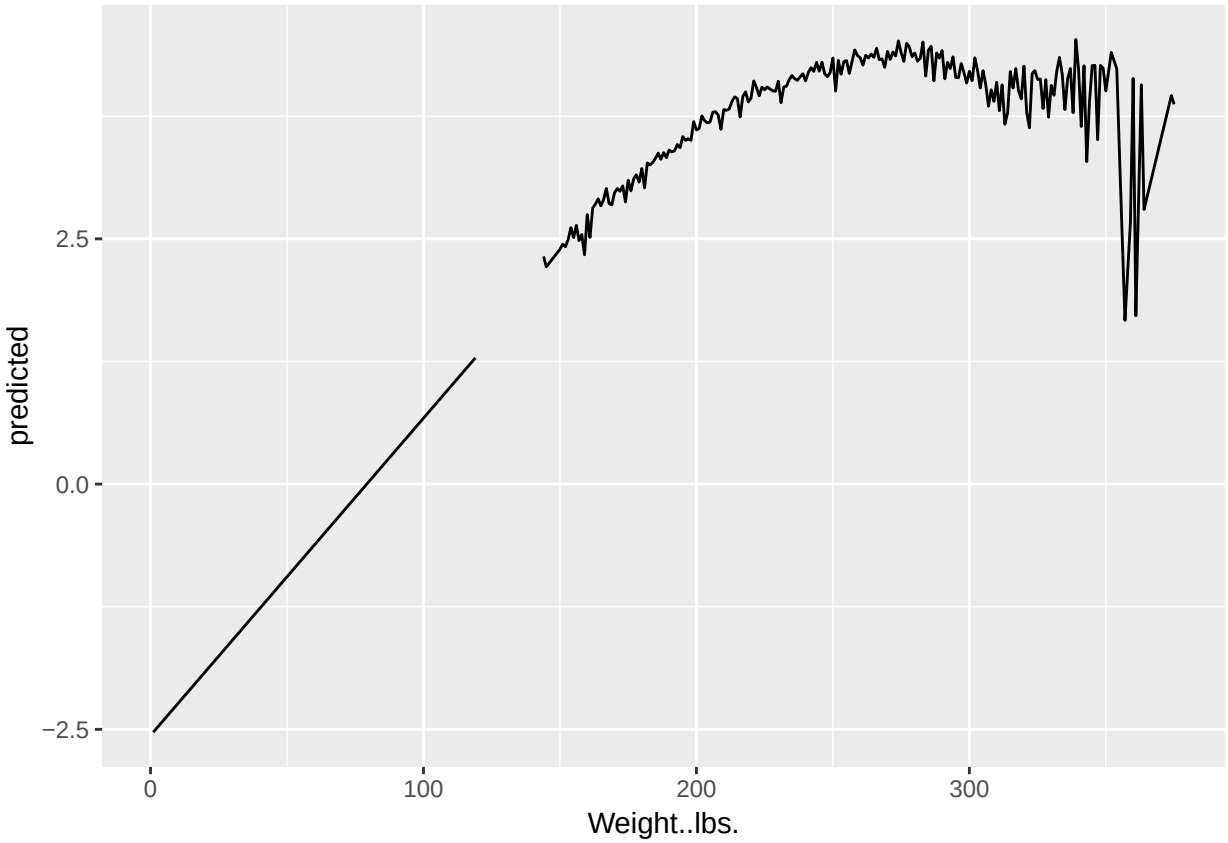
```
##    xvals    yvals    upper     lower
## 1        4.115724 4.192438 4.0390103
## 2     DE 3.355825 3.795100 2.9165502
## 3     TE 3.135628 3.613041 2.6582159
## 4     SS 5.810584 6.675610 4.9455585
## 5     DB 3.020197 3.482810 2.5575844
## 6     OT 1.587787 2.206845 0.9687291
## 7     CB 3.835550 4.296887 3.3742141
## 8      G 5.186364 6.218715 4.1540134
## 9     OG 2.232041 2.796802 1.6672809
## 10     C 3.367964 4.092346 2.6435816
## 11    QB 5.034986 5.637520 4.4324515
## 12    FB 3.900825 5.187565 2.6140846
## 13   MLB 5.443283 6.927889 3.9586767
## 14    FS 5.911734 6.798881 5.0245874
## 15    DT 2.760861 3.303587 2.2181344
## 16    WR 3.206678 3.556104 2.8572529
## 17   OLB 5.043570 5.616969 4.4701703
## 18     T 6.032191 6.872329 5.1920533
## 19    LB 2.050706 2.531878 1.5695342
## 20    RB 3.500877 3.959623 3.0421313
```



Create new variable predicted

```
predicted <- predict(nfl_model3)
ggplot(basic_stats %>% mutate(predicted = predict(nfl_model3, basic_stats)) %>% group_by(Weight..lbs.) %
```

```
## Warning: Removed 1 rows containing missing values (geom_path).
```



Fourth Regression Model for experience

The fourth linear model uses height in inches and Weight..lbs. * Position to predict the experince of a player.

```
nfl_model4 <- lm(Experience ~ Height..inches. + Weight..lbs. * Position, data = basic_stats)
AIC(nfl_model4)
```

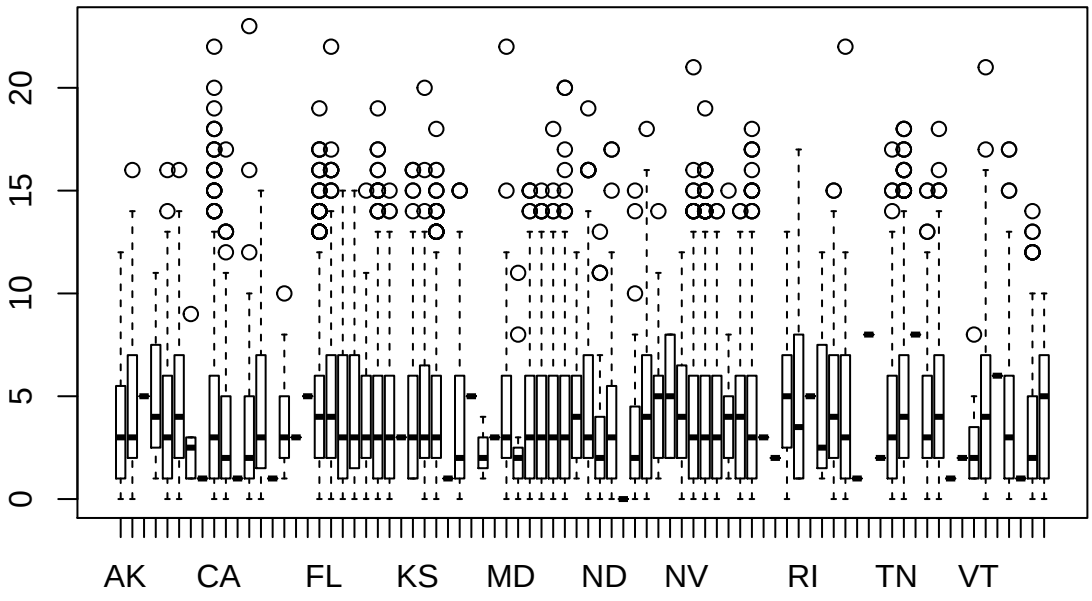
```
## [1] 90677.07
```

```
SSE4 <- sum(nfl_model4$residuals^2)
SSE4
```

```
## [1] 203768.4
```

The below box and whiskers plot which shows experience by state. It looks like the median experience is about 3 - 4 years for most states. The third quartile is generally from 5-7 years of experience. Which means most players will have retired between 5 and 7 years in the NFL.

```
boxplot(Experience~State, basic_stats)
```



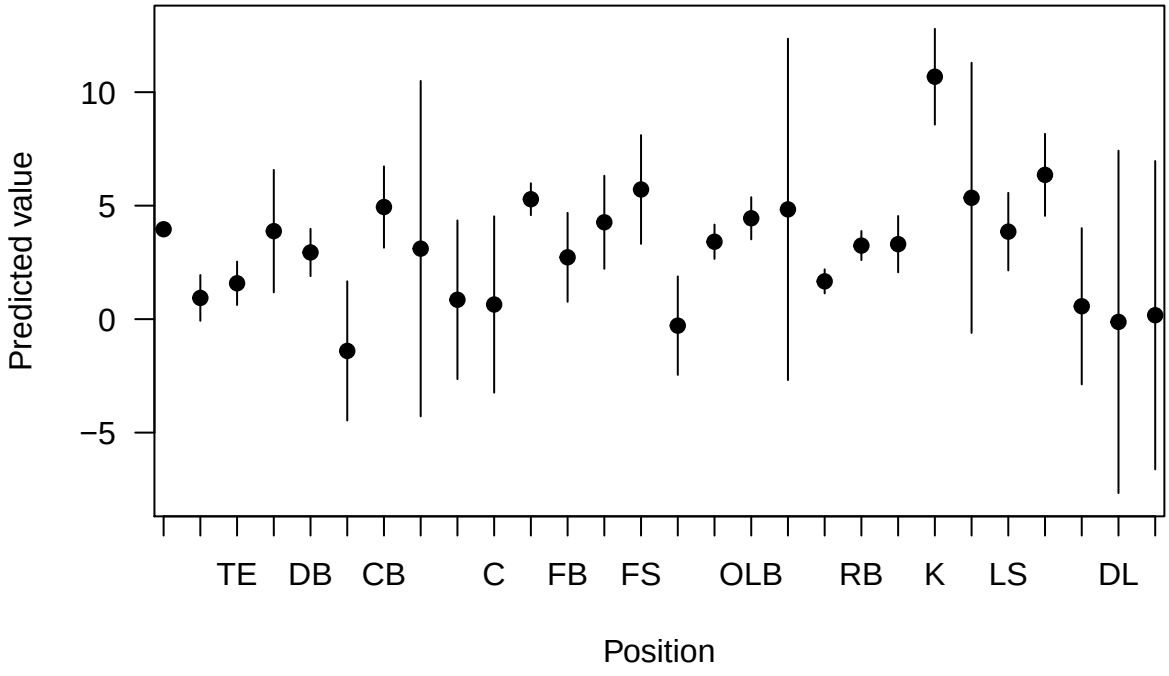
From the plot below we see the predicted years if experience for different positions. I should note that a predicted value cannot be negative as a player cannot play negative time. For the center back it shows 3 to 6 years of predicted experience and is predicted at 5 years of experience. From this example we gather that a CB will most likely have a lengthy career in the NFL.

```
cplot(nfl_model4, "Position")
```

```
## Warning in predict.lm(model, newdata = datalist, type = type, se.fit =
## TRUE, : prediction from a rank-deficient fit may be misleading
```

```
##    xvals      yvals     upper       lower
## 1         3.9609438  4.018667  3.90322041
## 2     DE  0.9343312  1.941890 -0.07322723
## 3     TE  1.5793875  2.535291  0.62348377
## 4     SS  3.8771678  6.573635  1.18070098
## 5     DB  2.9380458  3.980700  1.89539110
## 6     OT -1.4009231  1.665373 -4.46721880
## 7     CB  4.9390579  6.730510  3.14760570
## 8      G  3.1054842 10.494582 -4.28361301
```

```
## 9     OG  0.8538760  4.349599 -2.64184681
## 10     C  0.6451968  4.530779 -3.24038539
## 11    QB  5.2837130  5.986720  4.58070645
## 12    FB  2.7239310  4.681906  0.76595622
## 13   MLB  4.2677146  6.317144  2.21828544
## 14    FS  5.7122235  8.107165  3.31728180
## 15    DT -0.2871132  1.882539 -2.45676525
## 16    WR  3.4103999  4.166240  2.65455944
## 17   OLB  4.4444550  5.372566  3.51634377
## 18     T  4.8357451 12.357278 -2.68578770
## 19    LB  1.6646656  2.193116  1.13621552
## 20    RB  3.2407658  3.881741  2.59979075
```



Summary of Linear Models by Experience

According to SSE the best model was weight and college with an SSE of 198610.6. However, when I used AIC it preferred the model height and weight by position with a value of 90677.07.

Linear Model for drafting

For the first linear model to predict whether a player was drafted I used the dependent variable was_drafted and independent variables Year, Position, Games Played, and Games Started. From the table below we see the linear model draft1 Games played and Games Started are statistically significant. Games Started is more significant than Games Played. For Year variable Senior Year is statistically significant and Junior Year is almst statistically significant. The positions which are statistically significant are DL, K, RB, TE, and WR. Something to note is the prediction is based off of the DB position which may explain why DL is statistically significant and not a OL or QB.

```
draft1 <- lm(was_drafted ~ Yr + Pos + GP + GS, data = college_draft)
```

```r
summary(draft1)
```

```
##
## Call:
## lm(formula = was_drafted ~ Yr + Pos + GP + GS, data = college_draft)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.25036 -0.10185 -0.03785 -0.01220  1.01751
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0183544  0.0046283  -3.966 7.33e-05 ***
## YrJr         0.0065142  0.0037097   1.756 0.079101 .
## YrSo         0.0004666  0.0036653   0.127 0.898702
## YrSr         0.0135621  0.0039434   3.439 0.000584 ***
## PosDL        0.0203881  0.0043987   4.635 3.58e-06 ***
## PosK         0.0327359  0.0082592   3.964 7.40e-05 ***
## PosLB        0.0066126  0.0045983   1.438 0.150424
## PosOL       -0.0050915  0.0042910  -1.187 0.235416
## PosP         0.0102033  0.0110991   0.919 0.357949
## PosQB        0.0072095  0.0069873   1.032 0.302168
## PosRB        0.0267512  0.0052410   5.104 3.34e-07 ***
## PosTE        0.0269674  0.0063008   4.280 1.87e-05 ***
## PosWR        0.0253034  0.0045508   5.560 2.71e-08 ***
## GP           0.0029702  0.0003628   8.186 2.78e-16 ***
## GS           0.0123533  0.0003253  37.978  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2487 on 38091 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.07009,    Adjusted R-squared:  0.06974
## F-statistic: 205.1 on 14 and 38091 DF,  p-value: < 2.2e-16
```

For the second linear model to predict whether a player was drafted I used the dependent variable was_drafted and independent variables Year, position, games played, rushing attempts, rushing net yards, rushing yards gained. From the table below a players Junior and Senior year are statistically significant to being drafted into the NFL. While a players Sophmore year is almost statistically significant.

```r
draft2 <- lm(was_drafted ~ Yr + Pos + GP + Rush.Attempts + Rush.Net.Yards + Rush.YdsGained, data = colle
```

```r
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)
```

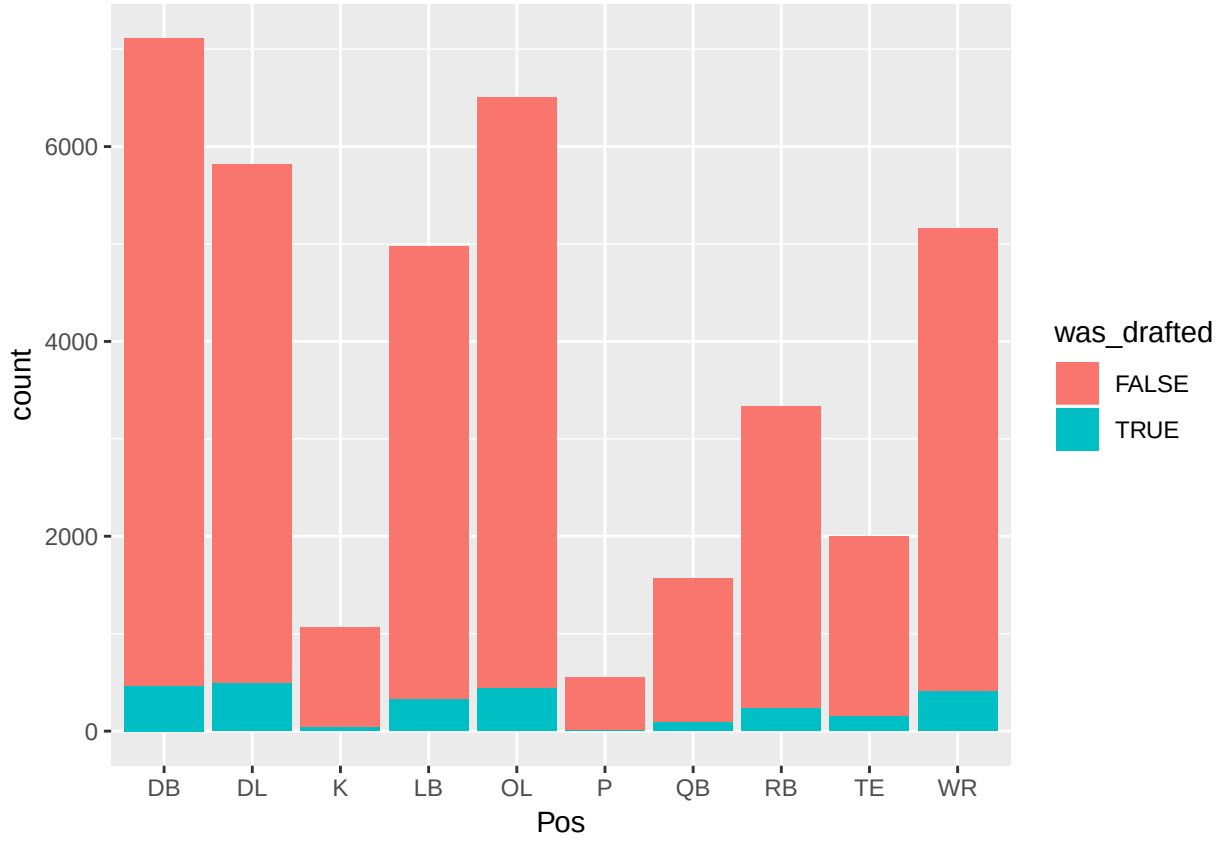
	Model 1	Model 2
(Intercept)	−0.02***	−0.01
	(0.00)	(0.03)
YrJr	0.01	0.06***
	(0.00)	(0.01)
YrSo	0.00	0.02
	(0.00)	(0.01)
YrSr	0.01***	0.08***
	(0.00)	(0.01)
PosDL	0.02***	0.01
	(0.00)	(0.06)
PosK	0.03***	0.03
	(0.01)	(0.05)
PosLB	0.01	0.05
	(0.00)	(0.05)
PosOL	−0.01	−0.03
	(0.00)	(0.09)
PosP	0.01	−0.04
	(0.01)	(0.04)
PosQB	0.01	−0.04
	(0.01)	(0.03)
PosRB	0.03***	−0.06
	(0.01)	(0.03)
PosTE	0.03***	0.04
	(0.01)	(0.04)
PosWR	0.03***	0.03
	(0.00)	(0.03)
GP	0.00***	0.01***
	(0.00)	(0.00)
GS	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	0.07	0.07
Adj. R^2	0.07	0.07
Num. obs.	38106	5427
RMSE	0.25	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

Using logistic regression I will predict whether a player will be drafted into the NFL. The outcome of the prediction will be TRUE or FALSE, TRUE if the player is drafted into the NFL and FALSE if the player is not drafted. I will use a joined data set of the NFL and NCAA data sets to run the prediction on.

```
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 defined below 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

I built the logistic regression model using the dependent variable was_drafted and the independent variables games played (GP), position (Pos), year (Yr), and team_name.

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

```
##
## Call:
## glm(formula = was_drafted ~ GP + Pos + Yr + team_name, family = binomial(),
##     data = college_draftTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5000  -0.5517  -0.3173  -0.1514   3.4087
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -8.45488    0.72778 -11.617  < 2e-16 ***
## GP                        0.24806    0.01049  23.647  < 2e-16 ***
## PosDL                     0.32509    0.08592   3.784 0.000154 ***
## PosK                     -0.42097    0.18530  -2.272 0.023098 *
## PosLB                    -0.09078    0.09452  -0.960 0.336879
## PosOL                     0.03288    0.08843   0.372 0.710030
## PosP                     -1.43324    0.30926  -4.634 3.58e-06 ***
## PosQB                     0.45099    0.14722   3.063 0.002189 **
## PosRB                     0.10485    0.10619   0.987 0.323436
## PosTE                     0.10453    0.12266   0.852 0.394084
## PosWR                     0.28800    0.09003   3.199 0.001379 **
## YrJr                      0.86972    0.08565  10.154  < 2e-16 ***
## YrSo                      0.54827    0.08930   6.139 8.28e-10 ***
## YrSr                      1.15762    0.08560  13.524  < 2e-16 ***
## team_nameAkron            2.31974    0.75920   3.056 0.002247 **
## team_nameAlabama          3.78481    0.72929   5.190 2.11e-07 ***
## team_nameArizona          2.08590    0.76538   2.725 0.006424 **
## team_nameArkansas         3.63646    0.73464   4.950 7.42e-07 ***
## team_nameAuburn           3.63350    0.73162   4.966 6.82e-07 ***
## team_nameBaylor           2.96476    0.74210   3.995 6.47e-05 ***
## team_nameBoston College   2.85545    0.74650   3.825 0.000131 ***
## team_nameBuffalo          1.92011    0.79144   2.426 0.015262 *
## team_nameCalifornia       3.42573    0.73834   4.640 3.49e-06 ***
## team_nameCincinnati       2.61657    0.75176   3.481 0.000500 ***
## team_nameClemson          3.60158    0.72993   4.934 8.05e-07 ***
## team_nameColorado         3.10946    0.74194   4.191 2.78e-05 ***
## team_nameDuke             2.96658    0.74056   4.006 6.18e-05 ***
## team_nameEast Carolina    2.34990    0.75674   3.105 0.001901 **
## team_nameFlorida          4.04445    0.73124   5.531 3.18e-08 ***
## team_nameGeorgia          3.46838    0.73549   4.716 2.41e-06 ***
## team_nameGeorgia Tech     3.01215    0.74322   4.053 5.06e-05 ***
## team_nameHawaii           1.82877    0.79164   2.310 0.020882 *
## team_nameHouston          3.11457    0.74224   4.196 2.71e-05 ***
## team_nameIdaho            2.02901    0.78428   2.587 0.009679 **
## team_nameIllinois         2.84328    0.74747   3.804 0.000142 ***
## team_nameIndiana          2.58050    0.75301   3.427 0.000610 ***
```

```
## team_nameIowa            3.46124    0.73685   4.697 2.64e-06 ***
## team_nameKansas          1.80044    0.79972   2.251 0.024364 *
## team_nameKentucky        2.48527    0.76272   3.258 0.001120 **
## team_nameLouisiana Tech  2.55052    0.74574   3.420 0.000626 ***
## team_nameLouisville      3.09955    0.74114   4.182 2.89e-05 ***
## team_nameLSU             4.47979    0.73006   6.136 8.45e-10 ***
## team_nameMarshall        1.54225    0.79137   1.949 0.051316 .
## team_nameMaryland        2.97538    0.74539   3.992 6.56e-05 ***
## team_nameMassachusetts   2.20855    0.77818   2.838 0.004539 **
## team_nameMemphis         2.81807    0.74608   3.777 0.000159 ***
## team_nameMichigan        4.41996    0.72774   6.074 1.25e-09 ***
## team_nameMinnesota       3.10701    0.74073   4.194 2.73e-05 ***
## team_nameMissouri        3.30501    0.73634   4.488 7.17e-06 ***
## team_nameNavy           -0.12581    1.00655  -0.125 0.900532
## team_nameNebraska        3.60285    0.73480   4.903 9.43e-07 ***
## team_nameNevada          1.93574    0.79177   2.445 0.014492 *
## team_nameNew Mexico      0.51724    0.92045   0.562 0.574152
## team_nameNorth Carolina  3.13228    0.73807   4.244 2.20e-05 ***
## team_nameNorth Texas     0.68425    0.92076   0.743 0.457401
## team_nameNorthwestern    2.51504    0.75508   3.331 0.000866 ***
## team_nameNotre Dame      3.42129    0.73762   4.638 3.51e-06 ***
## team_nameOklahoma        3.27933    0.73685   4.450 8.57e-06 ***
## team_nameOld Dominion    2.20088    0.77872   2.826 0.004709 **
## team_nameOregon          3.32004    0.73652   4.508 6.55e-06 ***
## team_namePittsburgh      3.96059    0.73358   5.399 6.70e-08 ***
## team_namePurdue          3.05744    0.74920   4.081 4.49e-05 ***
## team_nameRice            1.26546    0.82520   1.534 0.125146
## team_nameRutgers         3.17376    0.74388   4.266 1.99e-05 ***
## team_nameSouth Alabama   1.52515    0.79949   1.908 0.056435 .
## team_nameSouth Carolina  3.05335    0.74221   4.114 3.89e-05 ***
## team_nameStanford        3.12541    0.73333   4.262 2.03e-05 ***
## team_nameSyracuse        2.18371    0.77335   2.824 0.004747 **
## team_nameTemple          3.10871    0.74075   4.197 2.71e-05 ***
## team_nameTennessee       3.13435    0.74246   4.222 2.43e-05 ***
## team_nameTexas           3.47645    0.73689   4.718 2.38e-06 ***
## team_nameTexas A&M       3.64884    0.73394   4.972 6.64e-07 ***
## team_nameTexas Tech      2.46371    0.75720   3.254 0.001139 **
## team_nameToledo          2.68922    0.75362   3.568 0.000359 ***
## team_nameTroy            1.31625    0.80986   1.625 0.104105
## team_nameTulane          2.62835    0.75750   3.470 0.000521 ***
## team_nameTulsa           0.99430    0.87438   1.137 0.255472
## team_nameUCLA            4.00168    0.73047   5.478 4.30e-08 ***
## team_nameUtah            3.91131    0.73177   5.345 9.04e-08 ***
## team_nameVanderbilt      3.14200    0.74270   4.231 2.33e-05 ***
## team_nameVirginia        3.18530    0.74332   4.285 1.83e-05 ***
## team_nameVirginia Tech   3.25751    0.73937   4.406 1.05e-05 ***
## team_nameWake Forest     2.99709    0.74886   4.002 6.28e-05 ***
## team_nameWashington      3.47799    0.73407   4.738 2.16e-06 ***
## team_nameWest Virginia   3.35401    0.74142   4.524 6.07e-06 ***
## team_nameWisconsin       3.21104    0.73673   4.359 1.31e-05 ***
## team_nameWyoming         2.94563    0.74684   3.944 8.01e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 12400  on 16600  degrees of freedom
## Residual deviance: 10172  on 16514  degrees of freedom
## AIC: 10346
##
## Number of Fisher Scoring iterations: 7
```

Build prediction of college_draftLog

Here I take the logistic regression model created above and predict the likelihood of a player being drafted into the NFL. When we look at the summary of the prediction we have a minimum chance of being drafted in to the NFL of 0% and a maximum of 68.9%.

```
predict_college_draftTrain <- predict(college_draftLog, newdata = college_draftTrain, type = "response")
kable(as.matrix(summary(predict_college_draftTrain))) %>%
  kable_styling(latex_options = "striped")
```

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

Average Predicted Probabilities

For all of the TRUE cases where the player was drafted there is a probability of 0.247 and all of the TRUE cases where the player didn't get drafted of 0.105. We can see we are predicting slightly higher the TRUE cases that are drafted.

```
kable(tapply(predict_college_draftTrain, college_draftTrain$was_drafted,  FUN=mean)) %>%
  kable_styling(latex_options = "striped")
```

	x
FALSE	0.1058062
TRUE	0.2477268

Confusion Matrix

The below table gives us the total cases where players were predicted as True Negative, False Positive, False Negative, and True Positive. It shows the model predicted 9885 players as not being drafted correctly. There are 4669 players which the model said they would not be drafted but were drafted. The model shows 456 players which were predicted to be drafted but were not and 1591 players predicted to be drafted which were drafted. This model is biased more towards predicting palyers to not be drafted into the NFL.

```
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, Specificity, and Accuracy

Sensitivity is the correct number of predicted players drafted divided by the total number of true positives and false negatives. Sensitivity measures the percentage of the actual players who were drafted correctly. The sensitivity of the model was .777, meaning 77.7% of actual positives were correctly identified. Specificity is the correct number of predicted players not drafted by the total number of true negatives and false positives. Specificity measures the percentage of the actual players who were not drafted correctly. The specificity of the model measured 67.9% of the predictions correctly of players who were not drafted. The accuracy of the model was .684, meaning the proportion was .684 of the predictions were correct.

```
sens <- 1591/2047
spec <- 9885/14554
acc <- (9887 + 1616)/(9887 + 4862 + 447 + 1616)
kable(data.frame(Sensitivity = sens, Specificity = spec, Accuracy = acc)) %>%
  kable_styling(latex_options = "striped")
```

Sensitivity	Specificity	Accuracy
0.777235	0.6791947	0.6842137

Create a 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

Here I take the logistic regression model created above and predict the likelihood of a player being drafted into the NFL. When we look at the summary of the prediction we have a minimum chance of being drafted in to the NFL of 0% and a maximum of 66.2%.

```
predict_Test <- predict(college_draftLog, newdata = college_draftTest, type = "response")
kable(as.matrix(summary(predict_Test))) %>%
  kable_styling(latex_options = "striped")
```

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

Average Predicted Probabilities

For all of the TRUE cases where the player was drafted there is a probability of 0.246 and all of the TRUE cases where the player didn't get drafted of 0.109. We can see we are predicting slightly higher the TRUE cases that are drafted.

```
kable(tapply(predict_Test, college_draftTest$was_drafted,  FUN=mean)) %>%
  kable_styling(latex_options = "striped")
```

	x
FALSE	0.1093715
TRUE	0.2466993

Confusion Matrix

The below table gives us the total cases where players were predicted as True Negative, False Positive, False Negative, and True Positive. It shows the model predicted 3242 players as not being drafted correctly. There are 1681 players which the model said they would not be drafted but were drafted. The model shows 133 players which were predicted to be drafted but were not and 549 players predicted to be drafted which were drafted. This model is biased more towards predicting palyers to not be drafted into the NFL.

```
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, Specificity, and Accuracy

Sensitivity is the correct number of predicted players drafted by by the total number of true positives and false negatives. Sensitivity measures the percentage of the actual players who were drafted correctly. The sensitivity of the model was .804, meaning 80.4% of actual positives were correctly identified. Specificity is the correct number of predicted players not drafted by the total number of true negatives and false positives. Specificity measures the percentage of the actual players who were not drafted correctly. The specificity of the model measured 65.8% of the predictions correctly of players who were not drafted. The accuracy of the model was .698, meaning the proportion was .698 of the predictions were correct.

```
sens2 <- 549/682
spec2 <- 3242/4923
acc2 <- (3479 + 544)/(3479 + 1593 + 144 + 544)
kable(data.frame(Sensitivity = sens2,Specificity = spec2,Accuracy = acc2)) %>%
  kable_styling(latex_options = "striped")
```

Sensitivity	Specificity	Accuracy
0.8049853	0.6585415	0.6984375

ROC Curve, AUC, Recall & Precision

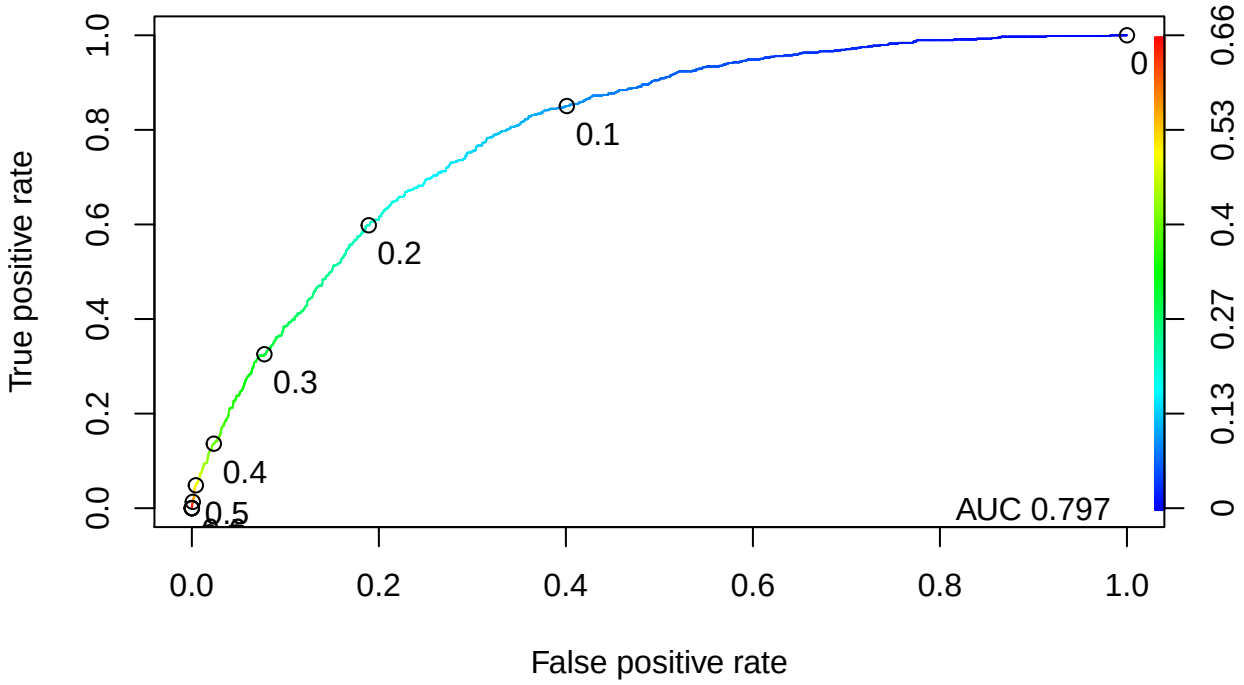
In the ROC Curve plot below it shows the true positive rate and the false positive rate. Based on the results I would like to choose a threshold value from 0.3 t0 0.2 as this threshold value range pulls the most towards the top left corner. A threshold value chosen in this range will provide the best tradeoff for the true positive and false positive rates. A random guess for area under the curve (AUC) is 0.5. So the model created needs to perform better than a random guess. The AUC for this model is 0.797 which is a lot better than a random guess but it's not perfect. If the decision threshold were at .2 then the recall would be .6 and the precision would be .75.

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

```
plot(ROCRperf_Test, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1), text.adj=c(-0.2,1.7))
text(.9,0,paste("AUC",round(auc(college_draftTest$was_drafted, predict_Test),3)))
```

```
## Setting levels: control = FALSE, case = TRUE
```

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



Logistic Regression Model 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_draftTra
```

Build the Logistic Regression Model

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

```
##
## 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 = "respons
kable(as.matrix(summary(predict_college_draftTrain2))) %>%
  kable_styling(latex_options = "striped")
```

Min.	0.0001989
1st Qu.	0.0189856
Median	0.0526550
Mean	0.1233207
3rd Qu.	0.1574638
Max.	0.9022158
NA's	2.0000000

Confusion Matrix

The below table gives us the total cases where players were predicted as True Negative, False Positive, False Negative, and True Positive. It shows the model predicted 11199 players as not being drafted correctly. There are 3353 players which the model said they would not be drafted but were drafted. The model shows 405 players which were predicted to be drafted but were not and 1642 players predicted to be drafted which were drafted. This model is biased more towards predicting palyers to not be drafted into the NFL.

```
train_fnfp2 <- table(college_draftTrain2$was_drafted, predict_college_draftTrain2 > mean(college_draftTr
kable(train_fnfp2) %>%
  kable_styling(latex_options = "striped")
```

	FALSE	TRUE
FALSE	11199	3353
TRUE	405	1642

Sensitivity, Specificity, Accuracy

Sensitivity is the correct number of predicted players drafted by by the total number of true positives and false negatives. Sensitivity measures the percentage of the actual players who were drafted correctly. The sensitivity of the model was .802, meaning 80.2% of actual positives were correctly identified. Specificity is the correct number of predicted players not drafted by the total number of true negatives and false positives. Specificity measures the percentage of the actual players who were not drafted correctly. The specificity of the model measured 76.9% of the predictions correctly of players who were not drafted. The accuracy of the model was .769, meaning the proportion was .769 of the predictions were correct.

```
sens3 <- 1642/2047
spec3 <- 11199/14552
acc3 <- (11291 + 1651)/(11291 + 3456 + 412 + 1651)
kable(data.frame(Sensitivity = sens3, Specificity = spec3, Accuracy = acc3)) %>%
  kable_styling(latex_options = "striped")
```

Sensitivity	Specificity	Accuracy
0.8021495	0.7695849	0.7698989

Average Predicted Probabilities

```
kable(tapply(predict_college_draftTrain2, college_draftTrain2$was_drafted,  FUN=mean, na.rm = TRUE)) %>%
  kable_styling(latex_options = "striped")
```

	x
FALSE	0.09420627
TRUE	0.33029327

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")
kable(as.matrix(summary(predict_cdTest2))) %>%
  kable_styling(latex_options = "striped")
```

Min.	0.0001989
1st Qu.	0.0215763
Median	0.0616054
Mean	0.1329174
3rd Qu.	0.1777606
Max.	0.9011971

Confusion Matrix

The below table gives us the total cases where players were predicted as True Negative, False Positive, False Negative, and True Positive. It shows the model predicted 3418 players as not being drafted correctly. There are 1120 players which the model said they would not be drafted but were drafted. The model shows 155 players which were predicted to be drafted but were not and 527 players predicted to be drafted which were drafted. This model is biased more towards predicting palyers to not be drafted into the NFL.

```
test_fnfp2 <- table(college_draftTest2$was_drafted, predict_cdTest2 > mean(college_draftTest2$was_draft
kable(test_fnfp2) %>%
  kable_styling(latex_options = "striped")
```

	FALSE	TRUE
FALSE	3418	1120
TRUE	155	527

Sensitivity, Specificity, and Accuracy

Sensitivity is the correct number of predicted players drafted by by the total number of true positives and false negatives. Sensitivity measures the percentage of the actual players who were drafted correctly. The sensitivity of the model was .772, meaning 77.2% of actual positives were correctly identified. Specificity is the correct number of predicted players not drafted by the total number of true negatives and false positives. Specificity measures the percentage of the actual players who were not drafted correctly. The specificity of the model measured 75.3% of the predictions correctly of players who were not drafted. The accuracy of the model was .769, meaning the proportion was .769 of the predictions were correct.

```
sens4 <- 527/682
spec4 <- 3418/4538
acc4 <- (3811 + 558)/(3811 + 1181 + 130 + 558)
kable(data.frame(Sensitivity = sens4, Specificity = spec4, Accuracy = acc4)) %>%
  kable_styling(latex_options = "striped")
```

Sensitivity	Specificity	Accuracy
0.7727273	0.7531952	0.7691901

Average Predicted Probabilities on Test

For all of the TRUE cases where the player was drafted there is a probability of 0.32 and all of the TRUE cases where the player didn't get drafted of 0.1. We can see we are predicting slightly higher the TRUE cases that are drafted.

```
tapply(predict_cdTest2, college_draftTest2$was_drafted, FUN=mean) %>% data.frame(was_drafted = .) %>% ka
```

	was_drafted
FALSE	0.1034471
TRUE	0.3290114

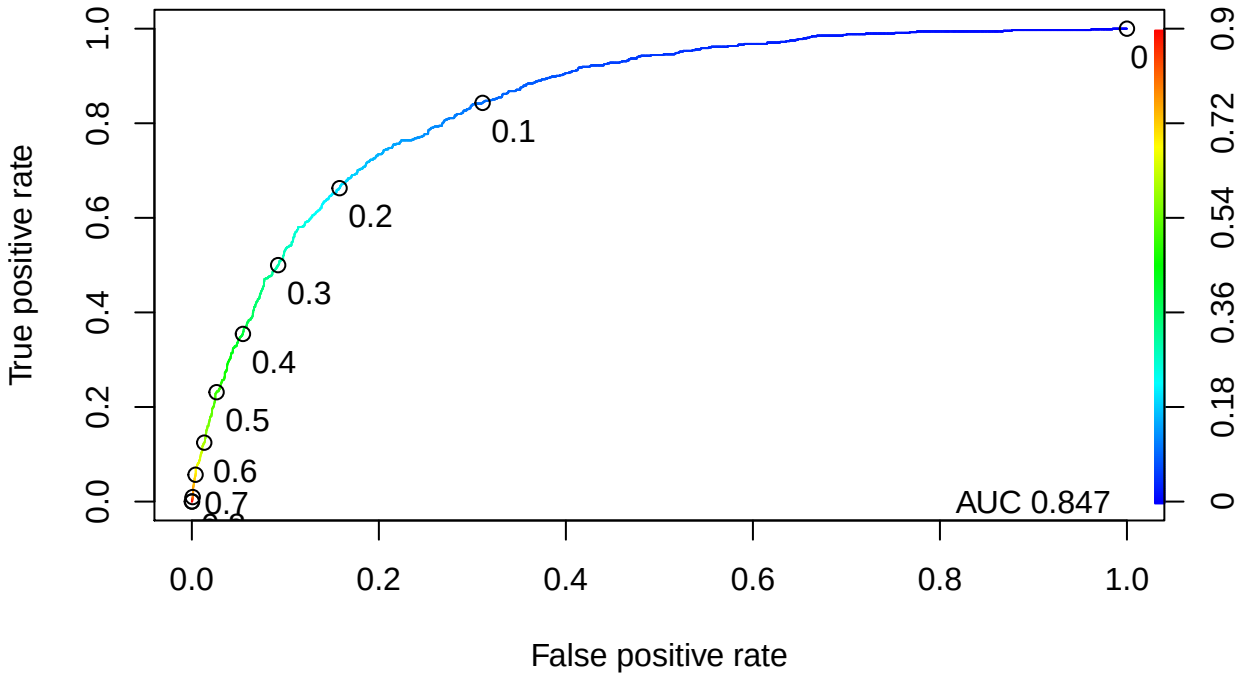
ROC Curve, AUC, and Recall and Precision

In the ROC Curve plot below it shows the true positive rate and the false positive rate. Based on the results I would like to choose a threshold value from 0.4 to 0.2 as this threshold value range pulls the most towards the top left corner. A threshold value chosen in this range will provide the best tradeoff for the true positive and false positive rates. A random guess for area under the curve (AUC) is 0.5. So the model created needs to perform better than a random guess. Let's see if we can improve upon the first model which had an AUC of 0.797. The AUC for this model is 0.846 which is better than our previous model and a lot better than a random guess. If the decision threshold were at .3 then the recall would be .5 and the precision would be 5/6.

```
ROCRpred_Test2 <- ROCR::prediction(predict_cdTest2, college_draftTest2$was_drafted)
ROCRperf_Test2 <- performance(ROCRpred_Test2, "tpr", "fpr")
plot(ROCRperf_Test2, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1), text.adj=c(-0.2,1.7))
text(.9,0,paste("AUC",round(auc(college_draftTest2$was_drafted, predict_cdTest2),3)))
```

```
## Setting levels: control = FALSE, case = TRUE
```

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



ROC plots stacked

From the plots below we can see the second logistic regression ROC plot is far superior as the curve pulls to the top left even more. The AUC for the second plot is 0.847 compared to the first is 0.797, based on these models the second will be more accurate no matter what threshold you use.

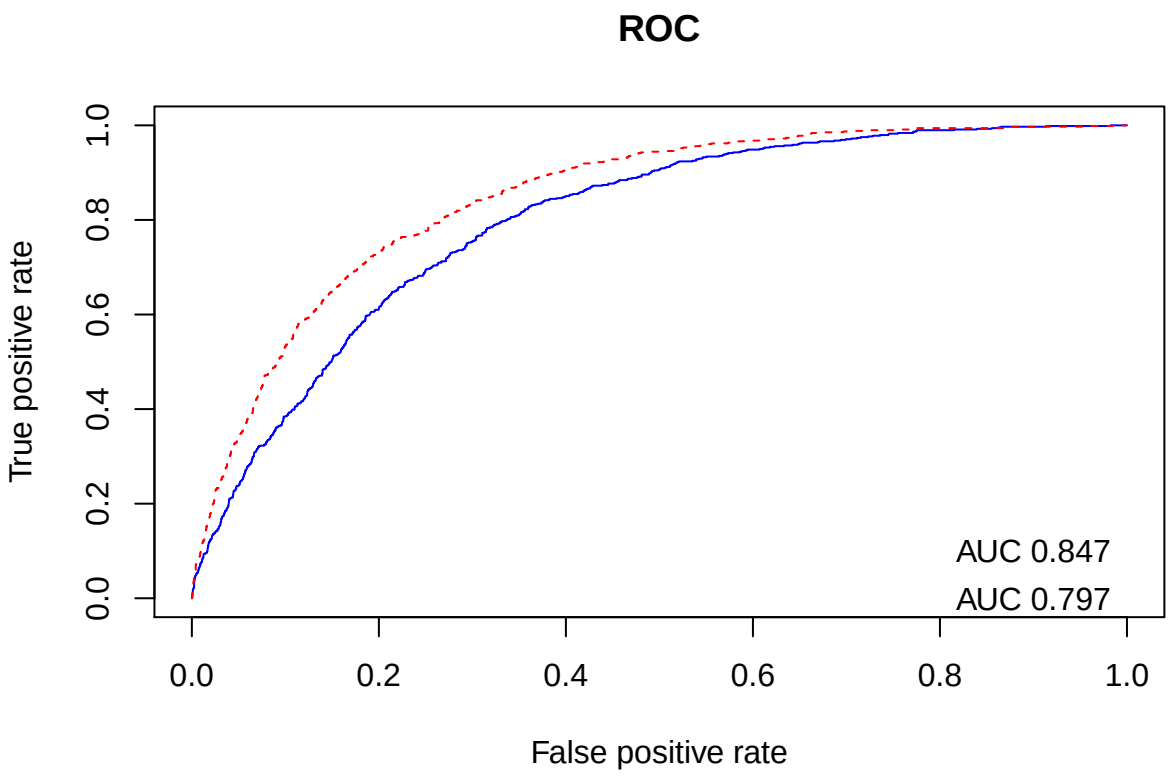
```
plot(ROCRperf_Test, col = 4, lty = 1, main = "ROC")
plot(ROCRperf_Test2, col = 2, lty = 2, add = TRUE)
text(0.9,0,paste("AUC",round(auc(college_draftTest$was_drafted, predict_Test),3)))
```

```
## Setting levels: control = FALSE, case = TRUE
```

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

```
text(0.9,0.1,paste("AUC",round(auc(college_draftTest2$was_drafted, predict_cdTest2),3)))
```

```
## Setting levels: control = FALSE, case = TRUE
## Setting direction: controls < cases
```

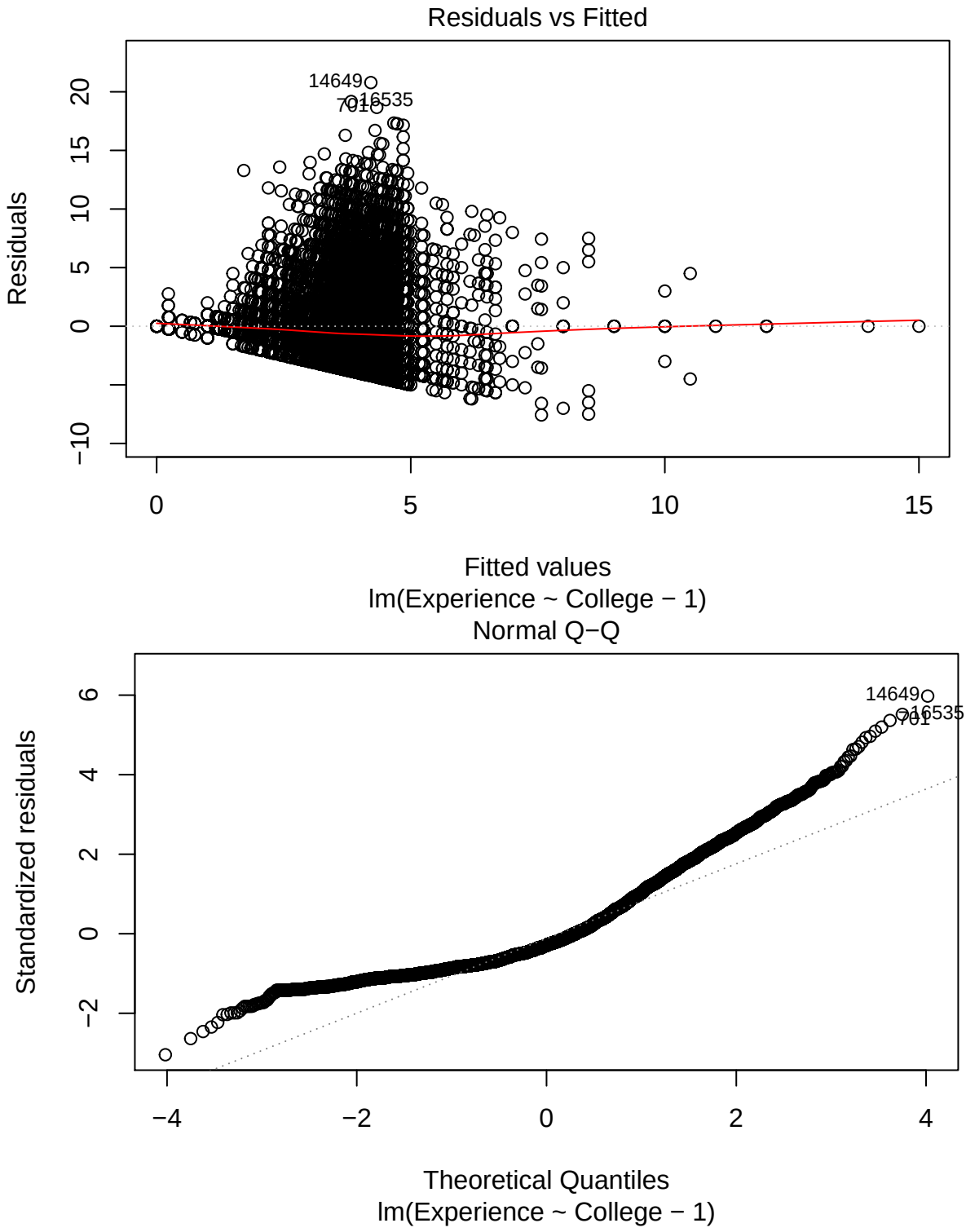


Appendix

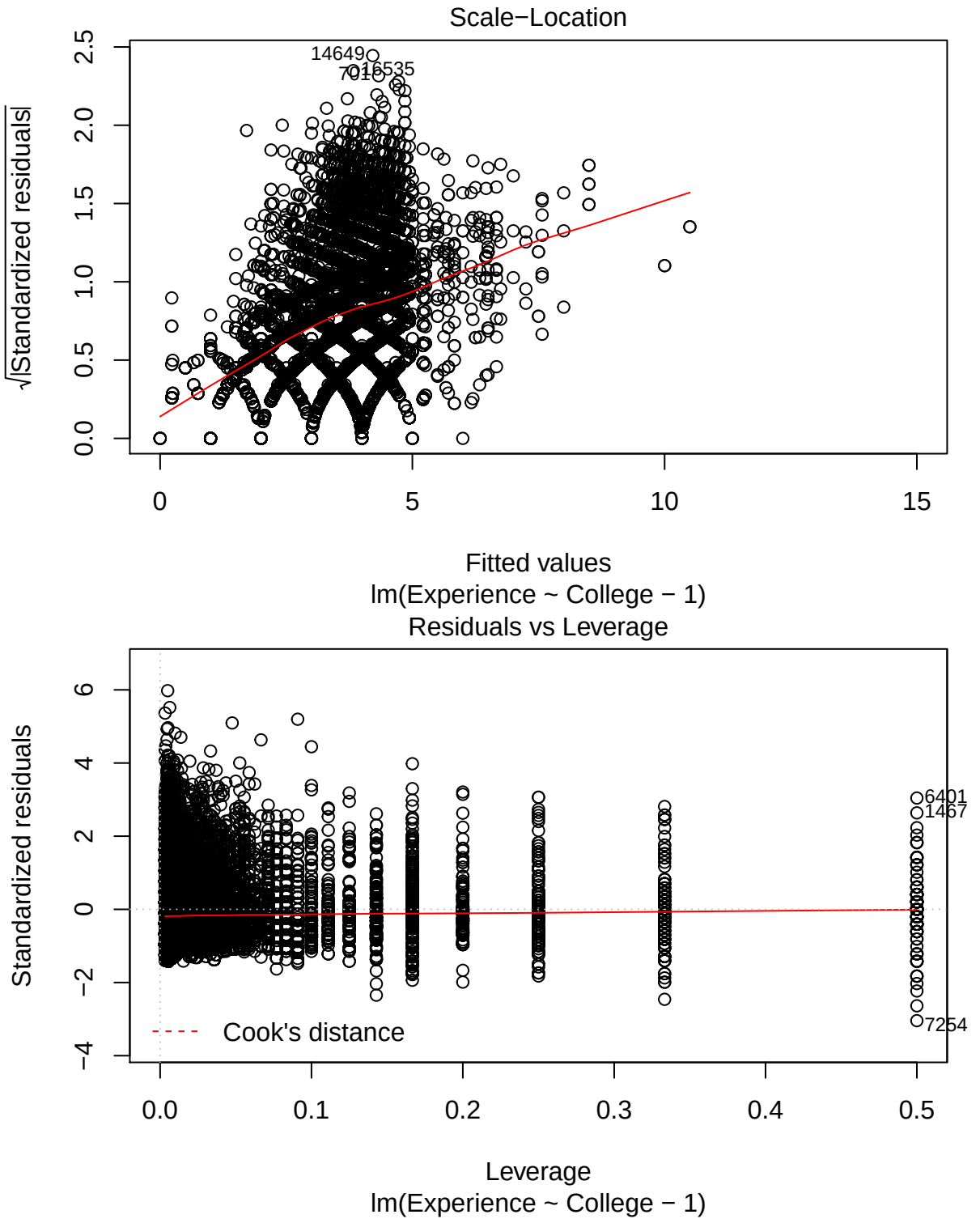
Plot of nfl_model

```
plot(nfl_model)
```

```
## Warning: not plotting observations with leverage one:
##   235, 291, 468, 545, 607, 907, 1060, 1190, 1294, 1297, 1305, 1339, 1392, 1595, 1598, 1608, 1734, 175
```



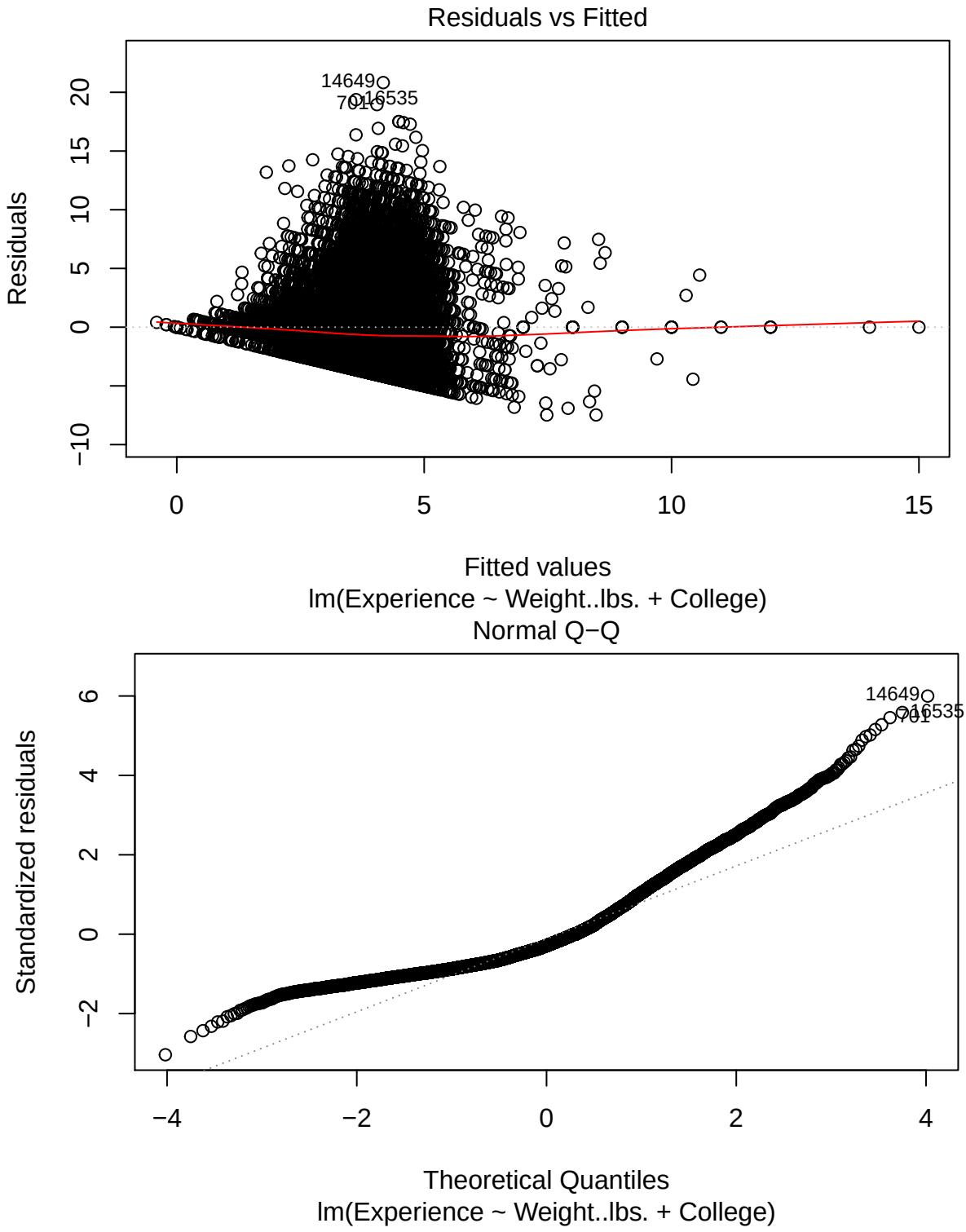

```
## Warning: not plotting observations with leverage one:
##   235, 291, 468, 545, 607, 907, 1060, 1190, 1294, 1297, 1305, 1339, 1392, 1595, 1598, 1608, 1734, 175
```



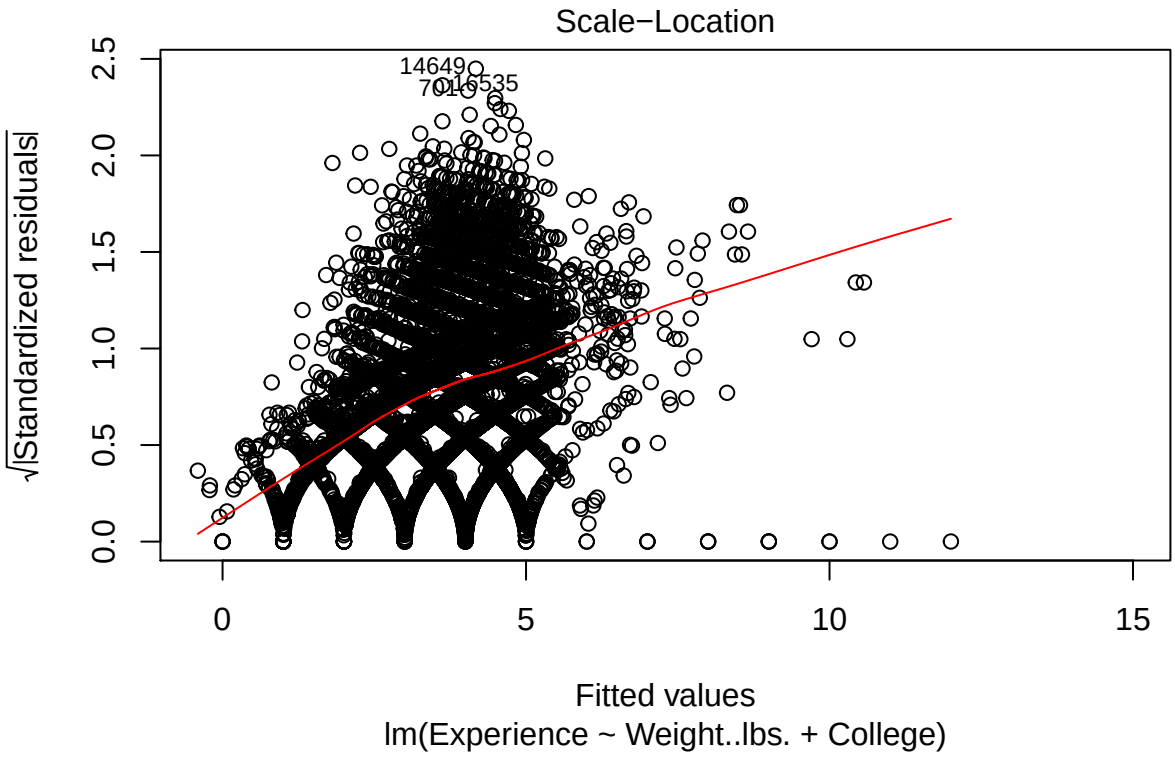
Plot of nfl_model2

```
plot(nfl_model2)
```

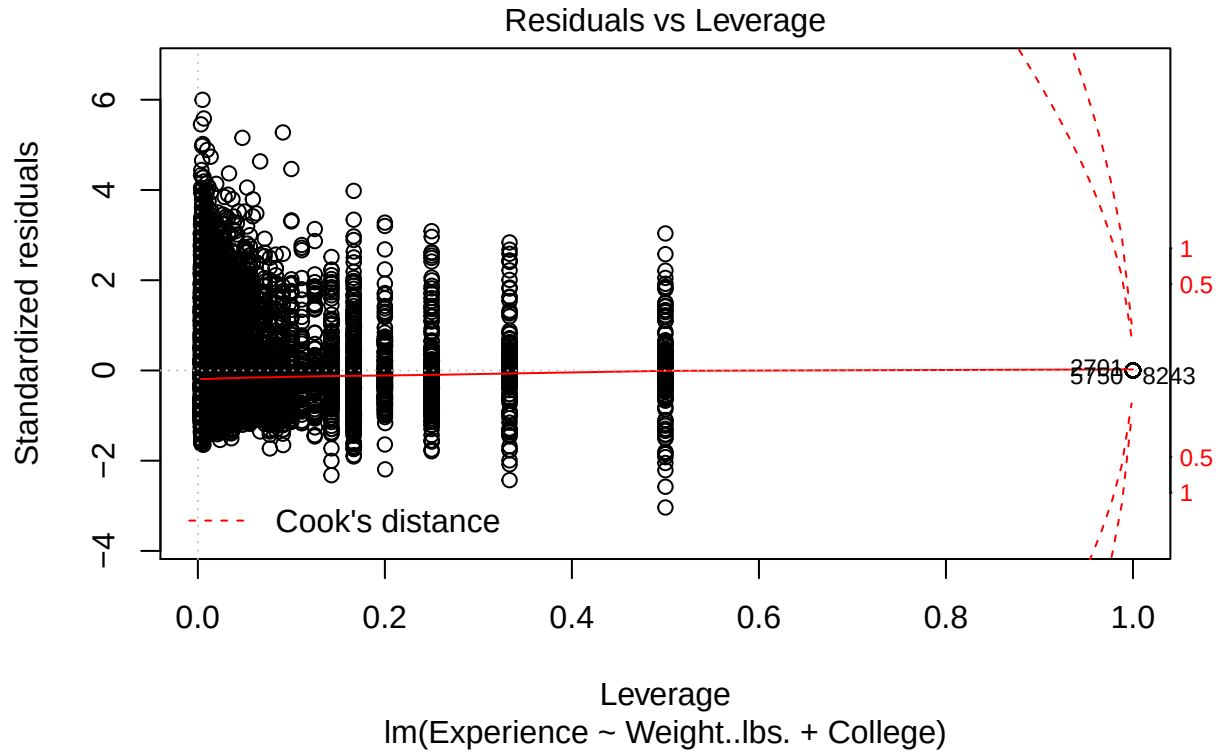
```
## Warning: not plotting observations with leverage one:
##   234, 1290, 1589, 1592, 1602, 1750, 1866, 1913, 2110, 2524, 2701, 2743, 2791, 2793, 2992, 3040, 313
```




```
## Warning: not plotting observations with leverage one:
##   234, 1290, 1589, 1592, 1602, 1750, 1866, 1913, 2110, 2524, 2701, 2743, 2791, 2793, 2992, 3040, 313
```



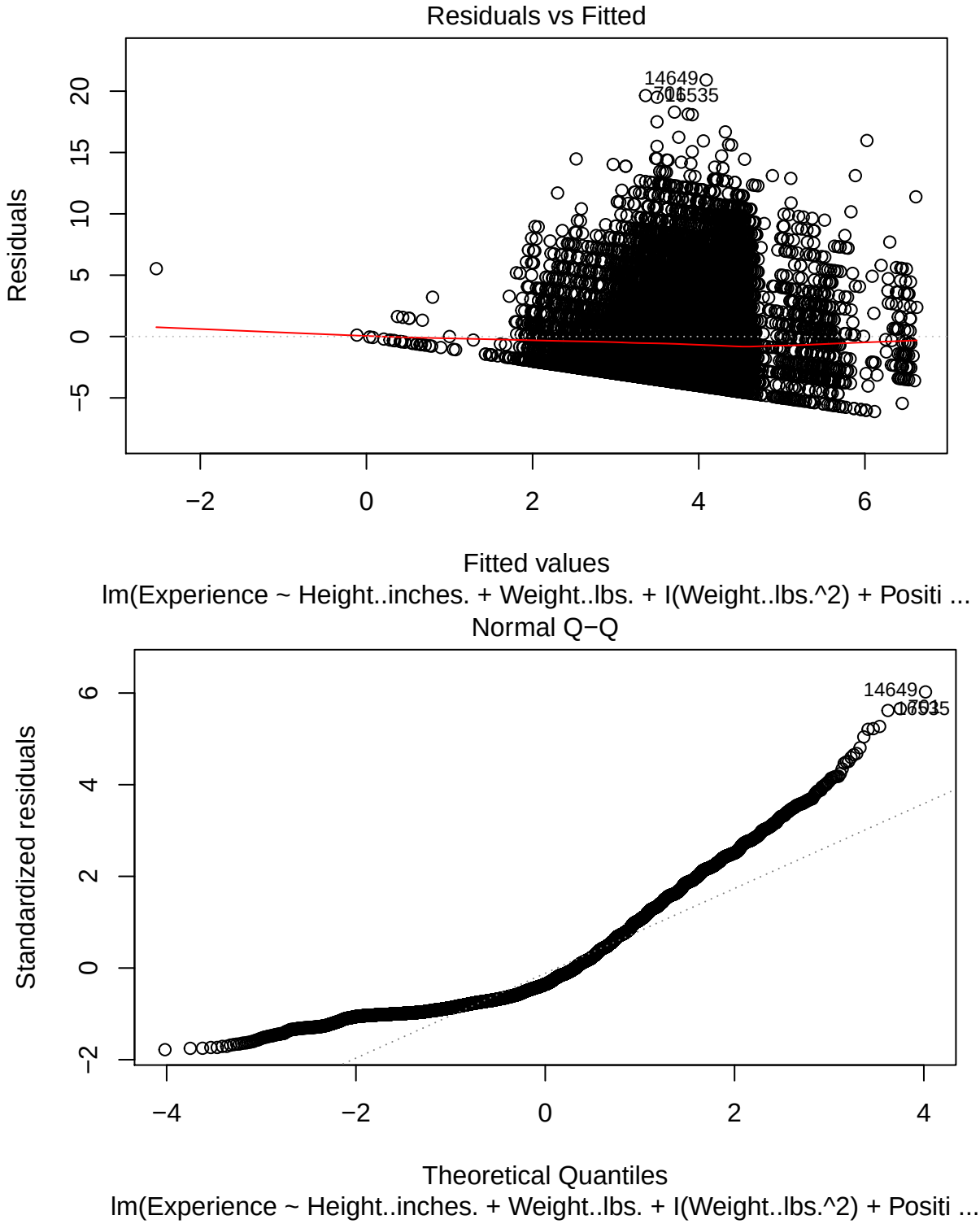

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```



Plot of nfl_model3

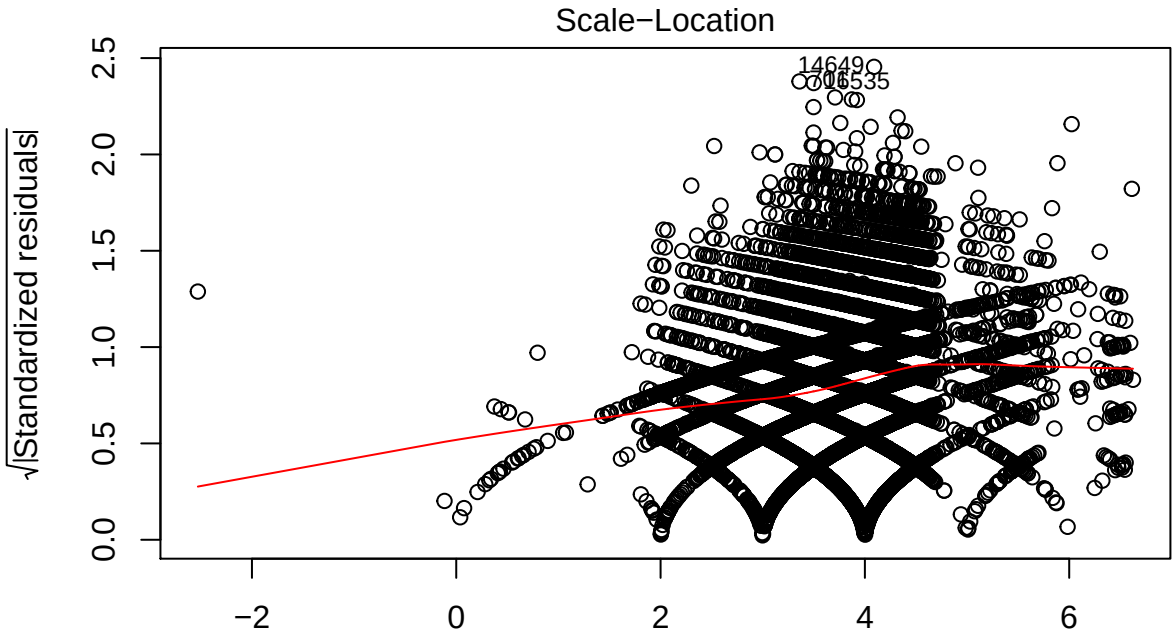
```
plot(nfl_model3)
```

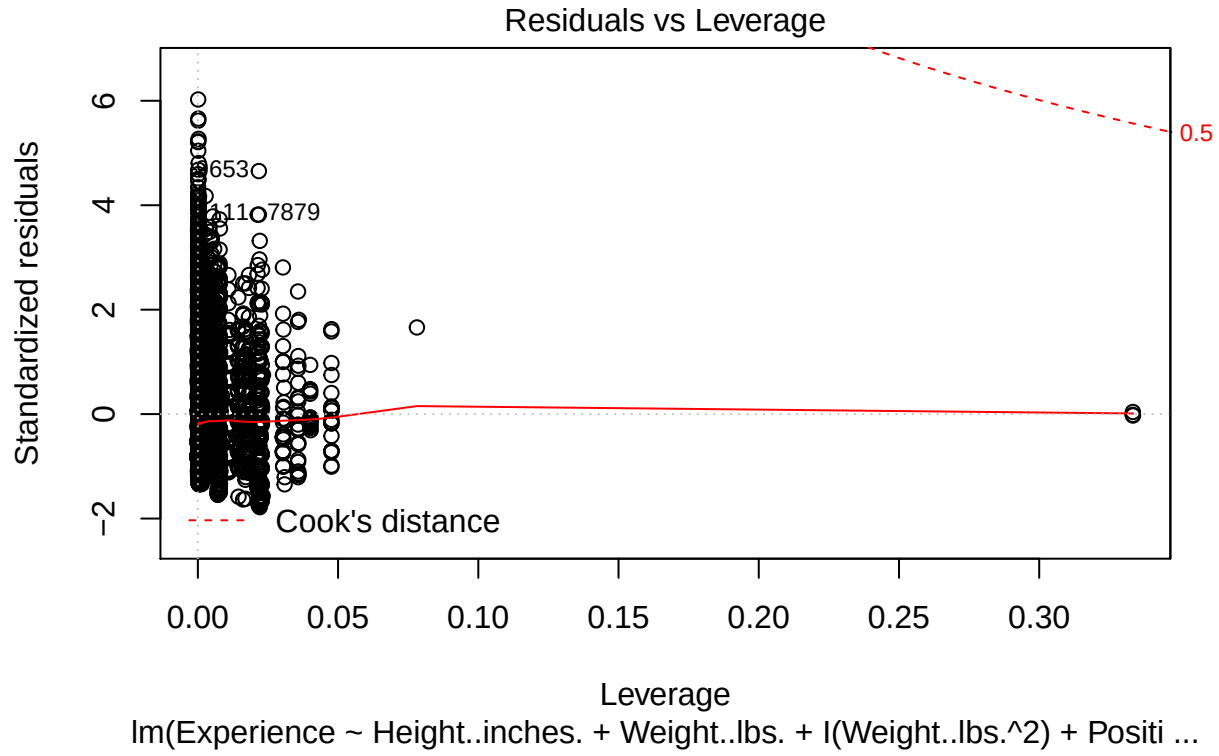
```
## Warning: not plotting observations with leverage one:
##   7821
```




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

```
##     7821
```


Scale−Location
√|Standardized residuals|
14649
16535
0.0
0.5
1.0
1.5
2.0
2.5
−2
0
2
4
6
Fitted values
lm(Experience ~ Height..inches. + Weight..lbs. + I(Weight..lbs.^2) + Positi ...


Residuals vs Leverage
Standardized residuals
653
11
7879
0.5
Cook's distance
−2
0
2
4
6
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Leverage
lm(Experience ~ Height..inches. + Weight..lbs. + I(Weight..lbs.^2) + Positi ...