Ravens Punt Project

Logan Donaldson

4/6/2021

```
library(faraway)
```

```
## Warning: package 'faraway' was built under R version 4.0.4
```

```
library(readxl)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
puntOrig <- read_excel("C:\\Users\\tropi\\Downloads\\Ravens Project\\punt_appended.xlsx", na = "NA")
```

```
## New names:
## * `` -> ...1
## * field_position -> field_position...19
## * `` -> ...43
## * field_position -> field_position...63
```

```
puntOrig <- subset(puntOrig, !is.na(return_yards))
puntOrig$penalty_yards[is.na(puntOrig$penalty_yards)] <- 0
puntOrig$punt_rush_count[is.na(puntOrig$punt_rush_count)] <- 0
puntNoPenalty <- subset(puntOrig, penalty_yards == 0)
puntPenalty <- subset(puntOrig, penalty_yards != 0)
nrow(puntOrig)
```

```
## [1] 2991
```

In the above code we load the punt_appended data set which is a subset of the total data set which contains only punts. A few extra fields were also added to aid in our analysis. After loading we remove all punts which did not have an attempted return. We also assign a value of 0 for the penalty_yards field for plays which did not have a penalty. Likewise we assign a value of 0 to the punt_rush_count field for plays which did not have any punt rushers. Lastly, we bucket the punt returns into plays with a penalty and plays without.

```
lmod<-lm(return_yards ~ clock_truncated + hash_R + hash_C + clean_catch_binary + actual_kick_direction_

summary(lmod)
```

```
##
## Call:
## lm(formula = return_yards ~ clock_truncated + hash_R + hash_C +
##     clean_catch_binary + actual_kick_direction_R + actual_kick_direction_C +
##     kick_type_NORMAL + return_direction_R + return_direction_C +
##     hang_time + quarter + field_position_pos + off_score + def_score +
##     score_differential + garbage_time + kick_depth + kick_width_neg +
##     kick_yards + punt_rush_count + fumble + gunners_count + vise_count,
##     data = puntNoPenalty)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.854  -5.465  -1.549   2.523  89.697
##
## Coefficients: (1 not defined because of singularities)
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.109683   4.247867   0.261  0.79394
## clock_truncated          -0.036376   0.050888  -0.715  0.47479
## hash_R                    0.389511   0.476228   0.818  0.41349
## hash_C                    0.927515   0.611102   1.518  0.12920
## clean_catch_binary        4.891939   0.759788   6.439 1.45e-10 ***
## actual_kick_direction_R   0.824149   1.352257   0.609  0.54228
## actual_kick_direction_C   2.085355   0.801035   2.603  0.00929 **
## kick_type_NORMAL          0.399397   0.693589   0.576  0.56478
## return_direction_R        3.950965   0.608901   6.489 1.05e-10 ***
## return_direction_C       -0.369221   0.519004  -0.711  0.47690
## hang_time                -3.496296   0.537640  -6.503 9.55e-11 ***
## quarter                  -0.025537   0.320199  -0.080  0.93644
## field_position_pos       -0.022931   0.022118  -1.037  0.29994
## off_score                -0.031157   0.032303  -0.965  0.33489
## def_score                 0.023674   0.029455   0.804  0.42162
## score_differential              NA         NA      NA       NA
## garbage_time             -1.500887   1.489713  -1.008  0.31380
## kick_depth                0.211037   0.123305   1.711  0.08712 .
## kick_width_neg            0.056546   0.034080   1.659  0.09721 .
## kick_yards                0.142928   0.119553   1.196  0.23200
## punt_rush_count           0.002162   0.124346   0.017  0.98613
## fumble                   -2.185144   1.563068  -1.398  0.16225
## gunners_count            -1.211119   1.672123  -0.724  0.46895
## vise_count                0.707266   0.299637   2.360  0.01833 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.33 on 2394 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.1307, Adjusted R-squared:  0.1227
## F-statistic: 16.36 on 22 and 2394 DF,  p-value: < 2.2e-16
```

```
finalLmod<-lm(return_yards ~ clean_catch_binary + actual_kick_direction_R + actual_kick_direction_C + r
summary(finalLmod)
```

```
##
## Call:
## lm(formula = return_yards ~ clean_catch_binary + actual_kick_direction_R +
##     actual_kick_direction_C + return_direction_R + return_direction_C +
##     kick_width_neg + hang_time + kick_depth + vise_count, data = puntNoPenalty)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.678  -5.405  -1.672   2.542  89.919
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -1.27761    2.22636  -0.574  0.56612
## clean_catch_binary        4.33418    0.62712   6.911 6.13e-12 ***
```

```
## actual_kick_direction_R  0.67738    1.34876   0.502  0.61556
## actual_kick_direction_C  2.11702    0.79966   2.647  0.00816 **
## return_direction_R       4.07106    0.60546   6.724 2.20e-11 ***
## return_direction_C      -0.35343    0.51616  -0.685  0.49358
## kick_width_neg           0.06865    0.03343   2.054  0.04012 *
## hang_time               -3.71894    0.52756  -7.049 2.34e-12 ***
## kick_depth               0.37242    0.03309  11.256  < 2e-16 ***
## vise_count               0.75751    0.27676   2.737  0.00625 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.33 on 2407 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.1262, Adjusted R-squared:  0.1229
## F-statistic: 38.61 on 9 and 2407 DF,  p-value: < 2.2e-16
```

```
anova(finalLmod, lmod)
```

```
## Analysis of Variance Table
##
## Model 1: return_yards ~ clean_catch_binary + actual_kick_direction_R +
##     actual_kick_direction_C + return_direction_R + return_direction_C +
##     kick_width_neg + hang_time + kick_depth + vise_count
## Model 2: return_yards ~ clock_truncated + hash_R + hash_C + clean_catch_binary +
##     actual_kick_direction_R + actual_kick_direction_C + kick_type_NORMAL +
##     return_direction_R + return_direction_C + hang_time + quarter +
##     field_position_pos + off_score + def_score + score_differential +
##     garbage_time + kick_depth + kick_width_neg + kick_yards +
##     punt_rush_count + fumble + gunners_count + vise_count
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1   2407 256686
## 2   2394 255347 13    1339.2 0.9658  0.483
```

```
lmod <- finalLmod
```

Above we perform a linear regression analysis on the punt returns without a penalty. Return_yards is the response variable. Through this regression we hope to better understand not only what factors contribute to long punt returns but the degree to which they can be used to accurately predict the outcome of a punt return. We start by regressing on the majority of variables in the data set and then remove the non-significant predictors. There removal is justified by the large p-value in the ANOVA test.

The near zero p-value associated with the regression suggests that there is indeed a relationship between the predictors and return_yards. However, a Multiple R-squared value of 0.1262 implies that there remains a large amount of variance in the data that is unaccounted for by the regression. In other words, the predictors certainly tell part of the story but not all of it.

Note that kick_width_neg is a transformation of the kick_width data field which changes the data to a (-26,26) scale to make it more usable for regression.

```
lmods<-lm(return_yards ~ clock_truncated + hash_R + hash_C + clean_catch_binary + actual_kick_direction_
summary(lmods)
```

```
##
## Call:
## lm(formula = return_yards ~ clock_truncated + hash_R + hash_C +
##     clean_catch_binary + actual_kick_direction_R + actual_kick_direction_C +
##     kick_type_NORMAL + return_direction_R + return_direction_C +
```

```
##     hang_time + quarter + field_position_pos + off_score + def_score +
##     score_differential + garbage_time + kick_depth + kick_width_neg +
##     kick_yards + punt_rush_count + fumble + gunners_count + vise_count,
##     data = puntPenalty)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.866  -4.240  -1.398   3.049  67.698
##
## Coefficients: (2 not defined because of singularities)
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.59474    3.95456   0.403  0.68691
## clock_truncated          -0.08717    0.07218  -1.208  0.22771
## hash_R                    1.53312    0.64742   2.368  0.01823 *
## hash_C                    0.12502    0.89524   0.140  0.88899
## clean_catch_binary        2.18171    1.25022   1.745  0.08153 .
## actual_kick_direction_R   0.38251    1.94392   0.197  0.84408
## actual_kick_direction_C   1.48681    1.14997   1.293  0.19658
## kick_type_NORMAL          0.24390    1.02937   0.237  0.81279
## return_direction_R        1.09159    0.78162   1.397  0.16311
## return_direction_C       -0.72147    0.72615  -0.994  0.32088
## hang_time                -2.60383    0.79801  -3.263  0.00117 **
## quarter                  -0.03179    0.43225  -0.074  0.94140
## field_position_pos        0.01104    0.02871   0.385  0.70073
## off_score                 0.04854    0.04302   1.128  0.25961
## def_score                -0.03040    0.04180  -0.727  0.46738
## score_differential             NA         NA      NA       NA
## garbage_time              2.36961    2.43517   0.973  0.33094
## kick_depth                0.30558    0.18271   1.672  0.09500 .
## kick_width_neg           -0.02751    0.04943  -0.557  0.57799
## kick_yards               -0.10555    0.17616  -0.599  0.54933
## punt_rush_count           0.09803    0.18614   0.527  0.59864
## fumble                   -0.69402    2.47278  -0.281  0.77907
## gunners_count                  NA         NA      NA       NA
## vise_count                0.27410    0.41469   0.661  0.50891
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.783 on 549 degrees of freedom
## Multiple R-squared:  0.08956,    Adjusted R-squared:  0.05473
## F-statistic: 2.572 on 21 and 549 DF,  p-value: 0.000168
```

```
finalLmod<-lm(return_yards ~ hash_R + hash_C + clean_catch_binary + hang_time + kick_depth, puntPenalty)
summary(finalLmod)
```

```
##
## Call:
## lm(formula = return_yards ~ hash_R + hash_C + clean_catch_binary +
##     hang_time + kick_depth, data = puntPenalty)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.027  -4.316  -1.389   2.923  69.050
##
## Coefficients:
```

```
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         4.33482    2.96869   1.460 0.144796
## hash_R              1.31487    0.61793   2.128 0.033782 *
## hash_C              0.22682    0.87427   0.259 0.795391
## clean_catch_binary  3.21032    0.99267   3.234 0.001292 **
## hang_time          -2.77053    0.78358  -3.536 0.000440 ***
## kick_depth          0.18025    0.04836   3.727 0.000213 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.795 on 565 degrees of freedom
## Multiple R-squared:  0.05969,    Adjusted R-squared:  0.05137
## F-statistic: 7.173 on 5 and 565 DF,  p-value: 1.601e-06
```

```
anova(finalLmod,lmods)
```

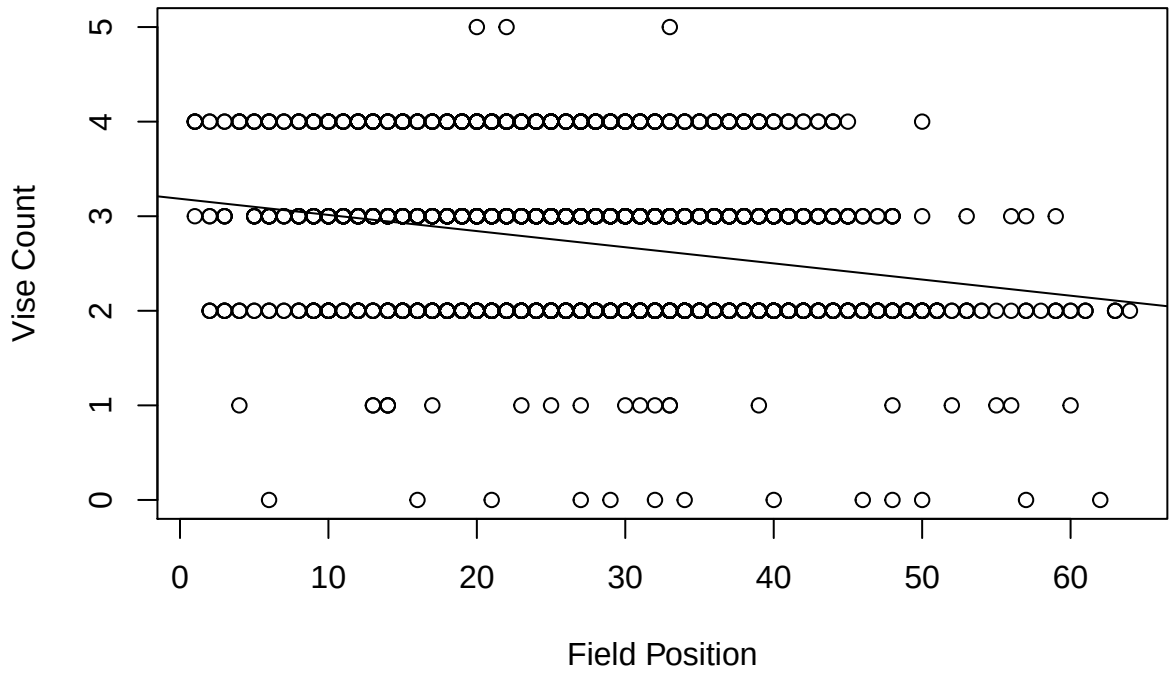
```
## Analysis of Variance Table
##
## Model 1: return_yards ~ hash_R + hash_C + clean_catch_binary + hang_time +
##     kick_depth
## Model 2: return_yards ~ clock_truncated + hash_R + hash_C + clean_catch_binary +
##     actual_kick_direction_R + actual_kick_direction_C + kick_type_NORMAL +
##     return_direction_R + return_direction_C + hang_time + quarter +
##     field_position_pos + off_score + def_score + score_differential +
##     garbage_time + kick_depth + kick_width_neg + kick_yards +
##     punt_rush_count + fumble + gunners_count + vise_count
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1    565 26091
## 2    549 25262 16    828.71 1.1256 0.3271
```

The process as described previously is applied to the punt returns which did have penalties. Interestingly, the significant predictors differ.

```
lmodVise <- lm(vise_count ~ field_position_pos, puntNoPenalty)
summary(lmodVise)
```

```
##
## Call:
## lm(formula = vise_count ~ field_position_pos, data = puntNoPenalty)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.08176 -0.63783  0.08898  0.41339  2.37924
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3.184210   0.041014   77.64   <2e-16 ***
## field_position_pos -0.017074   0.001381  -12.36   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.743 on 2418 degrees of freedom
## Multiple R-squared:  0.05945,    Adjusted R-squared:  0.05906
## F-statistic: 152.8 on 1 and 2418 DF,  p-value: < 2.2e-16
```

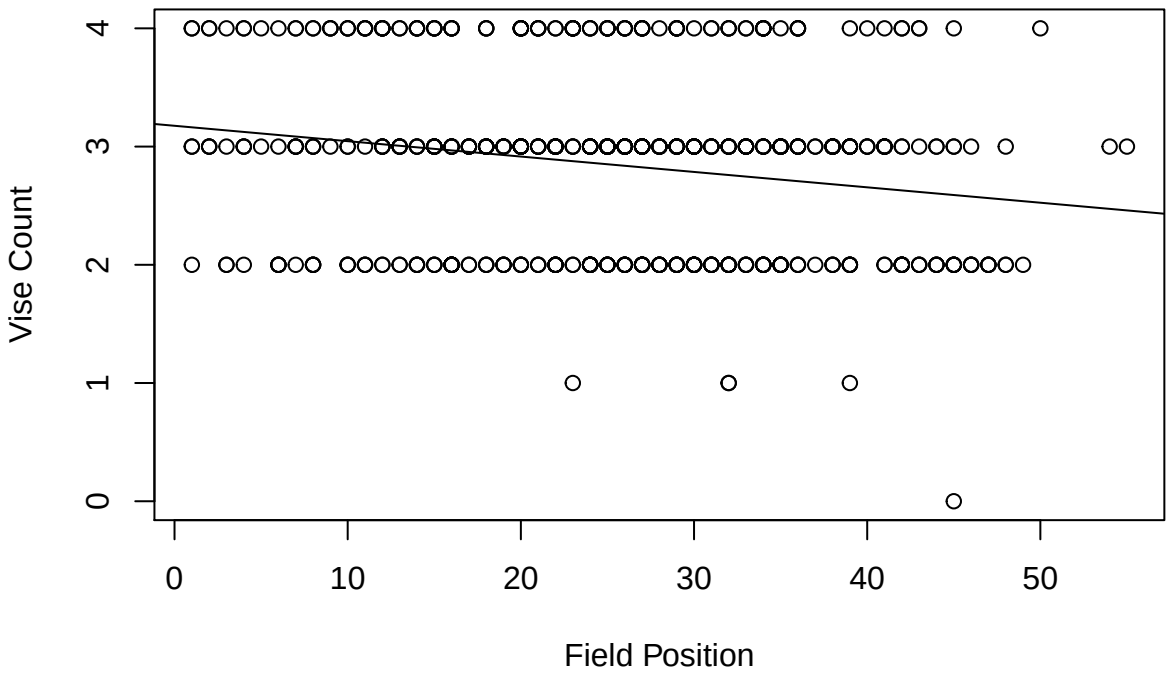
```r
plot(puntNoPenalty$field_position_pos, puntNoPenalty$vise_count, xlab = "Field Position", ylab = "Vise C
abline(lmodVise)
```




```r
lmodVise <- lm(vise_count ~ field_position_pos, puntPenalty)
summary(lmodVise)
```

```
##
## Call:
## lm(formula = vise_count ~ field_position_pos, data = puntPenalty)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5903 -0.7334  0.1104  0.3186  1.4748
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3.175898   0.077966  40.734  < 2e-16 ***
## field_position_pos -0.013013   0.002796  -4.654 4.06e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7335 on 569 degrees of freedom
## Multiple R-squared:  0.03666,    Adjusted R-squared:  0.03497
## F-statistic: 21.66 on 1 and 569 DF,  p-value: 4.062e-06
```

```r
plot(puntPenalty$field_position_pos, puntPenalty$vise_count, xlab = "Field Position", ylab = "Vise Coun
abline(lmodVise)
```

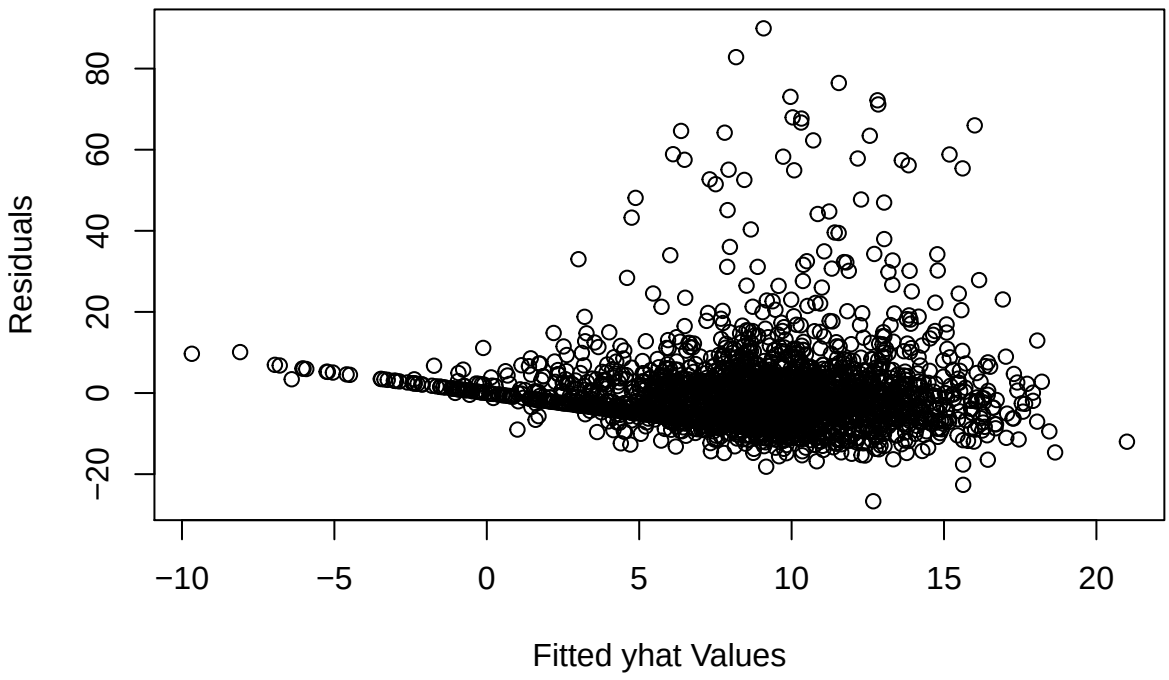


Having vise_count as a predictor in our model could be considered cumbersome and unintuitive. Thus, in the above code we check to see if vise_count is strongly correlated with field_position_pos and whether this correlation means that we can drop vise_count from the regression. We see that there is indeed a correlation between the two variables. However, in code not shown here, it was revealed that dropping vise_count from the regression would nonetheless significantly reduce the model's effectiveness at predicting return_yards.

We now turn to checking the validity of the assumptions inherent to all linear regression. Namely, that the residuals have constant variance, are normally distributed, and the observations are not correlated.

For the remainder of the analysis we focus on the punt returns without penalties.

```
plot(fitted(lmod), residuals(lmod), xlab = "Fitted yhat Values", ylab = "Residuals")
```

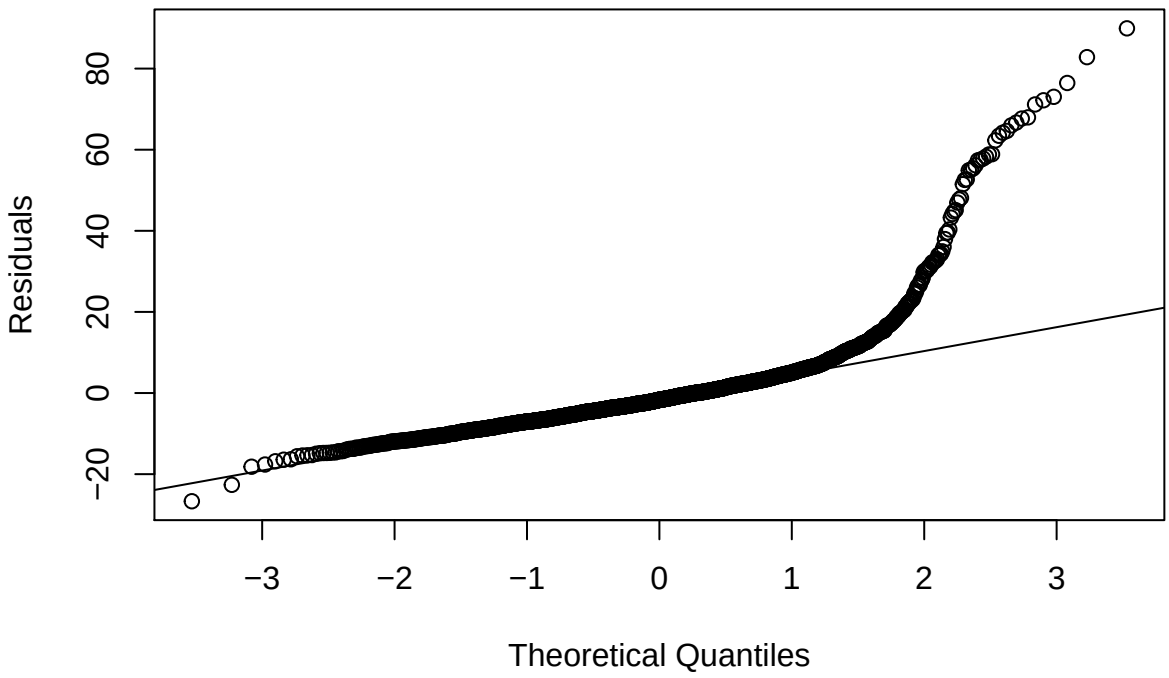



```
var.test(residuals(lmod)[fitted(lmod)>10], residuals(lmod)[fitted(lmod) < 10])
```

```
##
##  F test to compare two variances
##
## data:  residuals(lmod)[fitted(lmod) > 10] and residuals(lmod)[fitted(lmod) < 10]
## F = 1.7922, num df = 940, denom df = 1475, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.597507 2.013601
## sample estimates:
## ratio of variances
##           1.792179
```

Here we check if the residuals have non-constant variance. we find that they do and we will attempt to address this through remidial measures later.

```
qqnorm(residuals(lmod), ylab="Residuals", main="")
qqline(residuals(lmod))
```

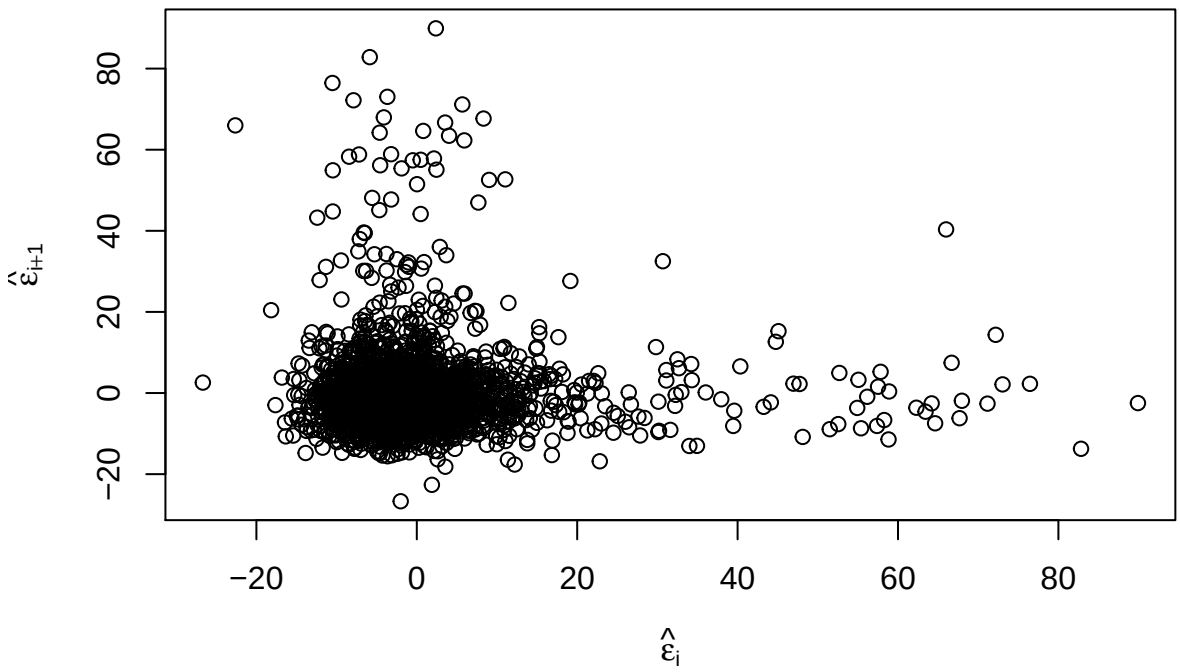



```
shapiro.test(residuals(lmod))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(lmod)
## W = 0.71613, p-value < 2.2e-16
```

Here we check if the residuals are normally distributed but they are clearly not. Again we will attempt to address this later through remidial measures.

```
n<-length(residuals(lmod))
plot(tail(residuals(lmod), n-1) ~ head(residuals(lmod),n-1), xlab = expression(hat(epsilon)[i]), ylab =
```




```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

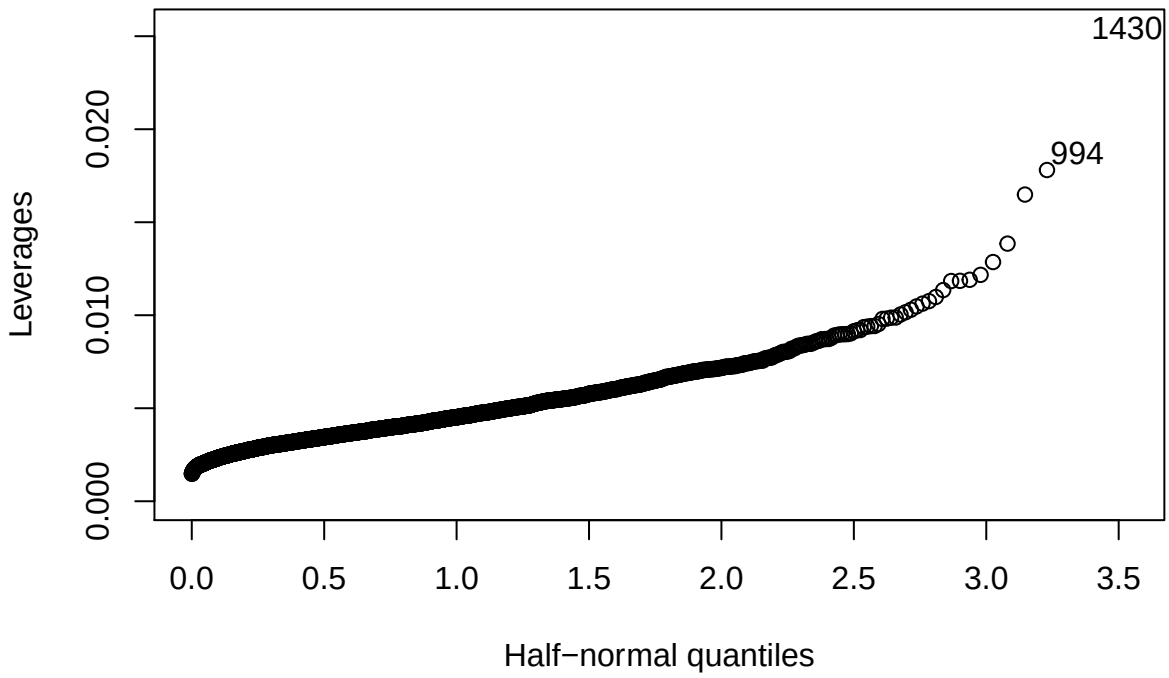
```
dwtest(return_yards ~ clean_catch_binary + actual_kick_direction_R + actual_kick_direction_C + return_di
```

```
##
##  Durbin-Watson test
##
## data:  return_yards ~ clean_catch_binary + actual_kick_direction_R +     actual_kick_direction_C + re
## DW = 2.0342, p-value = 0.7976
## alternative hypothesis: true autocorrelation is greater than 0
```

Here we check if the data is serially correlated and find with a Durbin-Watson test that it is not.

We now turn to finding individual points whose removal could substantially improve the model.

```
hatv<-hatvalues(lmod)
halfnorm(hatv, 2, labs = row.names(puntNoPenalty), ylab = "Leverages")
```



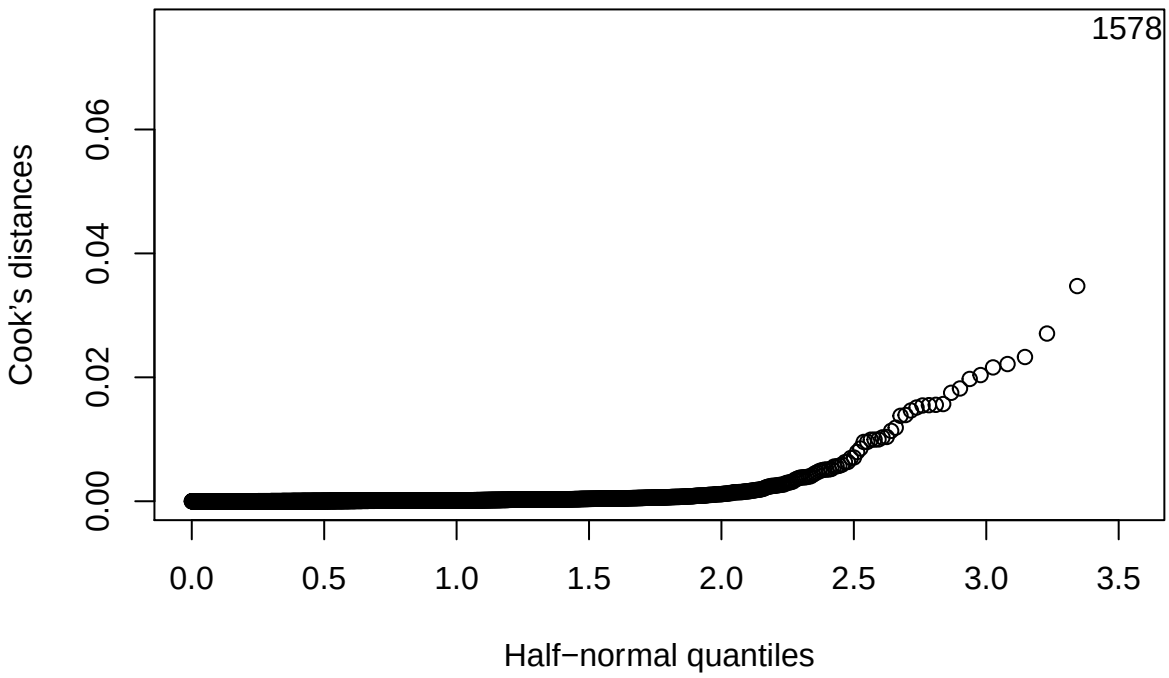
Above we look for high leverage points which are points which are extreme in the space of the predictor variables. In a two-dimensional regression high leverage points which were either far to the left or right on the x-axis. The two punts with highest leverage are labeled with their row number in the data set.

```
stud <- rstudent(lmod)
crit <- -1*qt(.05/(2420*2), 2420-5-1)
stud<-stud[abs(stud)>crit]
stud
```

```
##      101      206      243      262      333      405      552      585
## 7.501634 6.202390 6.459786 6.525484 5.760917 5.372628 5.136804 7.153384
##      608      641      657      724      803      926      945      976
## 5.124885 5.354300 6.283767 4.390227 5.017104 8.145371 6.625556 4.573626
##     1082     1098     1385     1434     1468     1527     1533     1581
## 4.357652 5.607586 5.480246 6.089807 4.294491 5.743552 5.649481 8.891422
##     1731     1807     1853     1877     2107     2252     2292     2324
## 4.652087 4.692606 6.319761 5.692293 6.666902 7.076174 5.618614 6.969703
##     2327
## 5.404281
```

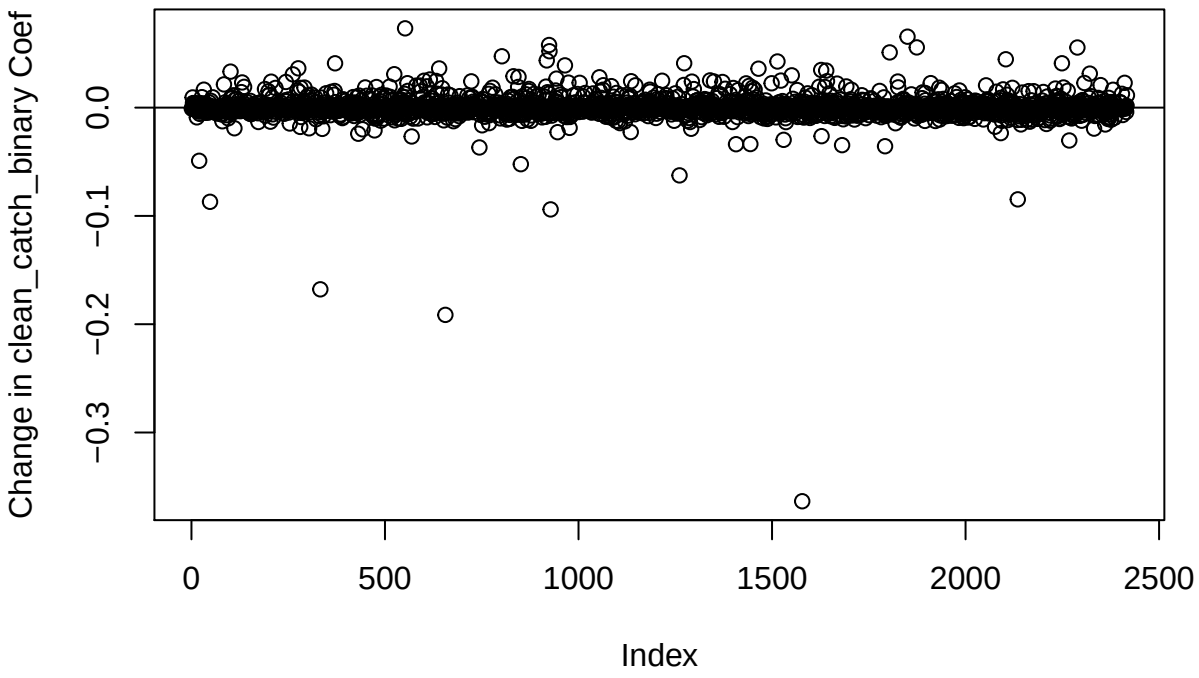
Above we look for outliers which are points with a relatively large residual. Due to the low Multiple R-squared value we have a large number of outliers. Each outlier is labeled with the punt's row number and the associated studentized residual.

```
cook <- cooks.distance(lmod)
halfnorm(cook,1,labs=row.names(puntNoPenalty),ylab="Cook's distances")
```



Above we look for influential points which are points whose inclusion results in large changes in the coefficients associated with the predictor variables. We see that there is one influential point of note.

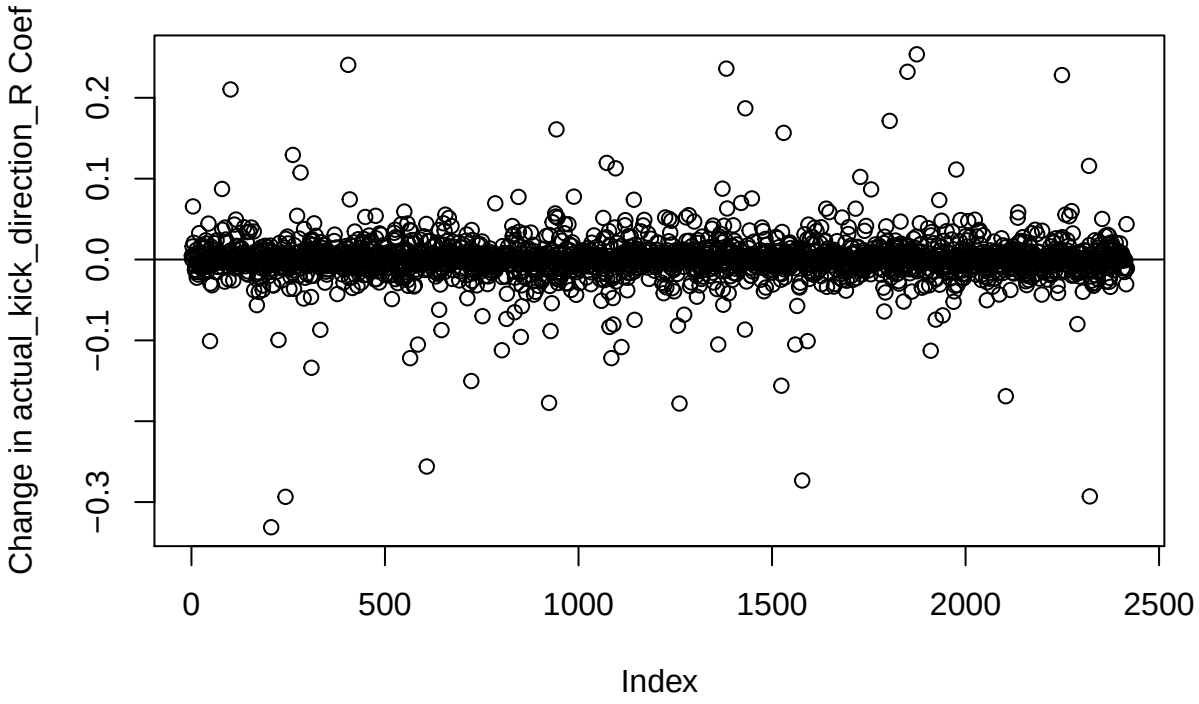
```
plot(dfbeta(lmod)[,2],ylab="Change in clean_catch_binary Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,2]
influential[which.max(abs(influential))]
```

```
##       1581
## -0.3634655
```

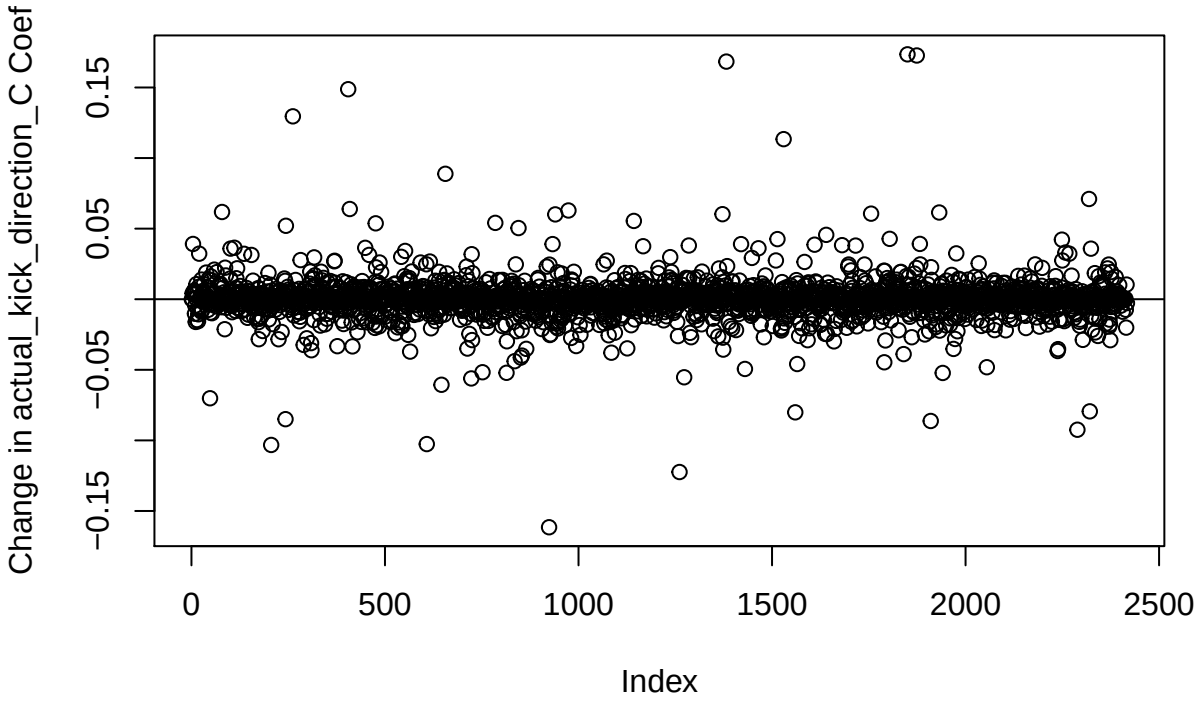
```
plot(dfbeta(lmod)[,3],ylab="Change in actual_kick_direction_R Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,3]
influential[which.max(abs(influential))]
```

```
##        206
## -0.3311372
```

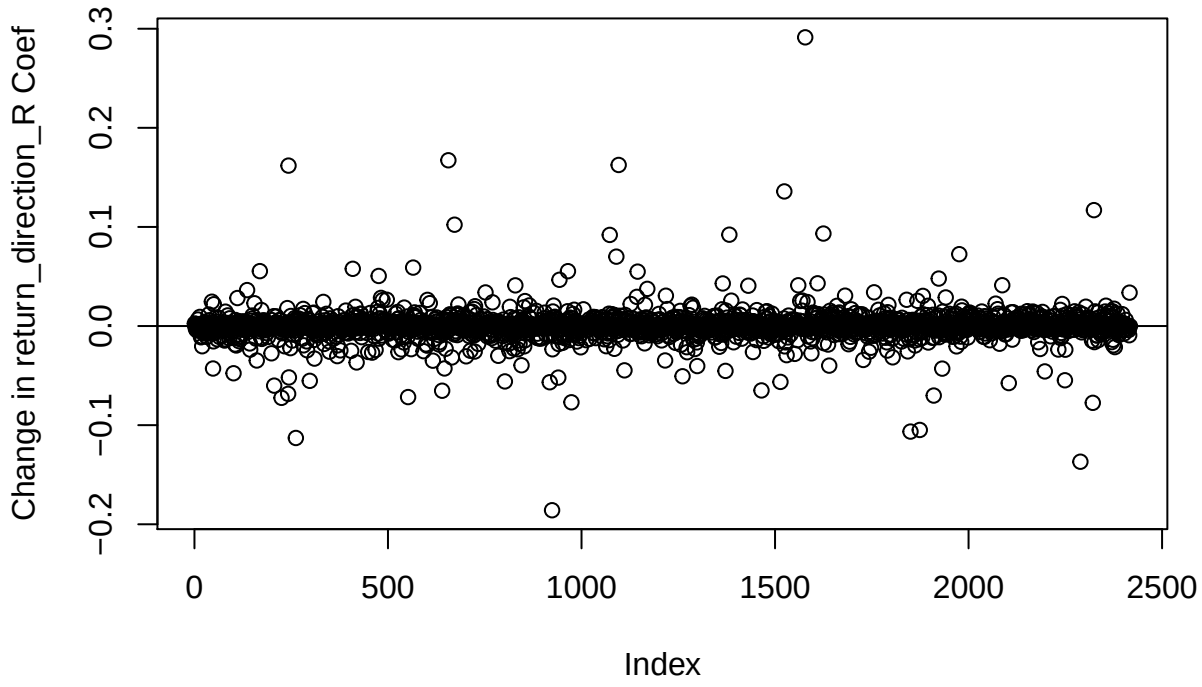
```
plot(dfbeta(lmod)[,4],ylab="Change in actual_kick_direction_C Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,4]
influential[which.max(abs(influential))]
```

```
##      1853
## 0.1734652
```

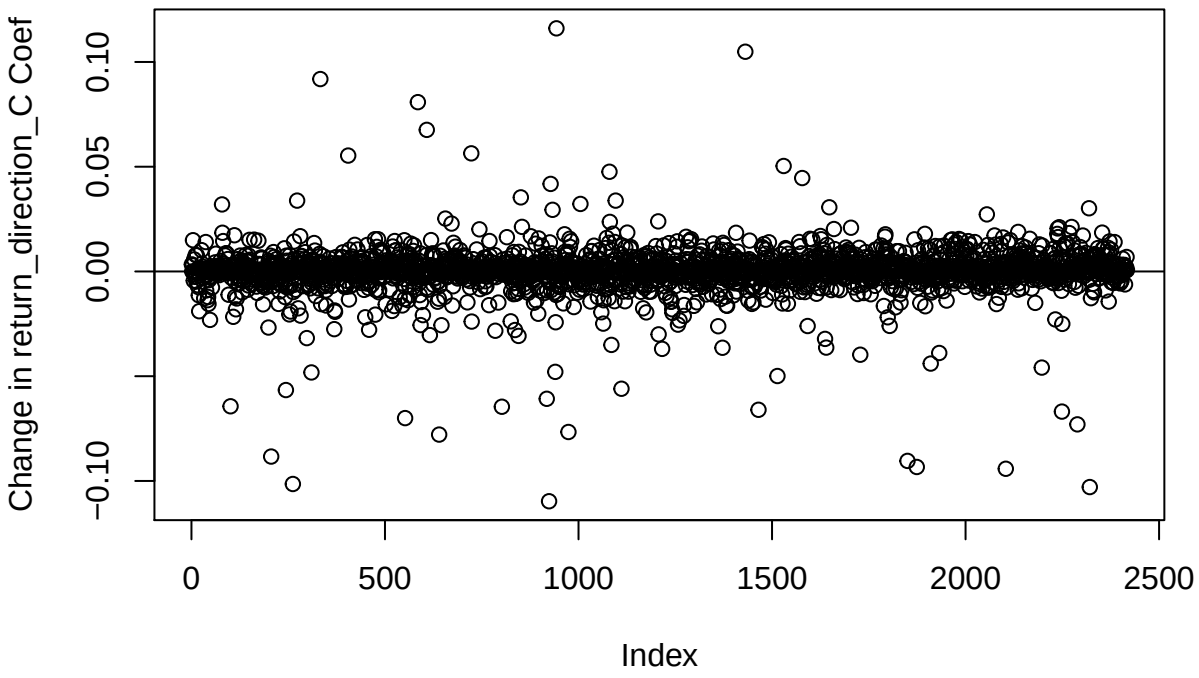
```
plot(dfbeta(lmod)[,5],ylab="Change in return_direction_R Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,5]
influential[which.max(abs(influential))]
```

```
##     1581
## 0.291293
```

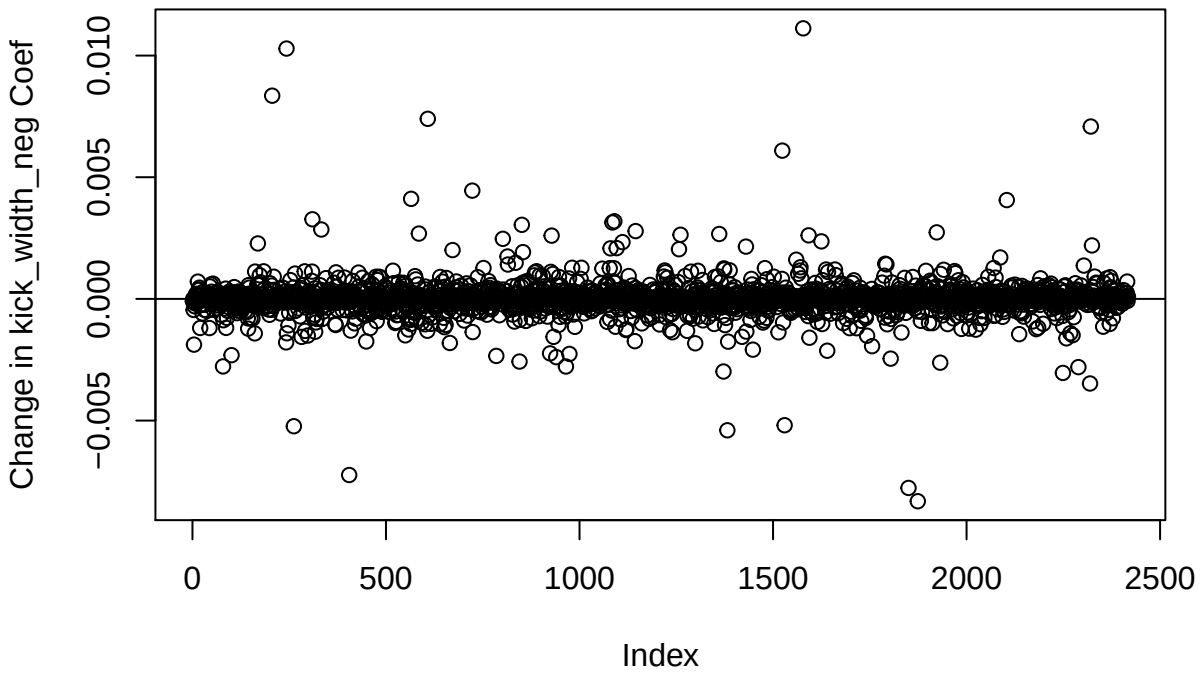
```
plot(dfbeta(lmod)[,6],ylab="Change in return_direction_C Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,6]
influential[which.max(abs(influential))]
```

```
##       945
## 0.1160466
```

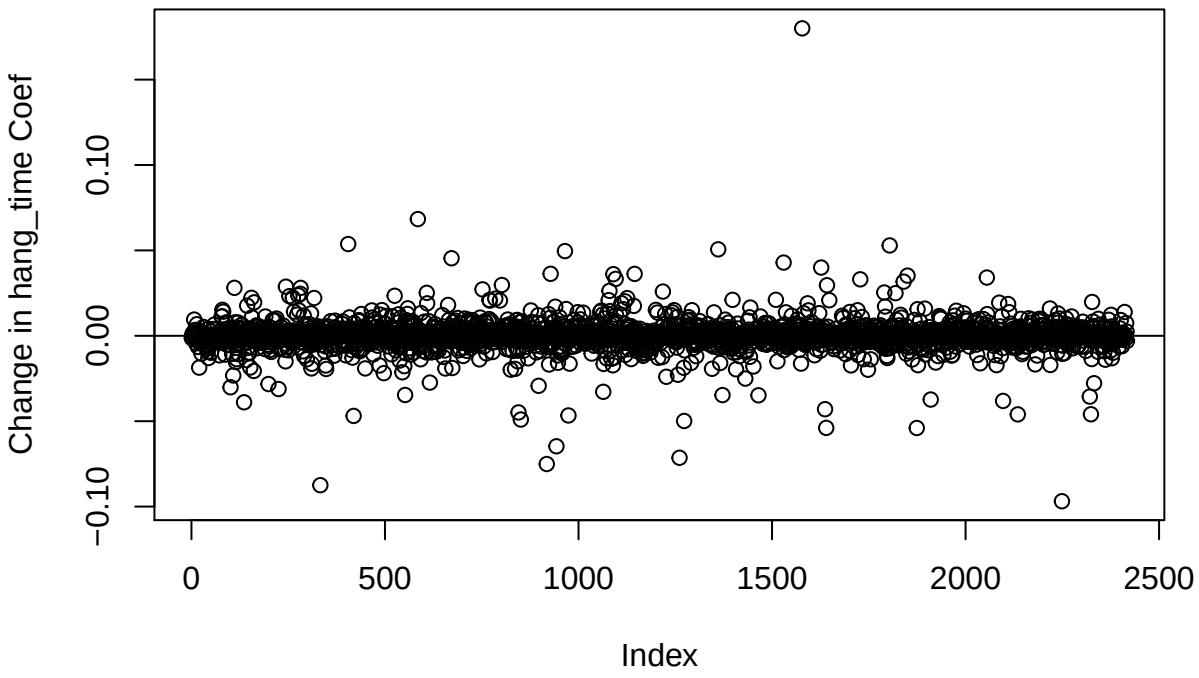
```
plot(dfbeta(lmod)[,7],ylab="Change in kick_width_neg Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,7]
influential[which.max(abs(influential))]
```

```
##       1581
## 0.01111791
```

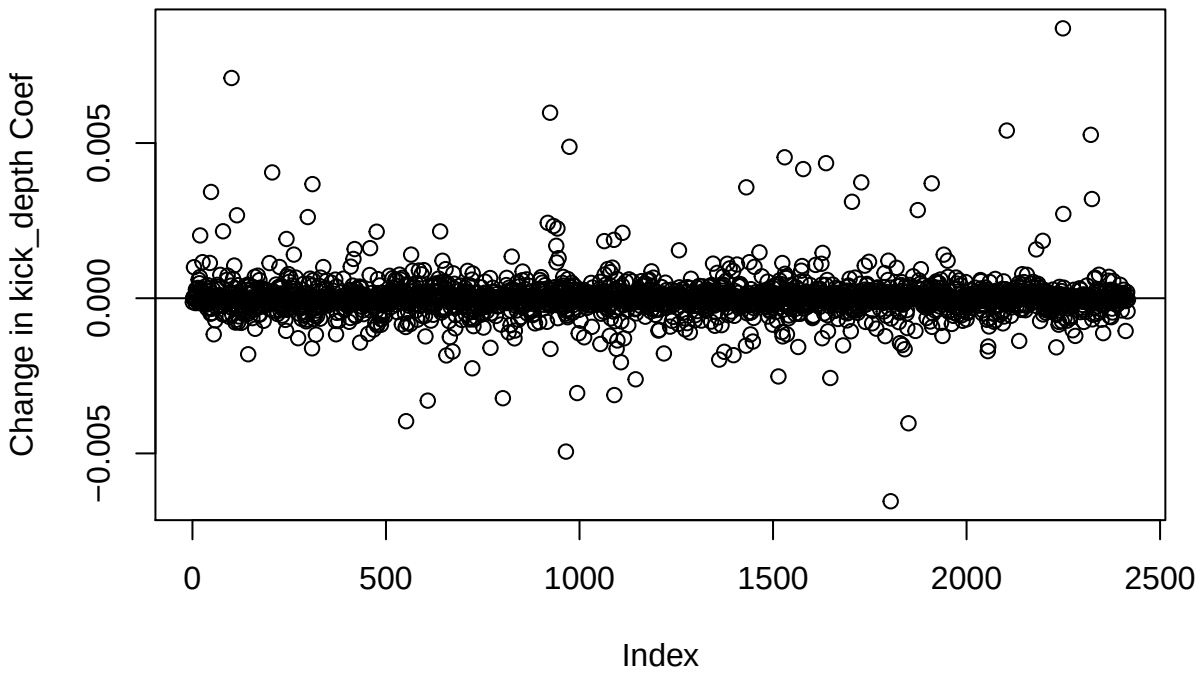
```
plot(dfbeta(lmod)[,8],ylab="Change in hang_time Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,8]
influential[which.max(abs(influential))]
```

```
##      1581
## 0.1800116
```

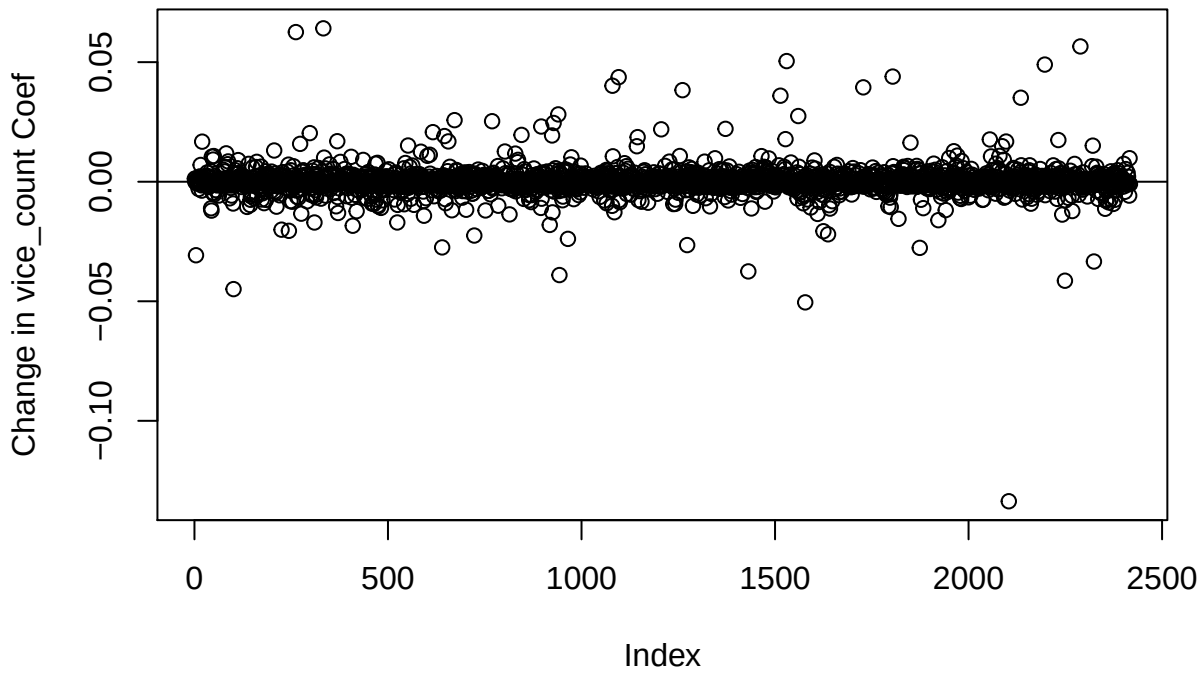
```
plot(dfbeta(lmod)[,9],ylab="Change in kick_depth Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,9]
influential[which.max(abs(influential))]
```

```
##       2252
## 0.00869544
```

```
plot(dfbeta(lmod)[,10],ylab="Change in vice_count Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,10]
influential[which.max(abs(influential))]
```

```
##       2107
## -0.1336258
```

Above we take a more granular look at influential points by graphing the change each point has on each coefficient individually. We see that that the inclusion of punt number 1581 and 2107 result in relatively large difference in each coefficient so we can improve the model significantly by removing them which we do below.

```
lmodu<-lm(return_yards ~ clean_catch_binary + actual_kick_direction_R + actual_kick_direction_C + return
summary(lmodu)
```

```
##
## Call:
## lm(formula = return_yards ~ clean_catch_binary + actual_kick_direction_R +
##     actual_kick_direction_C + return_direction_R + return_direction_C +
##     kick_width_neg + hang_time + kick_depth + vise_count, data = puntNoPenalty,
##     subset = (row.names(puntNoPenalty) != "2107" & row.names(puntNoPenalty) !=
##         "1581"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.702  -5.380  -1.669   2.628  82.717
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -0.95123    2.17175  -0.438 0.661426
```

```
## clean_catch_binary       4.65354    0.61284   7.593 4.43e-14 ***
## actual_kick_direction_R  1.12091    1.31570   0.852 0.394325
## actual_kick_direction_C  2.15978    0.77971   2.770 0.005649 **
## return_direction_R       3.83695    0.59130   6.489 1.05e-10 ***
## return_direction_C      -0.30360    0.50350  -0.603 0.546571
## kick_width_neg           0.05343    0.03262   1.638 0.101573
## hang_time               -3.89712    0.51479  -7.570 5.27e-14 ***
## kick_depth               0.36283    0.03227  11.242  < 2e-16 ***
## vise_count               0.94213    0.27064   3.481 0.000508 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.07 on 2405 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.1317, Adjusted R-squared:  0.1284
## F-statistic: 40.52 on 9 and 2405 DF,  p-value: < 2.2e-16
```

Above is the regression with the two influential points removed. Notice the increase in the Multiple-R squared value.

```
min(puntNoPenalty$return_yards)
```

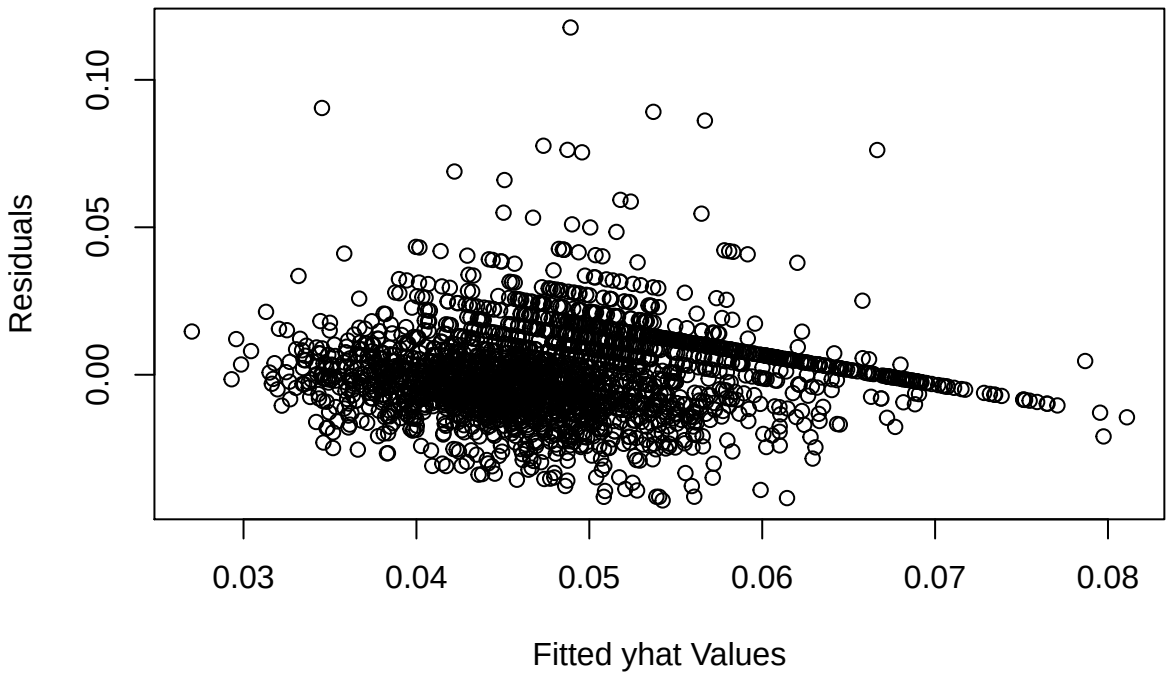
```
## [1] -14
```

```
lmodu<-lm(1/(return_yards-(-1+min(puntNoPenalty$return_yards))) ~ clean_catch_binary + actual_kick_dire
summary(lmodu)
```

```
##
## Call:
## lm(formula = 1/(return_yards - (-1 + min(puntNoPenalty$return_yards))) ~
##     clean_catch_binary + actual_kick_direction_R + actual_kick_direction_C +
##         return_direction_R + return_direction_C + kick_width_neg +
##         hang_time + kick_depth + vise_count, data = puntNoPenalty,
##     subset = (row.names(puntNoPenalty) != "2107" & row.names(puntNoPenalty) !=
##         "1581") & row.names(puntNoPenalty) != "225", na.action = na.exclude)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.042612 -0.008605 -0.001125  0.007532  0.117752
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              6.215e-02  3.131e-03  19.849  < 2e-16 ***
## clean_catch_binary      -1.028e-02  8.831e-04 -11.641  < 2e-16 ***
## actual_kick_direction_R -3.023e-03  1.897e-03  -1.594 0.111123
## actual_kick_direction_C -3.215e-03  1.124e-03  -2.861 0.004256 **
## return_direction_R      -1.006e-02  8.530e-04 -11.789  < 2e-16 ***
## return_direction_C      -2.686e-03  7.256e-04  -3.702 0.000219 ***
## kick_width_neg          -8.754e-05  4.701e-05  -1.862 0.062719 .
## hang_time                7.675e-03  7.420e-04  10.343  < 2e-16 ***
## kick_depth              -6.322e-04  4.651e-05 -13.593  < 2e-16 ***
## vise_count              -6.512e-04  3.902e-04  -1.669 0.095256 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01451 on 2404 degrees of freedom
```

```
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.2235, Adjusted R-squared:  0.2206
## F-statistic:  76.9 on 9 and 2404 DF,  p-value: < 2.2e-16
```

```
plot(fitted(lmodu), residuals(lmodu), xlab = "Fitted yhat Values", ylab = "Residuals")
```

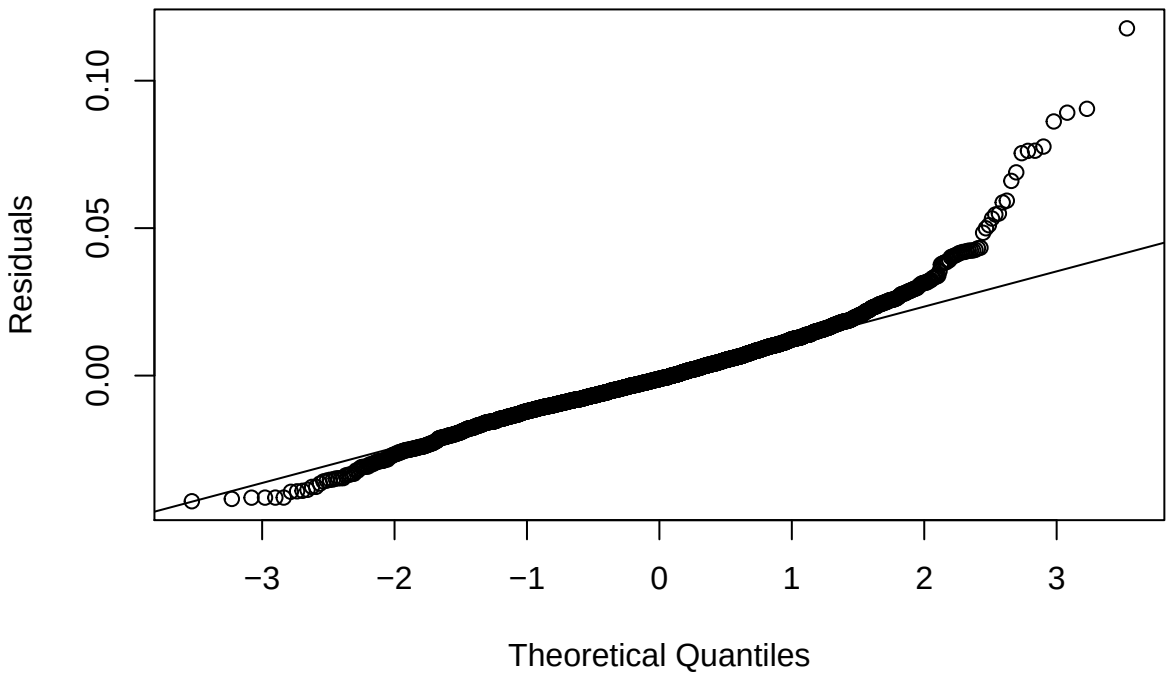



```
var.test(residuals(lmodu)[fitted(lmod)>0.045], residuals(lmodu)[fitted(lmod) < 0.045])
```

```
##
##  F test to compare two variances
##
## data:  residuals(lmodu)[fitted(lmod) > 0.045] and residuals(lmodu)[fitted(lmod) < 0.045]
## F = 2.0838, num df = 2341, denom df = 71, p-value = 0.0001418
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.449544 2.839452
## sample estimates:
## ratio of variances
##           2.083827
```

Above we apply a transformation to the response variable return_yards. Namely, we now regress on 1/(return_yards+15) where we add 15 to translate the data so that all punt_returns are in positive yardage. The minimum return_yards was -14 so adding 15 is sufficient. This transformation eliminates the non-constant variance of the residuals which we detected previously. On one hand the improves the fit, but also makes the model less intuitive. The value of this tradeoff is left to the discretion of the model user.

```
qqnorm(residuals(lmodu), ylab="Residuals", main="")
qqline(residuals(lmodu))
```

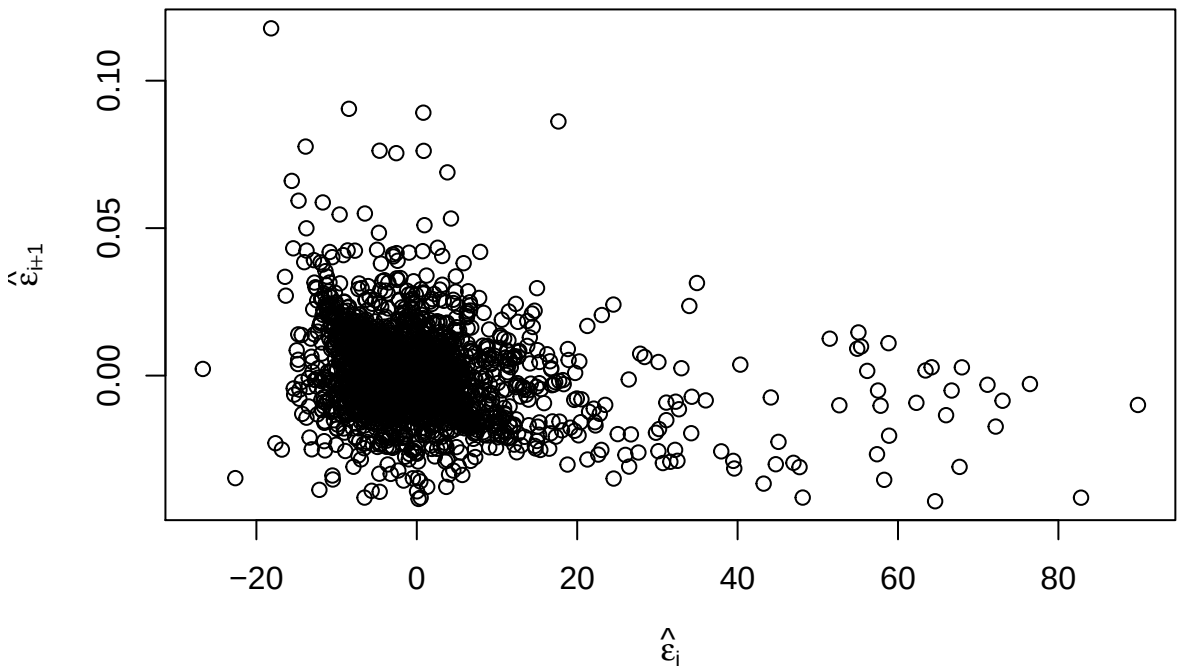



```
shapiro.test(residuals(lmodu))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(lmodu)
## W = 0.94508, p-value < 2.2e-16
```

As seen above the transformation also makes the residuals more closely follow a normal distribution, though not as closely as we would have liked.

```
n<-length(residuals(lmodu))
plot(tail(residuals(lmodu), n-1) ~ head(residuals(lmod),n-1), xlab = expression(hat(epsilon)[i]), ylab =
```



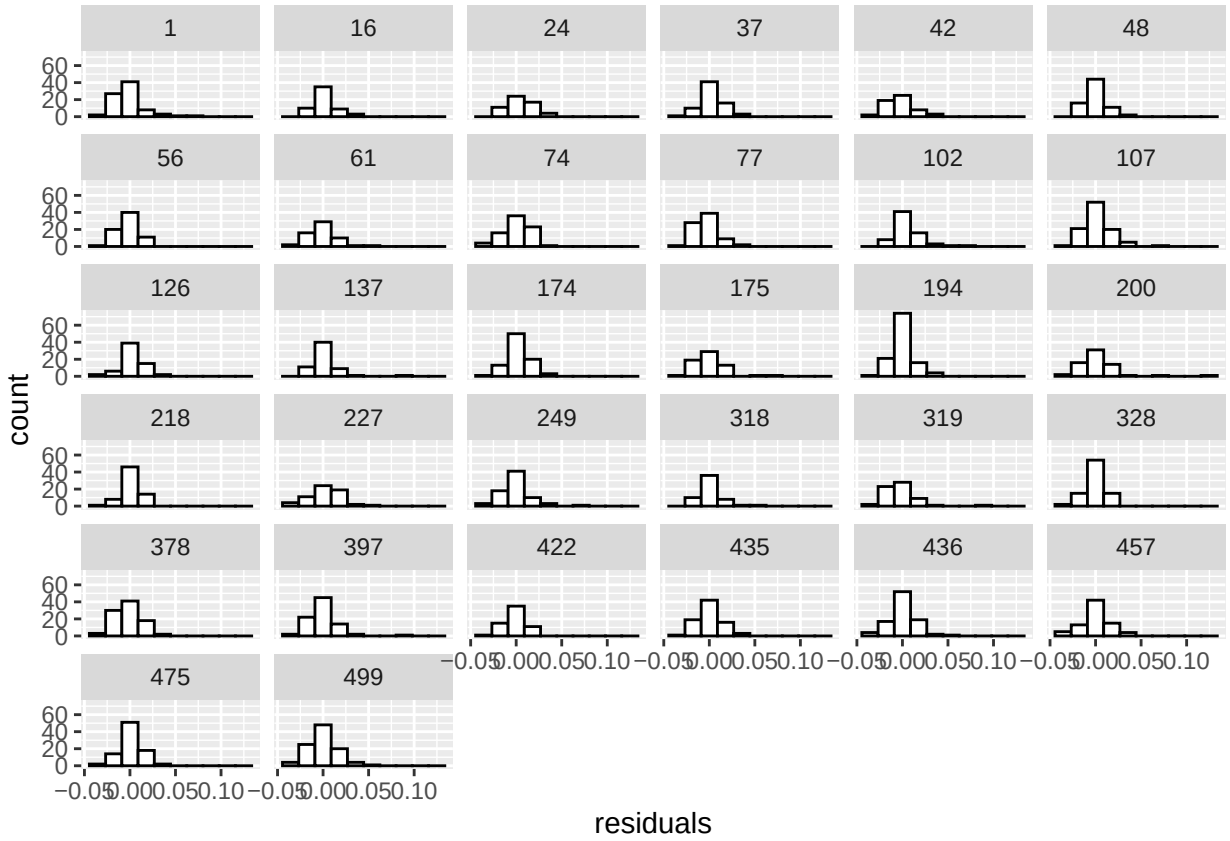

```
library(lmtest)
dwtest(return_yards ~ hang_time + kick_depth + kick_yards + vise_count + field_position_pos, data=puntN
```

```
##
##  Durbin-Watson test
##
## data:  return_yards ~ hang_time + kick_depth + kick_yards + vise_count +     field_position_pos
## DW = 2.0246, p-value = 0.7254
## alternative hypothesis: true autocorrelation is greater than 0
```

Lastly, we again check if the observations remain uncorrelated and indeed that is the case.

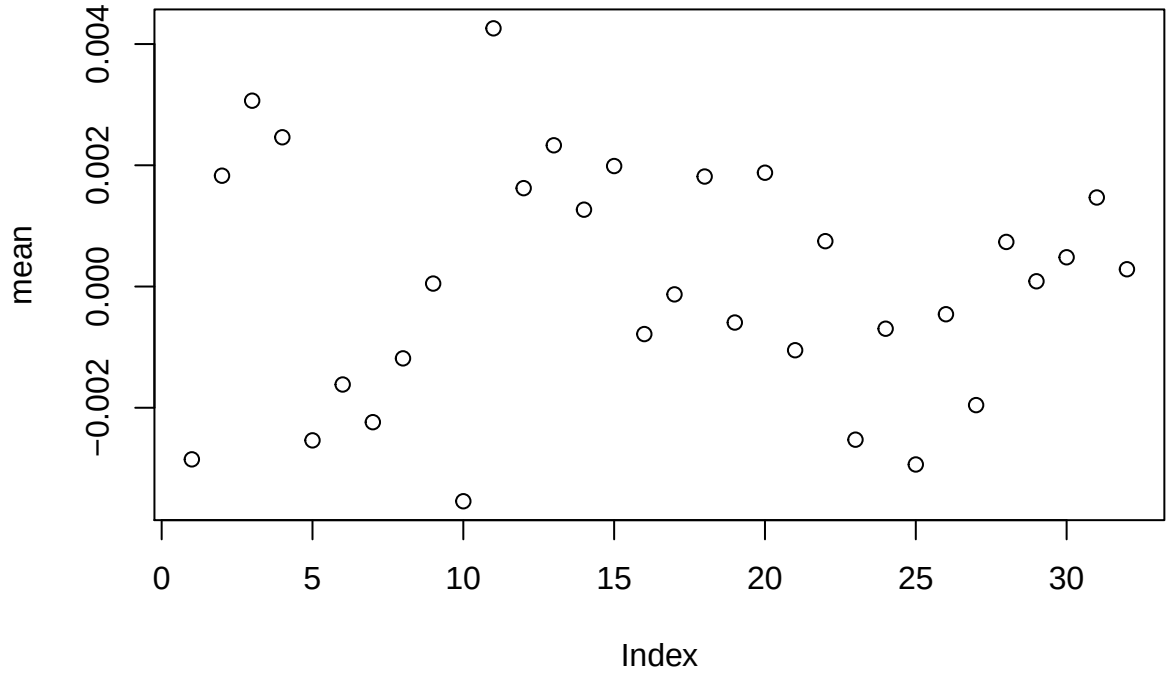
Future Directions: Using this model to grade punt returners. Summing a returner's residuals could serve as an alternative metric to average return yards.

```
puntNoPenalty <- puntNoPenalty[-c(2107, 1581, 225), ]
puntNoPenalty$residuals <- residuals(lmodu)
unique<-unique(puntNoPenalty$offense_team_id)

puntNoPenalty<-subset(puntNoPenalty, (!is.na(residuals)) & (!is.na(defense_team_id)))

ggplot(puntNoPenalty, aes(x = residuals)) +
  geom_histogram(fill = "white", colour = "black", bins = 10) +
  facet_wrap(offense_team_id ~ .)
```



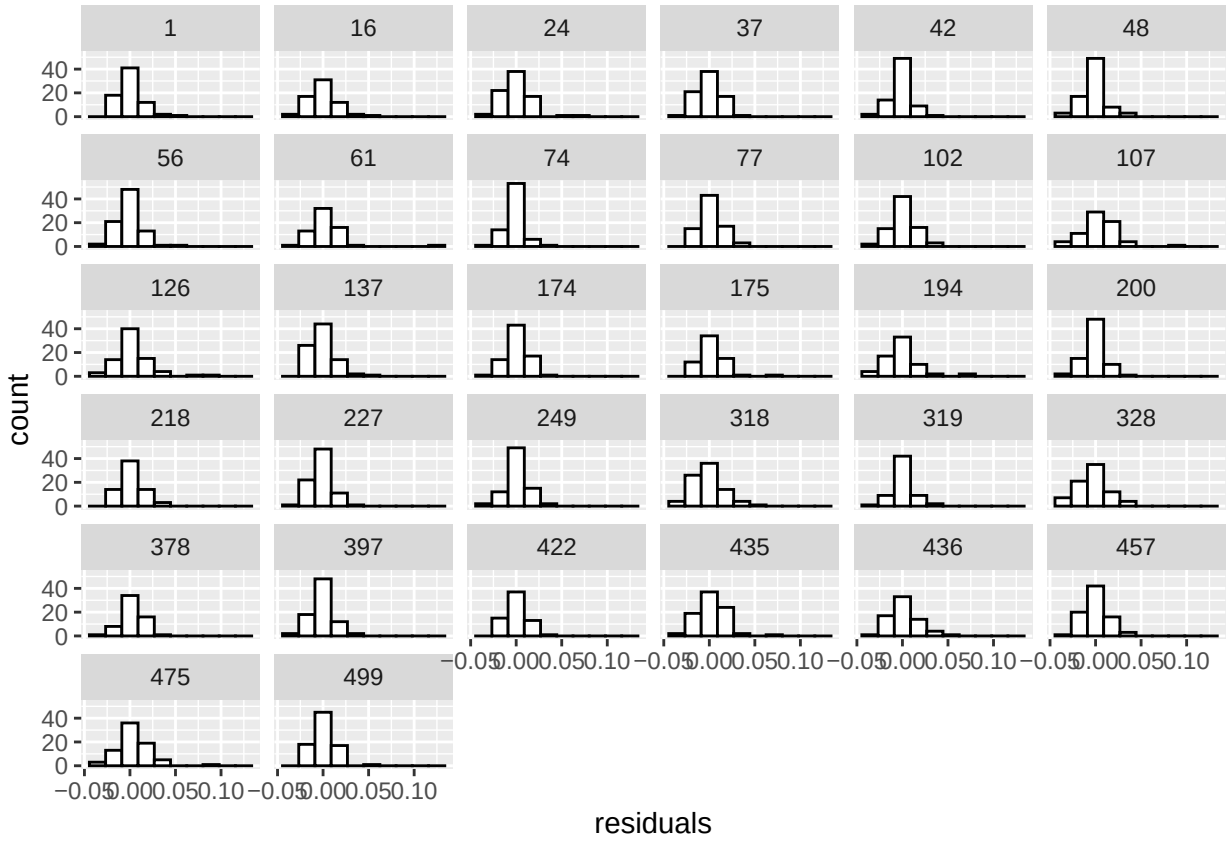
We now look at the residuals for each team's offensive unit. If one team in particular had a devised a strategy which allowed them to consistently outperform the model we would expect the corresponding histogram to be skewed left. However, this is not the case as all the histograms are approximately normally distributed. This observation is confirmed via the scatterplot for each team's mean residual below.

```
mean <- tapply(puntNoPenalty$residuals, puntNoPenalty$offense_team_id, mean)
plot(mean)
```

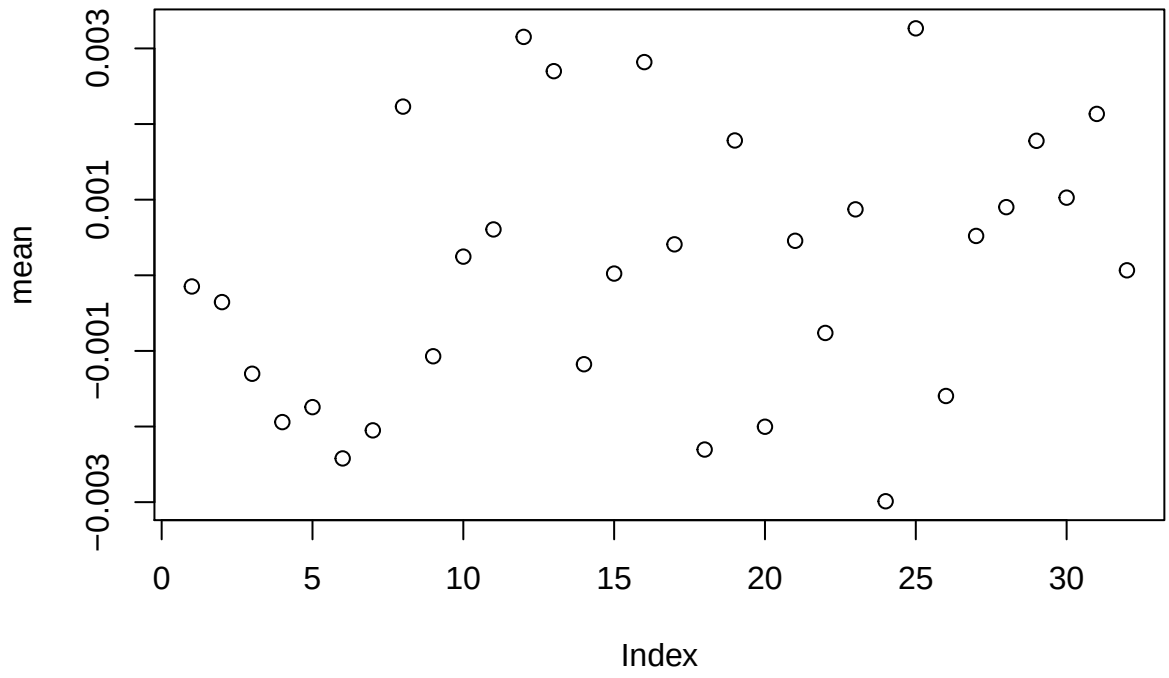


Below we performed the same analysis on each team's defensive unit. This time we were interested in determining whether one team had devised a strategy to routinely stop punt returners short of their expected return length. Again, this was not the case.

```
ggplot(puntNoPenalty, aes(x = residuals)) +
  geom_histogram(fill = "white", colour = "black", bins = 10) +
  facet_wrap(defense_team_id ~ .)
```




```
mean <- tapply(puntNoPenalty$residuals, puntNoPenalty$defense_team_id, mean)
plot(mean)
```



Now we attempt to create a universal model, one that predicts the length of a punt return on plays with and without penalty.

```
lmodAll<-lm(return_yards ~ clock_truncated + hash_R + hash_C + clean_catch_binary + actual_kick_directi
summary(lmodAll)
```

```
##
## Call:
## lm(formula = return_yards ~ clock_truncated + hash_R + hash_C +
##     clean_catch_binary + actual_kick_direction_R + actual_kick_direction_C +
##     kick_type_NORMAL + return_direction_R + return_direction_C +
##     hang_time + quarter + field_position_pos + off_score + def_score +
##     score_differential + garbage_time + kick_depth + kick_width_neg +
##     kick_yards + punt_rush_count + fumble + gunners_count + vise_count +
##     penalty_offensive + penalty_defensive, data = puntOrig)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.414  -5.210  -1.424   2.579  90.205
##
## Coefficients: (1 not defined because of singularities)
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               2.17437    3.85998   0.563  0.57326
## clock_truncated          -0.03420    0.04345  -0.787  0.43125
## hash_R                    0.56880    0.40419   1.407  0.15946
## hash_C                    0.82081    0.52442   1.565  0.11765
## clean_catch_binary        4.49157    0.66254   6.779 1.45e-11 ***
```

```
## actual_kick_direction_R  0.77587    1.16060   0.669  0.50386
## actual_kick_direction_C  2.00584    0.68657   2.922  0.00351 **
## kick_type_NORMAL         0.40133    0.59735   0.672  0.50173
## return_direction_R       3.41162    0.51016   6.687 2.71e-11 ***
## return_direction_C      -0.40136    0.44136  -0.909  0.36324
## hang_time               -3.47960    0.46260  -7.522 7.12e-14 ***
## quarter                 -0.06900    0.27094  -0.255  0.79899
## field_position_pos      -0.01635    0.01860  -0.879  0.37953
## off_score               -0.01666    0.02715  -0.614  0.53952
## def_score                0.02331    0.02503   0.931  0.35171
## score_differential            NA         NA      NA       NA
## garbage_time            -1.44055    1.28932  -1.117  0.26396
## kick_depth               0.22120    0.10586   2.090  0.03674 *
## kick_width_neg           0.04153    0.02926   1.420  0.15582
## kick_yards               0.11157    0.10257   1.088  0.27680
## punt_rush_count          0.02771    0.10738   0.258  0.79642
## fumble                  -1.75697    1.35860  -1.293  0.19604
## gunners_count           -0.95904    1.57286  -0.610  0.54208
## vise_count               0.58522    0.25488   2.296  0.02174 *
## penalty_offensive       -5.48176    0.48801 -11.233  < 2e-16 ***
## penalty_defensive       -2.07344    0.91722  -2.261  0.02386 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.755 on 2963 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.1432, Adjusted R-squared:  0.1363
## F-statistic: 20.63 on 24 and 2963 DF,  p-value: < 2.2e-16
```

```
finalLmodAll<-lm(return_yards ~ clean_catch_binary + actual_kick_direction_R + actual_kick_direction_C +
summary(finalLmodAll)
```

```
##
## Call:
## lm(formula = return_yards ~ clean_catch_binary + actual_kick_direction_R +
##     actual_kick_direction_C + return_direction_R + return_direction_C +
##     hang_time + kick_depth + vise_count + penalty_offensive +
##     penalty_defensive, data = puntOrig)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -25.950  -5.242  -1.515   2.620  90.829
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -0.32184    1.86825  -0.172  0.86324
## clean_catch_binary       4.05180    0.54746   7.401 1.75e-13 ***
## actual_kick_direction_R  2.47853    0.60616   4.089 4.45e-05 ***
## actual_kick_direction_C  2.94239    0.48593   6.055 1.58e-09 ***
## return_direction_R       3.32561    0.49975   6.655 3.37e-11 ***
## return_direction_C      -0.42127    0.43915  -0.959  0.33749
## hang_time               -3.63226    0.45627  -7.961 2.41e-15 ***
## kick_depth               0.34594    0.02853  12.128  < 2e-16 ***
## vise_count               0.61555    0.23661   2.602  0.00933 **
## penalty_offensive       -5.40051    0.48703 -11.089  < 2e-16 ***
```

```
## penalty_defensive       -2.03777    0.91589  -2.225  0.02616 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.758 on 2977 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.1386, Adjusted R-squared:  0.1357
## F-statistic: 47.92 on 10 and 2977 DF,  p-value: < 2.2e-16
```

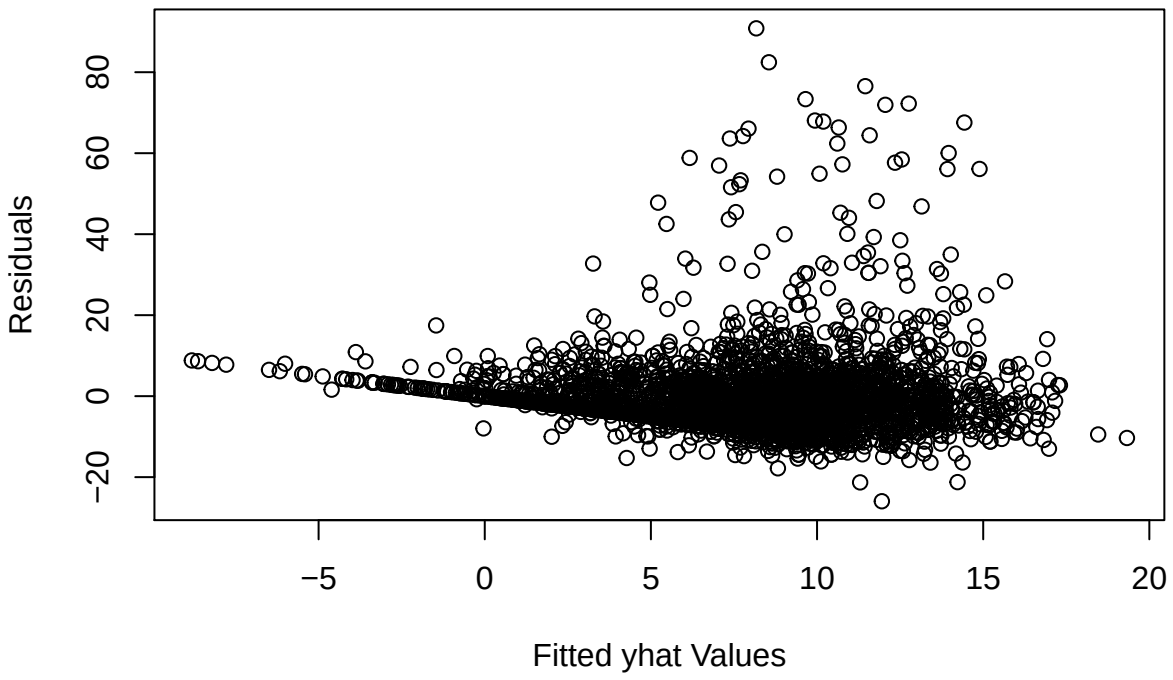
```
anova(finalLmodAll, lmodAll)
```

```
## Analysis of Variance Table
##
## Model 1: return_yards ~ clean_catch_binary + actual_kick_direction_R +
##     actual_kick_direction_C + return_direction_R + return_direction_C +
##     hang_time + kick_depth + vise_count + penalty_offensive +
##     penalty_defensive
## Model 2: return_yards ~ clock_truncated + hash_R + hash_C + clean_catch_binary +
##     actual_kick_direction_R + actual_kick_direction_C + kick_type_NORMAL +
##     return_direction_R + return_direction_C + hang_time + quarter +
##     field_position_pos + off_score + def_score + score_differential +
##     garbage_time + kick_depth + kick_width_neg + kick_yards +
##     punt_rush_count + fumble + gunners_count + vise_count + penalty_offensive +
##     penalty_defensive
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1   2977 283466
## 2   2963 281967 14    1499.2 1.1253 0.3292
```

```
lmod <- finalLmodAll
```

After the reducing the model to only the significant predictors we are left with our final model labeled finalLmodAll above. Below we check the validity of our inherent assumptions reharding linear regression and remove influential points/outliers just as we had previously done.

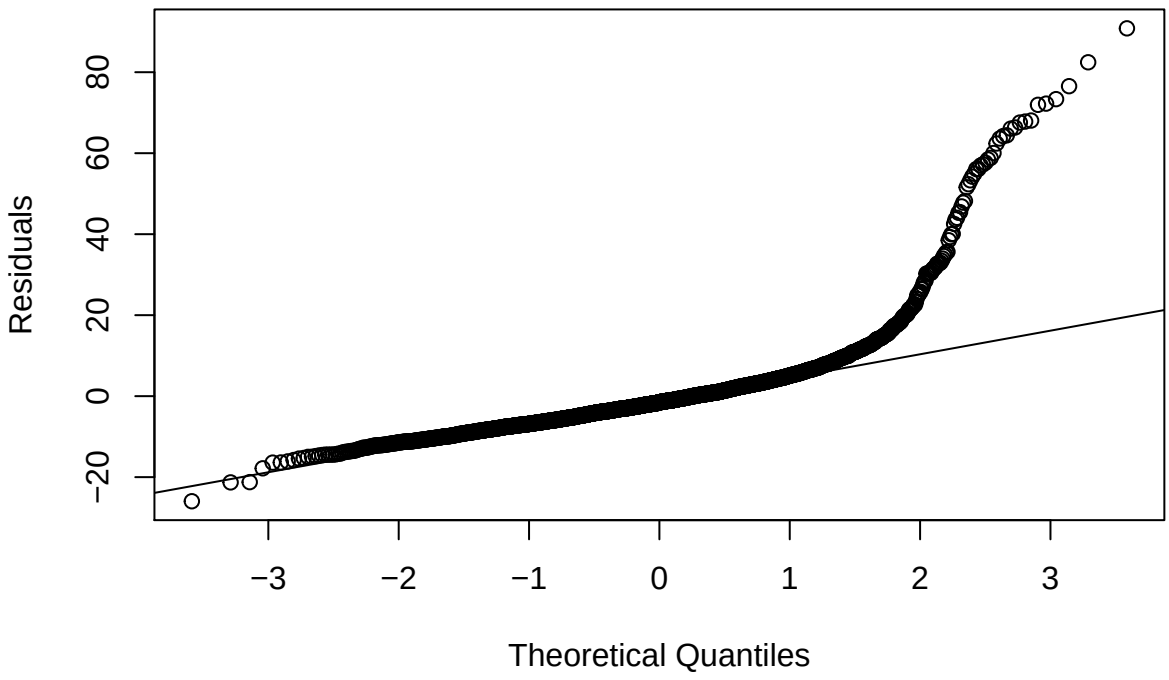
```
plot(fitted(lmod), residuals(lmod), xlab = "Fitted yhat Values", ylab = "Residuals")
```




```
var.test(residuals(lmod)[fitted(lmod)>9], residuals(lmod)[fitted(lmod)<9])
```

```
##
##  F test to compare two variances
##
## data:  residuals(lmod)[fitted(lmod) > 9] and residuals(lmod)[fitted(lmod) < 9]
## F = 1.8861, num df = 1296, denom df = 1690, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.703132 2.090153
## sample estimates:
## ratio of variances
##           1.886084
```

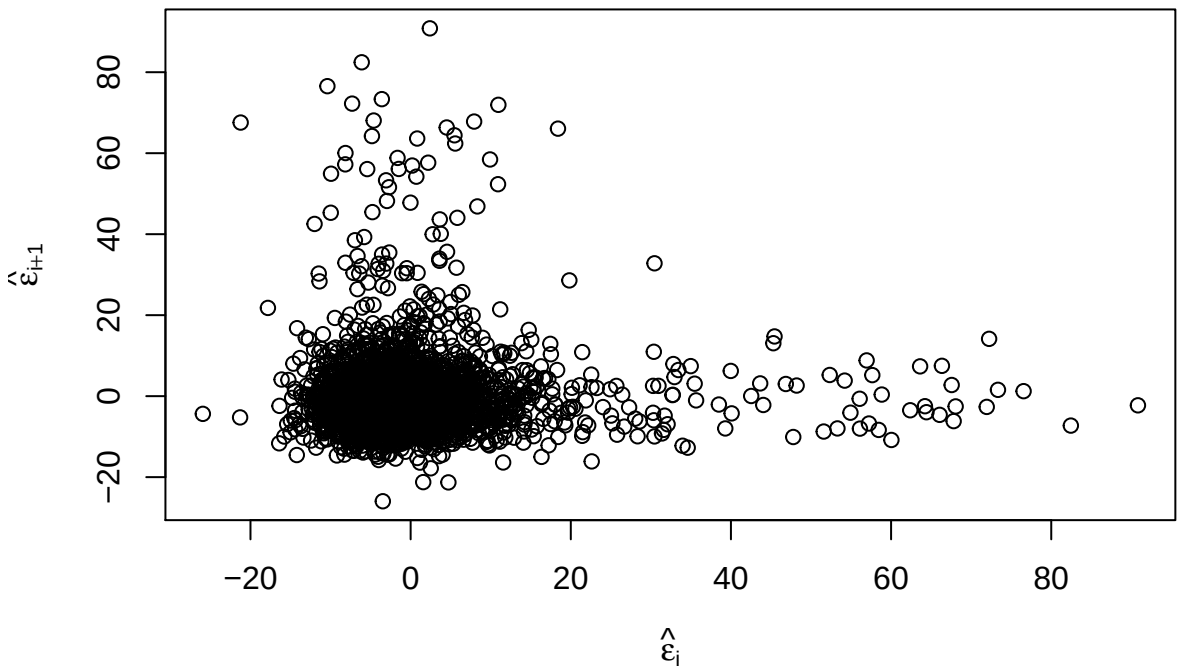
```
qqnorm(residuals(lmod), ylab="Residuals", main="")
qqline(residuals(lmod))
```




```
shapiro.test(residuals(lmod))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(lmod)
## W = 0.72061, p-value < 2.2e-16
```

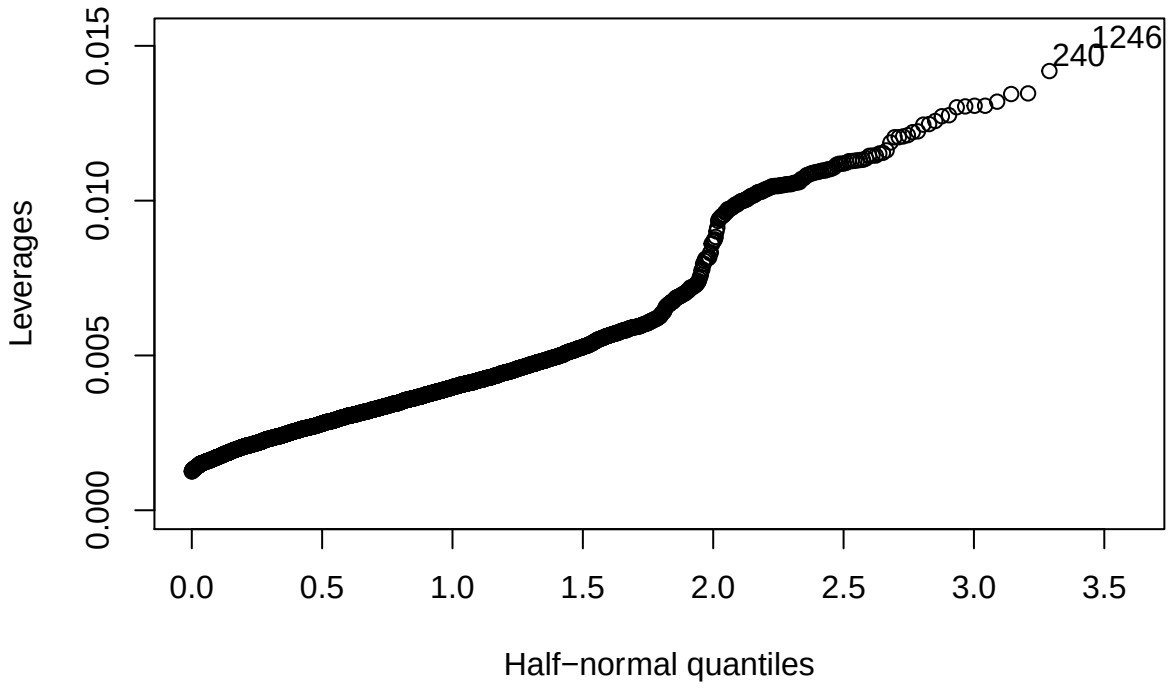
```
n<-length(residuals(lmod))
plot(tail(residuals(lmod), n-1) ~ head(residuals(lmod),n-1), xlab = expression(hat(epsilon)[i]), ylab =
```




```
library(lmtest)
dwtest(return_yards ~ clean_catch_binary + actual_kick_direction_R + actual_kick_direction_C + return_di
```

```
##
##  Durbin-Watson test
##
## data:  return_yards ~ clean_catch_binary + actual_kick_direction_R +     actual_kick_direction_C + re
## DW = 2.008, p-value = 0.585
## alternative hypothesis: true autocorrelation is greater than 0
```

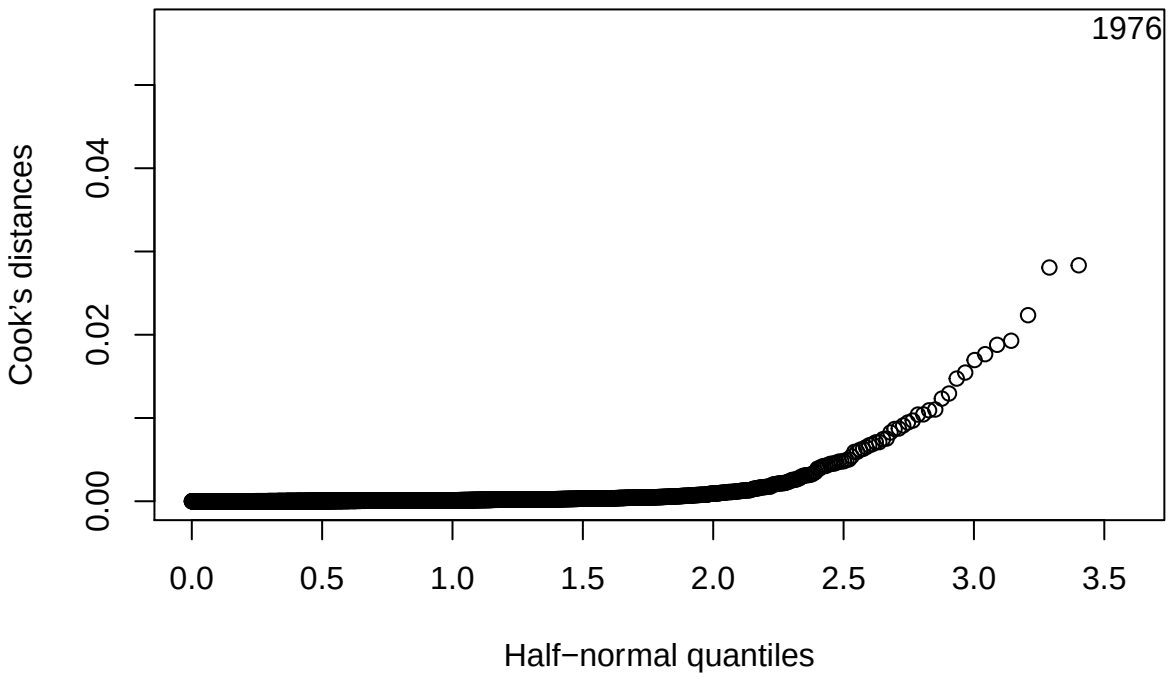
```
hatv<-hatvalues(lmod)
halfnorm(hatv, 2, labs = row.names(puntOrig), ylab = "Leverages")
```



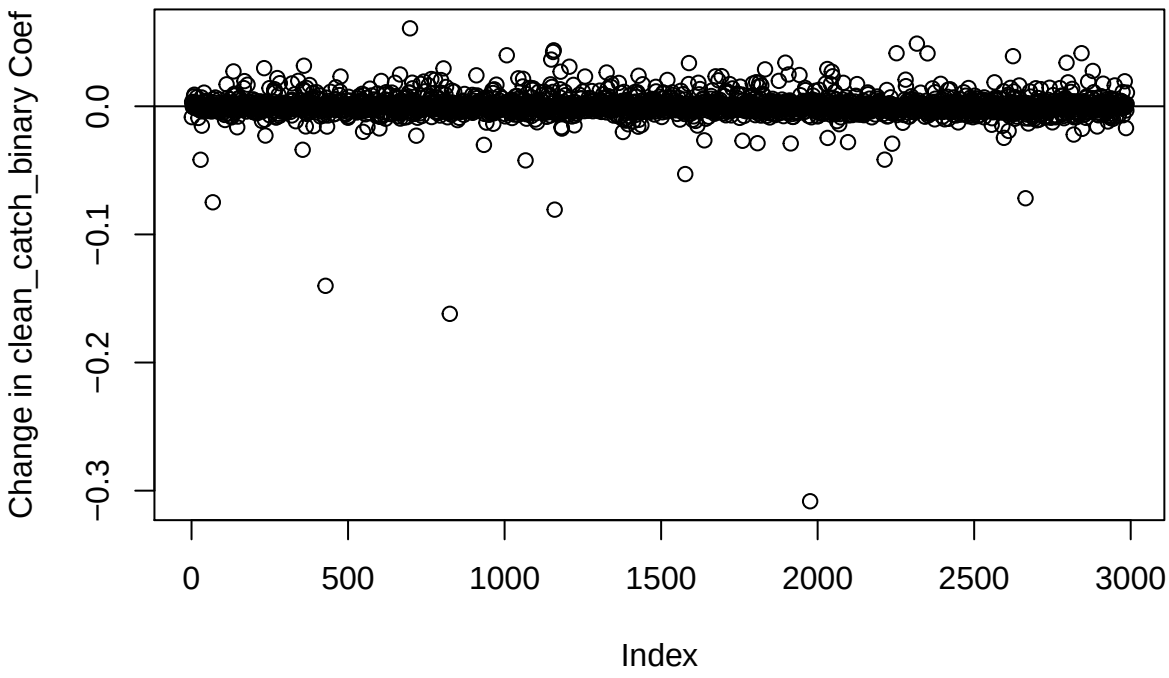

```
stud <- rstudent(lmod)
crit <- -1*qt(.05/(2988*2), 2988-10-1)
stud<-stud[abs(stud)>crit]
stud
```

```
##      134      274      319      341      355      428      519      698
## 7.937937 6.655521 6.986928 6.860676 6.845069 6.083135 5.587304 5.395802
##      736      766      806      826      911     1008     1157     1181
## 7.594155 5.493090 5.663301 6.645281 4.677267 5.315656 8.566159 7.017762
##     1209     1222     1361     1381     1667     1735     1794     1835
## 4.379974 4.825228 4.663635 6.039191 4.496431 5.783791 6.448326 4.531854
##     1911     1917     1979     2159     2255     2320     2354     2627
## 6.197713 5.952586 9.479831 4.969346 4.926495 6.572211 5.904796 7.054127
##     2798     2846     2882     2885
## 7.484525 5.880423 7.447311 5.789411
```

```
cook <- cooks.distance(lmod)
halfnorm(cook,1,labs=row.names(puntOrig),ylab="Cook's distances")
```



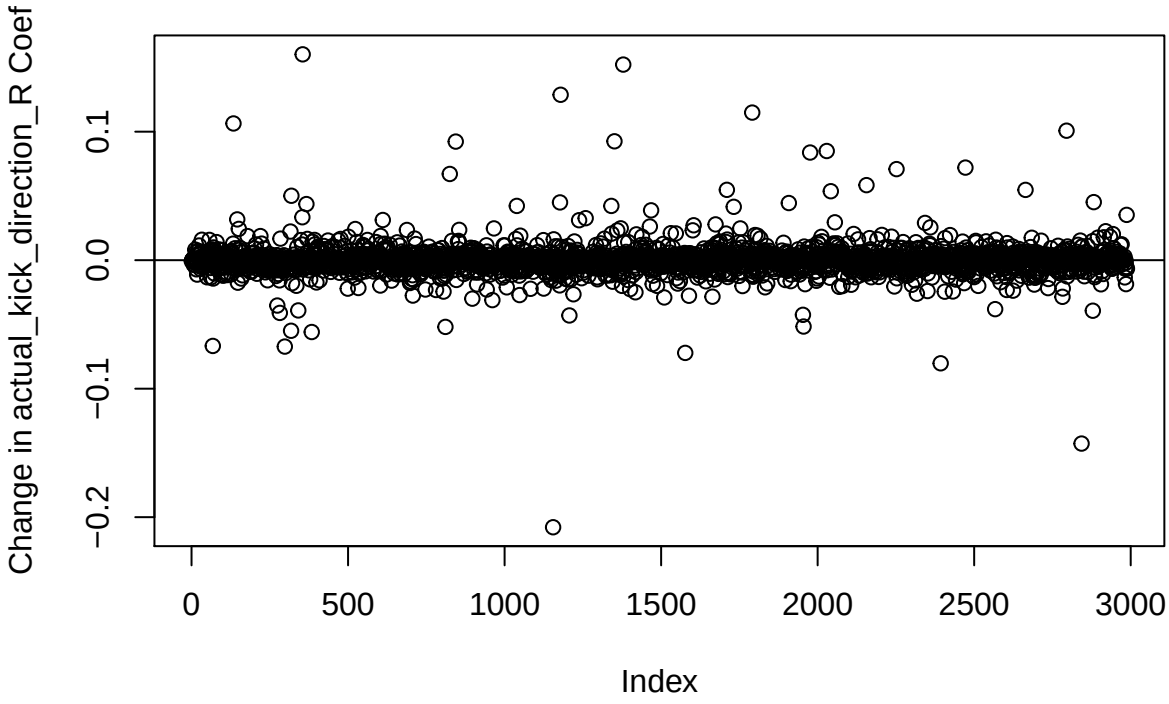

```
plot(dfbeta(lmod)[,2],ylab="Change in clean_catch_binary Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,2]
influential[which.max(abs(influential))]
```

```
##       1979
## -0.3082643
```

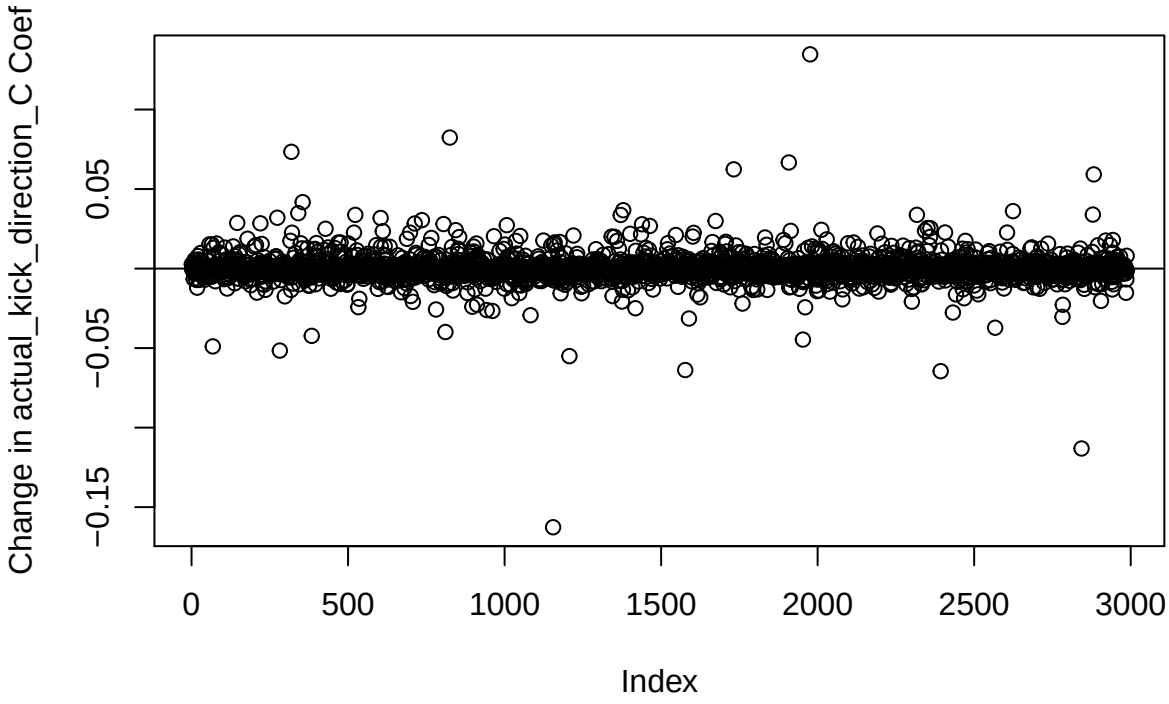
```
plot(dfbeta(lmod)[,3],ylab="Change in actual_kick_direction_R Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,3]
influential[which.max(abs(influential))]
```

```
##       1157
## -0.2078295
```

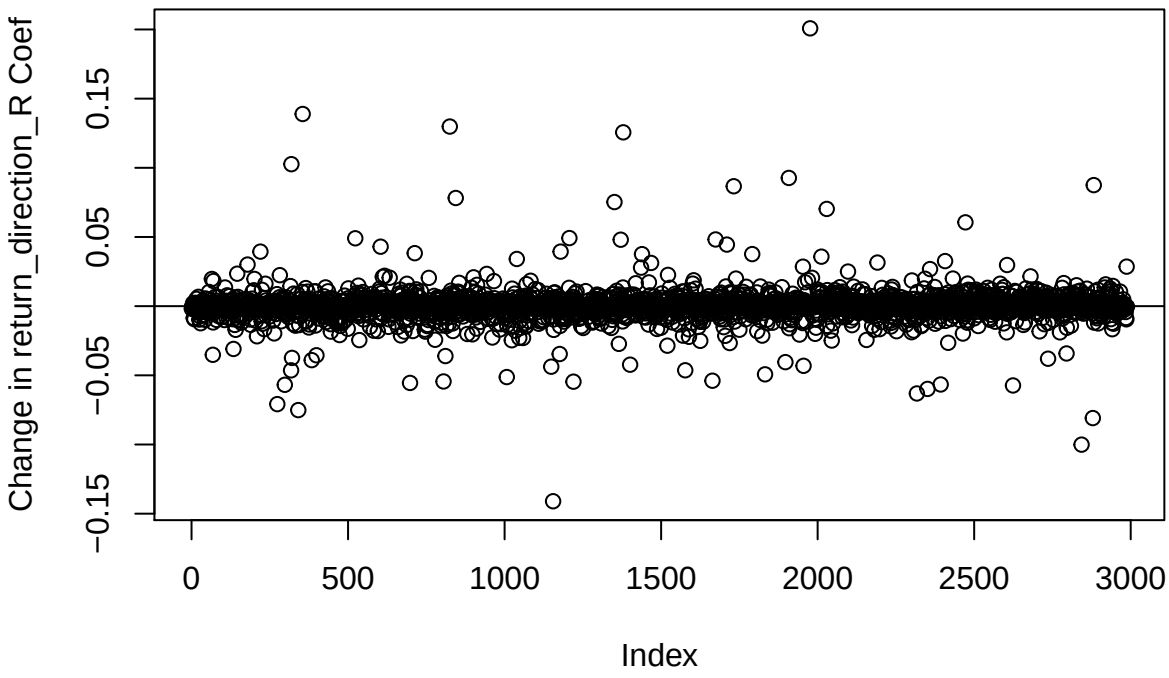
```
plot(dfbeta(lmod)[,4],ylab="Change in actual_kick_direction_C Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,4]
influential[which.max(abs(influential))]
```

```
##       1157
## -0.1626383
```

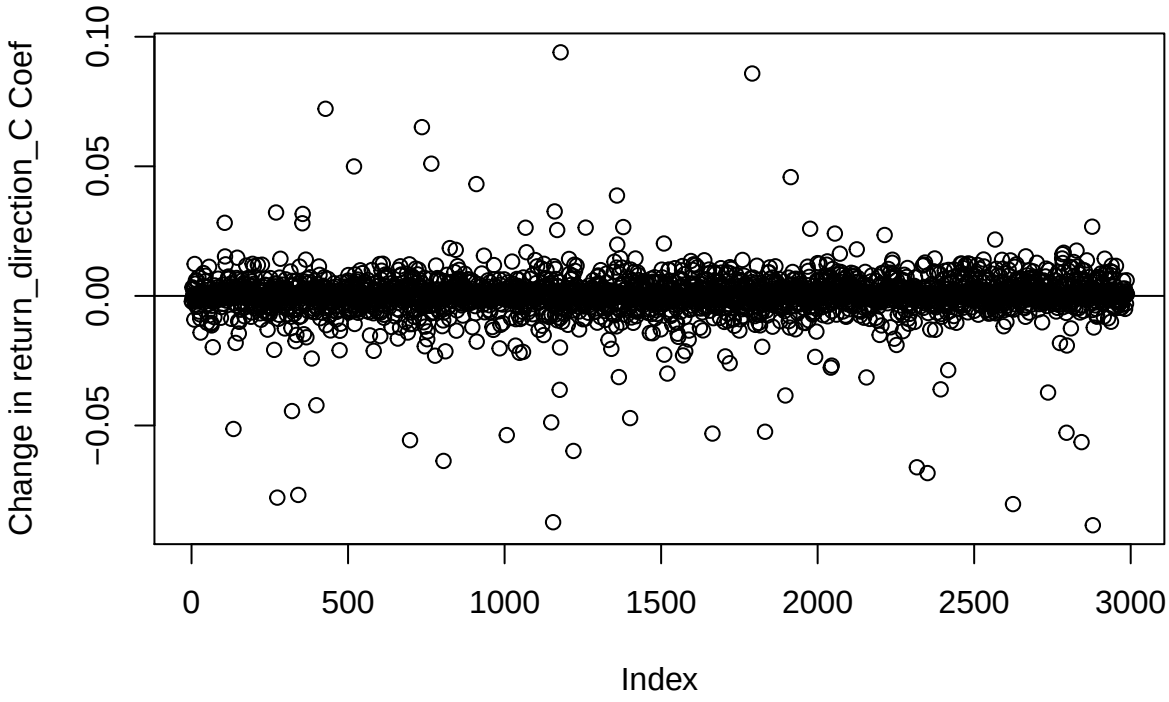
```
plot(dfbeta(lmod)[,5],ylab="Change in return_direction_R Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,5]
influential[which.max(abs(influential))]
```

```
##      1979
## 0.2008027
```

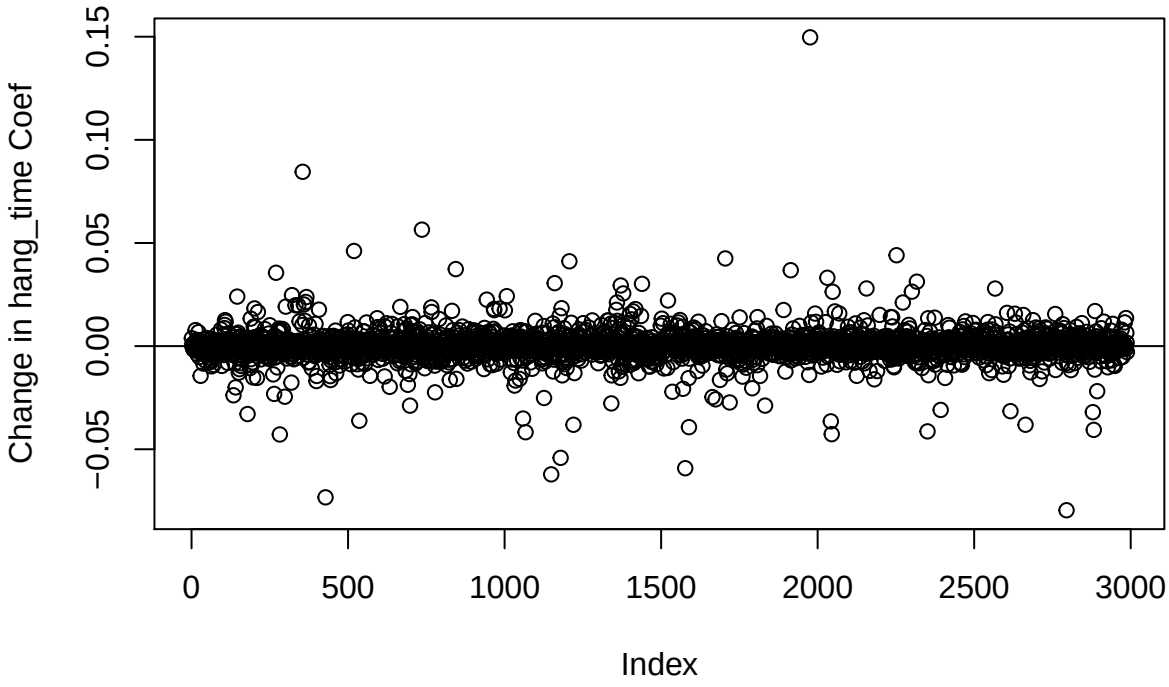
```
plot(dfbeta(lmod)[,6],ylab="Change in return_direction_C Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,6]
influential[which.max(abs(influential))]
```

```
##       1181
## 0.09397576
```

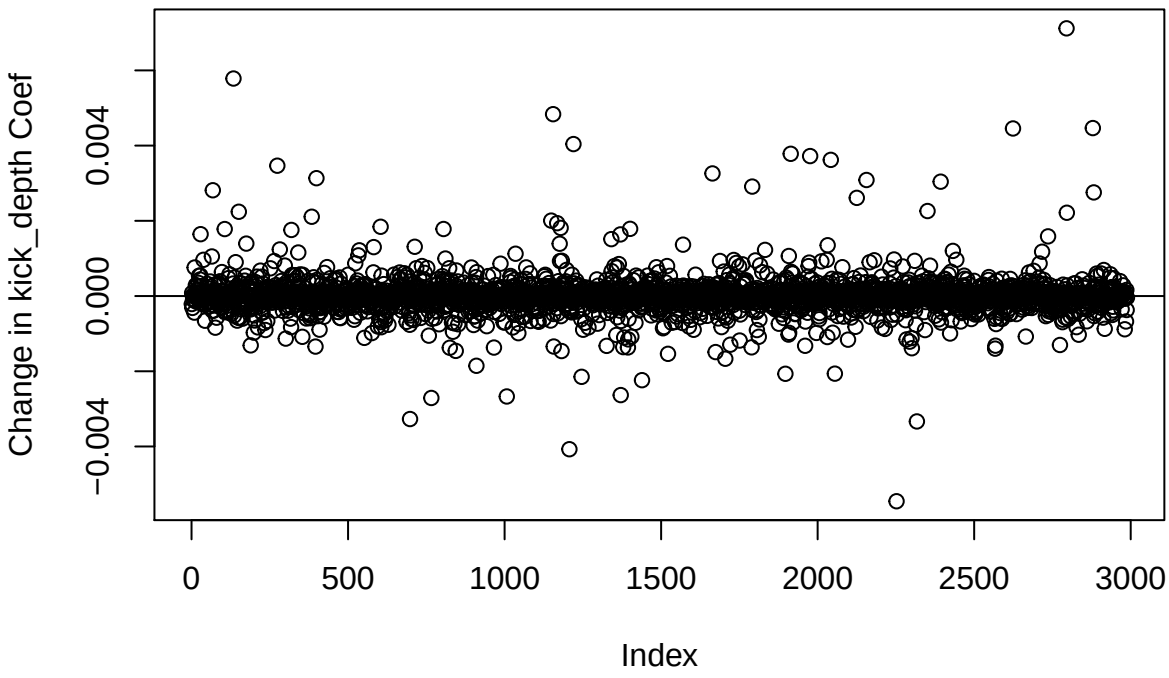
```
plot(dfbeta(lmod)[,7],ylab="Change in hang_time Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,7]
influential[which.max(abs(influential))]
```

```
##      1979
## 0.1496918
```

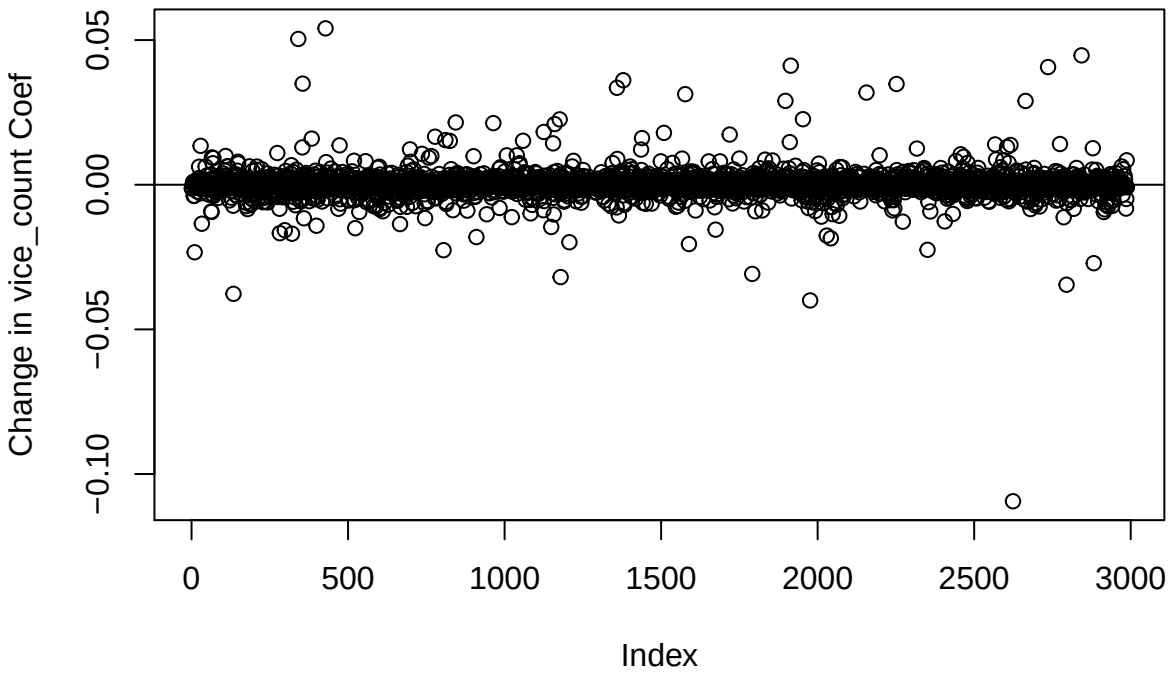
```
plot(dfbeta(lmod)[,8],ylab="Change in kick_depth Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,8]
influential[which.max(abs(influential))]
```

```
##        2798
## 0.007117713
```

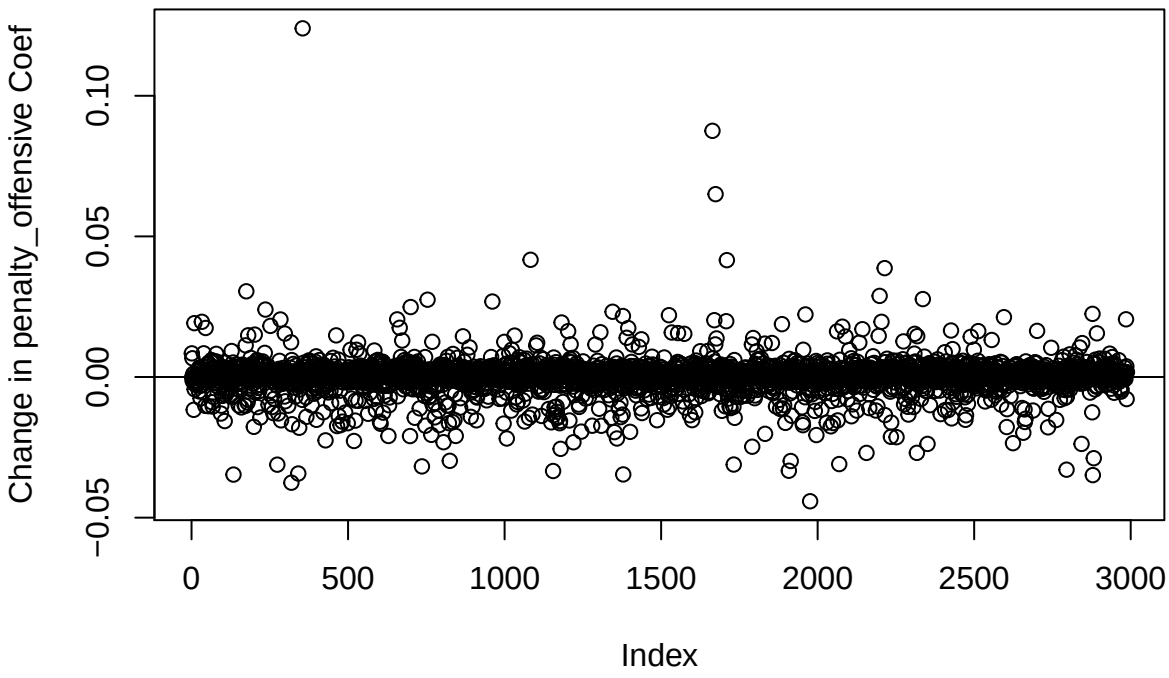
```
plot(dfbeta(lmod)[,9],ylab="Change in vice_count Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,9]
influential[which.max(abs(influential))]
```

```
##      2627
## -0.109407
```

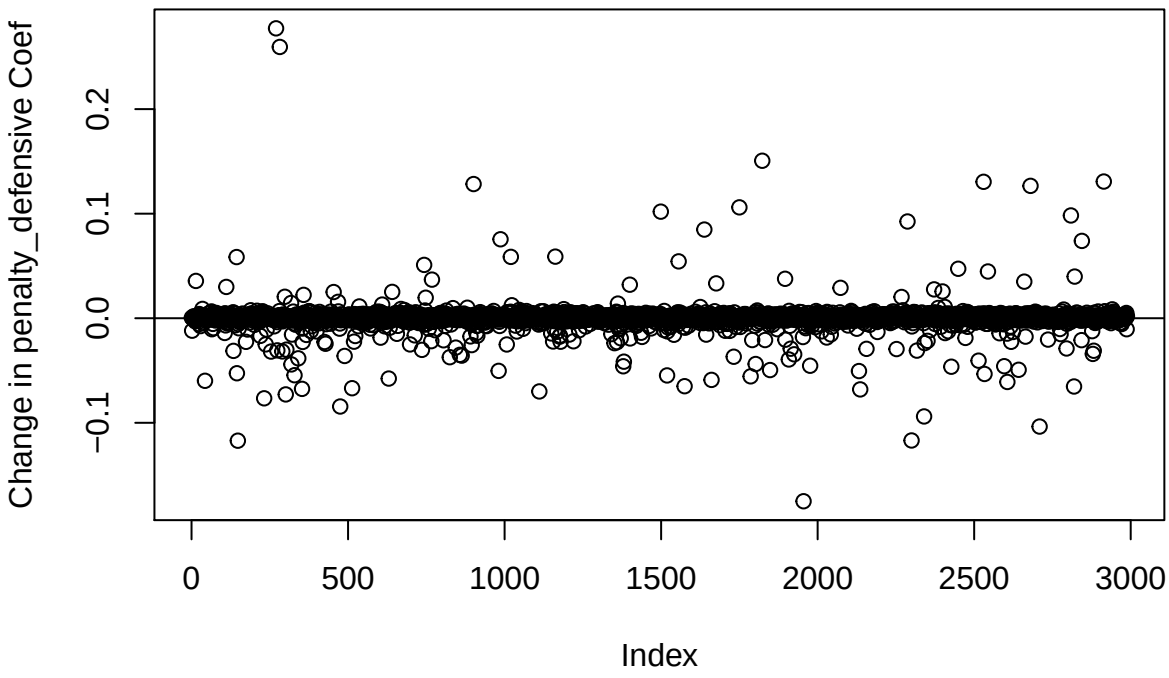
```
plot(dfbeta(lmod)[,10],ylab="Change in penalty_offensive Coef")
abline(h=0)
```




```
influential <- dfbeta(lmod)[,10]
influential[which.max(abs(influential))]
```

```
##       355
## 0.1239788
```

```
plot(dfbeta(lmod)[,11],ylab="Change in penalty_defensive Coef")
abline(h=0)
```




```
influential <- sort(dfbeta(lmod)[,11], decreasing = F)
tail(influential, 1)
```

```
##       270
## 0.2772372
```

```
head(tail(influential, 2),1)
```

```
##       282
## 0.2594507
```

```
finalLmodAllu<-lm(return_yards ~ clean_catch_binary + actual_kick_direction_R + actual_kick_direction_C
summary(finalLmodAllu)
```

```
##
## Call:
## lm(formula = return_yards ~ clean_catch_binary + actual_kick_direction_R +
##     actual_kick_direction_C + return_direction_R + return_direction_C +
##     hang_time + kick_depth + vise_count + penalty_offensive +
##     penalty_defensive, data = puntOrig, subset = (row.names(puntOrig) !=
##     "1979" & row.names(puntOrig) != "2627" & row.names(puntOrig) !=
##     "355" & row.names(puntOrig) != "270" & row.names(puntOrig) !=
##     "282"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -25.686  -5.130  -1.407   2.753  82.379
##
```

```
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               0.69396    1.80906   0.384  0.70130
## clean_catch_binary        4.34653    0.52986   8.203 3.45e-16 ***
## actual_kick_direction_R   2.31165    0.58619   3.943 8.22e-05 ***
## actual_kick_direction_C   2.77335    0.46994   5.902 4.01e-09 ***
## return_direction_R        3.02578    0.48370   6.255 4.53e-10 ***
## return_direction_C       -0.43107    0.42448  -1.016  0.30994
## hang_time                -3.85873    0.44149  -8.740  < 2e-16 ***
## kick_depth                0.33565    0.02757  12.176  < 2e-16 ***
## vise_count                0.74563    0.22920   3.253  0.00115 **
## penalty_offensive        -5.45817    0.47081 -11.593  < 2e-16 ***
## penalty_defensive        -2.49719    0.89197  -2.800  0.00515 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.426 on 2972 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.1468, Adjusted R-squared:  0.1439
## F-statistic: 51.14 on 10 and 2972 DF,  p-value: < 2.2e-16
```

After the tests and removing outliars/influential points we are left with the model above.

```
min(puntOrig$return_yards)
```

```
## [1] -14
```

```
finalLmodAllu<-lm(1/(return_yards-(-1+min(puntOrig$return_yards))) ~ clean_catch_binary + actual_kick_di
summary(finalLmodAllu)
```

```
##
## Call:
## lm(formula = 1/(return_yards - (-1 + min(puntOrig$return_yards))) ~
##     clean_catch_binary + actual_kick_direction_R + actual_kick_direction_C +
##         return_direction_R + return_direction_C + hang_time +
##         kick_depth + vise_count + penalty_offensive + penalty_defensive,
##     data = puntOrig, subset = (row.names(puntOrig) != "1979" &
##         row.names(puntOrig) != "2627" & row.names(puntOrig) !=
##         "355" & row.names(puntOrig) != "270" & row.names(puntOrig) !=
##         "282" & row.names(puntOrig) != "298"), na.action = na.exclude)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.041634 -0.008883 -0.000781  0.007967  0.192694
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               5.993e-02  2.866e-03  20.912  < 2e-16 ***
## clean_catch_binary       -9.616e-03  8.391e-04 -11.460  < 2e-16 ***
## actual_kick_direction_R -4.011e-03  9.291e-04  -4.317 1.63e-05 ***
## actual_kick_direction_C -4.199e-03  7.443e-04  -5.641 1.85e-08 ***
## return_direction_R       -8.033e-03  7.667e-04 -10.478  < 2e-16 ***
## return_direction_C       -2.036e-03  6.722e-04  -3.028  0.00248 **
## hang_time                 7.533e-03  6.993e-04  10.772  < 2e-16 ***
## kick_depth               -6.011e-04  4.366e-05 -13.769  < 2e-16 ***
## vise_count               -3.871e-04  3.631e-04  -1.066  0.28648
```

```
## penalty_offensive        9.621e-03  7.456e-04  12.903  < 2e-16 ***
## penalty_defensive        3.857e-03  1.413e-03   2.730  0.00637 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01493 on 2971 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.2072, Adjusted R-squared:  0.2046
## F-statistic: 77.66 on 10 and 2971 DF,  p-value: < 2.2e-16
```

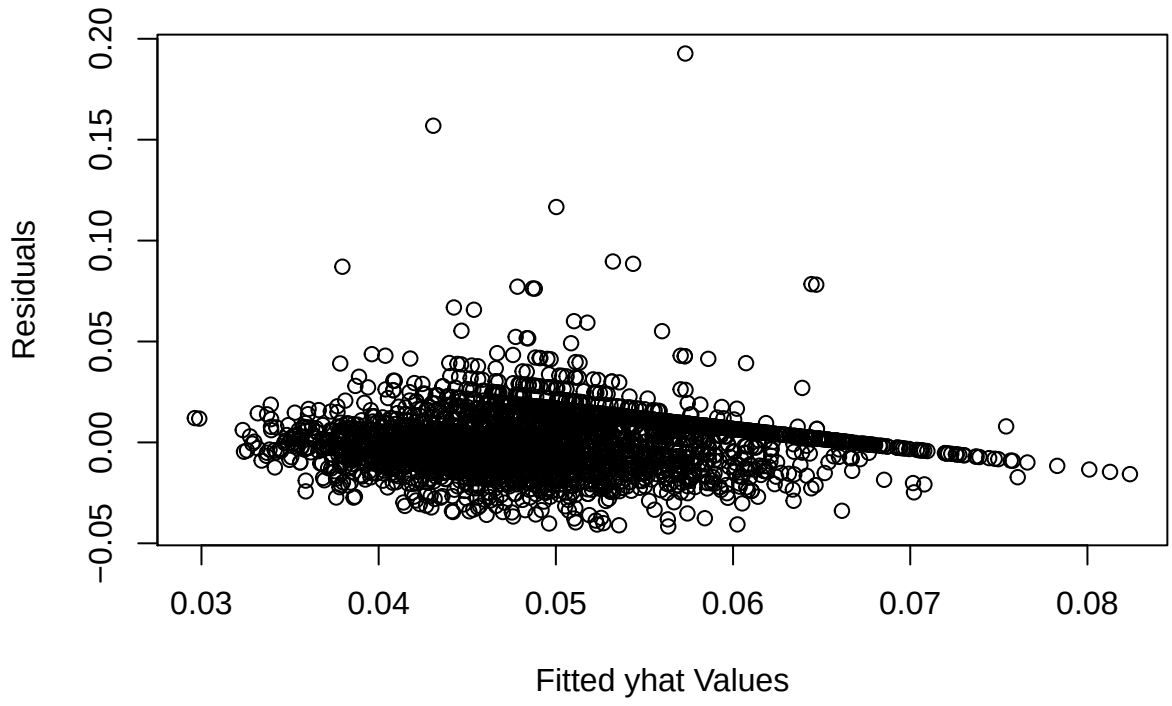
```
lmodu<-finalLmodAllu
```

We then perform a transformation similar to the ones we had already done. With the new tranformation we check out assumptions again, just as we had done previously.

```
residuals(lmodu)[which.max(abs(residuals(lmodu)))]
```

```
##      2072
## 0.1926944
```

```
plot(fitted(lmodu), residuals(lmodu), xlab = "Fitted yhat Values", ylab = "Residuals")
```

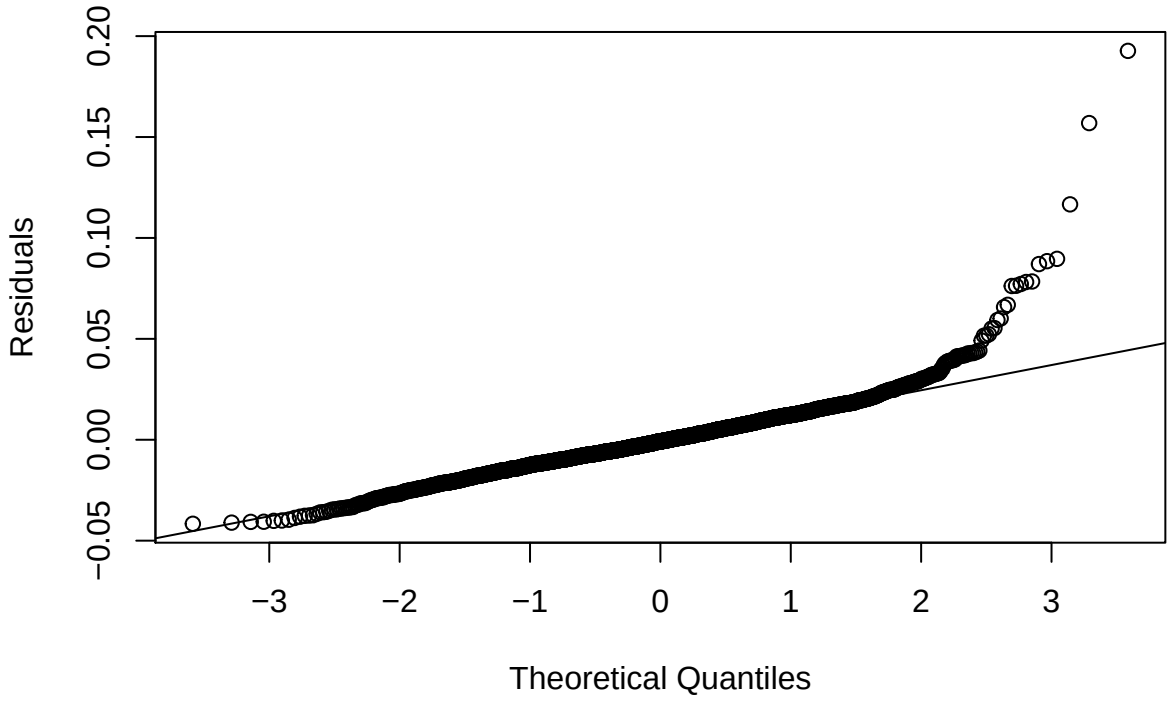



```
var.test(residuals(lmodu)[fitted(lmod)>0.0475], residuals(lmodu)[fitted(lmod) < 0.0475])
```

```
##
##  F test to compare two variances
##
## data:  residuals(lmodu)[fitted(lmod) > 0.0475] and residuals(lmodu)[fitted(lmod) < 0.0475]
## F = 1.6509, num df = 2885, denom df = 95, p-value = 0.001914
```

```
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.210307 2.163795
## sample estimates:
## ratio of variances
##           1.650854
```

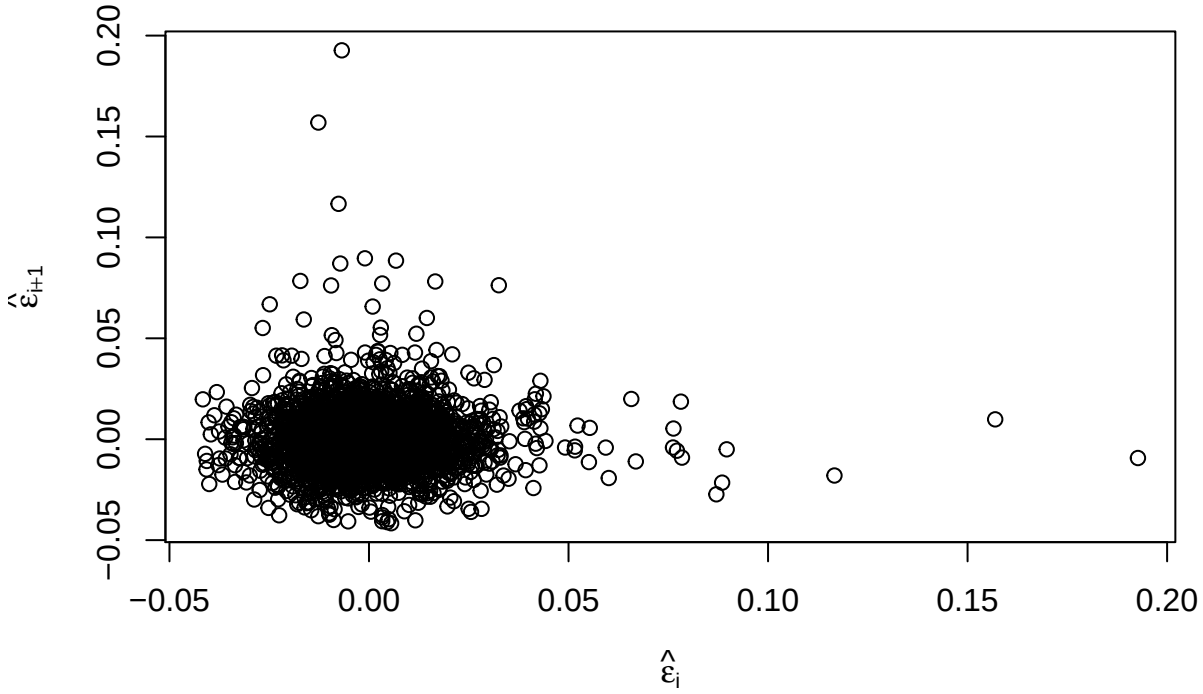
```
qqnorm(residuals(lmodu), ylab="Residuals", main="")
qqline(residuals(lmodu))
```




```
shapiro.test(residuals(lmodu))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(lmodu)
## W = 0.90559, p-value < 2.2e-16
```

```
n<-length(residuals(lmodu))
plot(tail(residuals(lmodu), n-1) ~ head(residuals(lmodu),n-1), xlab = expression(hat(epsilon)[i]), ylab
```



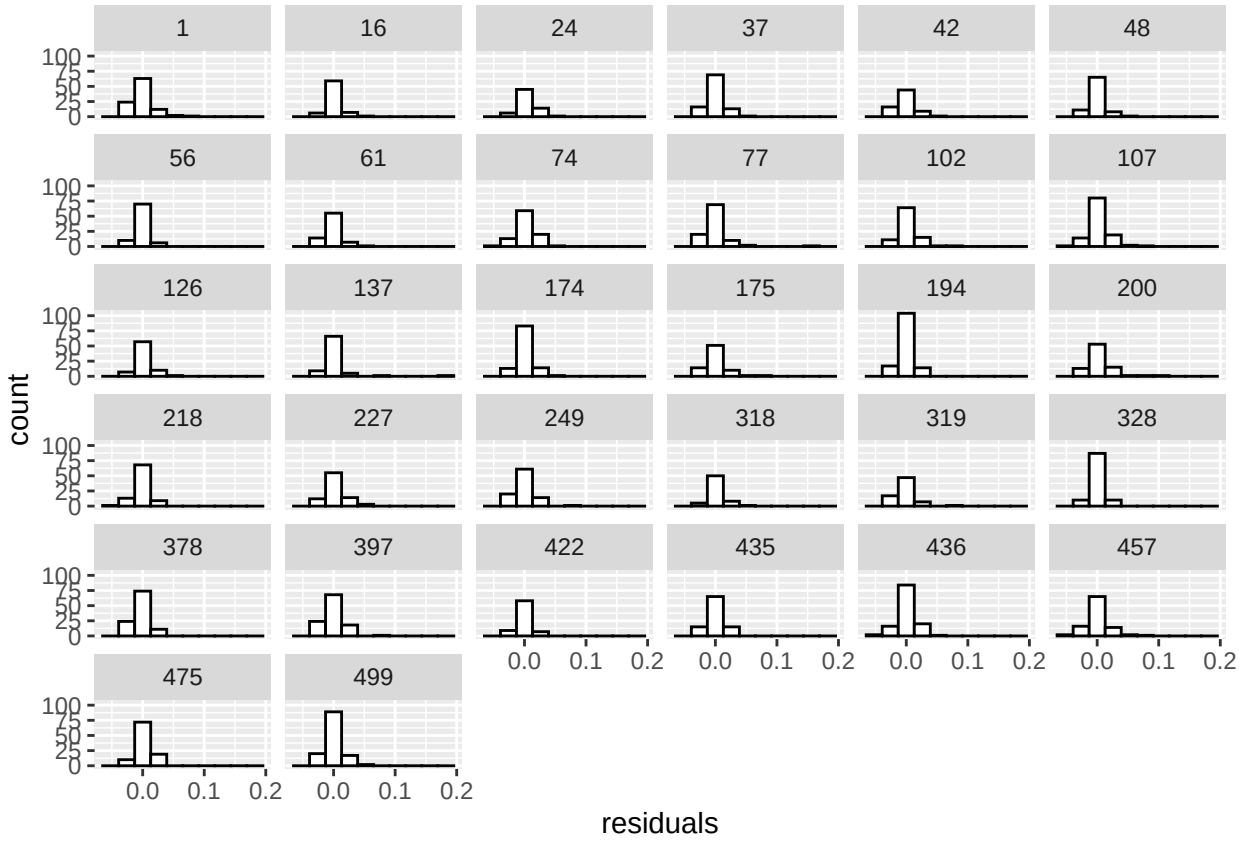

```
library(lmtest)
dwtest((1/(return_yards-(-1+min(puntOrig$return_yards)))) ~ clean_catch_binary + actual_kick_direction_l
```

```
##
##  Durbin-Watson test
##
## data:  (1/(return_yards - (-1 + min(puntOrig$return_yards)))) ~ clean_catch_binary +     actual_kick_
## DW = 1.9804, p-value = 0.2942
## alternative hypothesis: true autocorrelation is greater than 0
```

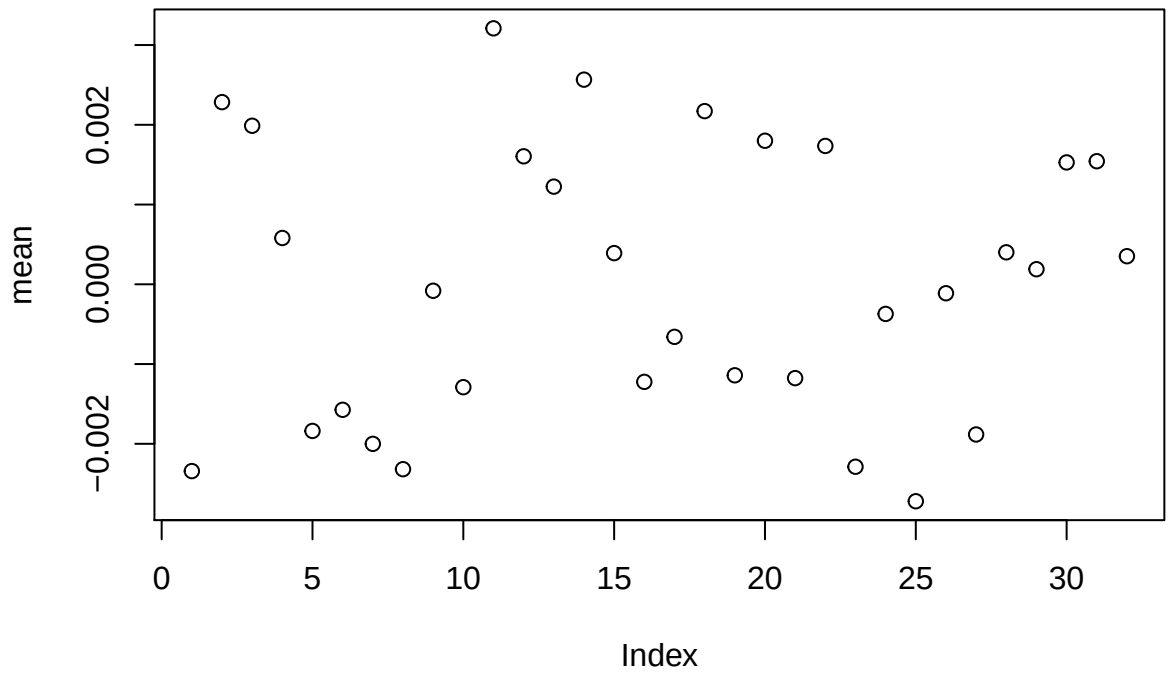
```
puntOrig <- puntOrig[-c(1979, 2627, 355, 270, 282, 298) ,]
puntOrig$residuals <- residuals(lmodu)
unique<-unique(puntOrig$offense_team_id)
puntOrig<-subset(puntOrig, (!is.na(residuals)) & (!is.na(defense_team_id)))
```

We now look at the same sets of histograms but this time with the combined penalty/no penalty model and reach the same conclusions. No team performs significantly better or worse then others on punt returns in the NFL.

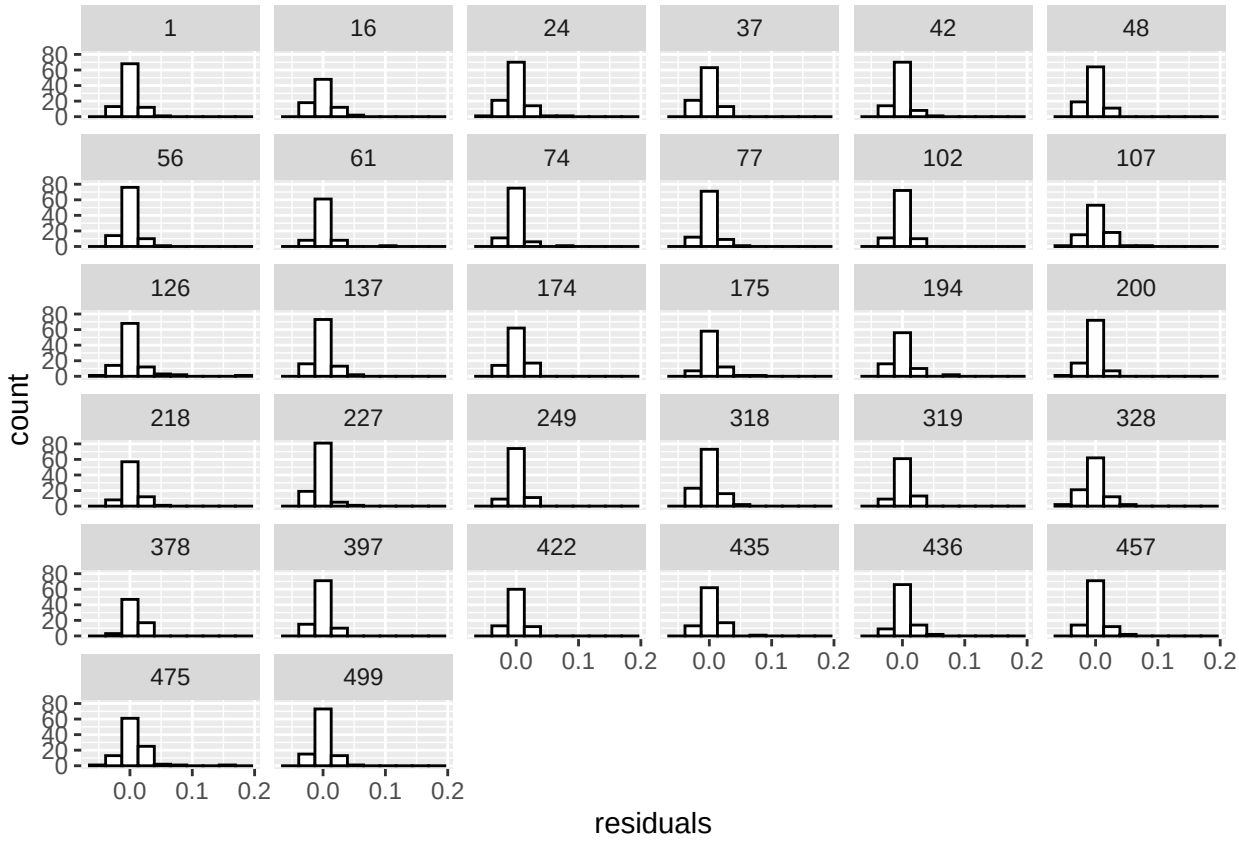
```
ggplot(puntOrig, aes(x = residuals)) +
  geom_histogram(fill = "white", colour = "black", bins = 10) +
  facet_wrap(offense_team_id ~ .)
```



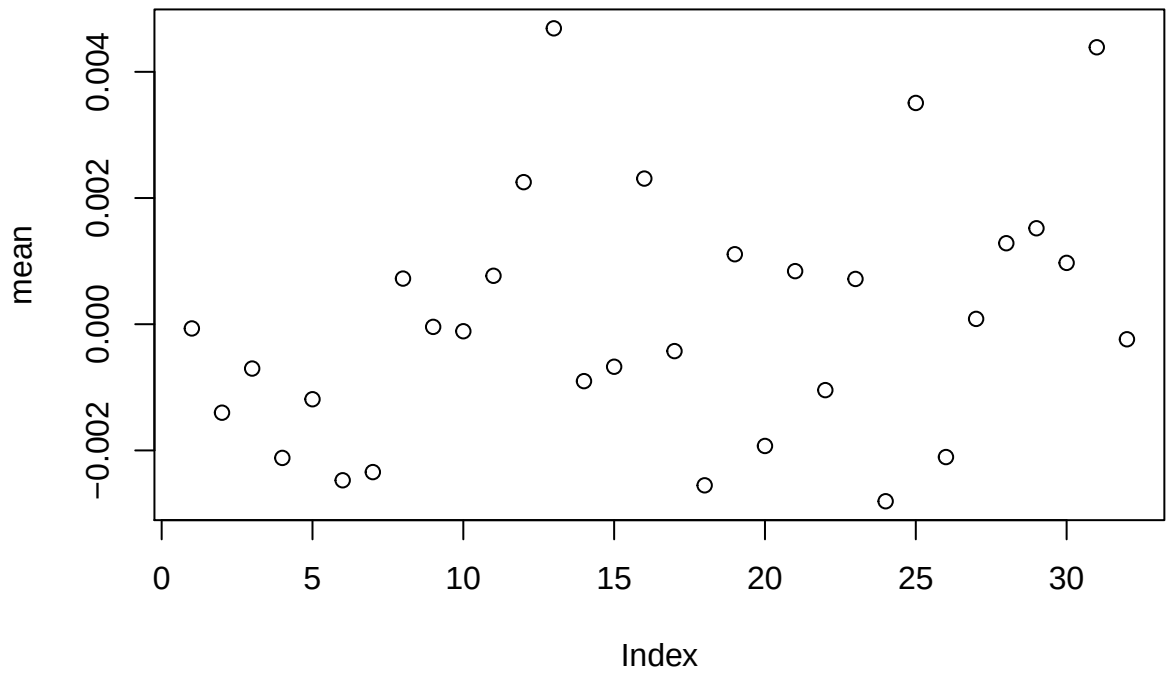

```
mean <- tapply(puntOrig$residuals, puntOrig$offense_team_id, mean)
plot(mean)
```




```
ggplot(puntOrig, aes(x = residuals)) +
  geom_histogram(fill = "white", colour = "black", bins = 10) +
  facet_wrap(defense_team_id ~ .)
```




```
mean <- tapply(puntOrig$residuals, puntOrig$defense_team_id, mean)
plot(mean)
```


mean
−0.002
0.000
0.002
0.004
0
5
10
15
20
25
30
Index