Predicting 40-Yard Dash Times of NFL Prospects

Hans Larson

```r
# Set global figure size
knitr::opts_chunk$set(fig.width=6, fig.height=3.5)
```

Abstract

This data comes from the NFL Combine between the years 2012 and 2017. It contains data from the physical tests that NFL prospects took to catch the eye of possible NFL draft recruiters.My primary goal is to see what predictor variables have the highest correlation with a player's 40-yard-dash time. The initial predictor variables will be weight (lbs), and physical tests including vertical leap, broad jump, shuttle and 3-cone-drill agility tests, and number of bench reps at 225 pounds. Quarterbacks were excluded from the dataset because of their inconsistent data, and I deleted any players that had N/As for any of the predictor variables. There are 627 observations in this data set.

Introduction

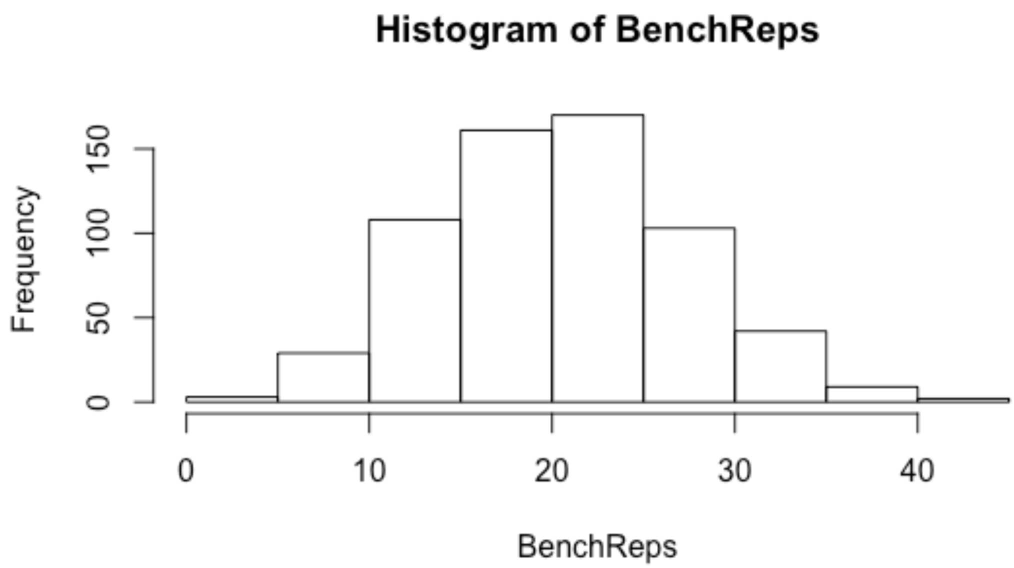
There are many factors that come into play for an NFL player's 40 yard dash time. It would be interesting to know which physical tests/attributes play the largerst role in this prediction because of the importance of the 40-yard dash time as an indicator of a player's overall physical ability. I analyzed combine data from 627 NFL players from 2012 to 2017 of all positions. Using multiple linear regression analysis, I examined the effects of 5 different predictor variables in my final model on 40 yard dash, including Weight, BenchReps, Broad Jump, 3 Cone drill, and Position.

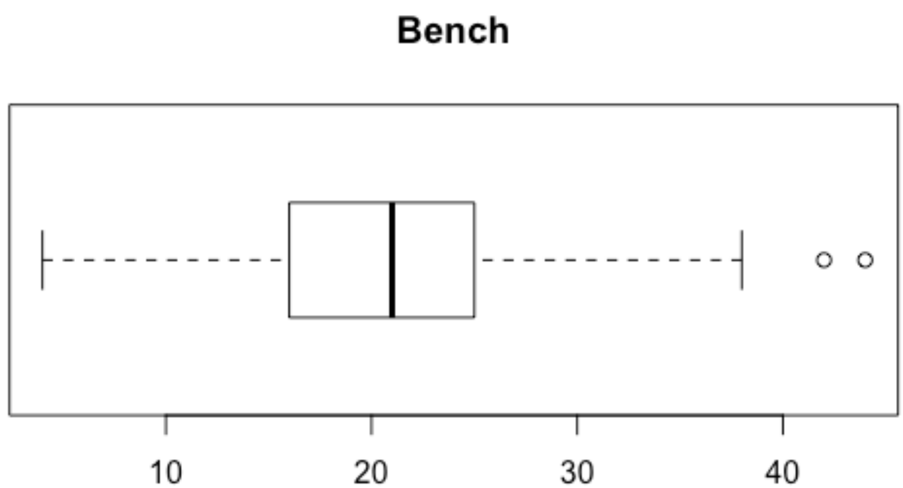
I found that among this set of players, a lower 40 yard dash time is associated with players that have a lower weight, a high vertical leap, a far broad jump, and low 3-cone and shuttle times. A number of the predictor variables were correlated with each other. For example, all the players that had a fast shuttle also had a fast 3-cone drill time. Similarly, players with a high vertical leap also had a long broad jump. These are only 2 of the many correlated predictor variables that were found within the data.I also found 3 significant interaction effects. These included the negative relationship of weight and broad jump, the negative relationship of broad jump and 3 cone drill, and 3-cone drill and position category. In the significant interaction plots, I discovered that the faster, more agile positions' 40-yard dash times are less affected by their three-cone drill times. I also discovered that the longer the broad jump, the less a player's 40-yard dash time is affected by weight. Finally, I discovered that three-cone drill's effect on 40-yard dash is consistent with each broad jump category, except for the farthest broad jumpers, who's 40-yard dash times are less affected by three-cone drill.

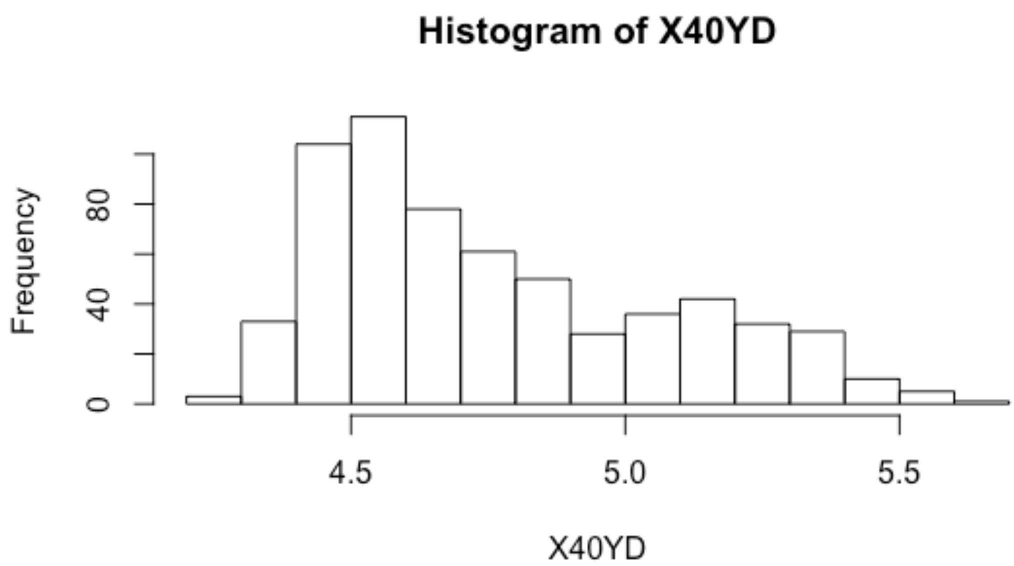
My prediction model predicted 95% of the variables within the upper and lower bounds

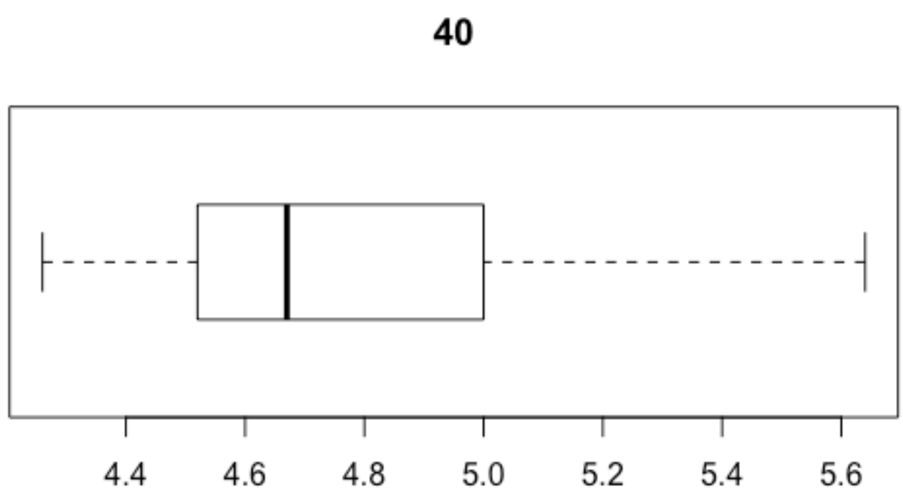
Exploratory Analysis

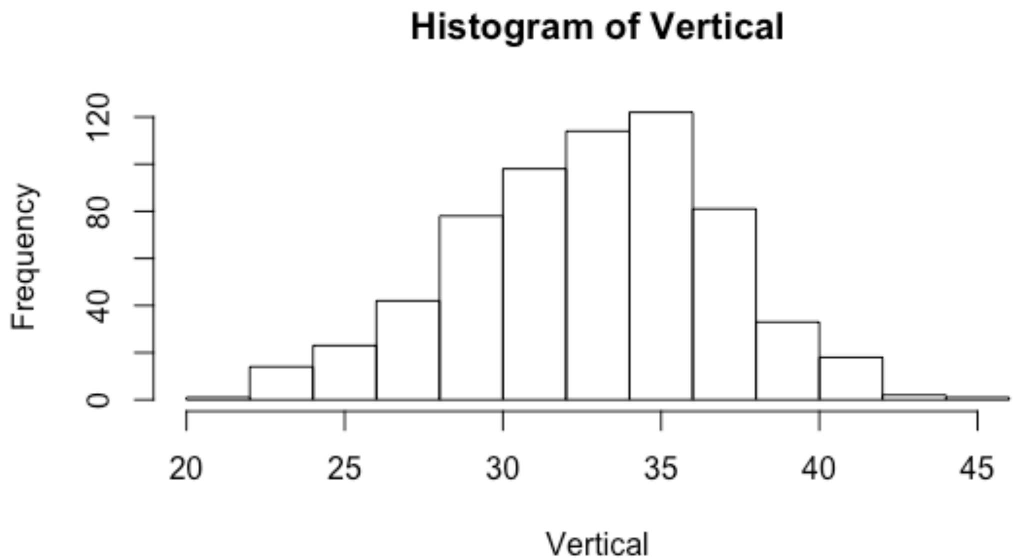
```
## corrplot 0.84 loaded
```

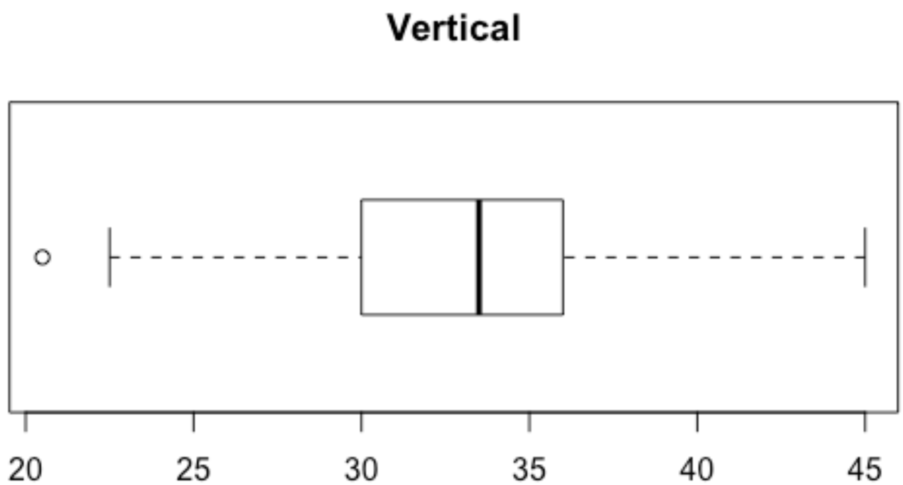


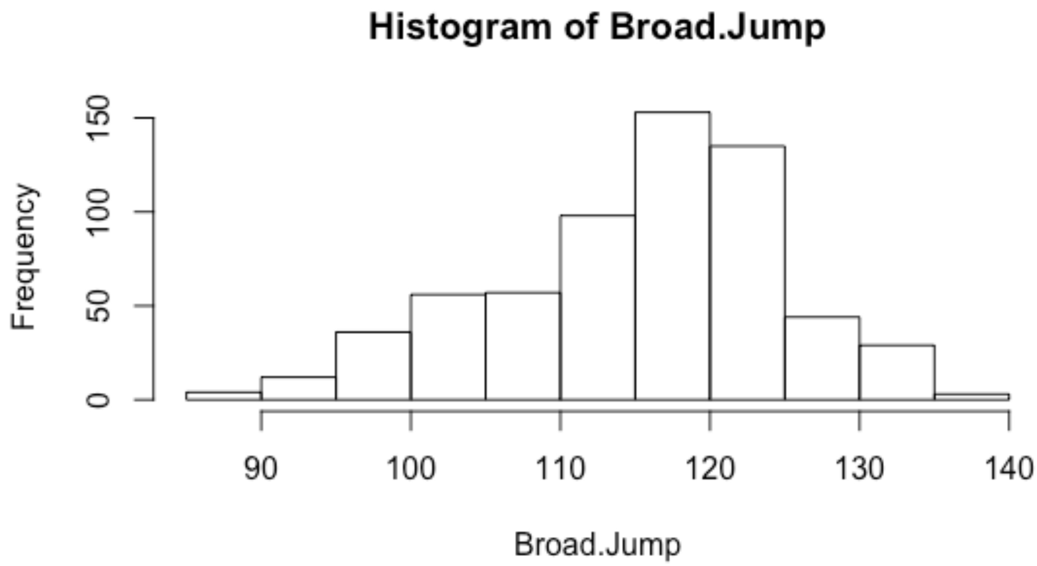


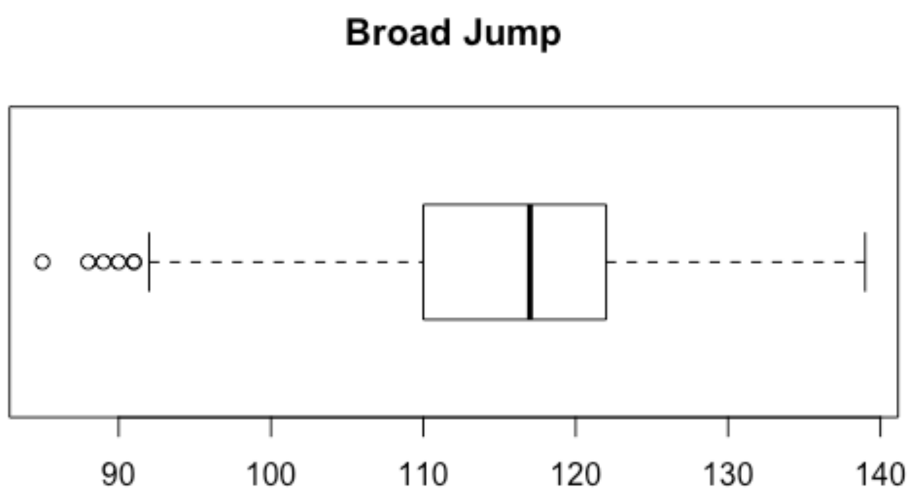

Histogram of X40YD
Frequency
0
40
80
4.5
5.0
5.5
X40YD

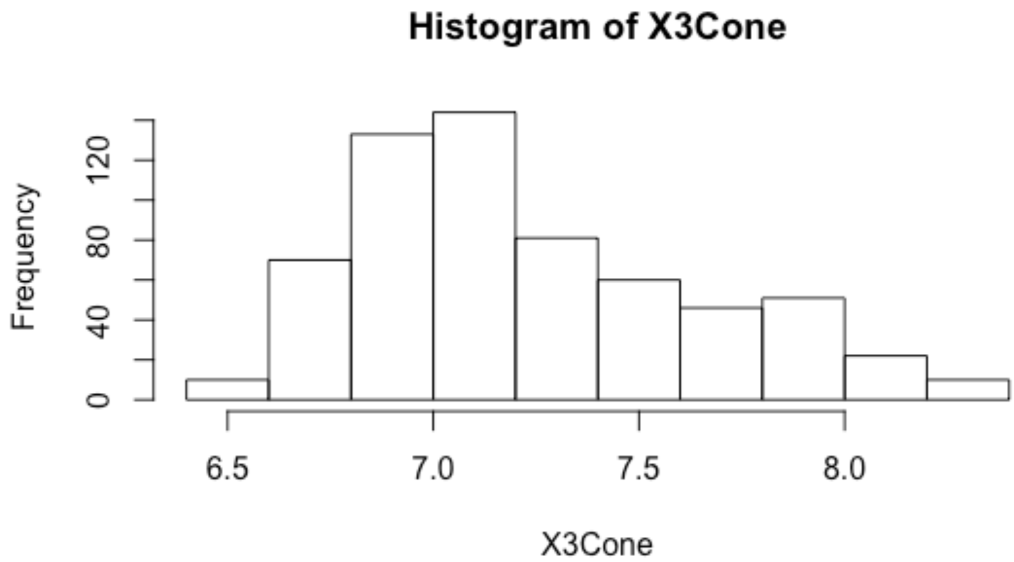

40
4.4
4.6
4.8
5.0
5.2
5.4
5.6

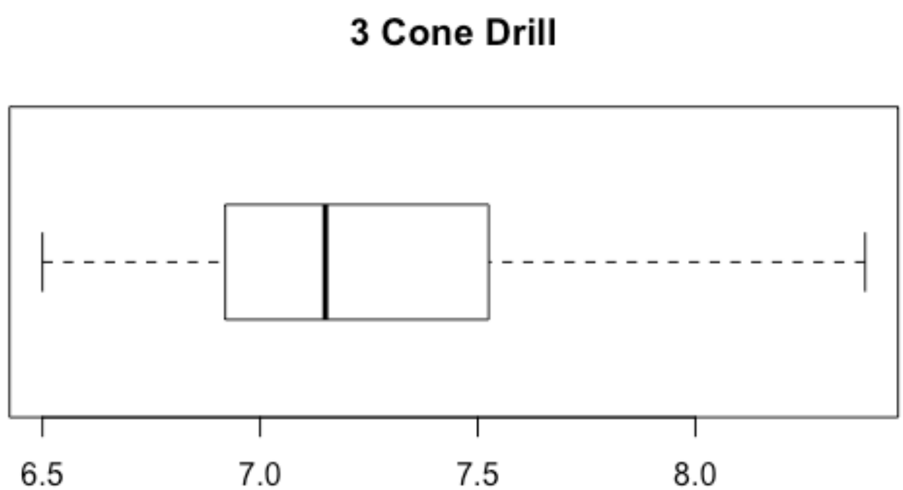

Histogram of Vertical
Frequency
0
40
80
120
20
25
30
35
40
45
Vertical

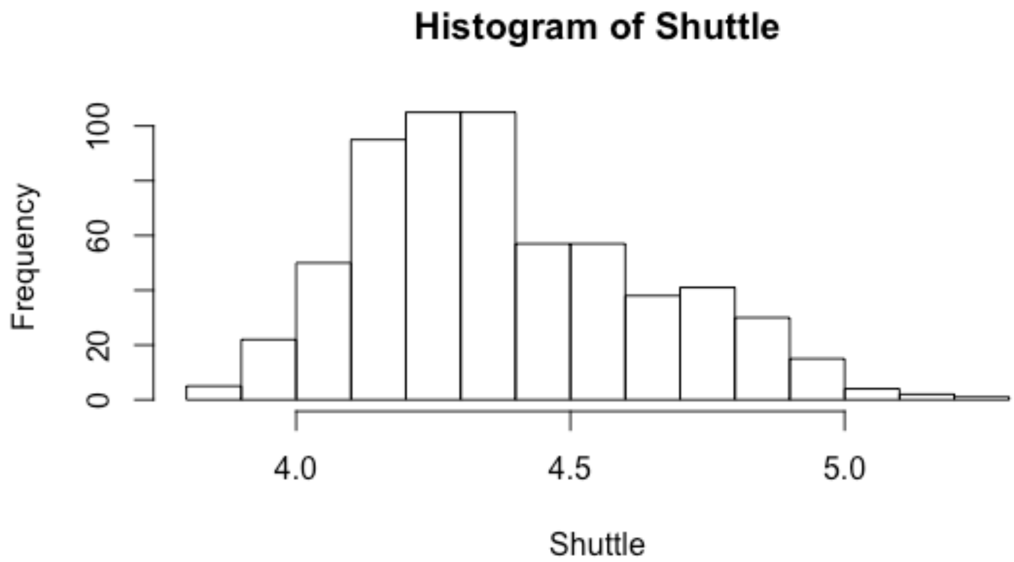

Vertical
20
25
30
35
40
45

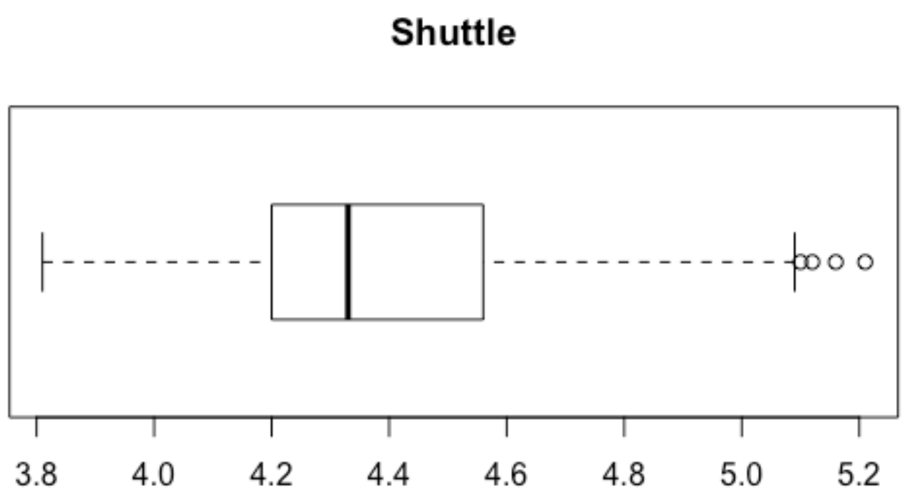

Histogram of Broad.Jump
Frequency
0
50
100
150
90
100
110
120
130
140
Broad.Jump


Broad Jump
90
100
110
120
130
140


Histogram of X3Cone
Frequency
0
40
80
120
6.5
7.0
7.5
8.0
X3Cone


3 Cone Drill
6.5
7.0
7.5
8.0





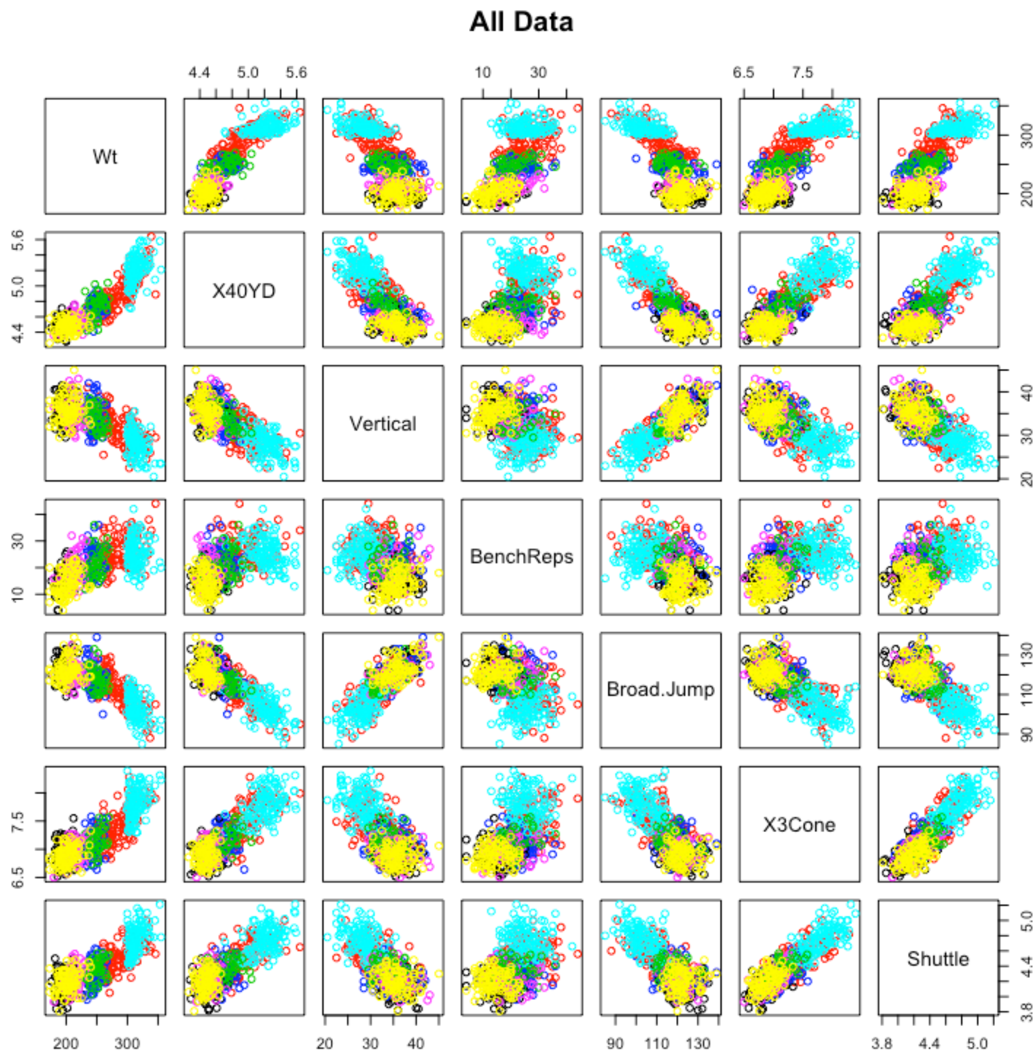
After analyzing histograms of our x variables, I came to the conclusion that the variables are normaly distributed and do not have any extreme skewness. Any skewness observed is due to the fact that I have more fast players like D-backs in the data, and less slow players like linemen, because D-backs were more likely to have data for all of the tests than linemen. These variables do not require any transformations.

Categorical Data

I have no categorical variables in the scatterplot matrix, but I now know that positions could be a categorical variable. I will add in positions in groups (i.e. CBs, FSs, and SSs will be put in the group DBs).

Scatterplot Matrix

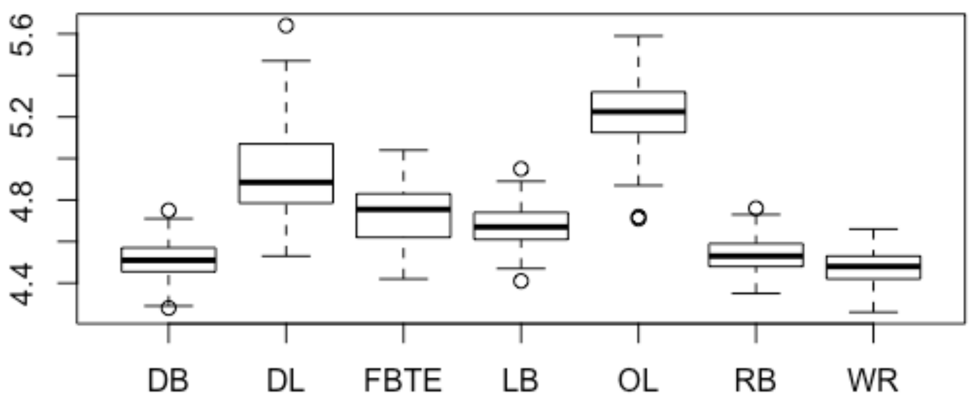
```r
all$pos.cat=as.factor(pos.cat)
pairs (all[,c(7:13)], main="All Data", col=as.factor(pos.cat))
```



##All positions Scatterplot matrix analysis From this scatterplot matrix, it is clear that when looking at all positions, most of the variables have a strong linear correlation with each other. The variables that do not have strong linear correlation appear to be any variable that correlates to Bench Reps, with the exception of weight. This is not entirely surprising, but it is somewhat unexpected. Variables that have the strongest linear correlation appear to be 3 cone drill and shuttle, and 40 yard dash and Broad Jump. Weight and 40 yard dash are also very correlated, which is expected, but it is not a linear correlation, although there is constant variance. Position groups also have very distinct locations in the scatterplots.

Black = DB, Red = DL, Green = FBTE, Blue = LB, LightBlue = OL, Purple = RB, Yellow = WR ##Box Plot by Position

```
plot(as.factor(pos.cat), X40YD)
```



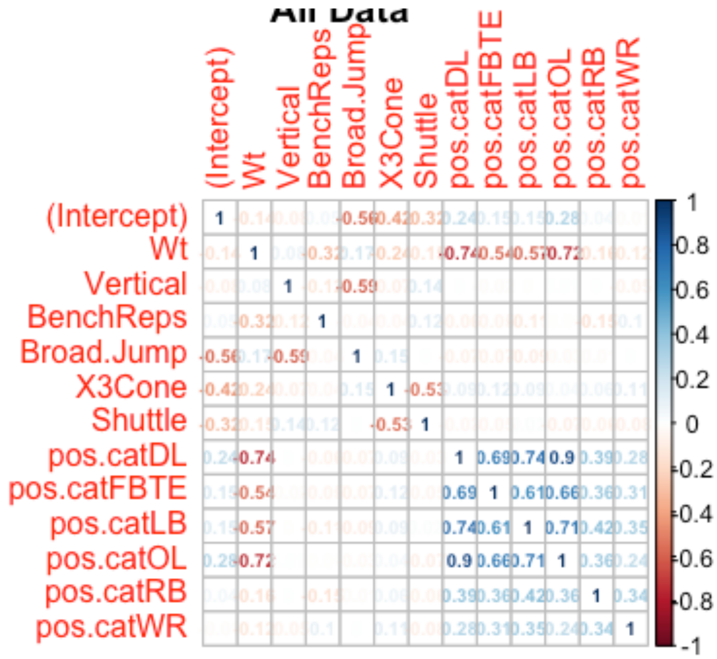
Here we can clearly see the quartiles of each position's 40-yard dash time.

```
#First-order model for complete data set
fit1 = lm(X40YD ~ Wt + Vertical + BenchReps + Broad.Jump + X3Cone + Shuttle +
pos.cat)
summary(fit1)

##
## Call:
## lm(formula = X40YD ~ Wt + Vertical + BenchReps + Broad.Jump +
##     X3Cone + Shuttle + pos.cat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35515 -0.05961  0.00155  0.06243  0.36772
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.2460549  0.1922023  22.092  < 2e-16 ***
## Wt           0.0035313  0.0003461  10.203  < 2e-16 ***
## Vertical    -0.0013759  0.0018658  -0.737 0.461144
## BenchReps   -0.0041684  0.0008972  -4.646 4.14e-06 ***
```

```
## Broad.Jump  -0.0083942  0.0008988  -9.340  < 2e-16 ***
## X3Cone        0.0898675  0.0238364   3.770 0.000179 ***
## Shuttle       0.0191337  0.0355028   0.539 0.590127
## pos.catDL     0.0032358  0.0281569   0.115 0.908546
## pos.catFBTE -0.0008956  0.0234698  -0.038 0.969573
## pos.catLB   -0.0094577  0.0196528  -0.481 0.630517
## pos.catOL     0.0802511  0.0345552   2.322 0.020537 *
## pos.catRB   -0.0339966  0.0180168  -1.887 0.059641 .
## pos.catWR   -0.0502122  0.0146411  -3.430 0.000645 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1018 on 614 degrees of freedom
## Multiple R-squared:  0.8925, Adjusted R-squared:  0.8904
## F-statistic: 424.8 on 12 and 614 DF,  p-value: < 2.2e-16
```

Correlation Plot

```
summfit1 = summary(fit1, correlation = T)
corrplot (summfit1$correlation, method = "number", number.cex=0.6, main="All
Data")
```

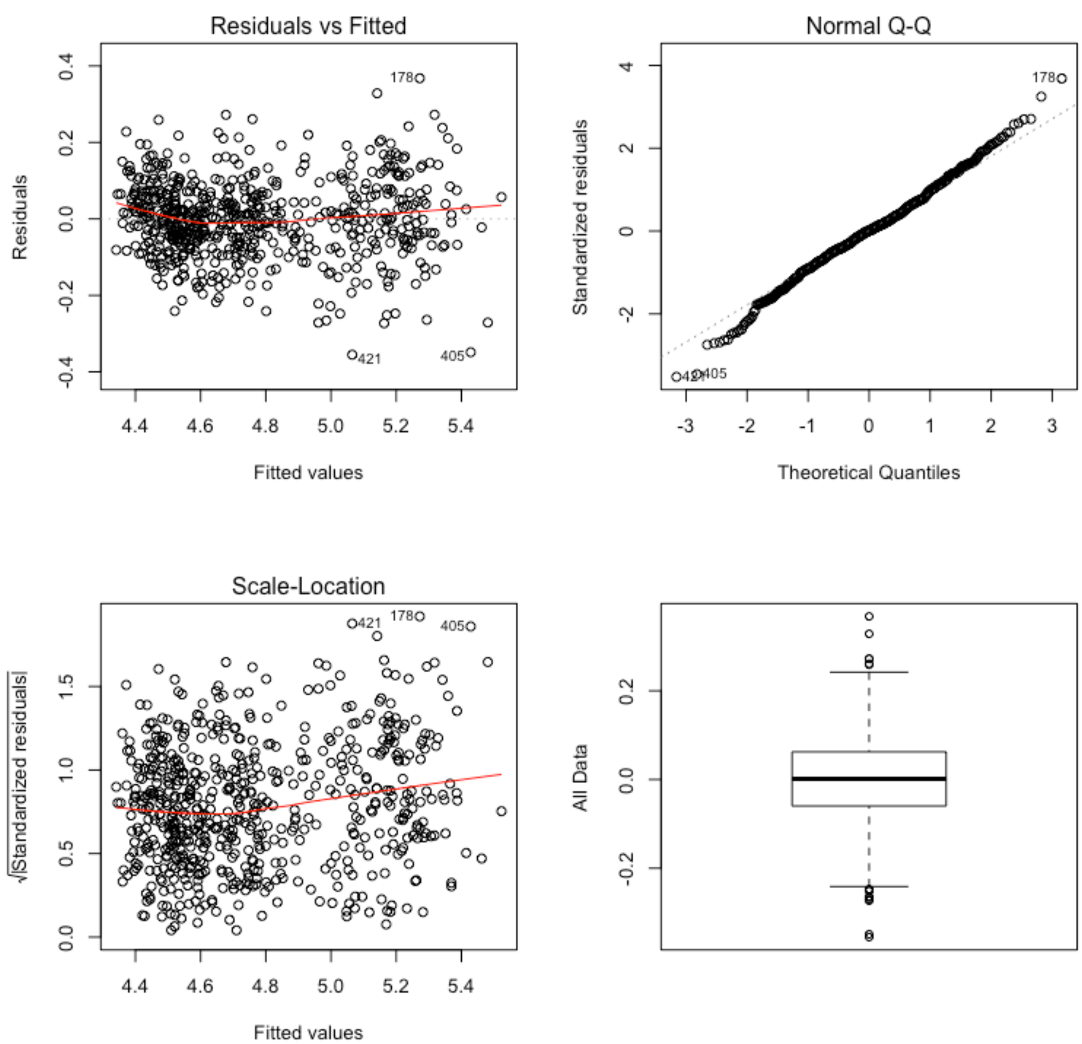


summary, anova, and corrplot analysis for all positions

From the summary and the correlation plot of the data, it can be reasonably concluded that there are many significant predictors, with weight having the most correlation with other predictor variables. Shuttle does not appear to be very significant, which is unexpected, but it is understandable because it is a very short test that has a small range of values. R^2 is 0.8925, Adjusted R^2 is 0.8904

First order model residual Analysis

```r
#add in color for position
par (mfrow=c(2,2))
plot(fit1, which = 1:3)
boxplot(fit1$residuals,ylab="All Data")
```



From the residuals vs. fitted plot it can be observed that the data has constant variance. From the normal Q-Q plot it can be seen that the data is also normally distributed because all of the residuals fall roughly on the line. I did not need to transform the response.

There also appears to be two groups of data that have faster and slower 40-yard dash times, with more data points being in the faster cluster. again, this is due to the fact that we have more fast players in this dataset than slow.

Backward Elimination Method - Manual

Initally I considered taking out bench reps from the model because it appeared to have the least correlation with 40 yard dash when looking at the scatter plot matrix. However, after analysis of the summary of fit1, bench reps proved to be a significant predictor. Instead I chose to take out vertical in our next model, fit2, becuase it is the least significant of all the predictor variables.

```
fit2 = lm(X40YD ~ Wt  + BenchReps + Broad.Jump + X3Cone + Shuttle + pos.cat)
summary(fit2)
```

```
##
## Call:
## lm(formula = X40YD ~ Wt + BenchReps + Broad.Jump + X3Cone + Shuttle +
##     pos.cat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36003 -0.06053  0.00100  0.06105  0.36218
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.2353177  0.1915789  22.107  < 2e-16 ***
## Wt           0.0035513  0.0003449  10.296  < 2e-16 ***
## BenchReps   -0.0042458  0.0008907  -4.767 2.34e-06 ***
## Broad.Jump  -0.0087847  0.0007259 -12.101  < 2e-16 ***
## X3Cone       0.0886182  0.0237673   3.729 0.000210 ***
## Shuttle      0.0228314  0.0351339   0.650 0.516039
## pos.catDL    0.0033005  0.0281463   0.117 0.906690
## pos.catFBTE -0.0011902  0.0234577  -0.051 0.959551
## pos.catLB   -0.0094923  0.0196454  -0.483 0.629138
## pos.catOL    0.0804897  0.0345409   2.330 0.020114 *
## pos.catRB   -0.0339773  0.0180101  -1.887 0.059690 .
## pos.catWR   -0.0508035  0.0146137  -3.476 0.000544 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1018 on 615 degrees of freedom
## Multiple R-squared:  0.8924, Adjusted R-squared:  0.8905
## F-statistic: 463.7 on 11 and 615 DF,  p-value: < 2.2e-16
```

Vertical did not appear to change R-squared by anything significant at all. Of the quantitative variables, Weight, Bench Reps, Broad Jump, and 3-cone drill are very significant. Of the position categorical variables, wide receivers appear to be the only significant predictor of 40-yard dash time. Residual standard error appears to be very minimal.

Wide receivers and running backs are the only significantly faster positions than defensive backs, while offensive line is the most significantly slower position. Bench reps and broad jump both cause a player to be faster as they increase, while weight, 3-cone drill, and shuttle #First step model Analysis

```
step1=step(fit1, direction = "both")

## Start:  AIC=-2851.7
## X40YD ~ Wt + Vertical + BenchReps + Broad.Jump + X3Cone + Shuttle +
##     pos.cat
##
##              Df Sum of Sq    RSS     AIC
## - Shuttle     1   0.00301 6.3712 -2853.4
## - Vertical    1   0.00564 6.3738 -2853.2
## <none>                    6.3681 -2851.7
## - X3Cone      1   0.14742 6.5156 -2839.3
## - BenchReps   1   0.22387 6.5920 -2832.0
## - pos.cat     6   0.41417 6.7823 -2824.2
## - Broad.Jump  1   0.90472 7.2729 -2770.4
## - Wt          1   1.07965 7.4478 -2755.5
##
## Step:  AIC=-2853.41
## X40YD ~ Wt + Vertical + BenchReps + Broad.Jump + X3Cone + pos.cat
##
##              Df Sum of Sq    RSS     AIC
## - Vertical    1   0.00700 6.3782 -2854.7
## <none>                    6.3712 -2853.4
## + Shuttle     1   0.00301 6.3681 -2851.7
## - BenchReps   1   0.23297 6.6041 -2832.9
## - X3Cone      1   0.23911 6.6103 -2832.3
## - pos.cat     6   0.41754 6.7887 -2825.6
## - Broad.Jump  1   0.90494 7.2761 -2772.1
## - Wt          1   1.12172 7.4929 -2753.7
##
## Step:  AIC=-2854.72
## X40YD ~ Wt + BenchReps + Broad.Jump + X3Cone + pos.cat
##
##              Df Sum of Sq    RSS     AIC
## <none>                    6.3782 -2854.7
## + Vertical    1   0.00700 6.3712 -2853.4
## + Shuttle     1   0.00438 6.3738 -2853.2
## - X3Cone      1   0.23955 6.6177 -2833.6
## - BenchReps   1   0.24861 6.6268 -2832.7
## - pos.cat     6   0.42355 6.8017 -2826.4
## - Wt          1   1.15161 7.5298 -2752.7
## - Broad.Jump  1   1.55276 7.9309 -2720.1
```

Using the step function, it recommeneded I take out vertical and shuttle. #Best subset model

```
fit3 = lm(X40YD ~ Wt + BenchReps + Broad.Jump + X3Cone + pos.cat)
summary(fit3)

##
## Call:
## lm(formula = X40YD ~ Wt + BenchReps + Broad.Jump + X3Cone + pos.cat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35806 -0.05926  0.00133  0.06118  0.35870
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.2738777  0.1820724  23.474  < 2e-16 ***
## Wt           0.0035877  0.0003402  10.546  < 2e-16 ***
## BenchReps   -0.0043232  0.0008823  -4.900 1.23e-06 ***
## Broad.Jump  -0.0088350  0.0007215 -12.246  < 2e-16 ***
## X3Cone       0.0968205  0.0201293   4.810 1.90e-06 ***
## pos.catDL    0.0039155  0.0281172   0.139  0.88929
## pos.catFBTE -0.0005008  0.0234227  -0.021  0.98295
## pos.catLB   -0.0096934  0.0196338  -0.494  0.62169
## pos.catOL    0.0821183  0.0344337   2.385  0.01739 *
## pos.catRB   -0.0332475  0.0179666  -1.851  0.06472 .
## pos.catWR   -0.0501266  0.0145697  -3.440  0.00062 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1018 on 616 degrees of freedom
## Multiple R-squared:  0.8923, Adjusted R-squared:  0.8906
## F-statistic: 510.5 on 10 and 616 DF,  p-value: < 2.2e-16
```

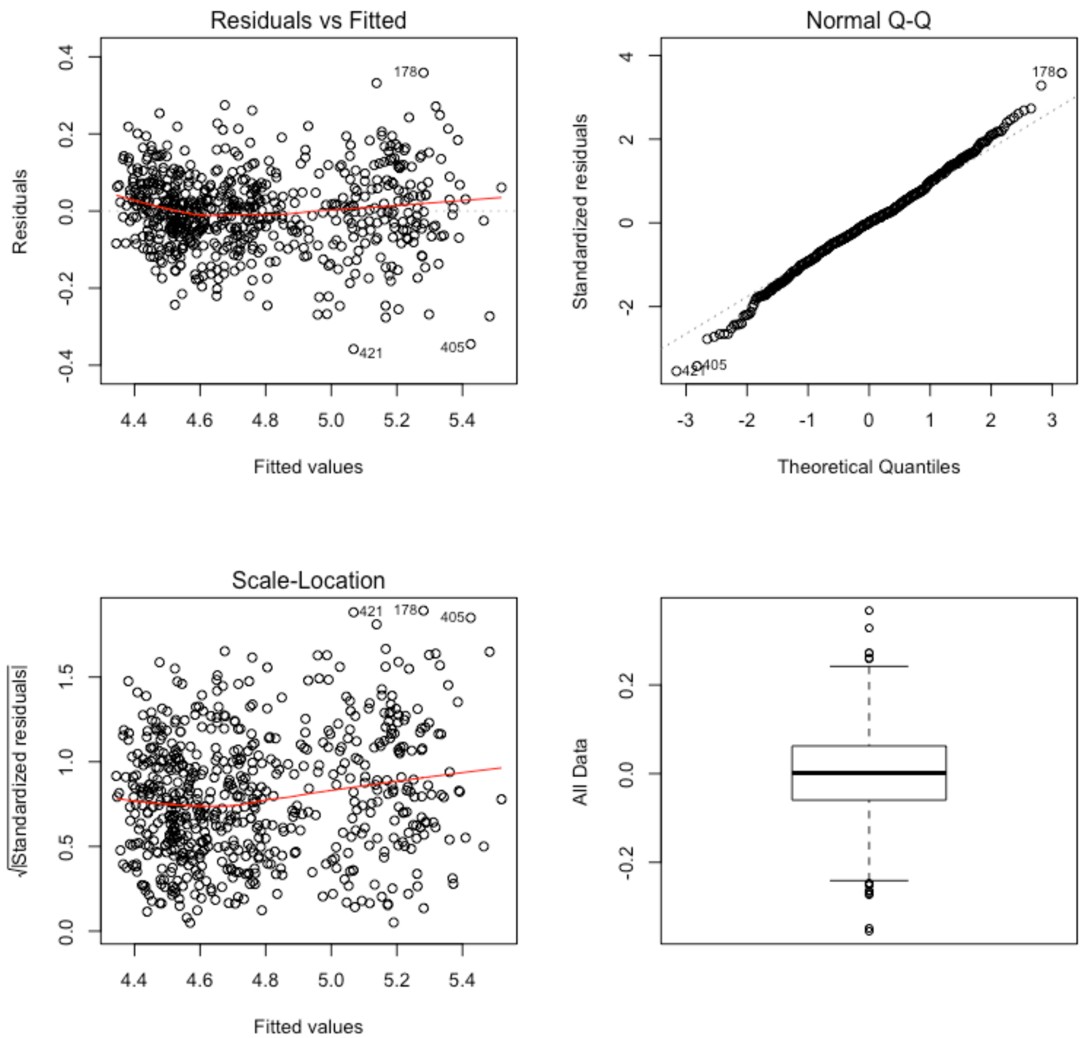
The estimates in my summary of fit3 tell me that an increase in weight and a slower 3-cone drill time will result in a slower 40-yard dash time, which is expected. A longer broad jump results in a faster dash time, which is also expected. However, an increase in bench reps results in a faster 40-yard dash time, which is unexpected. This could be a symptom of collinearity, or could also show that this bench test could be more of a test of a player's endurance, rather than his strength. Upon looking at the bench reps of players in the dataset, however, it appears that in many cases, lower bench reps results in faster 40-yard dash times, and vice versa. This must be an issue of collinearity. Weight, bench reps, broad jump, and three cone drill are all very significant

When looking at positions, it appears that fullbacks/tight ends and linebackers are faster than defensive backs, which is unexpected, but not very significant. Running backs and wide receivers are both faster than defensive backs, with wide receivers being the most significant. Offensive and defensive linemen are expectedly slower than defensive backs.

R-squared and adjusted R-squared did not have significant changes.

Residual Analysis on Best Subsets Model

```
par (mfrow=c(2,2))
plot(fit3, which = 1:3)
boxplot(fit1$residuals,ylab="All Data")
```



There still appears to be constant variance and normal distribution of values. The two groups of fast and slow players are still very distinct. #Interaction Effects

```
fit4 = lm(X40YD ~ (Wt + BenchReps + Broad.Jump + X3Cone + pos.cat)^2)
summary(fit4)

##
## Call:
## lm(formula = X40YD ~ (Wt + BenchReps + Broad.Jump + X3Cone +
##     pos.cat)^2)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.34517 -0.05682  0.00309  0.05514  0.34128
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             6.197e+00  3.688e+00   1.681   0.0934 .
## Wt                      1.646e-02  1.240e-02   1.328   0.1848
## BenchReps              -3.477e-02  3.869e-02  -0.899   0.3692
## Broad.Jump             -1.811e-02  2.050e-02  -0.883   0.3774
## X3Cone                 -7.017e-01  5.433e-01  -1.292   0.1970
## pos.catDL               2.603e-01  1.279e+00   0.204   0.8388
## pos.catFBTE             7.099e-01  1.147e+00   0.619   0.5363
## pos.catLB               9.247e-01  9.232e-01   1.002   0.3169
## pos.catOL               1.049e+00  1.640e+00   0.640   0.5225
## pos.catRB               1.296e+00  8.515e-01   1.523   0.1284
## pos.catWR               1.067e+00  6.812e-01   1.567   0.1176
## Wt:BenchReps           -3.373e-05  6.066e-05  -0.556   0.5783
## Wt:Broad.Jump          -1.157e-04  5.042e-05  -2.294   0.0221 *
## Wt:X3Cone               3.403e-04  1.352e-03   0.252   0.8014
## Wt:pos.catDL           -1.826e-03  1.515e-03  -1.205   0.2285
## Wt:pos.catFBTE         -4.059e-03  2.050e-03  -1.980   0.0482 *
## Wt:pos.catLB           -6.798e-04  1.784e-03  -0.381   0.7033
## Wt:pos.catOL           -4.452e-03  2.101e-03  -2.120   0.0345 *
## Wt:pos.catRB           -1.134e-03  1.738e-03  -0.652   0.5146
## Wt:pos.catWR           -2.374e-03  1.334e-03  -1.779   0.0758 .
## BenchReps:Broad.Jump    1.092e-04  1.529e-04   0.714   0.4754
## BenchReps:X3Cone        2.921e-03  4.163e-03   0.702   0.4832
## BenchReps:pos.catDL     5.989e-03  5.827e-03   1.028   0.3045
## BenchReps:pos.catFBTE   1.138e-02  4.799e-03   2.371   0.0181 *
## BenchReps:pos.catLB     4.809e-03  4.397e-03   1.094   0.2745
## BenchReps:pos.catOL     4.901e-03  6.700e-03   0.732   0.4648
## BenchReps:pos.catRB     6.828e-03  3.945e-03   1.731   0.0840 .
## BenchReps:pos.catWR     3.819e-03  3.603e-03   1.060   0.2896
## Broad.Jump:X3Cone       5.408e-03  3.300e-03   1.639   0.1018
## Broad.Jump:pos.catDL   -2.364e-03  4.712e-03  -0.502   0.6161
## Broad.Jump:pos.catFBTE -1.020e-02  4.100e-03  -2.487   0.0132 *
## Broad.Jump:pos.catLB   -2.797e-03  3.238e-03  -0.864   0.3880
## Broad.Jump:pos.catOL   -1.658e-03  5.557e-03  -0.298   0.7655
## Broad.Jump:pos.catRB   -3.424e-03  3.454e-03  -0.991   0.3220
## Broad.Jump:pos.catWR   -2.367e-03  2.630e-03  -0.900   0.3685
## X3Cone:pos.catDL        5.003e-02  1.274e-01   0.393   0.6946
## X3Cone:pos.catFBTE      1.778e-01  1.187e-01   1.497   0.1349
## X3Cone:pos.catLB       -7.388e-02  9.263e-02  -0.798   0.4254
## X3Cone:pos.catOL        5.589e-02  1.513e-01   0.370   0.7119
## X3Cone:pos.catRB       -1.129e-01  9.866e-02  -1.145   0.2528
## X3Cone:pos.catWR       -5.946e-02  8.367e-02  -0.711   0.4776
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09802 on 586 degrees of freedom
```

```
## Multiple R-squared:  0.905,  Adjusted R-squared:  0.8985
## F-statistic: 139.5 on 40 and 586 DF,  p-value: < 2.2e-16
```

When interaction effects were added, we can see that not many of them were significant, except for almost anything related to the FBTE position. Weight and position category were very significant, meaning that weight may show up in our final model.

Along with our our five significant predictors, we will now use step to create our final model with significant interaction effects added.

Final Model

Residual Analysis

```
#We will use step4 for our final model
step4=step(fit4, direction = "both")

## Start:  AIC=-2872.91
## X40YD ~ (Wt + BenchReps + Broad.Jump + X3Cone + pos.cat)^2
## 
##                         Df Sum of Sq    RSS     AIC
## - BenchReps:pos.cat      6  0.075204 5.7056 -2876.6
## - X3Cone:pos.cat         6  0.079867 5.7103 -2876.1
## - Broad.Jump:pos.cat     6  0.084240 5.7146 -2875.6
## - Wt:pos.cat             6  0.089160 5.7195 -2875.1
## - Wt:X3Cone              1  0.000608 5.6310 -2874.8
## - Wt:BenchReps           1  0.002971 5.6334 -2874.6
## - BenchReps:X3Cone       1  0.004730 5.6351 -2874.4
## - BenchReps:Broad.Jump   1  0.004901 5.6353 -2874.4
## <none>                              5.6304 -2872.9
## - Broad.Jump:X3Cone      1  0.025805 5.6562 -2872.0
## - Wt:Broad.Jump          1  0.050560 5.6809 -2869.3
## 
## Step:  AIC=-2876.59
## X40YD ~ Wt + BenchReps + Broad.Jump + X3Cone + pos.cat + Wt:BenchReps + 
##     Wt:Broad.Jump + Wt:X3Cone + Wt:pos.cat + BenchReps:Broad.Jump + 
##     BenchReps:X3Cone + Broad.Jump:X3Cone + Broad.Jump:pos.cat + 
##     X3Cone:pos.cat
## 
##                         Df Sum of Sq    RSS     AIC
## - Broad.Jump:pos.cat     6  0.076480 5.7821 -2880.2
## - Wt:pos.cat             6  0.078454 5.7840 -2880.0
## - Wt:BenchReps           1  0.000081 5.7057 -2878.6
## - Wt:X3Cone              1  0.000286 5.7059 -2878.6
## - BenchReps:X3Cone       1  0.001392 5.7070 -2878.4
## - BenchReps:Broad.Jump   1  0.012995 5.7186 -2877.2
## - X3Cone:pos.cat         6  0.108442 5.8140 -2876.8
## <none>                              5.7056 -2876.6
## - Broad.Jump:X3Cone      1  0.025056 5.7306 -2875.8
```

```
## + BenchReps:pos.cat     6  0.075204 5.6304 -2872.9
## - Wt:Broad.Jump         1  0.055409 5.7610 -2872.5
##
## Step:  AIC=-2880.24
## X40YD ~ Wt + BenchReps + Broad.Jump + X3Cone + pos.cat + Wt:BenchReps +
##     Wt:Broad.Jump + Wt:X3Cone + Wt:pos.cat + BenchReps:Broad.Jump +
##     BenchReps:X3Cone + Broad.Jump:X3Cone + X3Cone:pos.cat
##
##                        Df Sum of Sq    RSS     AIC
## - Wt:pos.cat            6  0.090339 5.8724 -2882.5
## - Wt:BenchReps          1  0.000007 5.7821 -2882.2
## - Wt:X3Cone             1  0.000793 5.7829 -2882.2
## - BenchReps:X3Cone      1  0.000858 5.7829 -2882.1
## - BenchReps:Broad.Jump  1  0.007970 5.7900 -2881.4
## <none>                              5.7821 -2880.2
## - Broad.Jump:X3Cone     1  0.045965 5.8280 -2877.3
## - X3Cone:pos.cat        6  0.145186 5.9273 -2876.7
## + Broad.Jump:pos.cat    6  0.076480 5.7056 -2876.6
## + BenchReps:pos.cat     6  0.067444 5.7146 -2875.6
## - Wt:Broad.Jump         1  0.179436 5.9615 -2863.1
##
## Step:  AIC=-2882.52
## X40YD ~ Wt + BenchReps + Broad.Jump + X3Cone + pos.cat + Wt:BenchReps +
##     Wt:Broad.Jump + Wt:X3Cone + BenchReps:Broad.Jump + BenchReps:X3Cone +
##     Broad.Jump:X3Cone + X3Cone:pos.cat
##
##                        Df Sum of Sq    RSS     AIC
## - Wt:BenchReps          1  0.001481 5.8739 -2884.4
## - BenchReps:X3Cone      1  0.003770 5.8762 -2884.1
## - BenchReps:Broad.Jump  1  0.004491 5.8769 -2884.0
## - Wt:X3Cone             1  0.014571 5.8870 -2883.0
## <none>                              5.8724 -2882.5
## - Broad.Jump:X3Cone     1  0.026979 5.8994 -2881.6
## + Wt:pos.cat            6  0.090339 5.7821 -2880.2
## + Broad.Jump:pos.cat    6  0.088365 5.7840 -2880.0
## + BenchReps:pos.cat     6  0.061320 5.8111 -2877.1
## - X3Cone:pos.cat        6  0.204945 6.0774 -2873.0
## - Wt:Broad.Jump         1  0.148481 6.0209 -2868.9
##
## Step:  AIC=-2884.36
## X40YD ~ Wt + BenchReps + Broad.Jump + X3Cone + pos.cat + Wt:Broad.Jump +
##     Wt:X3Cone + BenchReps:Broad.Jump + BenchReps:X3Cone +
Broad.Jump:X3Cone +
##     X3Cone:pos.cat
##
##                        Df Sum of Sq    RSS     AIC
## - BenchReps:X3Cone      1  0.002302 5.8762 -2886.1
## - BenchReps:Broad.Jump  1  0.006629 5.8805 -2885.7
## - Wt:X3Cone             1  0.018334 5.8922 -2884.4
## <none>                              5.8739 -2884.4
```

```
## - Broad.Jump:X3Cone     1  0.025591 5.8995 -2883.6
## + Wt:BenchReps          1  0.001481 5.8724 -2882.5
## + Wt:pos.cat            6  0.091813 5.7821 -2882.2
## + Broad.Jump:pos.cat    6  0.089534 5.7844 -2882.0
## + BenchReps:pos.cat     6  0.059106 5.8148 -2878.7
## - X3Cone:pos.cat        6  0.214877 6.0888 -2873.8
## - Wt:Broad.Jump         1  0.147526 6.0214 -2870.8
##
## Step:  AIC=-2886.11
## X40YD ~ Wt + BenchReps + Broad.Jump + X3Cone + pos.cat + Wt:Broad.Jump +
##     Wt:X3Cone + BenchReps:Broad.Jump + Broad.Jump:X3Cone + X3Cone:pos.cat
##
##                        Df Sum of Sq    RSS     AIC
## - BenchReps:Broad.Jump  1  0.004811 5.8810 -2887.6
## - Wt:X3Cone             1  0.016080 5.8923 -2886.4
## <none>                              5.8762 -2886.1
## - Broad.Jump:X3Cone     1  0.025443 5.9016 -2885.4
## + BenchReps:X3Cone      1  0.002302 5.8739 -2884.4
## + Wt:BenchReps          1  0.000013 5.8762 -2884.1
## + Wt:pos.cat            6  0.092793 5.7834 -2884.1
## + Broad.Jump:pos.cat    6  0.086986 5.7892 -2883.5
## + BenchReps:pos.cat     6  0.053755 5.8224 -2879.9
## - X3Cone:pos.cat        6  0.216280 6.0925 -2875.4
## - Wt:Broad.Jump         1  0.156613 6.0328 -2871.6
##
## Step:  AIC=-2887.6
## X40YD ~ Wt + BenchReps + Broad.Jump + X3Cone + pos.cat + Wt:Broad.Jump +
##     Wt:X3Cone + Broad.Jump:X3Cone + X3Cone:pos.cat
##
##                        Df Sum of Sq    RSS     AIC
## - Wt:X3Cone             1  0.016612 5.8976 -2887.8
## <none>                              5.8810 -2887.6
## - Broad.Jump:X3Cone     1  0.022822 5.9038 -2887.2
## + BenchReps:Broad.Jump  1  0.004811 5.8762 -2886.1
## + Wt:BenchReps          1  0.002804 5.8782 -2885.9
## + BenchReps:X3Cone      1  0.000484 5.8805 -2885.7
## + Wt:pos.cat            6  0.087010 5.7940 -2884.9
## + Broad.Jump:pos.cat    6  0.082347 5.7987 -2884.4
## + BenchReps:pos.cat     6  0.058547 5.8225 -2881.9
## - X3Cone:pos.cat        6  0.215785 6.0968 -2877.0
## - Wt:Broad.Jump         1  0.156366 6.0374 -2873.2
## - BenchReps             1  0.253727 6.1347 -2863.1
##
## Step:  AIC=-2887.83
## X40YD ~ Wt + BenchReps + Broad.Jump + X3Cone + pos.cat + Wt:Broad.Jump +
##     Broad.Jump:X3Cone + X3Cone:pos.cat
##
##                        Df Sum of Sq    RSS     AIC
## <none>                              5.8976 -2887.8
## + Wt:X3Cone             1  0.016612 5.8810 -2887.6
```

```
## - Broad.Jump:X3Cone     1  0.022508 5.9201 -2887.4
## + Wt:pos.cat            6  0.102500 5.7951 -2886.8
## + Wt:BenchReps          1  0.008735 5.8889 -2886.8
## + BenchReps:Broad.Jump  1  0.005343 5.8923 -2886.4
## + BenchReps:X3Cone      1  0.003442 5.8942 -2886.2
## + Broad.Jump:pos.cat    6  0.093947 5.8037 -2885.9
## + BenchReps:pos.cat     6  0.060058 5.8376 -2882.2
## - X3Cone:pos.cat        6  0.221916 6.1195 -2876.7
## - Wt:Broad.Jump         1  0.140653 6.0383 -2875.1
## - BenchReps             1  0.248018 6.1456 -2864.0
```

The stepwise procedure added three significant interactions to our model: Weight and Broad Jump, Broad Jump and 3-cone drill, and 3-cone drill and position category.

Broad jump and 3-cone drill appear to be the most significant interactions, which will be interesting to visualize with a scatterplot later.

```
summary(step4)

##
## Call:
## lm(formula = X40YD ~ Wt + BenchReps + Broad.Jump + X3Cone + pos.cat +
##     Wt:Broad.Jump + Broad.Jump:X3Cone + X3Cone:pos.cat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38535 -0.05867  0.00237  0.05522  0.34908
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         5.667e+00  1.933e+00   2.932 0.003497 **
## Wt                  1.302e-02  2.624e-03   4.960 9.16e-07 ***
## BenchReps          -4.362e-03  8.627e-04  -5.057 5.66e-07 ***
## Broad.Jump         -1.635e-02  1.558e-02  -1.049 0.294382
## X3Cone             -4.467e-01  3.355e-01  -1.331 0.183544
## pos.catDL          -7.809e-01  4.331e-01  -1.803 0.071868 .
## pos.catFBTE        -2.036e+00  6.389e-01  -3.187 0.001512 **
## pos.catLB           2.324e-01  4.927e-01   0.472 0.637334
## pos.catOL          -4.205e-01  5.505e-01  -0.764 0.445315
## pos.catRB           5.724e-01  6.266e-01   0.913 0.361355
## pos.catWR           4.143e-01  5.471e-01   0.757 0.449176
## Wt:Broad.Jump      -8.757e-05  2.300e-05  -3.808 0.000154 ***
## Broad.Jump:X3Cone   4.186e-03  2.748e-03   1.523 0.128207
## X3Cone:pos.catDL    1.198e-01  6.188e-02   1.936 0.053301 .
## X3Cone:pos.catFBTE  2.952e-01  9.024e-02   3.271 0.001130 **
## X3Cone:pos.catLB   -2.473e-02  7.012e-02  -0.353 0.724428
## X3Cone:pos.catOL    8.117e-02  7.538e-02   1.077 0.281992
## X3Cone:pos.catRB   -8.284e-02  8.990e-02  -0.921 0.357186
## X3Cone:pos.catWR   -6.690e-02  7.931e-02  -0.843 0.399308
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09849 on 608 degrees of freedom
## Multiple R-squared:  0.9004, Adjusted R-squared:  0.8975
## F-statistic: 305.5 on 18 and 608 DF,  p-value: < 2.2e-16
```

Summary of final model

Of the interaction effects, weight x broad jump appears to be the most significant, while 3-cone drill x position category has some significant interactions, including defensive line, and fullbacks and tight ends. I will use this model because of its interesting interaction plots that will be analyzed later.

The estimates of the original predictor variables appear to have been affected by the addition of interactions, but we can see that the interaction effect of weight and broad jump has a significant impact on 40-yard dash time, as well as 3-cone drill x FBTE and DL.

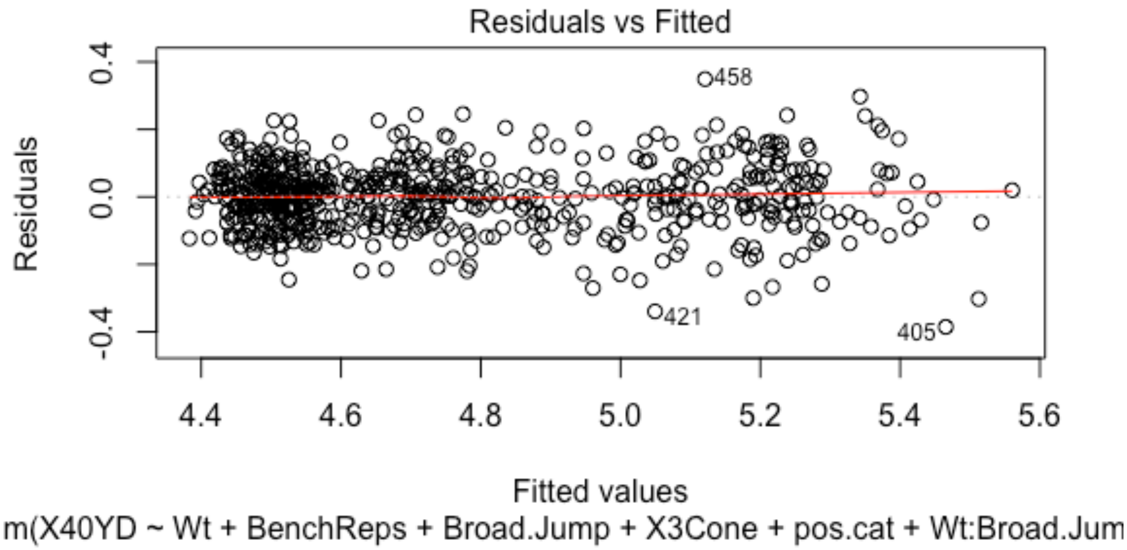
Comparing Adjusted R^2 values

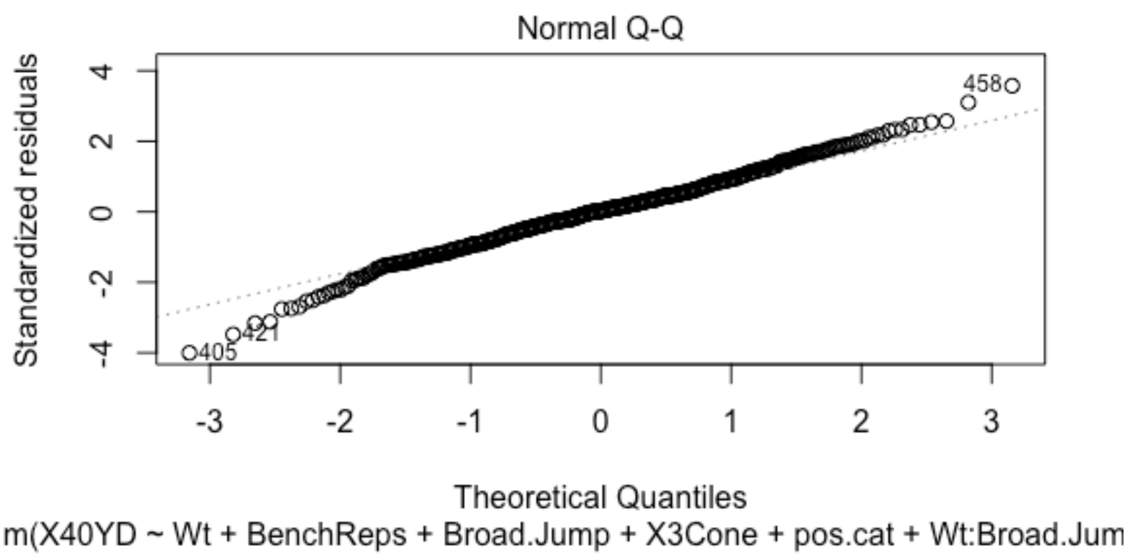
```
cbind.data.frame (Model=c("Original Model", "Manual Elim", "Stepwise 
Original", "Stepwise Interactions"),
                  Adj.R2 = c(summary (fit1)$adj.r.squared,
                             summary (fit2)$adj.r.squared,
                             summary (fit3)$adj.r.squared,
                             summary (step4)$adj.r.squared))

##                   Model    Adj.R2
## 1        Original Model 0.8903951
## 2           Manual Elim 0.8904765
## 3     Stepwise Original 0.8905792
## 4 Stepwise Interactions 0.8974919
```

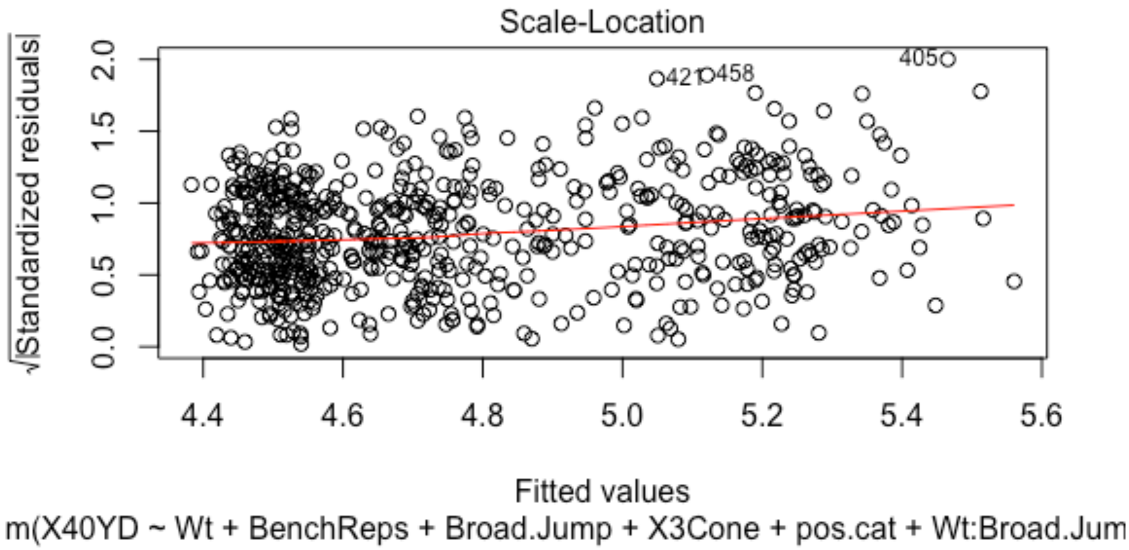
It appears that running the stepwise procedure on our original stepwise function with added interaction effects increased the adjusted R^2 significantly compared to the other procedures.

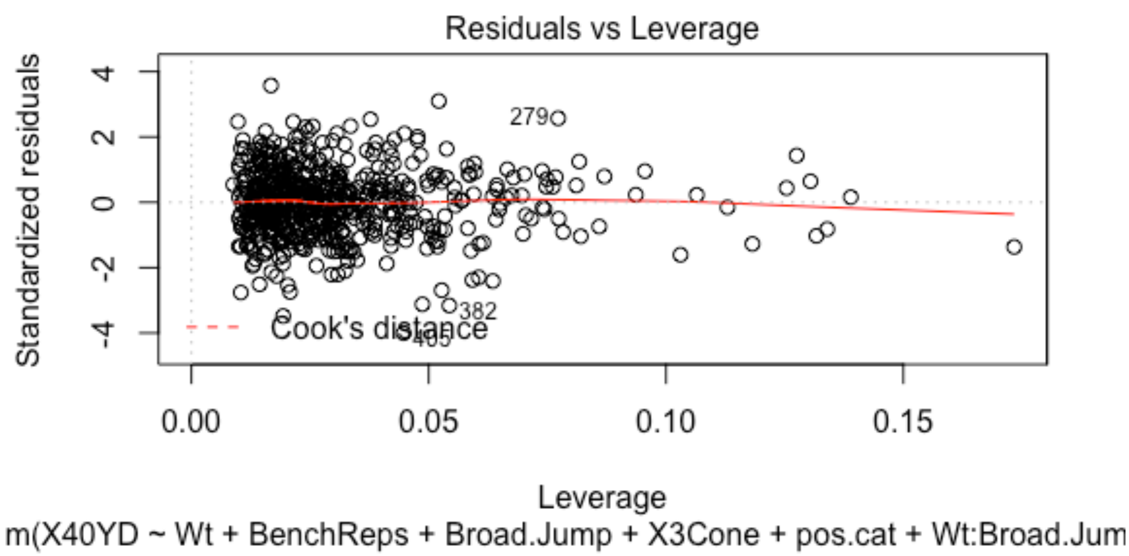
Residual Analysis of Final Model

```
plot(step4)
```


Residuals vs Fitted
Residuals
0.4
0.0
-0.4
458
421
405
4.4
4.6
4.8
5.0
5.2
5.4
5.6
Fitted values
m(X40YD ~ Wt + BenchReps + Broad.Jump + X3Cone + pos.cat + Wt:Broad.Jum


Normal Q-Q
Standardized residuals
4
2
0
-2
-4
458
421
405
-3
-2
-1
0
1
2
3
Theoretical Quantiles
m(X40YD ~ Wt + BenchReps + Broad.Jump + X3Cone + pos.cat + Wt:Broad.Jum

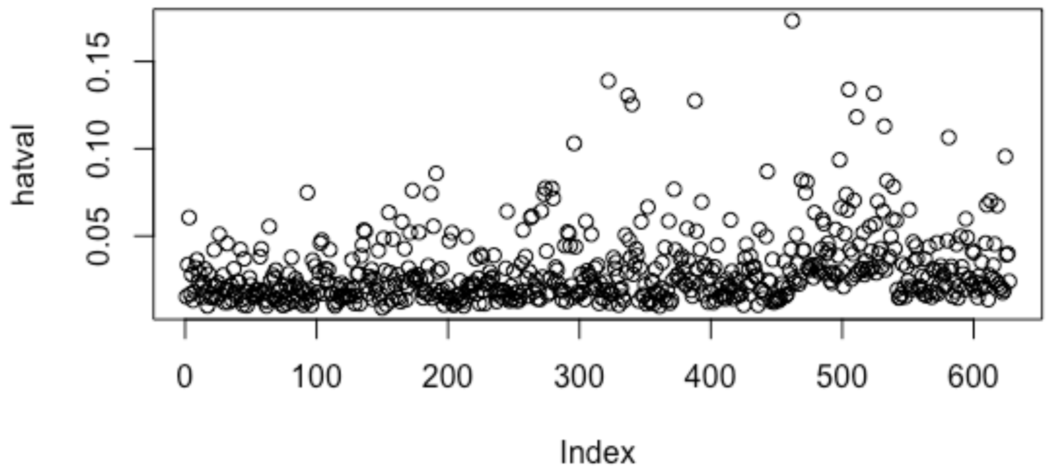




The residuals appear to be normal, with equal variance and no obvious outliers. The Q-Q plot looks normal, and there appears to be equal and constant variance in our scale-location plot. The cutoff for cook's distance does not even show in our window, indicating that their are no highly influential cook's values.

Hat values

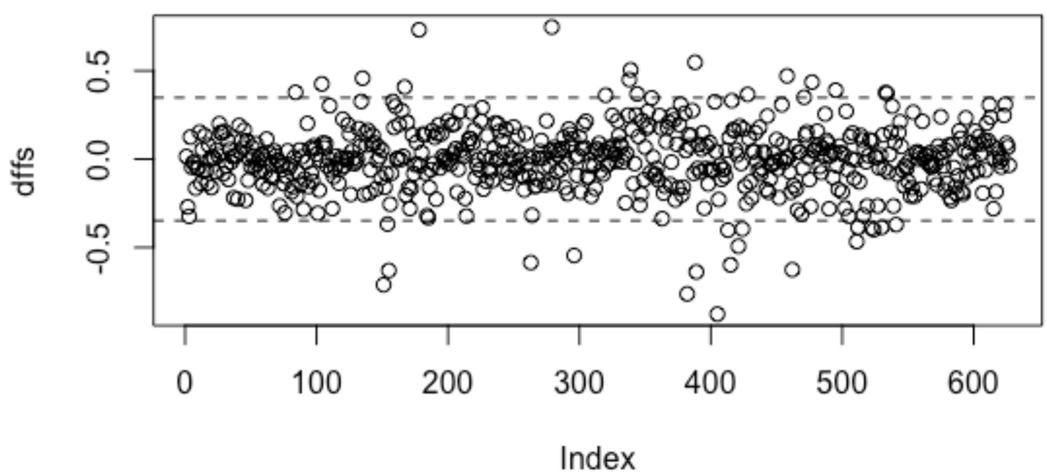
```
hatval = hatvalues(step4)
plot(hatval)
```



There are no obvious outliers to consider removing here.

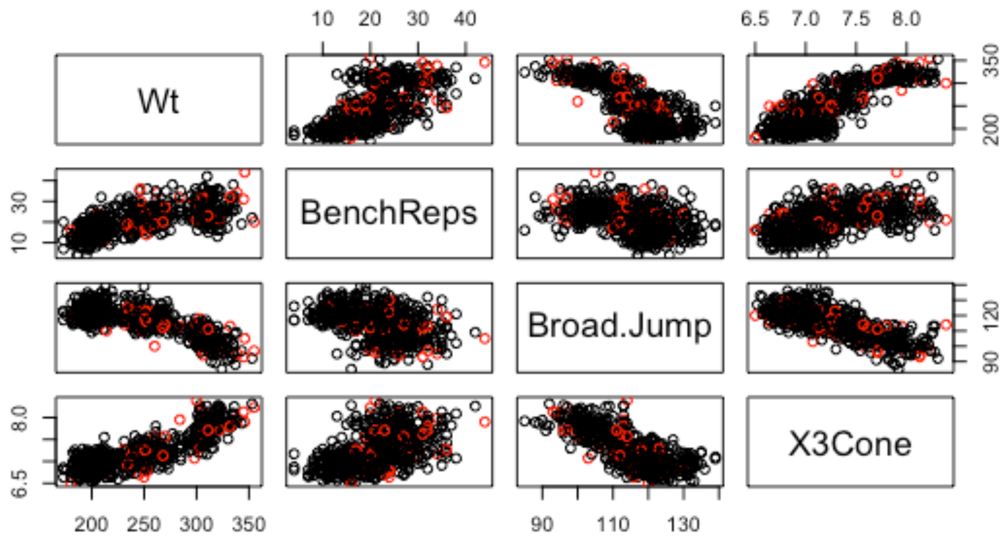
DFFITS

```
dffs = dffits(step4)
plot(dffs)

cutoff = 2*(sqrt(step4$rank/ (step4$rank + step4$df.residual)))
abline(cutoff, 0, lty=2)
abline(-cutoff, 0, lty=2)
```



From the dffits plot, there are a few values that are over the cutoff, indicating that these values may be influencing our data. However, we will check to see if these influential values are significant enough to be harmful to the data with a pairs function containing red points that indicate points above the cutoff.

```
pairs(all[,c(7, 10:12)], col=ifelse(abs(dffs)>cutoff, 2, 1))
```



The points above the cutoff (highlighted in red) don't appear to be outside of the clusters of data, meaning that there are no highly influential values.

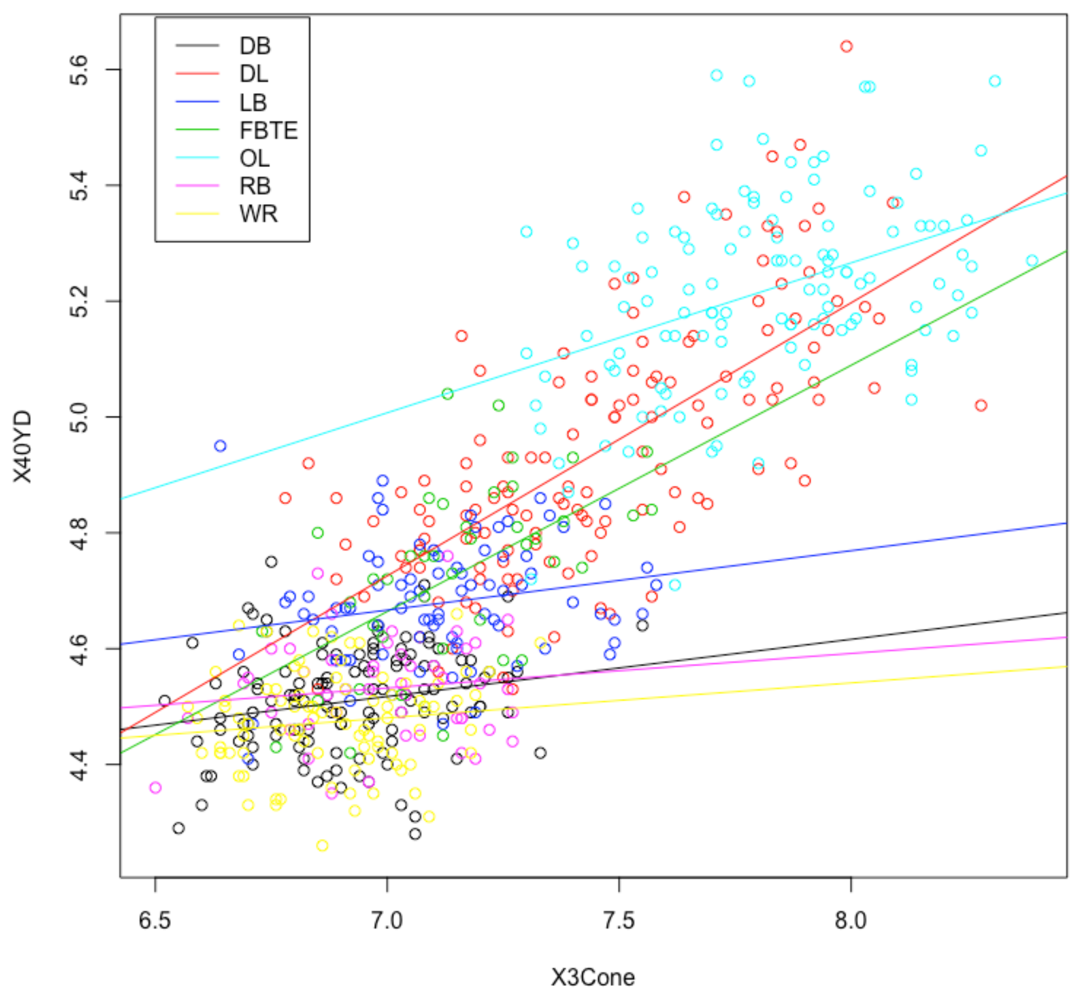
Variance Inflation Factors

```
library(car)

## Loading required package: carData

vif(step4)

##                            GVIF Df GVIF^(1/(2*Df))
## Wt                 9.877029e+02  1       31.427741
## BenchReps          2.181604e+00  1        1.477025
## Broad.Jump         1.433510e+03  1       37.861724
## X3Cone             1.232239e+03  1       35.103262
## pos.cat            5.667274e+18  6       36.541064
## Wt:Broad.Jump      4.990306e+02  1       22.338993
## Broad.Jump:X3Cone  1.068152e+03  1       32.682596
## X3Cone:pos.cat     6.504557e+18  6       36.963079
```

Although the VIF values are mostly above 10 and were improved by performing a k=log(n), it took away two of our interaction effects from our final model. We observed some very interesting interaction plots from our unlogged step4 model that we will observe later, so I decided to keep this model, although it contains relatively high VIF values.

Plot of 40-yard dash vs. 3-cone drill, fitted with an abline for each position

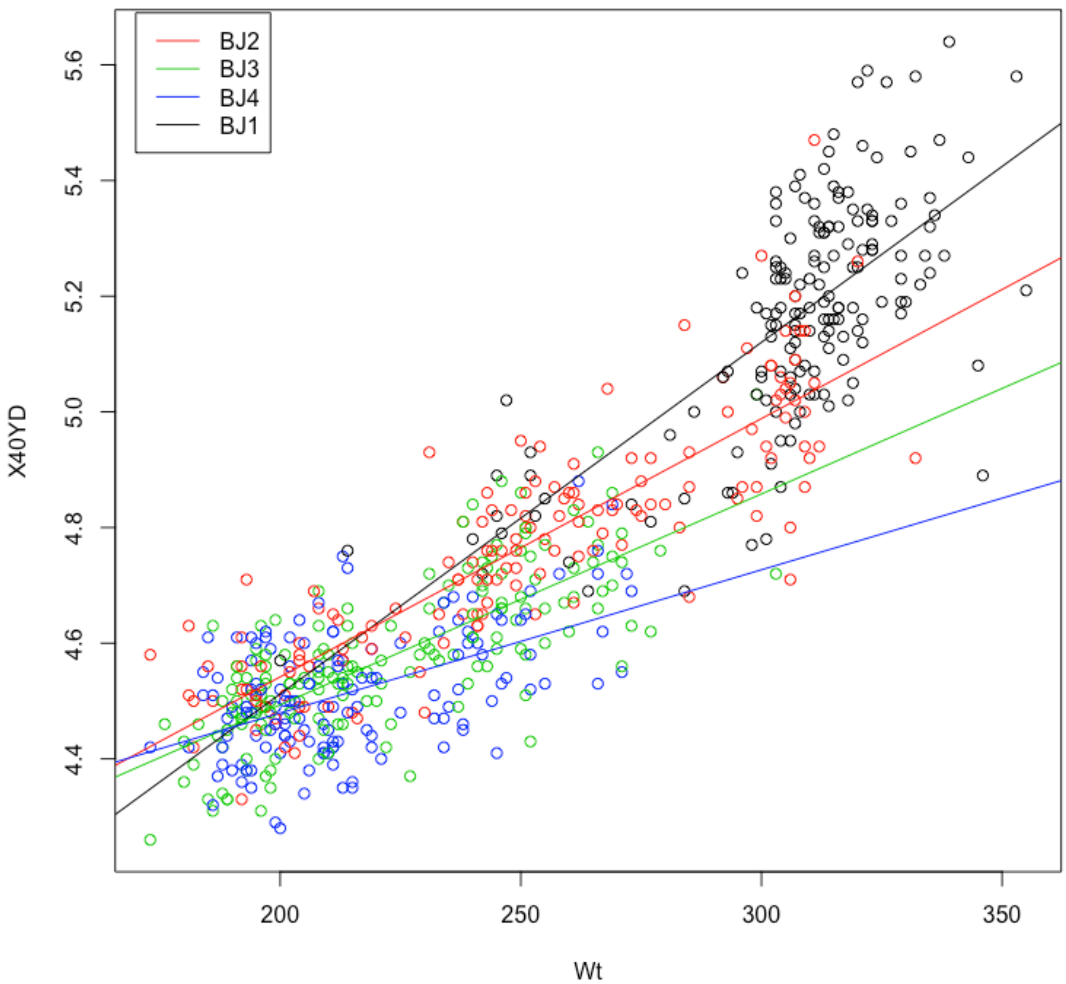
```
plot(X40YD ~ X3Cone, col = as.factor(pos.cat))
abline(lm(X40YD[pos.cat=="DB"] ~ X3Cone[pos.cat=="DB"]),col=1)
abline(lm(X40YD[pos.cat=="DL"] ~ X3Cone[pos.cat=="DL"]),col=2)
abline(lm(X40YD[pos.cat=="FBTE"] ~ X3Cone[pos.cat=="FBTE"]),col=3)
abline(lm(X40YD[pos.cat=="LB"] ~ X3Cone[pos.cat=="LB"]),col=4)
abline(lm(X40YD[pos.cat=="OL"] ~ X3Cone[pos.cat=="OL"]),col=5)
abline(lm(X40YD[pos.cat=="RB"] ~ X3Cone[pos.cat=="RB"]),col=6)
abline(lm(X40YD[pos.cat=="WR"] ~ X3Cone[pos.cat=="WR"]),col=7)
poscats = unique(pos.cat)
legend(6.5, 5.69, poscats, lty = rep(1,7), col=as.factor(poscats))
```



From this plot, it is clear that the 40-yard dash times of the faster player positions (DBs, RBs, WRs, LBs) were not affected very much by three-cone drill time. However, for the larger, slower players (FBs, TEs, OL, DL), their 40-yard dash times were much more affected by three-cone drill.

Compare Weight and 40-Yard Dash by Broad Jump Cluster

```
#categorize broad.jump
BJsum = summary(Broad.Jump)
all$BJ.cat = "BJ1"
all$BJ.cat[Broad.Jump > BJsum[2]] = "BJ2"
all$BJ.cat[Broad.Jump > BJsum[3]] = "BJ3"
all$BJ.cat[Broad.Jump > BJsum[5]] = "BJ4"
all$BJ.cat[Broad.Jump > BJsum[6]] = "BJ5"

plot(X40YD ~ Wt, col = as.factor(all$BJ.cat))
abline(lm(X40YD[all$BJ.cat=="BJ1"] ~ Wt[all$BJ.cat=="BJ1"]),col=1)
abline(lm(X40YD[all$BJ.cat=="BJ2"] ~ Wt[all$BJ.cat=="BJ2"]),col=2)
abline(lm(X40YD[all$BJ.cat=="BJ3"] ~ Wt[all$BJ.cat=="BJ3"]),col=3)
abline(lm(X40YD[all$BJ.cat=="BJ4"] ~ Wt[all$BJ.cat=="BJ4"]),col=4)

BJcats = unique(all$BJ.cat)
legend(170, 5.69, BJcats, lty = rep(1,7), col=as.factor(BJcats))
```



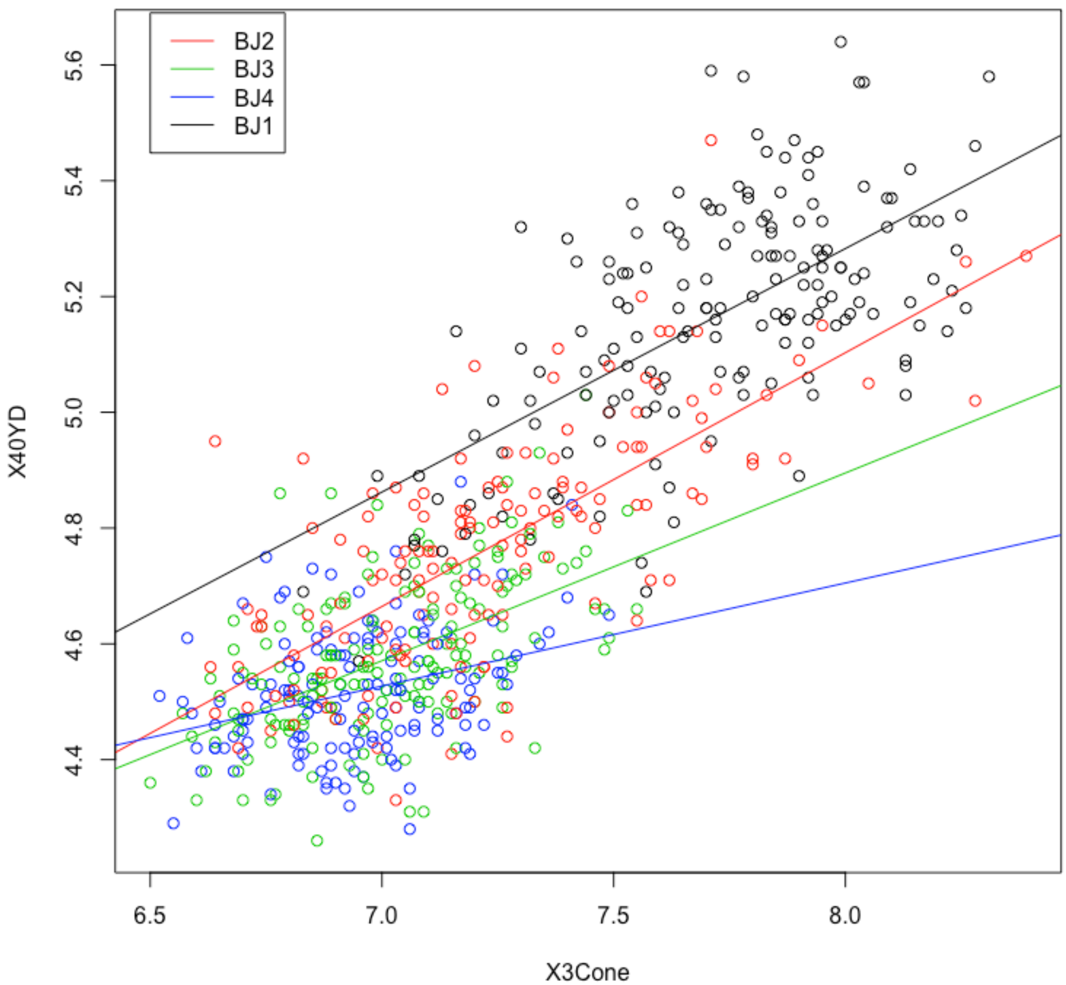

```
#detach(all)
#attach(all)

#Make scatterplot with Broad Jump and ablines
```

To make this plot, I first had to categorize the broad jump values into categories. I categorized them into their four quartiles. From this plot, we see four distinct slopes for each cluster. The black points, the shortest broad jumps' 40-yard dash times are the most effected by weight, and as broad jump gets longer, the player's 40-yard dash is less effected by their weight.

Compare 3-Cone Drill and 40-Yard Dash by Broad Jump Cluster

```
plot(X40YD ~ X3Cone, col = as.factor(all$BJ.cat))
abline(lm(X40YD[all$BJ.cat=="BJ1"] ~ X3Cone[all$BJ.cat=="BJ1"]),col=1)
abline(lm(X40YD[all$BJ.cat=="BJ2"] ~ X3Cone[all$BJ.cat=="BJ2"]),col=2)
```

```
abline(lm(X40YD[all$BJ.cat=="BJ3"] ~ X3Cone[all$BJ.cat=="BJ3"]),col=3)
abline(lm(X40YD[all$BJ.cat=="BJ4"] ~ X3Cone[all$BJ.cat=="BJ4"]),col=4)

#BJcats = unique(all$BJ.cat)
legend(6.5, 5.69, BJcats, lty = rep(1,7), col=as.factor(BJcats))
```




```
detach(all)
attach(all)
```

Here we can see again that the shortest broad jumps have the longest 40-yard dash times. All broad jump groups' 40-yard dash times seem to be affected about the same by 3-cone drill except for the longest broad jumps which is interesting.

Final Model Predictions

```
pred4 = predict(step4, interval="prediction")
```

```
## Warning in predict.lm(step4, interval = "prediction"): predictions on
current data refer to _future_ responses
```

```r
#names(pred4)=c("fit", "lwr", "upr")
all$fit = pred4[,1]
all$lwr = pred4[,2]
all$upr = pred4[,3]

all[c(46,223,231,289,393,369), c(3, 7, 8, 19:21, 9:13, 17)]
```

```
##                          Player  Wt X40YD      fit      lwr      upr
## 46        Trae Waynes\\WaynTr00 186  4.31 4.475254 4.280054 4.670453
## 223 Ha Ha Clinton-Dix\\ClinHa00 208  4.58 4.581678 4.385777 4.777579
## 231      Luke Kuechly\\KuecLu00 242  4.58 4.614727 4.419011 4.810442
## 289      Anthony Barr\\BarrAn00 255  4.66 4.732469 4.534796 4.930143
## 393       Zach Banner\\BannZa00 353  5.58 5.560363 5.360324 5.760401
## 369       Zach Fulton\\FultZa00 316  5.16 5.288613 5.094071 5.483154
##     Vertical BenchReps Broad.Jump X3Cone Shuttle pos.cat
## 46      38.0        19        122   7.06    4.39      DB
## 223     33.0        11        119   7.16    4.16      DB
## 231     38.0        27        123   6.92    4.12      LB
## 289     34.5        15        119   6.82    4.19      LB
## 393     23.5        22         92   8.31    5.21      OL
## 369     24.5        25         98   7.87    5.16      OL
```

```r
mean(all$lwr <= all$X40YD & all$X40YD <= all$upr)
```

```
## [1] 0.9505582
```

From these predictions, it is clear that I was as successful as expected, with a 95% confidence level. It appears that as 40-yard dash gets slower, it is easier to predict.

Conclusion

For NFL prospects between 2012 and 2017, I have found that:

1. When I cleaned the data, there was several more fast and agile players than slow players because of the way I cleaned the data. In the future, I should try to use an equal amount of players in each position category.

2. The data was very normally distributed, which was expected, because tge variables were based on physical tests. These typically have a lot of observations in the middle, with some high, and some low, but never have very high outliers.

3. Any skew observed was due to the fact that the data had more fast players than slow.

4. The initial predictor variables were weight, vertical jump, broad jump, shuttle, three-cone drill, and bench.

5. I decided to add in the categorical variables "position categories", which eventually increased the r-squared of our model.
6. Adding interaction effects increased the r-squared the most (increased from originally .8904 to .8975) and gave us some very interesting interaction plots.
7. The final model, after many methods of predictor variable elimination, boiled down to these most signicant predictors: weight, bench reps, broad jump, three-cone drill, and position category. Using stepwise regression on our interactions added three interaction effects to our final model: Weight x broad jump, broad jump x three-cone drill, and three-cone drill x position category.
8. Average 40-yard dash time is faster with faster three-cone and shuttle drills, higher vertical, and longer broad jump. Bench reps also causes 40-yard dash time to become faster, but we determined that this may be a result of collinearity. 40-yard dash time becomes slower with higher weight, which is expected.
9. In the interaction plots, I discovered that the faster, more agile positions' 40-yard dash times are less affected by their three-cone drill times. I also discovered that the longer the broad jump, the less a player's 40-yard dash time is affected by weight. Finally, I discovered that three-cone drill's effect on 40-yard dash is consistent with each broad jump category, except for the farthest broad jumpers, who's 40-yard dash times are less affected by three-cone drill.
10. Example prediction intervals predicted 95% correct, and appeared to become more accurate as the 40-yard dash time got slower.