Identifying Value in the NFL Draft

Caleb ONeel

I. Summary

This study examines historical NFL player data to find factors that have been overlooked predictors of how good a player will be, and which NFL teams have been good at drafting and developing talent for the past decade. The variables were a mix of physical attributes that are measured at the NFL draft combine and several categorical variables such as the players position, NFL team that drafted them, and what round they were selected in. To examine the relationship between these factors and the players performance, I tested both Hierarchical models and linear models before ultimately landing on a multiple linear regression model.

II. Introduction:

In this assignment I aggregated various data sources to look for predictors of a players on field performance relative to their expectations based on draft position. I quantified the players on field performances by using Pro Football Focus's player rating data, and determined their expected performance by what pick they were drafted. I calculated the "expected value" for a player taken at each pick in the draft by taking the player grade for every player selected at that pick in the ten drafts in my dataset. I used a rolling average for the 10 picks around them to smooth out the variance caused by thin sample size. Then, I subtracted the expected player grade based on draft position from every player's career player grade to generate a value score, with 0 meaning they met expectations of where they were drafted. This value score for each player is what I ultimately use for the dependent variable in my analysis.

In the NFL, there is an annual "Draft Combine" where hundreds of college football players that submitted their candidacy to the NFL draft are invited to showcase their abilities. At this combine, they perform a number of drills designed to measure their athleticism and physical attributes. These measurements are believed to have some predictive power on how the player will translate to the NFL. In addition to the physical attributes measured at the draft combine, I want to look at other variables such as the team that drafted them, the round they were selected in, and the position they play. All three of these factors could potentially have important roles in determining how well a player performs. The team that drafts them has the important job of developing the player, and their ability to develop a given player varies with the coaching, training and support staff at a given organization.

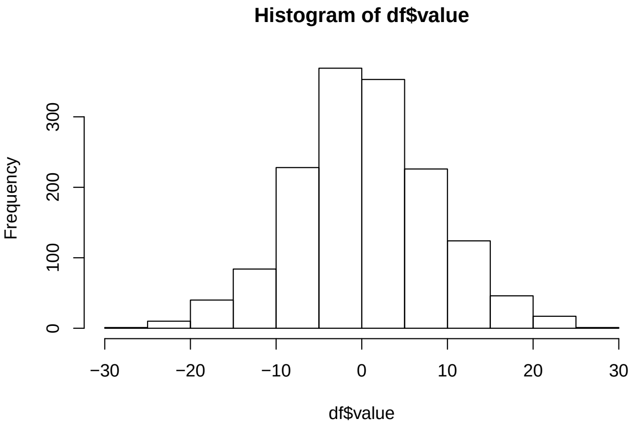
III. Data

The data I used in this study came primarily from Pro Football Focus and Football Reference. I used Pro Football Focus for the player performance, and Football Reference for draft information and combine measurements spanning the years 2010 to 2019. The players performances were quantified using Pro Football Focuses player grade data. This player grade data is very highly regarded and is used by professional gamblers, media organizations, and even NFL teams. Each player is graded for every snap of every game they play in. The graders are full time employees made up in large part of former NFL scouts, and analysts. Each grade is reviewed by multiple people. PFF adds much more context than traditional stats. Traditional statistics cannot differentiate between a simple screen pass that a receiver runs for 50 yards and a perfectly thrown pass into double coverage that the receiver catches. These both appear as 50 passing yards on a score sheet, but the second throw is exponentially harder to execute from the Quarterbacks perspective. Each player is given a grade of -2 to +2 in 0.5 increments on a given play with 0 generally being the average or "expected" grade.

Joining these two sources together was an extremely challenging and time consuming process. Profootball Focus data is very guarded and unavailable for extraction and manipulation. I worked with the PFF support staff to get access to the data, however the information I needed was spread over many different worksheets and datasets. Aggregating and organizing all these datasets together presented a unique set of challenges, and a lot of cleaning, name normalization and general work with the Regular Expression package in python. Merging the data from PFF with the information from Football reference was even more difficult. Ten years worth of draft information leads to many duplicate names, mismatched abbreviated names, and inconsistently punctuated names for the ones with hyphens, periods and apostrophes in them.

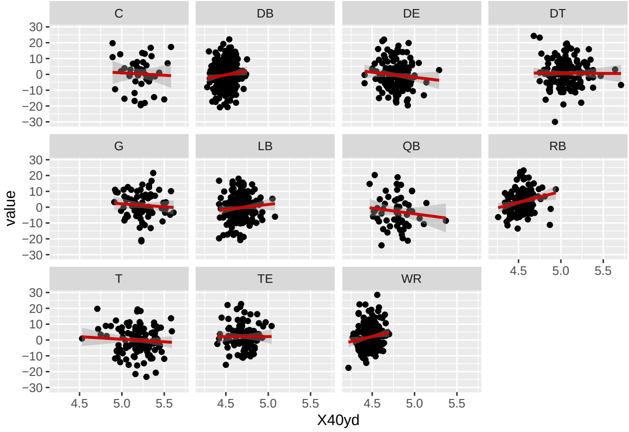
The data had no missing values for Round, Position, Team, or Player grade, but the draft measurements were a little spottier. Draft prospects will sometimes decline to participate in an event, and as a result somewhere between 10% to 15% of entries had missing rows for the continuous variables I used. There were a number of players in the dataset who declined to participate in any of these drills. Since these players were missing so many columns, I excluded them from the dataset. For the remaining players that were missing values, simply taking the population averages would not be a good reflection because there is too much difference between players from differnt positions for these different drills. For example, I would not expect an offensive lineman to be nearly as fast in the 40yd dash as a defensive back, just like I would not expect a defensive back to be nearly as many bench reps as the average offensive lineman. Similarly, I did not think assigning players values based on players that "look" like them with similar draft rounds, heights and weights would solve this problem either. In my opinion, the best option for populating these values was to take the mean for each column based on the players position.

After cleaning the data to the point it was ready to be used for analysis, I performed a general EDA. A histogram revealed that my response variable player grade has a very even distribution as shown below.



My response variable is "value" and is the delta between a players player grade and their expected player grade based upon draft position. It is a continuous variable that should have an average of roughly 0. From my initial exploration, it did not appear that it would require a transformation. For my independent variables, I factored Round, Team and Position as these three are categorical. Additionally, I mean centered 40yard dash time, bench press reps, height and weight before putting them into a model to help with interpretation.

One of the most interesting graphs I found in my EDA was the relationship between value and 40yard dash time, binned by position. The best fit lines for each position do not appear to be consistent. Defensive Backs, Wide Receivers, and running backs all have lines with increasing slopes for a slower 40 yard dash time while Defensive End and Quarterback have decreasing slopes with a slower 40 yard dash time. Although it will require further analysis to discover if this is statistically significant, it appears that speed could be overvalued for Defensive Backs, Running Backs and Receivers while it is undervalued for Quarterbacks and Defensive ends.



IV. Model

I attempted quite a few iterations of different linear models, both hierarchical and multiple linear regression. I used a baseline model with all variables, and the interaction between position and each of my continuous variables. Ultimately, I believe the multiple linear regression model shown below gave me the best results and fewest problems with model assumptions. The equation and output of my final model can be seen below:

$$y_i = \beta_0 + \beta_1 Team + \beta_2 Position + \beta_3 Round + \beta_4 HeightCentered + \beta_5 Round + \beta_6 40yd + \beta_7 Position : 40yd$$

term	estimate	std.error	statistic	p.value
(Intercept)	1.6825675	4.1695319	0.4035387	0.6866117
PositionDB	0.8592044	4.3197295	0.1989024	0.8423671
PositionDE	-1.3547872	4.0474302	-0.3347278	0.7378791
PositionDT	-0.8542322	4.2138886	-0.2027183	0.8393838
PositionG	2.0111930	5.0132987	0.4011716	0.6883532
PositionLB	-1.6107616	4.0601151	-0.3967281	0.6916266
PositionQB	-2.7783183	4.1613609	-0.6676466	0.5044659
PositionRB	4.6055449	4.2756567	1.0771550	0.2815909
PositionT	-0.7451272	4.4329527	-0.1680882	0.8665374
PositionTE	1.4676885	4.0925428	0.3586251	0.7199280
PositionWR	4.9898719	4.4711061	1.1160263	0.2645966
HtCent	-0.2465943	0.1345352	-1.8329356	0.0670179
X40ydCent	-2.9745174	7.8375461	-0.3795215	0.7043564
BenchCent	0.1334812	0.0504323	2.6467383	0.0082156
TeamBears	-1.6936035	1.6787088	-1.0088727	0.3132047
TeamBengals	-1.0029824	1.5687191	-0.6393639	0.5226876
TeamBills	-1.1501131	1.5870113	-0.7247038	0.4687511
TeamBroncos	-0.3368331	1.5733880	-0.2140814	0.8305138
TeamBrowns	-2.8879138	1.4849420	-1.9447990	0.0519931
TeamBuccaneers	-1.8678097	1.6276786	-1.1475298	0.2513528
TeamCardinals	-3.4927888	1.6159109	-2.1614984	0.0308208
TeamChargers	-0.6018302	1.6856166	-0.3570386	0.7211151
TeamChiefs	0.1915309	1.6459255	0.1163667	0.9073781
TeamColts	0.8931148	1.5695304	0.5690331	0.5694221
TeamCowboys	1.8266025	1.6039523	1.1388135	0.2549698
TeamDolphins	-1.6959992	1.6046159	-1.0569503	0.2907111
TeamEagles	-0.4644459	1.5813313	-0.2937056	0.7690250
TeamFalcons	-0.2642246	1.6656005	-0.1586362	0.8739777
TeamGiants	-1.1179469	1.6288367	-0.6863468	0.4926046
TeamJaguars	-2.5762593	1.7612760	-1.4627233	0.1437607
TeamJets	-1.3652879	1.6026064	-0.8519171	0.3944013
TeamLions	-0.9961734	1.6066177	-0.6200438	0.5353268
TeamPackers	0.1024773	1.5389583	0.0665887	0.9469183
TeamPanthers	0.9178259	1.7144451	0.5353487	0.5924912
TeamPatriots	2.6456678	1.6879342	1.5673998	0.1172404
TeamRaiders	-1.3082693	1.5605769	-0.8383242	0.4019874
TeamRams	-0.3871655	1.6593183	-0.2333281	0.8155397
TeamRavens	0.7630752	1.5367494	0.4965515	0.6195810
TeamRedskins	-0.1391923	1.5863248	-0.0877452	0.9300914
TeamSaints	2.2765581	1.7744634	1.2829558	0.1997134
TeamSeahawks	1.0403877	1.5870962	0.6555291	0.5122316
TeamSteelers	0.7798514	1.6074052	0.4851617	0.6276352
TeamTexans	-1.2779460	1.5782827	-0.8097067	0.4182422
TeamTitans	-0.5597203	1.6040174	-0.3489490	0.7271785
TeamVikings	1.0292604	1.5975281	0.6442832	0.5194942
PositionDB:X40ydCent	12.5661839	9.6308579	1.3047834	0.1921745
PositionDE:X40ydCent	-3.1403682	9.2913151	-0.3379896	0.7354201
PositionDT:X40ydCent	1.9165045	8.7878971	0.2180845	0.8273941
PositionG:X40ydCent	-1.5522632	9.7504148	-0.1591997	0.8735338
PositionLB:X40ydCent	8.1014677	9.3845085	0.8632810	0.3881264
PositionQB:X40ydCent	-3.6466574	10.0233068	-0.3638178	0.7160473
PositionRB:X40ydCent	17.8107793	9.8807723	1.8025696	0.0716643
PositionT:X40ydCent	2.3028428	8.8255986	0.2609277	0.7941855
PositionTE:X40ydCent	3.3401127	10.2085893	0.3271865	0.7435742
PositionWR:X40ydCent	17.5941558	10.2646517	1.7140529	0.0867336

This final model has an R-square of .08. This may seem low, but I would expect to be able to identify strong predictors of value with the amount of luck, variance, intangibles, and other factors not captured in this study that go into the deviance between a players performance and expected performance. I arrived at this final model by comparing R-squares, as well as checking assumptions, running a stepwise AIC test in both directions, and comparing lots of models with ANOVA tests. The model that generated the lowest AIC contained the independent variables of Position, Bench, Broad Jump and 3Cone. When I used ANOVA tests to check for significance in these variables and the ones excluded however, I reached different results. Changing only one interaction or variable at a time from a baseline model to a model without the variable I was testing, I found that the model did not think broad jump, or 3Cone were significant variables from my AIC generated model. Conversely, it found 40 yard, Height, and the interaction between Position and 40 yard to be significant. An ANOVA test comparing a model to a model without Team found the p-value to be .07. Although it may not be significant at the .05 level, one of the primary questions I wanted to answer is if certain teams have beer better at identifying value at a statistically significant level, so I kept the variable.

The intercept of my model carries the assumption of a player who plays at the Center position for the 49ers with an average number of Bench reps, height and 40yard dash time. For this player, my model predicts a value of 1.7.

Ultimately, only bench press reps, and being drafted by the Arizona Cardinals are significant at the 0.05 level. However there are a handful of variables that are extremely close to having a p-score of .05 or lower.

Bench press has a coefficient of 0.13 and is statistically significant with a p-score of 0.01. In the context of this model, this means that if all other variables are held constant, every additional bench press rep a player does at the combine increases their expected value by 0.13 points. This is an interesting finding because it implies strength is an under-appreciated metric when evaluating players. Although I do not know how NFL teams evaluate players, from the media coverage of the draft combine it seems that a lot of attention is given to a player's athleticism in terms of their high, jumping ability, and 40 yard dash speed. It could be that with all the focus on these flashier traits, how strong a player is can get overlooked and become underappreciated.

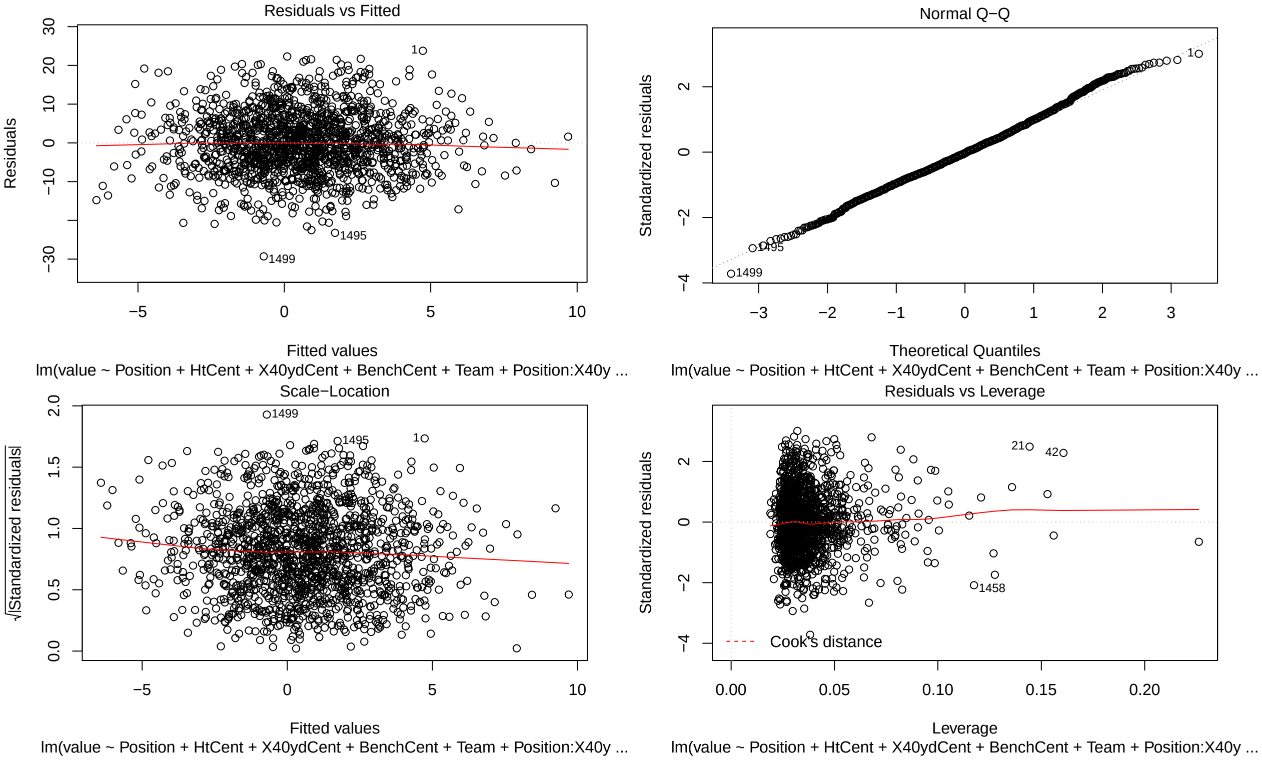
Being drafted by the Arizona Cardinals has a coefficient of -3.49 with a statistically significant p-score. This means that holding all other things equal, being drafted by the Cardinals reduces your expected value by 3.5 points. This carries the implication that the Cardinals are especially poor at drafting and developing players. Although the Cardinals have not been a terrible franchise from 2010 to 2019, they have certainly been below average and much of their success has come as a result of players they acquired as a result of free agency rather than the draft. It is worth noting that the Cleveland Browns have a coefficient of -2.88 and just missed out on statistical significance with a pscore of 0.052.

Although insignificant at the .05 level, the interactions between 40 yard dash and running back and wide receiver barely missed the cut. They have coefficients of 17.81 and 17.59 respectively meaning that a 1 second increase in a running backs 40 yard dash speed would result in an expected increase in player value of 17.81, and an increase in player value of 17.59 for receivers. Although this initially appears quite large, a one second increase in 40 yard dash speed at these positions makes a massive difference. Speed is one of the flashiest traits at these positions, and every single year a player's draft stock increases dramatically when they run a fantastic 40 yard dash. However, speed is just one aspect that goes into a players ability, and it does not surprise me that it receives far too much focus when a player is being evaluated.

The 95% confidence intervals for the Cardinals and Bench reps can be seen below. The Cardinals ranges from -6.66 to -0.33. This means that we are 95% confident the true effect of being drafted by the cardinals has on player value falls between -6.66 to -0.33. The Bench coefficient spans from 0.03 to 0.23, meaning that we are 95% confident the true effect of every additional bench rep is an increase in value of 0.03 to 0.23

	2.5 %	97.5 %
TeamCardinals	-6.66	-0.3257
BenchCent	0.03464	0.2323

There are no violations of linearity, independence, normality, or equal variance in my model. The residuals vs fitted plot is randomly distributed, almost all the points on the Normal Q-Q lie exactly on the 45 degree line, and the standardized residual plot is random as well. From the Residuals vs Leverage graph there appear to be no high leverage points with no points falling outside of the cook's distance line. The VIF scores are all below 5 except for the positions, and 40yd dash times which are all extremely high. This is expected as the interaction term of Position and 40yd will influince this.



```
## NULL
```

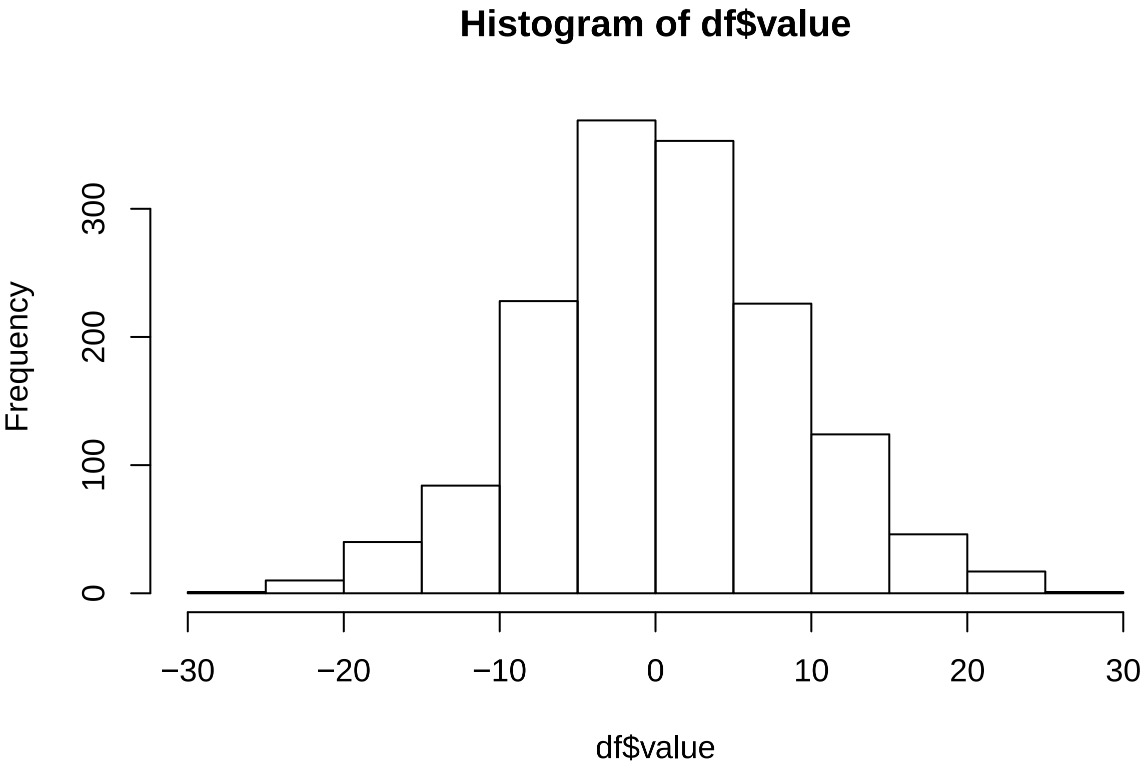
V. Conclusion

This study was attempting to identify what measurables go into finding value in the draft, and what teams are significantly better or worse at identifying and developing talent. Ultimately, my model identified answers to both of these questions. The Arizona Cardinals are worse than the rest of the NFL teams at finding and drafting talent at a statistically significant level, and the Browns are not far off. Strength (as measured by bench press reps) is an undervalued attribute in the draft that appears to go somewhat overlooked by teams. Conversely, although it is not statistically significant, speed (as measured by the 40 yard dash) tends to be overvalued when evaluating running backs and wide receivers. These findings could potentially be of use to NFL teams when evaluating players.

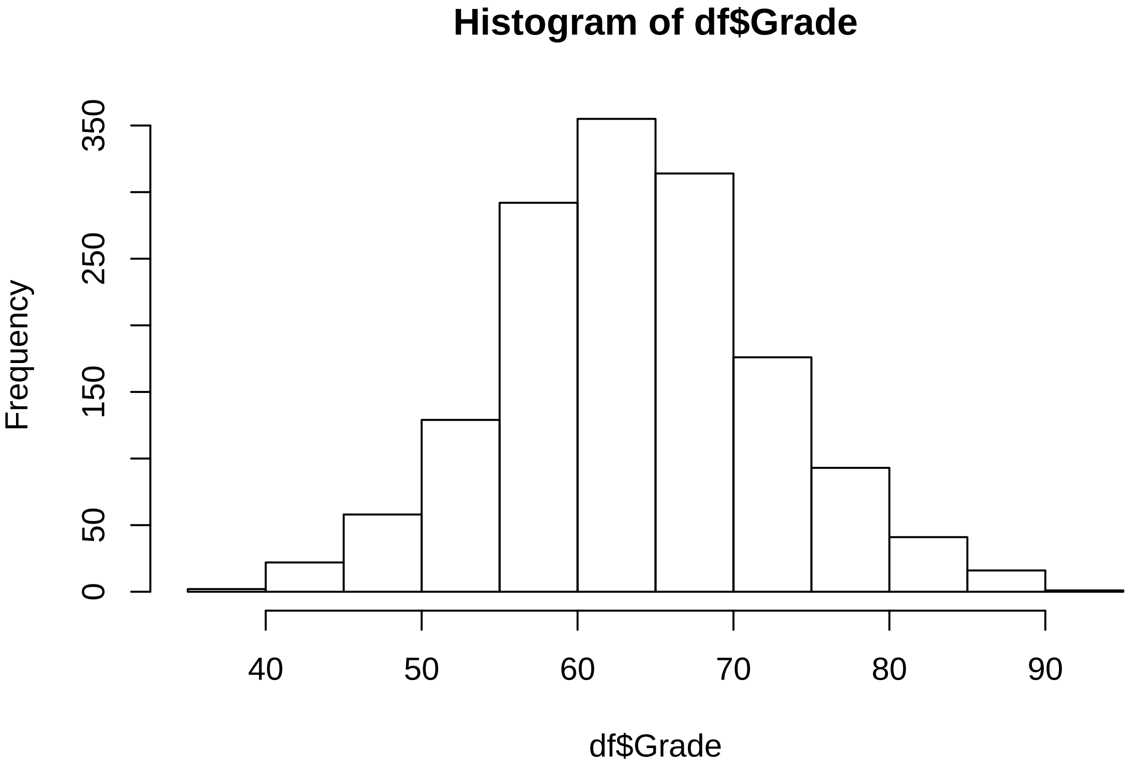
There are limitations to this study however. The findings in this study are based on player grade data, and although this data is highly regarded, it is ultimately the aggregated opinions of a few people, and are not a perfect representation of how good a player is. While more data is always preferable, this is especially true for an analysis like this that includes a variable like team that has 32 categories. This spreads the data and results even further. Finally, there is far more that goes into evaluating a player before the draft than the variables in this study. Their college performance, and personality, and intangibles all play a role in how teams perceive and select players. My model with an R-squared of .08 reflects this fact that it does not come close to capturing the full story of player evaluation.

APPENDIX

```
hist(df$value)
```



```
hist(df$Grade)
```

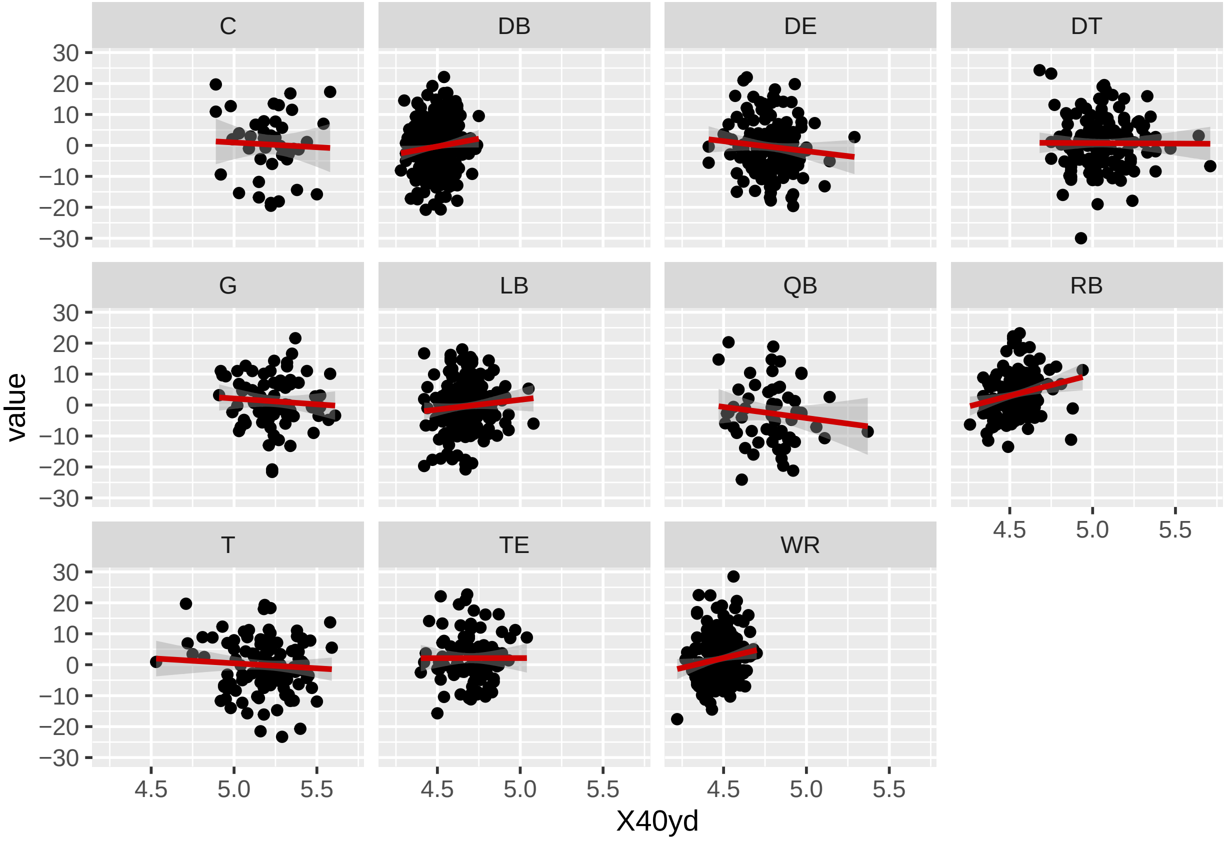


```r
summary(df)
```

```
##        X              index          Unnamed..0        Year      Round
##  Min.   :   0.0   Min.   :   0.0   Min.   :   0   Min.   :2010   1:283
##  1st Qu.: 374.5   1st Qu.: 428.5   1st Qu.: 478   1st Qu.:2012   2:269
##  Median : 749.0   Median : 865.0   Median : 969   Median :2014   3:284
##  Mean   : 749.0   Mean   : 859.7   Mean   : 985   Mean   :2014   4:254
##  3rd Qu.:1123.5   3rd Qu.:1296.5   3rd Qu.:1500   3rd Qu.:2017   5:188
##  Max.   :1498.0   Max.   :1711.0   Max.   :2071   Max.   :2019   6:125
##                                                                  7: 96
##       Pick                  Name           Team         Position
##  Min.   :  1.0   charles brown:   2   Browns :  64   DB     :287
##  1st Qu.: 43.0   chris carter :   2   49ers  :  55   WR     :193
##  Median : 88.0   jordan thomas:   2   Packers:  55   LB     :192
##  Mean   : 97.3   kevin white  :   2   Ravens :  55   DT     :148
##  3rd Qu.:143.0   mike davis   :   2   Bengals:  52   RB     :144
##  Max.   :256.0   mike edwards :   2   Raiders:  52   DE     :141
##                  (Other)      :1487   (Other):1166   (Other):394
##    player_id     player_game_count     Grade           count     pickgrade
##  Min.   : 1738   Min.   : 10.00    Min.   :35.50   Min.   :1   Min.   :58.30
##  1st Qu.: 7062   1st Qu.: 23.00    1st Qu.:58.05   1st Qu.:1   1st Qu.:60.70
##  Median : 8762   Median : 41.00    Median :63.60   Median :1   Median :62.10
##  Mean   :13926   Mean   : 49.95    Mean   :63.82   Mean   :1   Mean   :63.24
##  3rd Qu.:11768   3rd Qu.: 70.00    3rd Qu.:69.20   3rd Qu.:1   3rd Qu.:65.40
##  Max.   :66578   Max.   :158.00    Max.   :93.10   Max.   :1   Max.   :72.90
##
##      value                      Player          Pos            Ht
##  Min.   :-30.0000   charles brown:   2   WR     :196   Min.   :66.00
##  1st Qu.: -4.8000   chris carter :   2   CB     :181   1st Qu.:72.00
##  Median :  0.3000   jordan thomas:   2   DT     :137   Median :74.00
##  Mean   :  0.5741   kevin white  :   2   RB     :132   Mean   :73.95
##  3rd Qu.:  5.8000   mike davis   :   2   DE     :127   3rd Qu.:76.00
##  Max.   : 28.5000   mike edwards :   2   OT     :124   Max.   :81.00
##                     (Other)      :1487   (Other):602
##        Wt            X40yd         Vertical         Bench         Broad.Jump
##  Min.   :156.0   Min.   :4.22   Min.   :20.50   Min.   : 4.00   Min.   : 85.0
##  1st Qu.:207.0   1st Qu.:4.51   1st Qu.:30.50   1st Qu.:16.00   1st Qu.:110.0
##  Median :239.0   Median :4.65   Median :33.50   Median :20.67   Median :117.8
##  Mean   :246.6   Mean   :4.74   Mean   :33.21   Mean   :20.98   Mean   :115.9
##  3rd Qu.:293.0   3rd Qu.:4.93   3rd Qu.:35.89   3rd Qu.:25.00   3rd Qu.:122.0
##  Max.   :358.0   Max.   :5.71   Max.   :45.00   Max.   :49.00   Max.   :147.0
##
##      X3Cone          Shuttle      X_merge       X40ydCent
##  Min.   :6.280   Min.   :3.810   both:1499   Min.   :-0.51982
##  1st Qu.:6.917   1st Qu.:4.190               1st Qu.:-0.22982
##  Median :7.070   Median :4.290               Median :-0.08982
##  Mean   :7.186   Mean   :4.362               Mean   : 0.00000
##  3rd Qu.:7.410   3rd Qu.:4.520               3rd Qu.: 0.19018
##  Max.   :8.320   Max.   :5.210               Max.   : 0.97018
##
##    BenchCent            HtCent             WtCent
##  Min.   :-16.9768   Min.   :-7.94596   Min.   :-90.562
##  1st Qu.: -4.9768   1st Qu.:-1.94596   1st Qu.:-39.562
##  Median : -0.3101   Median : 0.05404   Median : -7.562
##  Mean   :  0.0000   Mean   : 0.00000   Mean   :  0.000
##  3rd Qu.:  4.0232   3rd Qu.: 2.05404   3rd Qu.: 46.438
##  Max.   : 28.0232   Max.   : 7.05404   Max.   :111.438
##
```

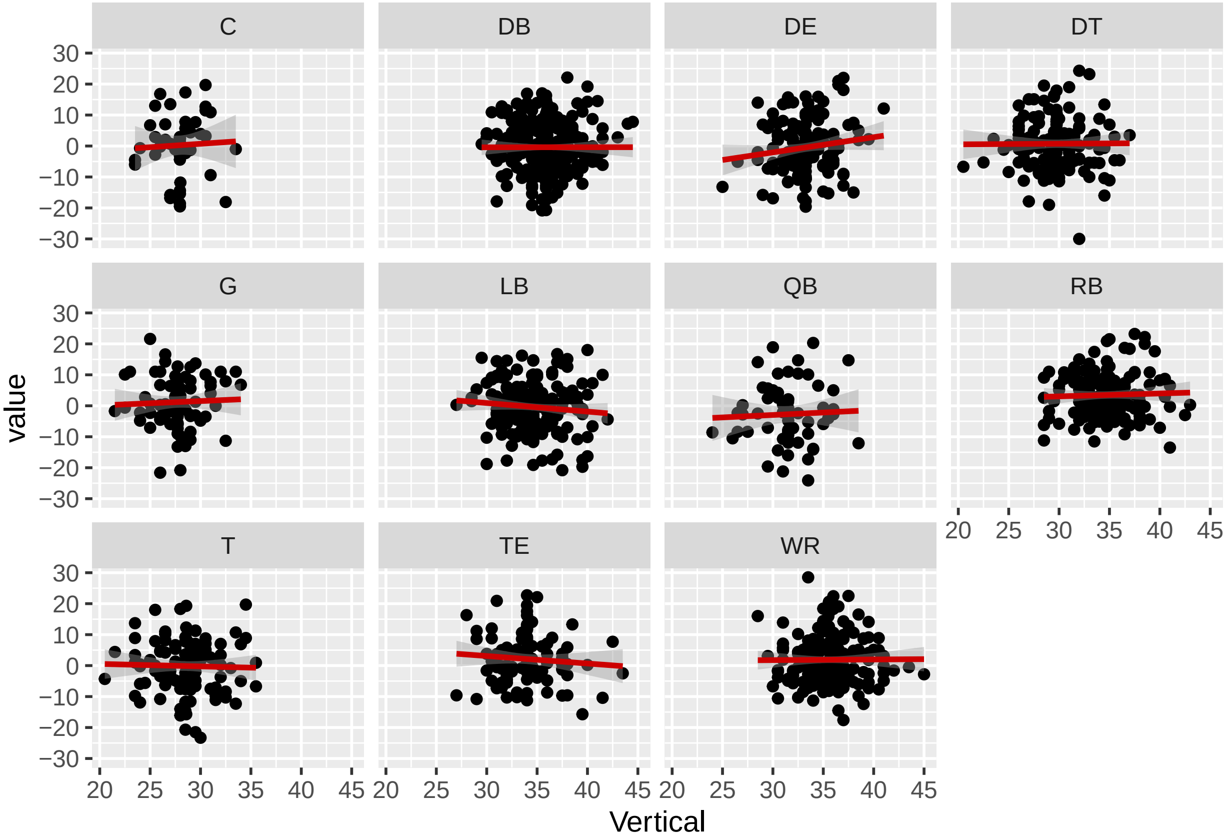
```
### Wrapped by Position ###

ggplot(df, aes(x=X40yd, y=value)) + geom_point() + facet_wrap(~Position) + geom_smooth(method="lm",col=
```

```
## `geom_smooth()` using formula 'y ~ x'
```



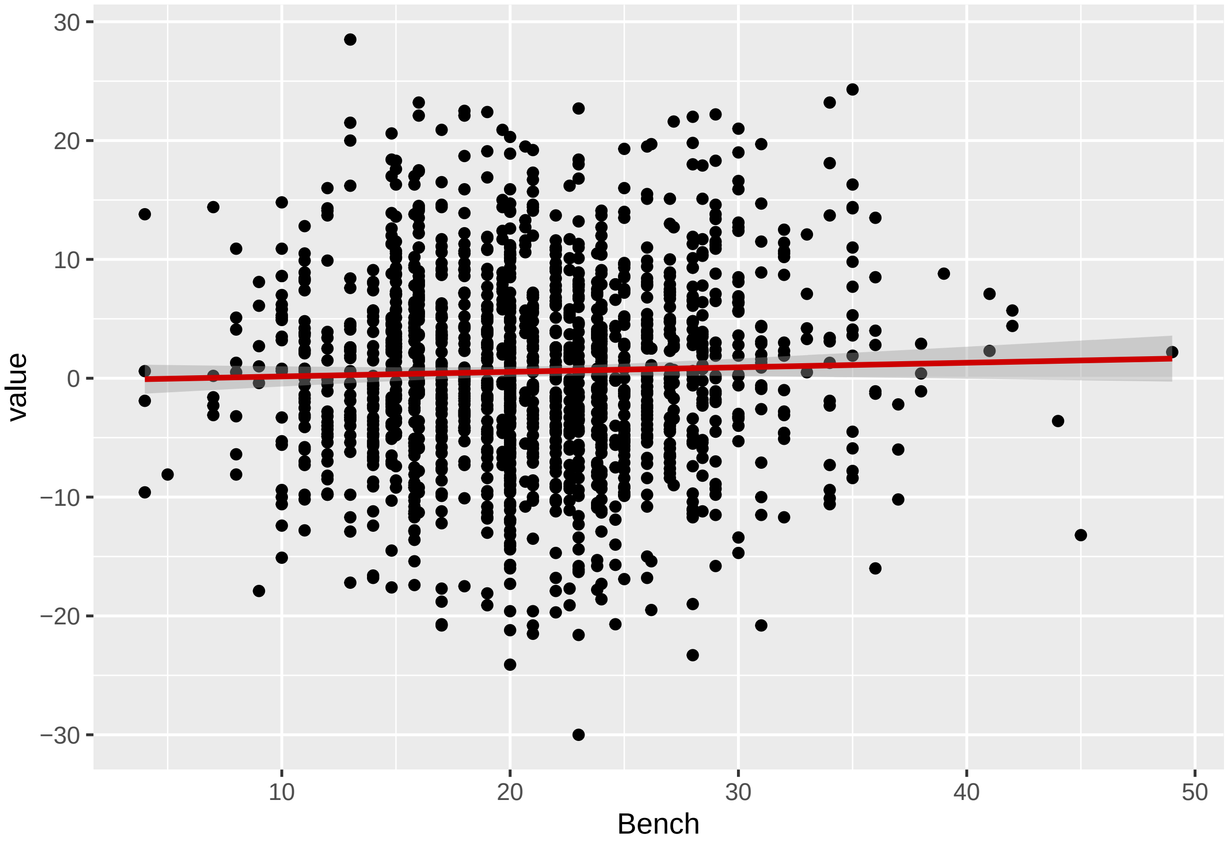
```
ggplot(df, aes(x=Vertical, y=value)) + geom_point() + facet_wrap(~Position) + geom_smooth(method="lm",c
```

```
## `geom_smooth()` using formula 'y ~ x'
```



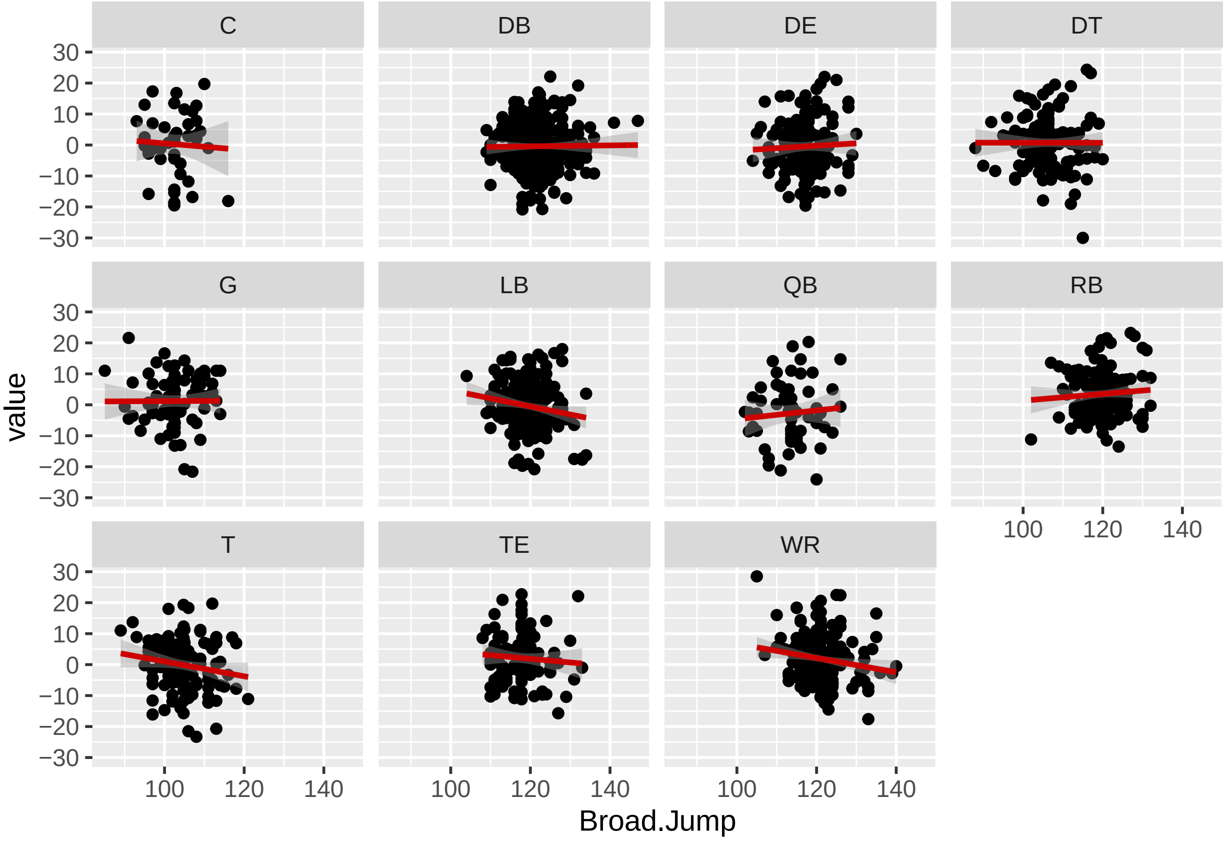
```r
ggplot(df, aes(x=Bench, y=value)) + geom_point() + geom_smooth(method="lm",col="red3")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



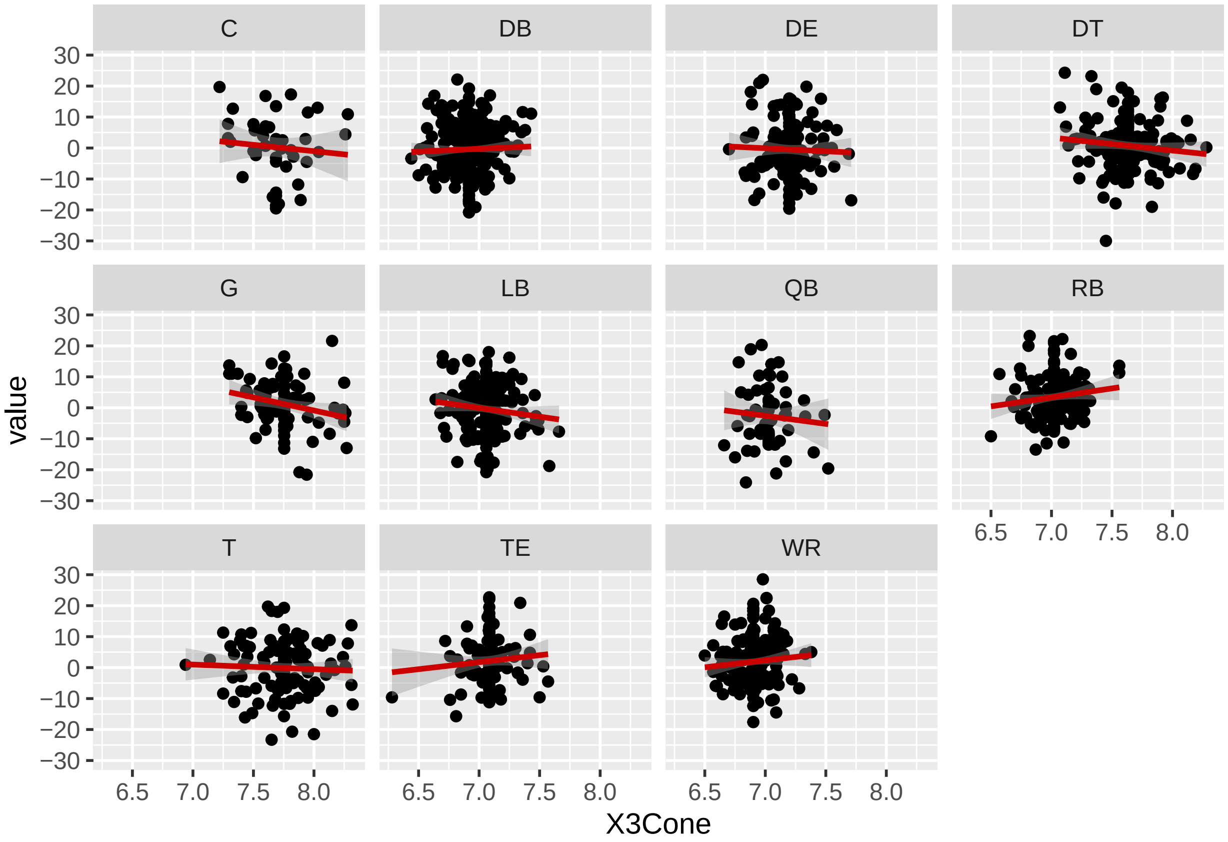
```r
ggplot(df, aes(x=Broad.Jump, y=value)) + geom_point() + facet_wrap(~Position) + geom_smooth(method="lm"
```

```
## `geom_smooth()` using formula 'y ~ x'
```



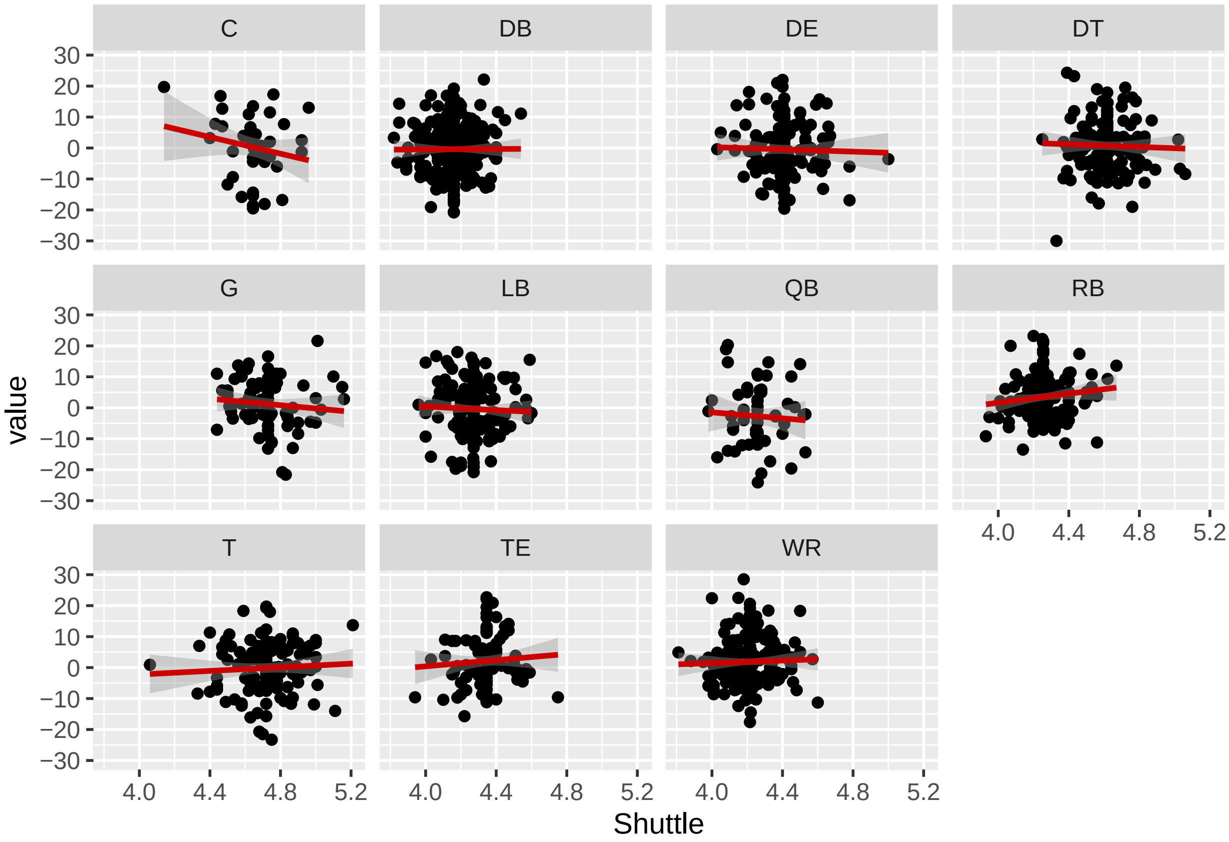
```
ggplot(df, aes(x=X3Cone, y=value)) + geom_point() + facet_wrap(~Position) + geom_smooth(method="lm",col=
```

```
## `geom_smooth()` using formula 'y ~ x'
```



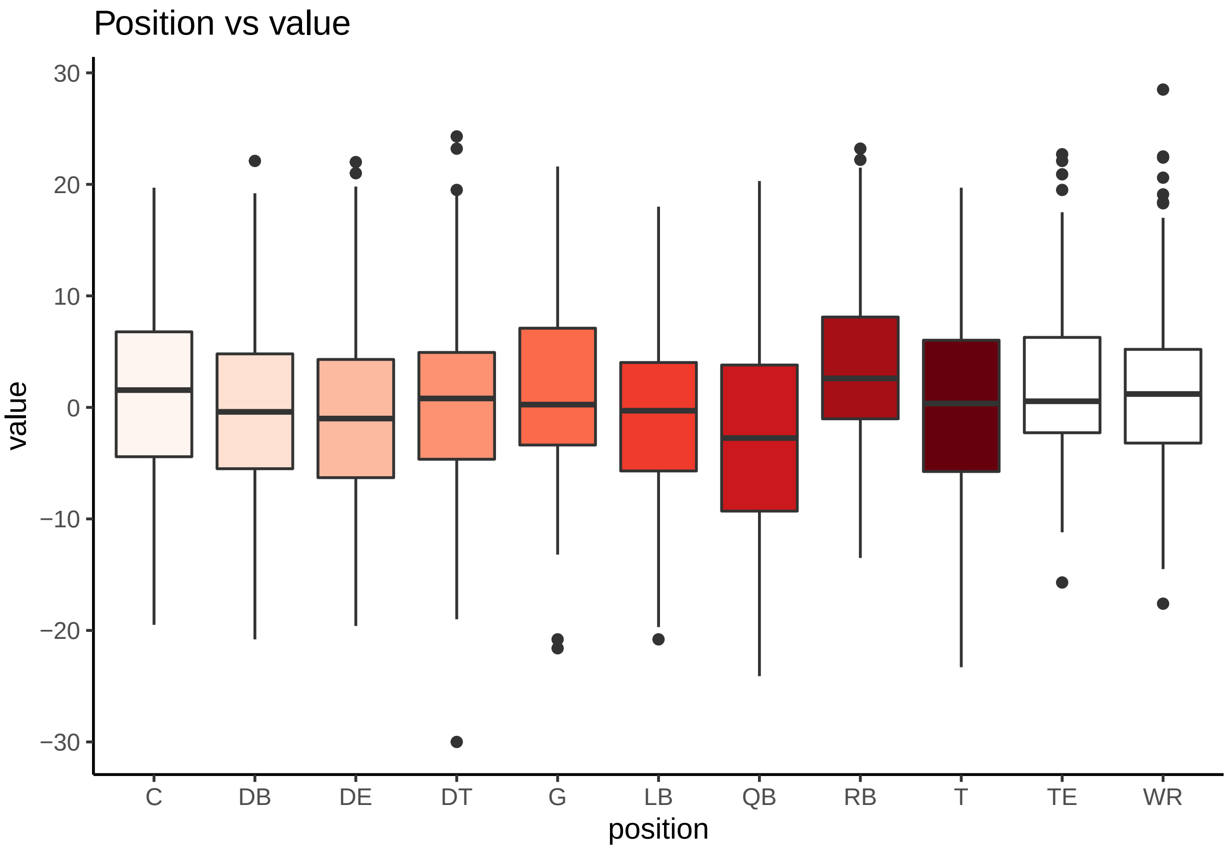
```r
ggplot(df, aes(x=Shuttle, y=value)) + geom_point() + facet_wrap(~Position) + geom_smooth(method="lm",col
```

```
## `geom_smooth()` using formula 'y ~ x'
```



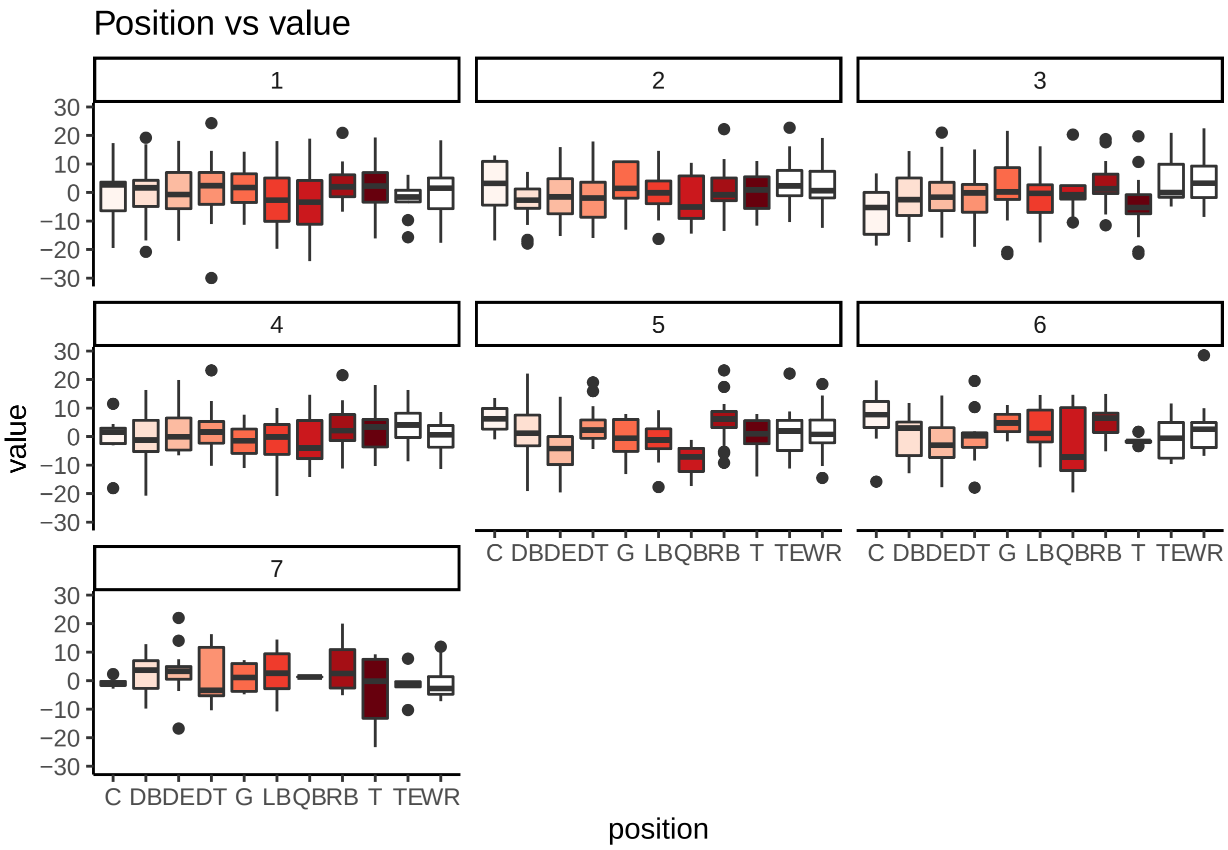
```
### Position EDA

ggplot(df, aes(x=Position, y=value, fill=Position)) + geom_boxplot() + scale_fill_brewer(palette="Reds"
```

```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Reds is 9
## Returning the palette you asked for with that many colors
```



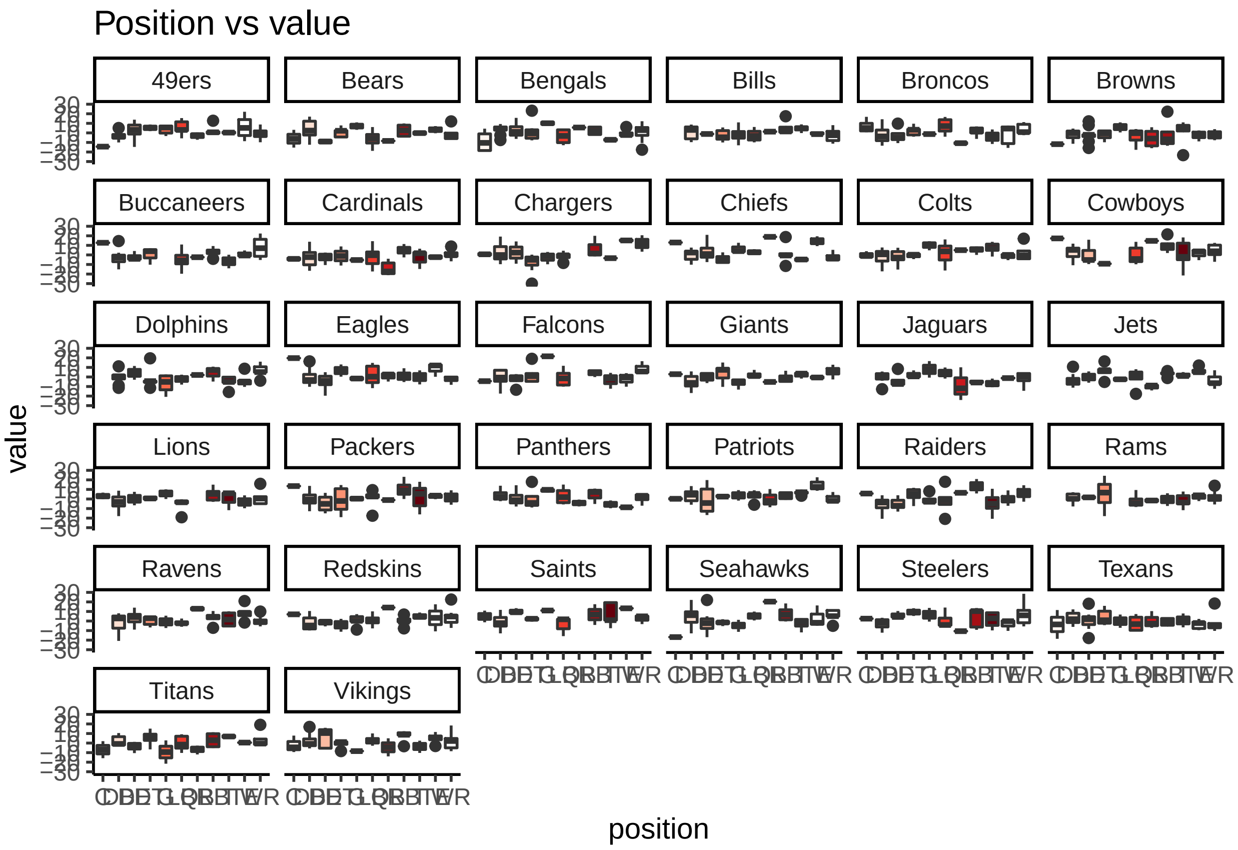
```
ggplot(df, aes(x=Position, y=value, fill=Position)) + geom_boxplot() + scale_fill_brewer(palette="Reds"
```

```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Reds is 9
## Returning the palette you asked for with that many colors
```

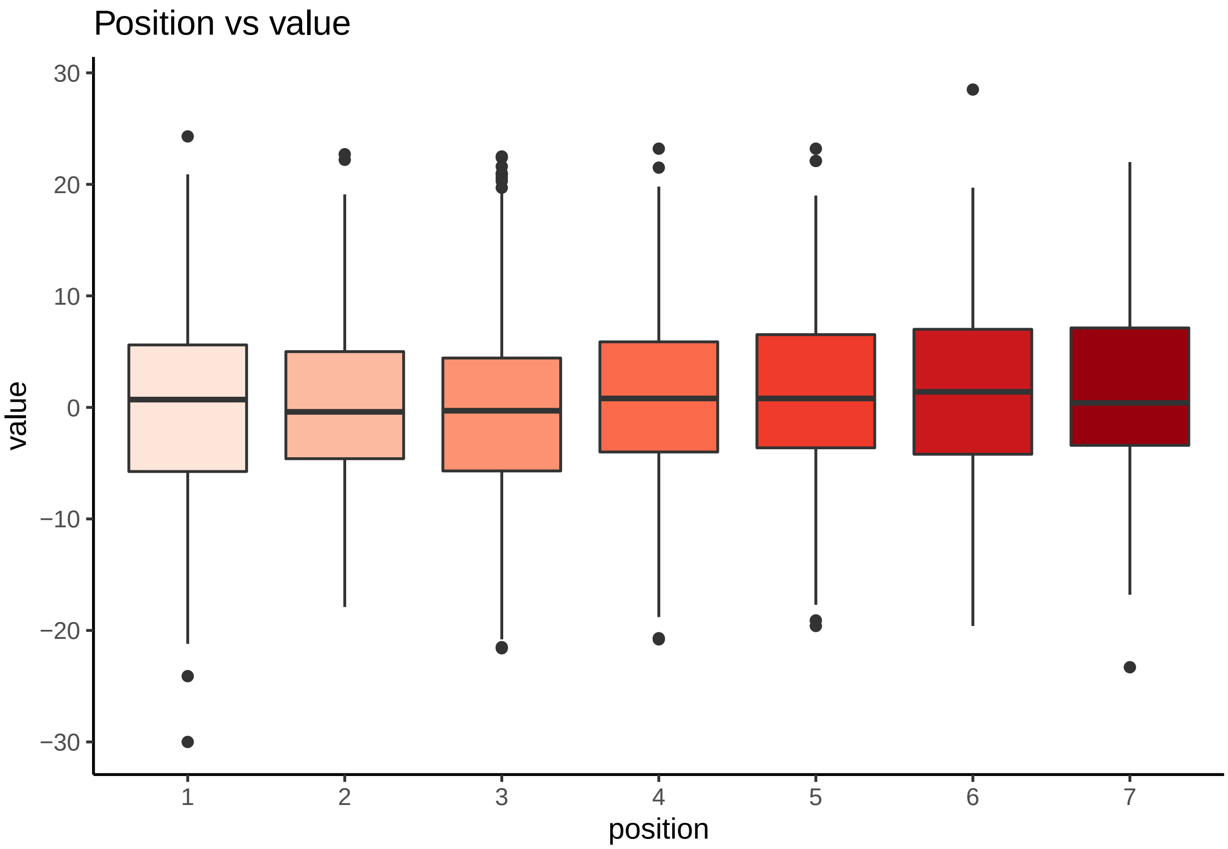


```
ggplot(df, aes(x=Position, y=value, fill=Position)) + geom_boxplot() + scale_fill_brewer(palette="Reds"
```

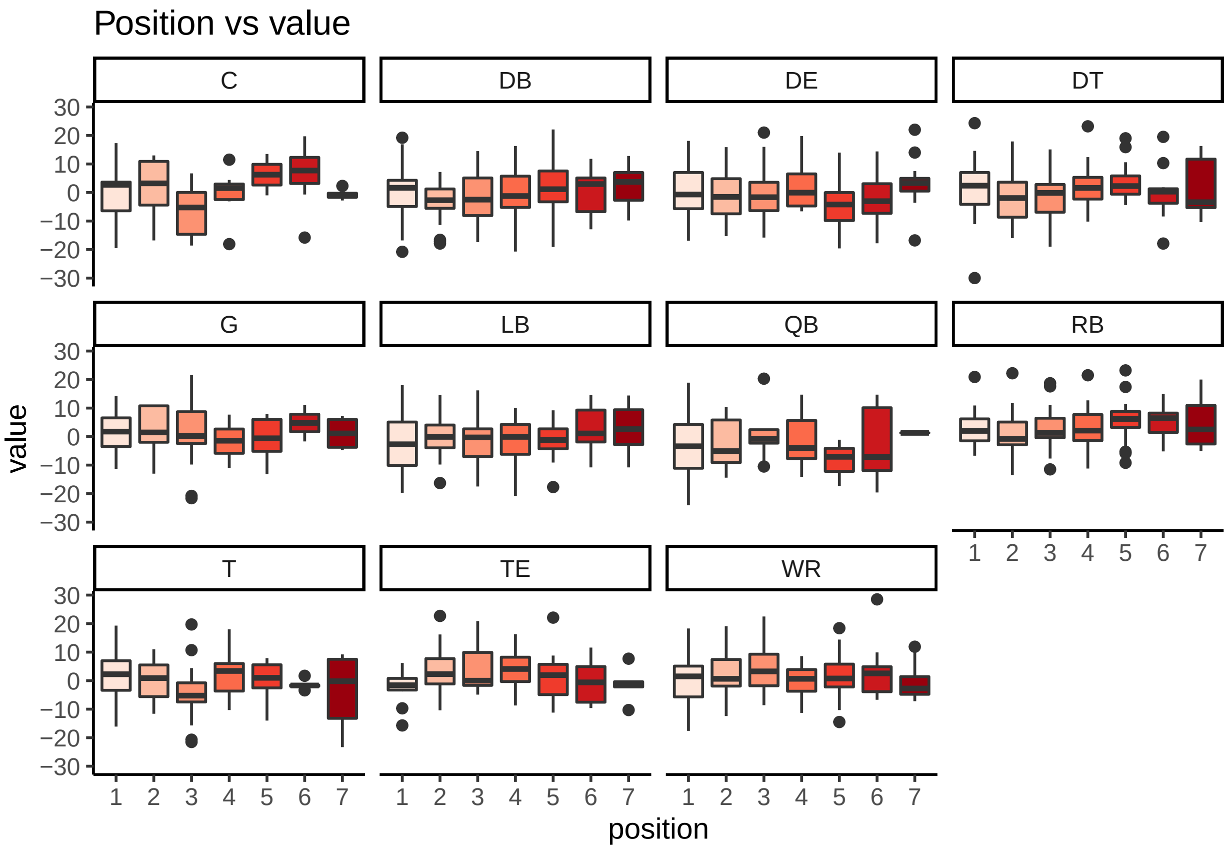
```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Reds is 9
## Returning the palette you asked for with that many colors
```



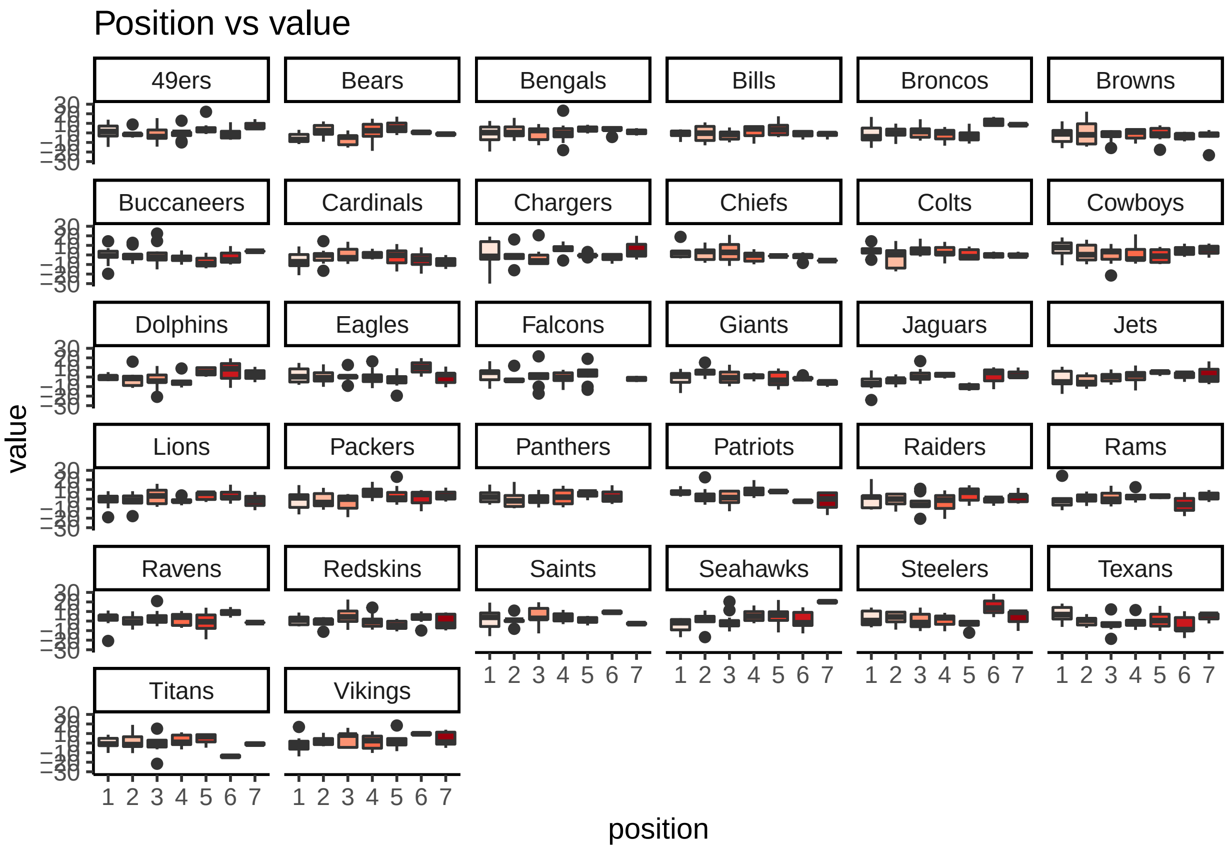
```
ggplot(df, aes(x=Round, y=value, fill=Round)) + geom_boxplot() + scale_fill_brewer(palette="Reds") +  la
```



```
### ROUND EDA
ggplot(df, aes(x=Round, y=value, fill=Round)) + geom_boxplot() + scale_fill_brewer(palette="Reds") +  la
```

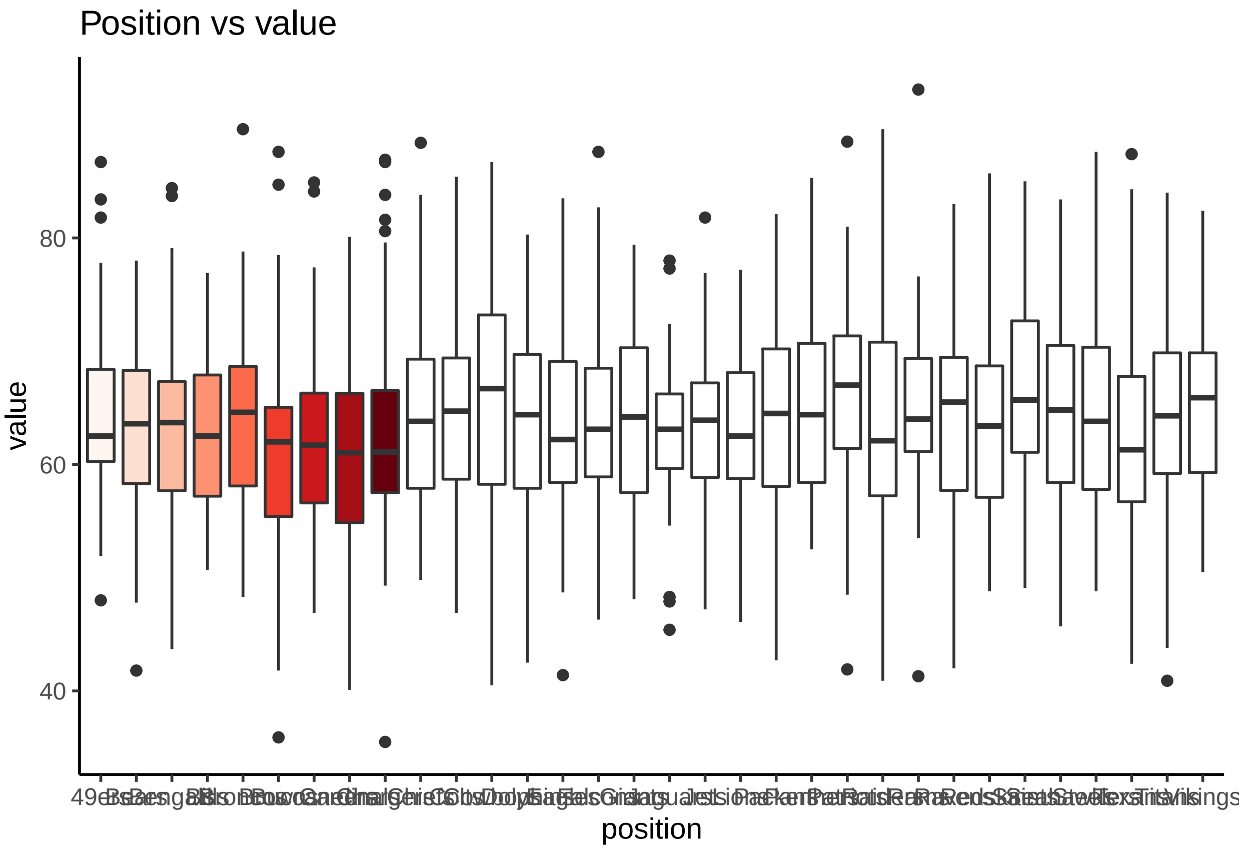


```
ggplot(df, aes(x=Round, y=value, fill=Round)) + geom_boxplot() + scale_fill_brewer(palette="Reds") +  la
```



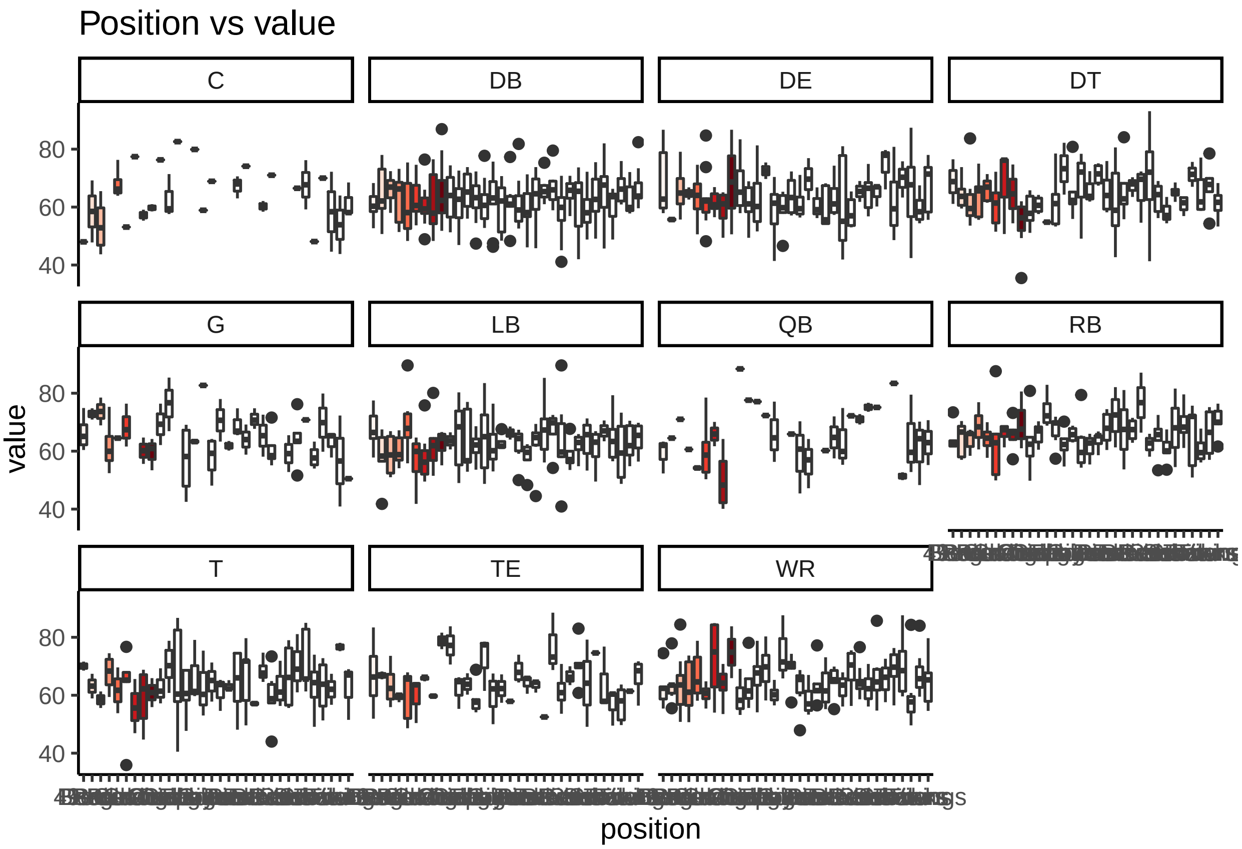
```
### TEAM
ggplot(df, aes(x=Team, y=Grade, fill=Team)) + geom_boxplot() + scale_fill_brewer(palette="Reds") +  labs
```

```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Reds is 9
## Returning the palette you asked for with that many colors
```



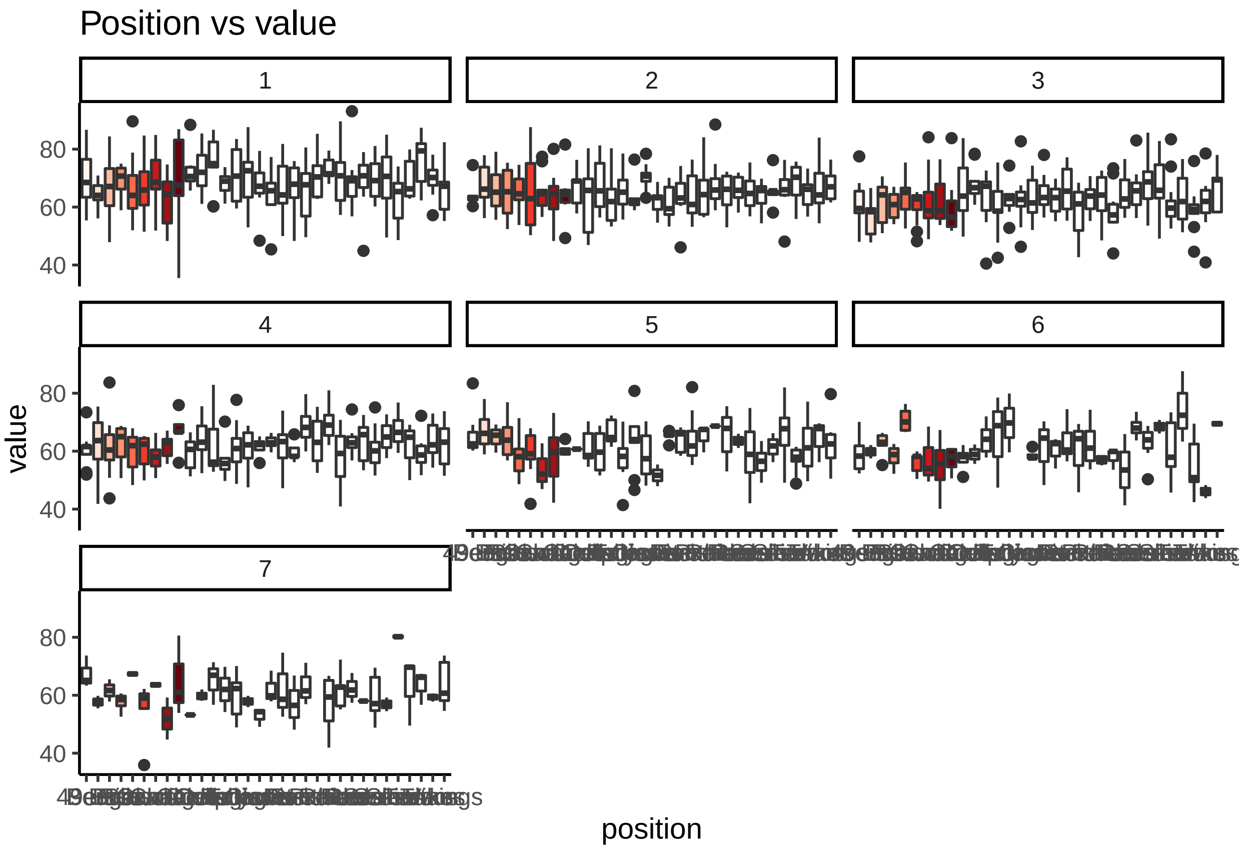
```
ggplot(df, aes(x=Team, y=Grade, fill=Team)) + geom_boxplot() + scale_fill_brewer(palette="Reds") +  lab
```

```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Reds is 9
## Returning the palette you asked for with that many colors
```



```
ggplot(df, aes(x=Team, y=Grade, fill=Team)) + geom_boxplot() + scale_fill_brewer(palette="Reds") +  lab
```

```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Reds is 9
## Returning the palette you asked for with that many colors
```



```
model1 <- lmer(Grade ~ Ht + Wt +X40yd + Vertical + Bench + Broad.Jump +X3Cone + Shuttle + Round + Team +
```

```
## boundary (singular) fit: see ?isSingular
```

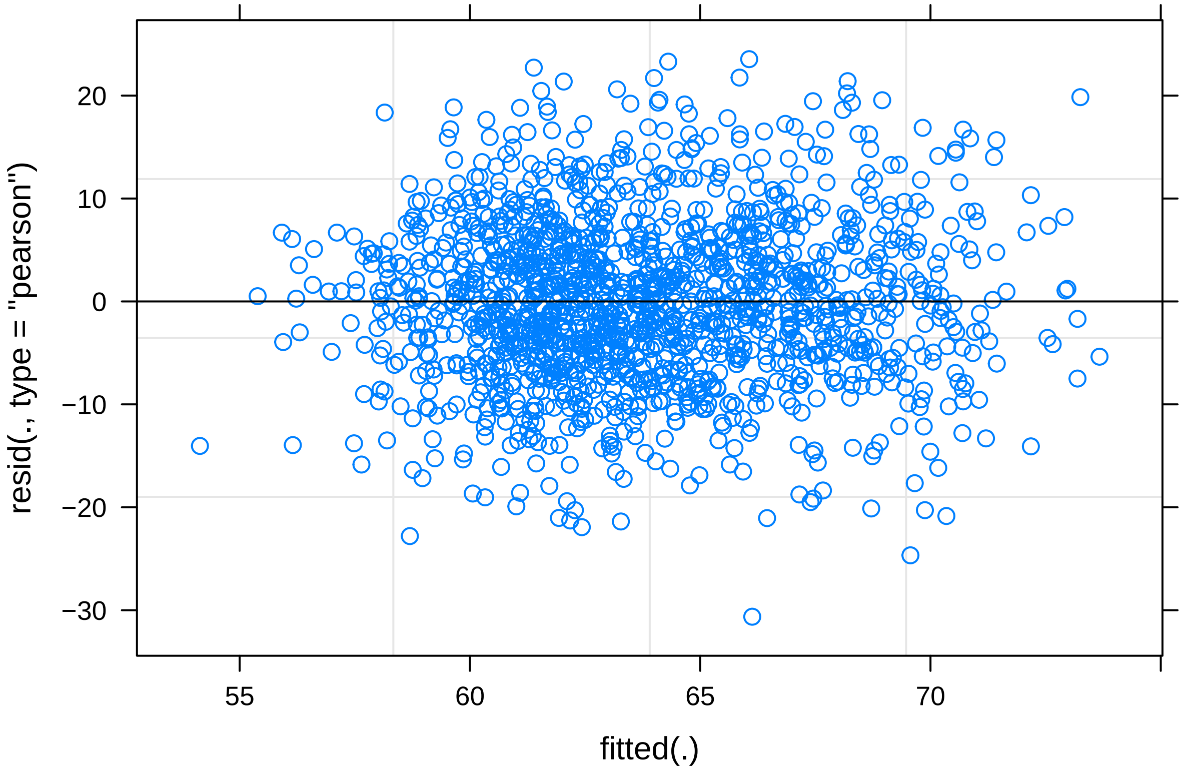
```
summary(model1)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Grade ~ Ht + Wt + X40yd + Vertical + Bench + Broad.Jump + X3Cone +
##     Shuttle + Round + Team + (Ht + Wt + X40yd + Vertical + Bench +
##     Broad.Jump + X3Cone + Shuttle | Position)
##    Data: df
##
## REML criterion at convergence: 10438.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7901 -0.6342 -0.0365  0.6233  2.9124
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  Position (Intercept) 6.505e+01 8.06532
##           Ht          1.212e-01 0.34811   0.72
##           Wt          7.407e-04 0.02722  -0.97 -0.88
##           X40yd       4.261e+00 2.06427  -0.36 -0.53  0.45
##           Vertical    3.544e-03 0.05953   0.63  0.72 -0.70 -0.93
##           Bench       2.765e-02 0.16629   0.72  0.28 -0.61 -0.01  0.19
##           Broad.Jump  2.186e-02 0.14785  -0.72 -0.93  0.85  0.64 -0.82 -0.15
##           X3Cone      5.419e+00 2.32794  -0.50 -0.24  0.44  0.21 -0.32 -0.66
##           Shuttle     1.658e+01 4.07167    -0.03 -0.19  0.09 -0.52  0.40 -0.02
##  Residual             6.531e+01 8.08151
##
##
##
##
##
##
##
##
##   0.13
##   0.05 -0.56
##
## Number of obs: 1499, groups:  Position, 11
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)    95.23495   17.04351   5.588
## Ht             -0.19264    0.18204  -1.058
## Wt              0.01647    0.02001   0.823
## X40yd          -0.62744    2.15360  -0.291
## Vertical        0.08204    0.10756   0.763
## Bench           0.12770    0.07571   1.687
## Broad.Jump     -0.07480    0.07177  -1.042
## X3Cone         -2.52059    1.66443  -1.514
## Shuttle         1.91683    2.47666   0.774
## Round2         -3.36550    0.70102  -4.801
## Round3         -5.89138    0.69758  -8.446
## Round4         -6.87775    0.72883  -9.437
## Round5         -6.85234    0.79689  -8.599
## Round6         -8.17867    0.90210  -9.066
## Round7         -7.94383    0.98290  -8.082
## TeamBears      -1.23235    1.69295  -0.728
## TeamBengals    -1.14861    1.58237  -0.726
## TeamBills      -1.02846    1.60299  -0.642
## TeamBroncos    -0.21067    1.58222  -0.133
## TeamBrowns     -2.60913    1.49605  -1.744
## TeamBuccaneers -1.54005    1.63853  -0.940
## TeamCardinals  -3.62272    1.62722  -2.226
## TeamChargers   -0.69584    1.69248  -0.411
## TeamChiefs      0.46651    1.66505   0.280
## TeamColts       0.81894    1.58289   0.517
## TeamCowboys     1.90959    1.61664   1.181
## TeamDolphins   -1.24050    1.61741  -0.767
## TeamEagles     -0.40855    1.59225  -0.257
## TeamFalcons    -0.30832    1.68311  -0.183
## TeamGiants     -0.94318    1.64308  -0.574
## TeamJaguars    -1.74282    1.77061  -0.984
## TeamJets       -0.86754    1.61840  -0.536
## TeamLions      -0.68279    1.61667  -0.422
## TeamPackers    -0.71037    1.55689  -0.456
## TeamPanthers    1.03211    1.72893   0.597
## TeamPatriots    2.30607    1.69465   1.361
## TeamRaiders    -0.94031    1.57598  -0.597
## TeamRams         0.26045    1.67416   0.156
## TeamRavens       0.47749    1.55314   0.307
## TeamRedskins     0.22709    1.59808   0.142
## TeamSaints       1.84364    1.77624   1.038
## TeamSeahawks     1.19544    1.60132   0.747
## TeamSteelers     0.33546    1.61762   0.207
## TeamTexans      -1.43148    1.58794  -0.901
## TeamTitans      -0.16214    1.61875  -0.100
## TeamVikings      0.63928    1.60887   0.397
```

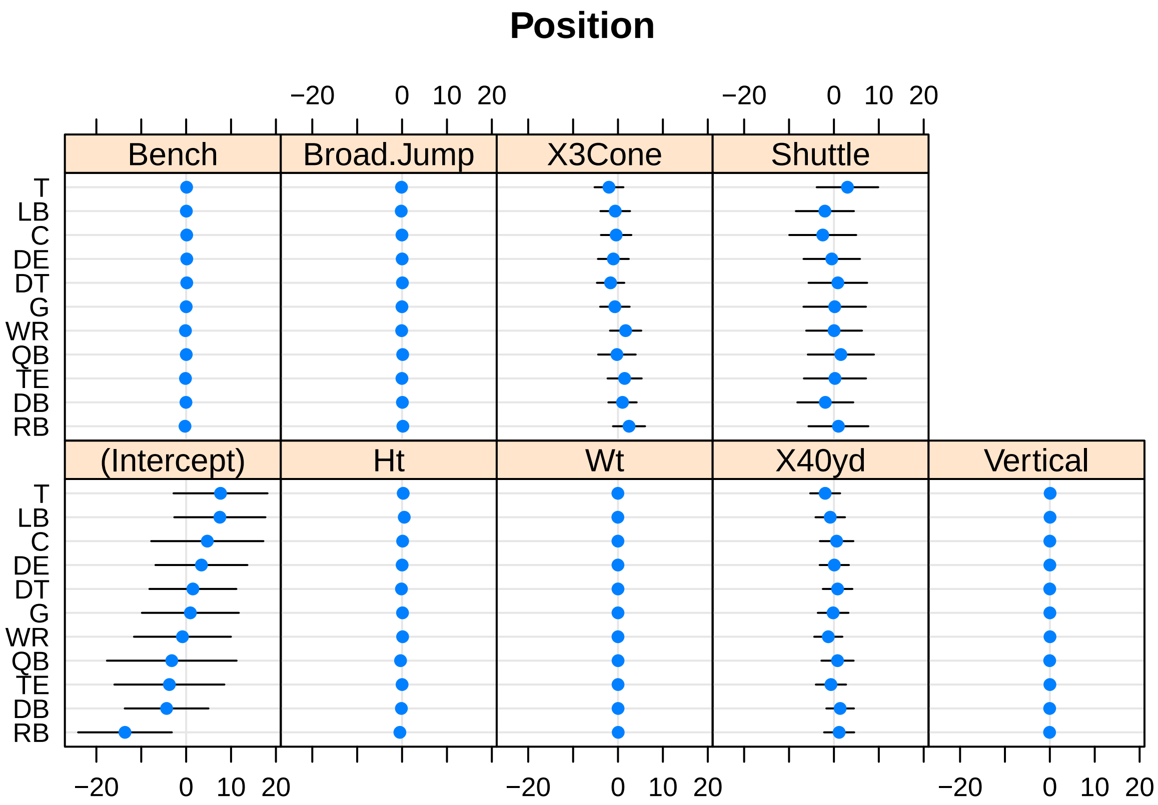
```
##
## Correlation matrix not shown by default, as p = 46 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```
plot(model1)
```



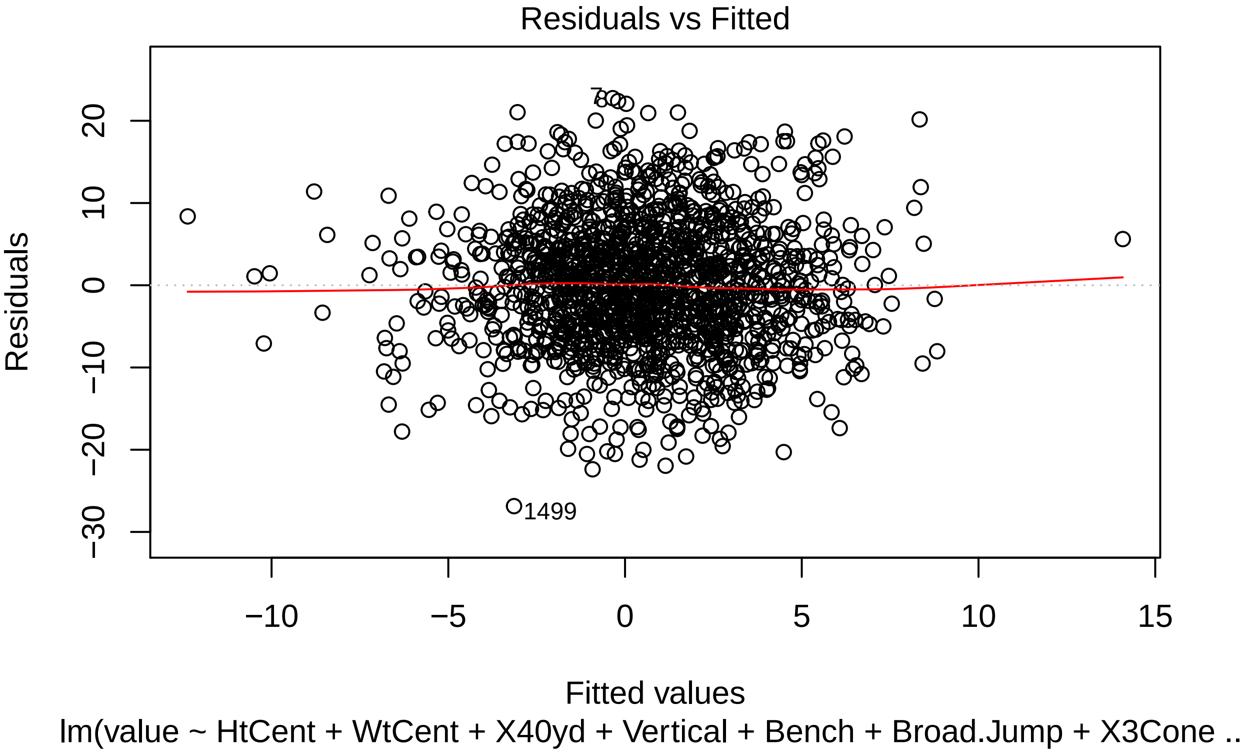
```
dotplot(ranef(model1,condVar=TRUE))$Position
```

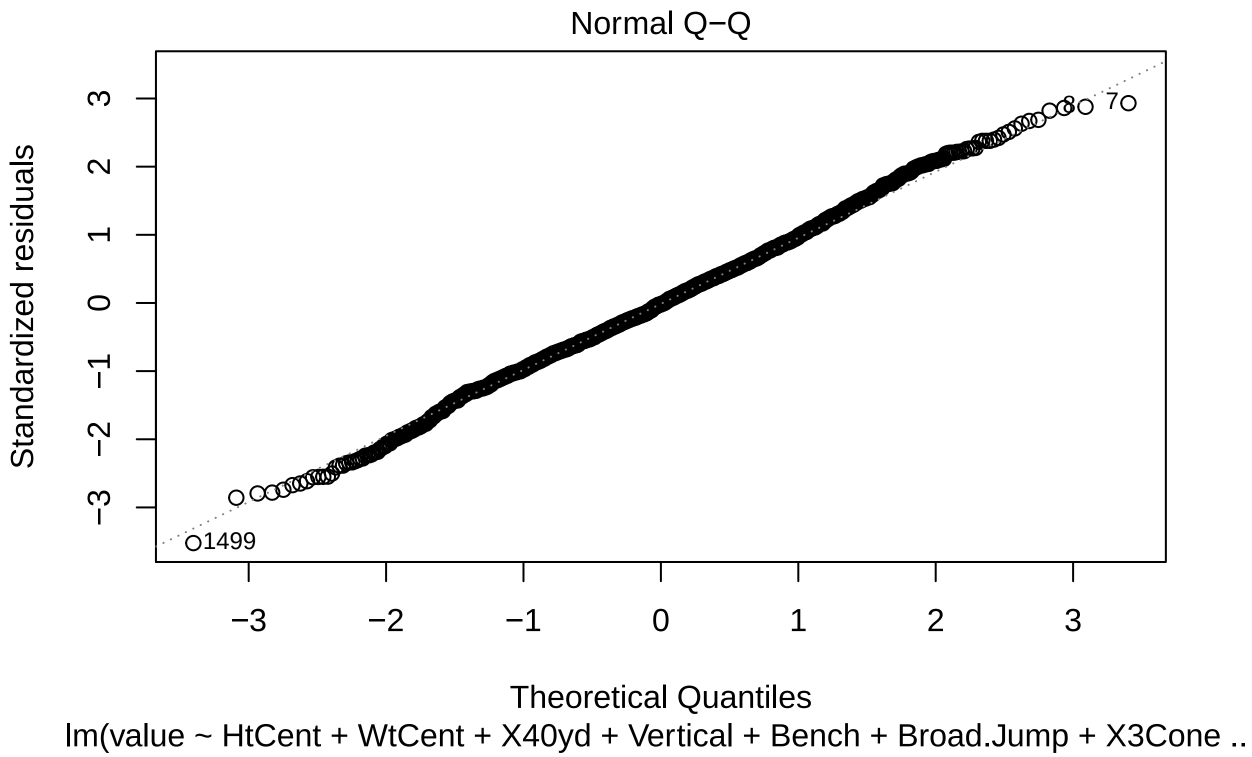


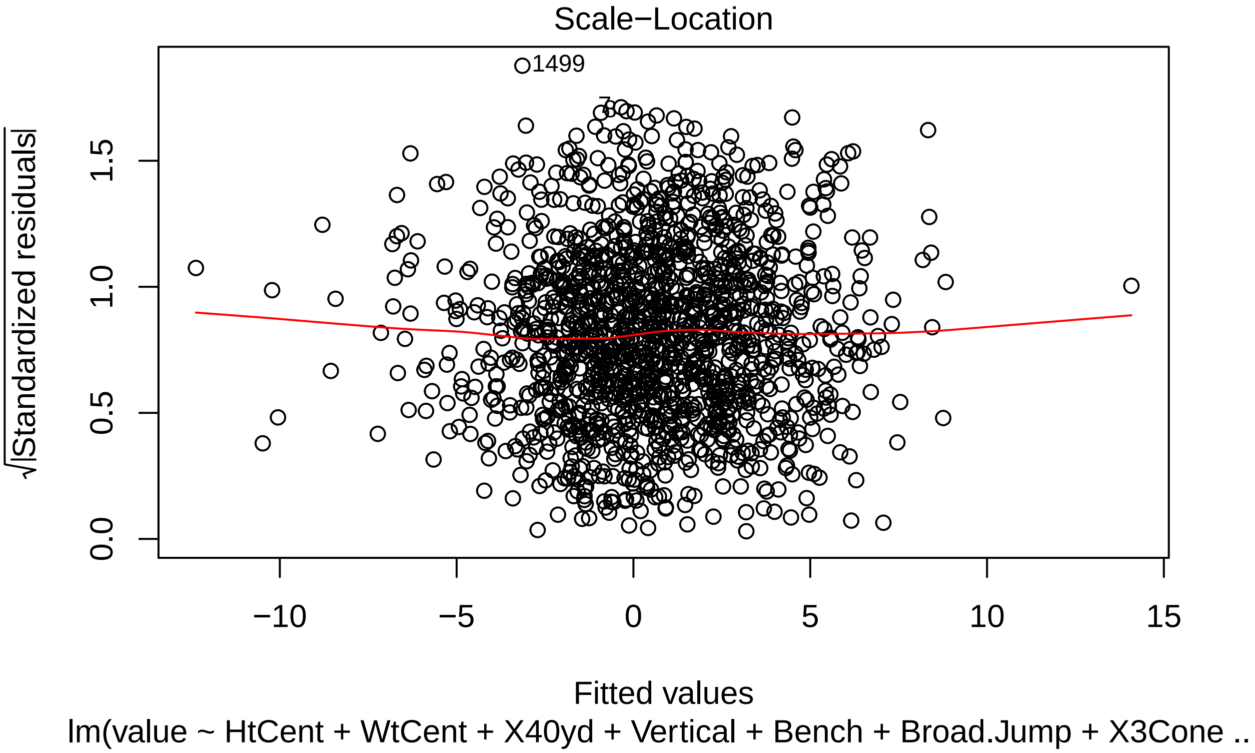
```
model2 <- lm(value ~ HtCent + WtCent +X40yd + Vertical + Bench + Broad.Jump +X3Cone + Shuttle + Round +

summary(model2)
```

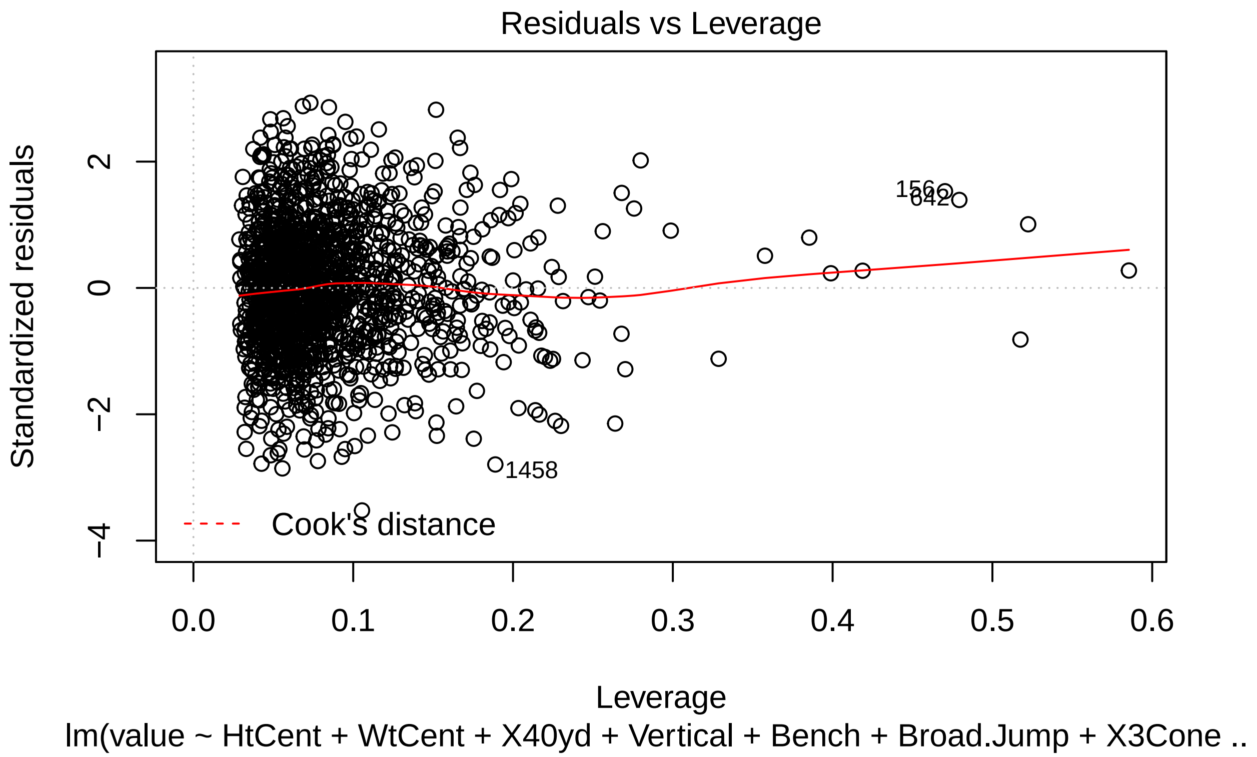
```
##
## Call:
## lm(formula = value ~ HtCent + WtCent + X40yd + Vertical + Bench +
##     Broad.Jump + X3Cone + Shuttle + Round + Team + Position:HtCent +
##     Position:WtCent + Position:X40yd + Position:Vertical + Position:Bench +
##     Position:Broad.Jump + Position:X3Cone + Position:Shuttle,
##     data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.8597  -5.0561  -0.1487   4.9533  22.7491
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               8.20477   17.39558   0.472   0.6372
## HtCent                    0.98267    1.60740   0.611   0.5411
## WtCent                   -0.46223    0.23595  -1.959   0.0503 .
## X40yd                    11.93013    8.19622   1.456   0.1457
## Vertical                  0.27518    0.67424   0.408   0.6832
## Bench                     0.41245    0.26229   1.573   0.1161
## Broad.Jump               -0.05953    0.26819  -0.222   0.8244
## X3Cone                   -0.75239    6.62875  -0.114   0.9096
## Shuttle                 -10.95510   10.97858  -0.998   0.3185
## Round2                   -0.35218    0.72078  -0.489   0.6252
## Round3                   -0.27595    0.71402  -0.386   0.6992
## Round4                  0.18984    0.75166   0.253   0.8006
## Round5                  0.62553    0.82082   0.762   0.4461
## Round6                  0.46799    0.93227   0.502   0.6158
## Round7                  0.56082    1.01934   0.550   0.5823
## TeamBears              -1.76619    1.72425  -1.024   0.3059
## TeamBengals            -0.72958    1.61634  -0.451   0.6518
## TeamBills              -1.24579    1.63206  -0.763   0.4454
## TeamBroncos            -0.01367    1.62112  -0.008   0.9933
## TeamBrowns             -3.20597    1.52813  -2.098   0.0361 *
## TeamBuccaneers         -1.61065    1.66392  -0.968   0.3332
## TeamCardinals          -3.18630    1.66280  -1.916   0.0555 .
## TeamChargers           -0.39393    1.72468  -0.228   0.8194
## TeamChiefs              0.52534    1.70264   0.309   0.7577
## TeamColts               1.07915    1.61207   0.669   0.5033
## TeamCowboys             1.74251    1.64351   1.060   0.2892
## TeamDolphins           -1.78898    1.65230  -1.083   0.2791
## TeamEagles             -0.36635    1.63426  -0.224   0.8227
## TeamFalcons             0.04514    1.71254   0.026   0.9790
## TeamGiants             -1.01079    1.69713  -0.596   0.5515
## TeamJaguars            -2.48597    1.80436  -1.378   0.1685
## TeamJets               -1.21741    1.65864  -0.734   0.4631
## TeamLions              -0.69817    1.64524  -0.424   0.6714
## TeamPackers            -0.59576    1.59087  -0.374   0.7081
## TeamPanthers            0.85004    1.75700   0.484   0.6286
## TeamPatriots            3.30063    1.74938   1.887   0.0594 .
## TeamRaiders            -1.16733    1.60533  -0.727   0.4673
## TeamRams               -0.30282    1.69827  -0.178   0.8585
## TeamRavens              0.65349    1.59137   0.411   0.6814
## TeamRedskins           -0.44532    1.62591  -0.274   0.7842
## TeamSaints              2.51525    1.83695   1.369   0.1711
## TeamSeahawks            1.01663    1.63453   0.622   0.5341
## TeamSteelers            0.72852    1.65547   0.440   0.6600
## TeamTexans             -1.59811    1.62294  -0.985   0.3249
## TeamTitans             -0.11553    1.65106  -0.070   0.9442
## TeamVikings             0.89107    1.64074   0.543   0.5872
## HtCent:PositionDB      -1.22916    1.65339  -0.743   0.4574
## HtCent:PositionDE      -1.21479    1.69325  -0.717   0.4732
## HtCent:PositionDT      -1.61162    1.68089  -0.959   0.3378
## HtCent:PositionG        0.23700    1.85820   0.128   0.8985
## HtCent:PositionLB      -0.37629    1.67590  -0.225   0.8224
## HtCent:PositionQB      -2.47612    1.90219  -1.302   0.1932
## HtCent:PositionRB      -1.51770    1.69833  -0.894   0.3717
## HtCent:PositionT       -1.33785    1.73815  -0.770   0.4416
## HtCent:PositionTE      -1.43602    1.77234  -0.810   0.4179
## HtCent:PositionWR      -1.17574    1.65938  -0.709   0.4787
## WtCent:PositionDB       0.49854    0.24223   2.058   0.0398 *
## WtCent:PositionDE       0.49624    0.24199   2.051   0.0405 *
## WtCent:PositionDT       0.47811    0.24253   1.971   0.0489 *
## WtCent:PositionG        0.44154    0.26418   1.671   0.0949 .
## WtCent:PositionLB       0.39209    0.24401   1.607   0.1083
## WtCent:PositionQB       0.31361    0.29668   1.057   0.2907
## WtCent:PositionRB       0.58981    0.24593   2.398   0.0166 *
## WtCent:PositionT        0.45940    0.24576   1.869   0.0618 .
## WtCent:PositionTE       0.55427    0.26272   2.110   0.0351 *
## WtCent:PositionWR       0.51493    0.24468   2.104   0.0355 *
## X40yd:PositionDB      -10.47190    9.14681  -1.145   0.2525
## X40yd:PositionDE      -17.61007    9.52546  -1.849   0.0647 .
## X40yd:PositionDT       -9.96429    9.48464  -1.051   0.2936
## X40yd:PositionG        -9.04176    9.87589  -0.916   0.3601
## X40yd:PositionLB       -5.07123    9.29409  -0.546   0.5854
## X40yd:PositionQB      -14.79115   12.79444  -1.156   0.2479
## X40yd:PositionRB       -9.96270    9.99960  -0.996   0.3193
## X40yd:PositionT       -12.22600    9.35000  -1.308   0.1912
## X40yd:PositionTE      -19.49757   10.38226  -1.878   0.0606 .
## X40yd:PositionWR       -8.54356    9.46780  -0.902   0.3670
## Vertical:PositionDB    -0.37265    0.72467  -0.514   0.6072
## Vertical:PositionDE     0.16268    0.76287   0.213   0.8312
## Vertical:PositionDT    -0.37156    0.74513  -0.499   0.6181
## Vertical:PositionG     -0.10938    0.83848  -0.130   0.8962
## Vertical:PositionLB    -0.20181    0.73651  -0.274   0.7841
## Vertical:PositionQB    -0.42623    0.90222  -0.472   0.6367
## Vertical:PositionRB    -0.24669    0.74618  -0.331   0.7410
## Vertical:PositionT     -0.02348    0.75215  -0.031   0.9751
## Vertical:PositionTE    -0.48435    0.79634  -0.608   0.5431
## Vertical:PositionWR     0.18106    0.73265   0.247   0.8048
## Bench:PositionDB       -0.39949    0.29507  -1.354   0.1760
## Bench:PositionDE       -0.13817    0.31816  -0.434   0.6641
## Bench:PositionDT       -0.07801    0.30217  -0.258   0.7963
## Bench:PositionG        -0.40460    0.31964  -1.266   0.2058
## Bench:PositionLB       -0.28355    0.30571  -0.928   0.3538
## Bench:PositionQB       -1.82149    4.97909  -0.366   0.7146
## Bench:PositionRB       -0.64525    0.30808  -2.094   0.0364 *
## Bench:PositionT        -0.03482    0.32860  -0.106   0.9156
## Bench:PositionTE       -0.55046    0.35209  -1.563   0.1182
## Bench:PositionWR       -0.32317    0.32557  -0.993   0.3211
## Broad.Jump:PositionDB   0.08660    0.29003   0.299   0.7653
## Broad.Jump:PositionDE  -0.00728    0.30869  -0.024   0.9812
## Broad.Jump:PositionDT   0.12955    0.29942   0.433   0.6653
## Broad.Jump:PositionG    0.10109    0.31797   0.318   0.7506
## Broad.Jump:PositionLB  -0.16695    0.30415  -0.549   0.5832
## Broad.Jump:PositionQB   0.14547    0.41233   0.353   0.7243
## Broad.Jump:PositionRB   0.12735    0.31314   0.407   0.6843
## Broad.Jump:PositionT   -0.22512    0.30353  -0.742   0.4584
## Broad.Jump:PositionTE   0.08767    0.33365   0.263   0.7928
## Broad.Jump:PositionWR  -0.30785    0.29734  -1.035   0.3007
## X3Cone:PositionDB       0.56337    7.62657   0.074   0.9411
## X3Cone:PositionDE       0.52285    8.12113   0.064   0.9487
## X3Cone:PositionDT      -6.28518    7.64190  -0.822   0.4110
## X3Cone:PositionG       -7.00358    8.30000  -0.844   0.3989
## X3Cone:PositionLB      -3.07286    7.76807  -0.396   0.6925
## X3Cone:PositionQB       2.11544   11.18695   0.189   0.8500
## X3Cone:PositionRB      -1.23049    8.48304  -0.145   0.8847
## X3Cone:PositionT       -6.00870    8.23214  -0.730   0.4656
## X3Cone:PositionTE       6.16653    9.60392   0.642   0.5209
## X3Cone:PositionWR       2.64643    7.97343   0.332   0.7400
## Shuttle:PositionDB      7.97640   12.22293   0.653   0.5141
## Shuttle:PositionDE     12.51361   12.50369   1.001   0.3171
## Shuttle:PositionDT     15.78371   12.53147   1.260   0.2081
## Shuttle:PositionG      16.91762   13.17422   1.284   0.1993
## Shuttle:PositionLB     12.83834   12.50317   1.027   0.3047
## Shuttle:PositionQB     14.70355   15.55738   0.945   0.3448
## Shuttle:PositionRB     10.53558   13.16954   0.800   0.4239
## Shuttle:PositionT      23.81175   13.34107   1.785   0.0745 .
## Shuttle:PositionTE     10.62914   13.84505   0.768   0.4428
## Shuttle:PositionWR      9.71812   12.34404   0.787   0.4313
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.061 on 1373 degrees of freedom
## Multiple R-squared:  0.115,  Adjusted R-squared:  0.0344
## F-statistic: 1.427 on 125 and 1373 DF,  p-value: 0.002099
```

```
plot(model2)
```




Normal Q−Q
Standardized residuals
3
2
1
0
−1
−2
−3
1499
8
7
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(value ~ HtCent + WtCent + X40yd + Vertical + Bench + Broad.Jump + X3Cone ..


Scale−Location
√|Standardized residuals|
1.5
1.0
0.5
0.0
1499
7
−10
−5
0
5
10
15
Fitted values
lm(value ~ HtCent + WtCent + X40yd + Vertical + Bench + Broad.Jump + X3Cone ..



```r
NullModel <- lm(value ~ 1, data = df)
FullModel <- lm(value ~ Position + HtCent + WtCent +X40yd + Vertical + Bench + Broad.Jump +X3Cone + Shu
```

```r
Model_stepwise <- step(NullModel, scope = formula(FullModel),direction="both",trace=0)
Model_stepwise$call
```

```
## lm(formula = value ~ Position + Bench + Broad.Jump + X3Cone,
##     data = df)
```

```r
pander(summary(Model_stepwise))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	22.24	11.17	1.991	0.04666
PositionDB	0.924	1.772	0.5215	0.6021
PositionDE	-0.1015	1.577	-0.06436	0.9487
PositionDT	0.3966	1.399	0.2835	0.7768
PositionG	0.9636	1.526	0.6317	0.5277
PositionLB	0.05772	1.625	0.03552	0.9717
PositionQB	-2.524	1.832	-1.377	0.1686
PositionRB	4.27	1.703	2.507	0.01229
PositionT	0.1624	1.439	0.1128	0.9102
PositionTE	2.731	1.708	1.599	0.1101
PositionWR	3.199	1.809	1.769	0.07713
Bench	0.1367	0.05007	2.73	0.006411
Broad.Jump	-0.09045	0.0406	-2.228	0.02604
X3Cone	-2.12	1.205	-1.76	0.07867

Table 3: Fitting linear model: value ~ Position + Bench + Broad.Jump + X3Cone

Observations	Residual Std. Error	R^2	Adjusted R^2
1499	8.076	0.03937	0.03096

```r
AIC(Model_stepwise)
```

```
## [1] 10532.3
```

```r
a3 <- lm(value ~ HtCent + WtCent +X40yd + Vertical + Bench + Broad.Jump + Round + Team + Position:HtCent
a4 <- lm(value ~ HtCent + WtCent +X40yd + Vertical + Bench + Broad.Jump + Round + Team + Position:HtCent


pander(anova(a4, a3))
```

Table 4: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1405	90869	NA	NA	NA	NA
1405	90869	0	0	NA	NA

```r
a3 <- lm(value ~ HtCent + WtCent +X40yd + Vertical + Bench + Broad.Jump + Round + Team + Position:HtCent
a4 <- lm(value ~ HtCent + WtCent +X40yd + Vertical + Bench + Round + Team + Position:HtCent + Position:W


pander(anova(a4, a3))
```

Table 5: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1406	90998	NA	NA	NA	NA
1405	90869	1	129.7	2.005	0.157

```r
a3 <- lm(Grade ~ HtCent + WtCent +X40yd + Vertical + Bench + Round + Team + Position:HtCent + Position:W
a4 <- lm(Grade ~ HtCent + WtCent +X40yd + Vertical + Bench + Round + Team + Position:HtCent + Position:W


pander(anova(a4, a3))
```

Table 6: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1416	94035	NA	NA	NA	NA
1406	93113	10	921.7	1.392	0.1781

```r
a3 <- lm(value ~ HtCent + WtCent +X40yd + Vertical + Bench + Round + Team + Position:HtCent + Position:W
a4 <- lm(value ~ HtCent + WtCent +X40yd + Vertical + Round + Team + Position:HtCent + Position:WtCent +
pander(anova(a4, a3))
```

Table 7: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1417	92050	NA	NA	NA	NA
1416	91770	1	279.6	4.314	0.03797

```
a3 <- lm(value ~ HtCent + WtCent +X40yd + Vertical + Bench + Round + Team + Position:HtCent + Position:W
a4 <- lm(value ~ HtCent + WtCent +X40yd + Bench + Round + Team + Position:HtCent + Position:WtCent + Pos


pander(anova(a4, a3))
```

Table 8: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1427	92256	NA	NA	NA	NA
1426	92256	1	0.1899	0.002935	0.9568

```
a3 <- lm(value ~ HtCent + WtCent +X40yd + Bench + Round + Team + Position:HtCent + Position:WtCent + Pos
a4 <- lm(value ~ HtCent + WtCent +X40yd + Bench + Round + Team + Position:HtCent + Position:WtCent, data


pander(anova(a4, a3))
```

Table 9: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1437	93750	NA	NA	NA	NA
1427	92256	10	1494	2.31	0.01081

```
a3 <- lm(value ~ HtCent + WtCent +X40yd + Bench + Round + Team + Position:HtCent + Position:X40yd, data
a4 <- lm(value ~ HtCent + X40yd + Bench + Round + Team + Position:HtCent + Position:X40yd, data = df)


pander(anova(a4, a3))
```

Table 10: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1438	92879	NA	NA	NA	NA
1437	92859	1	20.4	0.3156	0.5743

```
a3 <- lm(value ~ HtCent + X40yd + Bench + Round + Team + Position:HtCent + Position:X40yd, data = df)
a4 <- lm(value ~ HtCent + X40yd + Bench + Round + Team + Position:HtCent + Position:X40yd, data = df)
```

```
pander(anova(a4, a3))
```

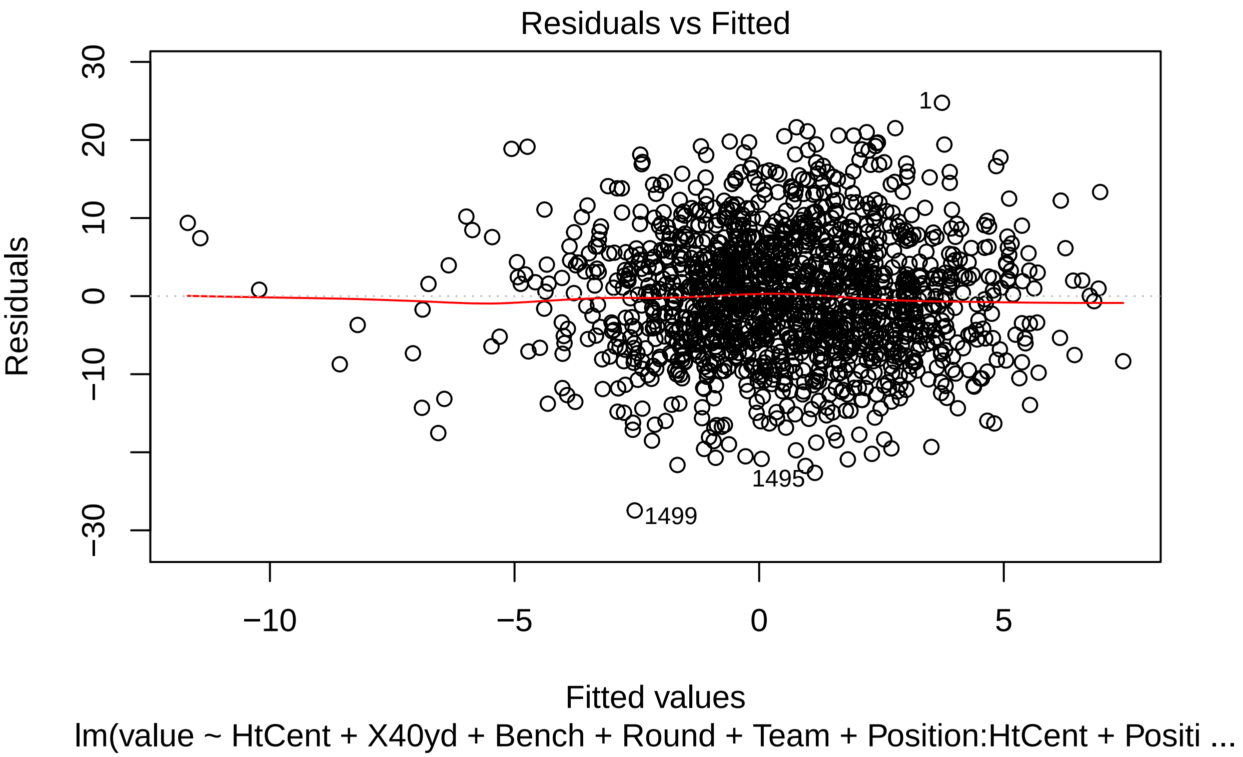
Table 11: Analysis of Variance Table

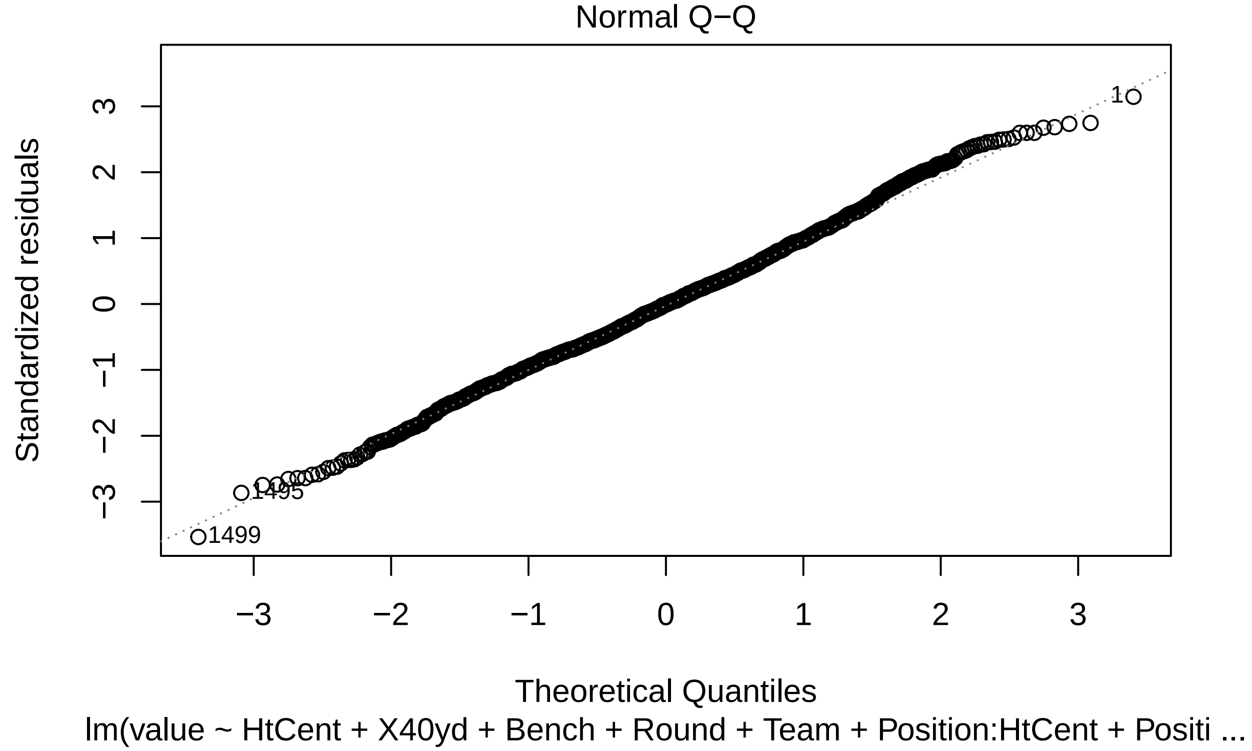
Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1438	92879	NA	NA	NA	NA
1438	92879	0	0	NA	NA

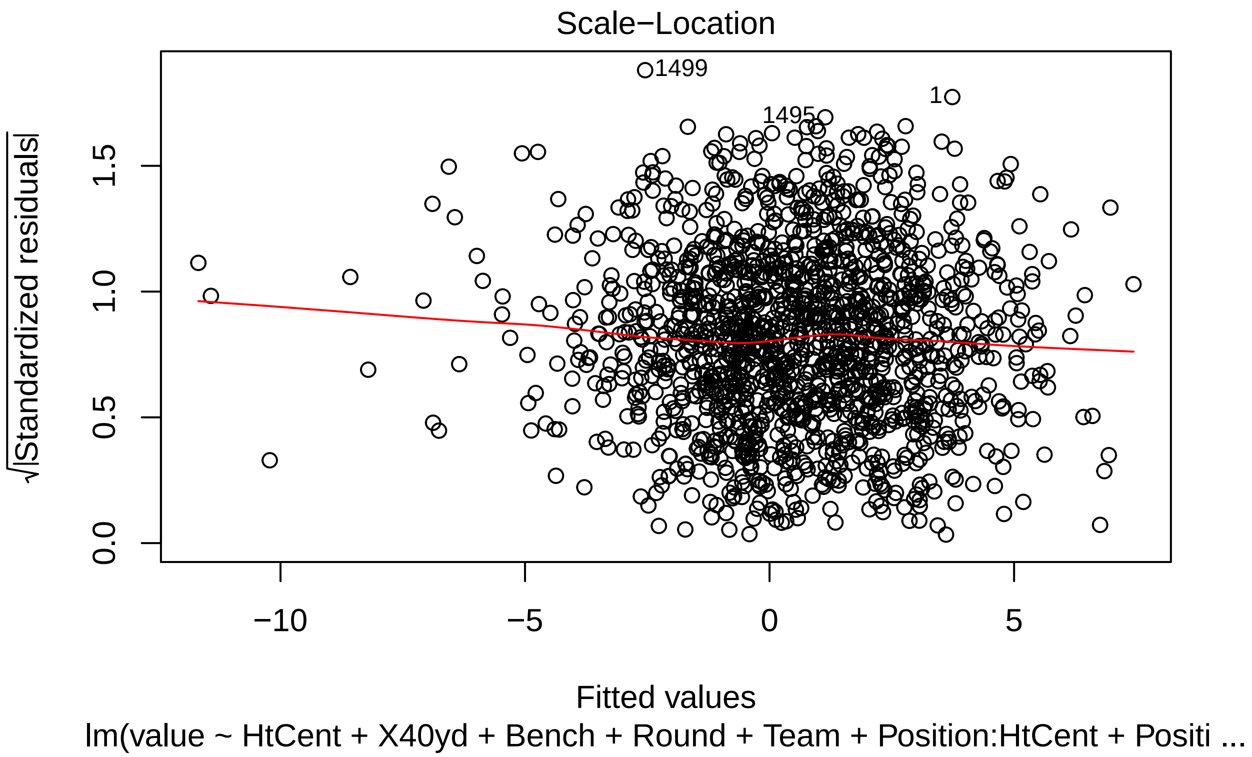
```
##### Model 3
model3 <- lm(value ~ HtCent + X40yd + Bench + Round + Team + Position:HtCent + Position:X40yd, data = d
summary(model3)
```

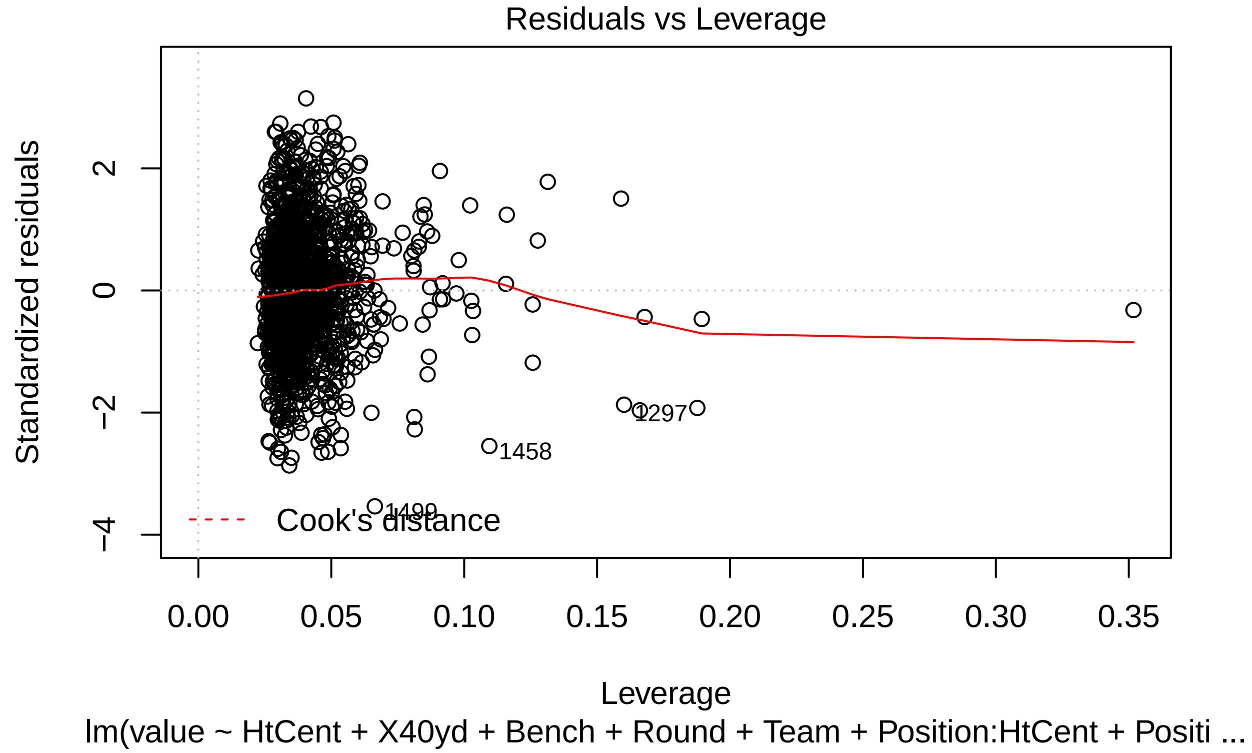
```
##
## Call:
## lm(formula = value ~ HtCent + X40yd + Bench + Round + Team +
##     Position:HtCent + Position:X40yd, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27.4560  -5.3635  -0.0618   4.9744  24.7633
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -8.67108    8.36244  -1.037  0.29995
## HtCent              -1.29428    1.17077  -1.105  0.26913
## X40yd                1.49131    1.62482   0.918  0.35886
## Bench                0.13687    0.05087   2.690  0.00722 **
## Round2              -0.40947    0.70048  -0.585  0.55894
## Round3              -0.40126    0.69578  -0.577  0.56423
## Round4               0.11001    0.72532   0.152  0.87947
## Round5               0.52023    0.79376   0.655  0.51232
## Round6               0.54282    0.89714   0.605  0.54524
## Round7               0.74701    0.98081   0.762  0.44641
## TeamBears           -1.81341    1.68281  -1.078  0.28139
## TeamBengals         -0.97173    1.57530  -0.617  0.53743
## TeamBills           -1.39296    1.58966  -0.876  0.38103
## TeamBroncos         -0.09694    1.57345  -0.062  0.95088
## TeamBrowns          -3.25710    1.49139  -2.184  0.02913 *
## TeamBuccaneers      -1.72111    1.62822  -1.057  0.29067
## TeamCardinals       -3.60979    1.61481  -2.235  0.02554 *
## TeamChargers        -0.95526    1.68469  -0.567  0.57079
## TeamChiefs          -0.01235    1.65530  -0.007  0.99405
## TeamColts            0.48690    1.57147   0.310  0.75673
## TeamCowboys          1.52356    1.60977   0.946  0.34408
## TeamDolphins        -1.79252    1.61026  -1.113  0.26581
## TeamEagles          -0.38877    1.58062  -0.246  0.80575
## TeamFalcons         -0.42020    1.67506  -0.251  0.80196
## TeamGiants          -1.21975    1.62503  -0.751  0.45302
## TeamJaguars         -2.83694    1.76221  -1.610  0.10764
## TeamJets            -1.59425    1.60925  -0.991  0.32201
## TeamLions           -0.86684    1.61300  -0.537  0.59107
## TeamPackers         -0.11045    1.54658  -0.071  0.94308
## TeamPanthers       0.94349    1.72010   0.549  0.58343
## TeamPatriots       2.46985    1.68394   1.467  0.14267
## TeamRaiders       -1.69838    1.56562  -1.085  0.27819
## TeamRams          -0.31997    1.66360  -0.192  0.84751
## TeamRavens         0.46464    1.54797   0.300  0.76410
## TeamRedskins      -0.38860    1.58725  -0.245  0.80662
## TeamSaints         2.15933    1.77529   1.216  0.22406
## TeamSeahawks       0.70120    1.59122   0.441  0.65952
## TeamSteelers       0.50874    1.60834   0.316  0.75181
## TeamTexans        -1.47338    1.58420  -0.930  0.35250
## TeamTitans        -0.45287    1.60927  -0.281  0.77844
## TeamVikings        0.87009    1.59884   0.544  0.58639
## HtCent:PositionDB  1.14764    1.21440   0.945  0.34480
## HtCent:PositionDE  1.02080    1.27112   0.803  0.42207
## HtCent:PositionDT  0.61679    1.26109   0.489  0.62485
## HtCent:PositionG   2.57061    1.47599   1.742  0.08179 .
## HtCent:PositionLB  1.61148    1.23300   1.307  0.19144
## HtCent:PositionQB -0.80671    1.31707  -0.613  0.54030
## HtCent:PositionRB  1.40674    1.23443   1.140  0.25465
## HtCent:PositionT   0.98879    1.30305   0.759  0.44808
## HtCent:PositionTE  1.29100    1.30362   0.990  0.32219
## HtCent:PositionWR  1.18127    1.20125   0.983  0.32559
## X40yd:PositionDB  -0.08076    0.56255  -0.144  0.88586
## X40yd:PositionDE  -0.23985    0.53335  -0.450  0.65299
## X40yd:PositionDT  -0.13616    0.46790  -0.291  0.77109
## X40yd:PositionG   -0.69911    0.59386  -1.177  0.23929
## X40yd:PositionLB  -0.26622    0.49861  -0.534  0.59347
## X40yd:PositionQB  -0.07535    0.54637  -0.138  0.89033
## X40yd:PositionRB   0.79239    0.59472   1.332  0.18295
## X40yd:PositionT   -0.18364    0.61145  -0.300  0.76396
## X40yd:PositionTE   0.30532    0.59234   0.515  0.60632
## X40yd:PositionWR   0.51110    0.55719   0.917  0.35915
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.037 on 1438 degrees of freedom
## Multiple R-squared:  0.07872,    Adjusted R-squared:  0.04028
## F-statistic: 2.048 on 60 and 1438 DF,  p-value: 6.411e-06
```

```
plot(model3)
```


Residuals vs Fitted
Residuals
30
20
10
0
−10
−30
1
1495
1499
−10
−5
0
5
Fitted values
lm(value ~ HtCent + X40yd + Bench + Round + Team + Position:HtCent + Positi ...


Normal Q−Q
Standardized residuals
3
2
1
0
−1
−2
−3
1
1495
1499
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(value ~ HtCent + X40yd + Bench + Round + Team + Position:HtCent + Positi ...





```
df$GradePct <- (df$Grade / 100)

logModel <- glm(GradePct~  HtCent + WtCent +X40yd + Vertical + Bench + Broad.Jump +X3Cone + Shuttle + Ro
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```r
pander(summary(logModel))
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1.214	4.506	0.2694	0.7876
HtCent	0.046	0.4124	0.1115	0.9112
WtCent	-0.02108	0.0621	-0.3394	0.7343
X40yd	0.5507	2.131	0.2584	0.7961
Vertical	0.01018	0.1726	0.05901	0.9529
Bench	0.01924	0.06749	0.2851	0.7756
Broad.Jump	-0.001247	0.06871	-0.01814	0.9855
X3Cone	-0.03807	1.709	-0.02228	0.9822
Shuttle	-0.5368	2.818	-0.1905	0.8489
Round2	-0.1591	0.19	-0.8374	0.4024
Round3	-0.265	0.1872	-1.415	0.1569
Round4	-0.3024	0.1965	-1.539	0.1238
Round5	-0.3027	0.2138	-1.416	0.1569
Round6	-0.3657	0.2411	-1.517	0.1293
Round7	-0.3579	0.2631	-1.361	0.1736
TeamBears	-0.07312	0.4463	-0.1638	0.8699
TeamBengals	-0.05219	0.4186	-0.1247	0.9008
TeamBills	-0.04086	0.4224	-0.09675	0.9229
TeamBroncos	-0.005665	0.4207	-0.01347	0.9893
TeamBrowns	-0.1302	0.3943	-0.3303	0.7412
TeamBuccaneers	-0.06475	0.4305	-0.1504	0.8804
TeamCardinals	-0.1393	0.428	-0.3254	0.7449
TeamChargers	-0.01488	0.4466	-0.03332	0.9734
TeamChiefs	0.02087	0.4426	0.04714	0.9624
TeamColts	0.03885	0.4191	0.09269	0.9261
TeamCowboys	0.07955	0.429	0.1854	0.8529
TeamDolphins	-0.07049	0.4277	-0.1648	0.8691
TeamEagles	-0.01802	0.4233	-0.04257	0.966
TeamFalcons	0.00217	0.444	0.004886	0.9961
TeamGiants	-0.04635	0.4402	-0.1053	0.9161
TeamJaguars	-0.07754	0.466	-0.1664	0.8678
TeamJets	-0.04796	0.4295	-0.1117	0.9111
TeamLions	-0.02786	0.4267	-0.0653	0.9479
TeamPackers	-0.0508	0.4126	-0.1231	0.902
TeamPanthers	0.03386	0.4584	0.07388	0.9411
TeamPatriots	0.1137	0.4579	0.2483	0.8039
TeamRaiders	-0.0345	0.4153	-0.08308	0.9338
TeamRams	-0.0004175	0.4415	-0.0009456	0.9992
TeamRavens	0.01165	0.4135	0.02818	0.9775
TeamRedskins	-0.01835	0.421	-0.04359	0.9652
TeamSaints	0.09737	0.4822	0.2019	0.84
TeamSeahawks	0.03236	0.4251	0.07613	0.9393
TeamSteelers	0.009135	0.4304	0.02123	0.9831
TeamTexans	-0.07521	0.4195	-0.1793	0.8577
TeamTitans	-1.932e-05	0.4287	-4.507e-05	1
TeamVikings	0.02194	0.4274	0.05133	0.9591
HtCent:PositionDB	-0.05488	0.4242	-0.1294	0.8971
HtCent:PositionDE	-0.04944	0.4348	-0.1137	0.9095
HtCent:PositionDT	-0.07727	0.4317	-0.179	0.8579
HtCent:PositionG	0.009008	0.4787	0.01882	0.985

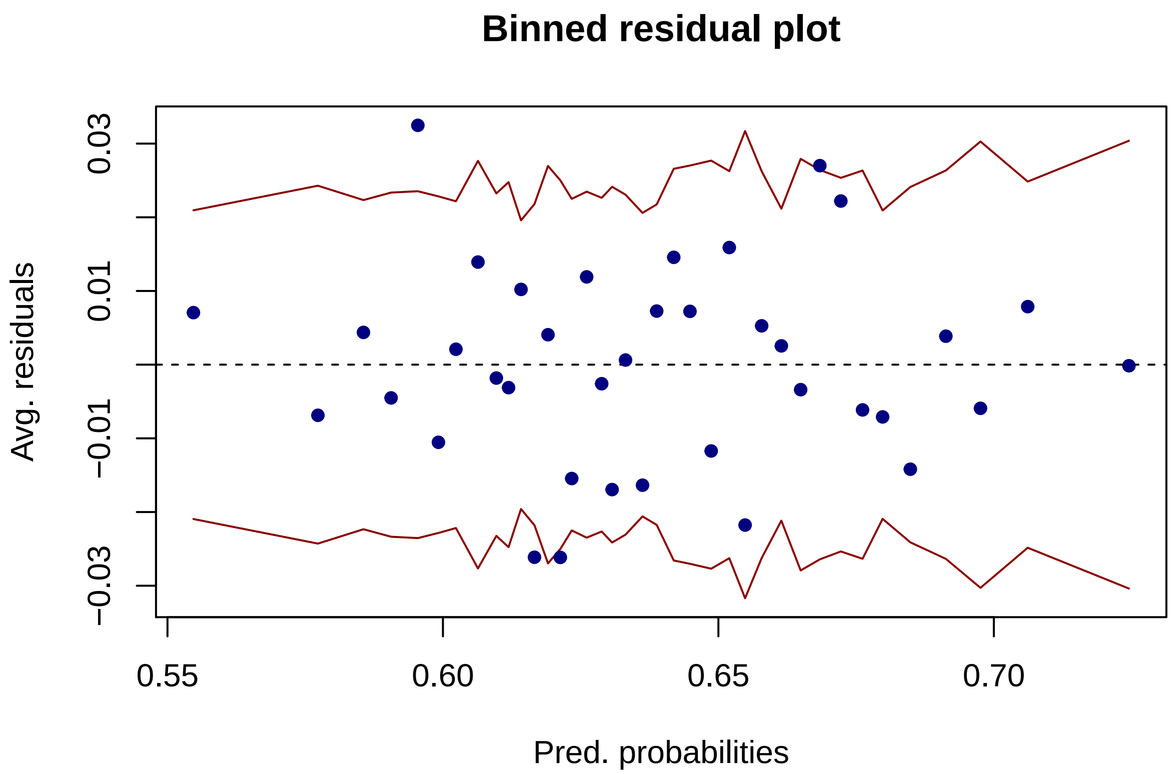
	Estimate	Std. Error	z value	Pr(>|z|)
HtCent:PositionLB	-0.02006	0.4301	-0.04664	0.9628
HtCent:PositionQB	-0.1093	0.4895	-0.2233	0.8233
HtCent:PositionRB	-0.06952	0.4366	-0.1592	0.8735
HtCent:PositionT	-0.06204	0.4469	-0.1388	0.8896
HtCent:PositionTE	-0.06111	0.4555	-0.1341	0.8933
HtCent:PositionWR	-0.05287	0.426	-0.1241	0.9012
WtCent:PositionDB	0.02279	0.06369	0.3578	0.7205
WtCent:PositionDE	0.02283	0.06363	0.3588	0.7198
WtCent:PositionDT	0.02209	0.06379	0.3463	0.7291
WtCent:PositionG	0.0211	0.06933	0.3044	0.7608
WtCent:PositionLB	0.01848	0.06413	0.2881	0.7732
WtCent:PositionQB	0.01562	0.07748	0.2016	0.8403
WtCent:PositionRB	0.02734	0.06472	0.4224	0.6728
WtCent:PositionT	0.02075	0.06464	0.3209	0.7483
WtCent:PositionTE	0.02476	0.06891	0.3592	0.7194
WtCent:PositionWR	0.02331	0.06435	0.3622	0.7172
X40yd:PositionDB	-0.5101	2.372	-0.215	0.8297
X40yd:PositionDE	-0.8018	2.478	-0.3235	0.7463
X40yd:PositionDT	-0.4833	2.466	-0.196	0.8446
X40yd:PositionG	-0.4454	2.566	-0.1736	0.8622
X40yd:PositionLB	-0.2783	2.41	-0.1155	0.9081
X40yd:PositionQB	-0.8171	3.292	-0.2482	0.804
X40yd:PositionRB	-0.4857	2.605	-0.1865	0.8521
X40yd:PositionT	-0.6318	2.434	-0.2596	0.7952
X40yd:PositionTE	-0.8588	2.693	-0.3189	0.7498
X40yd:PositionWR	-0.4436	2.462	-0.1802	0.857
Vertical:PositionDB	-0.0151	0.1856	-0.08138	0.9351
Vertical:PositionDE	0.01174	0.1957	0.06001	0.9521
Vertical:PositionDT	-0.01449	0.1913	-0.07576	0.9396
Vertical:PositionG	-0.01197	0.2158	-0.05546	0.9558
Vertical:PositionLB	-0.002702	0.1887	-0.01431	0.9886
Vertical:PositionQB	-0.0354	0.2321	-0.1525	0.8788
Vertical:PositionRB	-0.007919	0.192	-0.04124	0.9671
Vertical:PositionT	-0.002548	0.193	-0.0132	0.9895
Vertical:PositionTE	-0.01836	0.2048	-0.08962	0.9286
Vertical:PositionWR	0.006684	0.1879	0.03556	0.9716
Bench:PositionDB	-0.01833	0.07587	-0.2416	0.8091
Bench:PositionDE	-0.007183	0.08207	-0.08753	0.9303
Bench:PositionDT	-0.005063	0.07808	-0.06484	0.9483
Bench:PositionG	-0.01822	0.08246	-0.221	0.8251
Bench:PositionLB	-0.01444	0.07858	-0.1838	0.8542
Bench:PositionQB	-0.02389	1.289	-0.01854	0.9852
Bench:PositionRB	-0.03068	0.07968	-0.385	0.7002
Bench:PositionT	0.0003091	0.08507	0.003634	0.9971
Bench:PositionTE	-0.02432	0.09088	-0.2676	0.789
Bench:PositionWR	-0.01471	0.08409	-0.1749	0.8611
Broad.Jump:PositionDB	0.002118	0.07434	0.02849	0.9773
Broad.Jump:PositionDE	-0.0006198	0.0792	-0.007826	0.9938
Broad.Jump:PositionDT	0.005367	0.07695	0.06974	0.9444
Broad.Jump:PositionG	0.006523	0.08192	0.07963	0.9365
Broad.Jump:PositionLB	-0.009195	0.07796	-0.118	0.9061
Broad.Jump:PositionQB	0.009473	0.1058	0.08957	0.9286

	Estimate	Std. Error	z value	Pr(>|z|)
Broad.Jump:PositionRB	0.003616	0.08075	0.04478	0.9643
Broad.Jump:PositionT	-0.009617	0.078	-0.1233	0.9019
Broad.Jump:PositionTE	0.001181	0.08597	0.01374	0.989
Broad.Jump:PositionWR	-0.01287	0.07635	-0.1686	0.8661
X3Cone:PositionDB	0.03292	1.964	0.01677	0.9866
X3Cone:PositionDE	0.01024	2.096	0.004885	0.9961
X3Cone:PositionDT	-0.2647	1.974	-0.1341	0.8933
X3Cone:PositionG	-0.2995	2.145	-0.1396	0.889
X3Cone:PositionLB	-0.1181	1.998	-0.05909	0.9529
X3Cone:PositionQB	-0.02345	2.876	-0.008154	0.9935
X3Cone:PositionRB	-0.03139	2.203	-0.01425	0.9886
X3Cone:PositionT	-0.2901	2.125	-0.1365	0.8914
X3Cone:PositionTE	0.262	2.484	0.1055	0.916
X3Cone:PositionWR	0.1081	2.059	0.05248	0.9581
Shuttle:PositionDB	0.4211	3.138	0.1342	0.8932
Shuttle:PositionDE	0.5663	3.214	0.1762	0.8601
Shuttle:PositionDT	0.7176	3.222	0.2227	0.8238
Shuttle:PositionG	0.7599	3.392	0.224	0.8227
Shuttle:PositionLB	0.5963	3.207	0.186	0.8525
Shuttle:PositionQB	0.8057	3.994	0.2017	0.8402
Shuttle:PositionRB	0.5114	3.4	0.1504	0.8804
Shuttle:PositionT	1.185	3.433	0.345	0.7301
Shuttle:PositionTE	0.5186	3.568	0.1453	0.8844
Shuttle:PositionWR	0.495	3.177	0.1558	0.8762

(Dispersion parameter for binomial family taken to be 1)

Null deviance:	50.31 on 1498 degrees of freedom
Residual deviance:	41.08 on 1373 degrees of freedom

```
rawresid5 <- residuals(logModel,"resp")

binnedplot(x=fitted(logModel),y=rawresid5,xlab="Pred. probabilities",
          col.int="red4",ylab="Avg. residuals",main="Binned residual plot",col.pts="navy")
```



```
a3 <- lm(value ~ HtCent + WtCent +X40yd + Vertical + Bench + Broad.Jump +X3Cone + Shuttle + Round + Tea
```

```
a3 <- lm(value ~ Position + HtCent + X40yd + Bench + X3Cone + Team + Position:X40yd + Position:X3Cone,

a4 <- lm(value ~ Position + HtCent + X40yd + Bench+ X3Cone + Team + Position:X40yd, data = df)

pander(anova(a4, a3))
```

Table 14: Analysis of Variance Table

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1443	93081	NA	NA	NA	NA
1433	92265	10	816.4	1.268	0.2433

```
df$X40ydCent <- df$X40yd - mean(df$X40yd)
df$BenchCent <- df$Bench - mean(df$Bench)
model4 <- lm(value ~ Position + HtCent + X40ydCent + BenchCent +  Team + Position:X40ydCent, data = df)
pander(summary(model4))
```

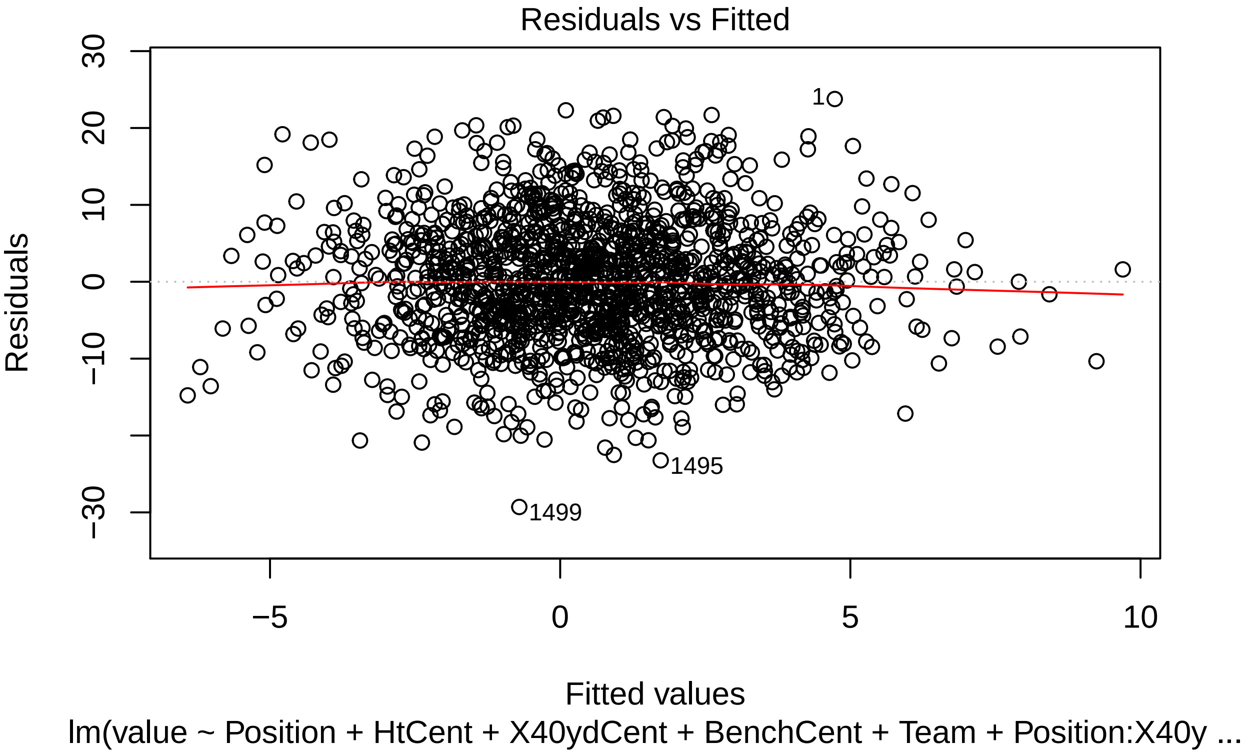
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.683	4.17	0.4035	0.6866
PositionDB	0.8592	4.32	0.1989	0.8424
PositionDE	-1.355	4.047	-0.3347	0.7379
PositionDT	-0.8542	4.214	-0.2027	0.8394
PositionG	2.011	5.013	0.4012	0.6884
PositionLB	-1.611	4.06	-0.3967	0.6916
PositionQB	-2.778	4.161	-0.6676	0.5045
PositionRB	4.606	4.276	1.077	0.2816

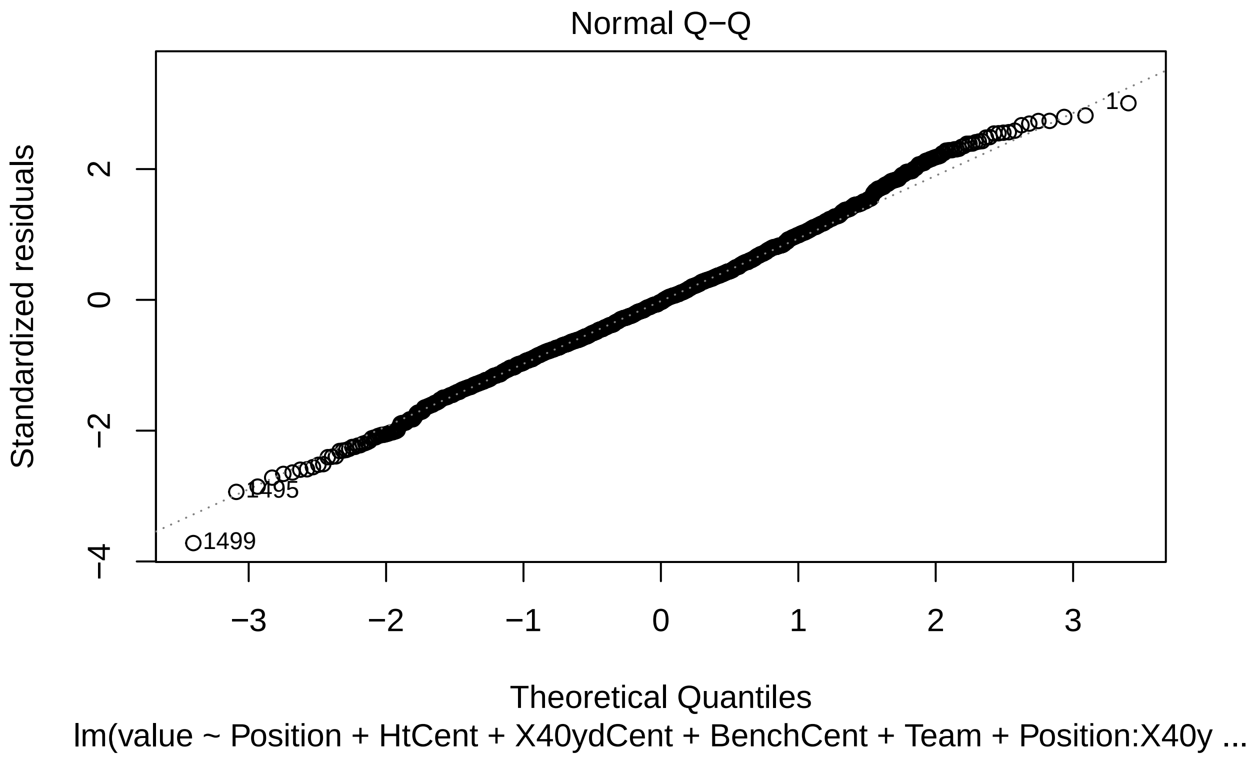
	Estimate	Std. Error	t value	Pr(>|t|)
PositionT	-0.7451	4.433	-0.1681	0.8665
PositionTE	1.468	4.093	0.3586	0.7199
PositionWR	4.99	4.471	1.116	0.2646
HtCent	-0.2466	0.1345	-1.833	0.06702
X40ydCent	-2.975	7.838	-0.3795	0.7044
BenchCent	0.1335	0.05043	2.647	0.008216
TeamBears	-1.694	1.679	-1.009	0.3132
TeamBengals	-1.003	1.569	-0.6394	0.5227
TeamBills	-1.15	1.587	-0.7247	0.4688
TeamBroncos	-0.3368	1.573	-0.2141	0.8305
TeamBrowns	-2.888	1.485	-1.945	0.05199
TeamBuccaneers	-1.868	1.628	-1.148	0.2514
TeamCardinals	-3.493	1.616	-2.161	0.03082
TeamChargers	-0.6018	1.686	-0.357	0.7211
TeamChiefs	0.1915	1.646	0.1164	0.9074
TeamColts	0.8931	1.57	0.569	0.5694
TeamCowboys	1.827	1.604	1.139	0.255
TeamDolphins	-1.696	1.605	-1.057	0.2907
TeamEagles	-0.4644	1.581	-0.2937	0.769
TeamFalcons	-0.2642	1.666	-0.1586	0.874
TeamGiants	-1.118	1.629	-0.6863	0.4926
TeamJaguars	-2.576	1.761	-1.463	0.1438
TeamJets	-1.365	1.603	-0.8519	0.3944
TeamLions	-0.9962	1.607	-0.62	0.5353
TeamPackers	0.1025	1.539	0.06659	0.9469
TeamPanthers	0.9178	1.714	0.5353	0.5925
TeamPatriots	2.646	1.688	1.567	0.1172
TeamRaiders	-1.308	1.561	-0.8383	0.402
TeamRams	-0.3872	1.659	-0.2333	0.8155
TeamRavens	0.7631	1.537	0.4966	0.6196
TeamRedskins	-0.1392	1.586	-0.08775	0.9301
TeamSaints	2.277	1.774	1.283	0.1997
TeamSeahawks	1.04	1.587	0.6555	0.5122
TeamSteelers	0.7799	1.607	0.4852	0.6276
TeamTexans	-1.278	1.578	-0.8097	0.4182
TeamTitans	-0.5597	1.604	-0.3489	0.7272
TeamVikings	1.029	1.598	0.6443	0.5195
PositionDB:X40ydCent	12.57	9.631	1.305	0.1922
PositionDE:X40ydCent	-3.14	9.291	-0.338	0.7354
PositionDT:X40ydCent	1.917	8.788	0.2181	0.8274
PositionG:X40ydCent	-1.552	9.75	-0.1592	0.8735
PositionLB:X40ydCent	8.101	9.385	0.8633	0.3881
PositionQB:X40ydCent	-3.647	10.02	-0.3638	0.716
PositionRB:X40ydCent	17.81	9.881	1.803	0.07166
PositionT:X40ydCent	2.303	8.826	0.2609	0.7942
PositionTE:X40ydCent	3.34	10.21	0.3272	0.7436
PositionWR:X40ydCent	17.59	10.26	1.714	0.08673

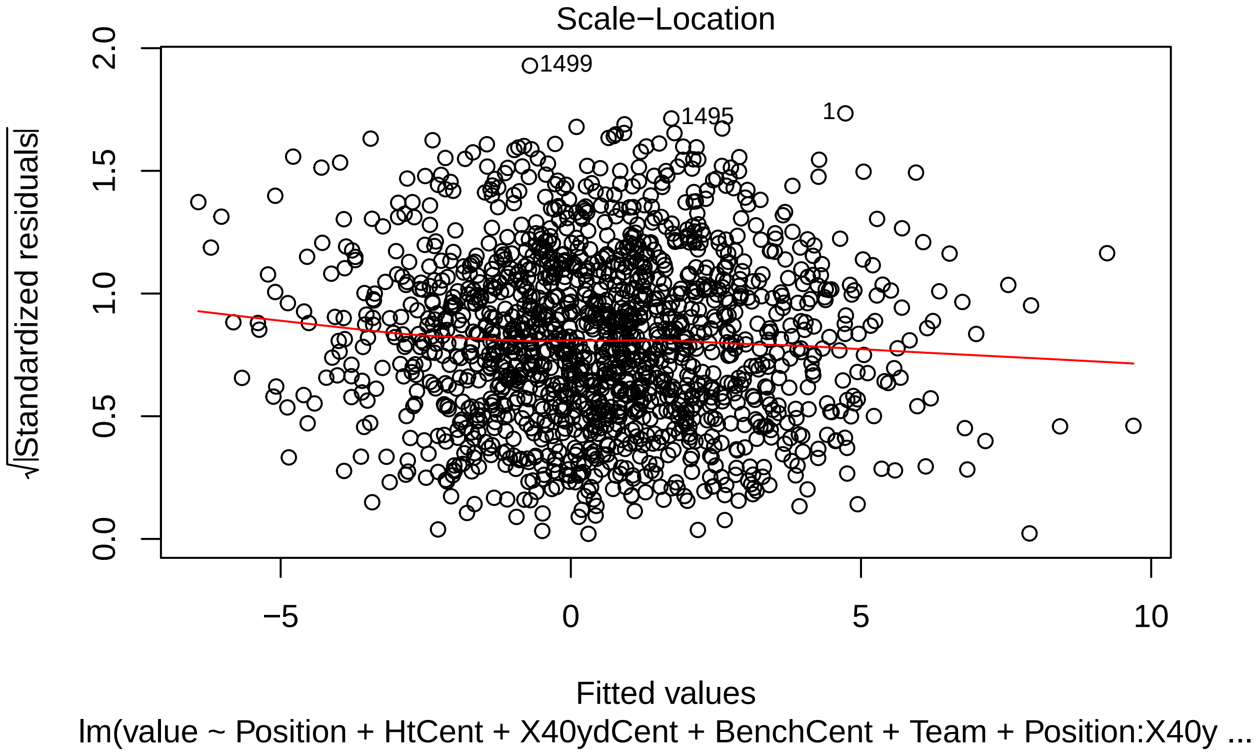
Table 16: Fitting linear model: value ~ Position + HtCent + X40ydCent + BenchCent + Team + Position:X40ydCent

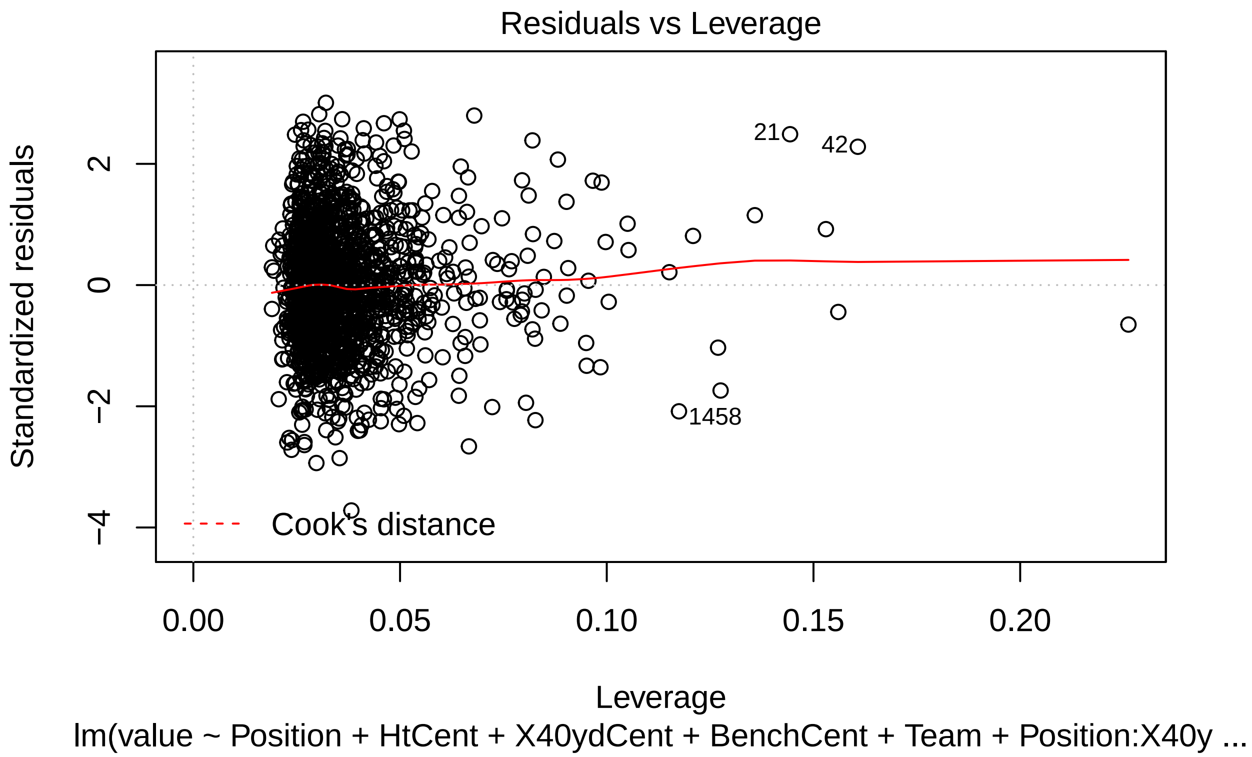
Observations	Residual Std. Error	R^2	Adjusted R^2
1499	8.032	0.0759	0.04134

```r
plot(model4)
```




Normal Q−Q
Standardized residuals
2
0
−2
−4
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(value ~ Position + HtCent + X40ydCent + BenchCent + Team + Position:X40y ...
1
1495
1499


Scale−Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
−5
0
5
10
Fitted values
lm(value ~ Position + HtCent + X40ydCent + BenchCent + Team + Position:X40y ...
1499
1495
1



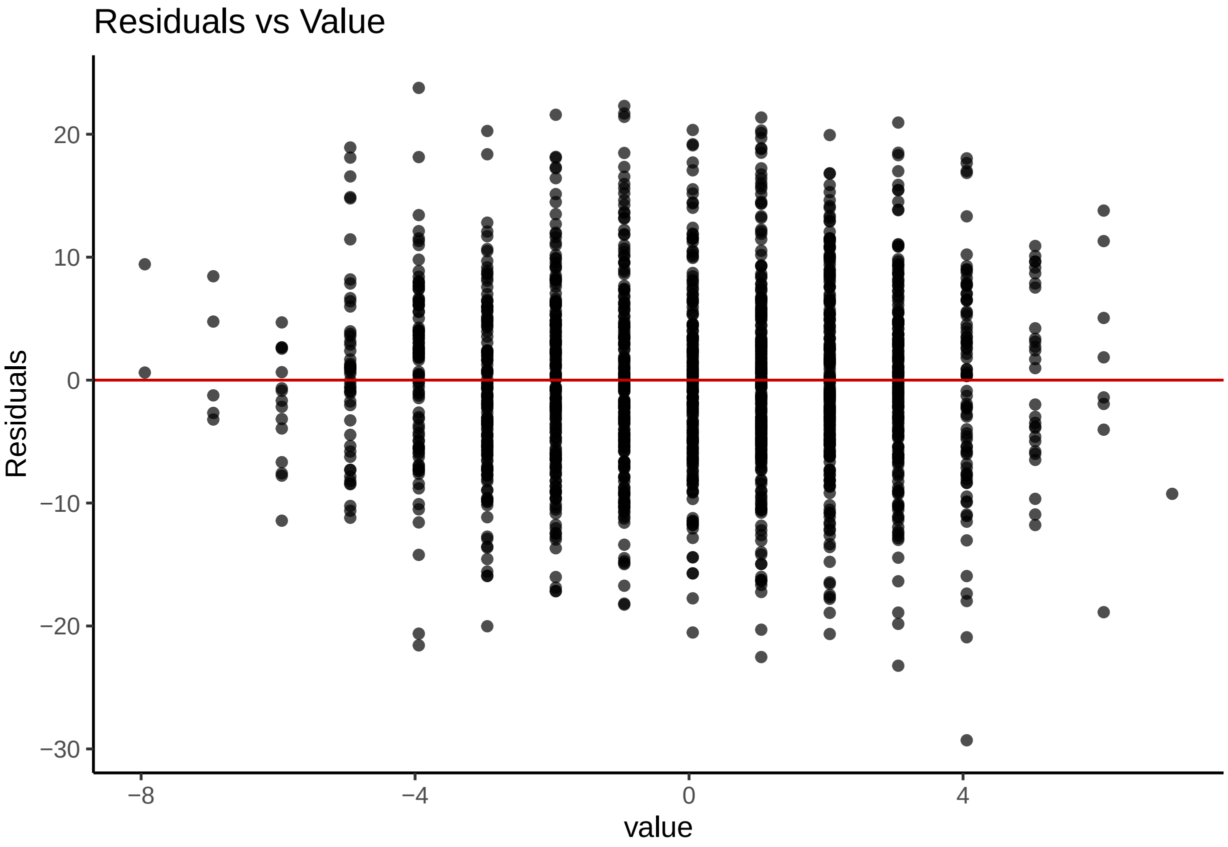
```
confint(model4)
```

```
##                        2.5 %      97.5 %
## (Intercept)      -6.49642036  9.86155544
## PositionDB       -7.61441239  9.33282127
## PositionDE       -9.29425942  6.58468495
## PositionDT       -9.12023062  7.41176616
## PositionG        -7.82293484 11.84532090
## PositionLB       -9.57511655  6.35359338
## PositionQB      -10.94127801  5.38464138
## PositionRB       -3.78161818 12.99270805
## PositionT        -9.44084344  7.95058911
## PositionTE       -6.56027707  9.49565403
## PositionWR       -3.78068649 13.76043036
## HtCent           -0.51049968  0.01731099
## X40ydCent       -18.34871205 12.39967723
## BenchCent         0.03455271  0.23240968
## TeamBears        -4.98657254  1.59936545
## TeamBengals      -4.08019458  2.07422988
## TeamBills        -4.26320741  1.96298112
## TeamBroncos      -3.42320390  2.74953768
## TeamBrowns       -5.80078831  0.02496064
## TeamBuccaneers   -5.06067726  1.32505789
## TeamCardinals    -6.66257284 -0.32300471
## TeamChargers     -3.90834945  2.70468905
## TeamChiefs       -3.03713011  3.42019194
## TeamColts        -2.18568886  3.97191851
## TeamCowboys      -1.31972342  4.97292837
## TeamDolphins     -4.84362694  1.45162851
## TeamEagles       -3.56639843  2.63750655
## TeamFalcons           -3.53148005  3.00303095
## TeamGiants            -4.31308638  2.07719253
## TeamJaguars           -6.03119271  0.87867404
## TeamJets              -4.50897370  1.77839799
## TeamLions             -4.14772788  2.15538111
## TeamPackers           -2.91635590  3.12131045
## TeamPanthers          -2.44524360  4.28089537
## TeamPatriots          -0.66539779  5.95673349
## TeamRaiders           -4.36950969  1.75297102
## TeamRams              -3.64209782  2.86776679
## TeamRavens            -2.25142506  3.77757549
## TeamRedskins          -3.25094011  2.97255545
## TeamSaints            -1.20424377  5.75736005
## TeamSeahawks          -2.07287316  4.15364866
## TeamSteelers          -2.37324776  3.93295063
## TeamTexans            -4.37391832  1.81802628
## TeamTitans            -3.70617385  2.58673330
## TeamVikings           -2.10446371  4.16298458
## PositionDB:X40ydCent  -6.32578582 31.45815366
## PositionDE:X40ydCent -21.36628790 15.08555145
## PositionDT:X40ydCent -15.32190644 19.15491552
## PositionG:X40ydCent  -20.67875671 17.57423029
## PositionLB:X40ydCent -10.30726098 26.51019630
## PositionQB:X40ydCent -23.30845796 16.01514325
## PositionRB:X40ydCent  -1.57142442 37.19298307
## PositionT:X40ydCent  -15.00952377 19.61520934
## PositionTE:X40ydCent -16.68513967 23.36536500
## PositionWR:X40ydCent  -2.54106897 37.72938055
```

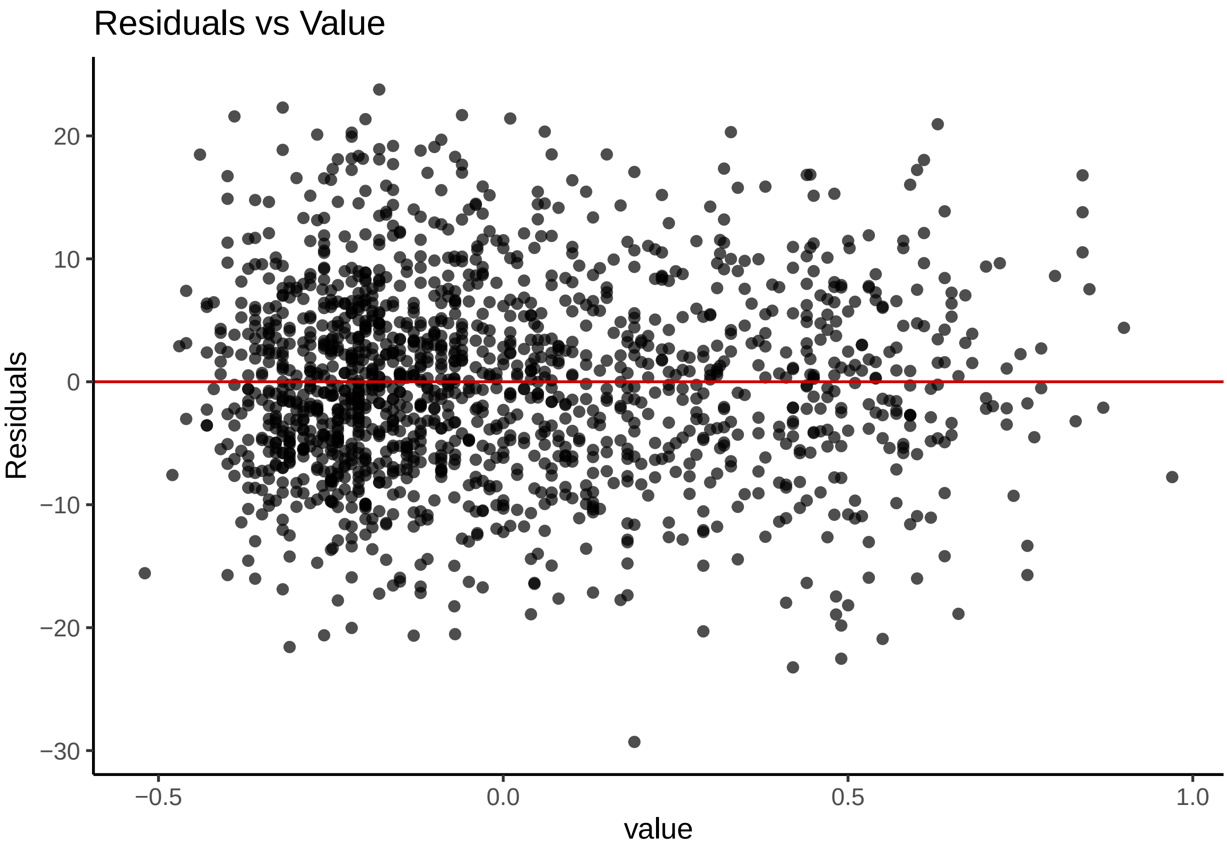
```
vif(model4)
```

```
##           PositionDB           PositionDE           PositionDT
##            67.114328            32.433606            36.711386
##            PositionG           PositionLB           PositionQB
##            28.804044            42.773139            14.965126
##           PositionRB            PositionT           PositionTE
##            36.882899            32.597476            23.777615
##           PositionWR               HtCent            X40ydCent
##            52.101241             2.944697           124.563178
##            BenchCent            TeamBears          TeamBengals
##             2.249789             1.700527             1.914608
##            TeamBills          TeamBroncos           TeamBrowns
##             1.850298             1.890288             2.093961
##       TeamBuccaneers        TeamCardinals         TeamChargers
##             1.792387             1.804579             1.714551
##           TeamChiefs            TeamColts          TeamCowboys
##             1.753751             1.881030             1.815369
##         TeamDolphins           TeamEagles          TeamFalcons
##             1.816872             1.873276             1.714749
##           TeamGiants          TeamJaguars             TeamJets
##             1.794939             1.597677             1.812324
##            TeamLions          TeamPackers         TeamPanthers
##             1.821408             1.944918             1.644046
##         TeamPatriots          TeamRaiders             TeamRams
##             1.719269             1.894784             1.742151
##            TeamRavens          TeamRedskins            TeamSaints
##              1.939339              1.848698              1.621692
##          TeamSeahawks          TeamSteelers            TeamTexans
##              1.850496              1.823194              1.866060
##            TeamTitans           TeamVikings PositionDB:X40ydCent
##              1.815517              1.837906             23.427840
## PositionDE:X40ydCent PositionDT:X40ydCent  PositionG:X40ydCent
##              3.828263             20.734058             30.420507
## PositionLB:X40ydCent PositionQB:X40ydCent PositionRB:X40ydCent
##              4.470153              2.934514             10.935307
##  PositionT:X40ydCent PositionTE:X40ydCent PositionWR:X40ydCent
##             30.439626              2.676618             21.347896
```

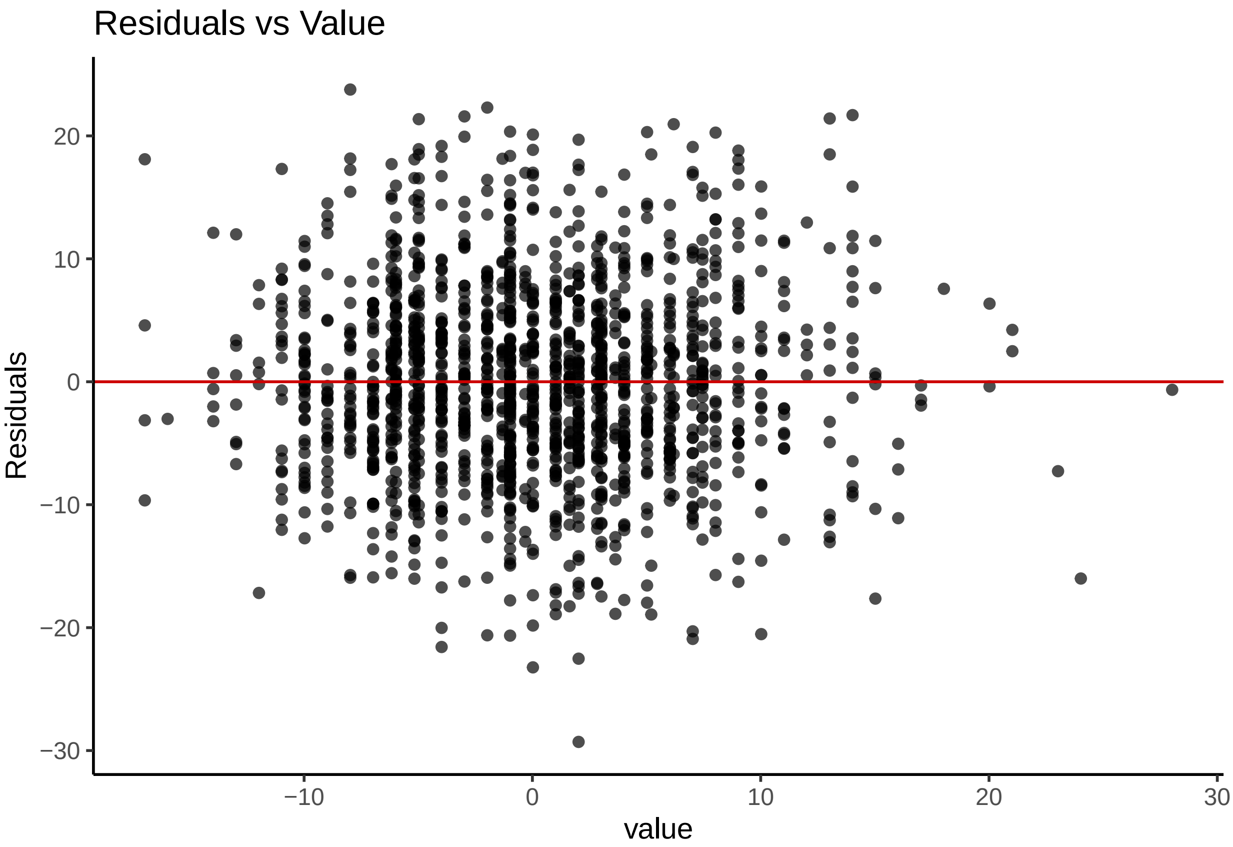
```
ggplot(df,aes(x=HtCent, y=model4$residual)) +
geom_point(alpha = .7) + geom_hline(yintercept=0,col="red3") + theme_classic() +
labs(title="Residuals vs Value",x="value",y="Residuals")
```



```
ggplot(df,aes(x=X40ydCent, y=model4$residual)) +
geom_point(alpha = .7) + geom_hline(yintercept=0,col="red3") + theme_classic() +
labs(title="Residuals vs Value",x="value",y="Residuals")
```



```
ggplot(df,aes(x=BenchCent, y=model4$residual)) +
geom_point(alpha = .7) + geom_hline(yintercept=0,col="red3") + theme_classic() +
labs(title="Residuals vs Value",x="value",y="Residuals")
```


Residuals vs Value
Residuals
20
10
0
−10
−20
−30
value
−10
0
10
20
30