NFL Linear Regression Model

Sean Lussmyer

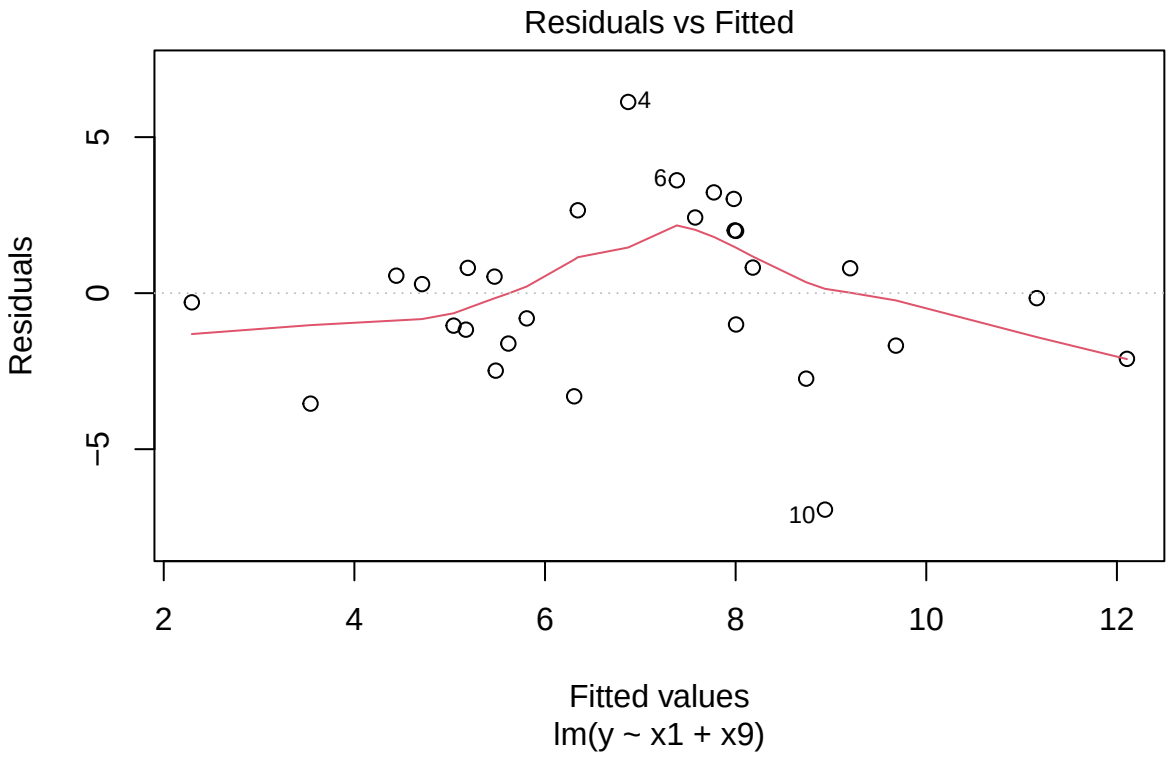
2024-12-06

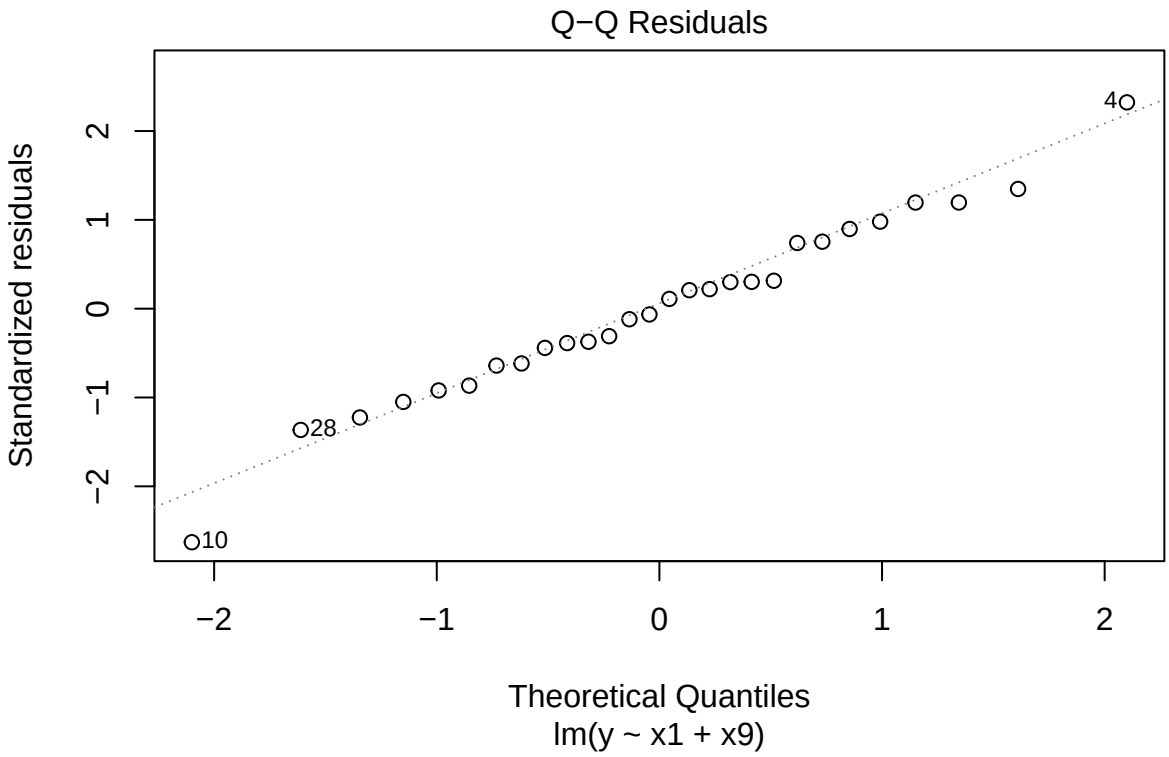
1.)

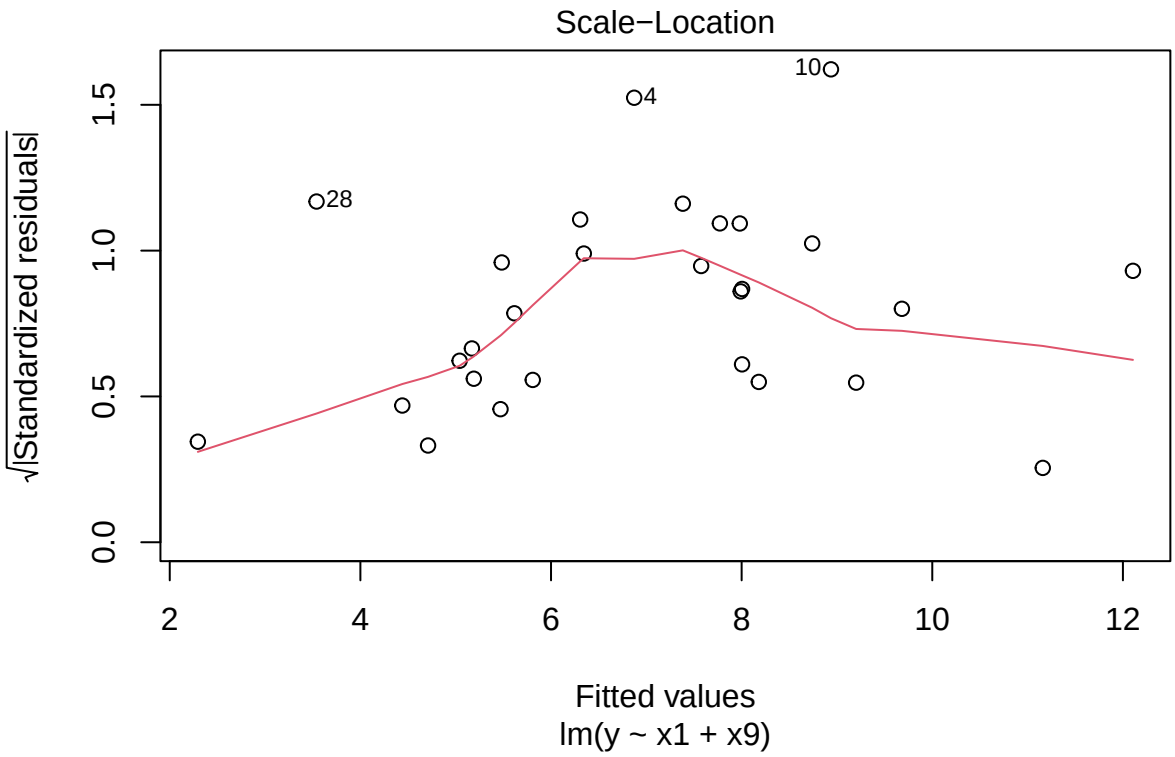
```
library("readxl")
 nfl <- read_excel("/Users/seanlusuer/Downloads/data_nfl.XLS")
nflmodel <- lm(y ~ x1+x9, data = nfl)
summary(nflmodel)
```

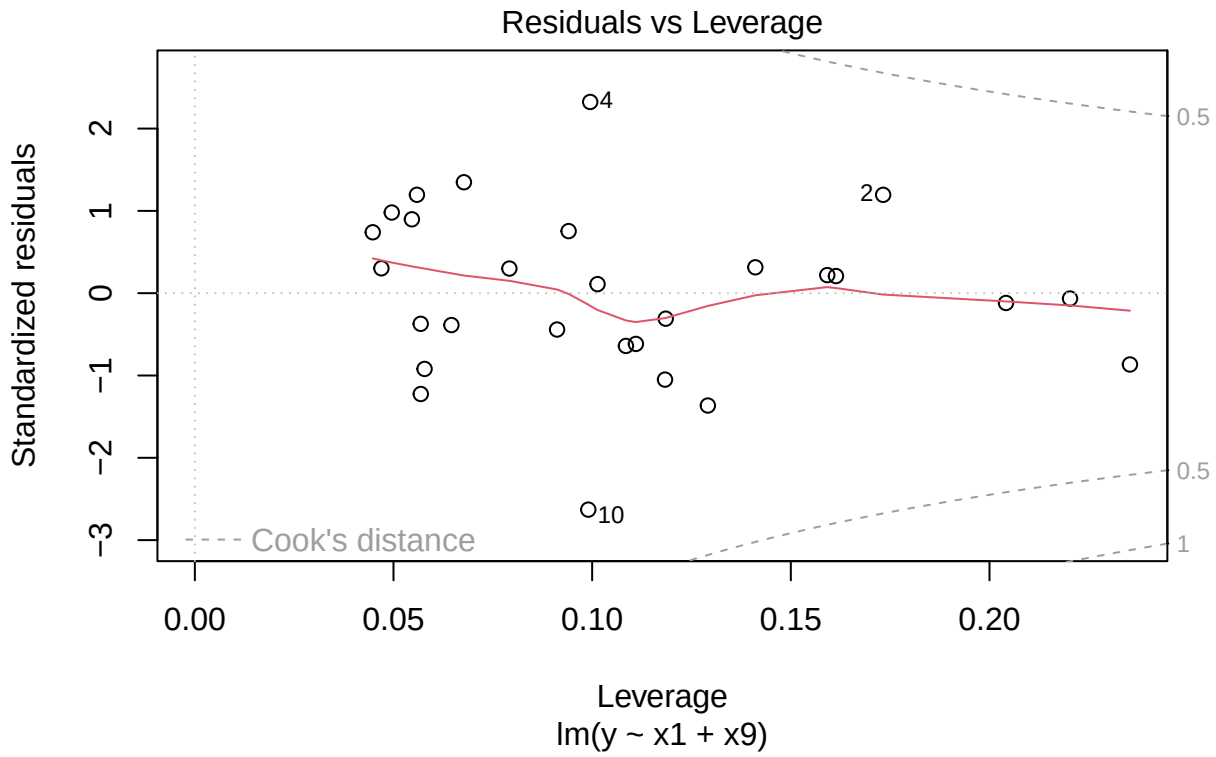
```
##
## Call:
## lm(formula = y ~ x1 + x9, data = nfl)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9372 -1.6319  0.0652  1.9991  6.1273
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.198046   5.135474   0.428  0.67231
## x1           0.005112   0.001395   3.663  0.00117 **
## x9          -0.002829   0.001815  -1.559  0.13157
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.779 on 25 degrees of freedom
## Multiple R-squared:  0.4094, Adjusted R-squared:  0.3621
## F-statistic: 8.663 on 2 and 25 DF,  p-value: 0.001386
```

```
plot(nflmodel)
```


Residuals vs Fitted
Residuals
5
0
−5
4
6
10
2
4
6
8
10
12
Fitted values
lm(y ~ x1 + x9)


Q−Q Residuals
Standardized residuals
−2
−1
0
1
2
4
28
10
−2
−1
0
1
2
Theoretical Quantiles
lm(y ~ x1 + x9)


Scale−Location
10
4
28
√|Standardized residuals|
0.0
0.5
1.0
1.5
2
4
6
8
10
12
Fitted values
lm(y ~ x1 + x9)



Assumptions

Linearity: Looking at the residuals vs. fitted plot, we see a horizontal line without a distinct pattern satisfying the assumption of linearity.

Normality: Looking at the QQ plot, we see the points all follow a linear pattern along the dashed line satisfying the normality assumption.

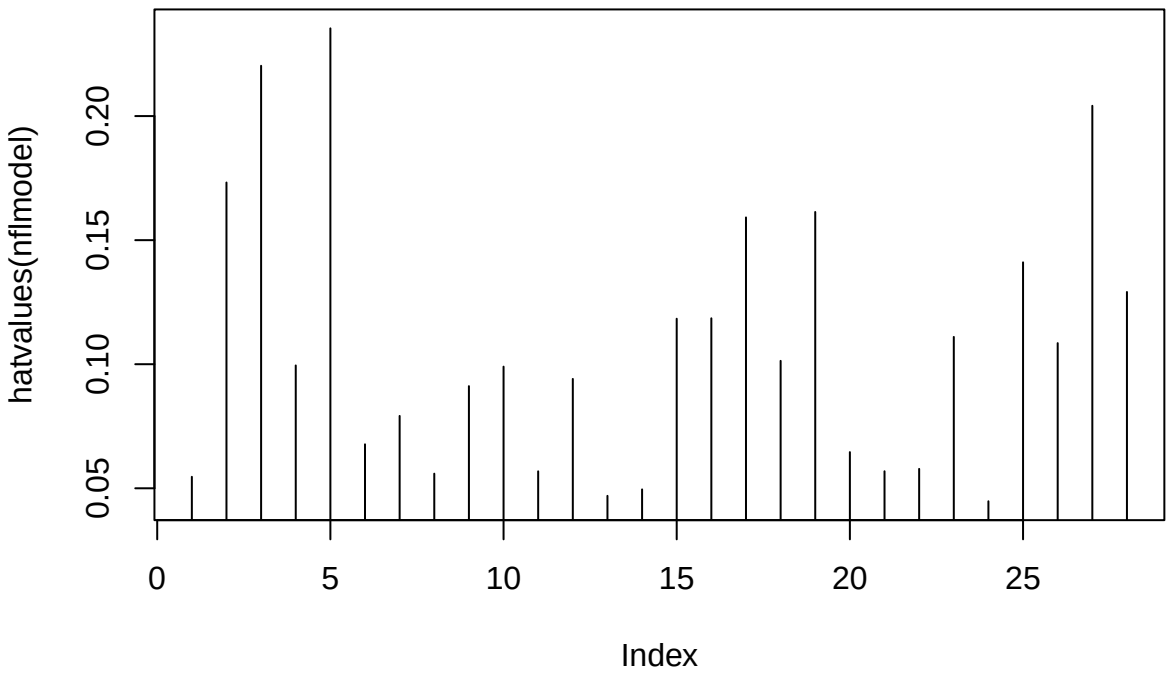
Independence of errors: Looking at the residuals vs. fitted plot, we see no pattern in the points, meaning the independence of errors assumption is satisfied.

Homoscedasticity: Looking at the scale location plot, we see the line of fit is not horizontal, and there is not an equal spread of points, meaning the homoscedasticity assumption is not satisfied.

2.) Looking at the residuals vs. leverage points, we see no points exceed 3 standard deviations, but 10 are close, meaning there are no extreme outliers.

Finding leverage points using hat values :

```
plot(hatvalues(nflmodel), type = 'h')
```




```
threshold<-2*(2/28)
threshold
```

```
## [1] 0.1428571
```

Using the threshold we calculated, we find the following points are leverage points: 2,3,5,17,19,25 and 27

To find influence points we will use Cook's distance and look for any points greater than .5.

```
cooks.distance(nflmodel)
```

```
##            1            2            3            4            5            6
## 0.0155098045 0.0997077318 0.0003973440 0.1987950643 0.0769856097 0.0439888179
##            7            8            9           10           11           12
## 0.0025697844 0.0282038617 0.0065205464 0.2533204526 0.0027806583 0.0196950777
##           13           14           15           16           17           18
## 0.0014996239 0.0166957264 0.0493114136 0.0043014452 0.0030474626 0.0004551761
##           19           20           21           22           23           24
## 0.0027793777 0.0034509707 0.0301467719 0.0173276415 0.0158162658 0.0085546894
##           25           26           27           28
## 0.0054099330 0.0166535631 0.0012089017 0.0920721543
```

Looking at the Cook's distance values we see no points greater than .5 meaning we can assume there are no influence points.

3.)

```r
nflmodel2 <- lm(y ~ x1+x2+x3+x4+x5+x6+x7+x8+x9, data = nfl)
library(car)
```

```
## Loading required package: carData
```

```r
vif(nflmodel2)
```

```
##       x1       x2       x3       x4       x5       x6       x7       x8
## 4.827645 1.420161 2.126597 1.566107 1.924035 1.275979 5.414572 4.535643
##       x9
## 1.423390
```

We see all variables have moderate correlation, with x7 having highest breaking 5. 4.)

```r
library(olsrr)
```

```
## Warning: package 'olsrr' was built under R version 4.3.3
```

```
##
## Attaching package: 'olsrr'
```

```
## The following object is masked from 'package:datasets':
##
##     rivers
```

```r
ols_eigen_cindex(nflmodel2)
```

```
##       Eigenvalue Condition Index    intercept           x1           x2
## 1  8.8267410711        1.000000 9.333334e-06 8.084778e-05 4.369059e-04
## 2  1.0209166174        2.940391 3.274536e-07 4.778823e-05 2.598711e-06
## 3  0.0562642390       12.525180 7.568911e-05 1.221448e-02 3.648643e-01
## 4  0.0368803934       15.470431 1.774055e-05 2.606080e-02 6.926393e-02
## 5  0.0246873831       18.908743 4.852774e-08 1.041246e-03 3.329609e-01
## 6  0.0185958325       21.786746 1.779154e-05 1.004190e-01 3.008558e-03
## 7  0.0110159118       28.306747 2.227891e-03 5.978962e-04 1.076070e-03
## 8  0.0032978069       51.735373 3.598318e-02 6.133248e-01 2.300935e-02
## 9  0.0010877273       90.082447 7.278534e-03 9.870097e-03 1.479898e-01
## 10 0.0005130174      131.169885 9.543895e-01 2.363431e-01 5.738752e-02
##              x3           x4           x5           x6           x7
## 1  1.524059e-05 2.379033e-04 3.122713e-06 2.230896e-04 1.931375e-05
## 2  2.959680e-07 3.936774e-07 4.980301e-01 6.088291e-06 1.797897e-06
## 3  2.743324e-04 2.282954e-02 1.867394e-02 4.603438e-02 1.875552e-03
## 4  5.011027e-04 4.630509e-02 2.504242e-01 7.460612e-03 7.685848e-04
## 5  6.082281e-04 5.276683e-01 8.309127e-03 3.761368e-02 3.307384e-04
## 6  4.301436e-04 3.443119e-02 2.725572e-02 5.449214e-01 1.912433e-03
## 7  9.551958e-03 2.175218e-03 9.660479e-03 9.659016e-02 2.423056e-03
## 8  6.732725e-03 9.231655e-03 1.844854e-02 2.381583e-01 1.438618e-01
## 9  6.420031e-01 1.359899e-02 2.425568e-02 4.106729e-03 4.009748e-01
## 10 3.398829e-01 3.435217e-01 1.449391e-01 2.488560e-02 4.478319e-01
##              x8           x9
```

```
## 1  7.707348e-05 1.633144e-04
## 2  6.433271e-05 1.550536e-05
## 3  7.900256e-04 1.025167e-03
## 4  8.404667e-02 4.250601e-02
## 5  9.959627e-04 2.309975e-03
## 6  6.202058e-04 5.012798e-02
## 7  7.362533e-02 7.819792e-01
## 8  2.504348e-01 4.579697e-02
## 9  5.853927e-01 6.843910e-02
## 10 3.952922e-03 7.636772e-03
```

Looking at the table above, we have 3 condition indices above 30, meaning there is a strong sign of multi-collinearity. Looking at the first case where the condition index equals 51.735373 based on the variance decomposition proportions, we see x1 has the highest value of .6 and the next highest ones are x6 = .238, x7= .143 and x8 = .250, meaning there is a very small linear relationship between the variables. Looking at the next condition index of 90.082, based on the variance decomposition proportions, we see x3 has the highest value of .64 and the next highest ones are x7 = .4 and x8= .58, meaning there is a small linear relationship between the variables. Looking at the last condition index of 131.169885 , based on the variance decomposition proportions, we see x3 has the highest value of .64 and the next highest ones are x7 = .4 and x8= .58, meaning there is a small linear relationship between the variables. Based on the vif scores and variance decomposition proportions, there is some multi-collinearity but not a significant amount.

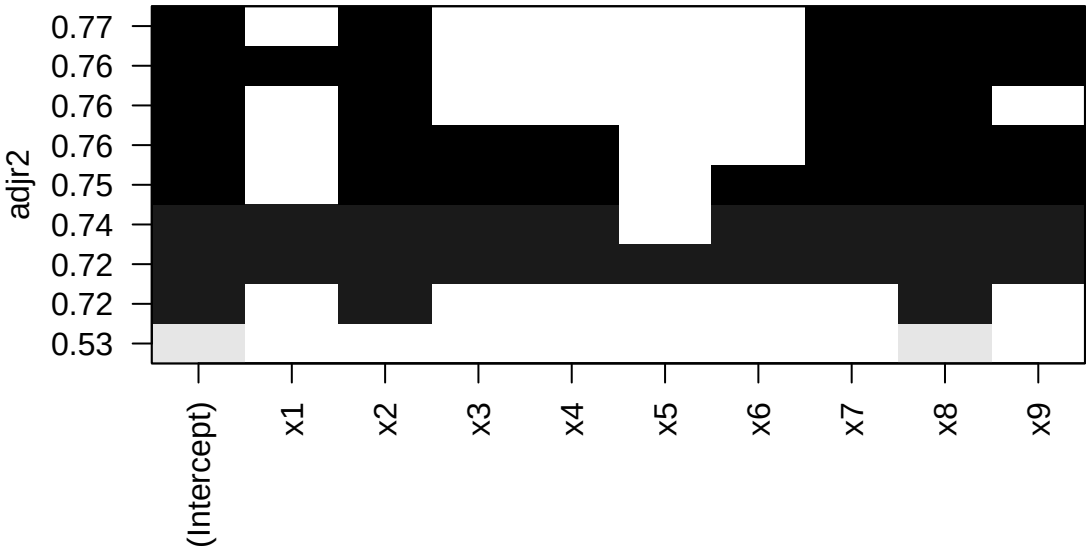
5.)

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.3.3
```

```
nflsubsets <- regsubsets(y ~x1+x2+x3+x4+x5+x6+x7+x8+x9 , data = nfl, nvmax = ncol(nfl)-1)
nflregsum <- summary(nflsubsets)
```

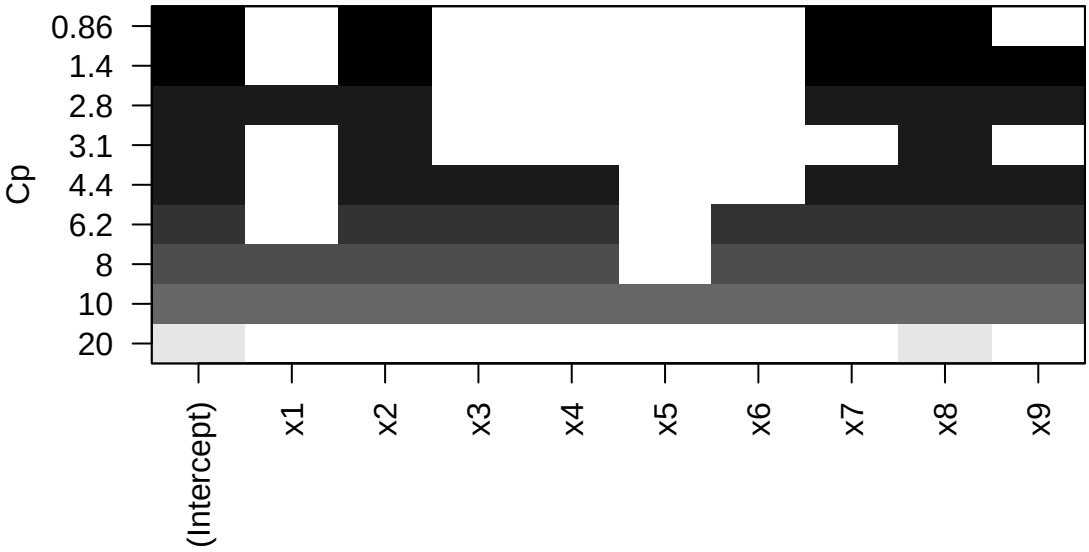
```
plot(nflsubsets, scale = "adjr2")
```




```
coef(nflsubsets, 4)
```

```
##  (Intercept)           x2           x7           x8           x9
## -1.821703427  0.003818572  0.216894094 -0.004014887 -0.001634926
```

```
plot(nflsubsets, scale = "Cp")
```




```
coef(nflsubsets, 3)
```

```
##  (Intercept)           x2           x7           x8
## -1.808372059  0.003598070  0.193960210 -0.004815494
```

We see that the subset of x2,x7,x8 andx9 have the highest Adjusted R-square value. Then we checked the Cp statistics and found that the subsets x2,x7 and x8 had the lowest score and when we look back at the Adjusted R-square for this graph we see that it is only .01 less than the highest subset, suggesting that x2, x7 and x8 make up the best subset model. 6.)

```
start <- lm(y~1,data = nfl)
step(start, direction = "forward", scope = formula(nflmodel2))
```

```
## Start:  AIC=70.81
## y ~ 1
##
##        Df Sum of Sq    RSS    AIC
## + x8    1   178.092 148.87 50.785
## + x1    1   115.068 211.90 60.669
## + x7    1    97.238 229.73 62.931
## + x5    1    86.116 240.85 64.255
## + x2    1    76.193 250.77 65.385
## + x9    1    30.167 296.80 70.104
## <none>               326.96 70.814
```

```
## + x4    1    21.844 305.12 70.878
## + x6    1    16.411 310.55 71.372
## + x3    1     2.135 324.83 72.631
##
## Step:  AIC=50.78
## y ~ x8
##
##        Df Sum of Sq     RSS    AIC
## + x2    1    64.934  83.938 36.741
## + x5    1    11.607 137.265 50.512
## <none>              148.872 50.785
## + x1    1     6.636 142.236 51.508
## + x3    1     6.368 142.504 51.561
## + x4    1     6.345 142.527 51.565
## + x7    1     0.974 147.898 52.601
## + x6    1     0.487 148.385 52.693
## + x9    1     0.008 148.864 52.783
##
## Step:  AIC=36.74
## y ~ x8 + x2
##
##        Df Sum of Sq    RSS    AIC
## + x7    1   14.0682 69.870 33.604
## + x1    1   11.1905 72.748 34.734
## + x3    1    8.9010 75.037 35.602
## + x5    1    5.8147 78.124 36.730
## <none>              83.938 36.741
## + x9    1    2.0256 81.913 38.057
## + x6    1    1.3216 82.617 38.296
## + x4    1    0.0161 83.922 38.735
##
## Step:  AIC=33.6
## y ~ x8 + x2 + x7
##
##        Df Sum of Sq    RSS    AIC
## + x9    1    4.8657 65.004 33.583
## <none>              69.870 33.604
## + x3    1    1.3873 68.483 35.043
## + x4    1    0.9792 68.891 35.209
## + x1    1    0.9022 68.968 35.240
## + x6    1    0.4879 69.382 35.408
## + x5    1    0.2987 69.571 35.484
##
## Step:  AIC=33.58
## y ~ x8 + x2 + x7 + x9
##
##        Df Sum of Sq    RSS    AIC
## <none>              65.004 33.583
## + x1    1   1.86452 63.140 34.768
## + x4    1   1.74260 63.262 34.822
## + x3    1   0.70148 64.303 35.279
## + x6    1   0.45071 64.554 35.388
## + x5    1   0.32667 64.678 35.442
```

```
##
## Call:
## lm(formula = y ~ x8 + x2 + x7 + x9, data = nfl)
##
## Coefficients:
## (Intercept)           x8           x2           x7           x9
##   -1.821703    -0.004015     0.003819     0.216894    -0.001635
```

Looking at the results of using forward elimination, we see the best subset model is made up of x2,x7,x8,x9. 7.)

```
step(nflmodel2, direction = "backward")
```

```
## Start:  AIC=41.48
## y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9
##
##        Df Sum of Sq     RSS    AIC
## - x5    1     0.000  60.293 39.476
## - x1    1     0.549  60.842 39.730
## - x3    1     0.746  61.039 39.821
## - x6    1     0.803  61.096 39.847
## - x4    1     1.968  62.261 40.376
## - x7    1     3.451  63.744 41.035
## <none>                60.293 41.476
## - x9    1     5.348  65.642 41.856
## - x8    1    12.072  72.365 44.587
## - x2    1    62.448 122.741 59.380
##
## Step:  AIC=39.48
## y ~ x1 + x2 + x3 + x4 + x6 + x7 + x8 + x9
##
##        Df Sum of Sq     RSS    AIC
## - x1    1     0.553  60.846 37.732
## - x3    1     0.750  61.043 37.822
## - x6    1     0.818  61.111 37.854
## - x4    1     2.053  62.346 38.414
## - x7    1     3.859  64.152 39.213
## <none>                60.293 39.476
## - x9    1     5.351  65.644 39.857
## - x8    1    12.086  72.379 42.592
## - x2    1    66.979 127.272 58.395
##
## Step:  AIC=37.73
## y ~ x2 + x3 + x4 + x6 + x7 + x8 + x9
##
##        Df Sum of Sq     RSS    AIC
## - x6    1     0.690  61.536 36.048
## - x3    1     1.715  62.561 36.510
## - x4    1     3.051  63.897 37.102
## <none>                60.846 37.732
## - x9    1     4.852  65.698 37.880
## - x7    1     8.961  69.807 39.579
## - x8    1    16.599  77.445 42.486
```

```
## - x2    1    67.010 127.856 56.524
##
## Step:  AIC=36.05
## y ~ x2 + x3 + x4 + x7 + x8 + x9
##
##        Df Sum of Sq     RSS    AIC
## - x3    1     1.726  63.262 34.822
## - x4    1     2.767  64.303 35.279
## <none>               61.536 36.048
## - x9    1     4.831  66.367 36.164
## - x7    1     9.390  70.926 38.024
## - x8    1    18.314  79.851 41.343
## - x2    1    66.447 127.984 54.552
##
## Step:  AIC=34.82
## y ~ x2 + x4 + x7 + x8 + x9
##
##        Df Sum of Sq     RSS    AIC
## - x4    1     1.743  65.004 33.583
## <none>               63.262 34.822
## - x9    1     5.629  68.891 35.209
## - x8    1    17.701  80.962 39.730
## - x7    1    18.583  81.845 40.033
## - x2    1    75.598 138.860 54.835
##
## Step:  AIC=33.58
## y ~ x2 + x7 + x8 + x9
##
##        Df Sum of Sq     RSS    AIC
## <none>               65.004 33.583
## - x9    1     4.866  69.870 33.604
## - x7    1    16.908  81.913 38.057
## - x8    1    23.299  88.303 40.160
## - x2    1    82.892 147.897 54.601


##
## Call:
## lm(formula = y ~ x2 + x7 + x8 + x9, data = nfl)
##
## Coefficients:
## (Intercept)           x2           x7           x8           x9
##   -1.821703     0.003819     0.216894    -0.004015    -0.001635
```

Using backward elimination we see the best subset model is made up of x2,x7,x8,x9. 8.)

```
nflmodel3 <- lm(y ~ x2+x7+x8+x9, data = nfl)
summary(nflmodel3)
```

```
##
## Call:
## lm(formula = y ~ x2 + x7 + x8 + x9, data = nfl)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -3.3519 -0.5612 -0.0856  0.6972  3.2802
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.8217034  7.7847061  -0.234  0.81705
## x2           0.0038186  0.0007051   5.416 1.67e-05 ***
## x7           0.2168941  0.0886759   2.446  0.02252 *
## x8          -0.0040149  0.0013983  -2.871  0.00863 **
## x9          -0.0016349  0.0012460  -1.312  0.20244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.681 on 23 degrees of freedom
## Multiple R-squared:  0.8012, Adjusted R-squared:  0.7666
## F-statistic: 23.17 on 4 and 23 DF,  p-value: 8.735e-08
```

In my judgment, the Y~ x2+x7+x8+x9 is the best subset model. This is because it has the highest Adjusted R-squared, and the second lowest cp score of 1.4, which is pretty low relative to the number of predictors. This was validated through forward and backward selection. This subset is better than using a full model because it reduces overfitting, increases efficiency and allows us to interpret the relationship between the predictors and the response variable. There are outside factors that would affect the ability of the model to predict future games, such as changes to rules, equipment and players. The models do have strong summary statics with a p-value of 8.735e-08, Adjusted R-squared = 0.7666 and Multiple R-squared = 0.8012. This model can be used for forecasting games within a couple of seasons, but after that it's tricky due to the constant change of factors.