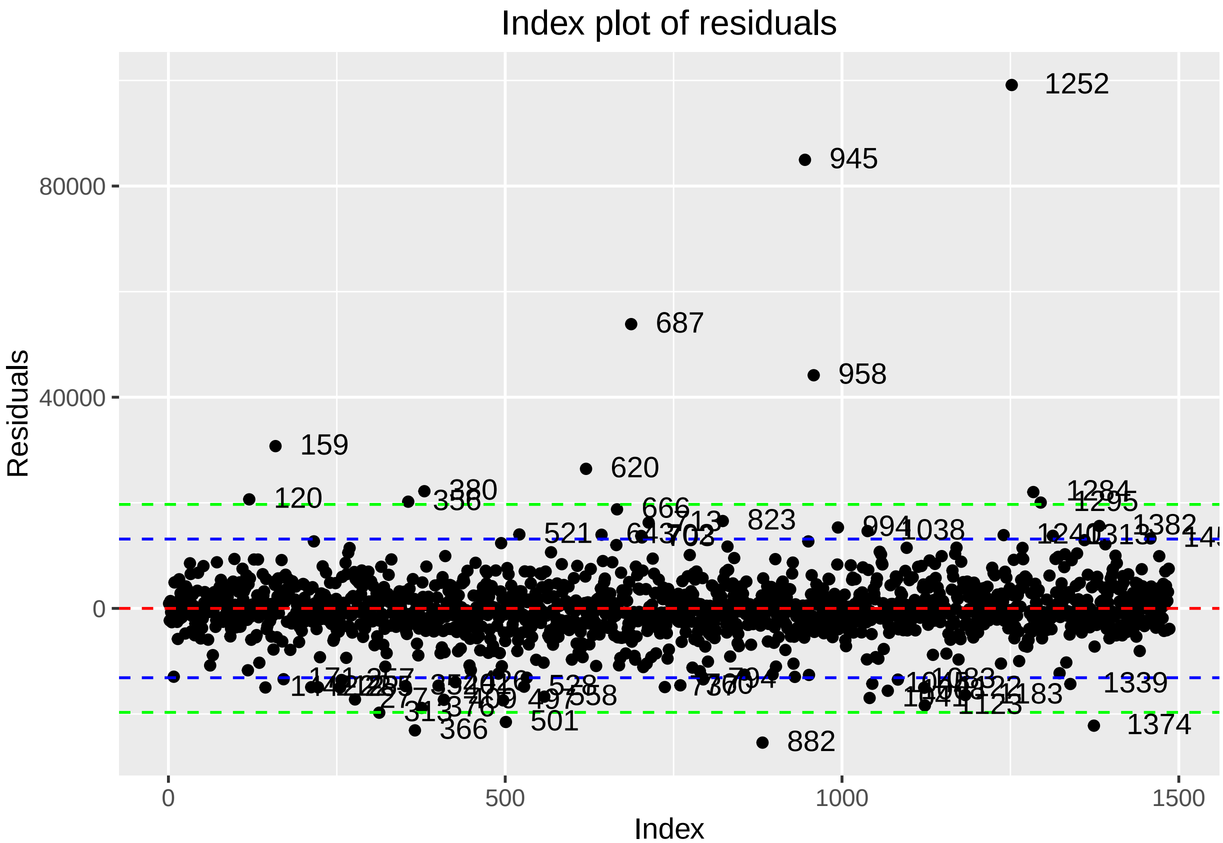
Frame5

Ottó Hólm Reynisson

15.september 2016

Residuals

Begin by looking at the residuals from this model

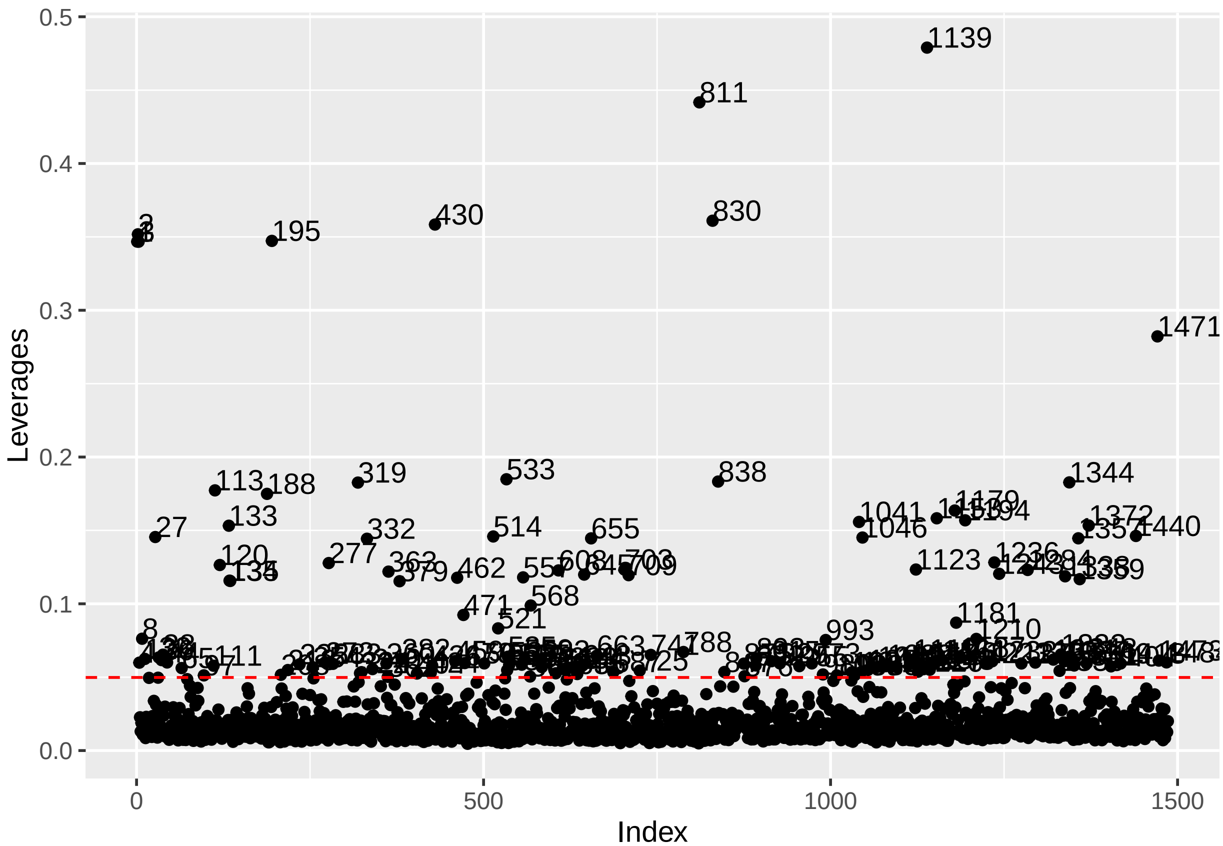


Mynd 1: Indexplot for the residuals.

Here the blue and green line represent 2 and 3 standard deviations from the mean. We identify those points that are two standard deviations away from the mean. We clearly see that there are some possible outliers that need further diagnostics.

Leverages

The next thing to do is looking at the leverages, that is the measure of how far independent variable values of an observation are from those of the other observation. Figure two marks those points that are more than $\frac{2p}{n} = \frac{2 \cdot 37}{1486} \approx 0.05$

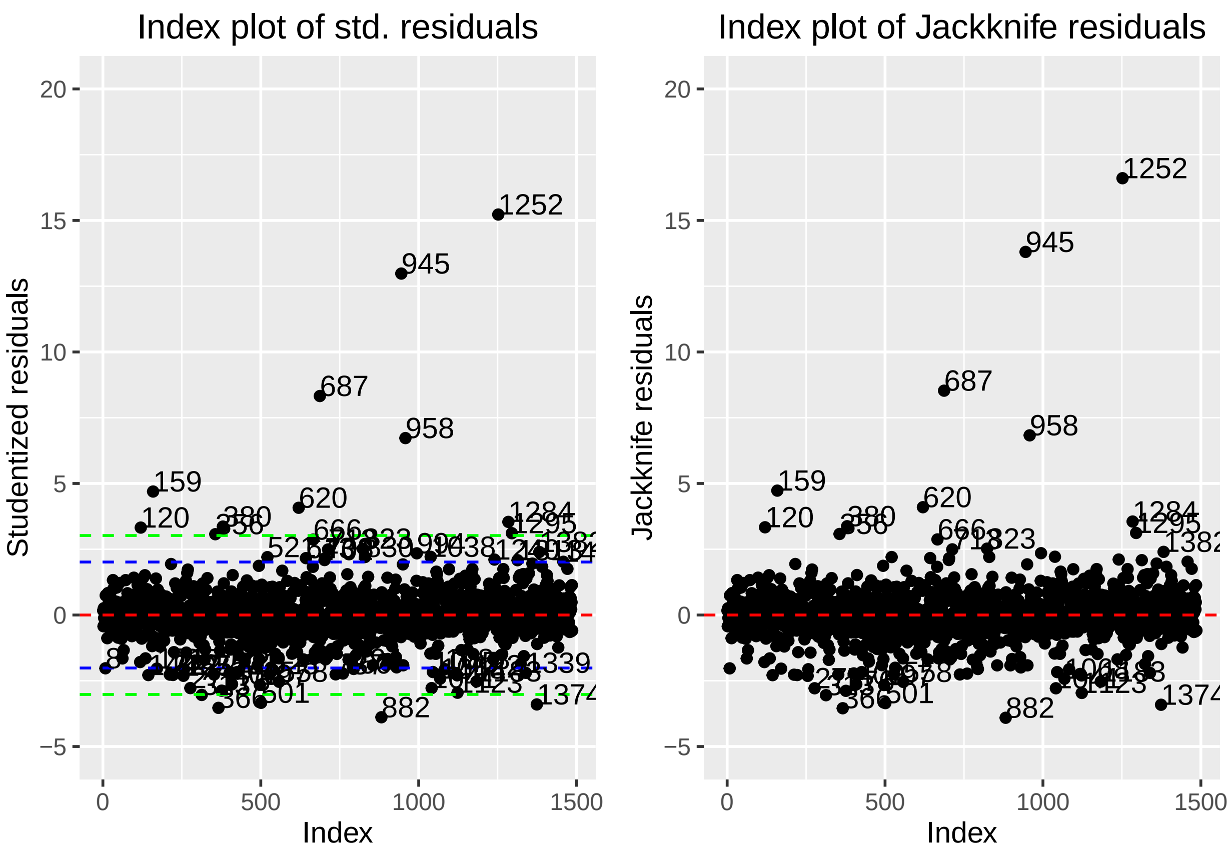


Mynd 2: Indexplot of leverages

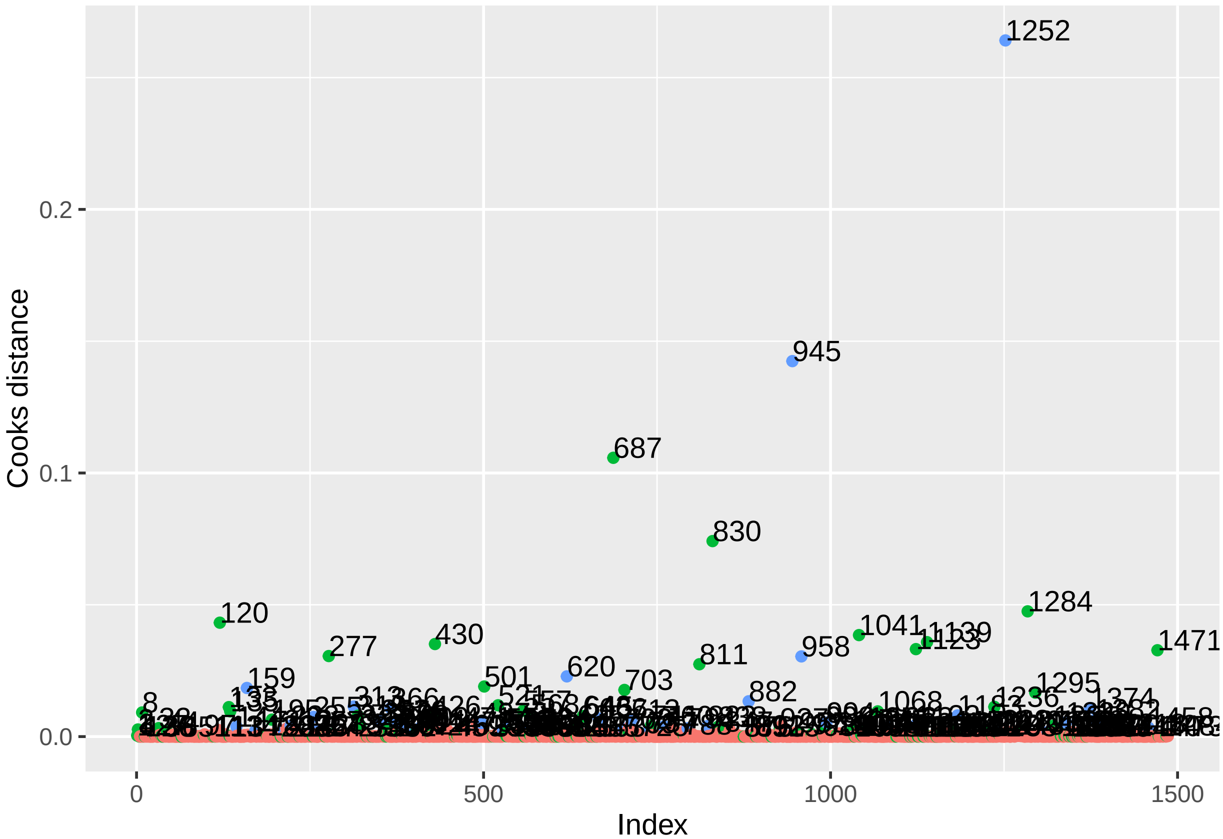
Studentized residuals

Studentized residuals are sometimes preferred in residual plots as they have been standardized to have equal variance. They are also a big part in the Jackkiife residulas that follows

Blue and green line represent as before 2 and 3 sd from the mean.

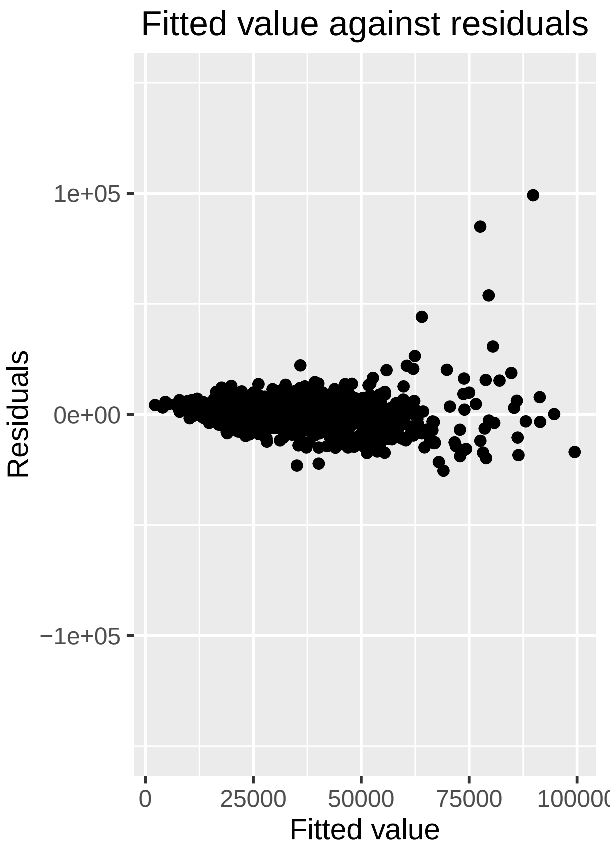


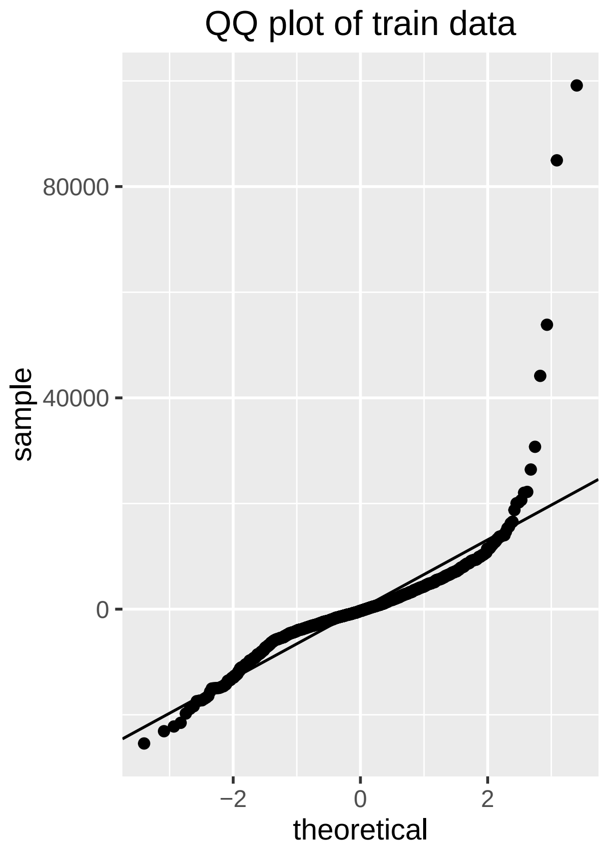
Cook's distance (calculated w.r.t Jackknife and Std.Residuals) is a good way to diagnose influential points in the model. Points with high Cooks distance are affecting the model more than the others. The green points have high Cooks distance, but the blue points have high Cooks distance but also high leverage.



To see how well the model fits the data, we plot the fitted value against residuals. This should be scatterplot with no specified form.

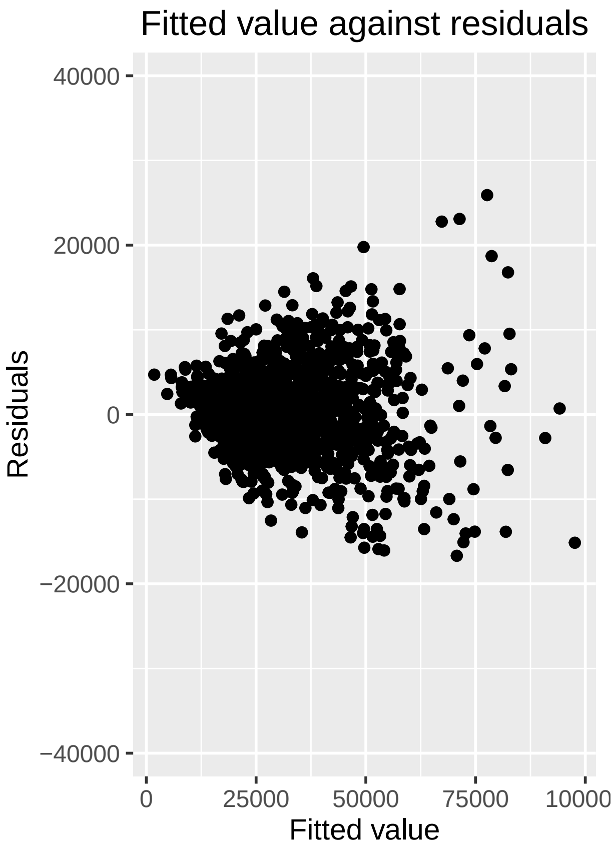
We clearly see this is not what we expected to see. Also the QQ plot This means we have to do some transformation and remove the biggest outliers.

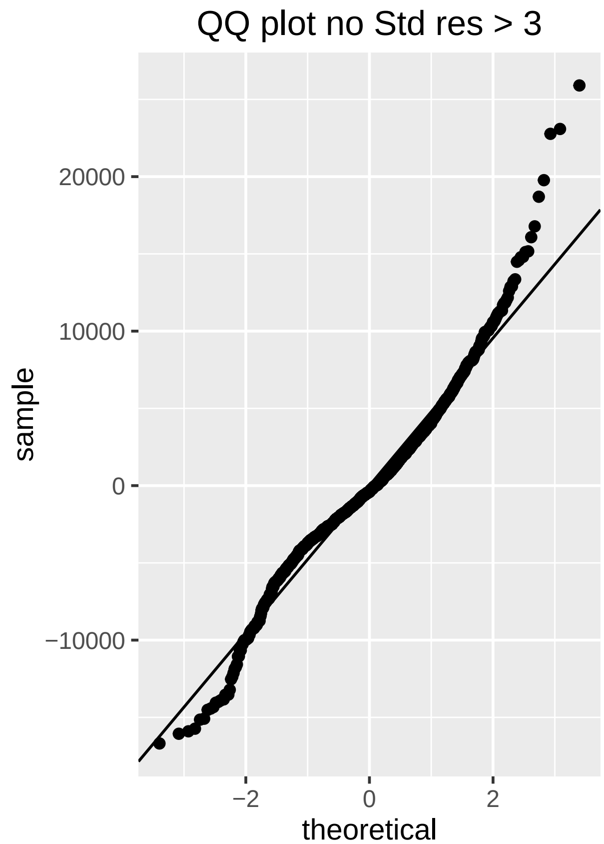




First we calculate the R-adjusted for the first model and the whole train data and get $R - adjusted$= 0.7947532. The R-squared for this data set and model is, $R - squared$=0.8252761 Now by removing the residuals that have std. residuals > 3 we have new model.

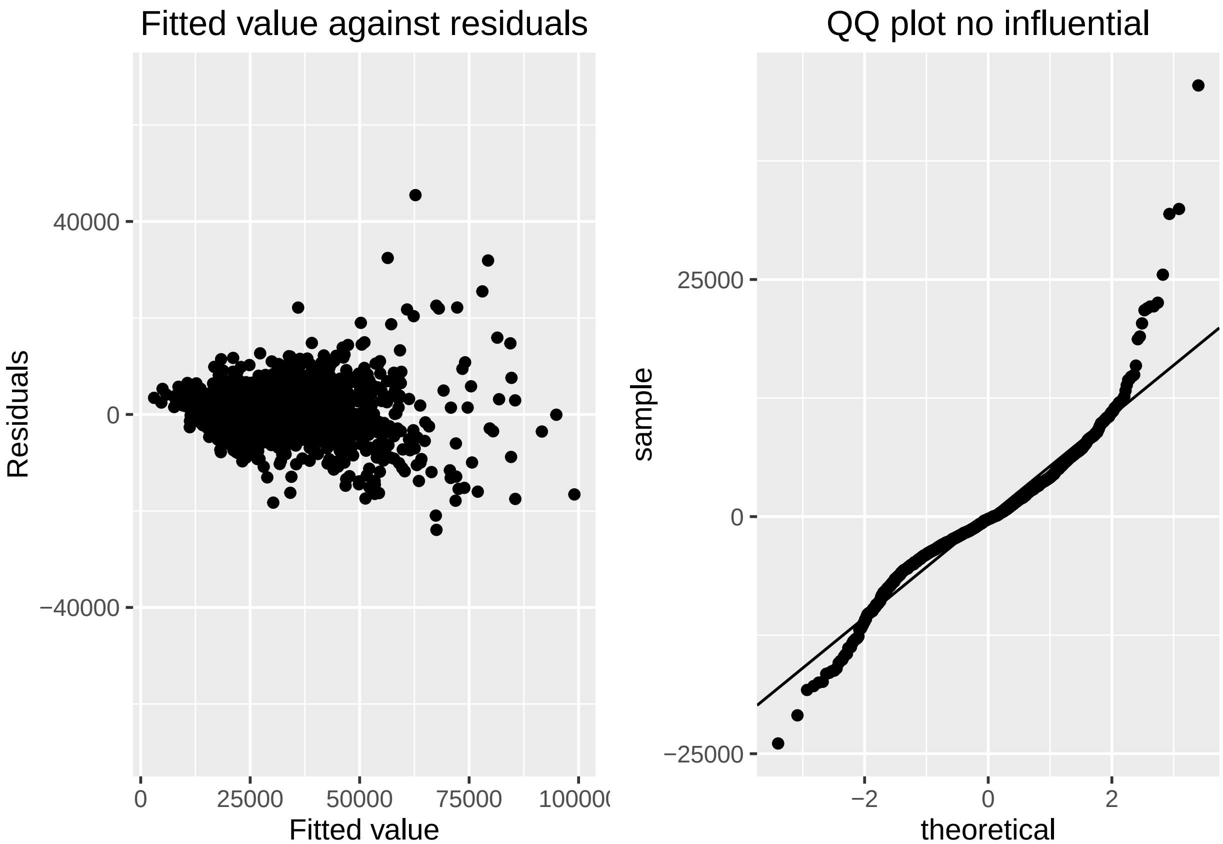
The plots below show that the model in instatnly better for our train data, just by removing some outliers.





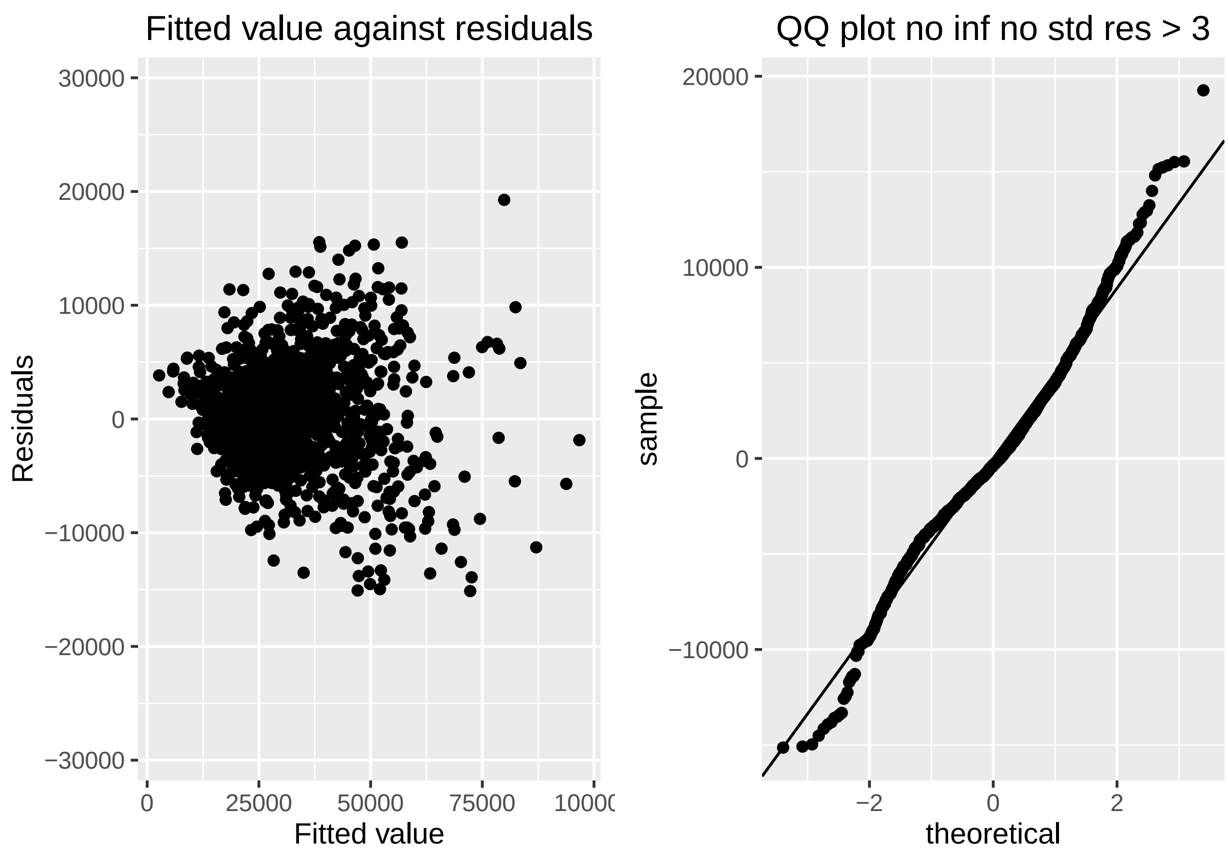
The R-adjusted for fitting the model with new train data is $R{-}adjustedNooutliers$=0.788516 while R-squared for the train data gets better, $R - squared$=0.8826383

With the Cooks distance we can find the most influential points affecting our model. We want to remove all influential points with Cooks distance > 0.0017953 and see how to model fits to that data.



Now our R-adjusted is still worse than for the whole train data, $R-adjusted$= 0.7868115 while R-squared keeps getting higher $R-squared$=0.8664278

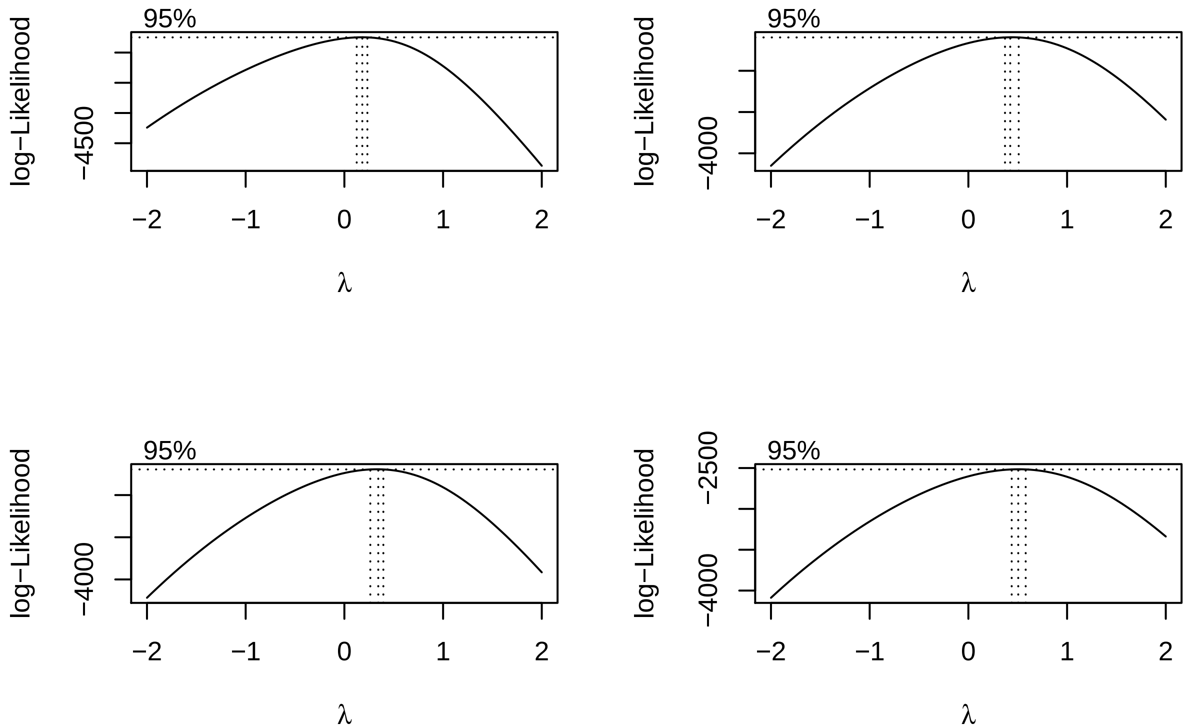
Last data set we make is with no influential points and no outliers. The previous model should fit this data set very well but on the other half R-adjuested might be getting lower.


Fitted value against residuals
Residuals
Fitted value
30000
20000
10000
0
−10000
−20000
−30000
0
25000
50000
75000
10000
QQ plot no inf no std res > 3
sample
theoretical
20000
10000
0
−10000
−2
0
2

```
## [1] 0.7809953
```

Transformation

We know that nuvirdi has an unusually heavy tale so we'll start by transforming our response variable using boxcox.



Mynd 3: Boxcox plot for the four models. Top right: Model with all the training data, top left: Model with no outliers, bottom right: Model with no influential points and bottom left: Model with no outliers and no influential points.

```
Radj.ALLBC <- BCTranformResponseRadj(lm.all, train, test)
Radj.NOBC <- BCTranformResponseRadj(lm.allNoOutlier, trainNO, test)
Radj.NIBC <- BCTranformResponseRadj(lm.allNoInfluential, trainNoInflu, test)
Radj.NONIBC <- BCTranformResponseRadj(lm.allNoInflueNoOutlier, trainNONI, test)
```

Here below we can see the R_{adj} for the four models after transforming the response variable. R_{adj} is calculated using the test set.

	No changes	No outl.	No infl.	No outl. and no infl.
R_{adj}	0.806299	0.8079886	0.804884	0.8011172

From the ggpairs image we can see that $ibm2$ has a heavy right tail as well so lets try log-transforming that variable to see if we get better results.

```
Radj.AllBCAndIBM2 <- TransformBCandIBM2(lm.all, train, test)
Radj.NOBCAndIBM2 <- TransformBCandIBM2(lm.allNoOutlier, train, test)
Radj.NIBCAndIBM2 <- TransformBCandIBM2(lm.allNoInfluential, trainNoInflu, test)
Radj.NONIBCAndIBM2 <- TransformBCandIBM2(lm.allNoInflueNoOutlier, trainNONI, test)
```

Here below we can see the R_{adj} for the four models after transforming the response variable and $ibm2$. R_{adj} is calculated using the test set. Now we get much better results for R_{adj}.

	No changes	No outl.	No infl.	No outl. and no infl.
R_{adj}	0.8339562	0.824357	0.818478	0.8009862

Variable selection

We saw from the transformation chapter that we got the best models by transforming both the response variable and $ibm2$. So we'll be using those models for variable selection.

```
# Feching models and datasets
ALL <- GetBCandIBM2ModelAndDt(lm.all,train, test)
NO <- GetBCandIBM2ModelAndDt(lm.allNoOutlier, trainNO, test)
NI <- GetBCandIBM2ModelAndDt(lm.allNoInfluential, trainNoInflu, test)
NONI <- GetBCandIBM2ModelAndDt(lm.allNoInflueNoOutlier, trainNONI, test)
```

Lets try to use BIC and AIC criteria to select our variables.

```
# BIC tests
ALLBIC <- findBestBICModel(lm(nuvirdi ~ 1, data = ALL$train), ALL$model, ALL$train, ALL$test, ALL$lambda
NOBIC <- findBestBICModel(lm(nuvirdi ~ 1, data = NO$train), NO$model, NO$train, NO$test, NO$lambda)
NIBIC <- findBestBICModel(lm(nuvirdi ~ 1, data = NI$train), NI$model, NI$train, NI$test, NI$lambda)
NONIBIC <- findBestBICModel(lm(nuvirdi ~ 1, data = NONI$train), NONI$model, NONI$train, NONI$test, NONI$

# AIC tests
ALLAIC <- findBestAICModel(lm(nuvirdi ~ 1, data = ALL$train), ALL$model, ALL$train, ALL$test, ALL$lambda
NOAIC <- findBestAICModel(lm(nuvirdi ~ 1, data = NO$train), NO$model, NO$train, NO$test, NO$lambda)
NIAIC <- findBestAICModel(lm(nuvirdi ~ 1, data = NI$train), NI$model, NI$train, NI$test, NI$lambda)
NONIAIC <- findBestAICModel(lm(nuvirdi ~ 1, data = NONI$train), NONI$model, NONI$train, NONI$test, NONI$
```

We can see that we get the best R_{adj} when using the AIC crite

	No changes	No outl.	No infl.	No outl. and no infl.
$R_{adj(BIC)}$	0.8383862	0.8173064	0.8248101	0.8060242
$R_{adj(AIC)}$	0.8381851	0.8173988	0.8243922	0.8047942

Lets now try something different. Lets use the transformed data without any changes and use the add1 function to add explanatory variables.

```
add1(lm(nuvirdi~1, data = ALL$train),~ ibm2 + kdagur + matssvaedi + teg_eign + undirmatssvaedi + haednr
```

```
## Single term additions
##
## Model:
```

```
## nuvirdi ~ 1
##                 Df Sum of Sq     RSS    AIC   F value    Pr(>F)    
## <none>                       17726.4 3685.7                        
## ibm2             1   12176.2  5550.3 1962.2 3255.6006 < 2.2e-16 ***
## kdagur           1    1578.6 16147.8 3549.1  145.0783 < 2.2e-16 ***
## matssvaedi       4    1073.6 16652.8 3600.9   23.8709 < 2.2e-16 ***
## teg_eign         3    7900.8  9825.7 2814.9  397.2228 < 2.2e-16 ***
## undirmatssvaedi 12    3034.2 14692.3 3430.8   25.3496 < 2.2e-16 ***
## haednr           1     328.0 17398.4 3660.0   27.9753 1.413e-07 ***
## fjhaed           1    4744.7 12981.7 3224.8  542.3888 < 2.2e-16 ***
## fjstof           1    6196.5 11529.9 3048.6  797.5382 < 2.2e-16 ***
## byggar           1     179.0 17547.4 3672.7   15.1400 0.0001042 ***
## fjsturt          1    2631.1 15095.3 3449.0  258.6577 < 2.2e-16 ***
## stig10           1       5.2 17721.2 3687.3    0.4351 0.5096102    
## fjbilast         1      38.7 17687.7 3684.5    3.2465 0.0717807 .  
## fjbkar           1    1301.6 16424.8 3574.4  117.5993 < 2.2e-16 ***
## ibteg            1       5.0 17721.4 3687.3    0.4167 0.5187007    
## k.ar             1    1641.4 16085.0 3543.4  151.4317 < 2.2e-16 ***
## lyfta            1     136.5 17589.9 3676.3   11.5165 0.0007080 ***
## fjklos           1    7015.5 10710.9 2939.1  972.0101 < 2.2e-16 ***
## fjeld            1     198.7 17527.7 3671.0   16.8259 4.319e-05 ***
## fjherb           1    8465.8  9260.6 2722.9 1356.6303 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Lets start by adding ibm2.

```r
add1(lm(nuvirdi~ibm2, data = ALL$train),~ ibm2 + kdagur + matssvaedi + teg_eign + undirmatssvaedi + haed
```

```
## Single term additions
## 
## Model:
## nuvirdi ~ ibm2
##                 Df Sum of Sq    RSS    AIC  F value    Pr(>F)    
## <none>                       5550.3 1962.2                       
## kdagur           1   1334.33 4215.9 1555.6 469.3678 < 2.2e-16 ***
## matssvaedi       4   1226.70 4323.6 1599.0 104.9781 < 2.2e-16 ***
## teg_eign         3    538.97 5011.3 1816.4  53.0944 < 2.2e-16 ***
## undirmatssvaedi 12   1016.56 4533.7 1685.6  27.5047 < 2.2e-16 ***
## haednr           1      6.84 5543.4 1962.3   1.8300  0.176333    
## fjhaed           1     19.79 5530.5 1958.9   5.3067  0.021381 *  
## fjstof           1    104.69 5445.6 1935.9  28.5090 1.078e-07 ***
## byggar           1    274.49 5275.8 1888.8  77.1576 < 2.2e-16 ***
## fjsturt          1    151.69 5398.6 1923.0  41.6694 1.461e-10 ***
## stig10           1      2.40 5547.9 1963.5   0.6405  0.423650    
## fjbilast         1     77.16 5473.1 1943.4  20.9084 5.218e-06 ***
## fjbkar           1     32.62 5517.6 1955.4   8.7673  0.003116 ** 
## ibteg            1      2.22 5548.0 1963.6   0.5925  0.441564    
## k.ar             1   1326.95 4223.3 1558.2 465.9552 < 2.2e-16 ***
## lyfta            1      4.94 5545.3 1962.9   1.3211  0.250575    
## fjklos           1     63.84 5486.4 1947.0  17.2553 3.454e-05 ***
## fjeld            1      3.50 5546.8 1963.2   0.9369  0.333226    
## fjherb           1      0.26 5550.0 1964.1   0.0682  0.794046    
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Lets now add matssvaedi. First lets see what model to use.

```r
drop1(lm(nuvirdi~ibm2*matssvaedi, data = ALL$train), test = "F")
```

```
## Single term deletions
##
## Model:
## nuvirdi ~ ibm2 * matssvaedi
##                 Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>                       4164.7 1551.4
## ibm2:matssvaedi  4    158.82 4323.6 1599.0  14.072 2.842e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see from the drop1 function that the best model seems to have different slope and different intercept when just using ibm2 and matssvaedi. Lets continue adding variables.

```r
lm.temp <- lm(nuvirdi~ibm2*matssvaedi, data = ALL$train)
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+byggar+fj
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# Adding kdagur
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur, data = ALL$train)
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+byggar+fj
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# Adding teg_eign
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign, data = ALL$train)
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+byggar+fj
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# Adding byggar
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+byggar, data = ALL$train)
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+byggar+fj
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# Adding undirmatssvaedi
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+byggar+undirmatssvaedi, data = ALL$train)
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+byggar+fj
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# Adding haednr
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+byggar+undirmatssvaedi+haednr, data = ALL$train)
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+byggar+fj
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# Adding fjhaed
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+byggar+undirmatssvaedi+haednr+fjhaed, data = ALL$
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+byggar+fj
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# Adding fjbilast
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+byggar+undirmatssvaedi+haednr+fjhaed+fjbilast, da
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+byggar+fj
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# Adding fjstof
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+byggar+undirmatssvaedi+haednr+fjhaed+fjbilast+fjs
```

```
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+byggar+fjs
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# adding lyfta
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+byggar+undirmatssvaedi+haednr+fjhaed+fjbilast+fjst
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+byggar+fjs
Radj.lyfta <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# adding fjsturt
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+byggar+undirmatssvaedi+haednr+fjhaed+fjbilast+fjst
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+byggar+fjs
Radj.fjsturt <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# adding stig10
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+byggar+undirmatssvaedi+haednr+fjhaed+fjbilast+fjst
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+byggar+fjs
Radj.add1Final <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
```

The table below shows R_{adj} for the last three steps when using the add1 function.

	Add lyfta	Add fjsturt	Add stig10
$R_{adj(add1)}$	0.8514617	0.8546783	0.8543543

After using the add1 function until there was no significant explanatory variable left we got $R_{adj} = 0.8543543$. Lets try using drop1 instead with different intercept and slope for matsvaedi.

```
lm.temp <- lm(nuvirdi ~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+
drop1(lm.temp, test = "F")
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# Dropping k.ar
lm.temp <- lm(nuvirdi ~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+
drop1(lm.temp, test = "F")
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# Dropping fjklos
lm.temp <- lm(nuvirdi ~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+
drop1(lm.temp, test = "F")
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# Dropping fjeld
lm.temp <- lm(nuvirdi ~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+
drop1(lm.temp, test = "F")
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# Dropping ibteg
lm.temp <- lm(nuvirdi ~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+
drop1(lm.temp, test = "F")
Radj.drIbteg <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# Dropping fjherb
lm.temp <- lm(nuvirdi ~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+
drop1(lm.temp, test = "F")
Radj.drfjherb <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# Dropping fjbkar
lm.temp <- lm(nuvirdi ~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+
drop1(lm.temp, test = "F")
Radj.drfjbkar<- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
```

Table below shows R_{adj} for the last three steps when using the drop1 function.

Lets now try to use BIC and AIC starting with the model (nuvirdi ~ ibm2*matssvaedi).

	Drop ibteg	Drop fjherb	Drop fjbkar
$R_{adj(drop1)}$	0.8546999	0.854487	0.8543543

```
null <- lm(nuvirdi~ibm2*matssvaedi, data = ALL$train)
full <- lm(nuvirdi~ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+byggai
# BIC tests
ALLBIC <- findBestBICModel(null, full, ALL$train, ALL$test, ALL$lambda)

# AIC tests
ALLAIC <- findBestAICModel(null, full, ALL$train, ALL$test, ALL$lambda)
```

The table below shows the best R_{adj} for each test when we start with different intercepts for matssvaedi.

	add1	drop1	AIC	BIC
R_{adj}	0.8546783	0.8543543	0.854487	0.849291