

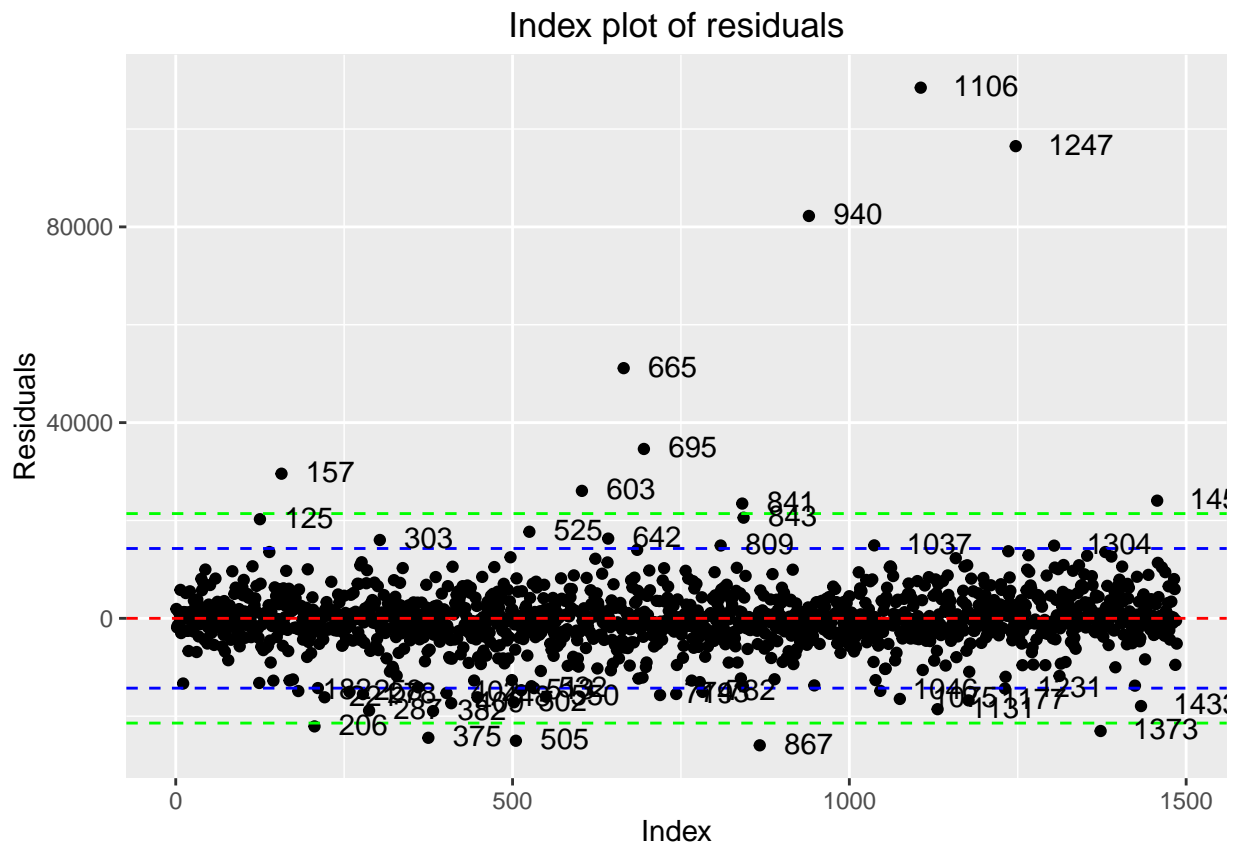
# Frame5

Ottó Hólm Reynisson

15.september 2016

## Residuals

Begin by looking at the residuals from this model

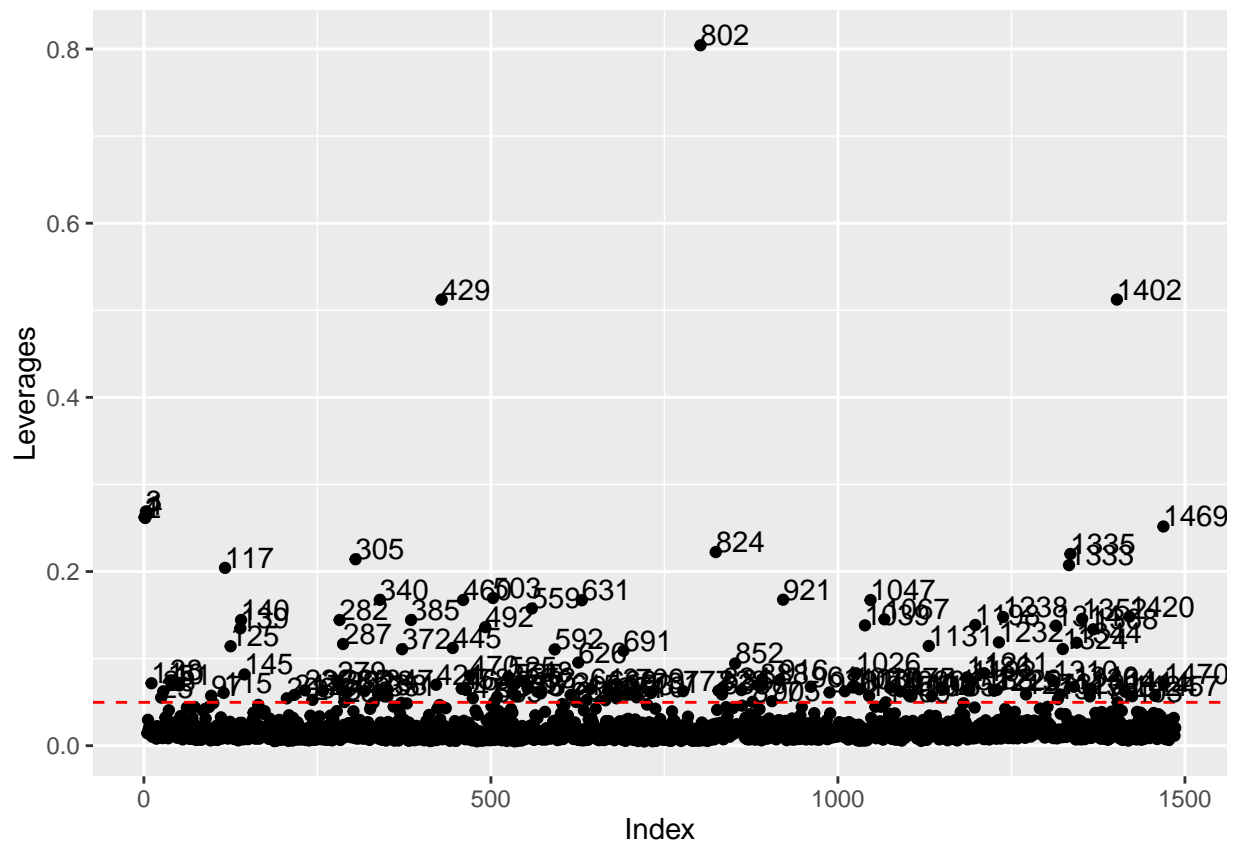


Mynd 1: Indexplot for the residuals.

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.

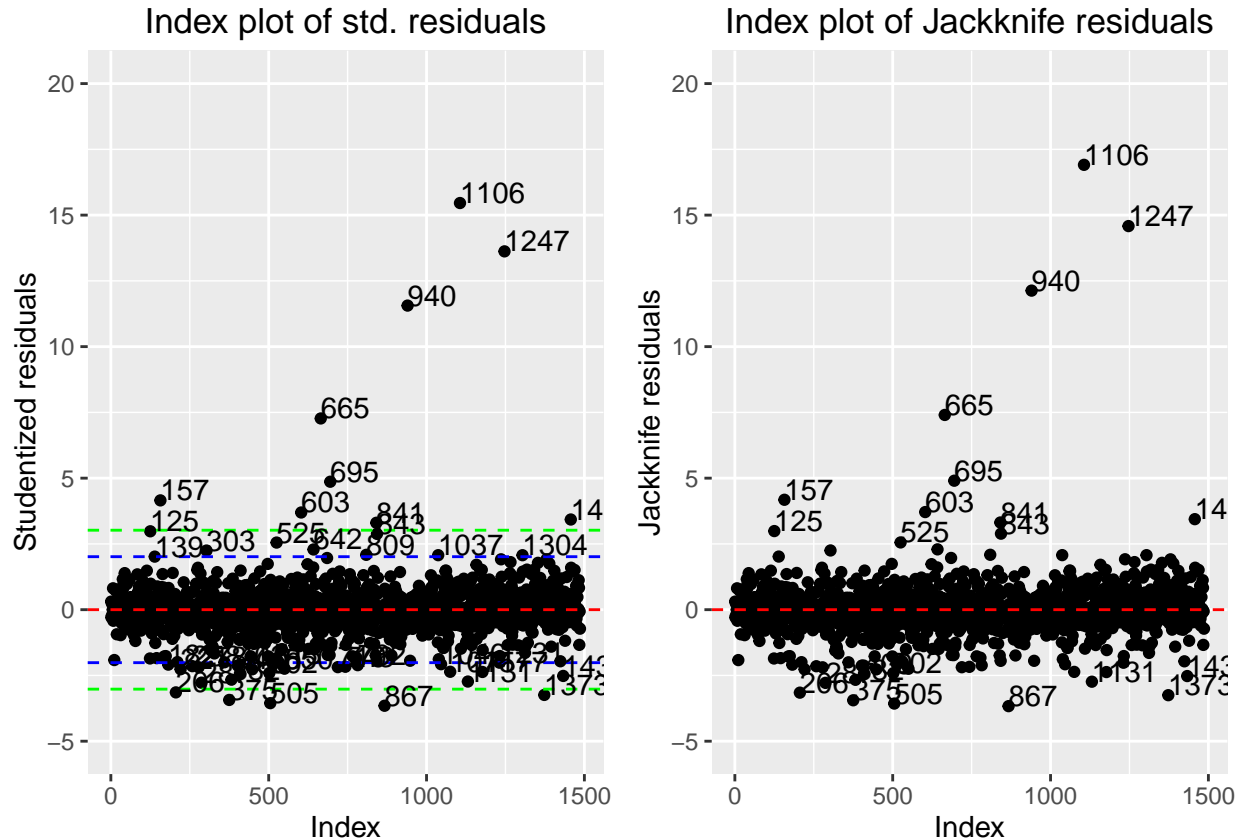
## Leverages

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  $\frac{2p}{n} = \frac{2 \cdot 37}{1486} \approx 0.05$



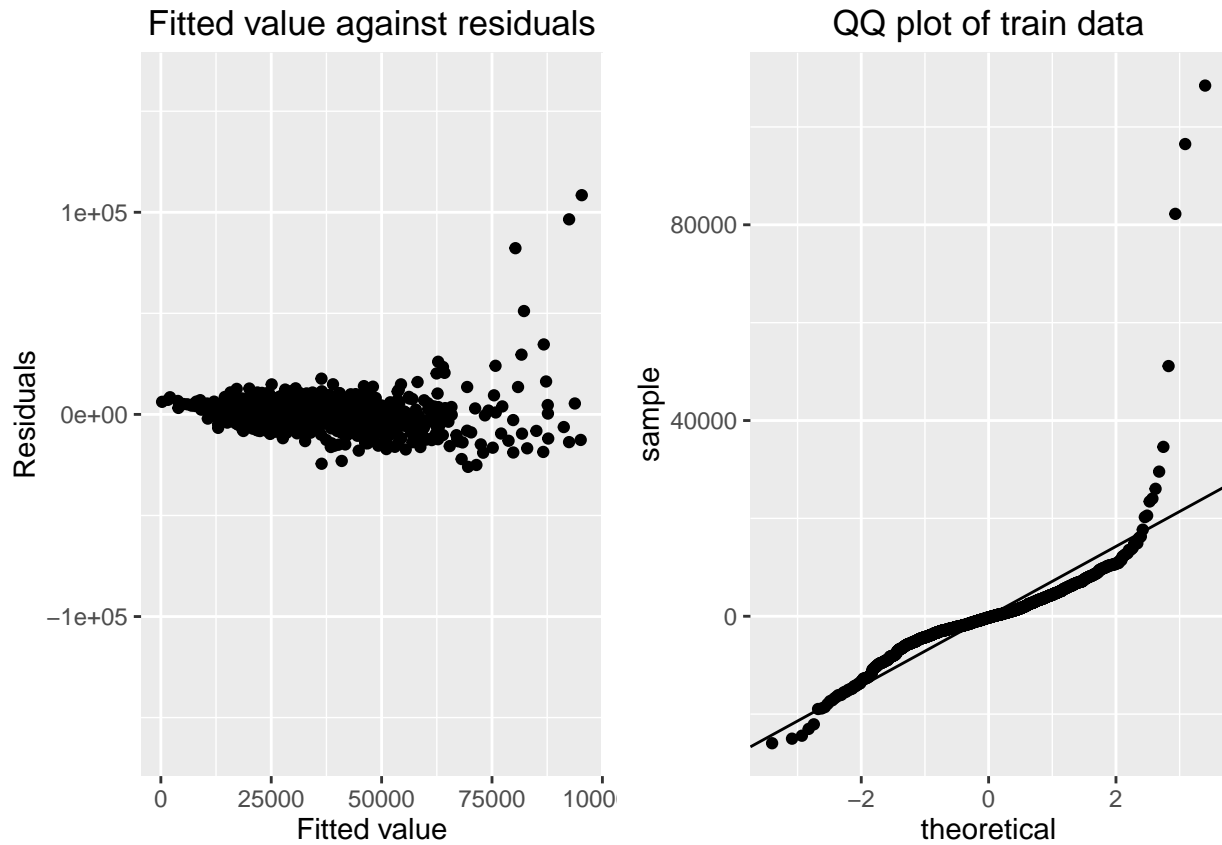
Mynd 2: Indexplot of leverages

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



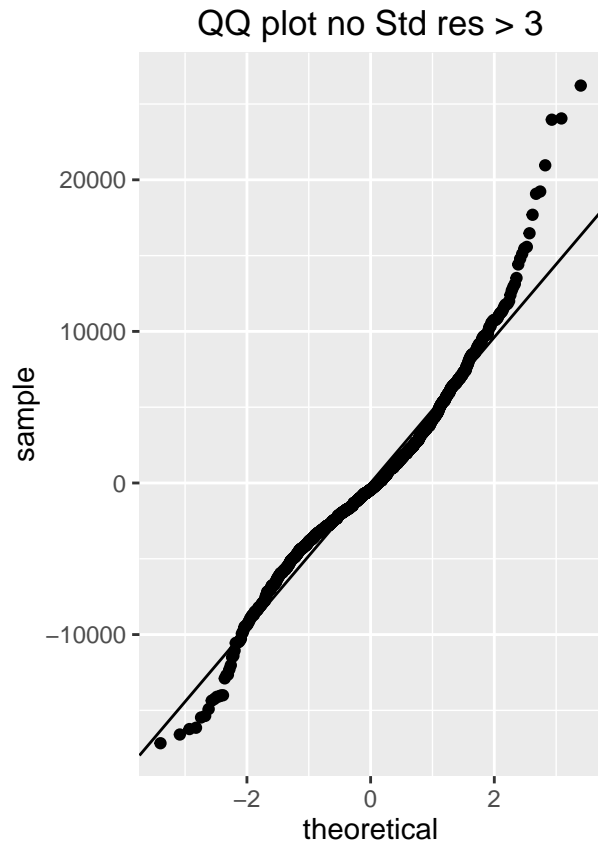
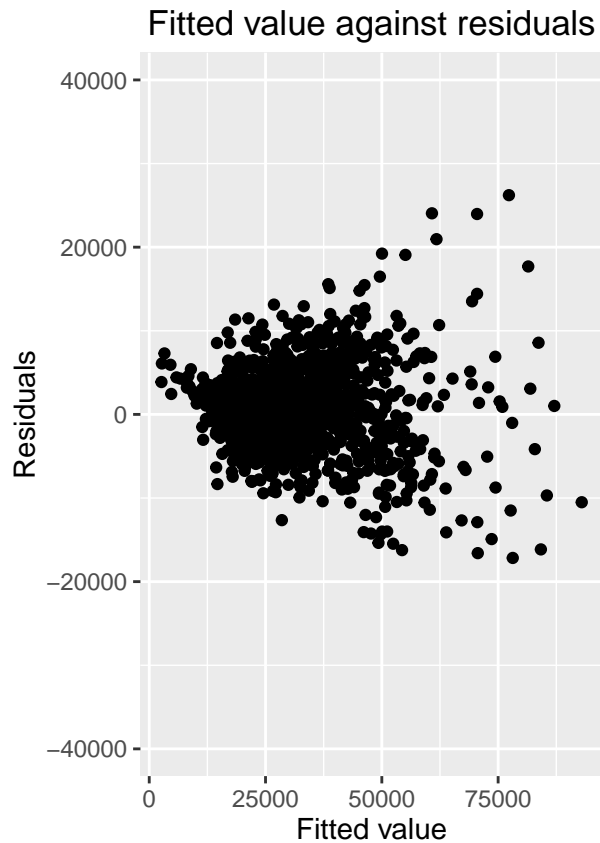
Cook's distance (calculated w.r.t Jackknife and Std.Residuals) is a good way to diagnose influential points in the model. Points with high Cooks distance are affecting the model more than the others. The green points have high Cooks distance, but the blue points have high Cooks distance but also high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





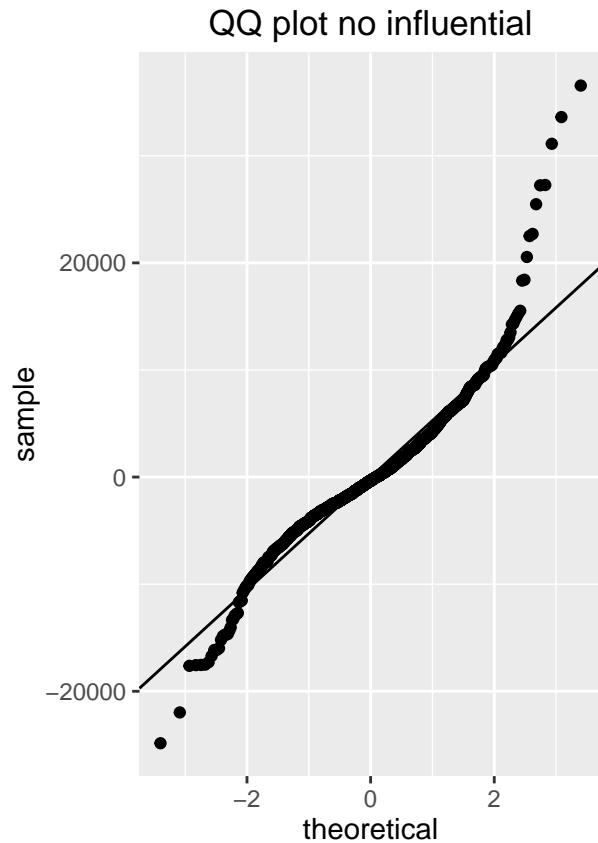
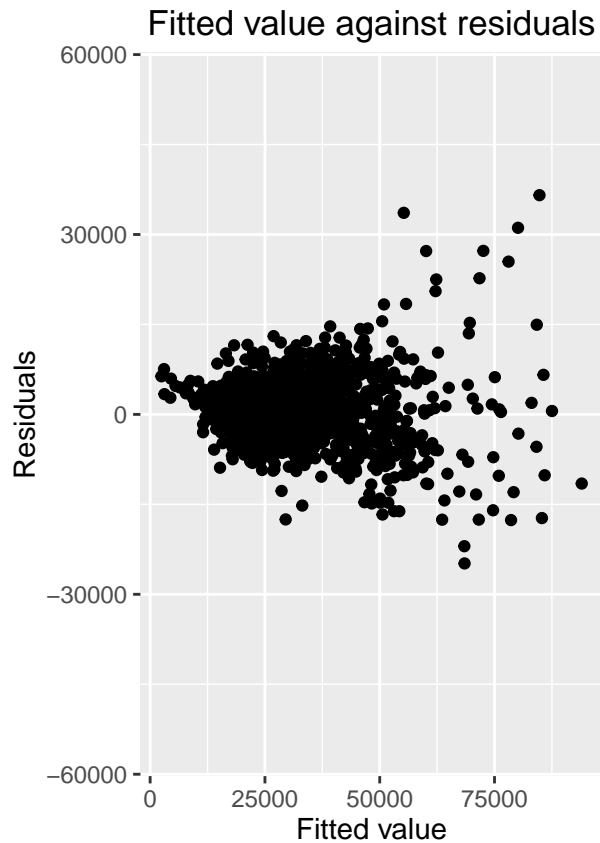
First we calculate the  $R$ -adjusted for the first model and the whole train data and get  $R - adjusted = 0.8520812$ . The  $R$ -squared for this data set and model is,  $R - squared = 0.8018309$ . Now by removing the residuals that have  $\text{std. residuals} > 3$  we have a new model.

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



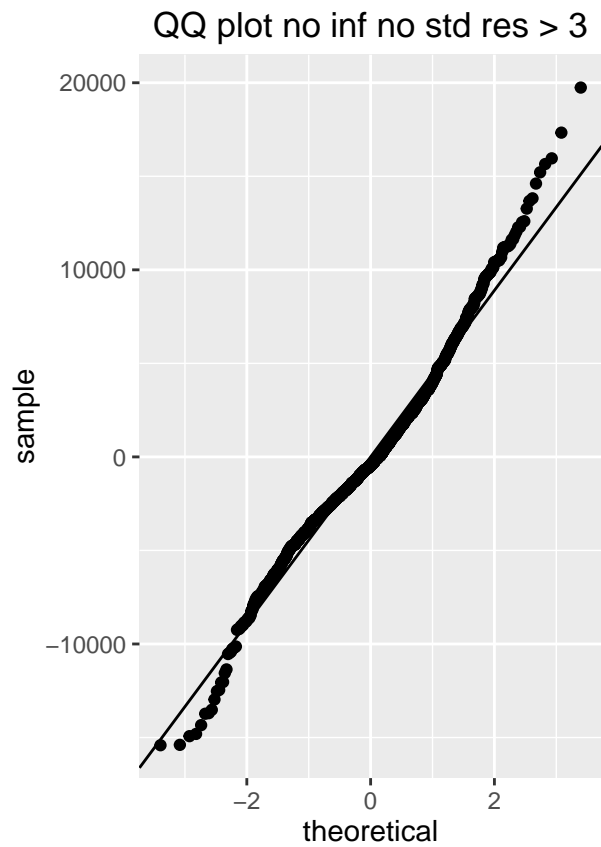
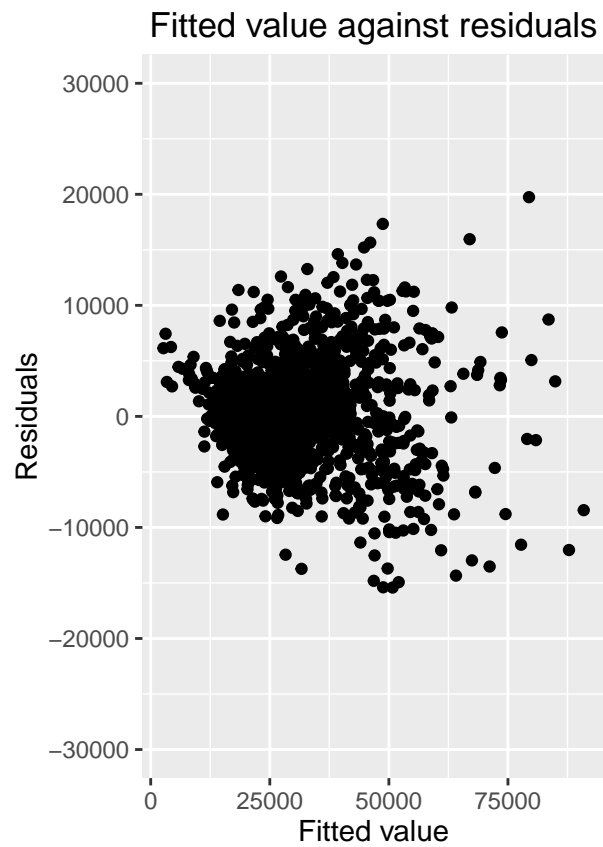
The  $R$ -adjusted for fitting the model with new train data is  $R - adjustedNooutliers=0.8572009$  while  $R$ -squared for the train data gets better,  $R - squared=0.875189$

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,  $R - adjusted = 0.8574566$  while  $R$ -squared keeps getting higher  $R - squared = 0.8624224$

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  $R$ -adjusted might be getting lower.

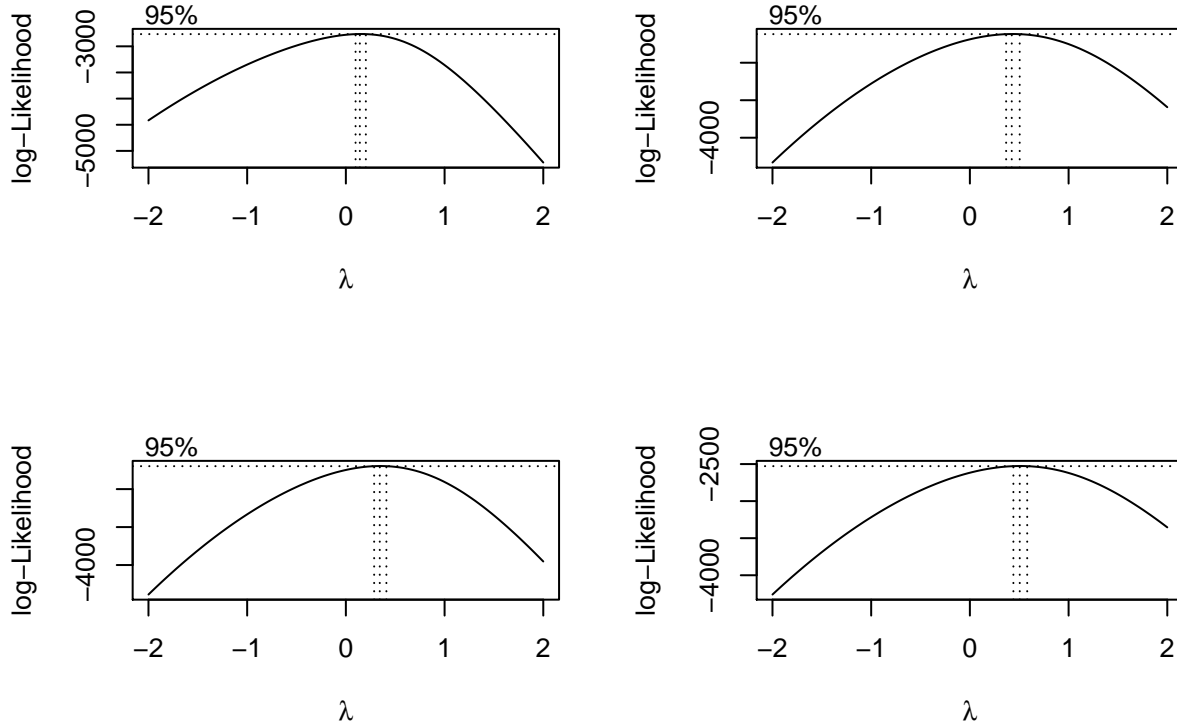


```
## [1] 0.8549536
```



## 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 <- BCTransformResponseRadj(lm.all, train, test)
Radj.NOBC <- BCTransformResponseRadj(lm.allNoOutlier, trainNO, test)
Radj.NIBC <- BCTransformResponseRadj(lm.allNoInfluential, trainNoInflu, test)
Radj.NONIBC <- BCTransformResponseRadj(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.
$R_{adj}$	0.8335258	0.8520399	0.8446827	0.8527298

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.
$R_{adj}$	0.8871675	0.8816331	0.8813979	0.8693338

## 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.

```
# Fetching 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$test, NONI$lambda)

# 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$test, NONI$lambda)
```

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.
$R_{adj}(BIC)$	0.8866053	0.8770973	0.8804325	0.8712037
$R_{adj}(AIC)$	0.8876416	0.8776122	0.8823058	0.8715814

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>                 17327.4 3651.9
## ibm2      1   11912.2   5415.2 1925.6 3264.4626 < 2.2e-16 ***
## kdagur    1    1474.7  15852.7 3521.7  138.0524 < 2.2e-16 ***
## matssvaedi 4    1104.3  16223.2 3562.1   25.2016 < 2.2e-16 ***
## teg_eign   3    7684.7   9642.8 2787.0  393.6864 < 2.2e-16 ***
## undirmatssvaedi 12  2267.9  15059.5 3467.5   18.4857 < 2.2e-16 ***
## haednr     1     282.8  17044.6 3629.5   24.6245 7.769e-07 ***
## fjhaed     1   4621.4  12706.1 3192.9  539.7504 < 2.2e-16 ***
```

```
## fjstof      1      6455.9 10871.6 2961.2  881.2439 < 2.2e-16 ***
## byggar      1       63.3 17264.2 3648.5   5.4374  0.019844 *
## fjsturt     1     2216.3 15111.1 3450.5  217.6580 < 2.2e-16 ***
## stig10      1       6.4 17321.0 3653.4   0.5490  0.458824
## fjbilast    1      58.8 17268.6 3648.9   5.0552  0.024698 *
## fjbkar      1     1501.4 15826.1 3519.2  140.7806 < 2.2e-16 ***
## ibteg       1      17.1 17310.4 3652.5   1.4627  0.226690
## k.ar        1     1443.0 15884.5 3524.7  134.8071 < 2.2e-16 ***
## lyfta       1     157.2 17170.2 3640.4   13.5869  0.000236 ***
## fjklos      1     6851.5 10475.9 2906.1  970.5656 < 2.2e-16 ***
## fjeld       1     521.2 16806.2 3608.5   46.0254 1.681e-11 ***
## fjherb      1     8330.5  8996.9 2680.0 1374.0763 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Lets start by adding ibm2.

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

```
## Single term additions
##
## Model:
## nuvirdi ~ ibm2
##
##          Df Sum of Sq  RSS   AIC F value    Pr(>F)
## <none>                5415.2 1925.6
## kdagur      1   1242.46 4172.7 1540.3 441.5724 < 2.2e-16 ***
## matssvaedi  4   1082.12 4333.1 1602.3  92.4013 < 2.2e-16 ***
## teg_eign    3    587.74 4827.5 1760.9  60.1034 < 2.2e-16 ***
## undirmatssvaedi 12   791.26 4623.9 1714.8  20.9911 < 2.2e-16 ***
## haednr      1     3.59 5411.6 1926.6   0.9834  0.32153
## fjhaed      1     4.87 5410.3 1926.2   1.3342  0.24824
## fjstof      1    115.62 5299.6 1895.5  32.3534 1.546e-08 ***
## byggar      1    310.41 5104.8 1839.9  90.1762 < 2.2e-16 ***
## fjsturt     1    132.33 5282.9 1890.8  37.1486 1.393e-09 ***
## stig10      1     16.41 5398.8 1923.1   4.5087  0.03389 *
## fjbilast    1    123.94 5291.3 1893.2  34.7361 4.666e-09 ***
## fjbkar      1     18.97 5396.2 1922.4   5.2126  0.02256 *
## ibteg       1     2.58 5412.6 1926.9   0.7057  0.40100
## k.ar        1   1200.89 4214.3 1555.0 422.5881 < 2.2e-16 ***
## lyfta       1     1.09 5414.1 1927.3   0.2976  0.58544
## fjklos      1     83.11 5332.1 1904.6  23.1155 1.681e-06 ***
## fjeld       1     9.77 5405.4 1924.9   2.6795  0.10186
## fjherb      1     2.44 5412.8 1926.9   0.6672  0.41417
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
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>                        4151.3 1546.6
## ibm2:matssvaedi  4      181.8 4333.1 1602.3   16.16 5.849e-13 ***
## ---
## 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.

```
lm.temp <- lm(nuvirdi~ibm2*matssvaedi, data = ALL$train)
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+bygggar+fjstuf)
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$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+bygggar+fjstuf)
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# Adding teg_eign
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign, data = ALL$train)
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+bygggar+fjstuf)
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# Adding bygggar
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+bygggar, data = ALL$train)
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+bygggar+fjstuf)
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda)
# Adding undirmatssvaedi
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+bygggar+undirmatssvaedi, data = ALL$train)
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+bygggar+fjstuf)
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda) # 0.8519097
# Adding haednr
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+bygggar+undirmatssvaedi+haednr, data = ALL$train)
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+bygggar+fjstuf)
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda) # 0.8537324
# Adding fjhaed
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+bygggar+undirmatssvaedi+haednr+fjhaed, data = ALL$train)
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+bygggar+fjstuf)
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda) # 0.8548856
# Adding fjbilast
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+bygggar+undirmatssvaedi+haednr+fjhaed+fjbilast, data = ALL$train)
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+bygggar+fjstuf)
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda) # 0.8560273
# Adding fjstof
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+bygggar+undirmatssvaedi+haednr+fjhaed+fjbilast+fjstof)
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+bygggar+fjstuf)
Radj.temp <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda) # 0.8584003
# adding lyfta
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+bygggar+undirmatssvaedi+haednr+fjhaed+fjbilast+fjstof+lyfta)
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+bygggar+fjstuf)
Radj.lyfta <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda) # 0.8584125
# adding fjsturt
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+bygggar+undirmatssvaedi+haednr+fjhaed+fjbilast+fjstof+fjsturt)
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+bygggar+fjstuf)
Radj.fjsturt <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda) # 0.8599631
```

```
# adding stig10
lm.temp <- lm(nuvirdi~ibm2*matssvaedi+kdagur+teg_eign+bygggar+undirmatssvaedi+haednr+fjhaed+fjbilast+fjs
add1(lm.temp,~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+bygggar+fj
Radj.add1Final <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda) # 0.8585105
```

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

	Add lyfta	Add fjsturt	Add stig10
$R_{adj}(add1)$	0.8961315	0.8975622	0.8952104

After using the add1 function until there was no significant explanatory variable left we got  $R_{adj} = 0.8952104$ .  
Lets try using drop1 instead.

```
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) # 0.8578069
# 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) # 0.8573713
# 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) # 0.8575344
# Dropping fjherb
lm.temp <- lm(nuvirdi ~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+
drop1(lm.temp, test = "F")
Radj.drIbherb <- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda) # 0.8580301
# Dropping fjbkar
lm.temp <- lm(nuvirdi ~ ibm2*matssvaedi + kdagur + teg_eign + undirmatssvaedi + haednr + fjhaed+fjstof+
drop1(lm.temp, test = "F")
Radj.drIbkar<- CalculateRadjLambda(lm.temp, ALL$test, ALL$lambda) # 0.8585105
```

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

	Drop ibteg	Drop fjherb	Drop fjbkar
$R_{adj}(drop1)$	0.8933716	0.8951836	0.8952104

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

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

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