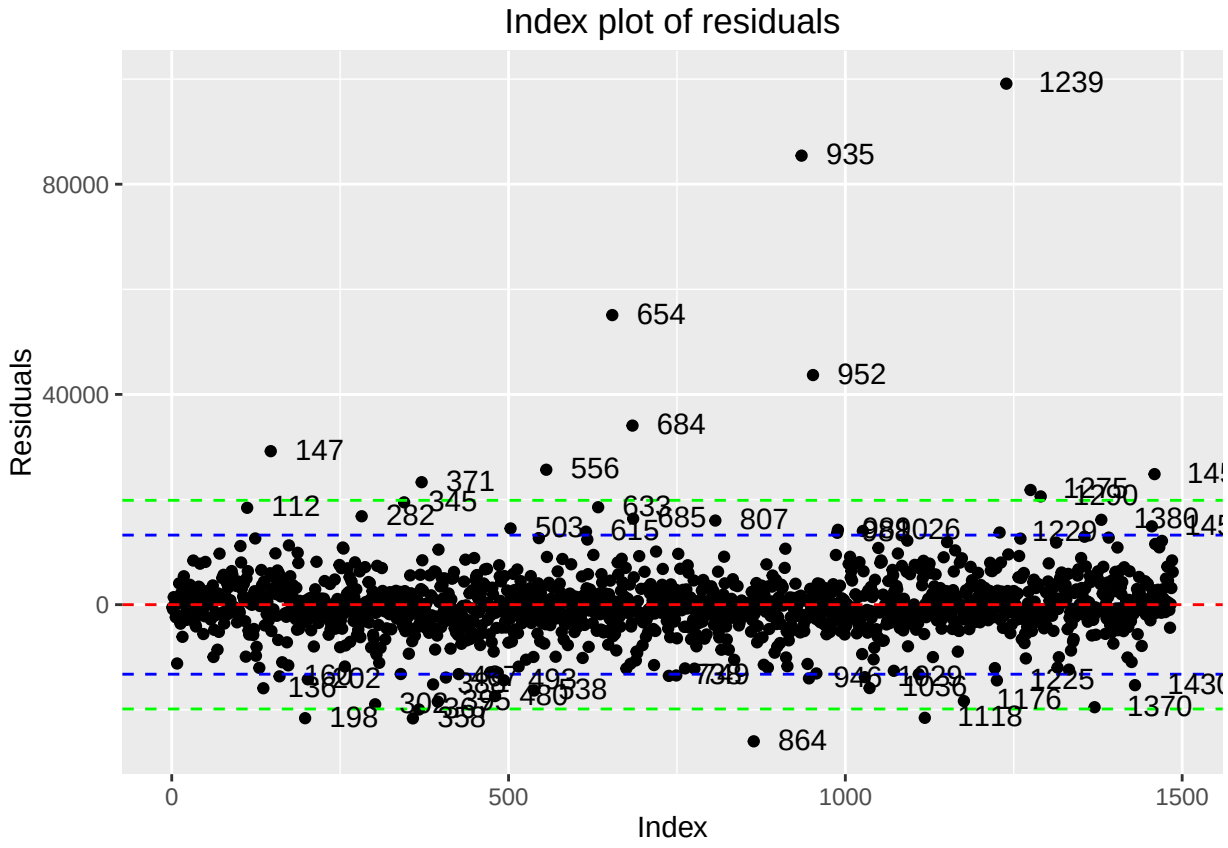
Frame5

Ottó Hólm Reynisson

15.september 2016

Residuals

Begin by looking at the residuals from this model

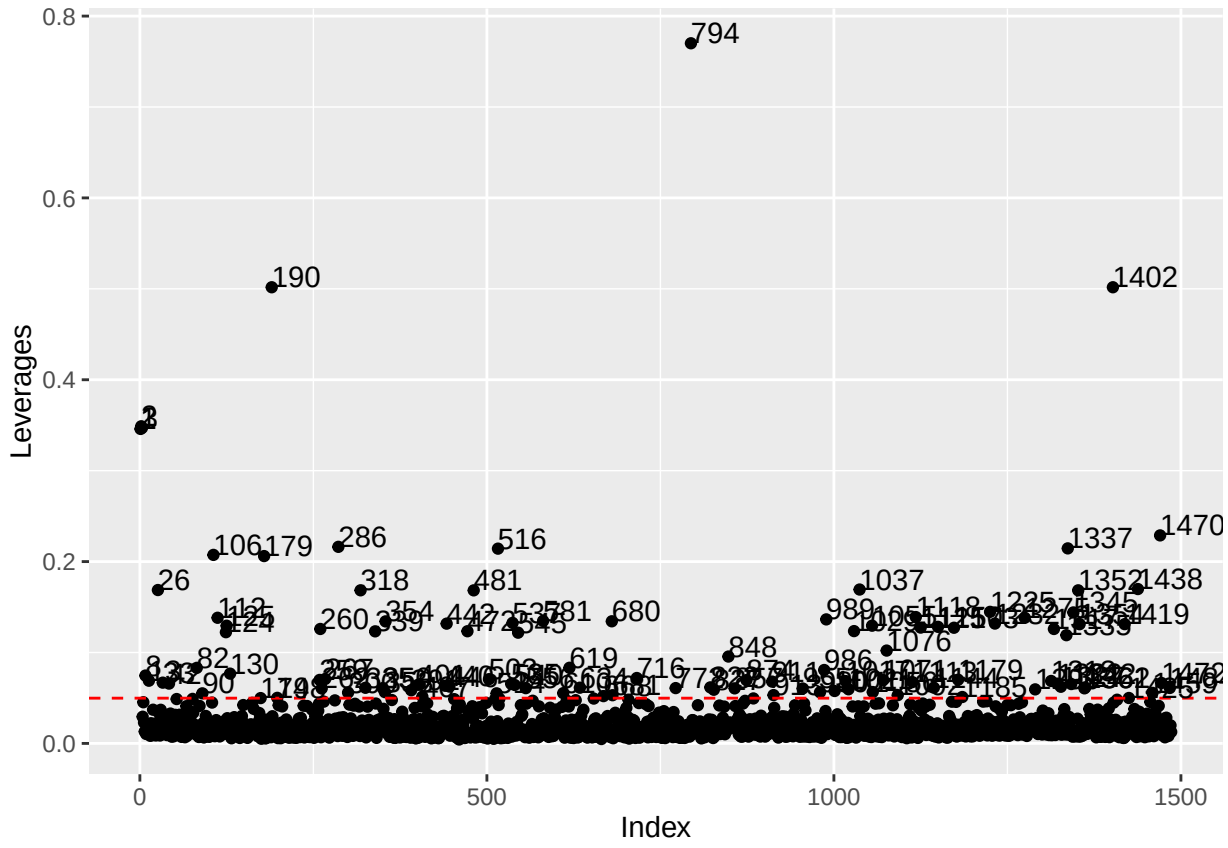


Mynd 1: Indexplot for the residuals.

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

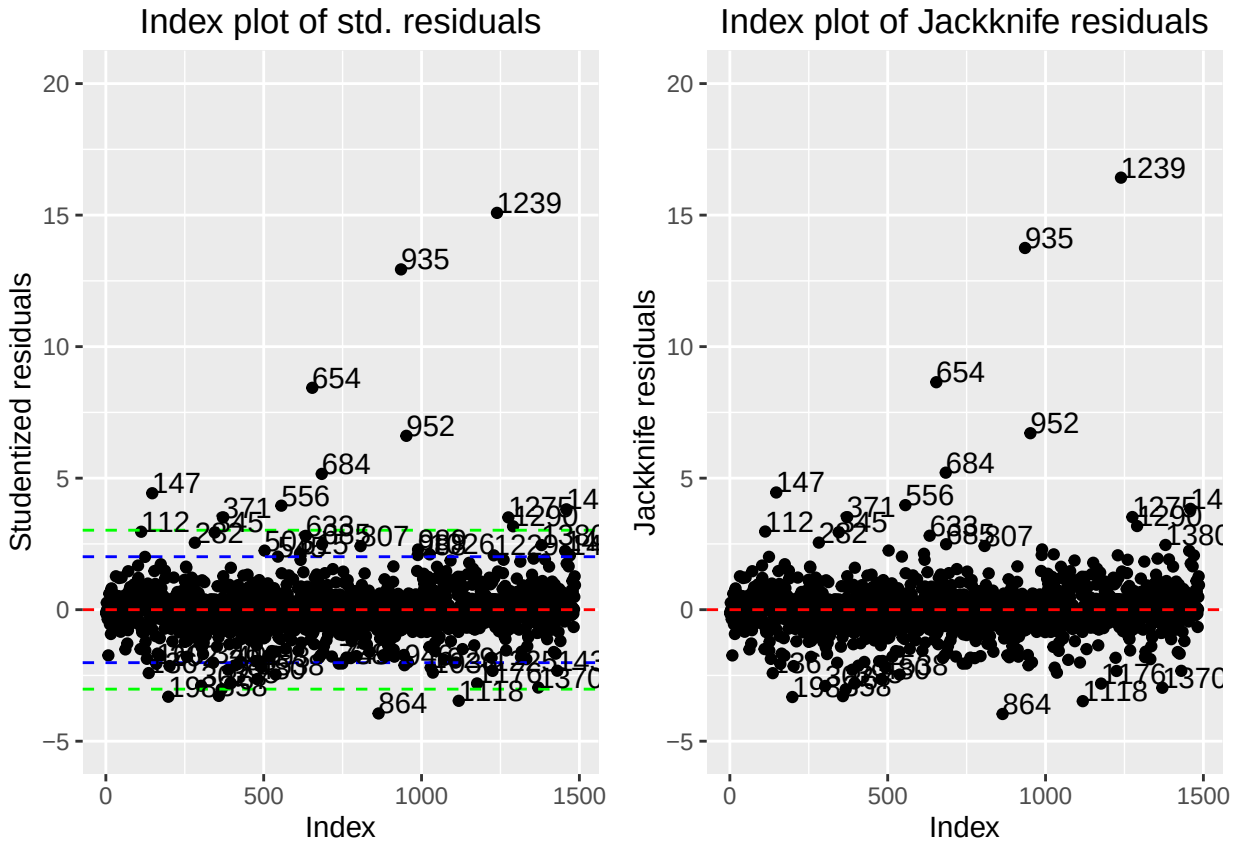


Mynd 2: Indexplot of leverages

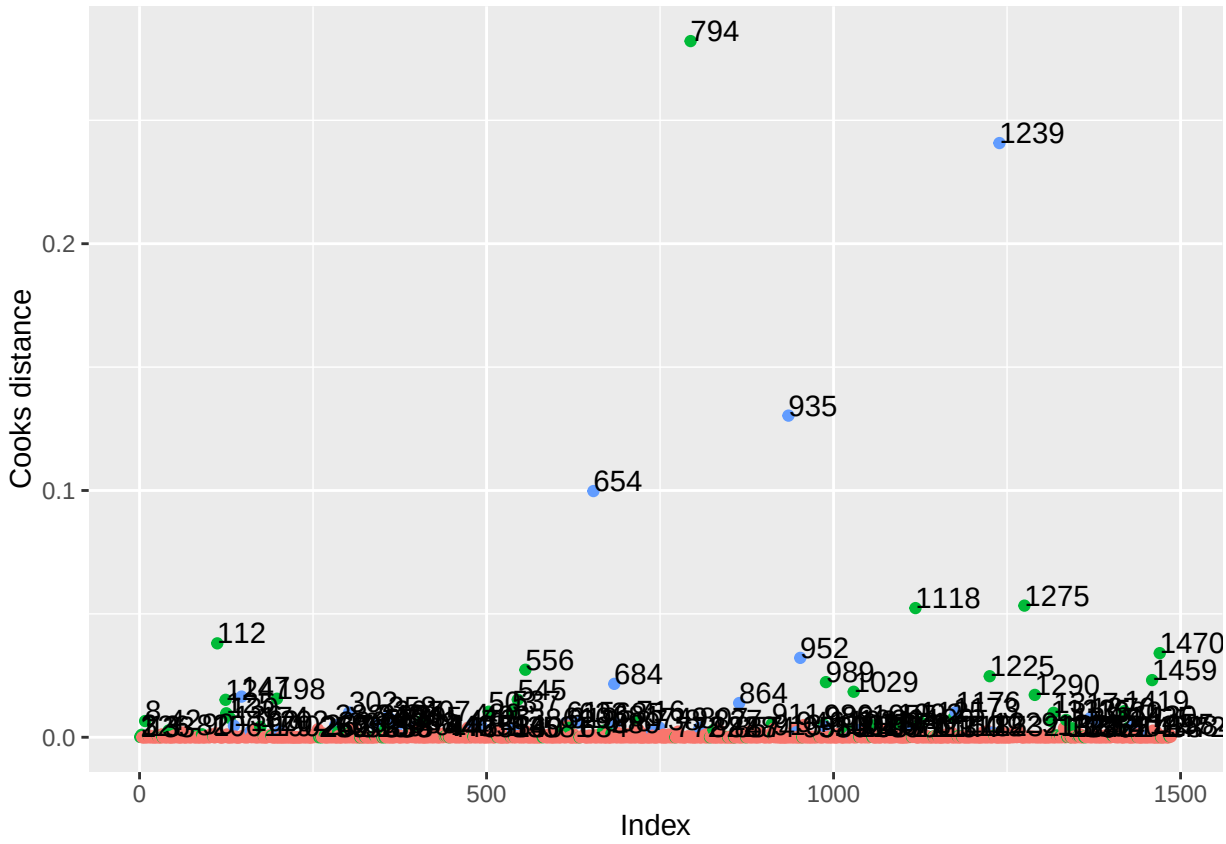
Studentized residuals

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.

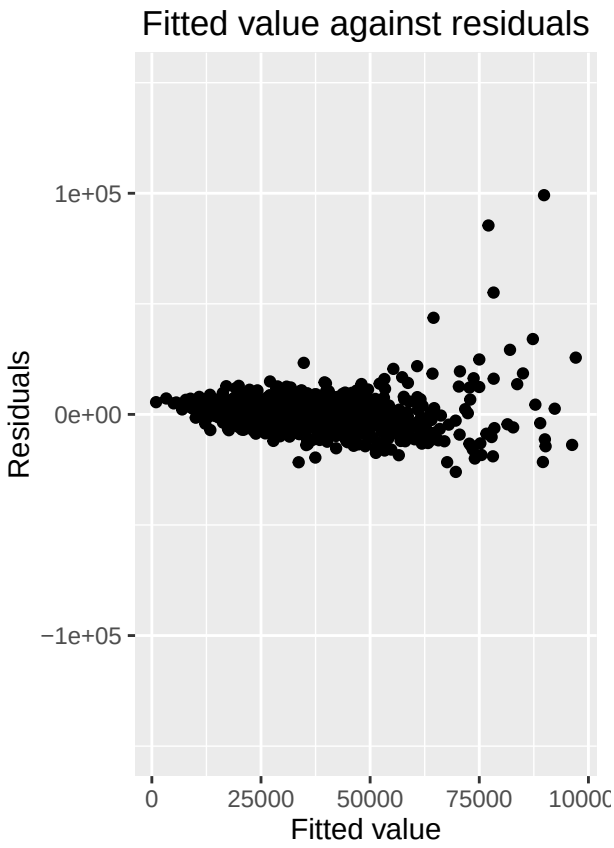


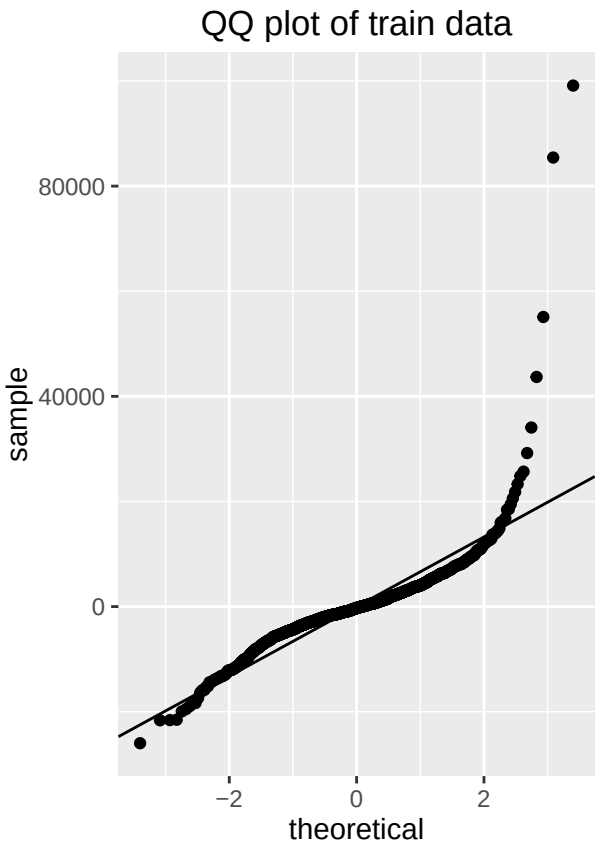
Cook's distance (calculated w.r.t Jackknife and Std.Residuals) is a good way to diagnose influential points in the model. Points with high Cooks distance are affecting the model more than the others. The green points have high Cooks distance, but the blue points have high Cooks distance but also high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

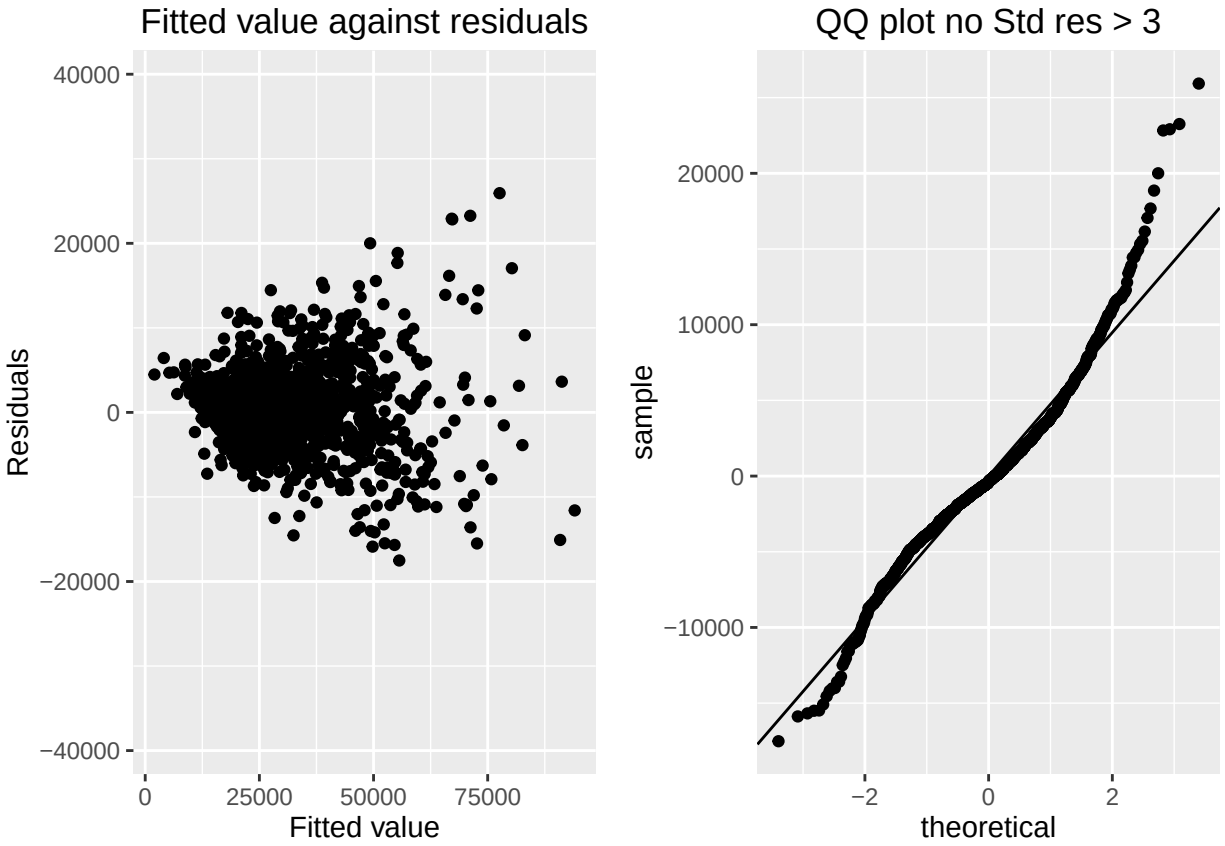
We clearly see this is not what we expected to see. Also the QQ plot This means we have to do some transformation and remove the biggest outliers.





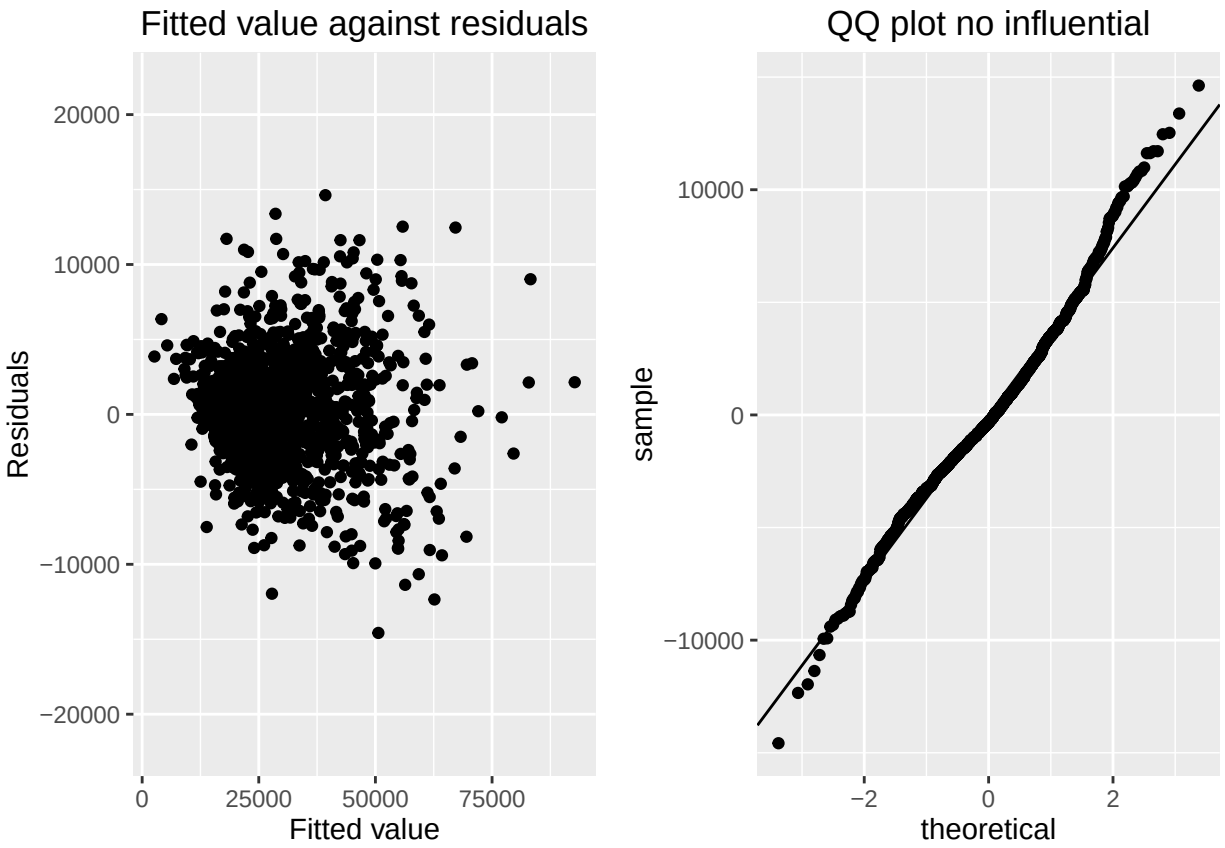
First we calculate the R-adjusted for the first model and the whole train data and get $R - adjusted$= 0.7990392. The R-squared for this data set and model is, $R - squared$=0.8241674 Now by removing the residuals that have std. residuals > 3 we have new model.

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



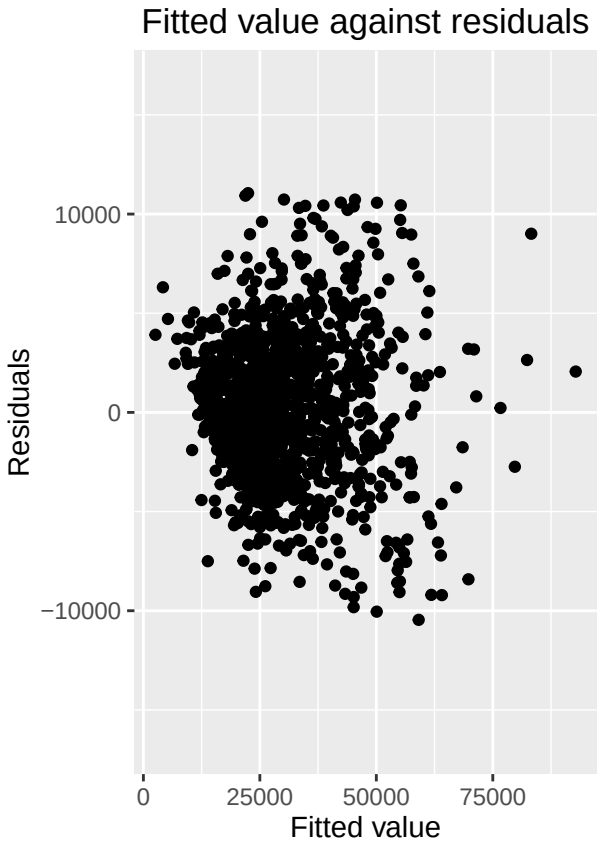
The R-adjusted for fitting the model with new train data is $R - adjustedNooutliers$=0.7905836 while R-squared for the train data gets better, $R - squared$=0.8803153

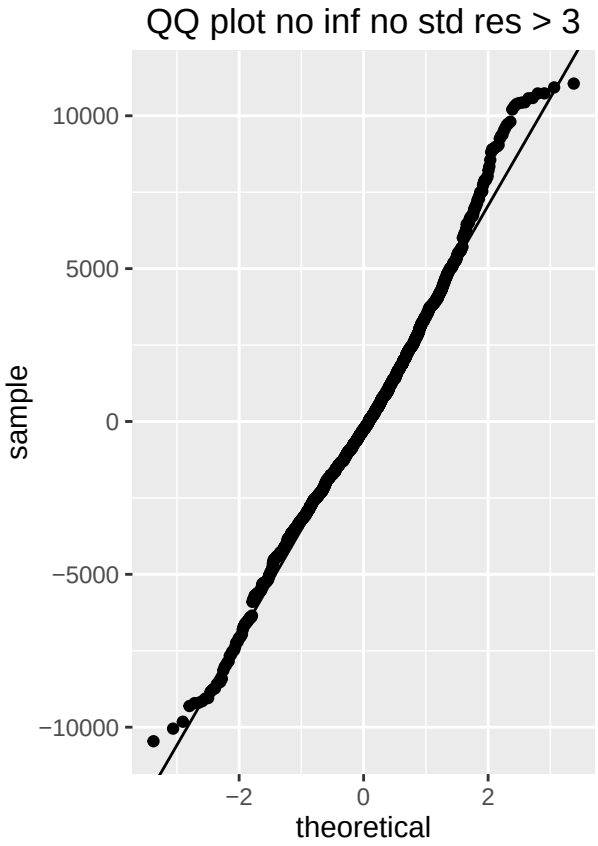
With the Cooks distance we can find the most influential points affecting our model. We want to remove all influential points with Cooks distance > 0.0017953 and see how to model fits to that data.



Now our R-adjusted is still worse than for the whole train data, $R - adjusted$= 0.7919843 while R-squared keeps getting higher $R - squared$=0.8661508

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.

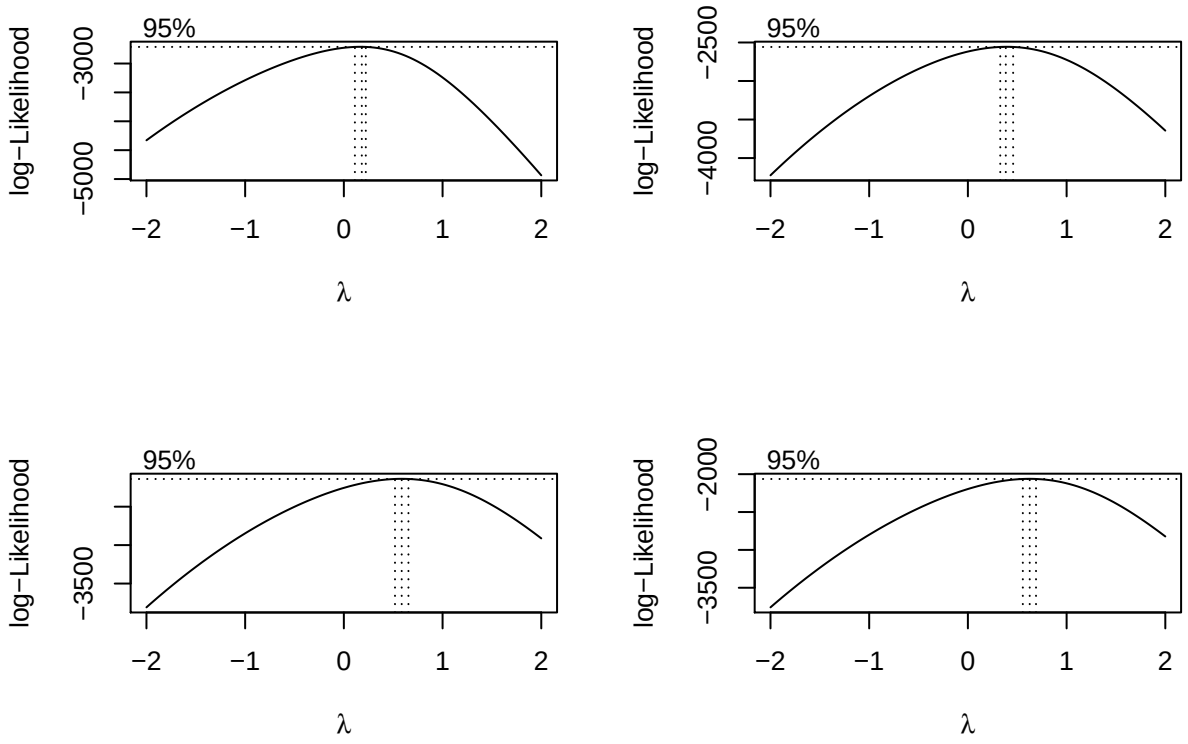

Fitted value against residuals
Residuals
10000
0
−10000
0
25000
50000
75000
Fitted value


QQ plot no inf no std res > 3
sample
10000
5000
0
−5000
−10000
−2
0
2
theoretical

```
## [1] 0.7880962
```

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}	0.7673279	0.7789248	0.7859791	0.7834565

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}	0.8342088	0.831317	0.8121102	0.8112766

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.

```
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
```

	No changes	No outl.	No infl.	No outl. and no infl.
$R_{adj(BIC)}$	0.8350148	0.8194292	0.8180259	0.8161922
$R_{adj(AIC)}$	0.8362594	0.8211832	0.815752	0.8150833