University of Waterloo

STAT 331 Fall 2017

Final Project Report

Group 21:
Ruijie Zhang(20487924)
Yi Qian(20568216)
Teng Liu(20508909)

Instructor:
Leilei Zeng

November 27, 2018

Contents

1 Summary

The main purpose of our project is to analyze the relationship between the price charged for motor transport service in Florida and the other variables (Distance: distance travelled, Weight: weight of product shipped, Pctload: weight as a percentage of truck load capacity, Origin: city of origin, Market: size of market destination, Dereg: deregulation in effect, Product: product classification, Carrier: truck carrier).

We used three automated methods (forward selection, backward elimination, stepwise regression) to fit three linear regression models. We did an error normality check by plotting the residuals and generating QQ-plots. Then, we found skewed errors and we need a transformation of the response. To find the proper transformation, we used Box-Cox transformation procedure. In the end, we found that log transformation is the best.

The three automated methods gives us the same model, and we use it as our first candidate model. To get our second candidate model, we tried replace DISTANCE with DISTANCE_INVERSE(1/DISTANCE), and used log transformation as well to come up with our second candidate model. By performing an in-depth comparison of the two candidate models (different types of residual plots, leverage and influence measures, cross validation, AIC/BIC comparision, agjusted R-Sqaured comaprision), we retained one final model:

lm(formula = log(PRICEPTM) ~ DISTANCE_INVERSE + PCTLOAD + PRODUCT + DEREG + CARRIER.B + ORIGIN + CARRIER.C + W_P_MULTIPLY, data = truck_data)

where DISTNACE_INVERSE = 1/DISTANCE, AND W_P_MULTIPLY = PCTLOAD * WEIGHT. We will discuss the findings of our model based on analysis above.

2 Feature Engineering

Before starting to fit any model, we need to take an overall look of the data. Our data consists of 9 covariates and 1 response variable(PRICEPTM). Out of the 9 covariates, 4 of them are categorical (ORIGIN, MARKET, DEREG, CARRIER), 3 are continous variables with floating points values(DISTANCE, WEIGHT, PCTLOAD), and the remaning 2 are continous variables with integers values(ID, PRODUCT).

Obviously, some feature engineering is needed before we can proceed.

2.1 Eliminate useless variable

By a simple observation, the covariate ID is simply an index of the data, and thus contains no useful information. We will drop covaraite ID.

2.2 Remove NA/inf

It is possible that our dataset constains incomplete fields, and we need to deal with them. First, we check whether our dataset contains NA/inf.

```r
apply(truck_data, 2, function(x) any(is.na(x) | is.infinite(x)))
```

Fortunately, our data does not contain any NA/inf.

2.3 Transform categorical variable

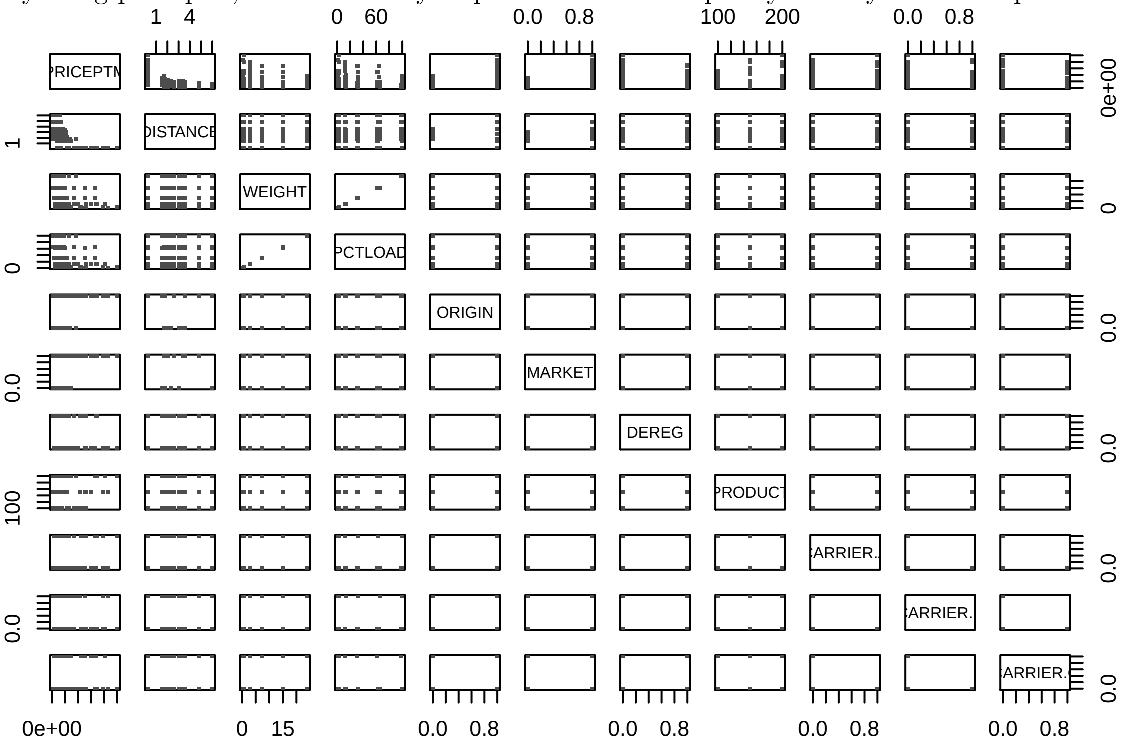
We first check how many levels there are for each categorical variable.

We notice that ORIGIN, MARKET AND DEREG all have binary values. We can use a simple 0/1 encoding to transform them.

For CARRIER, it has 4 different categories. We can use “one-hot” encoding schema. One-hot encoding scheme, simply speaking, is to to create a new 0/1 binary variable for each categorial in the original variable. In our case, we will create 4 new variables, CARRIER.A, CARRIER.B, CARRIER.C, CARRIER.D. And then, we can drop CARRIER.D, because for a linear model, we can choose to encode CARRIER.D as the intercept in the model.

2.4 Visually inspect data

By using pairs plot, we can visually inspect the data and quickly identify relationship between variables.



From the pairs plot above, we have some interesting observations:

- Collinearity: There is a clear linear relationship between two covariates: WEIGHT and PCTLOAD. It makes sense in real life as the the more weight of the shipment, the more pecentage it should take of the truck’s capacity. Since there is collonearity in our data, we need to add an extra covariate W_P_MULTIPLY in our model, which is the product of WEIGHT and PCTLOAD in order to deal with this collinearity.
- Possible Heteroskedasticity: When we look at PRICEPTM vs DISTANCE plot, it looks like there could possibly exist a linear relationship between them, but the variance of PRICEPTM is much larger when DISTANCE is small. And after the distance increased, the variance seems to decrease. We could handle this by doing a variable transformation on our response variable PRICEPTM. Or it is possible that there is no linear relationship between them, but PRICEPTM is proposional to the inverse of DISTANCE i.e (Y ~ beta/X). We will do a detailed analysis in section 3.
- Linear relationships with response variable: There is clear evidence that WEIGHT, PCTLOAD, ORIGIN, MARKET, DEREG and PRODUCT have linear relationship with PRICEPTM. It is not very clear if CARRIER.A, CARRIER.B and CARRIER.C has linear relationship with PRICEPTM.

2.5 Handle corllinearlity

```
truck_data$W_P_MULTIPLY = truck_data$PCTLOAD * truck_data$WEIGHT     # create new covariate W_P_MULTIPL
truck_data_copy <- truck_data                          # reserve a copy for later use
```

3 Model Selection

We are using three automated model selection methods: forward selection, backward elimination and stepwise.

First, we define the minimal and full model

```
full_model = lm(PRICEPTM~., data=truck_data)  # full model that includes all the covriates(after featur
min_model = lm(PRICEPTM~1, data=truck_data) # minimal model that only includes intercept
```

3.1 Automated selection

By using automated selection, we have three fitted models, and they are:

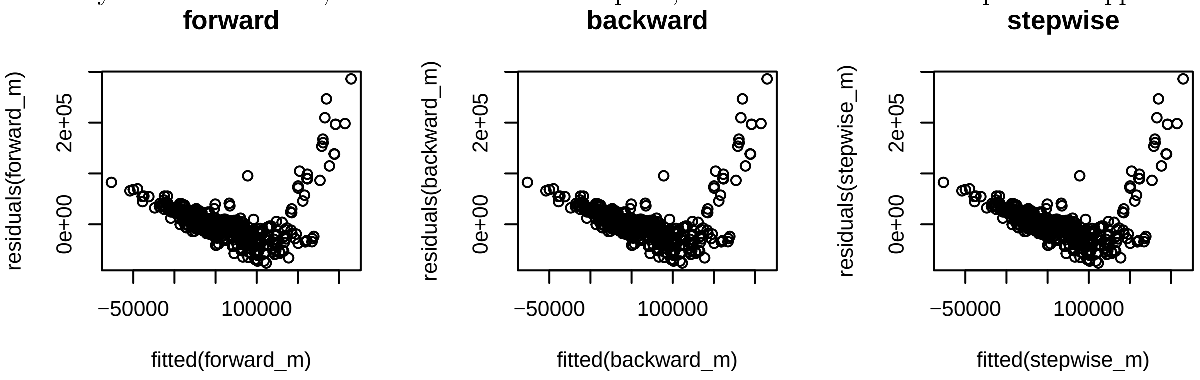
```
## lm(formula = PRICEPTM ~ DISTANCE + PCTLOAD + ORIGIN + PRODUCT +
##     DEREG + MARKET + CARRIER.B + W_P_MULTIPLY, data = truck_data)
```

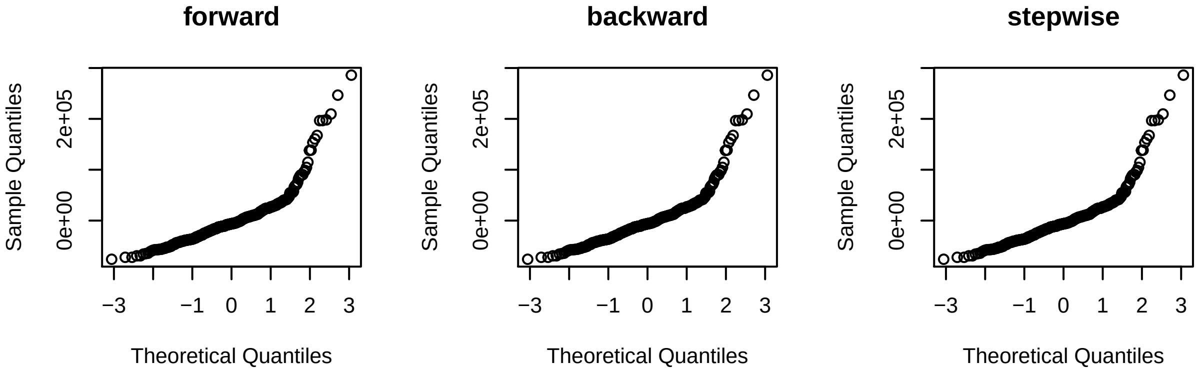
```
## lm(formula = PRICEPTM ~ DISTANCE + PCTLOAD + ORIGIN + MARKET +
##     DEREG + PRODUCT + CARRIER.B + W_P_MULTIPLY, data = truck_data)
```

```
## lm(formula = PRICEPTM ~ DISTANCE + PCTLOAD + ORIGIN + PRODUCT +
##     DEREG + MARKET + CARRIER.B + W_P_MULTIPLY, data = truck_data)
```

3.2 verify fitted model

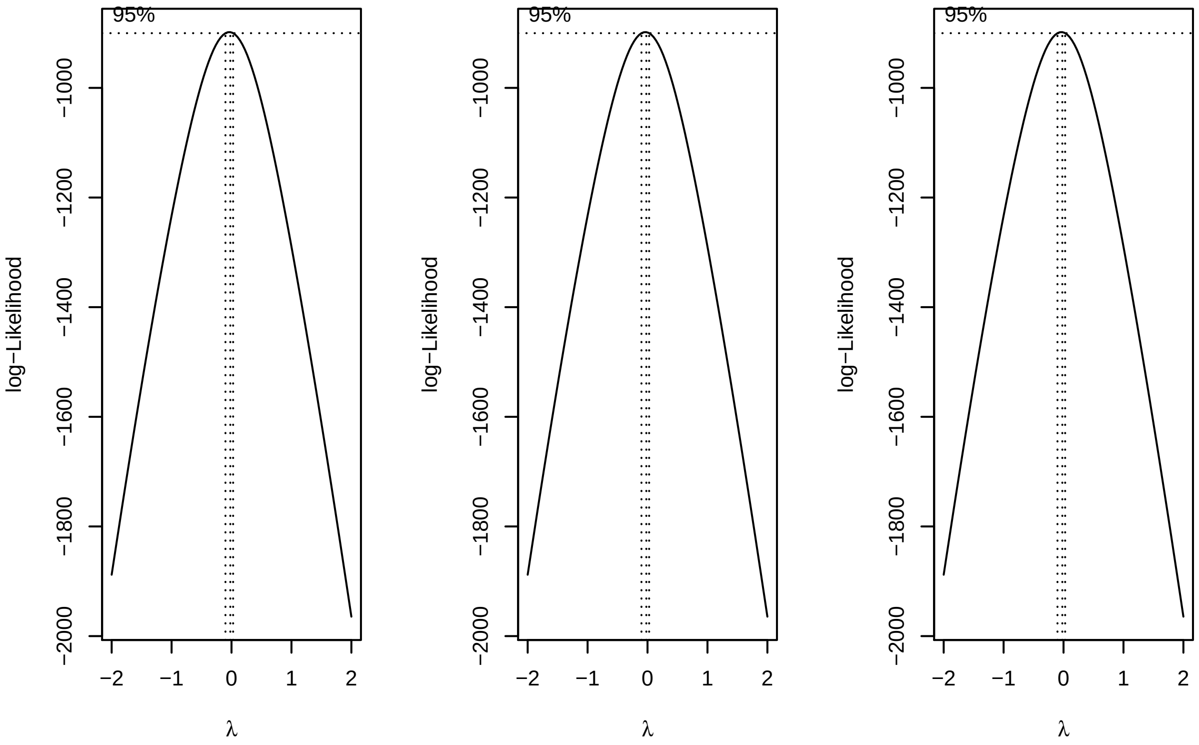
To verify our fitted model, we used two kinds of plots, residual vs fitted value plot and qqnorm plot:





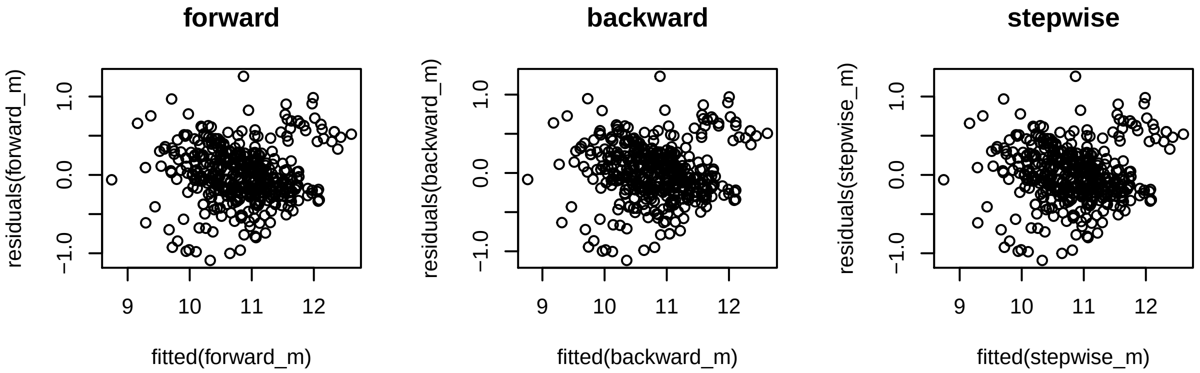
From the Residual vs Fitted plot, we can see the points are not randomly distributed around the 0 line, suggesting that our models are not normal. And the QQ-plot further proved our inspection, we can see that the plot is skewed, suggesting that we should transform our response variable.

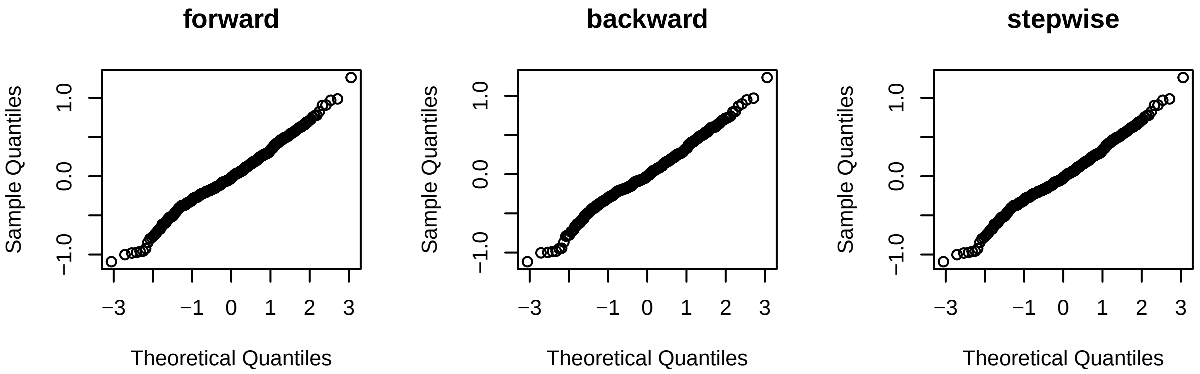
We then used box-cox method to determine what kind of transformation we should use.



From the box-cox plot, we can see that $\lambda = 0$ is the best choice, so we will use log transformation on our response variable.

3.3 Log transformation and fit

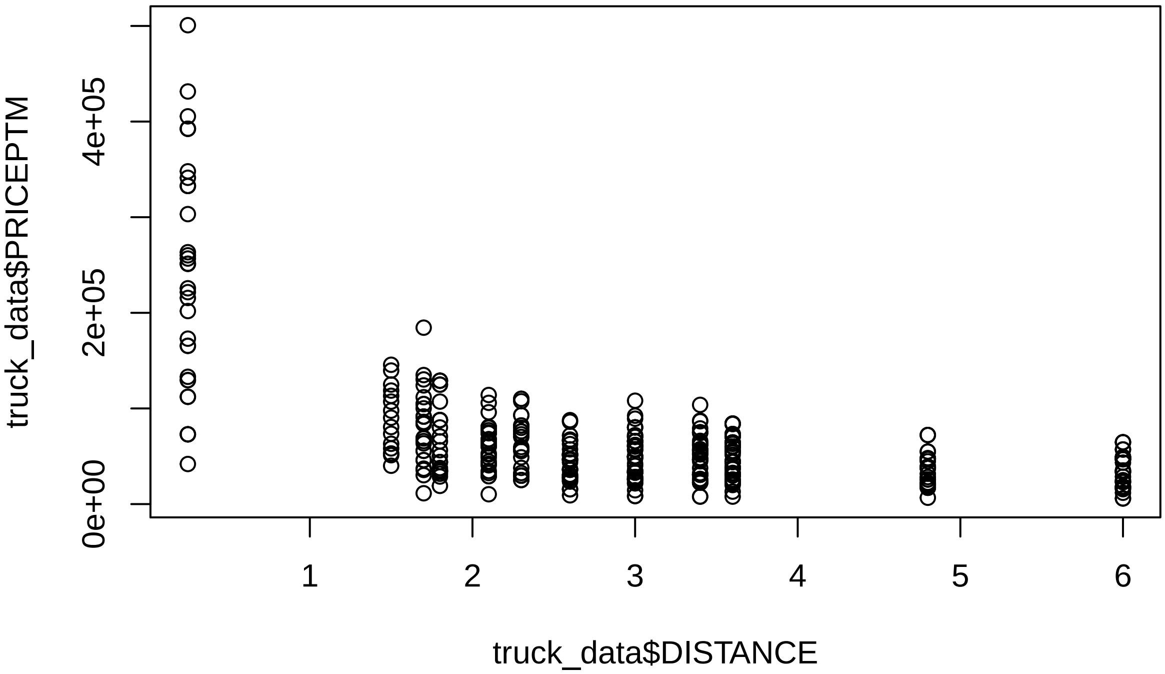




We can clearly see that the transformed data looks more like a normal distribution.

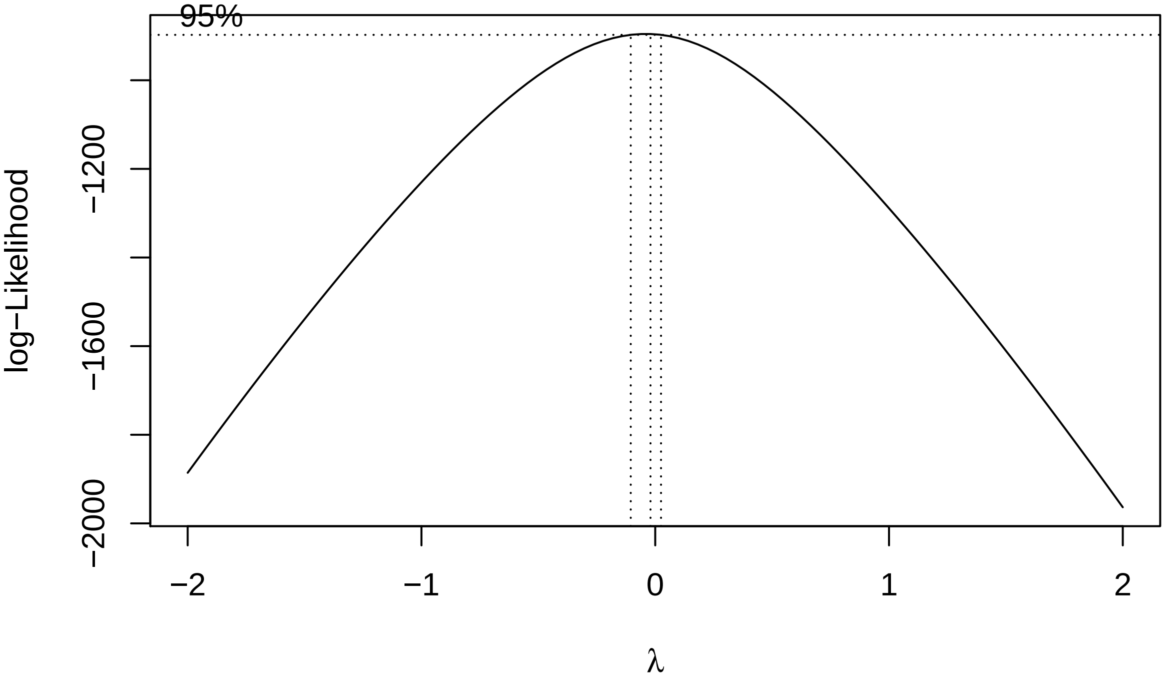
3.4 Choose two candidate models

Our 3 auto model selection methods yield exactly the same model. But we would like to have two candidate models for further analysis. Notice that in section 2.4, when we look at the plot PRICEPTM vs DISTANCE, we assume that there is a linear relationship between PRICEPTM and DISTANCE, and it is a heteroskedasticity issue(changing variance) makes the plot looks odd. But it is also possible that this aussmption is wrong. The true relationship between PRICEPTM and DISTANCE could be PRICEPTM ~ k / DISTANCE, in other words, the PRICEPTM is proportional to the inverse of the DISTANCE, as the plot does look like plot of f(x) =



k/x.

So we will fit another model with a new variate DISTANCE_INVERSE = 1/DISTANCE. Before we fit this model, we plot boxcox again and the plot tells us that we should use a log transformation.



After applying automated model selection, we have another 3 models: Inverse_1, Inverse_2, Inverse_3, and we need to choose another candidate from these three. This is easy since these 3 candidates are exactly the same.

So We will choose two models as our candidates, forward_m and Inverse_1.

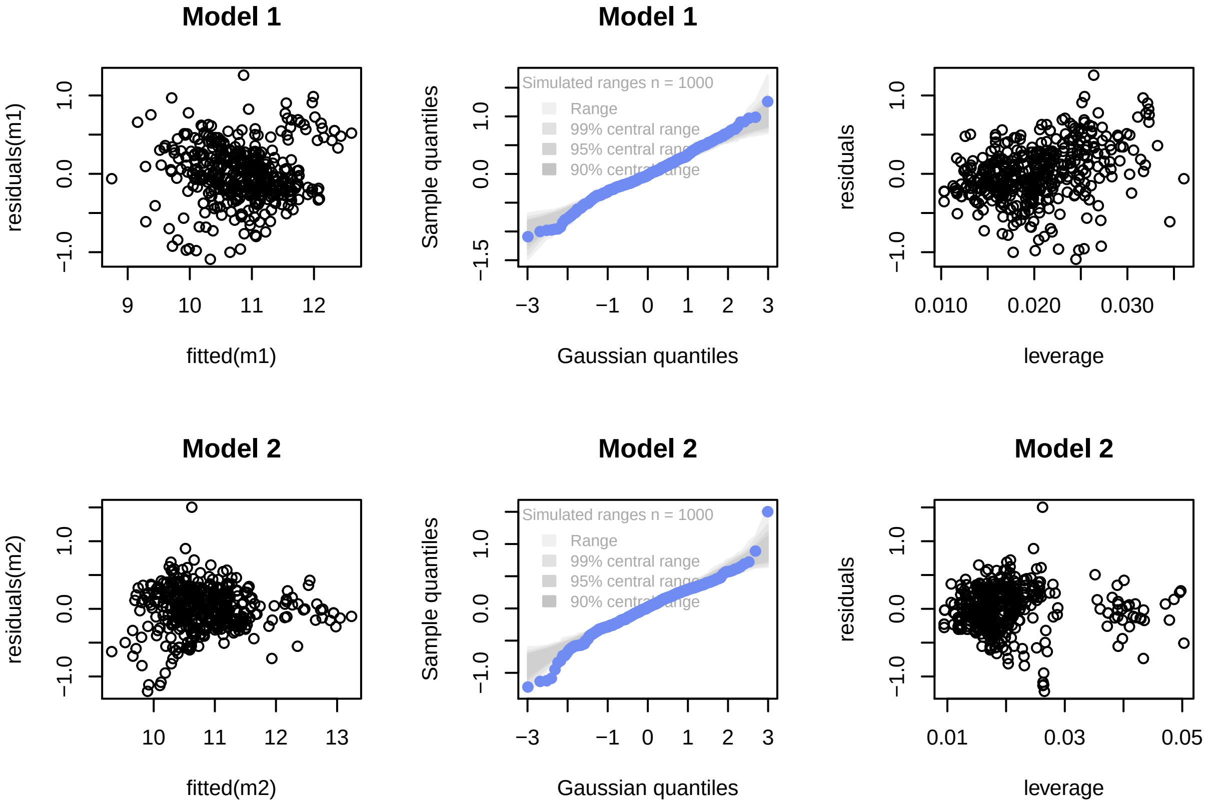
4 Model Diagnostics

In this section, we will perform an in-depth analysis of our two candidate models, forward_m and Inverse_1. For simplicity, we rename forward_m as m1, and Inverse_1 and m2.

4.1 Different types of residual plots

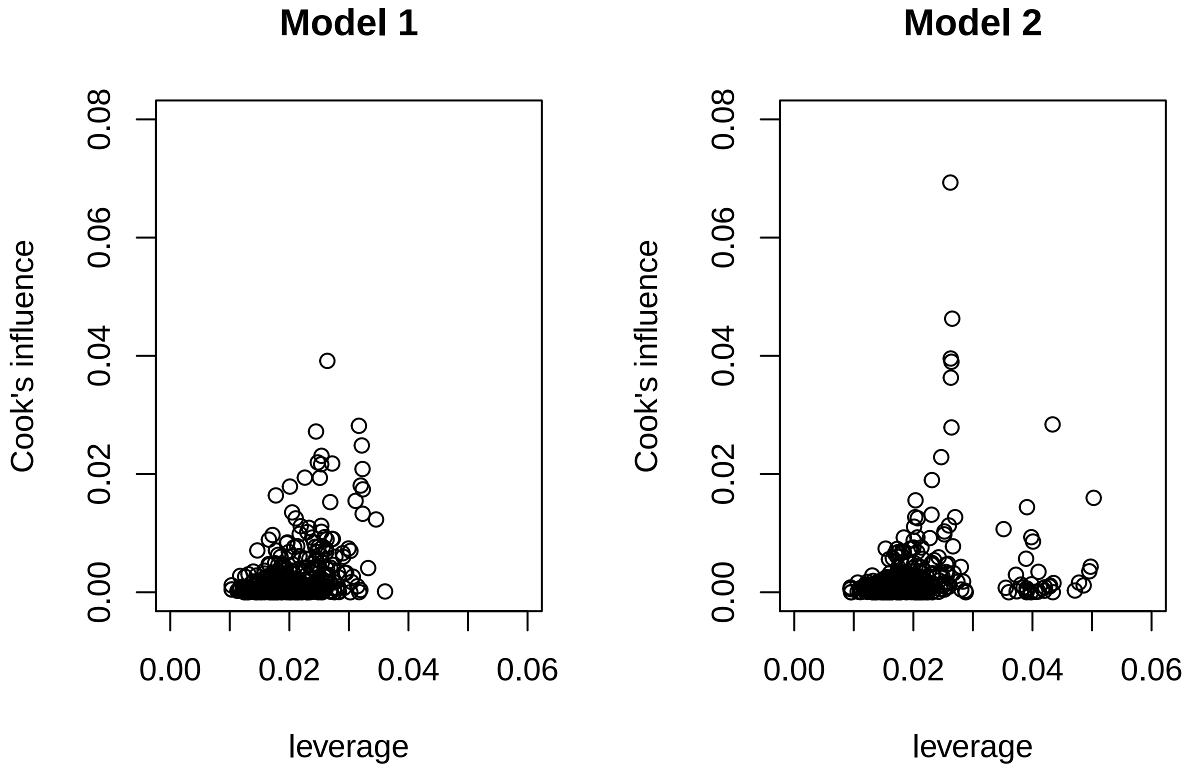
```
##
## Attaching package: 'qqtest'
```

```
## The following object is masked from 'package:MASS':
##
##     bacteria
```



We plotted three different types of residual plots, residuals vs fitted, residual quantiles vs Gaussian quantiles(qqtest plot) and residuals vs leverage plot. From the qqtest plot, we can see that model 1 looks like a normal distribution, and Model 2's left tail is out of the the 90% range, suggesting it is not likely to follow a normal distriution. From the residuals vs fitted plot, points in the Model 1 spread more evenly, and Model 2 has a heavy cluster on the left. This phenomenal becomes more obvious in our third residual plot, residuals vs leverage. In this plot, we can see Model 2 has a very heavy cluster on the left, and Model 1 spreads way more evenly. This suggests that Model 1 is more likely to follow normal distribution than Model 2.

4.2 Cook's distance vs Leverage



Next, we plot the Cook's influence vs Leverage. From the plot, we can see that there are more points with lower influence and lower leverage in Model 1. Model 2 has some points with high influencial value, and many more points that have high leverage value.

4.3 Cross validation

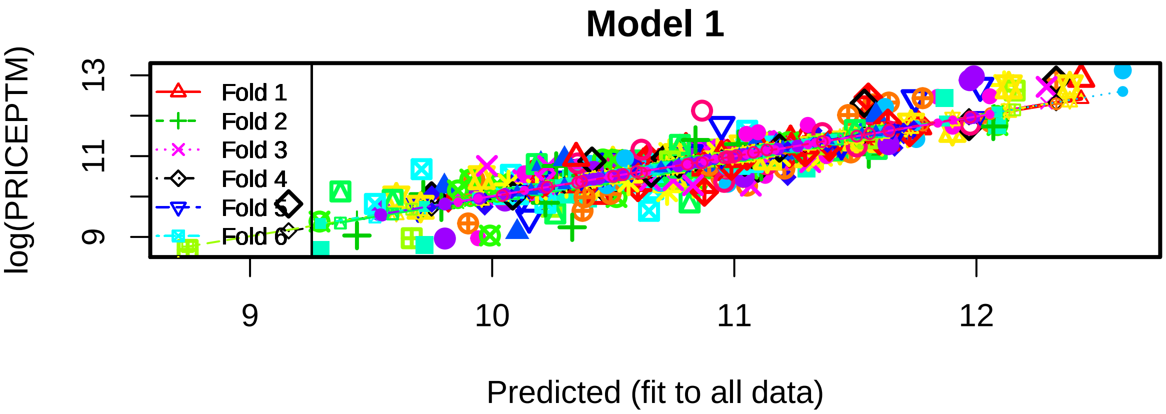
We are performing 20 fold cross-validation for both m1 and m2

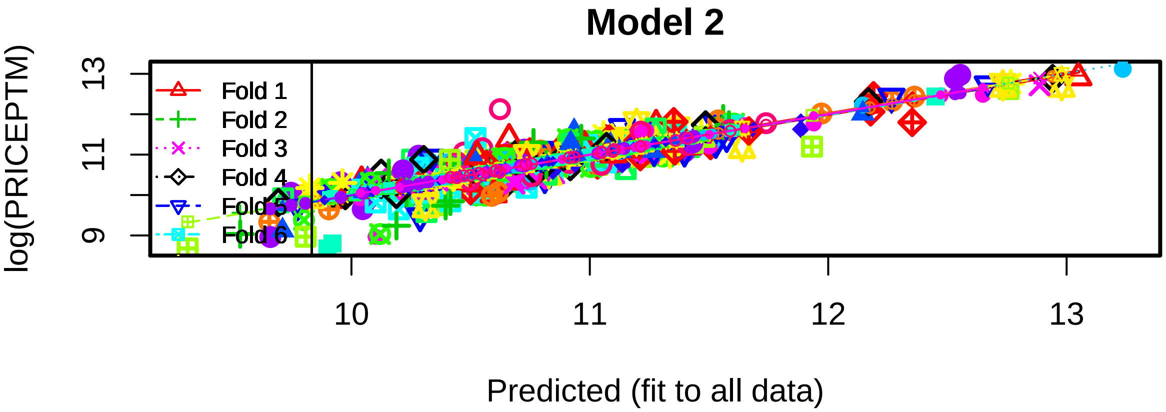
```
##
## Attaching package: 'DAAG'

## The following object is masked from 'package:MASS':
##
##     hills

## Warning in cv.lm(data = truck_data_copy, form.lm = m1, m = 20, plotit = T, :
##
##  As there is >1 explanatory variable, cross-validation
##  predicted values for a fold are not a linear function
##  of corresponding overall predicted values.  Lines that
##  are shown for the different folds are approximate

## Warning in cv.lm(data = truck_data, form.lm = m2, m = 20, plotit = T, printit = F, :
##
##  As there is >1 explanatory variable, cross-validation
##  predicted values for a fold are not a linear function
##  of corresponding overall predicted values.  Lines that
##  are shown for the different folds are approximate
```





Cross validation of model 1 gives us cross-validated standard error of estimate to be 0.128, and model 2 gives us 0.103, which is smaller than model 1. And from the plot, we can see that the model 2 fitted beter than model 1.

4.4 AIC and BIC

We compute AIC and BIC values for both models. For both AIC and BIC, model 2 has a much smaller value.

```
##    df      AIC
## m1 10 351.0673
## m2 10 250.9026
```

```
##    df      BIC
## m1 10 392.1152
## m2 10 291.9506
```

4.5 Final model and confidence intervals

Now we will choose our final model out of the two. We will compare the following 5 criterias: *Adjusted R-squared: $R_1^2 = 0.744$, $R_2^2 = 0.795$, Model 2 has a higher adjusted R-squared* Residual Plots: Model 1's residual plot looks more like a normal distribution than Model 2. *Leverage plot: Model 1's plot has more points with lower influence and lower leverage.* Cross Validation: Model 2 has lower standard error of estimate and fitted the data better *AIC and BIC: Model 2 has much lower AIC and BIC than model 1

Given the comparison above, we think Model 2 is a better model for predicting PRICEPTM. It has a higher Adjusted R-squared, and cross-validation shows that Model 2 has a stronger predictive ability. And for BIC and AIC statistics, Model 2 has a much smaller value, gives us preference to Model 2. Even though Model

2's residual is not completely normal, but in large proportion it is. So we make our final model to be Model 2, and the following is the summary of Model 2 and confidence interval of the covariates.

4.6 Confidence interval of coinficients

For model 1:

```
## lm(formula = log(PRICEPTM) ~ DISTANCE_INVERSE + PCTLOAD + PRODUCT +
##     DEREG + CARRIER.B + ORIGIN + CARRIER.C + W_P_MULTIPLY, data = truck_data)
```

```
##      (Intercept) DISTANCE_INVERSE          PCTLOAD          PRODUCT
##    10.3152138134     0.4793720155    -0.0099795883     0.0058700424
##            DEREG        CARRIER.B           ORIGIN        CARRIER.C
##    -0.3804184967    -0.4081502642    -0.1403727408    -0.0989284755
##     W_P_MULTIPLY
##     0.0001074392
```

```
##                          2.5 %        97.5 %
## (Intercept)      10.1850971022 10.4453305245
## DISTANCE_INVERSE  0.4477193746  0.5110246565
## PCTLOAD          -0.0130627285 -0.0068964480
## PRODUCT           0.0051497798  0.0065903050
## DEREG            -0.4420018170 -0.3188351765
## CARRIER.B        -0.4770635370 -0.3392369914
## ORIGIN           -0.2017992133 -0.0789462683
## CARRIER.C        -0.1767745658 -0.0210823852
## W_P_MULTIPLY     -0.0000198418  0.0002347203
```

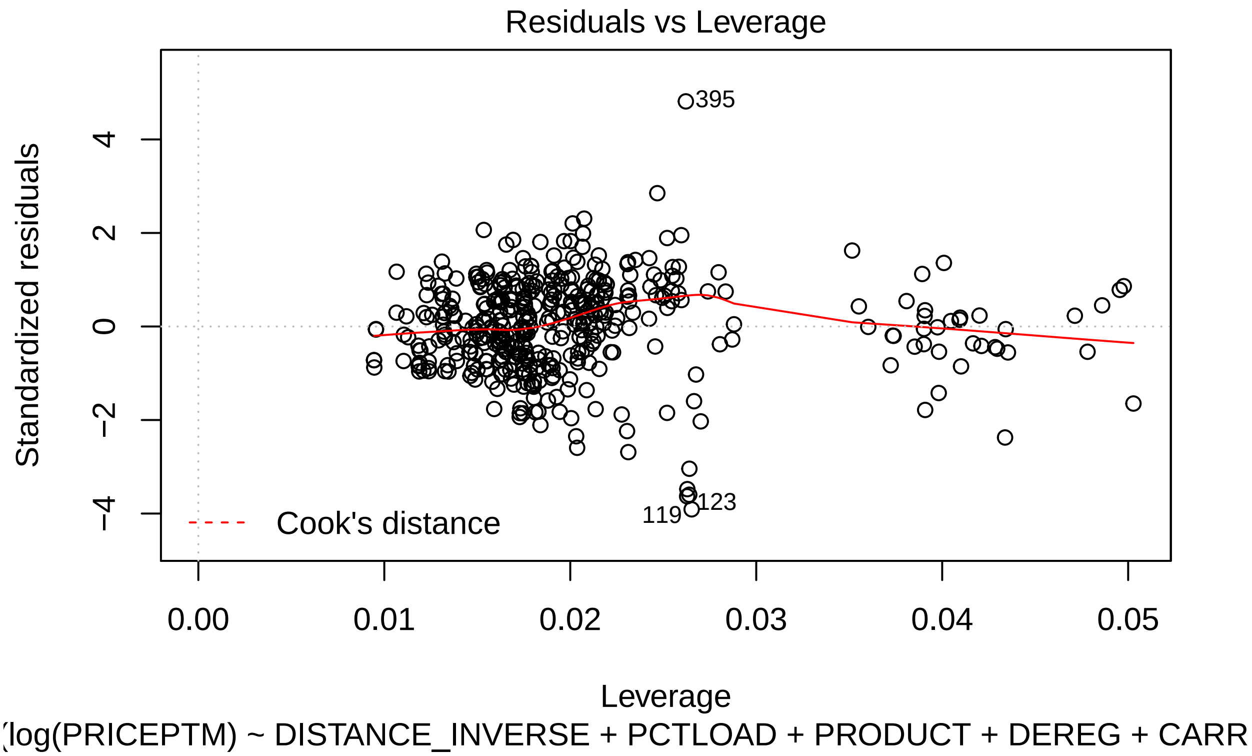
5 Discussion

After detailed analysis, we have choosen Model2 as our final model lm(formula = log(PRICEPTM) ~ DISTANCE_INVERSE + PCTLOAD + PRODUCT + DEREG + CARRIER.B + ORIGIN + CARRIER.C + W_P_MULTIPLY, data = truck_data)

5.1 What are the most important factors affecting the response? Are there any coeffcients with high p-values retained in the model? If so, why?

According to the Signif. codes in the summary of model 2, we think the most important factors are DISTANCE_INVERSE(equals to 1/DISTANCE), weight as the percentage the truck loading capacity, the product type, if deregulation is in effect and if the Carrier is in Part B. There is one covariate that has a relatively high p-value remained in our model, whcih is W_P_MULTIPLY. It remains in the model because it is the product of WEIGHT and PCTLOAD, and notice that WEIGHT is not in our final model. So this parameter basically replace's the role of WEIGHT in explaning the response variable. And its p-value is 0.098, it is not crazily high. It is just relatively high compare to other covariates.

5.2 Are there any outlying observations that might be appropriate to remove?



Yes, there is. Take a closer look at the residuals vs leverage plot, point 395, 119 looks like outliers that we should probabily remove.

5.3 Are any of the regression assumptions that the model violated? If so, which?

Yes, there is. When we fitted our model, we assume that the error should follow Normal distribution. But when we plot the qqtest for the residuals, the left tail is clearly out of the 90% range. It indicates that part of the data may not follow normal distribution as we assumed.

5.4 What are the possible deficiencies of the model. How the model can be improved?

Like we mentioned in section 2.4 and section 3.4, we are not quite sure the true relationship between PRICEPTM and DISTANCE. In our final model, we assumed that PRICEPTM has a linear relationship with the 1/DISTANCE. It is purely our assumption with no extra evidence to back it up. Even though our final model does have a better predictive power than Model 1, but that does not mean our assumption is true. In reality, the relationship could be PRICEPTM has a linear relationship with $1/DISTANCE^2$ or $e^{-DISTANCE+K}$ or some other relationships. To improve our model, we should do more analysis about their relationship and try more different fittings.

6 APPENDIX

6.1 Load data

```
truck_data = read.csv("trucking.csv") # load data
```

6.2 Transform data

```
# Remove ID
truck_data = truck_data[, !(names(truck_data) %in% c('ID'))]

# Check invalid values
apply(truck_data, 2, function(x) any(is.na(x) | is.infinite(x)))

# Check categorical variables
levels(truck_data$ORIGIN)
levels(truck_data$MARKET)
levels(truck_data$DEREG)
levels(truck_data$CARRIER)

# Use 0/1 encoding
truck_data$ORIGIN <- ifelse(grepl("JAX", truck_data$ORIGIN), 0, 1)    # change jax to 0, mia to 1
truck_data$MARKET <- ifelse(grepl("SMALL", truck_data$MARKET), 0, 1)  # change smalle to 0, large to 1
truck_data$DEREG <- ifelse(grepl("NO", truck_data$DEREG), 0, 1)       # change no to 0, yes to 1

# using one-hot encoding to transform carrier
library(ade4)
df_dummy = acm.disjonctif(truck_data['CARRIER'])       # create new variables based on catergories
colnames(df_dummy) <- gsub(" ","",colnames(df_dummy))  # remove empty spaces in the column names
truck_data$CARRIER = NULL                              # drop the originalCARREIR column
truck_data = cbind(truck_data, df_dummy)               # combine the dataframe to the original
truck_data$`CARRIER.D` = NULL
head(truck_data)

# Add new covariate W_P_MULTIPLY
truck_data$W_P_MULTIPLY = truck_data$PCTLOAD * truck_data$WEIGHT # create new covariate W_P_MULTIPLY
truck_data_copy <- truck_data                                    # reserve a copy for later use
```

6.3 Visually inspect data

```
pairs(truck_data[,], pch=".", cex=3, col="gray30")
```

6.4 Model selection

```
full_model = lm(PRICEPTM~., data=truck_data)  # full model that includes all the covriates(after featur
min_model = lm(PRICEPTM~1, data=truck_data) # minimal model that only includes intercept

# forward elimination
```

```
forward_m <- step(object = min_model, scope = list(lower = min_model, upper = full_model), direction =
backward_m <- step(object = full_model, scope = list(lower = min_model, upper = full_model), direction =
stepwise_m <- step(object = min_model, scope = list(lower = min_model, upper = full_model), direction =

# Plot residual vs fitted and QQ-plot
par(mfrow=c(2,3))
plot(fitted(forward_m), residuals(forward_m), main="forward")
plot(fitted(backward_m), residuals(backward_m), main="backward")
plot(fitted(stepwise_m), residuals(stepwise_m), main="stepwise")
qqnorm(residuals(forward_m), main="forward")
qqnorm(residuals(backward_m), main="backward")
qqnorm(residuals(stepwise_m), main="stepwise")

# Box-Cox inspection
library(MASS)
par(mfrow=c(1,3))
b1 = boxcox(forward_m)
b2 = boxcox(backward_m)
b3 = boxcox(stepwise_m)
index1 = rev(order(b1$y))[1]
index2 = rev(order(b2$y))[1]
index3 = rev(order(b3$y))[1]
b1$x[index1]
b2$x[index2]
b3$x[index3]

# Automated model selection
full_model = lm(log(PRICEPTM)~., data=truck_data)
min_model = lm(log(PRICEPTM)~1, data=truck_data)
forward_m <- step(object = min_model, scope = list(lower = min_model, upper = full_model), direction =
backward_m <- step(object = full_model, scope = list(lower = min_model, upper = full_model), direction =
stepwise_m <- step(object = min_model, scope = list(lower = min_model, upper = full_model), direction =

# Plot residual vs fitted and QQ-plot to verify our fitted model
par(mfrow=c(2,3))
plot(fitted(forward_m), residuals(forward_m), main="forward")
plot(fitted(backward_m), residuals(backward_m), main="backward")
plot(fitted(stepwise_m), residuals(stepwise_m), main="stepwise")
qqnorm(residuals(forward_m), main="forward")
qqnorm(residuals(backward_m), main="backward")
qqnorm(residuals(stepwise_m), main="stepwise")

# Check relationship between DISTANCE and PRICEPTM
plot(x=truck_data$DISTANCE, y=truck_data$PRICEPTM)

# Fit the model with DISTANCE_INVERSE added
full_model = lm(PRICEPTM~., data=truck_data)
boxcox(full_model)
```

6.5 Automated Model selection with DISTANCE INVERSE added

```
truck_data$DISTANCE_INVERSE = 1 / truck_data$DISTANCE     # create new covariate DISTANCE_INVERSE
truck_data$DISTANCE = NULL                                # drop the DISTANCE column

full_model = lm(log(PRICEPTM)~., data=truck_data)  # full model that includes all the couriates(after f
min_model = lm(log(PRICEPTM)~1, data=truck_data) # minimal model that only includes intercept

Inverse_1 <- step(object = min_model, scope = list(lower = min_model, upper = full_model), direction =
Inverse_2 <- step(object = full_model, scope = list(lower = min_model, upper = full_model), direction =
Inverse_3 <- step(object = min_model, scope = list(lower = min_model, upper = full_model), direction =
```

6.6 Model diagnose

```
library(qqtest)
m1 <- forward_m
m2 <- Inverse_1

par(mfrow=c(2,3))
# fitted vs residuals
plot(fitted(m1), residuals(m1), main="Model 1")
qqtest(residuals(m1), main="Model 1")
plot(ls.diag(m1)$hat, residuals(m1), xlab="leverage", ylab="residuals")

plot(fitted(m2), residuals(m2), main="Model 2")
qqtest(residuals(m2), main="Model 2")
plot(ls.diag(m2)$hat, residuals(m2), xlab="leverage", ylab="residuals", main="Model 2")

# Cook's distance plot
c1 = cooks.distance(m1)
c2 = cooks.distance(m2)
l1 = ls.diag(m1)$hat
l2 = ls.diag(m2)$hat

par(mfrow=c(1,2))
plot(x=l1, y=c1, main="Model 1", ylim=c(0,0.08), xlim=c(0,0.06), xlab="leverage", ylab="Cook's influence
plot(x=l2, y=c2, main="Model 2", ylim=c(0,0.08), xlim=c(0,0.06), xlab="leverage", ylab="Cook's influence
```

6.7 Cross validation

```
library("lattice")
library("DAAG")
par(mfrow=c(2,1))

r1 = cv.lm(data=truck_data_copy, form.lm=m1, m= 20, plotit = T, printit = F, main="Model 1")
r2 = cv.lm(data=truck_data, form.lm=m2, m= 20, plotit = T, printit = F, main = "Model 2")
```

6.8 AIC and BIC

```
AIC(m1, m2)
BIC(m1, m2)
```

6.9 Confidence interval

```
m2$call
coef(m2)
confint(m2)
```