Assignment 1 7040

Songze Yang, u7192786

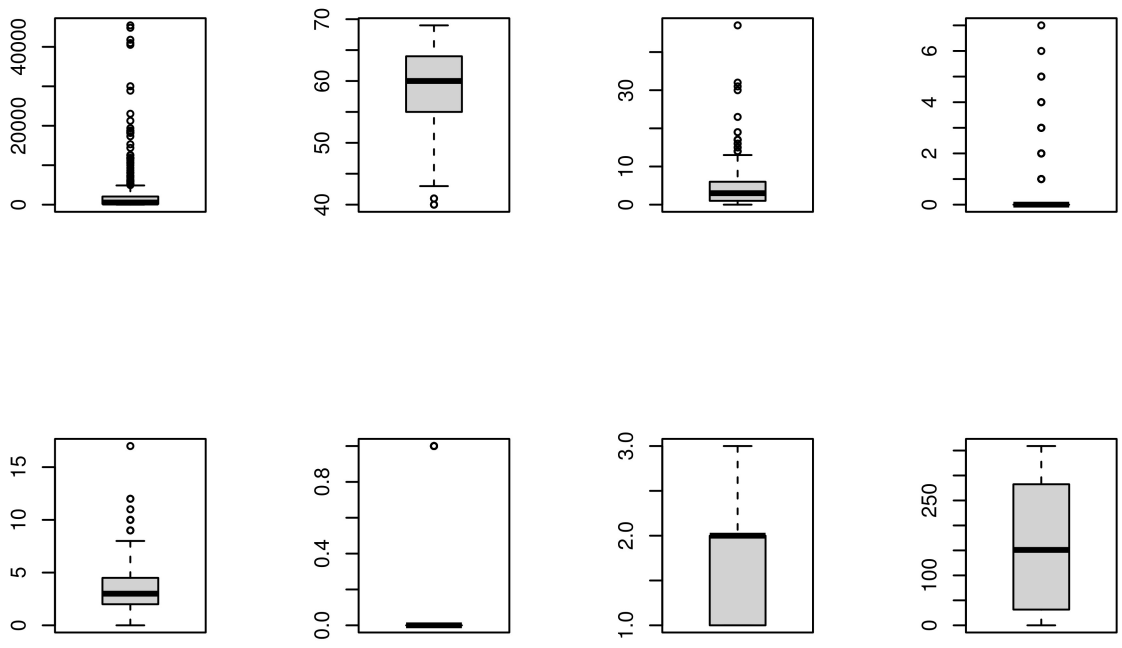
3/14/2022

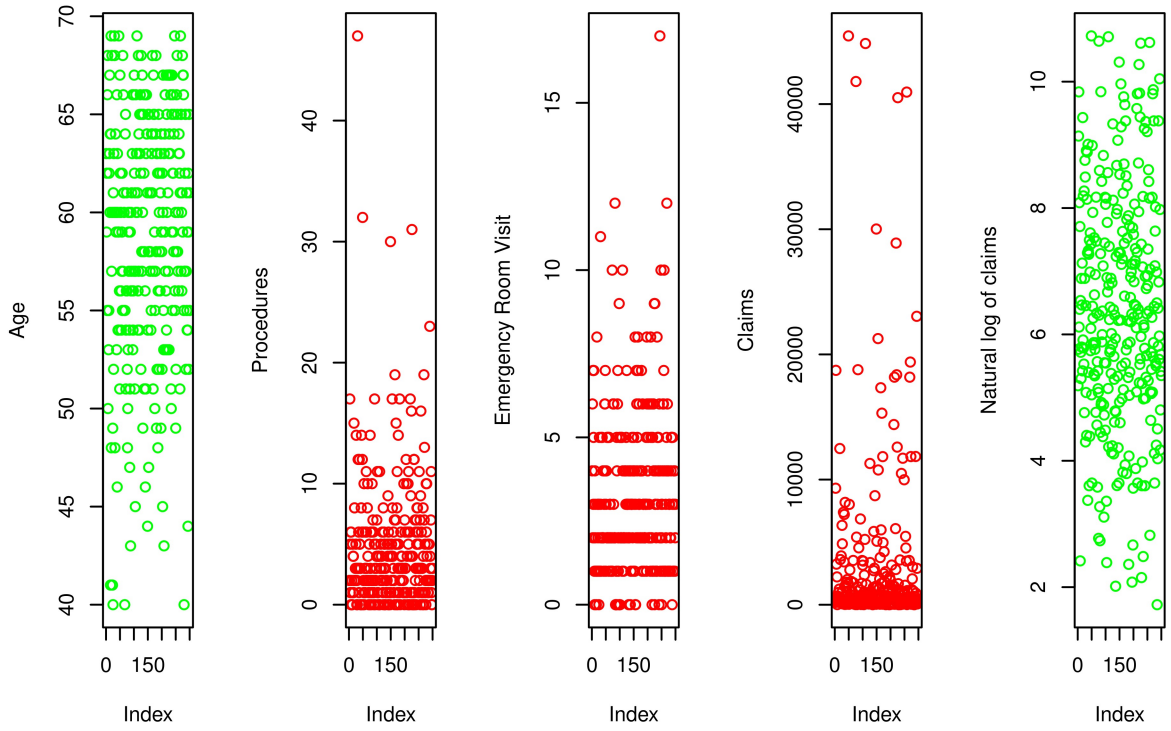
The data containing the health insurance claim over 2 years time in the US are included in the file called 'Data'. We split the data into training data, which include the first 300 claims, and test data, the other 308 cases. The data will be used to examine the relationship between insurance claims and all the covariates.

(a)

We can investigate outliers by plotting box plot:

```
par(mfrow = c(2,ncol(train.data)/2))
invisible(lapply(1:ncol(train.data), function(i)boxplot(train.data[,i])))
```





The result shows that the Y(Claims), x1(Age), x2(Procedures) and x4(Emergency Room Visits) have potential outliers in the data. The x5(Complications) have mean of approximately 0.04667, suggesting that complications are raw event. As the x5(Complications) only takes 0 and 1 value, we will not consider 1s as outliers, the reason that applied to x3(Prescribed Drugs). The other variables seem fine from the box plots.

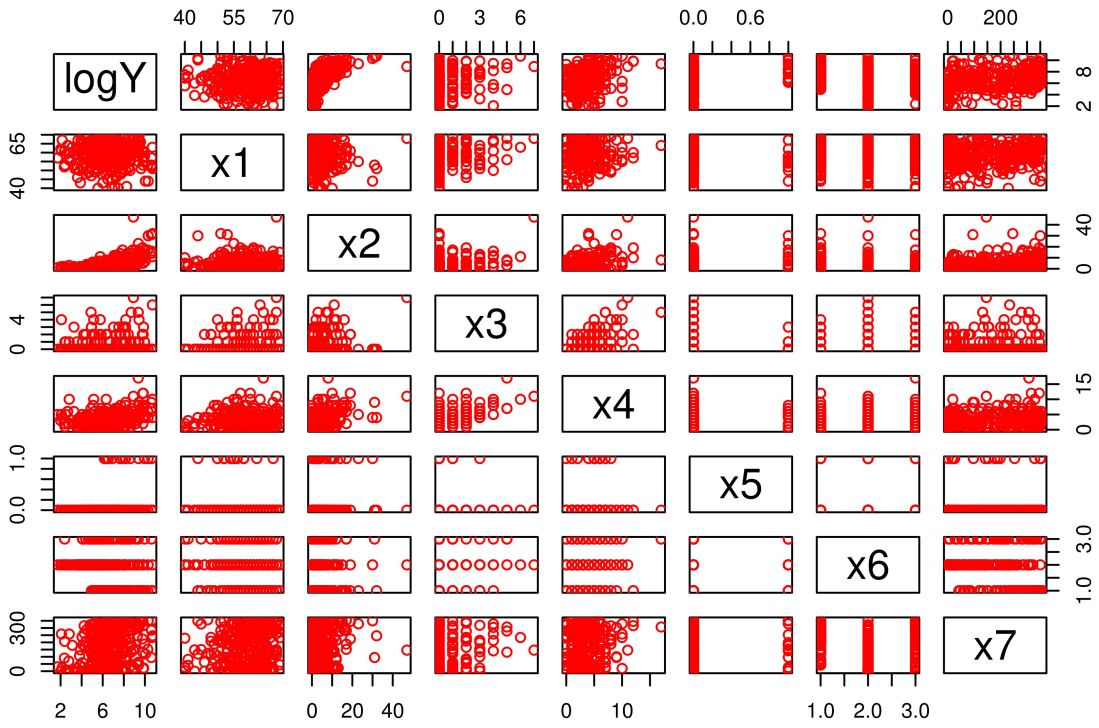
A small number of people make insurance claims at age 40, corresponding to the outliers from our box plot. However, the age data should not be excluded as we do not want left censored data. Also, many points on our scatter plot spread out in the range of 40 to 50, therefore, insurance claims made at age 40 to 50 should not be rare event.

The x2(Procedures) and x4(Emergency Room Visits) look similar on the scatter plot. Both has a few high values show on the top of the graph, showing that people need more prescribed drugs or conduct more emergency room visits in some cases.

The Y(Claims) variable seems spread out the most. The main points center around the 0 to 10,000, while many points reach out from 10,000 to more than 40,000. However, after a log transformation, the problem of unusual points become mild.

This suggests a transformation of Y is appropriate. We further look at the transformation of features.

```
train.data$logY<-log(train.data$Y);test.data$logY<-log(test.data$Y);pairs(train.data[,c(9,2:8)], col =
```



The scatter plot matrix shows no obvious curvature. Therefore, the transformation of covariates is dismissed.

(b)

Fitting the linear model between Y and x3, we reach the summary table as shown below.

```
fit<-lm(logY~x3, data = train.data);sumary(fit)
```

```
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 6.254244   0.116796 53.5483 < 2.2e-16
## x3          0.289141   0.099065  2.9187  0.003783
##
## n = 300, p = 2, Residual SE = 1.86982, R-Squared = 0.03
```

Here, we conduct the hypothesis testing on the β_1.

$$H_0 : \beta_1 = 0, H_A : \beta_1 \neq 0$$

As the t-value in our linear fit is 2.9187, this suggests the linear relationship exists as the coefficient for x3 is significant at 95% confident level. We reject the null hypothesis and conclude H_A.

We continue to explore the ploynominal model.

```
fit2 <- lm(logY~poly(x3 , 2), data = train.data);fit3 <- lm(logY~poly(x3 , 3), data = train.data)
sumary(fit2); sumary(fit3)
```

```
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   6.38436    0.10808 59.0685 < 2.2e-16
## poly(x3, 2)1  5.45743    1.87207  2.9152  0.003825
## poly(x3, 2)2  0.99632    1.87207  0.5322  0.594985
##
## n = 300, p = 3, Residual SE = 1.87207, R-Squared = 0.03
```

```
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   6.38436    0.10809 59.0643 < 2.2e-16
## poly(x3, 3)1  5.45743    1.87220  2.9150  0.003829
## poly(x3, 3)2  0.99632    1.87220  0.5322  0.595012
## poly(x3, 3)3  1.83263    1.87220  0.9789  0.328448
##
## n = 300, p = 4, Residual SE = 1.87220, R-Squared = 0.03
```

F-tests are applied here, for the quadratic model:

$$H_0 : \beta_1 = \beta_2 = 0, H_A : At\ least\ one\ of\ \beta_1\ or\ \beta_2 \neq 0$$

For the cubic model:

$$H_0 : \beta_1 = \beta_2 = \beta_3 = 0, H_A : At\ least\ one\ of\ \beta_1\ or\ \beta_2\ or\ \beta_3 \neq 0$$

For the F-tests of 4.391 and 3.246, those show that both the second and third polynomial model are significant at 5% level, advising that the higher terms improve the model residual sum of square from using the mean of Y.

```
## Analysis of Variance Table
##
## Model 1: logY ~ x3
## Model 2: logY ~ poly(x3, 2)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    298 1041.9
## 2    297 1040.9  1   0.99265 0.2832  0.595
```

```
## Analysis of Variance Table
##
## Model 1: logY ~ x3
## Model 2: logY ~ poly(x3, 3)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    298 1041.9
## 2    296 1037.5  2    4.3512 0.6207 0.5383
```

However, partial F-test should be conducted to see whether the polynomial improves significantly from the linear model. The test is shown below:

For quadratic model:

$$H_0 : \beta_2 = 0, H_A : \beta_2 \neq 0$$

For cubic model:

$$H_0 : \beta_2 = \beta_3 = 0, H_A : At\ least\ one\ of\ \ \beta_2\ or\ \beta_3 \neq 0$$

The higher terms improve the model very slightly, because the partial F-test for the second and third model is both not significant, namely with F-value 0.2832 and 0.6207 and p_value 0.595 and 0.5383. Hence, we conclude the linear model. The model should have flexibility as the linear model.

(c)

We can calculate the training MSE of our model as follow:

```
## [1] 3.472909
```

```
## [1] 3.4696
```

```
## [1] 3.458405
```

The training MSE decreases, indicating the use of the higher order terms. The result is different from 1b, however, it is accorded to the fact that the R^2 will always increase as more parameter added. We can calculate the test MSE using the method shown below:

```
## [1] 3.365838
```

```
## [1] 3.387838
```

```
## [1] 3.495392
```

The test MSE suggests the opposite. The MSE of the models increases with the term added into the model. This suggests that the linear model is the best among the three, result that is accorded to the 1b.

(d)

First, we fit the full model as follow:

```
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  6.58731456  0.74230154  8.8742 < 2.2e-16
## x1          -0.01231646  0.01224087 -1.0062   0.31517
## x2           0.19086283  0.01547202 12.3360 < 2.2e-16
## x3          -0.02006943  0.08453373 -0.2374   0.81250
## x4           0.07577304  0.03915775  1.9351   0.05395
## x5           0.67190129  0.36544522  1.8386   0.06700
## x6medium    -1.23654343  0.22343774 -5.5342 6.979e-08
## x6high      -0.60888289  0.23611632 -2.5787   0.01041
## x7           0.00047785  0.00083068  0.5753   0.56556
##
## n = 300, p = 9, Residual SE = 1.29025, R-Squared = 0.55
```

The full model reveals potentially x1, x3 and x7 may be insignificant to the model individually. We can use a backward selection based on partial F test to double comfirm.

```
##
##
##                            Elimination Summary
## ----------------------------------------------------------------------
##         Variable                  Adj.
## Step    Removed     R-Square    R-Square     C(p)        AIC        RMSE
## ----------------------------------------------------------------------
##    1    x3            0.5479       0.537    5.0564    1013.1832    1.2882
##    2    x7            0.5474      0.5381    3.3813    1011.5179    1.2867
##    3    x1             0.546      0.5383    2.2526    1010.4137    1.2864
## ----------------------------------------------------------------------
```

The result suggests that we can drop x1, x3 and x7 from the full model as we analysed. The hypothesis tested by partial F-test is shown as:

$$H_0 : \beta_p = 0, H_A : \beta_p \neq 0 \ , \ in\ the\ presence\ of\ all\ other\ variable$$

The final model we choose is:

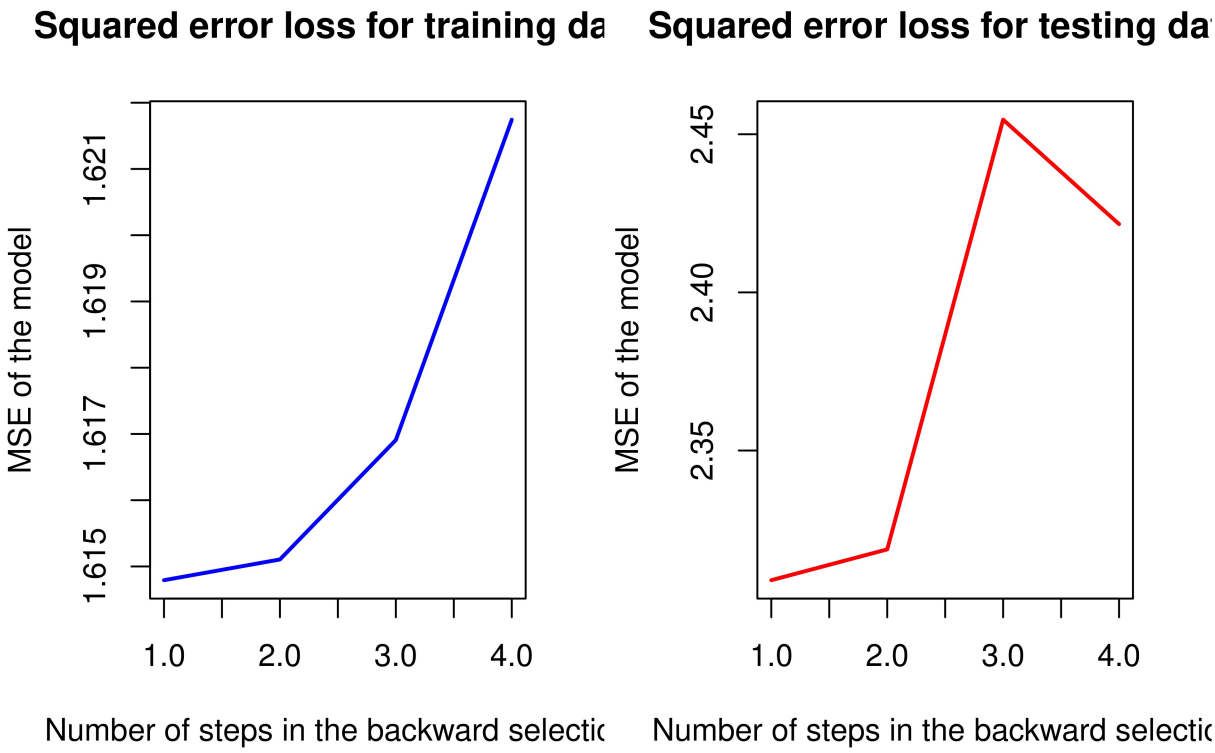
```
##
## Call:
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),
##     data = l)
##
## Coefficients:
## (Intercept)           x2           x4           x5     x6medium       x6high
##     5.97921      0.19282      0.07073      0.71571     -1.31537     -0.63572
```

Our final model fits as follow:

$$log(Y) = 5.97921 + 0.19282 * x_2 + 0.07073 * x_4 + 0.71571 * x_5 - 1.31537 * x_6 medium - 0.63572 * x_6 high$$

(e)

The MSE for train and test data are as followed:



The backward selection process gives us slightly higher training MSE. The testing MSE during the backward selection process appears to incline at first and decline slightly in the final model. In conclusion, the training and testing MSE both suggests that we should use the full model.

(f)

The interaction model of x6(Comorbidities) and other variables can be fitted as shown:

```
##
## Call:
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),
##     data = l)
##
## Coefficients:
## (Intercept)           x2           x4           x5     x6medium       x6high
##     5.97921      0.19282      0.07073      0.71571     -1.31537     -0.63572
```

A backward selection approach agrees with our approach as it reaches the same model as before. To investigate whether interactions are significant, we can do a partial F test on all the interactions.

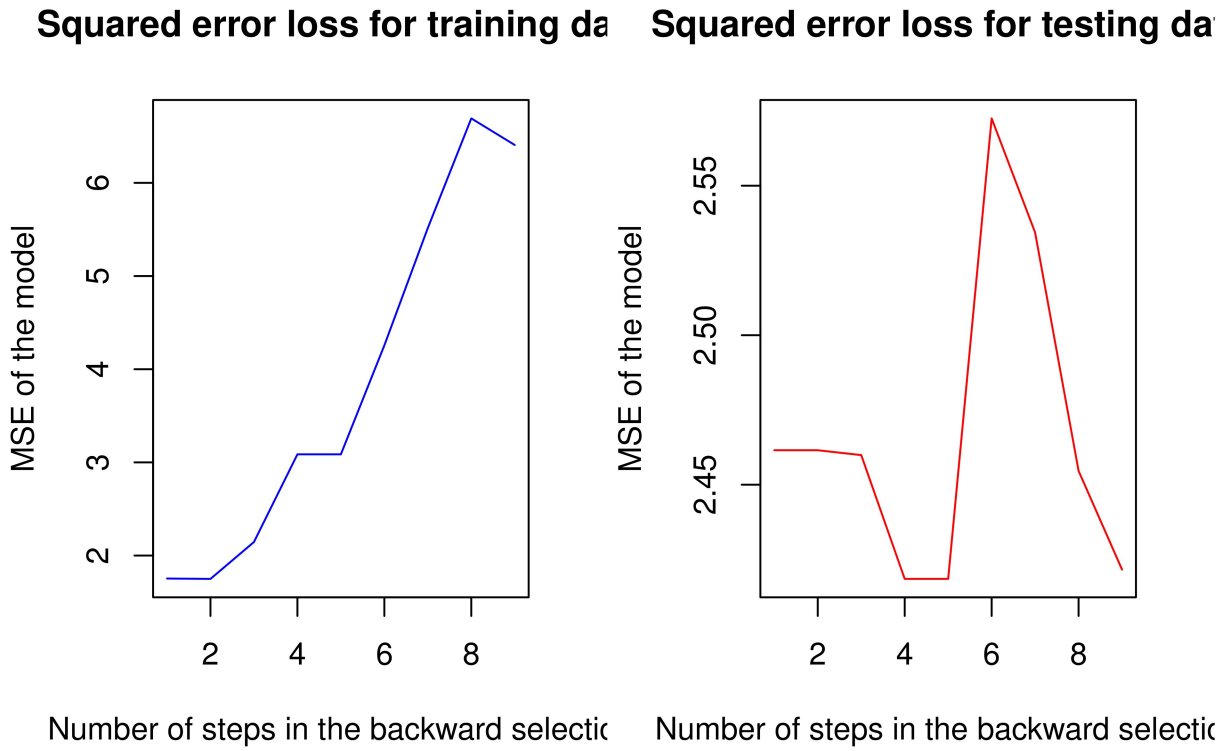
$$H_0 \ : \textit{all the interactions are } 0, H_A : \textit{at least one interaction is not } 0$$

The test produce a 0.5189 F-value with p-value of 0.9019, which is not significant. We fail to reject the null hypothesis and conclude H_0. This leads us to conclude that all the interaction is 0, therefore, there is not significant interaction between x6(Comorbidities) and all other features.

(g)

We can fit all the model in our backward selection again and calculate the MSE.

```
## [1] 1.753509 1.749599
```




```
## [1] 2.418497 2.418497 2.421596
```

The training squared error lost falls off slightly and quickly climbs up until the final model. The training MSE gets minimized at the step two, which is the step producing the best model by training MSE. The model we should pick is:

```
## lm(formula = logY ~ x1 + x2 + x4 + x5 + x6 + x7 + x1:x6 + x2:x6 +
##     x3:x6 + x4:x6 + x6:x7, data = subset(train.data, select = c(-Y)))
```

The squared error lost for test data fluctuates through the backward selection process. The model first decreases in MSE until step 5, then undergos a upward and then downward trend, eventually settles at the step 9. The minimum MSE of the models achieves at step 5, suggesting that the best model based on testing MSE should be model 5 but not the final model for the backward selection approach. The model we use is:

```
## lm(formula = logY ~ x1 + x2 + x4 + x5 + x6 + x2:x6 + x4:x6 +
##     x6:x7, data = subset(train.data, select = c(-Y)))
```

(h)

In this section, we examine ways in the regression modelling and how they produce better model based on the testing MSE. In the earlier section, we mainly use the backward selection, but we will use the step wise approach in terms of the criteria of partial F test, AIC and BIC. The three results are shown below. We then apply the exhausted search using BIC after. I will consider curvature also.

```
modelboth_p<-ols_step_both_p(lm(logY~., data=subset(train.data, select=c(-Y))),pent = 0.05,prem = 0.1)
nullmodel<-lm(logY~1, data = train.data)
modelboth_aic<-stepAIC(nullmodel, direction = "both", scope = list(lower = ~1, upper = ~x1+x2+x3+x4+x5+
n<-dim(train.data)[1]
modelboth_bic<-stepAIC(nullmodel, direction = "both", k=log(n),scope = list(lower = ~1, upper = ~x1+x2+
```

We can see the step wise processes with regard to partial F test and the AIC produce the same model. We can compare its testing MSE with the BIC model:

```
## [1] 2.087747
```

```
## [1] 2.104538
```

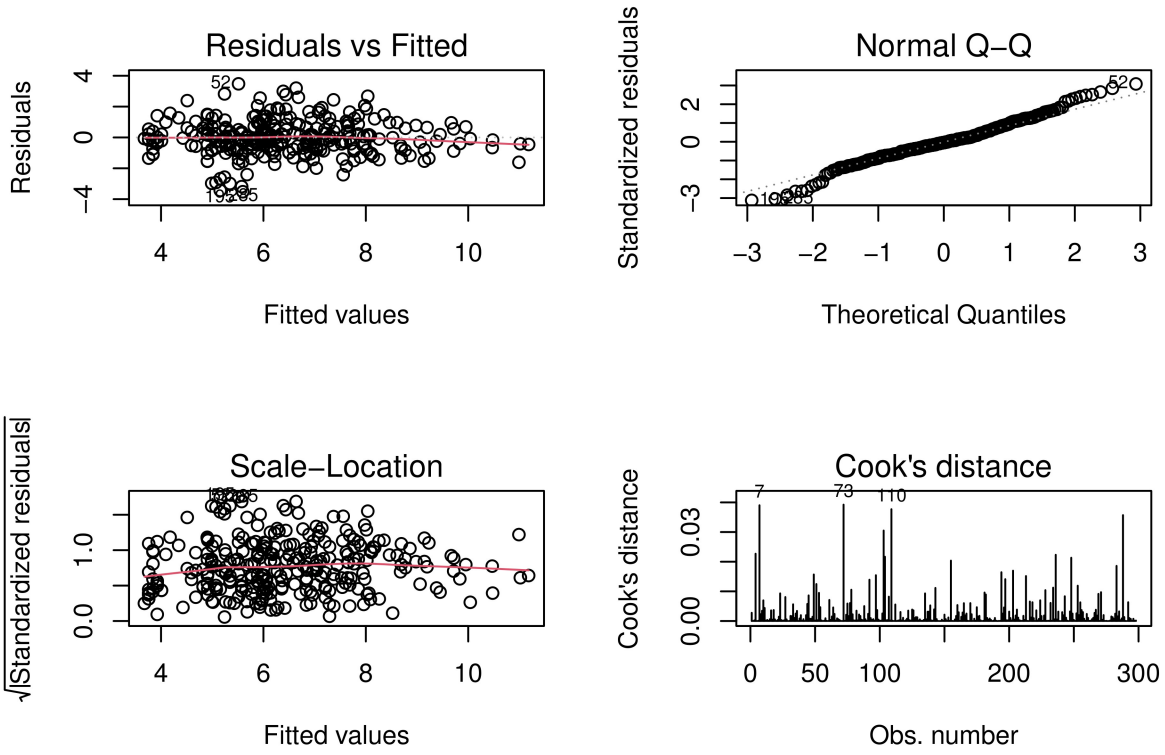
As we can see the results, the test data suggest the AIC model is a better model providing a lower testing MSE.

(i)

The model is shown as below.Three high leverage points appear as point 31,110 and 259. We consider to remove the highest two influential points and to refit the model again.

```
## lm(formula = logY ~ srx2 + x6 + x4 + x5, data = train.data)
```

```
modelaicad<-lm(logY ~ srx2 + x6 + x4 + x5, data = train.data[c(-31,-259),])
par(mfrow = c(2,2))
plot(modelaicad,which = c(1,2,3,4))
```



New influential points appear but as they are all similar value, which has much smaller cook distance than before. We will consider to keep them in our model. To check if the variables are correlated, the variance inflation factor is shown below. The model can be expressed as:

$$log(Y) = 4.949892 + 1.041477 * sqrt(x_2) - 1.209650 * x_6 medium - 0.488577 * x_6 high + 0.060145 * x_4 + 0.717215 * x_5$$

```
confint(modelaicad)
```

```
##                   2.5 %      97.5 %
## (Intercept)  4.48683937  5.17280389
## srx2         0.93888682  1.18800244
## x6medium    -1.44824439 -0.86104979
## x6high      -0.87572749 -0.09455589
## x4           0.02455508  0.14257240
## x5           0.01946922  1.26481725
```

As we can see from confident interval table, the top 4 predictors have narrow confident intervals and only x5 have large confident interval compared to others, but not too large as well. The coefficient in our model is reasonably checked.

The coefficient in our model can be interpreted as below:

Holding all the variables in our model constant: The procedure increases one unit, the claims payment increase the e to the power of square of 1.041477 times.

More times of emergency room visits correspond to the claims payment decrease e to the power of -0.060145 times.

The occurrence of complications increases one times, the claims payment decrease e to the power of -0.717215 times.

The median level of comorbidities decreases the claim payment by e to the power of -1.041477 times, while the high level of comorbidities decreases the claim payment by e to the power of -0.488577 times.