**Objective 1:**

This is the question we are asking: can we use the parameters that we could know before running the report to predict how much time this report will cost to delivery?

We are using all the parameters within the dataset to fit the response parameter “ReportDeliveryTime”. Although within the report, intuitively we know a few parameters are highly important and contribute to the delivery time, for example, ReportBuildTime, Queue Time, Lag Time and ReportBytes, we can not use them because realistically all those above parameters are unknowns before we submit the report.

**The Intuitive Model**

From the EDA analysis we To build an intuitive model, we learn that there are a lot of category variables which doesn’t show an apparent linear relationship with ReportDeliveryTime. Also, the variable ReportDeliveryTime is heavily right skewed so it may need a log transformation.

With some pre-knowledge of the dataset, we know that the ReportID(Type of Report) may related to the ReportBuildTime. We first convert the ReportID to factors and do a simple form linear regression, we got an R^2 of 0.79 which is very good, however the ReportID has over 1500 factor levels which make the explanation difficult, so we have to make other fittings and temporarily ignore ReportID.

Some additional pre-knowledge of the dataset make us filter the dataset further: Only the 3 “working” servers are included and only the TestProdIndicator equal to”P” are preserved. Because the amount of data are very large and we don’t find a dependency of ReportID with ReportDeliveryTime, we choose a single type of report “ReportID=93”, which is roughly 1/10 of the overall dataset as a representative subset and use it for later analysis.

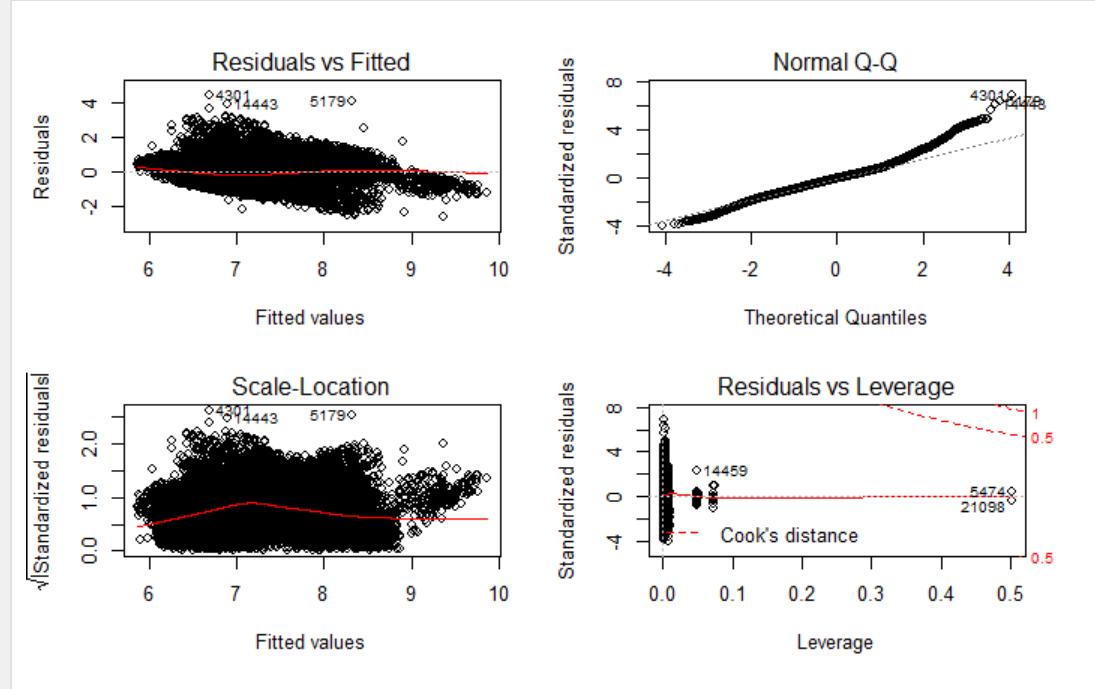
With this subset, we filtered out a few more parameters and the left 10 parameters for fittings are: Server, ReportCategory, SchedFreq, HourOfDay, DayOfMonth, ReportCategory,Server, AgentCount, GroupCount,DelivMethod. First we did a fit with all parameters and get a resulted R^2 of 0.36. It was not too bad.

**Assumptions:**

We check the assumptions: when validating a linear regression we have to check normality, linearity, constant variance (equal spread), and independence.

The q-q plot below (and previous histogram) shows that the assumption of normality may be violated so we need to transform response variables, we did another fitting with log transformation performed.

After the transformation performed we check on the assumptions again, we observe that this time the q-q plot exhbit that the data is approximately normal.



**Feature Selection:**

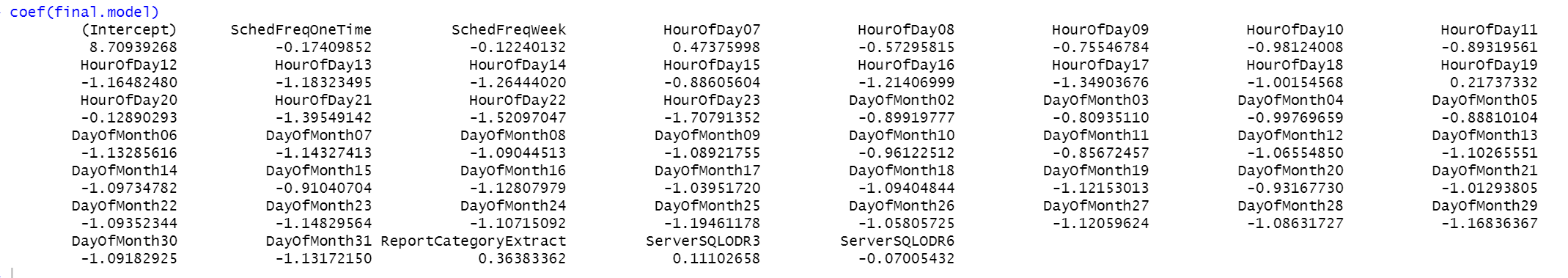
After the log transformation we re-do the fitting and this time we get a resulted R^2 of 0.54! This is good but because we have variables that have a lot of factor levels, we decide to do a round of feature selection using forward, backward and other criteria.

Using forward model selection and nvmax=30, we recognize that the BIC, adjusted R^2 and RSS shown in plot all show a “plateau” after roughly 25-30 parameters, indicating that we can get best fit and avoid further overfitting. By checking the coefficients we identify that for example, the variable SchedFreqOneTime and SchedFreqWeek are both significant so we include the variable SchedFreq in the final model. Some variables related to HourOfDay are siginificant, to include them all I have to include the HourOfDay as a variable. The final selection composes of 5 variables and the model is below:

final.model<-lm(log(ReportDeliveryTime)~SchedFreq + HourOfDay + DayOfMonth + ReportCategory + Server,data=reports\_serv\_new)

Alternatively we can use a stepwise selection process and recognize that the BIC, adjusted R^2 and RSS shown a similar trend. There are 4 variables that appear to be significant: SchedFreq, HourOfDay, DayOfMonth and ReportCategory. The lowest BIC/RSS is similar to forward selection.

**Interpretation:**

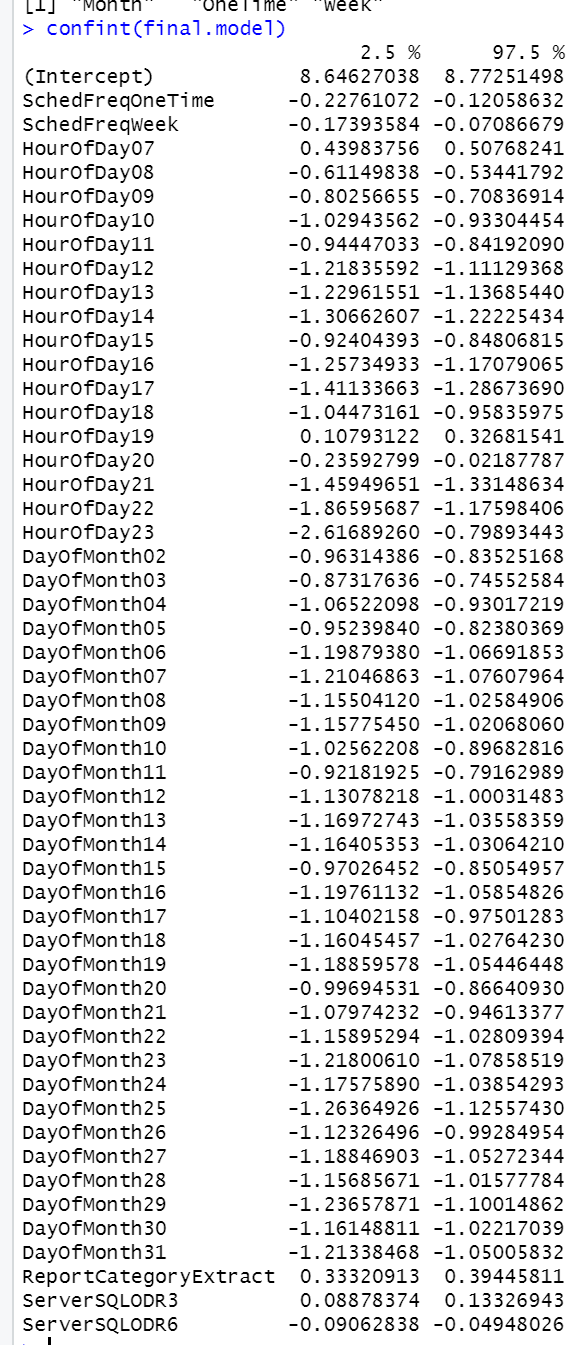


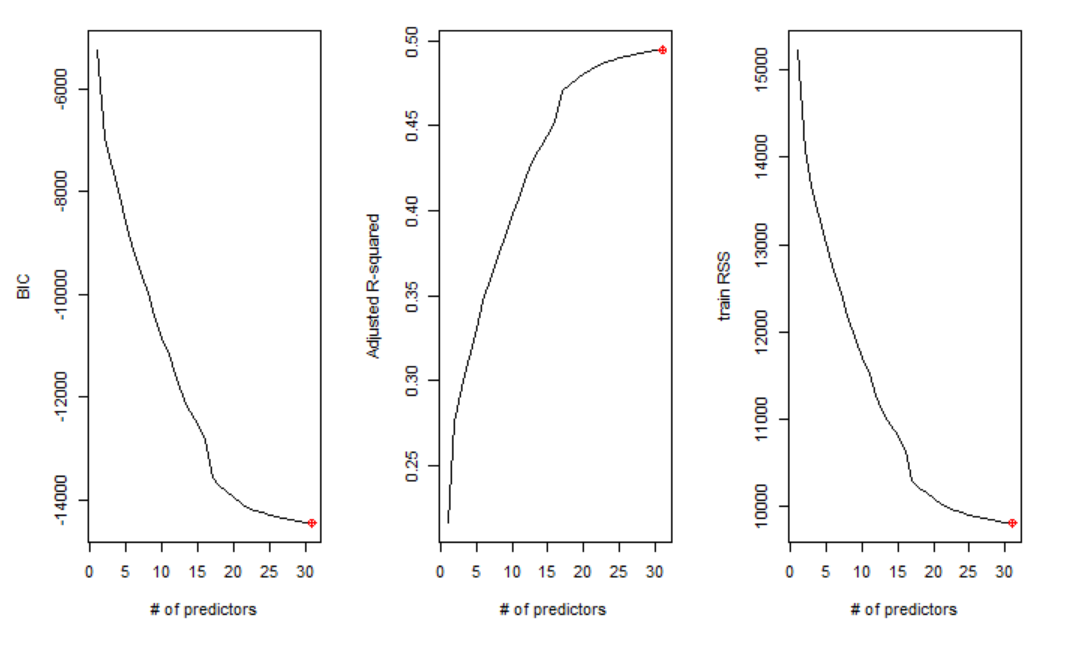
In order to interpret the outcome of the model we need to check the coefficients in the above plot and provide an explanation. In each category variable we can see one level has been chosen to use as the reference. We can check and find all the variables which listed below are with statistically significant p-value.

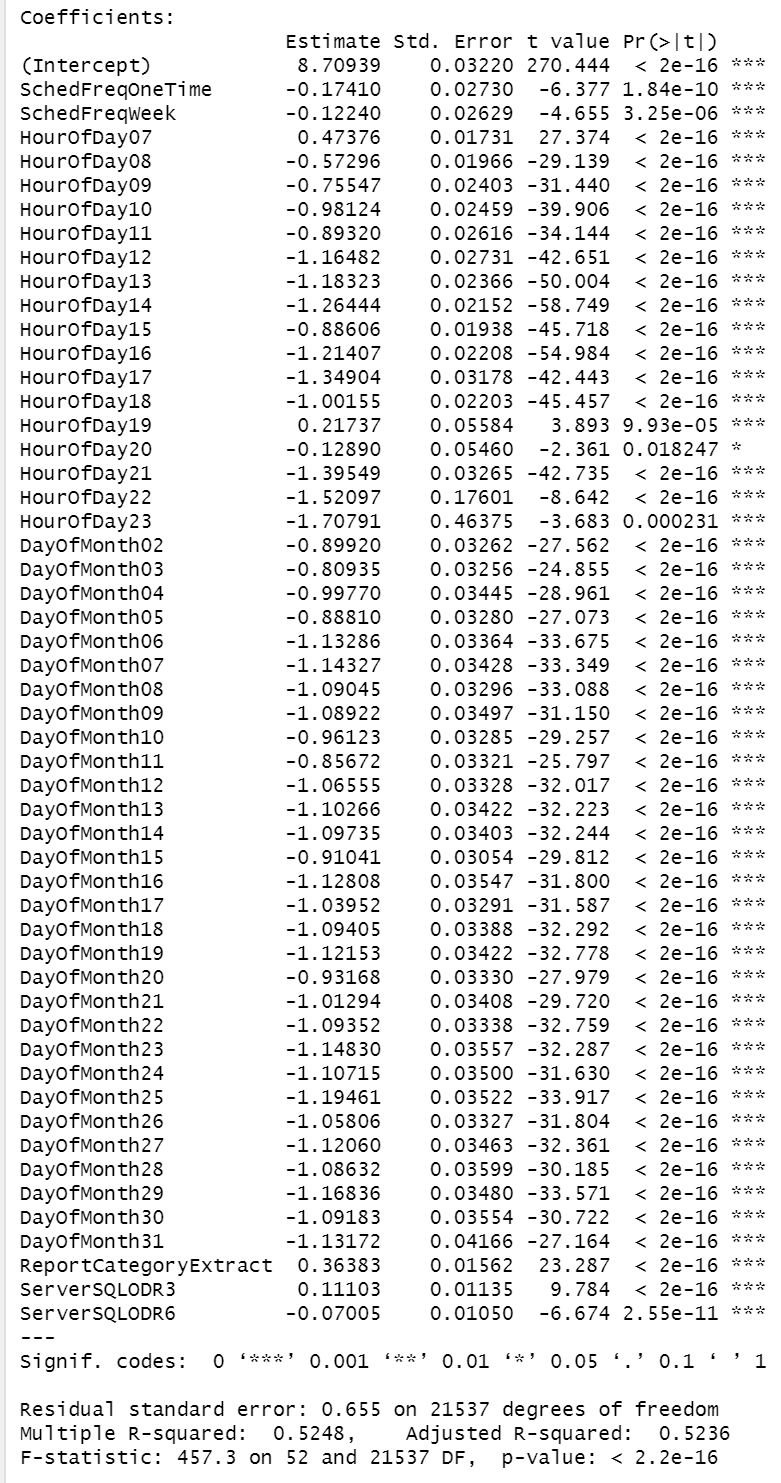
The intercept coefficient is the mean estimate for the report delivery time while keep other variables constant. The SchedFreqOneTime coefficient can be interpreted as the mean delivery time change from the reference of SchedFreqMonth. The HourOfDay07 can be interpreted as the mean delivery time change from the reference of HourOfDay06. The DayofMonth02 can be interpreted as the mean delivery time change from the reference of DayofMonth01. The ReportCategoryExtract can be interpreted as the mean delivery time change from the reference of ReportCategoryClaims. The ServerSQLODR3 can be interpreted as the mean delivery time change from the reference of ServerSQLODR2.

The logged variables need to be transformed back from their log in order to be properly interpreted in the context of other variables. Because this is a liner-log model, when interpreting the coefficient of ReportDelivTime, a one unit change in the ReportDelivTime would result in an exp(Beta) change in the mean of y.

The confidence intervals for the parameter estimates are in the table below. The confidence intervals show that all the listed variables don’t contain zero at the 95% level. This indicates that we can reject the null hypothesis and say that these estimates are significantly different from zero.







**Talk about the fitting and MSE**

When we initially build our intuitive model we did not split our dataset into train and test portions. We further conduct LASSO, Ridge and Elastic net for model fitting and there’s no fair way to compare the model of all above methods without a train/test experiment.

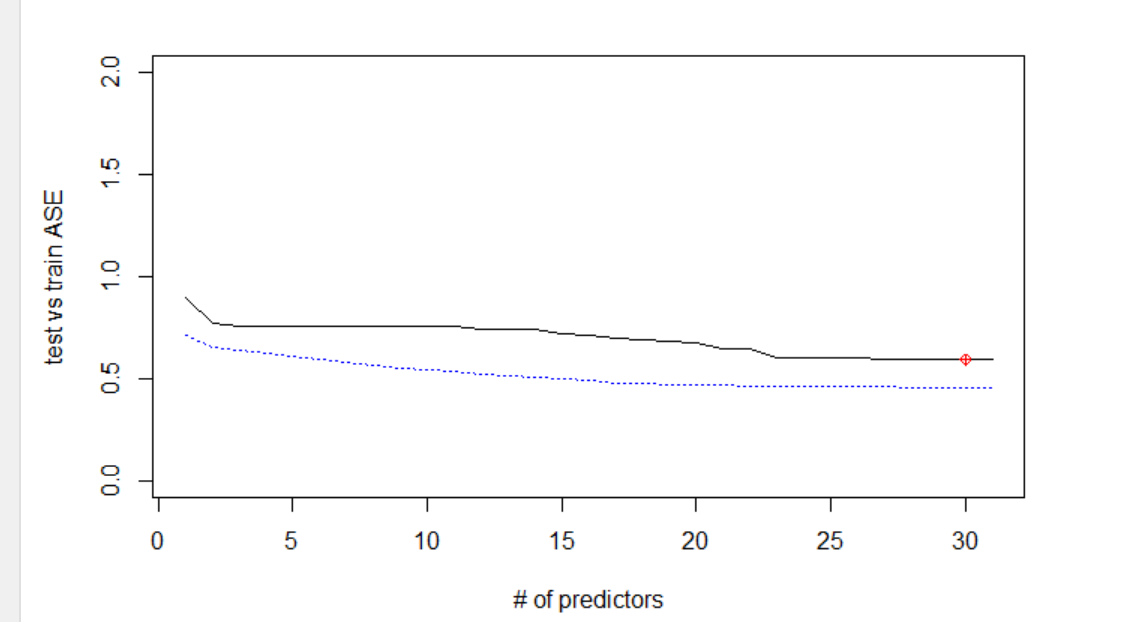
So we perform a roughly 70-30 split on our original dataset to create a train and a test set, then use our train set to fit our optimized linear model and create a prediction vector from our test dataset.

Then we can calculate our MSE of this model and compare it to those of forward, stepwise, LASSO, Ridge and Elastic net, we get the result below.

|  |  |
| --- | --- |
| Method | RMSE |
| Lasso | 1967.324 |
| Ridge | 1908.051 |
| Elastic Net | 1905.987 |
| Forward | 2099.483 |
| Stepwise | 2099.574 |
| LM (5 variables) | 1963.45 |

From the table we can see that Lasso, Ridge and Elastic Net all give us a RMSE that close to each other, and slightly smaller than the multi-linear model. And the forward and stepwise model gives us a higher-than-other RMSE. Because the Lasso models used all the variables we are at a risk of overfitting.

Below shows a test versus train ASE plot and we can see that the test and train error are very close with a suggestion of including ~30 variables (really 5 category variables). This is a graphical representation of the results of the forward modele.



**Conclusions:**

In conclusion, the mulit-linear model with 5 predictors leveraging a logged response gave the best MSE with avoidance of overfitting. We used a train and test split to validate the MSE across many method and reach this conclusion.