D4 – Resampling

I- Model Assessment

Resampling methods are an indispensable tool in modern statistics. They involve repeatedly drawing samples from a training set and refitting a model on each sample in order to obtain additional information about the fitted model. During our analysis, we came up with a linear regression model that was working fine on our training and testing data set, giving us a good reading of R-squared with the lowest residual squared error. To check the efficiency of the model, it was indeed important to run some sampling on various set of our dataset to check if the model was still holding true on those set. The results of the different sampling methods performed are the noted below.

It should be acknowledged that the linear regression model that we came up is the following:

```
lm.fit = lm(CleanData$`Standardized Mortality Ratio` ~
                CleanData$`SHR lower confi limit 5`+
                CleanData$`Nber of Patients included in Survival Summary` +
                CleanData$`% of patients w/ arteriovenous fistulae in place` +
                CleanData$`% of adult HD Patients with Ktv 1.2` +
                CleanData$`Nber of patients included in hospitalization`+
                CleanData$`Standardized Hospitalization Ratio`+
```

CleanData$`Transf Ratio Lower conf Limit 2.5` +

CleanData$`Facilities Transfusion Ratio`+

CleanData$`SRR upper confi limit`+

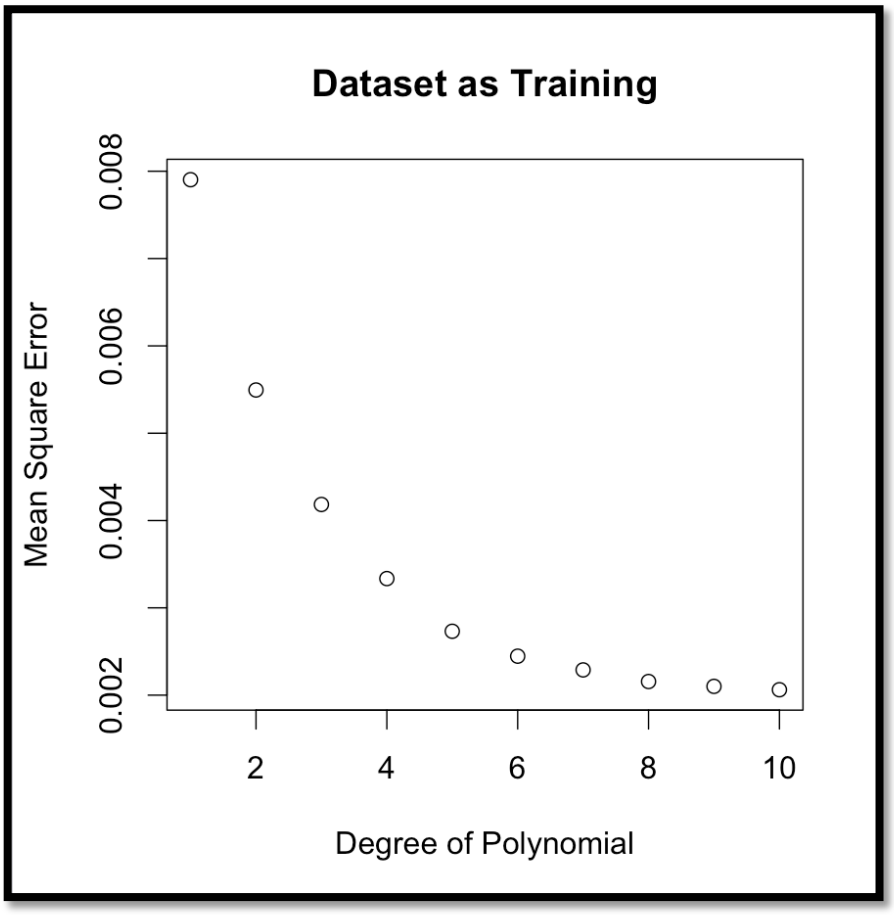
CleanData$`Stand Hospitalization Ratio Lower conf Limit 2.5`+

CleanData$`Nber of dialysis patients with Hgb data`, data = CleanData)

- SHR lower confidence limit 5 = The 5% confidence limit of the Standardized Hospitalization Ratio range
- Nber of Patients included in Survival Summary: Number of survival patients in the facility.
- % of patients w/ arteriovenous fistulae in place = Percentage of patients with arteruovenous fistulae in place
- % of adult HD Patients with Ktv 1.2 = Percentage of patients with hemodialysis ktv1.2
- Nber of patients included in hospitalization = Number of Hospitalized patients
- Standardized Hospitalization Ratio = the standardized hospitalization ratio
- Transf Ratio Lower conf Limit 2.5 = the lower confidence limit of transfusion ratio
- Facilities Transfusion Ratio = The facility's transfusion ratio
- SRR upper confi limit = The upper confidence limit of the standardized readmission rate

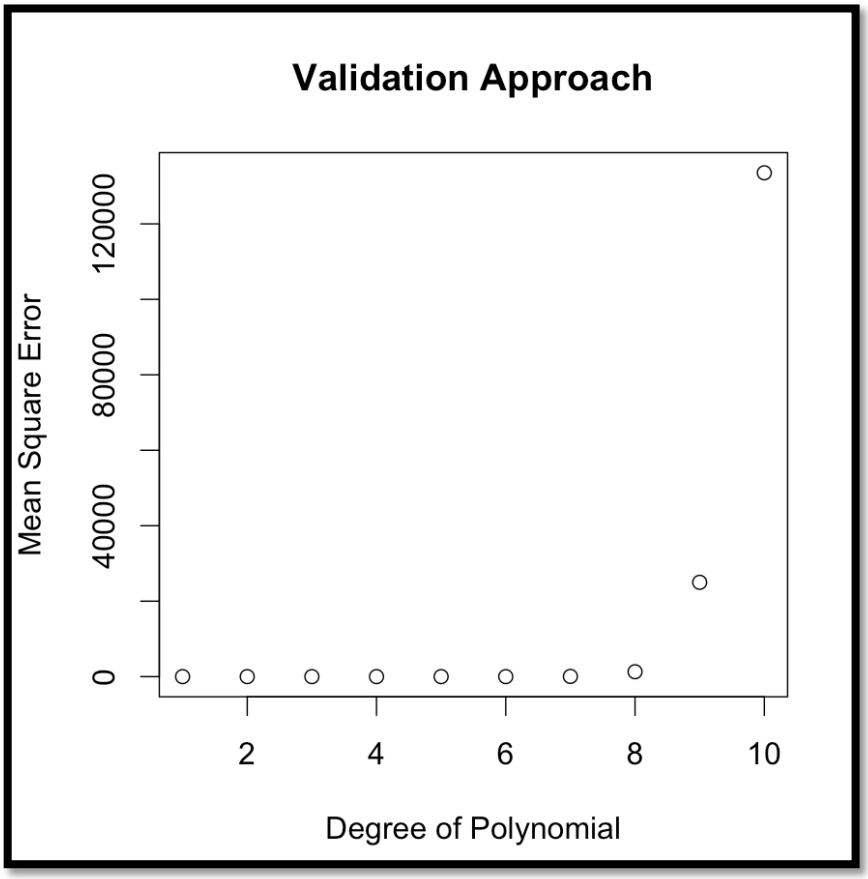
- Stand Hospitalization Ratio Lower conf Limit 2.5 = = the lower confidence limit of the hospitalization ratio
- Nber of dialysis patients with Hgb data = Number of dialysis patients with hemoglobin data.

1. The entire data set as the training data



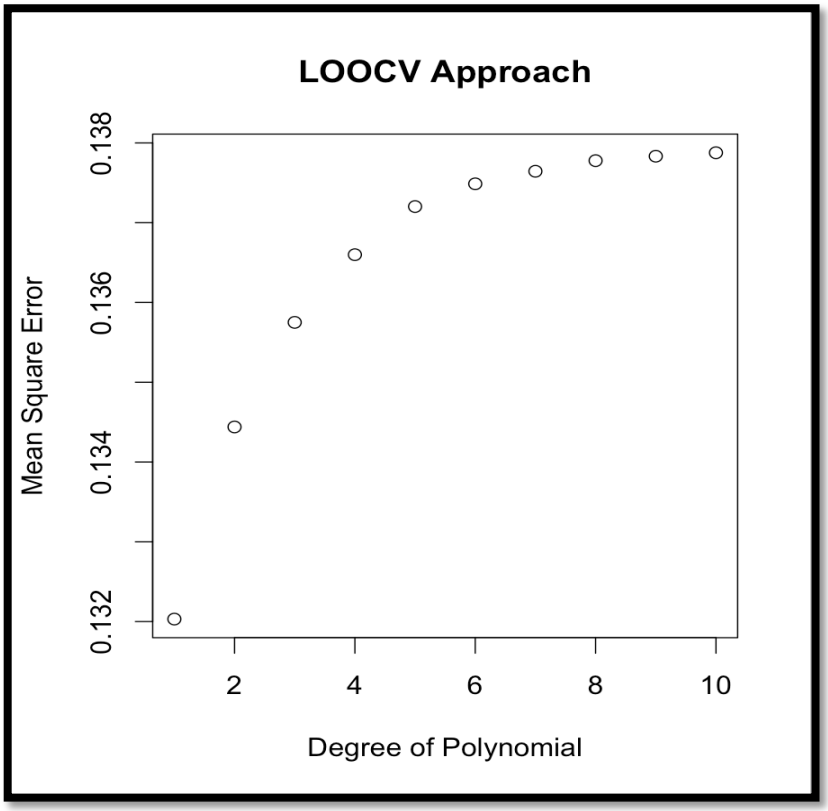
Using the entire dataset as training dataset, the linear regression performed pretty well on polynomial variables. Once we applied polynomial and cubic function, we noted that the mean squared error values were decreasing. This tells us that to have a better reading of our model in the data, we need to use polynomial variables, that means p^10. This statement holds true to the fact that bigger values of predictors are helpful to predict mortality rate given that we are using the whole pool of data at our disposal.

2- The Validation Set Approach



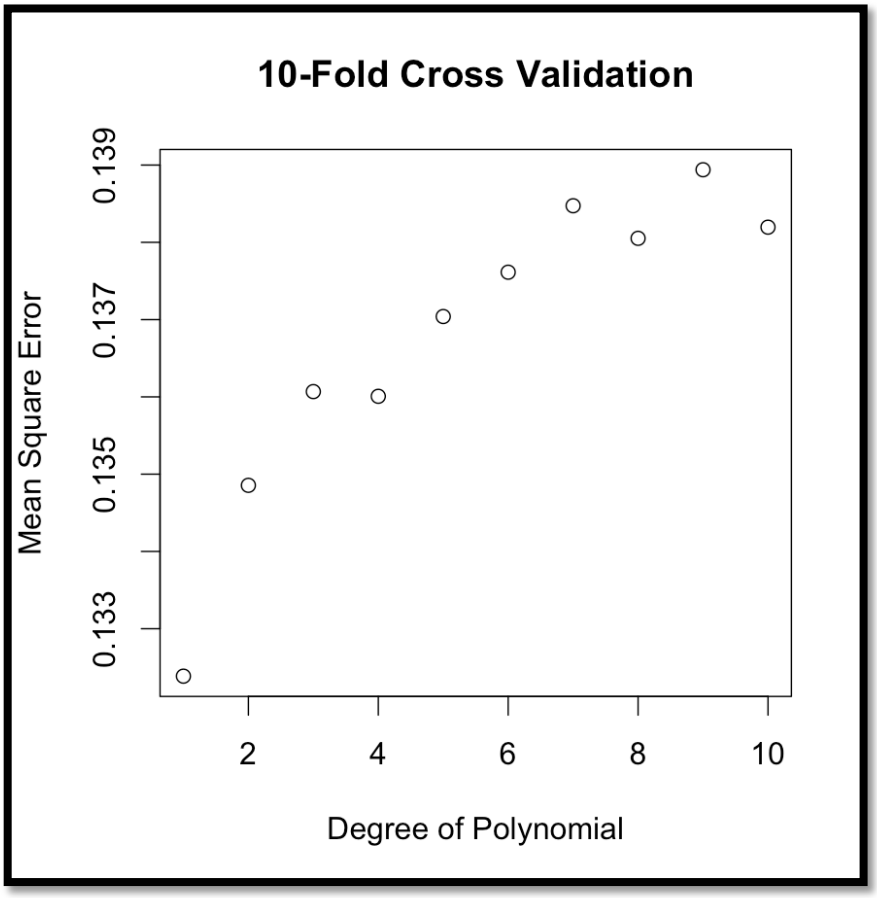
In the validation set approach, we divided the data set into two different sets, namely the training and testing dataset. We used our linear model on the training set and obtained a pretty strong prediction with R-Squared around 88%. We then applied our regression on the testing dataset, using a for loop to increase the degree of polynomial of the parameters. We observed that as the degree of polynomial increases, the mean squared error slowly increases and once we get to a degree of polynomial of 9 the mean squared error rises sharply. From this observation, we can safely attest that if we want to use the validation set approach for our analysis, we should make sure to use a regression with degree of polynomial as 1 through 8. This will give us a better prediction. Using a degree polynomial of 9 or above will give us an undesired result.

3- Leave-one-out cross validation



The Leave-one-out cross validation is a method that use the most resources as it applies the model to each observation of the dataset. Running this method took a very long processing time and 23 obtained quite an impressive result. The LOOCV showed us that using the regression with degree polynomial greater than 1, will be not appropriate. We observe a sharp increase in mean square error when degree polynomial is 2 to 7. The mean square error tends to rise slowly as the degree polynomial is 8 through 9. We also realized that there was not a big gap between the ranges of mean square error as the degree polynomial increases. The mean square error ranges from 0.132 to 0.138. Basically what this method shows us is that we can use any degree of polynomial on our model and we will achieve quite the same prediction.

4- 10-Fold Cross Validation

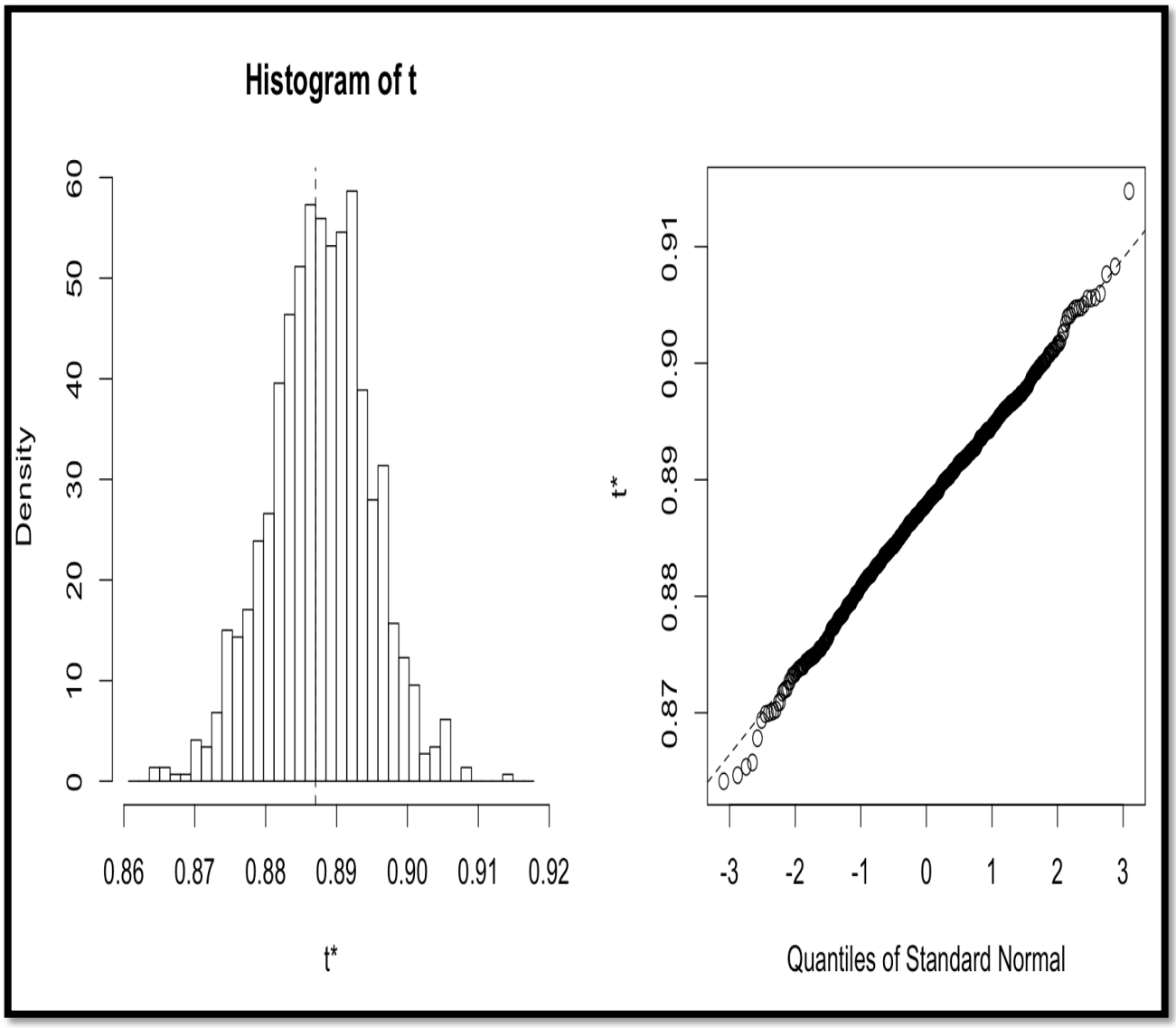


The last method we used was the 10-Fold Cross validation. We divided our dataset into 10 sets of equal sizes and applied our model to the different data set. We saw that we got almost the same value of mean squared error as the LOOCV. Although at some degree of polynomial, the mean squared error was almost the same as the previous degree. The mean squared error ranged from 0.133 to 0.139 and the trend is almost the same as the LOOCV. If we want to use the 10-fold cross validation for our analysis, we can use any degree of polynomial and the prediction will almost be the same.

II- Bootstrap

The bootstrap approach can be used to assess the variability if the coefficient estimates and predictions from a statistical learning method. In our analysis we wanted to quantify the parameter uncertainty. One way of doing this was to assess the r-squared obtained from our models to tell us how well the parameters chosen are related to our prediction. If the value of R-squared was less than 0.5 we could confirm that the predictors chosen do not really relate to our response variable. An r-squared value of 0.7 and above tells us that the parameters are strongly related to our response variable. We created a function that would compute the r-squared statistic; then used the boot() function to perform the bootstrap repeatedly sampling observations from the data set with replacement. The bootstrapping methods gave us a R-squared of 0.88 on a sample of 1000 observations with a standard deviation of 0.007095114 and a bias of 0.0006725087. Applying this technique confirmed that our linear model is quite strong at predicting the standard mortality rate of any given facility given the some parameters.

We then plotted the result of the bootstrap to see an histogram of the r-squared returned by our observations and we noted that the mean r-squared was around 0.89 like can be seen below.


Histogram of t
Density
0
10
20
30
40
50
60
0.86
0.87
0.88
0.89
0.90
0.91
0.92
t*
t*
0.87
0.88
0.89
0.90
0.91
-3
-2
-1
0
1
2
3
Quantiles of Standard Normal