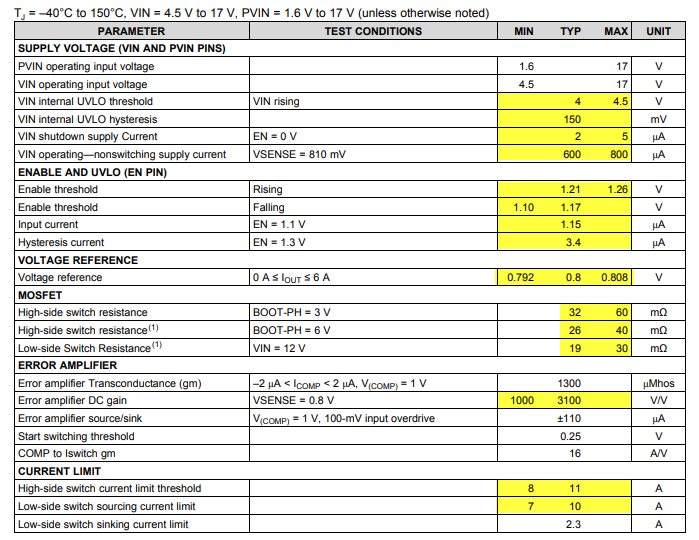
**Predicting Performance of Integrated Circuits using Regression Analysis**

**Joanna Duran, Nikhil Gupta, Max Moro**

**Introduction**

Semiconductor manufacturing is a variable process and outcomes depend on several factors. In order to meet target specifications, some parameters are controlled by engineers. However, some parameters are beyond human control (e.g. process variation). The output variables are typically measured after the manufacturing process is complete and all R&D has been performed. There is a set list of specified values that are acceptable, usually falling between a min and max range. Variation in the process leads issues if the outputs are outside target specifications. Due to time, resource and cost constraints not all values can be practically measured.

Below is an example specification sheet (Fig 1). Each column is a single output and their respective values but not all values are populated because they cannot be practically measured.

*Fig 1*

Like in most manufacturing environments, the question is “Can we predict the limits before the integrated circuits are manufactured to preemptively make changes when specs are expected to be out of range?” The answer is yes. Current practice is to use electrical simulation (+ running Monte Carlo simulations). However, this is very resource and time intensive as each electrical simulation can take several hours. Our objective is to build a model to predict the performance (min, typ, max) of an output variable.

Data has been sponsored and approved for use by Texas Instruments Inc who provided two files. Due to proprietary information, we cannot disclose the variables measured or the department. The data consists of 10,000 rows capturing the performance of the circuit under various conditions. There are 240 features consisting of:

* Engineer-Controlled variables (x1 – x23). Values differ, some are between 1 to 100 while others are in Nano or micro range.
* Process variation variables ( stat1 – stat217). These parameters are beyond human control. They represent various statistical parameters whose values represent the sigma variation around the mean. Range is from -4 (sigma) to 4 (sigma)
* Output Variables (y1 - y19) which represent various output variables

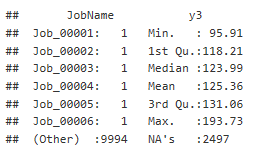
The engineer-controlled variables were uniformly randomly sampled from their design space (since engineers can select any values from range). [NG to add details] Statistical features were also uniformly randomly sampled since we wanted accuracy of model to be good throughout statistical variation range and not just closer to the population means.

The goal is to pick at least 1 output variable and build a model to predict the mean value and the statistical variation of the output with respect to the process. Target accuracy of +/- 10% is desired, but +/- 15% would be acceptable.

**Data Preparation** **and Exploratory Data Analysis**

After discussion with subject matter experts, it was decided to focus on Y3 as it is the most critical output of the process.

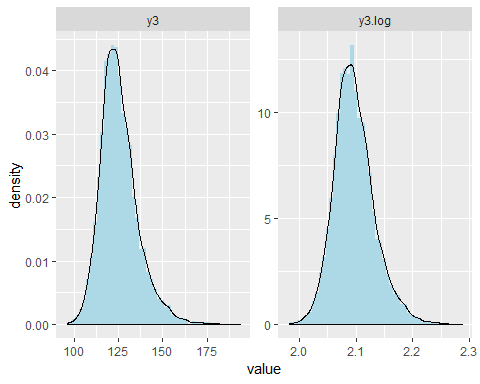
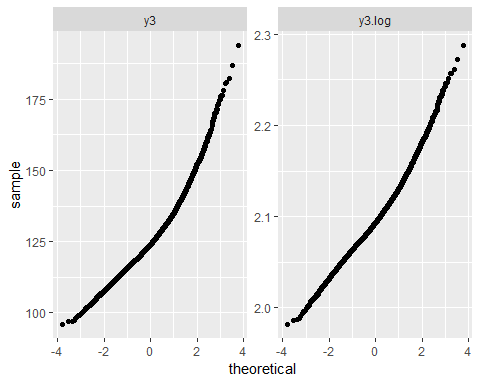
Like most data, it needed cleaning. We began by checking correlations to evaluate removal of redundant features. Basic descriptive statistics revealed how many NA values were present (Fig 2). [Needs to be updated - Max]



*Fig 2*

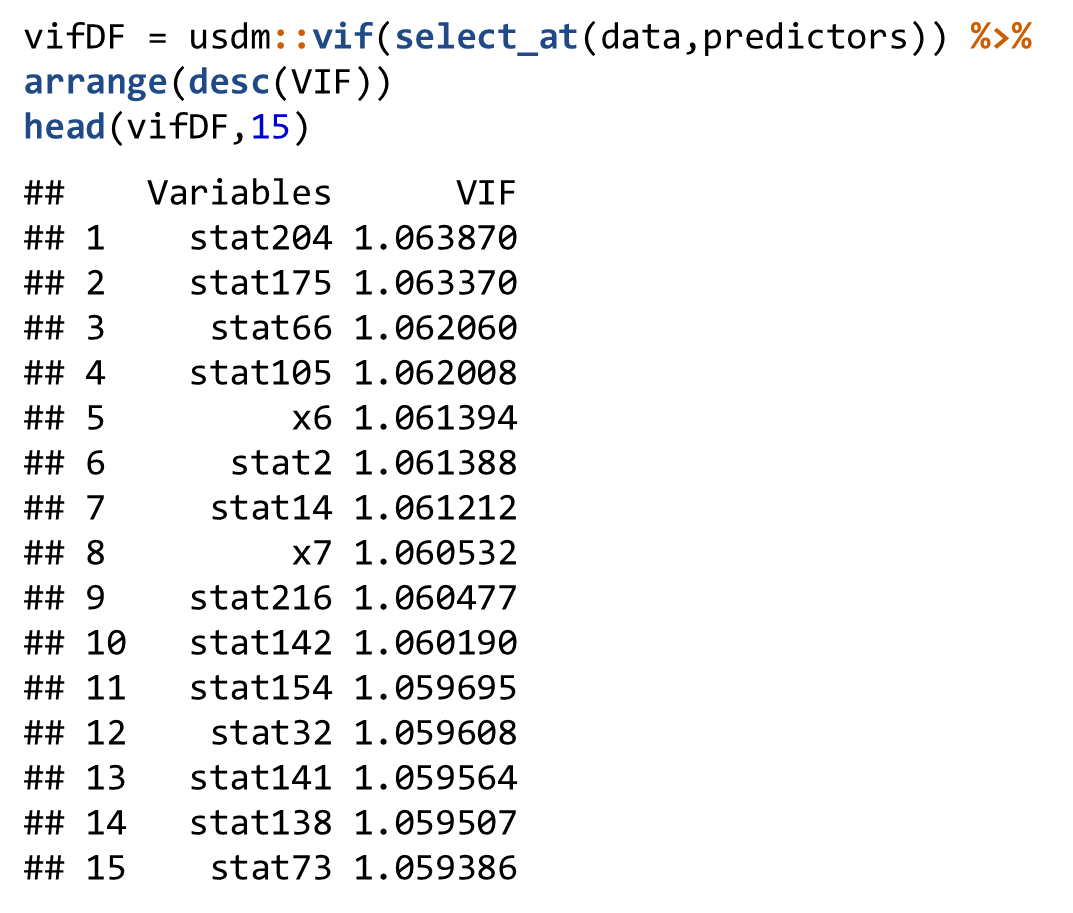
We cleaned up labels and removed NA values. Once the data was cleaned, the datasets were merged.

We begin exploratory data analysis on the target variable y3 and notice right skewness, therefore we perform a log transformation. Log transformation helps a bit, therefore we proceeded with the log transformed variable “y3.log” (Fig 3). It is important to note that base 10 is more common in this industry, which is why it was used instead of natural log.

**

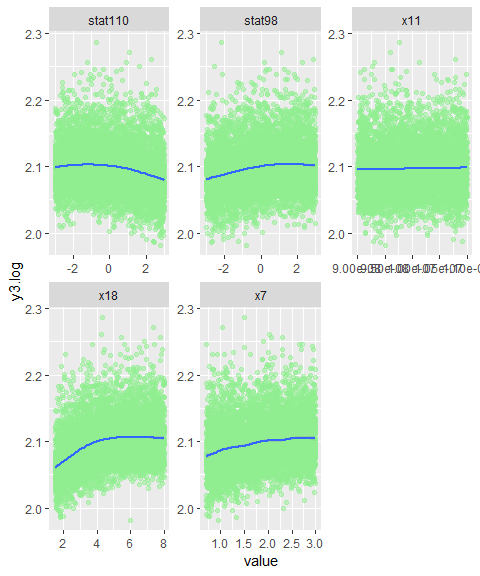
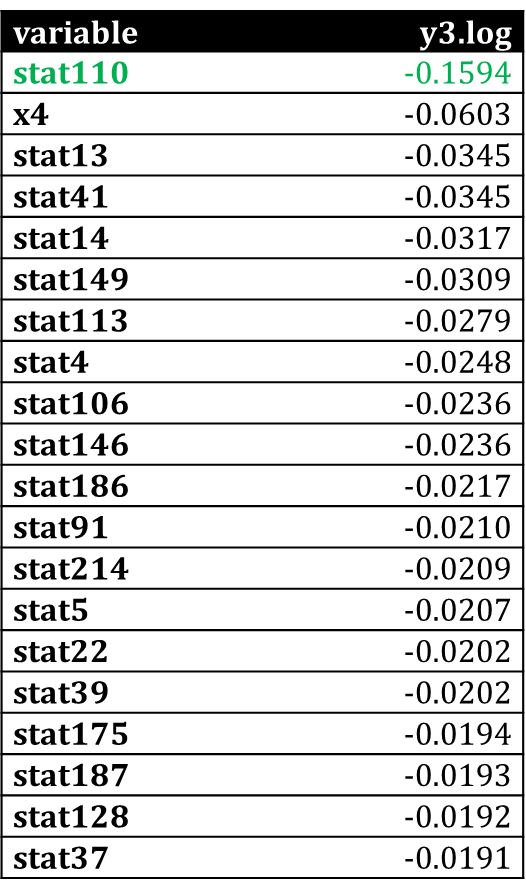
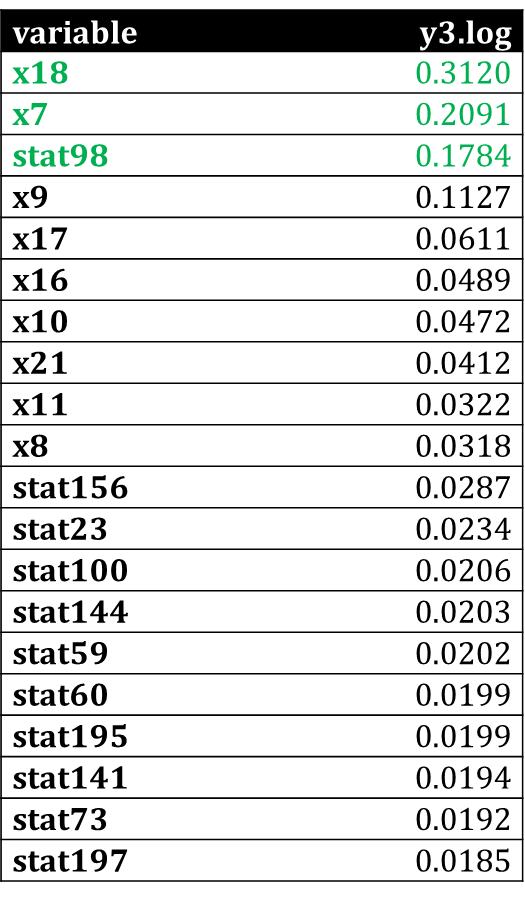
*Fig 3*

In order to determine if there is a correlation within features, we begin by checking for multicollinearity. Since inputs were randomly selected, we did not expect there to be multicollinearity. After running the analysis, the correlation table and VIF values (Fig 4) confirmed that there is no multicollinearity.



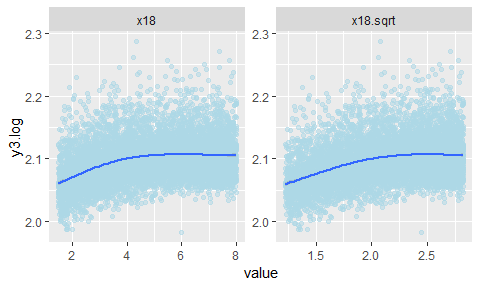
*Fig 4*

We also want to determine if there is a correlation of features to output “y3.log”. The output and the scatterplots (Fig 5) show that few features are highly correlated (positive or negative) with the y3.log.We deduce that features by themselves may not have a lot of predictive power.



*Fig 5*

It is noticed that one of the most correlated features, x18, shows a slight curvature in the scatter plot therefore a square root transformation is performed. From the transformation, it is noticed that there is only a minor improvement to linearity (Fig 6).



*Fig 6*

Based on our observations we proceed to modeling with the target variable y3 as log transformed and the predictor x18 transformed with square root.

When keeping in mind good modeling practices we want to make sure to avoid overfitting. Even though we have a lot of data points and many features, it is still possible to overfit the data. In order to avoid overfitting, we used an 80:20 ratio split on our dataset to develop the Train and Test sets. During the training process, a 10-fold cross validation technique was used for model selection. This allows for the use of the entire training dataset for model development.

**Full Model Analysis**

The full model analysis was performed by analyzing the full model consisting of all 240 features with appropriate transformations. The fit statistics show that only a handful of predictors were statistically significant which was expected based on the correlation of features to output analyzed in the exploratory analysis. It was determined that variable selection may be appropriate at this step to remove unnecessary predictor variables. The adjusted 𝑅^2 is 0.2275 (Fig 7) which is fairly low and expected based on low correlation of features to outputs. We wondered if two-way interactions would improve predictions but considering that this would increase our predictors from 239 to 28441, it is not practical and outside the scope of this project. In the future, we may need to do intelligent (selective) feature engineering using domain expertise to improve prediction capability.

We also analyzed the plots for issues with model assumption. For independence, the features were selected randomly therefore independence assumption has been met. For normality, histogram and QQ plot of residuals (Fig 8) shows that they are not normally distributed. The residual plot (Fig 9) also point to non-normality since there are several values beyond the 2-sigma line. However, given that we are working with such a large sample size, the concern is minimized. For equal variance, the residual plot (Fig 10) shows that the points do not exhibit equal variance at all predicted levels. The points at the lower predicted levels have much smaller variance compared to points t the higher predicted levels.

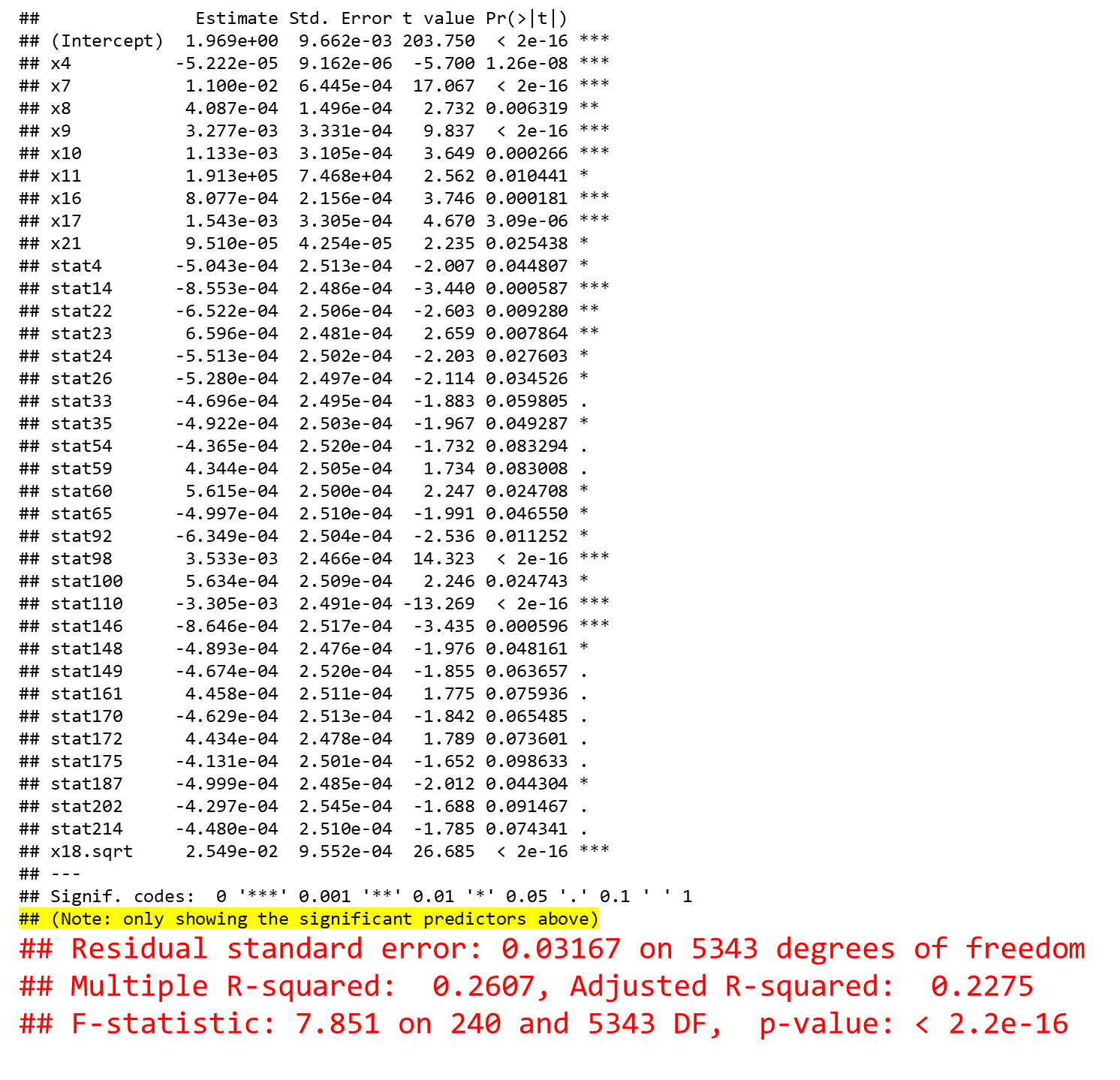
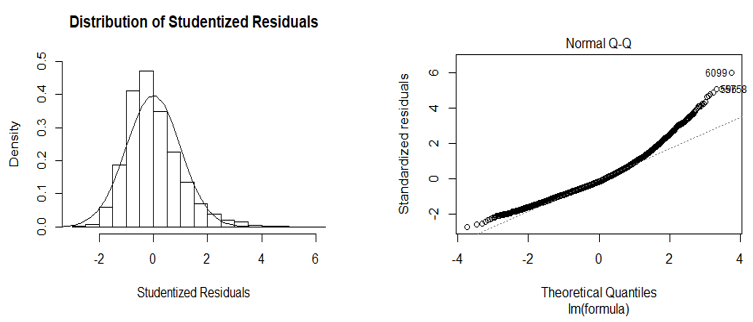
 

Fig 7 (Add text instead of image) Fig 8

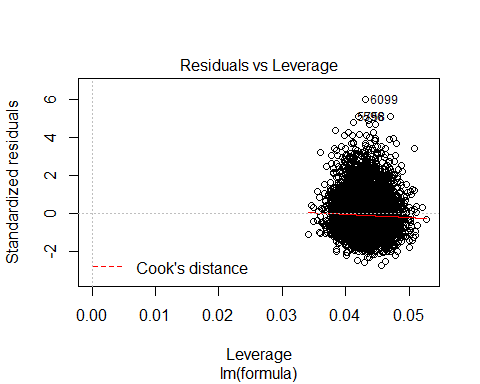
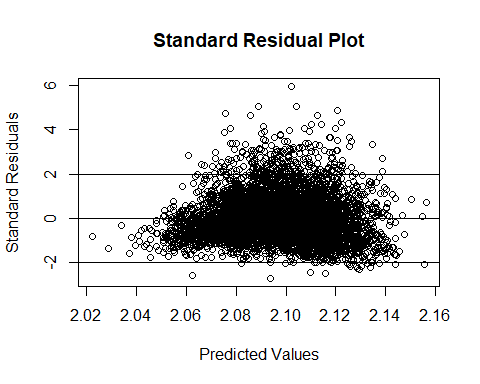


Fig 9 Fig 10

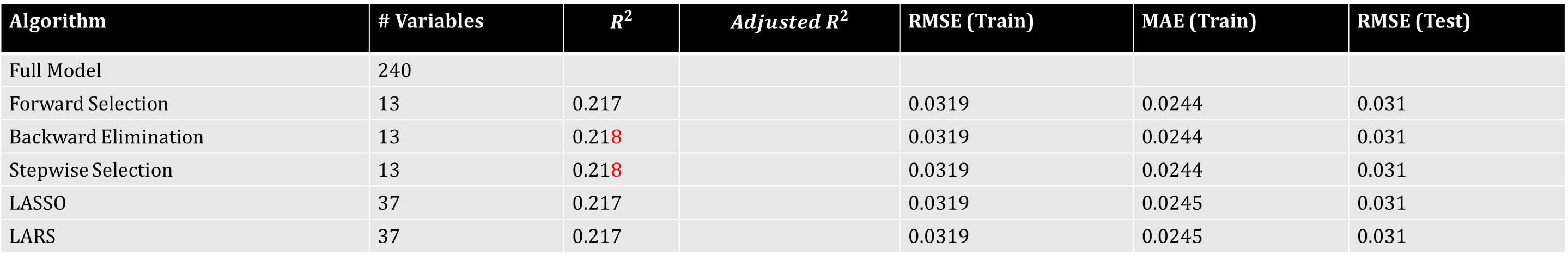
The influence statistics are also analyzed and from the Cook’s D plot we notice that there are 288 points beyond the 4/n line. This is a large amount of points and it is impractical to analyze all of them manually. These points are influential and can impact the model coefficients adversely. It is tempting to remove these points because it would improve the model fit drastically. But after consulting with domain experts, we found that these are valid points and there is no justification to remove them. We will proceed with these points included, noting that the assumptions for linear regression have not been satisfied entirely.

**Variable Selection Process**

In selecting variables, we know there are many techniques that provide many different results. The variable selection techniques that were used were:

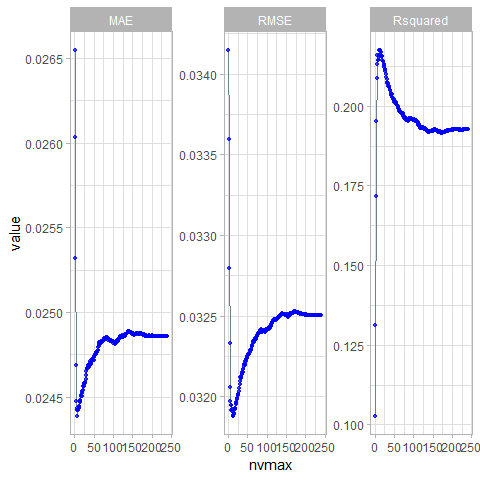
* 1. Forward Selection
  2. Backward Elimination
  3. Stepwise Selection
  4. LASSO
  5. LARS

We found that all techniques give essentially the same fit statistics. Forward Selection, Backward Elimination and Stepwise Selection have only 13 predictors while LASSO and LARS have 37 predictors in the final model (Fig 11).



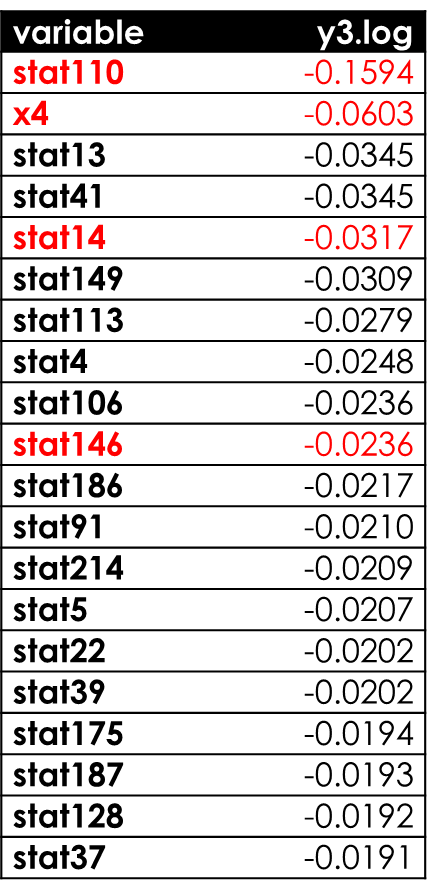
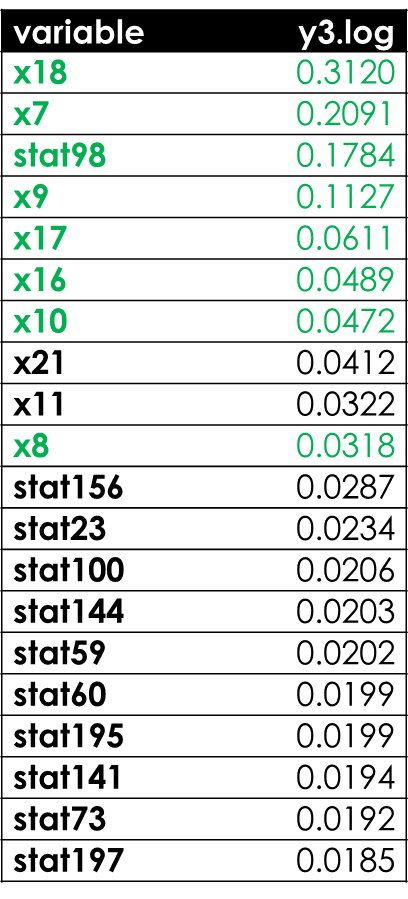
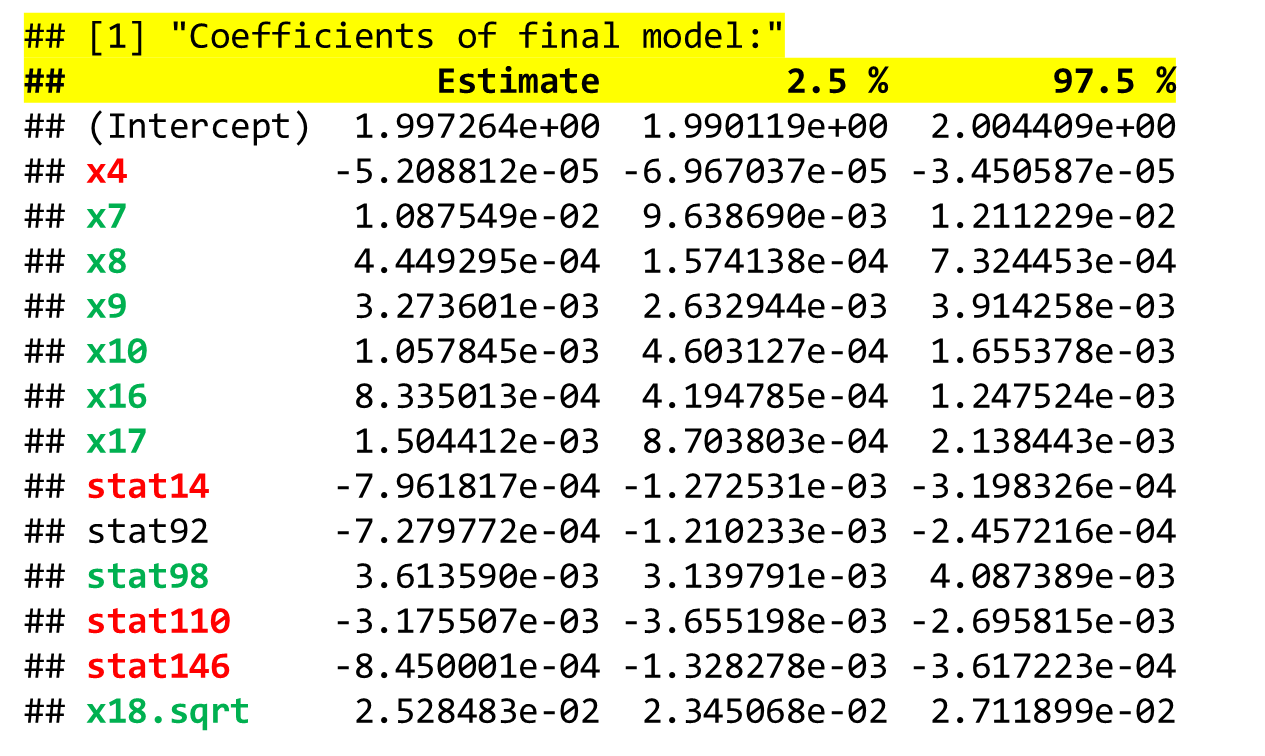
*Fig 11 (Update from PPT)*

The best model is Backward Elimination. It was chosen with one of the reasons being that it has the least number of predictors. The RMSE was also used to choose the final model. The lowest value of RMSE obtained for the model with 13 variables. The same for MAE and as well (Fig 12).



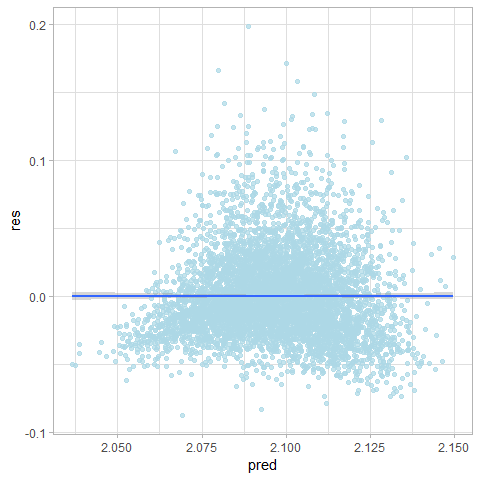
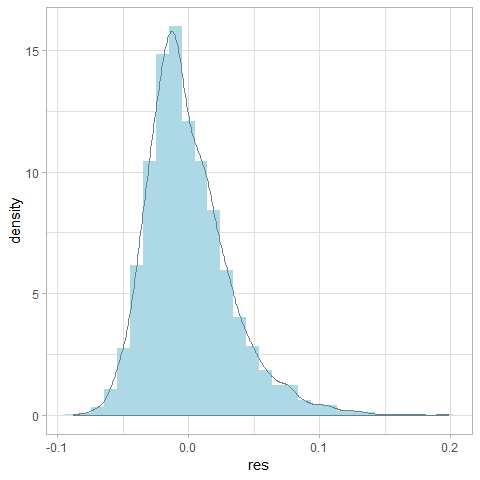
*Fig 12*

The final model includes some of the same variables that we identified to be highly correlated to output in the exploratory analysis (Fig 13).



*Fig 13*

In analyzing the residuals, we noticed the histogram of the residuals (Fig 14) shows similar right skewed distribution as was seen in the full model. In the residual scatterplot we noticed the equality of variance at lower values of prediction are questionable as before (Fig 15). We conclude that the final model suffers from same assumption violations as the full model.



*Fig 14 Fig 15*

Given the fact that the features were sampled randomly, any predictions drawn from this model can be applied to the entire design and manufacturing space of the integrated circuit.Causality is not a concern here since the goal is mainly prediction of performance using the model. However, given the fact that some of the model assumptions for multiple linear regression have not been met, we should be careful in using this model to predict new values.

**Conclusion**

In order to improve the model, we recommend that Intelligent feature engineering be used to improve prediction performance. Also, the use non-parametric models that do not need to satisfy normality and equal variance assumptions. Divide data into clusters and model each cluster separately. (show box plot of high vs. low Cook’s D points). Elaborate more on this.

Appendix

Refer to github <link>