Kerosene Merox Treater No. 2 (KMX-2) Salt Filter (D-4404) Salt Depletion Rate Modeling via Stepwise Linear Regression (SPLR)

Pascual, Ronald Patrick D.[1,2], Arellano, Roman Christopher T.[1,3], Pumatong, Jurrel D.[1,4]
[1]Process Engineering Department, Technical Services Division, Production A, Petron Bataan Refinery, Limay, Bataan, Philippines
[2]rdpascual@petron.com, [3]rtarellano@petron.com, [4]jdpumatong@petron.com

ABSTRACT

This study primarily aims to generate a linear model that can predict the salt depletion rate of Kerosene Merox Treater No. 2 (KMX-2) Salt Filter (D-4404). Operational data of KMX-2 were gathered, cleaned, standardized and transformed before proceeding with Stepwise Linear Regression (SPLR) technique. All of the steps involved were performed using R and the Comprehensive R Archive Network (CRAN). The final generated model, $\ln(DepRate) = 2.437 + 0.407KerFeed - 0.021ReactordP + 0.073ClayOP - 0.907(KerFeed \times ClayOP) + 0.035(ReactordP \times ClayOP) - 0.030(KerFeed \times ReactordP)$, led to an R^2 of 0.9932 with 9.67% average percent error. Future studies may include other techniques involving multicollinearity such as Principal Component Analysis to gather more meaningful insights and to further simplify the model.

I. Introduction

Kerosene Merox Treater No.2 (KMX-2) aims to chemically treat kerosene to convert sulfur present as mercaptans to a less objectionable sulfur form which are disulfides. Part of the purification process is passing the treated kerosene to a fixed bed of salt to remove entrained water.

The level of salt in the salt filter (D-4404) shall be maintained at a critical level of 40%. Hence, salt depletion rate prediction is vital not only in stable operation, but also in planning shutdown durations and salt topping activity requirements such as salt inventory and manpower.

Historically, salt depletion is based on typical or averaged values only, resulting to large deviations from the actual value. Thus, this study aims to explore stepwise linear regression technique in predicting salt depletion to eliminate this operational uncertainty.

The study is delimited to evenly distributed depletion rate per timeframe. Moreover, the study only focuses on stepwise linear regression. Other linear, non-linear and machine learning techniques are not covered.

II. Theoretical Framework

Multiple Linear Regression Model

The basic linear model assumes that there exists a linear relationship between two variables X and Y. This relationship is not perfect and distributed by some random error. For each value of X, say x, the corresponding y-value is of the form:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

where β_0 and β_1 are called as the intercept parameter and the slope parameter, respectively. If there are observed n values of x (i.e., $x_i, i = 1,2,\cdots,n$) with errors ε_i, then the resulting random variables $y_1, y_2, \cdots y_n$ will be defined as:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \qquad i = 1,2,\cdots,n.$$

Similarly, the multiple linear regression model relates a response variable to p predictor variables. The model can be written as:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

where,
$y_i = i^{th}$ value of the response
$x_{ij} = i^{th}$ value of the j^{th} predictor variable
β_j = regression coefficient of the j^{th} predictor
ε_i = random error term, $i = 1,2,\cdots,n;\ j = 1,2,\cdots,p$.

Simple Linear Regression Model (SLRM), as discussed above, is prevalently misused due to failure of testing for model adequacy. That is, simple linear regression models can only be used when mathematical assumptions are met. Classical assumptions are enumerated and discussed briefly below:

1. *Linearity of the model*: The regression model is linear in the parameters. It is not necessary that Y and X are linearly related.

2. *Exact Measurement of the Covariates*: The covariate/s (predictor variable/s) must be recorded without measurement error. Otherwise, the interpretation of the error term will include not only the effect of unspecified predictor variables but also the systematic errors incurred in measuring X.

3. *Correct Specification of the Regression Model*: The regression model is correctly specified. This consists of developing the appropriate functional form of the model and selecting which variables to include.

4. *Zero Mean*: This critical assumption states that the mean of the error terms, ε_i, is equal to zero. This means that the influence of other factors not included in the model is essentially random.
$$E(\varepsilon_i) = 0$$

5. *Homoscedasticity or Constant Variance*: The variance of the error terms, ε_i, must be constant in relation to zero mean assumption.
$$Var(\varepsilon_i) = \sigma^2$$

6. *No Autocorrelation*: The error terms must be not correlated.
$$Cov(\varepsilon_i, \varepsilon_j) = 0, \forall i \neq j$$

7. *No Multicollinearity*: Multicollinearity between two or more covariates (predictor variables) means that the one should not be a linear function of another.

8. *Normality of Error Terms*: It is assumed that the error terms follow a normal distribution with zero mean and constant variance.
$$\varepsilon_i \sim N(0, \sigma^2)$$

OLS and Gauss-Markov Theorem

The goal of linear regression is to obtain estimators , $\widehat{\beta}_i, i = 0,1,\cdots,p$ for the parameters: β_i, $i = 0,1,\cdots,p$. One way to do this is through *Ordinary Least Squares* (OLS) Method. This method draws a line through the data points that minimizes the sum of the squared differences between the observed values and the corresponding fitted values. Simply, OLS minimizes the sum of squared residuals (errors).

The properties of OLS proves that the resulting estimators are unbiased linear estimators. With this, the next question is whether or not these estimators satisfy a certain criterion of "optimality". This is where Gauss-Markov Theorem play its role.

The *Gauss-Markov Theorem* states that in a linear regression, with error terms having *zero mean, constant variance and uncorrelated* (classical assumptions 4, 5 and 6), the OLS estimators are the Best Linear Unbiased Estimators (BLUE). They are "best" in the sense that among the class of unbiased linear estimators, OLS estimators give the lowest variance.

Contrary to classical assumptions, the errors do not need to be normal, nor do they need to be independent and identically distributed. However, the requirement that the estimator be unbiased

cannot be dropped, since biased estimators with lower variance like James-Stein estimator exists.

Quantitative Data Transformations

Real data (complex with multiple predictors) often fail to meet normality and homoscedasticity criteria of simple linear regression model. To augment this, data analysts transform either the response variable, the predictor variable/s or both. Note that there is nothing illicit in transforming variables, however, the analysis of the results with transformed variables must be done cautiously.

There are several ways to conduct variance stabilizing transformations (to conform to $\varepsilon_i \sim N(0, \sigma^2)$ and consequently $Var(\varepsilon_i) = \sigma^2$) as summarized below.

Generally, for *right-skewed data* (tail is on the right or positively skewed), the appropriate transformations include $\sqrt{y}$, $\sqrt[3]{y}$, and $\ln(y)$. Conversely, for *left-skewed data* (tail is on the left or negatively skewed), the appropriate transformations include, y^2, $\sqrt{a-y}$, $\sqrt[3]{a-y}$, and $\ln(a-y)$ where a is an arbitrary constant.

1. Logarithmic Transformation

Holding all other factors constant, summarized below are the changes in interpretation once log transformation is employed.

Case	Regression Specification	Interpretation of β_1
linear-log	$Y = \beta_0 + \beta_1 \ln x + \varepsilon$	a percent change in x corresponds to $\ln(1.01) * \beta_1$ or $0.01\beta_1$ unit change in Y
log-linear	$\ln Y = \beta_0 + \beta_1 x + \varepsilon$	a unit change in x corresponds to $100 * (e^{\beta_1} - 1)$ or $100\beta_1$ percent change in Y
log-log	$\ln Y = \beta_0 + \beta_1 \ln x + \varepsilon$	a percent change in x corresponds to $100*(1.01^{\beta_1} - 1)$ percentage increase in Y

Note that because ln(0) is undefined, as is ln of any negative number, when using a log transformation, a constant should be added to all values to make them all positive before transformation.

2. Box-Cox Transformation

Box-Cox method considers a family of transformations on strictly positive response variables where the parameter, λ, is chosen by numerically maximizing the log-likelihood.

$$g_\lambda(y) = \begin{cases} \dfrac{y^\lambda - 1}{\lambda} & \lambda \neq 0 \\ \log(y) & \lambda = 0 \end{cases}$$

Note that when λ approaches 1, there is essential no transformation. This does not change the variance nor make the model fit better. Otherwise, when λ approaches 0, the appropriate transformation is logarithmic.

3. Tukey's Ladder of Power Transformation

Unlike Box-Cox where the response variable is transformed to $\frac{y^\lambda - 1}{\lambda}$, Tukey's Ladder of Power, with the form y^λ, iteratively finds a λ that maximizes the W-statistic. In essence, this finds the power transformation that makes the data fit the normal distribution as closely as possible.

Qualitative Data Transformations

Dummy variables are used to allow the use of categorical (non-numerical) variables as predictor variables in regression models. There are many instances where categorical variables are used as predictors, here are some:

a. The variable has no intrinsic quantitative characteristic (e.g. pass/fail, sex etc.)

b. The categorical responses are not measured numerically but instead by category (e.g. education attainment: elementary, undergraduate etc.)

c. The dummy variables are used as time identifiers for group data

A dummy variable, denoted by D is a dichotomous variable defined as:

$$D = \begin{cases} 1, & belongs\ to \\ 0, & o\ therwise \end{cases}$$

In general, if the variable has m categories, there should be $m-1$ dummy variables created. An example is shown below.

$$Categorical\ Variable = \begin{cases} a \\ b \\ c \end{cases}$$

$$D_1 = \begin{cases} 1, & a \\ 0, & otherwise \end{cases} \qquad D_2 = \begin{cases} 1, & b \\ 0, & otherwise \end{cases}$$

	Dummy Variable	
Category	D_1	D_2
a	1	0
b	0	1
c	0	0

Note that setting a third dummy variable, D_3, where c is assigned with 1 will result to a multicollinearity issue ($D_3 = 1 - D_1 - D_2$). Hence it should be omitted.

Standardization

Standardization is the process of putting different variables on the same scale. This is primarily rooted in the problematic effects of variation in magnitude of predictors. For instance, if two variables with ranges of 0 to 10 and $1e10^5$ to $1e10^6$ are used in linear modeling, slight changes in the latter will cause significant changes in the predicted response.

One way to scale data is via Z-score standardization where data are re-scaled so that the mean is 0 and the standard deviation is 1. Z-scores are computed by subtracting the mean, μ, and then dividing the standard deviation, σ, of the data to the observations.

$$Z_i = \frac{x_i - \mu}{\sigma}, \qquad i = 1,2,\cdots,n$$

It is important to make sure that the μ and σ used to standardize shall be the same for both the training and test set.

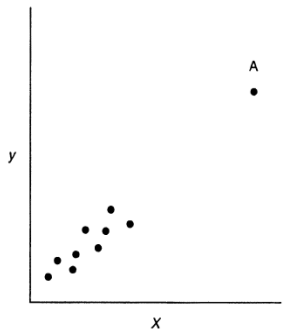
Outlier, Leverage and Measures of Influence

An *outlier* is a data point whose response, y, does not follow the general trend of the rest of the data. Conversely, a data point is a *leverage point* if it has extreme predictor x values.

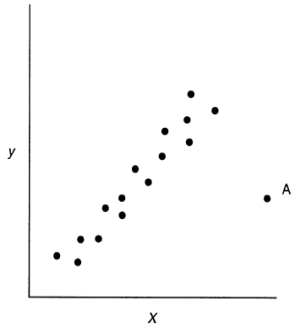
To discuss this further, note that leverage statistic measures the distance of an observation from the center of the x-space. With multiple predictors, extreme x values may be particularly high or low for one or more predictors, or may be "unusual" combinations of predictor values (e.g., with two predictors that are positively correlated, an unusual combination of predictor values might be a high value of one predictor paired with a low value of the other predictor).

A data point is *influential* if it unduly influences any part of a regression analysis, such as the predicted responses, the estimated slope coefficients, or the hypothesis test results. *Outliers and leverage points (high leverage statistic)* have the potential to be influential. However, generally, an investigation shall be pursued to determine whether or not they are actually influential.

There are no general rules in choosing whether an influential data point or sets of influential data points is/are to be omitted. Hence, analysts often compare models with or without the influential data points in terms of R^2, predictor coefficients, intercept, and overall accuracy of prediction of the developed models. Shown below is an example of influential data point.



unusual x value
very little effect on regression coefficients



unusual x and y value
influential on regression coefficients

Leverage statistic, h_i, is a standardized measurement of the distance of the i^{th} observation from the center of x-space, $\bar{x}$. If a given observation has a leverage statistic that exceeds $\frac{2*(k+1)}{n}$, then that point is considered to be a leverage point. Note that k is the number of predictor variables and n is the number of data points.

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Outliers are detected by studentized residuals, S_i. Observations whose studentized residuals are greater than 3 in absolute value are possible outliers.

$$S_i = \frac{\varepsilon_i}{\hat{\sigma}\sqrt{1 - h_i}}$$

Multicollinearity

Multicollinearity in linear regression means that some of the predictors are linear factors of other fellow predictors. This phenomenon disrupts the precision of estimate coefficients, that is, even if the p-values are less than $\propto$ (which means that the predictor is statistically significant), the model produced will not be accurate. This poses difficulty in deciding which variables to include in the model.

Variance Inflation Factor (VIF) helps identify correlation between independent variables. Generally, VIFs larger than 10 are considered significant.

$$VIF_j = (1 - R_j^2)^{-1}$$

R_j^2 is the coefficient of determination of the regression model when the predictor j is predicted from all other predictors.

Structural multicollinearity (caused by the model and interaction of chosen predictors) is often fixed through standardization. However, if VIF is still greater than 10 after standardization, the multicollinearity is therefore caused by the data itself and addition of interaction/s or omission of involved predictor variables will fix the problem.

Interactions

Inclusion of Interaction or Effect Modification in linear regression is done to take into account the multicollinearity of predictor variables. There are two questions to ask in deciding whether interaction terms are to be included in a linear model, namely: (1) Does the interaction make sense concepctually, or in other words, does it make sense that the effect of predictor variable 1 should depend on predictor variable 2?; and (2) Is the interaction term statistically significant, or whether or not the slopes of the regression lines are significantly different upon its inclusion?

Like influential data points, inclusion of interaction terms may or may not significantly affect the accuracy of the model. Hence, generated models shall be compared.

Dataset Partitioning and Cross Validation

In a dataset, a *training set* is used to build up a model, while a *test (validation) set* is used to validate the model built. There are no fixed partitioning rules of training and test set, however, several references prescribes that the training set should be at least 60% of the total dataset. More so, the important thing to consider is how well the training set represents the entire dataset of interest. If the number of points of the whole dataset is large, then, any division may work fine. However, when the data is limited, the training-test division may play a crucial role.

Stepwise Linear Regression

Stepwise Linear Regression (SPLR) is a linear modeling technique that iteratively compares combinations of predictor variables in terms of Akaike Information Criterion (AIC) to come up with the optimum model.

AIC is a single number score that makes use of a model's maximum log-likelihood estimation as measure of fit. Log-likelihood, l, is a measure of accuracy of predicted values. The model with the maximum likelihood is the one that "fits" the data the best.

$$AIC = -2\ln(l) + 2k$$

AIC is low for models with high log-likelihoods, but adds a penalty term for models with high number of predictors, k, since more predictors means a model is more likely to overfit.

As a model selection tool, AIC can only provide a relative test of model quality. That is to say that AIC does not and cannot provide a test of a model that results in information about the quality of the model in an absolute sense. So if each of the tested statistical models are equally unsatisfactory or ill-fit for the data, AIC would not provide any indication of this.

Performance Metrics

The null hypothesis of linear modeling, H_0, states that there is no relationship between the predictors and the response variable. An opposite statement given by the alternative hypothesis, H_a, states that there exist some relationship. To evaluate this, the following fit measures will be used. Each of which are simplified and discussed thoroughly below.

1. *Coefficient t-value* – is a measure of how many standard deviations the estimated coefficient's t-score is far away from 0. To reject the null hypothesis, the t-value must be far away from 0.
2. *Coefficient p-value* – $Pr(>|t|)$ is the probability of obtaining results as extreme as the observed results when H_0 is true. Conversely, the term significance level, α, is used to refer to a pre-chosen probability of rejecting the null hypothesis when it's true. Lower significance levels indicate that a stronger evidence is required to reject the null hypothesis.

   An α of 0.05 indicates that there is a 5% risk of concluding that a relationship exists when there is none. Hence, if the p-value is less than the significance level, the null hypothesis can be rejected and the relationship is statistically significant.

3. *Residual Standard Error (RSE)* – is the average amount that the response will deviate from the true regression line.

$$RSE = \sqrt{\frac{1}{DOF}\sum_{i=1}^{n}(y_i - y)^2}$$

4. *Coefficient of Determination* – R^2 shows the amount of variability in the response variable that can be explained by the predictors. Value ranges from 0 to 1, where values nearer 1 means better fit. Increase in the number predictor variables increases R^2 – making it unreliable. Hence, R^2_{adj} was introduced to take into account the number of variables.

$$R^2 = \frac{SSM}{SST} = 1 - \frac{SSR}{SST} = 1 - \frac{(y_i - y)^2}{(y_i - \overline{Y})^2}$$

5. *F-statistic* – is a good indicator of whether there is a relationship between a specific predictor and the response variables. The further the F-statistic from 1, the better it is. Generally, when the number of data points is large, an F-statistic that is only a little bit larger than 1 is already sufficient to reject the null hypothesis. The reverse is true as if the number of data points is small, a large F-statistic is required to be able to

ascertain that there may be a relationship between the variables.

$$F = \frac{MSM}{MSR} = \frac{SSM/_{DOF}}{SSR/_{n-p-DOF}}$$

6. *Mean Absolute Error (MAE)* – measures the average magnitude of errors (difference between predicted values, y_i, and the actual observation, y) in a set of n predictions, without considering their direction.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - y|$$

7. *Mean Square Error (MSE)* – measures the average of the squared differences between actual value and the value predicted by the regression model. As it squares the differences, it penalizes even a small error which leads to over-estimation of how bad the model is.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y)^2$$

8. *Root Means Square Error (RMSE)* – is the square root of MSE. It is preferred in some cases because the errors are squared first before averaging which poses high penalty on large errors. This implies that RMSE is useful when large errors are undesired.

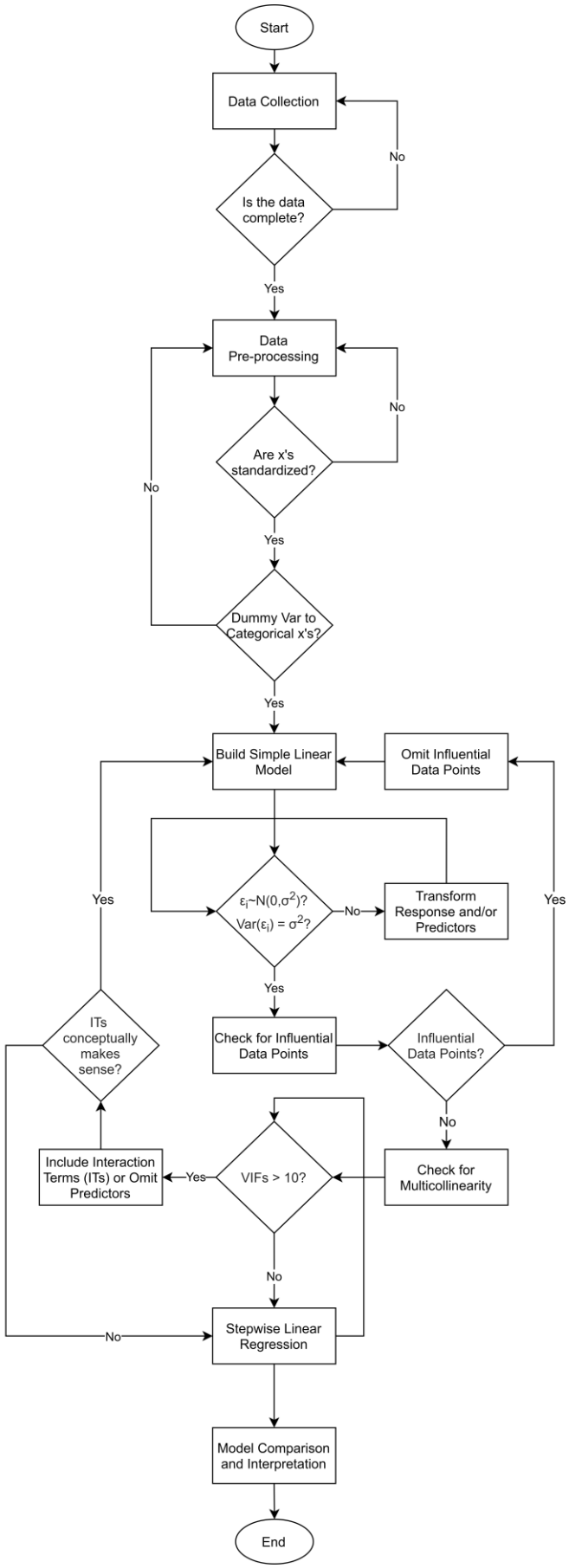
$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y)^2}$$

Predictors Relative Importance

There are different ways to estimate the relative importance of predictors, among which the method developed by Lindemann, Merenda and Gold (lmg; 1980) is often recommended. lmg calculates the relative contribution of each predictor to the R square with the consideration of the sequence of predictors appearing in the model.

III. Methodology

Linear Modeling Process Flowchart



The linear modeling process follows the flow chart shown. Steps taken in this study are discussed in detail below.

Data Collection

The data was sourced out from Honeywell Uniformance Process Studio, LabMate 5: Laboratory Information Mangement System and salt topping/changeout records from previous downstream engineers.

These were arranged in columns of MS Excel spreadsheet. Listed below are the data collected for this study.

Response Variable

D-4404 Salt Depletion Rate	-	mmpd	DepRate

Predictor Variables

Kerosene Feed	FI44005	MBSD	KerFeed
Kerosene Temperature	TI44002	°C	KerTemp
D-4403 (Water Wash) Level	LDIC44006	%	WaterLev
D-4406 (Drain Pot) Level	LDIC44008	%	BrineLev
D-4403 Water pH	-	-	WaterpH
D-4401 (Water Coalescer) Level	LDIC44002	%	WCBootLev
D-4402 (ECP) Inlet dP	PDIC44551	kPa(g)	ECPdP
D-4402 Level	LDIC44552	%	ECPLev
R-4401 (Reactor) Inlet Pressure	PI44005	kPa(g)	ReactorIP
R-4401 dP	PDI44004	kPa(a)	ReactordP
R-4401 Caustic Level	LDI44003	%	ReactorCauLev
D-4405 (Clay Filter) dP	PDI44006	kPa(a)	ClaydP
D-4405 Outlet Pressure	PIC44007	kPa(g)	ClayOP
Salt Topping Timeframe	-	-	D_1 to D_8

Other Significant Data

Shipping Jet A1 Density	-	used in correcting kerosene feed flowrate recorded by FI44005
Historical Salt Topping Data and Densities	-	used to compute historical salt depletion rates of D-4404
D-4404 Dimensions	-	used to compute historical salt depletion rates of D-4404
D-4404 Moisture Differerential	-	used to compare depletion rates based on design/typical, based on moisture differential and predicted

Data Pre-processing

Data Pre-processing or data cleaning is done to either complete or omit sets with missing data, re-scale or standardize predictors with different units and ranges, or assign dummy variables to categorical predictors.

For this study, all predictors are standardized using Z-score standardization. Moreover, eight (8) dummy variables is assigned to nine (9) salt topping timeframes. Notice that salt depletion per timeframe is independent of each other, that is, measurement of total salt depleted resets to 0 after each salt topping.

	Salt Topping/Changeout Timeframes	*Actual DepRate*
t_1	December 2014 to October 1,2015	2780.00
t_2	October 2, 2015 to April 1, 2016	1190.00
t_3	April 2, 2016 to September 8, 2016	1945.00
t_4	September 9, 2016 to April 29, 2017	2447.00
t_5	April 30, 2017 to September 27, 2017	1240.00
t_6	September 28, 2017 to May 7, 2018	2061.00
t_7	May 8, 2018 to November 26, 2018	1987.80
t_8	November 27, 2018 to June 7, 2019	2260.00
t_9	June 8, 2019 to January 10, 2020	1201.00

$$Timeframe = \begin{cases} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \end{cases} \quad D_1 = \begin{cases} 1, & t_1 \\ 0, & otherwise \end{cases} \quad D_2 = \begin{cases} 1, & t_2 \\ 0, & otherwise \end{cases} \quad D_3 = \begin{cases} 1, & t_3 \\ 0, & otherwise \end{cases} \quad D_4 = \begin{cases} 1, & t_4 \\ 0, & otherwise \end{cases} \quad D_5 = \begin{cases} 1, & t_5 \\ 0, & otherwise \end{cases} \quad D_6 = \begin{cases} 1, & t_6 \\ 0, & otherwise \end{cases} \quad D_7 = \begin{cases} 1, & t_7 \\ 0, & otherwise \end{cases} \quad D_8 = \begin{cases} 1, & t_8 \\ 0, & otherwise \end{cases}$$

	Dummy Variable							
Category	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8
t_1	1	0	0	0	0	0	0	0
t_2	0	1	0	0	0	0	0	0
t_3	0	0	1	0	0	0	0	0
t_4	0	0	0	1	0	0	0	0
t_5	0	0	0	0	1	0	0	0
t_6	0	0	0	0	0	1	0	0
t_7	0	0	0	0	0	0	1	0
t_8	0	0	0	0	0	0	0	1
t_9	0	0	0	0	0	0	0	0

Bases and assumptions for this study are summarized below:

1. The population dataset is divided into two subsets: training set and test set. The training set comprises 80% of the total data points.

2. Response variable, DepRate, is represented in millimeters of salt lost per day (mmpd).

3. Total actual salt depleted per timeframe was used to compute for actual salt depletion rate per day. The depletion rate was scaled against the corrected kerosene feed with the assumption that higher feed constitutes higher deleption rate. With this, shutdown periods are assumed to have zero depletion rates.

4. Kerosene feed flowrate measured by flow indicator FI44005 was corrected using averaged Jet-A1 shipping sample densities and kerosene temperature measured by TI44002.

5. Water wash vessel (D-4403) pH value ranges from 10-13 where values lowers per changeout. Since testing is not done everyday, previous values were copied until a new value comes up. For cases where pH is 0, values are replaced with 10 – signifying normal operation.

6. Moisture difference across salt filter's (D-4404) inlet and outlet is deemed unreliable in predicting salt depletion rate due to insufficient historical data.

7. Timeframe is treated as a categorical variable due to independence as discussed previously.

8. Predictor values are averaged per day. Moreover, standardization of the training set was done using the mean and standard deviation of the whole dataset from December 2014 to January 2020. Standardization for the test set is also done using the same mean and standard deviation.

Stepwise Linear Regression Modeling

The SLRM is done by using the `lm` function in R via RStudio (R Integrated Development Environment). Programming steps are simplified below.

1. Load the necessary R libraries. Libraries are depository of shortcut codes developed by R users.

2. Set the folder containing cleaned data as the working directory. Input the dataset and create a partition of training and test data using the `createDataPartition` function of `caret` package in R. For this study, 80% is used as training set.

3. Build SLRM using `lm` function. Check for model adequacy by plotting residuals vs fitted using `plot` fucntion. Also check for normality by using qqnorm and qqline function.

4. Transform SLRM as necessary. Recheck for model adequacy.

5. Check for leverage points and outliers by using `hatvalues` and `rstudent` functions, respectively. Draw line limits using `abline` function for leverage points and outliers as discussed in the theoretical framework. Use `ols_plot_resid_lev` function of `olsrr` package to have a summary of influential points.

6. Remove influential points and check overall effect in model accuracy by comparing performance metrics: R^2, R^2_{adj}, RSE, MAE, MSE, and $RMSE$.

7. Check for pairwise multicollinearty using `rcorr` function of `Hmisc` library. Include a correlation plot using `corrplot` function. Then, evaluate if correlations are conceptually

sound. Include these interaction terms to the linear model afterwards.

8. Perform initial stepwise AIC testing using `step` function. Verify if any multicollineariy issue persists using vif function of car package. Omit variables with the highest VIFs one-by-one until all VIFs are less than 10.

9. Evaluate model accuracy and conduct model comparisons. The optimum model shall satisfy the critical classic assumptions of linear regression as discussed in the theoretical framework.

10. Interpret the model generated and infer findings from the model behavior. This includes estimated coefficients, observed trends between numerical predictors and dummy variables. Further simplify the model if possible and as necessary.

11. Create necessary benchmarks to compare the results of developed models.

12. Identify how much each predictors affect the value of the repsonse. Calculate the relative importance of each predictor using `calc.relimp` function of `relaimpo` package.

13. Compare accuracy of linear model developed to previous estimation methods. This includes the declared typical depletion rate of 20 lbs salt lost/ MB of kerosene feed and moisture difference between D-4404 inlet and outlet.

14. Identify the optimum model. Draw conclusions and recommend any improvements.

IV. Results and Discussion

Data Pre-processing

Predictor variables of the training set are Z-score standardized over the means, μ, and standard deviations, σ, of their population set. Means and standard deviations are summarized in Table 1.

Table 1. *Z-Score Standardization Mean, μ, and Standard Deviation, σ*

Predictors, x_i	***μ***	***σ***
$KerFeed$	6.2042	2.9422
$KerTemp$	34.4100	3.4283
$WaterLev$	41.8265	14.31645
$BrineLev$	39.0035	15.1963
$WaterpH$	11.7125	0.4660
$WCBootLev$	44.9397	17.2298
$ECPdP$	19.09487	7.2755
$ECPLev$	77.3308	10.1767
$ReactorIP$	680.9490	215.7681
$ReactordP$	27.7571	37.4603
$ReactorCauLev$	51.4823	18.8022
$ClaydP$	46.0314	27.6244
$ClayOP$	593.0870	197.5045

The values present in Table 1 shall be used in the standardization of the test set and future values for prediction.

Simple Linear Regression Modeling

The Simple Linear Regression Model (SLRM) summarized below generated a coefficient of determination, R^2, of 0.9704.

Model 1 – SLRM:

$$DepRate = D1 + D2 + D3 + D4 + D5 + D6 + D7 + D8 + KerFeed + KerTemp + WaterLev + BrineLev + WaterpH + WCBootLev + ECPdP + ECPLev + ReactorIP + ReactordP + ReactorCauLev + ClaydP + ClayOP$$

Despite having high R^2, model diagnostic is still required to check for residuals' mean, variance, correlation and normality before proceeding with model interpretation, response variable prediction, model evaluation and model comparisons.

Model Diagnostics

Model diagnostics are performed below to check whether the model conforms to the classical assumptions of linear regression.

1. *Residuals Mean, Variance and Correlation*

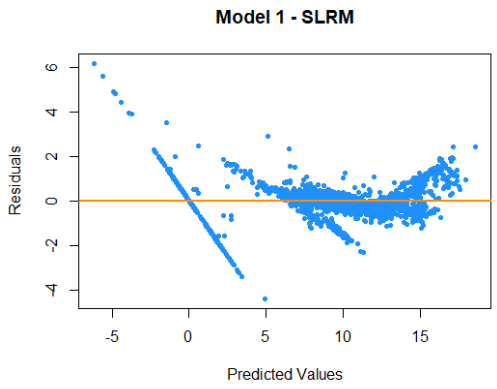


***Figure 1**: Model 1 - Residuals vs Fitted Plot*

Residual vs Fitted plot like Figure 1 is used for checking both the linearity and constant variance assumption. It shows the predicted values made by the model on the x-axis and the accuracy of prediction (residuals) on the y-axis. The distance of a point from the line zero shows how bad the prediction was for that specific value. A "good" residuals vs fitted plot should be relatively shapeless without clear patterns in the data, no obvious outliers, and be generally symmetrically distributed around the zero line without particularly large residuals.

From Figure 1 it can seen that there are negative predicted values. However, salt depletion rates shall always be greater than or equal to zero. Moreover, a linear relationship at the rightmost part of the graph is present in predicted values ranging from -5 to 0 and 0 to +5 as summarized below.

-5	to	0	↑ value, ↑ accuracy, ↓ residual
0	to	5	↑ value, ↓ accuracy, ↑ residual

Furthermore it can also be perceived that the magnitude of residuals increases as predicted values approaches and goes beyond 15.

These observations clearly violates three of the classical assumptions. That is, the residuals' mean is not zero and the residuals' variance is not constant with evident correlations seen.

2. *Residuals Distribution*

Recall that by virtue of Gauss-Markov Theorem, the residuals' distribution is not required to be normal given that the residuals are uncorrelated with zero mean and constant variance. Still, to check for the normality of residuals, refer to the histogram and Q-Q plot shown in Figure 2 and Figure 3.

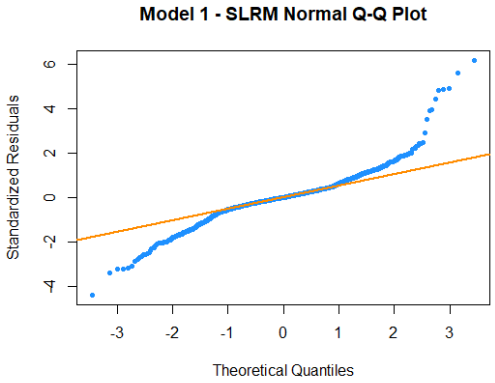


***Figure 2**: Model 1 – Residuals Normal Q-Q Plot*

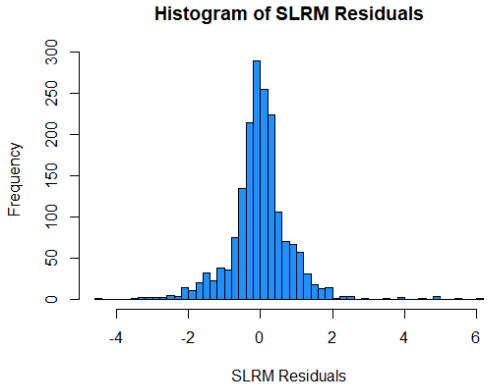


***Figure 3**: Model 1 – Residuals Histogram*

Since the residuals did not follow a straight line in Figure 2, the distribution of SLRM residuals is not normal. This was verified by the high peakedness of the histogram (leptokurtic) where the distribution indicates the presence of large outliers from either extremes (either too low or too high).

Transformed SLRM

To augment the violations discussed in model diagnostics, the response variable, $DepRate$, was log-transformed. However, since some of the values of $DepRate$ are zero, an arbitrary constant will be added to avoid undefined datapoints due to ln(zero) or ln(negative number). The constant, c, that will be used is 16 as summarized below.

Model 2 – SLRM Trans:

$$\ln(DepRate + 16) = D1 + D2 + D3 + D4 + D5 + D6 + D7 + D8 + KerFeed + KerTemp + WaterLev + BrineLev + WaterpH + WCBootLev + ECPdP + ECPLev + ReactorIP + ReactordP + ReactorCauLev + ClaydP + ClayOP$$

The transformation helped stabilize the variance of the rightmost part of the graph as shown in the comparison in Figure 5. However, in Figure 4, the linear relationship observed in the leftmost part of Figure 1 did not disappear.

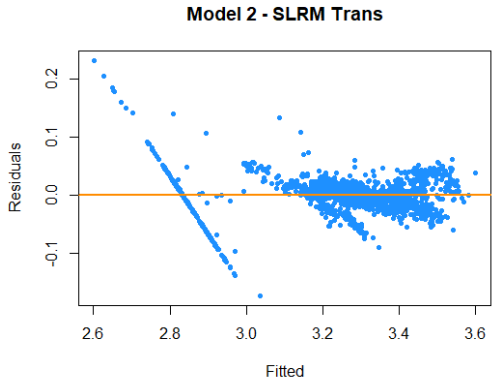


***Figure 4**: Model 2 – Residuals vs Fitted Plot*

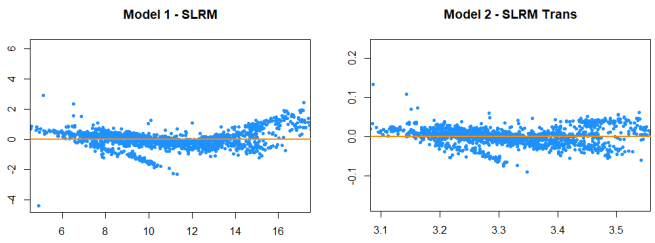


***Figure 5**: SLRM and SLRM Trans Variance Comparison*

Further investigation has shown that the linear relationship at the leftmost parts of Figure 1 and 4 was caused by shutdown periods of KMX-2 where kerosene feed ($KerFeed$) is zero and consequentially, the $DepRate$ is assumed to be zero. Negative predicted values of $DepRate$ are caused by coefficients of other predictors with operational measurements even if KMX-2 is on shutdown (e.g. level). Refer to log-tranformed SLRM without shutdown periods shown in Figure 6. Addition of arbitrary constant 16 is not needed anymore since all of the values are greater than zero.

Model 3 – SLRM Trans No SD:

$$\ln(DepRate) = D1 + D2 + D3 + D4 + D5 + D6 + D7 + D8 + KerFeed + WaterLev + BrineLev + WaterpH + WCBootLev + ECPdP + ECPLev + ReactorIP + ReactordP + ReactorCauLev + ClaydP + ClayOP$$

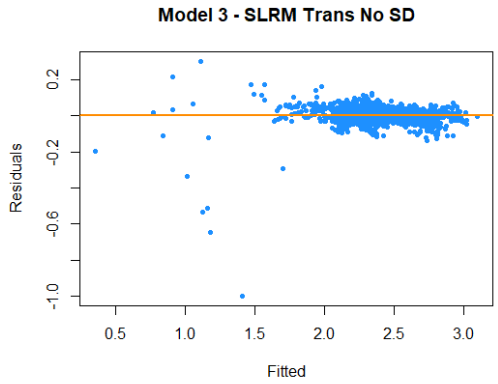


***Figure 6**: Model 3 - Residuals vs Fitted Plot*

Omitting shutdown values (zero $KerFeed$ and $DepRate$ values) from the population solved the problem with the constant variance assumption. However, it must be enforced during prediction phase that zero depletion rates are assumed to be zero during shutdowns.

Moreover, it look as if there some outliers in the data as shown in the leftmost part of Figure 6. The analysis whether or not these points will significantly affect the behaviour of the linear model will be verified and tested in the next section.

3. *Influential Data Points*

Existence of leverage points (extreme x-values) and outliers (extreme y-values) were detected by leverage statistic and studentized residual respectively. Results are plotted in Figures 7, 8 and 9.

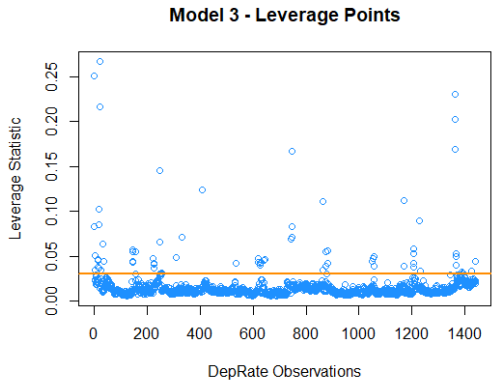


***Figure 7**: SLRM Trans Leverage Points*

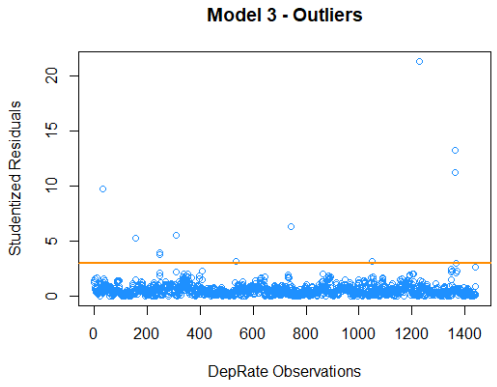


***Figure 8**: SLRM Trans Outliers (Studentized Residuals)*

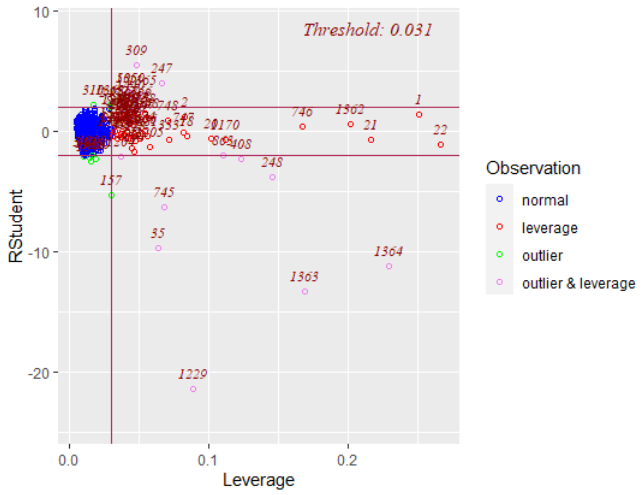


***Figure 9**: Outlier and Leverage Diagnostic for SLRM Trans*

A list of influential data points (outliers with high leverage statistic) generated from Figure 9 is summarized in Table 2. Removal of these influential points will be included in model comparisons to see if the accuracy of predicted values will drastically change.

***Table 2.** Influential Points - DepRate and KerFeed*

DepRate	*Feed*	*DepRate*	*Feed*	*DepRate*	*Feed*
0.90	0.63	0.97	0.67	0.81	0.45
2.08	2.27	1.07	0.85	7.54	4.22
0.17	0.19	4.19	3.32	9.90	5.54
3.10	3.38	15.56	11.29	10.65	5.96
1.85	1.02	0.51	0.21	7.01	3.93
4.69	3.26	0.71	0.40		

From the list, it can be understood that most of the influential data points are those approaching shutdown periods (approaching zero `KerFeed`) considering that the declared minimum feed rate of KMX-2 is 4.8 MBSD. The effect of removal of these influential points significantly improved the residuals vs fitted outlier residuals of the model as shown in Figure 10.

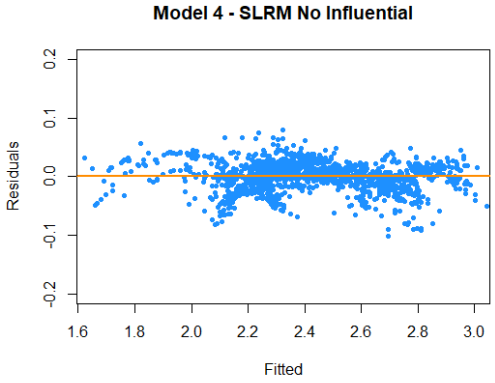


***Figure 10**: Model 4 (Model 3 without Influential Data Points) - Residuals vs Fitted Plot*

4. *Multicollinearity*

Since there are 21 predictors involved in the study, it is expected to have 210 pairwise combinations (without repetition) of predictors with possibility of correlation. Interactions involving dummy variables and other numerical predictors are discussed subsequent sections and are therefore omitted.

Figure 11 summarizes the generated positive and negative correlations out of all the numerical predictors. Out of 78 pairs, 57 pairs have correlations with p-value less than 0.05, which means that these 57 pairs have significant interactions affecting the model. In lieu with this, it is important to remember that correlation does not imply causation. Out of the 57 pairs, only 11 pairs are conceptually related as listed in Table 3. Inclusion of additional conceptually-sound pairs is done (despite having p-values greater than 0.05) to account for all possible interactions.

Table 3. *Pairwise Combinations with Interactions*

Interactions	***corr***	***p***
$ReactorIP:ClayOP$	0.7186	0.000
$KerFeed:ReactorIP$	0.7170	0.000
$ReactorIP:ClaydP$	0.4924	0.000
$KerFeed:ClaydP$	0.4006	0.000
$KerFeed:ClayOP$	0.3546	0.000
$ReactorIP:ReactordP$	0.2244	0.000
$ReactordP:ClayOP$	0.1867	0.000
$KerFeed:KerTemp$	0.1738	0.000
$KerFeed:ReactordP$	0.1636	0.000
$WaterLev:WaterpH$	-0.1533	0.000
$ReactorIP:KerTemp$	0.1118	0.000
$ReactordP:ClaydP$	0.0451	0.087
$KerFeed:BrineLev$	0.0422	0.110
$KerTemp:ClayOP$	0.0336	0.203
$ClaydP:ClayOP$	0.0122	0.643

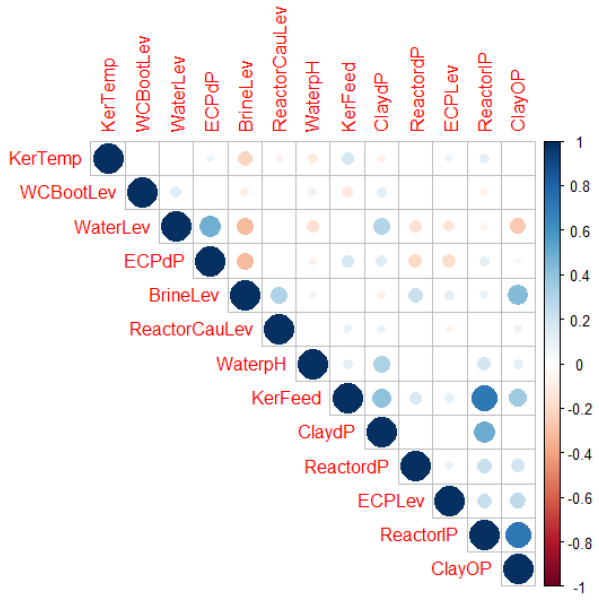


Figure 11*: Predictor Combination Correlation Plot*

Incorporating these interactions in Model 3 gives Model 5. Interpretations and comparison of models will be discussed in the next sections.

Model 5 – SLRM Trans No SD With Interactions:

$$\begin{aligned} ln(DepRate) = {} & D1 + D2 + D3 + D4 + D5 + D6 + D7 + D8 \\ & + KerFeed + KerTemp + WaterLev \\ & + BrineLev + WaterpH + WCBootLev \\ & + ECPdP + ECPLev + ReactorIP \\ & + ReactordP + ReactorCauLev + ClaydP \\ & + ClayOP + ReactorIP:ClayOP \\ & + KerFeed:ReactorIP + ReactorIP:ClaydP \\ & + KerFeed:ClaydP + KerFeed:ClayOP \\ & + ReactorIP:ReactordP + ReactordP:ClayOP \\ & + KerFeed:KerTemp + KerFeed:ReactordP \\ & + WaterLev:WaterpH + ReactorIP:KerTemp \\ & + ReactordP:ClaydP + KerFeed:BrineLev \\ & + KerTemp:ClayOP + ClaydP:ClayOP \end{aligned}$$

Model Tuning and Comparison

So far, this study have generated four models, Model 3, 4, 5 and 5a, conforming to the classical assumptions of linear regression. This section will also discuss fine-tuned versions of Model 5a: Model 6, 7, and 8. Model 9 where timeframe dummy variables are omitted will also be explored.

Table 4. *Model Summary*

Models	***Description***
Model 0	SLRM with unstandardized predictors
Model 3	SLRM Trans Without Shutdowns
Model 4	Model 3 Without Influential Points
Model 5	Model 3 With Interactions
Model 5a	Model 4 With Interactions
Model 6	Model 5a SPLR
Model 7	Model 6 SPLR without VIFs > 10
Model 8	Simplified Model 7
Model 9	Model Without Timeframe Dummy

1. Stepwise Linear Regression (SPLR)

Model 0 will serve as benchmark of improvement of the different models generated. Performance metrics for models 3 to 5a are tabulated in Table 5.

Table 5. *Performance Metrics Models 3 to 5a*

	Model 3	***Model 4***	***Model 5***	***Model 5a***
R^2	0.9610	0.9889	0.9909	0.9975
R^2_{adj}	0.9604	0.9887	0.9907	0.9975
RSE	0.0635	0.0292	0.0308	0.0138
MAE	0.9738	0.9870	1.0151	0.9947
MSE	1.1734	1.0698	1.0803	1.0128
$RMSE$	1.0832	1.0343	1.0394	1.0064

From the results, it can be inferred that the exclusion of influential data points and the inclusion of interaction terms does not significantly affect the accuracy of the model. Nonetheless, it is evident that Model 5a, the log-transformed SLRM without Influential Points and with Interactions, performed best as it have the lowest error in terms of RSE, MSE and $RMSE$. It also attained the highest R^2 and R^2_{adj}.

Moving on, SPLR is employed in Model 5a to check the best combination of predictor variables in terms of AIC score. The iterative omission of predictors led to an optimized AIC score of -12163.63. Overall, the following predictor variables (highlighted in *blue* in Model 6 summary) are omitted: $ECPLev$, $ReactorCauLev$, $ReactorIP:ClayOP$, $WaterLev:WaterpH$ and $ReactordP:ClaydP$. Moreover, the performance metrics of Model 5a after transitioning to Model 6 did not changed at all.

Model 6 – Model 5a SPLR:

$$\begin{aligned} ln(DepRate) = {} & D1 + D2 + D3 + D4 + D5 + D6 + D7 + D8 \\ & + KerFeed + KerTemp + WaterLev \\ & + BrineLev + WaterpH + WCBootLev \\ & + ECPdP + ECPLev + ReactorIP \\ & + ReactordP + ReactorCauLev + ClaydP \\ & + ClayOP + ReactorIP:ClayOP \\ & + KerFeed:ReactorIP + ReactorIP:ClaydP \\ & + KerFeed:ClaydP + KerFeed:ClayOP \\ & + ReactorIP:ReactordP + ReactordP:ClayOP \\ & + KerFeed:KerTemp + KerFeed:ReactordP \\ & + WaterLev:WaterpH + ReactorIP:KerTemp \\ & + ReactordP:ClaydP + KerFeed:BrineLev \\ & + KerTemp:ClayOP + ClaydP:ClayOP \end{aligned}$$

Model 6 is not the optimum model yet. It is imperative to check for VIFs breaching 10 to make sure that there are no multicollinearity issues left. VIFs for Model 6 are summarized in Table 6.

Table 6. *Model 6 Predictors VIFs*

x_i	***VIF***	x_i	***VIF***
$D1$	18.60	$ReactordP$	12.73
$D2$	10.74	$ClaydP$	16.26
$D3$	9.397	$ClayOP$	16.46
$D4$	13.15	$KerFeed:ReactorIP$	23.69
$D5$	5.680	$ReactorIP:ClaydP$	12.93
$D6$	10.43	$KerFeed:ClaydP$	9.361
$D7$	10.09	$KerFeed:ClayOP$	27.31
$D8$	11.66	$ReactorIP:ReactordP$	14.24
$KerFeed$	11.13	$ReactordP:ClayOP$	28.74
$KerTemp$	6.318	$KerFeed:KerTemp$	4.937
$WaterLev$	3.736	$KerFeed:ReactordP$	1.889
$BrineLev$	4.108	$ReactorIP:KerTemp$	15.58
$WaterpH$	1.555	$KerFeed:BrineLev$	3.284
$WCBootLev$	1.226	$ClayOP:KerTemp$	12.15
$ECPdP$	2.137	$ClaydP:ClayOP$	19.76
$ReactorIP$	19.31		

Recall that if there are VIFs greater than 10 after standardization, the multicollinearity is caused by the predictors itself and not the data structure. Hence, it is prescribed to remove the numerical predictor with highest VIF (within VIFs > 10) which is $ReactorIP$ at 19.31. The interactions involving $ReactorIP$ shall also be omitted.

Table 7. *VIFs Without ReactorIP*

x_i	***VIF***	x_i	***VIF***
$D1$	15.98	$ReactordP$	10.59
$D2$	8.654	$ClaydP$	13.56
$D3$	7.574	$ClayOP$	4.990
$D4$	10.30	$KerFeed:ReactorIP$	-
$D5$	4.558	$ReactorIP:ClaydP$	-
$D6$	8.561	$KerFeed:ClaydP$	2.550
$D7$	8.611	$KerFeed:ClayOP$	12.21
$D8$	10.66	$ReactorIP:ReactordP$	-
$KerFeed$	7.091	$ReactordP:ClayOP$	10.43
$KerTemp$	6.131	$KerFeed:KerTemp$	2.195
$WaterLev$	3.502	$KerFeed:ReactordP$	1.610
$BrineLev$	3.885	$ReactorIP:KerTemp$	-
$WaterpH$	1.436	$KerFeed:BrineLev$	2.249
$WCBootLev$	1.194	$ClayOP:KerTemp$	6.814
$ECPdP$	2.114	$ClaydP:ClayOP$	13.72
$ReactorIP$	-		

After the removal of $ReactorIP$, it was observed that there are still VIFs greater than 10. Repeating the step, $ClaydP$ (and related interactions) with VIF of 13.56 shall be removed.

Table 8. *VIFs Without ClaydP*

x_i	VIF	x_i	VIF
$D1$	10.35	$ReactordP$	9.885
$D2$	5.729	$ClaydP$	-
$D3$	5.249	$ClayOP$	4.613
$D4$	6.237	$KerFeed{:}ReactorIP$	-
$D5$	3.615	$ReactorIP{:}ClaydP$	-
$D6$	4.337	$KerFeed{:}ClaydP$	-
$D7$	3.892	$KerFeed{:}ClayOP$	3.737
$D8$	3.502	$ReactorIP{:}ReactordP$	-
$KerFeed$	2.892	$ReactordP{:}ClayOP$	9.682
$KerTemp$	6.007	$KerFeed{:}KerTemp$	2.189
$WaterLev$	3.464	$KerFeed{:}ReactordP$	1.518
$BrineLev$	3.817	$ReactorIP{:}KerTemp$	-
$WaterpH$	1.389	$KerFeed{:}BrineLev$	2.122
$WCBootLev$	1.184	$ClayOP{:}KerTemp$	6.674
$ECPdP$	2.080	$ClaydP{:}ClayOP$	-
$ReactorIP$	-		

Notice that $D1$ is still slightly above 10 at 10.35. However, it will not be omitted since comparing timeframes is one of this study's objectives. Furthermore, the model shall undergo another stepwise AIC testing to see if there are any unnecessary predictors left.

After testing, the following predictor variables were omitted: $WaterLev$ and $WaterpH$. The final form of Model 7 is shown below.

Model 7 – Model 6 SPLR Without VIFs > 10:

$$\begin{aligned} ln(DepRate) = {} & D1 + D2 + D3 + D4 + D5 + D6 + D7 + D8 \\ & + KerFeed + KerTemp + BrineLev \\ & + WCBootLev + ECPdP + ReactordP \\ & + ClayOP + KerFeed{:}ClayOP \\ & + ReactordP{:}ClayOP + KerFeed{:}KerTemp \\ & + KerFeed{:}ReactordP + KerFeed{:}BrineLev \\ & + KerTemp{:}ClayOP \end{aligned}$$

Notice that after the removal of $WaterLev$ and $WaterpH$, all of the VIFs are less than 10 as summarized in Table 9. With this, Model 7 is deemed as the best model under SPLR.

Table 9. *VIFs Without WaterLev and WaterpH*

x_i	VIF	x_i	VIF
$D1$	9.133	$ReactordP$	9.757
$D2$	4.621	$ClaydP$	-
$D3$	3.988	$ClayOP$	4.562
$D4$	5.223	$KerFeed{:}ReactorIP$	-
$D5$	3.021	$ReactorIP{:}ClaydP$	-
$D6$	3.675	$KerFeed{:}ClaydP$	-
$D7$	3.511	$KerFeed{:}ClayOP$	3.498
$D8$	3.099	$ReactorIP{:}ReactordP$	-
$KerFeed$	2.820	$ReactordP{:}ClayOP$	9.485
$KerTemp$	5.918	$KerFeed{:}KerTemp$	2.180
$WaterLev$	-	$KerFeed{:}ReactordP$	1.509
$BrineLev$	3.753	$ReactorIP{:}KerTemp$	-
$WaterpH$	-	$KerFeed{:}BrineLev$	2.097
$WCBootLev$	1.162	$ClayOP{:}KerTemp$	6.530
$ECPdP$	2.063	$ClaydP{:}ClayOP$	-
$ReactorIP$	-		

Finally, comparison of Models 5a, 6 and 7 performance metrics are summarized in Table 10. Despite having better performance metrics than Model 7, Models 5a and 6 violates the multicollinearity assumption of linear regression. Still, omission of several predictors did not significantly increased the amount of error in the predictions.

Table 10. *Performance Metrics of Models 5a to 7*

	Model 5a	Model 6	Model 7
R^2	0.9975	0.9975	0.9946
R^2_{adj}	0.9975	0.9975	0.9946
RSE	0.0138	0.0138	0.0203
MAE	0.9947	0.9947	0.9995
MSE	1.0128	1.0128	1.0631
$RMSE$	1.0064	1.0064	1.0311

Best Model Interpretation

This section includes interpretation of the optimum model, Model 7, coefficients, interactions, dummy variables and performance metrics.

1. Performance Metrics

Table 12. *Model 7 Estimated Coefficients, $\widehat{\beta}_i$*

x_i	$\widehat{\beta}_i$	t value	$Pr(>\|t\|)$
Intercept	2.431	695.825	< 2e-16
$D1$	-0.196	-45.527	< 2e-16
$D2$	-0.574	-156.195	< 2e-16

$D3$	0.041	10.790	< 2e-16
$D4$	-0.173	-49.774	< 2e-16
$D5$	-0.192	-56.498	< 2e-16
$D6$	-0.300	-98.052	< 2e-16
$D7$	-0.211	-69.035	< 2e-16
$D8$	0.325	99.647	< 2e-16
$KerFeed$	0.416	229.858	< 2e-16
$KerTemp$	-0.001	-0.541	0.5885
$BrineLev$	-0.004	-1.808	0.0708
$WCBootLev$	0.003	3.062	0.0022
$ECPdP$	0.004	2.236	0.0255
$ReactordP$	-0.047	-9.001	< 2e-16
$ClayOP$	0.073	16.715	< 2e-16
$KerFeed:ClayOP$	-0.125	-38.988	< 2e-16
$ReactordP:ClayOP$	0.070	10.163	< 2e-16
$KerFeed:KerTemp$	-0.007	-5.224	2.0e-07
$KerFeed:ReactordP$	-0.034	-10.405	< 2e-16
$KerFeed:BrineLev$	0.037	16.260	< 2e-16
$KerTemp:ClayOP$	0.009	3.093	0.0020

The t-values of most of the predictor variables are faraway from zero except for $KerTemp$ and $BrineLev$. Morever, the p-values of the aforementioned predictors are both greater than 0.05. Hence, it can be concluded that both $KerTemp$ and $BrineLev$ are statistically insignificant in predicting $DepRate$. However, these were not omitted during stepwise AIC because the interactions involved are statistically significant (i.e. $KerFeed:KerTemp$ and $KerFeed:BrineLev$).

Moroever, from Table 10, both the R^2 and R^2_{adj} is 0.9946 which means that 99.46% of the total deviations of $DepRate$ from its true value is explained by the linear model developed. Furthermore, interpreting its RSE, $DepRate$ will deviate from the true value by 0.0203 mmpd on average – calculated upon 1401 degrees of freedom. The mean absolute error is 0.9995 mmpd and both the MSE and RMSE are both near it at 1.0631 and 1.0311 mmpd^2, respectively. This signifies that there are no significantly small nor large prediction errors.

Moreover, the F-statistic of the model is very high at 12,370. Recall that F-statistic is defined by the ratio $\frac{MSM}{MSR}$ (Mean Squares due to the Model over the Mean Squares due to the Residuals). Notice that since the F-statistic is huge, it can be inferred that MSM is very high. From this, it can be concluded that the model performs very well since most of the deviations of the predicted $DepRate$ are explained by the model itself as is with the inferences made using R^2 and R^2_{adj}. On the contrary, in cases where MSR is very high, the model cannot explain the deviations anymore.

2. <u>Numerical Predictors Estimated Coefficients</u>

The interpretation will follow the log-linear interpretation as discussed in the theoretical framework.

log-linear	$\ln Y = \beta_0 + \beta_1 x + \varepsilon$	a unit change in x corresponds to $100 * (e^{\beta_1} - 1)$ or $100\beta_1$ percent change in Y

From Table 13 it can be inferred that changes in $KerFeed$, $ReactordP$ and $ClayOP$ significantly affects the $DepRate$. These variables are also logical operation-wise. That is, if $KerFeed$ increases, the salt depletion rate is expected to increase. Moreover, when $ReactordP$ decreases (most likely due to decrease in $KerFeed$), the depletion rate is likely to decrease also. Lastly, if $ClayOP$ is increased through increase in $KerFeed$ or through pressure control (PIC44007), the increased feed flow or the build-up of pressure in the system may lead to higher salt depletion.

Table 13. *Model 7 Numerical Predictors Interpretation*

x_i	$\widehat{\beta_i}$	$(e^{\widehat{\beta_i}} - 1)$	***Interpretation***
$KerFeed$	0.416	51.57%	One unit increase in $KerFeed$ increases $DepRate$ by 51.57%
$KerTemp$	-0.001	-0.08%	One unit increase in $KerTemp$ decreases $DepRate$ by 0.08%
$BrineLev$	-0.004	-0.36%	One unit increase in $BrineLev$ decreases $DepRate$ by 0.36%

$WCBootLev$	0.003	0.28%	One unit increase in $WCBootLev$ increases $DepRate$ by 0.28%
$ECPdP$	0.004	0.43%	One unit increase in $ECPdP$ increases $DepRate$ by 0.43%
$ReactordP$	-0.047	-4.59%	One unit increase in $ReactordP$ decreases $DepRate$ by 4.59%
$ClayOP$	0.073	7.56%	One unit increase in $ClayOP$ increases $DepRate$ by 7.56%

Conversely, predictor variables $KerTemp$, $BrineLev$, $WCBootLev$, and $ECPdP$ had relatively lower effects in predicting $DepRate$. Also, notice that a 0.08% decrease in $DepRate$ when $KerTemp$ is increased by one unit is counterintuitive. The solubility of rock salt increases with temperature. Morever, the perceived 0.36% decrease in $DepRate$ if $BrineLev$ increases by one unit is also counterintuitive. If $DepRate$ is decreasing by some phenomena, the level of brine in D-4406 ($BrineLev$) is expected to decrease as well. Finally, the predictors $WCBootLev$ and $ECPdP$ are operationally controlled and any drastic change in their values is not expected. With this, Model 7 can be further simplified by omitting these predictors and their related interaction terms. Refer to Model 8 summary, Tables 14, 15, 16, and 17 for the simplified Model 7.

Model 8 – Simplified Model 7:

$$ln(DepRate) = D1 + D2 + D3 + D4 + D5 + D6 + D7 + D8 + KerFeed + ReactordP + ClayOP + KerFeed{:}ClayOP + ReactordP{:}ClayOP + KerFeed{:}ReactordP$$

From Table 15 it is evident that Model 7 has better performance metrics. However, it is also noticeable that omitting aforementioned variables did not significantly lower the accuracy of the generated Model 8.

Table 14. *Model 8 Estimated Coefficients, $\widehat{\beta}_\iota$*

x_i	$\widehat{\beta_\iota}$	t value	$Pr(>\|t\|)$
$Intercept$	2.437	716.365	< 2e-16
$D1$	-0.197	-49.369	< 2e-16
$D2$	-0.585	-153.583	< 2e-16
$D3$	0.033	8.490	< 2e-16
$D4$	-0.179	-51.188	< 2e-16
$D5$	-0.196	-55.371	< 2e-16
$D6$	-0.301	-92.564	< 2e-16
$D7$	-0.211	-63.872	< 2e-16
$D8$	0.329	94.453	< 2e-16
$KerFeed$	0.407	227.407	< 2e-16
$ReactordP$	-0.021	-4.044	5.55e-05
$ClayOP$	0.073	16.009	< 2e-16
$KerFeed{:}ClayOP$	-0.097	-34.289	< 2e-16
$ReactordP{:}ClayOP$	0.035	5.102	3.81e-07
$KerFeed{:}ReactordP$	-0.030	-8.396	< 2e-16

Table 15. *Performance Metrics of Model 7 and 8*

	Model 7	***Model 8***
R^2	0.9946	0.9932
R^2_{adj}	0.9946	0.9932
RSE	0.0203	0.0227
MAE	0.9995	0.9973
MSE	1.0631	1.0684
$RMSE$	1.0311	1.0336

The VIFs of all the predictors in Model 8 are all less than 10 – passing the no multicollinearity criteria of linear regression.

Table 16. *VIFs of Model 8 Predictors*

x_i	VIF	x_i	VIF
$D1$	6.264	$D8$	2.828
$D2$	3.959	$KerFeed$	2.207
$D3$	3.283	$ReactordP$	8.076
$D4$	4.258	$ClayOP$	3.984
$D5$	2.611	$KerFeed{:}ClayOP$	2.181
$D6$	3.322	$ReactordP{:}ClayOP$	7.709
$D7$	3.301	$KerFeed{:}ReactordP$	1.449

Table 17. *Model 8 Numerical Predictors Interpretation*

x_i	$\widehat{\beta_\iota}$	$(e^{\widehat{\beta_\iota}} - 1)$	***Interpretation***
$KerFeed$	0.407	50.27%	One unit increase in $KerFeed$ increases $DepRate$ by 50.27%
$ReactordP$	-0.021	-2.13%	One unit increase in $ReactordP$ decreases $DepRate$ by 2.13%
$ClayOP$	0.073	7.57%	One unit increase in $ClayOP$ increases $DepRate$ by 7.57%

3. *Dummy Variables*

To interpret dummy variables, let Φ to be the net effect of numerical variables and related interactions. With this, Model 8 can be rewritten as:

$$ln(DepRate) = 2.437 - 0.197(D1) - 0.585(D2) + 0.033(D3) - 0.179(D4) - 0.196(D5) - 0.301(D6) - 0.211(D7) + 0.329(D8) + \Phi$$

$D1 = 0;\ D2 = 0;\ D3 = 0;\ D4 = 0;$
$D5 = 0;\ D6 = 0;\ D7 = 0;\ D8 = 0;$
$ln(DepRate) = 2.437 - 0.197(0) - 0.585(0) + 0.033(0) - 0.179(0) - 0.196(0) - 0.301(0) - 0.211(0) + 0.329(0) + \Phi$
$\boldsymbol{ln(DepRate) = 2.437 + \Phi}$ *at timeframe* $= t_9$

$D1 = 1;\ D2 = 0;\ D3 = 0;\ D4 = 0;$
$D5 = 0;\ D6 = 0;\ D7 = 0;\ D8 = 0;$
$ln(DepRate) = 2.437 - 0.197(1) - 0.585(0) + 0.033(0) - 0.179(0) - 0.196(0) - 0.301(0) - 0.211(0) + 0.329(0) + \Phi$
$\boldsymbol{ln(DepRate) = 2.240 + \Phi}$ *at timeframe* $= t_1$

$D1 = 0;\ D2 = 1;\ D3 = 0;\ D4 = 0;$
$D5 = 0;\ D6 = 0;\ D7 = 0;\ D8 = 0;$
$ln(DepRate) = 2.437 - 0.197(0) - 0.585(1) + 0.033(0) - 0.179(0) - 0.196(0) - 0.301(0) - 0.211(0) + 0.329(0) + \Phi$
$\boldsymbol{ln(DepRate) = 1.852 + \Phi}$ *at timeframe* $= t_2$

$D1 = 0;\ D2 = 0;\ D3 = 1;\ D4 = 0;$
$D5 = 0;\ D6 = 0;\ D7 = 0;\ D8 = 0;$
$ln(DepRate) = 2.437 - 0.197(0) - 0.585(0) + 0.033(1) - 0.179(0) - 0.196(0) - 0.301(0) - 0.211(0) + 0.329(0) + \Phi$
$\boldsymbol{ln(DepRate) = 2.469 + \Phi}$ *at timeframe* $= t_3$

$D1 = 1;\ D2 = 0;\ D3 = 0;\ D4 = 1;$
$D5 = 0;\ D6 = 0;\ D7 = 0;\ D8 = 0;$
$ln(DepRate) = 2.437 - 0.197(0) - 0.585(0) + 0.033(0) - 0.179(1) - 0.196(0) - 0.301(0) - 0.211(0) + 0.329(0) + \Phi$
$\boldsymbol{ln(DepRate) = 2.257 + \Phi}$ *at timeframe* $= t_4$

$D1 = 1;\ D2 = 0;\ D3 = 0;\ D4 = 0;$
$D5 = 1;\ D6 = 0;\ D7 = 0;\ D8 = 0;$
$ln(DepRate) = 2.437 - 0.197(0) - 0.585(0) + 0.033(0) - 0.179(0) - 0.196(1) - 0.301(0) - 0.211(0) + 0.329(0) + \Phi$
$\boldsymbol{ln(DepRate) = 2.241 + \Phi}$ *at timeframe* $= t_5$

$D1 = 1;\ D2 = 0;\ D3 = 0;\ D4 = 0;$
$D5 = 0;\ D6 = 1;\ D7 = 0;\ D8 = 0;$
$ln(DepRate) = 2.437 - 0.197(0) - 0.585(0) + 0.033(0) - 0.179(0) - 0.196(0) - 0.301(1) - 0.211(0) + 0.329(0) + \Phi$
$\boldsymbol{ln(DepRate) = 2.136 + \Phi}$ *at timeframe* $= t_6$

$D1 = 1;\ D2 = 0;\ D3 = 0;\ D4 = 0;$
$D5 = 0;\ D6 = 0;\ D7 = 1;\ D8 = 0;$
$ln(DepRate) = 2.437 - 0.197(0) - 0.585(0) + 0.033(0) - 0.179(0) - 0.196(0) - 0.301(0) - 0.211(1) + 0.329(0) + \Phi$
$\boldsymbol{ln(DepRate) = 2.225 + \Phi}$ *at timeframe* $= t_7$

$D1 = 1;\ D2 = 0;\ D3 = 0;\ D4 = 0;$
$D5 = 0;\ D6 = 0;\ D7 = 0;\ D8 = 1;$
$ln(DepRate) = 2.437 - 0.197(0) - 0.585(0) + 0.033(0) - 0.179(0) - 0.196(0) - 0.301(0) - 0.211(0) + 0.329(1) + \Phi$
$\boldsymbol{ln(DepRate) = 2.766 + \Phi}$ *at timeframe* $= t_8$

The summary of all the substitutions are found in Table 18. The computed mean response ($DepRate$) given a dummy variable, $\widehat{\mu_{y|Di}}_{Comp}$, is compared to the actual mean response ($DepRate$) given a dummy variable, $\widehat{\mu_{y|Di}}_{Act}$. Theoretically, if Φ (numerical predictors and related interactions) does not interact with any of the dummy variables, then $\widehat{\mu_{y|Di}}_{Comp} \approx \widehat{\mu_{y|Di}}_{Act}$. However, as can be seen in Table 18, this is not the case. Because of this, the study will consider interactions of Φ and dummy variables, however, it will not be included in the simplified model. The interactions presented in this section only aims to graphically show how timeframe affects the overall performance of the model.

Table 18. *Timeframe Mean Response Summary*

Timeframe	*Dummy Variable*	$\widehat{\mu_{y\|Di}}_{Comp}$	$\widehat{\mu_{y\|Di}}_{Act}$
t_1	D1	2.240	2.319
t_2	D2	1.852	1.902
t_3	D3	2.469	2.614
t_4	D4	2.257	2.335
t_5	D5	2.241	2.247
t_6	D6	2.136	2.256
t_7	D7	2.225	2.267
t_8	D8	2.766	2.720
t_9	-	2.437	2.657

Model 8a – Dummy and Φ Interactions

$$ln(DepRate) = D1 + D2 + D3 + D4 + D5 + D6 + D7 + D8 + \Phi D1 + \Phi D2 + \Phi D3 + \Phi D4 + \Phi D5 + \Phi D6 + \Phi D7 + \Phi D8 + \Phi$$

$$\Phi = KerFeed + ReactordP + ClayOP + KerFeed{:}\,ClayOP + ReactordP{:}\,ClayOP + KerFeed{:}\,ReactordP$$

$$ln(DepRate) = 2.439 - 0.218D1 - 0.584D2 + 0.036D3 - 0.177D4 - 0.199D5 - 0.282D6 - 0.206D7 + 0.325D8 + 0.991\Phi + 0.114\Phi D1 - 0.005\Phi D2 - 0.019\Phi D3 - 0.018\Phi D4 + 0.019\Phi D5 - 0.087\Phi D6 - 0.046\Phi D7 + 0.100\Phi D8$$

Substituted Equations

$ln(DepRate) = 2.221 + 1.105\Phi \; at \; t_1$
$ln(DepRate) = 1.855 + 0.986\Phi \; at \; t_2$
$ln(DepRate) = 2.475 + 0.972\Phi \; at \; t_3$
$ln(DepRate) = 2.262 + 0.973\Phi \; at \; t_4$
$ln(DepRate) = 2.240 + 1.010\Phi \; at \; t_5$
$ln(DepRate) = 2.157 + 0.904\Phi \; at \; t_6$
$ln(DepRate) = 2.223 + 0.945\Phi \; at \; t_7$
$ln(DepRate) = 2.764 + 1.091\Phi \; at \; t_8$
$ln(DepRate) = 2.439 + 0.991\Phi \; at \; t_9 - Reference\ Point$

The reference point chosen in this study is timeframe 9: June 8, 2019 to January 10, 2020. No dummy variable was assigned for this due to linearity issue as previously discussed. Nevertheless, for the purpose of the trend discussion of Figure 12 and 13, it will be called $D9$.

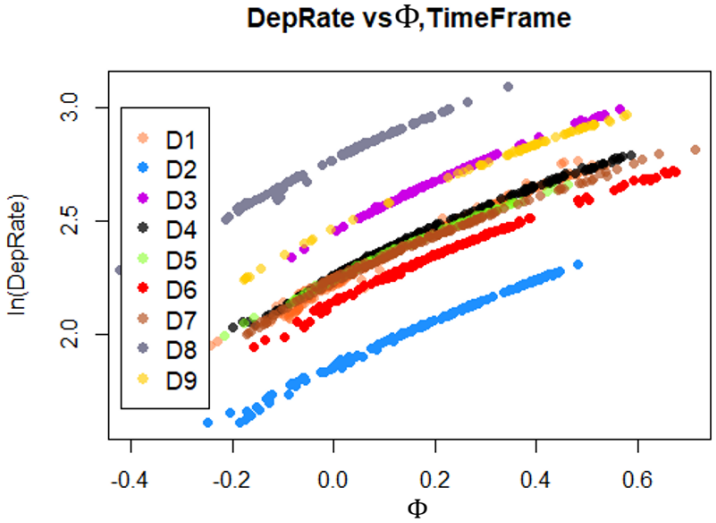


Figure 12. *DepRate vs Φ and Timeframe*

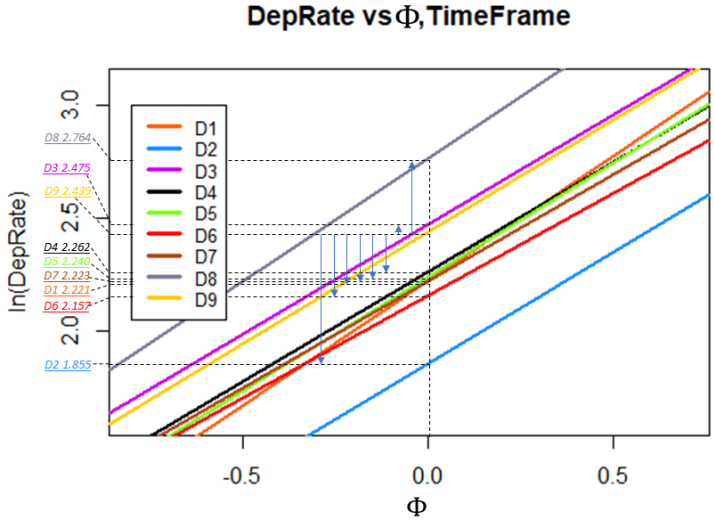


Figure 13. *Substituted Equations Form*

Notice that for all timeframes, $DepRate$ increases with fixed slopes as Φ increases. Overall, the slopes for all of the timeframes are almost the same with reference point, minimum and maximum of 0.991, 0.904, and 1.105, respectively. This is equivalent to 2.469 – 3.019 mmpd average increase in $DepRate$ per one unit increase in the net effect numerical predictors, Φ.

Moreover, it can be seen that on average, the $DepRate$ per day is 2.439 mm during timeframe 9 if the effect of numerical variable is zero. This inference about the intercept counters the initial assumption that all shutdown periods constitutes zero depletion rate. Nonetheless, the reference point intercept will help explain how other timeframes behaves relative to it.

For instance, it can be inferred that timeframes $D3$ and $D8$ will have a higher average $DepRates$ than $D9$ by 0.036 and 0.325 mmpd, respectively. Conversely, timeframes $D4$, $D5$, $D7$, $D1$, $D6$, and $D2$ will deviate lower than the average $DepRate$ of $D9$ by 0.177, 0.199, 0.206, 0.218, 0.282, and 0.584 mmpd, respectively. It can also be generalized that timeframes $D4$, $D5$, $D7$, $D1$ and $D6$ somehow performs similarly as can be seen by their overlapping lines. Same goes with timeframes $D3$ and $D9$. Furthermore, timeframes $D8$ and $D2$ seems to be in the opposite extremes of all the data.

The observations discussed above can help process engineers troubleshoot potential operational problems. Queries like: What happened during timeframe $D8$ where salt depletion is heightened?; Is the problem crude related?; Are there any APS-related issues?; will help initiating investigations. For now, for future predictions, it is deemed conservative to assume that it falls in category $D9$. This timeframe will lead to higher average $DepRate$ predictions than most of the timeframes ($D4$, $D5$, $D7$, $D1$, $D6$ and $D2$) and it also represents the recent state of the unit as it is the most recent timeframe.

Going to the interaction terms, it was found that by the inclusion of interaction terms, the effect of Φ is somehow altered through different timeframes as apparent with the changes in slopes. A higher slope constitutes a higher response value and vice versa. Hence, including these interaction terms may somehow increase the accuracy of prediction. However, this is clearly a sign of overfitting and settling with a simpler version (Model 8) will not drastically lower the error. Finally, it is also important to note that despite Φ having a positive net effect due to its positive slope, the involved predictors: $KerFeed$, $ReactordP$ and $ClayOP$ has their individual effects on $DepRate$ as discussed previously. Refer to Figure 14 , 15 and 16 to appreciate this.

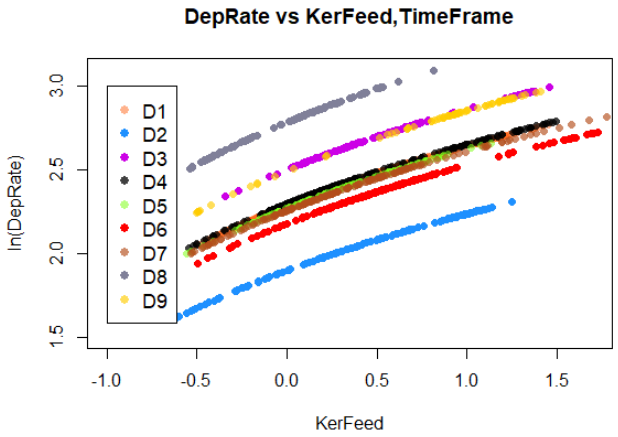


Figure 14. *DepRate vs KerFeed and Timeframe*

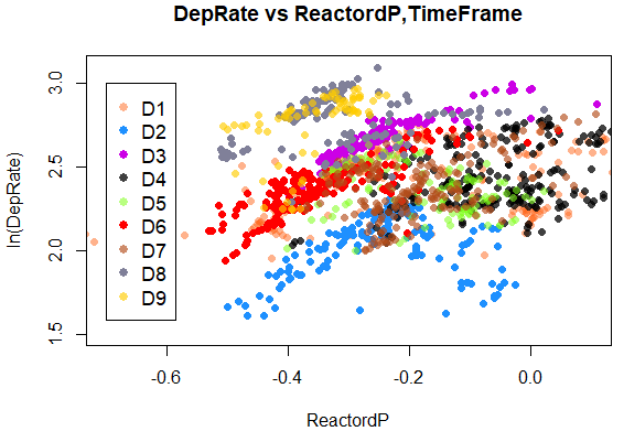


Figure 15. *DepRate vs ReactordP and Timeframe*

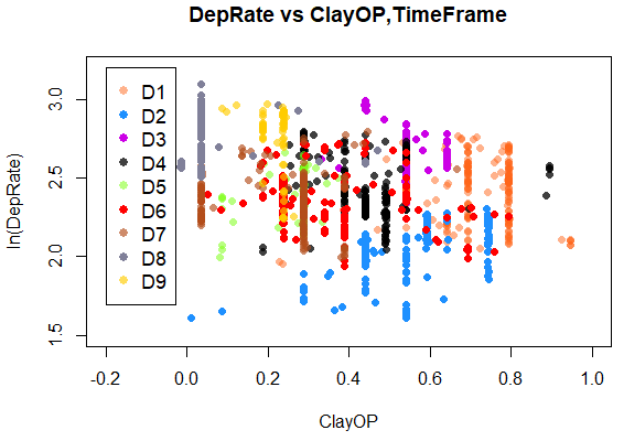


Figure 16. *DepRate vs ReactordP and Timeframe*

4. *Interaction Terms*

The need for effect modification is a result of two predictors having an effect on each other. That is, interaction terms are included in a linear model to correct and modify the linear relatonship between correlated predictors. Refer to Figure 17 for the correlation plots of $KerFeed$, $ReactordP$ and $ClayOP$.

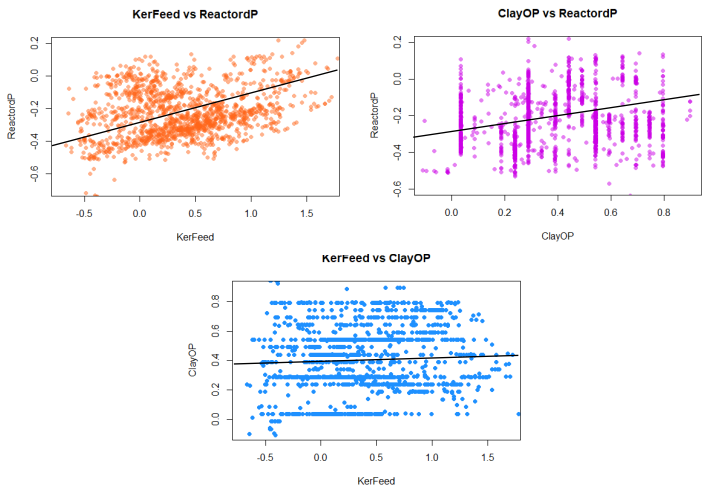


Figure 17. *Numerical Predictors Correlation Plot*

From Figure 17, it appears that the chosen predictors are correlated as previously tackled in previous sections. Thus, interaction terms to correct these correlations are incorporated in Model 8 as summarized below.

Model 8 – Simplified Model 7:

$$ln(DepRate) = 2.437 - 0.197D1 - 0.585D2 + 0.033D3 - 0.179D4 - 0.196D5 - 0.301D6 - 0.211D7 + 0.329D8 + 0.407KerFeed - 0.021ReactordP + 0.073ClayOP - 0.097KerFeed{:}ClayOP + 0.035ReactordP{:}ClayOP - 0.030KerFeed{:}ReactordP$$

$$KerFeed{:}ClayOP = KerFeed \times ClayOP$$
$$ReactordP{:}ClayOP = ReactordP \times ClayOP$$
$$KerFeed{:}ReactordP = KerFeed \times ReactordP$$

To understand this easier, let Z be the net effect of a selected pair of predictors. Visually, it can be seen in Figure 18, 19, 20, and 21 that interaction terms considered does alter the prediction of $DepRate$. Overall, the interaction terms increased the model slope which led to higher predicted values.

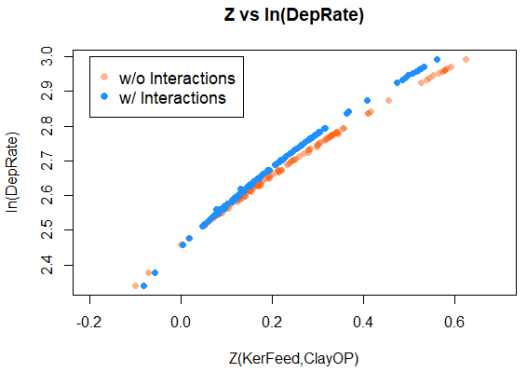


Figure 18. *KerFeed and ClayOP Interaction Effect*

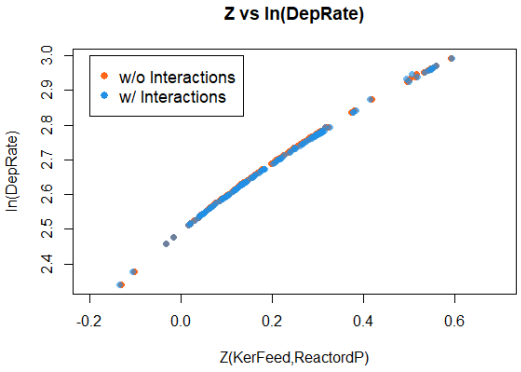


Figure 19. *KerFeed and ReactordP Interaction Effect*

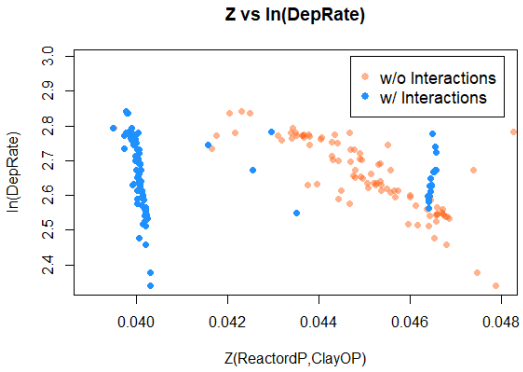


Figure 20. *ReactordP and ClayOP Interaction Effect*

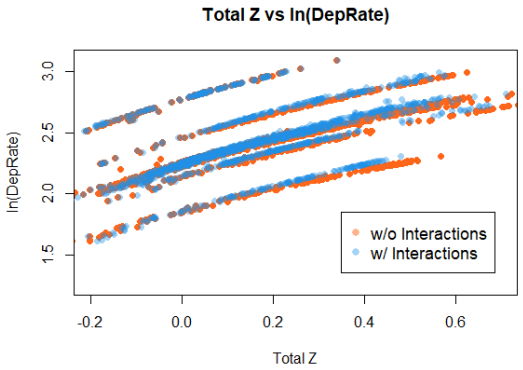


Figure 21. *Total Interaction Effects*

5. *Relative Importance*

The relative importance of predictors in Model 8 are visually presented in Figure 22. The normalized contribution are also summarized in Table 19 in descending order. Clearly, $KerFeed$ is the most important predictor with 39.86% relative importance. This is followed by the timeframes with lowest and highest actual $DepRate$ in the study, $D2$ and $D8$, respectively. Essentially, this means that if these timeframe dummy variables are removed, the predictions will be less accurate when the actual value is extremely high or extemely low.

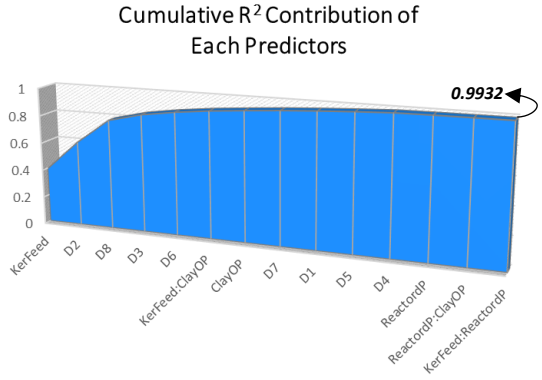


Figure 22. *KerFeed and ReactordP Interaction Effect*

Table 19. *Normalized Relative Importance of Predictors*

Predictors, x_i	***Relative Importance***
$KerFeed$	39.86%
$D2$	21.83%
$D8$	17.92%
$D3$	5.88%
$D6$	3.58%
$KerFeed:ClayOP$	2.71%
$ClayOP$	2.00%
$D7$	1.85%
$D1$	1.48%
$D5$	1.28%
$D4$	1.10%
$ReactordP$	0.26%
$ReactordP:ClayOP$	0.13%
$KerFeed:ReactordP$	0.11%

Model Accuracy

To make accuracy comparison of models discussed previously, an additional benchmark (Model 9) will be studied and included. The intricasies of model building will still follow the same procedure but will not be discussed thoroughly. Its final form is summarized below.

Model 9 – Model Without Timeframe Dummy:

$$\begin{aligned} ln(DepRate) = {} & 2.364 + 0.307KerFeed + 0.046KerTemp \\ & + 0.079BrineLev - 0.026WaterpH \\ & - 0.116ReactordP + 0.226ClaydP \\ & - 0.306ClayOP - 0.098KerFeed:ClaydP \\ & + 0.338KerFeed:ClayOP \\ & + 0.045KerFeed:BrineLev \\ & + 0.150ReactordP:ClayOP \\ & - 0.638ClaydP:ClayOP \end{aligned}$$

The models generated from this study are summarized again in Table 21 for easy reference in the discussion part of this section. The performance metrics of these models are also summarized again in Table 22. From this, it can be understood that the omission of timeframe as a dummy variable drastically reduced the viability of Model 9 to represent $DepRate$. The performance metrics measuring the error of prediction are also relatively higher than Models 3 to 8.

The accuracy of the models will not be discussed on a per point basis. It will alternatively be on a per timeframe basis since it is more meaningful for this study. To condense the data presentation, let Ψ represent the total depletion rate per timeframe predicted using the typical depletion rate declared by UOP which is 20 lbs salt lost per MB of kerosene feed. Moreover, let Ω represent the total depletion rate based on moisture differential across D-4404. The days without moisture differential results from the laboratory were filled with the typical depletion rate, 20 lbs/MB.

Table 21. *Model Summary*

Models	***Description***
Model 0	SLRM with unstandardized predictors
Model 3	SLRM Trans Without Shutdowns
Model 4	Model 3 Without Influential Points
Model 5	Model 3 With Interactions
Model 5a	Model 4 With Interactions
Model 6	Model 5a SPLR
Model 7	Model 6 SPLR without VIFs > 10
Model 8	Simplified Model 7
Model 9	Model Without Timeframe Dummy

Table 22. *Perfomance Metrics Summary*

	Model 3	***Model 4***	***Model 5***	***Model 5a***
R^2	0.9610	0.9889	0.9909	0.9975
R^2_{adj}	0.9604	0.9887	0.9907	0.9975
RSE	0.0635	0.0292	0.0308	0.0138
MAE	0.9738	0.9870	1.0151	0.9947
MSE	1.1734	1.0698	1.0803	1.0128
$RMSE$	1.0832	1.0343	1.0394	1.0064
	Model 6	***Model 7***	***Model 8***	***Model 9***
R^2	0.9975	0.9946	0.9932	0.6633
R^2_{adj}	0.9975	0.9946	0.9932	0.6604
RSE	0.0138	0.0203	0.0227	0.1600
MAE	0.9947	0.9995	0.9973	1.6875
MSE	1.0128	1.0631	1.0684	4.7602
$RMSE$	1.0064	1.0311	1.0336	2.1818

Table 23. *Actual and Predicted Salt Depletion Per Timeframe*

				Models		
t	**Actual**	**Ψ**	**Ω**	**0**	**3**	**4**
1	2780.00	1861.69	1728.54	2730.00	3051.19	3047.36
2	1190.00	1242.96	1027.57	1764.90	1379.16	1370.87
3	1945.00	1029.57	922.62	1555.71	2079.76	2093.68
4	2447.00	1622.42	1397.09	2500.72	2690.51	2684.72
5	1240.00	848.50	726.13	1272.06	1370.90	1365.02
6	2061.00	1560.65	1330.95	2523.73	2284.75	2282.07
7	1987.80	1840.43	1143.76	2230.98	2186.47	2195.21
8	2260.00	1097.45	797.31	1506.99	2389.58	2406.08
9	1201.00	859.79	541.69	1026.70	1270.05	1284.87

t	5	5a	6	7	8	9
1	3049.92	3045.97	3045.88	3044.81	3037.85	3015.61
2	1366.41	1364.69	1364.65	1365.06	1366.28	1736.17
3	2079.31	2087.90	2087.85	2089.19	2092.16	1762.42
4	2682.32	2679.21	2679.17	2681.25	2683.67	2586.85
5	1370.79	1369.29	1369.29	1368.06	1367.45	1384.58
6	2279.81	2276.75	2276.79	2281.45	2282.40	2501.19
7	2189.68	2191.88	2191.96	2196.07	2199.35	2383.11
8	2402.30	2406.06	2406.12	2406.58	2406.64	1985.35
9	1290.36	1291.04	1291.10	1292.19	1291.26	1155.32

Table 24. *Model Percent Errors Per Timeframe*

			Models			
t	**Ψ**	**Ω**	**0**	**3**	**4**	**5**
1	33.03%	37.82%	1.80%	9.76%	9.62%	9.71%
2	4.45%	13.65%	48.31%	15.90%	15.20%	14.82%
3	47.07%	52.56%	20.01%	6.93%	7.64%	6.91%
4	33.70%	42.91%	2.20%	9.95%	9.71%	9.62%
5	31.57%	41.44%	2.59%	10.56%	10.08%	10.55%
6	24.28%	35.42%	22.45%	10.86%	10.73%	10.62%
7	7.41%	42.46%	12.23%	9.99%	10.43%	10.16%
8	51.44%	64.72%	33.32%	5.73%	6.46%	6.30%
9	28.41%	54.90%	14.51%	5.75%	6.98%	7.44%
Ave	**29.04%**	**42.88%**	**17.49%**	**9.49%**	**9.65%**	**9.57%**

t	5a	6	7	8	9
1	9.57%	9.56%	9.53%	9.28%	8.48%
2	14.68%	14.68%	14.71%	14.81%	45.90%
3	7.35%	7.34%	7.41%	7.57%	9.39%
4	9.49%	9.49%	9.57%	9.67%	5.72%
5	10.43%	10.43%	10.33%	10.28%	11.66%
6	10.47%	10.47%	10.70%	10.74%	21.36%
7	10.27%	10.27%	10.48%	10.64%	19.89%
8	6.46%	6.47%	6.49%	6.49%	12.15%
9	7.50%	7.50%	7.59%	7.52%	3.80%
Ave	**9.58%**	**9.58%**	**9.64%**	**9.67%**	**15.37%**

From Table 24, it can be perceived that both of the existing ways to anticipate the amount of salt depletion, Ψ and Ω, are both erroneous with average prediction errors of 29.04% and 42.88%, respectively. As with the assumptions of this study, using the moisture difference across D-4404 is deemed to be an erroneous way of calculating for depletion.

Moreover, notice that Model 0 and Model 9 may have relatively lower percentage errors than Model Ψ and Model Ω, but due to the existing nonconformities in the classical assumptions, their percentage errors are still much higher than Models 3 to 8. Also notice that Model 0 has predictions with low percentage errors like timeframe 1, 4 and 5. It may be argued that Model 0 performed better in these timeframes however, it should not be missed that a linear model should be able to represent the whole dataset to be deemed as a reliable tool for prediction. Same goes with Model 9.

On the other hand, Models 3 to 8 performed quite alike in terms of percentage errors per timeframe as with the previous discussions involving their performance metrics. The simplified final model in this study, Model 8, has a maximum percentage error of 14.81% (1366 mmpd predicted vs 1190 mmpd actual) which happened in timeframe 2. Nonetheless, the model performed well and gave conservative predictions with 9.67% average positive error.

V. Conclusions and Recommendations

With all the analysis and the discussions aforementioned, the study drew to close that Model 8 – $\ln(DepRate) = 2.437 + 0.407KerFeed - 0.021ReactordP + 0.073ClayOP - 0.907(KerFeed \times ClayOP) + 0.035(ReactordP \times ClayOP) - 0.030(KerFeed \times ReactordP)$, with the assumption that future values are conservatively represented by timeframe 9, effectively predicts salt depletion rate of D-4404 with average prediction error of 9.67%. Overall, Model 8 conforms to all the classical assumptions of multiple linear regression and has satisfactory performance metrics and accuracy.

It is prescribed to apply predictive analytics through multiple linear regression to other process units in Petron Bataan Refinery. The theories and concepts discussed herewith shall hopefully serve as basis in building more models projecting refinery operations. It is also recommended to explore advanced techniques like Principal Component Analysis to gather more meaningful inferences and insights. It will also be fruitful to build an analysis tool for incipient fault detection for instruments since most the data inside the refinery are often sourced out from instrument transmitters.

VI. References

Wooldridge, J., *Introductory Econometrics: A Modern Approach 5th ed.*, South-western CENGAGE Learning, 2013.

Gujarati, D., *Econometrics By Example*, Palgrave Macmillan, 2011.

Draper, N., Smith, H., *Applied Regression Analysis, 3rd ed.*, John Wiley and Sons: Canada, 1998.

Kutner M., et al., *Applied Linear Statistical Models 5th ed.*, McGraw Hill/Irwin: New York, 2005.

Daquis, J.C., Applied Regression Analysis: *Statistics 223 Lecture Notes*, University of the Philippines Diliman – School of Statistics, 2019.

Jalao, E.R., Predictive Analytics – *Regression Methodologies*, University of the Philippines Diliman – National Engineering Center, 2019.

Jalao, E.R., Descriptive Analytics – *Descriptive Analystics with R*, University of the Philippines Diliman – National Engineering Center, 2019.

Kim, B., Understanding Diagnostic Plots of Linear Regression Analysis, University of Virginia Library, 2015, https://data.library.virginia.edu/diagnostic-plots/.

Lillis, D., Linear Models in R: *Diagnosing Our Regression Model*, The Analysis Factor. https://www.theanalysisfactor.com/linear-models-r-diagnosing-regression-model/.

Dalpiaz, D., Applied Statistics with R: *Transformations*, University of Illinois Urbana – Champaign https://daviddalpiaz.github.io/appliedstats/transformations.html.

Mangiafico, S., Summary and Analysis of Extension Program Evaluation in R: *Transforming Data*, https://rcompanion.org/handbook/I_12.html.

Bruce, Peter, and Andrew, B., Practical Statistics for Data Scientists. O'Reilly Media, 2017.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani, An Introduction to Statistical Learning: With Applications in R. Springer Publishing Company, Incorporated, 2014.

Measures of Influence, Comprehensive R Archive Network (CRAN), https://cran.r-project.org/web/packages/olsrr/vignettes/influence_measures.html.

Statistical tools for high-throughput data analysis – *Correlation matrix : A quick start guide to analyze, format and visualize a correlation matrix using R software*, http://www.sthda.com/english/wiki/correlation-matrix-a-quick-start-guide-to-analyze-format-and-visualize-a-correlation-matrix-using-r-software.

Raschka, S., About Feature Scaling and Normalization – *and the effect of standardization for machine learning algorithms*, https://sebastianraschka.com/Articles/2014_about_feature_scaling.html, 2014.

Distinction Between Outliers and High Leverage Observations, Pennsylvania State University, Eberly College of Science, Department of Statistics, STAT 501 Lectures.

VII. Appendices

SALT TOPPING HISTORY

	Commissioning December 2014	December 2014 to October 1,2015	October 2, 2015 to April 1, 2016	April 2, 2016 to September 8, 2016
	December 2014	**October 2015**	**April 2016**	**September 2016**
Salt height from last loading (from top manway), mm	1,920.00	1,920.00	1,920.00	1,745.00
As found Salt height (from top manway), mm	-	4,700.00	3,110.00	3,690.00
Salt Lost, mm	-	2,780.00	1,190.00	1,945.00
Salt height after topping (from top manway), mm	-	1,920.00	1,745.00	1,753.00
Salt Density Loaded	1,343.00	1,134.82	1,233.34	1,536.48
Remarks	-	22,300 kg loaded	11,900 kg loaded	

	September 9, 2016 to April 29, 2017	April 30, 2017 to September 27, 2017	September 28, 2017 to May 7, 2018	May 8, 2018 to November 26, 2018
	April 2017	**September 2017**	**May 2018**	**November 2018**
Salt height from last loading (from top manway), mm	1,753.00	1,940.00	1,890.00	1,920.00
As found Salt height (from top manway), mm	4,200.00	3,180.00	3,951.00	3,907.80
Salt Lost, mm	2,447.00	1,240.00	2,061.00	1,987.80
Salt height after topping (from top manway), mm	1,940.00	1,890.00	1,920.00	1,860.00
Salt Density Loaded	1,347.76	1,005.54	1020.46	1011.99
Remarks			14,650 kg loaded	14,650 kg loaded

	November 27, 2018 to June 7, 2019	June 7, 2019 to January 10, 2020
	June 2019	**January 2020**
Salt height from last loading (from top manway), mm	1,860.00	1,872.00
As found Salt height (from top manway), mm	4,120.00	3,073.00
Salt Lost, mm	2,260.00	1,201.00
Salt height after topping (from top manway), mm	1,872.00	1,920.00
Salt Density Loaded	1,140.84	1,140.84
Remarks	Complete Changeout 629 bags	No recorded number of bags, density assumed similar to previous

CORRELATION

row	column	corr	p	Makes Sense?
ReactorIP	ClayOP	0.7186	0.000	Yes
KerFeed	ReactorIP	0.7170	0.000	Yes
ReactorIP	ClaydP	0.4924	0.000	Yes
KerFeed	ClaydP	0.4006	0.000	Yes
KerFeed	ClayOP	0.3546	0.000	Yes
ReactorIP	ReactordP	0.2244	0.000	Yes
ReactordP	ClayOP	0.1867	0.000	Yes
KerFeed	KerTemp	0.1738	0.000	Yes
KerFeed	ReactordP	0.1636	0.000	Yes
WaterLev	WaterpH	-0.1533	0.000	Yes
KerTemp	ReactorIP	0.1118	0.000	Yes
ReactordP	ClaydP	0.0451	0.087	Yes
KerFeed	BrineLev	0.0422	0.110	Yes
KerTemp	ClayOP	0.0336	0.203	Yes
ClaydP	ClayOP	0.0122	0.643	Yes
KerTemp	BrineLev	-0.21584	0	No
WaterLev	BrineLev	-0.31707	0	No
WaterLev	ECPdP	0.483735	0	No
BrineLev	ECPdP	-0.31693	0	No
ECPLev	ReactorIP	0.220777	0	No
BrineLev	ReactordP	0.227344	0	No
BrineLev	ReactorCauLev	0.301241	0	No
WaterLev	ClaydP	0.290841	0	No
WaterpH	ClaydP	0.317626	0	No
WaterLev	ClayOP	-0.26714	0	No
BrineLev	ClayOP	0.439238	0	No
ECPLev	ClayOP	0.254458	0	No
ECPdP	ReactordP	-0.19993	1.91E-14	No
WaterpH	ReactorIP	0.184484	1.72E-12	No
ECPdP	ECPLev	-0.17836	9.29E-12	No
KerFeed	ECPdP	0.172182	4.79E-11	No
WaterLev	ReactordP	-0.15962	1.12E-09	No
ECPdP	ClaydP	0.143285	4.74E-08	No
WaterLev	ECPLev	-0.13191	5.08E-07	No
WaterLev	WCBootLev	0.125561	1.76E-06	No
KerFeed	WCBootLev	-0.1237	2.50E-06	No
WCBootLev	ClaydP	0.120985	4.14E-06	No
ECPdP	ReactorIP	0.116598	9.17E-06	No
KerTemp	WaterpH	-0.11379	1.50E-05	No
KerFeed	WaterpH	0.110869	2.48E-05	No
BrineLev	ECPLev	0.110243	2.76E-05	No
WaterpH	ClayOP	0.109142	3.32E-05	No
KerFeed	ECPLev	0.102149	0.000103136	No
BrineLev	WCBootLev	-0.0947	0.000320125	No
BrineLev	ReactorIP	0.093363	0.000388836	No
WaterpH	WCBootLev	0.09126	0.000525725	No
ReactorCauLev	ClayOP	0.089899	0.000636891	No

ReactorCauLev	ClaydP	0.087111	0.000936085	No
ECPLev	ReactordP	0.082368	0.001758483	No
BrineLev	ClaydP	-0.08136	0.002001774	No
KerFeed	ReactorCauLev	0.080973	0.002104423	No
KerTemp	ECPLev	0.078795	0.002770602	No
WaterpH	ECPdP	-0.0753	0.004248897	No
KerTemp	ClaydP	-0.07335	0.005355837	No
KerTemp	ReactorCauLev	-0.07307	0.00553493	No
BrineLev	WaterpH	0.066721	0.011325369	No
ECPLev	ReactorCauLev	-0.06299	0.016828252	No
KerTemp	ECPdP	0.062226	0.018199545	No
WCBootLev	ReactorIP	-0.06177	0.019063834	No
WaterLev	ReactorIP	-0.05698	0.030615027	No
ECPdP	ClayOP	-0.05628	0.032712306	No
WCBootLev	ReactorCauLev	-0.05028	0.056460169	No
WCBootLev	ClayOP	-0.04521	0.086328797	No
WCBootLev	ECPdP	0.044111	0.094274614	No
WCBootLev	ECPLev	0.039298	0.136087539	No
WaterpH	ReactorCauLev	-0.03679	0.162944226	No
WaterpH	ECPLev	0.032887	0.21231597	No
WaterLev	ReactorCauLev	0.032433	0.218700564	No
ReactorIP	ReactorCauLev	0.031039	0.23914858	No
WaterpH	ReactordP	0.027526	0.296559665	No
WCBootLev	ReactordP	0.025939	0.325289852	No
KerTemp	ReactordP	-0.02544	0.334735104	No
ReactordP	ReactorCauLev	-0.0205	0.437070776	No
ECPLev	ClaydP	0.01954	0.458751201	No
KerFeed	WaterLev	-0.01368	0.603856965	No
ECPdP	ReactorCauLev	0.01054	0.689416249	No
KerTemp	WCBootLev	0.005345	0.839419452	No
KerTemp	WaterLev	0.004273	0.871311236	No

R CODES

```
#LOAD NECESSARY LIBRARIES
library(zoo)
library(lmtest)
library(MASS)
library(carData)
library(car)
library(lattice)
library(ggplot2)
library(caret)
library(pls)
library(caret)
library(dplyr)
library(olsrr)
library(Hmisc)
library(corrplot)
library(rcompanion)
library(DescTools)
library(effects)
library(psych)
library(relaimpo)
```

```
Population = read.csv("SaltDepRateData.csv")
PopulationWithoutShutdowns = read.csv("SaltDepRateData3.csv")
set.seed(123456)
trainIndex = createDataPartition(Population$DepRate, p=0.80, list=FALSE)
TrainSet = Population[trainIndex,]
TestSet = Population[-trainIndex,]


trainIndex2 = createDataPartition(PopulationWithoutShutdowns$DepRate, p=0.90,
list=FALSE)
TrainSet2 = PopulationWithoutShutdowns[trainIndex2,]
TestSet2 = PopulationWithoutShutdowns[-trainIndex2,]


#MODEL TRANSFORMATIONS
#Simple Linear Regression Model = SLRM
SLRM=lm(DepRate~.,data=TrainSet)
summary(SLRM)
#1. Check for Normality of Residuals
qqnorm(SLRM$residuals, col="dodgerblue", pch = 20, xlab="Theoretical Quantiles",
ylab="Standardized Residuals", main="Model 1 - SLRM Normal Q-Q Plot")
qqline(SLRM$residuals,col="darkorange", lwd=2)
hist(SLRM$residuals, main = "Histogram of SLRM Residuals", xlab="SLRM Residuals",
breaks=75, col="dodgerblue", border="black")


#2. Check for constant variance
plot(fitted(SLRM), resid(SLRM), col = "dodgerblue", pch = 20,
     xlab = "Predicted Values", ylab = "Residuals", main = "Model 1 - SLRM")
abline(h = 0, col = "darkorange", lwd = 2)

#3. Transformation of Response Variable
#We used ln(DepRate+C) where C = 15 since there are zero values for deprate
SLRMTrans=lm(log1p(DepRate+16)~., data=TrainSet)
summary(SLRMTrans)

qqnorm(SLRMTrans$residuals, col="dodgerblue", pch = 20, xlab="Theoretical Quantiles",
ylab="Standardized Residuals", main="Model 2 - SLRM Trans Normal Q-Q Plot")
qqline(SLRMTrans$residuals,col="darkorange", lwd=2)

plot(fitted(SLRMTrans), resid(SLRMTrans), col = "dodgerblue", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Model 2 - SLRM Trans")
abline(h = 0, col = "darkorange", lwd = 2)

#Plot 2 - No Shutdowns
#No Additional constant since all values of DepRate is greater than zero
SLRMTransNoSD = lm(log1p(DepRate)~., data=TrainSet2)
summary(SLRMTransNoSD)

qqnorm(SLRMTransNoSD$residuals, col="dodgerblue", pch = 20, xlab="Theoretical
Quantiles", ylab="Standardized Residuals", main="Model 2 - SLRM Trans Normal Q-Q Plot")
qqline(SLRMTransNoSD$residuals,col="darkorange", lwd=2)

plot(fitted(SLRMTransNoSD), resid(SLRMTransNoSD), col = "dodgerblue", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Model 3 - SLRM Trans No SD")
abline(h = 0, col = "darkorange", lwd = 2)


#4. Check for Influential Points
LeveragePoint=abs(hatvalues(SLRMTransNoSD))
plot(LeveragePoint, col="dodgerblue", ylab="Leverage Statistic", xlab="DepRate
Observations", main = "Model 3 - Leverage Points")
```

```
abline(h=(2*(21+1)/1440), col="darkorange", lwd=2)
which(LeveragePoint>(2*(21+1)/1440),)

Outliers=abs(rstudent(SLRMTransNoSD))
plot(Outliers, col="dodgerblue", ylab="Studentized Residuals", xlab="DepRate
Observations", main = "Model 3 - Outliers")
abline(h=3, col="darkorange", lwd=2)
which(Outliers>3,)


ols_plot_resid_lev(SLRMTransNoSD)
InfluentialPlot=ols_plot_resid_lev(SLRMTransNoSD)
Influential=InfluentialPlot$data
write.table(Influential,"C:/Users/rpdlpascual/Desktop/Influential.txt",sep="\t" )
write.table(TrainSet2,"C:/Users/rpdlpascual/Desktop/TrainSet2.txt",sep="\t" )

TrainSet2NoInfluential = read.csv("SaltDepRateData5.csv")
SLRMTransNoSDNoInfluential = lm(log1p(DepRate)~., data=TrainSet2NoInfluential)
summary(SLRMTransNoSDNoInfluential)

qqnorm(SLRMTransNoSDNoInfluential$residuals, col="dodgerblue", pch = 20,
xlab="Theoretical Quantiles", ylab="Standardized Residuals")
qqline(SLRMTransNoSDNoInfluential$residuals,col="darkorange", lwd=2)

plot(fitted(SLRMTransNoSDNoInfluential), resid(SLRMTransNoSDNoInfluential), col =
"dodgerblue", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Model 4 - SLRM No Influential",
xlim=c(1.65,3), ylim=c(-0.2,0.2))
abline(h = 0, col = "darkorange", lwd = 2)

#5. Check for Multicollinearity
flattenCorrMatrix = function(cormat,pmat){
  ut = upper.tri(cormat)
  data.frame(
    row=rownames(cormat)[row(cormat)[ut]],
    column=rownames(cormat)[col(cormat)[ut]],
    cor=(cormat)[ut],
    p = pmat[ut]
  )
}
res=rcorr(as.matrix(TrainSet2[,10:22]))
Correlation=flattenCorrMatrix(res$r,res$P)
write.table(Correlation,"C:/Users/rpdlpascual/Desktop/Correlation.txt",sep="\t")
#Correlation Graph, insignificant correlations are crossed
corrplot(res$r, type="upper", order="hclust", p.mat=res$P, sig.level=0.05,
insig="blank")

#Include interactions with significant p-value (which conceptually makes sense)
SLRMTransNoSDWithInt=lm(log1p(DepRate)~ D1+D2+D3+D4+D5+D6+D7+D8+
                                KerFeed+KerTemp+WaterLev+BrineLev+WaterpH+WCBootLev+
                                ECPdP + ECPLev+ReactorIP+ReactordP + ReactorCauLev+
                                ClaydP+ClayOP+ReactorIP:ClayOP+KerFeed:ReactorIP+
                                ReactorIP:ClaydP+KerFeed:ClaydP+KerFeed:ClayOP+
                                ReactorIP:ReactordP+ReactordP:ClayOP+KerFeed:KerTemp+
                                KerFeed:ReactordP+WaterLev:WaterpH+KerTemp:ReactorIP+
                                ReactordP:ClaydP+KerFeed:BrineLev+KerTemp:ClayOP+
                                ClaydP:ClayOP, data=TrainSet2)
summary(SLRMTransNoSDWithInt)

SLRMTransNoSDNoInfWithInt=lm(log1p(DepRate)~ D1+D2+D3+D4+D5+D6+D7+D8+
                            KerFeed+KerTemp+WaterLev+BrineLev+WaterpH+WCBootLev+
                            ECPdP+ECPLev+ReactorIP+ReactordPReactorCauLev+
```

```
                            ClaydP+ClayOP+ReactorIP:ClayOP+KerFeed:ReactorIP+
                            ReactorIP:ClaydP+KerFeed:ClaydP+KerFeed:ClayOP+
                            ReactorIP:ReactordP+ReactordP:ClayOP+KerFeed:KerTemp+
                            KerFeed:ReactordP+WaterLev:WaterpH+KerTemp:ReactorIP+
                            ReactordP:ClaydP+KerFeed:BrineLev+KerTemp:ClayOP+
                            ClaydP:ClayOP, data=TrainSet2NoInfluential)
summary(SLRMTransNoSDNoInfWithInt)

#MODEL ACCURACY AND SPLR
#Model 3
summary(SLRMTransNoSD)
PredictModel3 = exp(predict(SLRMTransNoSD,TestSet2))
MAE(PredictModel3, TestSet2$DepRate)
MSE(PredictModel3, TestSet2$DepRate)
RMSE(PredictModel3, TestSet2$DepRate)

#Model4
summary(SLRMTransNoSDNoInfluential)
PredictModel4 = exp(predict(SLRMTransNoSDNoInfluential,TestSet2))
MAE(PredictModel4, TestSet2$DepRate)
MSE(PredictModel4, TestSet2$DepRate)
RMSE(PredictModel4, TestSet2$DepRate)

#Model5
summary(SLRMTransNoSDWithInt)
PredictModel5 = exp(predict(SLRMTransNoSDWithInt,TestSet2))
MAE(PredictModel5, TestSet2$DepRate)
MSE(PredictModel5, TestSet2$DepRate)
RMSE(PredictModel5, TestSet2$DepRate)

#Model5a
summary(SLRMTransNoSDNoInfWithInt)
PredictModel5a = exp(predict(SLRMTransNoSDNoInfWithInt,TestSet2))
MAE(PredictModel5a, TestSet2$DepRate)
MSE(PredictModel5a, TestSet2$DepRate)
RMSE(PredictModel5a, TestSet2$DepRate)


#Model6 - Stepwise Model 5a
ModelStep = step(SLRMTransNoSDNoInfWithInt)
SLRMTransNoSDNoInfWithIntStepReduced = lm(log1p(DepRate) ~ D1+D2+D3+D4+D5+D6+D7+D8+
                              KerFeed+KerTemp+WaterLev+BrineLev+WaterpH+WCBootLev+
                              ECPdP+ReactorIP+ReactordP+ClaydP+ClayOP+KerFeed:ReactorIP
                              +ReactorIP:ClaydP+KerFeed:ClaydP+KerFeed:ClayOP+
                              ReactorIP:ReactordP+ReactordP:ClayOP+KerFeed:KerTemp+
                              KerFeed:ReactordP+KerTemp:ReactorIP+KerFeed:BrineLev+
                              KerTemp:ClayOP + ClaydP:ClayOP, data =
                              TrainSet2NoInfluential)
summary(SLRMTransNoSDNoInfWithIntStepReduced)

Model6=SLRMTransNoSDNoInfWithIntStepReduced$coefficients
write.table(Model6,"C:/Users/rpdlpascual/Desktop/Model6.txt",sep="\t")
PredictModel6 = exp(predict(SLRMTransNoSDNoInfWithIntStepReduced,TestSet2))
MAE(PredictModel6, TestSet2$DepRate)
MSE(PredictModel6, TestSet2$DepRate)
RMSE(PredictModel6, TestSet2$DepRate)


#Check VIF of Model 6 - Model 7
VIF(SLRMTransNoSDNoInfWithIntStepReduced)

#Remove ReactorIP with VIF of 19.311
```

```
SLRMTransNoSDNoInfWithIntStepReduced2 = lm(log1p(DepRate)~D1+D2+D3+D4+D5+D6+D7+D8+
                                KerFeed+KerTemp+WaterLev+BrineLev+WaterpH+WCBootLev+ECPdP
                                +ReactordP+ClaydP+ClayOP+KerFeed:ClaydP+KerFeed:ClayOP+
                                ReactordP:ClayOP+KerFeed:KerTemp+KerFeed:ReactordP+
                                KerFeed:BrineLev+ClayOP:KerTemp+ClaydP:ClayOP, data =
                                TrainSet2NoInfluential)
summary(SLRMTransNoSDNoInfWithIntStepReduced2)
VIF(SLRMTransNoSDNoInfWithIntStepReduced2)


#Remove ClaydP with VIF of 13.56
SLRMTransNoSDNoInfWithIntStepReduced3 = lm(log1p(DepRate)~D1+D2+D3+D4+D5+D6+D7+D8+
                                KerFeed+KerTemp+WaterLev+BrineLev+WaterpH+WCBootLev+
                                ECPdP+ReactordP+ClayOP+KerFeed:ClayOP+ReactordP:ClayOP+
                                KerFeed:KerTemp+KerFeed:ReactordP+KerFeed:BrineLev+
                                ClayOP:KerTemp, data = TrainSet2NoInfluential)
summary(SLRMTransNoSDNoInfWithIntStepReduced3)
VIF(SLRMTransNoSDNoInfWithIntStepReduced3)


ModelStep2=step(SLRMTransNoSDNoInfWithIntStepReduced3)

#Model 7
SLRMTransFinalModel = lm(log1p(DepRate)~D1+D2+D3+D4+D5+D6+D7+D8+KerFeed+KerTemp+
                                BrineLev+WCBootLev+ECPdP+ReactordP+ClayOP+KerFeed:ClayOP+
                                ReactordP:ClayOP+KerFeed:KerTemp+KerFeed:ReactordP+
                                KerFeed:BrineLev + KerTemp:ClayOP, data =
                                TrainSet2NoInfluential)
summary(SLRMTransFinalModel)
VIF(SLRMTransFinalModel)


Model7=SLRMTransFinalModel$coefficients
write.table(Model7,"C:/Users/rpdlpascual/Desktop/Model7Coeffs.txt",sep="\t")


PredictModel7 = exp(predict(SLRMTransFinalModel,TestSet2))
MAE(PredictModel7, TestSet2$DepRate)
MSE(PredictModel7, TestSet2$DepRate)
RMSE(PredictModel7, TestSet2$DepRate)


#Model 8 - Simplified Model 7
SLRMTransFinalModelSimplified = lm(log1p(DepRate)~D1+D2+D3+D4+D5+D6+D7+D8+KerFeed+
                                ReactordP+ClayOP+KerFeed:ClayOP+ReactordP:ClayOP+
                                KerFeed:ReactordP, data = TrainSet2NoInfluential)
summary(SLRMTransFinalModelSimplified)
Model8=SLRMTransFinalModelSimplified$coefficients
write.table(Model8,"C:/Users/rpdlpascual/Desktop/Model8Coeffs.txt",sep="\t")

VIF(SLRMTransFinalModelSimplified)
PredictModel8 = exp(predict(SLRMTransFinalModelSimplified,TestSet2))
MAE(PredictModel8, TestSet2$DepRate)
MSE(PredictModel8, TestSet2$DepRate)
RMSE(PredictModel8, TestSet2$DepRate)


#DUMMY VARIABLE INTERPRETATION
#Model 8A
DummyVarInt=read.csv("SaltDepRateData7.csv")
DumVarMod = lm(log1p(DepRate)~D1+D2+D3+D4+D5+D6+D7+D8+Theta+
                                D1:Theta+D2:Theta+D3:Theta+
```

```
                                D4:Theta+D5:Theta+D6:Theta+
                                D7:Theta+D8:Theta, data=DummyVarInt)
summary(DumVarMod)
Model8a=DumVarMod$coefficients
write.table(Model8a,"C:/Users/rpdlpascual/Desktop/Model8aCoeffs.txt",sep="\t")
attach(DummyVarInt)

#Edit SaltDepRateData7 and Make zeroes in D8 to 2 to be able to detect D9
plot(Theta[D1=="1"], log1p(DepRate)[D1=="1"], col=rgb(1,0.4,0.1,0.5), xlab="Phi",
ylab="ln(DepRate)",
     main="DepRate vs P,TimeFrame", pch=16, ylim=c(1.6,3.1), xlim=c(-0.4,0.7))
points(Theta[D2=="1"], log1p(DepRate)[D2=="1"],col="dodgerblue", pch=16)
points(Theta[D3=="1"], log1p(DepRate)[D3=="1"],col=rgb(0.8,0,0.9), pch=16)
points(Theta[D4=="1"], log1p(DepRate)[D4=="1"],col=rgb(0,0,0,0.75), pch=16)
points(Theta[D5=="1"], log1p(DepRate)[D5=="1"],col=rgb(0.5,1,0.1,0.55),pch=16)
points(Theta[D6=="1"], log1p(DepRate)[D6=="1"],col=rgb(1,0,0), pch=16)
points(Theta[D7=="1"], log1p(DepRate)[D7=="1"],col=rgb(0.7,0.3,0.1,0.65), pch=16)
points(Theta[D8=="1"], log1p(DepRate)[D8=="1"],col=rgb(0.5,0.5,0.6), pch=16)
points(Theta[D8=="2"], log1p(DepRate)[D8=="2"],col=rgb(1,0.8,0,0.65), pch=16)
legend(-0.42,3.0,legend=c("D1","D2","D3","D4","D5","D6","D7","D8","D9"),
col=c(rgb(1,0.4,0.1,0.5),"dodgerblue",rgb(0.8,0,0.9),
                          rgb(0,0,0,0.75),rgb(0.5,1,0.1,0.55),
                          rgb(1,0,0),rgb(0.7,0.3,0.1,0.65),
                          rgb(0.5,0.5,0.6),rgb(1,0.8,0,0.65)),pch=16, cex=1.12)


#Substituted Equation Lines
plot(Theta[D1=="1"], log1p(DepRate)[D1=="1"], col=rgb(0,0,0,0), xlab="Phi",
ylab="ln(DepRate)",
     main="DepRate vs P,TimeFrame", pch=16, ylim=c(1.6,3.1), xlim=c(-0.8,0.7))

abline(a=2.221, b=1.105,col=rgb(1,0.4,0.1), lwd=2)
abline(a=1.855, b=0.986,col="dodgerblue", lwd=2)
abline(a=2.475, b=0.972,col=rgb(0.8,0,0.9), lwd=2)
abline(a=2.262, b=0.973,col=rgb(0,0,0), lwd=2)
abline(a=2.240, b=1.010,col=rgb(0.5,1,0.1), lwd=2)
abline(a=2.157, b=0.904,col=rgb(1,0,0), lwd=2)
abline(a=2.223, b=0.945,col=rgb(0.7,0.3,0.1), lwd=2)
abline(a=2.764, b=1.091,col=rgb(0.5,0.5,0.6), lwd=2)
abline(a=2.439, b=0.991,col=rgb(1,0.8,0), lwd=2)
legend(-0.8,3.0,legend=c("D1","D2","D3","D4","D5","D6","D7","D8","D9"),
col=c(rgb(1,0.4,0.1),"dodgerblue",rgb(0.8,0,0.9),
                            rgb(0,0,0),rgb(0.5,1,0.1),
                            rgb(1,0,0),rgb(0.7,0.3,0.1),
                            rgb(0.5,0.5,0.6),rgb(1,0.8,0)), lwd = 3, cex=0.8)

#DepRate vs Kerfeed,TimeFrame
DummyVarInt2=read.csv("SaltDepRateData9.csv")
attach(DummyVarInt2)
plot(KerFeed[D1=="1"], log1p(DepRate)[D1=="1"], col=rgb(1,0.4,0.1,0.5), xlab="KerFeed",
ylab="ln(DepRate)",
     main="DepRate vs KerFeed,TimeFrame", pch=16, ylim=c(1.5,3.1), xlim=c(-1,1.7))
points(KerFeed[D2=="1"], log1p(DepRate)[D2=="1"],col="dodgerblue", pch=16)
points(KerFeed[D3=="1"], log1p(DepRate)[D3=="1"],col=rgb(0.8,0,0.9), pch=16)
points(KerFeed[D4=="1"], log1p(DepRate)[D4=="1"],col=rgb(0,0,0,0.75), pch=16)
points(KerFeed[D5=="1"], log1p(DepRate)[D5=="1"],col=rgb(0.5,1,0.1,0.55),pch=16)
points(KerFeed[D6=="1"], log1p(DepRate)[D6=="1"],col=rgb(1,0,0), pch=16)
points(KerFeed[D7=="1"], log1p(DepRate)[D7=="1"],col=rgb(0.7,0.3,0.1,0.65), pch=16)
points(KerFeed[D8=="1"], log1p(DepRate)[D8=="1"],col=rgb(0.5,0.5,0.6), pch=16)
points(KerFeed[D8=="2"], log1p(DepRate)[D8=="2"],col=rgb(1,0.8,0,0.65), pch=16)
legend(-1,3.0,legend=c("D1","D2","D3","D4","D5","D6","D7","D8","D9"),
col=c(rgb(1,0.4,0.1,0.5),"dodgerblue",rgb(0.8,0,0.9),
```

```
rgb(0,0,0,0.75),rgb(0.5,1,0.1,0.55),
rgb(1,0,0),rgb(0.7,0.3,0.1,0.65),
rgb(0.5,0.5,0.6),rgb(1,0.8,0,0.65)),pch=16, cex=1.12)

#DepRate vs ReactordP,TimeFrame
plot(ReactordP[D1=="1"], log1p(DepRate)[D1=="1"], col=rgb(1,0.4,0.1,0.5),
xlab="ReactordP", ylab="ln(DepRate)",
     main="DepRate vs ReactordP,TimeFrame", pch=16, xlim=c(-0.7,0.1), ylim=c(1.5,3.1))
points(ReactordP[D2=="1"], log1p(DepRate)[D2=="1"],col="dodgerblue", pch=16)
points(ReactordP[D3=="1"], log1p(DepRate)[D3=="1"],col=rgb(0.8,0,0.9), pch=16)
points(ReactordP[D4=="1"], log1p(DepRate)[D4=="1"],col=rgb(0,0,0,0.75), pch=16)
points(ReactordP[D5=="1"], log1p(DepRate)[D5=="1"],col=rgb(0.5,1,0.1,0.55),pch=16)
points(ReactordP[D6=="1"], log1p(DepRate)[D6=="1"],col=rgb(1,0,0), pch=16)
points(ReactordP[D7=="1"], log1p(DepRate)[D7=="1"],col=rgb(0.7,0.3,0.1,0.65), pch=16)
points(ReactordP[D8=="1"], log1p(DepRate)[D8=="1"],col=rgb(0.5,0.5,0.6), pch=16)
points(ReactordP[D8=="2"], log1p(DepRate)[D8=="2"],col=rgb(1,0.8,0,0.65), pch=16)
legend(-0.7,3.0,legend=c("D1","D2","D3","D4","D5","D6","D7","D8","D9"),
col=c(rgb(1,0.4,0.1,0.5),"dodgerblue",rgb(0.8,0,0.9),
rgb(0,0,0,0.75),rgb(0.5,1,0.1,0.55),
rgb(1,0,0),rgb(0.7,0.3,0.1,0.65),
rgb(0.5,0.5,0.6),rgb(1,0.8,0,0.65)),pch=16, cex=1.12)

#DepRate vs ClayOP,TimeFrame
plot(ClayOP[D1=="1"], log1p(DepRate)[D1=="1"], col=rgb(1,0.4,0.1,0.5), xlab="ClayOP",
ylab="ln(DepRate)",
     main="DepRate vs ClayOP,TimeFrame", pch=16, ylim=c(1.5,3.2), xlim=c(-0.2,1))
points(ClayOP[D2=="1"], log1p(DepRate)[D2=="1"],col="dodgerblue", pch=16)
points(ClayOP[D3=="1"], log1p(DepRate)[D3=="1"],col=rgb(0.8,0,0.9), pch=16)
points(ClayOP[D4=="1"], log1p(DepRate)[D4=="1"],col=rgb(0,0,0,0.75), pch=16)
points(ClayOP[D5=="1"], log1p(DepRate)[D5=="1"],col=rgb(0.5,1,0.1,0.55),pch=16)
points(ClayOP[D6=="1"], log1p(DepRate)[D6=="1"],col=rgb(1,0,0), pch=16)
points(ClayOP[D7=="1"], log1p(DepRate)[D7=="1"],col=rgb(0.7,0.3,0.1,0.65), pch=16)
points(ClayOP[D8=="1"], log1p(DepRate)[D8=="1"],col=rgb(0.5,0.5,0.6), pch=16)
points(ClayOP[D8=="2"], log1p(DepRate)[D8=="2"],col=rgb(1,0.8,0,0.65), pch=16)
legend(-0.2,3.2,legend=c("D1","D2","D3","D4","D5","D6","D7","D8","D9"),
col=c(rgb(1,0.4,0.1,0.5),"dodgerblue",rgb(0.8,0,0.9),
rgb(0,0,0,0.75),rgb(0.5,1,0.1,0.55),
rgb(1,0,0),rgb(0.7,0.3,0.1,0.65),
rgb(0.5,0.5,0.6),rgb(1,0.8,0,0.65)),pch=16, cex=1.12)

attach(TrainSet2NoInfluential)
plot(KerFeed, ReactordP, col=rgb(1,0.4,0.1,0.5), xlab="KerFeed", ylab="ReactordP",
     main="KerFeed vs ReactordP", pch=16, xlim=c(-0.7,1.7), ylim=c(-0.7,0.2))


plot(KerFeed, ClayOP, col="dodgerblue", xlab="KerFeed", ylab="ClayOP",
     main="KerFeed vs ClayOP", pch=16, xlim=c(-2,5), ylim=c(-5,5))

plot(ClayOP, ReactordP, col=rgb(0.8,0,0.9,0.5), xlab="ClayOP", ylab="ReactordP",
     main="ClayOP vs ReactordP", pch=16, xlim=c(-0.1,0.9), ylim=c(-0.6,0.2))


plot((0.407*KerFeed+0.073*ClayOP)[D3=="1"], log1p(DepRate)[D3=="1"],
col=rgb(1,0.4,0.1,0.5), xlab="Z(KerFeed,ClayOP)", ylab="ln(DepRate)",
     main="Z vs ln(DepRate)", pch=16, xlim=c(-0.2,0.7))
points((0.407*KerFeed+0.073*ClayOP-(0.097*KerFeed*ClayOP))[D3=="1"],
log1p(DepRate)[D3=="1"],col="dodgerblue", pch=16)
legend(-0.2,3,legend=c("w/o Interactions","w/ Interactions"),
       col=c(rgb(1,0.4,0.1,0.5),"dodgerblue"),pch=16, cex=1.12)

plot((-0.021*ReactordP+0.073*ClayOP)[D3=="1"], log1p(DepRate)[D3=="1"],
col=rgb(1,0.4,0.1,0.5), xlab="Z(ReactordP,ClayOP)", ylab="ln(DepRate)",
```

```
     main="Z vs ln(DepRate)", pch=16, xlim=c(0.039,0.048))
points((-0.021*ReactordP+0.073*ClayOP+(0.035*ReactordP*ClayOP))[D3=="1"],
log1p(DepRate)[D3=="1"],col="dodgerblue", pch=16)
legend(0.0447,3,legend=c("w/o Interactions","w/ Interactions"),
       col=c(rgb(1,0.4,0.1,0.5),"dodgerblue"),pch=16, cex=1.12)

plot((0.407*KerFeed-0.021*ReactordP)[D3=="1"], log1p(DepRate)[D3=="1"],
col=rgb(1,0.4,0.1,1), xlab="Z(KerFeed,ReactordP)", ylab="ln(DepRate)",
     main="Z vs ln(DepRate)", pch=16, xlim=c(-0.2,0.7))
points((0.407*KerFeed-0.021*ReactordP-(0.03*KerFeed*ReactordP))[D3=="1"],
log1p(DepRate)[D3=="1"],col=rgb(0.12,0.565,0.9,0.65), pch=16)
legend(-0.2,3,legend=c("w/o Interactions","w/ Interactions"),
       col=c(rgb(1,0.4,0.1,1),rgb(0.12,0.565,0.9,1)),pch=16, cex=1.12)

plot((0.407*KerFeed-0.021*ReactordP+0.073*ClayOP), log1p(DepRate),
     col=rgb(1,0.4,0.1), xlab="Total Z", ylab="ln(DepRate)", ylim=c(1.25,3.1),
     main="Total Z vs ln(DepRate)", pch=16, xlim=c(-0.2,0.7))
points((0.407*KerFeed-0.021*ReactordP+0.073*ClayOP+(-0.03*KerFeed*ReactordP)+
     (0.035*ReactordP*ClayOP)-(0.097*KerFeed*ClayOP)),
      log1p(DepRate),col=rgb(0.12,0.565,0.9,0.4), pch=16)
legend(0.35,1.85,legend=c("w/o Interactions","w/ Interactions"),
       col=c(rgb(1,0.4,0.1,0.5),rgb(0.12,0.565,0.9,0.4)),pch=16, cex=1.12)


#ADDITIONAL BENCHMARK
#Model 9
NoDummy = read.csv("SaltDepRateData11.csv")
SLRMTransNoDummy = lm(log1p(DepRate)~., data = NoDummy)
summary(SLRMTransNoDummy)
step(SLRMTransNoDummy)
SLRMTransNoDummyReduced = lm(log1p(DepRate) ~ KerFeed+KerTemp+WaterLev+BrineLev+
                          WaterpH+ECPLev+ReactorIP+ ReactordP+ClaydP+
                          ClayOP+KerFeed:ReactorIP+KerFeed:KerTemp+
                          KerFeed:ReactordP+KerFeed:ClaydP+KerFeed:ClayOP+
                          KerFeed:BrineLev+WaterpH:WaterLev+ReactorIP:ReactordP+
                          ReactorIP:ClaydP+ReactorIP:ClayOP+ReactordP:ClaydP+
                          ReactordP:ClayOP+ClaydP:ClayOP, data = NoDummy)
summary(SLRMTransNoDummyReduced)
step(SLRMTransNoDummyReduced)

SLRMTransNoDummyReduced2 = lm(log1p(DepRate) ~ KerFeed+KerTemp+WaterLev+BrineLev+
                           WaterpH+ReactorIP+ReactordP+ClaydP+ClayOP+
                           KerFeed:ReactorIP+KerFeed:ClaydP+KerFeed:ClayOP+
                           KerFeed:BrineLev+ReactorIP:ClaydP+ReactordP:ClaydP+
                           ReactordP:ClayOP+ClaydP:ClayOP, data = NoDummy)
summary(SLRMTransNoDummyReduced2)
vif(SLRMTransNoDummyReduced2)

#Remove ReactorIP
SLRMTransNoDummyReduced3 = lm(log1p(DepRate) ~ KerFeed+KerTemp+WaterLev+BrineLev+
                           WaterpH+ReactordP+ClaydP+ClayOP+KerFeed:ClaydP+
                           KerFeed:ClayOP+KerFeed:BrineLev+ReactordP:ClaydP+
                           ReactordP:ClayOP + ClaydP:ClayOP, data = NoDummy)
summary(SLRMTransNoDummyReduced3)
vif(SLRMTransNoDummyReduced3)
step(SLRMTransNoDummyReduced3)

SLRMTransNoDummyReduced4 = lm(log1p(DepRate) ~ KerFeed+KerTemp+BrineLev+WaterpH+
                           ReactordP+ClaydP+ClayOP+KerFeed:ClaydP+KerFeed:ClayOP+
                           KerFeed:BrineLev+ReactordP:ClayOP+ClaydP:ClayOP,
                           data = NoDummy)
summary(SLRMTransNoDummyReduced4)
```

```
PredictModel9 = exp(predict(SLRMTransNoDummyReduced4,TestSet2))
MAE(PredictModel9, TestSet2$DepRate)
MSE(PredictModel9, TestSet2$DepRate)
RMSE(PredictModel9, TestSet2$DepRate)

#RELATIVE IMPORTANCE
calc.relimp(SLRMTransFinalModelSimplified, rela=TRUE)
calc.relimp(SLRMTransFinalModelSimplified, rela=FALSE)

#ACCURACY
#Model0
Base = read.csv("SaltDepRateData13.csv")
SLRMNoStandardizationNoDummy = lm(DepRate~., data = Base)
summary(SLRMNoStandardizationNoDummy)
Pred0 = predict(SLRMNoStandardizationNoDummy,Base)
write.table(Pred0,"C:/Users/rpdlpascual/Desktop/Pred0.txt",sep="\t")

#Model3
Pred3 = exp(predict(SLRMTransNoSD,Population))
write.table(Pred3,"C:/Users/rpdlpascual/Desktop/Pred3.txt",sep="\t")

#Model4
Pred4 = exp(predict(SLRMTransNoSDNoInfluential,Population))
write.table(Pred4,"C:/Users/rpdlpascual/Desktop/Pred4.txt",sep="\t")

#Model5
Pred5 = exp(predict(SLRMTransNoSDWithInt,Population))
write.table(Pred5,"C:/Users/rpdlpascual/Desktop/Pred5.txt",sep="\t")

#Model5a
Pred5a = exp(predict(SLRMTransNoSDNoInfWithInt,Population))
write.table(Pred5a,"C:/Users/rpdlpascual/Desktop/Pred5a.txt",sep="\t")

#Model6
Pred6 = exp(predict(SLRMTransNoSDNoInfWithIntStepReduced,Population))
write.table(Pred6,"C:/Users/rpdlpascual/Desktop/Pred6.txt",sep="\t")

#Model7
Pred7=exp(predict(SLRMTransFinalModel,Population))
write.table(Pred7,"C:/Users/rpdlpascual/Desktop/Pred7.txt",sep="\t")

#Model8
Pred8=exp(predict(SLRMTransFinalModelSimplified,Population))
write.table(Pred8,"C:/Users/rpdlpascual/Desktop/Pred8.txt",sep="\t")

#Model9
Pred9=exp(predict(SLRMTransNoDummyReduced4,Population))
write.table(Pred9,"C:/Users/rpdlpascual/Desktop/Pred9.txt",sep="\t")
```