11/19/2023

FINAL PROJECT REPORT

STUDY OF RELATIVE PERFORMANCE OF CPU

Shaarif Anas Mohammed and Sofiyan Ali Syed

ABSTRACT

The dataset was taken from the UCI machine learning repository from University of California, Irvine. The dataset concerns relative performance data of CPU. It has 209 cases and 6 continuous variables.

The aim of the project is to establish a best possible regression equation to predict the CPU performance using the explanatory variables. It aims to highlight the model selection and model building procedure with and satisfy the model assumptions via model selection and elimination of outliers.

First the first order model was fit to check for significant variables. The variable specifying minimum channels came out to be insignificant. Later stepwise model selection methods were used to select best model using the significant variables and including the interaction and second order terms.

After that Diagnostic plots were drawn to check for outliers. There were few noticeable outliers. Potential outliers were checked whether they are influential outliers or not based on their influence on prediction of CPU performance and then deleted later. The same process of diagnostics was repeated again after deleting the outliers.

Finally best model was chosen and interpreted. Although the diagnostic plots for outliers were not ideal, more observations could not be deleted as it will impact the size of the dataset. After that diagnostic plots for model assumptions were checked.

The model was not perfectly ideal. Possible explanations could be that the dataset was not suitable for linear regression analysis. Also, it could nor be suitable for regression at all. Other analysis methods might be suitable for analysis of such data.

INTRODUCTION

Central Processing Unit performance, is a critical aspect of a computer system's overall speed and efficiency. The CPU, often referred to as the brain of the computer, is responsible for executing instructions of a computer program, performing calculations, and managing data movement within the system.

The term "CPU performance" refers to the ability of a Central Processing Unit to execute instructions and tasks within a given period. It is often measured in terms of speed, throughput, and efficiency.

CPU performance is a multifaceted concept influenced by factors such as clock speed, architectural design, the number of cores, cache efficiency, and technological advancements. As technology evolves, CPUs continue to be a focal point for improvements, enabling faster and more efficient computing.

USE CASE:

Researchers and data scientists can use this dataset to build predictive models that estimate the relative CPU performance of a computer system based on its specifications. The dataset provides a valuable resource for studying the impact of different hardware attributes on overall system performance.

Potential use cases for studying the Relative CPU Performance are;

- Predictive Modelling
- System Optimization
- Benchmarking and Comparison
- System Specification Recommendations
- Research in Computer Architecture
- Educational Purposes

These use cases highlight the versatility of the Relative CPU Performance dataset, making it valuable for a wide range of applications in both industry and academia. The dataset's attributes offer a rich source of information for exploring and understanding the factors influencing CPU performance in computer systems.

DATASET

The Relative CPU Performance dataset is a collection of attributes related to the performance of computer systems, with a specific emphasis on CPU (Central Processing Unit) performance.

Data Source: The dataset is sourced from the UCI Machine Learning repository, which ensures the reliability and standardization of machine learning datasets. It is designed for researchers and data scientists interested in exploring the relationships between different attributes and the overall CPU performance of computer systems.

Attributes:

1. **MYCT** (Machine Cycle Time): The time taken for one machine cycle, measured in nanoseconds (integer).
2. **MMIN** (Minimum Main Memory): The minimum main memory required by the computer system, measured in kilobytes (integer).
3. **MMAX** (Maximum Main Memory): The maximum main memory capacity of the computer system, measured in kilobytes (integer).
4. **CACH** (Cache Memory): The size of the cache memory in kilobytes (integer).
5. **CHMIN** (Minimum Channels): The minimum number of channels in units (integer).
6. **CHMAX** (Maximum Channels): The maximum number of channels in units (integer).
7. **PRP** (Published Relative Performance): The published relative performance, which serves as the target variable for the dataset, measured in integer units.

The Relative CPU Performance dataset offers a comprehensive set of attributes for exploring and understanding the factors influencing CPU performance in computer systems. Its inclusion of the PRP variable as the target makes it suitable for regression analysis and predictive modelling in the field of machine learning.

METHODOLOGY

The methodology of the analysis is a simple model selection and model building methodology keeping in view the model assumptions and outliers.

First a full model with all the explanatory variables was fit to check significance of each variable. The model summary will give us results of individual t-test on each variable. Also, overall F-test statistic and p-value will also be given to check whether any of the explanatory variables is important in predicting the response variable.

After significant variables were selected, model selection was started. Here, we have used forward stepwise selection and backward stepwise selection process in R for model selection. Each model was shortlisted from forward and backward stepwise selection from 3 sets. First set contained first order variable terms and square terms, second set contained first order variable terms and interaction terms only and third set first order variable terms, square terms and interaction terms altogether. Full model and reduced model excluding the insignificant variable was also added to this pool of models for selection of best model. The best model was selected base on AIC and BIC values.

We checked for outliers using studentised deleted residuals, leverage, DFFITS, Cook's D and DFBETAS. Their respective plots were also examined. The outlying observations were divided into two sets based on their occurrence in these measures. The first set was more severe outliers while second set was less severe outliers.

The first set was of potential outliers were checked for their influence on predicted values and were deleted later. After deletion of first set of outliers the plots and measures were checked again. We still found some more outliers and then proceeded to check for second set of outliers. Their influence was checked on predicted values and diagnostics plots without them were checked to see if they actually improve out dataset or not.

Finally, we came to conclusion that the second set of outliers did not greatly affect our dataset and we decided to retain them. Thus, the model building procedure was concluded and the final model was built using updated dataset.

After choosing the final model, Diagnostics plots were examined to study the satisfaction of model assumptions. Residual vs explanatory variables for correct specification, residuals vs predicted value for checking constant variance, time series plot for Independence and histogram and QQ plot for normality was checked for model assumptions.

RESULTS AND OUTCOMES

FULL MODEL:

Firstly, a full model was built to check for significant variables. The summary of model provided us with individual t-test and an overall F-test to check if any of the variables are significant in predicting the relative performance of CPU.

Only minimum number of channels was not significant. The p-value of the variable was 0.7524 which is way higher than significance level. Also, machine cycle time had a p-value of 0.005 which is borderline significant. Other all variables were highly significant.

The test statistic of Overall F-test was 215.5 on 6 and 202 degrees of freedom. The p-value was <0.01 suggesting at least one of the variables is significant in predicting the relative performance of CPU.

Partial F-test: As minimum channels failed the t-test, a partial F-test was performed with full model and reduced model not containing the minimum channels. The F-value was 0.0998 and p-value was 0.7524 suggesting that the reduced model id better fit than the complete model.

SACTTERPLOT MATRIX AND CORRELATION MATRIX:

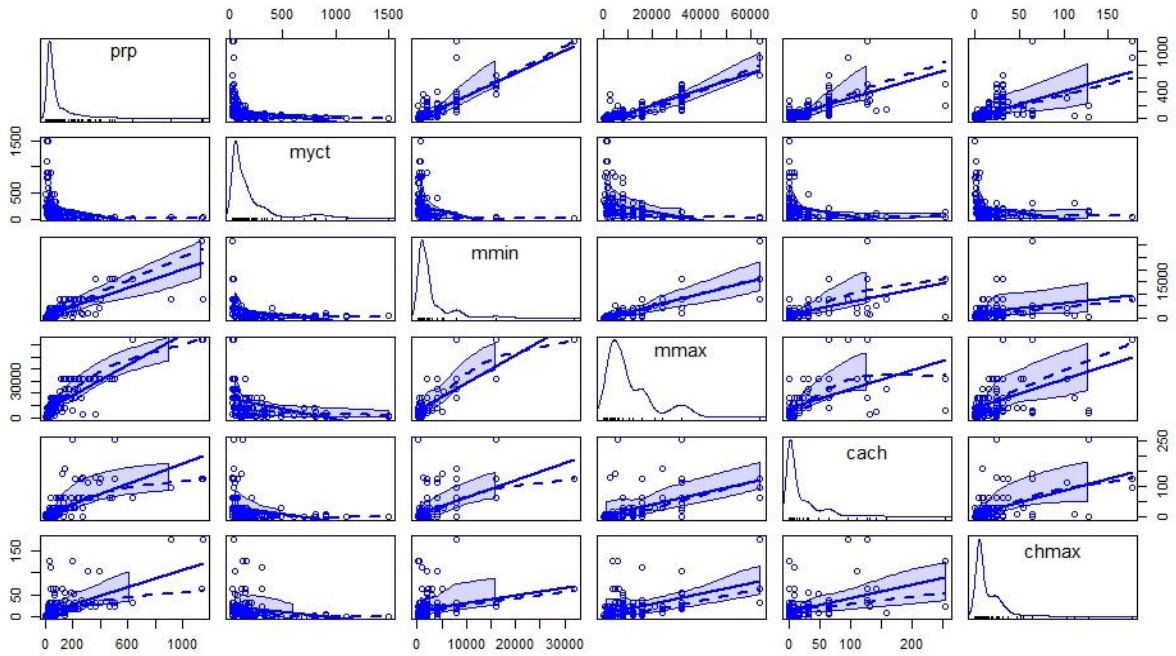


Fig-1; scatterplot matrix

Table-1; Correlation matrix

prp	myct	mmin	mmax	cach	chmax
1	-0.3071	0.7949	0.863	0.6626	0.6052
-0.3071	1	-0.3356	-0.3786	-0.321	-0.2505
0.7949	-0.3356	1	0.7582	0.5347	0.2669
0.863	-0.3786	0.7582	1	0.538	0.5272
0.6626	-0.321	0.5347	0.538	1	0.4878
0.6052	-0.2505	0.2669	0.5272	0.4878	1

We can see that there is fairly good correlation between the response variable and explanatory variables. Maximum main memory and minimum main memory has strong positive correlation of 0.86 and 0.79 respectively while machine cycle time has negative correlation of 0.3. Cache memory and maximum channels have moderate positive correlation of 0.6 each. Also, the question of multicollinearity does not appear to be in picture.

The same can be inferred from the scatterplot matrix.

ADDED VARIABLE PLOTS:

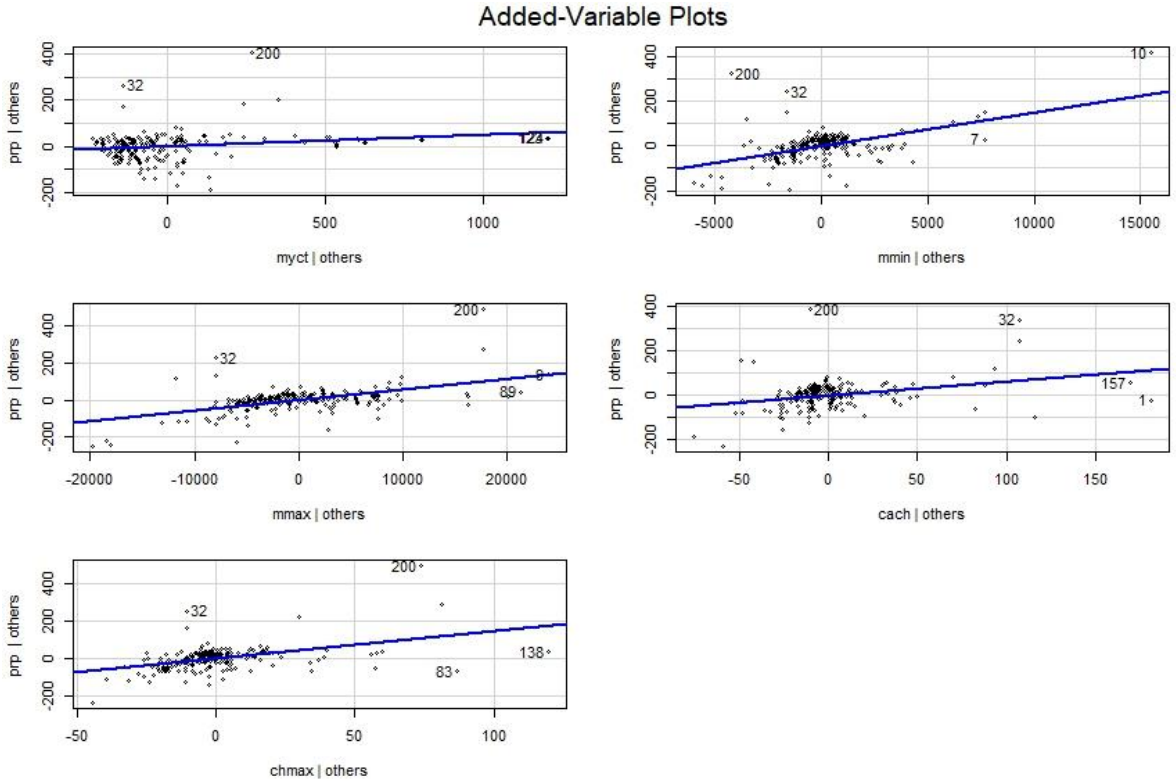


Fig-2; Added Variable plots

We can see from the added Variable plots that all five explanatory variables have a near linear relationship. The plots do not show any concrete evidence for any kind of quadratic or curvilinear relationship. Mostly we can infer a linear relationship from all explanatory variables.

STEPWISE MODEL SELECTION:

Two models from each set of variable terms were shortlisted. One from forward stepwise selection and one from backward stepwise selection. First set contained first order variable terms and square terms, second set contained first order variable terms and interaction terms only and third set first order variable terms, square terms and interaction terms altogether. Full model and reduced model excluding the insignificant variable was also added to this pool of models for selection of best model. The best model was selected base on AIC and BIC values. The AIC and BIC values for the eight shortlisted models are;

Table -2; Models AIC and BIC values

mod.name	AIC	BIC
modelsq.for	2162.448284	2195.871626
modelsq.back	2162.100356	2192.181364
modelint.for	2113.595448	2153.703459
modelint.back	2099.906804	2140.014815
model1.for	2149.089521	2189.197532
model1.back	2149.089521	2189.197532
model.r	2311.455233	2334.851573
model1	2313.351969	2340.090643

From the table we can see that the model with least AIC and BIC values is the model selected from backward stepwise selection containing the first order terms and interaction terms only. The summary of the selected model is as follows.

Call:

lm(formula = prp ~ myct + mmin + mmax + cach + chmax + myct:cach +

mmin:mmax + mmin:cach + mmin:chmax + mmax:chmax)

Residuals:

Min	*1Q*	*Median*	*3Q*	*Max*
-163.701	*-15.009*	*-4.016*	*11.331*	*156.740*

Coefficients:

	Estimate	*Std. Error*	*t value*	*Pr(>|t|)*
(Intercept)	*1.711e+01*	*6.648e+00*	*2.573*	*0.01080 **
myct	*1.801e-03*	*1.130e-02*	*0.159*	*0.87354*
mmin	*6.547e-03*	*2.117e-03*	*3.092*	*0.00227 ***
mmax	*7.968e-04*	*6.263e-04*	*1.272*	*0.20477*
cach	*1.662e+00*	*1.645e-01*	*10.104*	*< 2e-16 ****
chmax	*-4.367e-01*	*1.796e-01*	*-2.432*	*0.01592 **
myct:cach	*-6.296e-03*	*1.416e-03*	*-4.445*	*1.46e-05 ****
mmin:mmax	*2.842e-07*	*6.181e-08*	*4.598*	*7.60e-06 ****
mmin:cach	*-6.279e-05*	*1.470e-05*	*-4.272*	*3.01e-05 ****
mmin:chmax	*1.070e-04*	*7.090e-05*	*1.509*	*0.13284*
mmax:chmax	*5.083e-05*	*9.080e-06*	*5.598*	*7.16e-08 ****

---

*Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1*

Residual standard error: 35.67 on 198 degrees of freedom

Multiple R-squared: 0.9532, Adjusted R-squared: 0.9508

F-statistic: 403 on 10 and 198 DF, p-value: < 2.2e-16

In the summary we can see that few terms are insignificant, but the p-value shown is of individual t-test. Their significance is important in conjugation with other variable terms. Also, without the first order terms interaction terms can't be placed in the model. Thus, all first order terms are included in the model.

Thus, after model selection the above model was chosen as the best possible model with the variables. We now proceed to check for outliers in the dataset.

OUTLIERS:

Mainly 5 measures were used for identifying outliers. Studentised deleted residuals, leverage values, DFFITS, Cook's D, DFBETAS. Their respective plots were also drawn to check for outliers.

Assessing rule for determining influential observation is

I. |DFFITS| > 2*Sqrt(p/n)

II. $D_i > F_{(0.50,p,n-p)}$

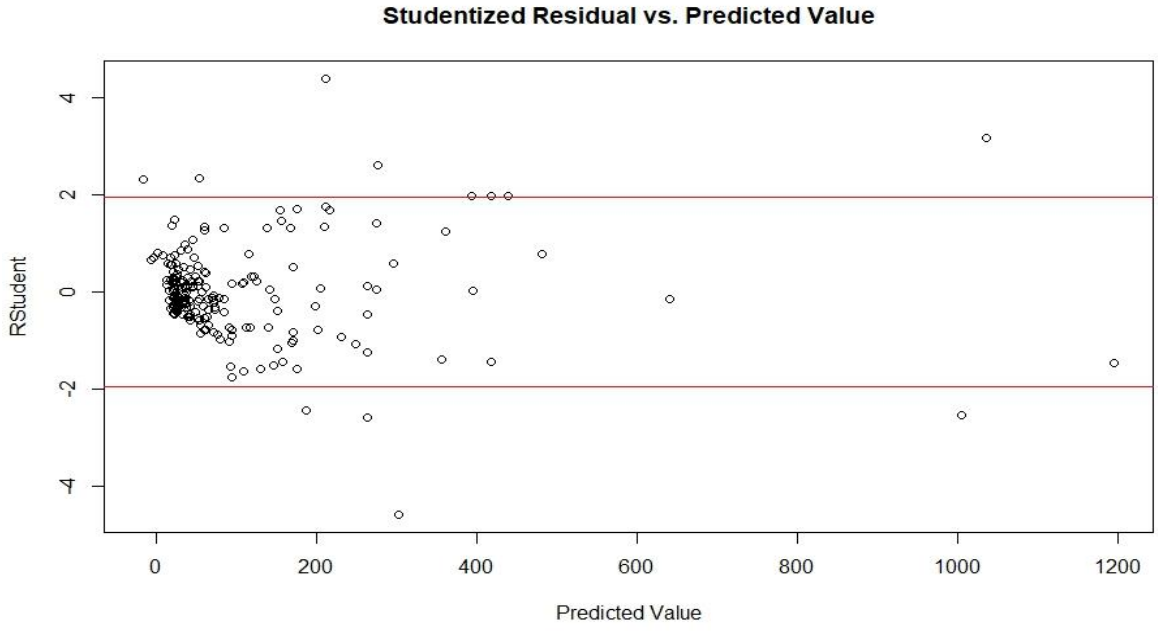
III. |DFBETAS| > 2/Sqrt(n)

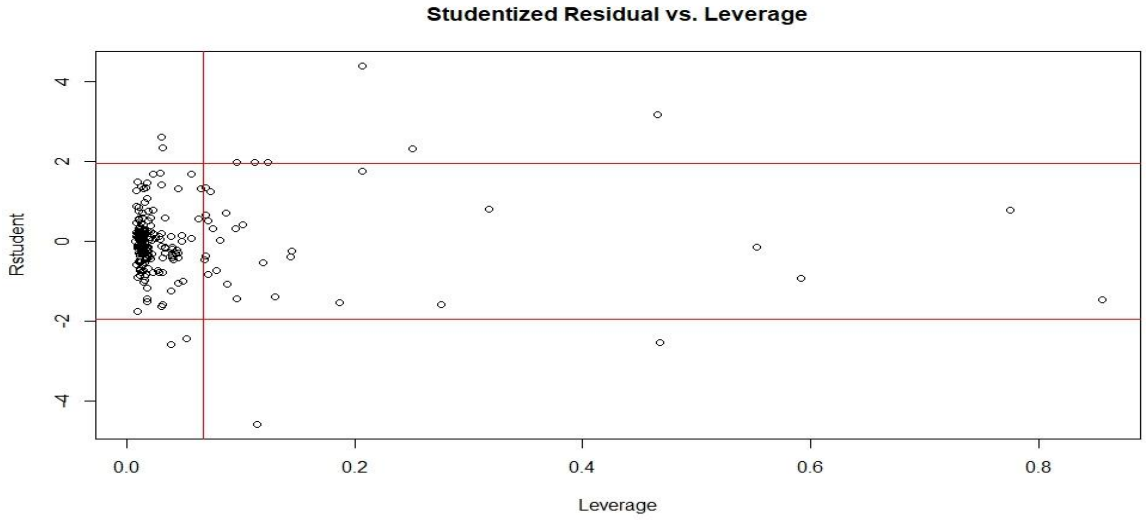
IV. Bonferroni's interval is used for studentised deleted residuals

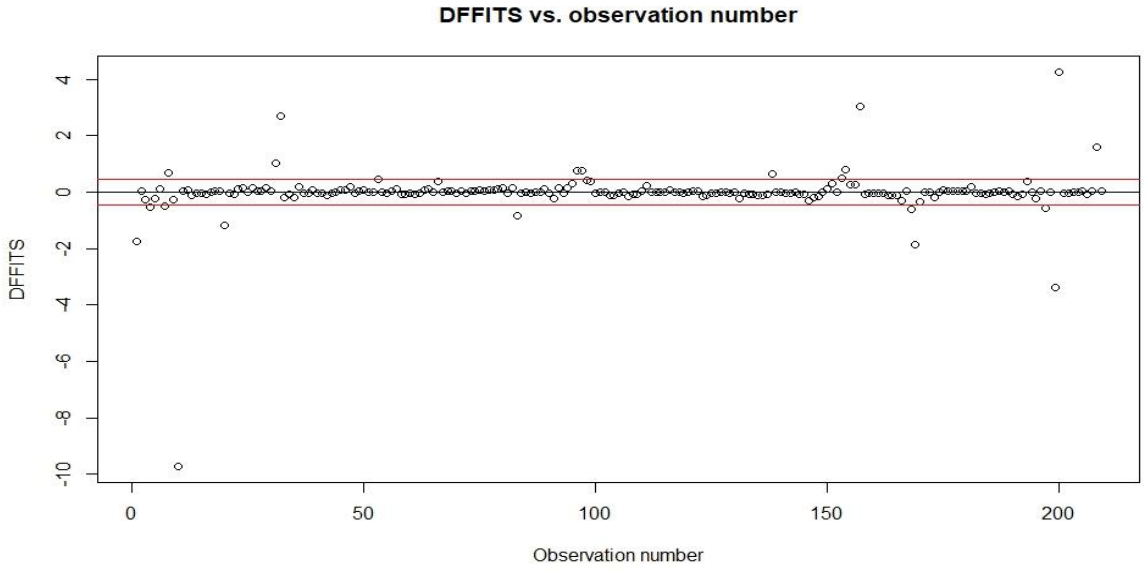
V. h.ii>2*p/n for leverage values.

During the process the DFBETAS turned out to be too sensitive to the dataset. It gave around 70 observations as outliers. Thus, we ignored the DFBETAS as it may not perform well with the given dataset. Therefore, we worked with the remaining 4 measures to identify potential outliers.

The plots of each measure were as follows;




Studentized Residual vs. Leverage
Rstudent
Leverage
4
2
0
-2
-4
0.0
0.2
0.4
0.6
0.8


DFFITS vs. observation number
DFFITS
Observation number
4
2
0
-2
-4
-6
-8
-10
0
50
100
150
200

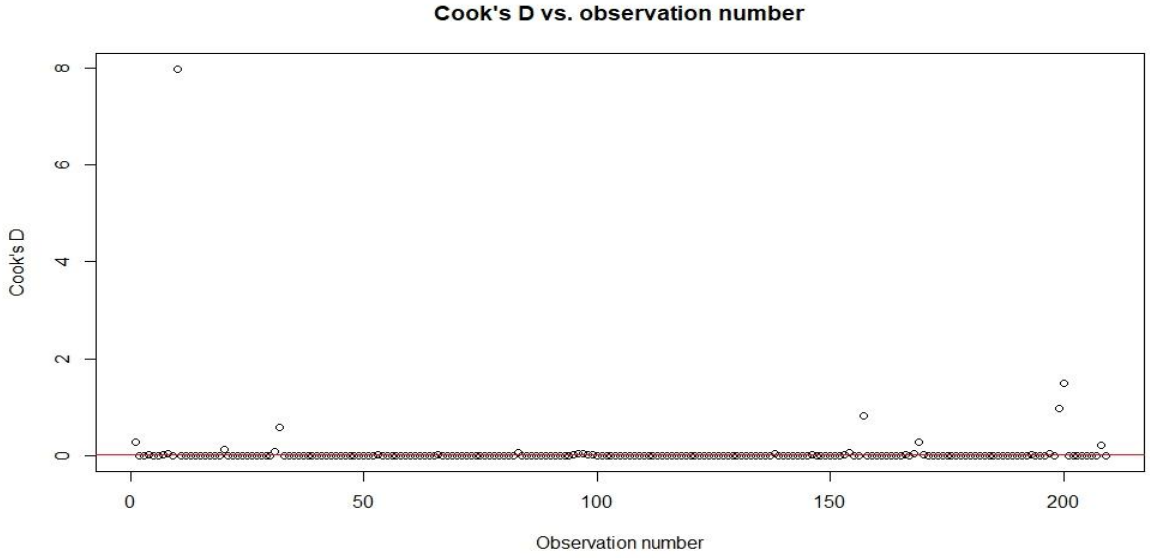

Cook's D vs. observation number
Cook's D
Observation number
8
6
4
2
0
0
50
100
150
200

Fig 3,4,5,6;

There are considerable outliers in the plots. We can see few extreme outliers as well. Based on the Assessing criteria for the four measures a table was made as which observations are outliers. The table is as follows.

Table-3; Outlying Observations for each measure

Deleted residuals	Leverage	DFFITS	cooks D
	1		
		8	
	9		
10	10		10
	20		
	31	31	
32	32	32	
		53	
		66	
	83		
	96	96	
	97	97	
		98	
	123		
	124		
	138	138	
		153	
	154	154	
	157	157	
169	169		
	173		
	197		
	199		199
200	200	200	200
	208	208	

We can see that the table has 25 values. But we can't blindly just delete all of them as important information from the dataset will be lost. Therefore, we make two sets of outlying observations. The first set contains severe outliers. The severe outliers are those that were classified as outliers by 3 or more measures. They are highlighted in red colour in the table.

The second set of outliers are the ones that are less severe. They are the ones that were classifies as outliers by two measures. They are highlighted in yellow colour in the table.
This kind of practice is done to conserve important information of the dataset and delete as less observations as possible.
First the set of severe outliers was examined for their influence on predicted value. The set 1 of outliers (observation numbers 10, 32 and 200) were removed from the dataset and chosen

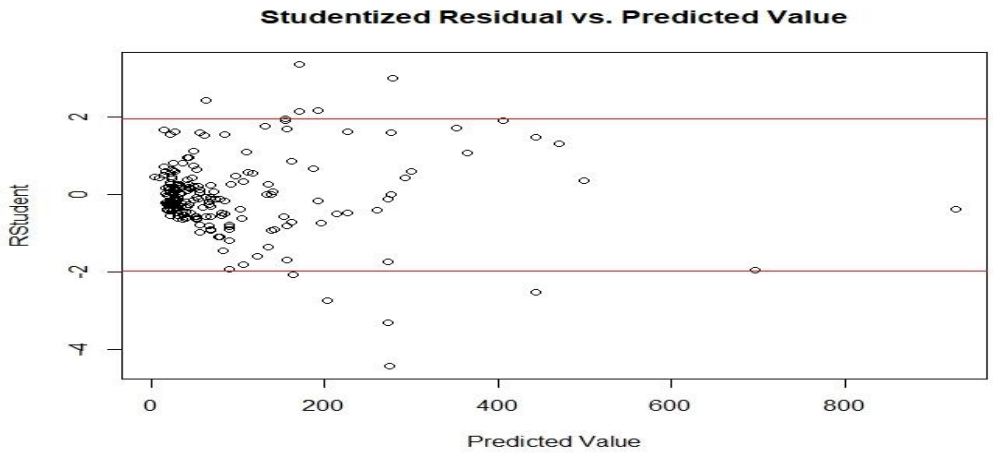
regression model was fit without these observations. Now the reduced dataset is predicted for y value using the both the regression equations (regression equation with and without the above said observations). And percentage change in y value was calculated. Here are some of the results of percentage change. The mean absolute percentage change was about 16.44%.

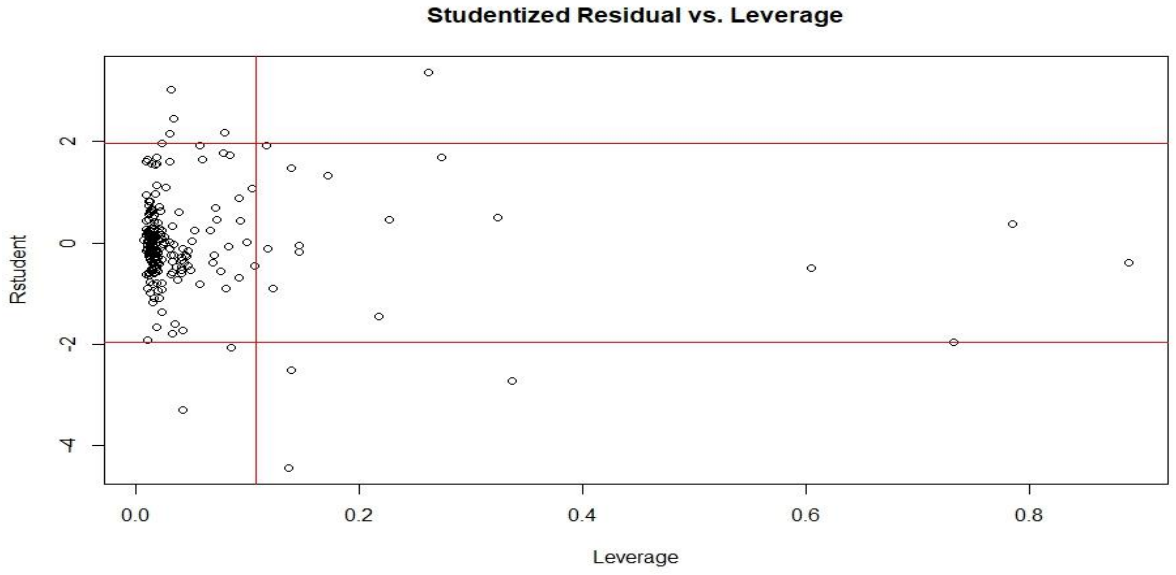
		1.341951	12.08945			
		15.00824	24.90829			
		10.48107	10.51149			
		-3.34059	23.74617			
	-11.3443	27.89166	15.87758			
	15.02267	29.24846	10.21331	-9.72684		
	-12.5751	18.1236	19.05026	21.4651		
	-9.37264	3.944895	56.14636	17.13255		
-0.89483	-1.3014	15.21664	8.651959	12.13082	-18.7468	0.18892
796.9094	5.758516	14.87047	-9.02254	-6.13077	29.80671	-200.236
-5.93408	7.667852	14.75815	67.10959	16.87826	-372.507	-9.81497
		21.86008	-11.9674			
		-18.9256				
		-4.66449				

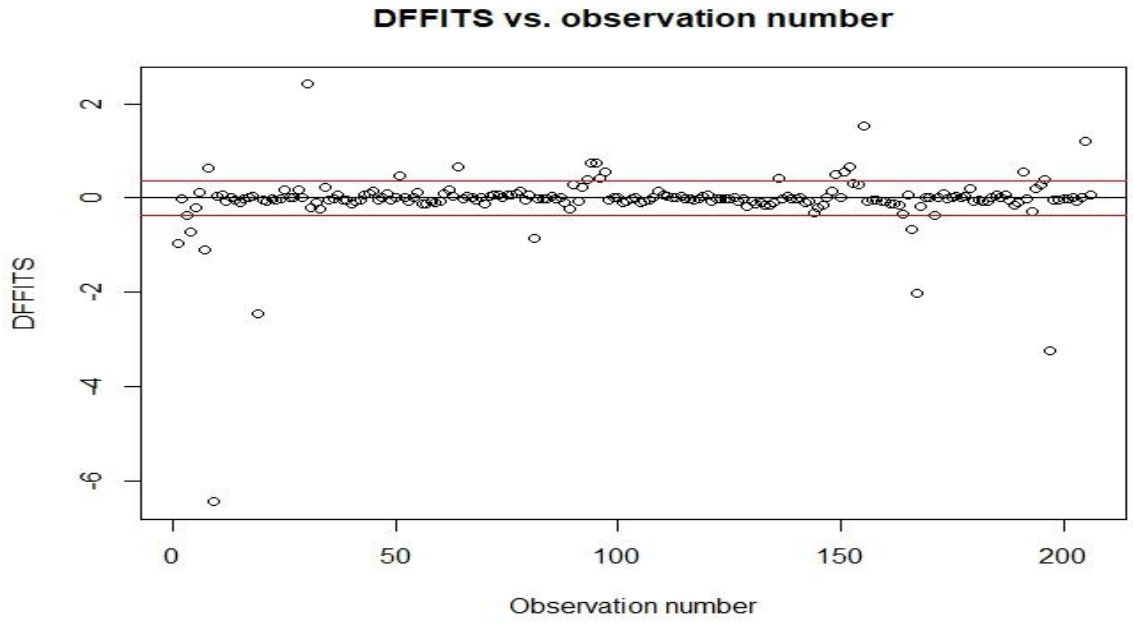
We can see that there are so many extreme values and many values higher than 15%. 15% change in prediction of relative performance of a CPU is a big deal for any researcher. Therefore, based on the above reasons we decide that the three observations and quite influential outliers and delete them .

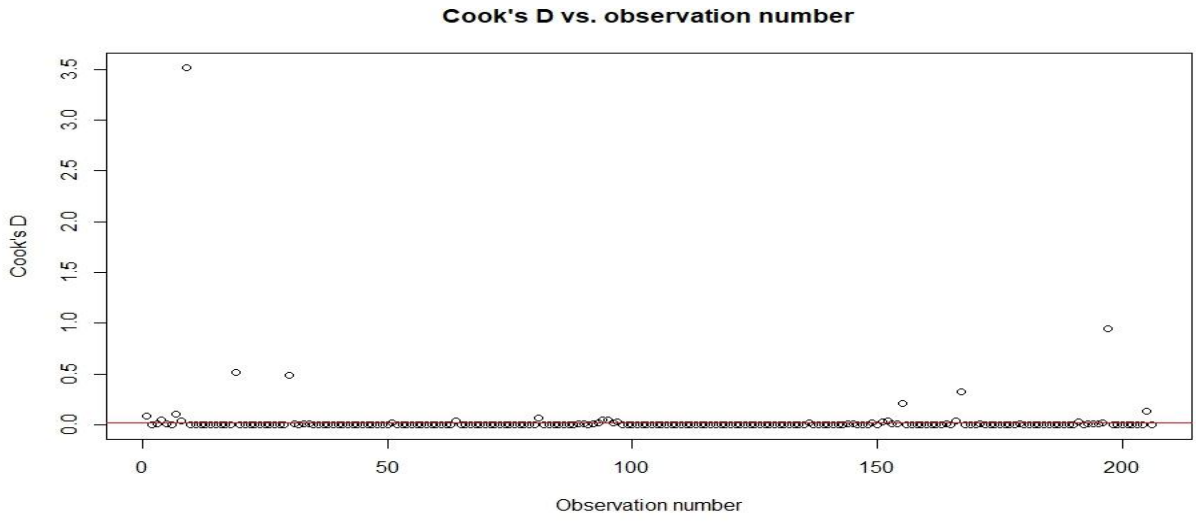
After their deletion, the diagnostic plots were checked again to see if the dataset is free of outliers. The plots were as follows;

DIAGNOSTIC PLOTS AFER REMOVING FIRST SET OF OUTLIERS:




Studentized Residual vs. Leverage
Rstudent
2
0
-2
-4
0.0
0.2
0.4
0.6
0.8
Leverage


DFFITS vs. observation number
DFFITS
2
0
-2
-4
-6
0
50
100
150
200
Observation number


Cook's D vs. observation number
Cook's D
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
0
50
100
150
200
Observation number

We can see that the diagnostics plots look little better. Although there are still outliers it is better than the previous plots. Few of the extreme outliers are eliminated although there might still be outliers present in the dataset. If we closely look at the scale of the plots, we can see that the scale is significantly reduced meaning the extreme outliers are not present.

As there are still outliers present let us see if they are Influential outliers and try to eliminate them for further analysis.

The second set of outliers which were present in the at least two measures will be taken and eliminated this time. This further reduces our size of the dataset from 206 to 197. The observation numbers will also change as we have already eliminated 3 observations. That has to be taken care of while eliminating the second set.

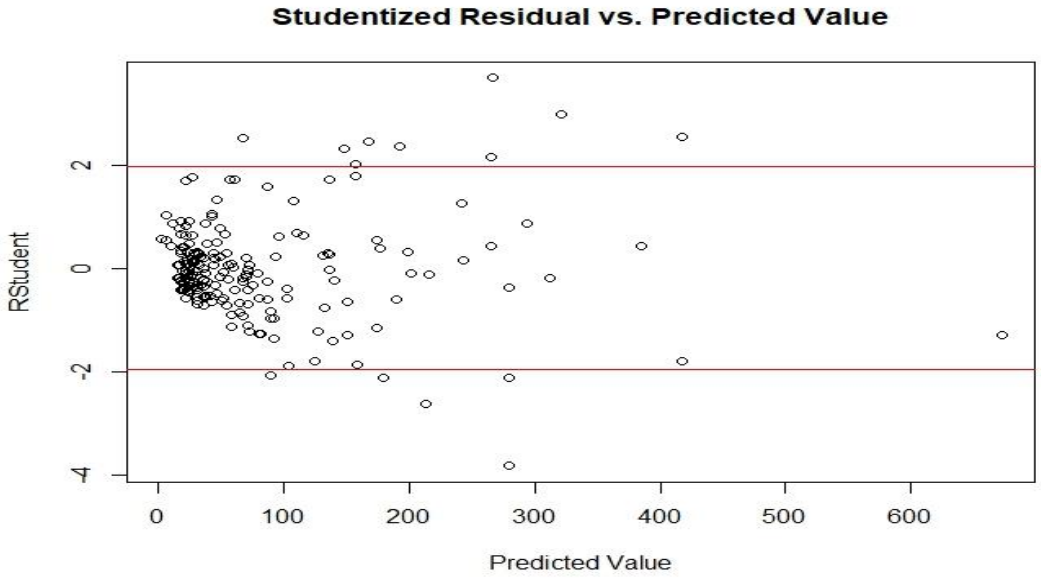
The set 2 of outliers (31,96,97,138,154,157,167,199,208) were removed from the dataset and chosen regression model was fit without these observations. Now the reduced dataset is predicted for y value using the both the regression equations (regression equation with and without the above said observations). And percentage change in y value was calculated. Here are some of the results of percentage change. The mean percentage change was about 4.91%.

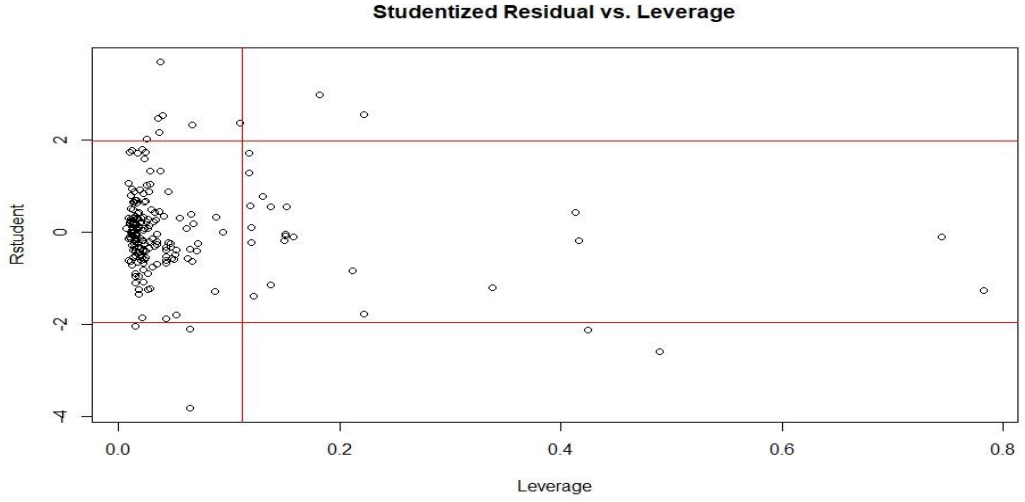
-5.73430556	0.433995803	-2.373601576	
2.565608256	-0.463823819	-0.424747518	6.965217962
2.565608256	-3.383293572	-40.46478776	6.965217962
2.565608256	-2.875195962	-23.94818102	-6.466703388
-4.266656885	-5.875484093	-23.5800446	3.19136943
-1.90748025	-23.18951815	-3.629988038	-1.952265186
-5.974903916	-2.931502414	-18.2209339	6.266676674
-5.974903916	-3.019330652	-5.030892003	4.662949629
-3.387101031	-3.019330652	-52.23179173	0.338966883
0.402647075	0.175166504	-5.133741492	-3.348638566
		-1.396571122	2.958155977
		-46.65133793	-1.194258244
		-11.88285497	

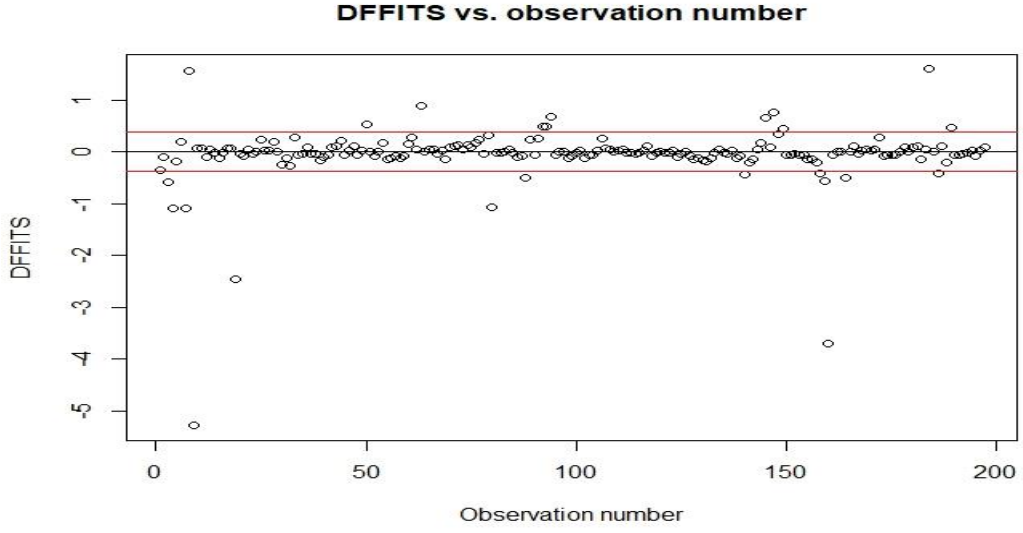
Compared to the previous percentage in predicted value when we removed the first set of outliers the percentage change for second set is very low. We can see that only a few observations have high percentage change. Also, the mean absolute percentage change is 4.9%, which is not that significant in terms of relative performance of CPU.

Let us still see the diagnostic plots if removing these observations improved the plots significantly.

DIAGNOSTIC PLOTS AFER REMOVING FIRST SET OF OUTLIERS:







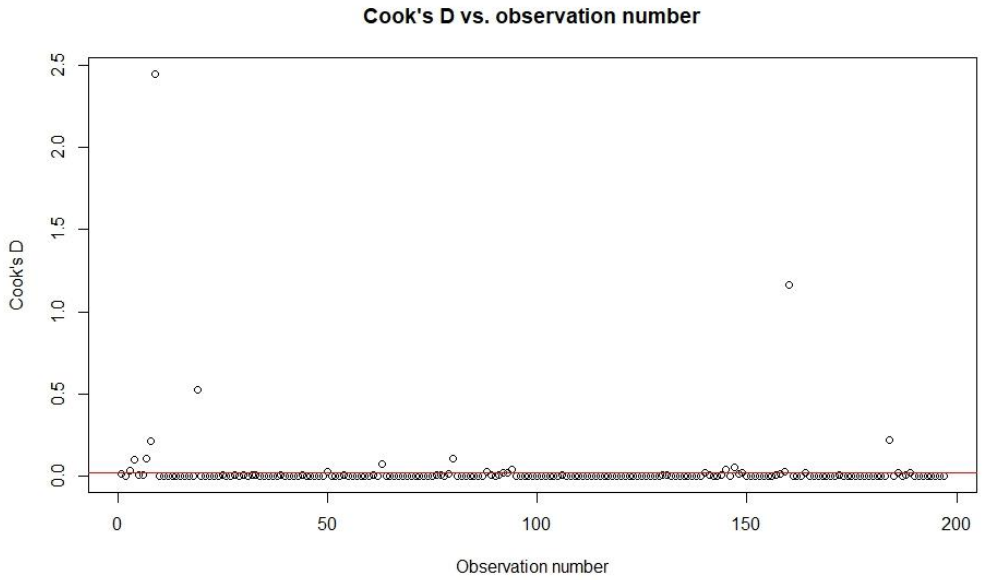


Fig 11,12,13,14;

We can see that after removal of second set of outliers there is no significant change in the diagnostics plot. Also, the influence of response variables is not that great. It is just around 5%, which in case of our dataset is not significant. Increase or decrease of CPU speed by 5% is a common variation is Computer Architecture.

Also, the other diagnostics plots didn't give any improvement either. The outliers were eliminated but other outliers were introduced due smaller sample size. We can't keep further deleting the observations as this will reduce the sample size and will significantly impact the prediction power of the regression equation. Thus, we conclude that the second set of outliers should be retained in the dataset as their elimination is not providing any valuable improvement.

Thus, the final model will the best selected model regressed upon the dataset with eliminated first set of outliers.

FINAL MODEL:

The summary of the final model is as follows;

Call:
lm(formula = prp ~ myct + mmin + mmax + cach + chmax + myct:cach +
mmin:mmax + mmin:cach + mmin:chmax + mmax:chmax, data = data.206)

Residuals:
Min 1Q Median 3Q Max

-135.547 -13.960 -1.998 11.026 102.722

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	*1.793e+01*	*6.008e+00*	*2.985*	*0.003201 **
myct	*-1.980e-03*	*9.756e-03*	*-0.203*	*0.839420*
mmin	*8.004e-04*	*2.448e-03*	*0.327*	*0.744057*
mmax	*1.710e-03*	*5.474e-04*	*3.123*	*0.002063 **
cach	*1.245e+00*	*1.569e-01*	*7.934*	*1.63e-13 ***
chmax	*-2.972e-01*	*1.572e-01*	*-1.891*	*0.060103 .*
myct:cach	*-3.642e-03*	*1.288e-03*	*-2.828*	*0.005168 **
mmin:mmax	*3.475e-07*	*5.732e-08*	*6.063*	*6.80e-09 ***
mmin:cach	*-4.057e-05*	*1.316e-05*	*-3.082*	*0.002355 **
mmin:chmax	*2.703e-04*	*7.719e-05*	*3.501*	*0.000574 ***
mmax:chmax	*1.856e-05*	*9.947e-06*	*1.866*	*0.063585 .*

---

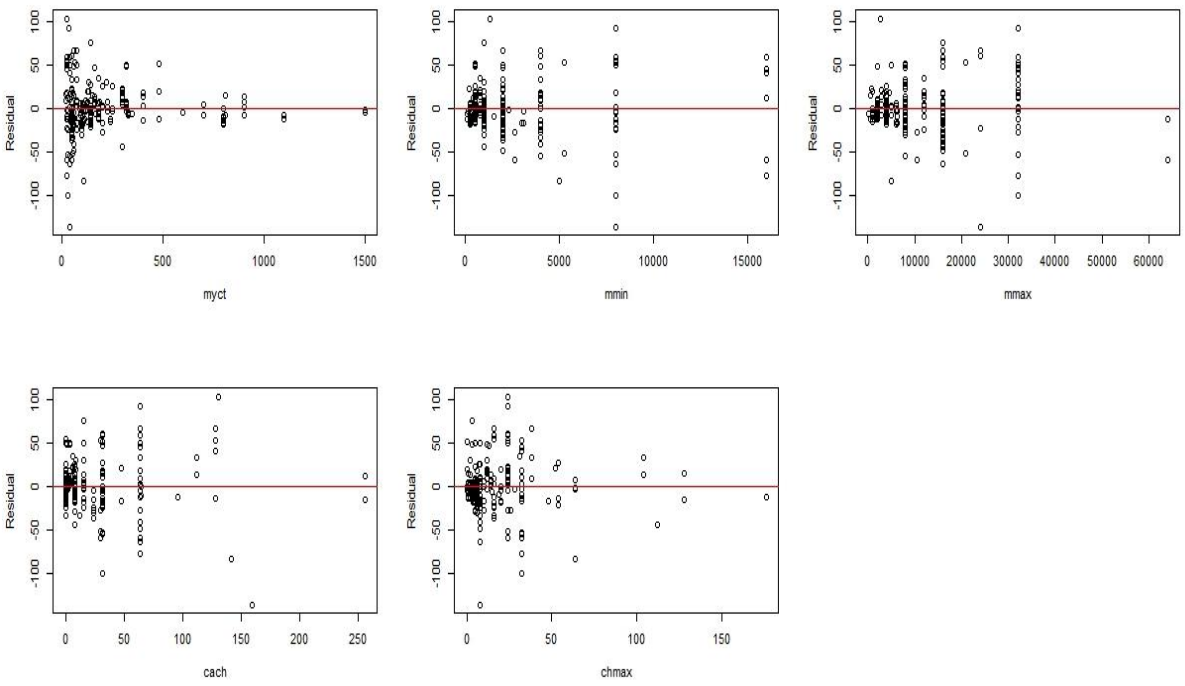
*Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1*

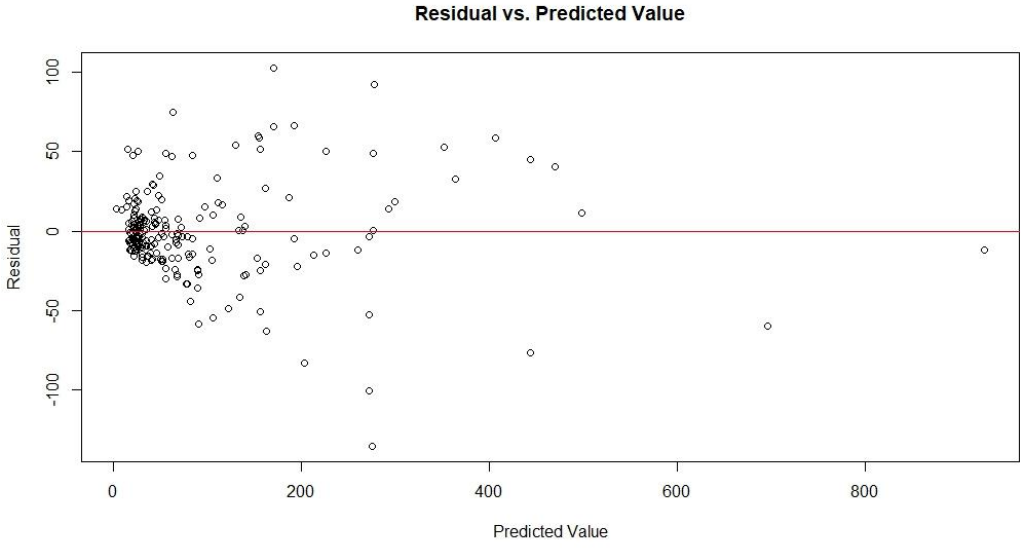
Residual standard error: 30.59 on 195 degrees of freedom

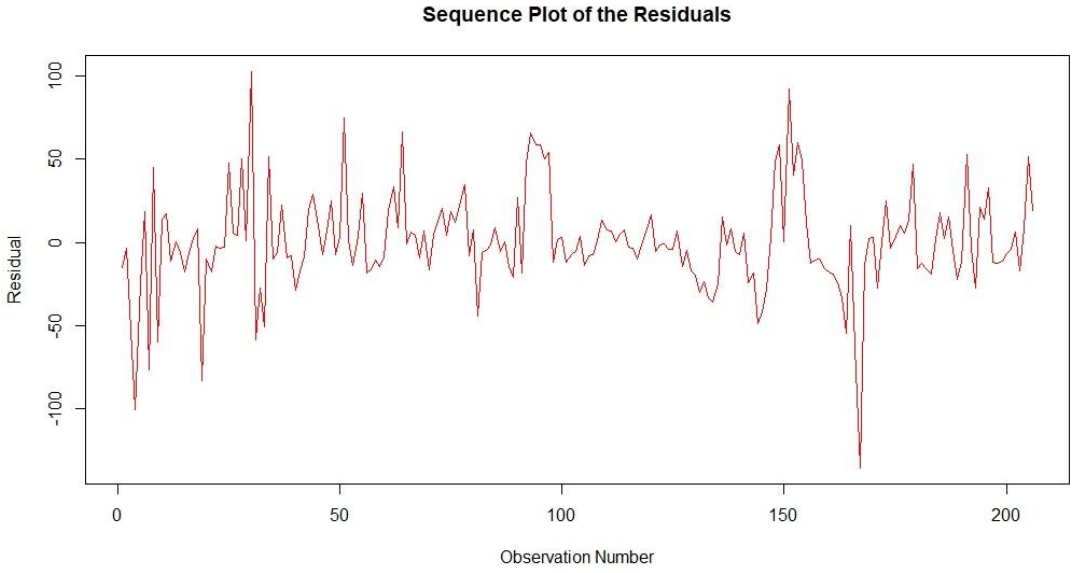
Multiple R-squared: 0.9415, Adjusted R-squared: 0.9385

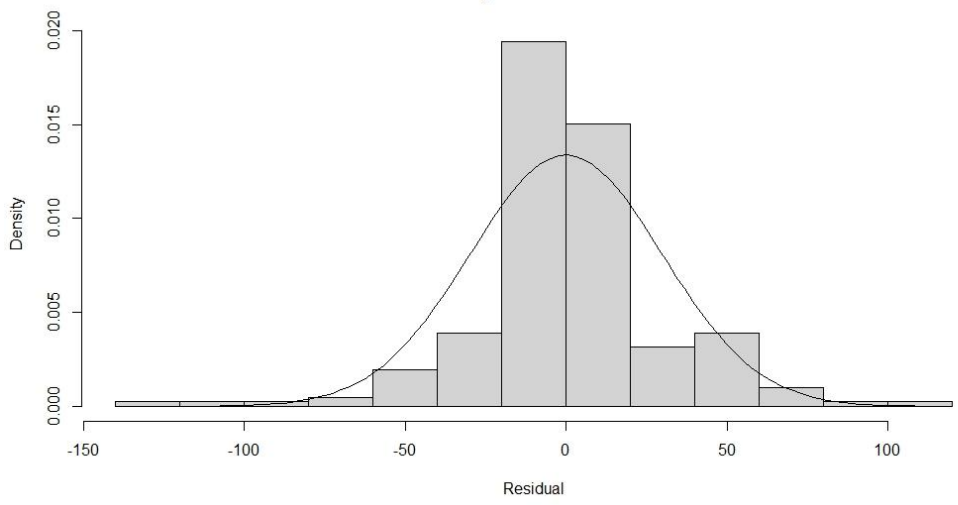
F-statistic: 313.6 on 10 and 195 DF, p-value: < 2.2e-16

CHECKING MODEL ASSUMPTIONS FOR OUR FINAL MODEL:




Residual vs. Predicted Value
Residual
Predicted Value
100
50
0
-50
-100
0
200
400
600
800


Sequence Plot of the Residuals
Residual
Observation Number
100
50
0
-50
-100
0
50
100
150
200


Histogram of Residual
Density
Residual
0.020
0.015
0.010
0.005
0.000
-150
-100
-50
0
50
100

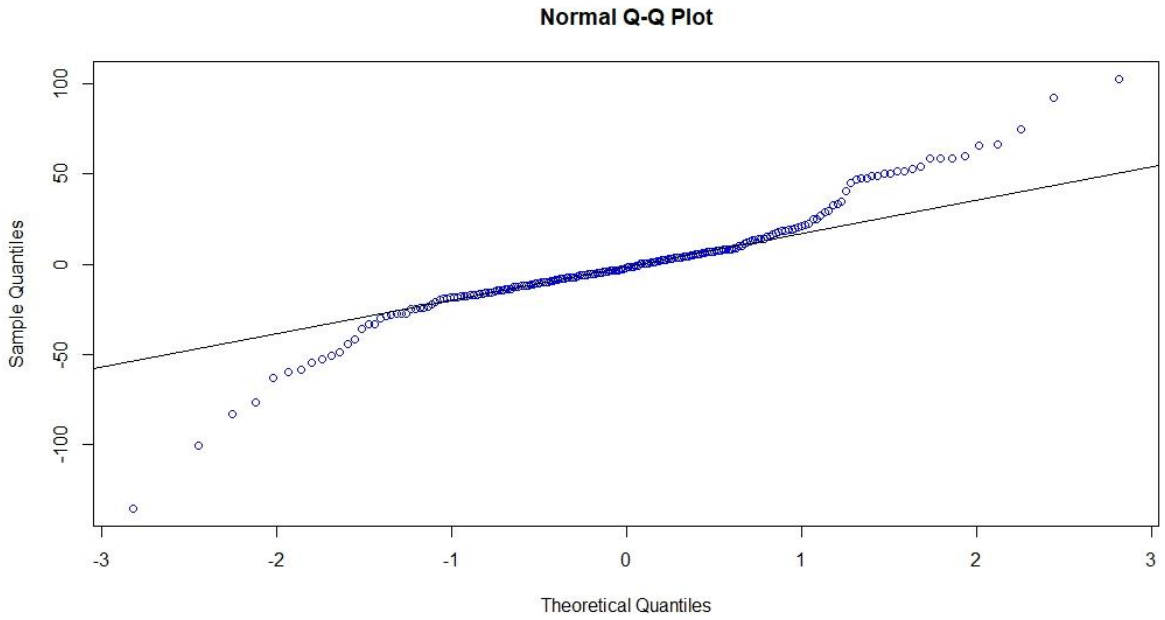


Fig 15,16,17,18,19;

The following plots were drawn to check the model assumptions. The Residual vs Explanatory variables show the correct specification of each explanatory variable. the Residual vs Predicted value plot does not show any discernible pattern meaning homoscedasticity. The time series plot does not show any pattern meaning the Independence of observations. Finally, the Normality was checked using histogram and Normal QQ plot. The bell-shaped curve in histogram and most of the observations closely following the straight line in QQ plot confirm the normality of the dataset. Thus, all the model assumptions are satisfied well.

MULTICOLLINEARITY:

The multicollinearity was checked using Variance Inflation factors. The VIF was reasonably low and it was higher for Interaction terms. The Overall large VIF's should be expected due to interaction terms. There is no effect of multicollinearity over here. The same was concluded from Correlation matrix of the variables.

CONCLUSION

The aim of the project was to demonstrate the model building and validation procedure. We have first checked for significant variables, then chose the best suitable model for the terms and then begun the diagnostics procedure. Model assumptions were evaluated at first and only after their satisfaction we further advanced towards outliers.

Outliers presented a unique challenge. We divided the outliers into two sets one being more outlying than the other. We first removed the more outlying set and evaluated the plots and the same was repeated with second set. The removal of second set did not provide any meaningful improvement to the model. Thus, we decided to retain them.

This process could not be carried further as removing more observations will result in low accuracy of the model.

Not all datasets are made for linear regression. Many other regression methods are there that should can be applied to the datasets which might explain the reoccurring outliers. But, nevertheless the process of model selection and building remains the same.

APPENDIX – I
R CODE

```
#Read the data with Appropriate column names;
data = read.delim(file.choose(),header=F,sep=",",col.names =
c('myct','mmin','mmax','cach','chmin','chmax','prp'))
attach(data)

model1 = lm(prp ~myct+mmin+mmax+cach+chmin+chmax, data)
summary(model1)

#partial F-tests on insignificant variables
model.r = lm(prp ~myct+mmin+mmax+cach+chmax)
summary(model.r)

anova(model1, model.r)

#scatter plot matrix and correlation matrix
library(car)
scatterplotMatrix(formula=~prp+myct+mmin+mmax+cach+chmax)

cor = round(cor(cbind(prp,myct,mmin,mmax,cach,chmax), method = "pearson"),4)
cor=as.data.frame(cor)
library("writexl")
write_xlsx(cor,"C:\\Users\\SHAARIF ANAS\\Desktop\\MAT 456\\Final Project\\corr.xlsx")

#Added variable plots
library(car)
avPlots(model=model.r)

#step wise model selection

#step wise using only square terms
modelsq.for = lm(prp~1)
sel.for.sq = step(modelsq.for, direction="forward", scope=list(lower=prp~1, upper=prp ~ myct
+ mmin + mmax + cach + chmax +
                                        I(myct^2) + I(mmin^2) + I(mmax^2) + I(cach^2) +
I(chmax^2)))
sel.for.sq
summary(sel.for.sq)

modelsq.back = lm(prp ~ myct + mmin + mmax + cach + chmax +
            I(myct^2) + I(mmin^2) + I(mmax^2) + I(cach^2) + I(chmax^2))
```

```
sel.back.sq = step(modelsq.back, direction="backward", scope=list(lower=prp~1, upper=prp ~
myct + mmin + mmax + cach + chmax +
                                        I(myct^2) + I(mmin^2) + I(mmax^2) + I(cach^2) +
I(chmax^2)))
sel.back.sq
summary(sel.back.sq)

#step wise selection using only interaction term
modelint.for = lm(prp~1)
sel.for.int = step(modelint.for, direction="forward", scope=list(lower=prp~1, upper=prp ~
myct + mmin + mmax + cach + chmax +
                                    myct:mmin + myct:mmax + myct:cach + myct:chmax +
                                    mmin:mmax + mmin:cach + mmin:chmax +
                                    mmax:cach + mmax:chmax +
                                    cach:chmax))
sel.for.int
summary(sel.for.int)

modelint.back = lm(prp ~ myct + mmin + mmax + cach + chmax +
            myct:mmin + myct:mmax + myct:cach + myct:chmax +
            mmin:mmax + mmin:cach + mmin:chmax +
            mmax:cach + mmax:chmax +
            cach:chmax)
sel.back.int = step(modelint.back, direction="backward", scope=list(lower=prp~1, upper=prp
~ myct + mmin + mmax + cach + chmax +
                                        myct:mmin + myct:mmax + myct:cach + myct:chmax +
                                        mmin:mmax + mmin:cach + mmin:chmax +
                                        mmax:cach + mmax:chmax +
                                        cach:chmax))
sel.back.int
summary(sel.back.int)

#stepwise selection using both square and interaction term
model1.for = lm(prp~1)
sel.for.1 = step(model1.for, direction="forward", scope=list(lower=prp~1, upper=prp ~ myct +
mmin + mmax + cach + chmax +
                                    I(myct^2) + I(mmin^2) + I(mmax^2) + I(cach^2) +
I(chmax^2) +
                                    myct:mmin + myct:mmax + myct:cach + myct:chmax +
                                    mmin:mmax + mmin:cach + mmin:chmax +
                                    mmax:cach + mmax:chmax +
                                    cach:chmax))
sel.for.1
summary(sel.for.1)
```

```
model1.back = lm(prp ~ myct + mmin + mmax + cach + chmax +
                 myct:mmin + myct:mmax + myct:cach + myct:chmax +
                 I(myct^2) + I(mmin^2) + I(mmax^2) + I(cach^2) + I(chmax^2) +
                 mmin:mmax + mmin:cach + mmin:chmax +
                 mmax:cach + mmax:chmax +
                 cach:chmax)
sel.back.1 = step(model1.back, direction="backward", scope=list(lower=prp~1, upper=prp ~
myct + mmin + mmax + cach + chmax +
                                        I(myct^2) + I(mmin^2) + I(mmax^2) + I(cach^2) +
I(chmax^2) +
                                        myct:mmin + myct:mmax + myct:cach + myct:chmax +
                                        mmin:mmax + mmin:cach + mmin:chmax +
                                        mmax:cach + mmax:chmax +
                                        cach:chmax))
sel.back.1
summary(sel.back.1)

#Function for AIC and BIC for First order model;
aic.bic = function (model){
  mod.fit = lm(model)
  aic.mod = AIC(mod.fit)
  bic.mod = BIC(mod.fit)
  data.frame(AIC=aic.mod, BIC=bic.mod)
}

sel.for.sq = aic.bic(sel.for.sq)
sel.back.sq = aic.bic(sel.back.sq)
sel.for.int = aic.bic(sel.for.int)
sel.back.int = aic.bic(sel.back.int)
sel.for.1 = aic.bic(sel.for.1)
sel.back.1 = aic.bic(sel.back.1)
model.r = aic.bic(model.r)
model1 = aic.bic(model1)



aic_bic = rbind(sel.for.sq, sel.back.sq, sel.for.int, sel.back.int,sel.for.1, sel.for.1,
model.r,model1)
mod.name =  c('modelsq.for', 'modelsq.back', 'modelint.for', 'modelint.back', 'model1.for',
'model1.back','model.r', 'model1')

aic_bic.table = as.data.frame(cbind(mod.name,aic_bic))
```

```
#Write the table results as .xlsx file
library("writexl")
write_xlsx(aic_bic.table,"C:\\Users\\SHAARIF ANAS\\Desktop\\MAT 456\\Final
Project\\Model AIC BIC.xlsx")


#chosen best model
model.best = lm(prp ~ myct + mmin + mmax + cach + chmax + myct:cach +
                mmin:mmax + mmin:cach + mmin:chmax + mmax:chmax)
summary(model.best)

#DIAGNOSTIC PLOTS FOR OUTLIERS

n = nrow(data)
p = ncol(data)

# Outliers
e_star <- model.best$residuals/summary(model.best)$sigma
plot(x=model.best$fitted.values, y=e_star, xlab='Predicted Value', ylab='RStudent',
main='Studentized Residual vs. Predicted Value', )
abline(h=1.96, col='red')
abline(h=-1.96, col='red')

#leverage
h.ii <- hatvalues(model=model.best)
p <- length(model.best$coefficients)
round(h.ii[h.ii>2*p/n], 4)

# rstudent vs. leverage
plot(x=h.ii, y=e_star, xlab='Leverage', ylab='Rstudent', main = 'Studentized Residual vs.
Leverage')
abline(v=2*p/n, col='red')
abline(h=1.96, col='red')
abline(h=-1.96, col='red')

# DFFITS vs. observation number
dffits.data <- dffits(model=model.best)
dffits.data[abs(dffits.data)>2*sqrt(p/n)]

plot(x=1:n, y=dffits.data, xlab='Observation number', ylab='DFFITS', main='DFFITS vs.
observation number')
abline(h=0)
abline(h=c(-2*sqrt(p/n), 2*sqrt(p/n)), col='red')
```

```
# cook's distance
cook.i <- cooks.distance(model=model.best)
plot(x=1:n, y= cook.i, xlab='Observation number', ylab="Cook's D", main = "Cook's D vs.
observation number")
abline(h=4/n, col='red')


#WORKING ON OUTLIERS
#Studentized deleted residuals;
alpha = 0.05
n = nrow(data)
t.d = rstudent(model.best)

#Bonferronis Test for outliers;
qt(p=1-alpha/(2*n), df=model.best$df.residual-1)
t.d[abs(t.d) > qt(p=1-alpha/(2*n), df=model.best$df.residual-1)]

#DFFITS, DFBETAS and Cook's Distance;
dffits.data = dffits(model.best)
cook.data = cooks.distance(model.best)
dfbeta.data = dfbetas(model.best)

#Check for Influence;
n = nrow(data)
p = ncol(data)
dffits.data[dffits.data > 2*(sqrt(p/n))]
cook.data[cook.data > qf(0.50,p,(n-p))]
dfbeta.data[dfbeta.data > 2/(sqrt(n))]

#leverage
h.ii <- hatvalues(model=model.best)
p <- length(model.best$coefficients)
round(h.ii[h.ii>2*p/n], 4)

#Add an observation number variable to existing data set;
library(dplyr)
data = data %>%
  mutate(obs = 1:209)

#Create new data frame without 10,32,200 observation
data.206 = data[c(-10,-32,-200),]
attach(data.206)
model.best.206 =lm(prp ~ myct + mmin + mmax + cach + chmax + myct:cach +
                     mmin:mmax + mmin:cach + mmin:chmax + mmax:chmax, data.206)
```

```
summary(model.best.206)

#Predict Y variables with regression equation of 209 and 206 observations respectively;
y.209 = predict(model.best,data.206)
y.206 = predict(model.best.206,data.206)

#create another variable for percent change in y;
y.per.chg = (((y.206 - y.209)/ y.209)*100)
mean(abs(y.per.chg))
#create a table with obs number, y.209, y.206 and y.per.chg;
y.table = data.frame(data.206$obs,y.209,y.206,y.per.chg)

#Write the table results as .xlsx file
library("writexl")
write_xlsx(y.table,"C:\\Users\\SHAARIF ANAS\\Desktop\\MAT 456\\Final
Project\\y.table.xlsx")

#CHECKING AFTER REMOVING 10,32 AND 200 th OBSERVATION

n = nrow(data.206)
p = ncol(data.206)

# Outliers
e_star <- model.best.206$residuals/summary(model.best.206)$sigma
plot(x=model.best.206$fitted.values, y=e_star, xlab='Predicted Value', ylab='RStudent',
main='Studentized Residual vs. Predicted Value', )
abline(h=1.96, col='red')
abline(h=-1.96, col='red')

#leverage
h.ii <- hatvalues(model=model.best.206)
p <- length(model.best.206$coefficients)
round(h.ii[h.ii>2*p/n], 4)

# rstudent vs. leverage
plot(x=h.ii, y=e_star, xlab='Leverage', ylab='Rstudent',main = 'Studentized Residual vs.
Leverage')
abline(v=2*p/n, col='red')
abline(h=1.96, col='red')
abline(h=-1.96, col='red')

# DFFITS vs. observation number
n = nrow(data.206)
p = ncol(data.206)
```

```
dffits.data.206 <- dffits(model=model.best.206)
dffits.data.206[abs(dffits.data.206)>2*sqrt(p/n)]

plot(x=1:n, y=dffits.data.206, xlab='Observation number', ylab='DFFITS', main='DFFITS vs.
observation number')
abline(h=0)
abline(h=c(-2*sqrt(p/n), 2*sqrt(p/n)), col='red')

# cook's distance
cook.i <- cooks.distance(model=model.best.206)
plot(x=1:n, y= cook.i, xlab='Observation number', ylab="Cook's D",main = "Cook's D vs.
observation number")
abline(h=4/n, col='red')



#REMOVING ANOTHER SET OF OBSERVATIONS
#Create new data frame without  observation
data.197 = data.206[c(-30,-94,-95,-136,-152,-155,-165,-197,-205),] #observation numbers
change as 3 observations are already removed
attach(data.197)
model.best.197 =lm(prp ~ myct + mmin + mmax + cach + chmax + myct:cach +
                   mmin:mmax + mmin:cach + mmin:chmax + mmax:chmax, data.197)
summary(model.best.197)

#Predict Y variables with regression equation of 209 and 206 observations respectively;
y.197 = predict(model.best.197,data.197)
y.206 = predict(model.best.206,data.197)

#create another variable for percent change in y;
y.per.chg = (((y.197 - y.206)/ y.206)*100)
mean(abs(y.per.chg))

#create a table with obs number, y.209, y.206 and y.per.chg;
y.table = data.frame(data.197$obs,y.206,y.197,y.per.chg)

#Write the table results as .xlsx file
library("writexl")
write_xlsx(y.table,"C:\\Users\\SHAARIF ANAS\\Desktop\\MAT 456\\Final
Project\\y.table197.xlsx")

#CHECKING AFTER REMOVING another set of OBSERVATION
n = nrow(data.197)
p = ncol(data.197)
```

```
# Outliers
e_star <- model.best.197$residuals/summary(model.best.197)$sigma
plot(x=model.best.197$fitted.values, y=e_star, xlab='Predicted Value', ylab='RStudent',
main='Studentized Residual vs. Predicted Value', )
abline(h=1.96, col='red')
abline(h=-1.96, col='red')

#leverage
h.ii <- hatvalues(model=model.best.197)
p <- length(model.best.197$coefficients)
round(h.ii[h.ii>2*p/n], 4)

# rstudent vs. leverage
plot(x=h.ii, y=e_star, xlab='Leverage', ylab='Rstudent',main = 'Studentized Residual vs.
Leverage')
abline(v=2*p/n, col='red')
abline(h=1.96, col='red')
abline(h=-1.96, col='red')

# DFFITS vs. observation number
dffits.data.197 <- dffits(model=model.best.197)
dffits.data.197[abs(dffits.data.197)>2*sqrt(p/n)]

plot(x=1:n, y=dffits.data.197, xlab='Observation number', ylab='DFFITS', main='DFFITS vs.
observation number')
abline(h=0)
abline(h=c(-2*sqrt(p/n), 2*sqrt(p/n)), col='red')

# cook's distance
cook.i <- cooks.distance(model=model.best.197)
plot(x=1:n, y= cook.i, xlab='Observation number', ylab="Cook's D",main = "Cook's D vs.
observation number")
abline(h=4/n, col='red')


#final model
model.final =lm(prp ~ myct + mmin + mmax + cach + chmax + myct:cach +
                mmin:mmax + mmin:cach + mmin:chmax + mmax:chmax, data.206)
summary(model.final)

#Checking Model Assumptions

n = nrow(data.206)
```

```
p = ncol(data.206)
#Correct specification of explanatory variables;
par(mfrow=c(2,3))
title(main = 'Residuals vs. Explanatory variables')
plot(x=myct, y=model.final$residuals, xlab='myct', ylab='Residual')
abline(h=0, col='red')

plot(x=mmin, y=model.final$residuals, xlab='mmin', ylab='Residual')
abline(h=0, col='red')

plot(x=mmax, y=model.final$residuals, xlab='mmax', ylab='Residual')
abline(h=0, col='red')

plot(x=cach, y=model.final$residuals, xlab='cach', ylab='Residual')
abline(h=0, col='red')

plot(x=chmax, y=model.final$residuals, xlab='chmax', ylab='Residual')
abline(h=0, col='red')

# e vs. Y_hat
plot(x=model.final$fitted.values, y=model.final$residuals, xlab='Predicted Value',
ylab='Residual', main='Residual vs. Predicted Value')
abline(h=0, col='red')

# Independence
plot(x=seq(1:n), y=model.final$residuals, xlab='Observation Number', ylab='Residual',
main='Sequence Plot of the Residuals', type='l', col='Red')

# Normality
hist(model.final$residuals, prob=TRUE, xlab='Residual', main='Histogram of Residual')
curve(dnorm(x, mean=mean(model.final$residuals), sd=sd(model.final$residuals)),
add=TRUE)

qqnorm(model.final$residuals, col='Blue')
qqline(model.final$residuals)

#VIF
vif(model.final)
```