

STAT 378 Project

Bao Nguyen, Adhiguna Pande, Zamaan Hussein

December 9, 2023

Abstract

In this paper we attempt to create and fit a statistical model to predict the number of serious crimes in a region based upon a set of factors. We use a data set containing 141 large Standard Metropolitan Statistical Areas (SMSAs) to train and test the model.

1 Introduction

Throughout time the mitigation and prediction of crime has been a problem that law enforcement and governments have struggled with. Using the SMSA dataset that we have been provided we set out to discover a linear relationship between the data and the number of total serious crimes that occur in a Standard Metropolitan Statistical Area. The data set contains an identification number for each SMSA as well as information on 11 other variables which we will refer to by the abbreviations enumerated here:

LA	Land Area (in square miles)
TP	Total population (in thousands)
PPCC	Percent of population in central cities
PPA	Percent of population 65 or older
NAP	Number of active physicians
NHB	Number of hospital beds
PHG	Percent high school graduates
CLF	Civilian labor force
TPI	Total personal income (in millions of dollars)
TSC	Total serious crimes
GR	Geographic region (W, NE, NC, S)

Table 1: Variables and abbreviations

TSC is our response variable and we will be conducting an analysis using all other variables as regressors. We choose West (W) region as our base/reference category to represent the categorical variable of Geographical region. We are attempting to test against the null hypothesis that none of the observations collected here are significant for predicting TSI.

2 Model Development

2.1 Data Splitting

Before beginning any statistical analysis on the data set we first split the data into 2 data sets using the DUPLEX algorithm. We do this in order to prepare a training data set which we will use to build our model, and a holdout data set which we will verify our model's predictive power against. The DUPLEX algorithm was chosen over a random seed method for data splitting due to its ability to create 2 sets with very similar statistical properties. We isolated 25 observations in the holdout group and trained the model with the remaining 116 results. The code for the DUPLEX algorithm used to split the data is in appendix A for perusal.

2.2 Variable Analysis

We start by plotting the variables and analyzing the results. From a Normal QQ-Plot we see a significant deviation from the reference line implying that the data may not be normally distributed; violating the normality assumption of linear regression.

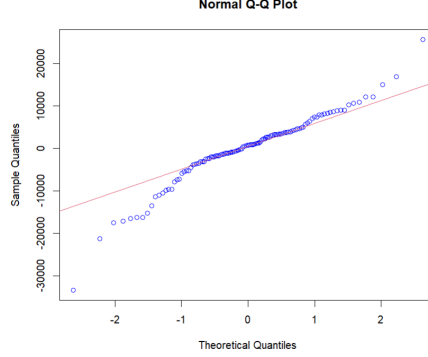


Figure 1: Normal Q-Q Plot

By plotting the R-Student residuals against the predicted response we can observe a random non-linear pattern implying that the constant variance assumption holds. Next we plot the residuals against the regressors to see that all but

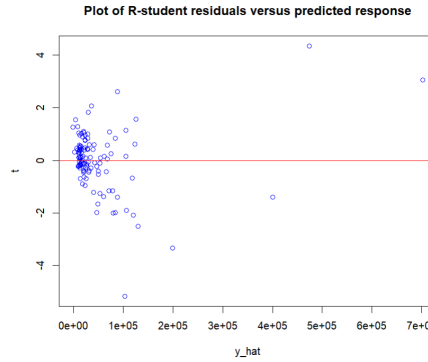


Figure 2: R-Student Residuals vs Predicted Response

Finally we plotted the partial residuals for each regressor to see that it lends credence to the necessity of transforming some of the variables. The plots for this section are in Appendix B

2.3 Transformations

After looking at the original variables it becomes clear that transformations are necessary in order to stabilize the variance. We applied these transformations to the variables:

$$LA_t = \sqrt{LA} \quad (1)$$

$$TP_t = \log(TP) \quad (2)$$

$$PPCC_t = PPCC \quad (3)$$

$$PPA_t = PPA \quad (4)$$

$$NAP_t = \log(NAP) \quad (5)$$

$$NHB_t = \log(NHB) \quad (6)$$

$$PHSG_t = PHSG \quad (7)$$

$$CLF_t = \log(CLF) \quad (8)$$

$$TPI_t = \log(TPI) \quad (9)$$

We used log and square root transforms in order to stabilize the variance and mitigate the relationship between the variance and the mean where it was necessary.

2.4 Variable Comparison

After applying the transformations we now compare the transformed model to the original model.

Comparing the transformed QQ Plot to the original model it is clear to see that the data fits the reference line significantly better; showing that the transformed data satisfies the normality assumption for linear regression.

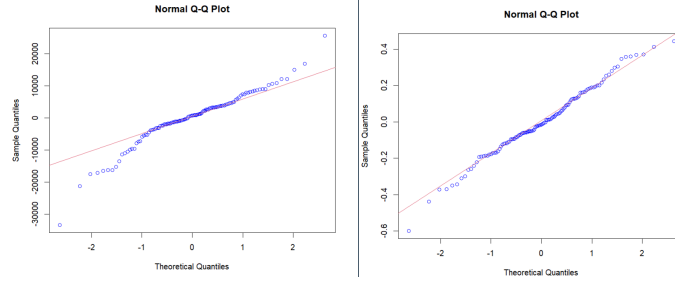


Figure 3: Normal QQPlots Original vs Transformed

Next we move on to the R-student residual against predicted response and both models maintain a random pattern leading us to believe that the constant variance assumption holds for both.

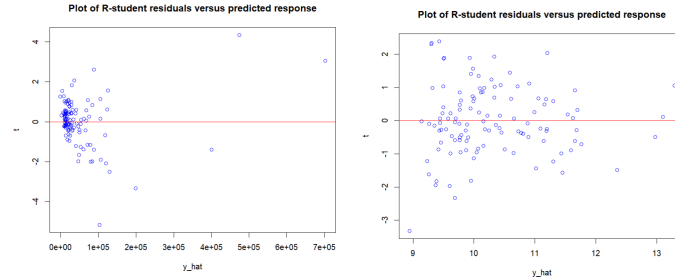


Figure 4: R-Student Residuals Original vs Transformed

Finally, the partial residual plots of the transformed data doesn't deviate significantly from the straight line. This indicates that no further transformation is required for any of the variables. The partial residual plots can be found in Appendix C.

Based on these comparisons it makes the most sense to continue our analysis using these transformed variables since they satisfy the assumptions for linear regression.

Hence, the linear model under consideration is:

$$\begin{aligned} \ln(TSC) = & \beta_0 + \beta_1 \sqrt{LA} + \beta_2 \ln(TP) + \beta_3(PPCC) + \beta_4(PPA) + \beta_5 \ln(NAP) + \beta_6 \ln(NHB) \\ & + \beta_7(PHSG) + \beta_8 \ln(CLF) + \beta_9 \ln(TPI) + \beta_{10}(NE) + \beta_{11}(NC) + \beta_{12}(S) + \epsilon \end{aligned}$$

2.5 Interaction Terms

After settling on the transformed variables we begin interaction terms between the transformed variables. Through testing we found that the interaction terms $PHSG \cdot \log(TPI)$ as well as $PPCC \cdot PPA$ are both statistically significant with P-values less than 0.05. We include them in the newly transformed model to be examined for our final variable selection methods.

The new proposed model is:

$$\ln(TSC) = \beta_0 + \beta_1 \sqrt{LA} + \beta_2 \ln(TP) + \beta_3(PPCC) + \beta_4(PPA) + \beta_5 \ln(NAP) + \beta_6 \ln(NHB)$$

$$+\beta_7(PHSG)+\beta_8 \ln (CLF)+\beta_9 \ln (TPI)+\beta_{10}(NE)+\beta_{11}(NC)+\beta_{12}(S)+\beta_{13}(PHSG \times \ln(TPI))+\beta_{14}(PPCC \times PPA)+\epsilon$$

Testing significance of PHSG and log(TPI): $H_0 : \beta_{13} = 0$ versus $H_1 : \beta_{13} \neq 0$

```
> anova(transformed_model, transformed_model_PHSG_TPI)
Analysis of variance Table

Model 1: log(TSC) ~ sqrt(LA) + log(TP) + PPCC + PPA + log(NAP) + log(NHB) +
  PHSG + log(CLF) + log(TPI) + NE + NC + S
Model 2: log(TSC) ~ sqrt(LA) + log(TP) + PPCC + PPA + log(NAP) + log(NHB) +
  PHSG + log(CLF) + log(TPI) + NE + NC + S + PHSG * log(TPI)
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     103 4.1832
2     102 3.9950  1    0.1882 4.8051 0.03065 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 5: Partial F-test for PHSG and log(TPI) Interaction

From the R-output, we see that the partial F-test returns a test statistic of 4.8051 and a p-value of 0.03065. The distribution of test statistic under null hypothesis is F-distribution with numerator df of 1 and denominator df of 102. At a significance level of 5%, we conclude that interaction between PHSG and log(TPI) is significant for predicting log(TSC) after accounting for the effect of all other predictors since the data provides strong evidence against H_0 .

Testing significance of PPCC and PPA: $H_0 : \beta_{14} = 0$ versus $H_1 : \beta_{14} \neq 0$

```
> anova(transformed_model, transformed_model_PPCC_PPA)
Analysis of variance Table

Model 1: log(TSC) ~ sqrt(LA) + log(TP) + PPCC + PPA + log(NAP) + log(NHB) +
  PHSG + log(CLF) + log(TPI) + NE + NC + S
Model 2: log(TSC) ~ sqrt(LA) + log(TP) + PPCC + PPA + log(NAP) + log(NHB) +
  PHSG + log(CLF) + log(TPI) + NE + NC + S + PPCC * PPA
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     103 4.1832
2     102 4.0189  1    0.16426 4.169 0.04375 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 6: Partial F-test for PPCC and PPA Interaction

From the R-output, we see that the partial F-test returns a test statistic of 4.169 and a p-value of 0.04375. The distribution of test statistic under null hypothesis is F-distribution with numerator df of 1 and denominator df of 102. At a significance level of 5%, we conclude that interaction between PPCC and PPA is significant for predicting log(TSC) after accounting for the effect of all other predictors since the data provides strong evidence against H_0 .

2.6 Variable Selection

The above interaction terms were added to the transformed model, forming the final model including all possible variables and interaction terms. From here, forward, backward, and stepwise selection were used based on a null model with no regressors and aforementioned final model to choose suitable variables for reduced models.

The models' AIC, BIC, adjusted R^2 , Mallows' Cp and PRESS statistics were calculated and summarized in the table below:

	Final	Transformed	Forward	Backward	Stepwise
AIC	-33.73	-28.22	-34.52	-37	-34.52
BIC	10.33	10.33	-6.99	1.55	-6.99
Adjusted R^2	0.9511	0.9479	0.9491	0.9517	0.9491
Mallows' Cp		19.63	13.25	11.64	13.25
PRESS	5.656	5.553	5.152	5.174	5.152

Note that the Mallows' Cp values calculated were with comparison to the final model, so the final model does not have Mallows' Cp. With high values of R_{adj}^2 , we can see that every model accurately fits the training data. Since the backward model has the lowest AIC and Mallows' Cp while giving the highest adjusted R^2 , the backward model will be chosen to test for validity and further analysis.

3 Model Validation

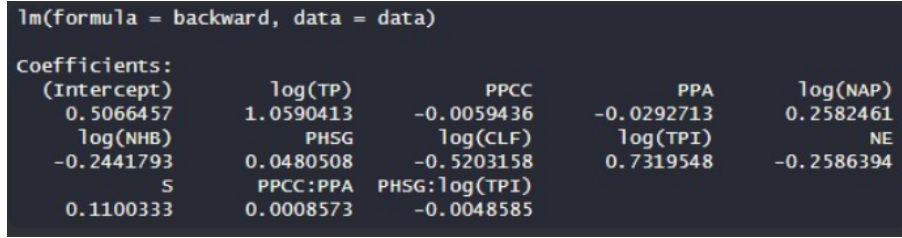
3.1 PRESS Statistic

We calculate the PRESS Statistic for the backward model to be: 5.174 which indicates that the model does a good job at predicting new observations that are not in training data.

Using this we calculate an R_{pred}^2 to be: $R_{pred}^2 = 1 - \frac{PRESS}{SS_{Total}} > 0.90$ With $R_{pred}^2 > 0.90$ we can state that the data is useful for predicting additional results and that the model accurately fits the data with minimal noise.

3.2 Coefficient Analysis

Using our chosen model using backward selection, we estimate the following coefficients using training data:



```
lm(formula = backward, data = data)

Coefficients:
(Intercept)      log(TP)          PPCC          PPA      log(NAP)
  0.5066457    1.0590413   -0.0059436   -0.0292713    0.2582461
log(NHB)        PHSG        log(CLF)      log(TPI)         NE
-0.2441793    0.0480508   -0.5203158    0.7319548   -0.2586394
S      PPCC:PPA  PHSG:log(TPI)
 0.1100333    0.0008573   -0.0048585
```

Figure 7: Model coefficients

Hence, estimated model equation is:

$$\ln(\hat{TSC}) = 0.5066457 + 1.0590413 \ln(TP) - 0.0059436(PPCC) - 0.0292713(PPA) + 0.2582461 \ln(NAP) - 0.2441793 \ln(NHB) + 0.0480508(PHSG) - 0.5203158 \ln(CLF) + 0.7319548 \ln(TPI) - 0.2586394(NE) + 0.1100333(S) + 0.0008573(PPCC \times PPA) - 0.0048585(PHSG \times \ln(TPI))$$

Interpreting model coefficients:

⇒ It is estimated that the mean of $\log(TSC)$ is 0.5066457 when all transformed predictors have value of 0.

⇒ It is estimated that the mean of $\log(TSC)$ increases by 1.0590413 when $\log(TP)$ increases by one unit, while keeping the value of all other transformed variables fixed.

⇒ It is estimated that when PPA is zero, the mean of $\log(TSC)$ decreases by 0.0059436 for one unit increase in PPCC, while keeping the values of all other transformed variables fixed.

⇒ It is estimated that when PPCC is zero, the mean of $\log(TSC)$ decreases by 0.0292713 for one unit increase in PPA, while keeping the values of all other transformed variables fixed.

⇒ It is estimated that the mean of $\log(TSC)$ increases by 0.25824 when $\log(NAP)$ increases by one unit, while keeping the value of all other transformed variables fixed.

⇒ It is estimated that the mean of $\log(TSC)$ decreases by 0.2441793 when $\log(NHB)$ increases by one unit, while keeping the value of all other transformed variables fixed.

⇒ It is estimated that when $\log(TPI)$ is zero, the mean of $\log(TSC)$ increases by 0.0480508 for one unit increase in PHSG, while keeping the values of all other transformed variables fixed.

⇒ It is estimated that the mean of $\log(TSC)$ decreases by 0.52031 when $\log(CLF)$ increases by one unit, while keeping the value of all other transformed variables fixed.

⇒ It is estimated that when PHSG is zero, the mean of $\log(TSC)$ increases by 0.7319 for one unit increase in $\log(TPI)$, while keeping the values of all other transformed variables fixed.

⇒ It is estimated that the mean of $\log(TSC)$ decreases by 0.2586394 for geographical region NE when compared to geographical region W.

⇒ It is estimated that the mean of $\log(TSC)$ increases by 0.1100333 for geographical region S when

compared to geographical region W.

⇒ It is estimated that the each additional unit in PPCC increases the effectiveness of PPA on mean of $\log(\text{TSC})$ by 0.0008573. Similarly, each additional unit in PPA increases the effectiveness of PPCC on mean of $\log(\text{TSC})$ by 0.0008573.

⇒ It is estimated that the each additional unit in PHSG decreases the effectiveness of $\log(\text{TPI})$ on mean of $\log(\text{TSC})$ by 0.0048585. Similarly, each additional unit in $\log(\text{TPI})$ decreases the effectiveness of PHSG on mean of $\log(\text{TSC})$ by 0.0048585.

Confidence intervals for estimated coefficients:

```
> confint(backward, level = 0.95)
              2.5 %      97.5 %
(Intercept) -2.026730e+00  3.091886e+00
log(TP)      7.950983e-01  1.765034e+00
PPCC        -1.359777e-02  3.260002e-03
PPA         -6.726651e-02  5.062563e-03
log(NAP)     1.532681e-01  4.953790e-01
log(NHB)    -4.001702e-01 -1.541372e-01
PHSG         7.848181e-03  1.013910e-01
log(CLF)    -1.009643e+00  3.017705e-02
log(TPI)    -3.858085e-02  1.049802e+00
NE          -2.988091e-01 -7.218556e-02
S           1.746364e-02  2.271504e-01
PPCC:PPA    -4.769244e-05  1.820465e-03
PHSG:log(TPI) -1.135445e-02  3.297748e-05
```

Figure 8: Confidence intervals

⇒ It is estimated with 95% confidence that the mean of $\log(\text{TSC})$ is between -2.0267 and 3.0918 when all transformed predictors have value of 0.

⇒ It is estimated with 95% confidence that increase in the mean of $\log(\text{TSC})$ when $\log(\text{TP})$ increases by 1 unit is between 0.795 and 1.765.

⇒ It is estimated with 95% confidence that change in the mean of $\log(\text{TSC})$ when PPCC increases by 1 unit is between -0.0136 and 0.00326.

⇒ It is estimated with 95% confidence that change in the mean of $\log(\text{TSC})$ when PPA increases by 1 unit is between -0.06726 and 0.005062.

⇒ It is estimated with 95% confidence that increase in the mean of $\log(\text{TSC})$ when $\log(\text{NAP})$ increases by 1 unit is between 0.15326 and 0.49538.

⇒ It is estimated with 95% confidence that decrease in the mean of $\log(\text{TSC})$ when $\log(\text{NHB})$ increases by 1 unit is between 0.40017 and 0.1541372.

⇒ It is estimated with 95% confidence that increase in the mean of $\log(\text{TSC})$ when PHSG increases by 1 unit is between 0.00785 and 0.1014.

⇒ It is estimated with 95% confidence that change in the mean of $\log(\text{TSC})$ when $\log(\text{CLF})$ increases by 1 unit is between -1.009643 and 0.03017.

⇒ It is estimated with 95% confidence that change in the mean of $\log(\text{TSC})$ when $\log(\text{TPI})$ increases by 1 unit is between -0.03858 and 1.0498.

⇒ It is estimated with 95% confidence that mean of $\log(\text{TSC})$ is between 0.007218 to 0.2988 less than mean $\log(\text{TSC})$ of geographical region W.

⇒ It is estimated with 95% confidence that mean of $\log(\text{TSC})$ is between 0.017463 to 0.022715 higher than mean $\log(\text{TSC})$ of geographical region W.

3.3 Testing Holdout Data

```
> backward_est_pred
[1] 0.1
> |
```

Figure 9: Backward Model Validity

Upon testing the backward model against holdout data, $R_{est}^2 = 0.96$ and $r_{y,\hat{y}}^2 = 0.86$. Clearly, $|R_{est}^2 - r_{y,\hat{y}}^2| = 0.1$. Since $|R_{est}^2 - r_{y,\hat{y}}^2| \not\geq 0.1$, we conclude that the model is valid for prediction on new data.

3.4 Influential Points/Leverage

3.4.1 Training Data

Fitting training data into the backward model, the following observations are found to be influential points : 1, 7, 24, 49, 69, 78, 102, 118, 127, 138, 141

3.4.2 All Data

Fitting all data into the backward model, the following observations are found to be influential points: 1, 24, 49, 69, 105, 127, 133, 138, 141

3.5 Multicollinearity

```
> coll_backward
Tolerance and Variance Inflation Factor
```

	variables	Tolerance	VIF
1	log(TP)	0.009207305	108.609417
2	PPCC	0.054451843	18.364851
3	PPA	0.172530543	5.796075
4	log(NAP)	0.047579892	21.017282
5	log(NHB)	0.120244346	8.316399
6	PHSG	0.009015430	110.920942
7	log(CLF)	0.007525223	132.886436
8	log(TPI)	0.006109362	163.683222
9	NE	0.605984401	1.650207
10	S	0.508257130	1.967508
11	PPCC:PPA	0.056710455	17.633433
12	PHSG:log(TPI)	0.005039969	198.413938

Figure 10: Enter Caption

With the VIF values above, it is clear that this model has multicollinearity issues. However, there are multiple approaches to correct multicollinearity. We try to correct for multicollinearity by centering variables, interaction terms and dropping some variables with high VIF.

$$\begin{aligned} \log(TSC) = & \beta_0 + \beta_1 PPCC + \beta_2 PPA + \beta_3 \log(NAP) + \beta_4 PHSG + \beta_5 \log(NHB) \\ & + \beta_6 ((PPCC - \text{mean}(PPCC)) \times (PPA - \text{mean}(PPA))) \\ & + \beta_7 ((PHSG - \text{mean}(PHSG)) \times (\log(TPI) - \text{mean}(\log(TPI)))) \\ & + \beta_8 ((\log(NAP) - \text{mean}(\log(NAP))) \times (\log(TPI) - \text{mean}(\log(TPI)))) \end{aligned}$$

We find the VIF values for the above model to be < 10 which means there is no significant multicollinearity. However, this comes at the cost of reduced predictive power of the model since dropping some variables lead to loss of information from the data.

```
> ols_coll_diag(backward_alt)
Tolerance and Variance Inflation Factor
```

	Variables	Tolerance	VIF
1	PPCC_t	0.6630896	1.508092
2	PPA_t	0.6663444	1.500726
3	log(NAP_t)	0.1291148	7.745044
4	PHSG_t	0.7945804	1.258526
5	log(NHB_t)	0.1366295	7.319064
6	center_PPCC_PPA	0.6964122	1.435931
7	center_PHSG_TPI	0.8057760	1.241040
8	center_NAP_TPI	0.5884492	1.699382

Figure 11: VIF values after correcting multicollinearity


```
> backward_alt_r2_pred
[1] 0.8774784
>
```

Figure 12: R^2_{pred} of Alternate Backward Model

As seen in Figure 11, the model was able to correct multicollinearity, but it lost significant predictive power as indicated by lower R^2_{pred} in Figure 12. Consequently, we choose to not use this model despite these improvements in multicollinearity. We choose to continue with the original model chosen using backward selection. Although this model has high values of VIF for some predictor variables, we choose to keep the variables due to the hierarchy principle. In section 2.5, we found that the interaction terms were significant for predicting $\log(\text{TSC})$. By hierarchy principle, we choose to include the main effect variables because the interaction terms are significant as stated here [YJL07]

4 Conclusion

In this article we have proposed that using the data collected in SMSA's we can create a model to predict the total number of serious crimes that will occur in a geographic region. We developed this model by transforming the data to minimize variance as well as constituting interaction terms in order to best fit the model. After developing the model we validated our model by testing it's predictive power using both PRESS and a holdout data set. After validating the model we can confidently state that the model has a strong predictive power with regards to TSC in a SMSA. Our model shows that population size has the strongest correlation with crime, a fact that is supported by the paper in Appendix [Nol04]

The main limitation we faced in the study was our limitations with respect to data. When analyzing the data we found a significant multicollinearity problem which we were unable to solve without significantly weakening the predictive power. Another possible solution to the multicollinearity problem has always been to gather more data, hence if we were able to gain access to a larger data set with additional observations we may have been able to minimize the multicollinearity problem without massively dropping the predictive power of the model. Gathering additional observations from the data set would have helped us to create additional interaction terms that may have decreased the VIF.

If possible in future analysis it may be worth exploring how principal component analysis could help minimize the multicollinearity problem and the variance of the estimates. Another possible future avenue to explore is to see how a penalized regression model such as lasso and ridge regression may help us to develop another model with less of a multicollinearity problem. Finally the last possible avenue to improve the model would be by using data from different temporal periods to see how the model's predictive power holds over different time periods.

References

- [Nol04] James J. Nolan. “Establishing the statistical relationship between population size and UCR crime rate: Its impact and implications”. In: *Journal of Criminal Justice* Volume 32 (Issue 6 2004), pp. 547–555. DOI: <https://doi.org/10.1016/j.jcrimjus.2004.08.002>. URL: <https://www.sciencedirect.com/science/article/pii/S0047235204000832>.
- [YJL07] Ming Yuan, V. Roshan Joseph, and Yi Lin. “An Efficient Variable Selection Approach for Analyzing Designed Experiments”. In: *Technometrics* 49 (Nov. 2007), pp. 430–439. DOI: 10.1198/004017007000000173.

A DUPLEX Algorithm

```
# install.packages("reshape2") ---- ALREADY INSTALLED
library(reshape2)
data = read.table("C:\\Users\\adhig\\OneDrive\\Desktop\\University Courses
\\STAT 378\\Term_Paper_-_DATA.txt", header = TRUE) # nolint
attach(data)
options(max.print = .Machine$integer.max)

colnames(data) <- cbind('ID',
                        'LA',
                        'TP',
                        'PPCC',
                        'PPA',
                        'NAP',
                        'NHB',
                        'PHSG',
                        'CLF',
                        'TPI',
                        'TSC',
                        'GR')

# Creating three indicator variables to represent GR variable consisting of 4 categories
# We choose 4==W as our reference category for Geographical Region
NE <- ifelse(data$GR == 1, 1, 0)
NC <- ifelse(data$GR == 2, 1, 0)
S <- ifelse(data$GR == 3, 1, 0)
data <- cbind(data, NE, NC, S)

# Correlation matrix
cor(data)

# DUPLEX Algorithm
s1 = (length(data$S) - 1) * var(data$S)
s2 = (length(data$LA) - 1) * var(data$LA)
s3 = (length(data$TP) - 1) * var(data$TP)
s4 = (length(data$PPCC) - 1) * var(data$PPCC)
s5 = (length(data$PPA) - 1) * var(data$PPA)
s6 = (length(data$NAP) - 1) * var(data$NAP)
s7 = (length(data$NHB) - 1) * var(data$NHB)
s8 = (length(data$PHSG) - 1) * var(data$PHSG)
s9 = (length(data$CLF) - 1) * var(data$CLF)
s10 = (length(data$TPI) - 1) * var(data$TPI)
s11 = (length(data$TSC) - 1) * var(data$TSC)
s12 = (length(data$NE) - 1) * var(data$NE)
s13 = (length(data$NC) - 1) * var(data$NC)

Z = cbind((data$S-mean(data$S))/sqrt(s1), (data$LA-mean(data$LA))/sqrt(s2),
          (data$TP-mean(data$TP))/sqrt(s3), (data$PPCC-mean(data$PPCC))/sqrt(s4),
          (data$PPA-mean(data$PPA))/sqrt(s5), (data$NAP-mean(data$NAP))/sqrt(s6),
          (data$NHB-mean(data$NHB))/sqrt(s7), (data$PHSG-mean(data$PHSG))/sqrt(s8),
          (data$CLF-mean(data$CLF))/sqrt(s9), (data$TPI-mean(data$TPI))/sqrt(s10),
          (data$TSC-mean(data$TSC))/sqrt(s11), (data$NE-mean(data$NE))/sqrt(s12),
          (data$NC-mean(data$NC))/sqrt(s13))
T=chol(t(Z)%*%Z)
W=Z%*%solve(T)
distances = dist(W)
```

```

# Creating a dataframe with all distances and observation pairs using reshape2 package
distances_df <- melt(as.matrix(distances), varnames = c("row_num", "col_num"))

# Removing all observations whose distance is 0
distances_df <- subset(distances_df, value != 0)

# Creating two lists to contain the observation number for prediction and estimation points
estimation_index <- c()
prediction_index <- c()
temp_df <- distances_df

# Assign 1 pair of observations into each data set
for (i in 1:2){
  # max_index is the index for the observations with max distance
  max_index = which.max(temp_df$value)

  # obs1 and obs2 are the observations with max_distance
  obs1 = temp_df[max_index, "row_num"] # row_num
  obs2 = temp_df[max_index, "col_num"] # col_num

  if ((i %% 2) == 1){
    # if i is odd, add observation to estimation data
    estimation_index <- append(estimation_index, obs1)
    estimation_index <- append(estimation_index, obs2)
  } else {
    # if i is even, add observation to prediction data
    prediction_index <- append(prediction_index, obs1)
    prediction_index <- append(prediction_index, obs2)
  }
  # Update the dataframe by removing all observations with obs1 or obs2
  temp_df <- subset(temp_df, row_num != obs1 & row_num != obs2)
  temp_df <- subset(temp_df, col_num != obs1 & col_num != obs2)
}
rm(temp_df)

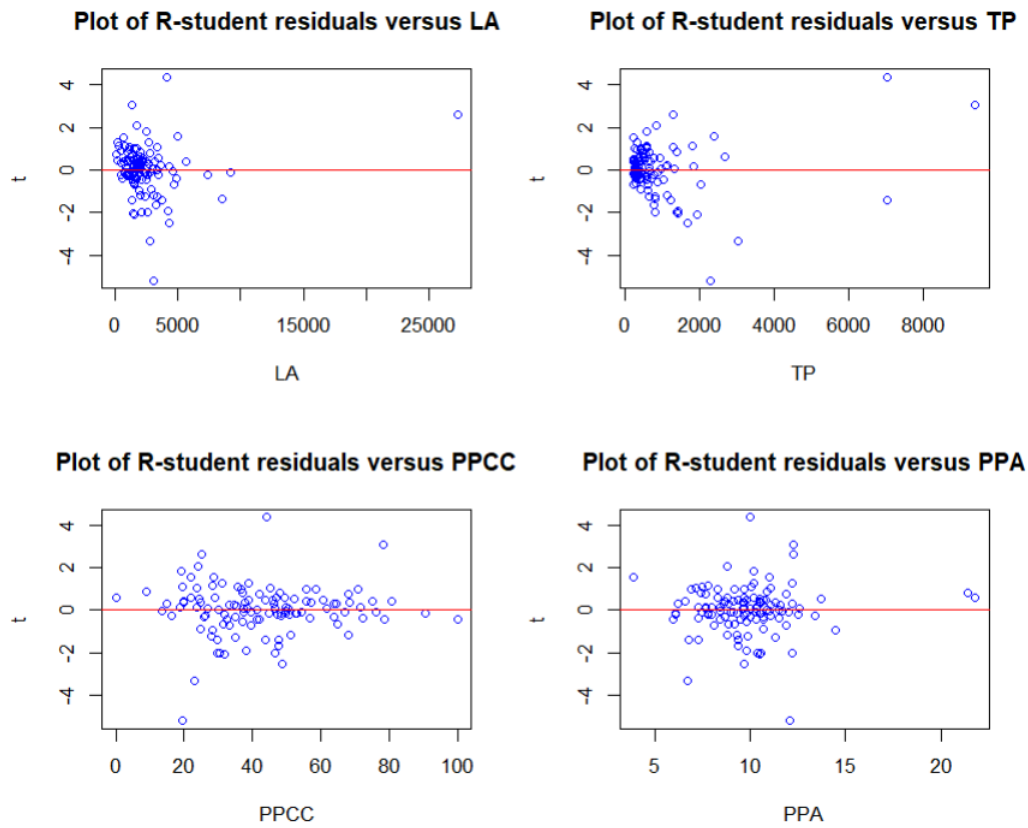
# Add 23 more observations to each data set
for (i in 1:(2*23)){
  if ((i %% 2) == 1){
    temp_df <- subset(distances_df, row_num %in% estimation_index)
    temp_df <- subset(temp_df, !(col_num %in% estimation_index))
    temp_df <- subset(temp_df, !(col_num %in% prediction_index))
    max_index = which.max(temp_df$value)
    obs = temp_df[max_index, "col_num"]
    estimation_index <- append(estimation_index, obs)
  } else {
    temp_df <- subset(distances_df, row_num %in% prediction_index)
    temp_df <- subset(temp_df, !(col_num %in% prediction_index))
    temp_df <- subset(temp_df, !(col_num %in% estimation_index))
    max_index = which.max(temp_df$value)
    obs = temp_df[max_index, "col_num"]
    prediction_index <- append(prediction_index, obs)
  }
  rm(temp_df)
}

# Now we create the data sets for training group and holdout group
holdout_data <- data[prediction_index,]

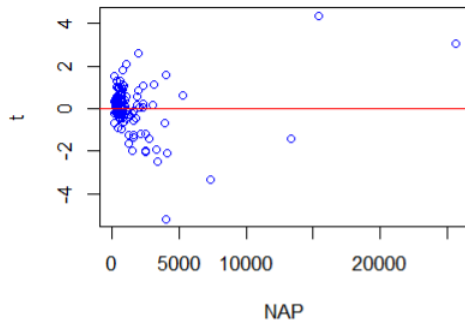
```

```
# All remaining observations will be used for training data
training_data <- data[-prediction_index,]
```

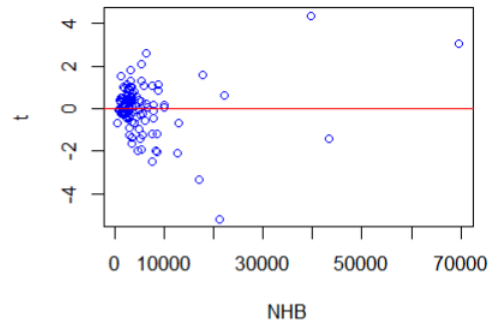
B Original Variable Plots



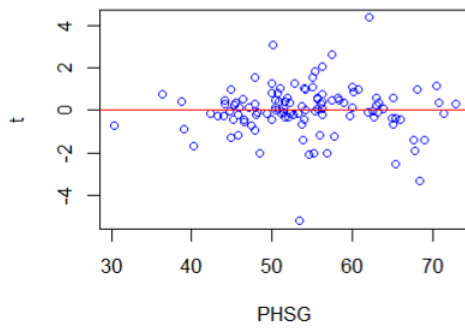
Plot of R-student residuals versus NAP



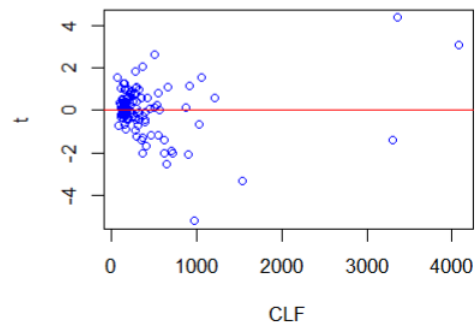
Plot of R-student residuals versus NHB



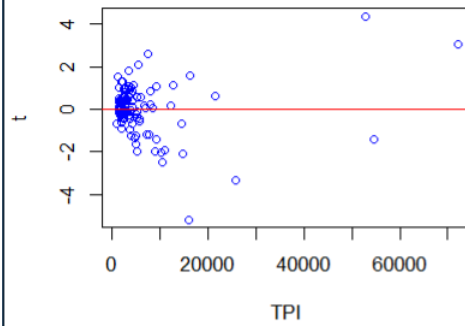
Plot of R-student residuals versus PHSG



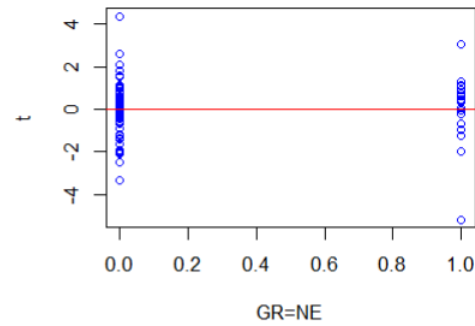
Plot of R-student residuals versus CLF



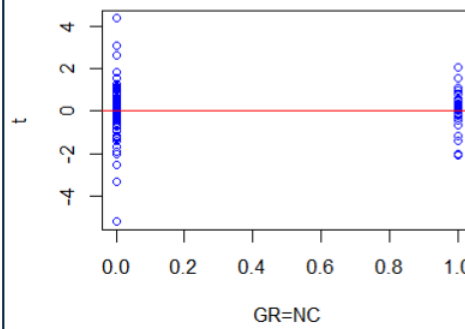
Plot of R-student residuals versus TPI



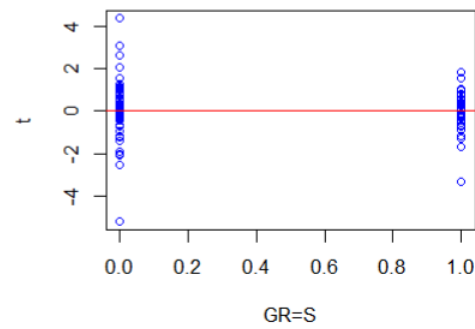
Plot of R-student residuals versus GR=NE



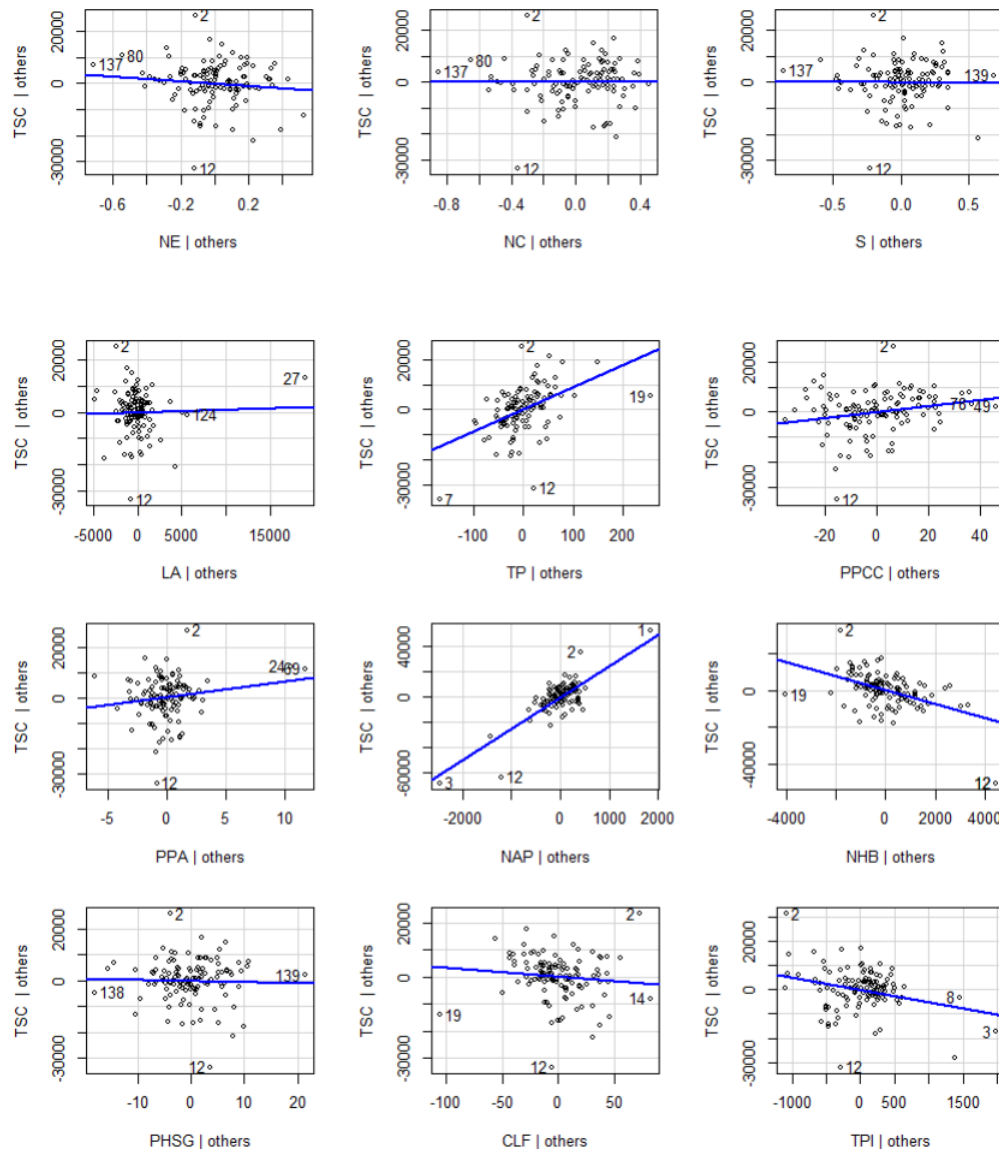
Plot of R-student residuals versus GR=NC



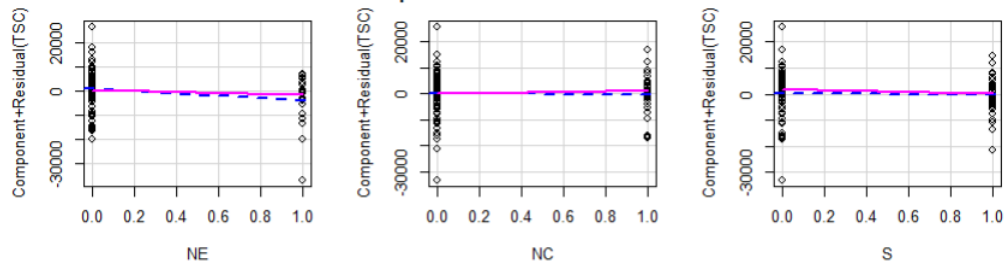
Plot of R-student residuals versus GR=S

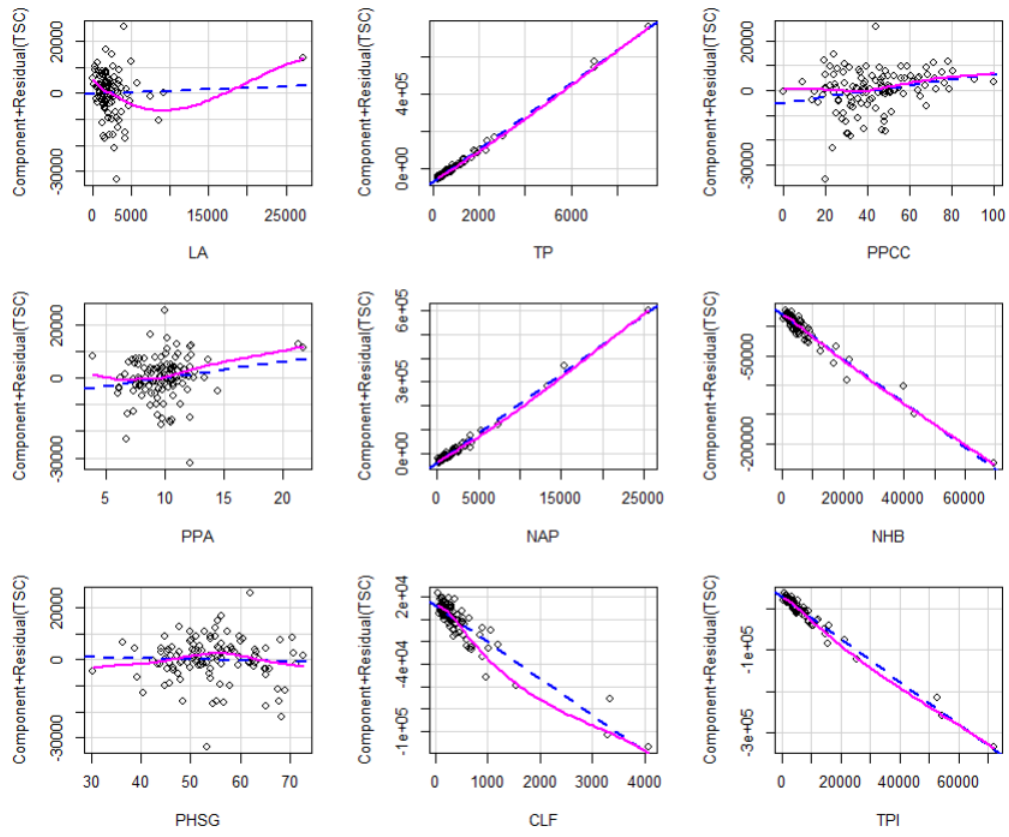


Added-Variable Plots



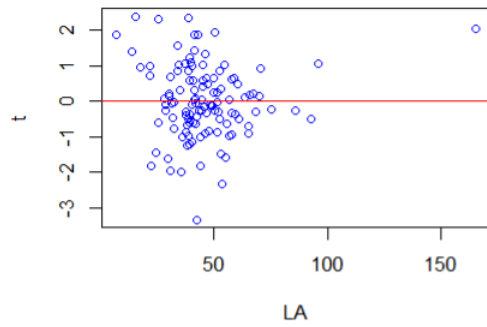
Component + Residual Plots



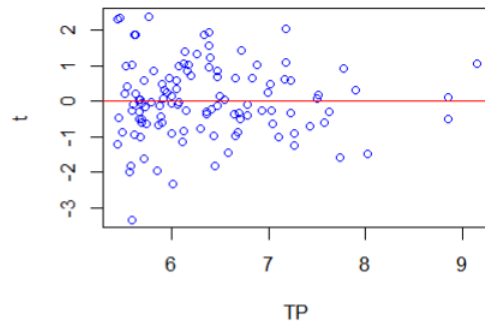


C Transformed Plots

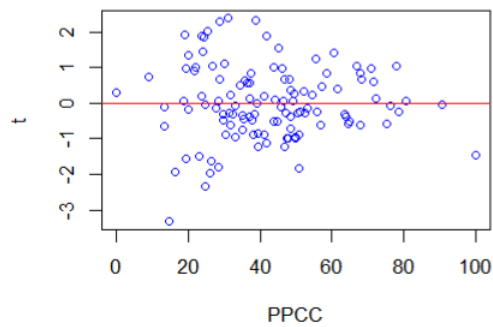
Plot of R-student residuals versus sqrt(LA)



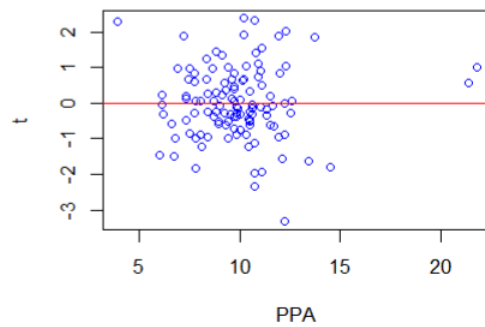
Plot of R-student residuals versus log(TP)



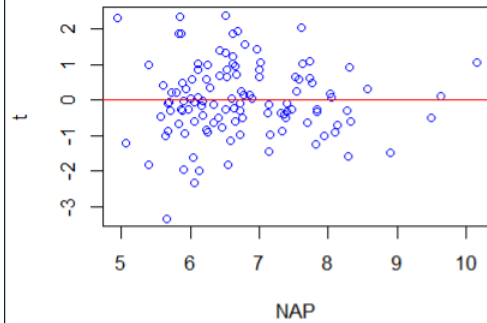
Plot of R-student residuals versus PPCC



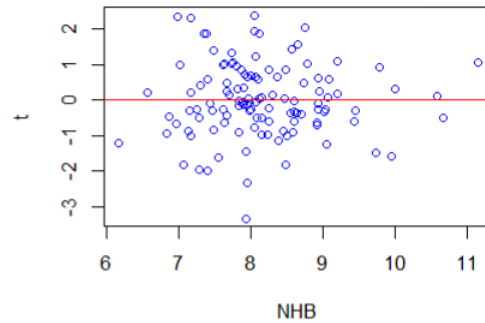
Plot of R-student residuals versus PPA



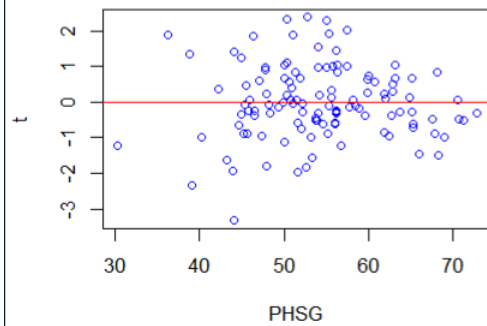
Plot of R-student residuals versus log(NAP)



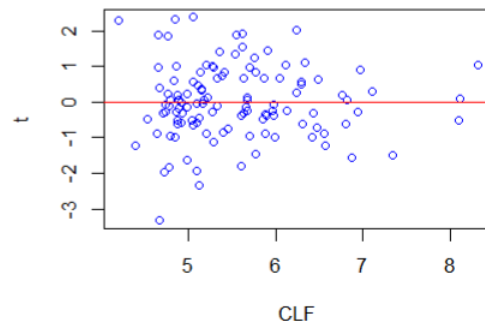
Plot of R-student residuals versus log(NHB)



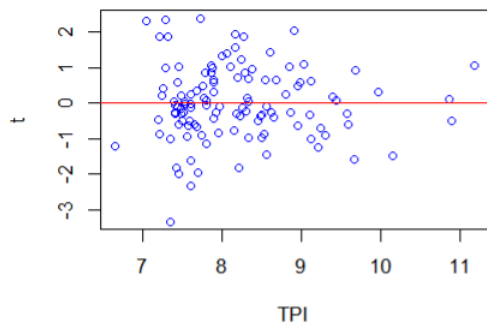
Plot of R-student residuals versus PHSG



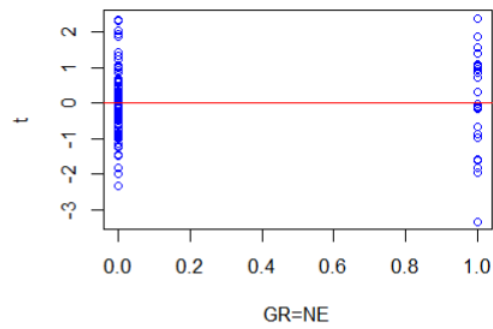
Plot of R-student residuals versus log(CLF)



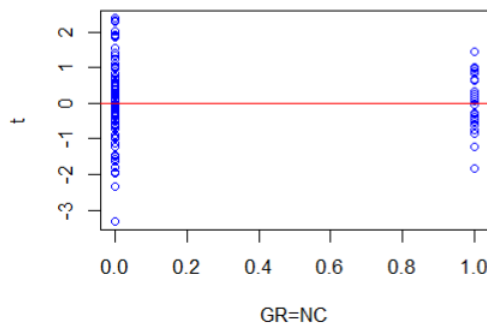
Plot of R-student residuals versus log(TPI)



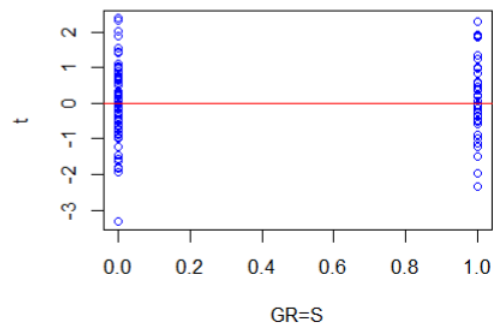
Plot of R-student residuals versus GR=NE



Plot of R-student residuals versus GR=NC



Plot of R-student residuals versus GR=S



Added-Variable Plots

