CRIME CHARACTERISTICS OF THE USA FOR 1995

STATISTICS FOR BUSINESS ANALYTICS 1

ATHINA SPANOU | AM: p2821924

Dataset: crimes_40.dat

Response: robbbPerPop

MSc IN BUSINESS ANALYTICS | PART-TIME 2019-2020

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Table of Contents

1. Introduction

The objective of this report is to identify the most significant variables in order to predict, through a regression model, the number of robberies per 100.000 habitants in the USA. The dataset that is used for this report's purpose is a Communities and Crime Un-normalized dataset. It consists of 100 crime instances that had been reported from different states across the country and 147 total attributes describing a crime. Specifically, the dataset contains 4 non-predictive attributes, 125 predictive attributes and 18 crime related attributes which are potential dependent variables.

According to the Federal Bureau of Investigation (FBI) a violent crime is defined as an offense which involves force or threat. The FBI's Uniform Crime Reporting (UCR) program categorizes these offenses into four categories: murder, forcible rape, robbery and aggravated assault. In this paper, the focus is on the violent crime of robberies. The FBI's UCR defines robbery as the taking or attempting to take anything of value from the care, custody, or control of a person or people by force or threat of force or violence and/or by putting the victim in fear.

The first hardship that increased the difficulty of the analysis was that the data was over-parameterized. Thus, the first goal was to reduce the number of attributes that were used as input in the multiple regression model. Moreover, all variables used in the model needed to be converted to numeric. Last but not least, the phenomenon of multi-collinearity in which the predictor variables of the multiple regression are highly correlated hindered the attribute selection process even more.

All in all, the question that has to be answered is the following: "**Are there any characteristics that can predict the number of robberies per 100.000 habitants of a typical area?** "

2. Data Cleaning and Data Transformation

After reading the dataset, the first inconsistency that was spotted and corrected was the existence in many variables of question marks ("?") which were transformed to NA. Additionally, 24 variables were deleted as more than 80% of their value were missing. Before, continuing with the process of data cleaning, some primary explanatory analysis was done in order to better understand the data.
The map in Figure 1 shows how robberies per 100k as distributed throughout the different states of the US for 1995. The states depicted in grey were missing from the dataset. What is more, the state of Alabama seems to have the highest value in robberies per 100k population.

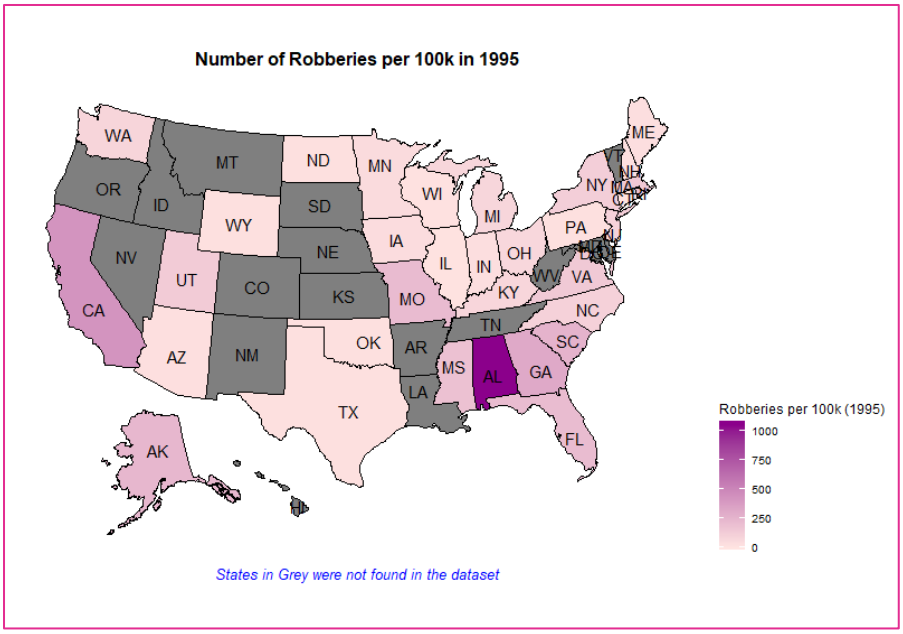


Figure 1 Map of Robberies per 100k

Another interesting chart is the one showing the proportion of robberies per 100k as part of the total Violent Crimes per 100k population. It is clearly seen that assaults are the most common violent crime in 1995 followed by robberies, rapes and murders.

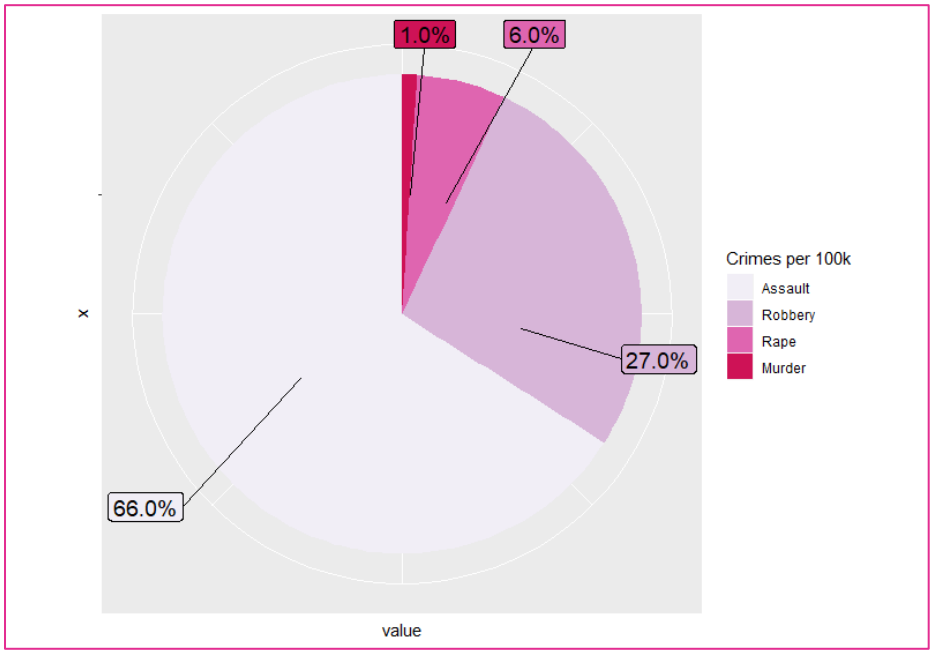


Figure 2 Pie Chart of Violent Crimes per 100k

The 4 non-predictive variables, communityname, countyCode, communityCode, fold, as well as the variable state were also deleted since they could possibly have a negative impact on the implementation of the regression analysis. What is more, since the aim is to predict the robberies per 100K, the rest of the potential dependent variables (murders, murdPerPop, rapes, rapesPerPop, robberies, assaults, assaultPerPop,

burglaries, burglPerPop, larcenies, larcPerPop, autoTheft, autoTheftPerPop, arsons, arsonsPerPop, ViolentCrimesPerPop, nonViolPerPop) have been deleted from the dataset. After these actions, the dataset contains 104 variables and no missing values. Finally, as part of the data transformation process, variables that were categorized as factors or integers were transformed to numeric variables in order to be processed later in the regression model.

Having completed the process of data cleaning, the next step was to detect any outliers in order to proceed with the analysis of the data.

The below code in Figure 3 was used for the outliers' detection:

```
#### OUTLIERS ####

outliers<-dataset[2:104]
for (i in 2:104){
  datatest<-dataset[,i]
  out <- boxplot( datatest, plot=FALSE )$out
  if(length(out)!=0){
    print('-------------------------------------------------------')
    print( paste('Outliers for variable', names(dataset)[i] ) )
    print( paste(length(out), 'outliers') )
    print( paste(round(100*length(out)/sum(!is.na(dataset)),1),
                 '% outliers', sep='' ) )
    print(which( datatest %in% out ))
  }
}
```

```
"Outliers for variable robbbPerPop"
"7 outliers"
"0.1% outliers"
13 36 42 71 78 96 98
```

Figure 3 Outliers Detection

Most of the variables in the dataset had outliers, while the variable of interest, robberies per 100k population, had 7 outliers out of 100 observations. All variables did not have more than 0.1% of outlying variables. Thus, as at this point the impact of the outliers in the future model is unknown, and no action will be taken in order to deal with the outliers.

3. Explanatory Data Analysis

In order to better apprehend the different variables of the dataset, numeric variables have been assigned to different nine groups. Two different kind of plots have been used for the visualization of the numeric variables, an histogram along with the density line as well as a qq-plot. A set of examples of the mentioned plots is represented below.

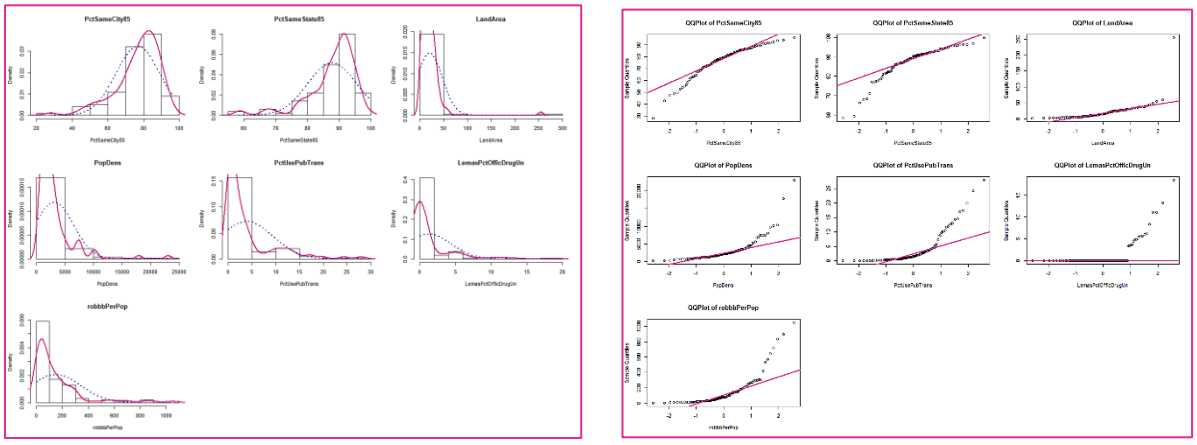


Figure 4 Histograms and QQplots of Numeric Variables

Comment: It is clearly spotted that variables PctSameCity85 and PctSameState85 have a left skewed distribution while the rest of the variables, including robberies per 100k have a right skewed distribution.

What is interesting from the QQplots is the fact that none of the above numeric variables are presented by a QQplot that could indicate normally distributed data. However, the QQplot of all variables need to be investigated before one can make an assumption of a non-normally regression model. At this part, the reader is encouraged to find the rest of histograms and qqplots in the section A.1 of the appendix.

Focusing on the dependent variable of interest, robberies per 100k population, one can conclude that the average number of robberies per 100k population is 148, while the maximum number that is reported in the dataset is about 1050 robberies per 100k. The purpose of the summary statistics presented in Figure 5 is to get a detailed view of the dependent variable.

```
Descriptive Statistics
numerics$robbbPerPop
N: 100

                    Mean   Std.Dev    Min      Q1   Median       Q3       Max     MAD      IQR
----------------- -------- --------- ------ ------- -------- -------- --------- ------- --------
      robbbPerPop   148.08    195.03   0.00   26.87    75.83   186.55   1049.77   91.25   158.44

Table: Table continues below


                     CV   Skewness   SE.Skewness   Kurtosis   N.Valid   Pct.Valid
----------------- ------ ---------- ------------- ---------- --------- -----------
      robbbPerPop   1.32       2.52          0.24       6.90    100.00      100.00
```

Figure 5 Summary Statistics of the Dependent Variable

A normality test was also implemented in order to justify if any of the variables in the dataset are normally distributed., For 16 out of the 103 numeric variables the null hypothesis of normally distributed data was not rejected after implementing the Kolmogornov Smirnoff Normality Test since the p-value was greater than the significance level (0.05). The depedent variable, robberies per 100k was not included in the variables that passed the normality tests.

4. Pairwise Comparisons

The next question asked is which variables are associated to the target, meaning the number of robberies per 100k. In order to reduce the number of variables for the comparisons, only the variables with correlation to

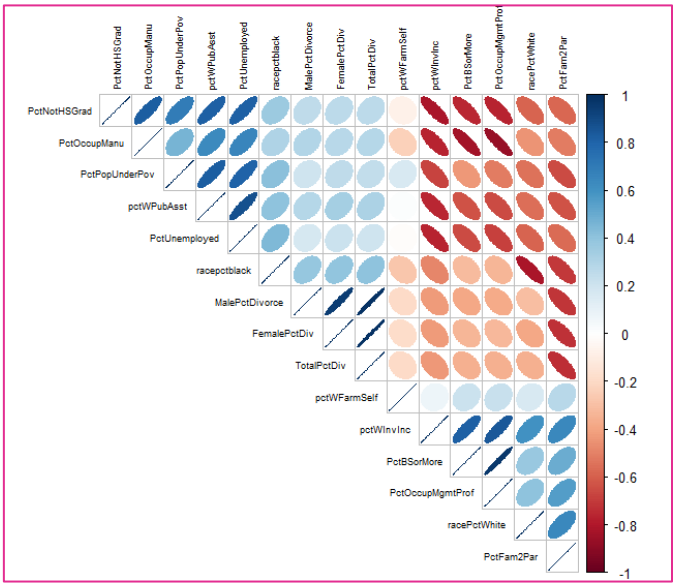


Figure 6 1st group of Numeric Variables (1-15)

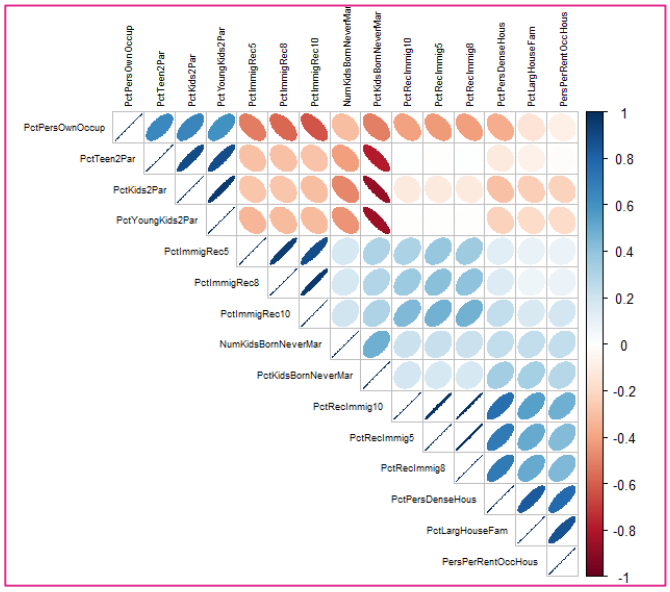


Figure 7 2nd group of Numeric Variables (16-30)

robberies per 100k population greater than 0.25 or less than -0.25 were kept in a different dataset. The new dataset consists of 37 variables which were then divided to 4 groups in order to implement the appropriate correlation plots between them. Figures 6, 7 and 8 represent the correlations between the different variables. High positive correlation is depicted the dark blue color while high negative correlation with dark red color. Low correlation is represented with lightly colored cells while white cells mean no correlation at all.

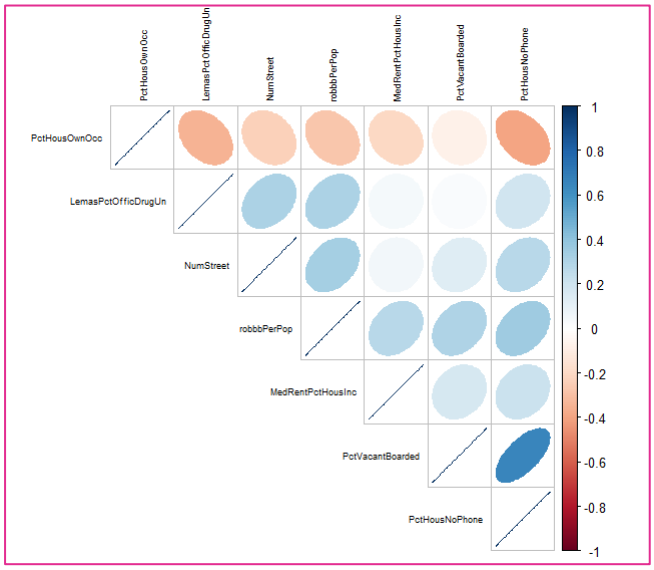


Figure 8 3rd group of Numeric Variables (31-37)

According to the correlation plots, there are some pairs of variables that can be considered as highly significant. For example, the variables that represent the percentage of population that is african American (racepctblack) as well as the percentage of population that is Caucasian (racePctWhite), as seen in Figure 6, are negatively correlated with a coefficient equal to -0.81. This means that both variables, racepctblack and racePctWhite have the same same effect to the response variable robberies per 100k. As a matter of fact, all variables that are characterized with a correlation coefficient above 0.7 (or 70%) are treated as highly correlated. In this kind of situation, it is common practice to keep one of the two highly correlated variables in order to avoid any multi-collinearity effect. Multi-collinearity refers to the case in which the predictor variables of the multiple regression are highly correlated. However, there will not be any variable selection at this point of the report.

5. Attribute Selection – Lasso

The dataset after the cleaning and the appropriate transformations contains 100 observations and 37 variables. For further attribute selection, Lasso is implemented through the Lars and Glmnet Lasso.

Lars Lasso

Given a set of input measurements x1, x2, .., xp and an outcome measurement robberies per 100k, the lasso fits a linear model: yhat=b0 + b1*x1+ b2*x2 + ... bp*xp. The The aim of Lars Lasso is to minimize sum((y-yhat)^2) subject to the parameter "s" which is calculates as sum[absolute value(bj)]. When "s" is large enough, the constraint has no effect and the solution is just the usual multiple linear least squares regression of on x1, x2,…,xp. However, for smaller values of s (s>=0) the solutions are shrunken versions of the least squares estimates. Regularly, some of the coefficients bj are zero. The "s" which minimizes the cp statistic is used in order to define the number of predictors to use in a regression model. Each colored line in figure 9 represents the value of a different coefficient in the model.

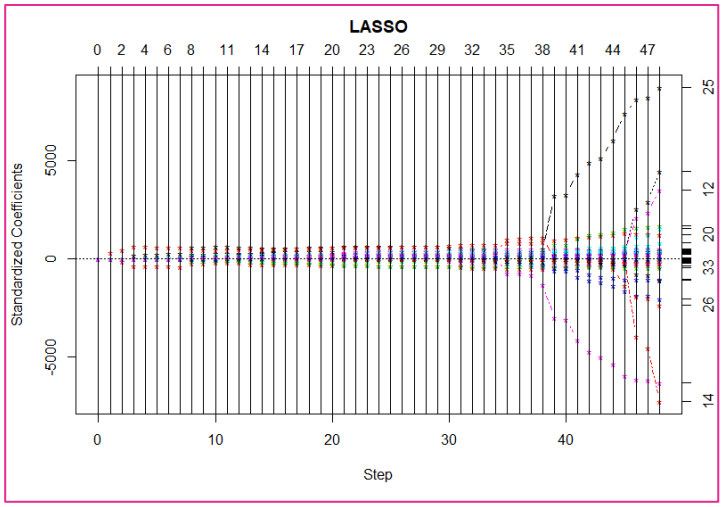


Figure 9 Lars Lasso Coefficients

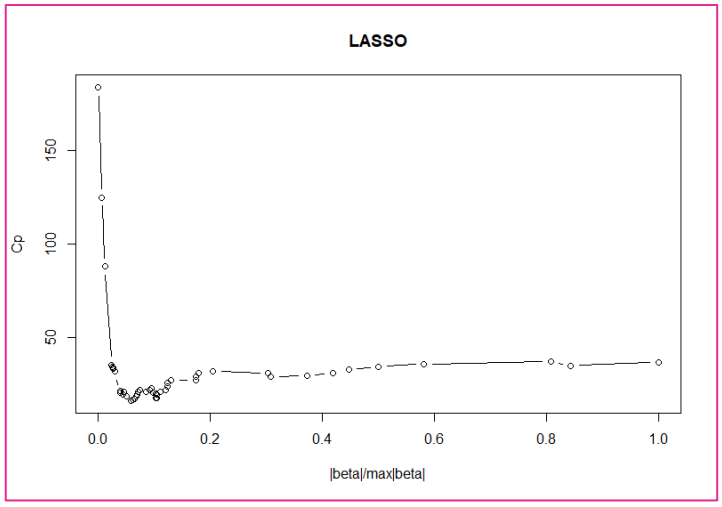


Figure 10 S of the optimal CP

Based on Lars Lasso, the attributes that should be excluded from the model are those followed by a zero coefficient, as presented in Figure 11.

```
> coef( lasso1, s=mincv.s, mode='fraction' )
       racepctblack        racePctWhite        pctWFarmSelf          pctWInvInc          pctWPubAsst       PctPopUnderPov       PctNotHSGrad
          0.9379258          -1.7371098           0.0000000           0.0000000           0.0000000           0.0000000           0.0000000
        PctBSorMore       PctUnemployed        PctOccupManu   PctOccupMgmtProf       MalePctDivorce       FemalePctDiv        TotalPctDiv
          0.0000000           0.0000000           0.0000000           0.0000000           0.0000000           0.0000000           0.0000000
         PctFam2Par         PctKids2Par     PctYoungKids2Par       PctTeen2Par  NumKidsBornNeverMar PctKidsBornNeverMar       PctImmigRec5
          0.0000000           0.0000000           0.0000000           0.0000000           0.0000000          19.9886636           0.0000000
       PctImmigRec8       PctImmigRec10        PctRecImmig5        PctRecImmig8       PctRecImmig10       PctLargHouseFam PersPerRentOccHous
          0.0000000           0.0000000           0.0000000           0.0000000           0.0000000           0.0000000           0.0000000
    PctPersOwnOccup     PctPersDenseHous        PctHousOwnOcc     PctVacantBoarded       PctHousNoPhone       MedRentPctHousInc          NumStreet
          0.0000000           0.0000000           0.0000000           0.0000000           0.0000000           0.0000000           0.0000000
LemasPctOfficDrugUn
          0.0000000
```

Figure 11 Lars Lasso Attribute Selection

Glmnet Lasso

Each colored line in Figure 12 depicts the value taken by a different coefficient in the model. Lambda, as seen in figure 13, is the regularization term (L1 norm), so as lambda approaches zero, the loss function of the model approaches the Ordinary Least Squares (OLS) loss functions. OLS basically chooses the parameters

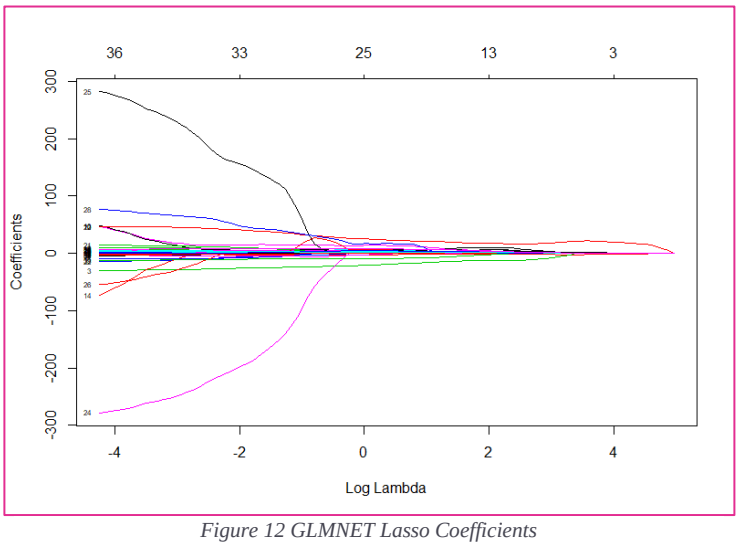


Figure 12 GLMNET Lasso Coefficients

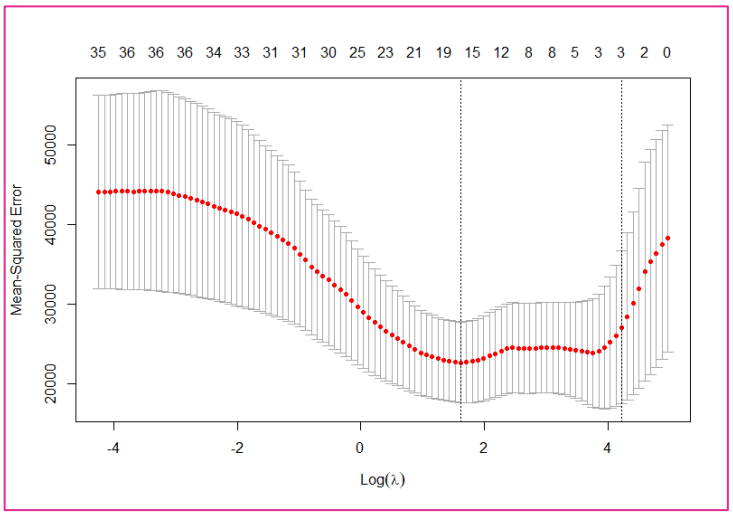


Figure 13 GLMENT Lasso Lambda Min

of a linear function out of a set of explanatory variables by minimizing the sum of squares of the differences between the observed dependent variable in the given dataset and those predicted by the linear function. Therefore, when lambda is very small, the LASSO solution should be very close to the OLS solution, and all of the coefficients are in the model. As lambda grows, the regularization term has greater effect and thus fewer variables will appear in the model as more and more coefficients will be zero valued. As a result, when L1, the regularization term for LASSO, is small then the regularization is high. Therefore, an L1 norm of zero gives an empty model, and as L1 norm increase, variables will "enter " the model as their coefficients take non-zero values. Lambda.min is the value that gives the minimum cross-validated error but it is way too complex and over fitted. On the other hand, lambda.1se returns the most regularized model such that error is within one standard error of the minimum. Hence, the output of the lambda.1se was taken under consideration during the final attribute selection.

Given the output of Lars as well as the Glmnet Lasso the following attributes are considered to be the most important in predicting the number of robberies per 100k population:

- racepctblack: the percentage of population that is african american
- racePctWhite: the percentage of population that is caucasian
- PctKidsBornNeverMar: the percentage of kids born to never married

The correlations between the selected attributes are represented in Figure 14. High positive correlation, on one hand, is depicted with dark blue color while high negative correlation with dark red color. Low correlation, on the other hand, is represented with lightly colored cells. The existing collinearity issues will be further examined in the report.

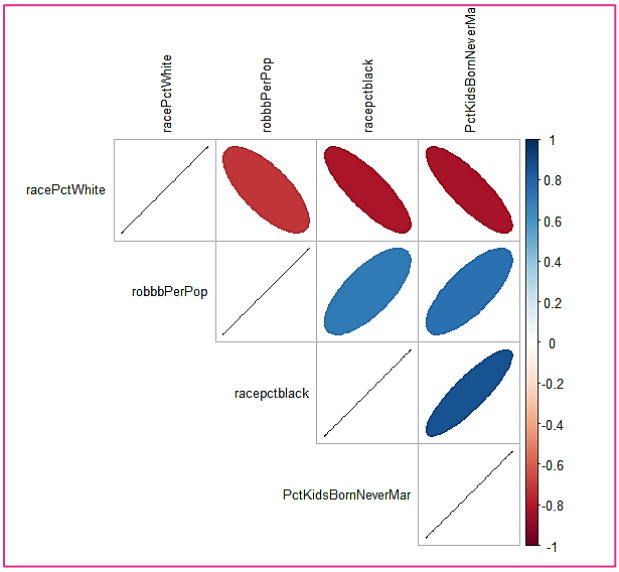


Figure 14 Collinearity between Lasso selected attributes

6. The Regression Model

All the selected variables will be used initially in order to fit a full linear model, as seen on figure 15.

```
Call:
lm(formula = robbbPerPop ~ ., data = regressionmodel)

Coefficients:
        (Intercept)         racepctblack         racePctWhite  PctKidsBornNeverMar
            312.736                2.259               -3.084               25.823
```

Figure 15 Regression Full Model

Formula of the Initial Model: lm(robbbPerPop~.,data=regressionmodel)

Adjusted R-squared: 0.5745 or 57%

Residual standard error: 127.2

Output of Code: Section B.1.1 of Appendix

Comment: According to the initial model when all variables are zero, the fixed number of robberies is 313 per 100k population. The adjusted R-squared shows the percent of the standard deviation which is described by the model. The initial model has a not so high adjusted R square but represents a regression that explains to some extent the variance in the response variable. The residual standard error in a measure of how well the model fits the data. Thus, the lower this value is, the more accurate the model prediction is. However, the standard error of the residuals, indicates that there is no close relationship among fitted values and observed values. What is more, in order to avoid any multicollinearity problems, variance inflation factors are applied, as seen on Figure 16. It is clearly seen that non of the single VIFs are above 10, so no variable nees to be removed at this point.

```
> round(vif(initial_model),2)
       racepctblack        racePctWhite PctKidsBornNeverMar
               4.47                3.62                4.88
>
```

Figure 16 VIF on variables of initial model

Linear Regression Assumptions and Model Improvement

The Initial Model

Having as a reference Figure 17, before any further improvement is done on the regression model the four assumptions of the linear model will be checked.

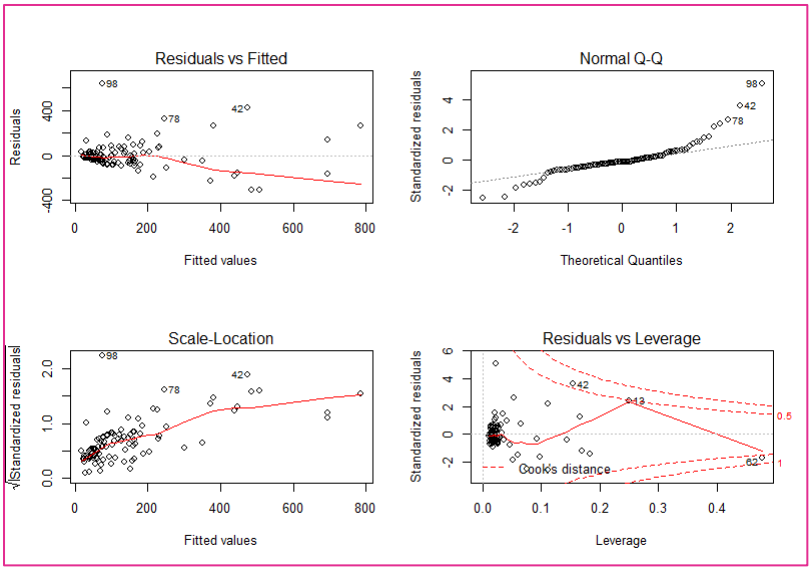


Figure 17 Plot of Initial Full Model

The four assumptions to be tested are the below:

1. **Linear Relationship** between the independent variables (X) and the dependent variable (Y). The non-linearity is mainly checked visually by plotting the residuals against the fitted values. What is more, scatterplots are created in order to project every dependent variable against the independent variable. In addition, the p-value of the Tukey test is used as an indication of the linearity of the model.
2. **Independence of Errors**. The independence is checked with tests like the Durbin-Watson test or by examining the pattern of autocorrelations. The Durbin-Watson's statstic desirable value is between 1.4 and 2.6 while the p-value should be greater than the significance level (0.05).
3. **Homoscedasticity of Errors**. The Homoscedasticity is checked by using the Levene's test, which tests the null hypothesis that the population variances are equal (homogeneity). If the p-value of Levene's test is less than the significance level (0.05), the null hypothesis of equal variances is rejected and it is concluded that there is a difference between the variances in the population.
4. **Normality of Errors** (and of Y). The normality of the residuals is assessed by tests like the Shapiro-Wilkoson, the Kolmogorov-Smirov and the Anderson-Darling tests as well as visually through a qq-plot. It is notes that this assumption is the generally the least important of the set as if it is not met, the beta estimates will still be unbiased, but the p-values will be inaccurate.

Figure 18 includes the symmary plots regarding the linear regression assumptions for the initial full model. The assumptions of Homoscedasticity and Normality were rejected for the initial model (Levene's $p<0.05$, KS $p<0.05$). The reader, is instructed to view the rest of the summary plots regarding the initial model at the section B.1.2 of the appendix.

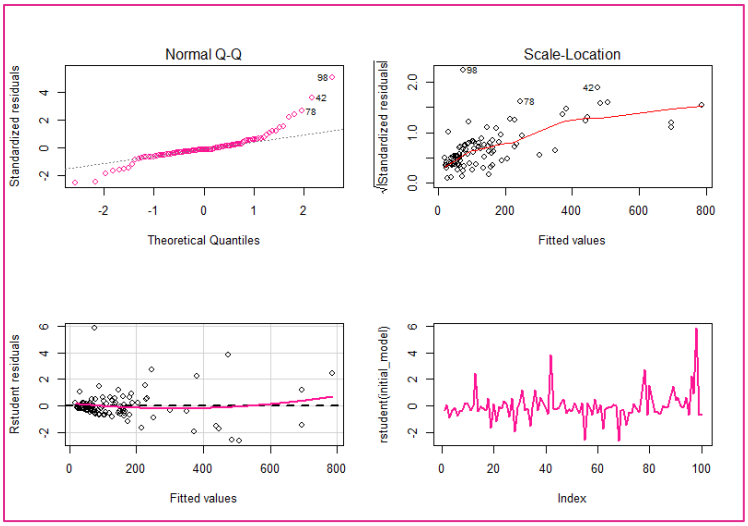


Figure 18 Summary Plots for the Initial Model

The Stepwise AIC Model

To fix the fit of the regression model stepwise regression with both ways is implemented as next step.

Formula of the AIC Model: lm(robbbPerPop~+racePctWhite +PctKidsBornNeverMar, data=regressionmodel)

Adjusted R-squared: 0.5723 or 57%

Residual standard error: 127.6

Assumptions met: Independence and Linearity

Assumptions not met: Homoscedasticity and Normality.

Output of Code: Section B.2.1 Appendix

Comment: The model after the stepwise AIC has not improved as neither the adjusted R-aquared has increased nor the standard error of the residuals has decreased. However, the method has noe kept two variable from the initial dataset which are racePctWhite, meaning the percentage of population that is Caucasian, and PctKidsBornNeverMar which represents the percentage of kids born to never married. What is more, after implementing the VIF function none of the predictors had a value greater than 10. Moreover, the assumptions of Homoscedasticity and Normality were rejected for the stepwise model (Levene's $p<0.05$, KS $p<0.05$). The reader is instructed to view the summary plots regarding the stepwise model at the section.B.2.2 of the appendix.

The Log Transformation

Since it has been noticed that the residuals are not normally distributed, it is suggested that logarithmic or polynomial transformations as used. First, the logarithmic transformation is applied to the model of the stepwise regression. When the logarithmic function is applied to the dependent variable the value of 1 is added since there is an observation which is valued as zero and the logarithmic function applies only to positive numbers.

Formula of the log Model: lm(log(robbbPerPop+1)~ +racePctWhite +log(PctKidsBornNeverMar), data=regressionmodel)

Adjusted R-squared: 0.448 or 44%

Residual standard error: 0.966

Assumptions met: Independence, Multi-Collinearity and Homoscedasticity, Normality, Linearity

Assumptions not met: None

Output of Code: Section B.3.1 of Appendix

Comment: The logarithmic transformation although it has not improved the adjusted R-squared of the model, it has les to a significant decrease of the standard error of the residuals from 127.6 to 0.99. What is more, after implementing the VIF function none of the predictors had a value greater than 10. Regarding the assumptions of the regression model, the model after the log transformation the assumption of linearity was not met while the model did not pass the Shapiro-Wilkoson(p-value<0.05) test for Normality unlike the Kolmogorov-Smirnov and the Anderson-Darling tests. The reader, is instructed to view the summary plots regarding the log model at the section B.3.2 of the appendix.

The Polynomial Transformation

Formula of the polynomial Model: lm(log(robbbPerPop+1) ~+poly(racePctWhite,3) +log(PctKidsBornNeverMar), data=regressionmodel)

Adjusted R-squared: 0.5292 or 53%

Residual standard error: 0.8924

Assumptions met: Linearity, Independence, Multi-Collinearity and Homoscedasticity, Normality

Assumptions not met: None

Output of Code: Section B.4.1 of Appendix

Comment: The log model has been enhanced as the cubic polynomial of predictor "racePctWhite" has been added as well as the logarithmic function of the predictor "PctKidsBornNeverMar". These transformation have significantly improved the fit of the model since the adjusted R-squared has increased to 53% while the residual standard error has decreased even more to 0.89. What is interesting, is the fact that this model has passed all the assumptions of a linear regression model. The reader, is instructed to view the summary plots regarding the log model at the section. B.4.2. of the appendix.

The Box-Cox Transformation

In linear regression, box-cox transformation is mainly used in order to transform the target variable so that linearity and normality assumptions can be satisfied. The Box-Cox transformation is implemented by using the polynomial model.

Formula of the box-cox Model: lm(powerTransform(log(robbbPerPop),lambda)~ +poly(racePctWhite,3) +log(PctKidsBornNeverMar), data=boxcox)

Adjusted R-squared: 0.5534 or 55%

Residual standard error: 1.318

Assumptions met: Linearity, Independence, Multi-Collinearity and Homoscedasticity, Normality

Assumptions not met: None.

Output of Code: Section B.5.1 of Appendix

Comment: The box-cox transformation has increased the adjusted R-squared of the model while the residual standard error has, unfortunately, increased too. These transformations have significantly improved the fit of the model since the adjusted R-squared has increased to 55% whereas the residual standard error has increased to 1.31. What is interesting is the fact that this model has passed all the assumptions of a linear regression model. The reader is instructed to view the summary plots regarding the log model at the section B.5.2 of the appendix.

Outliers Detection (Cook's Distance)

Influential points in the regression model can sometimes significantly distort the regression model. At this point, Cook's distance will be computed in order to find the points which are outside the interval [-2,2] of the residuals and remove them. Cook's distance is a measure computed based on a given regression model and therefore it is impacted only by the independent variables that are included in the model. Basically, Cook's distance, computes the influence exerted by each data point on the predicted outcome and the fitted values. Figure 19 shows those observations that have a cook's distance greater than 4 times the mean and therefore are classified as influential. According to Cook's distance there are 8 points which are marked as critical. Therefore, those points have been excluded from the dataset.

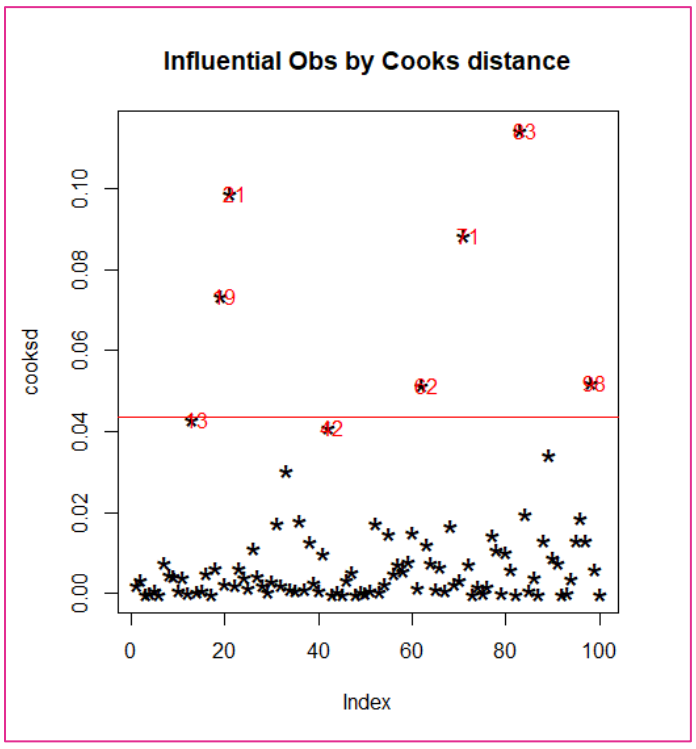


Figure 19 Influential Obs By Cook's Distance

Formula of the Model after Cook's distance: lm(powerTransform(log(robbbPerPop),lambda)~ +poly(racePctWhite,3)+log(PctKidsBornNeverMar),data=cooks)

Adjusted R-squared: 0.667 or 66%

Residual standard error: 1.092

Assumptions met: Linearity, Independence, Multi-Collinearity and Homoscedasticity, Normality

Assumptions not met: None.

Output of Code: Section B.6.1 of Appendix

Comment: The transformation after the implementation of the Cook's distance has increased the adjusted R-squared of the model while the residual standard error has, fortunately, decreased. These transformation have significantly improved the fit of the model since the adjusted R-squared has significantly increased to 66% whereas the residual standard error has decreased to 1.092. What is interesting, is the fact that this model has passed all the assumptions of a linear regression model. The reader, is instructed to view the summary plots regarding the log model at the section. B.6.2. of the appendix.

7. Cross Validation and out of Sample Predictive Ability of the model

After the data transformations, three models have been chosen in order to be cross-validated:

- The log model: lm(log(robbbPerPop)~+racePctWhite+log(PctKidsBornNeverMar),data=cooks)
- The polynomial model after cook's distance: lm(log(robbbPerPop+1)~+poly(racePctWhite,3)+log(PctKidsBornNeverMar),data=cooks)
- The Box-Cox model: lm(powerTransform(log(robbbPerPop),lambda)~+poly(racePctWhite,3)+log(PctKidsBornNeverMar),data=boxcox)

The method that is going to be used on the train dataset is the k-fold cross validation. Basically, the dataset is randomly split into k-Folds and for each k-fold, the model is built on k-1 folds of the dataset. Then the model is tested to check the effectiveness for the kth fold. The error is recorded for each of the predictions while the process is repeated until each of the k-folds has served as the test set. The average of the k recorded errors is called the cross-validation error and will serve as the performance metric for the model. A lower value of k is more biased while a higher value of k is less biased, but can suffer from large variability. For the current report, the 10-fold cross validation is used.

Cross Validation – LOG Model

RMSE: 0.894 | **Rsquared:** 0.637 | **MAE:** 0.762

Cross Validation – Polynomial Model

RMSE: 0.721 | **Rsquared:** 0.665 | **MAE:** 0.599

Cross Validation – Box-Cox Model

RMSE: 0.711 | **Rsquared:** 0.652 | **MAE:** 0.589

Comment: By comparing the three models, it is clearly seen that that the polynomial model is better in predictions than the other two model since it has yielded a high R-squared as well as a low Root Mean Square Error. It is obvious that the transformations were necessary in order to increase the predictive ability of the model. The reader is instructed to view the outputs of the 10-fold cross validation method at the section C.1.1 to C.1.3 of the appendix.

The next step is to evaluate the out of sample predictive ability of the models. To do so, a test dataset is used in order to evaluate the predictive ability of the already mentioned selected model in order to select the final model. The test dataset that has been used contains 100 observations about crimes for US in 1995. The performance of the 3 models on the test dataset is recorded below:

Cross Validation on Test Dataset – LOG Model

RMSE: 0.991 | **Rsquared:** 0.153 | **MAE:** 0.777

Cross Validation on Test Dataset – Polynomial Model

RMSE: 1.04 | **Rsquared:** 0.112 | **MAE:** 0.806

Cross Validation on Test Dataset – Box-Cox Model

RMSE: 1.39 | **Rsquared:** 0.085 | **MAE:** 1.12

Comment: The results of the predictive ability of the models on the test dataset are indeed very interesting. The log model seems to be the one with the best predictive ability yielding an R-squared of 25% and a low Root Mean Square Error of 1.02. The reader is instructed to view the outputs of the 10-fold cross validation method at the section.C.2.1-C.2.2 of the appendix.

8. Interpretation of the Final Model

The model that is chosen after taking into consideration the aforementioned analysis is the log model:

$$\log(\text{robbbPerPop}) = \beta o + \beta 1 * \text{racePctWhite} + \beta 2 * \log(\text{PctKidsBornNeverMar})$$

Figure 20 represents the final interpretation of the model which is charasterised with an adjusted R-ssquared of 45% and a Residual Standard Error of 0.97. What is more, Figure 21 depicts the main summary plots of the final model.

```
Call:
lm(formula = log(robbbPerPop + 1) ~ +racePctWhite + log(PctKidsBornNeverMar),
    data = regressionmodel)

Residuals:
   Min     1Q Median     3Q    Max
-3.833 -0.587  0.065  0.701  2.661

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)               6.65482    0.71875    9.26  5.3e-15 ***
racePctWhite             -0.03192    0.00768   -4.16  7.0e-05 ***
log(PctKidsBornNeverMar)  0.46196    0.13600    3.40  0.00099 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.966 on 97 degrees of freedom
Multiple R-squared:  0.459,     Adjusted R-squared:  0.448
F-statistic: 41.1 on 2 and 97 DF,  p-value: 1.15e-13
```

Figure 20 Final Model Interpretation

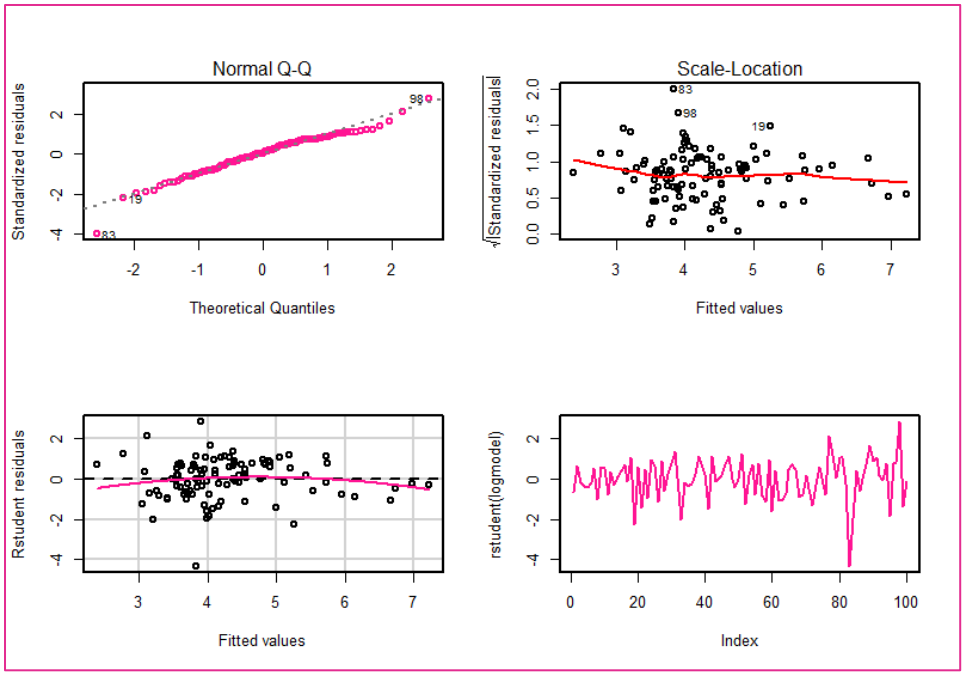


Figure 21 Final Model Summary Models

Model Interpretation

According to figure 20, the interpretion of the chosen model is the following:

Log(robberies_per_100k) = **6.65482** – **0.03192** x racePctWhite + **0.46196** x log(PctKidsBornNeverMar)

- The expected robberies per 100k populations when all covariates are zero is expected to be equal to exp(6.65482) = 777.
- For the score of the percentage of population that is Caucasian, one can say that for a one-unit increase, when all other covariates are zero, a decrease of about 30% is about to be noticed in the number of robberies per 100k since exp(0.03192)=1.03
- For any 1% increase in the percentage of kids born to never married, the expected increase in the number of robberies per 100k is about (1.01)^0.4619 = 1.00460 of about 4.6%.

9. Further Analysis

Characteristics of the Typical Profile of an Area

In order to calculate the overall characteristics of a typical area the covariates have been centered to the median, since the variables that have been centered are neither normal nor symmetric. Thus, the mean is not a descriptive measure of central location and the median is used.

```
Call:
lm(formula = log(robbbPerPop + 1) ~ +racePctWhite + PctKidsBornNeverMar,
    data = centered_model)

Residuals:
   Min     1Q Median     3Q    Max
-2.174 -0.706  0.116  0.715  1.524

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)          4.2535     0.0895   47.53   <2e-16 ***
racePctWhite        -0.0350     0.0109   -3.22   0.0018 **
PctKidsBornNeverMar  0.1288     0.0612    2.10   0.0382 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.854 on 88 degrees of freedom
Multiple R-squared:  0.498,     Adjusted R-squared:  0.486
F-statistic: 43.6 on 2 and 88 DF,  p-value: 6.98e-14
```

Figure 22 Typical Profile of an Area

Taking into consideration the findings in figure 22:

- If we compare two typical areas with the same characteristics which differ only by 1 percentage of population that is Caucasian, then the expected difference in the number of robberies per 100k will be 3% lower for the area with increased percentage of Caucasians.
- If we compare two typical areas with the same characteristics which differ only by 1 percentage of kids born to never married, then the expected difference in the number of robberies per 100k will be 13% in favor of the area with the increased percentage of kids born to never married parents.

Characteristics of the Worst and the Best Area

In order to characterize the worst and the best areas the median is set as a threshold for each one of the predictors. Thus, an area above the median number of robberies per 100k of each predictor, given that the rest variables remain un-changed, will be characterized as the worst area if there is a positive relationship between the dependent and the independent variable. More specifically, as the percentage of Caucasian on median number of robberies is 70.5, an area with Caucasians above that number is probable to get a lower rate of robberies, and so it is characterized as better area. Regarding the percentage of kids born to never married parents, as this percentage on median number of robberies is 3.43, an area with more kids born to never married above that number is probable to get a higher rate of robberies, and so it is characterized as worse area. To sum up, the worst area is the one with percentage of Caucasians less than 70 and with a percentage of kids born to never married parents above 3, while the best area is the one with percentage of Caucasians more than 70 and with a percentage of kids born to never married less than 3. Finally yet importantly, an overall estimate of the expected robberies per 100000 in total is 12322 as it has been computed by summarizing the fitted values of the model.

Exploration of Other Types of Regression

In order to explore other methods of analysis, the robust regression has also been tested on the current dataset. Robust regression is an alternative to least squares regression when data are contaminated with outliers or influential observations, and it can also be used for the purpose of detecting influential observations.

Formula of the Model: lmrob((log(robbbPerPop))~+(racePctWhite)+log(PctKidsBornNeverMar), data=cooks, method='MM', fast.s.large.n = Inf, cov = ".vcov.w")

Adjusted R-squared: 0.545or 55%

Residual standard error: 0.825

Output of Code: Section D of Appendix

Comment: The robust model has clearly a better predictive ability since the adjusted R-squared has significantly improved, compared to the final model that has been selected, while the residual standard error has also decreased.

10. Conclusions and Discussion

According to this paper's analysis the variables which foremost define the number of robberies per 100.000 habitats are the percentage of population that is Caucasian as well as the percentage of kids born to never married. The fact that the adjusted R-squared of the final model is not so high might suggest that a dataset with more observations should be used in order to further improve the predictive ability of the model.

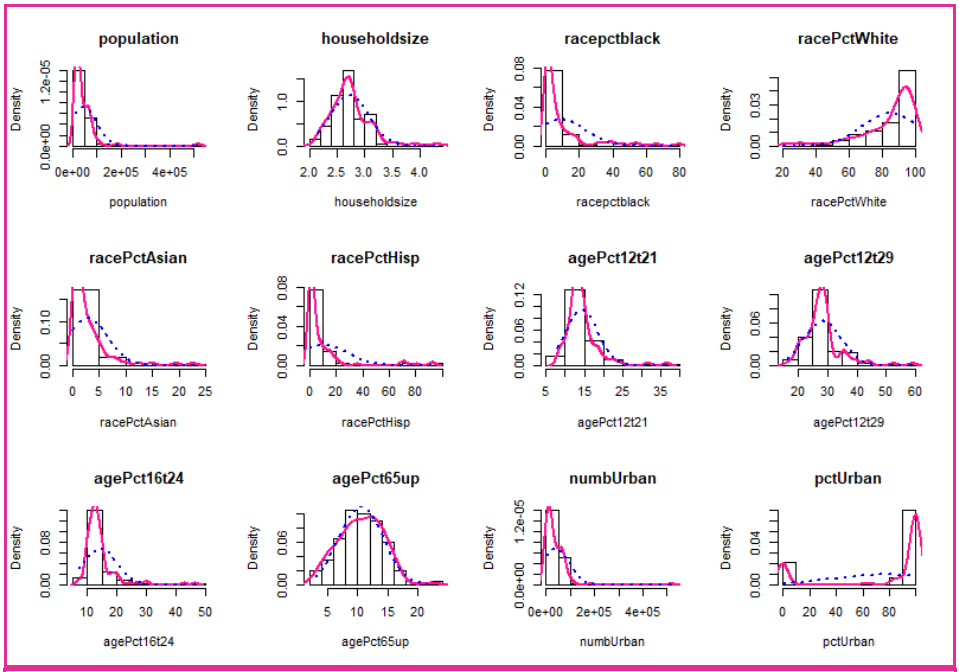
11. Citations and References

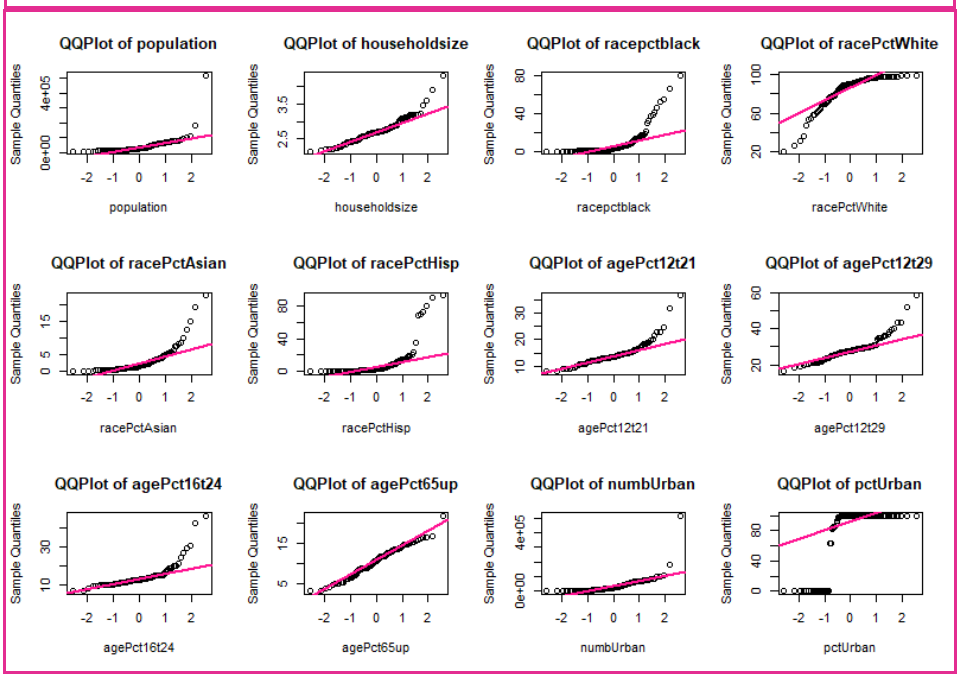
- Regression diagnostics: testing the assumptions of linear regression Fuqua School of Business - http://people.duke.edu/~rnau/testing.htm
- Introduction to SAS. UCLA: Statistical Consulting Group.
- From https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faqhow-do-i-interpret-a-regression-model-when-some-variables-are-log-transformed/ (accessed January 4, 2020).
- Outlier Treatment With R | Multivariate Outliers. (n.d.). Retrieved from http://r-statistics.co/Outlier-Treatment-With-R.html
- America's Health Rankings analysis of U.S. Department of Justice, Federal Bureau of Investigation, United Health Foundation, AmericasHealthRankings.org, Accessed 2020.
- U.S. Department Of Commerce, Bureau Of The Census Producer, Washington, DC and Inter-university Consortium for Political and Social Research Ann Arbor, Michigan. (1992)

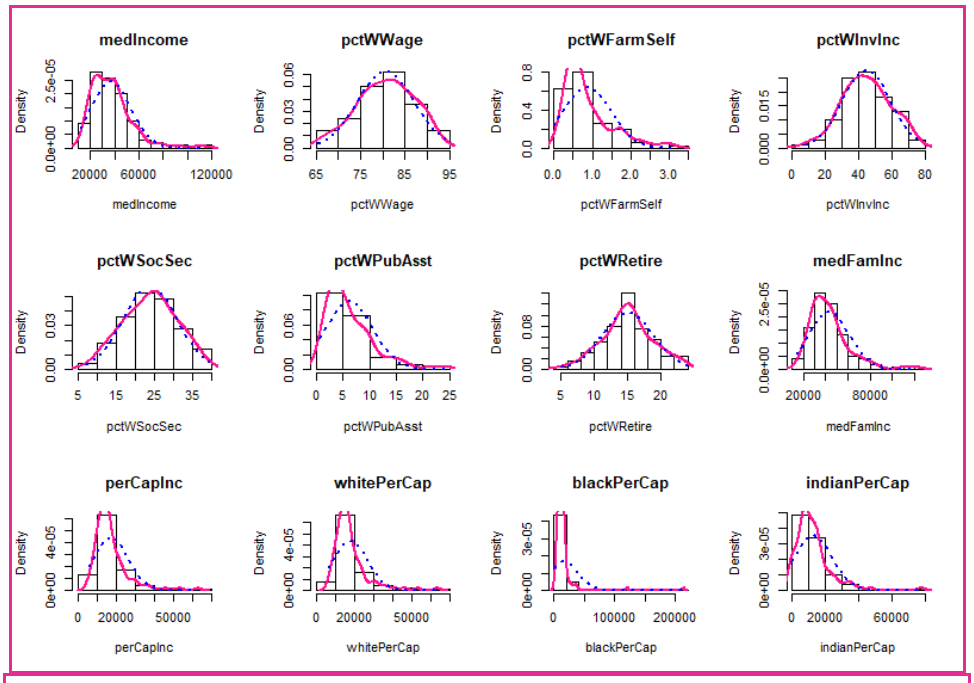
12. Appendix

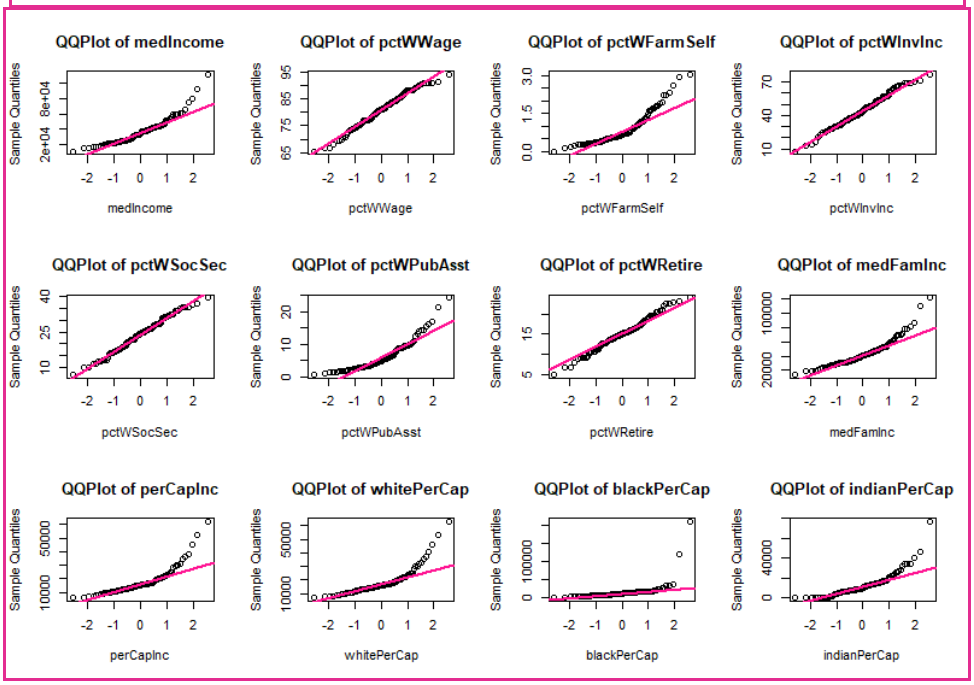
A. Explanatory Data Analysis

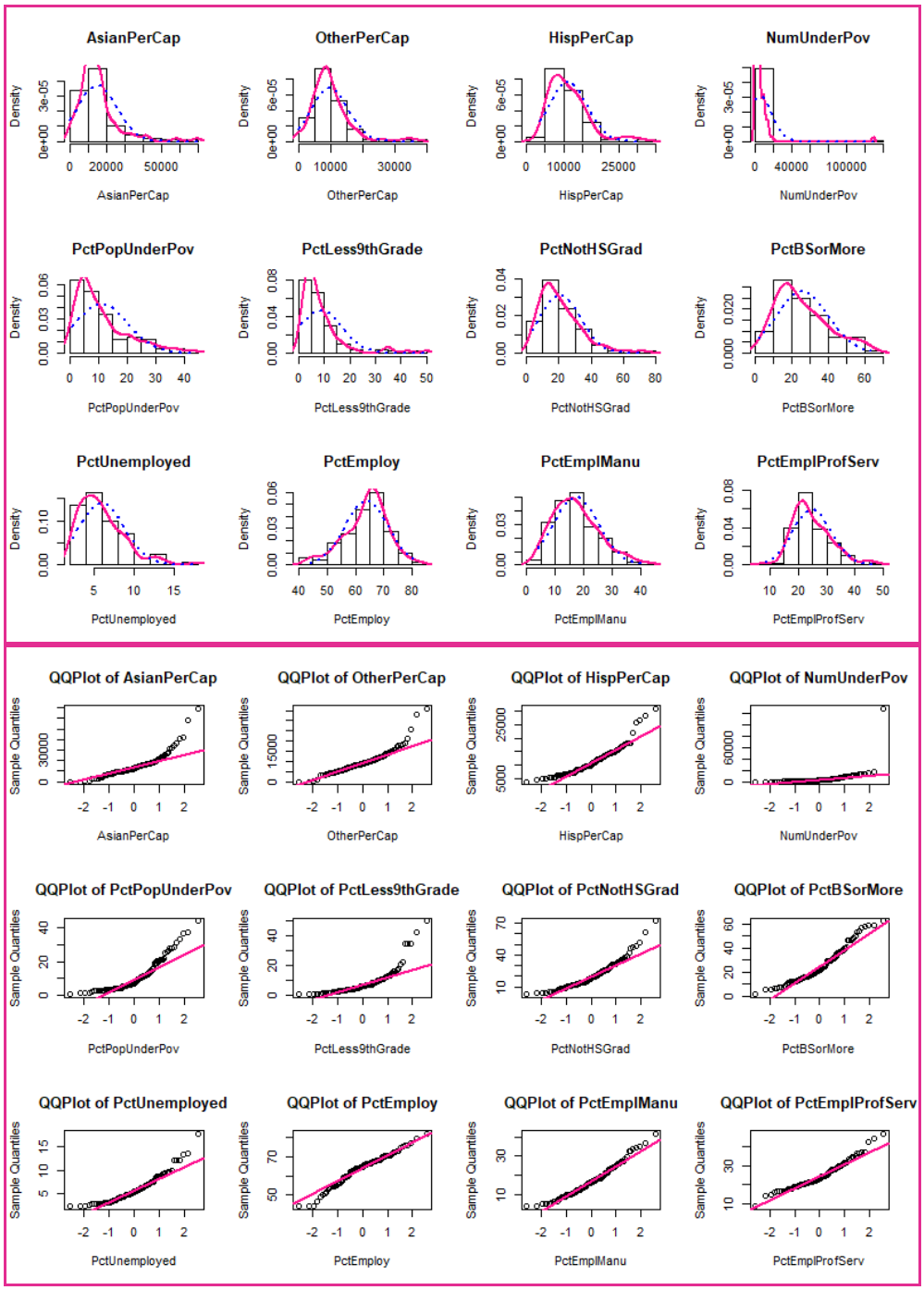
A.1 Histograms & QQplots of Numeric Variables

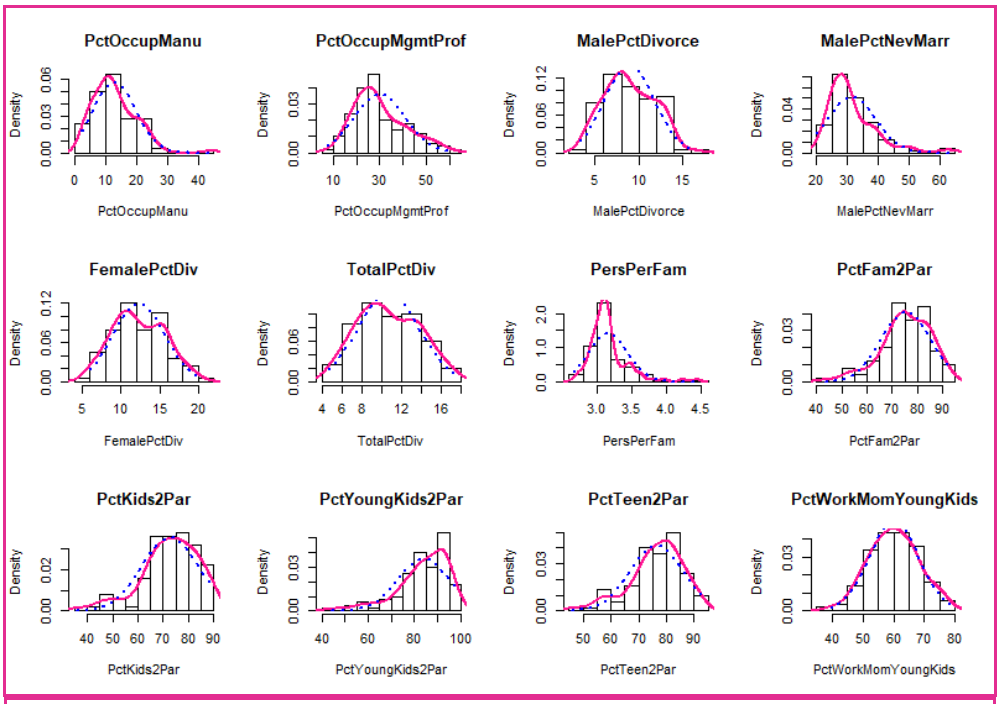


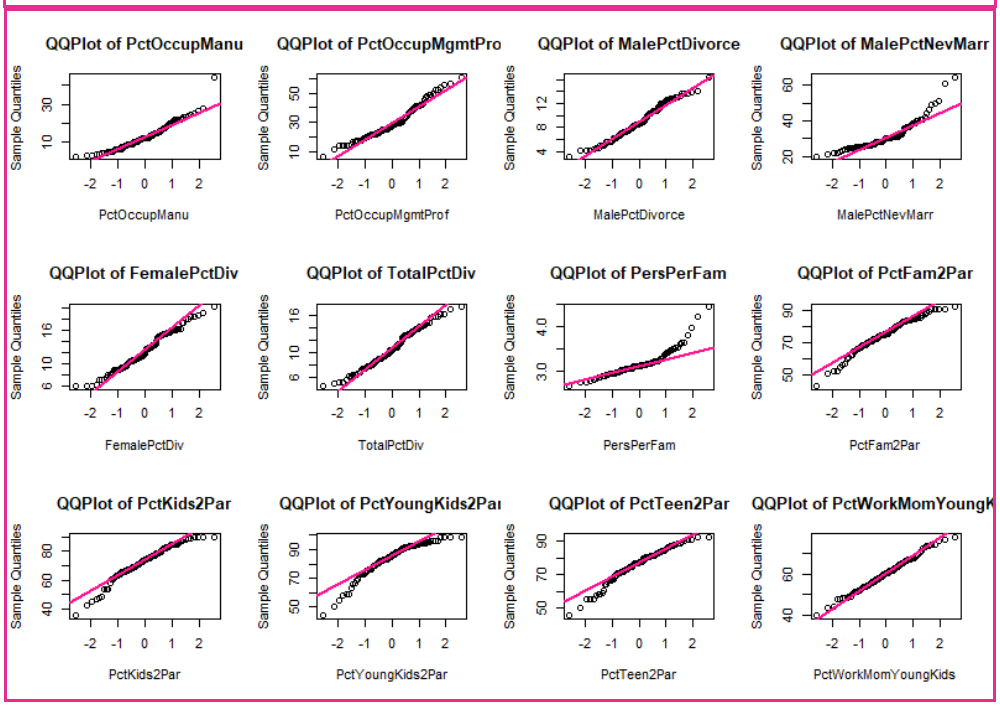


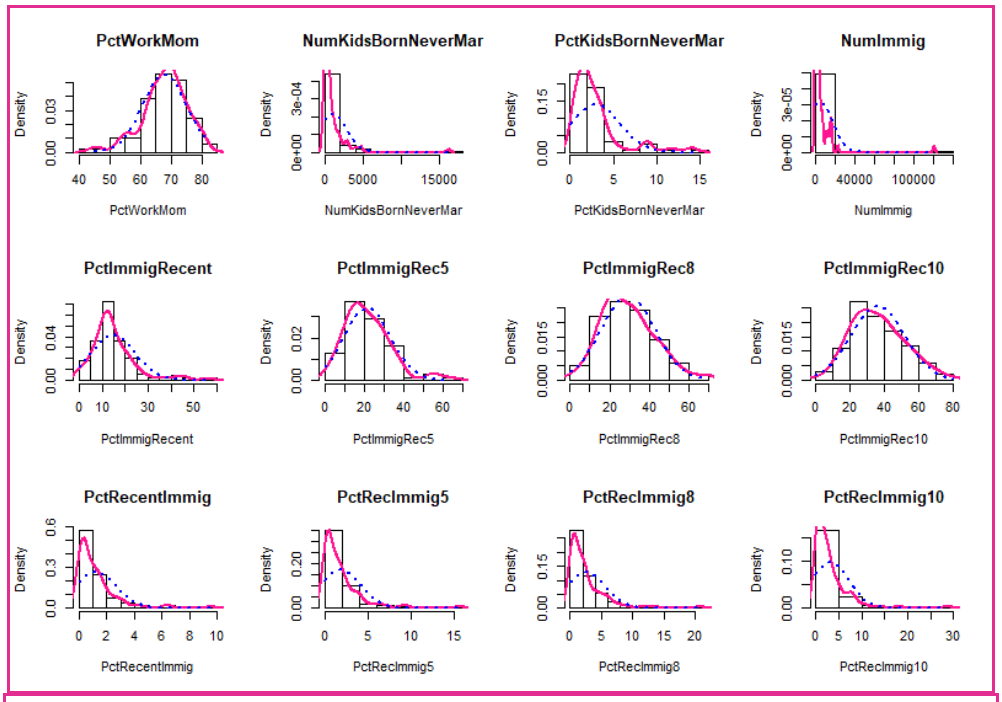

medIncome
pctWWage
pctWFarmSelf
pctWInvInc
pctWSocSec
pctWPubAsst
pctWRetire
medFamInc
perCapInc
whitePerCap
blackPerCap
indianPerCap
Density

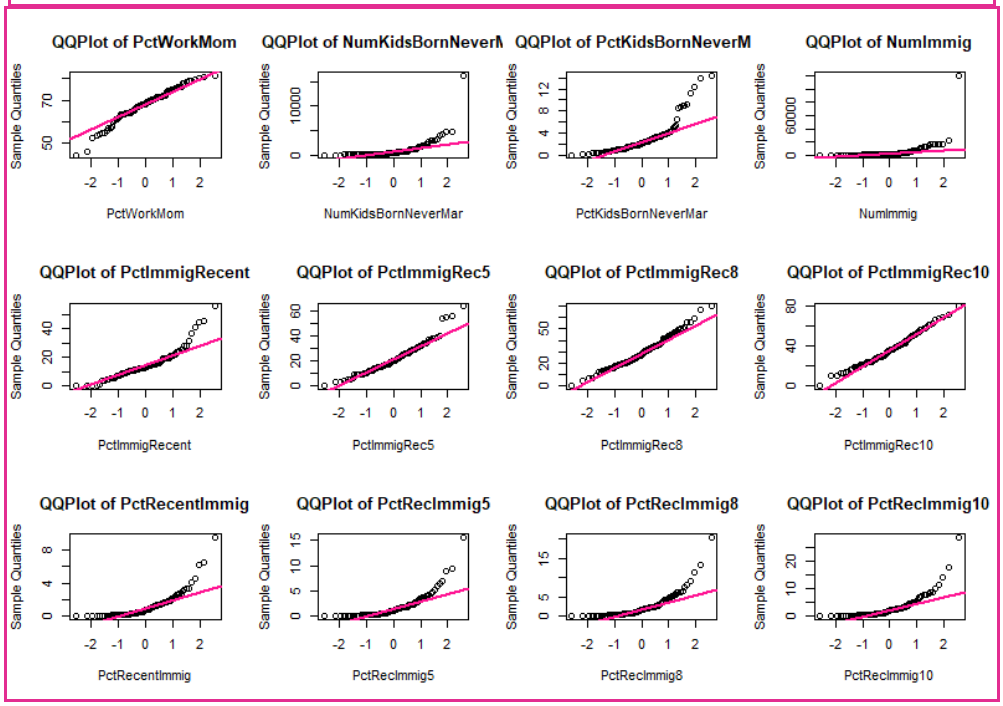

QQPlot of medIncome
QQPlot of pctWWage
QQPlot of pctWFarmSelf
QQPlot of pctWInvInc
QQPlot of pctWSocSec
QQPlot of pctWPubAsst
QQPlot of pctWRetire
QQPlot of medFamInc
QQPlot of perCapInc
QQPlot of whitePerCap
QQPlot of blackPerCap
QQPlot of indianPerCap
Sample Quantiles

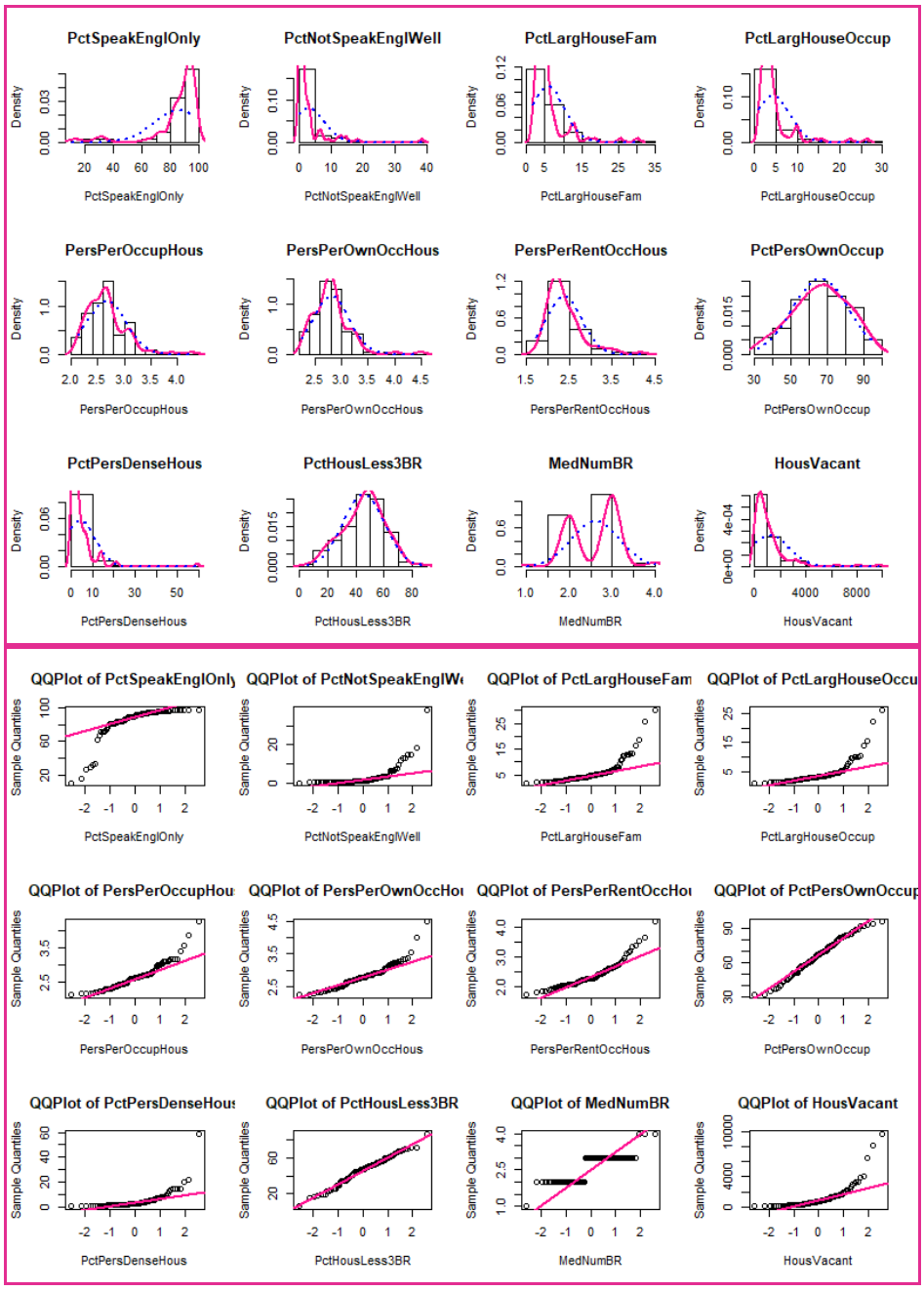

AsianPerCap
OtherPerCap
HispPerCap
NumUnderPov
PctPopUnderPov
PctLess9thGrade
PctNotHSGrad
PctBSorMore
PctUnemployed
PctEmploy
PctEmplManu
PctEmplProfServ
Density
QQPlot of AsianPerCap
QQPlot of OtherPerCap
QQPlot of HispPerCap
QQPlot of NumUnderPov
QQPlot of PctPopUnderPov
QQPlot of PctLess9thGrade
QQPlot of PctNotHSGrad
QQPlot of PctBSorMore
QQPlot of PctUnemployed
QQPlot of PctEmploy
QQPlot of PctEmplManu
QQPlot of PctEmplProfServ
Sample Quantiles

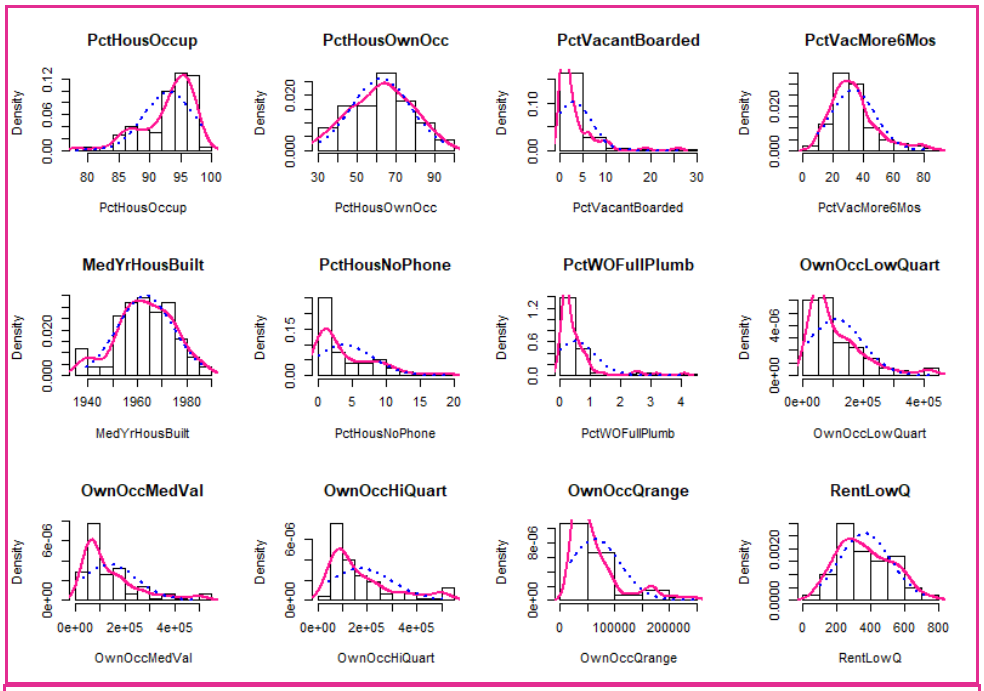

PctOccupManu
PctOccupMgmtProf
MalePctDivorce
MalePctNevMarr
FemalePctDiv
TotalPctDiv
PersPerFam
PctFam2Par
PctKids2Par
PctYoungKids2Par
PctTeen2Par
PctWorkMomYoungKids
Density

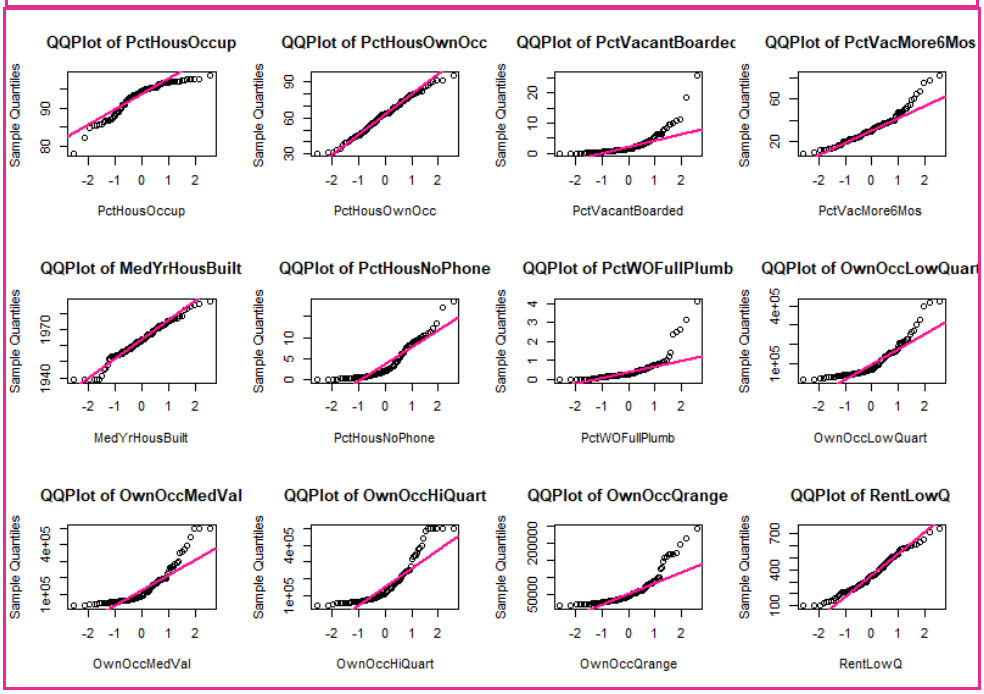

QQPlot of PctOccupManu
QQPlot of PctOccupMgmtPro
QQPlot of MalePctDivorce
QQPlot of MalePctNevMarr
QQPlot of FemalePctDiv
QQPlot of TotalPctDiv
QQPlot of PersPerFam
QQPlot of PctFam2Par
QQPlot of PctKids2Par
QQPlot of PctYoungKids2Pa
QQPlot of PctTeen2Par
QQPlot of PctWorkMomYoungK
Sample Quantiles

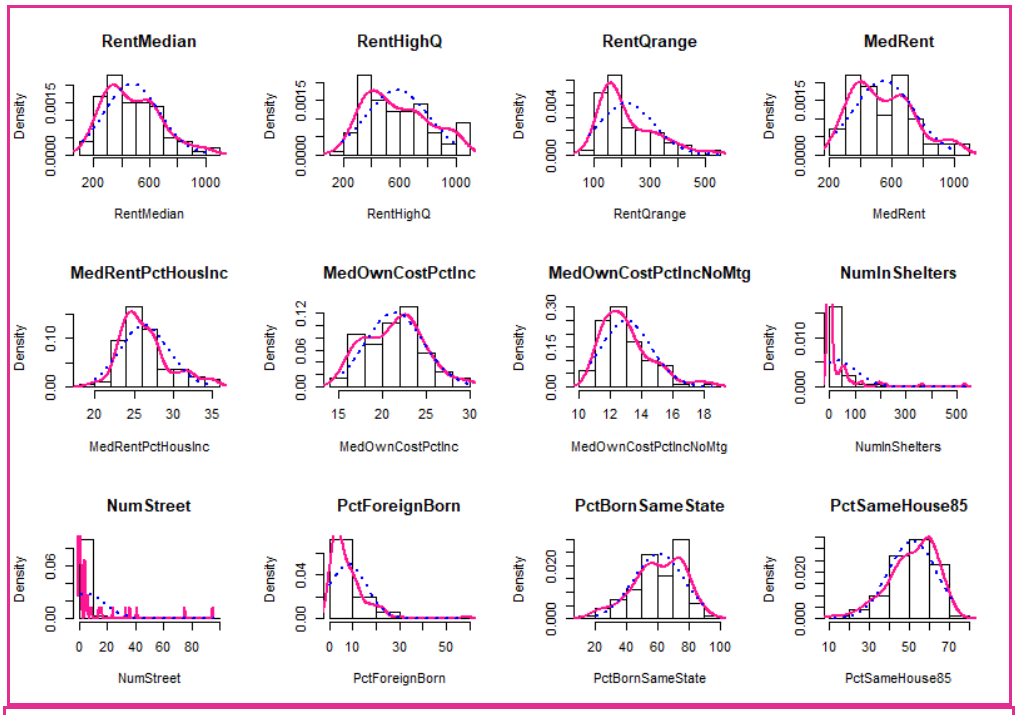

PctWorkMom
NumKidsBornNeverMar
PctKidsBornNeverMar
NumImmig
PctImmigRecent
PctImmigRec5
PctImmigRec8
PctImmigRec10
PctRecentImmig
PctRecImmig5
PctRecImmig8
PctRecImmig10
Density

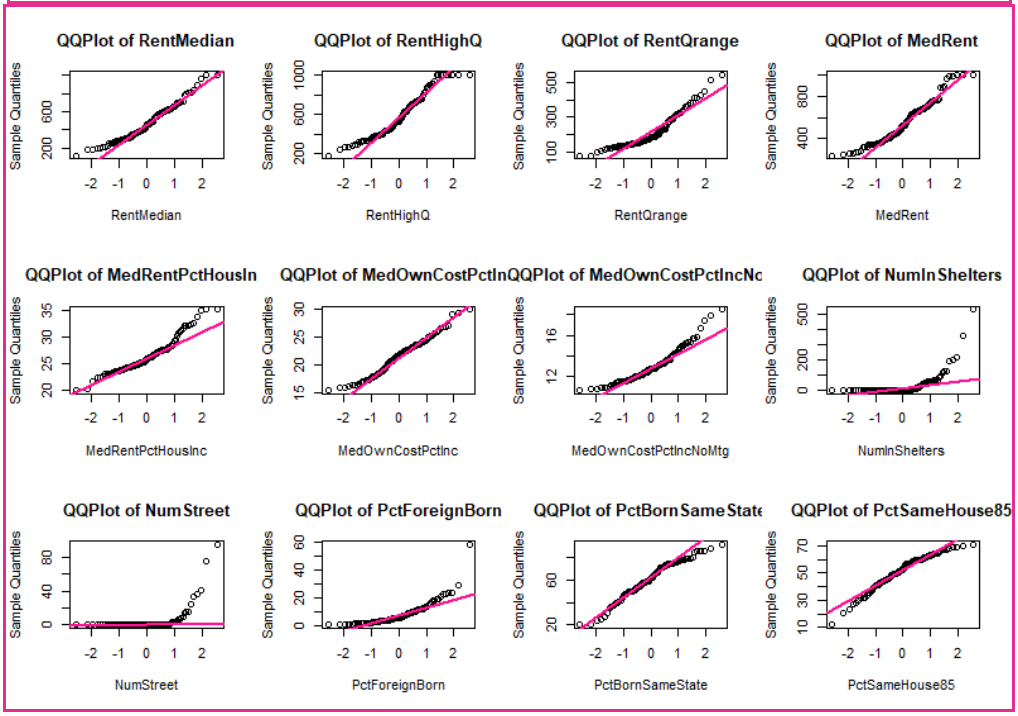

QQPlot of PctWorkMom
QQPlot of NumKidsBornNeverM
QQPlot of PctKidsBornNeverM
QQPlot of NumImmig
QQPlot of PctImmigRecent
QQPlot of PctImmigRec5
QQPlot of PctImmigRec8
QQPlot of PctImmigRec10
QQPlot of PctRecentImmig
QQPlot of PctRecImmig5
QQPlot of PctRecImmig8
QQPlot of PctRecImmig10
Sample Quantiles


PctSpeakEnglOnly
PctNotSpeakEnglWell
PctLargHouseFam
PctLargHouseOccup
PersPerOccupHous
PersPerOwnOccHous
PersPerRentOccHous
PctPersOwnOccup
PctPersDenseHous
PctHousLess3BR
MedNumBR
HousVacant
Density
QQPlot of PctSpeakEnglOnly
QQPlot of PctNotSpeakEnglWell
QQPlot of PctLargHouseFam
QQPlot of PctLargHouseOccup
QQPlot of PersPerOccupHous
QQPlot of PersPerOwnOccHous
QQPlot of PersPerRentOccHous
QQPlot of PctPersOwnOccup
QQPlot of PctPersDenseHous
QQPlot of PctHousLess3BR
QQPlot of MedNumBR
QQPlot of HousVacant
Sample Quantiles


PctHousOccup
PctHousOwnOcc
PctVacantBoarded
PctVacMore6Mos
MedYrHousBuilt
PctHousNoPhone
PctWOFullPlumb
OwnOccLowQuart
OwnOccMedVal
OwnOccHiQuart
OwnOccQrange
RentLowQ
Density


QQPlot of PctHousOccup
QQPlot of PctHousOwnOcc
QQPlot of PctVacantBoardec
QQPlot of PctVacMore6Mos
QQPlot of MedYrHousBuilt
QQPlot of PctHousNoPhone
QQPlot of PctWOFullPlumb
QQPlot of OwnOccLowQuart
QQPlot of OwnOccMedVal
QQPlot of OwnOccHiQuart
QQPlot of OwnOccQrange
QQPlot of RentLowQ
Sample Quantiles


RentMedian
RentHighQ
RentQrange
MedRent
MedRentPctHousInc
MedOwnCostPctInc
MedOwnCostPctIncNoMtg
NumInShelters
NumStreet
PctForeignBborn
PctBornSameState
PctSameHouse85
Density


QQPlot of RentMedian
QQPlot of RentHighQ
QQPlot of RentQrange
QQPlot of MedRent
QQPlot of MedRentPctHousIn
QQPlot of MedOwnCostPctIn
QQPlot of MedOwnCostPctIncNc
QQPlot of NumInShelters
QQPlot of NumStreet
QQPlot of PctForeignBorn
QQPlot of PctBornSameState
QQPlot of PctSameHouse85
Sample Quantiles

B. The Regression Model

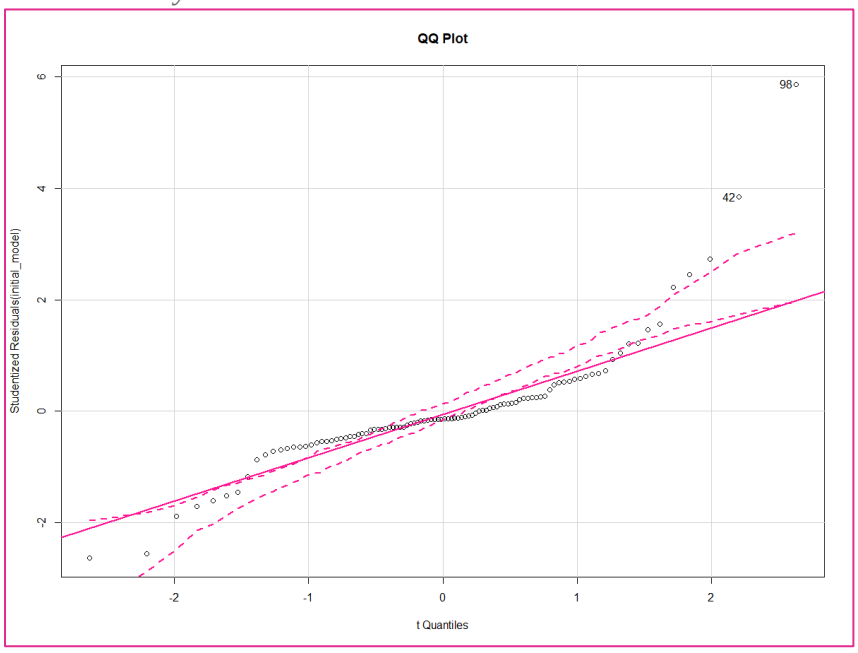
B.1 The initial Model

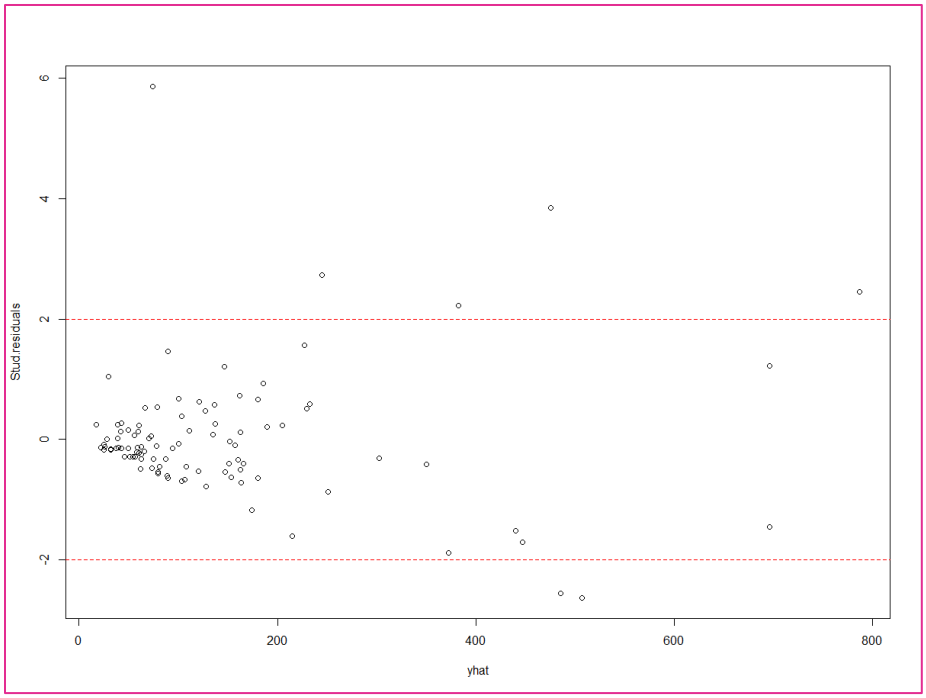
B.1.1 Output of Code

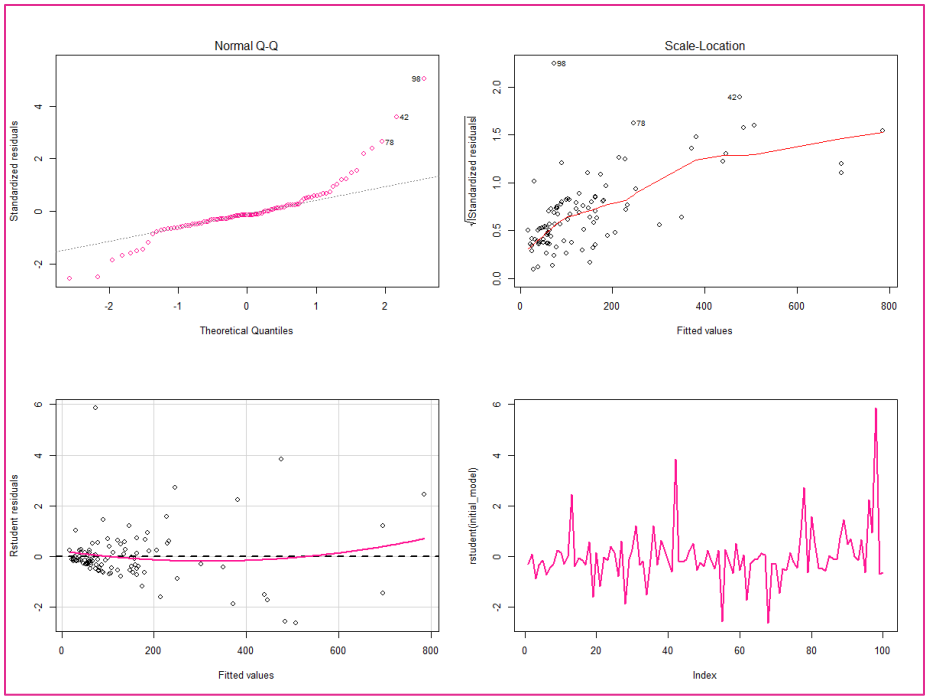
```
Call:
lm(formula = robbbPerPop ~ ., data = regressionmodel)

Residuals:
   Min     1Q Median     3Q    Max
-305.8  -56.7  -17.3   31.6  634.9

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)         312.74     142.38    2.20    0.030 *
racepctblack          2.26       1.84    1.23    0.222
racePctWhite         -3.08       1.46   -2.12    0.037 *
PctKidsBornNeverMar  25.82       9.97    2.59    0.011 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 127 on 96 degrees of freedom
Multiple R-squared:  0.587,     Adjusted R-squared:  0.575
F-statistic: 45.6 on 3 and 96 DF,  p-value: <2e-16
```

```
> round(vif(initial_model),2)
       racepctblack        racePctWhite PctKidsBornNeverMar
               4.47                3.62                4.88
>
```

B.1.2 Summary Plots







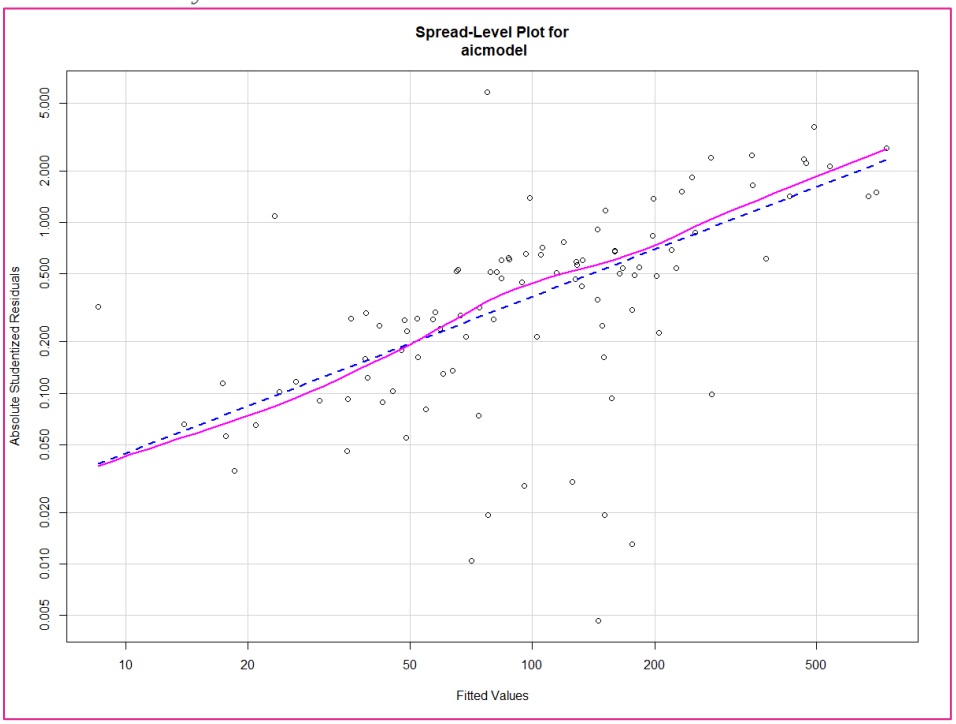
B.2 The Stepwise AIC Model

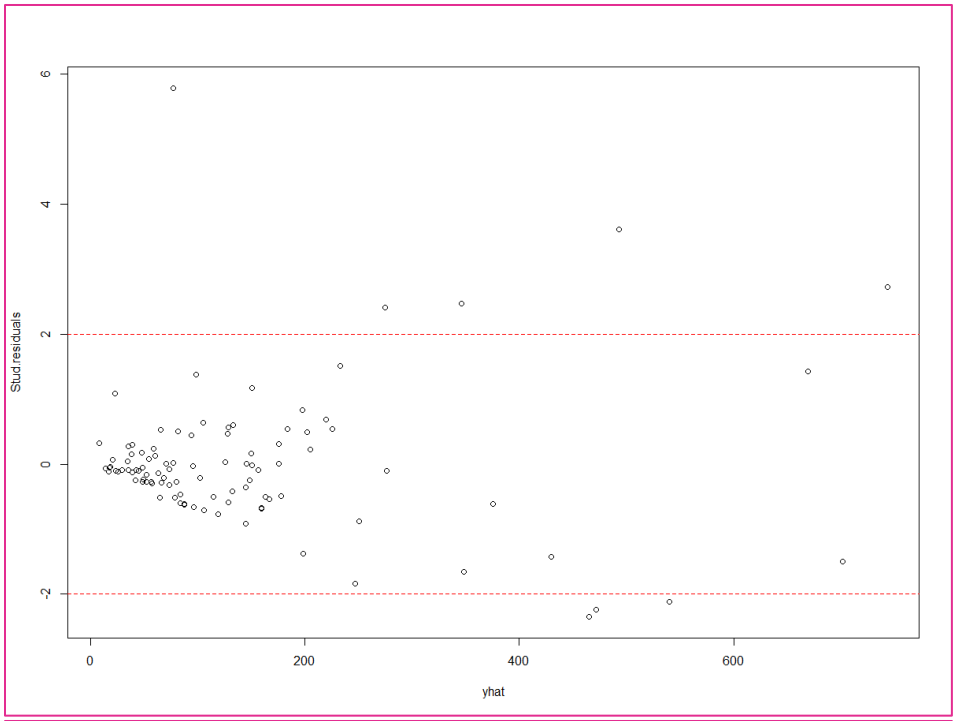
B.2.1 Output of Code

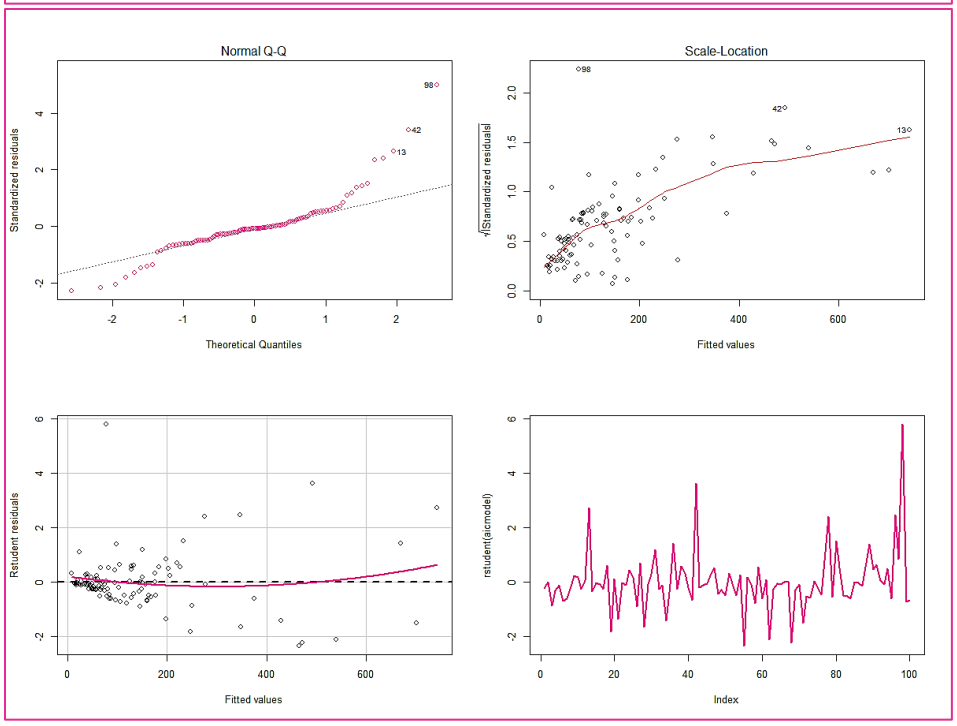
```
Call:
lm(formula = robbbPerPop ~ racePctWhite + PctKidsBornNeverMar,
    data = regressionmodel)

Residuals:
   Min     1Q Median     3Q    Max
-283.6  -62.2  -11.9   35.3  631.6

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)         362.09     136.95    2.64   0.0096 **
racePctWhite         -3.69       1.38   -2.68   0.0086 **
PctKidsBornNeverMar  32.99       8.10    4.07  9.5e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 128 on 97 degrees of freedom
Multiple R-squared:  0.581,     Adjusted R-squared:  0.572
F-statistic: 67.2 on 2 and 97 DF,  p-value: <2e-16
```

```
> round(vif(aicmodel),2)
       racePctWhite PctKidsBornNeverMar
               3.21                3.21
>
```

B.2.2 Summary Plots







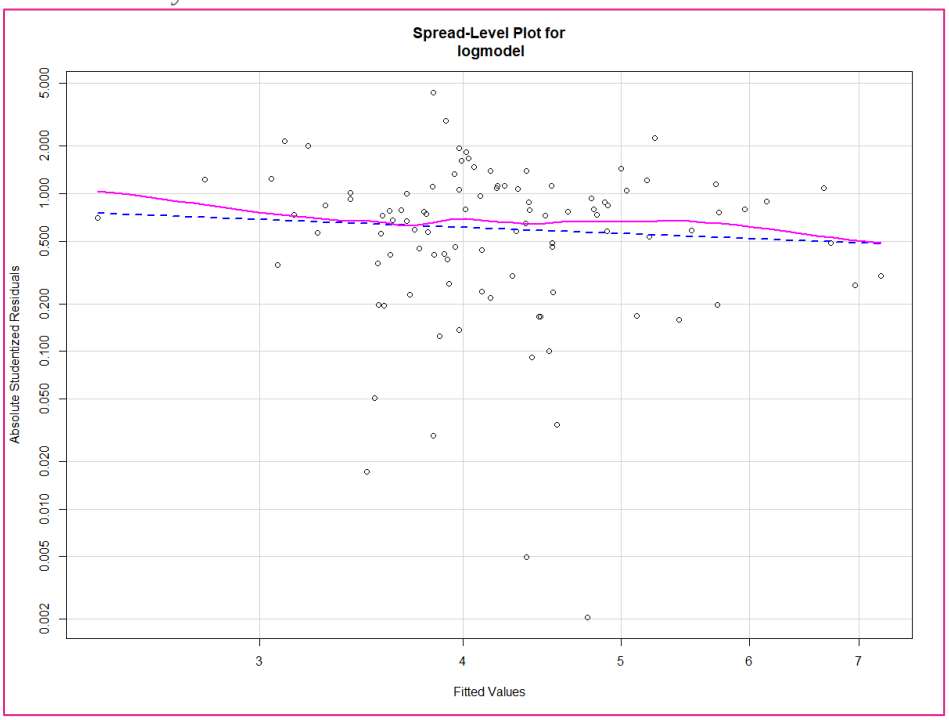
B.3 The Log Transformation

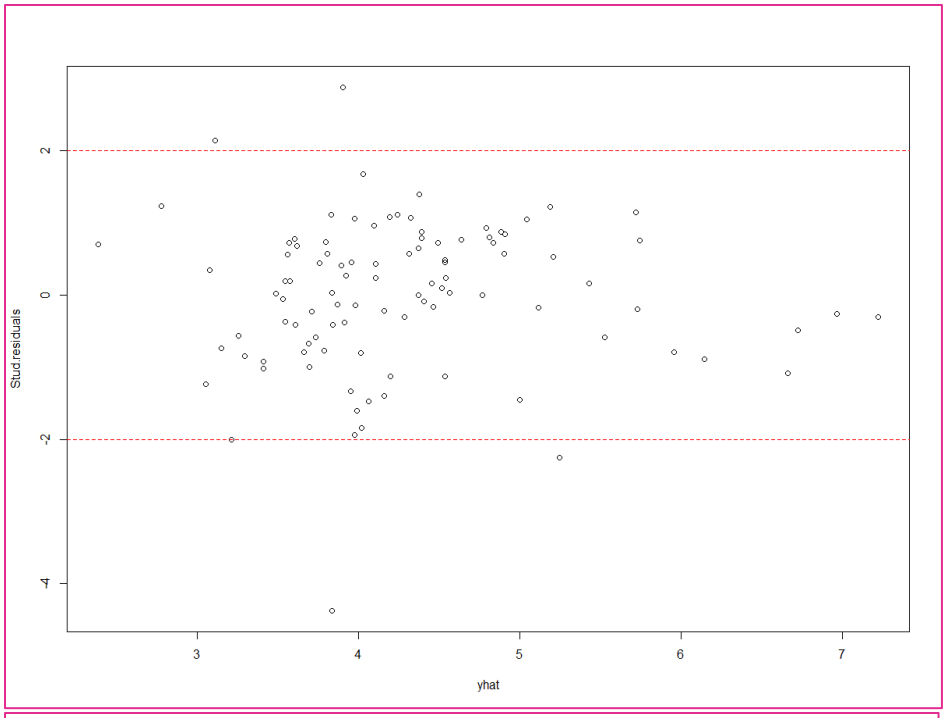
B.3.1 Output of Code

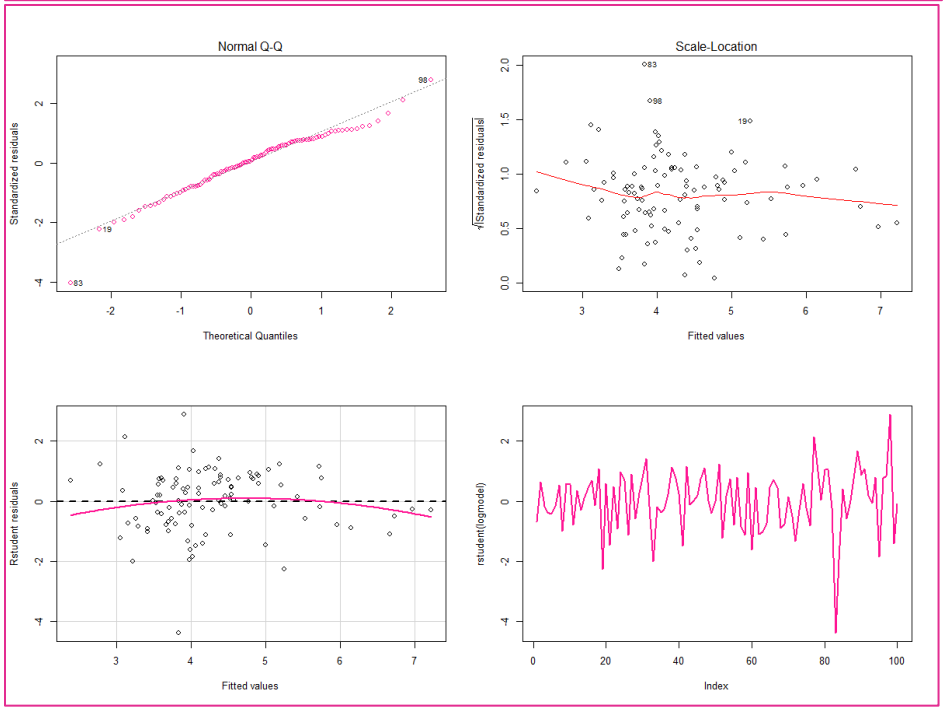
```
Call:
lm(formula = log(robbbPerPop + 1) ~ +racePctWhite + log(PctKidsBornNeverMar),
    data = regressionmodel)

Residuals:
   Min     1Q Median     3Q    Max
-3.833 -0.587  0.065  0.701  2.661

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)              6.65482    0.71875    9.26  5.3e-15 ***
racePctWhite            -0.03192    0.00768   -4.16  7.0e-05 ***
log(PctKidsBornNeverMar)  0.46196    0.13600    3.40  0.00099 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.966 on 97 degrees of freedom
Multiple R-squared:  0.459,     Adjusted R-squared:  0.448
F-statistic: 41.1 on 2 and 97 DF,  p-value: 1.15e-13
```

```
> round(vif(logmodel),2)
            racePctWhite log(PctKidsBornNeverMar)
                    1.74                     1.74
>
```

B.3.2 Summary Plots





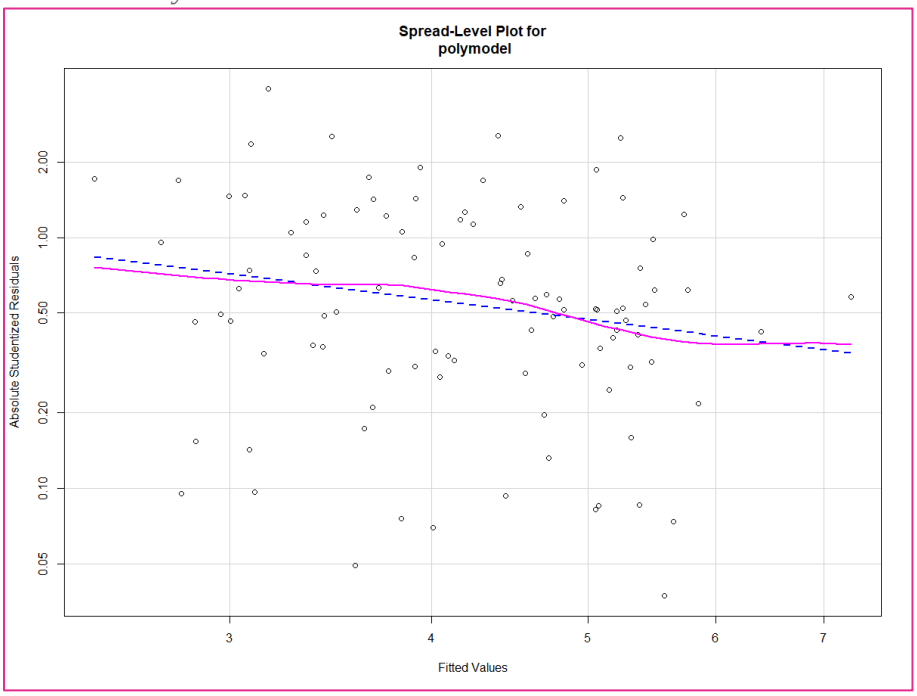


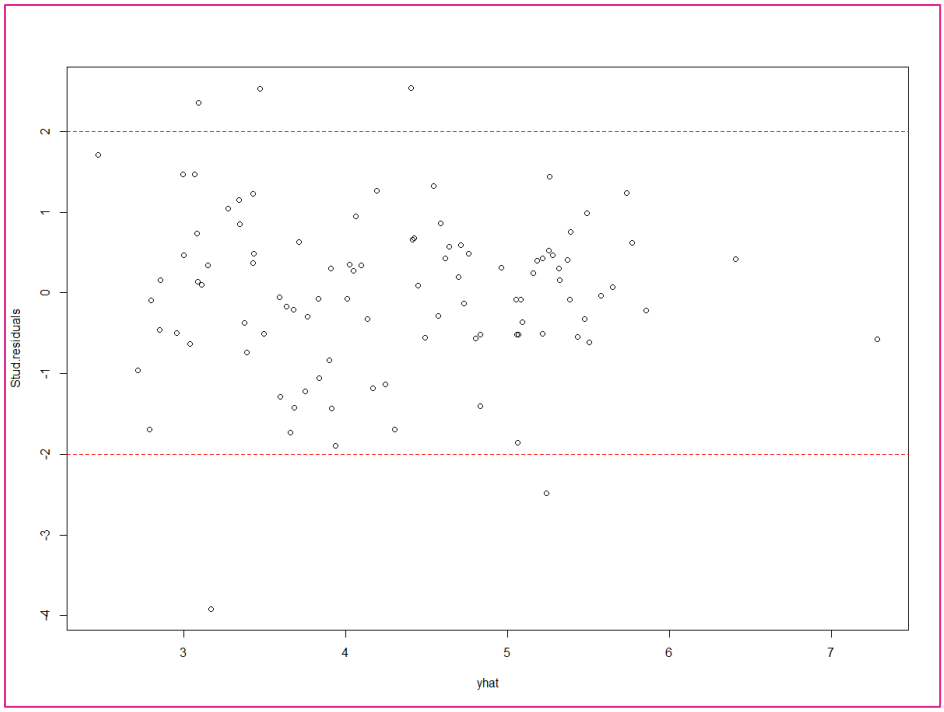
B.4 The Polynomial Transformation

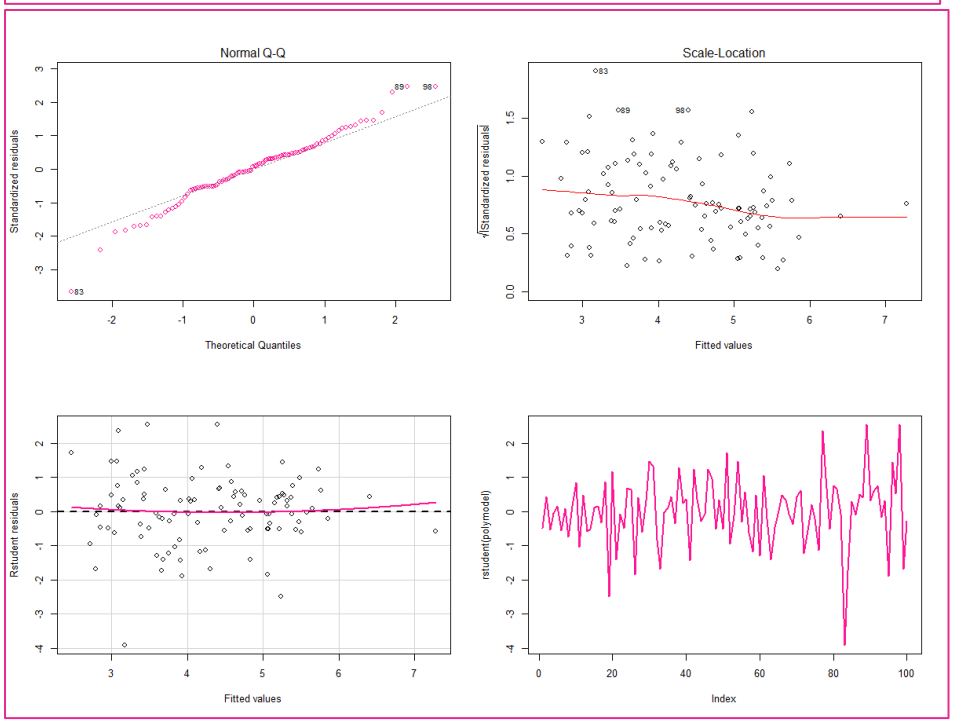
B.4.1 Output of Code

```
Call:
lm(formula = log(robbbPerPop + 1) ~ +poly(racePctWhite, 3) +
    log(PctKidsBornNeverMar), data = regressionmodel)

Residuals:
   Min     1Q Median     3Q    Max
-3.170 -0.454  0.074  0.472  2.164

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                  4.013      0.125   32.02   <2e-16 ***
poly(racePctWhite, 3)1      -5.730      1.184   -4.84    5e-06 ***
poly(racePctWhite, 3)2      -2.618      0.902   -2.90   0.0046 **
poly(racePctWhite, 3)3      -2.886      0.893   -3.23   0.0017 **
log(PctKidsBornNeverMar)     0.392      0.127    3.09   0.0027 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.892 on 95 degrees of freedom
Multiple R-squared:  0.548,     Adjusted R-squared:  0.529
F-statistic: 28.8 on 4 and 95 DF,  p-value: 1.1e-15

> round(vif(polymodel),2)
  poly(racePctWhite, 3)1   poly(racePctWhite, 3)2   poly(racePctWhite, 3)3 log(PctKidsBornNeverMar)
                    1.76                     1.02                     1.00                     1.78
> |
```

B.4.2 Summary Plots





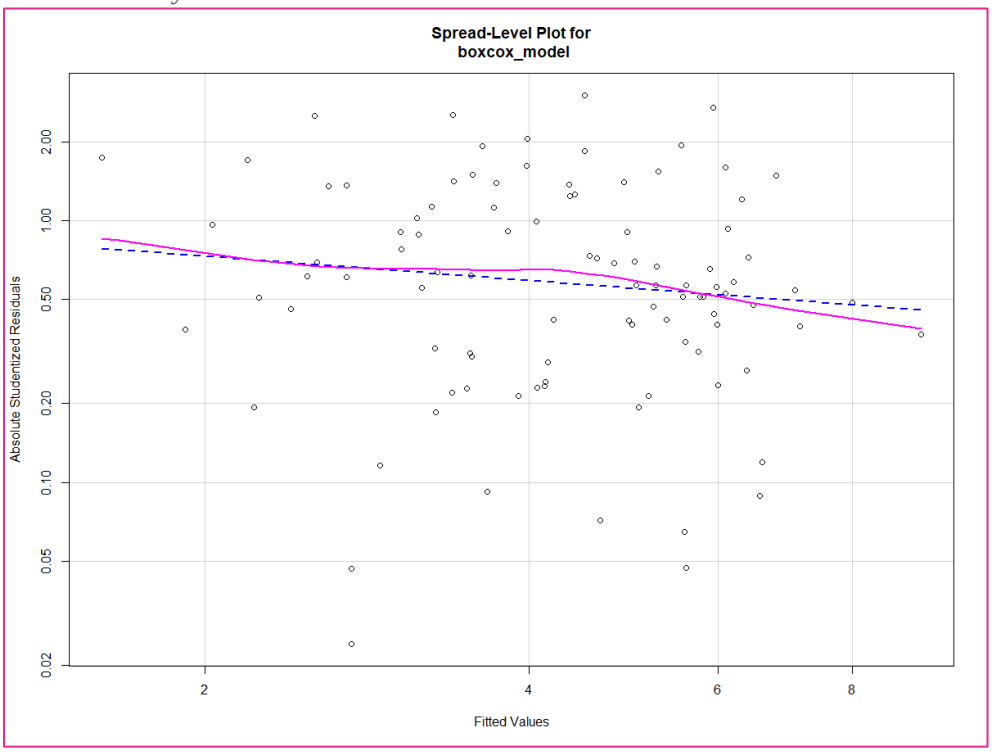


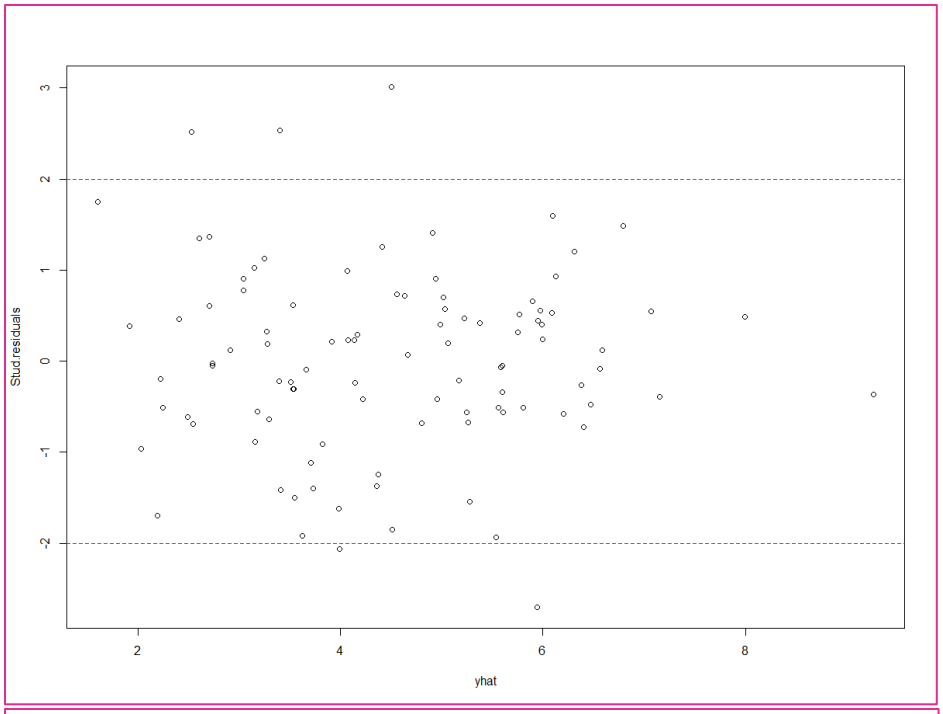
B.5 The Box-Cox Transformation

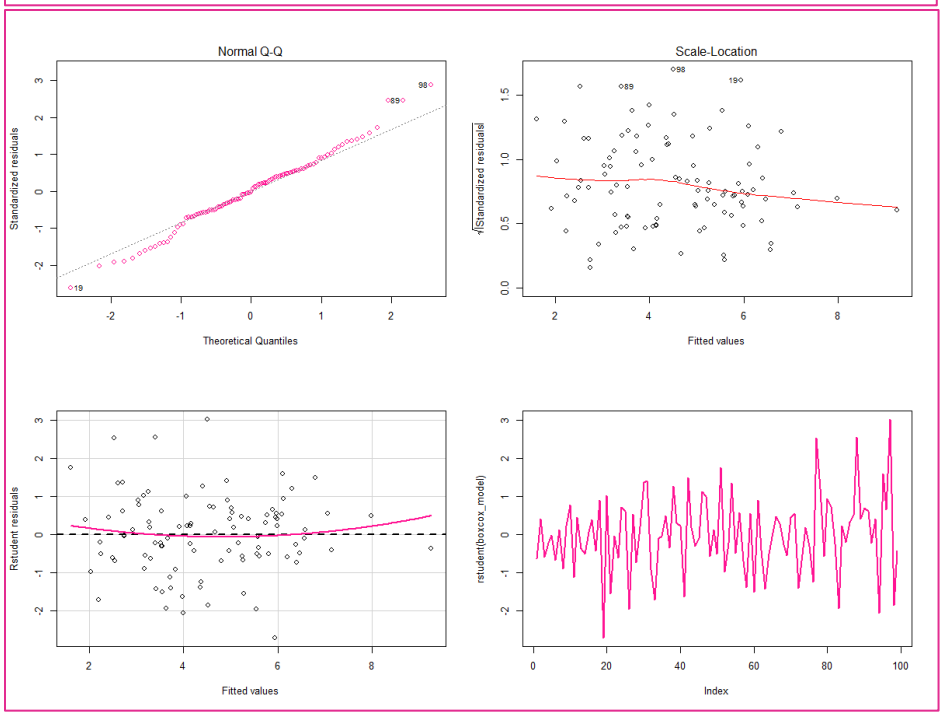
B.5.1 Output of Code

```
Call:
lm(formula = powerTransform(log(robbbPerPop), lambda) ~ +poly(racePctWhite,
    3) + log(PctKidsBornNeverMar), data = boxcox)

Residuals:
   Min     1Q Median     3Q    Max
-3.342 -0.737 -0.031  0.732  3.741

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)                4.022      0.186   21.62  < 2e-16 ***
poly(racePctWhite, 3)1    -8.419      1.756   -4.79  6.1e-06 ***
poly(racePctWhite, 3)2    -3.259      1.335   -2.44  0.01648 *
poly(racePctWhite, 3)3    -3.880      1.320   -2.94  0.00413 **
log(PctKidsBornNeverMar)   0.695      0.189    3.68  0.00039 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.32 on 94 degrees of freedom
Multiple R-squared:  0.572,     Adjusted R-squared:  0.553
F-statistic: 31.4 on 4 and 94 DF,  p-value: <2e-16

> round(vif(boxcox_model),2)
  poly(racePctWhite, 3)1   poly(racePctWhite, 3)2   poly(racePctWhite, 3)3 log(PctKidsBornNeverMar)
                    1.77                     1.03                     1.00                     1.80
>
```

B.5.2 Summary Plots





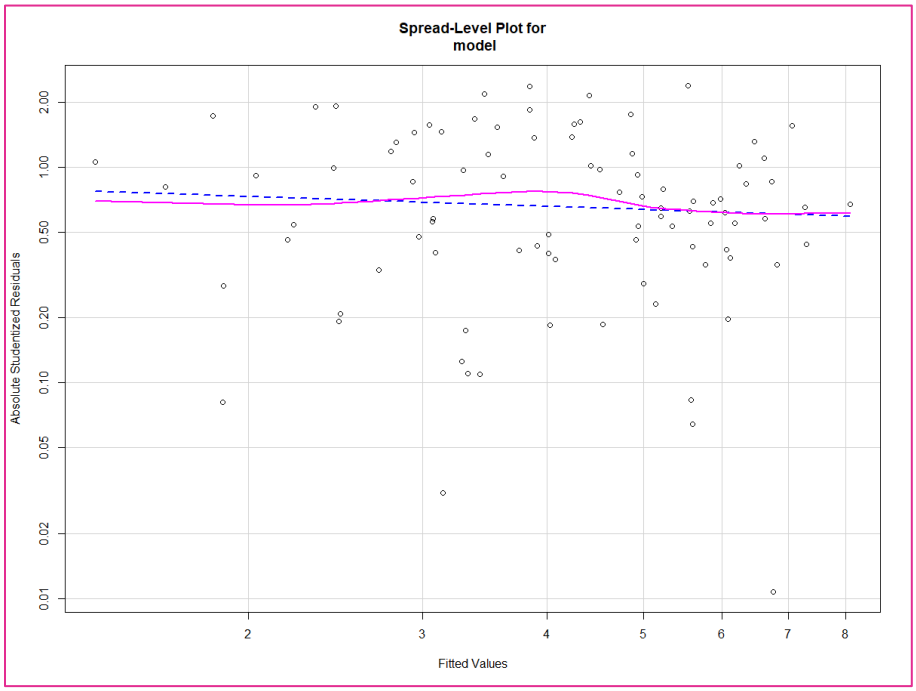


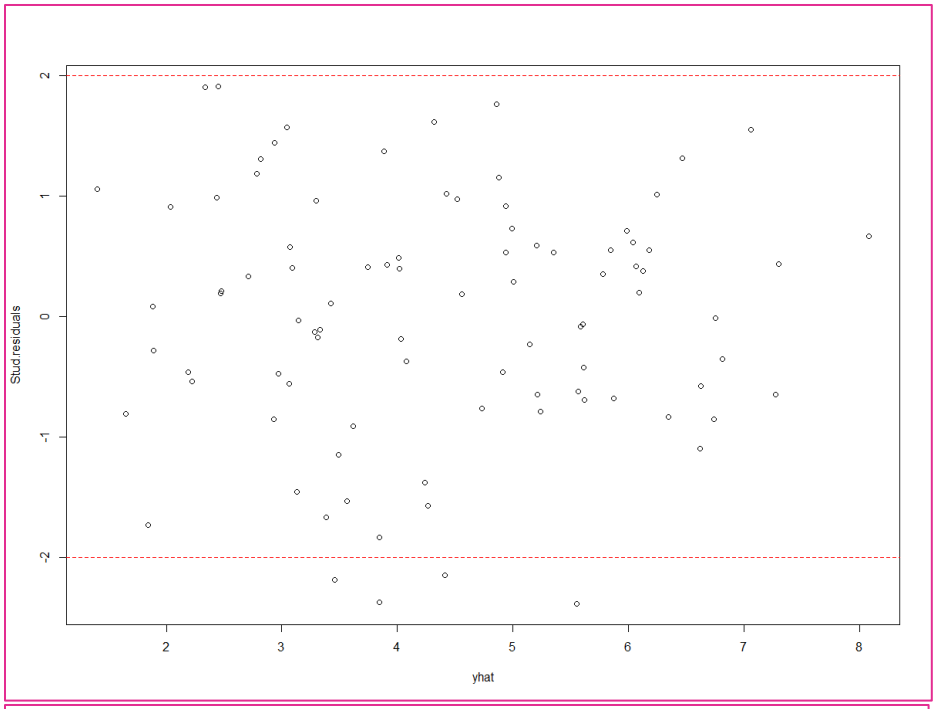
B.6 Outliers Detection (Cook's Distance)

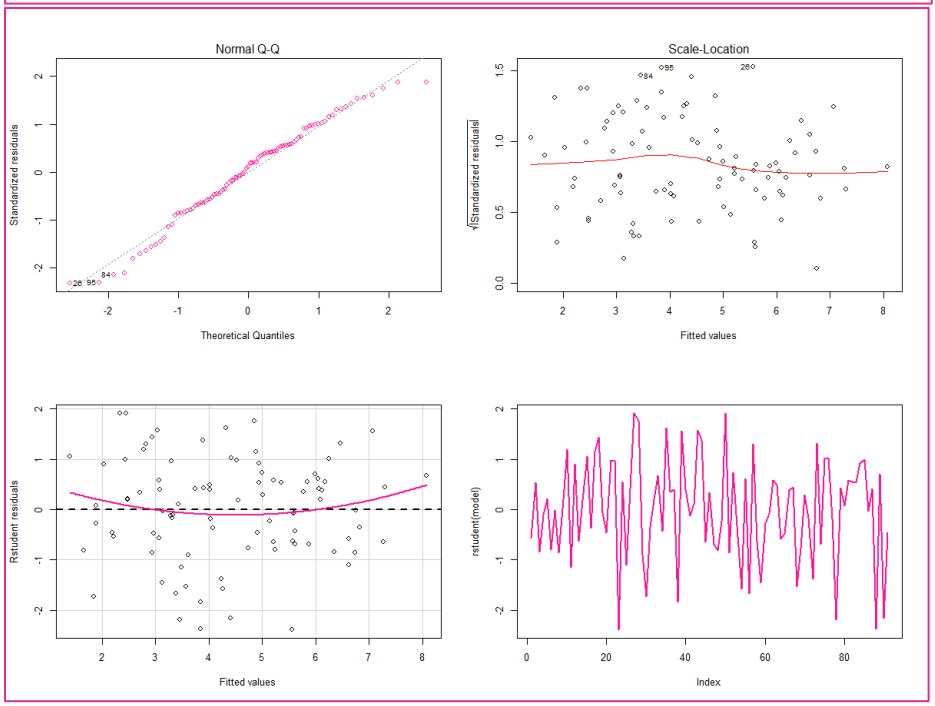
B.6.1 Output of Code

```
Call:
lm(formula = powerTransform(log(robbbPerPop), lambda) ~ +poly(racePctWhite,
    3) + log(PctKidsBornNeverMar), data = cooks)

Residuals:
   Min     1Q Median     3Q    Max
-2.498 -0.686  0.118  0.645  2.011

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                  3.812      0.168   22.76  < 2e-16 ***
poly(racePctWhite, 3)1      -8.014      1.469   -5.46  4.6e-07 ***
poly(racePctWhite, 3)2      -4.482      1.108   -4.04  0.00011 ***
poly(racePctWhite, 3)3      -2.782      1.092   -2.55  0.01265 *
log(PctKidsBornNeverMar)     0.802      0.173    4.63  1.3e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.09 on 86 degrees of freedom
Multiple R-squared:  0.681,     Adjusted R-squared:  0.667
F-statistic:   46 on 4 and 86 DF,  p-value: <2e-16

> round(vif(model),2)
  poly(racePctWhite, 3)1   poly(racePctWhite, 3)2   poly(racePctWhite, 3)3 log(PctKidsBornNeverMar)
                    1.81                     1.03                     1.00                     1.84
```

B.6.2 Summary Plots




Stud.residuals
yhat


Normal Q-Q
Standardized residuals
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
Fitted values
Rstudent residuals
Fitted values
rstudent(model)
Index

C. Cross Validation and out of Sample Predictive Ability of the model

C.1 The train dataset

C.1.1 The log model

```
Linear Regression

91 samples
 2 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 81, 83, 83, 82, 82, 81, ...
Resampling results:

  RMSE   Rsquared  MAE
  0.894  0.637     0.762

Tuning parameter 'intercept' was held constant at a value of TRUE
>
```

C.1.2 The polynomial model

```
Linear Regression

91 samples
 2 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 82, 81, 82, 83, 81, 82, ...
Resampling results:

  RMSE   Rsquared  MAE
  0.721  0.665     0.599

Tuning parameter 'intercept' was held constant at a value of TRUE
>
```

C.1.3 The box-cox model

```
Linear Regression

91 samples
 2 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 82, 81, 81, 83, 80, 82, ...
Resampling results:

  RMSE   Rsquared  MAE
  0.711  0.652     0.589

Tuning parameter 'intercept' was held constant at a value of TRUE
>
```

C.2 The test dataset

C.2.1 The log model

```
> print(log_test)
Linear Regression

100 samples
  2 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 89, 91, 91, 89, 89, 90, ...
Resampling results:

  RMSE   Rsquared  MAE
  0.991  0.153     0.777

Tuning parameter 'intercept' was held constant at a value of TRUE
```

C.2.2 The polynomial model

```
Linear Regression

100 samples
  2 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 91, 90, 90, 90, 91, 89, ...
Resampling results:

  RMSE  Rsquared  MAE
  1.04  0.112     0.806

Tuning parameter 'intercept' was held constant at a value of TRUE
```

C.2.3 The box-cox model

```
Linear Regression

100 samples
  2 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 90, 90, 91, 90, 90, 90, ...
Resampling results:

  RMSE  Rsquared  MAE
  1.39  0.085     1.12

Tuning parameter 'intercept' was held constant at a value of TRUE
```

D. Exploration of Other Types of Regression

```
Call:
lmrob(formula = (log(robbbPerPop)) ~ +(racePctWhite) + log(PctKidsBornNeverMar),
    data = cooks, method = "MM", fast.s.large.n = Inf, cov = ".vcov.w")
 \--> method = "MM"
Residuals:
     Min       1Q   Median       3Q      Max
-2.0148 -0.5961  0.0669  0.6253  1.2035

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)             6.27059    0.75585    8.30  1.1e-12 ***
racePctWhite           -0.02872    0.00802   -3.58  0.00056 ***
log(PctKidsBornNeverMar)  0.62717    0.13531    4.64  1.2e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Robust residual standard error: 0.825
Multiple R-squared:  0.555,     Adjusted R-squared:  0.545
Convergence in 10 IRWLS iterations

Robustness weights:
 6 weights are ~= 1. The remaining 85 ones are summarized as
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.530   0.886   0.943   0.915   0.979   0.999
Algorithmic parameters:
       tuning.chi                bb        tuning.psi        refine.tol           rel.tol         scale.tol         solve.tol
        1.55e+00         5.00e-01          4.69e+00          1.00e-07          1.00e-07          1.00e-10          1.00e-07
      eps.outlier             eps.x warn.limit.reject warn.limit.meanrw
        1.10e-03         1.80e-10          5.00e-01          5.00e-01
  nResample      max.it    best.r.s    k.fast.s       k.max maxit.scale   trace.lev         mts  compute.rd
        500          50           2           1         200         200           0        1000           0
               psi           subsampling                 cov compute.outlier.stats
         "bisquare"         "nonsingular"           ".vcov.w"                  "SM"
seed : int(0)
> round(vif(robust_model),1)
           racePctWhite log(PctKidsBornNeverMar)
                    1.8                     1.8
```

13. Code Appendix

```
library(car);library(caret);library(corrplot);library(dplyr);library(forcats);library(ggplot2);library(ggrepel)
library(glmnet);library(gvlma);library(Hmisc);library(lars);library(lawstat);library(MASS);library(nortest)
library(psych);library(summarytools);library(scales);library(usmap);library(vioplot)


setwd("D:/MSc Business Analytics/Statistics I/Main_Assignment")
dataset <- read.table("crime_40.dat", sep=",", header=F)
#_____________________________________________________________________________________________________________
#### DATA CLEANING & DATA TRANSFORMATION ####
dataset <- setNames(dataset,
c('communityname','state','countyCode','communityCode','fold','population','householdsize',
'racepctblack','racePctWhite','racePctAsian','racePctHisp','agePct12t21','agePct12t29',
'agePct16t24','agePct65up','numbUrban','pctUrban','medIncome','pctWWage','pctWFarmSelf',
'pctWInvInc','pctWSocSec','pctWPubAsst','pctWRetire','medFamInc','perCapInc','whitePerCap',
'blackPerCap','indianPerCap','AsianPerCap','OtherPerCap','HispPerCap','NumUnderPov',
'PctPopUnderPov','PctLess9thGrade','PctNotHSGrad','PctBSorMore','PctUnemployed',
'PctEmploy','PctEmplManu','PctEmplProfServ','PctOccupManu','PctOccupMgmtProf',
'MalePctDivorce','MalePctNevMarr','FemalePctDiv','TotalPctDiv','PersPerFam','PctFam2Par',
'PctKids2Par','PctYoungKids2Par','PctTeen2Par','PctWorkMomYoungKids','PctWorkMom',
'NumKidsBornNeverMar','PctKidsBornNeverMar','NumImmig','PctImmigRecent','PctImmigRec5',
'PctImmigRec8','PctImmigRec10','PctRecentImmig','PctRecImmig5','PctRecImmig8',
'PctRecImmig10','PctSpeakEnglOnly','PctNotSpeakEnglWell','PctLargHouseFam','PctLargHouseOccup',
'PersPerOccupHous','PersPerOwnOccHous','PersPerRentOccHous','PctPersOwnOccup','PctPersDenseHous',
'PctHousLess3BR','MedNumBR','HousVacant','PctHousOccup','PctHousOwnOcc','PctVacantBoarded',
'PctVacMore6Mos','MedYrHousBuilt','PctHousNoPhone','PctWOFullPlumb','OwnOccLowQuart',
'OwnOccMedVal','OwnOccHiQuart','OwnOccQrange','RentLowQ','RentMedian','RentHighQ',
'RentQrange','MedRent','MedRentPctHousInc','MedOwnCostPctInc','MedOwnCostPctIncNoMtg',
'NumInShelters','NumStreet','PctForeignBorn','PctBornSameState','PctSameHouse85',
'PctSameCity85','PctSameState85','LemasSwornFT','LemasSwFTPerPop','LemasSwFTFieldOps',
'LemasSwFTFieldPerPop','LemasTotalReq','LemasTotReqPerPop','PolicReqPerOffic',
'PolicPerPop','RacialMatchCommPol','PctPolicWhite','PctPolicBlack','PctPolicHisp',
'PctPolicAsian','PctPolicMinor','OfficAssgnDrugUnits','NumKindsDrugsSeiz','PolicAveOTWorked',
'LandArea','PopDens','PctUsePubTrans','PolicCars','PolicOperBudg','LemasPctPolicOnPatr',
'LemasGangUnitDeploy','LemasPctOfficDrugUn','PolicBudgPerPop','murders','murdPerPop',
'rapes','rapesPerPop','robberies','robbbPerPop','assaults','assaultPerPop','burglaries',
'burglPerPop','larcenies','larcPerPop','autoTheft','autoTheftPerPop','arsons',
'arsonsPerPop','ViolentCrimesPerPop','nonViolPerPop'))
dataset[dataset=="?"] <- NA
str(dataset)
```

```
colSums(sapply(dataset, is.na))
dataset <- subset(dataset, select=-c(LemasSwornFT, LemasSwFTPerPop, LemasSwFTFieldOps, LemasSwFTFieldPerPop,
LemasTotalReq, LemasTotReqPerPop, PolicReqPerOffic, PolicPerPop, RacialMatchCommPol, PctPolicWhite,
PctPolicBlack, PctPolicHisp, PctPolicAsian, PctPolicMinor,OfficAssgnDrugUnits, NumKindsDrugsSeiz,
PolicAveOTWorked, PolicCars, PolicOperBudg, LemasPctPolicOnPatr, LemasGangUnitDeploy, PolicBudgPerPop))

colSums(sapply(dataset, is.na))
datasetall <- dataset
# Removing communityname, countyCode, communityCode, fold since they are non-predictive
dataset <- subset(dataset, select=-c(communityname,countyCode,communityCode,fold))
#Transform factor variables to numeric
sapply(dataset,class)
dataset$rapes <- as.numeric(as.character(dataset$rapes))
dataset$rapesPerPop <- as.numeric(as.character(dataset$rapesPerPop))
dataset$assaults <- as.numeric(as.character(dataset$assaults))
dataset$assaultPerPop <- as.numeric(as.character(dataset$assaultPerPop))
dataset$arsons <- as.numeric(as.character(dataset$arsons))
dataset$arsonsPerPop <- as.numeric(as.character(dataset$arsonsPerPop))
dataset$ViolentCrimesPerPop <- as.numeric(as.character(dataset$ViolentCrimesPerPop))
dataset$nonViolPerPop <- as.numeric(as.character(dataset$nonViolPerPop))

# Transform integer variables to numeric
columns <-c("population","numbUrban","medIncome","blackPerCap","whitePerCap","perCapInc","medFamInc",
            "indianPerCap", "AsianPerCap", "OtherPerCap", "HispPerCap", "NumUnderPov","NumKidsBornNeverMar",
            "NumImmig","MedNumBR","HousVacant","MedYrHousBuilt", "OwnOccLowQuart","OwnOccMedVal",
            "OwnOccHiQuart", "OwnOccQrange","RentLowQ", "RentMedian", "RentHighQ",
            "RentQrange", "MedRent","NumInShelters","NumStreet")
dataset[, columns] <- lapply(columns, function(x) as.numeric(dataset[[x]]))
sapply(dataset,class)
# Removing the rest violent and non-violent crimes as potential dependent variables
violent <- subset(dataset, select=c(murders, murdPerPop, rapes, rapesPerPop, robberies, robbbPerPop,
                                    assaults, assaultPerPop, ViolentCrimesPerPop))
nonViolent <- subset(dataset, select=c(burglaries, burglPerPop, larcenies, larcPerPop, autoTheft,
autoTheftPerPop, arsons, arsonsPerPop,nonViolPerPop ))
dataset <- subset(dataset, select =-c(murders,murdPerPop,rapes,rapesPerPop, robberies,assaults, assaultPerPop,
                                      ViolentCrimesPerPop,burglaries, burglPerPop, larcenies, larcPerPop,
autoTheft, autoTheftPerPop,arsons, arsonsPerPop,nonViolPerPop ))
colSums(sapply(dataset, is.na))
#_____________________________________________________________________________________________________
#### DESCRIPTIVE ANALYSIS FOR CATEGORICAL VARIABLES ####

plot_usmap(data = datasetall, values = "robbbPerPop", color = "black",labels = TRUE ) +
  scale_fill_continuous(name = "Robberies per 100k (1995)",
                        low = "mistyrose",
                        high = "darkmagenta", labels = c("0", "250", "500", "750", "1000"),
                        breaks = c(0, 250, 500, 750, 1000)
  )+ labs(title = "Number of Robberies per 100k in 1995", caption="States in Grey were not found in the
dataset") +
  theme(plot.title = element_text(face = "bold",hjust = 0.5,size=12), legend.position = "right",
        plot.caption = element_text(color = "blue", face = "italic",hjust=0.5,size=10))

#_____________________________________________________________________________________________________
#### DESCRIPTIVE ANALYSIS FOR NUMERICS VARIABLES ####
```

```
numerics <- sapply(dataset, class) == "numeric"
numerics <- dataset[,numerics]
round(t(describe(numerics)),2)

# Summary Statistics for numeric variables
Hmisc::describe(numerics)
# Summary Statistics for robberies per 100k
summarytools::descr(numerics$robbbPerPop, transpose = TRUE)

# Barplot
violent <- violent[!is.na(violent$ViolentCrimesPerPop),]
colSums(sapply(violent, is.na))
round((sum(violent$murdPerPop,na.rm=TRUE)/sum(violent$ViolentCrimesPerPop,na.rm=TRUE)),2)*100
round((sum(violent$rapesPerPop,na.rm=TRUE)/sum(violent$ViolentCrimesPerPop,na.rm=TRUE)),2)*100
round((sum(violent$robbbPerPop,na.rm=TRUE)/sum(violent$ViolentCrimesPerPop,na.rm=TRUE)),2)*100
round((sum(violent$assaultPerPop,na.rm=TRUE)/sum(violent$ViolentCrimesPerPop,na.rm=TRUE)),2)*100

df <- data.frame(
  group = c("Murder", "Rape", "Robbery","Assault"),
  value = c(1, 6, 27,66)
)
head(df)

df %>%
  arrange(desc(value)) %>%
  mutate(prop = percent(value / sum(value))) -> df

pie <- ggplot(df, aes(x = "", y = value, fill = fct_inorder(group))) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start = 0) +
  scale_fill_brewer(palette="PuRd")+
  theme(axis.text.x=element_blank())+
  geom_label_repel(aes(label = prop), size=5, show.legend = F, nudge_x = 1) +
  guides(fill = guide_legend(title = "Crimes per 100k"))
pie

#---------------Plot 1---------------------

length1<-(seq(1,12))
par(mfrow=c(3,4));n <- nrow(numerics)
for (i in length1){
  hist(numerics[,i], main=names(numerics)[i], probability=TRUE, xlab=colnames(numerics[i]))
  lines(density(numerics[,i]),col = "deeppink", lwd=2 )
  index <- seq( min(numerics[,i]), max(numerics[,i]),
                length.out=100)
  ynorm <- dnorm( index, mean=mean(numerics[,i]),
                  sd(numerics[,i]) )
  lines( index, ynorm, col="blue", lty=3, lwd=2 )
}

par(mfrow=c(3,4))
for( i in length1){
  qqnorm(numerics[,i], main=paste('QQPlot of', colnames(numerics[i])),
```

```
        xlab=colnames(numerics[i]))
  qqline(numerics[,i], col='deeppink',lwd=2)
}

#---------------Plot 2---------------------

length2<-(seq(13,24))
par(mfrow=c(3,4));n <- nrow(numerics)
for (i in length2){
  hist(numerics[,i], main=names(numerics)[i], probability=TRUE, xlab=colnames(numerics[i]))
  lines(density(numerics[,i]),col = "deeppink", lwd=2 )
  index <- seq( min(numerics[,i]), max(numerics[,i]),
                length.out=100)
  ynorm <- dnorm( index, mean=mean(numerics[,i]),
                  sd(numerics[,i]) )
  lines( index, ynorm, col="blue", lty=3, lwd=2 )
}

par(mfrow=c(3,4))
for( i in length2){
  qqnorm(numerics[,i], main=paste('QQPlot of', colnames(numerics[i])),
         xlab=colnames(numerics[i]))
  qqline(numerics[,i], col='deeppink',lwd=2)
}

#---------------Plot 3---------------------

length3<-(seq(25,36))
par(mfrow=c(3,4));n <- nrow(numerics)
for (i in length3){
  hist(numerics[,i], main=names(numerics)[i], probability=TRUE, xlab=colnames(numerics[i]))
  lines(density(numerics[,i]),col = "deeppink", lwd=2 )
  index <- seq( min(numerics[,i]), max(numerics[,i]),
                length.out=100)
  ynorm <- dnorm( index, mean=mean(numerics[,i]),
                  sd(numerics[,i]) )
  lines( index, ynorm, col="blue", lty=3, lwd=2 )
}

par(mfrow=c(3,4))
for( i in length3){
  qqnorm(numerics[,i], main=paste('QQPlot of', colnames(numerics[i])),
         xlab=colnames(numerics[i]))
  qqline(numerics[,i], col='deeppink',lwd=2)
}

#---------------Plot 4---------------------

length4<-(seq(37,48))
par(mfrow=c(3,4));n <- nrow(numerics)
for (i in length4){
  hist(numerics[,i], main=names(numerics)[i], probability=TRUE, xlab=colnames(numerics[i]))
  lines(density(numerics[,i]),col = "deeppink", lwd=2 )
```

```
  index <- seq( min(numerics[,i]), max(numerics[,i]),
                length.out=100)
  ynorm <- dnorm( index, mean=mean(numerics[,i]),
                  sd(numerics[,i]) )
  lines( index, ynorm, col="blue", lty=3, lwd=2 )
}

par(mfrow=c(3,4))
for( i in length4){
  qqnorm(numerics[,i], main=paste('QQPlot of', colnames(numerics[i])),
         xlab=colnames(numerics[i]))
  qqline(numerics[,i], col='deeppink',lwd=2)
}

#---------------Plot 5---------------------

length5<-(seq(49,60))
par(mfrow=c(3,4));n <- nrow(numerics)
for (i in length5){
  hist(numerics[,i], main=names(numerics)[i], probability=TRUE, xlab=colnames(numerics[i]))
  lines(density(numerics[,i]),col = "deeppink", lwd=2 )
  index <- seq( min(numerics[,i]), max(numerics[,i]),
                length.out=100)
  ynorm <- dnorm( index, mean=mean(numerics[,i]),
                  sd(numerics[,i]) )
  lines( index, ynorm, col="blue", lty=3, lwd=2 )
}

par(mfrow=c(3,4))
for( i in length5){
  qqnorm(numerics[,i], main=paste('QQPlot of', colnames(numerics[i])),
         xlab=colnames(numerics[i]))
  qqline(numerics[,i], col='deeppink',lwd=2)
}

#---------------Plot 6---------------------

length6<-(seq(61,72))
par(mfrow=c(3,4));n <- nrow(numerics)
for (i in length6){
  hist(numerics[,i], main=names(numerics)[i], probability=TRUE, xlab=colnames(numerics[i]))
  lines(density(numerics[,i]),col = "deeppink", lwd=2 )
  index <- seq( min(numerics[,i]), max(numerics[,i]),
                length.out=100)
  ynorm <- dnorm( index, mean=mean(numerics[,i]),
                  sd(numerics[,i]) )
  lines( index, ynorm, col="blue", lty=3, lwd=2 )
}

par(mfrow=c(3,4))
for( i in length6){
  qqnorm(numerics[,i], main=paste('QQPlot of', colnames(numerics[i])),
         xlab=colnames(numerics[i]))
```

```
  qqline(numerics[,i], col='deeppink',lwd=2)
}

#---------------Plot 7---------------------

length7<-(seq(73,84))
par(mfrow=c(3,4));n <- nrow(numerics)
for (i in length7){
  hist(numerics[,i], main=names(numerics)[i], probability=TRUE, xlab=colnames(numerics[i]))
  lines(density(numerics[,i]),col = "deeppink", lwd=2 )
  index <- seq( min(numerics[,i]), max(numerics[,i]),
                length.out=100)
  ynorm <- dnorm( index, mean=mean(numerics[,i]),
                  sd(numerics[,i]) )
  lines( index, ynorm, col="blue", lty=3, lwd=2 )
}

par(mfrow=c(3,4))
for( i in length7){
  qqnorm(numerics[,i], main=paste('QQPlot of', colnames(numerics[i])),
         xlab=colnames(numerics[i]))
  qqline(numerics[,i], col='deeppink',lwd=2)
}

#---------------Plot 8---------------------

length8<-(seq(85,96))
par(mfrow=c(3,4));n <- nrow(numerics)
for (i in length8){
  hist(numerics[,i], main=names(numerics)[i], probability=TRUE, xlab=colnames(numerics[i]))
  lines(density(numerics[,i]),col = "deeppink", lwd=2 )
  index <- seq( min(numerics[,i]), max(numerics[,i]),
                length.out=100)
  ynorm <- dnorm( index, mean=mean(numerics[,i]),
                  sd(numerics[,i]) )
  lines( index, ynorm, col="blue", lty=3, lwd=2 )
}

par(mfrow=c(3,4))
for( i in length8){
  qqnorm(numerics[,i], main=paste('QQPlot of', colnames(numerics[i])),
         xlab=colnames(numerics[i]))
  qqline(numerics[,i], col='deeppink',lwd=2)
}

#---------------Plot 9---------------------

length9<-(seq(97,103))
par(mfrow=c(3,3));n <- nrow(numerics)
for (i in length9){
  hist(numerics[,i], main=names(numerics)[i], probability=TRUE, xlab=colnames(numerics[i]))
  lines(density(numerics[,i]),col = "deeppink", lwd=2 )
  index <- seq( min(numerics[,i]), max(numerics[,i]),
```

```
               length.out=100)
  ynorm <- dnorm( index, mean=mean(numerics[,i]),
                  sd(numerics[,i]) )
  lines( index, ynorm, col="blue", lty=3, lwd=2 )
}

par(mfrow=c(3,3))
for( i in length9){
  qqnorm(numerics[,i], main=paste('QQPlot of', colnames(numerics[i])),
         xlab=colnames(numerics[i]))
  qqline(numerics[,i], col='deeppink',lwd=2)
}
#_____________________________________________________________________________________________________________
#### OUTLIERS ####
outliers<-dataset[2:104]
for (i in 2:104){
  datatest<-dataset[,i]
  out <- boxplot( datatest, plot=FALSE )$out
  if(length(out)!=0){
    print('-------------------------------------------------------')
    print( paste('Outliers for variable', names(dataset)[i] ) )
    print( paste(length(out), 'outliers') )
    print( paste(round(100*length(out)/sum(!is.na(dataset)),1),
                 '% outliers', sep='' ) )
    print(which( datatest %in% out ))
  }
}
# No variable has outliers more than 0.1% of data
#_____________________________________________________________________________________________________________
#### NORMALITY TEST ####
# Shapiro-Wilk Test
lengthi<-(1:length(numerics));lengthi
count1<-0
for (i in lengthi){
  if (shapiro.test(numerics[,i])$p.value<0.05){
    print(paste('For variable', colnames(numerics[i]),'P-value',shapiro.test(numerics[,i])$p.value, 'is less
than a=0.05, thus the nuss hypothesis of normally distributed data is rejected'))}
  else{
    print(paste('For variable', colnames(numerics[i]),'P-value',shapiro.test(numerics[,i])$p.value, 'is greater
than a=0.05, thus the nuss hypothesis of normally distributed data is not rejected'))
    count1<-count1+1}
}
# Kolmogorov-Smirnov Test
lengthi<-(1:length(numerics));lengthi
count2<-0
for (i in lengthi){
    if (lillie.test(numerics[,i])$p.value<0.05){
    print(paste('For variable', colnames(numerics[i]),'P-value',lillie.test(numerics[,i])$p.value, 'is less than
a=0.05, thus the nuss hypothesis of normally distributed data is rejected'))}
  else{
    print(paste('For variable', colnames(numerics[i]),'P-value',lillie.test(numerics[,i])$p.value, 'is greater
than a=0.05, thus the nuss hypothesis of normally distributed data is not rejected'))
    count2<-count2+1}
```

```
}
# Symmetry Test
lengthi<-(1:length(numerics));length_i
count3 <- 0
for (i in length_i){
  if (symmetry.test(numerics[,i])$p.value<0.05){
    print(paste('In column', colnames(numerics[i]),'P-value',symmetry.test(numerics[,i])$p.value, 'is less than
a=0.05, thus I reject the null hypothesis that the distribution is assymmetric'))}
  else{
    print(paste('In column', colnames(numerics[i]),'P-value',symmetry.test(numerics[,i])$p.value, 'is greater
than a=0.05, thus I do not reject the null hypothesis that the distribution is assymmetric'))
    count3 <- count3 +1 }
}

#### Two sample Hypothesis testing (1 continous and 1 categorical variable) ####
# Question: Is the number of robberies per 100k equal in states California (CA) and Texas (TX)?
ca<-datasetall$robbbPerPop[which(datasetall$state=="CA")];length(ca)
tx<-datasetall$robbbPerPop[which(datasetall$state=="TX")];length(tx)
#Normality, n<50
shapiro.test(ca)
shapiro.test(tx)
par(mfrow=c(1,2))
hist(ca, main="Robberries per 100k of California state")
qqnorm(ca, main="Robberries per 100k of California state")
qqline(ca)
hist(tx, main="Robberries per 100k of Texas state")
qqnorm(tx, main="Robberries per 100k of Texas state")
qqline(tx)
symmetry.test(ca, boot=F)
symmetry.test(tx, boot=F)
wilcox.test(ca,tx)
par(mfrow=c(1,1))
boxplot(ca, tx,  names=c("CA","TX"), ylab='Robberies per 100k on two Sates', ylim=c(0,900))
#We do not accept H0: M1 = M2 => Significant difference is found about the
#median of robberies per 100k between California and Texas.

#_____________________________________________________________________________________________________
#### HIGH CORRELATED VARIABLES ####
# Is any variable correlated to robperpop??
# keep only attributes with correlation to price greater/less than 0.25/-0.25
names(numerics)
colSums(sapply(numerics, is.na))
correlations <- cor(numerics)
match("robbbPerPop",names(numerics))
cor <- correlations[,103] #robperpop
summary(cor)
cor<- data.frame(as.list(cor))
highcor<-cor[,colSums(cor > 0.25 | cor < -0.25) >= 1]
names(highcor)
highcor<-subset(numerics, select=c( racepctblack, racePctWhite, pctWFarmSelf, pctWInvInc, pctWPubAsst,
PctPopUnderPov, PctNotHSGrad, PctBSorMore, PctUnemployed, PctOccupManu, PctOccupMgmtProf, MalePctDivorce,
FemalePctDiv, TotalPctDiv, PctFam2Par, PctKids2Par,
PctYoungKids2Par,PctTeen2Par, NumKidsBornNeverMar, PctKidsBornNeverMar, PctImmigRec5,
```

```
PctImmigRec8,PctImmigRec10, PctRecImmig5, PctRecImmig8, PctRecImmig10, PctLargHouseFam, PersPerRentOccHous,
PctPersOwnOccup, PctPersDenseHous, PctHousOwnOcc, PctVacantBoarded, PctHousNoPhone, MedRentPctHousInc,
NumStreet, LemasPctOfficDrugUn, robbbPerPop))

#_____________________________________________________________________________________________________
#### PAIRWISE ASSOCIATIONS ####

#Correlation applying the Pearson's Correlation test: Identify Linear dependent variables
par(mfrow=c(1,1))
#1st group of high correlated variables, method: ellipse
corrplot(cor(highcor[1:15]), type = "upper", tl.pos = "td",
         method = "ellipse", tl.cex = 0.55, tl.col = 'black',
         order = "hclust", diag = T)
#2nd group of high correlated variables, method: ellipse
corrplot(cor(highcor[16:30]), type = "upper", tl.pos = "td",
         method = "ellipse", tl.cex = 0.55, tl.col = 'black',
         order = "hclust", diag = T)
#3rd group of high correlated variables, method: ellipse
corrplot(cor(highcor[31:37]), type = "upper", tl.pos = "td",
         method = "ellipse", tl.cex = 0.55, tl.col = 'black',
         order = "hclust", diag = T)

#_____________________________________________________________________________________________
#### LASSO ####

# Lars lasso
tolasso <-highcor
mfull <- lm(robbbPerPop~.,data=tolasso)
X<-model.matrix(mfull)[,-1]
lasso1 <- lars( X, tolasso$robbbPerPop )
plot(lasso1, xvar='n')
plot(lasso1, xvar='n', breaks=F)
plot(lasso1, xvar='n', breaks=F, xlim=c(0.5,1), ylim=c(-20,15) )
plot(lasso1, xvar='df')
plot(lasso1, xvar='arc')
plot(lasso1, xvar='step')
res.cv <- cv.lars( X, tolasso$robbbPerPop ) # default model='fraction'
lambda<-res.cv$index
cv    <-res.cv$cv
mincv.s <- lambda[cv==min(cv)]
coef( lasso1, s=mincv.s, mode='fraction' )

rescp<-summary(lasso1)
coef(lasso1, s=which.min(rescp$Cp), mode="step")
plot(lasso1, xvar='n', plottype='Cp')

#_____________________________________________________________________________________________
# Glmnet lasso
to_lasso<-highcor
colSums(sapply(to_lasso, is.na))
to_lasso[, 1:37] <- sapply(to_lasso[,1:37],as.numeric)
str(to_lasso)
mfull <- lm(robbbPerPop~.,data=to_lasso)
```

```
X <- model.matrix(mfull)[,-1]
lasso2 <- glmnet(X, to_lasso$robbbPerPop)
par(mfrow=c(1,1))
plot(lasso2, xvar = "lambda", label = T)
#Use cross validation to find a reasonable value for lambda
lasso2 <- cv.glmnet(X, to_lasso$robbbPerPop, alpha = 1)
lasso2$lambda
lasso2$lambda.min
lasso2$lambda.1se
plot(lasso2)
coef(lasso2, s = "lambda.min")
coef(lasso2, s = "lambda.1se")
plot(lasso2$glmnet.fit, xvar = "lambda")
abline(v=log(c(lasso2$lambda.min, lasso2$lambda.1se)), lty =2)

#_____________________________________________________________________________________________
#### FITTING THE REGRESSION MODEL ####

regressionmodel <- subset(highcor, select=c(racepctblack,racePctWhite,PctKidsBornNeverMar,robbbPerPop))
# Examine correlations
corrplot(cor(regressionmodel[1:4]), type = "upper", tl.pos = "td",
         method = "ellipse", tl.cex = 0.80, tl.col = 'black',
         order = "hclust", diag = T)
# Exploratory graphs for the selected attributes
eda.plots <- function(data, ask=F){
  graphics.off()
  numeric.only <- sapply(data,class)=='numeric'
  y <- data[,numeric.only]
  n<-ncol(y)
  for (i in 1:n){
    if (!ask) win.graph()
    par(mfrow=c(2,2), ask=ask)
    y1 <- y[,i]
    vioplot(y1, col="deeppink",main=names(y)[i])
    hist(y1, probability=TRUE, main=names(y)[i])
    lines(density(y1), col="deeppink",lwd=2)
    qqnorm(y1, main=names(y)[i])
    qqline(y1)
    boxplot(y1, main=names(y)[i], horizontal=TRUE, col="lightgrey")
  }
}
graphics.off()
eda.plots(regressionmodel, ask=T)

# Initial Model
initial_model <- lm(robbbPerPop~.,data=regressionmodel)
summary(initial_model)
par(mfrow=c(2,2))
plot(initial_model,ask=F)
spreadLevelPlot(initial_model)

round(vif(initial_model),2)
```

```
# Assessing Outliers
outlierTest(initial_model) # Bonferonni p-value for most extreme obs

# We can use a quantile comparison plots to compare the distribution of thestudentized residuals from our
regression model
# to thet-distribution. Observations that stray outside of the 95% confidence envelope are statistically
significant outliers
qqPlot(initial_model, main="QQ Plot", col.line="deeppink",simulate=T) #qq plot for studentized resid

# added variable plots
avPlots(initial_model, col.lines="deeppink") # leverage plots

#_____________________________________________________________________________________________________
#### TEST ASSUMPTIONS OF THE INITIAL FULL MODEL ####

# _____________________
# Global test of model assumptions
# _____________________
gvmodel <- gvlma(initial_model)
summary(gvmodel)

# _____________________
# Non Linearity
# _____________________
par(mfrow=c(1,1))
residualPlot(initial_model, type='rstudent', col.quad="deeppink")
residualPlots(initial_model, plot=F, type = "rstudent") #initial model passes
crPlots(initial_model)

ggscatter(regressionmodel, x = "robbbPerPop", y = "racepctblack",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "robbbPerPop", ylab = "racepctblack",
          add.params = list(color = "deeppink",
                            fill = "lightgray"))

ggscatter(regressionmodel, x = "robbbPerPop", y = "racePctWhite",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "robbbPerPop", ylab = "racePctWhite",
          add.params = list(color = "deeppink",
                            fill = "lightgray"))

ggscatter(regressionmodel, x = "robbbPerPop", y = "PctKidsBornNeverMar",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "robbbPerPop", ylab = "PctKidsBornNeverMar",
          add.params = list(color = "deeppink",
                            fill = "lightgray"))

# _____________________
# Independence
# _____________________
```

```
plot(rstudent(initial_model), type='l', col='deeppink', lwd=2.5)
library(randtests); runs.test(initial_model$res)
library(lmtest);dwtest(initial_model)
#D-W statistic is between the desirable margins (1.4-2.6) while the p-value is very high.
library(car); durbinWatsonTest(initial_model)  #initial model passes

# _____________________
# Equality of variances (Homoscedasticity)
# _____________________
yhat <- fitted(initial_model)
yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)), dig.lab=6)
table(yhat.quantiles)

leveneTest(rstudent(initial_model)~yhat.quantiles) # model does not pass

#Equlaity of variance is not met
boxplot(rstudent(initial_model)~yhat.quantiles)

# _____________________
# Normality Assumption
# _____________________

plot(initial_model, which = 2) #Normality of the residuals (step 1)
shapiro.test(initial_model$residuals) # model does not pass
lillie.test(initial_model$residuals) # model does not pass
ad.test(initial_model$residuals) # model does not pass
Stud.residuals <- rstudent(initial_model)
yhat <- fitted(initial_model)
par(mfrow=c(1,1))
plot(yhat, Stud.residuals)
abline(h=c(-2,2), col=2, lty=2)

#Summary of the final 4 plots
par(mfrow=c(2,2))
plot( initial_model, 2, col="deeppink")
plot( initial_model, 3)
residualPlot(initial_model, type='rstudent',col.quad="deeppink",lwd=2 )
plot(rstudent(initial_model), type='l', col="deeppink",lwd=2)

#_____________________________________________________________________________________________
#### STEPWISE REGRESSION ####

# Stepwise Model
aicmodel <- lm(robbbPerPop~.,data=regressionmodel)
summary(step(aicmodel), direction="both")
aicmodel <- lm(robbbPerPop~+racePctWhite+PctKidsBornNeverMar,data=regressionmodel)

round(vif(aicmodel),2)
summary(aicmodel)
par(mfrow=c(2,2))
plot(aicmodel,ask=F)
par(mfrow=c(1,1))
spreadLevelPlot(aicmodel)
```

```
# Assessing Outliers
outlierTest(aicmodel) # Bonferonni p-value for most extreme obs

# We can use a quantile comparison plots to compare the distribution of thestudentized residuals from our
regression model
# to thet-distribution. Observations that stray outside of the 95% confidence envelope are statistically
significant outliers
qqPlot(aicmodel, main="QQ Plot", col.line="deeppink",simulate=T) #qq plot for studentized resid

# added variable plots
avPlots(aicmodel, col.lines="deeppink") # leverage plots

#_____________________________________________________________________________________________
#### TEST ASSUMPTIONS OF THE STEPWISE FULL MODEL ####

# _____________________
# Global test of model assumptions
# _____________________
gvmodel <- gvlma(aicmodel)
summary(gvmodel)

# _____________________
# Non Linearity
# _____________________
par(mfrow=c(1,1))
residualPlot(aicmodel, type='rstudent', col.quad="deeppink")
residualPlots(aicmodel, plot=F, type = "rstudent") #initial model passes
crPlots(aicmodel)

# _____________________
# Independence
# _____________________
plot(rstudent(aicmodel), type='l', col='deeppink', lwd=2.5)
library(randtests); runs.test(aicmodel$res)
library(lmtest);dwtest(aicmodel)
#D-W statistic is between the desirable margins (1.4-2.6) while the p-value is very high.
library(car); durbinWatsonTest(aicmodel)  # model passes

# _____________________
# Equality of variances (Homoscedasticity)
# _____________________
yhat <- fitted(aicmodel)
yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)), dig.lab=6)
table(yhat.quantiles)

leveneTest(rstudent(aicmodel)~yhat.quantiles) # model does not pass

#Equlaity of variance is not met
boxplot(rstudent(aicmodel)~yhat.quantiles)

# _____________________
```

```
# Normality Assumption
# ____________________

plot(aicmodel, which = 2) #Normality of the residuals (step 1)
shapiro.test(aicmodel$residuals) # model does not pass
lillie.test(aicmodel$residuals) # model does not pass
ad.test(aicmodel$residuals) # model does not pass
Stud.residuals <- rstudent(aicmodel)
yhat <- fitted(aicmodel)
par(mfrow=c(1,1))
plot(yhat, Stud.residuals)
abline(h=c(-2,2), col=2, lty=2)

#Summary of the final 4 plots
par(mfrow=c(2,2))
plot( aicmodel, 2, col="deeppink")
plot( aicmodel, 3)
residualPlot(aicmodel, type='rstudent',col.quad="deeppink",lwd=2 )
plot(rstudent(aicmodel), type='l', col="deeppink",lwd=2)

#_____________________________________________________________________________________________
#### LOG TRANSFORMATION ####

# lOG Model
logmodel <- lm(log(robbbPerPop+1)~.,data=regressionmodel)
summary(step(logmodel), direction="both")
logmodel <- lm(log(robbbPerPop+1)~+racePctWhite+log(PctKidsBornNeverMar),data=regressionmodel)
summary(logmodel)

round(vif(logmodel),2)

par(mfrow=c(2,2))
plot(logmodel,ask=F)
par(mfrow=c(1,1))
spreadLevelPlot(logmodel)

# Assessing Outliers
outlierTest(logmodel) # Bonferonni p-value for most extreme obs

# We can use a quantile comparison plots to compare the distribution of thestudentized residuals from our
regression model
# to thet-distribution. Observations that stray outside of the 95% confidence envelope are statistically
significant outliers
qqPlot(logmodel, main="QQ Plot", col.line="deeppink",simulate=T) #qq plot for studentized resid

# added variable plots
avPlots(logmodel, col.lines="deeppink") # leverage plots

#_____________________________________________________________________________________________
#### TEST ASSUMPTIONS OF THE LOG MODEL ####

# ____________________
# Global test of model assumptions
```

```
# ____________________
gvmodel <- gvlma(logmodel)
summary(gvmodel)

# ____________________
# Non Linearity
# ____________________
par(mfrow=c(1,1))
residualPlot(logmodel, type='rstudent', col.quad="deeppink",ask=F)
residualPlots(logmodel, plot=F, type = "rstudent")
crPlots(logmodel)

# ____________________
# Independence
# ____________________

plot(rstudent(logmodel), type='l', col='deeppink', lwd=2.5)
library(randtests); runs.test(logmodel$res)
library(lmtest);dwtest(logmodel)
#D-W statistic is between the desirable margins (1.4-2.6) while the p-value is very high.
library(car); durbinWatsonTest(logmodel)  # model passes

# ____________________
# Equality of variances (Homoscedasticity)
# ____________________

yhat <- fitted(logmodel)
yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)), dig.lab=6)
table(yhat.quantiles)

leveneTest(rstudent(logmodel)~yhat.quantiles) # model passes

#Equlaity of variance is not met
boxplot(rstudent(logmodel)~yhat.quantiles)

# ____________________
# Normality Assumption
# ____________________

plot(logmodel, which = 2) #Normality of the residuals (step 1)
shapiro.test(logmodel$residuals) # model does not pass
lillie.test(logmodel$residuals) # model passes
ad.test(logmodel$residuals) # model passes
Stud.residuals <- rstudent(logmodel)
yhat <- fitted(logmodel)
par(mfrow=c(1,1))
plot(yhat, Stud.residuals)
abline(h=c(-2,2), col=2, lty=2)

#Summary of the final 4 plots
par(mfrow=c(2,2))
plot( logmodel, 2, col="deeppink")
plot( logmodel, 3)
```

```
residualPlot(logmodel, type='rstudent',col.quad="deeppink",lwd=2 )
plot(rstudent(logmodel), type='l', col="deeppink",lwd=2)

#_____________________________________________________________________________________________
#### POLYNOMIAL TRANSFORMATION ####

polymodel <- lm(log(robbbPerPop+1)~+poly(racePctWhite,3)+log(PctKidsBornNeverMar),data=regressionmodel)
summary(step(polymodel, direction='both'))
round(vif(polymodel),2)

par(mfrow=c(2,2))
plot(polymodel,ask=F)
par(mfrow=c(1,1))
spreadLevelPlot(polymodel)

# Assessing Outliers
outlierTest(polymodel) # Bonferonni p-value for most extreme obs

# We can use a quantile comparison plots to compare the distribution of thestudentized residuals from our
regression model
# to thet-distribution. Observations that stray outside of the 95% confidence envelope are statistically
significant outliers
qqPlot(polymodel, main="QQ Plot", col.line="deeppink",simulate=T) #qq plot for studentized resid

# added variable plots
avPlots(polymodel, col.lines="deeppink") # leverage plots

#_____________________________________________________________________________________________
#### TEST ASSUMPTIONS OF THE POLYNOMIAL MODEL ####

# _____________________
# Global test of model assumptions
# _____________________
gvmodel <- gvlma(polymodel)
summary(gvmodel)

# _____________________
# Non Linearity
# _____________________
par(mfrow=c(1,1))
residualPlot(polymodel, type='rstudent', col.quad="deeppink")
residualPlots(polymodel, plot=F, type = "rstudent")
crPlots(polymodel)

# _____________________
# Independence
# _____________________
plot(rstudent(polymodel), type='l', col='deeppink', lwd=2.5)
library(randtests); runs.test(polymodel$res)
library(lmtest);dwtest(polymodel)
#D-W statistic is between the desirable margins (1.4-2.6) while the p-value is very high.
library(car); durbinWatsonTest(polymodel)  # model passes
```

```
# ____________________
# Equality of variances (Homoscedasticity)
# ____________________
yhat <- fitted(polymodel)
yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)), dig.lab=6)
table(yhat.quantiles)

leveneTest(rstudent(polymodel)~yhat.quantiles) # model passes

#Equlaity of variance is not met
boxplot(rstudent(polymodel)~yhat.quantiles)

# ____________________
# Normality Assumption
# ____________________

plot(polymodel, which = 2) #Normality of the residuals (step 1)
shapiro.test(polymodel$residuals) # model passes
lillie.test(polymodel$residuals) # model passes
ad.test(polymodel$residuals) # model passes
Stud.residuals <- rstudent(polymodel)
yhat <- fitted(polymodel)
par(mfrow=c(1,1))
plot(yhat, Stud.residuals)
abline(h=c(-2,2), col=2, lty=2)

#Summary of the final 4 plots
par(mfrow=c(2,2))
plot( polymodel, 2, col="deeppink")
plot( polymodel, 3, col.line="deeppink")
residualPlot(polymodel, type='rstudent',col.quad="deeppink",lwd=2 )
plot(rstudent(polymodel), type='l', col="deeppink",lwd=2)

#_____________________________________________________________________________________________
#### BOX COX TRANSFORMATION ####

boxcox<-regressionmodel[(!regressionmodel$robbbPerPop==0),]
m <- lm(log(robbbPerPop+1)~+(racePctWhite)+(PctKidsBornNeverMar),data=regressionmodel)

# run the box-cox transformation
bc <- boxcox(log(robbbPerPop+1)~+poly(racePctWhite,3)+log(PctKidsBornNeverMar),data=boxcox)

(lambda <- bc$x[which.max(bc$y)])

powerTransform <- function(y, lambda1, lambda2 = NULL, method = "boxcox") {

  boxcoxTrans <- function(x, lam1, lam2 = NULL) {

    # if we set lambda2 to zero, it becomes the one parameter transformation
    lam2 <- ifelse(is.null(lam2), 0, lam2)

    if (lam1 == 0L) {
      log(y + lam2)
```

```
    } else {
      (((y + lam2)^lam1) - 1) / lam1
    }
  }

  switch(method
         , boxcox = boxcoxTrans(y, lambda1, lambda2)
         , tukey = y^lambda1
  )
}

boxcox_model <-
lm(powerTransform(log(robbbPerPop),lambda)~+poly(racePctWhite,3)+log(PctKidsBornNeverMar),data=boxcox)
summary(step(boxcox_model, direction='both'))
round(vif(boxcox_model),2)
par(mfrow=c(1,1))
spreadLevelPlot(boxcox_model)
qqnorm(boxcox_model$residuals); qqline(boxcox_model$residuals)
par(op)

#_____________________________________________________________________________________________
#### TEST ASSUMPTIONS OF THE BOX-COX MODEL ####

# _____________________
# Global test of model assumptions
# _____________________
gvmodel <- gvlma(boxcox_model)
summary(gvmodel)

# _____________________
# Non Linearity
# _____________________
par(mfrow=c(1,1))
residualPlot(boxcox_model, type='rstudent', col.quad="deeppink")
residualPlots(boxcox_model, plot=F, type = "rstudent")
crPlots(boxcox_model)

# _____________________
# Independence
# _____________________
plot(rstudent(boxcox_model), type='l', col='deeppink', lwd=2.5)
library(randtests); runs.test(boxcox_model$res)
library(lmtest);dwtest(boxcox_model)
#D-W statistic is between the desirable margins (1.4-2.6) while the p-value is very high.
library(car); durbinWatsonTest(boxcox_model)  # model passes

# _____________________
# Equality of variances (Homoscedasticity)
# _____________________
yhat <- fitted(boxcox_model)
yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)), dig.lab=6)
table(yhat.quantiles)
```

```
leveneTest(rstudent(boxcox_model)~yhat.quantiles) # model passes

#Equlaity of variance is not met
boxplot(rstudent(boxcox_model)~yhat.quantiles)

# _____________________
# Normality Assumption
# _____________________

plot(boxcox_model, which = 2) #Normality of the residuals (step 1)
shapiro.test(boxcox_model$residuals) # model passes
lillie.test(boxcox_model$residuals) # model passes
ad.test(boxcox_model$residuals) # model passes
Stud.residuals <- rstudent(boxcox_model)
yhat <- fitted(boxcox_model)
par(mfrow=c(1,1))
plot(yhat, Stud.residuals)
abline(h=c(-2,2), col=2, lty=2)

#Summary of the final 4 plots
par(mfrow=c(2,2))
plot( boxcox_model, 2, col="deeppink")
plot( boxcox_model, 3, col.line="deeppink")
residualPlot(boxcox_model, type='rstudent',col.quad="deeppink",lwd=2 )
plot(rstudent(boxcox_model), type='l', col="deeppink",lwd=2)

#_____________________________________________________________________________________________
#### COOK'S DISTANCE ####

sample_size<-nrow(regressionmodel)
mod <-  lm(log(robbbPerPop+1)~+poly(racePctWhite,3)+log(PctKidsBornNeverMar),data=regressionmodel)
cooksd <- cooks.distance(mod)
plot(cooksd, pch="*", cex=2, main="Influential Obs by Cooks distance")  # plot cook's distance
abline(h = 4*mean(cooksd, na.rm=T), col="red")  # add cutoff line
text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>4/sample_size, names(cooksd),""), col="red")  # add
labels

influential <- as.numeric(names(cooksd)[(cooksd > (4/sample_size))])
cooks <- regressionmodel[-influential, ]

#_____________________________________________________________________________-
#### MODEL AFTER APPLYING COOKS DISTANCE ####

model <- lm(powerTransform(log(robbbPerPop),lambda)~+poly(racePctWhite,3)+log(PctKidsBornNeverMar),data=cooks)
summary(step(model,direction="both"))
round(vif(model),2)
par(mfrow=c(1,1))
spreadLevelPlot(model)

#_____________________________________________________________________________________________
#### TEST ASSUMPTIONS AFTER COOKS DISTANCE ####

# _____________________
```

```
# Global test of model assumptions
# ____________________
gvmodel <- gvlma(model)
summary(gvmodel)

# ____________________
# Non Linearity
# ____________________
par(mfrow=c(1,1))
residualPlot(model, type='rstudent', col.quad="deeppink")
residualPlots(model, plot=F, type = "rstudent")
crPlots(model)

# ____________________
# Independence
# ____________________
plot(rstudent(model), type='l', col='deeppink', lwd=2.5)
library(randtests); runs.test(model$res)
library(lmtest);dwtest(model)
#D-W statistic is between the desirable margins (1.4-2.6) while the p-value is very high.
library(car); durbinWatsonTest(model)  # model passes

# ____________________
# Equality of variances (Homoscedasticity)
# ____________________
yhat <- fitted(model)
yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)), dig.lab=6)
table(yhat.quantiles)

leveneTest(rstudent(model)~yhat.quantiles) # model passes

#Equlaity of variance is not met
boxplot(rstudent(model)~yhat.quantiles)

# ____________________
# Normality Assumption
# ____________________

plot(model, which = 2) #Normality of the residuals (step 1)
shapiro.test(model$residuals) # model does not pass
lillie.test(model$residuals) # model passes
ad.test(model$residuals) # model passes
Stud.residuals <- rstudent(model)
yhat <- fitted(model)
par(mfrow=c(1,1))
plot(yhat, Stud.residuals)
abline(h=c(-2,2), col=2, lty=2)

#Summary of the final 4 plots
par(mfrow=c(2,2))
plot( model, 2, col="deeppink")
plot( model, 3)
residualPlot(model, type='rstudent',col.quad="deeppink",lwd=2 )
```

```
plot(rstudent(model), type='l', col="deeppink",lwd=2)

#_____________________________________________________________________________________________
#### FINAL MODELS FOR 10 FOLD CROSS VALIDATION ####


# LOG MODEL
logmodel <- lm(log(robbbPerPop)~+racePctWhite+log(PctKidsBornNeverMar),data=cooks)

log_cross<-train(log(robbbPerPop)~+racePctWhite+PctKidsBornNeverMar,data=cooks, na.action=na.exclude,
method="lm",
                    trControl = trainControl(method = "cv", number = 10,verboseIter = TRUE))
print(log_cross)

# POLYNOMIAL MODEL AFTER COOK'S DISTANCE
poly_model <- lm(log(robbbPerPop)~+poly(racePctWhite,3)+log(PctKidsBornNeverMar),data=cooks)
model_cross<-train(log(robbbPerPop)~+poly(racePctWhite,3)+log(PctKidsBornNeverMar),data=cooks,
na.action=na.exclude, method="lm",
                trControl = trainControl(method = "cv", number = 10,verboseIter = TRUE))
print(model_cross)

# BOX-COX MODEL
boxcox_model <-
lm(powerTransform(log(robbbPerPop),lambda)~+poly(racePctWhite,3)+log(PctKidsBornNeverMar),data=boxcox)
box_cross<-train(log(robbbPerPop)~+poly(racePctWhite,3)+log(PctKidsBornNeverMar),data=cooks,
na.action=na.exclude, method="lm",
                  trControl = trainControl(method = "cv", number = 10,verboseIter = TRUE))
print(box_cross)

#_____________________________________________________________________________________________
#### OUT OF SAMPLE PREDICTIVE ABILITY  OF THE MODELS ####

test<-read.table("crime_test.dat", sep=",", header=F)
test <- setNames(test,
c('communityname','state','countyCode','communityCode','fold','population','householdsize',
                        'racepctblack','racePctWhite','racePctAsian','racePctHisp','agePct12t21','agePct12t29',
'agePct16t24','agePct65up','numbUrban','pctUrban','medIncome','pctWWage','pctWFarmSelf','pctWInvInc','pctWSocSec
','pctWPubAsst','pctWRetire','medFamInc','perCapInc','whitePerCap','blackPerCap','indianPerCap','AsianPerCap','O
therPerCap','HispPerCap','NumUnderPov','PctPopUnderPov','PctLess9thGrade','PctNotHSGrad','PctBSorMore','PctUnemp
loyed', 'PctEmploy','PctEmplManu','PctEmplProfServ','PctOccupManu','PctOccupMgmtProf',
'MalePctDivorce','MalePctNevMarr','FemalePctDiv','TotalPctDiv','PersPerFam','PctFam2Par','PctKids2Par','PctYoung
Kids2Par','PctTeen2Par','PctWorkMomYoungKids','PctWorkMom','NumKidsBornNeverMar','PctKidsBornNeverMar','NumImmig
','PctImmigRecent','PctImmigRec5','PctImmigRec8','PctImmigRec10','PctRecentImmig','PctRecImmig5','PctRecImmig8',
'PctRecImmig10','PctSpeakEnglOnly','PctNotSpeakEnglWell','PctLargHouseFam','PctLargHouseOccup',
'PersPerOccupHous','PersPerOwnOccHous','PersPerRentOccHous','PctPersOwnOccup','PctPersDenseHous',
'PctHousLess3BR','MedNumBR','HousVacant','PctHousOccup','PctHousOwnOcc','PctVacantBoarded','PctVacMore6Mos','Med
YrHousBuilt','PctHousNoPhone','PctWOFullPlumb','OwnOccLowQuart','OwnOccMedVal','OwnOccHiQuart','OwnOccQrange','R
entLowQ','RentMedian','RentHighQ','RentQrange','MedRent','MedRentPctHousInc','MedOwnCostPctInc','MedOwnCostPctIn
cNoMtg','NumInShelters','NumStreet','PctForeignBorn','PctBornSameState','PctSameHouse85','PctSameCity85','PctSam
eState85','LemasSwornFT','LemasSwFTPerPop','LemasSwFTFieldOps','LemasSwFTFieldPerPop','LemasTotalReq','LemasTotR
eqPerPop','PolicReqPerOffic','PolicPerPop','RacialMatchCommPol','PctPolicWhite','PctPolicBlack','PctPolicHisp',
'PctPolicAsian','PctPolicMinor','OfficAssgnDrugUnits','NumKindsDrugsSeiz','PolicAveOTWorked','LandArea','PopDens
','PctUsePubTrans','PolicCars','PolicOperBudg','LemasPctPolicOnPatr','LemasGangUnitDeploy','LemasPctOfficDrugUn'
```

```
,'PolicBudgPerPop','murders','murdPerPop','rapes','rapesPerPop','robberies','robbbPerPop','assaults','assaultPer
Pop','burglaries','burglPerPop','larcenies','larcPerPop','autoTheft','autoTheftPerPop','arsons','arsonsPerPop','
ViolentCrimesPerPop','nonViolPerPop'))
test<-subset(test,select=c(racePctWhite,PctKidsBornNeverMar,robbbPerPop ))
test$robbbPerPop<-as.numeric(as.factor(test$robbbPerPop))
test[,1:3] <-sapply(test[,1:3],as.numeric)
str(test)

# LOG MODEL
train.control<-trainControl(method="cv", number=10)
log_test<-train(log(robbbPerPop)~+racePctWhite+log(PctKidsBornNeverMar)
                    ,data=test, na.action=na.exclude, method="lm",
                    trControl = trainControl(method = "cv", number = 10,verboseIter = TRUE))
print(log_test)

# POLYNOMIAL MODEL
train.control<-trainControl(method="cv", number=10)
poly_test<-train(log(robbbPerPop)~+poly(racePctWhite,3)+log(PctKidsBornNeverMar)
                ,data=test, na.action=na.exclude, method="lm",
                trControl = trainControl(method = "cv", number = 10,verboseIter = TRUE))
print(poly_test)

# BOX-COX MODEL
test$robbbPerPop <- powerTransform(log(test$robbbPerPop), lambda)
train.control<-trainControl(method="cv", number=10)
boxcox_test<-train(robbbPerPop~+poly(racePctWhite,3)+log(PctKidsBornNeverMar)
                   ,data=test, na.action=na.exclude, method="lm",
                   trControl = trainControl(method = "cv", number = 10,verboseIter = TRUE))
print(boxcox_test)

#_____________________________________________________________________________-
#### CENTER THE MEDIANS ####

#Center the means
centered_model<-as.data.frame(scale(cooks, center=T, scale=F))
centered_model$robbbPerPop<-cooks$robbbPerPop
sapply(centered_model,median)
sapply(centered_model,sd)
round(sapply(centered_model,median),5)
round(sapply(centered_model,median),2)
#centered_model<-centered_model[(centered_model$PctKidsBornNeverMar>0),]
model2<-lm(log(robbbPerPop+1)~+racePctWhite+PctKidsBornNeverMar, data=centered_model)
summary(step(model2, direction='both'))

confint(model2)

cooks$racePctWhite[median(cooks$robbbPerPop)] #70.5
cooks$PctKidsBornNeverMar[median(cooks$PctKidsBornNeverMar)] #3.43

sum(exp(logmodel$fitted.values)) #overall estimate of robberies

#_____________________________________________________________________________-
#### FURTHER ANALYSIS - ROBUST REGRESSION ####
```

```
library(MASS)
rlmmodel <- rlm(log(robbbPerPop)~+(racePctWhite)+log(PctKidsBornNeverMar),
                data=cooks, psi = psi.bisquare)  # robust reg model
summary(rlmmodel)

library(DMwR)
round(DMwR::regr.eval(cooks$robbbPerPop, rlmmodel$fitted.values),3)

library(robustbase)
library(robust)
robust_model<-lmrob((log(robbbPerPop))~+(racePctWhite)+log(PctKidsBornNeverMar),
                    data=cooks, method='MM', fast.s.large.n = Inf, cov = ".vcov.w" )
summary(robust_model, setting = "KS2014")
round(vif(robust_model),1)

##################################### END OF CODE #####################################
```