How to be taken by the UCLA master's program catalog?

Wenxuan Li

Abstract

Our data set is about students' relevant grades and their probability of being selected by the UCLA master's program catalog. We have two goals.

1. The first is to find the relationship between each variable in the student information and the probability of admission, and find out which variable has the most influence on the admission rate.
2. The second is to build as perfect a model as possible to help students assess the probability of admission.

In the modeling process, we used **stepwise regression**, **all-subsets regression**, **box-cox transformation** and other methods to help us complete the establishment of the model. At the same time, we also achieved two goals.

1. The model found the relationship between the remaining 6 variables (deleted SOP) and the admission rate. Among them, "CGPA" contributes the most to R-squared.
2. R^2 =0.8558 and R^2_{adj}= 0.854. We think this model can predict relatively accurate results

From the perspective of the model, the student's scores (GRE_Score, TOEFL_Score and CGPA) are the key to whether they can be taken into the UCLA master's program catalog. Purpose statement, letter of recommendation, and scientific research experience are just extra points, not necessary items.

Introduction

With the development of technology and economy, the world is becoming more prosperous. At the same time, in order to ensure the sustainability of development, education has become a top priority. While emphasizing education on a global scale, the status of a large number of institutions of higher learning has gradually increased. The data set of our project is a downloaded from the Kaggle website, which is truly

from UCLA University. The following will introduce the basic information of this data set and the goals we want to achieve through this project.

1. The meaning of this data set: Information about students applying for the UCLA master's program and the proabability of each being admitted.
2. Data set size: 8 columns (7 regressors and 1 response) * 500 rows (n=500).
3. Variable interpretation: All variables are the higher the better. See Table 1.

Table 1 Variable interpretation

Variable name	Range	Definition
GRE Score	0-340	Graduate Record Examination
TOEFL Score	0-120	An academic English language test
University Rating	1-5	American University Rankings
SOP	0-5	The Statement of purpose about why students coming to UCLA
LOR:	0-5	Letter of Recommendation Strength
CGPA	0-10	GPA during undergraduate course
Research Experience	0 or 1	Have scientific research experience or not

4. The reason and purpose of choosing this data set
    a) Find the relation between the chance of admission and seven variables.
    b) Predict the probability of being taken by the UCLA University Master’s Program Catalog.
    c) Advice for undergraduates who want to go to UCLA.

Methodology

In order to make the established multiple regression model as perfect as possible, we used many R built-in functions and custom functions in the project. In addition, we also standardized the modeling process to make the logic clearer.

1. Throughout the modeling process, we continue to repeat the four main steps until we find a better linear regression model. They are fitted model, diagnostic model, conversion model and test model. The flowchart is shown in Figure 1.

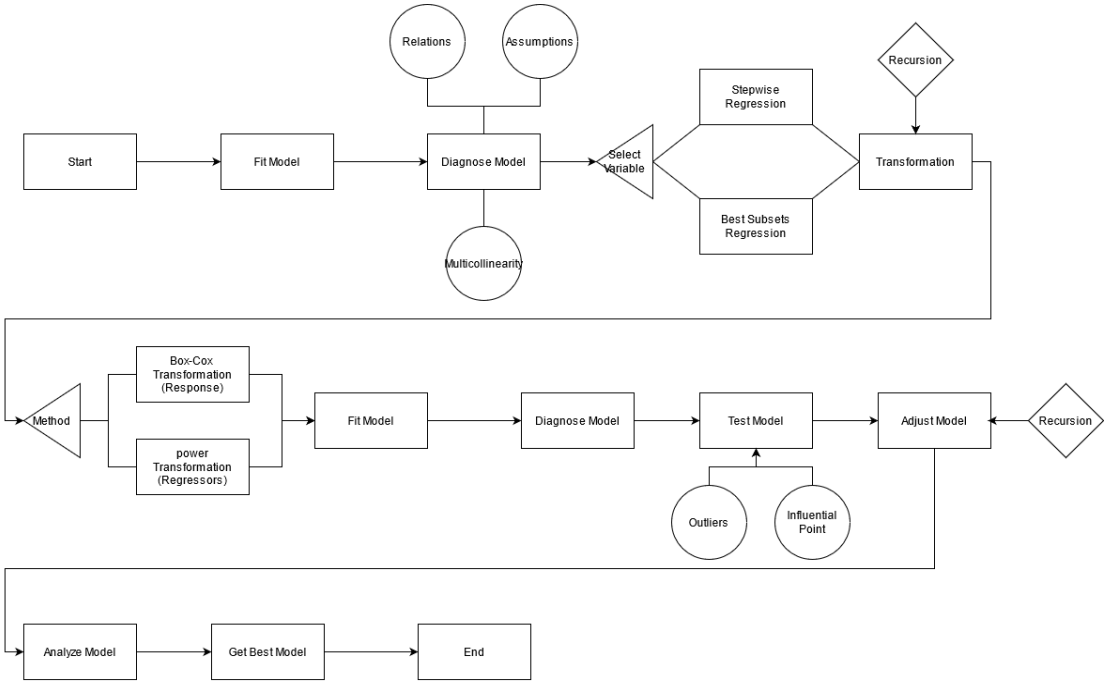

Figure 1 Modeling flowchart

2. We have used many functions and methods in various stages. Table 2 shows the functions and methods we used in each stage.

Table 2 Functions and methods used in each stage

Stage	Methods and functions
Data visualization	Calculate correlation between variables
	Correlation matrix between variables
Modeling	lm() function to establish multiple regression model
Model diagnosis	crPlots() –Test linearity
	qqPlot() -Test normality
	durbinWatsonTest() – Test residuals' independence
	ncvTest() – Test homoscedasticity
	vif() – Test multicollinearity
Outlier Test	outlierTest() – Test outlier

	hat.plot() – Test High leverage point
	Judge high influence points by cook distance
Variable selection	Stepwise regression
	All-subsets regression
Transformation	powerTransform() - Transform the regressor
	Box-Cox Transformation – Transform the response
Get the statistics	summary() - Obtain the estimated regression coefficient, R-squared, adjusted R-squared and other data of the model
	glance() - Get more statistics
	summary(regsubsets()) - Get statistics such as cp value
	PRESS() - Get the PRESS value of the model
Fit existing data	fitted() - Fitting the data used for modeling
Predict new data	predict() - Use the established model to make predictions on new data
Relative weight	calc.relimp() - Calculate the contribution of each variable to R-squared

Data analysis

1. Install packages

Because many built-in functions of the R language are used throughout the modeling process, we must first install the corresponding package. The packages we used are shown in Table 3.

Table 3 Packages used in R code

Packages' Name	
carData	boot
car	relaimpo
MASS	ISLR
olsrr	leaps
qpcR	broom
moments	robustbase

2. Data visualization

Before processing the data, we first observed the data and established a correlation matrix. as shown in figure 2.

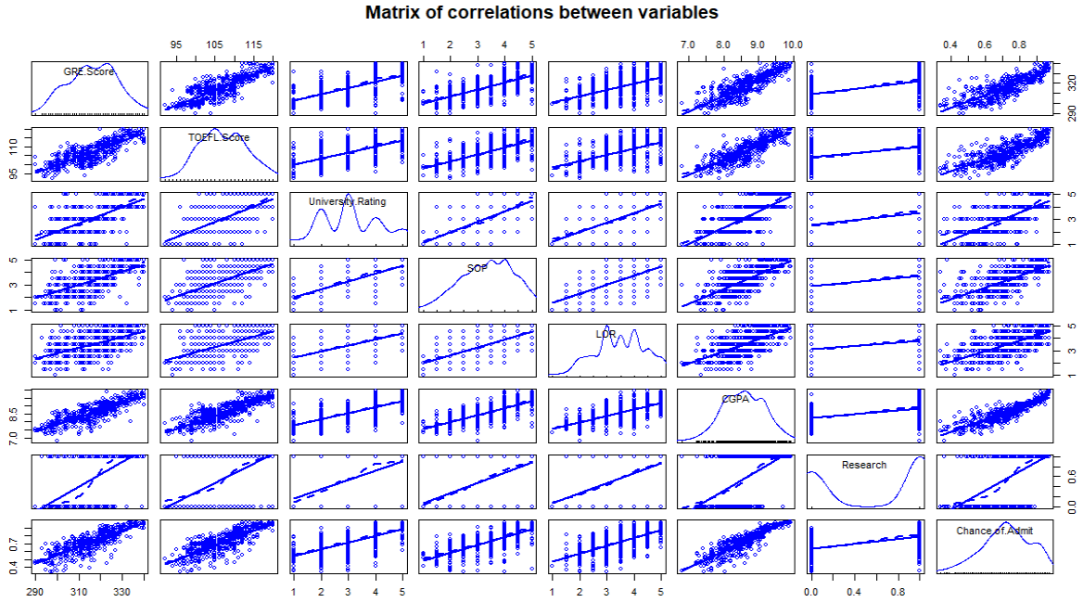

Figure 2 Correlation matrix

The diagonal graphs are the distribution graphs of the various variables. From these graphs, we can see that the variables other than the "Research" variable can be said to be normally distributed. The "Research" variable is regarded as a categorical variable because it only contains 0 or 1. Each regressor has a linear relationship with the shutdown of the response. In addition, we also calculated the correlation between variables. The data is shown in Table 4.

Table 4 Correlation between variables

	GRE_Score	TOEFL_Score	University.Rating	SOP	LOR	CGPA	Research	Chance.of.Admit
GRE_Score	1.0000	0.8272	0.6353	0.6134	0.5246	0.8258	0.5633	0.8103
TOEFL_Score	0.8272	1.0000	0.6497	0.6444	0.5415	0.8105	0.4670	0.7922
University.Rating	0.6353	0.6497	1.0000	0.7280	0.6086	0.7052	0.4270	0.6901
SOP	0.6134	0.6444	0.7280	1.0000	0.6637	0.7121	0.4081	0.6841
LOR	0.5246	0.5415	0.6086	0.6637	1.0000	0.6374	0.3725	0.6453
CGPA	0.8258	0.8105	0.7052	0.7121	0.6374	1.0000	0.5013	0.8824
Research	0.5633	0.4670	0.4270	0.4081	0.3725	0.5013	1.0000	0.5458
Chance.of.Admit	0.8103	0.7922	0.6901	0.6841	0.6453	0.8824	0.5458	1.0000

Through the last row or the last column of the table, we can see the correlation coefficient between each regressors and response. In addition, through this table we can initially see the correlation between the various regressors.

Before modeling, we renamed the variables. As shown in Table 5

Table 5 Change of variable name

Change of variable name	
Before	After
GRE_Score	X1
TOEFL_Score	X2
University.Rating	X3
SOP	X4
LOR	X5
CGPA	X6
Research	Z1
Chance.of.Admit	Y

3. Fitting – Preliminary model

Because what we want to build is a multiple regression model, the matrix form formula of the model is:

$$Y = X\beta + \epsilon \tag{1}$$

The formula for solving β is:

$$\beta = (X^T X)^{-1} X^T Y \tag{2}$$

In R, we use the lm() function to directly build a multiple regression model. And through the summary() function, we get the basic data about the preliminary model. The data is shown in Figure 3.

```
> #Build a regression model on the own data
> model1 <- lm(y~x1+x2+x3+x4+x5+x6+z1,data = data)
> summary(model1)

Call:
lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + z1, data = data)

Residuals:
      Min        1Q    Median        3Q       Max
-0.266657 -0.023327  0.009191  0.033714  0.156818

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.2757251  0.1042962 -12.232  < 2e-16 ***
x1           0.0018585  0.0005023   3.700 0.000240 ***
x2           0.0027780  0.0008724   3.184 0.001544 **
x3           0.0059414  0.0038019   1.563 0.118753
x4           0.0015861  0.0045627   0.348 0.728263
x5           0.0168587  0.0041379   4.074 5.38e-05 ***
x6           0.1183851  0.0097051  12.198  < 2e-16 ***
z11          0.0243075  0.0066057   3.680 0.000259 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.05999 on 492 degrees of freedom
Multiple R-squared:  0.8219,    Adjusted R-squared:  0.8194
F-statistic: 324.4 on 7 and 492 DF,  p-value: < 2.2e-16
```

Figure 3 Fitting results of the preliminary model

By analyzing the data in Figure 3, mine came to the following conclusions:

a) Under the condition that other regressors remain unchanged, the two variables x3 and x4 are not linearly related to y.

b) R^2 =0.8219 and R^2_{adj} =0.8194. which indicates all regressors can explain 82.19% of response variance.

Therefore, our model still has some problems. We will further observe the deficiencies of the model through model diagnosis.

4. Model diagnosis

In model diagnosis, we need to test the four assumptions of the multiple regression model, the multicollinearity between variables,.

a) Four assumptions

i. Linearity:

Through the crplot() function in the R language, we can get the component residual plot of each variable. As shown in Figure 4.

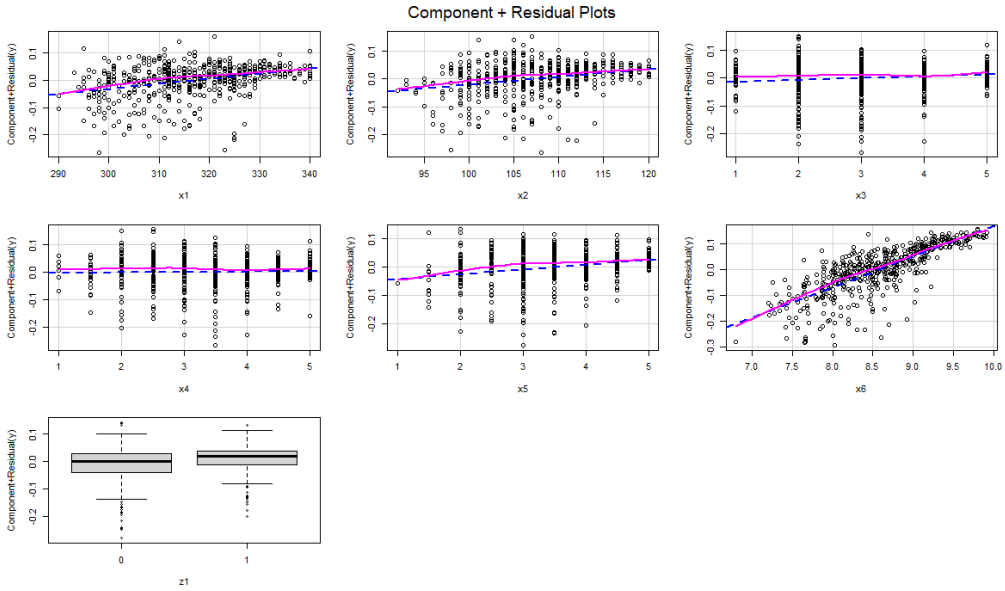

Figure 4 Component residual plot for each variable

From Figure 4 we can see that, except for the categorical variable z1, the distribution of every other variable has a linear relationship with the line in the figure. Although the values of x3, x4, and x5 are relatively discrete, since the distribution of these three variables is normally distributed, we believe that the relationship between each regressors and response is linear.

ii. Normality

We use the qqplot() function and the residplot() function we defined to visualize the residual distribution. As shown in Figure 5.

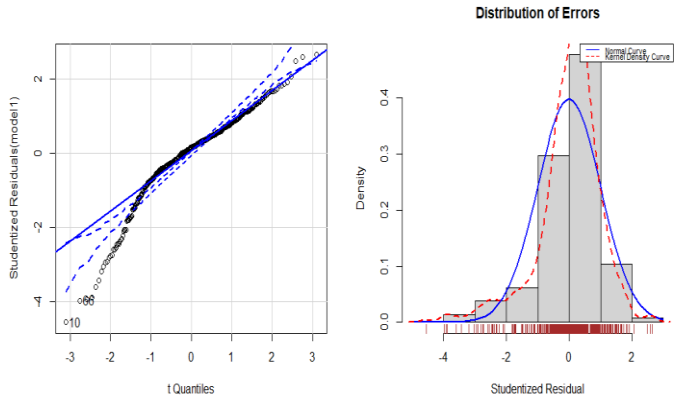

Figure 5 Residual visualization image

In the Q-Q plot on the left, we can see that there are many points that deviate from the 95% confidence interval at the left end. In the histogram on the right, the blue line is the standard normal distribution curve and the red line is the distribution curve corresponding to the preliminary model. Both figures show that the residual distribution of the preliminary model is different from the normal distribution, so the normality test of the preliminary model failed

iii. Independence of residuals

The car package in R provides a function that can be used for the Durbin-Watson test, which can be used to detect the sequence correlation of the residuals. Test as shown in Figure 6

```
> #Test Independence of residuals
> durbinWatsonTest(model1)
 lag Autocorrelation D-W Statistic p-value
   1       0.6016797      0.795901       0
 Alternative hypothesis: rho != 0
```

Figure 6 Test independence of residuals

In general, if the P value in the Durbin-Watson test is between 1 and 3, we say that the residuals have no autocorrelation, that is, independent. But the p value of the preliminary model in this test is 0, so the residuals of the preliminary model are not independent

iv. Homoscedasticity

We can use the ncvTest() function in R to generate a hypothesis test, where the H_0 is that the variance of the residuals does not change, and the H_a is that the variance of the residuals changes with the level of the fitted value. The test result is shown in Figure 7. The P-value = 4.2697e-13<<0.05, so we reject the H_0 that the variance of the residuals is changing.

```
> #Test Homoscedasticity
> par(mfrow = c(1,1))
> ncvTest(model1)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 52.51501, Df = 1, p = 4.2697e-13
```

Figure 7 Test homoscedasticity

In addition, the spreadLevelPlot() function can be used to create a scatter plot with the best fit curve, which can show the relationship between the absolute value of the standardized residual and the fitted value. See figure 8.

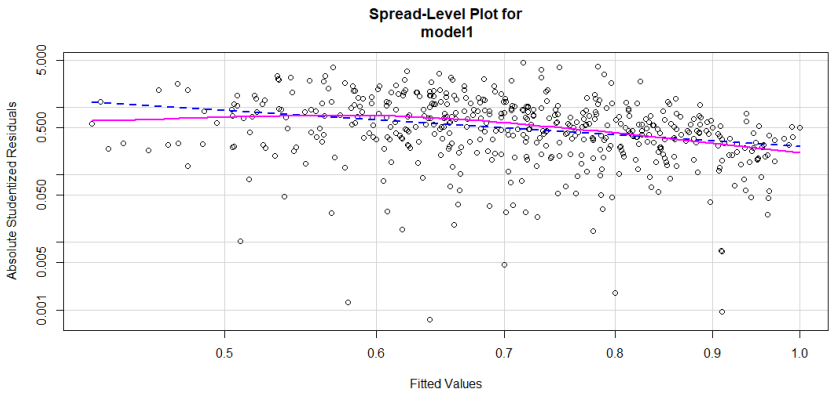

Figure 8 Standardized residual absolute value and fitted value

b) Multicollinearity test

The multicollinearity test between variables must be done before making variable selection. In the multiple regression model, the linear relationship between variables should be avoided as much as possible to reduce the number of variables. The vif value of each variable in the model can be obtained through the vif() function. Specific data are shown in Table 6.

Table 6 VIF-value of each variables

VIF-value of variables							
Variable	X1	X2	X3	X4	X5	X6	Z1
VIF-value	4.4642	3.9042	2.6210	2.8352	2.0335	4.7779	1.4940

The vif value of each variable in the preliminary model is not greater than 10, so we believe that there is no multicollinearity problem in the preliminary model.

5. Abnormal observations

Abnormal observations are generally divided into three categories: outliers, high leverage points, and high influence points. We will test these three abnormal observations separately

a) Outliers:

Outliers are observation points where the model predicts poorly. In the project, we regard the points that are outside the average value ± standard deviations from the predicted value as outliers, and then combine the outlierTest() function to pick out the more significant outliers. The outlier data is shown in Table 7.

Table 7 Outliers' data

Method	Outliers
± 3 standard deviations	10, 11, 41, 60, 65, 66, 67, 93
outlierTest()	10, 66, 93

b) High leverage points:

Observations with high leverage are outliers related to other predictors. Generally, the hat statistics are used to judge whether a point is a high leverage value. We stipulate that if the hat value of the observation point is greater than 3 times the average value of the hat, then the observation point is considered to be a point with a high leverage value. As shown in Figure 9.

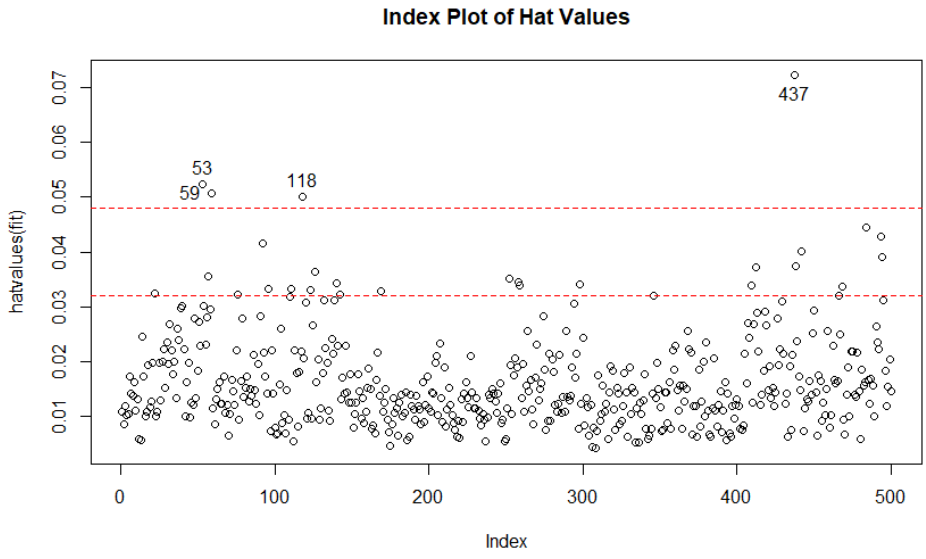

Figure 10 Hat values of preliminary model

c) High influence point

The strong influence point refers to the point where the influence on the estimated value of the model parameter is somewhat out of balance. We use the method of calculating the cook distance to determine whether it is a high-strength influence point.

If the cook distance of a point is greater than 4/(n-k-1), it indicates that the point is a high impact point. The strong influence points of the preliminary model are shown in Figure 11.

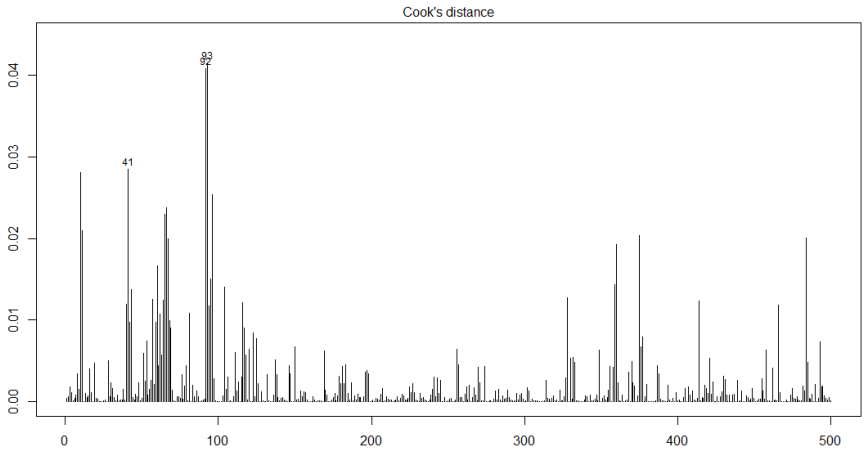

Figure 11 Observations' cook distance

We will not deal with abnormal observations in this part. Because some abnormal observations will not be dealt with due to variable selection and transformation, we deal with abnormal observations before building the final model.

6. Variable selection

The first step in our optimization of the model is variable selection. We need to determine the variables in the model before performing the transformation to avoid being affected by irrelevant variables during the transformation. In order to ensure the accuracy of variable selection, we perform both stepwise regression and all-subsets regression at the same time to compare the results.

a) Stepwise regression:

There are three sub-categories of stepwise regression: forward selection, backward selection and stepwise regression. In the project, we chose to use stepwise regression to make the results as accurate as possible. We use the step() function in R and specify direction="both" to use stepwise regression. The result is shown in Figure 12.

```
Step:  AIC=-2807.59
y ~ x1 + x2 + x3 + x5 + x6 + z1

       Df Sum of Sq    RSS     AIC
<none>               1.7708 -2807.6
- x3    1   0.01190 1.7827 -2806.2
+ x4    1   0.00043 1.7704 -2805.7
- x2    1   0.03760 1.8084 -2799.1
- z1    1   0.04893 1.8197 -2796.0
- x1    1   0.04901 1.8198 -2795.9
- x5    1   0.06892 1.8397 -2790.5
- x6    1   0.55954 2.3304 -2672.3
```

Figure 12 The result of stepwise regression

The step() function will return the model with the smallest AIC value during the test. From Figure 12, we can see that the number of variables has changed from 7 to 6, and the x4 variable (SOP) has been deleted. This is the result of variable selection using stepwise regression

b) All-subsets regression:

Stepwise regression can find the best model in the testing process, but sometimes it cannot guarantee that it has tested all possible outcomes of the current variable. In order to solve this limitation, we use all-subsets regression for verification.

First, we visualize the results of the all-subsets regression of regsubsets(). As shown in Figure 13. In the figure we can see four statistics: RSS, adjusted R^2_{adj}, Cp-value and BIC changes with the number of variables.

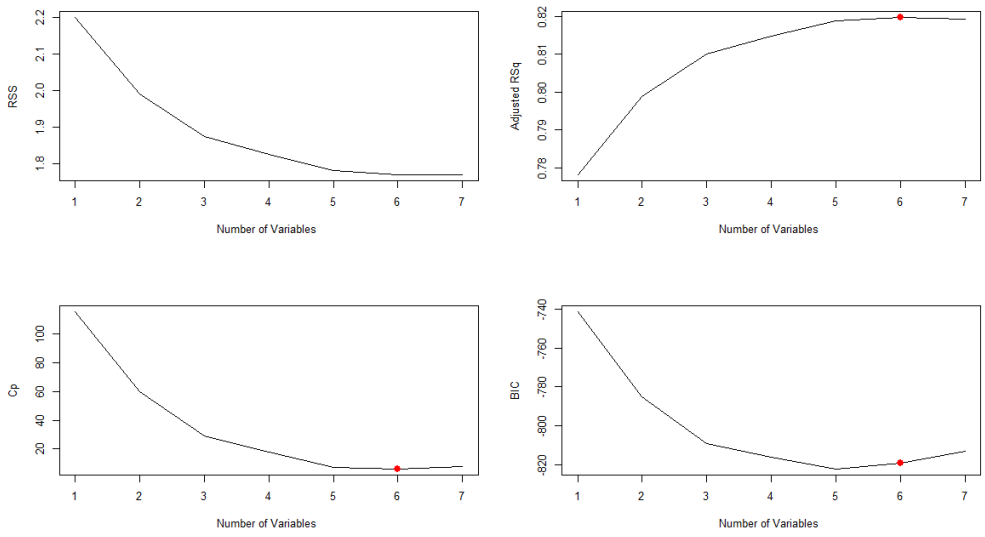

Figure 13 The changes of the four statistics with the number of variables

It can be seen from the image that when the number of variables is 6, each statistic reaches the optimal situation. This also preliminarily verified the results of the stepwise regression.

Next, we get all possible results of variable selection through the ols_step_all_possible() function. As shown in Table 8. According to the judgment condition: larger R^2, larger R^2_{adj} and smaller Cp, the variables of the model are x1, x2, x3, x5, x6, z1, and x4 is deleted.

Table 8 Results of allsubsets regression

Number of variable	Predictors	R^2	R^2_{adj}	Cp
1	X6	0.7782	0.7781	115.4750
1	X1	0.6567	0.6560	452.4558
1	X2	0.6276	0.6269	532.6887
2	X1, X6	0.7996	0.7988	59.6382
2	X2, X6	0.7959	0.7951	69.7595
3	X1, X5, X6	0.8113	0.8101	29.3366
3	X2, X6, Z1	0.8069	0.8057	41.5086
4	X1, X5, X6, Z1	0.8162	0.8147	17.7189
4	X2, X5, X6, Z1	0.8155	0.8140	19.6091
5	X1, X2, X5, X6, Z1	0.8207	0.8188	7.4274
5	X1, X3, X5, X6, Z1	0.8181	0.8162	14.5713
6	X1, X2, X3, X5, X6, Z1	0.8219	0.8197	6.1209
6	X1, X2, X4, X5, X6, Z1	0.8210	0.8188	8.4422
7	X1, X2, X3, X4, X5, X6, Z1	0.8219	0.8194	8.0000

Combining the results of stepwise regression and all subset regression, we decided to delete the x4 variable, that is, the "SOP--Statement of purpose" variable. Next, we will make changes to these six regressors and responses.

7. Transformation

After variable selection, our model expression is:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_5 X_5 + \beta_6 X_6 + \beta_Z Z_1 + \varepsilon \quad (3)$$

We built a multiple regression model based on this expression and found that the image of the four hypotheses in the model is still not significantly improved. As shown in Figure 14. So we decided to make transformations to the model.

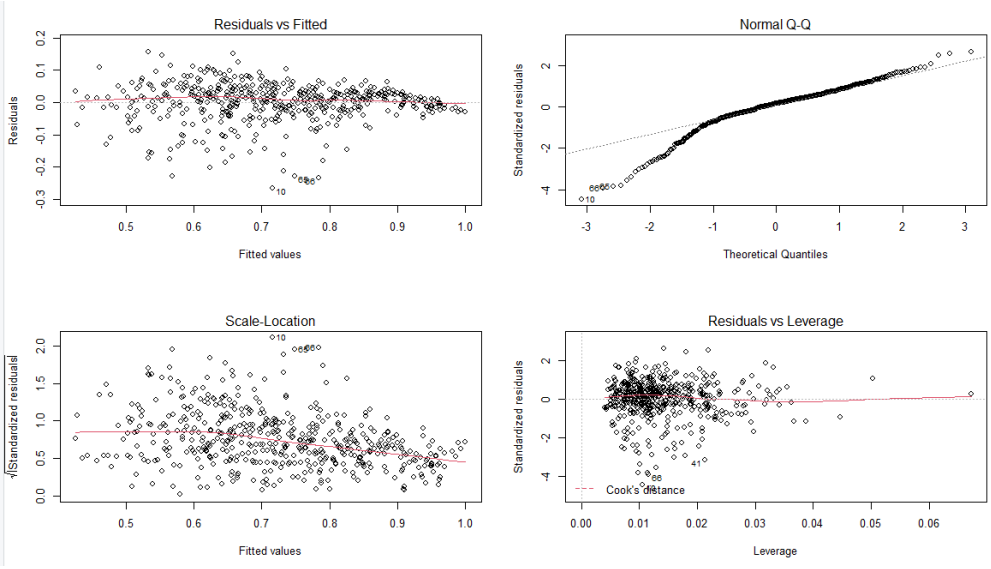

Figure 14 Model image after variable selection

We will transform response and regressors separately.

a) Response(y):

We are using Box-cox transformation, and its formula is as follows:

$$y(\lambda) = \begin{cases} \frac{y^{\lambda}-1}{\lambda}, & \lambda \neq 0 \\ lny, & \lambda = 0 \end{cases} \quad (4)$$

We use the boxcox() function in R to complete the Box-cox transformation. The result is shown in Figure 15.

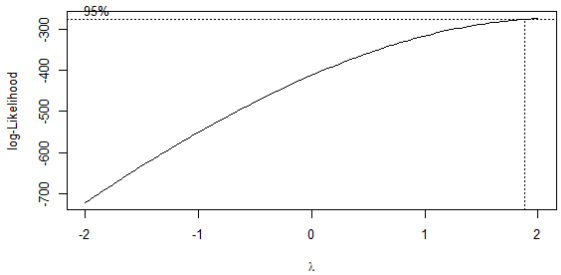

Figure 15 The result of Box-cox transformation

Through Figure 15, we get the best transformation of y as follows:

$$\lambda = 2 \tag{5}$$

$$Y \rightarrow \frac{Y^2-1}{2} \tag{6}$$

b) Regressors(x):

We use the car package in R to calculate the suggested power of the variable. We use the powerTransformation() function to calculate the power. We used this function for all variables in the model, and we found that only the variable x2 satisfies the conditions for transformation. The data is shown in Figure 16.

```
> summary(powerTransform(x2))
bcPower Transformation to Normality
   Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
x2    0.1495           1      -1.3092       1.6083

Likelihood ratio test that transformation parameter is equal
 to 0
 (log transformation)
                              LRT df    pval
LR test, lambda = (0) 0.04038574  1 0.84073

Likelihood ratio test that no transformation is needed
                             LRT df    pval
LR test, lambda = (1) 1.301687  1 0.25391
```

Figure 16 The result of x2 transformation

We can see from the value returned by the function that the recommended transformation power for x2 is 0.1459, and the p-value is 0.25391. Because p-value is greater than 0.05, we convert x2 to log(x2).

Now, we have completed the variable selection and transformation of the model. The expression of our final model is:

$$\frac{y^2-1}{2} = \beta_0 + \beta_1 X_1 + \beta_2 \log(X_2) + \beta_3 X_3 + \beta_5 X_5 + \beta_6 X_6 + \beta_Z Z_1 + \varepsilon \quad (7)$$

8. The second abnormal observation value test and processing

We established a new multiple regression model according to formula 7. We need to process abnormal observations that have not been processed after model optimization (variable selection and transformation).

a) Outliers and high leverage points

After calling the outlierTest() and hat.plot() functions, we get the outliers and high leverage points in the current model. The data is shown in Table 9.

Table 9 Outliers and high leverage of the optimized model

Outliers	10, 11, 65, 66, 67
High leverage points	53, 118, 437

b) High influence point

By calculating the cook distances of all points in the optimized model, we find high influence points. As shown in Figure 17.

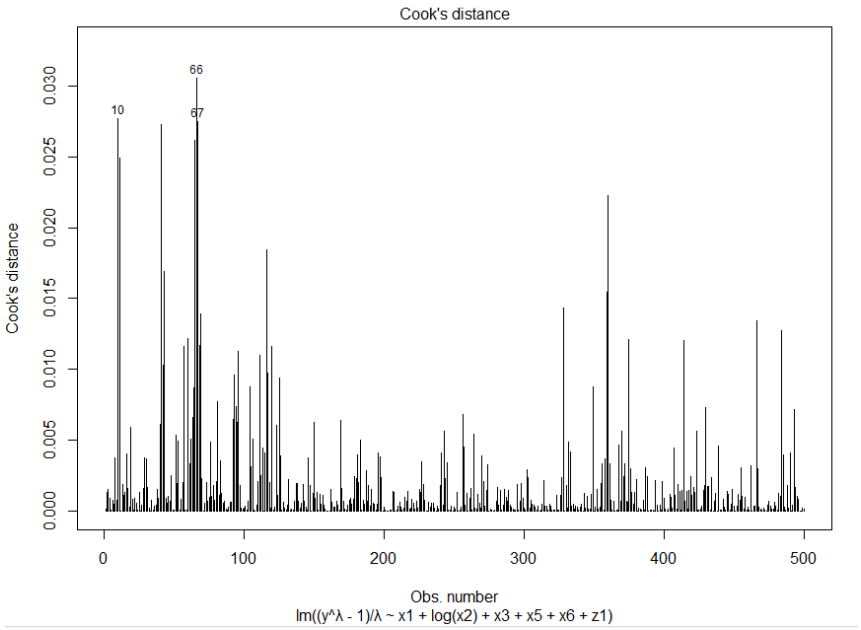

Figure 17 High impact points of the optimized model

We tested these high impact points. The cook distance of all points is not greater than 0.5, so we believe that these high impact points have no actual impact on our model.

c) Abnormal observation processing

We delete the outliers and high leverage mentioned in a). Re-establish the model after the output and perform the above operations again. Stop the operation until the model cannot find outliers.

9. Building and testing the final model

After removing all outliers and high leverage points, we built our latest model. The basic information is shown in Figure 18.

```
> model3<-lm((y^λ-1)/λ~x1+log(x2)+x3+x5+x6+z1,data = data_out_5)
> summary(model3)

Call:
lm(formula = (y^λ - 1)/λ ~ x1 + log(x2) + x3 + x5 + x6 + z1,
    data = data_out_5)

Residuals:
      Min        1Q    Median        3Q       Max
-0.093228 -0.020193  0.004108  0.021470  0.093500

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.5357606  0.1978404 -12.817  < 2e-16 ***
x1           0.0014559  0.0002847   5.114 4.60e-07 ***
log(x2)      0.2272756  0.0524518   4.333 1.80e-05 ***
x3           0.0057112  0.0019875   2.874  0.00424 **
x5           0.0091894  0.0022008   4.175 3.54e-05 ***
x6           0.0848042  0.0055009  15.416  < 2e-16 ***
z11          0.0171910  0.0037379   4.599 5.46e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.03273 on 472 degrees of freedom
Multiple R-squared:  0.8905,    Adjusted R-squared:  0.8892
F-statistic: 640.1 on 6 and 472 DF,  p-value: < 2.2e-16
```

Figure 18 Basic statistics of final model

We compare the data of the final model and the preliminary model, as shown in Table 10.

Table 10 Comparison of final model and preliminary model

	Preliminary model	Final model
R^2	0.8219	0.8905
R^2_{adj}	0.8194	0.8892
Degree of freedom	7	6

Multicollinearity	NULL	NULL
AIC	-1385.	-1907.
BIC	-1347.	-1874.

From the various data, the final model is much better than the preliminary model. The R^2 and R^2_{adj} increases by about 0.07. This means that the established model fits the data set better. In addition to the changes in basic statistics, the final model also performed better in terms of the four assumptions. As shown in Figure 19.

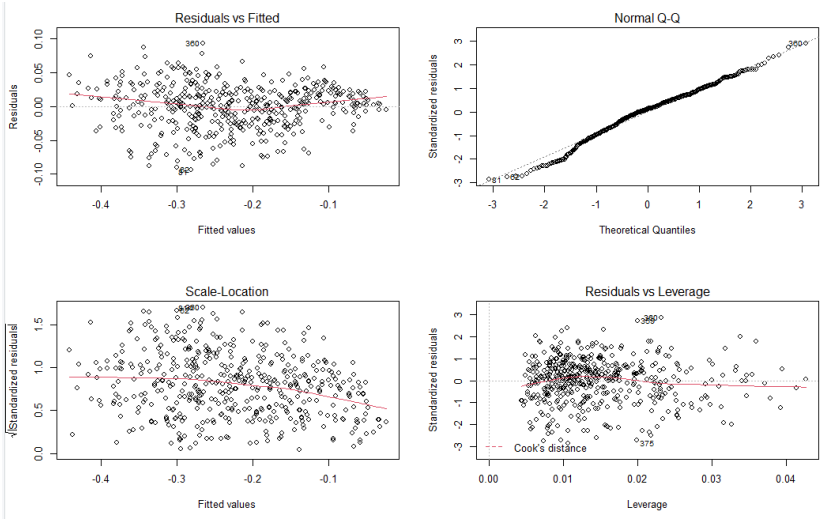

Figure 19 Image of the final model

Through Q-Q plot, we can see that the normality of the model is better, and the residuals are basically distributed on both sides of the straight line in the figure. The other three assumptions have also become better. The multiple regression model of this data set has been built. The multiple regression formula is:

$$\frac{y^2-1}{2} = -2.5357 + 0.0014X_1 + 0.2272\log(X_2) + 0.0057X_3 + 0.0092X_5 + 0.08480X_6 + 0.0172Z_1 + \varepsilon \quad (8)$$

10. Relative weight of variables

In order to find the regressors that affect the response, we use the rela.impo() function in R to get the contribution rate of each variable to the R^2. The data is shown in Table 11.

Table 11 The contribution of each variable to R^2

Variables' relative contribution to R^2	
X1	0.1966
Log(X2)	0.1771
X3	0.1075
X5	0.0960
X6	0.3490
Z1	0.0737

From Table 11, we can see that the largest contribution to R-squared is X6, which is CGPA. Therefore, the biggest impact on the admission rate is the CGPA, which is the student's usual grades. At the same time, X1 (GRE scores) and X2 (TOEFL scores) accounted for 19.66% and 17.71% respectively, which also occupied a relatively important position. Therefore, students who want to go to UCLA to study for a master's degree should focus on learning, and improving various grades is the key.

Conclusion

We completed the model construction of this data set through variable selection, conversion, and processing of abnormal observations. By adjusting the value of R-squared and R-squared, we basically judge that the final model can assume the function of prediction, and give the prediction result with reference function.

In addition to the predictive function, we have also ascertained the importance of each variable. The first is the "SOP" variable, this variable is directly deleted in the "variable selection" link. In addition, there are two variables "LOR" and "Research", which correspond to the strength of the recommendation letter and whether there is any scientific research experience. Although these two variables have not been deleted, the sum of their contribution to the R-squared ratio is about 15%. Therefore, these two variables are just extra points. The remaining variables: GRE scores, TOEFL scores, university rankings and CGPA, all reflect the students' learning ability and level. Therefore, we believe that if students want to enter UCLA to study for a master's degree,

the most important thing is to improve their academic performance. Participate in scientific research and get a good recommendation letter on the basis of good grades.

Reference

Data Sources:

https://www.kaggle.com/mohansacharya/graduate-admissions

Reference:

1. 徐静安,浦静雯,吴芳 & 许保云.(2018).第二十九讲 关于留一法 PRESS 统计量的应用讨论. *上海化工*(09),8-13. doi:10.16759/j.cnki.issn.1004-017x.2018.09.007.
2. 姚棣荣,俞善贤.(1992).基于 PRESS 准则选取预报因子的逐步算法. *大气科学*(02),129-135. doi:CNKI:SUN:DQXK.0.1992-02-000.
3. 储成顶,朱启星,成微,俞敏,鲍玉婷 & 马晓芹.(2006).多元线性回归——最优子集法编程及科研实例.*数理医药学杂志*(03),309-311. doi:CNKI:SUN:SLYY.0.2006-03-040.
4. 黄文珂.(2012).*多元回归建模过程中共线性的诊断与解决方法*(硕士学位论文,哈尔滨工业大学). https://kns.cnki.net/KCMS/detail/detail.aspx?dbname=CMFD201401&filename=1013037125.nh
5. 闫闯.(2011).*多元回归模型中变量选择问题研究*(硕士学位论文,黑龙江大学).https://kns.cnki.net/KCMS/detail/detail.aspx?dbname=CMFD2012&filename=1012287988.nh

Appendix

R code:

```
# Add the packages we need

library(carData)

library(car)

library(olsrr)

library(moments)

library(MASS)

library(boot)

library(relaimpo)

library(ISLR)

library(leaps)

library(broom)

library(robustbase)

library(qpcR)


#Set working path

#setwd('C:/Users/(userName)/Desktop')


#Read the data

data <- read.csv("Admission.csv")

data <- data.frame(data)


#Data processing

##Preview the correlation coefficient between each regressors and response

corr_variables <- cor(data,use = "pairwise.complete.obs")

corr_variables <- data.frame(corr_variables)

View(corr_variables)

##Generate correlation matrix diagram (function comes from car package)
```

```
spm(data,smooth=list(lty.smooth=2, spread = F),main="Matrix of correlations between variables")


#Change the variable name to facilitate subsequent operations

##x1 ~ GRE_Score,x2 ~ TOEFL_Score,x3 ~ University_Rating,x4 ~ SOP,x5 ~ LOR,x6 ~ cGPA,z1 ~ Research,y ~ Chance of dmit

names(data) <-c ("x1","x2","x3","x4","x5","x6","z1","y")

attach(data)

##Because z1 is a categorical variable, it is set as a factor

data$z1 <- as.factor(z1) # set z1 to be a factor

str(z1)

summary(z1)


#Define some functions we need

##Visualize normality histogram

residplot <- function(fit, nbreaks=10) {
   z <- rstudent(fit)
   hist(z, breaks=nbreaks, freq=FALSE,
           xlab="Studentized Residual",
           main="Distribution of Errors")
   rug(jitter(z), col="brown")
   curve(dnorm(x, mean=mean(z), sd=sd(z)),
            add=TRUE, col="blue", lwd=2)
   lines(density(z)$x, density(z)$y,
            col="red", lwd=2, lty=2)
   legend("topright",
             legend = c( "Normal Curve", "Kernel Density Curve"),
             lty=1:2, col=c("blue","red"), cex=.7)
}

##Plot high leverage points

hat.plot <- function(fit){
```

```
    p <- length(coefficients(fit))
    n <- length(fitted(fit))
    plot(hatvalues(fit),main = "Index Plot of Hat Values")
    abline(h=c(2,3)*p/n,col="red",lty=2)
    identify(1:n, hatvalues(fit), names(hatvalues(fit)))
}

#Build a preliminary model
model1 <- lm(y~x1+x2+x3+x4+x5+x6+z1,data = data)
summary(model1)
par(mfrow = c(2,2))
plot(model1)

#Model diagnosis
##Test linearity
crPlots(model1)
##Test Normality
par(mfrow = c(1,2))
qqPlot(model1,labels=row.names(states),id.method="identity",simulate=TRUE,main="Q-Q Plot")
residplot(model1)
##Test Independence of residuals
durbinWatsonTest(model1)
##Test Homoscedasticity
par(mfrow = c(1,1))
ncvTest(model1)
spreadLevelPlot(model1)
##Test multicollinearity
##Judgment criteria: if the vif value is less than 10, there is no multicollinearity
vif(model1)
```

```
#Test abnormal observations on preliminary model

##Outlier

plot(x=fitted(model1),y=rstudent(model1))

abline(h=3,col="red",lty=2)

abline(h=-3,col="red",lty=2)

which(abs(rstudent(model1))>3)

outlierTest(model1)

##High leverage point

hat.plot(model1)

##Hige influence point

cutoff <- 4/(nrow(500-length(model1$coefficients)-2))

plot(model1,which=4,cook.levels = cutoff)

abline(h=cutoff,lty=2,col="red")


#Variable selection

##Obtain the results of two variable selections and build models for comparison

##Stepwise Regression

stepwise_model <- step(model1,direction = 'both')

summary(stepwise_model)

par(mfrow = c(2,2))

plot(stepwise_model)

##All-Subsets Regression

leaps <- regsubsets(y ~.,data=data, nvmax =10)

summary.leaps <- summary(leaps)

par(mfrow=c(2,2))

plot(summary.leaps$rss ,xlab="Number of Variables ",ylab="RSS",type="l")

plot(summary.leaps$adjr2 ,xlab="Number of Variables ",ylab="Adjusted RSq",type="l")

points (6,summary.leaps$adjr2[6], col="red",cex=2,pch=20)
```

```
plot(summary.leaps$cp ,xlab="Number of Variables ",ylab="Cp",type='l')

points (6,summary.leaps$cp [6],col="red",cex=2,pch=20)

plot(summary.leaps$bic ,xlab="Number of Variables ",ylab="BIC",type='l')

points(6,summary.leaps$bic [6],col="red",cex=2,pch =20)

coef(leaps ,6)

ols_step_all_possible(model1)

##The model after all-subsets regression

all_subsets_model<-lm(y~x1+x2+x3+x5+x6+z1,data = data)

summary(all_subset_model)

par(mfrow = c(2,2))

plot(all_subset_model)


#Do Transformation

##If we use powerTransform(), we need to ensure all data >0.

min(y)

min(x1)

min(x2)

min(x3)

min(x4)

min(x5)

min(x6)

min(z1+1)

##Transform regressors using function in R

summary(powerTransform(x1))

summary(powerTransform(x2))

summary(powerTransform(x3))

summary(powerTransform(x5))

summary(powerTransform(x6))

summary(powerTransform(z1+1))
```

```
##Transform response by box-cox transformation
bc_model <- boxcox(y~x1+x2+x3+x4+x5+x6+z1,data = data)
lambda <- bc_model$x
lik <- bc_model$y
bc <- cbind(lambda,lik)
bc[order(-lik),]
λ=2 # calculate λ=2, the best value for y in transformation

#Remodel after variable selection and transformation
model2<-lm((y^λ-1)/λ~x1+log(x2)+x3+x5+x6+z1,data = data)
summary(model2)
par(mfrow = c(2,2))
plot(model2)

#Test abnormal observations on the new model and processing
##Outliers
par(mfrow = c(1,1))
plot(x=fitted(model2),y=rstudent(model2))
abline(h=3,col="red",lty=2)
abline(h=-3,col="red",lty=2)
which(abs(rstudent(model2))>3)
outlierTest(model2)
##High leverage point
hat.plot(model2)
##High influence point
cutoff <- 4/(nrow(500-length(model2$coefficients)-2))
plot(model2,which=4,cook.levels = cutoff)
abline(h=cutoff,lty=2,col="red")
##High influence point Test
```

```
##Prove that the influence point in this model have no influence on the model
olsrr::ols_plot_dffits(model2)
olsrr::ols_plot_dfbetas(model2)
olsrr::ols_plot_diagnostics(model2)
##Dealing with outliers and high leverage points
###Remove and re-detect outliers and high leverage values for the first time
data_out_0<-data[-c(10,11,53,65,66,67,118,437),]
model2_0<-lm((y^λ-1)/λ~x1+log(x2)+x3+x5+x6+z1,data = data_out_0)
which(abs(rstudent(model2_0))>3)
outlierTest(model2_0)
hat.plot(model2_0)
###The second time
data_out_1<-data_out_0[-c(39,57,62,63),]
model2_1<-lm((y^λ-1)/λ~x1+log(x2)+x3+x5+x6+z1,data = data_out_1)
which(abs(rstudent(model2_1))>3)
outlierTest(model2_1)
hat.plot(model2_1)
###The third time
data_out_2<-data_out_1[-c(39,40,106),]
model2_2<-lm((y^λ-1)/λ~x1+log(x2)+x3+x5+x6+z1,data = data_out_2)
which(abs(rstudent(model2_2))>3)
outlierTest(model2_2)
hat.plot(model2_2)
###The forth time
data_out_3<-data_out_2[-c(39,40,106),]
model2_3<-lm((y^λ-1)/λ~x1+log(x2)+x3+x5+x6+z1,data = data_out_3)
which(abs(rstudent(model2_3))>3)
outlierTest(model2_3)
hat.plot(model2_3)
```

```
###The fifth time
data_out_4<-data_out_3[-c(55,79),]
model2_4<-lm((y^λ-1)/λ~x1+log(x2)+x3+x5+x6+z1,data = data_out_4)
which(abs(rstudent(model2_4))>3)
outlierTest(model2_4)
hat.plot(model2_5)
###The sixth time
data_out_5<-data_out_4[-c(309),]
model2_5<-lm((y^λ-1)/λ~x1+log(x2)+x3+x5+x6+z1,data = data_out_5)
which(abs(rstudent(model2_5))>3)
outlierTest(model2_5)
hat.plot(model2_5)

#Final model
##Use the last deleted outlier data
model3<-lm((y^λ-1)/λ~x1+log(x2)+x3+x5+x6+z1,data = data_out_5)
summary(model3)
par(mfrow = c(2,2))
plot(model3)

#Final model diagnosis
##Test linearity
crPlots(model3)
##Test Normality
par(mfrow = c(1,2))
qqPlot(model3)
residplot(model3)
##Test Error Independence
durbinWatsonTest(model3)
```

```
##Test Homoscedasticity

par(mfrow = c(1,1))

ncvTest(model3)

spreadLevelPlot(model3)

##Test VIF

vif(model3)


#Conclusion

##Values of various statistics

glance(model3)

leaps_final <-regsubsets((y^λ-1)/λ~x1+log(x2)+x3+x5+x6+z1,data = data_out_5, nvmax =10)

summary_leaps_final<-summary(leaps_final)

View(summary_leaps_final)

PRESS(model3,verbose = TRUE)

##Relative weight between independent variables

###Which variable is obtained from the graph has the greatest impact on the result

crlm <- calc.relimp(model3, type =    "car", rela = TRUE )

crlm

par(mfrow = c(1,1))

plot(crlm)
```