Multivariate Regression Model for Explaining House Price

Yangxuan Xu

Section 1. Introduction

The purpose of this project is to fit a multivariate regression model for this real estate valuation data in Xindian District, New Taipei City, Taiwan, focusing on explaining how these factors influence the unit house price in this area.

From a macro point of view, the price of real estate is a significant reference to the local economy and living standards of people. Most people consider buying their own home thus the valuation of house price does matter for them. Real estate price is influenced by numerous factors, some of which cannot be well quantified (like monetary policy, investment expectation, etc). While others like location, house age, and convenience, can be expressed as quantitative numbers. Under this data, I will fit a multivariate regression model to try to figure out how these factors influence real estate price in this area.

The data comes from Professor I-ChengYeh and Tzu-KuangHsu in Department of Civil Engineering, Tamkang University, Taiwan. It was donated in 2018. Most of the data set was collected from the public database of Ministry of the Inferior during the period of June 2012 to May 2013 in Taipei City. For the data that cannot be directly acquired from the database, like X_3 (distance to the nearest MRT station) and X_4 (number of convenience stores), the authors got them directly from Google Map **(Yeh&Hsu, 2018)**. Therefore, we can think the data has a valid source.

The outline of the remainder of this report is as follows. In Section 2, I present a brief introduction of characteristics of the data. Based on these characteristics, in Section 3 I will use R to conduct the multivariate regression steps to find a good model. Discussion of the model deficiency and some important details will be the Section 4. Concluding remarks can be found in Section 5, followed by the appendix which involves data source and references.

Section 2. Data Characteristics

The data was compiled and donated in 2018 by Professor I-ChengYeh and Tzu-KuangHsu in Department of Civil Engineering, Tamkang University, Taiwan. These market historical data set of real estate valuation are collected from Xindian District, New Taipei City, Taiwan.

The total number of instances is 414, without any missing value. This set is a good multivariate regression data set, whose attributes are all integer or real numbers. The response variable Y is the house price of unit area (10000 New Taiwan Dollar/Ping, where Ping is a local unit and 1 Ping=3.3 m^2). The other 6 explanatory variables are as follows:

x1=the transaction date (for example, 2013.250=2013 March, 2013.500=2013 June, etc.)

x2=the house age (unit: year)

x3=the distance to the nearest MRT station (unit: meter)

x4=the number of convenience stores in the living circle on foot (integer)

x5=the geographic coordinate, latitude. (unit: degree)

x6=the geographic coordinate, longitude. (unit: degree)

Table 20.1. provides more details on the explanatory variables and response variable (some decimals are omitted).

Table 20.1. *Real Estate Valuation Summary Statistics (R Output)*

x1	x2	x3	x4
Min. :2012.67	Min. : 0.0000	Min. : 23.383	Min. : 0.0000
1st Qu.:2012.92	1st Qu.: 9.0250	1st Qu.: 289.325	1st Qu.: 1.0000
Median :2013.17	Median :16.1000	Median : 492.231	Median : 4.0000
Mean :2013.15	Mean :17.7126	Mean :1083.886	Mean : 4.0942
3rd Qu.:2013.42	3rd Qu.:28.1500	3rd Qu.:1454.279	3rd Qu.: 6.0000
Max. :2013.58	Max. :43.8000	Max. :6488.021	Max. :10.0000
x5	x6	y	
Min. :24.9321	Min. :121.474	Min. : 7.6000	
1st Qu.:24.9630	1st Qu.:121.528	1st Qu.: 27.7000	
Median :24.9711	Median :121.539	Median : 38.4500	
Mean :24.9690	Mean :121.533	Mean : 37.9802	
3rd Qu.:24.9775	3rd Qu.:121.543	3rd Qu.: 46.6000	
Max. :25.0146	Max. :121.566	Max. :117.5000	

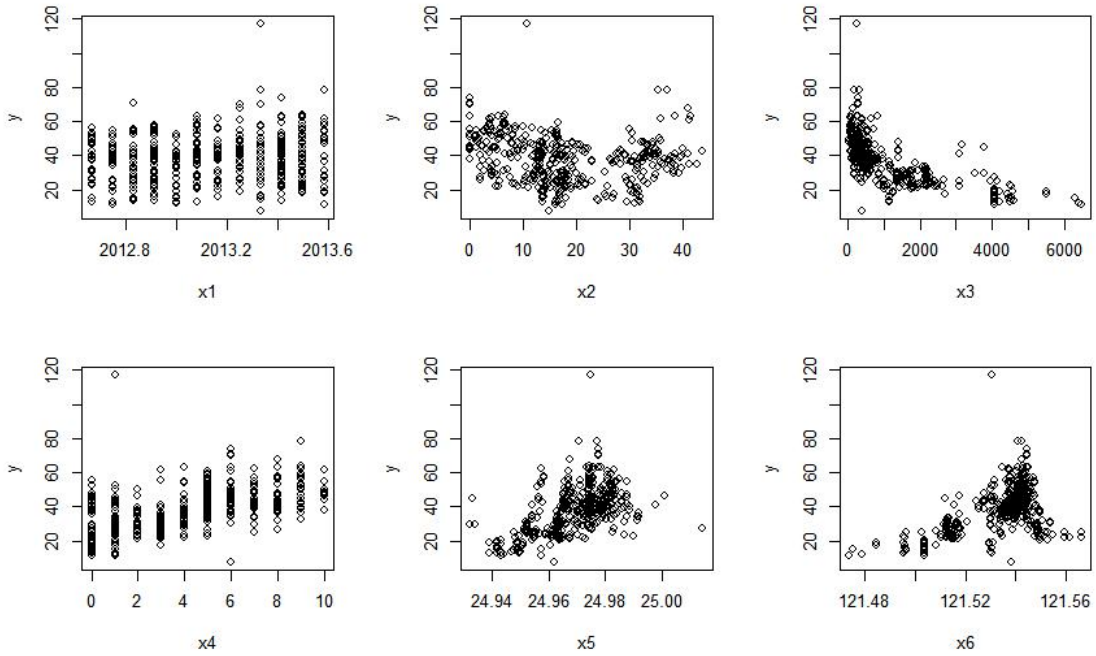
It's worth noting that x1 is not a usual form we use to record date. The author used the decimal parts to represent month. Here I still see it as numeric variables since it will

be more convenient interpretating the coefficient. From Figure 20.2, we can find the data was collected on certain dates thus possible transformation for x1 is to treat it as ordinal variables, which improves the goodness-of-fit but it's more difficult to interpret the coefficients.

The x5 and x6 are not preferred to be used directly under this circumstance since they are spatial data and it's hard to directly relate latitude or longitude to house price. A possible method is to conduct a polar coordination transformation of these two attributes to r(distance to the center) and θ(angular parameter). I will explain how I choose the center point and do the transformation in Section 3.

Figure 20.2. shows the relationship between explanatory variables and the response variable. It can be seen from these plots that the unit house price(y) increases with transaction date(x1), number of convenience stores(x4) and spatial variables(x5&x6). While house age(x2) and distance to the nearest station(x3) seem to have negative relationship with unit house price.

Figure 20.2. *Relationship between inputs and outputs (R Output)*



To understand how these explanatory variables influence house price in Xindian District, I will use multivariate regression to fit models, discuss findings and draw the conclusion.

Section 3. Multivariate Regression Analysis

A) Variable Selection & Model Construction

I will construct two different models and make comparison to see which one is better in explaining the house price.

1) Firstly, I fitted the full model with all the variables x1, ..., x6, including interaction terms. Using the step-wise method to do variable selection, checking variance inflation factor (VIF) and p-values, I tried to drop some less important variables or those with VIF>10(multicollinearity problem) and do some transformation, getting the best model (1):

$$\ln(Y) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \beta_4 X_{i4} + \beta_5 X_{i5} + \beta_6 (X_{i3} * X_{i4}) + \varepsilon_i \qquad (1)$$

Taking the natural logarithm of response helps to improve the goodness-of-fit a lot, so I did this transformation.

2) Another idea is to conduct a polar coordinate transformation of x5(latitude) and x6(longitude). This is a usually considered transformation when confronted with spatial variables.

Before conducting the transformation, we need to choose the origin. Given the general idea that the house located in the center of a city is often more expensive, I calculated the mean value of x5(latitude) and x6(longitude) for those with unit house price higher than 70. This point is set to be the origin, an approximate center of the “rich area”. And then we use r to denote the distance to the center point, together with the angular coordinate θ. The idea on how to choose the threshold to find origin will be discussed in Section 4.

Following the similar steps in 1), my model (2) is showed below:

$$\ln(Y) = \beta_0' + \beta_1' X_{i1} + \beta_2' X_{i2} + \beta_3' \ln(X_{i3}) + \beta_4' X_{i4} + \beta_5' r + \varepsilon_i' \qquad (2)$$

Taking the natural logarithm of some variables also improve the goodness-of-fit. θ is dropped during the variable selection, which means the angular coordinate is not important to decide house price. This is interesting because most of the time θ and r decide the position together and position is an important factor in real estate valuation. A possible explanation is most of the district are rural area (seen from the map Figure 20.4), and there is no big difference in house price among these places far from urban area.

The variable r can be seen as a non-linear transformation of x5 and x6, a better interpretation of the latitude and longitude.

B) Parameter Estimation

We use the summary function in R to get a view of the two models. Table 20.3. gives us the output and we can see the estimated parameters. The p-values are small which means that these variables are significant. The adjusted R square, an important evaluation metric in comparing models, are close between the two models.

Table 20.3. *Summary of model(1) and model(2) (R Output)*

```
lm(formula = log(y) ~ x1 + x2 + x3 + x4 + x5 + x3:x4, data = dat)

Residuals:
     Min       1Q   Median       3Q      Max
-1.66445 -0.11075  0.00606  0.10768  1.02761

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.119e+02  7.824e+01  -6.544 1.81e-10 ***
x1           1.411e-01  3.744e-02   3.768 0.000189 ***
x2          -7.100e-03  9.264e-04  -7.664 1.34e-13 ***
x3          -1.153e-04  1.318e-05  -8.749  < 2e-16 ***
x4           3.822e-02  4.870e-03   7.849 3.75e-14 ***
x5           9.280e+00  1.090e+00   8.510 3.37e-16 ***
x3:x4       -3.474e-05  5.988e-06  -5.802 1.32e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.213 on 407 degrees of freedom
Multiple R-squared:  0.7097,    Adjusted R-squared:  0.7054
F-statistic: 165.8 on 6 and 407 DF,  p-value: < 2.2e-16
```

Model (1)

```
lm(formula = log(y) ~ x1 + x2 + log(x3) + x4 + r, data = dat)

Residuals:
     Min       1Q   Median       3Q      Max
-1.68927 -0.10562 -0.00026  0.10647  0.99431

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.387e+02  7.548e+01  -4.487 9.39e-06 ***
x1           1.705e-01  3.750e-02   4.546 7.21e-06 ***
x2          -6.888e-03  9.385e-04  -7.339 1.18e-12 ***
log(x3)     -1.042e-01  1.781e-02  -5.849 1.01e-08 ***
x4           1.304e-02  4.990e-03   2.613  0.00931 **
r           -1.305e+01  1.245e+00 -10.484  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2122 on 408 degrees of freedom
Multiple R-squared:  0.7111,    Adjusted R-squared:  0.7076
F-statistic: 200.9 on 5 and 408 DF,  p-value: < 2.2e-16
```

Model (2)

C) Model Diagnostics

Based on the definition of multivariate regression model, four basic assumptions need to be obeyed. If not, the inferences about the model will be less effective. I have performed the following diagnostic check of the basic assumptions made by regression:

1. Residual plot & Breusch-Pagan Test to check Linearity and Equal Variance
2. Normal Q-Q plot & Shapiro-Wilks Test to check Normality
3. Durbin-Watson Test to check Error Independence

The results show that the normality assumption is violated for both two models, with all other assumptions substantially obeyed. Box-Cox transformation has been tried but doesn't help to improve. The residual not following normal distribution can be seen as a small deficiency of the models, which will be discussed in Section 4 on how it will influence interpreting.

Besides, some influential points and outliers exist. For model (1), there are 18 outliers and 28 influential points. While for model (2), 20 outliers and 24 influential points are found.

D) Evaluation

From AIC, BIC and adjusted R^2, we can see that both models can fit the data well, with model (1) performing slightly better. But based on the likelihood ratio test, since p-value(0.1518) is large, we have no evidence to say model (1) is better than model (2).

From the explaining aspect, the variable r used in model (2) is better than using x5(latitude) or x6(longitude). This variable shows that there is a negative relationship between the distance to center and unit house price, which is more universally meaningful. Instead, the relationship between latitude or longitude and unit house price makes less sense. Besides, r can be seen as a non-linear transformation of x5 and x6, which means that we didn't drop any initial variables in model (2).

In conclusion, we choose model (2) as our fitted model to do explain the unit house price in this district.

Section 4. Discussion

1) The Data Set

In Section 2 I have presented the basic characteristics of this data set. Here I want to discuss some more thinking about the data.

From the map (Figure 20.4.) we can see that Xindian is a small district of New Taipei City, most of which are rural area. One good thing on focusing on such a small region is that we don't need to consider macro factors like international trading and house pricing policy in different areas. The data set gives us only 6 explanatory variables and we only need to consider them. More clear and direct relationship can be found in order to explain how they influence unit house price.

Figure 20.4. *Map of Xindian District (from Google Maps)*

The deficiency of the data comes from the small sample size and limited variables. There will be no problem if we just use them to fit a model and interpret in Xindian, but when considering universality which I will discuss later, the model will be less meaningful if conducted in other places.

In my model (2), I conduct a transformation of x5(latitude) and x6(longitude) to r, which is the distance to the origin. The reason why I did this is that it's hard to conclude what relationship between latitude & longitude and unit house price. This transformation can be seen as a previous interpretation of the spatial variables. Other possible methods can be tried to interpret the spatial variables, like classifying the district into different small regions and using dummy variables to represent the categories. A better visual environment can also raise the surrounding house price (Liao, X., Deng, M., & Huang, H., 2021).

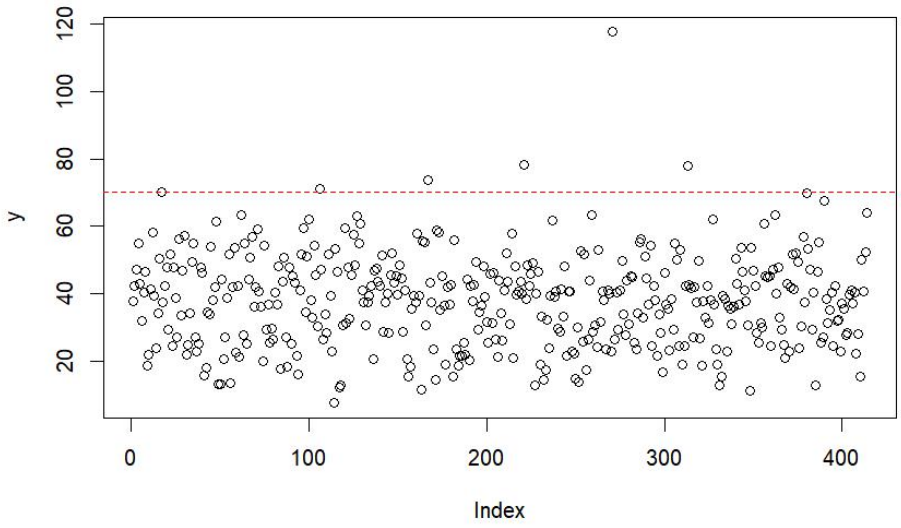
Except the spatial variables, explaining some other variables is also a tricky problem. For example, x1(transaction date) cannot be easily correlated with house price. One possible reason for the positive relationship between x1 and y is the appreciation in house value over time, which comes from investment perspective.

2) Threshold for choosing origin in model (2)

How to choose the threshold for deciding the origin significantly influences model (2). At the end of my R codes, I tried different thresholds (y=0, 30, 50, 80) to choose the center point to fit the models and compared them with model (1). Interestingly, the p-value for likelihood ratio test is small for these thresholds, which means model (1) is better. While when choosing y=70, model (2) is preferred than model (1).

In Figure 20.5. we can see most points have unit house price lower than 70. If I lower the threshold, much more points will be included and the mean value will deviate the most rich area easily. If the threshold raises, too few points are included. Thus choosing the threshold y=70 and calculating the mean value of spatial coordinates of those points helps us to find the center point which denotes so-called “rich area”. Since there is still some intuition in choosing the threshold, better mathematical method can be applied to do the parameter tuning.

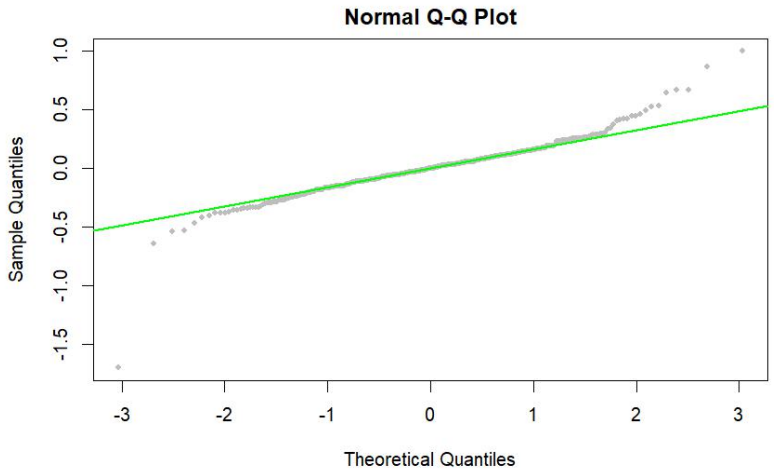
Figure 20.5. *Plots for unit house price (R Output)*



3) Model Assumption

Based on the normal Q-Q plot and S-W test, we can conclude that the residuals didn't follow the normal distribution. Figure 20.6. is the Q-Q plot for residuals of model (2). Since the points on the left side are lower and points on the right side are higher than the line, the residuals have a short-tail distribution, which means that the data is more centralized than normal distribution.

Figure 20.6. *Normal Q-Q Plot for residuals of Model (2) (R Output)*



I cannot find good transformation to fix this problem, thus I decide to keep the current model and see whether this problem hinders my goal of doing regression a lot. More centralized data means the variance will be deflated, resulting a smaller p-value. With smaller p-value, we are easier to reject the null hypothesis even though the variable is less important given H_0: beta = 0.

But in general, this problem only matters when the focus is the significance or confidence interval of the variance term (Maas, C. J. M., & Hox, J. J., 2004). Since my focus is interpretation, namely, the parameter estimation and overall good-of-fitness, we are less considerable about normal assumption. Further improvement can be made, but now we still use the model with this deficiency.

4) Universality

After getting the model from this data set, we need to consider will this model be useful under other circumstances. My conclusion for this question is this model has little universality.

One main reason is the small sample size and limited variables. Hundreds of factors influence the house price thus 5~6 variables can hardly explain house price in complicated cases. Besides, the choice of the variables in this data set are not important enough. Factors like whether in good school district or whether has good scenery and environment are more important in deciding house price.

But as I explained before, this model will have a good interpretation in this small area. Less universality is acceptable.

5) Model Limitation and Further Improvement

In this part I will discuss limitation of the multivariate model and further improvement.

Firstly, I use a linear model so we need to check the patterns between explanatory variables and response variable at the beginning. For this data, some variables have a significant linear pattern but some only a close linear pattern. Thus, some outliers occur which means that these points don't fit the model well.

Secondly, there are model assumptions for multivariate linear regression. For this fitted model, normality assumption is violated. If we focus on the confidence interval or variance, the model will be less effective.

To improve this, other non-linear model or model dealing with non-normal distributed residuals can be used. For house price prediction, hedonic method and cluster method can also considered to be conducted (Bourassa, S., Cantoni, E., & Hoesli, M., 2010).

Section 5. Conclusion

Section 1 and 2 present the general background of real estate valuation together with the characteristics of the data. In Section 3, I follow the multivariate regression steps to conduct variable selection, model construction, parameter estimation, model diagnostics, and evaluation. In Section 4, some important discussions are gone through. We talk about the shortcoming/advantage of the data set, the construction of

model (2), analyze the influence of normal assumption violation, discuss the universality and model limitation.

We get two feasible model, make comparison, and finally choose Model (2). From the R summary output (Table 20.3.), we can make an interpretation of the coefficients of the variables:

1. With other variables fixed, one unit increase in the transaction date(x1) leads to 1.1859 units increase in unit house price(y).
2. With other variables fixed, one unit increase in the house age(x2) leads to 0.9931 units increase in unit house price(y).
3. With other variables fixed, one unit increase in the natural logarithm of distance to the nearest MRT station(x3) leads to 0.9011 units increase in unit house price(y).
4. With other variables fixed, one unit increase in the number of convenience stores(x4) lead to 1.0131 units increase in unit house price(y).
5. With other variables fixed, one unit increase in the distance to the center point(r) leads to 0.00000214 unit increase in unit house price(y).

Appendix

A0. Data Resource

https://archive.ics.uci.edu/ml/datasets/Real+estate+valuation+data+set

A1. References

Bourassa, S., Cantoni, E., & Hoesli, M. (2010). Predicting House prices with spatial dependence: A comparison of alternative methods. Journal of Real Estate Research, 32(2), 139–160.

Liao, X., Deng, M., & Huang, H. (2021). Analyzing multiscale spatial relationships between the house price and visual environment factors. Applied Sciences, 12(1), 213.

Maas, C. J. M., & Hox, J. J. (2004). The influence of violations of assumptions on multi-level parameter estimates and their standard errors. *Computational Statistics & Data Analysis*, *46*(3), 427–440.

Yeh, I. C., & Hsu, T. K. (2018). Building real estate valuation models with comparative approach through case-based reasoning. Applied Soft Computing, 65, 260-271.

A2. R Codes

Multivariate Regression Model for Explaining House Price

Yangxuan Xu

Data Import & Variables Checking

```
library(readxl)
dat <- read_excel("Data/Real estate valuation data set.xlsx")
dat=dat[-1]
colnames(dat)=c("x1","x2","x3","x4","x5","x6","y")
anyNA(dat) # check if any missing value
```

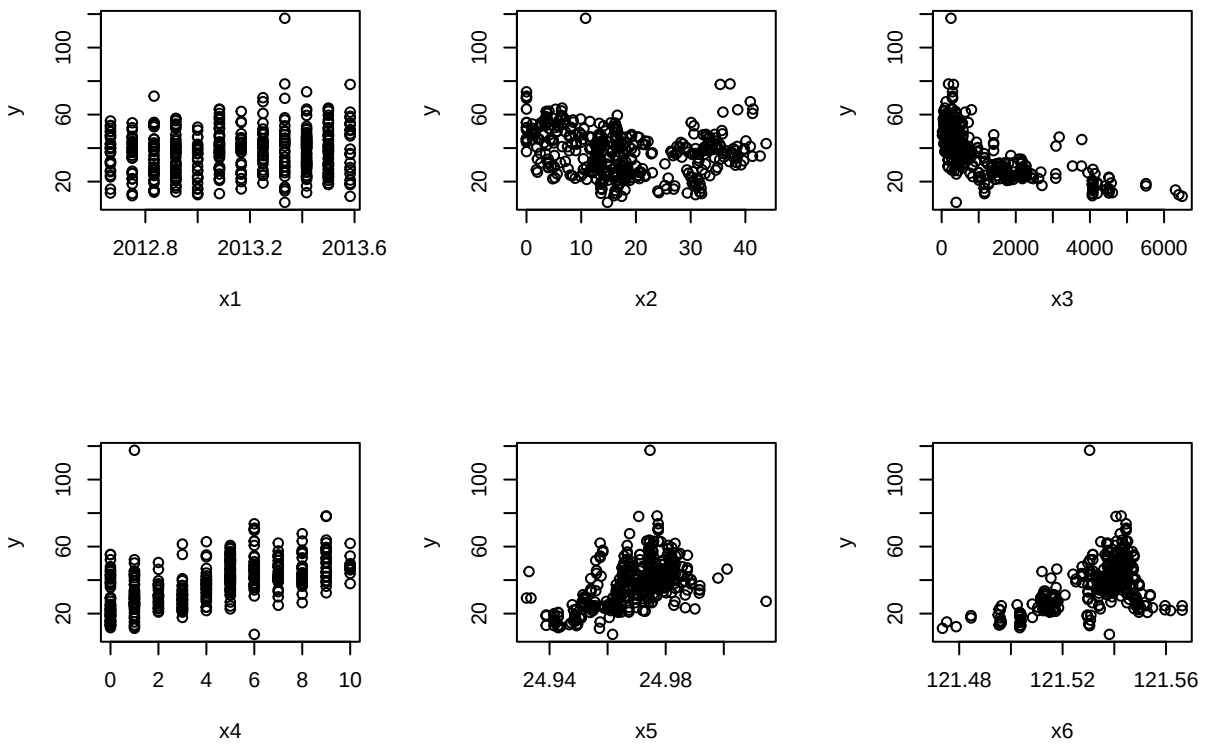
```
## [1] FALSE
```

```
summary(dat,digits=6)
```

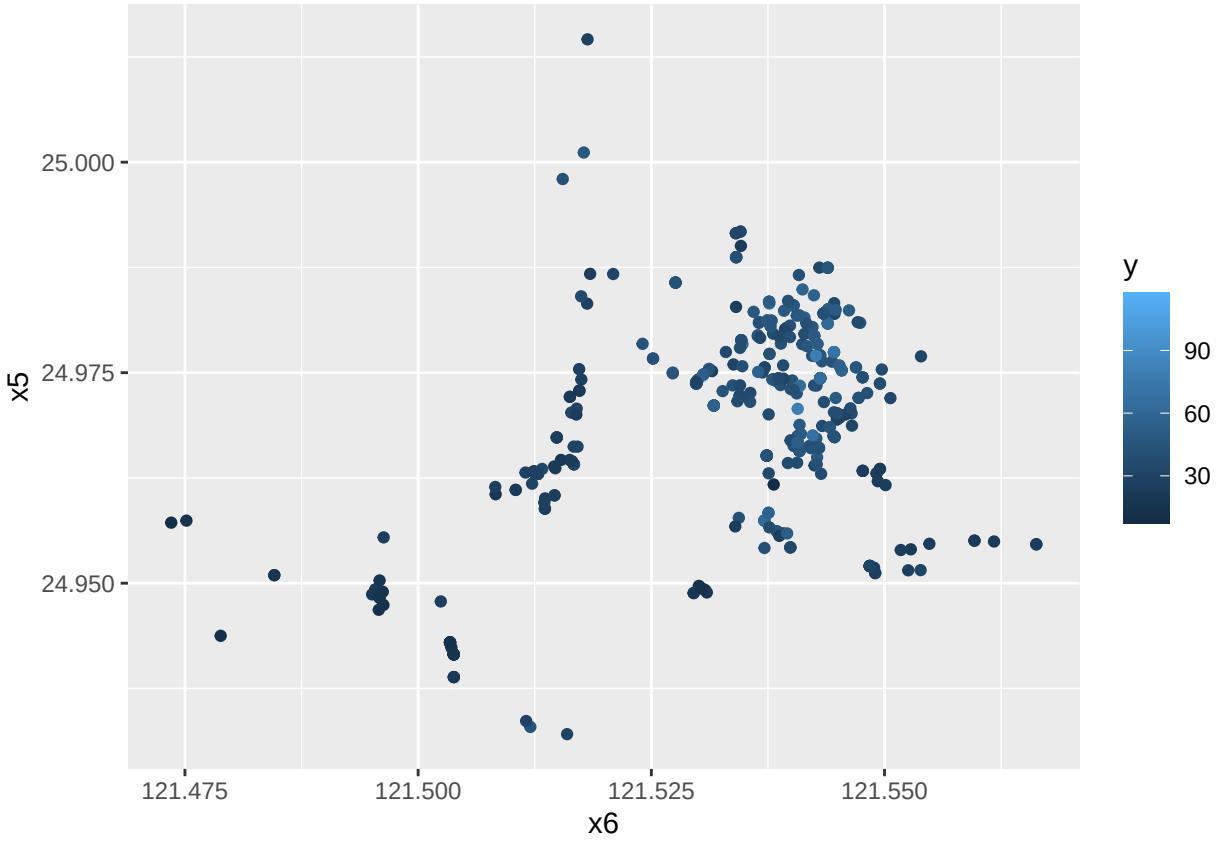
```
##        x1               x2               x3               x4
##  Min.   :2012.67   Min.   : 0.0000   Min.   :  23.383   Min.   : 0.0000
##  1st Qu.:2012.92   1st Qu.: 9.0250   1st Qu.: 289.325   1st Qu.: 1.0000
##  Median :2013.17   Median :16.1000   Median : 492.231   Median : 4.0000
##  Mean   :2013.15   Mean   :17.7126   Mean   :1083.886   Mean   : 4.0942
##  3rd Qu.:2013.42   3rd Qu.:28.1500   3rd Qu.:1454.279   3rd Qu.: 6.0000
##  Max.   :2013.58   Max.   :43.8000   Max.   :6488.021   Max.   :10.0000
##        x5               x6               y
##  Min.   :24.9321   Min.   :121.474   Min.   :  7.6000
##  1st Qu.:24.9630   1st Qu.:121.528   1st Qu.: 27.7000
##  Median :24.9711   Median :121.539   Median : 38.4500
##  Mean   :24.9690   Mean   :121.533   Mean   : 37.9802
##  3rd Qu.:24.9775   3rd Qu.:121.543   3rd Qu.: 46.6000
##  Max.   :25.0146   Max.   :121.566   Max.   :117.5000
```

Visualization

```
par(mfrow=c(2,3))
plot(dat$y~dat$x1,xlab='x1',ylab='y')
plot(dat$y~dat$x2,xlab='x2',ylab='y')
plot(dat$y~dat$x3,xlab='x3',ylab='y')
plot(dat$y~dat$x4,xlab='x4',ylab='y')
plot(dat$y~dat$x5,xlab='x5',ylab='y')
plot(dat$y~dat$x6,xlab='x6',ylab='y')
```




```
# patterns between inputs and response
par(mfrow=c(1,1))
library(ggplot2)
ggplot()+geom_point(data=dat,aes(x=x6,y=x5,col=y)) #geographical plots
```



Variable Selection/Model Construction

```
lm0=lm(y~(x1+x2+x3+x4+x5+x6)^2,data=dat)
summary(lm0)
```

```
##
## Call:
## lm(formula = y ~ (x1 + x2 + x3 + x4 + x5 + x6)^2, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.474  -4.329  -0.738   3.293  70.522
##
## Coefficients: (3 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.040e+04  1.724e+04  -2.923  0.00366 **
## x1           6.782e+00  4.415e+00   1.536  0.12527
## x2           3.325e+02  7.527e+02   0.442  0.65898
## x3           8.767e+00  4.172e+00   2.101  0.03625 *
## x4           5.422e+03  4.136e+03   1.311  0.19061
## x5           9.939e+02  1.631e+02   6.092 2.65e-09 ***
## x6           9.854e+01  1.178e+02   0.837  0.40329
## x1:x2        1.043e-01  1.259e-01   0.828  0.40825
```

```
## x1:x3       -8.601e-04  1.352e-03  -0.636  0.52490
## x1:x4       -3.887e-01  6.074e-01  -0.640  0.52257
## x1:x5               NA         NA      NA       NA
## x1:x6               NA         NA      NA       NA
## x2:x3        8.616e-06  7.877e-05   0.109  0.91295
## x2:x4        9.448e-03  1.384e-02   0.682  0.49534
## x2:x5       -2.897e+00  4.374e+00  -0.662  0.50816
## x2:x6       -3.870e+00  5.761e+00  -0.672  0.50219
## x3:x4       -1.687e-03  3.852e-04  -4.378 1.53e-05 ***
## x3:x5       -2.229e-01  4.396e-02  -5.072 6.08e-07 ***
## x3:x6       -1.212e-02  2.702e-02  -0.449  0.65394
## x4:x5       -1.071e+02  2.150e+01  -4.981 9.48e-07 ***
## x4:x6       -1.616e+01  3.118e+01  -0.518  0.60460
## x5:x6               NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.903 on 395 degrees of freedom
## Multiple R-squared:  0.6773, Adjusted R-squared:  0.6626
## F-statistic: 46.06 on 18 and 395 DF,  p-value: < 2.2e-16
```

```
# do step-wise variable selection
step(lm0,direction="both",trace=FALSE)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x2:x6 + x3:x4 +
##     x3:x5 + x4:x5, data = dat)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x4           x5
##  -4.299e+04    5.946e+00    5.645e+02    5.619e+00    2.695e+03    9.495e+02
##          x6        x2:x6        x3:x4        x3:x5        x4:x5
##   6.049e+01   -4.647e+00   -1.574e-03   -2.252e-01   -1.079e+02
```

```
lm1=lm(y~x1+x2+x3+x4+x5+x6+x2:x6+x3:x4+x3:x5+x4:x5,data=dat)
summary(lm1)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x2:x6 + x3:x4 +
##     x3:x5 + x4:x5, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.470  -4.295  -0.544   3.206  71.136
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.299e+04  9.226e+03  -4.659 4.32e-06 ***
## x1           5.946e+00  1.383e+00   4.301 2.14e-05 ***
## x2           5.645e+02  3.768e+02   1.498    0.135
## x3           5.619e+00  9.233e-01   6.086 2.70e-09 ***
```

```
## x4           2.695e+03  4.773e+02   5.646 3.11e-08 ***
## x5           9.495e+02  1.115e+02   8.518 3.28e-16 ***
## x6           6.049e+01  6.536e+01   0.925    0.355
## x2:x6       -4.647e+00  3.101e+00  -1.499    0.135
## x3:x4       -1.574e-03  2.507e-04  -6.279 8.80e-10 ***
## x3:x5       -2.252e-01  3.698e-02  -6.090 2.64e-09 ***
## x4:x5       -1.079e+02  1.911e+01  -5.643 3.16e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.845 on 403 degrees of freedom
## Multiple R-squared:  0.6756, Adjusted R-squared:  0.6676
## F-statistic: 83.93 on 10 and 403 DF,  p-value: < 2.2e-16
```

```
# drop x6 and x2:x6
lm2=lm(y~x1+x2+x3+x4+x5+x3:x4+x3:x5+x4:x5,data=dat)
summary(lm2)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x3:x4 + x3:x5 + x4:x5,
##     data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.414  -4.333  -0.746   3.328  71.191
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.640e+04  3.837e+03  -9.486  < 2e-16 ***
## x1           5.982e+00  1.383e+00   4.327 1.91e-05 ***
## x2          -2.618e-01  3.436e-02  -7.620 1.82e-13 ***
## x3           5.824e+00  8.661e-01   6.724 6.05e-11 ***
## x4           2.797e+03  4.464e+02   6.266 9.48e-10 ***
## x5           9.771e+02  1.032e+02   9.470  < 2e-16 ***
## x3:x4       -1.590e-03  2.385e-04  -6.667 8.56e-11 ***
## x3:x5       -2.334e-01  3.470e-02  -6.726 5.96e-11 ***
## x4:x5       -1.120e+02  1.788e+01  -6.263 9.65e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.848 on 405 degrees of freedom
## Multiple R-squared:  0.6738, Adjusted R-squared:  0.6673
## F-statistic: 104.6 on 8 and 405 DF,  p-value: < 2.2e-16
```

```
library(faraway)
# check multicollinearity
vif(lm2)
```

```
##           x1           x2           x3           x4           x5        x3:x4
## 1.019343e+00 1.027448e+00 8.013781e+06 1.159611e+07 1.099518e+01 1.520145e+00
##        x3:x5        x4:x5
## 8.007367e+06 1.159943e+07
```

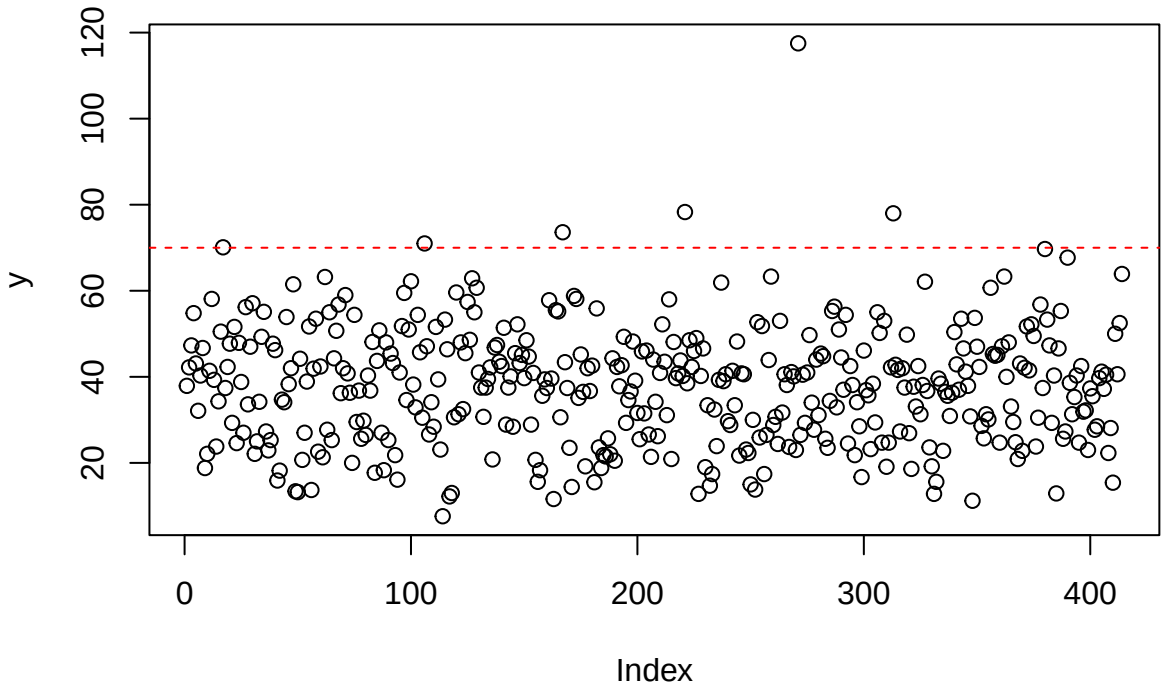
```r
# there exists multicollinearity
# drop x3:x5 and x4:x5
lm_1=lm(log(y)~x1+x2+x3+x4+x5+x3:x4,data=dat)
summary(lm_1) # predictors are significant
```

```
##
## Call:
## lm(formula = log(y) ~ x1 + x2 + x3 + x4 + x5 + x3:x4, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.66445 -0.11075  0.00606  0.10768  1.02761
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.119e+02  7.824e+01  -6.544 1.81e-10 ***
## x1           1.411e-01  3.744e-02   3.768 0.000189 ***
## x2          -7.100e-03  9.264e-04  -7.664 1.34e-13 ***
## x3          -1.153e-04  1.318e-05  -8.749  < 2e-16 ***
## x4           3.822e-02  4.870e-03   7.849 3.75e-14 ***
## x5           9.280e+00  1.090e+00   8.510 3.37e-16 ***
## x3:x4       -3.474e-05  5.988e-06  -5.802 1.32e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.213 on 407 degrees of freedom
## Multiple R-squared:  0.7097, Adjusted R-squared:  0.7054
## F-statistic: 165.8 on 6 and 407 DF,  p-value: < 2.2e-16
```

```r
vif(lm_1) # no multicollinearity
```

```
##       x1       x2       x3       x4       x5    x3:x4
## 1.014400 1.013775 2.518950 1.872938 1.666907 1.300180
```

```r
plot(dat$y,ylab='y')
abline(h=70,col='red',lty=2)
```




```
sum(dat$y>70)
```

```
## [1] 6
```

```
center_x=mean(dat[which(dat$y>70),]$x6)
center_y=mean(dat[which(dat$y>70),]$x5)
c(center_x,center_y)
```

```
## [1] 121.54126  24.97578
```

```
dat$r=sqrt((dat$x6-center_x)^2+(dat$x5-center_y)^2)
dat$theta=atan((dat$x6-center_x)/(dat$x5-center_y))


lm4=lm(log(y)~(x1+x2+x3+x4+r+theta)^2,data=dat)
summary(lm4)
```

```
##
## Call:
## lm(formula = log(y) ~ (x1 + x2 + x3 + x4 + r + theta)^2, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71744 -0.10860  0.00278  0.10429  0.93667
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.052e+02  2.792e+02  -1.451  0.14751
## x1           2.033e-01  1.387e-01   1.466  0.14352
## x2          -4.581e+00  6.835e+00  -0.670  0.50313
## x3          -1.571e-02  2.021e-01  -0.078  0.93808
## x4           4.753e+01  3.381e+01   1.406  0.16054
## r            3.769e+01  1.842e+04   0.002  0.99837
## theta       -4.320e+01  1.045e+02  -0.413  0.67955
## x1:x2        2.270e-03  3.395e-03   0.669  0.50415
## x1:x3        7.793e-06  1.004e-04   0.078  0.93817
## x1:x4       -2.361e-02  1.679e-02  -1.406  0.16058
## x1:r        -3.228e-02  9.149e+00  -0.004  0.99719
## x1:theta     2.150e-02  5.192e-02   0.414  0.67903
## x2:x3        6.082e-06  2.580e-06   2.358  0.01887 *
## x2:x4        6.053e-04  4.019e-04   1.506  0.13280
## x2:r        -3.664e-01  2.116e-01  -1.732  0.08415 .
## x2:theta     3.419e-04  1.250e-03   0.273  0.78464
## x3:x4       -4.505e-05  1.373e-05  -3.280  0.00113 **
## x3:r         2.384e-03  9.581e-04   2.488  0.01327 *
## x3:theta    -1.016e-04  5.390e-05  -1.885  0.06018 .
## x4:r         3.153e+00  9.956e-01   3.167  0.00166 **
## x4:theta    -2.093e-03  6.874e-03  -0.304  0.76098
## r:theta      3.903e+00  4.426e+00   0.882  0.37842
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2058 on 392 degrees of freedom
## Multiple R-squared:  0.7389, Adjusted R-squared:  0.7249
## F-statistic: 52.83 on 21 and 392 DF,  p-value: < 2.2e-16
```

```
# do step-wise variable selection
step(lm4,direction="both",trace=FALSE)
```

```
##
## Call:
## lm(formula = log(y) ~ x1 + x2 + x3 + x4 + r + theta + x1:x4 +
##     x2:x3 + x2:x4 + x2:r + x3:x4 + x3:r + x3:theta + x4:r, data = dat)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x4            r
##  -5.426e+02    2.715e-01   -1.109e-02   -4.661e-05    5.577e+01   -2.492e+01
##       theta        x1:x4        x2:x3        x2:x4         x2:r        x3:x4
##   8.269e-02   -2.770e-02    6.413e-06    6.287e-04   -3.801e-01   -4.189e-05
##        x3:r     x3:theta         x4:r
##   2.235e-03   -5.327e-05    2.852e+00
```

```
lm5=lm(log(y)~x1+x2+x3+x4+r+theta+x1:x4+x2:x3+x2:r+x3:x4+x3:r+x3:theta+x4:r,data=dat)
summary(lm5)
```

```
##
## Call:
```

```
## lm(formula = log(y) ~ x1 + x2 + x3 + x4 + r + theta + x1:x4 +
##     x2:x3 + x2:r + x3:x4 + x3:r + x3:theta + x4:r, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71979 -0.10347  0.00277  0.10384  0.97194
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.291e+02  1.252e+02  -4.226 2.94e-05 ***
## x1           2.648e-01  6.219e-02   4.258 2.57e-05 ***
## x2          -7.011e-03  1.416e-03  -4.951 1.09e-06 ***
## x3          -7.148e-05  8.307e-05  -0.860 0.390069
## x4           5.385e+01  2.475e+01   2.176 0.030124 *
## r           -2.200e+01  6.519e+00  -3.375 0.000811 ***
## theta        8.732e-02  2.244e-02   3.891 0.000117 ***
## x1:x4       -2.674e-02  1.229e-02  -2.176 0.030177 *
## x2:x3        6.187e-06  2.467e-06   2.508 0.012531 *
## x2:r        -4.562e-01  1.994e-01  -2.288 0.022641 *
## x3:x4       -3.594e-05  1.283e-05  -2.800 0.005351 **
## x3:r         2.590e-03  9.058e-04   2.859 0.004469 **
## x3:theta    -6.142e-05  2.055e-05  -2.989 0.002970 **
## x4:r         2.505e+00  9.119e-01   2.747 0.006287 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2051 on 400 degrees of freedom
## Multiple R-squared:  0.7355, Adjusted R-squared:  0.7269
## F-statistic: 85.54 on 13 and 400 DF,  p-value: < 2.2e-16
```

```
# drop insignificant predictors
lm6=lm(log(y)~x1+x2+x3+x4+r+theta+x2:x3+x2:r+x3:r+x3:theta,data=dat)
summary(lm6)
```

```
##
## Call:
## lm(formula = log(y) ~ x1 + x2 + x3 + x4 + r + theta + x2:x3 +
##     x2:r + x3:r + x3:theta, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.70506 -0.11044  0.00293  0.10933  0.96676
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.907e+02  7.381e+01  -3.938 9.69e-05 ***
## x1           1.463e-01  3.667e-02   3.990 7.84e-05 ***
## x2          -6.633e-03  1.428e-03  -4.644 4.63e-06 ***
## x3          -2.545e-04  5.583e-05  -4.559 6.84e-06 ***
## x4           2.801e-02  5.509e-03   5.085 5.64e-07 ***
## r           -8.174e+00  4.214e+00  -1.939 0.053149 .
## theta        9.231e-02  2.258e-02   4.088 5.25e-05 ***
## x2:x3        7.868e-06  2.355e-06   3.341 0.000914 ***
## x2:r        -5.946e-01  1.910e-01  -3.113 0.001985 **
```

```
## x3:r         3.362e-03  6.768e-04   4.967 1.01e-06 ***
## x3:theta    -7.889e-05  1.868e-05  -4.222 2.99e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2082 on 403 degrees of freedom
## Multiple R-squared:  0.7253, Adjusted R-squared:  0.7185
## F-statistic: 106.4 on 10 and 403 DF,  p-value: < 2.2e-16
```

```
# try to drop some interactions
lm7=lm(log(y)~x1+x2+x3+x4+x2:x3+x2:r+x3:r+theta+x3:theta,data=dat)
summary(lm7)
```

```
##
## Call:
## lm(formula = log(y) ~ x1 + x2 + x3 + x4 + x2:x3 + x2:r + x3:r +
##     theta + x3:theta, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.72025 -0.11073  0.00565  0.10720  0.96665
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.981e+02  7.397e+01  -4.030 6.66e-05 ***
## x1           1.500e-01  3.674e-02   4.082 5.38e-05 ***
## x2          -5.136e-03  1.206e-03  -4.260 2.55e-05 ***
## x3          -3.381e-04  3.564e-05  -9.485  < 2e-16 ***
## x4           2.872e-02  5.515e-03   5.207 3.07e-07 ***
## theta        8.669e-02  2.247e-02   3.858 0.000133 ***
## x2:x3        1.121e-05  1.610e-06   6.962 1.36e-11 ***
## x2:r        -8.964e-01  1.111e-01  -8.067 8.29e-15 ***
## x3:r         3.296e-03  6.782e-04   4.860 1.69e-06 ***
## x3:theta    -7.947e-05  1.875e-05  -4.239 2.78e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2089 on 404 degrees of freedom
## Multiple R-squared:  0.7227, Adjusted R-squared:  0.7166
## F-statistic:   117 on 9 and 404 DF,  p-value: < 2.2e-16
```

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
lrtest(lm7,lm6) # compare lm6 and lm7 --- lm6 is better
```

```
## Likelihood ratio test
##
## Model 1: log(y) ~ x1 + x2 + x3 + x4 + x2:x3 + x2:r + x3:r + theta + x3:theta
## Model 2: log(y) ~ x1 + x2 + x3 + x4 + r + theta + x2:x3 + x2:r + x3:r +
##     x3:theta
##   #Df LogLik Df Chisq Pr(>Chisq)
## 1  11 65.814
## 2  12 67.737  1 3.846    0.04986 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
vif(lm6) # multicollinearity exists
```

```
##        x1        x2        x3        x4         r     theta     x2:x3      x2:r
##  1.018331  2.521451 47.293269  2.507638 36.495953  3.367712 41.332762 39.128120
##      x3:r  x3:theta
## 21.963585  8.056467
```

```
lm8=lm(log(y)~x1+x2+log(x3)+x4+r,data=dat)
summary(lm8)
```

```
##
## Call:
## lm(formula = log(y) ~ x1 + x2 + log(x3) + x4 + r, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.68927 -0.10562 -0.00026  0.10647  0.99431
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.387e+02  7.548e+01  -4.487 9.39e-06 ***
## x1           1.705e-01  3.750e-02   4.546 7.21e-06 ***
## x2          -6.888e-03  9.385e-04  -7.339 1.18e-12 ***
## log(x3)     -1.042e-01  1.781e-02  -5.849 1.01e-08 ***
## x4           1.304e-02  4.990e-03   2.613  0.00931 **
## r           -1.305e+01  1.245e+00 -10.484  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2122 on 408 degrees of freedom
## Multiple R-squared:  0.7111, Adjusted R-squared:  0.7076
## F-statistic: 200.9 on 5 and 408 DF,  p-value: < 2.2e-16
```

```
vif(lm8) # no multicollinearity
```

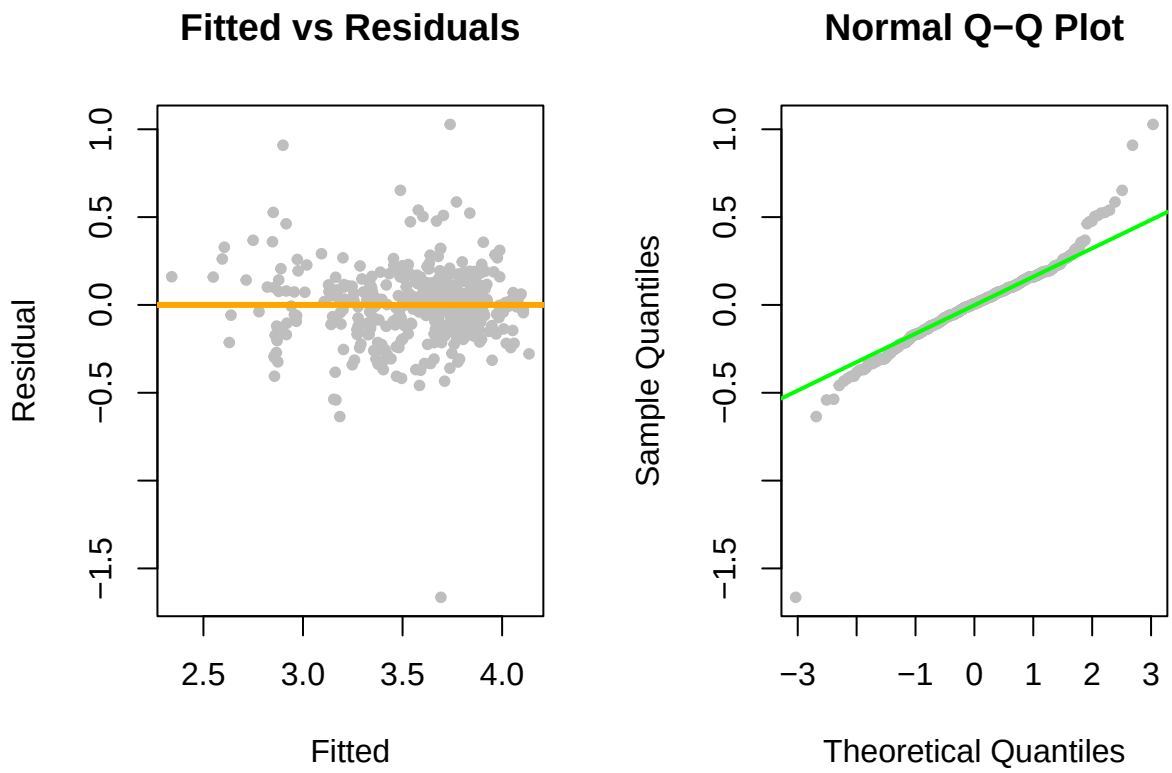
```
##       x1       x2  log(x3)       x4        r
## 1.025534 1.048311 3.646632 1.980900 3.065354
```

```
lm_2=lm8
```

Model Diagnostics

#Model(1)

```
library(zoo)
par(mfrow=c(1,2))
plot(fitted(lm_1),resid(lm_1),xlab='Fitted',ylab='Residual',
     main='Fitted vs Residuals',
     col = "grey",
     pch = 20)
abline(h=0, col = "orange", lwd = 3)
qqnorm(resid(lm_1),col="grey",pch=20)
qqline(resid(lm_1),col="green",lwd=2)
```




```
library(lmtest)
bptest(lm_1) # equal variance holds
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lm_1
## BP = 8.7722, df = 6, p-value = 0.1868
```

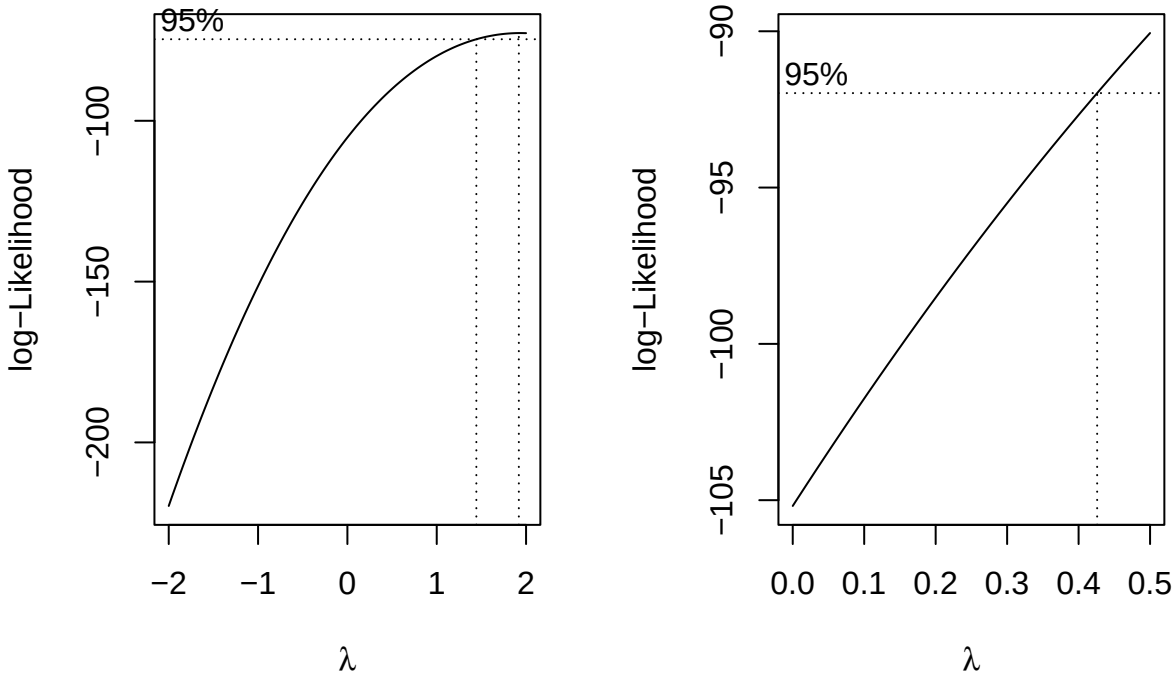
```
shapiro.test(resid(lm_1)) # normality is violated
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(lm_1)
## W = 0.90754, p-value = 3.413e-15
```

```
dwtest(lm_1) # no autocorrelation
```

```
##
##  Durbin-Watson test
##
## data:  lm_1
## DW = 2.1482, p-value = 0.9351
## alternative hypothesis: true autocorrelation is greater than 0
```

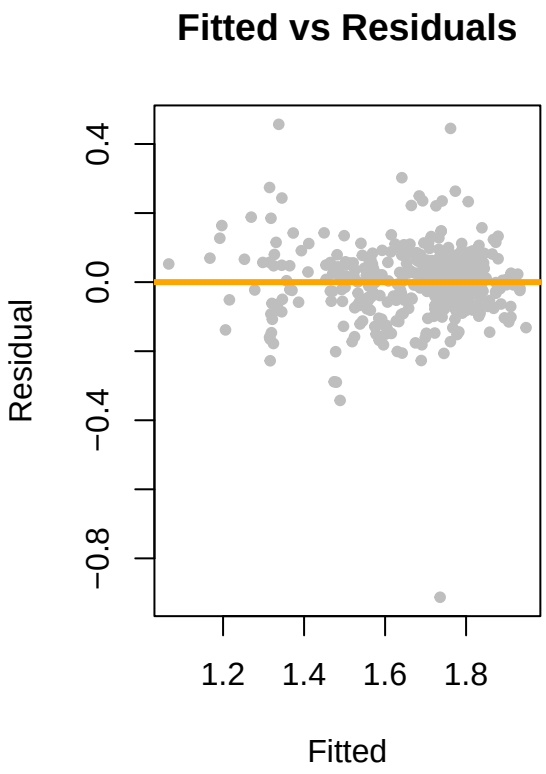
```
# try box-cox transformation
library(MASS)
boxcox(lm_1)
boxcox(lm_1, lambda = seq(0, 0.5, by = 0.05))
```

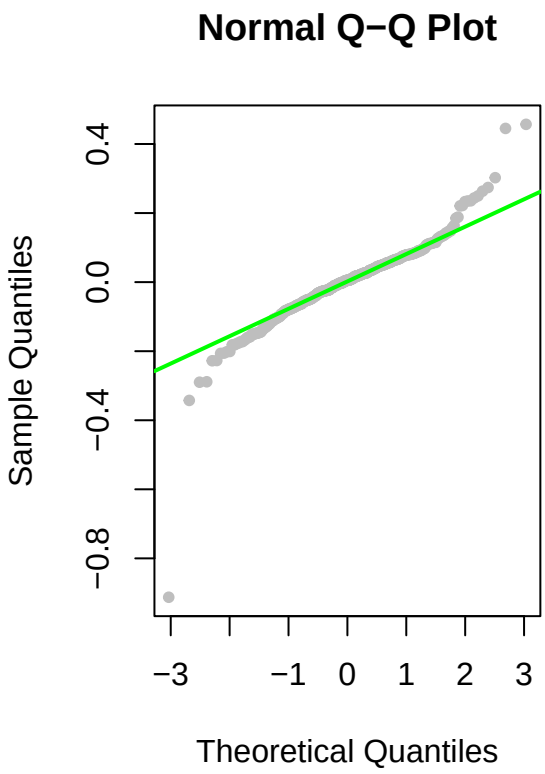


```r
# Let's transform Y using lambda = 0.42
lambda = 0.42
dat_lm_transf=lm(((log(y)^(lambda)-1)/(lambda))~x1+x2+x3+x4+x5+x3:x4,data=dat)
summary(dat_lm_transf)
```

```
##
## Call:
## lm(formula = ((log(y)^(lambda) - 1)/(lambda)) ~ x1 + x2 + x3 +
##     x4 + x5 + x3:x4, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.91253 -0.05151  0.00515  0.05545  0.45686
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.500e+02  3.869e+01  -6.462 2.95e-10 ***
## x1           6.829e-02  1.851e-02   3.689 0.000256 ***
## x2          -3.391e-03  4.581e-04  -7.402 7.77e-13 ***
## x3          -6.013e-05  6.519e-06  -9.225  < 2e-16 ***
## x4           1.751e-02  2.408e-03   7.269 1.87e-12 ***
## x5           4.578e+00  5.393e-01   8.489 3.94e-16 ***
## x3:x4       -1.533e-05  2.961e-06  -5.176 3.57e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1053 on 407 degrees of freedom
## Multiple R-squared:  0.7082, Adjusted R-squared:  0.7039
## F-statistic: 164.6 on 6 and 407 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(1,2))
plot(fitted(dat_lm_transf),resid(dat_lm_transf),
     xlab='Fitted',ylab='Residual',
     main='Fitted vs Residuals',
     col = "grey",
     pch = 20)
abline(h=0, col = "orange", lwd = 3)
qqnorm(resid(dat_lm_transf),col="grey",pch=20)
qqline(resid(dat_lm_transf),col="green",lwd=2)
```






```
bptest(dat_lm_transf)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  dat_lm_transf
## BP = 8.47, df = 6, p-value = 0.2056
```

```
shapiro.test(resid(dat_lm_transf))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(dat_lm_transf)
## W = 0.88545, p-value < 2.2e-16
```

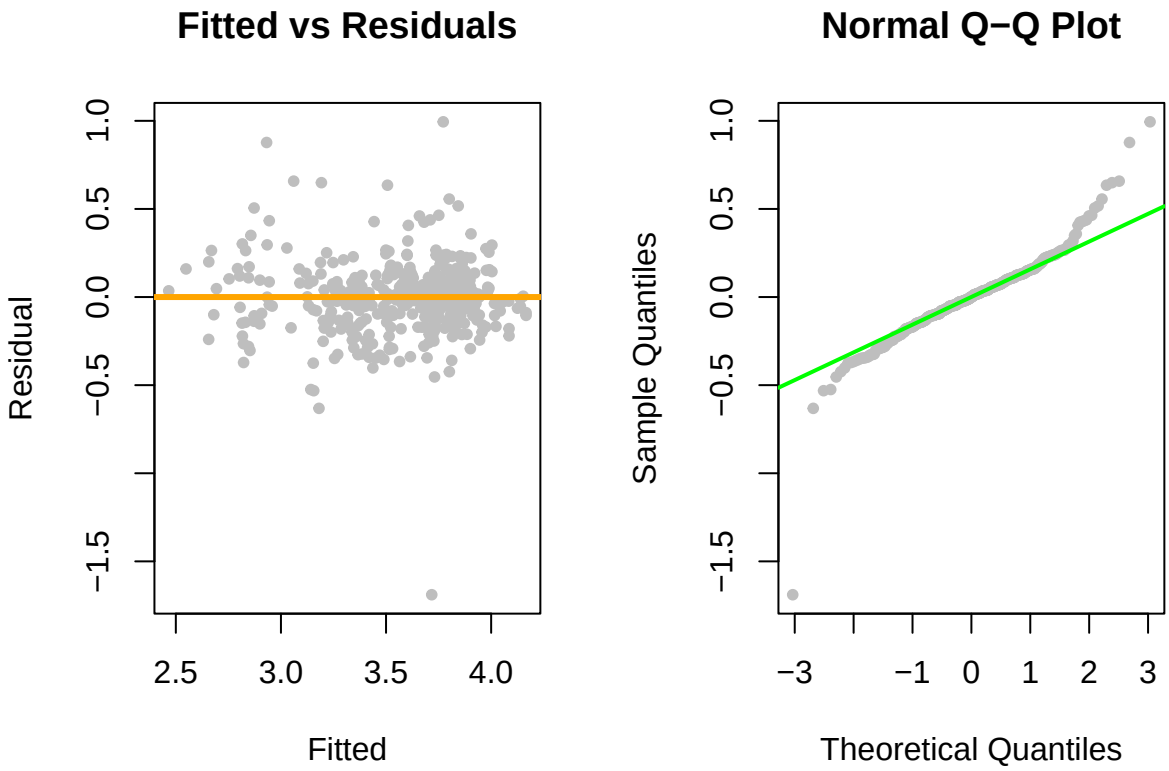
```
dwtest(dat_lm_transf)
```

```
##
##  Durbin-Watson test
##
## data:  dat_lm_transf
## DW = 2.135, p-value = 0.9164
## alternative hypothesis: true autocorrelation is greater than 0
```

```
# The Box-Cox transformation doesn't help to correct normality
```

#Model(2)

```
par(mfrow=c(1,2))
plot(fitted(lm_2),resid(lm_2),xlab='Fitted',ylab='Residual',
     main='Fitted vs Residuals',
     col = "grey",
     pch = 20)
abline(h=0, col = "orange", lwd = 3)
qqnorm(resid(lm_2),col="grey",pch=20)
qqline(resid(lm_2),col="green",lwd=2)
```




```
library(lmtest)
bptest(lm_2) # equal variance holds
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lm_2
## BP = 6.4616, df = 5, p-value = 0.2639
```

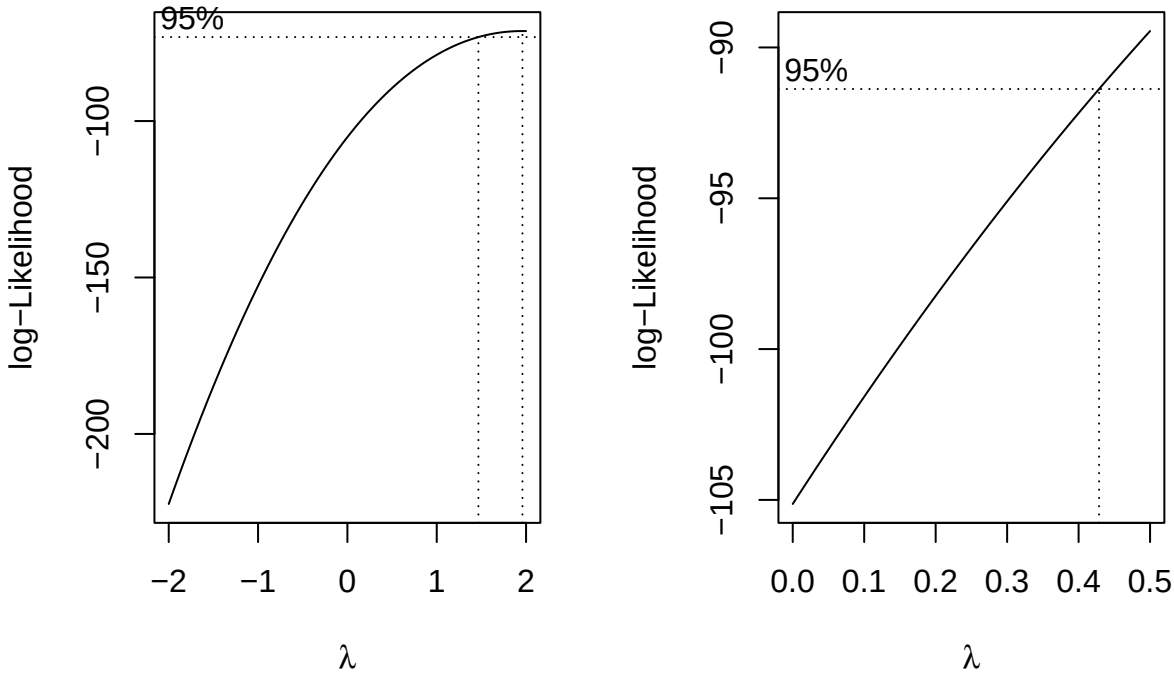
```
shapiro.test(resid(lm_2)) # normality is violated
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(lm_2)
## W = 0.9037, p-value = 1.577e-15
```

```
dwtest(lm_2) # no autocorrelation
```

```
##
##  Durbin-Watson test
##
## data:  lm_2
## DW = 2.1192, p-value = 0.89
## alternative hypothesis: true autocorrelation is greater than 0
```

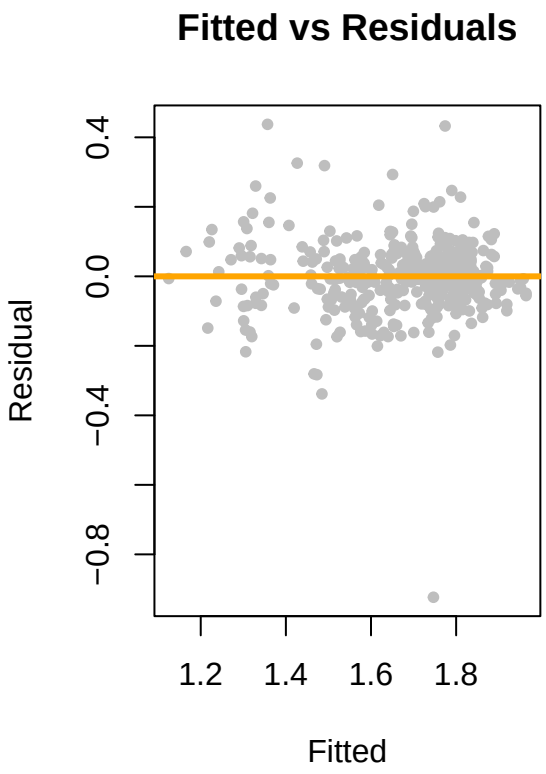
```
# try box-cox transformation
library(MASS)
boxcox(lm_2)
boxcox(lm_2, lambda = seq(0, 0.5, by = 0.05))
```

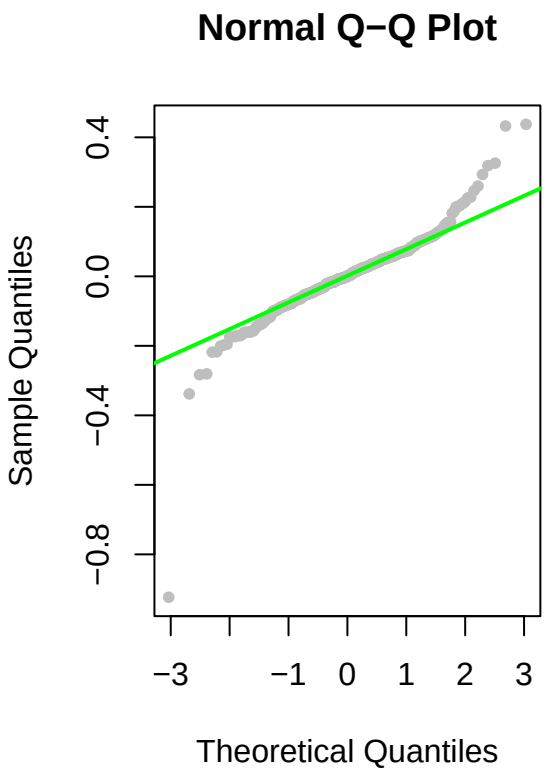


```r
# Let's transform Y using lambda = 0.42
lambda = 0.42
dat_lm_transf2=lm(((log(y)^(lambda)-1)/(lambda))~x1+x2+log(x3)+x4+r,data=dat)
summary(dat_lm_transf2)
```

```
##
## Call:
## lm(formula = ((log(y)^(lambda) - 1)/(lambda)) ~ x1 + x2 + log(x3) +
##     x4 + r, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.92337 -0.05012  0.00047  0.05334  0.43733
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.624e+02  3.737e+01  -4.345 1.76e-05 ***
## x1           8.172e-02  1.857e-02   4.401 1.38e-05 ***
## x2          -3.349e-03  4.647e-04  -7.206 2.81e-12 ***
## log(x3)     -4.717e-02  8.819e-03  -5.349 1.48e-07 ***
## x4           6.177e-03  2.471e-03   2.500   0.0128 *
## r           -6.784e+00  6.164e-01 -11.007  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1051 on 408 degrees of freedom
## Multiple R-squared:  0.7089, Adjusted R-squared:  0.7053
## F-statistic: 198.7 on 5 and 408 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(1,2))
plot(fitted(dat_lm_transf2),resid(dat_lm_transf2),
     xlab='Fitted',ylab='Residual',
     main='Fitted vs Residuals',
     col = "grey",
     pch = 20)
abline(h=0, col = "orange", lwd = 3)
qqnorm(resid(dat_lm_transf2),col="grey",pch=20)
qqline(resid(dat_lm_transf2),col="green",lwd=2)
```






```
library(lmtest)
bptest(dat_lm_transf2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  dat_lm_transf2
## BP = 5.5304, df = 5, p-value = 0.3546
```

```
shapiro.test(resid(dat_lm_transf2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(dat_lm_transf2)
## W = 0.88297, p-value < 2.2e-16
```

```
dwtest(dat_lm_transf2)
```

```
##
##  Durbin-Watson test
##
## data:  dat_lm_transf2
## DW = 2.1081, p-value = 0.8671
## alternative hypothesis: true autocorrelation is greater than 0
```

```
# The Box-Cox transformation doesn't help
```

Outliers,Influential Points

```
sum(abs(rstandard(lm_1)) > 2)
```

```
## [1] 18
```

```
sum(cooks.distance(lm_1) > 4 / length(cooks.distance(lm_1)))
```

```
## [1] 28
```

```
sum(abs(rstandard(lm_2)) > 2)
```

```
## [1] 20
```

```
sum(cooks.distance(lm_2) > 4 / length(cooks.distance(lm_2)))
```

```
## [1] 24
```

Evaluation

```
library(lmtest)
lrtest(lm_2,lm_1)
```

```
## Likelihood ratio test
##
## Model 1: log(y) ~ x1 + x2 + log(x3) + x4 + r
## Model 2: log(y) ~ x1 + x2 + x3 + x4 + x5 + x3:x4
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1   7 57.319
## 2   8 56.291  1 2.0545     0.1518
```

```
# p-value=0.1518 is large, we have no evidence to say lm_1 is better
AIC(lm_1,lm_2)
```

```
##      df        AIC
## lm_1  8  -96.58275
## lm_2  7 -100.63721
```

```
BIC(lm_1,lm_2)
```

```
##      df       BIC
## lm_1  8 -64.37583
## lm_2  7 -72.45615
```

```
summary(lm_1)$adj.r.square
```

```
## [1] 0.7054138
```

```
summary(lm_2)$adj.r.square
```

```
## [1] 0.7075905
```

```
coef(lm_2)
```

```
##   (Intercept)            x1            x2       log(x3)            x4
## -3.387203e+02  1.704993e-01 -6.887733e-03 -1.041800e-01  1.303711e-02
##             r
## -1.305119e+01
```

```
exp(coef(lm_2))
```

```
##   (Intercept)            x1            x2       log(x3)            x4
## 7.863642e-148  1.185897e+00  9.931359e-01  9.010631e-01  1.013122e+00
##             r
##  2.147540e-06
```

Furthur discussion

```
# try to use different thresholds for choosing origin of model(2)
center_x=mean(dat[which(dat$y>0),]$x6)
center_y=mean(dat[which(dat$y>0),]$x5)
dat$r=sqrt((dat$x6-center_x)^2+(dat$x5-center_y)^2)
dat$theta=atan((dat$x6-center_x)/(dat$x5-center_y))
lm_2_1=lm(log(y)~x1+x2+log(x3)+x4+r,data=dat)
lrtest(lm_2_1,lm_1)
```

```
## Likelihood ratio test
##
## Model 1: log(y) ~ x1 + x2 + log(x3) + x4 + r
## Model 2: log(y) ~ x1 + x2 + x3 + x4 + x5 + x3:x4
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1   7 21.995
## 2   8 56.291  1 68.592  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# lm_1 is better, p-value is very small

center_x=mean(dat[which(dat$y>50),]$x6)
center_y=mean(dat[which(dat$y>50),]$x5)
dat$r=sqrt((dat$x6-center_x)^2+(dat$x5-center_y)^2)
dat$theta=atan((dat$x6-center_x)/(dat$x5-center_y))
lm_2_2=lm(log(y)~x1+x2+log(x3)+x4+r,data=dat)
lrtest(lm_2_2,lm_1)
```

```
## Likelihood ratio test
##
## Model 1: log(y) ~ x1 + x2 + log(x3) + x4 + r
## Model 2: log(y) ~ x1 + x2 + x3 + x4 + x5 + x3:x4
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1   7 52.093
## 2   8 56.291  1 8.3968   0.003759 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# lm_1 is better, but p-value is larger

lrtest(lm_2,lm_1)
```

```
## Likelihood ratio test
##
## Model 1: log(y) ~ x1 + x2 + log(x3) + x4 + r
## Model 2: log(y) ~ x1 + x2 + x3 + x4 + x5 + x3:x4
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1   7 57.319
## 2   8 56.291  1 2.0545     0.1518
```

```
# p-value is large, lm_2 is better

center_x=mean(dat[which(dat$y>80),]$x6)
center_y=mean(dat[which(dat$y>80),]$x5)
dat$r=sqrt((dat$x6-center_x)^2+(dat$x5-center_y)^2)
dat$theta=atan((dat$x6-center_x)/(dat$x5-center_y))
lm_2_3=lm(log(y)~x1+x2+log(x3)+x4+r,data=dat)
lrtest(lm_2_3,lm_1)
```

```
## Likelihood ratio test
##
## Model 1: log(y) ~ x1 + x2 + log(x3) + x4 + r
## Model 2: log(y) ~ x1 + x2 + x3 + x4 + x5 + x3:x4
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1   7 39.448
## 2   8 56.291  1 33.687  6.472e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# lm_1 is better, p-value is very small
```