**REPORT**



|  |  |  |
| --- | --- | --- |
| 수강과목 | : | 회귀분석(II) |
| 담당교수 | : | 김충락 |
| 학 과 | : | 통계학과 |
| 학 번 | : | 201611539 |
| 이 름 | : | 하성진 |
| 제출일자 | : | 2019.10.28 |

**Regression Analysis (II) Project 1.**

Due October 28, 2019

201611539 HASEONGJIN

**1.Make your own dataset based on data in Example 6.5 (p. 251). Let X1 ← X1 + , X2 ← X2 + , Y ← Y + , where ∼ N(0, 1). Do the response surface analysis with contour plot.**

First, make dataset based on text book and change the datset according to the Problem ( ∼ N(0, 1) )

*> x1 <- c(4,20,12,12,12,12,12,6.3,6.3,17.7,17.7)*

*> x2 <- c(250,250,250,250,220,280,250,229,271,229,271)*

*> y <- c(83.8,81.7,82.4,82.9,84.7,67.9,81.2,81.3,83.1,85.3,72.7)*

*> x1 <- x1 + rnorm(11)*

*> x2 <- x2 + rnorm(11)*

*> y <- y + rnorm(11)*

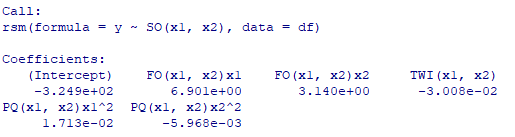
*> df <- data.frame(x1,x2,y)*

Based on results, create matrix for fitted model

*>install.packages('rsm')*

*>library(rsm)*

*>rsm(y ~ SO(x1, x2), data=df)*



* Y = -324.9 + 6.901\*X1 + 3.140\*X2 – 0.03008\*X1\*X2 +0.0173\*X1^2 -0.005968\*X2^2

*> <- matrix(c(6.901,3.140),2,1)*

*>n <- matrix(c(0.0173,-0.01504,-0.01504,-0.005968),2,2)*

*>n = solve(n)*

Calculate stationary point

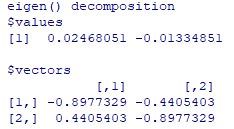
>*(-1/2)\*(n)%\*%(m)*



Calculate eigenvector estimate

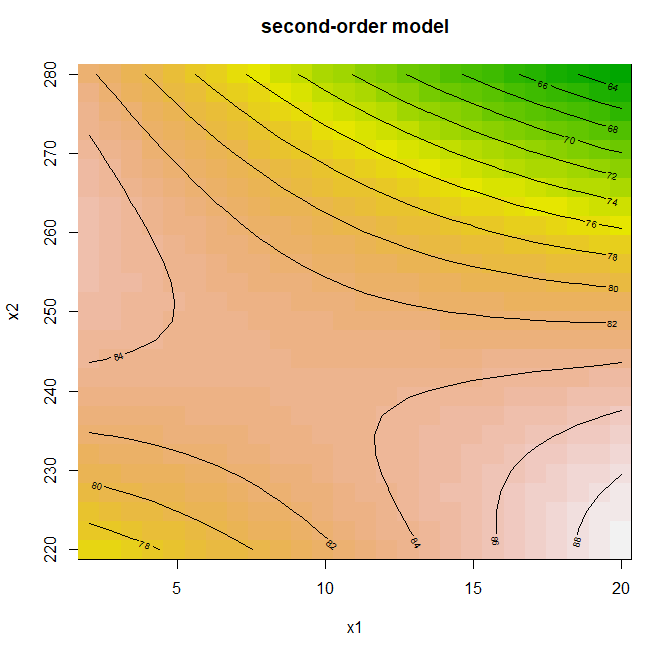
*> n =solve(n)*

*> eigen(n)*



Do response surface analysis with contour plot

*> contour(formula, ~ x1 + x2, image = TRUE, main="second-order model"*



.

It shows that stationary point is saddle point.

According to the graph, when x1(공정시간) increases, y(효율) increases, and when x2(공정온도) increases, y(효율) decreases

**2. Make your own dataset based on data in Example 6.6 (p. 254). Let X ← X1 + , Y ← Y + , where ∼ N(0, 0.1 2 ). Compute the WLSE using the same method as the one in text.**

First, make dataset based on text book and change the datset according to the Problem ( ∼ N(0, 1) )

*> x <- c(1.15,1.90,3,3,3,3,3,5.34,5.38,5.4,5.4,5.45,7.7,7.8,7.81,7.85,7.87,7.91,7.94,9.03,9.07,9.11,9.14,9.16,9.37,10.17,10.18,10.22,10.22,10.22,10.18,10.50,10.23,10.03,10.23)*

*> y <- c(0.99,0.98,2.60,2.67,2.66,2.78,2.80,5.92,5.35,4.33,4.89,5.21,7.68,9.81,6.52,9.71,9.82,9.81,8.5,9.47,11.45,12.14,11.5,10.65,10.64,9.78,12.39,11.03,8.00,11.9,8.68,7.25,13.46,10.19,9.93)*

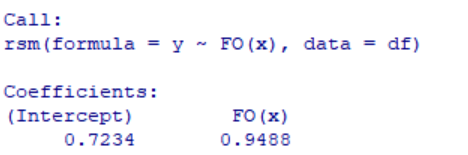
*> x <- x + rnorm(35)*

*> y <- y + rnorm(35)*

*> df <- data.frame(x,y)*

Based on data, create fitted model and check homoscedasticity

*> rsm(y ~ FO(x), data=df)*



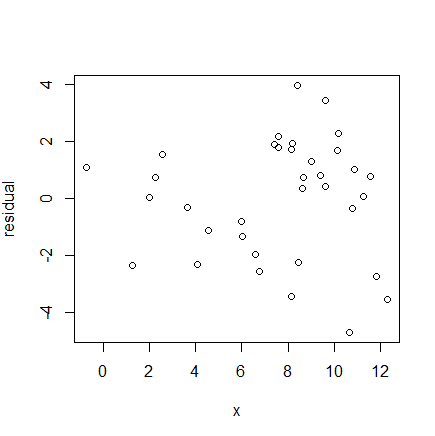
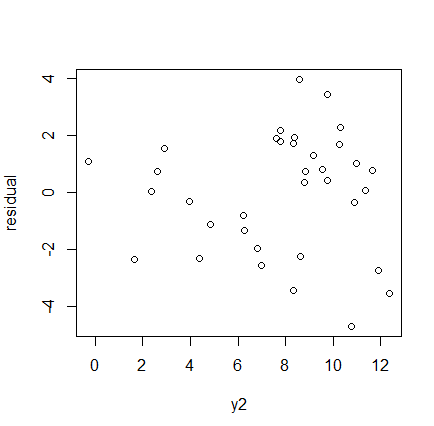
*> z <- function(x) 0.7234+0.9488\*x*

*> y2 <- z(x)*

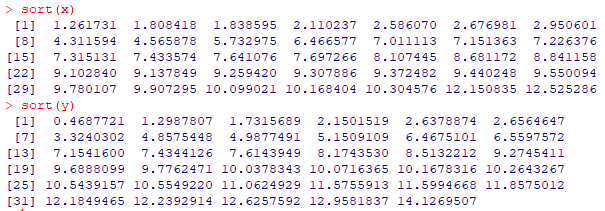
*> residual <- y - y2*

*> plot(formula=residual~y2)*

*> plot(formula=residual~x)*

Based on result, it does not satisfy homoscedasticity. We need to use WLSE method



Create 5 groups based on data and calculate mean and var for each group

>*g1x <- c(sort(x)[1:4])*

*>g2x <- c(sort(x)[5:8])*

*>g3x <- c(sort(x)[9:14])*

*>g4x <- c(sort(x)[15:21])*

*>g5x <- c(sort(x)[22:35])*

*>g1y <-c(sort(y)[1:4])*

*>g2y <-c(sort(y)[5:8])*

*>g3y <-c(sort(y)[9:14])*

*>g4y <-c(sort(y)[15:21])*

*>g5y <-c(sort(y)[22:35])*

*> meanx <- c(mean(g1x), mean(g2x), mean(g3x), mean(g4x), mean(g5x))*

*> vary <- c(var(g1y), var(g2y), var(g3y), var(g4y), var(g5y))*

*> meanx*

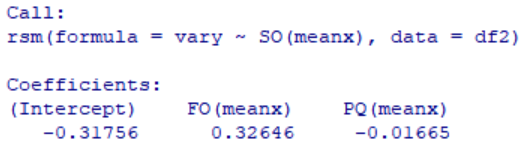
*> vary*

[1]

Based on data, create fitted model

>*df2 <- data.frame(meanx,vary)*

*>rsm(vary ~ SO(meanx), data=df2)*



>f <- function(x) -0.01665\*x^2+0.32646\*x -0.31756

Calculate weighted estimates

*>wi <- (f(df[,1])) ^(-1)*

*>s2 <- f(df[,1])*

*>df3 <- data.frame(x,s2,wi)*

Based on weighted estimates, apply WLSE

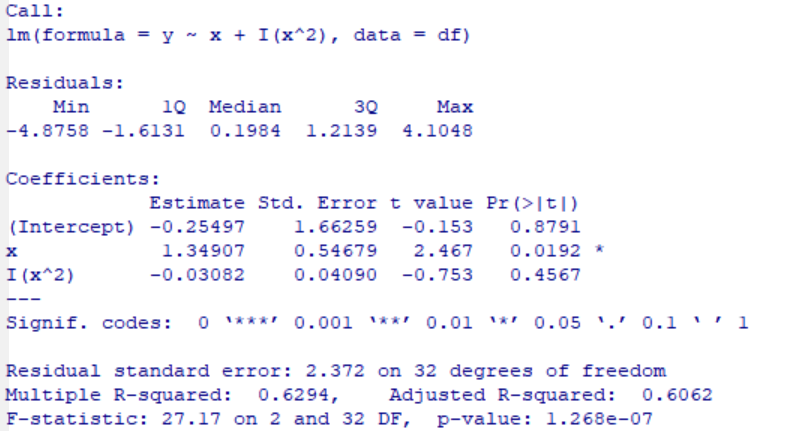
>wls <- lm(formula = y ~ x, weights = wi)

>anova(lm(wls,df))

To check homoscedasticity, do weighted regression analysis first

*> fit.wls <- lm(y~x+I(x^2),df)*

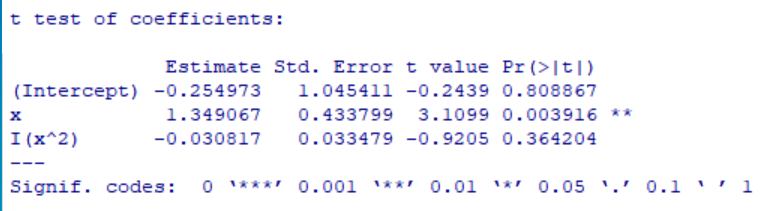
*> summary(fit.wls)*



And then, do white method to compare

*> HCO <- vcovHC(fit.wls, type="HC0")*

*> lmtest::coeftest(fit.wls,vcov=HCO)*



It shows that both std. error is similar, which means the problem of homoscedasticity is solved

**3. Make your own dataset based on data in Example 6.7 (p. 259). Let Y ← Y + , where ~ ∼ N(0,1)**

**(1) Fit the data to the multiple linear regression model**

First, make dataset based on text book and change the datset according to the Problem ( ∼ N(0, 1) )

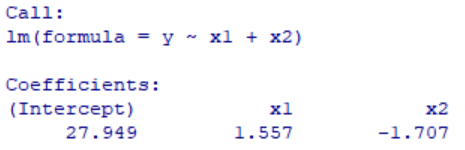
x1 <- c(0,0,0,12,12,12,12,24,24,24,24,36,36,36,36,48,48,48,48,60,60,60,60)

x2 <- c(0,10,20,0,10,20,30,0,10,20,30,0,10,20,30,0,10,20,30,0,10,20,30)

y <- c(26,17,13,38,26,20,15,50,37,27,22,76,53,37,27,108,83,57,41,157,124,87,63)

y <- y + rnorm(23)

n=lm(y~x1+x2)



=> y=27.949+1.557\*x1-1.707\*x2

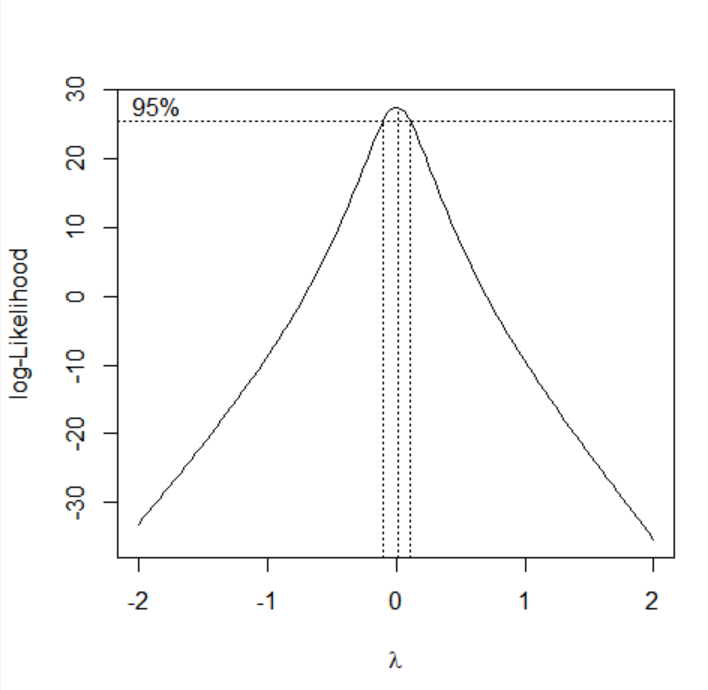
**(2) Fit the data to the Box-Cox transformation model.**

check the value of l()

boxcox(m)

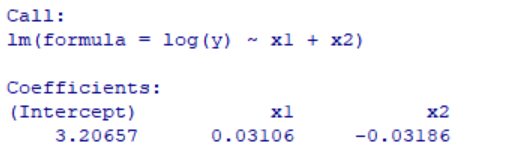
m=lm(y~x1+x2)

boxcox(m)



Because it shows that l()is largest when is 0, we need to use log transform

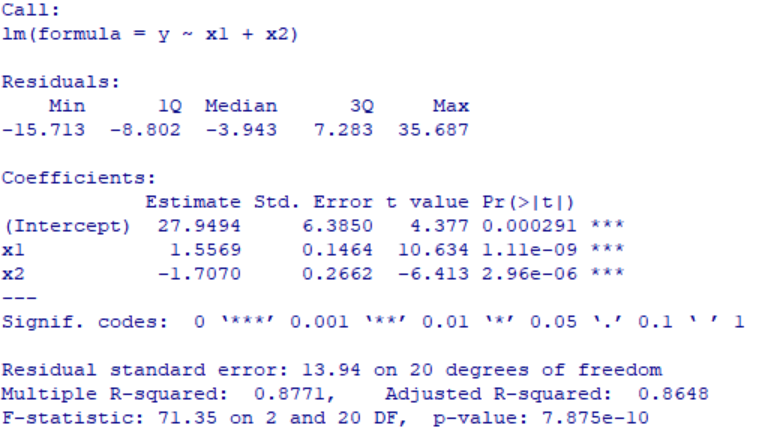
n=lm(log(y)~x1+x2)



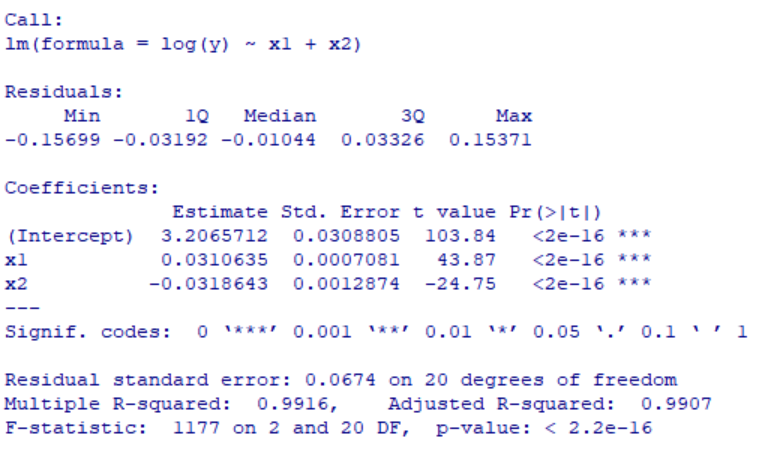
=>logy=3.20657+0.03106\*x1-0.03186\*x2

**(3) Compare two models in (10 and (2) by using the Q−Q plot of residuals in each model.**

summary(m)



summary(n)



f1 <- function(x1,x2) 27.949+1.557\*x1-1.707\*x2

f2 <- function(x1,x2) 3.20657+0.03106\*x1-0.03186\*x2

f1 <- function(x1,x2) 27.949+1.567\*x1-1.729\*x2

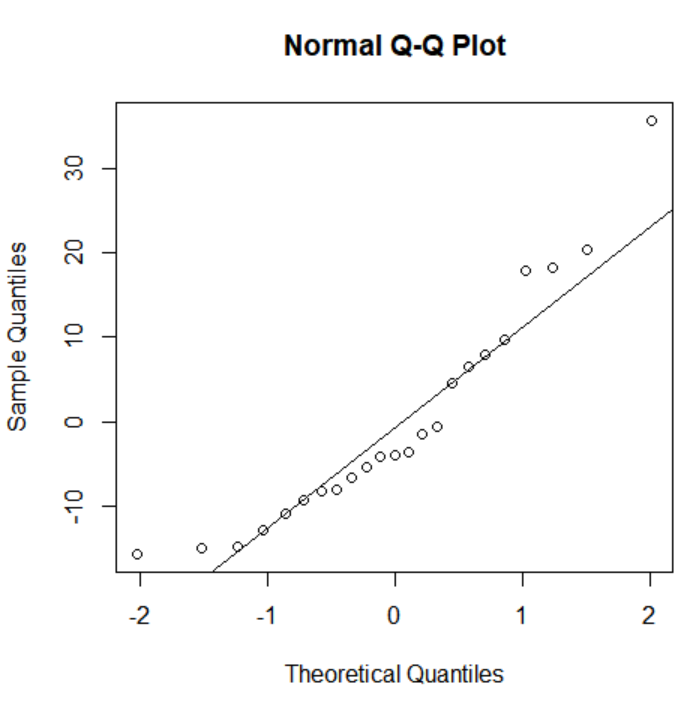
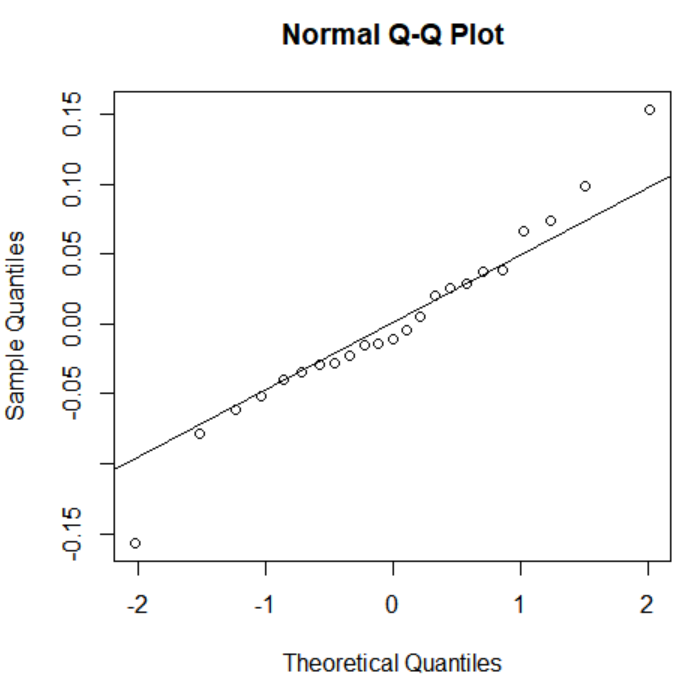
f2 <- function(x1,x2) 3.20622+0.03118\*x1-0.03259\*x2

qqnorm( y - f1(x1,x2) )

qqline( y - f1(x1,x2) )

qqnorm( log(y) - f2(x1,x2))

qqline( log(y) - f2(x1,x2))

It shows that log transformed version is far better fitted.