Homework 3

Juwon Lee, Economics and Statistics, UCLA

2023-01-31

tinytex::install_tinytex()

1.

(a).

Based on the output for model (3.7) a business analyst concluded the following:

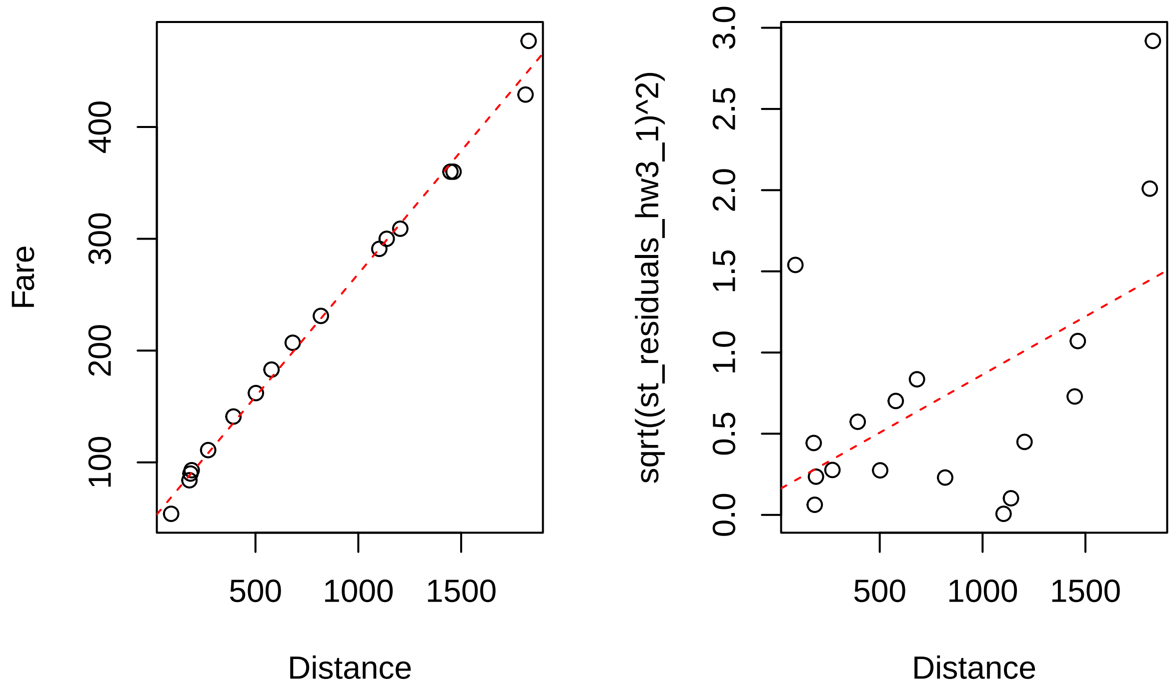
The regression coefficient of the predictor variable, Distance is highly statistically significant and the model explains 99.4% of the variability in the Y-variable, Fare. Thus model (1) is a highly effective model for both understanding the effects of Distance on Fare and for predicting future values of Fare given the value of the predictor variable, Distance.

```
airfare <- read.table("/Users/user/Desktop/Yonsei/Junior/3-2/Introduction to Data Analysis and Regressi

attach(airfare)

lm_data_hw3_1 <- lm(Fare~Distance)
summary(lm_data_hw3_1)
```

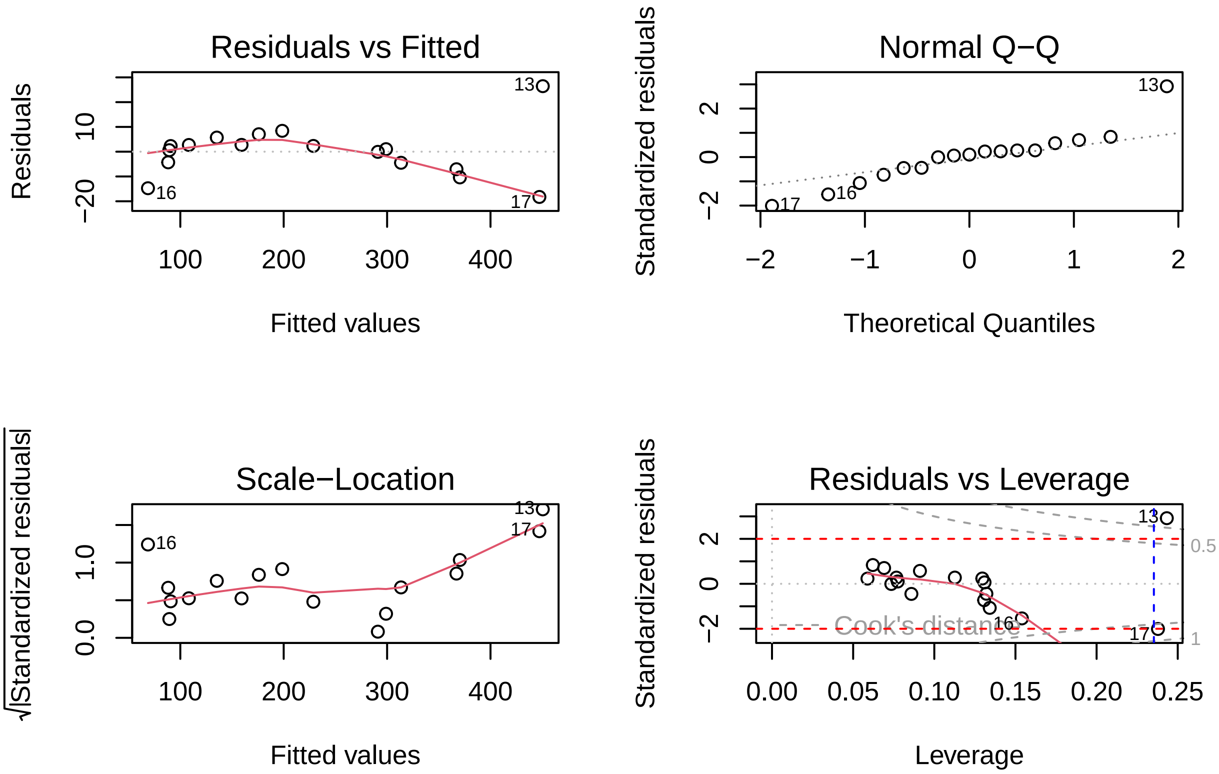
```
##
## Call:
## lm(formula = Fare ~ Distance)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.265  -4.475   1.024   2.745  26.440
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 48.971770   4.405493   11.12 1.22e-08 ***
## Distance     0.219687   0.004421   49.69  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.41 on 15 degrees of freedom
## Multiple R-squared:  0.994,  Adjusted R-squared:  0.9936
## F-statistic:  2469 on 1 and 15 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(1,2))
plot(Distance, Fare)
abline(lm_data_hw3_1, col='red', lty='dashed')

st_residuals_hw3_1 <- rstandard(lm_data_hw3_1)
```

```r
plot(Distance, sqrt((st_residuals_hw3_1)^2))
abline(lsfit(Distance, sqrt((st_residuals_hw3_1)^2)), col='red', lty='dashed')
```




```r
par(mfrow=c(2,2))
plot(lm_data_hw3_1)
abline(-2,0, col='red', lty='dashed')
abline(2,0, col='red', lty='dashed')
abline(v=4/(length(airfare$City)), col='blue', lty='dashed')
```



The model is pretty nice, but they have two bad leverage points, whose also having big Cook's distance, too.

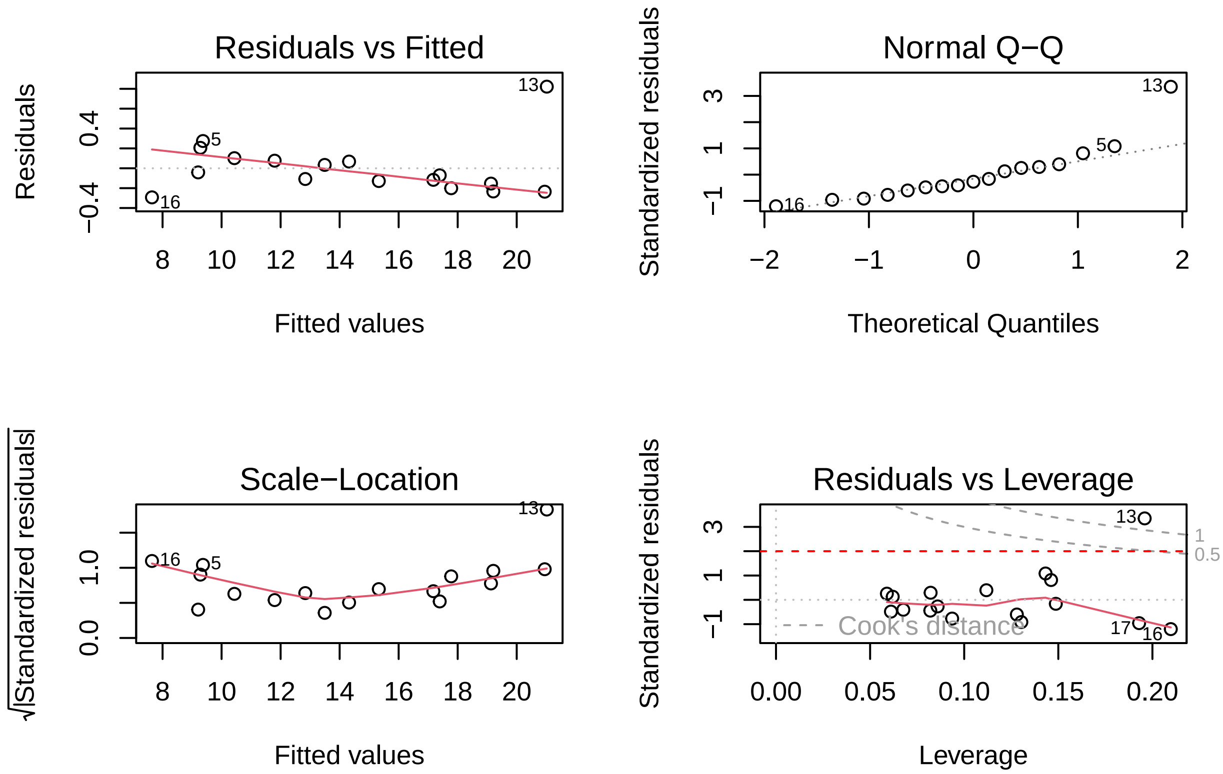
(b)

First, we can transform to $\sqrt{y} = \beta_0 + \beta_1\sqrt{x} + \varepsilon$.

```
sqrtDistance_hw3_1 <- sqrt(airfare$Distance)
sqrtFare_hw3_1 <- sqrt(airfare$Fare)

lm_data_sqrt_hw3_1 <- lm(sqrtFare_hw3_1~sqrtDistance_hw3_1)
summary(lm_data_sqrt_hw3_1)
```

```
##
## Call:
## lm(formula = sqrtFare_hw3_1 ~ sqrtDistance_hw3_1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29276 -0.15416 -0.07086  0.07634  0.82157
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3.82647    0.17830   21.46 1.13e-12 ***
## sqrtDistance_hw3_1  0.40211    0.00624   64.44  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2736 on 15 degrees of freedom
## Multiple R-squared:  0.9964, Adjusted R-squared:  0.9962
## F-statistic:  4153 on 1 and 15 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(2,2))
plot(lm_data_sqrt_hw3_1)
abline(-2,0, col='red', lty='dashed')
abline(2,0, col='red', lty='dashed')
abline(v=4/(length(airfare$City)), col='blue', lty='dashed')
```



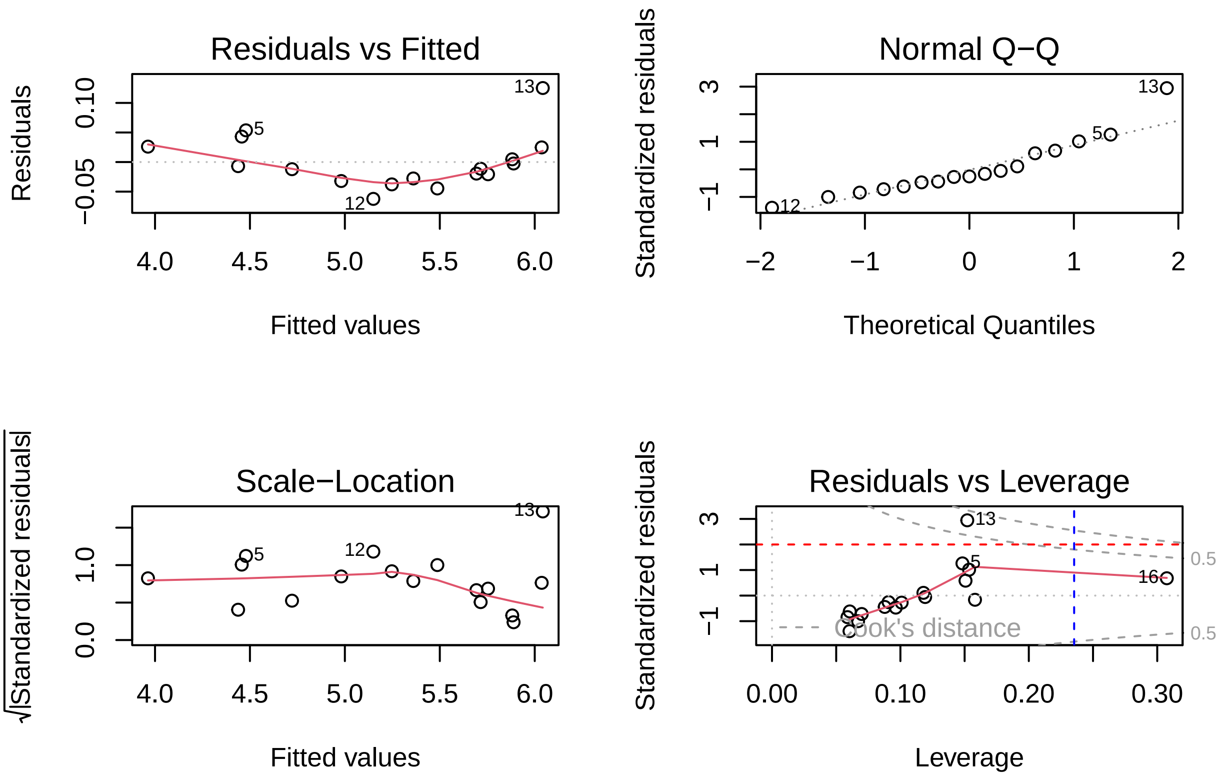
Or, we can also use $\log y = \beta_0 + \beta_1 \log x + \varepsilon$.

```r
logDistance_hw3_1 <- log(airfare$Distance)
logFare_hw3_1 <- log(airfare$Fare)

lm_data_log_hw3_1 <- lm(logFare_hw3_1~logDistance_hw3_1)
summary(lm_data_log_hw3_1)
```

```
##
## Call:
## lm(formula = logFare_hw3_1 ~ logDistance_hw3_1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.06228 -0.02776 -0.01128  0.02479  0.12515
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.85546    0.07925   10.79 1.81e-08 ***
## logDistance_hw3_1  0.69058    0.01232   56.05  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04615 on 15 degrees of freedom
```

```
## Multiple R-squared:  0.9952, Adjusted R-squared:  0.9949
## F-statistic:  3141 on 1 and 15 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm_data_log_hw3_1)
abline(-2,0, col='red', lty='dashed')
abline(2,0, col='red', lty='dashed')
abline(v=4/(length(airfare$City)), col='blue', lty='dashed')
```



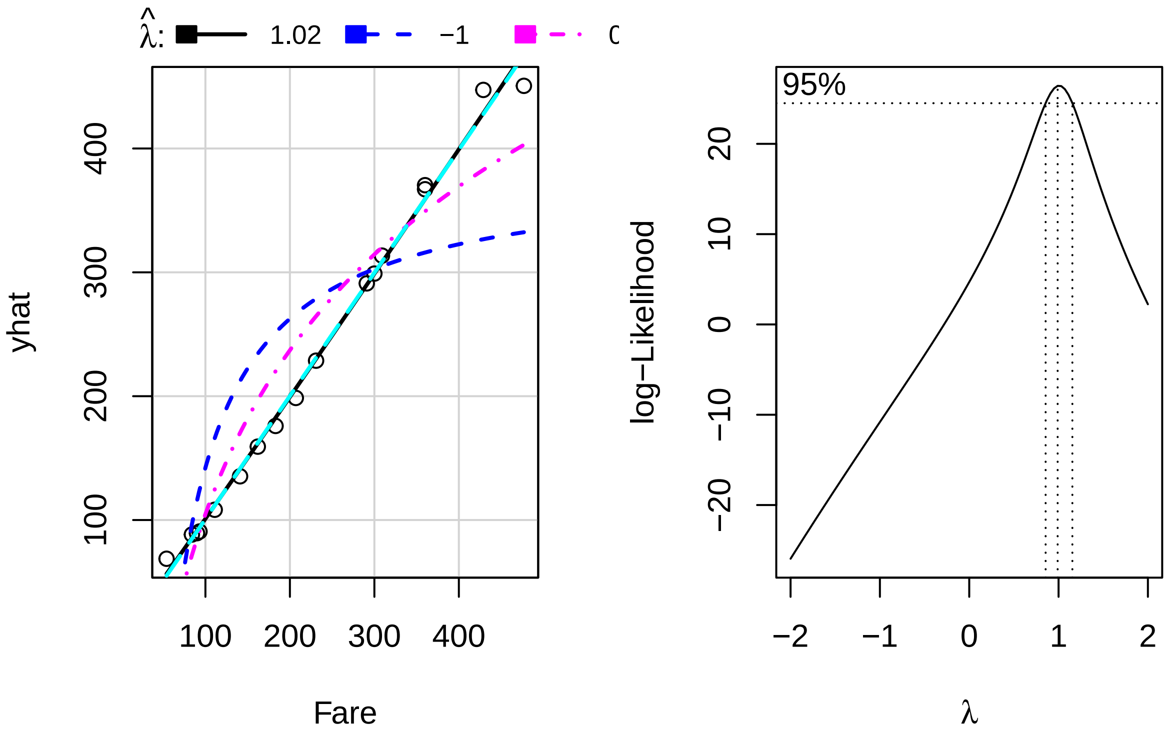

```
library(car)
```

```
## Loading required package: carData
```

```
library(MASS)

par(mfrow=c(1,2))
inverseResponsePlot(lm_data_hw3_1, key=TRUE)
```

```
##      lambda       RSS
## 1  1.024061  1605.994
## 2 -1.000000 81066.642
## 3  0.000000 22925.898
## 4  1.000000  1616.388
```

```
boxcox(lm_data_hw3_1)
```

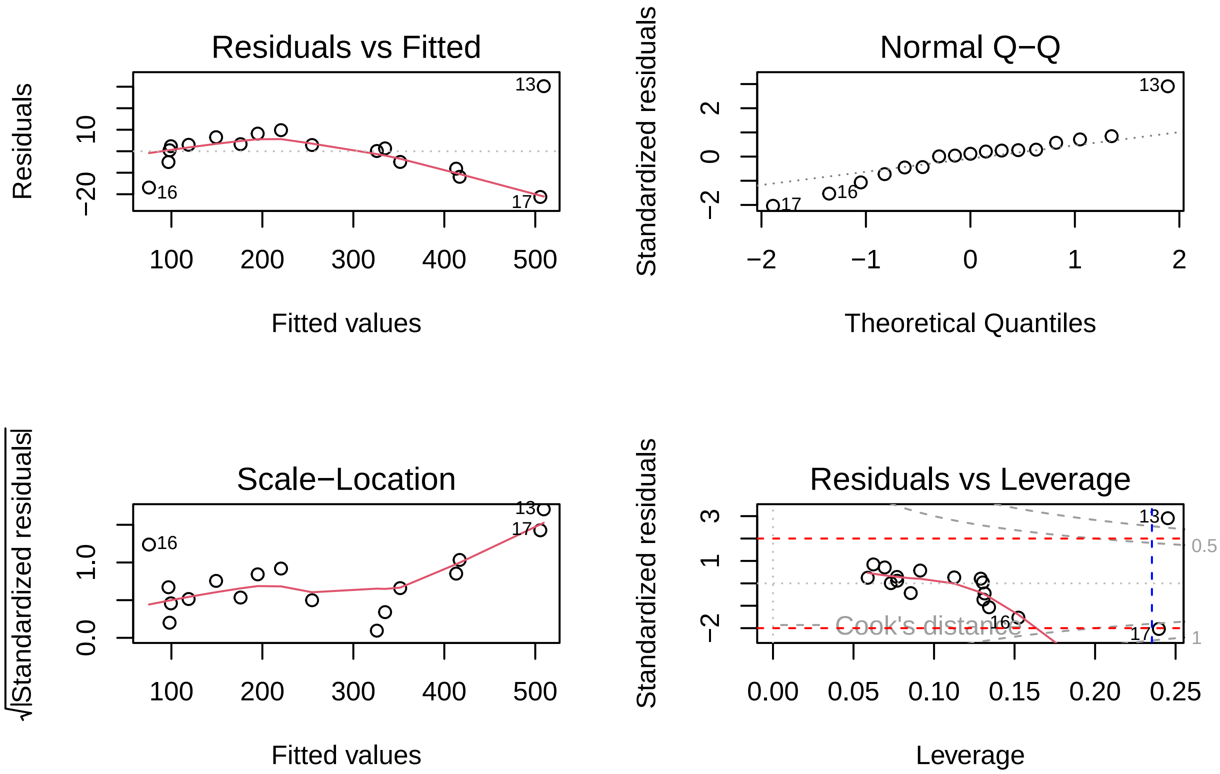


The inverse response plot shows that $\lambda = 1.02$ would produce best result.

```r
improveDistance_hw3_1 <- (airfare$Distance)^(1.02)
improveFare_hw3_1 <- (airfare$Fare)^(1.02)

lm_data_improve_hw3_1 <- lm(improveFare_hw3_1~improveDistance_hw3_1)
summary(lm_data_improve_hw3_1)
```

```
##
## Call:
## lm(formula = improveFare_hw3_1 ~ improveDistance_hw3_1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.264  -5.032   1.338   3.289  30.257
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           54.27614    5.01897   10.81 1.77e-08 ***
## improveDistance_hw3_1  0.21423    0.00436   49.13  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.98 on 15 degrees of freedom
## Multiple R-squared:  0.9938, Adjusted R-squared:  0.9934
## F-statistic:  2414 on 1 and 15 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(2,2))
plot(lm_data_improve_hw3_1)
abline(-2,0, col='red', lty='dashed')
abline(2,0, col='red', lty='dashed')
abline(v=4/(length(airfare$City)), col='blue', lty='dashed')
```



2.

An analyst for the auto industry has asked for your help in modeling data on the prices of new cars. Interest centers on modeling suggested retail price as a function of the cost to the dealer for 234 new cars. The data set, which is available on the book website in the file cars04.csv, is a subset of the data from http://www.amstat.org/publications/jse/datasets/04cars.txt

The first model to fit to the data was
Suggested Retail Price $= \beta_0 + \beta_1 *$ Dealer Cost $+ e$.

(a)

Based on the output for model, the analyst concluded the following:

Since the model explains just more than 99.8% of the variabilty in Suggested Retail Price and the coefficient of Dealer Cost has a t-value greater than 412, model (1) is a highly effective model for producting prediction intervals for Suggested Retail Price.

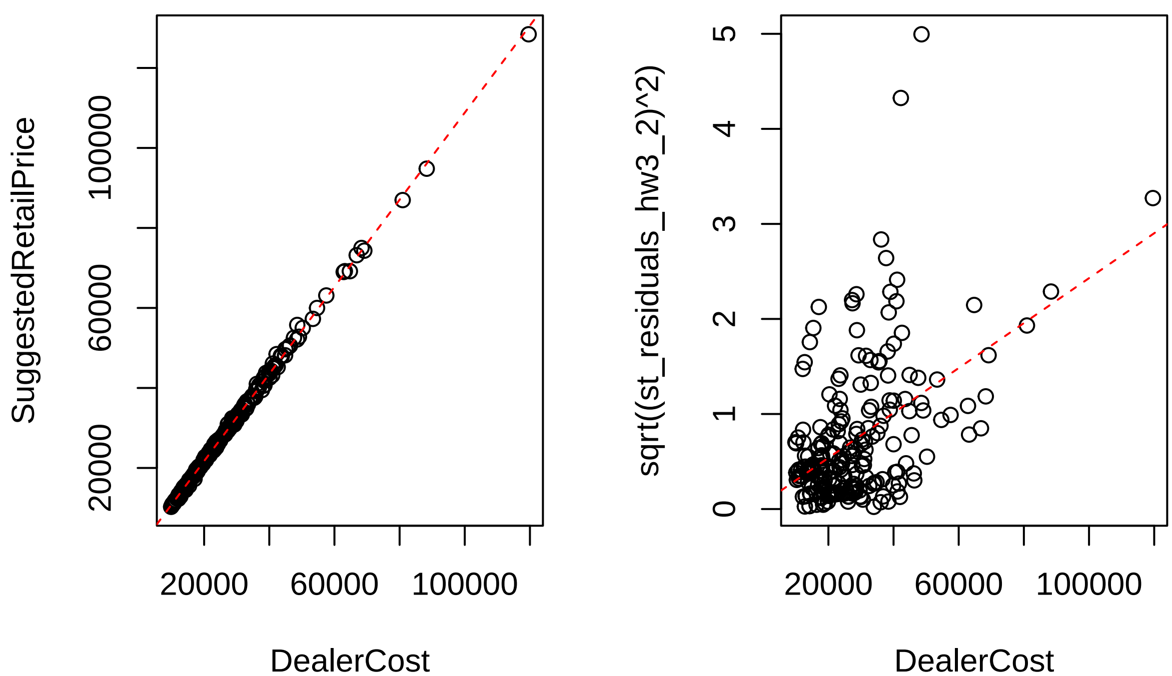
Provide a detailed critique of this conclusion.

```
cars <- read.csv("/Users/user/Desktop/Yonsei/Junior/3-2/Introduction to Data Analysis and Regression/Ho

attach(cars)

lm_data_hw3_2 <- lm(SuggestedRetailPrice~DealerCost)
summary(lm_data_hw3_2)
```

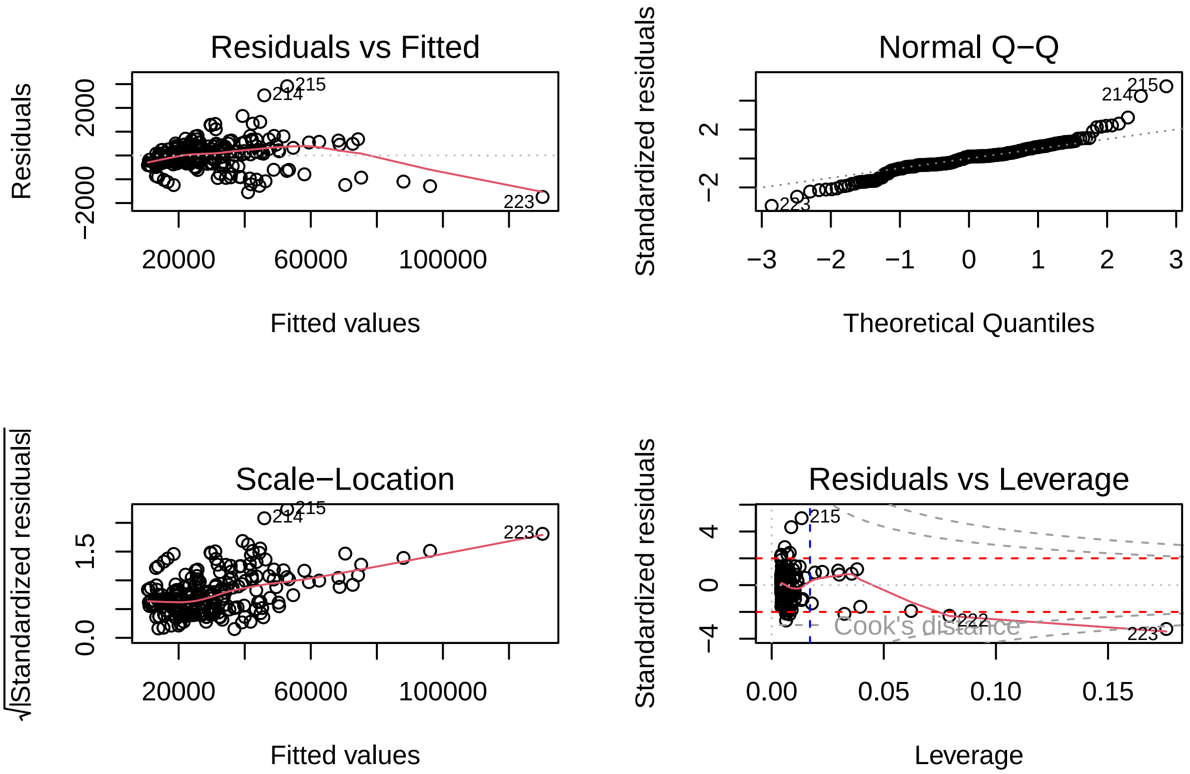
```
##
## Call:
## lm(formula = SuggestedRetailPrice ~ DealerCost)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1743.52  -262.59    74.92   265.98  2912.72
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -61.904248  81.801381  -0.757     0.45
## DealerCost    1.088841   0.002638 412.768   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 587 on 232 degrees of freedom
## Multiple R-squared:  0.9986, Adjusted R-squared:  0.9986
## F-statistic: 1.704e+05 on 1 and 232 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(1,2))
plot(DealerCost, SuggestedRetailPrice)
abline(lm_data_hw3_2, col='red', lty='dashed')

st_residuals_hw3_2 <- rstandard(lm_data_hw3_2)
plot(DealerCost, sqrt((st_residuals_hw3_2)^2))
abline(lsfit(DealerCost, sqrt((st_residuals_hw3_2)^2)), col='red', lty='dashed')
```




```
par(mfrow=c(2,2))
plot(lm_data_hw3_2)
abline(-2,0, col='red', lty='dashed')
abline(2,0, col='red', lty='dashed')
abline(v=4/(length(cars$SuggestedRetailPrice)), col='blue', lty='dashed')
```



(b)

Carefully describe all the shortcomings evident in model (3.10). For each shortcoming, describe the steps needed to overcome the shortcoming.

(1) The square root of absolute value of standardized residuals has a steep slope.
(2) QQ-plot have heavy-tail.

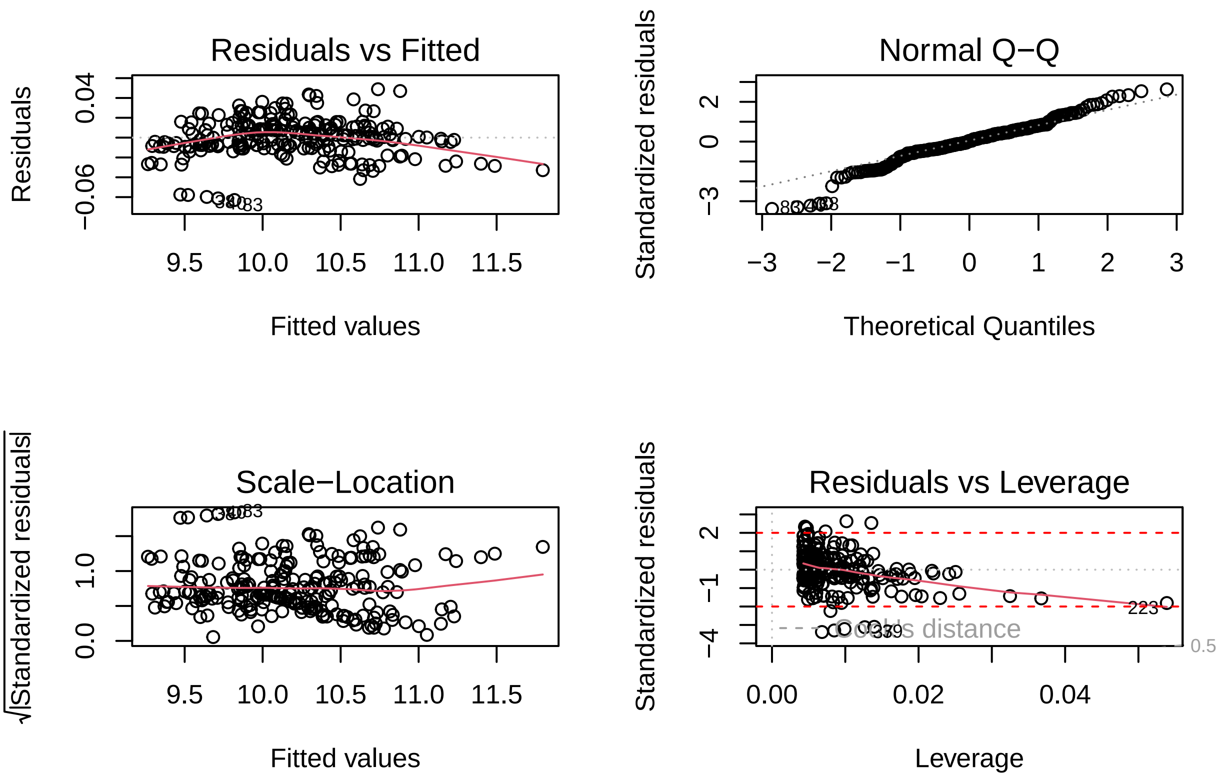
The second model fitted to the data was
log(Suggested Retail Price) $= \beta_0 + \beta_1$ log(Dealer Cost) $+ e$.

```
logDealerCost_hw3_2 <- log(cars$DealerCost)
logSuggestedRetailPrice_hw3_2 <- log(cars$SuggestedRetailPrice)

lm_data_log_hw3_2 <- lm(logSuggestedRetailPrice_hw3_2~logDealerCost_hw3_2)
summary(lm_data_log_hw3_2)
```

```
##
## Call:
## lm(formula = logSuggestedRetailPrice_hw3_2 ~ logDealerCost_hw3_2)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.062920 -0.008694  0.000624  0.010621  0.048798
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -0.069459   0.026459  -2.625  0.00924 **
## logDealerCost_hw3_2  1.014836   0.002616 387.942  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.01865 on 232 degrees of freedom
## Multiple R-squared:  0.9985, Adjusted R-squared:  0.9985
## F-statistic: 1.505e+05 on 1 and 232 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(2,2))
plot(lm_data_log_hw3_2)
abline(-2,0, col='red', lty='dashed')
abline(2,0, col='red', lty='dashed')
abline(v=4/(length(airfare$City)), col='blue', lty='dashed')
```



(c)

(3.11) is an improvement of (3.10). This is because
(1) The square root of absolute value of standardized residuals have flatter regression.
(2) the gap of fitted values are much more smaller, and the leverage, too.

(d)

This is the percentage change of Suggested Retail Price, when the Dealer Cost fluctuates.

(e)

(1) It still have points such that $|\gamma_i| > 2$, meaning that some of them don't have the constant variances.
(2) It improves for the large theoretical quantiles, but not works for small ones.

3.

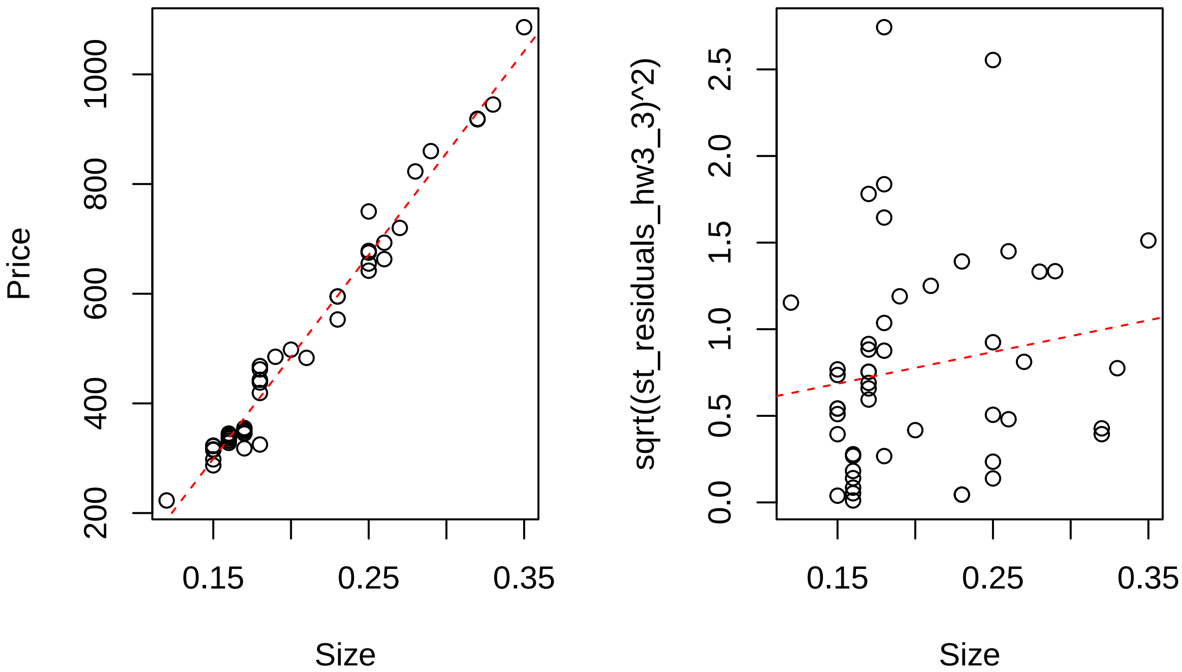
Chu (1996) discusses the development of a regression model to predict the price of diamond rings from the size of their diamond stones (in terms of their weight in carats). Data on both variables were obtained from a full page advertisement placed in the *Straits Times* newspaper by a Singapore-based retailer of diamond jewelry. Only rings made with 20 carat gold and mounted with a single diamond stone were included in the data set. There were 48 such rings of varying designs. (Information on the designs was available but not used in the modeling.)

Part 1 - (a)

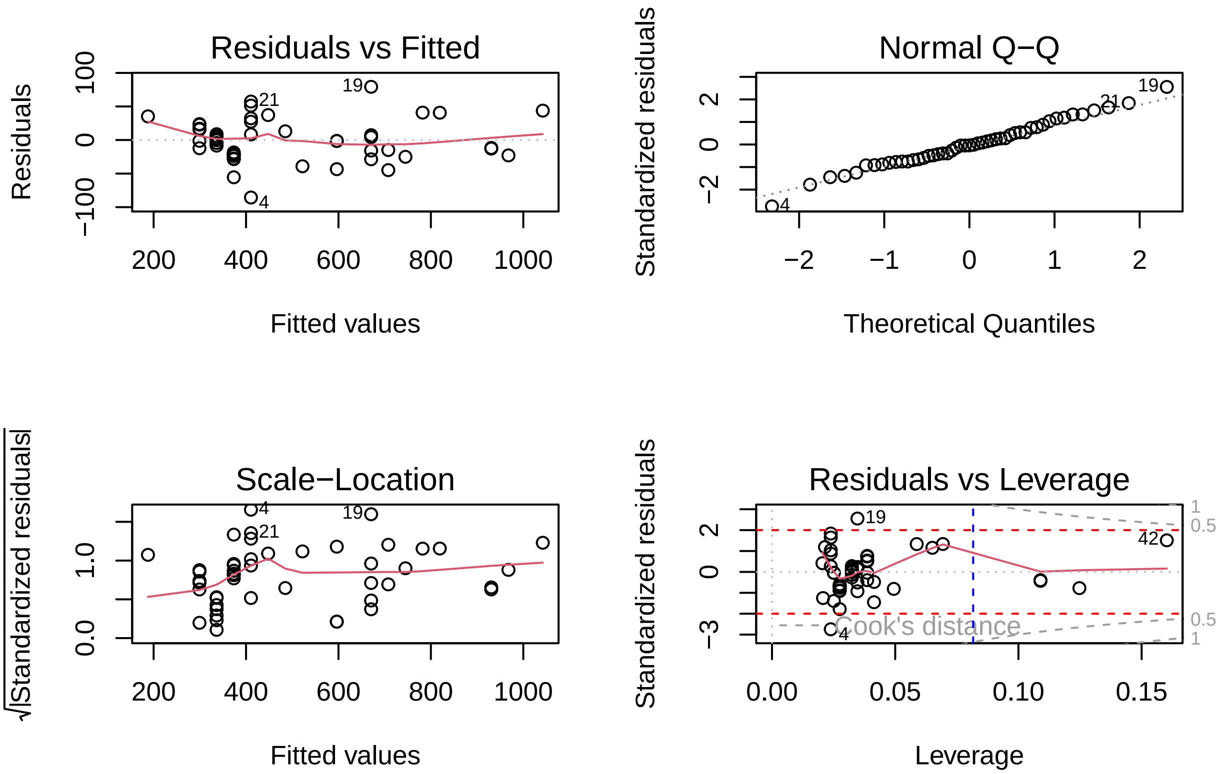
```r
diamonds <- read.table("/Users/user/Desktop/Yonsei/Junior/3-2/Introduction to Data Analysis and Regress

attach(diamonds)

lm_data_hw3_3 <- lm(Price~Size, data=diamonds)
summary(lm_data_hw3_3)
```

```
##
## Call:
## lm(formula = Price ~ Size, data = diamonds)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -85.654 -21.503  -1.203  16.797  79.295
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -258.05      16.94  -15.23   <2e-16 ***
## Size         3715.02      80.41   46.20   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.6 on 47 degrees of freedom
## Multiple R-squared:  0.9785, Adjusted R-squared:  0.978
## F-statistic:  2135 on 1 and 47 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(1,2))
plot(Size, Price)
abline(lm_data_hw3_3, col='red', lty='dashed')

st_residuals_hw3_3 <- rstandard(lm_data_hw3_3)
plot(Size, sqrt((st_residuals_hw3_3)^2))
abline(lsfit(Size, sqrt((st_residuals_hw3_3)^2)), col='red', lty='dashed')
```




```
par(mfrow=c(2,2))
plot(lm_data_hw3_3)
abline(-2,0, col='red', lty='dashed')
abline(2,0, col='red', lty='dashed')
abline(v=4/(length(diamonds$Size)), col='blue', lty='dashed')
```



Part 1 - (b)

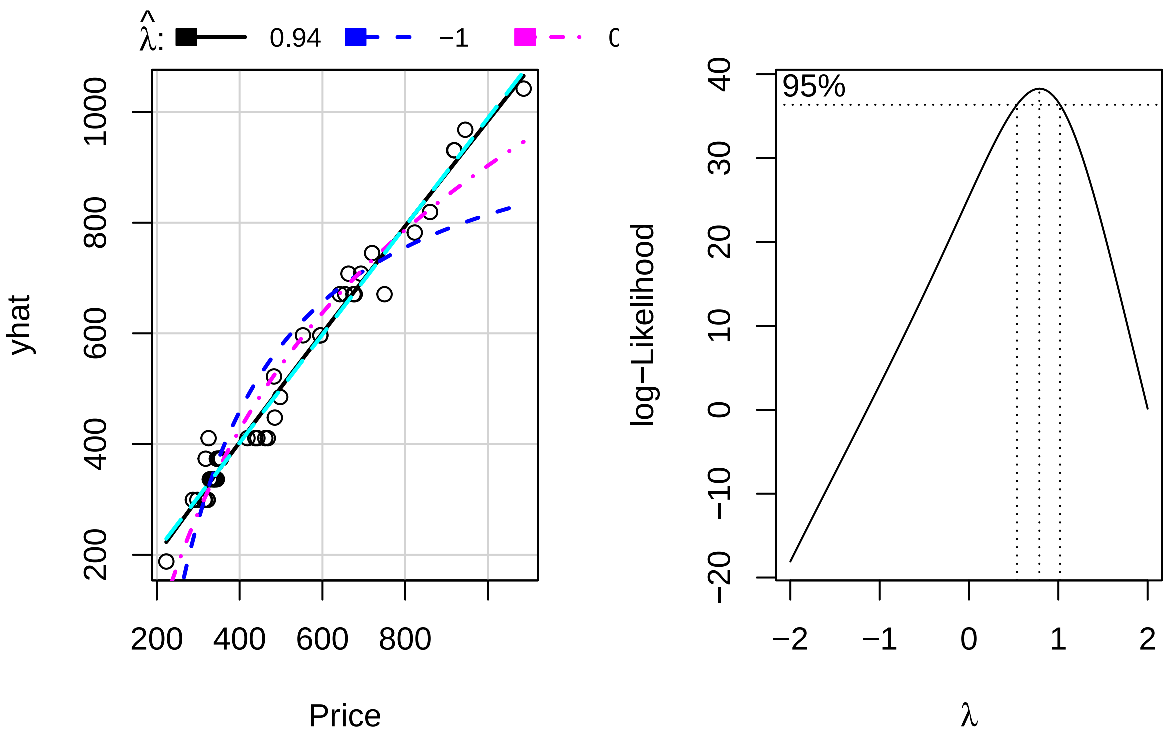
(1) The square root of absolute value of standardized residual has a steep slope.
(2) Square root of absolute value of standardized residual has critical points.

Part 2 - (a)

```
par(mfrow=c(1,2))
inverseResponsePlot(lm_data_hw3_3, key=TRUE)
```

```
##       lambda       RSS
## 1  0.9376257  45670.12
## 2 -1.0000000 272143.61
## 3  0.0000000 101071.53
## 4  1.0000000  45918.17
```

```
boxcox(lm_data_hw3_3)
```



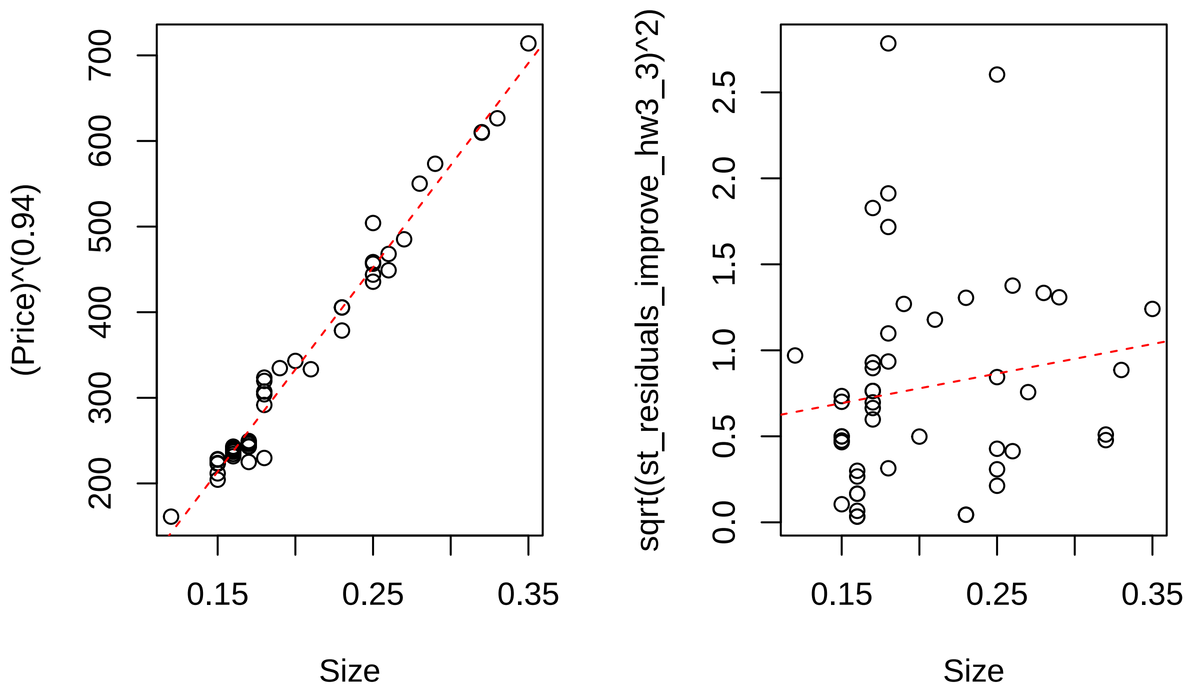
Thus, $\hat{\lambda} = 0.94$ is the best way.

```
lm_data_improve_hw3_3 <- lm((Price)^(0.94)~Size, data=diamonds)
summary(lm_data_improve_hw3_3)
```

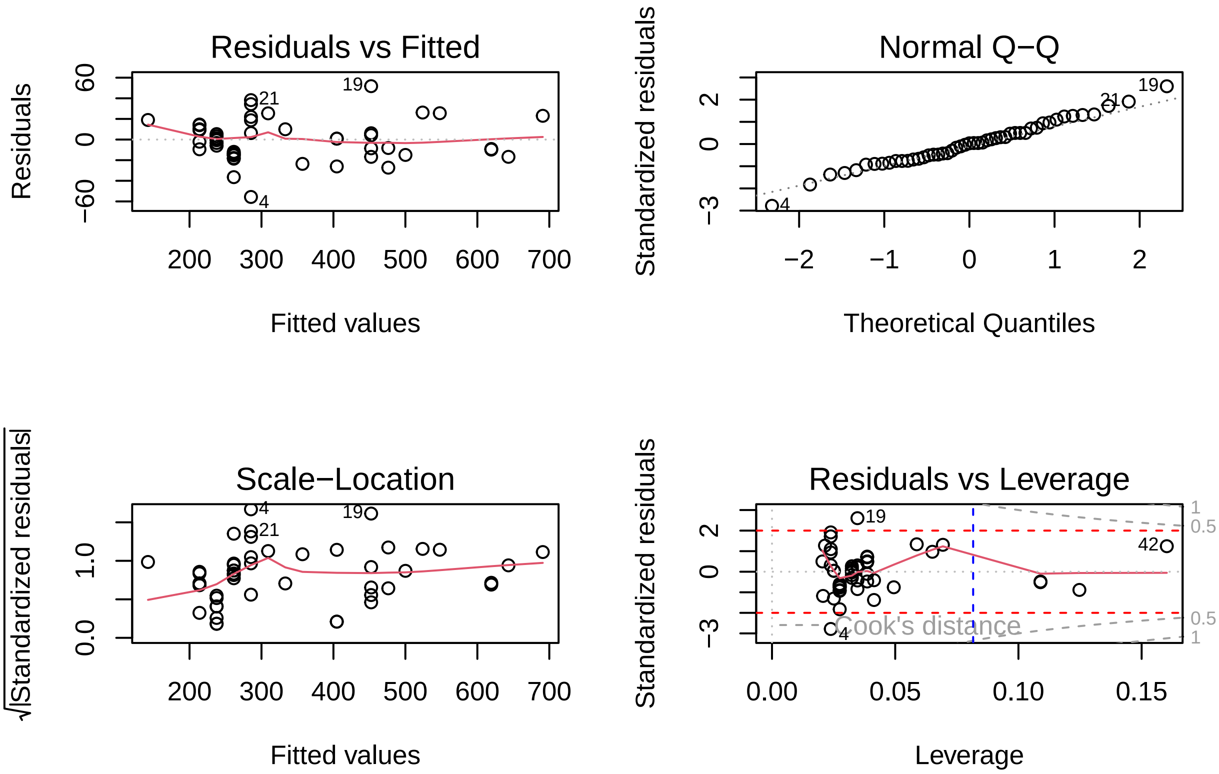
```
##
## Call:
## lm(formula = (Price)^(0.94) ~ Size, data = diamonds)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.675 -13.923   0.667   9.984  51.763
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  -144.06      10.85  -13.28   <2e-16 ***
## Size         2385.78      51.49   46.33   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.24 on 47 degrees of freedom
## Multiple R-squared:  0.9786, Adjusted R-squared:  0.9781
## F-statistic:  2147 on 1 and 47 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(1,2))
plot(Size, (Price)^(0.94))
abline(lm_data_improve_hw3_3, col='red', lty='dashed')

st_residuals_improve_hw3_3 <- rstandard(lm_data_improve_hw3_3)
plot(Size, sqrt((st_residuals_improve_hw3_3)^2))
abline(lsfit(Size, sqrt((st_residuals_improve_hw3_3)^2)), col='red', lty='dashed')
```




```
par(mfrow=c(2,2))
plot(lm_data_improve_hw3_3)
abline(-2,0, col='red', lty='dashed')
abline(2,0, col='red', lty='dashed')
abline(v=4/(length(diamonds$Size)), col='blue', lty='dashed')
```



Actually, it doesn't overcome weaknesses mentioned above.

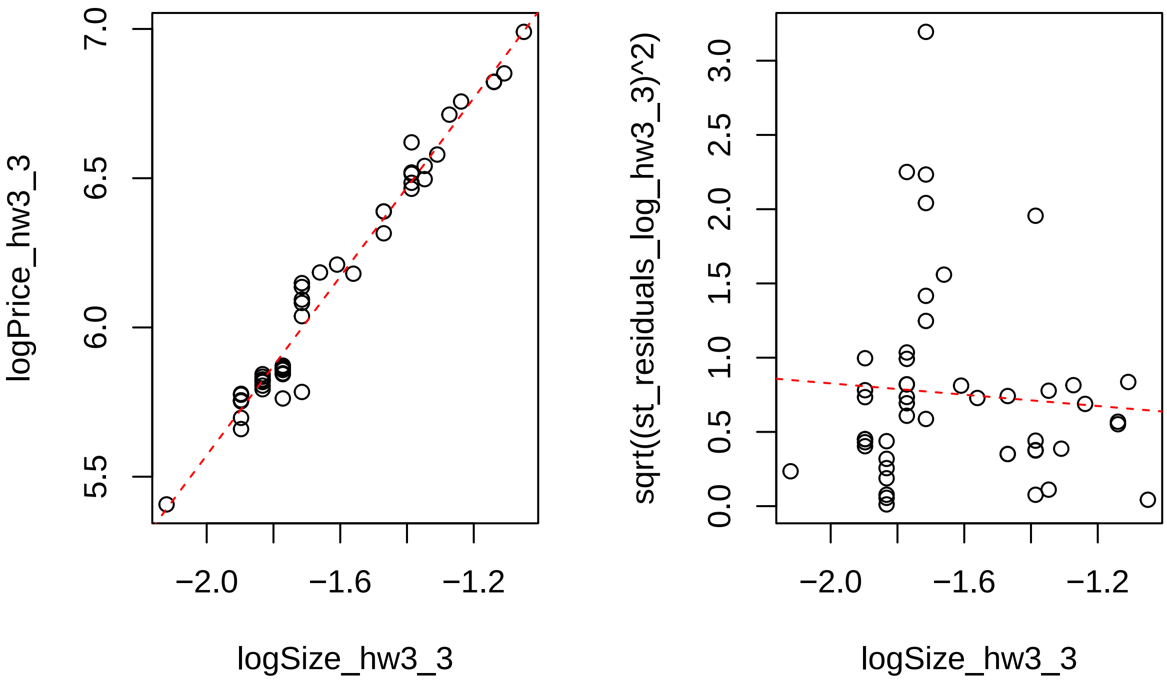
If we use the log-scale SLR model,

```
logSize_hw3_3 <- log(diamonds$Size)
logPrice_hw3_3 <- log(diamonds$Price)

lm_data_log_hw3_3 <- lm(logPrice_hw3_3~logSize_hw3_3, data=diamonds)
summary(lm_data_log_hw3_3)
```

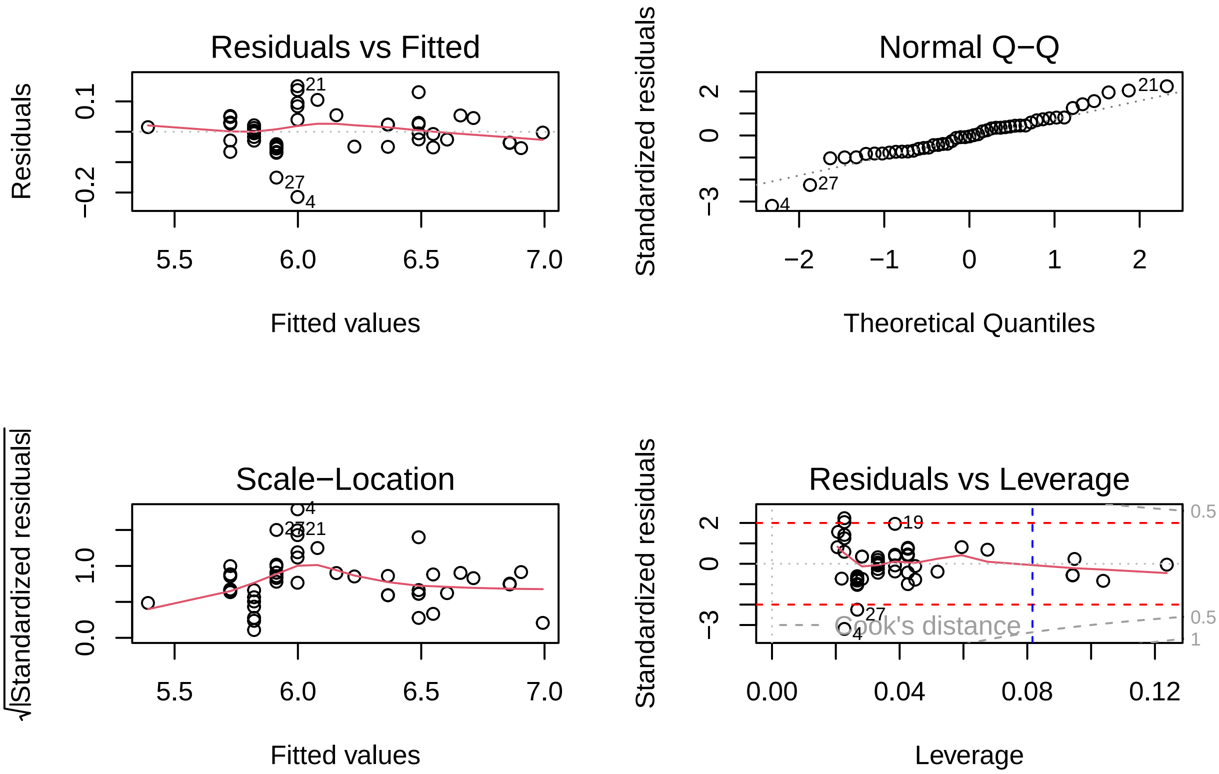
```
##
## Call:
## lm(formula = logPrice_hw3_3 ~ logSize_hw3_3, data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.21460 -0.04646 -0.00274  0.03001  0.15005
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     8.56317    0.06221  137.65   <2e-16 ***
## logSize_hw3_3  1.49566    0.03772   39.65   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06796 on 47 degrees of freedom
## Multiple R-squared:  0.971,  Adjusted R-squared:  0.9704
## F-statistic:  1572 on 1 and 47 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(1,2))
plot(logSize_hw3_3, logPrice_hw3_3)
```

```
abline(lm_data_log_hw3_3, col='red', lty='dashed')

st_residuals_log_hw3_3 <- rstandard(lm_data_log_hw3_3)
plot(logSize_hw3_3, sqrt((st_residuals_log_hw3_3)^2))
abline(lsfit(logSize_hw3_3, sqrt((st_residuals_log_hw3_3)^2)), col='red', lty='dashed')
```




```
par(mfrow=c(2,2))
plot(lm_data_log_hw3_3)
abline(-2,0, col='red', lty='dashed')
abline(2,0, col='red', lty='dashed')
abline(v=4/(length(diamonds$Size)), col='blue', lty='dashed')
```



It reduces the gradient of the absolute value of standardized residuals, but it cannot make the Scale-Location smoothly.

Part 2 - (b)

It improves the (1) weaknesses in Part 1 - (b),
but it doesn't for (2).

Part 3

Part B is better, because the gradient of regression of standardized residual is flatter, which guarantees constant variance.