# CS-E-106: Data Modeling

**Midterm Exam**

***Instructor: Hakan Gogtas***
***Submitted by:*** *Saurabh Kulkarni*

***Due Date:*** *10/21/2019*

**Solution 1:**

*The regression model we want to study:*

$Y_i = b_0 + \epsilon_i$

where, $\epsilon_i$ $N(\lambda, \sigma^2)$

**(A)**

$f(y_i) = f_i = \frac{1}{\sqrt{2*\pi*\sigma}} \exp(-\frac{1}{2}(\frac{y_i-(\beta_0+\lambda)}{\sigma})^2)$

*Likelihood Function:*

$L(\beta_0, \sigma^2) = \prod_{i=1}^{n} f_i = (2\pi)^{\frac{-n}{2}} \sigma^{-n} \exp(\frac{-1}{2} \sum_{i=1}^{n} (\frac{y_i-(\beta_0+\lambda)}{\sigma})^2)$

**(B)**

*Goal:* Choose values $\hat{\beta_0}$, $\hat{\sigma^2}$ that maximize L (or l = ln(L)).

$l = \frac{-n}{2} ln(2\pi) - \frac{n}{2} ln(\sigma^2) - \frac{1}{2} \sum_{i=1}^{n} (\frac{y_i-(\beta_0+\lambda)}{\sigma})^2$

*Calculating optimal $\beta_0$:*

$\frac{\partial l}{\partial \beta_0} = 2 \sum_{i=1}^{n} (\frac{y_i-(\beta_0+\lambda)}{\sigma})(-X_i) =^{set} 0$

$\implies \sum_{i=1}^{n} X_i y_i = (\beta_0 + \lambda) \sum_{i=1}^{n} X_i$

$\implies \beta_0 = \frac{\sum_{i=1}^{n} X_i y_i}{\sum_{i=1}^{n} X_i} - \lambda$

*Calculating optimal $\hat{\sigma^2}$:*

$\frac{\partial l}{\partial \sigma^2} = -\frac{n}{2}(\frac{1}{\sigma^2}) - (-1)\frac{1}{2} \sum_{i=1}^{n} (\frac{y_i-(\beta_0+\lambda)}{\sigma})^2 =^{set} 0$

$\implies \hat{\sigma^2} = \frac{\sum_{i=1}^{n} (y_i-(\beta_0+\lambda))^2}{n}$

**Solution 2:**

**(A)**

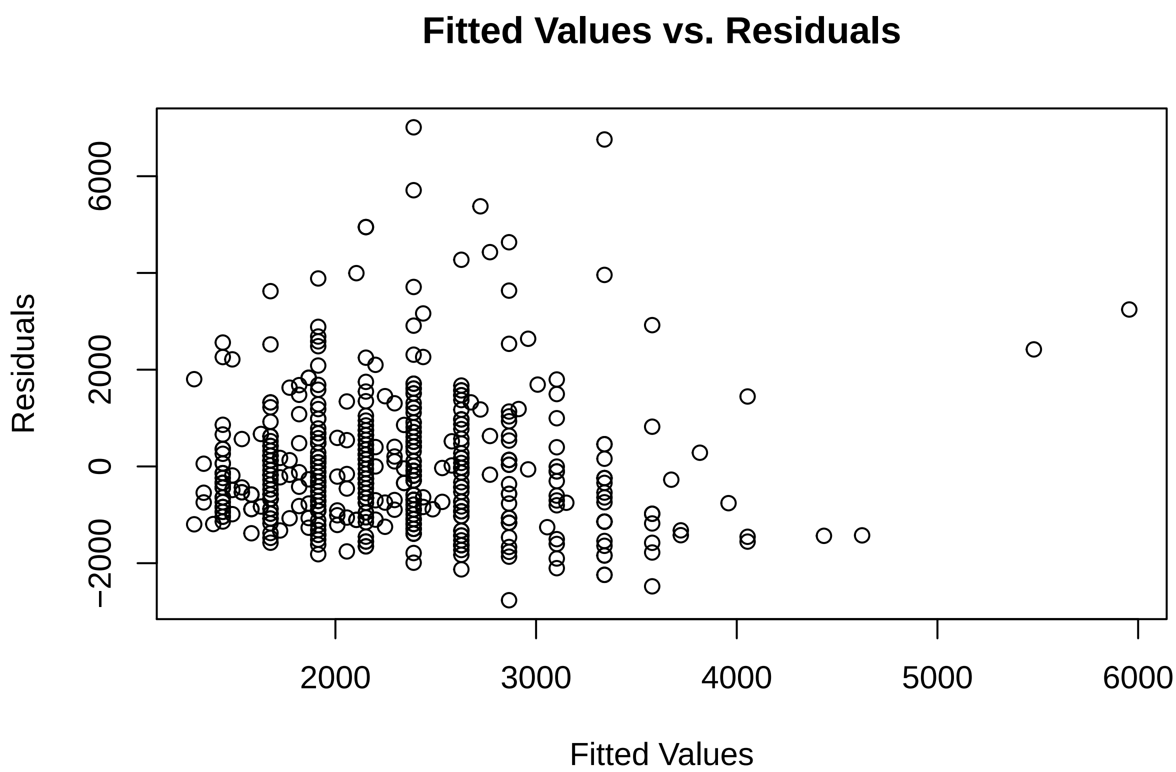
```r
q2_data = read.csv("question2.csv")
lm_q2 = lm(y~x, data=q2_data)
summary(lm_q2)
```

```
##
## Call:
## lm(formula = y ~ x, data = q2_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2765.3  -889.8  -239.8   536.8  7010.2
##
```

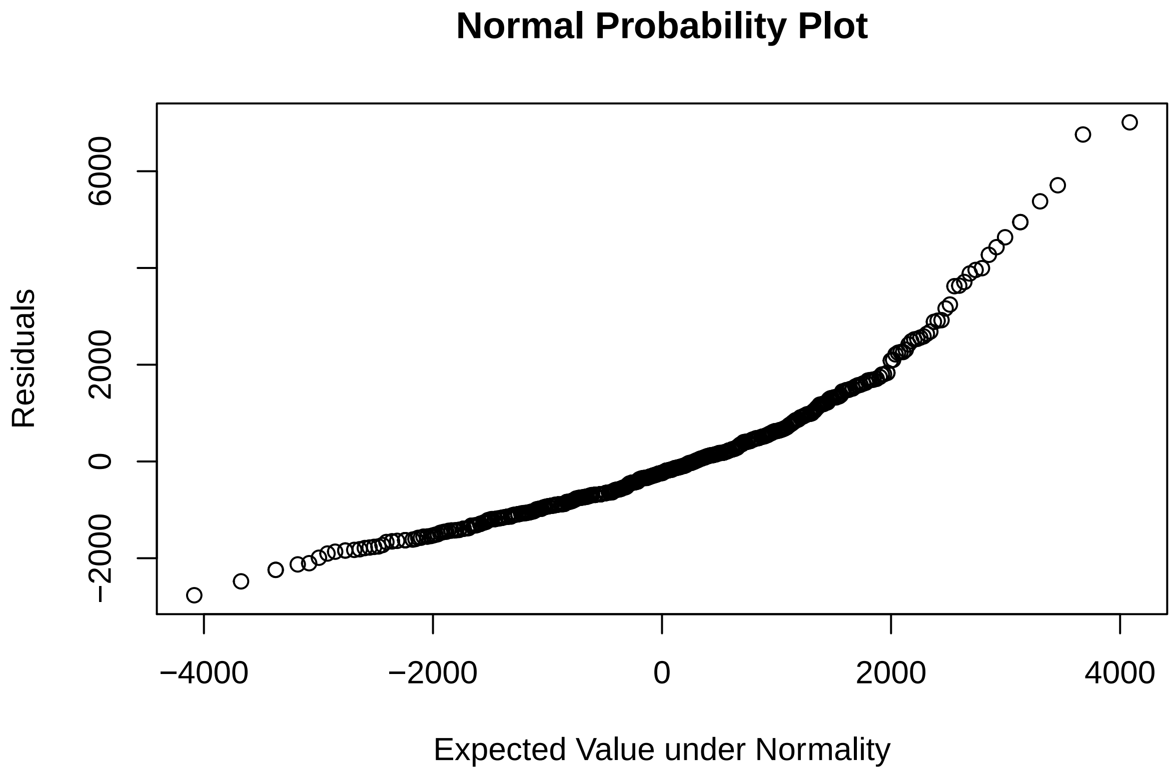
```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1201.124    123.325    9.74   <2e-16 ***
## x             47.549      4.652   10.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1352 on 494 degrees of freedom
## Multiple R-squared:  0.1745, Adjusted R-squared:  0.1729
## F-statistic: 104.5 on 1 and 494 DF,  p-value: < 2.2e-16
```

Regression Function: $y = 1201.124 + 47.549 * x$

```r
build_residual_qq <- function(lm, df, rse){
  ei = lm$residuals
  fitted_values = lm$fitted.values

  par(mfrow=c(1,1))
  plot(fitted_values, ei, xlab="Fitted Values", ylab="Residuals")
  title(main="Fitted Values vs. Residuals")


  ri = rank(ei)
  n = nrow(df)
  zr = (ri-0.375)/(n+0.25)

  #residual standard error from summary(lm) above
  zr1 = rse*qnorm(zr)

  print(cor.test(zr1, ei))

  plot(zr1, ei, xlab="Expected Value under Normality",ylab="Residuals")
  title(main="Normal Probability Plot")

}

build_residual_qq(lm=lm_q2, df=q2_data, rse=1352)
```




```
##
##  Pearson's product-moment correlation
##
## data:  zr1 and ei
## t = 63.43, df = 494, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9332385 0.9526287
## sample estimates:
##       cor
## 0.9437392
```




*Interpretation:*

*Fitted vs. Residual Plot:* The residual plot appears to be mostly equally spread and has no distinct patterns. We do see a few outliers. We can say that there is mostly a contant variance in the error term.

*Normal Probability Plot:* The plot is not linear, which means that the error is not in agreement with the normality.

**(B)**

**Note:** The question script only read: "Calculate the simultaneous 90% confidence interval for". Assuming we are supposed to calculate a 90% simultaneous confidence intervals for $\beta_0$ and $\beta_1$ using Bonferroni method.

```
confint(lm_q2, level=1-0.1/2)
```

```
##                   2.5 %    97.5 %
## (Intercept) 958.81911 1443.4296
## x            38.40798   56.6894
```

**(C)**

```
Xh = data.frame(x=c(85,90))
g = nrow(Xh)

alpha = 0.1
CI.New = predict(lm_q2, Xh, se.fit= TRUE, level = 1-alpha)
B = qt(1 -alpha / (2*g), lm_q2$df)
S = sqrt( g * qf( 1 -alpha, g, lm_q2$df))
spred = sqrt( CI.New$residual.scale^2 + (CI.New$se.fit)^2 ) # (2.38)

print(B)
```

```
## [1] 1.964778
```

```
print(S)
```

```
## [1] 2.150977
```

*Interpretation:* We see that Bonferroni is more efficient, since it has tigher limits.

```
pred_new_CI = t(
rbind(
"Xh" = array(t(Xh)),
"s.pred" = array(spred),
"fit" = array(CI.New$fit),
"lower.B" = array(CI.New$fit-B * spred),
"upper.B" = array(CI.New$fit+ B * spred))
)

pred_new_CI
```

```
##      Xh   s.pred      fit  lower.B  upper.B
## [1,] 85 1383.269 5242.763 2524.947 7960.580
## [2,] 90 1388.300 5480.507 2752.805 8208.208
```

*Double-check:*

```
predict(lm_q2, Xh, se.fit= TRUE, interval = "prediction", level = 1-alpha/g)
```

```
## $fit
##        fit      lwr      upr
## 1 5242.763 2524.947 7960.580
## 2 5480.507 2752.805 8208.208
##
## $se.fit
##        1        2
## 294.4081 317.2062
##
## $df
## [1] 494
##
## $residual.scale
## [1] 1351.576
```

**(D)**

*Brown-Forsythe Test*

*Note:* Assuming $\alpha = 0.05$, since not specified in part (D).

Null Hypothesis: $H_0$: Error variance is constant Alternate Hypothesis: $H_1$: Error variance is not constant

```
summary(q2_data$x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00   15.00   21.00   23.08   30.00  100.00
```

```
ei = lm_q2$residuals
df = data.frame(cbind(q2_data$y,q2_data$x,ei))
df1 = df[df[,2]<=21,]
df2 = df[df[,2]>21,]

med1 = median(df1[,3])
```

```r
med2 = median(df2[,3])

#n1
n1 = nrow(df1)
print(n1)
```

```
## [1] 252
```

```r
#n2
n2 = nrow(df2)
print(n2)
```

```
## [1] 244
```

```r
d1 = abs(df1[,3]-med1)
d2 = abs(df2[,3]-med2)

#calculate means for our answer
mean_d1 = mean(d1)
print(mean_d1)
```

```
## [1] 818.3534
```

```r
mean_d2 = mean(d2)
print(mean_d2)
```

```
## [1] 1104.361
```

```r
s2 = (var(d1)*(n1-1)+var(d2)*(n2-1))/(n1+n2-2)
print(s2)
```

```
## [1] 938356.2
```

```r
#calculate s
s = sqrt(s2)
print(s)
```

```
## [1] 968.6879
```

```r
#testStastic = (mean.d1 - mean.d2) / (s * sqrt((1/n1)+1/n2)
testStastic = (mean_d1-mean_d2)/(s*sqrt((1/n1)+(1/n2)))
print(testStastic)
```

```
## [1] -3.287369
```

```r
t = qt(1-0.05/2, lm_q2$df.residual)
print(t)
```

```
## [1] 1.964778
```

*Decision Rule:*

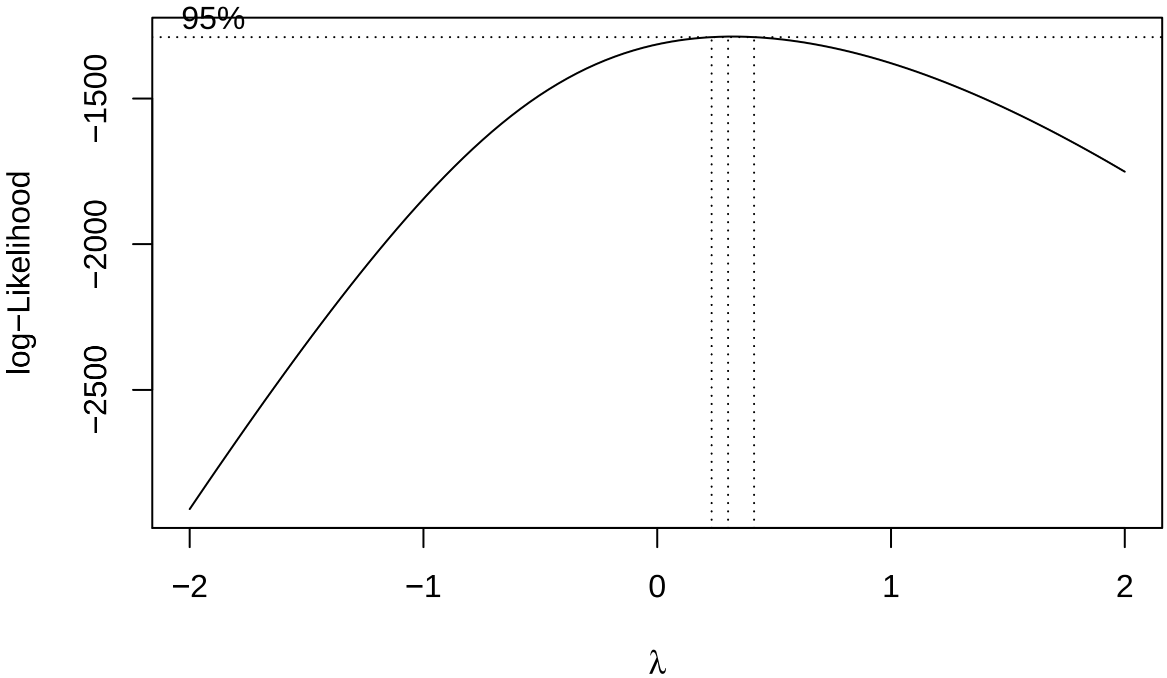
- If $|testStatistic| \leq t(1-\alpha/2, n-2)$, conclude $H_0$: constant error variance
- If $|testStatistic| > t(1-\alpha/2, n-2)$, conclude $H_1$: non-constant error variance

*Result:*

Since $|-3.287369| > 1.964778$ i.e. $|testStatistic| > t(1-\alpha/2, n-2)$, we conclude $H_1$. The error variance is not constant and thus varies with X.

**(E)**

```r
library(MASS)
par(mfrow=c(1,1))
boxcox(lm_q2)
```




*Interpretation:*

The suggested Y transformation with Box-Cox method is: $\lambda \approx 0$. Thus, we'll assume the suggested $\lambda = 0$ (as suggested in notes Ch.3, slide 77 - "a nearby lambda is easy to understand"), which implies the suggested transformation is: $Y' = log(Y)$.

```r
y1 = log(q2_data$y)
q2_data = cbind(q2_data, y1)
```

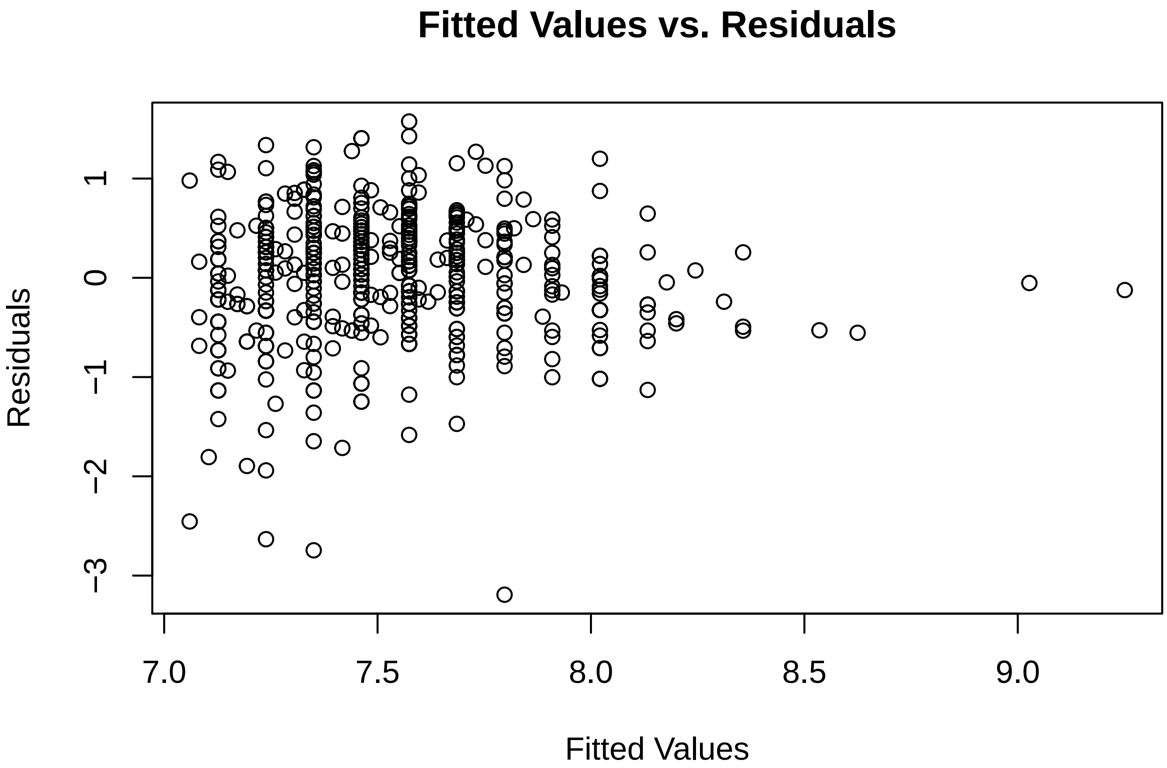
```r
lm_q2_t = lm(y1~x, data=q2_data)
summary(lm_q2_t)
```

```
##
## Call:
## lm(formula = y1 ~ x, data = q2_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1924 -0.3309  0.0536  0.4098  1.5745
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.015047   0.058037  120.87   <2e-16 ***
## x           0.022357   0.002189   10.21   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6361 on 494 degrees of freedom
## Multiple R-squared:  0.1743, Adjusted R-squared:  0.1726
```

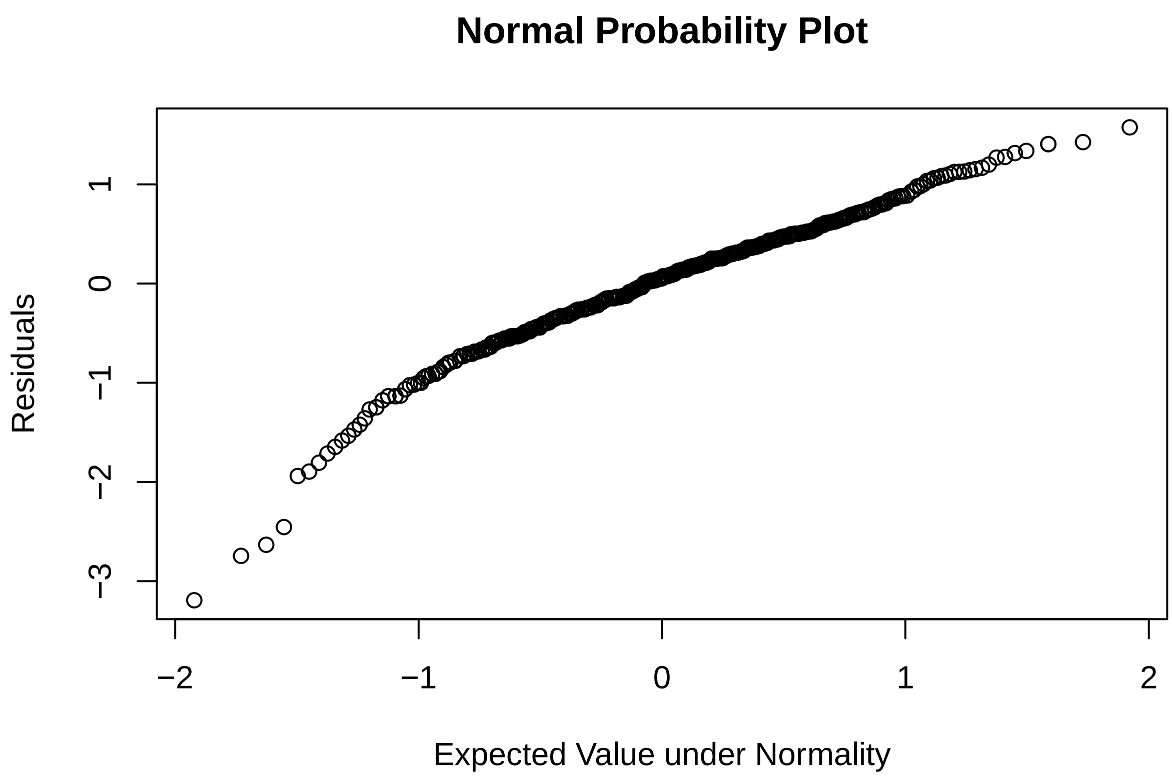
```
## F-statistic: 104.3 on 1 and 494 DF,  p-value: < 2.2e-16
```

The regression function using the transformed data $= log(y) = 7.015047 + 0.022357 * x$ or $y = \exp(7.015047 + 0.022357 * x)$

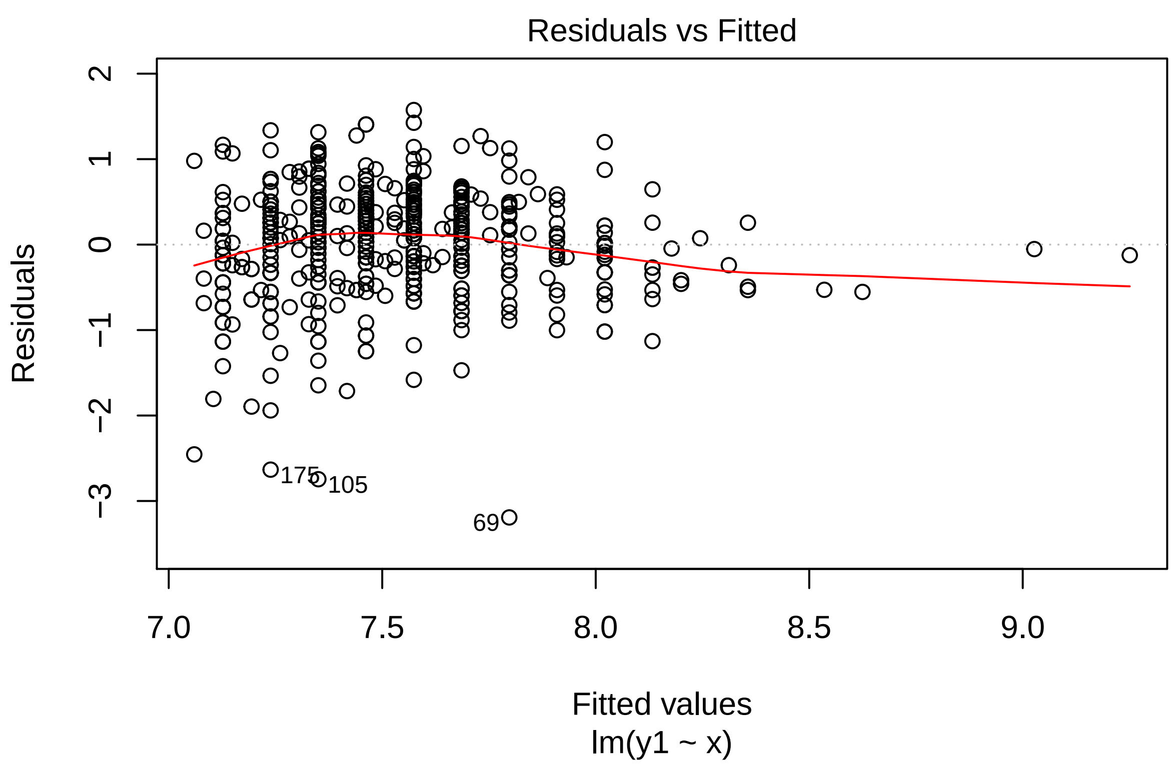
```r
build_residual_qq(lm=lm_q2_t, df=q2_data, rse=0.6361)
```

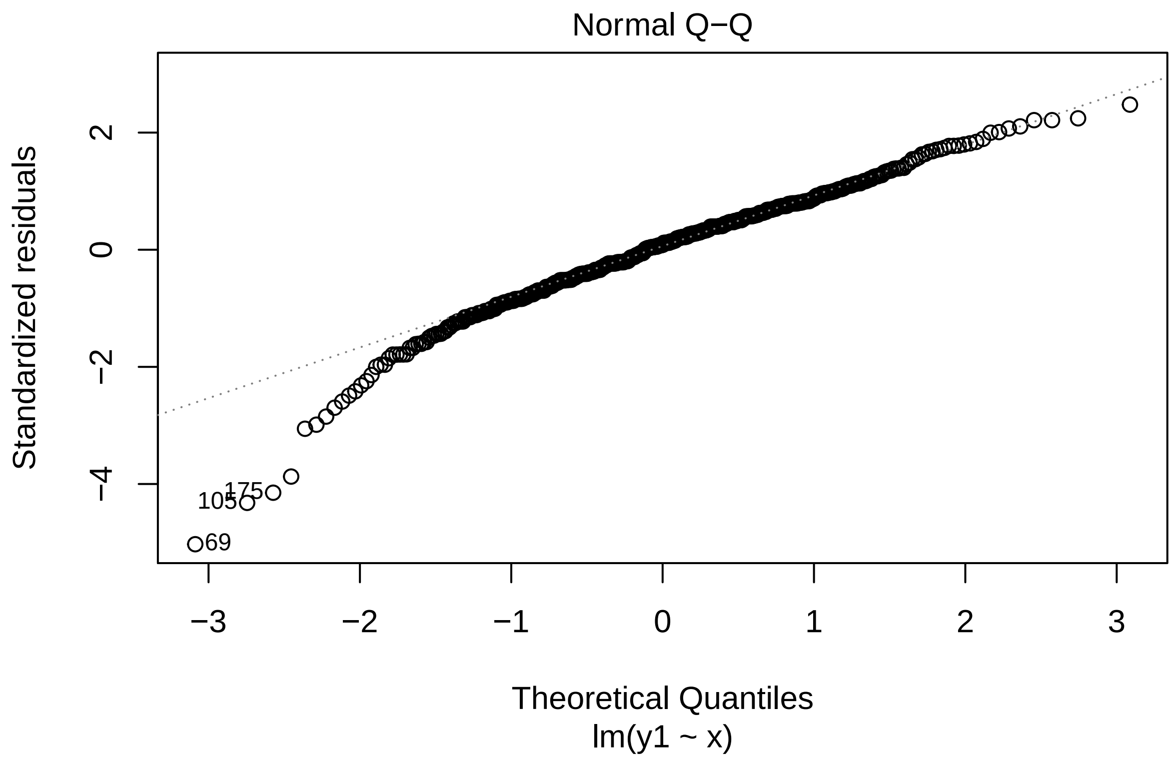


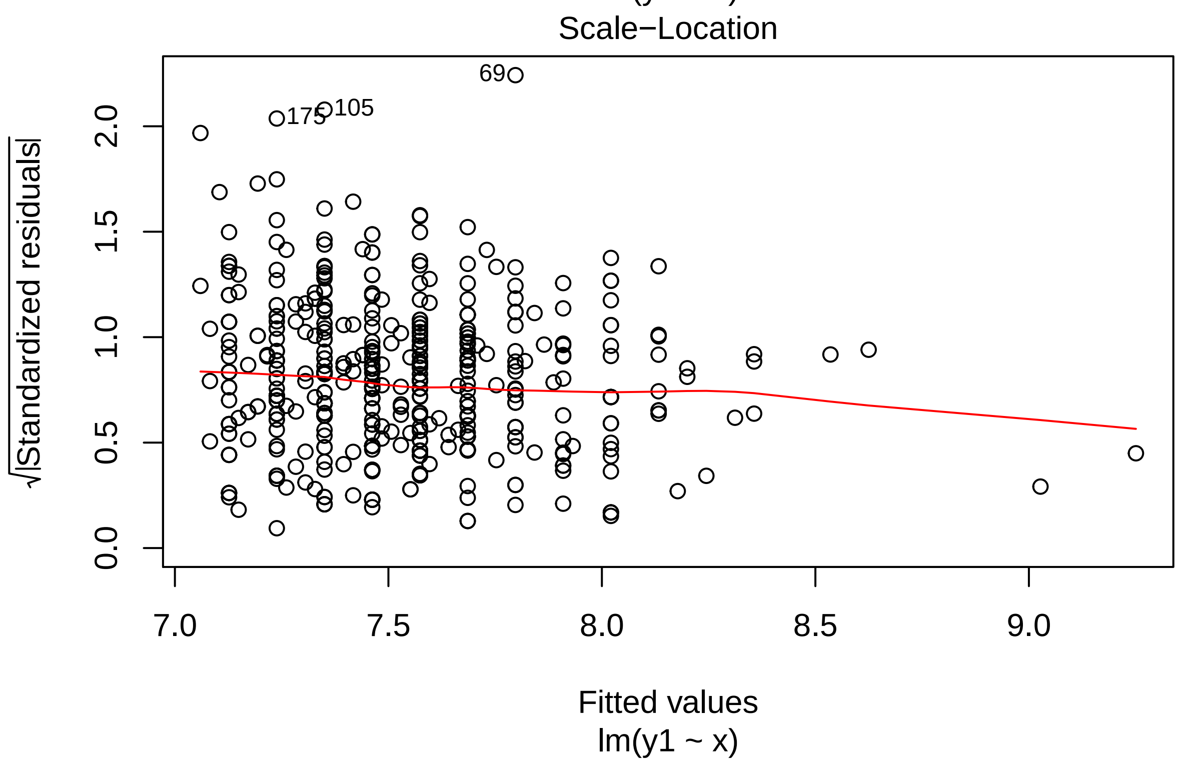

```
##
##  Pearson's product-moment correlation
##
## data:  zr1 and ei
## t = 111.39, df = 494, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9769787 0.9837716
## sample estimates:
##       cor
## 0.9806684
```

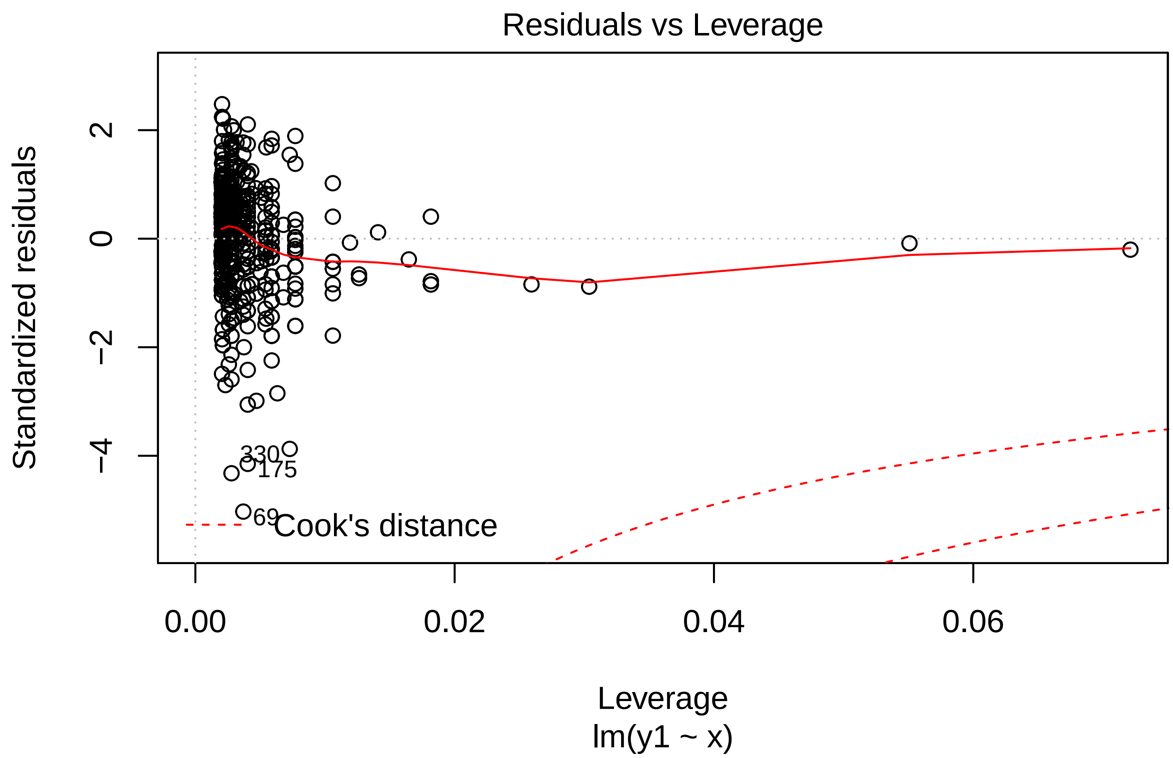

Normal Probability Plot
Residuals
Expected Value under Normality
−3
−2
−1
0
1
2

```r
plot(lm_q2_t)
```


Residuals vs Fitted
Residuals
Fitted values
lm(y1 ~ x)
175
105
69
7.0
7.5
8.0
8.5
9.0


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(y1 ~ x)
105
175
69


Scale−Location
√|Standardized residuals|
Fitted values
lm(y1 ~ x)
69
175
105




*Interpretation:*

*Fitted vs. Residual Plot:* The residual plot appears to be mostly equally spread and has no distinct patterns. We still do see a few outliers. We can say that there is mostly a contant variance in the error term.

*Normal Probability Plot:* The plot is mostly linear, which means that the error is mostly in agreement with the normality. This could be due to the approximation we did of the $\lambda$ value we got using Box-Cox method.

**Solution 3:**

**(A)**

```
q2_data = read.csv("question2.csv")
```

```
set.seed(1023)
train_ind = sample(1:nrow(q2_data), 0.7 * nrow(q2_data))
test_ind = setdiff(1:nrow(q2_data), train_ind)
train_df = q2_data[train_ind,]
test_df = q2_data[test_ind,]
```

**(B)**

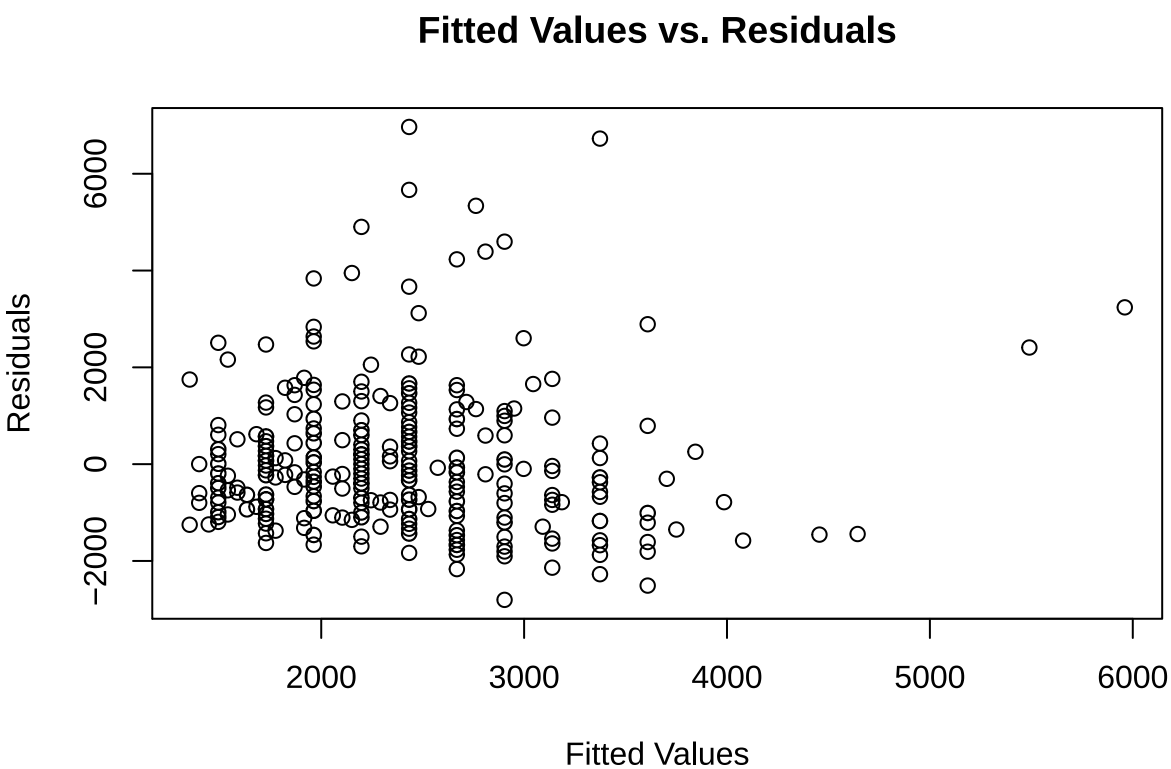
```
lm_q3_tr = lm(y~x, data=train_df)
summary(lm_q3_tr)
```

```
##
## Call:
## lm(formula = y ~ x, data = train_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2803.6  -933.3  -233.3   572.1  6966.7
##
```

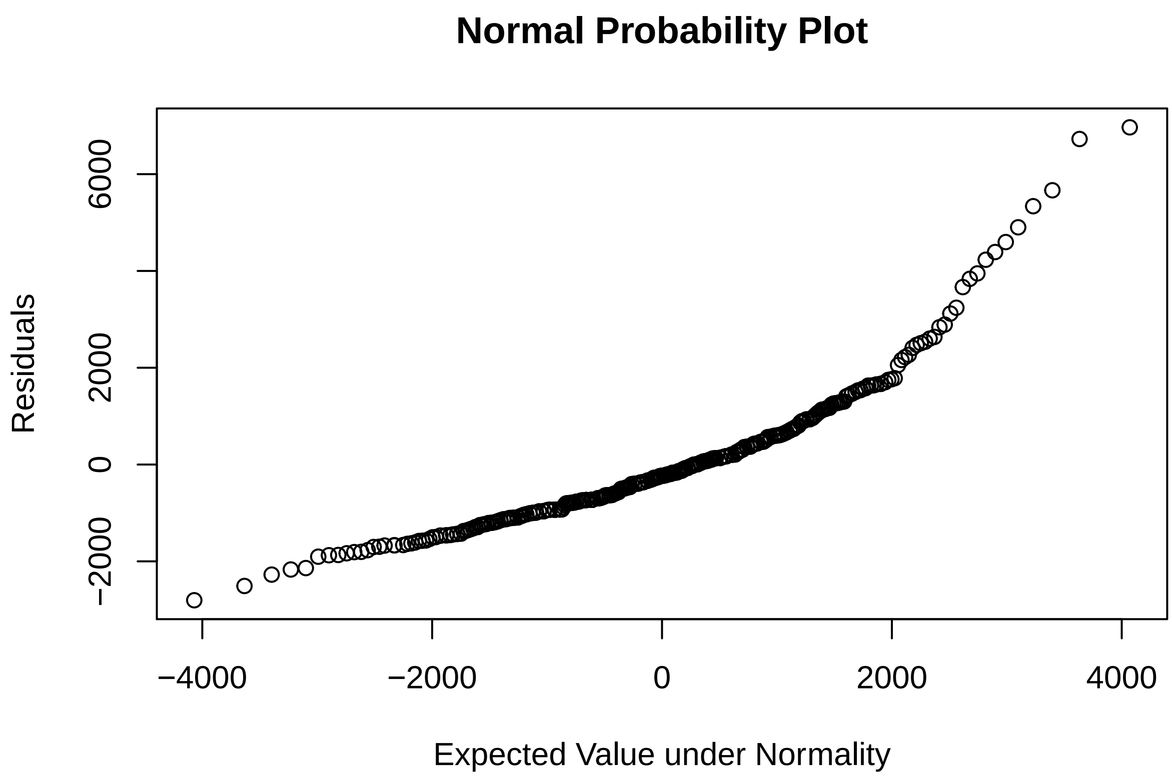
```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1257.562    146.831   8.565 3.65e-16 ***
## x             47.030      5.469   8.599 2.86e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1398 on 345 degrees of freedom
## Multiple R-squared:  0.1765, Adjusted R-squared:  0.1741
## F-statistic: 73.94 on 1 and 345 DF,  p-value: 2.858e-16
```

Regression Function on development sample: $y = 1257.562 + 47.030 * x$

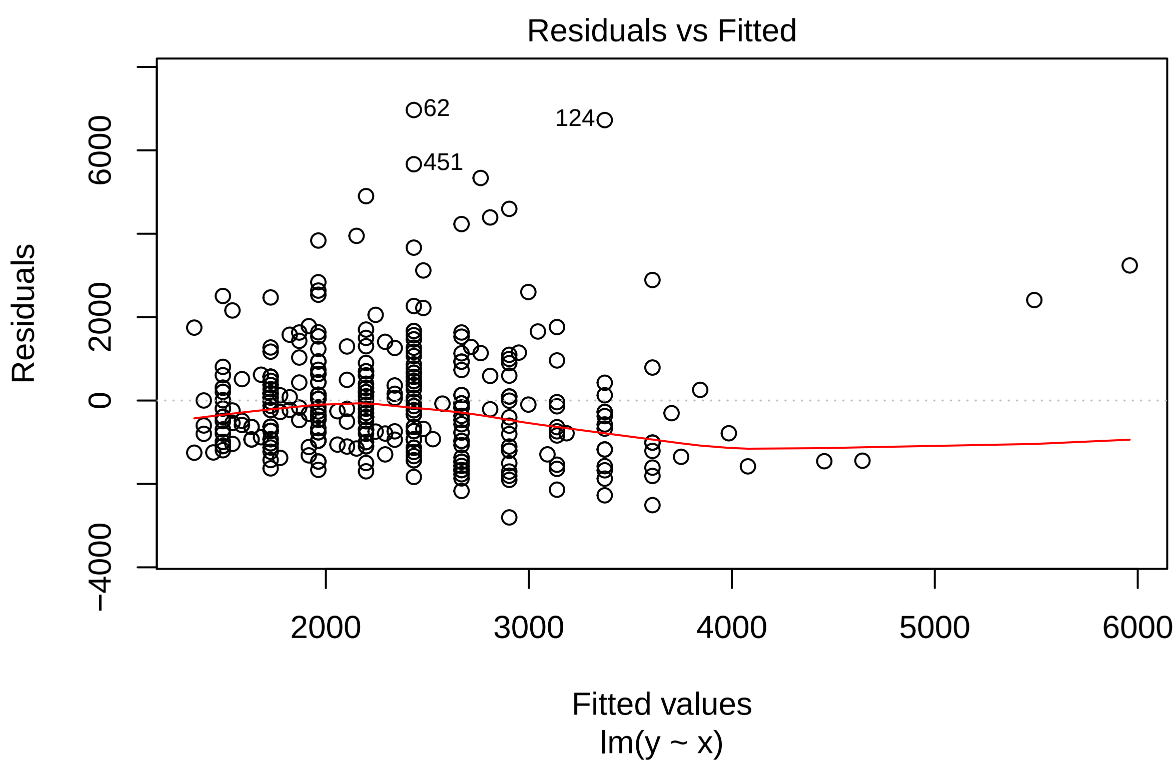
```
build_residual_qq(lm=lm_q3_tr, df=train_df, rse=1398)
```

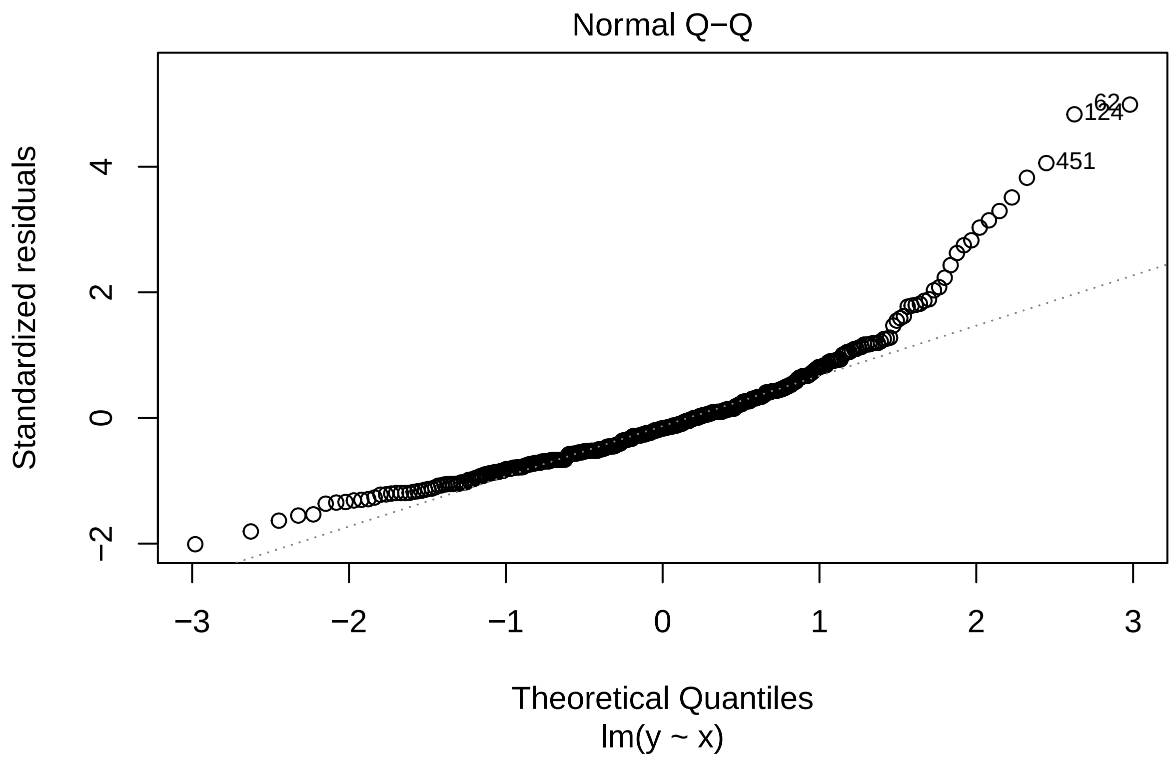


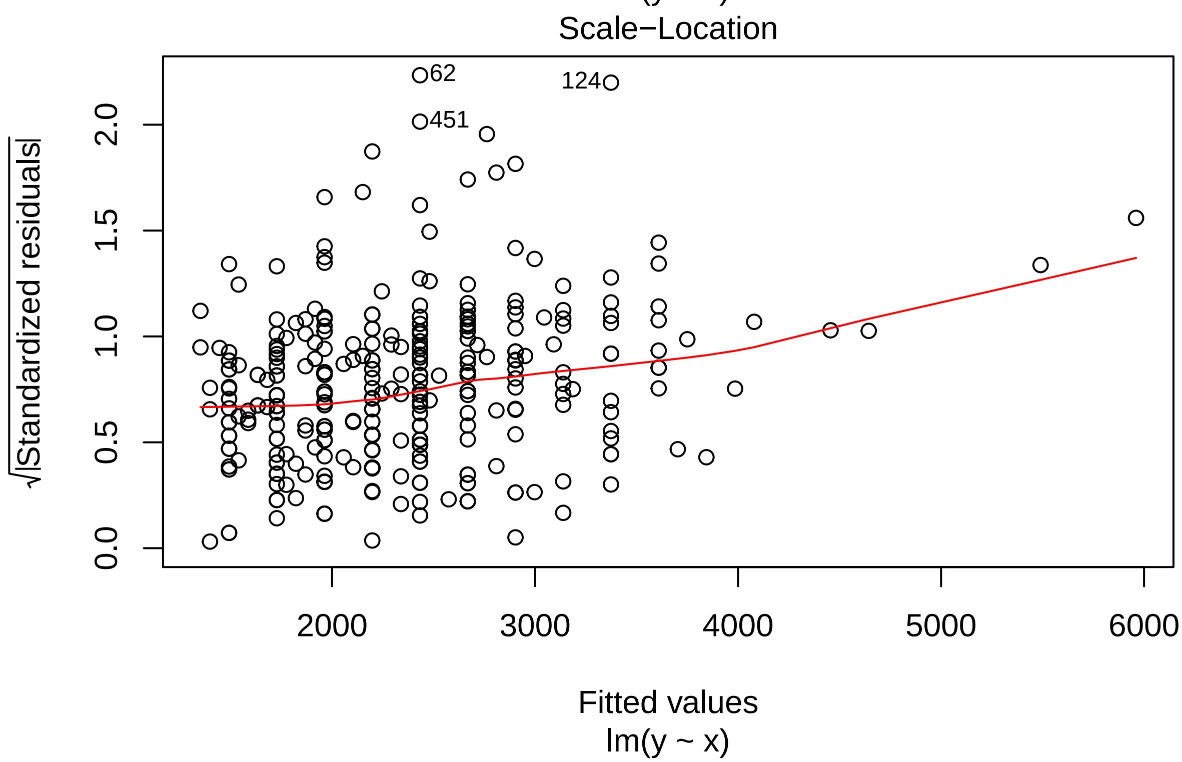

```
##
##  Pearson's product-moment correlation
##
## data:  zr1 and ei
## t = 50.481, df = 345, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9245582 0.9499134
## sample estimates:
##       cor
## 0.9384884
```

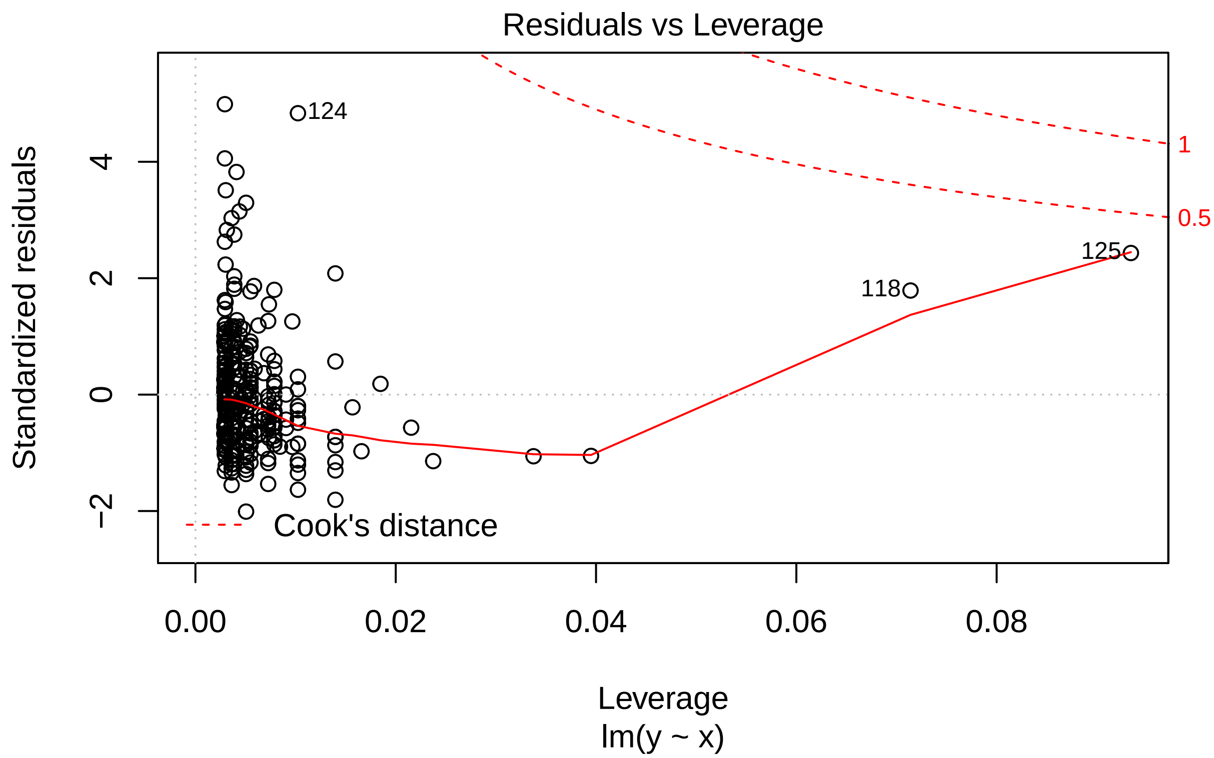

Normal Probability Plot
Residuals
6000
2000
0
−2000
−4000
−2000
0
2000
4000
Expected Value under Normality

```
plot(lm_q3_tr)
```


Residuals vs Fitted
62
124
451
Residuals
6000
2000
0
−4000
2000
3000
4000
5000
6000
Fitted values
lm(y ~ x)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(y ~ x)
62
124
451
−2
0
2
4
−3
−2
−1
0
1
2
3


Scale−Location
√|Standardized residuals|
Fitted values
lm(y ~ x)
62
451
124
0.0
0.5
1.0
1.5
2.0
2000
3000
4000
5000
6000




*Interpretation:*

Both plots are very similar to the plots obtained in Q2.A, with similar interpretaions.

*Fitted vs. Residual Plot:* The residual plot appears to be mostly equally spread and has no distinct patterns. We do see a few outliers. We can say that there is mostly a contant variance in the error term.

*Normal Probability Plot:* The plot is not linear, which means that the error is not in agreement with the normality.

**(C)**

```
yi = test_df$y
yBar = mean(test_df$y)
yHat = predict(lm_q3_tr, test_df)
resids = yi-yHat
SSE = sum(resids^2)
SST = sum((yi-yBar)^2)

R2 = 1 - SSE/SST

print(paste("R-squared on hold-out sample:",R2))
```

```
## [1] "R-squared on hold-out sample: 0.158098981561254"
```

**Solution 4:**

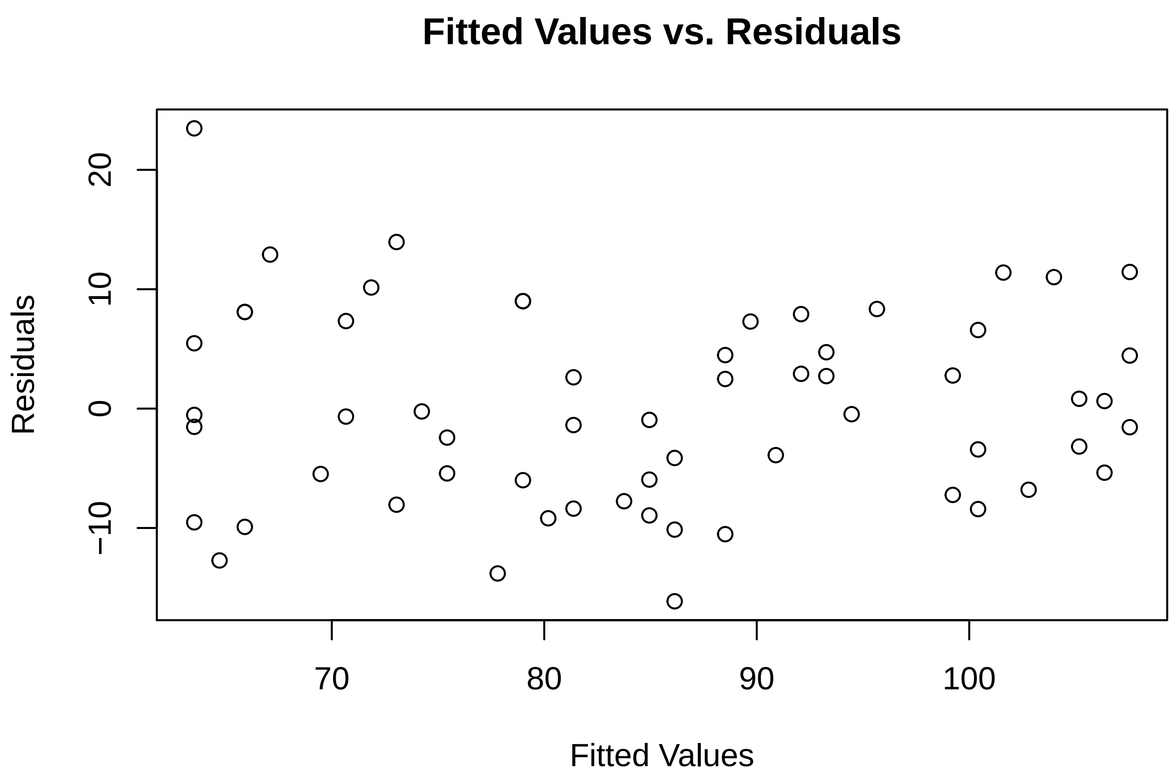
```
q4_data = read.csv("question4.csv")
lm_q4 = lm(Y~X, data=q4_data)
summary(lm_q4)
```

```
##
## Call:
## lm(formula = Y ~ X, data = q4_data)
```

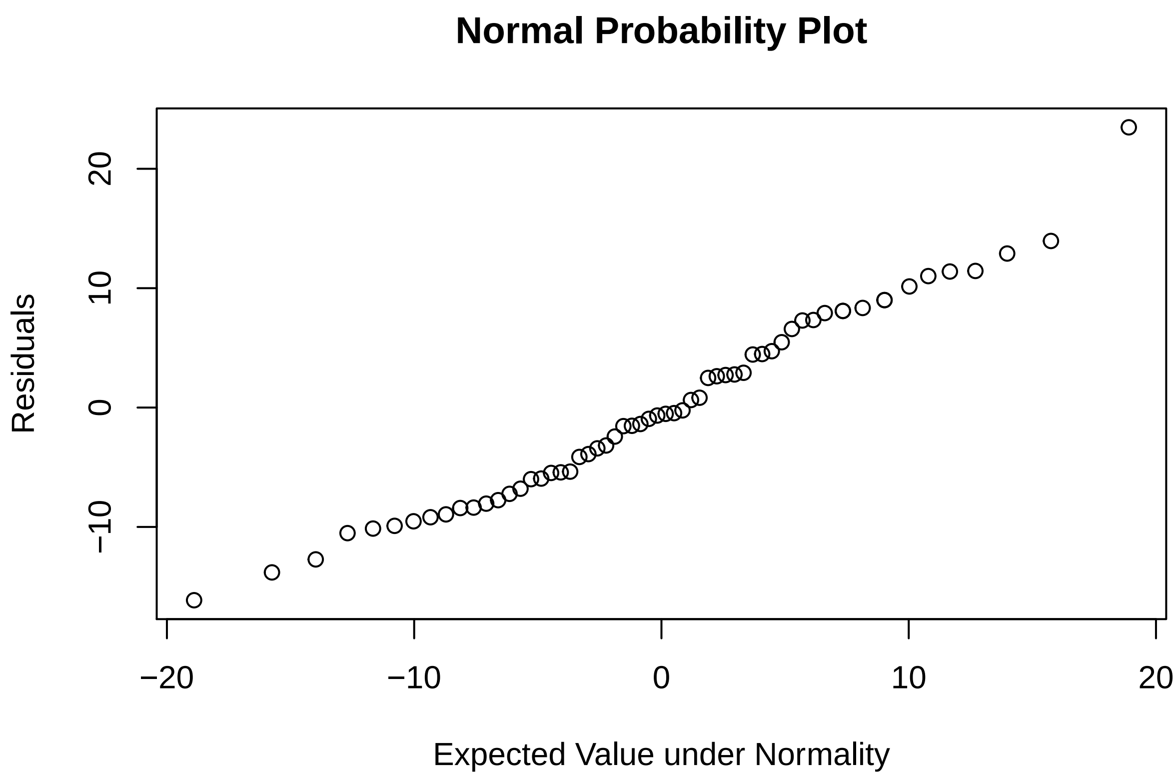
```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.1368  -6.1968  -0.5969   6.7607  23.4731
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 156.3466     5.5123   28.36   <2e-16 ***
## X            -1.1900     0.0902  -13.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.173 on 58 degrees of freedom
## Multiple R-squared:  0.7501, Adjusted R-squared:  0.7458
## F-statistic: 174.1 on 1 and 58 DF,  p-value: < 2.2e-16
```

The regression function: $Y = 156.3466 + 1.1900 * X$

```
build_residual_qq(lm=lm_q4, df=q4_data, rse=8.173)
```




```
##
##  Pearson's product-moment correlation
##
## data:  zr1 and ei
## t = 52.781, df = 58, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9828326 0.9938886
## sample estimates:
##       cor
## 0.9897499
```




*Interpretation:*

*Fitted vs. Residual Plot:* The residual plot appears to be equally spread and has no distinct patterns and no visible extreme outliers. We can say that there is mostly a contant variance in the error term.

*Normal Probability Plot:* The plot is mostly linear, which means that the error is in agreement with the normality.

**(B)**

*Breusch-Pagan Test*

Null Hypothesis: $H_0$: Error variance is constant Alternate Hypothesis: $H_1$: Error variance is not constant

```
ei = lm_q4$residuals
ei2 = ei^2
f = lm(ei2~q4_data$X)
summary(f)
```

```
##
## Call:
## lm(formula = ei2 ~ q4_data$X)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -99.77 -43.63 -20.29  12.80 450.94
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -53.5326    56.0149  -0.956   0.3432
## q4_data$X     1.9690     0.9166   2.148   0.0359 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 83.05 on 58 degrees of freedom
## Multiple R-squared:  0.0737, Adjusted R-squared:  0.05773
## F-statistic: 4.615 on 1 and 58 DF,  p-value: 0.03589
```

```
#to find SSE(R) and SSR(R)
anova(f)
```

```
## Analysis of Variance Table
##
## Response: ei2
##           Df Sum Sq Mean Sq F value  Pr(>F)
## q4_data$X  1  31833   31833  4.6148 0.03589 *
## Residuals 58 400089    6898
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#to find SSE(F) and SSR(F)
anova(lm_q4)
```

```
## Analysis of Variance Table
##
## Response: Y
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## X          1 11627.5 11627.5  174.06 < 2.2e-16 ***
## Residuals 58  3874.4    66.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
SSR_R = 31833
SSE_R = 400089

SSR_F = 11627.5
SSE_F= 3874.4

n = nrow(q4_data)

#chi-squared: [SSR(R)/2] / [SSE(F)/n]^2
chiTest = (SSR_R/2) / ((SSE_F/n))^2
print(chiTest)
```

```
## [1] 3.817167
```

```
#p
chi = qchisq(1-0.05,1)
print(chi)
```

```
## [1] 3.841459
```

Decision Rule:

- If $chiTest \leq \chi^2(1-\alpha, 1)$, conclude $H_0$: constant error variance
- If $chiTest > \chi^2(1-\alpha, 1)$, conclude $H_1$: non-constant error variance

Result: Since $3.817167 \leq 3.841459$ i.e. $chiTest \leq \chi^2(1-\alpha, 1)$, we conclude $H_0$. The error variance is constant.

**Solution 5:**

**(A)**

*Given:*

```r
n = 45
F = 970
MSE = 80
```

$F = \frac{MSR}{MSE}$

```r
MSR = F*MSE
MSR
```

```
## [1] 77600
```

$MSE = \frac{SSE}{n-2}$

```r
SSE = MSE*(n-2)
SSE
```

```
## [1] 3440
```

```r
SSR = MSR/1
SSR
```

```
## [1] 77600
```

```r
df_R = n-2
df_E = 1

print(df_R)
```

```
## [1] 43
```

```r
print(df_E)
```

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

**(B)**

```r
R2 = 1 - SSE/(SSR+SSE)
R2
```

```
## [1] 0.9575518
```

*Interpretation:* We get an R-squared value of 0.96 i.e. 95.7% of the variation in Y is explained by the independent variable X. Thus, the model is statistically significant based on $R^2$ value.