CS-E-106: Data Modeling

Assignment 4

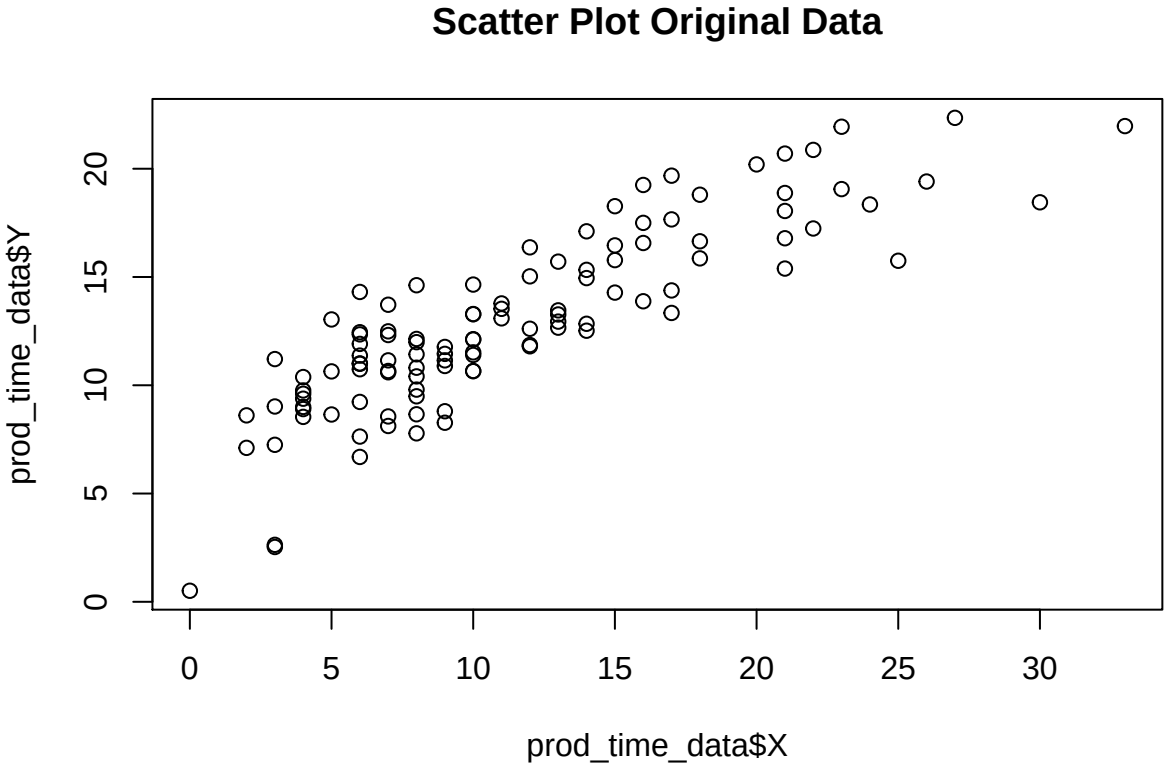
Instructor: Hakan Gogtas
Submitted by: *Saurabh Kulkarni*

Due Date: *10/14/2019*

Solution 1:

(a)

```
par(mfrow=c(1,1))
prod_time_data = read.csv("Production Time.csv")
plot(prod_time_data$X, prod_time_data$Y)
title(main="Scatter Plot Original Data")
```



Interpretation:

A linear relation does not seem adequate here. Based on the scatterplot, there seems to be a curvilinear relation between X and Y and for the same reason we need a transormation on either X or Y.

(b)

```
X1 = sqrt(prod_time_data$X)
prod_time_data2 = cbind(X1, prod_time_data)
lm_prod = lm(Y~X1, data=prod_time_data2)
summary(lm_prod)
```

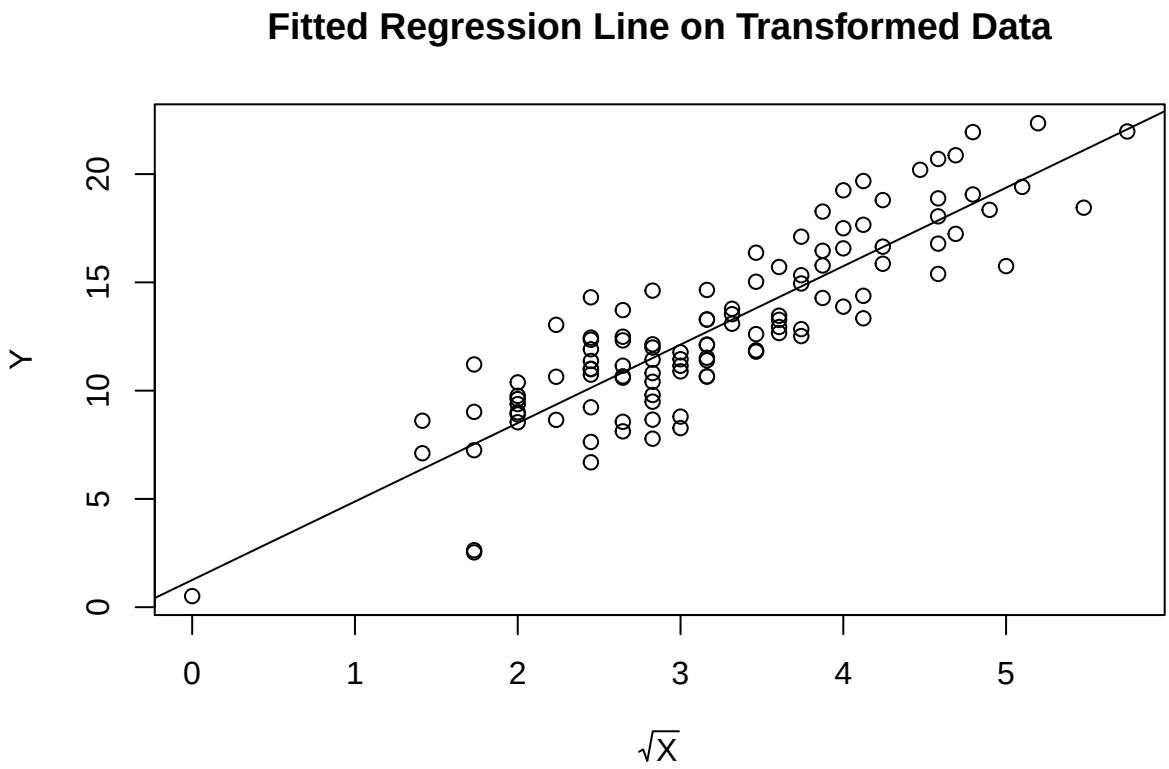
```
##
## Call:
```

```
## lm(formula = Y ~ X1, data = prod_time_data2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0008 -1.2161  0.0383  1.3367  4.1795
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.2547     0.6389   1.964   0.0521 .
## X1            3.6235     0.1895  19.124   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.99 on 109 degrees of freedom
## Multiple R-squared:  0.7704, Adjusted R-squared:  0.7683
## F-statistic: 365.7 on 1 and 109 DF,  p-value: < 2.2e-16
```

The regression function on *transformed data*: $Y = 1.2547 + 3.6235 * X1$

(c)

```
par(mfrow=c(1,1))
plot(prod_time_data2$X1, prod_time_data2$Y, xlab=expression(sqrt(X)), ylab="Y")
title(main="Fitted Regression Line on Transformed Data")
abline(lm_prod)
```

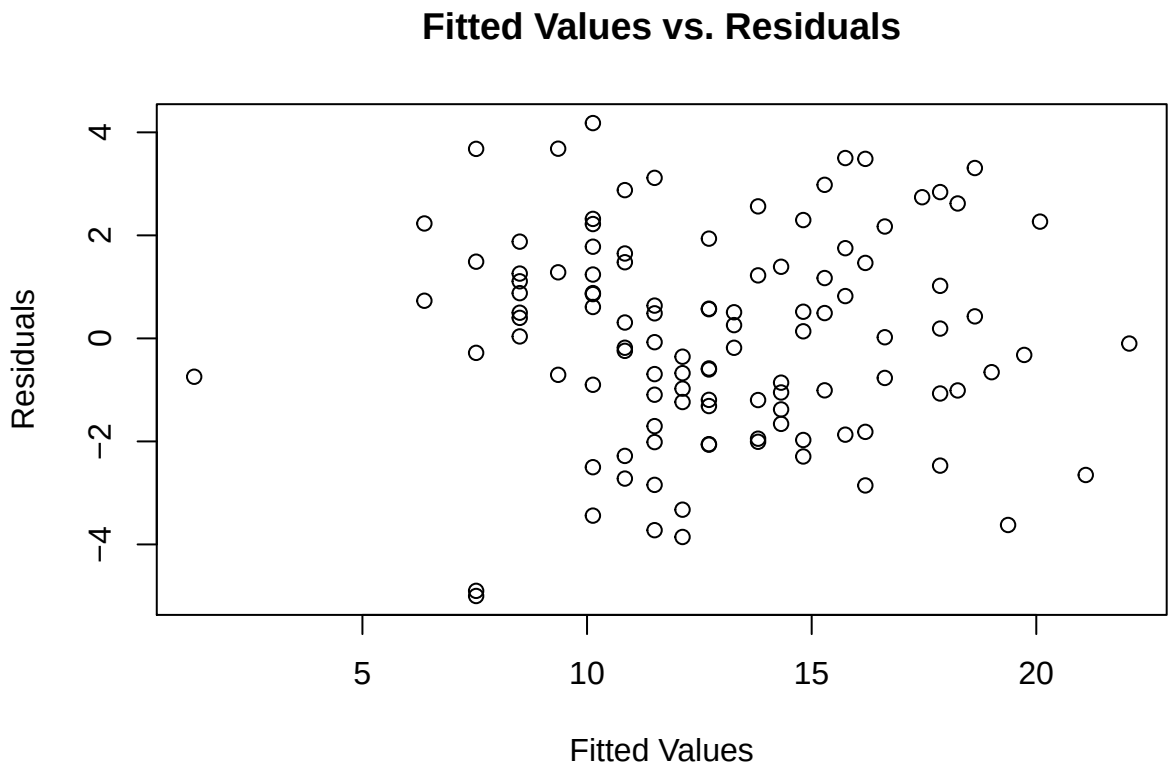


Interpretation:

Based on the scatter plot, the regression line appears to be a good fit on transformed data. Looking at the summary, we can also see that the $R^2 = 0.77$.

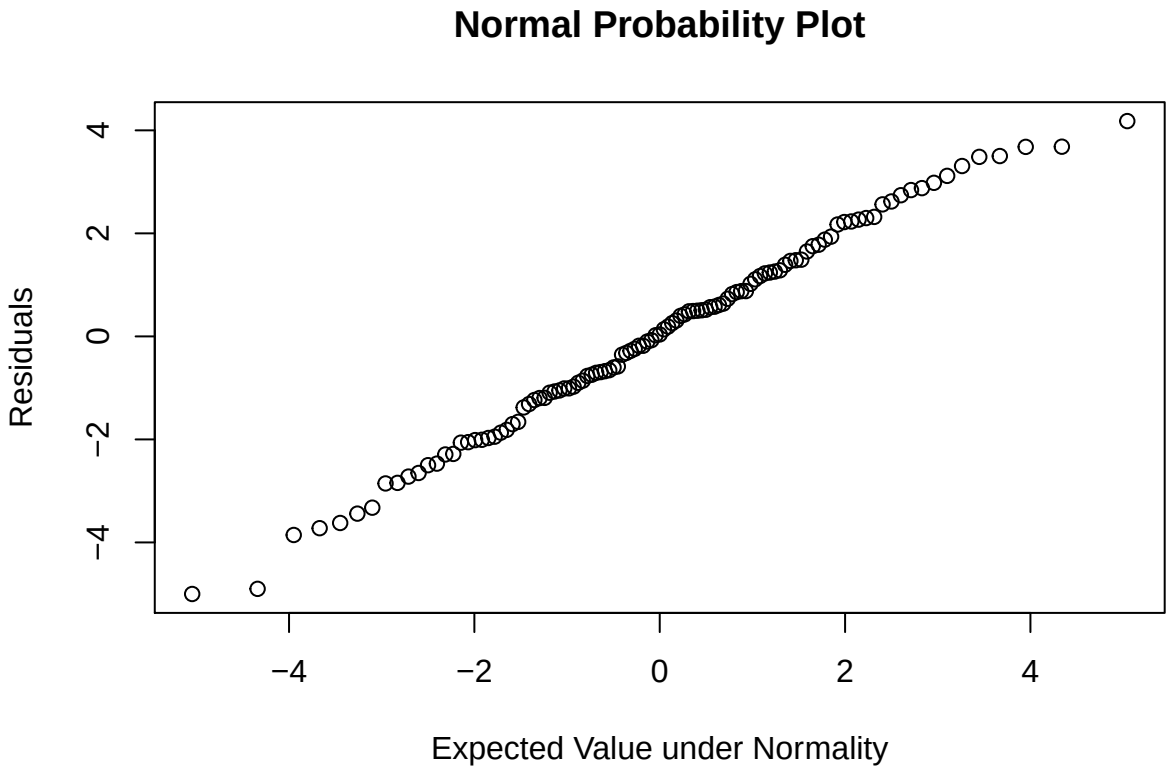
(d)

```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_prod, df=prod_time_data2, rse=1.99)
```




```
##
##  Pearson's product-moment correlation
##
## data:  zr1 and ei
```

```
## t = 136.99, df = 109, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9957863 0.9980161
## sample estimates:
##       cor
## 0.9971084
```



Interpretation:

Fitted vs. Residual Plot: The residuals appear to be equally spread and have no distinct patterns. Although there seem to be a few outliers. We can say that there is contant variance in the error term.

Normal Probability Plot: The plot seems to be almost linear, which means that the error is in agreement with the normality.

(e)

The regression function in *original units*: $Y = 1.2547 + 3.6235 * \sqrt{X}$

Solution 2:

(a)

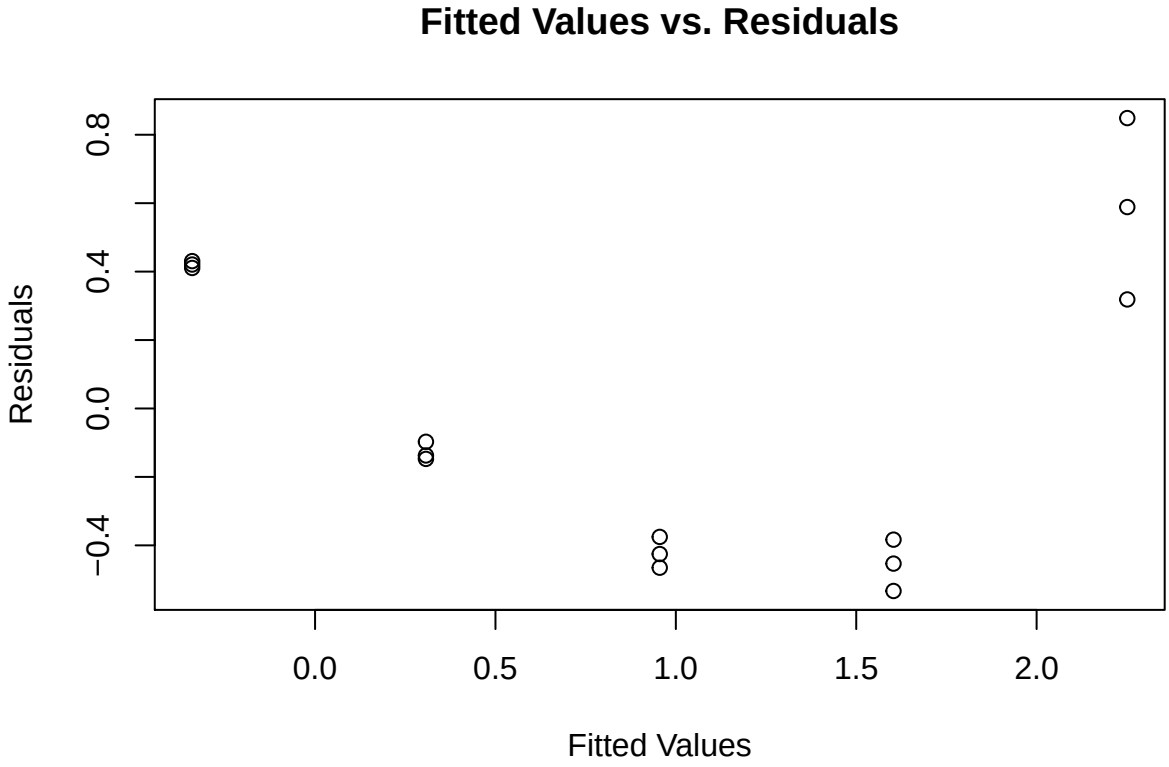
```
solution_data = read.csv("Solution Concentration.csv")
lm_soln = lm(Y~X, data=solution_data)
summary(lm_soln)
```

```
##
## Call:
## lm(formula = Y ~ X, data = solution_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

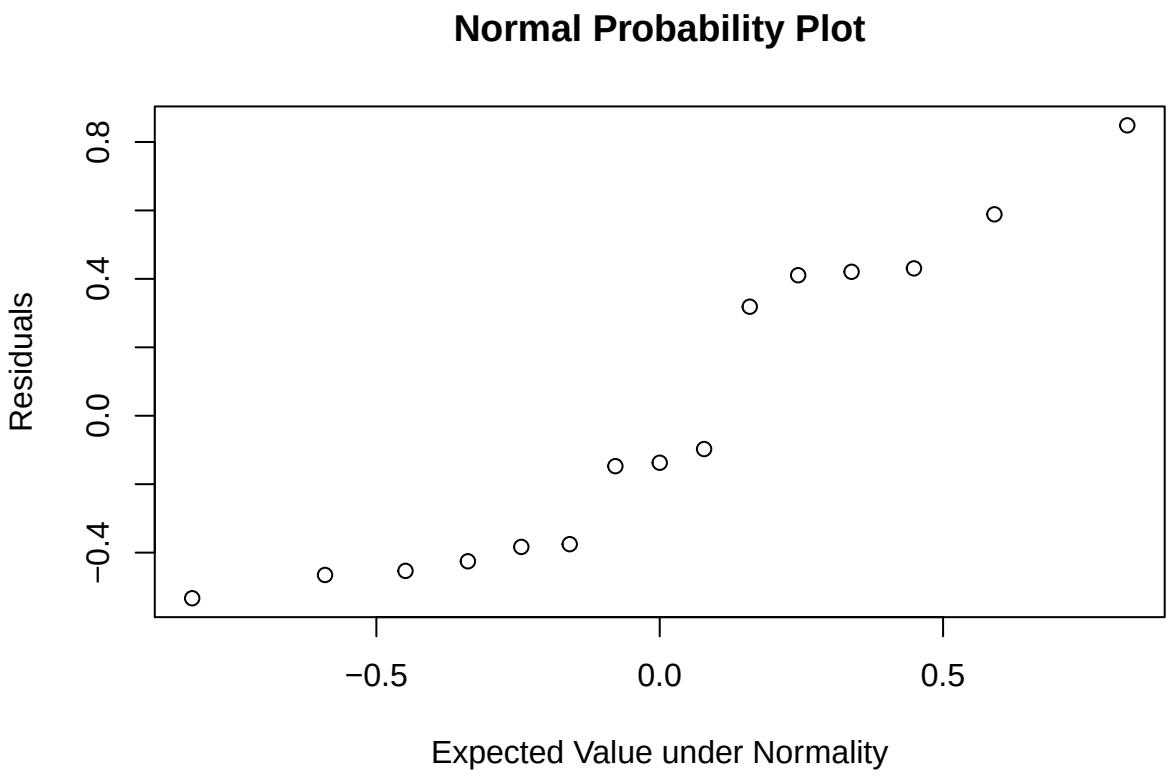
```
## -0.5333 -0.4043 -0.1373  0.4157  0.8487
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.5753     0.2487  10.354 1.20e-07 ***
## X            -0.3240     0.0433  -7.483 4.61e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4743 on 13 degrees of freedom
## Multiple R-squared:  0.8116, Adjusted R-squared:  0.7971
## F-statistic: 55.99 on 1 and 13 DF,  p-value: 4.611e-06
```

The regression function for *original data*: $Y = 2.5753 - 0.3240 * X$

```
build_residual_qq(lm=lm_soln, df=solution_data, rse=0.4743)
```




```
##
##  Pearson's product-moment correlation
##
## data:  zr1 and ei
## t = 10.974, df = 13, p-value = 6.057e-08
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8528129 0.9836088
## sample estimates:
##      cor
## 0.950038
```



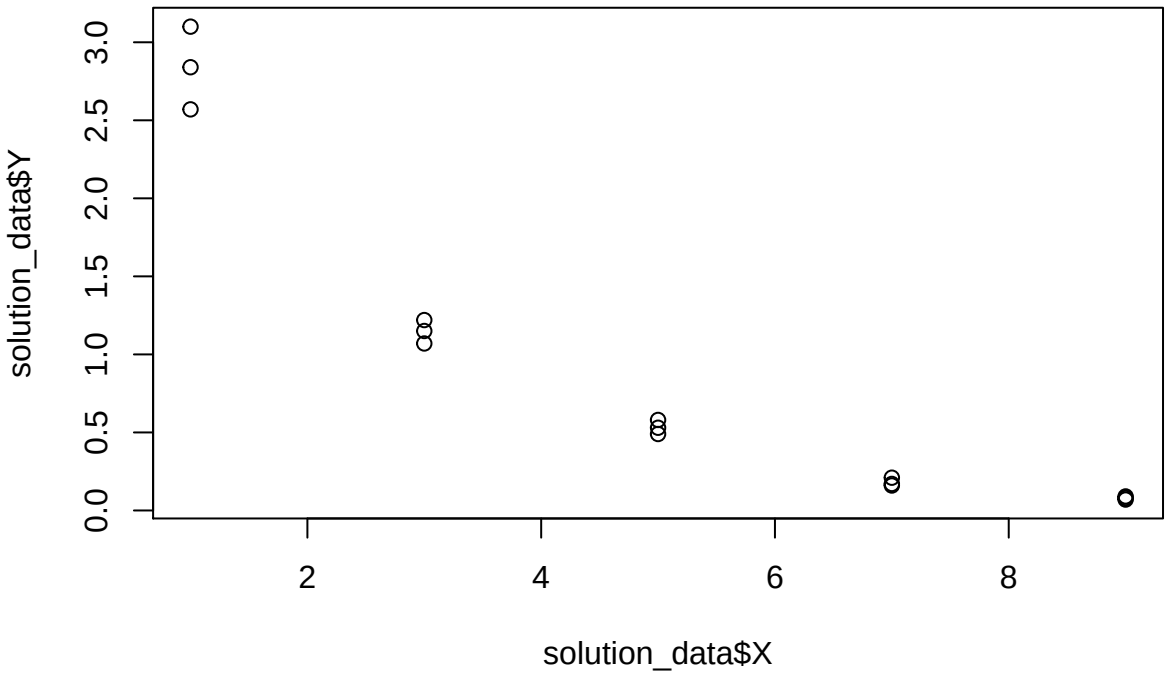
Interpretation:

Fitted vs. Residual Plot: The residuals are not equally spread and have a clear distinct pattern. Thus, we can say that, the error term does not have constant variance.

Normal Probability Plot: The plot seems to be non-linear, which means that the error is not in agreement with the normality.

(b)

```
plot(solution_data$X, solution_data$Y)
```

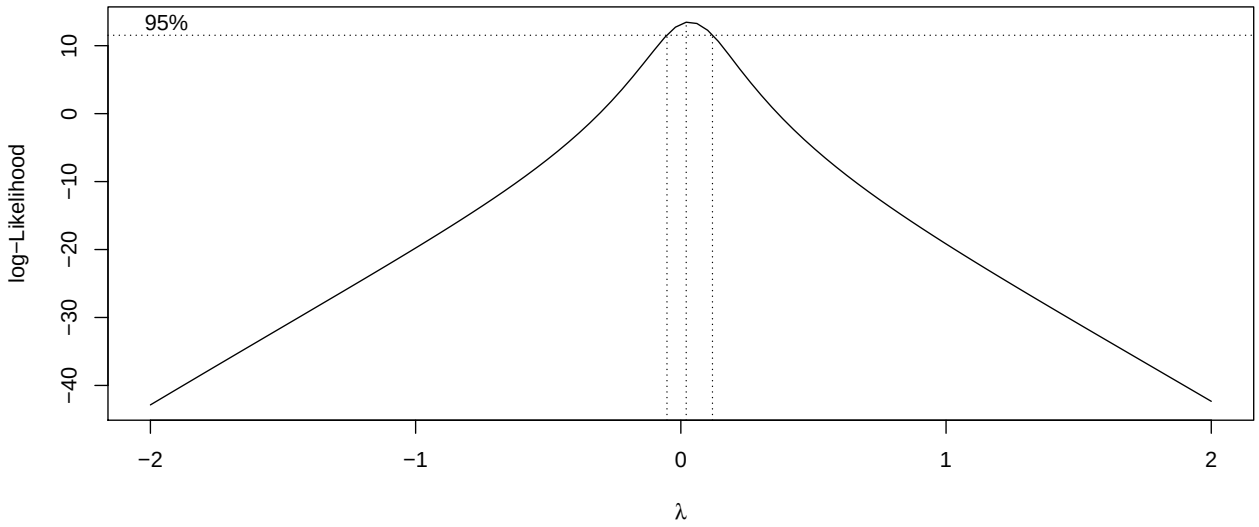


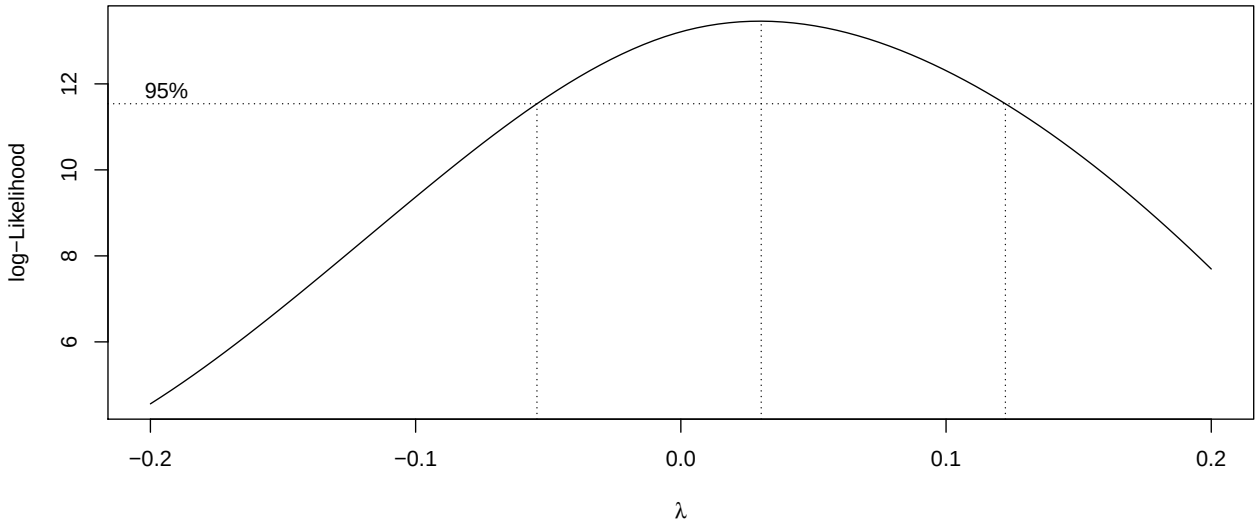
Interpretation:

Since the value of Y seems to be decreasing with the value of X and then smoothing out eventually, it seems like a logarithmic or exponential function. Thus I would like to try to transform Y to log(Y).

(c)

```
library(MASS)
par(mfrow=c(2,1))
boxcox(lm_soln)
boxcox(lm_soln, lambda=c(-.2,-.1,0, .1, .2))
```





Interpretation:

The suggested Y transformation with Box-Cox method is: $\lambda \approx 0$. Thus, we'll assume the suggested $\lambda = 0$, which implies the suggested transformation is: $Y' = log(Y)$.

(d)

```
Y1 = log(solution_data$Y)
solution_data = cbind(solution_data, Y1)
```

```
lm_soln_t = lm(Y1~X, data=solution_data)
summary(lm_soln_t)
```

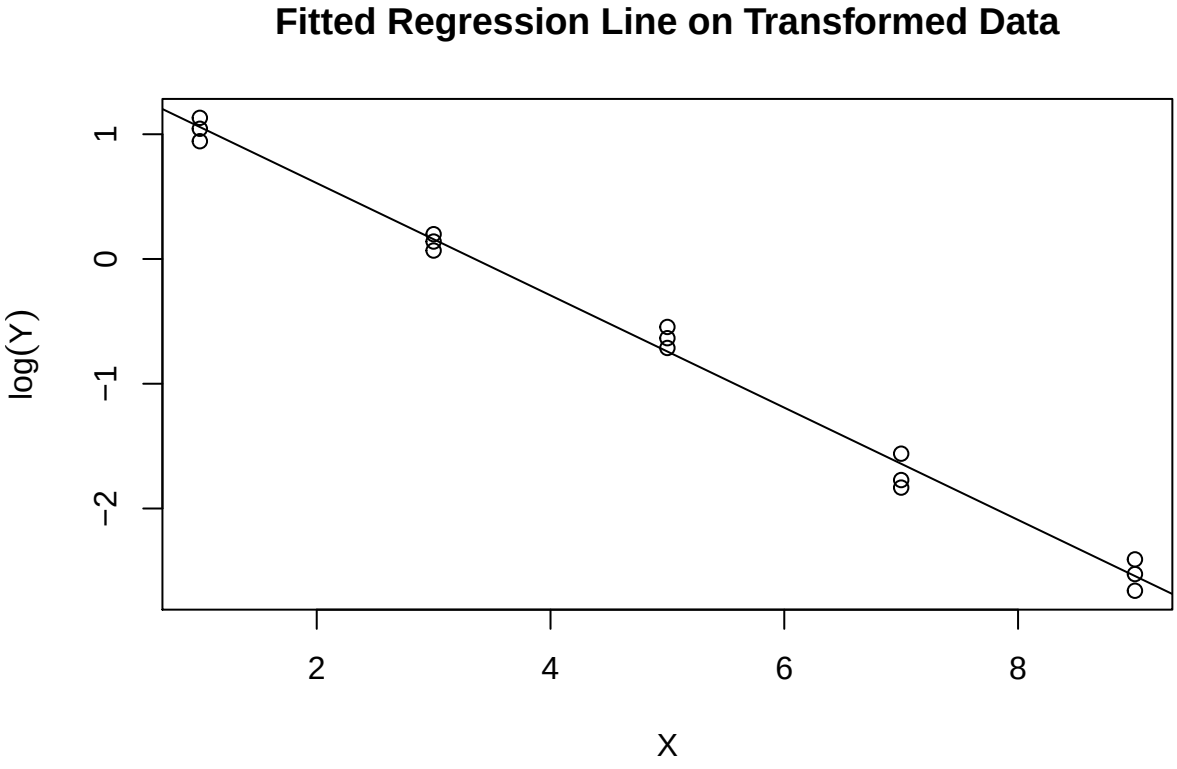
```
##
## Call:
## lm(formula = Y1 ~ X, data = solution_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.19102 -0.10228  0.01569  0.07716  0.19699
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.50792    0.06028   25.01 2.22e-12 ***
## X           -0.44993    0.01049  -42.88 2.19e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.115 on 13 degrees of freedom
## Multiple R-squared:  0.993,  Adjusted R-squared:  0.9924
## F-statistic:  1838 on 1 and 13 DF,  p-value: 2.188e-15
```

The regression function with *transformed data*: $Y' = 1.50792 - 0.44993*X $

(e)

```
par(mfrow=c(1,1))
plot(solution_data$X, solution_data$Y1, xlab="X", ylab=expression(log(Y)))
abline(lm_soln_t)
title(main="Fitted Regression Line on Transformed Data")
```

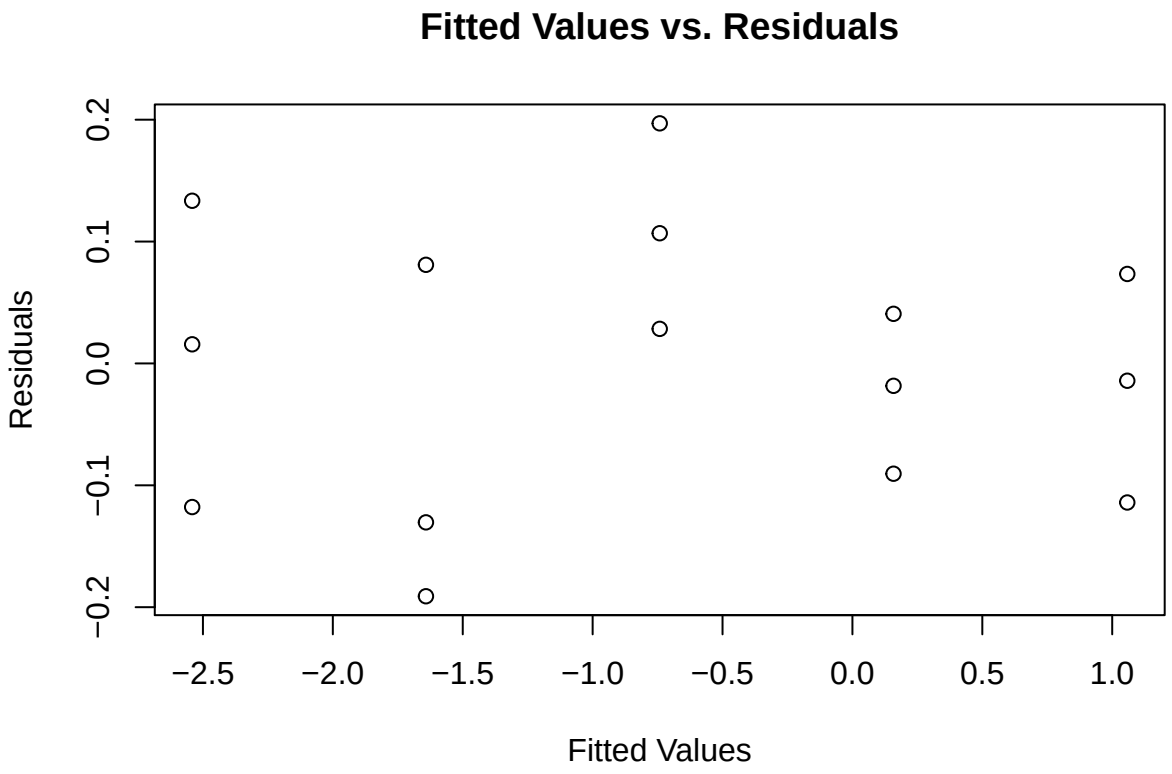


Interpretation:

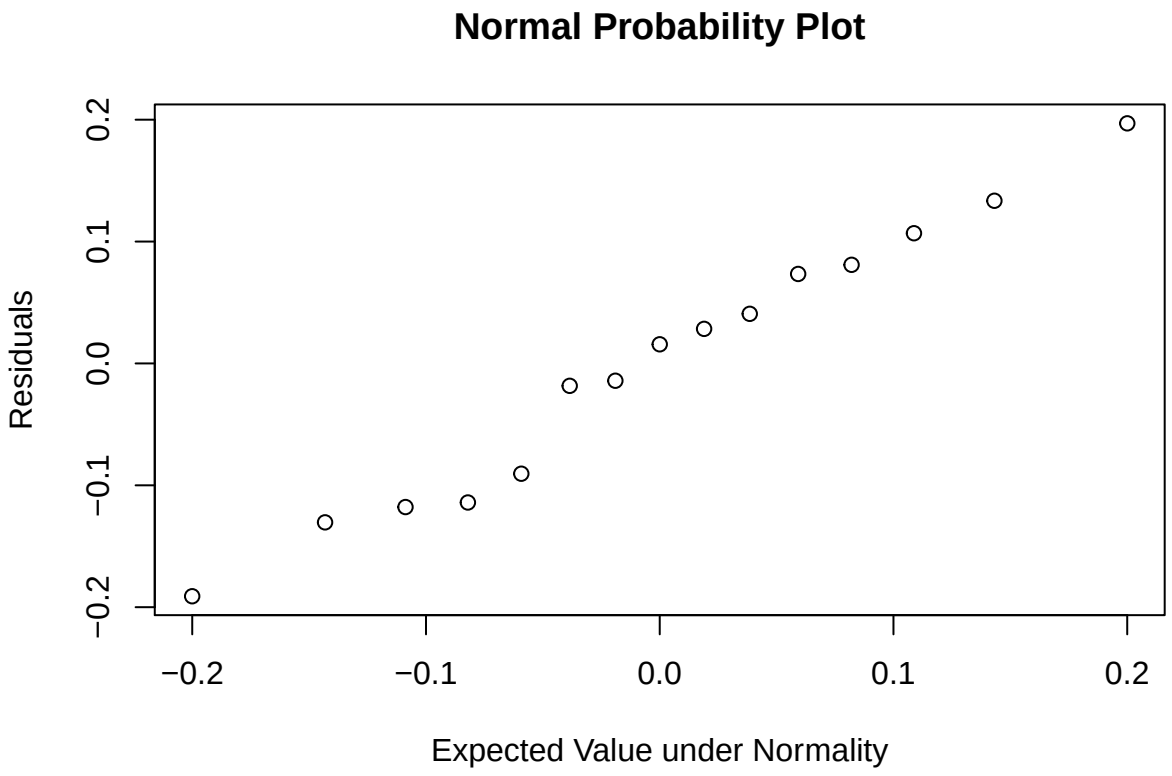
Based on the scatter plot, the regression line appears to be a good fit on transformed data. Looking at the summary, we can also see that the $R^2 = 0.993$.

(f)

```
build_residual_qq(lm=lm_soln_t, df=solution_data, rse=0.115)
```




```
##
##  Pearson's product-moment correlation
##
## data:  zr1 and ei
## t = 25.353, df = 13, p-value = 1.871e-12
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9694338 0.9967764
## sample estimates:
##       cor
## 0.9900386
```



Interpretation:

Fitted vs. Residual Plot: The residuals are equally spread and don't have a pattern. Thus, we can say that, the error term has a constant variance.

Normal Probability Plot: The plot seems to be non-linear, which means that the error is not in agreement with the normality.

(g)

The regression function with transformed data (in original units): $\log Y = 1.50792 - 0.44993 * X$

Solution 3:

(a)

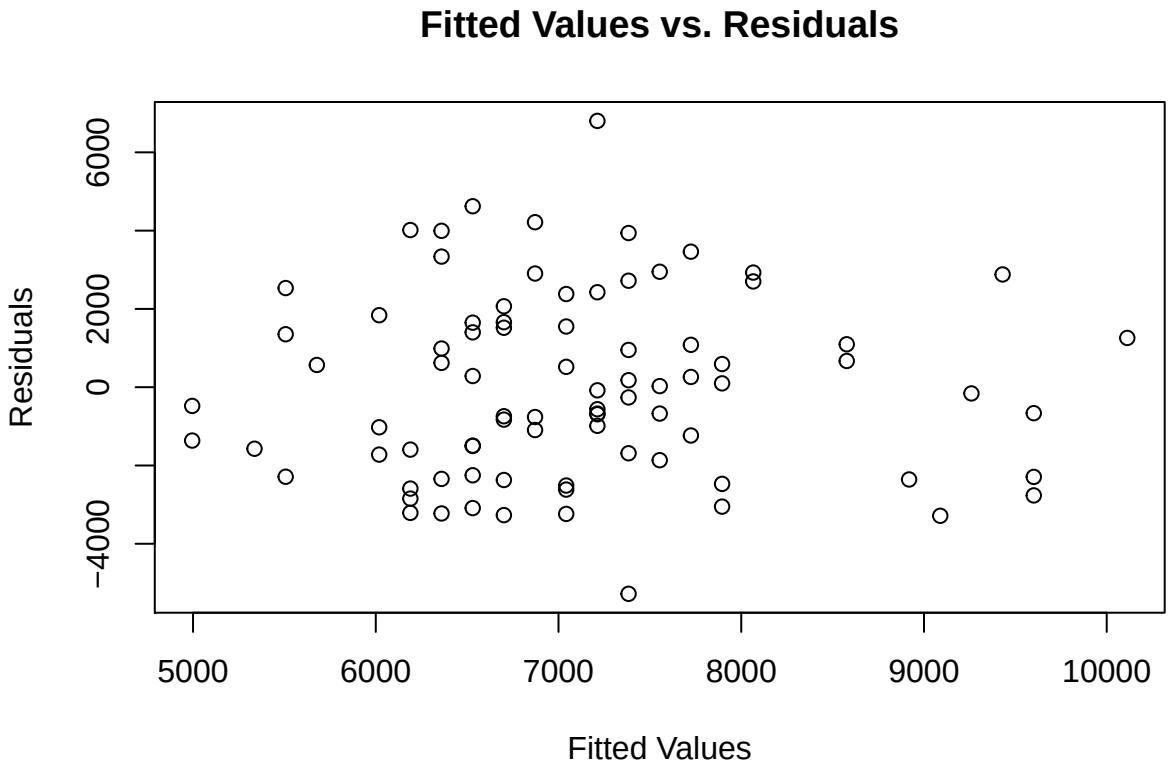
```
crime_data = read.csv("Crime Rate.csv")
lm_crime = lm(Y~X, data=crime_data)
summary(lm_crime)
```

```
##
## Call:
## lm(formula = Y ~ X, data = crime_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5278.3 -1757.5  -210.5  1575.3  6803.3
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20517.60    3277.64   6.260 1.67e-08 ***
## X            -170.58      41.57  -4.103 9.57e-05 ***
## ---
```

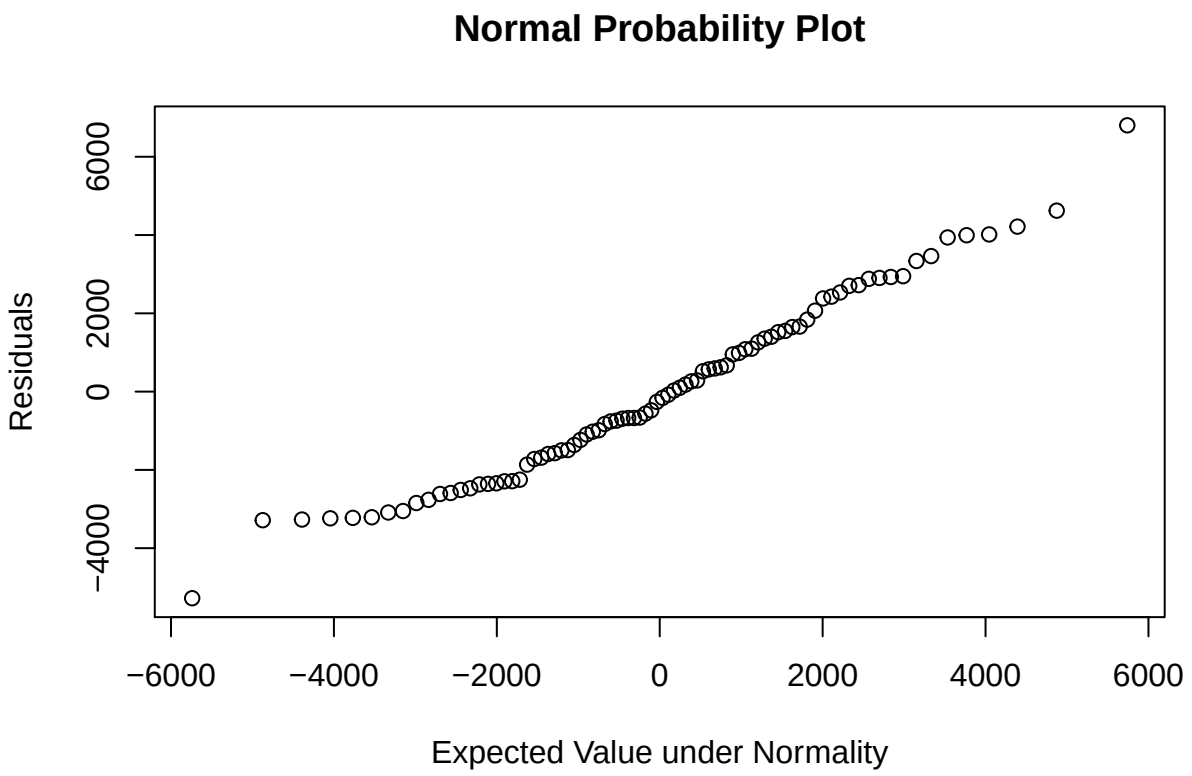
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2356 on 82 degrees of freedom
## Multiple R-squared:  0.1703, Adjusted R-squared:  0.1602
## F-statistic: 16.83 on 1 and 82 DF,  p-value: 9.571e-05
```

The regression function: $Y = 20517.60 - 170.58 * X$

```
build_residual_qq(lm=lm_crime, df=crime_data, rse=2356)
```




```
##
##  Pearson's product-moment correlation
##
## data:  zr1 and ei
## t = 59.883, df = 82, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9826766 0.9927136
## sample estimates:
##       cor
## 0.9887589
```



Interpretation:

Fitted vs. Residual Plot: The residuals are not equally spread and have some pattern. Thus, we can say that, the error term does not have constant variance.

Normal Probability Plot: The plot seems to be s-shaped with heavy tails, which means that the error is not in agreement with normality.

(b)

Brown-Forsythe Test

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

```
summary(crime_data$X)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   61.00   76.00   79.00   78.60   82.25   91.00
```

```
ei = lm_crime$residuals
df = data.frame(cbind(crime_data$Y,crime_data$X,ei))
df1 = df[df[,2]<=69,]
df2 = df[df[,2]>69,]

med1 = median(df1[,3])
med2 = median(df2[,3])

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

```
## [1] 8
```

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

```
## [1] 76
```

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

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

```
## [1] 1751.872
```

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

```
## [1] 1927.083
```

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

```
## [1] 1762978
```

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

```
## [1] 1327.772
```

```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] -0.3550185
```

```r
t = qt(1-0.05, 118)
print(t)
```

```
## [1] 1.65787
```

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 $|1.957763| > 1.65787$ i.e. $|testStatistic| > t(1-\alpha/2, n-2)$, we conclude H_1. The error variance is not constant and thus varies with X.

The conclusion supports the preliminary findings in part (a).

Note: The problem statement asks us to divide the dataset between X 69 and X > 69, however, the mean of X is 79. As confirmed on piazza, we can use either 69 or 79 as medians.

(c)

Breusch-Pagan Test

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

```r
ei2 = ei^2
f = lm(ei2~crime_data$X)
summary(f)
```

```
##
## Call:
## lm(formula = ei2 ~ crime_data$X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5407843 -4777840 -2620854  2430624 40870211
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4664945    9817145   0.475    0.636
## crime_data$X     9606     124523   0.077    0.939
##
## Residual standard error: 7058000 on 82 degrees of freedom
## Multiple R-squared:  7.257e-05,  Adjusted R-squared:  -0.01212
## F-statistic: 0.005951 on 1 and 82 DF,  p-value: 0.9387
```

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

```
## Analysis of Variance Table
##
## Response: ei2
##              Df     Sum Sq    Mean Sq F value Pr(>F)
## crime_data$X  1 2.9640e+11 2.9640e+11   0.006 0.9387
## Residuals    82 4.0843e+15 4.9809e+13
```

```r
#to find SSE(F) and SSR(F)
anova(lm_crime)
```

```
## Analysis of Variance Table
##
## Response: Y
##           Df    Sum Sq  Mean Sq F value    Pr(>F)
## X          1  93462942 93462942  16.834 9.571e-05 ***
## Residuals 82 455273165  5552112
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
SSR_R = 2.9640e+11
SSE_R = 4.0843e+15

SSR_F = 93462942
SSE_F= 455273165

n = nrow(crime_data)

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

```
## [1] 0.005045017
```

```r
#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 $0.005045017 \leq 3.841459$ i.e. $chiTest \leq \chi^2(1-\alpha,1)$, we conclude H_0. The error variance is constant.

This conclusion is inconsistent with the conclusions in part(a) and part(b) (using 69 as median).

Solution 4:

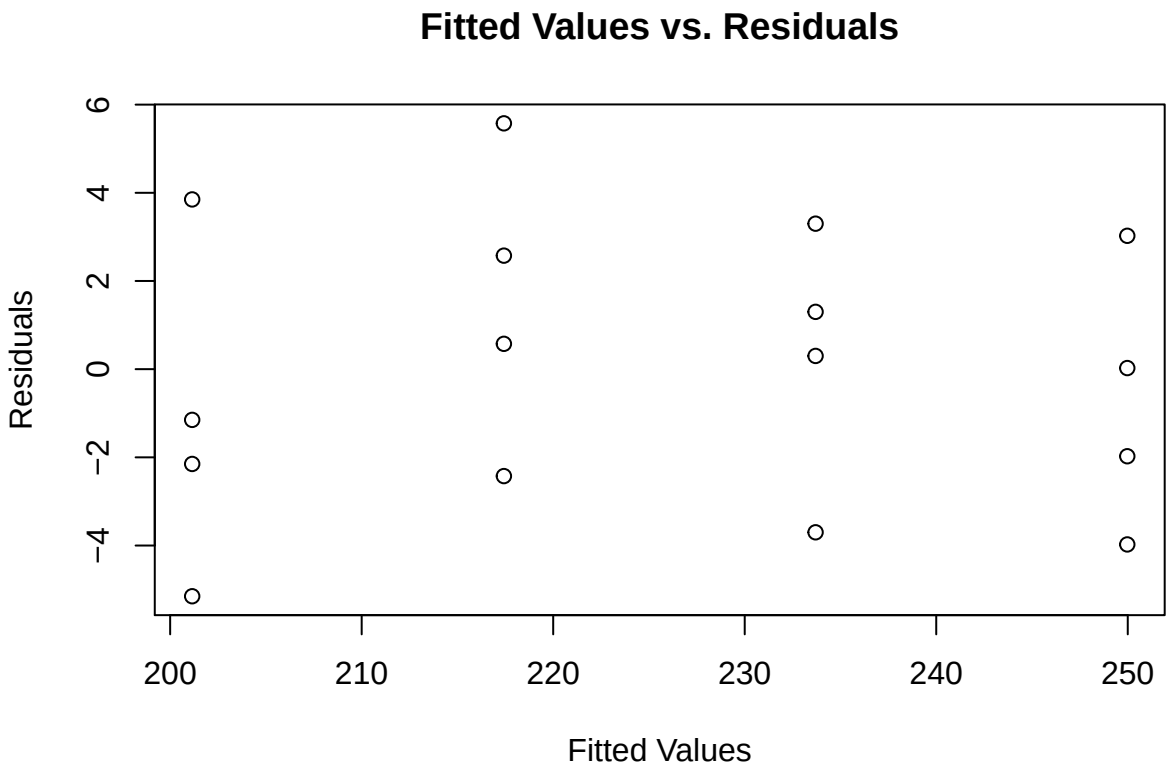
(a)

```r
plastic_data = read.csv("Plastic Hardness.csv")
lm_plastic = lm(Y~X, data=plastic_data)
summary(lm_plastic)
```

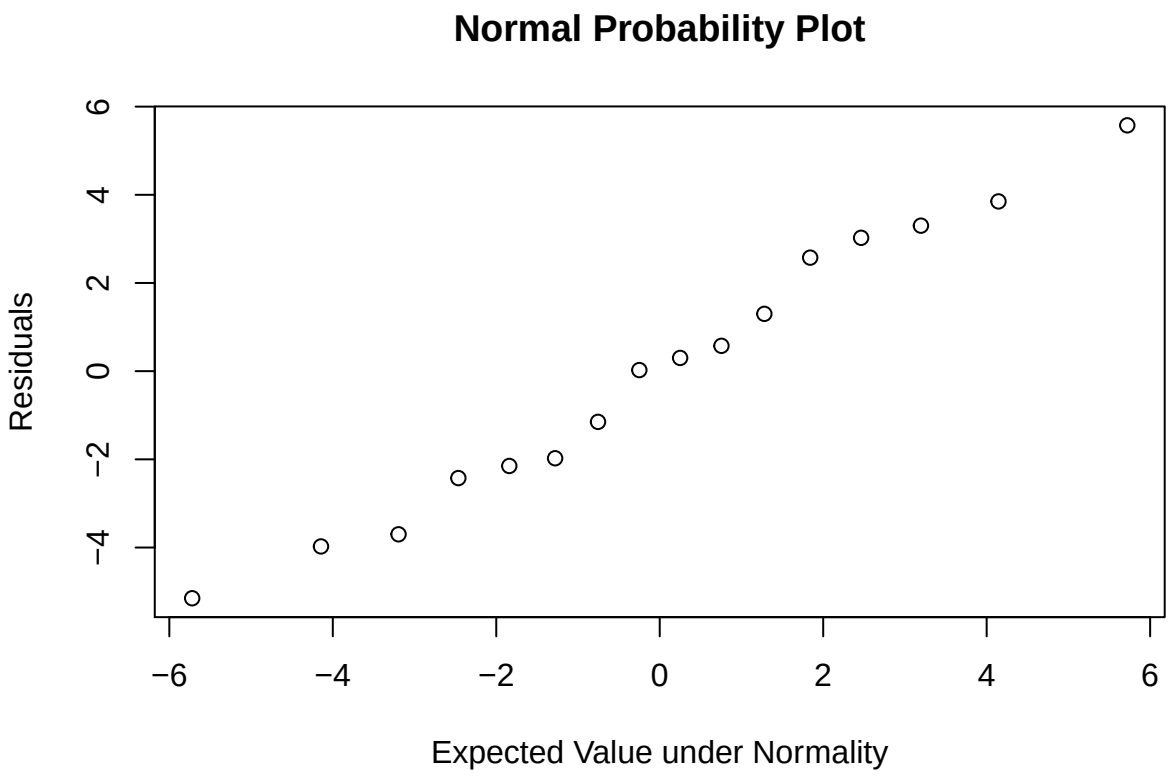
```
##
## Call:
## lm(formula = Y ~ X, data = plastic_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1500 -2.2188  0.1625  2.6875  5.5750
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 168.60000    2.65702   63.45  < 2e-16 ***
## X             2.03438    0.09039   22.51 2.16e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.234 on 14 degrees of freedom
## Multiple R-squared:  0.9731, Adjusted R-squared:  0.9712
## F-statistic: 506.5 on 1 and 14 DF,  p-value: 2.159e-12
```

Regression Function: $Y = 168.6 + 2.03438 * X$

```r
build_residual_qq(lm=lm_plastic, df=plastic_data, rse=3.234)
```




```
##
##  Pearson's product-moment correlation
##
## data:  zr1 and ei
## t = 28.813, df = 14, p-value = 7.28e-14
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9755039 0.9971848
## sample estimates:
##       cor
## 0.9916733
```



Interpretation:

Fitted vs. Residual Plot: The residuals are equally spread and don't have a pattern. Thus, we can say that, the error term has a constant variance.

Normal Probability Plot: The plot seems to be non-linear, which means that the error is not in agreement with the normality.

(b)

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

```
##                2.5 %    97.5 %
## (Intercept) 162.9013 174.29875
## X             1.8405   2.22825
```

Interpretation:

We can say with 90% family confidence coefficient that both of the above intervals for β_0 and β_1 are correct based on the given sample.

(c)

```
mean(plastic_data$X)
```

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

Thus, $\bar{X} > 0$ which means that β_0 and β_1 are negatively correlated. This is to balance the effect of one coefficient, on the response, with the other coefficient. So, for example, if β_1 is too high, β_0 is likely to be too low to balance out the effect of β_1 on Y.

The joint confidence intervals in part (b) do support this view.

(d)

```
alpha = 0.1
Xh = data.frame(X=c(20,30,40))
CI = predict(lm_plastic, Xh, se.fit=TRUE, level=1-alpha)
B = qt(1-alpha/(2*nrow(Xh)), lm_plastic$df)

est_resp_CI = t(
rbind(
"Xh" = array(t(Xh)),
"fit" = array(CI$fit),
"lower.B" = array(CI$fit-B* CI$se.fit),
"upper.B" = array(CI$fit+B* CI$se.fit)
)
)

est_resp_CI
```

```
##      Xh      fit  lower.B  upper.B
## [1,] 20 209.2875 206.7277 211.8473
## [2,] 30 229.6312 227.6762 231.5863
## [3,] 40 249.9750 246.7824 253.1676
```

Interpretation:

Family confidence coefficient means that the obtained confidence intervals, for several mean responses, are simultaneously accurate with a confidence coefficient of $1 - \alpha$.

(e)

```
Xh = data.frame(X=c(30,40))
g = nrow(Xh)

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

print(B)
```

```
## [1] 2.144787
```

```
print(S)
```

```
## [1] 2.335152
```

Interpretation:

Thus, we can see that the most efficient procedure is the Bonferroni using t-distribution (compared to Scheffe using F-distribution) as it will yeild tighter linits (since $B < S$).

```
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,] 30 3.338457 229.6312 222.4710 236.7915
## [2,] 40 3.505601 249.9750 242.4562 257.4938
```

Solution 5:

```
cdi = read.csv("CDI.csv")
lm_cdi = lm(Number.of.active.physicians~Total.population, data=cdi)
summary(lm_cdi)
```

```
##
## Call:
## lm(formula = Number.of.active.physicians ~ Total.population,
##     data = cdi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1969.4  -209.2   -88.0    27.9  3928.7
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1.106e+02  3.475e+01  -3.184  0.00156 **
## Total.population  2.795e-03  4.837e-05  57.793  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 610.1 on 438 degrees of freedom
## Multiple R-squared:  0.8841, Adjusted R-squared:  0.8838
## F-statistic:  3340 on 1 and 438 DF,  p-value: < 2.2e-16
```

(a)

```
confint(lm_cdi, level=1-0.05/2)
```

```
##                          1.25 %       98.75 %
## (Intercept)      -1.887833e+02 -32.486285498
## Total.population  2.686636e-03   0.002904214
```

(b)

Both the values suggested by the investigator, $\beta_1 = -100$ and $\beta_0 = 0.0028$, fall within the 95% joint confidence intervals obtained in part(a). Thus, the results in part(a) support the view of the investigator.

(c)

```
Xh = data.frame(Total.population=c(500,1000,5000))
g = nrow(Xh)
alpha = 0.1

CI = predict(lm_cdi, Xh, se.fit= TRUE, level = 1-alpha)
B = qt(1 -alpha / (2*g), lm_cdi$df)
W = sqrt(2*qf(1-alpha, 2, lm_cdi$df))

print(B)
```

```
## [1] 2.134781
```

```
print(W)
```

```
## [1] 2.151619
```

Interpretation:

Thus, we can see that the most efficient procedure is the Bonferroni using t-distribution (compared to Working-Hotelling using F-distribution) as it will yeild tighter limits (since $B < W$).

(d)

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

est_resp_CI
```

```
##        Xh   s.pred        fit   lower.B   upper.B
## [1,]  500 34.73280 -109.23706 -183.3840 -35.09015
## [2,] 1000 34.71958 -107.83935 -181.9581 -33.72064
## [3,] 5000 34.61430  -96.65765 -170.5516 -22.76370
```

Interpretation:

We can say with 90% family confidence coefficient that all of the above intervals are correct based on the given sample. However, we see that the predicted response (and the intervals) suggest negative values for Number of active physicians which is not practically possible. Thus, our model is not a good fit for our data for counties extremely low values of Total Population.

Solution 6:

(a)

```
senic_data = read.csv("SENIC.csv")
colnames(senic_data)
```

```
##  [1] "Length.of.stay"
##  [2] "Age"
##  [3] "Infection.risk"
##  [4] "Routine.culturing.ratio"
##  [5] "Routine.chest.X.ray.ratio"
##  [6] "Number.of.beds"
##  [7] "Medical.school.affiliation"
##  [8] "Region"
##  [9] "Average.daily.census"
## [10] "Number.of.nurses"
## [11] "Available.facilities.and.services"
```

```
reg_loop <- function(df, x_cols, y_str) {
  lm_regs = list({})
  for(i in 1:length(x_cols)){
    x_str = x_cols[i]
    formula = as.formula(paste(y_str, "~", x_str))
    lm_regs[[i]] = lm(formula, data=df)
```

```
    print(paste("Linear Regression Summary:", x_cols[i]))
    print(summary(lm_regs[[i]]))
  }
  lm_regs
}
```

```
x_cols = c("Infection.risk", "Available.facilities.and.services", "Routine.chest.X.ray.ratio")
y_str="Length.of.stay"
lm_fits = reg_loop(df=senic_data, x_cols=x_cols, y_str=y_str)
```

```
## [1] "Linear Regression Summary: Infection.risk"
##
## Call:
## lm(formula = formula, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0587 -0.7776 -0.1487  0.7159  8.2805
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      6.3368     0.5213  12.156  < 2e-16 ***
## Infection.risk   0.7604     0.1144   6.645 1.18e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.624 on 111 degrees of freedom
## Multiple R-squared:  0.2846, Adjusted R-squared:  0.2781
## F-statistic: 44.15 on 1 and 111 DF,  p-value: 1.177e-09
##
## [1] "Linear Regression Summary: Available.facilities.and.services"
##
## Call:
## lm(formula = formula, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2712 -1.0716 -0.2816  0.7584  9.5433
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       7.71877    0.51020  15.129  < 2e-16 ***
## Available.facilities.and.services  0.04471    0.01116   4.008 0.000111 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.795 on 111 degrees of freedom
## Multiple R-squared:  0.1264, Adjusted R-squared:  0.1185
## F-statistic: 16.06 on 1 and 111 DF,  p-value: 0.0001113
##
## [1] "Linear Regression Summary: Routine.chest.X.ray.ratio"
##
## Call:
## lm(formula = formula, data = df)
```

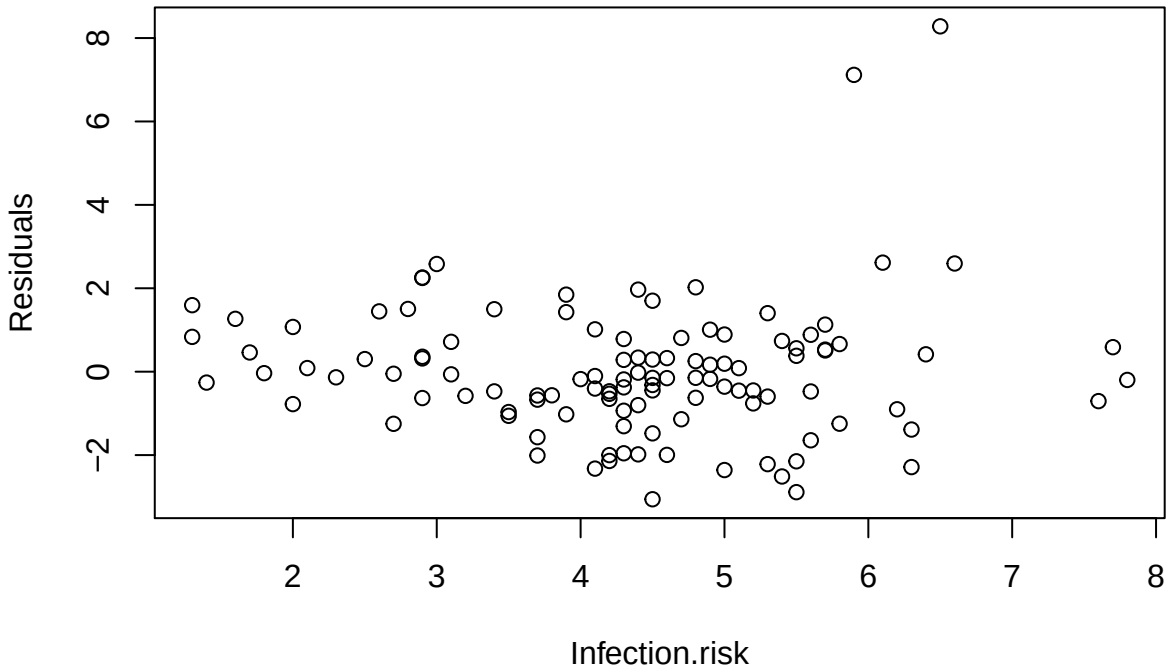
```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9226 -1.0810 -0.2708  0.8200  8.7008
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               6.566373   0.726094   9.043 5.67e-15 ***
## Routine.chest.X.ray.ratio 0.037756   0.008657   4.361 2.91e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.774 on 111 degrees of freedom
## Multiple R-squared:  0.1463, Adjusted R-squared:  0.1386
## F-statistic: 19.02 on 1 and 111 DF,  p-value: 2.906e-05
```

Three regression function are:

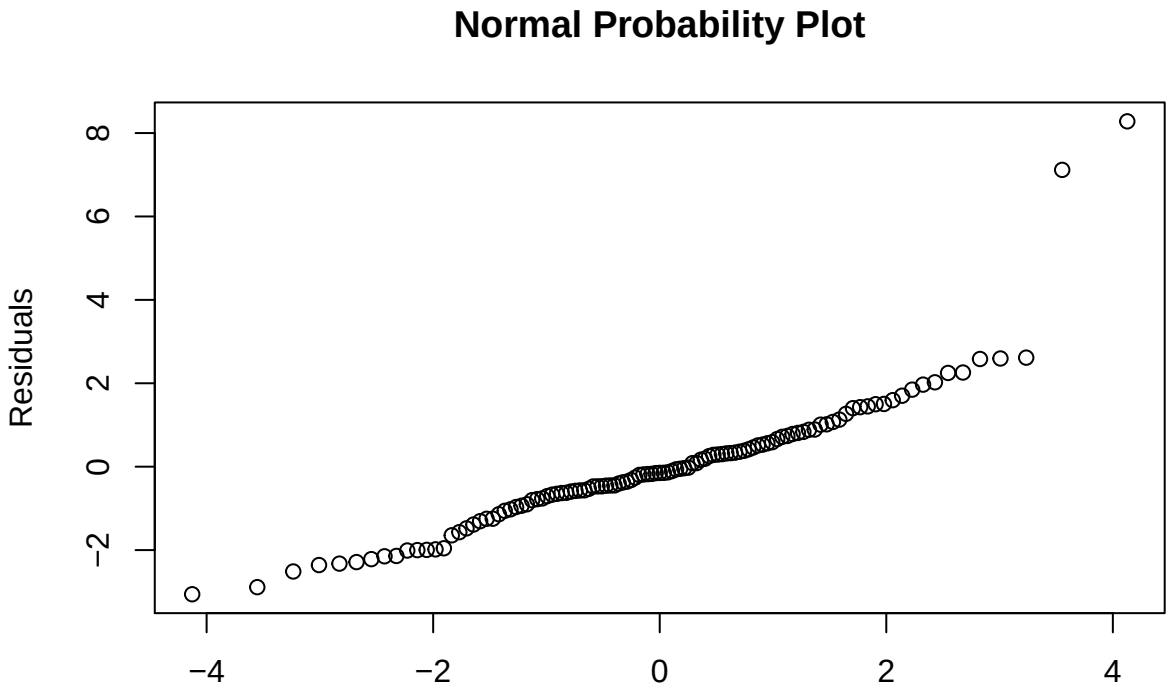
- $Length.of.stay = 6.3368 + 0.7604 * Infection.risk$
- $Length.of.stay = 7.71877 + 0.04471 * Available.facilities.and.services$
- $Length.of.stay = 6.566373 + 0.037756 * Routine.chest.X.ray.ratio$

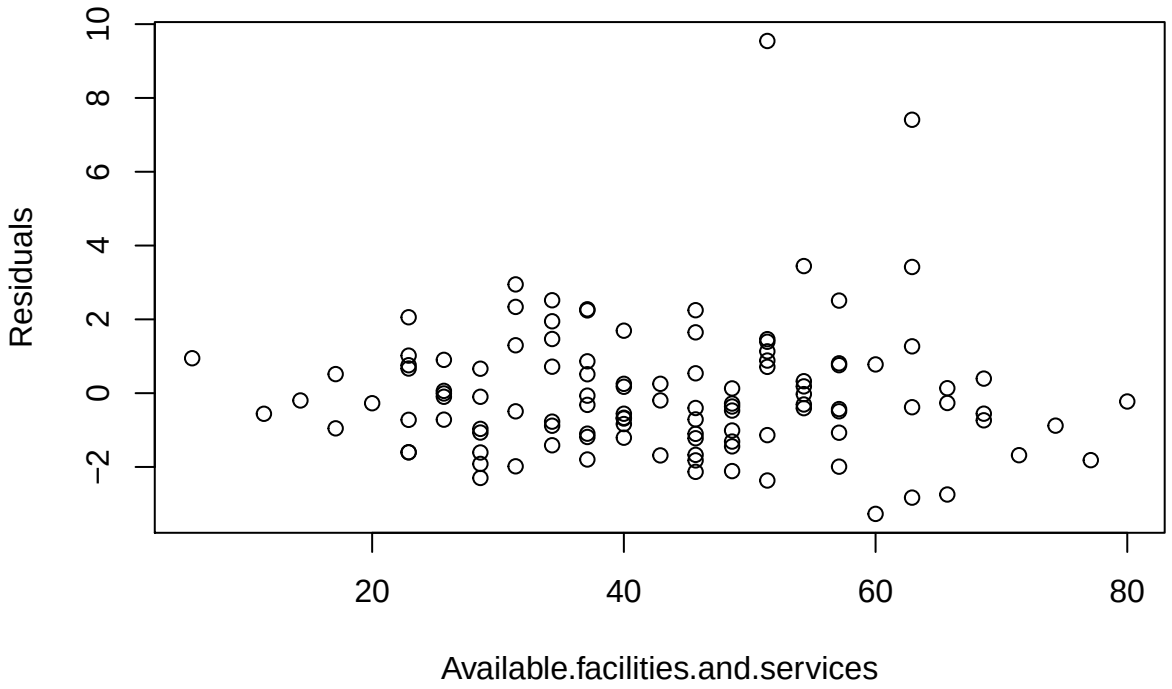
(b)

```r
rse = c(1.624, 1.795, 1.774)
for(i in 1:length(x_cols)){
  df = senic_data
  lm = lm_fits[[i]]
  ei = lm$residuals
  X = array(df[,x_cols[i]])

  par(mfrow=c(1,1))
  plot(X, ei, xlab=x_cols[i], 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[i]*qnorm(zr)

  print(cor.test(zr1, ei))

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

    }
```



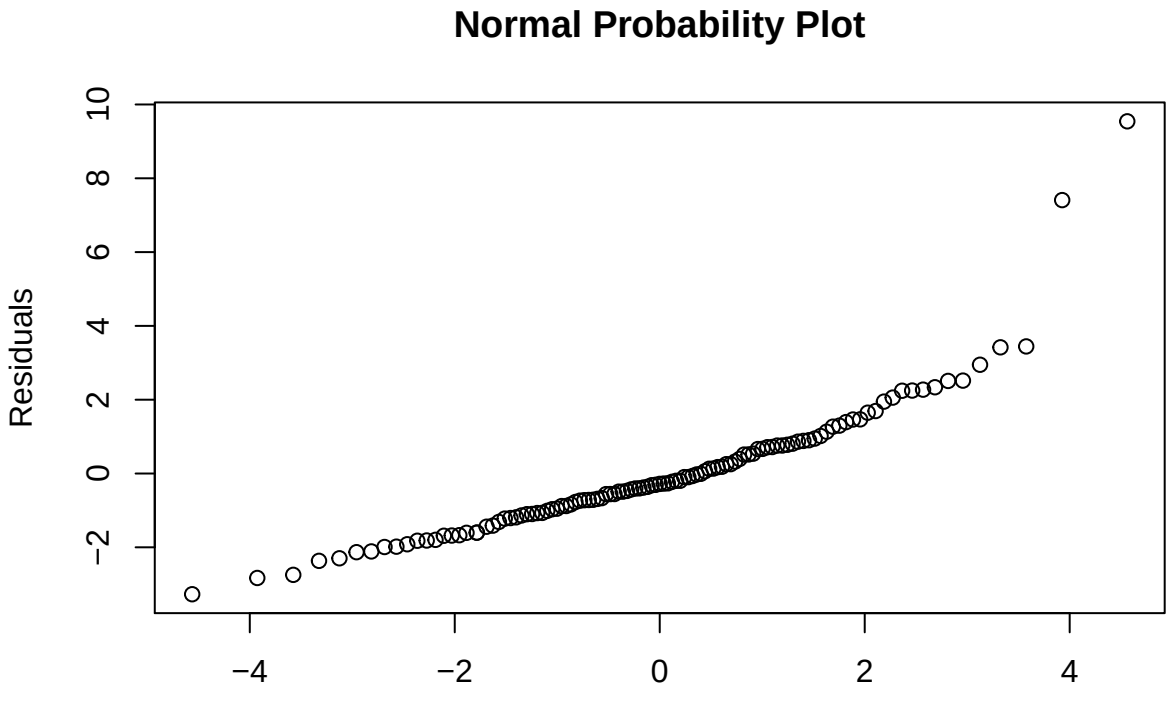

```
##
##  Pearson's product-moment correlation
##
## data:  zr1 and ei
## t = 26.318, df = 111, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8975754 0.9501531
## sample estimates:
##       cor
## 0.9283727
```

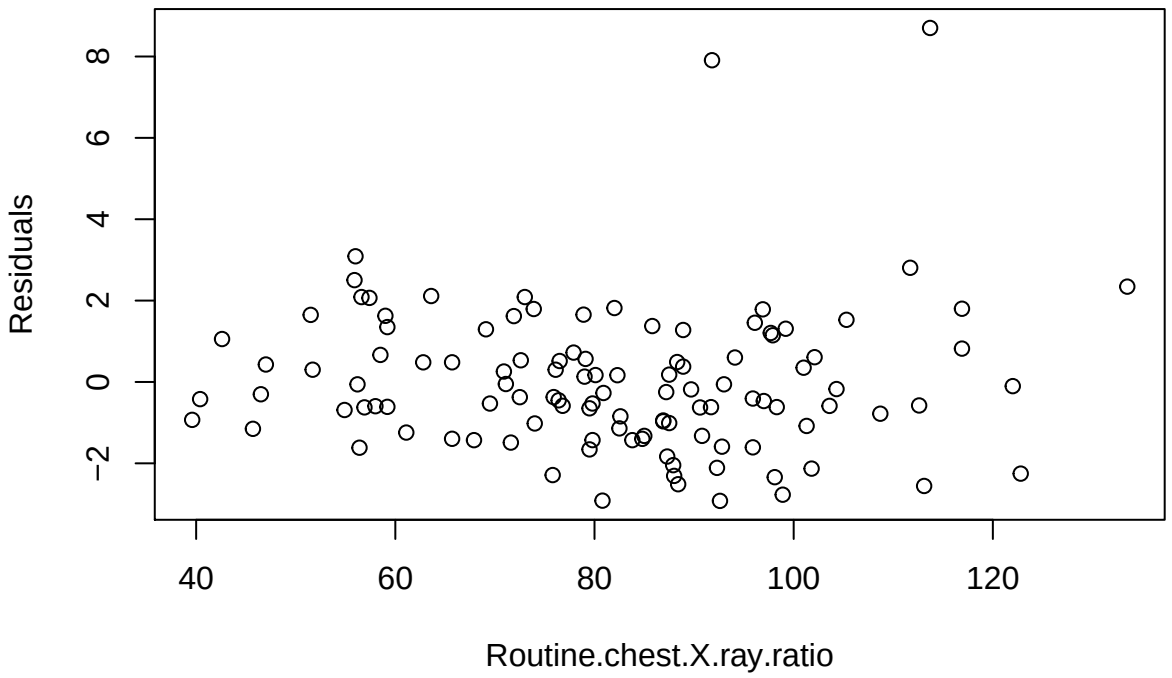




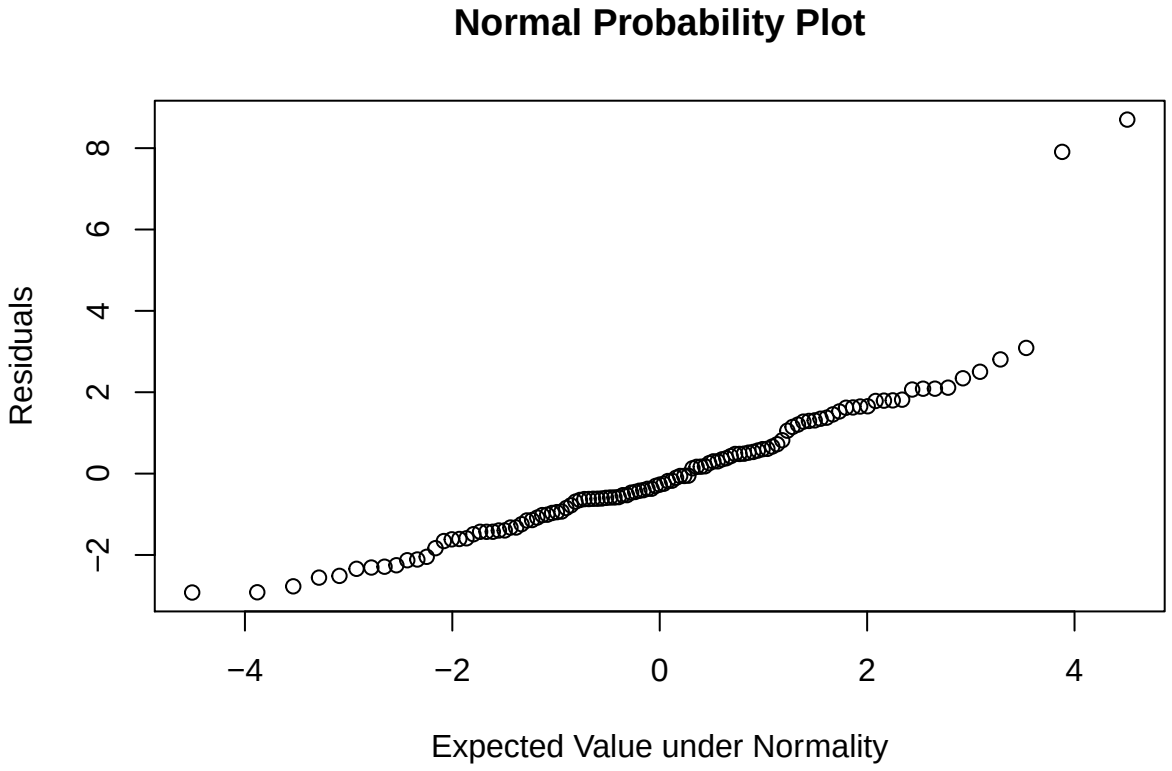

```
##
##  Pearson's product-moment correlation
##
## data:  zr1 and ei
## t = 25.3, df = 111, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
##  0.8902441 0.9464790
## sample estimates:
##       cor
## 0.9231565
```





```
##
##  Pearson's product-moment correlation
##
## data:  zr1 and ei
## t = 27.038, df = 111, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9023492 0.9525376
## sample estimates:
##       cor
## 0.9317627
```




```
for(i in 1:length(x_cols)){
  ei = lm_fits[[i]]$residuals
  print(which(ei>6))
}
```

```
##  47 112
##  47 112
##  47 112
##  47 112
##  47 112
##  47 112
```

Interpretation:

- For all three variables, we see that the residuals plotted against X and the normal probability plots show constant variance and conformity to normality, except for two outliers that have residuals > 6.
- As seen above, the outliers are the obervations: 47 and 112.

(c)

```r
senic_data2 = senic_data[-c(47,112),]
nrow(senic_data2)
```

```
## [1] 111
```

```r
lm_senic = lm(Length.of.stay~Infection.risk, data=senic_data2)
```

```r
Xh = data.frame(Infection.risk=c(6.5,5.9))
alpha = 0.05

CI.New.Ind = predict(lm_senic, Xh, se.fit= TRUE,interval = "prediction" ,level = 1-alpha)

CI.New.Ind
```

```
## $fit
##        fit      lwr      upr
## 1 10.81259 8.318631 13.30654
## 2 10.44674 7.966822 12.92665
##
## $se.fit
##         1         2
## 0.2263372 0.1828970
##
## $df
## [1] 109
##
## $residual.scale
## [1] 1.2378
```

```r
senic_data[c(47,112),]
```

```
##     Length.of.stay  Age Infection.risk Routine.culturing.ratio
## 47           19.56 59.9            6.5                    17.2
## 112          17.94 56.2            5.9                    26.4
##     Routine.chest.X.ray.ratio Number.of.beds Medical.school.affiliation
## 47                      113.7            306                           2
## 112                      91.8            835                           1
##     Region Average.daily.census Number.of.nurses
## 47       1                  273              172
## 112      1                  791              407
##     Available.facilities.and.services
## 47                               51.4
## 112                              62.9
```

Interpretation:

The observation Y47 and Y112 fall outside the individual confidence intervals obtained above. This means that if our sample were to be without Y47 and Y112, the actual values of these observations would not fall within out-of-sample C.I. given by the estimated regresion function. Thus, these observations are deemed to be outliers.