STAC67A2

Yulun Wu

24/02/2022

Q1

(a)

```
set.seed(1004912785)
Vote = read.table("vote-1.txt", header=T)
X = Vote$growth
error = rnorm(16, 0, sd=3.9)
Y  = 46.3 +4*X +error
df = data.frame(Y,X)
fit = lm(Y~X, data=df)
beta = fit$coefficients
beta
```

```
## (Intercept)           X
##   47.516039    3.731923
```

```
Y0 = 47.516039 + 0.1*3.731923
Y0
```

```
## [1] 47.88923
```

```
predict(fit,data.frame(X = 0.1))
```

```
##        1
## 47.88923
```

```
Sxx = sum((X-mean(X))^2)
Y0 + c(-1,1)*qnorm(0.975)*sqrt(1/16 + (0.1-mean(X))^2/Sxx)*3.9
```

```
## [1] 44.70896 51.06950
```

```
predict(fit, newdata = data.frame(X = 0.1), interval = "confidence")
```

```
##        fit      lwr      upr
## 1 47.88923 43.93035 51.84812
```

$\hat{\beta}_0 = 47.516039$ $\hat{\beta}_1 = 3.731923$ E(Y|X_0=0.1) is 47.88923 by hand and 47.88923 by R built-in function. The 95% CI for E(Y|X_0=0.1) is (44.70896, 51.06950) by hand and (43.93035, 51.84812) by R built-in function.

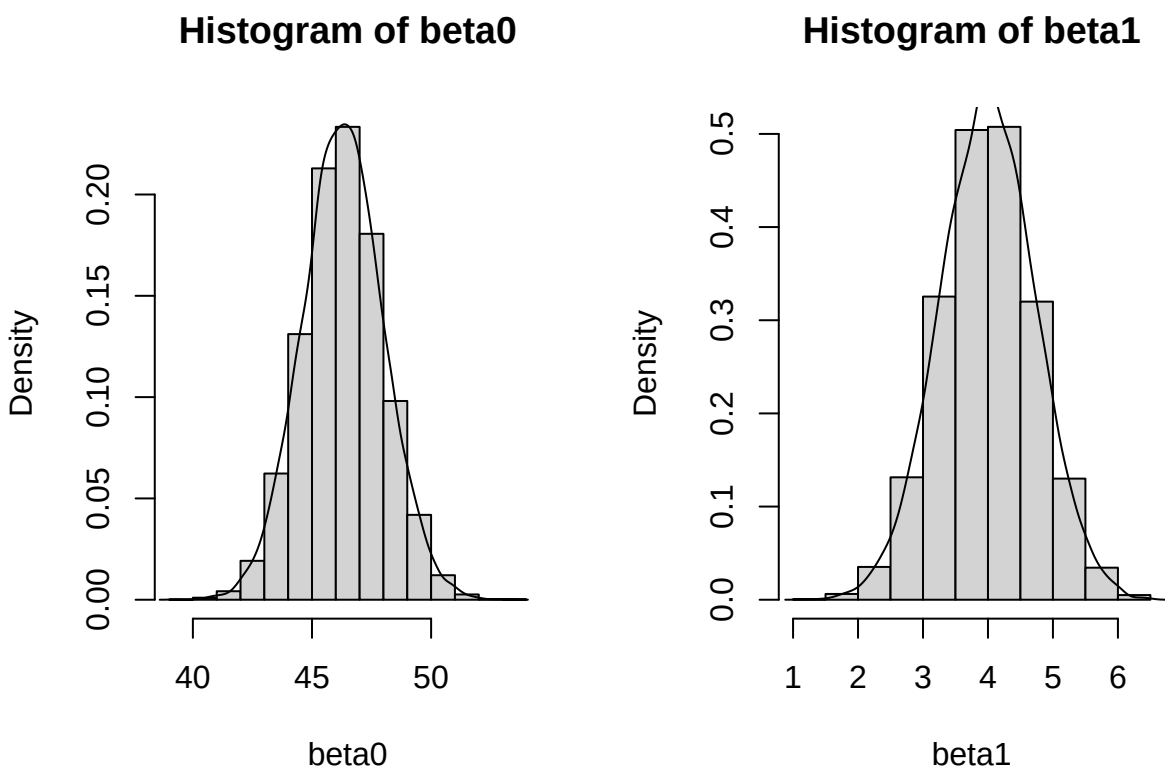
(b)

```
beta0 = rep(0, 10000)
beta1 = rep(0, 10000)
```

```r
set.seed(1004912785)

for(i in 1:10000){
error = rnorm(16, 0, sd=3.9)
Y  = 46.3 +4*X +error
df = data.frame(Y,X)
fit = lm(Y~X, data=df)
fit = lm(Y~X)
beta = fit$coefficients
beta0[i] = beta[1]
beta1[i] = beta[2]
}

par(mfrow=c(1, 2))
hist(beta0,freq = FALSE) # overlay hist with line
lines(density(beta0))
hist(beta1,freq = FALSE)
lines(density(beta1))
```




```r
result = c(mean(beta0),sd(beta0), mean(beta1), sd(beta1))
names(result) = c("Mean of Beta0", "SD of Beta0", "Mean of Beta1", "SD of Beta1")
result
```

```
## Mean of Beta0   SD of Beta0 Mean of Beta1   SD of Beta1
##    46.2985230     1.6945791     3.9970409     0.7285698
```

```r
Sxx = sum((X-mean(X))^2)
sd.beta0 = sqrt((1/16 + mean(X)^2/Sxx)*3.9^2)
sd.beta1 = sqrt(3.9^2/Sxx)
True.value = c(46.3, sd.beta0, 4, sd.beta1)
```

```
data.frame(True.value, result)
```

```
##                True.value     result
## Mean of Beta0  46.300000 46.2985230
## SD of Beta0     1.680853  1.6945791
## Mean of Beta1   4.000000  3.9970409
## SD of Beta1     0.721568  0.7285698
```

The estimated value is really closed to the theoretical value for both β_0 and β_1.

##(c)

```
#Q1(c)
set.seed(1004912785)
count = 0
for(i in 1:10000){

Vote = read.table("vote-1.txt", header=T)
X = Vote$growth
error = rnorm(16, 0, sd=3.9)
Y  = 46.3 +4*X +error
df = data.frame(Y,X)
fit = lm(Y~X, data=df)

beta = fit$coefficients
Y0 = 47.516039 + 0.1*3.731923

Sxx = sum((X-mean(X))^2)

N = predict(fit, newdata = data.frame(X = 0.1), interval = "confidence")

L = N[2]
U = N[3]
if (L<=47.88923 && 47.88923 <= U )
{
  count = count+1
}


}
print('this is the proportion ')
```

```
## [1] "this is the proportion "
```

```
print(count/10000)
```

```
## [1] 0.8912
```

Q2

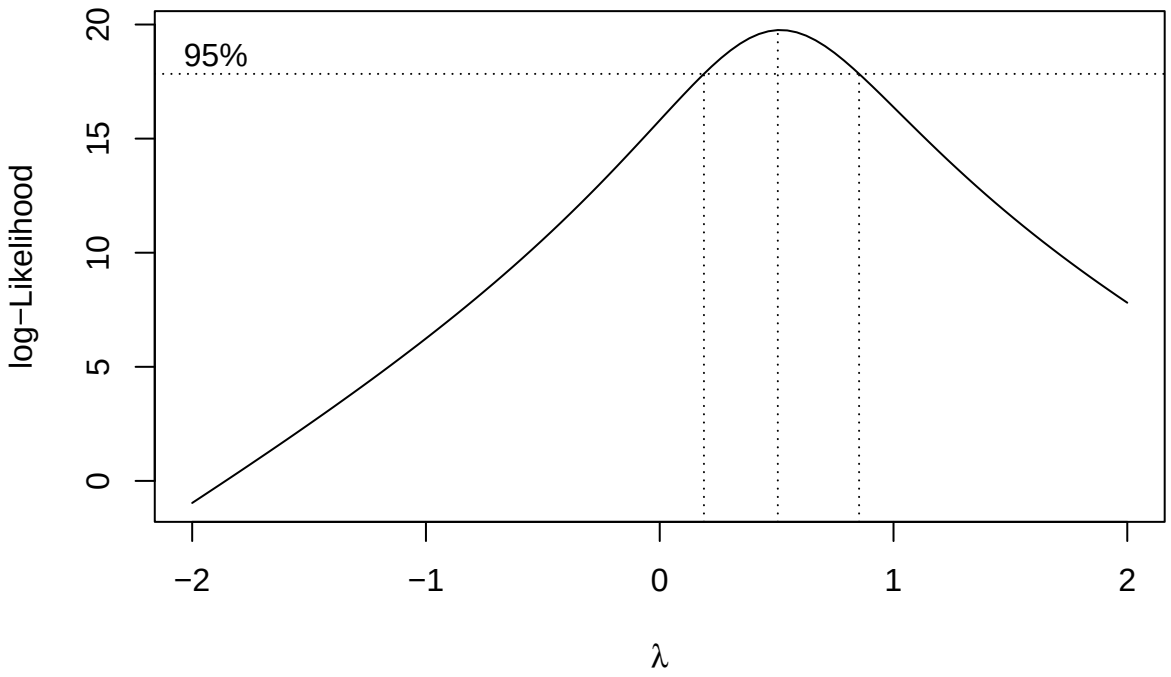
```
x <- c(0:9)
y <- c(98, 135, 162, 178, 221, 232, 283, 300, 374, 395)
k2 = prod(y)^(1/10)
box_cox = function(lambda) {
  if (lambda == 0) {
```

```
    W = k2*log(y)
  }
  else {
    k1 = 1/(lambda*k2^(lambda-1))
    W = k1*((y^lambda)-1)
  }
  fit = lm(W~x)
  SSE = sum((fit$residuals)^2)
  SSE
  return(SSE)
}

lambda = optimize(box_cox,interval=c(-2,2))
lambda
```

```
## $minimum
## [1] 0.5159801
##
## $objective
## [1] 915.4186
```

```
library(MASS)
fit = lm(y~x)
result = boxcox(fit)
```




```
bilambda = result$x[which.max(result$y)]
bilambda
```

```
## [1] 0.5050505
```

The λ given by R built-in function is 0.5050505, and the λ computed by function writen by me is 0.5159801.

Q3

(a)

```
kidiq = read.csv("kidiq.csv", header=T)
fit = lm(kid.score~mom.iq,data=kidiq)
summary(fit)
```

```
##
## Call:
## lm(formula = kid.score ~ mom.iq, data = kidiq)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.753 -12.074   2.217  11.710  47.691
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 25.79978    5.91741    4.36 1.63e-05 ***
## mom.iq       0.60997    0.05852   10.42  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.27 on 432 degrees of freedom
## Multiple R-squared:  0.201,  Adjusted R-squared:  0.1991
## F-statistic: 108.6 on 1 and 432 DF,  p-value: < 2.2e-16
```

```
n = dim(kidiq)[1]
beta = as.numeric(fit$coefficients)
Y0 = beta[1] + beta[2]*110
Sxx = sum((kidiq$mom.iq-mean(kidiq$mom.iq))^2)
SEY0 = sqrt(anova(fit)$`Mean Sq`[2]*(1/n + (110-mean(kidiq$mom.iq))^2/Sxx))
Y0 + c(-1, 1)*qt(0.975, n-2)*SEY0
```

```
## [1] 90.82506 94.96890
```

```
predict(fit, newdata=data.frame(mom.iq=110), interval="confidence", level=0.95)
```

```
##        fit      lwr     upr
## 1 92.89698 90.82506 94.9689
```

The 95% CI for or the mean test scores of all kids with their mother's IQ score = 110 is (90.82506, 94.96890) by hand and (90.82506, 94.9689) by R built-in function. ## (b)

```
Spred = sqrt(anova(fit)$`Mean Sq`[2]*(1+ 1/n + (110-mean(kidiq$mom.iq))^2/Sxx))
Y0 + c(-1, 1)*qt(0.995, n-2)*Spred
```

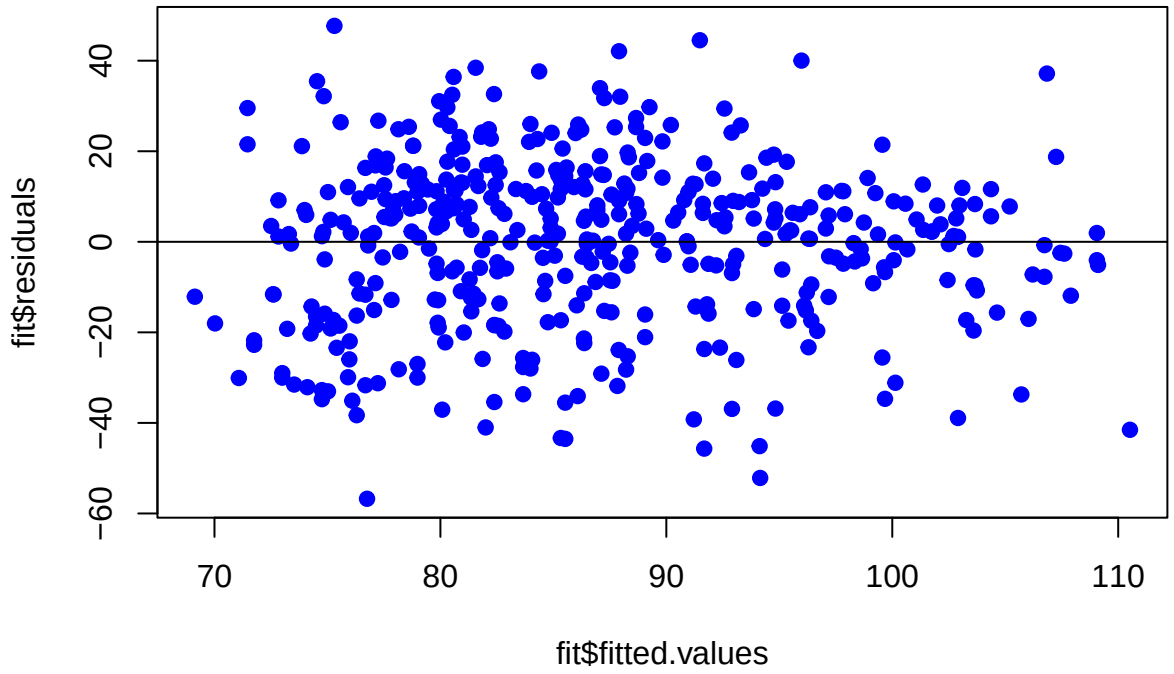
```
## [1]  45.55918 140.23478
```

```
predict(fit, newdata=data.frame(mom.iq=110), interval="predict", level=0.99)
```

```
##        fit      lwr      upr
## 1 92.89698 45.55918 140.2348
```

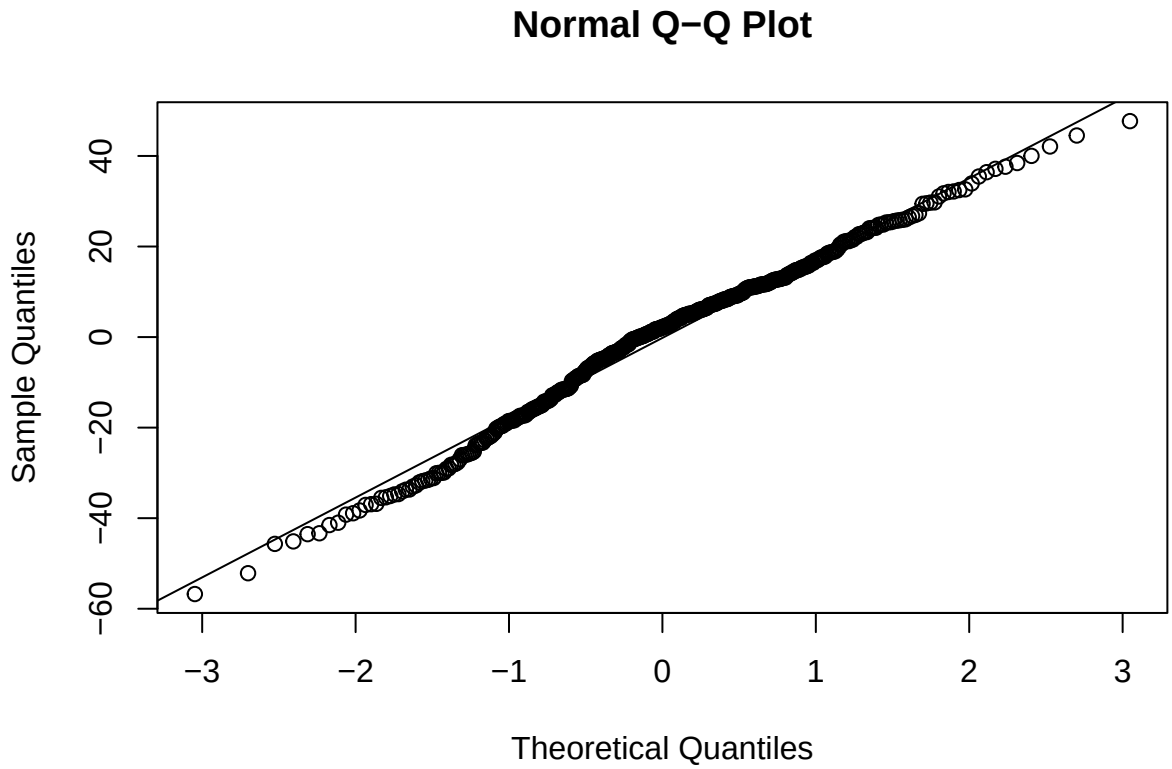
The 99% PI for new kid's test score when his or her mother's IQ score = 110 is (45.55918, 140.23478) by hand and (45.55918, 140.2348) by R built-in function. ## (c)

```
plot(fit$fitted.values, fit$residuals, pch=20, cex=1.5, col="blue")
abline(c(0,0))
```



No assumption violated, residual vs fitted value graph looks like random scatter. ## (d)

```
qqnorm(fit$residuals)
qqline(fit$residuals)
```



```
shapiro.test(fit$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  fit$residuals
## W = 0.98885, p-value = 0.00217
```

The normal Q-Q plot line is close to y=x line, but p-value=0.00217 $< \alpha = 0.05$, so we reject the null hypotheses that that sample is from normal population. Sample is not from normal population. ## (e)

```
library(lmtest)
```

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

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

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

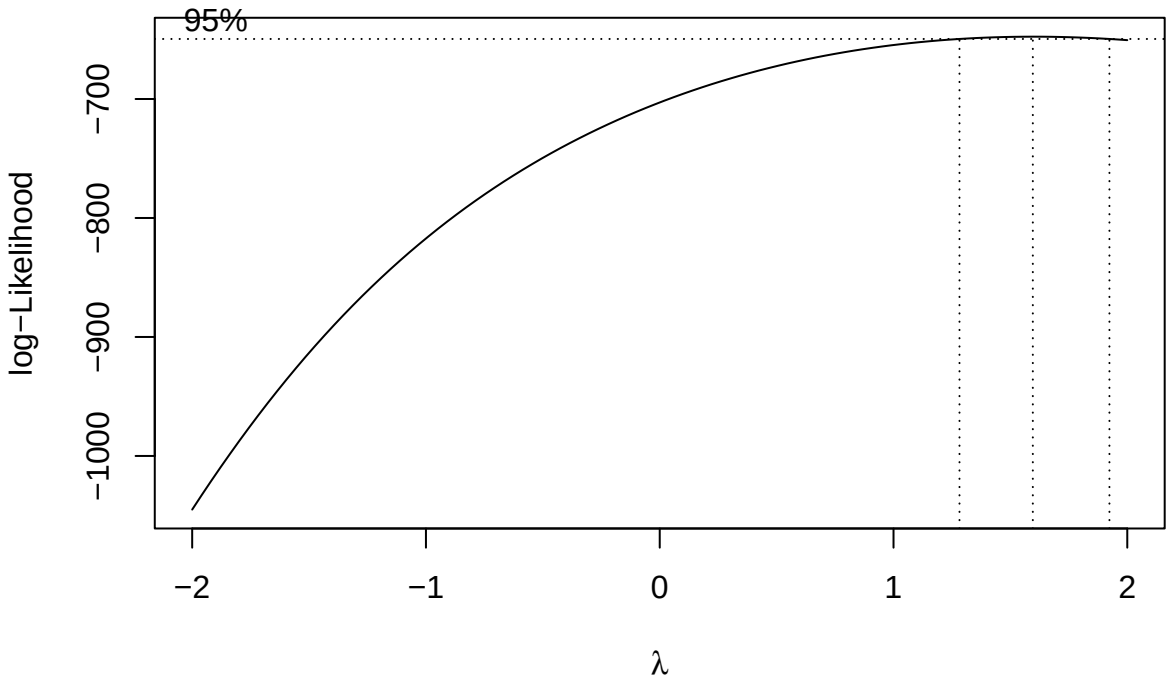
```
bptest(fit)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit
## BP = 6.0401, df = 1, p-value = 0.01398
```

Since p-value is 0.01398, p-value $< \alpha = 0.05$, so we reject the null hypotheses that the residuals are distributed with equal variance. The residuals are not distributed with equal variance

(f)

```
result = boxcox(fit)
```

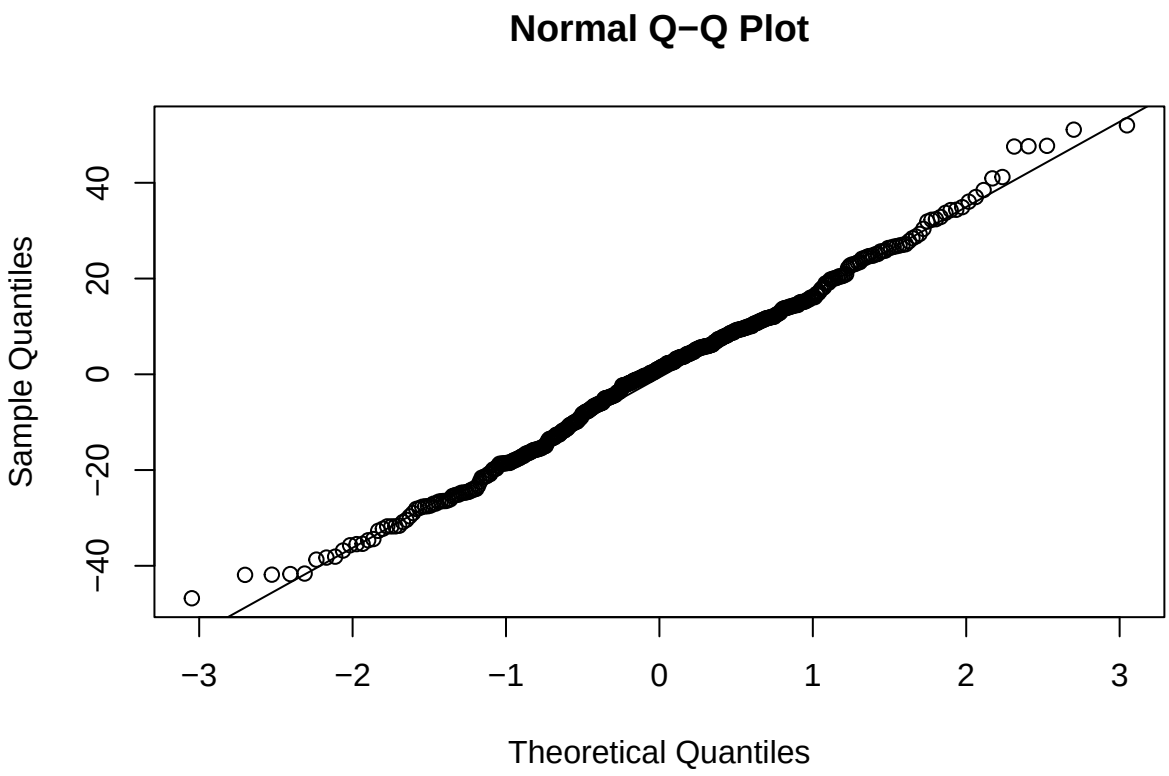



```
bilambda = result$x[which.max(result$y)]
bilambda
```

```
## [1] 1.59596
```

```
k2 = prod(kidiq$kid.score^(1/n))
k1 = 1/(bilambda*k2^(bilambda-1))
Y_star=k1*((kidiq$kid.score^bilambda)-1)
fit2 = lm(Y_star~mom.iq,data=kidiq)

qqnorm(fit2$residuals)
qqline(fit2$residuals)
```




```
shapiro.test(fit2$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  fit2$residuals
## W = 0.99484, p-value = 0.1556
```

```
bptest(fit2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit2
## BP = 0.75176, df = 1, p-value = 0.3859
```

Since p-value from Shapiro-Wilk normality test is 0.1556$>\alpha = 0.05$, so we fail to reject the null hypotheses that the sample is from normal population. Since p-value for Breusch-Pagan test is 0.3859, p-value=0.3859$>\alpha = 0.05$, so we fail to reject the null hypotheses that the residuals are distributed with equal variance.

Q4

(a)

```
x = c(0, 1, 2, 4, 6, 15)
Y = matrix(c(2.3, 4.2, 5.2, 5.9, 6.3, 7.2),nrow = 6)
X = matrix(rep(1,6),nrow = 6)
X = cbind(X,x)
X
```

```
##         x
```

```
## [1,] 1  0
## [2,] 1  1
## [3,] 1  2
## [4,] 1  4
## [5,] 1  6
## [6,] 1 15
```

```
Y
```

```
##      [,1]
## [1,]  2.3
## [2,]  4.2
## [3,]  5.2
## [4,]  5.9
## [5,]  6.3
## [6,]  7.2
```

##(b)

```
Beta_hat = solve(t(X)%*%X)%*%t(X)%*%Y
Beta_hat
```

```
##        [,1]
##   3.9848018
## x 0.2568282
```

```
Y_hat = X%*%Beta_hat
Y_hat
```

```
##          [,1]
## [1,] 3.984802
## [2,] 4.241630
## [3,] 4.498458
## [4,] 5.012115
## [5,] 5.525771
## [6,] 7.837225
```

```
residual = Y-Y_hat
residual
```

```
##             [,1]
## [1,] -1.68480176
## [2,] -0.04162996
## [3,]  0.70154185
## [4,]  0.88788546
## [5,]  0.77422907
## [6,] -0.63722467
```

##(c)

```
Var_hat_beta_hat = (sum(residual^2)/4)*solve(t(X)%*%X)
Var_hat_beta_hat
```

```
##                          x
##    0.39802045 -0.03951976
## x -0.03951976  0.00846852
```

##(d)

```
H = X%*%solve(t(X)%*%X)%*%t(X)
H
```

```
##            [,1]        [,2]       [,3]      [,4]      [,5]        [,6]
## [1,]  0.3105727  0.27973568  0.2488987 0.1872247 0.1255507 -0.15198238
## [2,]  0.2797357  0.25550661  0.2312775 0.1828194 0.1343612 -0.08370044
## [3,]  0.2488987  0.23127753  0.2136564 0.1784141 0.1431718 -0.01541850
## [4,]  0.1872247  0.18281938  0.1784141 0.1696035 0.1607930  0.12114537
## [5,]  0.1255507  0.13436123  0.1431718 0.1607930 0.1784141  0.25770925
## [6,] -0.1519824 -0.08370044 -0.0154185 0.1211454 0.2577093  0.87224670
```

```
library(psych)
tr(H)
```

```
## [1] 2
```

$\sum_{i=1}^{6} h_{ii} = 2$ which is the size of $\hat{\beta}$ vector.

##(e)

```
Var_hat_residual = (sum(residual^2)/4)*(diag(6)-H)
Var_hat_residual
```

```
##            [,1]       [,2]        [,3]       [,4]       [,5]        [,6]
## [1,]  0.8835489 -0.3585007 -0.31898093 -0.2399414 -0.1609019  0.19477596
## [2,] -0.3585007  0.9541199 -0.29639821 -0.2342957 -0.1721932  0.10726792
## [3,] -0.3189809 -0.2963982  1.00775390 -0.2286500 -0.1834846  0.01975988
## [4,] -0.2399414 -0.2342957 -0.22865004  1.0642107 -0.2060673 -0.15525620
## [5,] -0.1609019 -0.1721932 -0.18348460 -0.2060673  1.0529193 -0.33027229
## [6,]  0.1947760  0.1072679  0.01975988 -0.1552562 -0.3302723  0.16372472
```

‘‘‘