+20
90

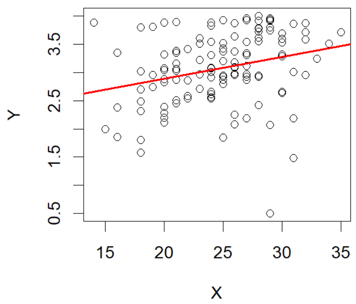
1.
(a)

```
> setwd("D:\\user\\Downloads\\data")
> GPA = read.table("Grade_Point_Average.csv", header
= T, sep = ",")
> colnames(GPA) = c("Y", "X")
> GPA.lm = lm(Y ~ X, data = GPA)
> GPA.lm

Call:
lm(formula = Y ~ X, data = GPA)

Coefficients:
(Intercept)            X
   2.11405      0.03883
```

The least squares estimates of beta0 is 2.11405 and beta1 is 0.03883. The estimated regression function is Y hat= 2.11405 + 0.03883 X.

(b)

```
> attach(GPA)
> plot(X, Y)
```



> abline(GPA.lm, col = "red", lwd = 2)

See the picture, the estimated regression function appear to fit the data well.

(c)

```
> predict(GPA.lm, data.frame(X = 30))
       1
3.278863
```

The point estimate of the mean freshman GPA for students with ACT test score $X = 30$ is 3.278863.

2.

(a)

+20

```
> summary.aov(GPA.lm)
             Df Sum Sq Mean Sq F value  Pr(>F)
X             1   3.59   3.588    9.24 0.00292 **
Residuals   118  45.82   0.388
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(b)

1.H0: beta1 = 0 v.s. H1: beta1 = 0 is not true

2.alph = 0.01

3.Test statistic: F

4.Decision rule: p-value<0.01, then we reject H0.
5. Decision:
From (a) the p-value is 0.00292 < 0.01. We reject H0.
6. Conclusion: Y and X has linear relationship.
(c)

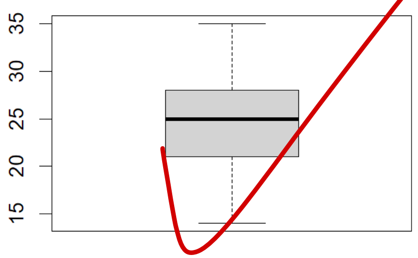
```
> summary(GPA.lm)

Call:
lm(formula = Y ~ X, data = GPA)

Residuals:
    Min       1Q   Median       3Q      Max
-2.74004 -0.33827  0.04062  0.44064  1.22737

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.11405    0.32089   6.588  1.3e-09 ***
X            0.03883    0.01277   3.040  0.00292 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6231 on 118 degrees of freedom
Multiple R-squared:  0.07262,	Adjusted R-squared:  0.06476
F-statistic:  9.24 on 1 and 118 DF,  p-value: 0.002917
```

The absolute magnitude of the reduction in the variation of Y when X is introduced into the regression model is 3.588 which is obtained by SSR.

+20

3.
(a)

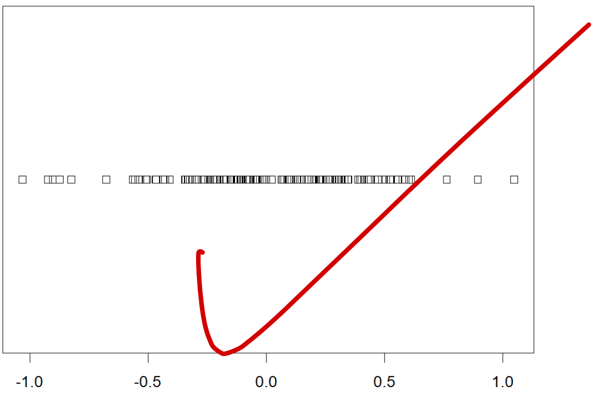
```
> setwd("D:\\user\\Downloads\\data")
> GPA_X = read.table("Grade_Point_Average_X.csv",
sep = ",", header = T)
> colnames(GPA_X) <- c("Y", "X", "X2", "X3")
```




```
> boxplot(GPA_X$X)
```

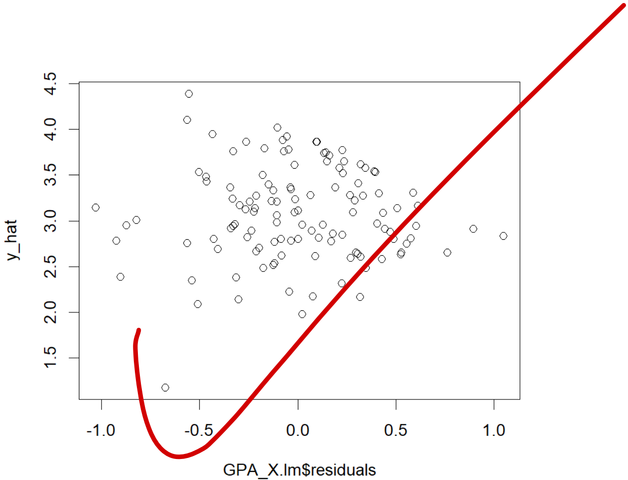
(b)

```
> GPA_X.lm = lm(Y ~ X + X2 + X3, data = GPA_X)
> stripchart(GPA_X.lm$residuals)
```



(c)

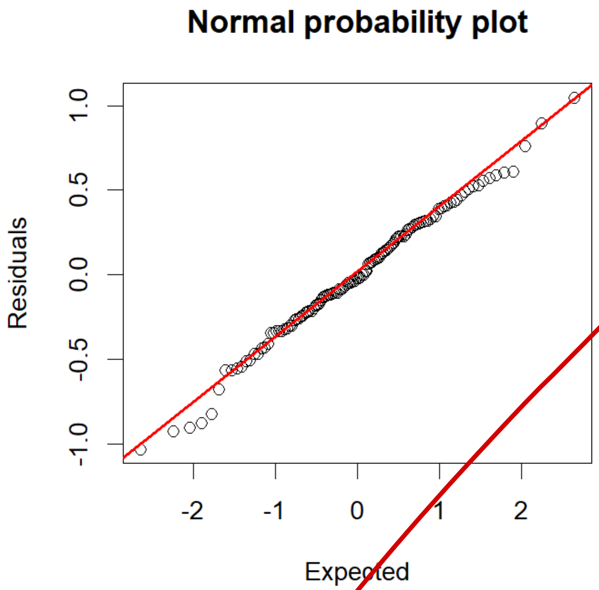
```
> b0 = GPA_X.lm$coefficients[1]
> b1 = GPA_X.lm$coefficients[2]
> b2 = GPA_X.lm$coefficients[3]
> b3 = GPA_X.lm$coefficients[4]
> y_hat = b0 + b1*GPA_X$X + b2*GPA_X$X2 +
b3*GPA_X$X3
```




```
>
plot(GPA_X.lm$residuals, y_hat)
```

(d)

```
> qqnorm(GPA_X.lm$residuals, xlab = "Expected", ylab
= "Residuals", pch = 16, main = "(d) Normal Probability
Plot")
> qqline(GPA_X.lm$residuals, col = 'red', lwd = 2)
```




```
>
> expected = qqnorm(GPA_X.lm$residuals, xlab =
"Expected",ylab = "Residuals", main = "Normal
probability plot")
> cor(expected$x, expected$y)
[1] 0.9962909
```

(e)

(f)

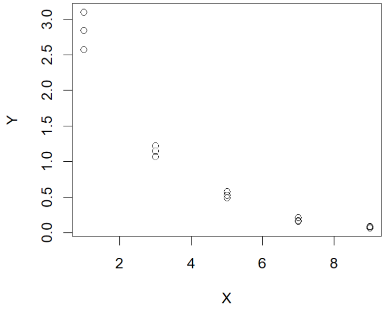
4.

(a)

+30

```
> setwd("D:\\user\\Downloads\\data")
> concentration =
read.table("Solution_concentration.csv", sep = ",",
header = T)
```

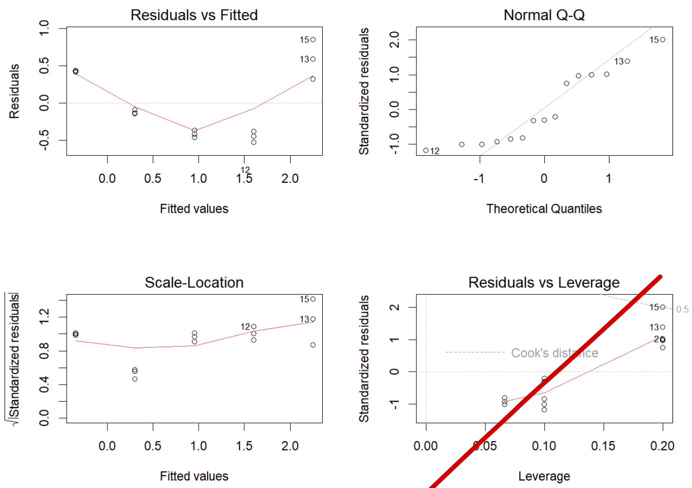
```
> colnames(concentration) = c("Y", "X")
> attach(concentration)
```




```
> plot(X, Y)
> concentration.lm = lm(Y ~ X, data = concentration)
> summary(concentration.lm)

Call:
lm(formula = Y ~ X, data = concentration)

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

> par(mfrow = c(2, 2))



>

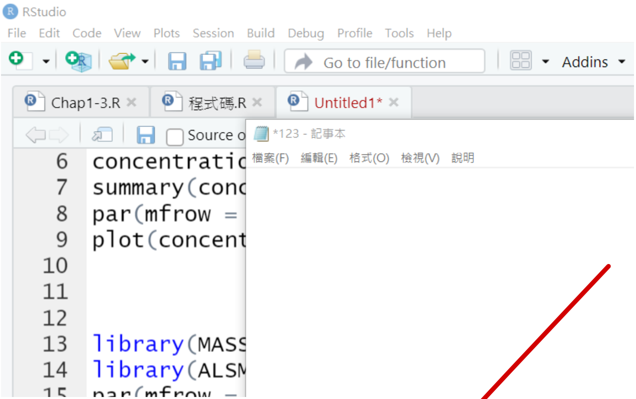
plot(concentration.lm)

From the output, we can see that unequal error variances and nonnormality of the error terms. To remedy these departures from the simple linear regression model, we need a transformation on Y, since the shapes and spreads of the distributions of Y need to be changed.

(b)

```
library(MASS)
library(ALSM)
par(mfrow = c(1, 1))
```


```
boxcox(concentration.lm,lambda = seq(-2, 2, by = 0.1))
boxcox.sse(X, Y, l = seq(-2, 2, by = 1))
```



	lambda	SSE
1	-2	68.84280491
2	-1	3.16846767
5	0	0.03897303
3	1	2.92465333
4	2	64.15599383

From the output, we take Y' = logY.

(c)

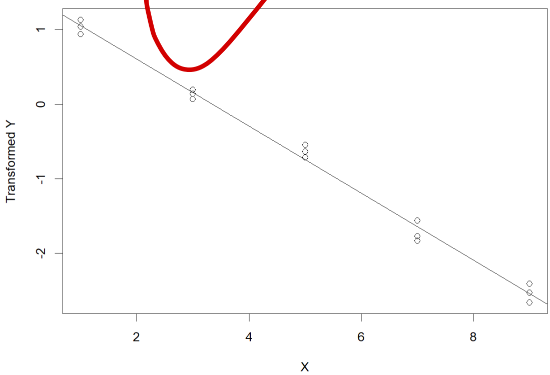
```
> concentration.log.lm = lm(log(Y) ~ X, data =
concentration)
> summary(concentration.log.lm)

Call:
lm(formula = log(Y) ~ X, data = concentration)

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 least squares estimates of beta0 is 1.50792 and beta 1 is -0.44993.
The estimated regression function is   new Y hat =1.50792 - 0.44993X.

(d)

```
> par(mfrow = c(1, 1))
> plot(X, log(Y), xlab = "X", ylab = "Transformed Y")
```



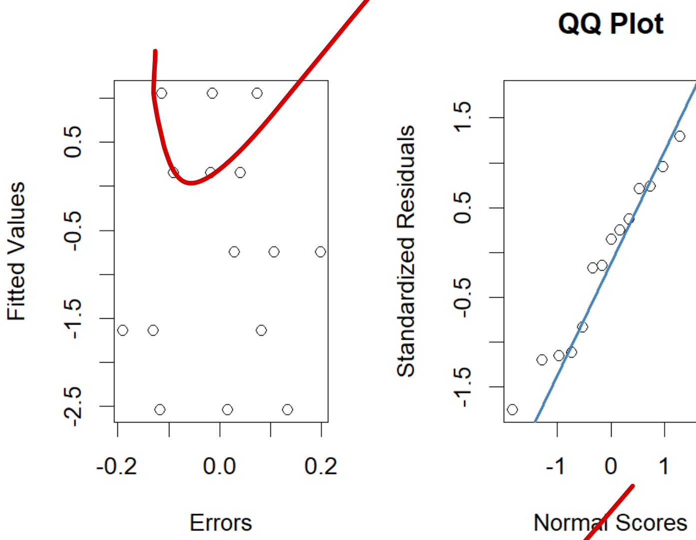

```
> abline(concentration.log.lm)
```

It was better than before.

(e)

```
> ei <- concentration.log.lm$residuals
> yhat <- concentration.log.lm$fitted.values
> par(mfrow = c(1, 2))
> plot(ei, yhat, xlab = "Errors", ylab = "Fitted Values")
> stdei <- rstandard(concentration.log.lm)
> qqnorm(stdei, ylab = "Standardized Residuals", xlab =
"Normal Scores", main = "QQ Plot")
> qqline(stdei, col = "steelblue", lwd = 2)
```

The error variances are constant. And errors are approximately normally distributed



.

(f)

Log(Y hat) = 1.50792 – 0.44993*X.

Then Y hat = exp(1.50792 – 0.44993*X).