Simple Linear Regression

Problem statement 1

Calories_consumed-> predict weight gained using calories consumed

Answer:

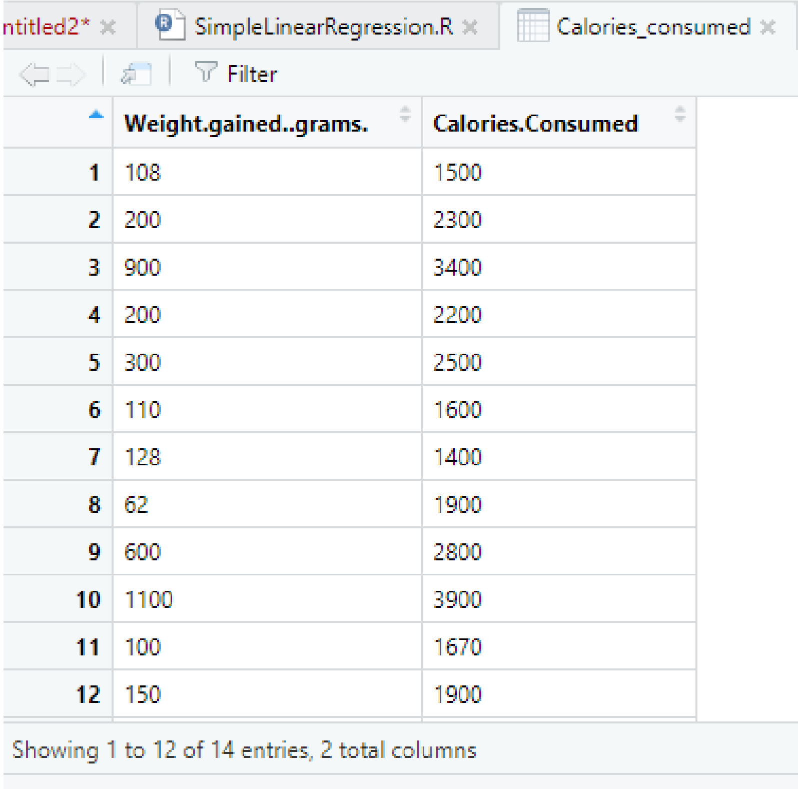
Rcode:

```
## 1) Calories_consumed-> predict weight gained using calories consumed ##
Calories_consumed <- read.csv(file.choose())
View(Calories_consumed)
attach(Calories_consumed)
plot(Calories_consumed)
summary(Calories_consumed)
cor(Weight.gained..grams.,Calories.Consumed)

reg <- lm(Weight.gained..grams.~Calories.Consumed,data = Calories_consumed)
summary(reg)
confint(reg,level = 0.95)
predict(reg,interval = "predict")

reg_log <- lm(Weight.gained..grams.~log(Calories.Consumed),data = Calories_consumed)
summary(reg_log)
confint(reg_log,level = 0.95)
predict(reg_log,interval = "predict")
reg_sqrt <- lm(Weight.gained..grams.~sqrt(Calories.Consumed,data) = Calories_consumed)
summary(reg_sqrt)
confint(reg_sqrt,level = 0.95)
```

```
predict(reg_sqrt,interval = "predict")

reg_exp <- lm(log(Weight.gained..grams.)~Calories.Consumed,data = Calories_consumed)

summary(reg_exp)

confint(reg_exp,level = 0.95)

predict(reg_exp,interval = "predict")


a <- sqrt(log(Calories.Consumed))

reg_1 <- lm(log(Weight.gained..grams.)~a,data = Calories_consumed)

summary(reg_1)

confint(reg_1,level = 0.95)

predict(reg_1,interval = "predict")
```

Console:

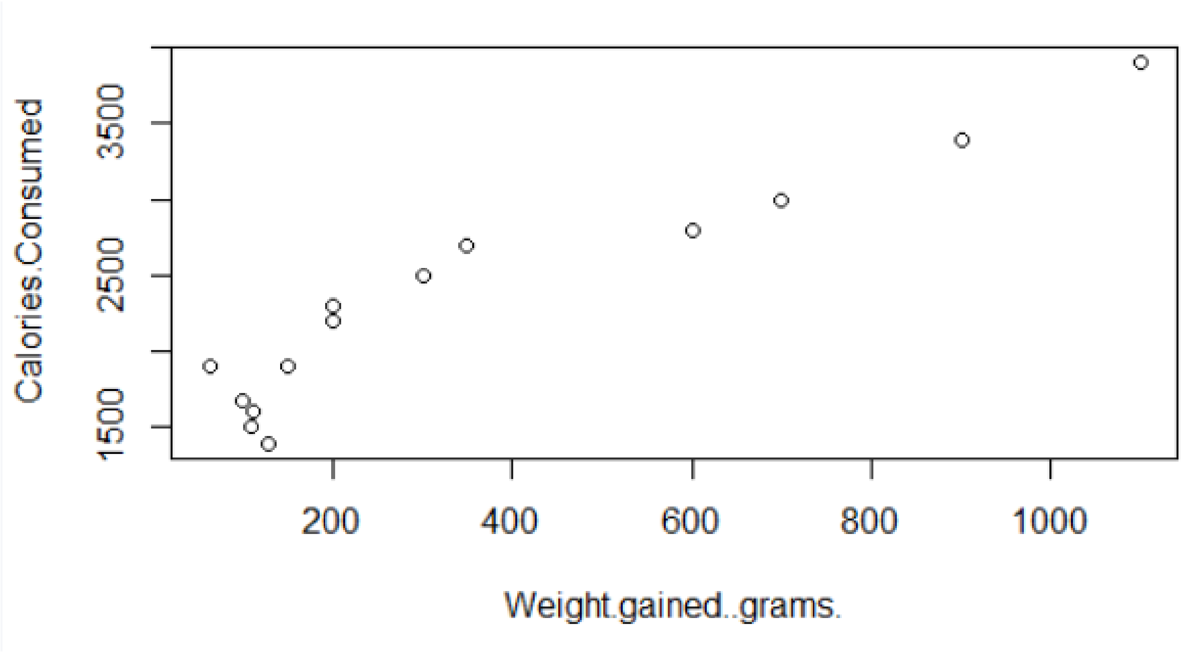
```
> Calories_consumed <- read.csv(file.choose())
> View(Calories_consumed)
```



	Weight.gained..grams.	Calories.Consumed
1	108	1500
2	200	2300
3	900	3400
4	200	2200
5	300	2500
6	110	1600
7	128	1400
8	62	1900
9	600	2800
10	1100	3900
11	100	1670
12	150	1900

Showing 1 to 12 of 14 entries, 2 total columns

```
> attach(Calories_consumed)
> plot(Calories_consumed)
```



```
> summary(Calories_consumed)
 Weight.gained..grams. Calories.Consumed
 Min.   :  62.0        Min.   :1400
 1st Qu.: 114.5        1st Qu.:1728
 Median : 200.0        Median :2250
 Mean   : 357.7        Mean   :2341
 3rd Qu.: 537.5        3rd Qu.:2775
 Max.   :1100.0        Max.   :3900
> cor(Weight.gained..grams.,Calories.Consumed)
[1] 0.946991
> reg <- lm(Weight.gained..grams.~Calories.Consumed,data = Calories_co
nsumed)
> summary(reg)

Call:
lm(formula = Weight.gained..grams. ~ Calories.Consumed, data = Calorie
s_consumed)

Residuals:
    Min      1Q  Median      3Q     Max 
-158.67 -107.56   36.70   81.68  165.53 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)       -625.75236  100.82293  -6.206 4.54e-05 ***
Calories.Consumed    0.42016    0.04115  10.211 2.86e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 111.6 on 12 degrees of freedom
Multiple R-squared:  0.8968,	Adjusted R-squared:  0.8882 
F-statistic: 104.3 on 1 and 12 DF,  p-value: 2.856e-07
> confint(reg,level = 0.95)
                       2.5 %       97.5 %
(Intercept)      -845.4266546 -406.0780569
Calories.Consumed    0.3305064    0.5098069
> predict(reg,interval = "predict")
          fit        lwr       upr
1     4.482599 -258.20569  267.1709
2   340.607908   88.93791  592.2779
3   802.780209  533.81393 1071.7465
4   298.592245   46.63271  550.5518
5   424.639236  172.59086  676.6876
6    46.498263 -213.75953  306.7561
7   -37.533065 -302.93258  227.8664
8   172.545254  -82.18110  427.2716
9   550.686227  295.69632  805.6761
10 1012.858527  724.99432 1300.7227
11   75.909227 -182.81852  334.6370
12  172.545254  -82.18110  427.2716
13  508.670563  254.97398  762.3671
14  634.717554  376.22600  893.2091

> reg_log <- lm(Weight.gained..grams.~log(Calories.Consumed),data = Ca
lories_consumed)
> summary(reg_log)

Call:
lm(formula = Weight.gained..grams. ~ log(Calories.Consumed), 
    data = Calories_consumed)

Residuals:
    Min      1Q  Median      3Q     Max 
-187.44 -142.96   23.13  113.20  213.82 

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)    
(Intercept)             -6955.7     1030.9  -6.747 2.05e-05 ***
log(Calories.Consumed)    948.4      133.6   7.100 1.25e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 152.3 on 12 degrees of freedom
Multiple R-squared:  0.8077,	Adjusted R-squared:  0.7917 
F-statistic:  50.4 on 1 and 12 DF,  p-value: 1.248e-05

> confint(reg_log,level = 0.95)
                           2.5 %    97.5 %
(Intercept)            -9201.8063 -4709.494
log(Calories.Consumed)   657.3251  1239.418
> predict(reg_log,interval = "predict")
        fit        lwr       upr
```

```
1  -19.99870 -382.5178898  342.5205
2  385.37711   41.7849717  728.9693
3  756.06367  391.4700627 1120.6573
4  343.22032   -0.2957275  686.7364
5  464.45388  119.4081720  809.4996
6   41.20781 -315.7491910  398.1648
7  -85.42959 -454.8597180  284.0005
8  204.18573 -142.5179686  550.8894
9  571.93160  222.2096884  921.6535
10 886.18133  506.3308457 1266.0318
11  81.81708 -271.9519877  435.5861
12 204.18573 -142.5179686  550.8894
13 537.44155  189.5540023  885.3291
14 637.36248  283.3161385  991.4088
> reg_sqrt <- lm(Weight.gained..grams.~sqrt(Calories.Consumed),data =
Calories_consumed)
> summary(reg_sqrt)

Call:
lm(formula = Weight.gained..grams. ~ sqrt(Calories.Consumed),
    data = Calories_consumed)

Residuals:
    Min      1Q  Median      3Q     Max
-175.37 -123.59   29.85  105.48  191.23

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             -1577.379    231.125  -6.825 1.84e-05 ***
sqrt(Calories.Consumed)    40.467      4.777   8.471 2.08e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 131.5 on 12 degrees of freedom
Multiple R-squared:  0.8567,      Adjusted R-squared:  0.8448
F-statistic: 71.76 on 1 and 12 DF,  p-value: 2.083e-06

> confint(reg_sqrt,level = 0.95)
                              2.5 %      97.5 %
(Intercept)             -2080.95760 -1073.80123
sqrt(Calories.Consumed)    30.05875    50.87597
> predict(reg_sqrt,interval = "predict")
         fit        lwr       upr
1  -10.08526 -321.30131  301.1308
2  363.36704   66.87411  659.8600
3  782.25294  466.29469 1098.2112
4  320.70807   24.06595  617.3502
5  445.98864  148.63117  743.3461
6   41.31503 -266.14037  348.7704
7  -63.22941 -378.86930  252.4105
8  186.55196 -113.18812  486.2920
9  563.95206  262.75467  865.1494
10 949.80647  616.49160 1283.1213
11  76.34493 -228.84928  381.5391
```

```
12 186.55196 -113.18812  486.2920
13 525.36635  225.75755  824.9751
14 639.10923  333.91346  944.3050


> reg_exp <- lm(log(Weight.gained..grams.)~Calories.Consumed,data = Ca
lories_consumed)
> summary(reg_exp)

Call:
lm(formula = log(Weight.gained..grams.) ~ Calories.Consumed, 
    data = Calories_consumed)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.86537 -0.10532  0.02462  0.13467  0.42632 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       2.8386724  0.2994581   9.479 6.36e-07 ***
Calories.Consumed 0.0011336  0.0001222   9.276 8.02e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3314 on 12 degrees of freedom
Multiple R-squared:  0.8776,	Adjusted R-squared:  0.8674 
F-statistic: 86.04 on 1 and 12 DF,  p-value: 8.018e-07

> confint(reg_exp,level = 0.95)
                         2.5 %      97.5 %
(Intercept)       2.1862091856 3.491135698
Calories.Consumed 0.0008673238 0.001399871
> predict(reg_exp,interval = "predict")
        fit      lwr      upr
1  4.539069 3.758848 5.319289
2  5.445947 4.698452 6.193442
3  6.692904 5.894036 7.491771
4  5.332587 4.584232 6.080942
5  5.672666 4.924047 6.421285
6  4.652428 3.879426 5.425430
7  4.425709 3.637435 5.213982
8  4.992508 4.235935 5.749080
9  6.012745 5.255390 6.770101
10 7.259702 6.404706 8.114699
11 4.731780 3.963323 5.500238
12 4.992508 4.235935 5.749080
13 5.899386 5.145871 6.652900
14 6.239465 5.471709 7.007221


> a <- sqrt(log(Calories.Consumed))
> reg_1 <- lm(log(Weight.gained..grams.)~a,data = Calories_consumed)
> summary(reg_1)
```

```
Call:
lm(formula = log(Weight.gained..grams.) ~ a, data = Calories_consumed)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.94248 -0.13474 -0.00203  0.23861  0.60715 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -35.291      5.085  -6.940 1.56e-05 ***
a             14.689      1.831   8.021 3.66e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3756 on 12 degrees of freedom
Multiple R-squared:  0.8428,	Adjusted R-squared:  0.8297 
F-statistic: 64.34 on 1 and 12 DF,  p-value: 3.658e-06

> confint(reg_1,level = 0.95)
                2.5 %    97.5 %
(Intercept) -46.37129 -24.21143
a            10.69929  18.67919
> predict(reg_1,interval = "predict")
        fit      lwr      upr
1  4.432707 3.538169 5.327245
2  5.577121 4.729814 6.424427
3  6.596245 5.697722 7.494767
4  5.459605 4.612568 6.306643
5  5.796646 4.945625 6.647668
6  4.607603 3.727196 5.488009
7  4.244885 3.332655 5.157114
8  5.069611 4.214881 5.924342
9  6.093150 5.230567 6.955732
10 6.948146 6.013366 7.882925
11 4.723220 3.850861 5.595580
12 5.069611 4.214881 5.924342
13 5.998233 5.140156 6.856309
14 6.272621 5.399499 7.145743
```

Conclusion- p-value is less than 0.05. and Multiple R- squared value is 0.8968. This means the regression model will predict the output 89.68% time correct

Problem statement 2

Delivery_time -> Predict delivery time using sorting time

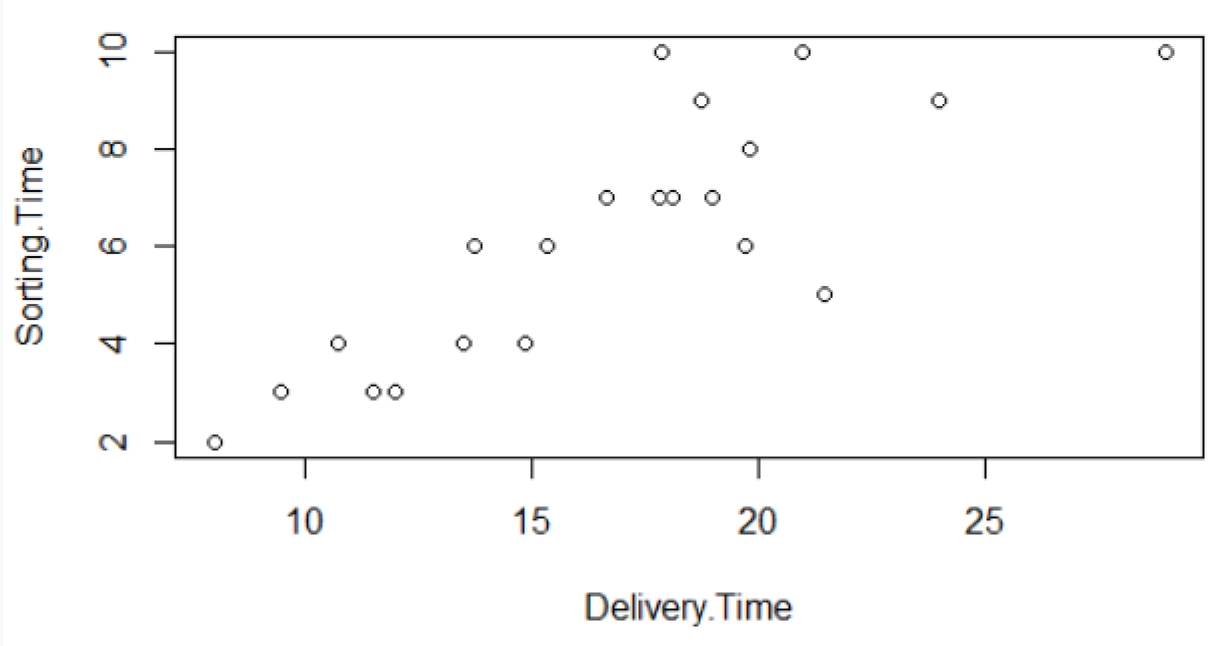
Answer:

Rcode:

```
##2) Delivery_time -> Predict delivery time using sorting time ##

Delivery_time <- read.csv(file.choose())
names(Delivery_time)
attach(Delivery_time)
summary(Delivery_time)
plot(Delivery.Time,Sorting.Time)
cor(Delivery.Time,Sorting.Time)
colnames(Delivery_time) <- c("DeliveryTime","SortingTime")

model1 <- lm(DeliveryTime~SortingTime,data = Delivery_time)
summary(model1)
confint(model1,level = 0.95)
predict(model1,interval = "predict")
plot(model1)

model2 <- lm(log(DeliveryTime)~log(SortingTime),data = Delivery_time)
summary(model2)
confint(model2,level = 0.95)
predict(model2,interval = "predict")
plot(model2)

influenceIndexPlot(model1)
model3 <- lm(DeliveryTime ~ SortingTime, data = Delivery_time[c(-5,-9,-21),])
summary(model3)
plot(model3)
```

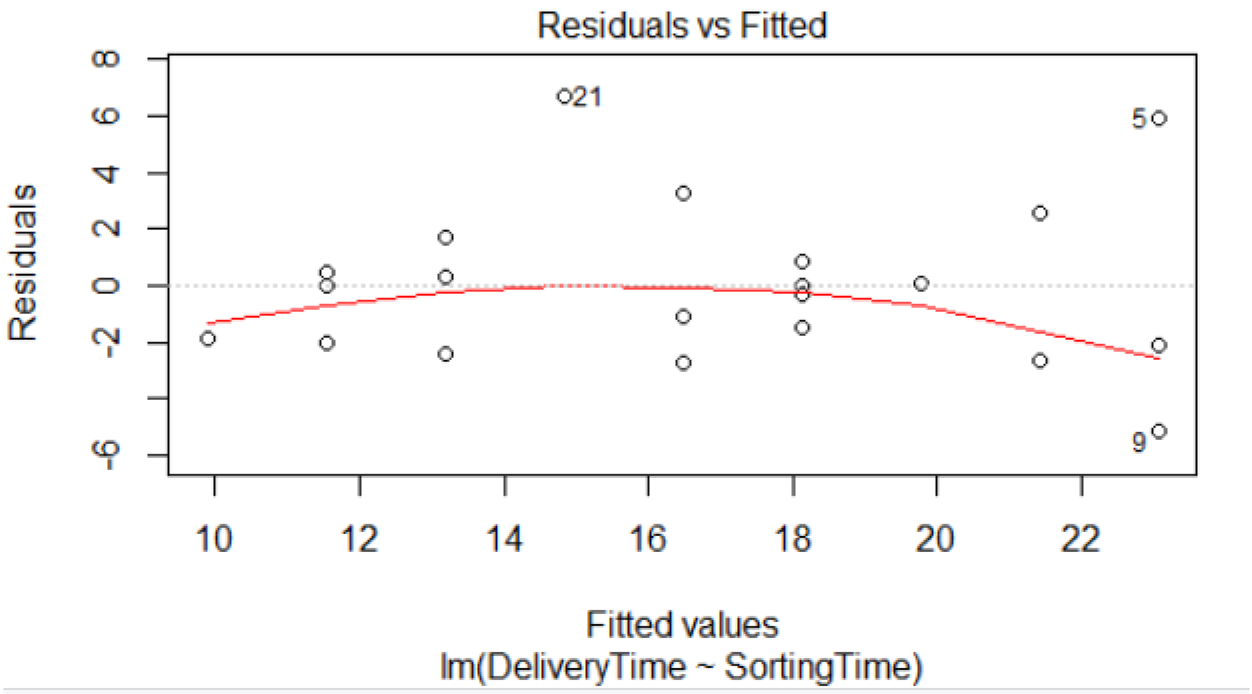
Console:

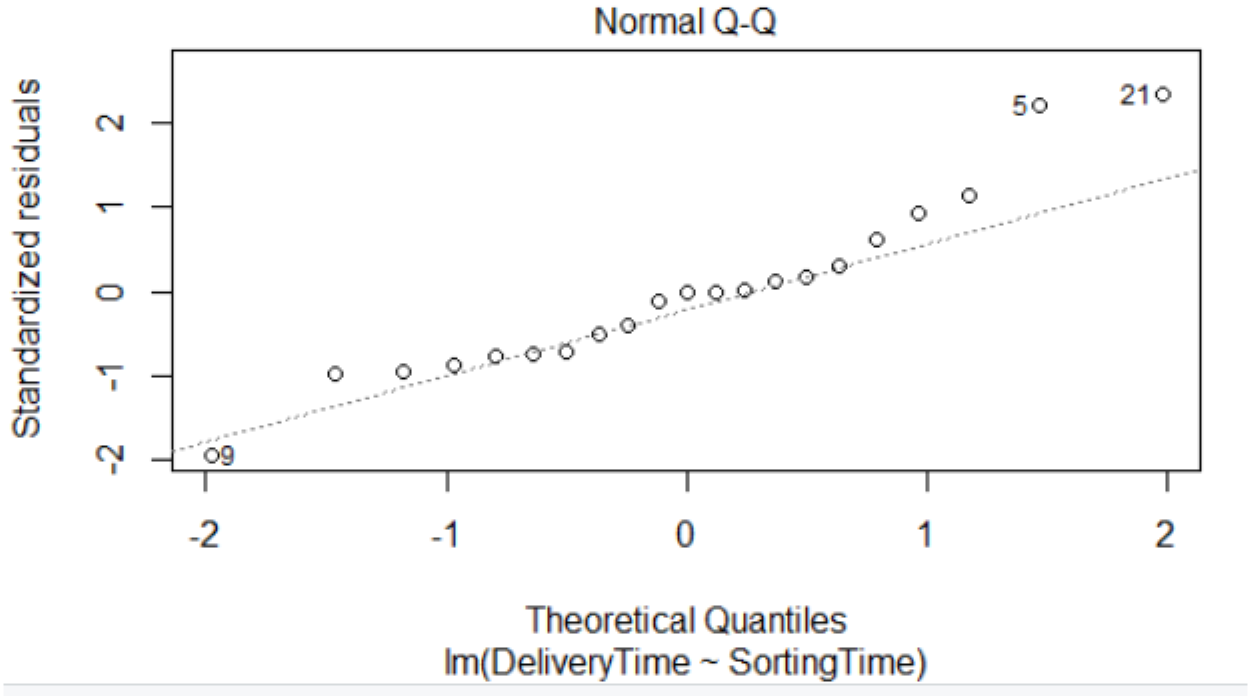
```
> Delivery_time <- read.csv(file.choose())
> names(Delivery_time)
[1] "Delivery.Time" "Sorting.Time"
> attach(Delivery_time)
> summary(Delivery_time)
 Delivery.Time    Sorting.Time  
 Min.   : 8.00   Min.   : 2.00  
 1st Qu.:13.50   1st Qu.: 4.00  
 Median :17.83   Median : 6.00  
 Mean   :16.79   Mean   : 6.19  
 3rd Qu.:19.75   3rd Qu.: 8.00  
 Max.   :29.00   Max.   :10.00  
> plot(Delivery.Time,Sorting.Time)
```

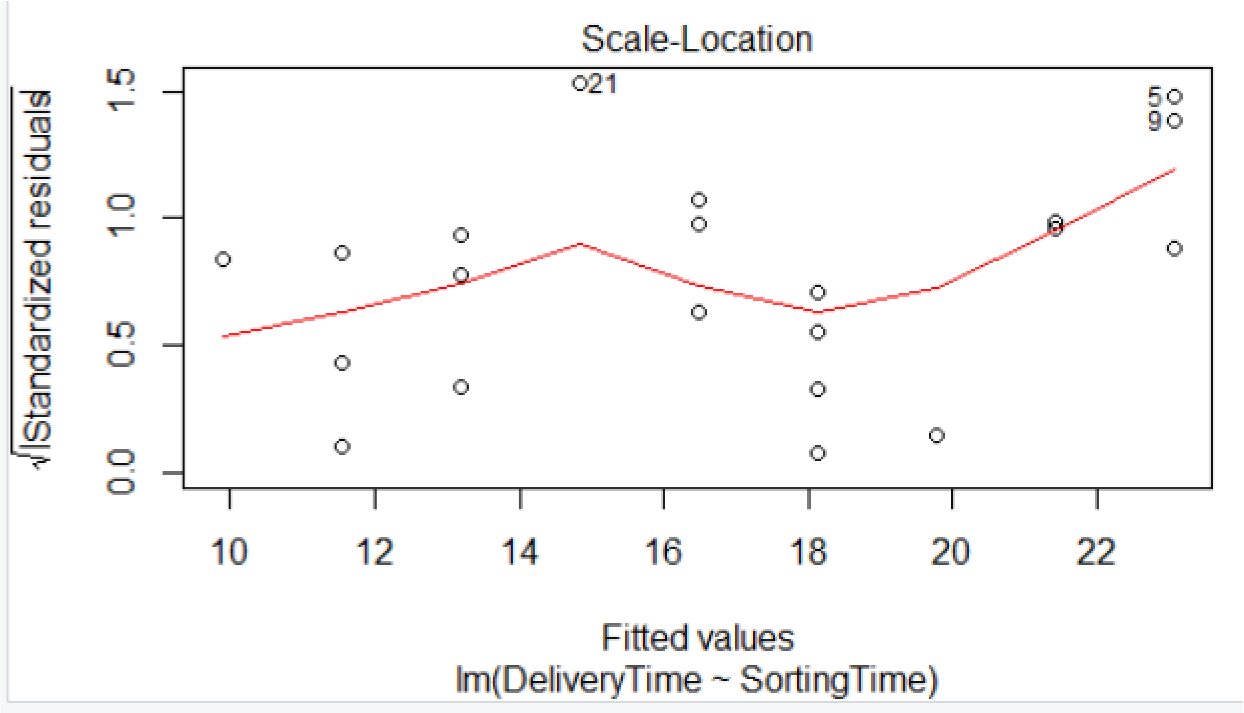


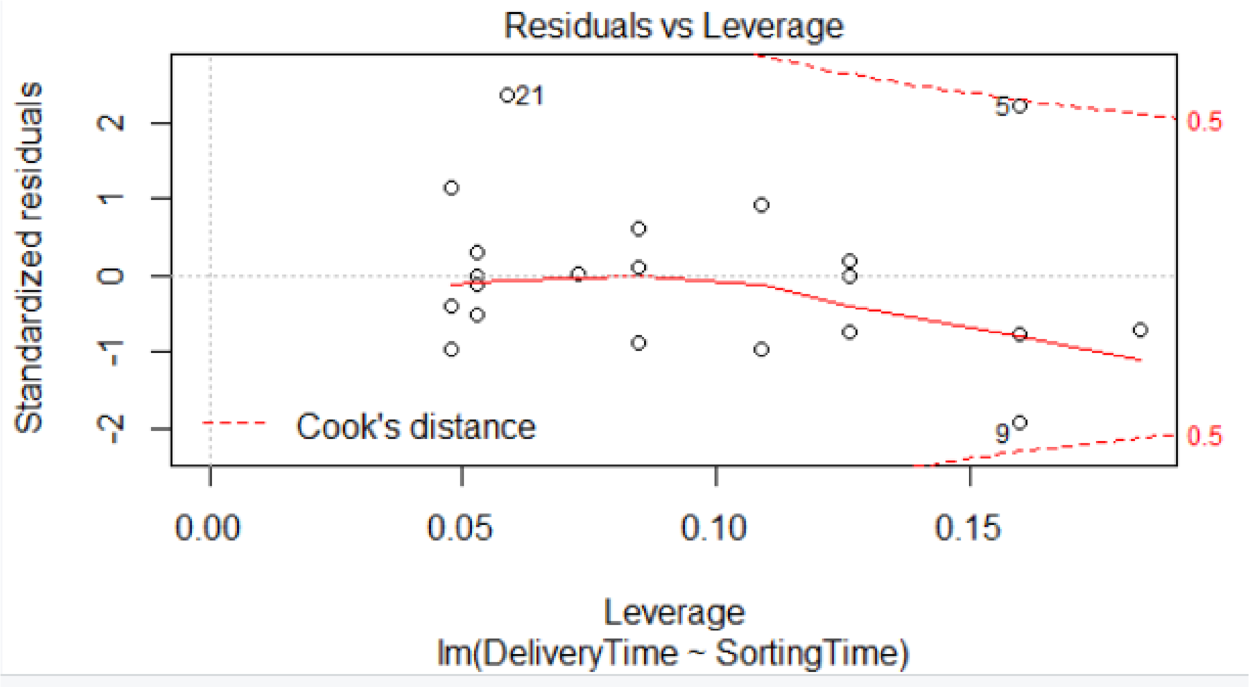
```
> cor(Delivery.Time,Sorting.Time)
[1] 0.8259973
> colnames(Delivery_time) <- c("DeliveryTime","SortingTime")
> model1 <- lm(DeliveryTime~SortingTime,data = Delivery_time)
> summary(model1)

Call:
lm(formula = DeliveryTime ~ SortingTime, data = Delivery_time)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.1729 -2.0298 -0.0298  0.8741  6.6722 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   6.5827     1.7217   3.823  0.00115 ** 
SortingTime   1.6490     0.2582   6.387 3.98e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 2.935 on 19 degrees of freedom
Multiple R-squared:  0.6823,	Adjusted R-squared:  0.6655 
F-statistic:  40.8 on 1 and 19 DF,  p-value: 3.983e-06

> confint(model1,level = 0.95)
               2.5 %    97.5 %
(Intercept) 2.979134 10.186334
SortingTime 1.108673  2.189367
> predict(model1,interval = "predict")
         fit       lwr      upr
1  23.072933 16.457161 29.68870
2  13.178814  6.780993 19.57663
3  16.476853 10.188630 22.76508
4  21.423913 14.955850 27.89198
5  23.072933 16.457161 29.68870
6  16.476853 10.188630 22.76508
7  18.125873 11.823294 24.42845
8  11.529794  5.010345 18.04924
9  23.072933 16.457161 29.68870
10 21.423913 14.955850 27.89198
11 19.774893 13.411938 26.13785
12 13.178814  6.780993 19.57663
13 18.125873 11.823294 24.42845
14 11.529794  5.010345 18.04924
15 11.529794  5.010345 18.04924
16 13.178814  6.780993 19.57663
17 16.476853 10.188630 22.76508
18 18.125873 11.823294 24.42845
19  9.880774  3.198090 16.56346
20 18.125873 11.823294 24.42845
21 14.827833  8.507631 21.14804

> plot(model1)
Hit <Return> to see next plot: 
Hit <Return> to see next plot: 
Hit <Return> to see next plot: 
Hit <Return> to see next plot:
```

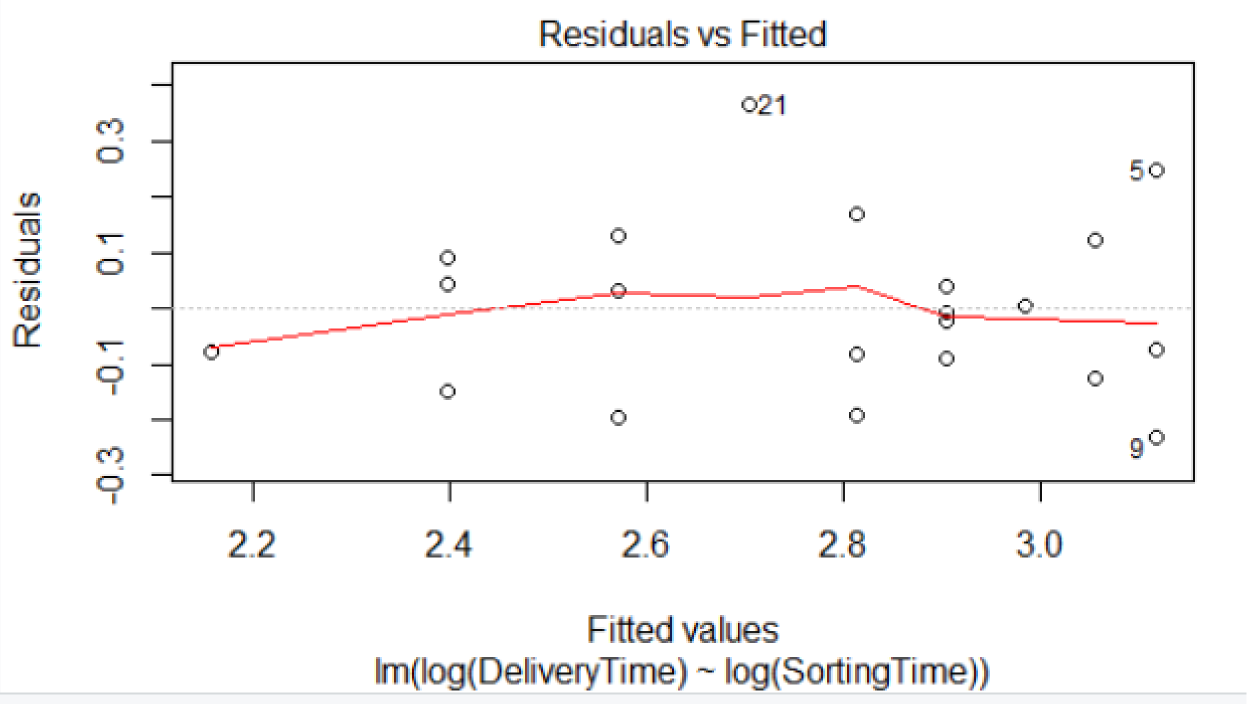

Residuals vs Fitted
Residuals
8
6
4
2
0
-2
-6
21
5
9
10
12
14
16
18
20
22
Fitted values
lm(DeliveryTime ~ SortingTime)

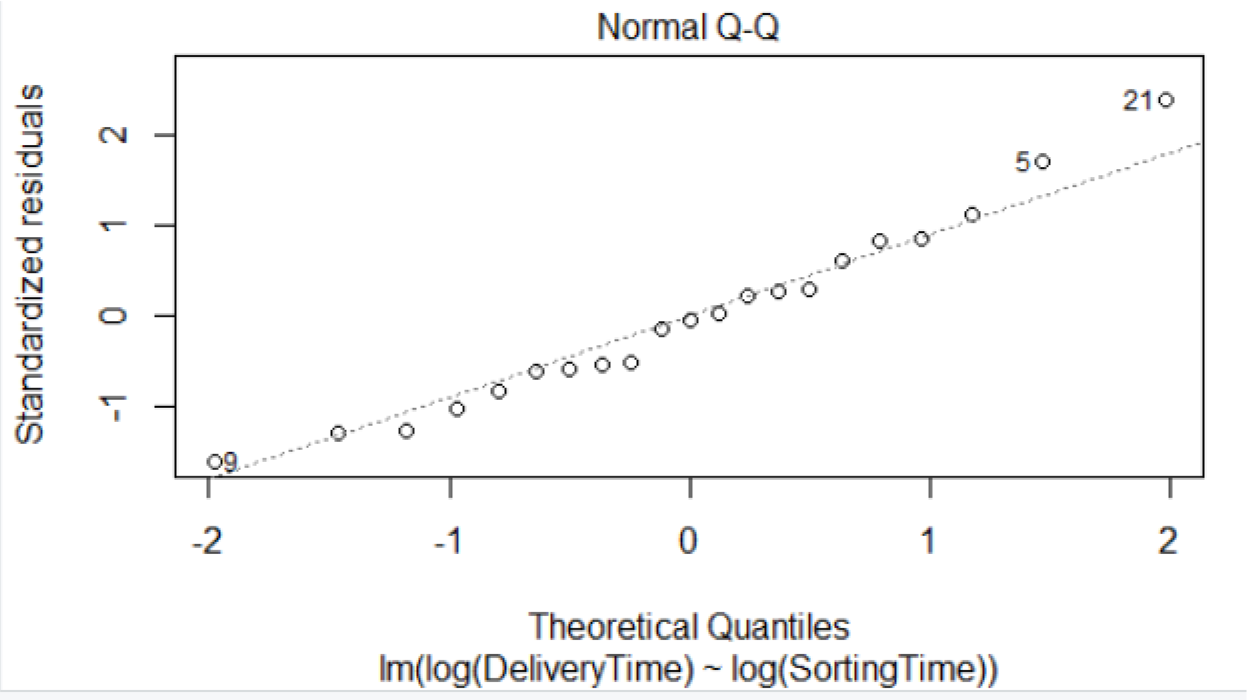

Normal Q-Q
Standardized residuals
2
1
0
-1
-2
5
21
9
-2
-1
0
1
2
Theoretical Quantiles
lm(DeliveryTime ~ SortingTime)

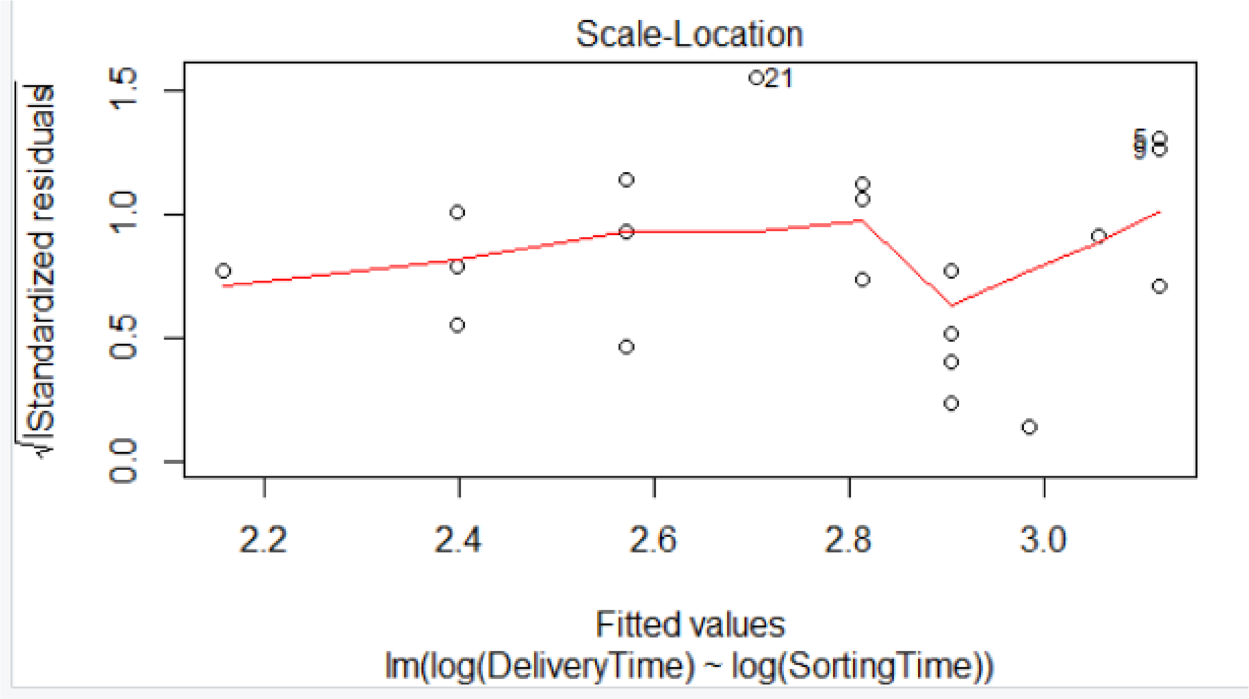


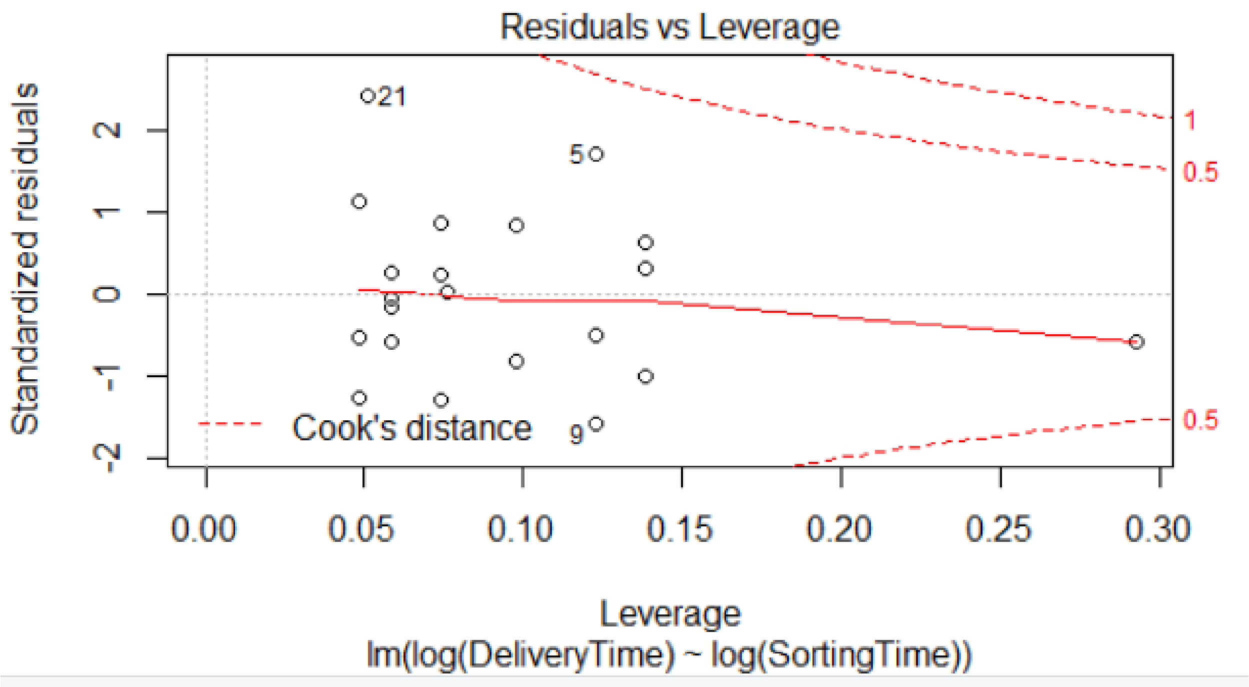


```
> model2 <- lm(log(DeliveryTime)~log(SortingTime),data = Delivery_time)
> summary(model2)

Call:
lm(formula = log(DeliveryTime) ~ log(SortingTime), data = Delivery_tim
e)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.23303 -0.09050 -0.00825  0.08897  0.36439 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)       1.74199    0.13312  13.086 5.92e-11 ***
log(SortingTime)  0.59752    0.07446   8.024 1.60e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1558 on 19 degrees of freedom
Multiple R-squared:  0.7722,	Adjusted R-squared:  0.7602 
F-statistic: 64.39 on 1 and 19 DF,  p-value: 1.602e-07

> confint(model2,level = 0.95)
                     2.5 %    97.5 %
(Intercept)      1.4633576 2.0206166
log(SortingTime) 0.4416707 0.7533739
> predict(model2,interval = "predict")
        fit      lwr      upr
1  3.117833 2.772199 3.463468
2  2.570329 2.232244 2.908413
3  2.812603 2.478606 3.146601
4  3.054878 2.713126 3.396630
5  3.117833 2.772199 3.463468
6  2.812603 2.478606 3.146601
7  2.904712 2.569144 3.240279
8  2.398432 2.050448 2.746417
9  3.117833 2.772199 3.463468
10 3.054878 2.713126 3.396630
11 2.984500 2.646196 3.322803
12 2.570329 2.232244 2.908413
13 2.904712 2.569144 3.240279
14 2.398432 2.050448 2.746417
15 2.398432 2.050448 2.746417
16 2.570329 2.232244 2.908413
17 2.812603 2.478606 3.146601
18 2.904712 2.569144 3.240279
19 2.156158 1.785357 2.526959
20 2.904712 2.569144 3.240279
21 2.703662 2.369295 3.038029

> plot(model2)
Hit <Return> to see next plot: 
Hit <Return> to see next plot: 
Hit <Return> to see next plot: 
Hit <Return> to see next plot: 
```

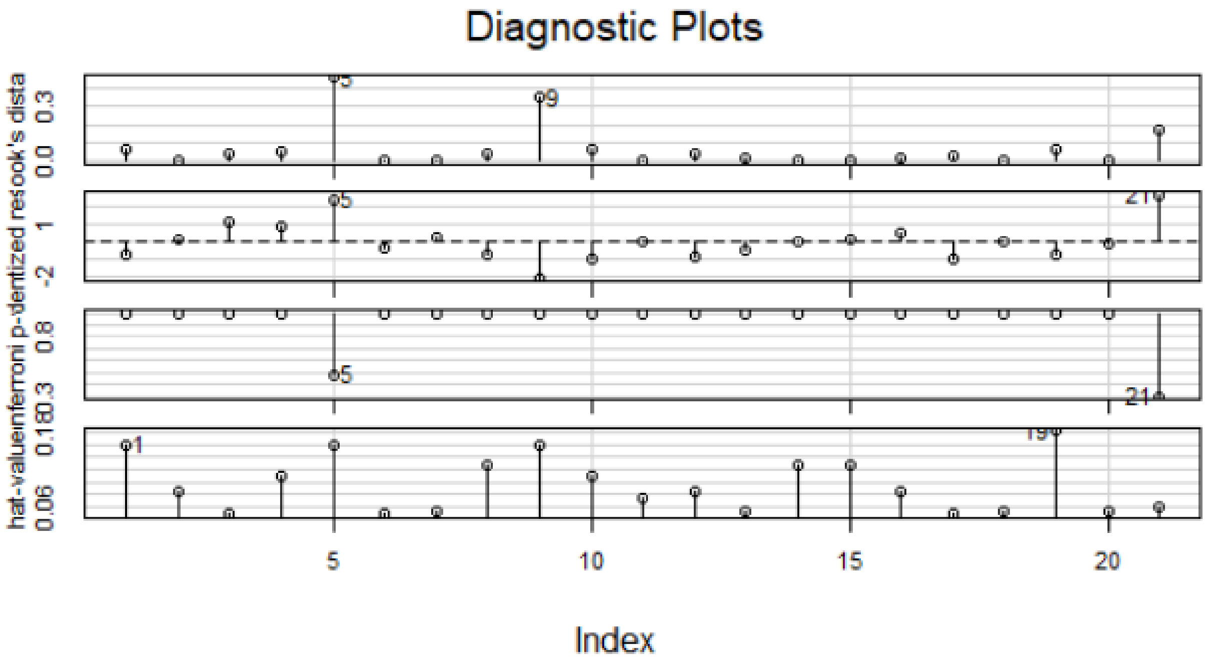

Residuals vs Fitted
Residuals
0.3
0.1
-0.1
-0.3
2.2
2.4
2.6
2.8
3.0
21
5
9
Fitted values
lm(log(DeliveryTime) ~ log(SortingTime))


Normal Q-Q
Standardized residuals
2
1
0
-1
-2
-1
0
1
2
21
5
9
Theoretical Quantiles
lm(log(DeliveryTime) ~ log(SortingTime))

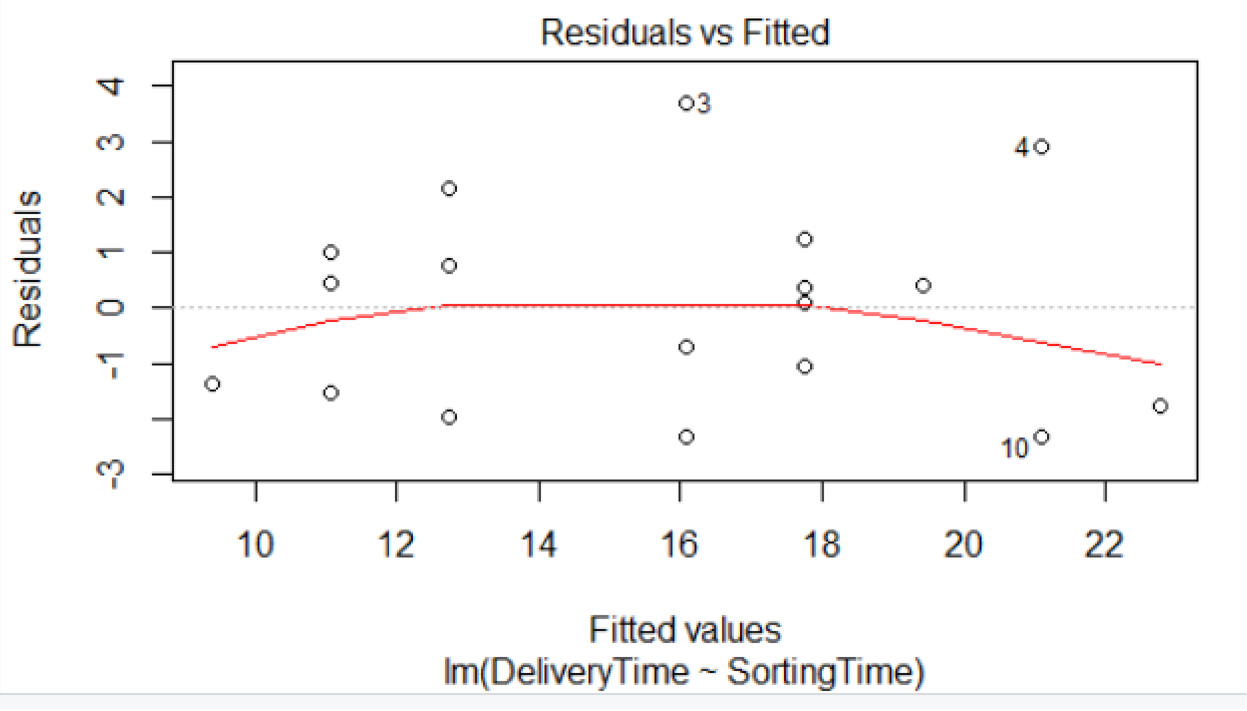


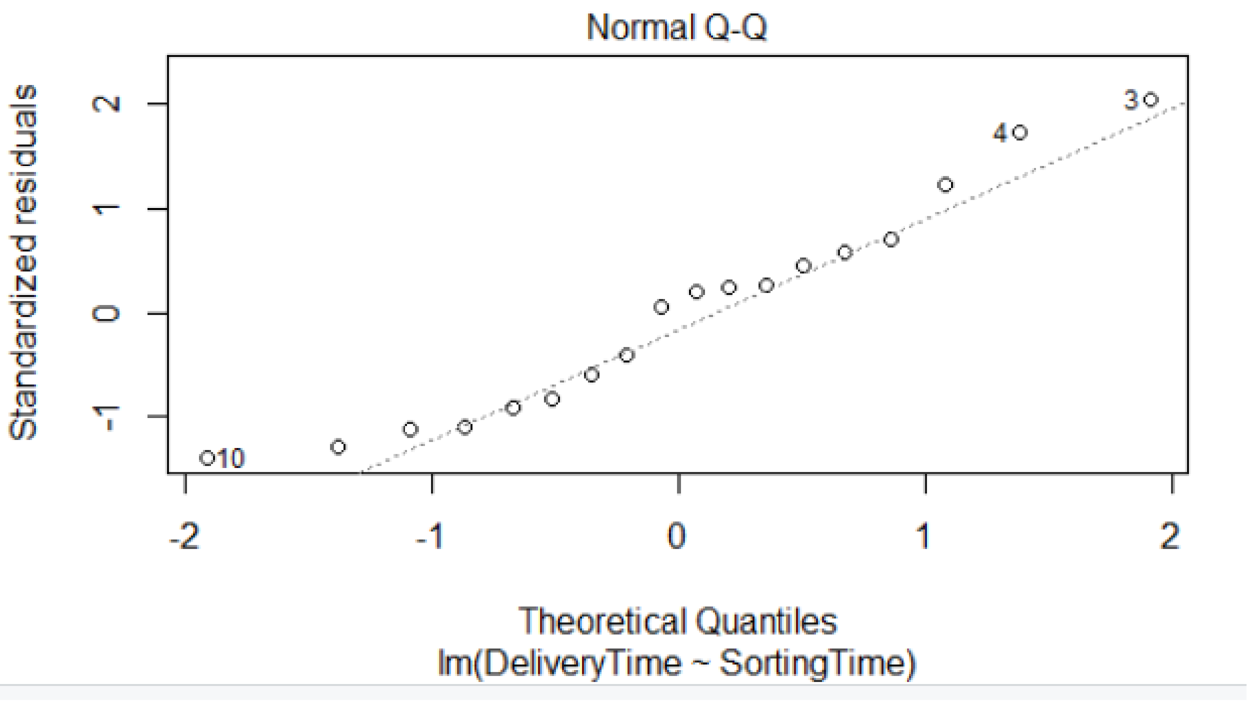


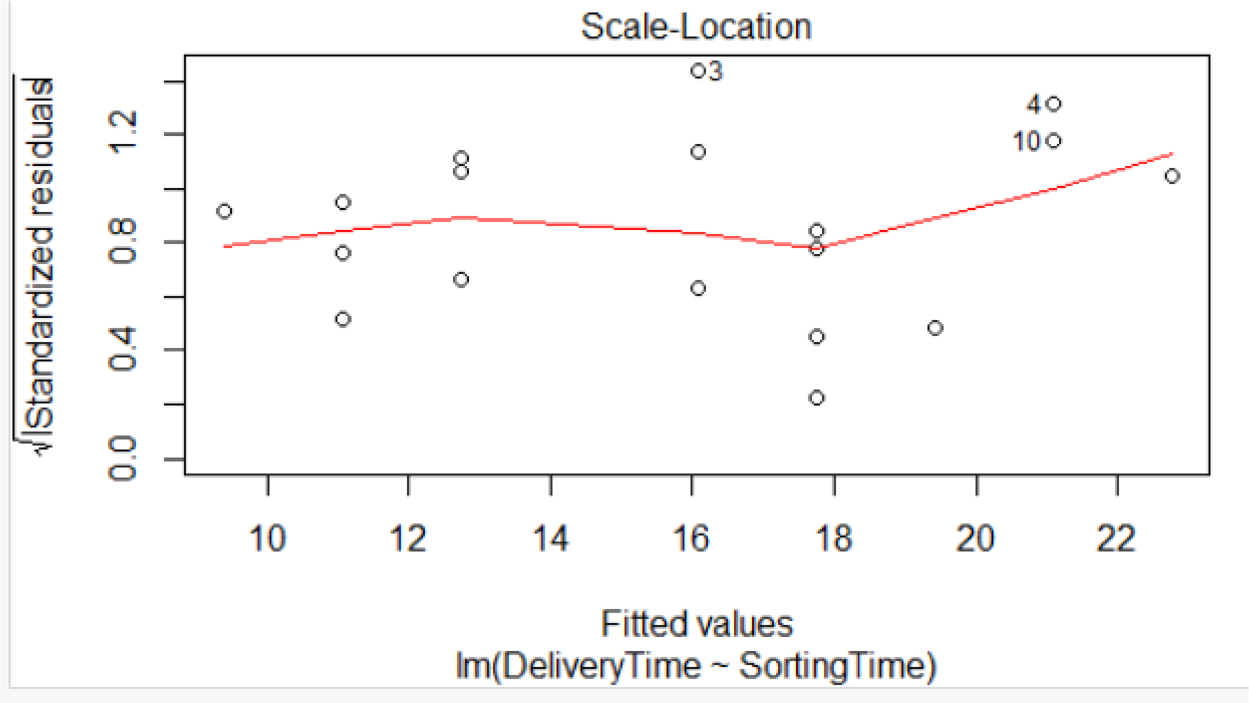
```
> library(mvinfluence)
> influenceIndexPlot(model1)
```

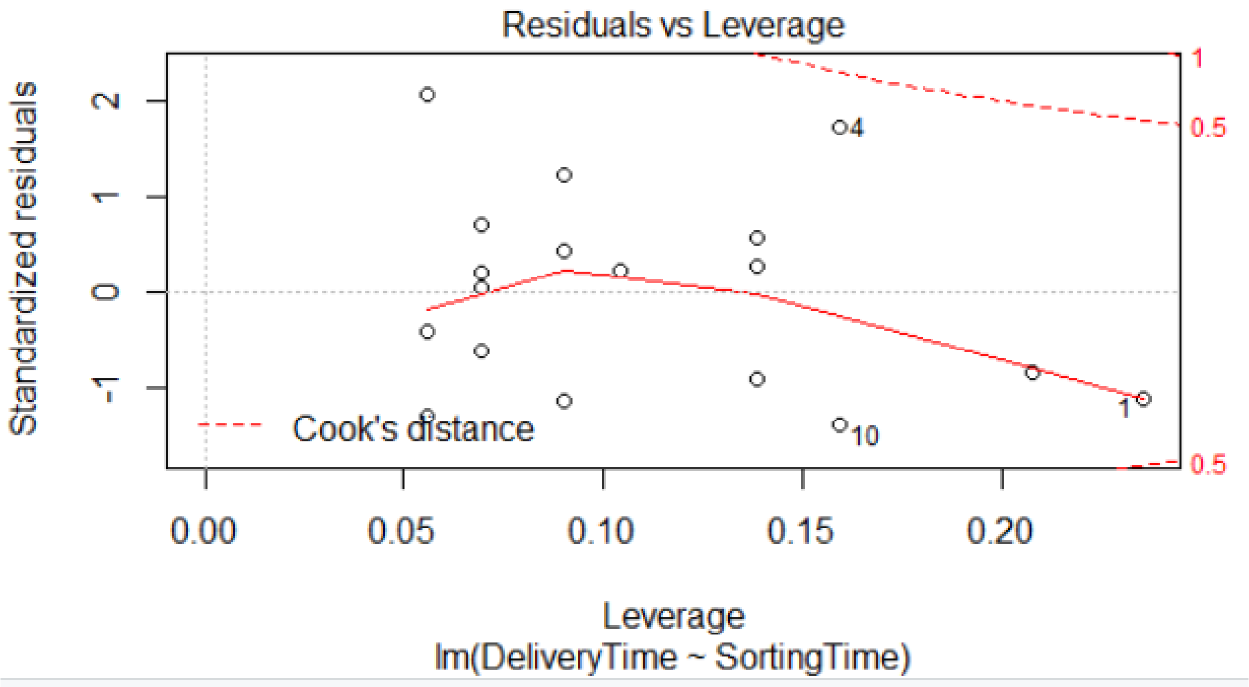


```
> model3 <- lm(DeliveryTime ~ SortingTime, data = Delivery_time[c(-5,-9,-21),])
> summary(model3)

Call:
lm(formula = DeliveryTime ~ SortingTime, data = Delivery_time[c(-5,
    -9, -21), ])

Residuals:
    Min      1Q  Median      3Q     Max 
-2.3407 -1.5027  0.2275  0.9328  3.6815 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   6.0240     1.1751   5.126 0.000102 ***
SortingTime   1.6741     0.1872   8.941 1.27e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.839 on 16 degrees of freedom
Multiple R-squared:  0.8332,	Adjusted R-squared:  0.8228 
F-statistic: 79.94 on 1 and 16 DF,  p-value: 1.273e-07
> plot(model3)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot
```


Residuals vs Fitted
Residuals
-3
-1
0
1
2
3
4
3
4
10
10
12
14
16
18
20
22
Fitted values
lm(DeliveryTime ~ SortingTime)


Normal Q-Q
Standardized residuals
-1
0
1
2
10
4
3
-2
-1
0
1
2
Theoretical Quantiles
lm(DeliveryTime ~ SortingTime)





Conclusion- p-value is less than 0.05. and Multiple R- squared value is 0.8332. This means the regression model will predict the output 83.32% time correct
Residual standard error: 1.839 on 16 degrees of freedom

Problem statement 3

Emp_data -> Build a prediction model for Churn_out_rate

Answer:

Rcode:

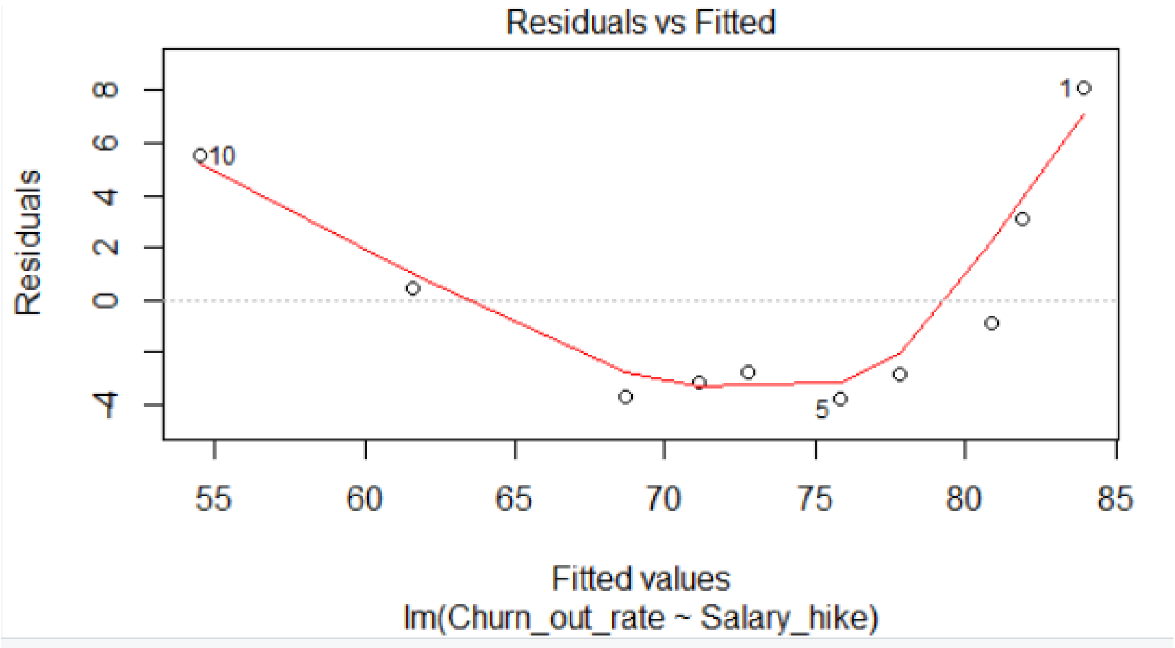
```
##3) Emp_data -> Build a prediction model for Churn_out_rate ##

Emp_data <- read.csv(file.choose())
summary(Emp_data)
attach(Emp_data)

Churn_out_rate_model1 <- lm(Churn_out_rate~Salary_hike,data = Emp_data)
summary(Churn_out_rate_model1)
confint(Churn_out_rate_model1,level = 0.95)
predict(Churn_out_rate_model1,interval = "predict")
plot(Churn_out_rate_model1)

Churn_out_rate_model2 <- lm(Churn_out_rate~log(Salary_hike),data = Emp_data)
summary(Churn_out_rate_model2)
confint(Churn_out_rate_model2,level = 0.95)
predict(Churn_out_rate_model2,interval = "predict")
plot(Churn_out_rate_model2)

Churn_out_rate_model3 <- lm(log(Churn_out_rate)~log(Salary_hike),data = Emp_data)
summary(Churn_out_rate_model3)
confint(Churn_out_rate_model3,level = 0.95)
```

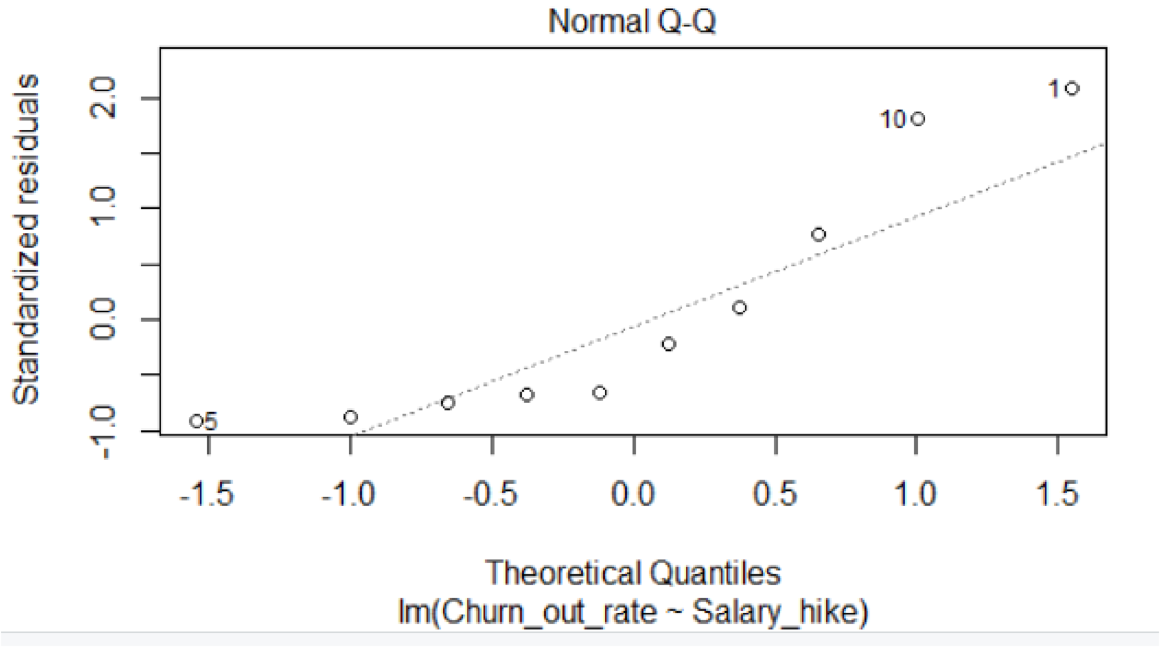
```
predict(Churn_out_rate_model3,interval = "predict")

plot(Churn_out_rate_model3)
```

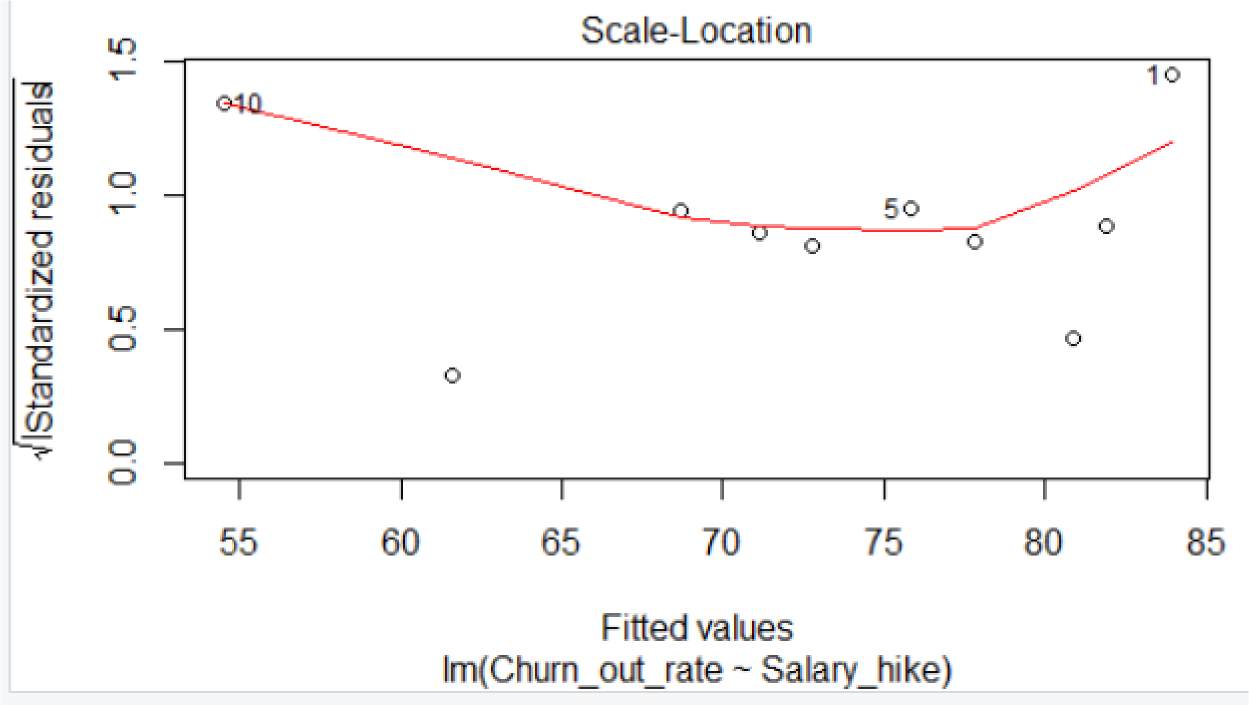
Console:

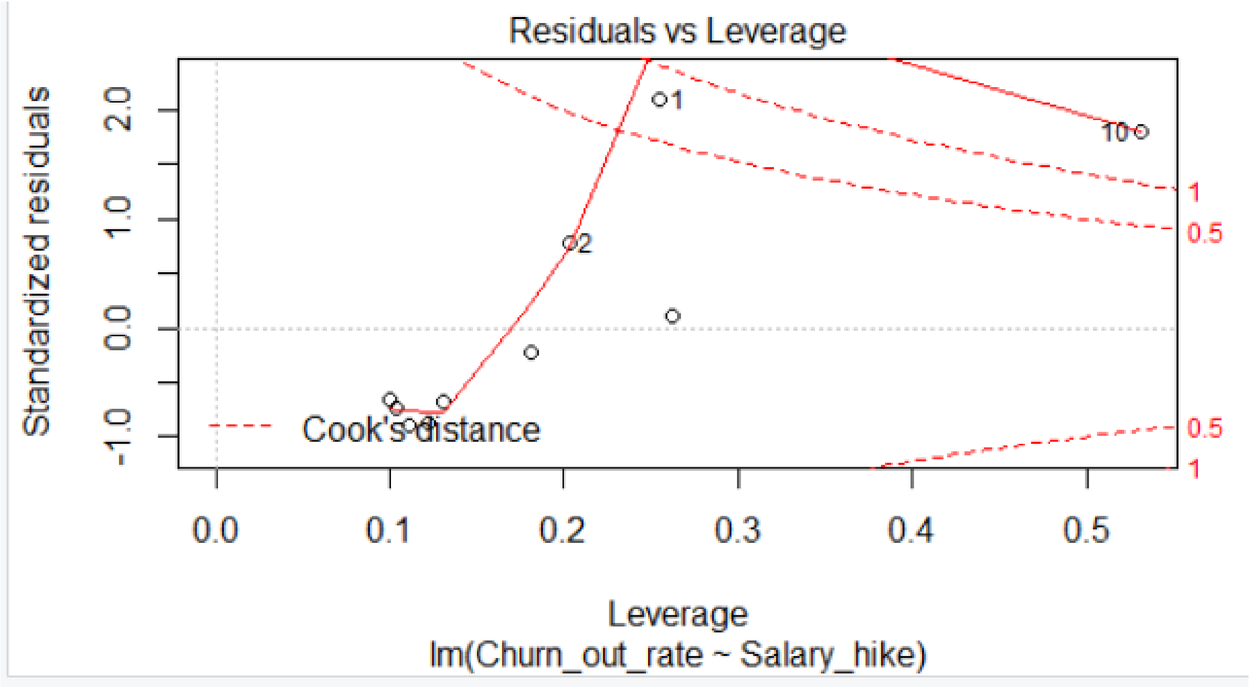
```
> Emp_data <- read.csv(file.choose())
> summary(Emp_data)
  Salary_hike    Churn_out_rate 
 Min.   :1580   Min.   :60.00  
 1st Qu.:1618   1st Qu.:65.75  
 Median :1675   Median :71.00  
 Mean   :1689   Mean   :72.90  
 3rd Qu.:1724   3rd Qu.:78.75  
 Max.   :1870   Max.   :92.00  
> attach(Emp_data)

> Churn_out_rate_model1 <- lm(Churn_out_rate~Salary_hike,data = Emp_
data)
> summary(Churn_out_rate_model1)

Call:
lm(formula = Churn_out_rate ~ Salary_hike, data = Emp_data)

Residuals:
   Min     1Q Median     3Q    Max 
-3.804 -3.059 -1.819  2.430  8.072 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 244.36491   27.35194   8.934 1.96e-05 ***
Salary_hike  -0.10154    0.01618  -6.277 0.000239 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.469 on 8 degrees of freedom
Multiple R-squared:  0.8312,	Adjusted R-squared:  0.8101 
F-statistic:  39.4 on 1 and 8 DF,  p-value: 0.0002386

> confint(Churn_out_rate_model1,level = 0.95)
                   2.5 %      97.5 %
(Intercept) 181.2912317 307.4385905
Salary_hike  -0.1388454  -0.0642399
> predict(Churn_out_rate_model1,interval = "predict")
        fit      lwr      upr
1  83.92753 72.38391 95.47115
2  81.89668 70.59327 93.20009
3  80.88125 69.68123 92.08127
4  77.83497 66.87456 88.79538
5  75.80412 64.94216 86.66607
6  72.75784 61.94828 83.56740
7  71.13316 60.30425 81.96206
8  68.69613 57.77694 79.61533
```

```
9  61.58815 50.00746 73.16884
10 54.48016 41.72742 67.23290

> plot(Churn_out_rate_model1)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
```



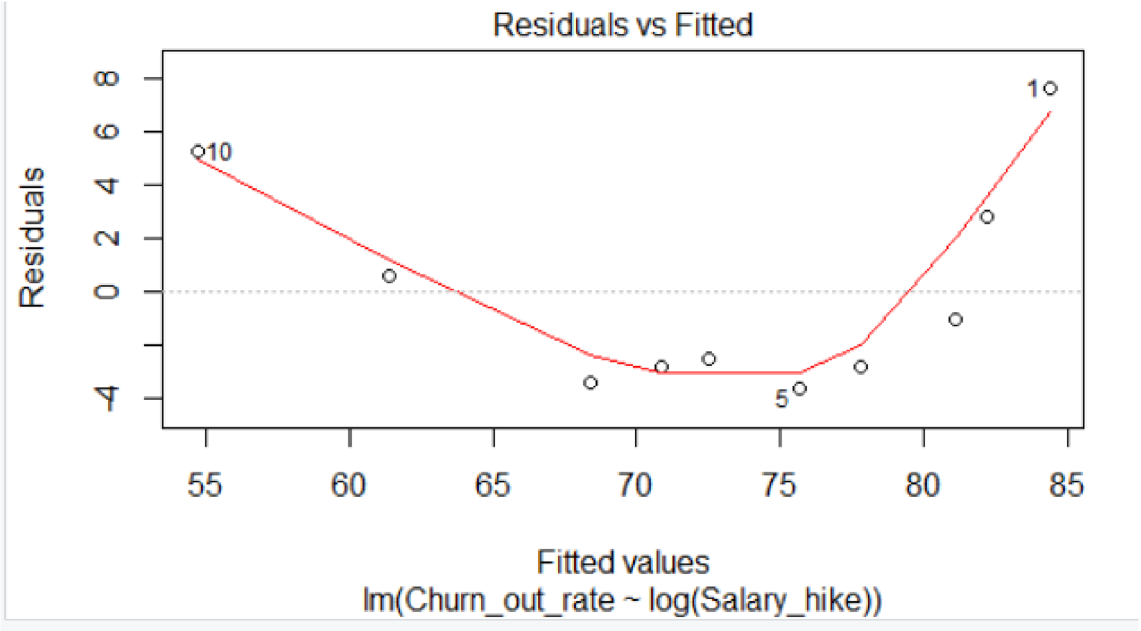


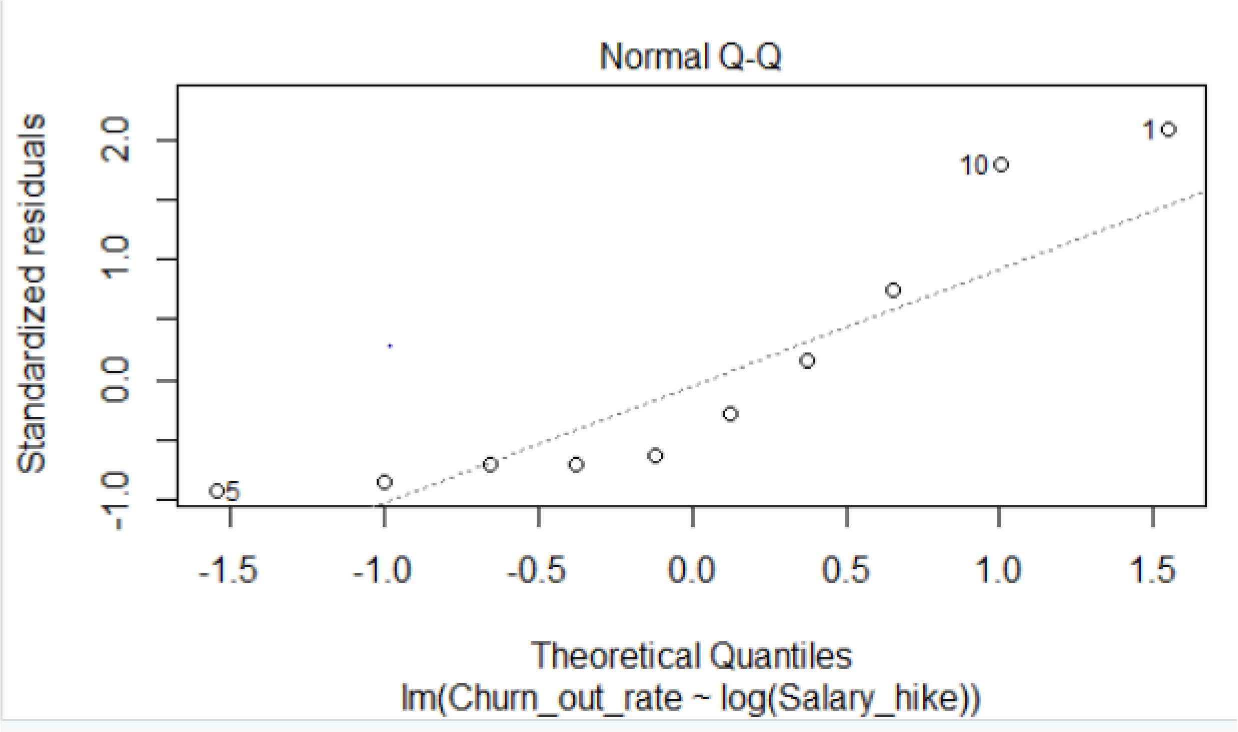


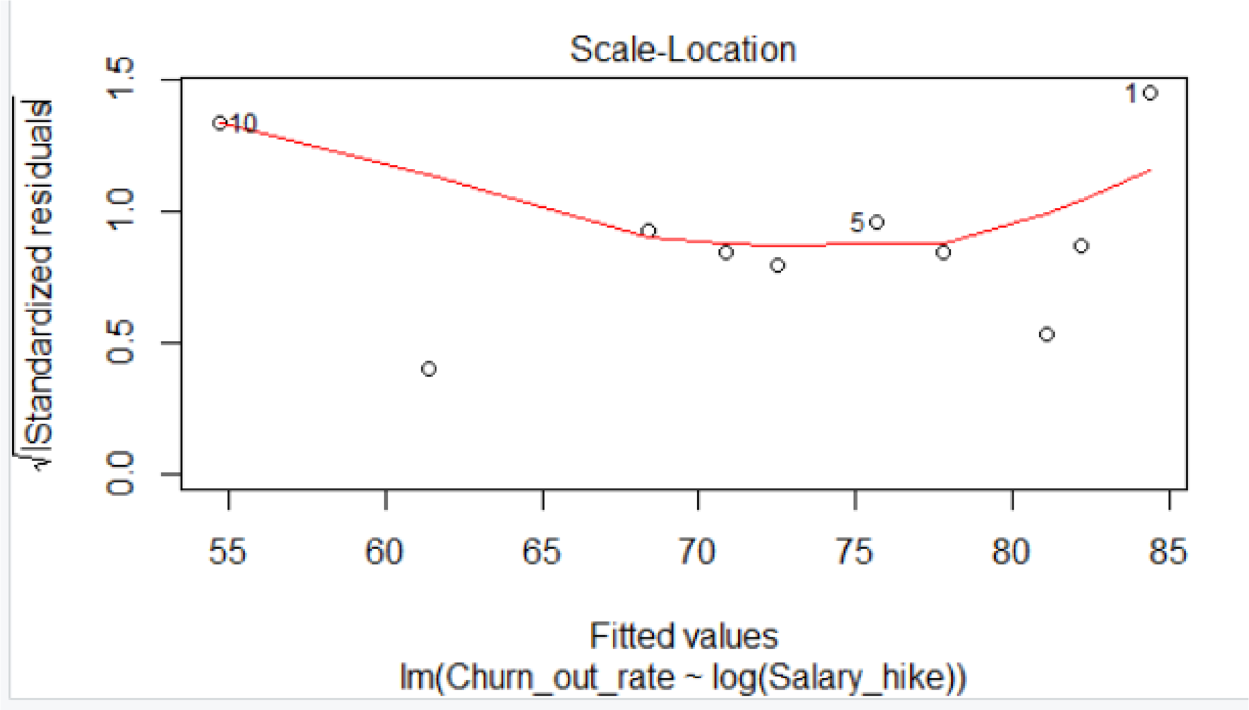


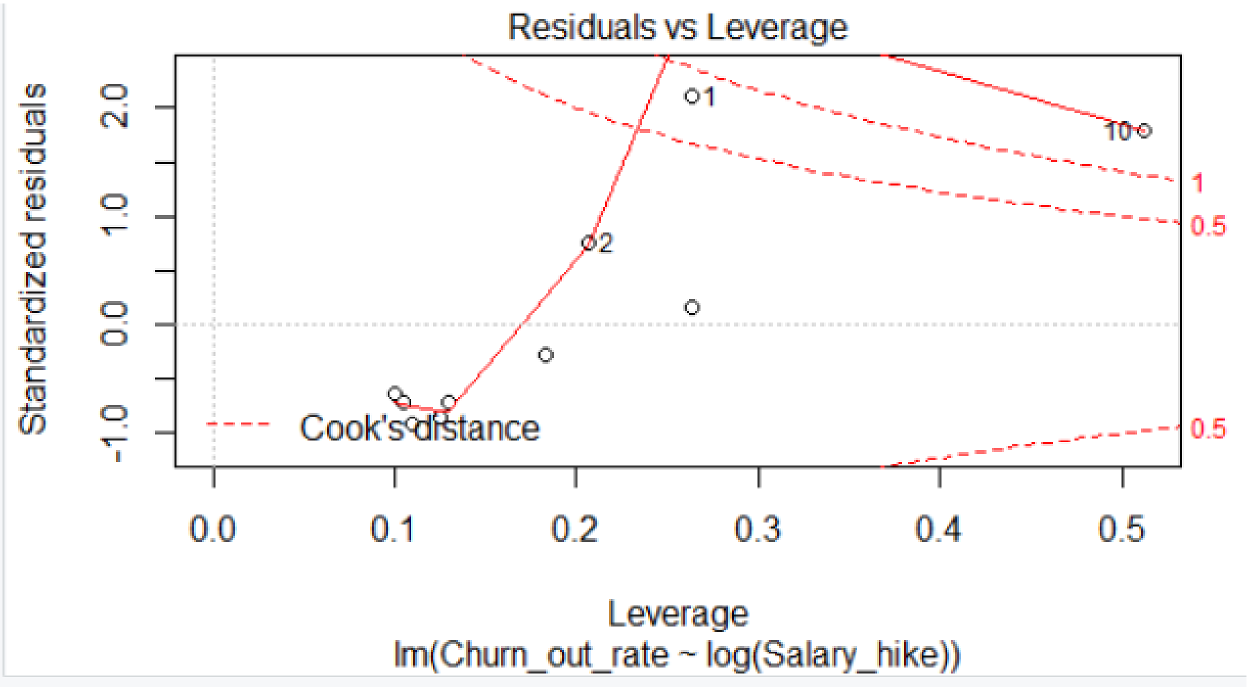
```
> Churn_out_rate_model2 <- lm(Churn_out_rate~log(Salary_hike),data = E
mp_data)
> summary(Churn_out_rate_model2)

Call:
lm(formula = Churn_out_rate ~ log(Salary_hike), data = Emp_data)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-3.678 -2.851 -1.794  2.275  7.624

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)        1381.5      195.4   7.070 0.000105 ***
log(Salary_hike)   -176.1       26.3  -6.697 0.000153 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.233 on 8 degrees of freedom
Multiple R-squared:  0.8486,	Adjusted R-squared:  0.8297 
F-statistic: 44.85 on 1 and 8 DF,  p-value: 0.0001532

> confint(Churn_out_rate_model2,level = 0.95)
                     2.5 %    97.5 %
(Intercept)       930.8584 1832.0540
log(Salary_hike) -236.7512 -115.4682
> predict(Churn_out_rate_model2,interval = "predict")
        fit      lwr      upr
1  84.37627 73.40258 95.34996
2  82.16102 71.43838 92.88366
3  81.06376 70.44736 91.68017
4  77.81241 67.43614 88.18869
5  75.67773 65.39568 85.95978
6  72.52344 62.28515 82.76172
7  70.86397 60.60253 81.12541
8  68.40372 58.04985 78.75760
9  61.41829 50.44392 72.39265
10 54.69939 42.69592 66.70286
> plot(Churn_out_rate_model2)
Hit <Return> to see next plot: 
Hit <Return> to see next plot: 
Hit <Return> to see next plot: 
Hit <Return> to see next plot: 
```



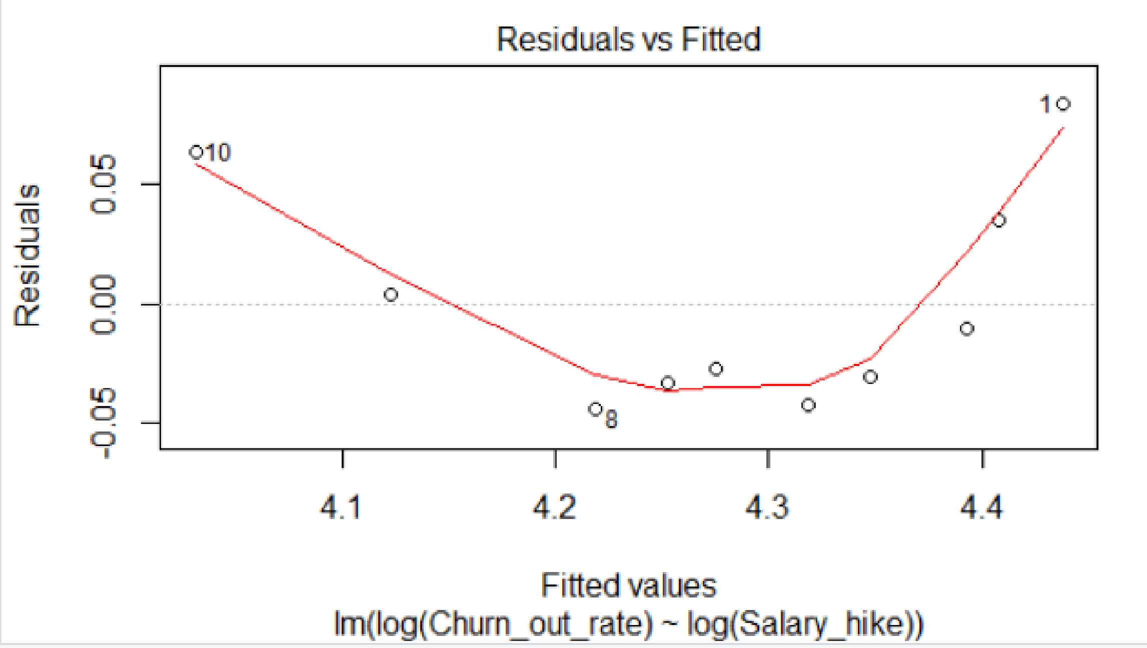

Normal Q-Q
Standardized residuals
2.0
1.0
0.0
-1.0
1
10
5
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
Theoretical Quantiles
lm(Churn_out_rate ~ log(Salary_hike))

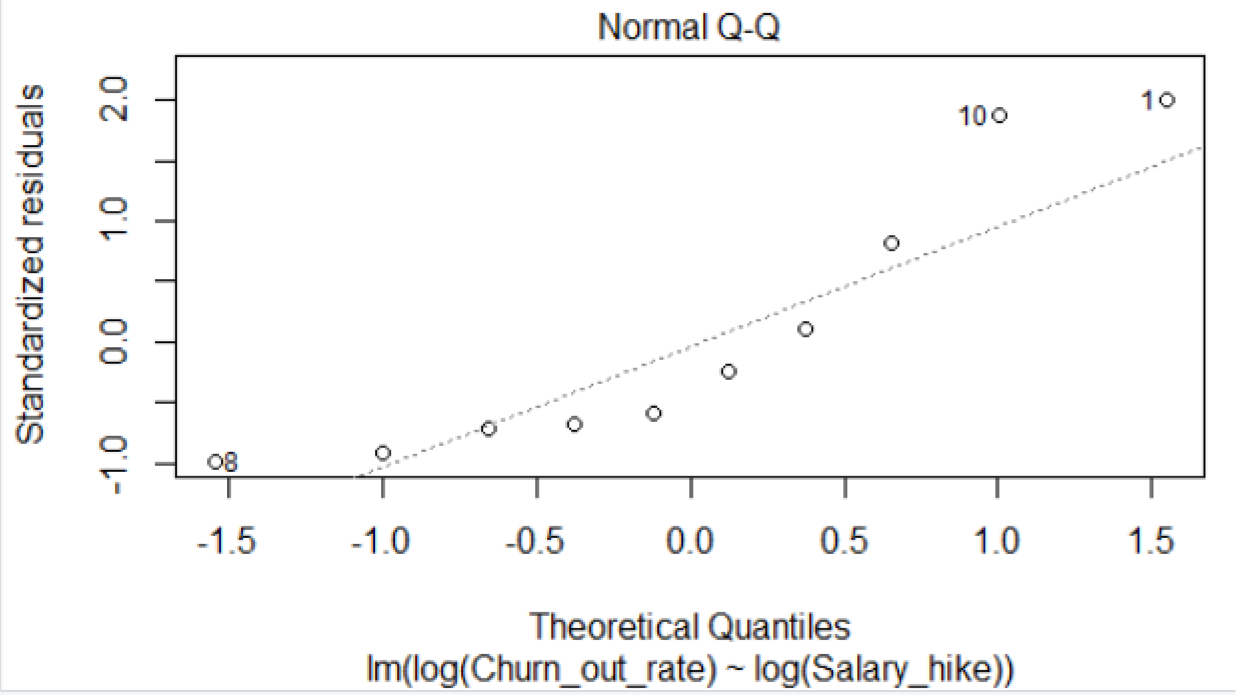

Scale-Location
√|Standardized residuals|
1.5
1.0
0.5
0.0
10
1
5
55
60
65
70
75
80
85
Fitted values
lm(Churn_out_rate ~ log(Salary_hike))

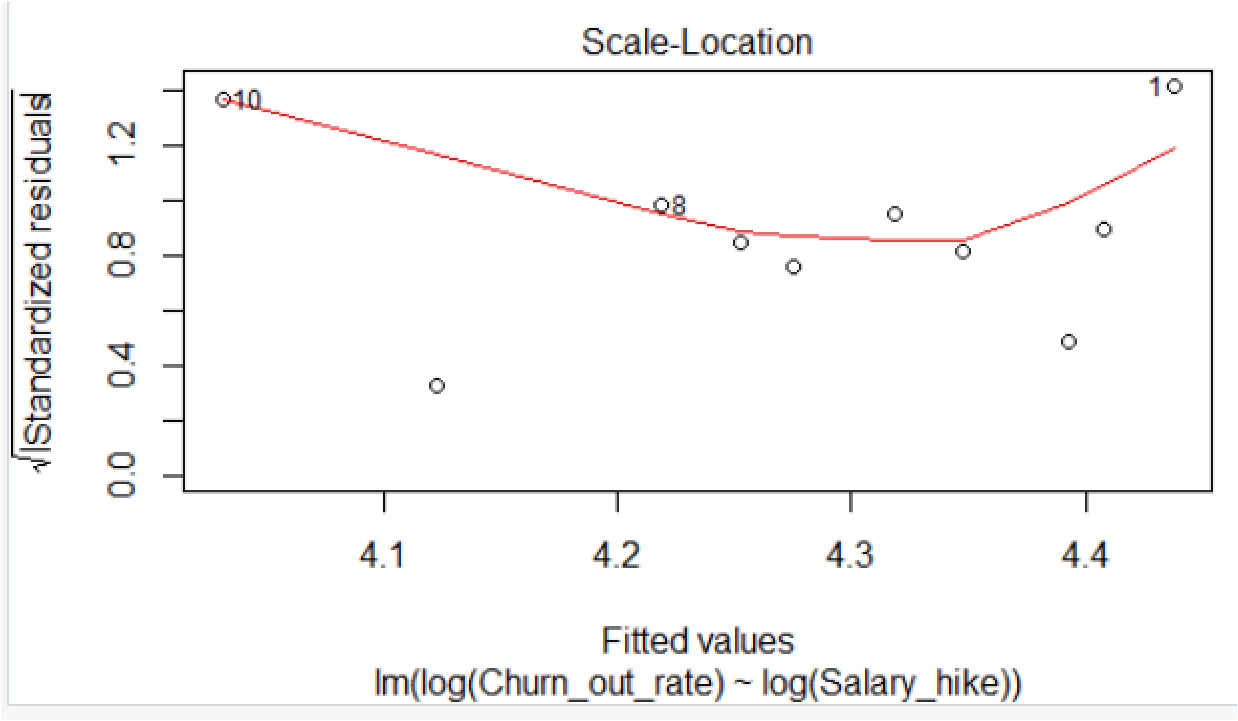


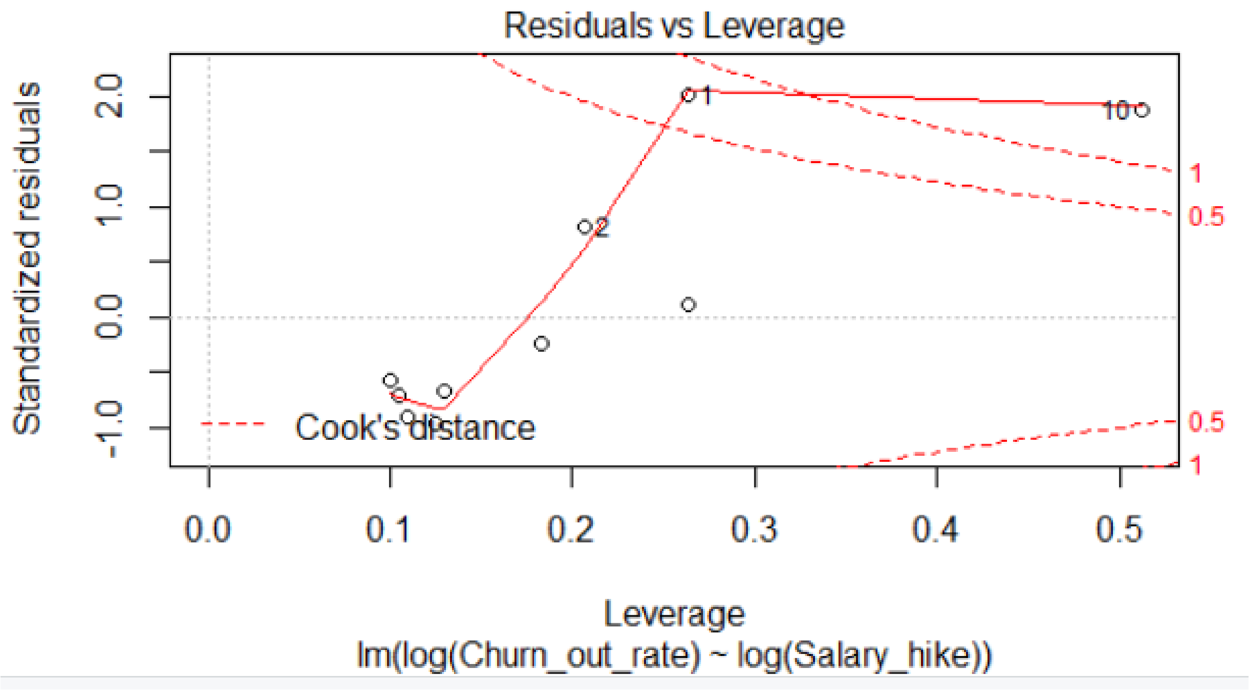
```
> Churn_out_rate_model3 <- lm(log(Churn_out_rate)~log(Salary_hike),da
ta = Emp_data)
> summary(Churn_out_rate_model3)

Call:
lm(formula = log(Churn_out_rate) ~ log(Salary_hike), data = Emp_data)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.04433 -0.03234 -0.01865  0.02737  0.08377 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)       22.2472     2.2436   9.916 9.04e-06 ***
log(Salary_hike)  -2.4180     0.3019  -8.008 4.33e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0486 on 8 degrees of freedom
Multiple R-squared:  0.8891,	Adjusted R-squared:  0.8752 
F-statistic: 64.13 on 1 and 8 DF,  p-value: 4.335e-05

> confint(Churn_out_rate_model3,level = 0.95)
                     2.5 %    97.5 %
(Intercept)      17.073481 27.420881
log(Salary_hike) -3.114298 -1.721744
> predict(Churn_out_rate_model3,interval = "predict")
        fit      lwr      upr
1  4.438020 4.312022 4.564019
2  4.407605 4.284489 4.530720
3  4.392539 4.270643 4.514435
4  4.347897 4.228759 4.467036
```

```
5  4.318588 4.200531 4.436645
6  4.275279 4.157724 4.392833
7  4.252494 4.134673 4.370314
8  4.218714 4.099832 4.337596
9  4.122803 3.996797 4.248809
10 4.030551 3.892729 4.168373

> plot(Churn_out_rate_model3)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
```









Conclusion- p-value is less than 0.05. and Multiple R- squared value is 0.8891.
This means the regression model will predict the output 88.91% time correct
Residual standard error: 0.0486 on 8 degrees of freedom

Problem statement 4

Salary_hike -> Build a prediction model for Salary_hike

Answer:

Rcode:

```
## 4) Salary_hike -> Build a prediction model for Salary_hike ##

Salary_hike <- read.csv(file.choose())
summary(Salary_hike)
names(Salary_hike)
attach(Salary_hike)

Salary_hike_model <- lm(Salary~YearsExperience,data = Salary_hike)
summary(Salary_hike_model)
confint(Salary_hike_model,level = 0.95)
predict(Salary_hike_model,interval = "predict")
plot(Salary_hike_model)

Salary_hike_model1 <- lm(Salary~log(YearsExperience),data = Salary_hike)
summary(Salary_hike_model1)
confint(Salary_hike_model1,level = 0.95)
predict(Salary_hike_model1,interval = "predict")
plot(Salary_hike_model1)

Salary_hike_model2 <- lm(log(Salary)~YearsExperience,data = Salary_hike)
summary(Salary_hike_model2)
confint(Salary_hike_model2,level = 0.95)
predict(Salary_hike_model2,interval = "predict")
```

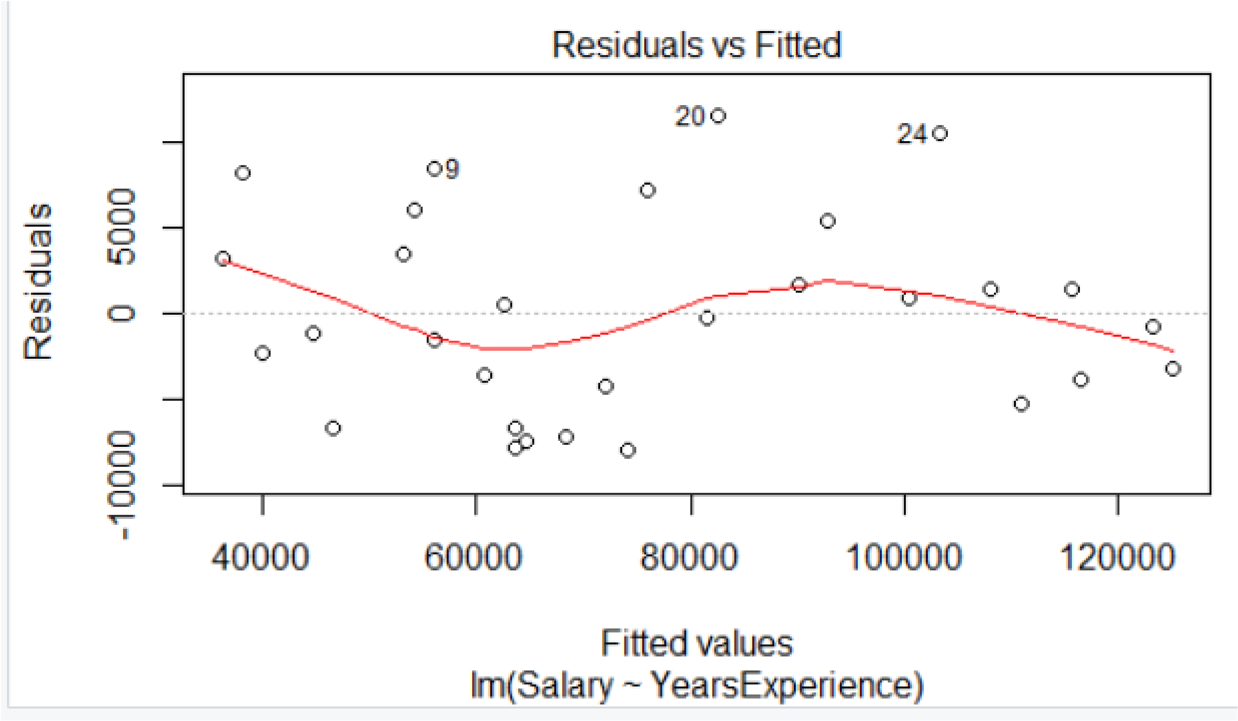
plot(Salary_hike_model2)

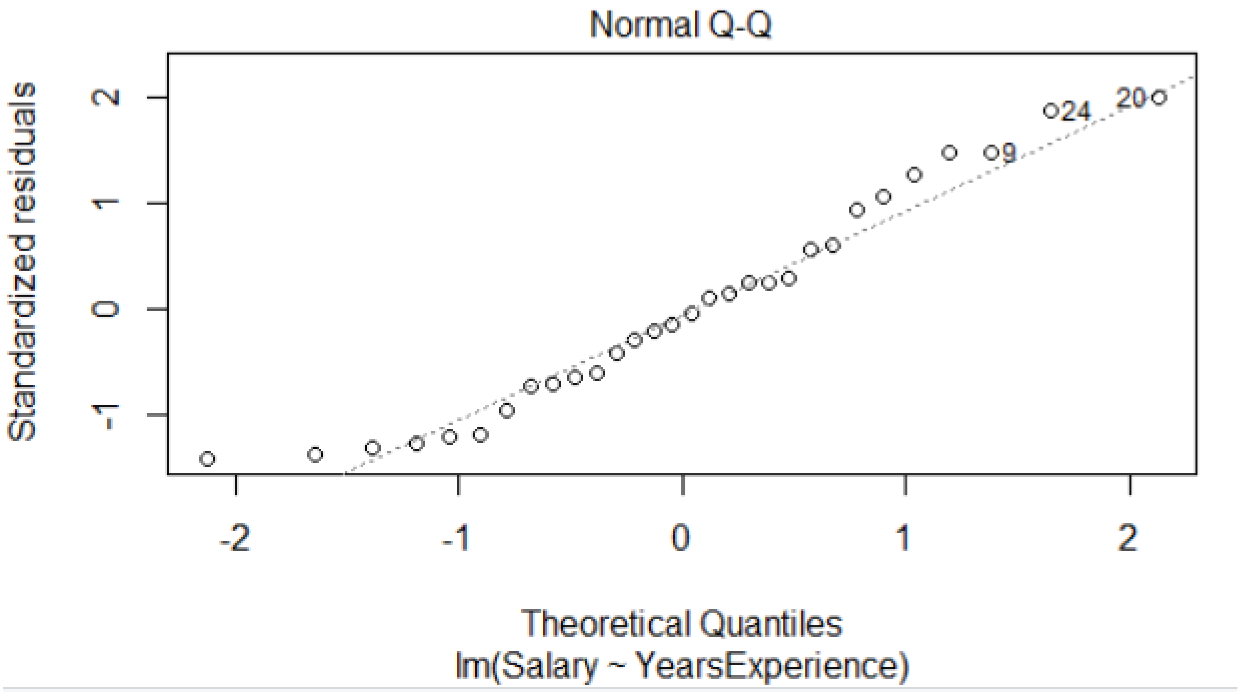
Console:

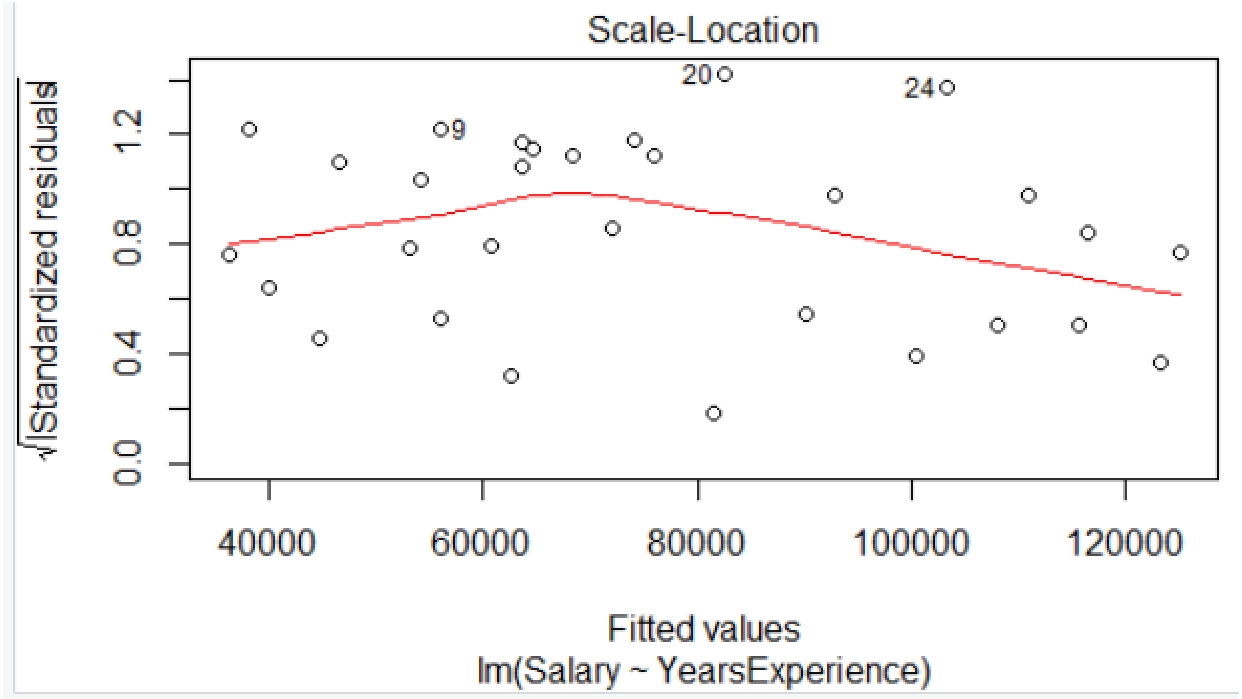
```
> Salary_hike <- read.csv(file.choose())
> summary(Salary_hike)
 YearsExperience      Salary      
 Min.   : 1.100   Min.   : 37731  
 1st Qu.: 3.200   1st Qu.: 56721  
 Median : 4.700   Median : 65237  
 Mean   : 5.313   Mean   : 76003  
 3rd Qu.: 7.700   3rd Qu.:100545  
 Max.   :10.500   Max.   :122391  
> names(Salary_hike)
[1] "YearsExperience" "Salary"         
> attach(Salary_hike)
> Salary_hike_model <- lm(Salary~YearsExperience,data = Salary_hike)
> summary(Salary_hike_model)

Call:
lm(formula = Salary ~ YearsExperience, data = Salary_hike)

Residuals:
    Min      1Q  Median      3Q     Max 
-7958.0 -4088.5  -459.9  3372.6 11448.0 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)      25792.2     2273.1   11.35 5.51e-12 ***
YearsExperience   9450.0      378.8   24.95  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5788 on 28 degrees of freedom
Multiple R-squared:  0.957,	Adjusted R-squared:  0.9554 
F-statistic: 622.5 on 1 and 28 DF,  p-value: < 2.2e-16

> confint(Salary_hike_model,level = 0.95)
                    2.5 %   97.5 %
(Intercept)     21136.061 30448.34
YearsExperience  8674.119 10225.81
> predict(Salary_hike_model,interval = "predict")
         fit       lwr       upr
1   36187.16  23698.92  48675.40
2   38077.15  25628.63  50525.67
3   39967.14  27556.52  52377.76
4   44692.12  32368.22  57016.03
5   46582.12  34289.64  58874.59
6   53197.09  40999.70  65394.48
7   54142.09  41956.37  66327.80
8   56032.08  43868.25  68195.91
9   56032.08  43868.25  68195.91
10  60757.06  48639.42  72874.70
```

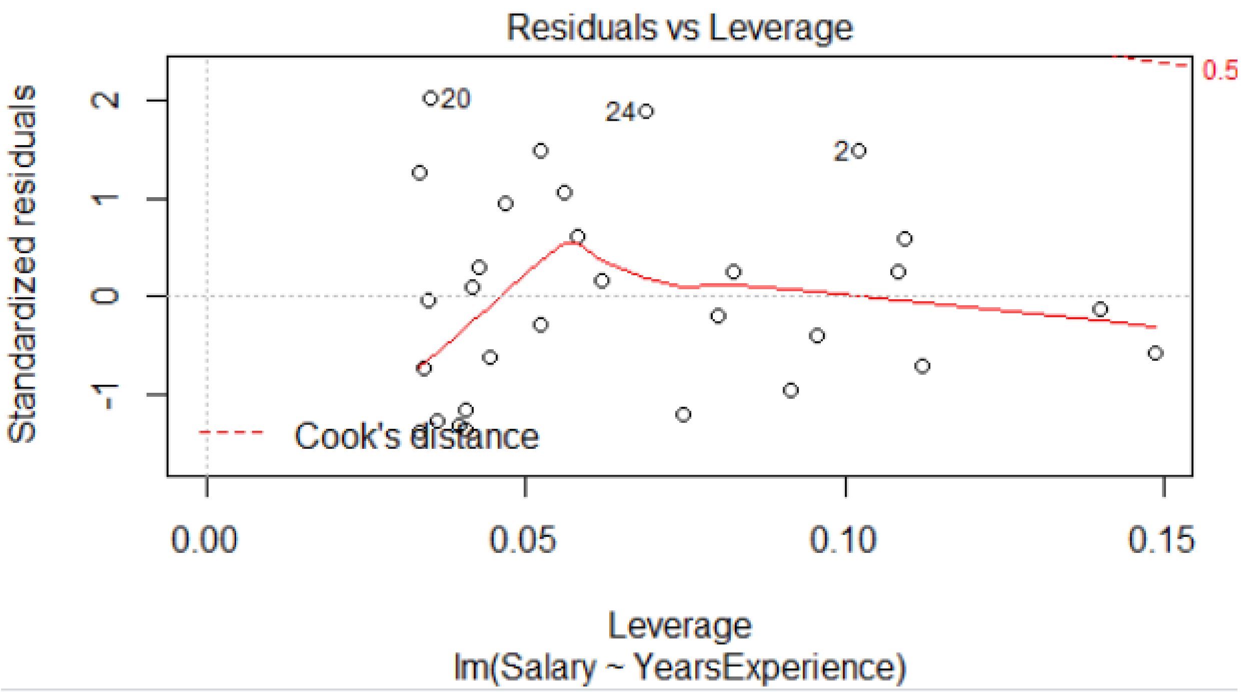
```
11  62647.05  50544.46  74749.65
12  63592.05  51496.24  75687.86
13  63592.05  51496.24  75687.86
14  64537.05  52447.52  76626.57
15  68317.03  56247.70  80386.36
16  72097.02  60039.93  84154.10
17  73987.01  61933.05  86040.96
18  75877.00  63824.18  87929.82
19  81546.98  69485.57  93608.39
20  82491.97  70427.39  94556.56
21  90051.94  77944.06 102159.83
22  92886.93  80754.66 105019.20
23 100446.90  88228.15 112665.65
24 103281.89  91022.76 115541.02
25 108006.87  95670.98 120342.77
26 110841.86  98454.30 123229.42
27 115566.84 103084.00 128049.68
28 116511.84 104008.59 129015.09
29 123126.81 110468.27 135785.35
30 125016.80 112309.98 137723.63
```

```
> plot(Salary_hike_model)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
```



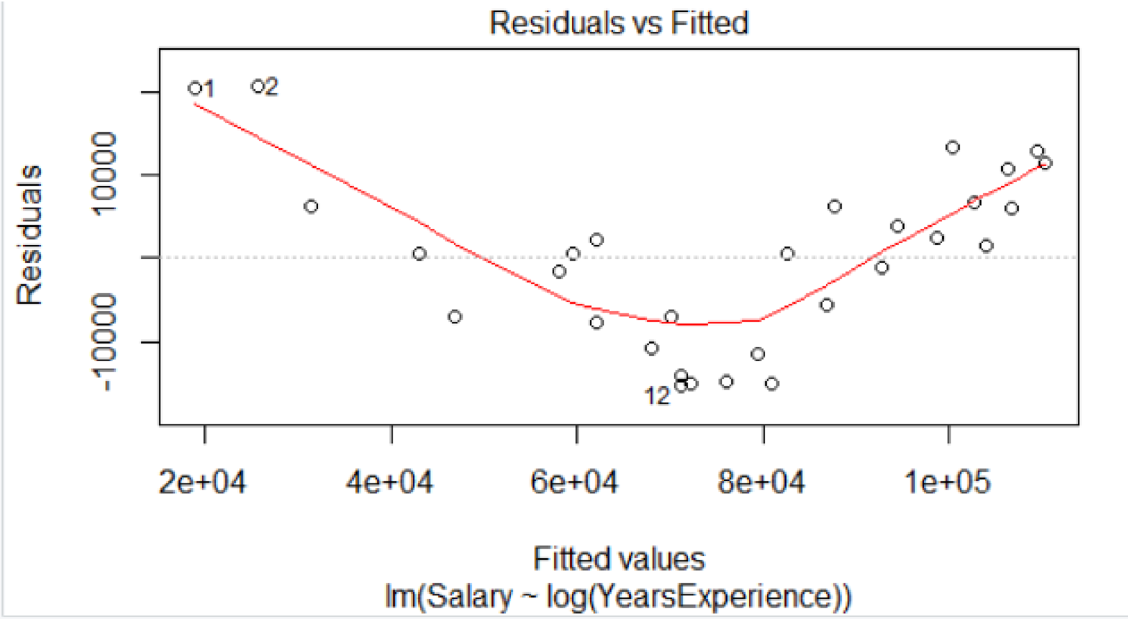

Normal Q-Q
Standardized residuals
2
1
0
-1
-2
-1
0
1
2
24
20
9
Theoretical Quantiles
lm(Salary ~ YearsExperience)

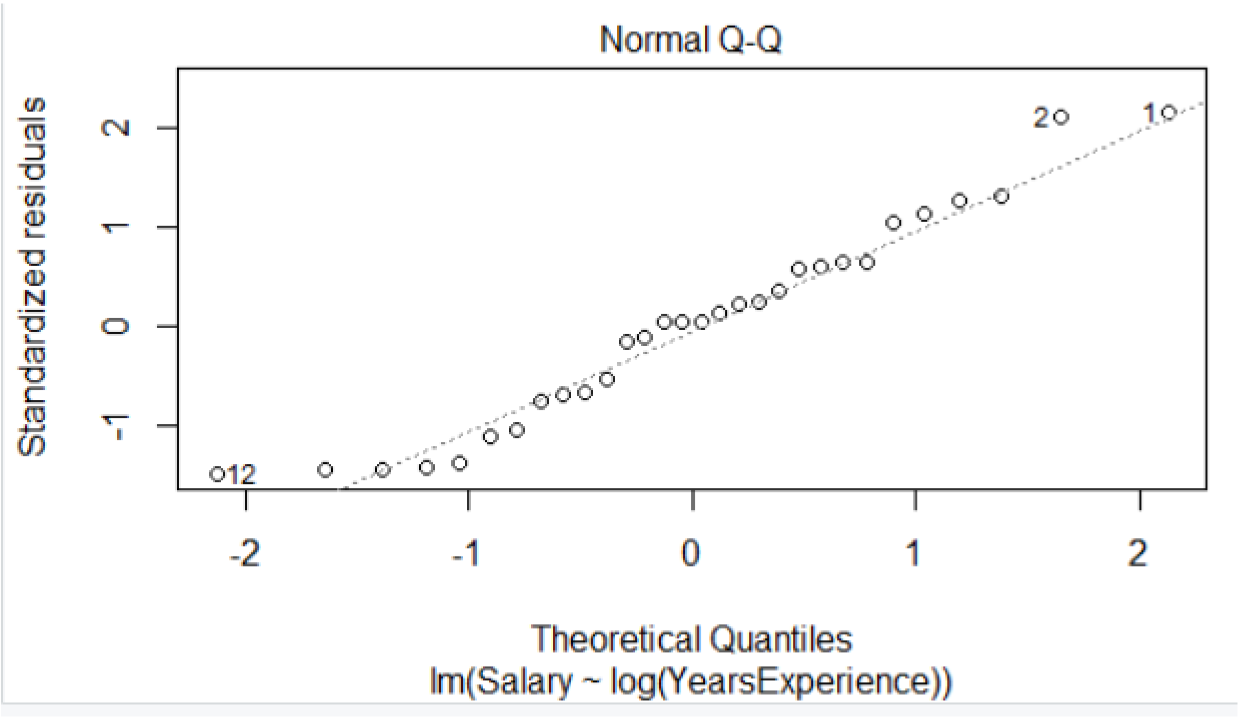

Scale-Location
$\sqrt{|\text{Standardized residuals}|}$
0.0
0.4
0.8
1.2
20
24
9
40000
60000
80000
100000
120000
Fitted values
lm(Salary ~ YearsExperience)

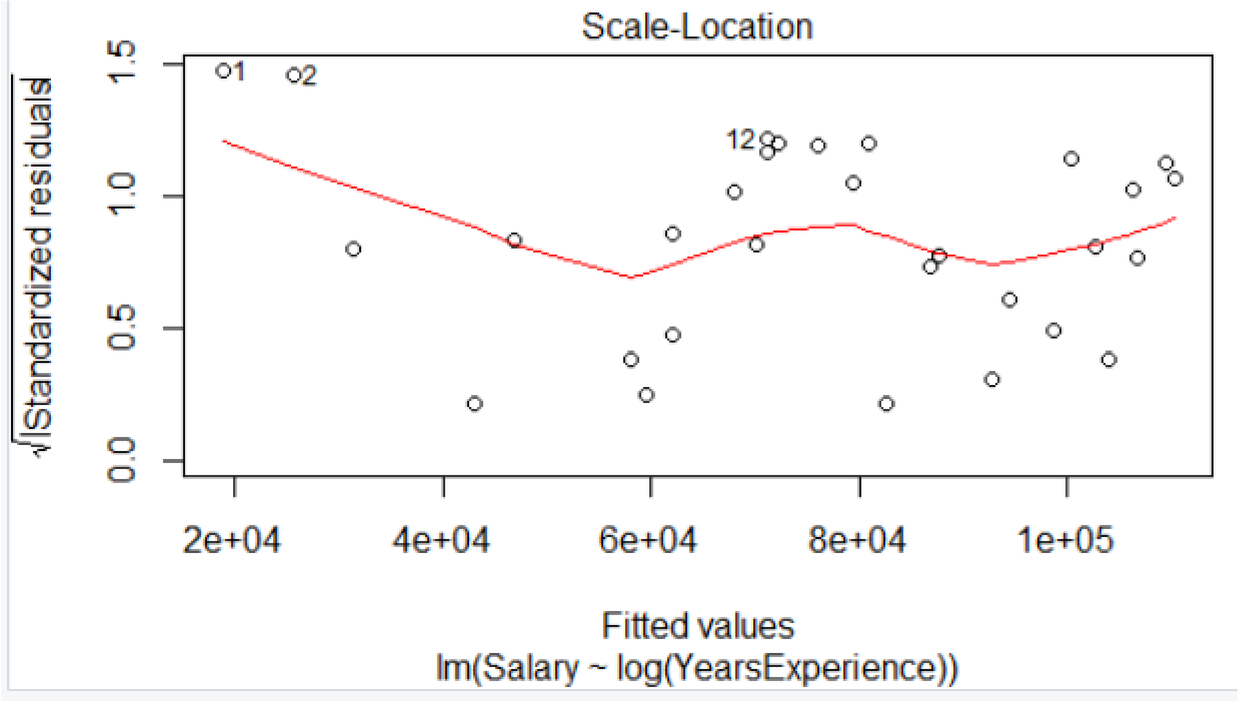


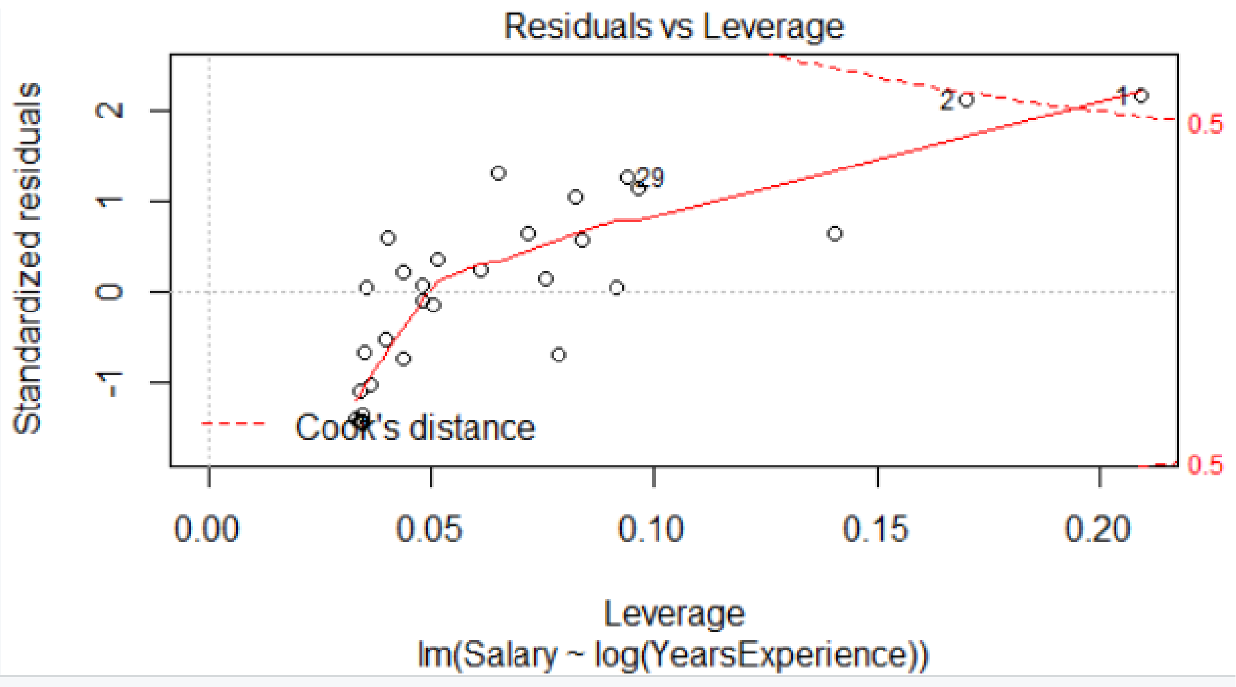
```
> Salary_hike_model1 <- lm(Salary~log(YearsExperience),data = Salary_
hike)
> summary(Salary_hike_model1)

Call:
lm(formula = Salary ~ log(YearsExperience), data = Salary_hike)

Residuals:
     Min       1Q   Median       3Q      Max 
-15392.6  -7523.0    559.7   6336.1  20629.8 

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)    
(Intercept)             14928       5156   2.895  0.00727 ** 
log(YearsExperience)    40582       3172  12.792 3.25e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10660 on 28 degrees of freedom
Multiple R-squared:  0.8539,	Adjusted R-squared:  0.8487 
F-statistic: 163.6 on 1 and 28 DF,  p-value: 3.25e-13

> confint(Salary_hike_model1,level = 0.95)
                         2.5 %   97.5 %
(Intercept)           4365.921 25490.02
log(YearsExperience) 34083.512 47080.46
> predict(Salary_hike_model1,interval = "predict")
        fit       lwr      upr
1  18795.85 -5225.823 42817.52
2  25575.24  1946.237 49204.23
```

```
3   31382.55  8054.979  54710.13
4   43057.26 20232.824  65881.70
5   46925.14 24235.859  69614.42
6   58136.05 35746.140  80525.96
7   59511.84 37149.017  81874.67
8   62130.94 39813.758  84448.13
9   62130.94 39813.758  84448.13
10  68022.72 45779.622  90265.82
11  70159.11 47933.039  92385.17
12  71186.55 48966.805  93406.30
13  71186.55 48966.805  93406.30
14  72188.63 49973.872  94403.38
15  75966.42 53760.064  98172.78
16  79422.30 57209.189 101635.40
17  81045.79 58824.757 103266.83
18  82606.83 60375.307 104838.35
19  86959.07 64683.513 109234.62
20  87641.13 65356.711 109925.56
21  92720.50 70353.368 115087.64
22  94472.51 72070.071 116874.96
23  98805.37 76300.817 121309.93
24 100317.92 77772.799 122863.04
25 102719.92 80105.200 125334.64
26 104095.71 81438.281 126753.14
27 106289.87 83560.068 129019.67
28 106714.81 83970.397 129459.23
29 109571.01 86723.331 132418.68
30 110351.45 87474.053 133228.86

> plot(Salary_hike_model1)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
```

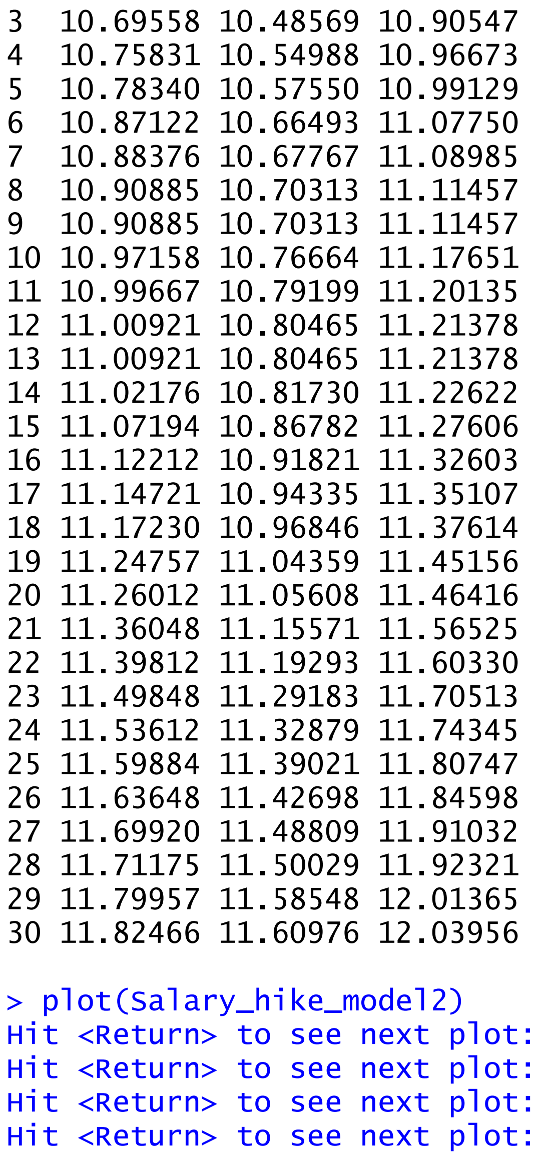



Normal Q-Q
Standardized residuals
-1
0
1
2
-2
-1
0
1
2
12
2
1
Theoretical Quantiles
lm(Salary ~ log(YearsExperience))

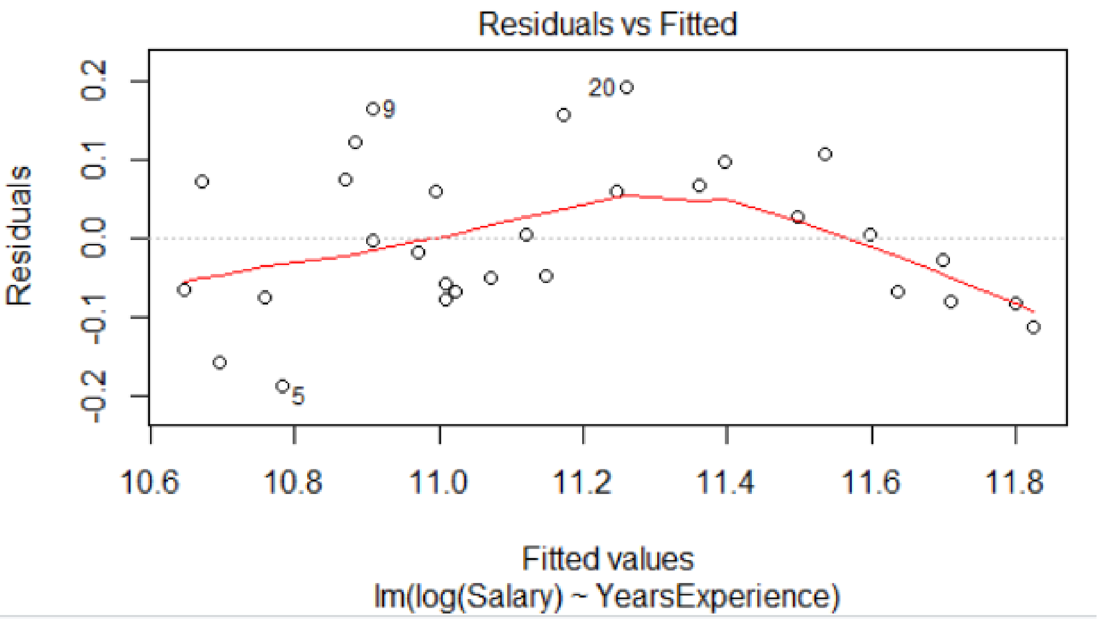

Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2e+04
4e+04
6e+04
8e+04
1e+05
1
2
12
Fitted values
lm(Salary ~ log(YearsExperience))

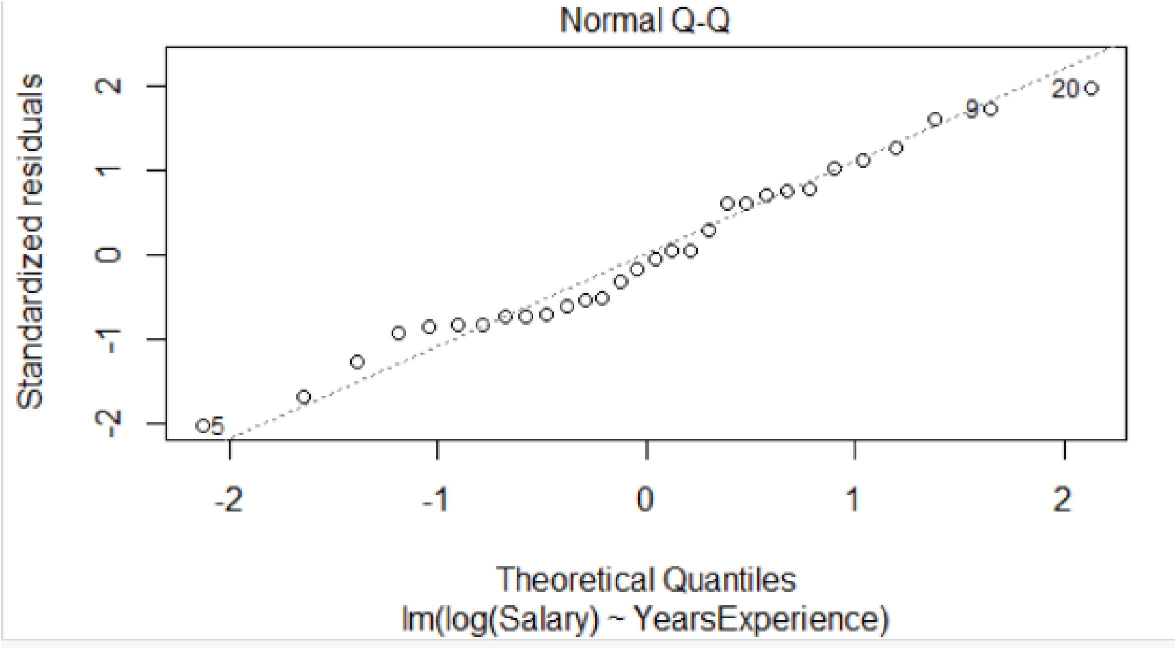


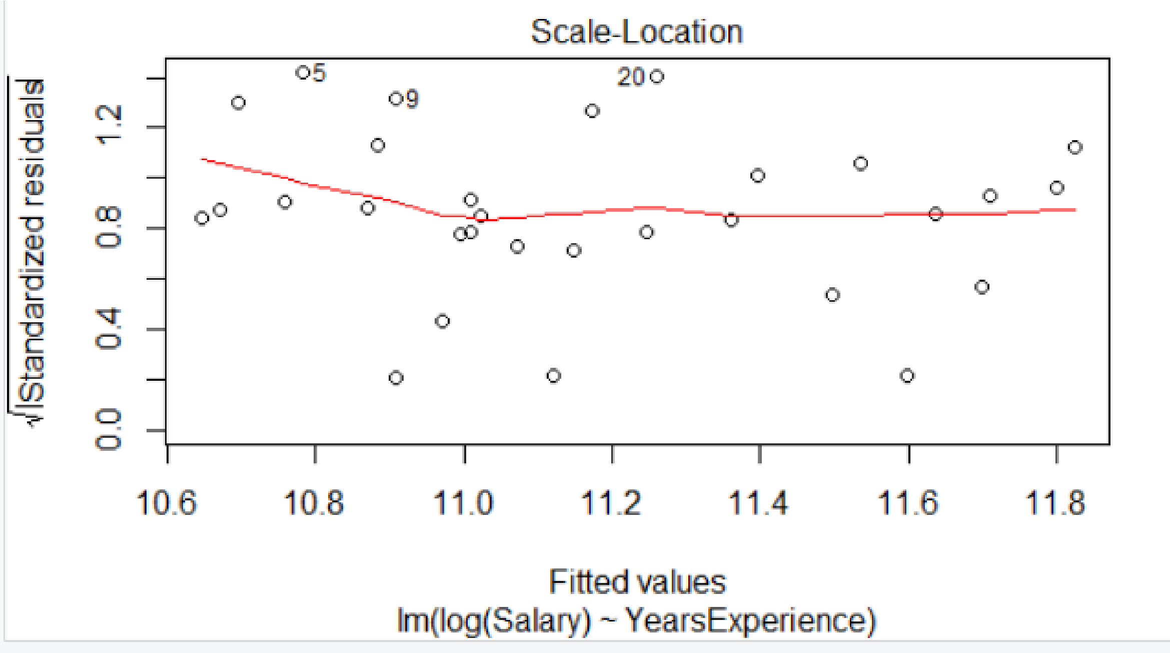
```
> Salary_hike_model2 <- lm(log(Salary)~YearsExperience,data = Salary_
hike)
> summary(Salary_hike_model2)

Call:
lm(formula = log(Salary) ~ YearsExperience, data = Salary_hike)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.18949 -0.06946 -0.01068  0.06932  0.19029 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)     10.507402   0.038443  273.33   <2e-16 ***
YearsExperience  0.125453   0.006406   19.59   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.09789 on 28 degrees of freedom
Multiple R-squared:  0.932, Adjusted R-squared:  0.9295 
F-statistic: 383.6 on 1 and 28 DF,  p-value: < 2.2e-16

> confint(Salary_hike_model2,level = 0.95)
                     2.5 %     97.5 %
(Intercept)     10.4286558 10.5861480
YearsExperience  0.1123316  0.1385742
> predict(Salary_hike_model2,interval = "predict")
        fit      lwr      upr
1  10.64540 10.43420 10.85661
2  10.67049 10.45996 10.88102
```

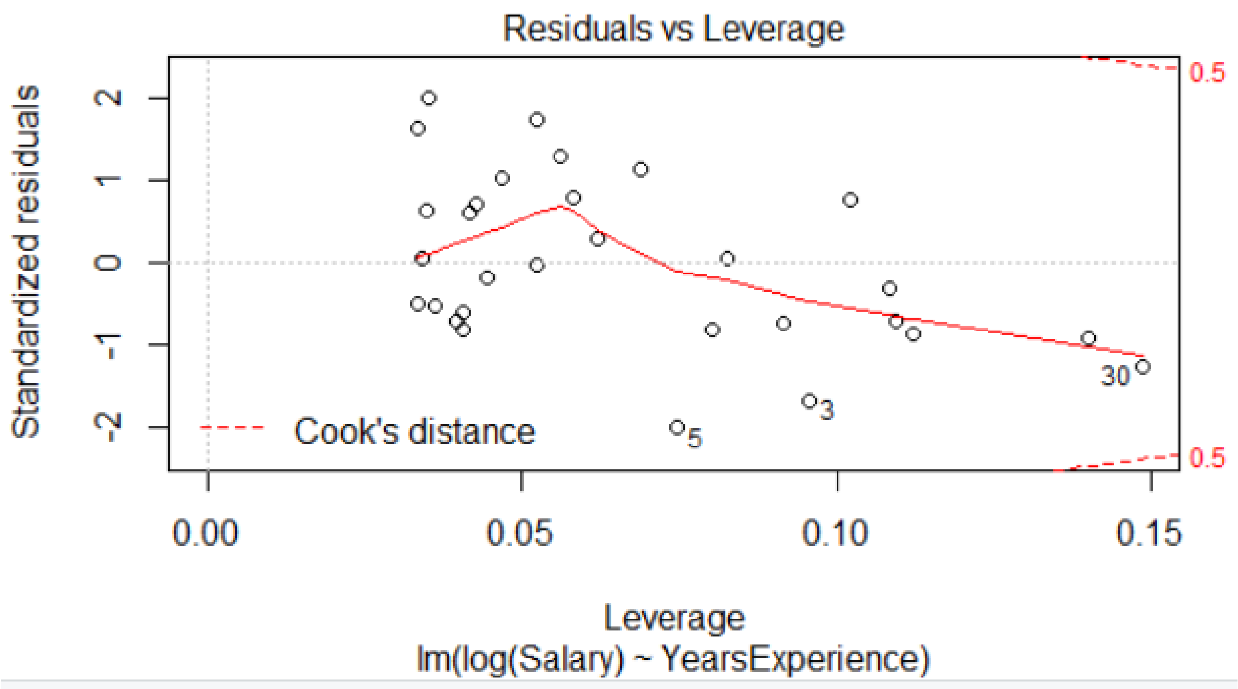


```
3  10.69558 10.48569 10.90547
4  10.75831 10.54988 10.96673
5  10.78340 10.57550 10.99129
6  10.87122 10.66493 11.07750
7  10.88376 10.67767 11.08985
8  10.90885 10.70313 11.11457
9  10.90885 10.70313 11.11457
10 10.97158 10.76664 11.17651
11 10.99667 10.79199 11.20135
12 11.00921 10.80465 11.21378
13 11.00921 10.80465 11.21378
14 11.02176 10.81730 11.22622
15 11.07194 10.86782 11.27606
16 11.12212 10.91821 11.32603
17 11.14721 10.94335 11.35107
18 11.17230 10.96846 11.37614
19 11.24757 11.04359 11.45156
20 11.26012 11.05608 11.46416
21 11.36048 11.15571 11.56525
22 11.39812 11.19293 11.60330
23 11.49848 11.29183 11.70513
24 11.53612 11.32879 11.74345
25 11.59884 11.39021 11.80747
26 11.63648 11.42698 11.84598
27 11.69920 11.48809 11.91032
28 11.71175 11.50029 11.92321
29 11.79957 11.58548 12.01365
30 11.82466 11.60976 12.03956

> plot(Salary_hike_model2)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
```




Normal Q-Q
Standardized residuals
-2
-1
0
1
2
-2
-1
0
1
2
5
9
20
Theoretical Quantiles
lm(log(Salary) ~ YearsExperience)


Scale-Location
√|Standardized residuals|
0.0
0.4
0.8
1.2
10.6
10.8
11.0
11.2
11.4
11.6
11.8
5
9
20
Fitted values
lm(log(Salary) ~ YearsExperience)



Conclusion- p-value is less than 0.05. and Multiple R- squared value is 0.932. This means the regression model will predict the output 93.2% time correct. Residual standard error: 0.09789 on 28 degrees of freedom