

R Codes for Chapter-3

Residual plots

Here “plot (X, Y, main=”TITLE OF THE PLOT”)” command is used with corresponding X and Y.

Ex. Plastic Hardness (continued....)

First residuals and fitted values should be extracted from the summary output of the fitted model.

```
> res=residuals(SLR)           #residuals
> res
      1      2      3      4      5      6      7      8      9     10
-2.150  3.850 -5.150 -1.150  0.575  2.575 -2.425  5.575  3.300  0.300
     11     12     13     14     15     16
  1.300 -3.700  0.025 -1.975  3.025 -3.975

> fitvals=fitted.values(SLR)   #fitted values
> fitvals
      1      2      3      4      5      6      7      8      9
201.150 201.150 201.150 201.150 217.425 217.425 217.425 217.425 233.700
     10     11     12     13     14     15     16
233.700 233.700 233.700 249.975 249.975 249.975 249.975
```

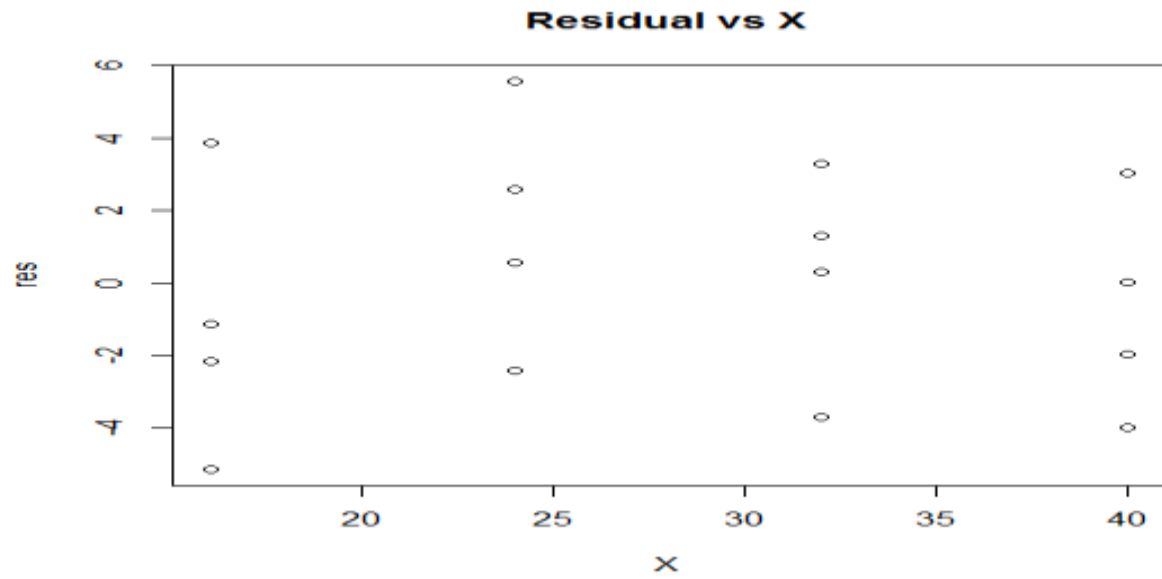
Then semistudentized residuals and absolute residuals can be calculated as follows.

```
> sstres=res/sigmahat          #Semistudentized residuals
> sstres
      1      2      3      4      5
-0.664805894  1.190466369 -1.592442026 -0.355593850  0.177796925
      6      7      8      9     10
  0.796221013 -0.749839206  1.723857145  1.020399745  0.092763613
     11     12     13     14     15
  0.401975657 -1.144084562  0.007730301 -0.610693787  0.935366433
     16
-1.229117875

> abres=abs(res)               #absolute residuals
> abres
      1      2      3      4      5      6      7      8      9     10     11     12
  2.150  3.850  5.150  1.150  0.575  2.575  2.425  5.575  3.300  0.300  1.300  3.700
     13     14     15     16
  0.025  1.975  3.025  3.975
```

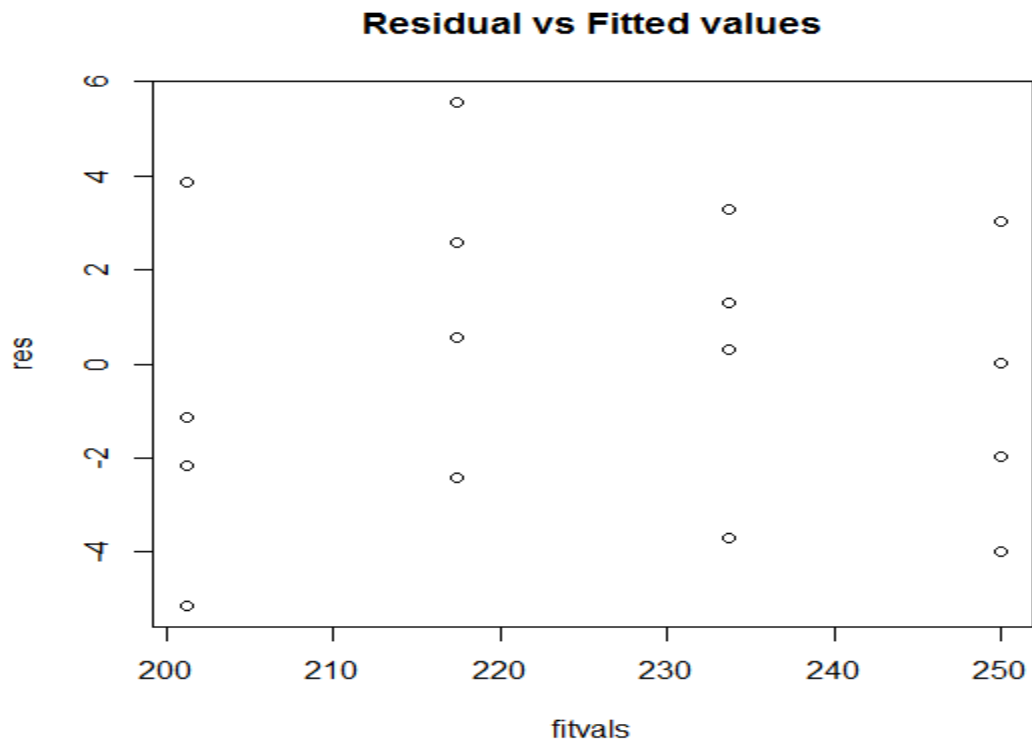
Plot of residuals Vs Predictoeor

```
> plot(X,res,main="Residual vs X") #plot of residuals vs X
```



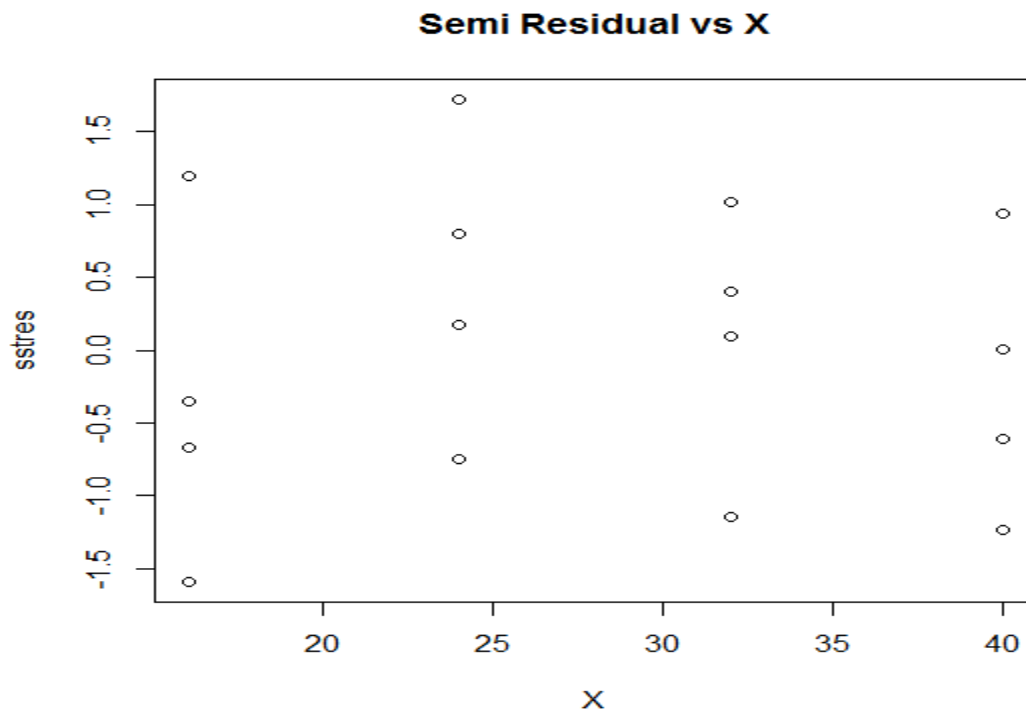
Plot of residuals Vs Fitted Values

```
> plot(fitvals,res,main="Residual vs Fitted values") #plot of residuals vs  
Fitted values
```



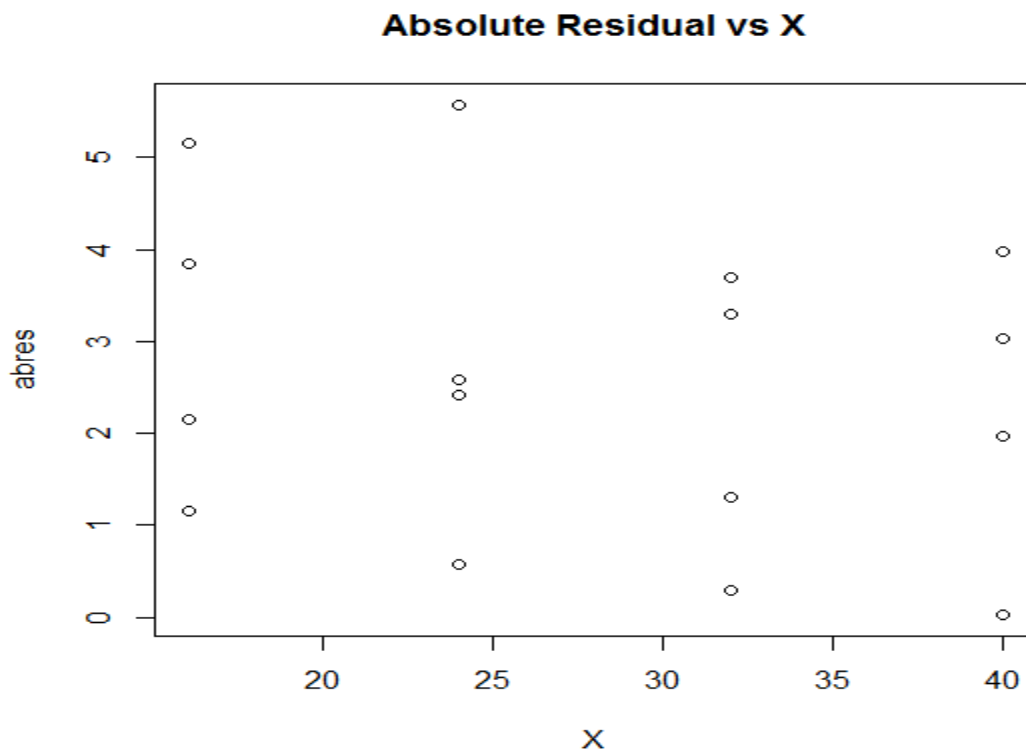
Plot of Semi-studentized residuals Vs Predictor

```
> plot(X,sstres,main="Semi Residual vs X")  =#plot of semi residuals vs X
```



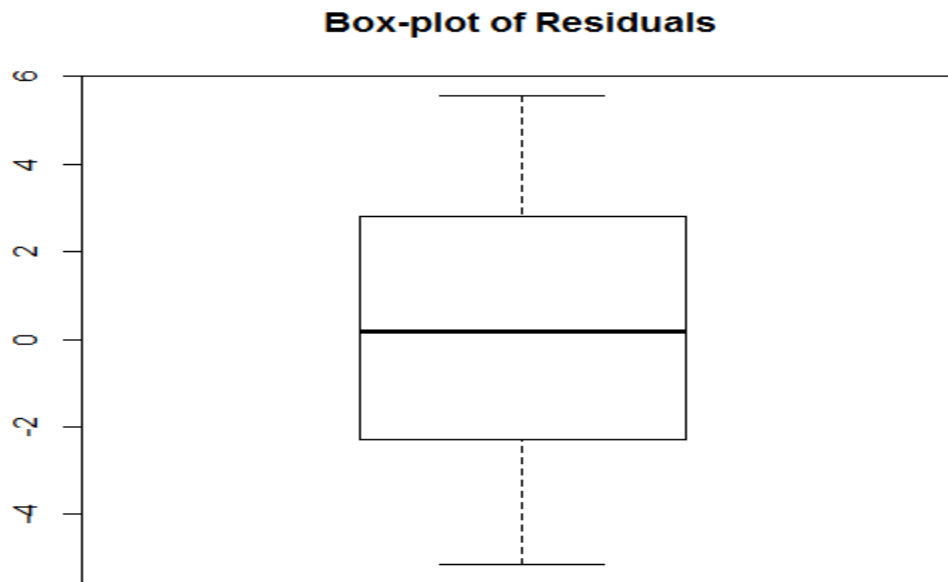
Plot of absolute residuals Vs Predictor

```
> plot(X,abres,main="Absolute Residual vs X") #plot of absolute residuals vs X
```



Box-Plot of Residuals

```
> boxplot(res,main="Box-plot of Residuals") #Box-plot of the residuals.
```



Normal Probability Plot

The following command can be used to assign ranks for residuals.

```
> rres=rank(res) #Ranking residuals
```

```
> rres
```

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
5 15  1  7 10 12  4 16 14  9 11  3  8  6 13  2
```

The expected residuals under normality can be obtained as follows

```
> smse=summary(SLR)$sigma #extracting mse.
```

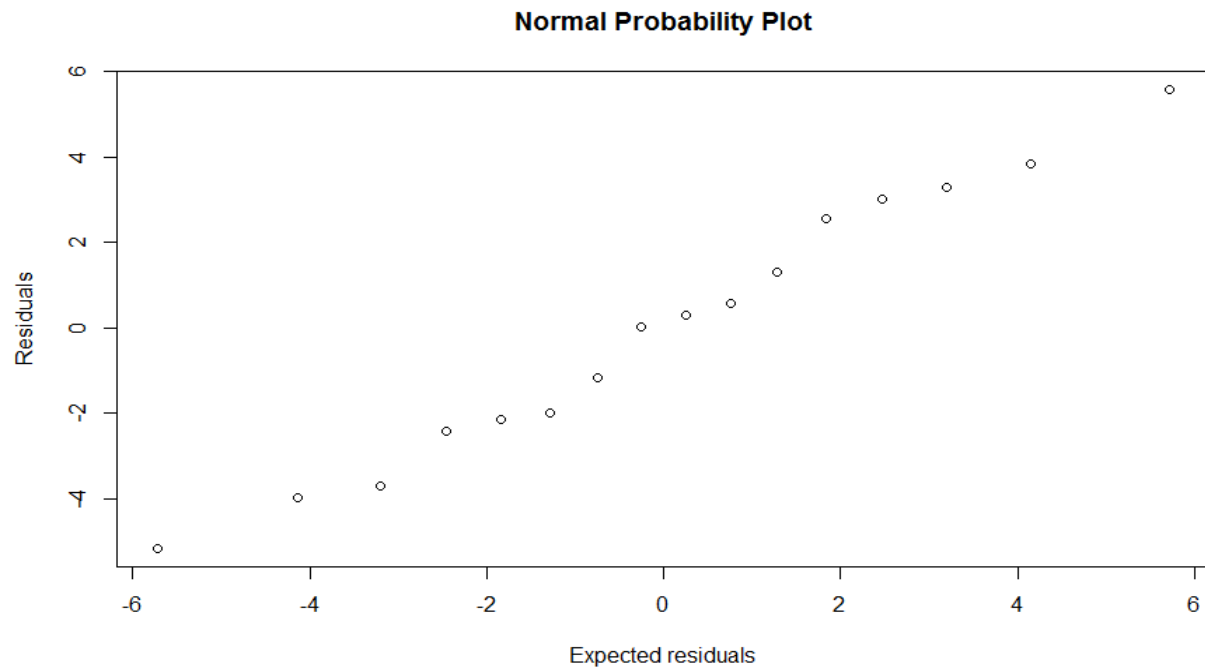
```
> eres=smse*(qnorm((rres-0.375)/(length(Time)+.25))) # Expected values of residuals.
```

```
> eres
```

```
      1      2      3      4      5      6      7      8
-1.8407587  4.1445720 -5.7204274 -0.7550976  0.7550976  1.8407587 -2.4638199  5.7204274
      9     10     11     12     13     14     15     16
 3.1957184  0.2496787  1.2797862 -3.1957184 -0.2496787 -1.2797862  2.4638199 -4.1445720
```

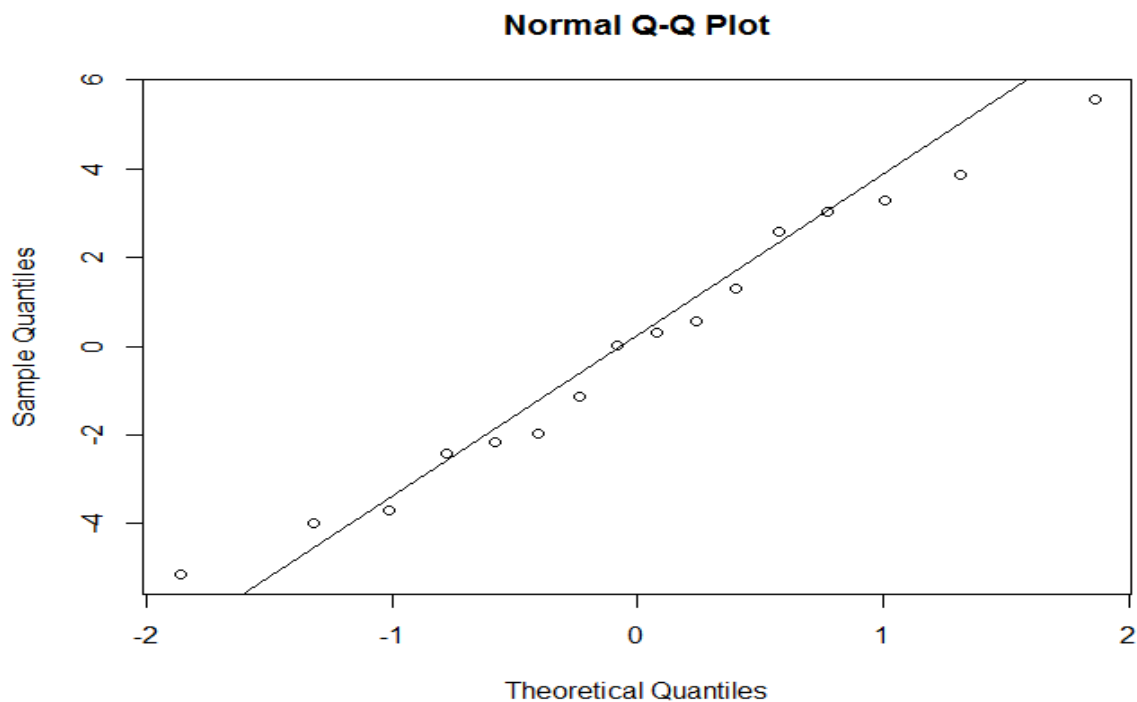
So the normal probability plot is

```
> plot(res,eres,main="Normal Probability Plot", xlab="Expected residuals",ylab
="Residuals")
```



An alternative method to check the normality assumption is using the **qqnorm(residuals)**. This draws the quantiles of the residuals vs the theoretical normal distribution quantiles. If points are around a line then error terms are normal and if points are away from the line normality assumption violates.

```
> qqnorm(res) #plot pf the Theoretical quantiles and the sample quantiles
> qqline(res) #adding a line to the plot
```



Test For Lack of fit

1. Correlation Test

The correlation coefficient between residuals and their expected residuals under normality can be calculated as follows.

```
> TLF=cor(eres,rres) #Here use ranked residuals.  
> TLF  
[1] 0.9899802
```

For this example $n=16$ and the corresponding value in Table B.6 is 0.941 (used 0.05 as the level of significance). So the observed value = 0.98998 > table value = 0.941. So the normal assumption is not violated significantly.

2. -test for Lack of Fit (General Linear Test)

Here we compare the Simple Linear Regression Model (reduced model) with the means modal (i.e., Analysis of variance model : the model without a predictor) (Full Model). To fit the means model, the command “aov(Y~factor(X))” can be used. Here factor(X) consider only the levels of X. To generate the ANOVA table, “anova(reduced model, full model)” is used.

Ex. Plastic Hardness (continued....)

```
> Means=aov(Hardness~factor(Time)) #Full model # factor(X) consider only  
                                the levels of X.  
> SLR=lm(Hardness~Time) #Reduced model (Simple Linear Regression model)  
> anova(SLR,Means)  
Analysis of Variance Table
```

```
Model 1: Hardness ~ Time  
Model 2: Hardness ~ factor(Time)  
  Res.Df    RSS Df Sum of Sq    F Pr(>F)  
1      14 146.43  
2      12 128.75  2    17.675 0.8237 0.4622
```

Here note that here we have all the values for the two additional rows of the general ANOVA table. So SSPE= 128.75, df(SSPE)= 12, SSLF=17.675, df(SSLF)=2, Test statistic: F= 0.8237, and the p-value= 0.4622.

Here since the p-value is greater than 0.05, the null hypotheses is not rejected. So the association between the elapsed time and the plastic hardness is approximately linear.

To generate the complete general anova table, the main anova table and the above result should be combined.

```
> anova(SLR)  
Analysis of Variance Table  
  
Response: Hardness  
          Df Sum Sq Mean Sq F value    Pr(>F)  
Time       1 5297.5   5297.5   506.51 2.159e-12 ***  
Residuals 14  146.4     10.5  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

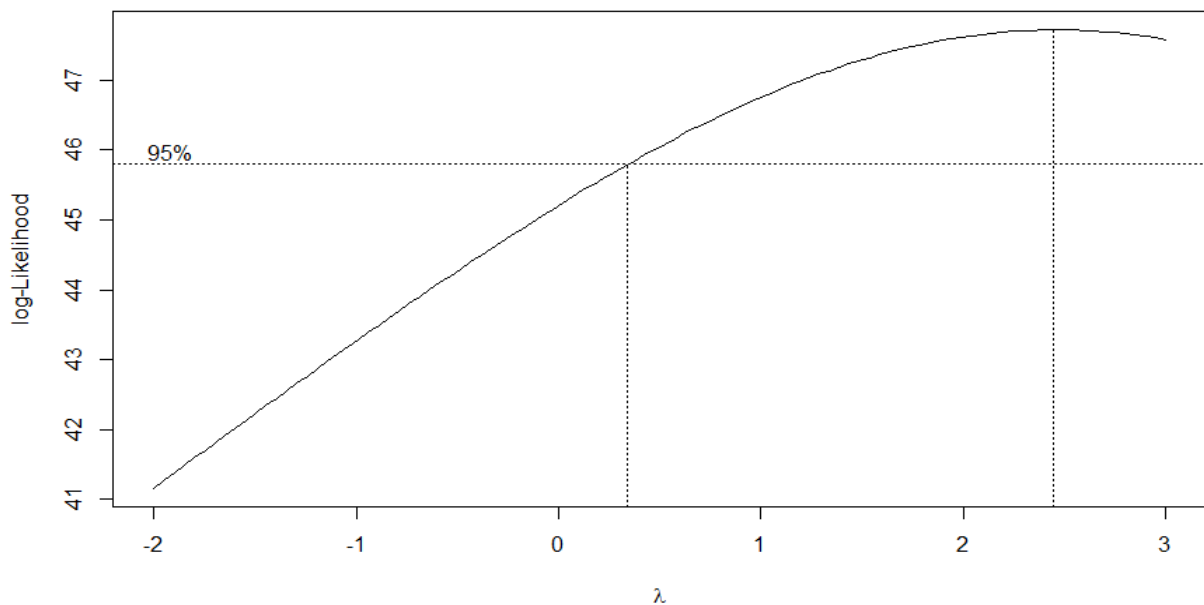
Therefore, the general ANOVA table is,

Source	SS	DF	MS	F	P-value
Time	5297.5	1	5297.5	506.51	2.159e-12
Error	146.4	14	10.5		
Pure Error	128.75	12		0.8237	0.4622
Lack of fit	17.675	2			

Remedial Measures (BOX-COX transformation)

The Likelihood function for several different values λ can be drawn using the command “boxcox”.

```
> BC=boxcox(Hardness~Time, lambda = seq(-2, 3, 1/10), xlab = expression(lambda), ylab = "log-Likelihood")
```



Then to find the value of λ which maximized the likelihood function, the following method can be used.

```
> lambda=BC$x[which.max(BC$y)]
> lambda
[1] 2.444444
```

So the maximum likelihood estimator of λ is 2.4444.