5303hw4

Jin Yao

2019/10/2

5303 hw3

E6.2

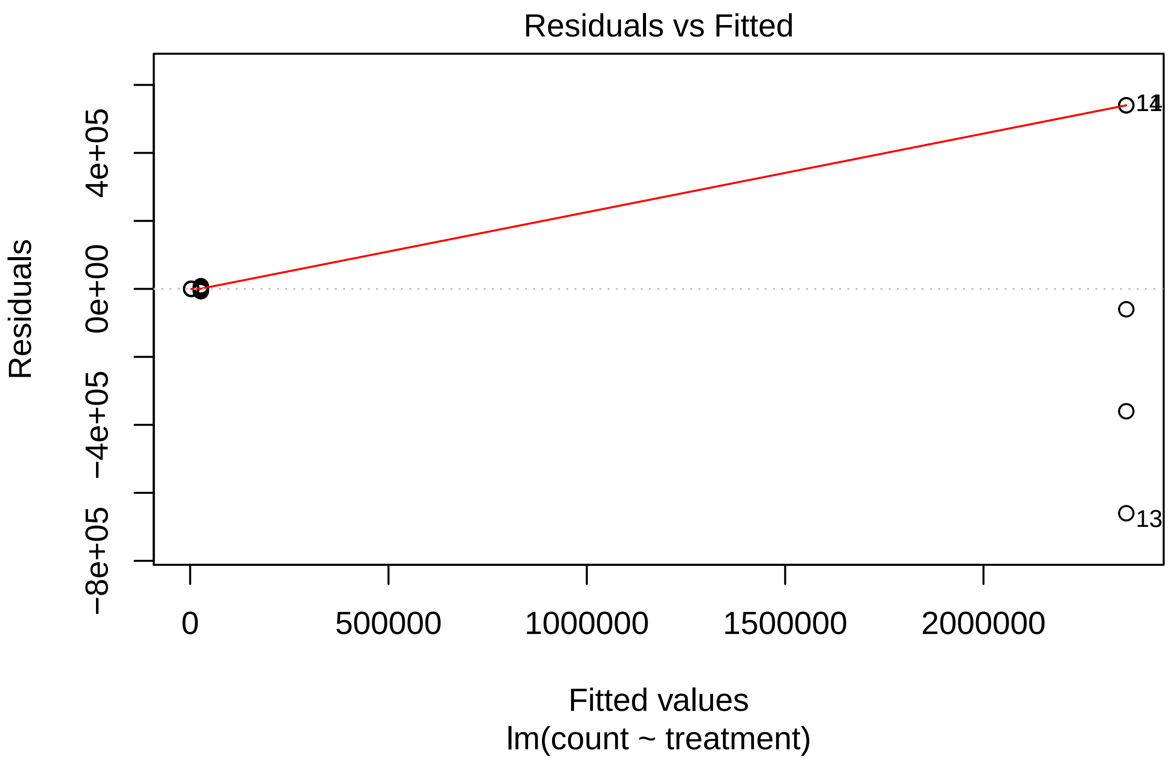
```
library(cfcdae)
data("Pasteurization")
head(Pasteurization)
```

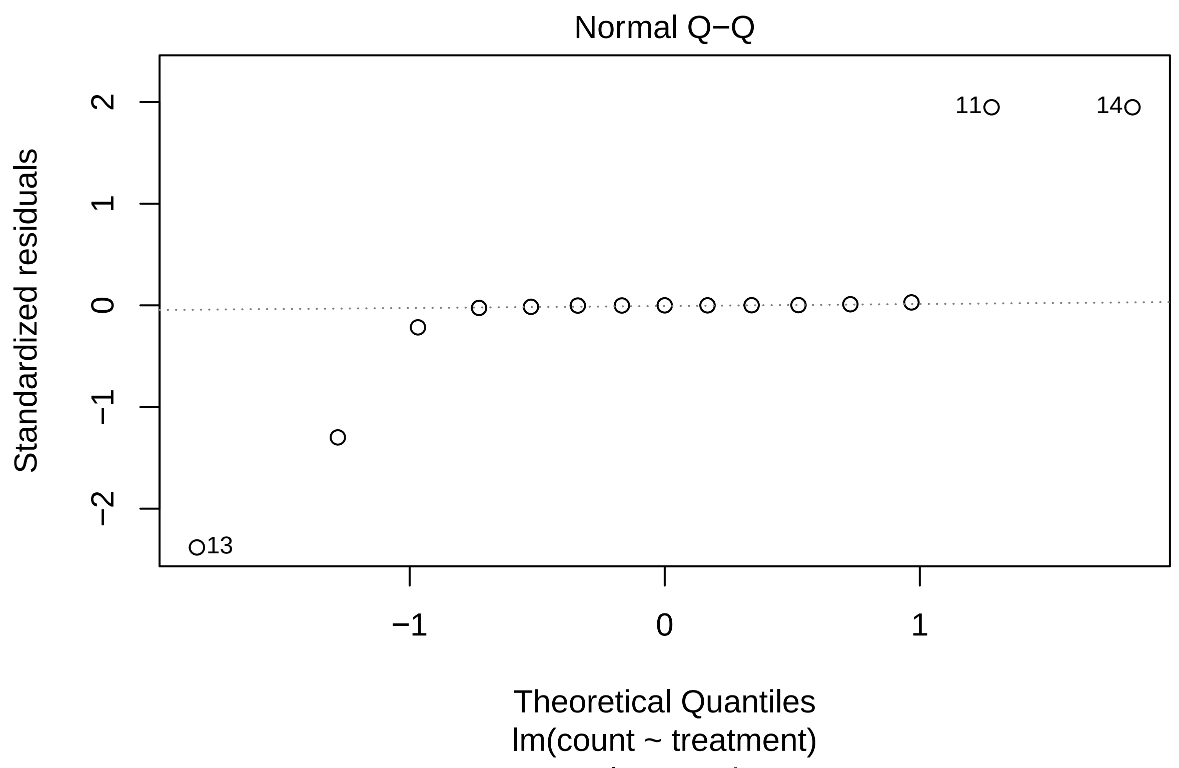
```
##   treatment count
## 1         1  2600
## 2         1  2900
## 3         1  2000
## 4         1  2200
## 5         1  3200
## 6         2 35000
```

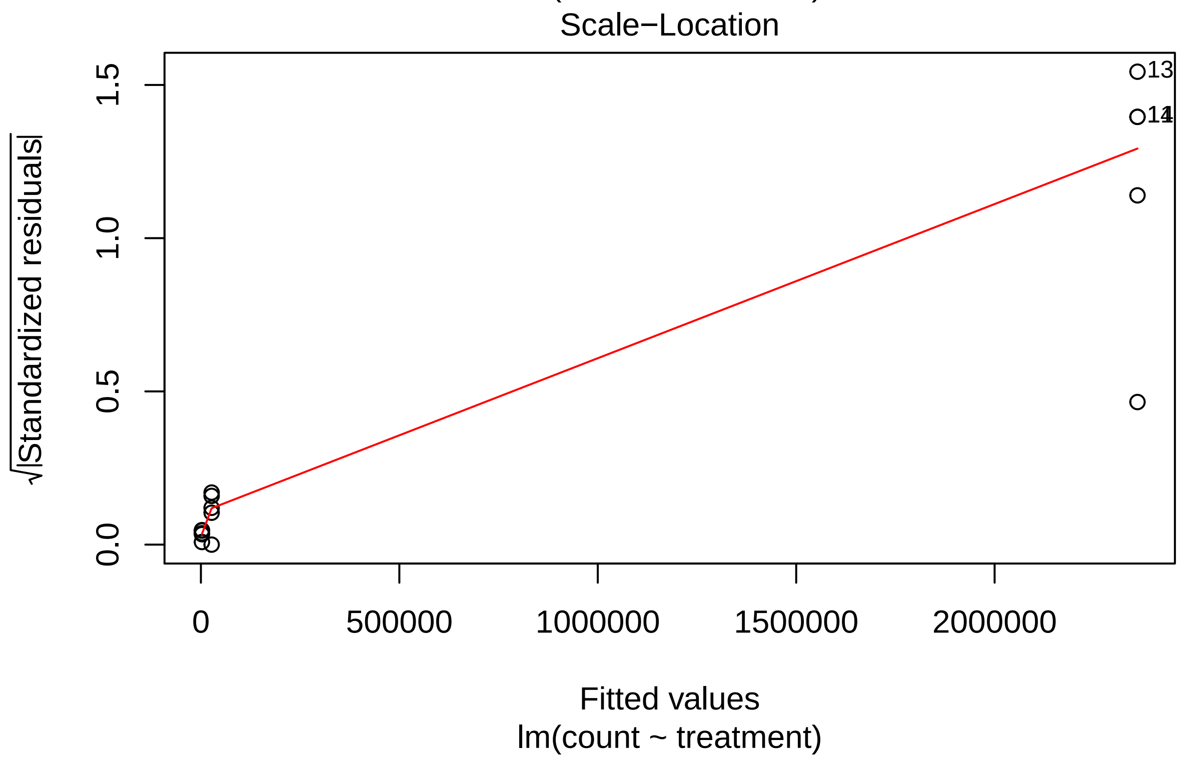
```
past <- lm(count~treatment, data = Pasteurization)
library(car)
```

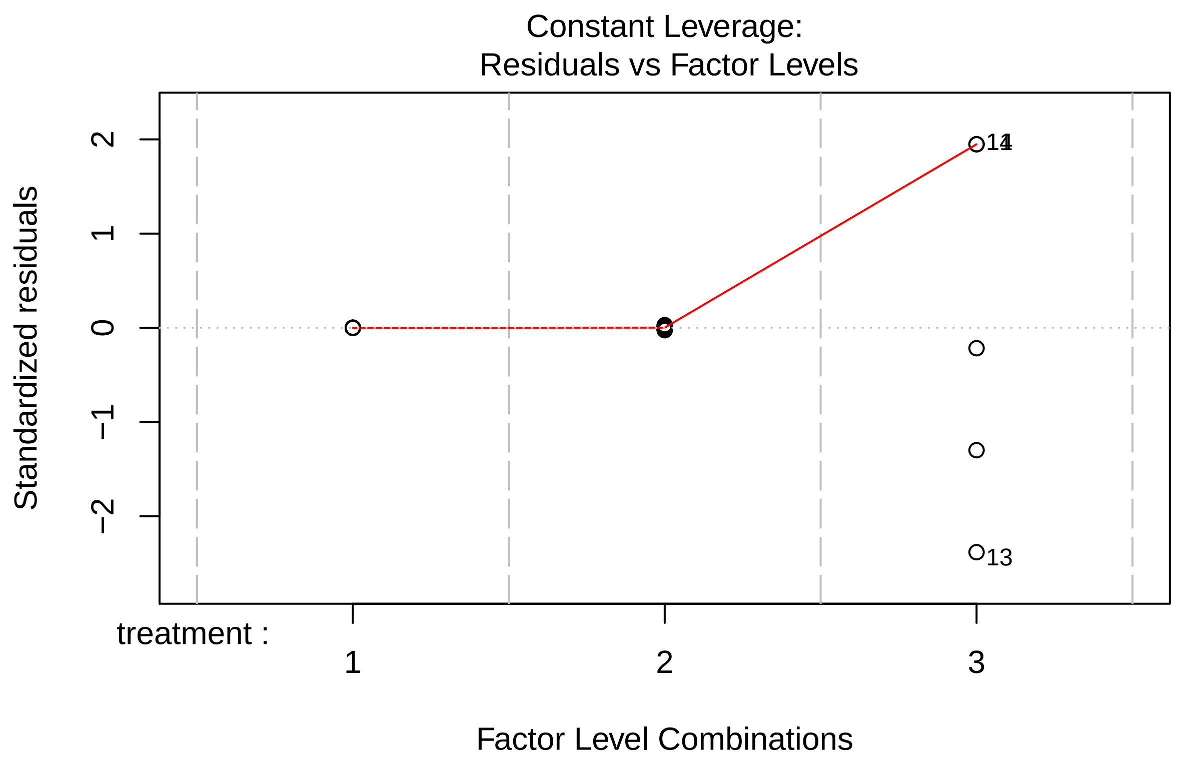
```
## Loading required package: carData
```

```
plot(past)
```

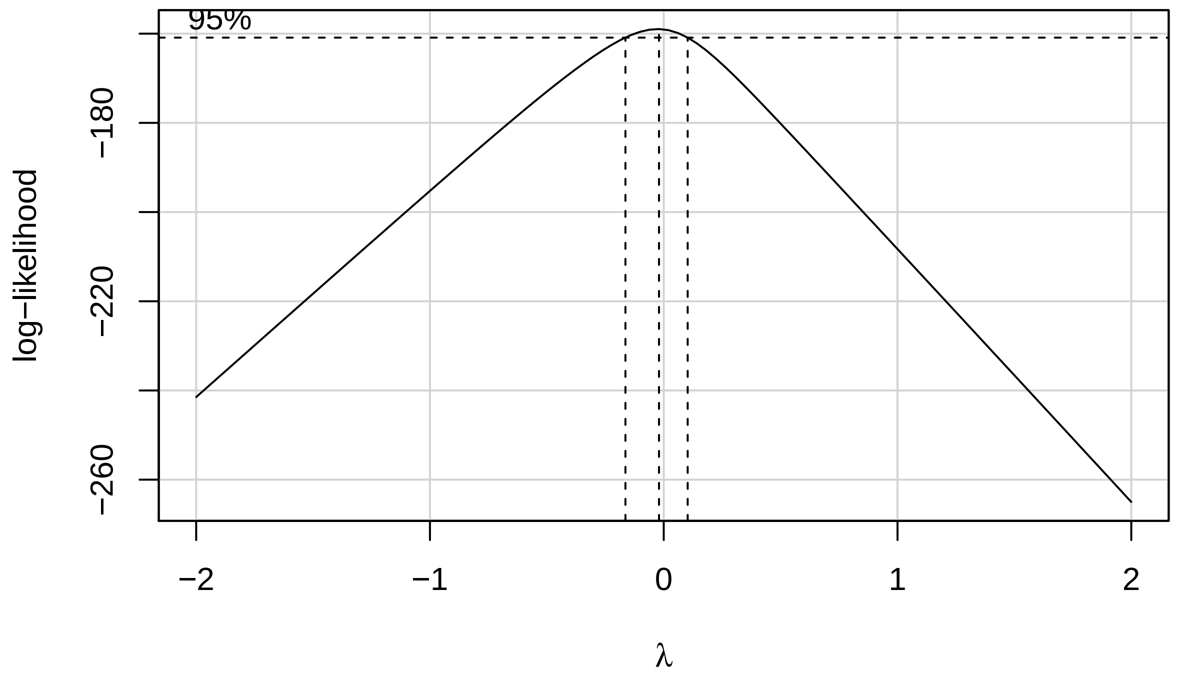



Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(count ~ treatment)
11
14
13
−2
−1
0
1
2
−1
0
1


Scale−Location
√|Standardized residuals|
Fitted values
lm(count ~ treatment)
13
14
11
0.0
0.5
1.0
1.5
0
500000
1000000
1500000
2000000



```r
# The residual plot shows a very strange pattern, check boxcox
boxCox(past)
```



```r
# Then we apply the log transformation
pastlog <- lm(log(count)~treatment, data = Pasteurization)
anova(pastlog)
```

```
## Analysis of Variance Table
##
```

```
## Response: log(count)
##           Df Sum Sq Mean Sq F value    Pr(>F)
## treatment  2 119.77  59.885  1283.4 1.015e-14 ***
## Residuals 12   0.56   0.047
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

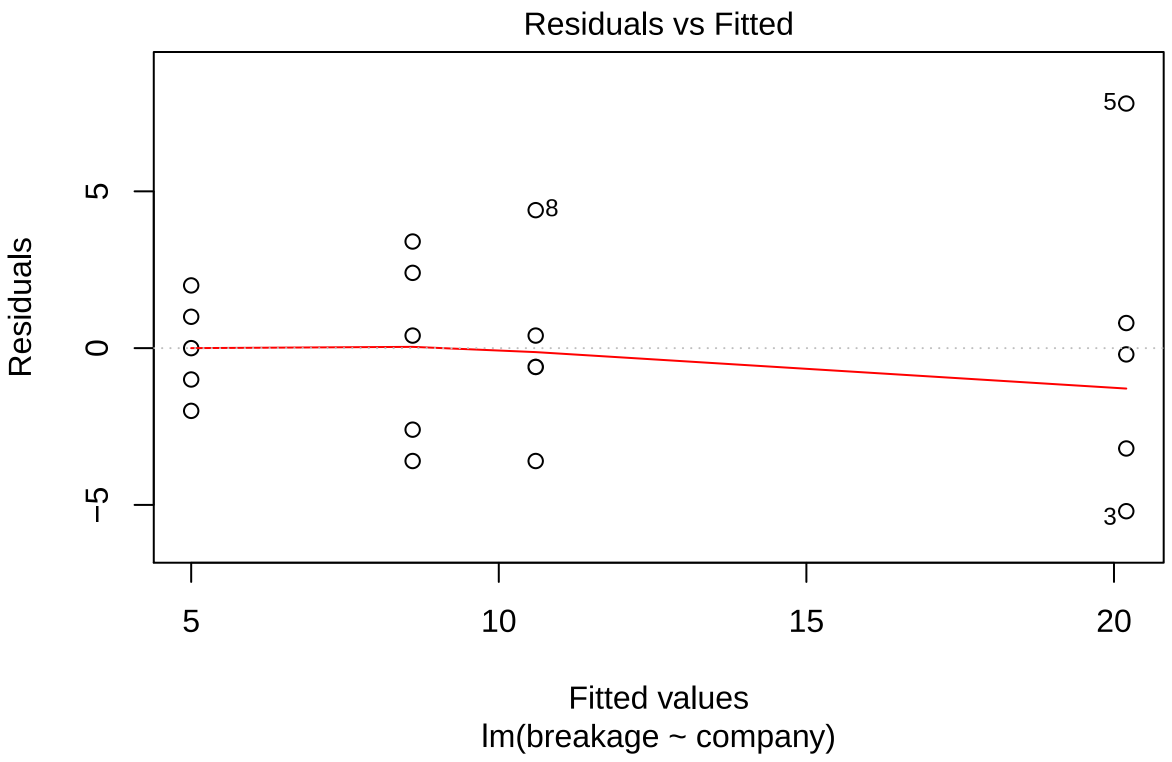
```
# Since the P value is so small, so we need to reject the null hypothesis that all treatments have the
```

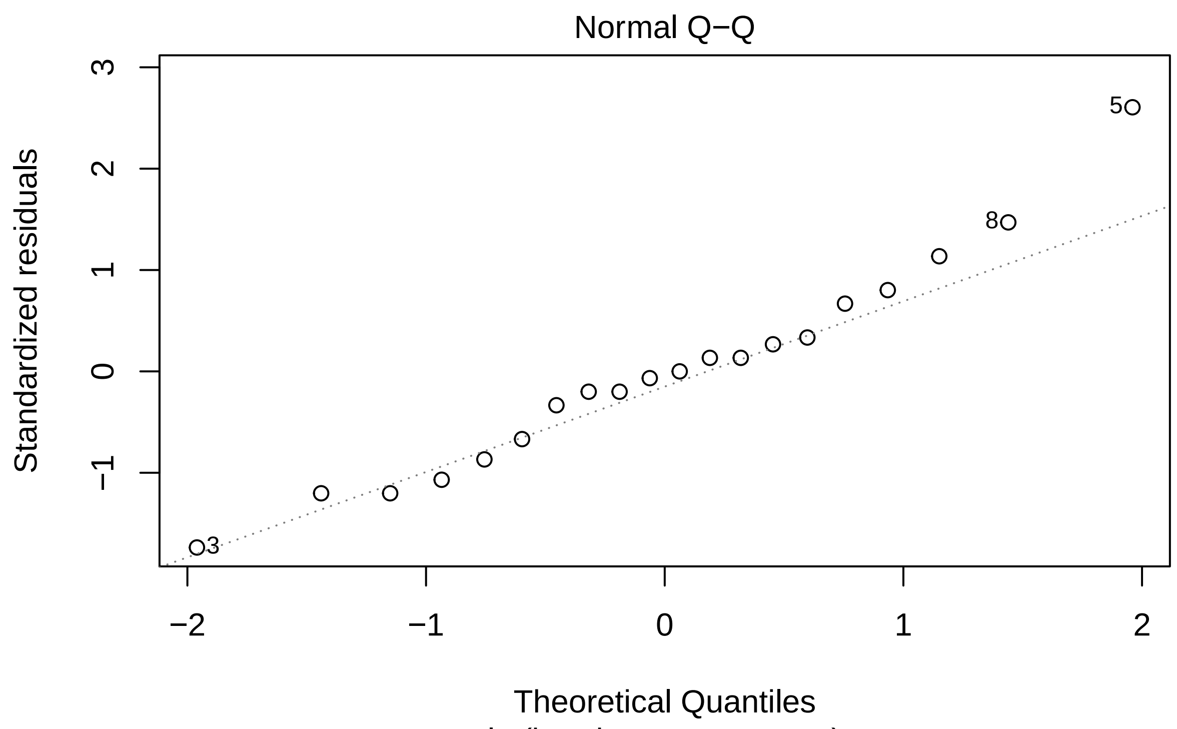
E6.4

```
data("Breakage")
head(Breakage)
```

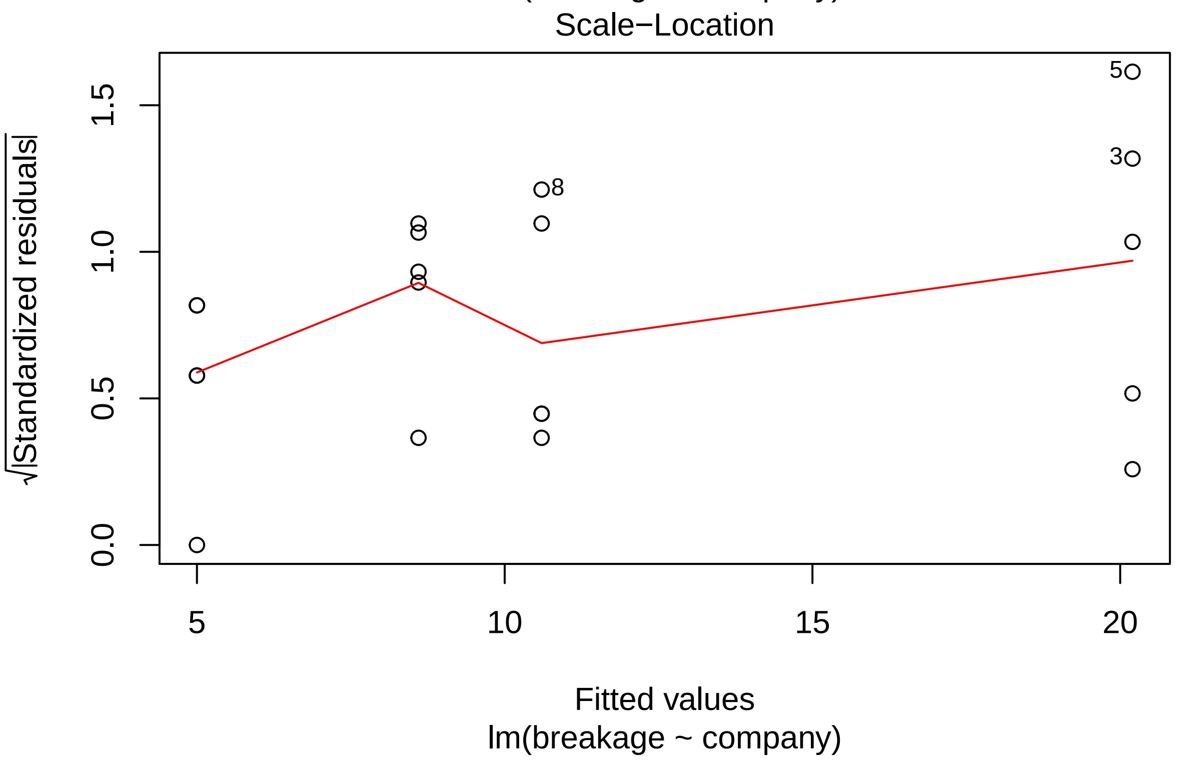
```
##   company breakage
## 1       A       17
## 2       A       20
## 3       A       15
## 4       A       21
## 5       A       28
## 6       B        7
```

```
library(car)
car <- lm(breakage ~ company, data = Breakage)
plot(car)
```

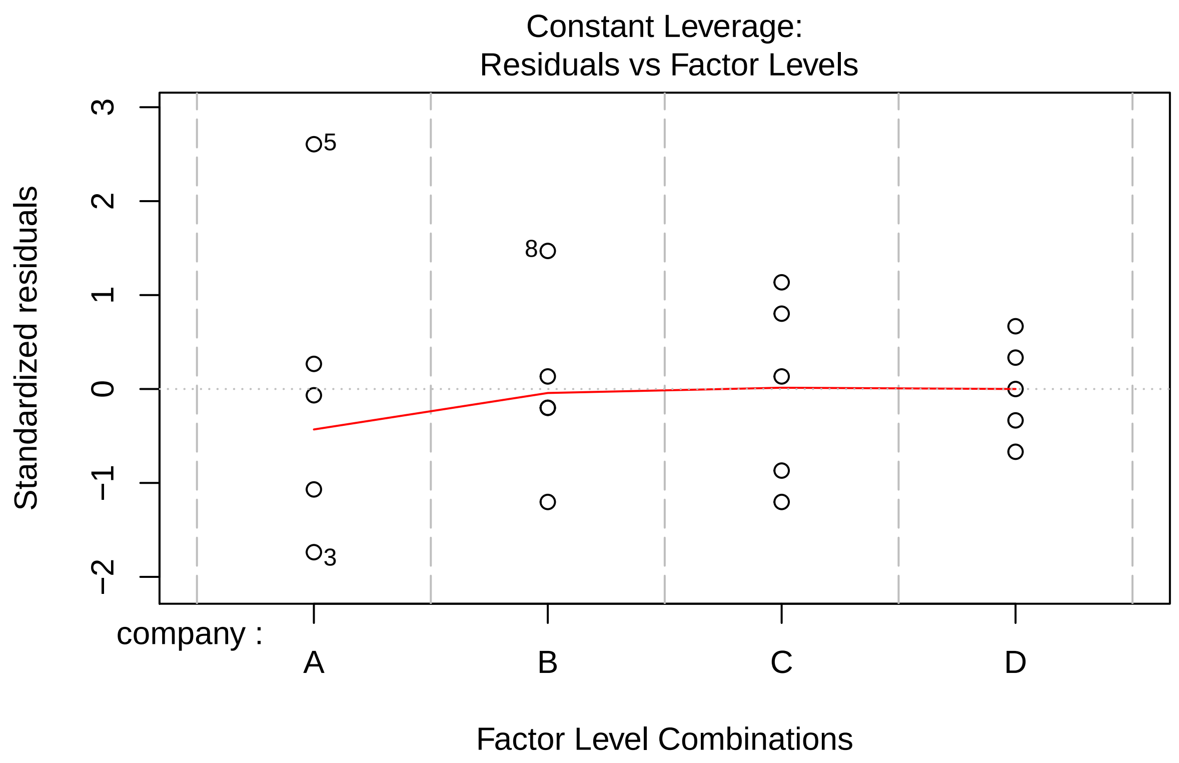




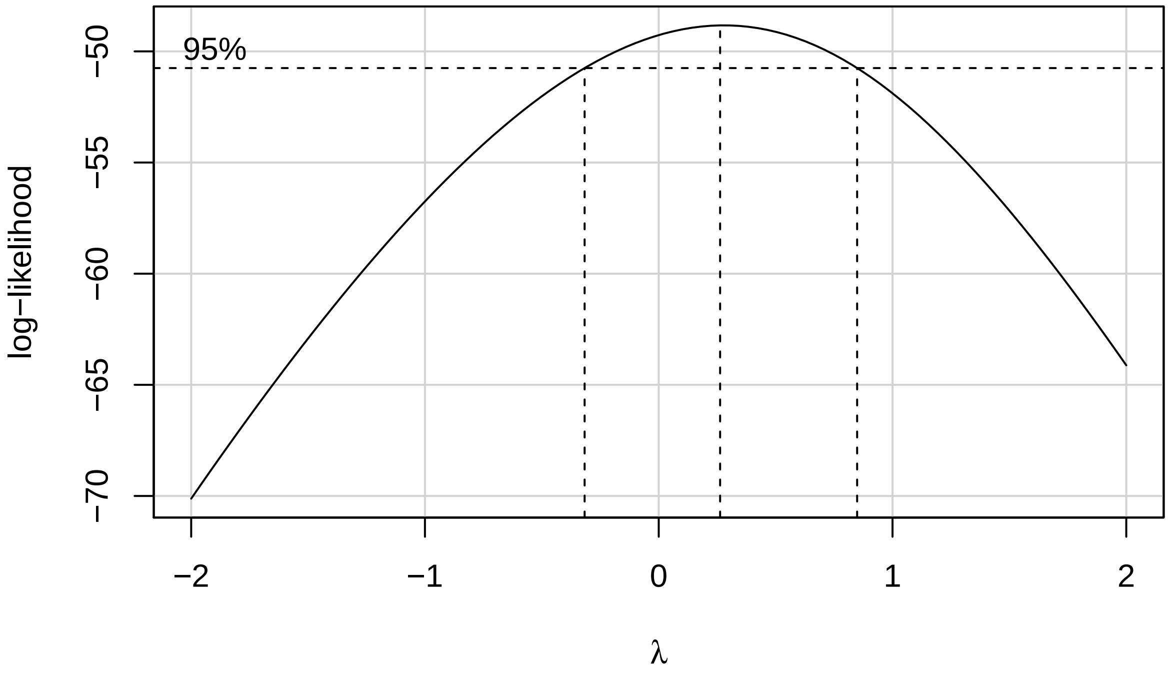
Normal Q−Q
Standardized residuals
3
2
1
0
−1
5
8
3
−2
−1
0
1
2
Theoretical Quantiles
lm(breakage ~ company)



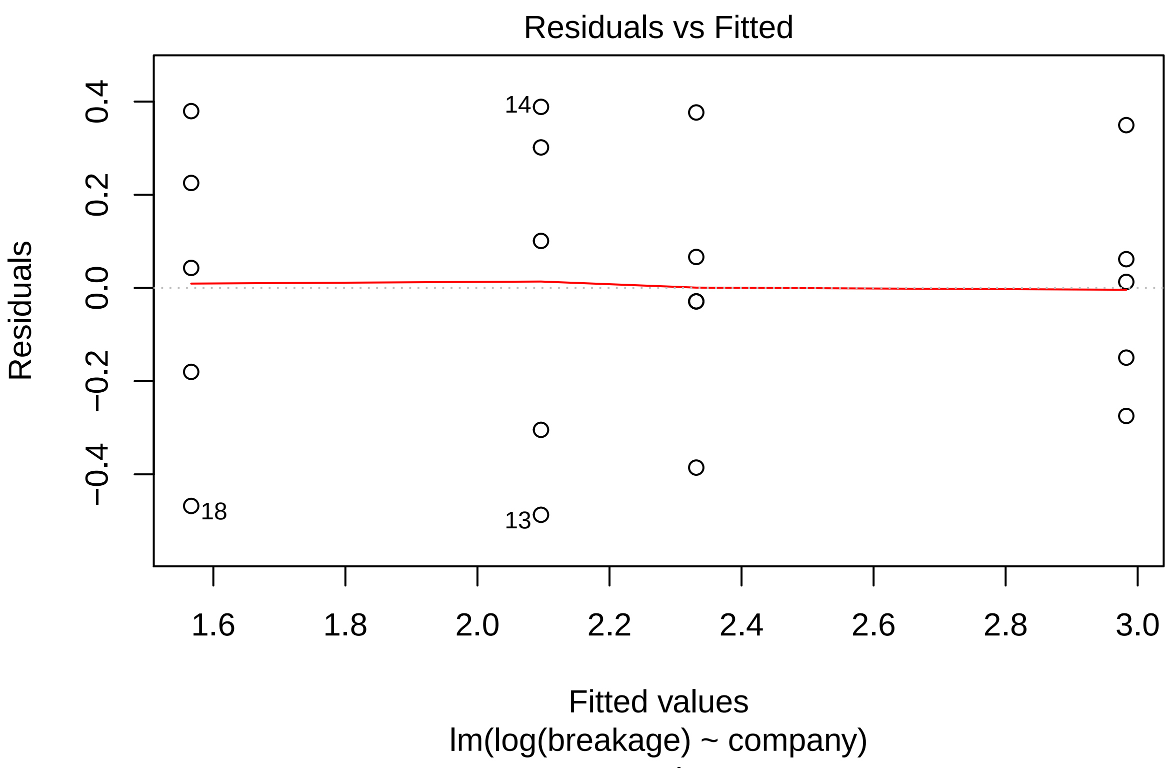
Scale−Location
√|Standardized residuals|
1.5
1.0
0.5
0.0
5
3
8
5
10
15
20
Fitted values
lm(breakage ~ company)



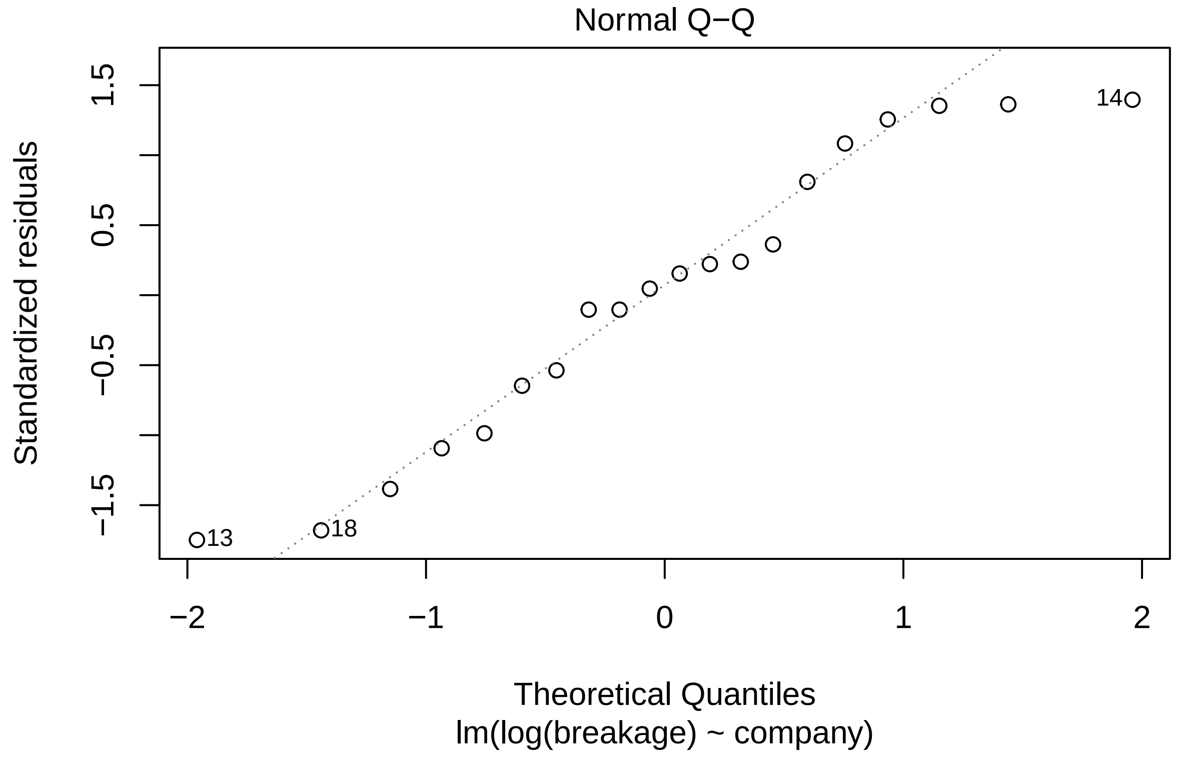
```
# I can see that the variance is non-constant
boxCox(car)
```



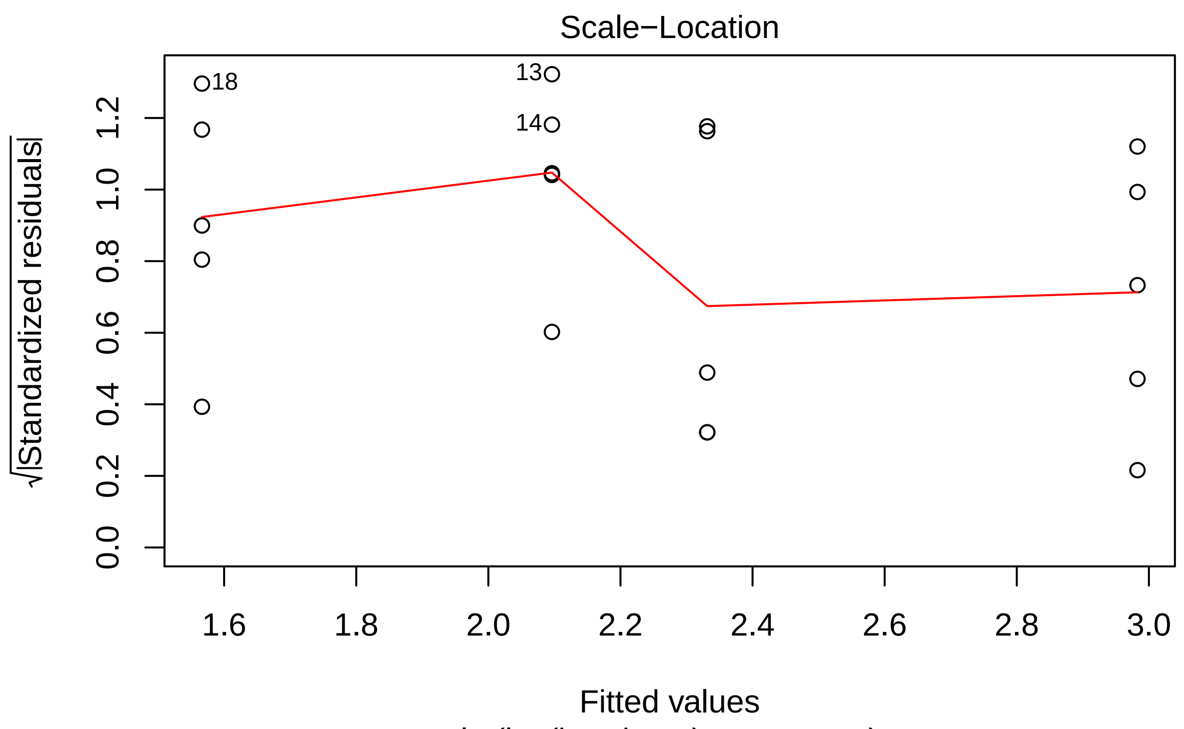
```
# we need log transformation
carlog <- lm(log(breakage) ~ company, data = Breakage)
plot(carlog)
```



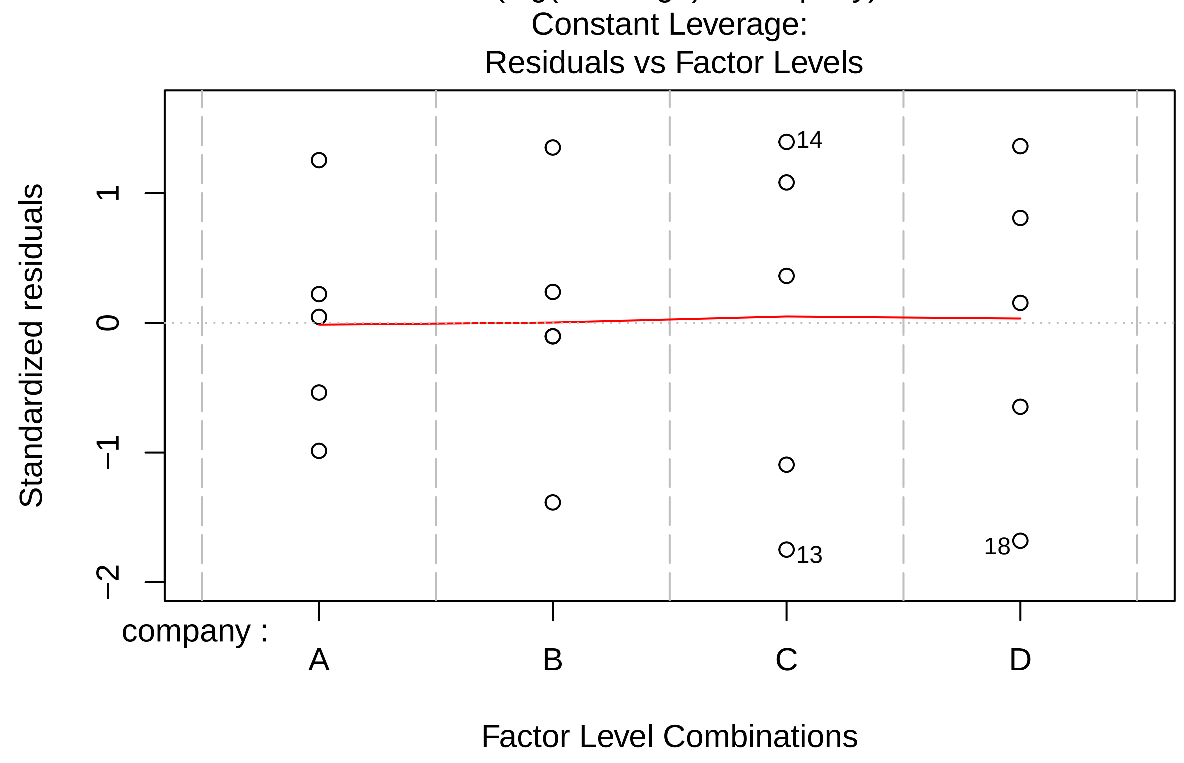
Residuals vs Fitted
Residuals
Fitted values
lm(log(breakage) ~ company)
14
18
13



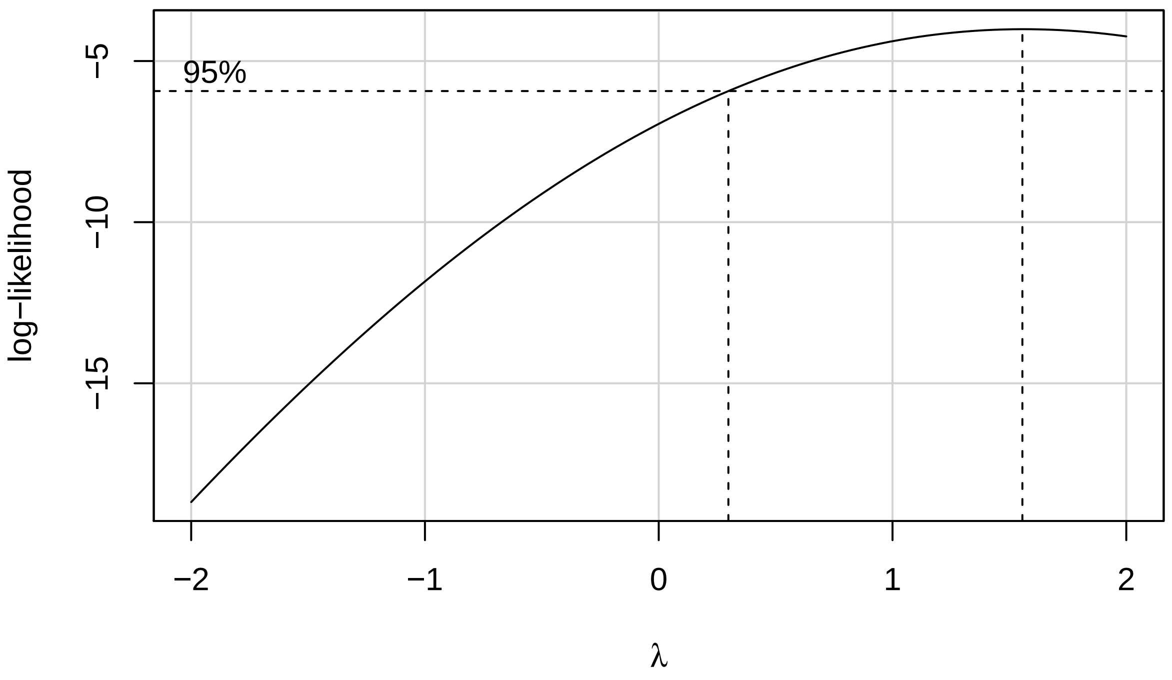
Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(log(breakage) ~ company)
14
18
13



lm(log(breakage) ~ company)



```r
# It looks better
boxCox(carlog)
```



```
# It gives us a confidence interval shown below
```

E6.5

(a) the first seems that it violates the independence, because as time goes by, there is a pattern that shows a serial correlation.
(b) it violates the non-constant variance, the pattern is a right-opening megaphone, with the increase of the fitted value, the residuals are more scattered.
(c) it is like a normal plot and it's good I think.
(d) it violates the normality, it looks like a long tail distribution.

P6.1

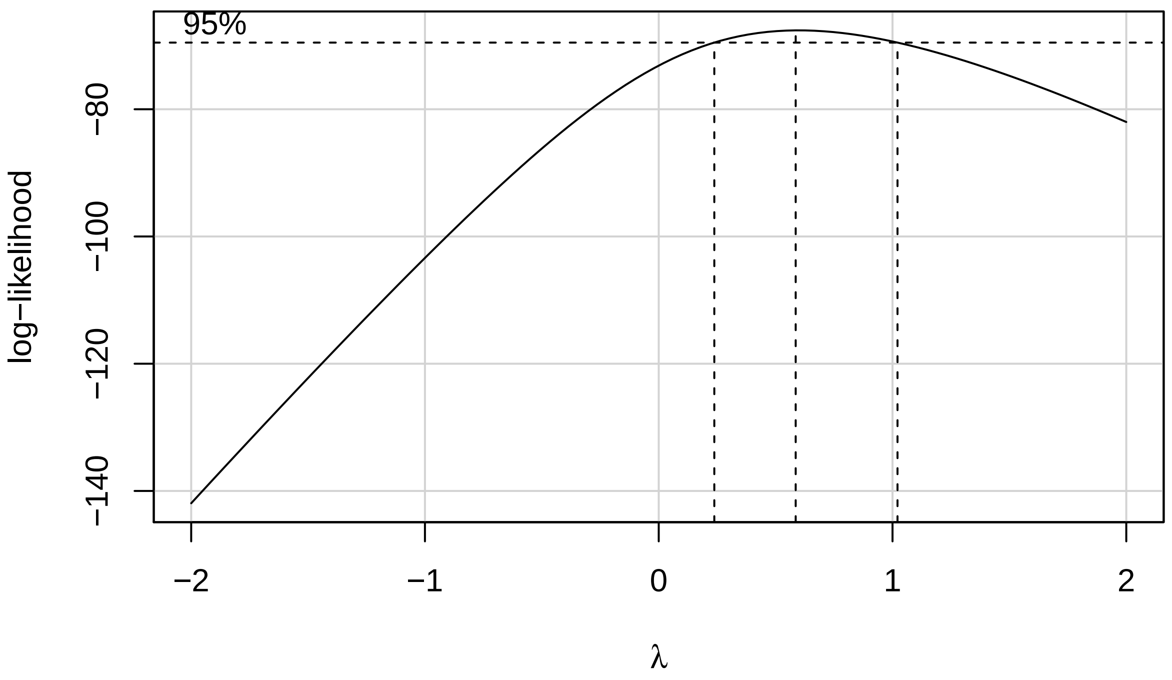
(a)

```
colApply <- function(dat, cols = colnames(dat), func = as.factor) {
  dat[cols] <- lapply(dat[cols], func)
  return(dat)
}
res1 <- c(100-97, 100-96, 100-92, 100-95)
res2 <- c(100-83, 100-87, 100-78, 100-81)
res3 <- c(100-85, 100-84, 100-78, 100-79)
res4 <- c(100-64, 100-72, 100-63, 100-74)
res5 <- c(100-52, 100-56, 100-44, 100-50)
res6 <- c(100-48, 100-58, 100-49, 100-53)
res <- c(res1,res2,res3,res4,res5,res6)
treat <- c(rep("treat1",4),rep("treat2",4),rep("treat3",4),rep("treat4",4),rep("treat5",4),
           rep("treat6",4))
data <- data.frame(treat,res)
```

```r
str(colApply(data, 'treat'))
```

```
## 'data.frame':    24 obs. of  2 variables:
##  $ treat: Factor w/ 6 levels "treat1","treat2",..: 1 1 1 1 2 2 2 2 3 3 ...
##  $ res  : num  3 4 8 5 17 13 22 19 15 16 ...
```

```r
x <- lm(res~treat)
library(car)
boxCox(x)
```



```r
# So from above, I can use the power to the square root to tast the large range between the ji8
# largest and the smallest.
```

(b)

```r
xlog <- lm(log(res)~treat)
anova(x)
```

```
## Analysis of Variance Table
##
## Response: res
##           Df Sum Sq Mean Sq F value    Pr(>F)
## treat      5 6398.3 1279.67  71.203 3.197e-11 ***
## Residuals 18  323.5   17.97
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(xlog)
```

```
## Analysis of Variance Table
##
## Response: log(res)
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## treat      5 15.4594 3.09188  60.124 1.337e-10 ***
## Residuals 18  0.9256 0.05142
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# the p values are very small, which tells that we should reject the null hypothesis and they are not e
#(c)
res1 <- c(97, 96, 92, 95)
res2 <- c(83, 87, 78, 81)
res3 <- c(85, 84, 78, 79)
res4 <- c(64, 72, 63, 74)
res5 <- c(52, 56, 44, 50)
res6 <- c(48, 58, 49, 53)
res <- c(res1,res2,res3,res4,res5,res6)
treat <- c(rep("treat1",4),rep("treat2",4),rep("treat3",4),rep("treat4",4),rep("treat5",4),
           rep("treat6",4))
data1 <- data.frame(treat,res)
x_qua <- lm(res~treat,data = data1)
linear.contrast(x_qua,treat,c(1,0,-1, -1, 1,0))
```

```
##   estimates       se   t-value   p-value  lower-ci upper-ci
## 1     -4.25 4.239366 -1.002508 0.3293854 -13.15658 4.656577
```

```r
# From the outcome, we can find that the p value is very large, so we fail to reject the null hypothesi
#(d)
linear.contrast(x_qua,treat,c(-0.5,0.5,0, 0, -0.5, 0.5))
```

```
##   estimates       se   t-value    p-value  lower-ci  upper-ci
## 1    -5.625 2.119683 -2.653699 0.01616203 -10.07829 -1.171711
```

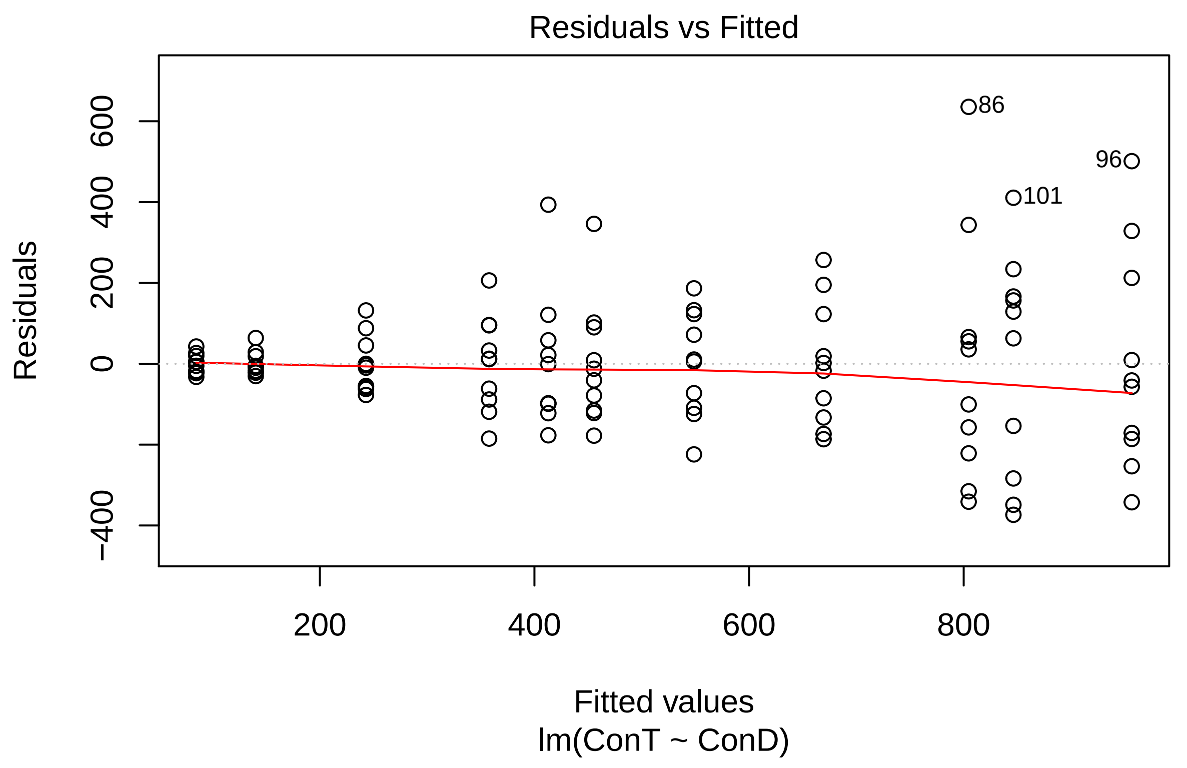
```r
# From the outcome, we can find that the p value is tiny, so we should reject the null hypothesis.
#(e)
# From the data, we can find that the fraction of blue is highest when the nitrogen is low with no
# irrgation.
xlog <- lm(log(res)~treat,data = data)
compare.to.best(xlog, treat, lowisbest = TRUE, confidence = 0.95)
```

```
##                    difference allowance
## * treat5 - treat1   2.354745   0.38589
## * treat6 - treat1   2.324275   0.38589
## * treat4 - treat1   1.902738   0.38589
## * treat3 - treat1   1.360604   0.38589
## * treat2 - treat1   1.314965   0.38589
## best is treat1      0.000000        NA
```

```r
# Only one of the treatment is in the group.
```

P6.3

```r
data("CookieTexture")
ConT <- CookieTexture$texture[CookieTexture$recipe=='Control']
ConD <- CookieTexture$day[CookieTexture$recipe=='Control']
data <- data.frame(ConT,ConD)
mod63 <- lm(ConT~ConD, data = CookieTexture)
plot(mod63, which = 1)
```



```
# try gls
library(nlme)
mod63glm <- gls(ConT~0+ConD, data = data,cor = corAR1())
summary(mod63glm)
```

```
## Generalized least squares fit by REML
##   Model: ConT ~ 0 + ConD
##   Data: data
##        AIC      BIC    logLik
##    1358.17 1391.907 -666.0851
##
## Correlation Structure: AR(1)
##  Formula: ~1
##  Parameter estimate(s):
##        Phi
## 0.06654711
##
## Coefficients:
##             Value Std.Error   t-value p-value
## ConD1   85.1406  60.24010  1.413354  0.1607
## ConD2  139.5306  60.22672  2.316755  0.0226
## ConD3  245.5865  60.22672  4.077699  0.0001
## ConD4  355.2540  60.22672  5.898611  0.0000
## ConD5  413.6163  60.22672  6.867655  0.0000
## ConD6  456.2342  60.22672  7.575278  0.0000
## ConD7  546.2012  60.22672  9.069084  0.0000
## ConD8  669.0195  60.22672 11.108349  0.0000
## ConD9  805.6572  60.22672 13.377072  0.0000
## ConD10 953.1624  60.22672 15.826237  0.0000
## ConD11 848.7295  60.24010 14.089112  0.0000
```

```
##
##  Correlation:
##        ConD1 ConD2 ConD3 ConD4 ConD5 ConD6 ConD7 ConD8 ConD9 ConD10
## ConD2  0.008
## ConD3  0.000 0.008
## ConD4  0.000 0.000 0.008
## ConD5  0.000 0.000 0.000 0.008
## ConD6  0.000 0.000 0.000 0.000 0.008
## ConD7  0.000 0.000 0.000 0.000 0.000 0.008
## ConD8  0.000 0.000 0.000 0.000 0.000 0.000 0.008
## ConD9  0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.008
## ConD10 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.008
## ConD11 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.008
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -2.09408234 -0.54891589 -0.02860737  0.36165603  3.53466533
##
## Residual standard error: 179.5199
## Degrees of freedom: 110 total; 99 residual
```

P6.4

(a)

```
data("WaterEjection")
modgls <- gls(distance~poly(depth.z,9),data=WaterEjection)
summary(modgls)
```

```
## Generalized least squares fit by REML
##   Model: distance ~ poly(depth.z, 9)
##   Data: WaterEjection
##        AIC      BIC    logLik
##   69.51142 80.46448 -23.75571
##
## Coefficients:
##                       Value Std.Error  t-value p-value
## (Intercept)       11.360000 0.1330831 85.36020  0.0000
## poly(depth.z, 9)1 18.509887 0.7289262 25.39336  0.0000
## poly(depth.z, 9)2 -3.598035 0.7289262 -4.93608  0.0001
## poly(depth.z, 9)3  2.986844 0.7289262  4.09759  0.0006
## poly(depth.z, 9)4  0.274214 0.7289262  0.37619  0.7107
## poly(depth.z, 9)5 -1.393324 0.7289262 -1.91147  0.0704
## poly(depth.z, 9)6  2.229354 0.7289262  3.05841  0.0062
## poly(depth.z, 9)7 -1.682041 0.7289262 -2.30756  0.0318
## poly(depth.z, 9)8  0.425356 0.7289262  0.58354  0.5661
## poly(depth.z, 9)9 -0.941044 0.7289262 -1.29100  0.2114
##
##  Correlation:
##                   (Intr) p(.,9)1 p(.,9)2 p(.,9)3 p(.,9)4 p(.,9)5 p(.,9)6
## poly(depth.z, 9)1 0
## poly(depth.z, 9)2 0      0
## poly(depth.z, 9)3 0      0       0
## poly(depth.z, 9)4 0      0       0       0
## poly(depth.z, 9)5 0      0       0       0       0
## poly(depth.z, 9)6 0      0       0       0       0       0
## poly(depth.z, 9)7 0      0       0       0       0       0       0
## poly(depth.z, 9)8 0      0       0       0       0       0       0
## poly(depth.z, 9)9 0      0       0       0       0       0       0
##                   p(.,9)7 p(.,9)8
## poly(depth.z, 9)1
## poly(depth.z, 9)2
## poly(depth.z, 9)3
## poly(depth.z, 9)4
## poly(depth.z, 9)5
## poly(depth.z, 9)6
## poly(depth.z, 9)7
## poly(depth.z, 9)8 0
## poly(depth.z, 9)9 0       0
##
## Standardized residuals:
##        Min          Q1         Med          Q3         Max
## -2.1950097 -0.5030231  0.1143234  0.5487524  1.2346930
##
## Residual standard error: 0.7289262
## Degrees of freedom: 30 total; 20 residual
```

```r
# so I choose power 2 or 3
mod2 <- lm(distance~depth.z + I(depth.z^2),data = WaterEjection)
summary(mod2)
```

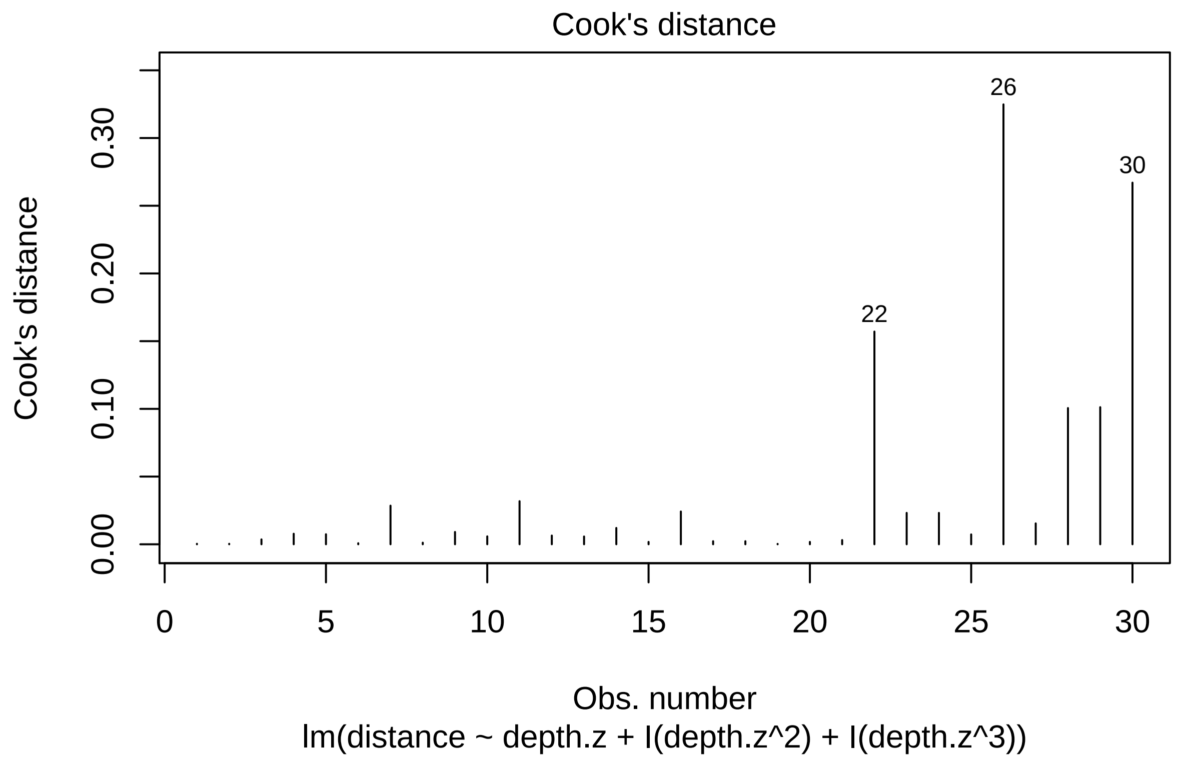
```
##
## Call:
## lm(formula = distance ~ depth.z + I(depth.z^2), data = WaterEjection)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.68040 -0.62419 -0.04848  0.60258  2.60061
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -10.21515    2.81820  -3.625  0.00118 **
## depth.z        3.07505    0.56421   5.450 9.11e-06 ***
## I(depth.z^2)  -0.09040    0.02667  -3.389  0.00217 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.062 on 27 degrees of freedom
## Multiple R-squared:  0.9212, Adjusted R-squared:  0.9153
## F-statistic: 157.7 on 2 and 27 DF,  p-value: 1.276e-15
```

```r
mod3 <- lm(distance~depth.z + I(depth.z^2) + I(depth.z^3),data = WaterEjection)
summary(mod3)
```

```
##
## Call:
## lm(formula = distance ~ depth.z + I(depth.z^2) + I(depth.z^3),
##     data = WaterEjection)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.94104 -0.44634  0.06685  0.43567  1.94901
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -41.361290   9.787016  -4.226 0.000259 ***
## depth.z       12.883074   3.025569   4.258 0.000238 ***
## I(depth.z^2)  -1.067793   0.298507  -3.577 0.001394 **
## I(depth.z^3)   0.031028   0.009449   3.284 0.002924 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9095 on 26 degrees of freedom
## Multiple R-squared:  0.9443, Adjusted R-squared:  0.9378
## F-statistic: 146.9 on 3 and 26 DF,  p-value: < 2.2e-16
```

```
# They all fit well, I choose the power3.
plot(mod3,which = 4)
```



```
# no outlier
```

(b)

```
modlin <- lm(distance~depth,data = WaterEjection)
anova(mod3, modlin)
```

```
## Analysis of Variance Table
##
## Model 1: distance ~ depth.z + I(depth.z^2) + I(depth.z^3)
## Model 2: distance ~ depth
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     26 21.509
## 2     20 10.627  6    10.882 3.4135 0.01745 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# reject the null hypothesis, larger model is better, so I choose my model.
```

```
# E7.2
sample.size.f.test(0.9,0.05,means = c(10,11,11),sigma2 = 4)
```

```
## $nis
## [1] 77 77 77
##
## $power
## [1] 0.9002725
```

```
# so we need 77 sample sizes.

# E7.3

power.f.test(ncp = 8, df1 = 3, df2 = 12,alpha = 0.01)
```

```
## [1] 0.2260942
```

```
# P7.2

sample.size.f.test(0.95, 0.01, means = c(45,32,60),sigma2 = 35)
```

```
## $nis
## [1] 4 4 4
##
## $power
## [1] 0.9808418
```

```
# P7.6

# the size is around 15 I think.
```