STAT 661: Project

LS v.s. EM

Jacob, Robin, Ryan & Shen

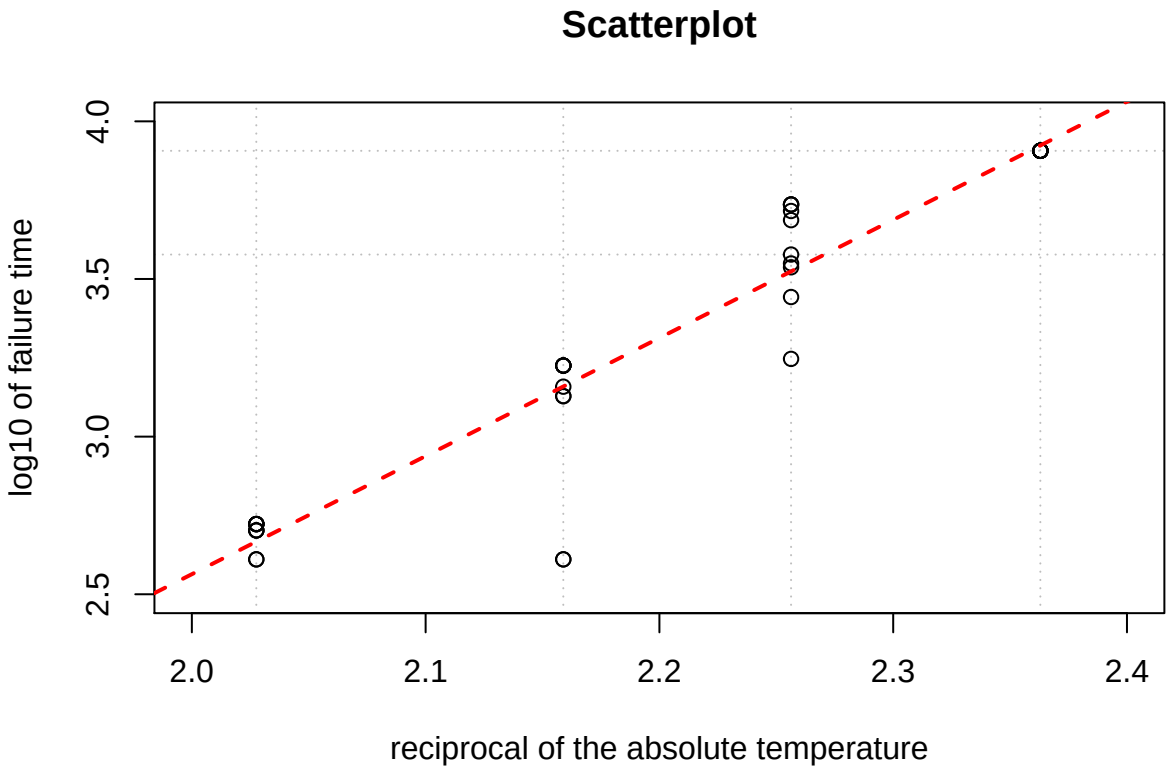
Dec, 2019

1 Least Square Method v.s. EM Method

2 Appendix

2.1 LS Code

```
temp <- c(150,170,190,220) #temperature levels
trec <- 1000/(temp+273.2) #reciprocal of the absolute temperature T
x <- c(rep(trec[1],10),rep(trec[2],10),rep(trec[3],10),rep(trec[4],10))
cen <- c(8064,5448,1680,528) #censoring times
logcen <- log10(cen) #log10 censoring times
y_uncensored <-log10(c(rep(1,10),
                       1764,2772,3444,3542,3780,4860,5196,rep(1,3),
                       408,408,1344,1344,1440,rep(1,5),
                       408,408,504,504,504,rep(1,5)))
y_censored <- c(rep(logcen[1],10),
                rep(0,7),rep(logcen[2],3),
                rep(0,5),rep(logcen[3],5),
                rep(0,5),rep(logcen[4],5))
S <- 23; Y<-matrix(nrow=S,ncol=40)
Y[1,] <- y_0 <- y_uncensored+y_censored
fit0 <- lm(y_0~x) #linear model between log10 of observed life time
```



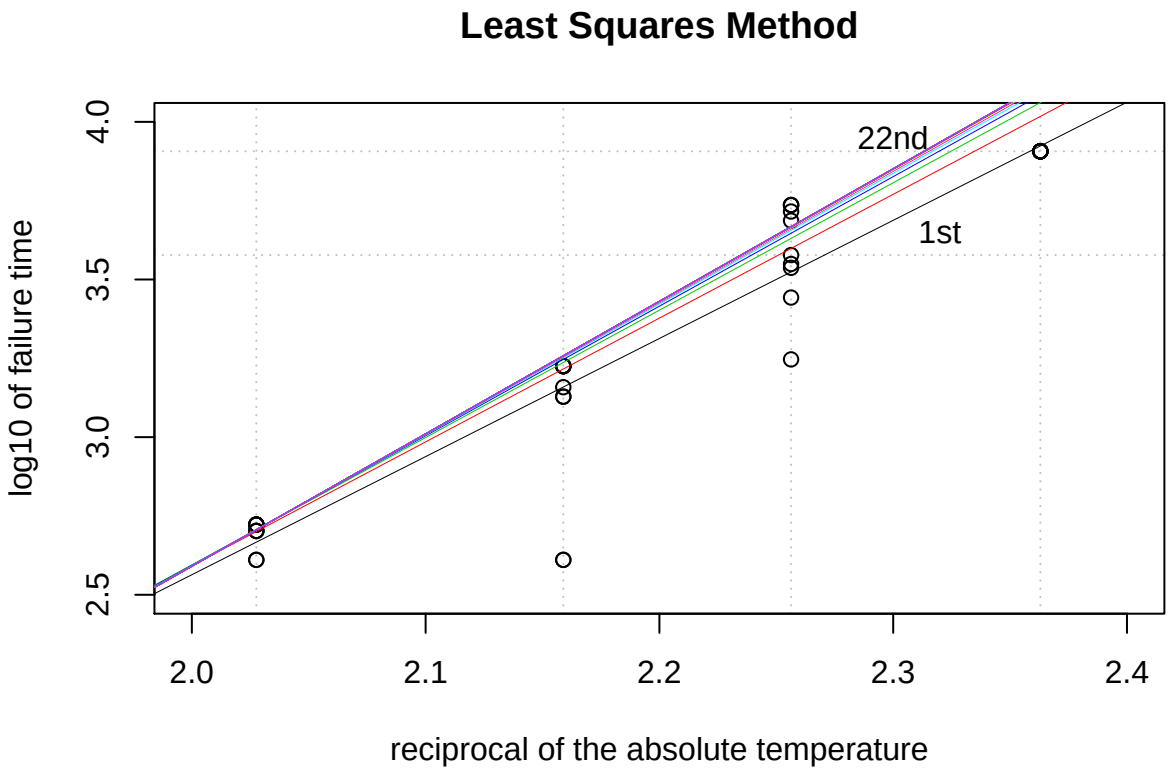

```
# Iteration 0
sigma_0 <- sigma(fit0) #standard error of residuals
beta_00 <- coef(fit0)[1] #intercept
beta_10 <- coef(fit0)[2] #slope
mu_0 <- beta_00 + beta_10*trec #mean log time to failure
z <- (logcen-mu_0)/sigma_0 #z-vector
ex_mu_0 <- mu_0 + sigma_0*dnorm(z)/(1-pnorm(z)) #new expected mean log times to failure

delta = 1e-006; iteration <- 1

PHI<-matrix(nrow=S,ncol=8,dimnames=list(NULL,
     c('mu150','mu170','mu190','mu220','Intercept','Slope','Sigma','Iteration')))

PHI[1,]<-phi<-c(ex_mu_0, beta_00, beta_10,sigma_0,iteration)
# Subsequent iteration
repeat {
 phi[8] <- phi[8]+1
 y_censored <- c(rep(phi[1],10),
                 rep(0,7),rep(phi[2],3),
                 rep(0,5),rep(phi[3],5),
                 rep(0,5),rep(phi[4],5))

 y<- y_uncensored+y_censored
 Y[phi[8],]<-y   # Replace the new censored values
 fit <- lm(y~x)    # fit a new model
 phi[5] <- coef(fit)[1] #intercept
 phi[6] <- coef(fit)[2] #slope
 phi[7] <- sigma(fit) #standard error of residuals
 mu <- phi[5] + phi[6]*trec
```

```
 z <- (logcen-mu)/phi[7] #z-vector
  #new expected mean log times to failure
 phi[1:4] <- mu + phi[7]*dnorm(z)/(1-pnorm(z))
 conv <- dist(rbind(PHI[phi[8]-1,1:4],phi[1:4]))
 if(conv < delta) break
  PHI[phi[8],]<-phi
}
```

2.2 LS Results

mu150	mu170	mu190	mu220	Intercept	Slope	Sigma	Iteration
4.038	3.809	3.329	2.83	-4.931	3.747	0.1572	1
4.099	3.84	3.366	2.858	-5.26	3.926	0.1799	2
4.131	3.856	3.382	2.869	-5.486	4.04	0.1911	3
4.149	3.865	3.39	2.874	-5.623	4.108	0.1969	4
4.159	3.87	3.395	2.877	-5.704	4.148	0.2001	5
4.165	3.873	3.397	2.878	-5.752	4.172	0.2019	6
4.168	3.874	3.399	2.879	-5.779	4.185	0.2029	7
4.17	3.875	3.4	2.879	-5.795	4.193	0.2035	8
4.171	3.876	3.4	2.879	-5.805	4.198	0.2039	9
4.172	3.876	3.401	2.88	-5.81	4.2	0.2041	10
4.172	3.877	3.401	2.88	-5.814	4.202	0.2042	11
4.173	3.877	3.401	2.88	-5.815	4.203	0.2042	12
4.173	3.877	3.401	2.88	-5.817	4.203	0.2043	13
4.173	3.877	3.401	2.88	-5.817	4.204	0.2043	14
4.173	3.877	3.401	2.88	-5.818	4.204	0.2043	15
4.173	3.877	3.401	2.88	-5.818	4.204	0.2043	16
4.173	3.877	3.401	2.88	-5.818	4.204	0.2043	17
4.173	3.877	3.401	2.88	-5.818	4.204	0.2043	18
4.173	3.877	3.401	2.88	-5.818	4.204	0.2043	19
4.173	3.877	3.401	2.88	-5.818	4.204	0.2043	20
4.173	3.877	3.401	2.88	-5.818	4.204	0.2043	21
4.173	3.877	3.401	2.88	-5.818	4.204	0.2043	22
NA	NA	NA	NA	NA	NA	NA	NA

2.3 LS figure



2.4 EM Method

- E-step

$$Q(\vec{\theta},\vec{\theta^\star}) = -\frac{n}{2}\ln(2\pi) - n\ln(\sigma) - \frac{1}{2\sigma^2}\sum_{j=1}^{m}(t_j-\beta_0-\beta_1\nu_j)^2 - \frac{1}{2\sigma^2}\sum_{i=m+1}^{n}E[(T_i-\beta_0-\beta_1\nu_i)^2|T_i>w_i,\vec{\theta^\star}]$$

$$E[T_i|T_i>w_i,\vec{\theta^\star}] = \mu_i^\star + \sigma^\star H(\frac{w_i-\mu_i^\star}{\sigma^\star})$$

$$E[T_i^2|T_i>w_i,\vec{\theta^\star}] = \mu_i^{\star 2} + \sigma^{\star 2} + \sigma^\star(w_i+\mu_i^\star)H(\frac{w_i-\mu_i^\star}{\sigma^\star})$$

$$E[(T_i-\beta_0-\beta_1\nu_i)^2|T_i>w_i,\vec{\theta^\star}] = \mu_i^{\star 2}+\sigma^{\star 2}+\sigma^\star(w_i+\mu_i^\star)H(\frac{w_i-\mu_i^\star}{\sigma^\star})-2(\beta_0+\beta_1\nu_i)[\mu_i^\star+\sigma^\star H(\frac{w_i-\mu_i^\star}{\sigma^\star})]+(\beta_0+\beta_1\nu_i)^2$$

- M-step

$$\frac{\partial Q}{\partial\beta_0} = -\frac{1}{\sigma^2}\left\{\sum_{j=1}^{m}[t_j-\beta_0-\beta_1\nu_j] + \sum_{i=m+1}^{n}[\mu_i^\star+\sigma^\star H(\frac{w_i-\mu_i^\star}{\sigma^\star})-\beta_0-\beta_1\nu_i]\right\} = 0$$

$$\frac{\partial Q}{\partial \beta_1} = -\frac{1}{\sigma^2}\left\{\sum_{j=1}^{m}[t_j - \beta_0 - \beta_1\nu_j]\nu_j + \sum_{i=m+1}^{n}[\mu_i^\star + \sigma^\star H(\frac{w_i - \mu_i^\star}{\sigma^\star}) - \beta_0 - \beta_1\nu_i]\nu_i\right\} = 0$$

$$\frac{\partial Q}{\partial \sigma^2} = \frac{1}{2\sigma^2}\left\{-n + \frac{1}{\sigma^2}\sum_{j=1}^{m}[t_j - \beta_0 - \beta_1\nu_j]^2 + \frac{1}{\sigma^2}\sum_{i=m+1}^{n}E[(T_i - \beta_0 - \beta_1\nu_i)^2 | T_i > w_i, \vec{\theta}^\star]\right\} = 0$$

$$\sum_{j=1}^{m}[t_j - \beta_0 - \beta_1\nu_j]^2 + \sum_{i=m+1}^{n}\left\{\mu_i^{\star 2} + \sigma^{\star 2} + \sigma^\star(w_i + \mu_i^\star - 2\mu_i)H(\frac{w_i - \mu_i^\star}{\sigma^\star}) - 2\mu_i\mu_i^\star + \mu_i^2\right\} = n\sigma^2$$

2.5 the EM algorithm's pseudo code

Algorithm 1: EM algorithm

input : observed data $\mathcal{D} = \{\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_N\}$,joint distribution $P(\vec{x}, \vec{z}|\vec{\theta})$
output: model's parameters $\vec{\theta}$
// 1. identify hidden variables $\vec{z}$, write out the log likelihood function $\ell(\vec{x}, \vec{z}|\vec{\theta})$
$\vec{\theta}^{(0)}$ = ... // initialize
while *(!convergency)* **do**
  // 2. E-step: plug in $P(\vec{x}, \vec{z}|\vec{\theta})$, derive the formula of $Q(\vec{\theta}, \vec{\theta}^{t-1})$
  $Q(\vec{\theta}, \vec{\theta}^{t-1}) = \mathbb{E}\left[\ell_c(\vec{\theta})|\mathcal{D}, \theta^{t-1}\right]$
  // 3. M-step: find $\vec{\theta}$ that maximizes the value of $Q(\vec{\theta}, \vec{\theta}^{t-1})$
  $\vec{\theta}^t = \arg\max_{\vec{\theta}} Q(\vec{\theta}, \vec{\theta}^{t-1})$

2.6 EM Code

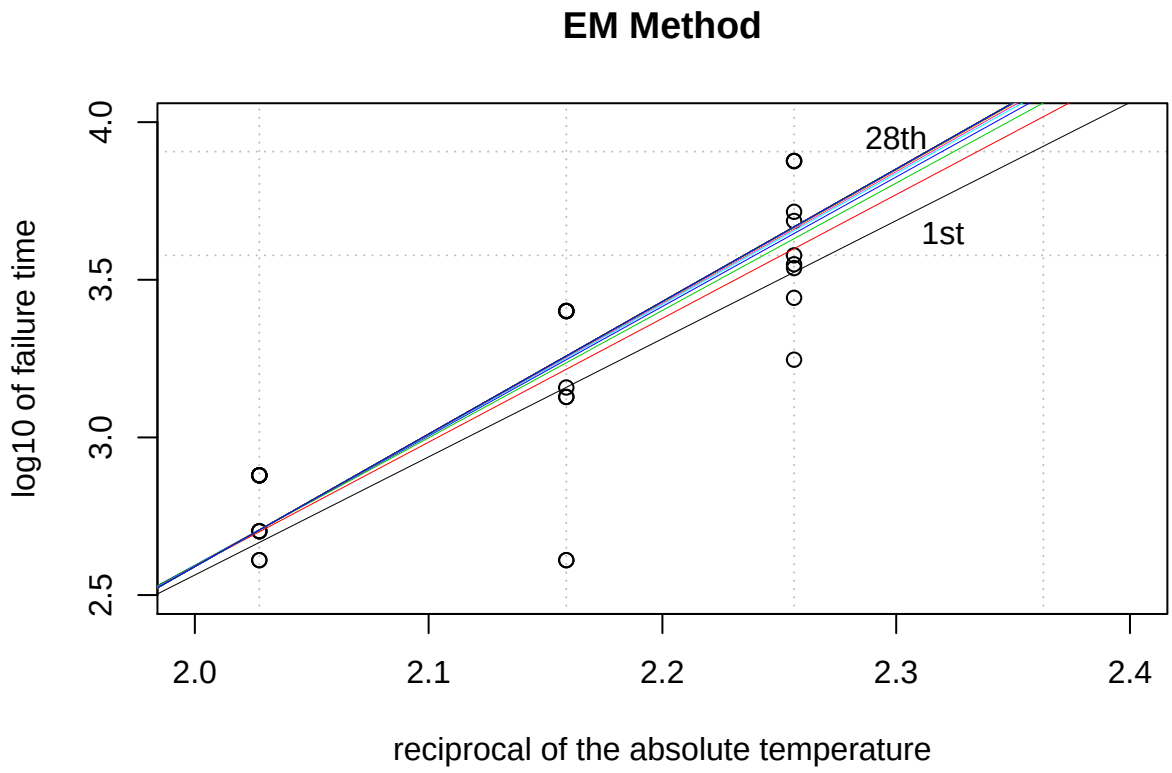
```
temp <- c(150,170,190,220) #temperature levels
trec <- 1000/(temp+273.2) #reciprocal of the absolute temperature T
nu <- c(rep(trec[1],10),rep(trec[2],10),rep(trec[3],10),rep(trec[4],10))
# index_nu <- c(rep(-2,7),rep(-3,5),rep(-4,5),rep(1,10),rep(2,3),rep(3,5),rep(4,5))

cen <- c(8064,5448,1680,528) #censoring times
w <- log10(cen) #log10 censoring times: last time still working
y_uncensored <-log10(c(rep(1,10),
                       1764,2772,3444,3542,3780,4860,5196,rep(1,3),
                       408,408,1344,1344,1440,rep(1,5),
                       408,408,504,504,504,rep(1,5)))
y_censored <- c(rep(w[1],10),
                rep(0,7),rep(w[2],3),
                rep(0,5),rep(w[3],5),
                rep(0,5),rep(w[4],5))

S <- 30
Y<-matrix(nrow=S,ncol=40)
Y[1,] <- y <-  (y_uncensored+y_censored)

index_censored <- which(y %in% w)
```

```
index_uncensored <- which(!(y %in% w))

m <- length(index_uncensored)
n<-length(index_censored)+m

# initial value
fit0 <- lm(y~nu) #linear model
sigma_0 <- sigma(fit0) #standard error of residuals
beta0_0 <- as.numeric(coef(fit0)[1]) #intercept
beta1_0 <- as.numeric(coef(fit0)[2]) #slope

mu_i0 <- beta0_0 + beta1_0 * nu[index_censored]
mu_j0 <- beta0_0 + beta1_0 * nu[index_uncensored]
D_j <- y[index_uncensored]-mu_j0

beta0_star <- beta0_0
beta1_star<- beta1_0
sigma_star<- sigma_0  # should be replaced by solutions

par=c(beta0_star,beta1_star,sigma_star)

# define mu_i function
mu_i_F <- function (beta0,beta1) {
  beta0+beta1*nu[index_censored]
}

k=1

# define H function
HF <- function (beta0,beta1,sigma) {
  dnorm((y_censored[index_censored]-mu_i_F(beta0,beta1))/sigma
       )/(
1-pnorm((y_censored[index_censored]-mu_i_F(beta0,beta1))/sigma
       ) )
}


ET <- mu_i_F(par[1],par[2])+ par[3]*HF(par[1],par[2],par[3])
ET_sq <- mu_i_F(par[1],par[2])^2+
         sigma_star^2+
         sigma_star*(y[index_censored]+mu_i_F(par[1],par[2]))*HF(par[1],par[2],par[3])
ER <- ET_sq-2*mu_i0*ET+mu_i0^2

Q <- 0
Q[1] <- -n*log(2*pi)/2-n*log(sigma_0)-(1/(2*sigma_0^2))*(sum((D_j)^2)+sum(ER))

THETA<-matrix(nrow=S,ncol=8,dimnames=list(NULL,
       c('Intercept','Slope','Sigma','mu150','mu170','mu190','mu220','Iteration')))
THETA[1,]<-c(par,unique(ET),k)
delta = 1e-6
repeat {
    # load THETA
beta0  <- THETA[k,1]
```

```r
beta1  <- THETA[k,2]
sigma  <- THETA[k,3]

# Update y
y[index_censored] <- ET
y[index_uncensored] <- y_uncensored[index_uncensored]

# E step

mu_i <- beta0 + beta1 * nu[index_censored]
mu_j <- beta0 + beta1 * nu[index_uncensored]
D_j <- y[index_uncensored]-mu_j  # difference between uncensored y and mean_j


Q_theta <- function(par){
 -n*log(2*pi)/2-
  n*log(sigma)-
 (1/(2*sigma^2))*sum((D_j)^2)-          # 1/2sigma^2*sum(j uncensored) -
 (1/(2*sigma^2))*sum(                   # 1/2sigma^2*sum(i censored
  (mu_i_F(par[1],par[2]))^2+            # mu_i_star^2+
   par[3]*(                               # sigma_star*(
           y[index_censored]+             #             w_i+
           mu_i_F(par[1],par[2])-         #             mu_i_star-
           2*mu_i                         #             2mu_i
           )*HF(par[1],par[2],par[3])-    #            )*H)-
   2*mu_i*mu_i_F(par[1],par[2])+          # 2*mu_i*mu_i_star+
   mu_i^2                                 # mu_i^2
                  )+                      #            )+
  (1/(2*sigma^2))*(n-m)*(par[3]^2)      # 1/2sigma^2*(n-m)*sigma_star^2+
}

# Get Q
Q[k]<- Q_theta(par)

# M step

############ from Jacob
# par <- optim(par,
#            Q_theta,
#            control=list(fnscale=-1),
#            method="L-BFGS-B",
#             lower=c(-10,2,0.1),
#             upper=c(-2,10,1))$par

############ cheat
fit <- lm(y~nu)   # fit a new model
par[1] <- coef(fit)[1] #intercept
par[2] <- coef(fit)[2] #slope
par[3] <- sigma(fit)
#############

k <-  k+1
```

```r
# Update ET
ET <- mu_i_F(par[1],par[2])+ par[3]*HF(par[1],par[2],par[3])

# Update Q
Q[k]<- Q_theta(par)

# Update THETA
 THETA[k,1] <- par[1]
 THETA[k,2] <- par[2]
 THETA[k,3] <- par[3]
 THETA[k,4:7]<-unique(ET)
 THETA[k,8] <- k

# Update Y
Y[k,] <- y

# par=c(beta0,beta1,sigma)
 if(abs(Q[k]-Q[k-1])<=delta) break
}
```

2.7 EM figure



2.8 The results of THETA

Intercept	Slope	Sigma	mu150	mu170	mu190	mu220	Iteration
-4.931	3.747	0.1572	4.038	3.809	3.329	2.83	1

Intercept	Slope	Sigma	mu150	mu170	mu190	mu220	Iteration
-5.26	3.926	0.1799	4.099	3.84	3.366	2.858	2
-5.486	4.04	0.1911	4.131	3.856	3.382	2.869	3
-5.623	4.108	0.1969	4.149	3.865	3.39	2.874	4
-5.704	4.148	0.2001	4.159	3.87	3.395	2.877	5
-5.752	4.172	0.2019	4.165	3.873	3.397	2.878	6
-5.779	4.185	0.2029	4.168	3.874	3.399	2.879	7
-5.795	4.193	0.2035	4.17	3.875	3.4	2.879	8
-5.805	4.198	0.2039	4.171	3.876	3.4	2.879	9
-5.81	4.2	0.2041	4.172	3.876	3.401	2.88	10
-5.814	4.202	0.2042	4.172	3.877	3.401	2.88	11
-5.815	4.203	0.2042	4.173	3.877	3.401	2.88	12
-5.817	4.203	0.2043	4.173	3.877	3.401	2.88	13
-5.817	4.204	0.2043	4.173	3.877	3.401	2.88	14
-5.818	4.204	0.2043	4.173	3.877	3.401	2.88	15
-5.818	4.204	0.2043	4.173	3.877	3.401	2.88	16
-5.818	4.204	0.2043	4.173	3.877	3.401	2.88	17
-5.818	4.204	0.2043	4.173	3.877	3.401	2.88	18
-5.818	4.204	0.2043	4.173	3.877	3.401	2.88	19
-5.818	4.204	0.2043	4.173	3.877	3.401	2.88	20
-5.818	4.204	0.2043	4.173	3.877	3.401	2.88	21
-5.818	4.204	0.2043	4.173	3.877	3.401	2.88	22
-5.818	4.204	0.2043	4.173	3.877	3.401	2.88	23
-5.818	4.204	0.2043	4.173	3.877	3.401	2.88	24
-5.818	4.204	0.2043	4.173	3.877	3.401	2.88	25
-5.818	4.204	0.2043	4.173	3.877	3.401	2.88	26
-5.818	4.204	0.2043	4.173	3.877	3.401	2.88	27
-5.818	4.204	0.2043	4.173	3.877	3.401	2.88	28
NA	NA	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA

2.9 The results of Q

20.08, *21.4*, *20.64*, *20.09*, *19.75*, *19.55*, *19.43*, *19.36*, *19.32*, *19.29*, *19.28*, *19.27*, *19.27*, *19.26*, *19.26*, *19.26*, *19.26*, *19.26*, *19.26*, *19.26*, *19.26*, *19.26*, *19.26*, *19.26*, *19.26*, *19.26*, *19.26*, *19.26* and *19.26*

3 Reference

Schmee, J., & Hahn, G. (1979). A Simple Method for Regression Analysis with Censored Data. Technometrics, 21(4), 417-432. doi:10.2307/1268280

Aitkin, M. (1981). A Note on the Regression Analysis of Censored Data. Technometrics, 23(2), 161-163. doi:10.2307/1268032