Reliability-HW5

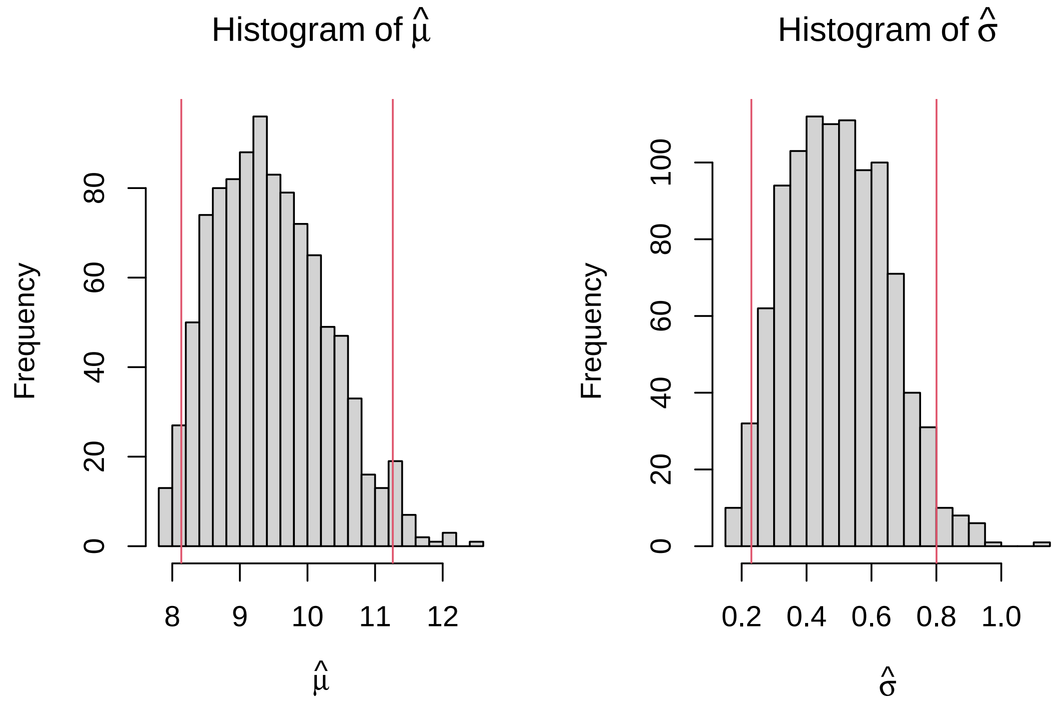
ID : 111024517 Name : 鄭家豪

Problem 1

(a)

Generate $(u_1, u_2, ..., u_{1703})$ using the uniform Dirichlet distribution,take $1703 \times (u_1, u_2, ..., u_{1703})$ as the weights at each bootstrap.
Compute the bootstrap ML estimates $\hat{\mu_i^*}$ and $\hat{\sigma_i^*}, i = 1, 2, ..., 1000$ of the SEV distribution and print the histogram of bootstrap samples:

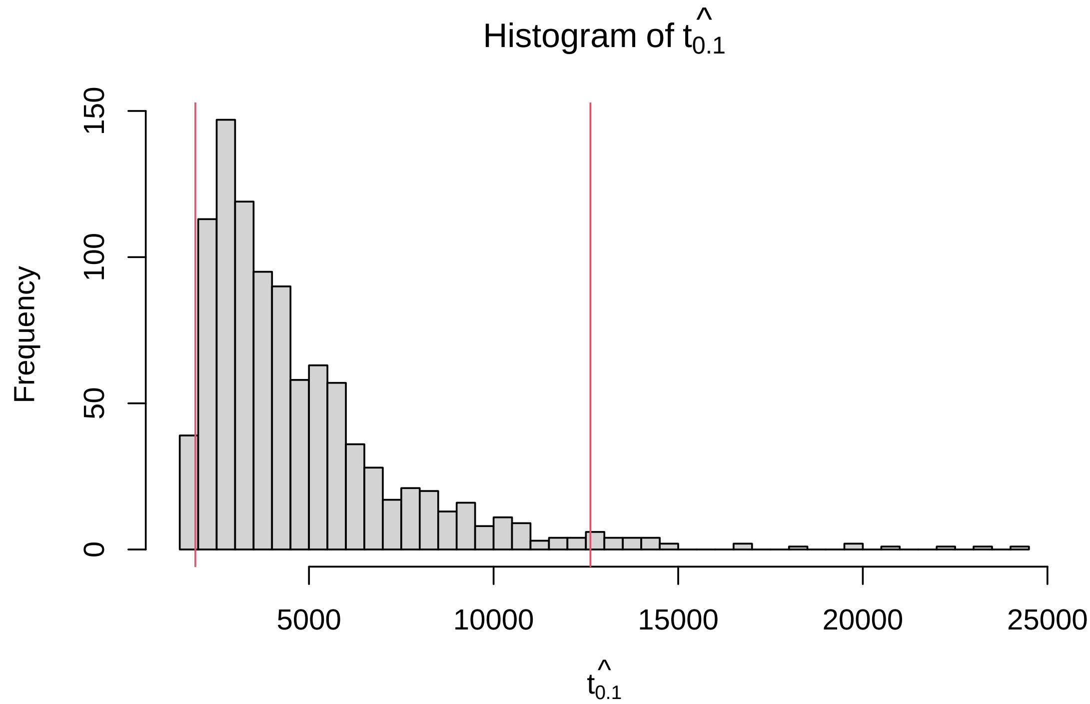
```
The 95% CI for mu:

     2.5%     97.5%
 8.133386 11.262154

The 95% CI for sigma:

     2.5%     97.5%
0.2298924 0.8004359
```



(b)

By the MLE result of part (a),

$$\hat{t}^*{}_{0.1,i} = \exp(\hat{\mu}_i{}^* + q_{0.1} \times \hat{\sigma}_i{}^*), \text{for } i = 1, 2, ..., 1000.$$



(c)

```
The 95% CI for 0.1 quantile:

     2.5%     97.5%
 1923.405 12621.354
```

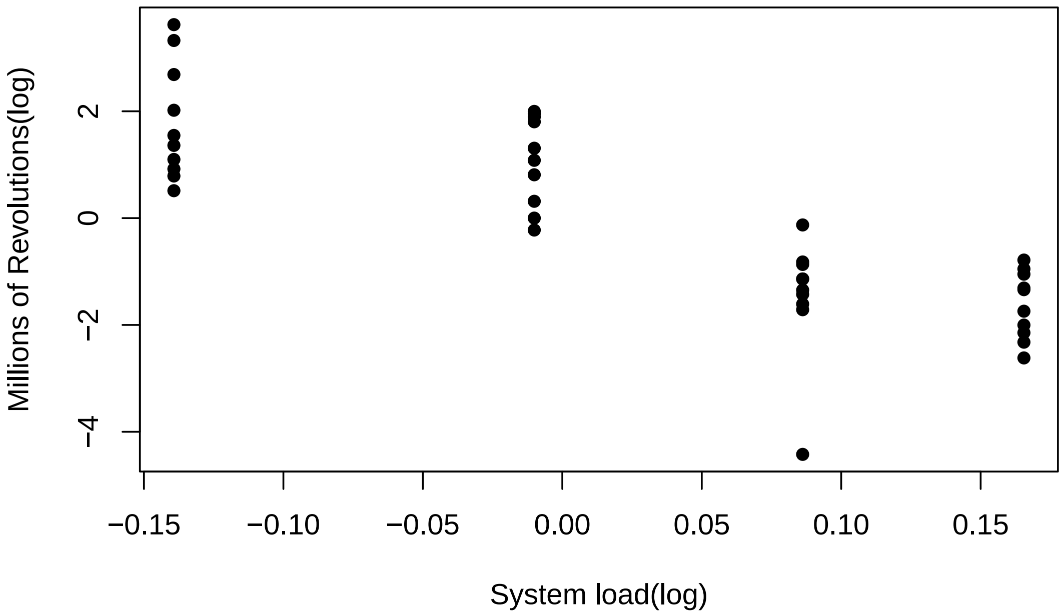
We know the Wald method assumes a normal distribution of the data and tends to produce narrow CI than other methods,and the LR-based method can produce wider intervals by LRT.The LR-based method is more robust and reliable,especially for non-normal distribution data.

In this question,the bootstrap uses a resampling method that does not assume any distribution and can obtain wider intervals than the Wald method.

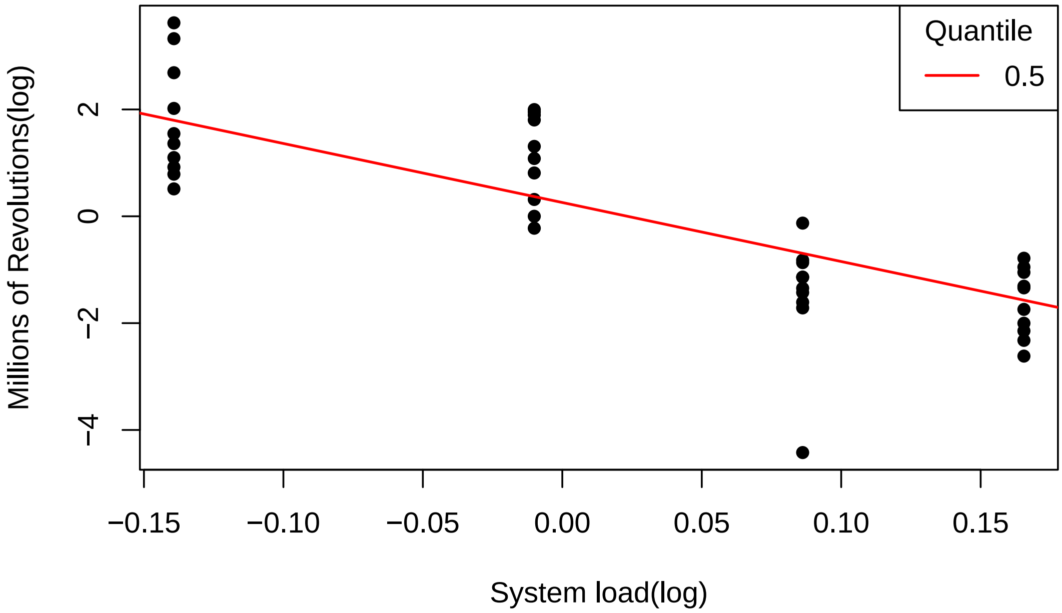
So,the LR-based method is conservative and less prone to overconfidence,the bootstrap method is more accurate and reliable than the Wald method.

Problem 2

(a)



(b)



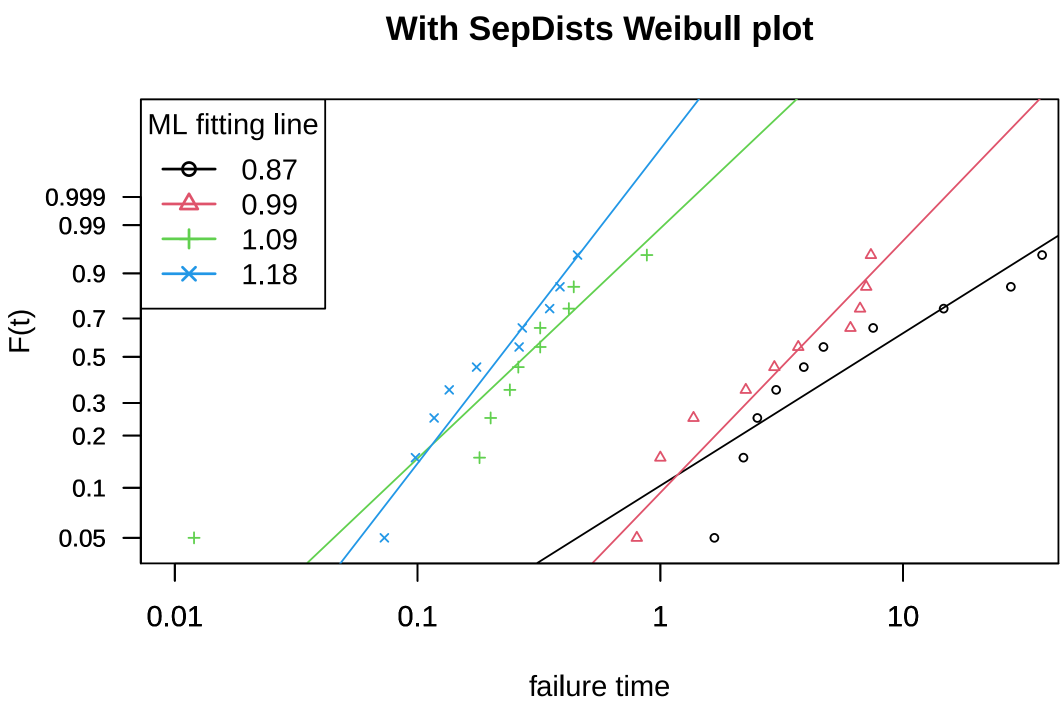
Due to the most of the failure times for each stress fall near the red line,the suggestion is reasonable in this case.

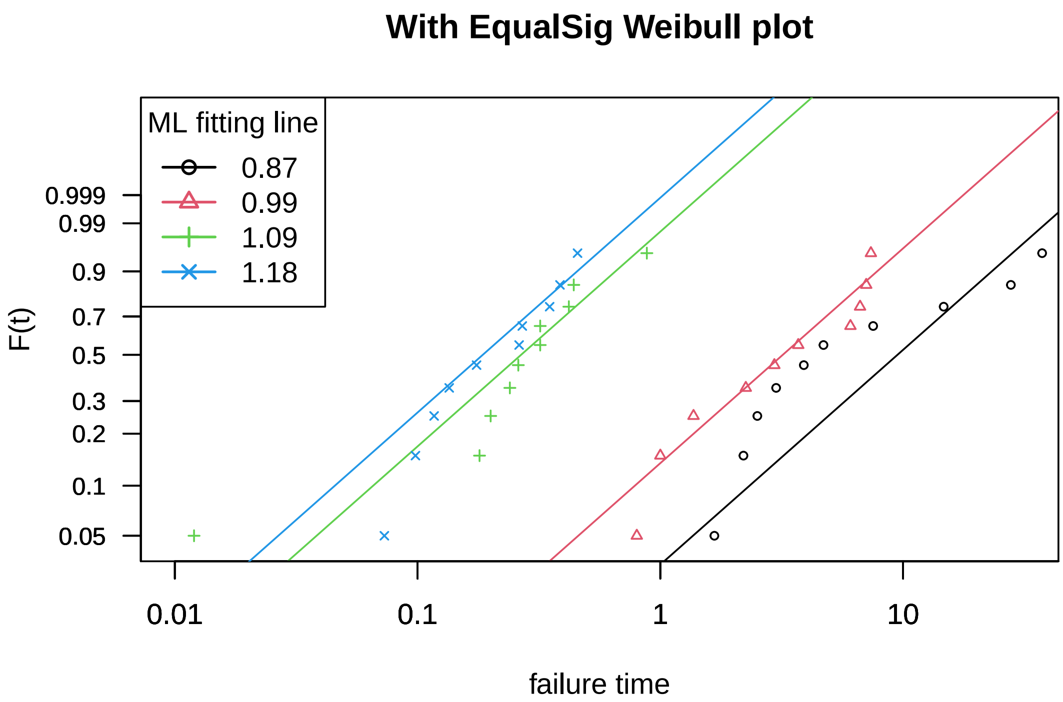
(c)

For log location family to Weibull,the loglikelihood function for SepDists:

$$L(\alpha_1,..,\alpha_4,\sigma_1,...,\sigma_4)=\prod_{j=1}^{4}[\prod_{x_i=t_j}[\frac{1}{\sigma t_i}\phi_{sev}(\frac{\log t_i-\alpha_j}{\sigma_j})]^{\delta_i}[1-\Phi_{sev}(\frac{\log t_i-\alpha_j}{\sigma_j})]^{1-\delta_i}],$$

By optim command ,obtain these ML estimates($\alpha_1,..,\alpha_4,\sigma_1,..,\sigma_4$),and plot the Weibull plot at each stress ablining the ML fitting line:





The assumption of using the same Weibull shape parameter may be inappropriate, we can observe the graph of part(a), we can find that there is a very obvious downward trend, that is, the shape parameter can not have constancy significantly.

(d)

The loglikelihood function :

$$L(\beta_0,\beta_1,\sigma)=\prod_i[\frac{1}{\sigma t_i}\phi_{sev}(\frac{\log t_i-\mu_i}{\sigma})]^{\delta_i}[1-\Phi_{sev}(\frac{\log t_i-\mu_i}{\sigma})]^{1-\delta_i}, \text{where } \mu_i=\beta_0+\beta_1 x_i.$$

By optim command,the MLE of (β_0,β_1,σ):

```
The ML estimate for beta0,beta1,sigma: 0.7880708 -13.89241 0.858046
```

(e)

By the result of part(d),

$$\log(t_{\hat{p}})=\hat{\beta}_0+\hat{\beta}_1\log(stress)+\hat{\sigma}\times \text{qsev(p)}$$

So,

```
The estimate of median time for stress 1.05: 0.8152773
```

```
The estimate of 0.01 quantile for stress 1.05: 0.02156073
```

```
The estimate of 0.01 quantile for stress 0.85: 0.4060455
```

Appendix(cdoe)

```
# Problem 1
## pre-processing
BearingCage <- read.csv("BearingCage.csv",header = T)
t.index <- unique(BearingCage$Hours)
di <- rep(0,length(t.index))
label=0
for (i in 1:length(t.index)){
  label <- which(BearingCage$Hours == t.index[i] & BearingCage$Censoring.Indicator=="Failed")
  di[i] <- ifelse(length(label)==0,0,BearingCage$Count[label])
}
ri <- rep(0,length(t.index))
for (i in 1:length(t.index)){
  label <- which(BearingCage$Hours == t.index[i] & BearingCage$Censoring.Indicator=="Censored")
  ri[i] <- ifelse(length(label)==0,0,BearingCage$Count[label])
}
```

```r
ni <- 0
ni[1] <- sum(BearingCage$Count)
for(i in 2:length(di)){
  ni[i] <- ni[1] - sum(ri[1:i-1])-sum(di[1:i-1])
}
org.BearingCage <- data.frame("ti"=t.index,"di"=di,
                              "ri" = ri,
                              "ni"=ni)
logL.Weibull=function(theta,ti,di,ri,w){
  mu=theta[1];sig=theta[2]
  if(sig<0){sig=0.001}
  l=0
  z=(log(ti)-mu)/sig
  for(i in 1:length(ti)){
    l=l+di[i]*w[i]*(-log(sig)-log(ti[i])+z[i]-exp(z[i]))+
      ri[i]*(-exp(z[i]))
  }
  return(-l)
} # negative log L of Weibull adding weights
rdirichlet<-function (n, alpha){
  l <- length(alpha)
  x <- matrix(rgamma(l * n, alpha), ncol = l, byrow = TRUE)
  sm <- x %*% rep(1, l)
  x/as.vector(sm)
} # generate Dirichlet

## (a)
n = sum(BearingCage$Count) #Total count
B = 1000
set.seed(12313)
bootstrap.est <- data.frame(mu.hat = rep(NA,B),sigma.hat = rep(NA,B))
ti <- t.index
for(k in 1:B){
  w=n*rdirichlet(1,rep(1,n))
  op=optim(par = c(10,0.5),ti=ti,di=di,ri=ri,w=w,logL.Weibull,hessian=F)
  mle=op$p
  bootstrap.est$mu.hat[k]=mle[1]
  bootstrap.est$sigma.hat[k]=mle[2]
}
mu.CI <- quantile(bootstrap.est$mu.hat,c(0.025,0.975))
```

```r
sigma.CI <- quantile(bootstrap.est$sigma.hat,c(0.025,0.975))
cat("The 95% CI for mu:","\n")
mu.CI
cat("The 95% CI for sigma:","\n")
sigma.CI
par(mfrow = c(1,2))
hist(bootstrap.est$mu.hat,xlab=expression(hat(mu)),30,
     main=expression(Histogram~of~hat(mu)))
abline(v=mu.CI,col=c(2,2),lwd = 1,lty=1)
hist(bootstrap.est$sigma.hat,xlab=expression(hat(sigma)),30,
     main=expression(Histogram~of~hat(sigma)))
abline(v=sigma.CI,col=c(2,2),lwd = 1,lty=1)

## (b)
qsev=function(p){
  log(qweibull(p,1,1))
} #quantile of SEV
tp = exp(bootstrap.est$mu.hat+qsev(0.1)*bootstrap.est$sigma.hat) # bootstrap t_0.1 samples
tp.CI <- quantile(tp,c(0.025,0.975))
hist(tp,xlab=expression(hat(t[0.10])),50,
     main=expression(Histogram~of~hat(t[0.10])))
abline(v=tp.CI,col=c(2,2),lwd = 1,lty=1)

## (c)
cat("The 95% CI for 0.1 quantile:","\n")
tp.CI

# Problem 2
## (a)
CeramicBearing02 <- read.csv("CeramicBearing02.csv",header=T)
stress <- CeramicBearing02$Stress..Mpsi.
time <- CeramicBearing02$Millions.of.Revolutions
plot(x=log(stress),y=log(time),xlab="System load(log)",
     ylab="Millions of Revolutions(log)",type = "p",pch=16)

## (b)
t.05 <- c()
for (i in 1:length(unique(stress))){
  t.05[i] <- quantile(time[which(stress == unique(stress)[i])], 0.5)
}
```

```r
lmfit <- lm(log(t.05)~log(unique(stress)))
plot(x=log(stress),y=log(time),xlab="System load(log)",
     ylab="Millions of Revolutions(log)",type = "p",pch=16)
abline(lmfit,col="red",lwd=1.5)
legend('topright', title = "Quantile", '0.5', lty = 1, col = "red", lwd = 1.5)

## (c)
di=rep(1,40) == 1
group=c()
for(i in 1:length(stress)){
  group[i]=rank(c(stress[i],unique(stress)),tie="min")[1]
} #build group for each observation
qsev=function(p){
  log(qweibull(p,1,1))
} #quantile of SEV
ld=function(x,mu,sig){
  z=(log(x)-mu)/sig
  (z-exp(z))-log(sig)-log(x)
  #-0.5*(log(2)+log(pi))-z^2/2-log(sig)-log(x) :log normal
} #log density function
lS=function(x,mu,sig){
  z=(log(x)-mu)/sig
  -exp(z)
  #log(pnorm(z,low=F)) :log normal
} #log survival function
logLS=function(theta,ti,di){
  mu=theta[1];sig=theta[2]
  if(sig<0.001)sig=0.001
  l=0
  for(i in 1:length(ti)){
    l=l+di[i]*ld(ti[i],mu,sig)+(1-di[i])*lS(ti[i],mu,sig)
  }
  -l
} # SepDists logLikelihood
mle=c()
for(i in 1:length(unique(stress))){
  op=optim(c(10,2),ti=time[which(group==i)],di=di[which(group==i)],
           logLS,hessian=T)
  mle=c(mle,op$p)
} #SepDists
```

```
logLE=function(theta,ti,di){
  sig=theta[5]
  if(sig<0.001)sig=0.001
  l=0
  for(i in 1:length(ti)){
    l=l+di[i]*ld(ti[i],theta[group[i]],sig)+(1-di[i])*lS(ti[i],theta[group[i]],sig)
  }
  -l
} # EqualSig logLikelihood
opLE=optim(c(rep(10,4),2),ti=time,di=di,logLE,hessian=T) #EqualSig MLE

## plot SeqDists Weibull plot
for(k in 1:length(unique(group))){
  tj=time[which(group==k)];dj=di[which(group==k)]
  dj=dj[order(tj)];tj=sort(tj)
  nj=length(tj):1
  pj=dj/nj
  Fj=1-Reduce("*",1-pj,acc=T)
  tab=cbind(tj,dj,nj,1-pj,Fj)[which(dj),]

  y=c(0,tab[,5])
  y=(y[-1]+y[-length(y)])/2 #kM estimate
  if(k==1){
    plot(log(tab[,1]),qsev(y),xlim=log(c(10^(-2),10^(1.5))),
         ylim=qnorm(c(.001,.999)),cex=0.6,col=k,
         xlab="failure time",ylab="F(t)",yaxt="n",xaxt="n",
         main = "With SepDists Weibull plot")
  }else{
    points(log(tab[,1]),qsev(y),cex=0.6,pch=k,col=k,
           xlab="Standardized residuals",ylab="Probability",yaxt="n",xaxt="n")
  }
  y_lab=c(.001,.003,.01,.02,.05,.1,.2,.3,.5,.7,.9,.99,.999)
  #axis(2,log(-log(1-y_lab)),y_lab,cex.axis=0.8,las=1)
  axis(2,qsev(y_lab),y_lab,cex.axis=0.8,las=1)
  x_lab=10^(-2:2)
  axis(1,log(x_lab),x_lab,las=1)
  abline(a=-mle[2*k-1]/mle[2*k],b=1/mle[2*k],col=k)
  #abline:refer to ch.8
}
legend('topleft', title = "ML fitting line", c('0.87','0.99','1.09','1.18'),
```

```
      lty = 1, col = 1:4, lwd = 1.5,pch=1:4)

## plot EqualSig Weibull plot
for(k in 1:length(unique(group))){
  tj=time[which(group==k)];dj=di[which(group==k)]
  dj=dj[order(tj)];tj=sort(tj)
  nj=length(tj):1
  pj=dj/nj
  Fj=1-Reduce("*",1-pj,acc=T)
  tab=cbind(tj,dj,nj,1-pj,Fj)[which(dj),]

  y=c(0,tab[,5])
  y=(y[-1]+y[-length(y)])/2 #kM estimate
  if(k==1){
    plot(log(tab[,1]),qsev(y),xlim=log(c(10^(-2),10^(1.5))),
         ylim=qnorm(c(.001,.999)),cex=0.6,col=k,
         xlab="failure time",ylab="F(t)",yaxt="n",xaxt="n",
         main = "With EqualSig Weibull plot")
  }else{
    points(log(tab[,1]),qsev(y),cex=0.6,pch=k,col=k,
           xlab="Standardized residuals",ylab="Probability",yaxt="n",xaxt="n")
  }
  y_lab=c(.001,.003,.01,.02,.05,.1,.2,.3,.5,.7,.9,.99,.999)
  #axis(2,log(-log(1-y_lab)),y_lab,cex.axis=0.8,las=1)
  axis(2,qsev(y_lab),y_lab,cex.axis=0.8,las=1)
  x_lab=10^(-2:2)
  axis(1,log(x_lab),x_lab,las=1)
  abline(a=-opLE$par[k]/opLE$par[5],b=1/opLE$par[5],col=k)
  #abline:refer to ch.8
}
legend('topleft', title = "ML fitting line", c('0.87','0.99','1.09','1.18'),
       lty = 1, col = 1:4, lwd = 1.5,pch=1:4)

## (d)
## Regmodel
logLReg=function(theta,ti,di,xi){
  b0=theta[1];b1=theta[2];sig=theta[3]
  if(sig<0.001)sig=0.001
  mu=b0+b1*xi
  l=0
```

```r
  for(i in 1:length(ti)){
    l=l+di[i]*ld(ti[i],mu[i],sig)+(1-di[i])*lS(ti[i],mu[i],sig)
  }
  -l
}#RegModel's -log L for Weibul
opReg=optim(c(50,-10,1),ti=time,di=di,xi=log(stress),logLReg,hessian=T)
cat("The ML estimate for beta0,beta1,sigma:",opReg$par,"\n")

## (e)
t0.5 <- as.numeric(exp(opReg$par %*% c(1,log(1.05),qsev(0.5))))
t0.01.1 <- as.numeric(exp(opReg$par %*% c(1,log(1.05),qsev(0.01))))
t0.01.2 <- as.numeric(exp(opReg$par %*% c(1,log(0.85),qsev(0.01))))
cat("The estimate of median time for stress 1.05:",t0.5,"\n")
cat("The estimate of 0.01 quantile for stress 1.05:",t0.01.1,"\n")
cat("The estimate of 0.01 quantile for stress 0.85:",t0.01.2,"\n")
```