Survival Analysis Ⅰ HW4

Example 7.1.1 of Lawless's book (2003)

- Data.frame

time	event	X	time	event	X	time	event	X
1	1	1	**8**	1	0	**22**	1	1
1	1	0	**9**	1	0	**26**	1	1
1	1	0	**10**	1	1	**27**	0	0
2	1	0	**11**	1	0	**28**	0	1
2	1	0	**12**	1	1	**29**	1	1
3	1	1	**12**	1	0	**31**	1	0
3	1	1	**14**	1	1	**34**	1	1
3	1	0	**14**	1	0	**38**	0	0
4	1	0	**15**	1	1	**40**	1	1
5	1	0	**16**	1	0	**44**	1	0
6	1	1	**18**	1	1	**48**	0	1
7	1	1	**18**	1	0	**49**	0	1
7	1	1	**19**	1	1			
8	1	0	**21**	1	0			

Where, time(t_i) is remission time, event(δ_i) is actually happened not censoring and X(x_i)denote which treatment data come from.

Notation:

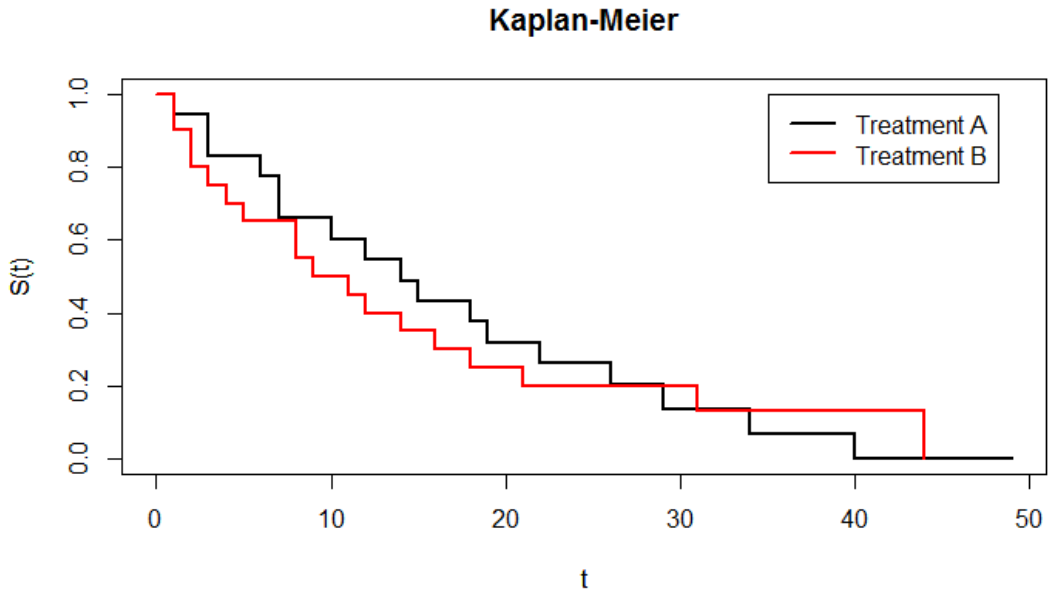
$N = the\ number\ of\ samples, \quad d_i = \delta_i, \quad d_{1i} = \delta_i x_i$

$R_i = \{j: t_j \geq t_{(i)}\}, \quad t_{(i)}\ is\ ordered\ times$

$n_{1i} = \sum_{\ell \in R_i} x_i, \quad n_{2i} = \sum_{\ell \in R_i} (1 - x_i), \quad n_i = n_{1i} + n_{2i}$

➢ Survival $t^* ordered\ time\ at\ death$ $\hat{S}(t) = \prod_{t_j^* < t}(1 - \frac{d_j}{n_j})$

where $d_j = \sum_{i=1}^{n} I(t_i = t_j^*, \delta_i = 1)$ $n_j = \sum_{i=1}^{n} I(t_i \geq t_j^*)$



The Kaplan-Meier estimates for two treatments show the correctness of the proportional hazard assumption.

➢ Score test

$$U(0) = \sum_{i=1}^{N}\left(d_{1i} - \frac{d_i n_{1i}}{n_i}\right) \quad \mathrm{I}(0) = \sum_{i=1}^{N} \frac{d_i n_{1i} n_{2i}}{n_i^2}$$

$$Z = \frac{U(0)}{I(0)^{\frac{1}{2}}} \dot{\sim} N(0,1) \quad \Rightarrow \quad Z^2 \equiv \mathcal{X}^2_{df=1} \ and\ \mathcal{X}^2_1(0.05) = 3.841459$$

We find $U(0) = -3.323137$ $I(0) = 8.408741$

$$\Rightarrow Z^2 = 1.313305 < 3.841459 \ and\ p - value = 0.2517972$$

All conclude that have no evidence prove $S_1(t) \neq S_2(t)$

- Score test 2

$$\text{Improve I}(0) = \sum_{i=1}^{k} \frac{d_i(n_i - d_i)n_{1i}n_{2i}}{n_i^2(n_i - 1)}$$

Where i indexes the k distinct times at which failures occur, and d_i is the total number of failures at the i_{th} time $t_{(i)}$

time	di	ni	n1	n2	time	di	ni	n1	n2
1	3	40	20	20	15	1	18	11	7
2	2	37	19	18	16	1	17	10	7
3	3	35	19	16	18	2	16	10	6
4	1	32	17	15	19	1	14	9	5
5	1	31	17	14	21	1	13	8	5
6	1	30	17	13	22	1	12	8	4
7	2	29	16	13	26	1	11	7	4
8	2	27	14	13	29	1	8	5	3
9	1	25	14	11	31	1	7	4	3
10	1	24	14	10	34	1	6	4	2
11	1	23	13	10	40	1	4	3	1
12	2	22	13	9	44	1	3	2	1
14	2	20	12	8					

We find $U(0) = -3.323137$ $I(0) = 8.196201$

$$\Rightarrow Z^2 = 1.347361 < 3.841459 \ and \ p - value = 0.2457401$$

Both conclude that have no evidence prove $S_1(t) \neq S_2(t)$

- Iteration for β

Fix-point algorithm:

Step1: β_0

Step2: $e^{\beta_j} = \frac{\sum_{i=1}^{N} d_i}{\sum_{i=1}^{N} \frac{d_i n_i}{n_{1i} e^{\beta_{j-1}} + n_{2i}}}$ $j = 1,2 \ldots$

Step3: $\hat{\beta} \equiv \beta_j$ if $|\beta_j - \beta_{j-1}| < 10^{-5}$

Where $U(\theta) = \frac{\partial \ell(\theta)}{\partial(\theta)}$ $H(\theta) = \frac{\partial^2 \ell(\theta)}{\partial(\theta)^2}$

Newton-Raphson algorithm:

Step1: β_0

Step2: $\beta_j = \beta_{j-1} - H(\beta_{j-1})^{-1} U(\beta_{j-1})$ $j = 1,2 \ldots$

Step3: $\hat{\beta} \equiv \beta_j$ if $|\beta_j - \beta_{j-1}| < 10^{-5}$

Where $U(\beta) = \frac{\partial \ell(\beta)}{\partial(\beta)}$ $H(\beta) = \frac{\partial^2 \ell(\beta)}{\partial(\beta)^2}$

From Fix-point algorithm within 10^{-5} error

$$\hat{\beta} = -0.387997 \quad \text{by 18 iterations and 0.056566 secs}$$

From Newton-Raphson algorithm within 10^{-5} error

$$\hat{\beta} = -0.3880057 \quad \text{by 4 iterations and 0.06106305 secs}$$

We can see the each iteration for beta in the below "Output".

Therefore, we found the Newton-Raphson algorithm which has less iterations than Fix-point algorithm. And the passing time are almost same between two method. But we know the Newton-Raphson algorithm is more complicated to estimate the parameters.

- Wald test

$$U(\beta)=\sum_{i=1}^{N}\left(d_{1i}-\frac{d_i n_{1i}e^{\beta}}{n_{1i}+n_{2i}e^{\beta}}\right)\quad I(\beta)=\sum_{i=1}^{N}\frac{d_i n_{1i}n_{2i}e^{\beta}}{(n_{1i}e^{\beta}+n_{2i})^2}$$

$$Z=\frac{\hat{\beta}}{se\left(\hat{\beta}\right)^2}\sim N(0,1)\quad\Rightarrow\quad Z^2\equiv\mathcal{X}^2_{df=1}\ and\ se\left(\hat{\beta}\right)^2=I(\beta)^{-\frac{1}{2}}$$

We find $\hat{\beta}=-0.387997\ \ I\left(\hat{\beta}\right)=8.619081$

$$\Rightarrow\ Z^2=1.297531<3.841459\ \ and\ p-value=0.2546647$$

Both conclude that have no evidence prove $S_1(t)\neq S_2(t)$

- Likelihood ratio test

$$\ell(\beta)=\sum_{i=1}^{N}d_i\left[\beta x_i-\log\left\{\sum_{\ell\in R_i}(1-x_i)\right\}\right]$$

$$\Lambda=2\{\ell\left(\hat{\beta}\right)-\ell(0)\}\sim\mathcal{X}^2_{df=1}\quad and\ \mathcal{X}^2_1(0.05)=3.841459$$

We find $\hat{\beta}=-0.387997\ \ \ell\left(\hat{\beta}\right)=-103.2979\ \ \ell(0)=-103.9453$

$$\Rightarrow\ \Lambda=1.294698<3.841459\ \ and\ p-value=0.255184$$

Both conclude that have no evidence prove $S_1(t)\neq S_2(t)$

- Output

```
>#scorce test
 [1] 3.841459
> U0
[1] -3.323137
> I0
[1] 8.408741
> Chi
[1] 1.313305
> p_value
[1] 0.2517972
> qchisq(0.95,1)
[1] 3.841459
```

```
>#scorce test2
 [1] 3.841459
> U0
[1] -3.323137
> I00
[1] 8.196201
> Chi
[1] 1.347361
> p_value
[1] 0.2457401
> qchisq(0.95,1)
[1] 3.841459
```

```
> # Fixed point iteration
> beta
[1] -0.38799702
> time1
Time difference of 0.056566 secs

> # Newton-Raphson iteration
> nbeta
[1] -0.3880057
> time2
Time difference of 0.06106305 secs
```

```
> #wald test
> #beta from Fixed point iteration
> I
[1] 8.619081
> Chi
[1] 1.297531
> p_value
[1] 0.2546647
```

```
> #LR test
> #beta from Fixed point iteration
> lb
[1] -103.2979
> l0
[1] -103.9453
> lambda
[1] 1.294698
> p_value
[1] 0.2546647

>
```

```
> Newton_bata
[1] -0.8281464 -0.3713842 -0.3880024 -0.3880057


> Fixedpoint_bata
[1]  0.51581373  0.15407233 -0.08544973 -0.22777721 -0.30590401 -0.34671324
-0.36744213 -0.37781673
[9] -0.38297001 -0.38552006 -0.38677954 -0.38740103 -0.38770756 -0.38785871
-0.38793323 -0.38796998
[17] -0.38798809 -0.38799702
```

- R-code

```
###data
timeA=c(1,3,3,6,7,7,10,12,14,15,18,19,22,26,28,29,34,40,48,49)
timeB=c(1,1,2,2,3,4,5,8,8,9,11,12,14,16,18,21,27,31,38,44)
eventA=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,0)
eventB=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,0,1)
xA=rep(1,20)
xB=rep(0,20)
time=c(timeA,timeB)
event=c(eventA,eventB)
X=c(xA,xB)
data=cbind(time,event,X)
data=as.data.frame(data[order(time),])
write.csv(data,"data.csv")
d1i=data[,"event"]*data[,"X"]
d2i=data[,"event"]*(1-data[,"X"])
di=data[,"event"]
n1=rep(1,nrow(data))
n1[1]=sum(data[,"X"])
for(idx in 2:nrow(data)){
  i=idx
  while(data[i,"time"]==data[i-1,"time"]){
    i=i-1
    if(i==1){break}
  }
```

```
  n1[idx]=sum(data[i:nrow(data),"X"])
}
n2=rep(1,nrow(data))
X2=1-data[,"X"]
n2[1]=sum(X2)
for(idx in 2:nrow(data)){
  i=idx
  while(data[i,"time"]==data[i-1,"time"]){
    i=i-1
    if(i==1){break}
  }
  n2[idx]=sum(X2[i:nrow(data)])
}
ni=n1+n2

#Kaplan-Meier
kplan=cbind(data[,"time"],d1i,d2i,n1,n2)
colnames(kplan)[1]="time"
kplan=as.data.frame(kplan)
k=kplan%>%group_by(time)%>%summarise(d1i=sum(d1i),d2i=sum(d2i),n
1=max(n1),n2=max(n2))
Ax=as.matrix(k["time"])
Ay=as.matrix(cumprod(1-(k[,"d1i"]/k[,"n1"])))
Ax=c(0,Ax)
Ay=c(1,Ay)
```

```
plot(Ax,Ay,type="s",xlab="t",ylab="S(t)",main="Kaplan-Meier",xaxt="n",y
axt="n",lwd=2)
Bx=as.matrix(k["time"])
By=as.matrix(cumprod(1-(k[,"d2i"]/k[,"n2"])))
Bx=c(0,Bx)
By=c(1,By)
par(new=T)
plot(Bx,By,type="s",xlab="t",ylab="S(t)",col="red",xaxt="n",yaxt="n",lwd
=2)
axis(1,seq(0,50,10))
axis(2,seq(0,1,0.2))

#scorce test
U0=sum(d1i-di*n1/ni)
I0=sum(di*n1*n2/(ni)^2)
Z=U0/(I0)^(1/2)
Chi=Z^2
p_value=1-pchisq(Chi,1)
qchisq(0.95,1)

#ties data
XX=cbind(data[,"time"],di,ni,n1,n2)
XX=subset(XX,di>0)
colnames(XX)[1]="time"
XX=as.data.frame(XX)
```

```
xx=XX%>%group_by(time)%>%summarise(di=
sum(di),ni=max(ni),n1=max(n1),n2=max(n2))
I00=sum(xx[,"di"]*(xx[,"ni"]-xx[,"di"])*xx[,"n1"]*xx[,"n2"]/(xx[,"ni"]^2*(
xx[,"ni"]-1)))
write.csv(xx,"ties.data.csv")


#scorce test2
Z=U0/(I00)^(1/2)
Chi=Z^2
p_value=1-pchisq(Chi,1)


# Fixed point iteration
start=Sys.time()
d=1
beta0=1
ebeta0=exp(beta0)
itern=0
Fixedpoint_bata=c()
while(d>10^-5){
  ebeta=sum(d1i)/sum(di*n1/(n1*ebeta0+n2))
  d=abs(log(ebeta)-log(ebeta0))
  ebeta0=ebeta
  Fixedpoint_bata=c(Fixedpoint_bata,log(ebeta))
  itern=itern+1
}
```

```
beta=log(ebeta)
end=Sys.time()
time1=end-start

# Newton-Raphson iteration
start=Sys.time()
d=1
nbeta0=1
itern=0
Newton_bata=c()
while(d>10^-5){
   U=sum(d1i-di*n1*exp(nbeta0)/(n1*exp(nbeta0)+n2))
   I=sum(di*(n1*n2*exp(nbeta0)/(n1*exp(nbeta0)+n2)^2))
   nbeta=nbeta0+I^(-1)*U
   d=abs(nbeta-nbeta0)
   nbeta0=nbeta
   Newton_bata=c(Newton_bata,nbeta)
   itern=itern+1
}
beta=nbeta
end=Sys.time()
time2=end-start

#wald test
I=sum(di*(n1*n2*exp(beta)/(n1*exp(beta)+n2)^2))
```

```
    Z=beta/(I)^(-1/2)
    Chi=Z^2
    p_value=1-pchisq(Chi,1)


    #LR test
    lb=sum(di*(beta*data[,"X"]-log(n1*exp(beta)+n2)))
    l0=sum(di*(0*data[,"X"]-log(n1*exp(0)+n2)))
    lambda=2*(lb-l0)
p_value=1-pchisq(lambda,1)
```