stat633 A2

2023-03-02

Question 1

part a)

```
library(survival)
library(KMsurv)
data(burn)
group = burn$Z1 # Treatment: 0-routine bathing 1-Body cleansing
survtime = burn$T3
indicator = burn$D3
fit = survfit(Surv(survtime, indicator)~group)
```

```
# One can check standard error using summary(fit)$std.err
print(summary(fit))
```

```
## Call: survfit(formula = Surv(survtime, indicator) ~ group)
##
##                 group=0
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1     70       1    0.986  0.0142        0.958        1.000
##     3     69       3    0.943  0.0277        0.890        0.999
##     4     66       4    0.886  0.0380        0.814        0.963
##     5     62       1    0.871  0.0400        0.796        0.953
##     6     60       2    0.842  0.0436        0.761        0.932
##     7     58       3    0.799  0.0481        0.710        0.899
##     8     53       1    0.784  0.0495        0.693        0.887
##     9     50       2    0.752  0.0522        0.657        0.862
##    10     47       1    0.736  0.0535        0.639        0.849
##    11     44       2    0.703  0.0561        0.601        0.822
##    13     40       1    0.685  0.0574        0.582        0.808
##    19     33       1    0.665  0.0593        0.558        0.791
##    21     30       1    0.642  0.0613        0.533        0.774
##    23     27       1    0.619  0.0635        0.506        0.756
##    32     18       1    0.584  0.0686        0.464        0.735
##    44      8       1    0.511  0.0910        0.361        0.725
##    47      6       1    0.426  0.1086        0.258        0.702
##    51      4       1    0.320  0.1231        0.150        0.680
##
##                 group=1
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     2     84       3    0.964  0.0202        0.925        1.000
##     3     81       1    0.952  0.0232        0.908        0.999
```

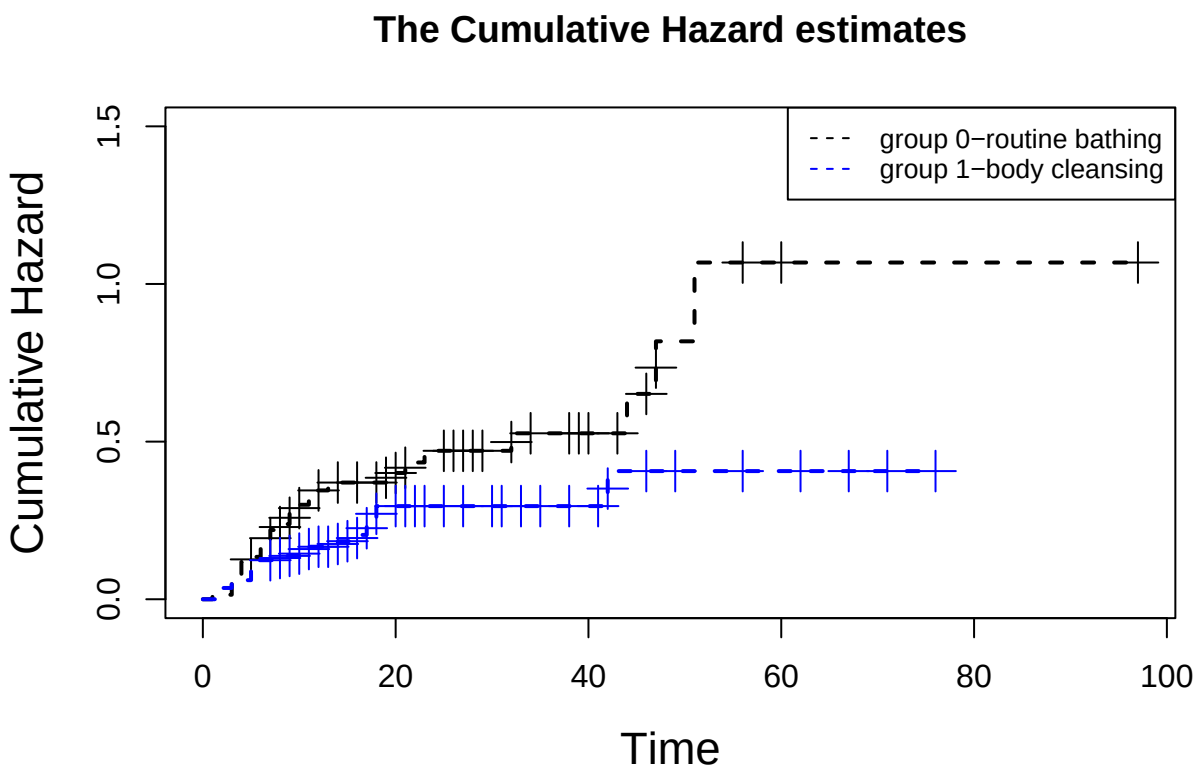
```
##      4      80        1    0.940  0.0258        0.891        0.992
##      5      79        5    0.881  0.0353        0.814        0.953
##      8      73        1    0.869  0.0369        0.800        0.944
##     10      70        1    0.856  0.0384        0.784        0.935
##     11      68        1    0.844  0.0398        0.769        0.926
##     14      57        1    0.829  0.0418        0.751        0.915
##     16      50        1    0.812  0.0441        0.730        0.904
##     17      47        2    0.778  0.0485        0.688        0.879
##     18      41        2    0.740  0.0531        0.643        0.852
##     42       9        1    0.658  0.0907        0.502        0.862
```

part b)

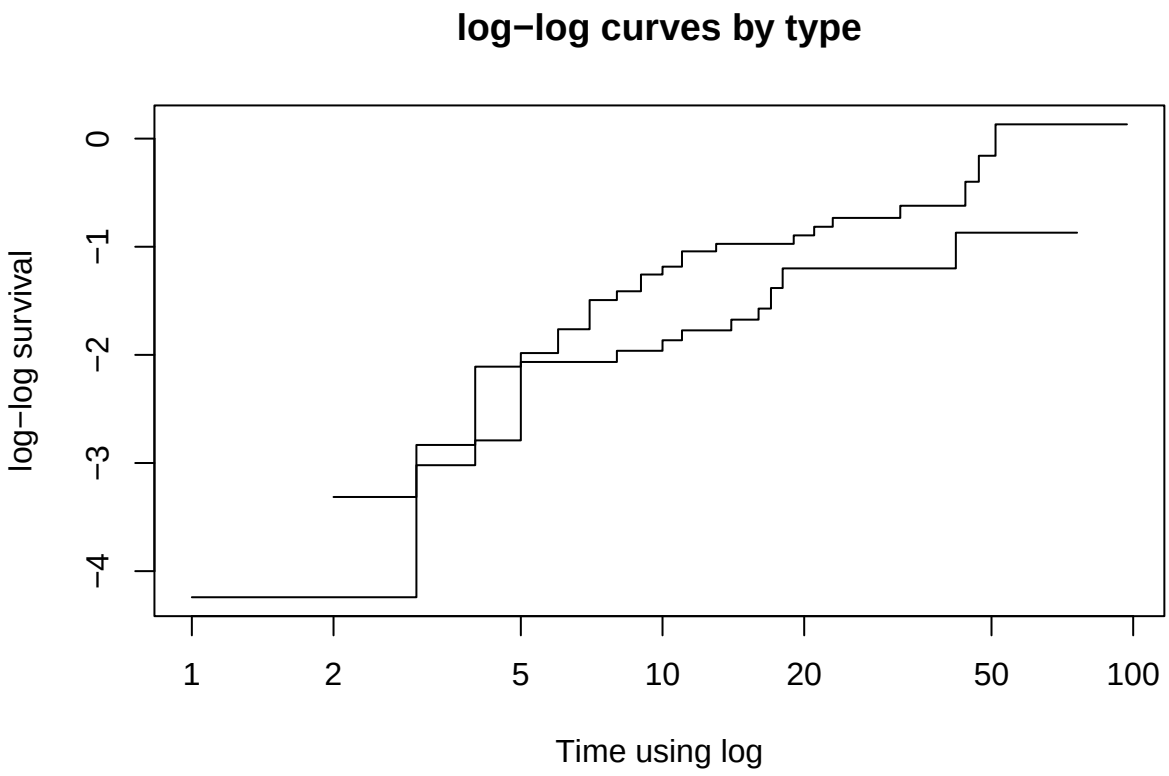
```
fitNA = survfit(Surv(survtime, indicator)~group, ctype = 1)
# One can check standard error using summary(fitNA)$std.err
summary(fitNA)
```

```
## Call: survfit(formula = Surv(survtime, indicator) ~ group, ctype = 1)
##
##                 group=0
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1     70       1    0.986  0.0142        0.958        1.000
##     3     69       3    0.943  0.0277        0.890        0.999
##     4     66       4    0.886  0.0380        0.814        0.963
##     5     62       1    0.871  0.0400        0.796        0.953
##     6     60       2    0.842  0.0436        0.761        0.932
##     7     58       3    0.799  0.0481        0.710        0.899
##     8     53       1    0.784  0.0495        0.693        0.887
##     9     50       2    0.752  0.0522        0.657        0.862
##    10     47       1    0.736  0.0535        0.639        0.849
##    11     44       2    0.703  0.0561        0.601        0.822
##    13     40       1    0.685  0.0574        0.582        0.808
##    19     33       1    0.665  0.0593        0.558        0.791
##    21     30       1    0.642  0.0613        0.533        0.774
##    23     27       1    0.619  0.0635        0.506        0.756
##    32     18       1    0.584  0.0686        0.464        0.735
##    44      8       1    0.511  0.0910        0.361        0.725
##    47      6       1    0.426  0.1086        0.258        0.702
##    51      4       1    0.320  0.1231        0.150        0.680
##
##                 group=1
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     2     84       3    0.964  0.0202        0.925        1.000
##     3     81       1    0.952  0.0232        0.908        0.999
##     4     80       1    0.940  0.0258        0.891        0.992
##     5     79       5    0.881  0.0353        0.814        0.953
##     8     73       1    0.869  0.0369        0.800        0.944
##    10     70       1    0.856  0.0384        0.784        0.935
##    11     68       1    0.844  0.0398        0.769        0.926
##    14     57       1    0.829  0.0418        0.751        0.915
##    16     50       1    0.812  0.0441        0.730        0.904
##    17     47       2    0.778  0.0485        0.688        0.879
##    18     41       2    0.740  0.0531        0.643        0.852
##    42      9       1    0.658  0.0907        0.502        0.862
```

```
plot(fitNA, lwd=2, lty=2, conf.int=FALSE, mark.time=TRUE, cex=2, cex.lab=1.4,
     cumhaz = TRUE, col=c("black", "blue"), xlab="Time", ylab="Cumulative Hazard",
     main="The Cumulative Hazard estimates", ylim=c(0,1.5))
legend("topright",c("group 0-routine bathing", "group 1-body cleansing"),lty=c(2),
       col=c("black", "blue"), cex=0.8)
```




```
# check the proportionality hazard assumption
plot(fit, fun = "cloglog", xlab = "Time using log", ylab = "log-log survival",
     main = "log-log curves by type")
```



Looking at the plot it seems that it seems that the proportional hazards assumption is not satisfied.

part c)

```
summary(fit)
```

```
## Call: survfit(formula = Surv(survtime, indicator) ~ group)
##
##                 group=0
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1     70       1    0.986  0.0142        0.958        1.000
##     3     69       3    0.943  0.0277        0.890        0.999
##     4     66       4    0.886  0.0380        0.814        0.963
##     5     62       1    0.871  0.0400        0.796        0.953
##     6     60       2    0.842  0.0436        0.761        0.932
##     7     58       3    0.799  0.0481        0.710        0.899
##     8     53       1    0.784  0.0495        0.693        0.887
##     9     50       2    0.752  0.0522        0.657        0.862
##    10     47       1    0.736  0.0535        0.639        0.849
##    11     44       2    0.703  0.0561        0.601        0.822
##    13     40       1    0.685  0.0574        0.582        0.808
##    19     33       1    0.665  0.0593        0.558        0.791
##    21     30       1    0.642  0.0613        0.533        0.774
##    23     27       1    0.619  0.0635        0.506        0.756
##    32     18       1    0.584  0.0686        0.464        0.735
##    44      8       1    0.511  0.0910        0.361        0.725
##    47      6       1    0.426  0.1086        0.258        0.702
##    51      4       1    0.320  0.1231        0.150        0.680
##
##                 group=1
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     2     84       3    0.964  0.0202        0.925        1.000
##     3     81       1    0.952  0.0232        0.908        0.999
##     4     80       1    0.940  0.0258        0.891        0.992
##     5     79       5    0.881  0.0353        0.814        0.953
##     8     73       1    0.869  0.0369        0.800        0.944
##    10     70       1    0.856  0.0384        0.784        0.935
##    11     68       1    0.844  0.0398        0.769        0.926
##    14     57       1    0.829  0.0418        0.751        0.915
##    16     50       1    0.812  0.0441        0.730        0.904
##    17     47       2    0.778  0.0485        0.688        0.879
##    18     41       2    0.740  0.0531        0.643        0.852
##    42      9       1    0.658  0.0907        0.502        0.862
```

Note: the estimate of the median time to infection for group 1-body cleansing with chlorhexidine is not defined/observed, so only estimate of the median time to infection for group 0-routine bathing with povidone-iodine is shown below, as well as 95% confidence interval

Linear:

```
fit.km.burn.routine = survfit(Surv(survtime, indicator)~1,
                              conf.type="plain", data = burn, subset=(group=="0"))
print(fit.km.burn.routine)
```

```
## Call: survfit(formula = Surv(survtime, indicator) ~ 1, data = burn,
```

```
##     subset = (group == "0"), conf.type = "plain")
##
##       n events median 0.95LCL 0.95UCL
## [1,] 70     28     47      23      NA
```

Log transformation:

```
fit.km.burn.routine = survfit(Surv(survtime, indicator)~1,
                              conf.type="log-log", data = burn, subset=(group=="0"))
print(fit.km.burn.routine)
```

```
## Call: survfit(formula = Surv(survtime, indicator) ~ 1, data = burn,
##     subset = (group == "0"), conf.type = "log-log")
##
##       n events median 0.95LCL 0.95UCL
## [1,] 70     28     47      23      NA
```

Arcsine:

```
fit.km.burn.routine = survfit(Surv(survtime, indicator)~1,
                              conf.type="arcsin", data = burn, subset=(group=="0"))
print(fit.km.burn.routine)
```

```
## Call: survfit(formula = Surv(survtime, indicator) ~ 1, data = burn,
##     subset = (group == "0"), conf.type = "arcsin")
##
##       n events median 0.95LCL 0.95UCL
## [1,] 70     28     47      23      NA
```

part d)

```
alpha=0.05
critical.Z=qnorm(1-alpha/2)
KMPCIestimator <- function (Z,delta){
  UZ<-unique(Z) #Unique observed times;
  N<-length(UZ)
  UZ.order<-order(UZ)
  UZ<-UZ[UZ.order] #sort data;
  KM <- rep(0,N)
  Y<-rep(0,N)
  D<-rep(0,N)
  D[1]<-sum(Z[delta==1]==UZ[1])
  Y[1]<-sum(Z >= UZ[1])
  KM[1] <- 1-D[1]/Y[1] #this is for right continuous value
  for (i in 2: N){
    D[i]<-sum(Z[delta==1]==UZ[i])
    Y[i]<-sum(Z >= UZ[i])
    KM[i] <- KM[i-1]*(1-D[i]/Y[i])
  }
  # Calculate variance and standard error;
  sigma2.s<-rep(0,N)
  for (i in 1: N){
    ## sigma2.s[i]<-sum( (UZ<=UZ[i])*(D/(Y*(Y-D))) ) #old version
    sigma2.s[i]<-sum( (UZ[1:i]<=UZ[i])*(D[1:i]/(Y[1:i]*(Y[1:i]-D[1:i]))))
    ## Note the data is sorted by UZ;
    ## Using this to avoid NaN for times smaller than the largest observation;
  }
  KM.var<-KM^2*sigma2.s
  KM.se<-sqrt(KM.var)
  sigma.s<-sqrt(sigma2.s)
  # 100(1-alpha)% linear confidence interval;
  ###according to lecture note in chapter 4 p36
  linearL<-KM-critical.Z*sigma.s*KM
  linearU<-KM+critical.Z*sigma.s*KM
  # 100(1-alpha)% log-transformed confidence interval;
  theta<-exp((critical.Z*sigma.s)/log(KM))
  ###according to lecture note in chapter 4 p37
  logtransL<-KM^(1/theta)
  logtransU<-KM^theta
  # 100(1-alpha)% arcsine-square-root-transformed confidence interval;
  ###according to lecture note in chapter 4 p38
  ass<-asin(KM^(1/2))
  zss<-critical.Z*sigma.s*(KM/(1-KM))^(1/2)
  azssL<-ass-0.5*zss
  azssU<-ass+0.5*zss
  asrtransL<-(sin(pmax(0,azssL)))^2
  asrtransU<-(sin(pmin(pi/2,azssU)))^2
  Full<-data.frame(UZ,D,Y,KM,sigma2.s,KM.var,KM.se)
  Reduced<-subset(Full, (Full$D>0))
  PCI.full<-data.frame(UZ,D,linearL, linearU, logtransL, logtransU, asrtransL, asrtransU)
  PCI.reduced<-subset(PCI.full, (PCI.full$D>0))
  list(Full, Reduced, PCI.full, PCI.reduced)
```

```
}
```

```
# for Group 0-routine bathing
group = burn$Z1 # Treatment: 0-routine bathing 1-Body cleansing
workdata = burn[burn$Z1==0,]
survtime = workdata$T3
indicator = workdata$D3
KMPCI.est<-KMPCIestimator(survtime, indicator)
subset(KMPCI.est[[3]], (KMPCI.est[[3]]$UZ == 10))
```

```
##   UZ D   linearL   linearU logtransL logtransU asrtransL asrtransU
## 9 10 1 0.6314903 0.8412695 0.6142199 0.8252047 0.6258115 0.8336148
```

```
# for Group 1-Body cleansing
group = burn$Z1 # Treatment: 0-routine bathing 1-Body cleansing
workdata = burn[burn$Z1==1,]
survtime = workdata$T3
indicator = workdata$D3
KMPCI.est<-KMPCIestimator(survtime, indicator)
subset(KMPCI.est[[3]], (KMPCI.est[[3]]$UZ == 10))
```

```
##   UZ D   linearL   linearU logtransL logtransU asrtransL asrtransU
## 8 10 1 0.7812868 0.9316571 0.7610676 0.9158361 0.7736969 0.9229174
```

part e)

For Group 0-routine bathing:

1. Recall distinct event times at $t = 1, 3, 4, ...8, ..19, 21, ..., 51$
2. So $\hat{\sigma_s}^2(8) = \hat{\sigma_s}^2(8)$ and $\hat{\sigma_s}^2(20) = \hat{\sigma_s}^2(19)$
3. need to plugin values into this equation: $\frac{\hat{V}(\hat{S}(8))}{\hat{S}(8)^2}$ and $\frac{\hat{V}(\hat{S}(19))}{\hat{S}(19)^2}$
4. Note that the calculation of the entries is very similar to the calculations performed for the 95% pointwise confidence intervals (recall variance of KM estimator is estimated using Greenwood's formula). We get $\frac{\hat{V}(\hat{S}(8))}{\hat{S}(8)^2} = \frac{0.0024482102}{0.7837376^2} = 0.003986$ and $\frac{\hat{V}(\hat{S}(19))}{\hat{S}(19)^2} = \frac{0.0035119308}{0.6645676^2} = 0.007952$
5. Lower bound: $a_L = \frac{n \times \hat{\sigma_s}^2(t_L)}{1+n \times \hat{\sigma_s}^2(t_L)}$ and we get $\frac{70 \times 0.003986}{1+70 \times 0.003986} = 0.2181514$
6. Upper bound: $a_U = \frac{n \times \hat{\sigma_s}^2(t_U)}{1+n \times \hat{\sigma_s}^2(t_U)}$ and we get $\frac{70 \times 0.007952}{1+70 \times 0.007952} = 0.3575907$
7. Look up values for $C_{0.05}(0.22, 0.36) = 2.5272$

```
# for Group 0-routine bathing
group = burn$Z1 # Treatment: 0-routine bathing 1-Body cleansing
workdata = burn[burn$Z1==0,]
survtime = workdata$T3
indicator = workdata$D3

KMestimator <- function (Z,delta)
{UZ<-unique(Z) #Unique observed times;
N<-length(UZ)
UZ.order<-order(UZ)
UZ<-UZ[UZ.order] #sort data;
KM <- rep(0,N)
Y<-rep(0,N)
D<-rep(0,N)
D[1]<-sum(Z[delta==1]==UZ[1]) #number of events at t1
Y[1]<-sum(Z >= UZ[1]) #number at risk just before t1
KM[1] <- 1-D[1]/Y[1] #this is for right continuous value
for (i in 2: N){
D[i]<-sum(Z[delta==1]==UZ[i])
Y[i]<-sum(Z >= UZ[i])
KM[i] <- KM[i-1]*(1-D[i]/Y[i])
}
# Calculate variance and sdandard error;
sigma2.s<-rep(0,N)
for (i in 1: N){
## sigma2.s[i]<-sum( (UZ<=UZ[i])*(D/(Y*(Y-D))) ) #old version
sigma2.s[i]<-sum( (UZ[1:i]<=UZ[i])*(D[1:i]/(Y[1:i]*(Y[1:i]-D[1:i]))))
## Note the data is sorted by UZ;
## Using this to avoid NaN for times smaller than the largest observation;
}
KM.var<-KM^2*sigma2.s #V(S(t))=S(t)^2*sum(d/(Y*(Y-d)))
KM.se<-sqrt(KM.var)
Full<-data.frame(UZ,D,Y,KM,sigma2.s,KM.var,KM.se) #full dataset
Reduced<-subset(Full, (Full$D>0)) #exclude censored observations
```

```
list(Full, Reduced)
}
KM.est<-KMestimator(survtime, indicator)
```

```
##according to lecture note in chapter 4 p50
sigma = sqrt(KM.est[[2]]$sigma2.s)
surv=KM.est[[2]]$KM
md = 2.5272*sigma
####Note: the CI limits should be restricted between -1 and 1;
linearL = surv - md * surv
linearU = surv + md * surv
linearL=linearL*(-1<=linearL)+(-1)*(-1>linearL)
linearU=1*(1<linearU)+linearU*(linearU<=1)
logL = surv^( 1/(exp(md/log(surv))))
logU = surv^(exp(md/(log(surv))))
logL=logL*(-1<=logL)+(-1)*(-1>logL)
logU=1*(1<logU)+logU*(logU<=1)
sinL = (sin(pmax(0, asin(surv^.5)-.5 * md *(surv/(1-surv))^.5)))^2
sinU = (sin(pmin(pi/2, asin(surv^.5) + .5 * md *(surv/(1-surv))^.5)))^2
sinL=sinL*(-1<=sinL)+(-1)*(-1>sinL)
sinU=1*(1<sinU)+sinU*(sinU<=1)
#ConfBand<-data.frame(KM.est[[2]]$UZ, KM.est[[2]]$KM,
#KM.est[[2]]$KM.se,KM.est[[2]]$sigma2.s,linearL,linearU,logL,logU,sinL,sinU)
ConfBand<-data.frame(KM.est[[2]]$UZ, KM.est[[2]]$KM,linearL,linearU,logL,logU,sinL,sinU)
##100(1-alpha)% confidence bands of S(x): [L(t),U(t)] where
##1-alpha=P(L(t)<S(t)<U(t),for all tL<t<tU)
ConfBand<-subset(ConfBand,(8<=KM.est[[2]]$UZ & KM.est[[2]]$UZ<=20))
ConfBand<-as.matrix(ConfBand)
cat("95% EP Confidence Band:","\n")
```

```
## 95% EP Confidence Band:
```

```
print(ConfBand)
```

```
##    KM.est..2...UZ KM.est..2...KM   linearL   linearU      logL      logU
## 7               8      0.7837376 0.6586933 0.9087820 0.6256272 0.8810768
## 8               9      0.7523881 0.6203916 0.8843847 0.5903155 0.8576481
## 9              10      0.7363799 0.6011340 0.8716257 0.5725300 0.8454301
## 10             11      0.7029081 0.5611989 0.8446172 0.5355086 0.8195582
## 11             13      0.6853354 0.5403772 0.8302936 0.5161560 0.8058357
## 12             19      0.6645676 0.5148019 0.8143334 0.4919829 0.7902577
##         sinL      sinU
## 7  0.6476197 0.8938808
## 8  0.6107292 0.8706267
## 9  0.5921987 0.8584625
## 10 0.5537811 0.8326788
## 11 0.5337634 0.8189945
## 12 0.5090907 0.8036243
```

For Group 1-Body cleansing:

```r
# for Group 0-routine bathing
group = burn$Z1 # Treatment: 0-routine bathing 1-Body cleansing
workdata = burn[burn$Z1==1,]
survtime = workdata$T3
indicator = workdata$D3

KMestimator <- function (Z,delta)
{UZ<-unique(Z) #Unique observed times;
N<-length(UZ)
UZ.order<-order(UZ)
UZ<-UZ[UZ.order] #sort data;
KM <- rep(0,N)
Y<-rep(0,N)
D<-rep(0,N)
D[1]<-sum(Z[delta==1]==UZ[1]) #number of events at t1
Y[1]<-sum(Z >= UZ[1]) #number at risk just before t1
KM[1] <- 1-D[1]/Y[1] #this is for right continuous value
for (i in 2: N){
D[i]<-sum(Z[delta==1]==UZ[i])
Y[i]<-sum(Z >= UZ[i])
KM[i] <- KM[i-1]*(1-D[i]/Y[i])
}
# Calculate variance and sdandard error;
sigma2.s<-rep(0,N)
for (i in 1: N){
## sigma2.s[i]<-sum( (UZ<=UZ[i])*(D/(Y*(Y-D))) ) #old version
sigma2.s[i]<-sum( (UZ[1:i]<=UZ[i])*(D[1:i]/(Y[1:i]*(Y[1:i]-D[1:i]))))
## Note the data is sorted by UZ;
## Using this to avoid NaN for times smaller than the largest observation;
}
KM.var<-KM^2*sigma2.s #V(S(t))=S(t)^2*sum(d/(Y*(Y-d)))
KM.se<-sqrt(KM.var)
Full<-data.frame(UZ,D,Y,KM,sigma2.s,KM.var,KM.se) #full dataset
Reduced<-subset(Full, (Full$D>0)) #exclude censored observations
list(Full, Reduced)
}
KM.est<-KMestimator(survtime, indicator)
```

1. Recall distinct event times at $t = 2, 3, 4, ...8, ..18, 42$
2. So $\hat{\sigma_s}^2(8) = \hat{\sigma_s}^2(8)$ and $\hat{\sigma_s}^2(20) = \hat{\sigma_s}^2(18)$
3. need to plugin values into this equation: $\frac{\hat{V}(\hat{S}(8))}{\hat{S}(8)^2}$ and $\frac{\hat{V}(\hat{S}(18))}{\hat{S}(18)^2}$
4. Note that the calculation of the entries is very similar to the calculations performed for the 95% pointwise confidence intervals (recall variance of KM estimator is estimated using Greenwood's formula). We get $\frac{\hat{V}(\hat{S}(8))}{\hat{S}(8)^2} = \frac{0.0013581815}{0.86888452^2} = 0.001799$ and $\frac{\hat{V}(\hat{S}(18))}{\hat{S}(18)^2} = \frac{0.0028162890}{0.7399692^2} = 0.005143$
5. Lower bound: $a_L = \frac{n \times \hat{\sigma_s}^2(t_L)}{1+n \times \hat{\sigma_s}^2(t_L)}$ and we get $\frac{84 \times 0.001799}{1+84 \times 0.001799} = 0.1312778$
6. Upper bound: $a_U = \frac{n \times \hat{\sigma_s}^2(t_U)}{1+n \times \hat{\sigma_s}^2(t_U)}$ and we get $\frac{84 \times 0.005143}{1+84 \times 0.005143} = 0.3016818$
7. Look up values for $C_{0.05}(0.13, 0.30) = 2.6210$

```
##according to lecture note in chapter 4 p50
sigma = sqrt(KM.est[[2]]$sigma2.s)
surv=KM.est[[2]]$KM
md = 2.621*sigma
####Note: the CI limits should be restricted between -1 and 1;
linearL = surv - md * surv
linearU = surv + md * surv
linearL=linearL*(-1<=linearL)+(-1)*(-1>linearL)
linearU=1*(1<linearU)+linearU*(linearU<=1)
logL = surv^( 1/(exp(md/log(surv))))
logU = surv^(exp(md/(log(surv))))
logL=logL*(-1<=logL)+(-1)*(-1>logL)
logU=1*(1<logU)+logU*(logU<=1)
sinL = (sin(pmax(0, asin(surv^.5)-.5 * md *(surv/(1-surv))^.5)))^2
sinU = (sin(pmin(pi/2, asin(surv^.5) + .5 * md *(surv/(1-surv))^.5)))^2
sinL=sinL*(-1<=sinL)+(-1)*(-1>sinL)
sinU=1*(1<sinU)+sinU*(sinU<=1)
#ConfBand<-data.frame(KM.est[[2]]$UZ, KM.est[[2]]$KM,
#KM.est[[2]]$KM.se,KM.est[[2]]$sigma2.s,linearL,linearU,logL,logU,sinL,sinU)
ConfBand<-data.frame(KM.est[[2]]$UZ, KM.est[[2]]$KM,linearL,linearU,logL,logU,sinL,sinU)
##100(1-alpha)% confidence bands of S(x): [L(t),U(t)] where
##1-alpha=P(L(t)<S(t)<U(t),for all tL<t<tU)
ConfBand<-subset(ConfBand,(8<=KM.est[[2]]$UZ & KM.est[[2]]$UZ<=20))
ConfBand<-as.matrix(ConfBand)
cat("95% EP Confidence Band:","\n")
```

```
## 95% EP Confidence Band:
```

```
print(ConfBand)
```

```
##    KM.est..2...UZ KM.est..2...KM   linearL   linearU      logL      logU
## 5               8      0.8688845 0.7722915 0.9654776 0.7334604 0.9382650
## 6              10      0.8564719 0.7559291 0.9570147 0.7185450 0.9299496
## 7              11      0.8438767 0.7395336 0.9482198 0.7035023 0.9213333
## 8              14      0.8290719 0.7195816 0.9385621 0.6844157 0.9115005
## 9              16      0.8124904 0.6968860 0.9280948 0.6623125 0.9006368
## 10             17      0.7779164 0.6507066 0.9051262 0.6177859 0.8772577
## 11             18      0.7399692 0.6008761 0.8790623 0.5699776 0.8510167
##         sinL      sinU
## 5  0.7586018 0.9491618
## 6  0.7427445 0.9410854
## 7  0.7268537 0.9326775
## 8  0.7072977 0.9232038
## 9  0.6849626 0.9128059
## 10 0.6397685 0.8902349
## 11 0.5912319 0.8647797
```

part f)

Based on previous results, especially the Cumulative Hazard v.s. Time Plot, it is clear to observe that the Cumulative Hazard for group 0 (routine bathing) is almost always higher than the Cumulative Hazard for group 1 (body cleansing) for any given time. This shows that group 0 has a higher risk of infection than group 1. This means that individuals in group 0 are experiencing a higher rate of infection than those in group 1. This indicates that the survival function of group 1 is higher than the survival function of group 0 since $H(t) = -log[S(t)]$. Therefore, the higher cumulative hazard for group 0 indicates a lower survival probability compared to group 1.

Question 2

```
Year.Interval = c("[0,2)","[2,4)","[4,6)","[6,8)","[8,10)","[10,12)","[12,14)")
Number.of.Events = c(2,1,4,3,2,2,3)
Number.of.Losts = c(3,2,8,10,18,21,21)

risk = c()
for (i in 2:7) {
  risk[1] = 100
  risk[i] = risk[i-1]-Number.of.Events[i-1]-Number.of.Losts[i-1]
}
Number.of.Risks = risk

Adj.Number.of.Risks = c()
for (i in 1:7) {
  Adj.Number.of.Risks[i] = Number.of.Risks[i]-Number.of.Losts[i]/2
}

p = c()
for (i in 2:7) {
  p[1] = 1 - Number.of.Events[1]/Adj.Number.of.Risks[1]
  p[i] = p[i-1]*(1-Number.of.Events[i]/Adj.Number.of.Risks[i])
}

life.table = data.frame(Year.Interval,Number.of.Events,Number.of.Losts,Number.of.Risks,
                        Adj.Number.of.Risks,p)
knitr::kable(life.table, "pipe",
             col.names = c("Year Interval", "# of HIV-Postive",
                           "# Lost to Follow-up", "# of At-risk",
                           "Adjusted # of At-risk",
                           "Probability of Individual Survives Beyond this Interval"))
```

Year Interval	# of HIV-Postive	# Lost to Follow-up	# of At-risk	Adjusted # of At-risk	Probability of Individual Survives Beyond this Interval
[0,2)	2	3	100	98.5	0.9796954
[2,4)	1	2	95	94.0	0.9692731
[4,6)	4	8	92	88.0	0.9252153
[6,8)	3	10	80	75.0	0.8882067
[8,10)	2	18	67	58.0	0.8575788
[10,12)	2	21	47	36.5	0.8105882
[12,14)	3	21	24	13.5	0.6304575

```
# use lifetab function
tis = c(0,2,4,6,8,10,12,14)
Number.of.Events = c(2,1,4,3,2,2,3)
Number.of.Losts = c(3,2,8,10,18,21,21)
knitr::kable(data.frame(lifetab(tis, 100, Number.of.Losts, Number.of.Events)), "latex")
```

	nsubs	nlost	nrisk	nevent	surv	pdf	hazard	se.surv	se.pdf	se.hazard
0-2	100	3	98.5	2	1.0000000	0.0101523	0.0102564	0.0000000	0.0071055	0.0072520
2-4	95	2	94.0	1	0.9796954	0.0052111	0.0053476	0.0142110	0.0051839	0.0053475
4-6	92	8	88.0	4	0.9692731	0.0220289	0.0232558	0.0174685	0.0107685	0.0116248
6-8	80	10	75.0	3	0.9252153	0.0185043	0.0204082	0.0272260	0.0104818	0.0117802
8-10	67	18	58.0	2	0.8882067	0.0153139	0.0175439	0.0334876	0.0106559	0.0124035
10-12	47	21	36.5	2	0.8575788	0.0234953	0.0281690	0.0387076	0.0161869	0.0199106
12-14	24	21	13.5	3	0.8105882	NA	NA	0.0488072	NA	NA

Question 3

part a)

$H_0 : h_0(t) = h_1(t) = h(t)$ for all $t < \tau$ vs.$H_a : h_0(t) \neq h_1(t)$ for some $t <= \tau$

where $h_0(t)$ is the hazard rate for group 0 (routine bathing) and $h_1(t)$ is the hazard rate for group 1 (body cleansing) and τ is the last observation time. Also note when we use logrank test, $W_j = 1$

```
library(survival)
library(KMsurv)
data(burn)
group = burn$Z1 # Treatment: 0-routine bathing 1-Body cleansing
survtime = burn$T3
indicator = burn$D3
```

```
# log-rank test using survdiff()
logrank.test = survdiff(Surv(survtime, indicator)~group, rho = 0, data = burn)
logrank.test
```

```
## Call:
## survdiff(formula = Surv(survtime, indicator) ~ group, data = burn,
##     rho = 0)
##
##           N Observed Expected (O-E)^2/E (O-E)^2/V
## group=0 70       28     21.4      2.07      3.79
## group=1 84       20     26.6      1.66      3.79
##
##  Chisq= 3.8  on 1 degrees of freedom, p= 0.05
```

```
logrank.test$chisq
```

```
## [1] 3.792429
```

```
#Self-written function for Log-rank Test and Gehan's Test

# Function for calculations of event numbers and risk sets;
# test=1-> Log-rank, 2-> Gehan, 3-> Fleming-Harrington, p=1, q=1, 4-> Peto;
#(Z, delta) represents all observations in the dataset;

RiskSet <- function (Z,delta,Z1,delta1, test) #(Z1,delta1) represent obs. in a group;
{
  UZ<-unique(Z[delta==1]) #Unique observed times;
  N<-length(UZ)
  UZ.order<-order(UZ)
  UZ<-UZ[UZ.order] #sort data;
  KM <- rep(0,N)
  KM.tilde<-rep(0,N) #Used in Peto-Peto weights;
  Y<-rep(0,N)
  D<-rep(0,N)
  Y1<-rep(0,N)
  D1<-rep(0,N)
```

```
YY<-rep(0,N)
for (i in 1 : N){
D[i]<-sum(Z[delta==1]==UZ[i])
Y[i]<-sum(Z >= UZ[i])
D1[i]<-sum(Z1[delta1==1]==UZ[i])
Y1[i]<-sum(Z1 >= UZ[i])
YY[i]<-Y[i]+1 #Used in Peto-Peto weights;
if (i==1){
KM[i]=1
KM.tilde[i]=1-D[1]/YY[1]
}
else{
KM[i]<-KM[i-1]*(1-D[i-1]/Y[i-1]) #left continuous version of K-M estimator for Fleiming-KM.tilde[i]<-
}
}
D2<-D-D1
Y2<-Y-Y1
Y1DY<-Y1*(D/Y)
Y2DY<-Y2*(D/Y)
DY1DY<-D1-Y1DY
YDY=((Y-D)/((Y-1)*(Y!=1)+1*(Y==1)))*(Y!=1)+ 0*(Y==1)
#This means if Y=1, (Y-D)/(Y-1)=0, which will not affect results;
Y1YYYD<-(Y1/Y)*(1-Y1/Y)*YDY*D
print(UZ)
dyTable<-data.frame(UZ, Y1, D1, Y2, D2, Y, D, Y1DY, DY1DY, Y1YYYD, KM, KM.tilde)
# Change test method here, e.g. 1->log-rank test, 2->Gehan test;
if (test==1){
  W=1
cat("The test is Log-rank test:", "\n")
}
if (test==2){
W<- Y
cat("The test is Gehan test:", "\n")
}
if (test==3){
W<- KM*(1-KM) # KM*(1-KM)-> Fleming-Harrinnton test with p=1 and q=1;
cat("The test is Fleming-Harrinnton test with p=1 and q=1:", "\n")
}
if (test==4){
W<- KM.tilde
cat("The test is Peto-Peto test:", "\n")
}
#Calculate Z(tau) values;
Z1Tau<-sum(W*(D1-Y1DY))
Z2Tau<-sum(W*(D2-Y2DY))
ZTau<-matrix(c(Z1Tau,Z2Tau),2,1)
# Calculate variance and covariance;
sigma11<-sum(W^2*(Y1/Y)*(1-Y1/Y)*YDY*D)
sigma22<-sum(W^2*(Y2/Y)*(1-Y2/Y)*YDY*D)
sigma12<--sum(W^2*(Y1/Y)*(Y2/Y)*YDY*D)
sigma21<-sigma12
varmatrix<-matrix(c(sigma11,sigma12, sigma21, sigma22), 2,2)
# Calculate Chi-Squred statistic and p-value using Z_1(tau);
```

```
  chiSq<-Z1Tau^2/sigma11
  pval<-1-pchisq(chiSq, 1)
  testval<-c(chiSq,pval)
  list(dyTable, ZTau, varmatrix, testval)
}
```

```
Z<-burn$T3
delta<-burn$D3
Z1<-burn$T3[burn$Z1==0] # 0-routine bathing 1-Body cleansing
delta1<-burn$D3[burn$Z1==0]

# Set a value for test for a specific test;
# test=1-> Log-rank, 2-> Gehan, 3-> Fleming-Harrington, p=1, q=1, 4-> Peto-Peto;
test=1 #Change test value here to get different weights;
Results.Logrank<-RiskSet(Z,delta,Z1,delta1, test)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 13 14 16 17 18 19 21 23 32 42 44 47 51
## The test is Log-rank test:
```

```
Results.Logrank
```

```
## [[1]]
##    UZ Y1 D1 Y2 D2   Y D      Y1DY      DY1DY    Y1YYYD        KM  KM.tilde
## 1   1 70  1 84  0 154 1 0.4545455  0.5454545 0.2479339 1.0000000 0.9935484
## 2   2 69  0 84  3 153 3 1.3529412 -1.3529412 0.7330177 0.9935065 0.0000000
## 3   3 69  3 81  1 150 4 1.8400000  1.1600000 0.9735946 0.9740260 0.0000000
## 4   4 66  4 80  1 146 5 2.2602740  1.7397260 1.2043406 0.9480519 0.0000000
## 5   5 62  1 79  5 141 6 2.6382979 -1.6382979 1.4254026 0.9155844 0.0000000
## 6   6 60  2 74  0 134 2 0.8955224  1.1044776 0.4908239 0.8766234 0.0000000
## 7   7 58  3 74  0 132 3 1.3181818  1.6818182 0.7276986 0.8635394 0.0000000
## 8   8 53  1 73  1 126 2 0.8412698  0.1587302 0.4835031 0.8439135 0.0000000
## 9   9 50  2 71  0 121 2 0.8264463  1.1735537 0.4808984 0.8305181 0.0000000
## 10 10 47  1 70  1 117 2 0.8034188  0.1965812 0.4765341 0.8167905 0.0000000
## 11 11 44  2 68  1 112 3 1.1785714  0.8214286 0.7026682 0.8028283 0.0000000
## 12 13 40  1 62  0 102 1 0.3921569  0.6078431 0.2383699 0.7813240 0.0000000
## 13 14 39  0 57  1  96 1 0.4062500 -0.4062500 0.2412109 0.7736639 0.0000000
## 14 16 38  0 50  1  88 1 0.4318182 -0.4318182 0.2453512 0.7656049 0.0000000
## 15 17 35  0 47  2  82 2 0.8536585 -0.8536585 0.4832514 0.7569049 0.0000000
## 16 18 35  0 41  2  76 2 0.9210526 -0.9210526 0.4902585 0.7384438 0.0000000
## 17 19 33  1 37  0  70 1 0.4714286  0.5285714 0.2491837 0.7190110 0.0000000
## 18 21 30  1 35  0  65 1 0.4615385  0.5384615 0.2485207 0.7087395 0.0000000
## 19 23 27  1 28  0  55 1 0.4909091  0.5090909 0.2499174 0.6978358 0.0000000
## 20 32 18  1 17  0  35 1 0.5142857  0.4857143 0.2497959 0.6851479 0.0000000
## 21 42 10  0  9  1  19 1 0.5263158 -0.5263158 0.2493075 0.6655722 0.0000000
## 22 44  8  1  7  0  15 1 0.5333333  0.4666667 0.2488889 0.6305421 0.0000000
## 23 47  6  1  6  0  12 1 0.5000000  0.5000000 0.2500000 0.5885059 0.0000000
## 24 51  4  1  5  0   9 1 0.4444444  0.5555556 0.2469136 0.5394638 0.0000000
##
## [[2]]
##           [,1]
## [1,]  6.643339
## [2,] -6.643339
```

```
##
## [[3]]
##           [,1]      [,2]
## [1,]  11.63739 -11.63739
## [2,] -11.63739  11.63739
##
## [[4]]
## [1] 3.79242901 0.05148488
```

Since the p-value is 0.05148488, which is slightly greater than $\alpha = 0.05$, we do not reject the null hypothesis, showing that the hazard rates of group 1 and group 0 are the same at 95% significance level.

part b)

$H_0 : h_0(t) = h_1(t) = h(t)$ for all $t < \tau$ vs.$H_a : h_0(t) \neq h_1(t)$ for some $t <= \tau$

where $h_0(t)$ is the hazard rate for group 0 (routine bathing) and $h_1(t)$ is the hazard rate for group 1 (body cleansing) and τ is the last observation time. Also note when we use logrank test, $W_j = Y_j$

```
test=2 #Change test value here to get different weights;
Results.Gehan<-RiskSet(Z,delta,Z1,delta1, test)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 13 14 16 17 18 19 21 23 32 42 44 47 51
## The test is Gehan test:
```

```
Results.Gehan
```

```
## [[1]]
##    UZ Y1 D1 Y2 D2   Y D      Y1DY      DY1DY    Y1YYYD        KM  KM.tilde
## 1   1 70  1 84  0 154 1 0.4545455  0.5454545 0.2479339 1.0000000 0.9935484
## 2   2 69  0 84  3 153 3 1.3529412 -1.3529412 0.7330177 0.9935065 0.0000000
## 3   3 69  3 81  1 150 4 1.8400000  1.1600000 0.9735946 0.9740260 0.0000000
## 4   4 66  4 80  1 146 5 2.2602740  1.7397260 1.2043406 0.9480519 0.0000000
## 5   5 62  1 79  5 141 6 2.6382979 -1.6382979 1.4254026 0.9155844 0.0000000
## 6   6 60  2 74  0 134 2 0.8955224  1.1044776 0.4908239 0.8766234 0.0000000
## 7   7 58  3 74  0 132 3 1.3181818  1.6818182 0.7276986 0.8635394 0.0000000
## 8   8 53  1 73  1 126 2 0.8412698  0.1587302 0.4835031 0.8439135 0.0000000
## 9   9 50  2 71  0 121 2 0.8264463  1.1735537 0.4808984 0.8305181 0.0000000
## 10 10 47  1 70  1 117 2 0.8034188  0.1965812 0.4765341 0.8167905 0.0000000
## 11 11 44  2 68  1 112 3 1.1785714  0.8214286 0.7026682 0.8028283 0.0000000
## 12 13 40  1 62  0 102 1 0.3921569  0.6078431 0.2383699 0.7813240 0.0000000
## 13 14 39  0 57  1  96 1 0.4062500 -0.4062500 0.2412109 0.7736639 0.0000000
## 14 16 38  0 50  1  88 1 0.4318182 -0.4318182 0.2453512 0.7656049 0.0000000
## 15 17 35  0 47  2  82 2 0.8536585 -0.8536585 0.4832514 0.7569049 0.0000000
## 16 18 35  0 41  2  76 2 0.9210526 -0.9210526 0.4902585 0.7384438 0.0000000
## 17 19 33  1 37  0  70 1 0.4714286  0.5285714 0.2491837 0.7190110 0.0000000
## 18 21 30  1 35  0  65 1 0.4615385  0.5384615 0.2485207 0.7087395 0.0000000
## 19 23 27  1 28  0  55 1 0.4909091  0.5090909 0.2499174 0.6978358 0.0000000
## 20 32 18  1 17  0  35 1 0.5142857  0.4857143 0.2497959 0.6851479 0.0000000
## 21 42 10  0  9  1  19 1 0.5263158 -0.5263158 0.2493075 0.6655722 0.0000000
## 22 44  8  1  7  0  15 1 0.5333333  0.4666667 0.2488889 0.6305421 0.0000000
## 23 47  6  1  6  0  12 1 0.5000000  0.5000000 0.2500000 0.5885059 0.0000000
## 24 51  4  1  5  0   9 1 0.4444444  0.5555556 0.2469136 0.5394638 0.0000000
##
## [[2]]
##      [,1]
## [1,]  691
## [2,] -691
##
## [[3]]
##           [,1]      [,2]
## [1,]  166721.5 -166721.5
## [2,] -166721.5  166721.5
##
## [[4]]
## [1] 2.86394399 0.09058515
```

Since the p-value is 0.09058515, which is greater than $\alpha = 0.05$, we do not reject the null hypothesis, showing that the hazard rates of group 1 and group 0 are the same at 95% significance level.

part c)

Although the conclusion from logrank and Gehan's test are the same: do not reject the null hypothesis, showing that the hazard rates of group 1 and group 0 are the same at 95% significance level, one should realize that the weights W_j assigned are not the same. Logrank test is unweighted ($W_j = 1$) and Gehan's Wilcoxon Test has $W_j = Y_j$, which means that as t_j increases, Y_j decreases, so more weights given to early events/departure. However, in this question, this should not be the case. So logrank test is more preferred.

Question 4

```
library(KMsurv)
data("bmt")
head(bmt,3)
```

```
##   group   t1   t2 d1 d2 d3   ta da  tc dc tp dp z1 z2 z3 z4 z5 z6   z7 z8 z9
## 1     1 2081 2081  0  0  0   67  1 121  1 13  1 26 33  1  0  1  1   98  0  1
## 2     1 1602 1602  0  0  0 1602  0 139  1 18  1 21 37  1  1  0  0 1720  0  1
## 3     1 1496 1496  0  0  0 1496  0 307  1 12  1 26 35  1  1  1  0  127  0  1
##   z10
## 1   0
## 2   0
## 3   0
```

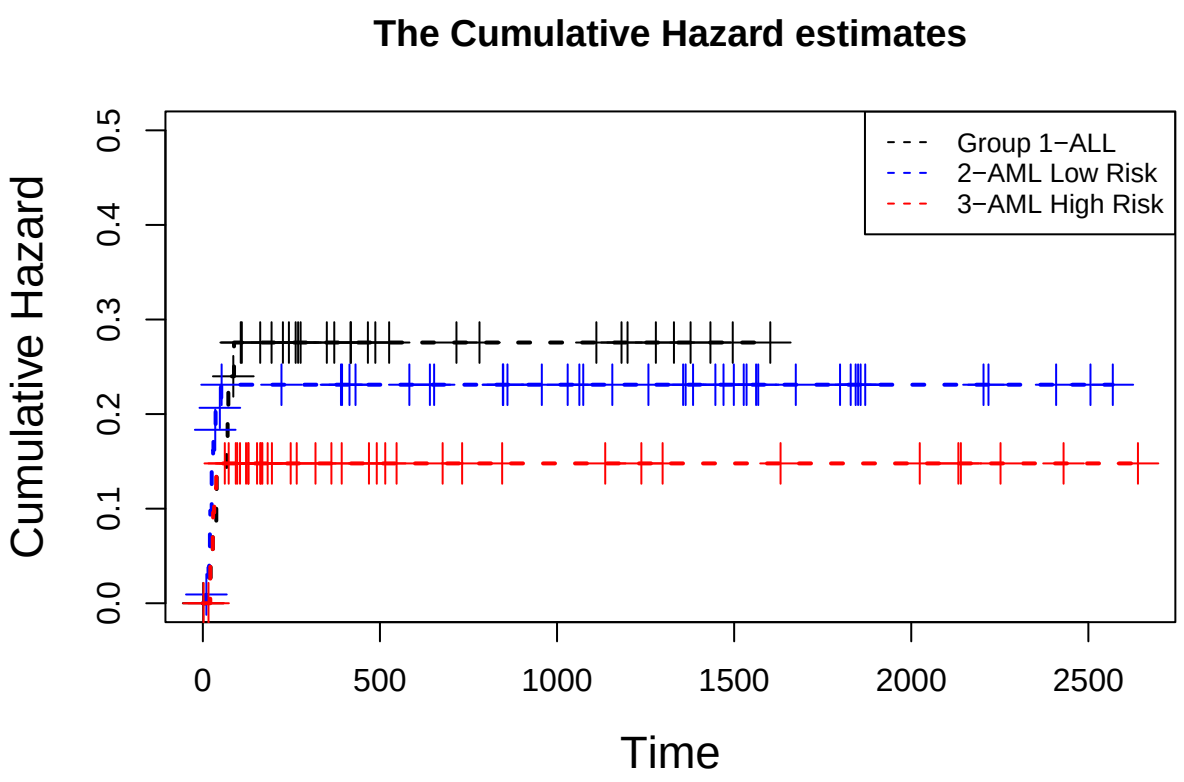
```
# group Disease Group 1-ALL, 2-AML Low Risk, 3-AML High Risk
# t2 Disease Free Survival Time (Time To Relapse, Death Or End Of Study)
# d2 Relapse Indicator 1-Relapsed, 0-Disease Free
# ta Time To Acute Graft-Versus-Host Disease
# da Acute GVHD Indicator 1-Developed Acute GVHD 0-Never Developed Acute GVHD)
```

part a)

```
survtime = bmt$ta
indicator = bmt$da
group = bmt$group
# log-rank test
logrank.test = survdiff(Surv(survtime, indicator)~group, rho = 0, data = bmt)
logrank.test
```

```
## Call:
## survdiff(formula = Surv(survtime, indicator) ~ group, data = bmt,
##     rho = 0)
##
##          N Observed Expected (O-E)^2/E (O-E)^2/V
## group=1 38        9     7.42     0.336     0.472
## group=2 54       11     9.90     0.121     0.197
## group=3 45        6     8.67     0.825     1.244
##
##  Chisq= 1.3  on 2 degrees of freedom, p= 0.5
```

```
fitNA = survfit(Surv(survtime, indicator)~group, ctype = 1)
plot(fitNA, lwd=2, lty=2, conf.int=FALSE, mark.time=TRUE, cex=2, cex.lab=1.4,
cumhaz = TRUE, col=c("black", "blue", "red"), xlab="Time", ylab="Cumulative Hazard",
main="The Cumulative Hazard estimates", ylim=c(0,0.5))
legend("topright",c("Group 1-ALL", "2-AML Low Risk", "3-AML High Risk"),lty=c(2),
col=c("black", "blue", "red"), cex=0.8)
```

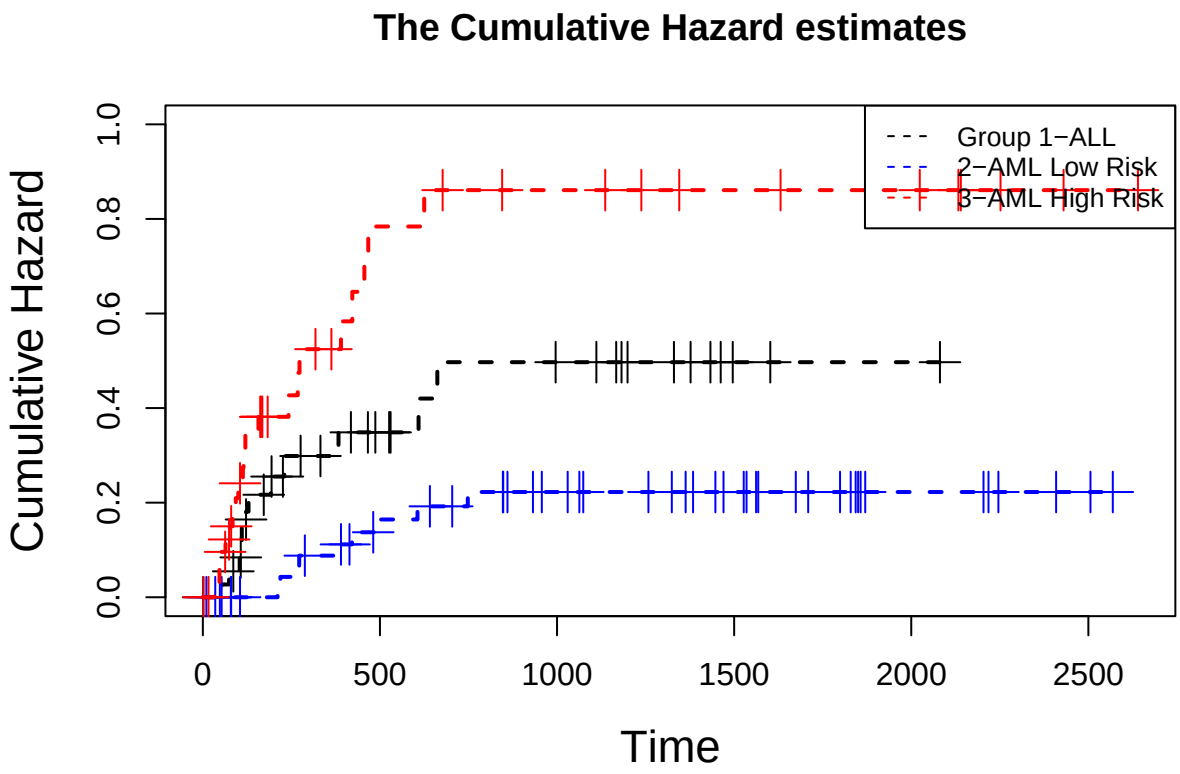

The Cumulative Hazard estimates
Cumulative Hazard
0.0
0.1
0.2
0.3
0.4
0.5
0
500
1000
1500
2000
2500
Time
Group 1−ALL
2−AML Low Risk
3−AML High Risk

part b)

```
survtime = bmt$t2
indicator = bmt$d2
group = bmt$group
# log-rank test
logrank.test = survdiff(Surv(survtime, indicator)~group, rho = 0, data = bmt)
logrank.test
```

```
## Call:
## survdiff(formula = Surv(survtime, indicator) ~ group, data = bmt,
##     rho = 0)
##
##          N Observed Expected (O-E)^2/E (O-E)^2/V
## group=1 38       12     11.2    0.0625    0.0854
## group=2 54        9     20.2    6.1851   12.0778
## group=3 45       21     10.7   10.0122   13.5301
##
##  Chisq= 16.5  on 2 degrees of freedom, p= 3e-04
```

```
fitNA = survfit(Surv(survtime, indicator)~group, ctype = 1)
plot(fitNA, lwd=2, lty=2, conf.int=FALSE, mark.time=TRUE, cex=2, cex.lab=1.4,
cumhaz = TRUE, col=c("black", "blue", "red"), xlab="Time", ylab="Cumulative Hazard",
main="The Cumulative Hazard estimates", ylim=c(0,1))
legend("topright",c("Group 1-ALL", "2-AML Low Risk", "3-AML High Risk"),lty=c(2),
col=c("black", "blue", "red"), cex=0.8)
```


The Cumulative Hazard estimates
Cumulative Hazard
0.0
0.2
0.4
0.6
0.8
1.0
0
500
1000
1500
2000
2500
Time
Group 1−ALL
2−AML Low Risk
3−AML High Risk

Question 5

See another pdf named **question 5**, which is a hand-written solution

Question 6

part a)

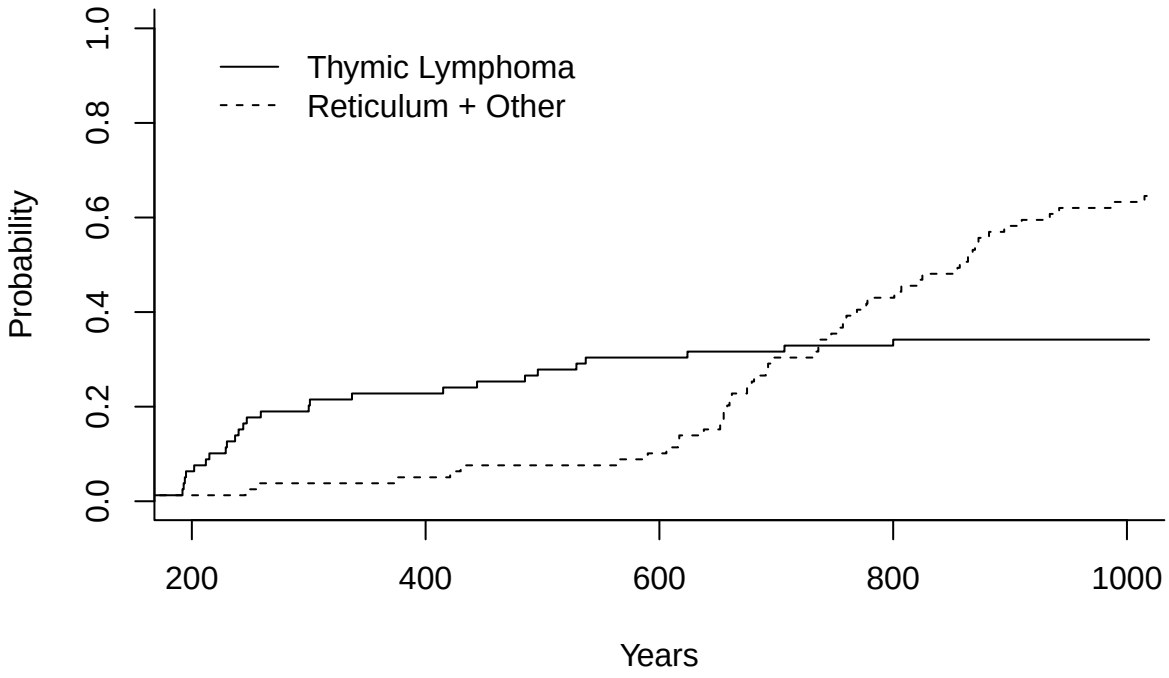
```
thymic = c(158,192,193,194,195,202,212,215,229,230,237,240,244,
           247,259,300,301,337,415,444,485,496,529,537,624,707,800)
reticulum = c(430,590,606,638,655,679,691,693,696,747,752,760,778,821,986)
other = c(136,246,255,376,421,565,616,617,652,655,658,660,662,675,681,734,
          736,737,757,769,777,801,807,825,855,857,864,868,870,873,882,895,
          910,934,942,1015,1019)
age = append(append(thymic, reticulum), other)
d1 = append(rep(1, length(thymic)), rep(0, length(reticulum)+length(other)))
d2 = append(append(rep(0, length(thymic)), rep(1, length(reticulum))), rep(0, length(other)))
d3 = append(append(rep(0, length(thymic)), rep(0, length(reticulum))), rep(1, length(other)))
workdata = data.frame(age, d1, d2, d3)
head(workdata)
```

```
##   age d1 d2 d3
## 1 158  1  0  0
## 2 192  1  0  0
## 3 193  1  0  0
## 4 194  1  0  0
## 5 195  1  0  0
## 6 202  1  0  0
```

```
library(cmprsk)
```

```
## Warning: package 'cmprsk' was built under R version 4.2.2
```

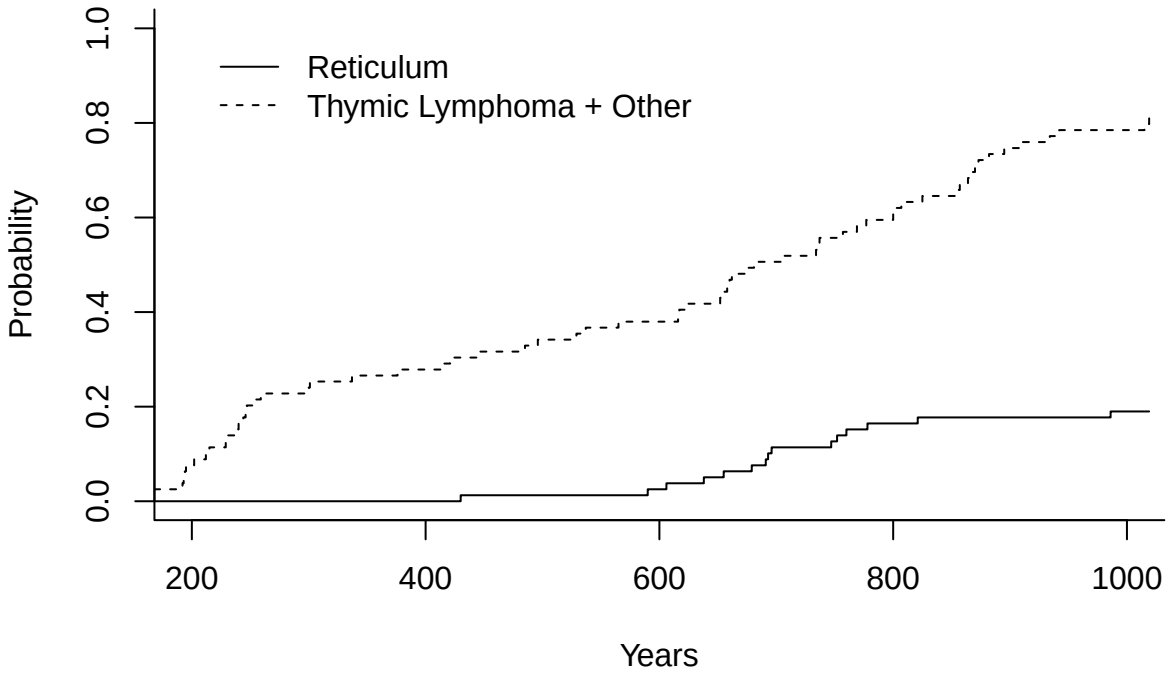
```
library(survival)
## Thymic Lymphoma
indicator = append(rep(1, length(thymic)), rep(2, length(reticulum)+length(other)))
data1 = data.frame(age, indicator)
CI_thymic = cuminc(age, indicator, cencode=0)
plot.cuminc(CI_thymic, xlim = c(200,1000), curvlab = c("Thymic Lymphoma", "Reticulum + Other"))
```




```
timepoints(CI_thymic, c(200,300,400,500,600,700,800,900,1000))
```

```
## $est
##            200        300        400        500       600       700       800
## 1 1 0.06329114 0.20253165 0.22784810 0.27848101 0.3037975 0.3164557 0.3417722
## 1 2 0.01265823 0.03797468 0.05063291 0.07594937 0.1012658 0.3037975 0.4303797
##           900      1000
## 1 1 0.3417722 0.3417722
## 1 2 0.5822785 0.6329114
##
## $var
##              200          300          400          500         600         700
## 1 1 0.0007604640 0.0020744603 0.0022604272 0.0025838985 0.002721197 0.002784103
## 1 2 0.0001602307 0.0004693196 0.0006179424 0.0009030443 0.001172436 0.002740906
##             800         900        1000
## 1 1 0.002904032 0.002904032 0.002904032
## 1 2 0.003192913 0.003204634 0.003085273
```

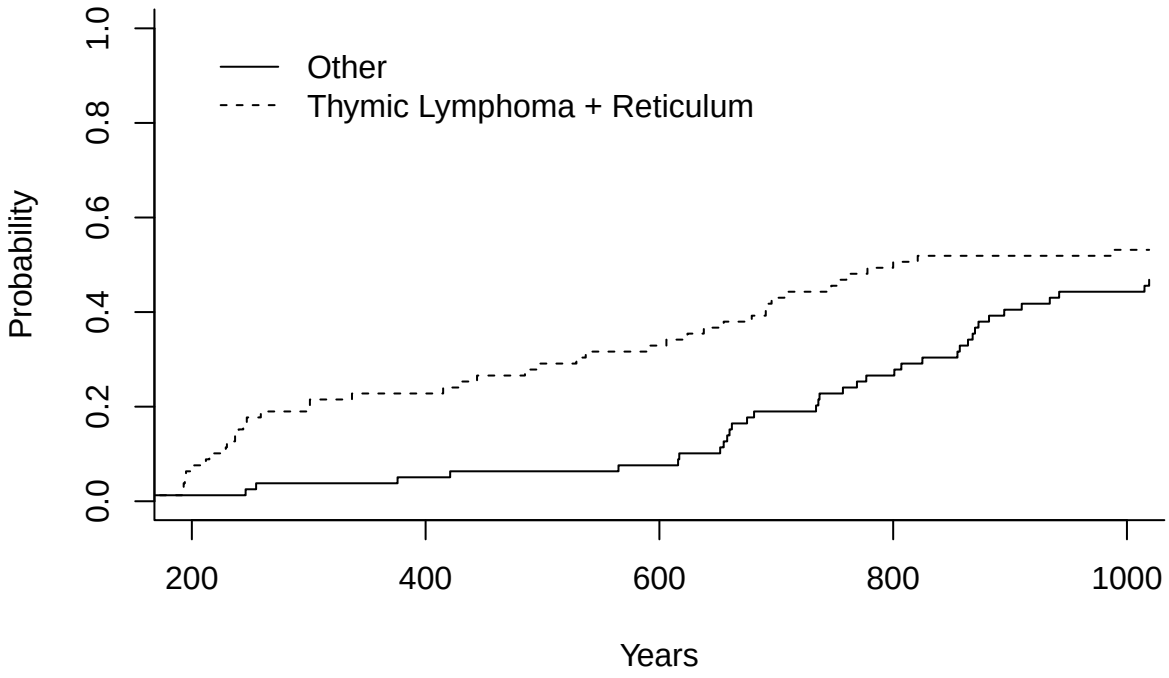
```
## Reticulum
indicator = append(append(rep(2,length(thymic)), rep(1, length(reticulum))), rep(2, length(other)))
data1 = data.frame(age, indicator)
CI_reticulum = cuminc(age, indicator, cencode=0)
plot.cuminc(CI_reticulum, xlim = c(200,1000), curvlab = c("Reticulum", "Thymic Lymphoma + Other"))
```




```
timepoints(CI_reticulum, c(200,300,400,500,600,700,800,900,1000))
```

```
## $est
##             200       300      400        500        600       700       800
## 1 1 0.00000000 0.0000000 0.000000 0.01265823 0.02531646 0.1139241 0.1645570
## 1 2 0.07594937 0.2405063 0.278481 0.34177215 0.37974684 0.5063291 0.6075949
##            900      1000
## 1 1 0.1772152 0.1898734
## 1 2 0.7468354 0.7848101
##
## $var
##               200         300        400          500          600         700
## 1 1 0.0000000000 0.000000000 0.00000000 0.0001611295 0.0003185554 0.001310278
## 1 2 0.0009001523 0.002346343 0.00258211 0.0028935812 0.0030315474 0.003228018
##              800         900        1000
## 1 1 0.001793436 0.001906460 0.002067662
## 1 2 0.003100453 0.002517452 0.002276842
```

```
## Other
indicator = append(append(rep(2,length(thymic)), rep(2, length(reticulum))), rep(1, length(other)))
data1 = data.frame(age, indicator)
CI_other = cuminc(age, indicator, cencode=0)
plot.cuminc(CI_other, xlim = c(200,1000), curvlab = c("Other", "Thymic Lymphoma + Reticulum"))
```




```
timepoints(CI_other, c(200,300,400,500,600,700,800,900,1000))
```

```
## $est
##            200        300        400        500        600       700       800
## 1 1 0.01265823 0.03797468 0.05063291 0.06329114 0.07594937 0.1898734 0.2658228
## 1 2 0.06329114 0.20253165 0.22784810 0.29113924 0.32911392 0.4303797 0.5063291
##           900      1000
## 1 1 0.4050633 0.4430380
## 1 2 0.5189873 0.5316456
##
## $var
##              200          300          400          500         600         700
## 1 1 0.0001602307 0.0004693196 0.0006179424 0.0007625343 0.000903408 0.001989947
## 1 2 0.0007604640 0.0020744603 0.0022604272 0.0026543687 0.002841853 0.003166256
##              800         900        1000
## 1 1 0.002538223 0.003186699 0.003281945
## 1 2 0.003246350 0.003247421 0.003296670
```

part b)

```
time = unique(sort(workdata$age))
thymic = timepoints(CI_thymic, time)$est[1,]
reticulum = timepoints(CI_reticulum, time)$est[1,]
other = timepoints(CI_other, time)$est[1,]
status = rep(1, nrow(workdata))
fit = survfit(Surv(age, status)~1, data = workdata)
KM = summary(fit)$surv
match = round((1-KM),5)==round((thymic+reticulum+other),5)
result = data.frame(thymic, reticulum, other, KM, 1-KM, thymic+reticulum+other, match)
knitr::kable(result, "pipe", col.names = c("CI_thymic", "CI_reticulum", "CI_other",
                                           "KM", "1-KM", "CI(t)", "Match"))
```

	CI_thymic	CI_reticulum	CI_other	KM	1-KM	CI(t)	Match
136	0.0000000	0.0000000	0.0126582	0.9873418	0.0126582	0.0126582	TRUE
158	0.0126582	0.0000000	0.0126582	0.9746835	0.0253165	0.0253165	TRUE
192	0.0253165	0.0000000	0.0126582	0.9620253	0.0379747	0.0379747	TRUE
193	0.0379747	0.0000000	0.0126582	0.9493671	0.0506329	0.0506329	TRUE
194	0.0506329	0.0000000	0.0126582	0.9367089	0.0632911	0.0632911	TRUE
195	0.0632911	0.0000000	0.0126582	0.9240506	0.0759494	0.0759494	TRUE
202	0.0759494	0.0000000	0.0126582	0.9113924	0.0886076	0.0886076	TRUE
212	0.0886076	0.0000000	0.0126582	0.8987342	0.1012658	0.1012658	TRUE
215	0.1012658	0.0000000	0.0126582	0.8860759	0.1139241	0.1139241	TRUE
229	0.1139241	0.0000000	0.0126582	0.8734177	0.1265823	0.1265823	TRUE
230	0.1265823	0.0000000	0.0126582	0.8607595	0.1392405	0.1392405	TRUE
237	0.1392405	0.0000000	0.0126582	0.8481013	0.1518987	0.1518987	TRUE
240	0.1518987	0.0000000	0.0126582	0.8354430	0.1645570	0.1645570	TRUE
244	0.1645570	0.0000000	0.0126582	0.8227848	0.1772152	0.1772152	TRUE
246	0.1645570	0.0000000	0.0253165	0.8101266	0.1898734	0.1898734	TRUE
247	0.1772152	0.0000000	0.0253165	0.7974684	0.2025316	0.2025316	TRUE
255	0.1772152	0.0000000	0.0379747	0.7848101	0.2151899	0.2151899	TRUE
259	0.1898734	0.0000000	0.0379747	0.7721519	0.2278481	0.2278481	TRUE
300	0.2025316	0.0000000	0.0379747	0.7594937	0.2405063	0.2405063	TRUE
301	0.2151899	0.0000000	0.0379747	0.7468354	0.2531646	0.2531646	TRUE
337	0.2278481	0.0000000	0.0379747	0.7341772	0.2658228	0.2658228	TRUE
376	0.2278481	0.0000000	0.0506329	0.7215190	0.2784810	0.2784810	TRUE
415	0.2405063	0.0000000	0.0506329	0.7088608	0.2911392	0.2911392	TRUE
421	0.2405063	0.0000000	0.0632911	0.6962025	0.3037975	0.3037975	TRUE
430	0.2405063	0.0126582	0.0632911	0.6835443	0.3164557	0.3164557	TRUE
444	0.2531646	0.0126582	0.0632911	0.6708861	0.3291139	0.3291139	TRUE
485	0.2658228	0.0126582	0.0632911	0.6582278	0.3417722	0.3417722	TRUE
496	0.2784810	0.0126582	0.0632911	0.6455696	0.3544304	0.3544304	TRUE
529	0.2911392	0.0126582	0.0632911	0.6329114	0.3670886	0.3670886	TRUE
537	0.3037975	0.0126582	0.0632911	0.6202532	0.3797468	0.3797468	TRUE
565	0.3037975	0.0126582	0.0759494	0.6075949	0.3924051	0.3924051	TRUE
590	0.3037975	0.0253165	0.0759494	0.5949367	0.4050633	0.4050633	TRUE
606	0.3037975	0.0379747	0.0759494	0.5822785	0.4177215	0.4177215	TRUE
616	0.3037975	0.0379747	0.0886076	0.5696203	0.4303797	0.4303797	TRUE
617	0.3037975	0.0379747	0.1012658	0.5569620	0.4430380	0.4430380	TRUE
624	0.3164557	0.0379747	0.1012658	0.5443038	0.4556962	0.4556962	TRUE

	CI_thymic	CI_reticulum	CI_other	KM	1-KM	CI(t)	Match
638	0.3164557	0.0506329	0.1012658	0.5316456	0.4683544	0.4683544	TRUE
652	0.3164557	0.0506329	0.1139241	0.5189873	0.4810127	0.4810127	TRUE
655	0.3164557	0.0632911	0.1265823	0.4936709	0.5063291	0.5063291	TRUE
658	0.3164557	0.0632911	0.1392405	0.4810127	0.5189873	0.5189873	TRUE
660	0.3164557	0.0632911	0.1518987	0.4683544	0.5316456	0.5316456	TRUE
662	0.3164557	0.0632911	0.1645570	0.4556962	0.5443038	0.5443038	TRUE
675	0.3164557	0.0632911	0.1772152	0.4430380	0.5569620	0.5569620	TRUE
679	0.3164557	0.0759494	0.1772152	0.4303797	0.5696203	0.5696203	TRUE
681	0.3164557	0.0759494	0.1898734	0.4177215	0.5822785	0.5822785	TRUE
691	0.3164557	0.0886076	0.1898734	0.4050633	0.5949367	0.5949367	TRUE
693	0.3164557	0.1012658	0.1898734	0.3924051	0.6075949	0.6075949	TRUE
696	0.3164557	0.1139241	0.1898734	0.3797468	0.6202532	0.6202532	TRUE
707	0.3291139	0.1139241	0.1898734	0.3670886	0.6329114	0.6329114	TRUE
734	0.3291139	0.1139241	0.2025316	0.3544304	0.6455696	0.6455696	TRUE
736	0.3291139	0.1139241	0.2151899	0.3417722	0.6582278	0.6582278	TRUE
737	0.3291139	0.1139241	0.2278481	0.3291139	0.6708861	0.6708861	TRUE
747	0.3291139	0.1265823	0.2278481	0.3164557	0.6835443	0.6835443	TRUE
752	0.3291139	0.1392405	0.2278481	0.3037975	0.6962025	0.6962025	TRUE
757	0.3291139	0.1392405	0.2405063	0.2911392	0.7088608	0.7088608	TRUE
760	0.3291139	0.1518987	0.2405063	0.2784810	0.7215190	0.7215190	TRUE
769	0.3291139	0.1518987	0.2531646	0.2658228	0.7341772	0.7341772	TRUE
777	0.3291139	0.1518987	0.2658228	0.2531646	0.7468354	0.7468354	TRUE
778	0.3291139	0.1645570	0.2658228	0.2405063	0.7594937	0.7594937	TRUE
800	0.3417722	0.1645570	0.2658228	0.2278481	0.7721519	0.7721519	TRUE
801	0.3417722	0.1645570	0.2784810	0.2151899	0.7848101	0.7848101	TRUE
807	0.3417722	0.1645570	0.2911392	0.2025316	0.7974684	0.7974684	TRUE
821	0.3417722	0.1772152	0.2911392	0.1898734	0.8101266	0.8101266	TRUE
825	0.3417722	0.1772152	0.3037975	0.1772152	0.8227848	0.8227848	TRUE
855	0.3417722	0.1772152	0.3164557	0.1645570	0.8354430	0.8354430	TRUE
857	0.3417722	0.1772152	0.3291139	0.1518987	0.8481013	0.8481013	TRUE
864	0.3417722	0.1772152	0.3417722	0.1392405	0.8607595	0.8607595	TRUE
868	0.3417722	0.1772152	0.3544304	0.1265823	0.8734177	0.8734177	TRUE
870	0.3417722	0.1772152	0.3670886	0.1139241	0.8860759	0.8860759	TRUE
873	0.3417722	0.1772152	0.3797468	0.1012658	0.8987342	0.8987342	TRUE
882	0.3417722	0.1772152	0.3924051	0.0886076	0.9113924	0.9113924	TRUE
895	0.3417722	0.1772152	0.4050633	0.0759494	0.9240506	0.9240506	TRUE
910	0.3417722	0.1772152	0.4177215	0.0632911	0.9367089	0.9367089	TRUE
934	0.3417722	0.1772152	0.4303797	0.0506329	0.9493671	0.9493671	TRUE
942	0.3417722	0.1772152	0.4430380	0.0379747	0.9620253	0.9620253	TRUE
986	0.3417722	0.1898734	0.4430380	0.0253165	0.9746835	0.9746835	TRUE
1015	0.3417722	0.1898734	0.4556962	0.0126582	0.9873418	0.9873418	TRUE
1019	0.3417722	0.1898734	0.4683544	0.0000000	1.0000000	1.0000000	TRUE

part c)

The complement of the marginal Kaplan-Meier estimator refers to the estimator of the survival probability for the complementary (non-censored) observations at each time point. Mathematically, the complement of the marginal Kaplan-Meier estimator is defined as:

$$\hat{S}_c(t) = \frac{d_c(t)}{n_c(t)}$$

where $d_c(t)$ is the number of complementary observations who survived beyond time t and $n_c(t)$ is the number of complementary observations at risk at time t.

The complement of the marginal Kaplan-Meier estimator estimates the survival probability for the non-censored observations, i.e., those individuals who have experienced the event of interest by the end of the study or have been lost to follow-up. This estimator is particularly useful when we want to estimate the survival probability for the entire population, including those individuals who are not censored.

In summary, the complement of the marginal Kaplan-Meier estimator estimates the survival probability for the non-censored observations and provides an estimate of the overall survival probability for the population.

```
fit1 = survfit(Surv(age, d1)~1, data = workdata)
fit2 = survfit(Surv(age, d2)~1, data = workdata)
fit3 = survfit(Surv(age, d3)~1, data = workdata)

km1 = summary(fit1, times = c(200,300,400,500,600,700,800,900,1000))$surv
km2 = summary(fit2, times = c(200,300,400,500,600,700,800,900,1000))$surv
km3 = summary(fit3, times = c(200,300,400,500,600,700,800,900,1000))$surv
```

```
comp1 = 1 - km1
CI1 = timepoints(CI_thymic, c(200,300,400,500,600,700,800,900,1000))$est[1,]
comp2 = 1 - km2
CI2 = timepoints(CI_reticulum, c(200,300,400,500,600,700,800,900,1000))$est[1,]
comp3 = 1 - km3
CI3 = timepoints(CI_other, c(200,300,400,500,600,700,800,900,1000))$est[1,]

partc.result = data.frame(comp1, CI1, round(comp1-CI1,5),
                          comp2, CI2, round(comp2-CI2,5),
                          comp3, CI3, round(comp3-CI3,5))
knitr::kable(partc.result, "latex", col.names = c("F_x1", "CI_1", "difference",
                                                  "F_x2", "CI_2", "difference",
                                                  "F_x3", "CI_3", "difference"))
```

	F_x1	CI_1	difference	F_x2	CI_2	difference	F_x3	CI_3	difference
200	0.0641026	0.0632911	0.00081	0.0000000	0.0000000	0.00000	0.0126582	0.0126582	0.00000
300	0.2061492	0.2025316	0.00362	0.0000000	0.0000000	0.00000	0.0432791	0.0379747	0.00530
400	0.2326109	0.2278481	0.00476	0.0000000	0.0000000	0.00000	0.0597743	0.0506329	0.00914
500	0.2879586	0.2784810	0.00948	0.0181818	0.0126582	0.00552	0.0765640	0.0632911	0.01327
600	0.3158818	0.3037975	0.01208	0.0386364	0.0253165	0.01332	0.0954096	0.0759494	0.01946
700	0.3314300	0.3164557	0.01497	0.2081866	0.1139241	0.09426	0.2822124	0.1898734	0.09234
800	0.3877306	0.3417722	0.04596	0.3358301	0.1645570	0.17127	0.4393456	0.2658228	0.17352
900	0.3877306	0.3417722	0.04596	0.3773408	0.1772152	0.20013	0.8006562	0.4050633	0.39559
1000	0.3877306	0.3417722	0.04596	0.5848938	0.1898734	0.39502	0.9003281	0.4430380	0.45729

part d)

For a particular risk J, let $\widehat{CI_J(t)}$ and $\widehat{CI_{J^C}(t)}$ be the cumulative incidence function estimators for risk J and for all other risks lumped together, respectively.

We could have the conditional probability function estimator:

$$\widehat{CP_J(t)} = \frac{\widehat{CI_J(t)}}{1 - \widehat{CI_{J^C}(t)}}$$

It estimates the conditional probability of event J's occurring by time t given that none of other causes have occurred by t.

```
# for t=500, k=1
numerator = timepoints(CI_thymic, c(200,300,400,500,600,700,800,900,1000))$est[1,4]
denominator = 1- timepoints(CI_thymic, c(200,300,400,500,600,700,800,900,1000))$est[2,4]
numerator/denominator
```

```
## [1] 0.3013699
```

```
# for t=800, k=1
numerator = timepoints(CI_thymic, c(200,300,400,500,600,700,800,900,1000))$est[1,7]
denominator = 1- timepoints(CI_thymic, c(200,300,400,500,600,700,800,900,1000))$est[2,7]
numerator/denominator
```

```
## [1] 0.6
```