Recurrent Events

Klaus Holst & Thomas Scheike

November 20, 2018

Overview

For recurrent events data it is often of interest to compute basis descriptive quantities as a first go at getting some basic understanding of the phenonmenon studied. We here demonstrate how one can compute

- the marginal mean
- the variance
- the probability of exceeding k events

In addition several tools can be used for simulating recurrent events and bivariate recurrent events data, in the case with a possible terminating event.

We start by simulating some recurrent events data with two type of events with cumulative hazards

- $\Lambda_1(t)$
- $\Lambda_2(t)$
- $\Lambda_D(t)$

where we consider types 1 and 4 and with a rate of the terminal event given by $\Lambda_D(t)$. We let the events be independent, but could also specify a random effects structure to generate dependence.

When simulating data we can impose various random-effects structures to generate dependence

- We can draw normally distributed random effects Z_1, Z_2, Z_d were the variance (var.z) and correlation can be specified (cor.mat) (dependence=2). Then the intensities are
  - $\exp(Z_1)\lambda_1(t)$
  - $\exp(Z_2)\lambda_2(t)$
  - $\exp(Z_3)\lambda_D(t)$
- We can one gamma distributed random effects Z. Then the intensities are (dependence=1)
  - $Z\lambda_1(t)$
  - $Z\lambda_2(t)$
  - $Z\lambda_D(t)$

- We can draw gamma distributed random effects Z_1, Z_2, Z_d were the sum-structure can be speicifed via a matrix cor.mat. Then we compute $\tilde{Z}_j = \sum_k Z_k^{cor.mat(j,k)}$ for $j = 1, 2, 3$ (dependence=3) Then the intensities are
  - $\tilde{Z}_1 \lambda_1(t)$
  - $\tilde{Z}_2 \lambda_2(t)$
  - $\tilde{Z}_3 \lambda_D(t)$
- The intensities can be independent (dependence=0)

We return to how to run the different set-ups later and start by simulating independent processes.

Utility functions

We here mention two utility functions

- tie.breaker for breaking ties among jump-times which is expected in the functions below.
- count.history that counts the number of jumps previous for each subject that is $N_1(t-)$ and $N_2(t-)$.

Marginal Mean

We start by estimating the marginal mean $E(N_1(t \wedge D))$ where D is the timing of the terminal event.

This is based on a rate model for

- the type 1 events
- the terminal event

and is defined as $\mu_1(t) = E(N_1^*(t))$

$$\int_0^t S(u) dR_1(u) \tag{1}$$

where $S(t) = P(D \geq t)$ and $dR_1(t) = E(dN_1^*(t) | D \geq t)$

and can therefore be estimated by a

- Kaplan-Meier estimator, $\hat{S}(u)$
- Nelson-Aalen estimator for $R_1(t)$

$$\hat{R}_1(t) = \sum_i \int_0^t \frac{1}{Y_\bullet(s)} dN_{1i}(s) \tag{2}$$

where $Y_\bullet(t) = \sum_i Y_i(t)$ such that the estimator is

$$\hat{\mu}_1(t) = \int_0^t \hat{S}(u) d\hat{R}_1(u). \tag{3}$$

Cook & Lawless (1997), and developed further in Gosh & Lin (2000).

The variance can be estimated based on the asymptotic expansion of $\hat{\mu}_1(t) - \mu_1(t)$

$$\sum_i \int_0^t \frac{S(s)}{\pi(s)} dM_{i1} - \mu_1(t) \int_0^t \frac{1}{\pi(s)} dM_i^d + \int_0^t \frac{\mu_1(s)}{\pi(s)} dM_i^d,$$

with mean-zero processes

- $M_i^d(t) = N_i^D(t) - \int_0^t Y_i(s) d\Lambda^D(s)$,
- $M_{i1}(t) = N_{i1}(t) - \int_0^t Y_i(s) dR_1(s)$.

as in Gosh & Lin (2000)

```
library(mets)
set.seed(1000) # to control output in simulatins for
    p-values below.

data(base1cumhaz)
data(base4cumhaz)
data(drcumhaz)
ddr <- drcumhaz
base1 <- base1cumhaz
base4 <- base4cumhaz
rr <- simRecurrent(1000,base1,death.cumhaz=ddr)
rr$x <- rnorm(nrow(rr))
rr$strata <- floor((rr$id-0.01)/500)
dlist(rr,.~id| id %in% c(1,7,9))
```

```
id: 1
  entry time  status rr dtime fdeath death start stop  x     strata
1 0     133.1 0      1  133.1 1      1     0     133.1 1.185 0
------------------------------------------------------------
id: 7
     entry  time   status rr dtime fdeath death start  stop   x       strata
7       0.0  813.3 1      1  1729  1      0        0.0  813.3  1.5495 0
1004  813.3 1288.4 1      1  1729  1      0      813.3 1288.4  1.0535 0
1658 1288.4 1315.4 1      1  1729  1      0     1288.4 1315.4  1.5330 0
2150 1315.4 1449.4 1      1  1729  1      0     1315.4 1449.4  0.8944 0
2539 1449.4 1726.1 1      1  1729  1      0     1449.4 1726.1 -0.1931 0
2851 1726.1 1729.4 0      1  1729  1      1     1726.1 1729.4  0.4081 0
------------------------------------------------------------
id: 9
     entry  time   status rr dtime fdeath death start  stop   x       strata
9       0.0  433.5 1      1  5110  0      0        0.0  433.5 -0.4660 0
1006  433.5 2451.1 1      1  5110  0      0      433.5 2451.1  1.0647 0
1659 2451.1 3629.7 1      1  5110  0      0     2451.1 3629.7 -0.2506 0
2151 3629.7 3644.7 1      1  5110  0      0     3629.7 3644.7 -0.6748 0
2540 3644.7 3695.8 1      1  5110  0      0     3644.7 3695.8  0.6510 0
2852 3695.8 3890.7 1      1  5110  0      0     3695.8 3890.7 -0.2033 0
3112 3890.7 5110.0 0      1  5110  0      0     3890.7 5110.0 -1.6981 0
```

The status variable keeps track of the recurrent evnts and their type, and death the timing of death.

```
# to fit non-parametric models with just a baseline
xr <- phreg(Surv(entry,time,status)~cluster(id),data=rr)
dr <- phreg(Surv(entry,time,death)~cluster(id),data=rr)
par(mfrow=c(1,3))
```

```
bplot(dr,se=TRUE)
title(main="death")
bplot(xr,se=TRUE)
# robust standard errors
rxr <- robust.phreg(xr,fixbeta=1)
bplot(rxr,se=TRUE,robust=TRUE,add=TRUE,col=4)

# marginal mean of expected number of recurrent events
out <- recurrentMarginal(xr,dr)
bplot(out,se=TRUE,ylab="marginal mean",col=2)
```

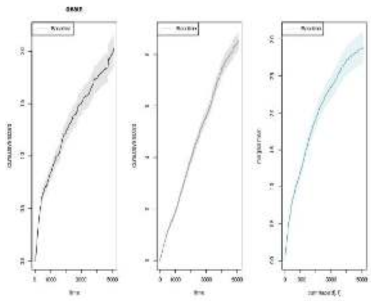


Figure 1: Marginal mean for number of type 1 events, rate for death (panel (a)), rate for type 1 among survivors (panel (b)), and marginal mean (panel (c)).

We can do the same with strata

```
xr <- phreg(Surv(entry,time,status)~strata(strata)+cluster(
     id),data=rr)
dr <- phreg(Surv(entry,time,death)~strata(strata)+cluster(id
     ),data=rr)
par(mfrow=c(1,3))
bplot(dr,se=TRUE)
title(main="death")
bplot(xr,se=TRUE)
rxr <- robust.phreg(xr,fixbeta=1)
bplot(rxr,se=TRUE,robust=TRUE,add=TRUE,col=1:2)

out <- recurrentMarginal(xr,dr)
bplot(out,se=TRUE,ylab="marginal mean",col=1:2)
```

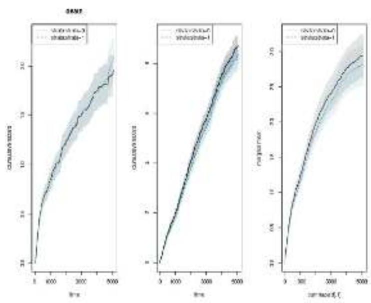


Figure 2: Recurrent events

Furhter, if we adjust for covariates for the two rates we can still do predictions of marginal mean, what can be plotted is the baseline marginal mean, that is for the covariates equal to 0 for both models.

```
# cox case
xr <- phreg(Surv(entry,time,status)~x+cluster(id),data=rr)
dr <- phreg(Surv(entry,time,death)~x+cluster(id),data=rr)
par(mfrow=c(1,3))
bplot(dr,se=TRUE)
title(main="death")
bplot(xr,se=TRUE)
rxr <- robust.phreg(xr)
bplot(rxr,se=TRUE,robust=TRUE,add=TRUE,col=1:2)

out <- recurrentMarginal(xr,dr)
bplot(out,se=TRUE,ylab="marginal mean",col=1:2)

# predictions witout se's
outX <- recmarg(xr,dr,Xr=1,Xd=1)
bplot(outX,add=TRUE,col=3)
```

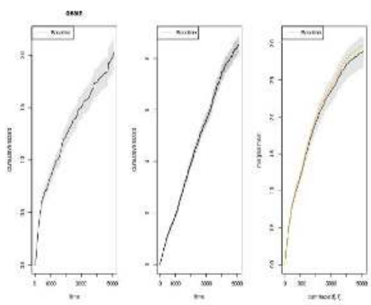


Figure 3: Recurrent events with cox models for rates.

Other marginal properties

- $P(N_1^*(t) \geq k)$
  - cumulative incidence of $T_k = \inf\{t : N_1^*(t) = k\}$ with competing D.

We note also that $N_1^*(t)^2$ can be written as

$$\sum_{k=0}^{K} \int_0^t I(D > s) I(N_1^*(s-) = k) f(k) dN_1^*(s)$$

with $f(k) = (k+1)^2 - k^2$, such that its mean can be written as

$$\sum_{k=0}^{K} \int_0^t S(s) f(k) P(N_1^*(s-) = k | D \geq s) E(dN_1^*(s) | N_1^*(s-) = k, D > s)$$

and estimated by

$$\hat{\mu}_{1,2}(t) = \sum_{k=0}^{K} \int_0^t \hat{S}(s) f(k) \frac{Y_{1\bullet}^k(s)}{Y_\bullet(s)} \frac{1}{Y_{1\bullet}^k(s)} dN_{1\bullet}^k(s) = \sum_{i=1}^{n} \int_0^t \hat{S}(s) f(N_{i1}(s-)) \frac{1}{Y_\bullet(s)} dN_{i1}(s),$$

Compared to "product-limit" estimator for $E((N_1^*(t))^2)$

$$\hat{\mu}_{1,2}(t) = \sum_{k=0}^{K} k^2 (\hat{F}_k(t) - \hat{F}_{k+1}(t)). \qquad (4)$$

Probabilty of exceeding "k"

Note also that $I(N_1^*(t) \geq k)$ is

$$\int_0^t I(D > s) I(N_1^*(s-) = k-1) dN_1^*(s),$$

suggesting that its mean can be computed as

$$\int_0^t S(s) P(N_1^*(s-) = k-1 | D \geq s) E(dN_1^*(s) | N_1^*(s-) = k-1, D > s)$$

and estimated by

$$\tilde{F}_k(t) = \int_0^t \hat{S}(s) \frac{Y_{1\bullet}^{k-1}(s)}{Y_\bullet(s)} \frac{1}{Y_{1\bullet}^{k-1}(s)} dN_{1\bullet}^{k-1}(s).$$

```
cor.mat <- corM <- rbind(c(1.0, 0.6, 0.9), c(0.6, 1.0, 0.5),
      c(0.9, 0.5, 1.0))
rr <- simRecurrent(1000,base1,cumhaz2=base4,death.cumhaz=ddr
     )
rr <- count.history(rr)
dtable(rr,~death+status)

oo <- prob.exceedRecurrent(rr,1)
bplot(oo)
```


```
cor.mat <- corM <- rbind(c(1.0, 0.6, 0.9), c(0.6, 1.0, 0.5),
      c(0.9, 0.5, 1.0))
rr <- simRecurrent(1000,base1,cumhaz2=base4,death.cumhaz=ddr
     )
rr <- count.history(rr)
dtable(rr,~death+status)

oo <- prob.exceedRecurrent(rr,1)
bplot(oo)
```

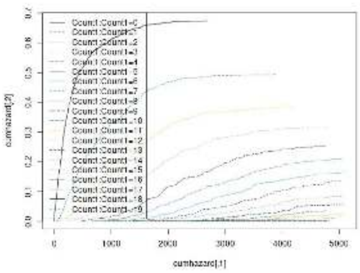
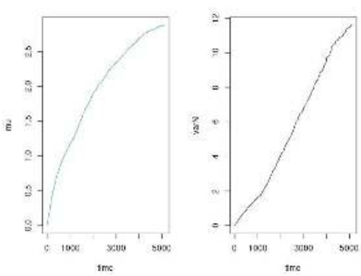



Figure 4: Recurrent events: probability of exceeding k events

Mean and variance of number of recurrent events



```r
par(mfrow=c(1,2))
with(oo,plot(time,mu,col=2,type="l"))
#
with(oo,plot(time,varN,type="l"))
```

```r
# Bivariate probability of exceeding
oo <- prob.exceedBiRecurrent(rr,1,2,exceed1=c(1,5,10),
    exceed2=c(1,2,3))
with(oo, matplot(time,pe1e2,type="s"))
nc <- ncol(oo$pe1e2)
legend("topleft",legend=colnames(oo$pe1e2),lty=1:nc,col=1:nc
    )
```

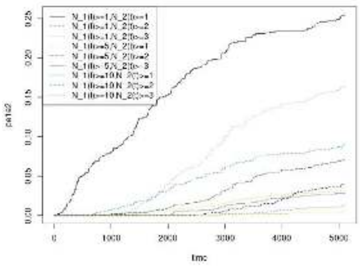


Figure 6: Recurrent events: probability of exceeding k events

Dependence between events: Covariance

Covariance among two types of events

$$\rho(t) = \frac{E(N_1^*(t)N_2^*(t)) - \mu_1(t)\mu_2(t)}{\mathrm{sd}(N_1^*(t))\mathrm{sd}(N_2^*(t))} \tag{5}$$

where

- $E(N_1^*(t)N_2^*(t))$.

  $$E(N_1^*(t)N_2^*(t)) = E(\int_0^t N_1^*(s-)dN_2^*(s)) + E(\int_0^t N_2^*(s-)dN_1^*(s))$$

  Recall that $N_1^*(t \wedge D)$ and $N_2^*(t \wedge D)$.

$$E(\int_0^t N_1^*(s-)dN_2^*(s)) = \sum_k E(\int_0^t kI(N_1^*(s-) = k)I(D \geq s)dN_2^*(s))$$

$$= \sum_k \int_0^t S(s)kP(N_1^*(s-) = k|D \geq s)E(dN_2^*(s)|N_1^*(s-) = k, D \geq s)$$

estimated by

$$\sum_k \int_0^t \hat{S}(s)k\frac{Y_1^k(s)}{Y_\bullet(s)}\frac{1}{Y_1^k(s)}d\tilde{N}_{2,k}(s),$$

- $Y_j^k(t) = \sum Y_i(t)I(N_{ji}^*(s-) = k)$ for $j = 1, 2$,
- $\tilde{N}_{j,k}(t) = \sum_i \int_0^t I(N_{ij^o}(s-) = k)dN_{ij}(s)$

  Estimate of \$ E(N$_1^*$(t) N$_2^*$(t))\$

$$\sum_k \int_0^t \hat{S}(s)k\frac{Y_1^k(s)}{Y_\bullet(s)}\frac{1}{Y_1^k(s)}d\tilde{N}_{2,k}(s) + \sum_k \int_0^t \hat{S}(s)k\frac{Y_2^k(s)}{Y_\bullet(s)}\frac{1}{Y_2^k(s)}d\tilde{N}_{1,k}(s).$$

- Without terminating event covariance is useful nonpar measure
- With terminating event dependence generated by terminating event.

- In reality what is of interest would be independence among survivors
  - if N_1 not predicitive for N_2
$$E(dN_2^*(t)|N_1^*(t-) = k, D \geq t) = E(dN_2^*(t)|D \geq t) \tag{6}$$
  - if N_2 not predicitive for N_1
$$E(dN_1^*(t)|N_2^*(t-) = k, D \geq t) = E(dN_1^*(t)|D \geq t) \tag{7}$$

If the two processes are independent among survivors then
$$E(dN_2^*(t)|N_1^*(t-) = k, D \geq t) = E(dN_2^*(t)|D \geq t) \tag{8}$$
so
$$E(\int_0^t N_1^*(s-)dN_2^*(s)) = \int_0^t S(s)E(N_1^*(s-)|D \geq s)E(dN_2^*(s)|D \geq s)$$
and
$$\int_0^t \hat{S}(s)\{\sum_k k\frac{Y_1^k(s)}{Y_\bullet(s)}\}\frac{1}{Y_\bullet(s)}dN_{2\bullet}(s),$$
where $N_{j\bullet}(t) = \sum_i \int_0^t dN_{j,i}(s)$.

Under the independence $E(N_1^*(t)N_2^*(t))$ is estimated
$$\int_0^t \hat{S}(s)\{\sum_k k\frac{Y_1^k(s)}{Y_\bullet(s)}\}\frac{1}{Y_\bullet(s)}dN_{2\bullet}(s) + \int_0^t \hat{S}(s)\{\sum_k k\frac{Y_2^k(s)}{Y_\bullet(s)}\}\frac{1}{Y_\bullet(s)}dN_{1\bullet}(s).$$

Both estimators, $\hat{E}(N_1^*(t)N_2^*(t))$ and $\hat{E}_I(N_1^*(t)N_2^*(t))$, as well as $\hat{E}(N_1^*(t))$ and $\hat{E}(N_2^*(t))$, have asymptotic expansions that can be written as a sum of iid processes, similarly to the arguments of Ghosh & Lin 2000, $\sum_i \Psi_i(t)$.

We can thus estimate the standard errors and of the estimators and their difference $\hat{E}(N_1^*(t)N_2^*(t)) - \hat{E}_I(N_1^*(t)N_2^*(t))$.

Terms for

- N1 -> N2 : $E(\int_0^t N_1^*(s-)dN_2^*(s))$
- N2 -> N1 : $E(\int_0^t N_2^*(s-)dN_1^*(s))$

```
rr$strata <- 1
dtable(rr,~death+status)

covrp <- covarianceRecurrent(rr,1,2,status="status",death="
    death",
          start="entry",stop="time",id="id",names.count="
              Count")
par(mfrow=c(1,3))
plot(covrp)

# with strata, each strata in matrix column, provides basis
    for fast Bootstrap
covrpS <- covarianceRecurrentS(rr,1,2,status="status",death=
    "death",
   start="entry",stop="time",strata="strata",id="id",names.
       count="Count")
```

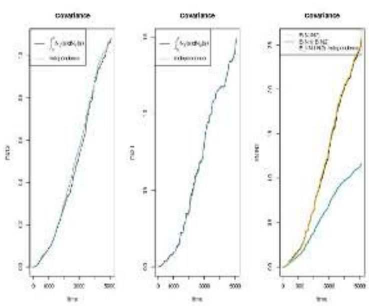


Figure 7: Covariance between events

Bootstrap standard errors for terms

First fitting the model again to get our estimates of interst, and then computing them for some specific time-points

```r
times <- seq(500,5000,500)

coo1 <- covarianceRecurrent(rr,1,2,status="status",start="
    entry",stop="time")
#
mug <- Cpred(cbind(coo1$time,coo1$EN1N2),times)[,2]
mui <- Cpred(cbind(coo1$time,coo1$EIN1N2),times)[,2]
mu2.1 <- Cpred(cbind(coo1$time,coo1$mu2.1),times)[,2]
mu2.i <- Cpred(cbind(coo1$time,coo1$mu2.i),times)[,2]
mu1.2 <- Cpred(cbind(coo1$time,coo1$mu1.2),times)[,2]
mu1.i <- Cpred(cbind(coo1$time,coo1$mu1.i),times)[,2]
cbind(mu2.1,mu2.i)
cbind(mu1.2,mu1.i)
```

```
          mu2.1      mu2.i
 [1,] 0.04101096 0.03656491
 [2,] 0.09303668 0.08572694
 [3,] 0.22613687 0.21906324
 [4,] 0.35727148 0.34562539
 [5,] 0.60258982 0.59071900
 [6,] 0.80089841 0.79020220
 [7,] 1.03031183 1.03424672
 [8,] 1.16860632 1.16686717
 [9,] 1.25782175 1.25105963
[10,] 1.38716306 1.40250244
          mu1.2      mu1.i
 [1,] 0.03501045 0.03259566
 [2,] 0.08803686 0.08526834
 [3,] 0.16709531 0.16634828
 [4,] 0.27720710 0.29485672
 [5,] 0.38034407 0.41985665
 [6,] 0.53057410 0.56459585
 [7,] 0.69387628 0.72234676
 [8,] 0.87226707 0.88771625
 [9,] 0.96949736 0.99728527
[10,] 1.06074066 1.06854228
```

To get the bootstrap standard errors there is a quick memory demanding function (with S for speed and strata) BootcovariancerecurrenceS and slow function that goes through the loops in R Bootcovariancerecurrence.

```r
bt1 <- BootcovariancerecurrenceS(rr,1,2,status="status",
    start="entry",stop="time",K=100,times=times)
#bt1 <-
    BootcovariancerecurrenceS(rr,1,2,status="status",start="entry",stop="time",K=K,times=times)

output <- list(bt1=bt1,mug=mug,mui=mui,
bse.mug=bt1$se.mug,bse.mui=bt1$se.mui,
dmugi=mug-mui,
bse.dmugi=apply(bt1$EN1N2-bt1$EIN1N2,1,sd),
mu2.1 = mu2.1 , mu2.i = mu2.i , dmu2.i=mu2.1-mu2.i,
mu1.2 = mu1.2 , mu1.i = mu1.i , dmu1.i=mu1.2-mu1.i,
bse.mu2.1=apply(bt1$mu2.i,1,sd), bse.mu2.1=apply(bt1$mu2
    .1,1,sd),
bse.dmu2.i=apply(bt1$mu2.1-bt1$mu2.i,1,sd),
```

```
bse.mu1.2=apply(bt1$mu1.2,1,sd), bse.mu1.i=apply(bt1$mu1.i
    ,1,sd),
bse.dmu1.i=apply(bt1$mu1.2-bt1$mu1.i,1,sd)
)
```

We then look at the test for overall dependence in the different time-points. We here have no suggestion of dependence.

```
tt <- output$dmugi/output$bse.dmugi
cbind(times,2*(1-pnorm(abs(tt))))
```

```
     times
 [1,]   500 0.3572253
 [2,]  1000 0.4577012
 [3,]  1500 0.7136132
 [4,]  2000 0.7956959
 [5,]  2500 0.3837459
 [6,]  3000 0.5134406
 [7,]  3500 0.4209237
 [8,]  4000 0.7632914
 [9,]  4500 0.6836682
[10,]  5000 0.6598813
```

We can also take out the specific components for whether N_1 is predictive for N_2 and vice versa. We here have no suggestion of dependence.

```
t21 <- output$dmu1.i/output$bse.dmu1.i
t12 <- output$dmu2.i/output$bse.dmu2.i
cbind(times,2*(1-pnorm(abs(t21))),2*(1-pnorm(abs(t12))))
```

```
     times
 [1,]   500 0.71706002 0.3918872
 [2,]  1000 0.81454942 0.3202626
 [3,]  1500 0.95715638 0.6006314
 [4,]  2000 0.21300406 0.4942293
 [5,]  2500 0.02182129 0.6086128
 [6,]  3000 0.11688970 0.6805457
 [7,]  3500 0.25587816 0.8965495
 [8,]  4000 0.63373150 0.9578608
 [9,]  4500 0.41743073 0.8548733
[10,]  5000 0.83041113 0.6805618
```

We finally plot the boostrap samples

```
par(mfrow=c(1,2))
matplot(bt1$time,bt1$EN1N2,type="l",lwd=0.3)
matplot(bt1$time,bt1$EIN1N2,type="l",lwd=0.3)
```

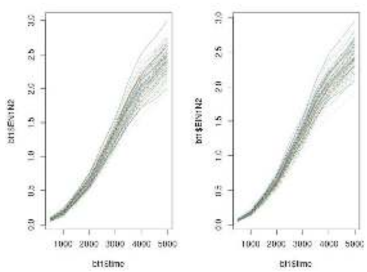


Figure 8: Bootstrap samples

Looking at other simulations with dependence

Using the normally distributed random effects we plot 4 different settings. We have variance 0.5 for all random effects and change the correlation. We let the correlation between the random effect associated with N_1 and N_2 be denoted ρ_{12} and the correlation between the random effects associated between N_j and D the terminal event be denoted as ρ_{j3}, and organize all correlation in a vector $\rho = (\rho_{12}, \rho_{13}, \rho_{23})$.

- Scenario I $\rho = (0, 0.0, 0.0)$ Independence among all efects.
- Scenario II $\rho = (0, 0.5, 0.5)$ Independence among survivors but dependence on terminal event
- Scenario III $\rho = (0.5, 0.5, 0.5)$ Positive dependence among survivors and dependence on terminal event
- Scenario IV $\rho = (-0.4, 0.5, 0.5)$ Negative dependence among survivors and positive dependence on terminal event

```
par(mfrow=c(2,2))

data(base1cumhaz)
data(base4cumhaz)
data(drcumhaz)
dr <- drcumhaz
base1 <- base1cumhaz
base4 <- base4cumhaz

var.z <- c(0.5,0.5,0.5)
# death related to both causes in same way
cor.mat <- corM <- rbind(c(1.0, 0.0, 0.0),
           c(0.0, 1.0, 0.0),
           c(0.0, 0.0, 1.0))
rr <- simRecurrentII(3000,base1,base4,death.cumhaz=dr,var.z=
    var.z,cor.mat=cor.mat,dependence=2)
rr <- count.history(rr,types=1:2)
cor(attr(rr,"z"))
coo <- covarianceRecurrent(rr,1,2,status="status",start="
    entry",stop="time")
par(mfrow=c(2,2))
with(coo, {
     plot(time, EN1N2, type = "l", lwd = 2,lty=1,ylab="",
         xlab="time (a)")
     lines(time, EN1EN2, col = 2, lwd = 2,lty=2)
     lines(time, EIN1N2, col = 3, lwd = 2,lty=3)
  })
  legend("topleft", c("E(N1N2)", "E(N1) E(N2) ", "E_I(N1 N2)-
      independence"),lty = 1:3, col = 1:3)
  title(main ="Scenario I")


var.z <- c(0.5,0.5,0.5)
# death related to both causes in same way
cor.mat <- corM <- rbind(c(1.0, 0.0, 0.5),
           c(0.0, 1.0, 0.5),
           c(0.5, 0.5, 1.0))
rr <- simRecurrentII(3000,base1,base4,death.cumhaz=dr,
            var.z=var.z,cor.mat=cor.mat,dependence=2)
rr <- count.history(rr,types=1:2)
coo <- covarianceRecurrent(rr,1,2,status="status",start="
    entry",stop="time")
with(coo, {
     plot(time, EN1N2, type = "l", lwd = 2,lty=1,ylab="",
         xlab="time (b)")
     lines(time, EN1EN2, col = 2, lwd = 2,lty=2)
```

```
     lines(time, EIN1N2, col = 3, lwd = 2,lty=3)
  })
  legend("topleft", c("E(N1N2)", "E(N1) E(N2) ", "E_I(N1 N2)-
      independence"),lty = 1:3, col = 1:3)
  title(main ="Scenario II")

var.z <- c(0.5,0.5,0.5)
# positive dependence for N1 and N2 all related in same way
cor.mat <- corM <- rbind(c(1.0, 0.5, 0.5),
           c(0.5, 1.0, 0.5),
           c(0.5, 0.5, 1.0))
rr <- simRecurrentII(3000,base1,base4,death.cumhaz=dr,
            var.z=var.z,cor.mat=cor.mat,dependence=2)
rr <- count.history(rr,types=1:2)
coo <- covarianceRecurrent(rr,1,2,status="status",start="
    entry",stop="time")
with(coo, {
     plot(time, EN1N2, type = "l", lwd = 2,lty=1,ylab="",
         xlab="time (d)")
     lines(time, EN1EN2, col = 2, lwd = 2,lty=2)
     lines(time, EIN1N2, col = 3, lwd = 2,lty=3)
  })
  legend("topleft", c("E(N1N2)", "E(N1) E(N2) ", "E_I(N1 N2)-
      independence"),lty = 1:3, col = 1:3)
  title(main ="Scenario III")


var.z <- c(0.5,0.5,0.5)
# negative dependence for N1 and N2 all related in same way
cor.mat <- corM <- rbind(c(1.0, -0.4, 0.5),
           c(-0.4, 1.0, 0.5),
           c(0.5, 0.5, 1.0))
rr <- simRecurrentII(3000,base1,base4,death.cumhaz=dr,
            var.z=var.z,cor.mat=cor.mat,dependence=2)
rr <- count.history(rr,types=1:2)
coo <- covarianceRecurrent(rr,1,2,status="status",start="
    entry",stop="time")
with(coo, {
     plot(time, EN1N2, type = "l", lwd = 2,lty=1,ylab="",
         xlab="time (d)")
     lines(time, EN1EN2, col = 2, lwd = 2,lty=2)
     lines(time, EIN1N2, col = 3, lwd = 2,lty=3)
})
legend("topleft", c("E(N1N2)", "E(N1) E(N2) ", "E_I(N1 N2)-
    independence"),lty = 1:3, col = 1:3)
title(main ="Scenario IV")
```

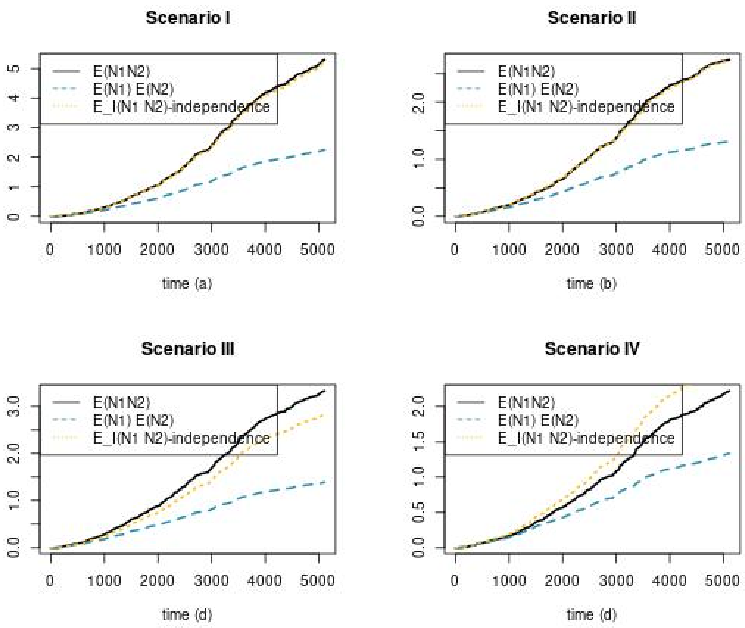


Figure 9: Bootstrap samples