

Recurrent Events

Klaus Holst & Thomas Scheike

February 18, 2020

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 phenomenon 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 where 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 where the sum-structure can be specified 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 $\mu_1(t) = E(N_1^*(t))$

$$\int_0^t S(u) dR_1(u) \quad (1)$$

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 $R_1(t)$

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

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

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

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)

```

1 library(mets)
2 set.seed(1000) # to control output in simulations for
  p-values below.
3
4 data(base1cumhaz)
5 data(base4cumhaz)
6 data(drcumhaz)
7 ddr <- drcumhaz
8 base1 <- base1cumhaz
9 base4 <- base4cumhaz
10 rr <- simRecurrent(1000,base1,death.cumhaz=ddr)
11 rr$x <- rnorm(nrow(rr))
12 rr$strata <- floor((rr$id-0.01)/500)
13 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      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      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 events and their type, and death the timing of death.

```

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

```

```

5 bplot(dr,se=TRUE)
6 title(main="death")
7 bplot(xr,se=TRUE)
8 # robust standard errors
9 rxr <- robust.phreg(xr,fixbeta=1)
10 bplot(rxr,se=TRUE,robust=TRUE,add=TRUE,col=4)
11
12 # marginal mean of expected number of recurrent events
13 out <- recurrentMarginal(xr,dr)
14 bplot(out,se=TRUE,ylab="marginal mean",col=2)

```

We can do the same with strata

```

1 xr <- phreg(Surv(entry,time,status)~strata(strata)+cluster(
  id),data=rr)
2 dr <- phreg(Surv(entry,time,death)~strata(strata)+cluster(id
  ),data=rr)
3 par(mfrow=c(1,3))
4 bplot(dr,se=TRUE)
5 title(main="death")
6 bplot(xr,se=TRUE)
7 rxr <- robust.phreg(xr,fixbeta=1)
8 bplot(rxr,se=TRUE,robust=TRUE,add=TRUE,col=1:2)
9
10 out <- recurrentMarginal(xr,dr)
11 bplot(out,se=TRUE,ylab="marginal mean",col=1:2)

```

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

```

1 # cox case
2 xr <- phreg(Surv(entry,time,status)~x+cluster(id),data=rr)
3 dr <- phreg(Surv(entry,time,death)~x+cluster(id),data=rr)
4 par(mfrow=c(1,3))
5 bplot(dr,se=TRUE)
6 title(main="death")
7 bplot(xr,se=TRUE)
8 rxr <- robust.phreg(xr)
9 bplot(rxr,se=TRUE,robust=TRUE,add=TRUE,col=1:2)
10
11 out <- recurrentMarginal(xr,dr)
12 bplot(out,se=TRUE,ylab="marginal mean",col=1:2)
13
14 # predictions without se's
15 outX <- recmarg(xr,dr,Xr=1,Xd=1)
16 bplot(outX,add=TRUE,col=3)

```

Other marginal properties

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

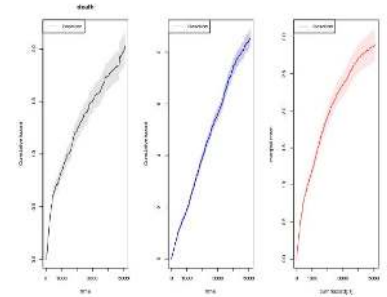


Figure 1: Marginal mean for number of type 1 events, rate for death (panel (a)), rate for type 1 among survivors (panel (b)), and marginal mean (panel (c)).

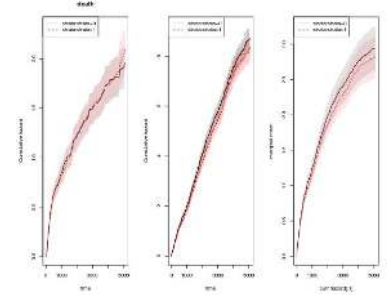


Figure 2: Recurrent events

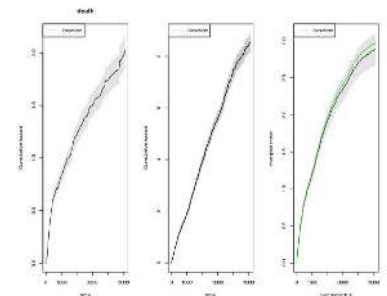


Figure 3: Recurrent events with cox models for rates.

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 $E((N_1^*(t))^2)$

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

Probability 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

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

```

1 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))
2 rr <- simRecurrent(1000,base1,cumhaz2=base4,death.cumhaz=ddr
  )
3 rr <- count.history(rr)
4 dtable(rr,~death+status)
5
6 oo <- prob.exceedRecurrent(rr,1)
7 bplot(oo)

```

```

1 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))
2 rr <- simRecurrent(1000,base1,cumhaz2=base4,death.cumhaz=ddr
  )
3 rr <- count.history(rr)
4 dtable(rr,~death+status)
5
6 oo <- prob.exceedRecurrent(rr,1)
7 bplot(oo)

```

Mean and variance of number of recurrent events

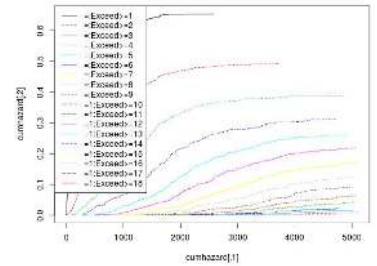


Figure 4: Recurrent events: probability of exceeding k events

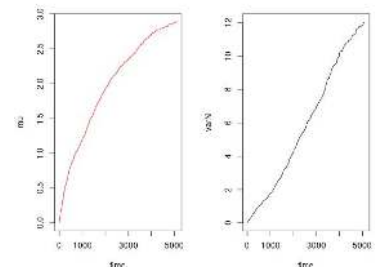


Figure 5: Recurrent events: mean and variance

```

1 par(mfrow=c(1,2))
2 with(oo,plot(time,mu,col=2,type="l"))
3 #
4 with(oo,plot(time,varN,type="l"))

```

```

1 # Bivariate probability of exceeding
2 oo <- prob.exceedBiRecurrent(rr,1,2,exceed1=c(1,5,10),
   exceed2=c(1,2,3))
3 with(oo, matplot(time,pele2,type="s"))
4 nc <- ncol(oo$pele2)
5 legend("topleft",legend=colnames(oo$pele2),lty=1:nc,col=1:nc
   )

```

Dependence between events: Covariance

Covariance among two types of events

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

where

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

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

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

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

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

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

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.

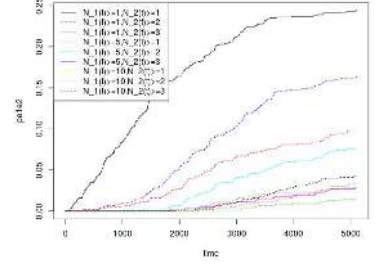


Figure 6: Recurrent events: probability of exceeding k events

- In reality what is of interest would be independence among survivors
 - if N_1 not predictive for N_2

$$E(dN_2^*(t)|N_1^*(t-) = k, D \geq t) = E(dN_2^*(t)|D \geq t) \quad (6)$$

- if N_2 not predictive for N_1

$$E(dN_1^*(t)|N_2^*(t-) = k, D \geq t) = E(dN_1^*(t)|D \geq t) \quad (7)$$

If the two processes are independent among survivors then

$$E(dN_2^*(t)|N_1^*(t-) = k, D \geq t) = E(dN_2^*(t)|D \geq t) \quad (8)$$

so

$$E\left(\int_0^t N_1^*(s-)dN_2^*(s)\right) = \int_0^t S(s)E(N_1^*(s-)|D \geq s)E(dN_2^*(s)|D \geq s)$$

and

$$\int_0^t \hat{S}(s) \left\{ \sum_k k \frac{Y_1^k(s)}{Y_\bullet(s)} \right\} \frac{1}{Y_\bullet(s)} dN_{2\bullet}(s),$$

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

$$\int_0^t \hat{S}(s) \left\{ \sum_k k \frac{Y_1^k(s)}{Y_\bullet(s)} \right\} \frac{1}{Y_\bullet(s)} dN_{2\bullet}(s) + \int_0^t \hat{S}(s) \left\{ \sum_k k \frac{Y_2^k(s)}{Y_\bullet(s)} \right\} \frac{1}{Y_\bullet(s)} dN_{1\bullet}(s).$$

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

- $N_1 \rightarrow N_2$: $E\left(\int_0^t N_1^*(s-)dN_2^*(s)\right)$
- $N_2 \rightarrow N_1$: $E\left(\int_0^t N_2^*(s-)dN_1^*(s)\right)$

```

1 rr$strata <- 1
2 dtable(rr,~death+status)
3
4 covrp <- covarianceRecurrent(rr,1,2,status="status",death="
   death",
5     start="entry",stop="time",id="id",names.count="
   Count")
6 par(mfrow=c(1,3))
7 plot(covrp)
8
9 # with strata, each strata in matrix column, provides basis
   for fast Bootstrap
10 covrpS <- covarianceRecurrentS(rr,1,2,status="status",death=
   "death",
11   start="entry",stop="time",strata="strata",id="id",names.
   count="Count")

```

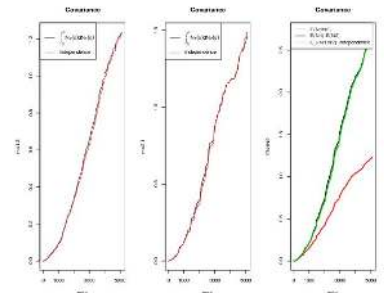


Figure 7: Covariance between events

Bootstrap standard errors for terms

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

```

1 times <- seq(500,5000,500)
2
3 cool <- covarianceRecurrent(rr,1,2,status="status",start="
  entry",stop="time")
4 #
5 mug <- Cpred(cbind(cool$time,cool$EN1N2),times)[,2]
6 mui <- Cpred(cbind(cool$time,cool$EIN1N2),times)[,2]
7 mu2.1 <- Cpred(cbind(cool$time,cool$mu2.1),times)[,2]
8 mu2.i <- Cpred(cbind(cool$time,cool$mu2.i),times)[,2]
9 mu1.2 <- Cpred(cbind(cool$time,cool$mu1.2),times)[,2]
10 mu1.i <- Cpred(cbind(cool$time,cool$mu1.i),times)[,2]
11 cbind(mu2.1,mu2.i)
12 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`.

```

1 bt1 <- BootcovariancerecurrenceS(rr,1,2,status="status",
  start="entry",stop="time",K=100,times=times)
2 #bt1 <-
  BootcovariancerecurrenceS(rr,1,2,status="status",start="entry",stop="time",K=K,times=times)
3
4 output <- list(bt1=bt1,mug=mug,mui=mui,
5 bse.mug=bt1$se.mug,bse.mui=bt1$se.mui,
6 dmugi=mug-mui,
7 bse.dmugi=apply(bt1$EN1N2-bt1$EIN1N2,1,sd),
8 mu2.1 = mu2.1 , mu2.i = mu2.i , dmu2.i=mu2.1-mu2.i,
9 mu1.2 = mu1.2 , mu1.i = mu1.i , dmu1.i=mu1.2-mu1.i,
10 bse.mu2.1=apply(bt1$mu2.i,1,sd), bse.mu2.1=apply(bt1$mu2
  .1,1,sd),
11 bse.dmu2.i=apply(bt1$mu2.1-bt1$mu2.i,1,sd),

```



```

12 bse.mu1.2=apply(bt1$mu1.2,1,sd), bse.mu1.i=apply(bt1$mu1.i
    ,1,sd),
13 bse.dmu1.i=apply(bt1$mu1.2-bt1$mu1.i,1,sd)
14 )

```

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

```

1 tt <- output$dmugi/output$bse.dmugi
2 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.

```

1 t21 <- output$dmu1.i/output$bse.dmu1.i
2 t12 <- output$dmu2.i/output$bse.dmu2.i
3 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 bootstrap samples

```

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

```

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 $\rho = (\rho_{12}, \rho_{13}, \rho_{23})$.

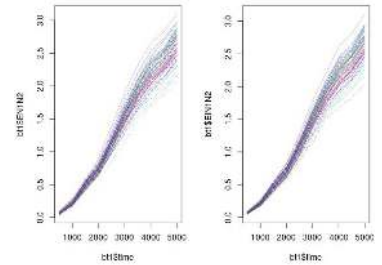


Figure 8: Bootstrap samples

- Scenario I $\rho = (0, 0.0, 0.0)$ Independence among all effects.
- 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

```

1  par(mfrow=c(2,2))
2
3  data(base1cumhaz)
4  data(base4cumhaz)
5  data(drcumhaz)
6  dr <- drcumhaz
7  base1 <- base1cumhaz
8  base4 <- base4cumhaz
9
10 var.z <- c(0.5,0.5,0.5)
11 # death related to both causes in same way
12 cor.mat <- corM <- rbind(c(1.0, 0.0, 0.0),
13                          c(0.0, 1.0, 0.0),
14                          c(0.0, 0.0, 1.0))
15 rr <- simRecurrentII(3000,base1,base4,death.cumhaz=dr,var.z=
16                     var.z,cor.mat=cor.mat,dependence=2)
17 rr <- count.history(rr,types=1:2)
18 cor(attr(rr,"z"))
19 coo <- covarianceRecurrent(rr,1,2,status="status",start="
20 entry",stop="time")
21 par(mfrow=c(2,2))
22 with(coo, {
23   plot(time, EN1N2, type = "l", lwd = 2,lty=1,ylab="",
24        xlab="time (a)")
25   lines(time, EN1EN2, col = 2, lwd = 2,lty=2)
26   lines(time, EIN1N2, col = 3, lwd = 2,lty=3)
27 })
28 legend("topleft", c("E(N1N2)", "E(N1) E(N2) ", "E_I(N1 N2)-
29 independence"),lty = 1:3, col = 1:3)
30 title(main = "Scenario I")
31
32
33 var.z <- c(0.5,0.5,0.5)
34 # death related to both causes in same way
35 cor.mat <- corM <- rbind(c(1.0, 0.0, 0.5),
36                          c(0.0, 1.0, 0.5),
37                          c(0.5, 0.5, 1.0))
38 rr <- simRecurrentII(3000,base1,base4,death.cumhaz=dr,
39                     var.z=var.z,cor.mat=cor.mat,dependence=2)
40 rr <- count.history(rr,types=1:2)
41 coo <- covarianceRecurrent(rr,1,2,status="status",start="
42 entry",stop="time")
43 with(coo, {
44   plot(time, EN1N2, type = "l", lwd = 2,lty=1,ylab="",
45        xlab="time (b)")
46   lines(time, EN1EN2, col = 2, lwd = 2,lty=2)

```

```

41     lines(time, EIN1N2, col = 3, lwd = 2,lty=3)
42   })
43   legend("topleft", c("E(N1N2)", "E(N1) E(N2) ", "E_I(N1 N2)-
44     independence"),lty = 1:3, col = 1:3)
45   title(main ="Scenario II")
46
47   var.z <- c(0.5,0.5,0.5)
48   # positive dependence for N1 and N2 all related in same way
49   cor.mat <- corM <- rbind(c(1.0, 0.5, 0.5),
50     c(0.5, 1.0, 0.5),
51     c(0.5, 0.5, 1.0))
52   rr <- simRecurrentII(3000,base1,base4,death.cumhaz=dr,
53     var.z=var.z,cor.mat=cor.mat,dependence=2)
54   rr <- count.history(rr,types=1:2)
55   coo <- covarianceRecurrent(rr,1,2,status="status",start="
56     entry",stop="time")
57   with(coo, {
58     plot(time, EN1N2, type = "l", lwd = 2,lty=1,ylab="",
59       xlab="time (d)")
60     lines(time, EN1EN2, col = 2, lwd = 2,lty=2)
61     lines(time, EIN1N2, col = 3, lwd = 2,lty=3)
62   })
63   legend("topleft", c("E(N1N2)", "E(N1) E(N2) ", "E_I(N1 N2)-
64     independence"),lty = 1:3, col = 1:3)
65   title(main ="Scenario III")
66
67   var.z <- c(0.5,0.5,0.5)
68   # negative dependence for N1 and N2 all related in same way
69   cor.mat <- corM <- rbind(c(1.0, -0.4, 0.5),
70     c(-0.4, 1.0, 0.5),
71     c(0.5, 0.5, 1.0))
72   rr <- simRecurrentII(3000,base1,base4,death.cumhaz=dr,
73     var.z=var.z,cor.mat=cor.mat,dependence=2)
74   rr <- count.history(rr,types=1:2)
75   coo <- covarianceRecurrent(rr,1,2,status="status",start="
76     entry",stop="time")
77   with(coo, {
78     plot(time, EN1N2, type = "l", lwd = 2,lty=1,ylab="",
79       xlab="time (d)")
80     lines(time, EN1EN2, col = 2, lwd = 2,lty=2)
81     lines(time, EIN1N2, col = 3, lwd = 2,lty=3)
82   })
83   legend("topleft", c("E(N1N2)", "E(N1) E(N2) ", "E_I(N1 N2)-
84     independence"),lty = 1:3, col = 1:3)
85   title(main ="Scenario IV")

```

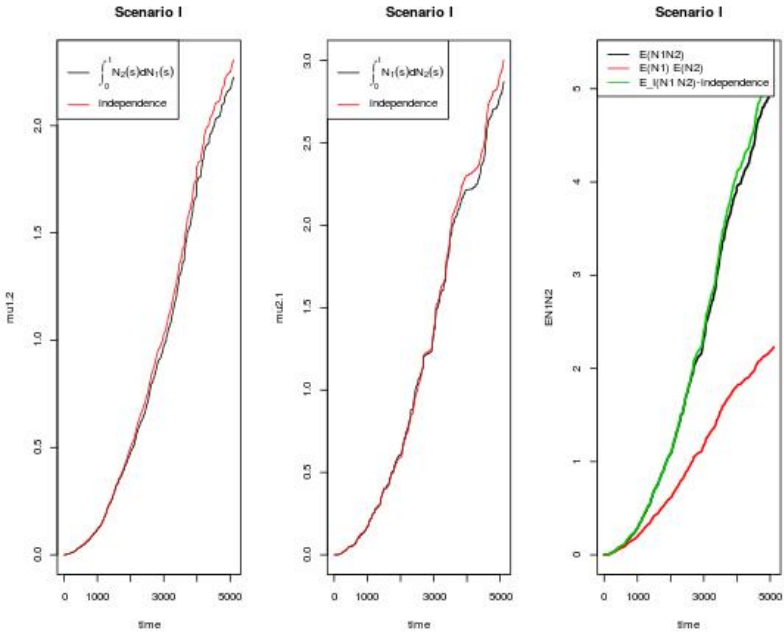


Figure 9: Bootstrap samples