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 $Z_1, Z_2, Z_d$ 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 specified via a matrix cor.mat. Then we compute $\tilde{Z}_j = \sum_k Z_{cor.mat}(j,k)$ for $j = 1, 2, 3$ (dependence=3) Then the intensities are
  
  $Z_1 \lambda_1(t)$
  $Z_2 \lambda_2(t)$
  $Z_3 \lambda_D(t)$

• 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 $\mu_1(t) = E(N_1^*(t))$

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

(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 $R_1(t)$

    $\hat{R}_1(t) = \sum_i \int_0^t \frac{1}{Y_\bullet(s)} dN_1(s)$

(2)

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

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

(3)
Cook & Lawless (1997), and developed further in Gosh & Lin (2000).

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

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

with mean-zero processes

- $M_i^d(t) = N_i^D(t) - \int_0^t Y_i(s) d\Lambda^D(s),$
- $M_{1i}(t) = N_{1i}(t) - \int_0^t Y_i(s) dR_1(s).$

as in Gosh & Lin (2000)

```r
library(mets)
set.seed(1000) # to control output in simulations 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))
```

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

```r
# 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
rrx <- robust.phreg(xr, fixbeta=1)
bplot(rrx, 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)

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)
rrx <- robust.phreg(xr, fixbeta=1)
bplot(rrx, se=TRUE, robust=TRUE, add=TRUE, col=1:2)
out <- recurrentMarginal(xr, dr)
bplot(out, se=TRUE, ylab="marginal mean", col=1:2)

Further, 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)
rrx <- robust.phreg(xr)
bplot(rrx, se=TRUE, robust=TRUE, add=TRUE, col=1:2)
out <- recurrentMarginal(xr, dr)
bplot(out, se=TRUE, ylab="marginal mean", col=1:2)

# predictions without se's
outX <- recmarg(xr, dr, Xr=1, Xd=1)
bplot(outX, add=TRUE, col=3)

Other marginal properties

- \( P(N^*_k(t) \geq k) \)
  - cumulative incidence of \( T_k = \inf \{ t : N^*_k(t) = k \} \) with competing \( D \).
We note also that $N_i^* (t)^2$ can be written as

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

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

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

and estimated by

$$\hat{\mu}_{1,2}(t) = \sum_{k=0}^{K} \int_0^t \hat{S}(s) f(k) \frac{Y^{-1}_k(s)}{Y^{-1}_k(s)} dN_i^*(s) = \sum_{k=1}^{n} \int_0^t \hat{S}(s) f(N_i^*(s-)) \frac{1}{Y_k(s)} dN_i^*(s),$$

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

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

Probability of exceeding "$k$"

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

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

suggesting that its mean can be computed as

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

and estimated by

$$\hat{F}_k(t) = \int_0^t \hat{S}(s) \frac{Y^{-1}_{k-1}(s)}{Y^{-1}_{k-1}(s)} \frac{1}{Y^{-1}_{k-1}(s)} dN_i^*(s)$$

---

```r
library(pec)

# Correlation matrix
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))

# Simulate recurrent events
rr <- simRecurrent(1000, base1, cumhaz2 = base4, death.cumhaz = ddr)

# Count history
rr <- count.history(rr)

dtable(rr, ~ death + status)

# Probability of exceeding k events
oo <- prob.exceedRecurrent(rr, 1)
bplot(oo)
```

---

**Figure 4:** Recurrent events: probability of exceeding k events

---

**Figure 5:** Recurrent events: probability of number of recurrent events
# 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)

Dependence between events: Covariance

Covariance among two types of events

\[
\rho(t) = \frac{E(N_1^*(t)N_2^*(t)) - \mu_1(t)\mu_2(t)}{\text{sd}(N_1^*(t))\text{sd}(N_2^*(t))}
\]  

(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

\[
\sum_k \int_0^t \hat{S}(s)kY_j^k(s)\frac{1}{Y_j^*_j(s)}\frac{1}{Y_j^*_j(s)}d\hat{N}_j^k(s),
\]

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

Estimate of \$E(N_1^*(t)N_2^*(t))\$

\[
\sum_k \int_0^t \hat{S}(s)kY_1^k(s)\frac{1}{Y_1^*_1(s)}\frac{1}{Y_1^*_1(s)}d\hat{N}_2^k(s) + \sum_k \int_0^t \hat{S}(s)kY_2^k(s)\frac{1}{Y_2^*_2(s)}\frac{1}{Y_2^*_2(s)}d\hat{N}_1^k(s).
\]

- 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 predictive for \( N_2 \)
    \[
    E(dN^*_2(t)|N^*_1(t-) = k, D \geq t) = E(dN^*_2(t)|D \geq t)
    \]
    \[\text{(6)}\]
  - if \( N_2 \) not predictive for \( N_1 \)
    \[
    E(dN^*_1(t)|N^*_2(t-) = k, D \geq t) = E(dN^*_1(t)|D \geq t)
    \]
    \[\text{(7)}\]

If the two processes are independent among survivors then
\[
E(dN^*_1(t)|N^*_2(t-) = k, D \geq t) = E(dN^*_1(t)|D \geq t)
\]
\[\text{(8)}\]

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
\[
\int_0^t \hat{S}(s)\{\sum_k k\frac{Y^k_1(s)}{Y_1(s)}\} \frac{1}{Y_2(s)}dN^*_2(s),
\]
where \( N^*_i(t) = \sum_i \int_0^t dN_{i,j}(s) \).

Under the independence \( E(N^*_1(t)N^*_2(t)) \) is estimated
\[
\int_0^t \hat{S}(s)\{\sum_k k\frac{Y^k_1(s)}{Y_1(s)}\} \frac{1}{Y_2(s)}dN^*_2(s) + \int_0^t \hat{S}(s)\{\sum_k k\frac{Y^k_2(s)}{Y_2(s)}\} \frac{1}{Y_1(s)}dN^*_1(s).
\]

Both estimators, \( \hat{E}(N^*_1(t)N^*_2(t)) \) and \( \hat{E}(N^*_1(t)N^*_1(t)) \), 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, \( \sum \Psi_i(t) \).

We can thus estimate the standard errors and of the estimators and their difference \( \hat{E}(N^*_1(t)N^*_2(t)) - \hat{E}(N^*_1(t)N^*_1(t)) \).

Terms for
- \( N_1 \rightarrow N_2 \): \( E(\int_0^t N^*_1(s-)dN^*_2(s)) \)
- \( N_2 \rightarrow N_1 \): \( E(\int_0^t N^*_2(s-)dN^*_1(s)) \)

```r
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 interest, 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 0.21906234
[4,] 0.35727148 0.34562539
[5,] 0.60258982 0.59071900
[6,] 0.80089841 0.79020220
[7,] 1.03031183 1.03424672
[8,] 1.16860632 1.16687117
[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 0.56495885
[7,] 0.69387628 0.72346767
[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.

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

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

<table>
<thead>
<tr>
<th>times</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,] 500</td>
<td>0.3572253</td>
<td>0.3918872</td>
</tr>
<tr>
<td>[2,] 1000</td>
<td>0.4577012</td>
<td>0.3202626</td>
</tr>
<tr>
<td>[3,] 1500</td>
<td>0.7136132</td>
<td>0.6006314</td>
</tr>
<tr>
<td>[4,] 2000</td>
<td>0.7956959</td>
<td>0.6086128</td>
</tr>
<tr>
<td>[5,] 2500</td>
<td>0.3837459</td>
<td>0.8965495</td>
</tr>
<tr>
<td>[6,] 3000</td>
<td>0.5134406</td>
<td>0.8548733</td>
</tr>
<tr>
<td>[7,] 3500</td>
<td>0.4209237</td>
<td>0.8505618</td>
</tr>
<tr>
<td>[8,] 4000</td>
<td>0.7632914</td>
<td>0.8548733</td>
</tr>
<tr>
<td>[9,] 4500</td>
<td>0.6836682</td>
<td>0.8548733</td>
</tr>
<tr>
<td>[10,] 5000</td>
<td>0.6598813</td>
<td>0.8548733</td>
</tr>
</tbody>
</table>

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.

```r
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))))
```

<table>
<thead>
<tr>
<th>times</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,] 500</td>
<td>0.71706002</td>
<td>0.3918872</td>
<td>0.3918872</td>
</tr>
<tr>
<td>[2,] 1000</td>
<td>0.81454942</td>
<td>0.3202626</td>
<td>0.3202626</td>
</tr>
<tr>
<td>[3,] 1500</td>
<td>0.95715638</td>
<td>0.6006314</td>
<td>0.6006314</td>
</tr>
<tr>
<td>[4,] 2000</td>
<td>0.21300406</td>
<td>0.4942293</td>
<td>0.4942293</td>
</tr>
<tr>
<td>[5,] 2500</td>
<td>0.02168212</td>
<td>0.6086128</td>
<td>0.6086128</td>
</tr>
<tr>
<td>[6,] 3000</td>
<td>0.11686970</td>
<td>0.6806457</td>
<td>0.6806457</td>
</tr>
<tr>
<td>[7,] 3500</td>
<td>0.25587816</td>
<td>0.8965495</td>
<td>0.8965495</td>
</tr>
<tr>
<td>[8,] 4000</td>
<td>0.63373150</td>
<td>0.9578608</td>
<td>0.9578608</td>
</tr>
<tr>
<td>[9,] 4500</td>
<td>0.41743073</td>
<td>0.8548733</td>
<td>0.8548733</td>
</tr>
<tr>
<td>[10,] 5000</td>
<td>0.83041113</td>
<td>0.6805618</td>
<td>0.6805618</td>
</tr>
</tbody>
</table>

We finally plot the boostrap samples

```r
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)
```

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 \( \rho_{12} \) and the correlation between the random effects associated between \( N_j \) and \( D \) the terminal event be denoted as \( \rho_{j3} \), and organize all correlation in a vector \( \rho = (\rho_{12}, \rho_{13}, \rho_{23}) \).
• Scenario I $\rho = (0, 0, 0, 0)$ 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

```r
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(0.5, 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)
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, EN1N2, 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, EN1N2, 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