Saemix with competing risks

Emmanuelle Comets

23/11/2021

Setup

- set up work directories
- two versions toggled by testMode
  - if testMode is FALSE, load the functions in R
  - if testMode is TRUE, load the library in a dev_mode environment
- aim: implement competing risk models

Testing library

Competing risks model - cause-specific assumption

- Simulation
  - both competing risks simulated with a constant hazard
  - small IIV added to the second event (but doesn't seem to change anything much)

```
## Simulation
constantHazardTTE.simul<-function(psi,id,xidep) {
  T<-xidep[,1]
  N <- nrow(psi)
  Nj <- length(T)
  censoringtime <- 3 # eg years
  lam1 <- psi[id,1]
  lam2 <- psi[id,2]
  Vj<-runif(Nj)
  T1 <- -log(Vj)/lam1
  Vj<-runif(Nj)
  T2 <- -log(Vj)/lam2
  event.id<-ifelse(T1<T2, 1, 2)
  event.id[pmin(T1,T2)>censoringtime]<-0
  event.time<-pmin(T1,T2,rep(censoringtime,Nj))
  return(data.frame(t=event.time, k=event.id, t1=T1, t2=T2))
}

set.seed(12345)

nsuj<-1000
lam1<-0.30 # S1(Tcens)=40%
lam2<-0.30
omega<-0.1
```

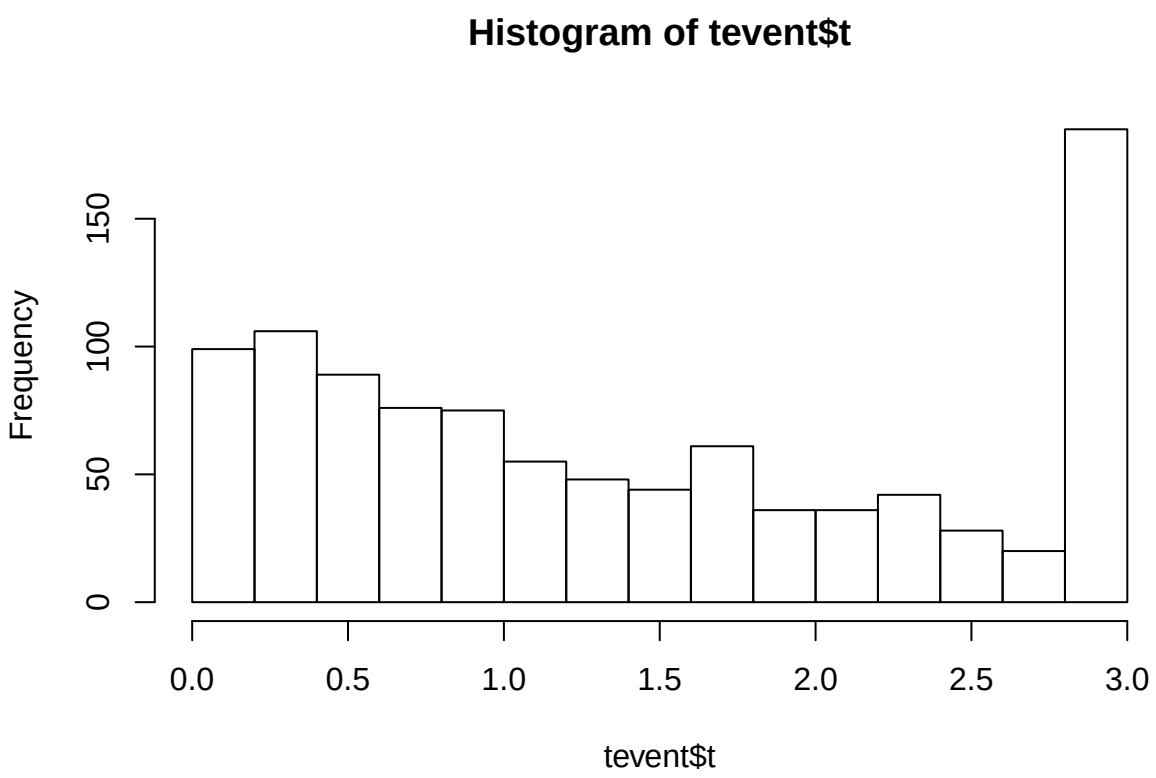
```
xtim<-c(0)
tte.data<-data.frame(id=rep(1:nsuj,each=length(xtim)),time=rep(xtim,nsuj))
psiM<-data.frame(lam1=rep(lam1, nsuj), lam2=lam2*exp(rnorm(nsuj, sd=omega)))
tevent<-constantHazardTTE.simul(psiM, 1:nsuj, tte.data)
summary(tevent$t)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001101 0.485226 1.201180 1.416426 2.311822 3.000000
```

```
table(tevent$k)
```

```
##
##   0   1   2
## 163 433 404
```

```
hist(tevent$t)
```



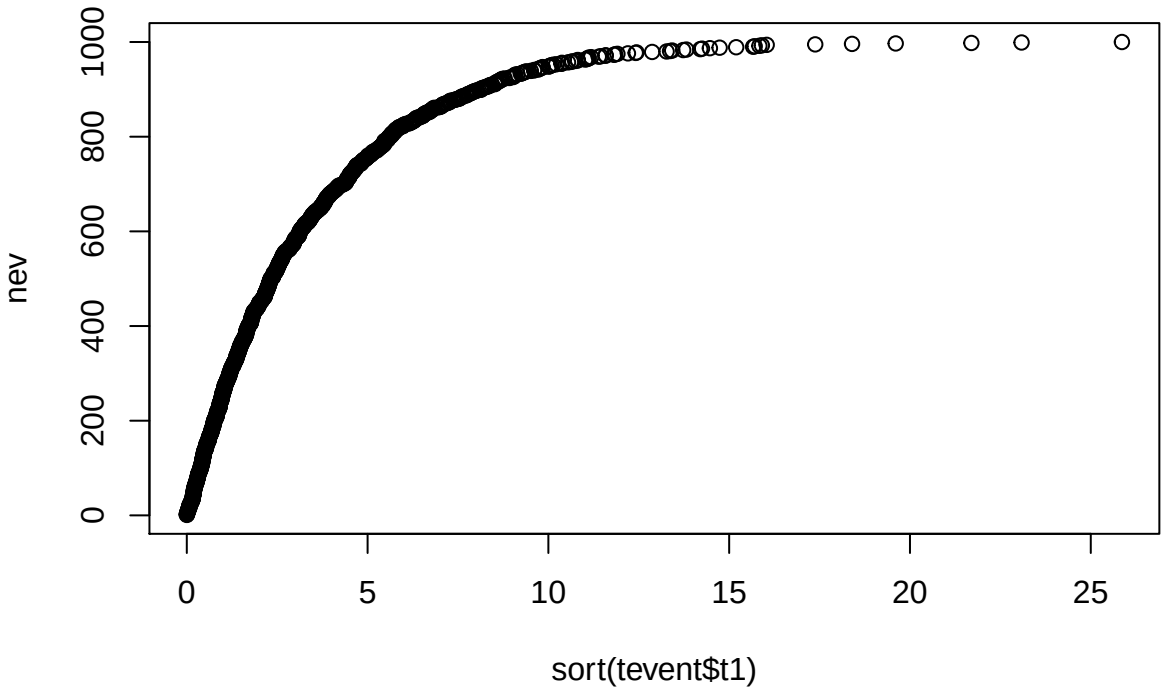

```
length(tevent$t1[tevent$t1<3])
```

```
## [1] 584
```

```
length(tevent$t2[tevent$t2<3])
```

```
## [1] 590
```

```
nev<-cumsum(rep(1,nsuj))
plot(sort(tevent$t1), nev)
```




```
tte.data<-cbind(tte.data, event=0)
tte.data2<-data.frame(id=tte.data$id, time=tevent$t, event=tevent$k)
tte.data<-rbind(tte.data, tte.data2)
tte.data <- tte.data[order(tte.data$id, tte.data$time),]

causeTTE<-tte.data
```

```
## Probability for a competing risk model - cause specific
competingRisks<-function(psi,id,xidep) {
  T<-xidep[,1] # time of the event
  idevent<-xidep[,2]  # events (0=no event (censored), 1=event 1, 2=event 2)
  cens<-which(T==max(T))  # censoring time=3
  #which(xidep[,3]==1) # censoring times (subject specific)
  init <- which(T==0)
  lam1 <- psi[id,1]
  lam2 <- psi[id,2]
  Nj <- length(T)
  ind <- setdiff(1:Nj, append(init,cens)) # indices of events
  h1 <- lam1 # H' event 1
  H1 <- lam1*T # H
  h2 <- lam2 # H' event 2
  H2 <- lam2*T # H

  logpdf <- rep(0, Nj)
#  logpdf[cens] <- log(exp(-H1[cens]) + exp(-H2[cens]) -1) # expression for when T0=0
#  logpdf[cens] <- log(exp(-H1[cens]) + exp(-H2[cens]) -1) -  log(exp(-H1[cens-1]) + exp(-H2[cens-1]) -
  logpdf[cens] <- -H1[cens]-H2[cens] + H1[cens-1] + H2[cens-1] # general expression not assuming T0=0
  logpdf[idevent==1] <-  -H1[idevent==1] + H1[which(idevent==1)-1] + log(h1[idevent==1]) -H2[idevent==1
  logpdf[idevent==2] <-  -H2[idevent==2] + H2[which(idevent==2)-1] + log(h2[idevent==2]) -H1[idevent==2
  return(logpdf)
}
```

```r
### Computing logpdf given psiM and tte.data (test function)
lpdf<-competingRisks(psiM, id=rep(1:nsuj, each=2), tte.data[,2:3])

### Trying saemix
saemix.data<-saemixData(name.data=causeTTE, name.group=c("id"), name.predictors=c("time","event"), name
```

```
##
##
## The following SaemixData object was successfully created:
##
## Object of class SaemixData
##     longitudinal data for use with the SAEM algorithm
## Dataset causeTTE
##     Structured data: event ~ time + event | id
##     X variable for graphs: time ()
```

```r
saemix.model<-saemixModel(model=competingRisks,description="Cause-specific competing risks model",model
                          psi0=matrix(c(0.2, 0.2),ncol=2,byrow=TRUE,dimnames=list(NULL,  c("lam1","lam2
                          transform.par=c(1,1),covariance.model=matrix(c(1,0,0,0),ncol=2, byrow=TRUE))
```

```
##
##
## The following SaemixModel object was successfully created:
##
## Nonlinear mixed-effects model
##   Model function:  Cause-specific competing risks model  Model type:  likelihood
## function(psi,id,xidep) {
##   T<-xidep[,1] # time of the event
##   idevent<-xidep[,2]  # events (0=no event (censored), 1=event 1, 2=event 2)
##   cens<-which(T==max(T))  # censoring time=3
##   #which(xidep[,3]==1) # censoring times (subject specific)
##   init <- which(T==0)
##   lam1 <- psi[id,1]
##   lam2 <- psi[id,2]
##   Nj <- length(T)
##   ind <- setdiff(1:Nj, append(init,cens)) # indices of events
##   h1 <- lam1 # H' event 1
##   H1 <- lam1*T # H
##   h2 <- lam2 # H' event 2
##   H2 <- lam2*T # H
##
##   logpdf <- rep(0, Nj)
## #  logpdf[cens] <- log(exp(-H1[cens]) + exp(-H2[cens]) -1) # expression for when T0=0
## #  logpdf[cens] <- log(exp(-H1[cens]) + exp(-H2[cens]) -1) -  log(exp(-H1[cens-1]) + exp(-H2[cens-1]]
##   logpdf[cens] <- -H1[cens]-H2[cens] + H1[cens-1] + H2[cens-1] # general expression not assuming T0=0
##   logpdf[idevent==1] <-  -H1[idevent==1] + H1[which(idevent==1)-1] + log(h1[idevent==1]) -H2[idevent=
##   logpdf[idevent==2] <-  -H2[idevent==2] + H2[which(idevent==2)-1] + log(h2[idevent==2]) -H1[idevent=
##   return(logpdf)
## }
## <bytecode: 0x564634eb5618>
##   Nb of parameters: 2
##       parameter names:  lam1 lam2
##       distribution:
##      Parameter Distribution Estimated
```

```
## [1,] lam1      log-normal   Estimated
## [2,] lam2      log-normal   Estimated
##   Variance-covariance matrix:
##      lam1 lam2
## lam1    1    0
## lam2    0    0
##     No covariate in the model.
##     Initial values
##              lam1 lam2
## Pop.CondInit  0.2  0.2
```

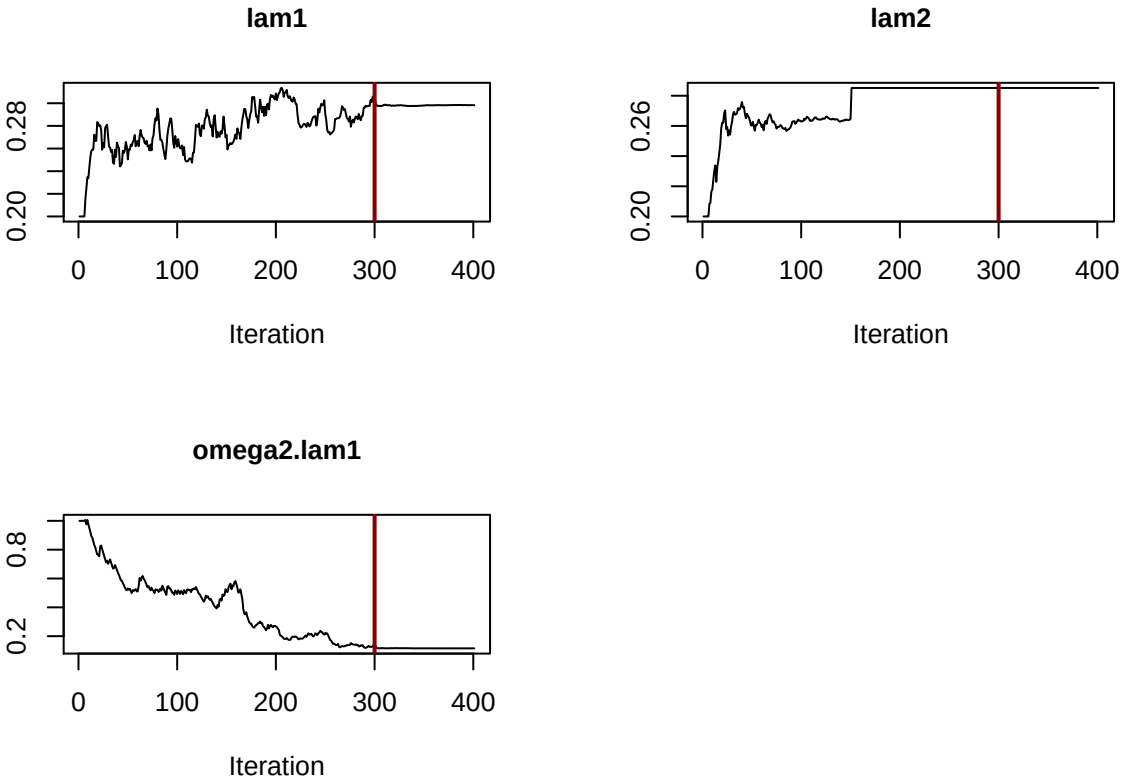
```
saemix.options<-list(seed=632545,save=FALSE,save.graphs=FALSE, fim=FALSE, displayProgress=FALSE)
tte.fit<-saemix(saemix.model,saemix.data,saemix.options)
```

```
## ind.fix10= 2 ind.fix11= 1 ind.fix1= 1 2 ind.fix0=
## -1.350101 -1.329498
## Nonlinear mixed-effects model fit by the SAEM algorithm
## -----------------------------------
## ----          Data             ----
## -----------------------------------
## Object of class SaemixData
##     longitudinal data for use with the SAEM algorithm
## Dataset causeTTE
##     Structured data: event ~ time + event | id
##     X variable for graphs: time ()
## Dataset characteristics:
##     number of subjects:     1000
##     number of observations: 2000
##     average/min/max nb obs: 2.00  /  2  /  2
## First 10 lines of data:
##      id      time event event.1 mdv cens occ ytype
## 1     1 0.0000000     0       0   0    0   1     1
## 1001  1 0.1594930     1       1   0    0   1     1
## 2     2 0.0000000     0       0   0    0   1     1
## 1002  2 0.4576440     2       2   0    0   1     1
## 3     3 0.0000000     0       0   0    0   1     1
## 1003  3 2.1057604     1       1   0    0   1     1
## 4     4 0.0000000     0       0   0    0   1     1
## 1004  4 0.3806178     1       1   0    0   1     1
## 5     5 0.0000000     0       0   0    0   1     1
## 1005  5 1.1457028     2       2   0    0   1     1
## -----------------------------------
## ----          Model            ----
## -----------------------------------
## Nonlinear mixed-effects model
##   Model function:  Cause-specific competing risks model  Model type:  likelihood
## function(psi,id,xidep) {
##   T<-xidep[,1] # time of the event
##   idevent<-xidep[,2]  # events (0=no event (censored), 1=event 1, 2=event 2)
##   cens<-which(T==max(T))  # censoring time=3
##   #which(xidep[,3]==1) # censoring times (subject specific)
##   init <- which(T==0)
##   lam1 <- psi[id,1]
##   lam2 <- psi[id,2]
##   Nj <- length(T)
```

```
##   ind <- setdiff(1:Nj, append(init,cens)) # indices of events
##   h1 <- lam1 # H' event 1
##   H1 <- lam1*T # H
##   h2 <- lam2 # H' event 2
##   H2 <- lam2*T # H
##
##   logpdf <- rep(0, Nj)
## #  logpdf[cens] <- log(exp(-H1[cens]) + exp(-H2[cens]) -1) # expression for when T0=0
## #  logpdf[cens] <- log(exp(-H1[cens]) + exp(-H2[cens]) -1) -  log(exp(-H1[cens-1]) + exp(-H2[cens-1]]
##   logpdf[cens] <- -H1[cens]-H2[cens] + H1[cens-1] + H2[cens-1] # general expression not assuming T0=0
##   logpdf[idevent==1] <-  -H1[idevent==1] + H1[which(idevent==1)-1] + log(h1[idevent==1]) -H2[idevent=
##   logpdf[idevent==2] <-  -H2[idevent==2] + H2[which(idevent==2)-1] + log(h2[idevent==2]) -H1[idevent=
##   return(logpdf)
## }
## <bytecode: 0x564634eb5618>
##   Nb of parameters: 2
##       parameter names:  lam1 lam2
##       distribution:
##      Parameter Distribution Estimated
## [1,] lam1      log-normal   Estimated
## [2,] lam2      log-normal   Estimated
##   Variance-covariance matrix:
##      lam1 lam2
## lam1    1    0
## lam2    0    0
##     No covariate in the model.
##     Initial values
##              lam1 lam2
## Pop.CondInit  0.2  0.2
## ----------------------------------
## ----    Key algorithm options  ----
## ----------------------------------
##     Estimation of individual parameters (MAP)
##     Estimation of log-likelihood by importance sampling
##     Number of iterations:  K1=300, K2=100
##     Number of chains:  1
##     Seed:  632545
##     Number of MCMC iterations for IS:  5000
##     Simulations:
##         nb of simulated datasets used for npde:  1000
##         nb of simulated datasets used for VPC:  100
##     Input/output
##         save the results to a file:  FALSE
##         save the graphs to files:  FALSE
## ----------------------------------------------------
## ----                  Results                   ----
## ----------------------------------------------------
## -----------------  Fixed effects  ------------------
## ----------------------------------------------------
##      Parameter Estimate
## [1,] lam1      0.30
## [2,] lam2      0.29
## ----------------------------------------------------
## -----------  Variance of random effects  -----------
```

```
## ----------------------------------------------------
##      Parameter   Estimate
## lam1 omega2.lam1 0.12
## ----------------------------------------------------
## ------  Correlation matrix of random effects  ------
## ----------------------------------------------------
##             omega2.lam1
## omega2.lam1 1
## ----------------------------------------------------
## ---------------  Statistical criteria  -------------
## ----------------------------------------------------
##
## Likelihood computed by importance sampling
##       -2LL= 3715.073
##       AIC = 3723.073
##       BIC = 3742.704
## ----------------------------------------------------
```

```
plot(tte.fit, plot.type="convergence")
```



Competing risks model - subdistribution assumption

- **TODO** change simulation if needed to reflect the subdistribution assumption (not sure needed...)

```
# Simulate with parameters so that S>0
nsuj<-1000
lam1<-0.17 # S1(Tcens)=60% at T=3
lam2<-0.17
omega<-0.1
```

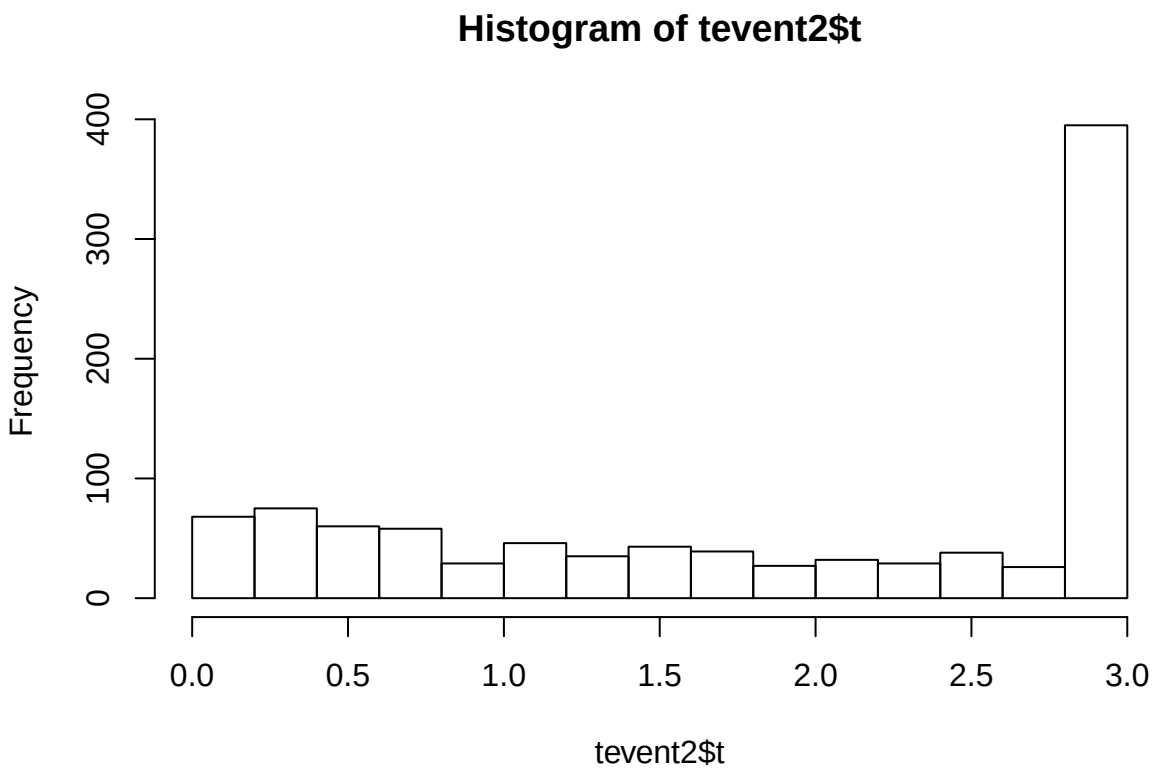
```
xtim<-c(0)
ttesub.data<-data.frame(id=rep(1:nsuj,each=length(xtim)),time=rep(xtim,nsuj))
psiM2<-data.frame(lam1=rep(lam1, nsuj), lam2=lam2*exp(rnorm(nsuj, sd=omega)))
tevent2<-constantHazardTTE.simul(psiM2, 1:nsuj, ttesub.data)
summary(tevent2$t)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.003508 0.765637 2.118217 1.886640 3.000000 3.000000
```

```
table(tevent2$k)
```

```
##
##   0   1   2
## 368 321 311
```

```
hist(tevent2$t)
```



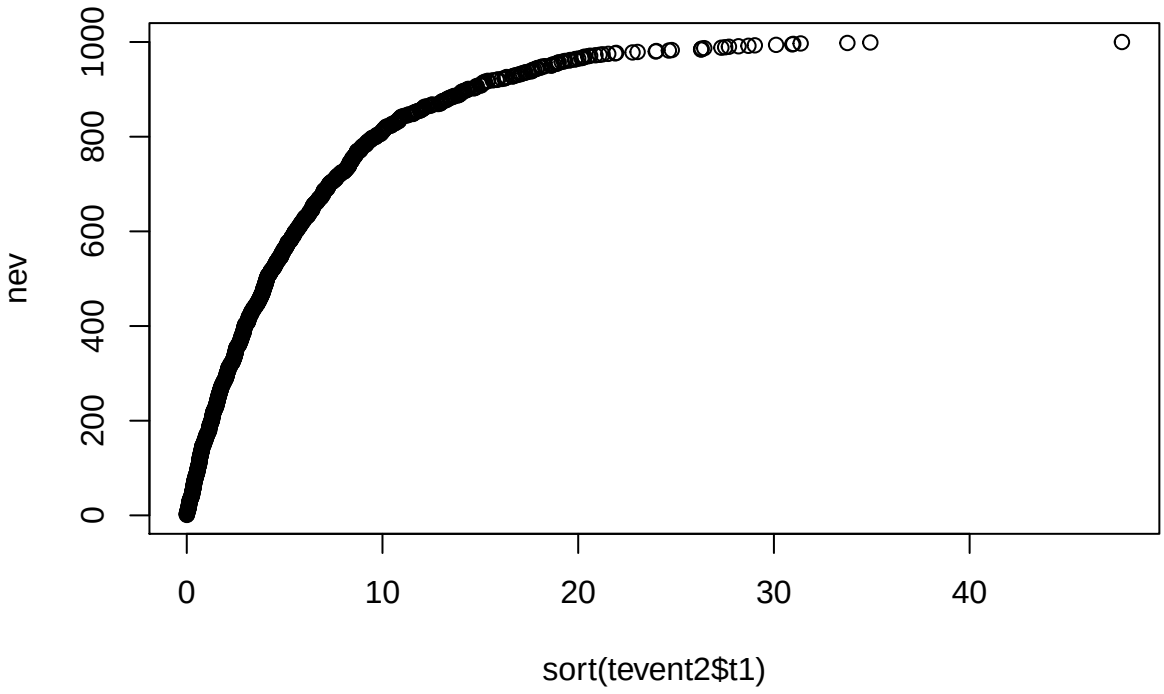

```
length(tevent2$t1[tevent2$t1<3])
```

```
## [1] 404
```

```
length(tevent2$t2[tevent2$t2<3])
```

```
## [1] 386
```

```
nev<-cumsum(rep(1,nsuj))
plot(sort(tevent2$t1), nev)
```




```
ttesub.data<-cbind(ttesub.data, event=0)
ttesub.data2<-data.frame(id=ttesub.data$id, time=tevent2$t, event=tevent2$k)
ttesub.data<-rbind(ttesub.data, ttesub.data2)
ttesub.data <- ttesub.data[order(ttesub.data$id, ttesub.data$time),]

subdistTTE <- ttesub.data
```

```
## Probability for a competing risk model - subdistribution
competingRisks.subdist<-function(psi,id,xidep) {
    T<-xidep[,1] # time of the event
  idevent<-xidep[,2]  # events (0=no event (censored), 1=event 1, 2=event 2)
  cens<-which(T==max(T))  # censoring time=3
  #which(xidep[,3]==1) # censoring times (subject specific)
  init <- which(T==0)
  lam1 <- psi[id,1]
  lam2 <- psi[id,2]
  Nj <- length(T)
  ind <- setdiff(1:Nj, append(init,cens)) # indices of events
  h1 <- lam1 # H' event 1
  H1 <- lam1*T # H
  h2 <- lam2 # H' event 2
  H2 <- lam2*T # H

  logpdf <- exp(-H1)+exp(-H2) -1 # 1-F1-F2 = S1+S2-1

#  logpdf[cens] <- oversuru[cens] # expression for when T0=0
#  logpdf[cens] <- oversuru[cens]-oversuru[cens-1] # expression for when T0 <> 0
  logpdf[idevent==1] <- logpdf[idevent==1] -H1[idevent==1]+ log(h1[idevent==1]) # assumes no RTTE (woul
  logpdf[idevent==2] <- logpdf[idevent==2] -H2[idevent==2] + log(h2[idevent==2])
  return(logpdf)
}
```

```r
lpdf2<-competingRisks.subdist(psiM2, id=rep(1:nsuj, each=2), ttesub.data[,2:3])
summary(lpdf2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -2.1130 -0.9424  0.6449  0.1133  1.0000  1.0000
```

```r
### Trying saemix
saemix.data2<-saemixData(name.data=subdistTTE, name.group=c("id"), name.predictors=c("time","event"), na
```

```
##
##
## The following SaemixData object was successfully created:
##
## Object of class SaemixData
##     longitudinal data for use with the SAEM algorithm
## Dataset subdistTTE
##     Structured data: event ~ time + event | id
##     X variable for graphs: time ()
```

```r
saemix.model2<-saemixModel(model=competingRisks.subdist,description="Subdistribution competing risks mo
                           psi0=matrix(c(0.3, 0.3),ncol=2,byrow=TRUE,dimnames=list(NULL,  c("lam1","lam2
                           transform.par=c(1,1),covariance.model=matrix(c(1,0,0,0),ncol=2, byrow=TRUE))
```

```
##
##
## The following SaemixModel object was successfully created:
##
## Nonlinear mixed-effects model
##   Model function:  Subdistribution competing risks model  Model type:  likelihood
## function(psi,id,xidep) {
##     T<-xidep[,1] # time of the event
##   idevent<-xidep[,2]  # events (0=no event (censored), 1=event 1, 2=event 2)
##   cens<-which(T==max(T))  # censoring time=3
##   #which(xidep[,3]==1) # censoring times (subject specific)
##   init <- which(T==0)
##   lam1 <- psi[id,1]
##   lam2 <- psi[id,2]
##   Nj <- length(T)
##   ind <- setdiff(1:Nj, append(init,cens)) # indices of events
##   h1 <- lam1 # H' event 1
##   H1 <- lam1*T # H
##   h2 <- lam2 # H' event 2
##   H2 <- lam2*T # H
##
##   logpdf <- exp(-H1)+exp(-H2) -1 # 1-F1-F2 = S1+S2-1
##
## #  logpdf[cens] <- oversurv[cens] # expression for when T0=0
## #  logpdf[cens] <- oversurv[cens]-oversurv[cens-1] # expression for when T0 <> 0
##   logpdf[idevent==1] <- logpdf[idevent==1] -H1[idevent==1]+ log(h1[idevent==1]) # assumes no RTTE (w
##   logpdf[idevent==2] <- logpdf[idevent==2] -H2[idevent==2] + log(h2[idevent==2])
##   return(logpdf)
## }
## <bytecode: 0x564639cfa368>
##   Nb of parameters: 2
##       parameter names:  lam1 lam2
```

```
##       distribution:
##      Parameter Distribution Estimated
## [1,] lam1      log-normal   Estimated
## [2,] lam2      log-normal   Estimated
##   Variance-covariance matrix:
##      lam1 lam2
## lam1    1    0
## lam2    0    0
##     No covariate in the model.
##     Initial values
##              lam1 lam2
## Pop.CondInit  0.3  0.3
```

```
saemix.options<-list(seed=632545,save=FALSE,save.graphs=FALSE, fim=FALSE, displayProgress=FALSE)
tte.fit2<-saemix(saemix.model2,saemix.data2,saemix.options)
```

```
## ind.fix10= 2 ind.fix11= 1 ind.fix1= 1 2 ind.fix0=
## -1.718082 -1.644902
## Nonlinear mixed-effects model fit by the SAEM algorithm
## -----------------------------------
## ----          Data             ----
## -----------------------------------
## Object of class SaemixData
##     longitudinal data for use with the SAEM algorithm
## Dataset subdistTTE
##     Structured data: event ~ time + event | id
##     X variable for graphs: time ()
## Dataset characteristics:
##     number of subjects:     1000
##     number of observations: 2000
##     average/min/max nb obs: 2.00  /  2  /  2
## First 10 lines of data:
##      id     time event event.1 mdv cens occ ytype
## 1     1 0.000000     0       0   0    0   1     1
## 1001  1 3.000000     0       0   0    0   1     1
## 2     2 0.000000     0       0   0    0   1     1
## 1002  2 2.129255     2       2   0    0   1     1
## 3     3 0.000000     0       0   0    0   1     1
## 1003  3 3.000000     0       0   0    0   1     1
## 4     4 0.000000     0       0   0    0   1     1
## 1004  4 2.470050     2       2   0    0   1     1
## 5     5 0.000000     0       0   0    0   1     1
## 1005  5 3.000000     0       0   0    0   1     1
## -----------------------------------
## ----          Model            ----
## -----------------------------------
## Nonlinear mixed-effects model
##   Model function:  Subdistribution competing risks model  Model type:  likelihood
## function(psi,id,xidep) {
##     T<-xidep[,1] # time of the event
##   idevent<-xidep[,2]  # events (0=no event (censored), 1=event 1, 2=event 2)
##   cens<-which(T==max(T))  # censoring time=3
##   #which(xidep[,3]==1) # censoring times (subject specific)
##   init <- which(T==0)
##   lam1 <- psi[id,1]
```

```
##   lam2 <- psi[id,2]
##   Nj <- length(T)
##   ind <- setdiff(1:Nj, append(init,cens)) # indices of events
##   h1 <- lam1 # H' event 1
##   H1 <- lam1*T # H
##   h2 <- lam2 # H' event 2
##   H2 <- lam2*T # H
##
##   logpdf <- exp(-H1)+exp(-H2) -1 # 1-F1-F2 = S1+S2-1
##
## #  logpdf[cens] <- oversurv[cens] # expression for when T0=0
## #  logpdf[cens] <- oversurv[cens]-oversurv[cens-1] # expression for when T0 <> 0
##   logpdf[idevent==1] <- logpdf[idevent==1] -H1[idevent==1]+ log(h1[idevent==1]) # assumes no RTTE (w
##   logpdf[idevent==2] <- logpdf[idevent==2] -H2[idevent==2] + log(h2[idevent==2])
##   return(logpdf)
## }
## <bytecode: 0x564639cfa368>
##   Nb of parameters: 2
##       parameter names:  lam1 lam2
##       distribution:
##      Parameter Distribution Estimated
## [1,] lam1      log-normal   Estimated
## [2,] lam2      log-normal   Estimated
##   Variance-covariance matrix:
##      lam1 lam2
## lam1    1    0
## lam2    0    0
##     No covariate in the model.
##     Initial values
##              lam1 lam2
## Pop.CondInit  0.3  0.3
## ----------------------------------
## ----    Key algorithm options  ----
## ----------------------------------
##     Estimation of individual parameters (MAP)
##     Estimation of log-likelihood by importance sampling
##     Number of iterations:  K1=300, K2=100
##     Number of chains:  1
##     Seed:  632545
##     Number of MCMC iterations for IS:  5000
##     Simulations:
##         nb of simulated datasets used for npde:  1000
##         nb of simulated datasets used for VPC:  100
##     Input/output
##         save the results to a file:  FALSE
##         save the graphs to files:  FALSE
## ----------------------------------------------------
## ----                  Results                   ----
## ----------------------------------------------------
## -----------------  Fixed effects  ------------------
## ----------------------------------------------------
##      Parameter Estimate
## [1,] lam1      0.19
## [2,] lam2      0.20
```

```
## ------------------------------------------------------
## -----------  Variance of random effects  -----------
## ------------------------------------------------------
##      Parameter   Estimate
## lam1 omega2.lam1 0.11
## ------------------------------------------------------
## ------  Correlation matrix of random effects  ------
## ------------------------------------------------------
##             omega2.lam1
## omega2.lam1 1
## ------------------------------------------------------
## ---------------  Statistical criteria  -------------
## ------------------------------------------------------
##
## Likelihood computed by importance sampling
##       -2LL= -461.5301
##       AIC = -453.5301
##       BIC = -433.8991
## ------------------------------------------------------
```

```
plot(tte.fit2, plot.type="convergence")
```