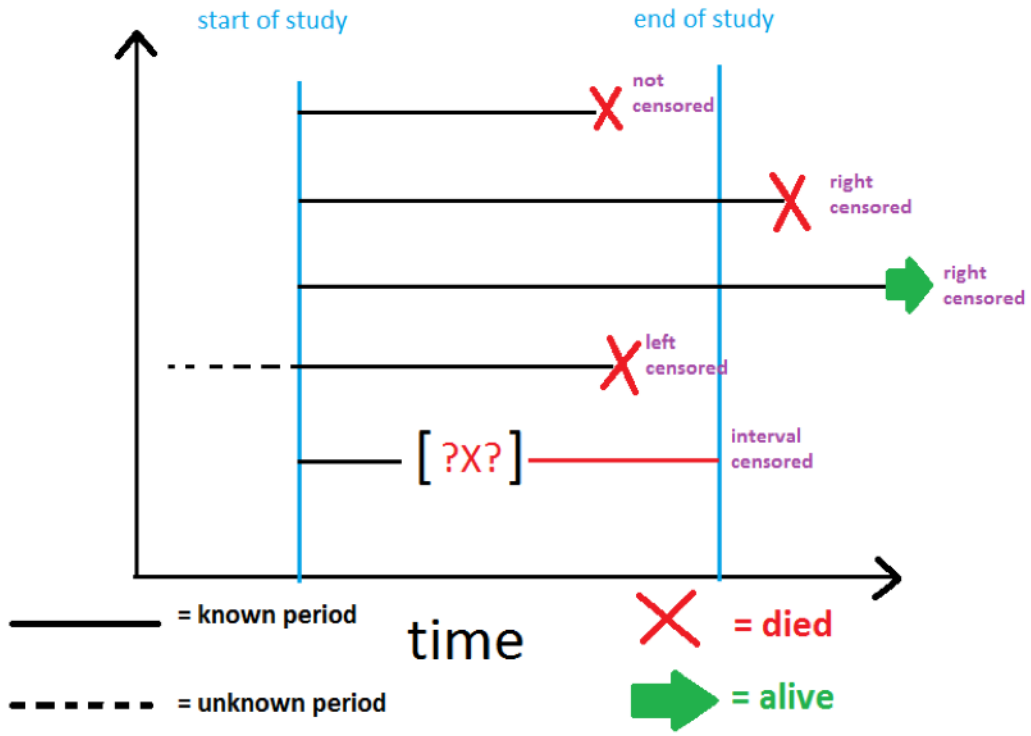
Survival analysis using R-INLA

UNISA, South Africa, 2019

Janet van Niekerk

What is survival data?



Parametric survival models

First we will consider parametric survival models. The parametric models available in *INLA* are *exponentialsurv, weibullsurv, loglogisticsurv, lognormalsurv, gamma.surv.*

Survival function models

1. Exponential :
$$t \sim Exp(\lambda), \lambda = \exp(\eta)$$
2. Weibull:
$$t \sim Weibull(\alpha, \lambda), \lambda = \exp(\eta)$$
3. Loglogistic:
$$\log(t) \sim Logistic(\alpha, \lambda), \lambda = \exp(\eta)$$
4. Lognormal:
$$\log(t) \sim N(\eta, \tau)$$

In all of these,

$$\eta = \sum_{j=1}^{n_\beta} \beta_j' \boldsymbol{X}_j + \sum_{k=1}^{n_f} f^k(\boldsymbol{u}_k).$$

Random components $f^k(\boldsymbol{u}_k)$ of covariates $\boldsymbol{u}_k$ can be splines, spatial components, frailties, clustering effects etc.

Examples

1. Ovarian Cancer Survival Data

Survival in a randomised trial to compare two treatments for ovarian cancer.

Covariates:
futime: survival or censoring time
fustat: censoring status
age: in years
resid.ds: residual disease present (1=no,2=yes)
rx: treatment group
ecog.ps: ECOG performance status (1 is better, see Edmunson (1979))

```
library(INLA)
```

```
## Loading required package: Matrix
```

```
## Loading required package: sp
```

```
## Loading required package: parallel
```

```
## This is INLA_19.05.19 built 2019-05-19 15:31:29 UTC.
## See www.r-inla.org/contact-us for how to get help.
## To enable PARDISO sparse library; see inla.pardiso()
```

```
library(survival)

ovarian_data<-ovarian
ovarian_data
```

```
##    futime fustat     age resid.ds rx ecog.ps
## 1      59      1 72.3315        2  1       1
## 2     115      1 74.4932        2  1       1
## 3     156      1 66.4658        2  1       2
## 4     421      0 53.3644        2  2       1
## 5     431      1 50.3397        2  1       1
## 6     448      0 56.4301        1  1       2
## 7     464      1 56.9370        2  2       2
## 8     475      1 59.8548        2  2       2
## 9     477      0 64.1753        2  1       1
## 10    563      1 55.1781        1  2       2
## 11    638      1 56.7562        1  1       2
## 12    744      0 50.1096        1  2       1
## 13    769      0 59.6301        2  2       2
## 14    770      0 57.0521        2  2       1
## 15    803      0 39.2712        1  1       1
## 16    855      0 43.1233        1  1       2
## 17   1040      0 38.8932        2  1       2
```

```
## 18   1106      0 44.6000        1  1       1
## 19   1129      0 53.9068        1  2       1
## 20   1206      0 44.2055        2  2       1
## 21   1227      0 59.5890        1  2       2
## 22    268      1 74.5041        2  1       2
## 23    329      1 43.1370        2  1       1
## 24    353      1 63.2192        1  2       2
## 25    365      1 64.4247        2  2       1
## 26    377      0 58.3096        1  2       1
```

Now we can examine the survival curve for each treatment. This can give us an indication of possible differences between treatments. The Kaplan-Meier curve is a nonparametric estimate of the survival curve and can be used as a visual summary of the survival times.
For numerical stability it is advisable to rescale the time axis to the unit axis.

```
#Scale the time axis
mtime<-max(ovarian_data$futime)
mtime
```

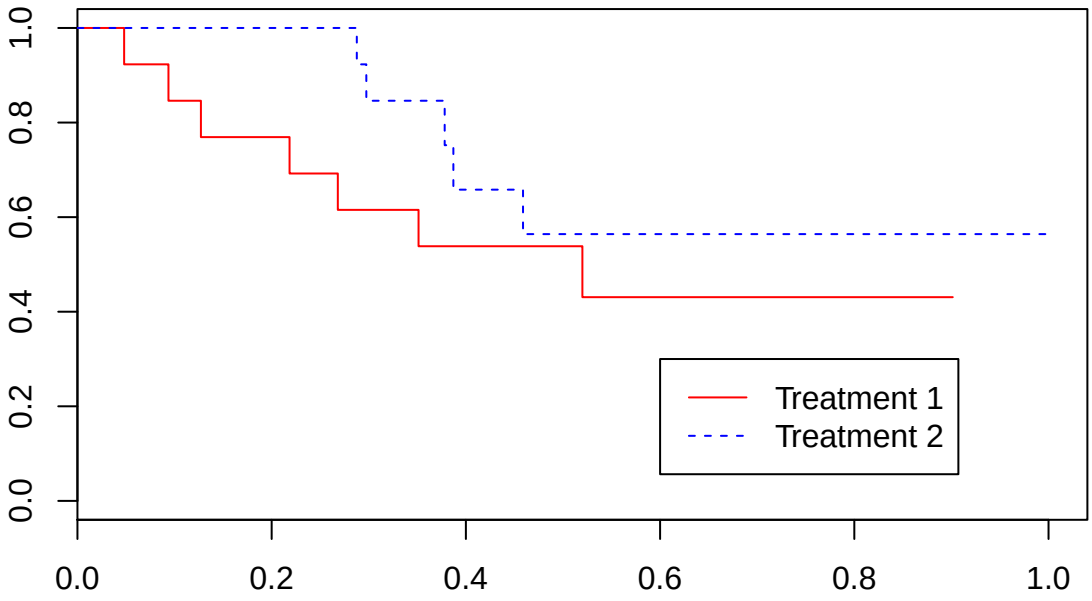
```
## [1] 1227
```

```
ovarian_data$futime<-ovarian_data$futime/mtime

#Kaplan-meier curves for each treatment
ovarian_surv<-Surv(ovarian_data$futime,ovarian_data$fustat)
ovarian_surv
```

```
##  [1] 0.04808476  0.09372453  0.12713936  0.34311328+ 0.35126324
##  [6] 0.36511817+ 0.37815811  0.38712306  0.38875306+ 0.45884271
## [11] 0.51996740  0.60635697+ 0.62673187+ 0.62754686+ 0.65444173+
## [16] 0.69682152+ 0.84759576+ 0.90138549+ 0.92013040+ 0.98288509+
## [21] 1.00000000+ 0.21841891  0.26813366  0.28769356  0.29747351
## [26] 0.30725346+
```

```
ovarian_km<-survfit(ovarian_surv~rx,data=ovarian_data)
plot(ovarian_km,col=c("red","blue"),lty=c(1,2))
legend(x=0.6,y=0.3,legend=c("Treatment 1", "Treatment 2"), col=c("red","blue"),lty=c(1,2))
```

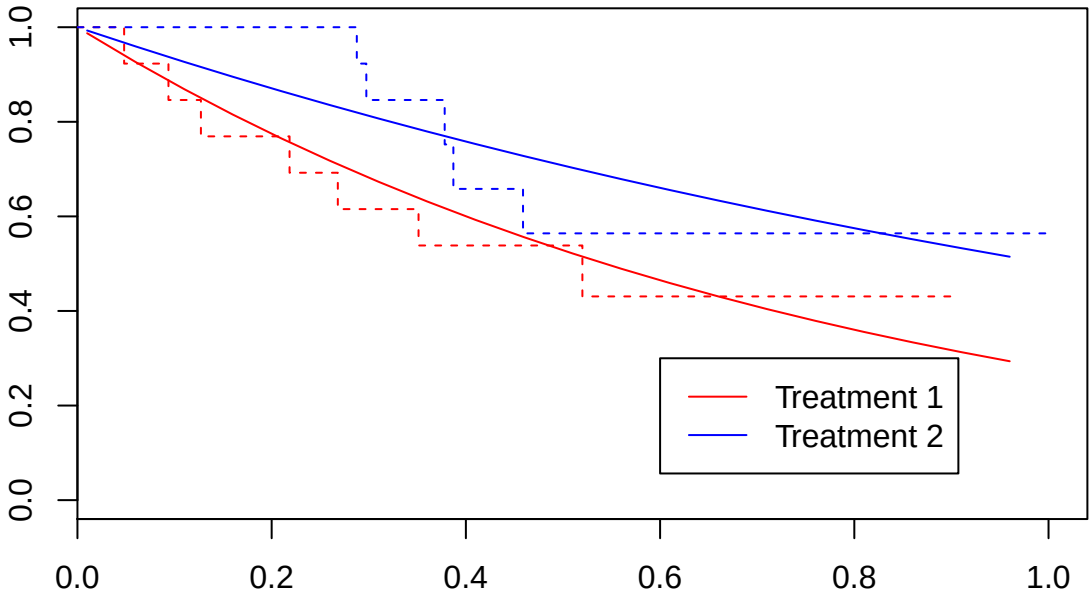


Note that in this curve no effect of other covariates are taken into account so the apparent difference can be due to some other factors as well. We thus need to do a proper model and test for significant covariate effects. We can do this using `inla`.
For this we need to create a survival object in INLA, using `inla.surv`.

```
ovarian_surv<-inla.surv(time=ovarian_data$futime,event=ovarian_data$fustat)
#View(ovarian_surv)
model_ovarian<-inla(formula=ovarian_surv~-1+as.factor(rx),family="exponentialsurv",
                    data=ovarian_data,verbose=TRUE)
summary(model_ovarian)
```

```
##
## Call:
##    c("inla(formula = ovarian_surv ~ -1 + as.factor(rx), family =
##    \"exponentialsurv\", ", " data = ovarian_data, verbose = TRUE)")
## Time used:
##     Pre = 1.59, Running = 0.133, Post = 0.108, Total = 1.83
## Fixed effects:
##                   mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## as.factor(rx)1  0.244 0.377     -0.562    0.267      0.922  0.315   0
## as.factor(rx)2 -0.369 0.446     -1.329   -0.338      0.424 -0.274   0
##
## Expected number of effective parameters(stdev): 2.00(0.00)
## Number of equivalent replicates : 13.00
##
## Marginal log-Likelihood:  -20.81
```

```
#Plot results
plot(ovarian_km,col=c("red","blue"),lty=c(2,2))
lines(seq(0.01,1,0.05),exp(-exp(model_ovarian$summary.fixed$mean[1])*seq(0.01,1,0.05)),
      col="red")
lines(seq(0.01,1,0.05),exp(-exp(model_ovarian$summary.fixed$mean[2])*seq(0.01,1,0.05)),
      col="blue")
legend(x=0.6,y=0.3,legend=c("Treatment 1", "Treatment 2"), col=c("red","blue"),lty=c(1,1))
```

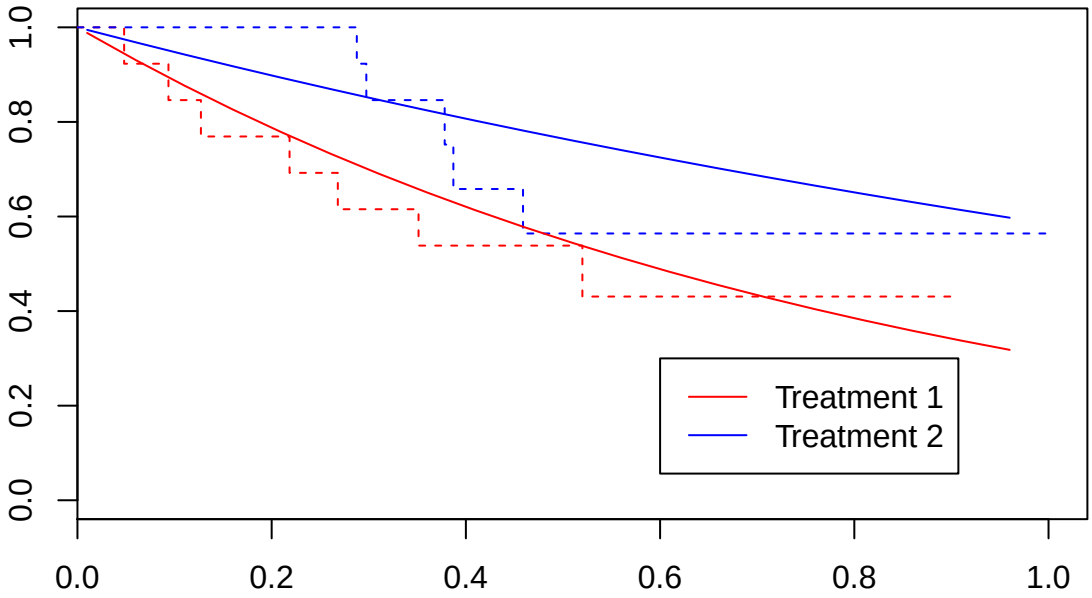


Now let's include the other covariates.

```r
model_ovarian1<-inla(formula=ovarian_surv~-1+as.factor(rx)+age+as.factor(ecog.ps),
                     family="exponentialsurv",data=ovarian_data)
summary(model_ovarian1)
```

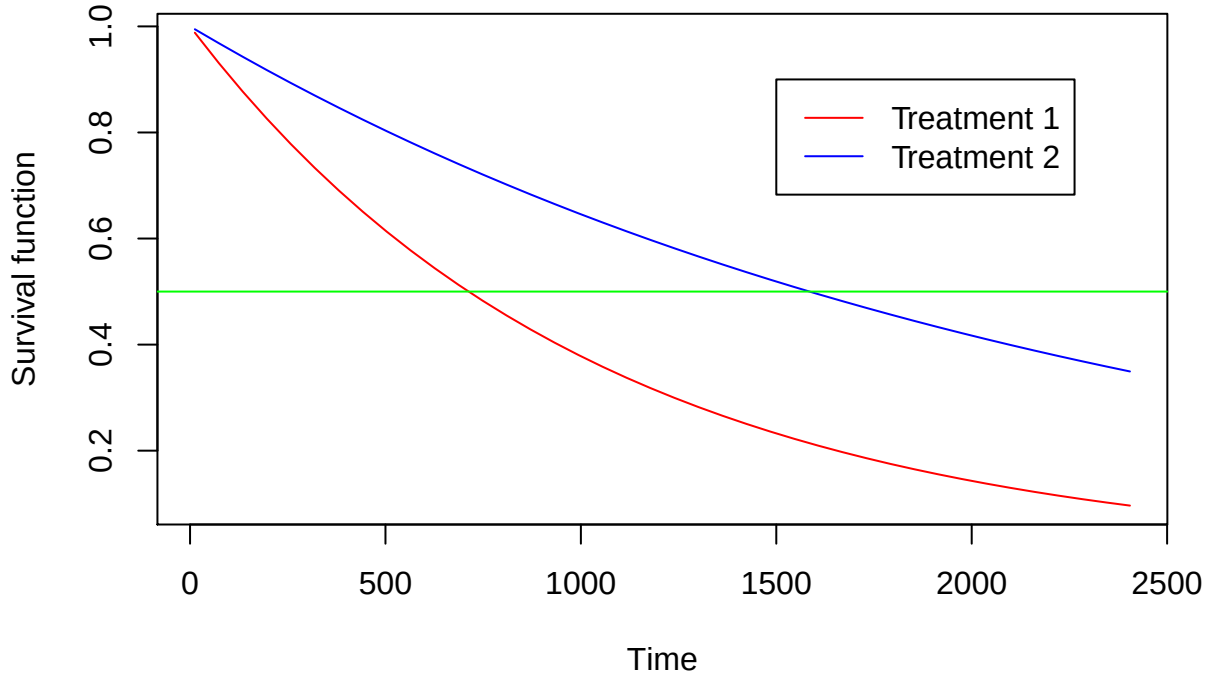
```
##
## Call:
##    c("inla(formula = ovarian_surv ~ -1 + as.factor(rx) + age +
##    as.factor(ecog.ps), ", " family = \"exponentialsurv\", data =
##    ovarian_data)")
## Time used:
##     Pre = 1.36, Running = 0.133, Post = 0.0952, Total = 1.59
## Fixed effects:
##                        mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## as.factor(rx)1       -5.645 2.051    -10.050   -5.508     -1.993 -5.225   0
## as.factor(rx)2       -6.445 1.888    -10.410   -6.354     -2.987 -6.172   0
## age                   0.104 0.032      0.045    0.102      0.170  0.099   0
## as.factor(ecog.ps)2   0.089 0.591     -1.039    0.078      1.280  0.056   0
##
## Expected number of effective parameters(stdev): 3.99(0.00)
## Number of equivalent replicates : 6.51
##
## Marginal log-Likelihood:  -25.46
```

```r
#Plot results
plot(ovarian_km,col=c("red","blue"),lty=c(2,2))
lines(seq(0.01,1,0.05),exp(-exp(model_ovarian1$summary.fixed$mean[1]+
            model_ovarian1$summary.fixed$mean[3]*mean(ovarian_data$age))*
            seq(0.01,1,0.05)),col="red")
lines(seq(0.01,1,0.05),exp(-exp(model_ovarian1$summary.fixed$mean[2]+
            model_ovarian1$summary.fixed$mean[3]*mean(ovarian_data$age))*
            seq(0.01,1,0.05)),col="blue")
legend(x=0.6,y=0.3,legend=c("Treatment 1", "Treatment 2"), col=c("red","blue"),lty=c(1,1))
```



So we can conclude that the treatment has a statistically significant effect. For interpretation we should remember to scale the time axis to the original scale to find for instance the median survival time.

```
plot(seq(0.01,2,0.05)*mtime,exp(-exp(model_ovarian1$summary.fixed$mean[1]+
              model_ovarian1$summary.fixed$mean[3]*mean(ovarian_data$age))*
                seq(0.01,2,0.05)),col="red",type="l",xlab="Time",ylab="Survival function")
lines(seq(0.01,2,0.05)*mtime,exp(-exp(model_ovarian1$summary.fixed$mean[2]+
              model_ovarian1$summary.fixed$mean[3]*mean(ovarian_data$age))*
                seq(0.01,2,0.05)),col="blue")
abline(a=0.5,b=0,col="green")
legend(x=1500,y=0.9,legend=c("Treatment 1", "Treatment 2"), col=c("red","blue"),lty=c(1,1))
```



Semi-parametric model - Cox proportional hazards model

The Cox proportional hazards model is a regression model based on the assumption of proportional hazards accross subjects. The covariates are included in the model through a parametric linear predictor but the baseline hazard function (when all covariate values are 0) is estimated nonparametrically.
The Cox model is implemented in `INLA` through the equivalence of the Cox model to a series of Poisson regression models [**?**].
Examples

1. Ovarian Cancer Survival Data (revisited)

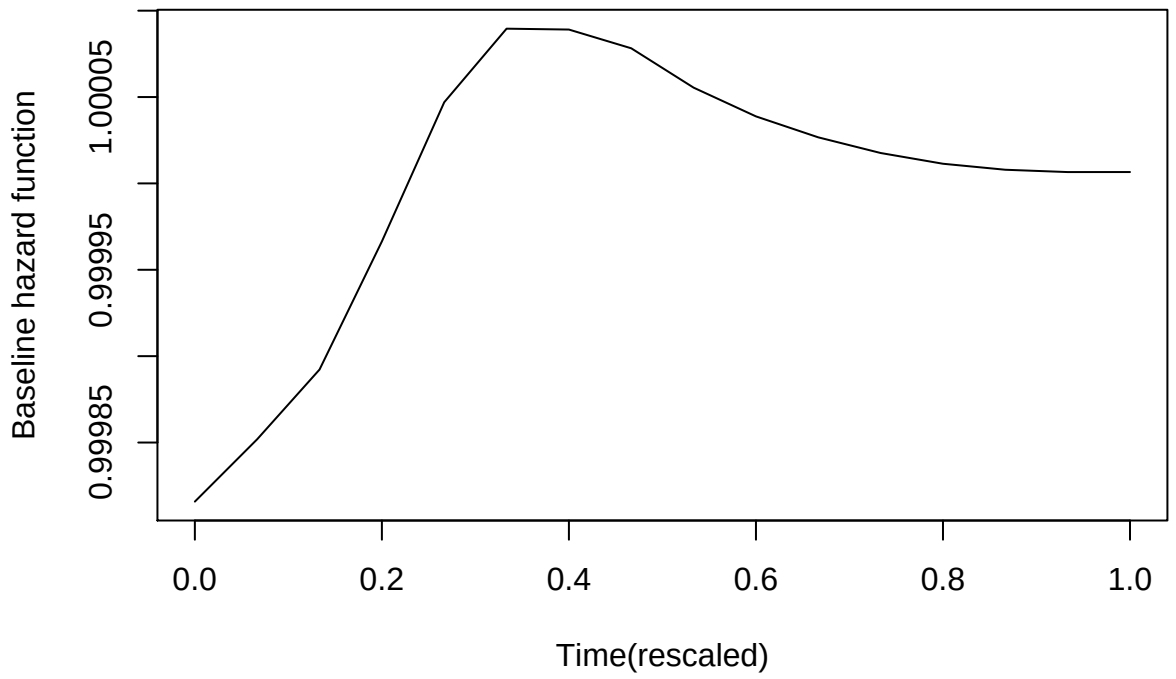
We revisit the ovarian cancer example and do the analysis using a Cox proportional hazards model.

```
model_ovarian2<-inla(formula=ovarian_surv~-1+as.factor(rx)+age+as.factor(ecog.ps),
                     family="coxph",data=ovarian_data)
summary(model_ovarian2)
```

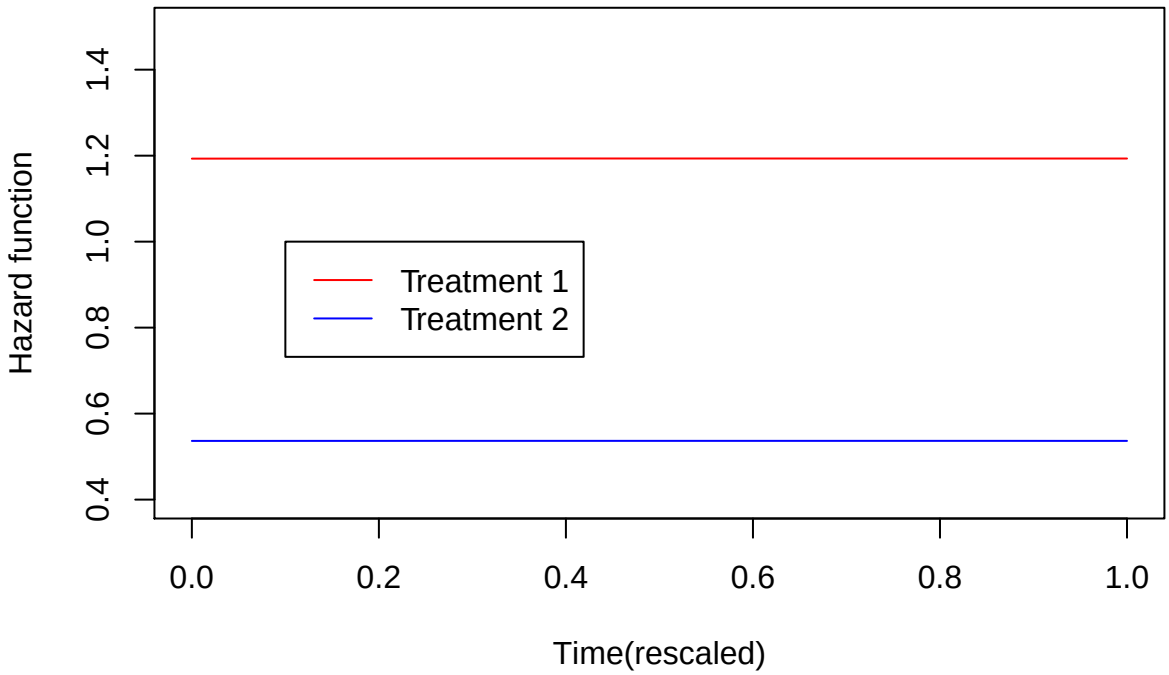
```
##
## Call:
##    c("inla(formula = cph$formula, family = cph$family, contrasts =
##    contrasts, ", " data = c(as.list(cph$data), cph$data.list),
```

```
##    quantiles = quantiles, ", " E = cph$E, offset = offset, scale =
##    scale, weights = weights, ", " Ntrials = NULL, strata = NULL,
##    verbose = verbose, lincomb = lincomb, ", " control.compute =
##    control.compute, control.predictor = control.predictor, ", "
##    control.family = control.family, control.inla = control.inla, ", "
##    control.results = control.results, control.fixed = control.fixed,
##    ", " control.mode = control.mode, control.expert = control.expert,
##    ", " control.hazard = control.hazard, control.lincomb =
##    control.lincomb, ", " control.update = control.update,
##    only.hyperparam = only.hyperparam, ", " inla.call = inla.call,
##    inla.arg = inla.arg, num.threads = num.threads, ", "
##    blas.num.threads = blas.num.threads, keep = keep,
##    working.directory = working.directory, ", " silent = silent, debug
##    = debug)")
## Time used:
##     Pre = 1.5, Running = 0.202, Post = 0.109, Total = 1.81
## Fixed effects:
##                        mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## as.factor(rx)1      -5.646 2.051    -10.050   -5.508     -1.993 -5.225   0
## as.factor(rx)2      -6.445 1.888    -10.410   -6.354     -2.987 -6.173   0
## age                  0.104 0.032      0.045    0.102      0.170  0.099   0
## as.factor(ecog.ps)2  0.089 0.591     -1.039    0.078      1.280  0.056   0
##
## Random effects:
##   Name     Model
##     baseline.hazard RW1 model
##
## Model hyperparameters:
##                                    mean       sd 0.025quant 0.5quant
## Precision for baseline.hazard 20021.99 19829.85     513.51 13877.25
##                               0.975quant  mode
## Precision for baseline.hazard   73345.92 13.09
##
## Expected number of effective parameters(stdev): 3.99(0.003)
## Number of equivalent replicates : 50.59
##
## Marginal log-Likelihood:  -90.36
```

```r
#Plot baseline hazard
plot(model_ovarian2$summary.random$baseline.hazard$ID,
     exp(model_ovarian2$summary.random$baseline.hazard$mean),
     type="l",xlab="Time(rescaled)",ylab="Baseline hazard function")
```




```
#Plot results
plot(model_ovarian2$summary.random$baseline.hazard$ID,
     exp(model_ovarian2$summary.random$baseline.hazard$mean+
     rep(1,16)*(model_ovarian2$summary.fixed$mean[1]+
     model_ovarian2$summary.fixed$mean[3]*mean(ovarian_data$age))),
     col="red",type="l",xlab="Time(rescaled)",ylab="Hazard function",
     ylim=c(0.4,1.5))
lines(model_ovarian2$summary.random$baseline.hazard$ID,
      exp(model_ovarian2$summary.random$baseline.hazard$mean+
      rep(1,16)*(model_ovarian2$summary.fixed$mean[2]+
      model_ovarian2$summary.fixed$mean[3]*mean(ovarian_data$age))),
      col="blue")
legend(x=0.1,y=1,legend=c("Treatment 1", "Treatment 2"), col=c("red","blue"),lty=c(1,1))
```



The (near) constant hazard functions imply that the exponential assumption (as in our previous example) was correct. The proportionality assumption should always be evaluated.

Spatial survival models

A spatial model is just a "normal" model with a latent random effect over space. In the next example we illustrate how straightforward a spatial random effect can be incuded in the `inla` call. (More detail about spatial models on Thursday) ## Examples

2. Leukemkia survival data

A dataset on the survival of acute myeloid leukemia in 1,043 pateietns, first analyzed by Henderson et al. (2002). It is of interest to investigate possible spatial variation in survival after accounting for known subject-specific prognostic factors, which include age, sex, white blood cell count (wbc) at diagnosis, and the Townsend score (tpi) for which higher values indicates less affluent areas. Both exact residential locations of all patients and their administrative districts (24 districts that make up the whole region) are available.

```
data("LeukSurv" , package="spBayesSurv")
d <- LeukSurv[order(LeukSurv$district), ]
d$time=d$time/max(d$time) #Very important
library(R2BayesX)
```

```
## Loading required package: BayesXsrc
## Loading required package: colorspace
## Loading required package: mgcv
## Loading required package: nlme
## This is mgcv 1.8-27. For overview type 'help("mgcv-package")'.
```

```
nwengland <- read.bnd(system.file("otherdata/nwengland.bnd", package = "spBayesSurv"))
```

```
## Note: map consists of 29 polygons
## Note: map consists of 24 regions
## Reading map ... finished
```

```
adj.mat <- bnd2gra(nwengland)
```

```
## Start neighbor search ...
## progress: 50%
## progress: 100%
## Neighbor search finished.
```

```
E <- diag(diag(adj.mat)) - as.matrix(adj.mat) #Sharing boundaries matrix
d$U<-d$district
E
```

```
##    1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
## 1  0 0 0 1 0 1 0 0 0  1  1  1  0  0  1  1  0  0  0  0  0  0  0  0
## 2  0 0 0 0 1 0 0 0 0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
## 3  0 0 0 0 0 1 0 1 0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0
## 4  1 0 0 0 0 0 0 0 0  0  0  1  1  0  1  0  0  0  0  0  0  0  0  1
## 5  0 1 0 0 0 0 0 0 1  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
## 6  1 0 1 0 0 0 0 0 0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0
## 7  0 0 0 0 0 0 0 0 0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0
## 8  0 0 1 0 0 0 0 0 0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 9  0 0 0 0 1 0 0 0 0  1  0  1  0  1  0  0  0  0  0  0  0  0  0  0
## 10 1 0 1 0 0 1 1 1 1  0  0  1  0  1  0  0  0  0  0  0  0  0  0  0
## 11 1 0 1 0 0 1 0 0 0  0  0  0  0  0  0  1  0  0  1  0  0  0  0  0
## 12 1 0 0 1 0 0 0 0 1  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0
## 13 0 0 0 1 0 0 0 0 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1
## 14 0 1 0 0 1 0 1 0 1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 15 1 0 0 1 0 0 0 0 0  0  0  0  0  0  0  1  0  0  0  1  0  0  0  1
## 16 1 0 0 0 0 0 0 0 0  0  1  0  0  0  1  0  1  0  1  1  0  0  0  0
## 17 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  1  0  1  1  1  1  1  1  0
## 18 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  1  0  1  0  0  1  0  0
## 19 0 0 0 0 0 0 0 0 0  0  1  0  0  0  0  1  1  1  0  0  0  0  0  0
## 20 0 0 0 0 0 0 0 0 0  0  0  0  0  0  1  1  1  0  0  0  0  0  1  1
## 21 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0
## 22 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  1  1  0  0  1  0  0  0  0
## 23 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  1  0  0  1  0  0  0  0  0
## 24 0 0 0 1 0 0 0 0 0  0  0  0  1  0  1  0  0  0  0  1  0  0  0  0
## attr(,"class")
## [1] "gra"
```

```
#convert to sparseMatrix
i_ind=NA
j_ind=NA
for (i in 1:nrow(E)){
  for (j in 1:ncol(E)){
    if (E[i,j]==1) {i_ind=c(i_ind,i)
                    j_ind=c(j_ind,j)}
  }
}

g = sparseMatrix(i=i_ind[2:length(i_ind)],j=j_ind[2:length(j_ind)],dims=c(24,24))
```

```
g
```

```
## 24 x 24 sparse Matrix of class "ngCMatrix"
##
##  [1,] . . . | . | . . . | | | . . | | . . . . . . . .
##  [2,] . . . . | . . . . . . . . | . . . . . . . . . .
##  [3,] . . . . . | . | . | | . . . . . . . . . . . . .
##  [4,] | . . . . . . . . . . | | . | . . . . . . . . |
##  [5,] . | . . . . . | . . . . | . . . . . . . . . . .
##  [6,] | . | . . . . . | | . . . . . . . . . . . . . .
##  [7,] . . . . . . . . | . . . | . . . . . . . . . . .
##  [8,] . . | . . . . . | . . . . . . . . . . . . . . .
##  [9,] . . . . | . . . . | . | . | . . . . . . . . . .
## [10,] | . | . . | | | | . . | . | . . . . . . . . . .
## [11,] | . | . . | . . . . . . . . | . . | . . . . .
## [12,] | . . | . . . . | | . . | . . . . . . . . . . .
## [13,] . . . | . . . . . . | . . . . . . . . . . . . |
## [14,] . | . . | . | . | | . . . . . . . . . . . . . .
## [15,] | . . | . . . . . . . . . . | . . . | . . . |
## [16,] | . . . . . . . . . | . . . | . | . | | . . . .
## [17,] . . . . . . . . . . . . . . | . | | | | | | .
## [18,] . . . . . . . . . . . . . . . | . | . . | . .
## [19,] . . . . . . . . . . | . . . . | | | . . . . . .
## [20,] . . . . . . . . . . . . . . | | | . . . . . | |
## [21,] . . . . . . . . . . . . . . . | . . . . | . .
## [22,] . . . . . . . . . . . . . . . | | . . | . . .
## [23,] . . . . . . . . . . . . . . . | . . | . . . . .
## [24,] . . . | . . . . . . . | . | . . . . | . . . . .
```

```
#Alternatively,
#class(adj.mat) = NULL
#g = inla.read.graph(adj.mat)

####################Fit the spatial survival model using a Besag random effect
Y = inla.surv(time=d$time,event=d$cens)
model_leuk<-inla(formula=Y~age+as.factor(sex)+tpi+f(U,district,model="besag",
                 graph=g,scale.model=TRUE),data=d,family="exponentialsurv",
                 control.predictor = list(compute = TRUE), control.compute = list(dic=TRUE))

#Summary and extract predicted hazard per observation (NOT district)
summary(model_leuk)
```

```
##
## Call:
##    c("inla(formula = Y ~ age + as.factor(sex) + tpi + f(U, district,
##    ", " model = \"besag\", graph = g, scale.model = TRUE), family =
##    \"exponentialsurv\", ", " data = d, control.compute = list(dic =
##    TRUE), control.predictor = list(compute = TRUE))" )
## Time used:
##     Pre = 1.8, Running = 0.591, Post = 0.151, Total = 2.55
## Fixed effects:
##                    mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept)     -0.502 0.152     -0.804   -0.501     -0.208 -0.498   0
## age              0.045 0.002      0.040    0.045      0.049  0.045   0
## as.factor(sex)1  0.060 0.070     -0.077    0.060      0.197  0.060   0
```

```
## tpi              0.030 0.010      0.009    0.030      0.050  0.030   0
##
## Random effects:
##   Name     Model
##     U Besags ICAR model
##
## Model hyperparameters:
##                    mean     sd 0.025quant 0.5quant 0.975quant   mode
## Precision for U 311.54 176.22     101.78   268.98     771.64 202.01
##
## Expected number of effective parameters(stdev): 21.81(1.26)
## Number of equivalent replicates : 47.82
##
## Marginal log-Likelihood:  1158.98
## Posterior marginals for the linear predictor and
##  the fitted values are computed
```

```
hazard.fitted<-exp(model_leuk$summary.linear.predictor$`0.5quant`)
d$hazard.f<-hazard.fitted

#Plot the predicted effect of the district on the hazard
nwsp = bnd2sp(nwengland)
pcens = SpatialPolygonsDataFrame(nwsp,
              aggregate(model_leuk$summary.random$U$mean,
                        list(d1=model_leuk$summary.random$U$ID),
                        function(v) {exp(v)}))
spplot(pcens,"x")
```