Hands on Survival Analysis

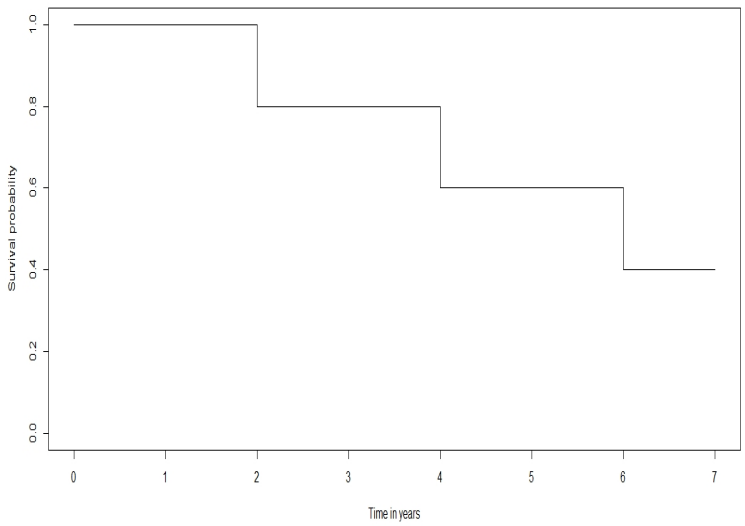
Atanu Bhattacharjee,Ph.D
atanustat@gmail.com
Section of Biostatistics,
Centre for Cancer Epidemiology,
Tata Memorial Centre

Kaplan-Meier estimate of the survival function using the data

```
> library(asaur)
> library(survival)
> tt <- c(7,6,6,2,4)
> cens <- c(0,1,0,1,1)
> Surv(tt, cens)

[1] 7+ 6  6+ 2  4

> result.km <- survfit(Surv(tt, cens) ~ 1, conf.type="log-log")
> summary(result.km)

Call: survfit(formula = Surv(tt, cens) ~ 1, conf.type = "log-log")

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    2      5       1      0.8   0.179        0.204        0.969
    4      4       1      0.6   0.219        0.126        0.882
    6      3       1      0.4   0.219        0.052        0.753

>
```

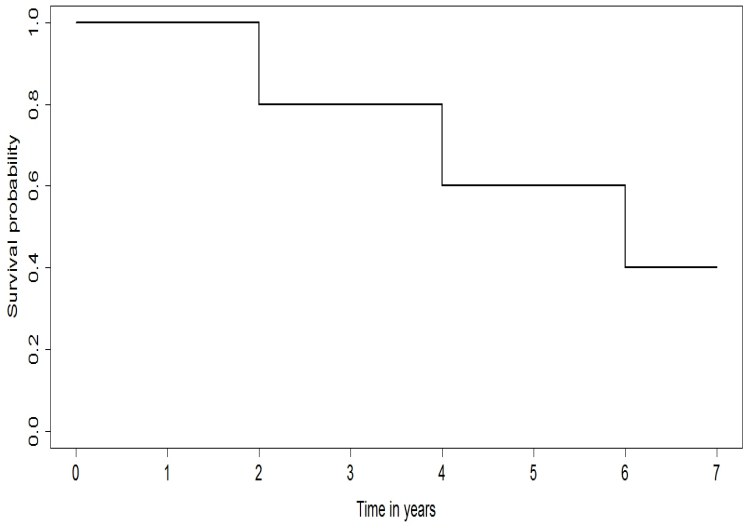
Kaplan-Meier plot

```
> plot(result.km, conf.int=F, ylab="Survival probability",
+       xlab="Time in years")
```



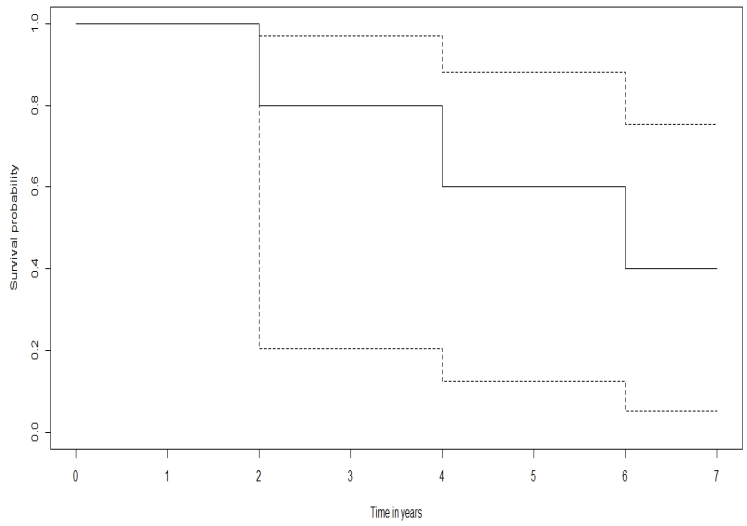
Nicer plot

```
> plot(result.km, conf.int=F, ylab="Survival probability",
+       xlab="Time in years", cex.lab=1.5, cex.axis=1.5, lwd=2)
```



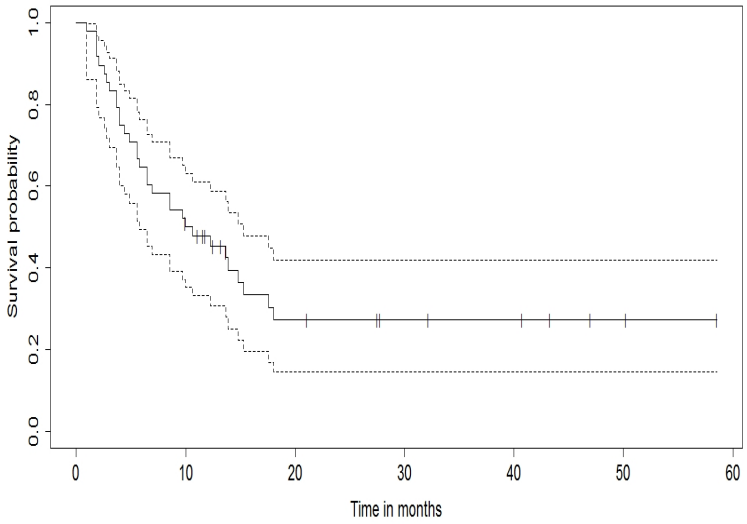
Kaplan-Meier survival curve with 95% confidence intervals

```
> plot(result.km, conf.int=T, ylab="Survival probability",xlab="Time in years")
```



Kaplan-Meier survival curve with 95% confidence intervals

```
> library(asaur)
> library(survival)
> timeMonths <- gastricXelox$timeWeeks*7/30.25
> result.km <- survfit(Surv(timeMonths, delta) ~ 1, conf.type="log-log", data=gastricXelox)
> plot(result.km, mark="|", ylab="Survival probability", xlab="Time in months",
+      cex.axis=1.5, cex.lab=1.5, lwd=1.5)
```

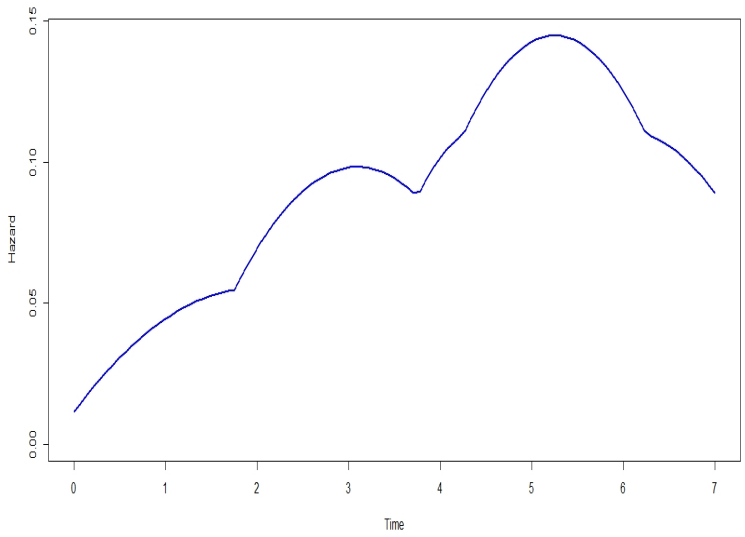


median survival and 95% confidence interval is printed as follows:

```
> result.km

Call: survfit(formula = Surv(timeMonths, delta) ~ 1, data = gastricXelox,
    conf.type = "log-log")

      n  events  median 0.95LCL 0.95UCL
  48.00   32.00   10.30    5.79   15.27

> # median follow-up time
> delta.followup <- 1 - gastricXelox$delta
> survfit(Surv(timeMonths, delta.followup) ~ 1)

Call: survfit(formula = Surv(timeMonths, delta.followup) ~ 1)

      n  events  median 0.95LCL 0.95UCL
   48.0    16.0    27.8    21.1    50.2
```

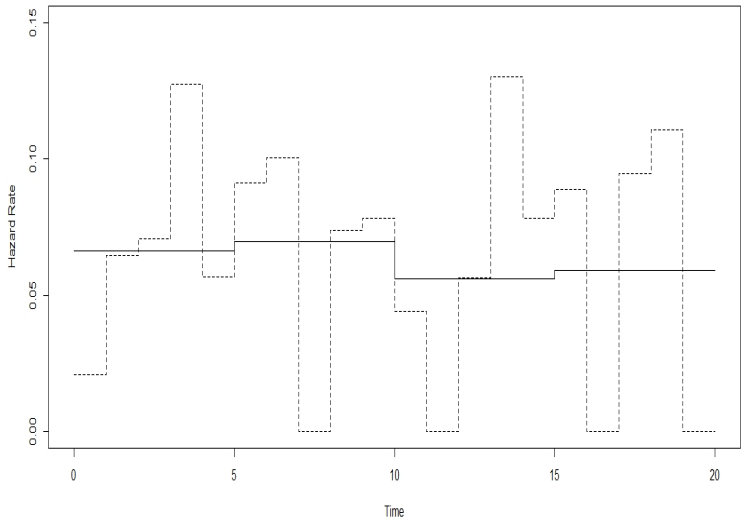
Smooth hazard estimate

```
> library(muhaz)
> t.vec <- c(7,6,6,5,2,4)
> cens.vec <- c(0,1,0,0,1,1)
> result.simple <- muhaz(t.vec, cens.vec, max.time=8,
+      bw.grid=2.25, bw.method="global", b.cor="none")
> plot(result.simple, xlab="Time", ylab="Hazard", lwd=2, col="blue")
```



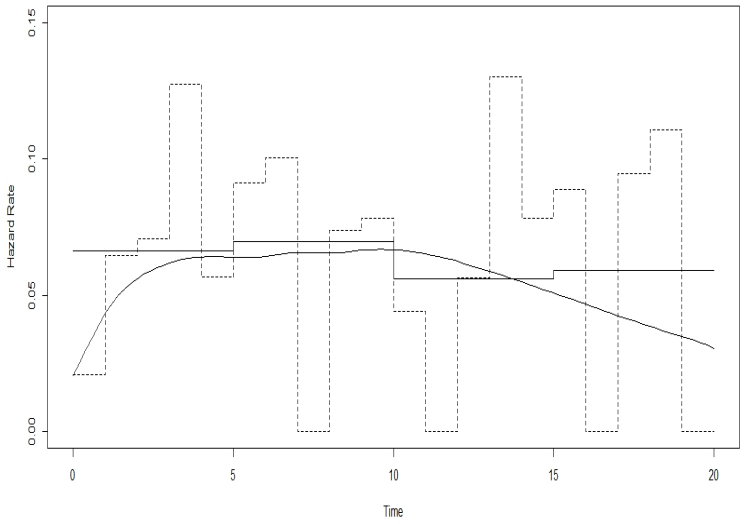
Step versus smooth hazard estimate

```
> result.pe5 <- pehaz(timeMonths, gastricXelox$delta, width=5, max.time=20)

max.time= 20
width= 5
nbins= 4

> plot(result.pe5, ylim=c(0,0.15), col="black")
> result.pe1 <- pehaz(timeMonths, gastricXelox$delta, width=1, max.time=20)

max.time= 20
width= 1
nbins= 20

> lines(result.pe1)
> result.smooth <- muhaz(timeMonths, gastricXelox$delta, bw.smooth=20,
+       b.cor="left", max.time=20)
> lines(result.smooth)
```



Kaplan-Meier survival plot and smoothed survival estimate for gastricXelox data

```
> haz <- result.smooth$haz.est
> times <- result.smooth$est.grid
> surv <- exp(-cumsum(haz[1:(length(haz)-1)]*diff(times)))
> result.km <- survfit(Surv(timeMonths, gastricXelox$delta) ~ 1,
+ conf.type="none")
> plot(result.km, conf.int=T, mark="|", xlab="Time in months",
+ xlim=c(0,30), ylab="Survival probability")
> lines(surv ~ times[1:(length(times) - 1)])
```



Left truncation simple example

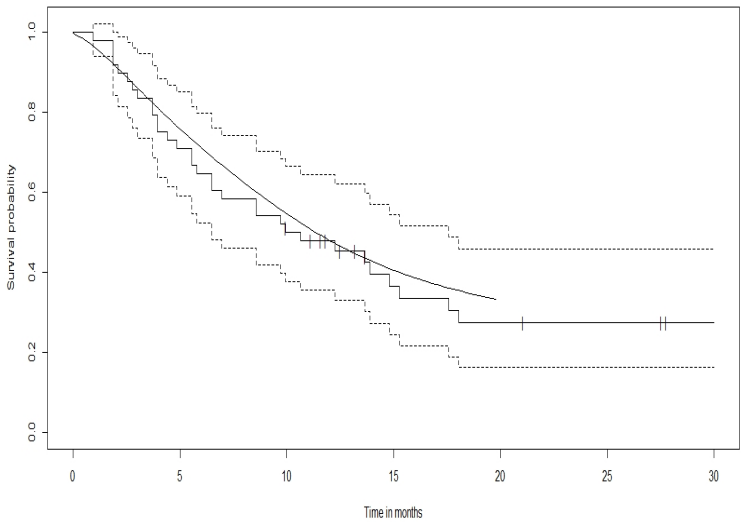
```
> tt <- c(7, 6, 6, 5, 2, 4)
> status <- c(0, 1, 0, 0, 1, 1)
> backTime <- c(-2, -5, -3, -3, -2, -5)
> tm.enter <- -backTime
> tm.exit <- tt - backTime
> result.left.trunc.km <- survfit(Surv(tm.enter, tm.exit, status,
+    type="counting") ~ 1, conf.type="none")
> summary(result.left.trunc.km)

Call: survfit(formula = Surv(tm.enter, tm.exit, status, type = "counting") ~
    1, conf.type = "none")

 time n.risk n.event censored survival std.err
    4      4       1        0    0.750   0.217
    9      4       1        3    0.562   0.230
   11      1       1        0    0.000     NaN

> result.left.trunc.naa <- survfit(Surv(tm.enter, tm.exit, status,
+    type="counting") ~ 1, type="fleming-harrington", conf.type="none")
> summary(result.left.trunc.naa)

Call: survfit(formula = Surv(tm.enter, tm.exit, status, type = "counting") ~
    1, type = "fleming-harrington", conf.type = "none")

 time n.risk n.event censored survival std.err
    4      4       1        0    0.779   0.195
    9      4       1        3    0.607   0.214
   11      1       1        0    0.223   0.237
```

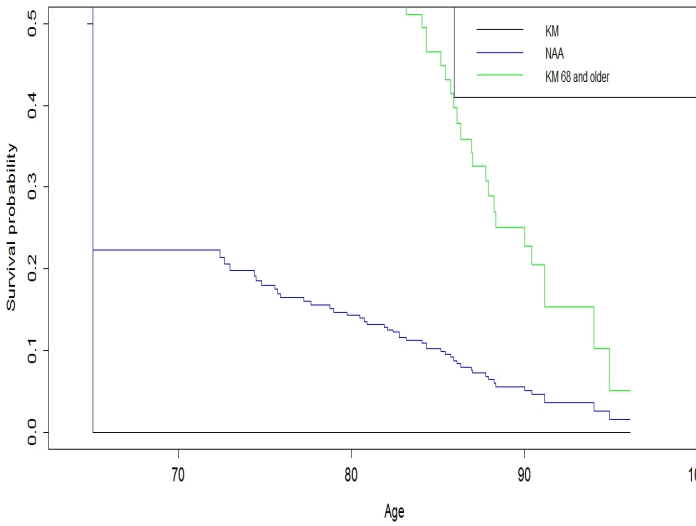
Channing House example

```
> library(asaur)
> # define entry and exit times
> ChanningHouse <- within(ChanningHouse, {
+   entryYears <- entry/12
+   exitYears <- exit/12})
> head(ChanningHouse)

   sex entry exit time cens exitYears entryYears
1 Male   782  909  127    1  75.75000   65.16667
2 Male  1020 1128  108    1  94.00000   85.00000
3 Male   856  969  113    1  80.75000   71.33333
4 Male   915  957   42    1  79.75000   76.25000
5 Male   863  983  120    1  81.91667   71.91667
6 Male   906 1012  106    1  84.33333   75.50000

> ChanningMales <- ChanningHouse[ChanningHouse$sex == "Male",]
> result.km <- survfit(Surv(entryYears, exitYears, cens, type="counting") ~ 1,
+                      data=ChanningMales)
```

Channing House males survival estimates

```
> plot(result.km, xlim=c(64, 101), xlab="Age", ylab="Survival probability",
+   conf.int=F,
+   cex.axis=1.3, cex.lab=1.3)
> result.naa <- survfit(Surv(entryYears, exitYears, cens, type="counting") ~ 1,
+                       type="fleming-harrington", data=ChanningMales)
> lines(result.naa, col="blue", conf.int=F)
> result.km.68 <- survfit(Surv(entryYears, exitYears, cens, type="counting") ~ 1,
+                         start.time=68, data=ChanningMales)
> lines(result.km.68, col="green", conf.int=F)
> legend("topright", legend=c("KM", "NAA", "KM 68 and older"),
+        lty=1, col=c("black", "blue", "green"))
```





Weibull Sampling Model

```
Gamma priors on the alpha's
model {
for(i in 1:n[1]) t1[i] ~ dweib(alpha[1],lambda[1]) }

for(i in 1:n[2]) t2[i] ~ dweib(alpha[2],lambda[2]) }

lambda[1] ~ dgamma(a[1],b[1])
alpha[1] ~ dgamma(c[1],d[1])
lambda[2] ~ dgamma(a[2],b[2])
alpha[2] ~ dgamma(c[2],d[2])
med[1] <- pow(log(2)/lambda[1],1/alpha[1])
med[2] <- pow(log(2)/lambda[2],1/alpha[2])
relmedian <- med[1]/med[2]
S[1] <- exp(-pow(24,alpha[1])*lambda[1])
S[2] <- exp(-pow(24,alpha[2])*lambda[2])
}
list(n=c(17,16), a= c(0.001,0.001), b=c(0,001,0,001), c=c(0.001,0.001),d=c(0.001,0.001),
t1=c(65,156,100,134,16,108,121,4,39,143,56,26,22,1,1,5,65),
t2=c(56,65,17,7,16,22,3,4,2,3,8,4,3,30,4,43))
list(alpha=c(1,1), lambda=c(1,1))
```

Survival Analysis with Bayesian AFT model1

```
model
{
C<-10000
for(i in 1:90) {
temp[i] <-(log(timeF[i]+0.01)-log(lam[i]))/sigma
log(lam[i]) <- beta0+mu[treat[i]]

logistic
s[i]<-1/(1+exp(temp[i]))
f[i]<-exp(temp[i])*pow(s[i],2)
Log likelihood
L[i]<-indic[i]*log(f[i]/(sigma*(timeF[i]+0.01)))+(1-indic[i])*log(s[i])
Poisson zeroes trick
zeros[i]<-0
new[i]<- -L[i]+C
zeros[i] dpois(new[i])
CPinv[i]<-exp(-L[i])
cpo[i]<-pow(CPinv[i],-1)
Lcpo[i]<-log(cpo[i]+0.001)
} diff<-mu[1]-mu[2]
MargL<-sum(Lcpo[])
tausigma<-pow(sigma,-2)
sigma dunif(0,5)
beta0   dnorm(0.0,tau0)
for(i in 1:2) mu[i]  dnorm(0.0, tau1)}
tau0<-pow(sd0,-2)
tau1<-pow(sd1,-2)
sd0 dunif(0,2)
sd1 dunif(0,2)
}
```

Survival Analysis with Bayesian AFT model2

```
model
{
C<-10000
for(i in 1:90) {
temp[i] <-(log(timeF[i]+0.01)-log(lam[i]))/sigma
log(lam[i]) <- beta0+mu[treat[i]]+v[i]  v[i] dnorm(0,tauv)
logistic
s[i]<-1/(1+exp(temp[i]))
f[i]<-exp(temp[i])*pow(s[i],2)
log likelihood
L[i]<-indic[i]*log(f[i]/(sigma*(timeF[i]+0.01)))+(1-indic[i])*log(s[i])
CPinv[i]<-exp(-L[i])
cpo[i]<-pow(CPinv[i],-1)
Lcpo[i]<-log(cpo[i])
Poisson zeroes trick
zeros[i]<-0
new[i]<- -L[i]+C
zeros[i] dpois(new[i])
}
MargL<-sum(Lcpo[])
tausigma<-pow(sigma,-2)
sigma dunif(0,5)
beta0   dnorm(0.0,tau0)
for(i in 1:2) mu[i]  dnorm(0.0, tau1)}
tau1<-pow(sd1,-2)
tauv<-pow(sdv,-2)
sdv dunif(0,5)
sd0 dunif(0,5)
sd1 dunif(0,5)
}
```

Survival Analysis with Bayesian AFT model 3

```
model
{
C<-10000
for(i in 1:90) {
temp[i] <-(log(timeF[i]+0.01)-log(lam[i]))/sigma[treat[i]]
log(lam[i]) <- beta0+mu[treat[i]]
Logistic
s[i]<-1/(1+exp(temp[i]))
f[i]<-exp(temp[i])*pow(s[i],2)
log likelihood
L[i]<-indic[i]*log(f[i]/(sigma[treat[i]]*(timeF[i]+0.01)))+(1-indic[i])*log(s[i])
CPinv[i]<-exp(-L[i])
cpo[i]<-pow(CPinv[i],-1)
Lcpo[i]<-log(cpo[i])
Poisson zeroes trick
zeros[i]<-0
new[i]<- -L[i]+C
zeros[i] dpois(new[i])
}
MargL<-sum(Lcpo[])
for( j in 1:2) {
sigma[j] dgamma(2.0,0.5) }
beta0   dnorm(0.0,tau0)
for(i in 1:2) mu[i]  dnorm(0.0, tau1)}
tau0<-pow(sd0,-2)
tau1<-pow(sd1,-2)
sd0 dunif(0,2)
sd1 dunif(0,2)
}
```

Survival Analysis with Bayesian spatial AFT model1

```
model{
for (i in 1: regions){
for(j in cum[i]+1:cum[i+1])
{
adj1[i,j]<- adj[j]
}}
for(i in 1:Nsubj) {
temp denotes (logt-beta0-beta*x-frailty)/sigma1
temp[i] <-(log(time[i]+0.1)-beta0 - beta[1]*age[i] - beta[2]*marital[i] -
beta[3]*race[i]- beta[4]*stage[i]-W[county[i]])/sigma
survival distribution and density function of standard normal
s[i]<-1-phi(temp[i])
f[i]<-pow(2*3.14, -0.5)*exp(-0.5*pow(temp[i],2))
survival distribution and density function of extreme value distribution
s[i]<-exp(-exp(temp[i]))
f[i]<-s[i]*exp(temp[i])
survival distribution and density function of logistic distribution
s[i]<-1/(1+exp(temp[i]))
f[i]<-exp(temp[i])*pow(s[i],2)
Loglikelihood Function
L[i]<-status[i]*log(f[i]/(sigma*(time[i]+0.01)))+(1-status[i])*log(s[i])
Poisson zero trick
zeros[i]<-0
new[i]<- -L[i]
zeros[i] dpois(new[i])
}  Spatial from independent normal distribution
for (i in 1:regions)
{ W[i] ~ dnorm(0, tau)
intercept[i]<-beta0+W[i]}
Parameter Prior for the parameters in the AFT model
beta0 ~ dnorm(0.0,0.001)
for(i in 1:4) beta[i] ~ dnorm(0.0, 0.001)}
inversesigma ~ dgamma(0.001,0.001)
sigma<-1/inversesigma
Parameters normal model
tau   dgamma(0.001, 0.001)
}
```

Survival Analysis with Bayesian spatial AFT model2

```
model {
for (i in 1: regions) {
for(j in cum[i]+1:cum[i+1])
{
adj1[i,j]<- adj[j]
}} for(i in 1:Nsubj) {
(logt-beta0-beta*x-frailty)/sigma1
temp[i] <-(log(time[i]+0.1)-beta0 - beta[1]*age[i] - beta[2]*marital[i] -
beta[3]*race[i]- beta[4]*stage[i]-W[county[i]])/sigma
s[i]<-1-phi(temp[i])
f[i]<-pow(2*3.14, -0.5)*exp(-0.5*pow(temp[i],2))
s[i]<-exp(-exp(temp[i]))
f[i]<-s[i]*exp(temp[i])
s[i]<-1/(1+exp(temp[i]))
f[i]<-exp(temp[i])*pow(s[i],2)
L[i]<-status[i]*log(f[i]/(sigma*(time[i]+0.01)))+(1-status[i])*log(s[i])
zeros[i]<-0
new[i]<- -L[i]
zeros[i] dpois(new[i])
}
for(j in 1:sumNum) {
weights[j] <- 1}
W[1:regions]   car.normal(adj[], weights[], num[], tau)
W.mean <- mean(W[]
)
beta0   dnorm(0.0,0.001)
for(i in 1:4) beta[i]   dnorm(0.0, 0.001)
inversesigma dgamma(0.001,0.001)
sigma<-1/inversesigma
tau   dgamma(0.001, 0.001)
}
```

Survival Analysis with Bayesian spatial AFT model3

```
model {
for (i in 1: regions) {
for(j in cum[i]+1:cum[i+1])
{
adj1[i,j]<- adj[j]
}}
for(i in 1:Nsubj) {
(logt-beta0-beta*x-frailty)/sigma1
temp[i] <-(log(time[i]+0.1)-beta0 - beta[1]*age[i] - beta[2]*marital[i] -
beta[3]*race[i]- beta[4]*stage[i]-W[county[i]])/sigma
s[i]<-1-phi(temp[i])
f[i]<-pow(2*3.14, -0.5)*exp(-0.5*pow(temp[i],2))
s[i]<-exp(-exp(temp[i]))
f[i]<-s[i]*exp(temp[i])
s[i]<-1/(1+exp(temp[i]))
f[i]<-exp(temp[i])*pow(s[i],2)
L[i]<-status[i]*log(f[i]/(sigma*(time[i]+0.01)))+(1-status[i])*log(s[i])
Poisson zero trick
zeros[i]<-0
new[i]<- -L[i]
zeros[i] dpois(new[i])
}
for(j in 1:sumNum)  weights[j] <- 1
W1[1:regions]   car.normal(adj[], weights[], num[], tau)
W1.mean <- mean(W1[])
for (j in 1:regions) W2[j]   dnorm(0, tau1)
W[j]<-W1[j]+W2[j]
intercept[j]<-beta0+W[j]
}
beta0   dnorm(0.0,0.001)
for(i in 1:4) beta[i]   dnorm(0.0, 0.001)
inversesigma dgamma(0.001,0.001)
sigma<-1/inversesigma
Parameter in the mix car model
tau   dgamma(0.001, 0.001)
tau1 dgamma(0.001,0.001)
```