Implementing Approximate inference for Latent Gaussian Markov Random Field Models: the INLA package for R.

Implementing Integrated Nestad Laplace Approximation

All procedures needed to perform INLA need to be carefully implemented to achieve an optimal speed

- The `GMRFLib`-library
- The `inla` program
- The `INLA` package for `R`

Implementing Integrated Nestad Laplace Approximation

All procedures needed to perform INLA need to be carefully implemented to achieve an optimal speed

- The `GMRFLib`-library
  - Require a lot of C programming, not user friendly
- The `inla` program
- The `INLA` package for `R`

Implementing Integrated Nestad Laplace Approximation

All procedures needed to perform INLA need to be carefully implemented to achieve an optimal speed

- The `GMRFLib`-library
- The `inla` program
  - Interface to `GMRFLib`-library
  - Avoids all C programming
  - Requires to write the `ini` file
  - Requires to write input files in a special format
  - Output files can be "tricky" to read
- The `INLA` package for `R`

Implementing Integrated Nestad Laplace Approximation

All procedures needed to perform INLA need to be carefully implemented to achieve an optimal speed

- The `GMRFLib`-library
- The `inla` program
- The `INLA` package for `R`
  - Interface to the `inla` program
  - Similar to other `R` packages
  - Sligthly less flexible than the `inla` program (but can be used as a starting point.)

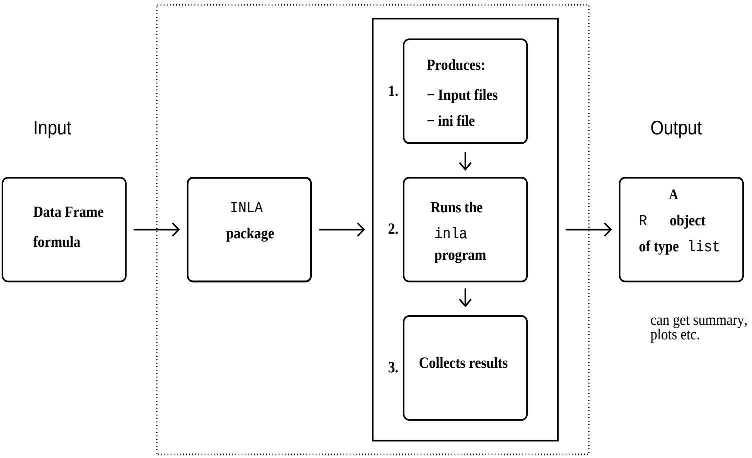
The INLA package for R

- Can be downloaded from `http://www.math.ntnu.no/ hrue/GMRFLib/R-INLA/`
- Available for Unix, Windows and Mac
- Requires the `inla` program to be also installed

The INLA package for R

- Can be downloaded from
  `http://www.math.ntnu.no/ hrue/GMRFLib/R-INLA/`
- Available for Unix, Windows and Mac
- Requires the `inla` program to be also installed

The INLA package for R



Model specification the `INLA` package I

Assume the following model:

$$\begin{aligned} y &\sim \pi(y|\lambda) \\ \eta &= g(\lambda) = \beta_0 + \beta_1 x_1 + \beta_2 x_1 + f(x_3) \end{aligned}$$

where

$$\begin{aligned} x_1, x_2 & \quad \text{are covariate with linear effect} \\ \beta_i &\sim \mathcal{N}(0, \tau_1^{-1}) \\ x_3 & \quad \text{can be spatial effect, random effect ....} \\ f_1, f_2, \ldots &\sim \mathcal{N}(0, Q_f^{-1}(\tau_2)) \end{aligned}$$

Model specification the INLA package II

The model is specified in R through a formula, similar to the one used in the glm routine:

```
> formula <- y~ x1 + x2 + f(x3)
```

The f() function is used to specify non-linear effects in the model. The implemented model are :

- iid random effects
- rw1 rw2 ar1 smooth effect of covariates or time effect
- seasonal seasonal effect
- besag spatial effect (CAR model)
- generic user defined precision matrix

Model specification the INLA package II

The model is specified in R through a formula, similar to the one used in the glm routine:

```
> formula <- y~ x1 + x2 + f(x3)
```

The f() function is used to specify non-linear effects in the model. The implemented model are :

- iid random effects
- rw1 rw2 ar1 smooth effect of covariates or time effect
- seasonal seasonal effect
- besag spatial effect (CAR model)
- generic user defined precision matrix

Main functions of the `INLA` package

- `f()` : Helps defining non linear effects in the model specification.
- `inla()` : Performs a Bayesian analysis of additive models
- `surv.inla()` : Performs a Bayesian analysis of some survival models (experimental!!)

All functions are provided with a help file which can be viewed using

```
> ?inla
```

Additional functions of the `INLA` package

- `summary()` : produces a summary of the main results from a fitted model
- `plot()` : produces some plots from the fitted model

Some examples of usage of the INLA package

- A mixed effect model
- A model with time series component
- A model with spatial component
- A model for survival data

Mixed model with repeated poisson counts I

Seizure counts in a randomised trial of anti-convulsant therpay in epilepsy. From WinBUGS manual.

Patient	y1	y2	y3	y4	Trt	Base	Age
1	5	3	3	3	0	11	31
2	3	5	3	3	0	11	30
....							
59	1	4	3	2	1	12	37

Mixed model with repeated poisson counts II
The model

$$
\begin{aligned}
y_{jk} &\sim \text{Poisson}(\mu_{jk});\ j = 1, \ldots, 59; k = 1, \ldots, 4 \\
log(\mu_{jk}) &= \alpha_0 + \alpha_1 \log(\text{Base}_j/4) + \alpha_2 \text{Trt}_j \\
&\quad +\alpha_3 \text{Trt}_j \log(\text{Base}_j/4) + \alpha_4 \text{Age}_j \\
&\quad +\alpha_5 V4 + \text{Ind}_j + \beta_{jk} \\
\alpha_i &\sim \mathcal{N}(0, \tau_\alpha)\ \tau_\alpha \quad \text{known} \\
\text{Ind}_j &\sim \mathcal{N}(0, \tau_{\text{Ind}})\ \tau_{\text{Ind}} \sim \text{Gamma}(a_1, b_1) \\
\beta_j k &\sim \mathcal{N}(0, \tau_\beta)\ \tau_\beta \sim \text{Gamma}(a_2, b_2)
\end{aligned}
$$

Mixed model with repeated poisson counts III
Fitting the model using INLA

The `Epil` data frame:

y	Trt	Base	Age	V4	rand	Ind
5	0	11	31	0	1	1
3	0	11	31	0	2	1
⋮						

Specifying the model:

```
> formula <- y ~ log(Base/4) + Trt + I(Trt *
log(Base)) + log(Age) + V4 + f(Ind, model = "iid") +
f(rand, model = "iid")
```

Running `inla`

```
> model=inla(formula,family="poisson")
```

Mixed model with repeated poisson counts III

Fitting the model using INLA

The `Epil` data frame:

y	Trt	Base	Age	V4	rand	Ind
5	0	11	31	0	1	1
3	0	11	31	0	2	1
⋮						

Specifying the model:

```
> formula <- y ~ log(Base/4) + Trt + I(Trt *
log(Base)) + log(Age) + V4 + f(Ind, model = "iid") +
f(rand, model = "iid")
```

Running inla

```
> model=inla(formula,family="poisson")
```

Mixed model with repeated poisson counts III

Fitting the model using INLA

The `Epil` data frame:

y	Trt	Base	Age	V4	rand	Ind
5	0	11	31	0	1	1
3	0	11	31	0	2	1
⋮						

Specifying the model:

```
> formula <- y ~ log(Base/4) + Trt + I(Trt *
log(Base)) + log(Age) + V4 + f(Ind, model = "iid") +
f(rand, model = "iid")
```

Running `inla`

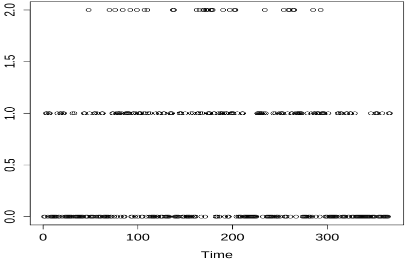
```
> model=inla(formula,family="poisson")
```

Mixed model with repeated poisson counts IV

Some option of the `inla` function:

- `verbose=TRUE` shows the steps of the process
- `keep.data.files=TRUE` keeps the `ini` file and all built input files.
- `keep.results.files=TRUE` keeps results files (but they're binary files!)

A model with time series component I



Number of days in Tokyo with rainfall above 1 mm in 1983-84.
We want to estimate the probability of rain p_t for calendar day $t = 1, \ldots, 366$

A model with time series component II
The model

$$
\begin{aligned}
y_t &\sim \text{Binomial}(n_t, p_t);\;\; t = 1, \ldots, 365 \\
p_t &= \tfrac{\exp(\eta_t)}{1+\exp(\eta_t)} \\
\eta_t &= f(t) \\
\mathbf{f} &= \{f_1, \ldots, f_{366}\} \sim \text{cyclic RW2}(\tau) \\
\tau &\sim \text{Gamma}(1, 0.0001)
\end{aligned}
$$

A model with time series component III
Fitting the model using INLA

The `Tokyo` data frame:

y	n	time
0	2	1
0	2	2
1	2	3
⋮		

A model with time series component III
Fitting the model using INLA

The Tokyo data frame:

y	n	time
0	2	1
0	2	2
1	2	3
⋮		

Specifying the model:

```
> formula <- y ~
f(time,model="rw2",cyclic=TRUE,param=c(1,0.0001))-1
```

A model with time series component III
Fitting the model using `INLA`

The `Tokyo` data frame:

y	n	time
0	2	1
0	2	2
1	2	3
⋮		

Specifying the model:

```
> formula <- y ~
f(time,model="rw2",cyclic=TRUE,param=c(1,0.0001))-1
```

A model with time series component III
Fitting the model using INLA

The Tokyo data frame:

y	n	time
0	2	1
0	2	2
1	2	3
⋮		

Specifying the model:

```
> formula <- y ~
f(time,model="rw2",cyclic=TRUE,param=c(1,0.0001))-1
```

A model with time series component III

Fitting the model using INLA

The Tokyo data frame:

y	n	time
0	2	1
0	2	2
1	2	3
⋮		

Specifying the model:

```
> formula <- y ~
f(time,model="rw2",cyclic=TRUE,param=c(1,0.0001))-1
```

Running inla

```
> mod.tokyo <-
inla(formula,family="binomial",Ntrials=n,data=Tokyo)
```

A model with time series component III

Fitting the model using INLA

The Tokyo data frame:

y	n	time
0	2	1
0	2	2
1	2	3
⋮		

Specifying the model:

```
> formula <- y ~
f(time,model="rw2",cyclic=TRUE,param=c(1,0.0001))-1
```

Running inla

```
> mod.tokyo <-
inla(formula,family="binomial",Ntrials=n,data=Tokyo)
```

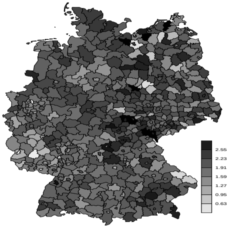
Geoadditive model I
Disease mapping in Germany

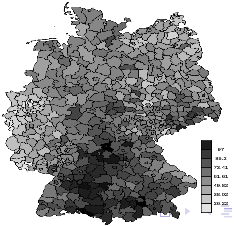
Larynx cancer mortality counts are observed in the 544 district of Germany from 1986 to 1990 and level of smoking consumption (100 possible values).

y_i, $i = 1, \dots, 544$ counts of cancer mortality in Region i

E_i, $i = 1, \dots, 544$ known variable accounting for demographic variation in Region i

c_i, $i = 1, \dots, 544$ level of smoking consumption registered in Region i





Geoadditive model II
The model

$$
\begin{aligned}
y_i &\sim \text{Poisson}\{E_i \exp(\eta_i)\}; \;\; i = 1, \ldots, 544 \\
\eta_i &= \mu + f(c_i) + f_s(s_i) + u_i
\end{aligned}
$$

where:

- $f(c_i)$ is a smooth effect of the covariate

$$\mathbf{f} = \{f_1, \ldots, f_{100}\} \sim \text{RW2}(\tau_f)$$

- $f_s(s_i)$ is a spatial effect modelled as an intrinsic GMRF

$$f_s(s) | f_s(s'), s \neq s', \lambda_s \sim \mathcal{N}\left(\frac{1}{n_s} \sum_{s \sim s'} f_s(s'), \frac{\tau_{f_s}}{n_s}\right)$$

- u_i is a random effect

$$\mathbf{u} = \{u_1, \ldots, u_{544}\} \sim \mathbf{N}(0, \tau_u \mathbf{I})$$

- μ is an intercept term $\mu \sim \mathcal{N}(0, 0.0001)$

Geoadditive model II
The model

$$\begin{aligned} y_i &\sim \text{Poisson}\{E_i \exp(\eta_i)\}; \; i = 1, \ldots, 544 \\ \eta_i &= \mu + f(c_i) + f_s(s_i) + u_i \end{aligned}$$

where:

- $f(c_i)$ is a smooth effect of the covariate

$$\mathbf{f} = \{f_1, \ldots, f_{100}\} \sim \text{RW2}(\tau_f)$$

- $f_s(s_i)$ is a spatial effect modelled as an intrinsic GMRF

$$f_s(s)|f_s(s'), s \neq s', \lambda_s \sim \mathcal{N}(\frac{1}{n_s}\sum_{s \sim s'} f_s(s'), \frac{\tau_{f_s}}{n_s})$$

- u_i is a random effect

$$\mathbf{u} = \{u_1, \ldots, u_{544}\} \sim \mathbf{N}(0, \tau_u \mathbf{I})$$

- μ is an intercept term $\mu \sim \mathcal{N}(0, 0.0001)$

Geoadditive model II
The model

$$\begin{aligned} y_i &\sim \text{Poisson}\{E_i \exp(\eta_i)\};\ i = 1, \ldots, 544 \\ \eta_i &= \mu + f(c_i) + f_s(s_i) + u_i \end{aligned}$$

where:

- $f(c_i)$ is a smooth effect of the covariate

$$\mathbf{f} = \{f_1, \ldots, f_{100}\} \sim \text{RW2}(\tau_f)$$

- $f_s(s_i)$ is a spatial effect modelled as an intrinsic GMRF

$$f_s(s)|f_s(s'), s \neq s', \lambda_s \sim \mathcal{N}(\frac{1}{n_s}\sum_{s\sim s'} f_s(s'), \frac{\tau_{f_s}}{n_s})$$

- u_i is a random effect

$$\mathbf{u} = \{u_1, \ldots, u_{544}\} \sim \mathbf{N}(0, \tau_u \mathbf{I})$$

- μ is an intercept term $\mu \sim \mathcal{N}(0, 0.0001)$

Geoadditive model II
The model

$$\begin{aligned} y_i &\sim \text{Poisson}\{E_i \exp(\eta_i)\};\ i = 1, \ldots, 544 \\ \eta_i &= \mu + f(c_i) + f_s(s_i) + u_i \end{aligned}$$

where:

- $f(c_i)$ is a smooth effect of the covariate

$$\mathbf{f} = \{f_1, \ldots, f_{100}\} \sim \text{RW2}(\tau_f)$$

- $f_s(s_i)$ is a spatial effect modelled as an intrinsic GMRF

$$f_s(s)|f_s(s'), s \neq s', \lambda_s \sim \mathcal{N}(\frac{1}{n_s}\sum_{s\sim s'} f_s(s'), \frac{\tau_{f_s}}{n_s})$$

- u_i is a random effect

$$\mathbf{u} = \{u_1, \ldots, u_{544}\} \sim \mathbf{N}(0, \tau_u \mathbf{I})$$

- μ is an intercept term $\mu \sim \mathcal{N}(0, 0.0001)$

Geoadditive model II
The model

$$\begin{aligned} y_i &\sim \text{Poisson}\{E_i \exp(\eta_i)\}; \;\; i = 1, \ldots, 544 \\ \eta_i &= \mu + f(c_i) + f_s(s_i) + u_i \end{aligned}$$

where:

- $f(c_i)$ is a smooth effect of the covariate

$$\mathbf{f} = \{f_1, \ldots, f_{100}\} \sim \text{RW2}(\tau_f)$$

- $f_s(s_i)$ is a spatial effect modelled as an intrinsic GMRF

$$f_s(s)|f_s(s'), s \neq s', \lambda_s \sim \mathcal{N}(\frac{1}{n_s}\sum_{s\sim s'} f_s(s'), \frac{\tau_{f_s}}{n_s})$$

- u_i is a random effect

$$\mathbf{u} = \{u_1, \ldots, u_{544}\} \sim \mathbf{N}(0, \tau_u \mathbf{I})$$

- μ is an intercept term $\mu \sim \mathcal{N}(0, 0.0001)$

Geoadditive model II
The model cont.

Prior for the precision parameters:

$$\begin{aligned}
\tau_f &\sim \text{Gamma}(1, 0.00005) \\
\tau_{f_s} &\sim \text{Gamma}(1, 0.05) \\
\tau_u &\sim \text{Gamma}(1, 0.001)
\end{aligned}$$

Geoadditive model III
Fitting the model using INLA

The `Germany` data frame:

region	E	Y	x
0	7.965008	8	56
1	22.836219	22	65

The model is:

$$\eta_i = \mu + f(c_i) + f_s(s_i) + u_i$$

- The data set has to contain *one separate column for each term specified through* `f()` so in this case we have to add one column..

```
> Germany<-cbind(Germany,region.struct=Germany$region)
```

- We also need the graph file where the nighbourhood structure is specified `germany.graph`

Geoadditive model III
Fitting the model using `INLA`

The `Germany` data frame:

region	E	Y	x
0	7.965008	8	56
1	22.836219	22	65

The model is:

$$\eta_i = \mu + f(c_i) + f_s(s_i) + u_i$$

- The data set has to contain *one separate column for each term specified through* `f()` so in this case we have to add one column..

```
> Germany<-cbind(Germany,region.struct=Germany$region)
```

- We also need the graph file where the nighbourhood structure is specified `germany.graph`

Geoadditive model III

Fitting the model using `INLA`

The `Germany` data frame:

region	E	Y	x
0	7.965008	8	56
1	22.836219	22	65

The model is:

$$\eta_i = \mu + f(c_i) + f_s(s_i) + u_i$$

- The data set has to contain *one separate column for each term specified through* `f()` so in this case we have to add one column..

```
> Germany<-cbind(Germany,region.struct=Germany$region)
```

- We also need the graph file where the nighbourhood structure is specified `germany.graph`

Geoadditive model III
Fitting the model using INLA

The new data set is:

region	E	Y	x	region.struct
0	7.965008	8	56	0
1	22.836219	22	65	1

Then the formula is

```
formula <-
Y~f(region.struct,model="besag",graph.file="germany.graph",
param=c(1,0.00005))+f(x,model="rw2",param=c(1,0.05))+f(region)
```

Geoadditive model III
Fitting the model using INLA

The new data set is:

region	E	Y	x	region.struct
0	7.965008	8	56	0
1	22.836219	22	65	1

Then the formula is

```
formula <-
Y~f(region.struct,model="besag",graph.file="germany.graph",
param=c(1,0.00005))+f(x,model="rw2",param=c(1,0.05))+f(region)
```

Geoadditive model III
Fitting the model using INLA

The new data set is:

region	E	Y	x	region.struct
0	7.965008	8	56	0
1	22.836219	22	65	1

Then the formula is

```
formula <-
Y~f(region.struct,model="besag",graph.file="germany.graph",
param=c(1,0.00005))+f(x,model="rw2",param=c(1,0.05))+f(region)
```

Geoadditive model III
Fitting the model using INLA

The new data set is:

region	E	Y	x	region.struct
0	7.965008	8	56	0
1	22.836219	22	65	1

Then the formula is

```
formula <-
Y~f(region.struct,model="besag",graph.file="germany.graph",
param=c(1,0.00005))+f(x,model="rw2",param=c(1,0.05))+f(region)
```

Geoadditive model III
Fitting the model using INLA

The new data set is:

region	E	Y	x	region.struct
0	7.965008	8	56	0
1	22.836219	22	65	1

Then the formula is

```
formula <-
Y~f(region.struct,model="besag",graph.file="germany.graph",
param=c(1,0.00005))+f(x,model="rw2",param=c(1,0.05))+f(region)
```

The location of the graph file has to be provided here (the graph file cannot be loaded in R)

The graph file

The `germany.graph` file:

```
544
0   1   11
1   2   9   10
2   4   5   7   14   386
⋮
```

- Total number of nodes in the graph
- Identifier for the node
- Number of neoghbours
- Identifiers for the neighbours

The graph file

The `germany.graph` file:

544					
0	1	11			
1	2	9	10		
2	4	5	7	14	386
⋮					

- Total number of nodes in the graph
- Identifier for the node
- Number of neoghbours
- Identifiers for the neighbours

The graph file

The `germany.graph` file:

544					
0	1	11			
1	2	9	10		
2	4	5	7	14	386
⋮					

- Total number of nodes in the graph
- Identifier for the node
- Number of neoghbours
- Identifiers for the neighbours

The graph file

The `germany.graph` file:

544					
0	1	11			
1	2	9	10		
2	4	5	7	14	386
⋮					

- Total number of nodes in the graph
- Identifier for the node
- Number of neoghbours
- Identifiers for the neighbours

The graph file

The `germany.graph` file:

```
544
0   1   11
1   2   9   10
2   4   5   7   14   386
⋮
```

- Total number of nodes in the graph
- Identifier for the node
- Number of neoghbours
- Identifiers for the neighbours

Geoadditive model III
Running the model using INLA

```
mod <-
inla(formula,family="poisson",data=Germany,E=E,
control.inla=list(h=0.01),verbose=TRUE)
```

- Defines the known variable adjusting for demographic variation E_i (*only used for Poisson likelihood*)
- Controls some parameters in the `inla` program, in this case the step length to compute the hessian at the mode of $\pi(\boldsymbol{\theta}|\mathbf{y})$

Geoadditive model III
Running the model using INLA

```
mod <-
inla(formula,family="poisson",data=Germany,E=E,
control.inla=list(h=0.01),verbose=TRUE)
```

- Defines the known variable adjusting for demographic variation E_i (*only used for Poisson likelihood*)
- Controls some parameters in the inla program, in this case the step length to compute the hessian at the mode of $\pi(\boldsymbol{\theta}|\mathbf{y})$

Geoadditive model III
Running the model using INLA

```
mod <-
inla(formula,family="poisson",data=Germany,E=E,
control.inla=list(h=0.01),verbose=TRUE)
```

- Defines the known variable adjusting for demographic variation E_i (*only used for Poisson likelihood*)
- Controls some parameters in the `inla` program, in this case the step length to compute the hessian at the mode of $\pi(\boldsymbol{\theta}|\mathbf{y})$

Geoadditive model IV
COmputing DIC and marginal likelihood

```
mod <-
inla(formula,family="poisson",data=Germany,E=E,
control.compute=list(dic=1,mlik=1),
control.inla=list(h=0.01),verbose=TRUE)
```

A model for survival data
Kidney infection data

patient	time	event	age	sex
1	8,16	1,1	28,28	0
2	23,13	1,0	48,48	1
3	22,18	1,1	32,32	0

Times of infection from the time of insertion of catheter on 38 kidney patients using portable dialisis equipment.
2 observation for each patient (38 patients).
Each time can be an *event* (infection) or a *censuring* (no infection)

Hazard rate and survival function

Density function:

$$y \sim f(y)$$

Survival function:

$$S(y) = 1 - F(y) = \int_y^{\infty} f(u)\ du$$

Hazard function:

$$\begin{aligned} h(y)\ dy &= \text{Prob}(y \leq Y < y + \ dy | Y > y) \\ h(y) &= \tfrac{f(t)}{S(t)} \end{aligned}$$

A model for survival data II
The model

We model the hazard function for each patient as:

$$h(y_{ij}|w_i, \mathbf{x}_{ij}) = h_0(y_{ij})\ w_i\ \exp(\mathbf{x}_{ij}^T \boldsymbol{\beta});\ i = 1, \ldots, 38;\ j = 1, 2$$

where

$h_0(\cdot)$	is the baseline hazard function
w_i	is the frailty effect associated with patient i
$\mathbf{x}_{ij}$	is the vector of observed covariates for patient i at observation j
$\boldsymbol{\beta}$	is a vector of unknown parameters

A model for survival data II
The model cont.

We assume the frailty term to be a series of i.i.d. lognormal random variables:

$$b_i = \log(w_i) \sim \mathcal{N}(0, \tau_w)$$

We divide the time axis into J intervals $I_k = (s_{k-1}, s_k]$ for $k = 1, 2, \ldots, J$ where $0 = s_0 < s_1 < \cdots < s_J < \infty$, s_J censured time and assume the baseline hazard to be piecewise constant

$$h_0(y) = \lambda_k, \text{ for } y \in I_k.$$

Moreover we assume

$$(\epsilon_1, \ldots, \epsilon_J) = \log(\lambda_1, \ldots, \lambda_J) \sim \mathrm{RW1}(10^{-4})$$

A model for survival data II
The model cont.

The hazard rate can then be written as

$$h(y_{ij}|w_i, \mathbf{x}_{ij}) = \exp(\mathbf{x}_{ij}^T \boldsymbol{\beta} + b_i + \epsilon(y_{ij})); \ \ i = 1, \ldots, 38; \ \ j = 1, 2$$

where

$$\epsilon(y_{ij}) = \epsilon_k \text{ if } y_{ij} \in I_k$$

A model for survival data III
Fitting the model using `INLA`

The `Kidney` data frame

time	event	age	sex	ID
8	1	28	0	1
16	1	28	0	1
23	1	48	1	2
13	0	48	1	2
22	1	32	0	3
28	1	32	0	3

Specifying the model:

```
formula =
time~age+sex+f(ID,param=c(0.001,0.001),initial=0.6)
```

A model for survival data III
Fitting the model using `INLA` cont.

To run the model we use the function `surv.inla`:

```
> model=surv.inla(formula,family="piecewise.constant",
n.intervals=10, data=Kidney,event=event,
control.hazard=list(initial=log(0.0001),fixed=1))
```

A model for survival data III
Fitting the model using INLA cont.

To run the model we use the function surv.inla:

```
> model=surv.inla(formula,family="piecewise.constant",
n.intervals=10, data=Kidney,event=event,
control.hazard=list(initial=log(0.0001),fixed=1))
```

A model for survival data III
Fitting the model using INLA cont.

To run the model we use the function surv.inla:

```
> model=surv.inla(formula,family="piecewise.constant",
n.intervals=10, data=Kidney,event=event,
control.hazard=list(initial=log(0.0001),fixed=1))
```

A model for survival data III
Fitting the model using INLA cont.

To run the model we use the function surv.inla:

```
> model=surv.inla(formula,family="piecewise.constant",
n.intervals=10, data=Kidney,event=event,
control.hazard=list(initial=log(0.0001),fixed=1))
```

A model for survival data III
Fitting the model using INLA cont.

To run the model we use the function surv.inla:

```
> model=surv.inla(formula,family="piecewise.constant",
n.intervals=10, data=Kidney,event=event,
control.hazard=list(initial=log(0.0001),fixed=1))
```

Other available models for survival data

- `family=exponential`
- `family=weibull`