

R-INLA: An R-package for INLA

February 27, 2023

1

Getting R-INLA

- ▶ The web page www.r-inla.org contains source-code, worked-through examples, reports and instructions for installing the package.

Getting R-INLA

- ▶ The web page www.r-inla.org contains source-code, worked-through examples, reports and instructions for installing the package.
- ▶ The R-package R-INLA works on Linux, Windows and Mac and can be installed by

```
1  install.packages("INLA", repos=c(getOption("repos"),  
2      INLA="https://inla.r-inla-download.org/R/  
3      stable"),  
    dep=TRUE)
```

Later, it can be upgraded with

```
1  update.packages("INLA", dep=TRUE)
```

Data organization

The responses and covariates are collected in a list or data frame. Assume response y , covariates x_1 and x_2 , and time index t . Then they can be organized with

```
1  # Option 1
2  data = list(y = y, x1 = x1, x2 = x2, t = t)
3
4  # Option 2
5  data = data.frame(y = y, x1 = x1, x2 = x2, t = t)
```

formula: specifying the linear predictor

The model is specified through **formula** similar to `glm`:

$$\text{formula} = y \sim x1 + x2 + f(t, \dots)$$

- ▶ `y` is the name of the response in the data
- ▶ The fixed effects are given i.i.d. Gaussian priors
- ▶ The **f function specifies random effects** (e.g. temporal, spatial, smooth effect of covariates and Besag model)
- ▶ Use **-1** if you don't want an automatic intercept

The inla function

```
1  result = inla(  
2      # Description of linear predictor  
3      formula,  
4      # Likelihood  
5      family = "gaussian",  
6      # List or data frame with response, covariates, etc.  
7      data = data,  
8  
9      ## This is all that is needed for a basic call  
10     # check what happens  
11     verbose = TRUE,  
12     # keep working files  
13     keep = TRUE,  
14  
15     # there are also some "control statements"  
16     # to customize things  
17 )
```

Likelihood functions

- ▶ "gaussian"
- ▶ "poisson"
- ▶ "nbinomial"
- ▶ "binomial"
- ▶ To see the list of available likelihood models

```
1 names(inla.models())$likelihood
```

Example: Simple linear regression

$$y_i = \underbrace{\beta_0 + \beta_1 x_i}_{\eta_i} + \epsilon_i \text{ with } \epsilon_i \sim \mathcal{N}(0, \sigma_0^2)$$

Stage 1: Gaussian likelihood

$$y_i \mid \eta_i \sim \mathcal{N}(\eta_i, \sigma_o^2)$$

Stage 2: Covariates are connected to likelihood by

$$\eta_i = \beta_0 + \beta_1 x_i$$

Stage 3: σ_o^2 : variance of observation noise

Example: Simple linear regression

```
> library(INLA)
> # Generate data
> x = sort(runif(100))
> y = 1 + 2*x + rnorm(n = 100, sd = 0.1)
> # Run inla
> formula = y ~ 1 + x
> result = inla(formula,
+               data = list(x = x, y = y),
+               family = "gaussian")
```

Get summary

```
> summary(result)
```

Call:

```
c("inla(formula = formula, family = \"gaussian\", data = list(x = x, \",  
  \" y = y)))")
```

Time used:

```
Pre = 1.05, Running = 0.12, Post = 0.0666, Total = 1.24
```

Fixed effects:

	mean	sd	0.025quant	0.5quant	0.975quant	mode	kld
(Intercept)	1.01	0.020	0.970	1.01	1.050	1.01	0
x	1.97	0.037	1.898	1.97	2.043	1.97	0

Model hyperparameters:

	mean	sd	0.025quant	0.5quant
Precision for the Gaussian observations	107.19	15.16	79.54	106.47
	0.975quant	mode		
Precision for the Gaussian observations	138.91	105.04		

Expected number of effective parameters(stdev): 2.06(0.009)

Number of equivalent replicates : 48.44

Marginal log-Likelihood: 75.05

Fixed Effects

```
> result$summary.fixed
```

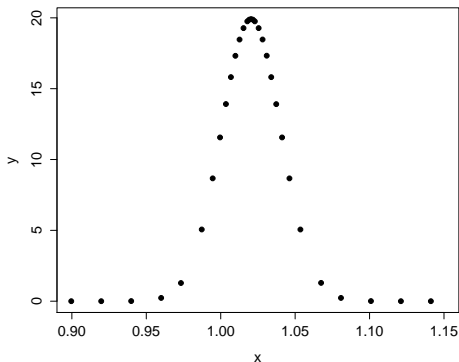
	mean	sd	0.025quant	0.5quant	0.975quant	mode
(Intercept)	1.009816	0.02030048	0.9698763	1.009815	1.049722	1.009816
x	1.970149	0.03691115	1.8975289	1.970148	2.042708	1.970149

	kld
(Intercept)	4.072744e-06
x	4.072741e-06

Marginal posterior densities

The marginal posterior densities are stored as a matrix with x- and y-values

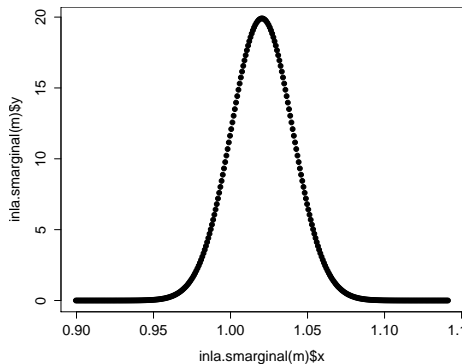
```
1 m = result$marginals.fixed[[1]]  
2 plot(m)
```



Marginal posterior densities

The rough shape can be interpolated to higher resolution

```
1 plot(inla.s marginal(m))
```



Marginal posterior densities

```
1  # Extract quantiles
2  > inla.qmarginal(0.05, m)
3  [1] 0.9818604
4
5  # Distribution function
6  > inla.pmarginal(0.975, m)
7  [1] 0.02314047
8
9  # Density function
10 > inla.dmarginal(1, m)
11 [1] 15.80794
12
13 # Generate realizations
14 > inla.rmarginal(4, m)
15 [1] 1.009122 1.013116 1.032004 1.007458
```

Organisation of the returned inla-object

```
1 > names(result)
2 [1] "names.fixed" "summary.fixed"
3 [3] "marginals.fixed" "summary.lincomb"
4 [5] "marginals.lincomb" "size.lincomb"
5 [7] "summary.lincomb.derived" "marginals.lincomb.
   derived"
6 [9] "size.lincomb.derived" "mlik"
7 [11] "cpo" "po"
8 [13] "waic" "model.random"
9 [15] "summary.random" "marginals.random"
10 [17] "size.random" "summary.linear.
   predictor"
11 [19] "marginals.linear.predictor" "summary.fitted.values"
12 [21] "marginals.fitted.values" "size.linear.predictor"
13 [23] "summary.hyperpar" "marginals.hyperpar"
14 ...
```

Add random effects

```
1 f(name, model="...", hyper=...,  
2   constr=FALSE, cyclic=FALSE, ...)
```

- ▶ name – the index of the effect (each f-function needs its own!)
- ▶ model – the type of latent model. E.g. "iid", "rw2", "ar1", "besag", and so on
- ▶ hyper – specify the prior on the hyperparameters
- ▶ constr – sum-to-zero constraint?
- ▶ cyclic – is it cyclic?
- ▶ ...

Example: Add random effect

Add an AR(1) random effect to the linear predictor.

Stage 1:

$$y_i | \eta_i \sim \mathcal{N}(\eta_i, \sigma_o^2)$$

Stage 2: Covariates and AR(1) component connected to likelihood by:

$$\eta_i = \beta_0 + \beta_1 x_i + a_i.$$

- ▶ x_i : covariate
- ▶ a_i : AR(1) process

Stage 3:

- ▶ σ_o^2 : variance of observation noise
- ▶ ρ : dependence in AR(1) process
- ▶ σ^2 : variance of the innovations in AR(1) process

Example: Add random effect

```
1  # Generate AR(1) sequence
2  t = 1:100
3  ar = rep(0,100)
4  for(i in 2:100)
5      ar[i] = 0.8*ar[i-1]+rnorm(n = 1, sd = 0.1)
6
7  # Generate data with AR(1) component
8  x = runif(100)
9  y = 1 + 2*x + ar + rnorm(n = 100, sd = 0.1)
10
11 # Run inla
12 formula = y ~ 1 + x + f(t, model="ar1")
13 result = inla(formula,
14               data = list(x = x, y = y, t = t),
15               family = "gaussian")
16
17 # Get summary
18 summary(result)
```

summary(result)

Fixed effects:

	mean	sd	0.025quant	0.5quant	0.975quant	mode	kld
(Intercept)	1.0354	0.0624	0.913	1.0344	1.1635	1.0328	0
x	2.0173	0.0459	1.927	2.0173	2.1077	2.0173	0

Random effects:

Name	Model
t	AR1 model

Model hyperparameters:

	mean	sd	0.025quant	0.5quant
Precision for the Gaussian observations	129.8753	49.6529	60.8214	120.5645
Precision for t	38.3033	13.9965	16.8866	36.4192
Rho for t	0.8031	0.0817	0.6028	0.8181

	0.975quant	mode
Precision for the Gaussian observations	251.9389	104.1904
Precision for t	70.9695	32.7097
Rho for t	0.9185	0.8463

Other choices for f-terms

For example:

- ▶ `rw1, rw2`
- ▶ `besag`
- ▶ `iid`

For a complete list see:

```
names(inla.models())$latent)
```

Changing the prior: Internal scale

- ▶ Hyperparameters are represented internally with more well-behaved transformations, e.g. precision τ and correlation ρ are internally represented as

$$\theta_1 = \log(\tau)$$

$$\theta_2 = \log\left(\frac{1 + \rho}{1 - \rho}\right)$$

- ▶ The prior must be set on the parameter in **internal scale**

Changing the prior: Code

```
1 hyper = list(prec = list(prior = "loggamma",  
2                        param = c(1, 0.1))  
3  
4 formula = y ~ f(idx, model = "iid", hyper = hyper) + ...
```

EPIL example

Seizure counts in a randomised trial of anti-convulsant therapy 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
3	2	4	0	5	0	6	25
....							
59	1	4	3	2	1	12	37

Covariates are treatment (0,1), 8-week baseline seizure counts, and age in years.

Repeated Poisson counts

$$y_{jk} \sim \text{Poisson}(\mu_{jk}); \quad j = 1, \dots, 59; \quad k = 1, \dots, 4$$

$$\begin{aligned} \log(\mu_{jk}) = & \alpha_0 + \alpha_1 \log(\text{Base}_j/4) + \alpha_2 \text{Trt}_j \\ & + \alpha_3 \text{Trt}_j \log(\text{Base}_j/4) + \alpha_4 \log(\text{Age}_j) \\ & + \alpha_5 V4 + \text{Ind}_j + \beta_{jk} \end{aligned}$$

$$\alpha_i \sim \mathcal{N}(0, \tau_\alpha) \quad \tau_\alpha \text{ known (0.001)}$$

$$\text{Ind}_j \sim \mathcal{N}(0, \tau_{\text{Ind}}) \quad \tau_{\text{Ind}} \sim \text{Gamma}(1, 0.01)$$

$$\beta_{jk} \sim \mathcal{N}(0, \tau_\beta) \quad \tau_\beta \sim \text{Gamma}(1, 0.01)$$

Here, $V4$ is an indicator variable for the 4th visit.

Prepare the dataset

```
> library(tidyverse)
> data(Epil)
> head(Epil, n = 2)
```

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

```
> my.center = function(x) (x - mean(x))
> Epil = Epil %>% mutate(
+   CTrt    = my.center(Trt),
+   ClBase4 = my.center(log(Base/4)),
+   CV4     = my.center(V4),
+   ClAge   = my.center(log(Age)))
> Epil %>% round(2) %>% head(n=2)
```

	y	Trt	Base	Age	V4	rand	Ind	CTrt	ClBase4	CV4	ClAge
1	5	0	11	31	0	1	1	-0.53	-0.76	-0.25	0.11
2	3	0	11	31	0	2	1	-0.53	-0.76	-0.25	0.11

Model specification in INLA

```
1 > data(Epil)
2 > head(Epil,n=3)
3   y Trt Base Age V4 rand Ind      CTrt      C1Base4      CV4      C1Age
4  1  5   0   11  31  0   1   1 -0.5254237 -0.75635379 -0.25  0.11420370
5  2  3   0   11  31  0   2   1 -0.5254237 -0.75635379 -0.25  0.11420370
6  3  3   0   11  31  0   3   1 -0.5254237 -0.75635379 -0.25  0.11420370
7  4  3   0   11  31  1   4   1 -0.5254237 -0.75635379  0.75  0.11420370
```

Model specification in INLA

```
1 > data(Epil)
2 > head(Epil,n=3)
3   y Trt Base Age V4 rand Ind      CTrt      ClBase4      CV4      ClAge
4 1  5   0   11  31  0   1   1 -0.5254237 -0.75635379 -0.25  0.11420370
5 2  3   0   11  31  0   2   1 -0.5254237 -0.75635379 -0.25  0.11420370
6 3  3   0   11  31  0   3   1 -0.5254237 -0.75635379 -0.25  0.11420370
7 4  3   0   11  31  1   4   1 -0.5254237 -0.75635379  0.75  0.11420370
```

```
1 > formula = y ~ ClBase4*CTrt + ClAge + CV4 +
2   f(Ind, model="iid",
3     hyper = list(prec = list(prior = "loggamma",
4                               param = c(1,0.01)))) +
5   f(rand, model="iid",
6     hyper = list(prec = list(prior = "loggamma",
7                               param = c(1,0.01))))
```

Model specification in INLA

```
1 > data(Epil)
2 > head(Epil,n=3)
3   y Trt Base Age V4 rand Ind      CTrt      ClBase4      CV4      ClAge
4  1  5   0   11  31  0   1   1 -0.5254237 -0.75635379 -0.25  0.11420370
5  2  3   0   11  31  0   2   1 -0.5254237 -0.75635379 -0.25  0.11420370
6  3  3   0   11  31  0   3   1 -0.5254237 -0.75635379 -0.25  0.11420370
7  4  3   0   11  31  1   4   1 -0.5254237 -0.75635379  0.75  0.11420370
```

```
1 > formula = y ~ ClBase4*CTrt + ClAge + CV4 +
2   f(Ind, model="iid",
3     hyper = list(prec = list(prior = "loggamma",
4                               param = c(1,0.01)))) +
5   f(rand, model="iid",
6     hyper = list(prec = list(prior = "loggamma",
7                               param = c(1,0.01))))
```

```
1 > result = inla(formula, family="poisson", data = Epil,
2   control.fixed = list(prec.intercept = 0.001,
3   prec = 0.001))
```

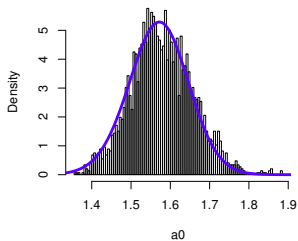
Comparing results with MCMC

- ▶ When comparing the results of R-INLA with MCMC, it is important to use the **same model**. That means, same data, same priors, same constraints on parameters, intercept included or not,

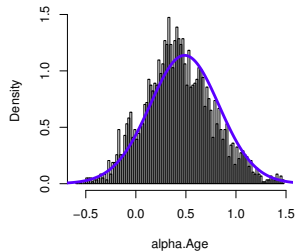
Comparing results with MCMC

- ▶ When comparing the results of R-INLA with MCMC, it is important to use the **same model**. That means, same data, same priors, same constraints on parameters, intercept included or not,
- ▶ Here we have compared the results with those obtained using JAGS via the `rjags` package

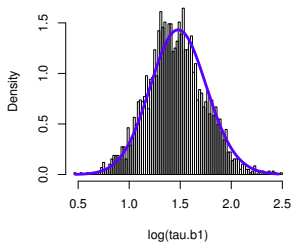
Intercept, 0.125 minutes



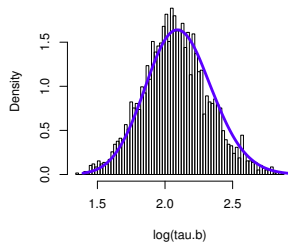
Age



log(tau.Ind)

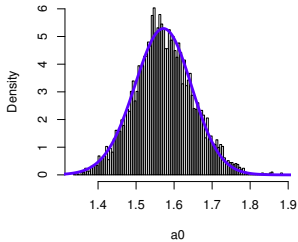


log(tau.Rand)

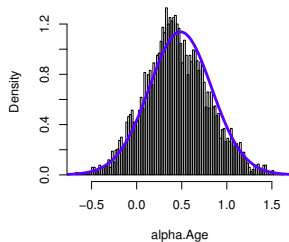


Running time of INLA < 0.5 seconds

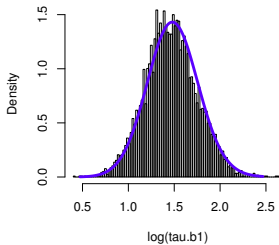
Intercept, 0.25 minutes



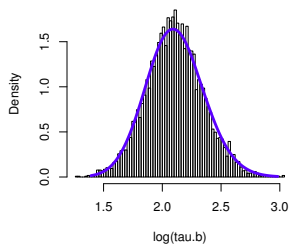
Age



log(τ_{Ind})

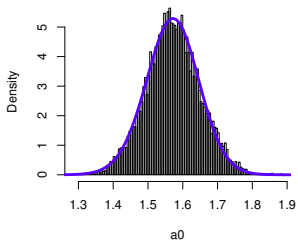


log(τ_{Rand})

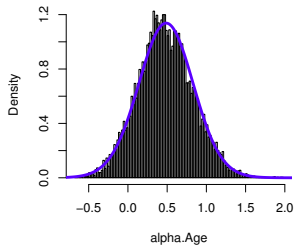


Running time of INLA < 0.5 seconds

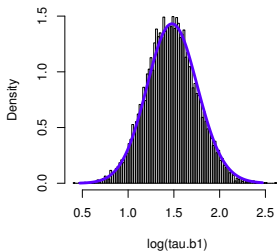
Intercept, 0.5 minutes



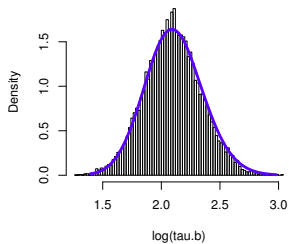
Age



log(tau.lnd)

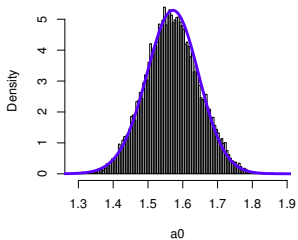


log(tau.Rand)

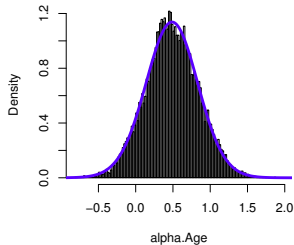


Running time of INLA < 0.5 seconds

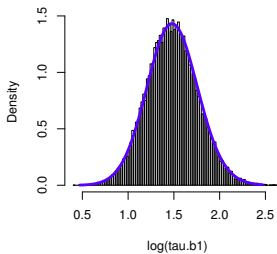
Intercept, 1 minutes



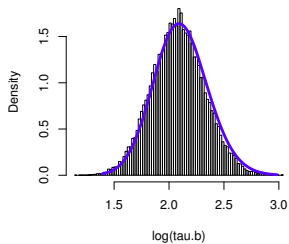
Age



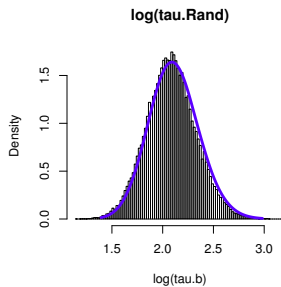
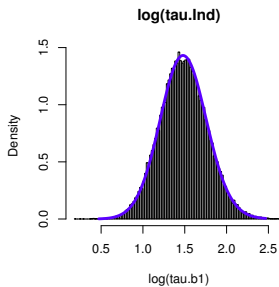
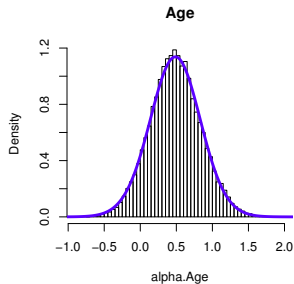
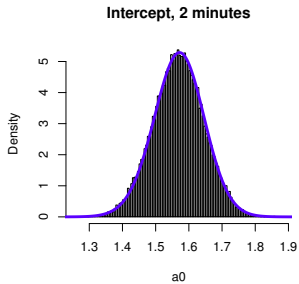
log(τ_{Ind})



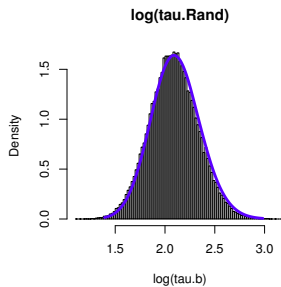
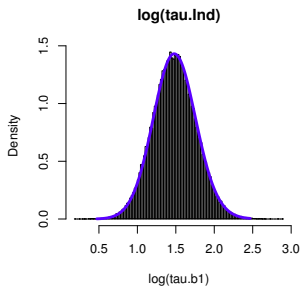
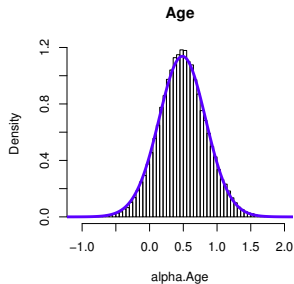
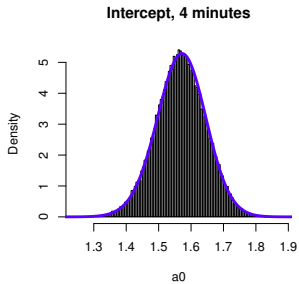
log(τ_{Rand})



Running time of INLA < 0.5 seconds

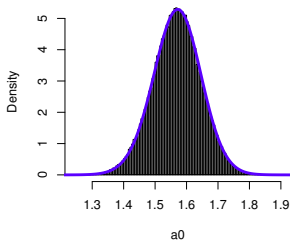


Running time of INLA < 0.5 seconds

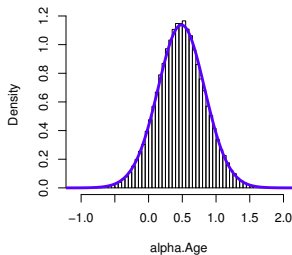


Running time of INLA < 0.5 seconds

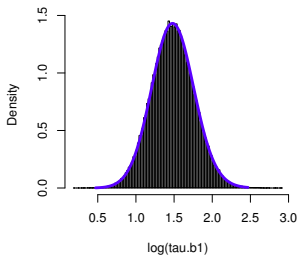
Intercept, 8 minutes



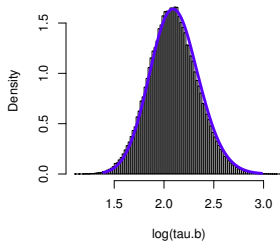
Age



log(tau.Ind)

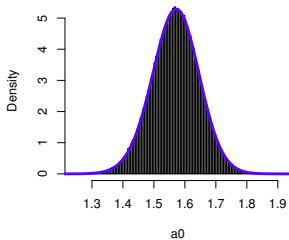


log(tau.Rand)

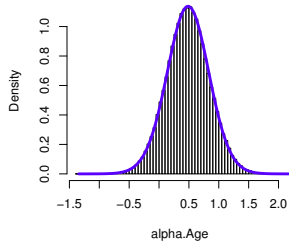


Running time of INLA < 0.5 seconds

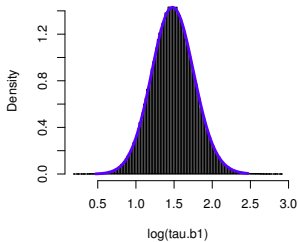
Intercept, 16 minutes



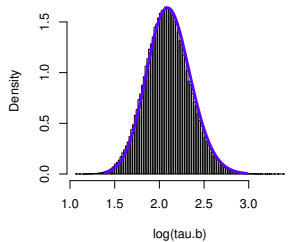
Age



log(tau.lnd)

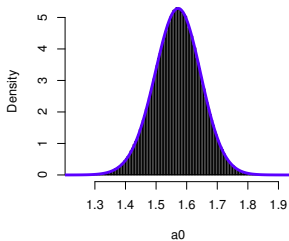


log(tau.Rand)

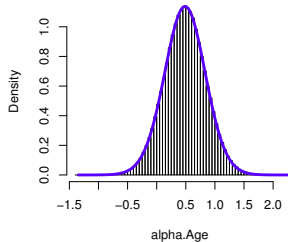


Running time of INLA < 0.5 seconds

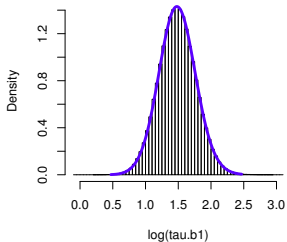
Intercept, 32 minutes



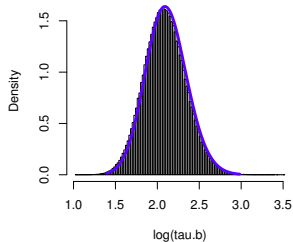
Age



log(tau.lnd)

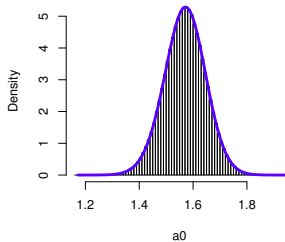


log(tau.Rand)

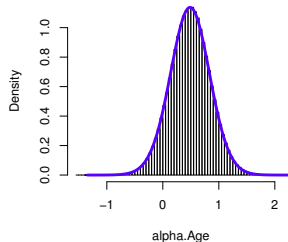


Running time of INLA < 0.5 seconds

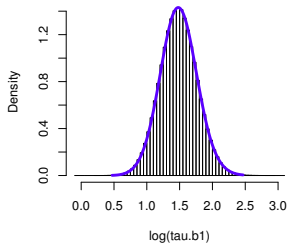
Intercept, 64 minutes



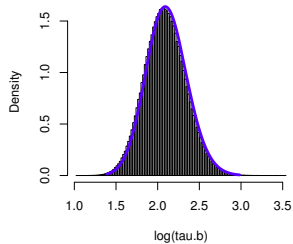
Age



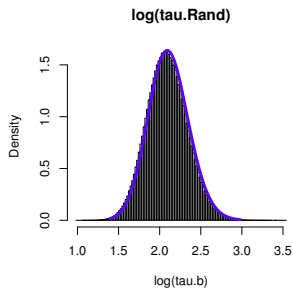
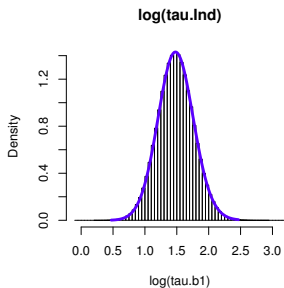
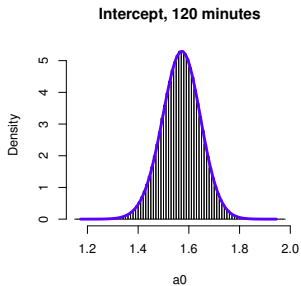
log(tau.Ind)



log(tau.Rand)



Running time of INLA < 0.5 seconds



Running time of INLA < 0.5 seconds

Control statements

`control.xxx` statements control computations

- ▶ `control.fixed`

Control statements

`control.xxx` statements control computations

- ▶ `control.fixed`
 - ▶ `prec`: Default precision for all fixed effects except the intercept. `prec.intercept`: Precision for intercept (Default: 0.0)

Control statements

`control.xxx` statements control computations

- ▶ `control.fixed`
 - ▶ `prec`: Default precision for all fixed effects except the intercept. `prec.intercept`: Precision for intercept (Default: 0.0)
- ▶ `control.predictor`

Control statements

`control.xxx` statements control computations

- ▶ `control.fixed`
 - ▶ `prec`: Default precision for all fixed effects except the intercept. `prec.intercept`: Precision for intercept (Default: 0.0)
- ▶ `control.predictor`
 - ▶ `compute`: Compute posterior marginals of linear predictors

Control statements

`control.xxx` statements control computations

- ▶ `control.fixed`
 - ▶ `prec`: Default precision for all fixed effects except the intercept. `prec.intercept`: Precision for intercept (Default: 0.0)
- ▶ `control.predictor`
 - ▶ `compute`: Compute posterior marginals of linear predictors
- ▶ `control.compute`

Control statements

`control.xxx` statements control computations

- ▶ `control.fixed`
 - ▶ `prec`: Default precision for all fixed effects except the intercept. `prec.intercept`: Precision for intercept (Default: 0.0)
- ▶ `control.predictor`
 - ▶ `compute`: Compute posterior marginals of linear predictors
- ▶ `control.compute`
 - ▶ `dic`: Compute measures of fit, here DIC, to do model comparison?

Control statements

`control.xxx` statements control computations

- ▶ `control.fixed`
 - ▶ `prec`: Default precision for all fixed effects except the intercept. `prec.intercept`: Precision for intercept (Default: 0.0)
- ▶ `control.predictor`
 - ▶ `compute`: Compute posterior marginals of linear predictors
- ▶ `control.compute`
 - ▶ `dic`: Compute measures of fit, here DIC, to do model comparison?
- ▶ There are various others as well; see help.

Model choice

There is a need to compare and choose between various models, i.e. with covariates versus without, smoothed effects versus linear, etc.

One option to this in R-INLA is the deviance information criterion (DIC):

```
1  result = inla(formula,  
2              data = data,  
3              control.compute=list(dic=TRUE))  
4  
5  # See result  
6  result$dic$dic
```

Deviance information criterion

DIC is a measure of complexity and fit. It is used to compare complex hierarchical models and is defined as:

$$\text{DIC} = \overline{D} + p_D$$

where \overline{D} is the posterior mean of the deviance (measures model fit) and p_D is the effective number of parameters (measures model complexity).

⇒ Smaller values of the DIC indicate a better trade-off between complexity and fit of the model to the data.

The DIC is based on the *deviance*

$$D(\boldsymbol{\theta}, \mathbf{x}; \mathbf{y}) = -2 \log(\pi(\mathbf{y}|\boldsymbol{\theta}, \mathbf{x}))$$

through

$$\text{DIC} = E_{\boldsymbol{\theta}, \mathbf{x}|\mathbf{y}}[D(\boldsymbol{\theta}, \mathbf{x}; \mathbf{y})] + D(\boldsymbol{\theta}^*, E_{\mathbf{x}|\mathbf{y}}[\mathbf{x}]; \mathbf{y}),$$

where $\boldsymbol{\theta}^*$ is the mode of $\pi(\boldsymbol{\theta}|\mathbf{y})$.

Useful features

There are several features that can be used to extend the standard models in R-INLA.

However, we do not have time to cover those in this course.