

Rev. Bayes' best friend: Markov or Laplace?

Comparing MCMC vs. Approximate Bayesian Inference

Krishna Padmanabhan,
Cytel Inc.

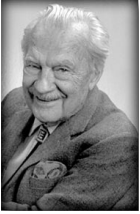
March 2022



What to expect....

- No heavy math
- Quick treatment of traditional Bayesian Inference (knowing basics of Bayesian inference will help understand the topic more easily)
- Emphasis on intuition over formulae
- Real world inspired examples using anonymized data
- Discussion of the Pros/Cons of two different Bayesian approaches

Protagonists



Metropolis, Hastings, Gelfand,
Tierney and many more...



Andrey Markov



Rev. Thomas Bayes



Pierre Simon de Laplace

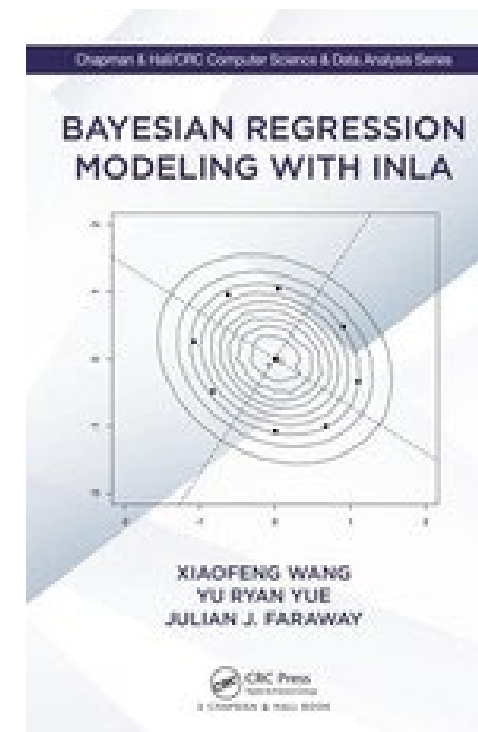
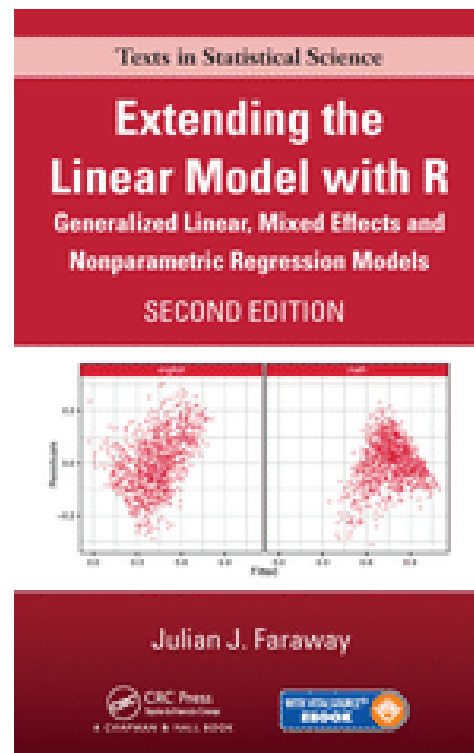
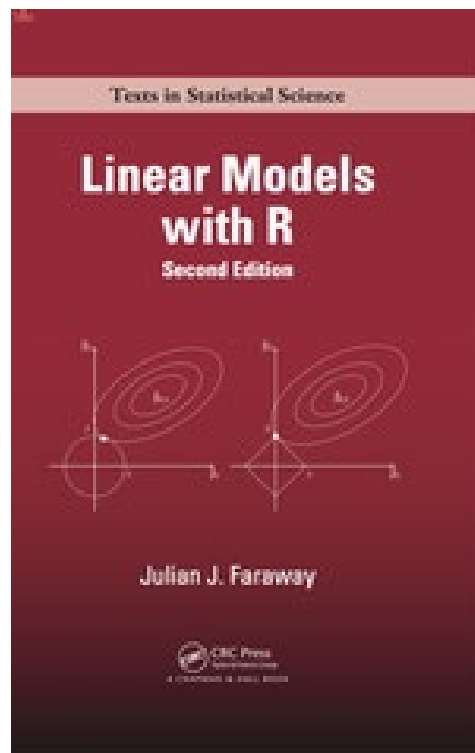


Havard Rue

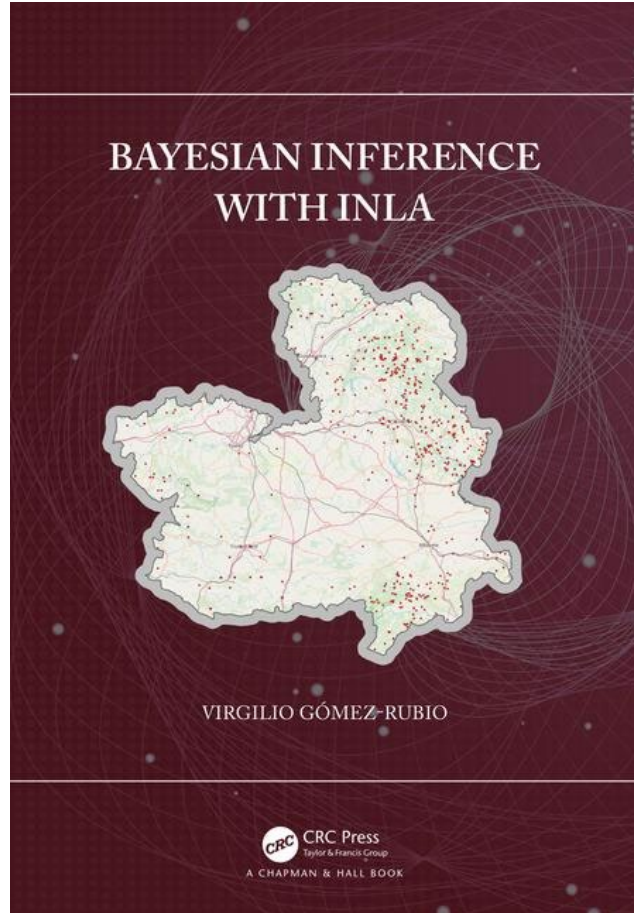
Motivation



Julian Faraway



Just as good (if not a tad better)



Full textbook seems available online:

[Bayesian inference with INLA
\(becarioprecario.bitbucket.io\)](http://becarioprecario.bitbucket.io)

Agenda

- Introduction ✓
- Brief Review of Bayesian Inference with MCMC
- (Really fast) Bayesian Inference w/ Laplace Approximations
- MCMC vs INLA: with real-world inspired clinical trial datasets
 - Continuous, Binary, Survival and Repeated Measures endpoints
- Conclusions and Discussion

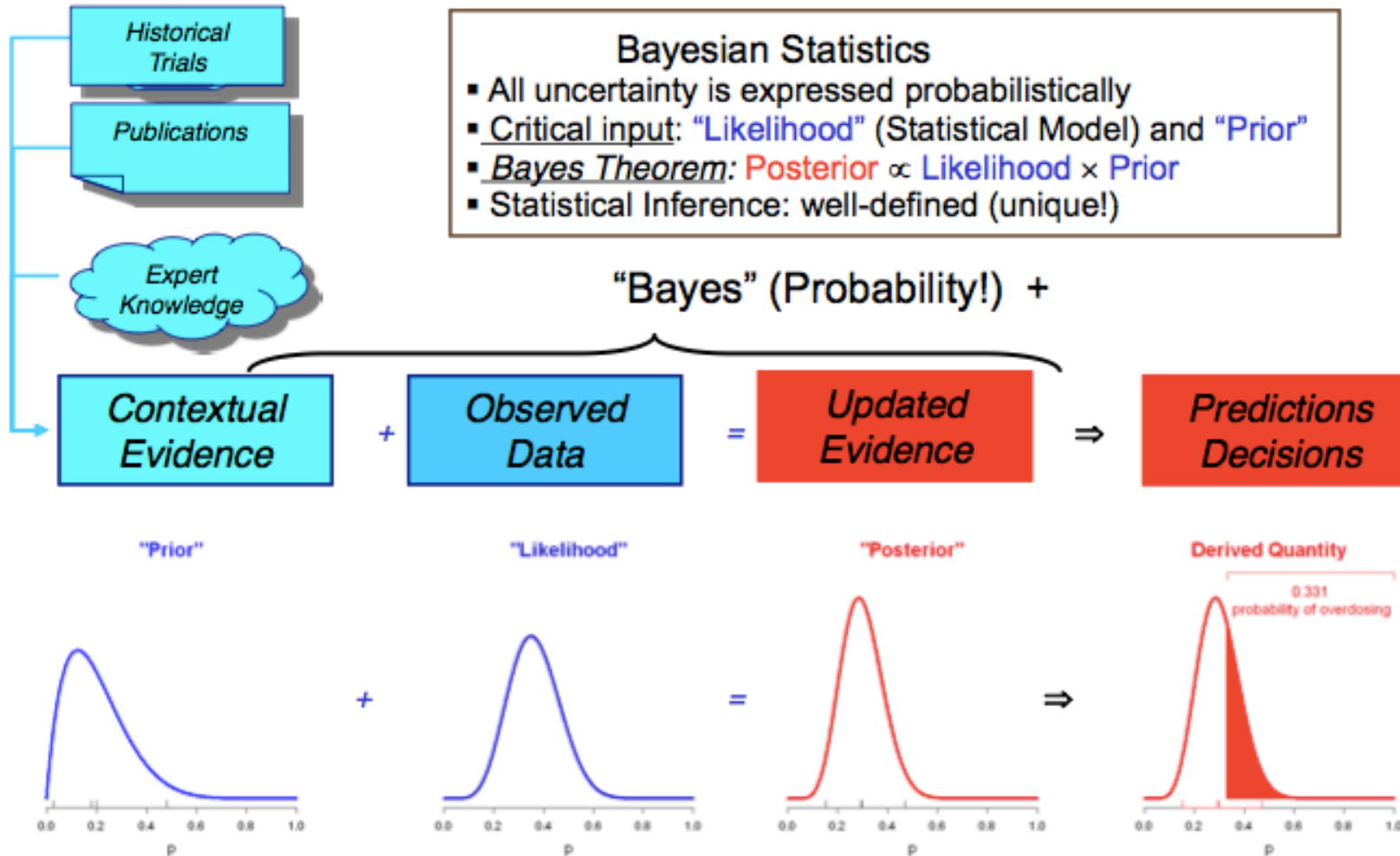
MCMC = Markov Chain Monte Carlo
INLA = Integrated Nested Laplace Approximations



Bayesian Inference and MCMC Review



The Bayesian Framework



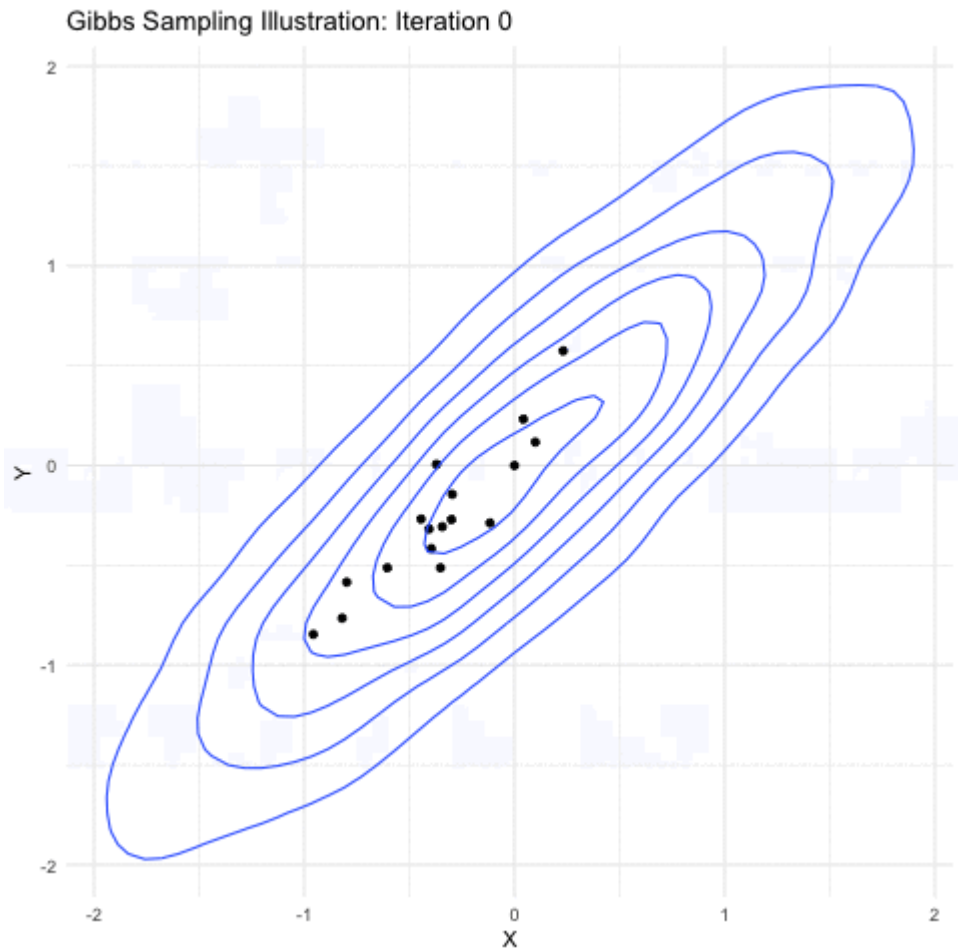
Typically, in Bayesian Inference, we need to compute the posterior and the predictive distributions:

- $P(\theta | y)$, i.e., (params | data)
 - $P(y^* | y)$ i.e., new obs.

Usually accomplished via simulation, which with large samples gives us approximately exact inferences

Markov Chain Monte Carlo (MCMC) methods allow us to compute these probabilities

Gibbs sampling: A convenient way to do MCMC



Gibbs Sampling is an MCMC method that iteratively draws an instance from the distribution of each variable, conditional on the current values of the other variables

Algorithm: Gibbs sampling

Define target density p

Define initial sample $\mathbf{x}^0 = (x_1^0, x_2^0, \dots, x_d^0)$

for $i = 1, \dots, \text{samples}$ **do**

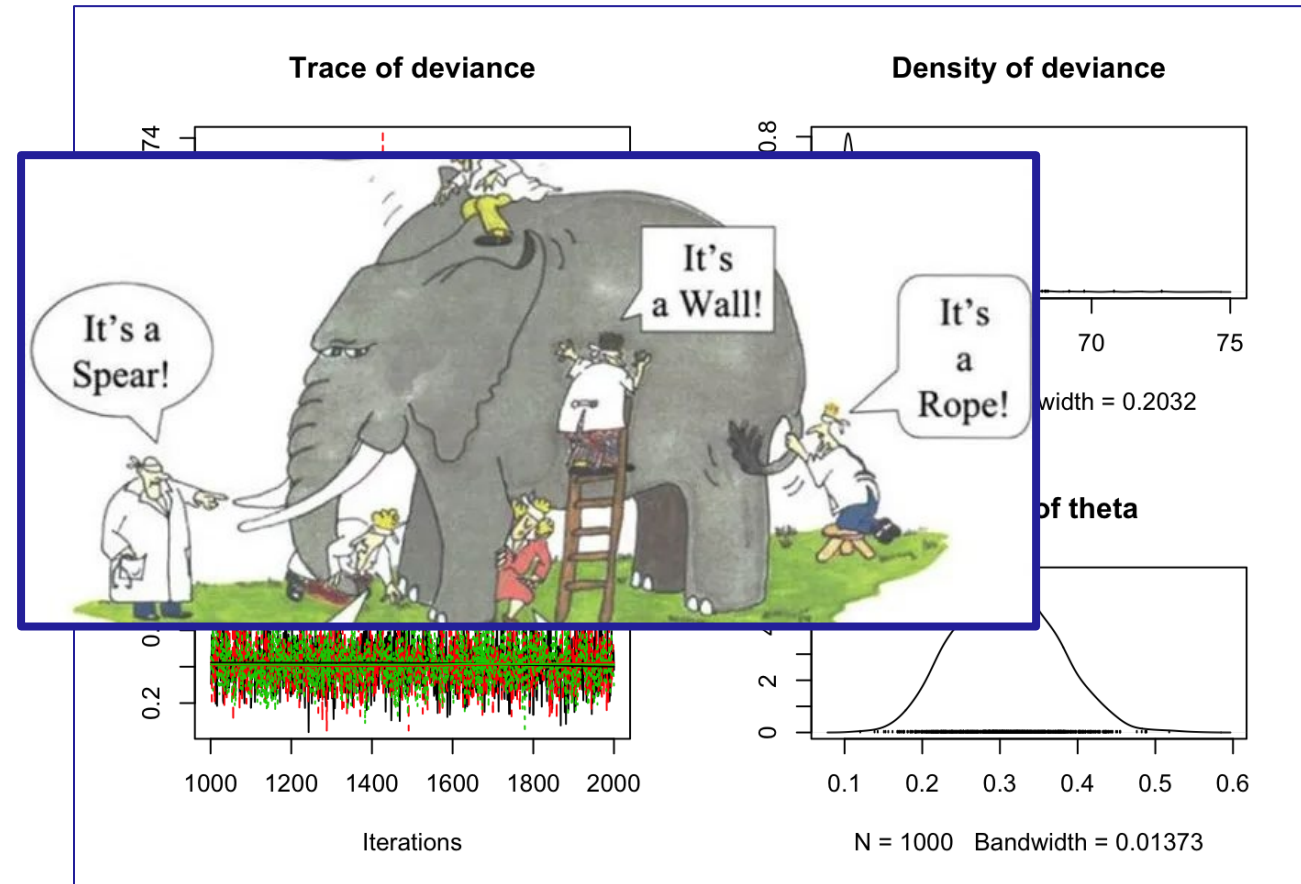
for $k = 1, \dots, d$ **do**

$x_k^i \sim p(x_k | x_1^i, \dots, x_{k-1}^i, x_{k+1}^{i-1}, \dots, x_d^{i-1})$

end

end

How we typically operationalize Bayesian Inference




A blazingly fast alternative, for a smaller class of models

If you are
function
This a

But for
v

Centre International de Rencontres Mathématiques
— Marseille - Luminy —



CIRM Bayesian computation with INLA
Håvard Rue

Summarising the ExamplesLatent Gaussian Models

Further Examples

Almost everything we care about is covered!!!

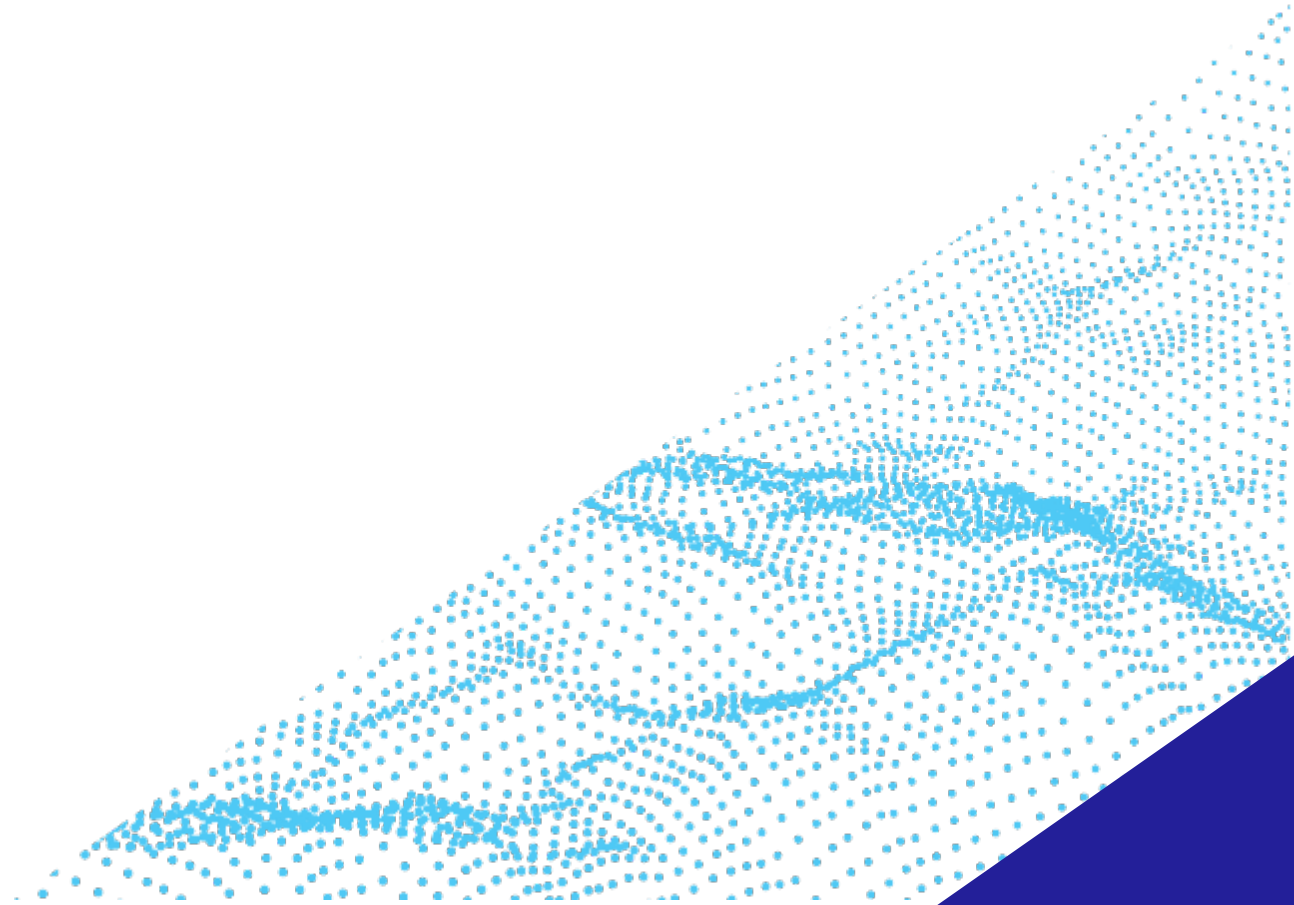
- Dynamic linear models
- Stochastic volatility
- Generalised linear (mixed) models
- Generalised additive (mixed) models
- Measurement error models
- Spline smoothing
- Semiparametric regression
- Space-varying (semiparametric) regression models
- Disease mapping
- Log-Gaussian Cox-processes
- Model-based geostatistics (*)
- Spatio-temporal models
- Survival analysis
- Joint survival/longitudinal models
- +++

Håvard Rue (haavard.rue@kaust.edu.sa)bayescomp.kaust.edu.saOct 201822 / 135



INLA Basics

(only 4 slides with Greek!)



Usual Bayesian Inference setup

- Consider a simple Bayesian Hierarchical model:
 - where y is the observed data
 - \mathbf{x} are the parameters
 - θ are the hyperparameters

$$y|\mathbf{x}, \theta_2 \sim \prod_i p(y_i|\eta_i, \theta_2)$$

$$\mathbf{x}|\theta_1 \sim p(\mathbf{x}|\theta_1)$$

$$\theta = [\theta_1, \theta_2]^T \sim p(\theta)$$

- For this Hierarchical model, the objective is usually to compute the posterior distributions:
 - $P(\mathbf{x} | y)$ – mainly
 - $P(\theta | y)$ – on occasion



What is Approximate Bayesian Inference?

- Consider the simple conditional probability definition:

$$p(b) = \frac{p(a, b)}{p(a|b)}$$

$$p(b|c) = \frac{p(a, b|c)}{p(a|b, c)}$$

- Therefore, for \mathbf{x} (parameters) and $\boldsymbol{\theta}$ (hyperparameters), we similarly have:

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})}{p(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})}$$

Likelihood

Prior

Hyperprior

UGLY

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{\boxed{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})} \boxed{p(\mathbf{x}|\boldsymbol{\theta})} \boxed{p(\boldsymbol{\theta})}}{\boxed{p(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})}}$$

$$p(\boldsymbol{\theta}|\mathbf{y}) \approx \frac{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\boxed{\tilde{p}(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})}}$$

Much more pleasant if \exists

Laplace Approximation Trick

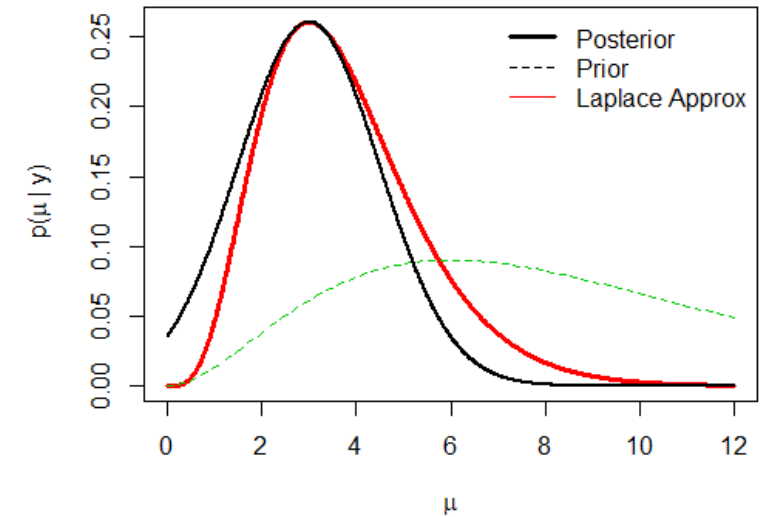
- Taylor's series expansion formula:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 + R$$

- If a global max is x_0 (think mode of a pdf) the first derivative goes to zero and the approximation becomes:

$$f(x) \approx f(x_0) - \frac{1}{2} |f''(x_0)| (x - x_0)^2$$

- LA: used to approximate integrals of the form: $\int_a^b e^{Mf(x)} dx$,
- Therefore: $\int_a^b e^{Mf(x)} dx \approx e^{Mf(x_0)} \int_a^b e^{-\frac{1}{2} M |f''(x_0)| (x-x_0)^2} dx$
- Very Gaussian-density-like integrand.
- Any PDF $f(x)$ approximated to a Normal with mean x_0



Simple Poisson-Gamma Conjugate model
R: ~20 lines of code

INLA works great for approximating the posterior,
but only for LGMs where params. are a GMRF
More ugliness? ☹ - No!!! 😊



So, what is an LGM and a GMRF?

LGM

$$\begin{aligned} \mathbf{y}|\mathbf{x}, \theta_2 &\sim \prod_i p(y_i|\eta_i, \theta_2) \\ \mathbf{x}|\theta_1 &\sim p(\mathbf{x}|\theta_1) = \mathcal{N}(0, \Sigma) \\ \theta &= [\theta_1, \theta_2]^T \sim p(\theta) \end{aligned}$$

- This model is an LGM *iff* we assume that these \mathbf{x} parameters have a joint Gaussian distribution.
- This can be achieved by putting normal priors on each parameter):

$$\mathbf{x} \sim \mathcal{N}(0, \Sigma)$$

GMRF

- Any LGM is a GMRF if elements of \mathbf{x} are conditionally independent.
- Usually, this is an acceptable assumption for our models

$$x_i \perp x_j \Leftrightarrow \Sigma_{ij} = 0$$

$$x_i \perp x_j | \mathbf{x}_{-ij} \Leftrightarrow \mathbf{Q}_{ij} = 0$$

- Havard Rue and team proved that if x_i, x_j are conditionally independent, then the ij^{th} element in the precision matrix is zero.
- Therefore, GMRF implies a very sparse precision matrix
 - When paired with a Cholesky decomposition, enables extremely fast computations of matrix inverses.

Posterior of $x_j|y$: Some computational details

- We can obtain $P(\text{all } x_j | \theta, y)$, using the Laplace approximation trick, but usually the fit is poor due to the conditional distribution assumptions being too strong.
- We can alternatively approximate $P(x_j | \mathbf{X}_{-j}, \theta, y)$, i.e., each element in the \mathbf{x} vector. Much better approximation since $P(\mathbf{X}_{-j} | x_j, \theta, y)$ are usually close to Gaussian, but takes longer when $\dim(\mathbf{x})$ is large.

$$p(x_j | \theta, \mathbf{y}) \propto \frac{p(\mathbf{x}, \theta | \mathbf{y})}{p(\mathbf{x}_{-j} | x_j, \theta, \mathbf{y})} \propto \frac{p(\theta)p(\mathbf{x} | \theta)p(\mathbf{y} | \mathbf{x})}{p(\mathbf{x}_{-j} | x_j, \theta, \mathbf{y})}$$

- Compromise approach: Use up to the 3rd order term in Taylor's series expansion of both the numerator and denominator of $p(x_j | \theta, \mathbf{y})$. Good enough approximation and faster compute times. Default in R package

Bringing it together: Why is it called INLA?

- LA = Because it uses Laplace approximations
- N = Nested because the Laplace approximations are nested within one another
 - You need to approximate $P(x|\theta, y)$ in order to compute $P(\theta|y)$
- I = Integrated, because numerical integration methods are then used compute the posterior $P(x|y)$ from $P(x|\theta, y)$ obtained in the previous step

$$\tilde{p}(x_j|\mathbf{y}) \approx \sum_{h=1}^H \tilde{p}(x_j|\theta_h^*, \mathbf{y}) \tilde{p}(\theta_h^*|\mathbf{y}) \Delta_h$$

R-INLA: A convenient implementation of INLA in R



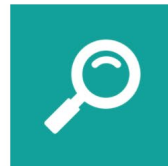
<https://www.r-inla.org/>

To install the INLA-package in R, you have to manually add the r-inla repository as they are not on CRAN.

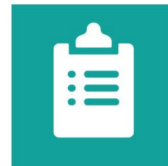
To install the stable/testing version, do one of

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

INLA



[What is INLA?](#)



[Documentation](#)



[Examples & Tutorials](#)



[Download & Install](#)



[Learn More](#)



[Team](#)



[News](#)



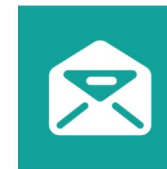
[Discussion Group](#)



[FAQ](#)



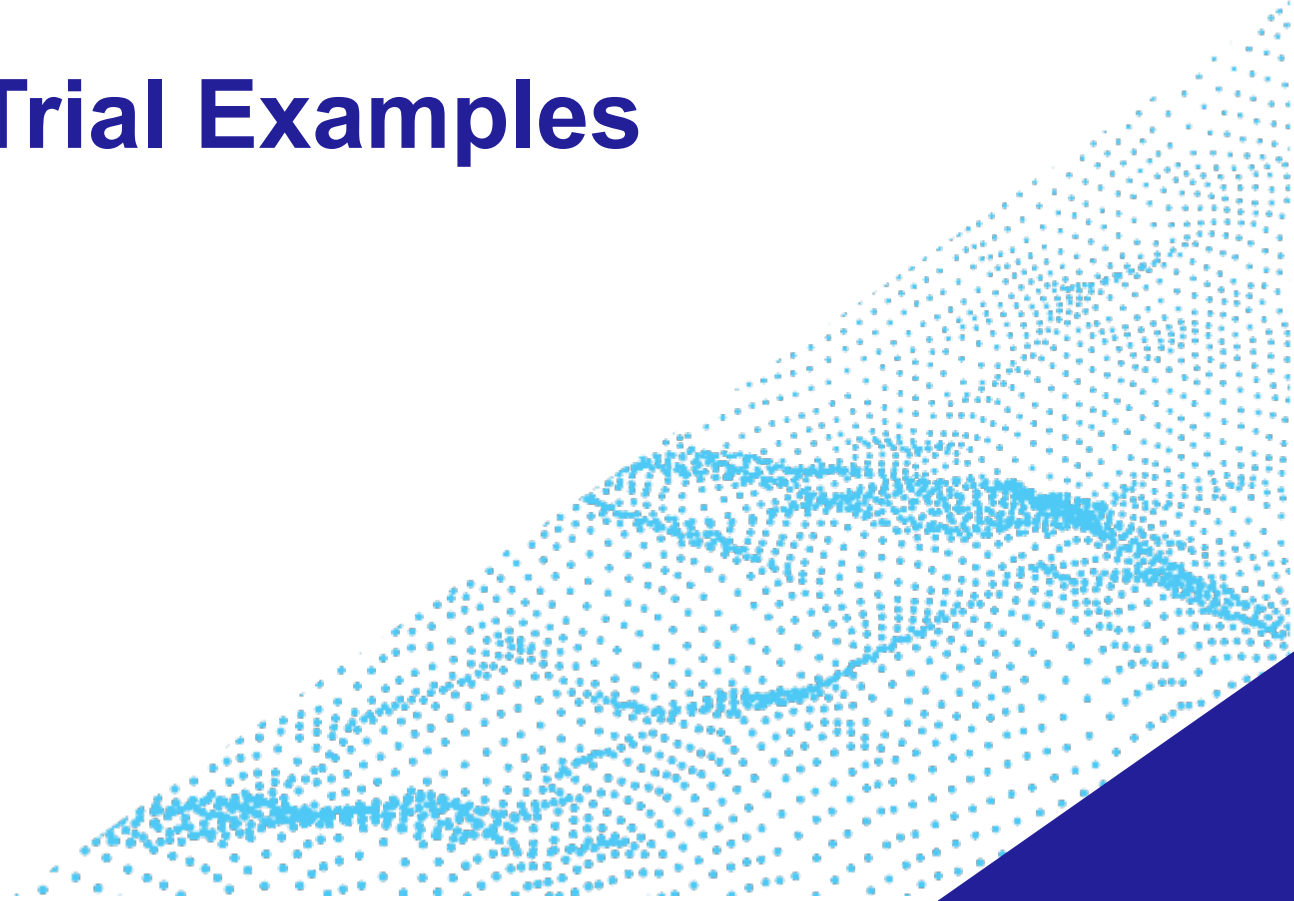
[Related Projects](#)



[Contact us](#)



MCMC vs. INLA: Clinical Trial Examples

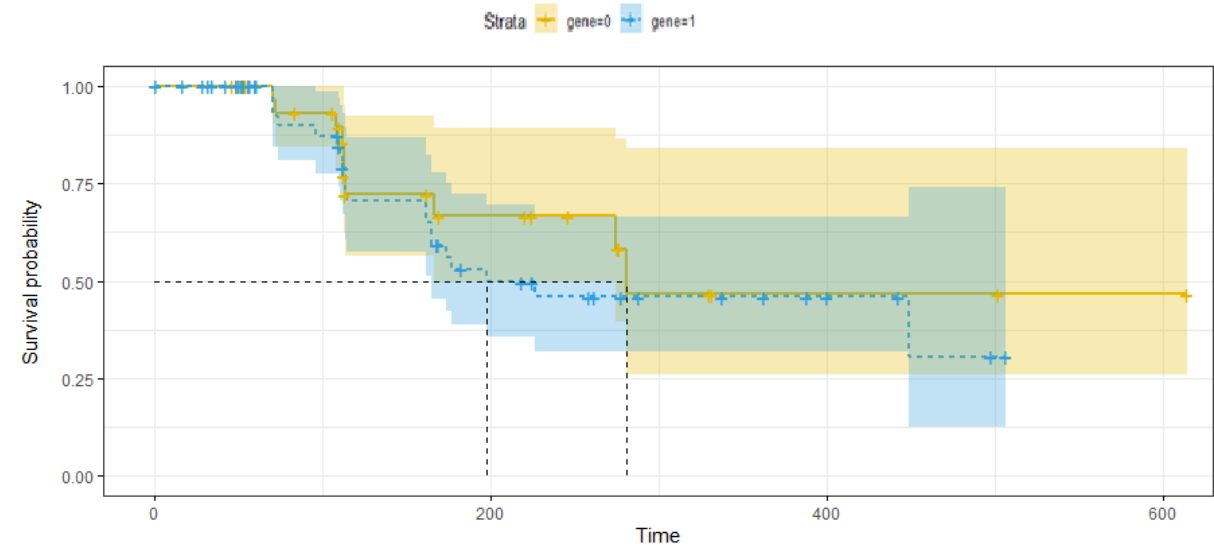


Data Sources

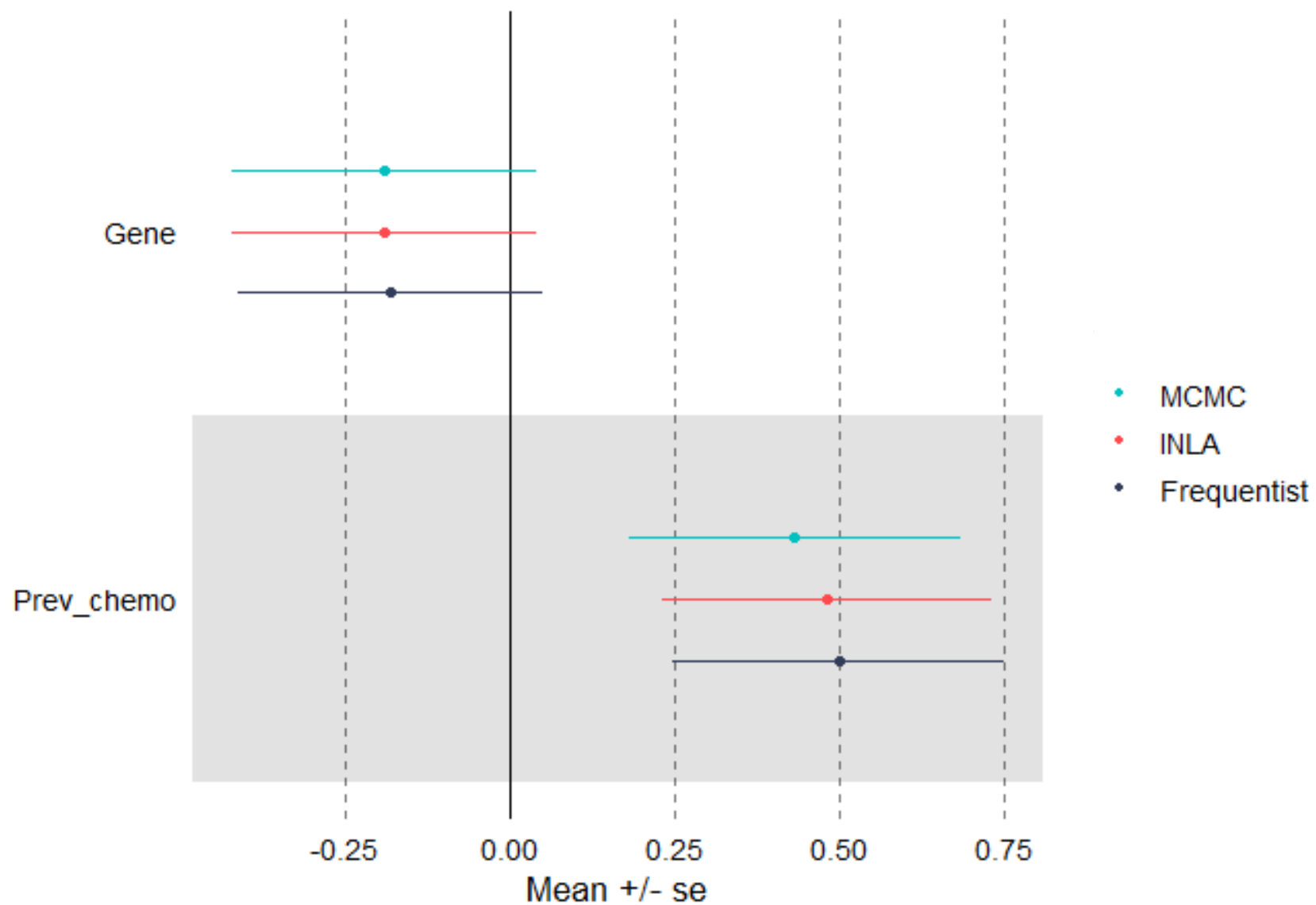
- Inspired by real world clinical datasets, but not wholly adopted from any actual trial
- 1 for each type of endpoint
 - Binary
 - Continuous
 - Repeated Measures
 - Time to Event
- Across a variety of therapeutic areas
- Examples here are for larger scale clinical trials, although methodology isn't constrained as such

Example 1: Survival Endpoint, CoxPH Model

- **Setting:** Stage 2 Cancer (solid tumor) patients with stratified enrollment based on a specific mutation (N~ 350-400)
- Study still blinded, but current hypothesis of interest is whether the gene offers a protective effect
- Fitting a CoxPH model with:
 - Bsl. Hazards = mix of 4 piecewise constants
 - Covariates = # of Prev. Chemo, Gene status
 - $h(t|h_0, \boldsymbol{\beta}) = h_0(t) \exp(\boldsymbol{\beta}_1 \text{ Mut} + \boldsymbol{\beta}_2 \text{ Chemo})$
- Comparing JAGS, INLA and Frequentist approaches



Comparing MCMC vs INLA vs Frequentist estimates



Code: RJAGS vs R-INLA

```
## RJAGS code (65 lines)
library(RJAGS)
data <- read.csv("data.csv")
modelCode <- "
data {
  int N;
  int time[N];
  int event[N];
  int gene[N];
  int prev_chemo[N];
}
parameters {
  real mu;
  real sigma;
  real alpha;
  real beta;
  real gamma;
  real delta;
  real epsilon;
  real zeta;
  real eta;
  real theta;
  real iota;
  real kappa;
  real lambda;
  real mu;
  real nu;
  real xi;
  real omicron;
  real pi;
  real rho;
  real sigma;
  real tau;
  real upsilon;
  real phi;
  real chi;
  real psi;
  real omega;
  real v;
  real w;
  real x;
  real y;
  real z;
}
model {
  mu ~ dlnorm(0, 1);
  sigma ~ dlnorm(0, 1);
  alpha ~ dlnorm(0, 1);
  beta ~ dlnorm(0, 1);
  gamma ~ dlnorm(0, 1);
  delta ~ dlnorm(0, 1);
  epsilon ~ dlnorm(0, 1);
  zeta ~ dlnorm(0, 1);
  eta ~ dlnorm(0, 1);
  theta ~ dlnorm(0, 1);
  iota ~ dlnorm(0, 1);
  kappa ~ dlnorm(0, 1);
  lambda ~ dlnorm(0, 1);
  mu ~ dlnorm(0, 1);
  nu ~ dlnorm(0, 1);
  xi ~ dlnorm(0, 1);
  omicron ~ dlnorm(0, 1);
  pi ~ dlnorm(0, 1);
  rho ~ dlnorm(0, 1);
  sigma ~ dlnorm(0, 1);
  tau ~ dlnorm(0, 1);
  upsilon ~ dlnorm(0, 1);
  phi ~ dlnorm(0, 1);
  chi ~ dlnorm(0, 1);
  psi ~ dlnorm(0, 1);
  omega ~ dlnorm(0, 1);
  v ~ dlnorm(0, 1);
  w ~ dlnorm(0, 1);
  x ~ dlnorm(0, 1);
  y ~ dlnorm(0, 1);
  z ~ dlnorm(0, 1);
}
```

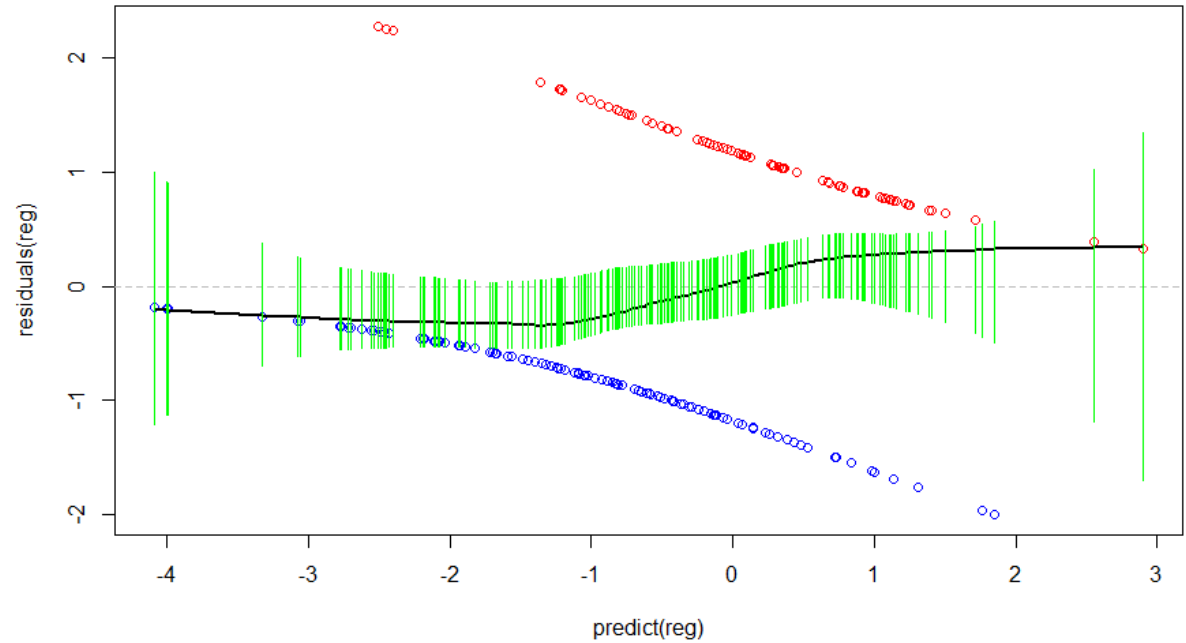
```
sinla.dat <- inla.surv(dat$time, dat$event)
coxinla <- inla(sinla.dat ~ 1 + gene + prev_chemo, data = dat,
  family = "coxph",
  control.hazard = list(hyper = list(prec = list(param = c(0.001, 0.001))))
summary(coxinla)
```

MCMC: 65 lines of R code (187 secs.)
vs.
INLA: 5 lines (1.1 secs.)



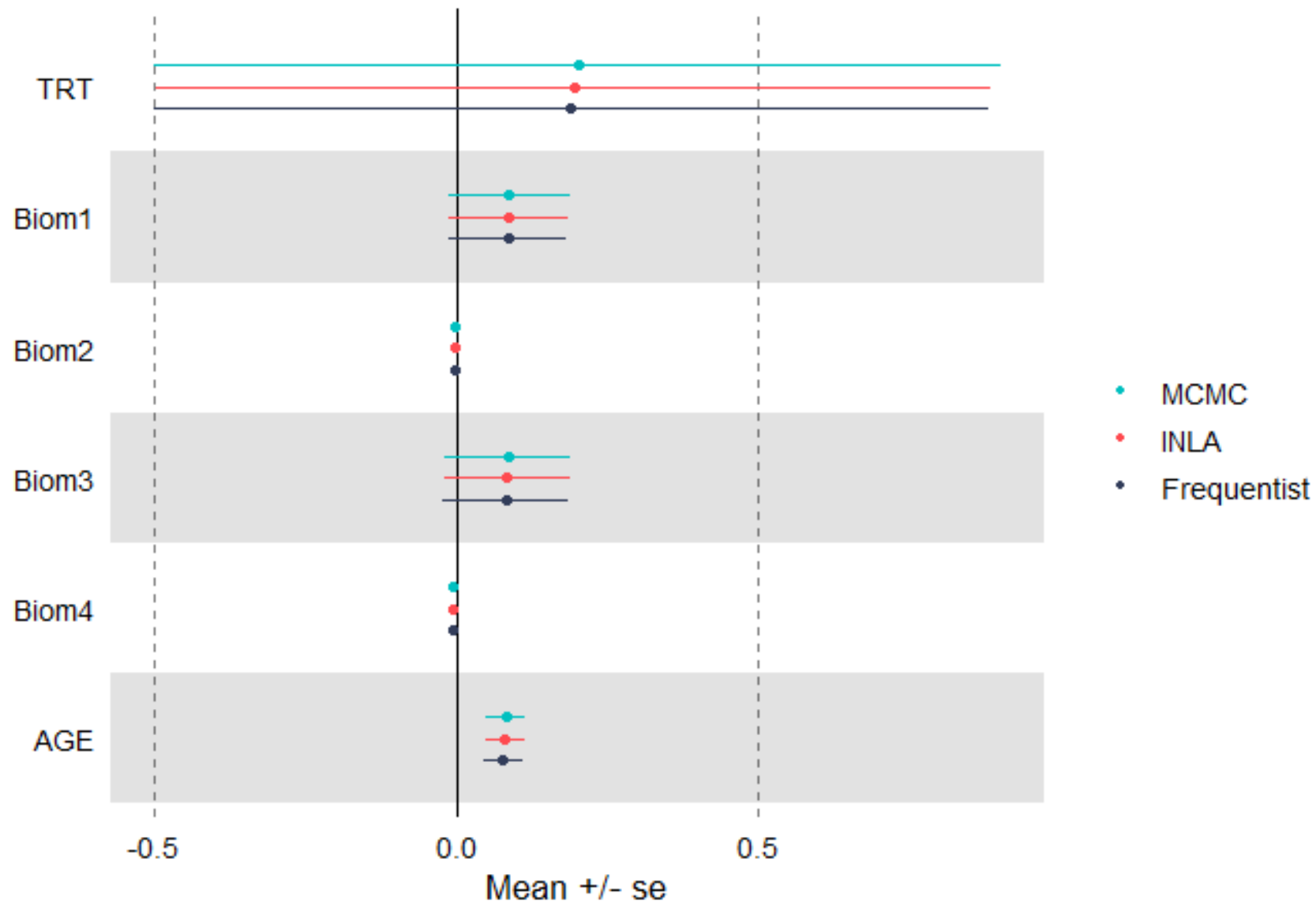
Example 2: Binary Endpoint, Logistic Regression

- **Setting:** Infectious disease study with severe cases
- Survival Status at Day 15 was the primary outcome of interest (Analyzed as binary outcome in SAP)
- N = 200 patients approx.
- Fit a Logistic Regression model with:
 - TRT, AGE and 3 other biomarkers as explanatory variables
- Comparing JAGS, INLA and Frequentist approaches



Deviance Residual Plot

Comparing MCMC vs INLA vs Frequentist estimates



Code: RJAGS vs R-INLA

```
library(RJAGS)
data = read.csv("data.csv")
N = nrow(data)
J = ncol(data)
Y = data[,1:J]
X = data[,J+1:J+J]
mod1 <- jags.model(text = "
  for (i in 1:N) {
    for (j in 1:J) {
      Y[i,j] ~ dpois(mu[i,j])
      mu[i,j] ~ dlnk(
        TRT[i] + AGE[i,j] + BIOM1[i] + BIOM2[i] + BIOM3[i] + BIOM4[i]
      )
    }
  }
", data = list(X = X, Y = Y), verbose = 0)
summarize <- function(x) {
  for (j in 1:J) {
    for (i in 1:N) {
      print(paste("mu", i, j, " = ", x[i,j]))
    }
  }
}
summary(mod1)
```

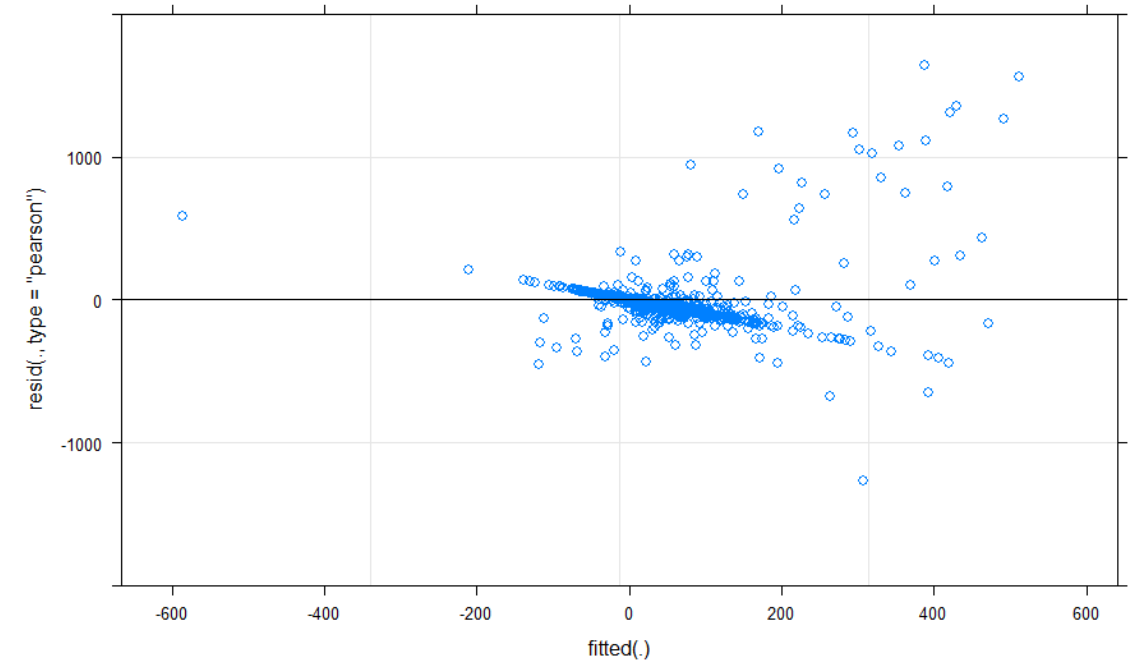
```
library(INLA)
mod3<-inla(Surv ~ TRT + AGE + BIOM1 + BIOM2 + BIOM3 + BIOM4,
  data=dat, family = "binomial", Ntrials = 1,control.compute = list(waic=T),
  num.threads = 2)

summary(mod3)
```

MCMC: 36 lines of R code (239 secs.)
vs.
INLA: 6 lines (0.9 secs.)

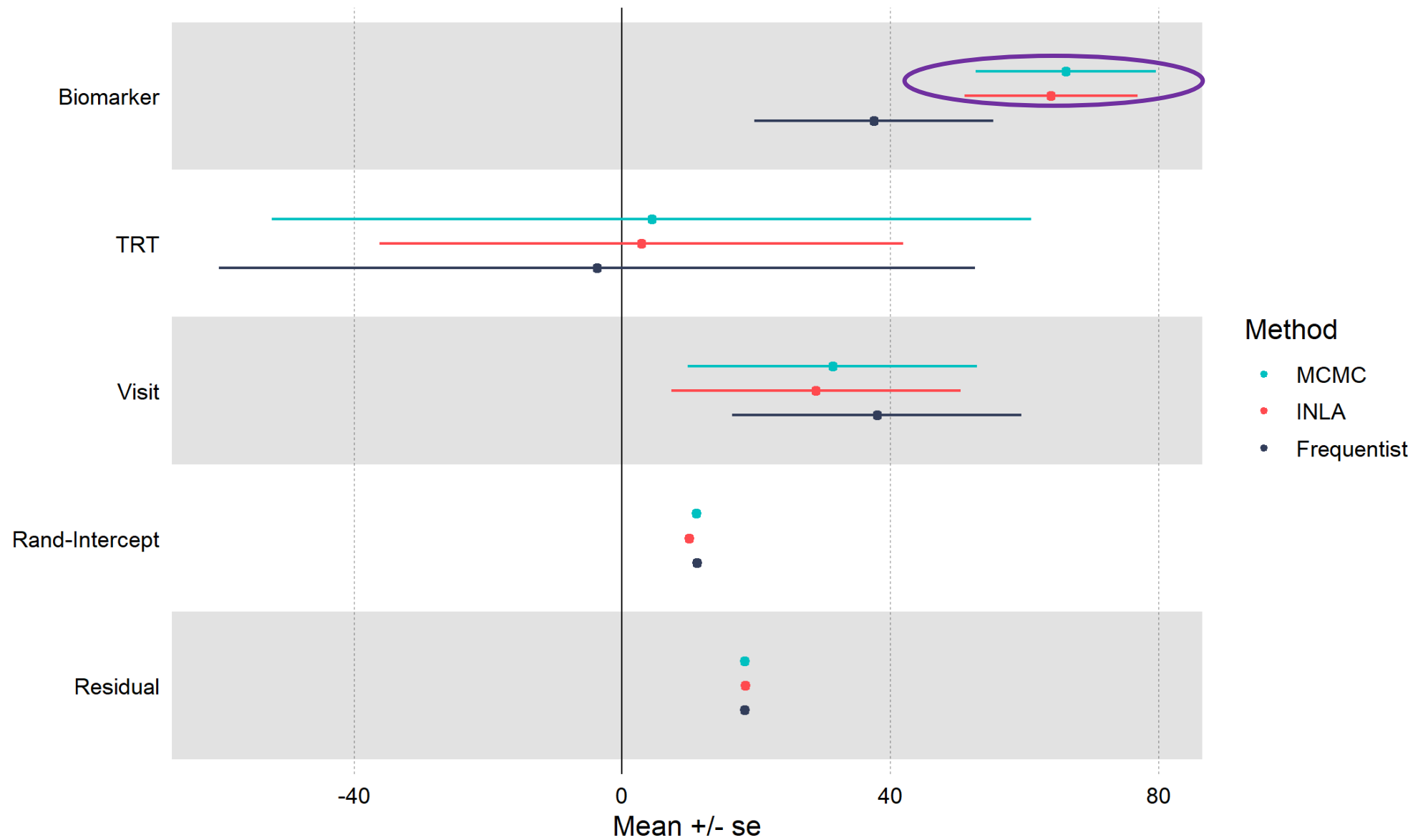
Example 3: Repeated Measures, Linear Mixed Effects Model

- **Setting:** Nephrology Study
- eGFR like variable is the endpoint
- N = ~100 patients, 4 visits per patient
- Fit a Linear Mixed Effects Model with:
 - TRT, TIME and a Biomarker as fixed effects
 - Random Intercept (only) + Slope for each patient
- Both Bayesian models converged, but the *lmer4* Random slopes model had a singular fit due to insufficient DoF.
- Comparing JAGS, INLA and Frequentist approaches for random intercept model. Informative Prior for Biomarker.



Model 1: Random Intercept only

Comparing MCMC vs INLA vs Frequentist estimates



Code: RJAGS vs R-INLA

```
dat<-dat_inla
#Random Intercepts model with default priors
formula <- diff ~ Biomarker + visit + trt + f(subj, model="iid")
inla_lme <- inla(formula, family="gaussian", data=dat)
summary(inla_lme)
```

MCMC: 58 lines of R code (153 secs.)
vs.
INLA: 5 lines (2.3 secs).

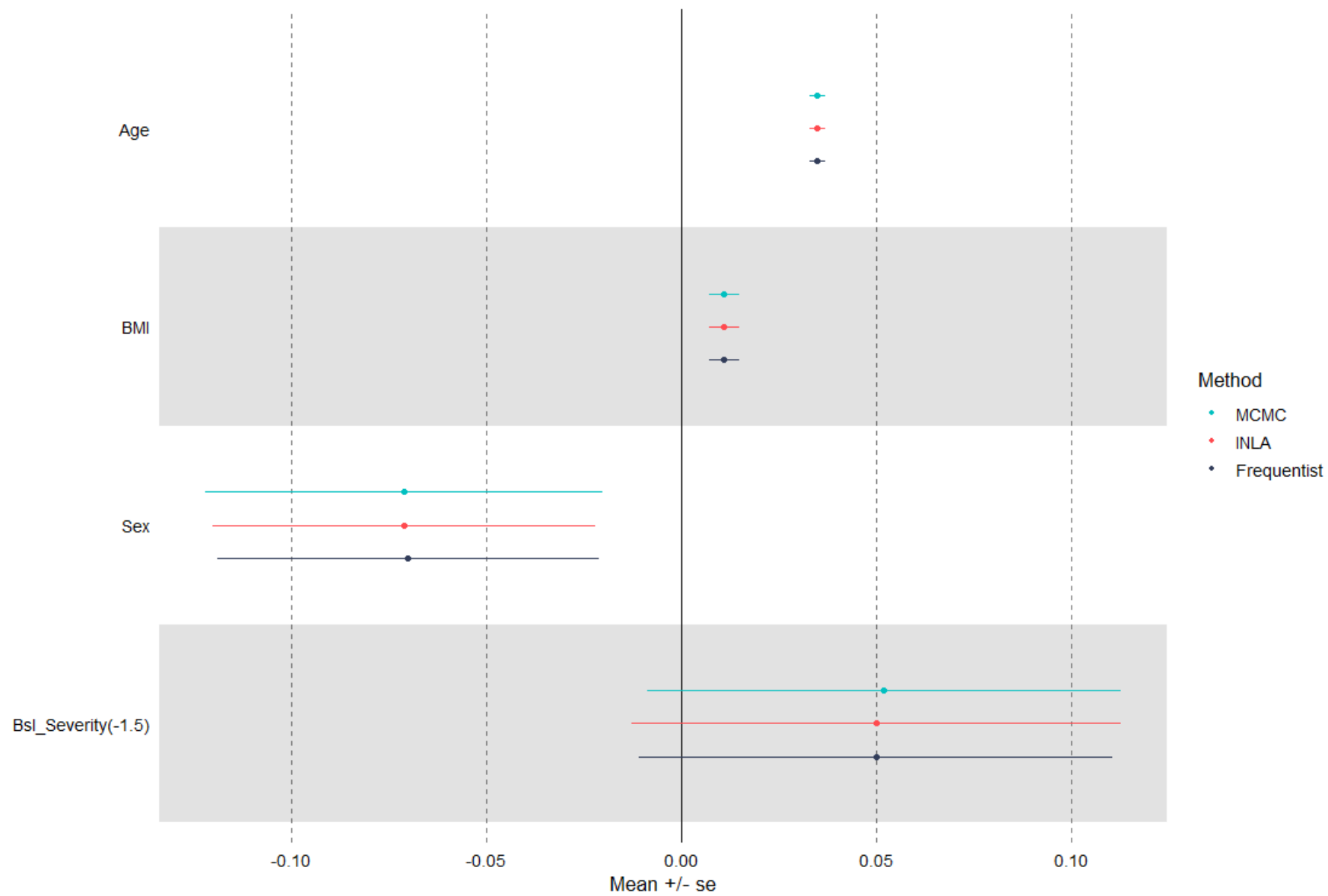
Example 4: Continuous Endpoint, ANCOVA Example

- **Setting:** Exploratory Biomarker Study in ACS
- A prognostic biomarker (continuous) was chosen as the outcome.
 - log transformed data for un-skewing
- N = ~1100 samples. 1 obs. per subject.
- Fit an ANCOVA with:
 - GENDER as factor and BMI, AGE, Baseline value as covariates
- Comparing JAGS, INLA and Frequentist approaches
- 1 interim analysis with RAR – not in slides



Adj R²= 0.745

Comparing MCMC vs INLA vs Frequentist estimates



Code: RJAGS vs R-INLA

```
library(INLA)
formula <- logcharges~age+bmi+as.factor(sex)+as.factor(smoker)
imod <- inla(formula, family="gaussian", data=dat)
summary(imod)
```

MCMC: 38 lines of R code (36 secs.)
vs.
INLA: 4 lines (1.0 secs.)

Comparison of Compute Times: RJAGS vs. R-INLA

Example	MCMC (sec.)	INLA (sec.)
Survival (BC)	187	1.1
Binary (Inf. Disease)	238	0.9
LME (Nephrology)	153	2.7
Continuous (ACS)	36.2	1.0
Survival2 (CV) (N=3000+)	>49K* (13.7 Hours)	27.35
LME2 (Lipid) (N=7000+)	>250K* (~3 days)	396.2

MCMC:
50K iterations,
3 chains

INLA:
Standard R-INLA Simplified
Laplace Approx.

Would this have contributed
to some of the time
differentials?

* "Killed" R processes at 75% and 33% completion, respectively. Total runtime is a projection.

Pros and Cons: INLA vs MCMC

MCMC	INLA
Simulation-based	Approximation-based
Slow	Fast (extremely)
Works on any model, if you can write the likelihood	LGMs only, but covers most models that we need
Wide Range of Priors	Limited Priors, but seem sufficient (TBC)
Implementation is very transparent, even if verbose coding needed	Somewhat obfuscated, but easy to learn and code.
Can obtain joint posterior distributions	Marginal distributions only (Aug 2020 Update: Joint Posteriors in R-INLA)
More software options, Incumbent advantage in terms of resources	R-INLA only(?); Newcomer, limited online forums - but growing

Conclusions

- INLA is an extremely fast alternative for Bayesian inference for certain classes of models.
 - LMs, GLMs, LMEs, GLMMs, GAMs, TTE etc. are all covered
- INLA learning curve is gentler than expected.
- Black box nature of INLA takes a while to get used to
 - Learn with INLA, confirm with MCMC
- Extremely useful tool in design optimization stage

Which one would you prefer?

INLA



MCMC

Personally, will continue to use MCMC as the primary workhorse for final models – for now.
But will use INLA extensively during model development and tuning.

References

- Håvard Rue Short Course: Bayesian computation with INLA: <https://www.youtube.com/watch?v=a8QvxCjWieg>
- Bayesian Regression Modeling with INLA: Xiaofeng Wang, Yu Yue Ryan, Julian J. Faraway 2018. CRC press / Routledge
- Rue H, Martino S, Chopin N. 2009. "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations." Journal of the Royal Statistical Society: Series B (Statistical Methodology) 71(2): 319-392.
- Bayesian Computing with INLA: A Review; Annual Review of Statistics and Its Application Vol. 4:395-421 (Volume publication date March 2017) <https://doi.org/10.1146/annurev-statistics-060116-054045>
- A gentle INLA tutorial (precision-analytics) Kathryn Morrison, 2017

Thank you.