Présentation des logiciels d'inférence bayésienne

Matthieu Authier, Emily Walker, Sophie Ancelet

19 mars 2019, Biobayes

Plan

- Introduction à l'inférence bayésienne et aux MCMC
- Outils et logiciels
- Prise en main WinBUGS / JAGS / Stan
- Comparaison WinBUGS / JAGS / Stan

Introduction à l'inférence bayésienne et aux MCMC

Loi a posteriori (cf. cours Eric) :

$$[\theta|y] \propto [y|\theta] \times [\theta]$$

- Ce que l'on cherche à quantifier en bayésien :
  - la loi a posteriori

$$f(\theta \mid Y) = \frac{f(Y \mid \theta)\pi(\theta)}{\int_\Theta f(Y \mid \alpha)\pi(\alpha)d\alpha}$$

  - et ses caractéristiques : moments a posteriori, maximum a posteriori, quantiles a posteriori, intervalles de crédibilité...

Introduction à l'inférence bayésienne et aux MCMC

- Ce que l'on cherche à quantifier en bayésien :
  - la loi a posteriori

$$f(\theta \mid Y) = \frac{f(Y \mid \theta)\pi(\theta)}{\int_\Theta f(Y \mid \alpha)\pi(\alpha)d\alpha}$$

  - et ses caractéristiques : moments a posteriori, maximum a posteriori, quantiles a posteriori, intervalles de crédibilité...

- Mais la loi a posteriori peut être difficilement calculable
  - Modèle avec nombreux paramètres et variables latentes :

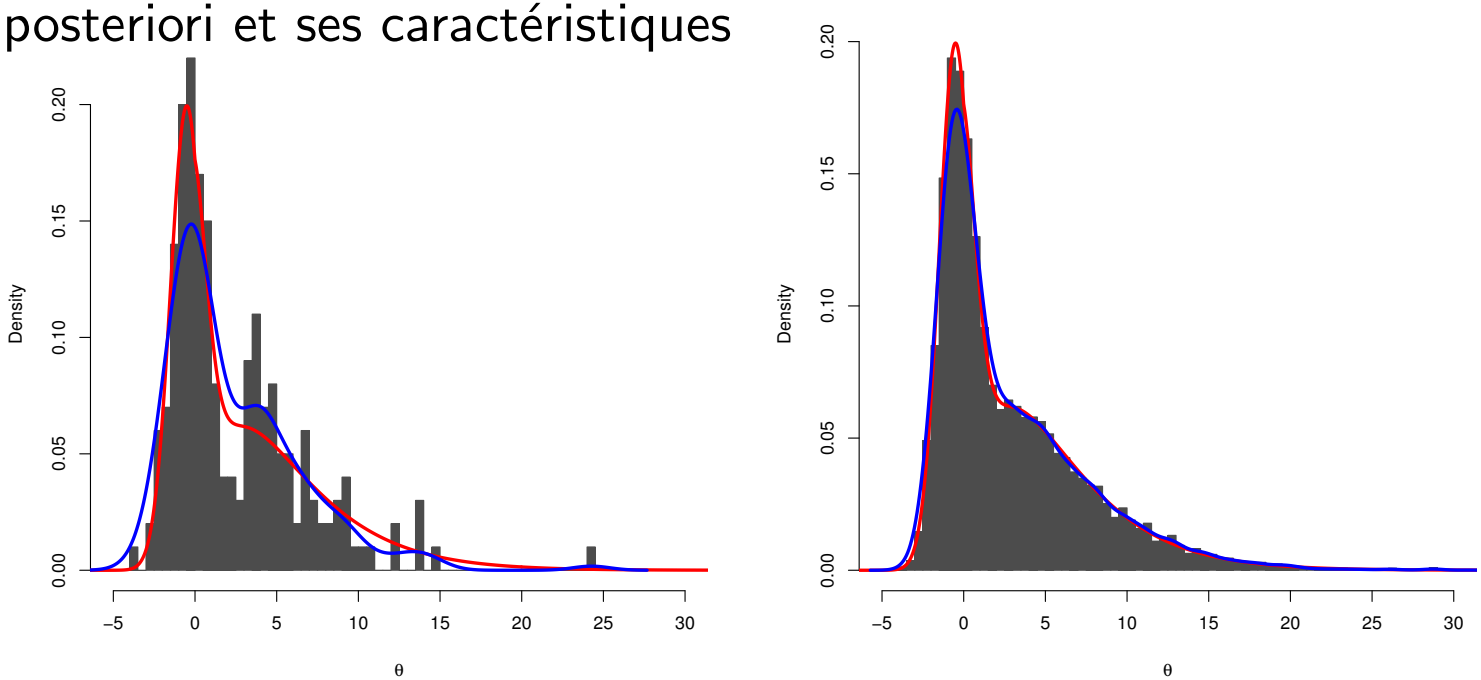
$$f(\theta_1, \ldots, \theta_K \mid Y) = \frac{f(Y \mid \theta_1, \ldots, \theta_K)\pi(\theta_1, \ldots, \theta_K)}{\int_{\Theta_1} \cdots \int_{\Theta_K} f(Y \mid \theta_1, \ldots, \theta_K)\pi(\alpha_1, \ldots, \alpha_K)d\alpha_1 \cdots d\alpha_K}$$

  - Les intégrales multiples (grande dimension) rendent difficile le calcul de la posterior jointe $f(\theta_1, \ldots, \theta_K \mid Y)$, des posteriors marginales $f(\theta_k \mid Y)$, des moments a posteriori $E(\theta_i^q \mid Y)$...

Introduction à l'inférence bayésienne et aux MCMC

- Méthodes numériques pour :
  - générer un échantillon issu de la loi a posteriori
  - sans passer par le calcul d'intégrales multiples

  Algorithme de Monte Carlo par chaînes de Markov (MCMC) par exemple
- Générer un grand échantillon pour correctement approcher la loi a posteriori et ses caractéristiques

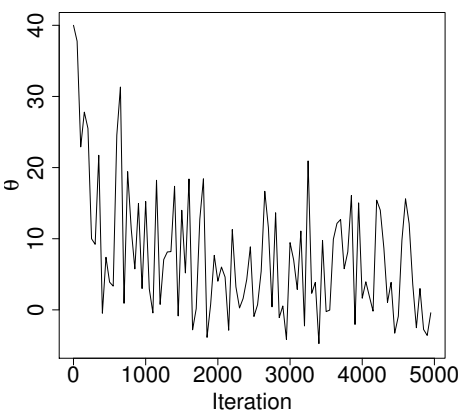


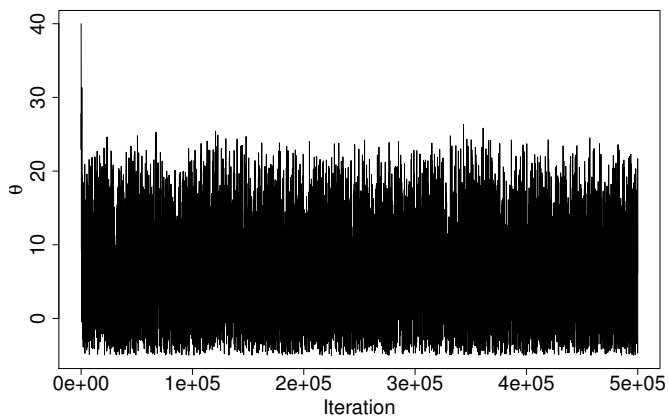
Echantillons de taille 200 (histo gauche) et 10000 (histo droite)

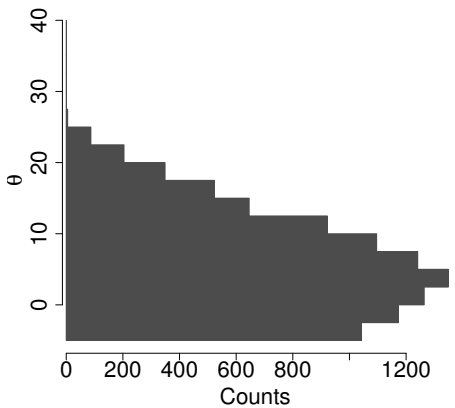
Densités a posteriori vraies (rouge) et estimées (bleu)

MCMC : Présentation

- Méthodes de Monte Carlo par Chaînes de Markov
- Algorithmes séquentiels : une séquence de réalisations dépendantes (i.e. une chaîne) de θ est générée
- Exploration ciblée de l'espace des paramètres (et des variables latentes)
- Qu'est-ce qu'une chaîne ?



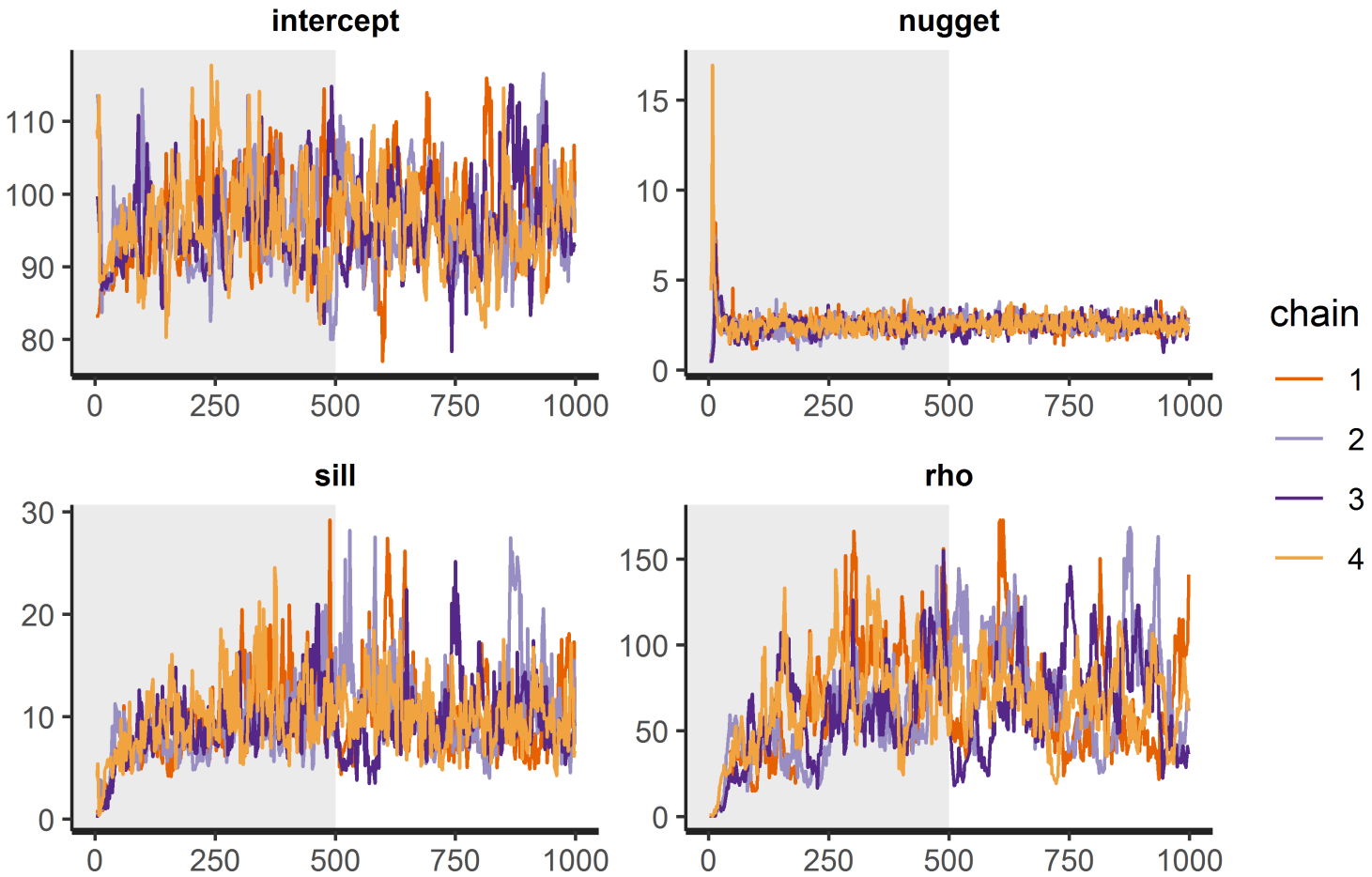




cf. cours Samuel Soubeyrand jour 3

MCMC : Présentation

chaînes à problème...



Logiciels

- MCMC (Markov Chain Monte Carlo) :**WinBUGS**, OpenBUGS, **JAGS** (appelé depuis R avec package rjags) , Nimble
- HMC (Hamiltonian Monte Carlo) : **Stan** (appelé depuis R avec package rstan)
- INLA (Integrated Nested Laplace Approximations) : R-INLA
- PMC (Population Monte Carlo) : BIIPS, Nimble

Logiciels pendant l'école chercheurs

- WinBUGS : BUGS écrit en Pascal (sur Windows seulement), interface clique-bouton
- JAGS : Just Another Gibbs Sampler (Martyn Plummer), package R **rjags**
- Stan : Stanislas Ulaw, co-inventeur des méthodes de Monte Carlo, package R **rstan**

Logiciel	WinBUGS	JAGS	Stan	ABC
Package	R2winBUGS	rjags	rstan	rABC
Convivialité	R/interface	R	R	R/interface
	+++	++	+	++
Algorithme	MCMC	MCMC	HamiltonienMC	
	MHwGibbs	MHwGibbs		
Activité	$\searrow$	$\rightarrow$	$\nearrow$	$\rightarrow$

Prise en main des 3 logiciels

- WinBUGS (version interface + version R2WinBUGS)
- JAGS
- Stan

Trois étapes

1. modèle bayésien
2. simulation de données
3. faire tourner le modèle

Prise en main des 3 logiciels

1. modèle bayésien : script du modèle BUGS

```
model{
#Poisson model script
for (i in 1:N) {
y[i] ~ dpois(lambda)
}
log(lambda) <- theta
theta ~ dnorm(0, .0001)
ynew ~ dpois(lambda)
}
```

- Twiddles symbol $\sim$ for a stochastic relation
- Left arrow $\leftarrow$ for a deterministic relation
- Hash key # for comments
- Arrays are indexed by terms within square brackets.

Prise en main des 3 logiciels

1. modèle bayésien

2. simulation de données

```
set.seed(1234)
N <- 100
y <- rpois(N, 10)
```

3. faire tourner le modèle

Comparaison WinBUGS / JAGS / Stan

Logiciel	WinBUGS	JAGS (rjags)	Stan (rstan)
Manuel	BUGS book	$\leq$ 100 pages	$\geq$ 600 pages
Rédaction du code	non structuré	non structuré	très structuré
Paramétrisation y $\leftarrow$ dnorm(mu, sd)	y $\leftarrow$ dnorm(mu, 1/(sd*sd))	y $\leftarrow$ dnorm(mu, 1/(sd*sd))	y = normal(mu, sd) ;

Comparaison WinBUGS / JAGS / Stan

Logiciel		WinBUGS	JAGS (rjags)	Stan (rstan)
Manuel		BUGS book	$\leq$ 100 pages	$\geq$ 600 pages
Rédaction du code		non structuré	non structuré	très structuré
Paramétrisation	y $\leftarrow$ dnorm(mu, sd)	y $\leftarrow$ dnorm(mu, 1/(sd*sd))	y $\leftarrow$ dnorm(mu, 1/(sd*sd))	y = normal(mu, sd) ;
ALGO	Gibbs	oui	oui	non
	Metropolis	oui	oui	non
	HMC	non	non	oui
	MLE	non	non	oui mais ?
	ODE	oui	non	oui mais ?

Comparaison WinBUGS / JAGS / Stan

Logiciel		WinBUGS	JAGS (rjags)	Stan (rstan)
Manuel		BUGS book	$\leq$ 100 pages	$\geq$ 600 pages
Rédaction du code		non structuré	non structuré	très structuré
Paramétrisation	y $\leftarrow$ dnorm(mu, sd)	y $\leftarrow$ dnorm(mu, 1/(sd*sd))	y $\leftarrow$ dnorm(mu, 1/(sd*sd))	y = normal(mu, sd) ;
ALGO	Gibbs	oui	oui	non
	Metropolis	oui	oui	non
	HMC	non	non	oui
	MLE	non	non	oui mais ?
	ODE	oui	non	oui mais ?
Message d'erreur		TRAP (debug = TRUE)	pas très précis	précis
Complexité du modèle	(nombre d'effets aléatoires)	lent	plus ou moins rapide	plus ou moins rapide
Temps de calcul		burn-in	burn-in	warm-up

Comparaison WinBUGS / JAGS / Stan

Comparaison sur le modèle Bernoulli
Comparaison sur le modèle Normal

Exemple 1 : données binaires

- vraies valeurs : $\mu = 0$ & $sd = \frac{\pi^2}{3}$
- Modèle

$$\mu \sim \text{Normale}(0, prior_{prec})$$

$$prior_{prec} = 1/(1.5 * 1.5)$$

$$prec = \frac{1}{sd^2}$$

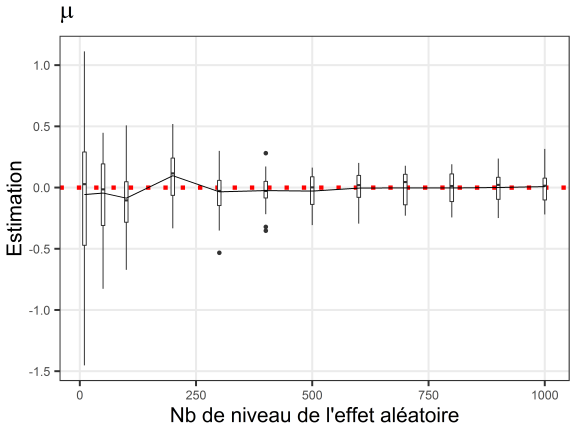
$$sd \sim \text{Normale}(0, 1)$$

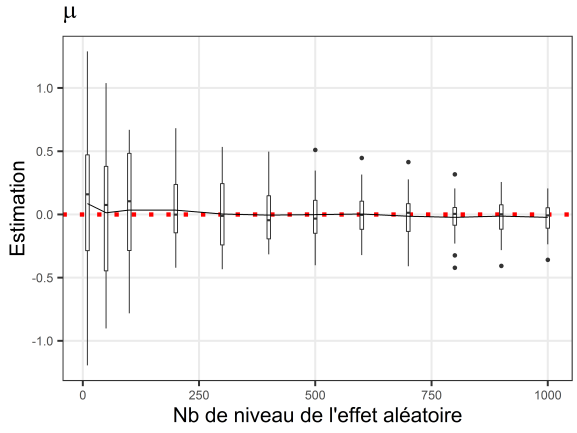
$$precision_{res} = \frac{1}{sd_{res}^2}$$

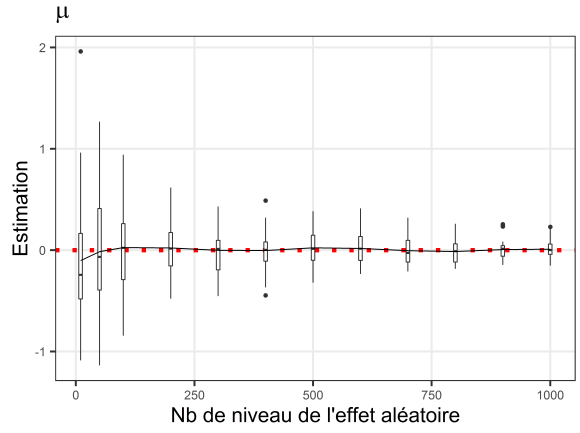
$$for\ i\ in\ 1 : n_{ind}\ \alpha_i \sim \text{Normale}(\mu, prec)$$

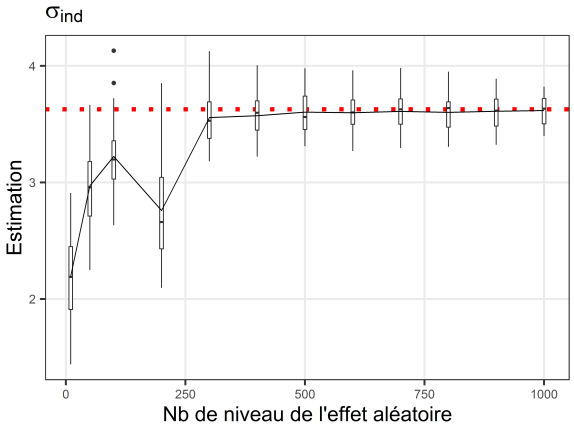
$$for\ j\ in\ 1 : n_{n_{obs}}\ y_j \sim \text{Bernoulli}(logit(\alpha[ind[j]]))$$

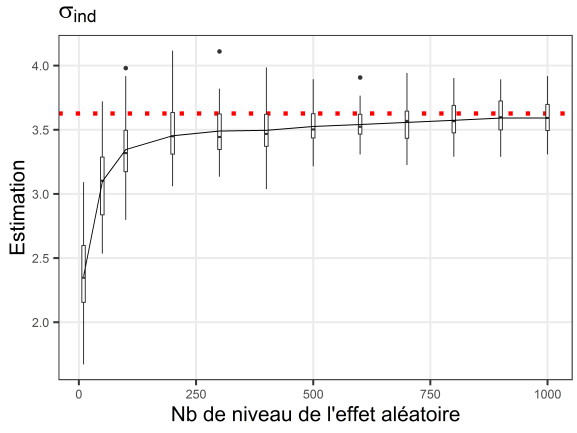
Comparaison WinBUGS / JAGS / Stan (Bernoulli)

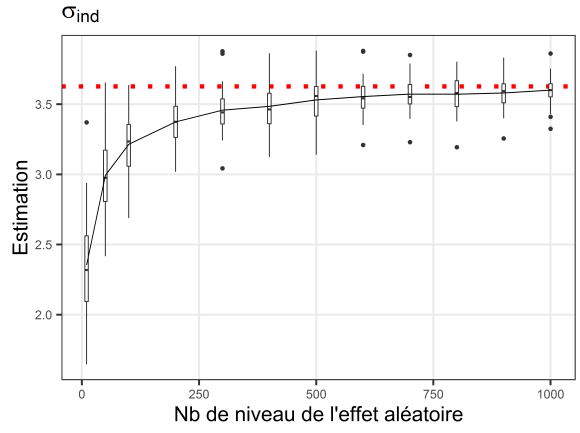




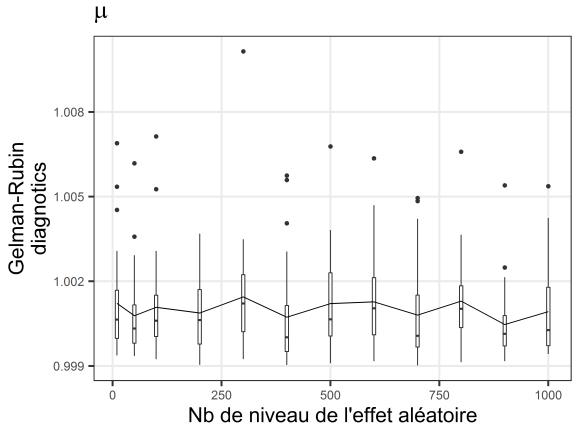


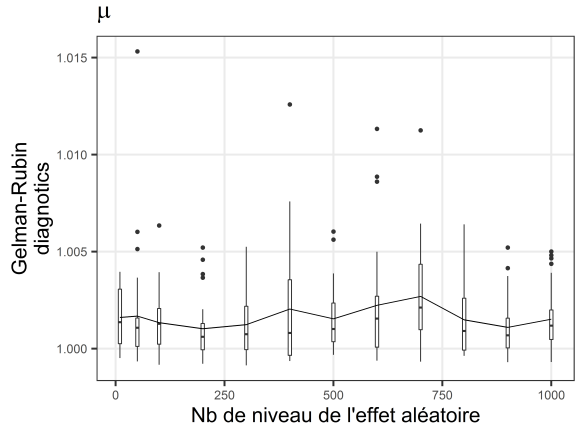


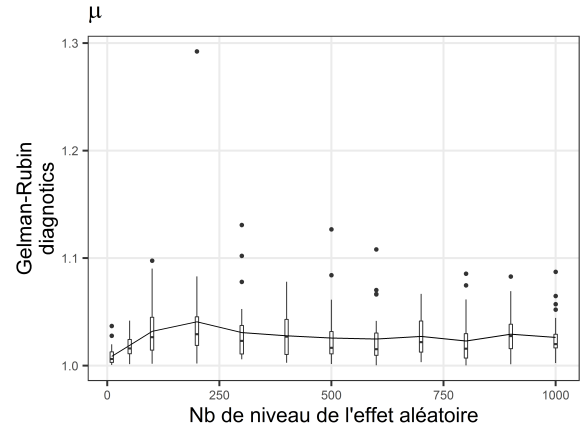


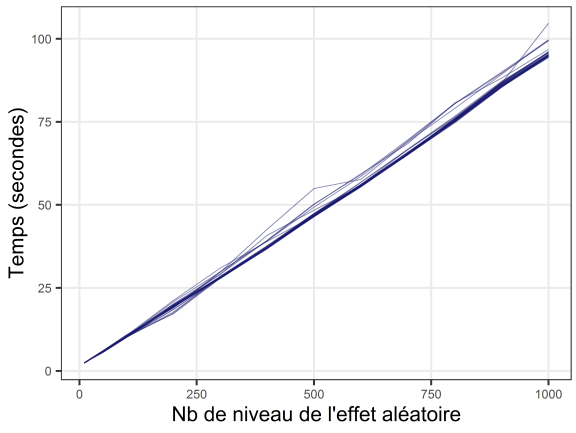


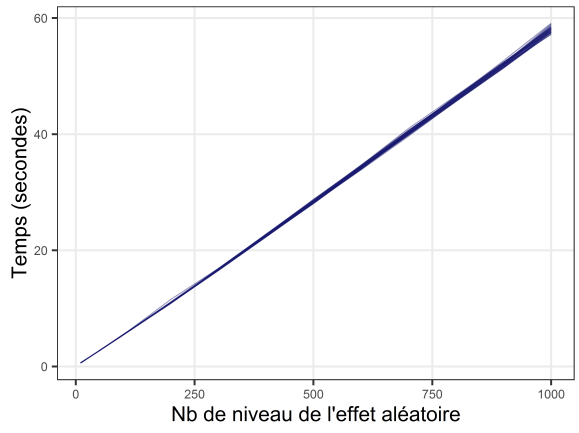
Comparaison WinBUGS / JAGS / Stan (Bernoulli)

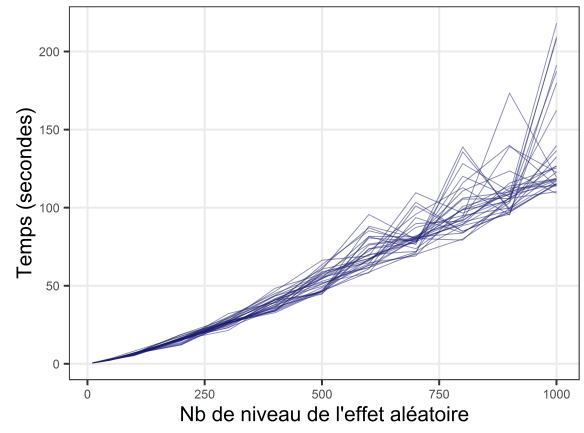




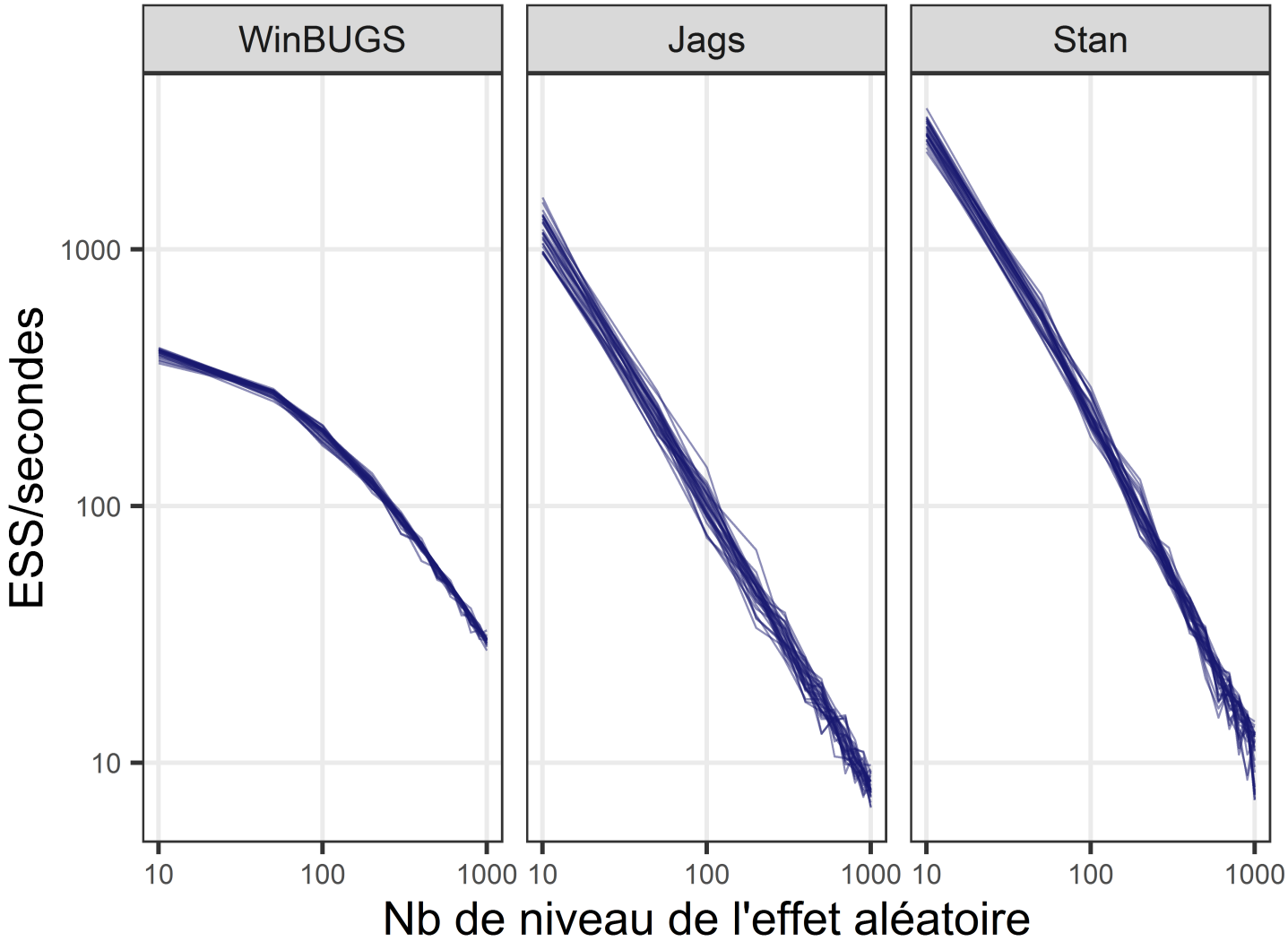








Comparaison WinBUGS / JAGS / Stan (Bernoulli)



Exemple 2 : données Normales

- vraies valeurs : $\mu = 0$ & $sd = 2$
- Modèle

$$\mu \sim \text{Normale}(0, 1)$$

$$prec = \frac{1}{sd^2}$$

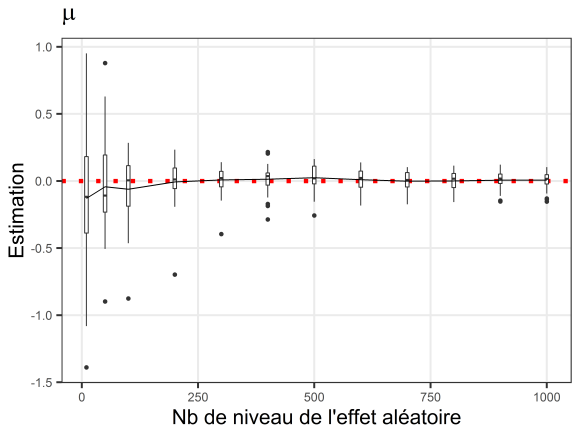
$$sd \sim \text{Normale}(0, 1)$$

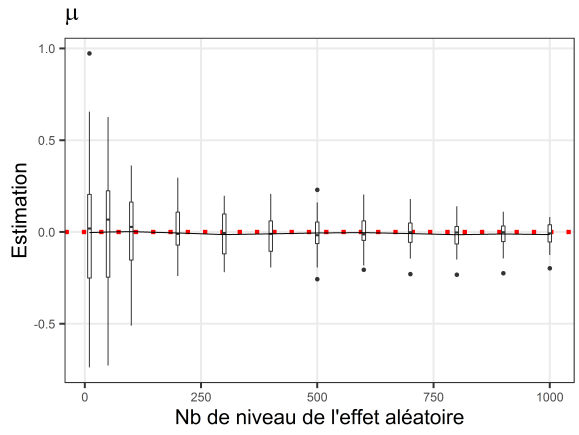
$$precision_{res} = \frac{1}{sd_{res}^2}$$

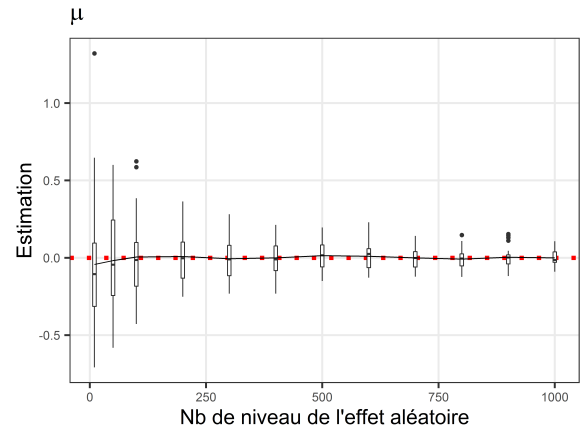
$$for\ i\ in\ 1 : n_{ind}\ \alpha_i \sim \text{Normale}(\mu, prec)$$

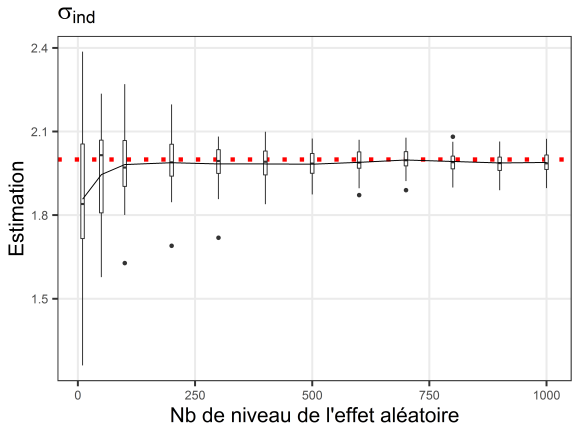
$$for\ j\ in\ 1 : n_{n_{obs}}\ y_j \sim \text{Normale}(\alpha[ind[j]], prec_{res})$$

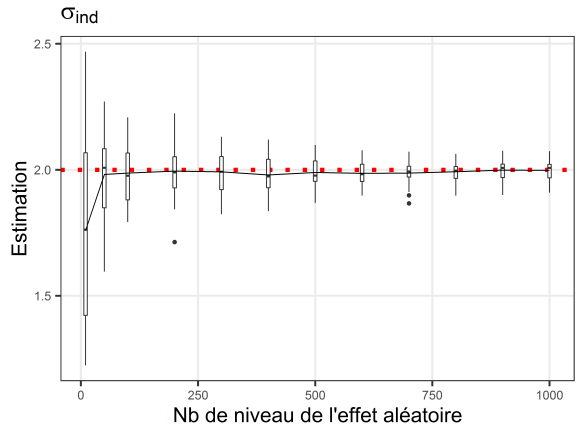
Comparaison WinBUGS / JAGS / Stan (Normal)

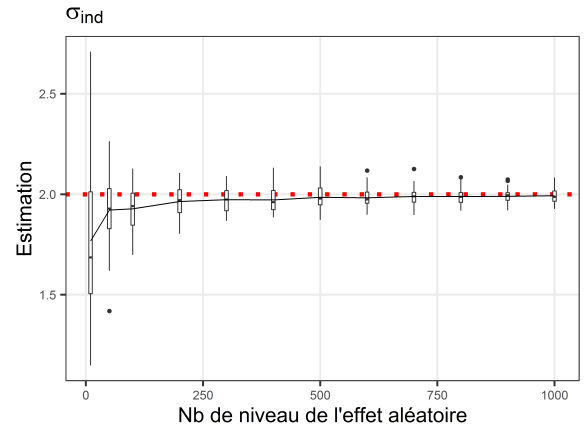




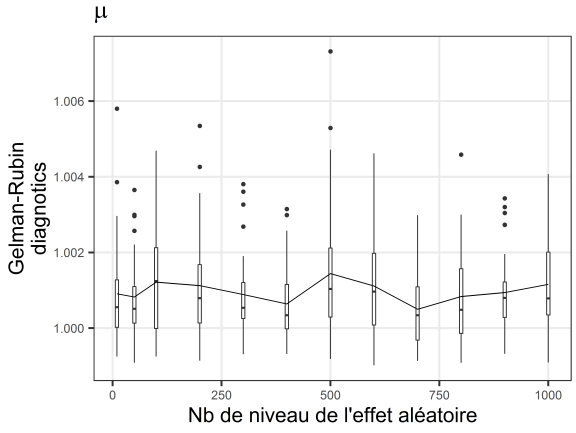


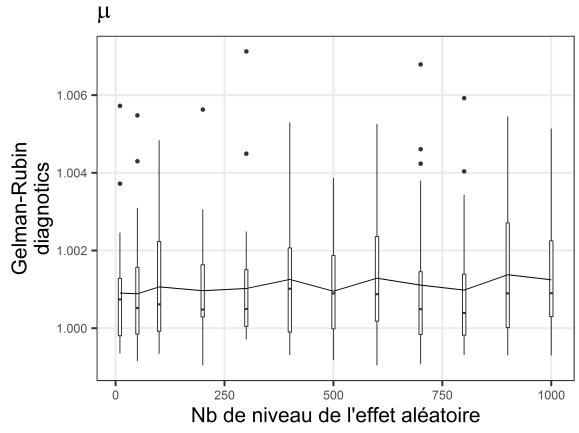


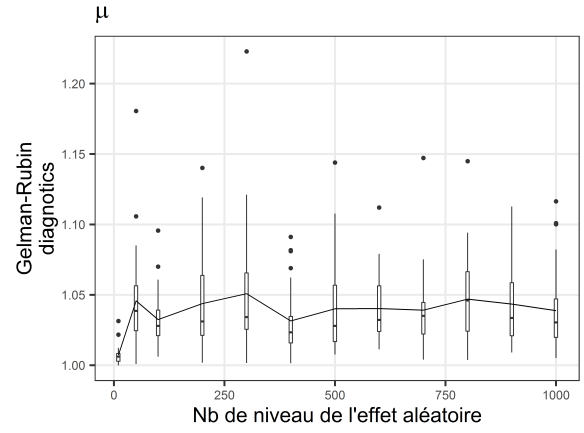


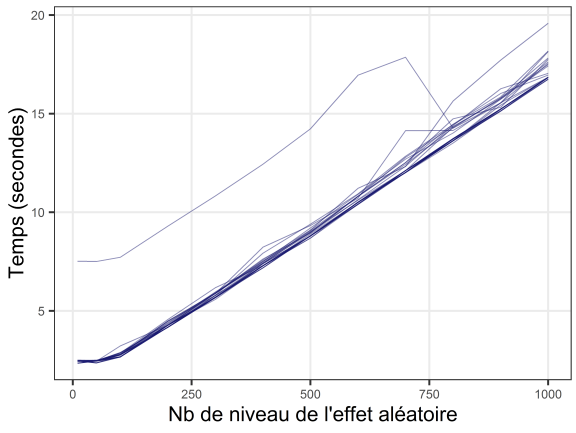


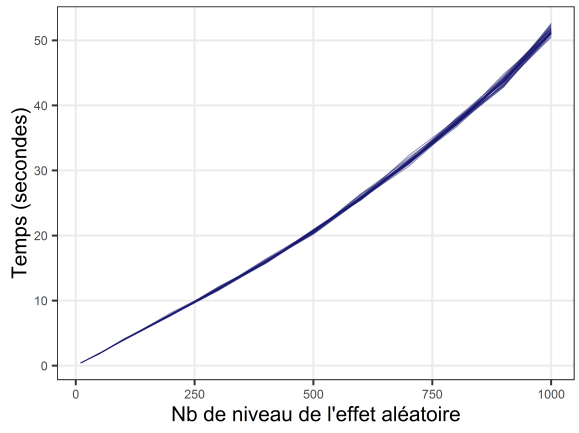
Comparaison WinBUGS / JAGS / Stan (Normal)

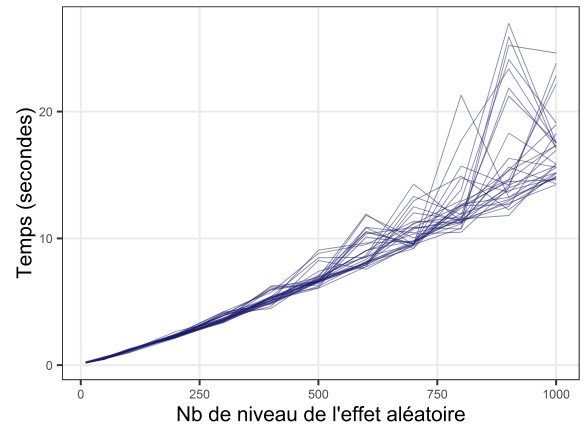




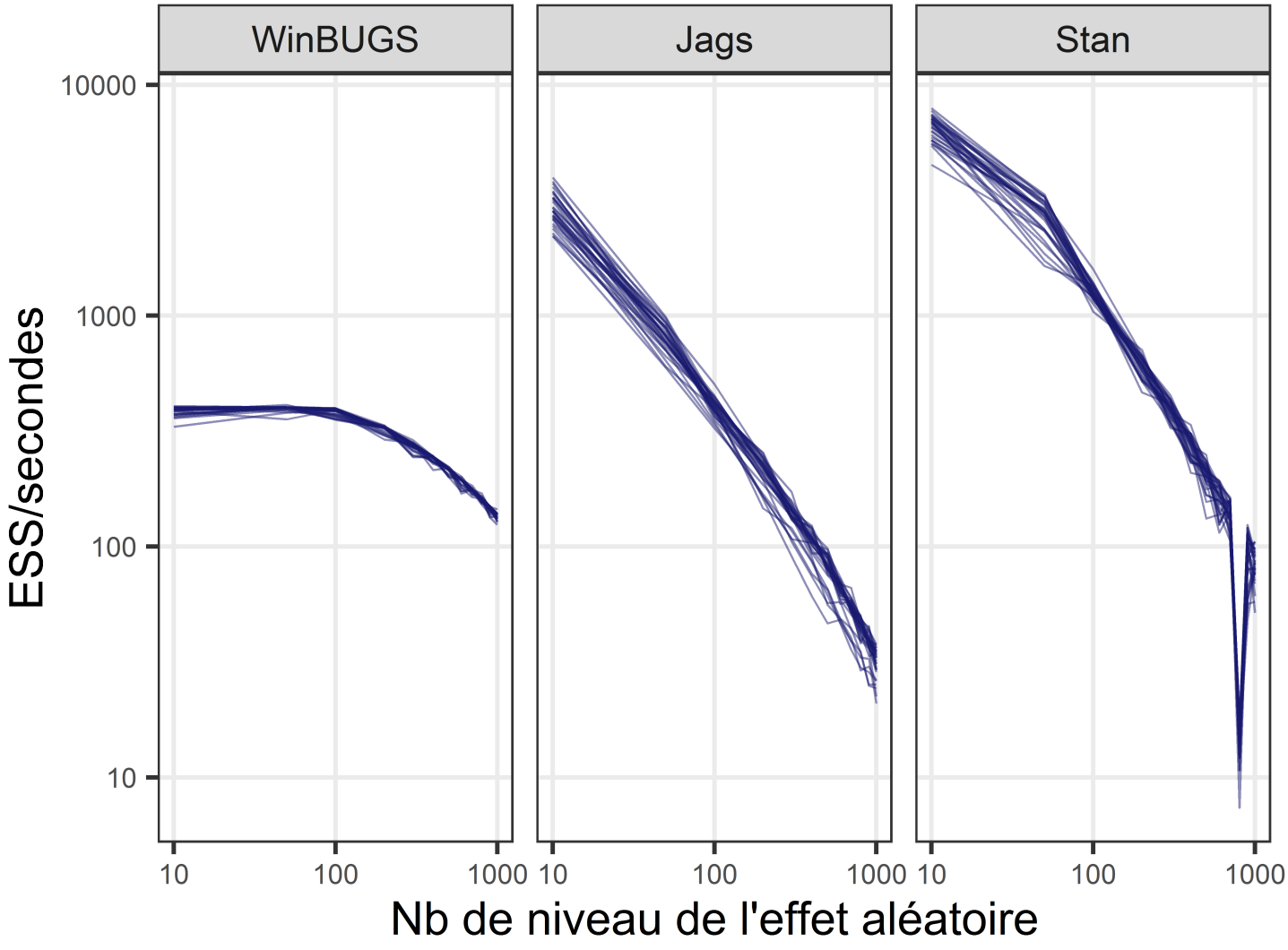








Comparaison WinBUGS / JAGS / Stan (Normal)



Groupes de TD

Choix pour les TD 1, 2 et 4

- WinBUGS
- JAGS
- Stan

TD 3 (pour tous les groupes) et TD 2 (JAGS et Stan) : utilisation de **Jupyter / Anaconda** (interface Python permettant d'inclure différents langages, dont R)

Pour ceux que ça intéresse : un MOOC sur Jupyter ouvre bientôt !

Groupes de TD

Choix pour les TD 1, 2 et 4

- WinBUGS
- JAGS
- Stan

TD 3 (pour tous les groupes) et TD 2 (JAGS et Stan) : utilisation de **Jupyter / Anaconda** (interface Python permettant d'inclure différents langages, dont R)

Pour ceux que ça intéresse : un MOOC sur Jupyter ouvre bientôt !

Pour la suite de l'école-chercheurs, 3 groupes : **à vous de choisir** parmi WinBUGS, rjags, rstan !

Comparaison WinBUGS / JAGS / Stan

Comparaison JAGS / Stan (Monnahan et al., MEE, 2016)

Table 3. Summary of key differences between JAGS and Stan

	JAGS	Stan
Inference	Bayesian only (MCMC)	Bayesian (MCMC with NUTS and variational inference) and penalized maximum likelihood
Tuning	Automatic with no options	Automatic with options for target acceptance rate (`adapt_delta`), mass matrix (diagonal or dense)
Discrete parameters	Use directly	Incompatible – must be marginalized out analytically
General pros	Easy to use, no tuning, discrete parameters	Scales well with dimensionality, posterior complexity; suitable for hierarchical models, especially the non-centered form
General cons	Few alternatives to reduce run-time when prohibitively slow	No discrete parameters, more difficult modelling language and additional MCMC diagnostics to check
Potential pathologies	No feedback	Divergences and excessive tree depths warn of steep or flat curvature, respectively