

# JAGS MODELISATION AVEC R

Abdoulaye OUATTARA

ORCID: 0009-0007-3024-3571

[contact@abouattara.bf](mailto:contact@abouattara.bf)

2025-09-09

## Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Diagnostics classiques pour s'assurer que les chaînes ont convergé</b>   | <b>1</b> |
| 1.1      | Traceplots (Graphiques des traces) . . . . .                                | 1        |
| 1.2      | Diagnostic de Gelman-Rubin (ou facteur de réduction de l'échelle) . . . . . | 2        |
| 1.3      | Résumé des outils de diagnostic de convergence MCMC . . . . .               | 2        |
| 1.4      | References . . . . .  | 4        |

## 1 Diagnostics classiques pour s'assurer que les chaînes ont convergé

Pour s'assurer que les chaînes de Markov Monte Carlo (MCMC) ont convergé vers la distribution cible dans une analyse bayésienne, on utilise des diagnostics classiques comme le diagnostic de Gelman-Rubin et les traceplots. Voici une explication simple de ces outils :

### 1.1 Traceplots (Graphiques des traces)

Ce sont des graphiques représentant l'évolution des valeurs échantillonnées d'un paramètre au fil des itérations de la chaîne MCMC.

**But** : vérifier visuellement si la chaîne « explore » bien l'espace des paramètres et si elle semble stationnaire (pas de tendance ou dérive).

**Interprétation** : Une chaîne convergente ressemble à un « *bruit blanc* » autour d'une valeur stable. Si la chaîne « *saute* » beaucoup, ou présente une tendance, cela suggère que la convergence n'est pas atteinte.

On trace souvent plusieurs chaînes indépendantes (avec des points de départ différents) sur le même graphique pour comparer leur comportement.

```
library(coda)

# Supposons que 'samples' est un objet mcmc.list contenant plusieurs chaînes

# Traceplots
plot(samples) # Affiche les traces et densités pour chaque paramètre
```

## 1.2 Diagnostic de Gelman-Rubin (ou facteur de réduction de l'échelle)

Ce diagnostic compare la variance **inter-chaînes** à la variance **intra-chaîne** pour chaque paramètre.

**Principe** : si plusieurs chaînes ont convergé vers la même distribution, leurs moyennes et variances doivent être similaires. Le diagnostic calcule un facteur appelé *Potential Scale Reduction Factor* (PSRF), noté souvent  $\hat{R}$ .

### Interprétation

$\hat{R} \approx 1$  signifie que les chaînes ont convergé. En pratique, on considère que la convergence est satisfaisante si  $\hat{R} < 1.1$  (parfois  $\hat{R} < 1.05$ ). Des valeurs supérieures indiquent que les chaînes n'ont pas encore convergé et qu'il faut prolonger la simulation.

Ce diagnostic nécessite au moins deux chaînes indépendantes initialisées différemment (idéalement « overdispersées »).

```
library(coda)

# Supposons que 'samples' est un objet mcmc.list contenant plusieurs chaînes

# Diagnostic Gelman-Rubin
gelman_results <- gelman.diag(samples)
print(gelman_results)
```

- La fonction `gelman.diag()` retourne pour chaque paramètre un PSRF avec un intervalle de confiance.
- On attend que ces valeurs soient proches de 1.

## 1.3 Résumé des outils de diagnostic de convergence MCMC

| Outil               | But                                    | Indication d'une bonne convergence     |
|---------------------|--|--|
| Traceplot           | Visualiser la stabilité                | Chaînes stables, sans tendance ni saut |
| Gelman-Rubin (PSRF) | Comparer variances intra/inter-chaînes | PSRF proche de 1 (ex. $< 1.1$ )        |

Table 1: Résumé des outils de diagnostic de convergence MCMC

### 1.3.1 Erreurs souvent rencontrées

**Erreur:** *Erreur dans chol.default(W) : the leading minor of order 2 is not positive*

Ce message, rencontré lors de l'utilisation de *gelman.diag()*, indique un échec dans le calcul du diagnostic de Gelman-Rubin. Les causes fréquentes incluent :

- Un nombre insuffisant d'itérations ou de chaînes MCMC ;
- Des chaînes constantes ou presque constantes ;
- Des problèmes numériques dans le calcul de la matrice de covariance ;
- Des paramètres non identifiables ou mal spécifiés dans le modèle.

*Comment y remédier ?*

#### 1. Vérifier la structure des échantillons MCMC

- Utilisez au moins deux chaînes (idéalement trois ou plus).
- Chaque chaîne doit comporter un grand nombre d'itérations (généralement plusieurs milliers).
- Vérifiez que les chaînes varient (à l'aide de *plot(samples)*).

#### 2. Augmenter le nombre d'itérations et la période d'adaptation

- Augmentez *n.iter* et *n.adapt* dans *coda.samples()* ou *jags.samples()*.

#### 3. Vérifier la spécification du modèle

- Assurez-vous que les paramètres sont identifiables.
- Supprimez ou fixez certains paramètres pour simplifier le modèle.

#### 4. Reparamétriser le modèle

- Appliquez des transformations (ex. *log*) pour des paramètres contraints.
- Centrez et réduisez les variables.

#### 5. Inspecter les diagnostics visuels

- Utilisez *plot(samples)* pour observer les traces.
- Utilisez *densplot(samples)* pour analyser les densités marginales.

#### 6. Exclure les paramètres constants

- *gelman.diag()* échoue sur des paramètres invariants.
- Supprimez ces paramètres du jeu d'échantillons.

#### Autres stratégies si la convergence échoue

- Simplifiez le modèle (réduisez le nombre de paramètres).
- Vérifiez l'intégrité et le format des données.
- Utilisez des priors plus informatifs.
- Testez des versions simplifiées du modèle pour localiser le problème.
- Augmentez le nombre de chaînes et d'itérations.
- Testez d'autres outils (ex. *nimble*, *Stan*).

## 1.4 References

- [The MCMC procedure – Gelman-Rubin Diagnostics, documentation SAS](#)
- [Bayesian Analysis: Gelman-Rubin Convergence, documentation Stata](#)
- [MCMC Diagnostics, support de cours \(University of Iowa\)](#)
- [Good convergence diagnostic, bad trace plot \(Stackexchange\)](#)