Devoir Méthodes bayésiennes : session 1

Benoit Gachet, Guillaume Mulier

17/12/2020

Table of Contents

# Les données

sncf\_machines <- tibble(  
 machine = 1:10,  
 anciennete = c(2, 14, 2, 9, 15, 7, 3, 14, 5, 2),  
 nb\_pannes = c(3, 50, 7, 20, 44, 3, 1, 58, 8, 7)  
)

# Modèle 1

Le modèle est le suivant :

On a .

## Question 1

## Question 2

Réalisation du modèle avec JAGS :

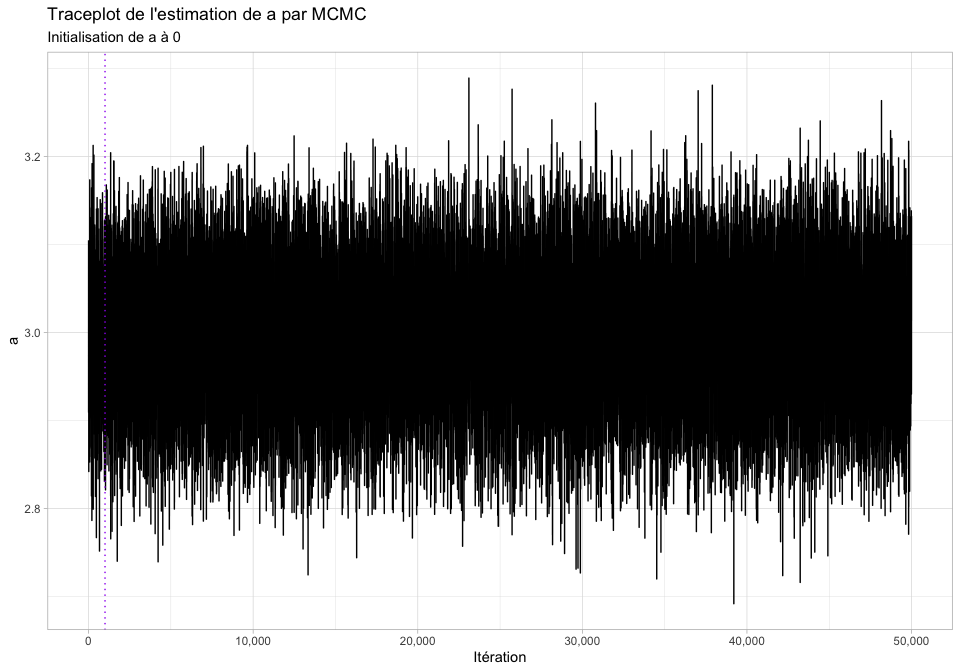
# Données à présenter sous forme d'une liste  
donnees <- as.list(sncf\_machines)  
# Modèle dans langage BUGS et pas en langage R  
modele\_1 <- function() {  
 # Modèle pour yi  
 for (i in 1:10) {  
 nb\_pannes[i] ~ dpois(exp(a))  
 }  
 # Loi a priori de a  
 a ~ dnorm(0, 0.001)  
}  
# Paramètres à recueillir  
parametres\_modele1 <- c("a")  
# Valeurs initiales  
inits\_1 <- list("a" = 0)  
inits\_modele1 <- list(inits\_1)  
n\_iter <- 50000 # Nombre d'iterations   
n\_burn <- 1000 # Burn in  
  
# Faire tourner le modèle avec la fonction jags  
# D'abord sans thin, ni burn in  
modele1\_fit <- jags(data = donnees,  
 inits = inits\_modele1,  
 parameters.to.save = parametres\_modele1,   
 n.chains = length(inits\_modele1),  
 n.iter = n\_iter,   
 n.burnin = 0,   
 n.thin = 1,  
 model.file = modele\_1)

## Compiling model graph  
## Resolving undeclared variables  
## Allocating nodes  
## Graph information:  
## Observed stochastic nodes: 10  
## Unobserved stochastic nodes: 1  
## Total graph size: 14  
##   
## Initializing model

modele1\_fit\_mcmc <- as.mcmc(modele1\_fit)

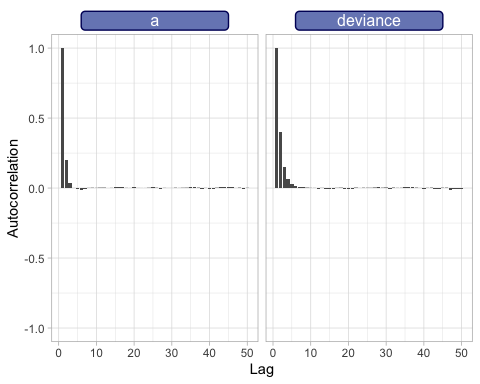
On regarde si le paramètre a estimé a bien convergé.

gg\_modele1 <- ggs(modele1\_fit\_mcmc)  
ggplot(gg\_modele1 %>% filter(Parameter == "a"), aes(x = Iteration, y = value)) +  
 geom\_line() +  
 geom\_vline(xintercept = 1000, color = "purple", linetype = "dotted") +  
 scale\_x\_continuous(labels = scales::comma\_format()) +  
 labs(x = "Itération",  
 y = "a",  
 title = "Traceplot de l'estimation de a par MCMC",  
 subtitle = "Initialisation de a à 0")



On voit que la valeur du paramètre *a* reste autour de la valeur 3 et n’a pas l’air de s’écarter beaucoup de cette valeur. Nous vérifierons par la suite si cette convergence est conservée en augmentant le nombre de chaînes. Nous allons ensuite vérifier si les valeurs des paramètres estimés

ggs\_autocorrelation(gg\_modele1)



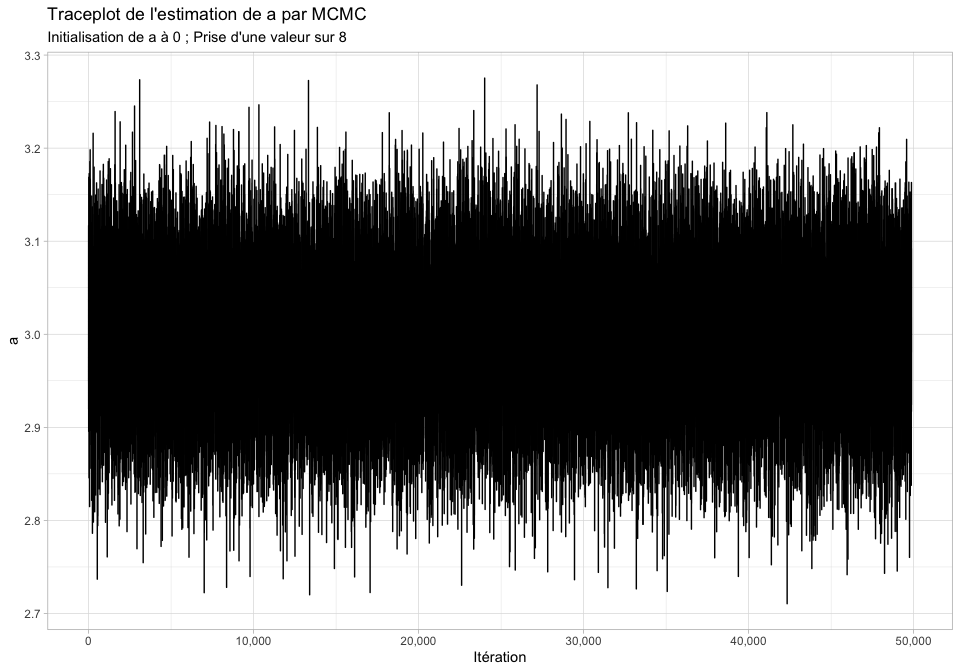
On voit que pour l’estimation de a, il y a corrélation jusqu’à la 3ème mesure. Pour la déviance, en revanche, cela va jusquà 8. Nous allons donc prendre une estimation sur 8 pour essayer de casser cette autocorrélation.

n\_thin <- 8  
# Faire tourner le modèle avec la fonction jags  
# D'abord sans thin, ni burn in  
modele1\_fit\_thin <- jags(data = donnees,  
 inits = inits\_modele1,  
 parameters.to.save = parametres\_modele1,   
 n.chains = length(inits\_modele1),  
 n.iter = n\_iter \* n\_thin,   
 n.burnin = n\_burn,   
 n.thin = n\_thin,  
 model.file = modele\_1)

## Compiling model graph  
## Resolving undeclared variables  
## Allocating nodes  
## Graph information:  
## Observed stochastic nodes: 10  
## Unobserved stochastic nodes: 1  
## Total graph size: 14  
##   
## Initializing model

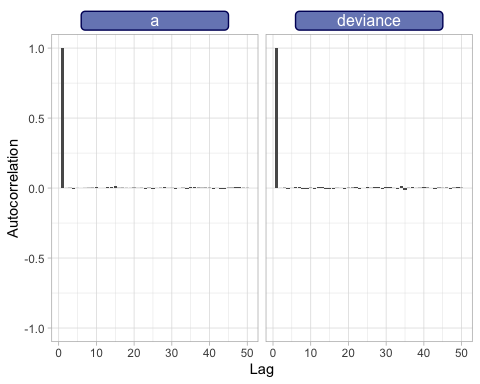
modele1\_fit\_thin\_mcmc <- as.mcmc(modele1\_fit\_thin)

gg\_modele1\_thin <- ggs(modele1\_fit\_thin\_mcmc)  
ggplot(gg\_modele1\_thin %>% filter(Parameter == "a"), aes(x = Iteration, y = value)) +  
 geom\_line() +  
 scale\_x\_continuous(labels = scales::comma\_format()) +  
 labs(x = "Itération",  
 y = "a",  
 title = "Traceplot de l'estimation de a par MCMC",  
 subtitle = "Initialisation de a à 0 ; Prise d'une valeur sur 8")



En ne prenant qu’une observation sur 8, on voit que le traceplot reste similaire avec une bonne convergence de l’estimation de a autour de 3 après le retrait de 1000 observations de burn in.

ggs\_autocorrelation(gg\_modele1\_thin)



Sur le graphique d’auto-corrélations, on voit que le problème de mélangeance a été réglé et que maintenant il n’y a plus d’auto-corrélation pour le paramètre a estimé.

Afin de nous assurer de la convergence du modèle, nous avons réalisé 3 chaînes avec des départ pour des valeurs différentes. Nous avons initié a à -5, 0 et 5 et regardé comment se comportait le modèle.

inits\_2 <- list("a" = 5)  
inits\_3 <- list("a" = -5)  
inits\_modele1\_mult <- list(inits\_1, inits\_2, inits\_3)  
modele1\_fit\_mult <- jags(data = donnees,  
 inits = inits\_modele1\_mult,  
 parameters.to.save = parametres\_modele1,   
 n.chains = length(inits\_modele1\_mult),  
 n.iter = n\_iter \* n\_thin,   
 n.burnin = n\_burn,   
 n.thin = n\_thin,  
 model.file = modele\_1)

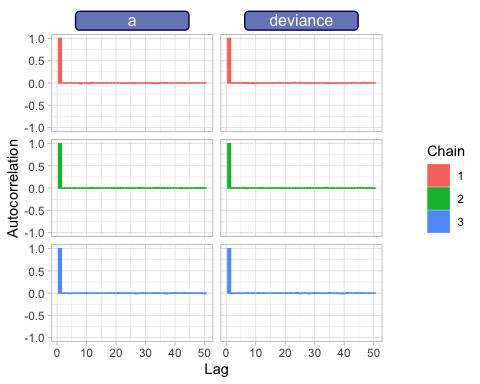
## Warning in jags.model(model.file, data = data, inits = init.values, n.chains =  
## n.chains, : Unused variable "machine" in data

## Warning in jags.model(model.file, data = data, inits = init.values, n.chains =  
## n.chains, : Unused variable "anciennete" in data

## Compiling model graph  
## Resolving undeclared variables  
## Allocating nodes  
## Graph information:  
## Observed stochastic nodes: 10  
## Unobserved stochastic nodes: 1  
## Total graph size: 14  
##   
## Initializing model

modele1\_fit\_mult\_mcmc <- as.mcmc(modele1\_fit\_mult)  
gg\_modele1\_mult <- ggs(modele1\_fit\_mult\_mcmc)

ggs\_autocorrelation(gg\_modele1\_mult) +  
 facet\_grid(Chain ~ Parameter)



ggplot(gg\_modele1\_mult %>% filter(Parameter == "a"), aes(x = Iteration, y = value)) +  
 geom\_line(aes(color = as.factor(Chain)), alpha = 0.3) +  
 geom\_vline(xintercept = 1000, color = "purple", linetype = "dotted") +  
 scale\_x\_continuous(labels = scales::comma\_format()) +  
 labs(x = "Itération",  
 y = "a",  
 title = "Traceplot de l'estimation de a par MCMC avec 3 chaînes") +  
 theme(legend.position = "none")

