Metropolis Hastings

MP Etienne

11/14/2019

La fonction `rprop`

```r
rprop <- function(p, sd_explore){
  rnorm(1, mean = p , sd = sd_explore)
}
```

La fonction `vrais`

```r
vrais <- function(y, n, p){
  dbinom(x = y, size = n, prob = p )
}
```

La fonction `dprop`

```r
dprop <- function(p_depart, p_arrivee, sd_explore){
  dnorm(x = p_arrivee, mean = p_depart, sd = sd_explore)

}
```

La fonction `dprior`

```r
dprior <- function(p){
  dunif(x = p, min = 0, max = 1)
}
```

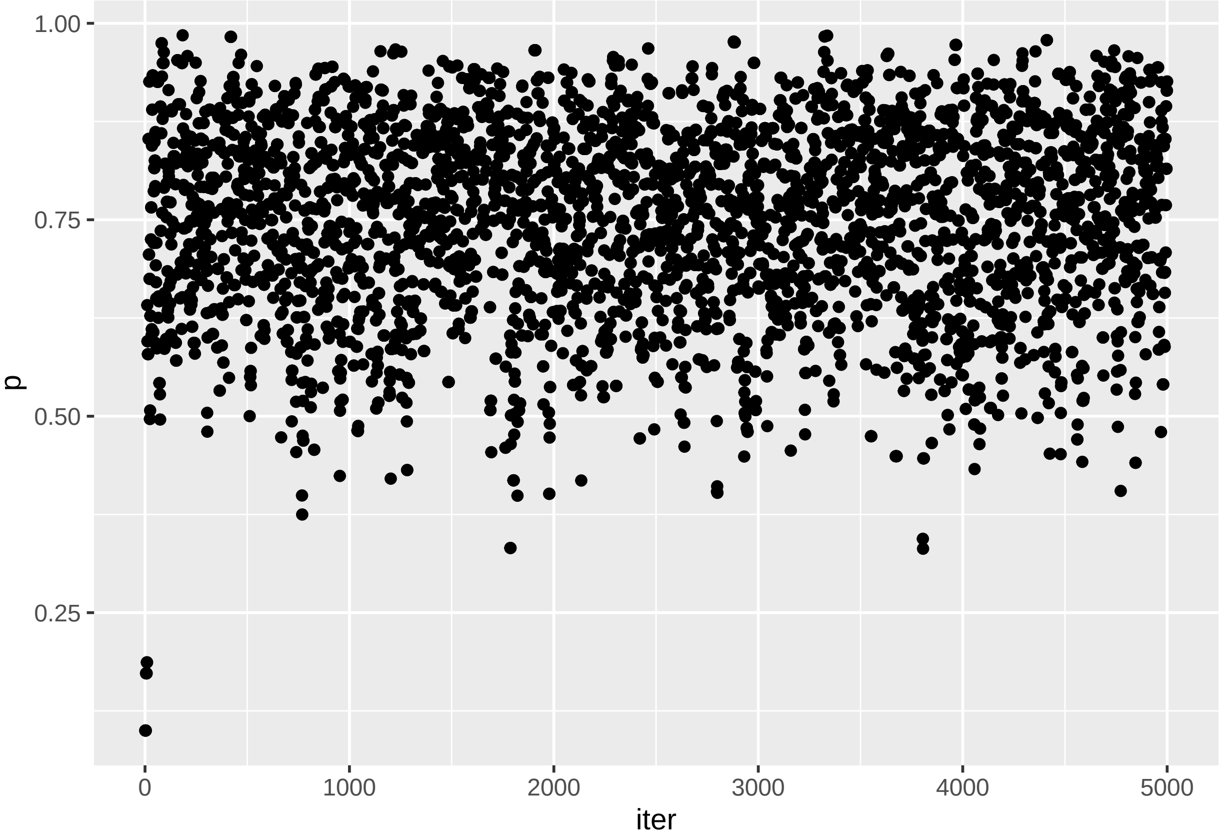
La fonction `compute_ratio`

```r
compute_ratio <- function(y, n, p_courant, p_candidat, sd_explore){
  if(p_candidat > 1 | p_candidat <= 0){
    ratio <- 0
  } else {

    num_bleu <- vrais(y = y, n = n, p = p_candidat) * dprior(p_candidat)
    denom_bleu <- vrais(y = y, n = n, p = p_courant) * dprior(p_courant)

    num_rouge <- dprop(p_depart = p_candidat, p_arrivee = p_courant, sd_explore = sd_explore)
    denom_rouge <- dprop(p_depart = p_courant, p_arrivee = p_candidat, sd = sd_explore)

    ratio <- num_bleu/denom_bleu* num_rouge/denom_rouge
  }
  return(ratio)
}
```

##Mise en oeuvre l'algo MH

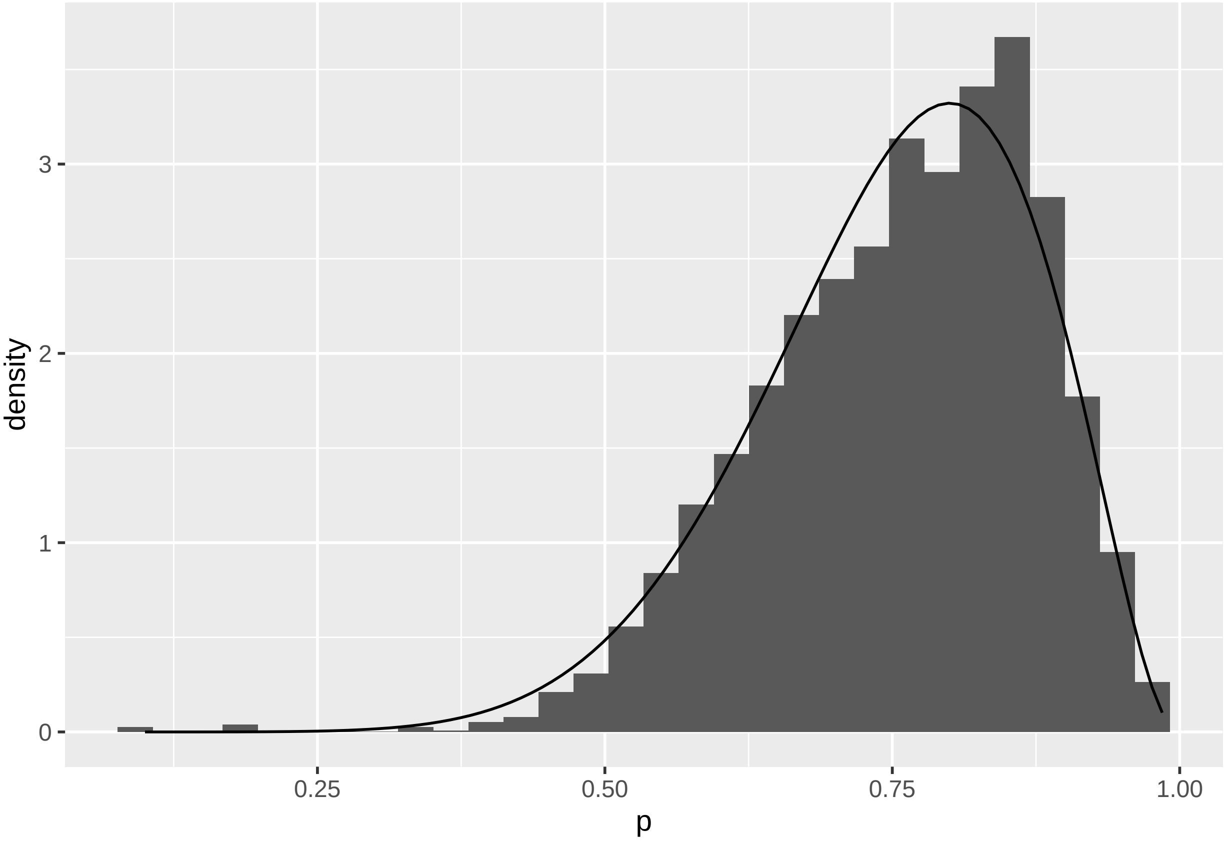
```
G <- 5000
p_sample <- numeric(5)
p_sample[1]  <- 0.1
y <- 8
n <- 10

sd_explore <- 0.2

for( i in 2:G){
  p_cand <- rprop(p = p_sample[i-1], sd_explore = sd_explore)
  ratio <- compute_ratio(y = y, n = n, p_courant = p_sample[i-1], p_candidat = p_cand, sd_explore = sd_

  p_sample[i] <- ifelse( runif(1, min = 0, max =1) < ratio,
                    p_cand, p_sample[i-1])
}
```

```
df <- data.frame(p = p_sample, iter = 1:G)

library(ggplot2)
ggplot(df) + geom_point(aes(x= iter, y =p))
```



```
ggplot(df) + geom_histogram(aes( x = p, y = ..density..)) +
  stat_function(fun = dbeta, args = list(shape1 = y+1, shape2 = n-y +1))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



## Mise en oeuvre du modèle CMR depuis R avec JAGS

```
library('rjags')
```

```
## Loading required package: coda

## Linked to JAGS 4.3.0

## Loaded modules: basemod,bugs
```

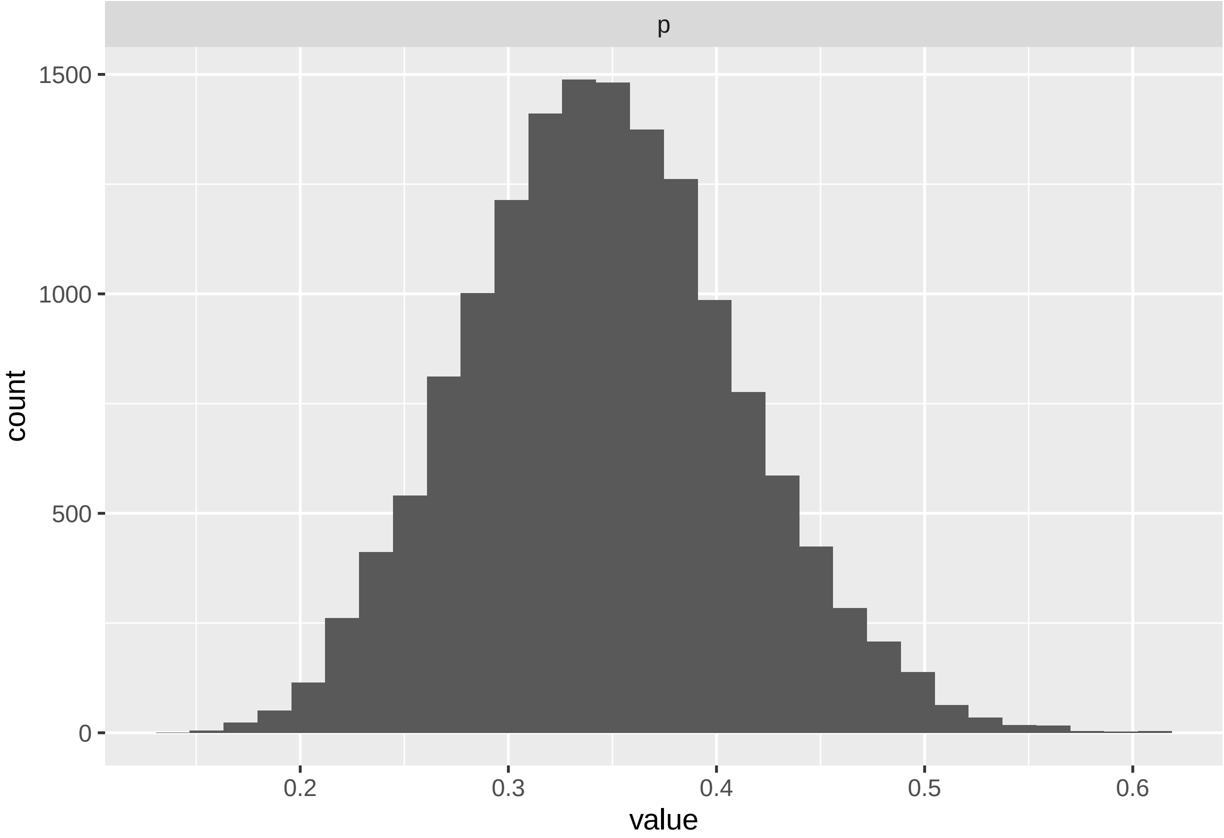
```
library('ggmcmc')
```

```
## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: tidyr

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

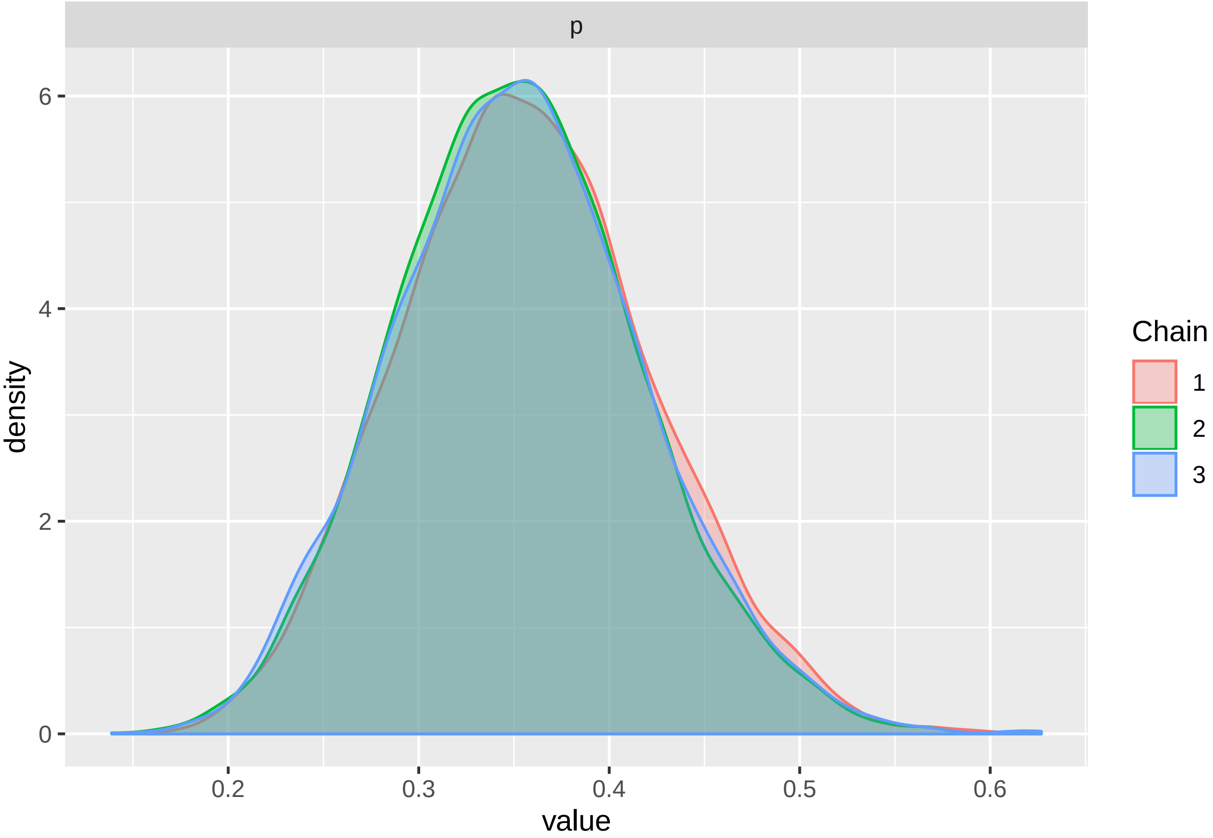
```r
data.list <- dget(file = 'data_CMR.txt')

init.list <- dget(file = 'init_CMR.txt')

mjags <- jags.model(file = 'modelCMR.txt',
           data = data.list,
           inits = init.list, n.chains = 3)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 2
##    Unobserved stochastic nodes: 2
##    Total graph size: 10
##
## Initializing model
```

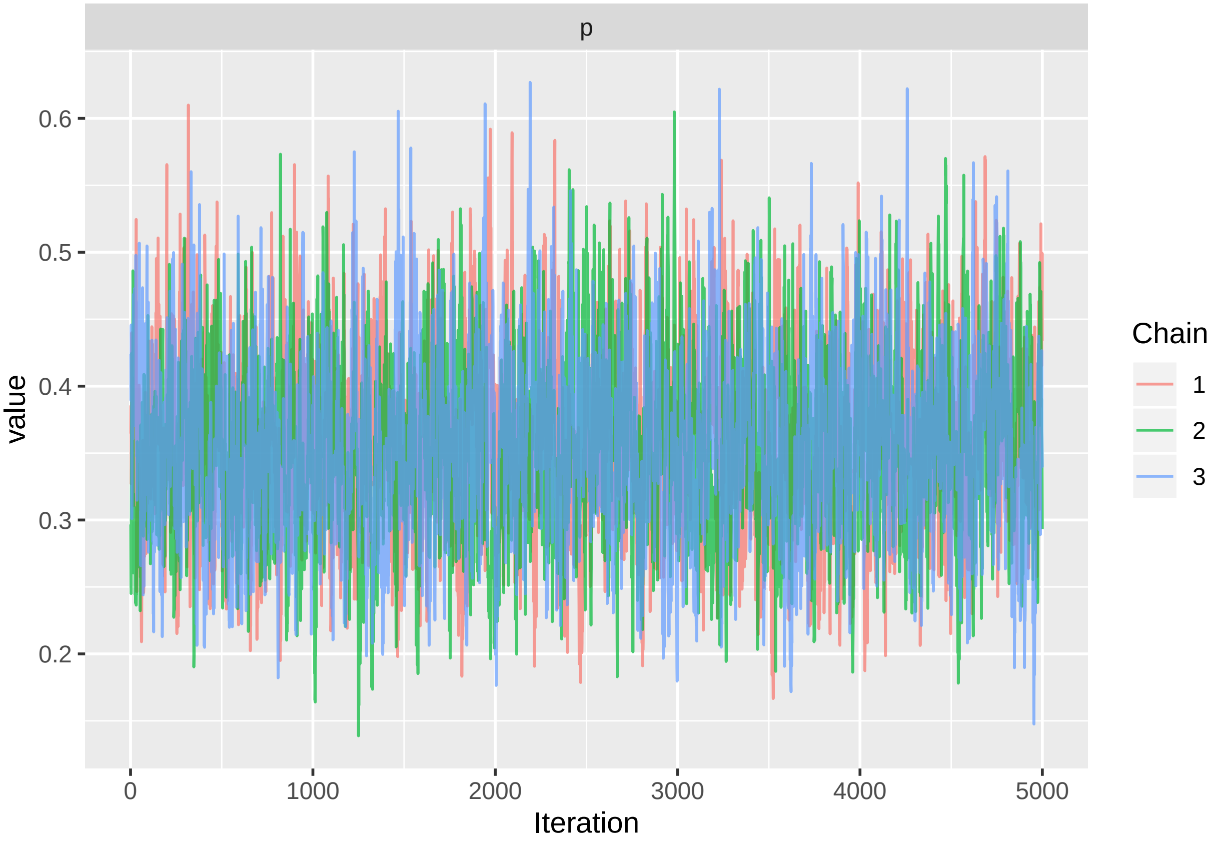
```r
post_sample <- jags.samples(mjags, variable.names = c('p', 'N'),
                            n.iter = 5000)

post_sample_mcmc_list <- as.mcmc.list(post_sample[[2]])

post <- ggs(S = post_sample_mcmc_list)

ggs_histogram((post))
```



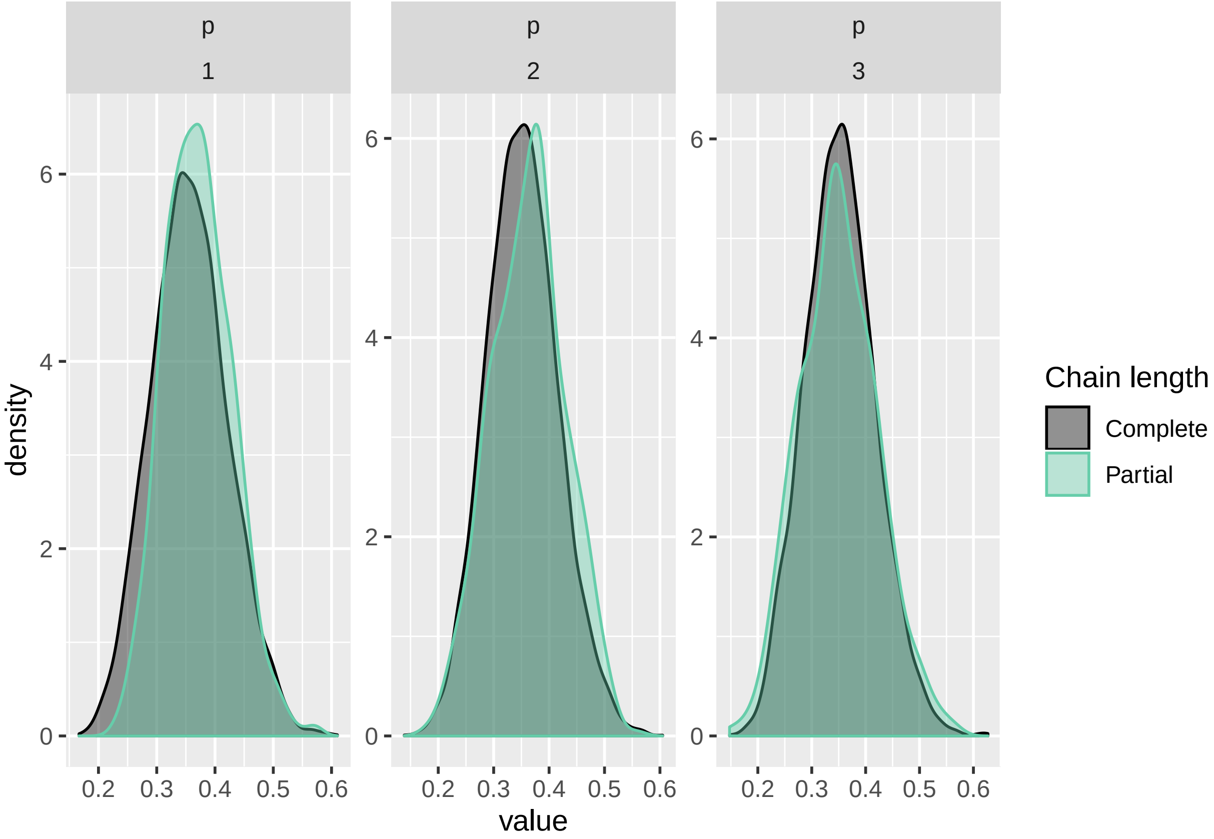
```
ggs_density((post))
```



```
ggs_traceplot(post)
```



```
ggs_compare_partial(post)
```


p
1
p
2
p
3
density
value
0
2
4
6
0.2
0.3
0.4
0.5
0.6
Chain length
Complete
Partial