Tareita 14/44

Simulación estocástica

Marco Antonio Andrade Barrera

3 de mayo de 2018

Considere $m = 13$ y $m_i : i \in J_m \sim U(m_i|25, 30)$

1. Genere $\tau^* \sim Ga(\tau|a = 0.5, b = 1)$.
2. Genere $\mu^* \sim N(\mu|33, \delta \cdot \tau^* = 1.7 \cdot \tau^*)$
3. $\forall j \in J_{m_i}\ \forall i \in J_{13}$

- Genere $\theta_i^* \sim N(\theta_i|\mu^*, \sigma^{-2} = h_i^* \cdot \tau^*)$
- Genere $h_i^* \sim Ga(h_i|\alpha = 1.77, \beta = 1)$

4. Genere $X_{ij} \sim N(x_i|\theta_i^*, \sigma^{-2} = h_i^*)$.

```
#fijar semilla
set.seed(1234)
#valores fijos
m <- 13
a <- 0.5
b <- 1
delta <- 1.7
alpha <- 1.77
beta <- 1
omega <- 33

mi <- sample(x = 25:30,size = m,replace = TRUE)
#Veamos los m valores generados de mi
mi
```

```
##  [1] 25 28 28 28 30 28 25 26 28 28 29 28 26
```

```
#Generamos tau*
tauStar <- rgamma(n = 1,shape = a, rate = b)
tauStar
```

```
## [1] 1.707601
```

```
#Generamos mu*
muStar <- rnorm(n = 1,mean = omega,sd = sqrt(1/(delta * tauStar)))
muStar
```

```
## [1] 33.57718
```

```
#Generamos h* (m valores)
hStar <- rgamma(n = m,shape = alpha,rate = beta)
hStar
```

```
##  [1] 0.66538994 0.78915936 0.06329051 1.34368013 2.58145210 1.14875528
##  [7] 1.29072566 2.18911111 0.64550501 1.42560466 0.77721824 1.27564881
## [13] 1.22794686
```

```r
#Generamos theta* (m valores)
thetaStar <- unlist(
  lapply(hStar,FUN = function(hi){
    thetai <- rnorm(n = 1,mean = muStar,sd = sqrt(1/(hi*tauStar)))
    thetai
    }))
thetaStar
```

```
##  [1] 32.21856 34.07230 30.46338 33.56719 33.13140 34.36422 33.25683
##  [8] 33.21025 33.09974 32.53306 32.56365 32.10010 32.65111
```

```r
#Generamos xij
xij <- lapply(1:m,FUN = function(i){
  #para i fijo, generamos mi valores (los j's)
  xj <- rnorm(n = mi[i],mean = thetaStar[i],sd = sqrt(1/(hStar[i])))
  xj
})
#mostramos los xij. Cada renglón corresponde a un i=1,...,m.
#Los j son variables, de acuerdo a los mi generados anteriormente
xij
```

```
## [[1]]
##  [1] 31.85778 31.64741 33.99553 30.90849 31.16995 31.87454 30.99958
##  [8] 31.03124 30.86108 30.68373 31.57639 31.60946 30.00451 31.50498
## [15] 30.85915 30.97430 32.01958 32.90882 34.23865 31.27049 34.18728
## [22] 30.79918 33.02349 35.34342 32.17595
##
## [[2]]
##  [1] 33.31850 34.06374 36.07274 32.79059 35.61205 35.56897 34.45106
##  [8] 34.08006 33.55959 33.65971 34.80207 36.40278 33.89962 32.50681
## [15] 33.25777 34.36302 33.71539 33.87217 33.88094 32.52752 33.87667
## [22] 35.02940 34.85759 34.69143 33.61895 33.85663 32.72764 34.01246
##
## [[3]]
##  [1] 31.47777 37.24449 34.44433 28.49346 31.87667 25.95338 33.95419
##  [8] 34.33067 38.89469 32.11109 28.57640 30.72570 28.46606 27.18008
## [15] 31.12715 26.90078 31.13191 31.87436 30.25626 29.68455 27.88336
## [22] 26.05212 33.83919 30.55227 33.76711 25.51741 31.13525 33.13917
##
## [[4]]
##  [1] 33.54452 33.40208 32.89265 35.34273 34.21464 35.14091 33.63626
##  [8] 33.02248 32.26170 33.01844 33.76242 34.44169 33.78523 32.55617
## [15] 34.14408 32.14367 33.25157 33.29448 31.88647 34.36091 33.02985
## [22] 33.27902 34.77076 34.11644 33.47365 34.01041 33.91164 35.00171
##
## [[5]]
##  [1] 33.30311 33.44650 33.34771 32.89661 33.19216 34.15135 32.58643
##  [8] 33.20718 33.97918 32.98537 32.47577 32.59005 32.88858 32.60401
## [15] 32.96918 32.87346 33.01747 33.38475 33.52017 34.17591 33.08864
## [22] 32.93171 34.04695 34.19217 33.15831 32.92435 31.99724 34.00976
## [29] 32.61009 32.43197
##
## [[6]]
##  [1] 37.20408 34.58349 34.33318 31.81503 34.27111 35.27486 34.75036
##  [8] 35.21542 36.21506 35.45500 33.88956 35.02122 34.17909 33.86219
```

```
## [15] 31.69976 33.62747 34.81935 36.38701 34.83137 34.94288 33.46302
## [22] 34.51597 32.42520 34.81694 35.01431 34.53730 35.01801 34.65502
##
## [[7]]
##  [1] 33.92620 34.87858 34.23594 33.28558 32.27589 33.62481 32.90455
##  [8] 34.57141 31.84228 32.89089 33.62829 33.12327 32.72330 32.98862
## [15] 33.81095 34.04477 33.83795 35.25796 34.28975 33.51008 32.67610
## [22] 35.82627 33.85310 32.65449 33.42098
##
## [[8]]
##  [1] 32.99100 33.02458 32.57932 33.28922 33.42596 32.48197 31.02504
##  [8] 33.03799 33.23020 33.61190 33.25022 33.48966 32.46829 33.69091
## [15] 33.69613 33.38033 34.12760 33.48362 33.38893 33.39141 33.50556
## [22] 33.92676 33.51587 33.65849 32.44220 32.95984
##
## [[9]]
##  [1] 34.93807 31.57640 33.42095 33.60383 34.31429 32.66551 33.29718
##  [8] 30.90509 33.52118 32.27010 32.80271 31.62138 33.57886 33.92941
## [15] 32.72060 35.37126 33.93436 34.28047 35.65055 32.28933 34.10620
## [22] 34.32770 33.09206 33.49686 31.84037 33.68494 32.22728 34.11250
##
## [[10]]
##  [1] 31.85346 32.80056 31.82407 32.32722 31.23249 32.64062 33.35840
##  [8] 32.68653 31.05379 32.01334 33.92004 34.04882 31.54893 32.22593
## [15] 32.82923 32.80036 32.04733 31.73466 32.38278 33.37880 32.55285
## [22] 31.98948 32.11063 33.88516 32.15871 33.17224 33.76567 32.90464
##
## [[11]]
##  [1] 32.08529 32.51828 32.00526 33.95626 32.39402 34.32180 31.92662
##  [8] 31.82963 32.72601 32.59109 31.99145 30.76987 32.59776 31.75084
## [15] 33.79166 31.48302 33.84143 31.82744 32.89540 33.58305 31.97524
## [22] 33.19255 32.46387 31.27598 32.25730 34.40098 32.32076 31.63604
## [29] 32.50238
##
## [[12]]
##  [1] 32.39240 32.94593 33.11295 32.18910 33.13120 31.42343 30.02429
##  [8] 31.68247 31.64337 30.04950 32.59810 31.40615 31.89995 30.69489
## [15] 32.58487 33.77457 31.32266 32.00044 33.82547 32.92689 33.79390
## [22] 32.09546 31.96529 31.64888 33.37026 30.96163 32.37219 32.05909
##
## [[13]]
##  [1] 34.68323 32.10241 31.28910 32.86105 32.61533 31.89387 32.77050
##  [8] 32.40247 32.03859 33.10308 32.35181 30.99519 30.25812 32.12718
## [15] 33.96341 33.40746 33.74761 33.53775 32.93607 31.29111 32.83663
## [22] 34.09249 29.58643 31.94600 33.64600 33.12827
```

Después olvidar esto y aplicar el algoritmo para generar muestras de las distribución final de $\boldsymbol{\theta}, \boldsymbol{h}, \mu, \tau$. Estimar y verificar convergencia. Como calentamiento genere 3,140 y use las siguientes 11,325.

Algoritmo

0. Dados $\boldsymbol{x_1}, \ldots, \boldsymbol{x_m}$, dar como semillas iniciales a $\mu^{(0)}, \tau^{(0)}, \boldsymbol{h^{(0)}}$.
1. La transición de t a $t+1$ está dada por

- Para cada $i \in J_m$ generar

$$\theta_i^{(t+1)} \sim N\left(\theta_i \,\middle|\, \frac{m_i \bar{x}_i + \tau^{(t)} \mu^{(t)}}{m_i + \tau^{(t)}},\ h_i^{(t)}(m_i + \tau^{(t)})\right)$$

$$h_i^{(t+1)} \sim Ga\left(h_i \,\middle|\, \frac{2d + m_i + 1}{2},\ \beta + \frac{1}{2}\left(m_i\left(s_i + (\bar{x}_i - \theta_i^{(t+1)})^2\right) + \tau_i^{(t)}\left(\mu^{(t)} - \theta_i^{(t+1)}\right)^2\right)\right)$$

- Hacer $h_\bullet^{(t+1)} = \sum_{i=1}^{m} h_i^{(t+1)}$ y generar

$$\tau^{(t+1)} \sim Ga\left(\tau \,\middle|\, \frac{2a + m + 1}{2},\ b + \frac{1}{2}\left(\delta(\mu^{(t)} - w)^2 + \sum_{i=1}^{m} h_i^{(t+1)}\left(\theta_i^{(t+1)} - \mu^{(t)}\right)^2\right)\right)$$

$$\mu^{(t+1)} \sim N\left(\mu \,\middle|\, \frac{\delta w + \sum_{i=1}^{m} h_i^{(t+1)} \theta_i^{(t+1)}}{\delta + h_\bullet^{(t+1)}},\ \tau(\delta + h_\bullet^{(t+1)})\right)$$

2. Repetir 1 hasta obtener convergencia.

```
#Valores iniciales
mu0 <- 30
tau0 <- 1.2
h0 <- rep(1,m)

#función para generar n transiciones
#datos los valores iniciales
algo <- function(n,xij,mu0,tau0,h0,mi){

  #arreglos donde se guardarán las transiciones
  theta <- list()
  h <- list()
  tau <- list()
  mu <- list()

  #asignar valores iniciales
  tau[[1]] <- tau0
  mu[[1]] <- mu0
  h[[1]] <- h0

  for(t in 2:n){
    #theta_i^{t+1}
    theta[[t]] <- unlist(
      lapply(X = 1:m,FUN = function(i){
        rnorm(n = 1,mean = (mi[i] * mean(xij[[i]]) +
                              tau[[t-1]]*mu[[t-1]] )/( mi[i] + tau[[t-1]] ),
              sd = sqrt(1/( h[[t-1]][i]*(mi[i] + tau[[t-1]]) ))  )
      })
    )
    #h_i^{t+1}
```

```
    h[[t]] <- unlist(
      lapply(X = 1:m,FUN = function(i){
        rgamma(n = 1,shape = (2*alpha+mi[i]+1)/2,
               rate = beta + 0.5*(mi[i]*(var(xij[[i]]) + (mean(xij[[i]]) - theta[[t]][i])^2) +
                                    tau[[t-1]]*(mu[[t-1]]-theta[[t]][i])^2))
      })
    )
    hPunto <- sum(h[[t]])
    #tau^{t+1}
    tau[[t]] <- rgamma(n = 1,shape = (2*a+m+1)/2,
                       rate = b + 0.5*(delta*(mu[[t-1]] - omega)^2 +
                                         sum( h[[t]]*(theta[[t]] - mu[[t-1]]) )^2 ) )
    #mu^{t+1}
    mu[[t]] <- rnorm(n = 1,mean = (delta*omega + sum(h[[t]] * theta[[t]]))/(delta+hPunto),
                     sd = 1/sqrt(tau[[t]]*(delta + hPunto)) )
  }
  return(list(theta = theta,h = h,tau = tau,mu = mu))
}

#Generar 3140+11325
dta <- algo(n = 3140+11325,xij,mu0,tau0,h0,mi)

#Promedios de los últimos 11325 theta_i
thetaiGorro <- matrix(unlist(tail(dta$theta,11325)),byrow = T,ncol = m)
colMeans(thetaiGorro)
```

```
##  [1] 32.03743 34.00096 31.18324 33.57377 33.15820 34.37765 33.54344
##  [8] 33.18355 33.31476 32.60479 32.57349 32.24181 32.59177
```

```
#Promedios de los últimos 11325 h
hiGorro <- matrix(unlist(tail(dta$h,11325)),byrow = T,ncol = m)
colMeans(hiGorro)
```

```
##  [1] 0.56537387 0.94760057 0.09664105 1.26856683 2.78542432 0.64593454
##  [7] 1.17050562 2.41702834 0.79932082 1.49580714 1.18816752 0.92337051
## [13] 0.74356172
```

```
#Promedio de los últimos 11325 mu
muGorro <- unlist(tail(dta$mu,11325))
mean(muGorro)
```

```
## [1] 33.08359
```

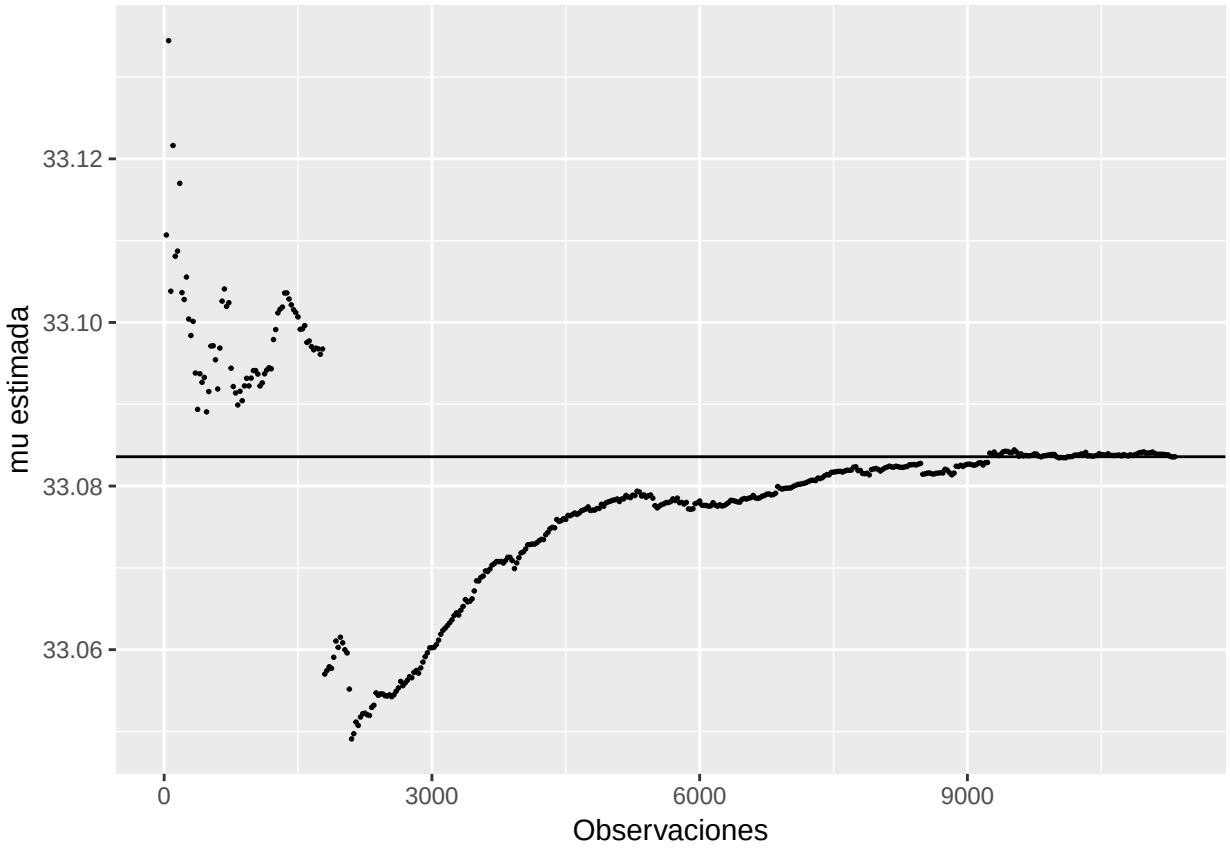
```
#Promedio de los últimos 11325 tau
tauGorro <- unlist(tail(dta$tau,11325))
mean(tauGorro)
```

```
## [1] 3.645361
```

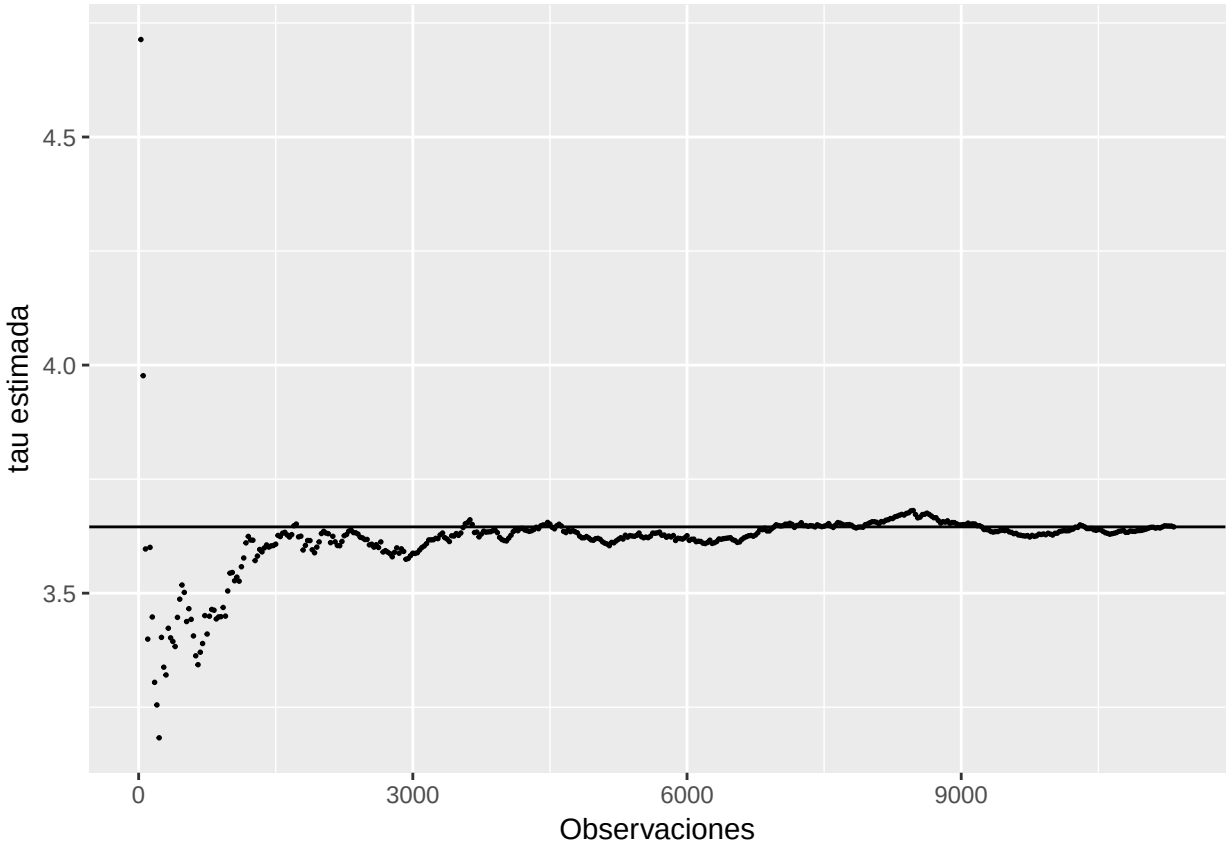
Veamos cómo los parámetros se estabilizan a medida que el número de observaciones se incrementan.

```
#Convergencia de mu
grafMu <- lapply(X = seq(25,11325,25),FUN = function(l){
  mean(muGorro[1:l])
})
grafMu = data.frame(unlist(grafMu))
colnames(grafMu) = "y"
mu =
```

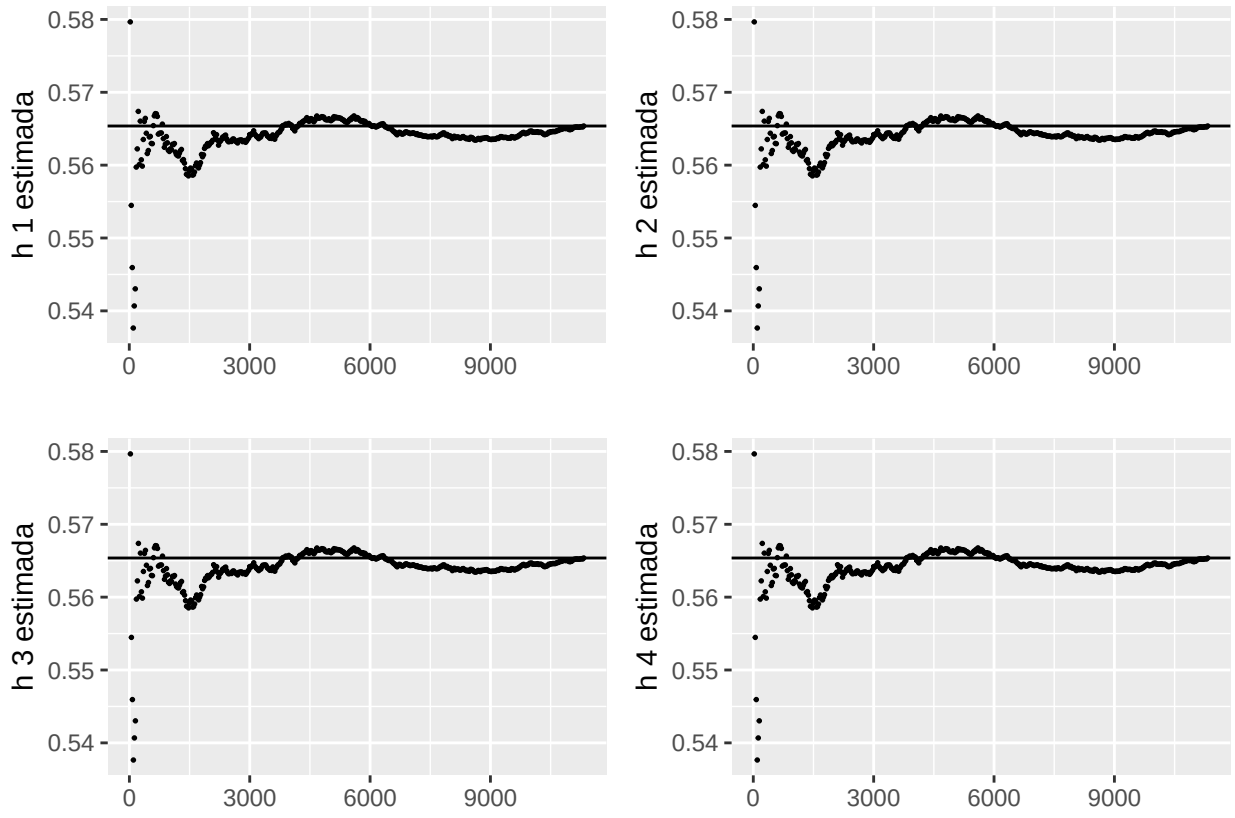
```
ggplot(data = grafMu,aes(x=seq(25,11325,25),y = y)) +
  geom_point(size = 0.3) +
  geom_hline(yintercept = mean(muGorro)) +
  xlab("Observaciones") + ylab("mu estimada")
mu
```



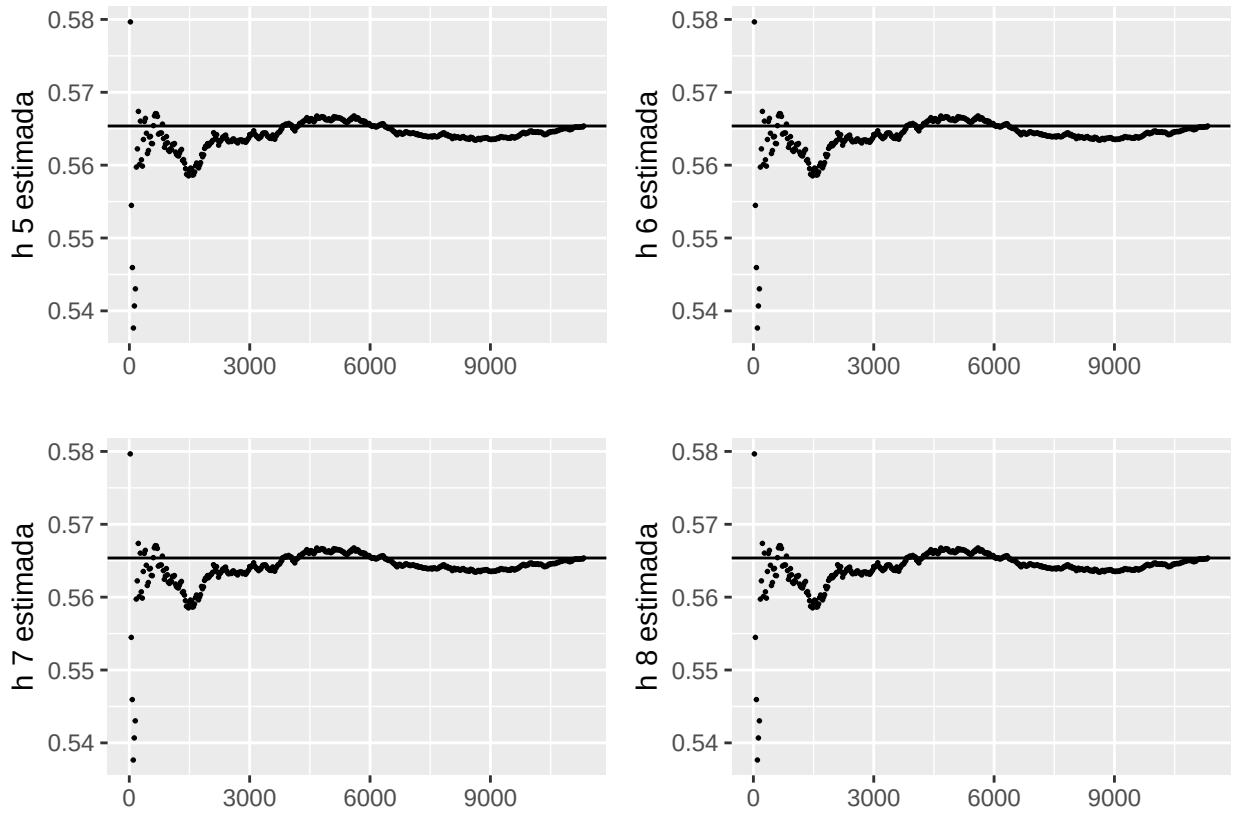

```
#Convergencia de tau
grafTau <- lapply(X = seq(25,11325,25),FUN = function(l){
  mean(tauGorro[1:l])
})
grafTau = data.frame(unlist(grafTau))
colnames(grafTau) = "y"
tau =
ggplot(data = grafTau,aes(x=seq(25,11325,25),y = y)) +
  geom_point(size = 0.3) +
  geom_hline(yintercept = mean(tauGorro)) +
  xlab("Observaciones") + ylab("tau estimada")
tau
```



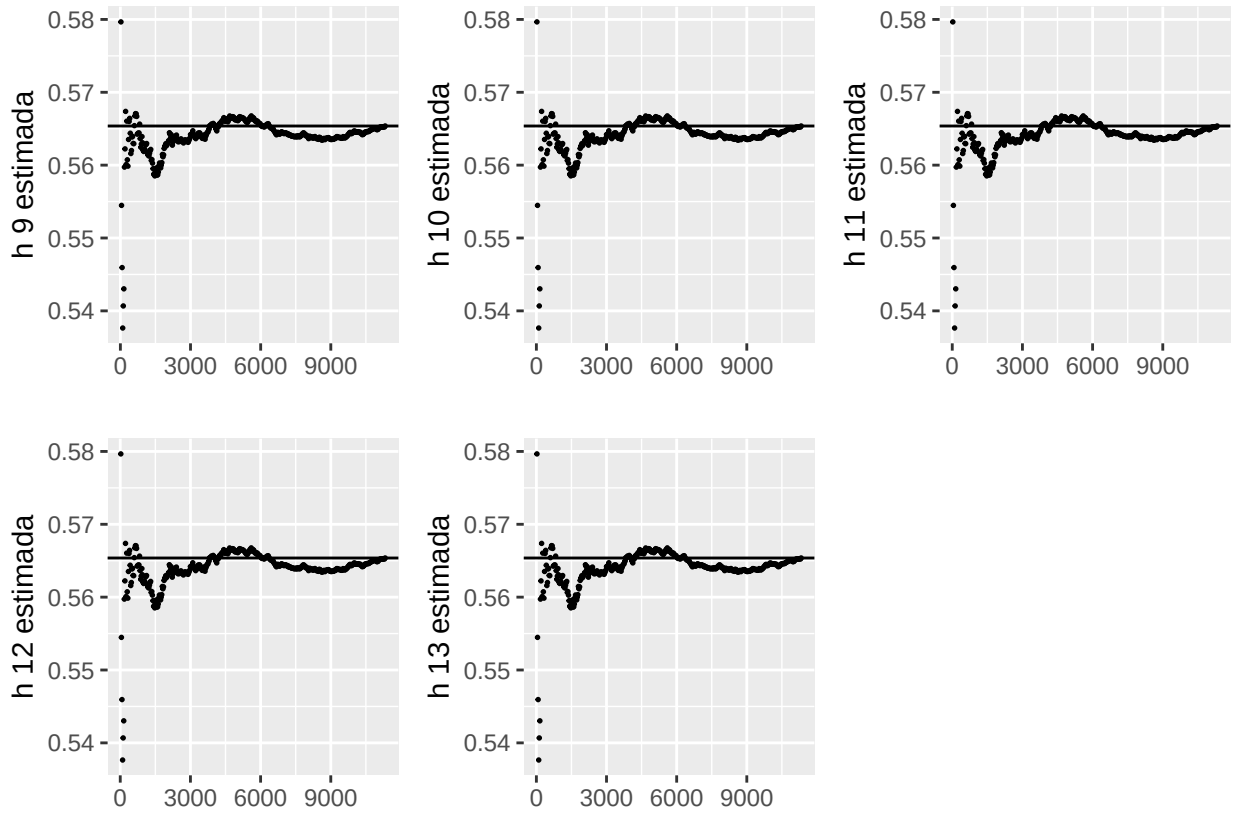

```
#Convergencia de los hi
p <- list()
for(i in 1:13){
  graf <- lapply(X = seq(25,11325,25),FUN = function(l){
    mean(hiGorro[1:l,1])
  })
  graf = data.frame(unlist(graf))
  colnames(graf) = "y"
  g =
  ggplot(data = graf,aes(x=seq(25,11325,25),y = y)) +
    geom_point(size = 0.3) +
    geom_hline(yintercept = mean(hiGorro[,1])) +
    xlab("") + ylab(sprintf("h %i estimada",i))
  p[[i]] <- g
}
grid.arrange(p[[1]],p[[2]],p[[3]],
             p[[4]],nrow = 2)
```



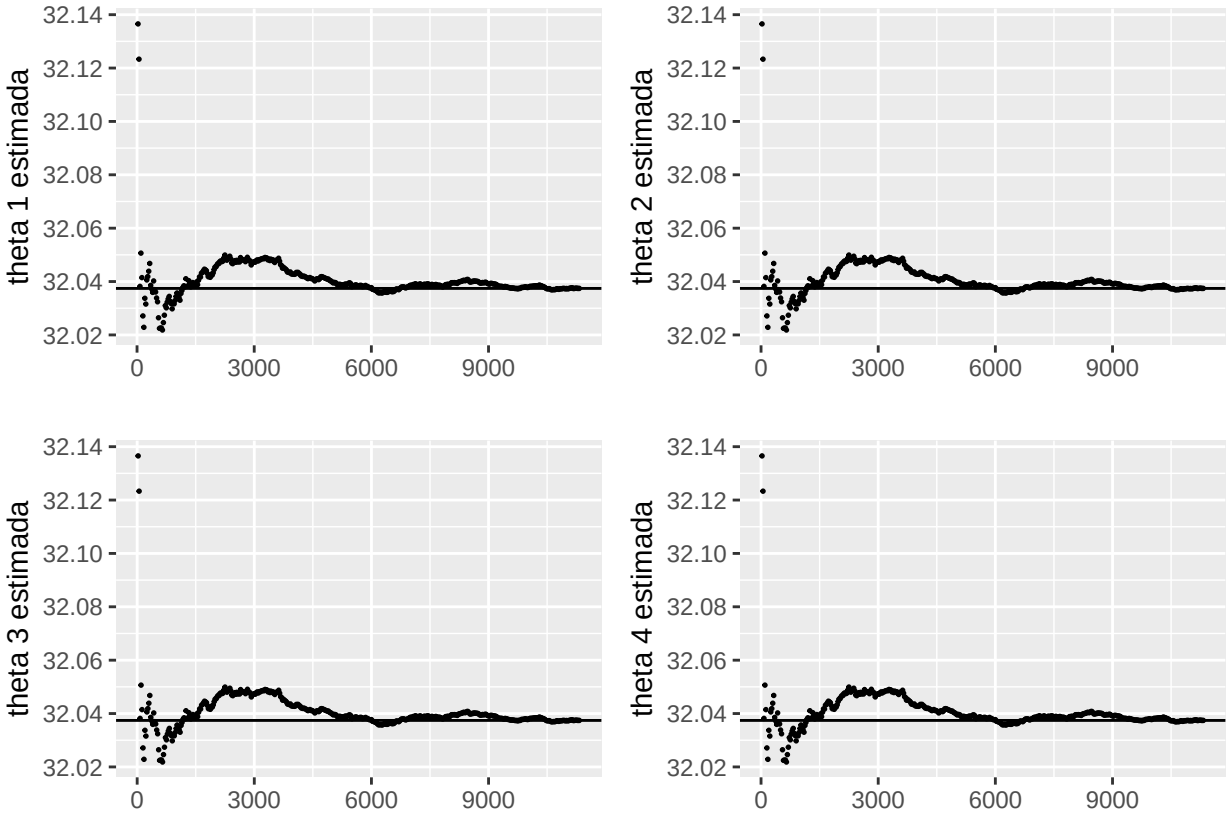

```
grid.arrange(p[[5]],p[[6]],
             p[[7]],p[[8]],nrow = 2)
```



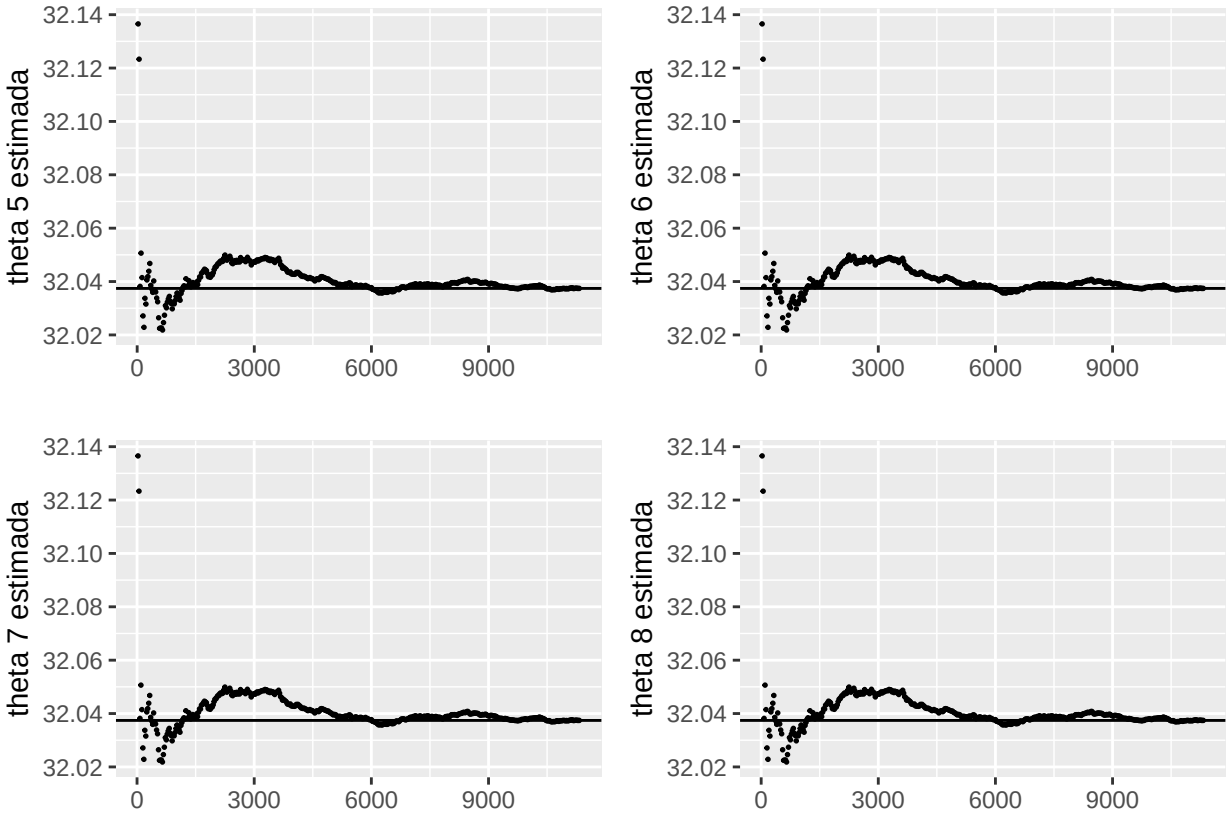

```
grid.arrange(p[[9]],
             p[[10]],p[[11]],p[[12]],
             p[[13]],nrow = 2)
```



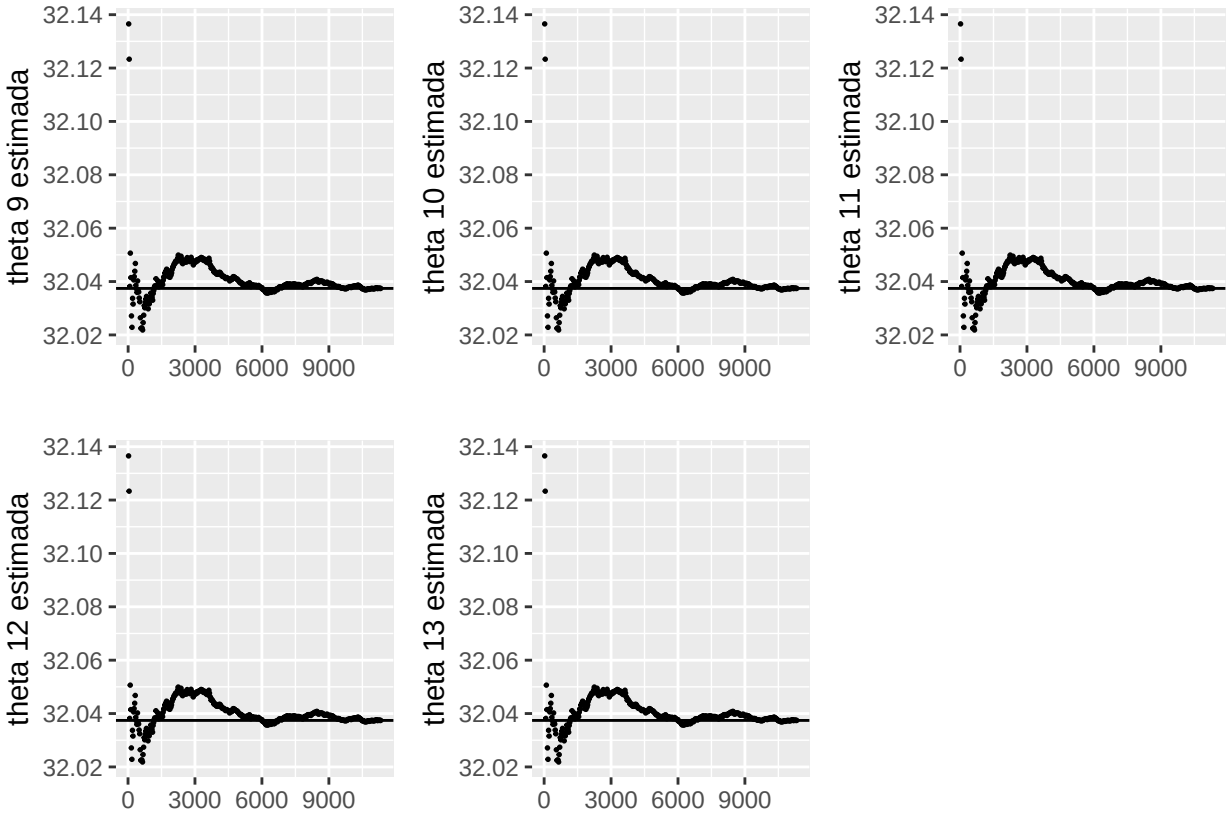

```
#Convergencia de los thetai
p <- list()
for(i in 1:13){
  graf <- lapply(X = seq(25,11325,25),FUN = function(l){
    mean(thetaiGorro[1:l,1])
  })
  graf = data.frame(unlist(graf))
  colnames(graf) = "y"
  g =
  ggplot(data = graf,aes(x=seq(25,11325,25),y = y)) +
    geom_point(size = 0.3) +
    geom_hline(yintercept = mean(thetaiGorro[,1])) +
    xlab("") + ylab(sprintf("theta %i estimada",i))
  p[[i]] <- g
}
grid.arrange(p[[1]],p[[2]],p[[3]],
             p[[4]],nrow = 2)
```




```
grid.arrange(p[[5]],p[[6]],
             p[[7]],p[[8]],nrow = 2)
```




```
grid.arrange(p[[9]],
             p[[10]],p[[11]],p[[12]],
             p[[13]],nrow = 2)
```



En todos los casos, se observa que los parámetros $\boldsymbol{\theta}, \boldsymbol{h}, \mu, \tau$ se estabilizan a medida que el número de observaciones se incrementa.