Tarea 12

Simulación estocástica

Marco Antonio Andrade Barrera

26 de abril de 2018

1. Hacer $\boldsymbol{\mu} = \mathbf{0}$. Generar 7,323 de $N_2\left(\boldsymbol{x}|\mathbf{0}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$ con $\rho \in \{0.07, 0.9143\}$. Usar las últimas 5112 y estimar $\boldsymbol{\mu}, \rho$.

En el siguiente bloque de código se encuentra una función donde se implementa el algoritmo para generar m observaciones, dado $\boldsymbol{\mu}$ y ρ.

```
m = 7323
N2 <- function(m,x0,mu = c(0,0),rho){
  x1 <- list(); x1[1] <- x0
  x2 <- list(); x2[1] <- x0
  for(t in 2:m){
    x2[t] <- rnorm(n = 1,mean = mu[2] + rho*(x1[[t-1]] - mu[1]), sd = sqrt(1-rho^2))
    x1[t] <- rnorm(n = 1,mean = mu[1] + rho*(x2[[t]] - mu[2]), sd = sqrt(1-rho^2))
  }
  data.frame(x1 = unlist(x1), x2 = unlist(x2))
}
```

Para $\rho = 0.07$.

```
set.seed(12345)
d10 <- N2(m,x0 = 3,mu = c(0,0),rho = 0.07)
#usar las últimas 5112
d1 <- tail(d10,5112)
#Estimar mu
colMeans(d1)
```

```
##           x1           x2
##  0.009616991 -0.006318365
```

```
#Estimar rho
cor(d1$x1,d1$x2)
```

```
## [1] 0.05860976
```

Observamos que el vector $\boldsymbol{\mu}$ es aproximadamente $\mathbf{0}$ y la estimación de ρ también es cercana al valor real, con una diferencia de poco más de 1 centésima.

Para $\rho = 0.9143$

```
set.seed(54321)
d20 <- N2(m,x0 = 3,mu = c(0,0),rho = 0.9143)
#usar las últimas 5112
d2 <- tail(d20,5112)
```

```r
#Estimar mu
colMeans(d2)
```

```
##          x1          x2
## 0.003107488 0.006437063
```

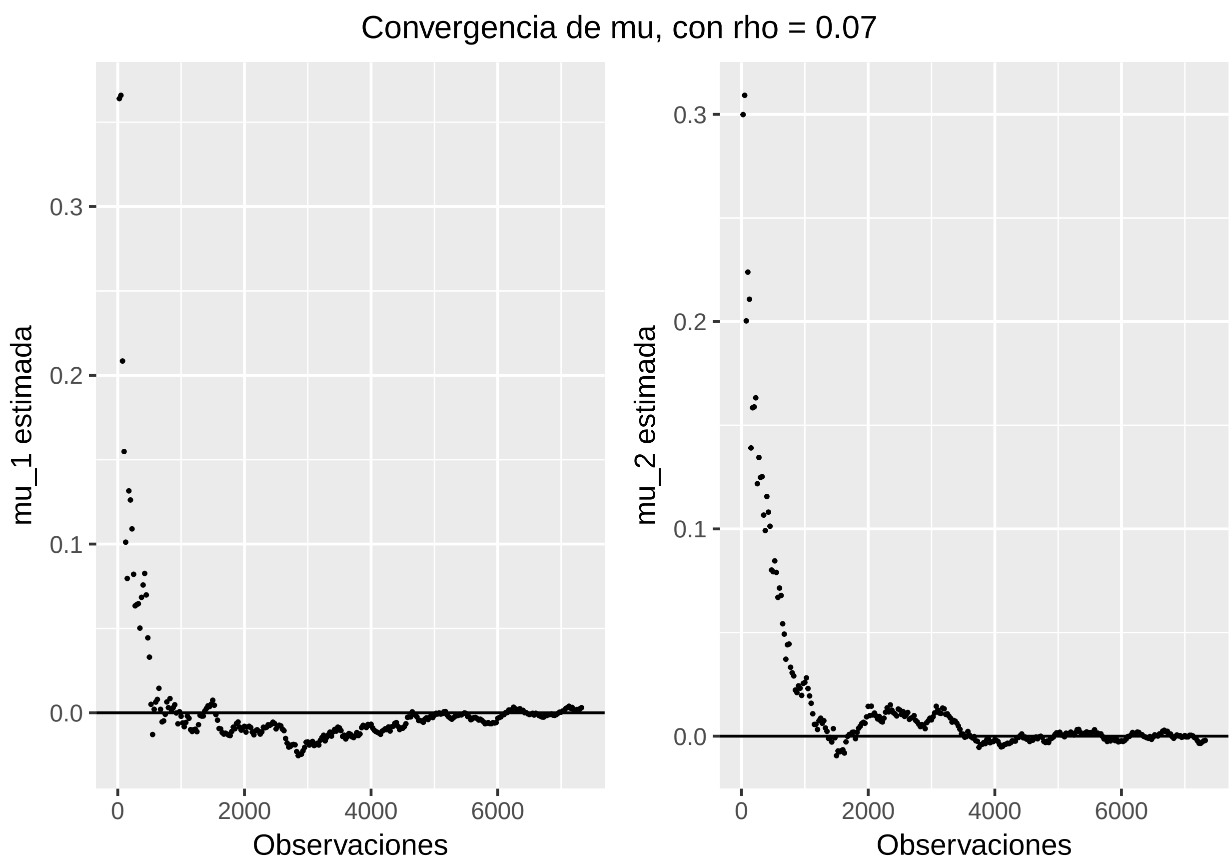
```r
#Estimar rho
cor(d2$x1,d2$x2)
```

```
## [1] 0.917658
```

Nuevamente el vector $\boldsymbol{\mu}$ es aproximadamente $\mathbf{0}$ y en este caso la estimación de ρ difiere apenas en 3 milésimas.

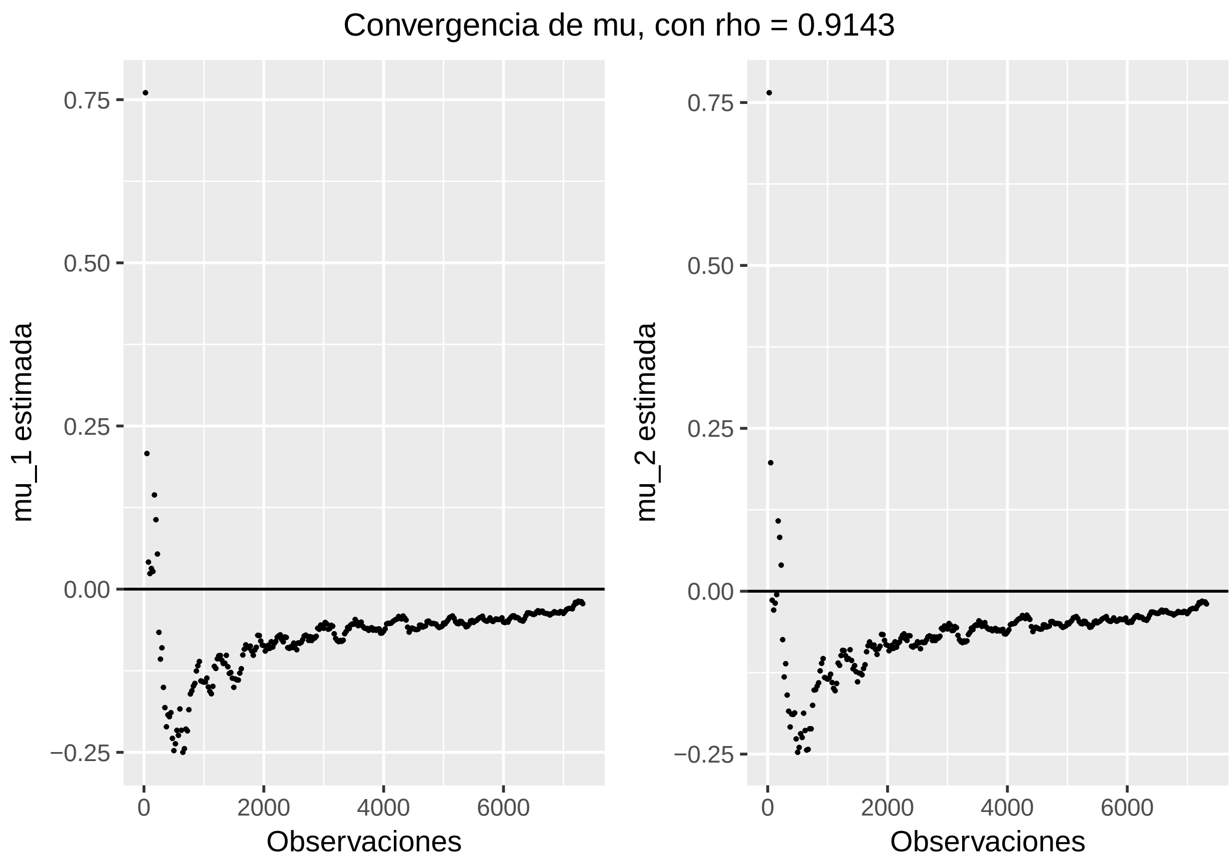
Graficar los promedios anteriores de 25 en 25 para observar la convergencia a medida que el número de observaciones se incrementa.

```r
#Mu1
#Para rho = 0.07
grafMu1 <- lapply(X = c(seq(25,m,25),m),FUN = function(l){
  mean(d10$x1[1:l])
})
grafMu1 = data.frame(unlist(grafMu1))
colnames(grafMu1) = "y"
mu_1 =
ggplot(data = grafMu1,aes(x=c(seq(25,m,25),m),y = y)) +
  geom_point(size = 0.3) +
  geom_hline(yintercept = 0) +
  xlab("Observaciones") + ylab("mu_1 estimada")

#Mu2
#Para rho = 0.07
grafMu2 <- lapply(X = c(seq(25,m,25),m),FUN = function(l){
  mean(d10$x2[1:l])
})
grafMu2 = data.frame(unlist(grafMu2))
colnames(grafMu2) = "y"
mu_2 =
ggplot(data = grafMu2,aes(x=c(seq(25,m,25),m),y = y)) +
  geom_point(size = 0.3) +
  geom_hline(yintercept = 0) +
  xlab("Observaciones") + ylab("mu_2 estimada")

grid.arrange(mu_1,mu_2,nrow = 1,top = "Convergencia de mu, con rho = 0.07")
```



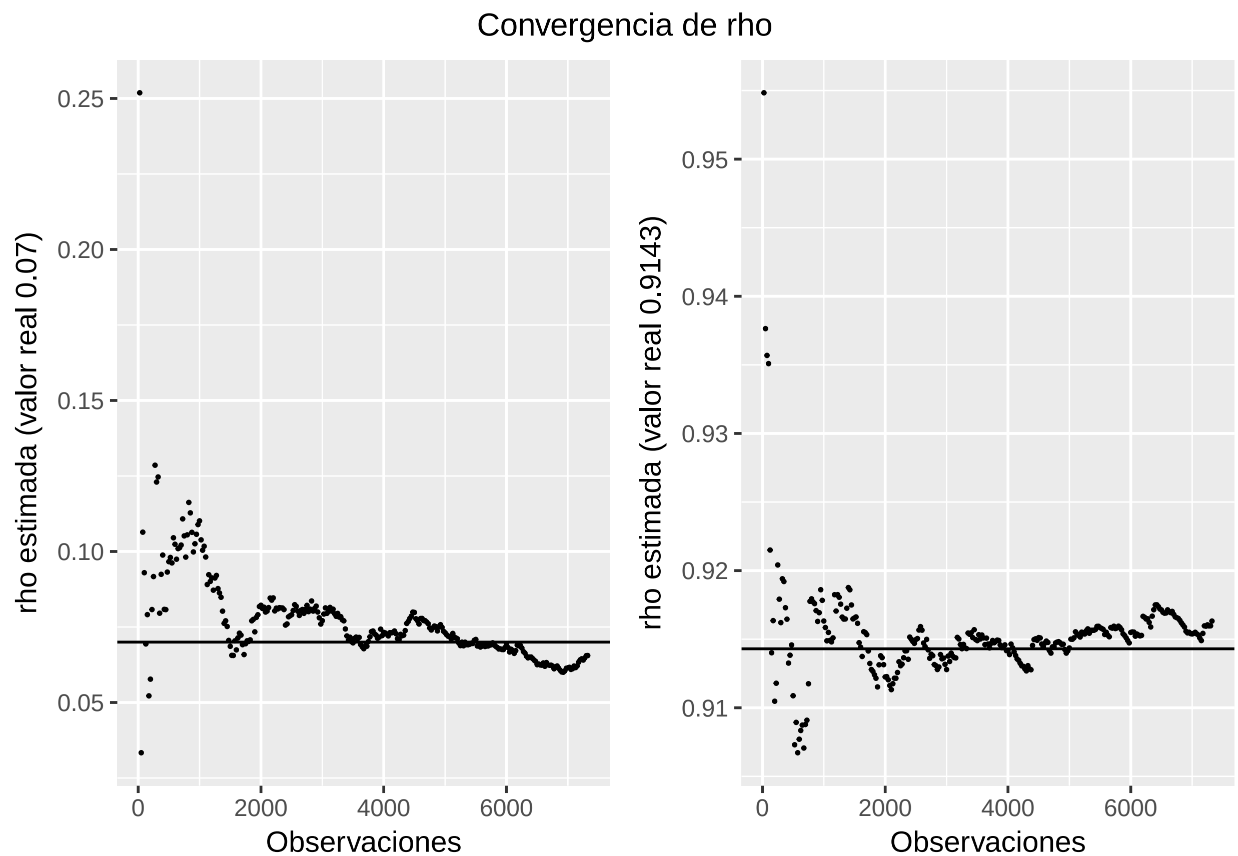

```r
#Mu1
#Para rho = 0.9143
grafMu1 <- lapply(X = c(seq(25,m,25),m),FUN = function(l){
  mean(d20$x1[1:l])
})
grafMu1 = data.frame(unlist(grafMu1))
colnames(grafMu1) = "y"
mu_1 =
ggplot(data = grafMu1,aes(x=c(seq(25,m,25),m),y = y)) +
  geom_point(size = 0.3) +
  geom_hline(yintercept = 0) +
  xlab("Observaciones") + ylab("mu_1 estimada")

#Mu2
#Para rho = 0.9143
grafMu2 <- lapply(X = c(seq(25,m,25),m),FUN = function(l){
  mean(d20$x2[1:l])
})
grafMu2 = data.frame(unlist(grafMu2))
colnames(grafMu2) = "y"
mu_2 =
ggplot(data = grafMu2,aes(x=c(seq(25,m,25),m),y = y)) +
  geom_point(size = 0.3) +
  geom_hline(yintercept = 0) +
  xlab("Observaciones") + ylab("mu_2 estimada")
```

```r
grid.arrange(mu_1,mu_2,nrow = 1,top = "Convergencia de mu, con rho = 0.9143")
```




```r
#rho
grafrho <- lapply(X = c(seq(25,m,25),m),FUN = function(l){
  cor(d10$x1[1:l],d10$x2[1:l])
})
grafrho = data.frame(unlist(grafrho))
colnames(grafrho) = "y"
rho_1 =
ggplot(data = grafrho,aes(x=c(seq(25,m,25),m),y = y)) +
  geom_point(size = 0.3) +
  geom_hline(yintercept = 0.07) +
  xlab("Observaciones") + ylab("rho estimada (valor real 0.07)")

#rho
grafrho <- lapply(X = c(seq(25,m,25),m),FUN = function(l){
  cor(d20$x1[1:l],d20$x2[1:l])
})
grafrho = data.frame(unlist(grafrho))
colnames(grafrho) = "y"
rho_2 =
ggplot(data = grafrho,aes(x=c(seq(25,m,25),m),y = y)) +
  geom_point(size = 0.3) +
  geom_hline(yintercept = 0.9143) +
  xlab("Observaciones") + ylab("rho estimada (valor real 0.9143)")
```

```
grid.arrange(rho_1,rho_2,nrow = 1,top = "Convergencia de rho")
```



Para el caso de la media, ambas gráficas (para cada uno de los valores de ρ) muestran el comportamiento esperado, es decir, a medida que el número de observaciones se incrementa, las estimaciones de $\boldsymbol{\mu}$ se aproximan a $\mathbf{0}$. El caso de ρ es un poco extraño, pues aunque efectivamente al principio la estimación parece muy alejada del valor real y posteriormente se acerca al valor verdadero (a medida que se incrementa el número de observaciones), el comportamiento es errático aunque sí está alrededor del valor verdadero.