# Tareita 15/45

Simulación estocástica

*Marco Antonio Andrade Barrera*

*3 de mayo de 2018*

Supongamos que queremos generar observaciones de la cola derecha de una normal.

## i. Implementar la solución 2. Generar $x_1, \ldots, x_{1000} \sim N(x|0,1)_{I_{[a,\infty)}(a)}$ para $a \in \{1,2,3\}$ y contar los rechazos que se generan.

Algoritmo:

1. Sean $h(x,a,\theta) = \frac{1}{\theta} \exp\{-\frac{x^2}{2} + \theta(x-a)\}$, $\theta = \frac{1}{2}(a + \sqrt{4+a^2})$ y $M = \frac{1}{\theta}\exp\{\frac{1}{2}\theta^2 - a\theta\}$
2. Generar $u_1, u_2 \sim U(u|0,1)$ y hacer $y = a - \frac{1}{\theta}\log u_2$
3. Si $u_1 \leq \frac{h(y,a,\theta)}{M}$, aceptar $x = y$; en caso contrario, ir a 2.

```r
a <- 1:3
n <- 1000

h <- function(x,a,theta){
  (1/theta)*exp(-x^2/2+theta*(x-a))
}

algo <- function(n,a){
  x <- list()
  rechazos <- 0
  i <- 1
  theta <- 0.5*(a+sqrt(4+a^2))
  M <- (1/theta)*exp(0.5*theta^2-a*theta)

  repeat{
    u <- runif(2)
    y <- a - (1/theta)*log(u[2])
    if(u[1] <= h(y,a,theta)/M){
      x[i] <- y
      i <- i+1
    } else{
      rechazos <- rechazos + 1
    }
    if(n<i) break
  }
  return(list(x = unlist(x),rechazos = rechazos))
}

set.seed(51)
#Usando a = 1
a1 <- algo(n,1)
```

```r
#Cuantiles (min, X(0.25),X(0.50),X(0.75),max)
fivenum(a1$x)
```

```
## [1] 1.000848 1.163560 1.414792 1.762930 4.015381
```

```r
#Usando a = 2
a2 <- algo(n,2)
#Cuantiles (min, X(0.25),X(0.50),X(0.75),max)
fivenum(a2$x)
```

```
## [1] 2.000245 2.118860 2.281381 2.549781 4.223043
```

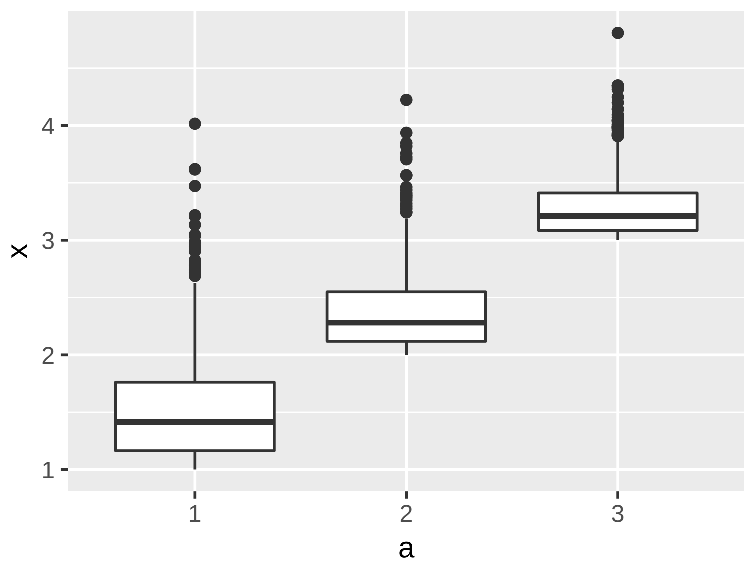
```r
#Usando  a = 3
a3 <- algo(n,3)
#Cuantiles (min, X(0.25),X(0.50),X(0.75),max)
fivenum(a3$x)
```

```
## [1] 3.000260 3.085000 3.210012 3.411499 4.807049
```

```r
#Número de rechazos usando
dta <- data.frame(a1$rechazos,a2$rechazos,a3$rechazos)
colnames(dta) <- c("1","2","3")
rownames(dta) <- "# de rechazos"
dta
```

```
##                  1  2  3
## # de rechazos 144 77 48
```

```r
#Boxplots
dta <- rbind(cbind(1,a1$x),cbind(2,a2$x),cbind(3,a3$x))
colnames(dta) <- c("a","x")
dta <- data.frame(dta)
ggplot(dta,aes(x = as.factor(a),y = x)) + geom_boxplot() + xlab("a")
```




## ii. Repetir el ejercicio con la solución 3.

Algoritmo:

1. Hacer $M = 2/ea$ y $h(x) = (1/a) \cdot x^2 \cdot exp(-x^2/2)$
2. Generar $u_1, u_2 \sim U(u|0,1)$ y hacer $y = a/u_2$
3. Si $u_1 \leq h(y)/M$ hacer $x = y$ si no regresar a 2.

```
h <- function(x,a){
  (1/a)*x^2*exp(-x^2/2)
}

algo <- function(n,a){
  x <- list()
  rechazos <- 0
  i <- 1
  M <- 2/(exp(1)*a)

  repeat{
    u <- runif(2)
    y <- a/u[2]
    if(u[1] <= h(y,a)/M){
      x[i] <- y
      i <- i+1
    } else{
      rechazos <- rechazos + 1
    }
    if(n<i) break
  }
  return(list(x = unlist(x),rechazos = rechazos))
}

set.seed(114)
#Usando a = 1
a1 <- algo(n,1)
#Cuantiles (min, X(0.25),X(0.50),X(0.75),max)
fivenum(a1$x)
```

```
## [1] 1.001170 1.177047 1.390435 1.728139 3.629603
```

```
#Usando a = 2
a2 <- algo(n,2)
#Cuantiles (min, X(0.25),X(0.50),X(0.75),max)
fivenum(a2$x)
```

```
## [1] 2.000411 2.120840 2.269441 2.522188 4.191906
```

```
#Usando  a = 3
a3 <- algo(n,3)
#Cuantiles (min, X(0.25),X(0.50),X(0.75),max)
fivenum(a3$x)
```

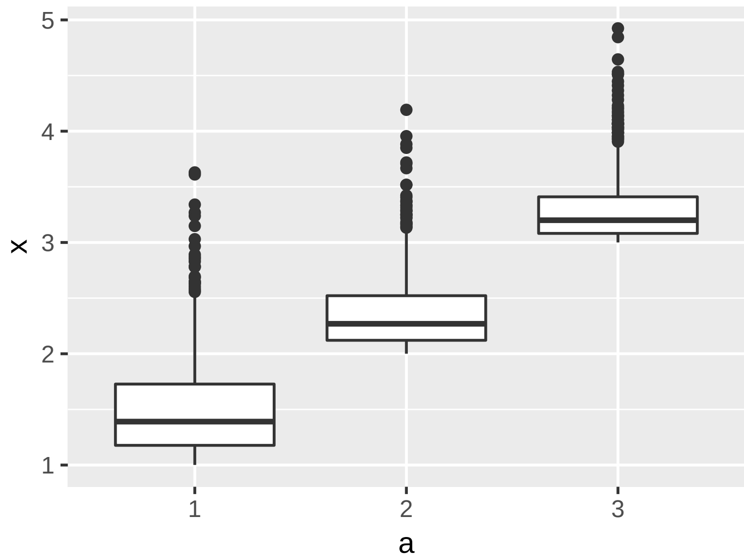
```
## [1] 3.000574 3.081630 3.199994 3.410658 4.924289
```

```
#Número de rechazos usando
dta <- data.frame(a1$rechazos,a2$rechazos,a3$rechazos)
colnames(dta) <- c("1","2","3")
rownames(dta) <- "# de rechazos"
dta
```

```
##                  1    2     3
```

```
## # de rechazos 881 5650 74410
```

```
#Boxplots
dta <- rbind(cbind(1,a1$x),cbind(2,a2$x),cbind(3,a3$x))
colnames(dta) <- c("a","x")
dta <- data.frame(dta)
ggplot(dta,aes(x = as.factor(a),y = x)) + geom_boxplot() + xlab("a")
```




Notemos como la solución 3 genera muchísimos más rechazos que la solución 2.

## iii. Hacer $x^{(0)} = a.579$ en la solución 4 y generar $x_1, \ldots, x_{1231}$, tirar las primeras 231.

Algoritmo:

0. Dado a, dar $y^{(0)}$
1. La transición de t a t+1, será

$$x^{(t+1)} \sim U\left(x \mid a, \sqrt{-2\log y^{(t)}}\right)$$

$$y^{(t+1)} \sim U\left(y \mid 0, \exp\left\{-\frac{x^{(t+1)2}}{2}\right\}\right)$$

2. Repetir 1 hasta convergencia.

```
x0 <- a + 0.579

algo <- function(n,x0,a){
  x <- list()
```

```r
  y <- list()
  x[[1]] <- x0
  for(i in 2:n){
    y[[i]] <- runif(n = 1,min = 0,max = exp(-x[[i-1]]^2/2))
    x[[i]] <- runif(n = 1,min = a,max = sqrt(-2*log(y[[i]])))
  }
  return(unlist(x))
}

#Generar 1231
set.seed(162)

#a = 1
a1 <- algo(n = 1231,x0=1.579,1)
#tirar primeras 231
a1 <- a1[-(1:231)]
#Cuantiles (min, X(0.25),X(0.50),X(0.75),max)
fivenum(a1)
```

```
## [1] 1.000375 1.189698 1.418437 1.748301 3.465250
```

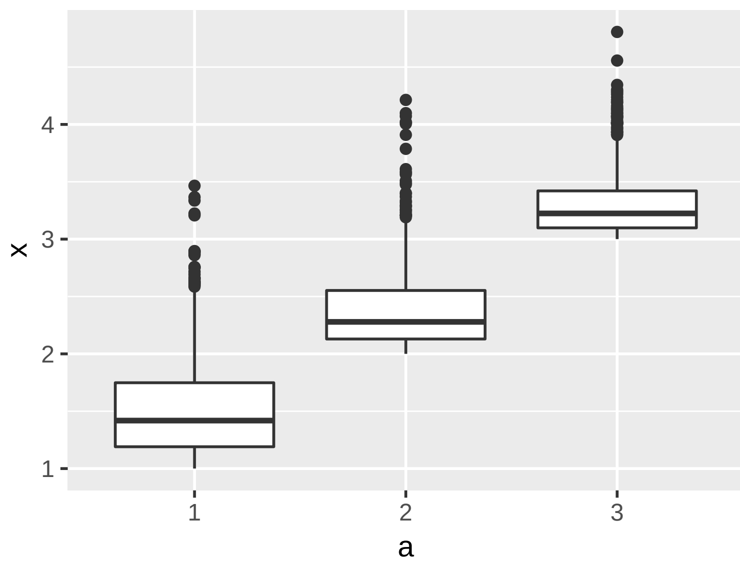
```r
#a = 2
a2 <- algo(n = 1231,x0=2.579,2)
#tirar primeras 231
a2 <- a2[-(1:231)]
#Cuantiles (min, X(0.25),X(0.50),X(0.75),max)
fivenum(a2)
```

```
## [1] 2.000329 2.129447 2.278728 2.552647 4.213689
```

```r
#a = 3
a3 <- algo(n = 1231,x0=3.579,3)
#tirar primeras 231
a3 <- a3[-(1:231)]
#Cuantiles (min, X(0.25),X(0.50),X(0.75),max)
fivenum(a3)
```

```
## [1] 3.000377 3.098289 3.224200 3.421534 4.806174
```

```r
#Boxplots
dta <- rbind(cbind(1,a1),cbind(2,a2),cbind(3,a3))
colnames(dta) <- c("a","x")
dta <- data.frame(dta)
ggplot(dta,aes(x = as.factor(a),y = x)) + geom_boxplot() + xlab("a")
```




Por los resultados anteriores, parece que la solución 2 es con la que se obtienen las observaciones de forma más rápida, pues no se requieren más allá de 144 rechazos (para el caso $a = 1$), en cambio la solución 3 lleva los rechazos al orden de los miles. En la caso de la cuarta solución pusimos un número fijo de rechazos.

En todos los casos las gráficas de caja nos muestran cómo los mínimos generados efectivamente están muy cercanos al valor de a. También, a medida que a es pequeño, el cuerpo de la caja es más amplio (pues se está más cerca de la media de la distribución).