Exame Parcial 2 Solución

Jorge de la Vega

24 de noviembre de 2018

Pregunta 1

- Aplicar el algoritmo de Metropolis-Hastings para simular 500 observaciones de la distribución doble exponencial con densidad:
$$f(x) = \frac{\lambda}{2}e^{-\lambda|x|}, \quad x \in \mathbb{R}$$
  Usar la distribución normal como distribución propuesta.

Solución.

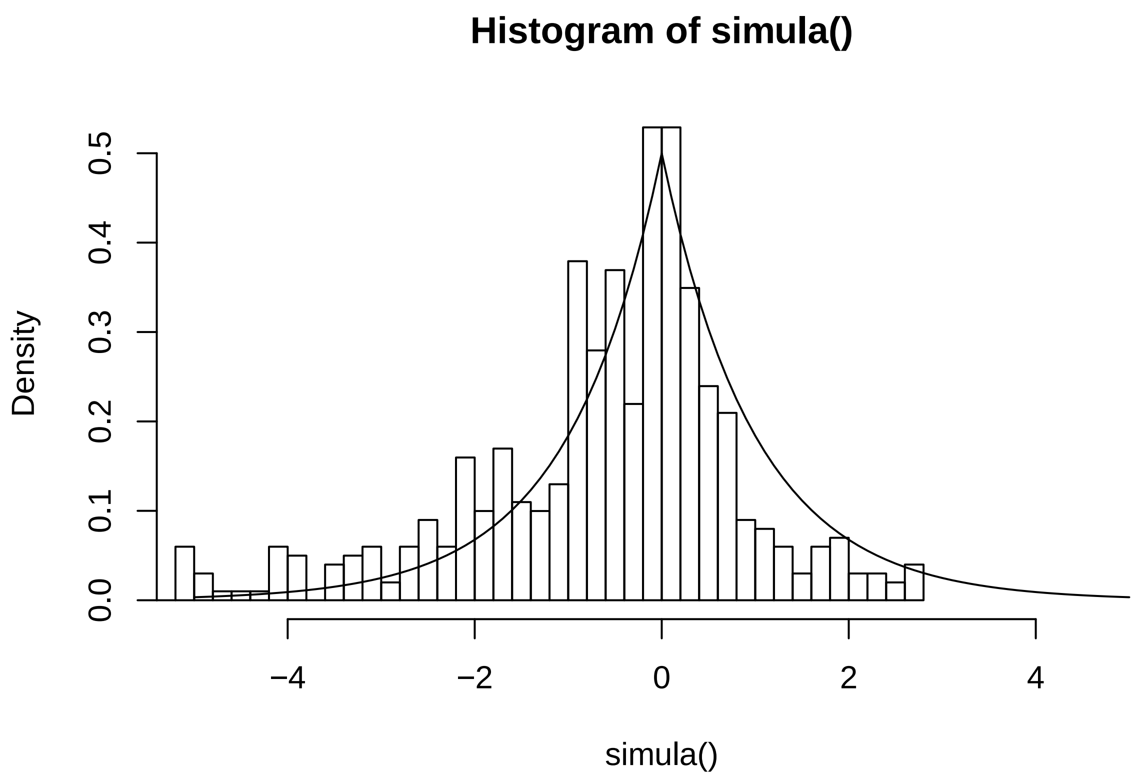
La distribución objetivo es $\pi(x) = f(x)$ y el kernel propuesto debe ser la distribución normal, entonces podemos tomar $y|x \sim N(x, 1)$ (la varianza puede ser cualquier número positivo). En este caso, como se vió en clase, la probabilidad $\alpha(x, y)$ de transición se simplifica, quedando $\alpha(x, y) = \min\left(1, \frac{\pi(y)}{\pi(x)}\right)$. Podemos entonces calcular los valores del siguiente modo:

```
f <- function(x,lambda=1){(lambda/2)*exp(-lambda*abs(x))}
simula <- function(n = 500){
  x <- runif(1) #iniciamos con un valor arbitrario en (0,1)
  for(i in 1:n){
      y <- rnorm(1,mean=x[i])
      alfa <- min(1,f(y)/f(x[i]))
      x <- append(x,ifelse(runif(1) < alfa,y,x[i]))
  }
  return(x)
}
set.seed(1)

head(simula())
```

```
## [1]  0.2655087 -0.0607247 -0.0607247  0.3539167 -0.4665516 -0.7612721
```

```
hist(simula(),breaks=40,prob=T,xlim=c(-5,5))
curve(f,from=-5,to=5,add=T)
```



□

- Comprobar estadísticamente con un nivel de confianza del 95% que la muestra obtenida proviene de las distribución indicada.

Solución.

Para hacer la prueba de bondad de ajuste, podemos utilizar varias opciones, pero la más recomendada para el caso continuo es la prueba de Kolmogorov-Smirnov, como hemos visto, es mejor para probar hipótesis $H_0 : F(x) = G(x)$, con funciones de distribución continuas. Podemos complementar el análisis haciendo un qqplot. Para ambas cosas, necesitamos una muestra de una doble exponencial, que sepamos que es correcta.

Podemos comparar las distribuciones considerando que si $X \sim exp(1)$ entonces $|X|$ es doble exponencial.

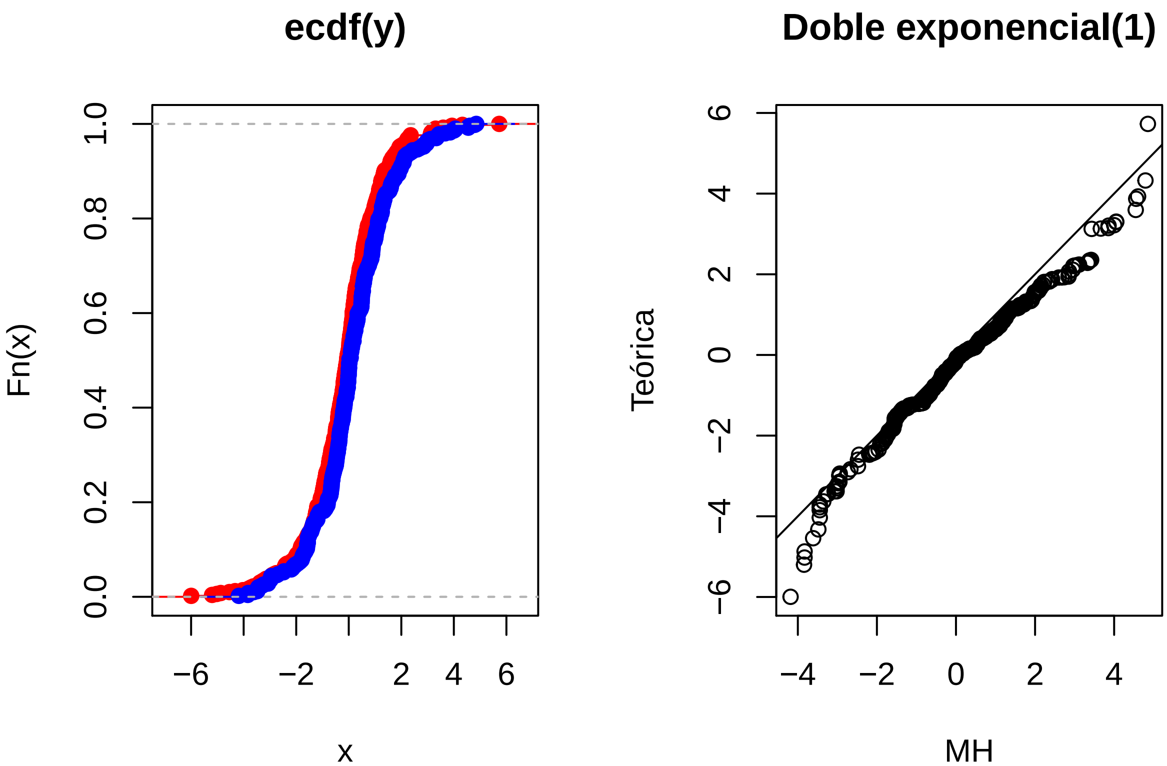
```
set.seed(1)
y <- rexp(500)-rexp(500) #muestra de doble exponencial con lambda=1
x <- simula()  #muestra generada por MH
ks.test(abs(x),"pexp",alternative = "two.sided")
```

```
## Warning in ks.test(abs(x), "pexp", alternative = "two.sided"): ties should
## not be present for the Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  abs(x)
## D = 0.069784, p-value = 0.0152
## alternative hypothesis: two-sided
```

```
par(mfrow=c(1,2))
plot(ecdf(y),col="red")
```

```
plot(ecdf(x),col="blue",add=T)
qqplot(x,y,xlab="MH",ylab="Teórica",main="Doble exponencial(1)")
abline(a=0,b=1)
```



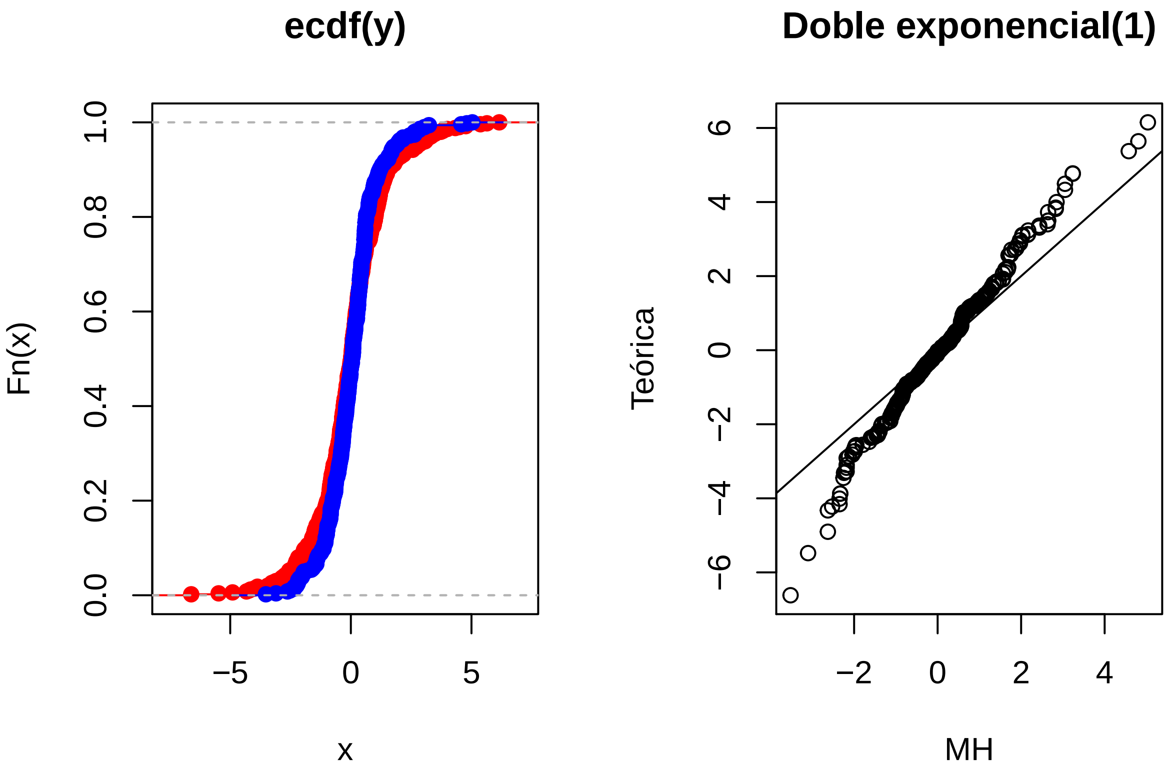
Como el p-value es cercano a cero, entonces la prueba de igualdad se rechaza al 95%. Esto se debe posiblementa a que la cadena de del algoritmo no se ha mezclado suficiente. Podemos repetir el ejercicio, por ejemplo, generando más observaciones, digamos $n = 50000$, y tomando las últimas 500 observaciones. También hay que tomar en cuenta que la muestra de los 'valores teóricos' también es una muestra, que puede agregar al ruido de la estimación.

```
y <- rexp(500)-rexp(500) #muestra de doble exponencial con lambda=1
x <- simula(n=50000)[49501:50000]  #muestra generada por MH
ks.test(x,y,alternative = "two.sided")
```

```
## Warning in ks.test(x, y, alternative = "two.sided"): p-value will be
## approximate in the presence of ties

##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  x and y
## D = 0.088, p-value = 0.04163
## alternative hypothesis: two-sided
```

```
plot(ecdf(y),col="red")
plot(ecdf(x),col="blue",add=T)
par(mfrow=c(1,2))
plot(ecdf(y),col="red")
plot(ecdf(x),col="blue",add=T)
qqplot(x,y,xlab="MH",ylab="Teórica",main="Doble exponencial(1)")
abline(a=0,b=1)
```



Vemos que los las principales diferencias se dan en las colas de la distribución.

□

Pregunta 2

Supongan que $V \sim \exp(1)$ y consideren que dado $V = v, W \sim \exp(1/v)$ (entonces, $\mathrm{E}(W|V = v) = v$). Describir un algoritmo para estimar $P(VW \leq 3)$, que solo requiera generar una variable aleatoria por muestra. Programar el algoritmo y mostrar que funciona, generando 100 muestras.

Solución.

Este ejercicio da el parámetro en términos de la tasa (que es la inversa de la media) Podemos escribir:

$$\begin{aligned}
\theta = P(VW \leq 3) &= P(W \leq 3/V) \\
&= \int_0^\infty P(W \leq 3/v | V = v) f_V(v)\, dv \\
&= \int_0^\infty F_{W|V}(3/v) f_V(v)\, dv \\
&= \int_0^\infty 1 - \exp(-3/v^2) f_V(v)\, dv \\
&= \mathrm{E}_V[1 - \exp(-3/v^2)]
\end{aligned}$$

Entonces, basta que obtengamos una muestra de exponenciales con media 1 para calcular esta probabilidad

```r
n <- 100
v <- rexp(n)
theta <- mean(1-exp(-3/v^2))
theta
```

```
## [1] 0.8509088
```

Comparando con un caso en donde generemos dos variables aleatorias:

```r
n <- 100
v <- rexp(n)
w <- sapply(v,function(x)rexp(1,rate = x))
length(which(v*w <= 3))/n
```

```
## [1] 0.95
```

Las discrepancias están asociadas al pequeño tamaño de muestra

□

Pregunta 3

Probar que si se elige a la distribución candidata como una caminata aleatoria en el algoritmo de Metropolis-Hastings, entonces $\frac{q(y|x)}{q(x|y)}$ es de la forma $h(|y-x|)$ para alguna función h.

Solución.

Si se elige una caminata aleatoria, entonces sabemos que $y = x + \epsilon$ donde $\epsilon \sim g$ tiene una distribución independiente de x.

Entonces podemos escribir al kernel de transición $q(y|x) = q(x+\epsilon|x) = q(x+(y-x)|x)$. Podemos suponer sin pérdida de generalidad que $y > x$. Entonces también podemos escribir $q(y|x) = q(x+|y-x||x)$ y $q(x|y) = q(y-\epsilon|y) = q(y-(y-x)|y) = q(y-|y-x||y)$.

Así que $\frac{q(y|x)}{q(x|y)} = \frac{q(x+|y-x||x)}{q(y-|y-x||y)} = h(|y-x|)$.

□

Pregunta 4

Si $\hat{\theta}_1$ y $\hat{\theta}_2$ son cualesquiera dos estimadores insesgados de θ, encontrar el valor de c^* que minimiza la varianza del estimador $\hat{\theta}_c = c\hat{\theta}_1 + (1-c)\hat{\theta}_2$.

Solución.

Consideremos $f(x) = \text{Var}(\hat{\theta}_c) = x^2\text{Var}(\hat{\theta}_1) + (1-x)^2\text{Var}(\hat{\theta}_2) + 2x(1-x)\text{Cov}(\hat{\theta}_1, \hat{\theta}_2)$. Hagamos $\text{Var}(\hat{\theta}_1) = A$, $\text{Var}(\hat{\theta}_2) = B$ y $\text{Cov}(\hat{\theta}_1, \hat{\theta}_2) = C$

Entonces $f(x) = Ax^2 + B(1-x)^2 + 2x(1-x)C$. Tomando la primera derivada e igualando a 0, $f'(x) = 2Ax - 2B(1-x) + 2C(1-2x) = 0$ y despejando x tenemos:

$$c^* = \frac{B-C}{A+B-2C} = \frac{\text{Var}(\hat{\theta}_2) - \text{Cov}(\hat{\theta}_1, \hat{\theta}_2)}{\text{Var}(\hat{\theta}_1 - \hat{\theta}_2)} = \frac{\text{Var}(\hat{\theta}_2)(1 - \text{Var}(\hat{\theta}_1)\rho^2)))}{\text{Var}(\hat{\theta}_1 - \hat{\theta}_2)}$$

Este valor de c^* es claramente un mínimo, por la desiguadad de Cauchy-Schwarz.

□

Pregunta 5

Encontrar dos funciones de importancia f_1 y f_2 que tengan soporte en $(1, \infty)$ y estén 'cerca' de

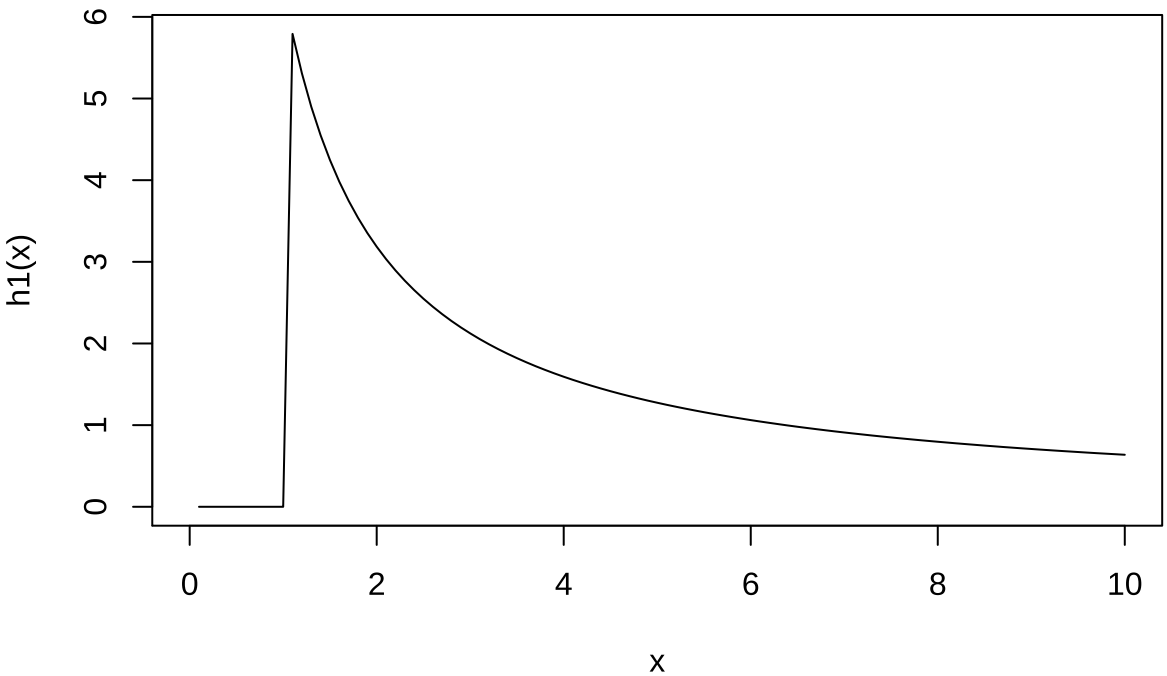
$$g(x) = \frac{x^2}{\sqrt{2\pi}} \exp(-x^2/2)\, dx, \quad x > 1$$

Solución.

La propuesta es usar dos funciones de densidad de tal manera que el cociente $h(x) = f_i/g(x)$ esté acotado

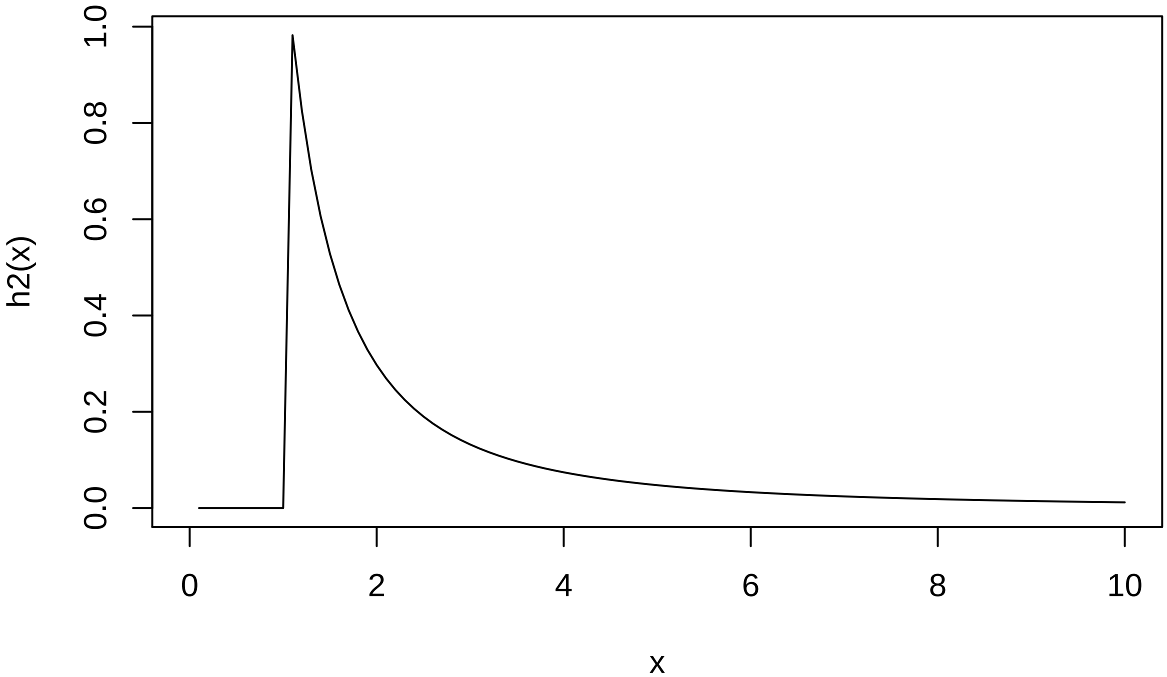
Una posibilidad es usar directamente $f_1(x) = xe^{-x^2/2}$. Esta es la densidad de una distribución de Rayleigh, y si consideramos que $\int_0^1 f_1(x)dx = 1 - e^{-1/2}$, entonces podemos reescalar para quedar en el soporte que queremos:

```
f1 <- function(x)x*exp(-x^2/2)*ifelse(x>1,1,0)/(1-exp(-1/2))
g <- function(x)(1/sqrt(2*pi))*x^2*exp(-x^2/2)
h1 <- function(x)f1(x)/g(x)
curve(h1,from=0,to=10)
```



La segunda función que podemos tomar es una normal truncada, $f_2(x) = \frac{1}{\sqrt{2\pi}\Phi(1)}$

```
f2 <- function(x)exp(-x^2/2)*ifelse(x>1,1,0)/(sqrt(2*pi)*pnorm(1))
h2 <- function(x)f2(x)/g(x)
curve(h2,from=0,to=10)
```



Ambas funciones pueden utilizarse para resolver la integral utilizando muestreo de importancia.

□

Pregunta 6

Consideren una distribución Poisson con parámetro $\lambda = 3$ condicionada a que no sea 0. Implementar un algoritmo MCMC para simular de esta distribución, usando una distribución propuesta que sea geométrica con parámetro $p = 1/3$. Usar la simulación para estimar la media y la varianza.

Solución.

La variable a considerar es $Y|Y \geq 1$ con función de masa de probabilidad $f(y) = \frac{e^{-3}3^y I(y \geq 1)}{y!(1-e^{-3})}$. Además, se pide que el kernel de transición sea una geométrica, por lo que el kernel es independiente de x: $q(\cdot|x) = geo(1/3)$.

```
f <- function(x){exp(-3)*3^x*ifelse(x==0,0,1)/(factorial(x)*(1-exp(-3)))}

simula <- function(n){
  x <- 1 #inicializa con valor arbitrario
  for(i in 1:n){
      y <- rgeom(1,prob=1/3)
      alfa <- min(1,f(y)*dgeom(x,prob=1/3)/(f(x[i])*dgeom(y,prob=1/3)))
      x <- append(x,ifelse(runif(1) < alfa,y,x[i]))
  }
  return(x)
}

sim <- simula(1000)
head(sim,20)
```

```
##  [1] 1 1 1 3 3 3 3 3 3 3 3 3 4 4 1 4 7 7 7 7
```

```r
mean(sim)
```

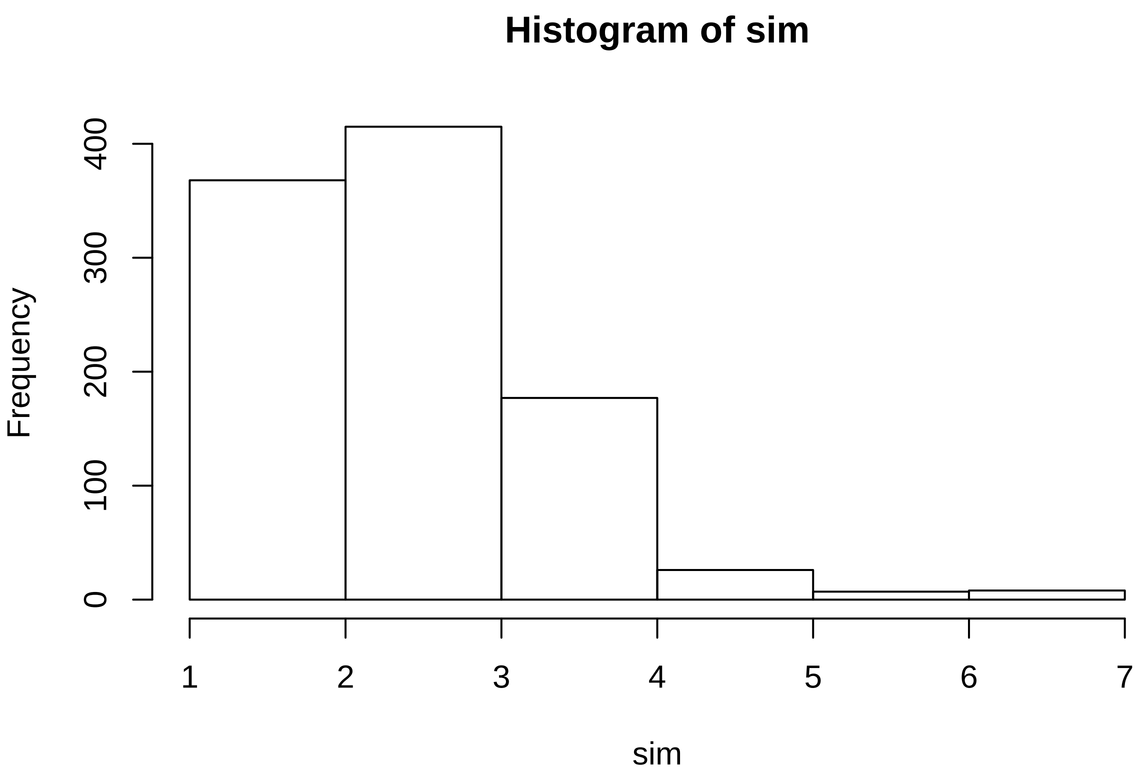
```
## [1] 2.764236
```

```r
var(sim)
```

```
## [1] 1.23436
```

Podemos ver un histograma de la distribución:

```r
hist(sim,breaks=max(sim)+1)
```



□