Simulación ITAM
6 de marzo de 2018 Jorge de la Vega
2018-I

Tarea 2. Fecha de entrega: Sábado 3 de marzo 2018

Lecturas

- Casella y Robert, capítulo 2
- Dagpunar, Capítulos 3 y 4. Secciones 7.1-7.3

Problemas

1. Proponer los algoritmos (método y pseudocódigo o código así como una corrida) para generar muestras de las siguientes densidades:

   (a) Cauchy

   $$f(x) = \frac{1}{\pi\beta\left[1+\left(\frac{x-\gamma}{\beta}\right)^2\right]}$$

   donde $\gamma, x \in \Re$, $\beta > 0$.

   (b) Gumbel (o de valor extremo)

   $$f(x) = \frac{1}{\beta}\exp\left(-e^{-(x-\gamma)/\beta} - \frac{x-\gamma}{\beta}\right)$$

   donde $\gamma, x \in \Re$, $\beta > 0$.

   (c) Logística

   $$f(x) = \frac{(1/\beta)e^{-(x-\gamma)/\beta}}{(1+e^{-(x-\gamma)/\beta})^2}$$

   donde $\gamma, x \in \Re$, $\beta > 0$.

   (d) Pareto

   $$f(x) = \frac{\alpha_2 c^{\alpha_2}}{x^{\alpha_2+1}}$$

   donde $c > 0, \alpha_2 > 0, x > c$.

   Para $\gamma = 0$ y $\beta = 1$ en cada inciso (a), (b) y (c), usen los algoritmos que obtuvieron para generar una muestra aleatoria de 5000 valores y obtengan $\bar{X}(n) = \sum_{i=1}^{n} X_i/n$ para $n = 50, 100, 150, \ldots, 5000$ para verificar empíricamente la ley fuerte de los grandes números. Hacer lo mismo para (d) con $c = 1$ y $\alpha_2 = 2$.

   Solución.

   (a) Este caso ya lo hicimos como ejemplo de clase, con una distribución Cauchy estandarizada. Basta con estandarizar la variable $Z = \frac{X-\gamma}{\beta}$. Aquí nos sirve directamente el método de la transformación inversa.

(b) La distribución Gumbel puede invertirse fácilmente. La distribución es de la forma $F(x) = e^{-e^{-\frac{x-\mu}{\beta}}}$ y entonces $X = -\beta \log(-log(u)) + \mu$. El método de la transformación inversa se puede aplicar de manera directa.

(c) La distribución logística también es muy fácil de invertir. Tiene distribución $F(x) = \frac{1}{1+e^{-(x-\mu)/\beta}}$. Entonces $X = -\beta \log(1/u - 1) + \mu$.

(d) La función de distribución está dada por $F(x) = 1 - \left(\frac{c}{x}\right)^{\alpha_2}$ Lo que resulta en la inversa $X = \frac{c}{(1-u)^{1/\alpha_2}}$.

El resto del ejercicio es un ejercicio estándar fácil de completar.

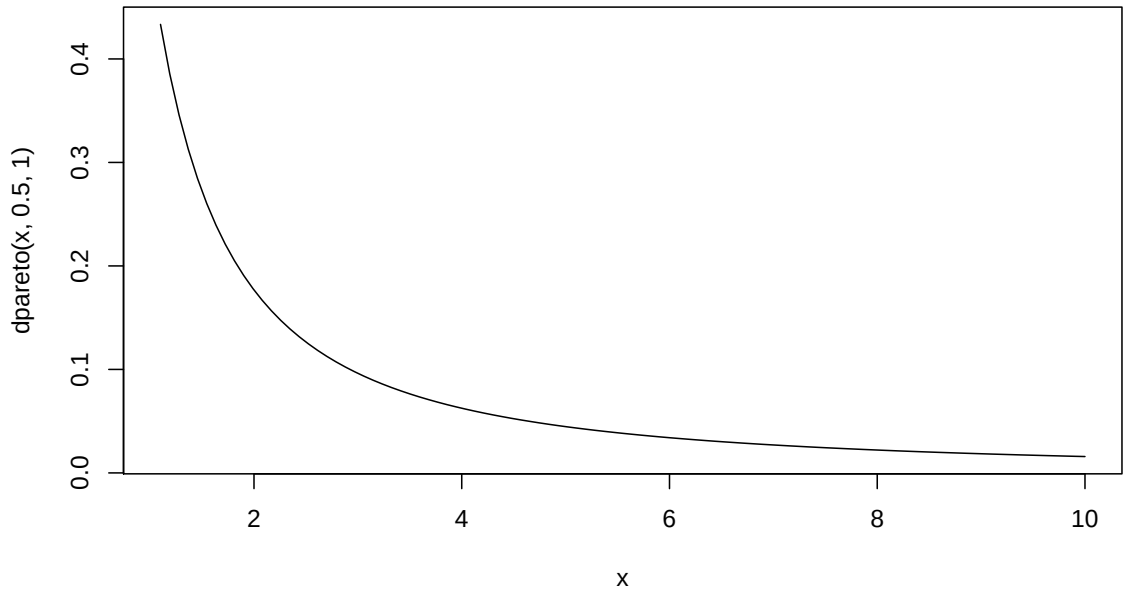
□

2. Para la distribución Pareto, crear funciones `rpareto`, `dpareto` `qpareto` y `ppareto` en R, para generar números aleatorios, obtener valores de la densidad, de la distribución y cuantíles de la familia Pareto de distribuciones. Mostrar su funcionalidad con ejemplos a través de una viñeta.

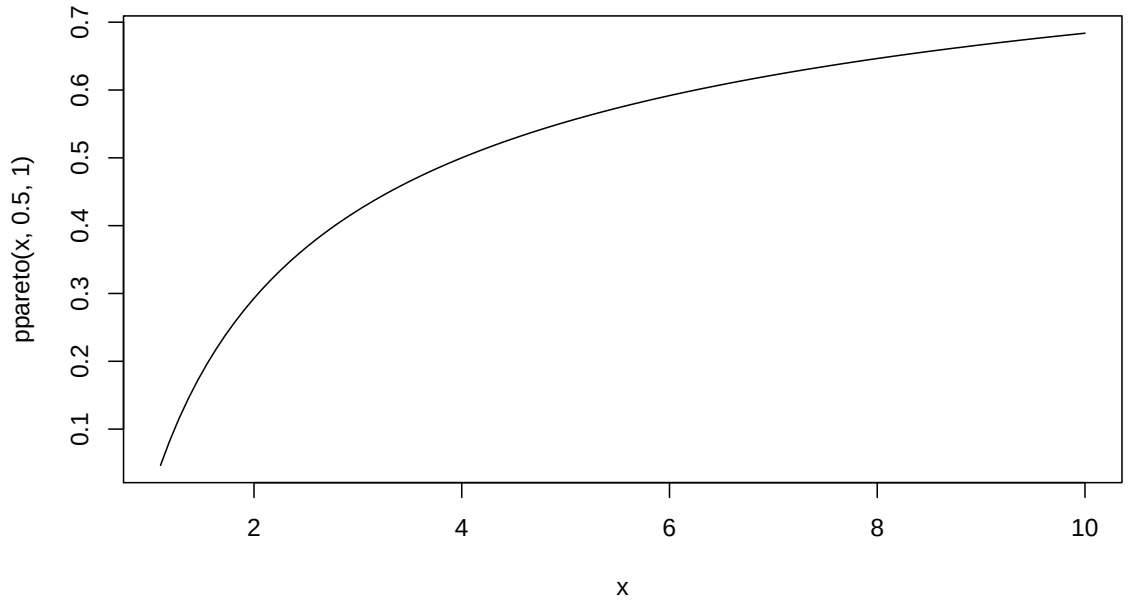
Solución.

Como las funciones de densidad y de distribución están definidas, y es fácil invertir, las tres funciones son directas del inciso (d) del ejercicio anterior.

```
rpareto <- function(n,a,c){
#función que genera muestras aleatorias de tamaño n de pareto con parámetros a y c
#a y c tienen que ser valores positivos
        if (a <= 0 || c<=0) stop("Los valores de a y c tienen que ser positivos")
        return(c/(1-runif(n))^(1/a))
}

dpareto <- function(x,a,c){
#función que genera densidad de pareto con parámetros a y c
#a y c tienen que ser valores positivos
        if (a <= 0 || c<=0) stop("Los valores de a y c tienen que ser positivos")
        return(a*c^a/x^(a+1))
}

ppareto <- function(x,a,c){
#función que genera la distribución de pareto con parámetros a y c
#a y c tienen que ser valores positivos
        if (a <= 0 || c<=0) stop("Los valores de a y c tienen que ser positivos")
        return(1-(c/x)^a)
}

qpareto <- function(p,a,c){
#función que genera muestras aleatorias de tamaño n de pareto con parámetros a y c
#a y c tienen que ser valores positivos
        if (a <= 0 || c<=0) stop("Los valores de a y c tienen que ser positivos")
        return(c/(1-p)^(1/a))
}

#Ejemplos de aplicación:

rpareto(10,2,1)


 [1] 1.149957 1.019405 1.097748 1.185089 5.143838 3.643188 1.325813
 [8] 1.172566 2.622330 1.257843


curve(dpareto(x,0.5,1),from=1.1,to=10)
```




```
curve(ppareto(x,0.5,1),from=1.1,to =10)
```




```
qpareto(c(0.5,0.6,0.7),1,1)
```

```
[1] 2.000000 2.500000 3.333333
```

□

3. Una variable aleatoria discreta X tiene función de masa de probabilidad

x	0	1	2	3	4
$p(x)$	0.1	0.2	0.2	0.2	0.3

Utilicen el teorema de la transformación inversa para generar una muestra aleatoria de tamaño 1000 de la distribución de X. Construyan una tabla de frecuencias relativas y comparen las probabilidades empíricas con las teóricas.

Repitan considerando la función de `R sample`.

Solución.

Para el primer ejercicio: haciéndolo como dice el teorema de la transformación inversa, nos auxiliamos de la función `findInterval` para hacer la búsqueda en tabla de las probabilidades acumuladas: Recordar que el teorema de la transformación inversa en el caso discreto se traduce a resolver, para cada u, si $u < 0.1$, $x = 0$, si $0.1 \leq u < 0.2$, $x = 1$, si $0.2 \leq u < 0.4$, $x = 2$, si $0.4 \leq u < 0.6$, $x = 3$ y finalmente, si $u \geq 0.6$, $x = 4$.

```
u <- runif(0:1000)
x <- findInterval(x=u,vec=cumsum(c(0.1,0.2,0.2,0.2,0.3)))
table(x)/1000
```

```
x
    0     1     2     3     4
0.119 0.179 0.194 0.182 0.327
```

La segunda parte del ejercicio es muy fácil.

```
x <- sample(0:4,1000,prob=c(0.1,0.2,0.2,0.2,0.3),replace=T)
table(x)/1000
```

```
x
    0     1     2     3     4
0.088 0.199 0.196 0.216 0.301
```

□

4. Graficar las siguientes densidades. Dar los algoritmos de transformación inversa, composición y aceptación-rechazo para cada una de las siguientes densidades. Discutir cuál algoritmo es preferible para cada densidad.

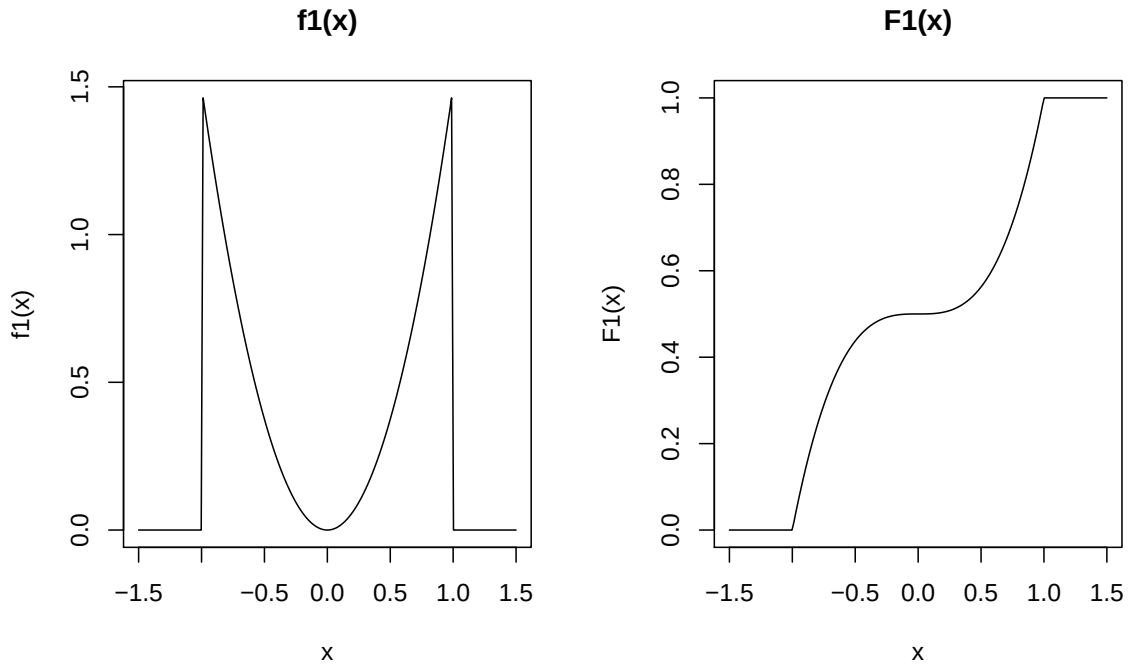
   (a)
   $$f(x) = \frac{3x^2}{2} I(x)_{[-1,1]}$$

   (b) Para $0 < a < \frac{1}{2}$,
   $$f(x) = \begin{cases} 0 & x \leq 0 \\ \frac{x}{a(1-a)} & 0 \leq x \leq a \\ \frac{1}{1-a} & a \leq x \leq 1-a \\ \frac{1-x}{a(1-a)} & 1-a \leq x \leq 1 \\ 0 & x \geq 1 \end{cases}$$

Solución.

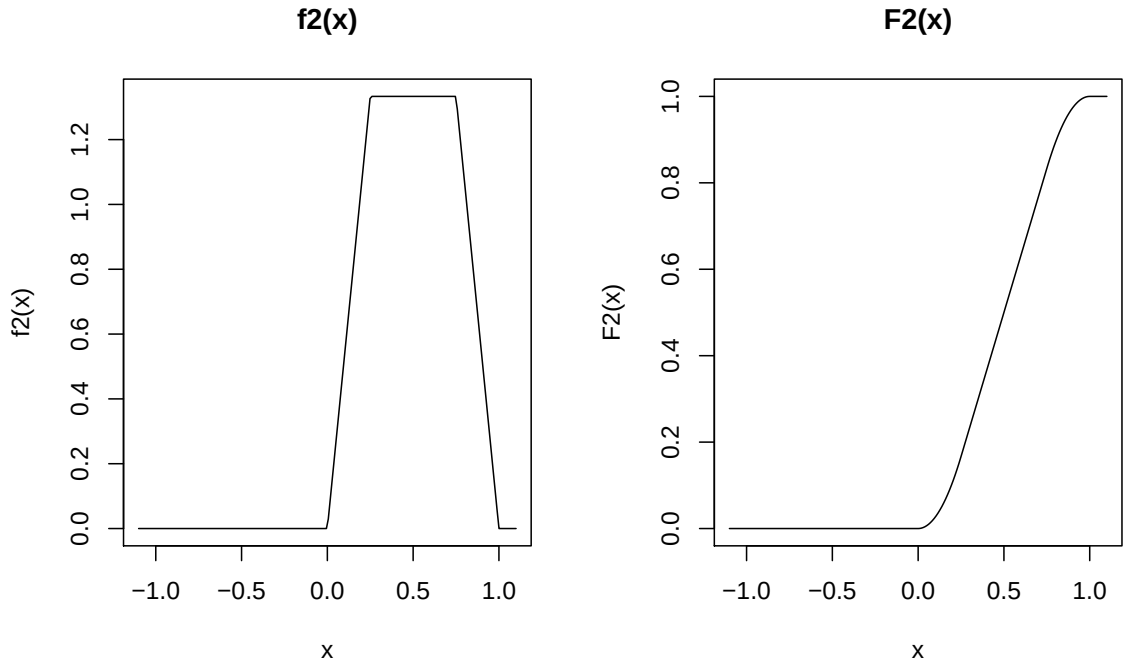
Para el inciso (a), consideremos las funciones de densidad y distribución dadas a continuación:

```
indicator <- function(x,a,b){ifelse(x <=b & x >=a,1,0)} #función indicadora en el intervalo (a,b)
f1 <- function(x){3*x^2/2*indicator(x,-1,1)}
F1 <- function(x){ifelse(x<=-1,0,ifelse(x<=1,0.5*(x^3+1),1))}
x <- seq(-1.5,1.5,length=200)  #intervalo de graficación
par(mfrow=c(1,2))
plot(x,f1(x),type="l",main="f1(x)")
plot(x,F1(x),type="l",main="F1(x)")
```



Para el inciso (b)

```
f2 <- function(x,a=0.25){indicator(x,-1,1)*(indicator(x,0,a)*(x/(a*(1-a))) +
                                            indicator(x,a,1-a)/(1-a) +
                                            indicator(x,1-a,1)*(1-x)/(a*(1-a)) )}
F2 <- function(x,a=0.25){indicator(x,0,a)*x^2/(2*a*(1-a)) +
                         (x-a/2)/(1-a)*indicator(x,a,1-a) +
                         ((1-3*a/2)/(1-a) + (x*(1-x/2)-(1-a)*(1+a)/2)/(a*(1-a)))*indicator(x,1-a,1) +
                         indicator(x,1,100)
}
par(mfrow=c(1,2))
x <- seq(-1.1,1.1,length=200)  #intervalo de graficación
plot(x,f2(x),type="l",main="f2(x)")
plot(x,F2(x),type="l",main="F2(x)")
```



□

5. Considerando la transformación polar de Marsaglia para generar muestras de normales estándar, muestren que la probabilidad de aceptación de $S = V_1^2 + V_2^2$ en el paso 2 es $\pi/4$, y encuentren la distribución del número de rechazos de S antes de que ocurra una aceptación. ¿Cuál es el número esperado de ejecuciones del paso 1?

Solución.

Para la primera parte, basta un algoritmo geométrico. La región en donde se rechazan los puntos corresponden al área sobrante del cuadrado que circunscribe el círculo con radio unitario. Esa región tiene área $\frac{4-\pi}{4} = 1 - \pi/4 = 0.215$. Entonces se rechaza 21.5 % del tiempo. Ahora bien, si X = número de rechazos antes de aceptar, sabemos que $X \sim geom(\pi/4)$. Entonces $E(X) = 1/\frac{\pi}{4} = 4/\pi \approx 1.2732395$

□

6. Obtengan una muestra de 10,000 números de la siguiente distribución discreta:

$$p(x) = \frac{2x}{k(k+1)}, x = 1, 2, \ldots, k$$

para $k = 100$.

Solución.

Noten que la función de distribución tiene la forma: $F(j) = P(X \leq j) = \frac{2}{k(k+1)} \sum_{i=1}^{j} x = \frac{2}{k(k+1)} \times \frac{j(j+1)}{2}$ ya que la suma corresponde a la suma de los primeros j naturales. Entonces podemos generar los "cajones" de la función acumulativa relativamente simple.

```r
set.seed(1)
k <- 100   #parámetro de la distribución
n <- 10000 #tamaño de muestra
Fn <- ((1:k)*(1+(1:k)) )/(k*(k+1)) #Escalones de distribución
u <- runif(10000) #bten unifotmes
muestra <- findInterval(u,Fn,left.open=T) #muestra solicitada
hist(muestra,prob=T,main="histograma de la muestra")
lines(1:100,2*(1:100)/(100*101),col="red") #función de probabilidad
points(muestra,rep(0,10000),pch="|",cex=0.5) #puntos de la muestra
```

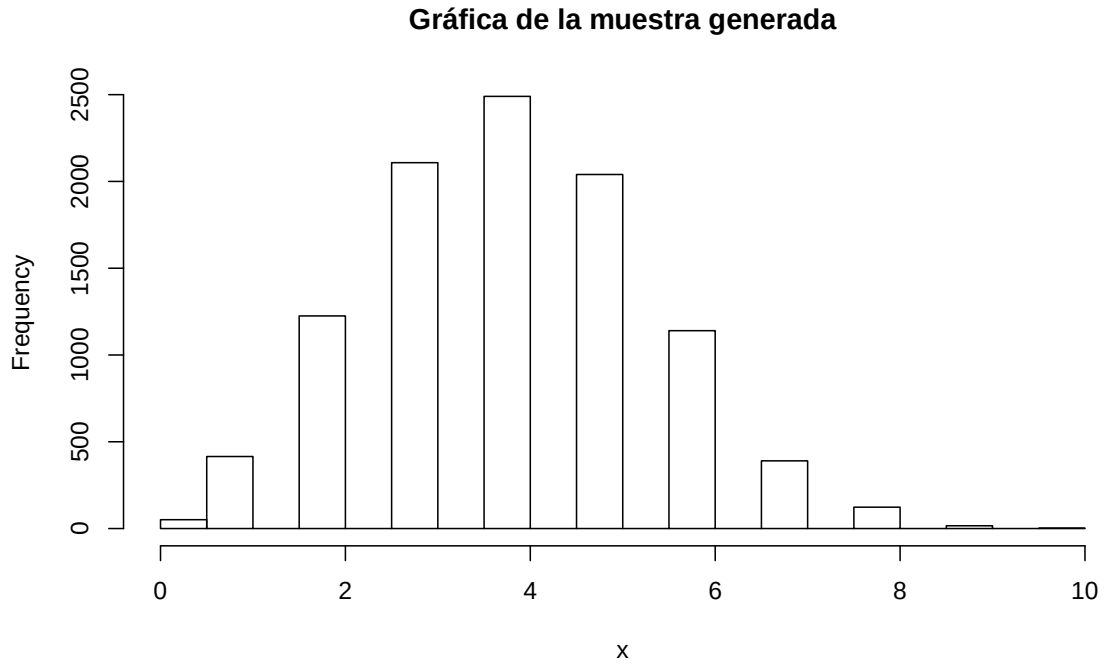
□

7. Desarrollen un algoritmo para generar una variable aleatoria binomial, usando la técnica de convolución (Hint: ¿cuál es la relación entre binomiales y Bernoullis?) Generar una muestra de 100,000 números. ¿Qué método es más eficiente, el de convoluciones o la función `rbinom` en R?

Solución.

El método de convolución es el que genera sumas de variables aleatorias. En este caso, una binomial con parámetros n y p es la suma de n variables Bernoulli con parámetro p. Por lo tanto, basta con generar n Bernoullis y agregar

```r
muestraBinomial <- function(N,n,p){
 x <- NULL
 for(i in 1:N){
        u <-runif(n)
        y <- ifelse(u<=p,1,0)
        x <- c(x,sum(y))
 }
 return(x)
}
 x <- muestraBinomial(10000,10,0.4)
 hist(x,main="Gráfica de la muestra generada")
```



Por último, para ver cuál es más eficiente, podemos tomar el tiempo de ejecución de ambos métodos

```
system.time(x <- muestraBinomial(1000,10,0.4))
```

```
   user  system elapsed 
  0.017   0.000   0.017 
```

```
system.time(x <- rbinom(1000,10,0.4))
```

```
   user  system elapsed 
  0.000   0.000   0.001 
```

Claramente, el método de R es mucho más eficiente.

□

8. Probar que si X tiene función de distribución F, y si $h : \mathbb{R} \to B$ es una función estrictamente creciente donde $B \subseteq \mathbb{R}$, entonces $h(X)$ es una variable aleatoria con distribución $F(h^{-1}(x))$. Además, si F tiene densidad f y h^{-1} es absolutamente continua, entonces $h(X)$ tiene densidad $(h^{-1})'(x)f(h^{-1}(x))$ para casi toda x.

Solución.

Como h es estrictamente creciente $\exists h^{-1} : B \to \mathbb{R}$. Necesitamos calcular $G(y)$, donde G es la distribución de $h(X)$.

Para $y \in B$,

$$\begin{aligned} G(y) &= P(h(X) \leq y) \\ &= P(X \leq h^{-1}(y)) \\ &= F(h^{-1}(y)) \end{aligned}$$

Además, considerando la regla de la cadena, $G'(y) = F'(h^{-1}(y))(h^{-1})'(y) = f(h^{-1}(y))(h^{-1})'(y)$

□

9. El kernel de Epanechnikov reescalado es una función de densidad simétrica:

$$f_e(x) = \frac{3}{4}(1 - x^2), \qquad |x| \leq 1.$$

Consideren el siguiente algoritmo para generar muestras de esta distribución:

(1) Generar uniformes iid $U_1, U_2, U_3 \sim \mathcal{U}(-1, 1)$.

(2) Si $|U_3| \geq |U_2|$ y $|U_3| \geq |U_1|$, utilizar U_2. En otro caso, usar U_3.

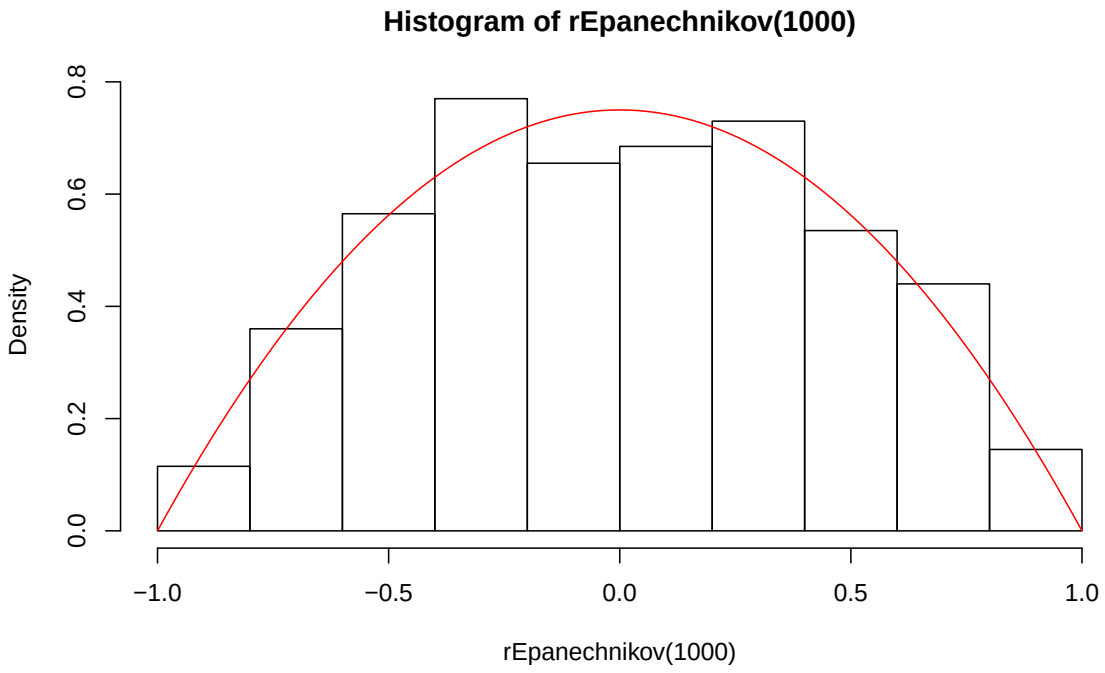
a. Escribir una función para generar variables aleatorias de f_e, y construir el histograma de una muestra de tamaño 1000.

b. Prueben que el algoritmo dado genera muestras de la densidad f_e.

Solución.

Para el inciso (a), la siguiente función genera una muestra de tamaño n

```
rEpanechnikov <- function(n){
        muestra <- numeric(n)
        for(i in 1:n){ u <- runif(3,min=-1,max=1)
                muestra[i] <- ifelse(abs(u[3])>=max(abs(u[2]),abs(u[1])),u[2],u[3])
         }
        return(muestra)
}
hist(rEpanechnikov(1000),prob=T)
curve(0.75*(1-x^2),from=-1,to=1,col="red",add=T)
```



Para el inciso (b), consideren el vector (U_1, U_2, U_3). Este vector es un punto en el cubo $[-1,1]^3$. Sea X la variable aleatoria generada con el algoritmo dado. Esta distribución es simétrica en $[-1,1]$. Si se toma $\alpha \in [0,1]$, $P(|X| < \alpha) = P(\text{máx}(|U_1|, |U_2|) \leq |U_3|), |U_2| \leq \alpha) + P(\text{máx}(|U_1|, |U_2|) > |U_3|), |U_2| \leq \alpha)$, por lo tanto:

$$\int_0^\alpha \left(\int_0^{u_2} (1-u_2)du_1 + \int_{u_1}^1 (1-u_1)du_1 \right) du_2 + \int_0^\alpha (1-u_3^2)du_3 = \frac{3\alpha - \alpha^3}{2}$$

Entonces $P(0 \leq X \leq \alpha) = \frac{3\alpha-\alpha^3}{4}$. Diferenciando con respecto a α, se obtiene que la densidad de X está dada por $f(x) = \frac{3}{4}(1-x^2)$.

□

10. Una compañía de seguros tiene 1000 asegurados, cada uno de los cuales presentará de manera independiente una reclamación en el siguiente mes con probabilidad $p = 0.09245$. Suponiendo que las cantidades de los reclamos hechos son variables aleatorias Gamma(7000,1), hagan simulación para estimar la probabilidad de que la suma de los reclamos exceda $500,000.

Solución.

En este ejercicio está claro que la reclamación presentada es una variable Bernoulli con probabilidad $p = 0.05$, y el total de reclamos es Binomial con parámetros 1000 y 0.05. Si la media de los reclamos es $8,000, entonces el reclamo total promedio es $8000Y$ donde $Y \sim Bin(1000, 0.05)$. Nos piden calcular $P(8000Y > 500000)$ o $P(Y > 500000/8000) = P(Y > 62.5)$.

```
N <- 1e6
y <- rbinom(N,1000,0.05)
Prob <- length(which(y>62.5))/N
Prob
```

```
[1] 0.038282
```

```
#En este caso podemos conocer el valor exacto de la probabilidad
pbinom(62.5,1000,0.05,lower.tail=F)
```

```
[1] 0.03839324
```

□

11. Supongan que la densidad de una variable aleatoria X es $f(x) = \frac{1}{8}xI_{[0,4]}^{(x)}$. Supóngan que se toman muestras de tamaño 8 e interesa generar una muestra de 1000 observaciones del mínimo y del máximo de estas muestras. Obtener las 1000 muestras del minimo y el máximo y hacer los respectivos histogramas.

Solución.

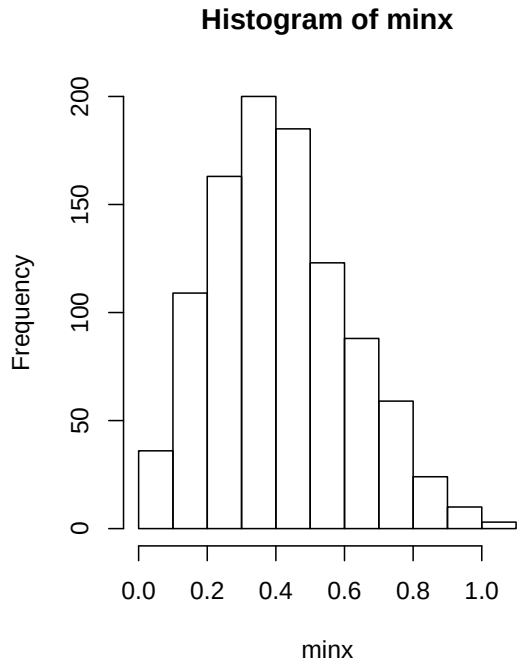
De acuerdo a lo visto en clase, hay dos formas al menos de resolver este problema. Una forma es obtener muestras directamente de la densidad dada, ordenar los valores y obtener el mínimo y el máximo.

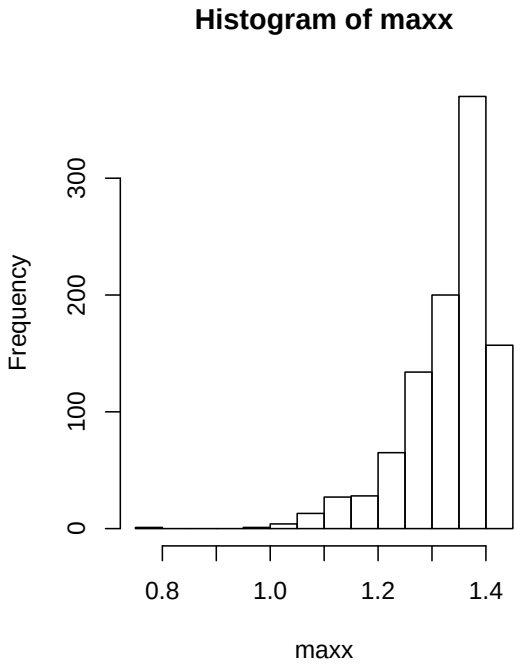
La segunda forma sugiere usar la distribución beta. Aquí lo haremos por ambos métodos.

La función de distribución es $F(x) = \frac{x^2}{4}I_{[0,4]}^{(x)} + I_{x\geq 4}^{(x)}$ Invirtiendo: $X = \sqrt{(2u)}$.

Por el primer método:

```
system.time({
 X <- matrix(sqrt(2*runif(8000)),nrow=1000)
 minx <- apply(X,1,min)
 maxx <- apply(X,1,max)
 par(mfrow=c(1,2))
 hist(minx)
 hist(maxx)})
```

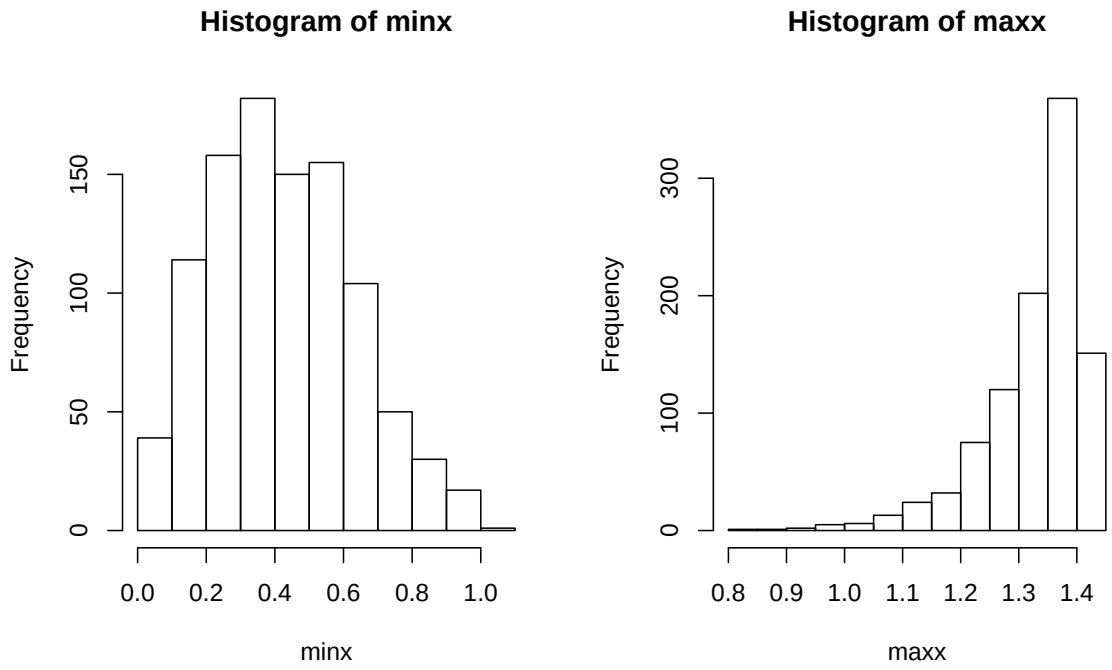





```
   user  system elapsed 
  0.005   0.000   0.005 
```

Por el segundo método, de acuerdo al resultado visto en clase, $n = 8$, $i = 1$ e $i = 8$ son los casos requeridos

```
system.time({
 minx <- sqrt(2*rbeta(1000,1,8))
 maxx <- sqrt(2*rbeta(1000,8,1))
 par(mfrow=c(1,2))
 hist(minx)
 hist(maxx)})
```




```
   user  system elapsed
  0.003   0.000   0.003
```

El segundo método es más rápido.

□

12. Obtener una muestra de tamaño 500 de la distribución conjunta de las siguientes tres variables aleatorias X_1, X_2, X_3 con distribución dada a continuación:

		$X_1 =$					
		0			1		
	$X_2 =$	1	2	3	1	2	3
$X_3 =$	0	0.12	0.10	0.08	0.08	0.06	0.05
	1	0.08	0.06	0.04	0.05	0.04	0.03
	2	0.06	0.04	0.02	0.04	0.03	0.02

Solución.

Para resolver este problema, podemos proceder de dos maneras: una es obteniendo directamente los puntos usando la función `sample`, considerando toda la distribución conjunta, y la otra es usando las condicionales sucesivas. Yo usaré las condicionales sucesivas como un ejemplo del resultado visto en clase.

Para esto, notemos que la marginal de X_1 es (después de corregir el error para que sume 1, conforme lo comentamos) $X_1 = 0$ con probabilidad 0.6 y $X_1 = 1$ con probabilidad 0.4. Por otra parte la condicional de X_2 dado X_1 toma valores, $0.26/0.6, 0.20/0.6, 0.14/0.6$ para $X_2 = 1, 2, 3$ cuando $X_1 = 0$ respectivamente, y $0.17/0.4, 0.13/0.4, 0.1/0.4$ cuando $X_1 = 1$.

```r
muestracond <- function(x){
#función para muestrear de la condicional bivariada X3|X1,X2
        x3 <- switch(paste0(x,collapse=""),
                                "01" = sample(0:2,1,prob = c(0.12,0.08,0.06)/0.26),
                                "02" = sample(0:2,1,prob = c(0.10,0.06,0.04)/0.2),
                                "03" = sample(0:2,1,prob = c(0.08,0.04,0.02)/0.14),
                                "11" = sample(0:2,1,prob = c(0.08,0.05,0.04)/0.17),
                                "12" = sample(0:2,1,prob = c(0.06,0.04,0.03)/0.13),
                                "13" = sample(0:2,1,prob = c(0.05,0.03,0.02)/0.10)
)
return(x3)
}

X1 <- as.numeric(runif(500) < 0.6) #muestrea X1
X2 <- X3 <- numeric(500)
#muestrea X2|X1 y X3|X1,X2
for(i in 1:length(X1)) {
        X2[i] <- ifelse(X1[i]==1,sample(1:3,1,prob = c(0.26,0.20,0.14)/0.6),
        sample(1:3,1,prob = c(0.17,0.13,0.1)/0.4))
        X3[i] <- muestracond(c(X1[i],X2[i]))
}
#muestra de vectores aleatorios
X <- cbind(X1,X2,X3)
head(X,30)
```

```
      X1 X2 X3
 [1,]  1  2  1
 [2,]  0  2  1
 [3,]  0  1  0
 [4,]  1  1  0
 [5,]  0  1  0
 [6,]  1  1  1
 [7,]  0  2  2
 [8,]  1  3  0
 [9,]  0  2  2
[10,]  1  2  0
[11,]  0  1  1
[12,]  1  2  2
[13,]  1  1  0
[14,]  1  1  1
[15,]  0  2  2
[16,]  1  3  0
[17,]  1  1  2
[18,]  0  3  0
[19,]  0  2  2
[20,]  1  1  1
[21,]  0  1  0
[22,]  0  1  0
[23,]  0  3  0
[24,]  1  1  0
[25,]  1  1  1
[26,]  1  2  2
[27,]  1  1  0
[28,]  0  1  0
[29,]  0  1  1
[30,]  1  3  0
```