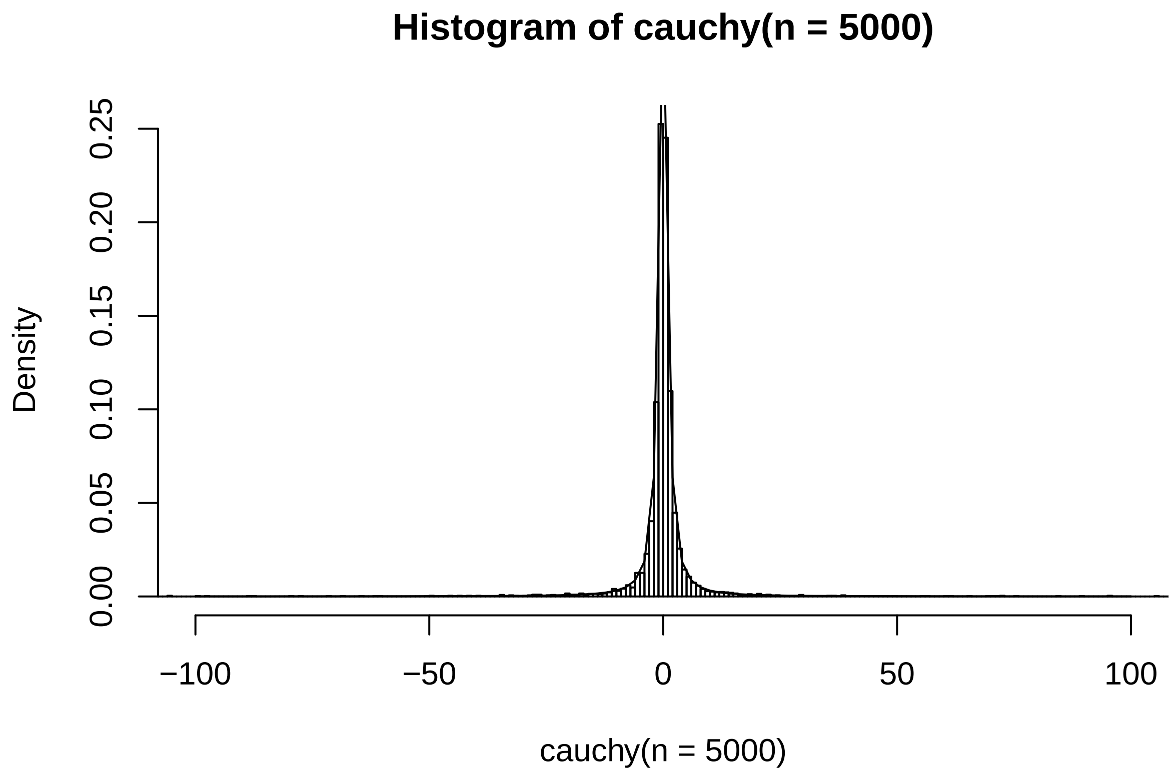
Tarea 2

Pablo Gracia Galeana
Cesar Gonzalez Macedo
Miguel Ángel Fuentes Borboa
Roberto Antonio Yglesias Galeana

Pregunta 1

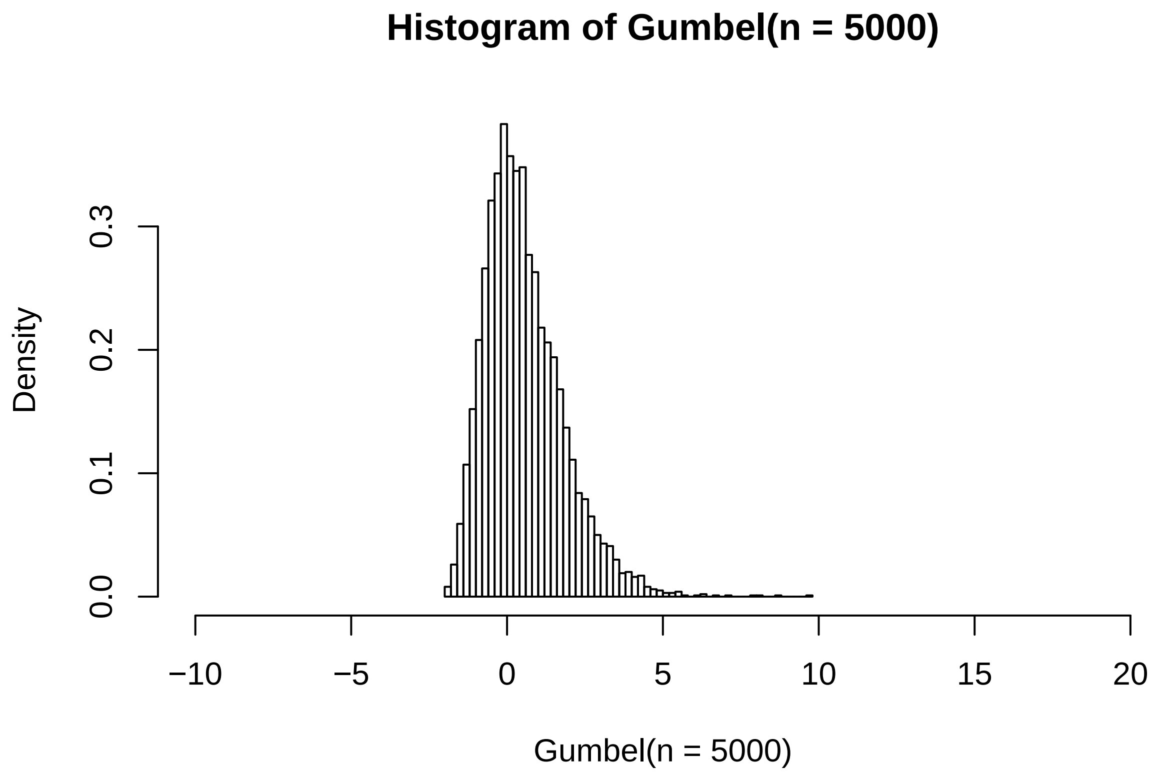
(A) Cauchy

```
cauchy <- function(gamma = 0, beta = 1, n = 50){
  uniformes <- NULL
  uniformes <- runif(n)
  uniformes <- tan(pi*uniformes)*beta+gamma
  return(uniformes)
}
x <- 1:100
hist(cauchy(n=5000), xlim = c(-100,100),breaks = 3000,probability = T)
curve(dcauchy(x),add = T,from = -100,to =100)
```



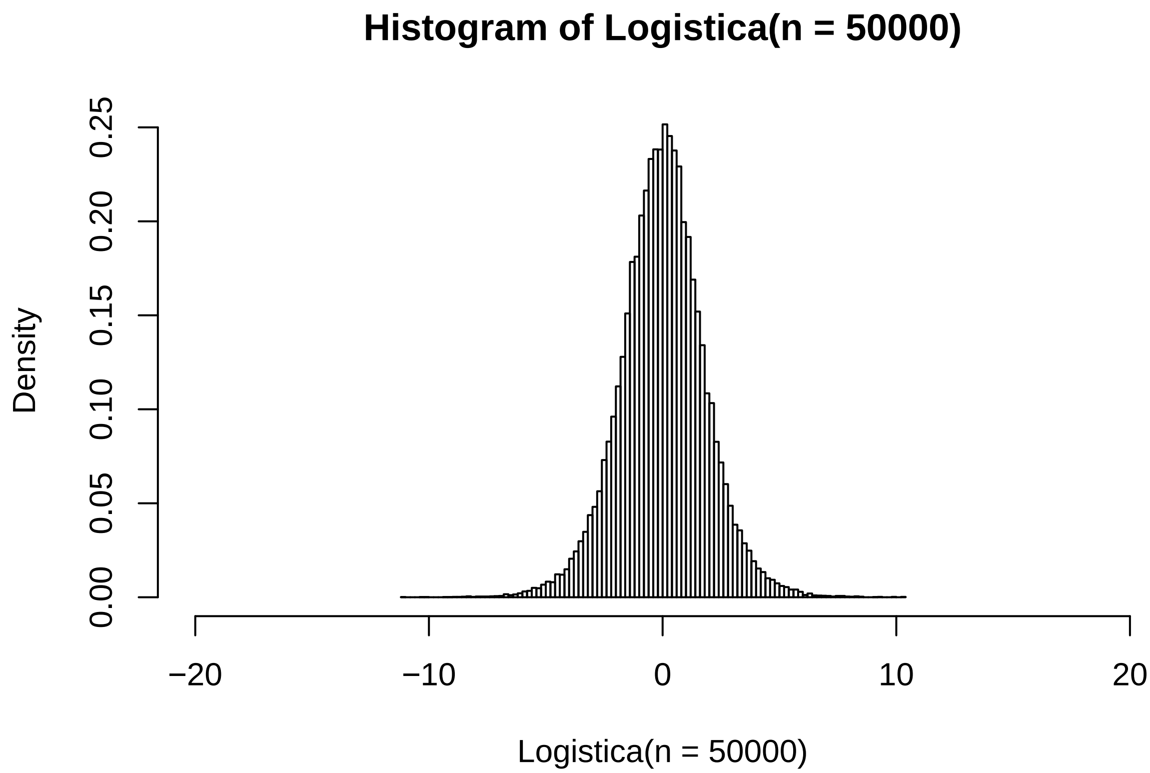
(B) Gumbel

```
Gumbel <- function(gamma = 0, beta = 1, n = 50){
  uniformes <- NULL
  uniformes <- runif(n)
  uniformes <- -beta*log(-log(uniformes))+gamma
  return(uniformes)
}
hist(Gumbel(n=5000), xlim = c(-10,20),breaks = 70,probability = T)
```



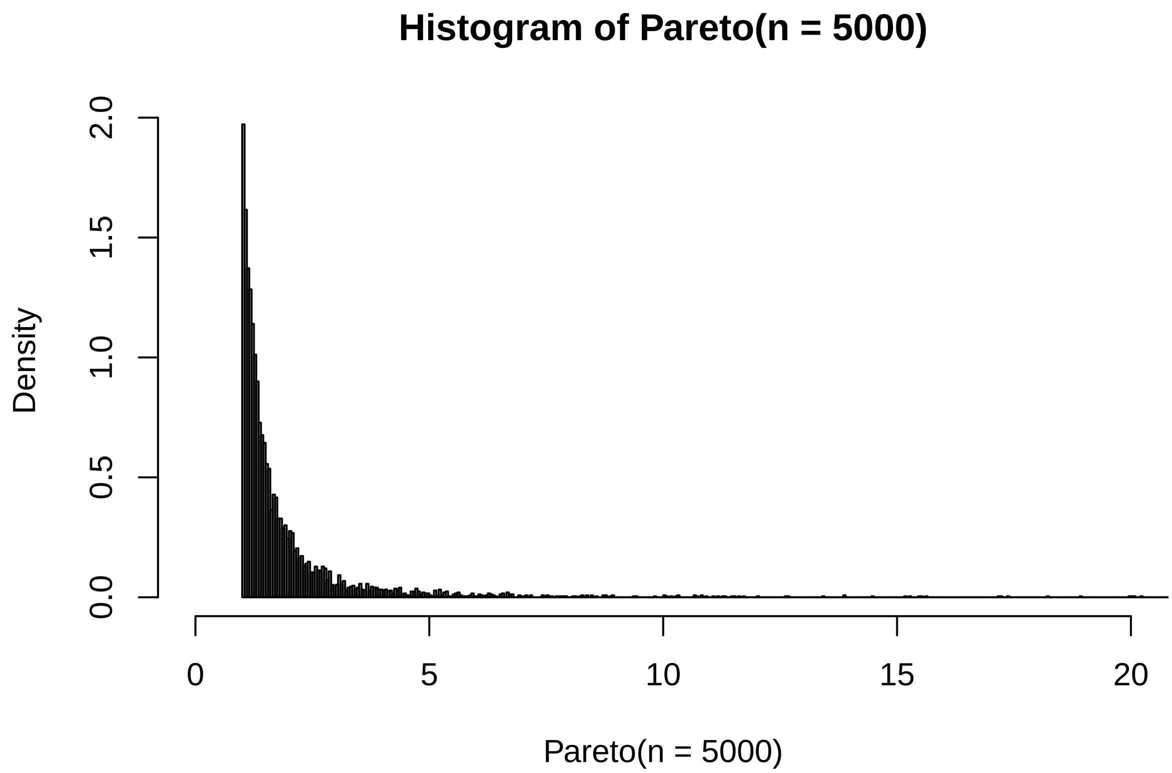
(C) Logistica

```
Logistica <- function(gamma = 0, beta = 1, n = 50){
  uniformes <- NULL
  uniformes <- runif(n)
  uniformes <- -beta*log(1/uniformes-1)+gamma
  return(uniformes)
}
hist(Logistica(n=50000), xlim = c(-20,20),breaks = 100,probability = T)
```

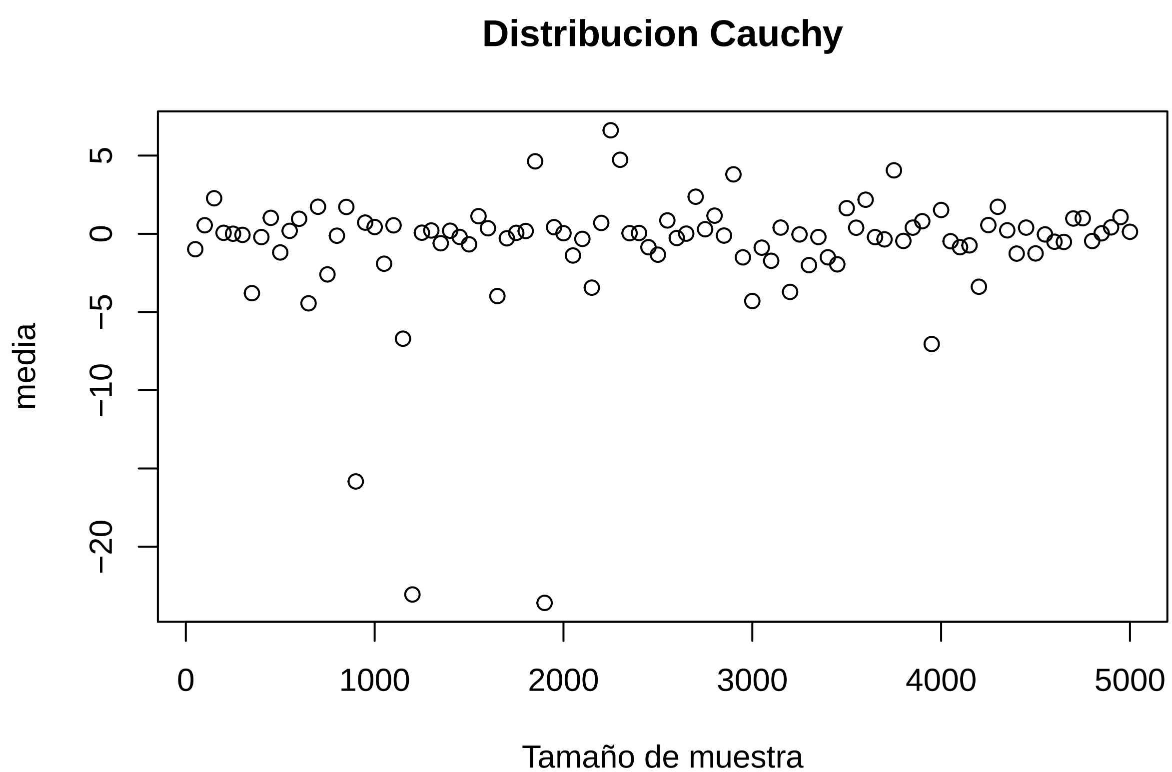


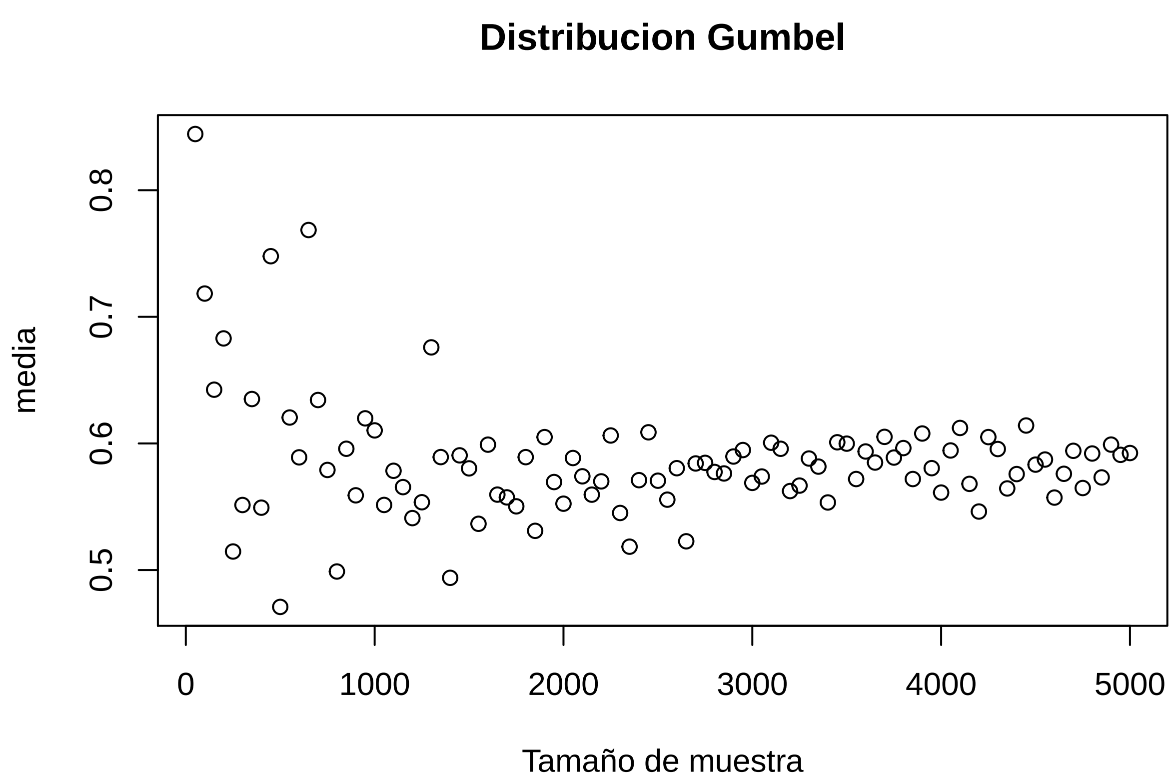
(D) Pareto

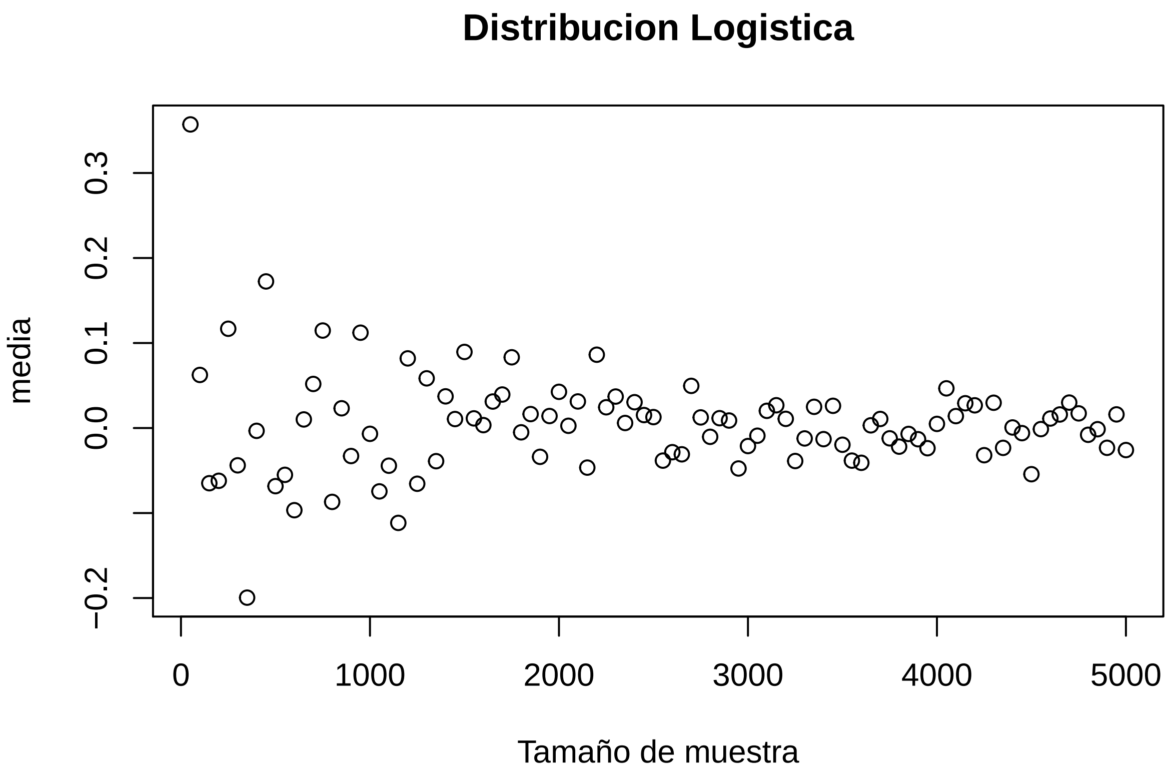
```
Pareto <- function(c = 1, alpha = 2, n = 50){
  uniformes <- NULL
  uniformes <- runif(n)
  uniformes <- c/(1-uniformes)^(1/alpha)
  return(uniformes)
}
hist(Pareto(n=5000), xlim = c(0,20),breaks = 800,probability = T)
```


Histogram of Pareto(n = 5000)
Density
0.0
0.5
1.0
1.5
2.0
0
5
10
15
20
Pareto(n = 5000)

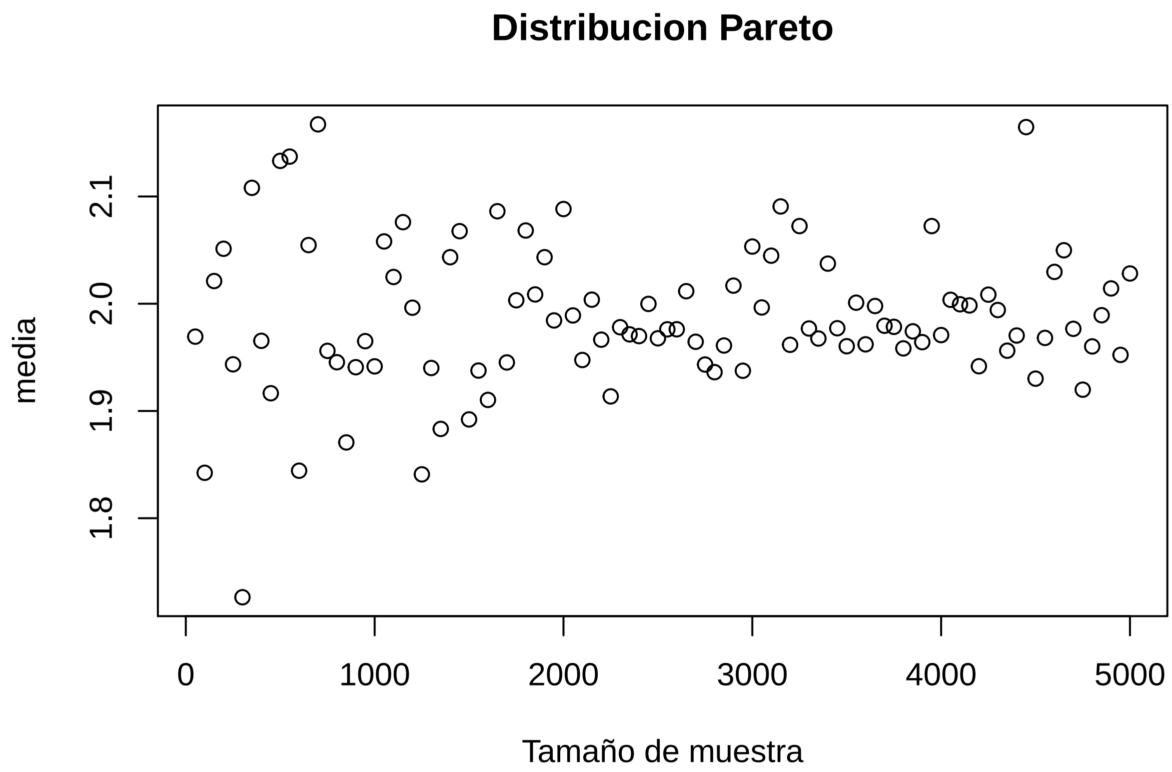
Verificacion de la ley fuerte de los grandes numeros







Distribucion Logistica
media
0.3
0.2
0.1
0.0
−0.2
0
1000
2000
3000
4000
5000
Tamaño de muestra



Pregunta 2

```
ppareto <- function(x){
  ac <- NULL
  for(i in 1:1000){
  obs <- Pareto(n = 5000)
  bool <- obs<x
  ac[i] <- sum(bool)/5000
  }
  return(round(mean(ac),3))
}

ppareto(2)
```

```
## [1] 0.75
```

```
dpareto <- function(x){
  ac <- NULL
  for(i in 1:1000){
    obs <- Pareto(n = 5000)
    bool <- obs<x+1/1000 & obs>x
    ac[i] <- sum(bool)/5000
  }
  return(round(mean(ac),3))
```

```
}

dpareto(2)
```

```
## [1] 0
```

```
qpareto <- function(x){
  ac <- NULL
  for(i in 1:1000){
    obs <- Pareto(n = 5000)
    bool <- sort(obs)[x*5000]
    ac[i] <- bool
  }
  return(round(mean(ac),3))
}

qpareto(.75)
```

```
## [1] 2
```

Pregunta 3

```
prob <- c(.1,.3,.5,.7,1)

frec <- findInterval(runif(1000),prob)

table(frec)
```

```
## frec
##   0   1   2   3   4
##  85 185 226 207 297
```

Pregunta 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 $0 < a < \frac{1}{2}$

Solución

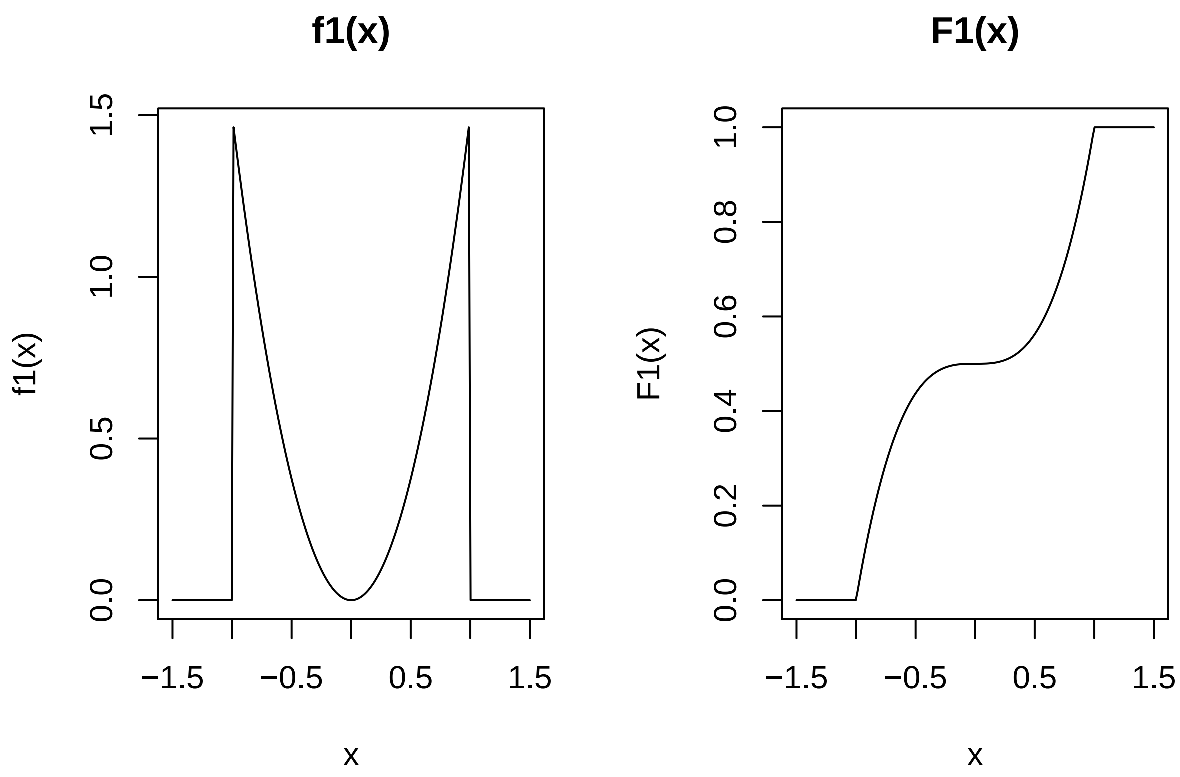
Para el inciso (a), consideremos las funciones de densidad y distribución dadas a continuación:

```
indicadora <- function(x,a,b){
  ifelse(x <=b & x >=a,1,0)
  } #funciÃ³n indicadora en el intervalo (a,b).
```

```
f1 <- function(x){
  3*x^2/2*indicadora(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 en el que graficaremos.
par(mfrow=c(1,2))
plot(x,f1(x),type="l",main="f1(x)")
plot(x,F1(x),type="l",main="F1(x)")
```



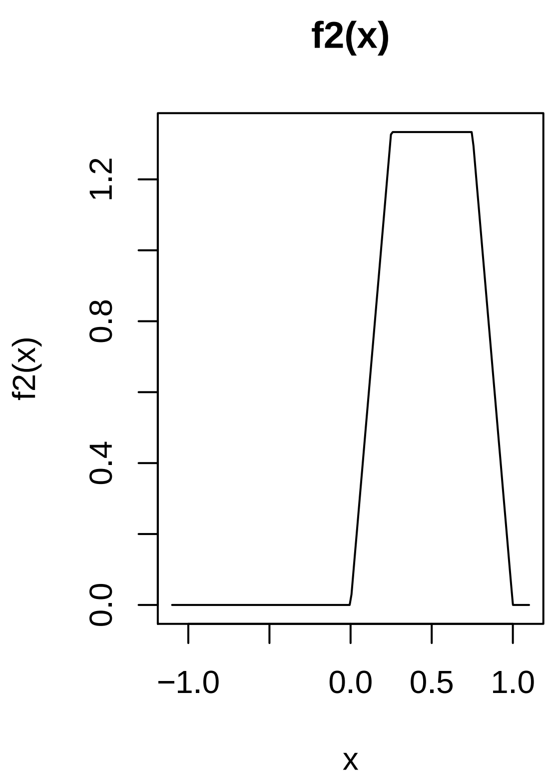
Para el inciso (b)

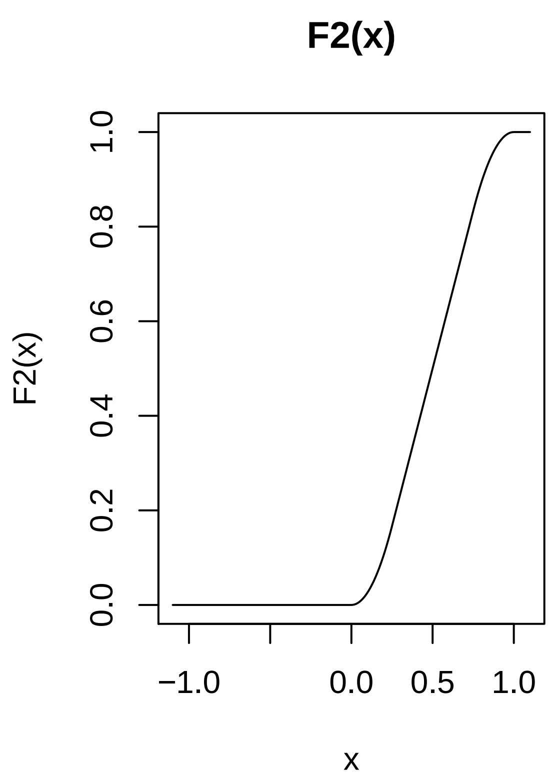
```
f2 <- function(x,a=0.25){
  indicadora(x,-1,1)*(indicadora(x,0,a)*(x/(a*(1-a)))+indicadora(x,a,1-a)/(1-a)+indicadora(x,1-a,1)*(1-
}

F2 <- function(x,a=0.25){
  indicadora(x,0,a)*x^2/(2*a*(1-a)) +
(x-a/2)/(1-a)*indicadora(x,a,1-a) +
((1-3*a/2)/(1-a) + (x*(1-x/2)-(1-a)*(1+a)/2)/(a*(1-a)))*indicadora(x,1-a,1) + indicadora(x,1,100)
}

par(mfrow=c(1,2))
```

```r
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)")
```





Pregunta 5

Considerando la transformacion polar de Marsaglia para generar muestras de normales estÃ¡ndar, muestren que la probabilidad de aceptación de $S = V12 + V22$ en el paso 2 es $\frac{\pi}{4}$, y encuentren la distribución del número de rechazos de S antes de que ocurra una aceptación ¿Cuál es el númerp esperado de ejecuciones del paso 1?

Solución

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 $4â\pi = (1â\frac{\pi}{4}) = 0.215$. Entonces se rechaza 21.5% del tiempo. Ahora bien, si $X =$ número de rechazos antes de aceptar, sabemos que $Xgeom(\pi/4)$. Entonces $E(X) = 1/\pi = 4/\pi â = 1.2732395$.

Pregunta 6

Generamos el vector con las probabilidades para cada valor posible de x

```r
p <- 1:100
for(i in 1:100){
  p[i] <- (2*i)/(100*(100+1))
}
```

Muestra de 10,000 numeros

```r
m_disc <- sample(1:100, size = 10000, replace = T, prob = p)
head(m_disc, n=100)
```

```
##   [1]  79  83  73  71  37  72  60  77  26  72  51  54  80  39  89  14  75
##  [18]  88  99  50  30  39  78  66  50  93  49  94  86  70  95  91  94  54
##  [35]  83  46  53 100  83  51  91   2  53  43  83  89  72  91  91  48  93
##  [52]  31  53  79  75  42  17  91  24  17  32  85  41  76  26  90  76  91
##  [69]  86  48 100  97  16  43  45  90  94  93  73  76  51  76  64  92  38
##  [86]  64  41  43  16  82  68  62  90  84  98  54  32  28  42  96
```

Pregunta 7

Algortimo que genera una variable aleatoria binomial

```r
binom_sim <- function(n,p){
  x <- sum(sample(c(0,1), size = n, replace = T, prob = c(1-p,p)))
}
```

Muestra de 100,000 numeros

```r
start_time1 <- Sys.time()
s1 <- 1:100000
for(i in 1:100000){
  s1[i] <- binom_sim(5,0.30)
}
end_time1 <- Sys.time()
end_time1 - start_time1
```

```
## Time difference of 0.687304 secs
```

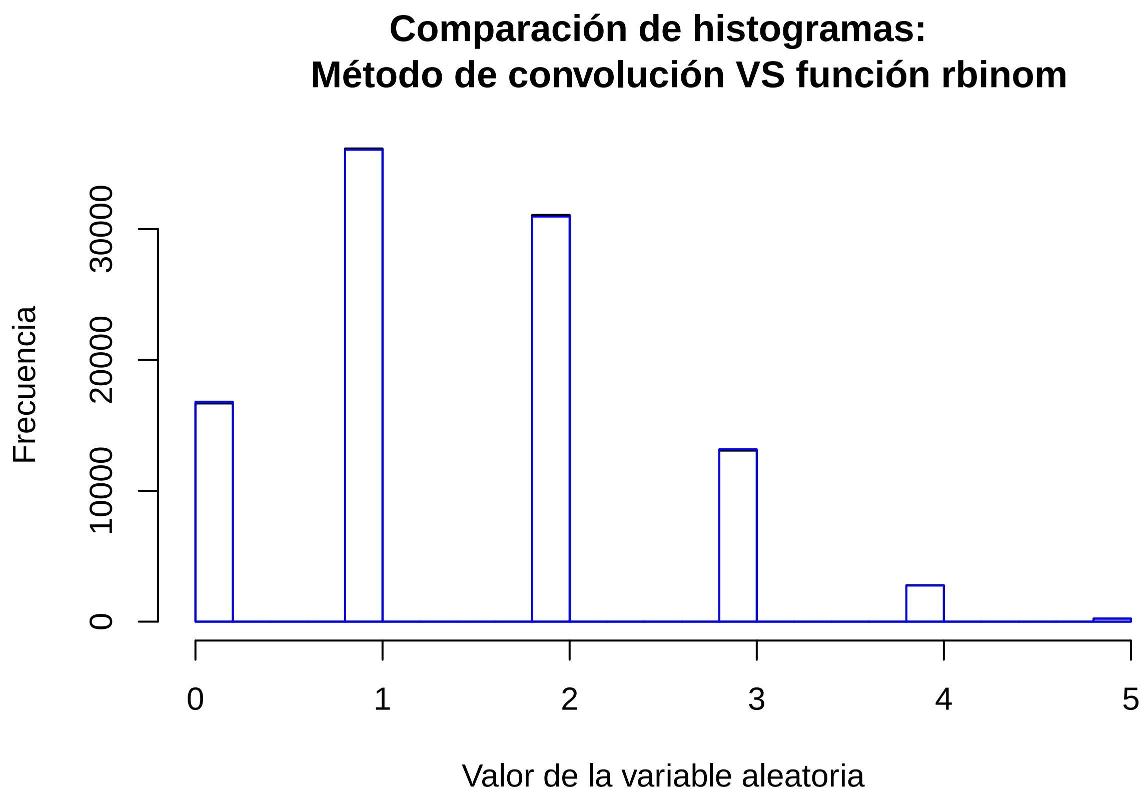
```r
start_time2 <- Sys.time()
s2 <- rbinom(100000, size = 5, prob = 0.30)
end_time2 <- Sys.time()
end_time2 - start_time2
```

```
## Time difference of 0 secs
```

La funcion rbinom es mas eficiente que el metodo de convoluciones ya que su tiempo de ejecucion es mucho menor.

```r
hist(s1, main = "Comparación de histogramas:
     Método de convolución VS función rbinom", xlab = "Valor de la variable aleatoria", ylab = "Frecuen

hist(s2, add = T, border ="blue")
```



Pregunta 8

Sea X una variable aleatoria con función de distribución F, densidad f y $h : \mathbb{R} \to B$ estrictamente creciente.

Por demostrar: $h(X)$ tiene como función de distribución $F(h^{-1}(x))$.

Sea $G(u)$ la función de distribución de $h(x)$.

$$G(u) = P(h(X) \leq u) = P(X \leq h^{-1}(u)) = F(h^{-1}(u))$$

h^{-1} existe ya que h es estrictamente creciente.

Por demostrar: $h(X)$ tiene como densidad $(h^{-1})'(x)f(h^{-1}(x))$.

Para encontrar la densidad hay que derivar $G(u)$.

$$\frac{d}{du}G(u) = \frac{d}{du}F(h^{-1}(u)) = f(h^{-1}(u))\frac{d}{du}h^{-1}(u) = f(h^{-1}(u))(h^{-1})'(u)$$

Pregunta 9

Función que genera muestras de tamano n de la distribucion Kernel de Epanechnikov

```r
repa_ker <- function(n){
  z <- 1:n
  for(i in 1:n){
    u <- runif(3, min = -1, max = 1) #Generación de uniformes aleatorias (-1,1)
      if(abs(u[3]) > abs(u[2]) & abs(u[3]) > abs(u[1])){
        u_opt <- u[2]
      }else{
        u_opt <- u[3]
      }
    z[i] <- u_opt
  }
  return(z)
}
```

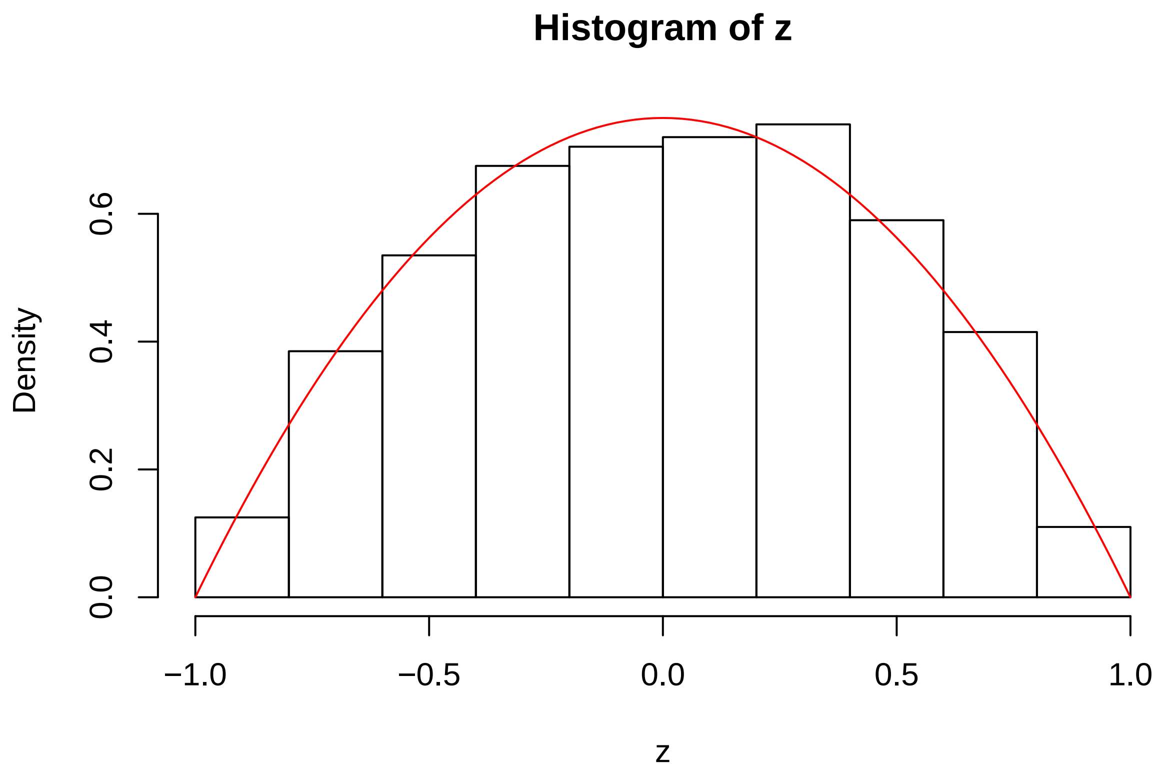
Muestra de tamaño 1000

```r
z<-repa_ker(1000)
head(z, n=100)
```

```
##   [1]  0.51000449  0.12974746 -0.67228003 -0.07778297 -0.54533009
##   [6] -0.33151430 -0.68299183  0.16686446  0.41491143 -0.33128275
##  [11]  0.22394810 -0.73829040 -0.09375745 -0.91438040 -0.45695859
##  [16]  0.28501038  0.34004263 -0.06802906  0.20069508  0.29233568
##  [21]  0.45301814  0.11139391 -0.02197356  0.40741274 -0.23531580
##  [26] -0.49321934 -0.23428603 -0.08139575  0.41039707  0.43359062
##  [31] -0.33375177  0.62774649  0.02372859  0.88333665  0.04998663
##  [36] -0.37872265 -0.23307714 -0.77148667 -0.59976992 -0.14809496
##  [41] -0.52375358  0.15456940 -0.75695159  0.34731587 -0.55754746
##  [46]  0.11912790  0.48534469 -0.24146955 -0.28092354  0.45803340
##  [51] -0.69406080  0.15554174 -0.04689945  0.04041218  0.55715659
##  [56] -0.30562957 -0.47716696  0.43073559  0.71704671  0.16701461
##  [61]  0.40207499 -0.26166440  0.35185825 -0.18038187  0.59392596
##  [66] -0.39627890 -0.05093317 -0.06615017 -0.38711061 -0.55184085
##  [71]  0.25504262  0.41090433  0.63663555  0.09229962 -0.55651940
##  [76] -0.58973490  0.73679345  0.24768577 -0.80164804  0.59582098
##  [81] -0.32048836 -0.21121943 -0.23313747  0.89537553  0.25246446
##  [86] -0.20533793  0.89274704  0.68024350 -0.22452215 -0.13547424
##  [91]  0.48175374  0.60385793  0.88655381 -0.45413432  0.51363856
##  [96] -0.28186398  0.66993040 -0.09635134 -0.36766883  0.62024534
```

Histograma de la muestra con la gráfica de la distribución Kernel de Epanechnikov

```r
hist(z, prob = T)
curve((3/4)*(1-x^2), from = -1, to = 1, add = T, col = "red")
```



Pregunta 10

Primero se simula el numero de reclamaciones que se van a tener.

```
numeroreclammaciones<-sum(rbinom(n = 1000, size = 1, prob = 0.09245))
numeroreclammaciones
```

```
## [1] 85
```

Luego se simula las muestras con una Gamma(7000,1)

```
TotalMontos <- sum(rgamma(numeroreclammaciones, shape = 7000, scale = 1))
TotalMontos
```

```
## [1] 595196.4
```

Se hace este procedimiento para 5000 simulaciones.

```
mayor500M <- NULL
for(i in 1:10000){
  numeroreclammaciones<-sum(rbinom(n = 1000, size = 1, prob = 0.09245))
  TotalMontos <- sum(rgamma(numeroreclammaciones, shape = 7000, scale = 1))
  if(TotalMontos > 500000){
    mayor500M <- c(mayor500M,TotalMontos)
  }
}
```

Probabilidad estimada

```r
length(mayor500M)/10000
```

```
## [1] 0.9899
```

Pregunta 11

X es una variable aleatoria con densidad $f(x) = xI_{[0,4]}^{(x)}$. Su funcion de distribución es:

$$F(x) = 0I_{x<0}^{(x)} + \int_0^x \frac{1}{8}uduI_{[0,4]}^{(x)} + 1I_{x>4}^{(x)} = 0I_{x<0}^{(x)} + \frac{x^2}{16}I_{[0,4]}^{(x)} + 1I_{x>4}^{(x)}$$

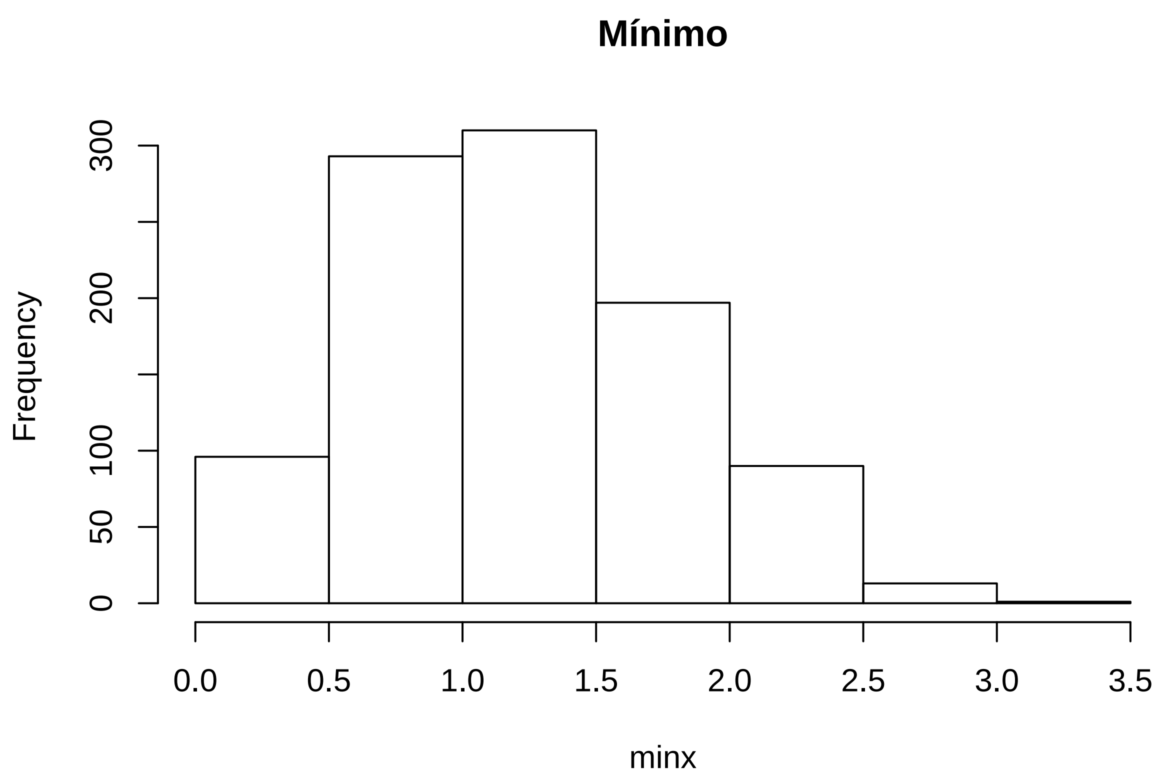
Para simular esta variable aleatoria hay que usar el teorema de la transformación inversa.

$$u = \frac{x^2}{16}$$

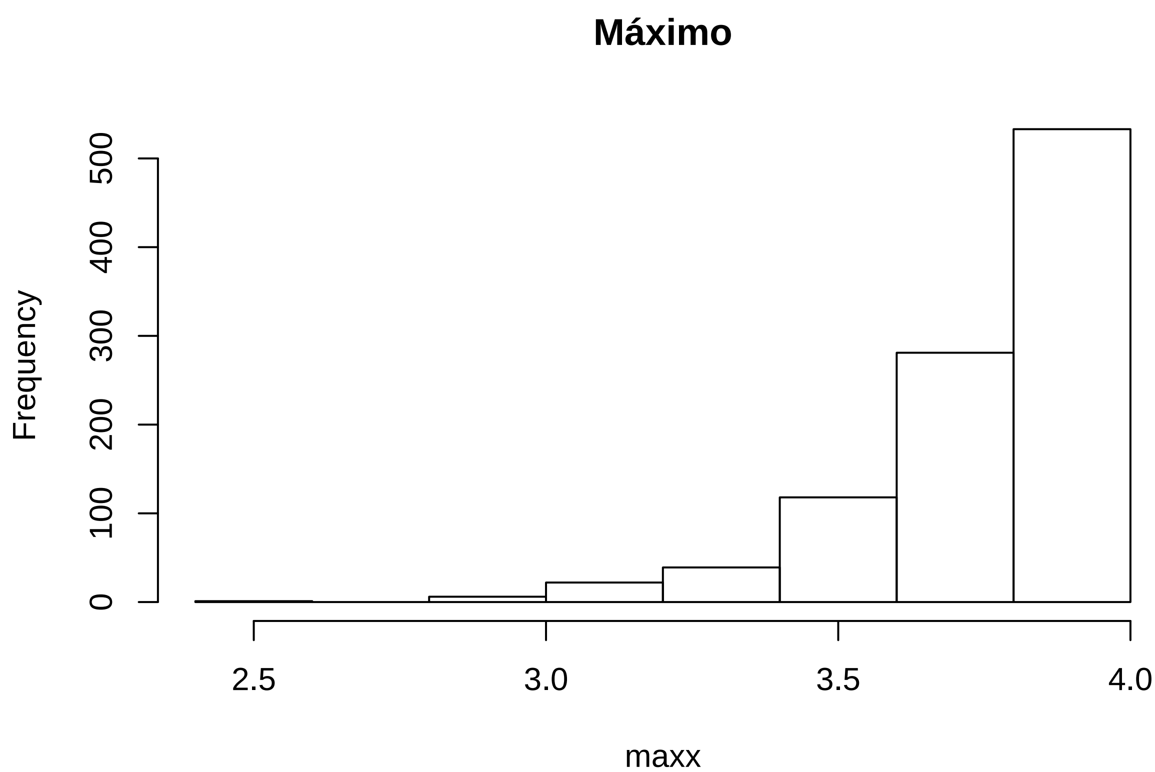
Entonces

$$x = 4\sqrt{u}$$

```r
minx <- NULL
maxx <- NULL
for(i in 1:1000){
  x <- 4*sqrt(runif(8))
minx[i] <- min(x)
maxx[i] <- max(x)
}
hist(minx, breaks = 8, main = "Mínimo")
```



```r
hist(maxx, breaks = 8, main = "Máximo")
```



Pregunta 12

Primero obtenemos las densidades marginales de cada variable

$$f_{X_1}(x) = \begin{cases} 0.7 & x = 0 \\ 0.3 & x = 1 \end{cases}$$

$$f_{X_2}(x) = \begin{cases} 0.39 & x = 1 \\ 0.3 & x = 2 \\ 0.31 & x = 3 \end{cases}$$

$$f_{X_3}(x) = \begin{cases} 0.55 & x = 0 \\ 0.27 & x = 1 \\ 0.18 & x = 2 \end{cases}$$

```r
p1 <- cumsum(c(0,0.7, 0.3))
p2 <- cumsum(c(0,0.39, 0.3, 0.31))
p3 <- cumsum(c(0,0.55, 0.27, 0.18))
x1 <- NULL
x2 <- NULL
x3 <- NULL
```

```r
for(i in 1:500){
  x1[i] <- findInterval(runif(1), p1)-1
  x2[i] <- findInterval(runif(1), p2)
  x3[i] <- findInterval(runif(1), p3)-1
}

cbind(x1,x2,x3)
```

```
##        x1 x2 x3
##   [1,]  0  3  0
##   [2,]  0  1  2
##   [3,]  0  3  0
##   [4,]  1  3  1
##   [5,]  0  2  1
##   [6,]  0  3  0
##   [7,]  0  1  1
##   [8,]  0  2  0
##   [9,]  0  2  0
##  [10,]  0  1  0
##  [11,]  0  2  2
##  [12,]  1  1  0
##  [13,]  1  1  0
##  [14,]  0  1  0
##  [15,]  1  3  0
##  [16,]  1  1  1
##  [17,]  1  1  0
##  [18,]  1  2  1
##  [19,]  0  1  0
##  [20,]  1  1  0
##  [21,]  1  2  2
##  [22,]  0  1  0
##  [23,]  1  3  1
##  [24,]  1  2  1
##  [25,]  0  1  1
##  [26,]  0  2  0
##  [27,]  0  2  0
##  [28,]  1  1  1
##  [29,]  1  2  0
##  [30,]  0  3  1
##  [31,]  0  2  0
##  [32,]  0  1  0
##  [33,]  1  2  1
##  [34,]  1  3  0
##  [35,]  0  3  2
##  [36,]  0  3  0
##  [37,]  0  1  0
##  [38,]  0  2  0
##  [39,]  1  2  0
##  [40,]  0  3  1
##  [41,]  0  1  0
##  [42,]  0  1  0
##  [43,]  0  3  2
##  [44,]  0  2  0
```

```
## [45,]    1    1    0
## [46,]    0    1    2
## [47,]    0    1    0
## [48,]    0    3    0
## [49,]    1    2    1
## [50,]    1    1    2
## [51,]    1    3    0
## [52,]    0    3    0
## [53,]    0    3    0
## [54,]    0    1    1
## [55,]    0    1    0
## [56,]    1    3    1
## [57,]    0    3    2
## [58,]    1    2    1
## [59,]    0    2    0
## [60,]    0    3    1
## [61,]    1    2    0
## [62,]    0    1    1
## [63,]    0    1    1
## [64,]    0    1    0
## [65,]    0    1    0
## [66,]    0    2    0
## [67,]    0    3    1
## [68,]    1    1    0
## [69,]    0    1    0
## [70,]    0    2    0
## [71,]    0    1    1
## [72,]    0    1    0
## [73,]    0    3    0
## [74,]    1    1    0
## [75,]    0    1    2
## [76,]    0    3    0
## [77,]    1    2    0
## [78,]    0    1    0
## [79,]    0    3    0
## [80,]    0    2    0
## [81,]    0    1    2
## [82,]    1    1    0
## [83,]    0    1    0
## [84,]    0    1    2
## [85,]    1    2    0
## [86,]    0    2    0
## [87,]    0    2    2
## [88,]    1    1    0
## [89,]    1    2    0
## [90,]    0    3    0
## [91,]    1    1    1
## [92,]    0    2    1
## [93,]    1    2    0
## [94,]    0    1    0
## [95,]    0    3    1
## [96,]    0    1    0
## [97,]    0    3    2
## [98,]    1    2    1
```

```
##  [99,]  0  3  0
## [100,]  1  1  2
## [101,]  0  1  2
## [102,]  1  1  0
## [103,]  0  2  0
## [104,]  1  3  0
## [105,]  1  1  0
## [106,]  0  2  2
## [107,]  0  2  0
## [108,]  0  2  1
## [109,]  1  2  0
## [110,]  0  2  0
## [111,]  0  2  2
## [112,]  1  2  1
## [113,]  0  1  0
## [114,]  0  3  2
## [115,]  0  3  1
## [116,]  0  2  0
## [117,]  1  3  0
## [118,]  0  2  1
## [119,]  0  1  0
## [120,]  1  2  0
## [121,]  0  1  0
## [122,]  0  2  0
## [123,]  0  2  0
## [124,]  0  2  0
## [125,]  0  2  0
## [126,]  0  3  0
## [127,]  0  1  2
## [128,]  0  2  0
## [129,]  0  3  0
## [130,]  0  2  2
## [131,]  0  2  0
## [132,]  0  1  0
## [133,]  0  1  0
## [134,]  1  2  2
## [135,]  0  1  0
## [136,]  0  1  0
## [137,]  0  1  1
## [138,]  0  3  0
## [139,]  0  3  1
## [140,]  0  2  0
## [141,]  0  3  2
## [142,]  0  3  0
## [143,]  1  1  1
## [144,]  1  3  1
## [145,]  0  2  1
## [146,]  0  2  0
## [147,]  0  2  2
## [148,]  0  1  1
## [149,]  0  1  0
## [150,]  0  1  1
## [151,]  1  1  2
## [152,]  0  2  0
```

```
## [153,]    0    3    0
## [154,]    0    1    1
## [155,]    0    3    0
## [156,]    0    1    0
## [157,]    0    3    0
## [158,]    1    1    0
## [159,]    1    1    1
## [160,]    1    3    0
## [161,]    0    3    0
## [162,]    0    1    0
## [163,]    0    2    1
## [164,]    0    1    0
## [165,]    1    2    0
## [166,]    0    3    0
## [167,]    0    2    1
## [168,]    0    3    0
## [169,]    1    3    0
## [170,]    0    1    1
## [171,]    0    1    0
## [172,]    1    2    0
## [173,]    1    3    1
## [174,]    1    3    0
## [175,]    0    3    1
## [176,]    1    1    1
## [177,]    0    2    1
## [178,]    0    1    0
## [179,]    0    1    2
## [180,]    1    3    1
## [181,]    0    1    1
## [182,]    1    1    2
## [183,]    1    3    0
## [184,]    0    2    0
## [185,]    0    3    0
## [186,]    0    2    1
## [187,]    0    1    0
## [188,]    0    2    2
## [189,]    0    1    1
## [190,]    1    1    0
## [191,]    0    1    1
## [192,]    0    2    0
## [193,]    0    3    1
## [194,]    1    3    1
## [195,]    0    3    0
## [196,]    0    1    1
## [197,]    0    1    0
## [198,]    1    2    2
## [199,]    0    1    1
## [200,]    1    1    2
## [201,]    1    3    0
## [202,]    0    2    0
## [203,]    0    2    0
## [204,]    0    2    0
## [205,]    0    2    0
## [206,]    0    3    0
```

```
## [207,]  0  1  0
## [208,]  0  3  1
## [209,]  0  1  0
## [210,]  1  2  0
## [211,]  0  1  0
## [212,]  1  2  1
## [213,]  1  3  2
## [214,]  0  1  0
## [215,]  0  1  0
## [216,]  0  2  0
## [217,]  1  1  0
## [218,]  0  1  0
## [219,]  0  2  0
## [220,]  0  1  2
## [221,]  0  3  2
## [222,]  0  1  2
## [223,]  0  2  0
## [224,]  0  1  1
## [225,]  1  2  1
## [226,]  1  2  2
## [227,]  1  3  0
## [228,]  1  3  1
## [229,]  0  1  0
## [230,]  0  2  1
## [231,]  0  2  0
## [232,]  0  2  1
## [233,]  0  2  0
## [234,]  0  3  0
## [235,]  1  1  0
## [236,]  1  1  0
## [237,]  1  1  0
## [238,]  1  2  0
## [239,]  0  3  0
## [240,]  0  2  1
## [241,]  0  2  0
## [242,]  1  3  0
## [243,]  1  2  1
## [244,]  0  2  1
## [245,]  0  3  0
## [246,]  0  3  2
## [247,]  0  1  1
## [248,]  0  1  0
## [249,]  0  3  0
## [250,]  0  1  0
## [251,]  1  3  0
## [252,]  0  3  2
## [253,]  0  1  0
## [254,]  0  2  2
## [255,]  1  2  1
## [256,]  1  1  0
## [257,]  0  3  0
## [258,]  0  2  0
## [259,]  0  1  1
## [260,]  0  3  0
```

```
## [261,]  1  3  0
## [262,]  1  1  0
## [263,]  1  1  1
## [264,]  1  1  0
## [265,]  0  2  0
## [266,]  1  1  2
## [267,]  0  2  0
## [268,]  1  1  2
## [269,]  0  1  1
## [270,]  1  1  0
## [271,]  0  3  0
## [272,]  1  2  2
## [273,]  0  1  1
## [274,]  0  3  0
## [275,]  1  2  2
## [276,]  1  3  0
## [277,]  1  1  0
## [278,]  0  2  0
## [279,]  1  2  0
## [280,]  0  1  1
## [281,]  1  2  0
## [282,]  1  2  0
## [283,]  1  2  2
## [284,]  0  2  2
## [285,]  0  1  0
## [286,]  0  2  0
## [287,]  0  3  1
## [288,]  0  3  1
## [289,]  0  1  2
## [290,]  0  1  1
## [291,]  0  3  0
## [292,]  1  1  0
## [293,]  0  3  1
## [294,]  0  3  0
## [295,]  0  3  1
## [296,]  0  1  0
## [297,]  1  2  1
## [298,]  0  1  1
## [299,]  0  1  0
## [300,]  0  2  1
## [301,]  0  1  1
## [302,]  0  1  1
## [303,]  0  2  1
## [304,]  0  1  0
## [305,]  0  3  2
## [306,]  1  2  1
## [307,]  0  3  2
## [308,]  0  1  0
## [309,]  0  3  1
## [310,]  0  1  0
## [311,]  0  2  1
## [312,]  0  2  1
## [313,]  1  3  1
## [314,]  0  3  1
```

```
## [315,]    0    3    1
## [316,]    0    3    2
## [317,]    0    2    0
## [318,]    0    1    1
## [319,]    1    1    0
## [320,]    0    1    0
## [321,]    1    1    0
## [322,]    1    1    0
## [323,]    1    3    1
## [324,]    0    3    2
## [325,]    1    2    0
## [326,]    0    1    0
## [327,]    0    1    2
## [328,]    0    1    0
## [329,]    1    1    1
## [330,]    0    3    0
## [331,]    0    3    1
## [332,]    0    3    1
## [333,]    0    3    0
## [334,]    1    1    0
## [335,]    0    1    2
## [336,]    0    3    0
## [337,]    1    1    0
## [338,]    0    1    2
## [339,]    1    1    0
## [340,]    0    1    0
## [341,]    0    2    0
## [342,]    0    3    0
## [343,]    1    2    0
## [344,]    0    1    0
## [345,]    0    3    0
## [346,]    1    2    0
## [347,]    0    2    0
## [348,]    0    2    2
## [349,]    0    3    1
## [350,]    0    1    2
## [351,]    1    2    2
## [352,]    1    2    1
## [353,]    0    1    2
## [354,]    0    1    0
## [355,]    0    3    1
## [356,]    1    1    0
## [357,]    1    3    0
## [358,]    0    1    0
## [359,]    0    3    0
## [360,]    0    1    0
## [361,]    1    2    2
## [362,]    0    2    1
## [363,]    0    1    2
## [364,]    0    2    1
## [365,]    0    2    0
## [366,]    0    1    0
## [367,]    0    2    0
## [368,]    0    3    1
```

```
## [369,]    0    2    2
## [370,]    0    2    2
## [371,]    0    1    0
## [372,]    0    3    0
## [373,]    0    3    0
## [374,]    1    3    1
## [375,]    0    1    0
## [376,]    1    2    1
## [377,]    1    2    1
## [378,]    1    1    1
## [379,]    0    2    1
## [380,]    0    3    0
## [381,]    0    3    1
## [382,]    1    2    0
## [383,]    0    3    0
## [384,]    0    2    0
## [385,]    0    3    0
## [386,]    1    2    0
## [387,]    0    1    0
## [388,]    0    2    1
## [389,]    0    1    2
## [390,]    0    2    2
## [391,]    0    2    0
## [392,]    0    2    0
## [393,]    0    2    2
## [394,]    0    1    1
## [395,]    0    2    0
## [396,]    1    1    0
## [397,]    0    2    0
## [398,]    0    1    1
## [399,]    0    3    1
## [400,]    0    1    2
## [401,]    0    3    2
## [402,]    0    3    0
## [403,]    0    2    2
## [404,]    0    1    2
## [405,]    0    3    1
## [406,]    1    1    0
## [407,]    0    1    1
## [408,]    0    3    0
## [409,]    0    1    1
## [410,]    1    2    0
## [411,]    0    1    0
## [412,]    0    3    2
## [413,]    0    3    2
## [414,]    0    3    0
## [415,]    0    2    2
## [416,]    0    3    2
## [417,]    0    1    2
## [418,]    0    3    2
## [419,]    0    2    0
## [420,]    0    1    0
## [421,]    0    3    0
## [422,]    1    3    1
```

```
## [423,]    0    1    0
## [424,]    0    3    0
## [425,]    0    1    1
## [426,]    0    2    0
## [427,]    1    2    0
## [428,]    1    3    0
## [429,]    0    1    0
## [430,]    0    1    0
## [431,]    0    1    2
## [432,]    1    2    0
## [433,]    0    2    0
## [434,]    0    3    0
## [435,]    0    3    2
## [436,]    1    3    0
## [437,]    0    3    0
## [438,]    0    2    1
## [439,]    0    3    0
## [440,]    0    3    0
## [441,]    0    3    0
## [442,]    0    2    0
## [443,]    0    2    2
## [444,]    1    1    0
## [445,]    0    3    1
## [446,]    1    3    1
## [447,]    1    1    1
## [448,]    0    1    1
## [449,]    0    3    0
## [450,]    0    2    0
## [451,]    0    3    0
## [452,]    0    1    0
## [453,]    1    1    0
## [454,]    0    3    1
## [455,]    0    3    0
## [456,]    0    2    1
## [457,]    1    1    1
## [458,]    0    1    0
## [459,]    0    2    0
## [460,]    0    2    0
## [461,]    0    3    1
## [462,]    1    2    0
## [463,]    0    2    0
## [464,]    1    1    2
## [465,]    1    3    1
## [466,]    0    3    1
## [467,]    0    1    0
## [468,]    0    3    0
## [469,]    1    1    0
## [470,]    0    2    1
## [471,]    0    2    0
## [472,]    0    3    0
## [473,]    0    3    1
## [474,]    0    2    1
## [475,]    0    3    2
## [476,]    0    3    2
```

```
## [477,]  1  1  0
## [478,]  0  2  0
## [479,]  1  3  0
## [480,]  0  1  0
## [481,]  0  3  1
## [482,]  1  1  1
## [483,]  0  3  0
## [484,]  0  3  0
## [485,]  0  2  1
## [486,]  1  2  2
## [487,]  0  2  0
## [488,]  0  1  2
## [489,]  0  3  0
## [490,]  0  1  0
## [491,]  0  2  0
## [492,]  0  1  0
## [493,]  1  2  0
## [494,]  0  2  0
## [495,]  0  3  2
## [496,]  1  2  2
## [497,]  0  2  2
## [498,]  0  2  1
## [499,]  0  1  1
## [500,]  0  1  0
```