1 El método de la transformada inversa

Sea X una variable aleatoria continua con función de distribución F. Para simular una realización de X se puede hacer lo siguiente:

1. Generar un número aleatorio U $\sim$ Uniforme(0,1)
2. Tomar X = F^{-1}(U)

2 Weibull

La variable aleatoria Weibull tiene la siguiente función de densidad:

$$f(x;\lambda,k) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} \exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\};$$

su función de distribución de probabilidad es:

$$F(x;\lambda,k) = 1 - \exp\left\{-\left(\frac{x}{\lambda}\right)^{k}\right\}.$$

3 Pareto

La variable aleatoria Pareto tiene la siguiente función de densidad:

$$f(x;a,b) = \frac{ab^a}{(b+x)^{a+1}};$$

su función de distribución de probabilidad es:

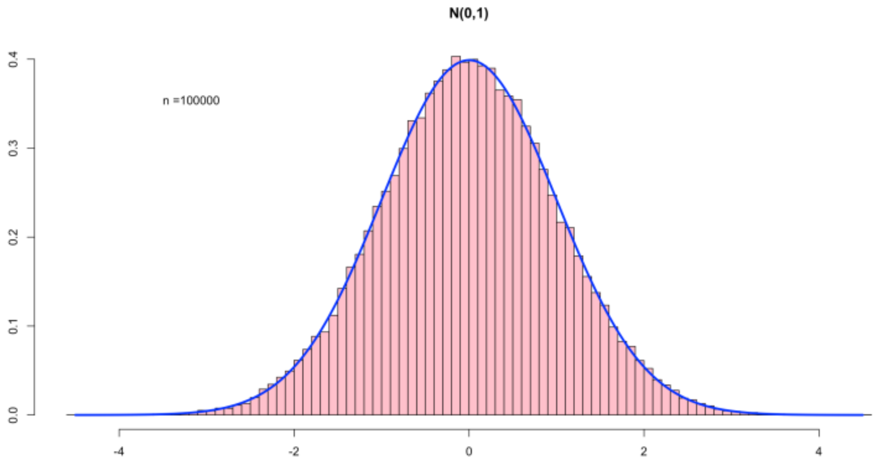
$$F(x;a,b) = 1 - \left(\frac{b}{b+x}\right)^{a}.$$

4 Ejercicios

Para esta práctica deberán utilizar el método de la transformación inversa para simular variables Weibull(1,5) y Pareto(10,3).

En un archivo **PDF** agreguen el procedimiento para encontrar la función F^{-1} para cada caso.

Tambien deberán agregar tres histogramas para cada variable, que corresponderán a tomar un mayor número de uniformes. Usar $n = 100$, $n = 1000$, $n = 100000$, y comparar estos histogramas con la función de densidad teórica. Por lo tanto su reporte deberá tener seis gráficas como la siguiente:



Para esta figura hay que usar *hist()* con el parametro *probability = TRUE*, *par()* con el parámetro *new = TRUE* que permite superponer gráficas y el comando *lines(x,f(x))*, agregará la curva de la densidad dada por f.

Para esta figura hay que tomar en cuenta el soporte, y generar correctamente la x.

Agregar un breve comentario que explique en que situaciones se utilizan estas variables, y finalmente un comentario que explique por que la implementación de este algoritmo sería ineficiente para generar la variable aleatoria $Normal(\mu,\sigma^2)$.

Generación de variables aleatorias a partir del método de la transformada inversa.

Ubicaremos nuestra función de distribución acumulada y obtendremos su función inversa, a partir de ahí podremos generar variables aleatorios que estarán ubicadas en eje x de la función de distribución.

Para hacer la generación de variables aleatorias nosotros propondremos valores pseudoaleatorios entre [0,1] que representar probabilidades de nuestra función de distribución acumulada y usando la función inversa obtendremos sus variables aleatorias.

Procedimiento para encontrar la $F^{-1}(u)$ de la variable aleatoria Weibull:

$$F(x,\lambda,k) = 1 - e^{-(\frac{x}{\lambda})^k}$$
$$u = 1 - e^{-(\frac{x}{\lambda})^k}$$
$$u - 1 = -e^{-(\frac{x}{\lambda})^k}$$
$$1 - u = e^{-(\frac{x}{\lambda})^k}$$
$$ln(1-u) = -(\frac{x}{\lambda})^k$$
$$-ln(1-u) = (\frac{x}{\lambda})^k$$
$$[-ln(1-u)]^{\frac{1}{k}} = \frac{x}{\lambda}$$
$$(\lambda)[-ln(1-u)]^{\frac{1}{k}} = x$$
$$x = (\lambda)[-ln(1-u)]^{\frac{1}{k}}$$

$$x = F^{-1}(u,\lambda,k)$$
$$F^{-1}(u,\lambda,k) = [-ln(1-u)]^{\frac{1}{5}}$$

```
library("SciViews")
#' Funcin de densidad weibul
#'
#' @param x - valor de la variable aleatoria
#' @param lamda - parmetro de forma.
#' @param k - parmetro de escala.
#'
#' @return - probabilidad del valor v.a
densidad_weibul <- function(x, lamda, k) {
  return((k/lamda) * (x/lamda)^(k-1) * exp(-(x/lamda)^k))
}

#' Funcin inversa de la distribucin weibul.
#'
#' @param u - probabilidad acumulada.
#' @param lamda - parmetro de forma.
#' @param k - parmetro de escala.
#'
#' @return variable aleatoria correspondiente.
inversa_distribucion_weibul <- function(u, lamda, k) {
  return(lamda * (-ln(1-u))^(1/k))
}
```

Una vez creadas nuestra función de distribucion y densidad, creamos nuestra funcion que genarará n variables aleatorias:

```
#' Generador de variables aleatorias weibul.
#'
#' @param n - numero de variables.
```

```
#' @param lamda - parmetro de forma.
#' @param k - parmetro de escala.
#'
#' @return vector de variables aleatorias.
generacion_va_weibul <- function(n, lamda, k) {
  va = c()
  for (i in 1:n) {
    u = runif(1)
    va[i] = inversa_distribucion_weibul(u, lamda, k)
  }
  return(va)
}
```

Generaremos un número aleatoria uniforme entre [0,1] que representa la probabilidad acumulada de nuestra función de distribución, depues usaremos la función inversa para generar su correspondiente variables aleatoria.

```
# Histograma para v.a weibul 1
va_weibul = generacion_va_weibul(100, 1, 5)
hist(va_weibul, probability = TRUE, main = 'Weibul n = 100', xlab = '', col = 'salmon2')
par(new=TRUE)
plot(va_weibul, densidad_weibul(va_weibul, 1 ,5), pch = 20, cex = 0.8, col = 'blue')
```

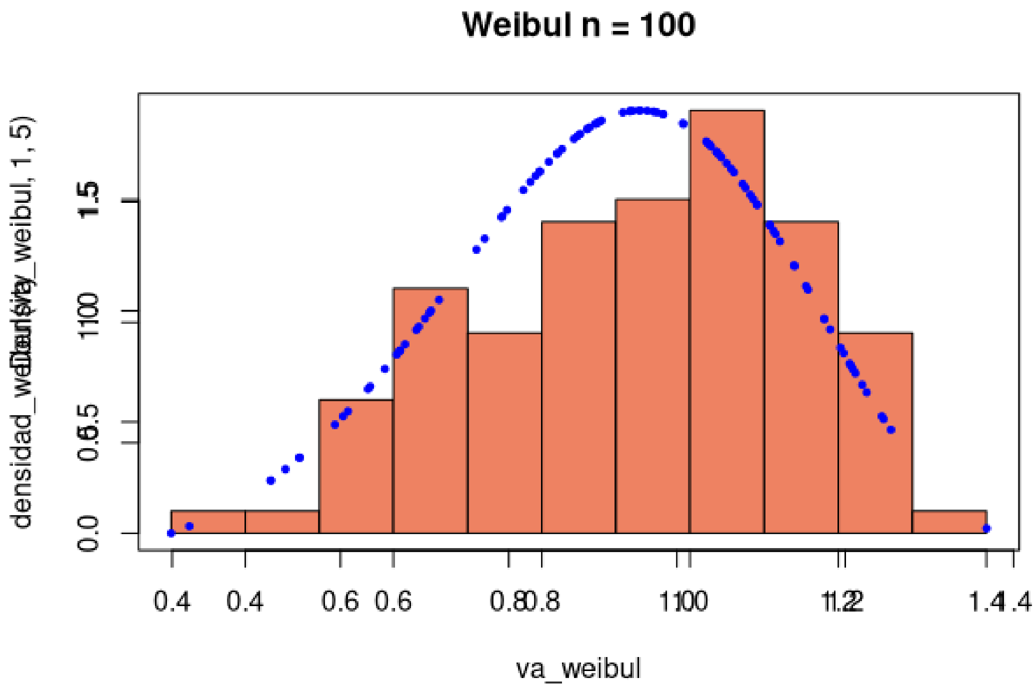


Fig. 1 – *Comparación con n=100*

```
# Histograma 2 para v.a weibul
va_weibul = generacion_va_weibul(1000, 1, 5)
hist(va_weibul, probability = TRUE, main = 'Weibul n = 1000', xlab = '', col = 'salmon2')
par(new=TRUE)
plot(va_weibul, densidad_weibul(va_weibul, 1 ,5), pch = 20, cex = 0.3, col = 'blue')
```

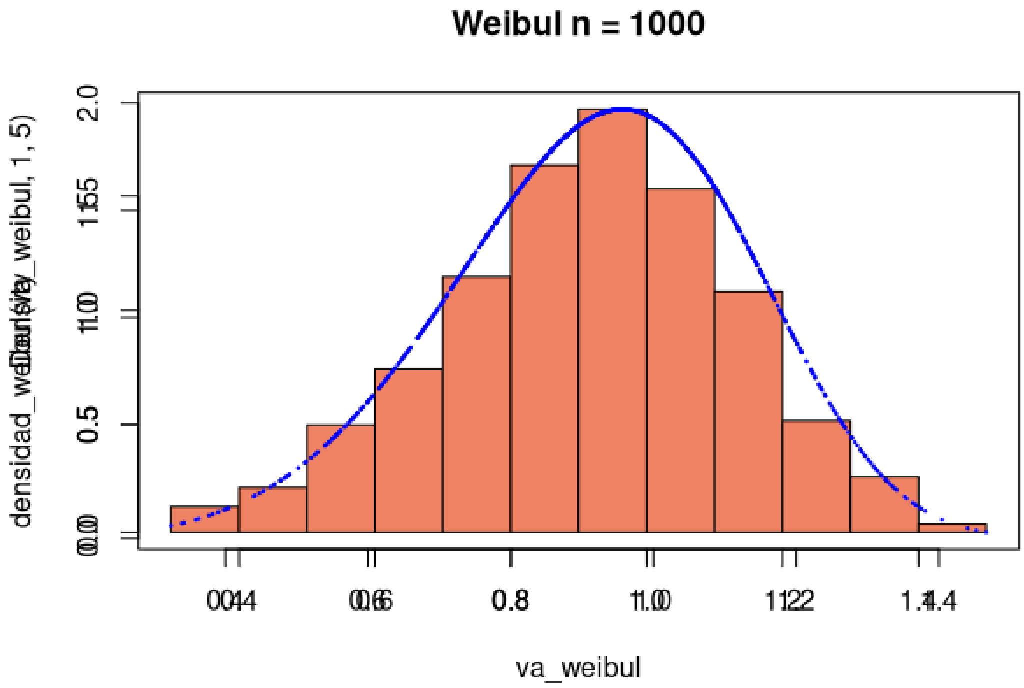


FIG. 2 – *Comparación con n=1000*

```
# Histograma 3 para v.a weibul
va_weibul = generacion_va_weibul(100000, 1, 5)
hist(va_weibul, probability = TRUE, main = 'Weibul n = 100000', xlab = '', col = 'salmon2')
par(new=TRUE)
plot(va_weibul, densidad_weibul(va_weibul, 1 ,5), pch = 20, cex = 0.3, col = 'blue')
```

Vemos que nuestras variables aleatorias generados con el método de la transformación inversa al evaluarlas en las función de densidad y graficarlas, siguen la forma de nuestro histograma, esto quiere decir que nuestras variables aleatorias corresponden precisamente a su distribución.

Procedimiento para encontrar la $F^{-1}(u)$ de la variable aleatoria Pareto:

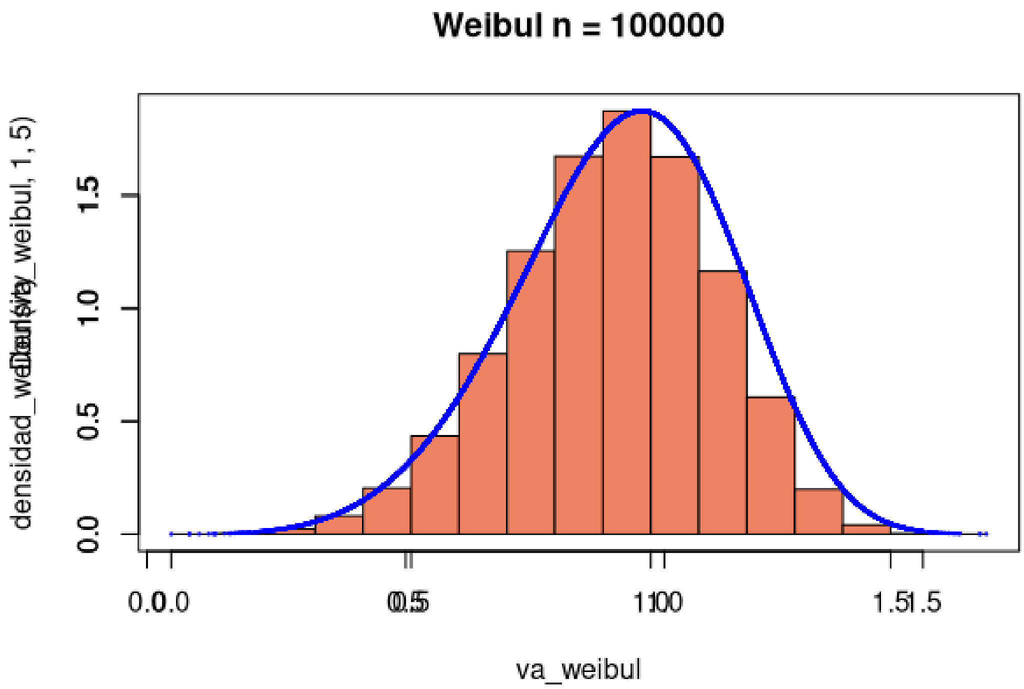


FIG. 3 – *Comparación con n=10000*

$$F(x,a,b) = 1 - (\frac{b}{b+x})^a$$
$$u = 1 - (\frac{b}{b+x})^a$$
$$u - 1 = -(\frac{b}{b+x})^a$$
$$1 - u = (\frac{b}{b+x})^a$$
$$(1-u)^{\frac{1}{a}} = \frac{b}{b+x}$$
$$b + x = \frac{b}{(1-u)^{\frac{1}{a}}}$$
$$x = \frac{b}{(1-u)^{\frac{1}{a}}} - b$$

$$x = F^{-1}(u,a,b)$$
$$F^{-1}(u,a,b) = \frac{3}{(1-u)^{\frac{1}{10}}} - 3$$

```r
#'  Funcin  de densidad pareto.
#'
#' @param x - valor de la variable aleatorio.
#' @param a - es el lmite inferior de los datos.
#' @param b - es el parmetro de forma.
#'
#' @return - probabilidad que suceda el valor de la v.a
densidad_pareto <- function(x, a, b) {
  return((a * b^a)/(b+x)^(a+1))
}

#'  Funcin  inversa de la distribucin pareto
#'
#' @param u - probabilidad acumulada.
#' @param a - es el lmite inferior de los datos
#' @param b - es el parmetro de forma
#'
#' @return variable aleatorio correspondiente.
inversa_distribucion_pareto <- function(u, a, b) {
  return((b/(1-u)^(1/a))-b)
}
```

Generaremos un número aleatoria uniforme entre [0,1] que representa la probabilidad acumulada de nuestra función de distribución, despues usaremos la función inversa para generar sus correspondientes variables aleatoria.

```r
#' Generador de variables aleatorias pareto
#'
#' @param n - numero de variables aleatorias
#' @param a - es el lmite inferior de los datos
#' @param b - es el parmetro de forma
#'
#' @return vector de variables aleatorias.
generacion_va_pareto <- function(n, a, b) {
  va = c()
  for (i in 1:n) {
    u = runif(1)
    va[i] = inversa_distribucion_pareto(u, a, b )
  }
  return(va)
}
```

```r
# Histograma 1 para v.a pareto
va_pareto = generacion_va_pareto(100, 10, 3)
hist(va_pareto, probability = TRUE, main = 'Pareto n = 100', xlab = '', col = 'pink')
par(new=TRUE)
plot(va_pareto, densidad_pareto(va_pareto, 10 ,3), pch = 20, cex = 0.8, col = 'blue')
```

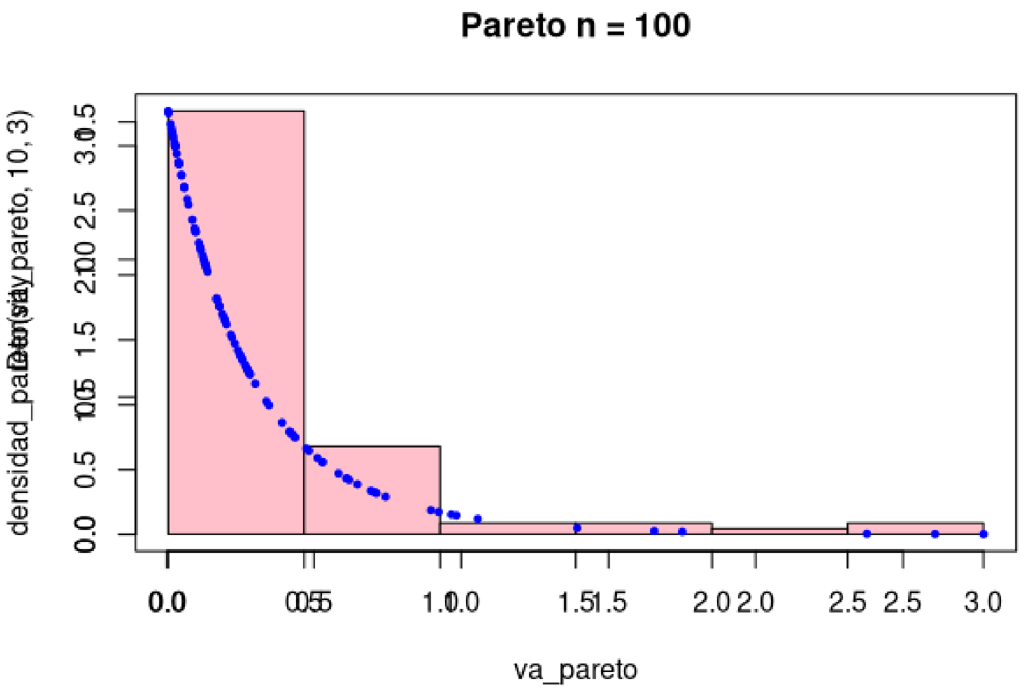


FIG. 4 – *Comparación con n=100*

```r
# Histograma 2 para v.a pareto
va_pareto = generacion_va_pareto(1000, 10, 3)
hist(va_pareto, probability = TRUE, main = 'Pareto n = 1000', xlab = '', col = 'pink')
par(new=TRUE)
plot(va_pareto, densidad_pareto(va_pareto, 10 ,3), pch = 20, cex = 0.3, col = 'blue')
```

```r
# Histograma 3 para v.a pareto
va_pareto = generacion_va_pareto(100000, 10, 3)
hist(va_pareto, probability = TRUE, main = 'Pareto n = 100000', xlab = '', col = 'pink')
par(new=TRUE)
plot(va_pareto, densidad_pareto(va_pareto, 10 ,3), pch = 20, cex = 0.3, col = 'blue')
```

Agregar un breve comentario que explique en que situaciones se utilizan estas variables, y finalmente un comentario que explique por que la implementación de este algoritmo sería ineficiente para generar la variable aleatoria $Normal(\mu,\sigma^2)$.

Las situaciones en las cuales podremos utilizar este procedimiento será cuando nuestra función sea biyectiva, ya que si no es biyectiva no podria tener una función inversa. Este procedimiento es muy util para hacer simulaciones y ver el comportamiento.

Creo que este algoritmo seria ineficiente para la distribución normal por como estaría conformada su función inversa, es decir, una vez conseguido la función de distribución, no

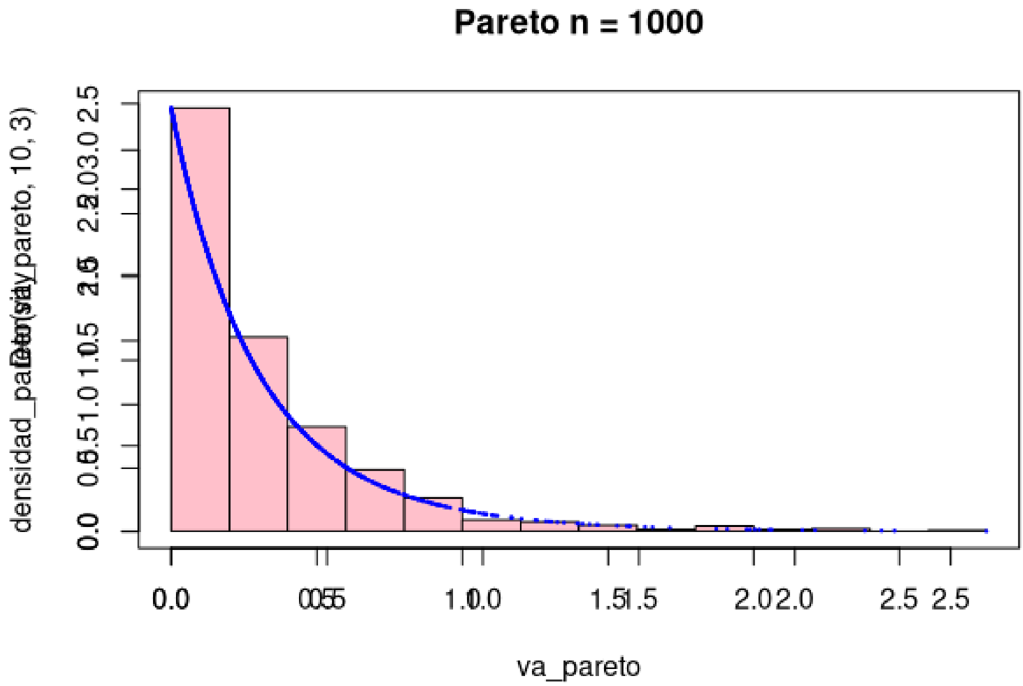


FIG. 5 – *Comparación con n=1000*

es tan sencillo despejar x para conseguir su inversa, vemos que simplemente la integración de su función de distribución suele ser tan compleja que utilizamos tablas para conocer su probabilidad acumulada. Esa seria para mi la principal razón.

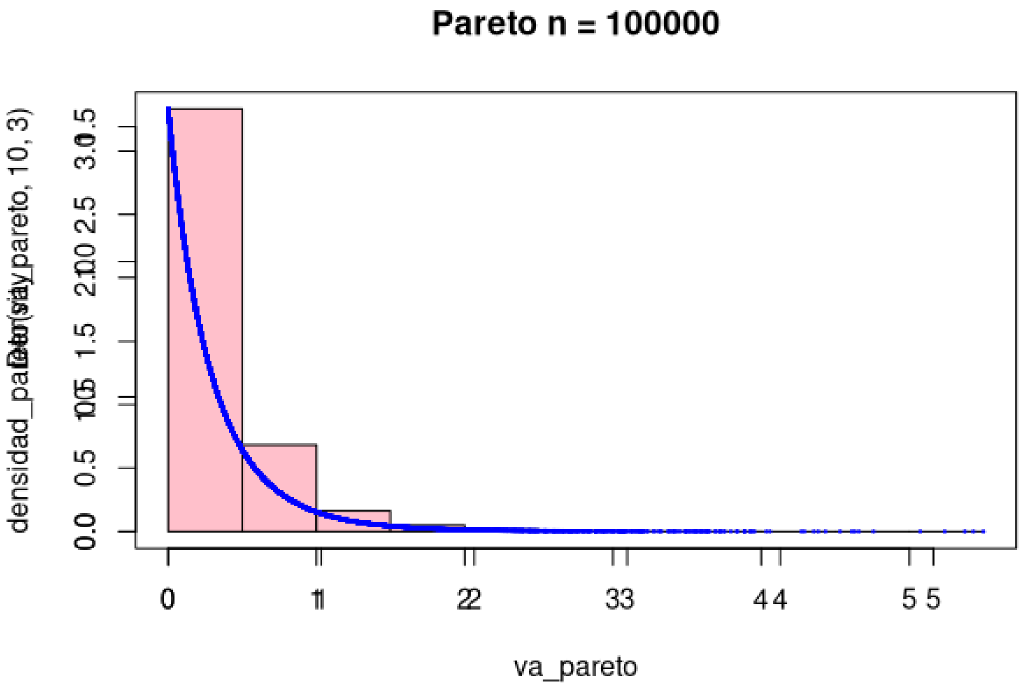


Fig. 6 – *Comparación con n=10000*