Examen

David García Curbelo

2022-05-31

Ejercicio 1

Establecer la semilla del generador de números aleatorios escribiendo `set.seed(1)`. A continuación generar 50 valores desde una distribución Chi-cuadrado con $n = 30$ grados de libertad y almacenarlos en un vector y. Después generar 50 valores desde una distribución Normal con media $\mu = 30$ y desviación típica $\sigma = 5$ y almacenarlos en un vector x. A partir de estos vectores se pide resolver la siguientes tareas:

```
set.seed(1)
y <- rchisq(30, n = 50)
x <- rnorm(50, mean = 30, sd = 5)
```

Ejercicio 1.1

Calcular la media, la desviación típica y los tres cuartiles de y.

```
mean(y)
```

```
## [1] 29.39873
```

```
sd(y)
```

```
## [1] 5.872935
```

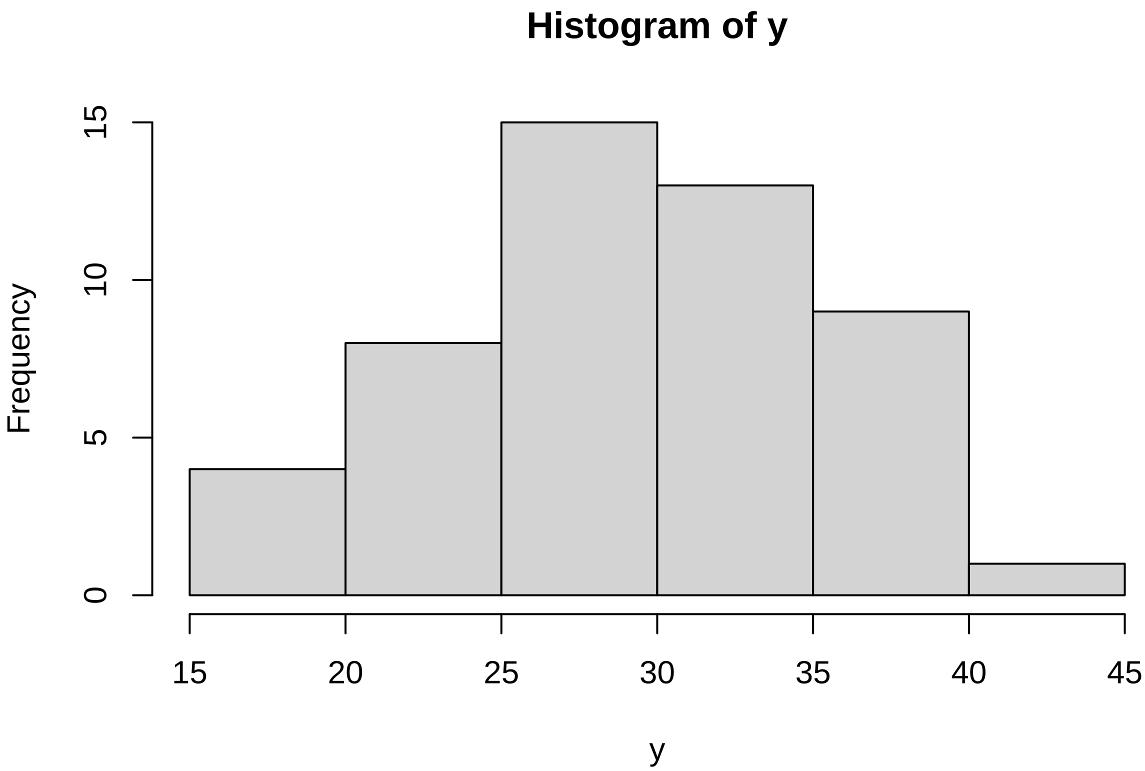
```
quantile(y, probs = c(0.25, 0.5, 0.75))
```

```
##      25%      50%      75%
## 25.51818 28.78737 33.84948
```

Ejercicio 1.2

Representar un histograma de los datos almacenados en y.

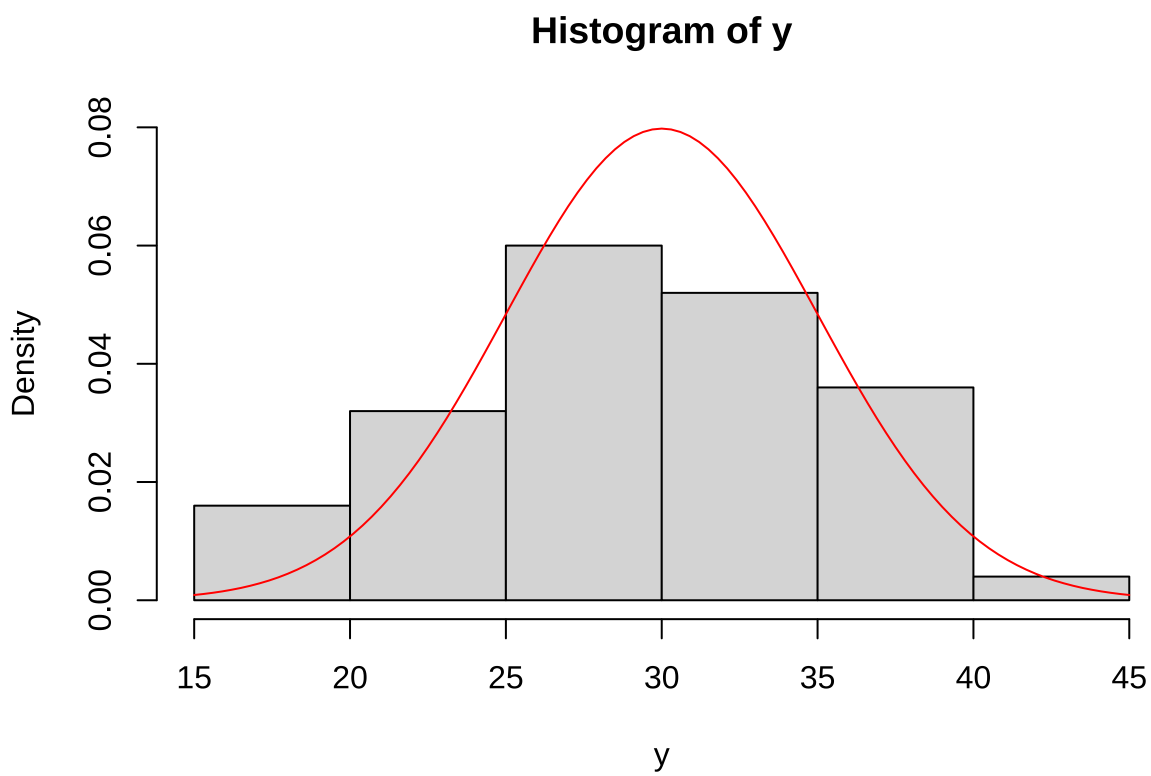
```
hist(y)
```



Ejercicio 1.3

Superponer al histograma anterior la curva de la densidad Normal con media y desviación típica la de los datos.

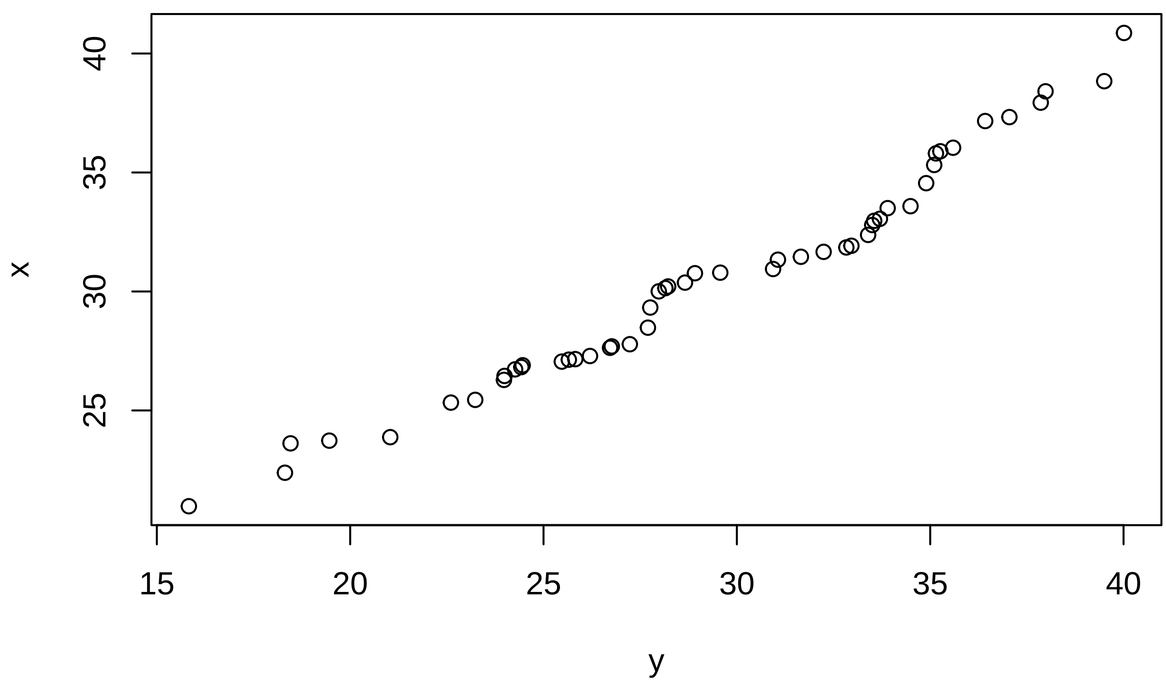
```
hist(y, freq = FALSE, ylim = c(0, 0.08))
curve(dnorm(x, mean = 30, sd = 5), add = TRUE, col = "red")
```



Ejercicio 1.4

Representar un gráfico de probabilidad de los valores de y.

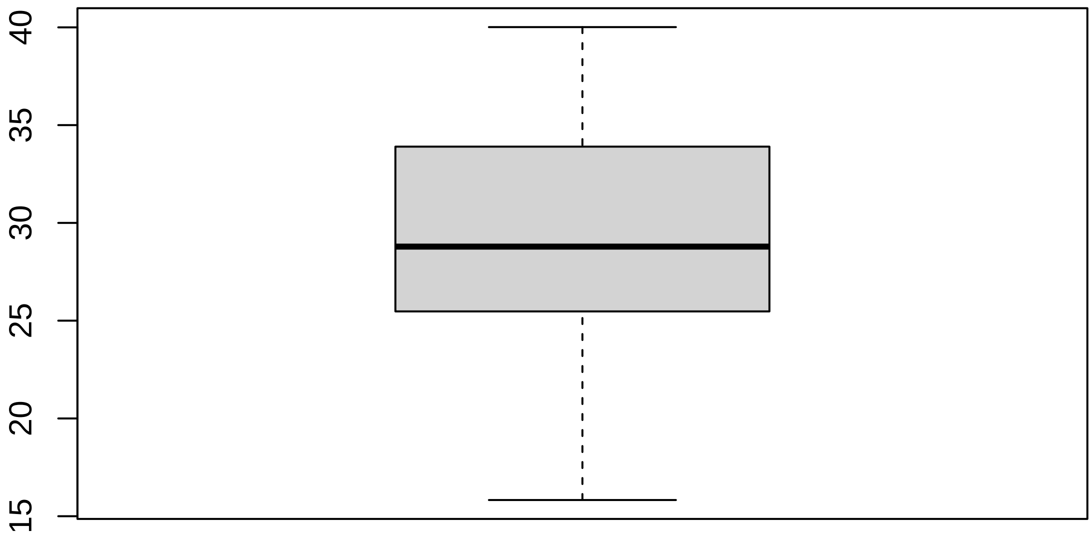
```
qqplot(y,x) # Esto creo que no está bien
```



Ejercicio 1.5

Representar un gráfico de cajas de los valores de y.

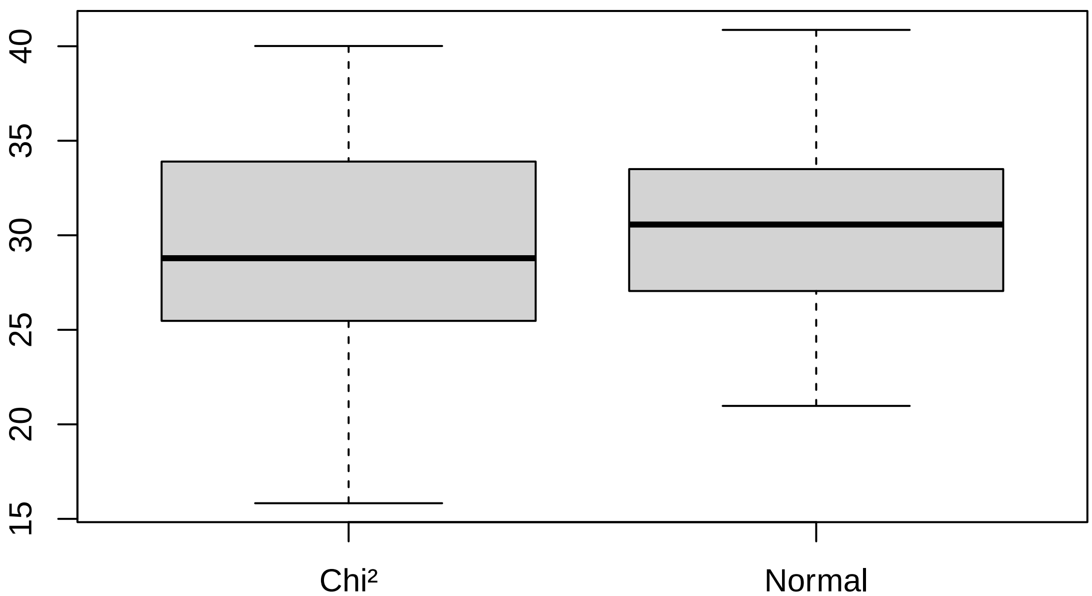
```
boxplot(y)
```



Ejercicio 1.6

Representar un gráfico de cajas múltiples (2 cajas) que permita comparar la distribución de x e y.

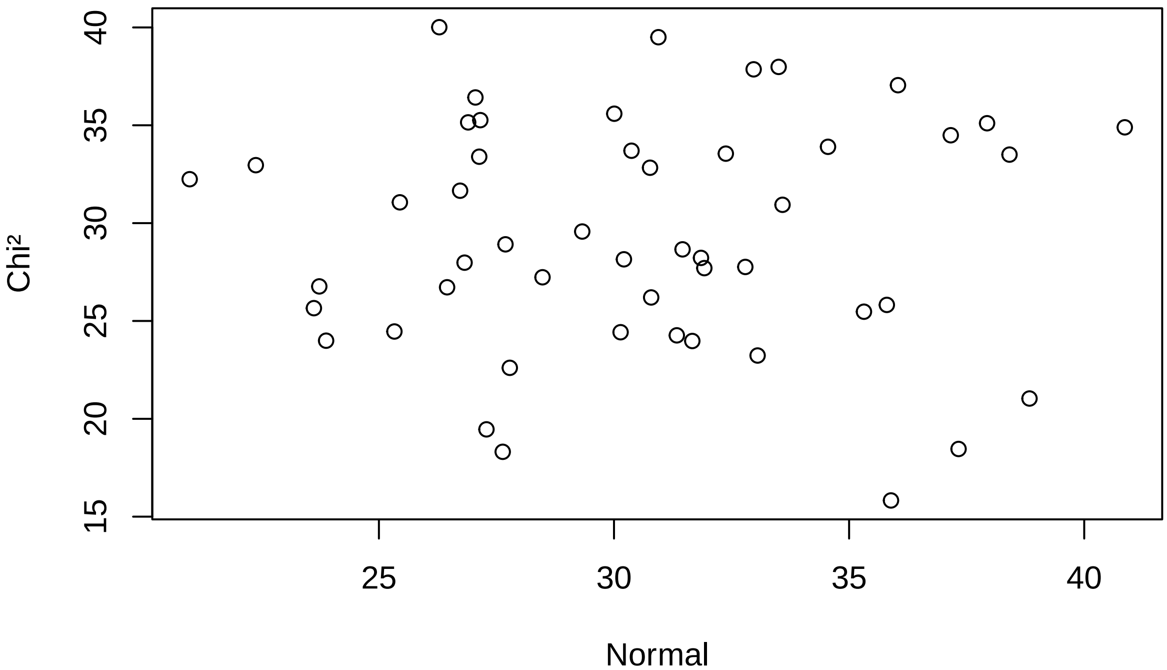
```
boxplot(y, x, names = c("Chi²", "Normal"))
```



Ejercicio 1.7

Representar un diagrama de dispersión de los valores de x (en el eje horizontal) frente a los de y (en el eje vertical).

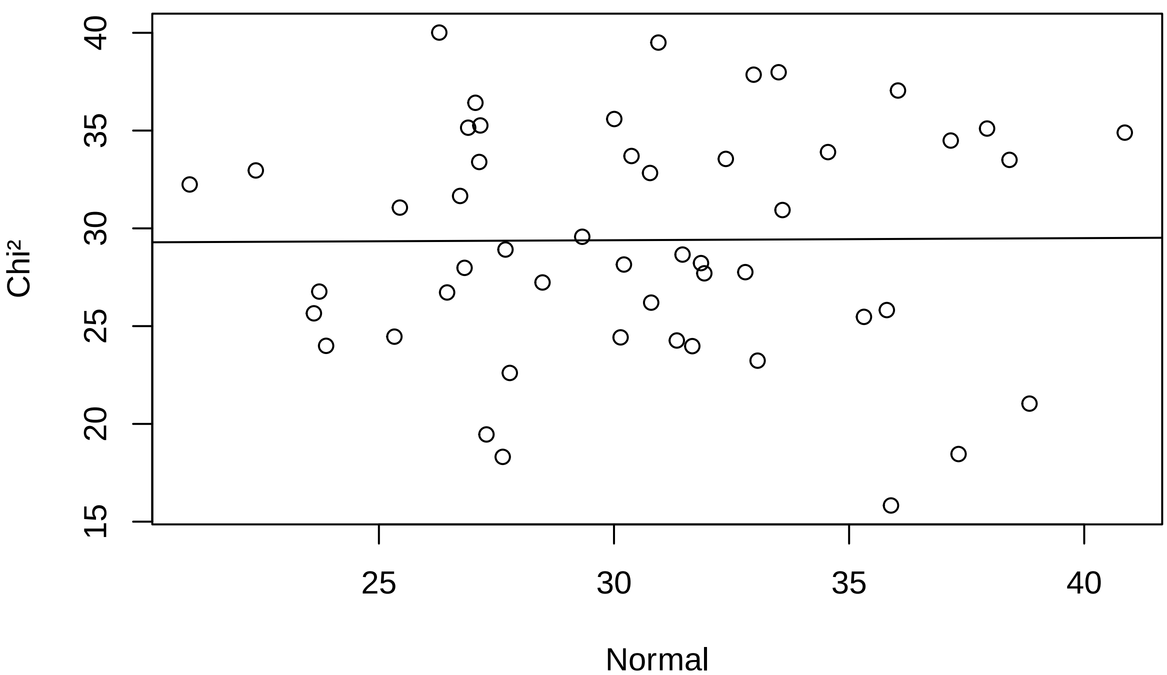
```
plot(x, y, xlab = "Normal", ylab = "Chi²")
```



Ejercicio 1.8

Ajustar un modelo de regresión lineal $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$ $(i = 1, ..., 50)$, con los datos almacenados en x e y. Representar el modelo ajustado (la recta de regresión) sobre el diagrama de dispersión anterior.

```
reg <- lm(y ~ x)
plot(x, y, xlab = "Normal", ylab = "Chi²")
abline(reg, xlab = "Normal", ylab = "Chi²")
```




```
reg
```

```
##
```

```
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##    29.07146      0.01072
```

Ejercicio 2

En 1651, el Caballero de Méré le planteó a Pascal una pregunta relacionada con las apuestas en juegos de azar: ¾sería adecuado apostar a que en cuatro lanzamientos de un dado se obtendrá al menos un seis? Este problema generó una fructífera correspondencia entre Pascal y Fermat que se contribuyó al nacimiento del Cálculo de Probabilidades. Se pide:

Ejercicio 2.1

Escribir una función que simule el lanzamiento de n dados. La función tendrá un único argumento, el número de lanzamientos n, y tomará el valor 4 por defecto. La función devolverá como posibles valores `TRUE`, si se obtiene al menos un 6 y `FALSE` en caso contrario

```r
sim_dados <- function(n=4) {
    n_dados <- sample(1:6, n, replace = FALSE)
    return(6 %in% n_dados)
}
```

Ejercicio 2.2

Utilizar la función anterior para simular `nsim=10000` jugadas de este juego y calcular la proporción de veces que se gana la apuesta "obtener al menos un 6 en `n = 4` lanzamientos". Comparar el resultado con la probabilidad exacta que es $1 - (5/6)^n$.

```r
nsim <- 10000
exacta <- 1 - (5/6)^4
resultado <- numeric(nsim)

for (i in 1:nsim) {
    resultado[i] <- sim_dados()
}

simulada <- mean(resultado)
```

```r
# Valor exacto
exacta
```

```
## [1] 0.5177469
```

```r
# Valor simulado
simulada
```

```
## [1] 0.667
```

Ejercicio 3

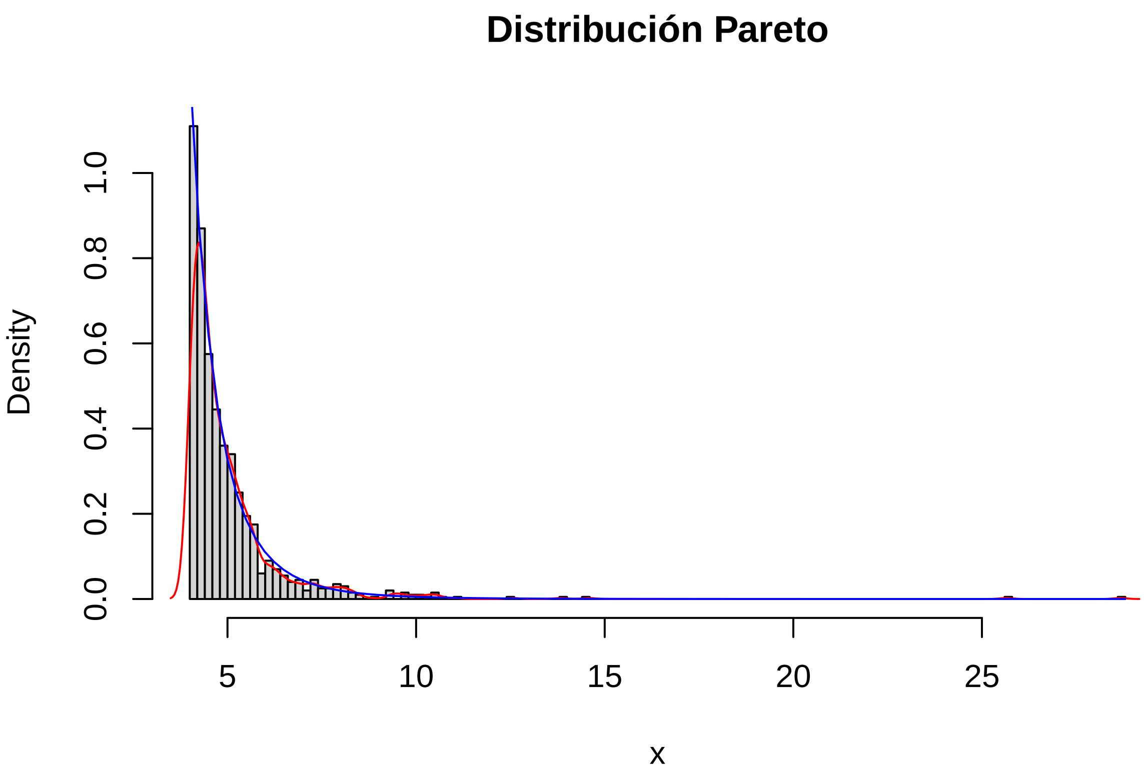
Utilizando el método de inversión se pide generar $n = 1000$ valores de un distribución Pareto de parámetros $a = 5$ y $b = 4$. Evaluar usando gráficos y el contraste de Kolmogorov-Smirknow que en efecto los valores generados provienen de dicha distribución.

```r
a <- 5
b <- 4
n <- 1000
u <- runif(n)

Pareto <- function(x, a, b) {
    x[x < b] <- b
    return((a*b^a)/ (x^(a+1)))
}

x <- b / (1 - u)^(1/a)

hist(x, freq = FALSE, breaks = 'FD', main = 'Distribución Pareto')
lines(density(x), col = 'red')
curve(Pareto(x, a, b), col = 'blue', add = TRUE)
```




```r
# Test de Kolmogorov-Smirnov
ks.test(x, Pareto(x, a, b))
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  x and Pareto(x, a, b)
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

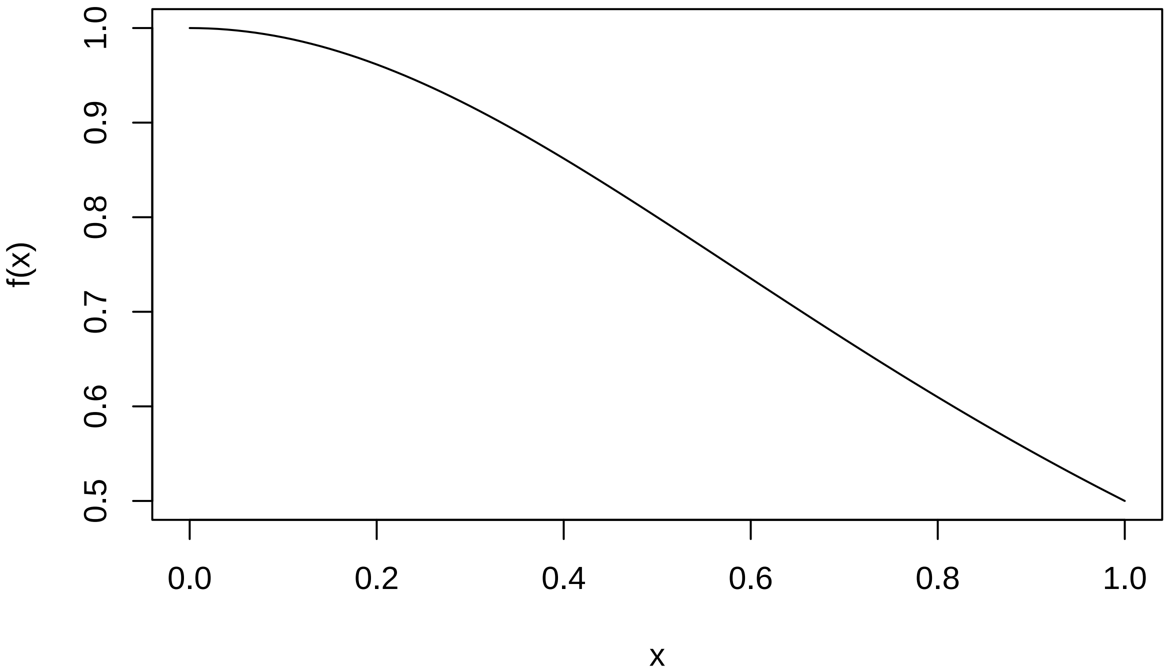
Ejercicio 4

Utilizando integración de Monte Carlo se pide aproximar la integral

$$\int_0^1 \frac{1}{1+x^2}dx$$

y calcular el error de la aproximación. Considerar para ello un número suficientemente grande de simulaciones, evaluando la convergencia mediante un gráfico. Como referencia tener en cuenta que el valor exacto de la integral es $\pi/4$.

```r
f <- function(x) {
    return(1/(1+x^2))
}
curve(f(x), 0, 1)
```




```r
nsim <- 100000
x <- runif(nsim) # por defecto en R está entre 0 y 1

sx <- sapply(x, f)
mx <- mean(sx)
```

```r
# Valor exacto
pi/4
```

```
## [1] 0.7853982
```

```r
# Valor simulado
mx
```

```
## [1] 0.7857867
```