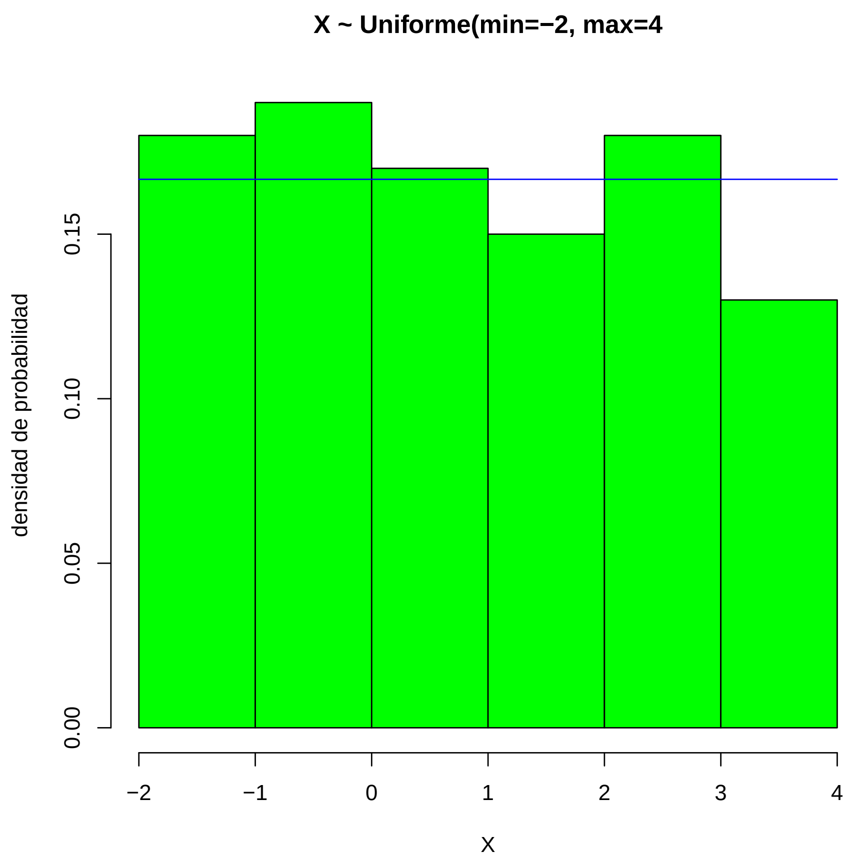
```r
#GUIA 15

x <- 55; a=0; b <- 90
#usando la funcin propia de R
punif(x, min=a, max=b, lower.tail=TRUE)

## [1] 0.6111111

F55=punif(55, min=a, max=b, lower.tail=TRUE)
F15=punif(15, min=a, max=b, lower.tail=TRUE)
F55-F15

## [1] 0.4444444

F55=punif(55, min=a, max=b, lower.tail=TRUE);F55

## [1] 0.6111111

#Luego multiplicando ambas probabilidades se obtiene el valor pedido 0.1728.
(1-F55)*( F55-F15)

## [1] 0.1728395

#y los cuantiles-normales para la variable X:
p <- c(0.80); media=5; d.t=1
qnorm(p, mean=media, sd=d.t, lower.tail=TRUE)

## [1] 5.841621

#y los cuantiles-t para la variable Y:
p <- c(0.80); g.l <- 10
qt(p, df=g.l, lower.tail=TRUE)

## [1] 0.8790578

#Como se desea calcular P(x ??? 4.5) :
n <- 16; x <- 4.5; mu=5; sigma=1; d.t=sigma/sqrt(n)
pnorm(x, mean=mu, sd=d.t, lower.tail=FALSE)

## [1] 0.9772499

#La probabilidad P(X ??? 5) se obtiene as:
x <- 5; teta=7
pexp(x, rate=1/teta, lower.tail=FALSE)

## [1] 0.4895417

#y de igual forma P(X < 3) :
x <- 3; teta=7
pexp(x, rate=1/teta, lower.tail=TRUE)
```

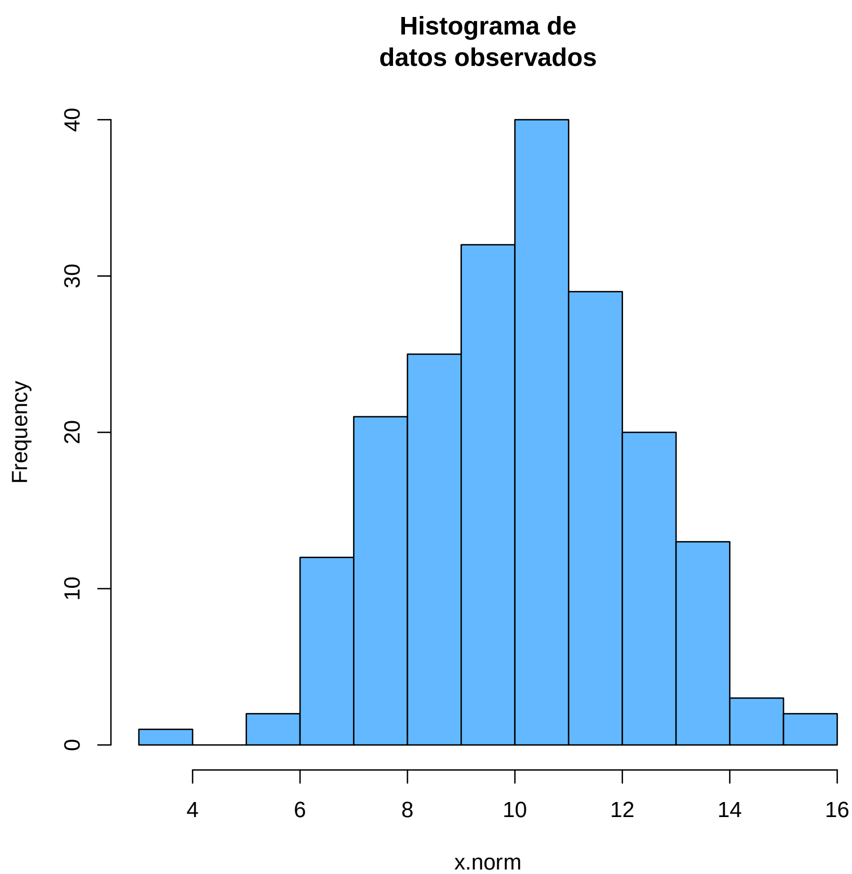
```
## [1] 0.3485609

pexp(4, rate=1/teta, lower.tail=FALSE)

## [1] 0.5647181

#Hay que calcular el percentil 90:
p <- 0.9; teta <- 7
qexp(p, rate=1/teta, lower.tail=TRUE)

## [1] 16.1181

#resultando 16.12 aos.

qexp(0.5, rate=1/teta, lower.tail=TRUE)

## [1] 4.85203

#y en el segundo caso, el percentil 68, b = 7.97
qexp(0.68, rate=1/teta, lower.tail=TRUE)

## [1] 7.97604

#o de esta otra manera
qexp(0.32, rate=1/teta, lower.tail=FALSE)

## [1] 7.97604

# Definir los parmetros apropiados
min <- -2; max <- 4
# generar 100 nmeros aleatorios de la distribucin
x = runif(100, min, max); x

##   [1] -1.78672151  0.85148167  1.12108331  2.94604945  1.85549998
##   [6] -0.10239208  2.74980291  2.80787862 -1.52833875  0.40530748
##  [11]  2.05157856  3.70321236 -1.60565074 -0.87015618  0.92881572
##  [16] -1.55256795 -0.61725472  3.04658179  0.82549123 -0.30499210
##  [21]  1.56175958  2.63558638 -1.54437910  1.59376547  0.33532904
##  [26]  2.04604289  1.43658819 -1.87119228  2.16550460  0.34940446
##  [31]  1.78040802 -0.75552233 -0.41103175 -0.66404002  0.20232600
##  [36] -0.59240274  2.60445342  3.57645639  3.71405089  2.94699732
##  [41] -0.18213900  1.01361124  2.93989904  3.33712766  1.74952192
##  [46]  3.34520821 -0.02171428  3.02379991 -1.34041660 -0.51851305
##  [51]  0.16958473  3.50735721 -0.95405323 -0.90590808  1.25112336
##  [56]  0.15536461  0.66958083 -0.32527305  1.93682252  1.01384368
##  [61]  3.92737085  1.54075329  0.52211769  3.28633773 -0.59121128
##  [66]  1.45632177  2.32490851  0.35540944  0.55875677  1.50443290
##  [71] -1.35523254 -1.86720029  0.10356513 -0.90074222 -1.17610321
```

```
## [76]  1.88856098 -0.20652818  3.13401175 -0.60038070  0.23658700
## [81]  3.18489457 -1.03450437 -0.18819307  2.42355529  2.15219891
## [86]  2.01921593 -1.10257699 -1.17598028 -1.18196594  2.34052945
## [91]  0.03049696 -1.87056786 -1.12698389  2.90228020  0.42798421
## [96]  2.33321620  3.56270414 -1.70632363 -1.42930933  2.96808952

# Histograma para la nuestra aleatoria de tamao 100
hist(x, main="X ~ Uniforme(min=-2, max=4", xlab="X", ylab="densidad de probabilidad",
probability=TRUE, col="green")
# Graficar la funcin de densidad, use la funcin curve() para variable continua
curve(dunif(x, min, max), col="blue", add=TRUE)
```

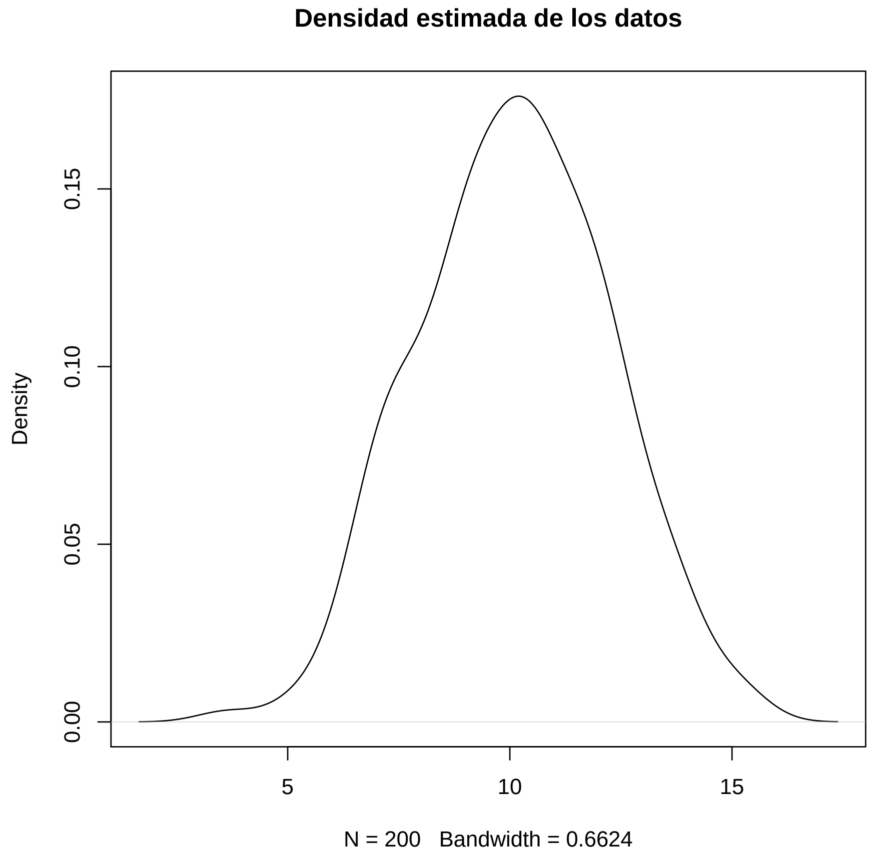



```
#genera los valores aleatorios de la distribucin
x.norm <- rnorm(n=200,mean=10, sd=2)

# Podemos obtener un histograma usando la funcin hist()
```

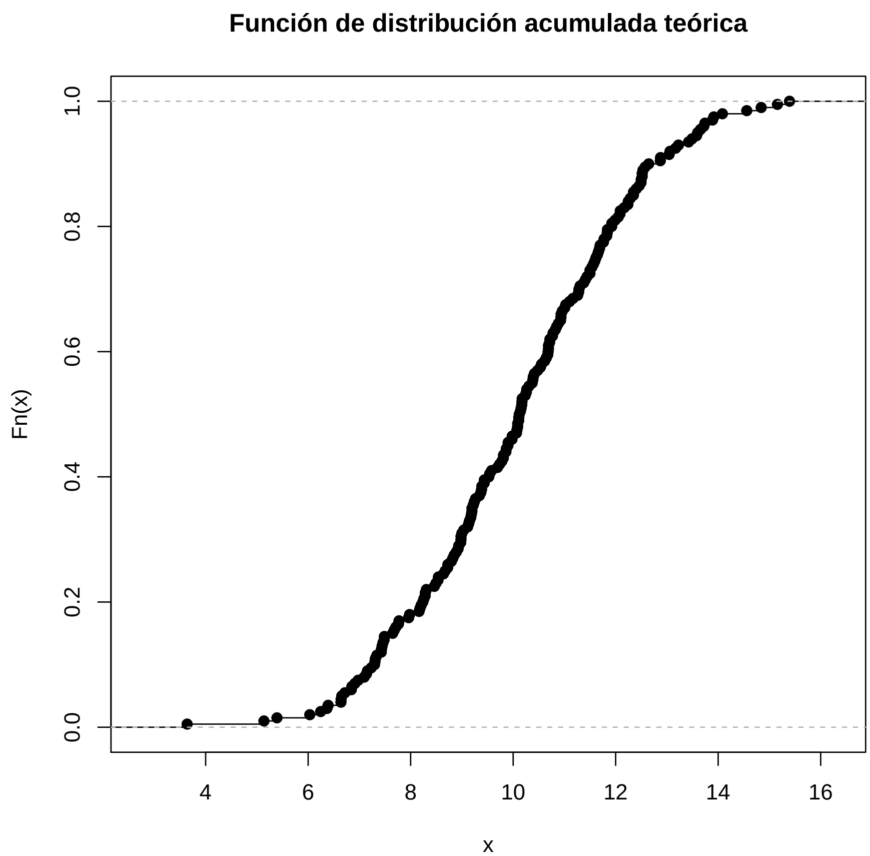
```r
hist(x.norm, breaks = "Sturges", freq = TRUE, probability = FALSE, include.lowest = TRUE, ri
= TRUE, density = NULL, angle = 45, col = "steelblue1", border = NULL, main = "Histograma de
datos observados", axes = TRUE, plot = TRUE, labels = FALSE)
```




```r
plot(density(x.norm), main="Densidad estimada de los datos")
```



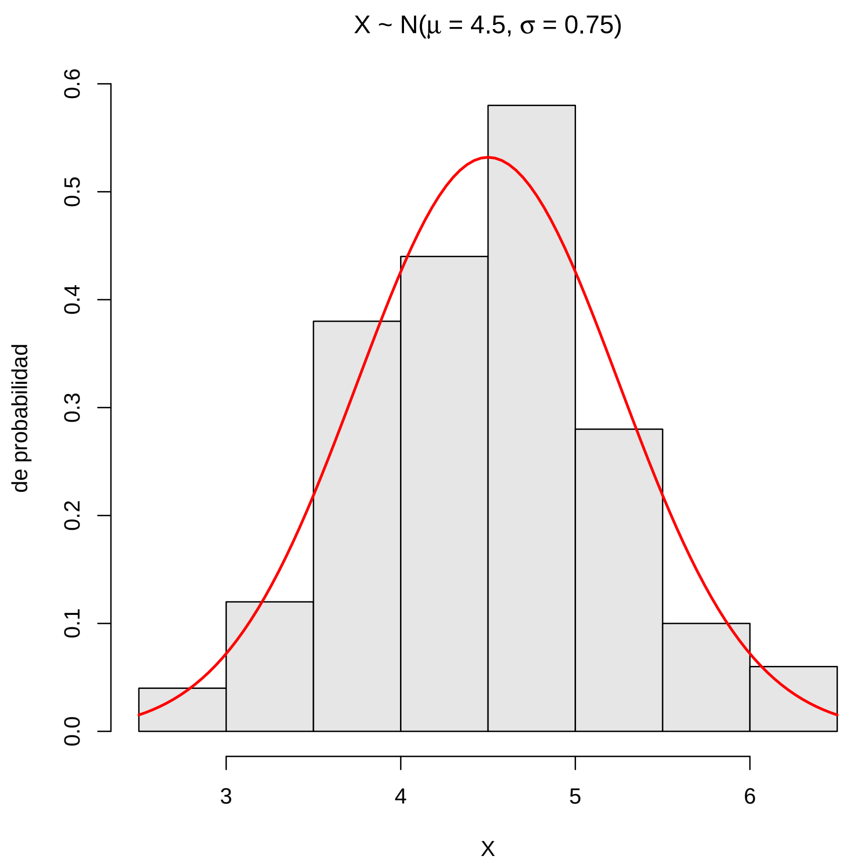

```
plot(ecdf(x.norm),main="Funcin de distribucin acumulada terica")
```




```
# Definir los parmetros apropiados
media <- 4.5; desviacion <- 0.75
#Generar 100 nmeros aleatorios de la distribucin
x = rnorm(100, media, desviacion); x

##   [1] 4.029920 5.821386 4.641438 3.449032 3.627814 3.935307 3.758119
##   [8] 5.915371 3.792341 3.529290 4.584295 4.024424 4.249860 4.009177
##  [15] 5.070156 4.204805 3.684109 2.772989 4.377355 5.917758 3.018638
##  [22] 4.568907 4.569531 6.004569 4.734568 3.371605 4.483181 4.290884
##  [29] 4.634000 3.555631 4.935092 6.247813 4.930580 4.712999 4.694150
##  [36] 4.535441 4.058150 5.181461 4.363378 4.548020 5.349070 4.423880
##  [43] 3.822976 3.297098 4.842226 4.455268 4.818021 5.270818 3.815976
##  [50] 4.108719 4.029195 4.141446 3.919951 3.609675 3.780259 5.338455
##  [57] 5.214300 4.233535 6.347621 3.723949 5.075039 3.744412 4.758563
##  [64] 3.894228 3.383547 3.886360 5.796572 4.351904 4.872332 4.528315
##  [71] 4.531842 4.927489 5.464806 5.025385 4.082680 3.297737 5.214124
```

```
##  [78] 4.084597 4.566551 4.445609 4.622740 4.576680 5.052774 3.797365
##  [85] 3.654596 5.209509 2.983339 5.280811 4.795568 4.900734 4.049733
##  [92] 4.624722 5.341009 4.282888 5.785911 3.736081 4.949583 4.593777
##  [99] 4.613655 4.797490
```

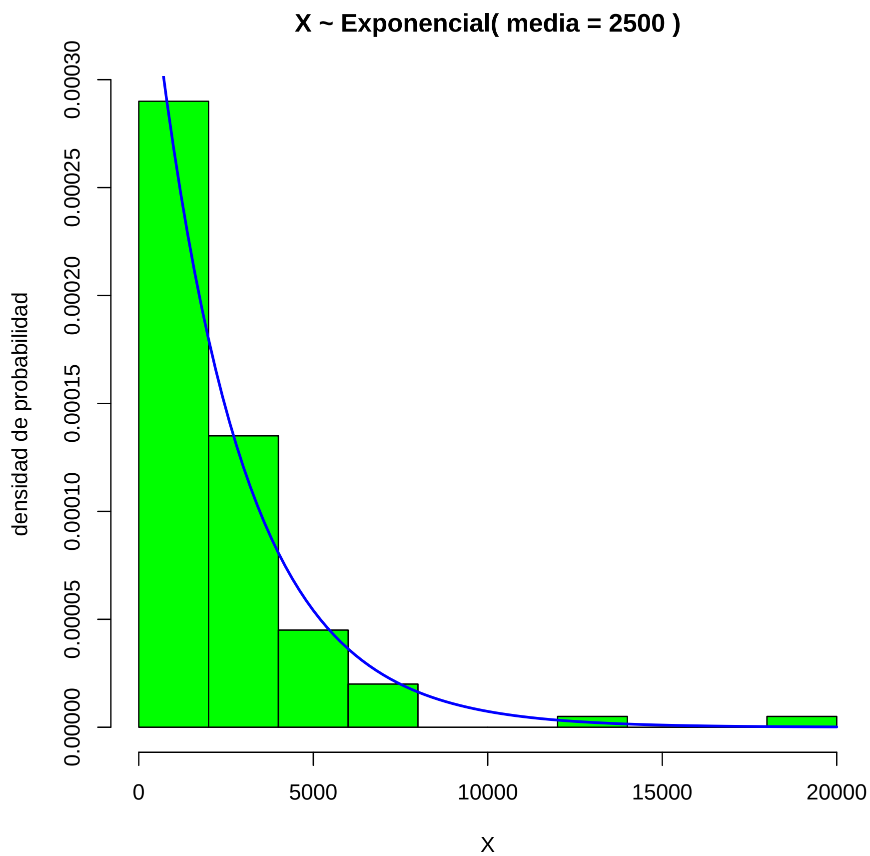
```
# Histograma para la nuestra aleatoria de tamao 100
hist(x,main=expression(paste("X ~ N(", mu, " = 4.5, ", sigma, " = 0.75)")), xlab="X", ylab='
de probabilidad", probability=TRUE, col=gray(0.9))
curve(dnorm(x, media, desviacion), col="red", lwd=2, add=TRUE)
```




```
# Definir el parmetro apropiado
media <- 2500; razon <- 1/media;n=100
# generar 100 nmeros aleatorios de la distribucin
x = rexp(n, razon); x
```

```
##   [1] 19760.35572   307.75717  5619.60345  2559.45081  4001.14564
```

```
##   [6]  1176.34429 13891.49632  1175.76155   389.79929  3242.60891
##  [11]   712.13648   862.90596  1895.92928  6033.00871  4178.98859
##  [16]  3213.49653  2831.33511  1862.33767  3790.25644   102.35523
##  [21]   572.85660  1362.69657    41.77218  1409.35913  2116.03617
##  [26]  2873.05052   909.01085  4016.74853   834.97153   897.77319
##  [31]  2764.45945   766.67769   926.10437    34.80731   864.51839
##  [36]  2185.20984   828.51697  1379.21548   697.26236   605.44715
##  [41]  1942.30793  1041.32961   137.68112  5635.62383  1017.62088
##  [46]  6117.64887  6605.00451  1823.57488  3268.44387  1161.68630
##  [51]  1325.85331   985.74760   971.46592  2570.25076   661.17817
##  [56]  1659.41369  1432.45270   401.57865  5781.98247  3077.51985
##  [61]   442.29178   257.39089  2178.98322   226.08883  2105.56829
##  [66]  2731.48244  1979.25972  4852.03465  3176.73497  3674.89756
##  [71]   901.98397  1117.61819  2445.20432  4029.85698  2136.55071
##  [76]  5719.89317  1194.77528  2232.32886  3677.43191   386.57073
##  [81]  1232.50977   902.41699   338.88838  1503.28907   803.62819
##  [86]  2195.38963   873.26272  3323.61484   152.24885  7138.52412
##  [91]   552.95901  3611.55573  1260.97157  1769.35024  3534.44546
##  [96]  1579.08208  2503.20230  3371.44982   269.78498  1037.28200

# Histograma para la nuestra aleatoria de tamao 100
hist(x, main="X ~ Exponencial( media = 2500 )", xlab="X", ylab="densidad de probabilidad",
probability=TRUE, col="green")
curve(dexp(x, razon), col="blue", lwd=2, add=TRUE)
```




```
x <- 0.7
p <- pnorm(x, mean=1, sd=1, lower.tail = TRUE); p

## [1] 0.3820886

z <- 0.7
p1 <- pnorm(z, mean=0, sd=1); p1

## [1] 0.7580363

p2 <- pnorm(z, mean=0, sd=1, lower.tail=FALSE); p2

## [1] 0.2419637

p3 <- 1-pnorm(z, mean=0, sd=1);p3

## [1] 0.2419637
```

```r
p <- 0.75
z <- qnorm(p, mean=0, sd=1, lower.tail = TRUE); z

## [1] 0.6744898

x <- 18.55; gl <- 12
p <- pchisq(x, gl, lower.tail = FALSE); p

## [1] 0.09998251
```