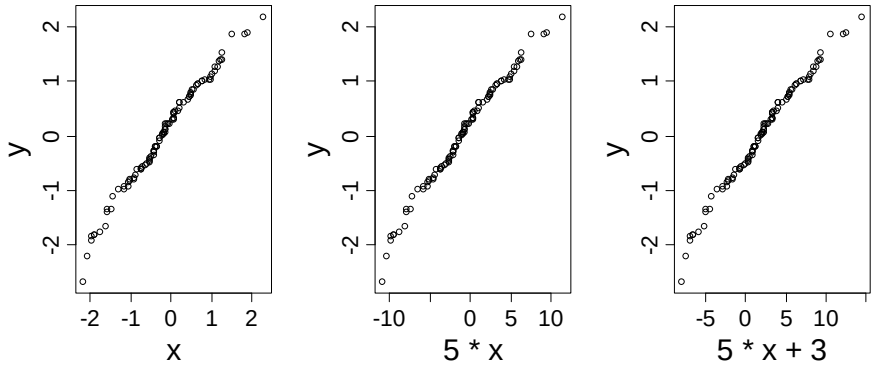
Gráficos Cuantil-Cuantil (Q-Q plots)

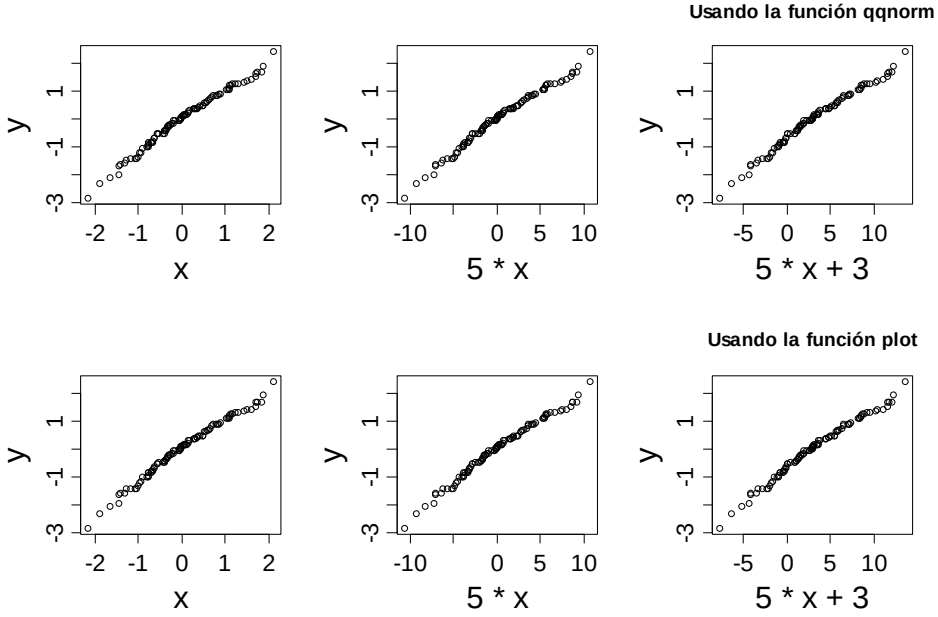
Un gráfico Cuantil-Cuantil permite observar cuan cerca está la distribución de un conjunto de datos a alguna distribución teórica ó comparar la distribución de dos conjuntos de datos.

Comparación de la distribución de dos conjuntos de datos

La función **`qqplot`**(x, y, plot=T) produce un diagrama de dispersión de la función **`quantile`** de los valores de un conjunto de datos versus la función **`quantile`** de los valores del otro conjunto. Vemos que el Q-Q plot no cambia por una transformación lineal de los datos.




```
set.seed (14)
x <- rnorm(100 )
y <- rnorm(100 )
  par(mfrow=c(2,3))
  qqplot(x,y ,cex.lab=2,cex.axis=1.5)
  qqplot(5*x,y,cex.lab=2,cex.axis=1.5 )
  qqplot(5*x+3,y,cex.lab=2,cex.axis=1.5)
```




```
set.seed (14)
x <- sort(rnorm(100 ))
y <- sort(rnorm(100 ))
```

```
plot(x,y ,cex.lab=2,cex.axis=1.5)
 plot(5*x,y,cex.lab=2,cex.axis=1.5 )
 plot(5*x+3,y,cex.lab=2,cex.axis=1.5)
```

Comparación de la distribución de un conjunto de datos con una distribución teórica

Por ejemplo, si interesa comparar la distribución de un conjunto de datos con la distribución gaussiana se ordenan los datos y se grafica el i-ésimo dato contra el correspondiente cuantil gaussiano.

Hoaglin, Mosteller y Tukey (1993) sugieren tomar el i-esimo cuantil como:

$$\phi^{-1}\left(\frac{i-1/3}{n+1/3}\right)$$

La función **qqnorm** reemplaza una de las muestras, en **qqplot,** por los cuantiles de la distribución normal.

Ejemplo detallado de un gráfico Cuantil-Cuantil

Comparamos los percentiles empíricos de un conjunto de datos, con los percentiles teóricos de una Normal
Los datos se encuentran en el archivo ***ejemploQQ.txt***.

A partir de ellos se genera el data.frame **ejemploQQ** que inicialmente contiene una única variable **V1**

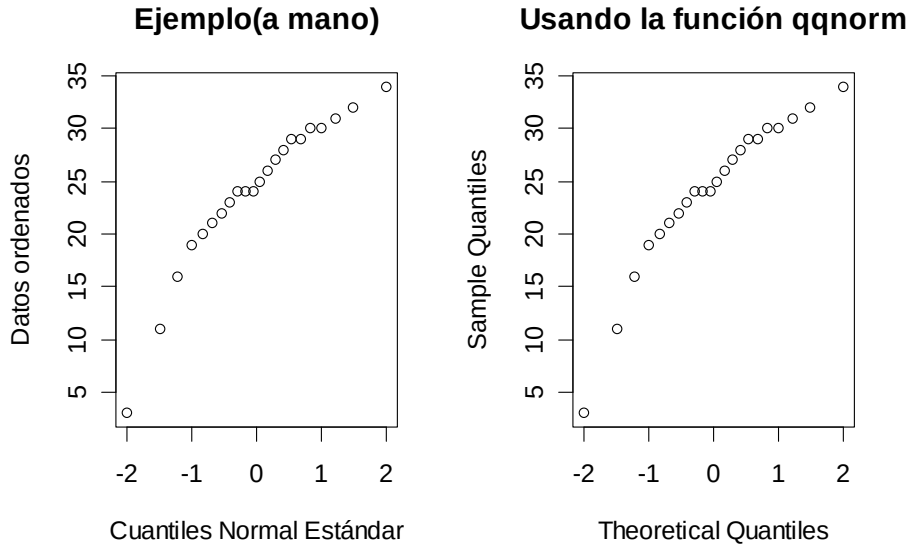
Datos	Posición i	pi (i-0.5) / n	phi^{-1} (pi)	Datos	Posición i	pi (i-0.5) / n	phi^{-1} (pi)
3	1	0.02	-2.00	25	12	0.52	0.06
11	2	0.07	-1.49	26	13	0.57	0.17
16	3	0.11	-1.21	27	14	0.61	0.29
19	4	0.16	-1.00	28	15	0.66	0.41
20	5	0.20	-0.83	29	16	0.70	0.54
21	6	0.25	-0.67	29	17	0.75	0.67
22	7	0.30	-0.54	30	18	0.80	0.83
23	8	0.34	-0.41	30	19	0.84	1.00
24	9	0.39	-0.29	31	20	0.89	1.21
24	10	0.43	-0.17	32	21	0.93	1.49
24	11	0.48	-0.06	34	22	0.98	2.00

```
> ejemploQQ <- read.table(file.choose(), header=TRUE)
> ejemploQQ
   V1
1   3
2  11
3  16
4  19
5  20
```

```
.......

ejemploQQ$V1 <- sort(ejemploQQ$V1)

n <- length(ejemploQQ$V1)
a <- 0.5
```

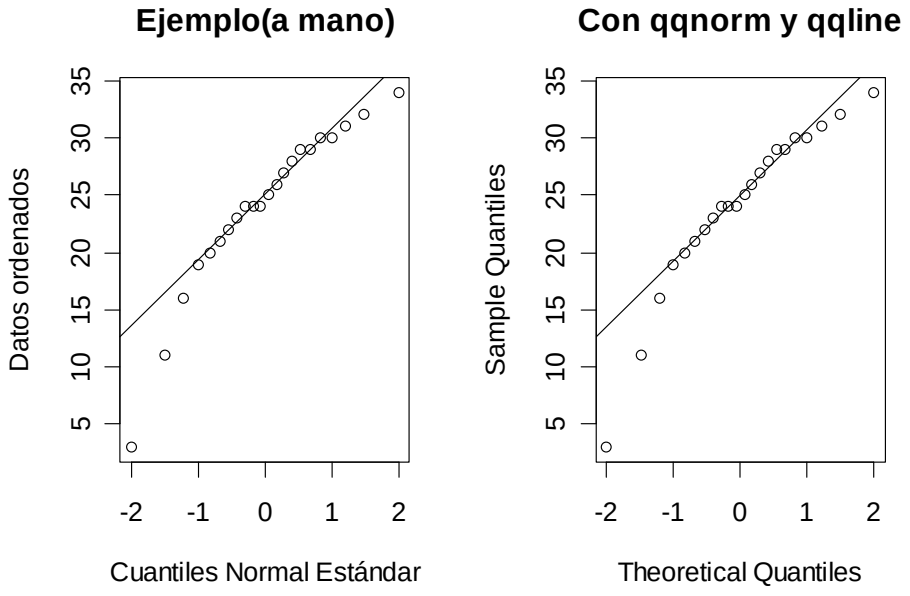
Agregamos las variables, **pi** (vector de probabilidades que asignamos a los estadísticos de orden i) y **phiInv** (cuantiles teóricos Gaussianos de esas **pi**), al data frame

```
  ejemploQQ$pi  <- (1:n-a)/(n+1-2*a)
  ejemploQQ$phiInv  <-
qnorm(ejemploQQ$pi)

  par(mfrow=c(1,2))

  plot(ejemploQQ$phiInv,ejemploQQ$V1,
      xlab="Cuantiles Normal
   Estándar",
      ylab="Datos ordenados",
   main="Ejemplo(a mano)"
   )

 qqnorm(ejemploQQ$V1)
 title("Usando la función qqnorm")
```


```
# agrego qqline

plot(ejemploQQ$phiInv,ejemploQQ$V1,
     xlab="Cuantiles Normal Estándar",
     ylab="Datos ordenados",
main="Ejemplo(a mano)"
 )

datos.cuartiles <-
quantile(ejemploQQ$V1, c(0.25,0.75))
norm.cuartiles <- qnorm(c(0.25, 0.75))
```

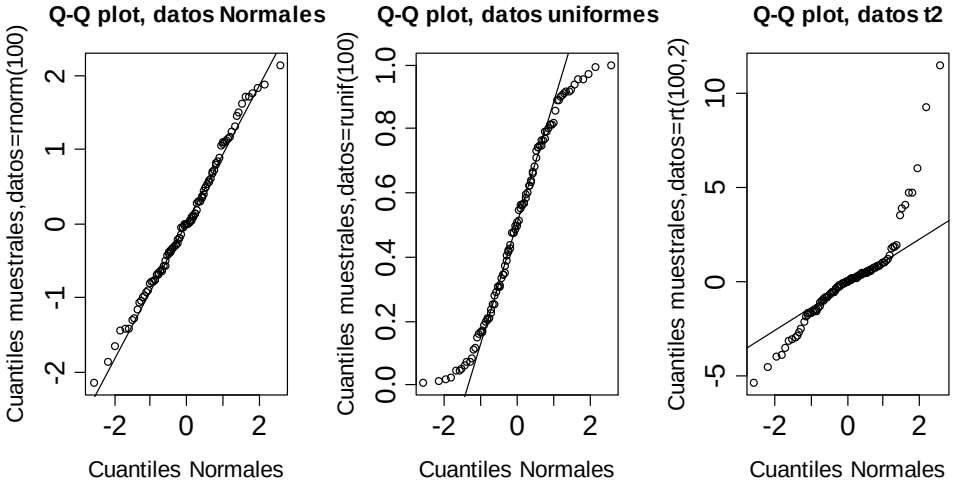
```
b <- (datos.cuartiles[2] -
   datos.cuartiles[1] )
         /
     (norm.cuartiles[2]  -
   norm.cuartiles[1])
a <- datos.cuartiles[1] -
   norm.cuartiles[1] * b

abline(a, b)
qqnorm(ejemploQQ$V1,main="")
title("Con qqnorm y qqline")
qqline(ejemploQQ$V1)
```



Función qqline

```
> qqline
function (y, datax = FALSE, ...)
{
    y <- quantile(y[!is.na(y)], c(0.25,
0.75))
    x <- qnorm(c(0.25, 0.75))
    if (datax) {
        slope <- diff(x)/diff(y)
        int <- x[1] - slope * y[1]
    }
    else {
        slope <- diff(y)/diff(x)
        int <- y[1] - slope * x[1]
    }
    abline(int, slope, ...)
}
```

Gráficos cuantil - normal para datos con diferentes distribuciones




```
> par(mfrow=c(1,3))
> set.seed(14)

>  y1 <- rnorm(100)
>      qqnorm(y1,main="Q-Q plot, datos
Normales", xlab="Cuantiles Normales",
ylab="Cuantiles
muestrales,datos=rnorm(100)"
+ ,cex.lab=1.3,cex.axis=1.5)
>   qqline(y1)

>   y2 <- runif(100)
```



```
>   qqnorm(y2,main="Q-Q plot, datos
uniformes", xlab="Cuantiles Normales",
ylab="Cuantiles
muestrales,datos=runif(100)"
+ ,cex.lab=1.3,cex.axis=1.5)
>   qqline(y2)

>   y3 <- rt(100,2)
>   qqnorm(y3,main="Q-Q plot, datos t2"
, xlab="Cuantiles Normales",
ylab="Cuantiles
muestrales,datos=rt(100,2)"
+ ,cex.lab=1.3,cex.axis=1.5)
>   qqline(y3)
```

Comparación de datos "normales" con la distribución t2

Tenemos que realizar un gráfico similar al que se obtiene con qqnorm pero que la comparación sea con los percentiles de una t2, además la función qqline no sirve.¿Por qué?

Definimos una función similar a qqline:

```
qqt2linea <-
function (y, ...)
{
```

```
    y <- quantile(y[!is.na(y)], c(0.25,
0.75))
    x <- qt(c(0.25, 0.75),2)
        slope <- diff(y)/diff(x)
        int <- y[1] - slope * x[1]
        abline(int, slope, ...)
}
```

Generamos datos con distribución Normal:

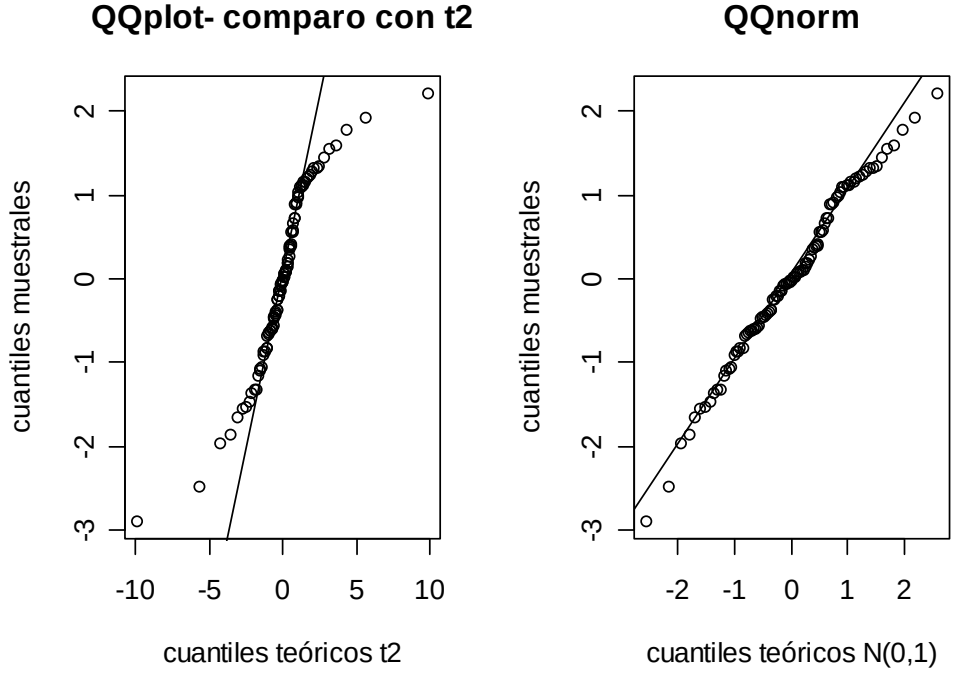
```
set.seed(20) ; x <- rnorm(100)
par(mfrow=c(1,2))
 plot(qt(ppoints(x),2),sort(x),ylab=
 "cuantiles
   muestrales", xlab="cuantiles teóricos
   t2", main="QQplot- comparo con t2")
qqt2linea(x)

qqnorm(x, main="QQnorm", ylab=
"cuantiles
   muestrales",xlab="cuantiles teóricos
N(0,1)")
qqline(x)
```

La función **ppoints** genera *n* “probabilidades” mediante la siguiente fórmula

$$\frac{i-a}{n+1-2a}, \quad i=1,\cdots,n, \qquad 0 \le a \le 1$$

Por defecto a=0.5.

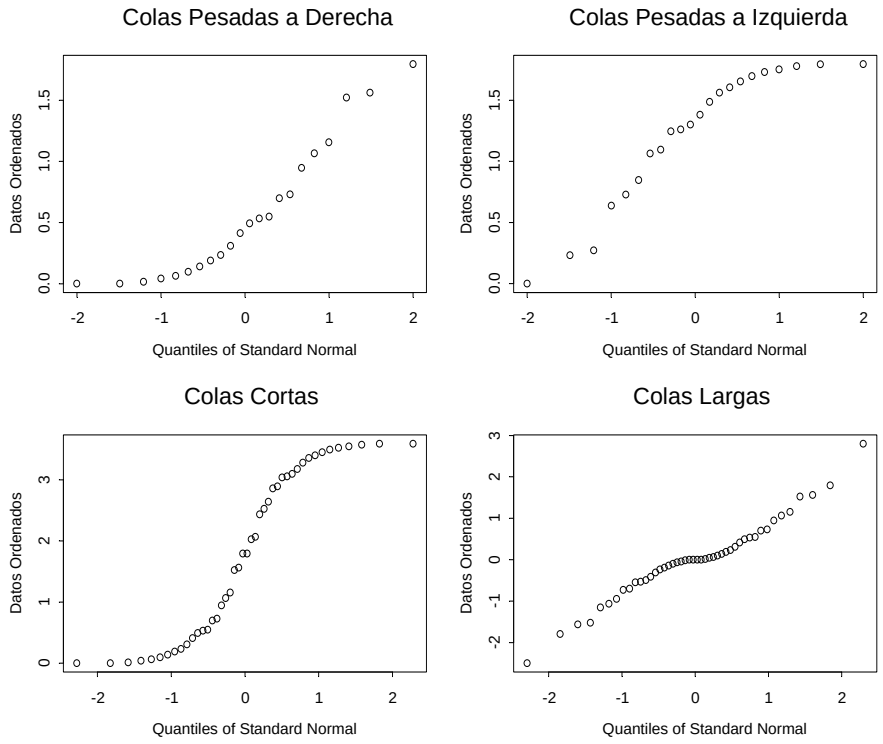
La función **qt** devuelve los cuantiles de la distribución t, correspondientes a los valores generados por **ppoints**.

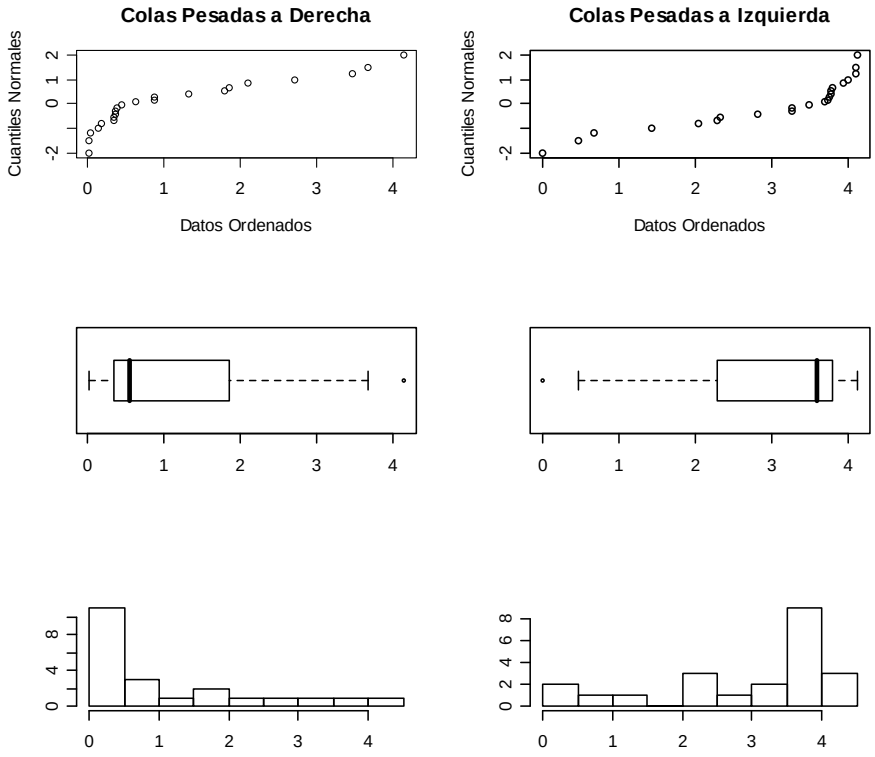


Vemos que los datos generados por la función **rnorm** presentan colas más cortas que lo esperable para una distribución t con 2 grados de libertad.

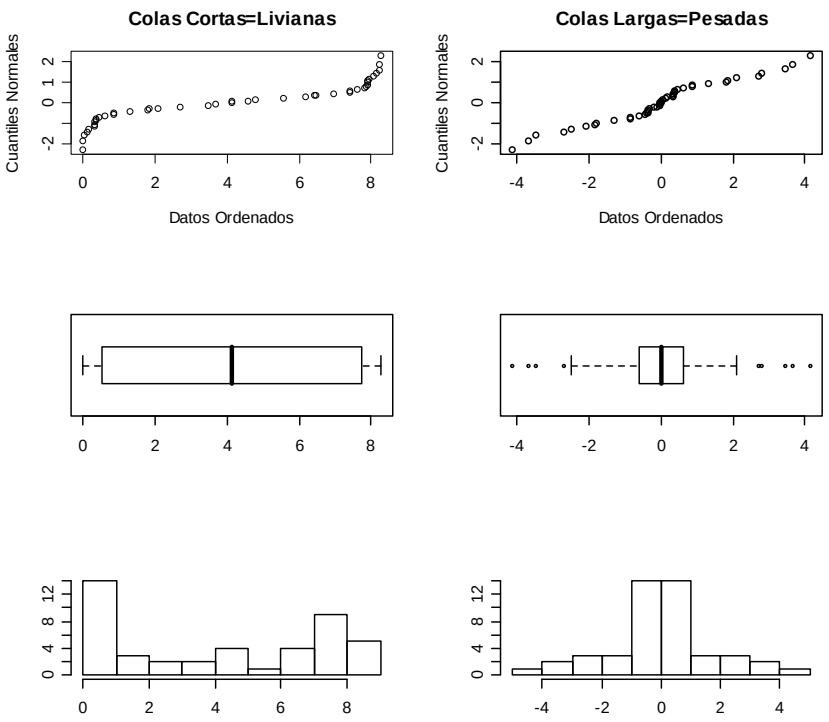
Comparaciones con la distribución Normal: tipos de alejamientos

Principales alejamientos respecto de la distribución Normal
que pueden visualizarse en un gráfico cuantil- cuantil






```
x11()
par(mfcol=c(3,2))
qqnorm(ejemploQQ$der,ylab="Datos
Ordenados",xlab="Cuantiles Normales",
main="Colas Pesadas a Derecha",datax=TRUE)
boxplot(ejemploQQ$der,horizontal =T )
hist(ejemploQQ$der,  xlab="",ylab="",main="")

qqnorm(ejemploQQ$izq,,ylab="Datos
Ordenados",xlab="Cuantiles Normales",
main="Colas Pesadas a Izquierda",datax=TRUE)
boxplot(ejemploQQ$izq,horizontal =T )
hist(ejemploQQ$izq,xlab="",ylab="",main="")
```




```
x11()
par(mfcol=c(3,2))
qqnorm(cort,ylab="Datos Ordenados",xlab="Cuantiles
Normales",
main="Colas Cortas=Livianas",datax=TRUE)
boxplot(cort,horizontal =T  )
hist(cort,xlab="",ylab="",main="")

qqnorm(largo, ylab="Datos Ordenados",xlab="Cuantiles
Normales",
main="Colas Largas=Pesadas", datax=TRUE)
boxplot(largo, horizontal =T )
hist(largo,xlab="",ylab="",main="")
```


Funciones de Distribución

Permiten calcular probabilidades (incluyendo acumuladas), la evaluación de funciones de densidad y probabilidad puntual y la generación de valores pseudo-aleatorios siguiendo diferentes funciones de distribución habituales (tanto discretas como continuas). La tabla siguiente muestra los nombres en R y S-plus de varias funciones junto con argumentos adicionales.

Distribución	**nombre R**	**Argumentos adicionales**[2]	**Argumentos por defecto**
beta	`beta`	`shape1 (α), shape2 (β)`	
binomial	`binom`	`size (n), prob (p)`	
Chi-square	`chisq`	`df (degrees of freedom r)`	
uniforme	`unif`	`min (a), max (b)`	`min = 0, max = 1`
exponential	`exp`	`rate (λ= 1/θ)`	`rate = 1`
F distribution	`f`	`df1 (r1), df2 (r2)`	
gamma	`gamma`	`shape (α), rate (λ)`	`rate = 1`
hypergeometric	`hyper`	`m = N1, n = N2, k = n (sample size)`	
normal	`norm`	`mean (μ), sd (σ)`	`mean = 0, sd = 1`
Poisson	`pois`	`lambda (λ)`	
t distribution	`t`	`df (degrees of freedom r)`	
Weibull	`weibull`	`shape (α), scale (β)`	`scale = 1`

A cada nombre de función dado (tabla anterior) se le agrega un prefijo

'**d**' para obtener la función de densidad o de probabilidad puntual,
'**p**' para la función de distribución acumulada FDA,
'**q**' para la función cuantil o percentil y
'**r**' para generar variables pseudo-aleatorias (random). La sintaxis es la siguiente:

```
> drname(x, ...) # evalúa la fdp o la fpp en x
> prname(q, ...) # evalúa la FDA en q
> qrname(p, ...) # evalúa el p-ésimo cuantil
> rrname(n, ...) # simula n observaciones
```

donde ***rname*** (wildcard) indica el nombre de cualquiera de las distribuciones, **x** y **q** son vectores que toman valores en el soporte de la distribución, **p** es un vector de probabilidades y **n** es un valor entero. Los siguientes son ejemplos:


```
> x <- rnorm(100) # asigna a x 100 valores
                  # generados de una normal estándar
> w <- rexp(100,rate=.1)
# asigna a x 100 valores
# generados de una Exp(θ = 10)
> dbinom(3,size=10,prob=.25)
 # P(X = 3) para X ~ Bin(n=10, p=.25)
> pbinom(3,size=10,prob=.25)
 # P(X ≤ 3) en la distr. anterior
> pnorm(12,mean=10,sd=2)
  # P(X ≤ 12) para  X~N(mu = 10, sigma = 2)
> qnorm(.75,mean=10,sd=2)
 # cuartil superior de una
 # N(mu = 10,sigma = 2)
> qchisq(.10,df=8)
# percentil del 10% de χ 2(8)
> qt(.95,df=20) #  percentil del 95% de t(20)
```

Medidas resumen con R

Si

```
> x <- seq(1,120,0.5)
> length(x)
[1] 239
```

La función **summary** da

```
> summary(x)
   Min. 1st Qu. Median Mean 3rd Qu.
Max.
      1   30.75   60.5 60.5   90.25
120
```

También están las funciones **max, min, range:**

```
> max(x)
[1] 120
> min(x)
[1] 1
> range(x)
[1]   1 120
```

Definimos, en general el p - ésimo cuantil como el número x_P, tal que

#(Datos $< x_P$) / n $\leq p$ y #(Datos > x_P) / n $\leq 1 - p$

Nuevamente, existe un intervalo de valores posibles para el
p - ésimo cuantil.

La función

```
  quantile(x, probs=seq(0,1,.25),
na.rm=F)
```

da, por defecto, el mínimo, el cuartil inferior, la mediana, el cuartil superior y el máximo del vector **x** un resumen de 5 números (“five number summary”).

```
> quantile(x)
  0%   25%  50%   75% 100%
   1 30.75 60.5 90.25  120
```

Permite calcular cuantiles en proporciones (con valores entre 0 y 1 inclusives) especificadas:

```
> quantile(x,
c(0.05,0.10,0.90,0.95))
     5%  10%   90%    95%
   6.95 12.9 108.1 114.05
```

Los valores faltantes no son permitidos, si se especifica **na.rm=TRUE** son eliminados del cálculo.

```
> y<- c(x[1:10], NA ,x[11:239])

> quantile(y, c(0.05,0.10,0.90,0.95))
Error en quantile.default(y, c(0.05,
0.1, 0.9, 0.95)) : missing values and
NaN's not allowed if 'na.rm' is FALSE



> quantile(y, c(0.05,0.10,0.90,0.95),

na.rm=TRUE)
    5%   10%   90%    95%
 6.95 12.9 108.1 114.05
```

Del **help del S-plus**

> The algorithm linearly interpolates between order
> statistics of x, assuming that the ith order statistic
> is the (i-1)/(length(x)-1) quantile.

O sea que el dato en la posición i (el estadístico de orden i) es considerado un p-cuantil con

$$p(i)= (i -1) / (n-1)$$

Si nos interesa calcular la posición del cuartil resulta

$$1/4 = (i -1) / (n-1)$$

luego la posición para el cuartil resulta

$$i\ (¼) = ¼\ (n-1) + 1 = (n+3)/4$$

En general teníamos $p(i) = (i -1) / (n-1)$

y la posición resulta

$$i(p) = p\ (n-1) + 1$$

Si el valor obtenido no es entero la función interpola linealmente. Si n = 100, i = 24.25 por lo tanto

$$\boldsymbol{x_{1/4} = (1- 0.25)\ x_{(24)} + 0.25\ x_{(25)}}$$

En general, si

$$x_{(1)} , x_{(2)} , . . . , x_{(n)}$$

es la muestra ordenada, el cuantil p-ésimo es

$$\text{quantile}(x, p) = \{1-(i(p)-[i(p)])\}\, x_{(1+[i(p)])} + \{ i(p)-[i(p)]\}\, x_{(2+[i(p)])}$$

En R se pueden elegir entre varios criterios de asignación de p-cuantiles a los datos, por defecto toma el tipo 7 igual que el S-plus

Type 4
***p(i) =i / n*.**
Type 5
***p(i) = (i - 0.5) / n*. This is popular amongst hydrologists.**
Type 6
***p(i) = i / (n + 1)*. This is used by Minitab and by SPSS.**
Type 7
***p(i) = (i - 1) / (n - 1)*. This is used by S.**
Type 8
***p(i) = (i - 1/3) / (n + 1/3)*. The resulting quantile estimates are approximately median-unbiased regardless of the distribution of x.**
Type 9
***p(i) = (i - 3/8) / (n + 1/4)*. The resulting quantile estimates are approximately unbiased for the expected order statistics**

Función de Distribución Empírica

Supongamos que $(x_1, x_2,..., x_n)$ es un lote de números.

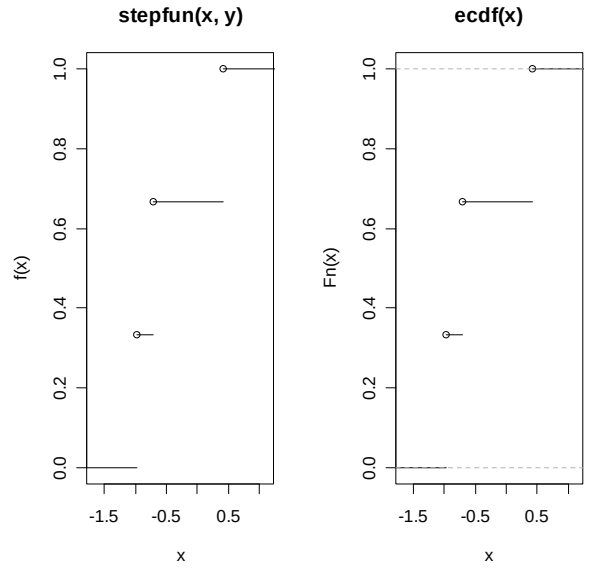
$$F_n(x) = \frac{1}{n}(\#\, x_i \le x) \qquad (1)$$

$F_n(x)$ da la proporción de los datos que son menores o iguales que x, es decir, la *frecuencia relativa acumulada*. Es una función escalera que tiene un escalón de altura 1/n en cada dato. Es continua a derecha.

Gráfico de la función de distribución empírica

En R, las funciones **`ecdf`** y **`plot.ecdf`** permiten graficar funciones de distribuciones empíricas acumuladas. Son implementaciones especiales de la **`stepfun`**.

Veamos un ejemplo

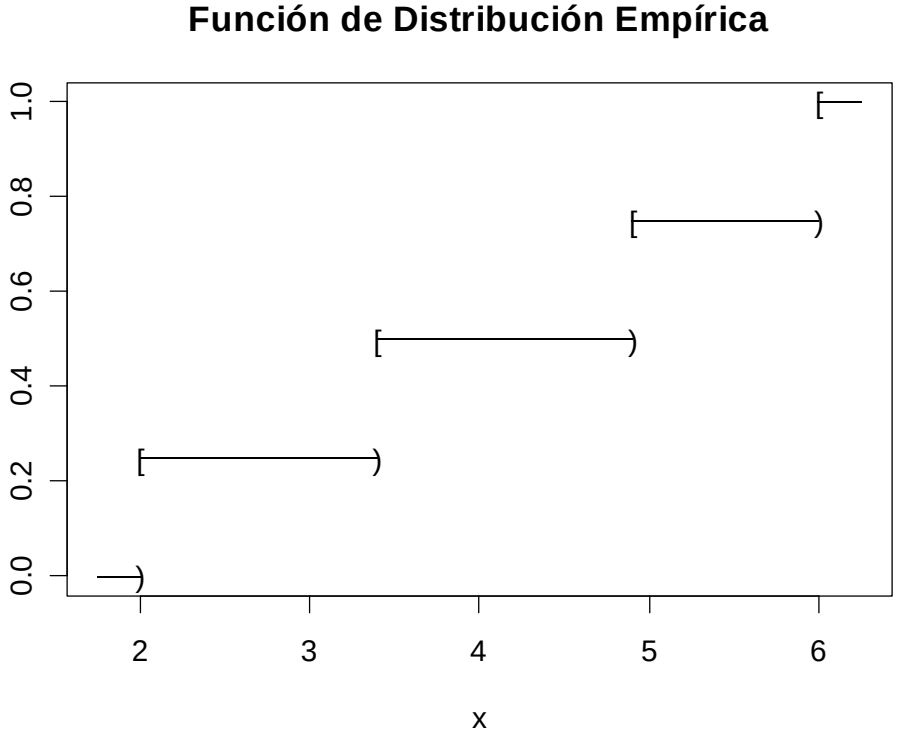



```
> n <- 3
> x <- sort(rnorm(n) )
>  x
[1] -0.9713815 -0.7106415  0.4190012
> y <- 0:length(x)/length(x)# desde
cero


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

 > plot.ecdf(ecdf(x)) # x no
necesariamente ordenado
o
 > plot(ecdf(x))
```



```
 distr.empirica3<- function(x){
 # tarda mucho
#cambiar los for por operaciones
 vectoriales

 n <- length(x)
 m <- 500*n
 grilla<-(max(x)-min(x))*seq(1:m)/m +
 min(x)
 grillaini<-  min(x) - seq(1:500)/m
 grillafin<-  max(x) + seq(1:500)/m
      acumulini<- rep(0,500)
      acumulfin<- rep(1,500)
 acumul<-vector(mode="numeric",
 length=m)
 saltos<-vector(mode="numeric",
 length=n)
 sx <- sort(x)
 for (i in 1:m )
  for (j in 1:n){
   saltos[j]<-j/n
   if (grilla[i] > sx [j])
    acumul[i] <- j/n
            }

 plot(c(grillaini,grilla,grillafin),c(a
```

```
  cumulini,acumul,acumulfin),pch='.',
  ylim=c(0,1),xlab="x", ylab="")
     points(sx,saltos,pch='[ ')
     points(sx,c(0,saltos[-n]), pch=')')
   title("Función de Distribución
  Empírica")
  }

distr.empirica3(c(2, 3.4, 4.9, 6))
```



La figura siguiente contiene el gráfico de una función de distribución empírica (acumulada) FDA hipotética, con 3 datos,