

Introducción a R

Modelos no paramétricos y de regresión

Enrique Reyes

12 de febrero de 2018

Probabilidad

R de forma predeterminada (en el paquete STATS), ya tiene cargadas 16 distribuciones, 5 discretas (binomial, poisson, geométrica, hipergemétrica, binomial negativa) y 11 continuas (uniforme, beta, cauchy, chi cuadrada, exponencial, gamma, F, log-normal, normal,]T-student, weibull), estas distribuciones son las más conocidas. Con el paquete *actuar* se añaden otras 21 distribuciones más, con las que se pueden hacer mezclas y transformaciones conforme sea el caso de estudio.

Con las distribuciones de R se pueden generar:

- Muestras *pseudoaleatorias* (r debe anteceder a la identificación de la distribución)
- Cuantiles (q debe anteceder a la identificación de la distribución)
- Función de densidad (d debe anteceder a la identificación de la distribución)
- Función de distribución (p debe anteceder a la identificación de la distribución)

Muestra aleatoria

```
#Ejemplo con distribución normal estándar
#Necesitamos una muestra aleatoria de tamaño 1000
#semilla
set.seed(1978)
#creamos la muestra
sim<-rnorm(1000)
#observamos los primeros 20 elementos de la muestra
sim[1:20]
```

```
## [1] -1.02331764  0.39436819 -0.56102909  2.45981477  0.86510097
## [6] -1.05075361 -1.42896137  0.21684883 -0.34349509 -1.69987825
## [11]  0.59720539 -0.01361331  0.46775929 -1.70486307 -1.55470442
## [16] -0.60600234  0.54179878  1.83888064  0.39619325  0.61401606
```

```
#calculamos media, varianza y desviación estándar de la muestra
mean(sim); var(sim); sd(sim)
```

```
## [1] 0.05557488
```

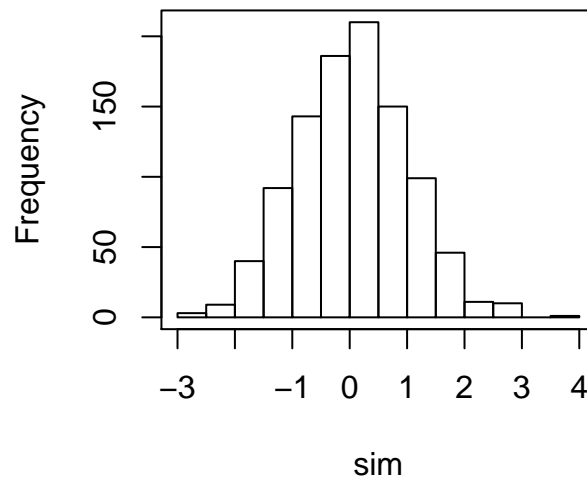
```
## [1] 0.9358718
```

```
## [1] 0.9674047
```

Podemos ver que la esperanza muestral $\bar{x} = 0.0556$ es cercana a la esperanza teórica $\mu = 0$, lo mismo pasa con la desviación estándar muestral $s = 0.9674$ es similar a la desviación teórica $\sigma = 1$

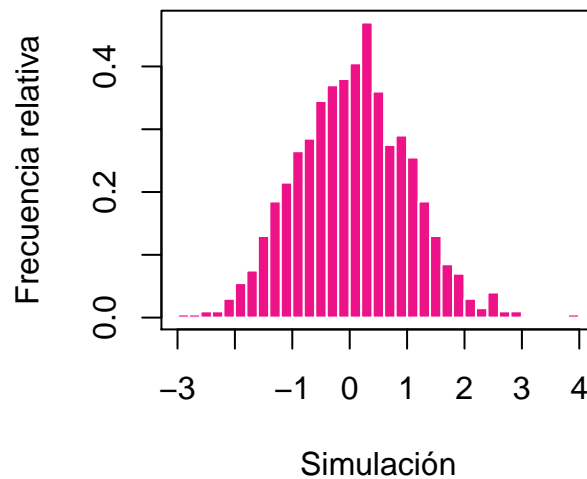
```
#pintamos el histograma que R crea por default
hist(sim)
box()
```

Histogram of sim



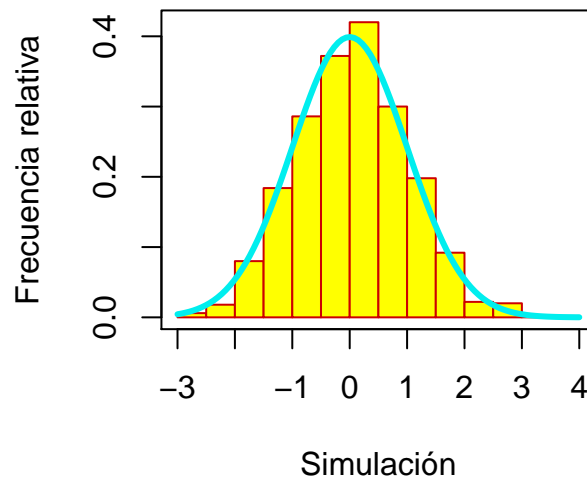
```
#agregamos algunos parámetros a la función, para obtener un mejor resultado
hist(sim,col="deeppink2",breaks=30,border="white",freq=0,xlab="Simulación",
      ylab = "Frecuencia relativa", main="Histograma")
box()
```

Histograma



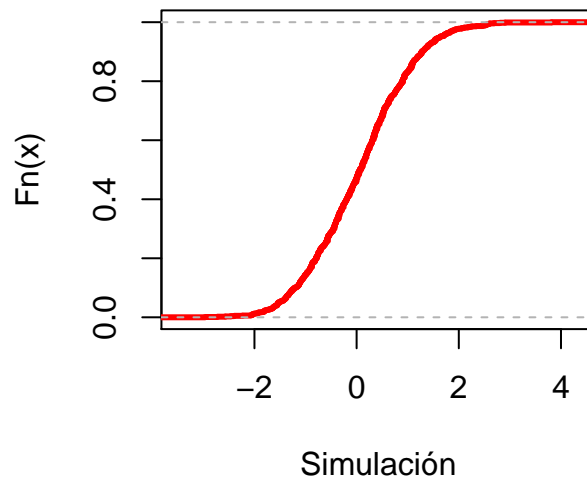
```
#ponemos en un mismo plano la densidad empírica y el histograma
hist(sim,col="yellow",border="red3",freq=0, xlab="Simulación",
      main="Histograma de la simulación",ylab = "Frecuencia relativa")
curve(dnorm(x),-3,4,add=T,col="cyan2",lwd=3)
box()
```

Histograma de la simulación



```
#graficamos la función de distribución empírica  
fde=ecdf(sim)  
plot(fde, col="red",lwd=3,xlab="Simulación",  
     main="Función de distribución empírica")
```

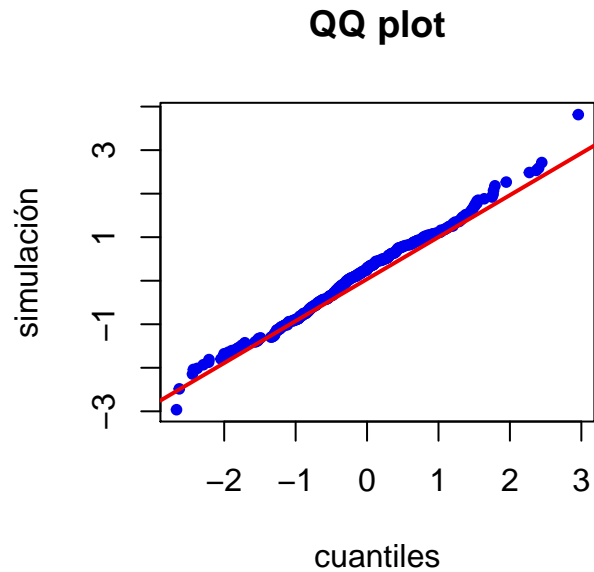
Función de distribución empírica



```
#gráfica cuantil cuantil  
qqplot(qnorm(sim),sim,col="blue2",pch=20,xlab="cuantiles",  
       ylab="simulación",main="QQ plot")
```

```
## Warning in qnorm(sim): NaNs produced
```

```
qqline(sim,col="red2",lwd=2)
```



Cuantiles

```
#cuantiles  
probabilidades=c(0.8,0.9,0.95,0.975,0.99)  
qnorm(probabilidades)
```

```
## [1] 0.8416212 1.2815516 1.6448536 1.9599640 2.3263479
```

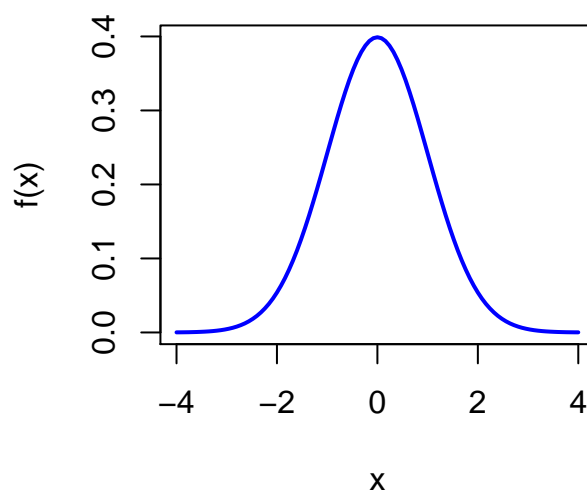
Densidad

```
#función de densidad  
valores=c(-2.3,-2,-1.6,-1.3,-1.0,1,1.3,1.6,2,2.3)  
dnorm(valores)
```

```
## [1] 0.02832704 0.05399097 0.11092083 0.17136859 0.24197072 0.24197072  
## [7] 0.17136859 0.11092083 0.05399097 0.02832704
```

```
#densidad teórica  
curve(dnorm(x),xlim=c(-4,4),col="blue",lwd=2, xlab="x",ylab="f(x)",  
      main="Función de Densidad N(0,1)")
```

Función de Densidad N(0,1)



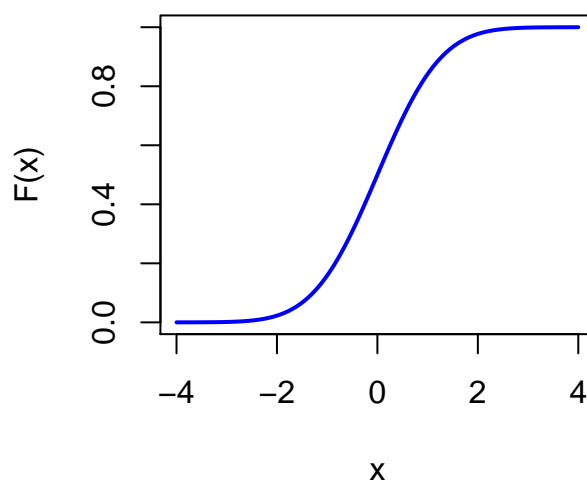
Distribución

```
#función de distribución  
valores=c(-2,-1.6,-1.0,1,1.6,2)  
pnorm(valores)
```

```
## [1] 0.02275013 0.05479929 0.15865525 0.84134475 0.94520071 0.97724987
```

```
#distribución teórica  
curve(pnorm(x),xlim=c(-4,4),col="blue",lwd=2, xlab="x",ylab="F(x)",  
      main="Función de Distribución N(0,1)")
```

Función de Distribución N(0,1)



Programación

IF

```
#Se ejecuta el comando si la condición es TRUE
if(3 > 2) print(' :D ')
```

```
## [1] " :D "
```

```
x=1; Y=NULL
if(x > 3){
  y <- 10
}else{
  y<-0};
y
```

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

```
ifelse(3 < 2,print(':D'),print(':O'))
```

```
## [1] ":O"
```

```
## [1] ":O"
```

FOR

```
x<-c("a","b","c","d")
for(i in 1:6) {
  print(x[i]) # Imprime cada uno de los elementos de x.
} # Los dos últimos, al no existir en x aparecen como "NA"
```

```
## [1] "a"
## [1] "b"
## [1] "c"
## [1] "d"
## [1] NA
## [1] NA
```

WHILE

```
f <- 5 # Valor inicial
n <- 0
while(f > 0.001) {
  n <- n + 1
  f <- f / n
  print(f)
}
```

```
## [1] 5
## [1] 2.5
## [1] 0.8333333
## [1] 0.2083333
## [1] 0.04166667
## [1] 0.006944444
## [1] 0.0009920635
```

REPEAT

```
v <- c("Hola","mundo")
cnt <- 2
repeat {
  print(v)
  cnt <- cnt+1
  if(cnt > 5) {
    break
  }
}
```

```
## [1] "Hola" "mundo"
## [1] "Hola" "mundo"
## [1] "Hola" "mundo"
## [1] "Hola" "mundo"
```

FUNCIÓN

```
#Sucesion de fibonacci
fibo=function(n){
  Res=numeric(n)
  if(n==1){
    Res[1]=1}
  if(n==2){
    Res[1:2]=c(1,1)}
  if(n>2){
    Res[1:2]=c(1,1)
    for(i in 3:n){
      Res[i]=Res[i-1]+Res[i-2]
    }
  }
  Res
}
```

####Ejemplo

```
fibo(10)
```

```
## [1] 1 1 2 3 5 8 13 21 34 55
```