

### Probabilidades discretas

#### Distribución uniforme

dunif(k,a,b)  
punif(k,a,b)  
runif(n,a,b)

variable que puede tomar n valores distintos con la misma probabilidad.  
Probabilidad de que x sea k en un intervalo de a a b  
runif: n muestras distintas

#### Distribución binomial

dbinom(x, size, prob)  
pbinom(q, size, prob)  
rbinom(n, size, prob)

cuenta el número de éxitos en n pruebas independientes. x es el número de éxitos, size el número de pruebas y prob la probabilidad.

#### Distribución geométrica

dgeom(x, prob)  
pgeom(q, prob)  
rgeom(n, prob)

probabilidad de que tenga que realizarse un número k de repeticiones antes de obtener un éxito por primera vez

#### Distribución hipergeométrica

dhyper(x, m, n, k)  
phyper(q, m, n, k)  
rhyper(nn, m, n, k)

Tenemos una cesta con m pelotas blancas y n pelotas negras. Si sacamos k pelotas, probabilidad de que x o q pelotas sean blancas.

#### Distribución de Poisson

dpois(x, lambda)  
ppois(q, lambda)  
rpois(n, lambda)

es una forma límite de la distribución binomial que surge cuando se observa un evento raro después de un número grande de repeticiones.

**Lambda** es la media esperada, y **x o q** es el resultado que queremos consultar.

### Distribuciones de probabilidad continuas

### Distribuciones de probabilidad continuas (cont)

dlnorm(x, meanlog = 0, sdlog = 1)  
plnorm(q, meanlog = 0, sdlog = 1)  
rlnorm(n, meanlog = 0, sdlog = 1)

si una variable x sigue una distribución lognormal entonces la variable ln(x) se distribuye normalmente. Es útil para cuando los valores de x se encuentra muy separados.

A meanlog también se le llama parámetro de escala y a sdlog forma

#### Distribución beta

adecuada para variables aleatorias continuas que toman valores en el intervalo (0,1)

dbeta(x, shape1, shape2)  
pbeta(q, shape1, shape2)  
rbeta(n, shape1, shape2)

x o q es la proporción que queremos calcular

#### Distribución gamma

dgamma(x, shape, rate = 1)  
pgamma(q, shape, rate = 1)  
rgamma(n, shape, rate = 1)

Mide el tiempo transcurrido hasta obtener n ocurrencias de un evento generado por un proceso de Poisson de media lambda

#### Distribución exponencial

dexp(x, rate = 1)  
pexp(q, rate = 1)  
rexp(n, rate = 1)

Es un caso particular de la distribución gamma. describe procesos en los que interesa saber el tiempo hasta que ocurre determinado evento-

Todas las que empiezan por p tienen lower.tail = TRUE

### Distribución que mejor se ajusta a unos datos

### Comparación de modelos/ajustes con AIC y BIC

AIC (Criterio de información de Akaike)

AIC =  $-2\log(\text{likelihood}) + 2 \times \text{no parámetros}$

BIC (Bayesian information criterion)

BIC =  $-2\log(\text{likelihood}) + \log(\text{no observaciones}) \times \text{no parámetros}$

### AIC y BIC

require(fitdistrplus)

dist\_lnorm <- fitdistr(datos\$price, distr = "lnorm")

dist\_weibull <- fitdistr(datos\$price, distr = "weibull")

comparacion <- gofstat(f = list(dist\_lnorm, dist\_weibull))

Además de los estadísticos AIC y BIC, la función gofstat() devuelve 3 estadísticos de bondad de ajuste, (Kolmogorov-Smirnov, Cramer-von Mises y Anderson-Darling).

Estos estadísticos, también conocidos como goodness-of-fit, contrastan la similitud entre la distribución empírica obtenida y la distribución teórica con los parámetros estimados. Ninguno de estos 3 últimos tiene en consideración el número de parámetros, por lo que no deben emplearse para comparar distribuciones con distintos grados de libertad.

```
gr = denscomp(
  list(dist_lnorm, dist_weibull), legendtext =
  c("lognormal", "Weibull"), xlab = "precio",
  fitcol = c("red", "blue"),
  fitlty = 1, xlegend = "topright", plotstyle = "ggplot",
  addlegend = FALSE)
```

Veamos gráficamente cuál de las dos distribuciones se ajustan mejor a nuestros datos

### Distribución normal

```
dnorm(x, mean = 0, sd = 1)
```

```
pnorm(q, mean = 0, sd = 1)
```

```
mnorm(n, mean = 0, sd = 1)
```

Media es 0 y sd 1 por defecto

### Distribución log normal

```
descdist(data = datos$price)
```

Análisis exploratorio de la base de datos

```
distribucion=fitdist(datos$price, distr = "lnorm")
```

```
summary(distribucion)
```

Ajuste a una distribución lognormal

```
x=rlnorm(x, meanlog, sdlog)
```

```
hist(x,freq=FALSE,col="lightsalmon",main="Histograma",sub="Datos simulados de una N(meanlog, sdlog)")
```

simular una muestra procedente de dicha distribución

C

By **julenx**

[cheatography.com/julenx/](https://cheatography.com/julenx/)

Published 29th November, 2022.

Last updated 22nd November, 2022.

Page 1 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>