

## PRÁCTICA 1: Distribuciones de probabilidad, números aleatorios, simulación de variables y estimación puntual.

### DESARROLLO DE LA PRÁCTICA CON RSTUDIO

Para la creación de números aleatorios, lo más práctico es realizar un sencillo programa de instrucciones, que invoque a las diferentes distribuciones. Para una distribución dada, se puede solicitar **d** (density functions) función de densidad o distribución de probabilidades; **p** (probability distribution functions) la función de distribución; **q** (quantile functions) los cuantiles de la distribución o **r** (random number generation) la generación de valores aleatorios. Letra que se debe anteponer a la distribución en particular. Por ejemplo para la distribución normal “norm” podría ponerse *dnorm*, *pnorm*, *qnorm* o *rnorm*, según el caso.

#### Distribución discreta

A) Supongamos que queremos hallar la probabilidad de que una variable X de Poisson con  $\lambda=3.5$  tome el valor 0

```
dpois(x=0,lambda=3.5)
```

*# o más sencillo:*

```
dpois(0, 3.5)
```

¿Cuál es la probabilidad de que  $X=5$ ? ¿Y de que X sea par y menor o igual a 20?

Para hallar la probabilidad acumulada de que una variable X de Poisson con  $\lambda=3.5$  tome valor como máximo 6

```
ppois (6, 3.5)
```

¿Cuál es la probabilidad de que X valga más de 2? ¿Y de que X sea distinto de 3? ¿Y que X valga entre 3 y 7, ambos incluidos? ¿En qué valor de X se tiene una probabilidad acumulada de 0.999?

B) Considerar ahora una variable X binomial con  $n=10$  y  $p=0.65$ . Para hallar la probabilidad acumulada de que esta variable X tome valor como máximo 8

```
pbinom (8, 10, 0.65)
```

¿De qué otra forma se puede hallar esta probabilidad utilizando la función de densidad de probabilidad, es decir, con *dbinom*?

#### Distribución continua

Supongamos que queremos hallar la probabilidad de que una variable X normal con  $\mu=15$  y  $\sigma=5$  valga menos de 13

```
pnorm(13,mean=15,sd=5)
```

*# o más sencillo:*

```
pnorm (13, 15,5)
```

¿Cuál es la probabilidad de que X valga más de 16? ¿Y de que X esté entre 10 y 20?

¿Para qué valor de X se tiene una probabilidad acumulada de 0.2, es decir,  $\Pr(X < x) = 0.2$ ? ¿Y para qué valor se tiene que la probabilidad acumulada por encima del mismo sea 0.2, es decir,  $\Pr(X > x) = 0.2$ ?

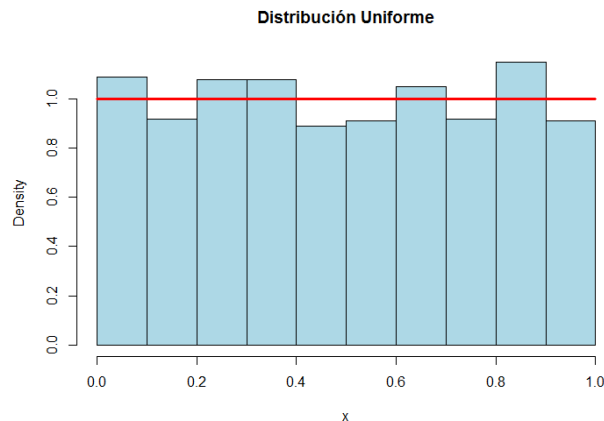
¿Cómo se hallaría el valor anterior considerando la probabilidad que hay por debajo del mismo, es decir,  $\Pr(X < x) = 0.8$ ?

Hallar  $\Pr(|X - \mu| < \sigma)$ ,  $\Pr(|X - \mu| < 2 * \sigma)$  y  $\Pr(|X - \mu| < 3 * \sigma)$

## NÚMEROS ALEATORIOS

Para generar 1000 números de una distribución uniforme  $U(0,1)$  y posterior histograma (ver NOTA al final del documento), tenemos las siguientes líneas de código:

```
n <- 1000
x <- runif(n)
hist(x, freq=FALSE, col='light blue', main='Distribución Uniforme')
curve(dunif(x), add=T, col='red', lty=1, lwd=3)
```



Por defecto, los números aleatorios se generan según el método de Mersenne-Twister. Si deseamos generar el mismo conjunto de datos aleatorios, debemos indicar la semilla (seed) elegida, esto se puede conseguir con el siguiente comando

```
set.seed(149653)
```

Si los habitantes de un municipio son 26712, y deseamos realizar una m.a.s. de tamaño 20 con reemplazamiento, se puede ejecutar los siguientes comandos

```
set.seed(149653)
n <- 20
x1 <- runif(n)
x2 <- round(26712*x1, digits=0)
x3 <- sort(x2)
print(x3)
#save(x3, file="uniforme.Rdata")
```

Para la realización de un m.a.s. también se puede aplicar el comando

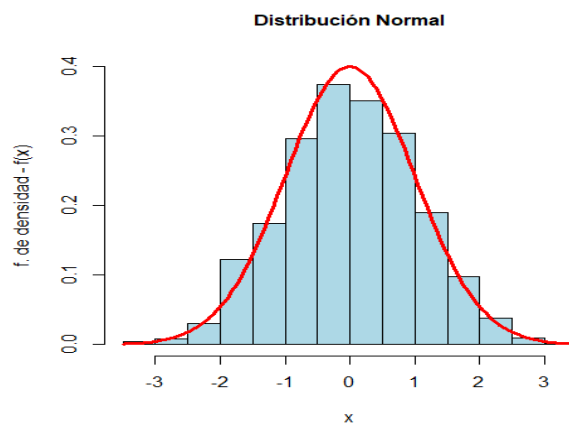
```
y <- sample(1:26712, size=20, replace = FALSE, prob = NULL)
print(sort(y))
```

siempre que se disponga de una base de datos con todas las unidades de la población.

## SIMULAR DISTRIBUCIONES

Con las siguientes líneas de código se generan 1000 números aleatorios de una normal  $N(0,1)$  y se realiza una gráfica

```
set.seed(149653)
n <- 1000
x <- rnorm(n)
hist(x, freq=FALSE, ylim=c(0,0.4), col='light blue', main='Distribución Normal')
curve(dnorm(x), add=T, col='red', lty=1, lwd=3)
```

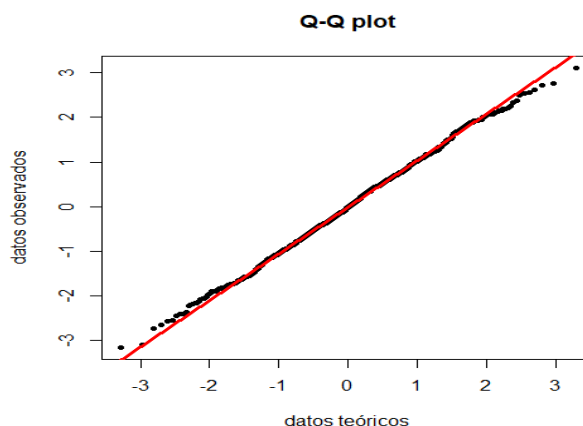


Para evaluar si los datos generados aleatoriamente se pueden considerar como procedentes de una distribución normal, podemos realizar los gráficos Q-Q y P-P.

Los siguientes comandos

```
qqplot(qnorm(ppoints(1000)), x, pch=20, main="Q-Q plot", ylab="datos observados",
xlab="datos teóricos")
qqline(x, distribution = qnorm, prob = c(0.25, 0.75), col = 2, lwd=3, qtype=7)
```

nos proporcionan el siguiente gráfico cuantil-cuantil (Q-Q plot)

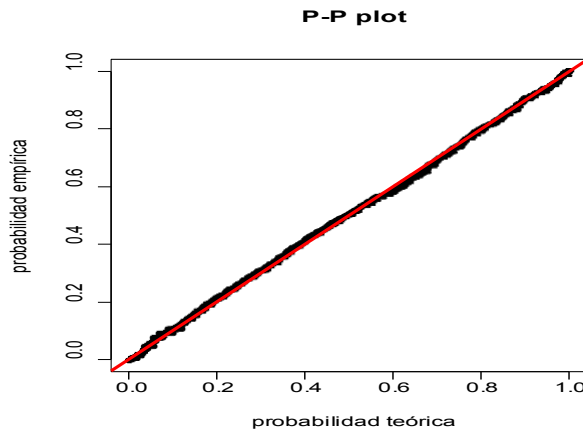


El gráfico Q-Q plot representa los cuantiles observados  $x_{(i)}$  frente a los cuantiles teóricos  $F^{-1}(p_i)$ , donde  $p_i = \frac{i-0.5}{n}$  son las probabilidades acumuladas (generadas con ppoints) para  $i = 1, \dots, n$ .

Los comandos

```
z3<-pnorm(sort(x))  
plot (ppoints(1000), z3, type = 'p', ylim = c(0, 1), pch=20, xlab = 'probabilidad teórica',  
ylab = 'probabilidad empírica', main = 'P-P plot')  
abline(0,1, col = 2, lwd=3)
```

nos proporciona la gráfica probabilidad-probabilidad (P-P plot) siguiente



El gráfico P-P plot representa las probabilidades acumuladas  $F(x_{(i)})$  (empíricas) frente a las probabilidades acumuladas  $p_i = \frac{i-0.5}{n}$  (teóricas) para  $i = 1, \dots, n$ .

El procedimiento para generar valores de diversas variables, se puede realizar con los mismos comandos ya expuestos, pero tomando la distribución correspondiente.

## COMPROBAR PROPIEDADES

Supongamos que  $X_i \sim N(\mu = 10, \sigma^2 = 4)$ ,  $i=1, \dots, n$ .

Comprobar que  $E[X_1 + \dots + X_n] = n\mu$  y  $Var[X_1 + \dots + X_n] = n\sigma^2$

```
set.seed(149653)  
vec<- sum(rnorm(50,10,2))  
simul<-replicate(1000, vec )
```

```
mean(simul) # próximo a 50*10  
var(simul)  # próximo a 50*4
```

Para obtener los estimadores de máxima verosimilitud (MV) de los parámetros de determinadas distribuciones, podemos utilizar la librería **"MASS"** y si queremos obtener un estimador MV en condiciones generales debemos recurrir a la librería **"bbmle"**.

## ESTIMACIÓN PUNTUAL: Estimadores de máxima verosimilitud

En este apartado también optamos por generar valores aleatorios de determinadas distribuciones y solicitar los estimadores de máxima verosimilitud

### Caso1: Distribución Normal

Generamos 100 valores aleatorios de una distribución normal  $N(5, 4)$  y después le indicamos que nos obtenga los estimadores de ML de  $\mu$  y  $\sigma$ . Se consigue con los comandos

```
set.seed(1496)
x<-rnorm(100,5,2)
library(fitdistrplus)
fitdist(x, distr = "norm")
```

sus resultados son

	estimate	Std. Error
mean	4.883205	1.989778
sd	0.1989778	0.1406984

$$\hat{\mu} = \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \hat{\sigma} = \frac{1}{n} \sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}$$

### Caso2: Distribución Exponencial

Generamos 100 valores aleatorios de una distribución exponencial  $\text{Exp}(0.5)$  y después le indicamos que nos obtenga el estimador de ML de  $\beta$ . Se consigue con los comandos

```
set.seed(1496)
x<-rexp(100,0.5)
fitdist(x, distr = "exp")
```

su resultado es

	estimate	Std. Error
rate	0.4800118	0.04800097

$$\hat{\beta} = \frac{1}{x}$$

### Caso3: Distribución Gamma

Generamos 100 valores aleatorios de una distribución normal  $Ga(4, 0.5)$  y después le indicamos que nos obtenga los estimadores de MV de  $\alpha$  y  $\beta$ , respectivamente. Se consigue con los comandos

```
set.seed(1496)
x<-rgamma(100,4,0.5)
fitdist(x, distr = "gamma")
```

sus resultados son

	estimate	Std. Error
shape	4.1279016	0.56176847
rate	0.5578038	0.08072361

### Caso4: Distribución Binomial

Generamos 100 valores aleatorios de una distribución normal  $Bi(12, \theta = 0.3)$  y después le indicamos que nos obtenga el estimador de ML de  $\theta$ . En este caso,

Se consigue con los comandos

```
set.seed(1496)
x<-rbinom(100,12,0.3)
fitdist(x, discrete = T, distr = "binom", fix.arg=list(size=12), start=list(prob=0.2))
```

su resultado es

	estimate	Std. Error
prob	0.2924996	0.01313201

$$\hat{p} = \frac{x}{n}$$

## EJERCICIOS:

1.- Simular 1000 valores de las siguientes distribuciones

$$\begin{aligned}Bi(n, p) &\rightarrow Bi(12, 0.3), \\P(\lambda) &\rightarrow P(3.1), \\BiNe(r, p) &\rightarrow BiNe(4, 0.6), \\Geo(p) &\rightarrow Geo(0.35), \\H(N, M, n) &\rightarrow H(500, 200, 37), \\N(\mu, \sigma^2) &\rightarrow N(7.2, 6), \\Ga(\alpha, \beta) &\rightarrow Ga(11, 3), \\Exp(\beta) &\rightarrow Exp(1.5); \\Be(\alpha, \beta) &\rightarrow Be(2, 7), \\U(a, b) &\rightarrow U(5, 12)\end{aligned}$$

comprobando que los datos simulados se pueden considerar que proceden de las correspondientes distribuciones.

2.- Hallar los estimadores MLE de la variable “peso” en el archivo de datos “HIPER200.RData” y comparar con la media y desv. típica. ¿Se puede considerar que la variable sigue una distribución normal (comprobar con histograma y Q-Q plot)? Filtrar las personas con hipertensión (variable es\_hip=1) y analizar la normalidad para el peso de dichos individuos.

**NOTA:** La función hist() de R representa, por defecto, un gráfico similar a un histograma. Al ejecutar con la opción freq=TRUE, se utilizan las frecuencias absolutas (counts)

```
vec<-rnorm(40)
h<-hist(vec,freq=TRUE) # se representa h$counts
```

pero al especificar freq=FALSE, el gráfico que se obtiene es tal que el área del histograma es 1

```
h<-hist(vec,freq=FALSE) # observar que sum(h$density)>1
```

Si se desea representar un histograma con frecuencias relativas, habría que hacer

```
h <- hist(vec, freq=TRUE, plot=FALSE)
h$counts=h$counts/sum(h$counts)
plot(h)
```