

Trabajo Práctico 4: Introducción al lenguaje R - cont.

6. Probabilidad, Distribuciones y Simulación

6.1 Funciones de Distribución en R

R permite 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 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)	
continuous uniform	unif	min (a), max (b)	min = 0, max = 1
exponential	exp	rate ($\lambda = 1/\theta$)	rate = 1
F distribution	f	df1 (r_1), df2 (r_2)	
gamma	gamma	shape (α), rate (λ)	rate = 1
hypergeometric	hyper	m = N_1 , n = N_2 , 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 por R (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 pésimo percentil de esta distribución
> rrname(n, ...) # simula n observaciones de esta distribución
```

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( $\theta = 10$ )
```

```
> 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 χ²(8)
> qt(.95,df=20) # percentil del 95% de t(20)
```

²

Hogg and Tanis (2006) parameter names are given in parentheses. See the help files for the exact distribution parameterizations.

6.2 Una aplicación de simulación: Integración de Monte Carlo Integration

Supongamos que queremos calcular

$$I = \int_a^b g(x)dx,$$

pero que la primitiva de $g(x)$ no puede darse en una forma cerrada. Las técnicas estándar involucran aproximar la integral por una suma, muchos paquetes de computación hacen esto. Otro procedimiento para hallar I es llamado integración de Monte Carlo.

Supongamos que generamos observaciones pseudo-aleatorias $\mathbf{x} = (x_1, x_2, \dots, x_n)$ de n variables aleatorias independientes $\mathbf{X} = X_1, X_2, \dots, X_n$ con distribución uniforme en el intervalo $[a, b]$ y calculamos

$$\hat{I}(x) = (b-a) \frac{1}{n} \sum_{i=1}^n g(x_i)$$

Por la Ley de los Grandes Números, a medida que n crece (sin ninguna cota),

$$\hat{I}(X) = (b-a) \frac{1}{n} \sum_{i=1}^n g(X_i) \xrightarrow{n \rightarrow \infty} (b-a)E[g(X)]$$

Pero,

$$E[g(X)] = \int_a^b g(x) \frac{1}{b-a} dx = \left(\frac{1}{b-a} \right) I$$

De manera que, $\hat{I}(x)$ puede ser utilizada como una aproximación de I que mejora a medida que n crece.

Este método puede ser modificado para utilizar otras distribuciones (aparte de la Uniforme) definidas sobre el mismo intervalo. Comparado con otros métodos numéricos para aproximar integrales, el método de Monte Carlo no es especialmente eficiente. Sin embargo, el método de Monte Carlo tiene una eficiencia creciente a medida que aumenta la dimensionalidad de la integral (por ej. integrales dobles, triples).

Ejemplo: Consideremos integral definida: $\int_0^2 e^{x^3} dx$

Un estimador de Monte Carlo (utilizando 1 000 000 de observaciones) estará dado por:

```
> 2*mean(exp(runif(1000000,min=0,max=2)^3)) # a=0, b=2, luego b-  
a=2  
[1] 277.0928
```

Otra llamada da un resultado diferente (son distintas estimaciones del valor límite!):

```
> 2*mean(exp(runif(1000000,min=0,max=2)^3))  
[1] 276.6815
```

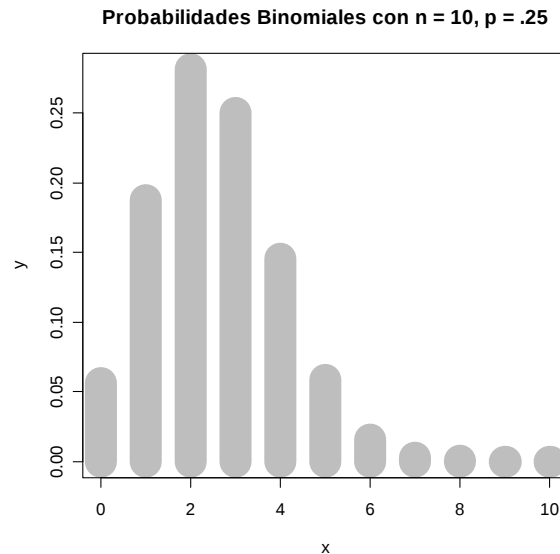
6.3 Graficando Distribuciones

6.3.1 Distribuciones Discretas

Los gráficos de las funciones de probabilidad puntual y sus funciones de distribuciones acumuladas pueden realizarse utilizando la función `plot()` dándole el soporte de la función y las probabilidades en esos puntos. Por defecto, R simplemente producirá un diagrama de dispersión. La presentación se mejora utilizando los argumentos `type` y `lwd` (line width; ancho de línea). Por ejemplo, para graficar la función de probabilidad puntual para la distribución binomial con $n = 10$ y $p = .25$:

```
> x <- 0:10 # generamos el soporte de la distribución  
> y <- dbinom(x, size=10, prob=.25) # evaluamos las probabilidades  
> plot(x, y, type = "h", lwd = 30, main = "Probabilidades  
Binomiales con n = 10, p = .25", col = "gray") # enter
```

(queda feo !?!, porqué?)

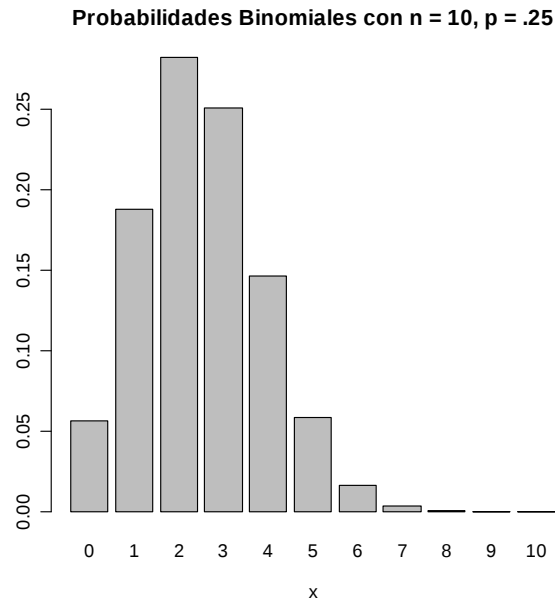


Hemos realizado algunas cosas aquí. Primero hemos creado el vector **x** que contiene los enteros del 0 al 10. Luego calculamos las probabilidades binomiales en cada uno de los puntos de **x** y las guardamos en el vector **y**. Luego especificamos un gráfico de tipo "h" que da el aspecto de histograma con líneas verticales y hemos engordado las líneas con la opción `lwd = 20` (el ancho por defecto es 1, que corresponde al grosor de la línea). Finalmente le dimos un color y un título informativo. Notemos que para ahorrar espacio en el espacio de trabajo, podríamos haber obtenido el mismo gráfico sin crear los vectores **x** e **y**:

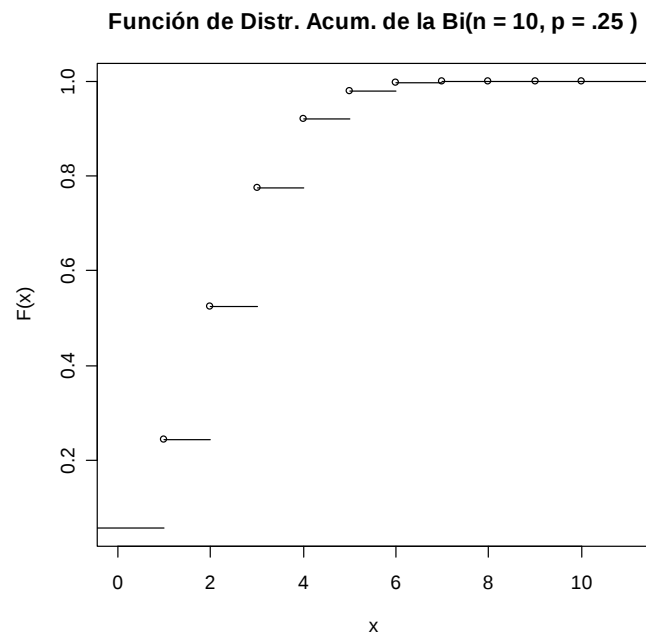
```
> plot(0:10, dbinom(0:10, size=10, prob=.25), xlab="x", ylab="y", type
= "h", lwd = 20, main = "Probabilidades Binomiales con n = 10, p =
.25", col = "gray") # enter
```

Otra forma de obtener un diagrama de barras de una función de probabilidad puntual es mediante la función `barplot()`:

```
> barplot(y, names.arg=as.character(0:10), main = "Probabilidades
Binomiales con n = 10, p = .25", xlab="x")
```



Para crear el gráfico de la función de distribución acumulada de la binomial:



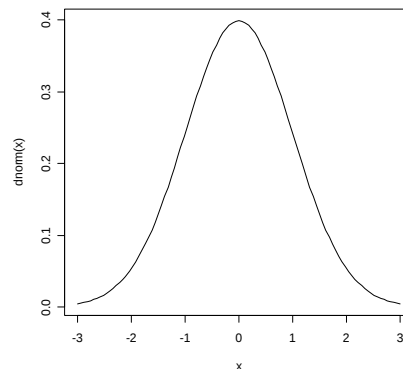
```
> plot(stepfun(1:10, pbinom(c(0,1:10), size=10, prob=.25))
+       , verticals=FALSE, ylab="F(x)",
+       main = "Función de Distr. Acum. de la Bi(n = 10, p = .25 )",)
```

Utilice el **help** para ver que acción realiza la función **stepfun**.

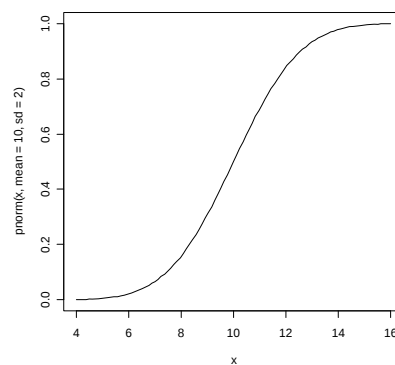
6.3.2 Distribuciones Continuas

Para graficar funciones suaves como una función de densidad de probabilidad o la FDA de una variable aleatoria continua podemos utilizar la función **curve()** que fue introducida en el Capítulo 4. Podemos utilizar los nombres de las funciones de densidad de R como el argumento de expresión (expression argument). Mostramos algunos ejemplos a continuación:

```
> curve(dnorm(x), from = -3, to = 3) # curva normal estándar
```



```
> curve(pnorm(x, mean=10, sd=2), from = 4, to = 16) # CDF normal
```



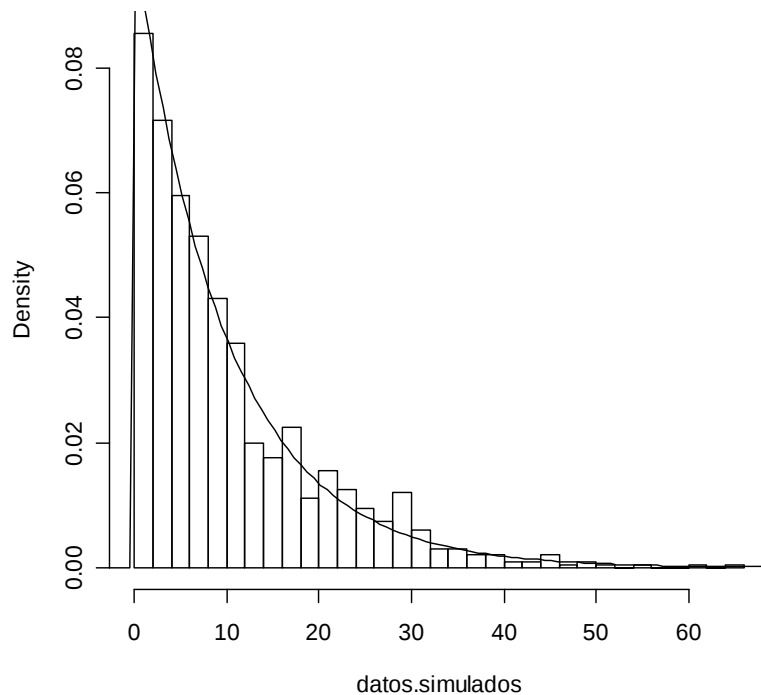
Note que en el primer gráfico hemos restringido los valores al intervalo $-3, 3$ donde la normal estándar tiene la mayor parte de su área. En forma alternativa podríamos haber usado percentiles superior e inferior (digamos del .5% y 99.5%) calculándolos mediante la función `qnorm()`:

```
> curve(dnorm(x), from = qnorm(0.005), to = qnorm(0.995))
```

Cuando se utiliza la función `curve()` para superponer una curva a un gráfico no deben especificarse los argumentos `from` y `to` debido a que R los toma por defecto de los valores mínimo y máximo de x en el gráfico original plot, sí debe ponerse `add = T`. Veamos como se comparan el histograma de 1000 datos simulados de una variable exponencial con $\lambda = 0.1$ y su función de densidad:

```
> datos.simulados <- rexp(1000, rate=.1)
                                     # 1000 Exponenciales
(lambda=0.1)
> hist(datos.simulados, prob = T,
       breaks = "FD", main="V.A. Exponenciales lambda =
0.1")
> curve(dexp(x, rate=.1), add = T # superponer la curva de densidad
```

V.A. Exponenciales $\lambda = 0.1$



6.4 Muestreo aleatorio

En R pueden simularse experimentos aleatorios simples como “elegir un número al azar entre 1 y 100” y “extraer tres bolitas de una urna”. La teoría detrás de este tipo de juegos constituye la base de la teoría de muestreo -sampling theory- (extracción de muestras de poblaciones fijas); además, *los métodos de resampling* (muestreo repetido dentro de una muestra) como el de bootstrap son herramientas importantes en estadística. La función clave en R es la función `sample()`:

```
sample(x, size, replace = FALSE, prob = NULL)
```

Argumentos:

x: un vector (numérico, complejo, de carácter o lógico) de más de un elemento del que se realizará la elección, o un entero positivo.

size: entero no negativo dando la cantidad de ítems a elegir.

replace: indica si el muestreo se realiza con o sin reemplazo.

prob: un vector opcional de pesos que da la probabilidad con que cada elemento es muestreado.

Ejemplos:

```
> sample(1:100, 1) # elige un número entre 1 y 100
```

```
[1] 88

> sample(1:6, 10, replace = T) # arroja un dado equilibrado 10
veces
[1] 6 4 2 5 5 5 2 2 4 1

> sample(1:6, 10, T, c(.6,.2,.1,.05,.03,.02)) # dado no
equilibrado!!
[1] 1 6 1 1 3 1 3 3 1 1

> urna <- c(rep("rojo",8),rep("azul",4),rep("amarillo",3))
> sample(urna, 6, replace = F) # elige 6 balitas de la urna
[1] "amarillo" "rojo"      "rojo"      "amarillo" "rojo"      "rojo"
```

6.5 Ejercicios

1. Genere 20 observaciones de una distribución binomial con $n = 15$ y $p = 0.2$.
2. Halle el 20^{ésimo} percentil de la distribución gamma de parámetros $\alpha = 2$, $\lambda = 0.1$.
3. Halle $P(T > 2)$ para $T \sim t_8$.
4. Grafique la función de probabilidad puntual de una Poisson ($\lambda = 4$) sobre el rango de valores $x = 0, 1, \dots, 15$.
5. Utilice el método de integración aproximado de Monte Carlo, para estimar

$$\int_0^{\pi/4} \log(1 + \tan^2(x)) dx$$

con $n = 1\,000\,000$

6. Genere 100 observaciones de una distribución Normal ($\mu = 50$, $\sigma^2 = 4^2$). Grafique la función de distribución empírica (empirical cdf) $F_n(x)$ para esta muestra y superpóngala con la verdadera FDA $F(x)$.
7. Simule 25 tiradas de una moneda equilibrada registrando los resultados con “cara” y “ceca.”

Estas primeras 4 guías de R están basadas en
W. J. Owen 2006. **The R Guide**

<http://cran.r-project.org/doc/contrib/Owen-TheRGuide.pdf>