

Tema IR

Introducción al entorno de programación *R*

Bernardo D'Auria

Universidad Carlos III de Madrid

Procesos Estocásticos
Grado en Estadística y Empresa

Objetivo del tema

El **objetivo** principal de esta tema es introducir los comandos básicos de R para:

- Simular variables aleatorias.
- Calcular medidas características de variables y vectores aleatorios



¿Qué es R?

R es un lenguaje de programación, creado por Ross Ihaka y Robert Gentleman, cuya característica principal es que forma un entorno de análisis estadístico para la manipulación de datos, su cálculo y creación de gráficos.

La página principal del proyecto “R-project” es

`www.r-project.org`

En esta dirección se puede descargar gratuitamente el programa, sus manuales y los paquetes añadidos.

Otras direcciones útiles:

- `es.wikipedia.org/wiki/R_(lenguaje_de_programación)`
- `halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/Progra/SobreR.html`

Outline

- 1 Comandos bases de R
- 2 Cálculo de Probabilidades - LGN
- 3 Simulación de un movimiento Browniano



Comandos bases de R para vectores

- Pedir ayuda acerca de una función (por ejemplo la media):
`?mean`
- Generar secuencias: `x <- 1:30`
- Crear un vector de componentes constantes:
`x <- c(10.4, 5.6, 3.1, 6.4, 21.7)`
- Crear un vector llamando una función:
`x <- replicate(n,rpois(N, 5))`
- Crear un vector aplicando una función (ej. square):
`square <-function(s) return(s^2)`
`sapply(1:8,square)`
- Calcular la suma: `sum(x)`
- Calcular la media: `mean(x)`
- Calcular la varianza: `var(x)`
- Función indicadora $1_A(x)$ con $A = (3, \infty)$:
`1 * (x > 3)`
- Dibujar un histograma: `hist(x)`



Generar vectores aleatorios

Generamos vector, x , de tamaño N , con componentes independientes

- Uniforme(4,7): `runif(N, min=4, max=7)`
- Bernoulli(0.3): `rbinom(N, 1, 0.3)`
- Binomial($n = 5$, $p = 0.3$): `rbinom(N, 5, 0.3)`
- Poisson(5): `rpois(N, 5)`
- Exponencial(4): `rexp(N, rate=4)`
- Normal($\mu = 3$, $\sigma = 5$): `rnorm(N, mean=3, sd=5)`

Para sacar n veces un elemento al azar de un vector, v , con una distribución de probabilidad data, p ,

- `sample(v, size=n, prob=p, replace=TRUE)`

Ejemplo:

```
> v <- c(2,4,6)
> p <- c(0.5,0.25,0.25)
> sample(v,size=10,prob=p,replace=TRUE)
[1] 6 4 2 2 2 6 2 6 6 6
```



Comandos bases de R para matrices

- Crear una matriz por columnas:

```
M <- matrix(c(1,2,3,4,5,6),nrow = 2, ncol = 3)
```

$$M = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$

- Calcular la traspuesta: $t(M)$
- Crear una matriz por lineas:

```
M <- t(matrix(c(1,2,3,4,5,6),2))
```

$$M = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

- Producto matricial: $c(10.4, 5.6) \%*\% M$
- Matriz diagonal de diagonal x : $\text{diag}(x)$
- Matriz identidad: $\text{diag}(c(1,1,1))$
- Matriz inversa: $\text{solve}(M)$



Instalar y cargar paquetes

Instalamos el paquete **expm** que permite calcular las potencias de una matriz

```
> install.packages("expm")
```

Cargamos el paquete

```
> require("expm")
```

Definimos la matriz

```
> matP<-matrix(c(1,2,3,4,5,6,7,8,9),3)
```

Calculamos su potencia de orden 4, usando el operador `%^%`

```
> matP %^% 4
```

	[,1]	[,2]	[,3]
[1,]	7560	17118	26676
[2,]	9288	21033	32778
[3,]	11016	24948	38880



Definir una función

Para definir una función, `mySum`, que suma dos números, a y b , y que por omisión asume que $b = 4$

```
mySum<-function(a,b=4){a+b}
```

Y podemos usarla con los comandos

```
> mySum(5)
```

```
[1] 9
```

```
> mySum(5,9)
```

```
[1] 14
```



Comandos if y for

El comando if permite expresiones condicionales, ejemplo

```
> x=5  
> if(x>4){5}else{7}  
[1] 5  
> if(x<4){5}else{7}  
[1] 7
```

El comando for permite generar bucles

```
> x <- 1:10  
> z = NULL # para eliminar valores anteriores de z  
> for(i in x){z <- c(z,2*i)}  
> z  
[1] 2 4 6 8 10 12 14 16 18 20
```



Ejemplo - Distribución Binomial

Generamos una distribución Binomial(7,0.4) y dibujamos su histograma.

```
N <- 10000  
sim <- rbinom(N, 7, 0.4)  
prob <- dbinom(0:7, 7, 0.4) * N  
hist(sim, breaks=(0:8)-0.5)  
points(0:7,prob)  
quartz.save(path.expand("~/binomial.pdf"),type="pdf")
```

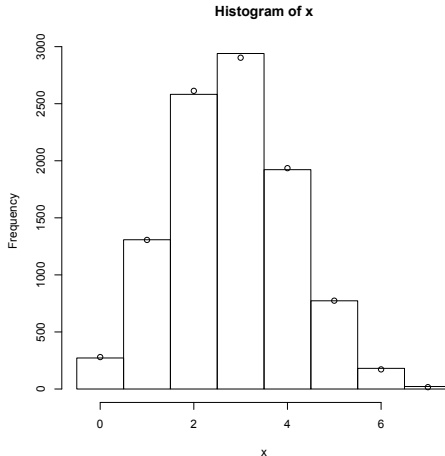


Ejemplo - Distribución Binomial

Generamos
un histograma

SU

```
N <- 1000
sim <- rbinom(N, 7, 0.5)
prob <- dbinom(0:7, 7, 0.5)
hist(sim, freq = FALSE, col = "lightblue", border = "black",
      main = "Histogram of x", xlab = "x", ylab = "Frequency",
      las = 1, cex.lab = 1.5, cex.main = 1.5)
points(0:7, prob, col = "black", pch = 1, cex = 1.5)
```



= "pdf")



Ley de los grandes números - LGN (LLN - en)

Sea X_1, X_2, X_3, \dots una sucesión infinita de variables aleatorias independientes e idénticamente distribuidas que cumple

$$\mathbb{E}[|X_1|] = \mu < \infty ,$$

entonces

$$\lim_{n \rightarrow \infty} \frac{\sum_{i=0}^n X_i}{n} = \mu \quad \text{casi seguramente,}$$

es decir, el promedio de las variables aleatorias converge a μ en un conjunto de probabilidad 1.



Ley de los grandes números - aplicación

Simulamos un numero grande, N , de variables aleatorias

$$Y_i \sim \text{Bin}(7, 0.4), \quad i = 1, \dots, N$$

y definimos

$$X_i(k) \sim 1\{Y_i = k\}, \quad k = 0, \dots, 7$$

las variables indicadoras que nos dicen si cada variables Y_i ha caído en la clase k del histograma. Tenemos que

$$X_i \sim \text{Be}(p_Y(k)) \quad \text{y} \quad \mathbb{E}[X_i] = p_Y(k)$$

y por la ley de los grandes números tenemos que

$$\sum_{i=0}^N X_i \approx p_Y(k) \times N \quad N \gg 1.$$

Es decir el histograma normalizado de las Y_i se aproxima a la función de probabilidad de Y .



Simulación de un movimiento Browniano

Este es un ejemplo de código para generar un aproximación de un movimiento Browniano usando un camino aleatorio

```
num <- 10000  
sim <- runif(num)  
steps <- 2*(sim>0.5)-1  
rw <- cumsum(steps)  
plot(rw)
```



Distribución al tiempo n

Simulamos el camino un número grande de veces y fijamos solo en el valor que toma en el tiempo t . Y partimos el intervalo $[0, 1]$ en un número num de puntos

```
N <- 100  
t <- 4  
num <- 10000  
numsteps <- floor(num*t)  
rwt <- replicate(N, sum(2*(runif(numsteps)>0.5)-1))  
hist(rwt)
```

