

Probabilitat i modelització estocàstica

Pràctica 1 Càlcul de probabilitats amb R

1 El programa R

El programa **R** (versió lliure del programa estadístic **S-PLUS**) es considera un estàndard entre els investigadors o usuaris avançats d'estadística i probabilitats numèriques. Nosaltres veurem la seva utilització per a fer càlculs amb probabilitats i ens limitarem a estudiar les instruccions que necessitem; a l'assignatura d'Estadística en fareu un estudi més complet.

Podeu obtenir la darrera versió del programa (R-3.6.1) al web

<http://cran.r-project.org/>

i instal·lar-lo on vulgueu.

2 La finestra de comandes

L'**R** és un llenguatge de comandes (cal escriure les instruccions) que s'executen a través de l'anomenada finestra de comandes, que és la pantalla que surt quan s'arrenca el programa.

Abans de començar a treballar, al menú principal de la finestra de comandes

File Edit View Misc Packages Windows Help

seleccioneu

File ► Change dir...


i seleccioneu el vostre directori de treball (a algun disc o llaipis de memòria). Quan feu treballs complexos, convé tenir un directori per a cada treball.

A partir de **Help** es poden consultar alguns manuals o demanar ajudes més específiques.

Ara ja podem començar. Després del *prompt* **>** s'escriuen expressions, que al prémer **Enter** s'avaluen. Així,

```
> 2*3      (Enter)
[1] 6
>
```

Uns comentaris generals

- Amb la tecla  es repeteix l'última instrucció introduïda (i iterant, les anteriors). Això és molt útil per a corregir instruccions errònies.
- Quan al prémer **Enter** la instrucció no està acabada, per exemple

```
> 2*      (Enter)
```

aleshores a la pantalla surt un signe +

```
> 2*
+
```

indicant que està esperant rebre més instruccions. Si no sabeu completar la instrucció –perquè no trobeu l'error o alguna altra causa– podeu prémer la icona on diu **STOP**.

- L'**R** és sensible a les majúscules i minúscules.
- **Posar comentaris al programa.** Moltes vegades interessa escriure un comentari per recordar què fa una instrucció o per algun altre motiu. Per tal que l'**R** entengui que es tracta d'un comentari i que no ho ha d'executar, els comentaris han de començar per **#**. Per exemple

```
> sqrt(28)    # sqrt es l'arrel quadrada
```

3 Classes de dades

A l'**R** les dades es classifiquen en

- Numèriques. `2.58`, `pi`, `6.02e23`.
- Complexos. `3+1.23i`. Si fem `sqrt(-4)` dona un error, i cal dir-li que ho tracti com un nombre complex; això s'aconsegueix fent `sqrt(-4+0i)` o `sqrt(as.complex(-4))`
- Caràcter: `"Barcelona"`
- Lògiques, que poden prendre els valors **T** (True) i **F** (False).
- *Missing*, és a dir, una dada que falta a algun lloc o que no entén: posa `NA`, *not available*.

4 Vectors

Per entrar el vector $a = (1, 3, 4, 2)$ es fa

```
> a=c(1,3,4,2)
```

Nota. En versions anteriors de l'**R** calia posar

```
> a <- c(1,3,4,2)
```

Actualment pot fer-se de les dues maneres. De fet, aquesta instrucció assigna el valor (vectorial) (1,3,4,2) a la variable **a**, i per tant l'escriptura amb fletxa expressa millor aquesta operació. D'altra banda, la **c** que hi ha a **c(1,3,4,2)** ve de *concatenar*.

Per veure el vector (o el valor que hi ha a la variable **a**) fem

```
> a          (Enter)
[1] 1 3 4 2
```

Si tenim dos vectors, **a** i **b**, els podem ajuntar amb **c(a,b)**:

```
> a=c(1,3,4,2)
> b=c(10,12)
> d=c(a,b)
> d
[1] 1,3,4,2,10,12
```

Per seleccionar, per exemple, la segona component del vector **a** fem **a[2]**. Per seleccionar la segona, tercera i quarta components, **a[2:4]**.

Algunes instruccions més:

- **rep(5,3)**, escriurà 5,5,5.
- Successions d'enters: **1:5**, que equival a 1,2,3,4,5
- Altres successions: **seq(-pi,pi,by=.5)**, que escriurà $-3.1415, -2.6415, \dots$
O bé **seq(-pi,pi,length=10)**, o **seq(1,by=.05,length=10)**.
- Si el vector té dades de tipus caràcter cal posar-les entre cometes:

```
> lletres=c("A", "B", "C")
```

4.1 Operacions aritmètiques amb nombres o vectors.

Les operacions s'indiquen per **+**, **-**, *****, **/**, **^** (exponent), **sqrt** (arrel quadrada). **Totes les operacions amb vectors es fan component a component.**

Proveu les següents instruccions: Suposem que tenim en memòria el vector **a=(3,4,2,1)**. Aleshores

- ```
> a-1
[1] 2 3 1 0
```

- ```
> 2*(a-1)
[1] 4 6 2 0
```
- Si volem guardar el resultat en una variable, que es digui, posem, `b`,

```
> b=2*(a-1)
```
- ```
> n = 5
> (n-1):2
[1] 4 3 2
```

Veiem-ne un exemple més interessant: Introduïm el pes (en kg) i l'alçada (en m) de 6 persones. L'índex de massa corporal (IMC) es defineix com el pes (en kg) dividit per l'alçada (en m) al quadrat. Aleshores

---

```
> pes=c(60, 72, 57, 90, 95, 72)
> alçada=c(1.75, 1.80, 1.65, 1.90, 1.74, 1.91)
> IMC=pes/alçada^2
> IMC
[1] 19.59184 22.22222 20.93664 24.93075 31.37799 19.73630
```

---

Tal com hem dit, les operacions amb vectors es fan component a component. Però si fem una operació entre dos vectors de diferent mida, aleshores quan s'acaba el vector més curt torna a utilitzar les seves components fins acabar el vector més llarg; s'anomena *regla de reutilització*. Comproveu aquesta propietat amb `a*b` on `a=c(1,2,3,4)` i `b=c(-1,1)`.

### Exercici 1.

Amb els vectors  $a = (2, 1, 4, 7)$  i  $b = (4, 0, -1, 7)$ , calculeu  $c = 2a + 5b$  i  $d = a * b$ , on aquesta darrera expressió vol dir el producte component a component.

## 5 Funcions

### 5.1 Funcions

L'**R** té algunes funcions programades. Utilitzarem

- `max(x)`: dona el màxim del vector `x`; també hi ha `min(x)`
- `sum(x)`: que fa la suma de les components del vector `x`
- `prod(x)`: producte de totes les components
- `choose(n,k)`:  $\binom{n}{k}$
- `factorial(n)`
- `lfactorial(n)`: logaritme del factorial

## 5.2 Programant altres funcions

L'**R** permet programar funcions. Considerem el següent exemple. Per definir

$$f(x) = x^2 - 1,$$

s'escriu

```
> f = function(x){x^2-1}
```

Ara feu `f(2)`. Si tenim `a=(1,2,5,-1,4)`, mireu quan dóna `f(a)`

En general, la instrucció per a definir una funció és

```
nom funcio = function(variable1,variable2,...){operacions amb les variables}
```

Més endavant necessitarem les variacions de  $m$  elements agafats de  $n$  en  $n$ . Òbviament es pot escriure

$$\text{Var}(m, n) = \frac{m!}{(m-n)!},$$

però els factorials de seguida desborden l'ordinador. Per exemple, proveu de calcular  $365!$ . Aleshores s'ha d'utilitzar la funció `choose` o bé escriure una funció amb `prod`.

**Exercici 2.** Escriviu una funció de  $m$  i  $n$  que calculi les variacions  $\text{Var}(m, n)$ . Calculeu  $\text{Var}(365, 10)$ .

## 6 Gràfics

El gràfic més senzill es construeix amb dos vectors numèrics de la mateixa mida, posem `x` i `y`, amb la instrucció

```
> plot(x,y)
```

Proveu-ho amb els vectors `x=c(1, 2, 4, 6, 7)` i `y=c(3, 4, 2, 3, 1)`. Mireu també la instrucció `plot(x,y, type="l")`

Per dibuixar gràfics de funcions podem fer, per exemple,

```
> curve(exp(-x^2),from=-3,to=3,col=2)
```

Podem afegir una llegenda fent

```
> legend('topright', 'exp(-x^2)', lty=1, col=2)
```

També podem afegir rectes horitzontals i verticals pels eixos:

```
> abline(h=0)
> abline(v=0)
```

Diversos gràfics junts:

```
> curve(sin(x),-pi,pi, main="dibuix")
> curve(cos(x),add=T,lty=2,col=2)
> legend('topright',c('sin(x)','cos(x)'),lty=c(1,2),col=c(1,2))
> abline(h=0,v=0)
> points(-1:1,c(0,1,0.5),pch=c(18,20,22), col=c(3,4,5)) #alguns punts
```

Diversos gràfics en una sola finestra:

```
> par(mfrow=c(2,1)) #par(mfrow=c(#files,#columnes))
> curve(sin(x),-pi,pi,lty=3,col=3)
> curve(cos(x),lty=2,col=2)
```

Un altre gràfic molt més flexible:

```
> x=seq(0,2*pi,by=0.1)
> plot(x,sin(x))
> plot(x,sin(x), type="l")
```

Es poden escriure diverses instruccions en la mateixa línia de comandes separades per un punt i coma:

```
> abline(h=0); abline(v=0)
> lines(x,cos(x),type="l", col=2)
```

A base de paciència es poden modificar els eixos, les etiquetes, les escales, les marques sobre els eixos, etc. Mireu al help la instrucció `plot`.

**El problema de l'aniversari.** Recordem que la probabilitat que en un grup de  $n$  persones ( $n \leq 365$ ), almenys dues celebrin l'aniversari el mateix dia és

$$p_n = 1 - \mathbf{P}(A^c) = 1 - \frac{\text{Var}(365, n)}{365^n}$$

L'objectiu és fer un gràfic d'aquestes probabilitats en funció de  $n$ . De cara a aplicar la funció a un vector, no va bé utilitzar la funció `prod` i utilitzarem la fórmula de les variacions amb `choose`

---

```
> Var=function(m,n){choose(m,n)*factorial(n)} #variacions(m,n)
> p=function(n){1-Var(365,n)/365^n} #probabilitat
> n=1:100
> plot(n,p(n),type="l")
```

---

Per veure a partir de quantes persones aquesta probabilitat és del 0.9, podem fer

```
> n[p(n)>=0.9]
```

Intenteu entendre que fa aquesta funció.

Però millor,

```
> min(n[p(n)>=0.9])
```

**Exercici 3. Captura i recaptura.** Anem a fer numèricament el següent problema. Un llac té  $N$  peixos, amb  $N$  desconegut. Per tal d'estimar  $N$  fem el següent: pesquem  $n_1$  peixos, els marquem i tornem al llac. Esperem una estona i pesquem  $n_2$  peixos, dels quals n'hi ha  $m$  de marcats. Suposem:

- La primera vegada es pesquen  $n_1 = 50$  peixos, que es marquen i es tornen al llac.
- La segona vegada també es pesquen  $n_2 = 50$  peixos, dels quals n'hi ha  $m = 3$  marcats.

Designem per  $p_N$  la probabilitat que, si al llac hi ha  $N$  peixos, en traiem exactament 3 de marcats d'entre els 50. Tenim

$$p_N = \frac{\binom{50}{3} \binom{N-50}{47}}{\binom{N}{50}}, \quad N \geq 50$$

Definiu una funció que calculi aquesta probabilitat. Feu un dibuix amb  $N = 50, \dots, 2000$ . Calculeu la  $N$  que maximitza aquesta funció. *Indicació:* Primer trobeu el màxim de  $p_N$  amb la funció `max(...)`.

## 7 Condicionals i bucles

En general, per qüestions d'eficiència en el temps de CPU, amb l'**R** sempre que es pugui cal utilitzar funcions amb vectors i evitar l'ús de condicionals i bucles. Malgrat tot, a vegades no hi ha més remei que utilitzar-los. Aquestes són les instruccions:

**Si...**

```
if (condicio) resultatSI else resultatNO
```

Per exemple, per a  $n \geq 171$  la instrucció `factorial(n)` dóna error. Aleshores podem definir una nova funció:

```
factorial2=function(n){if (n<=170) factorial(n) else print("Nombre massa gran")}
```

Les cometes són necessàries dintre del print, però si es vol que no surtin en pantalla, es posa `print("Nombre massa gran", quote=F)`

**For ...**

```
For (variable en el conjunt E) resultat
```

El resultat es pot posar entre claus, i si es vol en línies successives o separades amb punts i comeses `{operació1; operació2;...}`

Per exemple, suposem que volem fer una taula on hi hagi els productes dels nombres  $n(n+1)(n+2)$  per a  $n = 1, \dots, 10$ . Aquesta funció es defineix amb `prod(n : (n + 2))`. Tal com hem comentat, aquesta funció no es pot aplicar directament a  $n = 1 : 10$ , però nosaltres ho farem mitjançant un `for`. Noteu que primer cal definir un vector on posarem els resultats.

---

```
> h=rep(0,10) #vector on posarem el resultat
> g=function(n){prod(n:(n+2))}
> for(i in 1:10){h[i]=g(i)}
```

---

Mireu el vector `h`

### While ...

S'utilitza de manera similar (mireu el `help`).

## 8 Matrius

Per entrar una matriu cal introduir els números i després les dimensions de la matriu. Així, la instrucció més directa és

```
> b=matrix(c(5,3,-1,3,1,4),nrow=2) # nrow no cal posar-lo:
 # amb el 2 n'hi prou,
 # però així queda mes clar
```

on el 2 del final és el nombre de files. Feu

```
> b
```

i s'obté

```
 [,1] [,2] [,3]
[1,] 5 -1 1
[2,] 3 3 4
```

Noteu que s'omple la matriu per columnes. Si és vol que l'ompli per files cal posar `matrix(c(5,3,-1,3,1,4),2,byrow=T)`, on la `T` és una abreviatura per `True`. Si es vol donar el nombre de columnes enlloc del de files, es posa `matrix(c(5,3,-1,3,1,4), ncol=3)`.

Una segona manera d'introduir una matriu és ajuntant vectors de la mateixa dimensió. Per exemple:

```
> a = c(6,7,8,9)
> b = cbind(a,c(-1,0,3,4))
```

ajunta els dos vectors indicats com a columnes. Per ajuntar-los com a files s'utilitza la instrucció `rbind()`.

D'una matriu de nom `b` en podem demanar la següent informació:



- `dim(b)`, `ncol(b)`, `nrow(b)`, que dona les dimensions, el nombre de columnes, etc.
- `length(b)`, nombre d'elements.
- `dimnames(b)`, noms de les files i les columnes (*si les files i columnes tenen nom.*)
- `b[i,j]` dona l'element  $(i,j)$  de la matriu.
- `b[i,]` la fila  $i$ . Anàlogament amb `b[,j]` s'obté la columna  $j$ .

Finalment, una instrucció útil és la que permet tractar altres objectes com una matriu, per exemple, el vector `a`:

```
> a.mat = as.matrix(a)
```

Aquí la instrucció és `as.matrix`, mentre que el nom que hem posat, `a.mat`, segueix una de les convencions de l'R, que consisteix en posar noms separats per un punt.

## Operacions amb matrius.

- `t(A)` és la transposta de la matriu  $A$ .
- Quan  $A$  i  $B$  són matrius, `A*B` i `A/B` vol dir producte o divisió **element a element**, com en els vectors. El producte ordinari de matrius s'escriu `A%*%B`.
- No hi ha definida la inversa d'una matriu, i cal utilitzar la instrucció `solve`:

```
> c = matrix(c(1,2,3,4),2)
> solve(c)
```

La instrucció `solve` serveix, en general, per resoldre sistemes lineals posant `solve(a,b)` on  $a$  és la matriu del sistema i  $b$  el vector de termes independents.

### Exercici 4.

- Entreu les matrius

$$A = \begin{pmatrix} 2 & 3 & 4 \\ 4 & 0 & -1 \end{pmatrix} \quad \text{i} \quad B = \begin{pmatrix} 0 & 1 & 2 \\ -3 & 2 & 1 \end{pmatrix}$$

Calculeu  $C = A + B$ ,  $D = 2A$ ,  $E = AB^t$ , on aquesta última vol dir el producte ordinari de matrius.

- Sigui

$$F = \begin{pmatrix} 2 & 3 \\ 4 & -1 \end{pmatrix}.$$

Calculeu  $G = F^{-1}$ .

- Amb les notacions de l'exercici 1, calculeu el producte escalar dels vectors  $a$  i  $b$ .

## Exercici 5

Tenim una capsula amb 8 boles. Cinc boles tenen nombres positius: 1,2,3,4,5, i les altres boles nombres negatius: -1,-2 i -3. Traiem dues boles (sense reposició). Volem calcular la probabilitat que el producte dels nombres corresponents sigui positiu. Resoleu aquest problema amb les tècniques habituals.

Anem a resoldre aquest problema amb l'**R** i després comprovar que ambdues resolucions donen el mateix. Per fer-ho, necessitem un paquet addicional que escrigui totes les combinacions de  $n$  elements agafats de  $k$  en  $k$ . Els paquets que estan disponibles al teu ordinador estan en una llista a

[Packages](#) ► [Load package](#) o si utilitzem una versió en castellà [Cargar paquete](#)

El paquet que necessitem es diu [Combinat](#); si està a la llista, pots seleccionar-lo i carregar-lo. En cas contrari, primer cal portar-lo fins a l'ordinador; amb aquest objectiu, feu

[Packages](#) ► [Set CRAN mirror](#)

i seleccioneu algun dels centres que hi ha, per exemple, [0-Cloud](#). Després a [Install package](#) a la llista seleccioneu [Combinat](#). Finalment, carregueu a la memòria aquest paquet a [Load package](#) o [Cargar paquete](#) tal com hem explicat abans.

Per fer el problema, construirem totes les combinacions dels elements 1,2,3,4,5,-1,-2,-3 agafats de dos en dos. Això es fa de la següent manera:

```
> urna=c(1,2,3,4,5,-1,-2,-3)
> CP=combn(urna,2)
```

Llavors **CP** és una matriu  $2 \times N$  on a les columnes hi ha les combinacions, que són els casos possibles. Busqueu els resultats favorables i calculeu la probabilitat demanada; intenteu-ho fer el més automàtic possible. Compareu-la amb el resultat que heu obtingut abans.