

Práctica 3

Índice:

- 1) Representación gráfica de funciones
- 2) Ejercicio bucles
- 3) Bucles anidados
- 4) Funciones en R

1) Representación gráfica de funciones

En este caso, se nos pide representar una gráfica que contenga los meses y la producción de manzanas para cada mes.

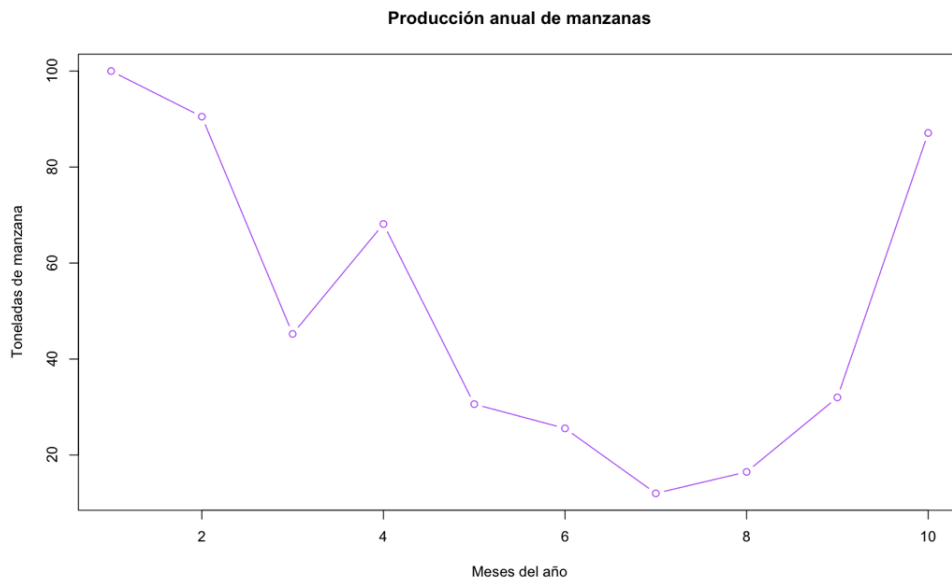
1- Para ello, debemos construir un vector que contenga meses, utilizando el comando `seq(valor inicial, valor final, incremento)`, y otro que contenga la producción de manzanas, los cuales se toman de la siguiente tabla:

1	2	3	4	5	6	7	8	9	10
100	90.50	45.23	68.14	30.60	25.55	12.01	16.48	32	87.10

2- A continuación, se representarán los datos utilizando el comando `plot(vector1, vector2, type="b", col="blue")`. Type indicará el tipo de gráfico realizado ("l" líneas, "b" una los puntos, "h" histogramas, "s" con escalones, ...) y col el color asignado. Además, nombraremos los ejes como "meses del año" y "Toneladas de manzanas", utilizando los comandos `xlab="nombre"` e `ylab="nombre"`; seleccionaremos el símbolo que marca los puntos con `pch= número entre 0-25` y nombraremos el gráfico con `main="título"`.

Código

```
meses= seq(1,10,1) } 1
manzanas= c(100,90.5,45.23,68.14,30.60,25.55,12.01,16.48,32,87.10)
plot(meses,manzanas,type="b",col="blue",xlab="Meses del año",ylab="Toneladas de
manzana", main="Producción anual de manzanas",pch=1) } 2
```



2) Ejercicio bucles

Utilizando la estructura básica de bucle `for (in in vfin: vfin) { Proceso de cálculo }` y con los vectores `v= (12, -3, 5, 18.7)` y `w= (12, 0.25, 77, exp(2))`

1- Primero debemos obtener la suma de los dos vectores, para lo que usaremos un bucle y sumaremos componente a componente cada vector. Comprobación `v+w`.

2- Después, tendremos que obtener la suma de los componentes del vector `v` y almacenarlos en `SumaC`. Para ello realizaremos un sumatorio donde iremos añadiendo las componentes de dicho vector.

3- Tercero, hay que obtener el producto escalar de `v` y `w`, para lo que usaremos también un sumatorio, añadiendo esta vez la multiplicación de cada componente de los vectores (se podrá comprobar con `v%*%w`).

4- Más tarde, debemos multiplicar los vectores componente a componente (comprobando con `v*w`).

5- Realizaremos la operación `z[j]= v[j] + 2w[j]` con `j=1, length(v)`

6- Finalmente, construiremos una tabla 2x2 que contenga "Suma" y el resultado del apartado 2 y "Producto escalar" y el resultado del apartado 3. Para ello creamos dos vectores: `vnombres= ("suma", "Producto escalar")` `vvalores= (SumaC, Pesc)`. Ahora creamos la tabla con el comando `data.frame (v1, v2)`.

IMPORTANTE Cuando creamos un vector o un sumatorio, hay que iniciarlo a 0 primero.

Código

```
v= c(12,-3,5,18.7)
```

```
w= c(12,0.25,77,exp(2))
```

```
u= 0
```

```
for (i in 1: length(v)) {
```

```
  u[i]=v[i]+w[i]
```

```
}
```

```
v+w
```

```
SumaC=0
```

}

1

```

for (i in 1: length(v)) {
  SumaC=SumaC+v[i]
}
Pesc=0
for (i in 1:length(v)) {
  Pesc=Pesc+v[i]*w[i]
}
v%%*%w
Mult=0
for (i in 1:length(v)) {
  Mult[i]=v[i]*w[i]
}
for (j in 1: length(v)) {
  z[j]= v[j] + 2*w[j]
}
v*w

a= c("Suma", "Producto escalar")
b= c(SumaC, Pesc)
data.frame (a,b)

```

	a	b
1	Suma	32.7000
2	Producto escalar	666.4253

3) Bucles anidados

- 1- Construiremos una matriz A1 con los vectores v y w del ejercicio anterior y estos serán sus filas. Para ello, utilizaremos `rbind(v1,v2)`
- 2- Construimos una matriz A2 con los mismos vectores, pero en este caso, siendo sus columnas. Para ello nos serviremos de `cbind (v1, v2)`
- 3- Multiplicamos utilizando bucles anidados las matrices, obteniendo la matriz C. Para que podamos multiplicar dos matrices, la primera matriz debe tener el mismo número de columnas que la segunda de filas. Es por esto que nombramos `m=nrow(A1)` (filas de la primera matriz), `n=ncol(A2)` (columnas de la segunda matriz) y, más importante, `p=ncol(A1)`. Así nos aseguramos que se cumpla `m x p` `p x n`. Una vez realizado esto, creamos la matriz C con el comando `matrix= (c(0), nrow=m, ncol=n)` y las multiplicamos. Para ello, abrimos tres bucles y realizamos un sumatorio. Añadiremos a la componente de la matriz C la multiplicación de los componentes de las matrices A1 y A2. (`A1%%*%A2`)
- 4- Inventamos una matriz D con el comando `matrix` mencionado anteriormente, pero esta vez añadiendo un vector (valores de la matriz), y la sumamos con la matriz C obtenida en el apartado anterior. Para ello iniciamos a 0 la matriz CD, ya que va a ser un sumatorio, y sumamos las componentes con un bucle. Esto se podrá comprobar con `C+D`
- 5- Finalmente, multiplicaremos las matrices C y D elemento a elemento, por lo que será distinto del apartado 3. Para ello, simplemente multiplicaremos los componentes de cada una de estas matrices en un bucle. Se podrá comprobar con `C*D`

Código

```

A1=rbind(v,w)

```

```

A2=cbind(v,w)
m=nrow(A1)
n=ncol(A2)
p=ncol(A1)

C=matrix(c(0),nrow=m,ncol=n)
for (i in 1:m){
  for (j in 1:n){
    for (k in 1:p){
      C[i,j]=C[i,j]+A1[i,k]*A2[k,j]
    }
  }
}
A1%%A2

```

3

```

D=matrix(c(2,2,5,1),nrow=2,ncol=2)
CD=matrix(c(0),nrow=2, ncol=2)

for (i in 1:2){
  for(j in 1:2){
    CD[i,j]=D[i,j]+C[i,j]
  }
}
C+D
D=matrix(c(2,2,5,1),nrow=2,ncol=2)
CD=matrix(c(0),nrow=2, ncol=2)

for (i in 1:2){
  for(j in 1:2){
    CD[i,j]=D[i,j]*C[i,j]
  }
}
C*D

```

4

5

4) Funciones en R

- 1- Definir la función $f(x) = x^2 \cos(x^2)$, para lo que utilizaremos el comando `f= function(x) {introducimos la función deseada, en este caso, x^2*cos(x^2) }` y obtenemos el valor de $f(\pi/4)$.
- 2- Obtenemos un vector `xx` utilizando el comando `seq`, en este caso, `seq(0,10,0.01)`
- 3- Representamos la función con los valores `xx` en las abscisas y $f(xx)$ en las ordenadas. Nombraremos los ejes con `xlab` e `ylab`. Su color es verde y `type="b"`.
- 4- Definimos la función $g(x) = \sin(x^2) e^{\frac{-x^2}{10}}$ y la añadimos a la gráfica, para lo que utilizaremos el comando `par(new="true")`. Esta función llevará `xx` en las abscisas y $g(xx)$

en las ordenadas. Además, será de color azul (`col="blue"`); utilizará `pch=4` y eliminaremos los ejes con el comando `axes=false`

5- Definimos la función $h(x) = \sin(x^8)e^{-x}$ y la añadimos a la gráfica de la misma manera. Esta función llevará xx en las abscisas y h(xx) en las ordenadas; tendrá que ir en rojo (`col="red"`) y con `pch=18`.

6- Finalmente añadimos una leyenda. Para ello, utilizaremos el comando `legend(x="localización en el gráfico", c(v1,v2,v3), fill=c("color1", "color2", "color3"))`.

IMPORTANTE Es recomendable poner todas las funciones al principio juntas. Además, se pueden establecer límites para que queden mejor representadas (`xlim= c(liminf, limsup)` e `ylim= c(liminf,limsup)`).

Código

```
f<-function(x) {  
  x^2*cos(x^2)  
}  
f(pi/4)
```

```
g=function(x) {  
  sin(x^2)*exp(-x^2/10)  
}
```

```
h=function(x) {  
  sin(x^2)*exp(-x)  
}
```

```
xx=seq(0,10,0.01)
```

```
plot(xx,f(xx),xlab="Abscisa", ylab="Mis funciones f,g,h", col="green",xlim=c(0,10),  
      ylim=c(-1,1), type='b')  
par(new="true")  
plot(xx,g(xx),col="blue",xlab="",ylab="",xlim=c(0,10), ylim=c(-1,1), axes=FALSE, pch=4)  
par(new="true")  
plot(xx,h(xx),col="red",xlab="",ylab="",xlim=c(0,10), ylim=c(-1,1),axes=FALSE, pch=18)
```

```
legend(x="top",c("funcion f","funcion g","funcion h"), fill=c("green","blue","red"))
```

