

Programación y Uso de Librerías en R: Herramientas de Análisis y Visualización de Datos

Juan Luis Peñaloza Figueroa
Universidad Complutense de Madrid

Milagros Dones Tacero
Universidad Autónoma de Madrid

Carmen Gladys Vargas Pérez
Universidad Complutense de Madrid

AÑO: 2025

SCRIPT_14: CAPÍTULO XVI: ESTRUCTURAS DE PROGRAMACIÓN Y CREACIÓN DE FUNCIONES EN R

```
## ESTRUCTURAS DE PROGRAMACIÓN
# BUCLE FOR
for (i in listadevalores) { secuencia de comandos }
> for (i in -5:5) print(i)
> for(i in 1:10) { print(i) }
> x<-seq(-10,10)
> plot(x,x,xlim=c(0,10),ylim=c(0,10))
> for(i in 1:10)
+ abline(h=i,col=i)
> for(i in 1:10)
+ abline(v=i,col=i)
# if dentro de un bucle for
> a<-5
> for(i in 1:10){
+ cat(i,"\n")
+ if(i==a) cat("i vale5\n")}
1
2
3
4 5
i
vale5
6
7
8
9
10
# anidar bucles for
> MatrizHilbert <- function(m,n) {
+   y <- matrix(1,nrow = m,ncol = n)
+   for (i in 1:m) {
+     for (j in 1:n) {
+       y[i,j] = 1/(i+j-1)
+     }
+   }
+   return(y)
+ }
```

```

> MatrizHilbert(3,3)
# BUCLE WHILE
> umbral <- 2000
> n <- 0
> cuadrado<-0
# Bucle while(valor <= umbral)
> while(cuadrado<=2000){
+ n<-n+1
+ cuadrado<-n^2
+ }
> n
> cuadrado
# crear un bucle while en R
> x<-c(1,2,3,4)
> y<-c(0,0,5,1)
> n<-length(x)
> i<-0
> k<-numeric(n)
# Bucle while
> while(i <=n){
+ k[i] <-x[i] + y[i]
+ i<-i+1}
> k
# EJECUCIÓN CONDICIONAL: IF
> n<-10
> pares<-c()
> impares<-c()
> for(i in 1:n){
+ if(i%%2==0) pares<-c(pares,i)
+ else impares<-c(impares,i)}
> pares
> impares
# BUCLE REPEAT
i = 1 repeat { print(i) i <- i+1 if (i==5) break}
# BREAK EN UN FOR
> matriz <- matrix(0,nrow = 5 ,ncol = 5)
> for(i in 1:dim(matriz)[1]) {
+ for(j in 1:dim(matriz)[2]) {
+ if (i == j) break
+ else matriz[i,j] <- i*j
+ }
+ }
> matriz
# COMANDO NEXT
> x <- 1:5
> for (val in x) {
+ if (val == 3){
+ next
+ }
+ print(val)
+ }
# CREACIÓN DE FUNCIONES EN R
# Función desviación típica

```

```

> desv<- function(x){sqrt(var(x))}
> x<-1:10
> desv(x)
> sd(x)
> x<-matrix(rnorm(15),nrow=3)
> x
> apply(x,2,desv)
# ALCANCE DE LAS VARIABLES
> y <- 10
> cuadrado <- function(x){ y <- x^2 ; return(y) }
> x<-2
> cuadrado(x)
> y
# PARÁMETROS POR DEFECTO
> desv <- function(x,n=length(x)-1){ sum((x-mean(x))^2)/n}
> x<-1:10
> desv(x)
> desv(x,10)
# FUNCIONES CON UN NÚMERO VARIABLE DE ARGUMENTOS
> f<-function(...){ L <- list(...) ; return(L) }
> f(1,2,3)
> f(c(1,2),c(3,4,5))
# O también
> x<-rnorm(100)
> y<-runif(50)
> f(x,y)
# CREACIÓN DE FUNCIONES MATEMÁTICAS
# FUNCIÓN SUMA
> suma(x=2, y=3)
> suma <- function(x, y) { return(x + y) }
# Aplicación
> suma(x=4, y=6)
# FUNCIÓN CUADRÁTICA
> f <-function(x){ x^2+2 }
> x = 0
> f(0)
# FUNCIONES EN 3 DIMENSIONES
> Z<-function(x,y){ x + 4 + 4*y }
> x = 0
> y = 2.
> Z(0,2)
#Función potencia
> potencia(2,3)
# Función a trozos con condicionales (if)
> ftrozos(3)
> ftrozos(5)
# Función varianza
> x<-1:10
> cuasivar(x)
> n1<-length(x)
> varpob<-function(x, n=length(x)){sum((x-mean(x))^2/n)}
> varpob(x)
# Función desviación típica
> y<-c(12,25,32,15,26)
> desv2(y)
# Función Coeficiente de variación
> y<-c(12,25,32,15,26)
> coef_var<-function(x,na.rm=F){
+ sd(x,na.rm=na.rm)/mean(x,na.rm=na.rm)
+ }
> coef_var(y)

```

```

# Función de densidad N( $\mu$ ,  $\sigma$ )
> N<- function(x) { ..... }
> N(0.2)
> plot(N)
# Función de percentiles o cuantiles
> dat1<-rnorm(100)
> iqr(dat1)
> IQR(dat1)
# Función hipotenusa de un triángulo
> hipot(2,4)
> hipot1(2,4)
# Función coseno
> par(mfcol=c(1,3))
> fcoseno()
> fcoseno(w=2,col="red",type="l",lwd=2)
> fcoseno(w=3,col="blue",min=-3*pi)
# Función Lanzar un dado
> par(mfcol = c(1, 3))
> dado(100)
> dado(500)
> dado(100000)

```