

CHULETARIO PRÁCTICA 2 Y 3

Práctica 2

Para generar vectores aleatorios siguiendo un intervalo:
`seq(valor inicial, final, incremento).`

Para generar valores aleatorios en un intervalo:
`runif (valores a generar, valor mín, valor máx)`

Comando rbind y cbind:

Para combinar vectores, matrices o data.frame por filas (rbind) o columnas (cbind).

Suponiendo que los elementos a combinar son v y w (antes definidos):

- **Si son vectores:** se forma una matriz de dos filas, siendo v la 1ª fila y w la 2ª.
- **Si son matrices:** si tienen el mismo número de columnas, se pueden combinar por filas.

El resultado de ese comando se puede almacenar en otro elemento que contendrá la matriz o tabla resultante de combinar los elementos por filas o columnas.

Operaciones con bucles

Vectores:

Suma:

Definir primero el vector resultado de la suma. Ej. `u <- v + w`. Definido: `u <- c(0)`.

Iniciar bucle: **Estructura**

```
for (i in valor inicial: valor final) {  
  u[i] <- v[i] + w[i]  
}
```

Comprobación:

- Escribir u.
- Comparar con resultado de v+w.

Considerando:

Valor inicial: 1

Valor final: length (v)

definiendo `n <- length(v)`

```
for (i in 1:n) {  
  u[i] <- v[i] + w[i]  
}
```

Hemos sumado las componentes [i] de los vectores.

Producto de dos vectores mediante bucles (v y w):

Para definir el producto: `producto <- 1`, para que el resultado pueda ser ≠0.

Iniciar bucle: **Estructura**

Considerando:

Valor inicial: 1

Valor final: length (v)

definiendo `n <- length(v)`

```
for (i in valor inicial: valor final) {  
  producto <- producto + (v[i] * w[i])  
}
```

Comprobación:

`v%o * %ow`

```
for (i in 1:n) {  
  producto <- producto + (v[i] * w[i])  
}
```

Multiplicación de dos vectores componente a componente:

Definir vector (en este caso utilizo "multiplicación"): **multiplicación <- c(0)**

Iniciar bucle: **Estructura:**

Considerando:

Valor inicial: 1

Valor final: length (v)

definiendo n <- length(v)

```
for (i in valor inicial: valor final) {  
  producto [i] <- v[i] * w[i]  
}
```

Comprobación:

v * w

```
for (i in 1: n) {  
  multiplicación [i] <- v[i] * w[i]  
}
```

Operaciones propuestas mediante bucles

Operación: zj = vj+2wj

Definir primero z-< c(0)

```
for (i in 1: n) {  
  z[j] <- v[j] + 2 * w[j]  
}
```

Multiplicación de matrices con bucles anidados

Definir las matrices que se vayan a usar. Ej. A1 y A2.

Resultado de la multiplicación C <- A1 * A2

Definir C <- matrix (c(0))

```
for ( i in valor inicial1: valor final1) {  
  for (j in valor inicial2: valor final 2) {  
    for (k in valor inicial3: valor final3)  
      C[i, j] <- C[i, j] + A1[i, k]*A2[k, j]  
    }  
  }  
}
```

Considerando:

Valores iniciales: 1

Valor final1: length (v)

Definiendo m <- nrow (A1), n <- ncol(A2), p <- ncol(A1)

```
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]  
    }  
  }  
}
```

Suma de matrices mediante bucles

Definir la matriz resultante de la suma. Ej. E <- C + otra matriz (D)

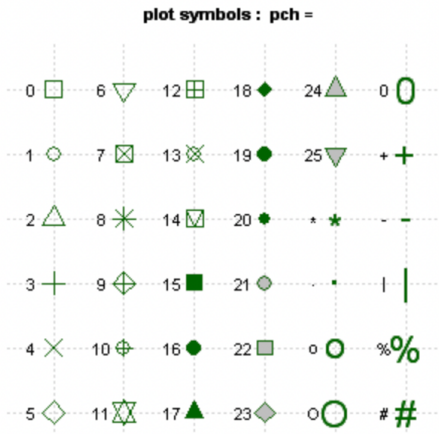
```
for (i in: m){  
  for(j in 1: n){  
    E[i, j] <- E[i, j] + C [i, j] + D[i, j]  
  }  
}
```

Representación de gráficas:

Comando:

plot (vector, vector, col= " ", type= " ", xlab=" ", ylab= " ", main=" ", pch= __)

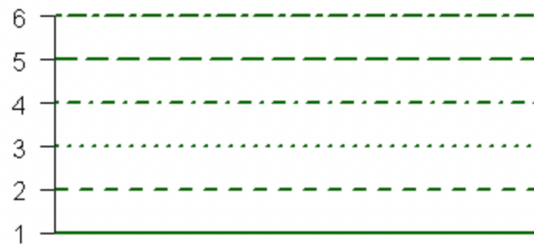
col =	Color con el que representas en gráfica
type = "b"	Se debe dibujar un gráfico de líneas y puntos
xlab = " "	Nombre del eje de abscisas
ylab = " "	Nombre del eje de ordenadas
main = " "	Nombre de la gráfica
pch =	Determina los símbolos de los puntos
axes = FALSE	Para eliminar ejes
par(new=TRUE)	Superponer múltiples gráficos en la misma ventana. Útil para comparar o combinar gráficos.



Además, para definir el estilo de línea y grosor, podemos añadir al comando lo siguiente:

- **lty**: tipo de línea.
- **lwd**: grosor (lwd = 2 es el doble de grosor...).

Line Types: lty=



Leyenda de la gráfica

Comando:

legend (x o y= "", c ("", " ", fill = c("color", "", ""))

Dónde colocar la leyenda	Dirección	x=	horizontal
		y=	vertical
	Zona	"top"	arriba
		"bottom"	abajo
		"right"	derecha
"left"	izquierda		
Ejemplo:	c ("", "", ...)	Proporciona etiquetas a los elementos identificados por la leyenda	
Ejemplo:	fill = c ("", "" ...)	Para ponerle color a las componentes del vector, colocar en cada componente el color que quieres asignarle.	
	fill = c ("green", "blue", "red")		

Representar funciones en gráficas

> Nombre de la función <- function (x) (función dada en ejercicio).

Si piden definir un vector que tome valores en un intervalo → seq (valor inicial, valor final, saltos)

plot (vector, nombre de función(vector), type="h", xlab="", ylab="", axes=FALSE, pch= , col=)

Para representar varias funciones en un mismo gráfico:

1. Definir el vector que se utiliza.
2. f <- function (x)(...)
3. par(new=TRUE)
4. plot

Representación de tablas:

Comando:

data.frame (vector definido antes, vector definido antes)