



# Sesión de prácticas basadas en statsTeachR

<http://www.statsteachr.org/>

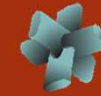
Prof. Magdiel ABLAN BORTONE

Prof. Paolo RAMONI PERAZZI



Facultad de Ingeniería

Universidad de Los Andes



# introR : Introduction to Programming with R

Autora: Andrea S Foulkes

- **Descripción**
- Este módulo ofrece una introducción para programar en R a un nivel elemental. Sub-tópicos adicionales incluyen la creación de vectores y matrices, manipulación de objetos y resúmenes básicos de los datos. Incluye tres laboratorios diseñados para ser completados usando R/RStudio.

- **Ejercicio 1. Vectores**

1. Cree una variable llamada "a" con un valor de "uno" y muestre el resultado

```
> a <- 1
> print(a)
> a
```

2. Cree un vector numérico llamado "b" con los elementos 1, 2 y 3, al menos de tres modos diferentes

```
> b <- c(1:3)
> assign("b", c(1, 2, 3))
> b <- seq(1, 3)
```

3. Cree los siguientes vectores: (a) (1 3 5); (b) (1 2 3 0 1 2 3); (c) (1 1 1 1); y (d) (1 2 3 1 2 3 1 2 3)

```
> seq(1, 6, 2)
> c(b, 0, b)
> rep(1, 4)
> rep(b, 3)
```

- **Ejercicio 1. Vectores**

4. Cree vectores de caracteres que contengan: (a) los nombres de al menos cinco personas en esta clase;

```
> nombres <- c("Magdiel", "Alejandra", "Oswaldo",  
              "Carlos", "Yorman")  
  
> nombres
```

(b) los valores x1, x2, x3 y x4.

```
> etiquetas <- paste("X", 1:4, sep="")  
  
> etiquetas
```



- **Ejercicio 2. Matrices**

1. Cree las siguientes matrices: (a) una *matriz de identidad* de  $3 * 3$  con 1s en la diagonal y 0s en lo demás

```
> I <- diag(1,3)
```

```
> I
```

(b) Igual pero con los valores 1, 2 y 3 en la diagonal

```
> diag(b)
```

(c) Una matriz vacía con dos líneas y tres columnas

```
> A <- matrix(nrow=2,ncol=3)
```

```
> A
```



- **Ejercicio 2. Matrices**

2. Cree las siguientes matrices:

$$(a) \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix} \quad (b) \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

```
> A <- matrix(seq(1,6),2)
```

```
> A
```

```
> matrix(seq(1,6),2,byrow=TRUE)
```



- **Ejercicio 2. Matrices**

3. Cree las siguientes matrices:

$$(a) \begin{pmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{pmatrix} \quad (b) \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \end{pmatrix}$$

```
> B <- array(seq(1,12),dim=c(3,4))
```

```
> B
```

```
> array(seq(1,3),dim=c(3,4))
```



- **Ejercicio 3. Matrices**

4. Cree las siguientes matrices:

$$(a) \begin{pmatrix} 1 & 4 \\ 2 & 4 \\ 3 & 4 \end{pmatrix} \quad (b) \begin{pmatrix} 1 & 2 & 3 \\ 4 & 4 & 4 \end{pmatrix}$$

```
> cbind(b, rep(4, 3))
```

```
> rbind(b, rep(4, 3))
```



- **Ejercicio 3. Operaciones con matrices**

1. (a) Agregue una columna con los elementos 10 y 11 a la matriz A

```
> cbind(A,c(10,11))
```

(b) Agregue una línea con los elementos 10, 11 y 12 a la matriz A

```
> rbind(A,c(10,11,12))
```



- **Ejercicio 3. Operaciones con matrices**

2. Usando la matriz B separe lo siguiente: (a) el valor de la primera línea y segunda columna

```
> B
```

```
> B[1,2]
```

(b) La primera línea

```
> B[1,]
```

(c) La segunda columna

```
> B[,2]
```

(d) Las primeras dos líneas

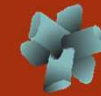
```
> B[1:2,]
```

(e) Las primera y tercera columnas

```
> B[,c(1,3)]
```

(f) Los elementos de la diagonal

```
> diag(B)
```



- **Ejercicio 3. Operaciones con matrices**

3. Usando la matriz A: (a) reemplace el elemento 1,2 con el valor 400

```
> A[1,2] <- 400
```

```
> A
```

(b) Luego reemplace cualquier elemento que sea igual a 400 con "NA"

```
> A[A==400] <- NA
```

```
> A
```



- **Ejercicio 4. Algunas operaciones simples**

```
> a <- 1
> a
> b <- c(1,2,3)
> a+b
> 2*b
> sum(b)
> prod(b)
> sqrt(b)
> 1/b
> sort(1/b)
> order(1/b)
> b==2
> b>=2
> b<2
```



- **Ejercicio 4. Algunas operaciones simples**

```
> A <- matrix(seq(1:12),3)
```

```
> A
```

```
> t(A)
```

```
> A*2
```

```
> A%*%c(1,1,0,0)
```

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

```
> a*b
```

```
> sum(a*b)
```

```
> a%*%b
```



## Lazos

- **Descripción**
- ‘Replicar’, ‘iterar’, etc., instrucciones es una práctica que se originó antes de la invención de la computadora, es común en la computación y formará parte muy importante de este postgrado.
- No es otra cosa más que automatizar un cierto proceso de múltiples pasos, organizando secuencias de acciones y agrupando las partes que requieren repetición.
- Anteriormente se solía “saltar” de regreso a una de las instrucciones previas del código.
- Permiten compactar el código
- Existen diversas variantes, particularmente:
  - Lazos que se ejecutan un número preestablecido de veces, indicadas por un índice, los cuales pertenecen a la familia `for`;
  - Lazos que corren mientras no se verifique una condición lógica, pertenecientes a la familia `while`;



- **El lazo for**

```
> # DESCIFRA EL MENSAJE OCULTO
> # Vector con las letras del alfabeto inglés
> LETTERS

> # Crea un vector numérico
> vector <- c(8, 1, 7, 1, 14, 12, 1, 20, 1, 18, 5, 1)
> # Crea un vector vacío
> mensaje <- NULL
> # El lazo traduce el mensaje
> for(i in length(vector): 1){
>   mensaje[i] <- LETTERS[vector[i]]
> }
> print(mensaje)
```



- **El lazo for**

Por supuesto, nada impide “anidar” lazos

```
> # UNA TABLA DE MULTIPLICAR

> # Crea una...
> tabla = matrix(nrow=10, ncol=10)
> # Lazo anidado
> for(i in 1:dim(tabla)[1]){ # Para cada línea
>   for(j in 1:dim(tabla)[2]){ # Para cada columna
>     tabla[i,j] = i*j      # Asigna valores
>   }
> }
> print(tabla)
```





- **El lazo while**

Lazo condicional

```
> prueba <- "X"
> entrenamiento <- "Y"
> contador <- 0

> while(entrenamiento != prueba){
>   contador <- contador + 1
>   entrenamiento <- sample(LETTERS, 1)
> }
> contador
```



- **El lazo while**

Ídem: nada impide anidar lazos, inclusive diferentes

```
> prueba <- "X" # Parámetro de prueba
> entrenamiento <- "Y" # Parámetro de entrenamiento
> contador <- NULL # Un vector contador

> for(i in 1:10){ # 10 réplicas
>   contador[i] <- 0
>   while(entrenamiento != prueba){
>     contador[i] <- contador[i] + 1
>     entrenamiento <- sample(LETTERS, 1)
>   }
>   entrenamiento <- "Y"
> }
> contador
```



## graphics : Graphic Methods

Autor: Eric A Cohen

- **Descripción**

- Este módulo se enfoca en métodos gráficos disponibles en R para la análisis exploratorio y presentación de resultados, incluyendo: stripcharts, histogramas, tallos y hojas, gráficas de densidad, boxplots, tortas, barras apiladas. Otras representaciones gráficas son mencionadas brevemente. Se espera que el usuario aprenda a crear las gráficas y leer la información esencial que éstas proveen.

- **Ejercicio 1. Stripchart**

1. Cambie el directorio de trabajo y cargue los datos

```
> getwd( )  
> setwd( "C: /DIRECTORIO/CON/LOS/DATOS" )  
> dir( )  
> load( "data_for_graphing.RData" )
```

2. Genere una stripchart de data\_1:

```
> stripchart( data_1 )
```

3. Genere una stripchart de data\_2:

```
> stripchart( data_2 )
```

4. Dele un vistazo a data\_2:

```
> data_2  
> str( data_2 )  
> summary( data_2 )
```

¿Hay datos repetidos? ¿Cree que esta gráfica resuelve su problema? ¿Cómo se mejora?