Ejercicios Estructura de Datos

Vectores

Ejercicio A1. Múltiplos de 2 desde el 1 hasta el 49.

In [3]:
```
seq(2, 49, by = 2)
```

2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48

In [4]:
```
2*1:24
```

2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48

Ejercicio A2. Cuatro veces el valor 3.

In [5]:
```
c(3,3,3,3)
```

3 3 3 3

In [6]:
```
rep(3,4)
```

3 3 3 3

Ejercicio A3. Los valores del 1 al 5.

In [7]:
```
1:5
```

1 2 3 4 5

In [13]:
```
seq(5)
sequence(5)
```

1 2 3 4 5

1 2 3 4 5

Ejercicio A4. Los valores 2, 3, 1, 2, 1.

In [8]:
```
c(2,3,1,2,1)
```

```
2  3  1  2  1
```

Ejercicio A5. Los valores “naranja”, “rojo”, “azul”, “amarillo”, “morado”.

In [10]:
```
c("naranja", "rojo", "azul", "amarillo", "morado")
```

```
'naranja'  'rojo'  'azul'  'amarillo'  'morado'
```

Ejercicio A6. Los valores en la posición 10 al 12 del vector del ej. A1.

In [19]:
```
EA6 <- seq(2, 49, by = 2)
EA6[10:12]
```

```
20  22  24
```

In [20]:
```
EA6[10]
EA6[11]
EA6[12]
10:12
```

```
20
```

```
22
```

```
24
```

```
10  11  12
```

Ejercicio A7. Los valores del vector del ej. A1, pero sin los primeros 5 elementos.

In [28]:
```
EA6[-5:-1]
EA6
```

```
12  14  16  18  20  22  24  26  28  30  32  34  36  38  40  42  44  46  48
```

```
2  4  6  8  10  12  14  16  18  20  22  24  26  28  30  32  34  36  38  40  42  44  46  48
```

In [31]:
```
EA6[c(1:5,8:10)]
```

```
2  4  6  8  10  16  18  20
```

Ejercicio A8. Los valores ordenados del vector del ej. A4.

In [32]:
```
EA4 <- c(2,3,1,2,1)
```

In [33]:
```
order(EA4)
```

```
3  5  1  4  2
```

In [34]:
```
EA4[order(EA4)]
```

```
1  1  2  2  3
```

Ejercicio A9. El vector que contiene al vector del ej. A4 y el del ej. A3 concatenados.

In [35]:
```
EA3 <- 1:5
```

In [36]:
```
c(EA4,EA3)
```

```
2  3  1  2  1  1  2  3  4  5
```

Ejercicio A10. El vector con los valores del vector del ej. A5 ordenados de la “z” a la “a”.

In [37]:
```
EA5 <- c("naranja", "rojo", "azul", "amarillo", "morado")
```

In [39]:
```
EA5[order(EA5, decreasing=TRUE)]
```

```
'rojo'  'naranja'  'morado'  'azul'  'amarillo'
```

Ejercicio A11. Generar el vector $(1, 2, 3, 4, 5, 6, 7, 8, 7, 6, 5, 4, 3, 2, 1)$.

In [ ]:
```

```

Ejercicio A12. Generar el vector $(1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5)$.

In [ ]:
```

```

Ejercicio A13: ¿Cómo se verán los vectores `x`, `y` y `z` en cada línea de código?

```
rm(list=ls())
x <- 1
x[3] <- 3
y <- c()
y[2] <- 2
y[3] <- y[1]
y[2] <- y[4]
z[1] <- 0
```

In [ ]:
```

```

Ejercicio A14: ¿Cuánto valen los vectores `d` y `e`?

```
rm(list=ls())
a <- c(5,2,8,3,2,1,7)
b <- c(1,5,3,0,1,5,3)
c <- 2
d <- (a<b)*(a*b) + (a>b)*(a+b)
d <- d-c
e <- c^d
```

In [ ]:

Ejercicio A15: Calcular el dígito de las decenas del valor de la variable `x` (con cualquier valor que le asignes).

In [ ]:

Ejercicio A16: Obtener el vector que contenga todos los enteros del 1 al 100 que no sean divisibles por 2, 3 o 7. Construye una matriz identidad de 10 x 10. Ahora, usando código en R, convierte todos los elementos distintos a cero en 5. Hazlo de dos maneras distintas.

In [ ]:

Ejercicio A17: Cuatro estudiantes de licenciatura en FCFM desean hacer trámites en el departamento escolar de la Facultad. Podemos representar esta lista de espera del departamento escolar como `queue <- c("Alan","Belén","Cristóbal","Diana")`, donde "Alan" es el primero de la fila y "Diana" es la última. Escribe un código en R que represente los siguientes acontecimientos:

1. Llega Esteban.

In [ ]:

2. Atienden a Alan.

In [ ]:

3. Belén deja que Fátima se meta a la fila justo frente a ella.

In [ ]:

4. Esteban se tuvo que ir porque olvidó un documento.

In [ ]:

5. Cristóbal se tuvo que ir porque tiene clase.

In [ ]:

6. Sin asumir que se sabe la posición de Esteban y Cristóbal (utiliza funciones especiales). Encuentra la posición en la fila de Belén.

In [ ]:

Ejercicio A18: Suponiendo que vec es un vector con valores positivos de longitud 2 que representa las coordenadas de un punto en R2. Usa R para expresarlo en coordenadas polares. Hint: Necesitarás utilizar al menos una de las siguientes funciones (todas son las funciones trigonométricas inversas): `acos(x)`, `asin(x)`, o `atan(x)`.

In [ ]:

```
```

Matrices

Dataframes

Ejercicio C1: Crea un dataframe vacío.

In [40]:

```
data.frame()
```

Ejercicio C2: Un dataframe con 4 vectores previamente dados.

In [60]:

```
v1 <- c(4.5,5,3,2,7)
v2 <- factor(c("Si","No","Si","Si","No"))
v3 <- c("Ana","Beto","Carolina","Daniel","Erika")
v4 <- c(TRUE,TRUE,FALSE,TRUE,FALSE)
EC2 <- data.frame(v1,v2,v3,v4)
colnames(EC2) <- c("numero","SiNo","nombre","logico")
EC2
```

numero	SiNo	nombre	logico
4.5	Si	Ana	TRUE
5.0	No	Beto	TRUE
3.0	Si	Carolina	FALSE
2.0	Si	Daniel	TRUE
7.0	No	Erika	FALSE

Ejercicio C3: Obtener la estructura del dataframe 2.

In [61]:

```
str(EC2)
```

```
'data.frame':   5 obs. of  4 variables:
 $ numero: num  4.5 5 3 2 7
 $ SiNo  : Factor w/ 2 levels "No","Si": 2 1 2 2 1
 $ nombre: Factor w/ 5 levels "Ana","Beto","Carolina",..: 1 2 3 4 5
 $ logico: logi  TRUE TRUE FALSE TRUE FALSE
```

Ejercicio C4: Obtén el resumen estadístico y naturaleza de los datos del dataframe del ej. C2.

In [62]:

```
summary(EC2)
```

```
     numero    SiNo        nombre    logico
 Min.   :2.0   No:2   Ana     :1   Mode :logical
 1st Qu.:3.0   Si:3   Beto    :1   FALSE:2
 Median :4.5          Carolina:1   TRUE :3
 Mean   :4.3          Daniel  :1
 3rd Qu.:5.0          Erika   :1
 Max.   :7.0
```

Ejercicio C5: Extrae una columna específica del dataframe del ej. C2 usando su nombre de columna.

In [63]:
```
EC2$numero
```

4.5 5 3 2 7

Ejercicio C6: Extrae las primeras 2 filas del dataframe C2.

In [64]:
```
EC2[1:2,]
```

numero	SiNo	nombre	logico
4.5	Si	Ana	TRUE
5.0	No	Beto	TRUE

Ejercicio C7: Extrae la 3er y 5ta fila de la 1er y 3er columna del dataframe 2.

In [66]:
```
EC2[c(3,5),c(1,3)]
EC2
```

	numero	nombre
3	3	Carolina
5	7	Erika

numero	SiNo	nombre	logico
4.5	Si	Ana	TRUE
5.0	No	Beto	TRUE
3.0	Si	Carolina	FALSE
2.0	Si	Daniel	TRUE
7.0	No	Erika	FALSE

Ejercicio C8: Añade una quinta columna al dataframe 2.

In [67]:
```
numero2 <- c(6,3,5,2,4)
EC2 <- cbind(EC2,numero2)
EC2
```

numero	SiNo	nombre	logico	numero2
4.5	Si	Ana	TRUE	6
5.0	No	Beto	TRUE	3
3.0	Si	Carolina	FALSE	5
2.0	Si	Daniel	TRUE	2
7.0	No	Erika	FALSE	4

Ejercicio C9: Añáde más filas al dataframe 2.

In [69]:
```
datosextra <- data.frame(8.4,factor("Si"),"Federico",FALSE,4)
names(datosextra) <- c("numero","SiNo","nombre","logico","numero2")
EC2 <- rbind(EC2,datosextra)
EC2
```

numero	SiNo	nombre	logico	numero2
4.5	Si	Ana	TRUE	6
5.0	No	Beto	TRUE	3
3.0	Si	Carolina	FALSE	5
2.0	Si	Daniel	TRUE	2
7.0	No	Erika	FALSE	4
8.4	Si	Federico	FALSE	4

In [70]:
```
numero <- c(23,45,23,34,56,75)
EC2 <- cbind(EC2,numero)
```

In [73]:
```
EC2
```

numero	SiNo	nombre	logico	numero2	numero
4.5	Si	Ana	TRUE	6	23
5.0	No	Beto	TRUE	3	45
3.0	Si	Carolina	FALSE	5	23
2.0	Si	Daniel	TRUE	2	34
7.0	No	Erika	FALSE	4	56
8.4	Si	Federico	FALSE	4	75

In [ ]:
```
datos <- read.csv("Archivo.csv")
```

Ejercicio C10: Crea el siguiente data frame. Después, invierta `Sex` para todos los individuos.

	Age	Height	Weight	Sex
Alex	25	177	57	F
Lilly	31	163	69	F
Mark	23	190	83	M
Oliver	52	179	75	M
Martha	76	163	70	F
Lucas	49	183	83	M
Caroline	26	164	53	F

In [ ]:
```

```

Ejercicio C11: Crea el siguiente data frame.

	Working
Alex	Yes
Lilly	No
Mark	No
Oliver	Yes
Martha	Yes
Lucas	No
Caroline	Yes

In [ ]:
```

```

1. Añade este data frame por columna al data frame anterior.

In [ ]:

2. ¿Cuántas filas y cuántas columnas tiene el nuevo data frame?

In [ ]:

3. ¿Qué clase de información tiene cada columna?

In [ ]:

Ejercicio C12: Revisa qué tipo de datos tiene el objeto `state.center` y conviértelo a dataframe.

In [74]:
```
state.center
```

$x

-86.7509 -127.25 -111.625 -92.2992 -119.773 -105.513 -72.3573 -74.9841 -81.685 -83.3736
-126.25 -113.93 -89.3776 -86.0808 -93.3714 -98.1156 -84.7674 -92.2724 -68.9801 -76.6459
-71.58 -84.687 -94.6043 -89.8065 -92.5137 -109.32 -99.5898 -116.851 -71.3924 -74.2336
-105.942 -75.1449 -78.4686 -100.099 -82.5963 -97.1239 -120.068 -77.45 -71.1244 -80.5056
-99.7238 -86.456 -98.7857 -111.33 -72.545 -78.2005 -119.746 -80.6665 -89.9941 -107.256

$y

32.5901 49.25 34.2192 34.7336 36.5341 38.6777 41.5928 38.6777 27.8744 32.3329 31.75
43.5648 40.0495 40.0495 41.9358 38.4204 37.3915 30.6181 45.6226 39.2778 42.3645 43.1361
46.3943 32.6758 38.3347 46.823 41.3356 39.1063 43.3934 39.9637 34.4764 43.1361 35.4195
47.2517 40.221 35.5053 43.9078 40.9069 41.5928 33.619 44.3365 35.6767 31.3897 39.1063
44.2508 37.563 47.4231 38.4204 44.5937 43.0504

In [75]:
```
EC12 <- as.data.frame(state.center)
EC12
```

x	y
-86.7509	32.5901
-127.2500	49.2500
-111.6250	34.2192
-92.2992	34.7336
-119.7730	36.5341
-105.5130	38.6777
-72.3573	41.5928
-74.9841	38.6777
-81.6850	27.8744
-83.3736	32.3329
-126.2500	31.7500
-113.9300	43.5648
-89.3776	40.0495
-86.0808	40.0495
-93.3714	41.9358
-98.1156	38.4204
-84.7674	37.3915
-92.2724	30.6181
-68.9801	45.6226
-76.6459	39.2778
-71.5800	42.3645
-84.6870	43.1361
-94.6043	46.3943
-89.8065	32.6758
-92.5137	38.3347
-109.3200	46.8230
-99.5898	41.3356
-116.8510	39.1063
-71.3924	43.3934
-74.2336	39.9637
-105.9420	34.4764
-75.1449	43.1361
-78.4686	35.4195
-100.0990	47.2517
-82.5963	40.2210
-97.1239	35.5053
-120.0680	43.9078
-77.4500	40.9069
-71.1244	41.5928
-80.5056	33.6190

x	y
-99.7238	44.3365
-86.4560	35.6767
-98.7857	31.3897
-111.3300	39.1063
-72.5450	44.2508
-78.2005	37.5630
-119.7460	47.4231
-80.6665	38.4204
-89.9941	44.5937
-107.2560	43.0504

Ejercicio C13: Crea un data frame sencillo a partir de tres vectores. Después, ordénalo completamente por la primera columna.

In [ ]:

In [ ]:

In [ ]:

Ejercicio C15: Para este ejercicio utilizaremos el conjunto de datos predefinido llamado `VADeaths` . Asegúrate de que el objeto sea un dataframe. Si no lo es, cámbialo a dataframe.

In [81]:
```
EC15 <- as.data.frame(VADeaths)
EC15
```

	Rural Male	Rural Female	Urban Male	Urban Female
50-54	11.7	8.7	15.4	8.4
55-59	18.1	11.7	24.3	13.6
60-64	26.9	20.3	37.0	19.3
65-69	41.0	30.9	54.6	35.1
70-74	66.0	54.3	71.1	50.0

1. Crea una nueva variable, llamada `Total` , que será la suma de cada fila.

In [85]:
```
Total <- rowSums(EC15)
Total
```

50-54	44.2
55-59	67.7
60-64	103.5
65-69	161.6
70-74	241.4

2. Cambia el orden de columnas para que el total sea la primera variable.

In [86]:
```
data.frame(Total,EC15)
```

	Total	Rural.Male	Rural.Female	Urban.Male	Urban.Female
50-54	44.2	11.7	8.7	15.4	8.4
55-59	67.7	18.1	11.7	24.3	13.6
60-64	103.5	26.9	20.3	37.0	19.3
65-69	161.6	41.0	30.9	54.6	35.1
70-74	241.4	66.0	54.3	71.1	50.0

Ejercicio C16: Para este ejercicio utilizaremos el conjunto de datos predefinido llamado `state.x77` . Asegúrate de que el objeto sea un dataframe. Si no lo es, cámbialo a dataframe.

In [ ]:
```

```

1. Averigua cuántos estados tienen un ingreso menor a 4300.

In [ ]:
```

```

2. Averigua cuál es el estado con mayor ingreso

In [ ]:
```

```