

1. Asigna a una variable  $x$  el valor de 17. Posteriormente, crea un vector  $y$  con los valores [2, 4, 6, 10, 100]. Multiplica esos vectores por componente y guarda el resultado en un objeto  $z$ . Calcula la suma de todos los elementos en  $z$ .

```
# Creamos la variable x:
x <- 17

# Creamos el vector y:
y <- c(seq(2,6,2),10,100)

# Multiplicamos esos vectores y creamos el vector z:
z <- x*y

# Calculamos la suma de z:
sum(z)
```

```
## [1] 2074
```

2. Define dos vectores con los siguientes datos:  $s$  incluye los strings “lun”, “mar”, “mier”, “jueves”, “viernes” y “sabado”. El vector  $n$  incluye los valores [90, 70, 30, 50, 5, 10]. Une estos dos vectores de manera columnas en una matriz con 5 renglones y 2 columnas y guárdalo en un nuevo objeto llamado  $datos_{sem}$ .

```
# Definimos el vector s:
s <- c("lun", "mar", "mier", "jueves", "viernes", "sabado")

# Definamos el vector n:
n <- c(90, 70, 30, 50, 5, 10)

# Creamos la matrix de tamaño 6x2:
datos_sem <- matrix(c(s,n),nrow = 6)
```

3. Crea la siguiente data frame

Calcula el máximo y el mínimo en la columna de edad. Al parecer, hubo algunos problemas en la transcripción de la información. Genera una variable que contenga los resultados de la verificación lógica de edad debajo de 20 y arriba de 80. Usa esta variable para poner el valor de NA en las observaciones correspondientes. Crear el índice de masa corporal (IMC)  $IMC = \text{Peso en kg} / \text{Altura en metros}$ . Guarda los resultados en la variable BMI y agrégala a la dataframe. Redondea los valores obtenidos.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
# Creamos el dataframe:
edad <- c(21, 35, 829, 2)
sexo <- c("m", "f", "m", "e")
altura <- c(181, 173, 171, 166)
peso <- c(89, 58, 75, 60)

df_ejer3 <- data.frame(edad=edad, sexo=sexo, altura=altura, peso=peso)
```

```

# máximo y mínimo de la columna edad:
max(df_ejer3$edad)

## [1] 829

min(df_ejer3$edad)

## [1] 2

# verificación de la edad:
df_ejer3 <- df_ejer3 %>%
  mutate(veri_edad=ifelse(edad<=20|edad>=80, 1, 0))

# Agregamos las NaN en donde la edad esta fuera de rango:
df_ejer3 <- df_ejer3 %>%
  mutate(edad=ifelse(veri_edad==1, NaN, edad))

# Creamos la variable BMI= índice de masa corporal:
df_ejer3 <- df_ejer3 %>%
  mutate(BMI = round(peso/(altura/100),1))

```

4. Genera una secuencia de  $-5$  a  $5$  en incrementos de  $0.01$ . Grafique la función  $Y = x^2$  donde  $X$  es la secuencia previamente generada. Compara la función a:  $Y = -2 + x^2$ ,  $y = 5x^2$ ?

```

library(ggplot2)

# Creamos la secuencia:
x <- seq(-5, 5, 0.01)

# Creamos un dataframe
graficas <- data.frame(x=x)
# Creamos las 3 funciones:
graficas <- graficas %>%
  mutate(y_1 = x**2,
         y_2 = -2+x**2,
         y_3 = 5*(x**2))

```