



R para Finanzas

Guía de Ejercicios 1

Vectores, matrices, listas y data frames

Profesor: Víctor Macías E.

Pregunta 1

(a) Genere un vector que contenga la siguiente secuencia de números: 1, 2, 3, ..., 10

```
seq(1,10,1) #forma 1  
1:10 #forma 2  
seq(10) #forma 3  
c(1,2,3,4,5,6,7,8,9,10) #forma 4
```

(b) Asigne el vector creado a x , determine su tipo y dimensión.

```
x <- seq(1, 10, 1)  
  
typeof(x)  
length(x)
```

(c) Calcule la suma y el producto de los elementos del vector x

```
sum(x)  
prod(x)
```

(d) Calcule la suma acumulada y el producto acumulado de los elementos del vector x

```
cumsum(x)  
cumprod(x)
```

(e) Construya un vector lógico a partir de los elementos del vector x que son divisibles por 2 ¿Cuántos números son divisibles por 2? ¿Cuáles son divisibles por 2?

```
y <- x %% 2 == 0  
  
sum(x %% 2 == 0)  
  
x[x %% 2 == 0]
```

(f) Genere el siguiente vector: “mes_1”, “mes_2”, ..., “mes_10”

```
paste0("mes_", x)
```

Pregunta 2

Genere un vector que repita *roja* cuatro veces y *azul* tres veces.

```
c(rep("roja", 4), rep("azul", 3)) # Forma 1  
rep(c("roja", "azul"), c(4, 3))  # Forma 2
```

Pregunta 3

La siguiente tabla contiene las edades de 5 estudiantes:

Nombre	Edad
Pedro	17
Ana	NA
Maya	23
Max	NA
Paula	20

- (a) A partir de esta tabla genere un vector x con las edades de los estudiantes. Los nombres de cada uno de los elementos del vector deben ser los nombres de los estudiantes.

```
x = c("Pedro"=17, "Ana"=NA, "Maya"=23, "Max"=NA, "Paula"=20)
```

- (b) Defina un vector y que excluya los missing values (NA).

```
y = x[-c(2,4)] #forma 1  
y
```

```
y=x[c(-2,-4)] # forma 2  
y
```

```
y=x[c("Pedro", "Maya", "Paula")] # forma 3  
y
```

```
y=x[!is.na(x)] # forma 4  
y
```

```
y = na.omit(x) # forma 5  
y
```

- (c) Asigne las edades de Pedro y Ana a un vector z

```
z <- x[c("Pedro", "Ana")] #forma 1  
z
```

```
z <- x[c(1,2)] #forma 2  
z
```

- (d) Calcule la media aritmética de las edades de los 5 estudiantes usando la función `mean` y la fórmula para el cálculo de la media aritmética:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

```
mean(x) # ¿Qué obtiene?
```

```
mean(x, na.rm = TRUE) #forma 1
```

```
sum(x[!is.na(x)])/length(x[!is.na(x)]) #forma 2
```

- (e) Calcule la desviación estándar de las edades de los 5 estudiantes usando la función `sd` y la fórmula para el cálculo de la desviación estándar:

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

```
sd(x, na.rm = TRUE) # Forma 1
```

```
sqrt(sum((x[!is.na(x)]-mean(x, na.rm = TRUE))^2)/(length(x[!is.na(x)]) - 1)) # Forma 2
```

```
sqrt(sum((na.omit(x)-mean(x, na.rm = TRUE))^2)/(length(na.omit(x)) - 1)) # Forma 3
```

- (f) Genere un vector que incluya sólo los estudiantes que tienen una edad menor o igual a 20 años

```
x[x<=20 & !is.na(x)] # Forma 1
```

```
x[which(x<=20)] # Forma 2
```

- (g) Calcule la proporción de estudiantes que tienen 20 años o menos

```
mean(x<=20, na.rm = TRUE)
```

- (h) Calcule el promedio de edad de los estudiantes cuyas edades son 20 años o menos

```
mean(x[x<=20], na.rm = TRUE) #Forma 1
```

```
mean(x[which(x<=20)]) # Forma 2
```

- (i) Escriba `summary(x, na.rm = TRUE)` en el script ¿Qué obtiene?

```
summary(x, na.rm = TRUE)
```

Pregunta 4

A partir de los siguientes vectores, responda las siguientes preguntas:

```
x1 = c(10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5)
```

```
y1 = c(8.04, 6.95, 7.58, 8.81, 8.33, 9.96, 7.24, 4.26, 10.84, 4.82, 5.68)
```

- (a) Ordene los elementos del vector en orden creciente

```
sort(x1)      # Forma 1
```

```
x1[order(x1)]  # Forma 2
```

(b) Ordene los elementos del vector en orden decreciente

```
sort(x1, decreasing = TRUE)  # Forma 1
```

```
x1[order(x1, decreasing = TRUE)]  # Forma 2
```

(c) Calcule las medias aritméticas de x_1 y y_1

```
mean(x1)
```

```
mean(y1)
```

(d) Calcule las desviaciones estándar de x_1 y y_1 , usando la siguiente fórmula:

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

```
sd_x1 = sqrt(sum((x1-mean(x1))**2)/(length(x1) - 1))
```

```
sd_y1 = sqrt(sum((y1-mean(y1))**2)/(length(y1) - 1))
```

(e) Calcule la covarianza entre x_1 y y_1 , usando la siguiente fórmula:

$$s_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n - 1}$$

```
s_x1y1 <- sum((x1-mean(x1))*(y1-mean(y1)))/(length(x1)-1) # Forma 1  
s_x1y1
```

```
cov(x1, y1) # Forma 2
```

(f) Calcule el coeficiente de correlación entre x_1 y y_1 , usando la siguiente fórmula:

$$r_{xy} = \frac{s_{xy}}{s_x s_y}$$

```
s_x1y1 / (sd_x1*sd_y1)  # Forma 1
```

```
cor(x1, y1)  # Forma 2
```

(g) Calcule la mediana de x_1

```
median(x1) # Forma 1
```

```
quantile(x1, 0.5)  # Forma 2
```

Pregunta 5

Calcule la tasa de crecimiento promedio anual del PIB si las tasas son: 20% y -20%

```
r_g <- c(20, -20)
```

```
(prod(1 + r_g/100)^(1/length(r_g)) - 1)*100
```

Pregunta 6

Si una variable crece a un $x\%$ por período, construya una fórmula que le permita obtener el número de períodos para que la variable duplique su valor. Calcule el número de períodos si las tasas de crecimiento son 4%, 6%, 8% y 10%.

```
x = c(4, 6, 8, 10)
log(2)/log(1+(x/100))
```

Pregunta 7

- (a) Calcule el valor presente de un flujo de efectivo de \$1000 por un total de diez períodos. Asuma una tasa de descuento de 10%

El valor presente (VP) es igual a:

$$VP = \sum_{t=0}^T \frac{F_t}{(1+r)^t}$$

```
r = 0.1 #tasa de descuento
flujo_efectivo = 1000 #flujo por período
n=10
periodo = 1:n #períodos
```

```
valor_presente <- sum(flujo_efectivo * (1 + r) ^ -periodo)
```

```
paste0("Valor Presente", "=", "$", round(valor_presente,0)) #forma 1
```

```
paste0("Valor Presente", "=", "$", round((flujo_efectivo/r)*(1 - (1 + r) ^ -n),0)) #forma 2
```

- (b) Calcule el valor presente neto de un flujo de efectivo de \$300, \$1200, \$1000 y una inversión inicial de \$900. Asuma que las tasas de descuento para cada uno de los períodos son 10%, 20%, 15%, respectivamente.

```
# Tasa de interés
r <- c(0.1, 0.2, 0.15)

# Factores de descuento
yearly_discount_factors <- (1 + r) ^ - 1
yearly_discount_factors

discount_factors <- c(1, cumprod(yearly_discount_factors))
discount_factors
```

```
# Definir vector de flujos
cash_flow <- c(-900, 300, 1200, 1000)

# Calcular valor presente
VPN <- sum(cash_flow * discount_factors)
paste0("Valor Presente Neto", "=", "$", round(VPN,0))
```

- (c) Calcule el número de períodos en que se recupera la inversión (*payback period*) del flujo de efectivo de \$300, \$1200, \$1000 y una inversión inicial de \$900.

```
paste0("Payback period = ", min(which(cumsum(cash_flow) >= 0)) - 1)
```

Pregunta 8

Genere una muestra de 10000 números aleatorios provenientes de la distribución normal estándar y calcule la *skewness* y *kurtosis*, usando las siguientes fórmulas:

$$skewness = \frac{\frac{\sum_{i=1}^n (x_i - \bar{x})^3}{n}}{\left[\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n} \right]^{\frac{3}{2}}}$$

$$kurtosis = \frac{\frac{\sum_{i=1}^n (x_i - \bar{x})^4}{n}}{\left[\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n} \right]^2}$$

```
z = rnorm(10000)

skewness <- (sum((z - mean(z))^3)/length(z))/(sum((z - mean(z))^2)/length(z))^1.5
skewness

kurtosis <- (sum((z - mean(z))^4)/length(z))/(sum((z - mean(z))^2)/length(z))^2
kurtosis
```

Pregunta 9

Sea X una matriz :

$$\mathbf{X} = \begin{bmatrix} 0 & 10 & -5 \\ 5 & 8 & -4 \\ 0 & 1 & -1 \end{bmatrix}$$

- (a) Construya la matriz X y asigne nombres a sus filas y columnas

```
a <- c(0,10,-5)
b <- c(5,8,-4)
c <- c(0,1,-1)
```

```
X <- rbind(a,b,c)      # Forma 1
rownames(X) <- c("fila1", "fila2", "fila3")
colnames(X) <- c("columna1", "columna2", "columna3")
X
```

```
cbind(c(0,5,0), c(10,8,1),c(-5,-4,-1))  # Forma 2
```

```
e = c(0,10,-5, 5,8,-4, 0,1,-1)
matrix(e, nrow=3, ncol=3, byrow = TRUE)  # Forma 3
```

(b) Almacene la segunda fila de X en el vector z

```
z <- X[2,]
z
```

(c) Almacene los elementos (2,3) y (3,3) en el vector w

```
w <- X[c(2,3),3]
w
```

Pregunta 10

(a) Genere una lista que consista del vector a igual a $\{8, 12, 16, \dots, 36\}$, un vector b que contenga las palabras *hola* y *¿Cómo estás?* y un elemento c que contiene el número 5.

```
lista_A = list(a=seq(8, 36, 4), b=list("hola","¿Cómo estás?"), c = 5)
lista_A
```

(b) Obtenga el primer elemento de la lista

```
lista_A[["a"]] # Forma 1
```

```
lista_A$a      # Forma 2
```

```
lista_A["a"]   # Forma 3
```

(c) Obtenga los primeros dos elementos de la lista

```
lista_A[1:2]   # Forma 1
```

(d) Asigne *¿Cómo estás?* a s

```
lista_A[["b"]][2] # Forma 1
```

```
lista_A[["b"]][[2]] # Forma 2
```

Pregunta 11

Genere el siguiente *tibble*

```
tb <- tibble::tibble(x1 = c(10, 8, 13, 9, 11),
                    x2 = 5,
                    x3 = x1 + x2,
                    x4 = list(1:3, 3:4, 1:5, 2:7, 3))
```

(a) ¿Qué tipo de datos incluye cada columna del tibble?

```
dplyr::glimpse(tb)
```

(b) Extraer la variable x_3 del tibble

```
tb$x3      # Forma 1
```

```
tb[["x3"]]
```

```
tb[[3]]    # Forma 3
```

Pregunta 12

La siguiente tabla presenta la distribución por zona de una muestra, separando entre hombres y mujeres:

zona	hombres	mujeres
A	800	700
B	200	300
C	700	600
D	300	400

(a) Construya dicha tabla como un data frame

```
zona <- c("A", "B", "C", "D")
hombres <- c(800, 200, 700, 300)
mujeres <- c(700, 300, 600, 400)

muestra_total_df <- data.frame(zona, hombres, mujeres)      # dataframe
```

(b) Construya dicha tabla como un tibble

```
muestra_total_tb <- tibble::tibble(zona = c("A", "B", "C", "D"),
                                   hombres = c(800, 200, 700, 300),
                                   mujeres = c(700, 300, 600, 400)
                                   )
```

(c) Escriba en el script `summary(muestra_total_tb)` ¿Qué obtiene?

```
summary(muestra_total_tb)
```

(d) Escriba en el script `str(muestra_total_tb)` y `dplyr::glimpse(muestra_total_tb)` ¿Qué obtiene?

```
str(muestra_total_tb)
dplyr::glimpse(muestra_total_tb)
```

(e) Presente las primeras tres y últimas dos observaciones del `muestra_total_tb`


```
library(dplyr)
```

```
head(muestra_total_tb, n=3)
```

```
tail(muestra_total_tb, n=2)
```

```
muestra_total_tb %>% slice_head(n = 3)
```

```
muestra_total_tb %>% slice_tail(n = 2)
```

(f) Extraer la columna *mujeres*

```
muestra_total_tb %>% pull(mujeres)
```

(g) Construya una tabla con las observaciones de las variables *zona* y *mujeres*

```
muestra_total_tb %>% select(-hombres) # forma 1
```

```
muestra_total_tb %>% select(zona, mujeres) # forma 2
```

(h) Construya una tabla con las muestras de las zonas *A* y *C*

```
muestra_total_tb %>% filter(zona=="A" | zona=="C")
```

(i) Construya una variable *muestra_total* que sea igual a la suma de *hombres* y *mujeres*

```
muestra_total_tb <- muestra_total_tb %>% mutate(muestra_total = hombres + mujeres)
```

(j) Genere una variable que toma el valor “alto” si la muestra total de una zona determinada es mayor a 1000 e igual a “bajo”, si no son mayores.

```
muestra_total_tb <- muestra_total_tb %>%  
  mutate(mayor = if_else(muestra_total > 1000, "alto", "bajo"))
```

(k) Guarde *muestra_total_tb* con extensión *.csv*

```
library(readr)
```

```
write_csv(muestra_total_tb, file = "muestra_total.csv")
```