



R para Finanzas

Guía de Ejercicios 2

Vectores, matrices, listas y data frames/tibbles

Profesor: *Víctor Macías E.*

Pregunta 1

En la última pregunta de la Guía 1 se presentó el siguiente caso:

“Una persona ahorra \$ 10000. Si la tasa de interés mensual es 1%, ¿Cuánto dinero recibe después de 2 meses?”

A partir de esta pregunta, formularemos tres adicionales:

- (a) Si el monto depositado fuera \$10000, \$15000, \$18000, \$25000 ¿Cuál sería el monto obtenido después de dos meses?

```
S = c(10000,15000,18000,25000) # Vector con monto de dinero depositado
T = 2                          # Número de meses
r = 1                          # Tasa de interés mensual
```

```
M = S*(1+r/100)**T
```

El monto a recibir es:

```
M
```

En este caso, el monto a recibir (M) se calculó para cada uno de los montos depositados incluidos en el vector S , asumiendo que $T = 2$ y $r = 1$.

- (b) Si el número de meses fuera 2, 3, 4, ..., 10 ¿Cuál sería el monto obtenido?

```
S = 10000                                # Monto de dinero depositado
T = seq(from = 2, to = 10, by = 1)        # Vector con número de meses
r = 1                                     # Tasa de interés mensual
```

```
M = S*(1+r/100)**T
```

El monto a recibir es:

```
M
```

En este caso, el monto a recibir (M) se calculó para 2,3,4,5,6,7,8,9 y 10 meses incluidos en el vector T , asumiendo que $S = 10000$ y $r = 1$.

(c) Si la tasa de interés fuera 1%, 2%, 3%, 4%, 5% ¿Cuál sería el monto obtenido?

```
S = 10000                                # Monto de dinero depositado
T = 2                                     # Número de meses
# r = seq(from = 1, to = 5, by = 1)      # Vector de tasa de interés mensual
```

```
M = S*(1+r/100)**T
```

El monto a recibir es:

```
M
```

En este caso, el monto a recibir (M) se calculó para las tasas de interés mensuales iguales a 1%, 2%, 3%, 4%, 5% incluidos en el vector r , asumiendo que $S = 10000$ y $T = 2$.

Nota: Escriba en la console ?seq ¿Qué obtiene?

Los argumentos que incluye seq son los siguientes:

```
seq(from = 1, to = 1, by = ((to - from)/(length.out - 1)), length.out = NULL,
    along.with = NULL, ...)
```

Como se muestra a continuación cuando asigna el vector de tasas de interés a r en (c), no necesita explicitar `from`, `to` y `by`, porque R entiende que el primer argumento corresponde a `from`, el segundo, a `to` y el tercer argumento a `by`.

```
S = 10000                                # Monto de dinero depositado
T = 2                                     # Número de meses
r = seq(1, 5, 1)                         # Vector de tasa de interés mensual
```

```
M = S*(1+r/100)**T
```

Pregunta 2

- (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 y determine su tipo y dimensión.

```
x <- seq(1, 10, 1) # asignándolo a x
```

Nota: En R se puede usar $x <- \text{seq}(1, 10, 1)$ o $x = \text{seq}(1, 10, 1)$ para asignar el vector a x .

```
typeof(x)      # tipo
length(x)      # número de elementos de x
```

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

```
sum(x)      # suma de elementos de x
prod(x)     # producto de elementos de x
```

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

```
cumsum(x)    # suma acumulada de los elementos de x
cumprod(x)   # producto acumulado de los elementos de 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 # vector lógico
y
```

```
sum(x %% 2 == 0) # número de elementos de x que son divisibles por 2
```

```
x[x %% 2 == 0]  # elementos de x que son divisibles por 2
```

Pregunta 3

El retorno de una acción en los períodos 1 y 2 son 20% y -20%, respectivamente. Calcule el retorno promedio.

R : Sea \bar{r} el retorno promedio y r_1 y r_2 los retornos en los períodos 1 y 2, respectivamente. Por lo tanto, se tiene:

$$(1 + \bar{r})^2 = (1 + r_1)(1 + r_2) \Rightarrow (1 + \bar{r}) = \sqrt{(1 + r_1)(1 + r_2)} \Rightarrow \bar{r} = \sqrt{(1 + r_1)(1 + r_2)} - 1$$

```
r_g <- c(20,-20) # vector de retornos de períodos 1 y 2
```

A continuación se presenta la fórmula para n períodos:

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

Pregunta 4

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%.

R: El problema a resolver es el valor de t que resuelve:

$$2A_0 = A_0(1 + g)^t \Rightarrow 2 = (1 + g)^t$$

Aplicando logaritmos a ambos lados, se obtiene:

$$\log 2 = t \log(1 + g) \Rightarrow t = \frac{\log 2}{\log(1 + g)}$$

```
g = c(4, 6, 8, 10) # vector de tasas de crecimiento
```

```
log(2)/log(1+(g/100)) # número de períodos para que variable duplique sus valor
```

Pregunta 5

- (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:

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

Dado que $F_t = F$, $\forall t$, se tiene:

$$VP = \sum_{t=1}^T \frac{F}{(1 + r)^t} = \frac{F}{r} \times \left[1 - \frac{1}{(1 + r)^T} \right]$$

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

```
periodo
```

```
flujo_efectivo * (1 + r) ^ -periodo
```

```
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.

El valor presente neto es igual a:

$$VPN = -I_0 + \frac{F_1}{1 + r_1} + \frac{F_2}{(1 + r_1)(1 + r_2)} + \frac{F_3}{(1 + r_1)(1 + r_2)(1 + r_3)}$$

Reemplazando los valores respectivos, se tiene:

$$VPN = -900 + \frac{300}{1.1} + \frac{1200}{1.1 \times 1.2} + \frac{1000}{1.1 \times 1.2 \times 1.15} \approx 940.58$$

```
# 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,2))
```

- (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 6

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 7

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 *edad* con las edades de los estudiantes. Los nombres de cada uno de los elementos del vector deben ser los nombres de los estudiantes.

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

- (b) Defina un vector *y* que excluya los missing values (*NA*). Los datos faltantes se representan por *NA* que significa *Not Available* (No disponible).

```
y=edad[!is.na(edad)]
y
```

Si no entiende esta sintaxis, se recomienda realizar los siguientes pasos:

```
is.na(edad)      # ¿Qué obtiene?
!is.na(edad)     # ¿Qué obtiene?
```

```
edad[!is.na(edad)] # ¿Qué obtiene?
```

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

```
z <- edad[c("Pedro", "Ana")] # Forma 1
z
```

```
z <- edad[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(edad) # ¿Qué obtiene?
```

Nota: Escriba en la console ¿mean ¿Qué obtiene?

Si no se explicita un valor **TRUE** o **FALSE** para **na.rm**, tomará el valor por defecto que es **FALSE**. En la ayuda se señala que **na.rm** es “a logical value indicating whether NA values should be stripped before the computation proceeds”.

```
mean(edad, na.rm = TRUE) # Forma 1
```

```
sum(edad[!is.na(edad)])/length(edad[!is.na(edad)]) # 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(edad, na.rm = TRUE)
```

¿Cómo calcularía la desviación estándar programando la fórmula?

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

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

Si no entiende esta sintaxis, se recomienda realizar los siguientes pasos:

```
edad <= 20 # ¿Qué obtiene?
mean(edad <= 20, na.rm = TRUE) # ¿Qué obtiene?
```

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

```
mean(edad[edad<=20], na.rm = TRUE)
```

Si no entiende esta sintaxis, se recomienda realizar los siguientes pasos:

```
edad <= 20 # ¿Qué obtiene?
edad[edad<=20] # ¿Qué obtiene?
mean(edad[edad<=20], na.rm = TRUE) # ¿Qué obtiene?
```

Pregunta 8

La siguiente tabla que muestra la distribución del tenure, en años, de un grupo de trabajadores de una empresa:

Tenure	Frecuencia
2	5
3	6
4	9
5	3

¿Cuál es la mediana del tenure de este grupo de trabajadores?

```
# Forma 1
tenure_1 <- c(2,2,2,2,2,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,5,5,5)
```

```
median(tenure_1)
```

```
quantile(tenure_1, probs = 0.5)
```

```
# Forma 2
tenure_2 <- c(rep(2,5), rep(3,6), rep(4,9), rep(5,3))
tenure_2
```

```
median(tenure_2)
```

```
quantile(tenure_2, probs = 0.5)
```

Pregunta 9

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

$$skewness = \frac{\frac{\sum_{i=1}^n (x_i - \bar{x})^3}{n}}{\left[\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n} \right]^{\frac{3}{2}}}$$
$$kurtosis = \frac{\frac{\sum_{i=1}^n (x_i - \bar{x})^4}{n}}{\left[\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n} \right]^2}$$

```
# Generando 10000 números (pseudo)aleatorios de una normal estándar
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
```

Podría haber llegado al mismo resultado usando el paquete *skewness*.

```
library(moments)
skewness(z)
kurtosis(z)
```

Pregunta 10

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 11

- (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]
```

- (d) Asigne *¿Cómo estás?* a s

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

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

Pregunta 12

La siguiente tabla presenta el stock por zona de los productos XX y ZZ de una empresa:

zona	XX	ZZ
A	800	700
B	200	300
C	700	600
D	300	400

- (a) Construya dicha tabla como un tibble

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

```
library(tibble)
```

```
stock_total <- tibble(zona, XX, ZZ)
```

```
stock_total
```

(b) Escriba en el script `summary(stock_total)` ¿Qué obtiene?

```
summary(stock_total)
```

(c) Escriba en el script `glimpse(stock_total)` ¿Qué obtiene?

```
library(dplyr)
```

```
glimpse(stock_total) # Forma 1
```

```
stock_total %>% glimpse() # Forma 2
```

Nota: El operador %>% se denomina pipe y su propósito es concatenar múltiples operaciones.

(d) Presente las primeras tres y últimas dos observaciones del `stock_total`

```
stock_total %>% slice_head(n = 3) # Primeras 3 observaciones
```

```
stock_total %>% slice_tail(n = 2) # Últimas 2 observaciones
```

(e) Construya una tabla con las observaciones de las variables `zona` y `ZZ`

```
stock_total %>% select(-XX) # Forma 1
```

```
stock_total %>% select(zona, ZZ) # Forma 2
```

(f) Construya una tabla con los stocks de las zonas *A* y *C*

```
stock_total %>% filter(zona=="A" | zona=="C") # Forma 1
```

```
stock_total %>% filter(zona %in% c("A","C")) # Forma 2
```

(g) Construya una variable `stock` que sea igual a la suma de `XX` y `ZZ` y una variable `prop_stock` que represente la proporción del stock de una determinada zona.

```
stock_total <-  
  stock_total %>%  
  mutate(stock = XX + ZZ,  
         prop_stock = (stock/sum(stock))*100)  
stock_total
```

- (h) Genere una variable que toma el valor “alto” si el stock en una zona determinada es mayor a 1000 e igual a “bajo”, si no es mayor.

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

- (i) Guarde stock_total con extensión .csv

```
readr::write_csv(stock_total, file = "datos/stock_total.csv")
```