

# Econometría I

## Guía de Ejercicios 1

### *Vectores, data frames y tibbles*

**Profesor:** Víctor Macías E.

### Pregunta 1

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

```
seq(1,10,1) #forma 1
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
1:10 #forma 2
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
seq(10) #forma 3
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

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

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

(b) Genere un vector del 1 al 5 que contenga 10 elementos

```
seq(1,5, length=10)
```

```
## [1] 1.000000 1.444444 1.888889 2.333333 2.777778 3.222222 3.666667 4.111111
```

```
## [9] 4.555556 5.000000
```

(c) Genere un vector que contenga los números: 10, 30, 50 y luego calcule su suma, suma acumulada y suma de los cuadrados de cada uno de los números.

```
z <- c(10, 30, 50)
```

```
names(z) <- c("A", "B", "C")
```

```
z
```

```
## A B C
```

```
## 10 30 50
```

```
sum(z) # Suma
```

```
## [1] 90
```

```
cumsum(z) # Suma acumulada
```

```
## A B C
```

```
## 10 40 90
```

```
sum(z**2) # Suma de los cuadrados
```

```
## [1] 3500
```

## Pregunta 2

El vector *vida\_util* contiene los años de vida útil restante de 5 máquinas, excepto para las máquinas *B* y *D*:

```
vida_util = c("A"=6, "B"=NA, "C"=10, "D"=NA, "E"=20)
```

(a) Defina un vector *y* que excluya los missing values (NA).

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

```
y
```

```
## A C E
```

```
## 6 10 20
```

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

```
y
```

```
## A C E
```

```
## 6 10 20
```

```
y=vida_util[!is.na(vida_util)] # forma 3
```

```
y
```

```
## A C E
```

```
## 6 10 20
```

```
y = na.omit(vida_util) # forma 4
```

```
y
```

```
## A C E
```

```
## 6 10 20
```

```
## attr("na.action")
```

```
## B D
```

```
## 2 4
```

```
## attr("class")
```

```
## [1] "omit"
```

(b) Calcule el promedio aritmético de la vida útil de las 5 máquinas

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

```
## [1] 12
```

```
mean(vida_util[!is.na(vida_util)]) #forma 2
```

```
## [1] 12
```

```
sum(vida_util[!is.na(vida_util)])/length(vida_util[!is.na(vida_util)]) #forma 3
```

```
## [1] 12
```

(c) Calcule la proporción de máquinas que tienen una vida útil restante de 10 años o menos

```
mean(vida_util<=10, na.rm = TRUE)
```

```
## [1] 0.6666667
```

(d) ¿Cuántas máquinas tienen una vida útil menor o igual a 10 años?

```
sum(vida_util<=10, na.rm = TRUE)
```

```
## [1] 2
```

## Pregunta 3

La siguiente tabla presenta el número de unidades en stock de los productos 1 y 2 en cuatro zonas:

zona	stock_1	stock_2
A	800	700
B	200	300
C	700	600
D	300	400

(a) Construya dicha tabla como un data frame/tibble y asigne el dataframe a `stock_total_df` y el tibble a `stock_total_tb`

```
zona <- c("A", "B", "C", "D")
stock_1 <- c(800,200,700,300)
stock_2 <- c(700,300,600,400)
```

```
stock_total_df <- data.frame(zona, stock_1, stock_2)    # dataframe
stock_total_df
```

```
##   zona stock_1 stock_2
## 1   A      800      700
## 2   B      200      300
## 3   C      700      600
## 4   D      300      400
```

```
# Tibble
```

```
library(tibble) # loading the package
```

```
stock_total_tb <- tibble(zona = c("A", "B", "C", "D"),
                        stock_1 = c(800,200,700,300),
                        stock_2 = c(700,300,600,400)
                        )
```

```
stock_total_tb
```

```
## # A tibble: 4 x 3
##   zona stock_1 stock_2
##   <chr>   <dbl>   <dbl>
## 1 A         800     700
## 2 B         200     300
## 3 C         700     600
## 4 D         300     400
```

```
#class(stock_total_df)
```

```
class(stock_total_tb)
```

```
## [1] "tbl_df"      "tbl"        "data.frame"
```

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

```
summary(stock_total_tb)
```

```
##      zona      stock_1      stock_2
## Length:4      Min.    :200      Min.    :300
## Class :character 1st Qu.:275      1st Qu.:375
## Mode  :character Median :500      Median :500
##              Mean  :500      Mean  :500
##              3rd Qu.:725      3rd Qu.:625
##              Max.   :800      Max.   :700
```

(c) Escriba en el script `str(stock_total_tb)` y `glimpse(stock_total_tb)` ¿Qué obtiene?

```
library(dplyr)
```

```
str(stock_total_tb) #forma 1
```

```
## tibble [4 x 3] (S3: tbl_df/tbl/data.frame)
## $ zona : chr [1:4] "A" "B" "C" "D"
## $ stock_1: num [1:4] 800 200 700 300
## $ stock_2: num [1:4] 700 300 600 400
```

```
glimpse(stock_total_tb) #forma 2 - parte de dplyr
```

```
## Rows: 4
## Columns: 3
## $ zona    <chr> "A", "B", "C", "D"
## $ stock_1 <dbl> 800, 200, 700, 300
## $ stock_2 <dbl> 700, 300, 600, 400
```

(d) Presente las primeras tres y últimas dos observaciones de `stock_total_tb`

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

```
## # A tibble: 3 x 3
##   zona stock_1 stock_2
##   <chr>   <dbl>   <dbl>
## 1 A       800     700
## 2 B       200     300
## 3 C       700     600
```

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

```
## # A tibble: 2 x 3
##   zona stock_1 stock_2
##   <chr>   <dbl>   <dbl>
## 1 C       700     600
## 2 D       300     400
```

(e) Construya un tibble con las observaciones de las variables `zona` y `stock_2`

```
stock_total_tb %>% select(-stock_1) # forma 1
```

```
## # A tibble: 4 x 2
##   zona stock_2
##   <chr>   <dbl>
## 1 A       700
## 2 B       300
## 3 C       600
```

```
## 4 D          400
stock_total_tb %>% select(zona, stock_2) # forma 2
```

```
## # A tibble: 4 x 2
##   zona stock_2
##   <chr>   <dbl>
## 1 A       700
## 2 B       300
## 3 C       600
## 4 D       400
```

(f) Construya un tibble con los stocks de las zonas A y C

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

```
## # A tibble: 2 x 3
##   zona stock_1 stock_2
##   <chr>   <dbl>   <dbl>
## 1 A       800     700
## 2 C       700     600
```

(g) Construya dos variables: (1) `stock_total` que sea igual a la suma de `stock_1` y `stock_2` y (2) `prop_stock` que es el porcentaje del stock total existente en una zona determinada.

```
stock_total_tb <- stock_total_tb %>%
  mutate(stock_total = stock_1 + stock_2,
         prop_stock = (stock_total/sum(stock_total))*100)
stock_total_tb
```

```
## # A tibble: 4 x 5
##   zona stock_1 stock_2 stock_total prop_stock
##   <chr>   <dbl>   <dbl>       <dbl>     <dbl>
## 1 A       800     700        1500      37.5
## 2 B       200     300         500      12.5
## 3 C       700     600        1300      32.5
## 4 D       300     400         700      17.5
```

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

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

```
## # A tibble: 4 x 6
##   zona stock_1 stock_2 stock_total prop_stock mayor
##   <chr>   <dbl>   <dbl>       <dbl>     <dbl> <chr>
## 1 A       800     700        1500      37.5 alto
## 2 B       200     300         500      12.5 bajo
## 3 C       700     600        1300      32.5 alto
## 4 D       300     400         700      17.5 bajo
```

(i) Guarde `stock_total_tb` con extensión `.csv`

```
library(readr) #loading the package
write_csv(stock_total_tb, path = "stock_total.csv")
```