

MA0501 – Tarea 1

Diego Alberto Vega Víquez

2025-08-12

Tabla de contenidos

Ejercicio 1	2
Ejercicio 2	2
Ejercicio 3	3
Ejercicio 4	3
Ejercicio 5	3
Ejercicio 6	4
Ejercicio 7	4
Ejercicio 8	4
Ejercicio 9	5
Ejercicio 10	5
Ejercicio 11	5
Ejercicio 12	5
Ejercicio 13	6
Ejercicio 14	6
Ejercicio 15	6
Ejercicio 16	6
Ejercicio 17	8
Ejercicio 18	9
Ejercicio 19	10

Ejercicio 1

Dado:

$$x = (3, -5, 31, -1, -9, 10, 0, 18) \quad y = (1, 1, -3, 1, -99, -10, 10, -7)$$

Realice:

1. Introduzca **x** y **y** como vectores en R.

```
x <- c(3, -5, 31, -1, -9, 10, 0, 18)
y <- c(1, 1, -3, 1, -99, -10, 10, -7)
```

2. Calcule media, varianza, raíz cuadrada y desviación estándar de **y**.
3. Calcule media, varianza, raíz cuadrada y desviación estándar de **x**.
4. Calcule la correlación entre **x** y **y**.
5. Comando en R para extraer entradas 2 a 7 de **x**.
6. Comando en R para extraer entradas de **y** excepto la 2 y la 7.
7. Comando en R para extraer entradas de **y** menores a -3 o mayores a 10.
8. Comando en R para extraer entradas de **x** mayores a 0 y pares.

Solución

Ejercicio 2

Introduzca en R la matriz 4×3 :

```
A = matrix(c(1,2,3,4,5,6,7,8,9,10,11,12), nrow=4, "byrow"="true")
```

Luego, obtenga algunos elementos de la matriz de la siguiente manera: $A[1,1:3]$, $A[1:4,2]$, $A[3,3]$, $A[11]$, $A[20]$, $A[5,4]$, $A[1,1,1]$ y explique qué pasa en cada caso.

Solución

Ejercicio 3

Investigue para qué sirven los comandos de R `as.matrix(...)` y `as.data.frame(...)`. Explique y dé un ejemplo de cada uno.

Solución

Ejercicio 4

Introduzca usando código R (no archivos) en un DataFrame la siguiente tabla de datos:

Peso	Edad	Nivel Educativo
76	25	Lic
67	23	Bach
55	19	Bach
57	18	Bach
87	57	Dr
48	13	MSc

Solución

Ejercicio 5

En muchas ocasiones nos interesa hacer referencia a determinadas partes o componentes de un vector.

Defina el vector:

$$x = (2, -5, 4, 6, -2, 8)$$

Luego, a partir de este vector defina instrucciones en R para generar los siguientes vectores:

- $y = (2, 4, 6, 8)$, así definido y es el vector formado por las componentes positivas de x .
- $z = (-5, -2)$, así definido z es el vector formado por las componentes negativas de x .
- $v = (-5, 4, 6, -2, 8)$, así definido v es el vector x eliminada la primera componente.

- $w = (2, 4, -2)$, así definido w es el vector x tomando las componentes con índice impares, es decir, $x[1] = 2$, $x[3] = 4$ y $x[5] = -2$.

Solución

Ejercicio 6

Cargue la tabla de datos que está en el archivo `SAheartv.csv` y realice:

1. Calcule la dimensión de la tabla de datos.
2. Despliegue las primeras 3 columnas de la tabla.
3. Ejecute `summary()` y `str()` de los datos.
4. Usando `cor()` de R calcule la correlación entre las variables `tobacco` y `alcohol`.
5. Calcule la suma de las columnas con variables cuantitativas (numéricas).
6. Calcule para todas las variables cuantitativas presentes en `SAheart.csv`: mínimo, máximo, media, mediana y para la variable `chd` determine la cantidad de **Sí** y **No**.

Solución

Ejercicio 7

Programa en R una función que genere 200 números al azar entre 1 y 500 y calcule cuántos están entre 50 y 450, ambos inclusive.

Solución

Ejercicio 8

Desarrolle una función que calcule el costo de una llamada telefónica que ha durado t minutos, sabiendo que:

- Si $t < 1$: costo = 0.4 USD.
- Si $t \geq 1$: costo = $0.4 + (t - 1)/4$ USD.

La función debe recibir el valor de t .

Solución

Ejercicio 9

Desarrolle una función que reciba una matriz cuadrada A de tamaño $n \times n$ y calcule su traza, es decir, la suma de los elementos de la diagonal.

Ejemplo:

$$\begin{pmatrix} 9 & 3 & 4 \\ 1 & 3 & -1 \\ 4 & 12 & -2 \end{pmatrix}$$

Traza = 10.

Solución

Ejercicio 10

Escriba una función que genere los n primeros términos de la serie de Fibonacci.

Solución

Ejercicio 11

Escriba una función que retorne el mayor número entero cuyo cuadrado no exceda de x , donde x es un número real que se recibe como parámetro, utilizando `while`.

Solución

Ejercicio 12

Cree un Data Frame con diez alumnos con su edad, año de nacimiento y número de teléfono. Deberá aparecer el nombre de la columna (`edad`, `año de nacimiento`, `teléfono`) y el nombre de la fila, que será el nombre del alumno.

Solución

Ejercicio 13

Desarrolle una función en **R** que reciba un **DataFrame** y retorne la cantidad de entradas que son divisibles entre 3.

Solución

Ejercicio 14

Desarrolle una función en **R** que reciba un **DataFrame** y dos números de columna y que retorne en una lista:

- El nombre de las variables correspondientes a las columnas.
- La covarianza entre dichas variables.
- La correlación entre dichas variables.

Solución

Ejercicio 15

Programa la siguiente función recursiva:

$$U(n) = \begin{cases} 5 & \text{si } n = 0 \\ -5 & \text{si } n = 1 \\ 2 & \text{si } n = 2 \\ 4U_{n-1} - 15U_{n-2} + U_{n-3} & \text{si } n \geq 3 \end{cases}$$

Ejercicio 16

El **Algoritmo de Luhn** es una fórmula de suma de verificación utilizada para validar una diversidad de números de identificación.

Por ejemplo, para validar si un número es o no un número de tarjeta de crédito válido.

Este algoritmo es muy simple: nos dice que, dado un número que contenga solamente dígitos $[0-9]$, una tarjeta de crédito es válida si y solo si, obteniendo la reversa de este número, la suma especial de sus dígitos debe ser un múltiplo de 10, es decir, que la suma módulo 10 debe ser igual a cero.

La suma especial se realiza de la siguiente forma:

1. Invertir el número.
2. Si la posición es impar, sumar el dígito tal cual.
3. Si la posición es par, multiplicar el dígito por 2 y sumar los dígitos del resultado (o restar 9 si es mayor o igual a 10).
4. Verificar que la suma total sea un múltiplo de 10.

Ejemplo 1

Número: *49927398716*

- a) Multiplicamos por 2 los dígitos que ocupan las posiciones pares empezando por el final (reversando el número):

$$(1 \times 2) = 2, \quad (8 \times 2) = 16, \quad (3 \times 2) = 6, \quad (2 \times 2) = 4, \quad (9 \times 2) = 18$$

4	9	9	2	7	3	9	8	7	1	6
4	18	9	4	7	6	9	16	7	2	6

- b) Sumamos los dígitos que ocupaban las posiciones impares con los dígitos de los productos obtenidos:

$$6 + (2) + 7 + (1 + 6) + 9 + (6) + 7 + (4) + 9 + (1 + 8) + 4 = 70$$

O equivalentemente, sumando y restando 9 en los productos mayores o iguales a 10:

$$6 + (2) + 7 + (16 - 9) + 9 + (6) + 7 + (4) + 9 + (18 - 9) + 4 = 70$$

- c) Si el resto de dividir 70 entre 10 es igual a 0, el número es válido:

$$70 \bmod 10 = 0$$

Ejemplo 2

Número de tarjeta: 4539319503436467

- a) Primero se toman todos los números en las posiciones pares empezando desde la derecha, es decir, reversando el número:

4_3_3_3_9_0_4_6_6_

- b) Luego se deben cambiar por el doble de ese número; si el número obtenido es mayor a 9, entonces se le resta 9 (o se suman los dígitos):

8_6_6_9_0_8_3_3_

- c) Finalmente se suman todos los dígitos:

$$8 + 5 + 6 + 9 + 6 + 1 + 9 + 5 + 0 + 3 + 8 + 3 + 3 + 4 + 3 + 7 = 80$$

Como el número obtenido es divisible entre 10, entonces es un número de tarjeta válido.

Programa en **R** las funciones necesarias para implementar el **Algoritmo de Luhn**.

Solución

Ejercicio 17

La **Factorización Prima** es el proceso de descomponer un número en sus factores primos.

El algoritmo consiste en dividir entre números primos hasta que el cociente sea 1.

Por ejemplo, para obtener los factores primos de 36 se hace:

$$36 \div 2 = 18; \quad 18 \div 2 = 9; \quad 9 \div 3 = 3; \quad 3 \div 3 = 1$$

Entonces los factores primos son: 2, 2, 3, 3 y la factorización prima de $36 = 2 \times 2 \times 3 \times 3$.

Programa en **R** una función recursiva que reciba un valor y retorne una lista con los factores primos de dicho valor.

Los factores primos de un número entero son los números primos divisores exactos de ese número entero.

El proceso de búsqueda de esos divisores se denomina factorización de enteros, o factorización en números primos.

Por ejemplo, para 60 el proceso es el siguiente:

- $60/2 = 30$
- $30/2 = 15$
- $15/3 = 5$
- $5/5 = 1$

Por tanto, el resultado es: **(2 2 3 5)**.

Solución

Ejercicio 18

La **Conjetura de Goldbach** dice que todo número par mayor a 2 se puede expresar como la suma de 2 números primos.

Por tanto, desarrolle un sistema que reciba un número (por teclado) y retorne una lista con los 2 números primos que componen la suma.

Se puede asumir que el número ingresado siempre es mayor a 2 y es par.

Debe programar en **R** lo siguiente:

- a) **es_primo**: determina si un valor es primo. Debe programarla con recursión.
- b) **valores**: obtiene en una lista los 2 números que conforman la suma. Debe programarla con recursión.
- c) **resultado**: para un valor dado despliega la lista con los números primos.

Solución

Ejercicio 19

Desarrolle una función en **R** que reciba un Data Frame y dos números de columna y que retorne en una lista:

- El nombre de las variables correspondientes a las columnas.
- La covarianza entre esas dos variables.
- La correlación entre esas dos variables.

Pruebe la función usando el archivo `EjemploEstudiantes.csv`.