

Tema 1: Introducción y preliminares. Ejercicios resueltos.

1. Crear un vector `x` con los datos

10, 11, 13, -1, 6, 3

Guardar el área de trabajo (workspace) con el nombre `ejercicio1.RData` y salir de \mathbb{R} . Volver a entrar y recuperar la imagen salvada y comprobar que tiene el vector `x`. Notar que `X` no tiene nada. ¿Qué se puede usar como nombre de un vector?

Ayuda: `x <- c(10, 11, 13, -1, 6, 3)`

2. Cálculo de estadísticos simples de `x`. Calcular la media, la desviación típica y la varianza. Crear un objeto (vector) con el nombre `est.x` en el que guardar los estadísticos más relevantes. Eliminar el vector `x` y el vector `est.x`.

Ayuda: `est.x <- c(mean(x), sd(x), var(x), median(x))`

3. ¿Cuál es la diferencia entre `median(x)` y `mean(x, trim=0.5)`? Comprobar el resultado con ejemplos.

Ayuda: `x <- c(1, 3, 2, 5, 8, 9, 8, 2, 50); median(x); mean(x, trim=0.5)`

4. Obtener información sobre la función `rnorm()`. ¿Qué hace exactamente? ¿Para qué puede servir de utilidad?

```
sim.n <- rnorm(n=1000, mean=5, sd=2)
      Simula 1000 valores de una normal de media 5 y varianza 4 y los asigna a
      sim.n
```

```
hist(sim.n)
      Hace un histograma del vector sim.n
```

Obtener la media y la varianza del vector `sim.n`. Investigar la función `summary()`. Utilizarla para calcular los estadísticos más básicos del vector `sim.n`.

Solución:

`rnorm(n, mu, sd)` simula `n` valores de una normal de media `mu` y desviación típica `sd`. Nos puede ayudar a observar como se comporta una variable con esa distribución (se tratará con mayor detalle en la asignatura de Probabilidad y Simulación).

`mean(sim.n)` y `sqrt(var(sim.n))` nos dan la media y la desviación típica del vector con las 1000 simulaciones, y como era de esperar nos salen valores cercanos a la media 5 y desviación típica 2 con la que hemos trabajado.

`summary()` calcula un vector con los estadísticos más habituales. Ver Sesión 3.

5. Comprobar si es cierto (TRUE) o falso (FALSE) que 2 es mayor que 3. Esto es lo que se conoce como un resultado *lógico*. Los objetos (como los vectores) formados por elementos lógicos se pueden utilizar para seleccionar datos cumpliendo una característica.

Crear un objeto x con los datos

$$-2, 1, -1, -1, 3, 6, 1$$

y crear un vector `posit` con el resultado de la operación `x>0`. ¿Qué significado tiene este vector? ¿Qué ocurre si hacemos `x[posit]`?

Solución:

2>3 produce un FALSE.

`posit <- x>0; length(x[posit])` `posit` es un vector lógico que indica que componentes de x son positivas.

`x[posit]` nos muestra las componentes de x que son positivas. Ver Sesión 2: vectores índices.

6. \mathbb{R} tiene grandes facilidades para generar secuencias de números. El operador más usual es `:` que genera una secuencia `desde:hasta` en incrementos (o decremento si `hasta` es menor que `desde`) de uno, por ejemplo:

`1:5`

genera los valores 1 2 3 4 5

¿Qué ocurre si hacemos `x[1:3]` cuando x tiene los datos 2, 1, 0, 3, 6, 1? Observar que este operador tiene la prioridad más alta en una expresión. Comprobarlo con la expresión `2*1:5`. ¿Puedes predecir el valor de la expresión `1:7*1:2`? ¿Puedes predecir el valor de la expresión `1:8*1:2`? Comprueba el resultado y trata de explicar que ha ocurrido.

Solución:

`x[1:3]` nos muestra las componentes 1, 2 y 3 del vector x .

El operador `:` tiene la prioridad más alta, por eso, al hacer `2*1:5`, el resultado es la multiplicación de 2 por el vector 1,2,3,4,5.

`1:7*1:2` equivale a (1 2 3 4 5 6 7)*(1 2) y como el primer vector tiene 7 valores y el vector 1,2 solo 2, se produce un aviso para indicarlo. El resultado es 1 4 3 8 5 12 7.

En el caso de `1:8*1:2` como el vector es par no se produce ningún aviso y el resultado es 1 4 3 8 5 12 7 16.