

## SOLUCIONARIO DEL CUESTIONARIO SOBRE OPERACIONES BÁSICAS DE R CLÁSICO

**Pregunta 1** → Si queremos darle a una variable "a" el valor "4", debemos escribir en la consola:

- a<-4
- a->4
- a=4
- 4->a
- 4=a

- Explicación teórica: El valor que se le da a la variable debe ir siempre a la derecha o, en caso de utilizarse flechas, esta debe señalar a la variable.

**Pregunta 2** → Si escribimos "variable 1=3", la consola R no nos dará error.

- Verdadero
- Falso

- Debería haberse escrito "Variable1" o "Variable\_1", ya que no se pueden dejar espacios en las variables.

**Pregunta 3** → Si escribimos 10\*runif(9), la consola hará lo siguiente:

- Verdadero
- Falso

- Debería haberse escrito "Variable1" o "Variable\_1", ya que no se pueden dejar espacios en las variables.

**Pregunta 4** → ¿Cómo se denomina este botón de la consola?

- Creará nueve valores aleatorios en el intervalo [0,10]
- Creará nueve valores aleatorios en el intervalo (0,10)
- Creará diez valores aleatorios en el intervalo (0,9)
- Creará diez valores aleatorios en el intervalo [0,9]

- El comando `runif(...)` sirve para crear valores aleatorios en el intervalo [0,1]. Se darán tantos valores como se introduzca entre el paréntesis. Si se coloca un número multiplicando al runif, el intervalo de valores se encontrará entre 0 y ese número.

**Pregunta 5** → Si escribimos "sample(1:50,7, replace=TRUE)", la consola hará lo siguiente:

- Seleccionar todo
- Remover todos los objetos
- Guardar script
- Correr línea o seleccionar

**Pregunta 6** → Si escribimos "seq(a,b,length=n)", la consola crea:

- Creará 7 números enteros en el intervalo [1,50], pudiendo repetirse valores.
- Creará 7 números reales en el intervalo (1,50), sin poder repetirse valores.
- Creará 7 números reales en el intervalo [1,50], pudiendo repetirse valores.
- Creará 7 números reales en el intervalo (1,50), sin poder repetirse valores.

- Ya hemos mencionado "runif", pero existen otras operaciones para generar números aleatorios. `Sample(a:b,M)` permite obtener M números enteros aleatorios en el intervalo [a, b] con o sin repetición. En el caso de que añadamos la opción `replace=TRUE` hay valores repetidos.

**Pregunta 7** → Si introducimos "ls(pat='^a')", la consola nos mostrará:

- un listado de las variables que empiezan por "a" que hemos introducido
- un listado de las variables con la "a" que hemos introducido en la consola
- un listado de todas las variables que hemos introducido en la consola

- el comando está mal escrito, nos dará error
- Si introducimos `ls()` nos saldrá un listado de todas las variables que hemos introducido en la consola. Podemos también especificar qué variables se nos muestra. Para que se nos muestren variables que contienen un determinado carácter escribimos `ls(pat='...')`. Para concretar más, si queremos que se nos muestren las variables que comienzan por un determinado carácter escribimos `ls(pat='^...')`.

**Pregunta 8** → Para limpiar la consola hacemos lo siguiente:

- escribimos en la consola `rm(list=ls(all=TRUE))`
- en la barra superior pulsamos "misc" y "remove todos los objetos"
- pulsamos las teclas control + L
- en la barra superior pulsamos "editar" y "limpar consola"
- Las respuestas verdaderas implican que se eliminará todo lo escrito en la consola pero el valor de las variables se mantendrá guardado.

**Pregunta 9** → Si escribimos la operación  $2+4*5$ , la consola:

- Da error, no se pueden hacer dos operaciones seguidas
- Realiza solo la primera operación
- Sigue un orden de izquierda a derecha: primero se hace la suma y luego la multiplicación
- La multiplicación tiene prioridad sobre la suma, así que se realiza primero
- Si aparecen dos operaciones aritméticas distintas en una misma fila, el orden que sigue R es: primero las potencias, segundo los productos y divisiones y tercero las sumas y restas. Si hay operaciones entre paréntesis, estas tienen prioridad sobre el resto.

**Pregunta 10** → Si queremos escribir un vector `z1` que contenga a los valores: 10, -3.5 y a pepito, escribimos:

- `z1<-(10, -3.5, "pepito")`
- `z1<-c(10, -3.5, pepito)`
- `z1<-c(10, -3,5, "pepito")`
- Ninguna es correcta
- La respuesta correcta sería `"z1<-c(10, -3.5, "pepito")"`

**Pregunta 11** → Dado el vector anterior `z1<-c(10, -3.5, "pepito")`, si intentamos realizar la operación `"z1[1]+z1[3]"`, nos dará error.

- Verdadero
- Falso
- No se puede realizar la suma entre 10 y "pepito", porque el segundo valor es no numérico.

**Pregunta 12** → Considerando a "vector1" y a "vector2" como vectores, al realizar `"vector1%*%vector2"` la operación que realiza la consola es el producto escalar entre ellos:

- Falso
- Verdadero
- Para realizar producto vectorial de los vectores anteriores escribiríamos `"vector1*vector2"`.