

Ejercicios de R

Curso: Introducción a la Estadística y Probabilidades CM-274

Lecturas Importantes (Sug. Hadley Wickham.)

1. Para entender como los objetos de R, funcionan de la manera que lo hacen es útil el libro *Structure and Interpretation of Computer Program* (SICP) de Harold Abelson y Gerald Jay Sussman.
<https://mitpress.mit.edu/sicp/full-text/book/book.html>.
2. Para aprender a ser un mejor programador, el libro de Andrew Hunt y David Thomas *The Pragmatic Programmer* es demasiado útil.
3. Para entender las ventajas y desventajas de R, con respecto a otros lenguajes, es necesario leer, *Concepts, Techniques and Models of Computer Programming* de Peter van Roy y Sef Haridi.

Preguntas

1. Escribe expresiones en R que generen vectores conteniendo las siguientes secuencias de vectores.
 - Las primeras 50 potencias de 3, empezando por 3^1 .
 - $1, 2^2, 3, 4^2, 5, 6^2, \dots, 99, 100^2$.
 - $1, -1, 2, -1, 3, -1, \dots, 20, -1$.

2. Sea el conjunto de datos dado por

```
> x = c(1, 8, 2, 6, 3, 8, 5, 5, 5, 5)
```

Usa R, para calcular las siguientes funciones. Usa la notación x_1 para denotar el primer elemento.

- $(x_1 + x_2 + \dots + x_{10})/10$.
 - Encontrar $\log_{10} x_i$ para cada i .
 - Resolver $(x_1 - 4.4)/2.875$ para cada i . (Hacerlo todo de una sola vez).
 - Encontrar la diferencia, entre el mayor y menor x . (Eso es el rango).
3. ¿Cuál es el funcionamiento de `is.vector()` y `is.numeric()` y explica que tiene de diferente con `is.list()` y `is.character()`?
 4. ¿Por qué es `1 = "1"` verdad (TRUE)?, ¿ Por qué `-1 < FALSE` verdad?, ¿ Por qué `"one" < 2` es falso (FALSE)?.
 5. Predice la salida del siguiente código

```
> c(1, FALSE)
> c("a", 1)
> c(list(1), "a")
> c(TRUE, 1L)
```

6. Describe como insertar un valor entre dos elementos de un vector en una posición dada, usando la función `append()` (usa la ayuda de R `help()`). Por ejemplo, muestra algunas expresiones en R, para insertar 3.4 entre `x[7]` y `x[8]` en un vector de 10 elementos. Sin `append`, ¿como se podría hacer?. ¿Qué sucede en la frontera del vector. Escribe código R en general.
7. Supongamos que `x` es un vector numérico. **Explica en detalle**, como las siguientes expresiones son evaluadas y que valores toman

```
> sum(!is.na(x))
> c(x, x[-(1:length(x))])
> x[length(x) + 1]/length(x)
> sum(x > mean(x))
```

8. Escribe código en R, que usa la función `apply`, para calcular las siguientes cantidades desde una matriz almacenada en la variables `x`.
- El máximo elemento en cada fila de `x`.
 - La media de los elementos positivos de cada fila de `x`.
 - El primer elemento de cada fila que es mayor que el valor precedente en la fila o NA si es que ese elemento no existiese.
- 9.
- Describe importantes diferencias entre las estructuras fundamentales de R: vectores, matrices, arrays y listas. Usa ejemplos para demostrar esas diferencias.
 - Explicar las diferencias entre las funciones `rbind()`, `cbind()` y `merge()` para combinar dos estructuras de dos dimensiones en R. Usa ejemplos para demostrar esas diferencias.
- 10.
- La función seno hiperbólico es definida como

$$\sinh x = \frac{e^x - e^{-x}}{2}$$

donde e^x es la función exponencial. Usando sólo operaciones aritméticas y la función `exp`, escribe una función vectorizada en R, que calcule el `sinh`.

11. Esta pregunta es acerca de vectorización (vectorization) y reciclado (recycling)
- Define que significa que una función R pueda ser vectorizada o que cumple con la vectorización. Justifica con ejemplos en R.
 - Define que significa que una función obedece la regla de reciclado. Justifica con ejemplos en R.
12. Considera la función matemática `h` definida por

$$h(x, y) = \sqrt{x^2 + y^2}$$

Escribe una función llamada `hipot`, con argumentos `x` y `y`, que implementa una versión de `h` el cuál es vectorizado y cumple la regla del `recycling`.

13. Sin usar la función `diag` de R, escribir una función que extrae los elementos de la diagonal de una matriz y retorna esos elementos como un vector.
14. ¿Qué sucede con el uso de `structure()`

```
> structure(1:5, comment = "un atributo")
```

Pero cuando se imprime ese objeto, no se ve el valor de `comment` ¿Por qué? ¿Hay algo especial en esto?.

15. ¿Qué ocurre cuando a un Factor modificas sus niveles(levels)

```
> f1 <- factor(letters)
> levels(f1) <- rev(levels(f1))
```

16. ¿Qué hace que este código?

```
> f2 <- rev(factor(letters))
> f3 <- factor(letters, levels = rev(letters))
```

¿Qué hace de diferente f2 y f3 difiere f1?.

17. ¿Como deberias describir los tres objetos?.

```
> x1 <- array(1:5, c(1, 1, 5))
> x2 <- array(1:5, c(1, 5, 1))
> x3 <- array(1:5, c(5, 1, 1))
```

18. Este script, que simula la probabilidad de obtener 3 caras en lanzamientos de monedas, está dividido en 3 partes: Escribiendo el código para la prueba, determinando el éxito de la prueba y implementando la replicación. El número 1 representa las caras y 0 los sellos.

```
> #Prueba Experimental
> prueba <- sample(0:1, 3, replace=TRUE)
> # Exito
> if (sum(prueba )==3) 1 else 0
> # Repeticion
> n <- 10000 # Numero de iteraciones
> simlista <- replicate(n, 0) ## Inicializa la lista con 0's
> for (i in 1:n)
+ {
+   prueba <- sample(0:1, 3, replace=TRUE)
+   exito <- if (sum(prueba )==3) 1 else 0
+   simlista[i] <- exito
+ }
> # Resultado simulado
> mean(simlista)
```

Modifica el código anterior para simular la probabilidad de obtener exactamente una cara en cuatro lanzamientos de moneda.