INDEXANDO

**Introducción**

R proporciona una forma poderosa y conveniente de indexar vectores. Podemos, por ejemplo, subconjuntar un vector basado en las propiedades de otro vector.

Es muy importante prestarle mucha atención a este tema, tiene el mismo peso de los factores y al dominarlo es de suma importancia pues es una parte importante en operaciones como simulaciones de Montecarlo, exploración de datos, regresiones y muchas otras operaciones que realizaremos mas adelante, este tema lo vamos a repetir las veces que sea necesario hasta que lo domine así que no se preocupe anticipadamente.

**Librerías**

Rápidamente podemos explicar que una librería es una extensión de R, la cual nos permite ampliar la funcionalidad del lenguaje.

Para instalar una librería simplemente introducimos el siguiente código:

install.packages**(**"dslabs"**)**

**Librería DSLABS**

Está es una de las primeras librerías que usaremos, fue creada por el señor Rafael Irizarry, profesor de bioestadística y biología computacional en la Universidad de Harvard, fundador del proyecto “Bioconductor”.

<https://en.wikipedia.org/wiki/Rafael_Irizarry_(scientist)>

<https://www.bioconductor.org/>

Y también catedrático de su servidor en ciencia de datos por 2 años.

Es una de las librerías principales que la Universidad de Harvard utiliza para enseñar bio estadística y ciencia de datos en sus instalaciones.

Aquí su documentación:

<https://cran.r-project.org/web/packages/dslabs/dslabs.pdf>

**Murders**

En esta sección, iniciaremos trabajando con un ejemplo de datos de asesinatos en EE. UU., que contiene la librería dslabs el cual podemos cargar así:

library**(**dslabs**)**

data**(**"murders"**)**

**Tasa de asesinatos**

Vamos a crear un objeto en donde calcularemos la tasa de asesinatos por cada 100,000 habitantes en USA, con la formula:

* Total/Población \* Cantidad de Habitantes

Aquí el código de dos formas de hacerlo.

murder\_rate **<-** murders**$**total **/** murders**$**population **\*** 100000

murder\_rate **<-** murders**$**total **/** murders**$**population **\*** 10**^**5

**Subconjuntos Lógicos**

Imagine que vive en Italia donde, según un informe de noticias de ABC, la tasa de asesinatos es de solo 0.71 por cada 100,000 habitantes.

Y debe ir a vivir a USA por un par de años; En donde claro preferiría mudarse a un estado con una tasa de homicidios similar.

Otra característica poderosa de R es que podemos usar la lógica para indexar vectores. Si comparamos un vector con un solo número, en realidad realizará la prueba para cada entrada.

En el siguiente es un ejemplo pondremos un código relacionado con la pregunta anterior en el cual crearemos un índice lógico que nos va a indicar que estados tienen una tasa inferior a 071:

ind **<-** murder\_rate **<** 0.71

ind

**[**1**]** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **TRUE** **FALSE** **FALSE** **FALSE** **TRUE** **FALSE** **FALSE** **FALSE**

**[**20**]** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **TRUE** **FALSE** **FALSE** **FALSE** **FALSE** **TRUE** **FALSE** **FALSE** **FALSE**

**[**39**]** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **TRUE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE**

Como vemos en el resultado nos arroja en un vector lógico cuales cumplen con la condición (TRUE) y cuales no (FALSE).

Si, en cambio, queremos saber si un valor es menor o igual, podemos usar:

ind **<-** murder\_rate **<=** 0.71

ind

**[**1**]** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **TRUE** **FALSE** **FALSE** **FALSE** **TRUE** **FALSE** **FALSE** **FALSE**

**[**20**]** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **TRUE** **FALSE** **FALSE** **FALSE** **FALSE** **TRUE** **FALSE** **FALSE** **FALSE**

**[**39**]** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE** **TRUE** **FALSE** **FALSE** **FALSE** **FALSE** **FALSE**

Nuevamente nos da como resultado un vector lógico con los que cumplen la condición solicitada.

**Recuperando la información**

Para recuperar la información con este índice dentro de nuestro dataframe vamos a utilizar el siguiente código, aprovechando el hecho de que los vectores pueden indexarse lógicamente.

murders**$**state**[**ind**]**

**[**1**]** "Hawaii" "Iowa" "New Hampshire" "North Dakota" "Vermont"

Como vemos en los resultados, realmente trae los datos que traemos del índice, vamos a continuar con unos ejemplos más.

Para contar cuántos son VERDADEROS, la suma de funciones devuelve la suma de las entradas de un vector.

Los vectores lógicos se convierten en numéricos con VERDADERO codificado como 1 y FALSO como 0. Por lo tanto, podemos contar los estados usando:

**>** sum**(**ind**)**

**[**1**]** 5

**Operadores Lógicos**

Supongamos que nos gustan las montañas y queremos mudarnos a un estado seguro en la región occidental de ese país.

Queremos que la tasa de asesinatos sea como máximo 1.

En este caso, queremos que dos cosas diferentes sean ciertas.

Aquí podemos usar el operador lógico, que en R se representa con **&**.

Esta operación da como resultado **VERDADERO** solo cuando ambas lógicas son VERDADERAS.

Para ver esto, considere este ejemplo:

**>** **TRUE** **&** **TRUE**;

**[**1**]** **TRUE**

**>** **TRUE** **&** **FALSE**;

**[**1**]** **FALSE**

**>** **FALSE** **&** **FALSE**

**[**1**]** **FALSE**

**>**

Ahora regresando a nuestra pregunta de vivir en el occidente y que la tasa de asesinatos sea como máximo 1 podemos tomar dos rutas.

La primera sería algo así:

west **<-** murders**$**region **==** "West"

safe **<-** murder\_rate **<=** 1

murders**$**state**[**west **&** safe**]**

**[**1**]** "Hawaii" "Idaho" "Oregon" "Utah" "Wyoming"

O creamos un índice con un vector lógico de esta manera:

ind **<-** safe **&** west

murders**$**state**[**ind**]**

**[**1**]** "Hawaii" "Idaho" "Oregon" "Utah" "Wyoming"

**Función which()**

Supongamos que queremos ver la tasa de homicidios de California.

Para este tipo de operación, es conveniente convertir vectores de lógicos en índices en lugar de mantener largos vectores de lógicos.

La función which(), nos dice qué entradas de un vector lógico son VERDADERAS.

Entonces podemos escribir:

ind **<-** which**(**murders**$**state **==** "California"**)**

murder\_rate**[**ind**]**

**[**1**]** 3.374138

**Función match()**

Si en lugar de un solo estado queremos averiguar las tasas de homicidio de varios estados, digamos Nueva York, Florida y Texas, podemos usar la función match().

Esta función nos dice qué índices de un segundo vector coinciden con cada una de las entradas de un primer vector:

Aquí el código:

ind **<-** match**(**c**(**"New York", "Florida", "Texas"**)**, murders**$**state**)**

ind

**[**1**]** 33 10 44

Ahora lo consultamos al dataframe.

murder\_rate**[**ind**]**

**[**1**]** 2.667960 3.398069 3.201360

**Función %in%**

Si en lugar de un índice queremos una lógica que nos diga si cada elemento de un primer vector está en un segundo, podemos usar la función %in%.

Imaginemos que no está seguro de si Boston, Dakota y Washington son estados.

Puedes averiguar codificando así:

c**(**"Boston", "Dakota", "Washington"**)** %in% murders**$**state

**Tenga en cuenta que usaremos %in% a menudo en todo el curso.**

**Avanzado:** hay una conexión entre coincidencias e %in%.

Para ver esto, observe que las siguientes dos líneas producen el mismo índice (aunque en orden diferente):

match**(**c**(**"New York", "Florida", "Texas"**)**, murders**$**state**)**

**[**1**]** 33 10 44

which**(**murders**$**state%in%c**(**"New York", "Florida", "Texas"**))**

**[**1**]** 10 33 44

**OPERADORES LOGICOS EN R**

**& Comparación lógica ‘AND’ (Y) elemento a elemento**

**&& Comparación lógica ‘AND’ de vectores**

**| Comparación lógica ‘OR’ (O) elemento a elemento**

**|| Comparación lógica ‘OR’ de vectores**

**! Negación lógica ‘NOT’ (NO)**

**xor() Exclusión ‘OR’ elemento a elemento**

**equivalente a !( x | y)**

**Practica**

**Instrucciones:** Utilice la librería de “dslabs” y los datos de “murders” para resolver las siguientes preguntas.

**Pregunta No. 1**

Calcule la tasa de asesinatos por cada 100,000 personas, para cada estado y guárdela en un objeto llamado murder\_rate. Luego, use operadores lógicos para crear un vector lógico llamado low que nos indique qué entradas de murder\_rate son inferiores a 1.

**Pregunta No. 2**

Ahora use los resultados del ejercicio anterior y la función whitch() para determinar los índices de tasa de asesinatos asociados con valores inferiores a 1.

**Pregunta No. 3**

Use los resultados del ejercicio anterior para informar los nombres de los estados con tasas de asesinatos inferiores a 1.

**Pregunta No. 4**

Ahora extienda el código del ejercicio 2 y 3 para informar los estados en el noreste con tasas de homicidios inferiores a 1.

Sugerencia: use el vector lógico previamente definido y el operador lógico &.

**Pregunta No. 5**

En un ejercicio anterior, calculamos la tasa de homicidios de cada estado y el promedio de estos números. ¿Cuántos estados están por debajo del promedio?

**Pregunta No. 6**

Use la función match() para identificar los estados con abreviaturas AK, MI e IA. Sugerencia: comience definiendo un índice de las entradas de asesinatos $ abb que coincidan con las tres abreviaturas, luego use el operador “[]“ para extraer los estados.

**Pregunta No. 7**

Use el operador %in% para crear un vector lógico que responda a la pregunta: ¿cuáles de las siguientes son abreviaturas reales: MA, ME, MI, MO, MU?

**Pregunta No. 8**

Extienda el código que utilizó en el ejercicio 7 para informar la entrada que no es una abreviatura real. Sugerencia: use el operador “!”, que convierte FALSO en VERDADERO y viceversa, para obtener un índice.