Básicos de manejar R

Análisis Exploratorio de Datos

En esta lección, vamos a explorar algunos elementos básicos de la programación en el lenguaje R. Me imagino que pocos quieran aprender programación en sí, pero algunos básicos son necesarios para que saquen lo mejor de R. Mientras más comodidad tiene con el programa, más atención pueda dirigir a nuestro propósito verdadero de análisis.

Antes de empezar, sepa que si en algún momento desea obtener más información (en inglés) sobre un tema concreto relacionado con R, puede escribir help.start() en el indicador >. Se abrirá un menú de recursos (ya sea dentro o RStudio su navegador web por defecto, dependiendo de su configuración). Como alternativa, una simple búsqueda en Internet a menudo da la respuesta que estás buscando.

Ahora, en su forma más simple, R puede ser utilizado como una calculadora interactiva. Escribe 2 + 2 y pulsa intro.

```
2 + 2
```

```
## [1] 4
```

Como se puede imaginar, todos las operaciones normales de calculador van a funcionar así. Para multiplicar usamos el asterisco, teclea 2 * 2 para averiguarlo.

```
2 * 2
```

```
## [1] 4
```

También se puede levantar números a cualquier poder mediante el uso del símbolo '^'). Depende su teclado, no se ubicara facilmente. Si no aparece nada, pulsa space antes de cumplir la combinación de teclas. Vas a buscarlo ahora mismo para cuadrar 2. Escribe 2^2 y presione Intro.

```
2^2
```

```
## [1] 4
```

Ahora, otro ejemplo sencillo. Escribe 5 + 7 y pulsa Intro.

```
5 + 7
```

```
## [1] 12
```

Ojo que con estos ejemplos R simplemente imprime el resultado de 12 a la consola por defecto. Sin embargo, R es un lenguaje de programación y a menudo la razón por la que utiliza un lenguaje de programación, en oposición a una calculadora, es para automatizar algunos procesos o evitar repeticiones innecesarias.

En este caso, es posible que desee utilizar nuestro resultado de arriba en un segundo cálculo. En lugar de volver a escribir 5 + 7 cada vez que lo necesitamos, podemos simplemente crear un nuevo objeto que almacena el resultado.

La forma en que asigna un valor a un objeto en R es mediante el uso del operador de asignación, que es sólo un símbolo ''menor que''seguido de un signo''menos". Eso se parece a esto: <-

Piense en el operador de asignación como una flecha. Usted está asignando el valor en el lado derecho de la flecha para el nombre del objeto en el lado izquierdo de la flecha.

Para asignar el resultado de 5 + 7 a un nuevo objeto llamada x, escriba x <- 5 + 7. Esto puede leerse como 'x toma el valor de 5 más 7'. Dese una oportunidad ahora.

```
x <- 5 + 7
```

Se dará cuenta de que R no imprime el resultado de 12 en esta ocasión. Cuando se utilice el operador de asignación ("<-"), R asume que no quieres ver el resultado inmediatamente, sino que tiene la intención de utilizar el resultado para otra cosa más adelante.

Para ver el contenido del objeto x, simplemente escriba x y pulse Intro. Intentalo ahora.

```
x
```

```
## [1] 12
```

Ahora, almacene el resultado de x - 3 en un objeto llamado nueva y.

```
y <- x - 3
```

¿Cuál es el valor de y? Escriba y para averiguarlo.

```
y
```

```
## [1] 9
```

Ahora, vamos a crear una pequeña colección de números llamados un vector. Cualquier objeto que contiene datos se denomina una estructura de datos numéricos y los vectores son de las más simples tipos de estructuras de datos en R. De hecho, incluso un solo número se considera un vector de longitud uno.

La forma más fácil de crear un vector es con la función c(), que se usa para 'concatenar' o 'combinar'. Para crear un vector que contiene los números 1.1, 9 y 3.14, escriba c(1.1, 9, 3.14). Pruébalo ahora y guarde el resultado en un objeto llamado z.

```
z <- c(1.1, 9, 3.14)
```

Escriba z para ver su contenido. Verás que los valores impresos no son separados por comas.

```
z
```

```
## [1] 1.10 9.00 3.14
```

Usted puede combinar vectores para hacer un nuevo vector. Crear un nuevo vector que contiene z, 555, entonces z de nuevo en ese orden. No asigne este vector a un nuevo objeto, de modo que sólo podemos ver el resultado inmediatamente.

```
c(z, 555, z)
```

```
## [1]   1.10   9.00   3.14 555.00   1.10   9.00   3.14
```

Vectores numéricos se puede utilizar en expresiones aritméticas. Escriba lo siguiente para ver qué pasa: z * 2 + 100.

```
z * 2 + 100
```

```
## [1] 102.20 118.00 106.28
```

En primer lugar, R multiplica cada uno de los tres elementos en z por 2. Entonces se añadieron 100 a cada elemento para obtener el resultado que ves arriba.

Como ya te decía, otros operadores aritméticos comunes son `+`, `-`, `/` y `^` (donde x ^ 2 significa 'x al cuadrado'). Para tomar la raíz cuadrada, utilice la función sqrt() y tomar el valor absoluto, utilice la función abs().

Dese una oportunidad, calcula la raiz quadrado de 100 con la función sqrt().

```
sqrt(100)
```

```
## [1] 10
```

Ahora, tome la raíz cuadrada de z - 1 y asignarlo a un nuevo objeto llamada mi_sqrt.

```
mi_sqrt <- sqrt(z - 1)
```

Antes de ver el contenido del objeto mí_sqrt, antes de seguir, entonces, ¿qué crees que contiene?

- un vector de longitud 3
- un solo número (es decir, un vector de longitud 1)
- un vector de longitud 0 (es decir, un vector vacío)

Ahora imprime el contenido de mi_sqrt.

```
mi_sqrt
```

```
## [1] 0.3162278 2.8284271 1.4628739
```

Como habrás adivinado, R primero resta 1 de cada elemento de la z, luego tomó la raíz cuadrada de cada elemento. Esto le deja con un vector de la misma longitud que el vector z inicial.

Ahora, cree un objeto llamado nueva mí_div que obtiene el valor de z dividido por mi_sqrt.

```
mi_div <- z / mi_sqrt
```

¿Cuál crees que es la declaración verdadera?

- El primer elemento de mi_div es igual al primer elemento de z dividido por el primer elemento de mi_sqrt, y así sucesivamente ...
- mi_div es un solo número (es decir, un vector de longitud 1)
- mi_div es indefinido

Adelante e imprimir el contenido de mi_div.

```
mi_div
```

```
## [1] 3.478505 3.181981 2.146460
```

Cuando se le presenten dos vectores de la misma longitud, R simplemente realiza la operación aritmética especificada (+, -, *, etc.) elemento por elemento. Si los vectores son de diferentes longitudes, R 'recicla' el vector más corto hasta que es la misma longitud que el vector más largo.

Cuando hicimos z * 2 + 100 en nuestro ejemplo anterior, z era un vector de longitud 3, pero técnicamente 2 y 100 son cada uno de vectores de longitud 1.

Detrás de las escenas, R es reciclar, el 2 para hacerlo un vector de 2s y el 100 para hacer un vector de 100s. En otras palabras, cuando usted pide R para calcular z * 2 + 100, lo que realmente se calcula es z * c(2, 2, 2) + c(100, 100, 100).

Para ver otro ejemplo de cómo funciona este vector «reciclado», trate de sumar c(1, 2, 3, 4) y c(0, 10). No te preocupes por guardar el resultado en un nuevo objeto.

```
c(1, 2, 3, 4) + c(0, 10)
```

```
## [1]  1 12  3 14
```

Si la longitud del vector más corto no divide uniformemente en la longitud del vector ya, R se seguirá aplicando el método «reciclado», pero lanzará una advertencia para hacerle saber algo sospechoso podría estar pasando.

Trate c(1, 2, 3, 4) + c(0, 10, 100) para un ejemplo.

```
c(1, 2, 3, 4) + c(0, 10, 100)
```

```
## Warning in c(1, 2, 3, 4) + c(0, 10, 100): longer object length is not a
## multiple of shorter object length
```

```
## [1]   1  12 103   4
```

Antes de concluir esta lección, me gustaría mostrarte un par de trucos para ahorrar tiempo.

A principios de la lección, z multiplica * 2 + 100. Supongamos que has cometido un error y querías sumar 1.000 en lugar de 100. Usted podría volver a escribir la expresión, o . . .

En muchos entornos de programación, la tecla flecha hacia arriba recorre a través de los comandos anteriores. Trate de teclear la flecha hacia arriba en el teclado hasta que se llegue a este comando (z * 2 + 100), a continuación, cambiae de 100 a 1000 y pulse Enter. Si la flecha hacia arriba no funciona para ti, sólo tienes que escribir el comando corregido.

```
z * 2 + 1000
```

```
## [1] 1002.20 1018.00 1006.28
```

Por último, vamos a suponer que desea ver el contenido de un objeto que ha creado anteriormente, pero parece que no puede recordar si usted la nombró como mi_div o miDiv. Podrías probar ambos y ver lo que funciona, o . . .

Puede escribir las dos primeras letras del nombre del objeto, y luego presionar la tecla Tab (posiblemente más de una vez). La mayoría de los entornos de programación proporcionará una lista de los objetos que ha creado que comiencen con 'mi'. Esto se llama auto-completado y puede ser muy útil cuando se tienen muchos objetos en el espacio de trabajo. Dese una oportunidad. (Si el autocompletado no funciona para ti, sólo tienes que escribir mi_div y pulse Intro.)

```
mi_div
```

```
## [1] 3.478505 3.181981 2.146460
```

Pues, felicitaciones, has terminado con esta lección en los básicos de manejar R. Espero que ya te sientas más cómod@ trabajando con ella. Puedes seguir con la siguiente class, experimentar con R, o volver otro día.

Esta lección se escribió con el paquete swirl, representa el labor de muchos voluntarios. A mi me gustaría recibir retroalimentación en el tema de que tan útiles son estas lecciones. Por favor, espera hasta que hayas completado varias antes de juzgar. ¡Hasta luego!