

Resumen de funciones para el Taller 1 y el curso en general

Simon Pedro Galeano Muñoz

11 de octubre de 2021

1 Operaciones básicas de R

- Aritmética
- Lógica

2 Trabajo con bases de datos

- Operaciones básicas con bases de datos

3 Tidyverse

- Pipes

Creando variables en R

Supongamos que se quiere crear una variable x en el lenguaje de programación R y se desea almacenar el valor de π allí.

Dicha tarea se consigue usando la sintaxis que se muestra a continuación.

$x \leftarrow \pi$, donde \leftarrow se escribe como `<-`

En las variables se pueden guardar múltiples tipos de datos, como números, cadenas de caracteres, bases de datos, vectores y matrices, entre otros. En nuestro caso particular los valores numéricos, bases de datos, vectores y matrices serán de gran interés a lo largo del curso.

Vectores con el operador `c()`

Suponga que queremos crear el vector $z = [1 \ 3 \ 5 \ 7]$ en R, la instrucción para obtener el resultado deseado es

$$z \leftarrow c(1, 3, 5, 7)$$

La instrucción `c()` significa concatenar, es decir, R está juntando los valores 1, 3, 5 y 7. Si se deseara agregar el número 9 al vector z simplemente basta con hacer

$$z \leftarrow c(z, 9)$$

Pues es cuestión de simplemente concatenar el número 9 al vector z .



Los vectores creados usando concatenación no son vectores fila ni vectores columna, más adelante se discutirá acerca de la orientación de los vectores.

Aritmética

La tabla que se muestra a continuación lista las operaciones aritméticas básicas que son más frecuentes en lo que respecta al curso.

Operación	Símbolo
Suma	+
Resta	-
Multiplicación	*
División	/
Exponenciación	^

Algunos ejemplos de operaciones aritméticas

Considere los vectores a , b y d definidos como se muestra

$$a \leftarrow c(1, 2, 3), \quad b \leftarrow c(4, 5, 6) \quad \text{y} \quad d \leftarrow c(7, 8, 9, 10)$$

Veamos el resultado de algunas operaciones

Operación	Resultado
$2 + 2$	4
$a + b$	$c(5, 7, 9)$
$d - a$	$c(6, 6, 6, 9)$
$a * a$	$c(1, 4, 9)$
$d + 2$	$c(9, 10, 11, 12)$

R hace operaciones elemento a elemento y en caso de que las dimensiones no coincidan, R se encarga de extender el elemento de menor dimensión tanto como sea necesario para poder efectuar la operación.

¡R es vectorizado!

R es un lenguaje vectorizado, es decir, aplica operaciones a vectores enteros, ejecutando dicha operación elemento a elemento.

Operación	Resultado
a^3	$c(1^3, 2^3, 3^3)$
$\log(b)$	$c(\log(4), \log(5), \log(6))$
$\exp(d)$	$c(\exp(7), \exp(8), \exp(9), \exp(10))$

Algunas funciones de utilidad son las siguientes

Función	Descripción
$\text{mean}(\cdot)$	Retorna la media de \cdot
$\text{sum}(\cdot)$	Retorna la suma de los elementos de \cdot
$\text{length}(\cdot)$	Retorna la cantidad de elementos de \cdot
$\text{sd}(\cdot)$	Retorna la desviación estándar de \cdot

Operaciones lógicas

Durante las sesiones de taller no va a ser muy frecuente el uso de operadores lógicos, sin embargo se listan ya que son importantes a la hora de filtrar.

Operador	Descripción
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
==	Igual que
!=	Diferente
&	Y lógico
	O lógico
!	Negación

Es importante recordar que dichos operadores lógicos retornan un Booleano, es decir, TRUE o FALSE.

Vamos a ver una base de datos simplemente como una tabla con filas y columnas, donde las filas representan las observaciones y las columnas representan las variables. Se considera la siguiente base de datos (la cual llamaremos X) para las explicaciones posteriores.

X1	X2	X3	X4	X5
-0.63	0.18	-0.84	1.60	0.33
0.22	-0.54	0.89	0.60	1.64
-0.77	-0.82	-0.14	-0.28	0.44
0.48	-0.13	1.10	-1.44	1.15
-0.21	-1.04	-1.15	0.32	-1.50

Extrayendo las dimensiones y columnas de la base de datos

Para saber las dimensiones de una base de datos, se usa la función `dim(.)` donde `.` es la base de datos en cuestión. Dicha función retorna un vector con el número de filas y columnas. En el caso particular de `X` se tiene que `dim(X)` es igual `c(5, 5)`.

Para extraer alguna variable (en general elementos de objetos) de una base de datos se usa el operador `$`. Por ejemplo, el comando para extraer la variable `X3` de la base de datos `X` es `X$X3`.

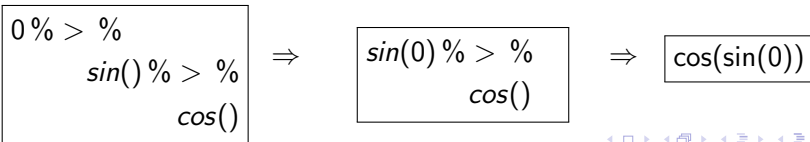
Pipes o tuberías

El símbolo `%>%` denota el operador pipe u operador de tubería, cuya función permite el encadenamiento de acciones. Específicamente permite componer funciones.

Por ejemplo, suponga que se quiere calcular $\cos(\sin(0))$, la manera de realizar esto usando el operador pipe es la siguiente.

$$\begin{aligned} 0 \%>\% & \sin() \%>\% & \cos() \end{aligned}$$

Analizando la ejecución del pipe paso a paso



Encadenando funciones de varias variables

Supongamos que se tienen funciones $f(\cdot)$, $g(\cdot, \cdot)$ y $h(\cdot, \dots)$. Si se quisiera calcular $h(g(f(x), y), \dots)$ la manera de hacerlo usando pipes sería la siguiente

$$\begin{aligned} x \%>\ \% \\ &f() \%>\ \% \\ &g(y) \%>\ \% \\ &h(\dots) \end{aligned}$$

Observe que si la función tiene más parámetros, estos se ingresan normalmente, ya que el pipe lo que hace es enviar el resultado anterior como primer argumento a la función siguiente.

Analizando con más detalle

```
x %>%
  f() %>%
    g(y) %>%
      h(...)
```



```
f(x) %>%
  g(y) %>%
    h(...)
```



```
h(g(f(x), y), ...)
```



```
g(f(x), y) %>%
  h(...)
```

Información a usar

Objeto	Ecuación
S_{xx}	$\sum_{i=1}^n (x_i - \bar{x})^2$
S_{yy}	$\sum_{i=1}^n (y_i - \bar{y})^2$
S_{xy}	$\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$
$\hat{\beta}_1$	$\frac{S_{xy}}{S_{xx}}$
$\hat{\beta}_0$	$\bar{y} - \hat{\beta}_1 \bar{x}$
$\hat{\sigma}^2$	$\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n-2}$
Intervalo para β_0	$\hat{\beta}_0 \pm t_{\frac{\alpha}{2}, n-2} \sqrt{\frac{\hat{\sigma}^2 \sum_{i=1}^n x_i^2}{n S_{xx}}}$
Intervalo para β_1	$\hat{\beta}_1 \pm t_{\frac{\alpha}{2}, n-2} \sqrt{\frac{\hat{\sigma}^2}{S_{xx}}}$