

# Introducción a R

## Comandos para R

### Comandos para interactuar con el interfaz o dentro del script en particular

- Ctrl + shift + N: crea un nuevo script. Sirve también cuando accidentalmente cierran el script, sólo vean el ambiente “consola” del lado izquierdo y no sepan como volver a escribir sobre el script.
- Ctrl + L: limpia la consola (no hace falta posicionarse en ella).
- Ctrl + enter: correr el código seleccionado. En caso de no haber seleccionado nada, ejecutará el código sobre donde se encuentre uno. Por ejemplo, si hacemos clic en la línea 32 en cualquier parte del código, se ejecutará todo el código. Siempre que uno quiera ejecutar una línea en concreto, se recomienda hacer clic (es decir, posicionarse) sobre el final del código para controlar mejor la extensión del mismo. Ctrl + shift + enter: ejecuta todo el script. No se recomienda su uso por razones muy sencillas: 1. ¿para qué queremos volver a ejecutar códigos anteriores que sabemos que funcionan y nos devuelven lo que esperábamos que devolvieran? 2. A mayor cantidad de códigos que se escriban en el script y/o mayor cantidad de datos que con los que estemos trabajando, más demora tendrá Rstudio en devolvernos el resultado. Este comando se menciona, mas si el alumno puede olvidarse de haberlo leído, mejor.
- Ctrl + F: activa el modo búsqueda y reemplazo.
- Ctrl + +/-: Zoom-in/Zoom-out.
- Ctrl + D: borrar todo lo escrito en la línea.
- Ctrl + I: desplazarse a la línea de abajo (nótese que línea es diferente a renglón).
- Comandos útiles para desplazarse que por lo general sirven en todos los editores de texto.
  - Ctrl + flecha izquierda/derecha: se desplaza hacia la palabra de atrás o hacia la palabra posterior.
  - Tecla Inicio: nos desplazamos hasta el primer caracter de la línea.
  - Tecla Fin: nos desplazamos hasta el último caracter de la línea.
  - Ctrl + Tecla inicio: Nos desplazamos hacia el inicio del documento de texto.
  - Ctrl + Tecla fin: Nos desplazamos hacia el final del documento de texto.
  - Ctrl + shift + flecha derecha (izquierda): seleccionamos toda el texto hasta el próximo (anterior) espacio en blanco, si se vuelve a presionar se añadirá el texto hasta el siguiente (anterior) espacio en blanco y así...
  - Shift + inicio (final): selecciona el texto hasta el comienzo (final) del renglón.
  - Ctrl + tab: nos movemos hacia la pestaña de la derecha (en caso de existir).

- Ctrl + shift + tab: nos movemos hacia la pestaña de la izquierda (en caso de existir)[^nota1: el alumno puede verificar que también funciona en google chrome].

A veces es útil realizar anotaciones o aclaraciones en el espacio donde se codifica, ya sea para mencionar lo que se está haciendo, esto puede realizarse mediante la utilización del caracter '#'. Nótese que, en cuanto uno escriba este caracter e intente correr la línea, la consola no devolverá un resultado, sino que omitirá por completo la línea o imprimirá la línea con un resultado seguido de la anotación. Por ejemplo, intente correr la siguiente oración -sin las comillas: "2+4 #esto es dos más cuatro", la consola registrará este "pedido" y devolverá simplemente 6. Como se dijo, es útil para escribir aclaraciones en el mismo espacio en el que se codifica.

### Atajos y comentarios de Rmarkdown

- Ctrl + Alt + I: crea un *chunk*. Un *chunk* es, básicamente un espacio dónde hay fragmentos de código, o donde funcionan las mismas "reglas de juego" que en el script de R.
- Nótese que en Rmarkdown no es necesario escribir # para que R no la interprete como código. En este ambiente, si no se escribe un *chunk*, Rmarkdown devolverá todo en forma de texto.

La implementación de estos comandos a la hora de escribir ahorran tiempo y mantienen las manos en el espacio de teclado, lo que incrementa también el rendimiento de escritura.

### Consejos útiles

Antes de comenzar, mencionar lo que se cree más importante en el manejo de R, Rstudio y, en general, la práctica de programar: ser ordenados y claros. Los resultados que se desprendan de los códigos pueden reflejar lo que uno busca obtener, pero en cuanto ocurran errores (que efectivamente ocurrirán), será mucho más sencillo encontrarlos. Para ello, se recomienda lo siguiente a la hora de escribir en el script:

- Dejar un renglón en blanco por cada código que se escribe.
- Dejar espacios entre los operadores matemáticos
- Crear la cantidad justa de objetos y eliminar aquellos con los que no se trabaja (uno siempre puede volver a crearlos, se borra el objeto, no el código)
- Muchas veces sucederá que ciertas líneas de código sean similares o se confundan, para combatir esto a veces es útil realizar aclaraciones mediante el caracter #, como si se escribiera una receta con los códigos y al costado (o antes de escribir el código) se especificara lo que uno esta haciendo. Realizar esto no sólo brindará información al usuario (dado que uno puede olvidarse de qué fue lo que quiso obtener cuando escribió el código), sino también que, en caso de compartir el archivo que contiene el código, otra persona puede guiarse de una mejor manera.

Otra idea que puede llegar a aligerar el proceso no sólo de aprendizaje, sino también de trabajo en R es la siguiente: hay muchas maneras o alternativas para lograr un mismo resultado en R. Sin embargo, es cierto que hay vías más eficientes que otras. Esta cuestión tiene especial importancia cuando uno busca comparar con otra persona el procedimiento

por el cual llego al resultado y encuentra que la otra persona utilizó vías diferentes. En este sentido, se cree que la comparación de procedimiento a la hora de llegar un resultado enriquece el proceso de aprendizaje.

Otro consejo de utilidad puede ser confeccionar una lista de los comandos (códigos, funciones, etc.) que vaya aprendiendo para tener a mano en caso de necesitar realizar una determinada operación.