

R inicial

Valentín Vergara Hidd

18 de septiembre de 2017

La intención de este documento es ser una primera aproximación al uso del *lenguaje* R. Como tal, su intención no es comenzar a trabajar en funciones ni en estructuras más avanzadas, sino que simplemente familiarizarse con la instalación, espacios de trabajo, uso de paquetes y funcionamiento de objetos.

Como requisito para el trabajo con este documento, se requiere tener instalado R¹ y aunque no es necesario, sugiero complementarlo con una GUI², de las cuales recomiendo R Studio³.

1. Contexto

R es una implementación del lenguaje de programación S, que siguió su propio camino de desarrollo a partir de 1992 en la Universidad de Auckland, Nueva Zelanda. Su desarrollo estuvo a cargo de Ross Ihaka y Robert Gentleman (de ahí el nombre R). La primera versión estable (beta) se lanzó en 2000, momento desde el que ha adquirido cada vez más popularidad en el mundo académico.

Como se observa en la imagen a continuación, el uso de R ha ido en constante aumento desde su primera versión estable. Se atribuye esta creciente popularidad en que es *software libre*, pero que además utiliza un lenguaje para programar funciones que es amigable con quienes no son programadores. Esto ha posibilitado que muchos académicos dedicados a la estadística, hayan creado *paquetes* para implementar algoritmos nuevos en R. A todo lo anterior se suman sus capacidades profesionales para gráficos, que generalmente es un punto débil de otros software estadísticos.

¹Software libre que se puede obtener en [este sitio](#)

²*Graphical User Interface*

³Software comercial, con una de sus licencias de uso gratuito, que se puede encontrar en [este sitio](#)

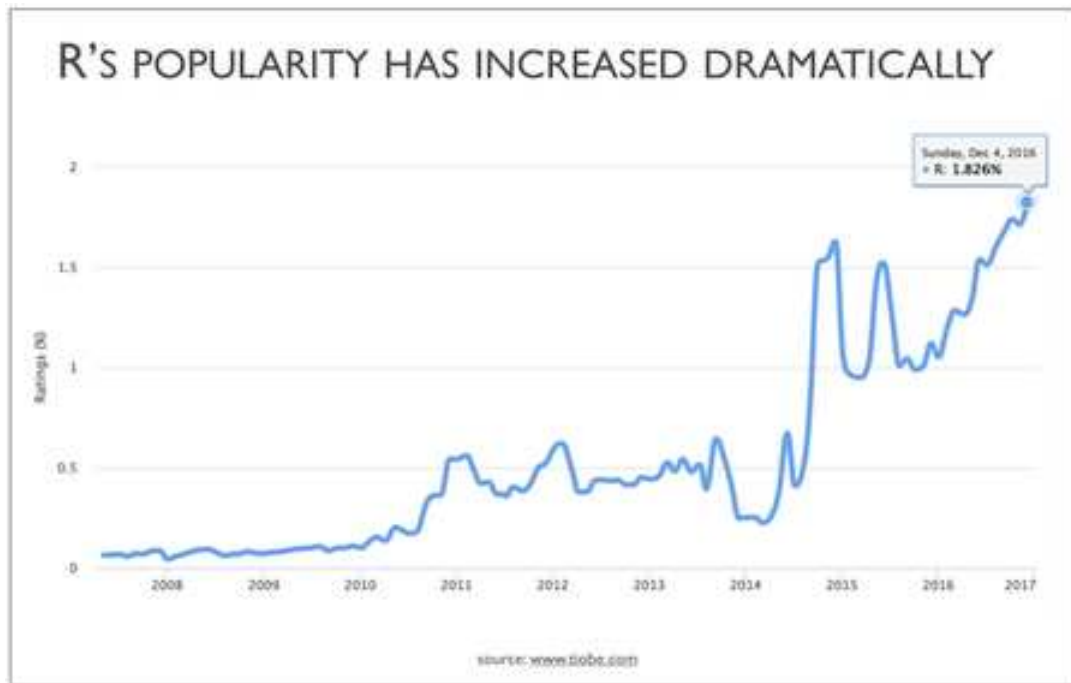


Figura 1: Uso de R en *data science* desde el año 2000

2. La consola como punto de partida

R trabaja con una consola, que no es otra cosa que un terminal de texto donde se escriben las instrucciones y donde se muestran algunos resultados. Al ejecutar R (no R Studio, eso lo veremos más adelante), desde cualquier sistema operativo, deberían visualizar lo siguiente:

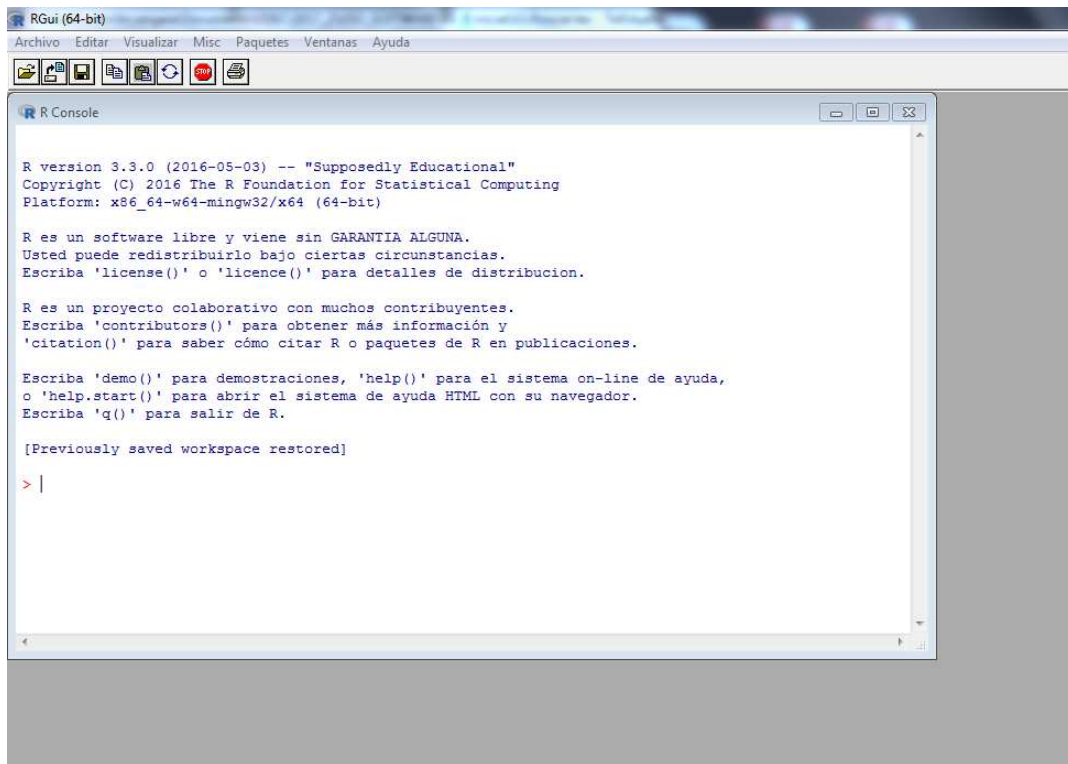


Figura 2: Vista inicial de R, en Windows

Donde se encuentra el cursor, se puede comenzar a escribir, con la consideración que para ejecutar una instrucción, se debe presionar la tecla Enter.

2.1. R como una calculadora

R funciona *como* una calculadora, principalmente debido a que al ingresar expresiones en la consola, obtenemos su resultado, si eso es posible. Por ejemplo, si nos interesa una suma entre dos números (6 y 19), ingresamos

```
> 6 + 19  
[1] 25
```

El signo *mayor que* se conoce como el **prompt**, lo que significa que al aparecer ese signo, la consola de R está lista para recibir más instrucciones. Podrán notar que en el ejemplo anterior se utilizó el operador +. De la misma forma, existen otros operadores que se pueden utilizar, algunos de los cuales se encuentran en la tabla a continuación.

Tabla 1: Algunos operadores numéricos en R. Se asume que x y y son números.

Operador	Definición	Ejemplo
+	Adición	$x + y$
-	Sustracción	$x - y$
*	Multiplicación	$x*y$
/	División	x/y
^	Potencia	x^y
log()	Logaritmo natural	log(x)
log10()	Logaritmo base 10	log10(x)
log(x,y)	Logaritmo base y	log(x,y)
sqrt()	Raíz Cuadrada	sqrt(x)

Es importante señalar que R considera las expresiones en un sentido algebraico, por lo que se debe cuidar el uso de paréntesis donde corresponda. Si sabemos, por ejemplo, que

$$23 + 7 + 14 \div 2 \neq (23 + 7 + 14) \div 2$$

Lo podemos probar en R.

```
> 23 + 7 + 14 / 2
[1] 37
> (23 + 7 + 14) / 2
[1] 22
```

Como nota adicional, considere que el 1 que aparece entre corchetes indica que el resultado de la operación solicitada es un vector cuyo primer elemento se *imprime* en la pantalla. Además, existe la posibilidad de obtener un error, si es que lo que se entrega como valor entre los operadores no corresponde a un número ni a un objeto previamente asignado en R.

```
> x + y
Error: object 'x' not found
```

Al intentar ingresar la instrucción anterior en la consola, se obtiene un error, al comprobar que x no es un número; ni tampoco un objeto en la memoria, por lo que el proceso se detiene sin siquiera alcanzar a evaluar y .

También se debe considerar que se pueden combinar distintos operadores, por ejemplo.

```
> log(500*2, sqrt(100))
[1] 3
```

3. Objetos

Como ya se mencionó, si intentamos sumar x y y sin que estos sean *objetos* previamente guardados en la memoria de la sesión abierta de R, vamos a obtener un error. La solución, es asignar valores a estas letras. El operador fundamental para esto es `<-` (El signo *menor que* seguido de un guión). Lo que sea que quede a la izquierda de este operador, será el objeto; y lo que sea que quede a la derecha, va a ser el contenido de ese objeto.

Lo más simple que podemos asignar a un objeto es un valor numérico. Es decir, una constante:

```
> x<-10
> y<-17
> x + y
[1] 27
```

También podemos usar algunas expresiones ya creadas dentro de otras. Por ejemplo:

```
> x<-27/3
> y<-log(x)
> z<-exp(y)
> z
[1] 9
```

Aprovechando las propiedades de e y de los logaritmos naturales, probamos que se pueden incluir valores de objetos en otros objetos.