

Tutorial de Introducción a R

Programación en lenguajes estadísticos

Carrascal Deiver¹, Díaz Eduard², Guevara Eduardo³, Mejia Bunkuaney⁴
dcarrascal@unal.edu.co¹, eddiaz@unal.edu.co²,
edguevarap@unal.edu.co³, bmejia@unal.edu.co⁴

junio de 2022

1. Resumen

R es un lenguaje de programación y un entorno de software para análisis estadístico, gráficos representación y presentación de informes. R fue creado por Ross Ihaka y Robert Gentleman en el Universidad de Auckland, Nueva Zelanda, y actualmente es desarrollado por R Development Equipo central.

El núcleo de R es un lenguaje informático interpretado que permite la ramificación y el bucle así como la programación modular mediante funciones. R permite la integración con los procedimientos escritos en los lenguajes C, C++, .Net, Python o FORTRAN para mayor eficiencia.

R está disponible gratuitamente bajo la Licencia pública general de GNU y el binario precompilado proporciona versiones para varios sistemas operativos como Linux, Windows y Mac. R es un software libre distribuido bajo una copia de estilo GNU a la izquierda y una parte oficial de GNU proyecto llamado GNUS.

2. Abstract

R is a programming language and software environment for statistical analysis, graphics representation and reporting. R was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team. The core of R is an interpreted computer language which allows branching and looping as well as modular programming using functions. R allows integration with the procedures written in the C, C++, .Net, Python or FORTRAN languages for efficiency.

R is freely available under the GNU General Public License, and pre-compiled binary versions are provided for various operating systems like Linux, Windows and Mac. R is free software distributed under a GNU-style copy left, and an official part of the GNU project called GNU S.

3. Instalación Local de R

Para la instalación local de un lenguaje de programación como lo es R en nuestros computadores de confianza es conveniente para su buen funcionamiento e instalación seguir los siguientes pasos:

- **Paso 1:** Dirigirnos a nuestro navegador de confianza y escribir en la barra de búsqueda lo siguiente: Download R for Windows, o dirigirse al siguiente link (<https://cran.r-project.org/bin/windows/base/>), donde nos encontraremos la siguiente ventana:

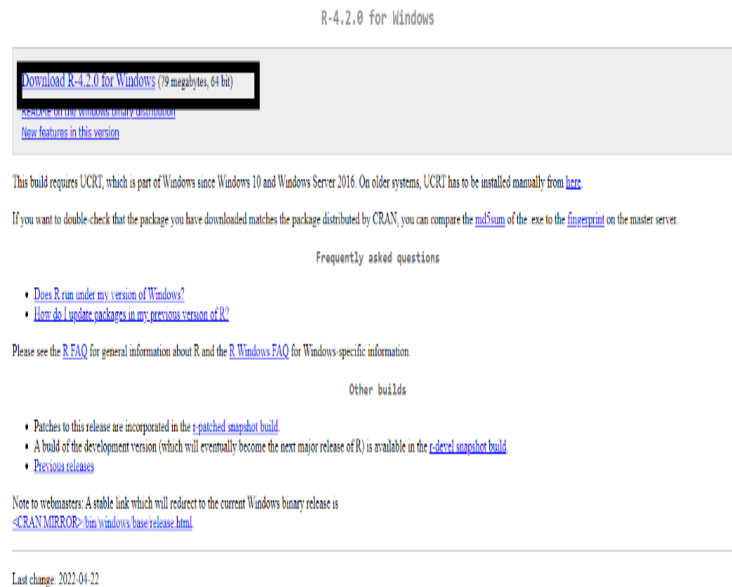


Figura 1:

- **Paso 2:** En la pestaña anterior daremos clic en el rectángulo que se muestra, donde se nos descargar un archivo .exe, que al ejecutarlo como administrador y este nos mostrara la siguiente ventana, donde elegiremos el idioma y aceptamos la elección:

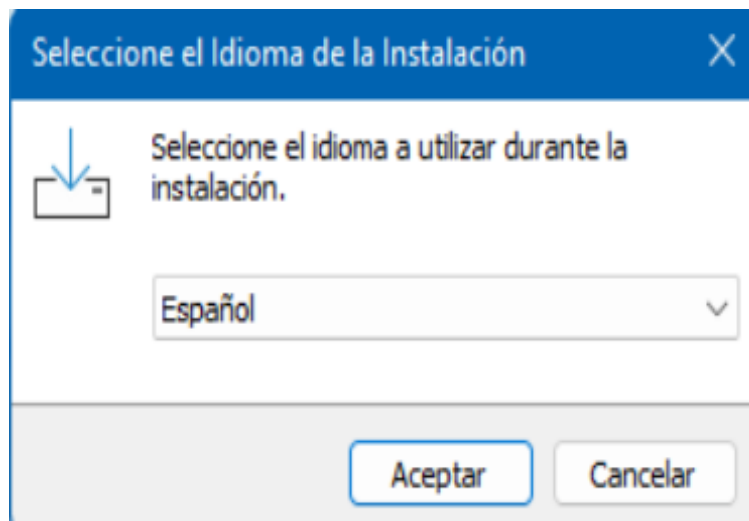


Figura 2:

- **Paso 3:** Luego de aceptar en la ventana anterior aparecerá una nueva ventana a la cual le daremos en el botón de siguiente:

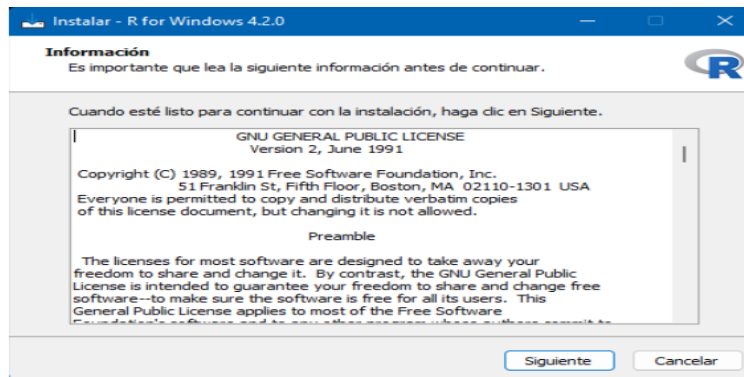


Figura 3:

- **Paso 4:** A continuación elegiremos la carpeta donde queremos que se guarde la instalación de dicho programa, acompañado de un clic en el botón siguiente:

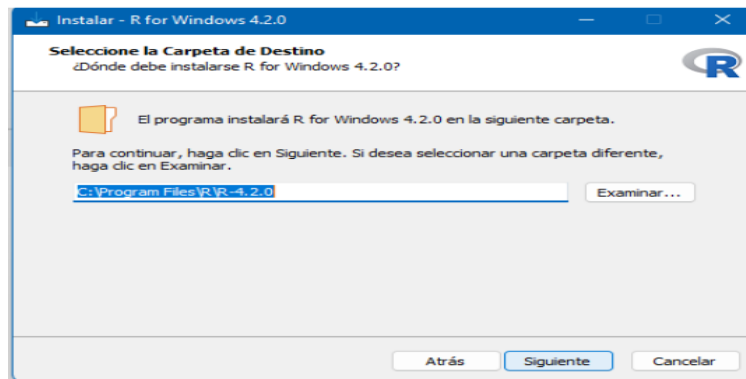


Figura 4:

- **Paso 5:** En la siguiente ventana elegiremos los componentes del programa que queremos que se descarguen, de los cuales seleccionaremos todos como se muestra en la siguiente imagen:

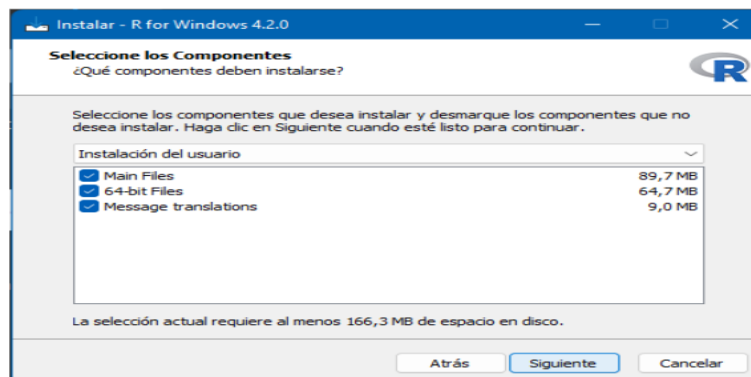


Figura 5:

- **Paso 6:** Luego especificaremos si queremos o no usar las opciones de configuración del programa:

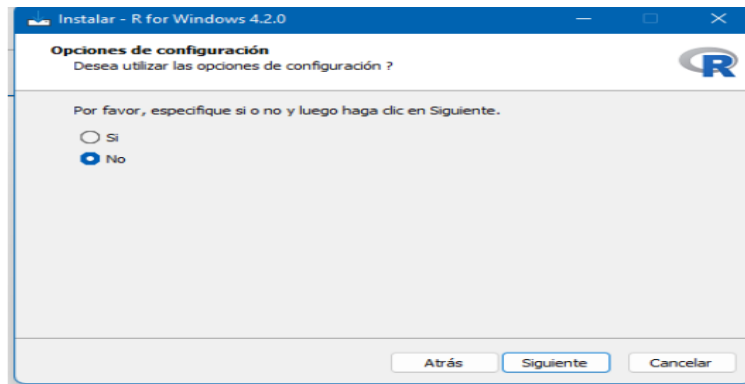


Figura 6:

- **Paso 7:** De igual forma el programa nos preguntará donde queremos que estén los accesos directos de este mismo, el cual elegiremos en el explorador de archivos :

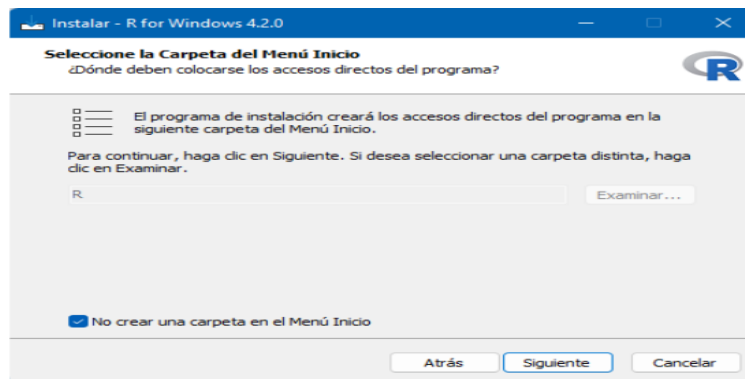


Figura 7:

- **Paso 8:** Por último el programa comenzará su instalación:

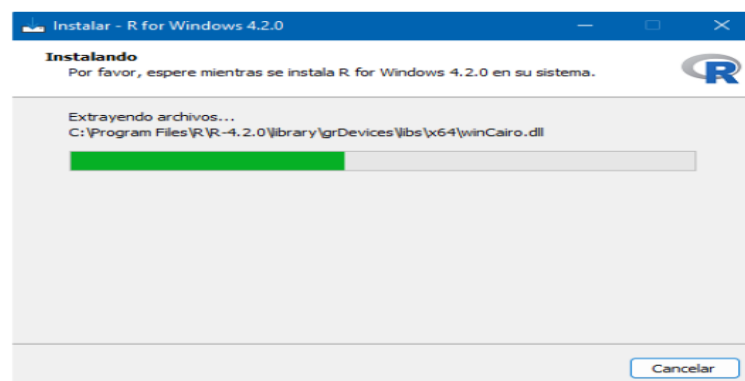


Figura 8:

- **Paso 9:** Luego de realizar todos los pasos ya tendremos R instalado localmente en nuestro computador de confianza y listo para usar:



Figura 9:

4. Alternativas virtuales de R

Dado el caso que dispongas de un equipo que no cuente con los recursos necesarios para la instalación local de **R** o simplemente no desees realizar la instalación por que prefieres trabajar en la web, existen varias alternativas que te servirán para ejecutar tus códigos de manera online, algunos de estos son: **Rstudio cloud**, **Rdrr**, **jdoodle**, **Coding Ground**.

■ *Rstudio cloud:*

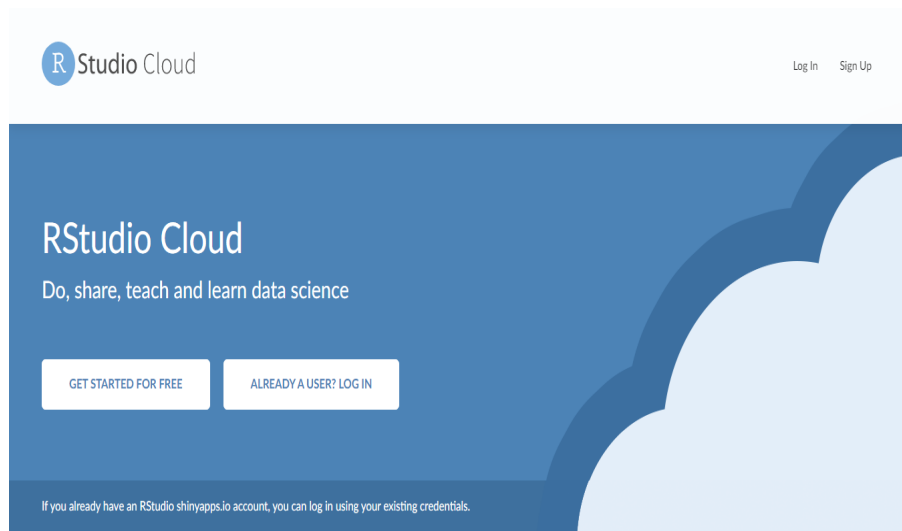


Figura 10: Vista principal de Rstudio cloud.

RStudio Cloud es una plataforma online que permite crear, ejecutar y compartir proyectos realizados en el lenguaje R, esta plataforma es de paga aunque posee también una opción gratuita (funciones un poco restringida con relación al número de proyectos que puedes realizar). Para acceder a esta plataforma se debe tener una cuenta, si ya la tienes solo sería iniciar sesión y empezar, de lo contrario te puedes registrar siguiendo los siguientes pasos:

1. Desde su navegador de preferencia ingrese a Rstudio cloud pinchando el siguiente link (<https://rstudio.cloud/>).
2. Luego se debe hacer click el botón Sign Up que aparece en la parte superior derecha de la pantalla.
3. Seleccione la opción **Cloud Free o cloud Premium** según prefieras
4. Luego debes hacer clic en el botón Sign Up que aparece en la parte inferior de la pantalla
5. Por ultimo solo sería completar el formulario y seguir las instrucciones de registro.

Después de esto estarías listo para usar **Rstudio Cloud**.

■ Rdr:

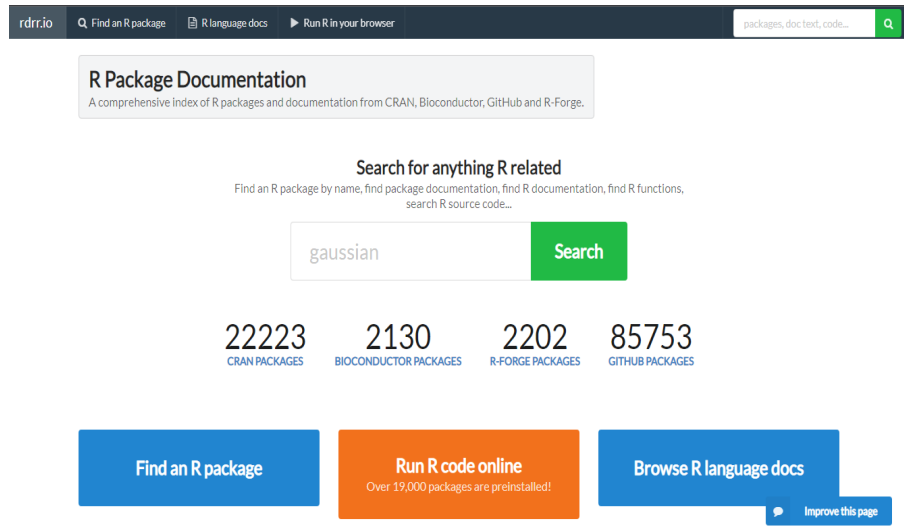


Figura 11: Vista principal de RDRR.

Rdr es al igual que Rstudio cloud una plataforma que permite crear y ejecutar códigos realizados el lenguaje R y además es completamente gratuita, dentro del plataforma se pueden encontrar más de diecinueve mil paquetes R precargados y bibliografía sobre el uso de los paquetes, el único inconveniente de esta plataforma es que no permite trabajar de manera colaborativa. Para acceder no se requiere tener una cuenta y basta con seguir las siguientes instrucciones:

1. Primero pinchar el siguiente link (<https://rdr.io/>), que te dirigirá a la pagina principal de la plataforma.
2. Después eliges una de las tres opciones:
 - a) Find an R package.
 - b) Run R code online.
 - c) Browse R language docs.

Luego de seleccionar la opción deseada y estarías listo para usar la plataforma.

■ Jdoodle:

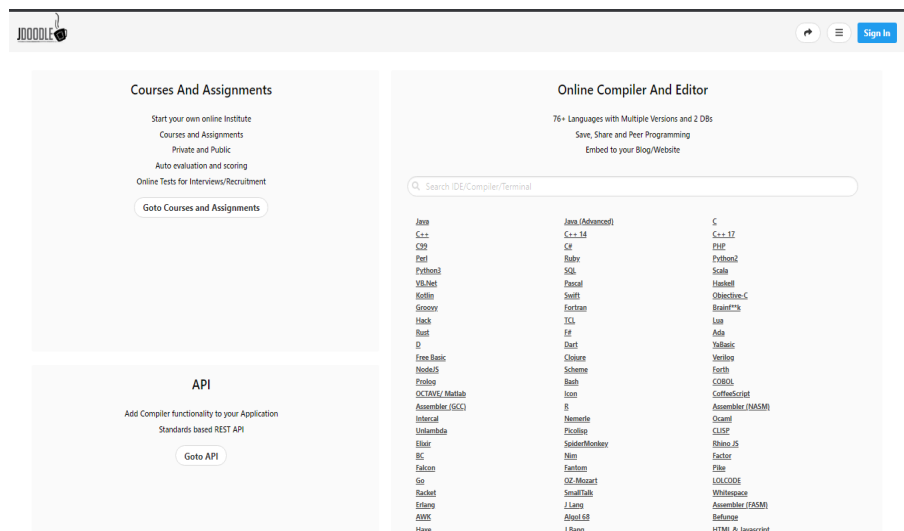


Figura 12: Vista principal de Jdoodle.

Jdoodle es un compilador online con el cual puedes ejecutar código de múltiples lenguajes de programación como c, c++, java, entre otros, incluyendo el lenguaje que nos interesa que en este caso es R. En esta plataforma puedes realizar códigos sin necesidad de registrarte o también si lo de seas puedes hacerlo. Para hacer uso de esta plataforma hay que seguir los siguientes pasos:

1. Primero hay que acceder a la pagina principal, la cual la puedes encontrar dando clic en el siguiente link(<https://www.jdoodle.com/>)
2. En segundo lugar solo seria seleccionar el lenguaje R y comenzar a programar, pero si deseas registrarte puede seguir los siguientes pasos:
 - a) Primeramente se debe hacer clic en el botón **Sign Ln** que aparece en las parte superior derecha de la pantalla.
 - b) Luego seleccionar una de las opciones de registro que ofrece la plataforma las cuales son:
 - 1) sing in with google
 - 2) sing in whith Microsoft
 - c) por ultimo regresar al primer paso

Seguidamente ya estarían listos para programar en R dentro de esta plataforma.

■ *Coding Ground:*



Figura 13: Vista principal de Coding Ground.

Condng Ground al igual que Jdoodle es un compilador online con el cual puedes ejecutar código de múltiples lenguajes de programación incluido R. En esta plataforma también se pueden realizar códigos sin necesidad de registrarse o si lo de seas puedes hacerlo. Para hacer uso de esta plataforma puedes seguir los siguientes pasos:

1. En primer lugar hay que acceder a la pagina principal, la cual la puedes encontrar dando clic en el siguiente link(<https://www.tutorialspoint.com/codingground.htm>)
2. En segundo lugar solo seria seleccionar el lenguaje R y comenzar a programar, pero si deseas registrarte puede seguir los siguientes pasos:
 - a) Primeramente se debe hacer clic en el botón **Signup for Free** que aparece en las parte superior derecha de la pantalla en el centro de la pantalla
 - b) Luego rellenar el formulario que aparece.
 - c) por ultimo regresar al primer paso

Las opciones de compilación de R online aquí expuestas son solo unas de las tantas opciones que se pueden encontrar en la web para correr vuestros códigos de R

5. Estructuras de Datos Básicos de R

6. Estructuras de Datos Básicos de R

Las estructuras más primitivas dentro de **R** son: vectores, matrices y data frames. A continuación se encontraran las definiciones de cada una de estas y su forma de uso.

- **Vectores:** Un vector es la estructura de datos más sencilla en R. Un vector es una colección de uno o más datos del mismo tipo.

1. Propiedades de los vectores

Todos los vectores tienen tres propiedades:

- Tipo:** Un vector es del mismo tipo que los datos que contiene. Si tenemos un vector que contiene datos numéricos, entonces el vector también será de este tipo. Los vectores son atómicos, porque solo pueden contener datos de un tipo, no se pueden mezclar datos de diferentes tipos dentro de ellos.
- Largo:** Es el número de elementos que contiene un vector. El largo es la única dimensión que tiene esta estructura de datos.
- Atributos:** Los vectores pueden tener metadatos de muchos tipos, los cuales describen características de los datos que contienen. Todos ellos son incluidos en esta propiedad.

2. Creación de vectores:

- En R los vectores se crean usando la función `c()` (combinar). Llamamos esta función y le damos como argumento los elementos que deseamos combinar en un vector, separados por comas.

```
> # vector Numérico
> c(1,2,3,4,5,4,3,2,1)
[1] 1 2 3 4 5 4 3 2 1
>
> # vector de cadena de texto
> c("gato","perro","lobo","vaca","leon")
[1] "gato" "perro" "lobo"  "vaca" "leon"
>
> #vector logico
> c(FALSE,TRUE,FALSE,FALSE,TRUE)
[1] FALSE TRUE FALSE FALSE TRUE
>
```

Figura 14:

- Si deseamos agregar un elemento a un vector ya existente, podemos hacerlo combinando nuestro vector original con los elementos nuevos y asignando el resultado a nuestro vector original.

```
> vector<- c(FALSE,TRUE,FALSE)
> vector <-c(vector,TRUE,TRUE)
> vector
[1] FALSE TRUE FALSE TRUE TRUE
> |
```

Figura 15:

- También podemos crear vectores que son combinación de vectores.


```

> vec1<- c(0,2,4,6)
> vec2<-c(1,3,5,7)
> vec3<-c(vec1,vec2)
> vec3
[1] 0 2 4 6 1 3 5 7

```

Figura 16:

- Si intentamos combinar datos de diferentes tipos en un mismo vector, R realizará coerción automáticamente. El vector resultante será del tipo más flexible entre los datos que contenga, siguiendo las reglas de coerción.

```

> vec<- c(0,2,4,6)
> class(vec1)
[1] "numeric"
> NuevoVec<-c(vec,"E")
> class(NuevoVec)
[1] "character"

```

Figura 17:

- se pueden crear vectores de secuencias numéricas usando `:`. De un lado de los dos puntos escribimos el número de inicio de la secuencia y del otro el final. Las secuencias creadas con `:` son consecutivas con incrementos o decrementos de 1. Estas secuencias pueden empezar con cualquier número, incluso si este es negativo o tiene cifras decimales

```

> v<-c(1:10)
> v
[1] 1 2 3 4 5 6 7 8 9 10

```

Figura 18:

3. Operaciones con vectores: Existen algunas operaciones al aplicarlas a un vector, se aplican a cada uno de sus elementos. A este proceso le llamamos vectorización. Las operaciones aritméticas y relacionales pueden vectorizarse. Si las aplicamos a un vector, la operación se realizará para cada uno de los elementos que contiene

```

> vector1<-c(1:10)
> #suma
> vector1
[1] 1 2 3 4 5 6 7 8 9 10
> vector1 + 2
[1] 3 4 5 6 7 8 9 10 11 12
> #Multiplicacion
> vector2<-c(1:10)
> vector2
[1] 1 2 3 4 5 6 7 8 9 10
> vector2 * 2
[1] 2 4 6 8 10 12 14 16 18 20
> #modulo
> vector3<-c(1:11)
> vector3
[1] 1 2 3 4 5 6 7 8 9 10 11
> vector3 %% 2
[1] 1 0 1 0 1 0 1 0 1 0 1

```

Figura 19:

- **Matrices:** Las matrices pueden ser descritas como vectores multidimensionales. Al igual que un vector, únicamente pueden contener datos de un sólo tipo, pero además de largo, tienen más dimensiones.

En un sentido estricto, las matrices son un caso especial de un array, que se distingue por tener específicamente dos dimensiones, un “largo” y un “alto”. Las matrices son, por lo tanto, una estructura con forma rectangular, con renglones y columnas.

Como las matrices son usadas de manera regular en matemáticas y estadística, es una estructura de datos de uso común en R común.

1. Creación de Matrices

- Creamos matrices en R con la función `matrix()`. La función `matrix()` acepta dos argumentos, `nrow` y `ncol`. Con ellos especificamos el número de renglones y columnas que tendrá nuestra matriz.
- Los datos que intentemos agrupar en una matriz serán acomodados en orden, de arriba a abajo, y de izquierda a derecha, hasta formar un rectángulo. *Colocar un ejemplo*
- Si multiplicamos el número de renglones por el número de columnas, obtendremos el número de celdas de la matriz.
- Ejemplo:

```

R 4.1.2 · ~/
> # dos renglones y cuatro columnas
> Ma = matrix(1:8, nrow = 2, ncol = 4)
> print(Ma)
      [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8
> |

```

Figura 20:

- **Data Frames:** Los data frames son estructuras de datos de dos dimensiones (rectangulares) que pueden contener datos de diferentes tipos, por lo tanto, son heterogéneas. Esta estructura de datos es la más usada para realizar análisis de datos

1. Se puede entender a los data frames como una versión más flexible de una matriz. Mientras que en una matriz todas las celdas deben contener datos del mismo tipo, los renglones de un data frame admiten datos de distintos tipos, pero sus columnas conservan la restricción de contener datos de un sólo tipo. Un data frame está compuesto por vectores. Debido a esto podemos asignar un nombre a cada vector, que se convertirá en el nombre de la columna. Como todos los nombres, es recomendable que este sea claro, no ambiguo y descriptivo

```
> mi_dataframe <- data.frame(  
+   "enteros" = 1:4,  
+   "Letra" = c("a", "b", "c", "d"),  
+   "decimales" = c(1.2, 3.4, 4.5, 5.6),  
+   "cadena" = as.character(c("a", "b", "c", "d"))  
+ )  
> print(mi_dataframe)  
  enteros Letra decimales cadena  
1       1    a       1.2     a  
2       2    b       3.4     b  
3       3    c       4.5     c  
4       4    d       5.6     d  
> |
```

Figura 21:

2. propiedades de los Data Frames

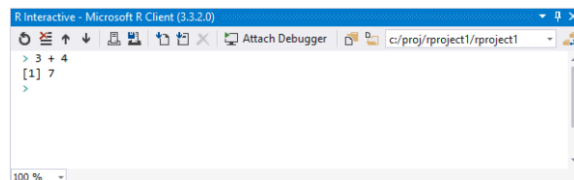
- a) Al igual que con una matriz, si aplicamos una operación aritmética a un data frame, esta se vectorizará. Los resultados que obtendremos dependerán del tipo de datos de cada columna. R nos devolverá todas las advertencias que ocurran como resultado de las operaciones realizadas, por ejemplo, aquellas que hayan requerido una coerción.

```
R 4.1.2 ~ /  
> mi_dataframe <- data.frame(  
+   "enteros" = 1:4,  
+   "Letra" = c("a", "b", "c", "d"),  
+   "decimales" = c(1.2, 3.4, 4.5, 5.6),  
+   "cadena" = as.character(c("a", "b", "c", "d"))  
+ )  
> mi_dataframe * 2  
Error in FUN(left, right) : non-numeric argument to binary operator
```

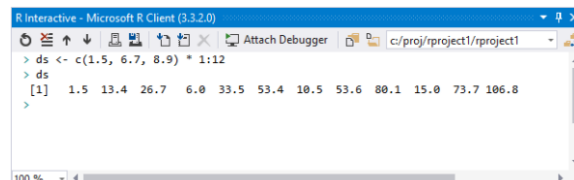
Figura 22:

7. Visualización de datos en R

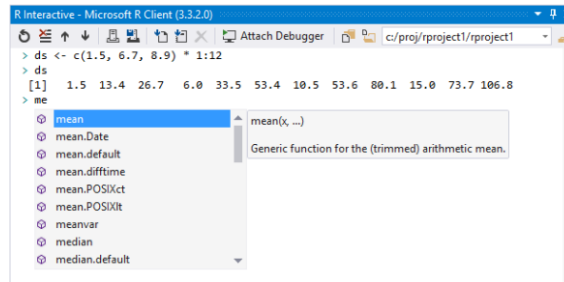
Compruebe que la ventana R interactivo funciona escribiendo $3 + 4$ y luego presionando **Entrar** para ver el resultado:



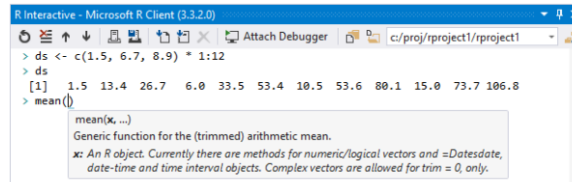
Escriba algo un poco más complicado, `ds <- -c(1.5, 6.7, 8.9) * 1:12`, y luego escriba `ds` para ver el resultado:



Escriba `mean(ds)` pero tenga en cuenta que tan pronto como escriba `m` o `me`, Visual Studio IntelliSense proporciona opciones de Autocompletar. Cuando la opción de Autocompletar que quiera esté seleccionada en la lista, presione **Tab** para insertarla:

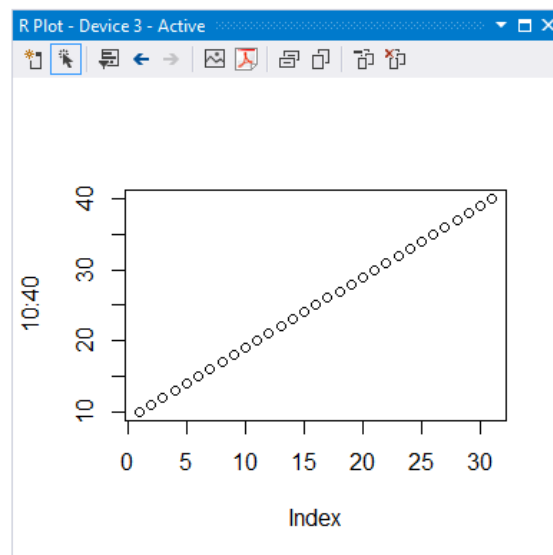


Después de completar **mean**, escriba el paréntesis de apertura (y observe cómo IntelliSense proporciona ayuda en línea para la función:



Complete la línea **mean(ds)** y presione Entrar para ver el resultado ([1] 39.51667).

Algunos comandos, como **plot(1:100)**, abren una nueva ventana en Visual Studio cuando los resultados no se pueden mostrar directamente en la ventana interactiva:



La ventana interactiva también permite revisar el historial, cargar y guardar áreas de trabajo, adjuntar e interactuar con los archivos de código fuente en lugar de usar las operaciones.

8. Referencias

- Vega, J. B. M. (n.d.). R para principiantes. [Bookdown.Org](https://bookdown.org/jboscomendoza/r-principiantes4/matrices-y-arrays.html). Retrieved June 10, 2022, from <https://bookdown.org/jboscomendoza/r-principiantes4/matrices-y-arrays.html>
- Vega, J. B. M. (n.d.). R para principiantes. [Bookdown.Org](https://bookdown.org/jboscomendoza/r-principiantes4/vectores.html). Retrieved June 10, 2022, from <https://bookdown.org/jboscomendoza/r-principiantes4/vectores.html>
- Vega, J. B. M. (n.d.). R para principiantes. [Bookdown.Org](https://bookdown.org/jboscomendoza/r-principiantes4/data-frames.html). Retrieved June 10, 2022, from <https://bookdown.org/jboscomendoza/r-principiantes4/data-frames.html>

- Tutorial de introducción a R - Visual Studio (Windows). (n.d.). [Microsoft.com](https://docs.microsoft.com/es-es/visualstudio/rtvs/getting-started-with-r?view=vs-2017). Retrieved June 12, 2022, from <https://docs.microsoft.com/es-es/visualstudio/rtvs/getting-started-with-r?view=vs-2017>
- (N.d.). [Tutorialspoint.Com](https://www.tutorialspoint.com/r/r_tutorial.pdf). Retrieved June 12, 2022, from https://www.tutorialspoint.com/r/r_tutorial.pdf
- R-4.2.0 for Windows. (n.d.). [R-Project.Org](https://cran.r-project.org/bin/windows/base/). Retrieved June 12, 2022, from <https://cran.r-project.org/bin/windows/base/>