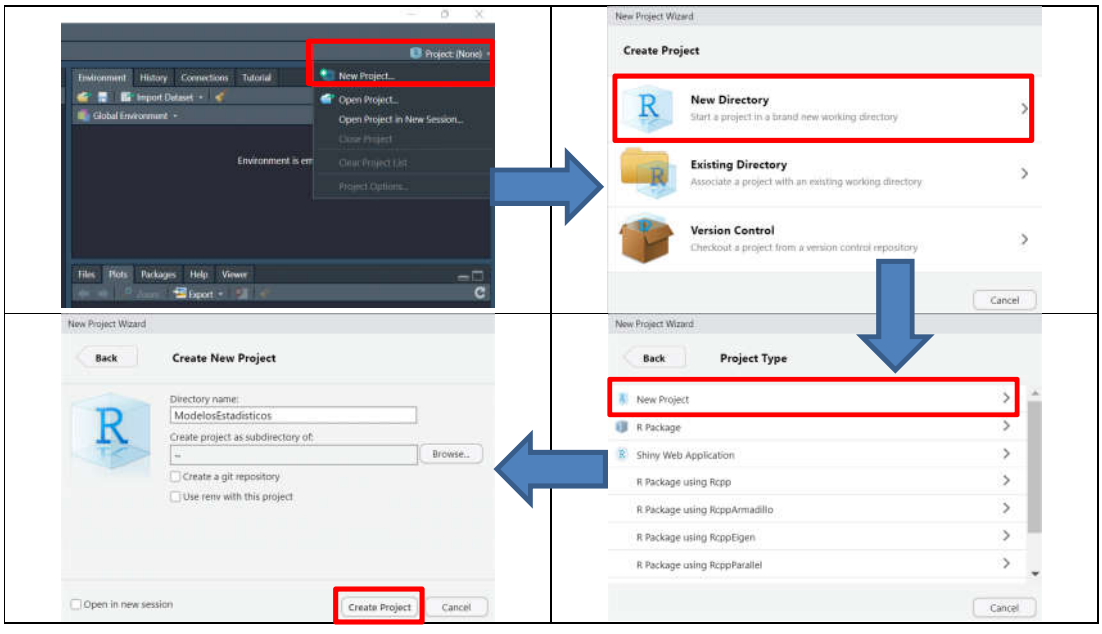
Introducción a R con RStudio

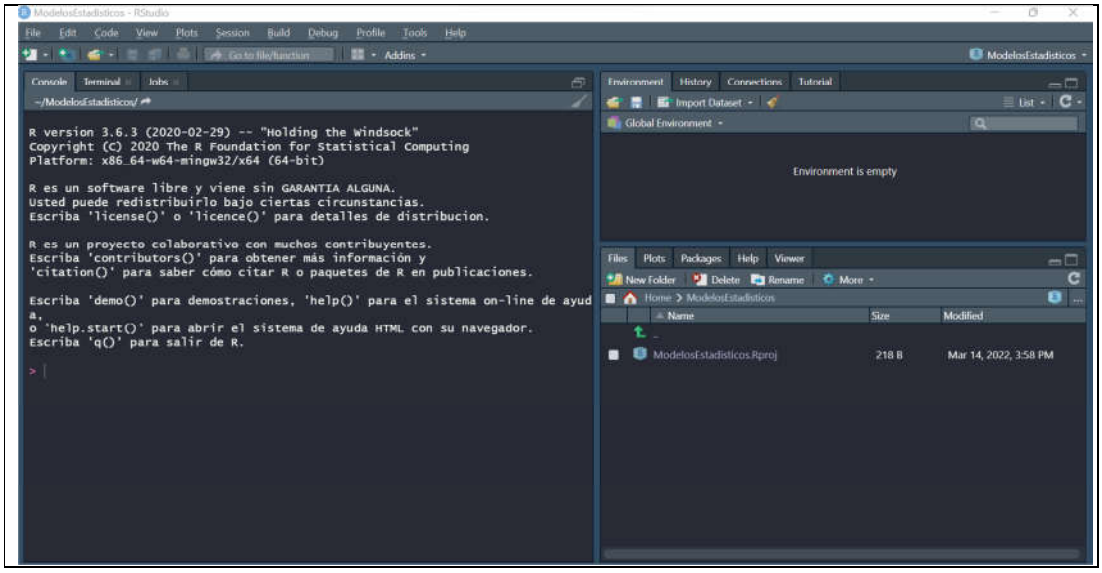
Crear un proyecto en RStudio

Cuando uno inicia el estudio o la solución a un problema con RStudio, generalmente va a tener los scripts donde escribe el código general, puede tener scripts con funciones, archivos con los datos y probablemente tenga gráficas o tablas de resultados de la ejecución. Toda esta información, se podría guardar en una carpeta, aunque al generar más archivos de código, modificar los existentes o agregar más datos o resultados, puede volverse algo desordenado.

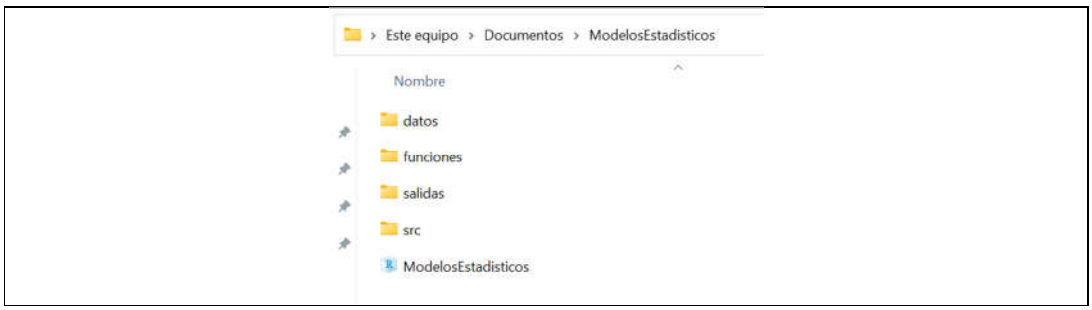
Para solucionar esto, vamos a trabajar generando un proyecto en RStudio. A partir de él, vamos a tener por defecto la ruta de nuestro proyecto como Directorio de Trabajo. Esto nos va a facilitar el acceso a los archivos que utilicemos (datos, funciones, etc.) y para guardar nuestros resultados.

Una vez seleccionado la opción “New Project…”, en la ventana que se abre elegimos New Directory, New Project, el nombre de la carpeta del proyecto y la ubicación de la misma.





Una vez que tenemos creado el proyecto, vamos a generar la estructura de carpetas para trabajar:



En la carpeta del proyecto, al momento de generarlo, se creó el archivo: ***nombredelproyecto.Rproj***. Haciendo doble click sobre él se va a abrir el RStudio e ir directamente al proyecto, donde lo habíamos dejado.

Scrip en R

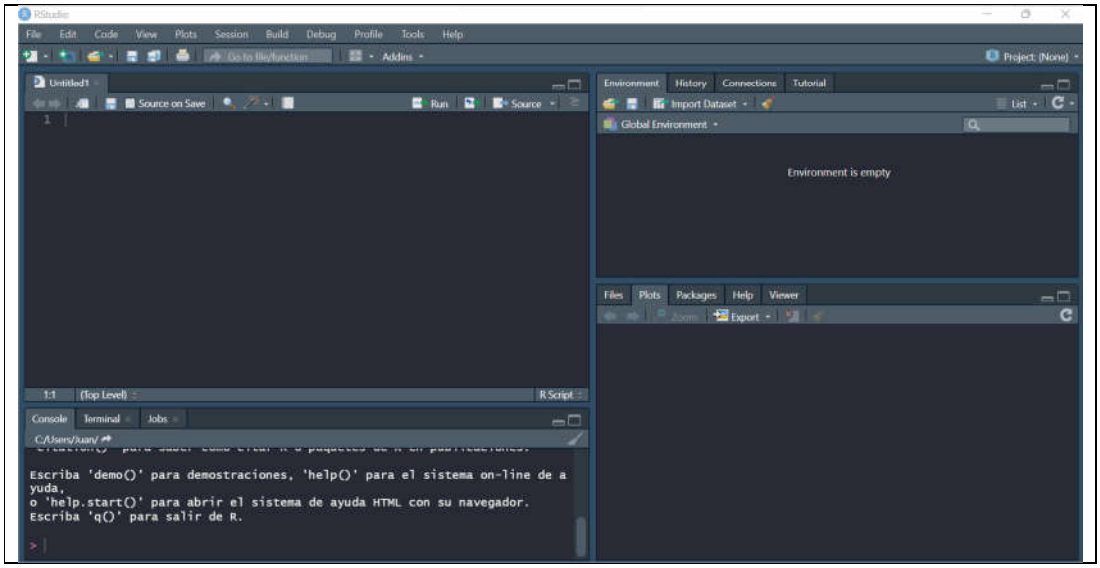
Vamos a tener nuestro código en un archivo de texto *.R, que vamos a poder editar desde RStudio o cualquier editor de texto.

Para generar un nuevo script de R:

Desde **File** → **New File** → **R Script**, o bien con el atajo por teclado: **Ctrl + Shift + N**.

Entorno de trabajo de RStudio

El entorno de desarrollo de RStudio se divide en cuatro partes principales: fuente, consola, entorno y gráficos.



En particular, para ver que objetos tenemos en el entorno de trabajo, usamos la función *ls()* y para eliminar un objeto, la función *rm()*.

Tipos y estructuras de datos en R

Como la mayoría de los lenguajes de programación, R nos permite trabajar con distintos tipos de datos y estructuras de estos. No es la finalidad de esta guía describir en detalle todos ellos, solamente nombraremos los que más vamos a utilizar [1]:

Numéricos: los tipos *integer* y *numeric*, corresponden a los tipos de datos entero y real.

Texto: el tipo de cadena de texto, comprende letras, palabras o frases. Es el tipo *character*. Se declaran entre comillas simples o dobles.

Lógico: datos que solo pueden tomar dos valores, verdadero o falso (TRUE o FALSE). Es el tipo *logical*.

Tipo *factor*: es un tipo de dato específico de R, para representar categorías. Puede ser un número o una cadena, que van a estar etiquetados.

NA y NULL: son dos tipos de datos para casos particulares. *NA* representa un dato faltante por cualquier motivo, mientras que *NULL* aparece cuando se intenta recuperar un dato que no existe.

Vectores: es la estructura más simple de datos en R. Se genera como una concatenación de valores, que pueden ser numéricos o no, pero del mismo tipo. Los valores se pueden acceder por su índice. Un número, es un vector de largo unitario.

Matrices: es una estructura de datos de dos dimensiones. De forma general las creamos con la función específica de R *matrix()*. Al igual que los vectores, sus elementos van a ser todos del mismo tipo.

Data Frames: este tipo de estructura de datos es de dos dimensiones, está compuesto de vectores. Pero nos van a permitir trabajar con distintos tipos de datos, por lo que va a ser uno de los que más vamos a utilizar. Hay distintas formas de generarlos, pero la más utilizada es con la función *data.frame()*.

Listas: son estructuras unidimensionales cómo los vectores. Pero a diferencia de ellos, cada elemento puede ser de un tipo y una estructura distintos. Por lo que podemos tener listas de números, de vectores, matrices, data frames u otras listas. Las creamos con la función *list()*.

Funciones

Nos vamos a referir aquí a las funciones definidas por el usuario, es decir, las funciones que nosotros vamos a crear.

Tienen la siguiente estructura:

nombre <- function(arg$_1$, …, arg$_n$){

operaciones

result <- resultado

}

Si bien podemos crear la función en el script principal de nuestro código, para poder ejecutar paso a paso y probar el funcionamiento, las vamos a guardar en un scrip con el nombre: ***nombrefuncion.R*** dentro de otra carpeta del proyecto.

Para llamarla desde el script principal y tenerla disponible, vamos a usar la función:

source("./carpeta/nombrefuncion.R")

Datos en R

La instalación base de R tiene conjuntos de datos de prueba que pueden utilizarse, del paquete *Datasets*. Ejecutando la función data(), sin argumento, podemos acceder al listado completo.

Para cargar el conjunto de datos, usamos la función mencionada con el nombre:

data(nombre)

Algunos de los más utilizados son: iris, cars, mtcars, airpassengers, entre otros.

También podemos utilizar datos externos, cómo archivos de texto (TXT o CSV) o archivos de Excel, que son los más comúnmente utilizados.

Para leer archivos de texto vamos a utilizar la función *read.csv()* y para archivos de excel, la función *read_excel()* del paquete **readxl** [4].

Para el caso de archivos grandes (incluso de varios gigabytes), es recomendable utilizar la función *fread()* del paquete **data.table** [5] ya que es significativamente más rápida que *read.csv()* del paquete base de R.

De la misma forma que podemos abrir archivos de datos, podemos guardar nuestros resultados. En el caso que sean tablas, podemos hacerlo en archivos de texto o Excel y si fueran gráficas, en archivos de imagen (jpg, png, etc.) [1].

Gráficos en R

La instalación base de R nos va a permitir realizar distintos tipos de gráficos (líneas, barras, puntos, cajas, etc.) y trabajar de forma flexible con distintos parámetros (colores, tipo y grosor de línea, etc.).

La función base para realizar gráficas es *plot()* [3] y a partir de ella, variando los parámetros y tipos de datos vamos a poder cambiar el tipo de gráfico. Además existen funciones específicas para algunos tipos de gráfico, como son: *hist()*, *boxplot()* o *pie()*.

Actividades

1 – Abra el archivo de datos “bank.csv” e indique:

a. A qué tipo de datos corresponde cada variable?

b. Los parámetros estadísticos típicos de las variables numéricas.

c. Contiene datos faltantes en alguna de las variables?

d. Realice una función que tome los datos correspondientes al archivo abierto, seleccione una variable numérica y una categórica (interesantes) y devuelva un nuevo conjunto de datos que contenga estas dos variables.

e. Realice un gráfico completo (título, etiquetas y color) de las variables obtenidas antes y explique la relación.

f. Guarde los resultados (datos y gráfico).

Referencias

1. “R para principiantes”, Juan Bosco Mendoza Vega. Disponible en: https://bookdown.org/jboscomendoza/r-principiantes4/
2. “Colors in R”, Ying Wei. Disponible en: http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf
3. Plot: Generic X-Y Plotting. Disponible en: https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/plot
4. Package readxl. Disponible en: https://cran.r-project.org/web/packages/readxl/readxl.pdf
5. Package data.table. Disponible en: https://cran.r-project.org/web/packages/data.table/data.table.pdf