
MANUAL ESTADÍSTICA Y PROBABILIDAD CON PYTHON Y R

REALIZADO POR:

GLEN NICOLÁS RICO

MAYKOLL GIL

MIGUEL GÓMEZ

Índice

1. Breve introducción a R y Python	2
1.1. Descarga e instalación de R y R-Studio	2
1.1.1. Instalación de R (Si no desea instalar r-studio)	3
1.1.2. Instalación de R-studio	4
1.2. Descarga de Python	6
1.3. Código Fuente de los ejemplos presentados en el manual	8
1.4. El lenguaje R	8
1.4.1. El entorno	9
1.4.2. Sintáxis	9
1.4.3. Definición y declaración de variables	9

Acerca de éste documento

Este documento fue creado para el beneficio de todos los estudiantes que desean utilizar R y Python para entender algunos conceptos de probabilidad y estadística.

Breve introducción a R y Python

En esta sección vamos a preparar el entorno con el que se van a ejecutar todos los ejemplos presentados en este libro, por lo que si ya tiene alguna instalación de R y Python sólo resta verificar la versión, a fecha de este documento las versiones estables que utilizaremos son:

- R 3.5.3
- R-studio 1.2.5019
- Python 3.8.

en este manual no vamos a profundizar en el lenguaje, por lo que una versión compatible con python 3 es suficiente para los ejemplos que vamos a ver aquí.

1.1. Descarga e instalación de R y R-Studio

R es un lenguaje y entorno de de código libre bajo la licencia GNU y R Studio no esde código abierto pero es una herramienta muy útil para el manejo de R, y es por ello que lo instalaremos. Estos son los links de descarga

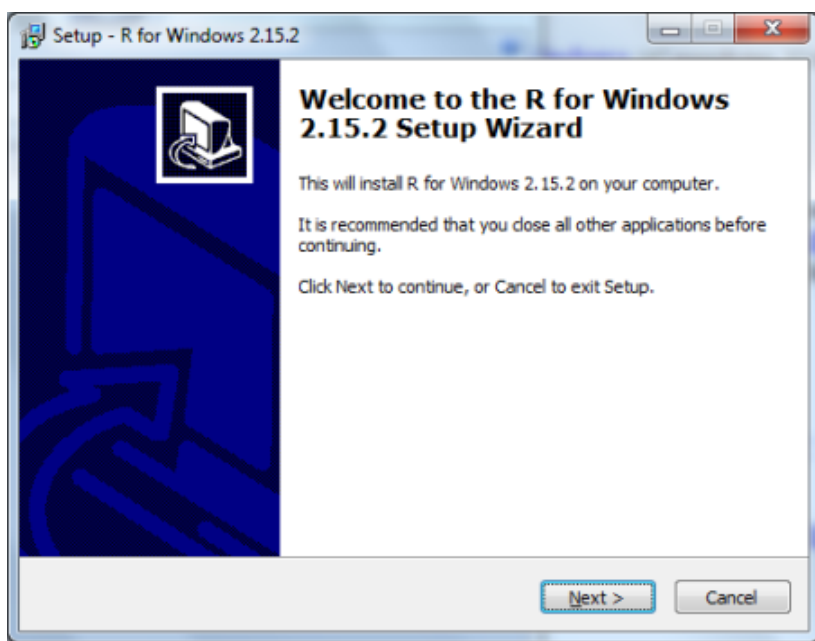
- Para descargar únicamente R ir a <https://www.r-project.org/>, allí se encuentran los enlaces de descarga de R, se recomienda descargar la versión más estable si hasta ahora se está iniciando en R.
- Si se quiere llevar a cabo una instalación completa junto con R, R-studio es la mejor opción, para descargarlo ir a <https://rstudio.com/> allí se encuentra el link que redirige a

la versión Free. Usualmente ocurre que se debe elegir la versión de R que se va instalar dependiendo del sistema operativo, en este manual, sin embargo si lleva tiempo utilizando cualquier sistema operativo encontrará bastante sencilla su instalación.

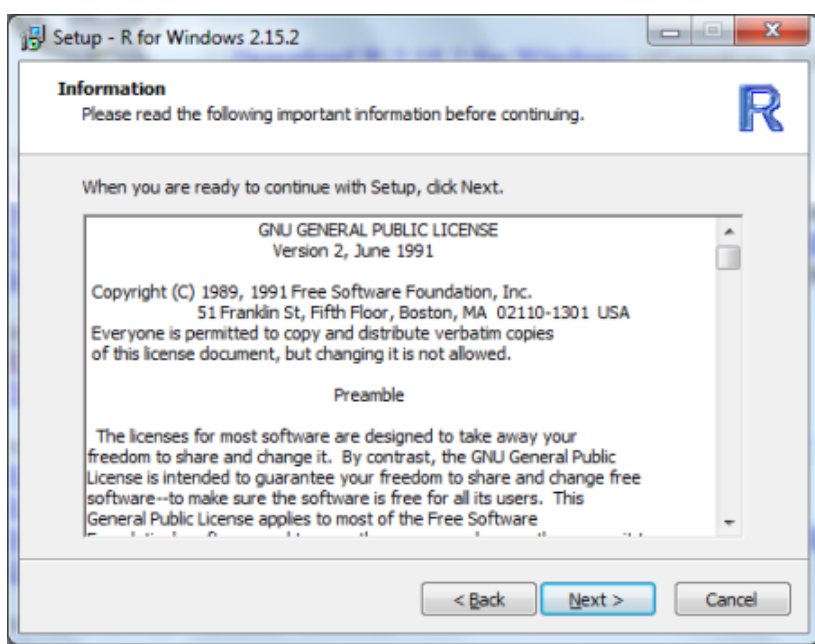
Nota: La única versión que no se recomienda descargar en cualquier caso es la descarga del código fuente, si la descarga deberá compilar el código desde cero (*from scratch*), es mucho más complicado, sólo se recomienda para usuarios avanzados.

1.1.1. Instalación de R (Si no desea instalar r-studio)

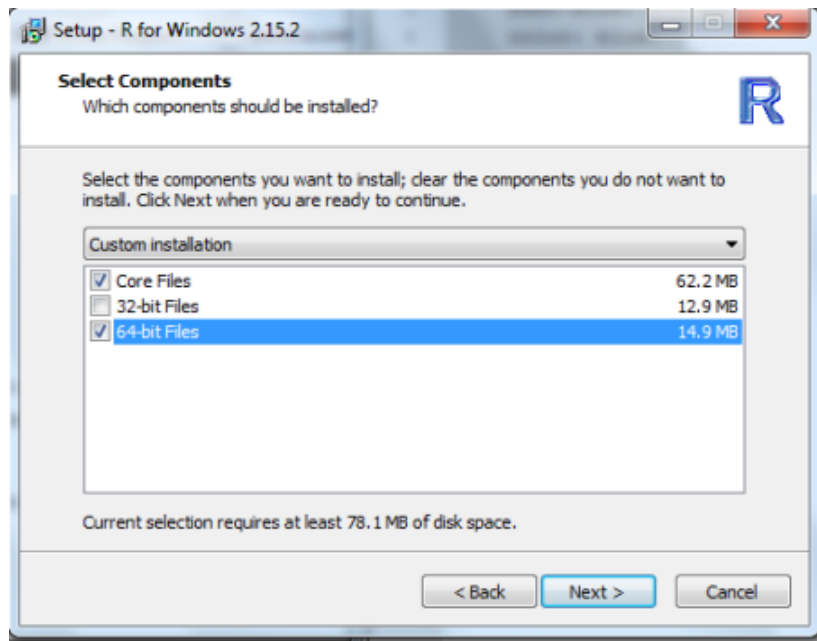
La instalación es muy sencilla, primero ejecute el .exe descargado en la página de R.



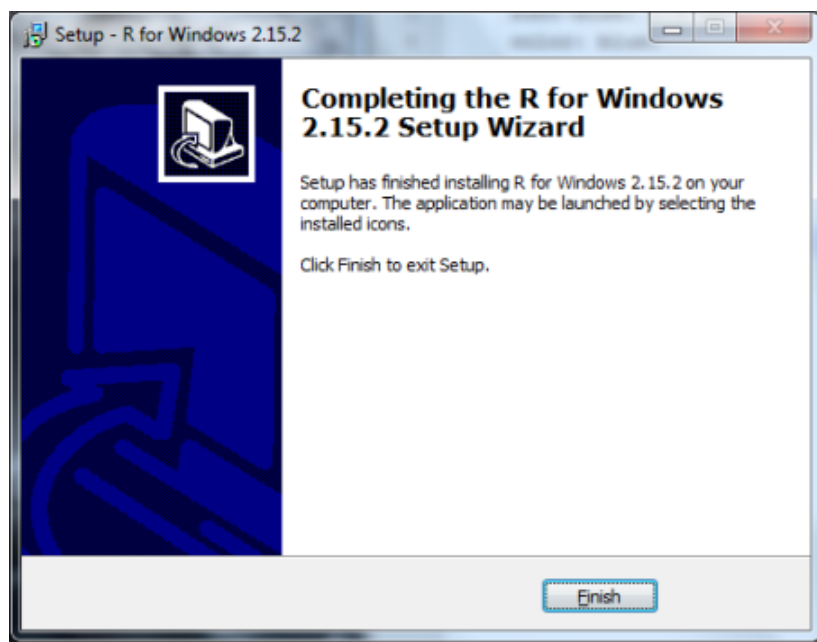
Al dar siguiente estamos aceptando los términos de uso de la licencia GNU



, dependiendo de la arquitectura de su computador deberá elegir entre un tipo de instalación, usualmente el valor por defecto del instalador tiene la elección que requiere



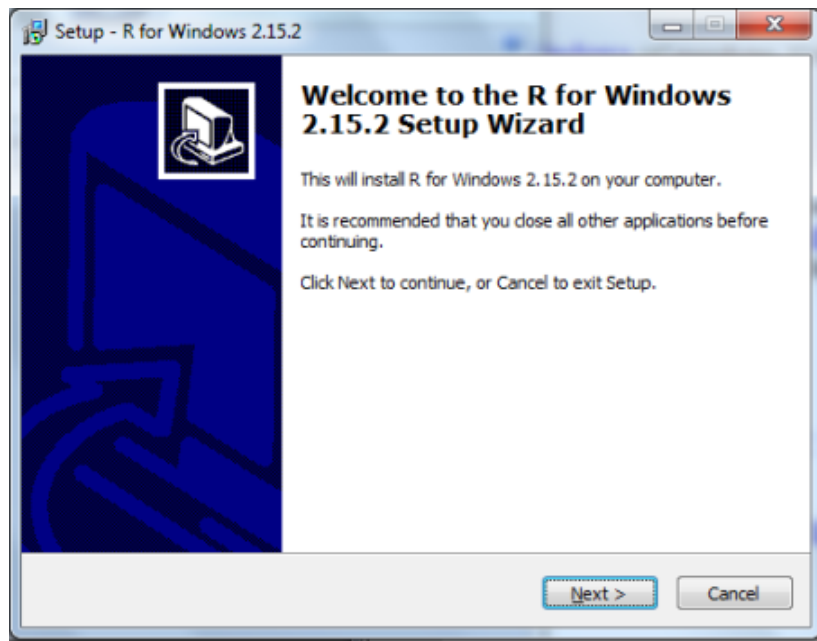
En los siguientes pasos daremos siguiente, eventualmente llegaremos a



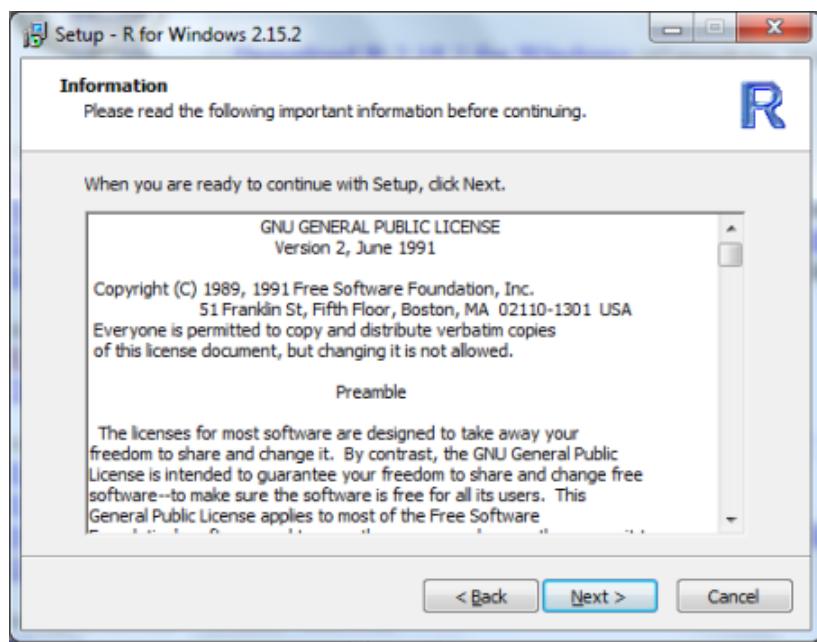
Al finalizar, ya tendremos instalado R.

1.1.2. Instalación de R-studio

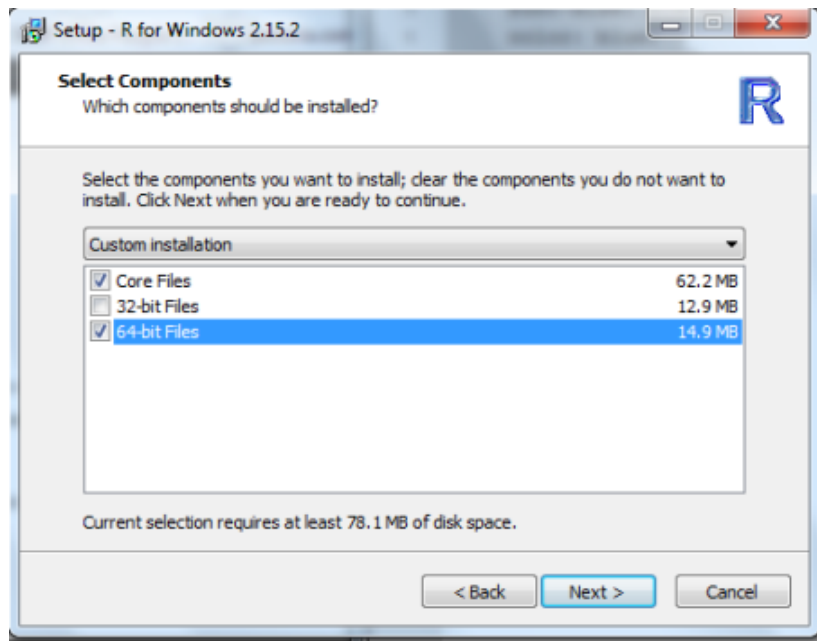
La instalación es muy sencilla, primero ejecute el .exe descargado en la página de R.



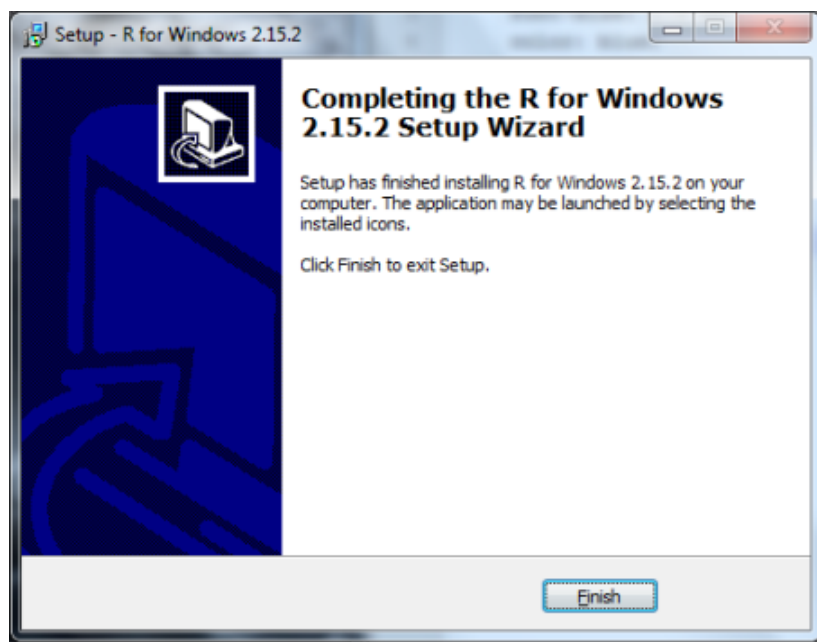
Al dar siguiente estamos aceptando los términos de uso de la licencia GNU



, dependiendo de la arquitectura de su computador deberá elegir entre un tipo de instalación, usualmente el valor por defecto del instalador tiene la elección que requiere



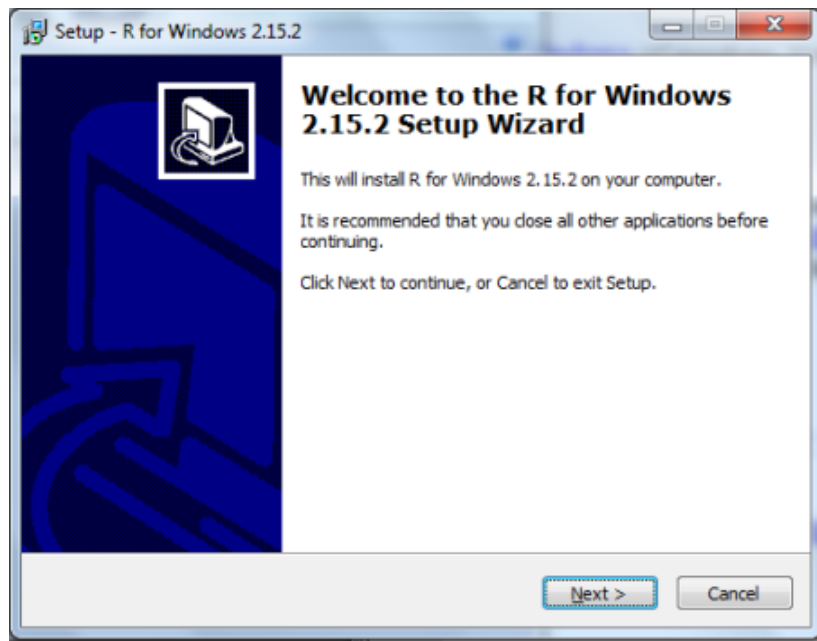
En los siguientes pasos daremos siguiente, eventualmente llegaremos a



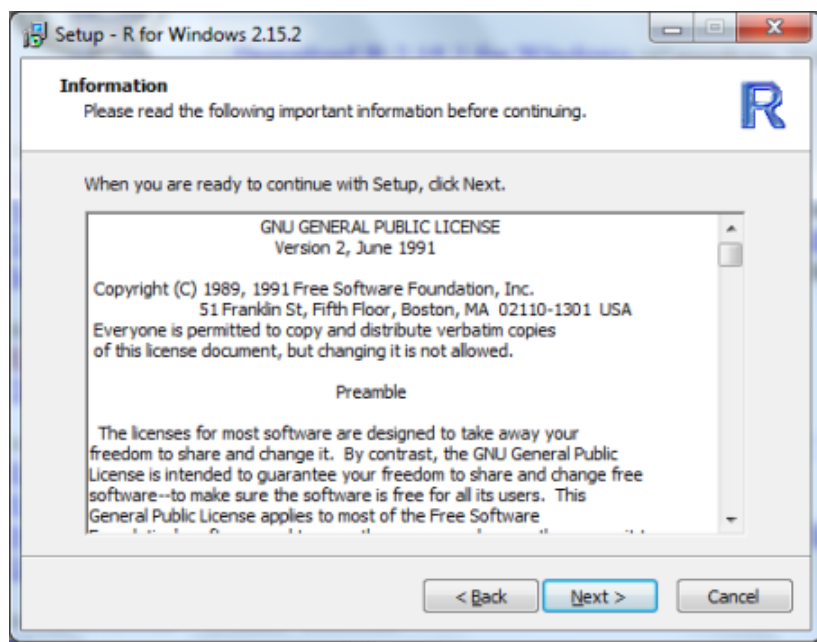
Al finalizar, ya tendremos instalado R.

1.2. Descarga de Python

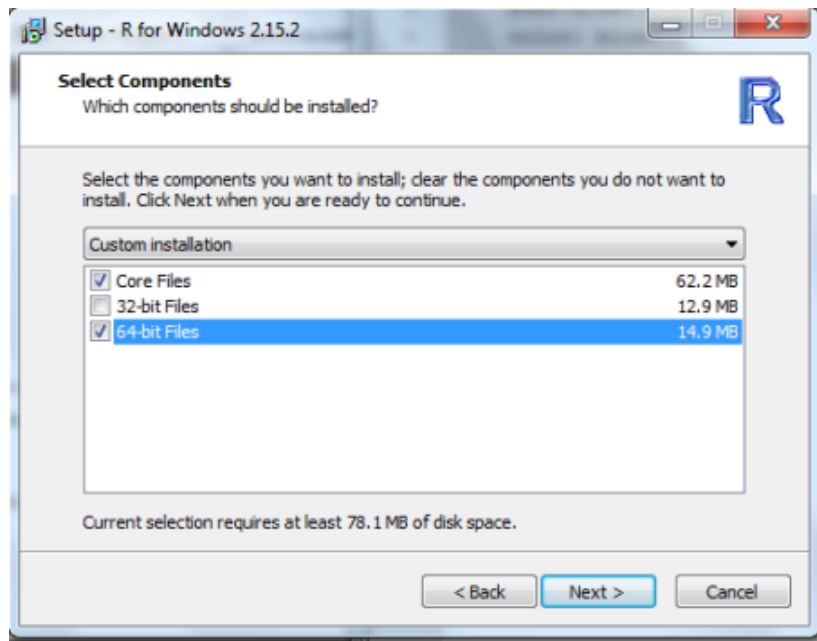
La instalación es muy sencilla, primero ejecute el .exe descargado en la página de R.



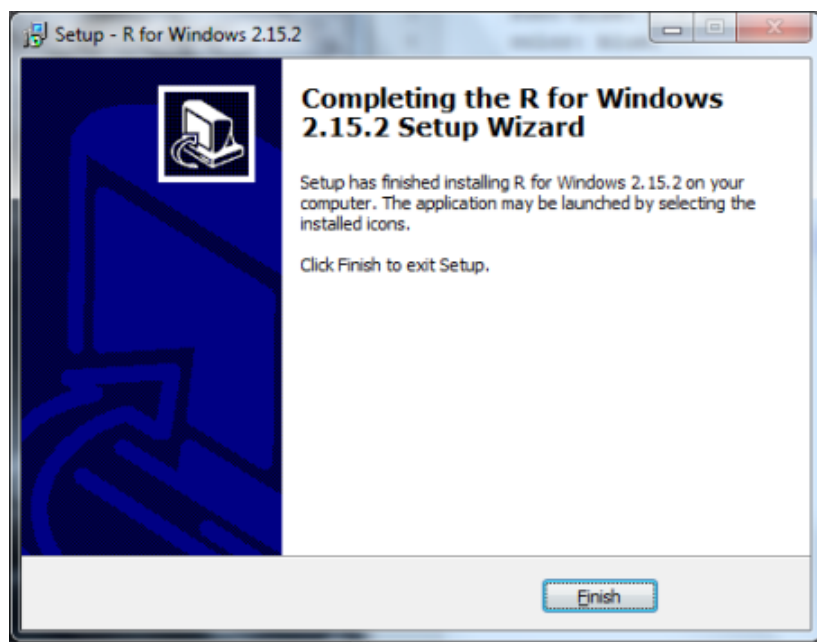
Al dar siguiente estamos aceptando los términos de uso de la licencia GNU



, dependiendo de la arquitectura de su computador deberá elegir entre un tipo de instalación, usualmente el valor por defecto del instalador tiene la elección que requiere



En los siguientes pasos daremos siguiente, eventualmente llegaremos a



Al finalizar, ya tendremos instalado R.

1.3. Código Fuente de los ejemplos presentados en el manual

Los fuentes de la construcción de este libro así como los ejemplos se encuentran en [LINK DE GITHUB](#).

1.4. El lenguaje R

El lenguaje R es un lenguaje y entorno para el desarrollo de aplicaciones estadísticas, computacionales y gráficas.

R ofrece un amplio conjunto de herramientas estadísticas (para modelamiento, pruebas de hipótesis), herramientas gráficas y es altamente extensible, actualmente es uno de los proyectos de software libre más grandes del mundo, lo cual es una de sus fortalezas ya que al ser un proyecto de código abierto (en el que cualquiera puede aportar) toda la comunidad que utiliza el software se beneficia de las contribuciones de todos, haciendo que R sea una herramienta en crecimiento y de uso extendido.

Este lenguaje tiene soporte para diferentes sistemas operativos.

1.4.1. El entorno

Cuando decimos que R también es un entorno, es debido a que no solamente tiene las funciones que normalmente encontramos en un lenguaje de programación sino que también tiene un amplio abanico en crecimiento de 'paquetes'¹ para manipulación de texto, carga de archivos que a su vez pueden ser extendidas. R ha sido escrito de una manera muy similar a los lenguajes de programación interpretados con el rendimiento del lenguaje de programación C.

1.4.2. Sintaxis

Como todo lenguaje debemos comprender algunas reglas básicas, a éstas reglas las conocemos como la sintaxis del lenguaje, dentro de esto encontramos varias categorías:

- **Definición y declaración de variables.** En todo lenguaje declaramos variables por lo que aprenderemos a definir variables junto con los tipos de datos en un programa R.
- **Operadores.** Repasaremos las operaciones básicas en R.
- **Estructuras de control.** Aquí ubicamos a las estructuras condicionales, condiciones en donde evaluamos el valor de verdad de una sentencia que permite alterar el flujo de un programa.
- **Estructuras de repetición.** Aquí se encuentran las estructuras que nos permiten iterar, por ejemplo recorrer un archivo.
- **Funciones.** Son piezas de código, que nos ayudan a organizar el código. En general son una muy buena práctica de programación.

1.4.3. Definición y declaración de variables

Primero repasaremos sobre los tipos de datos en R. R utiliza el paradigma de la orientación a objetos al límite, cada elemento en R es un objeto. R tiene 6 tipos de datos:

- **caracter.** "a", "b", "hola"
- **numérico.** 2, 3, 3.14
- **entero.** 28L, 3L. el caracter L le indica a R que almacene el dato como un entero.

¹Que son componentes adicionales que se pueden instalar según sea necesario.

- lógico. TRUE, FALSE.
- complejo. $3+14i$. En este manual no utilizaremos este tipo de datos para los problemas que manejamos es suficiente con los reales.

Ejemplo:

```
x <- "Hola"
x
```

Salida:

```
[1] "Hola"
```

De esta forma acabamos de declarar una variable en R. Nótese que no fue necesario llevar a cabo una declaración del tipo de variable, R se encarga de asignar el tipo de dato dependiendo de lo que recibe. Por ejemplo:

```
x <- 1
x
```

Salida:

```
[1] 1
```

Adicionalmente R también se encuentra provisto de funciones que nos permiten identificar propiedades de los objetos que declaramos:

- `class()`. Nos permite obtener la clase a la que está asociada a un objeto.
- `typeof()`. Nos permite obtener el tipo de dato de una variable.
- `length()`. Nos permite obtener la longitud de un tipo de dato.
- `attributes()`. nos permite conocer si un objeto tiene determinados metadatos.

Ejemplo:

```
x <- 1L
class(x)
typeof(x)
length(x)
attributes(x)
```

Salida:

```
class(x)
[1] "integer"
typeof(x)
[1] "integer"
length(x)
[1] 1
attributes(x)
[1] NULL
```

Cuando declaramos el tipo de dato sabíamos que era un entero, es por ello que obtenemos como tipo de dato y clase `integer`, como solo hemos ocupado 1 byte para almacenar el valor '1' obtenemos como valor de la función 1. Finalmente, `x` es un valor escalar, es por ello que no posee atributos y obtenemos el valor `NULL`, que indica que no tiene ningún atributo.

R cuenta con estructuras de datos para contener colecciones, nombraremos algunas de ellas:

- vectores atómicos.
- listas.
- matrices.
- data frame.
- factores.

En aplicaciones estadísticas encontraremos que es muy frecuente el uso de este tipo de estructuras. Los vectores se pueden declarar con varias funciones, las más comunes son `vector`, `logical`, `character`, `integer`, `numeric`.

Ejemplo:

```
vector()
vector("character", length = 5)
```

Salida

```
vector()
[1] logical(0)
vector("character", length = 5)
[1] "" "" "" "" ""
```

Con las funciones anteriores podemos analizar estos objetos.

Matrices las matrices son una extensión de los vectores pero tienen varias mas de una dimensión. Ejemplo:

```
m <- matrix(nrow = 2, ncol = 2)
m
```

Salida:

```
[,1] [,2]
[1,] NA NA
[2,] NA NA
```

Listas Las listas en R actúan como contenedores. A diferencia de los vectores atómicos las listas pueden contener diferentes tipos de datos, por ejemplo:

```
x <- list(1, "hola", TRUE, 3+14i)
```

x

salida:

```
[[1]]
```

```
[1] 1
```

```
[[2]]
```

```
[1] "hola"
```

```
[[3]]
```

```
[1] TRUE
```

```
[[4]]
```

```
[1] 3 + 14i
```