

Introducción a R y RStudio

Angela Di Serio

Contents

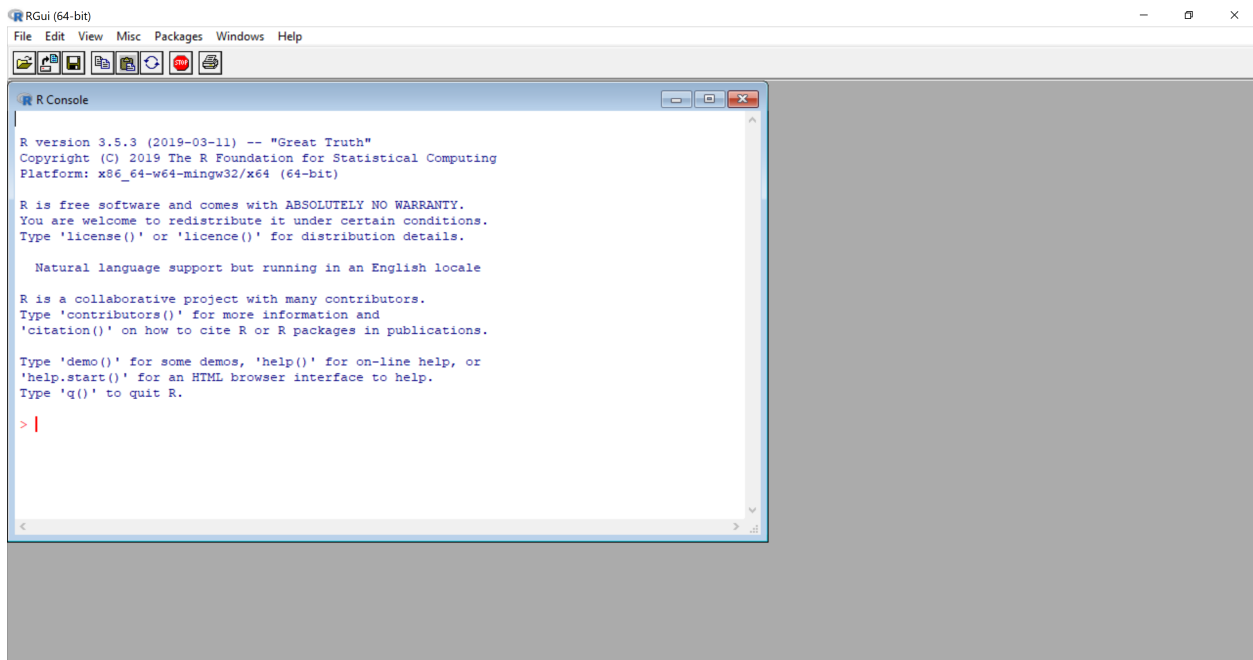
1.	1
Ambiente de R	2
2.	2
2.1 Entorno de trabajo RStudio	3
2.2 Paneles	3
Console	4
Source	5
Environment/History	7
Environment	7
History	8
Connections	8
Tutorial	9
Files/Plots/Packages/Help/Viewer	10
Files	10
Plots	11
Packages	12
2.3 Paquetes/Packages en R	13

1.

- R es un lenguaje de programación y un entorno para el análisis estadístico y gráfico.
- R es software libre y gratuito.
- Es multiplataforma disponible para Windows, Macintosh y GNU/Linux.
- Permite un manejo y almacenamiento efectivo de los datos.
- Conjunto de operadores para la realización de cálculos con matrices.

- Gran colección de herramientas para el análisis de datos.
- Utilidades gráficas para la visualización de datos.
- Lenguaje de programación bien desarrollado que incluye saltos condicionales, bucles, funciones recursivas, utilidades para la entrada y salida de datos, etc.

Ambiente de R



La interfaz de R no es muy amigable ni versátil. Los resultados de un análisis no se basan en recordar la sucesión de clics, sino en una serie de comandos escritos.

Si deseas rehacer un análisis porque se recopilieron más datos, no tienes que recordar en qué botón se hizo clic y en qué orden; solo tienes que ejecutar el **script** o guión de nuevo.

2. R Studio

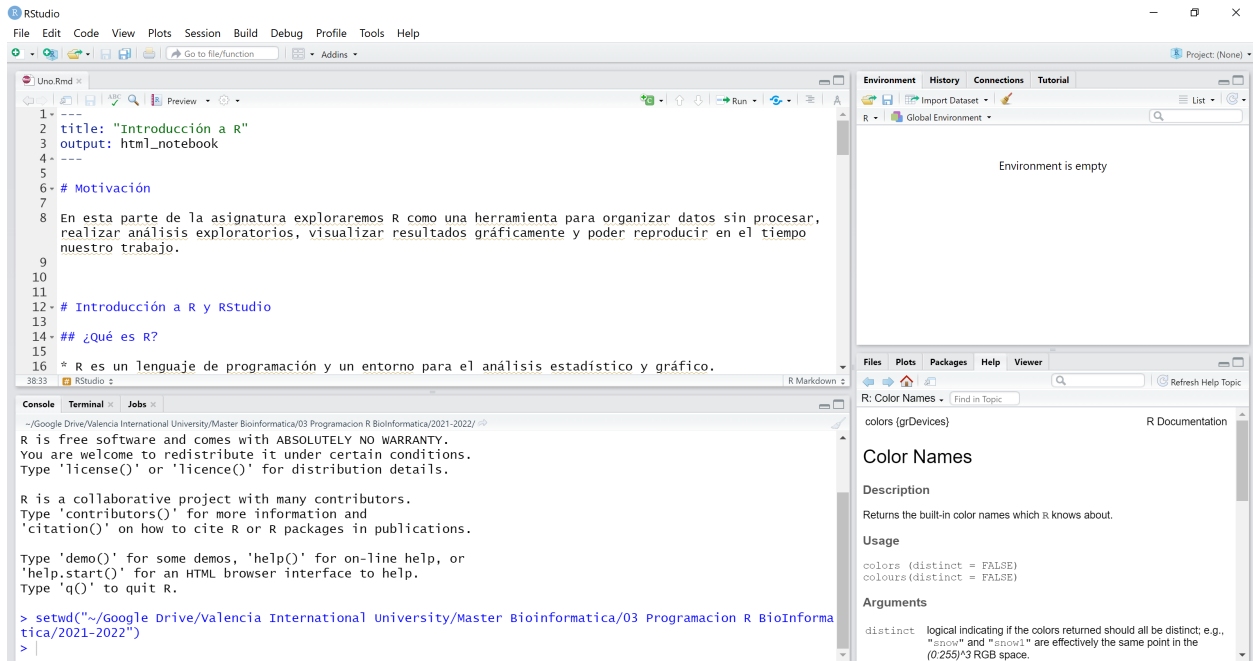
- Es un programa que permite interactuar con R de forma más amigable, además de facilitar muchas de las tareas de programación y análisis de datos en R.

- Es un entorno de desarrollo integrado (IDE) para R.
- Tiene un formato de documentación basado en LaTeX, que se utiliza para proporcionar documentación completa tanto en formato físico como digital.

En el escritorio virtual de **Amazon WorkSpaces** tenéis disponible Rstudio pero si deseas instalar R y RStudio en tu computador personal puedes hacerlo fácilmente instalando primero la versión adecuada de R dependiendo de tu plataforma, y posteriormente instalar RStudio.

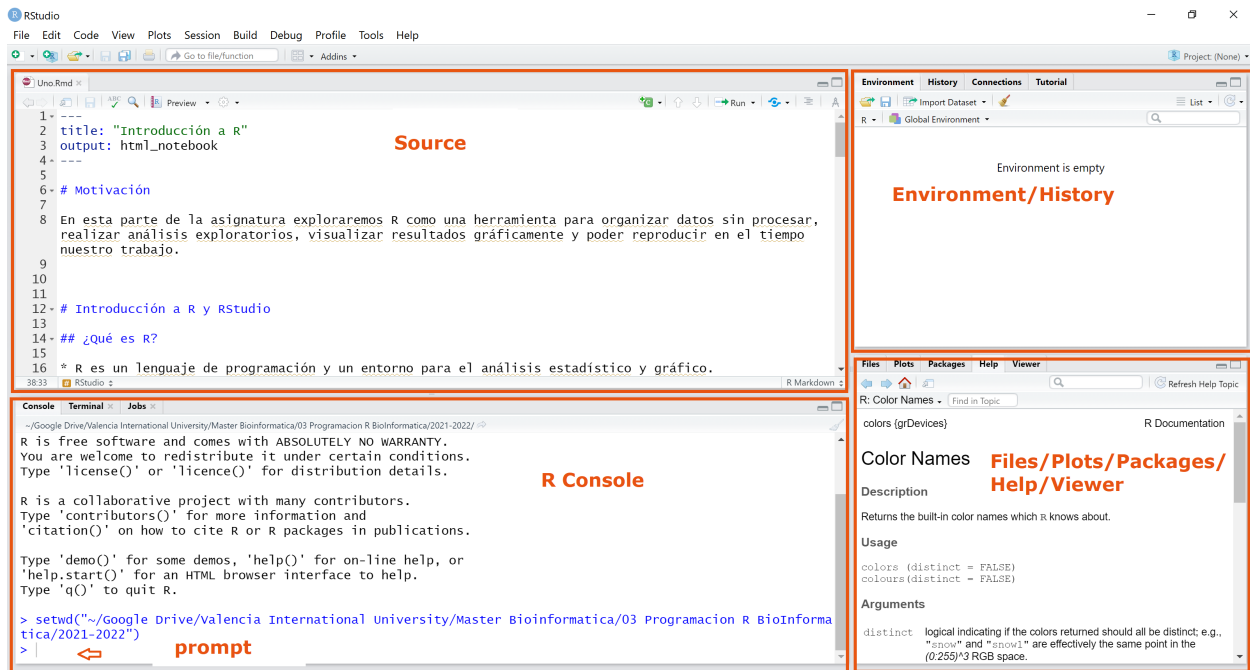
2.1 Entorno de trabajo RStudio

Cuando abrimos RStudio veremos algo parecido a:



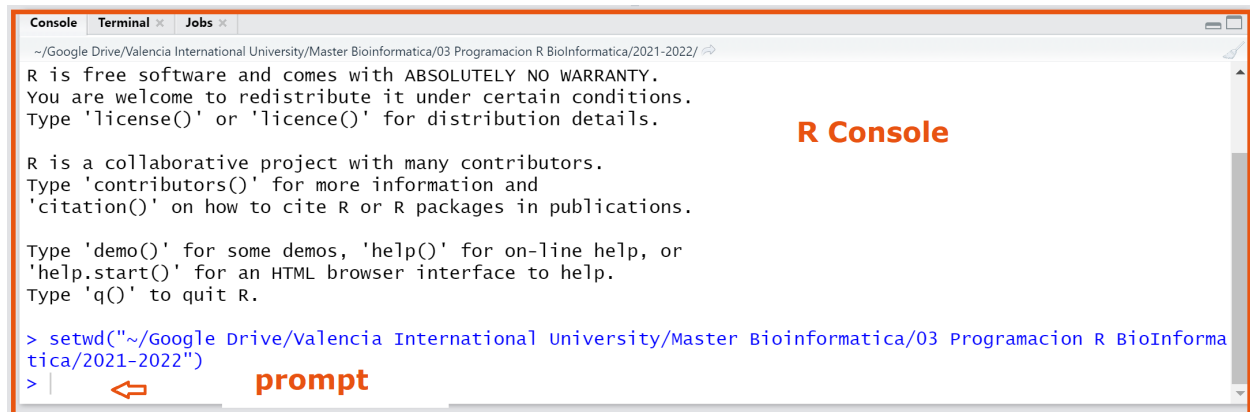
2.2 Paneles

RStudio está (normalmente) dividido en 4 paneles



Console

Por defecto, la consola se encuentra en el panel inferior-izquierdo. Inmediatamente debajo aparece un texto informativo y, finalmente, el símbolo “>”.



Aquí es donde R espera que le demos instrucciones. Para ejecutarlas y obtener el resultado pulsamos enter

```

2+4
2*(2*3)^3
125/5
  
```

```

2+4
## [1] 6
2*(2*3)^3
  
```

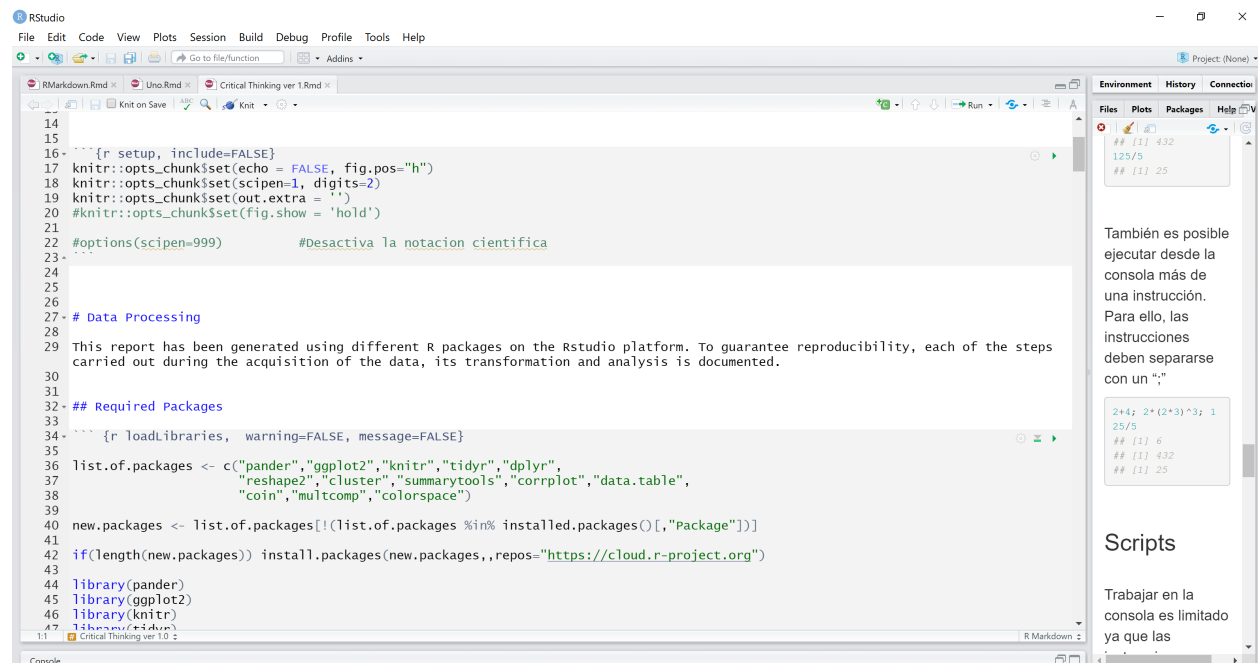
```
## [1] 432
125/5
## [1] 25
```

También es posible ejecutar desde la consola más de una instrucción. Para ello, las instrucciones deben separarse con un “;”

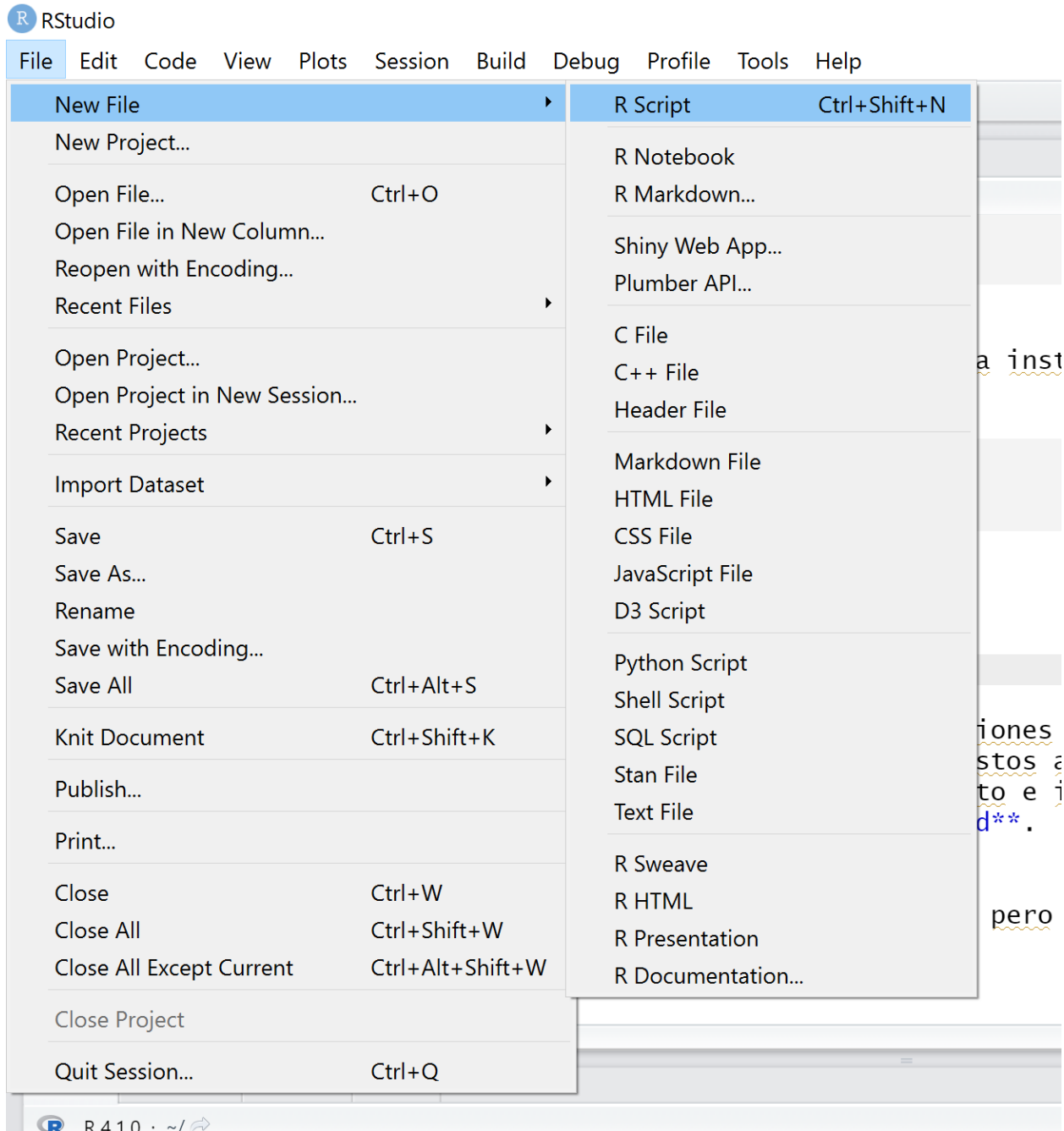
```
2+4; 2*(2*3)^3; 125/5
## [1] 6
## [1] 432
## [1] 25
```

Source

Trabajar en la consola es limitado ya que las instrucciones se han de introducir una a una. Lo habitual es trabajar con **scripts** (guiones) o archivos de instrucciones. Estos archivos tienen extensión **.R**. También hay otro tipo de archivo en donde podemos combinar texto e instrucciones que nos permitirán documentar nuestro trabajo. Estos archivos tienen extensión **.Rmd**.



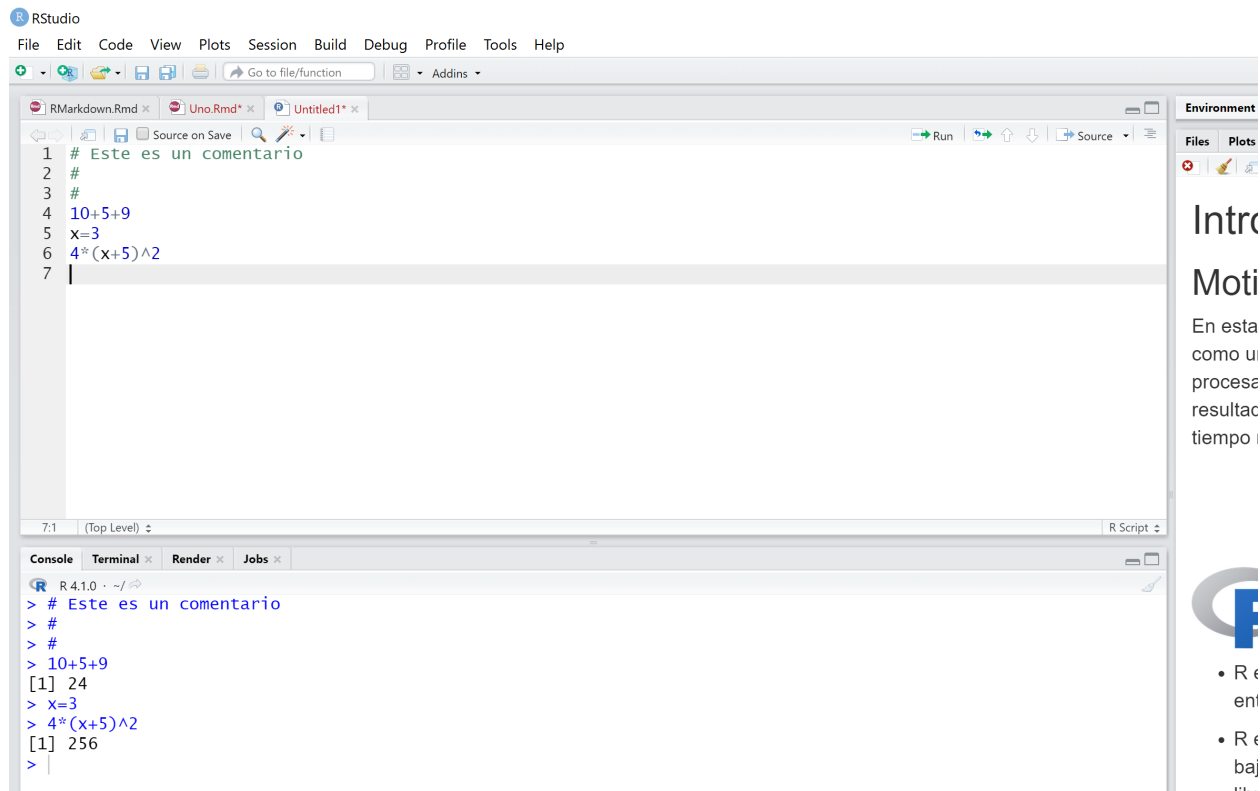
Se puede crear un script con cualquier editor de texto pero RStudio incluye su propio editor de texto.



El **panel source** se sitúa en la parte superior-izquierda de RStudio. Ahora podemos escribir las instrucciones línea por línea. Las instrucciones pueden ser ejecutadas una a una o podemos seleccionar un grupo de instrucciones y ejecutarlas en bloque.

Para ejecutar las instrucciones tenemos varias alternativas:

- Clic en el botón: Run (botón situado en la parte derecha de las opciones del panel)
- Pulsar Ctrl+Enter
- Desde las opciones del menú Code







Environment/History



Compuesto de cuatro pestañas:

- Environment
- History
- Connections
- Tutorial






Environment En **Environment** se muestran los objetos que se van creando en la sesión de trabajo. También tenemos la opción de cargar y guardar una sesión de trabajo, importar datos y limpiar los objetos de la sesión.

Environment History Connections Tutorial

  Import Dataset -  172 MiB - 

R -  Global Environment - 

Data

audio	Formal class Wave	
AudioRec	Formal class Wave	
df	42497 obs. of 2 variables	
df_frecuencias	1181 obs. of 3 variables	
frecuencias	4186 obs. of 3 variables	

Values

d	num [1:42497]	362 30 -767 19 -753 ...
delta	1556.5	
distorsion	190775.281584582	
dq	num [1:42497]	362 778 -778 778 -778 ...
entropiaDifIzq	8.2214124510885	
entropiaIzq	10.9226852839361	
i	42497L	
k	2	
M	2	
Rango	3113L	
SNR	4.03361533383295	

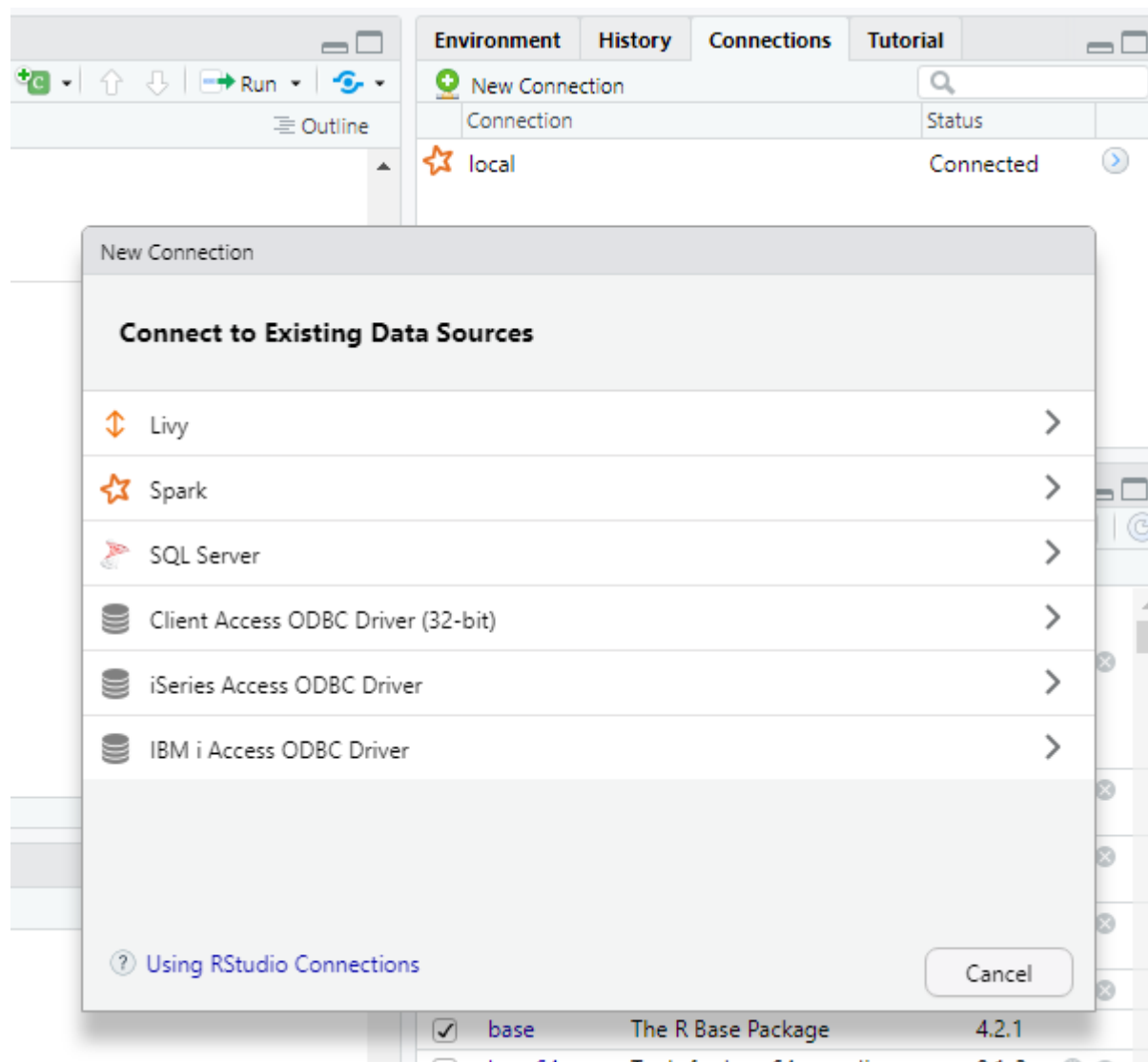
History En **History** se registran las instrucciones ejecutadas. Como opciones, podemos cargar y guardar el historial de la sesión, seleccionar una o más instrucciones y enviarlas a la consola o al script, y limpiar el historial.

```

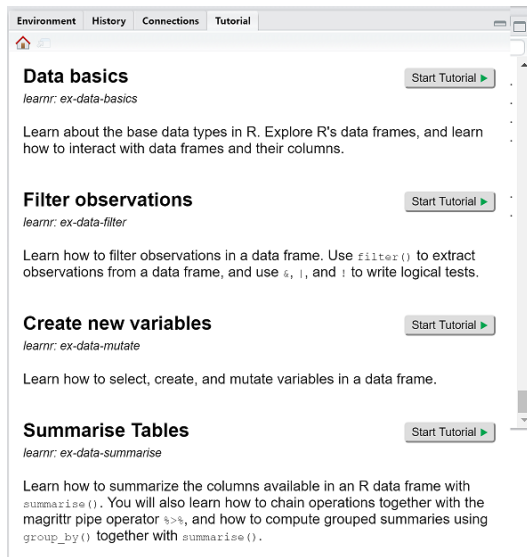
Environment History Connections Tutorial
To Console To Source
distorsion[k] = sum((x-xqr)^2)/length(x)
plot(1:length(x),xqr,type="l",xlab="Tiempo",ylab="Ampli...
main=paste("Codificación Diferencial y cuantificada con "...
plot(1:length(x),x,type="l",xlab="Tiempo",ylab="Amplitu...
main=paste("Codificación Diferencial y cuantificada con "...
lines(1:length(x),xqr,type="l",col="blue",lty=2)
lines(1:length(x),x-xqr,type="l",col="green",lty=3)
legend("bottomright",legend=c("Señal","Reconstruccion","E...
AudioRec= normalize(Wave(left=xqr,samp.rate=44000,bit=16)...
# play(AudioRec)
k=k+1
}
x=()
x=c()
for (i in c(1:10)) { x[i]= i +3}
x
library(bookdown)
sessionInfo()
install.packages("learnr")
install.packages("learnr")

```

Connections **Connections** permite la conexión a una variedad de orígenes de datos y explorar los objetos y datos dentro de la conexión. Está diseñado para trabajar con una variedad de bases de datos en R.

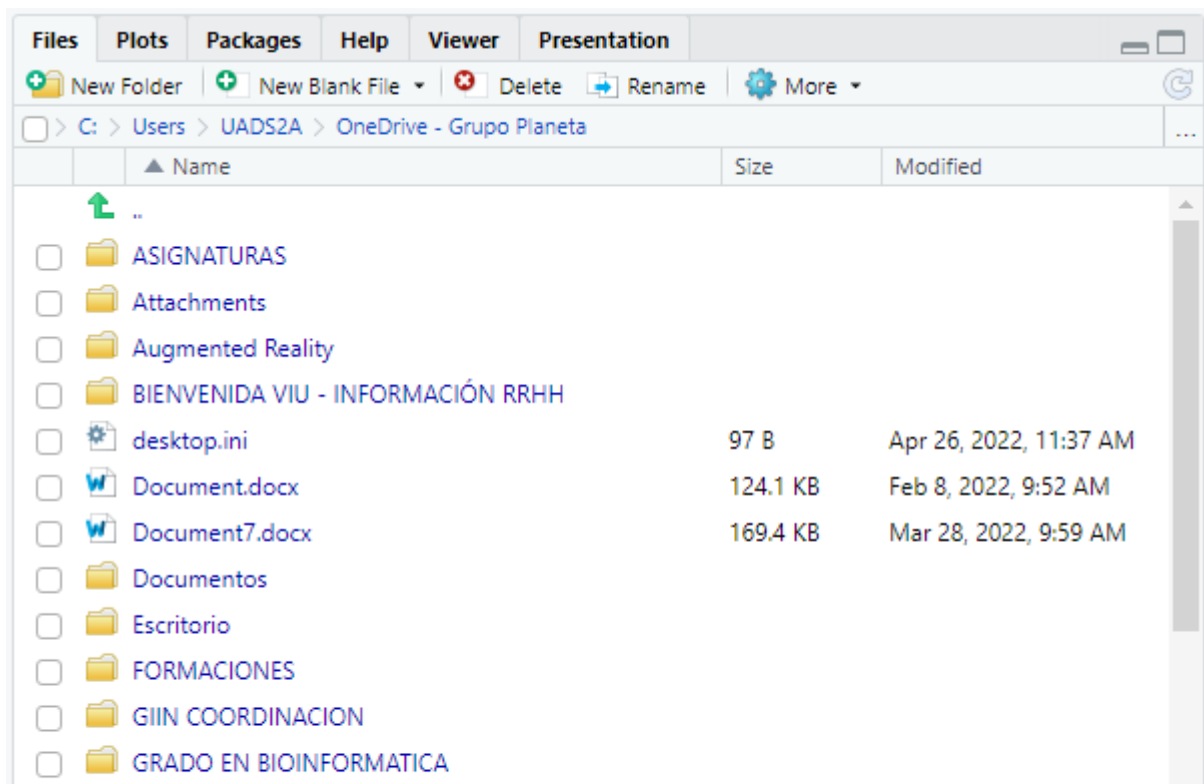


Tutorial **Tutorial** nos provee tutoriales diversos sobre aspectos de R.

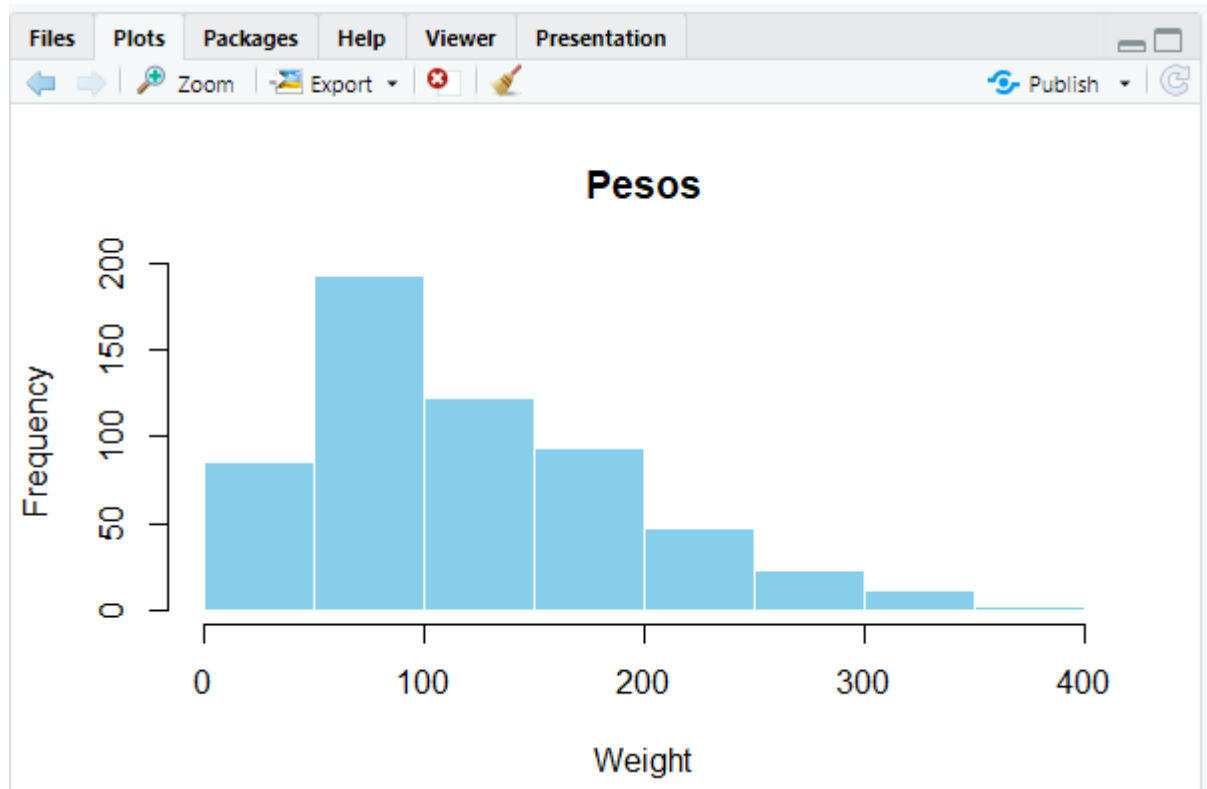


Files/Plots/Packages/Help/Viewer

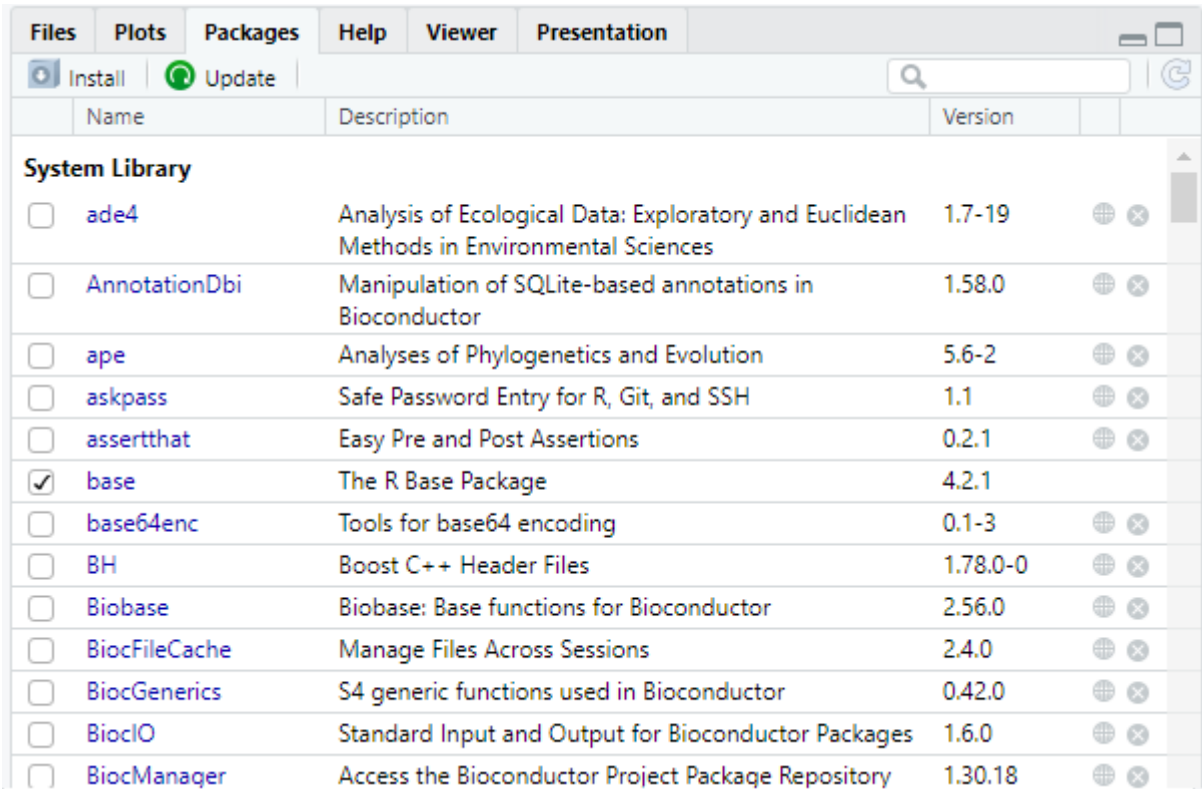
Files El panel **Files** permite el acceso a la estructura de directorios y archivos del disco duro. Se puede usar para configurar el directorio de trabajo: una vez que navega a una carpeta que desea leer y guardar archivos, haga clic en **More** y luego en **Set As Working Directory**. También se puede establecer el directorio de trabajo a través de una función que veremos más adelante.



Plots Muestra los gráficos que se han generado. Hay botones para abrir el gráfico en una ventana separada y exportarlo como pdf o jpeg (aunque también se puede hacer usando funciones).



Packages Muestra una lista de todos los paquetes R instalados en el disco duro e indica si están o no cargados actualmente. Los paquetes que están cargados en la sesión actual están marcados, mientras que los que están instalados pero aún no cargados están desmarcados.



2.3 Paquetes/Packages en R

La instalación básica de R viene equipada con múltiples funciones que nos permitirán la importación de datos, transformaciones de datos, ajuste y evaluación de modelos estadísticos, representaciones gráficas.

Sin embargo, la potencia y versatilidad de R radica en su capacidad de incorporar en cualquier momento nuevas funciones capaces de realizar nuevas tareas.

Un paquete (package) es una colección de funciones, datos y código en R que se almacenan en una carpeta conforme a una estructura bien definida, fácilmente accesible para R.

En la web de R puede consultar la lista de paquetes disponibles. En octubre de 2023 esta lista incluye algo más de 19550 paquetes. Asimismo en la sección Task Views se puede consultar una lista de paquetes ordenada según áreas de aplicación.

Cuando instalamos R se incorporan por defecto numerosos paquetes. Podemos ver una lista de los paquetes que actualmente tenemos instalados en nuestro ordenador ejecutando la función **library()**

```

Packages in library 'C:/Users/UADS2A/AppData/Local/Programs/R/R-4.2.1/library':

ade4                Analysis of Ecological Data: Exploratory and
                    Euclidean Methods in Environmental Sciences
AnnotationDbi       Manipulation of SQLite-based annotations in
                    Bioconductor
ape                 Analyses of Phylogenetics and Evolution
askpass             Safe Password Entry for R, Git, and SSH
assertthat          Easy Pre and Post Assertions
base                The R Base Package
base64enc           Tools for base64 encoding
BH                  Boost C++ Header Files
Biobase             Biobase: Base functions for Bioconductor
BiocFileCache       Manage Files Across Sessions
BiocGenerics        S4 generic functions used in Bioconductor
BiocIO              Standard Input and Output for Bioconductor Packages
BiocManager         Access the Bioconductor Project Package Repository
BiocParallel        Bioconductor facilities for parallel evaluation
BiocVersion         Set the appropriate version of Bioconductor packages
biomaRt             Interface to BioMart databases (i.e. Ensembl)
biomartr            Genomic Data Retrieval
Biostrings          Efficient manipulation of biological strings
bit                 Classes and Methods for Fast Memory-Efficient Boolean
                    Selections
bit64               A S3 Class for Vectors of 64bit Integers
bitops              Bitwise Operations
blob                A Simple S3 Class for Representing Vectors of Binary

```

Al abrir RStudio, se cargan automáticamente algunos paquetes. Podemos ver los paquetes cargados usando el comando **loadedNamespaces()**

Con los paquetes básicos se pueden hacer análisis estadísticos completos; sin embargo, la comunidad de usuarios de R publica constantemente nuevos paquetes que extienden los métodos y capacidades de R.

```
loadedNamespaces()
```

```
## [1] "compiler"  "fastmap"   "cli"       "graphics"  "tools"
## [6] "htmltools" "rstudioapi" "utils"     "yaml"      "grDevices"
## [11] "stats"     "datasets"  "rmarkdown" "knitr"     "methods"
## [16] "xfun"      "digest"    "rlang"     "base"      "evaluate"
```

Es importante distinguir entre tener un paquete instalado en el computador y tenerlo cargado en memoria:

- Tenerlo instalado significa que en algún momento lo hemos bajado de internet y lo hemos copiado en algún directorio en que R lo puede localizar.
- Tenerlo cargado en memoria significa que durante nuestra sesión de trabajo R ha leído el contenido del paquete y ha incorporado las funciones que contiene a su espacio de trabajo, de tal forma que tales funciones pueden ya ser invocadas y ejecutadas.