

Módulo 1. Conceptos básicos de estadística y RStudio

El rendimiento deportivo se trata de cómo los deportistas se desempeñan, la tarea de un científico del deporte es investigar y monitorear constantemente las condiciones de carga de trabajo para maximizar el rendimiento y reducir el riesgo de lesiones. El término oficial de científico del deporte se define como un rol que no solo recopila, organiza, analiza e interpreta los datos, sino que también integra los datos de diferentes disciplinas en un esfuerzo por generar una perspectiva integral del deportista (Martin, 2019).

Cuantificar medidas de rendimiento, como los indicadores clave de rendimiento descritos en los módulos anteriores, nos permite capturar conocimientos clave sobre cosas minúsculas y casi insignificantes a simple vista.

Eso es lo que buscamos hacer con el análisis de datos, encontrar lo que está mal y que podemos corregir, examinar si los números respaldan la experiencia de los entrenadores y los cazatalentos, o si pueden abrir nuestros ojos a una idea que de otro modo pasaría desapercibida.

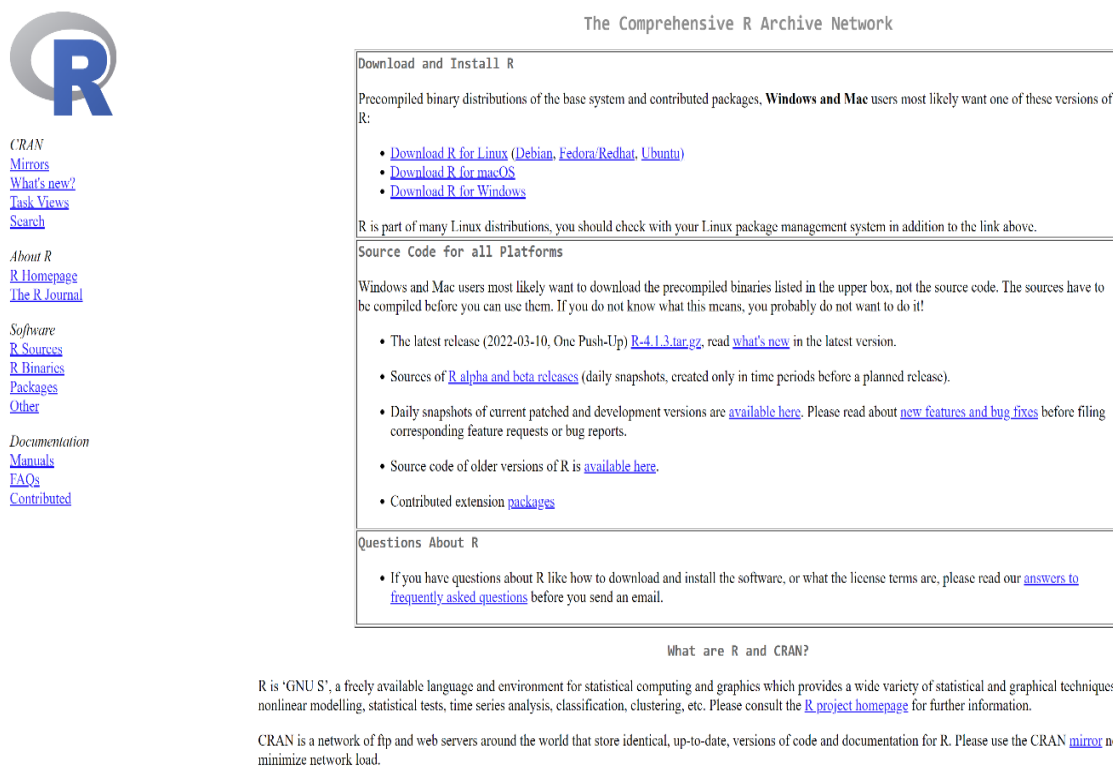
Poder hacer esto es una ciencia, pero también, visualizar los datos correctamente tiene un elemento artístico. Pero primero, debemos aprender a implementar la herramienta para poder importar, analizar e interpretar, y extraer los datos accionables. Para hacer esto, es ideal aprender un lenguaje de programación como R y RStudio, que fueron desarrollados específicamente para análisis estadísticos y visualización de datos. Sin embargo, lo fundamental, es tener al menos una comprensión básica de estadísticas para que puedan aplicarse con precisión a los datos (Basole y Saupe, 2016).

Existen muchas herramientas analíticas que se pueden utilizar para analizar datos; R y RStudio, Excel, Python, STATA, SAS y SPSS. R y Python son las plataformas más utilizadas para el análisis y la visualización de datos en la industria del deporte profesional. La razón para usar R es que se desarrolló solamente para ejecutar análisis estadísticos y, como tal, es una herramienta analítica poderosa. Otra razón para comenzar a aprender R y RStudio es que son de código abierto y gratuitos. Por tal motivo, contribuyentes de todo el mundo que siguen mejorando y evolucionando la plataforma con nuevos paquetes y funciones. Además, RStudio es un entorno de desarrollo integrado para R que consta de cuatro paneles principales, aunque inicialmente se abre con tres paneles (ver figura 3): la consola, un editor con resaltado de sintaxis que puede ejecutar el código directamente; el entorno global que gestiona la administración del espacio de trabajo; y el panel de ayuda y gráficos. En este curso, nos centraremos específicamente en aprender R y RStudio.

Los guiaremos en el proceso de instalación de R y RStudio. R y RStudio deben descargarse en la siguiente secuencia para funcionar correctamente;

- 1) Descargar R del siguiente sitio web: <https://cran.r-project.org/> (ver figura 1 abajo)

Figura 1: Descargar R



The screenshot shows the CRAN website with the following content:

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages. **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux \(Debian, Fedora/Redhat, Ubuntu\)](#)
- [Download R for macOS](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2022-03-10, One Push-Up) [R-4.1.3.tar.gz](#); read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

What are R and CRAN?

R is 'GNU S', a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc. Please consult the [R project homepage](#) for further information.

CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. Please use the CRAN [mirror](#) nearest to you to minimize network load.

Fuente: Captura de pantalla del autor (<https://cran.r-project.org/index.html>)

- 2) Descargar R del siguiente sitio web: <https://www.RStudio.com/products/RStudio/download/#download> (ver figura 2 abajo).

RStudio está disponible en ediciones de código abierto y comerciales, y se ejecuta en el escritorio (Windows, Mac y Linux) o en un navegador conectado (<https://www.RStudio.com/products/RStudio/>).

Figura 2: Descargar RStudio

posit PRODUCTOS SOLUCIONES APRENDA Y APOYE EXPLORA MÁS PRECIOS Q

1: Instalar R

RStudio requiere R 3.3.0+. Elija una versión de R que coincida con el sistema operativo de su computadora.

DESCARGAR E INSTALAR R

2: Instalar RStudio

DESCARGAR RSTUDIO DESKTOP PARA WINDOWS

Tamaño: 212,77 MB | SHA-256: A8325AD5 | Versión: 2023.06.1+524 | Lanzamiento: 2023-07-07

Todos los instaladores y Tarballs

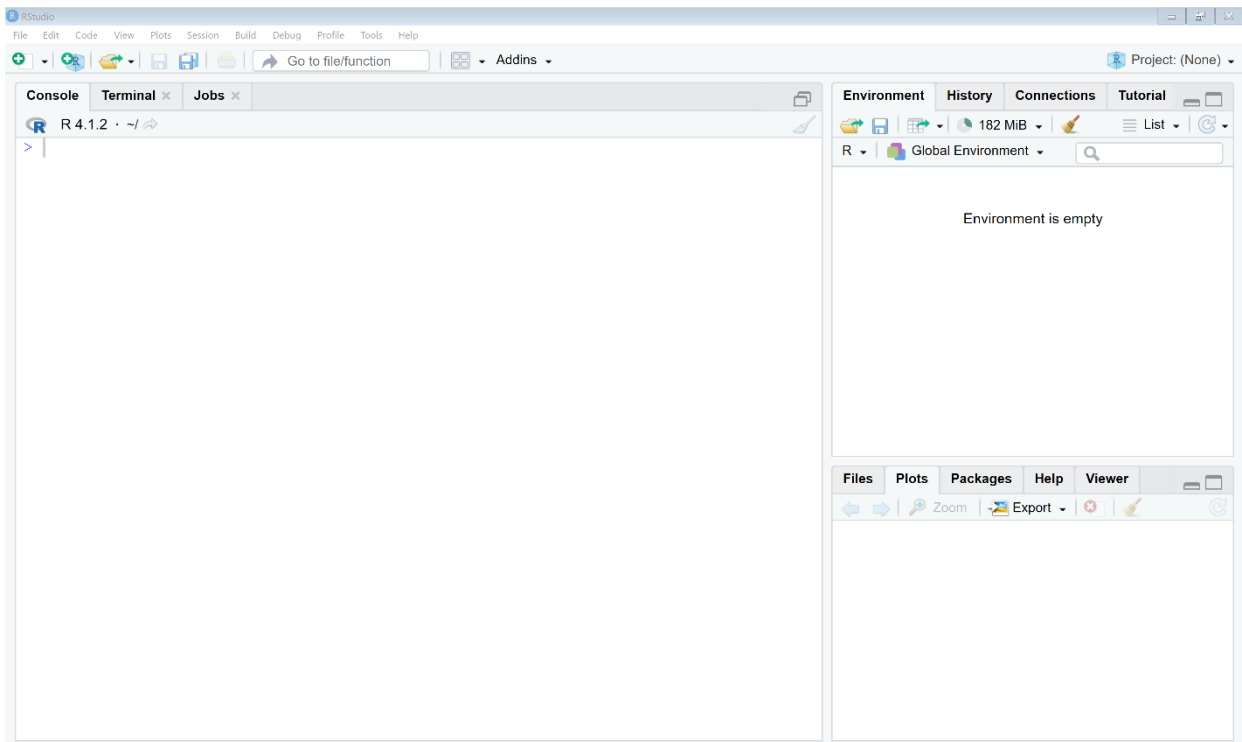
RStudio requiere un sistema operativo de 64 bits. Si está en un sistema de 32 bits, puede usar [una versión anterior de RStudio](#).

Es posible que los usuarios de Linux necesiten importar [la clave de firma de código pública de Posit](#) antes de la instalación, según la política de seguridad del sistema operativo.

TÚ	Descargar	Tamaño	SHA-256
ventanas 10/11	RSTUDIO-2023.06.1-524.EXE ⬇	212,77 MB	A8325AD5
mac OS 11+	RSTUDIO-2023.06.1-524.DMG ⬇	380,82 MB	184804EA
Ubuntu 20/Debian 11	RSTUDIO-2023.06.1-524-AMD64.DEB ⬇	145,85 MB	49E24A69
ubuntu 22	RSTUDIO-2023.06.1-524-AMD64.DEB ⬇	146,82 MB	C030EC83

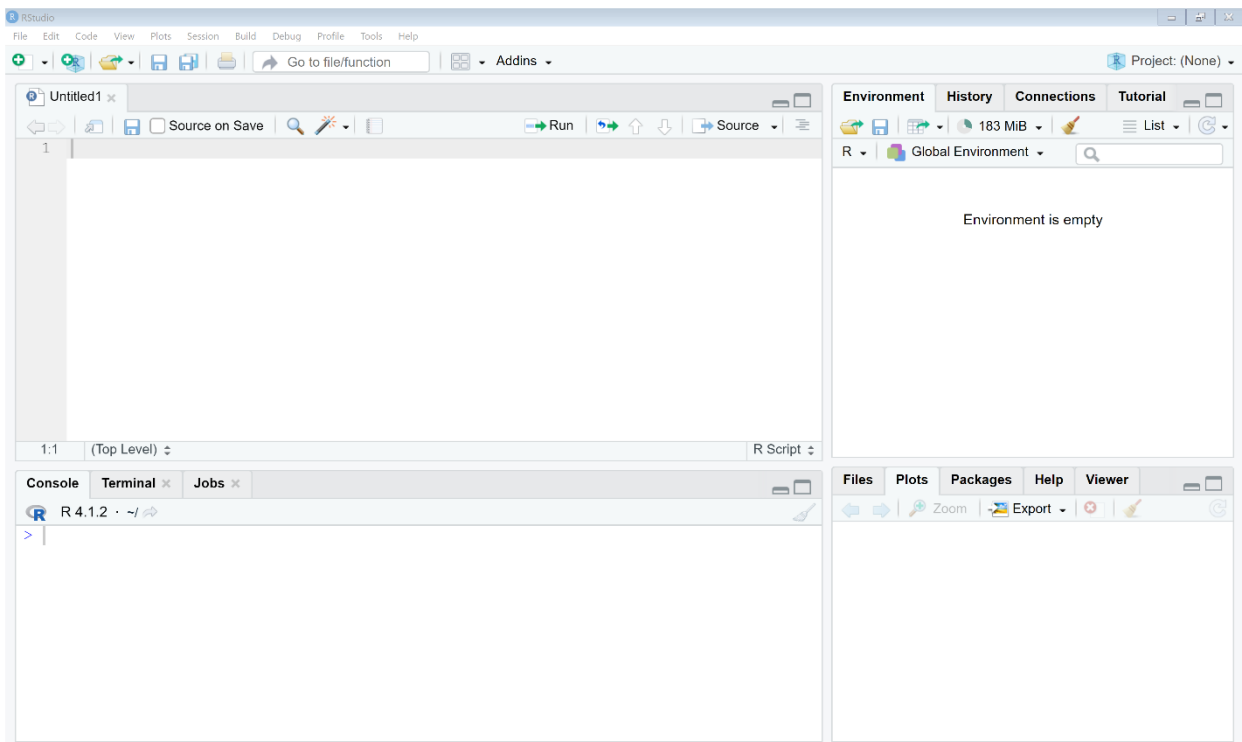
Fuente: Captura de pantalla del autor (<https://www.RStudio.com/products/RStudio/download/#download>)

Figura 3: Un ejemplo de la vista por defecto de los tres paneles



Fuente: Captura de pantalla de RStudio realizada por el autor (RStudio, 2022).

Figura 4: Un ejemplo de los cuatro paneles



Fuente: Captura de pantalla de RStudio realizada por el autor (RStudio, 2022).

Descripción de para qué sirve cada panel de ventana

El panel de ventana superior izquierdo, comúnmente conocido como el panel de scripts, es donde codificas y escribes la sintaxis. La sintaxis es un conjunto de estructuras y comandos legales que utilizas para dar instrucciones para implementar ciertas funciones y ejecutarlas.

El panel de ventana inferior izquierdo se llama console, y es donde se muestra la salida, es decir, los resultados ejecutados a partir del código en el panel de ventana superior izquierdo, en otras palabras, los mensajes impresos del programa que llegan al usuario.

El panel de ventana inferior derecho se llama panel de tareas y consta de varias pestañas en donde se pueden encontrar archivos (files), gráficos (plots), paquetes (packages) y solicitar ayuda (help). La pestaña packages muestra procedimientos de R disponibles, mientras que la pestaña help es para cualquier tema de R y muestra respuestas a preguntas relacionadas con las funciones.

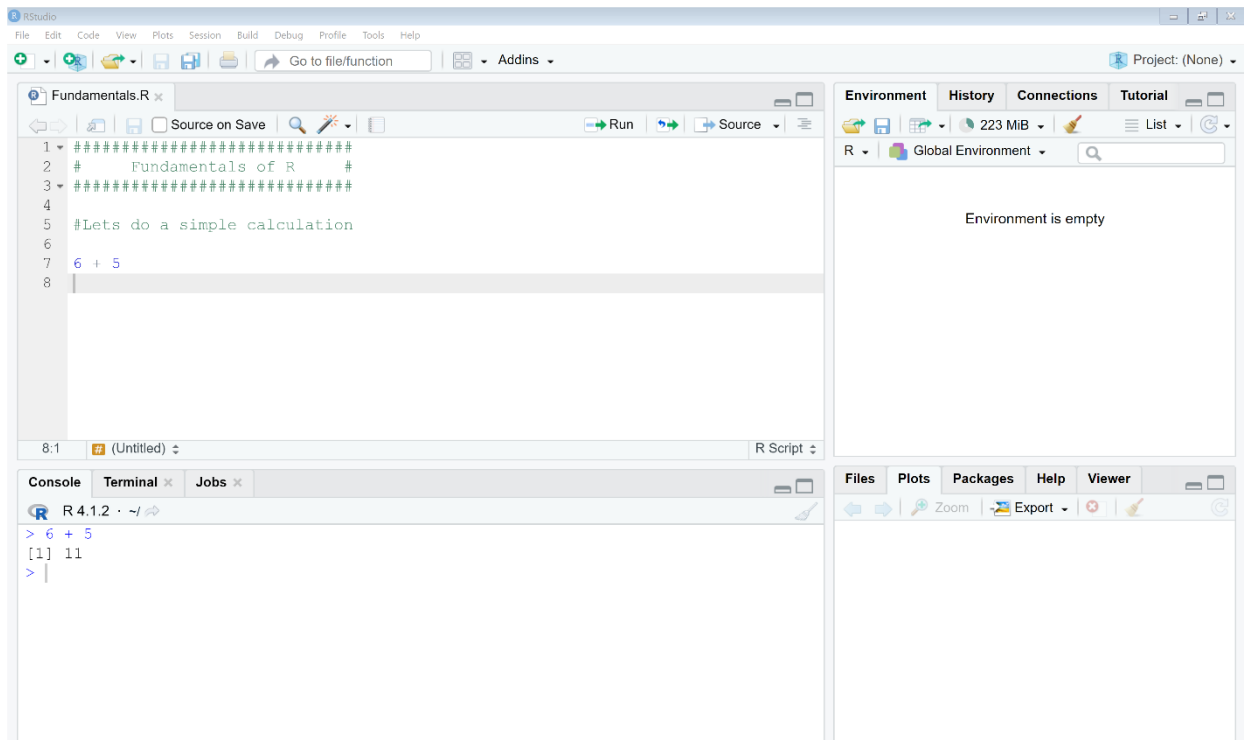
El panel de ventana superior derecho también se denomina tu environment y se refiere a tu entorno de trabajo en R, que incluye cualquier objeto (vectores, matrices, data frames, listas, funciones) que puedas haber creado o importado en R.

Antes de continuar con la codificación, es importante comprender algunos términos del lenguaje R, como el objeto, vectores, funciones, argumentos, paquetes, matrices, data frames y listas (<http://www.sthda.com/english/wiki/easy-r-programming-basics>).

A continuación, se presenta un listado de términos generales y descripciones:

- Un **objeto** es cualquier cosa que se crea en R.
 - Antes de comenzar a crear un objeto, ejecutemos un cálculo simple de la suma de $6 + 5$ en el script de R (panel superior izquierdo).
 - Cuando presionas CTRL + Enter, verás el resultado de la salida generado en la consola (panel inferior izquierdo), como se muestra en la figura a continuación.
 - También debes presionar el botón *File* en la barra superior y elegir *Save as* para nombrar este archivo como «*Fundamentals*». Esto nombrará automáticamente la pestaña como "*Fundamentals.R*", como se muestra en la figura a continuación (Figura 5).

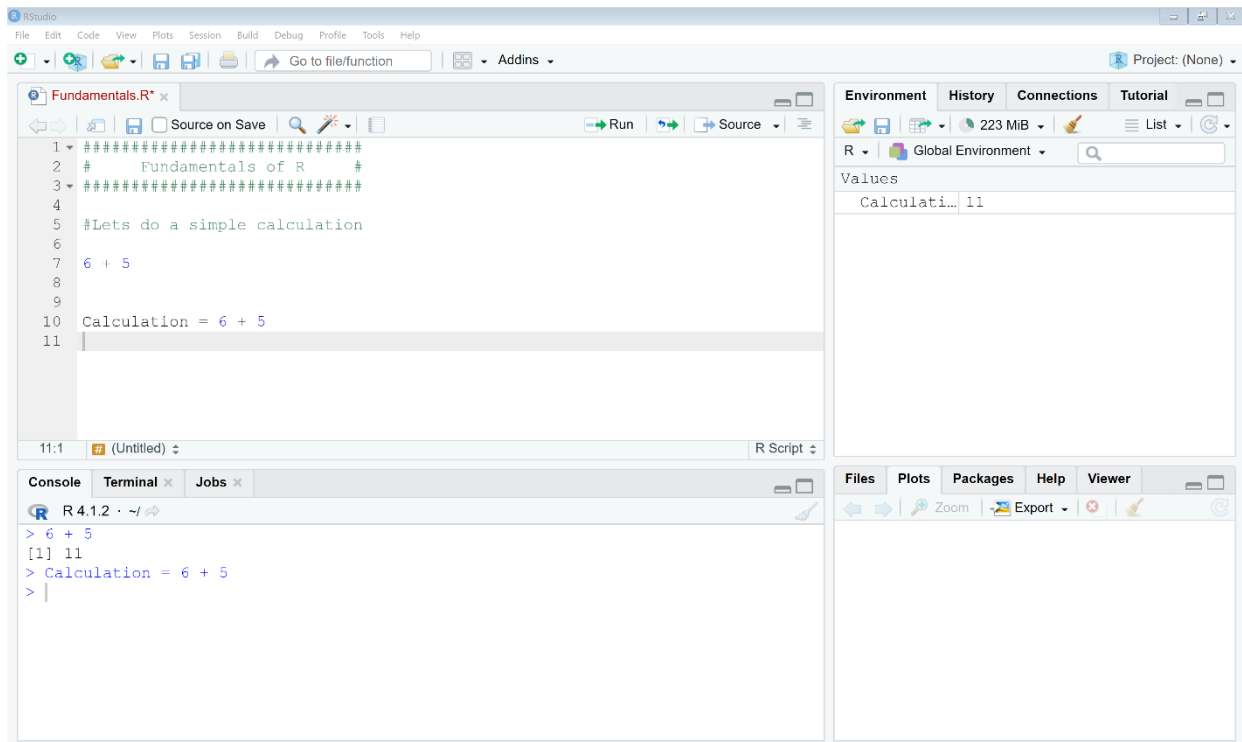
Figura 5: Cálculos en R



Fuente: Captura de pantalla de RStudio realizada por el autor (RStudio, 2022).

- Luego, si deseas llevarlo un paso más allá y crear un objeto, escribimos de la siguiente manera (suponiendo que estás creando un objeto llamado *Calculation*: Ten en cuenta que R distingue entre mayúsculas y minúsculas):
 - *Calculation* = 6 + 5 (Ahora, nota algunas cosas; si observas tu *Global Environment* en el panel de la ventana superior derecha, verás que tu entorno ya no dice *Environment is empty* como lo hacía antes, revisa las capturas de pantalla anteriores). Ahora contiene un objeto llamado *Calculation*, como se muestra a continuación en la Figura 6.

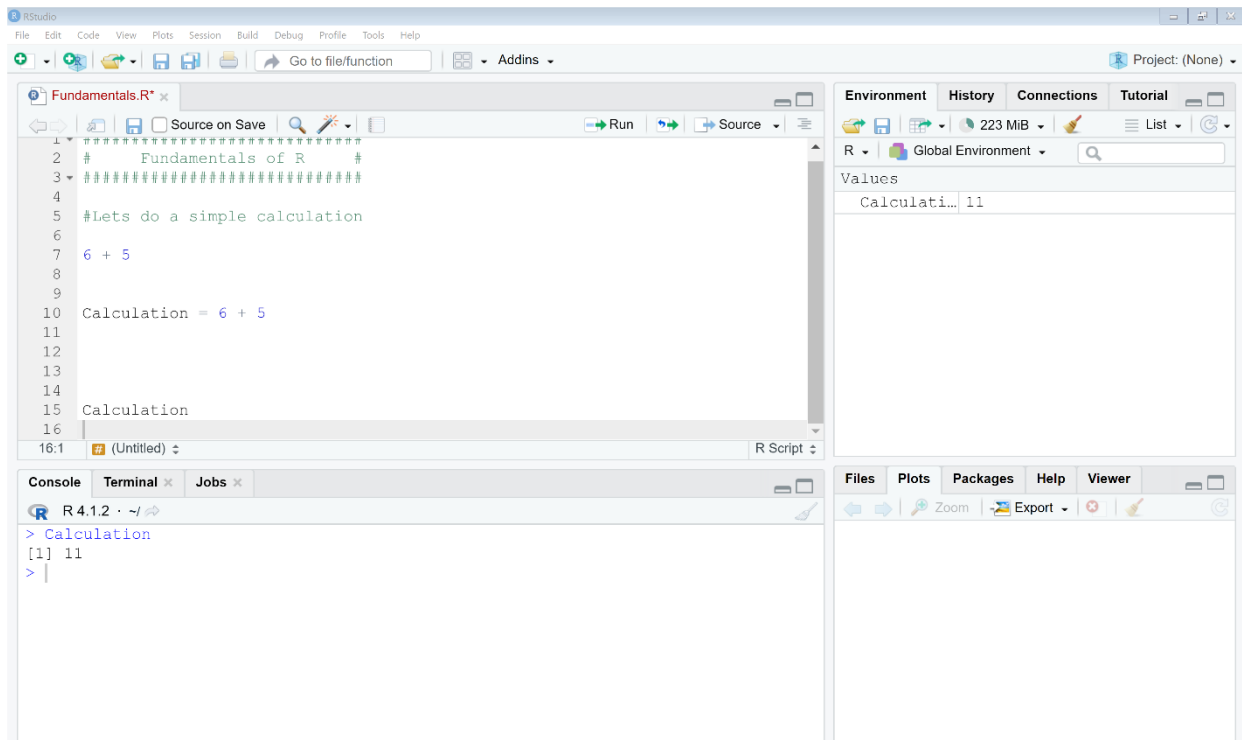
Figura 6: Crear un objeto en R



Fuente: Captura de pantalla de RStudio realizada por el autor (RStudio, 2022).

- Fíjate que cuando ejecutaste el código presionando CTRL + Enter se muestra Calculation, pero no imprimió el resultado del cálculo. Para ver el resultado del objeto Calculation, deberás llamar a Calculation, y mostrará la salida en la consola.
- Lo que se entiende por llamar al objeto es simplemente escribir en la consola de la siguiente manera: Calculation, o escribirlo en el script de R y luego presionar CTRL + Enter, lo que producirá una salida similar a la siguiente figura, figura 7.

Figura 7: Llamar al objeto

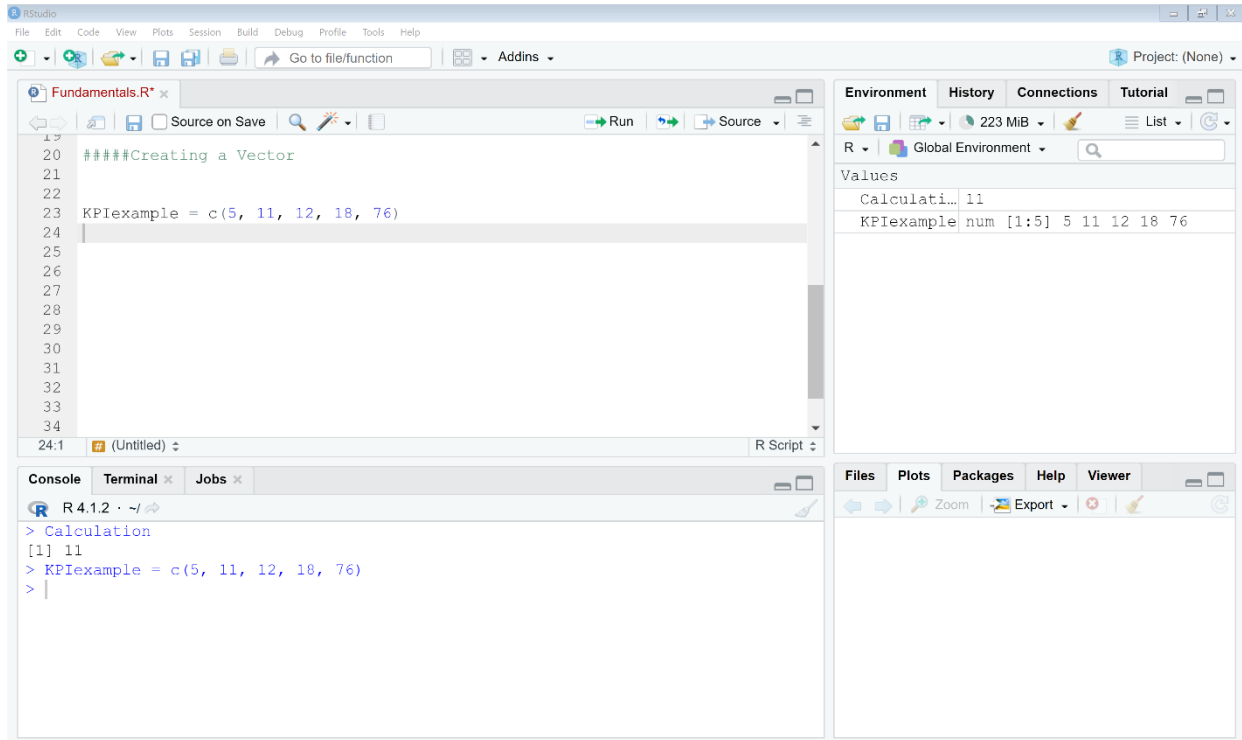


Fuente: Captura de pantalla de RStudio realizada por el autor (RStudio, 2022).

- Un **vector** es una colección de datos del mismo tipo. Es una combinación de múltiples valores (numéricos, de caracteres o lógicos) en el mismo objeto.
 - Es conveniente pensar en los vectores como conjuntos de datos. En términos simples, imagina y visualiza una columna de una tabla de Excel y luego transponla horizontalmente. Por ejemplo, un vector en R podría incluir cinco valores: 5, 11, 12, 18, 76, en un objeto llamado KPIExample, creando un vector de la siguiente manera:
 - $KPIExample = c(5, 11, 12, 18, 76)$
 - Es importante tener en cuenta que, para agrupar varios valores numéricos, estamos utilizando una función. La función que estamos usando es `c()`. Punto clave: las funciones suelen ir seguidas de paréntesis abiertos y cerrados. Esta función en particular significa concatenar, que es análogo a combinar.
 - Técnicamente, estamos reuniendo los valores numéricos 5, 11, 12, 18, 76 en una columna. En R se llama un vector de números.
 - Si observas la figura a continuación, notarás que en el panel de Global Environment tienes un objeto llamado KPIexample, que es de tipo numérico (num) y tiene valores del 1 al 5. Como puedes ver, el Global Environment es el

lugar al que debemos acudir para obtener información sobre tus datos. Ver figura 8.

Figura 8: Crear un vector



Fuente: Captura de pantalla de RStudio realizada por el autor (RStudio, 2022).

- En Excel se hubiese mostrado de la siguiente manera:

Figura 9: Vector en Excel

	A
1	KPIExample
2	5
3	11
4	12
5	18
6	76

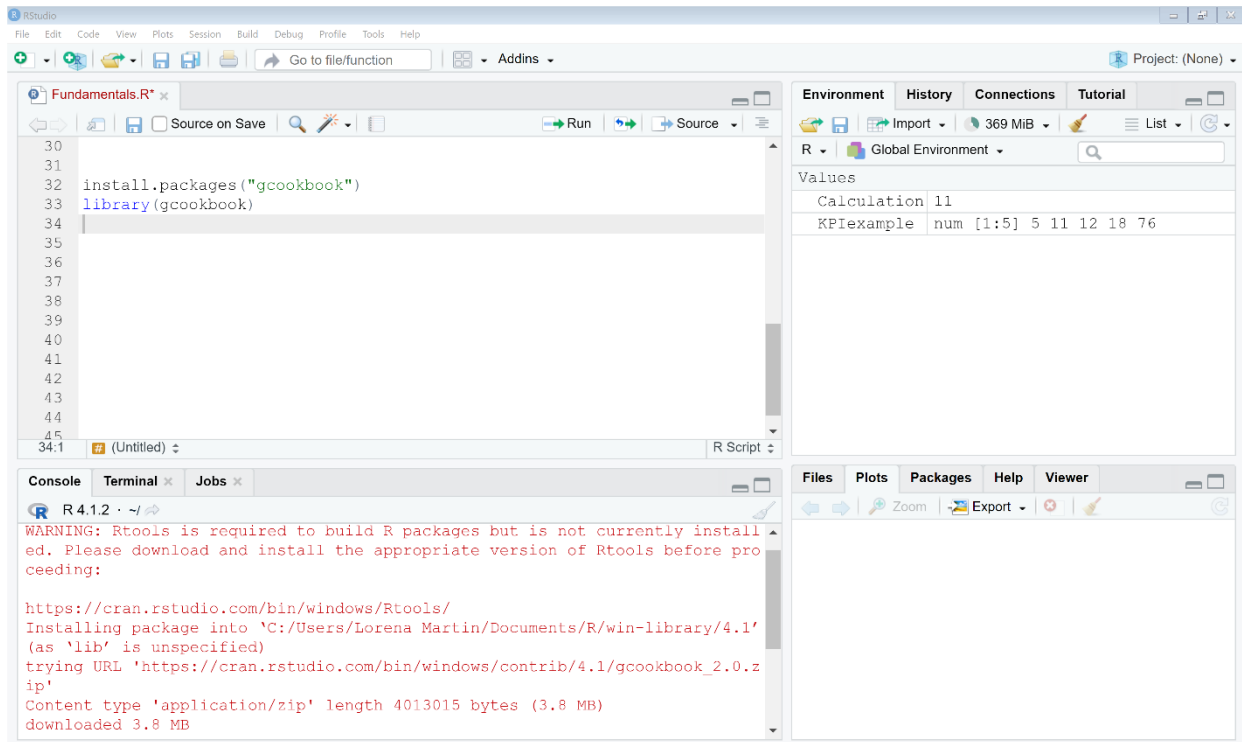
Fuente: Captura de pantalla de Excel realizada por el autor (Microsoft Excel, 2021)

- Una **función** es una herramienta que R utiliza para llevar a cabo algún tipo de operación y se la suele identificar con un patrón que normalmente es una palabra en minúsculas seguida de paréntesis abiertos y cerrados, por ejemplo: `summary()`.

Esta función proporciona al científico del deporte un resumen de seis números de estadísticas descriptivas (más información más adelante). También debes tener en cuenta que, eventualmente, puedes crear tus propias funciones que realicen operaciones complejas.

- Para mostrar cómo usar funciones en R, se recomienda ejecutar funciones en datos para adquirir experiencia.
- RStudio tiene paquetes que, si se instalan, ponen ciertos datos a tu disposición.
- Por favor, instala el paquete **gcookbook**, ya que contiene un conjunto de datos relacionados con el deporte del béisbol llamado **tophitters2001**, para el cual podemos ejecutar varias funciones y ver cómo funcionan en R, como se muestra en la figura 10 a continuación.
- Ten en cuenta que cuando escribas las líneas de código (asegúrate de que estén todas en minúsculas), no debes preocuparte por las advertencias que puedan aparecer en la consola.
- ***install.packages("gcookbook")***
- ***library(gcookbook)***

Figura 10: Funciones en R

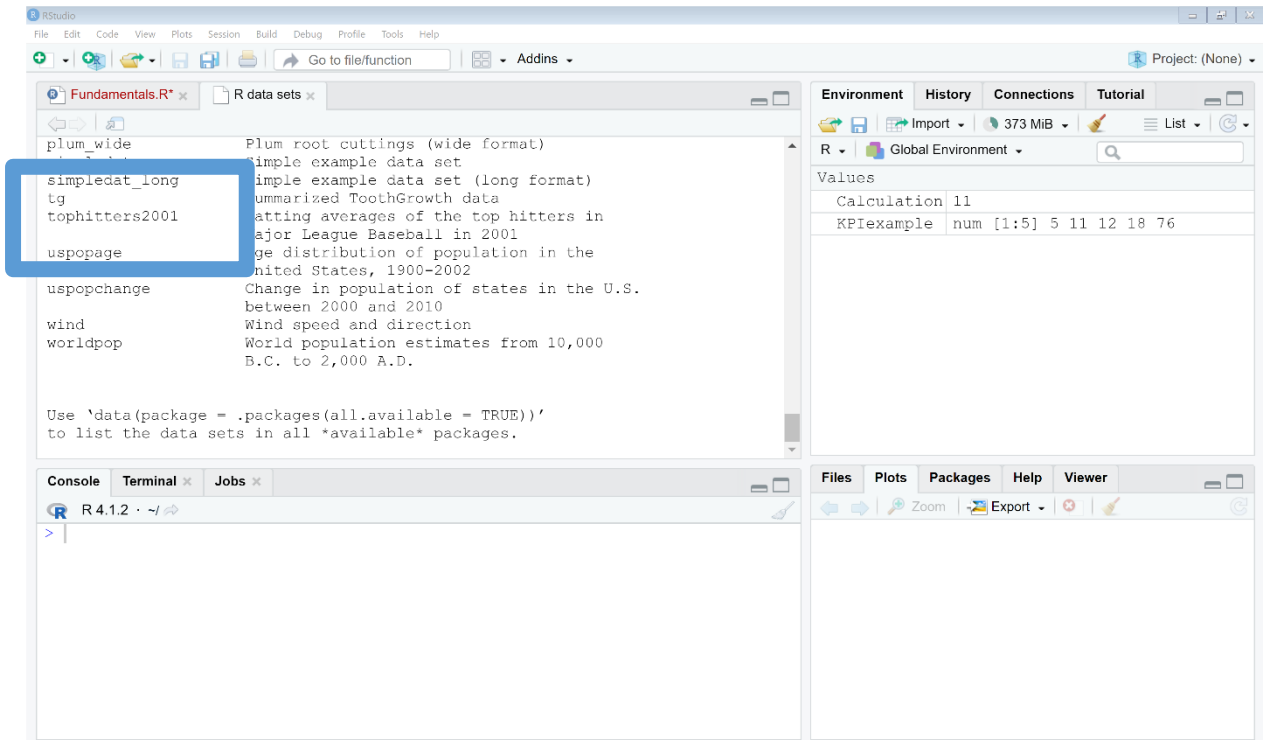


Fuente: Captura de pantalla de RStudio realizada por el autor (RStudio, 2022).

Una vez que el paquete gcookbook haya sido instalado y cargado utilizando la función library, escribe la siguiente línea para mostrar los conjuntos de datos disponibles: data()

Esto mostrará algunos conjuntos de datos que están disponibles de forma gratuita dentro de R y el paquete gcookbook. Dado que este es un curso de análisis de rendimiento deportivo, te recomendamos que sigas esta guía e instales gcookbook, lo llames con library y luego identifiques el conjunto de datos tophitters2001. Ver la figura a continuación (Figura 11).

Figura 11: Conjunto de datos top hitters 2001

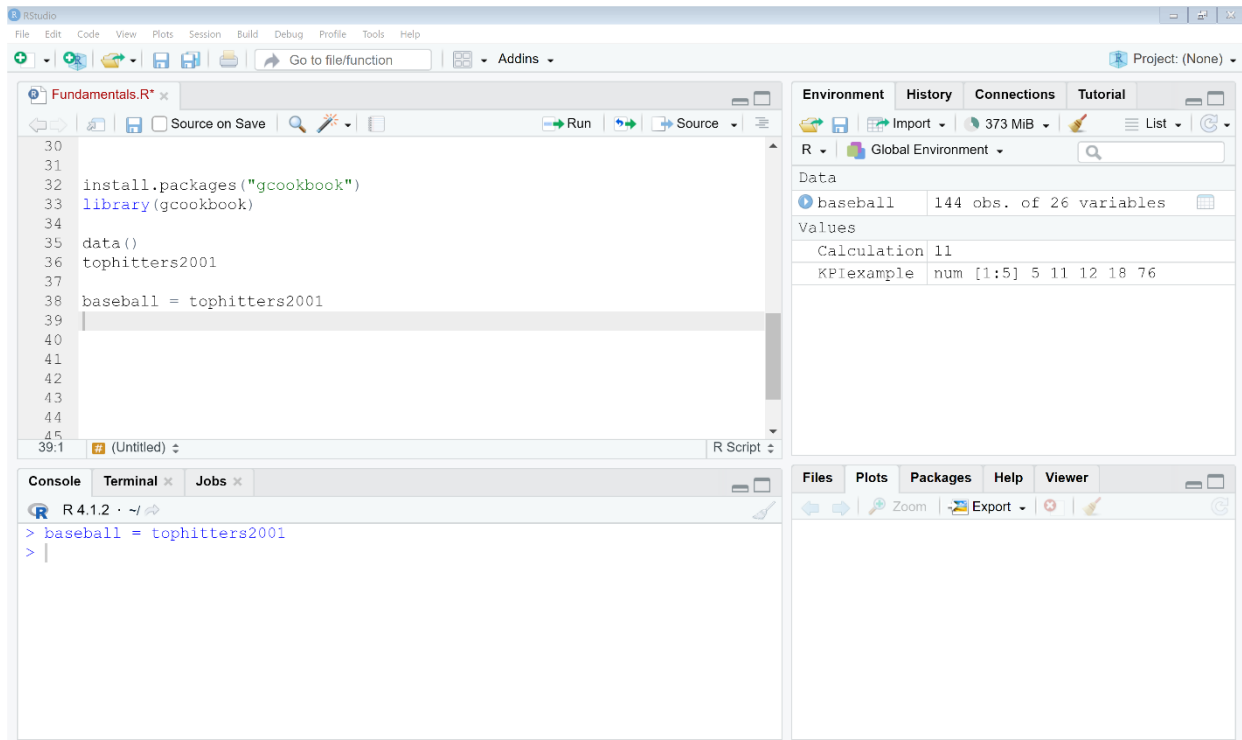


Fuente: Captura de pantalla de RStudio realizada por el autor (RStudio, 2022).

Ahora, crearemos un objeto llamado baseball que contendrá los datos de tophitters2001.

- Observa que en el Global Environment ahora tienes un ícono que aparece como un círculo azul. Esto muestra que hay un objeto que es un data frame (más información sobre esto más adelante. Por ahora solo debes saber que los datos están organizados en filas y columnas).

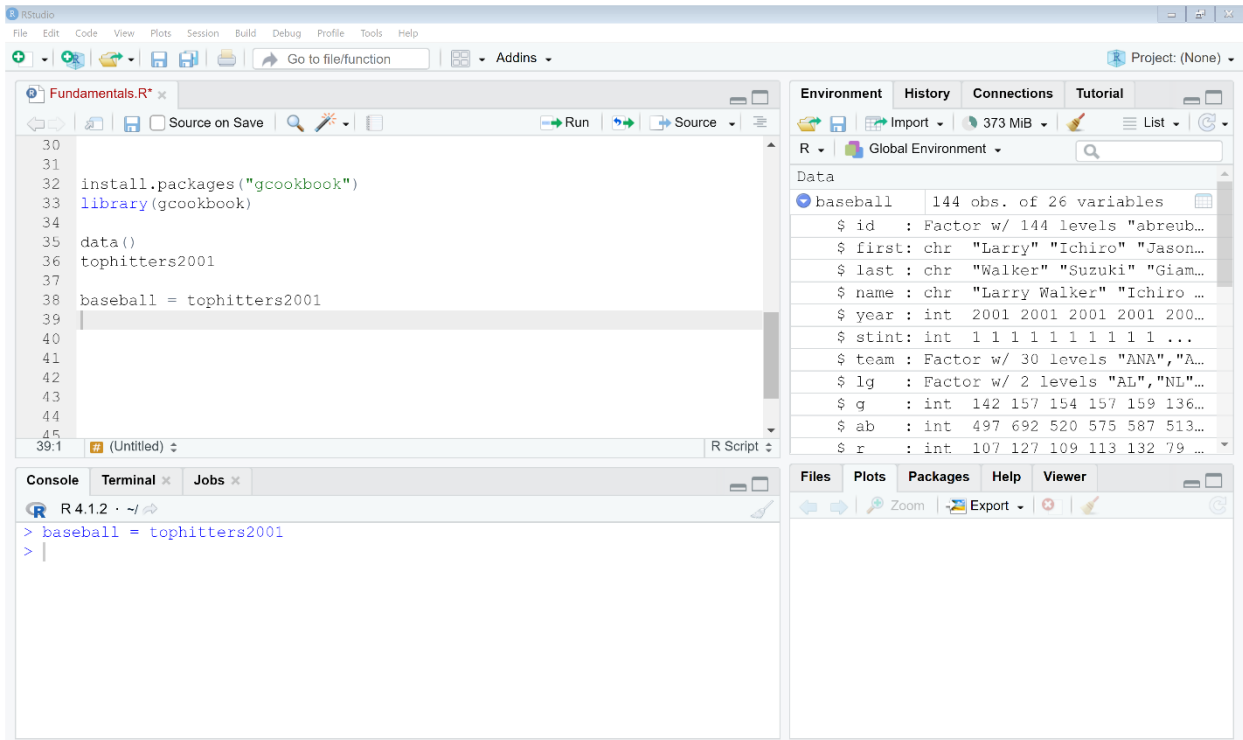
Figura 12: Objeto que contiene datos de tophitters2001



Fuente: Captura de pantalla de RStudio realizada por el autor (RStudio, 2022).

- Si haces clic una vez en el círculo azul, se mostrarán todas las variables e indicadores clave de rendimiento contenidos en él, como se muestra en la figura 13.

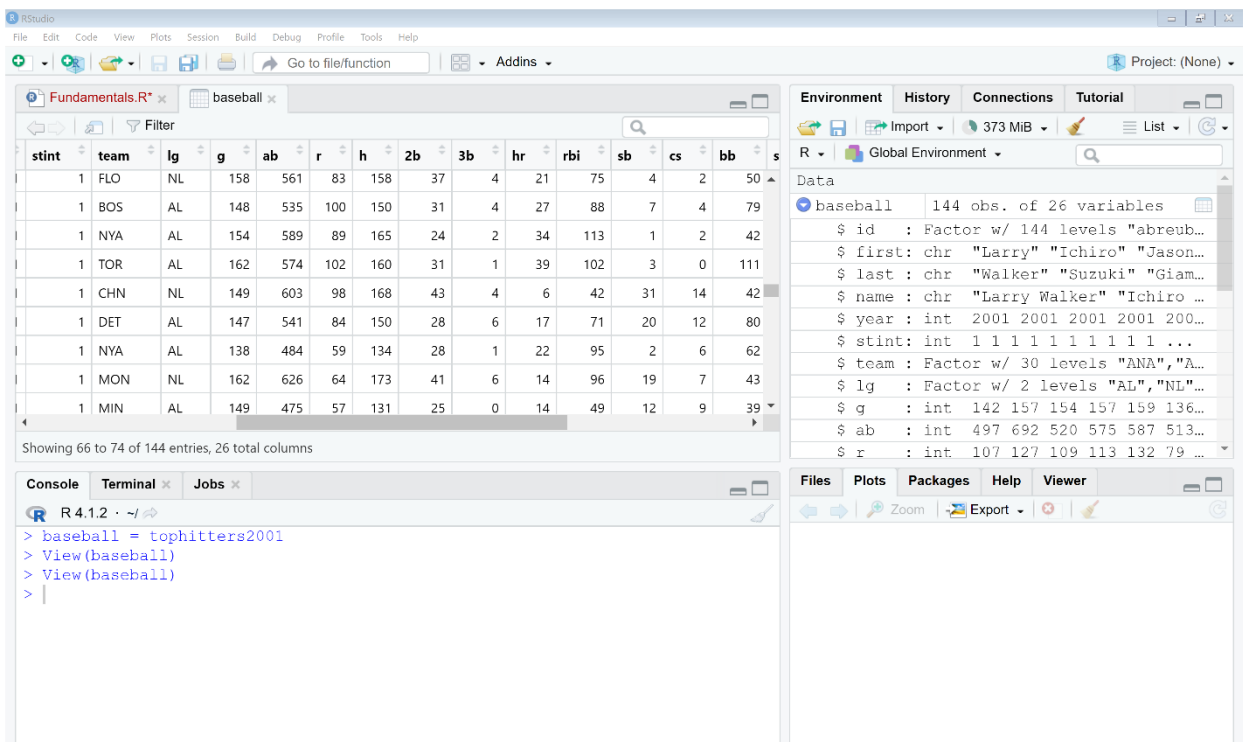
Figura 13: Variables del objeto



Fuente: Captura de pantalla de RStudio realizada por el autor (RStudio, 2022).

- Si haces doble clic en el ícono, se abrirá una pestaña similar a cuando escribiste `data()`, pero esta vez mostrará los datos en una tabla con formato similar a Excel, como se muestra en la figura 14 a continuación.

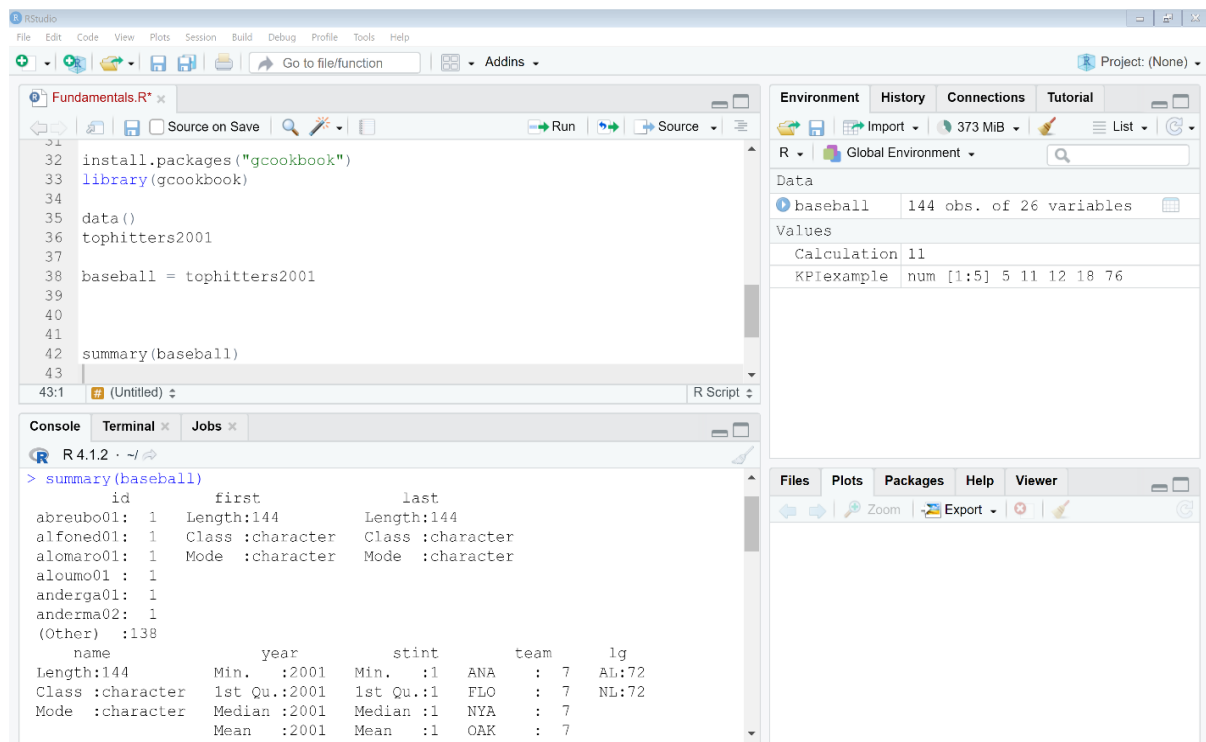
Figura 14: Datos en formato de tabla



Fuente: Captura de pantalla de RStudio realizada por el autor (RStudio, 2022).

Volviendo a las funciones, ejecutemos la función `summary()` en todo el data frame `baseball`, como se muestra en la figura a continuación (Figura 15).

Figura 15: Función `summary()`



Fuente: Captura de pantalla de RStudio realizada por el autor (RStudio, 2022).

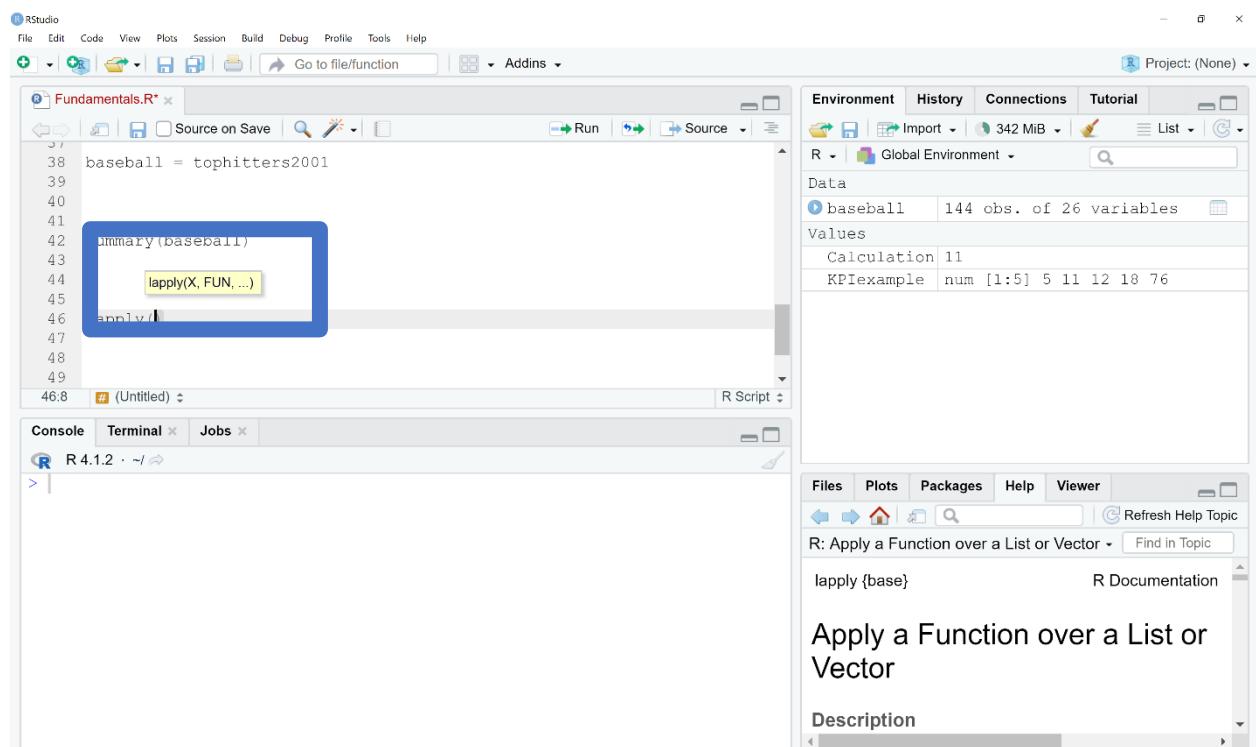
Cuando escribes `summary(baseball)`, obtienes la salida que se muestra en la figura anterior, donde automáticamente se proporciona un resumen de seis números para cada variable numérica en el conjunto de datos. Los tipos de datos que son exclusivamente caracteres muestran solo la siguiente información: longitud (`length`), clase (`class`) y modo

(mode) (más sobre tipos de datos más adelante). Para las variables de tipo numérico, se proporcionan el mínimo, el primer cuartil, la mediana, la media, el tercer cuartil y el máximo.

Esta fue una forma sencilla y práctica de obtener estadísticas descriptivas de todo tu conjunto de datos.

- Un **argumento** es sobre lo que actúa una función. Por ejemplo, en el punto anterior sobre funciones, utilizamos el ejemplo de la función `summary()`. Ahora, si implementáramos una nueva función llamada `lapply`, que se utiliza para aplicar una función sobre una lista o un vector, y comenzaras a escribir `lapply()` cuando abres los paréntesis, verás que aparece un indicador como se muestra en la figura a continuación. Este indica los argumentos que se recomienda completar para que el código se ejecute correctamente. En el caso que se muestra en la figura a continuación (Figura 16), se ve que muestra una "X", seguida de "FUN".
- La "X" se utiliza para denotar tu conjunto de datos u objeto.
- "FUN" representa la función que deseas aplicar a tu vector, por ejemplo, `mean()`. Nota al margen, sí, es una función cuasi-nidificada.

Figura 16: Función `lapply()`

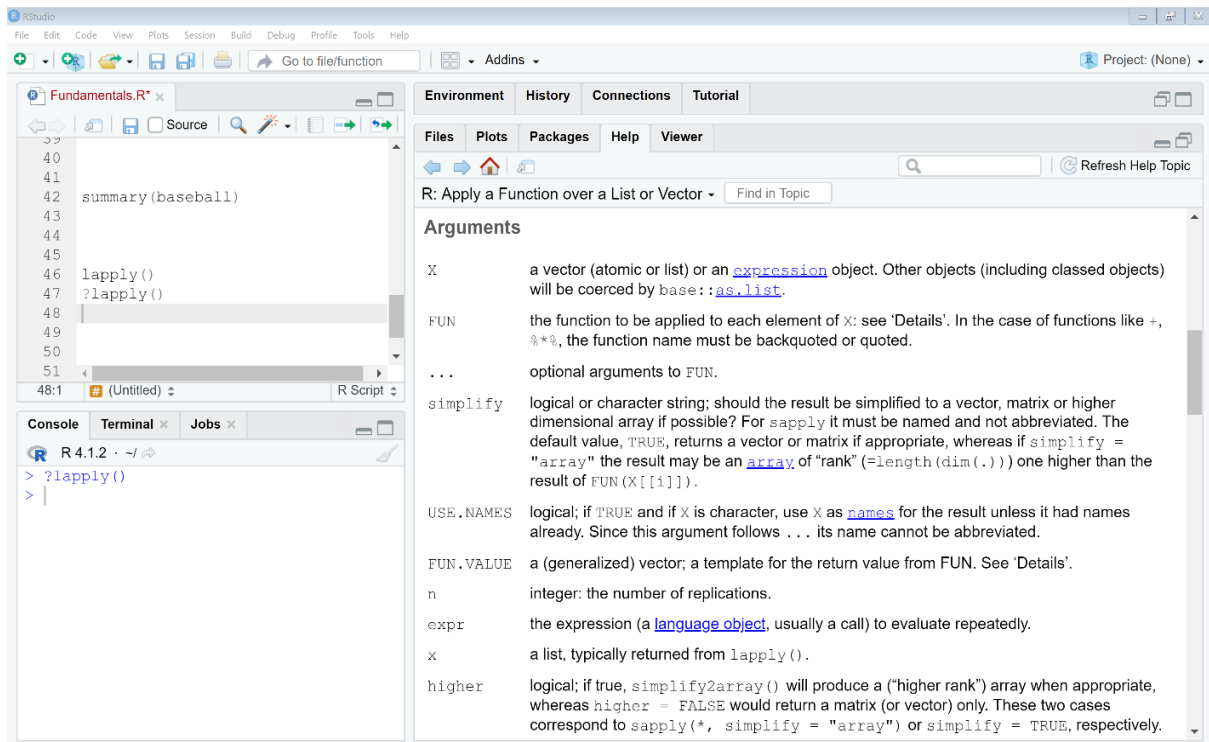


Fuente: Captura de pantalla de RStudio realizada por el autor (RStudio, 2022).

Supongamos que quieres utilizar una función, pero no estás seguro de cuáles son los argumentos que se requieren, puedes pedir ayuda a R escribiendo un signo de interrogación delante de la función y luego presionando CTRL + Enter.

Tan pronto como ejecutes la línea de código para obtener ayuda, el panel de ventana inferior derecho se dirigirá a la pestaña de ayuda y abrirá información sobre la función, así como los argumentos necesarios dentro de esa función, como se muestra en la figura 17 a continuación.

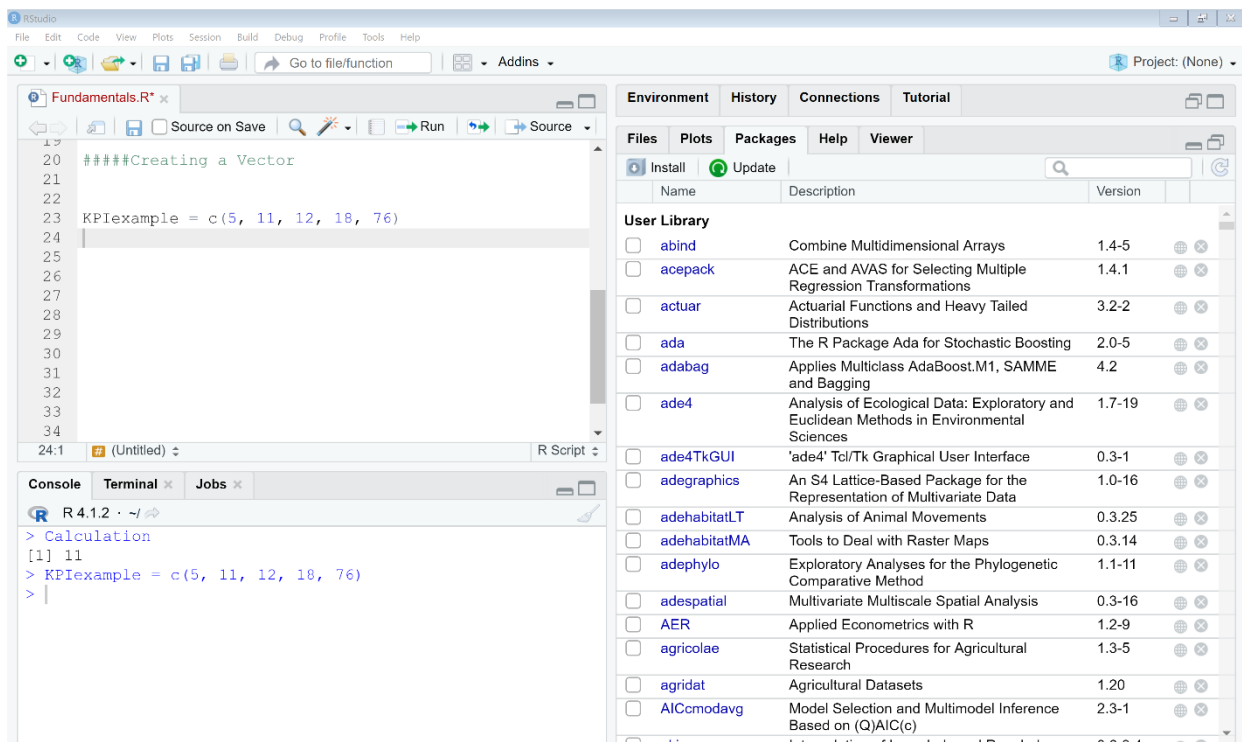
Figura 17: Pestaña ayuda



Fuente: Captura de pantalla de RStudio realizada por el autor (RStudio, 2022).

- Un **paquete** es una colección de funciones. En R, existen multitud de paquetes que constan de numerosas funciones; de hecho, actualmente existen más de 18.375 paquetes que pueden ayudarte a realizar las acciones que deseas.
- Si maximizas el panel de la ventana inferior derecha (paneles Files/Plots/Packages/Help/Viewer), y haces clic en la pestaña de Packages, podrás ver una lista de paquetes disponibles en RStudio que puedes instalar simplemente marcando la casilla a la izquierda del nombre del paquete, como se muestra en la figura 18 a continuación. De lo contrario, puedes utilizar las líneas de código habituales para instalar los paquetes de tu elección utilizando las siguientes:
 - ***install.packages("package")***
 - ***library(package)***

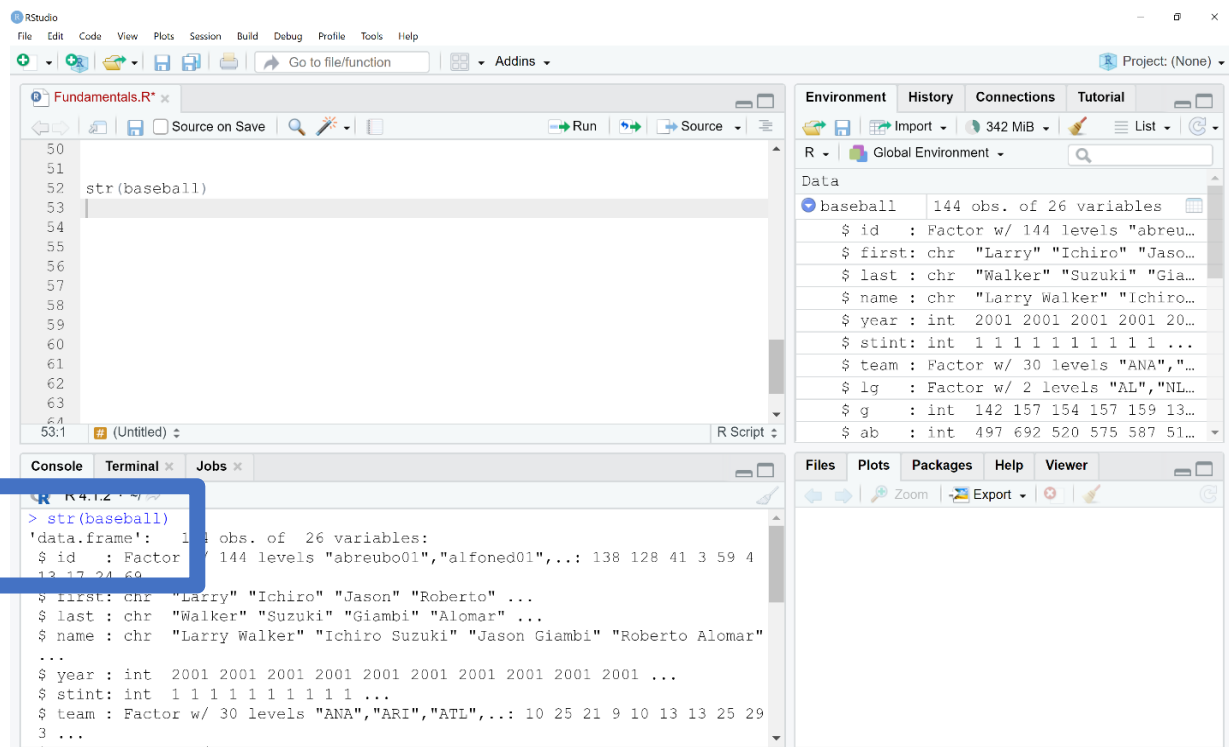
Figura 18: Paquetes en R



Fuente: Captura de pantalla de RStudio realizada por el autor (RStudio, 2022).

- Los **data frames** son tablas, similares a las matrices. La principal diferencia es que las columnas pueden tener diferentes tipos de datos (numéricos, de caracteres, lógicos). También es importante tener en cuenta que se construyen a partir de filas y columnas. En R, las filas se denominan observaciones y las columnas se conocen como variables.
- Entonces, si examinas el Global Environment y ves un ícono con un círculo azul, eso indica que tienes un objeto en R en forma de data frame. Sin embargo, en esta ubicación no se indicará explícitamente que es un data frame.
- Para confirmar la estructura de datos, se recomienda que ejecutes la función `str()` con el nombre del objeto dentro de ella, como se muestra en la figura 19 a continuación. La primera línea en la consola, en el panel inferior izquierdo de la ventana, mostrará la estructura de datos junto con el tipo de datos de cada una de las variables.

Figura 19: Estructura de datos de data frames en R



Fuente: Captura de pantalla de RStudio del autor (RStudio, 2022).

- Las **Matrices** consisten en la combinación de vectores con el mismo tipo de elemento, que puede ser numérico, de caracteres o lógico.
- Las **listas** son una colección de objetos, y su característica distintiva es que pueden contener todo tipo de objetos de R. Por lo general, cuando extraemos datos de la web, los importamos a R como una lista, ya que los sitios web suelen contener números, caracteres, imágenes y otros elementos no estructurados.

Para este curso, nos concentraremos en vectores y data frames, ya que son las estructuras de datos comunes con las que más trabajarás como científico del deporte.

Importar datos en forma de archivos de Excel y archivos CSV

Por lo general, un conjunto de datos regular que recibimos una vez que hemos recopilado datos se caracteriza por filas y columnas, como es habitual en los archivos de Excel y los archivos CSV. En R, cuando importamos un archivo CSV o de Excel y lo guardamos como un objeto, queremos asegurarnos de que asuma la estructura de un data frame, un formato de tabla comparable a una tabla de Excel con filas y columnas. Para hacer esto, importamos el conjunto de datos utilizando el siguiente comando:

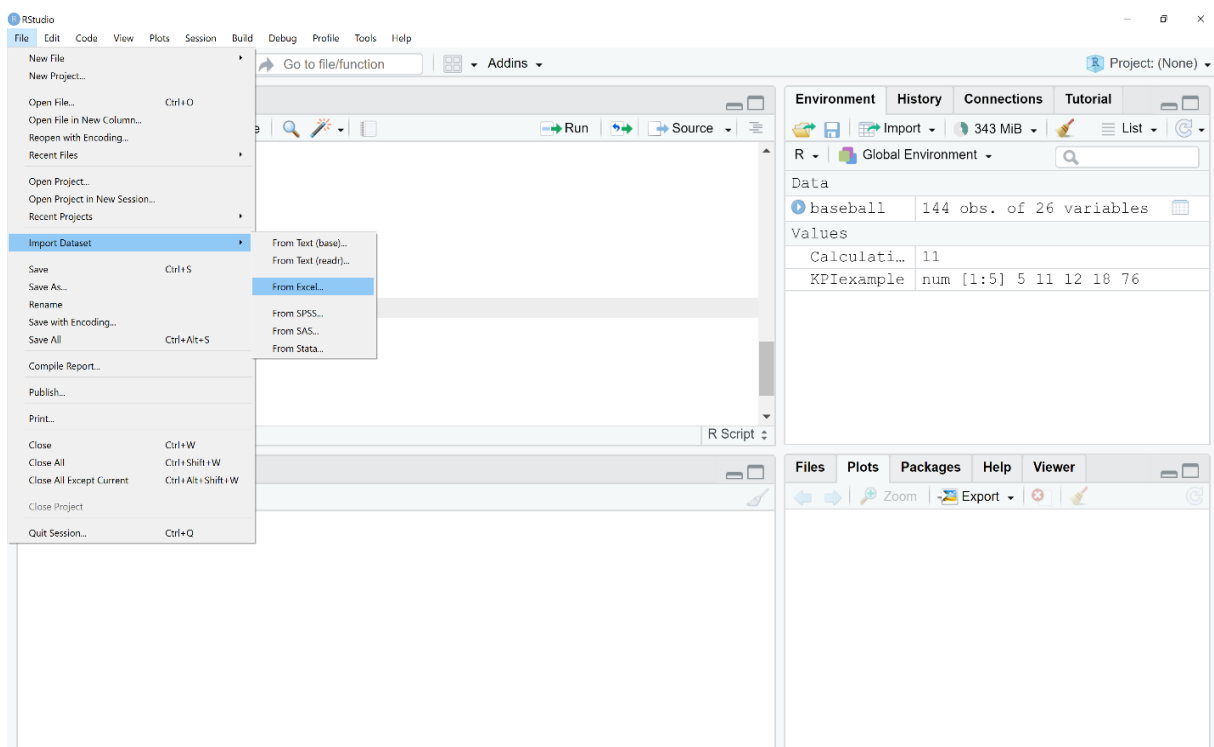
➤ `read.csv(file.choose())`

- Estas dos funciones, `read.csv` y `file.choose`, que está anidada dentro de la función `read.csv`, abrirán una ventana de diálogo para que elijas el archivo CSV adecuado.
- Una vez importado, lo asignamos a un objeto (llamémoslo `sportsinfo`) utilizando los operadores `=` o `<-`.
- Por ejemplo, si importas un conjunto de datos que deseas almacenar y guardar como un objeto llamado `sportsinfo`, el código sería el siguiente:
 - `sportsinfo = read.csv(file.choose())`
- Luego, puedes verificar la estructura del archivo importado para ver si se importó como una tabla o no, utilizando la función `str()`.
- Si no se importó como una tabla o data frame en RStudio, puedes transformarlo utilizando la función `data.frame(sportsinfo)`. Esto es importante porque muchos de los modelos estadísticos y visualizaciones de datos realizados con el paquete `ggplot2` requieren que los datos tengan una estructura de data frame.
- La función `data.frame(sportsinfo)` convertirá el objeto `sportsinfo` en un data frame que luego se puede utilizar para realizar análisis estadísticos y crear visualizaciones de datos utilizando `ggplot2`, otro paquete en R diseñado específicamente para generar visualizaciones de datos efectivas.

También hay una forma alternativa de importar archivos en R mediante clics y puntos:

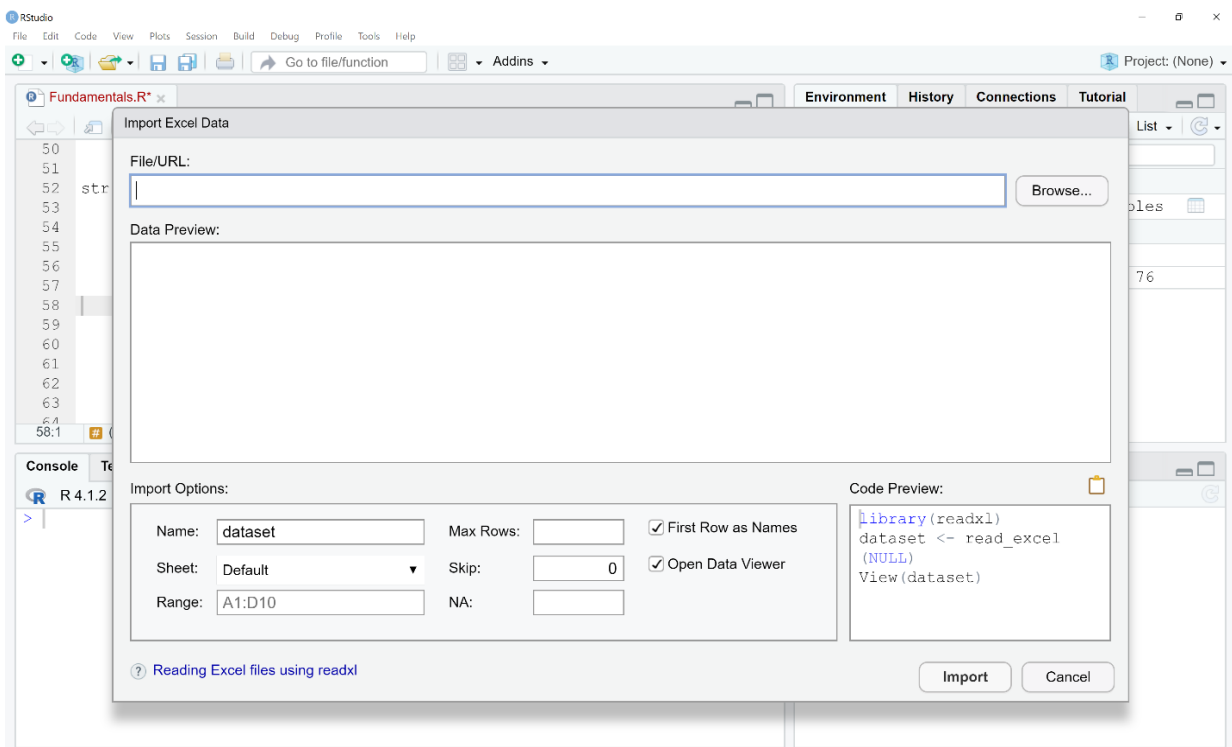
- Ve a `File > Import Dataset > From Excel` en la esquina superior izquierda del panel de scripts.

Figura 20: Importar un archivo en R mediante clics y puntos



Fuente: Captura de pantalla de RStudio realizada por el autor (RStudio, 2022).

Figura 21: Importar una hoja específica dentro de un libro de Excel en R



Fuente: Captura de pantalla de RStudio realizada por el autor (RStudio, 2022).

Hay una tercera forma de importar datos utilizando el directorio al que R está apuntando. Para averiguar dónde está apuntando RStudio en tu sistema, puedes utilizar el siguiente código:

- `getwd()`
 - Esta función representa obtener tu directorio de trabajo y te mostrará la ubicación a la que R está apuntando directamente en tu sistema de computadora.
 - Una vez que identifiques dónde está apuntando, si efectivamente está apuntando al archivo CSV que deseas importar, entonces no hay más pasos que seguir, simplemente lee el archivo usando `read.csv` y guárdalo como un objeto:
 - `sportsinfo <- read.csv("from the file directory")`
- Si tu directorio de trabajo no está apuntando al archivo CSV que deseas importar, o si deseas cambiar hacia dónde apunta R, entonces necesitarás usar el siguiente código para redirigir el directorio de trabajo y establecerlo en una ubicación diferente:
- `setwd()`

- Esta función representa establecer tu directorio de trabajo en una nueva ubicación.
- Si deseas cambiar tu directorio de trabajo a una ubicación diferente, debes identificar la nueva ubicación.
- Una forma de hacerlo
 - es haciendo clic con el botón derecho en el archivo que deseas importar en R, luego ir a properties y, en la pestaña general, encontrarás Location. Copia esa ubicación y escríbela dentro de la función `setwd("Ubicación del archivo que deseas importar")` entre comillas dobles.

Estos son algunos conceptos básicos de R y RStudio. Además, en R, si deseas llevar un registro de información y tomar notas en el camino, por ejemplo, si creaste una nueva variable. Para hacerlo, puedes incluir fácilmente comentarios en el script de R usando el símbolo de numeral (#). Los comentarios pueden colocarse antes o después del código real (Wickham and Grolemund, 2020).

Sugerencia: Si deseas usar el símbolo numeral para comentar y desactivar varias líneas de código o cientos de líneas de código, utiliza el siguiente atajo de teclado:

CTRL + SHIFT + HASHTAG

Esto concluye la introducción al proceso de instalación de R y RStudio, así como los conceptos básicos. En el próximo módulo, aprenderemos sobre los fundamentos de la metodología estadística y comenzaremos a implementar análisis descriptivos con RStudio.

Referencias

Allaire, J. J. (2022). R 4.2.1 [Computer Software]. RStudio, Inc. <https://cran.r-project.org/index.html>

Basole, R. C. y Saupe, D. (2016). Sport Data Visualization [Guest editors' introduction]. IEEE Computer Graphics and Applications, 36, 24-26. <https://doi.org/10.1109/MCG.2016.85>

Martin, L. (2019). Sports science data protocol. *Sports and Exercise Medicine Open Journal*, 5(2), 36-41. <http://dx.doi.org/10.17140/SEM0J-5-174>

Microsoft Corporation. (2018). *Microsoft Excel*. Recuperado de <https://office.microsoft.com/excel>

Wickham, H. y Grolemund, G. (2020). R for Data Science. O'Reilly.