

# Módulo 1. Introducción al software y funcionalidades



☰ Introducción al software y funcionalidades

☰ Referencias

# Introducción al software y funcionalidades

---

## **¿Por qué es necesario el uso de un *software* avanzado de análisis de datos para un *sport scientist*?**

Como profesionales del rendimiento físico, buscamos dar respuesta a múltiples preguntas que se nos presentan en el día a día, así como también desarrollar proyectos a largo plazo, relacionados con la investigación de las cualidades físicas, su desarrollo y su impacto en el rendimiento en el campo.

En los últimos años, el volumen de información del que disponemos ha crecido exponencialmente (Almulla *et al.*, 2020), recogida por fuentes y tecnología diversas de formatos variados. La necesidad de usar esta información para conseguir conocimiento y conseguir una ventaja competitiva respecto a otras organizaciones ha desarrollado el crecimiento del ámbito del análisis de datos en el sector del rendimiento deportivo.

Vamos a poner el ejemplo de lo que podría ser la información que tenemos durante un entrenamiento de fútbol, únicamente aquella información que tiene que ver con el componente físico, el cual es solamente una de las piezas del puzzle para conseguir acercarnos al objetivo principal: rendir y ganar. Este ejemplo pretende reflejar la diversidad y volumen de datos de los que recogemos en nuestro contexto y el producto en bruto del que disponemos para realizar análisis que tengan impacto en la toma de decisiones.

**Fuente de datos 1:** el jugador dispone de una pulsera de monitorización de movimiento y frecuencia cardíaca que usa durante la noche para el cálculo de variables derivadas que están relacionadas con el sueño, el descanso y la recuperación (Miller *et al.*, 2022).

Usando como ejemplo el uso de una de las marcas comerciales más usadas y con validez, por la mañana, obtenemos la información del resumen de la noche, mediante archivos .csv que presentan la siguiente información:

- **Tres archivos .csv**
  - El primero con ciclos fisiológicos.
  - El segundo con el registro del sueño.

- El tercero con información de la actividad física.
- Cada archivo contiene más de diez columnas o variables.
- El formato de las columnas es variable.
  - Fecha.
  - Horas.
  - Texto.
  - Numérico.

**Fuente de datos 2:** después de una pequeña activación en el gimnasio, el jugador hace un protocolo de saltos estandarizados en una plataforma de fuerza con registro a 1000 Hz (Bromley *et al.*, 2021).

La mayoría de las compañías tecnológicas que disponen de plataformas de fuerza disponen de un *software* específico para la visualización y descarga o exportación de resultados. La información que podemos obtener es la siguiente:

- Un archivo Excel.

- Primeras ocho filas con información identificativa del jugador.
- Una fila con el promedio de todos los resultados. También en distintos formatos y con tantas variables como hayamos querido seleccionar (1 a >50).

**Tabla 1: Archivo Excel**

	A	B	C	D	E	F	G	H	I	J	K
1	Device										
2	Weight										
3	Frequency										
4	Recording Date										
5	Recording Info										
6	Data Source										
7	Analysis Date										
8	AthleteId										
9											
10	Name	ExternalId	Test Type	Date	Time	BW [KG]	Reps	Tags	Additional	Concentri	Concentri
11											

Fuente: Elaboración propia.

**Fuente de datos 3:** durante la sesión de gimnasio, el departamento de rendimiento registra, mediante el uso de un *encoder* lineal, información de las características del movimiento que se está haciendo (Hernández-Belmonte *et al.*, 2023). La información que podemos obtener es la siguiente:

- Archivo, pero en formato .csv.

- La primera fila contiene información relevante dentro de cada celda (ejemplo en la imagen).
- Las columnas no son resumen en este caso, sino que contienen la información de cada repetición.

**Tabla 2: Ejemplo**

	A	B	C	D	E	F
1	first name: Miguel	last name: Vazquez	klipfolioID:	date: time: 12:0	exercise: l	
2						
3	Row Type	Rep Number	Conc Mean Ve	Conc Peak Velocity (m/s)		
4	Rep	1				
5	Rep	2				
6	Rep	3				
7	Rep	4				
8	Rep	5				
9	Rep	6				
10	Average	6				
11	Maximum	6				
12	Std.Dev.	6				
13	Total	6	-	-		

Fuente: Elaboración propia.

**Fuente de datos 4:** el jugador dispone de un dispositivo LPS que permite el control de la actividad locomotora y mecánica durante el entrenamiento (Caro *et al.*, 2022). Después del entrenamiento, se descarga un archivo con características similares a los dos anteriores:

- Archivo Excel o csv, dependiendo de la marca comercial.
- Identificadores en las primeras filas (fecha/número de jugadores/equipo).
- Múltiples columnas con las variables a analizar que seleccionemos.
  - En este caso, también son variables resumen.
  - La particularidad es que el resumen se hace por cada uno de los periodos que seleccionemos, por ejemplo:
    - Calentamiento.
    - Tarea 1.
    - Tarea 2.
    - Sesión completa.

**Fuente de datos 5:** al finalizar el entrenamiento, un miembro del *staff* de rendimiento le pregunta al jugador sobre su percepción del esfuerzo en la sesión (Kuhlman et al., 2023). Hay múltiples opciones

para dicha recogida de datos, mediante un Excel a medida, usando *apps* disponibles en el mercado.

De nuevo, obtenemos otro documento o archivo con respuestas numéricas y más columnas que reflejan las variables que queremos analizar.

La cantidad de información del ejemplo anterior varía según el deporte, el nivel y el contexto (recursos humanos y materiales, influencia del *staff*, aceptación por parte de los jugadores para el registro, etcétera), pero las posibilidades, aunque el volumen de datos sea menor en esos otros contextos, son ilimitadas. Como hemos mencionado anteriormente, hay casos en los que estas fuentes de datos son reducidas, pero encontramos contextos en los que esa cantidad de información puede ser mayor (información de composición corporal, nutrición o ingesta calórica, cuestionarios *wellness*, etcétera).

Las posibilidades de aumentar la extensión o el alcance del impacto que tienen los datos físicos son ilimitadas. Imaginemos otro tipo de fuentes de datos, como cámaras de video con las que crucemos acciones del partido de las que queramos conocer la carga a la que corresponden. ¿En qué momentos está esprintando el jugador? ¿En acciones defensivas u ofensivas? Necesitamos cruzar esa información de video (usando la hora exacta hh:mm:ss, por ejemplo, y vincularlos a la hora que proporciona el sistema GPS en el mismo o distinto formato).

Este extenso ejemplo no pretende dejar de lado la premisa fundamental en el análisis de datos; el objetivo no es el volumen de datos, sino las preguntas que queremos responder en nuestro contexto. Los datos serán un facilitador para encontrar respuestas, pero sin olvidar la complejidad del deporte y los atletas.

Al ser conscientes de ese gran volumen de datos que tenemos a nuestra disposición, debemos preguntarnos lo siguiente: ¿cómo podemos ser eficientes para poder proveer respuestas en el menor tiempo posible y que estas tengan un impacto directo en la toma de decisiones del equipo o *staff* en el que trabajamos? Si no somos capaces de usar esa información, debemos reconsiderar nuestro enfoque o modelo de recogida de datos, ya que la información que estamos recolectando no está dando ninguna solución y, posiblemente, esté quitando tiempo a jugadores y al *staff* que se podría usar en otros aspectos.

El ejemplo anterior permite mostrar el volumen y la variedad de datos que somos capaces de registrar. Por otra parte, la dificultad para iniciar un análisis de calidad reside en ser capaces de juntar/cruzar/relacionar dichos datos para aportar mayor contexto e información en nuestro ámbito profesional.

Las posibilidades de aumentar la extensión o el alcance del impacto que tienen los datos físicos son:

---

- Limitadas.
- Ilimitadas.

SUBMIT

### **¿Qué herramientas tenemos para poder analizar la información de la manera más rápida y eficiente posible?**

Existen diversas opciones que nos ayuden a conseguir dicho objetivo, algunas serán de gran utilidad la visualización, y otras con más potencial para la extracción y transformación de datos. Cada parte herramienta tendrá sus ventajas o puntos fuertes en cada parte del proceso de análisis de datos pero debemos escoger la que nos permita una mayor flexibilidad para el desempeño profesional.

El objetivo de este certificado es desarrollar los conocimientos necesarios que nos permitan utilizar una herramienta con un gran potencial en la mayoría de los procesos del análisis de datos.

R y RStudio se presenta como el *software* ideal que cumple las características para potenciar las habilidades de los *sport scientists*, lo que facilita el acceso a la información, al análisis y a la comunicación.

### **¿Qué es R y RStudio?**

R es un *software*, originalmente ideado para el análisis estadístico, de libre descarga, que emplea un lenguaje de programación específico para ejecutar ciertas acciones que deseemos hacer con los datos. RStudio es un entorno de desarrollo integrado con funcionalidades más adaptadas al usuario que permite usar R de manera más cómoda y parecida a los *softwares* más comunes como Excel.

A simple vista, esta primera parte puede parecer muy compleja, los términos programación, *software*, estadística, a menudo, parecen muy alejados del día a día. Sin embargo, vamos a utilizar una analogía de Ismay y Kim (2023) que simplifica la interacción entre R y RStudio. R es el motor del coche y RStudio la parte interior del chasis con la que hacemos funcionar al motor y conducimos el coche (cuadro de cambios, pedales y volante).

Cuando hablamos de programar, nos referimos a escribir código; de nuevo, aunque parezca una terminología avanzada o muy alejada del ámbito de las ciencias del deporte o del rendimiento físico, escribir código puede ser tan simple al principio como pedirle al *software* que haga una operación matemática básica, de manera muy parecida a Excel. Sin embargo, el potencial del *software* reside en que esa programación permite disponer de opciones mucho más avanzadas.

Por lo tanto, necesitaremos instalar en primer lugar R para poder disponer de ese «motor» que ejecutara nuestras «órdenes» y RStudio para crear la interacción entre nuestros datos y R.

¿Qué es R? Selecciona las dos opciones correctas.

---

- Un software ideado para el análisis estadístico.
- Un software que emplea un lenguaje de programación específico para ejecutar ciertas acciones que deseemos hacer con los datos.
- Un entorno de desarrollo integrado con funcionalidades más adaptadas al usuario.

Un hardware diseñado para el control de datos deportivos.

SUBMIT

## Pasos iniciales

Enlace para descargar R:

<https://cran.r-project.org>

Enlace para descargar RStudio según el software de nuestro ordenador:

<https://posit.co>

En nuestra carpeta de aplicaciones, deberemos comprobar que tenemos ambas aplicaciones instaladas.

## Figura 1: R y RStudio



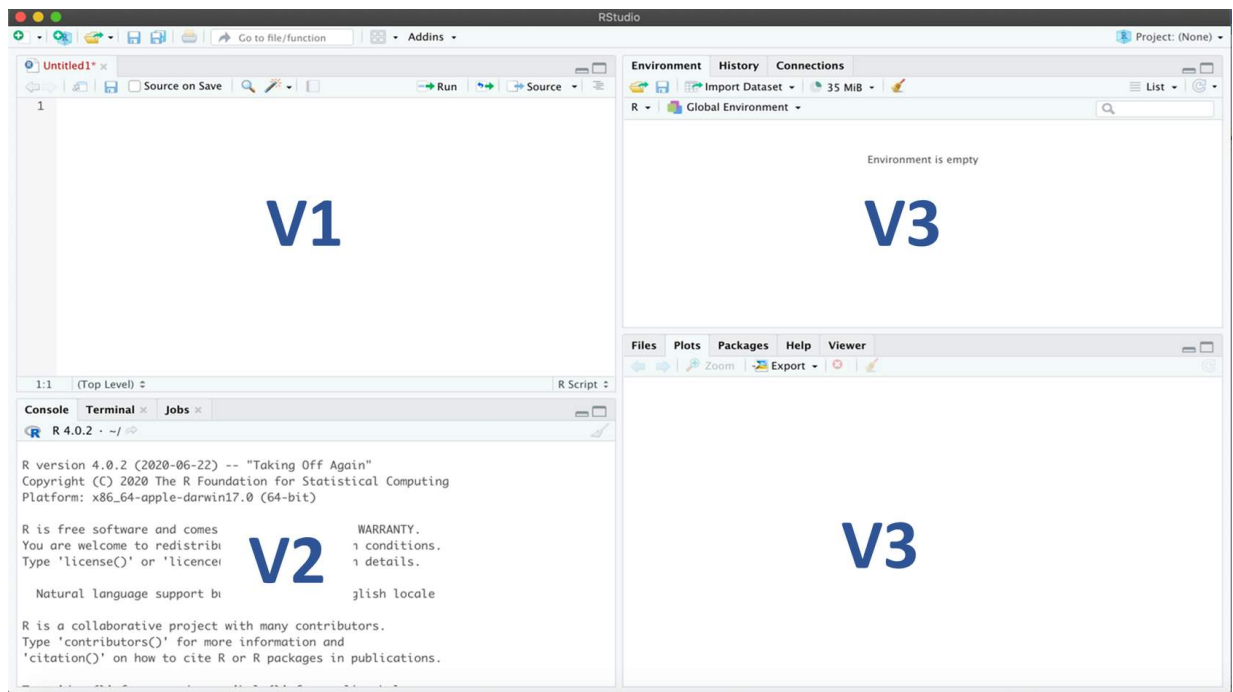
Fuente: Imagen Recuperada de <https://www.linkedin.com/pulse/everything-you-need-know-r-andor-studio-angela-cao/>

---

Abriremos únicamente RStudio, ya que este se encargará, automáticamente, de usar R.

Esta es la pantalla principal que encontraremos.

**Figura 2: Pantalla de RStudio**



Fuente: Elaboración propia.

Se han destacado las cuatro ventanas o paneles principales de RStudio. Los videos que acompañan el curso facilitarán la comprensión de la utilidad de cada una de las ventanas, pero vamos a describir cuáles son las funciones y la descripción de cada una de las pantallas, para que sirvan de referencia durante los videos.

**V1- Panel editor:** aquí es donde escribiremos el código para ejecutar todas las acciones que pretendamos (importar los datos, crear variables, aplicar modelos estadísticos, etcétera).

**V2- Panel de la consola:** en este panel, podremos ver los resultados de la ejecución del código que hemos escrito.

**V3- Panel de entorno/de trabajo:** en este panel, encontraremos los distintos objetos, vectores, variables o tablas que hemos creado.

**V4- Panel de archivos y gráficos:** encontraremos la carpeta en la que estamos trabajando, los archivos que podemos importar y la visualización de los gráficos que hemos creado.

A continuación, también se recoge un listado de términos comunes con el uso de RStudio que pueden servir como referencia durante todo el certificado.

- **Ejecutar código:** se refiere a accionar todo lo que hemos escrito, es decir, pedirle a R que haga lo que queramos después de haberlo programado. Puede ser tan sencillo como dar clic a un botón o a dos teclas del teclado. La clave será tener un código sin errores para que se ejecute correctamente.
- **Sintaxis del código:** como en cualquier lenguaje, en programación, hay una serie de reglas que hay que respetar para que se pueda ejecutar el código correctamente. Por ejemplo, si el nombre de nuestra tabla contiene mayúsculas, debemos escribirlo de la misma manera.
- **Funciones:** elementos que desarrollan tareas en R. Exactamente igual que en Excel, usamos el nombre

de la función, por ejemplo, SUM para las sumas, añadimos los valores que queremos sumar y nos da un resultado después de ejecutar el código.

- **Paquetes o librerías:** RStudio tiene unas funcionalidades básicas, pero, al ser una herramienta gratuita y colaborativa, hay usuarios que han desarrollado funcionalidades extra, como, por ejemplo, combinaciones de código propias que permiten hacer visualizaciones más avanzadas utilizando una función sencilla (cuando utilizamos esas funciones, escribiendo una sola palabra, podemos obtener el mismo resultado que escribiendo veinte líneas de código). Hay que instalar dichos paquetes (a continuación).
- **Objetos:** son valores/tablas/texto/listas de números que hemos guardado en RStudio para uso posterior; si volvemos a utilizar Excel como ejemplo, serían valores o texto que guardamos en una celda para poder referenciar luego en nuestros cálculos o análisis.
- **Data frames/matrix/tablas:** distintos tipos de tablas, es decir, objetos que sirven para almacenar información.
- **Directorio:** de manera muy simple, carpeta desde la que estamos trabajando, podremos utilizar los

archivos que hay en ella y, si producimos resultados, también se pueden guardar en esta.

- **Script:** archivo de código que podemos guardar para utilizar varias veces.

Para finalizar esta introducción, vamos a ver uno de los primeros pasos que se deben llevar a cabo al empezar a trabajar con RStudio: instalar paquetes para poder utilizar sus funciones. Con este ejemplo, repasaremos la interfaz de nuevo y algunos de los conceptos del glosario anterior.

Como hemos descrito anteriormente, los paquetes son ese conjunto de funciones que nos permiten agilizar nuestro trabajo, han sido desarrolladas por otros programadores y no pertenecen a las funciones básicas de R; por lo tanto, hay que instalarlos.

Esto también se repetirá a lo largo del certificado, pero es una premisa que se repite, habitualmente, en R; el *software* permite múltiples maneras de conseguir el mismo resultado. Habrá que elegir con la que nos sentimos más cómodos para utilizarla en el futuro.

En este ejemplo de la instalación, vamos a instalar un paquete llamado «ggplot2»; es uno de los paquetes más utilizados en RStudio para la creación de gráficos y visualizaciones.

Un archivo de código que podemos guardar para utilizar varias veces, se denomina

Type your answer here

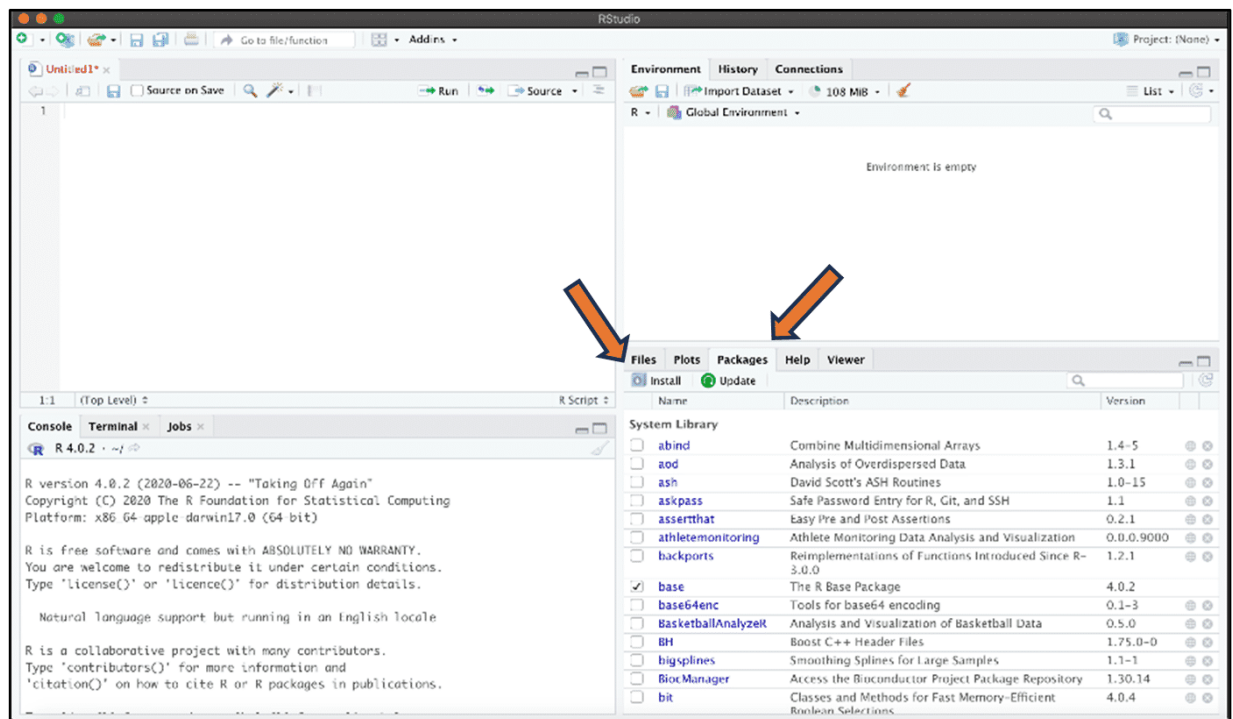
---

SUBMIT

### Primera opción

- RStudio dispone, en una de las ventanas (V3 mencionada anteriormente), una pestaña que permite instalar los paquetes. Daremos clic en esa pestaña, Packages, y luego a la opción Install.

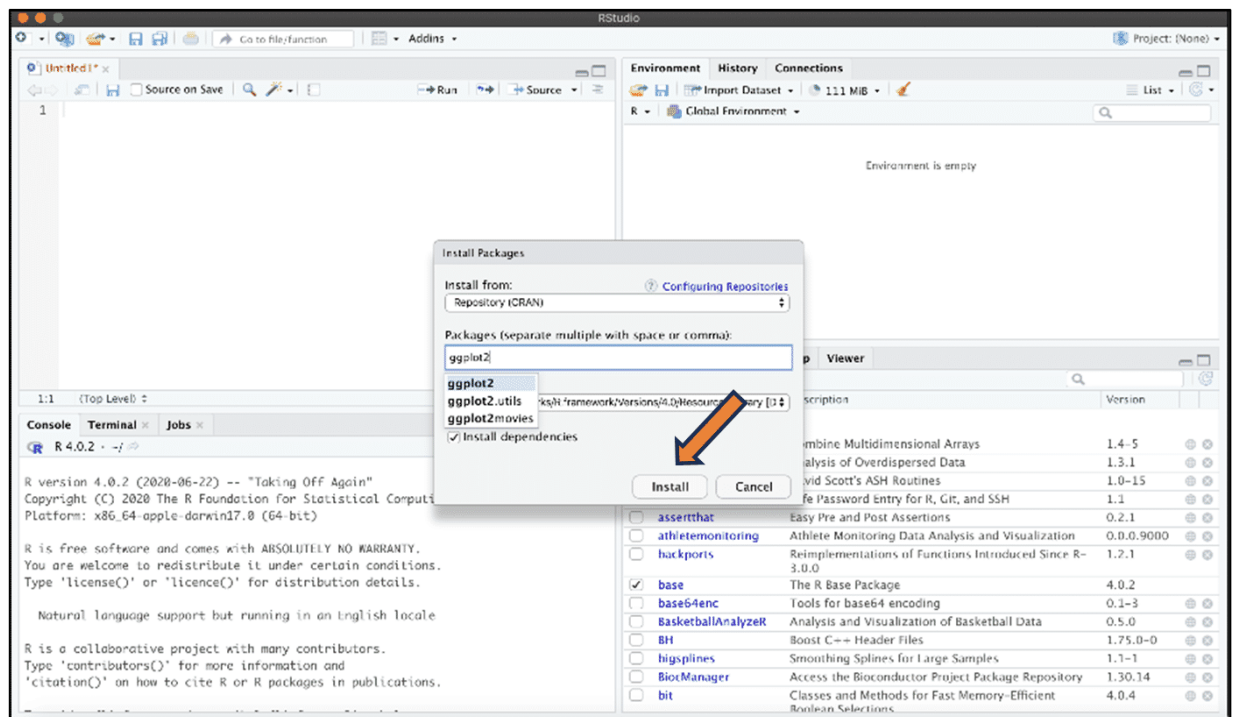
### Figura 3: Primera opción



Fuente: Elaboración propia.

- Aparecerá una ventana en la que debemos escribir el nombre el paquete que queremos instalar; aparecen, automáticamente, sugerencias. Al hacer clic en el botón Install, dicho paquete quedará instalado en nuestro software.

**Figura 4: Clic en el botón Install**



Fuente: Elaboración propia.

Este es un proceso que hay que hacer solo una vez, es decir, cuando hayamos instalado el paquete, este estará disponible en nuestro *software* para las veces que lo utilicemos en el futuro.

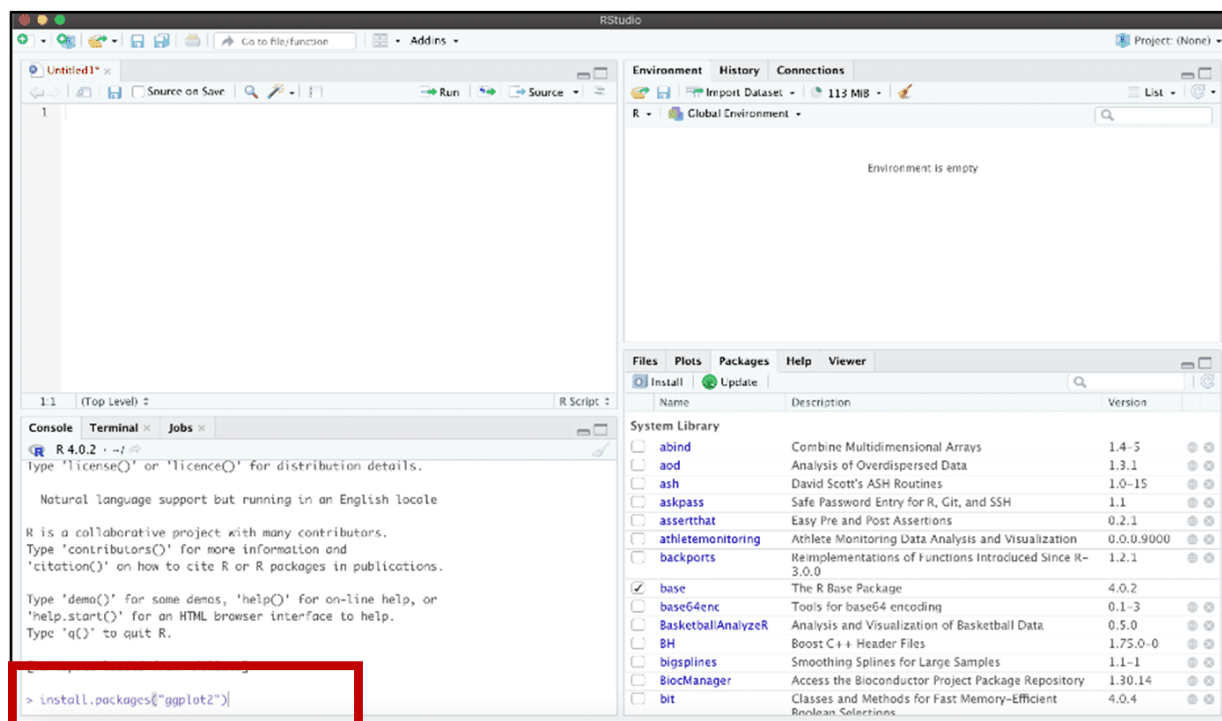
## Segunda opción

- En este caso, vamos a escribir código. Aunque anteriormente hemos comentado que el código se escribe en el panel Editor (V1), el panel Consola también permite escribir en él. Habitualmente, el código que queramos usar solo una vez se puede

escribir en el panel Consola. Como hemos dicho anteriormente, vamos a instalar los paquetes una vez, por lo que, en este caso, podemos usar la consola.

- Las funciones son elementos que desarrollan tareas y, en este caso, queremos usar la función de instalar paquetes. Así pues, vamos a utilizar la función `install.packages`. A esta función hay que darle información para que pueda ejecutarse; en este caso, la información es el nombre del paquete `ggplot2`.
- La información que les damos a las funciones siempre va entre paréntesis y, al ser `ggplot` texto, lo debemos poner entre comillas.
- El texto que debemos escribir es `install.packages("ggplot2")`.
- Damos a la tecla Intro y R se encargará de ejecutarlo e instalarlo en el *software*.

**Figura 5: Segunda opción**



Fuente: Elaboración propia.

## Funcionalidades y código

En los videos de este primer módulo, únicamente, veremos ejemplos del resultado final; la sintaxis correspondiente se verá en los próximos cursos cuando nuestra destreza con el programa sea mayor, pero pretenden mostrar el potencial de la herramienta. El código y los datos utilizados en todos los videos estarán disponibles en el material extra del curso.

## Funcionalidades y ejemplos en el material de video

- Importar datos.
- Crear variables.
- Visualizaciones.
- Cruzar datos de distintas fuentes.
- Crear *apps* a medida (en cursos posteriores).
- Reportes automatizados (en cursos posteriores).
- Pruebas estadísticas (en cursos posteriores).

CONTINUAR

## Referencias

---

**Almulla, J., Takiddin, A., y Househ, M.** (2020). The use of technology in tracking soccer players' health performance: a scoping review. *BMC Medical Informatics and Decision Making*, 20(1), 184. <https://doi.org/10.1186/s12911-020-01156-4>.

**Bromley, T., Turner, A., Read, P., Lake, J., Maloney, S., Chavda, S., y Bishop, C.** (2021). Effects of a competitive soccer match on jump performance and interlimb asymmetries in elite academy soccer players. *Journal of Strength and Conditioning Research*, 35(6), 1707-1714. <https://doi.org/10.1519/JSC.0000000000002951>.

**Caro, E., Campos-Vázquez, M. Á., Lapuente-Sagarra, M., y Caparrós, T.** (2022). Analysis of professional soccer players in competitive match play based on submaximum intensity periods. *PeerJ*, 10, e13309. <https://doi.org/10.7717/peerj.13309>.

**Hernández-Belmonte, A., Alegre, L. M., y Courel-Ibáñez, J.** (2023). Velocity-Based Resistance Training in Soccer: Practical Applications

and Technical Considerations. *Strength and Conditioning Journal*, 45(2), 140-148. <https://doi.org/10.1519/SSC.0000000000000707>.

**Imagen** **Recuperada** **de**  
<https://www.linkedin.com/pulse/everything-you-need-know-r-andor-studio-angela-cao/>

**Ismay, C., y Kim, A. Y.** (2023). *A ModernDive into R and the Tidyverse* (Foreword by K. S. McConville). <https://moderndive.com/index.html>.

**Kuhlman, N. M., Jones, M. T., Jagim, A. R., Feit, M. K., Aziz, R., y Crabill, T.** (2023). Relationships between external loads, sRPE-load, and self-reported soreness across a men's collegiate soccer season. *Biology of Sport*, 40(4), 1141-1150. <https://doi.org/10.5114/biolsport.2023.125587>.

**Miller, D. J., Sargent, C., y Roach, G. D.** (2022). A validation of six wearable devices for estimating sleep, heart rate, and heart rate variability in healthy adults. *Sensors*, 22(16), 6317. <https://doi.org/10.3390/s22166317>.

CONTINUAR