# Module 1. Introduction to software and functionalities

**Why does a sports scientist need to use advanced data analysis software?**

As physical performance professionals, we seek to answer multiple questions that arise on a daily basis, as well as to develop long-term projects related to the research of physical qualities, their development and their impact on performance in the field.

In recent years, the volume of information available, collected by diverse sources and technologies in different formats, has grown exponentially (Almulla et al., 2020). The need to use this information to gain insight and a competitive advantage over other organizations has fuelled the growth of the field of data analytics in the field of sports performance.

We will provide an example of the information we could have during a football training, focusing only on the information that has to do with the physical component, which is only one of the pieces of the puzzle to achieve our main objective: to perform and win. This example aims to reflect the diversity and volume of data we collect in our context and the raw product that is available to carry out analyses that impact decision-making.

**Data source 1:** the player has a movement and heart rate monitoring bracelet that they use during the night for the calculation of derived variables related to sleep, rest and recovery (Miller et al., 2022).

Using one of the most used and valid trademarks as an example, in the morning, we get information summarizing the night through .csv files that present the following information:

- Three .csv files
  - The first with physiological cycles.
  - The second with the sleep record.
  - The third with information on physical activity.

- Each file contains more than ten columns or variables.

- The format of the columns may vary.

  - Date.
  - Hours.
  - Text.
  - Numerical.

**Data source 2:** after a short activation at the gym, the player follows a protocol of standardized jumps on a force platform with registry at 1000 Hz (Bromley et al., 2021).

Most technology companies with force platforms have specific software to display and download or export results.
The information we can get is as follows:

- An Excel file.
- First eight rows with player-identifying information.
- A row with the average of all the results. Also in different formats and with as many variables as we wanted to select (1 to >50).

**Table 1: Excel file**

| | 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 | Additiona | Concentri | Concentri |
| 11 | | | | | | | | | | | |

Source: Author's own production.

**Data source 3:** during the gym session, the performance department records information on the characteristics of the movement being made, through the use of a linear encoder (Hernández-Belmonte et al., 2023). The information we can get is as follows:

- A file, but in .csv format.
- The first row includes relevant information within each cell (example in the image).
- The columns are not summaries in this case, but display the information for each repetition.

**Table 2: Example**

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | first name: Miguel | last name: Vazquez | klipfolioID: | date: | time: 12:0 | exercise: |
| 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 | - | - | | |

Source: Author's own production.

**Data source 4:** the player has an LPS device that records locomotor and mechanical activity during training (Caro et al., 2022).
After training, a file with similar features to the previous two is downloaded:

- Excel or csv file, depending on the trademark.
- Identifiers in the first rows (date/number of players/team).
- Various columns with the variables to be analysed that we have selected.

    - In this case, summary variables are also available.
    - One outstanding feature is that the summary is made for each of the periods that we select, for example:

        - Warm-up.
        - Task 1.
        - Task 2.
        - Full session.

**Data source 5:** At the end of training, a member of the performance staff asks the player about their perception of the effort made in the session (Kuhlman et al., 2023). There are multiple options for this data collection, from a customized Excel, to using apps available on the market.

Once again, we get a file with numerical answers and columns that reflect the variables we want to analyse.

The amount of information in the example above varies according to the sport, the category and the context (human and material resources, influence of the staff, player-acceptance to such registry, etc.), but the possibilities, even if the volume of data is smaller in other contexts, are limitless. As mentioned before, there are cases in which these data sources are limited, but we find contexts in which the amount of information may be greater (information on body composition, nutrition or caloric intake, wellness questionnaires, etc.).

There are unbounded possibilities for increasing the extent or scope of the impact that physical data has. Let us imagine other types of data sources, such as video cameras with which we cross-reference match actions whose workload we want to know. At what times is the player sprinting? During defensive or offensive actions? We need to cross-reference that video information (using the exact time hh:mm:ss, for example), and link it to the time provided by the GPS system in the same or different format.

This extensive example does not leave aside the fundamental premise in data analysis: the goal is not the volume of data, but the questions we want to answer in our context. Data will be an enabler to find answers, but considering the complexity of sport and athletes.

Being aware of this large volume of data that we have at our disposal, we must ask ourselves the following: how can we efficiently be able to provide answers in the shortest time possible so that these have a direct impact on the decision-making of the team or staff with which we work? If we are not able to use that information, we must reconsider our approach or data collection model, since the information we are collecting is not providing any solution and is potentially taking time away from players and staff that could be devoted to other issues.

The above example shows the volume and variety of data that we are able to collect. On another note, the main difficulty to set up a quality analysis lies in being able to gather/cross-reference/relate these data to provide more context and information in our professional field.

Las posibilidades de aumentar la extensión o el alcance del impacto que tienen los datos físicos ¿son limitadas o ilimitadas?

Ilimitadas

**What tools enable us to analyse the information as quickly and efficiently as possible?**

There are several options to help us achieve this goal, some will be very useful for visualization, and others have more potential for data extraction and transformation. Each tool has advantages or strengths in each part of the data analysis process, but we must choose the one that provides greater flexibility for professional performance.

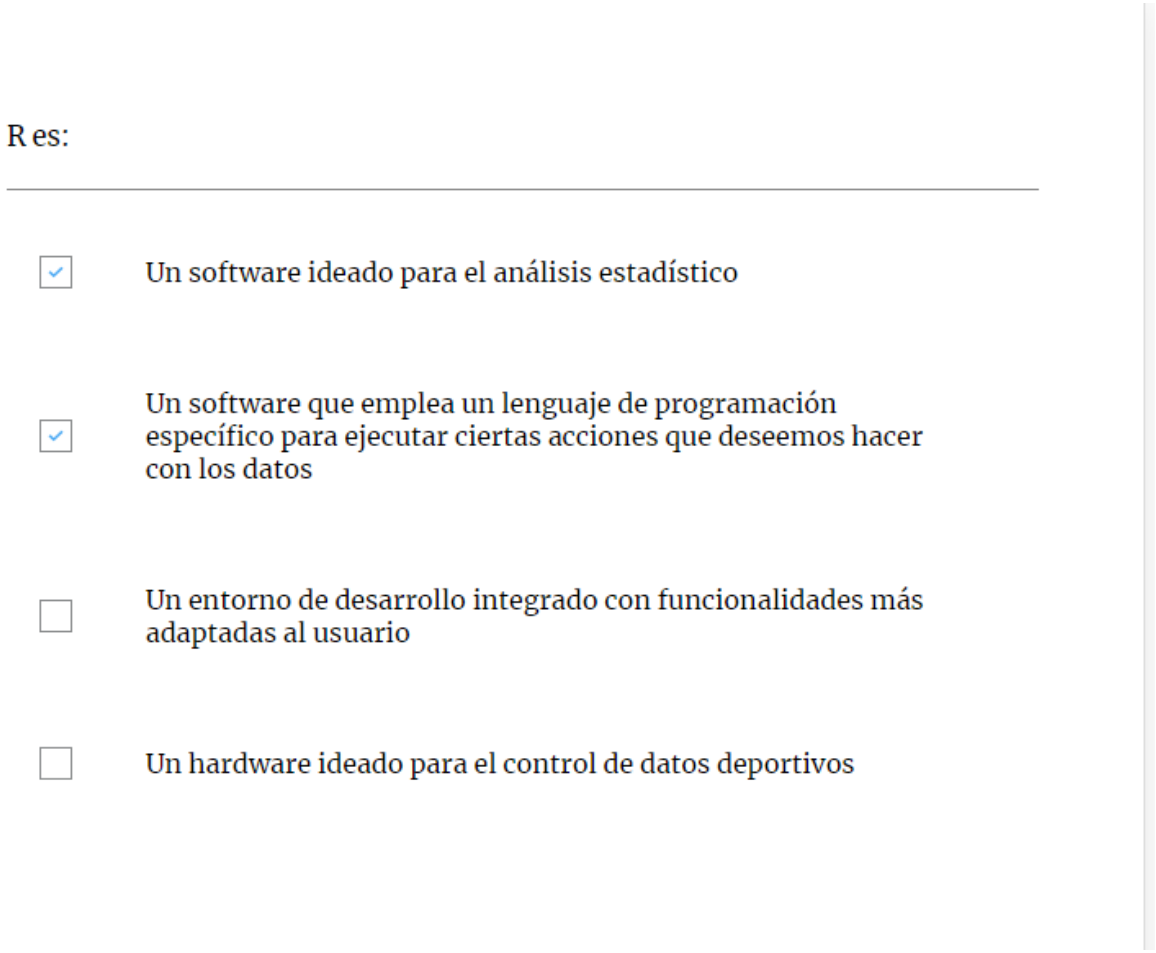
The objective of this certificate is to develop the knowledge to use a tool with great potential in most data analysis processes. R and RStudio were introduced as the ideal software with characteristics to enhance the skills of sports scientists, which facilitate access to information, analysis and communication.

**What are R and RStudio?**

R is free software originally designed for statistical analysis, which uses a specific programming language to execute certain actions that we want to do with the data. RStudio is an integrated development environment with more usesr-friendly functionalities that allows you to use R in a more comfortable way and similar to the most common software such as Excel.

At a first glance, this first part may seem very complex; the terms programming, software, and statistics, often seem very far from day-to-day life. To clarify, we will resort to an analogy from Ismay and Kim (2023) that simplifies the interaction between R and RStudio. R is the car's engine and RStudio is the inner part of the chassis with which we run the engine and drive the car (gearbox, pedals and steering wheel).

When we talk about programming, we mean writing code; again, although it may seem advanced terminology or far removed from the field of sports science or physical performance, writing code can be as simple as asking the software to do a basic mathematical operation, much like Excel. However, the potential of the software lies in the fact that this programming allows for much more advanced options.
Therefore, we will need to install R first so as to have the "engine" that executes our "commands" and RStudio to create the interaction between our data and R.




**First Steps**

Link to download R:
https://cran.r-project.org

Link to download RStudio according to our computer's software:
https://posit.co

In our Applications folder, we must check that both applications are installed.

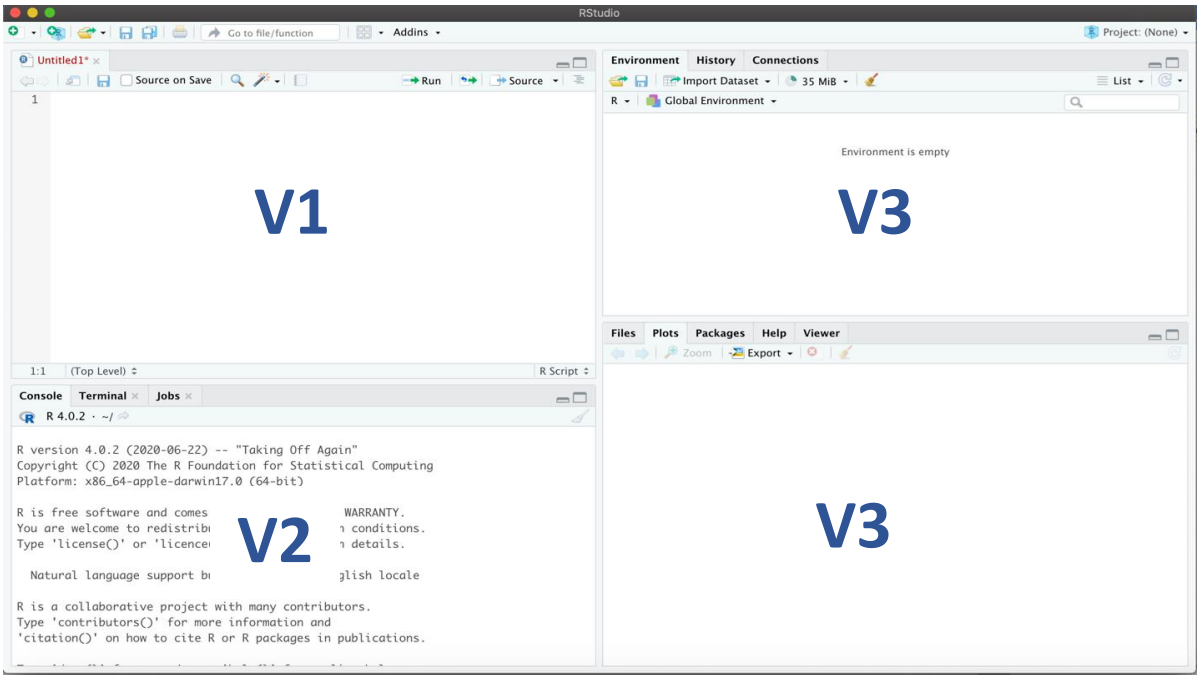
**Figure 1: R and RStudio**

Source: Image retrieved from https://www.linkedin.com/pulse/everything-you-need-know-r-andor-studio-angela-cao/

We will only open RStudio, since it will automatically use R.

This is the main screen that we will see.

**Figure 2: RStudio display**



Source: Author's own production.

The four main windows or panels in RStudio have been highlighted. The videos that are part of the course will aid the understanding of the usefulness of each of the windows, but we are going to describe each of the screens and their functions, so that this is a point of reference when you watch the videos.

**V1- Editor panel:** this is where we will write the code to execute all the actions we intend (import the data, create variables, apply statistical models, etc.).
**V2- Console panel:** in this panel we can see the results of the execution of the code we will have written.
**V3- Environment/work panel:** in this panel we will find the different objects, vectors, variables or tables that we have created.
**V4- Files and graphics panel:** we will find the folder in which we are working, the files that we can import and the visualization of the graphics that we have created.

Below there is a list of common terms used in RStudio that can also serve as reference throughout the course.

- **Execute code:** it refers to acting on everything we have written, that is, asking R to do what we want after programming it. It can be as simple as clicking a button or two keys on the keyboard. It will be essential to have an error-free code for it to run correctly.

- **Code syntax:** as in any language, in programming there are a series of rules that must be adhered to so that the code can be executed correctly. For example, if the name of our table is capitalized, we should write it in the same way.

- **Functions:** these are elements that perform tasks in R. Exactly like in Excel, we use the name of the function, for example, SUM for sums, we add the values we want to add and it gives us a result after executing the code.

- **Packages or libraries:** RStudio has some basic functionalities, but, as it is a free and collaborative tool, there are users who have developed extra functionalities, such as, for example, their own code combinations that enable a more advanced visualizations using a simple function (when we use these functions, writing a single word, we can get to the same result as when writing twenty lines of code). You need to install these packages (see below).

- **Objects:** these are values/tables/text/lists of numbers that we have saved in RStudio for later use; if we take Excel as an example, they would be values or text that we save in a cell in order to reference them later in our calculations or analysis.

- **Data frames/matrix/tables:** these are different types of tables, i.e. objects that are used to store information.

- **Directory:** simply put, it is the folder from which we are working. We can use the files that are in it and, if we produce any results, they can also be saved there.

- **Script:** it is code file that we can save to use several times.

To conclude with this introduction, we will look at one of the first steps to take when you start working with RStudio: install packages in order to use their features. With this example, we will review the interface and some of the concepts from the glossary above.
As described above, packages are the set of functions that allow us to speed up our work; they have been developed by other programmers and are not part of the basic functions in R; therefore, they must be installed.

This will also be repeated throughout the course, but it is a premise that is usually repeated in R; The software allows for multiple ways to achieve the same result.
We will have to choose the one we feel most comfortable with to use in the future.

In this installation example, we are going to install a package called "ggplot2", one of the most widely used packages in RStudio to create graphs and visualizations.

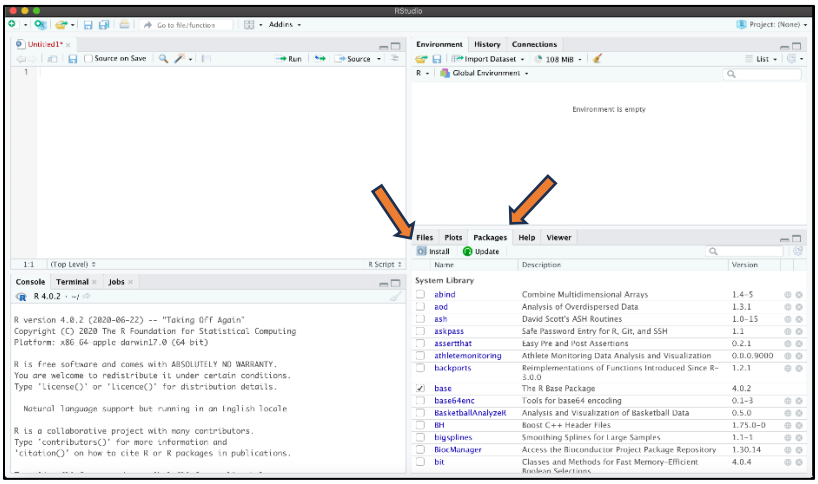
Un archivo de código que podemos guardar para utilizar varias veces se lo denomina:

Script

### First option

- In one of the windows of RStudio there is a tab that allows you to install the packages (V3 mentioned above). We will click on that tab, Packages, and then the Install option.

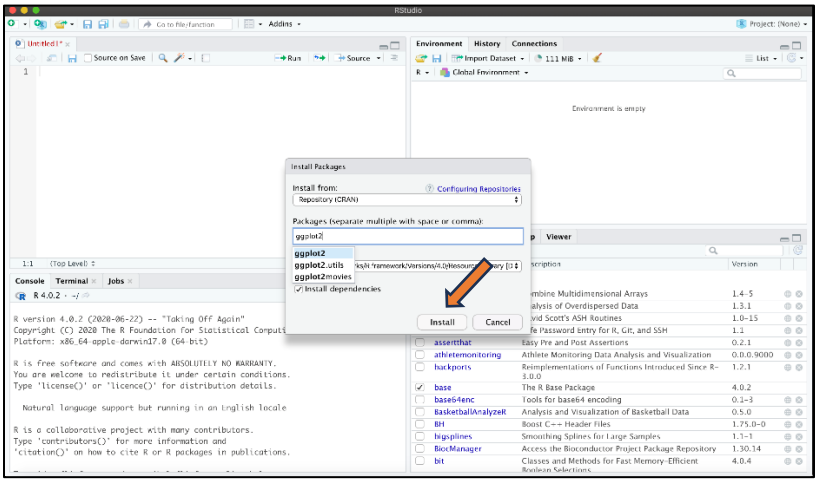
### Figure 3: First option



Source: Author's own production.

- A window will pop up; we have to type the name of the package we want to install (suggestions appear automatically). By clicking the Install button, the package will be installed in our software.

### Figure 4: Click on the Install button



Source: Author's own production.

This process has to be done only once; that is, once we have installed the package, it will be available in our software for the future.

### Second option

- In this case, we are going to write code. Even though we have mentioned that the code is written in the Editor panel (V1), the Console panel also allows you to write in it. Usually, code that we want to use only once can be written in the Console panel. As said before, we are going to install the packages once, which means that in this case, we can use the Console.

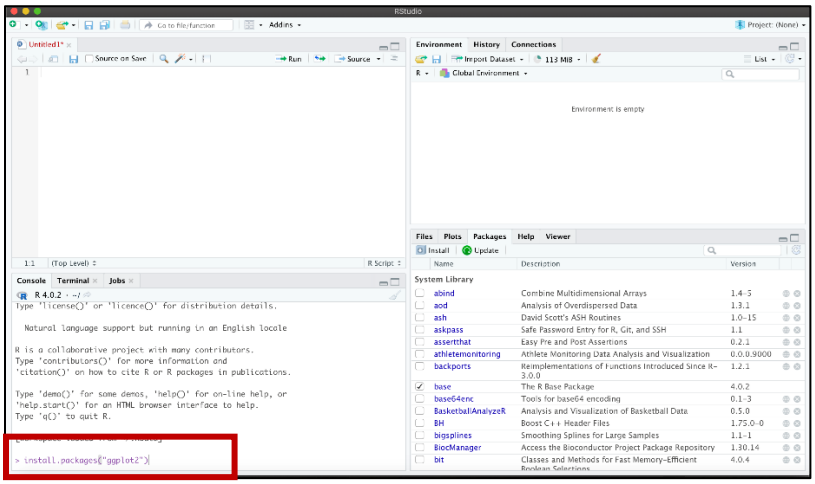
- Functions are elements that perform tasks, and in this case, we want to use the install packages function. So, we will use the install.packages function. This function must be given information so that it can be executed; In this case, the information is the name of the ggplot2 package.

- The information we give to the functions is always in between parentheses and, since ggplot is text, we must put it in between quotation marks.

- The text we need to type is install.packages("ggplot2").

- We hit the Enter key and R will take care of executing and installing it in the software.

### Figure 5: Second option




Source: Author's own production.

## Functionalities and code

In the videos of this first module, we will only see examples of the final result, but they intend to show the potential of the tool; the corresponding syntax will be seen in the next courses when our skill with the program is greater. The code and data used in all videos will be available in the course additional material.

## Functionalities and examples in the video material

- Import data.
- Create variables.
- Views.
- Cross-referencing data from different sources.
- Create custom apps (in subsequent courses).
- Automated reports (in subsequent courses).
- Statistical tests (in subsequent courses).

## References


**Almulla, J., Takiddin, A., and 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., and 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., and 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., and 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.

**Image retrieved from** https://www.linkedin.com/pulse/everything-you-need-know-r-andor-studio-angela-cao/

**Ismay, C., and 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., and 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., and 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.