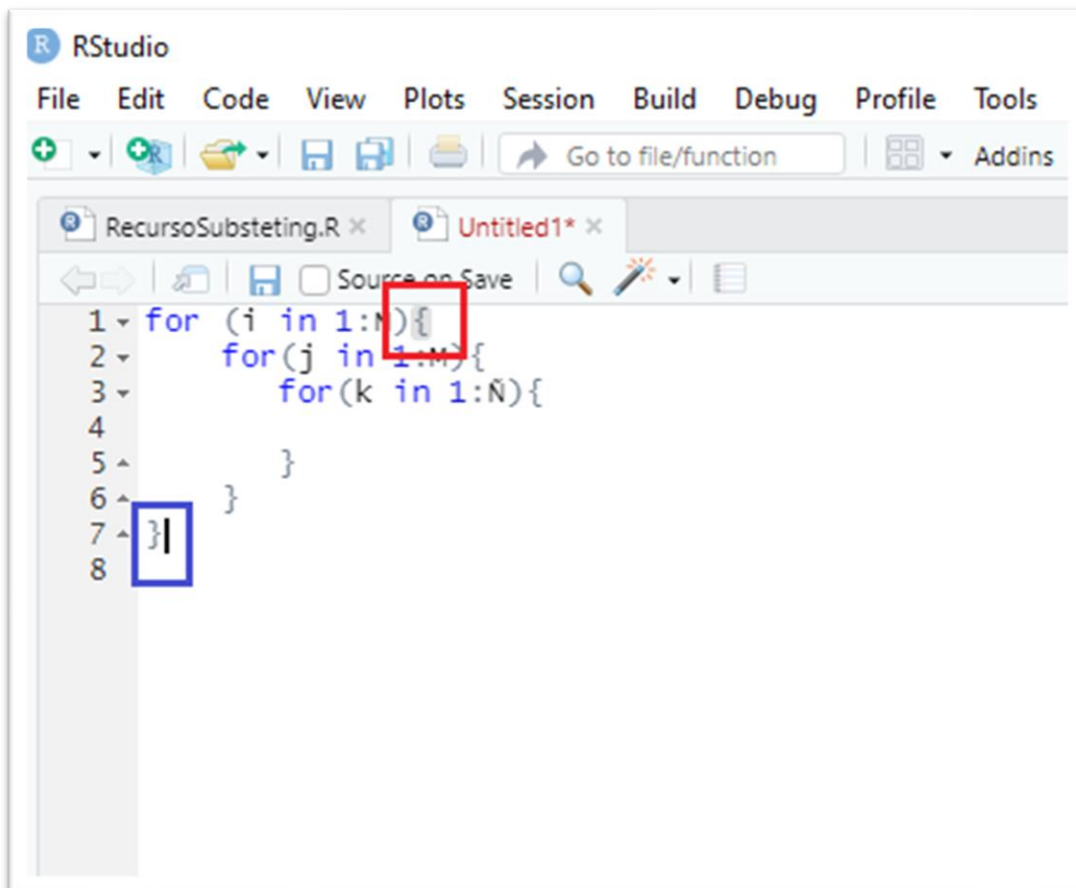
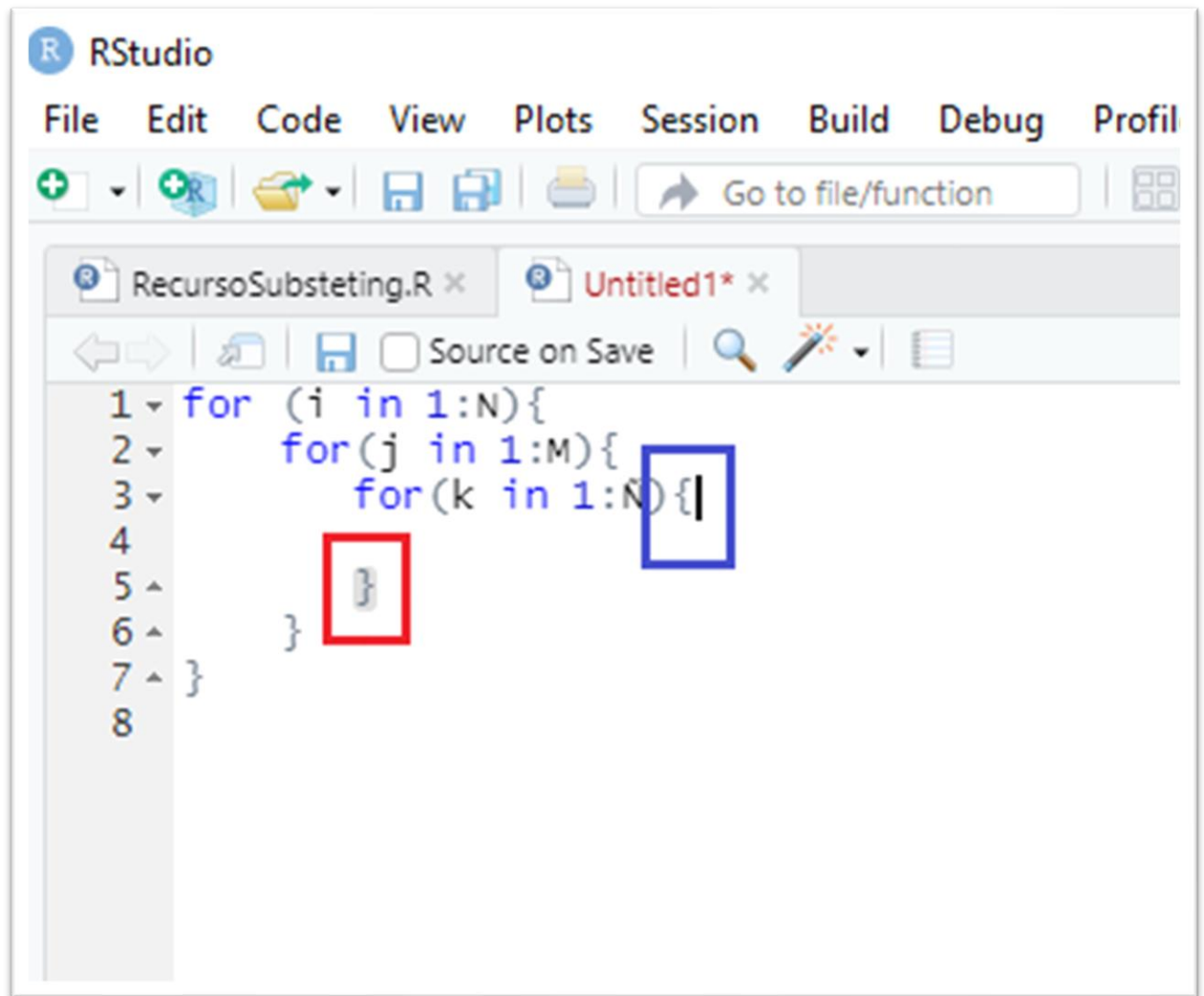


## 1. Permite ver los cierres de llaves en un bucle o función

R Studio permite ver las "parejas de llaves" en un bucle o una función simplemente clicando a la derecha de una de ellas. Al clicar a la derecha de una llave de apertura, nos indicará, sombreado de gris, la llave que la cierra, y viceversa, si clicamos a la derecha de una llave de cierre, se sombreadá de gris la llave de apertura a la que está cerrando.

Esta utilidad nos permite evitar errores fáciles de cometer, al dejarnos una llave sin poner o, por el contrario, al poner una llave extra, lo cual daría lugar a que el bucle o la función no funcionaran. Podemos utilizar este recurso para asegurarnos de que la última llave de cierre está cerrando a la primera llave de apertura, y que por lo tanto, hay el mismo número de llaves de cierre que de apertura.





## 2. Autocompletado del nombre de funciones/variables

Este recurso evita que el programador cometa errores ortográficos al autocompletar el nombre de funciones o variables previamente creadas. Es de especial utilidad cuando hemos creado variables/funciones con nombres largos y complejos, o cuando hemos creado muchas variables/funciones con nombres parecidos.

El funcionamiento del recurso es muy simple. Al escribir las 3 primeras letras de la variable/función, aparecerá una lista que se desplegará hacia abajo con las variables/funciones sugeridas por R Studio debido a que empiezan por dichas 3 letras que hayamos escrito. Podemos seleccionar la que deseemos clicando sobre ella directamente o bajando en el desplegable con las flechas del teclado y luego pulsando Enter.

En el caso de las funciones, R Studio nos sugerirá tanto funciones creadas por nosotros mismos, que aparecerán las primeras en el desplegable, como funciones nativas de la programación en R, que aparecerán después.

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

+ + R ↗ Save Undo Print Go to file/function Addins

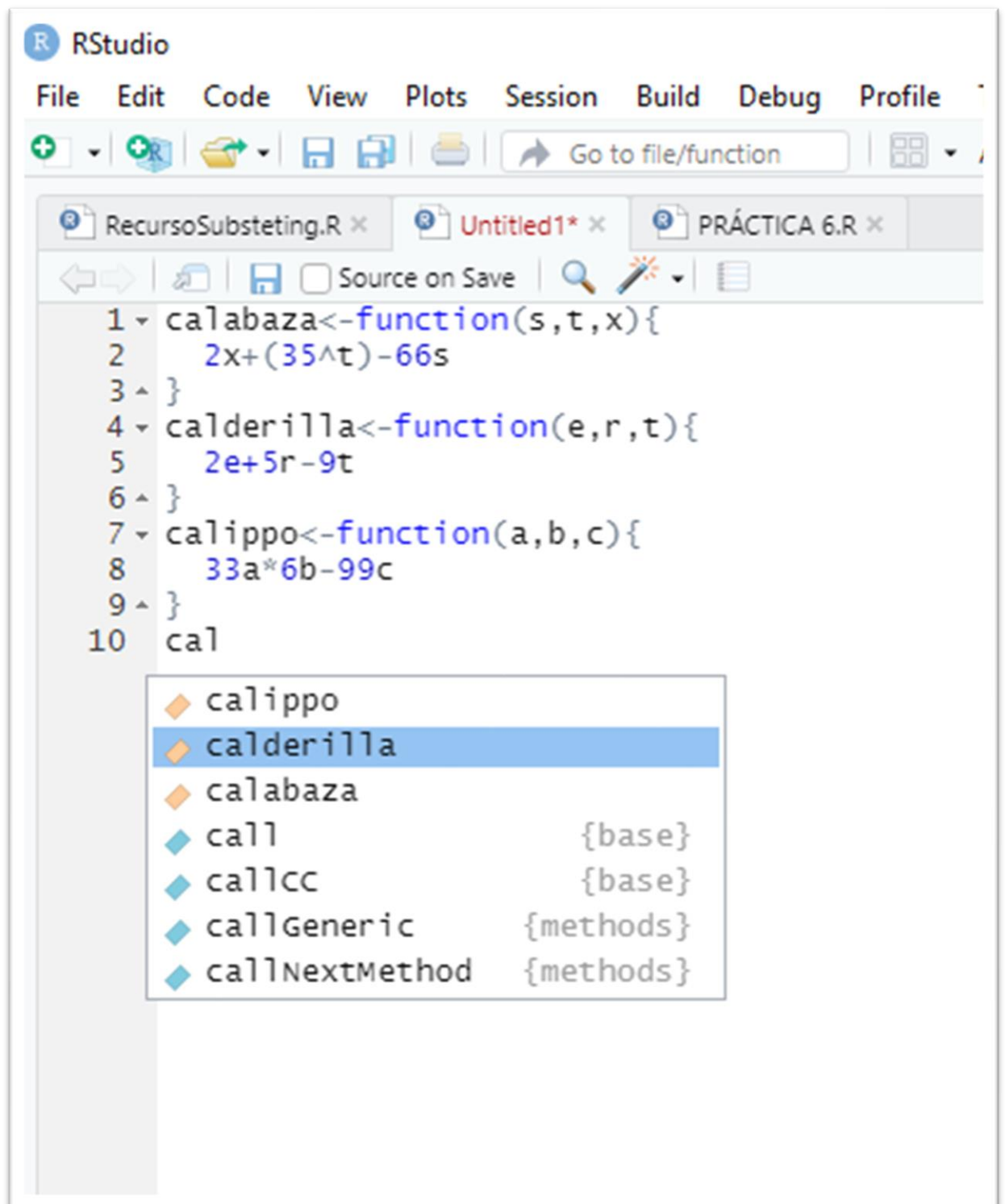
RecursoSubsteting.R x Untitled1\* x PRÁCTICA 6.R x

← → Source on Save 🔍 ✨

```
1 caravana<-3
2 caracas<-5
3 caramelo<-8
4 carrefour<-8
5 carlos<-99
6
7
8 22+car
```

- carlos
- carrefour
- caramelo
- caracas
- caravana
- cars

The image shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for file operations and a search bar. The editor window has three tabs: 'RecursoSubsteting.R', 'Untitled1\*', and 'PRÁCTICA 6.R'. The code in the editor consists of eight lines: five variable assignments (caravana, caracas, caramelo, carrefour, carlos) and a final expression '22+car'. A dropdown menu is open below the code, listing variables: 'carlos' (selected), 'carrefour', 'caramelo', 'caracas', 'caravana', and 'cars'. Each variable is preceded by a small diamond icon.

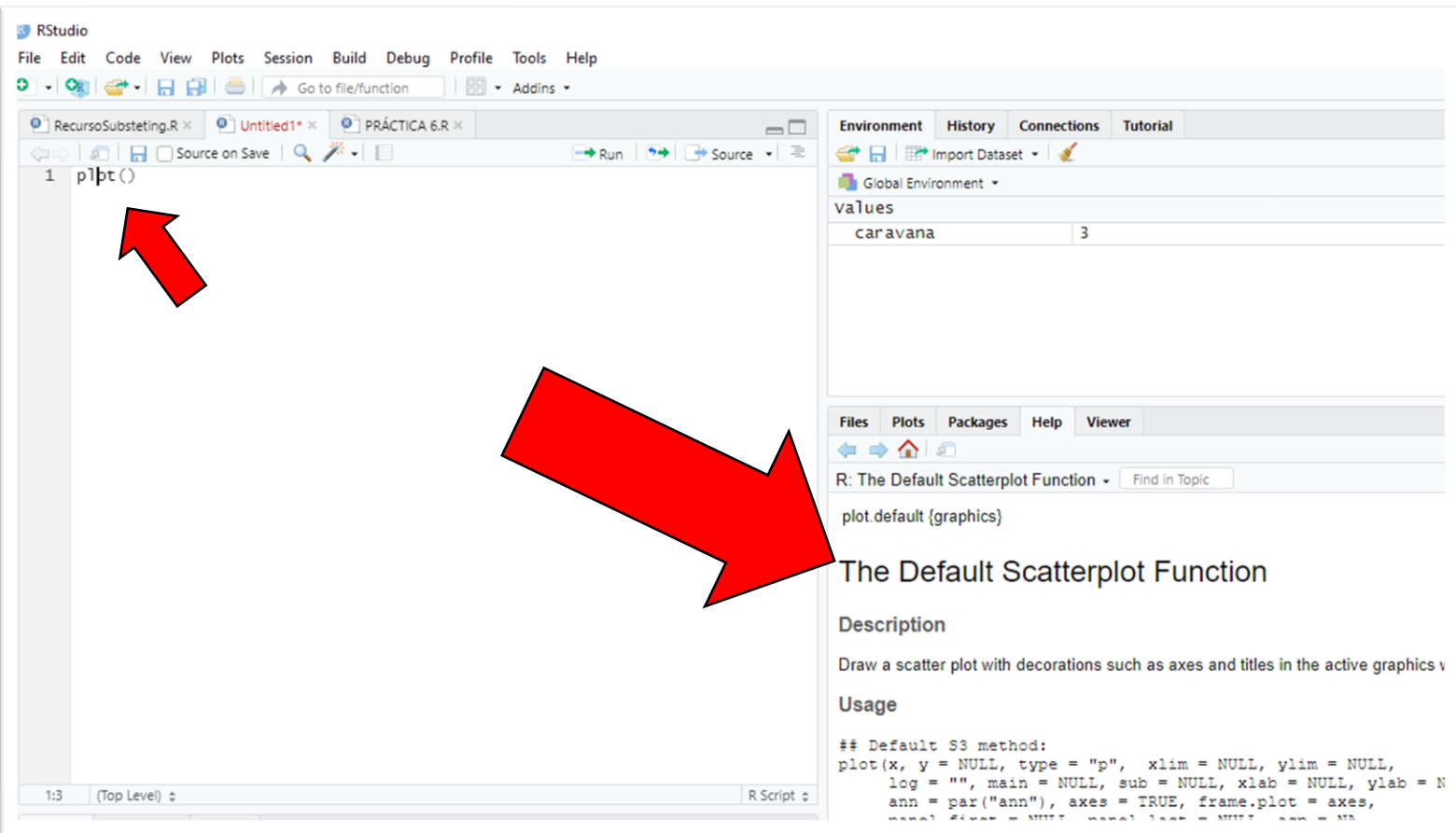


### 3. Solicitar manual de ayuda

Esta función no es exclusiva de R ni de R Studio, sino que es una herramienta "universal" de programación, presente en la mayoría de lenguajes de programación.

Una vez hemos escrito una función, podemos abrir un manual de ayuda sobre como utilizarla clicando sobre ella y pulsando las teclas (Fn+F1) en el teclado. En el manual

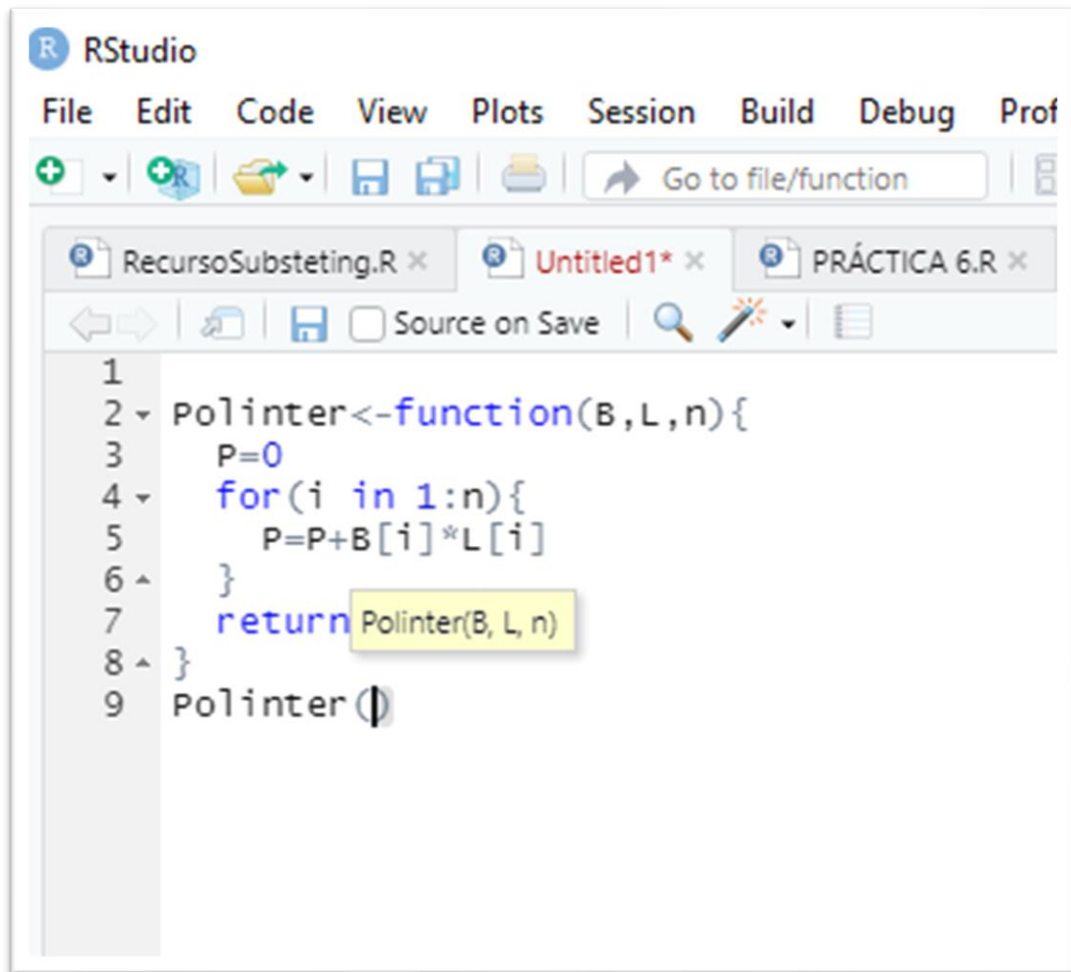
se nos instruirá acerca de cómo usar la función y cuáles son sus argumentos, así como otras notas y detalles a tener en cuenta.



The screenshot shows the RStudio interface. In the script editor, the first line of code is `plot()`. A red arrow points to this code. The help pane on the right displays the documentation for the `plot.default` function. A larger red arrow points from the `plot()` code to the help pane. The help pane shows the title "The Default Scatterplot Function", a description: "Draw a scatter plot with decorations such as axes and titles in the active graphics v", and usage information including the S3 method signature: `plot(x, y = NULL, type = "p", xlim = NULL, ylim = NULL, log = "", main = NULL, sub = NULL, xlab = NULL, ylab = N, ann = par("ann"), axes = TRUE, frame.plot = axes, ...)`.

#### 4. Orden de las variables de una función

En R Studio, una vez hayamos escrito el nombre de una función previamente creada y los paréntesis, aparecerá una pequeña pestaña de color amarillo en la que se nos mostrará el orden de las variables, creado por nosotros mismos cuando hemos definido la función previamente. Es importante escribir los paréntesis; solo con el nombre de la función no aparecerá la pestaña amarilla.



The screenshot shows the RStudio interface with a code editor window. The code is as follows:

```
1
2 Polinter<-function(B,L,n){
3   P=0
4   for(i in 1:n){
5     P=P+B[i]*L[i]
6   }
7   return Polinter(B, L, n)
8 }
9 Polinter()
```

The code is partially folded. Lines 2 through 8 are collapsed, indicated by a downward arrow on line 2 and a bidirectional arrow on line 8. A yellow highlight is visible over the `return` statement on line 7.

## 5. Plegamiento de líneas de código

A la hora de programar programas muy largos y complejos, este es un recurso extremadamente útil, ya que ayuda al programador a organizar y clasificar sus líneas de código, pudiendo plegarlas o desplegarlas cuando le sea necesario. Distinguimos 2 tipos:

### 5.1. Plegamiento en bucles y funciones

R Studio permite plegar funciones y bucles simplemente clicando sobre la flechita que aparece al lado del número de la línea de código en la que empieza el bucle/función. Si hubiera bucles dentro de funciones, podríamos plegar tanto la función entera como solo el bucle, quedando desplegado el resto de la función.

Una flechita hacia abajo/arriba indica que el código se encuentra desplegado, mientras que una flechita hacia la derecha y la presencia de un símbolo (flecha bidireccional dentro de un área azul oscuro) entre las llaves indica que el código se encuentra plegado.

RStudio

File Edit Code View Plots Session Build Debug Profile To

+ +R + Save Save Print Go to file/function Ad

RecursoSubsteting.R x Untitled1\* x PRÁCTICA 6.R x

← → ↺ Source on Save 🔍 ⚡ 📄

⚠ Breakpoints cannot be set until the file is saved.

```
1
2
3 Polinter<-function(B,L,n){
4   P=0
5   for(i in 1:n{
6     P=P+B[i]*L[i]-u
7   }
8 }
9
10
```

RStudio

File Edit Code View Plots Session Build Debug Profile To

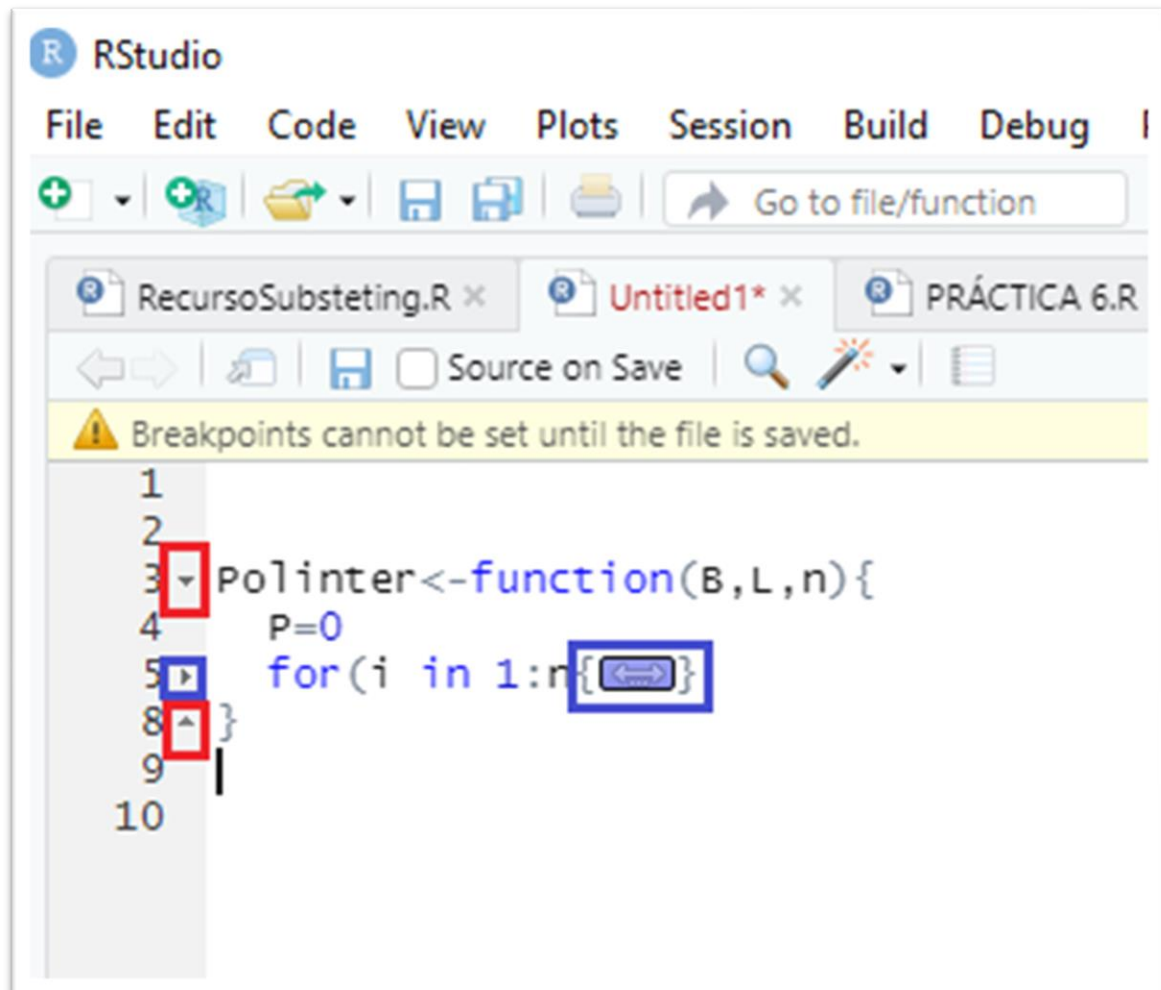
+ +R + Save Save Print Go to file/function Ad

RecursoSubsteting.R x Untitled1\* x PRÁCTICA 6.R x

← → ↺ Source on Save 🔍 ⚡ 📄

⚠ Breakpoints cannot be set until the file is saved.

```
1
2
3 Polinter<-function(B,L,n){
9
10
```



## 5.2. Plegamiento de código con la utilidad #Nombre####

Podemos plegar cualquier grupo de líneas de código usando la utilidad #Nombre####

Al escribir dichos # en ese orden, aparecerá una pestaña para plegar o desplegar código. Podremos además añadir un nombre para reconocer fácilmente el código plegado.



