

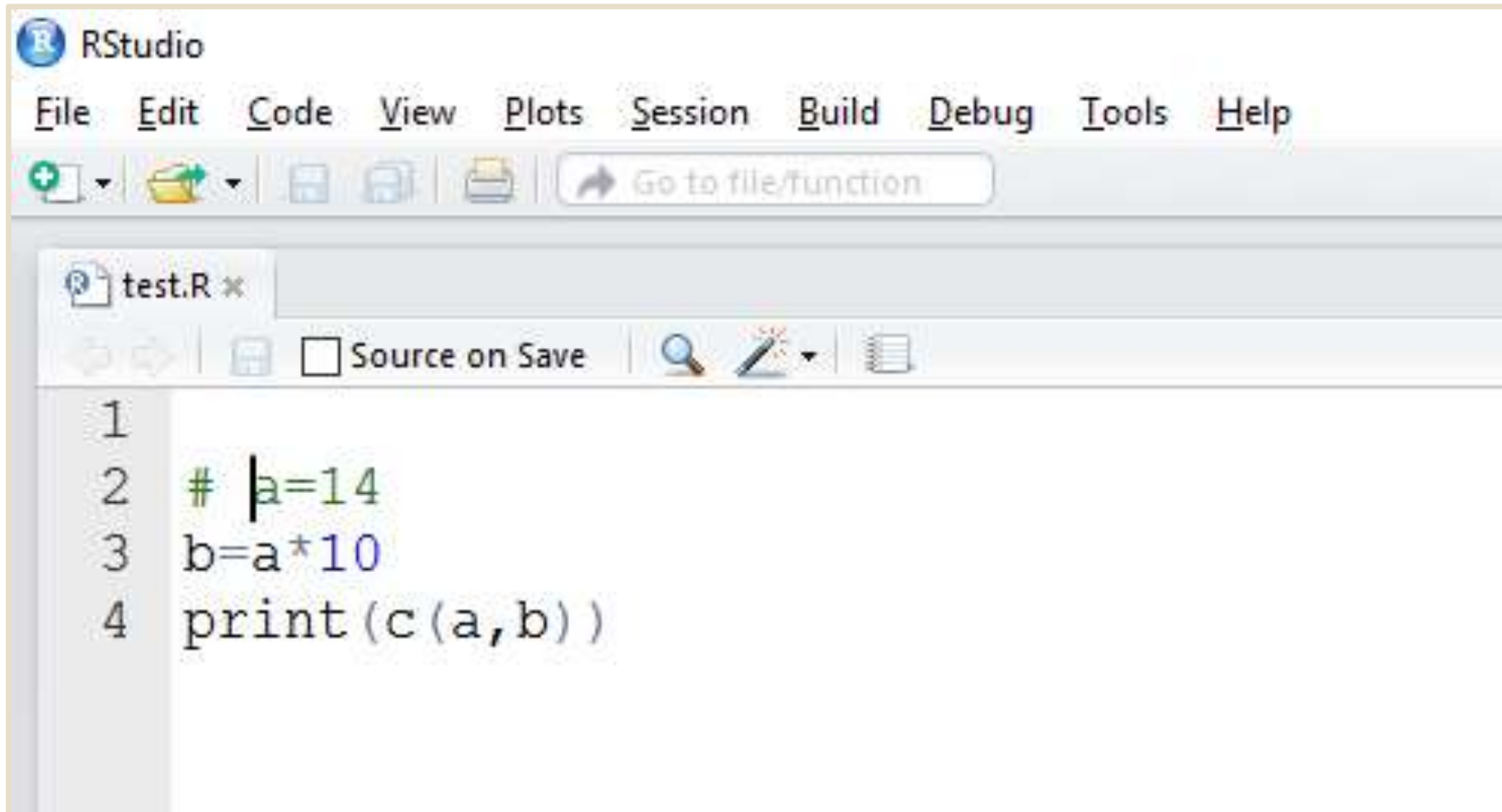
Introduction to R

In this lecture

- How to
 - add comments
 - clear the environment
 - saving the workspace

Add comments –single line

For single line comment, insert '#' at the start of the line

A screenshot of the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Tools, and Help. Below the menu is a toolbar with icons for file operations and a search bar labeled 'Go to file/function'. The main editor window shows a script named 'test.R' with four lines of code. Line 2 is a single-line comment starting with a green hash symbol followed by 'a=14'. Line 3 is 'b=a*10' and line 4 is 'print(c(a,b))'.

```
1  
2 # a=14  
3 b=a*10  
4 print(c(a,b))
```

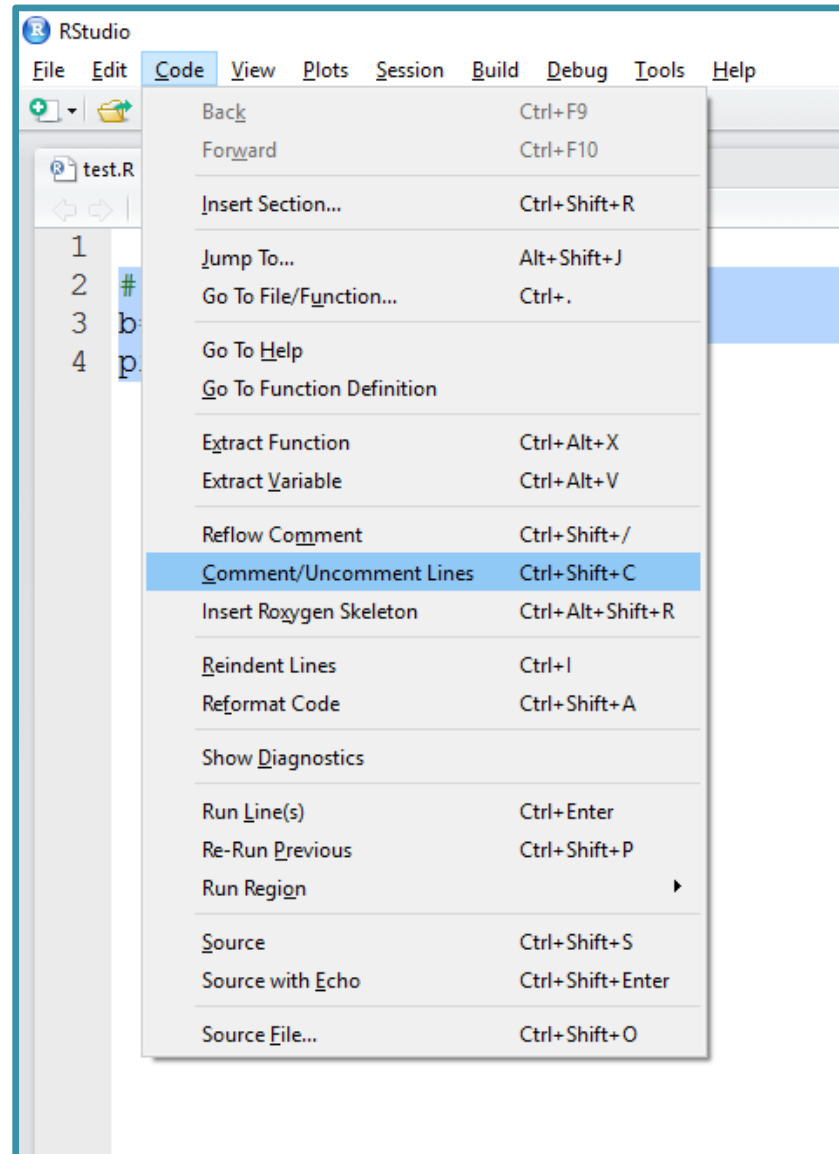
Add comments –Multiple lines

Two ways:

1) Select multiple lines using cursor, then press “Ctrl + Shift + C”

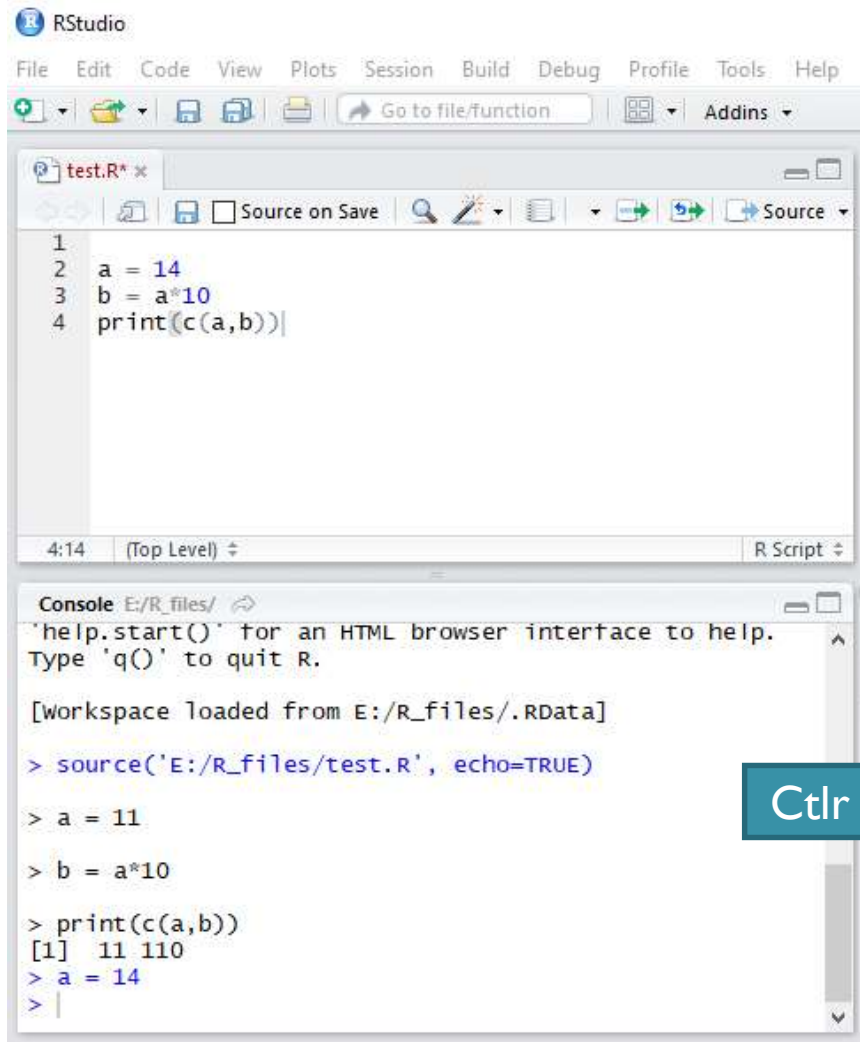
(OR)

2) Select multiple lines using cursor, click on “Code” in menu and select “Comment/Uncomment Lines”



Clear the console

“control +L”



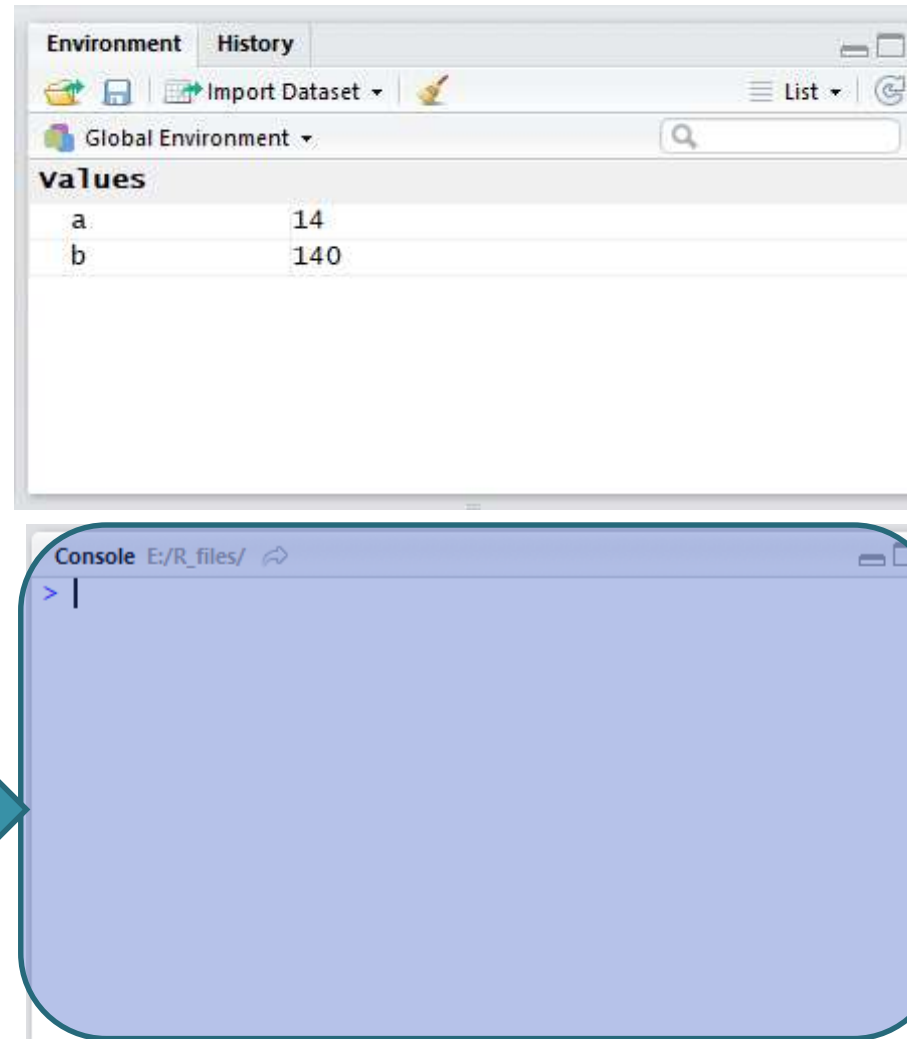
The RStudio interface is shown with a script editor and a console. The script editor contains the following code:

```
1  
2 a = 14  
3 b = a*10  
4 print(c(a,b))
```

The console shows the output of the script execution:

```
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
[workspace loaded from E:/R_files/.RData]  
> source('E:/R_files/test.R', echo=TRUE)  
> a = 11  
> b = a*10  
> print(c(a,b))  
[1] 11 110  
> a = 14  
>
```

Ctrl + L

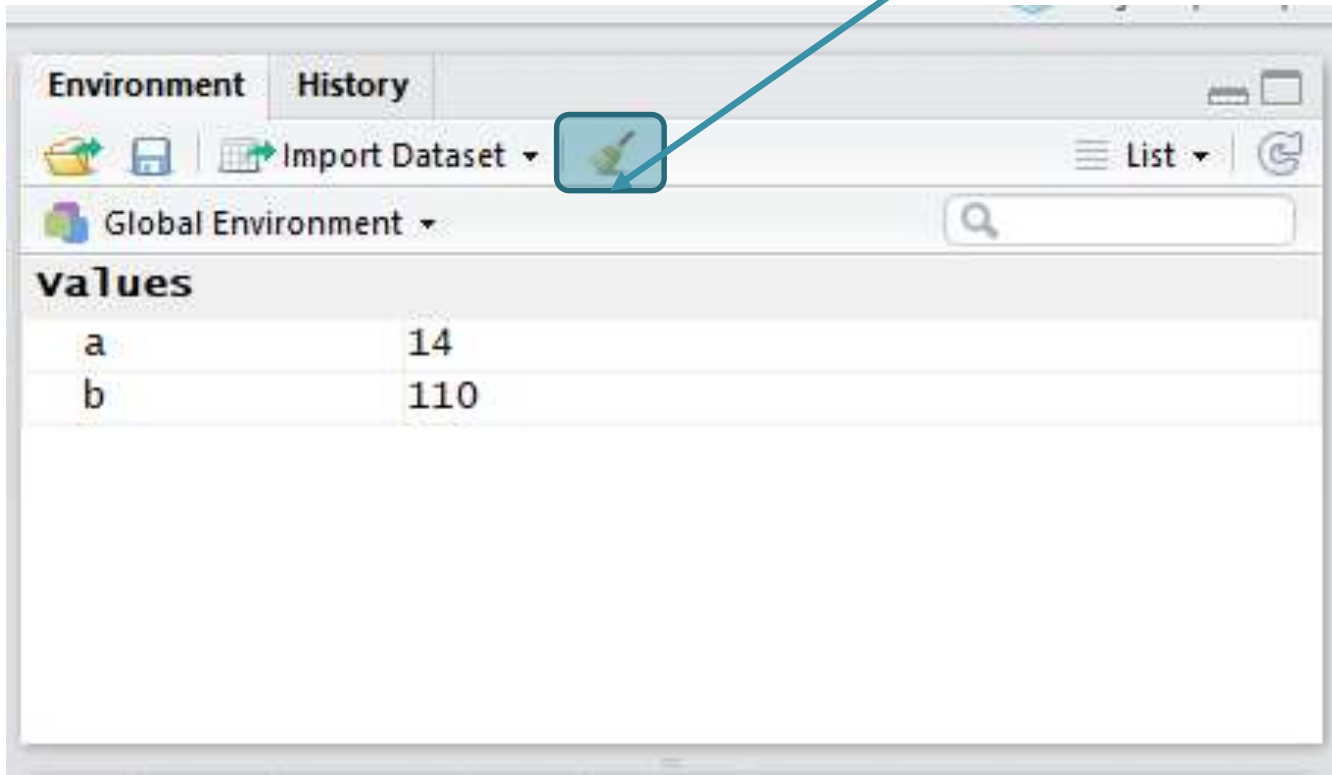


Clear the environment –rm()

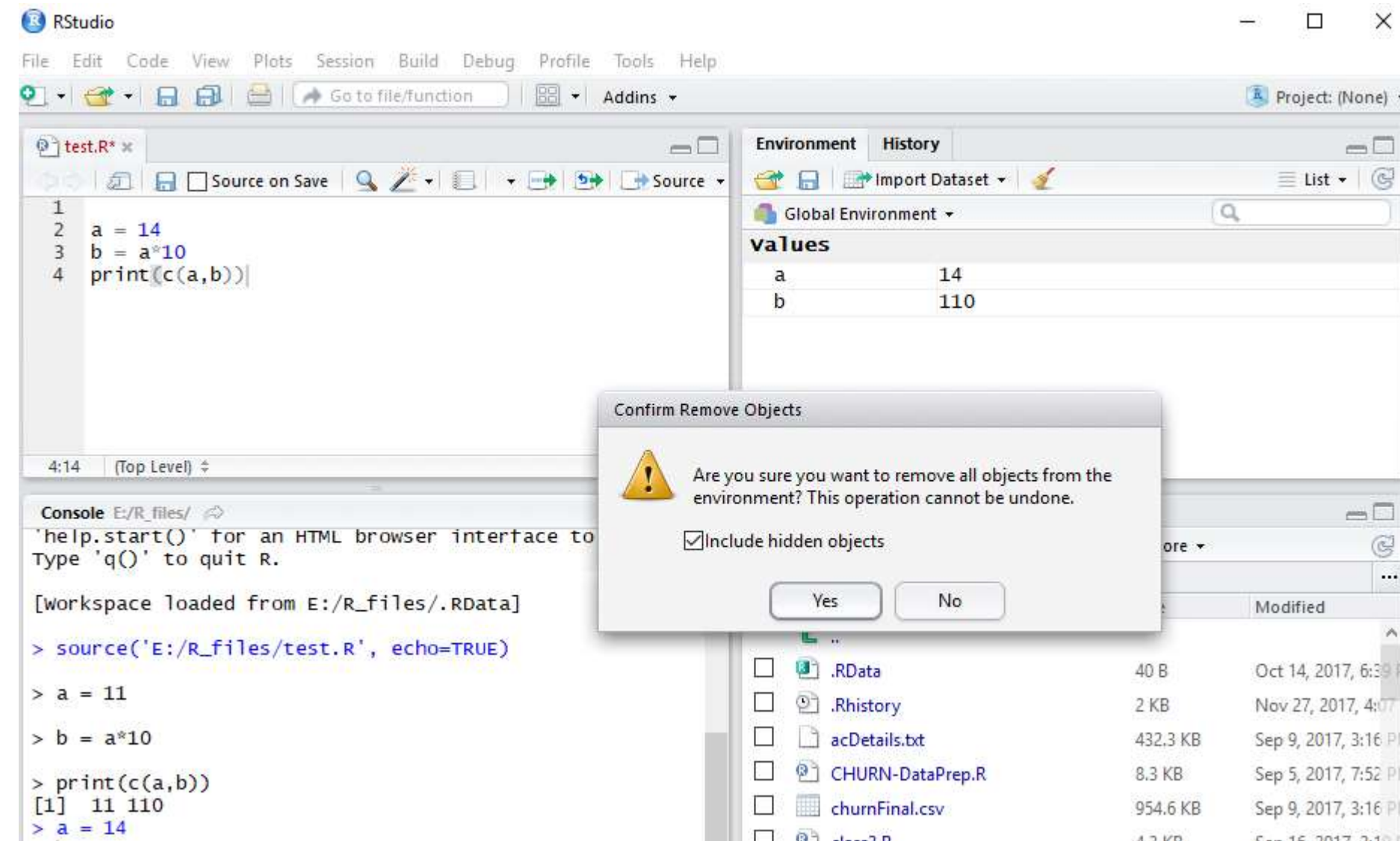
Single variable: Enter in console/R script : `rm(variable)`

All variables: Enter in console/R script : `rm(list=ls())`

OR



Confirmation dialog



Empty environment

The screenshot displays 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 labeled 'Go to file/function'. The 'Project: (None)' dropdown is visible in the top right.

The left pane shows the source editor with a file named 'test.R'. The code contains four lines:

```
1
2 a = 14
3 b = a*10
4 print(c(a,b))
```

The right pane is divided into 'Environment' and 'History' tabs. The 'Environment' tab is active, showing 'Global Environment' with a search bar. A blue box highlights the message 'Environment is empty'.

The bottom pane is the console, showing the output of the R script. It includes the help message, the workspace loading status, and the execution of the script with the following output:

```
> source('E:/R_files/test.R', echo=TRUE)
> a = 11
> b = a*10
> print(c(a,b))
[1] 11 110
> a = 14
>
```

The bottom right pane shows the file explorer for the directory 'E:/R_files'. It lists several files and folders:

Name	Size	Modified
..		
.RData	40 B	Oct 14, 2017, 6:39 P
.Rhistory	2 KB	Nov 27, 2017, 4:07
acDetails.txt	432.3 KB	Sep 9, 2017, 3:16 P
CHURN-DataPrep.R	8.3 KB	Sep 5, 2017, 7:52 P
churnFinal.csv	954.6 KB	Sep 9, 2017, 3:16 P
class3.R	4.3 KB	Sep 16, 2017, 2:19 P

Saving data from workspace

Workspace data

- Workspace information is temporary
- Is not retained after the session
 - If you close the R-session
 - If you restart the computer

Automatic option

The screenshot shows the RStudio interface with a script editor, environment pane, console, and package manager. A dialog box titled "Quit R Session" is open, asking "Save workspace image to E:/R_files/.RData?". The "Save" button is highlighted.

Script Editor (test.R):

```
1
2 a = 14
3 b = a*10
4 print(c(a,b))
5
```

Environment Pane:

values	
a	14
b	140

Console:

```
)
Error in runApp("MyAppDir", launch.browser = rstudioapi:
:viewer) :
  could not find function "runApp"
> library("httpuv")
warning message:
package 'httpuv' was built under R version 3.4.2
> library("shiny")
warning message:
package 'shiny' was built under R version 3.4.2
> runApp("MyAppDir")
```

Package Manager:

Name	Description
<input type="checkbox"/> arules	Mining Association Itemsets
<input type="checkbox"/> backports	Reimplementation: Introduced Since R
<input type="checkbox"/> base64enc	Tools for base64 en

Manual saving

- Can be permanently saved in a file – save command
- Can be reloaded for future sessions – load command

Example code

```
save(a, file="sess1.Rdata") # to save a single variable 'a'
```

```
# to save a full workspace with specified file name
```

```
save(list=ls(all.names=TRUE), file="sess1.Rdata")
```

```
save.image() # short cut function to save whole workspace
```

```
load(file="sess1.Rdata") # to load saved workspace
```