

## R Basics using RStudio



### Hands on Lab: R Basics using RStudio

Estimated time needed: 15 minutes

#### Learning Objectives

- Get familiar with RStudio
- Write your first R code snippet in RStudio

#### RStudio main UI

In this lab, you will be introduced to RStudio, the most popular and powerful IDE for developing R projects.

The main UI of RStudio is shown here:

The screenshot shows the RStudio interface with the following panels:

- File Editor:** Displays the following R code:
 

```
1 new.function <- function(a,b,c) {
2   result <- a * b + c
3   print(result)
4 }
5
6 a<-1
7 b<-2
8 c<-3
9
10 new.function(a, b, c)
```
- Console:** Displays the following R session history:
 

```
x <- 1
x <- 2
x <- 3
> new.function(a, b, c)
<-- # Simple Scatterplot
> attach(mtcars)
> plot(wt, mpg, main="Scatterplot Example",
+       xlab="Car Weight", ylab="Miles Per Gallon ", pch=19)
```
- Workspace:** Shows the following environment variables:
 

Values	Environment
a	1
b	2
c	3
sum	3
- Plots:** A scatterplot titled "Scatterplot Example" showing Miles Per Gallon versus Car Weight.

- In the **Console** panel, you can quickly try some R commands and see the results immediately.
- In the **File Editor** panel, you can write your R code or other text files with the help of syntax highlighting and auto completion.
- In the **Workspace** panel, you can review and manage the created objects.
- In the **File/Plots/Packages Explorer** panel, you can manage your files and other assets, such as plots or packages.

#### Write the first Hello World code snippet in the Console

Let us write your first Hello World in RStudio Console.

- Find the blinking cursor in the Console panel, type an incomplete `print` or `print` and pause a little bit for RStudio to show a pop-up suggestion list:

The screenshot shows the RStudio Console with the following code:

```
> print
```

A suggestion list is open, showing various `print` functions from different packages. The `base::print` function is highlighted, with its documentation displayed in a tooltip:

`print(x, ...)`  
 prints its argument and returns it *invisibly* (via `invisible(x)`). It is a generic function which means that new printing methods can be easily added for new classes.

The auto-complete feature of RStudio can help avoid the need for memorizing the code details and reducing keystrokes by just selecting from a suggestion list.

- Select the `print` function and add a character input `mtcars`, then press the Enter key:

`print("Hello World")`

You should see `Hello World` printed on the console.

The screenshot shows the RStudio interface with the following panels:

The screenshot shows the RStudio interface with the following panels:

- Console:** Displays the command `print("Hello World")`.
- Environment:** Shows the following environment variables:
 

Values	Environment
x	1

That's it, you have written your first Hello World code snippet in RStudio.

For practice, you can play with the console by typing anything you have learned so far, such as creating variables and doing basic math operations.

If you want to clear the console, you can press `ctrl` or `ctrl + l` key combination.

#### Review R objects in the Environment panel

Now let's try the Environment panel to review the R objects we created in the console.

- Type and run the following three lines of code in the console:

`x<-1`

`y<-2`

`z<-x+y`

You should see three variables `x`, `y`, `z` with assigned values in the Environment panel.

The screenshot shows the RStudio interface with the following panels:

- Console:** Displays the commands `x<-1`, `y<-2`, and `z<-x+y`.
- Environment:** Shows the following environment variables:
 

Values	Environment
x	1
y	2
z	3

- To clean the workspace, you can click the trash icon as shown below:

The screenshot shows the RStudio interface. In the top menu bar, the 'File' tab is selected. Below it, the 'Console' tab is active, displaying the R code: `> x<-1  
> y<-2  
> z<-x+y  
>`. To the right of the console is the 'Environment' pane, which lists the variables 'x', 'y', and 'z' with their corresponding values '1', '2', and '3'. There are tabs for 'History' and 'Connections' as well.

### Create your first R script file

By now, you have written some simple R code in the console interactively. Next, create an R script file with multiple lines of code and run them in batch mode.

- First, from the menu click `File > New File > R Script`.

The screenshot shows the RStudio interface with the 'File' menu open. The 'New File' submenu is displayed, and the 'R Script' option is highlighted with a red rectangle. Other options in the submenu include 'New Project...', 'Open File...', 'Text File', 'C++ File', 'R Sweave', 'R HTML', 'R Presentation', and 'R Documentation'.

- Then click `File > Save`, and name the file something like `first_script.R`. After the script file is saved, you can see an empty file called `first_script.R` created in your working directory.

The screenshot shows the RStudio interface. The 'Code Editor' panel on the left contains the R code: `x<-1  
y<-2  
z<-x+y  
print(z)`. The 'Environment' pane on the right shows the variables 'x', 'y', and 'z' with values '1', '2', and '3' respectively. The status bar at the bottom indicates 'Environment is empty'.

- Next, click `first_script.R` file to add the following code snippet:

```
x<-1
y<-2
z<-x+y
print(z)
```

You need to make sure the last line of the file is a new empty line.

So after copying the code snippet above, press the `Enter` key to start a new line in the script file.

Now, you can run the code in the script file, there are two running modes:

- The first mode is called `Run the current line or selection`. You can click and drag your mouse or use Shift + Up/Down keys to select all lines and then click the following `Run` icon to run them:

The screenshot shows the RStudio interface. The 'Code Editor' panel contains the R code: `x<-1  
y<-2  
z<-x+y  
print(z)`. The 'Run' icon (a green triangle) is highlighted with a red rectangle. The status bar at the bottom indicates 'Environment is empty'.

- The second mode is called `source` where it runs all lines of code in the file by clicking the following `Source` icon:

The screenshot shows the RStudio interface. The 'Code Editor' panel contains the R code: `x<-1  
y<-2  
z<-x+y  
print(z)`. The 'Source' icon (a blue square with a white 'S') is highlighted with a red rectangle. The status bar at the bottom indicates 'Environment is empty'.

You should see the results in the console:

```
source('/resources/rstudio/first_script.R')
```

```
[1] 3
```

The screenshot shows the RStudio interface. The 'Console' pane at the bottom displays the output of running the script: `source('/resources/rstudio/first_script.R')` followed by `[1] 3`. The 'Environment' pane on the right shows the variables 'x', 'y', and 'z' with values '1', '2', and '3' respectively.

That's it about creating and running the R script file!

**Exercise: Practice Tasks**

Task 1: Create a new R script

- Assign two variables where  $x = 10$  and  $y = 20$ .
- Save the file as **Subtract.R**.

Click here for solution

```
x <- 10  
y <- 20
```

The screenshot shows the RStudio interface with the code editor open. The code consists of two lines: `x <- 10` and `y <- 20`. The code editor has a light gray background with syntax highlighting for the R code.

```
x <- 10  
y <- 20
```

Task 2: Subtract x from y. Store in variable `result` and print the output.

Click here for solution

```
x <- 10  
y <- 20  
result <- y - x  
print(result)
```

The screenshot shows the RStudio interface with the code editor open. The code consists of four lines: `x <- 10`, `y <- 20`, `result <- y - x`, and `print(result)`. The code editor has a light gray background with syntax highlighting for the R code.

```
x <- 10  
y <- 20  
result <- y - x  
print(result)
```

**Summary**

In this lab, you have been introduced to RStudio. You have practiced how to write and run R code in both the console and in R script files. You used the Environment panel to review the R objects in your workspace.

Thank you for completing this lab!

**Author(s)**

[Prashant Verma](#)

IBM Corporation 2021. All rights reserved.