

Lesson 1

by Lorraine Gaudio

Lesson generated on September 05, 2025



BOISE STATE UNIVERSITY

Contents

1	📄 Welcome to Lesson 1	2
2	Running Code	3
3	Using R Scripts	4
4	Objects & Assignment	6
5	📄 Part I Assignment	8
6	Character Strings	9
7	📄 Part II Assignment	10
8	📄 Part III Assignment	11
9	Save and Upload	12
10	Today you practiced	13

1. ☒ Welcome to Lesson 1

1. Open your course project for RStudio (or create a new project if you don't have one yet).
2. Create a new R script file (File > New File > R Script).
3. Type in the code provided in this document and as you follow along with the video. Pause the video at anytime to answer assignment questions, dig deeper or add memo notes.

1.1 Lesson Overview

In the next few minutes you will learn how to:

- ☐ Run code from a script or the console.
- ☐ Do simple arithmetic in R.
- ☐ Create and inspect objects with the assignment operator (`<-`).
- ☐ Combine words with the `paste()` function.
- ☐ Save your workspace so nothing is lost.
- ☐ Follow R's object-naming rules.

2. Running Code

- Try running the next line in your console. Notice that R immediately prints the answer.

3 + 4

Type this line in your console and click Run or press Ctrl + Enter

3 + 4

[1] 7

- Nice! You just ran your first R command. The simplest usage of R is performing basic calculations. You can do addition (+), subtraction (-), multiplication (*), and division (/).

3. Using R Scripts

Documenting your work allows you and other to understand how you came to your results. In R Studio, you can open an existing R script by going to the File menu, selecting Open File..., and then navigating to the location of your script. Alternatively, you can use the keyboard shortcut Ctrl + O (or Cmd + O on a Mac) to bring up the file browser and choose your script.

To add comments in an R script, simply use the # symbol at the beginning of a line to indicate that the following text is a comment and should be ignored by the R interpreter.

```
# This script performs simple arithmetic operations.
```

```
# Addition
```

```
2 + 3 # Adds 2 and 3
```

```
## [1] 5
```

```
# Subtraction
```

```
10 - 4 # Subtracts 4 from 10
```

```
## [1] 6
```

```
# Multiplication
```

```
5 * 6 # Multiplies 5 and 6
```

```
## [1] 30
```

```
# Division
```

```
20 / 4 # Divides 20 by 4
```

```
## [1] 5
```

```
# Exponents
```

```
3^2 # Calculates 3 to the power of 2
```

```
## [1] 9
```

You can add a line break to a long comment by using the # symbol on each new line. This is necessary because R interprets everything on a single line after the first # as part of the comment. For example:

```
# This is a very long comment that explains what the following code does,  
# and we need to use a new line to keep it readable and within the editor's width.  
# It helps to explain complex functions or the purpose of an entire script.
```

□ Break Things! Try different ways of typing equations. Type all kinds of basic calculations.

```
# Create a memo note in your R script if you did □ Break Things!.  
# This is where you could earn points for demonstrating learning skills.  
## 1. Explain what you did, what your observations are, and what you expect  
# to happen.  
  
## 2. Demonstrate learning skill(s) in your memo  
  
## Example:  
  
# I ran the command 3 + 4 in the console. I expect to see the result 7 printed  
# in the console. I wondered what happens if there are no spaces between  
# the numbers and the plus sign. I tested it and it still works. I learned that  
# R ignores spaces in the code.
```

The value 7 isn't saved anywhere yet; it's just printed as a result in the console. Next we'll capture results as objects so we can reuse them.

4. Objects & Assignment

In R, *everything* is an object. Use “<-” to assign a value to a name. Type the following into your R script. Then run each line and watch the Environment pane for new entries.

```
result <- 3 + 4    # store the arithmetic output

number <- 42       # store a simple number

ratio <- number / result # reuse objects to calculate 42 / 7
```

Find the Environment pane (top-right) and see the new objects.

- ☐ What happened? Type the name of one of the objects in the Console and press Enter. What happened?
- ☐ Break Things! What are ways that creating a vector causes an Error? Try naming an object with a number at the start, like `3numbers <- seq(from = 1, to = 10, by = 2)`.
- ☐ Explore and Play: Are there different ways to assign an object? Try using `=` or `->` instead of `<-` and see what happens.

```
# Create a memo note in your R script if you did ☐ What happened?, ☐ Break Things!, and ☐ Explore and Play.
# This is where you could earn points for demonstrating learning skills.
## 1. Explain what you did, what your observations are, and what you expect
# to happen.

## 2. Demonstrate learning skill(s) in your memo

# Example:

# I created three objects: result, number, and ratio. I assigned the value of
# 3 + 4 to result, 42 to number, and calculated the ratio of number to result.
# When I typed the name of one object in the console, it printed the value of
# that object.

# I decided to click around in the Environment pane and found a down carrot
# next to list. I switched it to Grid and then back again. I then decided to
# learn more about this by typing this into a internet search engine. "Explain
# the drop down carrot next to List in R Studio. What is the difference between
# Grid and List?" I learned that "Grid" is used to visualize data in a
# structured format where "List" is a simple way of organizing objects.
```

Reminder: Now would be a great time to click save (or press Ctrl + S) to save your script.

- Note: Check your working directory.

```
getwd() # This shows your current working directory.
```

If you have the course project open, it should show the correct folder. Additionally, you can set it with `setwd()` if needed.

□ Look deeper: Does RStudio have an auto-save feature? What happens if you close RStudio without saving? How often should you save your work?

```
# Create a memo note in your R script if you did □ Look deeper.
```

```
# This is where you could earn points for demonstrating learning skills.
```

```
## 1. Explain what you did to discover saving in R, what your observations are, and what you expect  
# to happen.
```

```
## 2. Demonstrate learning skill(s) in your memo
```


5. ☒ Part I Assignment

Now it's your turn to practice. Follow the tasks below to complete part of the **technical skill practice assignment**.

1. Work through each task in order. Replace the ____ with your code.
2. Run each completed line to be sure no errors appear and objects show in the Environment.

5.1 Task 1

Store the quantity 62 in an object called items_sold

```
items_sold <- ____
```

5.2 Task 2

Store the unit price 100 in an object called unit_price

```
unit_price <- ____
```

5.3 Task 3

Calculate total_revenue by multiplying items_sold and unit_price

```
total_revenue <- ____ * ____ # edit only the blanks
```

6. Character Strings

Computers don't just crunch numbers—they handle text too! Strings live inside quotation marks. `paste()` joins them with spaces by default. Type the following into your R script and run each line:

```
greeting <- "Welcome to R."
statement <- "Productive code is clear and reproducible."
message <- paste(greeting, statement) # combine the two sentences
message # Run to print the full message
```

□ Create your own *statement* with something you'd like to tell future□ you, then run it the message lines again.

Example:

```
own_message <- "I will remember to save my work often."
```

```
# Create a memo note in your R script if you did □ Create your own statement.
# This is where you could earn points for demonstrating learning skills.
## 1. Explain what you did, what your observations are, and what you expect
# to happen.

## 2. Demonstrate learning skill(s) in your memo
```

7. ☒ Part II Assignment

1. Work through each task in order. Replace the ____ with your code.
2. Run each completed line to be sure no errors appear and objects show in the Environment.

7.1 Task 4: Character objects

Store "yellow" in an object first_word

```
first_word <- ____
```

7.2 Task 5

Store "banana" in an object second_word

```
second_word <- ____
```

7.3 Task 6

Paste first_word and second_word with a space; call the result sentence

```
sentence <- paste(____, ____)
```

7.4 Task 7: Inspect & save

View object types

```
class(total_revenue)
```

☐ What happened? What prints?

```
class(sentence)
```

☐ What happened? What prints?

8. ☒ Part III Assignment

Now it's your turn to practice. Follow the tasks below to complete part of the **technical skill practice assignment**.

1. Work through each task in order. Replace the ____ with your code.
2. Run each completed line to be sure no errors appear and objects show in the Environment.

8.1 Task 8

☐ Create-your-own (challenge)

It's play time! Create new object(s) that combines skills above.

Name it `my_object` and write a comment describing what it does.

```
my_object <- ____
```

```
# Create a memo note in your R script if you struggled with the technical skill assignment question ☐ Create-your-own (challenge)  
# You can earn points for demonstrating learning skills while completing your technical skill assignment!
```

```
## 1. Explain what you did, what your observations are, and what you did  
# to figure out the error.  
## 2. Demonstrate a learning skill(s) in your memo
```

```
# Example:
```

```
# I tried to create an object called my_object that combines the sentence and  
# total_revenue but I forgot that the quotation mark needed to be before the  
# comma, not after. I had an error and it took me a few minutes to figure it  
# out. I review the earlier script to compare and then I finally saw it.
```

9. Save and Upload

1. You will be submitting **both** the R script and the workspace file. The workspace file saves all the objects in your environment that you created in this lesson. You can save the workspace by running the following command in your R script:

```
save.image("Assignment1_Workspace.RData")
```

Or you can click the “Save Workspace” button in the Environment pane.

☐ **Always save the R documents before closing. If you don’t, you will lose your work.**

2. Find the assignment in this week’s module in Canvas and upload **both** the R script and the workspace file.

10. Today you practiced

- Running code
- Storing objects in the environment
- Creating character strings
- Saving your workspace and R script

□ Great progress! In the next lesson, we'll learn more about objects called vectors.