

Lesson 0: Introduction to R Studio

Lesson Plan

Learning Targets

Students will learn how to access and install R Studio on their computers.

Students will learn the 4 panels of R Studio.

Students will learn math computations in R Studio.

Students will learn how to input numeric and character vectors in R Studio.

Task 1: Getting R and R Studio

Here is a link for a video on R Studio basics.

If you want to download R and R Studio on your computer, here is a link:

<https://youtu.be/TFGYIKvQE04>

This link will help for downloading R to a computer:

<https://cran.r-project.org/bin/windows/base/>

Here is a link to download R Studio:

<https://www.rstudio.com/>.

Task 2: Playing with R Studio

R can do math. We are going to look at the operations for adding (+), subtracting (-), multiplying (*), dividing (/), and finding powers (^)

- a. In the R Console type the following math problems. Make sure to use the Operations R recognizes.

5+50

$\frac{16}{9}$

20.1×17.8

3^5

$(5 - 2)(5 - 2)^6$

```
> #Task 2
> #a
> 5+50
[1] 55
> 16/9
[1] 1.777778
> 20.1*17.8
[1] 357.78
> 3^5
[1] 243
> (5-2)*(5-2)^6
[1] 2187
```

- b. Create some of your own math problems and place them in the Console. Check what R outputs with your calculator.

```

> #b
> 5-50
[1] -45
> 16/9*3
[1] 5.333333
> 0.1*-7.8
[1] -0.78
> 3^10
[1] 59049
> (5-2)^6*(1-2)^6
[1] 729

```

c. Use the same math problems from Part a, type them into the text editor and then press run.

```

> #Task 2
> #a
> 5+50
[1] 55
> 16/9
[1] 1.777778
> 20.1*17.8
[1] 357.78
> 3^5
[1] 243
> (5-2)*(5-2)^6
[1] 2187

```

d. Which coding area do you prefer to run code, the text editor or the console? Why?
 The text editor saves your work, but the console does not.

Task 3: Putting Numeric Data into R Studio

The data R reads are called vectors. To put in the numbers 0, 1, 1, 1.5, 2, 7, 9 to R and save it, we use the following code:

```
falcon <- c(0, 1, 1, 1.5, 2, 7, 1, 0.5)
```

The object or vector containing the numbers is now called falcon. To assign a vector to a variable, falcon, you use the symbol <- . The c() portion of the code stands for concatenate which means to put the numbers together.

R can add, subtract, multiply, divide, and sort the data.

a. Enter the following code into R.

```

falcon <- c(0, 1, 1, 1.5, 2, 7, 1, 0.5)
falcon + 5
falcon +      5
falcon - 2.5
falcon / 2

```

```

> #a
> falcon + 5
[1] 5.0 6.0 6.0 6.5 7.0 12.0 6.0 5.5
> falcon + 5
[1] 5.0 6.0 6.0 6.5 7.0 12.0 6.0 5.5
> falcon - 2.5
Error: unexpected input in "falcon -"
> falcon / 2
[1] 0.00 0.50 0.50 0.75 1.00 3.50 0.50 0.25
> falcons ^ 2
Error: object 'falcons' not found
> sort(falcon)
[1] 0.0 0.5 1.0 1.0 1.0 1.5 2.0 7.0

```

```
falcons ^ 2
sort(falcon)
```

- b. What did you notice about what happened to the vector falcon when you completed each step.

```
falcon + 5
```

Added 5 to each element of the vector.

```
falcon +      5
```

Added 5 to each element of the vector.

```
falcon - 2.5
```

Subtracted 2.5 from each element of the vector.

```
falcon / 2
```

Divided each element of the vector by 2.

```
falcons ^ 2
```

Created an error because the vector is named falcon not falcons.

```
sort(falcon)
```

Ordered each element of the vector from least to greatest.

- c. Examine the structure of the data falcon using the code.

```
str(falcon)
```

Note, num [1:8] means a numeric vector with 8 elements.

Task 4: Putting Character Data into R Studio

R can also store character vectors. Enter the following vector into R.

```
thebest <- c("f", "a", "l", "c", "o", "n", "s", "a", "r", "e")
```

We will examine how R stores character vectors.

- a. Examine the structure of thebest using the following code.

```
str(thebest)
```

Note chr [1:10] means it is a character vector with 10 elements.

b. Suppose we want to know the 8th letter or element in the vector thebest.
Type the following code.
`thebest[8]`

c. Find the code to find the 4th letter in the vector.
d. `thebest[4]`

Task 4 Key

```
> str(thebest)
chr [1:10] "f" "a" "l" "c" "o" "n" "s" "a" "r" "e"
> thebest <- c('f','a','l','c','o','n','s','a','r','e')
> str(thebest)
chr [1:10] "f" "a" "l" "c" "o" "n" "s" "a" "r" "e"
> thebest[8]
[1] "a"
> thebest[4]
[1] "c"
```

Perform the following math operations in R. Record the R output.

1. $5+70.294$
2. $\frac{16}{10}$
3. -20.1×12.9
4. 23^3
5. $(5 - 12)(5 - 12)^2$

```
> #1
> 5+70.294
[1] 75.294
> #2
> 16/10
[1] 1.6
> #3
> -20.1*12.9
[1] -259.29
> #4
> 23^3
[1] 12167
> #5
> (5-12)*(5-12)^2
```

Enter the following numbers as a numeric vector. Save the vector as ekhs.

7, 4, 2.5, 7, -12, 35, 73

6. Multiply each element of ekhs by -2

```
> #6
> ekhs*-2
[1] -14 -8 -5 -14 24 -70 -146
> #7
> ekhs+5.9
[1] 12.9 9.9 8.4 12.9 -6.1 40.9 78.9
```

7. Add 5.9 to each element

```
> #8
> ekhs^3
[1] 343.000 64.000 15.625 343.000 -1728.000 42875.000 389017.000
> #9
> sort(ekhs)
[1] -12.0 2.5 4.0 7.0 7.0 35.0 73.0
```

8. Find the third power of each element

9. Sort the elements.

10. Use the function `summary()` on ekhs by typing `summary(ekhs)`. Record your results.

```
> summary(ekhs)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-12.00   3.25   7.00   16.64   21.00   73.00
~ |
```

Tell what is wrong, or debug the following codes. Try putting them into R and read the error codes.

11. falcon < c(0, 1, 1, 1.5, 2, 7, 1, 0.5)
The sign is wrong <- .
12. falcon <- c(0, 1, 1, 1.5, 2, 7, 1 0.5)
A comma is missing between the 1 and 0.5.
13. a <- ("f", "r", "l", "e", "n", "d")
Missing the c() in front of the vector.
14. a <- c("f", "r", "l", "e", "n", "d")

Check for Understanding

Put the numbers 1, 3, 4.6, -1, 67, -76, 14, 16.2, 1.79 into R as a vector names xyz. Write the code and output needed.

- a. Multiply each element of xyz by -3.

- b. Sort xyz.

```
> xyz <- c(1, 3, 4.6, -1, 67, -76, 14, 16.2, 1.79)
> #a. Multiply each element of xyz by -3.
> xyz*-3
[1] -3.00 -9.00 -13.80  3.00 -201.00 228.00 -42.00 -48.60 -5.37
> #b. Sort xyz.
> sort(xyz)
[1] -76.00 -1.00  1.00  1.79  3.00  4.60 14.00 16.20 67.00
> #c. Find the summary statistics of xyz.
> summary(xyz)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-76.000  1.000   3.000   3.399 14.000  67.000
> |
```

- c. Find the summary statistics of xyz.

