

Welcome to rstudio

Jeremy McWilliams

Welcome to rstudio!

Let's start off by clicking the "Visual" button above, so this document is a little easier to read and work with. You'll likely get a popup window the first time, confirming that you want to switch to Visual Mode.

Just click *Use Visual Mode*, with *Don't show this message again* checked.

Learning objectives

- Gain some familiarity with rstudio
- Learn how to run R code in "code chunks"
- Perform some arithmetic with R
- Create variables in R, and assign values to them

rstudio windows

You'll notice rstudio is divided into 4 windows. Let's take a look at each one:

- upper-left: this is the area for reading / writing files. We'll probably use this window the most this semester.
- lower-left: this is the console, where you can run R code directly. You might use this occasionally, but for the most part probably won't have to.
- upper-right: this window has a couple useful tabs:
 - Environment: any variables you create will be listed here. This can be useful for quick reference.
 - History: this shows the R commands you've run within a session.

- lower-right: this window lists any files and folders that are part of your project/assignment. It also has tabs that can display plots and help documentation.

You can resize windows / borders as needed in order to more easily focus on the particular window you're using.

Code chunks

The grey rectangles below are “code chunks”, where you can type and execute R code. To run the code, click the green triangle in the upper-right corner of the code chunk.

Let's try running the code chunk below, which is just some simple arithmetic:

```
3+7
```

```
[1] 10
```

Now it's your turn. In the code chunk below, enter a command to subtract 7 from 18, and run it:

```
# Enter code below (by the way, I'm a comment)
```

```
18-7
```

```
[1] 11
```

Variables

Programming languages, including R, rely heavily on the use of *variables*. A variable is something you define and assign a value to. In the code chunk below, we're creating a variable called `x`, and assigning the value of 7 to it:

```
# Notice we use a 'left pointing arrow' to assign the value
```

```
x<-7
```

We don't get any real feedback from running the code above, but it's often a good idea to check your work by printing a variable's value to the screen. To do this, you can run code that simply lists the variable:

```
# prints to the screen:
```

```
x
```

```
[1] 7
```

It's worth noting that you can run multiple lines of code within a code chunk:

```
# create a variable "penguinCount" and give it the value 3
```

```
penguinCount<-3
```

```
# print to screen
```

```
penguinCount
```

```
[1] 3
```

Now it's your turn. Work with your group / people next to you, and do the following:

- create a variable called `y`, assign three plus eight to it, and print to the screen.
- create a variable `z`, and assign `x+y` to it, and print to the screen

```
# create the variable "y" here, and print to screen
```

```
y<-3+8
```

```
y
```

```
[1] 11
```

```
# create the variable "z" here, and print to screen
```

```
z<-x+y
```

```
z
```

```
[1] 18
```

You made it! That's it for now. We'll do a quick review of variables during the next lesson, and will work our way toward some more exciting topics.

Make sure you have this completed by midnight on Tuesday, September 6th. And you don't have to "turn it in" - we'll be able to view your project on the class work space.