

# Worksheet 1: R Basics

Amanda Regan

2022-12-14

## R Basics

*This is the first in a series of worksheets for History 8510 at Clemson University. The goal of these worksheets is simple: practice, practice, practice. The worksheet introduces concepts and techniques and includes prompts for you to practice in this interactive document.*

### What is R?

To start let's define what exactly R is. R is a language and environment for statistical computing and graphics. R provides a variety of statistical and graphical techniques and its very extensible which makes it an ideal language for historians.

### Foundational Concepts

*Calculations:* There are a few concepts that are foundational to working in R. The first is that you can do math in R. In other words, R can easily be used as a calculator - a fancy calculator - but yes, a calculator.

You can do basic math:

```
5 * 5
```

```
## [1] 25
```

More complex math...

```
(500000 * 200 / 7500 + 2)
```

```
## [1] 13335.33
```

And the computer will barely break a sweat.

**Variables** are another important building block for any programming language. A variable stores a value. For example if I want x to equal 5 I can do that like this:

```
x <- 5
```

<- is known as an assignment operator. Technically, you could use =: here too but it is considered bad practice and can cause complicated issues when you write more advanced code. So its important to stick with <- whenever you are coding in R.

Variable names can be almost anything.

```
MyFavoriteNumber <- 25
```

Variable or object names must start with a letter, and can only contain letters, numbers, \_, and .. You want your object names to be descriptive, so you'll need a convention for multiple words. People do it different ways. I tend to use periods but there are several options:

```
i_use_snake_case
otherPeopleUseCamelCase
some.people.use.periods
And_aFew.People.RENOUNCEconvention
```

Whatever you prefer, be consistent and your future self will thank you when your code gets more complex.

- (1) You try, create a variable and assign it a number:
- (2) Can you assign a number with a decimal point?
- (3) Once we've assigned a variable we can use that variable to run calculations just like we did with raw numbers.

```
x <- 25
x * 5
```

```
## [1] 125
```

R can also handle more complex equations.

```
(x + x)/10
```

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

```
(x + x * x) - 100
```

```
## [1] 550
```

Two things to notice here. First, when you run the above code, the variable `x` shows up in your environment tab in R studio. This is a convenient place to see all the variables you've created in this environment. Second, note that the multiplication symbol is `*` not an `x`. Other mathematical functions follow similar practices `/` for division, `-` for subtraction, and `+` for addition.

- (4) You try. Assign a number to `x` and a number to `y`. Add those two numbers together.
- (5) Can you take `x` and `y` and multiply the result by 5?
- (6) Try creating two variables with names other than `x` and `y`. Descriptive names tend to be more useful. Can you multiply the contents of your variables?

R also has *built in functions*. We'll talk more about some of the others in a bit, but for now let's look at a few that relate to math. For example, `sqrt()` which does what you think it does, finds the square root of a number.

```
sqrt(1000)
```

```
## [1] 31.62278
```

Many functions have options that can be added to them. For example, the `round()` function allows you to include an option specifying how many digits to round to.

You can run it without that option and it'll use the default:

```
round(15.492827349)
```

```
## [1] 15
```

Or we can tell it to round it to 2 decimal places.

```
round(15.492827349, digits = 2)
```

```
## [1] 15.49
```

How would you know what options are available for each function in R? Every function and package in R comes with documentation that is built into R studio and can be pulled by by typing a question mark in front of the function in your console.

We can do it here too:

```
?round()
```

(7) Now you try, find the documentation for the function `signif()`.

In real life, you typically you wouldn't want to store this code in your script file. You probably don't need to pull up the documentation for the function everytime you run that piece of code. But for the purposes of this worksheet we're adding it to our `.Rmd` document.

(8) Try to use the console to find the documentation for `floor()`? Try it and then tell me, what does that function do? <