Objects in R

{% include toc title="In This Lesson" icon="file-text" %}

Learning Objectives

At the end of this activity, you will be able to:

- Be able to create, modify and use objects or variables in `R`.
- Be able to define the key differences between the str (string) and num (number) classes in `R` in terms of how R can or can not perform calculations with each.

What you need

You need R and RStudio to complete this tutorial. Also we recommend have you have an `earth-analytics` directory setup on your computer with a `/data` directory with it.

- How to Setup R / R Studio
- Setup your working directory

Creating objects

You can get output from `R` by typing a mathematical equation into the console - For example, if you type in `3 + 5`, `R` will calculate the output value.

```
# add 3 + 5
3 + 5
## [1] 8
# divide 12 by 7
12/7
## [1] 1.714286
```

However, is it more useful to assign *values* to *objects*. To create an object, we need to give it a name followed by the assignment operator `<-`, and the value we want to give it:

```
# assign weight_kg to the value of 55
weight_kg <- 55

# view object value
weight_kg
## [1] 55
```

Use Useful Object Names

Objects can be given any name such as `x`, `current_temperature`, or `subject_id`. However, it is best to use clear and descriptive words when naming objects to ensure your code is easy to follow.

We will discuss best practicing for coding in this module - in the clean coding lesson.

1. **Keep object names short:** this makes them easier to read when scanning through code.
2. **Use meaningful names:** For example: `precip` is a more useful name that tells us something about the object compared to `x`

3. **Don't start names with numbers!** Objects that start with a number are NOT VALID in R.
4. **Avoid names that are existing functions in R:** e.g., `if`, `else`, `for`, see here

A few other notes about object names in `R`:

- `R` is case sensitive (e.g., `weight_kg` is different from `Weight_kg`).
- Avoid other function names (e.g., `c`, `T`, `mean`, `data`, `df`, `weights`).
- Use nouns for variable names, and verbs for function names.
- Avoid using dots in object names - e.g. `my.dataset` - dots have a special meaning in R (for methods) and other programming languages. Instead use underscores `my_dataset`.

View object value

When assigning a value to an object, `R` does not print anything. You can force it to print the value by using parentheses or by typing the name:

```
weight_kg <- 55    # doesn't print anything
(weight_kg <- 55)  # but putting parenthesis around the call prints the value of `weight_kg`
## [1] 55
weight_kg          # and so does typing the name of the object
## [1] 55
```

Now that `R` has `weight_kg` in memory, we can do arithmetic with it. For instance, we may want to convert this weight in pounds (weight in pounds is 2.2 times the weight in kg):

```
2.2 * weight_kg
## [1] 121
```

We can also change a variable's value by assigning it a new one:

```
weight_kg <- 57.5
2.2 * weight_kg
## [1] 126.5
```

This means that assigning a value to one variable does not change the values of other variables. For example, let's store the animal's weight in pounds in a new variable, `weight_lb`:

```
weight_lb <- 2.2 * weight_kg
```

and then change `weight_kg` to 100.

```
weight_kg <- 100
```

What do you think is the current content of the object `weight_lb`? 126.5 or 200?

Optional challenge activity

What are the values of each object defined in EACH LINE OF code below?

```
mass <- 47.5           # mass?
age  <- 122            # age?
mass <- mass * 2.0     # mass?
age  <- age - 20       # age?
mass_index <- mass/age # mass_index?
```