Working with function arguments

Learning Objectives

After completing this tutorial, you will be able to:

- Find and download data from the USGS Earth Explorer Website
- Filter data by cloud cover to find datasets with the least amount of clouds for a study area.

What you need

You will need a computer with internet access to complete this lesson and the data for week 6 / 7 of the course.

{% include/data_subsets/course_earth_analytics/_data-week6-7.md %}

In the previous lessons, we have used many different functions and function arguments to customize our code.

For example, we used numerous arguments to plot our data including:

1. `main=""` to add a title
2. `axes=F` to remove the axes of our plot
3. `box=F` to remove the box surrounding the plot.

In the example below, we call each argument by name and then assign it a value based on the type of argument it is. For example the value for the `main=` argument is a text string which is the title that we want R to add to our plot.

```
# import and plot landsat
landsat_ndvi <- raster("data/week6/outputs/landsat_ndvi.tif")
plot(landsat_ndvi,
     main="landsat ndvi title - this title is rendered using a function argument",
     axes=F,
     box=F)
```

Matching Arguments

To be precise, R has three ways that arguments are supplied by you are matched to the *formal arguments* of the function definition:

1. by complete name
2. by position
3. by partial name (matching on initial n characters of the argument name) - we are not going to review this in class!

Arguments are matched in the manner outlined above in *that order*: by complete name, then by partial matching of names, and finally by position.

With that in mind, let's look at the help for `read.csv()`:

```
# view help for the csv function
?read.csv
```

There's a lot of information available in the help. The the most important part is the first couple of lines:

```
read.csv(file, header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = "", ...)
```

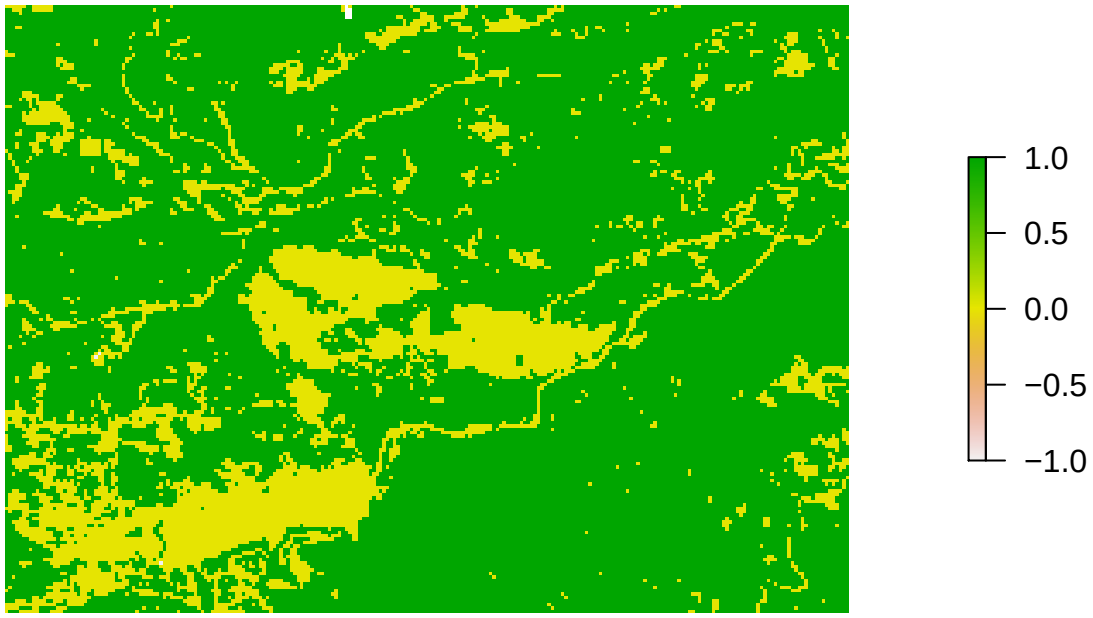

Figure 1: ndvi plot

This tells us that `read.csv()` has one argument, `file`, that doesn't have a default value, and six other arguments that do have a default value.

Now we understand why the following code returns an error:

```
dat <- read.csv(FALSE, "data/week2/precipitation/precip-boulder-aug-oct-2013.csv")
## Error in read.table(file = file, header = header, sep = sep, quote = quote, : 'file' must be a chara
```

The code above fails because `FALSE` is assigned to `file` and the filename is assigned to the argument `header`.

Default function arguments

We have passed arguments to functions in two ways:

1. directly: `plot(landsat_ndvi)`,
2. and by name: `read.csv(file = "data/inflammation-01.csv", header = FALSE)`.

We can pass the arguments to `read.csv` without naming them if they are in the order that R expects.

```
precip_data <- read.csv("data/week2/precipitation/precip-boulder-aug-oct-2013.csv",
                FALSE)
```

However, the position of the arguments matter if they are not named. Does the code below return an error?

```
# import csv
precip_data <- read.csv(header = FALSE,
                file = "data/week2/precipitation/precip-boulder-aug-oct-2013.csv")
```

What about this code?

```
dat <- read.csv(FALSE,
                "data/week2/precipitation/precip-boulder-aug-oct-2013.csv")
## Error in read.table(file = file, header = header, sep = sep, quote = quote, : 'file' must be a chara
```