Settlers of Antarctica - Module 1 - Days 2-3

Keaton Wilson

11/22/2019

Day 2 - Introduction to RStudio

Learning Objects

- Students will be able to succesfully install and open RStudio.
- Students will be able to differentiate the function of RStudio versus R, and generally describe the advantages of using an IDE.
- Students will be able to describe the utility of each panel in RStudio.
- Students will understand what a working directory is, and perform basic file system navigation using the files panel and autocompletion within RStudio.
- Students will be able to write a small script with appropriate documentation and commenting, run the script in the console, and save it to a specific place on their computers.

Check-in (5 minutes)

For those using their own computers? Successful installation? If there are a few that had problems or didn't get to it, do it now… ask folks sitting next to you for help. For those on classroom machines - open up RStudio and poke around a bit.

We will also set up the new groups now! Rearrange!

What is an IDE (interactive development environment)? (10 minutes)

Get on the web - look up the definition of an IDE, and focus on RStudio. What are some advantages of using an IDE. Why do you think we're using an IDE over base R? Do this in your group. Come up with an answer to each question above and be ready to report out.

RStudio exploration (10 minutes)

Open up RStudio. Spend some time familiarizing yourself with the struture of what is in front of you. On your own (not in groups), come up with a question or something you don't really understand about the piece of software you're working with. Be ready to report out.

Rstudio Mini-tour (10 minutes)

Perform a live-coding mini-tour of RStudio. Should focus on:
1. What the different panels do.
2. Navigating file structures in the files pane.
3. What is a working directory?
4. Console versus a script?
5. Appropriate commenting and script creation
6. Getting help!

```r
# My first example script
# Keaton Wilson
# keatonwilson@me.com
# 2019-11-22

# Examining a pre-built data set in R
Orange
```

```
##    Tree  age circumference
## 1     1  118            30
## 2     1  484            58
## 3     1  664            87
## 4     1 1004           115
## 5     1 1231           120
## 6     1 1372           142
## 7     1 1582           145
## 8     2  118            33
## 9     2  484            69
## 10    2  664           111
## 11    2 1004           156
## 12    2 1231           172
## 13    2 1372           203
## 14    2 1582           203
## 15    3  118            30
## 16    3  484            51
## 17    3  664            75
## 18    3 1004           108
## 19    3 1231           115
## 20    3 1372           139
## 21    3 1582           140
## 22    4  118            32
## 23    4  484            62
## 24    4  664           112
## 25    4 1004           167
## 26    4 1231           179
## 27    4 1372           209
## 28    4 1582           214
## 29    5  118            30
## 30    5  484            49
## 31    5  664            81
## 32    5 1004           125
## 33    5 1231           142
## 34    5 1372           174
## 35    5 1582           177
```

```r
# printing maximum circumference of the biggest tree
max(Orange$circumference)
```

```
## [1] 214
```

Group discussion (5 minutes)

What is happening in the last bit of code that I just wrote?
What do you think `max()` is? What do the parentheses do? What is the $ doing after Orange?

Group challenge (Until the end of class)

Construct a small script (with all of the appropriate formatting we discussed) that calculates the average (mean) circumference of trees in the orange data set. Assign this to a variable called mean_circ and print it to the console.

Save the script and be ready to share it at the beginning of next class.

Hints and tips: search the web for 'assignment operators in R' and 'creating variables in R' for help on how to make new variables.

Day 3 - Introduction to Programming

Learning Objects

1. Students will be able to perform basic math functions in the R console.

2. Students will be able to write scripts that assign values to variables and use these variables to perform various operations.

3. Students will be able to use helpfiles to learn how to use functions.

4. Students will be able to recall and explain how functions operate, and the basic syntax around functions (arguments, autocompletion, parentheses).

5. Students will be able to differentiate different data classes in R.

6. Students will learn how to create their own data structures (vectors) and 2d data (dataframes).

Revisiting the exercise at the end of last class.

Let's revisit your assignment from the end of last class. Let's have someone from each group (1 at a time), come up and wrote what they came up with.

Assignment: Construct a small script (with all of the appropriate formatting we discussed) that calculates the average (mean) circumference of trees in the orange data set. Assign this to a variable called mean_circ and print it to the console.

What were some problems? What difficulties did folks face?

Using R as a calculator - getting comfortable with the console

You can do basic math in the console (the bottom left part of the screen). For example:

```
3+5
```

```
## [1] 8
```

```
45^7
```

```
## [1] 373669453125
```

```
890*4.76523
```

```
## [1] 4241.055
```

Math - Group Challenge

Run the following equation in the console. You'll need to be creative, dig into the helpfiles and probably the web to figure out a few of the more complicated operations. Be ready to share your code.

The log (base 10) of the square root of pi, raised to the power of 12.

```
log10(sqrt(pi))^12
```

```
## [1] 5.565293e-08
```

Variables, assignments, vectors

Let's begin by opening a new script - we've been working mainly in the console, but it's important to get familiar with both the console and a script, and how to move back and forth between the two.

Demonstrate how to open a new script - highlight headings again. Remind them that this is best-coding practices. Do it every time!

Assignments are really key to almost everything we do in R. We can save stuff to ram, manipulate it, save it to our harddrive, etc.... this is how we create permanence in R. Anything can be saved to an object, and we do this with an assignment operator.

```
# R assignment things

# Let's say we want to store some stuff.
# These two are equivalent
x = 4
x <- 4

# Talk about differences.

x = 4
y = 10
z = x + y
z
```

```
## [1] 14
```

```
# This is simple and you'll rarely do it in real-world scenarios. We can also assign more complex lists

x = c(1,2,3,4,5,6)
x
```

```
## [1] 1 2 3 4 5 6
```

```
# This is a vector. R is inherently a vectorized language... here is what i mean.
x = c(1,2,3,4,5,6)
y = c(1,2,3,4,5,6)

#What is x+y going to look like?


# R does everything in vectors
```

2-dimensional data, functions, helpfiles

Most of the data you will encounter is two-dimensional, i.e. it has columns and rows (should be familiar for those of you who have worked with Excel before). R is really great at working with these types of data.

```
new_df = data.frame(sex = c("M", "F", "M", "F", "F", "M"),
                    height = c(182, 170, 191, 151, 150, 161))

new_df
```

```
##   sex height
## 1   M    182
## 2   F    170
## 3   M    191
## 4   F    151
## 5   F    150
## 6   M    161
```

Discussion point. This is a really simple data set, but let's come up with some questions. Let's write them on the board.

Probably going to lead to the mean height overall, and for men and women.

```
# Specifying certain columns
new_df$height
```

```
## [1] 182 170 191 151 150 161
```

```
mean(new_df$height)
```

```
## [1] 167.5
```

```r
# but what about men and women separately?

new_df[,2]
```

```
## [1] 182 170 191 151 150 161
```

```r
new_df[new_df$sex == "F",]
```

```
##   sex height
## 2   F    170
## 4   F    151
## 5   F    150
```

```r
mean(new_df[new_df$sex == "F",2])
```

```
## [1] 157
```

```r
mean(new_df[new_df$sex == "M", 2])
```

```
## [1] 178
```

Group Challenge - Using helpfiles on functions, using functions

As a group, find the standard deviation `sd()` of the height of both males and females and determine which is larger. Additionally, come up with a definition of standard deviation, use the helpfile to find out how the function works, and be prepared to show the code you used.

Data classes

There are a few main types in R, and they behave differently:
1. Numerical
2. Character
3. Integer
4. Logical
5. Complex
6. Factors (kind of)

The big takeaways:

1. You can use `class()` to determine the class (or some easier ways for big dataframes that we'll cover later in the course)

2. Columns that end up being character vectors can trip you up a lot

3. Vectors must be of all the same data type

```
# Comparison
1 + 1 + 1 + 1
"1" + "1" + "1" + "1"

vec = (1, 1.000, "1", TRUE)
```

Quick challenge - what data type is the is the sex column of the data frame we just used? Discussion of why it isn't character?