

Practical R: Lecture 1

Introductions

- Gene Buehler (NIH/NCATS)
 - PhD in Computer Science
 - R user for past ten years
 - Use R for computational biology

Class Structure

- Weekly Lectures
- Suggested activities and online resources
- Semester Project

Semester Project

- Use R to import, manipulate, analyze and visualize a dataset
- Ideally it will be your data, or your lab's data, or at least data relevant to your field
- Short presentations will be given on the last day of class for each project
- Grade will be based on your project presentation (although if you can't show up we can grade based on your presentation slides).

What this class isn't

- A programming class
- A bioinformatics class
- A workshop

Very Brief Course Outline

- Basics of using R (variables, functions, arguments)
- Getting Help (R help system, Google, command history)
- Importing data into R (from files, from URLs, and preloaded data)
- Packages: finding them, loading them, reading their documentation
- Visualizing Data: Graphs, graphs and more graphs
- Statistical Analysis: How and why
- Manipulating data: selecting, filtering, merging and more.
- Science specific analysis and visualization (eg. Heatmaps and dendograms for gene expression data)

There's more than one way

- Lots of ways to run R: command line, GUI, RStudio.
- Frequently more than one package to do what you want
- More than one statistical test you can use on a given type of data

Homework: Installing R and RStudio

- <https://cran.rstudio.com>
- <https://www.rstudio.com/products/rstudio/download/>
- Both are available for Windows, MacOSX, and Linux.

R Start-up Message

R version 3.2.0 (2015-04-16) -- "Full of Ingredients"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

>

Unpacking the Startup

Version Number

Often silly version name

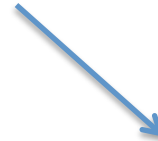
```
R version 3.2.0 (2015-04-16) -- "Full of Ingredients"  
Copyright (C) 2015 The R Foundation for Statistical Computing  
Platform: x86_64-apple-darwin13.4.0 (64-bit)
```

Operating system this version is made for

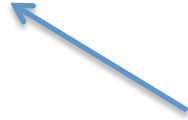
32 or 62 bit

Unpacking the Startup

“Don’t sue us”



R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.



Terms under which you can distribute the software

Unpacking the Startup

R has support for different local conventions. For example, you can get it to use “,” instead of “.” for decimal notation



Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.



R is made by real people who deserve credit (citations) for their work as much as you do. Please give it to them.

Unpacking the Startup

Some R packages have “demos” or their capabilities. For example, `demo(graphics)` will show you some of the cool graphs R can make.

R’s help function. You will probably use this a lot.

Type `'demo()'` for some demos, `'help()'` for on-line help, or `'help.start()'` for an HTML browser interface to help. Type `'q()'` to quit R.

How you quit R from the console.

[Workspace loaded from ~/.RData]

A “workspace” is where the stuff you do in R gets stored when you quit.

Start of RMarkdown Presentation

Expressions

Typing “1+1” on the console will cause R to evaluate the expression and tell us the results

```
> 1+1
```

```
[1] 2
```

The “[1]” before the answer is R’s line numbering scheme for reading long sets of numbers.

For example, if we were to type “letters” which is R’s built in variable with the letters of the alphabet:

```
> letters
```

```
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l"  
[13] "m" "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x"  
[25] "y" "z"
```

Standard Math Stuff

Addition and subtraction symbols are what you would expect (+ and -). Multiplication and division are * and / respectively.

```
> 5-3
```

```
[1] 2
```

```
> 5*3
```

```
[1] 15
```


Order of Operations

Order of operations is as you might remember from math class. Multiplication and division before addition and subtraction. You can force the order of operations to be what you want by enclosing things you want done first in parentheses.

```
> 5 * 3 + 2
```

```
[1] 17
```

```
> 5 * (3 + 2)
```

```
[1] 25
```

Functions

Functions are how things get done in R. They take “arguments” as their input and output something useful (hopefully). R has a lot of built in functions. You can download and install packages to get access to more functions. You can easily write your own functions, but that won't be the focus of this course.

```
> sqrt(25)
```

```
[1] 5
```

Arguments, optional and required

```
log()
```

Error: argument "x" is missing, with no default

```
> log(100)
```

```
[1] 4.60517
```

```
> log(100, base=10)
```

```
[1] 2
```

Variable assignment

```
> x <- 2  
> x * 5
```

```
[1] 10
```

Variable Naming

If you name your variables things like “x” and “y”, chances are excellent you will soon forget what they mean. R variable names can be long, so make them as long as you need to explain what they are.

```
> plate_map.csv <- read.csv("plate_map.csv")  
> plate_map.csv.colnames <- colnames(plate_map.csv)
```

Basic “Atomic” Data types

For the purposes of this class, R has three “types” of data (there are a few more we won’t concern ourselves with, like for complex numbers).

```
> class(12)
```

```
[1] "numeric"
```

```
> class("Susan")
```

```
[1] "character"
```

```
> class(TRUE)
```

```
[1] "logical"
```