Module 2

Variables

Andrew Jaffe Instructor

Getting Started

- · You should have the latest version of R installed (R 3.1 as of 6/30/2014)!
- · Open R Studio
- Files --> New --> R Script
- · Save the blank R script as "day1.R" in a directory of your choosing
- · Add a comment header

Commenting in Scripts

Add a comment header to day1.R: '#' is the comment symbol

Explaining output on slides

In slides, a command (we'll also call them code or a code chunk) will look like this

> print("I'm code")

[1] "I'm code"

And then directly after it, will be the output of the code.

So print("I'm code") is the code chunk and [1] "I'm code" is the output.



Note, when you type your command, R inherently thinks you want to print the result.

- · The R console is a full calculator
- · Try to play around with it:
 - +, -, /, * are add, subtract, multiply, and divide
 - ^ or ** is power
 - parentheses -- (and) -- work with order of operations



Try evaluating the following:

- · 2+2*3/4-3
- · 2*3/4*2
- · 2⁴-1

- \cdot You can create variables from within the R environment and from files on your computer
- $\cdot\,\,$ R uses "=" or "<-" to assign values to a variable name
- · Variable names are case-sensitive, i.e. X and x are different



- · We will start with 1 dimensional classes first; these are often referred to as 'vectors'
- · Vectors can have multiple observations, but each observation has to be the same class.



Try assigning your full name to an R variable called name

Try assigning your full name to an R variable called ${\tt name}$

```
> name = "Andrew Jaffe"
> name
```

[1] "Andrew Jaffe"

The 'combine' function

The function c() collects/combines/joins single R objects into a vector of R objects. It is mostly used for creating vectors of numbers, character strings, and other data types.



The 'combine' function

Try assigning your first and last name as 2 separate character strings into a length-2 vector called name2

The 'combine' function

Try assigning your first and last name as 2 separate character strings into a length-2 vector called name2

```
> name2 = c("Andrew", "Jaffe")
> name2

[1] "Andrew" "Jaffe"
```

An Aside on functions

The c() command is called a function: it takes inputs and gives an output. In R, functions always go function(input), or name of function, then parentheses. The input can many different things, such as function(x, y, z). We will cover functions in more detail later, and show you some simple functions in this session such as length.

length(): Get or set the length of vectors (including lists) and factors, and of any other R
object for which a method has been defined.



What do you expect for the length of the name variable? What about the name2 variable?

What are the lengths of each?

What do you expect for the length of the name variable? What about the name2 variable?

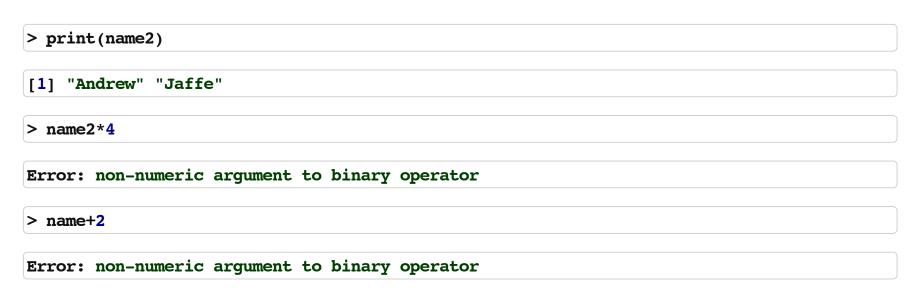
What are the lengths of each?

> length(name)		
[1] 1		
> length(name2)		
rengen (namez)		
[1] 2		

You can perform functions to entire vectors of numbers very easily.



But things like algebra can only be performed on numbers.



And save these modified vectors as a new vector.

```
> y = x + c(1,2,3,4)
> y
```

```
[1] 2 6 9 12
```

Note that the R object \mathbf{y} is no longer "Hello World!" - It has effectively been overwritten by assigning new data to the variable

· You can get more attributes than just class. The function str gives you the structure of the object.

```
> str(x)

num [1:4] 1 4 6 8

> str(y)

num [1:4] 2 6 9 12
```

This tells you that x is a numeric vector and tells you the length.

R Help

For any function, you can type <code>?function_name</code>, which will bring up the help page for that function. You can also do <code>help("function_name")</code>.

```
> ## ?str
> ## help("str")
```

Basic Summarization

Here are some simple functions for making calculations on data.

sum(): takes the sum of all numeric variables in a vector
mean(): takes the mean of all numeric variables in a vector
median(): takes the median of all numeric variables in a vector

Review

- · Creating a new script
- · Using R as a calculator
- · Assigning values to variables
- · Performing algebra on numeric variables

Installing packages

install.packages() is the function for installing R packages from the Comprehensive R
Archive Network (CRAN)

You will often get asked to select a mirror from which to download packages to install

These packages are typically installed to one of your personal directories

Try installing the swirl package now - it's a tool for interactively teaching R, which we will use for some labs

install.packages("swirl")

Swirl

```
library(swirl)
install_course_url("http://www.biostat.jhsph.edu/~ajaffe/summerR2014/Summer2014_Swirl.zip")
# swirl()
```

· Start with the "01 Introduction to R" module

```
> info()
| When you are at the R prompt (>):
| -- Typing skip() allows you to skip the current question.
| -- Typing play() lets you experiment with R on your own; swirl will ignore what you do...
| -- UNTIL you type nxt() which will regain swirl's attention.
| -- Typing bye() causes swirl to exit. Your progress will be saved.
| -- Typing main() returns you to swirl's main menu.
| -- Typing info() displays these options again.
```