Introduction to R and RStudio

author: Andrew Flowers date: Wed, Apr 27, 2016 autosize: true font-family: 'Helvetica' Goals ==

- Hands-on
- Complete beginners welcome
- No dumb questions
- 1 hour or less
- Gradually learn more R

Get in the habit of using R, making mistakes and asking questions!

Outline

- Why R?
- Installing R and RStudio
- Tour of RStudio interface
- Working directories
- R data types
- Getting help
- Running an .R script
- Packages
- Reading in data
- Understanding your data

Why R? Hybrid of data analysis and programming

Interactive data analysis - SPSS, Stata, SAS - Command-line interface - Quick and instantaneous - Built by statisticians, for statisticians

---

Writing scripts (programming) - Python, Ruby, C++, JavaScript - Set of instructions - Scripts (or code) automates work - Built by computer scientists, for computer scientists

Installing R and RStudio

- Download and install R: https://cloud.r-project.org/
- Download and install RStudio: https://www.rstudio.com/products/rstudio/download/
- Keep RStudio in dock (Mac)

Tour of RStudio interface

What is RStudio?

- It's an IDE = Integrated Development Environemnt
- A tool that makes using R as effective and easy as possible

What am I looking at? (four quandrants, clockwise from bottom-left)

- Console (or “Terminal”)
- For typing interactive commands
- Source (or “Text editor”)
- For writing a script (or chunk of code)
- Environment/History
- For keeping track of your variables and data
- For reviewing past commands you’ve run
- Files/Plots/Packages/Help
- For reviewing your files and packages you’ve installed
- For viewing plots and getting help

Review settings

Tab and margin settings

- Tab width: 2
- Margin column: 80

Working directories (1/2)

“We all need a place to call home”

- Your working directory is the folder location out of which you’re working
- Use the getwd() command to display your current working directory

```r
getwd()
```

```
[1] "/Users/flowersa/repos/r-learning/sessions/intro-to-R-and-RStudio"
```

- Note: R commands often have a parenthesis “()”" where you input arguments

Make sure you’ve set your working directory correctly

- Use the setwd() command to set your working directory

```r
setwd("/Users/flowersa/repos/r-learning")
```

```r
setwd("/Users/flowersa/repos/r-learning/sessions/intro-to-R-and-RStudio/")
```

Working directories (2/2)

Display the files in your current working directory

- Use the list.files() command

```r
list.files()
```

```
[1] "intro-to-R-and-RStudio-figure" "intro-to-R-and-RStudio.R"
[3] "intro-to-R-and-RStudio.Rpres"
```

- Side note: dir() command does the same thing

```r
dir()
```

```
[1] "intro-to-R-and-RStudio-figure" "intro-to-R-and-RStudio.R"
[3] "intro-to-R-and-RStudio.Rpres"
```

Important notes (1/2)

R has a weird-looking "assingment operator"

```r
x <- 1
x
```

```
[1] 1
```

```r
x = 2
x
```

```
[1] 2
```

While "=" works, please use "<-"

- Short-cut for Mac (two keys): "alt/option" + "-"

Important notes (2/2)

Commenting

- Use the # sign for commenting
- Commenting is text or code that you do NOT want R to run

```r
# Do not display this
```

```r
print("Display this")
```

```
[1] "Display this"
```

- Make sure to comment you code so you can understand it later

Tab completion

- When typing, hit “Tab” and RStudio will suggest completed commands for you

R data types (1/2)

Three “atomic” data types:

1.) Numbers (“numeric”)

```
x <- 1
```

2.) Strings (“character”)

```
y <- 'hello'
# Don't forget the quotes!
```

2.) Boolean (“logical”)

```
z <- FALSE
z
```

```
[1] FALSE
```

R data types (2/2)

An important data type: DataFrames

- DataFrames are like matrices or spreadsheets

```
data <- data.frame(name="Andrew Flowers", age=30, female=F)
data
```

```
            name age female
1 Andrew Flowers  30  FALSE
```

R comes will some built-in DataFrames

```
head(iris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

Getting help and Running a .R Script

- Use the help window to search for information on an R command
- Or, use the ? prefix to a command to see help automatically

```
?data.frame
```

Running a .R script

- Copy some commands we've already written into a .R script
- Name your scripts

Packages

- Use the install.packages() command
- Then the require() command