Introduction to R

C. Ryan Campbell

Duke University

c.ryan.campbell@duke.edu

07 Sept 2017

Overview

1. Intro
2. Goals
3. R Basics
   - Variables
   - Data Frames & Matrices
   - Functions
   - Reading in Data
   - T-Test
   - Exercise

Software-Based Labs

- R
  - Already have all the software needed

Software-Based Labs

- R
  - Already have all the software needed
- bin/bash
  - Waiting for cluster access
  - Next week (fingers crossed)

Software-Based Labs

- R
  - Already have all the software needed
- bin/bash
  - Waiting for cluster access
  - Next week (fingers crossed)
- git
  - Best to wait for groups to be formed
  - Next week as well (fingers crossed)

Software-Based Labs

- Take a minute to download a text editor
  - TextEdit
  - Sublime
  - TextWrangler
  - Notepad++

Software-Based Labs

- Take a minute to download a text editor
  - TextEdit
  - Sublime
  - TextWrangler
  - Notepad++
- Work along with the slides (in R-Studio or an editor), generating a document to test each command

Software-Based Labs

- Take a minute to download a text editor
  - TextEdit
  - Sublime
  - TextWrangler
  - Notepad++
- Work along with the slides (in R-Studio or an editor), generating a document to test each command
- I'll upload my finished document after class as a reference

Today's Goals

- Get familiarized with R basics
- Import your own data
- Visualize it
- Test something about it
- Produce a graph and save the code

Variables

- Notes and tips go here
- Descriptions of actions not meant to be typed

```
within these boxes will be typed into R
it will also have monospaced font
> it may also begin with a carat character,
which symbolizes the prompt
when it is italicized, it is output from R
```

Variables

- Scalars, Vectors, Data Frames
- Access parts of each with brackets:

```
> a <- c(15,12,10,93)
> a[2]
 12
```

Variables

- or multiple parts at once with a second vector

```
> a <- c(15,12,10,93)
> a[c(2,4)]
 12 93
```

Variables

- or with a TRUE/FALSE vector

```
> a <- c(15,12,10,93)
> a[c(FALSE,TRUE,TRUE,FALSE)]
 12 93
```

Variables & Math

- Scalar mathematical operations apply to an entire vector

```
> a * 5
 75  60  50 465
```

Variables & Math

- Vector mathematical operations apply to each entry

```
> a + c(1,1,10,-50)
 16 13 20 43
```

- This is useful because it allows you to manipulate entire sets of data at once
  - Take the absolute value of a column
  - Combine columns to create new measures
  - Plot the log(value) of a dataset instead of the raw value

Data Frames and Matrices

- Used to hold data
- Can keep track of large sets of data
- Like the data from the hypoxia example

Data Frames and Matrices

- Can also be accessed with brackets []
  - Brackets require 2 entries instead of 1
  - because the data has two dimensions
- dataframe[row,column]
- For example, access a column of the data with brackets:

```
> hypoxiaTested[,1]
[1] TIMM TIMM TIMM TIMM TIMM
```

- (leaving either row or column blank returns the entire range)

Functions

- Things you do to/with variables

Functions

- Things you do to/with variables
- Are filled with arguments

Functions

- Things you do to/with variables
- Are filled with arguments
- Examples:

Functions

- Things you do to/with variables
- Are filled with arguments
- Examples:
  - plot()

Functions

- Things you do to/with variables
- Are filled with arguments
- Examples:
  - plot()
  - ttest()

Functions

- Things you do to/with variables
- Are filled with arguments
- Examples:
  - plot()
  - ttest()
  - read.table()

Functions

- Things you do to/with variables
- Are filled with arguments
- Examples:
  - plot()
  - ttest()
  - read.table()
- What are the arguments for these functions?

Functions

- Things you do to/with variables
- Are filled with arguments
- Examples:
  - plot()
  - ttest()
  - read.table()
- What are the arguments for these functions?
- How would you go about answering this question?

?read.table

read.table {utils} R Documentation

Data Input

Description

Reads a file in table format and creates a data frame from it, with cases corresponding to lines and variables to fields in the file.

Usage

```
read.table(file, header = FALSE, sep = "", quote = "\"'",
           dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
           row.names, col.names, as.is = !stringsAsFactors,
           na.strings = "NA", colClasses = NA, nrows = -1,
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,
           strip.white = FALSE, blank.lines.skip = TRUE,
           comment.char = "#",
           allowEscapes = FALSE, flush = FALSE,
           stringsAsFactors = default.stringsAsFactors(),
           fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)

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

read.csv2(file, header = TRUE, sep = ";", quote = "\"",
          dec = ",", fill = TRUE, comment.char = "", ...)

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

read.delim2(file, header = TRUE, sep = "\t", quote = "\"",
            dec = ",", fill = TRUE, comment.char = "", ...)
```

Arguments

file	the name of the file which the data are to be read from. Each row of the table appears as one line of the file. If it does not contain an *absolute* path, the file name is *relative* to the current working directory, `getwd()`. Tilde-expansion is performed

Console ~/

```
> ?read.csv
>
>
>
>
```

Reading in Data

- What are the (relevant) arguments of read.table?
  - file - name of file to bring in, can include location
    - “filename.txt” or “/Users/ryan/Documents/filename.txt”

Reading in Data

- What are the (relevant) arguments of read.table?
  - file - name of file to bring in, can include location
    - “filename.txt” or “/Users/ryan/Documents/filename.txt”
  - header - T/F variable to treat the first row like a header

Reading in Data

- What are the (relevant) arguments of read.table?
  - file - name of file to bring in, can include location
    - “filename.txt” or “/Users/ryan/Documents/filename.txt”
  - header - T/F variable to treat the first row like a header
  - dec - what character represents decimals

Reading in Data

- What are the (relevant) arguments of read.table?
  - file - name of file to bring in, can include location
    - “filename.txt” or “/Users/ryan/Documents/filename.txt”
  - header - T/F variable to treat the first row like a header
  - dec - what character represents decimals

```
 > read.table(‘‘hsap_hypoxia_gene_exp_FORCLASS.diff’’,
header = TRUE, dec = ‘‘.’’)
gene_family sample_1 sample_2...
1 TIMM hypo norm...
```

Reading in Data

- You can store the data as a variable in R
  - This allows for easier manipulation

Reading in Data

- You can store the data as a variable in R
  - This allows for easier manipulation
  - Be sure to name it appropriately

Reading in Data

- You can store the data as a variable in R
  - This allows for easier manipulation
  - Be sure to name it appropriately
  - Variable names are usually all letters
  - Capitalization can help split up words (e.g. hypoxiaTested, hypoxiaTIMM)

Reading in Data

- You can store the data as a variable in R
  - This allows for easier manipulation
  - Be sure to name it appropriately
  - Variable names are usually all letters
  - Capitalization can help split up words (e.g. hypoxiaTested, hypoxiaTIMM)

```
> hypoxiaRAW <- read.table(``hsap_hypoxia_gene_exp_FORCLASS.diff'',
header = TRUE, dec = ``.'')
```

Reading in Data

- You can store the data as a variable in R
  - This allows for easier manipulation
  - Be sure to name it appropriately
  - Variable names are usually all letters
  - Capitalization can help split up words (e.g. hypoxiaTested, hypoxiaTIMM)

```
> hypoxiaRAW <- read.table(``hsap_hypoxia_gene_exp_FORCLASS.diff'',
header = TRUE, dec = ``.'')
```

  - Was there output from the command? Why or why not?

Data Frames & Matrices

- In addition to brackets to access parts of a data frame you can also use the $ in combination with the column name:

```
> hypoxiaTested$gene_family
[1] TIMM TIMM TIMM TIMM TIMM
```

- Gives the same as using [,1] next to hypoxiaTested
- However it has several advantages
  - Clearer in code, so you recognize it later
  - You can use the TAB key to pick a column name within R-Studio

Data Frames & Matrices

- Another useful way to use the $ is to add new data to an existing data frame
- Simply generate data and define it as a new column name:

```
> hypoxiaTested$r_value <- ( hypoxiaTested$p_value +
hypoxiaTested$q_value ) / 2
```

- What variables did I manipulate?
- What is the new “r value” column I created?

Visualize the Data

- You can use functions to see what the data look like:

Visualize the Data

- You can use functions to see what the data look like:
  - Histograms (hist) can show you what a single variable looks like

Visualize the Data

- You can use functions to see what the data look like:
  - Histograms (hist) can show you what a single variable looks like
  - Scatterplots (plot) can show you how variables relate

A Simple T-Test

- To conduct a simple T-Test I'm going to generate some random, normal data:

  (pick your own values for the n and mean to fill out the class chart)

```
> meanzero <- rnorm(100,0,1)
> meanthree <- rnorm(100,3,1)
```

- How would I confirm that this data is normally distributed?

A Simple T-Test

- To conduct a simple T-Test use the t.test() function

```
> t.test(meanzero,meanthree,var.equal = T)

 Two Sample t-test

data:  meanzero and meanthree
t = -21.004, df = 198, p-value < 2.2e-16
```

Bring in your own Data

- Pick some data from the github
- Bring it into a session of R
- Make sure the column names are correct
- Make sure R interprets numbers correctly
  - What do I mean by this?
  - How does one do this?

Bring in your own Data

- Pick some data from the github
  - Biology Data:
    - hsap_hypoxia_gene_exp_FORCLASS.diff
  - Political Data:
    - 2008to2016PresElections.csv
  - Movie Data:
    - movie_metadata_genre_FORCLASS.csv
  - Sports Data:
    - NCAA_BBall_KenPom_summary17.csv
    - NFL_game_scores.csv, Game_Logs_Quarterback.csv, Career_Stats_Passing.csv
- or Choose your own!

Visualize the Data

- Use the functions hist or plot to explore your data

Visualize the Data

- Use the functions hist or plot to explore your data
  - Histograms (hist) can show you what a single variable looks like

Visualize the Data

- Use the functions hist or plot to explore your data
  - Histograms (hist) can show you what a single variable looks like
  - Scatterplots (plot) can show you how variables relate

Visualize the Data

- Use the functions hist or plot to explore your data
  - Histograms (hist) can show you what a single variable looks like
  - Scatterplots (plot) can show you how variables relate
- Take notes on it within your Rmd session (e.g. status1 is experimental condition, value2 is numeric and ranges from...)

Visualize the Data

- Use the functions hist or plot to explore your data
  - Histograms (hist) can show you what a single variable looks like
  - Scatterplots (plot) can show you how variables relate
- Take notes on it within your Rmd session (e.g. status1 is experimental condition, value2 is numeric and ranges from...)
- Plot different variables against each other to see how they relate

Visualize the Data

- Use the functions hist or plot to explore your data
  - Histograms (hist) can show you what a single variable looks like
  - Scatterplots (plot) can show you how variables relate
- Take notes on it within your Rmd session (e.g. status1 is experimental condition, value2 is numeric and ranges from...)
- Plot different variables against each other to see how they relate
- Decide on a pair of variables to test with a T-Test

Test the Data

- Make a hypothesis about two subsets of your data

Test the Data

- Make a hypothesis about two subsets of your data
- Use a T-Test to look for differences in two sets of your data

Test the Data

- Make a hypothesis about two subsets of your data
- Use a T-Test to look for differences in two sets of your data
- Add a plot to show the T-Test results

Assignment - Due 9/14

- Import a dataset
- Create two subsets
- Make a hypothesis about them
- Compare the subsets with T-Test
- Visualize the results

Turn In

- your .Rmd file
- your data file

- So that when I run the .Rmd I see your output

The End