Fast-R

D-Lab

11/1/2018

Table of Contents

[Introduction to R 1](#_Toc528912233)

[How to use R to “do research” 2](#_Toc528912234)

[Navigating RStudio 2](#_Toc528912235)

[Variable Assignment 3](#_Toc528912236)

[Data types 3](#_Toc528912237)

[Data structures 3](#_Toc528912238)

[Vectors 3](#_Toc528912239)

[Data frame 4](#_Toc528912240)

[Load data from files 4](#_Toc528912241)

[Subsetting your data 4](#_Toc528912242)

[In one dimension $ 4](#_Toc528912243)

[In two dimensions [ , ] 4](#_Toc528912244)

[Data summarization 5](#_Toc528912245)

[Visualization 5](#_Toc528912246)

[Statistical testingtest 5](#_Toc528912247)

[Key points: 5](#_Toc528912248)

[Part 2 preview 5](#_Toc528912249)

Download these materials Visit this **Error! Hyperlink reference not valid.** to download these materials. 1. Click the green “CLone or Download” button  
2. Click “Download zip” 3. Extract these files to your Desktop

This is an [.HTML](https://html.com/) file made using [RMarkdown and knitr](https://rmarkdown.rstudio.com/) and will serve as our guide. Also open the “script.R” file - that is where you will show your work!

# Introduction to R

This two-part workshop series will teach you the basics of R programming for “doing research”.

##### Click these links to learn more about D-Lab:

* [workshops](http://dlab.berkeley.edu/training)
* [working groups](http://dlab.berkeley.edu/working-groups)
* [consulting services](http://dlab.berkeley.edu/consulting)
* [data resources](http://dlab.berkeley.edu/data-resources).
  + View the [calendar](http://dlab.berkeley.edu/calendar-node-field-date) for a compact view

##### The first two hours will cover:

* How to **navigate RStudio**
* **Variable Assignment** - how to save data in R
* Different **Data Types** - numeric, integer, character, logical, factor
* Different **Data Structures** - vectors, lists, matrices, and data frames
* Loading a dataset from a .csv file on your computer
* Subsetting your data - one column at a time, or by rows and columns
* Saving your data and figures

##### The second two hours:

* **Data summarization**
* **Visualization**
* **Testing**

### How to use R to “do research”

R can do [a lot!](https://en.wikipedia.org/wiki/R_(programming_language)) R makes it easy to import, clean, summarize, visualize, and test data. Start by thinking about your own interests.

Read literature in your field to help you ask a well-informed research question - where are there gaps in the research? In which directions is research going? What is worthwhile to you? What is possible? What is too much?

You can turn these questions into sets of [testable hypotheses](http://mathworld.wolfram.com/HypothesisTesting.html) and use data summaries, visualizations, and tests to discuss your conclusions.

# Navigating RStudio

1. Script (upper left window pane)

* Scripts are where you can write your **input**
* Click File –> New File –> R Script
* Code are the instructions we write to accomplish tasks
* R syntax is highly specific! # Command + Enter (Mac) and Ctrl + Enter (PC) will run a line of code.

1. Console (lower left)

* This is your **output**
* You can also type directly into the console, but it is harder to save than the script. For this reason it is fun to mess around in the console.
* > is the prompt - this means RStudio is waiting for you to give it instructions.

1. Global environment (upper right)

* When you save data into R as variables (see below), they actually live in a real place that we can see.

1. Plots, packages, help (lower right)

* Plots - this is where your visualizations will appear
* Packages - install packages here
* Help - the help files are really helpful, you just don’t know it yet!

# Variable Assignment

Data are saved into R’s memory via variables. Think of a simple three-piece recipe to assign a variable: 1. Left side: unique\_name 2. Middle: equals sign 3. Right side: value, expression, other variable, function, etc.

Let’s try it. Switch over to the “my\_code” script. In which window pane do these variables appear in RStudio?

# Data types

The class() function will tell us what type of data a variable is. The most common types are: \* Numeric - this is the default type for all numbers in R \* Integer - negative and positive whole numbers, including zero \* Character - always wrapped in quotations! " " \* Logical - only two options (TRUE and FALSE) \* Factor - categorical data

NOTE: for integer data, we must coerce (change) numeric to integer type with the as.integer(). Also try using as.numeric(), as.character(), as.logical(), and as.factor(). What happens? Does it always work?

# Data structures

One obvious reason to use R is because we can store complex data inside of a single variable!

### Vectors

Vectors are organized groups of the same type of data. It doesn’t matter what type the data are, as long as they are all the same.

Use the c() function to concatenate some vectors.

### Data frame

Data frames are ordered groups of equal length vectors. The data.frame() function will combine equal-length vectors into a data frame.

# Load data from files

The read.csv() function makes it super easy to load data from files. However you first must set your “working directory”, or the folder location on your computer that RStudio is pointing to.

Click Session –> Set Working Directory –> Choose Directory, and set it to your “Fast-R” folder.

getwd() will show us the path location of this folder. dir() will show us the contents of this folder. ls() will list the variables in our global environment (same as the upper right window pane).

Let’s load the “gapminder-FiveYearData.csv” dataset as a variable named gap and explore it.

What do these functions do? str(gap) nrow(gap) ncol(gap) dim(gap) names(gap)

# Subsetting your data

Once you get data loaded into R, you might want to clean it up (i.e., fix spelling variations, typos, formatting, etc.) but you might also want to subset it - you might only be interested in certain ranges of values.

### In one dimension $

The dollar sign operator will extract one single column.

### In two dimensions [ , ]

Bracket notation is an open and closed bracket with a comma in the middle. Everything before the comma refers to rows, everything after refers to columns.

For example, gap\_sub = gap[c(1:8, 10) , c(4:8)] will extract only \* rows 1 thru 8 and 10, and \* columns 4 thru 8 … and will save it in a new variable named gap\_sub

Let’s try it!

# Data summarization

# Visualization

# Statistical testingtest

# Key points:

1. Data are saved in **variables**
2. Syntax for **Variable Assignment** Looks like this: \* x = 5 (numeric) \* y = cos(pi) (numeric) \* city = "Berkeley" (character) \* value = TRUE (logical) \* factor1 = factor(variable\_name) (factor)
3. R is comprised of functions and arguments: \* **Functions** perform an action on a thing \* These “things” are called **arguments** - values, expressions, other variables, and even entire datasets. They can also be options inside of functions that we can define or turn on or off.
4. All variables in are have a class, or type. A variable’s **class** determines how we can manipulate that data.
5. **Data structures** allow us to store more than a single number or word inside of a single variable: \* Vectors \* Lists \* Matrices \* Data frames
   * NOTE: lists and matrices will not be covered in this workshop.
6. We can **load data from files** and easily load data from Excel ([.CSV files](https://www.computerhope.com/issues/ch001356.htm) are preferred), and [many other softwares and types.](https://www.datacamp.com/community/tutorials/r-data-import-tutorial)
7. These data often need to be subsetted in one ($) or two dimensions ([ , ]) to make working with the data easier.
8. Part 2 \* Summarize \* Visualize \* Test

# Part 2 preview

What does the following code do? What arguments do these functions take?

?table  
?summary  
  
?hist  
?plot  
?boxplot

What does this code do? Click the first [Useful link](https://ggplot2.tidyverse.org/) to read more about part 2!

install.packages("ggplot2")  
library(ggplot2)  
?ggplot2

What does this function do? Check out [this excellent blog post](http://blog.yhat.com/posts/r-lm-summary.html) to learn more.

?lm