

Introduction to R for Data Management and Analysis

Marcel Ramos, MPH

Session 2

Announcements

- Login to Gitter
 - <https://gitter.im/CUNYSPHCode/IntroR>

Notes on last lecture

- Pull up documentation (`help`, `?`)
- Use `c()` to create a 'vector'
- '+' in the console indicates 'waiting for additional input...'
- relative vs absolute paths
 - when to use them?
- Any questions?

Brief recap

- Basic features of the language
 - interactive and *interpreted*
 - commands are entered in the console / via script
 - the commands are pre-processed before evaluation
 - case sensitive, ignores spaces except between objects and functions
 - an object is any type of variable stored in R (i.e., `data.frame`, `vector`, `function`, etc.)
 - Be familiar with the parts of a function
 - function name, argument, inputs
 - know how to get help (use `?functionname` or `help("functionname")`)
 - Finding help and troubleshooting are critically important
 - check for examples online
 - ask on the `#programming` Slack channel
 - Help pages can be intimidating but useful

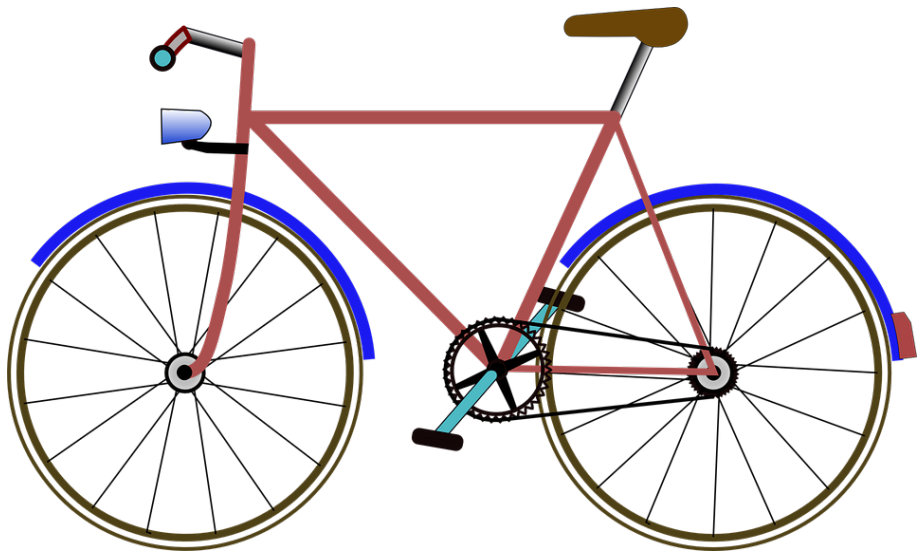
Today's lecture overview

- Common classes in R
 - vector types
 - numeric
 - logical
 - character
 - tabular types e.g., `data.frame`
 - other types
- Creating objects (assignment)
- Class type coercion
- Importing and Exporting Data
- Subsetting (time-permitting)
 - using brackets and dollar signs (`[`, `$`)
 - using vector operands

Common data types (vectors)

- numeric - `c(1, 2, 3)` or `1:3`
- logical - `c(TRUE, FALSE)`
- character - `c("A", "B")`

Classes in R



What are classes?

- R objects have class attributes
- Define what functions/operations can be performed
- `class()` function
- Examples

Vectors, classes, and functions

- Vectors of class
 - character, integer, numeric, logical, complex, raw (bytes)
 - factor: discrete levels
 - mostly used in regressions
 - set reference by releveling categories
 - relevel does not work with ordered factors
 - missing (NA)
- Tabular classes:
 - `data.frame`
 - `matrix`
- Non-tabular:
 - `function`
 - `list`
 - `array`
 - custom classes

data.frame

- A unique type of list with uniform lengths in all elements
- Great for data analysis
- Most common class you will use to do analysis
- The return value of most import functions is usually a form of a `data.frame`
- Use `data.frame` to create a `data.frame`

Exploratory functions on a data.frame

- head
- tail
- dim
- colnames
- rownames
- sapply(x, class)

- can handle only one type of data at a time
- can be character or numeric
- create using `matrix()` function

- custom tabular data class
- 'tidyverse' representation of a `data.frame`
- a neat print output
- metadata on the columns (chr, int, etc.)

- Handle multiple data types in one
- Can contain vectors, data.frames, and even functions
- The `data.frame` is a special type of list
- Use the `list` function to create a list

Assignment (object creation)

- Use the `<-` arrow to assign to an object (variable)
- The object being created is on the left-hand side (LHS)
- LHS can have any arbitrary name
- pseudocode: `object <- value`
- The output or return value of an operation on the RHS is the 'value'
- RHS can be anything that R 'understands'
- *Without* assignment the output will not be 'saved'!

Type (class) coercion

- `as.__(class type here)__`
 - `as.character`
 - `as.numeric`
 - `as.logical`
 - `as.data.frame`
 - `as_tibble` (in 'tibble' package)

Importing and Exporting Data

- Recognizing file types
 - File extensions (e.g., .tsv, .csv, .xlsx, .txt, .sav, .sasb7dat)
- Show file extensions on Windows
 - Windows File Explorer > View tab > Show File Extensions
- Downloading files from the internet
 - `read.csv()`
 - `download.file()`

Packages for reading foreign data

- readr, readxl, haven
 - readr - provides fast and efficient read-in for large files
 - readxl - allows you to read MS Excel files (.xls, .xlsx)
 - haven - support for SPSS, SAS, and other data

Exporting Data

- `write.table()`
- `write.csv()` / `write_csv()`
- `write_delim()`