SOCIAL DATA SCIENCE

Introduction to R

Sebastian Barfort

August 05, 2016

University of Copenhagen
Department of Economics

WELCOME

Course Description

> *The objective of this course is to learn how to analyze, gather and work with modern quantitative social science data.*

We will do this using a program called R.

R

Advantages

Free, open source statistical programming language

Offers a massive set of packages for statistical modelling, machine learning, visualisation, and importing and manipulating data

An enthusiastic community (Stackoverflow, R-help mailing list)

Used by New York Times, Facebook, Google, Twitter...

Disadvantages

R is not perfect

R is not always the best tool for everything

R works for small/medium sized data

WORKING WITH R

Today

1. Arithmetic operations
2. Creating objects
3. Installing packages
4. Importing data
5. Functions
6. Getting help

LEARNING HOW TO CODE (I)

Hadley Wickham

> *The bad news is that when ever you learn a new skill you're going to suck. It's going to be frustrating. The good news is that is typical and happens to everyone and it is only temporary. You can't go from knowing nothing to becoming an expert without going through a period of great frustration and great suckiness.*

LEARNING HOW TO CODE (II)

Kosuke Imai

> *One can learn data analysis only by doing, not by reading.*

ADVICE

Do not use the console, write scripts instead

Be lazy (write functions)

Think before you code

Code is a medium of communication

1. Between you and the computer
2. Between you and other people (or future you)

TWO TYPES OF CONSTRAINTS

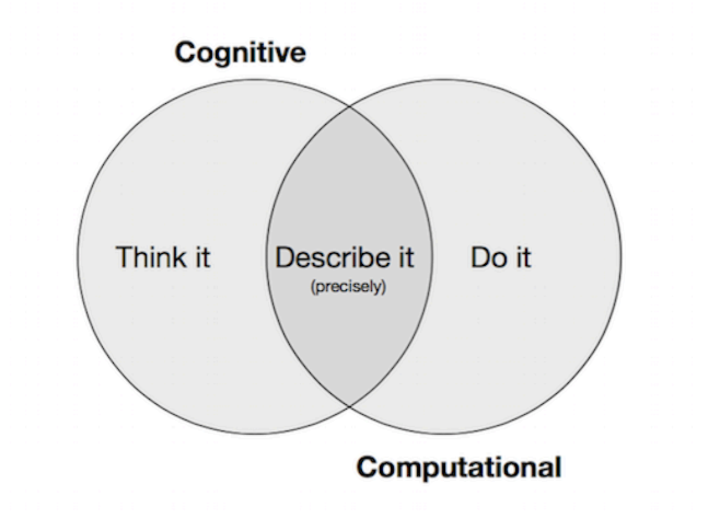

Figure 1:

Arithmetic in R

R AS A CALCULATOR

You can use R to do standard arithmetic operations

```r
1 + 100
```

```
## [1] 101
```

```r
7 / 2
```

```
## [1] 3.5
```

```r
sqrt(3)
```

```
## [1] 1.732051
```

Objects

CREATING OBJECTS

R Rules

Everything has a name

Everything is an object

Every object has a class

OBJECTS

R stores all information as an object with a name of our choice

Objects are created using an assignment operator (`<-` or `=`)

```
y = ”welcome to social data science”
y
```

```
## [1] ”welcome to social data science”
```

```
class(y)
```

```
## [1] ”character”
```

We create and manipulate objects by feeding them to functions and getting output back as a result.

MORE OBJECTS

```
x = c(1, 3, 100)
class(x)
```

```
## [1] ”numeric”
```

```
x * 2
```

```
## [1]   2   6 200
```

```
y * 2
```

```
## Error in y * 2: non-numeric argument to binary operat
```

OBJECT FUNDAMENTALS

Examples of R objects

- character string (e.g. words)
- number
- vector
- matrix
- data frame
- list

We verify the class of an object using the `class` function

EXAMPLE

Question: What is the class of the objects given below?

```
z = ”text”
p = c(1, 3, 5)
q = 2
y = NA
k = FALSE
```

SPECIAL VALUES

- `NA`: not avaliable, missing (`is.na`)
- `NULL`: undefined (`is.null`)
- `TRUE`: logical true (`isTRUE`)
- `FALSE`: logical false (`!isTRUE`)

SYMBOLS

<table>
<tr><th>Operator</th><th>Meaning</th></tr>
<tr><td>&lt;</td><td>less than</td></tr>
<tr><td>&gt;</td><td>greater than</td></tr>
<tr><td>==</td><td>equal to</td></tr>
<tr><td>&lt;=</td><td>less than or equal to</td></tr>
<tr><td>&gt;=</td><td>greater than or equal to</td></tr>
<tr><td>!=</td><td>not equal to</td></tr>
<tr><td>a | b</td><td>a or b</td></tr>
<tr><td>a &amp; b</td><td>a and b</td></tr>
</table>

VECTORS

The most basic type of R object is a vector.

There is really only one rule about vectors in R, which is that a vector can only contain objects of the same class.

We create vectors using the `c` (concatenate/combine) function

```
my_vector = c(1, 3, 5, 10)
another_vector = 1:100
a_third_vector = c(”yes”, ”no”, ”hello”)
my_logical_vector = c(TRUE, FALSE, FALSE, TRUE)
```

DATA FRAMES

R stores spreadsheet like data in a `data frame`

These are really collections of vectors of the same length

Tip: Create data frames whenever you can

WORKING WITH DATA FRAMES

We select variables using $, known as the component selector

You can also call variables/observations using indexing

We can select the first row and the first column

```
df[1, 1]
```

Select the entire first column

```
df[, 1]
```

Select the second row

```
df[2, ]
```

WORKING WITH DATA FRAMES

Some useful functions for working with data frames:

- `names`: returns the column names of the data frame
- `rownames`: returns the row names (if any) of the data frame
- `summary`: returns summary statistics
- `head`: returns the first 5 or 10 observations of the data frame

Installing Packages

INSTALLING PACKAGES

On its own, R can't do all that much

To really make use of R's capabilities, we need packages

A package bundles together code, data, documentation, and tests

We install packages from two sources

- the Comprehensive R Archive Network (CRAN)
- github

INSTALLING PACKAGES FROM CRAN

We can install the `readr` package, for example, by running

```
install.packages(”readr”)
```

Afterwards, we can access all the functions available in the package by running

```
library(”readr”)
```

INSTALLING PACKAGES FROM GITHUB

It’s slightly more difficult to install from github since we need to load a package from CRAN first: `devtools`

Installing from Github now looks like

```
library(”devtools”)
install_github(”hadley/purrr”)
```

The `purrr` package can now be loaded using the `library` command

```
library(”purrr”)
```

Importing Data

IMPORTING DATA: RIO

There is a new package that reads almost all file formats: rio

```
install.packages("rio")
```

Only two functions: import and export

IMPORTING DATA: BASE R

Base R includes functions for reading flat files: `read.csv`, `read.table`, etc.

But I suggest using them only if `rio` fails

They are slower and have bad defaults (`stringsAsFactors = TRUE`)

Functions

FUNCTIONS

Functions operate on objects

R has many built in functions such as `summary`, `mean`, `table`, etc

```r
x = 1:10
mean(x)
```

```
## [1] 5.5
```

```r
sd(x)
```

```
## [1] 3.02765
```

```r
median(x)
```

```
## [1] 5.5
```

WRITING FUNCTIONS

```
my_function = function(input1, input2, ..., inputN)
  {
  # define ’output’ using input1,...,inputN
  return(output)
  }
```

Getting Help

GETTING HELP

If you know the command, type `?` followed by the function in the console

```
?summary
```

Search your version of R using `??` followed by the function name

Use Google and Stackoverflow

Exercises

YOUR TURN

You will work in groups on Exercise 1.5.1 from Imai (2016).

The data is available here

You can read the data as follows

```
library(”rio”)
filepath = ”https://raw.githubusercontent.com/
            kosukeimai/qss/master/INTRO/
            turnout.csv”
df = import(filepath)
```

QUESTIONS

1. Load the data into R and check the dimensions of the data. Also, obtain a summary of the data. How many observations are there? What is the range of years covered in this data set?
2. Calculate the turnout rate based on the voting age population or VAP. Note that for this data set, we must add the total number of eligible overseas voters since the VAP variable does not include these individuals in the count. Next, calculate the turnout rate using the voting eligible population or VEP. What difference do you observe?
3. Compute the difference between VAP and ANES estimates of turnout rate. How big is the difference on average? What is the range of the difference? Conduct the same comparison for the VEP and ANES estimates of voter turnout. Briefly comment on the results.