Quantitative Content Analysis: Lecture 2

Olga Kononykhina

15 Februar 2017

Today's Outline

R and RStudio

- What it is (and what it's not)
- Quantitative vs. qualitative text analysis
- Two examples

Intro to R (Part I)

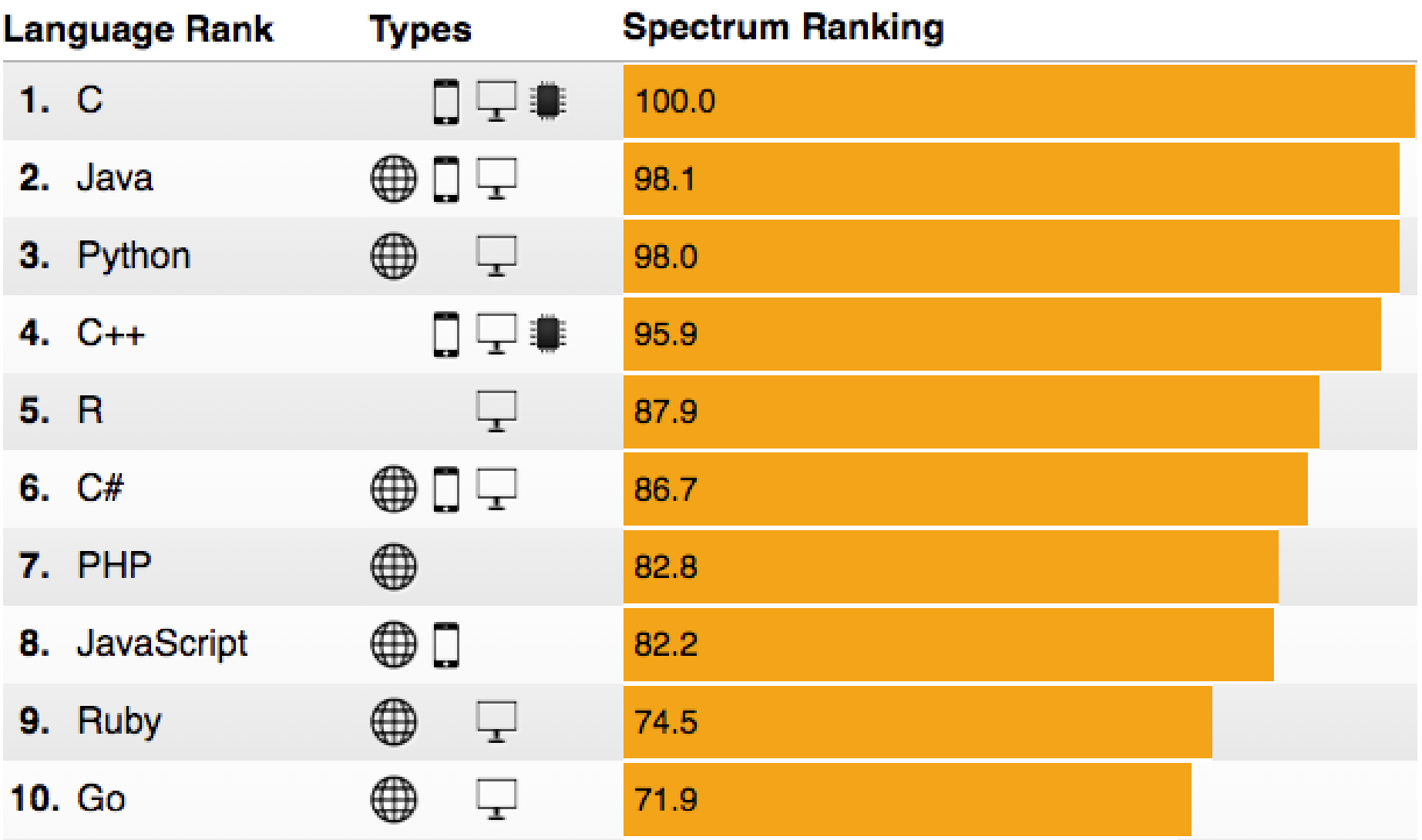
- Basic syntax
- Assigning objects
- Scalars, vectors, matrices
- Control flow: Loops and conditions

R

What is R?

- Open source programming language
- Particular focus on statistical programming
- Originally developed as 'S' by Bell Labs in 1993

R in comparison



(source: IEEE Spectrum)

R interface (console-only)

- Command line, unlike SPSS/Stata
- An interpreted programming language
- Purist's interface: Text-editor & CTRL-C / CTRL-V

R with RStudio

RStudio: An IDE for R

- RStudio is an Integrated Developer Environment (IDE) that makes using R easier
- Code Editor
  - Code highlighting/completion, indentation, ...
  - Feed code from editor to R-console
- Projects‘ to manage your work
- Workspace viewer
- Data Browser
- Enhanced output viewer
- Help browser

Using RStudio

RStudio at startup

- Script editor
- Console
- Workspace
- Multi-purpose-panel

Projects

- A working directory for each project
- Code: .r files
- Dataset/Workspace: .Rdata files

Using RStudio (II)

Basic Workflow:

- Edit in code editor (.r-file)
- Paste to console
- Save Workspace/Datasets (.Rdata-file)
- Save code routinely (no auto-save!)

Shortcuts

- Run code from editor: Select line and ctrl+Enter
- Change windows: ctrl+1, ctrl+2
- Purge dicconsole: ctrl+L
- „Arrow up“ gives you the last line of code in the console
- Alt+Shift+K to see all keyboard shortcuts

Using RStudio (III)

Basic Syntax:

- Use # to comment code (will not be run)
- Case-sensitivity: `data` vs `Data`
- Assigning objects: = and <-
  - `i=5` or `i<-5` stores the scalar `5` under the name `i` in your workspace
  - Outputting works by calling the name: `i` +Enter
- Functions perform operations on the input given
  - End in ()
  - `class(x)`

Fundamentals of the R language

R is object-oriented

- Objects are R's nouns and include (not exhaustive):
  - character strings (e.g. words)
  - numbers
  - vectors of numbers or character strings
  - matrices
  - data frames
  - lists

Special values in R

- `NA`: not available, missing
- `NULL`: does not exist, is undefined
- `TRUE`, `T`: logical true. **Logical** is also an object class.
- `FALSE`, `F`: logical false

Operations on scalars

- Can use it just like a calculator: 2+3 or 2^5
- Assign intermediate quantities
- Functions on scalars: factorial(x)
- Type of input: numeric

Vectors

A container of objects put together in an order

- Has a length, and each item holds something (or missing 'NA')
- Can define them using c() or numeric() (. . .)
- Can concatenate them
- Can find their length
- Can reference their components

Vectors (II)

Operations on vectors

```
sort(x), sum(x), mean(x),; median(x); var(x); sd(x);
sample(x,size,replace=FALSE); factorial(x)
```

Vectors (III)

Task 1

- Calculate the mean of the vector 1,99,3,4,5,6,8.
  - What's the mean if you out the 'outlier'?

Matrices

- A square 2 dimensional container i.e. vectors combined by row or column**
- Must specify number or rows and columns**
  - matrix(x,nrow,ncol,byrow)
  - x: vector of length nrow*ncol
  - nrow: number of rows
  - ncol: number of columns
  - byrow: TRUE or FALSE, specifies direction of input

Matrices (II)

Task 2

- Assign a 6 x 10 matrix with 1,2,3,. . . ,60 as the data.
  - `matrix(x,nrow,ncol,byrow)`

Matrices (III)

Referencing matrices

- Like vectors, can reference matrices by elements
- `a[i,j]` refers to the ith row, jth column element of a
- Can also reference rows/columns, these are vectors
- `a[i,]` is ith row, `a[,j]` is jth column

Matrices (IV)

Task 3

- Extract the 9th column of the matrix from the previous problem.
  - How can you find the 4th element in the 9th column?

For() loops

- Used to loop around a vector/matrix to do something
-
```
for (i in 1:10) {
 a[i]= **something**
}
```
- Does 'something' to a[1],a[2],a[3],. . . , a[10]
- You can 'nest' a `for()` in another `for()`

For() loops

Task 4

- Create a matrix with `matrix(1:20,nrow=4,ncol=5)` and another with `matrix(NA,nrow=4,ncol=5)`
- Adapt the second to the first matrix using `for()`
  - (hint: define a 'counter' variable that increments by 1 each time you move to the next cell)
- How can you fill the matrix by rows?
  - (i.e. so it looks like matrix(1:20,nrow=4,ncol=5, byrow=T)

If() statements

- Used to make conditions on executing some code
- ```
  if (**condition**) {
  **action**
  }
  ```
- If condition is true, action is done
- Tests for conditions: `==`; `>`; `<`; `>=`; `<=`; `!=`
- ==and =are different
- Check condition on its own first
  ```

If() statements (II)

Task 6

- `a=sample(c(4,0),20,replace=TRUE)`
- `b=matrix(a,nrow=4,ncol=5)`
- Now recode all the 4s into 1s using if() and for() statements.

Help

- Type help( ) to see documentation
- Check the R-Introduction (Syllabus)
- Check out the help function for a couple of functions used in today's course to see 'what is what' in the documentation

Next Session

- More sophisticated objects (dataframes, lists etc.)
- Basic plotting
- Reading and saving data
- Basic text-manipulation with R