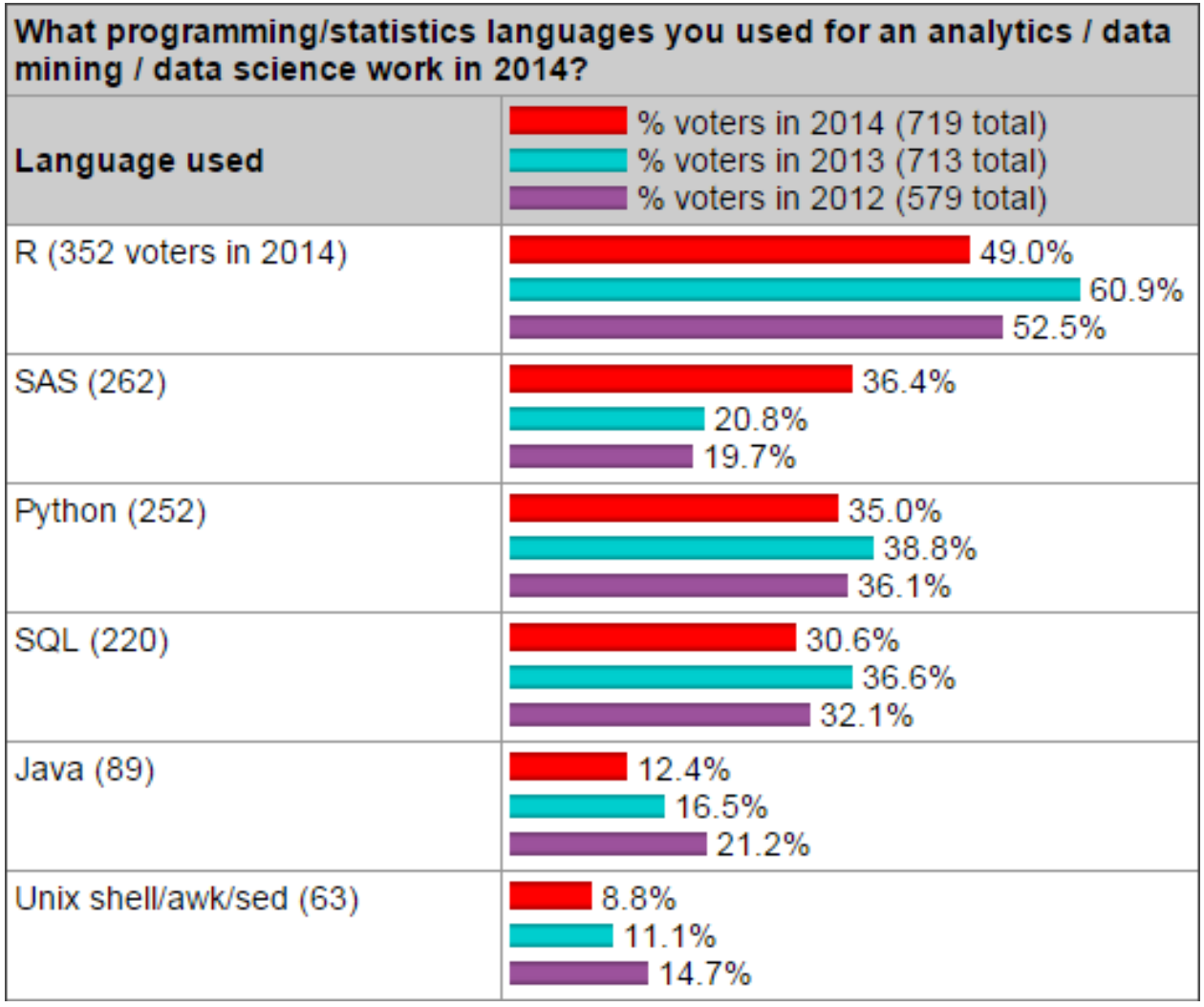
Introduction to R

January 11, 2015

What languages are used for data analysis?



http:

//www.kdnuggets.com/polls/2014/languages-analytics-data-mining-data-science.html

Requirements

- Data-centric
  - The main goal is to explore the data.
  - Users must be able to ask meaningful questions quickly and flexibly.
- Trustworthy
  - Results *can be shown* to be correct.

What about performance?

- Data analysis is a creative process of exploration;
  human time and efficiency are most important.
- Data processing can be automated;
  CPU time becomes important at this step.

About R

- R is an open source implementation of S language.
- Designed for statistical and data analysis.
- Functional style of programming.
- Object-oriented style of programming.
- Data frame is a built-in type.
- Missing values are built-in.
- Model formulas are first-class objects.
- Contains advanced statistical routines.
- State of the art graphics capabilities.

R Books and Resources

- *R Cookbook*: `http://www.cookbook-r.com/`
- *The R Inferno*: `http://bit.ly/1mpZabc`
- *Quick-R*: `http://www.statmethods.net/`
- *Software For Data Analysis: Programming with R*
  by John Chambers

Installation

- `http://www.r-project.org/`
- In Windows:

  `http://cran.utstat.utoronto.ca/bin/windows/base/R-3.2.3-win.exe`
  - There may be a problem with privileges when installing packages.
  - If you don't have admin rights simply change the installation directory to one where you have write permissions.
- In Linux or Mac OS X: Use the package system of your OS to install R.
  - Building from source is possible, although not straightforward.

Development environments

- R-studio: `http://www.rstudio.com/`
- Tinn-R: `http://sourceforge.net/projects/tinn-r/`
  Windows only
- Notepad++: `http://notepad-plus-plus.org/`
  with NppToR plug-in, Windows only

Basic types

- vector: logical, integer, double, double complex, string
  - each has a missing value literal: `NA`
  - `NULL` is not `NA`!
  - scalars are vectors of length 1
- structured: list, factor, data frame, array, matrix, etc.
- factor represents categorical variable
  - similar to vector of strings
  - stored as an integer vector
- data frame represents observations of statistical variables
  - behaves as a list of vectors
  - also behaves as a matrix

Basic operators

- usual arithmetic, comparison, logical operators are elementwise
- for matrix multiplication use `%*%`
- There are 5 assignment operators!!!
  - `=, <-, ->, <<-, ->>`
  - best practice is to use `<-`
- There are 3 indexing operators
  - `[[` access to a single element by index or name
  - `[` access to multiple elements by index or name
  - `$` access to single element by name

Basic flow control

- `if(cond) expr`
- `if(cond) cons.expr else alt.expr`
  `if` returns the value of the evaluated expression
- `for(var in seq) expr`
- `while(cond) expr`
- `repeat expr`
  loops return value `NULL`
- `break`
- `next`

Hint: avoid loops, they can be very slow!

Functions

- Functions are first-class objects
- Function arguments passing can be complicated!
  - parameters are assigned by position or name
  - use = to assign parameter by name
  - lazy evaluation
- User defined functions are closures with lexical scope!

Hint: try using an `*apply` instead of a loop whenever possible

- Load one of the standard datasets in R

```
> data(cars)
```

- It's a data frame with two variables. Compute the mean of each variable.

```
> colMeans(cars)
```

- Compute the minimum of each variable.

```
> for(i in 1:2) print(min(cars[,i]))
> sapply(cars, min)
```

- Plot

```
> plot(cars)
```

- Linear regression

```
> reg <- lm(dist ~ speed, data=cars)
> abline(coefs=reg$coefficients)
```

- Summary

```
> summary(cars)
> summary(reg)
```