R
Története, alapjellemzők

Csima Judit

BME, VIK,
Számítástudományi és Információelméleti Tanszék

2014. február 12.

Vázlat

1. Mi is ez az R?
   - Történet
   - Az R jellemzői

Az R gyökerei

- az R az S nyelv egy dialektusa
- az S egy programozási nyelv és környezet statisztikai számításokhoz
- John Chambers (az S nyelv fő fejlesztője):
  “[W]e wanted users to be able to begin in an interactive environment, where they did not consciously think of themselves as programming. Then as their needs became clearer and their sophistication increased, they should be able to slide gradually into programming, when the language and system aspects would become more important.”
  http://www.stat.bell-labs.com/S/history.html

Az R története

- 1991: Ross Ihaka and Robert Gentleman (Új-Zéland)
- 1995: meggyőzik Ihaka-t és Gentleman-t hogy az R legyen open source, szabad szoftver (GNU)
- 1996: Az R-help és R-devel levlisták megszületnek
- 2000: R 1.0.0 verzió
- 2013: R 3.0.2. verzió 2013. szeptember 25-én (gyakori frissítések)

Az R jellemzői

- szinte minden platformra van (Windows, Linuxok, Mac)
- rendszeres fejlesztések, hibajavítás
- moduláris felépítés, package-ek
- fejlett grafikai lehetőségek
- interaktív és programozási használat (fokozatos átmenet)
- aktív közösség (R-help, R-devel listák)
- ingyenes (http://www.fsf.org)

Az R felépítése

- package-ek
- "alap" R: ebben minden benne van, ami kell, ha nem akar az ember spéci dolgokat
- például az "alap" R-ben van a **base** package, ami a legalapvetőbb függvényeket tartalmazza
- további "alap" package-ek: **utils, stats, datasets, graphics, grDevices, grid, methods, tools, parallel, compiler, splines, tcltk, stats4**
- letölthető még kb. 4000 package, nagyrészt a felhasználók munkái
- letöltések, sok hasznos infó: http://cran.r-project.org
- R főoldal: http://www.r-project.org/

Az rstudio

- egyben minden, ami kellhet, ha az ember R-t akar használni
- szövegszerkesztő kódíráshoz
- konzol az interaktív munkához
- help-ablak
- mindenféle platformra, letölthető: http://www.rstudio.com/

AI and Social Science – Brendan O'Connor

cognition, language, social systems;
statistics, visualization, computation

Comparison of data analysis packages: R, Matlab, SciPy, Excel, SAS, SPSS, Stata

Posted on February 23, 2009

Lukas and I were trying to write a succinct comparison of the most popular packages that are typically used for data analysis. I think most people choose one based on what people around them use or what they learn in school, so I've found it hard to find comparative information. I'm posting the table here in hopes of useful comments.

Name	**Advantages**	**Disadvantages**	**Open source?**	**Typical users**
R	Library support; visualization	Steep learning curve	Yes	Finance; Statistics
Matlab	Elegant matrix support; visualization	Expensive; incomplete statistics support	No	Engineering
SciPy/NumPy/Matplotlib	Python (general-purpose programming language)	Immature	Yes	Engineering
Excel	Easy; visual; flexible	Large datasets	No	Business
SAS	Large datasets	Expensive; outdated programming language	No	Business; Government
Stata	Easy statistical analysis		No	Science
SPSS	Like Stata but more expensive and worse			

[**7/09 update:** tweaks incorporating some of the excellent comments below, esp. for SAS, SPSS, and Stata.]

There's a bunch more to be said for every cell. Among other things:

- Two big divisions on the table: The more programming-oriented solutions are R, Matlab, and Python. More analytic solutions are Excel, SAS, Stata, and SPSS.
- Python "immature": matplotlib, numpy, and scipy are all separate libraries that don't always get along. Why does matplotlib come with "pylab" which is supposed to be a unified namespace for everything? Isn't scipy supposed to do that? Why is there duplication between numpy and scipy (e.g. numpy.linalg vs. scipy.linalg)? And then there's package compatibility version hell. You can use SAGE or Enthought but neither is standard (yet). In terms of functionality and approach, SciPy is closest to Matlab, but it feels much less mature.
- Matlab's language is certainly weak. It sometimes doesn't seem to be much more than a scripting language wrapping the matrix libraries. Python is clearly better on most counts. R's is surprisingly good (Scheme-derived, smart use of named args, etc.) if you can get past the bizarre language constructs and weird functions in the standard library. Everyone says SAS is very bad.
- Matlab is the best for developing new mathematical algorithms. Very popular in machine learning.
- I've never used the Matlab Statistical Toolbox. I'm wondering, how good is it compared to R?
- Here's an interesting reddit thread on SAS/Stata vs R.
- SPSS and Stata in the same category: they seem to have a similar role so we threw them together. Stata is a lot