Introduction to R

Statistical computing

Arni Magnusson

Hafro, 8 Nov 2010

Outline

Outline

Statistical software

GUI	**Excel**, SPSS, Statistica
Interpreted	BUGS, Gauss, Matlab, Python, **R**, SAS, Stata
Compiled	**ADMB**, C++, Fortran, Java

Excel / OpenOffice

Spreadsheets are great tools for many tasks in scientific work:

- Initial exploration of data
- Final summary of analysis
- Organize (projects, data sources, cost, timeline, people)

Extremely limited and unreliable for statistical analysis

Use only `+ - / *`, `sum`, average, and statistical software for everything else

Using the right tool

Imagine writing a 20-page text document in Excel

$\Rightarrow$ inferior quality, hard to modify, prone to errors

Likewise, R is not always the right tool in statistical computing:

Databases for large amounts of data

C or Fortran for computationally intensive subtasks

AD Model Builder for nonlinear models

Outline

R features

Large collection of tools for statistical analysis, constantly updated by a large user commity, including leading authors in statistical fields

Graphics for exploratory analysis and publications

Language for expressing statistical models, object-oriented and extensible by users

Embraced by university stats departments around the world

R history

S Programming language, first version in 1976, now 4. Created by John Chambers et al., Bell Laboratories.

S-Plus Statistical software based on S, first version in 1988, now 8.1. Created by R Douglas Martin, maintained by TIBCO Inc. Individual license is \$199/month. Most developers and users have moved from S-Plus to R by now.

R Statistical software based on S, first version in 2000, now 2.12. Created by Ross Ihaka and Robert Gentleman, maintained by R Development Team. Free software.

R Project website

http://www.r-project.org

Download R, manuals, etc.

Outline

Open source

Most R functions are written in the R language, and the full code is shown if you type the name of the function.

Low-level functions and R itself are written in C, and the full code can be browsed at `http://svn.r-project.org/R/trunk/`.

This access to the source code is of critical value for complex statistical models.

Open source principles (making a thorough description of methods publicly available) have been a foundation of scientific research for centuries.

Repeatable research

How meaningful is the sentence

> *We used the GetValues module in AnalyzeThis 2.1 (Biotech Inc. 1990) to estimate the . . .*

in a journal article that was published 20 years ago?

The software is no longer available, and the printed user manual is not archived anywhere.

Open source statistical software from the 1970s and 1980s continues to be available for download on the web. Statistical methods can be extracted from the code, so studies using that software are repeatable.

Scientific method

Open source statistical software has become a cornerstone of scientific inference, and is a modern element of the scientific method. Medical research, astronomy, everywhere.

The software development process is a collaborative effort of scientists worldwide, and relies on users contributing code, documentation, bug reports, etc.

The R Development Team consists of 19 professors and senior scientists in 10 different countries.

Hafro staff are involved in the development of R packages and other statistical software (Gadget, AD Model Builder) that is used around the world.

Open source vs. proprietary software

Statistical computing: **open source**

- publicly available description of methods
- anyone can repeat the analysis
- better performance, 1000s of developers

Other software: **personal choice**

- performance
- time required to learn
- what colleagues use
- cost

Outline

First steps in R

Install

At home download from `http://www.r-project.org`

At Hafro contact `help@hafro.is`

Configure

Windows Edit - GUI preferences - ... - Save
Shortcut (startup options, shortcut key)

Linux `options(help_type="html")`

Optional create '.Rprofile' in 'HOME' directory
startup options: `--quiet --save` (or `--no-save`)

Calculator with functions

```
2 + 2
```

```
sqrt(10)
```

```
log(10)
```

```
# try the up arrow
```

Objects in workspace

```
x <- 2

10 * x


ls()

rm(x)

rm(list=ls())
```

Data objects

Vectors

```
stack.loss
```

```
month.abb
```

Data frames

```
BOD
```

```
Puromycin
```

Select column

```
Puromycin$conc
```

Plots

```
x <- 1:10

y <- 3 * x


plot(x, y)


y <- 100 * x
```

The plot is not "alive", so the y coordinates are not updated unless `plot(x, y)` is called again

Help system

```
help(log, help_type="html")

help(log)

?log


args(log)
```

Help system

If you get an error message:

- press the up arrow and try to rewrite
- the error message sometimes describes the problem

If R doesn't respond to user input:

- press the `Esc` key