Porting R to the Macintosh

by Stefano M. Iacus

In late July of last year, Ross Ihaka kindly gave me the code he used to build version 0.64 of R on the Macintosh while the current R release passed version 1.0. So some of the newest code didn't fit these old sources and the Macintosh device needed also an update due to the migration of MacOS towards MacOS X. In any case, without that initial starting point, there probably wouldn't now be a beta release of "R for Mac" 1.2.1 that is quite stable and equivalent to different platform binaries.

What is the actual state of this port? Probably, starting from one of the 1.2.x releases the Macintosh code will be included in the standard distribution of the sources as well as the binaries. The currently available release of binaries and sources for MacOS is 1.2.1 and can be found starting from the R Developer Page (`http://developer.r-project.org/`), as to say, it is still a beta developer version.

What is implemented that was not in the past? Actually, one of the most important things is the support for dynamic libraries[2] on which most of the contributed packages are based. Also the XDR save/load format has been implemented so that Macintosh users can exchange '.RData' session files with other platforms having this ability (i.e., Windows systems). Also temporary files, the help (`man`-like) engine, and the file I/O routines needed for the Postscript, PicTex and XFig devices work correctly. Many minor bugs have been fixed and some further functionality has been added like the metric information technology to write mathematical expressions on plots.

Environment variables actually do no exist under MacOS systems as well as the shell. Much functionality based on these two has been partially suppressed, other such functions have been implemented using system-level interactions with the Finder (a sort of window manager under MacOS). Some of the environment variables are stored in the preference file of the R Application.

Luckily, the memory manager has changed in version 1.2.0 of R and due to this fact the current release of R for MacOS ignores the size settings. The R task can use as much memory as it likes till the memory space reserved to the application by the user is full. The user simply needs to change memory parameters using a Finder/Information procedure as she does for all other applications under MacOS.

What is it left to do? The Tcl support and interactions with other applications are in the works. These imply the interactions with Emacs-like editors and the simulation of the `system` command under other systems. One other important need for R for Macintosh is the migration toward MacOS X. One first step is to substitute completely all the former system's API with the new Carbon API provided by Apple. This will allow a Macintosh machine with System 8.6 or greater to run applications either under the standard MacOS and under the new MasOS X.

MacOS X will soon be released by Apple. This system can be viewed as a Unix-like environment with a window manager that looks like the former Macintosh Finder. Under MacOS X the environment variables, different shells, Tcl/Tk support and many other utilities are implemented.

One drawback of MacOS X is that it needs very powerful and new-generation machines to run. The temptation to simply recompile R sources for Unix under MacOS X and to write a specific MacOSX device is always there but this will imply to abandon most of the Macintosh users to their fate without R.

So, in my opinion, migration to MacOS X can be implemented in two steps: the current one that simply tries to make R a native application both under MacOS Systems $\leq$ 9.04 and MacOS X using the Carbon Support by Apple. At the same time taking care for specific MacOS X facilities when available and write a MacOSX device parallel to the current Macintosh one. At least for one year or so this should be a good strategy, then a complete abandon of pre-MacOS X systems will be acceptable for the Macintosh audience.

Currently Macintosh specific code is handled using `#ifdef macintosh` and the Macintosh driver is simply implemented by the `Macintosh()` function. A not so bad idea, could be to use `#ifdef macosx` and `MacOSX()` respectively to take advantage of the new MacOS X.

There is really a great need for R for the Macintosh. Since the first public pre-alpha release of R 1.0, I've received a lot of feedback from users. This feedback has been invaluable. I would like to thank particularly A. Antoniadis, R. Beer, G. Sawitzki and G. Janacek among the others, and of course Luke, Martin, Kurt, Brian and Guido from the R Team helping me with different hints, tips and tricks along with C-code.

Stefano M. Iacus
Università degli Studi di Milano, Italy
`stefano.iacus@unimi.it`

[2] for `dyn.loading` compiled C or Fortran functionality