DSC 2001 Proceedings of the 2nd International Workshop on Distributed Statistical Computing
March 15-17, Vienna, Austria
http://www.ci.tuwien.ac.at/Conferences/DSC-2001
K. Hornik & F. Leisch (eds.) *ISSN 1609-395X*

R porting for the Macintosh

Stefano M. Iacus*
Department of Economics
University of Milan
Via Conservatorio 7, I-20122, Milan.

Abstract

We briefly discuss here the current status, along with its past and future development, of the porting of R for the Apple Macintosh machines. We also touch incidentally the migration from the old MacOS to the new MacOSX system an their R implementations.

1 The history and the future of R for Mac

A first binary release dates from 1995 and it was developed by the original R creators Ross Ihaka and Robert Gentleman. It was a pre-Alpha release. Next Macintosh release has been developed and released later in 1998. Most of the GUI has been rewritten and several functionality were added by Wing Wong Wan. This was a 0.64 R release that still was missing dynamic loading library support. Also metric information for the `Macintosh()` device was not available and temporary file were not implemented as well. When I received the sources for 0.64 release these were also out of date with respect to the current 1.0 release of R and the project files – the analogous of the makefile under shell based systems – where too old to be used with the new release of the Macintosh C compiler. But anyway, this was an essential starting point and the Macintosh community now have an R application to be used natively on their machines. The last current released version has been out together with the official 1.2.2 release of R. Both binaries and sources are now available at CRAN.

*E-mail address: *stefano.iacus@unimi.it*

1.1 The current 1.2.2 release

Now the R application for the Macintosh is functionally equivalent to other implementations. R behaves like standard Macintosh application on MacOS but almost differently from what a typical R user expects if it comes from different platforms (Unix-based or MS-Windows). This is due to the peculiar conception of the MacOS. In particular, there is no shell available (not even the idea of it) and the applications do not expect any file redirection from the user. There is also no straightforward equivalent to environment variables. In same cases it has been possibile to emulate `Sys.getenv()` function and file redirection. Nothing can be done for the C `system()`-like commands in general, but just *ad-hoc* procedures have to be written. This is the case when a user wants to call the external HTML browser or simply wants to read compressed help files. Apart from these complications, the R application can now handle temporary files and, most important, it has support for dynamic loading of shared libraries. This means that most of the packages can be used with this release. Table 1 is a list of the packages currently available. This list grows rapidly and the limitation in the number of packages is almost a question of time. Concerning the redirection of the I/O of the R application, this is currently implemented in a way that is familiar to the Macintosh user but probably counterintuitive for the rest of the R users. A user can specify I/O files when she run R. At R startup a dialog box will appear (see Figure 1.1) where the user can specify input and output files simply browsing his disks. There is also the possibility to specify some options, but very few of them are supported. In particular the `R CMD` facility is not available.

akima	base	boot	bootstrap
cclust	class	cluster	ctest
date	eda	foreign	gld
integrate	KernSmooth	lqs	MASS
mclust	modreg	multiv	mva
nlme	nls	nnet	polymars
polynom	princurve	pspline	rmeta
rpart	Rstreams	SASmixed	scatterplot3d
spatial	splines	stepfun	survival5
tree	tripack	wavethresh	xtable

Table 1: List of the precompiled packages for the Macintosh porting

One feature that is still absent in this release is the Tcl/Tk support that will be hopefully released starting from 1.3 version.

One other difference to know for non-Macintosh users is that for each application the System has to reserve a fixed amount of memory in advance. This means that you cannot control dynamically the memory consumption of R during your session. The only thing to do is to reserve a sufficiently large amount of memory for the R application starting from the Finder/Information menu. Then R can use as much memory as it needs in within this limit.

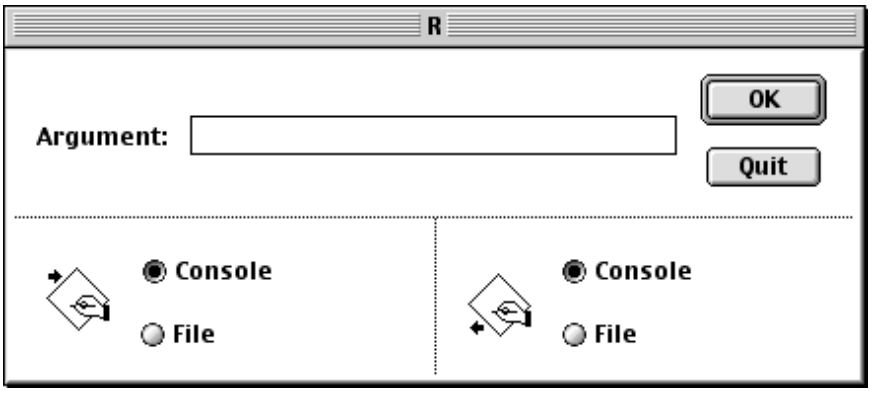

Figure 1: Input-Output redirection under Macintosh

The last thing that is really missing for a standard R user is the interaction with an external editor such as Emacs (this one is currently available for MacOS).

From the developer side, it has to be said that there is not a straight forward procedure to built R or the contributed packages. Proejct files heavily depends on the compiler we choose to build R. This is most due to the fact that a shell environment is not available for Macintosh. This will change in the near future.

1.2 The Carbonized R release

What the adjective "Carbonized" means is known probably only in the Macintosh developers community. Starting from March 24th 2001, Apple will release a completely new system for its machines called MacOS X where the "X" stands both for 10 and for Unix. This new system is based on a Mach-o kernel and it is built on BSD. CarbonLib is a shared library that can be used by old MacOS applications to run natively under MacOSX too. Of course this had to be previewed in advance when the candidate application is built. The CarbonLib library contains lots of new APIs that substitute in part the older ones available to developers of MacOS applications. The source code thus needs to be partially rewritten having in mind the new features and requirement of MacOSX. But once this has been done, the application can take advantage of the new system. In particular the new Aqua interface is one of the battle-horses of the brand new MacOSX. The carbonized release of R will be probably the 1.2.3. It is almost finished, the delay is just because the `Macintosh()` device has been partially rewritten.

Version 1.2.3 will also fix some few numerical bugs. One thing to note is that, even if this application will run natively under MacOSX, it still behaves like a non-Unix application. Thus, file redirection is emulated and so on. But a great feature that is available for free is that under MacOSX the user will be no longer concerned with reserving memory for R in advance as it can have as much memory as needed (see Figure 2 for a screen snapshot of Carbon-R and it memory allocation

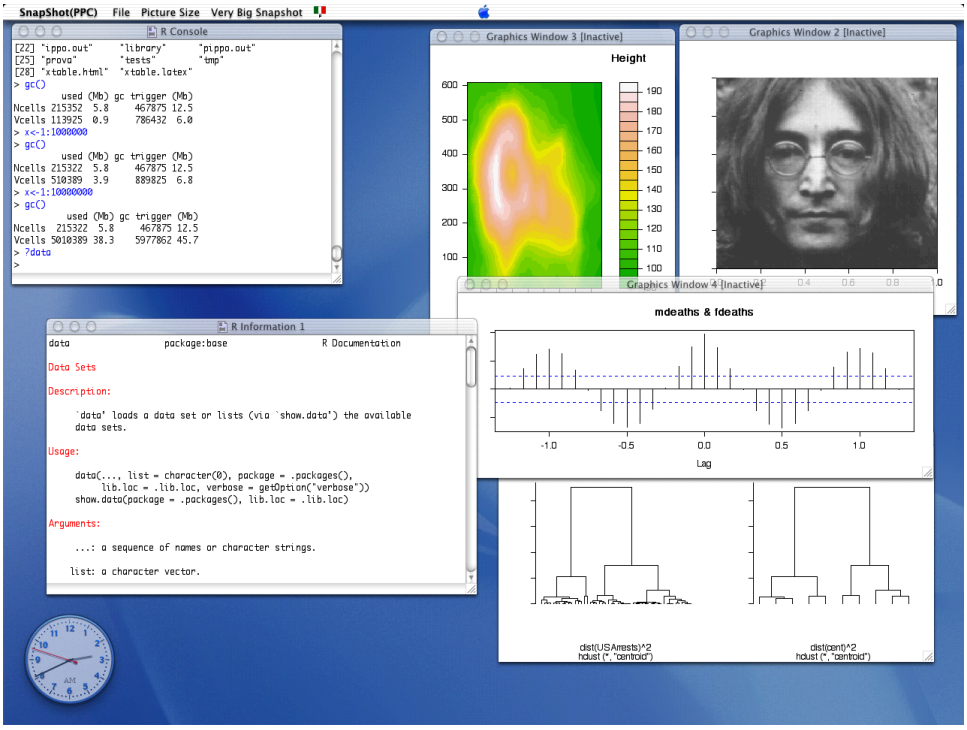

Figure 2: R for Macintosh taking advantages of the new Aqua interface. It can be noticed that we have allocated a 10000000 object without problems. Multiple graphic devices are open containg the result of the `acf` example from the "ts" library, the `demo(image)`, some cluster analysis and `lennon` example from the "wavethresh" library. Please also note the time when the snapshot has been taken!

management).

This new release of Carbon-R will also be scriptable even if not probably yet ready to interact with an external editor.

1.3 R for MacOS X

At this time a native version of R 1.2.2 for MacOSX has been successfully built by Jan de Leeuw (http://www.stat.ucla.edu/~deleeuw). This is based on Darwin 1.3 and X11R6 and will not run on the standard MacOS System. A version based on the native MacOSX window manager will be probably available in time for the release 1.3 of R.

Acknowledgment

Many people with their feedback and developers with their source code support have yield to the current status of R for Macintosh. Among the others I warmly thank Ross Ihaka, Luke Tierney, Brian Ripley, Günter Sawitzki, Martin Maechler, Anestis Antoniadis.