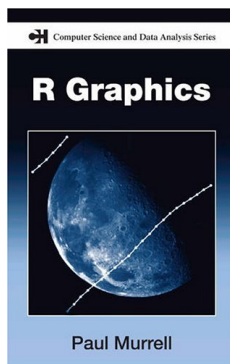


The good, the bad, and the ugly—Review of Paul Murrell’s new book: “R Graphics”

by David Meyer



At the DSC 2001 workshop in Vienna, Ross Ihaka presented an overview on the “old” R graphics system using the title: “The good, the bad, and the ugly”, basically meaning that the traditional graphics system, albeit flexible, was getting old and the interface was cumbersome in several aspects. Think, e.g., of the mysteries of the dozens of `par()` parameters whose lengthy help

page most useRs keep browsing on a regular basis to filter out the appropriate settings. Thanks to Paul Murrell’s new book, the secrets of both traditional graphics and the new, modern `grid` system get unveiled, preventing useRs from writing “ugly” code. Starting from scratch, both architectures are presented and compared in detail, complemented with illustrative examples and summary tables. The text not only covers basic elements such as points, lines, segments, and text, but also discusses the most important high-level plot functions available in base R, as well as some popular extension packages such as `scatterplot3d`. In addition, a clear effort has been made to enable the readers to develop their own graphical methods, in the R spirit of “users becoming developers”.

The first part of the book covers the traditional graphics system. It starts with some simple plots using `plot()` and friends, and proceeds with a detailed

chapter on customizing, covering all low-level and high-level `par()` settings, multiple plots, annotation, and more complex, superposed plots. Even experienced users will discover “new” features. For example, most people will be familiar with line types—but what about the various line join and line ending styles, or the correct handling of fonts?

The second part is devoted to `grid`, the new graphics system developed by the book author himself. Murrell chooses a practical approach and first introduces the `lattice` package, R’s version of Trellis graphics. After giving an overview on all high-level `lattice` functions, the author provides customization examples. Only after this “teaser” is the reader confronted with the basic elements of the `grid` package. The text explains the idea behind concepts like units, graphical context, viewports, and how existing plot elements can easily be integrated in more complex displays. Particular emphasis is given to the possibilities of interactive plot modifications and off-screen computations. Finally, a whole chapter is devoted to the development of new `grid`-based functions and objects that can be reused by others.

Paul Murrell’s book is the first publication entirely devoted to R graphics, written by *the* authoritative expert in the field. It is definitely a must-have for novices and professionals alike, the ultimate guide to the power (and beauty) of R graphics.

David Meyer

Department of Information Systems and Operations, Vienna University of Economics and Business Administration

Augasse 2-6, A-1090 Wien, Austria

david.meyer@wu-wien.ac.at

Changes in R

by the R Core Team

User-visible changes

- In the `grid` package there are new ‘arrow’ arguments to `grid.line.to()`, `grid.lines()`, and `grid.segments()` (`grid.arrows()` has been deprecated).

The new ‘arrow’ arguments have been added BEFORE the ‘name’, ‘gp’ and ‘vp’ arguments so existing code that specifies any of these arguments *by position* (not by name) will fail.

- `all.equal()` is more stringent, see the PR#8191 bug fix below.

- The data frame argument to `transform()` is no longer called ‘x’, but ‘_data’. Since this is an invalid name, it is less likely to clash with names given to transformed variables. (People were getting into trouble with `transform(data, x=y+z)`.)

New features

- `arima.sim()` has a new argument `'start.innov'` for compatibility with S-PLUS. (If not supplied, the output is unchanged from previous versions in R.)
- `arrows()` has been changed to be more similar to `segments()`: for example `col=NA` omits the arrow rather than as previously (undocumented) using `par("col")`.
- `as.list()` now accepts symbols (as given by `as.symbol()` aka `as.name()`).
- `atan2()` now allows one complex and one numeric argument.
- The 'masked' warnings given by `attach()` and `library()` now only warn for functions masking functions or non-functions masking non-functions.
- New function `Axis()`, a generic version of `axis()`, with Date and POSIX[cl]t methods. This is used by most of the standard plotting functions (`boxplot`, `contour`, `coplot`, `filled.contour`, `pairs`, `plot.default`, `rug`, `stripchart`) which will thus label x or y axes appropriately.
- `pbeta()` now uses TOMS708 in all cases and so is more accurate in some cases (e.g. when `lower.tail = FALSE` and when one of the shape parameters is very small).
- `[qr]beta()`, `[qr]f()` and `[qr]t()` now have a non-centrality parameter.
- `[rc]bind()` and some more cases of sub-assignment are implemented for raw matrices. (PR8529 and 8530)
- The number of lines of deparsed calls printed by `browser()` and `traceback()` can be limited by the option `"deparse.max.lines"`. (Wish of PR#8638.)
- New `canCoerce()` utility function in "methods" package.
- `[pq]chisq()` are considerably more accurate for moderate (up to 80) values of `ncp`, and `lower.tail = FALSE` is fully supported in that region. (They are somewhat slower than before.)
- `chol(pivot = TRUE)` now gives a warning if used on a (numerically) non-positive-definite matrix.
- `chooseCRANmirror()` consults the CRAN master (if accessible) to find an up-to-date list of mirrors.
- `cov.wt()` is more efficient for `'cor = TRUE'` and has a new `'method'` argument which allows 'Maximum Likelihood'.
- `do.call()` gains an `'envir'` argument.
- `eigen()` applied to an asymmetric real matrix now uses a tolerance to decide if the result is complex (rather than expecting the imaginary parts of the eigenvalues to be exactly zero).
- New function `embedFonts()` for embedding fonts in PDF or PostScript graphics files.
- `fisher.test()` now uses p-values computed via hypergeometric distributions for all 2 by 2 tables. This might be slightly slower for a few cases, but works much better for tables with some large counts.
There is a new option to simulate the p-value for larger than 2 x 2 tables.
- `for()` now supports raw vectors as the set of indices.
- `getNativeSymbolInfo()` is vectorized for the `'name'` argument. It returns a named list of `NativeSymbolInfo` objects, but is backward compatible by default when called with a character vector of length 1, returning the `NativeSymbolInfo` object.
- `help.search()` no longer attempts to handle packages installed prior to R 2.0.0, and reports the current path to the package (rather than where it was originally installed: this information is not shown by the `print()` method).
- Added "hexmode" to parallel "octmode".
- `install.packages()` now does tilde expansion on file paths supplied as `'pkgs'`.
- `install.packages()` has additional arguments `'configure.args'` and `'clean'` which allow the caller to provide additional arguments to the underlying R CMD INSTALL shell command when installing source packages on a Unix-alike.
- `is.loaded()` has a new argument `'type'` to confine the search to symbols for `.C`, `.Fortran`, `.Call` or `.External`: by default it looks for a symbol which will match any of them. It is now internal and not primitive, so argument matching works in the usual way.
- The symmetry test for matrices used in `eigen()` has been "exported" as the `'matrix'` method of a new S3-generic `isSymmetric()`.
- `.leap.seconds` and the internal adjustment code now know about the 23rd leap second on 2005-12-31: the internal code uses a run-time test to see if the OS does.

- The `'col'` argument of `legend()` now defaults to `par("col")` (which defaults to "black", the previous default), so that the lines/symbols are shown in the legend in the colour that is used on the plot.
- `log2()` and `log10()` call C functions of the same name if available, and will then be more likely to be precise to machine accuracy.
- `new.packages()` gains a `...` argument to pass e.g. `'destdir'` to `install.packages()`. (Wish of PR#8239.)
- `nls()` now supports `'weights'`.
- The vector passed as the first argument of the `'fn'` and `'gr'` arguments of `optim()` has the names (if any) given to argument `'par'`.
- `options(expressions)` is temporarily increased by 500 during error-handling. This enables e.g. `traceback()` to work when the error is reaching the limit on the nesting of expressions.
- `page()` accepts general R objects, not just names (and previously undocumented) character strings. This allows the object to be specified as a call, for example. More options are allowed in its `'...'` argument.
- `pairs()` allows a wider class of inputs, including data frames with date and date-time columns.
- `par()` and the in-line use of graphical parameters produce more informative error messages, distinguishing between non-existent pars and inappropriate use of valid pars.
Graphical parameters `'family'`, `'lend'`, `'ljoin'` and `'lmitre'` can now be set in-line.
There is no longer a warning if non-settable pars are used in-line, but there is an appropriate warning if unknown pars are passed.
The length limit for the `'family'` parameter has been increased to 200 bytes, to allow for the names of some CID-keyed fonts in multi-byte locales.
- The `pdf()` device now allows `'family'` to be specified in the same generality as `postscript()`.
- The `pdf()` device writes `/FontDescriptor` entries for all fonts except the base 14, and does not write font entries for unused fonts.
- `Plotmath` allows `'vartheta'`, `'varphi'` and `'var-sigma'` (or `'stigma'`) as synonyms for `'theta1'`, `'phi1'` and `'sigma1'`, and the help page has a note for TeX users.
- `plot.xy()` now takes its default arguments from the corresponding `par()` settings, so `points(type="l")` and `lines(type="p")` behave in the same way (and more obviously, also for `type="b"`).
- `poly()` has a new argument `'raw'`, mainly for pedagogical purposes.
- The class "POSIXlt" now supports fractional seconds (as "POSIXct" has always done). The printing of fractional seconds is controlled by the new option "digits.secs", and by default is off.
- `postscript()` supports `family = "Computer-ModernItalic"` for Computer Modern with italic (rather than slanted) faces.
- The `postscript()/pdf()` font metrics for the 14 standard fonts (only, not the rest of the common 35) have been updated to versions from late 1999 which cover more glyphs. There are also a few differences in the metrics and hence the output might be slightly different in some cases.
- The way families can be specified for `postscript()` and `pdf()` has been expanded to include CID-keyed fonts, with new functions `Type1Font()` and `CIDFont()` to set up such fonts families.
- `prettyNum()` has new arguments `'preserve.width'` and `'zero.print'`. When the former is not "none", as in calls from `format()` and `formatC()`, the resulting strings are kept at the desired width when possible even after adding of `'big.mark'` or `'small.mark'`.
- `proc.time()` and `system.time()` now record times to 1 ms accuracy where available (most Unix-like systems).
- The initialization methods for the `quasi()` family have been changed to depend on the variance function, and in particular to work better for the "mu(1-mu)" variance function. (PR#8486)
- `read.table()` gains a `'flush'` argument passed to `scan()`.
- `require()` now takes a `'lib.loc'` argument.
- The second argument `'size'` to `sample()` is required to have length 1, so that errors when supplying arguments are more easily detected.
- The default is now `compress = !ascii` in `save()` (but not `save.image()`).

- `scan()` and `write.table()` now have some interruptibility, which may be useful when processing very large files.
 - A new heuristic test, `seemsS4object()` is supplied, along with a similar C-level test, `R_seemsS4object(object)`. The test detects probable S4 objects by their class's attribute. See the help page.
 - S3 classes can now be made non-virtual S4 classes by supplying a prototype object in the arguments to `setOldClass()`.
 - `splinefun()` returns a function that now also has a 'deriv' argument and can provide up to the 3rd derivative of the interpolating spline, thanks to Berwin Turlach.
 - `stopifnot(A)` now gives a better error message when A has NAs, and uses "not all TRUE" when A has length ≥ 2 .
 - `str()`'s default method has a new argument 'strict.width' which can be used to produce strict 'width' conforming output. A new options(`str = list(strict.width = *)`) setting allows one to control this for a whole session.
 - `summary.nls()` has a new argument 'correlation' that defaults to FALSE (like `summary.lm`).
 - `Sys.sleep()` has sub-millisecond resolution on Unix-alikes with `gettimeofday()`.
 - `Sys.time()` now has sub-millisecond accuracy on systems supporting the POSIX call `gettimeofday`, and clock-tick accuracy on Windows.
 - The new function `timestamp()` adds a time stamp to the saved command history on consoles which support it.
 - New function `tcrossprod()` for efficiently computing $x \%*\% t(x)$ and $x \%*\% t(y)$.
 - The suffix used by `tempfile()` is now in hex on all platforms and guaranteed to be at least 6 hex digits (usually 8).
 - `trace()` now works more consistently and more like its documentation, in particular the assertions about old tracing being removed for new. For debugging purposes (of R) a mechanism for debugging the trace computations themselves was added. See `trace.R`.
 - The implementation of `trace()` has been made more general by calling a function to do the trace interaction, and `recover()` now detects trace calls to trim the irrelevant code underneath.
 - `unserialize()` can now also read a byte stream from a raw vector.
 - The `useDynLib()` directive in the NAMESPACE file now accepts the names of the native/foreign symbols that are to be resolved in the DLL for use in `.C/.Call/.Fortran/.External` calls. These can be used as regular R variables instead of the (routine name, PACKAGE) pairs currently recommended. Alternative names can be given for the R variables mapping to these symbols. The native routine registration information can also be used directly via `useDynLib(name, .registration = TRUE)`. See the 'Writing R Extensions' manual for more details.
- `checkFF()` (package 'tools') has been updated accordingly.
- `validObject()` has an option `complete=TRUE` that recursively checks the objects in the slots. Not used when `new(...)` checks validity.
 - New `Vectorize()` function, a wrapper for `mapply()`.
 - `write.ftable()` has gained an argument 'append = FALSE' (thanks to Stephen Weigand).
 - On Unix-alikes, `X11()` now has arguments to request the initial position of the window, and 'gamma' defaults to the value of `getOption("gamma")`. These changes are consistent with the `windows()` device.
 - `X11()` and the Unix-alike data entry window can have properties (including geometry) set by X resources: see their help files.
 - `xy.coords()` & `xyz.coords()` now have NULL defaults for their 'y' or 'y' and 'z' arguments. This is more consistent with their earlier documentation, and may be convenient for using them.
 - Non-syntactic names of list elements are now printed quoted by backticks rather than double quotes.
 - There is some basic checking for imminent C stack overflow (when the evaluation depth and the user interrupts are checked). On systems with suitable OS support (not Windows), segfaults from C stack overflow are caught and treated as an R error.
- New function `Cstack_info()` reports on stack size and usage.
- `options(expressions)` reverts to the default of 5000 now that stack checking is in place.

- Package tcltk does not try to initialize Tk on Unix-alikes unless a DISPLAY variable is present. This allows packages dependent on tcltk to be installed without access to an X server.
- The code used to guess timezone offsets where not supplied by the OS uses a different algorithm that is more likely to guess the summer-time transitions correctly.
- Package tools contains translation tables 'Adobe_glyphs' and 'charset_to_Unicode'.
- Changed the environment tree to be rooted in an empty environment, available as emptyenv(). baseenv() has been modified to return an environment with emptyenv() as parent, rather than NULL.
- gettext has been updated to 0.14.5.
- PCRE has been updated to version 6.4.
- The method \$.DLLInfo resolves the specified symbol in the DLL, returning a NativeSymbolInfo object. Use [[to access the actual values in the DLLInfo object.
- On systems with either vasprintf or both va_copy and a vsnprintf which reports the size of buffer required, connections such as gzfile() and bzfile() can now write arbitrarily long lines, not just 100000 chars.
- The R session temporary directory is now set in C code using the same algorithm whether or not the shell front-end is used and on all platforms. This looks at environment variables TMPDIR, TMP and TEMP in turn, and checks if they point to a writable directory.
- Some of the classical tests put unnecessary restrictions on the LHS in the formula interface (e.g., t.test(x+y ~ g) was not allowed).
- On suitably equipped Unix-alike systems, segfaults, illegal operations and bus errors are caught and there is a simple error-handler which gives the user some choice as to what to do in interactive use. [Experimental.]
On Windows access violations and illegal instructions are caught with a simple error handler. [Experimental.]
- Tracebacks now include calls such as .C/.Fortran/.Call, which will help if errors occur in R code evaluated by compiled code and in tracebacks presented by the segfault etc handlers.

- Treatment of signature objects and method definition objects has been modified to give cleaner printing and more consistency in the treatment of signatures. A sometimes useful utility, methodSignatureMatrix(), is now exported.
- R refrains from printing a final EOL upon exiting the main loop if the quiet flag is on and if the save action is known (e.g. this is true for -slave).

Deprecated & defunct

- The deprecated and undocumented use of atan() with two arguments has been removed: instead use atan2().
- write.table0() is defunct in favour of write.table().
- format.char() is defunct in favour of format.default().
- Support for the long-deprecated (and no longer documented) arguments -min-vsize -min-nsz -max-vsize -max-nsz -vsize -nsz of R CMD BATCH has been removed.
- The 'debian' subdirectory has been removed from the sources.
- The 'vfont' argument of axis() and mtext() has been removed: use par(family=) instead.
- The unused graphical parameter "type" has been removed: it invited confusion with the 'type' argument to default methods of plot(), points() and lines().
- nlsMethod() and profiler() are no longer exported from the stats namespace (and nlsMethod.plinear() is no longer registered as a method, as nlsMethod() was not generic).
- The re-named tcltk functions tkcmd, tkfile.tail, tkfile.dir, tkopen, tkclose, tkputs, and tkread are now formally deprecated.
- Argument 'col' of bxp() is now formally deprecated.
- Use of NULL as an environment is deprecated and gives a warning.
- postscriptFont() is deprecated in favour of Type1Font() (which is just a change of name).
- La.chol() and La.chol2inv() are deprecated (they have since R 1.7.0 been the same as the default options of chol() and chol2inv()).
- La.svd(method = "dgesvd") is deprecated.

- The use of `install.R` and `R_PROFILE.R` files in packages is deprecated: use the `DESCRIPTION` file instead to arrange to save an image or to load dependent packages.

The following command-line options to `INSTALL` are deprecated (use the fields in the `DESCRIPTION` file instead): `-s` `-save` `-no-save` `-lazy` `-no-lazy` `-lazy-data` `-no-lazy-data`

- Graphical parameter `'tmag'` (which is long unused) is deprecated.

Internationalization

A set of patches supplied by Ei-ji Nakama has been incorporated.

- New postscript encodings for CP1253, CP1257 and Greek (ISO 8859-7).
- Support for East Asian CID-keyed fonts in `pdf()` and `postscript()`. Although these usually contain Latin characters no accurate AFMs are available and so CID-keyed fonts are intended only for use with CJK characters.
- Wide-character width functions `wc[s]width` are provided that overcome problems found with OS-supplied ones (and those previously used by R on Windows). This means that double-width CJK characters are now supported on all platforms. It seems that the width of some characters (and not just CJK characters) depends on which CJK locale's fonts are in use and also on the OS.

Revised wide-character classification functions are provided for use on Windows, AIX and MacOS X to replace deficient OS-supplied ones.

- There is support for MBCS charsets in the `pictex()` graphics device, and rotated (by 90 degrees) text may work better.
- The `\u` (and `\U` except on Windows) notation for characters which is supported by the parser in all MBCS charsets is now always interpreted as a Unicode point, even on platforms which do not encode `wchar_t` in Unicode. These are now a syntax error in single-byte locales.
- The default encoding for `postscript()` and `pdf()` is chosen to be suitable for the current locale, if that is a single-byte locale which is supported. This covers European (including Greek) and Cyrillic languages.

In UTF-8 locales, a suitable single-byte encoding is chosen for `postscript()` and `pdf()`, and text translated to it.

- `xfig()` gains an `'encoding'` argument.

- There are some message translations into Spanish.

Installation changes

- The encoding files for `pdf()`/`postscript()` have been moved to directory `'enc'` in package `'grDevices'`.
- Support for MBCS is only enabled if `iconv` is found and it supports enough conversions. (`libiconv` does.)
- In an MBCS locale, `make check` now translates the graphics examples from Latin-1. This ensures that they will work correctly in UTF-8: it is possible that in other MBCS locales they will now fail (rather than work completely incorrectly).
- There is a new test, `'test-Docs'`, which as part of `'make check-devel'` tests the code in the documentation. Currently it runs `doc/manual/R-{exts,intro}.R` and the compiled code in `R-exts.c`.
- The workaround to allow an external LAPACK-containing BLAS such as `libsunperf` to be used with the internal LAPACK has been removed. If you have such a library you may now need to use `-with-lapack`. It is no longer possible to use some older versions of `libsunperf`, e.g. Forte 7 on 64-bit builds.
- A substitute for `mkdtemp` is provided, so it is now always used for `R_TempDir`.
- Most of the functions checked for by `'configure'` also have declarations checked for in the appropriate header.
- The top-level documentation files `AUTHORS`, `COPYING.LIB`, `COPYRIGHTS`, `FAQ`, `RESOURCES`, `THANKS` have been moved to `doc`, and `COPYING` and `NEWS` are installed there. The file `Y2K` has been removed from the distribution.
- The extension `.lo` is no longer used in building R (only in the optional build of `libRmath.so`): this allows a considerable simplification of the Makefiles.
- Direct support for `f2c` has been removed: it can still be used via a script which makes it look like a Fortran compiler. (`src/scripts/f77_f2c` is an example of such a script.)
- There is a new flag `SAFE_FFLAGS` which is used for the compilation of `dlaamc.f`. It is set by `configure` for known problem cases (recent `g77` and `gfortran`), but can be overridden by the user.

- The standard autoconf macros for large-file support are now used, and these are enabled unless `-disable-largefile` is specified. This replaces `-enable-linux-lfs` (and is now selected by default).
- Visibility attributes are used where supported (gcc4/gfortran on some platforms, also gcc3/g77 on FC3 and partially elsewhere). The main benefit should be faster loading (and perhaps better optimized code) in some of the dynamic shared objects (e.g. `libR.so` and `stats.so`).
- The `*PICFLAGS` are taken to be `-fpic` rather than `-fPIC` where possible. This will make no difference on most platforms: `-fPIC` is needed on Sparc (and still used there), but `-fpic` should give slightly better performance on PowerPC (although `-fPIC` is used on PPC64 as it is needed to build `libR.so` there).
- More use is made of inlining for small utility functions such as `isReal`. Because this can only be done portably with C99 constructs (and we know of no actual implementation), this is only done for the GNU C compiler.
- There is an experimental feature to allow shared installations of sub-architectures. See the R-admin manual.
- All platforms now use R's internal implementation of `strptime`, which allows fractional seconds. (The major platforms were already using it.)
- The `dlcompat` work-around for old Mac OS X systems (≤ 10.2) has been removed. External `dlcompat` must be installed if needed.

Utilities

- R CMD check now uses an install log by default.
- R CMD check works for packages whose package name is different from the directory name in which it is located.
- R CMD INSTALL now uses more randomness in the temporary directory name even on systems without `mktemp -d`.
- R CMD `f77` has been removed now that `f2c` is no longer supported.
- The version string shown in the startup message and by `"R -version"`, and that stored in variable `R.version.string`, are now in exactly the same format.
- The base name of a help file needs to be valid as part of a file:// URL, so R CMD check now checks that the names are ASCII and do not contain `%`.
- R CMD check now warns about unknown sections in Rd files, and invalid names for help, demo and R files, as well as unlikely file names in the 'src' directory. The latter is controlled by option `-check-subdirs` and by default is done if checking a tarball without a configure script. R CMD build excludes invalid files in the 'man', 'R' and 'demo' subdirectories.
- `\usepackage[noae]{Sweave}` in the header of an Sweave file suppresses auto-usage of the ae package ("almost European" fonts) and T1 input encoding.

Documentation

- Rd format now allows `\var{}` markup inside `\code{}` and `\examples{}`.
- Markup such as `--`, `---`, `<` and `>` is handled better when converting .Rd files to [C]HTML.
- There is new markup `\link[=dest]{name}` to generate a link to topic 'dest' which is shown as 'name', and `\linkS4class{abc}` which expands to `\link[=abc-class]{abc}`, for cross-referencing the recommended form of documentation for S4 classes.

Package Installation

- There is now some support for Fortran 90/95 code in packages: see 'Writing R Extensions'.
- Installation of man sources and demos is now done by R code. The restrictions on the names of help files, R files and of demos are now enforced (see 'Writing R Extensions').
- Packages that contain compiled code can now have more than one dot in their name even on Windows.
- The `Meta/hsearch.rds` database saved now contains `LibPath=""`. This information is now always recreated when `help.search()` is run, but the field is retained for back-compatibility.
- `update.packages()` now has a `'...'` argument to be passed to `install.packages()`, including the formerly separate arguments `'destdir'` and `'installWithVers'`.
- Make macros `AR` and `RANLIB` are now declared in `etc/Makeconf` for use by packages that wish to make static libraries.

C-level facilities

- `qgamma` and `rgamma` in `Rmath.h` now check for non-positive arguments.
- The BLAS that ships with R now contains the complete set of double-complex BLAS routines, rather than just those used in R.
`<R_ext/BLAS.h>` has been corrected to add the missing double-precision BLAS functions `drotmg` and `drotm`, and to exclude `lsame` (which is a Lapack auxiliary function and is now declared in `<R_ext/Lapack.h>`). It also includes the double complex routines added for this release of R provided Fortran double-complex is usable on the platform.
- `<R_ext/BLAS.h>` and `<R_ext/Lapack.h>` now declare all the entry points as 'extern'.
- The flag `SAFE_FFLAGS` is made available to packages via `etc/Makeconf` and R CMD `config`. It can be used where optimization needs to be defeated, e.g. in LAPACK setup.
- `getNativeSymbolInfo` has a `withRegistrationInfo` argument which causes the address field to be a reference to the registration information if it is available for that symbol. If the registration information is not available, the address is a reference to the native symbol. The default is `FALSE` which is backward compatible, returning just the address of the symbol and ignoring registration information.
- `errorcall` and `warningcall` are now declared in `<Rinternals.h>` (they might be needed in front-ends).
- `R_FlushConsole` and `R_ProcessEvents` are now declared in `<R.h>`.
- The `R_Sock*` functions supporting socket connections are no longer declared in `R-ftp-http.h` as they are not loaded into R itself, and are now hidden in the module's DLL on suitable systems.

Bug fixes

- Quoted arguments to the R script after `-args` are now passed quoted to the R executable and so will be shown as expected by `commandArgs()`. (They were previously split at whitespace even inside quotes on Unix-alikes but not on Windows.)
- `axis()` now supports pars `'xaxp'/'yaxp'` as inline arguments.

- `sort()` now does not return inappropriate attributes such as "dim" and "tsp": it only returns names.
`sort(x, partial=)` no longer returns unsorted names, and drops names (since it is supplied for efficiency).
- Use of non-central F in `pf()` gives accurate values for larger `n`.
- R CMD `build -binary` does a better job of cleaning up after failure to re-make vignettes.
- `reg-test-1.R` tested `system(intern=TRUE)` which depends on `popen` and so is not supported on all platforms.
- Changed apparent mis-spelling of "Gibraltar" in dataset 'eurodist'.
- `sysconf()` is now used to find the number of clock ticks/second: under some circumstances `glibc` reported `CLK_TCK = 60` when the true value was 100.
- `identical()` was not allowing for embedded nuls in character strings. (NB: the comparison operators including `==` do not, and never will.)
- The `profile()` and `profiler()` methods for "nls" objects now support `algorithm = "plinear"` and `algorithm = "port"`.
- The signal handlers for signals `USR1` and `USR2` where not restored if the signal arrived when interrupts were suspended.
- Certain combinations of S4 inheritance could cause inherited methods to override some directly specified methods.
- Some cases of named signatures in calls to `setMethod()` caused errors.
- `all.equal()` is now more consistent and "picky" about mismatching attributes, in particular `names()`; this is a part of the propositions by Andy Piskorkski (PR#8191).
- `load()` when applied to a connection leaves it open/not as it found it, and checks explicitly for having a binary readable connection.
- The p-values given by `stat.anova()` (called from several `anova()` methods) are now NA (rather than spurious) if non-nested models give rise to changes in deviance with a different sign from changes in degrees of freedom.
- Built-ins were reported as the relevant call in `C-level error()`s iff R profiling was in progress. Now they are never reported.
- Too-long signatures (with no names) were not being caught in `setMethod()`.

- Slot names in `prototype()` are being more thoroughly checked.
- `signif()` is more likely to follow the 'round to even' rule for exactly representable numbers, e.g. `signif(0.25, 1)`. (Related to PR#8452.)
- `nls()` now works correctly with some low-dimensional fits, e.g. with one or zero non-linear parameters.
- `glm()` could give an inappropriate error message if all possible coefficients were invalid (e.g. a log-linear binomial model with no intercept and a not all positive predictor).
- `solve()` gives clearer error messages for some incorrect usages. (PR#8494 and similar)
- The `gaussian()` family was missing the 'valideta' component (which could be needed for the "inverse" link function).
The starting values supplied by the `gaussian` family could be invalid for the "log" and "inverse" link functions. This is now reported.
- `data.matrix()` did not work correctly on zero-row data frames. (PR#8496 and other problems.)
- The DSC comments in the files from `postscript(onefile=FALSE)` now label all files as having page 1 of 1, as some other software seems to expect that.
- The axis labels chosen for logarithmic axis are now less likely to be linear and inappropriate (when the range is more than 10 and less than 100). (PR#1235)
- Staircase lines (types "s" and "S") are now drawn continuously rather than a point at a time and so line types, `mitring` and so on work. (PR#2630)
- Calling `par(mfg)` before doing any plotting resulted in `NewPage` never being called on the device, which in turn resulted in incorrect output for `postscript()` and `pdf()` devices. (Reported by Marc Schwartz in discussion of the non-bug PR#7820.)
- `terms.formula` needed to add parentheses to formulae with terms containing '|'. (PR#8462)
- `pbirthday()` and `qbirthday()` now also work for very improbable events — those you are typically *not* interested in.
- Only source help files starting with an upper- or lower-case letter or digit and extension `.Rd` or `.rd` are documented to be processed. This is more liberal in that starting with a digit is now also allowed, but the rule is now enforced.
- `nls(algorithm="port")` was always taking positive numeric differences and so could exceed the upper bounds.
- `methods:::asEnvironmentPackage()` was not allowing for versioned installs.
- `.find.package()` now reports which package(s) it cannot find in case it stops with an error.
- The standard Unix-alike version of `file.show()` gives an informative message if it cannot open a file rather than the (possibly incorrect) 'NO FILE'.
- `window()` did not allow non-overlapping ranges with `extend = TRUE`. (PR#8545)
- `pbinom(size = 0)` now returns correct values (not NaN). (PR#8560)
- `[dp]binom(x, *)` for $x < 0$ now always returns 0 (PR#8700). Analogous change in `pgeom()`, `pnbinom()` and `ppois()`.
- `[dqpr]geom()` and `[dqpr]nbinom()` now all consistently accept `prob = 1` but not `prob = 0`. `qgeom(prob=1)` now gives the correct values (not -1).
- `INSTALL` on Unix-alikes was not loading dependent packages when preparing for lazy-loading.
- `qcauchy(1)` now gives `+Inf` instead of just a very large number.
- `df(0, f1, *)` now properly returns `Inf`, `1`, or `0` for $f1 <, =, \text{ or } > 2$.
- `qbinom()`, `qnbinom()` and `qpois()` now use a better search and normally reach the answer very quickly when it is large (instead of being slow or infinite-looping).
- `pt(x, df)` lost accuracy in the far tails (when $|x| > 1e154$) for small `df` (like `df = 0.001` for which such extremes are not unlikely).
- `dbeta(x, a, b)` underflowed internally and incorrectly gave 0 for very small `x` and `a`.
- None of the warnings about convergence failures or loss of precision in `nmath` (distribution and special functions) were being reported to the R user.
- `dnt` was missing from standalone `nmath` (under Unix-alikes).
- `split()` now accepts factors with numeric (but not storage mode integer) codes.

- The utilities such as 'check' now report active version numbers again, as SVN 'last changed revision' numbers.
- `addmargins()` did not accept a name for 'FUN', only an expression.
- '+' for POSIXt objects now takes the tzone from whichever object has it, so `date+x` is the same as `x+date` if `x` is numeric.
- `mean.default()` and `var()` compute means with an additional pass and so are often more accurate, e.g. the variance of a constant vector is (almost) always zero and the mean of such a vector will be equal to the constant value to machine precision. (PR#1228)
`sum()`, `prod()`, `mean()`, `rowSums()` and friends use a long double accumulator where available and so may be more accurate. (This is particularly helpful on systems such as Sparc and AMD64 where long double gives considerably greater exponent range and precision than double.)
- `read.dcf()` now gives a warning on malformed lines.
- `add1.[g]lm` now try harder to use the environment of the formula in the original fit to look for objects such as the 'data' and 'subset' arguments.
- `gaussian()$aic` was inconsistent with e.g. the `lm` results from `AIC()` and `extractAIC()` for weighted fits: it treated the weights as case weights and not variance factors.
- `system()` on Unix-alikes ignored non-logical values of 'intern' and treated 'intern = NA' as true.
- `as.table()` now produces non-NA rownames when converting a matrix of more than 26 rows. (PR#8652)
- Partial sorting used an algorithm that was intended only for a few values of 'partial' and so could be far slower than a full sort. It now switches to a barebones full sort for more than 10 values of 'partial' and uses a more efficient recursive implementation for 2...10.
- `summary.glm()` returned an estimate of dispersion of Inf for a gaussian `glm` with zero residual degrees of freedom and then treated that as a known value. It now uses the estimate `NaN`, which is consistent with `summary.lm()`.
- `Sys.sleep()` on Unix-alikes was restricted to about 2147 seconds and otherwise might never have returned. (PR#8678)
- `is(obj, CI)` could wrongly report TRUE when `CI` was a classUnion and multiple inheritance was involved.
- `confint[.lm / .default]` used label "100 %" for `level = 0.999`
- Empty entries (i.e., extraneous ",") in NAMESPACE files now give a better error message early at parsing time instead of a less comprehensible one later at load time.
- `all.equal(n1, n2)` could erroneously return NA when `n1, n2` contained large integers.
- `anova.mlm()` didn't handle multi-df effects properly in the single-model case (PR#8679)
- `anova.mlm()` had its colnames mangled by `data.frame()` (needed `check.names=FALSE`).
- `summary.glm()` gave an NA estimate of dispersion for fits with zero weights. (PR#8720)
- `qhyper()` had too small a tolerance for right-continuity on some platforms so was not always an inverse to `phyper()`.
- `rownames<-data.frame()` tested the `length(s)` of the replacement value(s) before coercion, which can change the length (e.g. for class "POSIXt"). Ditto `dimnames<-data.frame()`.
- `max()` and `min()` ignored the largest/smallest representable integer, as well as Inf/-Inf. (PR#8731)
- `write.table()` assumed factors had integer codes: it now allows malformed factors with numeric codes (and otherwise throws an error).
- Worked around a Solaris restriction which meant that `Sys.sleep()` was only effective for times of up to one second.
- `sink(, split=TRUE)` now works correctly, but is allowed only on platforms that support `va_copy` or `__va_copy`. (PR#8716)
- `factanal()`, `prcomp()` and `princomp()` now only check that columns in the model frame that will be used are numeric (they previously also checked columns which were part of negative terms in the formula).
- Misuse of \$ in `apply` could corrupt memory. (PR#8718)
- `apply()` could fail if the function returned NULL (e.g. if there was a single row).
- `registerS3method()` failed due to a typo. (It was almost never used.)

- Registering an S3 method for an S3 generic in another package that was converted to an S4 generic in the same package as the S3 method, registered the method in the wrong place.
- `Recall()` used lookup for the function in use and so could fail if that was an S3 method not on the search path.
- `Rdconv -t Ssgm` failed if it encountered `\link[opt]{arg}`.
- `uniroot()` did not give a warning (as documented) if it failed to converge in 'maxiter' steps. (PR#8751)
- `eapply` (and `as.list.environment`) did not work for the base environment/namespace. (PR#8761)
- Added protection in configure against systems for which using `xmkmf` fails to report a C or C++

compiler.

- `expand.grid()` was constructing a data frame 'by hand' and so setting integer row.names (which are documented to be character). It now sets character row names, and `row.names.data.frame()` coerces to character.
- `qbeta()` used `==` on volatile doubles for its convergence test, which failed with gcc 3.3.x on ix86 Linux. We now use a less fragile test (and lose a negligible amount of accuracy).
- `ls.str()` was missing `inherits=FALSE`, and so could have reported on an object of the same name but a different mode in the enclosure of the given environment.
- `logLik.nls` assumed that σ^2 had been estimated, but did not count this in the 'df' attribute.

Changes on CRAN

by Kurt Hornik

New contributed packages

BayesTree Software accompanying the paper "Bayesian Additive Regression Trees" by Chipman, George and McCulloch (2005), see <http://gsbwww.uchicago.edu/fac/robert.mcculloch/research>. By Hugh Chipman and Robert McCulloch.

BayesValidate Implements the software validation method described in the paper "Validation of Software for Bayesian Models using Posterior Quantiles" (Cook, Gelman, and Rubin, 2005). It inputs a function to perform Bayesian inference as well as functions to generate data from the Bayesian model being fit, and repeatedly generates and analyzes data to check that the Bayesian inference program works properly. By Samantha Cook.

CTFS The CTFS Large Plot Forest Dynamics Analyses. By Pamela Hall.

CVThresh Carries out a level-dependent cross-validation method for the selection of a thresholding value in wavelet shrinkage. This procedure is implemented by coupling a conventional cross validation with an imputation method due to a limitation of data length, a power of 2. It can be easily applied to classical leave-one-out and k -fold cross validation. Since

the procedure is computationally fast, a level-dependent cross validation can be performed for wavelet shrinkage of various data such as a data with correlated errors. By Donghoh Kim and Hee-Seok Oh.

DescribeDisplay Produce publication quality graphics from output of GGobi's describe display plug-in. By Hadley Wickham, Di Cook, and Andreas Buja.

FLCore Contains the core classes and methods for FLR, a framework for fisheries modeling and management strategy simulation in R. Developed by a team of fisheries scientists in various countries. More information can be found at <http://flr-project.org/>, including a development mailing list. By the FLR Team and various contributors. Initial design by Laurence T. Kell & Philippe Grosjean.

FactoMineR Exploratory data analysis. By François Husson, Sébastien Lê, and Jérémy Mazet.

FortranCallsR Teaches how to implement Fortran Code calling R functions. By Diethelm Wuertz.

FracSim Perform simulation of one- and two-dimensional fractional and multifractional Lévy motions. By S. Déjean and S. Cohen.

FunCluster Performs a functional analysis of microarray expression data based on Gene Ontology & KEGG functional annotations. From expression data and functional annotations