This specifies the targets undoc, checkDocFiles, and checkDocStyle, which all depend on any files in the man directory, as well as any code files R/*.R. The output from the R sessions that runs *undoc()* and *checkDocFiles()* print errors and warnings, but these do not automatically produce a shell error signal as a flag that *make* recognizes. It is possible to do this using *R* code that determines if the result should indicate an error, and sets *q(status=1)* but that is not done in this example. Instead, a test on a grep of the output is used to determine the shell error status. (This may change in the future.) If the signal does not indicate a failure (exit 1) then the output is moved to the *FLAGS* directory to indicate that the target has completed successfully.

Summary

There are trade-offs in the way R code is organized into packages. If all code is in one package then there are no package inter-dependencies, but everything must be tested after any change. Faster computers make it possible to consider this, and the *make*/QC system described here would be extra overhead and of limited value in that situation. However, more documentation and examples, along with more extensive test suites, take longer to run, and so encourage a finer breakdown into packages. In addition to this, there are two complementary reasons for organizing functions into packages. One is to limit dependencies, as much as reasonably possible, between groups of functions that are not closely related and may not often be used together. The second is to group together "kernel" functions which are tools used by several other packages. The dependencies among packages must be carefully mapped out, which forces one to think carefully about what is kernel code and what is not. These reasons for organizing code into packages may be even more important in a situation where multiple programmers or users are maintaining packages.

It is important to see that the savings in this *make*/QC system come from a few different aspects. The first is that packages of kernel code used by other packages tend to be more stable and less frequently changed than the packages that use them. If kernel packages are not changed, they do not need to be re-made. The second aspect is that dependencies among packages are in the code, not in the documentation. Thus documentaion changes imply only that the documentation for that particular package needs to be checked. The aspect that results in the most important savings, however, is that the need for many documentation changes are flagged immediately, while you still remember what that marvelous change in the code really did.

Acknowledgments

I am grateful to Kurt Hornik for many helpful explanations and comments.

Paul Gilbert,
Department of Monetary and Financial Analysis,
Bank of Canada,
234 Wellington St.,
Ottawa, Canada, K1A 0G9
pgilbert@bank-banque-canada.ca

Changes in R

by the R Core Team

User-visible changes in 2.0.0

- The stub packages from 1.9.x have been removed: the `library()` function selects the new home for their code.

- 'Lazy loading' of R code has been implemented, and is used for the standard and recommended packages by default. Rather than keep R objects in memory, they are kept in a database on disc and only loaded on first use. This accelerates startup (down to 40% of the time for 1.9.x) and reduces memory usage – the latter is probably unimportant of itself, but reduces commensurately the time spent in garbage collection.

  Packages are by default installed using lazy loading if they have more than 25Kb of R code and did not use a saved image. This can be overridden by `INSTALL --[no-]lazy` or via a field in the `DESCRIPTION` file. Note that as with `--save`, any other packages which are required must be already installed.

  As the lazy-loading databases will be consulted often, R will be slower if run from a slow network-mounted disc.

- All the datasets formerly in packages 'base' and 'stats' have been moved to a new package 'datasets'. `data()` does the appropriate substitution, with a warning. However, calls to `data()` are not normally needed as the data objects are visible in the 'datasets' package.

  Packages can be installed to make their data objects visible via R CMD INSTALL --lazy-data or via a field in the DESCRIPTION file.

- Package 'graphics' has been split into 'grDevices' (the graphics devices shared between base and grid graphics) and 'graphics' (base graphics). Each of the 'graphics' and 'grid' packages load 'grDevices' when they are attached. Note that ps.options() has been moved to grDevices and user hooks may need to be updated.

- The semantics of data() have changed (and were incorrectly documented in recent releases) and the function has been moved to package 'utils'. Please read the help page carefully if you use the 'package' or 'lib.loc' arguments.

  data() now lists datasets, and not just names which data() accepts.

- Dataset 'phones' has been renamed to 'WorldPhones'.

- Datasets 'sunspot.month' and 'sunspot.year' are available separately but not via data(sunspot) (which was used by package lattice to retrieve a dataset 'sunspot').

- Packages must have been re-installed for this version, and library() will enforce this.

- Package names must now be given exactly in library() and require(), regardless of whether the underlying file system is case-sensitive or not. So library(mass) will not work, even on Windows.

- R no longer accepts associative use of relational operators. That is, 3 < 2 < 1 (which used to evalute as TRUE!) now causes a syntax error. If this breaks existing code, just add parentheses — or braces in the case of plotmath.

- The R parser now allows multiline strings, without escaping the newlines with backslashes (the old method still works). Patch by Mark Bravington.

New features

- There is a new atomic vector type, class "raw". See ?raw for full details including the operators and utility functions provided.

- The default barplot() method by default uses a gamma-corrected grey palette (rather than the heat color palette) for coloring its output when given a matrix.

- The 'formula' method for boxplot() has a 'na.action' argument, defaulting to NULL. This is mainly useful if the response is a matrix when the previous default of 'na.omit' would omit entire rows. (Related to PR#6846.)

  boxplot() and bxp() now obey global 'par' settings and also allow the specification of graphical options in more detail, compatibly with S-PLUS (fulfilling wishlist entry PR#6832) thanks to contributions from Arni Magnusson. For consistency, 'boxwex' is not an explicit argument anymore.

- chull() has been moved to package 'graphics' (as it uses xy.coords).

- There is now a coef() method for summaries of "nls" objects.

- compareVersion(), packageDescription() and read.00Index() have been moved to package 'utils'.

- convolve(), fft(), mvfft() and nextn() have been moved to package 'stats'.

- coplot() now makes use of 'cex.lab' and 'font.lab' par() settings.

- cumsum/prod/max/min() now preserve names.

- data(), .path.packages() and .find.packages() now interpret package = NULL to mean all loaded packages.

- data.frame() and its replacement methods remove the names from vector columns. Using I() will ensure that names are preserved.

- data.frame(check.names = TRUE) (the default) enforces unique names, as S does.

- .Defunct() now has 'new' and 'package' arguments like those of .Deprecated().

- The plot() method for "dendrogram" objects now respects many more nodePar and edgePar settings and for edge labeling computes the extents of the diamond more correctly.

- deparse(), dput() and dump() have a new 'control' argument to control the level of detail when deparsing. dump() defaults to the most detail, the others default to less. See ?.deparseOpts for the details.

  They now evaluate promises by default: see ?dump for details.

- dir.create() now expands ~ in filenames.

- download.file() has a new progress meter (under Unix) if the length of the file is known — it uses 50 equals signs.

- dyn.load() and library.dynam() return an object describing the DLL that was loaded. For packages with namespaces, the DLL objects are stored in a list within the namespace.

- New function `eapply()`: apply for environments. The supplied function is applied to each element of the environment; the order of application and the order of the results are not specified.

- `edit()` and `fix()` use the object name in the window caption on some platforms (e.g. Windows).

- Function `file.edit()` function added: like `file.show()`, but allows editing.

- Function `file.info()` can return file sizes > 2Gb if the underlying OS supports such.

- `fisher.test(*, conf.int=FALSE)` allows the confidence interval computation to be skipped.

- `formula()` methods for classes "lm" and "glm" used the expanded formula (with '.' expanded) from the terms component.

- The 'formula' method for `ftable()` now looks for variables in the environment of the formula before the usual search path.

- A new function `getDLLRegisteredRoutines()` returns information about the routines available from a DLL that were explicitly registered with R's dynamic loading facilities.

- A new function `getLoadedDLLs()` returns information about the DLLs that are currently loaded within this session.

- The package element returned by `getNativeSymbolInfo()` contains reference to both the internal object used to resolve symbols with the DLL, and the internal DllInfo structure used to represent the DLL within R.

- `help()` now returns information about available documentation for a given topic, and notifies about multiple matches. It has a separate `print()` method.

  If the latex help files were not installed, `help()` will offer to create a latex file on-the-fly from the installed .Rd file.

- `heatmap()` has a new argument 'reorderfun'.

- Most versions of `install.packages()` have a new optional argument `dependencies = TRUE` which will not only fetch the packages but also their uninstalled dependencies and their dependencies.

  The Unix version of `install.packages()` attempts to install packages in an order that reflects their dependencies. (This is not needed for binary installs as used under Windows.)

- `interaction()` has new argument 'sep'.

- `interaction.plot()` allows `type = "b"` and doesn't give spurious warnings when passed a `matplot()`-only argument such as 'main'.

- `is.integer()` and `is.numeric()` always return `FALSE` for a factor. (Previously they were true and false respectively for well-formed factors, but it is possible to create factors with non-integer codes by underhand means.)

- New functions `is.leaf()`, `dendrapply()` and a `labels()` method for dendrogram objects.

- `legend()` has an argument 'pt.lwd' and setting 'density' now works because 'angle' now defaults to 45 (mostly contributed by Uwe Ligges).

- `library()` now checks the version dependence (if any) of required packages mentioned in the `Depends:` field of the `DESCRIPTION` file.

- `load()` now detects and gives a warning (rather than an error) for empty input, and tries to detect (but not correct) files which have had LF replaced by CR.

- `ls.str()` and `lsf.str()` now return an object of class `ls_str` which has a print method.

- `make.names()` has a new argument `allow_`, which if false allows its behaviour in R 1.8.1 to be reproduced.

- The 'formula' method for `mosaicplot()` has a 'na.action' argument defaulting to 'na.omit'.

- `model.frame()` now warns if it is given `data = newdata` and it creates a model frame with a different number of rows from that implied by the size of 'newdata'.

  Time series attributes are never copied to variables in the model frame unless `na.action = NULL`. (This was always the intention, but they sometimes were as the result of an earlier bug fix.)

- There is a new 'padj' argument to `mtext()` and `axis()`. Code patch provided by Uwe Ligges (fixes PR#1659 and PR#7188).

- Function `package.dependencies()` has been moved to package 'tools'.

- The 'formula' method for `pairs()` has a 'na.action' argument, defaulting to 'na.pass', rather than the value of `getOption("na.action")`.

- There are five new `par()` settings:

  'family' can be used to specify a font family for graphics text. This is a device-independent family specification which gets mapped by the graphics device to a device-specific font specification (see, for example, `postscriptFonts()`). Currently, only PostScript, PDF, X11, Quartz, and Windows respond to this setting.

  'lend', 'ljoin', and 'lmitre' control the cap style and join style for drawing lines (only noticeable on thick lines or borders). Currently, only PostScript, PDF, X11, and Quartz respond to these settings.

  'lheight' is a multiplier used in determining the vertical spacing of multi-line text.

  All of these settings are currently only available via `par()` (i.e., not in-line as arguments to `plot()`, `lines()`, ...)

- PCRE (as used by grep etc) has been updated to version 5.0.

- A 'version' argument has been added to `pdf()` device. If this is set to "1.4", the device will support transparent colours.

- `plot.xy()`, the workhorse function of points, lines and plot.default now has 'lwd' as explicit argument instead of implicitly in `"..."`, and now recycles 'lwd' where it makes sense, i.e. for line based plot symbols.

- The `png()` and `jpeg()` devices (and the `bmp()` device under Windows) now allow a nominal resolution to be recorded in the file.

- New functions to control mapping from device-independent graphics font family to device-specific family: `postscriptFont()` and `postscriptFonts()` (for `postscript()` and `pdf()`); `X11Font()` and `X11Fonts()`; `windowsFont()` and `windowsFonts()`; `quartzFont()` and `quartzFonts()`.

- power (`x^y`) has optimised code for y = 2.

- `prcomp()` is now generic, with a formula method (based on an idea of Jari Oksanen).

  `prcomp()` now has a simple `predict()` method.

- `printCoefmat()` has a new logical argument 'signif.legend'.

- `quantile()` has the option of several methods described in Hyndman and Fan (1996). (Contributed by Rob Hyndman.)

- `rank()` has two new 'ties.method's, "min" and "max".

- New function `read.fortran()` reads Fortran-style fixed-format specifications.

- `read.fwf()` reads multiline records, is faster for large files.

- `read.table()` now accepts `"NULL"`, `"factor"`, `"Date"` and `"POSIXct"` as possible values of colClasses, and colClasses can be a named character vector.

- `readChar()` can now read strings with embedded nuls.

- The "dendrogram" method for `reorder()` now has a 'agglo.FUN' argument for specification of a weights agglomeration function.

- New `reorder()` method for factors, slightly extending that in lattice. Contributed by Deepayan Sarkar.

- Replaying a plot (with `replayPlot()` or via autoprinting) now automagically opens a device if none is open.

- `replayPlot()` issues a warning if an attempt is made to replay a plot that was recorded using a different R version (the format for recorded plots is not guaranteed to be stable across different R versions). The Windows-menu equivalent (History...Get from variable) issues a similar warning.

- `reshape()` can handle multiple 'id' variables.

- It is now possible to specify colours with a full alpha transparency channel via the new 'alpha' argument to the `rgb()` and `hsv()` functions, or as a string of the form "#RRGGBBAA".

  NOTE: most devices draw nothing if a colour is not opaque, but PDF and Quartz devices will render semitransparent colours.

  A new argument 'alpha' to the function `col2rgb()` provides the ability to return the alpha component of colours (as well as the red, green, and blue components).

- `save()` now checks that a binary connection is used.

- `seek()` on connections now accepts and returns a double for the file position. This allows >2Gb files to be handled on a 64-bit platform.

- `source()` with `echo = TRUE` uses the function source attribute when displaying commands as they are parsed.

- `setClass()` and its utilities now warn if either superclasses or classes for slots are undefined. (Use `setOldClass` to register S3 classes for use as slots)

- `str(obj)` now displays more reasonably the STRucture of S4 objects. It is also improved for language objects and lists with promise components.

  The method for class "dendrogram" has a new argument 'stem' and indicates when it's not printing all levels (as typically when e.g., `max.level = 2`).

  Specifying `max.level = 0` now allows to suppress all but the top level for hierarchical objects such as lists. This is different to previous behavior which was the default behavior of giving all levels is unchanged. The default behavior is unchanged but now specified by `max.level = NA`.

- `system.time()` has a new argument 'gcFirst' which, when `TRUE`, forces a garbage collection before timing begins.

- `tail()` of a matrix now displays the original row numbers.

- The default method for `text()` now coerces a factor to character and not to its internal codes. This is incompatible with S but seems what users would expect.

  It now also recycles (x,y) to the length of 'labels' if that is longer. This is now compatible with `grid.text()` and S. (See also PR#7084.)

- `TukeyHSD()` now labels comparisons when applied to an interaction in an `aov()` fit. It detects non-factor terms in 'which' and drops them if sensible to do so.

- There is now a replacement method for `window()`, to allow a range of values of time series to be replaced by specifying the start and end times (and optionally a frequency).

- If `writeLines()` is given a connection that is not open, it now attempts to open it in `mode = "wt"` rather than the default mode specified when creating the connection.

- The screen devices `x11()`, `windows()` and `quartz()` have a new argument 'bg' to set the default background colour.

- Subassignments involving `NA`s and with a replacement value of length > 0 are now disallowed. (They were handled inconsistently in R < 2.0.0, see PR#7210.) For data frames they are disallowed altogether, even for logical matrix indices (the only case which used to work).

- The way the comparison operators handle a list argument has been rationalized so a few more cases will now work – see `?Comparison`.

- Indexing a vector by a character vector was slow if both the vector and index were long (say 10,000). Now hashing is used and the time should be linear in the longer of the lengths (but more memory is used).

- Printing a character string with embedded nuls now prints the whole string, and non-printable characters are represented by octal escape sequences.

- Objects created from a formally defined class now include the name of the corresponding package as an attribute in the object's class. This allows packages with namespaces to have private (non-exported) classes.

- Changes to package 'grid':

  - Calculation of number of circles to draw in `circleGrob` now looks at length of y and r as well as length of x.

  - Calculation of number of rectangles to draw in `rectGrob` now looks at length of y, w, and h as well as length of x.

  - All primitives (rectangles, lines, text, ...) now handle non-finite values (`NA, Inf, -Inf, NaN`) for locations and sizes. Non-finite values for locations, sizes, and scales of viewports result in error messages. There is a new vignette(`nonfinite`) which describes this new behaviour.

  - Fixed (unreported) bug in drawing circles. Now checks that radius is non-negative.

  - `downViewport()` now reports the depth it went down to find a viewport. Handy for "going back" to where you started.

  - The "alpha" `gpar()` is now multiplied by the alpha channel of colours when creating a gcontext. This means that `gpar(alpha=)` settings now affect internal colours so grid alpha transparency settings now are sent to graphics devices. The alpha setting is also cumulative.

  - Editing a gp slot in a grob is now incremental.

  - The "cex" gpar is now cumulative. For example ...

  - New childNames() function to list the names of children of a gTree.

  - The "grep" and "global" arguments have been implemented for `grid.[add|edit|get|remove]Grob()` functions.

    The "grep" argument has also been implemented for the `grid.set()` and `setGrob()`.

  - New function grid.grab() which creates a gTree from the current display list (i.e., the current page of output can be converted into a single gTree object with all grobs on the current page as children of the gTree and all the viewports used in drawing the current page in the childrenvp slot of the gTree).
  - New "lineend", "linejoin", and "linemitre" gpar()s: line end can be "round", "butt", or "square"; line join can be "round", "mitre", or "bevel"; line mitre can be any number larger than 1 (controls when a mitre join gets turned into a bevel join; proportional to angle between lines at join; very big number means that conversion only happens for lines that are almost parallel at join).
  - New grid.prompt() function for controlling whether the user is prompted before starting a new page of output.

    Grid no longer responds to the par(ask) setting in the "graphics" package.

- The tcltk package has had the tkcmd() function renamed as tcl() since it could be used to invoke commands that had nothing to do with Tk. The old name is retained, but will be deprecated in a future release. Similarly, we now have tclopen(), tclclose(), tclread(), tclputs(), tclfile.tail(), and tclfile.dir() replacing counterparts starting with "tk", with old names retained for now.

New and changed utilities

- R CMD check now checks for file names in a directory that differ only by case.

- R CMD check now checks Rd files using R code from package tools, and gives refined diagnostics about "likely" Rd problems (stray top-level text which is silently discarded by Rdconv).

- R CMD INSTALL now fails for packages with incomplete/invalid DESCRIPTION metadata, using new code from package tools which is also used by R CMD check.

- list_files_with_exts (package 'tools') now handles zipped directories.

- Package 'tools' now provides Rd_parse(), a simple top-level parser/analyzer for R documentation format.

- tools::codoc() (and hence R CMD check) now checks any documentation for registered S3 methods and unexported objects in packages with namespaces.

- Package 'utils' contains several new functions:

  - Generics toBibtex() and toLatex() for converting R objects to BibTeX and LaTeX (but almost no methods yet).

  - A much improved citation() function which also has a package argument. By default the citation is auto-generated from the package DESCRIPTION, the file inst/CITATION can be used to override this, see help(citation) and help(citEntry).

  - sessionInfo() can be used to include version information about R and R packages in text or LaTeX documents.

Documentation changes

- The DVI and PDF manuals are now all made on the paper specified by R_PAPERSIZE (default 'a4'), even the .texi manuals which were made on US letter paper in previous versions.

- The reference manual now omits 'internal' help pages.

- There is a new help page shown by help("Memory-limits") which documents the current design limitations on large objects.

- The format of the LaTeX version of the documentation has changed. The old format is still accepted, but only the new resolves cross-references to object names containing _, for example.

- HTML help pages now contain a reference to the package and version in the footer, and HTML package index pages give their name and version at the top.

- All manuals in the 2.x series have new ISBN numbers.

- The *R Data Import/Export* manual has been revised and has a new chapter on *Reading Excel spreadsheets*.

Changes in C-level facilities

- The PACKAGE argument for .C/.Call/.Fortran/ .External can (and should) be omitted if the call is within code within a package with a namespace. This ensures that the native routine being called is found in the DLL of the correct version of the package if multiple versions of a package are loaded in the R session. Using a namespace and omitting the PACKAGE argument is currently the only way to ensure that the correct version is used.

- The header Rmath.h contains a definition for R_VERSION_STRING which can be used to track different versions of R and libRmath.
- The Makefile in src/nmath/standalone now has 'install' and 'uninstall' targets – see the README file in that directory.
- More of the header files, including Rinternals.h, Rdefines.h and Rversion.h, are now suitable for calling directly from C++.

Newly deprecated and defunct

- Direct use of R INSTALL|REMOVE|BATCH|COMPILE| SHLIB has been removed: use R CMD instead.
- La.eigen(), tetragamma(), pentagamma(), package.contents() and package.description() are defunct.
- The undocumented function newestVersion() is no longer exported from package utils. (Mainly because it was not completely general.)
- C-level entry point ptr_R_GetX11Image has been removed, as it was replaced by R_GetX11Image at 1.7.0.
- The undocumented C-level entry point R_IsNaNorNA has been removed. It was used in a couple of packages, and should be replaced by a call to the documented macro ISNAN.
- The gnome/GNOME graphics device is now defunct.

Installation changes

- Arithmetic supporting +/-Inf, NaNs and the IEC 60559 (aka IEEE 754) standard is now required — the partial and often untested support for more limited arithmetic has been removed.

  The C99 macro isfinite is used in preference to finite if available (and its correct functioning is checked at configure time).

  Where isfinite or finite is available and works, it is used as the substitution value for R_FINITE. On some platforms this leads to a performance gain. (This applies to compiled code in packages only for isfinite.)
- The dynamic libraries libR and libRlapack are now installed in R_HOME/lib rather than R_HOME/bin.
- When --enable-R-shlib is specified, the R executable is now a small executable linked against libR: see the R-admin manual for further discussion. The 'extra' libraries bzip2, pcre, xdr and zlib are now compiled in a way that allows the code to be included in a shared library only if this option is specified, which might improve performance when it is not.
- The main R executable is now R_HOME/exec/R not R_HOME/R.bin, to ease issues on MacOS X. (The location is needed when debugging core dumps, on other platforms.)
- Configure now tests for inline and alternatives, and the src/extra/bzip2 code now (potentially) uses inlining where available and not just under gcc.
- The XPG4 sed is used on Solaris for forming dependencies, which should now be done correctly.
- Makeinfo 4.5 or later is now required for building the HTML and Info versions of the manuals. However, binary distributions need to be made with 4.7 or later to ensure some of the links are correct.
- f2c is not allowed on 64-bit platforms, as it uses longs for Fortran integers.
- There are new options on how to make the PDF version of the reference manual — see the *R Administration and Installation Manual* section 2.2.
- The concatenated Rd files in the installed 'man' directory are now compressed and the R CMD check routines can read the compressed files.
- There is a new configure option --enable-lfs that will build R with support for > 2Gb files on suitable 32-bit Linux systems.

Package installation changes

- The DESCRIPTION file of packages may contain a Imports: field for packages whose namespaces are used but do not need to be attached. Such packages should no longer be listed in Depends:.
- There are new optional fields SaveImage, LazyLoad and LazyData in the DESCRIPTION file. Using SaveImage is preferred to using an empty file install.R.
- A package can contain a file R/sysdata.rda to contain system datasets to be lazy-loaded into the namespace/package environment.

- The packages listed in `Depends:` are now loaded before a package is loaded (or its image is saved or it is prepared for lazy loading). This means that almost all uses of `R_PROFILE.R` and `install.R` are now unnecessary.
- If installation of any package in a bundle fails, `R CMD INSTALL` will back out the installation of all of the bundle, not just the failed package (on both Unix and Windows).

Bug fixes

- Complex superassignments were wrong when a variable with the same name existed locally, and were not documented in R-lang.
- `rbind.data.frame()` dropped names/rownames from columns in all but the first data frame.
- The `dimnames<-` method for data.frames was not checking the validity of the row names.
- Various memory leaks reported by valgrind have been plugged.
- `gzcon()` connections would sometimes read the crc bytes from the wrong place, possibly uninitialized memory.
- Rd.sty contained a length `\middle` that was not needed after a revision in July 2000. It caused problems with LaTeX systems based on e-TeX which are starting to appear.
- `save()` to a connection did not check that the connection was open for writing, nor that non-ascii saves cannot be made to a text-mode connection.
- `phyper()` uses a new algorithm based on Morten Welinder's bug report (PR#6772). This leads to faster code for large arguments and more precise code, e.g. for `phyper(59, 150,150, 60, lower=FALSE)`. This also fixes bug (PR#7064) about `fisher.test()`.
- `{print.default(*, gap = <n>)` now in principle accepts all non-negative values `<n>`.
- `smooth.spline(...)$pen.crit` had a typo in its computation; note this was printed in `print.smooth.spline()` but not used in other "smooth.spline" methods.
- `write.table()` handles zero-row and zero-column inputs correctly.
- `debug()` works on trivial functions instead of crashing (PR#6804)
- `eval()` could alter a data.frame/list second argument, so `with(trees, Girth[1] <- NA)` altered `trees` (and any copy of `trees` too).
- `cor()` could corrupt memory when the standard deviation was zero. (PR#7037)
- `inverse.gaussian()` always printed `1/mu^2` as the link function.
- `constrOptim` now passes `...` arguments through `optim` to the objective function.
- `object.size()` now has a better estimate for character vectors: it was in general too low (but only significantly so for very short character strings) but over-estimated `NA` and duplicated elements.
- `quantile()` now interpolates correctly between finite and infinite values (giving +/-Inf rather than NaN).
- `library()` now gives more informative error messages mentioning the package being loaded.
- Building the reference manual no longer uses roman upright quotes in typewriter output.
- `model.frame()` no longer builds invalid data frames if the data contains time series and rows are omitted by `na.action`.
- `write.table()` did not escape quotes in column names. (PR#7171)
- Range checks missing in recursive assignments using `[[ ]]`. (PR#7196)
- `packageStatus()` reported partially-installed bundles as installed.
- `apply()` failed on an array of dimension ≥ 3 when for each iteration the function returns a named vector of length ≥ 2 (PR#7205)
- The GNOME interface was in some circumstances failing if run from a menu — it needed to always specify that R be interactive.
- `depMtrxToStrings` (part of `pkgDepends`) applied `nrow()` to a non-matrix and aborted on the result.
- Fix some issues with nonsyntactical names in modelling code (PR#7202), relating to backquoting. There are likely more.
- Support for S4 classes that extend basic classes has been fixed in several ways. `as()` methods and `x@.Data` should work better.
- `hist()` and `pretty()` accept (and ignore) infinite values. (PR#7220)
- It is no longer possible to call `gzcon()` more than once on a connection.

- `t.test()` now detects nearly-constant input data. (PR#7225)

- `mle()` had problems if ndeps or parscale was supplied in the control arguments for `optim()`. Also, the profiler is now more careful to reevaluate modified `mle()` calls in its parent environment.

- Fix to rendering of accented superscripts and subscripts e.g., `expression((b[dot(a)]))`. (Patch from Uwe Ligges.)

- `attach(*, pos=1)` now gives a warning (and will give an error).

- `power.*test()` now gives an error when 'sig.level' is outside [0,1]. (PR#7245)

- Fitting a binomial glm with a matrix response lost the names of the response, which should have been transferred to the residuals and fitted values.

- `print.ts()` could get the year wrong because rounding issue (PR#7255)

Changes on CRAN

by Kurt Hornik

New contributed packages

Malmig The Malmig package provides an implementation of Malecot migration model in R together with a number of related functions. By Federico C. F. Calboli and Vincente Canto Cassola together with Martin Maechler authored the function `mtx.exp`.

PBSmapping This software has evolved from fisheries research conducted at the Pacific Biological Station (PBS) in Nanaimo, British Columbia, Canada. It extends the R language to include two-dimensional plotting features similar to those commonly available in a Geographic Information System (GIS). Embedded C code speeds algorithms from computational geometry, such as finding polygons that contain specified point events or converting between longitude-latitude and Universal Transverse Mercator (UTM) coordinates. It includes data for a global shoreline and other data sets in the public domain. By Nicholas Boers, Jon Schnute, Rowan Haigh, and others.

RCurl The package allows one to compose HTTP requests to fetch URIs, post forms, etc., and process the results returned by the Web server. This provides a great deal of control over the HTTP connection and the form of the request while providing a higher-level interface than is available just using R socket connections. Additionally, the underlying implementation is robust and extensive, supporting SSL/HTTPS, cookies, redirects, authentication, etc. By Duncan Temple Lang.

RNetCDF This package provides an interface to Unidata's NetCDF library functions (version 3) and furthermore access to Unidata's udunits calendar conversions. The routines and the documentation follow the NetCDF and udunits C interface, so the corresponding manuals can be consulted for more detailed information. By Pavel Michna.

Rstem An R interface to the C code that implements Porter's word stemming algorithm for collapsing words to a common root to aid comparison of texts. There is code to for different languages (i.e., Danish, Dutch, English, Finnish, French, German, Norwegian, Portuguese, Russian, Spanish, Swedish). However, these may not be applicable if the words require UTF encoding. This is extensible by allowing different routines to be specified to create the C routines used in the stemming, permitting debugging, profiling, pool management, caching, etc. By Duncan Temple Lang.

UNF Computes a universal numeric fingerprint of the data. By Micah Altman.

accuracy This is a suite of tools designed to test and improve the accuracy of statistical computation, including: Summarization of the sensitivity of linear and non-linear models (lm, glm, mle, nls) to measurement and numerical error; A generalized Cholesky method for correcting non-invertible Hessians; Tests for the global optimality of non-linear regression and maximum likelihood results; Tools for obtaining true random numbers using entropy collected from the system and/or entropy servers on the internet; A method for converting floating point numbers to normalized fractions; Benchmark data for checking the accuracy of basic distribution functions. By Micah Altman, Jeff Gill, and Michael P. McDonald.

adehabitat A collection of tools for the analysis of habitat selection by animals. By Clément Calenge, contributions from Mathieu Basille.