M. A. Newton, C. M. Kendziorski, C. S. Richmond, F. R . Blattner, and K. W. Tsui. On differential variability of expression ratios: Improving statistical inference about gene expression changes from microarray data. *Journal of Computational Biology*, 8: 37–52, 2001. 29

M. Sapir and G. A. Churchill. *Estimating the posterior probability of differential gene expression from microarray data*. Poster, The Jackson Laboratory, 2000. `http://www.jax.org/research/churchill/`. 29

Y. H. Yang, M. J. Buckley, S. Dudoit, and T. P. Speed. Comparison of methods for image analysis on cDNA microarray data. *Journal of Computational and Graphical Statistic*, 11(1), 2002a. 25

Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed. Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, 30(4), 2002b. 27, 28

Y. H. Yang, S. Dudoit, P. Luu, and T. P. Speed. Normalization for cDNA microarray data. In Michael L. Bittner, Yidong Chen, Andreas N. Dorsel, and Edward R. Dougherty, editors, *Microarrays: Optical Technologies and Informatics*, volume 4266 of *Proceedings of SPIE*, May 2001. 27, 28

Sandrine Dudoit
University of California, Berkeley
http://www.stat.berkeley.edu/~sandrine
sandrine@stat.berkeley.edu

Yee Hwa (Jean) Yang
University of California, Berkeley
yeehwa@stat.berkeley.edu

Ben Bolstad
University of California, Berkeley
bolstad@stat.berkeley.edu

Changes in R 1.4.0

by the R Core Team

The 'NEWS' file of the R sources and hence this article have been reorganized into several sections (user-visible changes, new features, deprecated & defunct, documentation changes, utilities, C-level facilities, bug fixes) in order to provide a better overview of all the changes in R 1.4.0.

User-visible changes

This is a new section to highlight changes in behaviour, which may be given in more detail in the following sections. Many bug fixes are also user-visible changes.

- The default save format has been changed, so saved workspaces and objects cannot (by default) be read in earlier versions of R.
- The number of bins selected by default in a histogram uses the correct version of Sturges' formula and will usually be one larger.
- `data.frame()` no longer converts logical arguments to factors (following S4 rather than S3).
- `read.table()` has new arguments 'nrows' and 'colClasses'. If the latter is `NA` (the default), conversion is attempted to logical, integer, numeric or complex, not just to numeric.
- `model.matrix()` treats logical variables as a factors with levels `c(FALSE, TRUE)` (rather than 0-1 valued numerical variables). This makes R compatible with all S versions.
- Transparency is now supported on most graphics devices. This means that using `par("bg")`, for example in `legend()`, will by default give a transparent rather than opaque background.
- [dpqr]gamma now has third argument 'rate' for S-compatibility (and for compatibility with exponentials). Calls which use positional matching may need to be altered.
- The meaning of `spar = 0` in `smooth.spline()` has changed.
- `substring()` and `substring<-()` do nothing silently on a character vector of length 0, rather than generating an error. This is consistent with other functions and with S.
- For compatibility with S4, any arithmetic operation using a zero-length vector has a zero-length result. (This was already true for logical operations, which were compatible with S4 rather than S3.)
- `undoc()` and `codoc()` have been moved to the new package **tools**.
- The name of the site profile now defaults to 'R_HOME/etc/Rprofile.site'.

- The startup process for setting environment variables now first searches for a site environment file (given by the environment variable `R_ENVIRON` if set or 'R_HOME/etc/Renviron.site' if not), *and* then for a user '.Renviron' file in the current or the user's home directory.
- Former `stars(*, colors = v)` must now be `stars(*, col.segments = v)`.
- The default methods for `La.svd` and `La.eigen` have changed and so there may be sign changes in singular/eigen vectors, including in `cancor`, `cmdscale`, `factanal`, `princomp`, and `varimax`.

New features

- Transparency is now supported on most graphics devices. Internally colors include an alpha channel for opacity, but at present there is only visible support for transparent/opaque. The new color '"transparent"' (or 'NA' or '"NA"') is transparent, and is the default background color for most devices. Those devices (postscript, XFig, PDF, Windows metafile and printer) that previously treated 'bg = "white"' as transparent now have '"transparent"' as the default and will actually print '"white"'. (NB: you may have 'bg = "white"' saved in `.Postscript.options` in your workspace.)
- A package **methods** has been added, containing formal classes and methods ("S4" methods), implementing the description in the book "Programming with Data". See `?Methods` and the references there for more information.
  - In support of this, the '`@`' operator has been added to the grammar.
  - Method dispatch for formal methods (the `standardGeneric` function), is now a primitive. Aside from efficiency issues, this allows S3-style generics to also have formal methods (not really recommended in the long run, but it should at least work). The C-level dispatch is now implemented for primitives that use either `DispatchGroup` or `DispatchOrEval` internally.
  - A version of the function `plot` in the methods package has arguments `x` and `y`, to allow methods for either or both. See `?setMethod` for examples of such methods.
  - The **methods** package now uses C-level code (from within `DispatchOrEval`) to dispatch any methods defined for primitive functions. As with S3-style methods, methods can only be defined if the first argument satisfies `is.object(x)` (not strictly required for formal methods, but imposed for now for simplicity and efficiency).
- Changes to the **tcltk** package:
  - New interface for accessing Tcl variables, effectively making the R representations lexically scoped. The old form is being deprecated.
  - Callbacks can now be expressions, with slightly unorthodox semantics. In particular this allows bindings to contain 'break' expressions (this is necessary to bind code to e.g. Alt-x without having the key combination also insert an 'x' in a text widget.)
  - A bunch of file handling and dialog functions (previously only available via `tkcmd`) have been added.
- The '?' operator is now an actual function. It can be used (as always) as a unary operator ('?plot') and the grammar now allows it as a binary operator, planned to allow differentiating documentation on the same name but different type ('class?matrix,' for example). So far, no such documentation exists.
- New methods `AIC.default()` and `logLik.glm()`, also fixing the `AIC` method for glm objects.
- `axis.POSIXct()` allows the label date/times to be specified via the new 'at' argument.
- `arrows()` now allows 'length = 0' (and draws no arrowheads).
- Modifications to the access functions for more consistency with S: arguments 'name', 'pos' and 'where' are more flexible in `assign()`, `exists()`, `get()`, `ls()`, `objects()`, `remove()` and `rm()`.
- Three new primitive functions have been added to base: `dataClass()`, `objWithClass()`, and `as.environment()`. The first two are support routines for `class()` and `class<-()` in package **methods**. The third replaces `pos.to.env()` in the functions `get()`, `exists()`, and friends.
- `barplot()` now respects an inline 'cex.axis' argument and has a separate 'cex.names' argument so names and the numeric axis labels can be scaled separately. Also, graphics parameters intended for `axis()` such as 'las' can now be used.
- Shading by lines added to functions `barplot()`, `hist()`, `legend()`, `piechart()`, `polygon()` and `rect()`.

- `bxp()` has a show.names argument allowing labels on a single boxplot; it and hence `boxplot()` now makes use of 'pch', 'cex', and 'bg' for outlier `points()`.

  `bxp()` and `boxplot()` also have an argument 'outline' to suppress outlier drawing S-Plus compatibly.

- New `capabilities()` options '"GNOME"' and '"IEEE754"'.

- New function `casefold()`, a wrapper for `tolower/toupper` provided for compatibility with S-Plus.

- `contour()` is now generic.

- `cor.test()` in package **ctest** now also gives an asymptotic confidence interval for the Pearson product moment correlation coefficient.

- `data()`, `demo()` and `library()` now also return the information about available data sets, demos or packages. Similarly, `help.search()` returns its results.

- `density()` allows 'bw' or 'width' to specify a rule to choose the bandwidth, and rules '"nrd0"' (the previous default), '"nrd"', '"ucv"', '"bcv"', '"SJ-ste"' and '"SJ-dpi"' are supplied (based on functions in package **MASS**).

- `df.residual()` now has a default method, used for classes '"lm"' and '"glm"'.

- New argument 'cacheOK' to `download.file()` to request cache flushing.

  All methods for `download.file()` do tilde-expansion on the path name.

  The internal `download.file()` etc now allow URLs of the form 'ftp://user@foo.bar/' and 'ftp://user:pass@foo.bar/'

- `duplicated()` and `unique()` are now generic functions with methods for data frames (as well as atomic vectors).

- `factanal()` and `princomp()` use `napredict()` on their scores, so 'na.action = na.exclude' is supported.

- Function `getNativeSymbolInfo()` returns details about a native routine, potentially including its address, the library in which it is located, the interface by which it can be called and the number of parameters.

- Functions such as `help()` which perform library or package index searches now use NULL as default for their 'lib.loc' argument so that missingness can be propagated more easily. The default corresponds to all currently known libraries as before.

- Added function `file.rename()`.

- `hist.default()` allows 'breaks' to specify a rule to choose the number of classes, and rules '"Sturges"' (the previous default), '"Scott"' and '"FD"' (Freedman-Diaconis) are supplied (based on package **MASS**).

- Function `identical()`, a fast and reliable way to test for exact equality of two objects.

- New generic function `is.na<-()`, from S4. This is by default equivalent to 'x[value] <- NA' but may differ, e.g., for factors where '"NA"' is a level.

- `is.`*xxx* reached through `do_is` are now generic.

- `La.eigen()` and `La.svd()` have new default methods to use later (and often much faster) LAPACK routines. The difference is most noticeable on systems with optimized BLAS libraries.

- `length()` is now generic.

- New function `.libPaths()` for getting or setting the paths to the library trees R knows about. This is still stored in '.lib.loc', which however should no longer be accessed directly.

- Using lm/glm/... with 'data' a matrix rather than a data frame now gives a specific error message.

- `loess()`, `lqs()`, `nls()` and `ppr()` use the standard NA-handling and so support 'na.action = na.exclude'.

- `mahalanobis()` now has a 'tol' argument to be passed to `solve()`.

- `mean()` has a 'data.frame' method applying mean column-by-column. When applied to non-numeric data `mean()` now returns 'NA' rather than a confusing error message (for compatibility with S4). Logicals are still coerced to numeric.

- The formula interface to `mosaicplot()` now allows a contingency table as data argument.

- `new.env()` is now internal and allows you to set hashing. Also, `parent.env()` and `parent.env<-()` are included to provide direct access to setting and retrieving environments.

- Function `nsl()` to look up IP addresses of hosts: intended as a way to test for internet connectivity.

- `Ops()`, `cbind()`, `diff()` and `na.omit()` methods for time series objects moved from package **ts** to package **base**.
- New option 'download.file.method' can be used to set the default method for `download.file()` and functions which use it such as `update.packages()`.
- `order()` and `sort.list()` now implement 'na.last = FALSE' and 'na.last = NA'.
- Started work on new package management system: `packageStatus()` and friends.
- `page()` has a new 'method' argument allowing 'method = print'.
- `png()`, `jpeg()` and `bmp()` devices now have a 'bg' argument to set the background color: useful to set '"transparent"' on `png()`.
- Changes to the `postscript()` device:
  - The symbol font can now be set on a `postscript()` device, and support has been added for using Computer Modern type-1 fonts (including for symbols). (Contributed by Brian D'Urso.)
  - There is now support for URW font families: this will give access to more characters and more appropriate metrics on PostScript devices using URW fonts (such as ghostscript).
  - %%IncludeResource comments have been added to the output. (Contributed by Brian D'Urso.)
- `predict.ppr()` now predicts on 'newdata' containing NAs.
- `princomp()` now has a formula interface.
- `readChar()` now returns what is available if fewer characters than requested are on the file.
- `readline()` allows up to 256 chars for the prompt.
- `read.table()`, `scan()` and `count.fields()` have a new argument 'comment.char', default '#', that can be used to start comments on a line.
- New function `reg.finalizer()` to provide R interface to finalization.
- `reshape()` extends `reshapeLong` and `reshapeWide`, which are deprecated.
- `rle()` now returns a classed object, has a print method and an inverse.
- Changes to `save()` and friends:
  - `save()` now takes an 'envir' argument for specifying where items to be saved are to be found.
  - A new default format for saved workspaces has been introduced. This format provides support for some new internal data types, produces smaller save files when saving code, and provides a basis for a more flexible serialization mechanism.
  - Modified 'save' internals to improve performance when saving large collections of code.
  - `save()` and `save.image()` now take a 'version' argument to specify the workspace file-format version to use. The version used from R 0.99.0 to 1.3.1 is version 1. The new default format is version 2. `load()` can read a version 2 saved workspace if it is compressed.
  - `save()` and `save.image()` now take a 'compress' argument to specify that the saved image should be written using the zlib compression facilities.
  - `save.image()` now takes an argument 'ascii'.
  - `save.image()` now takes an argument 'safe'. If 'TRUE', the default, a temporary file is used for creating the saved workspace. The temporary file is renamed if the save succeeds. This preserves an existing workspace if the save fails, but at the cost of using extra disk space during the save.
  - `save.image()` default arguments can be specified in the 'save.image.defaults' option. These specifications are used when `save.image()` is called from `q()` or GUI analogs.
- `scan()` allows unlimited (by R) lengths of input lines, instead of a limit of 8190 chars.
- `smooth.spline()` has a new 'control.spar' argument and returns 'lambda' besides 'spar'. 'spar' ≤ 0 is now valid and allows to go more closely towards interpolation (lambda $\rightarrow 0$) than before. This also fixes `smooth.spline()` behavior for 'df $\approx$ n - 2'. Better error messages in several situations.

  Note that spar = 0 is no longer the default and no longer entails cross-validation.
- `stars()` has been enhanced; new 'mar' argument uses smaller `mar(gins)` by default; further 'nrow' and 'ncol' as S-Plus, 'frame.plot', 'flip.labels', 'lty' and explicit 'main', 'sub', 'xlab' and 'ylab'. Note that 'colors' has been

replaced by 'col.segments' and there's a new 'col.stars'.

`stars()` now returns the locations invisibly.

- `step()` is now closer to `stepAIC()` and so handles a wider range of objects (but stepAIC [in **MASS**] is still more powerful).
- `symbols()` now has automatic xlab and ylab and a main argument which eliminates an incorrect warning. It better checks wrongly scaled arguments.
- `Sys.setlocale()` now issues a warning if it fails.
- An enhanced function `type.convert()` is now a documented function, rather than just internal to `read.table()`.
- `warning()` allows multiple arguments, following S4's style.
- New function `with()` for evaluating expressions in environments constructed from data.
- Unix `x11()` devices can now have a canvas color set, which can help to distinguish plotting '"white"' from plotting '"transparent"'.
- On Unix, `X11()`, `png()` and `jpeg()` now give informative warnings if they fail to open the device.
- The startup processing now interprets escapes in the values of environment variables set in 'R_HOME/etc/Renviron' in a similar way to most shells.
- The operator '=' is now allowed as an assignment operator in the grammar, for consistency with other languages, including recent versions of S-Plus. Assignments with '=' are basically allowed only at top-level and in braced or parenthesized expressions, to make famous errors such as `if(x=0) 1 else 2` illegal in the grammar. (There is a plan to gradually eliminate the underscore as an assignment in future versions of R.)
- Finalizers can be registered to be run on system exit for both reachable and unreachable objects.
- integer addition, subtraction, and multiplication now return NA's on overflow and issue a warning.
- Printing factors with both level '"NA"' and missing values uses '<NA>' for the missing values to distinguish them.
- Added an experimental interface for locking environments and individual bindings. Also added support for "active bindings" that link a variable to a function (useful for example for linking an R variable to an internal C global).
- GNOME interface now has separate colours for input and output text (like the windows GUI). These can be modified via the properties dialogue.
- Output from the GNOME console is block buffered for increased speed
- The GNOME console inherits standard emacs-style keyboard shortcuts from the GtkText widget for cursor motion, editing and selection. These have been modified to allow for the prompt at the beginning of the command line.
- One can register R functions and C routines to be called at the end of the successful evaluation of each top-level expression, for example to perform auto-saves, update displays, etc. See `addTaskCallback()` and `taskCallbackManager()`. See http://developer.r-project.org/TaskHandlers.pdf.

Deprecated & defunct

- `.Alias` has been removed from all R sources and deprecated.
- `reshapeLong()` and `reshapeWide()` are deprecated in favour of `reshape()`.
- Previously deprecated functions `httpclient()`, `parse.dcf()`, `read.table.url()`, `scan.url()`, and `source.url()` are defunct. Method '"socket"' for `download.file()` no longer exists.

Documentation changes

- *Writing R Extensions* has a new chapter on generic/method functions.

Utilities

- New package **tools** for package development and administration tools, containing the QA tools `checkFF()`, `codoc()` and `undoc()` previously in package **base**, as well as the following new ones:
  - `checkAssignFuns()` for checking whether the final argument of assignment functions in a package is named 'value'.

  - `checkDocArgs()` for checking whether all arguments shown in `\usage` of Rd files are documented in the corresponding `\arguments`.
  - `checkMethods()` for checking whether all methods defined in a package have all arguments of their generic.
  - `checkTnF()` for finding expressions containing the symbols 'T' and 'F'.

- `R CMD Rd2dvi` has more convenient defaults for its output file.

- `R CMD check` now also fully checks the 'Depends' field in the package 'DESCRIPTION' file. It also tests for syntax errors in the R code, whether all methods in the code have all arguments of the corresponding generic, for arguments shown in `\usage` but not documented in `\arguments`, and whether assignment functions have their final argument named 'value'.

C-level facilities

- `arraySubscript` and `vectorSubscript` are now available to package users. All "array-like" packages can use a standard method for calculating subscripts.

- The C routine `type2symbol`, similar to `type2str`, returns a symbol corresponding to the type supplied as an argument.

- The macro `SHLIB_EXT` now includes '.', e.g. '".so"' or '".dll"', since the Mac uses "Lib" without a '.'.

- New FORTRAN entry points `rwarn()` and `rexit()` for warnings and error exits from compiled Fortran code.

- A new serialization mechanism is available that can be used to serialize R objects to connections or to strings. This mechanism is used for the version 2 save format. For now, only an internal C interface is available.

- `R_tryEval()` added for evaluating expressions from C code with errors handled but guaranteed to return to the calling C routine. This is used in embedding R in other applications and languages.

- Support for `attach()`'ing user-defined tables of variables is available and accessed via the **RObjectTables** package currently at http://www.omegahat.org/RObjectTables.

Bug fixes

- Fixed 'share/perl/massage-examples.pl' to detect instances of `par()` at the very start of a line.

- Fixed Pearson residuals for glms with non-canonical `link.(PR#1123)`. Fixed them again for weights (PR#1175).

- Fixed an inconsistency in the evaluation context for on.exit expressions between explicit calls to 'return' and falling off the end returns.

- The code in `model.matrix.default()` handling contrasts was assuming a response was present, and so without a response was failing to record the contrasts for the first variable if it was a factor.

- `diffinv()` could get the time base wrong in some cases.

- `file.append()` was opening all files in text mode: mattered on Windows and classic Macintosh. (PR#1085)

- `f[] <- g` now works for factor `f`.

- `substr<-()` was misbehaving if the replacement was too short.

- The version of 'packages.html' generated when building R or installing packages had an incorrect link to the style sheet. The version used by `help.start()` was correct. (PR#1090)

- `rowsum()` now gives character (not factor codes) as rownames. (PR#1092)

- `plot.POSIXct` and `plot.POSIXlt` now respect the 'xaxt' parameter.

- It is now possible to predict from an intercept-only model: previously `model.matrix.default()` objected to a 0-column model frame.

- `c.POSIXct` was not setting the right classes in 1.3.x.

- `cor(*, use = "all.obs")` ≤ 1 is now guaranteed which ensures that `sqrt(1 - r^2)` is always ok in `cor.test()`. (PR#1099)

- `anova.glm()` had a missing 'drop=FALSE' and so failed for some intercept-less models.

- `predict.arima0()` now accepts vector as well as matrix 'newxreg' arguments.

- `cbind(d,f)` now works for 0-column dataframes. This fixes PR#1102.

- `plot(ts(0:99), log = "y")` now works.

- method '"gnudoit"' of `bug.report()` was incorrectly documented as '"gnuclient"' (PR#1108)

- saving with 'ascii=TRUE' mangled backslashes. (PR#1115)
- `frac(,)` and others now adds a gap appropriately. (PR#1101)
- `logLik.lm()` now uses the correct '"df"' (**nlme** legacy code).
- `closeAllConnections()` works again, and closes all `sink()` diversions.
- `sink(type="message")` works again.
- `sink.number` was (accidentally) returning the result invisibly.
- `as.POSIXct("NA")` (or ..lt) now work; hence, `merge(*, all=TRUE)` now works with dataframes containing POSIXt date columns.
- `integer(2^30+1)` and similar ones do not segfault anymore but duly report allocation errors.
- `seq(0, 0, 1)` now works (PR#1133).
- `reshapeWide()` got it wrong if the '"i"' factor was not sorted (the function is now deprecated since `reshape()` is there, but the bug still needed fixing ...)
- PR#757 was fixed incorrectly, causing improper subsetting of 'pch' etc. in `plot.formula()`.
- `library()` no longer removes environments of functions that are not defined in the top-level package scope. Also, packages loaded by `require()` when sourcing package code are now visible in the remaining source evaluations.
- `names(d) <- v` now works (again) for '"dist"' objects d. (PR#1129)
- Workarounds for problems with incompletely specified date-times in `strptime()` which were seen only on glibc-based systems (PR#1155).
- `promax()` was returning the wrong rotation matrix. (PR#1146)
- The [pqr]signrank and [pqr]wilcox functions failed to check that memory has been allocated (PR#1149), and had (often large) memory leaks if interrupted. They now can be interrupted on Windows and MacOS and don't leak memory.
- `range(numeric(0))` is now `c(NA, NA)` not `NA`.
- `round(x, digits)` for `digits` ≤ 0 always gives an integral answer. Previously it might not due to rounding errors in fround. (PR#1138/9)
- Several memory leaks on interrupting functions have been circumvented. Functions `lqs()` and `mve()` can now be interrupted on Windows and MacOS.
- `image()` was finding incorrect breakpoints from irregularly-spaced midpoints. (PR#1160)
- Use fuzz in the 2-sample Kolmogorov-Smirnov test in package **ctest** to avoid rounding errors (PR#1004, follow-up).
- Use exact Hodges-Lehmann estimators for the Wilcoxon tests in package **ctest** (PR#1150).
- Arithmetic which coerced types could lose the class information, for example 'table - real' had a class attribute but was not treated as a classed object.
- Internal ftp client could crash R under error conditions such as failing to parse the URL.
- Internal clipping code for circles could attempt to allocate a vector of length -1 (related to PR#1174)
- The hash function used internally in `match()`, `unique()` and `duplicated()` was very inefficient for integers stored as numeric, on little-endian chips. It was failing to hash the imaginary part of complex numbers.
- `fifo()` no longer tries to truncate on opening in modes including '"w"'. (Caused the fifo example to fail on HP-UX.)
- Output over 1024 characters was discarded from the GNOME console.
- `rug()` now correctly warns about clipped values also for logarithmic axes and has a 'quiet' argument for suppressing these (PR#1188).
- `model.matrix.default` was not handling correctly 'contrasts.arg' which did not supply a full set of contrasts (PR#1187).
- The 'width' argument of `density()` was only compatible with S for a Gaussian kernel: now it is compatible in all cases.
- The `rbinom()` C code had a transcription error from the original Fortran which led to a small deviation from the intended distribution. (PR#1190)
- `pt(t, , ncp=0)` was wrong if `t` was +/-Inf.
- Subsetting grouping factors gave incorrect degrees of freedom for some tests in package **ctest**. (PR#1124)
- `writeBin()` had a memory leak.
- `qbeta(0.25, 0.143891, 0.05)` was (incorrectly) 3e-308. (PR#1201)

- Fixed alignment problem in 'ppr.f' on Irix. (PR#1002, 1026)
- `glm()` failed on null binomial models. (PR#1216)
- `La.svd()` with 'nu' = 0 or 'nv' = 0 could fail as the matrix passed to DGESVD was not of dimension at least one (it was a vector).
- Rownames in 'xcoef' and 'ycoef' of `cancor()` were wrong if 'x' or 'y' was rank-deficient.
- `lqs()` could give warnings if there was an exact fit. (PR#1184)
- `aov()` didn't find free-floating variables for `Error()` terms when called from inside another function.
- `write.table()` failed if asked to quote a numerical matrix with no row names. (PR#1219)
- `rlnorm( *, *, sd=0)` now returns the mean, `rnbinom(*, *, prob=1)` gives 0, (PR#1218).

Changes in R 1.4.1

by the R Core Team

Bug fixes

- `scan(multi.line = FALSE)` now always gives an immediate error message if a line is incomplete. (As requested in PR#1210)
- `read.table()` is no longer very slow in processing comments: moved to C code and fewer lines checked.
- `type.convert()` could give stack imbalance warnings if used with 'as.is = TRUE'.
- predict.mlm ignored newdata (PR#1226) and also offsets.
- `demo(tkttest)` was inadvertently changed in 1.4.0 so that it would evaluate the requested test, but not display the result.
- `stars(scale = TRUE)` (the default) now works as documented (and as S does). Previously it only scaled the maximum to 1. (PR#1230)
- `d0 <- data.frame(a = 0); data.matrix(d0[0, 0])` and `data.matrix(d0[, 0])` now work.
- `plot(`*multiple time series*`, plot.type = "single")` was computing 'ylim' from the first series only.
- `plot.acf()` has a new 'xpd = `par("xpd")`' argument which by default *does* clipping (of the horizontal lines) as desired ('xpd' = NA was used before, erroneously in most cases).
- `predict(smooth.spline(.), deriv = 1)` now works.
- `identify()` failed when `x` is a structure/matrix. (PR#1238)
- `getMethod()` returns NULL when 'optional=TRUE' as promised in the documentation.
- `setMethod()` allows `...` to be one of the arguments omitted in the method definition (but so far no check for `...` being missing)
- Allow `round()` to work again on very large numbers (introduced in fixing PR#1138). (PR#1254)
- 'Rinternals.h' is now accepted by a C++ compiler.
- `type.convert()` was failing to detect integer overflow.
- `piechart()` was defaulting to foreground colour (black) fills rather than background (as used in 1.3.1 and earlier). Now background is used, but be aware that as from 1.4.0 this may be transparent.
- `La.eigen(*, only.values=TRUE)` does not segfault anymore in one branch (PR#1262).
- `cut()` now produces correct default labels even when 'include.lowest = TRUE' (PR#1263).
- `reformulate()` works properly with a response.
- `cmdscale(*, k = 1)` now works properly.
- Options 'by = "month"' and 'by = "year"' to `seq.POSIXt()` will always take account of changes to/from daylight savings time: this was not working on some platforms.
- `glm.fit.null()` now accepts all the arguments of `glm.fit()` (it could be called from `glm.fit` with arguments it did not accept), and is now documented.
- `cov.wt(cbind(1), cor = TRUE)` now works.
- `predict(glm.object, se.fit = TRUE)` was failing if the fit involved an offset.