

- Uwe Ligges, Stefano Iacus and Simon Urbanek on R on various platforms
- Paul Murrell on drawing graphs in R
- Peter Rossi on Bayesian statistics with marketing data in R

The conference web page hosts abstracts and slides for all presentations, highlighting the role R is playing in many fields from teaching to research to commercial applications. The conference concluded with a panel discussion on getting recognition for excellence in computational statistics. The panelists included Jan de Leeuw (Editor, *Journal of Statistical Software*), Timothy Hesterberg (Past-Chair, ASA Section on Statistical Computing), Martina Mittlböck, (Associate Editor, *Computational Statistics & Data Analysis*), Paul Murrell (Chair, ASA Section on Statistical Graphics), Erich Neuwirth (Associate Editor, *Computational Statistics*), and Luke Tierney (Editor,

Journal of Computational and Graphical Statistics). A video of the panel discussion is available from the website and is recommended viewing.

I would be remiss if I did not mention the routinely excellent arrangements made by the organizers. The conference provided a wonderful milieu for discussions over coffee breaks, lunches, pub outings and dinner, and judging by the activity, it was a roaring success. Social events included a cocktail reception at the Vienna City Hall (Rathaus), and a traditional Viennese dinner at Heuriger Fuhrgasl-Huber restaurant with generous sampling of Viennese white wine. Everyone was enthusiastic in praise of the conference and I am sure that the next userR! is eagerly anticipated.

Balasubramanian Narasimhan
Stanford University, USA
naras@stanford.edu

Changes in R

by the R Core Team

User-visible changes

- The startup message now prints first the version string and then the copyright notice (to be more similar to R `--version`).
- `save()` by default evaluates promise objects. The old behaviour (to save the promise and its evaluation environment) can be obtained by setting the new argument `'eval.promises'` to `FALSE`. (Note that this does not apply to promises embedded in objects, only to top-level objects.)
- The functions `read.csv()`, `read.csv2()`, `read.delim()`, `read.delim2()` now default their `'comment.char'` argument to `"#"`. (These functions are designed to read files produced by other software, which might use the `#` character inside fields, but are unlikely to use it for comments.)
- The bindings in the base environment/namespace (currently the same thing) are now locked. This means that the values of base functions cannot be changed except via `assignInNamespace()` and similar tricks.
- `[[` on a factor now returns a one-element factor (and not an integer), `as.list()` on a factor returns a list of one-element factors (and not of character vectors), and `unlist()` on a list of

factors returns a factor (and not an integer vector). These changes may affect the results of `sapply()` and `lapply()` applied to factors.

- `mauchly.test()` now returns the W statistic (for comparability with SAS and SPSS), rather than the z (which was accidentally not named in the output)
- `sort(x, decreasing = FALSE, ...)` is now a generic function. This means that `'partial'` is no longer the second argument, and calls which used positional matching may be incorrect: we try to detect them.
- See the section on `'Changes to S4 methods'`: all packages depending on `methods` need to be re-installed.

New features

- `agrep()`, `grep()`, `strwrap()`, `strtrim()`, `substr()` and related functions now coerce arguments which should be character via `as.character()` rather than internally (so method dispatch takes place, e.g. for factors). `chartr()`, `charfold()`, `tolower()` and `toupper()` now coerce their main argument if necessary to a character vector via `as.character()`.

Functions which work element-by-element on character vectors to give a character result now preserve attributes including names, dims and dimnames (as suggested by the Blue

Book p. 144). Such functions include `charfold()`, `chartr()`, `gsub()`, `strtrim()`, `sub()`, `substr()`, `tolower()` and `toupper()`. (Note that coercion of a non-character argument may lose the attributes.)

`agrep(value = TRUE)` preserves names for compatibility with `grep()`.

`nchar()` has always preserved `dims/dimnames` (undocumented before) and now also preserves names.

- `.Deprecated` and `.Defunct` take a new parameter, `msg`, that allows for the specification of the message printed and facilitates deprecation of calling sequences etc.
- `.Fortran()` will map 'name' to lower case, and will work with 'name' containing underscores.
- The default is now `.saveRDS(compress = TRUE)`
- The `::` operator now also works for packages without name spaces that are on the search path.
- `[[` on a list does not duplicate the extracted element unless necessary. (It did not duplicate in other cases, e.g. a pairlist.)
- `argsAnywhere()` works like `args()` on non-exported functions.
- `as.data.frame()` gains a '...' argument.
- Added an `as.data.frame()` method for class "ftable".
- `as.list(<an expression>)` is now handled by internal code and no longer loses attributes such as names.
`as.list(<a list>)` no longer duplicates (unnecessarily).
- `as.POSIX[cl]t` can now convert character strings containing fractional seconds.
- `attach()` can now attach a copy of an environment.
- `available.packages()` and `installed.packages()` gain a 'fields' argument thanks to Seth Falcon.
- `axis.POSIXct()` uses a different algorithm for ranges of 2 to 50 days that will mark days at midnight in the current timezone (even if the graph crosses a DST change).
- `body<-()` and `formals<-()` default to `envir = environment(fun)`, that is they do not by default change the environment. (Previously they changed it to `parent.frame()`.)

- New function `combn(x, m, ...)` for computing on all combinations of size 'm' (for small 'm'!).
- The `cumxxx()` functions now handle logical/integer arguments separately from numeric ones, and so return an integer result where appropriate.
- `data.frame()` has a new argument 'stringsAsFactor'. This and the default for `read.table(as.is=)` are set from the new global option 'stringsAsFactors' via the utility function `default.stringsAsFactors()`.
- `dev.interactive()` now has an optional argument 'orNone'.
- `df()` now has a noncentrality argument 'ncp', based on a contribution by Peter Ruckdeschel.
- `example()` gains an argument 'ask' which defaults to "TRUE when sensible", but the default can be overridden by setting option 'example.ask'.
- `expand.grid()` now has an argument 'KEEP.OUT.ATTRS' which can suppress (the potentially expensive) "out.attrs" attribute. It no longer returns an extraneous 'colnames' attribute.
- The subset and subassign methods for factors now handle factor matrices, and `dim()` can be set on a factor.
- There is now a `format()` method for class "ftable".
- `head(x, n)` and `tail(x, n)` now also work for negative arguments, thanks to Vincent Goulet.
- `head.matrix()` and `tail.matrix()` are no longer hidden, to be used for building `head()` and `tail()` methods for other classes.
- If `help()` finds multiple help files for a given topic, a menu of titles is used to allow interactive choice.
- `help.search()` now rebuilds the database if 'package' specifies a package not in the saved database.
- `hist(*, plot = FALSE)` now warns about unused arguments.
- `history()` gains a 'pattern' argument as suggested by Romain Francois.
- `integer(0)` now prints as that rather than "numeric(0)" (it always deparsed as "integer(0)").

- `interaction(..., drop=TRUE)` now gives the same result as `interaction(...)[,drop=TRUE]` (it used to sometimes give a different order for the levels).
- `lag.plot()` produces a conventional plot (not setting `mfrow`) if only one plot is to be produced.
- `lapply()` does much less copying. Vector `x` are handled without duplication, and other types are coerced via `as.list()`. (As a result, package **boot** runs its examples 4% faster.)
`lapply(<a pairlist>)` now coerces to a list (rather than traverse the pairlist from the beginning for each item).
- `legend()` has new parameters `'box.lwd'` and `'box.lty'`.
- `lines()` gains a simple method for `isoreg()` results.
- `load()` no longer coerces pairlists to lists (which was undocumented, but has been happening since 1998).
- `make.link()` now returns an object of class `"link-glm"`. The GLM families accept an object of this class for their `'link'` argument, which allows user-specified link functions. Also, `quasi()` allows user-specified variance functions.
- `mapply()` uses names more analogously to `lapply()`, e.g..
- `matplot()` now accepts a `'bg'` argument similarly to `plot.default()` etc.
- `median()` is now generic, and its default method uses `mean()` rather than `sum()` and so is more widely applicable (e.g. to dates).
- Dummy functions `memory.size()` and `memory.limit()` are available on Unix-alikes, for people who have not noticed that documentation is Windows-specific.
- `merge()` works more efficiently when there are relatively few matches between the data frames (for example, for 1-1 matching). The order of the result is changed for `'sort = FALSE'`.
- `merge()` now inserts row names as a character column and not a factor: this makes the default sort order more comprehensible.
- Raw, complex and character vectors are now allowed in model frames (there was a previously undocumented restriction to logical, integer and numeric types.). Character vectors in a formula passed to `model.matrix()` are converted to factors and coded accordingly.
- `modifyList()` utility, typically for housekeeping nested lists.
- `x <- 1:20`
`y <- rnorm(x)`
`nls(y ~ A*exp(-x^2/sig))`
no longer returns an unhelpful error message. In this and similar cases, it now tries a wild guess for starting values.
- `Ops.difftime()` now handles unary minus and plus.
- `Ops.Date()` and `Ops.POSIXt()` now allow character arguments (which are coerced to the appropriate class before comparison, for `Ops.POSIXt()` using the current time zone).
- There is a new option (`max.contour.segments = 25000`) which can be raised to allow extremely complex contour lines in `contour()` and `contourLines()`. (PR#9205)
- `options(max.print = N)` where `N` defaults to 99999 now cuts printing of large objects after about `N` entries. `print(x, ..., max = N)` does the same for the default method and those building on `print.default()`.
`options("menu.graphics")` controls if graphical menus should be used when available.
`options("par.ask.default")` allows the default for `par("ask")` to be set for a newly-opened device. (Defaults to `FALSE`, the previous behaviour.)
The way option `("papersize")` is set has been changed. On platforms which support the `LC_PAPER` locale category, the setting is taken first from the `R_PAPERSIZE` environment variable at run time, then from the `LC_PAPER` category (`"letter"` for `_US` and `_CA` locales and `"a4"` otherwise). On other platforms (including Windows and older Unixen), the choice is unchanged.
- `package.skeleton()` gains arguments `'namespace'` and `'code_files'`.
- `par(ask=TRUE)` now only applies to interactive R sessions.
- `parse()` now returns up to `'n'` expressions, rather than fill the expressions vector with `NULL`. (This is now compatible with `S`.)
- The `'version'` argument for `pdf()` is now increased automatically (with a warning) if features which need a higher level are used.
- `pie()` now allows expressions for `'labels'`, and empty slices.

- There is a new `'%.%'` operator for mathematical annotations (plotmath) which draws a centred multiplication dot (a `\cdot` in LaTeX), thanks to Uwe Ligges.
- `predict.lm()` gains a `'pred.var'` argument. (Wishlist PR#8877.)
- `print.summary.{aov,glm,lm,nls}` and `print.{aov,glm}` make use of `naprint()` to report when `na.action` altered the model frame.
- `print.table(T, zero.print=ch)` now also replaces 0 by `ch` when `T` is non-integer with integer entries.
- Recursive `rapply()` which is similar to `lapply` but used recursively and can restrict the classes of elements to which it is applied.
- `r2dtable()` has been moved to package **stats**.
- New function `read.DIF()` to read Data Interchange Format files, and (on Windows) this format from the clipboard.
- New experimental function `readNEWS()` to read R's own "NEWS" file and similarly formatted ones.
- `readLines()` has a new argument `'warn'` to suppress warnings: the default behaviour is still to warn.
- `reg.finalizer()` has a new argument `'onexit'` to parallel the C-level equivalent `R_RegisterFinalizerEx`.
- `rep()` is now a primitive function and under some conditions very much faster: `rep.int()` is still a little faster (but does less). (Because it is primitive there are minor changes to the call semantics: see the help page.)
- The `'row.names'` of a data frame may be stored internally as an integer or character vector. This can result in considerably more compact storage (and more logical row names from `rbind`) when the row.names are `1:nrow(x)`. However, such data frames are not compatible with earlier versions of R: this can be ensured by supplying a character vector as `'row.names'`.

`row.names()` will always return a character vector, but direct access to the attribute may not.

The internal storage of `row.names = 1:n` just records `'n'`, for efficiency with very long vectors.

The `"row.names"` attribute must be a character or integer vector, and this is now enforced by the C code.

- The `"data.frame"` and `"matrix"` methods for `rowsum()` gain an `'na.rm'` argument.
- Experimental support for memory-use profiling via `Rprof()`, `summaryRprof()`, `Rprofmem()` and `tracemem()`.
- `save.image()` [also called by `sys.save.image()` and hence from `q()`] now defaults to saving compressed binary images. To revert to the previous behaviour set option `"save.image.defaults"`: see `?save.image`.
- There is a new primitive `seq.int()` which is slightly more restricted than `seq()` but often very much faster, and new primitives `seq_along()` and `seq_len()` which are faster still.
- `serialize(connection = NULL)` now returns a raw vector (and not a character string). `unserialize()` accepts both old and new formats (and has since 2.3.0).
- `setwd()` now returns the previously current directory (invisibly).
- The function `sort()` is now `sort.int()`, with a new generic function `sort()` which behaves in the same way (except for the order of its argument list) for objects without a class, and relies on the `'[]'` method for objects with a class (unless a specific method has been written, as it has for class `"POSIXlt"`).
- `sort.list()` now implements complex vectors (PR#9039), and how complex numbers are sorted is now documented.
- `spline()` and `splinefun()` now follow `approx[fun]` to have an argument `'ties = mean'` which makes them applicable also when `'x'` has duplicated values.
- `str(x)` does not print the S3 `"class"` attribute when it is the same as `'mode'` (which is printed anyway, possibly abbreviated) and it puts it beside `mode` for atomic objects such as S3 class `"table"`.
- `str(<data.frame>)` now outputs `'data.frame'` instead of ``data.frame``; this may affect some strict (Package) tests.
- `str()` now takes also its defaults for `'vec.len'` and `'digits.d'` from `options('str')` which can be set by the new `strOptions()`.
- `symnum()` has a new argument `'numeric.x'` particularly useful for handling 0/1 data.
- `Sys.getlocale()` and `Sys.setlocale()` support `LC_MESSAGES`, `LC_PAPER` and `LC_MEASUREMENT` if the platform does.

- Sweave has a new options 'pdf.encoding' and 'pdf.version' for its Rweave driver.
- The character vector used by an output `textConnection()` has a locked binding whilst the connection is open.
There is a new function `textConnectionValue()` to retrieve the value of an output `textConnection()`.
- `traceback()` gains a 'max.lines' argument. `.Traceback` is no longer stored in the workspace.
- `warning(immediate. = TRUE)` now applies to `getOption("warn") < 0` and not just `== 0`.
- `warnings()` is now an accessor function for 'last.warning' (which is no longer stored in the workspace) with a `print()` method.
- The internal internet download functions have some new features from libxml 2.6.26.
- There is an option "HTTPUserAgent" to set the User Agent in R download requests etc. Patch from S. Falcon.
- PCRE has been updated to version 6.7.
- The C function `substituteList` now has tail recursion expanded out, so C stack overflow is less likely. (PR#8141, fix by Kevin Hendricks)
- The (somewhat soft) 1023/4 byte limit on command lines is now documented in 'An Introduction to R'.
- The maximum number of open connections has been increased from 50 to 128.
- There is a new manual 'R Internals' on R internal structures plus the former appendices of 'Writing R Extensions'.
- The autoloads introduced at the package reorganization have been almost completely removed: the one that remains is for `ts()`.
- The setting of the various Java configuration variables has been improved to refer to `JAVA_HOME`, and they are now documented in the R-admin manual.
- It is (again) possible to calculate prediction intervals from "lm" objects for the original data frame, now with a warning that the intervals refer to future observations. Weighted intervals have also been implemented, with user-specifiable weights. Warnings are given in cases where the default behaviour might differ from user expectations. See the `?predict.lm` for details.

Changes to S4 Methods

- The default prototype object for S4 classes will have its own internal type in 2.4.0, as opposed to being an empty list (the cause of several errors in the code up to 2.3.1). Note that old binary objects, including class definitions, will be inconsistent with the type, and should be recreated.
- S4 method dispatch has been completely revised to use cached generic functions and to search for the best match among inherited methods. See `?Methods` and <http://developer.r-project.org/howMethodsWork.pdf>
- Objects created from an S4 class are now marked by an internal flag, tested by `isS4()` in R and by macro `IS_S4_OBJECT()` in C. This is an efficient and reliable test, and should replace all earlier heuristic tests.
- Some changes have been made to automatic printing of S4 objects, to make this correspond to a call to `show()`, as per 'Programming with Data'.
- S4 generic and class definitions are now cached when the related package is loaded. This should improve efficiency and also avoid anomalous situations in which a class or generic cannot be found.
- `trace()` now creates a new S4 class for the traced object if required. This allows tracing of user-defined subclasses of "function".

Deprecated & defunct

- The re-named tcltk functions `tkcmd`, `tkfile.tail`, `tkfile.dir`, `tkopen`, `tkclose`, `tkputs`, `tkread` are now defunct.
- Argument 'col' of `bxp()` has been removed: use 'boxfill'.
- Use of NULL as an environment is now an error.
- `postscriptFont()` is defunct: use `Type1Font()`.
- `La.chol()` and `La.chol2inv()` are defunct (they were the same as the default options of `chol()` and `chol2inv()`).
- `La.svd(method = "dgesvd")` is defunct.
- Files `install.R` and `R_PROFILE.R` in packages are now ignored (with a warning).

- The following deprecated command-line options to INSTALL have been removed (use the fields in the DESCRIPTION file instead): `-s` – save `-no-save` `-lazy` `-no-lazy` `-lazy-data` `-no-lazy-data`
- Graphical parameter `'tmag'` is obsolete.
- `mauchley.test()` (package `stats`) is now defunct.
- `symbol.C()` and `symbol.For()` are deprecated. They are required in S for use with `is.loaded()`, but are not so required in R.
- `load()`ing an object saved in one of the formats used prior to R 1.4.0 is deprecated. Such objects should be re-saved in the current format.
- `save(version = 1)` is now deprecated.

C-level facilities

- The convenience function `ScalarLogical` now coerces all non-zero non-NA values to TRUE.
- The vector accessor functions such as `INTEGER`, `REAL` and `SET_VECTOR_ELT` now check that they are called on the correct SEXPTYPE (or at least on a compatible one). See 'Writing R Extensions' for the details and for a stricter test regime.
- It is no longer possible to pass list variables to `.C(DUP = FALSE)`: it would have given rise to obscure garbage collection errors.
- `allocString` is now a macro, so packages using it will need to be reinstalled.
- `R_ParseVector` was returning with `object(s)` protected in the parser if the status was `PARSE_INCOMPLETE` or `PARSE_ERROR`.
- There is a new function `Rf_endEmbeddedR` to properly terminate a session started by `Rf_initEmbeddedR`, and both are now available on Windows as well as on Unix-alikes. These and related functions are declared in a new header `<Rembedded.h>`.

If `R_TempDir` is set when embedded R is initialized it is assumed to point to a valid session temporary directory: see 'Writing R Extensions'.

- There is a new interface allowing one package to make C routines available to C code in other packages. The interface consists of the routines `R_RegisterCCallable` and `R_GetCCallable`. These functions are declared in `<R_ext/Rdynload.h>`. This interface is experimental and subject to change.

In addition, a package can arrange to make use of header files in another (already installed) package via the `'LinkingTo'` field in the DESCRIPTION file: see 'Writing R Extensions'.

Utilities

- R CMD SHLIB now handles (as linker commands) `-L*`, `-l*` and `*.a`.
- R CMD check now:
 - warns if there are non-ASCII characters in the R code (as these will likely be syntax errors in some locale).
 - tests Rd cross-references by default, and tests for (syntactically) valid CITATION metadata.
 - tests that the package can be loaded, and that the package and namespace (if there is one) can each be loaded in startup code (before the standard packages are loaded).
 - tests for empty `'exec'` or `'inst'` directories.
 - checks if `$(FLIBS)` is used when `$(BLAS_LIBS)` is.
 - checks that all packages (except non-S4-using standard packages) used in `::`, `:::`, `library()` and `require()` calls are declared in the DESCRIPTION file, and `'methods'` is declared if S4 classes or methods are set.
 - throws an error if the standard packages `methods` and `stats4` are imported from in the NAMESPACE file and not declared in the DESCRIPTION file.
- The test script produced by `message-Examples.pl` no longer creates objects in the base environment.
- New utilities R CMD Stangle and R CMD Sweave for extracting S/R code from and processing Sweave documentation, respectively.
- The DESCRIPTION file of packages may contain an `'Enhances:'` field.
- An R CMD javareconf script has been added to allow Java configuration to be updated even after R has been installed.

INSTALLATION

- The C function `realpath` (used by `normalizePath()`) is hidden on some systems and we try harder to find it.

- There is a new option `-enable-BLAS-shlib`, which compiles the BLAS into a dynamic library `-lRblas` and links against that. For the pros and cons see the R-admin manual.

The defaults are now `-without-blas` (so you have explicitly to ask for an external BLAS), and `-enable-BLAS-shlib` unless a usable external BLAS is found or on AIX or on MacOS X 10.2 and earlier.

- MacOS X did not like having LSAME in both BLAS and LAPACK libraries, so it is no longer part of the R-internal LAPACK. We now require an external BLAS to provide LSAME: it seems that nowadays all do.
- The configure test for 'whether mixed C/Fortran code can be run' has been improved as on one system that test passed but the Fortran run-time library was broken.
- A precious configure variable `DEFS` can be set to pass defines (e.g. `-DUSE_TYPE_CHECKING_STRICT`) to C code when compiling R.
- There is now a test for visible `__libc_stack_end` on Linux systems (since it is not visible on some recent glibc's built from the sources).
- MacOS X 10.4 and higher now use two-level namespaces, single module in a shared library and allow undefined symbols to be resolved at run-time. This implies that common symbols are now allowed in package libraries. `-enable-BLAS-shlib` is supported for internal BLAS, external BLAS framework and external static BLAS. An external dynamic library BLAS is NOT supported. (But it can be easily used by replacing internal BLAS library file later.) MacOS X < 10.4 does not support `-enable-BLAS-shlib`.
- Dynamic libraries and modules use a flat namespace on MacOS X 10.4 and higher if either Xcode tools don't support dynamic lookup (Xcode < 2.3) or the `FORCE_FLAT_NAMESPACE` environment variable is set. (The latter was introduced temporarily for testing purposes and may go away anytime.)
- configure now defaults to 'run-time linking' on AIX (and AIX < 4.2 is no longer allowed), using `-bexpall` rather than `export/import` files. If this works, it allows R to be built in the same way as other Unix-alikes, including with R as a shared library and with a shared BLAS.
- The `mac.binary` package type now defaults to universal binary. If a repository supports

architecture-specific Mac binaries, they can be requested by using `"mac.binary.xxx"` in `contrib.url()`, where `xxx` is the desired architecture.

Bug fixes

- The name of a Fortran symbol reported to be missing by `.Fortran()` is now the actual name. (What was reported to be an 'entry point' was missing the common leading underscore.)
 - `print()` on a MBCS character string now works properly a character at a time rather than a byte at time. (This does not affect MBCSs like UTF-8 and the Windows DBCSes which have non-ASCII lead bytes and always worked correctly.)
 - `glm()` now recalculates the null deviance whenever there is an offset (even if it is exactly zero to avoid a discontinuity in that case, since the calculations with and without offset are done by different algorithms).
 - Amongst families, `quasi()` accepted an expression for link and no other did. Now all accept an expression which evaluates to a one-element character vector (although e.g. `'logit'` is taken as a name and not an expression).
 - `trace()` now accepts arguments `where=` and `signature=` for the old-style trace (no tracer or `exit`, `edit==FALSE`) and just prints a message on entry. Also the undocumented feature of `where=function` now works for generic functions as well.
 - `callNextMethod()` failed for recursive use when the methods had nonstandard argument lists. Now enforces the semantic rule that the inheritance is fixed when the method containing the `callNextMethod()` is installed. See Details in the documentation.
 - `UseMethod()` looked for the defining environment of 'generic' as if it were the current function, although some functions are generic for methods of a different generic.
- Lookup for S3 methods is confined to functions: previously a non-function 'fun.class' could have masked a function of the same name.
- Line types (`lty`) specified as hex strings were documented not to allow zero, but some devices accepted zero and handled it in a device-dependent way. Now it is an error on all devices. (PR#8914)

- Subassignment for a time series can no longer extend the series: it used to attempt to but failed to adjust the `tsp` attributes. Now `window()` must be used.
- Function `AIC()` in package `stats4` was not dispatching correctly on S4 classes via `logLik()` because of namespace issues.
- Subsetting LANGSXPs could break the call-by-value illusion. (PR#7924) (patch from Kevin Hendricks).
- `parse()` with `n > 1` gave a syntax error if fewer than `n` statements were available.
- `parse()` with `n > 1` gave strange results on some syntax errors. (PR#8815)
- `lag.plot()` now respects graphical parameters for the axes.
- Using a wrong link in `family()` now gives more consistent error messages.
- `sort.list(method="radix")` works on factors again.
- `object.size()` is more accurate for vector objects (it takes into account the smaller header and also the fixed sizes used in the node classes for small vector objects).
- `addmargins(T, ...)` now returns a "table" when 'T' is a "table", as its help page has always suggested.
- `remove()` now explicitly precludes removing variables from `baseenv()` and throws an error (this was previously ignored).
- Saving the workspace at the end of a session now works as has long been intended, that is it is saved only if something has been added/deleted/changed during the current session.
- The search for bindings in `<<-`, `->` and `assign(inherits=TRUE)` was omitting the base package, although this was not documented. Now the base package is included (but most bindings there are locked).
- `dweibull(0, shape)` was NaN not Inf for `shape < 1`. Also, the help for `dgamma` and `dweibull` gave support as `x > 0`, but returned non-zero values for `x = 0`. (PR#9080)
- Subsetting arrays no longer preserves attributes (it was removed for matrices in 1998).
- The "factor" method of `as.character()` no longer maps level "NA" to "<NA>" (a legacy of before there were NA character strings).
- `terms(keep.order=TRUE)` was not returning a valid "order" attribute.
- The DLL registration code was not freeing .External symbols.
- The internet download routines expected URLs of less than 4096 bytes, but did not check. Now this is checked, and `http://` URLs are allowed to be up to 40960 bytes.
- `parse(n=-1)` threw a stack-imbalance error, and `parse(n=3)` did not cope correctly with EOF during input.
- Zero-column data frames had no names (rather than `character(0)`).
- `by()` and `acf()` could get confused when they used very long expressions as names.
- `residuals(<glm object>, type="working")` was NA for cases with zero weight (whereas they are well-defined even though the case was not used during the fitting) and the actual value is now returned. This allows residuals to be computed from fits with `'y = FALSE'`.
The residuals in a fitted "glm" object are computed more accurately: the previous formula was subject to cancellation.
- `loess()` now checks the validity of its 'control' argument.
- `rownames(<0-row matrix>, do.NULL=FALSE)` was wrong. (PR#9136)
- `apply()` now works as documented when applied over 2 or more margins with one of zero extent. (It used to drop dimensions.)
- `head()` and `tail()` now also work row-wise for "table" and "ftable" objects.
- `NextMethod()` could throw an error/crash if called from a method that was called directly rather than from a generic (so `.Method` was unset).
- `order(x, na.last = NA)` failed for a zero-length `x`.
- `grep(pat, x, value = TRUE, perl = L)` preserved names for `L == TRUE && !is.na(pat)` but not otherwise. Now it always does.
- `[rc]bind()` now find registered methods and not just visible ones.
- Printing a factor no longer ignores attributes such as names and `dim/dimnames`.
- Command-line arguments after `-encoding` were ignored.

- The check for impossible confidence levels was off by one in `wilcox.test` (PR#8557)
- `[[` on an environment could create aliases. (PR#8457)
- `pt()` with a very small (or zero) non-centrality parameter could give an unduly stringent warning about 'full precision was not achieved'. (PR#9171)
- `writeChar()` could segfault if 'nchars' was given silly values.
- `qt()` and `rt()` did not work for vector 'ncp', and `qt()` did not work for negative 'ncp'.
- `ns()` failed to work correctly when 'x' was of length one.
- `identical()` ignored tags on pairlists (including names of attributes) and required an identical ordering for attribute values in their pairlists. Now names are compared on pairlists, and attribute sets are treated as unordered.
- If they were unused arguments supplied to a closure, only the first non-empty one was reported, despite the message.
Unmatched empty arguments (such as `f(1,,)` for a function of one argument) were ignored. They are now an error.
- Calling a builtin with empty arguments used to silently remove them (and this was undocumented). Now this is an error unless builtin is `c()` or `list()` or there are only trailing empty arguments, when it is a warning (for the time being: this will be made an error in R 2.5.0).
- `install.packages()` ignored 'configure.args' if the vector was unnamed.
- `biplot()` now works if there are missing values in the data.
- `biplot()` now passes `par()` values to all four axes (not just those on sides 1 and 2).
- `[.acf` now handles an empty first index.
- Deparsing uses backticks more consistently to quote non-syntactic names.
- Assigning to the symbol in a `for()` loop with a list/expression/pairlist index could alter the index. Now the loop variable is explicitly read-only. (PR#9216)
- Using `old.packages()` (and hence `update.packages()`) on an empty (or non-existent) library failed with an obscure message.
- `plot.xy()` could segfault if supplied with an invalid 'col' argument. (PR#9221)
- `menu()` with `graphics=TRUE` attempted to use Tcl/Tk on unix even if `DISPLAY` was not set (in which case Tk is not available and so the attempt is bound to fail).
- The `print()` method for 'dist' objects prints a matrix even for `n = 2`.
- The `cum<xxx>` functions were missing some PROTECTs and so could segfault on long vectors (especially with names or where coercion to numeric occurred).
- The `X11()` device no longer produces (apparently spurious) 'BadWindow (invalid Window parameter)' warnings when run from Rcmdr.
- `legend()` assumed that widths and heights of strings were positive, which they need not be in user coordinates with reversed axes. (In part, PR#9236)
- The `plot()` methods for "profile.nls" objects could get confused if 'which' had been used in the `profile()` call. (PR#9231)
- `boxplot()` did not passed named arguments (except graphics parameters) to `bxp()` as documented. (PR#9183)
- Only genuinely empty statements act as 'return' in the browser, not say those starting with a comment char. (PR#9063)
- `summary.mlm()` incorrectly used accessor functions to fake an "lm" object. (PR#9191)
- `prettyNum()` was not preserving attributes, despite being explicitly documented to. (PR#8695)
- It was previously undocumented what happened if a graphical parameter was passed in both '...' and 'pars' to `boxplot()` and `bxp()`, and they behaved differently. Now those passed in '...' have precedence in both cases.
- A failed subassignment could leave behind an object '*tmp*'. The fix also sometimes gives better error messages.
- Using `SIGUSR1` on Unix now always terminates a session, and no longer is caught by browser contexts and restarts (such as `try()`).
- In the **graphics** package, in-line 'font=5' was being ignored (report by Tom Cook).
- `nls()` looked for non-parameter arguments in a function call in the wrong scope (from the body of nls).

- Printing of complex numbers could misbehave when one of the parts was large (so scientific notation was used) and the other was so much smaller that it had no significant digits and should have been printed as zero (e.g. 1e80+3e44i).
- Using `install.packages` with `type="mac.binary"` and target path starting with `failed` with a cryptic message while unpacking.

- `getwd()` now works correctly when the working directory is unavailable (e.g. unreadable).
- The alternative hypothesis in `wilcox.test()` was labelled by an unexplained quantity `'mu'` which is now spelled out.

The alternative hypothesis in `ks.test()` is clearer both in the documentation and in the result. (PR#5360)

Changes on CRAN

by Kurt Hornik

New contributed packages

AdaptFit Adaptive semiparametric regression. Fits semiparametric regression models with spatially adaptive penalized splines. By Tatyana Krivobokova.

CompetingRiskFrailty Competing risk model with frailties for right censored survival data. Offers fitting of smooth varying coefficients in a competing risks modeling of hazards as well as estimating of the frailties (or unobserved heterogeneities) for clustered observations. Non-parametric penalized spline (p-spline) fitting of smooth covariates effects is proposed. As a spline basis truncated polynomial functions are chosen. The frailties are also fitted (via the EM algorithm) in a flexible way using a penalized mixture of Gamma distributions. By Pavel Khomski.

DPpackage Semiparametric Bayesian analysis using Dirichlet process priors. Contains functions for posterior analysis for a number of semiparametric statistical models. Simulation is done in compiled Fortran. By Alejandro Jara Vallejos.

FLEDA Exploratory Data Analysis for **FLR**. Develops several procedures to explore fisheries data. By Ernesto Jardim and Manuela Azevedo.

Flury Data sets from Bernard Flury (1997), "A First Course in Multivariate Statistics", Springer NY. By Bernard Flury.

GAMBoost Routines for fitting generalized additive models by likelihood based, using penalized B-splines. By Harald Binder.

GeneCycle Identification of periodically expressed genes. Implements the approaches of Wichert

et al. (2004) and Ahdesmaki et al. (2005) for detecting periodically expressed genes from gene expression time series data. By Miika Ahdesmaki, Konstantinos Fokianos, and Korbinian Strimmer.

GeneNet Modeling and inferring gene networks. A package for analyzing gene expression (time series) data with focus on the inference of gene networks. In particular, it implements the methods of Schaefer and Strimmer (2005a,b,c) and Opgen-Rhein and Strimmer (2006) for learning large-scale gene association networks. For plotting the gene networks, **GeneNet** uses the **graph** and **Rgraphviz** packages, available from <http://www.bioconductor.org>. By Rainer Opgen-Rhein, Juliane Schaefer, and Korbinian Strimmer.

HH Support software for the book "Statistical Analysis and Data Display" by Richard M. Heiberger and Burt Holland (Springer, ISBN 0-387-40270-5). This contemporary presentation of statistical methods features extensive use of graphical displays for exploring data and for displaying the analysis. The authors demonstrate how to analyze data—showing code, graphics, and accompanying computer listings—for all the methods they cover. They emphasize how to construct and interpret graphs, discuss principles of graphical design, and show how accompanying traditional tabular results are used to confirm the visual impressions derived directly from the graphs. Many of the graphical formats are novel and appear for the first time in print. By Richard M. Heiberger.

JGR JGR — Java Gui for R. By Markus Helbig.

JavaGD Java Graphics Device, routing all graphics commands to a Java program. The actual functionality of the JavaGD depends on the Java-side implementation. Simple AWT and Swing implementations are included. By Simon Urbanek.