```
    test<-fitted(T)
    disease<-(test+resid(T,type="response"))
    disease<-disease*weights(T)
    if (max(abs(disease %% 1))>0.01)
        warning("Y values suspiciously far
                          from integers")

    TT<-rev(sort(unique(test)))
    DD<-table(-test,disease)

    sens<-cumsum(DD[,2])/sum(DD[,2])
    mspec<-cumsum(DD[,1])/sum(DD[,1])

    new("ROC",sens=sens, mspec=mspec,
              test=TT,call=sys.call())
  }
)
```

Finally, a method for `identify` shows one additional feature of `setGeneric`. The `signature` argument to `setGeneric` specifies which arguments are permitted in the signature of a method and thus are used for method dispatch. Method dispatch for `identify` will be based only on the first argument, which saves having to specify a `"missing"` second argument in the method.

```
setGeneric("identify",signature=c("x"))

setMethod("identify", "ROC",
  function(x, labels=NULL,
                  ...,digits=1){
    if (is.null(labels))
      labels<-round(x@test, digits)
    identify(x@mspec, x@sens,
             labels=labels,...)
    }
)
```

Discussion

Creating a simple class and methods requires very similar code whether the S3 or S4 system is used, and a similar incremental design strategy is possible. The S3 and S4 method system can coexist peacefully, even when S4 methods need to be defined for a function that already has S3 methods.

This example has not used inheritance, where the S3 and S4 systems differ more dramatically. Judging from the available examples of S4 classes, inheritance seems most useful in defining data structures, rather than objects representing statistical calculations. This may be because inheritance extends a class by creating a special case, but statisticians more often extend a class by creating a more general case. Reusing code from, say, linear models in creating generalised linear models is more an example of delegation than inheritance. It is not that a generalised linear model "is" a linear model, more that it "has" a linear model (from the last iteration of iteratively reweighted least squares) associated with it.

Thomas Lumley
Department of Biostatistics
University of Washington, Seattle

Changes in R

by the R Core Team

New features in 1.9.1

- `as.Date()` now has a method for "POSIXlt" objects.
- `mean()` has a method for "difftime" objects and so `summary()` works for such objects.
- `legend()` has a new argument `pt.cex`.
- `plot.ts()` has more arguments, particularly `yax.flip`.
- `heatmap()` has a new `keep.dendro` argument.
- The default `barplot` method now handles vectors and 1-d arrays (e.g., obtained by `table()`) the same, and uses grey instead of heat color palettes in these cases. (Also fixes PR#6776.)
- `nls()` now looks for variables and functions in its formula in the environment of the formula before the search path, in the same way `lm()` etc look for variables in their formulae.

User-visible changes in 1.9.0

- Underscore _ is now allowed in syntactically valid names, and `make.names()` no longer changes underscores. Very old code that makes use of underscore for assignment may now give confusing error messages.
- Package 'base' has been split into packages 'base', 'graphics', 'stats' and 'utils'. All four are loaded in a default installation, but the separation allows a 'lean and mean' version of R to be used for tasks such as building indices.

  Packages ctest, eda, modreg, mva, nls, stepfun and ts have been merged into stats, and lqs has

been returned to MASS. In all cases a stub has been left that will issue a warning and ensure that the appropriate new home is loaded. All the time series datasets have been moved to package stats. `Sweave` has been moved to utils.

Package mle has been moved to stats4 which will become the central place for statistical S4 classes and methods distributed with base R. Package mle remains as a stub.

Users may notice that code in .Rprofile is run with only the new base loaded and so functions may now not be found. For example, `ps.options(horizontal = TRUE)` should be preceded by `library(graphics)` or called as `graphics::ps.options` or, better, set as a hook – see `?setHook`.

- There has been a concerted effort to speed up the startup of an R session: it now takes about 2/3rds of the time of 1.8.1.

- A warning is issued at startup in a UTF-8 locale, as currently R only supports single-byte encodings.

New features in 1.9.0

- `$`, `$<-`, `[[`, `[[<-` can be applied to environments. Only character arguments are allowed and no partial matching is done. The semantics are basically that of `get/assign` to the environment with `inherits=FALSE`.

- There are now `print()` and `[` methods for "acf" objects.

- `aov()` will now handle singular `Error()` models, with a warning.

- `arima()` allows models with no free parameters to be fitted (to find log-likelihood and AIC values, thanks to Rob Hyndman).

- `array()` and `matrix()` now allow 0-length 'data' arguments for compatibility with S.

- `as.data.frame()` now has a method for arrays.

- `as.matrix.data.frame()` now coerces an all-logical data frame to a logical matrix.

- New function `assignInNamespace()` paralleling `fixInNamespace`.

- There is a new function `contourLines()` to produce contour lines (but not draw anything). This makes the CRAN package clines (with its `clines()` function) redundant.

- `D()`, `deriv()`, etc now also differentiate `asin()`, `acos()`, `atan()`, (thanks to a contribution of Kasper Kristensen).

- The `package` argument to `data()` is no longer allowed to be a (unquoted) name and so can be a variable name or a quoted character string.

- There is a new class "Date" to represent dates (without times) plus many utility functions similar to those for date-times. See `?Date`.

- Deparsing (including using `dump()` and `dput()`) an integer vector now wraps it in `as.integer()` so it will be source()d correctly. (Related to PR#4361.)

- `.Deprecated()` has a new argument `package` which is used in the warning message for non-base packages.

- The `print()` method for "difftime" objects now handles arrays.

- `dir.create()` is now an internal function (rather than a call to mkdir) on Unix as well as on Windows. There is now an option to suppress warnings from mkdir, which may or may not have been wanted.

- `dist()` has a new method to calculate Minkowski distances.

- `expand.grid()` returns appropriate array dimensions and dimnames in the attribute `"out.attrs"`, and this is used by the `predict()` method for loess to return a suitable array.

- `factanal()`, `loess()` and `princomp()` now explicitly check for numerical inputs; they might have silently coded factor variables in formulae.

- New functions `factorial()` defined as `gamma(x+1)` and for S-PLUS compatibility, `lfactorial()` defined as `lgamma(x+1)`.

- `findInterval(x, v)` now allows +/-Inf values, and NAs in `x`.

- `formula.default()` now looks for a `"terms"` component before a `formula` argument in the saved call: the component will have '.' expanded and probably will have the original environment set as its environment. And what it does is now documented.

- `glm()` arguments `etastart` and `mustart` are now evaluated via the model frame in the same way as `subset` and `weights`.

- Functions `grep()`, `regexpr()`, `sub()` and `gsub()` now coerce their arguments to character, rather than give an error.

  The `perl=TRUE` argument now uses character tables prepared for the locale currently in use each time it is used, rather than those of the C locale.

- New functions `head()` and `tail()` in package 'utils'. (Based on a contribution by Patrick Burns.)

- `legend()` has a new argument 'text.col'.

- `methods(class=)` now checks for a matching generic, and so no longer returns methods for non-visible generics (and eliminates various mismatches).

- A new function `mget()` will retrieve multiple values from an environment.

- `model.frame()` methods, for example those for `"lm"` and `"glm"`, pass relevant parts of `...` onto the default method. (This has long been documented but not done.) The default method is now able to cope with model classes such as `"lqs"` and `"ppr"`.

- `nls()` and `ppr()` have a `model` argument to allow the model frame to be returned as part of the fitted object.

- `"POSIXct"` objects can now have a `"tzone"` attribute that determines how they will be converted and printed. This means that date-time objects which have a timezone specified will generally be regarded as in their original time zone.

- `postscript()` device output has been modified to work around rounding errors in low-precision calculations in gs >= 8.11. (PR#5285, which is not a bug in R.)

  It is now documented how to use other Computer Modern fonts, for example italic rather than slanted.

- `ppr()` now fully supports categorical explanatory variables,

  `ppr()` is now interruptible at suitable places in the underlying FORTRAN code.

- `princomp()` now warns if both `x` and `covmat` are supplied, and returns scores only if the centring used is known.

- `psigamma(x, deriv=0)`, a new function generalizes, `digamma()` etc. All these (psigamma, digamma, trigamma,...) now also work for x < 0.

- `pchisq( , ncp > 0)` and hence `qchisq()` now work with much higher values of `ncp`; it has become much more accurate in the left tail.

- `read.table()` now allows embedded newlines in quoted fields. (PR#4555)

- `rep.default(`0-length-vector, `length.out=n)` now gives a vector of length `n` and not length 0, for compatibility with S.

  If both `each` and `length.out` have been specified, it now recycles rather than fills with NAs for S compatibility.

  If both `times` and `length.out` have been specified, `times` is now ignored for S compatibility. (Previously padding with NAs was used.)

  The `"POSIXct"` and `"POSIXlt"` methods for `rep()` now pass `...` on to the default method (as expected by PR#5818).

- `rgb2hsv()` is new, an R interface the C API function with the same name.

- User hooks can be set for `onLoad`, `library`, `detach` and `onUnload` of packages/namespaces: see `?setHook`.

- `save()` default arguments can now be set using option `"save.defaults"`, which is also used by `save.image()` if option `"save.image.defaults"` is not present.

- New function `shQuote()` to quote strings to be passed to OS shells.

- `sink()` now has a `split=` argument to direct output to both the sink and the current output connection.

- `split.screen()` now works for multiple devices at once.

- On some OSes (including Windows and those using glibc) `strptime()` did not validate dates correctly, so we have added extra code to do so. However, this cannot correct scanning errors in the OS's strptime (although we have been able to work around these on Windows). Some examples are now tested for during configuration.

- `strsplit()` now has `fixed` and `perl` arguments and `split=""` is optimized.

- `subset()` now allows a `drop` argument which is passed on to the indexing method for data frames.

- `termplot()` has an option to smooth the partial residuals.

- `varimax()` and `promax()` add class `"loadings"` to their loadings component.

- Model fits now add a "dataClasses" attribute to the terms, which can be used to check that the variables supplied for prediction are of the same type as those used for fitting. (It is currently used by predict() methods for classes "lm", "mlm", "glm" and "ppr", as well as methods in packages MASS, rpart and tree.)

- New command-line argument --max-ppsize allows the size of the pointer protection stack to be set higher than the previous limit of 10000.

- The fonts on an X11() device (also jpeg() and png() on Unix) can be specified by a new argument 'fonts' defaulting to the value of a new option "X11fonts".

- New functions in the tools package: pkgDepends, getDepList and installFoundDepends. These provide functionality for assessing dependencies and the availability of them (either locally or from online repositories).

- The parsed contents of a NAMESPACE file are now stored at installation and if available used to speed loading the package, so packages with namespaces should be reinstalled.

- Argument asp, although not a graphics parameter, is accepted in the ... of graphics functions without a warning. It now works as expected in contour().

- Package stats4 exports S4 generics for AIC() and BIC().

- The Mac OS X version now produces an R framework for easier linking of R into other programs. As a result, R.app is now relocatable.

- Added experimental support for conditionals in NAMESPACE files.

- Added as.list.environment to coerce environments to lists (efficiently).

- New function addmargins() in the stats package to add marginal summaries to tables, e.g. row and column totals. (Based on a contribution by Bendix Carstensen.)

- dendrogam edge and node labels can now be expressions (to be plotted via stats:::plotNode called from plot.dendrogram). The diamond frames around edge labels are more nicely scaled horizontally.

- Methods defined in the methods package can now include default expressions for arguments. If these arguments are missing in the call, the defaults in the selected method will override a default in the generic. See ?setMethod.

- Changes to package 'grid':

  - Renamed push/pop.viewport() to push/popViewport().

  - Added upViewport(), downViewport(), and seekViewport() to allow creation and navigation of viewport tree (rather than just viewport stack).

  - Added id and id.lengths arguments to grid.polygon() to allow multiple polygons within single grid.polygon() call.

  - Added vpList(), vpStack(), vpTree(), and current.vpTree() to allow creation of viewport "bundles" that may be pushed at once (lists are pushed in parallel, stacks in series).

    current.vpTree() returns the current viewport tree.

  - Added vpPath() to allow specification of viewport path in downViewport() and seekViewport().

    See ?viewports for an example of its use.

    NOTE: it is also possible to specify a path directly, e.g., something like vp1::vp2, but this is only advised for interactive use (in case I decide to change the separator :: in later versions).

  - Added just argument to grid.layout() to allow justification of layout relative to parent viewport **IF** the layout is not the same size as the viewport. There's an example in help(grid.layout).

  - Allowed the "vp" slot in a grob to be a viewport name or a vpPath. The interpretation of these new alternatives is to call downViewport() with the name or vpPath before drawing the grob and upViewport() the appropriate amount after drawing the grob. Here's an example of the possible usage:

```
pushViewport(viewport(w=.5, h=.5,
                      name="A"))
grid.rect()
pushViewport(viewport(w=.5, h=.5,
                      name="B"))
grid.rect(gp=gpar(col="grey"))
upViewport(2)
grid.rect(vp="A",
          gp=gpar(fill="red"))
grid.rect(vp=vpPath("A", "B"),
          gp=gpar(fill="blue"))
```

– Added `engine.display.list()` function. This allows the user to tell grid NOT to use the graphics engine display list and to handle ALL redraws using its own display list (including redraws after device resizes and copies).

  This provides a way to avoid some of the problems with resizing a device when you have used `grid.convert()`, or the gridBase package, or even base functions such as `legend()`.

  There is a document discussing the use of display lists in grid on the grid web site http://www.stat.auckland.ac.nz/~paul/grid/grid.html

– Changed the implementation of grob objects. They are no longer implemented as external references. They are now regular R objects which copy-by-value. This means that they can be saved/loaded like normal R objects. In order to retain some existing grob behaviour, the following changes were necessary:

  * grobs all now have a `"name"` slot. The grob name is used to uniquely identify a `"drawn"` grob (i.e., a grob on the display list).
  * `grid.edit()` and `grid.pack()` now take a grob name as the firs argument instead of a grob. (Actually, they take a gPath; see below)
  * the `"grobwidth"` and `"grobheight"` units take either a grob OR a grob name (actually a gPath; see below). Only in the latter case will the unit be updated if the grob "pointed to" is modified.

  In addition, the following features are now possible with grobs:

  * grobs now `save()/load()` like any normal R object.
  * many `grid.*()` functions now have a `*Grob()` counterpart. The `grid.*()` version is used for its side-effect of drawing something or modifying something which has been drawn; the `*Grob()` version is used for its return value, which is a grob. This makes it more convenient to just work with grob objects without producing any graphical output (by using the `*Grob()` functions).
  * there is a `gTree object` (derived from grob), which is a grob that can have children. A gTree also has a `"childrenvp"` slot which is a viewport which is pushed and then "up"ed before the children are drawn; this allows the children of a `gTree` to place themselves somewhere in the viewports specified in the `childrenvp` by having a `vpPath` in their `vp` slot.
  * there is a `gPath` object, which is essentially a concatenation of grob names. This is used to specify the child of (a child of ...) a `gTree`.
  * there is a new API for creating/accessing/modifying grob objects: `grid.add()`, `grid.remove()`, `grid.edit()`, `grid.get()` (and their `*Grob()` counterparts can be used to add, remove, edit, or extract a grob or the child of a gTree. NOTE: the new `grid.edit()` API is incompatible with the previous version.

  – Added `stringWidth()`, `stringHeight()`, `grobWidth()`, and `grobHeight()` convenience functions (they produce "strwidth", "strheight", "grobwidth", and "grobheight" unit objects, respectively).

  – Allowed viewports to turn off clipping altogether. Possible settings for viewport clip arg are now:

  "on" clip to the viewport (was `TRUE`)

  "inherit" clip to whatever parent says (was `FALSE`)

  "off" turn off clipping

  Still accept logical values (and NA maps to "off")

- `R CMD check` now runs the (Rd) examples with default RNGkind (uniform & normal) and codeset.seed(1). `example(*, setRNG = TRUE)` does the same.

- `undoc()` in package 'tools' has a new default of 'use.values = NULL' which produces a warning whenever the default values of function arguments differ between documentation and code. Note that this affects `R CMD check` as well.

- Testing examples via `massage-examples.pl` (as used by `R CMD check`) now restores the search path after every help file.

- `checkS3methods()` in package 'tools' now also looks for generics in the loaded namespaces/packages listed in the Depends fields of the package's DESCRIPTION file when testing an installed package.

- The DESCRIPTION file of packages may contain a 'Suggests:' field for packages that are used only in examples or vignettes.

- Added an option to `package.dependencies()` to handle the 'Suggests' levels of dependencies.
- Vignette dependencies can now be checked and obtained via `vignetteDepends`.
- Option `"repositories"` to list URLs for package repositories added.
- `package.description()` has been replaced by `packageDescription()`.
- `R CMD INSTALL/build` now skip Subversion's `.svn` directories as well as CVS directories.
- `arraySubscript` and `vectorSubscript` take a new argument which is a function pointer that provides access to character strings (such as the names vector) rather than assuming these are passed in.
- `R_CheckUserInterrupt` is now described in 'Writing R Extensions' and there is a new equivalent subroutine `rchkusr` for calling from FORTRAN code.
- `hsv2rgb` and `rgb2hsv` are newly in the C API.
- `Salloc` and `Srealloc` are provided in `S.h` as wrappers for `S_alloc` and `S_realloc`, since current S versions use these forms.
- The type used for vector lengths is now `R_len_t` rather than `int`, to allow for a future change.
- The internal header nmath/dpq.h has slightly improved macros `R_DT_val()` and `R_DT_Cval()`, a new `R_D_LExp()` and improved `R_DT_log()` and `R_DT_Clog()`; this improves accuracy in several [dpq]-functions for extreme arguments.
- `print.coefmat()` is defunct, replaced by `printCoefmat()`.
- `codes()` and `codes<-()` are defunct.
- `anovalist.lm` (replaced in 1.2.0) is now defunct.
- `glm.fit.null()`, `lm.fit.null()` and `lm.wfit.null()` are defunct.
- `print.atomic()` is defunct.
- The command-line arguments `--nsize` and `--vsize` are no longer recognized as synonyms for `--min-nsize` and `--min-vsize` (which replaced them in 1.2.0).
- Unnecessary methods `{coef.{g}lm` and `fitted.{g}lm` have been removed: they were each identical to the default method.
- `La.eigen()` is deprecated now `eigen()` uses LAPACK by default.
- `tetragamma()` and `pentagamma()` are deprecated, since they are equivalent to `psigamma(, deriv=2)` and `psigamma(, deriv=3)`.
- `LTRUE/LFALSE` in `Rmath.h` have been removed: they were deprecated in 1.2.0.
- `package.contents()` and `package.description()` have been deprecated.
- The defaults for configure are now `--without-zlib` `--without-bzlib` `--without-pcre`.

  The included PCRE sources have been updated to version 4.5 and PCRE >= 4.0 is now required if `--with-pcre` is used.

  The included zlib sources have been updated to 1.2.1, and this is now required if `--with-zlib` is used.
- configure no longer lists bzip2 and PCRE as 'additional capabilities' as all builds of R have had them since 1.7.0.
- `--with-blasgoto=` to use K. Goto's optimized BLAS will now work.

The above lists only new features, see the 'NEWS' file in the R distribution or on the R homepage for a list of bug fixes.

Changes on CRAN

by Kurt Hornik

New contributed packages

AlgDesign Algorithmic experimental designs. Calculates exact and approximate theory experimental designs for D, A, and I criteria. Very large designs may be created. Experimental designs may be blocked or blocked designs created from a candidate list, using several criteria. The blocking can be done when whole and within plot factors interact. By Bob Wheeler.

BradleyTerry Specify and fit the Bradley-Terry model and structured versions. By David Firth.

BsMD Bayes screening and model discrimination