Changes in R

From version 3.3.3 to version 3.4.0 patched

by R Core Team

CHANGES IN R 3.4.0 patched

INSTALLATION on a UNIX-ALIKE

- The deprecated support for PCRE versions older than 8.20 has been removed.

BUG FIXES

- `getParseData()` gave incorrect column information when code contained multi-byte characters. (PR#17254)
- Asking for help using expressions like `?stats::cor()` did not work. (PR#17250)
- `readRDS(url(....))` now works too.
- `installed.packages()` on Windows would fail when `file.mtime()` (?) returned NA.
- `R CMD Sweave` again returns '`status = 0`' on successful completion.
- Vignettes listed in '.Rbuildignore' were not being ignored properly. (PR#17246)
- `file.mtime()` no longer returns NA on Windows when the file or directory is being used by another process.
- `R CMD INSTALL` Windows .zip file obeys `--lock` and `--pkglock` flags.
- (Windows only) The `choose.files()` function could return incorrect results when called with `multi = FALSE`. (PR#17270)

CHANGES IN R 3.4.0

SIGNIFICANT USER-VISIBLE CHANGES

- (Unix-alike) The default methods for `download.file()` and `url()` now choose `"libcurl"` except for '`file://`' URLs. There will be small changes in the format and wording of messages, including in rare cases if an issue is a warning or an error. For example, when HTTP re-direction occurs, some messages refer to the final URL rather than the specified one.

  Those who use proxies should check that their settings are compatible (see `?download.file`: the most commonly used forms work for both `"internal"` and `"libcurl"`).

- `table()` has been amended to be more internally consistent and become back compatible to R $\leq$ 2.7.2 again. Consequently, `table(1:2,exclude = NULL)` no longer contains a zero count for `<NA>`, but `useNA = "always"` continues to do so.

- `summary.default()` no longer rounds, but its print method does resulting in less extraneous rounding, notably of numbers in the ten thousands.

- `factor(x,exclude = L)` behaves more rationally when `x` or `L` are character vectors. Further, `exclude = <factor>` now behaves as documented for long.

- Arithmetic, logic (`&`, `|`) and comparison (aka 'relational', e.g., `<`, `==`) operations with arrays now behave consistently, notably for arrays of length zero.

  Arithmetic between length-1 arrays and longer non-arrays had silently dropped the array attributes and recycled. This now gives a warning and will signal an error in the future, as it has always for logic and comparison operations in these cases (e.g., compare `matrix(1,1) + 2:3` and `matrix(1,1) <2:3`).

- The JIT ('Just In Time') byte-code compiler is now enabled by default at its level 3. This means functions will be compiled on first or second use and top-level loops will be compiled and then run. (Thanks to Tomas Kalibera for extensive work to make this possible.)

  For now, the compiler will not compile code containing explicit calls to `browser()`: this is to support single stepping from the `browser()` call.

  JIT compilation can be disabled for the rest of the session using `compiler::enableJIT(0)` or by setting environment variable `R_ENABLE_JIT` to `0`.

- `xtabs()` works more consistently with `NA`s, also in its result no longer setting them to `0`. Further, a new logical option `addNA` allows to count `NA`s where appropriate. Additionally, for the case `sparse = TRUE`, the result's `dimnames` are identical to the default case's.

- Matrix products now consistently bypass BLAS when the inputs have `NaN/Inf` values. Performance of the check of inputs has been improved. Performance when BLAS is used is improved for matrix/vector and vector/matrix multiplication (DGEMV is now used instead of DGEMM).

  One can now choose from alternative matrix product implementations *via* `options(matprod = )`. The `"internal"` implementation is not optimized for speed but consistent in precision with other summations in R (using `long double` accumulators where available). `"blas"` calls BLAS directly for best speed, but usually with undefined behavior for inputs with `NaN/Inf`.

- `factor()` now uses `order()` to sort its levels, not `sort.list()`. This makes `factor()` support custom vector-like objects if methods for the appropriate generics are defined. This change has the side effect of making `factor()` succeed on empty or length-one non-atomic vector(-like) types (e.g., list), where it failed before.

NEW FEATURES

- User errors such as `integrate(f,0:1,2)` are now caught.

- Add `signature` argument to `debug()`, `debugonce()`, `undebug()` and `isdebugged()` for more conveniently debugging S3 and S4 methods. (Based on a patch by Gabe Becker.)

- Add `utils::debugcall()` and `utils::undebugcall()` for debugging the function that would be called by evaluating the given expression. When the call is to an S4 generic or standard S3 generic, `debugcall()` debugs the method that would be dispatched. A number of internal utilities were added to support this, most notably `utils::isS3stdGeneric()`. (Based on a patch by Gabe Becker.)

- Add `utils::strcapture()`. Given a character vector and a regular expression containing capture expressions, `strcapture()` will extract the captured tokens into a tabular data structure, typically a `data.frame`.

- `str()` and `strOptions()` get a new option `drop.deparse.attr` with improved but *changed* default behaviour for expressions. For `expression` objects `x`, `str(x)` now may remove extraneous white space and truncate long lines.

- `str(<looooooooong_string>)` is no longer very slow; inspired by Mikko Korpela's proposal in PR#16527.
- `str(x)`'s default method is more "accurate" and hence somewhat more generous in displaying character vectors; this will occasionally change R outputs (and need changes to some '*.Rout(.save)' files).
  For a classed integer vector such as `x <-xtabs(~ c(1,9,9,9))`, `str(x)` now shows both the class and `"int"`, instead of only the latter.
- `isSymmetric(m)` is much faster for large asymmetric matrices `m` *via* pre-tests and a new option `tol1` (with which strict back compatibility is possible but not the default).
- The result of `eigen()` now is of class `"eigen"` in the default case when eigenvectors are computed.
- Zero-length date and date-time objects (of classes `"POSIX[cl]?t"`) now `print()` "recognizably".
- `xy.coords()` and `xyz.coords()` get a new `setLab` option.
- The `method` argument of `sort.list()`, `order()` and `sort.int()` gains an `"auto"` option (the default) which should behave the same as before when `method` was not supplied.
- `stopifnot(E,..)` now reports differences when `E` is a call to `all.equal()` and that is not true.
- `boxplot(<formula>,*)` gain optional arguments `drop`, `sep`, and `lex.order` to pass to `split.default()` which itself gains an argument `lex.order` to pass to `interaction()` for more flexibility.
- The `plot()` method for `ppr()` has enhanced default labels (`xmin` and `main`).
- `sample.int()` gains an explicit `useHash` option (with a back compatible default).
- `identical()` gains an `ignore.srcref` option which drops `"srcref"` and similar attributes when true (as by default).
- `diag(x,nrow = n)` now preserves `typeof(x)`, also for logical, integer and raw `x` (and as previously for complex and numeric).
- `smooth.spline()` now allows direct specification of `lambda`, gets a `hatvalues()` method and keeps `tol` in the result, and optionally parts of the internal matrix computations.
- `addNA()` is faster now, e.g. when applied twice. (Part of PR#16895.)
- New option `rstandard(<lm>,type = "predicted")` provides the "PRESS"–related leave-one-out cross-validation errors for linear models.
- After seven years of deprecation, duplicated factor levels now produce a warning when printed and an error in `levels<-` instead of a warning.
- Invalid factors, e.g., with duplicated levels (invalid but constructable) now give a warning when printed, *via* new function `.valid.factor()`.
- `sessionInfo()` has been updated for Apple's change in OS naming as from '10.12' ('macOS Sierra' *vs* 'OS X El Capitan').

  Its `toLatex()` method now includes the `running` component.
- `options(interrupt=)` can be used to specify a default action for user interrupts. For now, if this option is not set and the `error` option is set, then an unhandled user interrupt invokes the `error` option. (This may be dropped in the future as `interrupt` conditions are not `error` conditions.)

- In most cases user interrupt handlers will be called with a `"resume"` restart available. Handlers can invoke this restart to resume computation. At the browser prompt the `r` command will invoke a `"resume"` restart if one is available. Some read operations cannot be resumed properly when interrupted and do not provide a `"resume"` restart.
- Radix sort is now chosen by `method = "auto"` for `sort.int()` for double vectors (and hence used for `sort()` for unclassed double vectors), excluding 'long' vectors.

  `sort.int(method = "radix")` no longer rounds double vectors.
- The `default` and `data.frame` methods for `stack()` preserve the names of empty elements in the levels of the `ind` column of the return value. Set the new `drop` argument to `TRUE` for the previous behavior.
- Speedup in `simplify2array()` and hence `sapply()` and `mapply()` (for the case of names and common length > 1), thanks to Suharto Anggono's PR#17118.
- `table(x,exclude = NULL)` now sets `useNA = "ifany"` (instead of `"always"`). Together with the bug fixes for this case, this recovers more consistent behaviour compatible to older versions of R. As a consequence, `summary()` for a logical vector no longer reports (zero) counts for `NA` when there are no `NA`s.
- `dump.frames()` gets a new option `include.GlobalEnv` which allows to also dump the global environment, thanks to Andreas Kersting's proposal in PR#17116.
- `system.time()` now uses `message()` instead of `cat()` when terminated early, such that `suppressMessages()` has an effect; suggested by Ben Bolker.
- `citation()` supports 'inst/CITATION' files from package source trees, with `lib.loc` pointing to the directory containing the package.
- `try()` gains a new argument `outFile` with a default that can be modified *via* `options(try.outFile = .)`, useful notably for `Sweave`.
- The unexported low-level functions in package **parallel** for passing serialized R objects to and from forked children now support long vectors on 64-bit platforms. This removes some limits on higher-level functions such as `mclapply()` (but returning gigabyte results from forked processes *via* serialization should be avoided if at all possible).
- Connections now `print()` without error even if invalid, e.g. after having been destroyed.
- `apropos()` and `find(simple.words = FALSE)` no longer match object names starting with '`.`' which are known to be internal objects (such as `.__S3MethodsTable__.`).
- Convenience function `hasName()` has been added; it is intended to replace the common idiom `!is.null(x$name)` without the usually unintended partial name matching.
- `strcapture()` no longer fixes column names nor coerces strings to factors (suggested by Bill Dunlap).
- `strcapture()` returns `NA` for non-matching values in `x` (suggested by Bill Dunlap).
- `source()` gets new optional arguments, notably `exprs`; this is made use of in the new utility function `withAutoprint()`.
- `sys.source()` gets a new `toplevel.env` argument. This argument is useful for frameworks running package tests; contributed by Tomas Kalibera.
- `Sys.setFileTime()` and `file.copy(copy.date = TRUE)` will set timestamps with fractions of seconds on platforms/filesystems which support this.

- (Windows only.) `file.info()` now returns file timestamps including fractions of seconds; it has done so on other platforms since R 2.14.0. (NB: some filesystems do not record modification and access timestamps to sub-second resolution.)
- The license check enabled by `options(checkPackageLicense = TRUE)` is now done when the package's namespace is first loaded.
- `ppr()` and `supsmu()` get an optional `trace` argument, and `ppr(..,sm.method = ..spline)` is no longer limited to sample size $n \leq 2500$.
- The `POSIXct` method for `print()` gets optional `tz` and `usetz` arguments, thanks to a report from Jennifer S. Lyon.
- New function `check_packages_in_dir_details()` in package **tools** for analyzing package-check log files to obtain check details.
- Package **tools** now exports function `CRAN_package_db()` for obtaining information about current packages in the CRAN package repository, and several functions for obtaining the check status of these packages.
- The (default) Stangle driver `Rtangle` allows `annotate` to be a function and gets a new `drop.evalFALSE` option.
- The default method for `quantile(x,prob)` should now be monotone in `prob`, even in border cases, see PR#16672.
- `bug.report()` now tries to extract an email address from a '`BugReports`' field, and if there is none, from a '`Contacts`' field.
- The `format()` and `print()` methods for `object.size()` results get new options `standard` and `digits`; notably, `standard = "IEC"` and `standard = "SI"` allow more standard (but less common) abbreviations than the default ones, e.g. for kilobytes. (From contributions by Henrik Bengtsson.)
- If a reference class has a validity method, `validObject` will be called automatically from the default initialization method for reference classes.
- `tapply()` gets new option `default = NA` allowing to change the previously hardcoded value.
- `read.dcf()` now consistently interprets any 'whitespace' to be stripped to include newlines.
- The maximum number of DLLs that can be loaded into R e.g. *via* `dyn.load()` can now be increased by setting the environment variable `R_MAX_NUM_DLLS` before starting R.
- Assigning to an element of a vector beyond the current length now over-allocates by a small fraction. The new vector is marked internally as growable, and the true length of the new vector is stored in the `truelength` field. This makes building up a vector result by assigning to the next element beyond the current length more efficient, though pre-allocating is still preferred. The implementation is subject to change and not intended to be used in packages at this time.
- Loading the **parallel** package namespace no longer sets or changes the `.Random.seed`, even if `R_PARALLEL_PORT` is unset.

  NB: This can break reproducibility of output, and did for a CRAN package.
- Methods `"wget"` and `"curl"` for `download.file()` now give an R error rather than a non-zero return value when the external command has a non-zero status.
- Encoding name `"utf8"` is mapped to `"UTF-8"`. Many implementations of `iconv` accept `"utf8"`, but not GNU **libiconv** (including the late 2016 version 1.15).

- `sessionInfo()` shows the full paths to the library or executable files providing the BLAS/LAPACK implementations currently in use (not available on Windows).
- The binning algorithm used by bandwidth selectors `bw.ucv()`, `bw.bcv()` and `bw.SJ()` switches to a version linear in the input size `n` for `n >nb/2`. (The calculations are the same, but for larger `n/nb` it is worth doing the binning in advance.)
- There is a new option `PCRE_study` which controls when `grep(perl = TRUE)` and friends 'study' the compiled pattern. Previously this was done for 11 or more input strings: it now defaults to 10 or more (but most examples need many more for the difference from studying to be noticeable).
- `grep(perl = TRUE)` and friends can now make use of PCRE's Just-In-Time mechanism, for PCRE $\geq$ 8.20 on platforms where JIT is supported. It is used by default whenever the `pattern` is studied (see the previous item). (Based on a patch from Mikko Korpela.)

  This is controlled by a new option `PCRE_use_JIT`.

  Note that in general this makes little difference to the speed, and may take a little longer: its benefits are most evident on strings of thousands of characters. As a side effect it reduces the chances of C stack overflow in the PCRE library on very long strings (millions of characters, but see next item).

  Warning: segfaults were seen using PCRE with JIT enabled on 64-bit Sparc builds.
- There is a new option `PCRE_limit_recursion` for `grep(perl = TRUE)` and friends to set a recursion limit taking into account R's estimate of the remaining C stack space (or 10000 if that is not available). This reduces the chance of C stack overflow, but because it is conservative may report a non-match (with a warning) in examples that matched before. By default it is enabled if any input string has 1000 or more bytes. (PR#16757)
- `getGraphicsEvent()` now works on `X11(type = "cairo")` devices. Thanks to Frederick Eaton (for reviving an earlier patch).
- There is a new argument `onIdle` for `getGraphicsEvent()`, which allows an R function to be run whenever there are no pending graphics events. This is currently only supported on X11 devices. Thanks to Frederick Eaton.
- The `deriv()` and similar functions now can compute derivatives of `log1p()`, `sinpi()` and similar one-argument functions, thanks to a contribution by Jerry Lewis.
- `median()` gains a formal `...` argument, so methods with extra arguments can be provided.
- `strwrap()` reduces `indent` if it is more than half `width` rather than giving an error. (Suggested by Bill Dunlap.)
- When the condition `code` in `if(.)` or `while(.)` is not of length one, an error instead of a warning may be triggered by setting an environment variable, see the help page.
- Formatting and printing of bibliography entries (`bibentry`) is more flexible and better documented. Apart from setting `options(citation.bibtex.max = 99)` you can also use `print(<citation>,bibtex=TRUE)` (or `format(..)`) to get the BibTeX entries in the case of more than one entry. This also affects `citation()`. Contributions to enable `style = "html+bibtex"` are welcome.

C-LEVEL FACILITIES

- Entry points `R_MakeExternalPtrFn` and `R_ExternalPtrFn` are now declared in header 'Rinternals.h' to facilitate creating and retrieving an R external pointer from a C function pointer without ISO C warnings about the conversion of function pointers.

- There was an exception for the native Solaris C++ compiler to the dropping (in R 3.3.0) of legacy C++ headers from headers such as 'R.h' and 'Rmath.h' — this has now been removed. That compiler has strict C++98 compliance hence does not include extensions in its (non-legacy) C++ headers: some packages will need to request C++11 or replace non-C++98 calls such as `lgamma`: see §1.6.4 of 'Writing R Extensions'.

  Because it is needed by about 70 CRAN packages, headers 'R.h' and 'Rmath.h' still declare

  ```
  use namespace std;
  ```

  when included on Solaris.

- When included from C++, the R headers now use forms such as `std::FILE` directly rather than including the line

  ```
  using std::FILE;
  ```

  C++ code including these headers might be relying on the latter.

- Headers 'R_ext/BLAS.h' and 'R_ext/Lapack.h' have many improved declarations including `const` for double-precision complex routines. *Inter alia* this avoids warnings when passing 'string literal' arguments from C++11 code.

- Headers for Unix-only facilities 'R_ext/GetX11Image.h', 'R_ext/QuartzDevice.h' and 'R_ext/eventloop.h' are no longer installed on Windows.

- No-longer-installed headers 'GraphicsBase.h', 'RGraphics.h', 'Rmodules/RX11.h' and 'Rmodules/Rlapack.h' which had a LGPL license no longer do so.

- `HAVE_UINTPTR_T` is now defined where appropriate by `Rconfig.h` so that it can be included before `Rinterface.h` when `CSTACK_DEFNS` is defined and a C compiler (not C++) is in use. `Rinterface.h` now includes C header 'stdint.h' or C++11 header 'cstdint' where needed.

- Package **tools** has a new function `package_native_routine_registration_skeleton()` to assist adding native-symbol registration to a package. See its help and §5.4.1 of 'Writing R Extensions' for how to use it. (At the time it was added it successfully automated adding registration to over 90% of CRAN packages which lacked it. Many of the failures were newly-detected bugs in the packages, e.g. 50 packages called entry points with varying numbers of arguments and 65 packages called entry points not in the package.)

INSTALLATION on a UNIX-ALIKE

- `readline` headers (and not just the library) are required unless configuring with '`--with-readline=no`'.

- `configure` now adds a compiler switch for C++11 code, even if the compiler supports C++11 by default. (This ensures that `g++` 6.x uses C++11 mode and not its default mode of C++14 with 'GNU extensions'.)

  The tests for C++11 compliance are now much more comprehensive. For gcc < 4.8, the tests from R 3.3.0 are used in order to maintain the same behaviour on Linux distributions with long-term support.

- An alternative compiler for C++11 is now specified with '`CXX11`', not '`CXX1X`'. Likewise C++11 flags are specified with '`CXX11FLAGS`' and the standard (e.g., '`-std=gnu++11`' is specified with '`CXX11STD`'.

- `configure` now tests for a C++14-compliant compiler by testing some basic features. This by default tries flags for the compiler specified by '`CXX11`', but an alternative compiler, options and standard can be specified by variables '`CXX14`', '`CXX14FLAGS`' and '`CXX14STD`' (e.g., '`-std=gnu++14`').

- There is a new macro `CXXSTD` to help specify the standard for C++ code, e.g. '`-std=c++98`'. This makes it easier to work with compilers which default to a later standard: for example, with `CXX=g++6 CXXSTD=-std=c++98 configure` will select commands for g++ 6.x which conform to C++11 and C++14 where specified but otherwise use C++98.

- Support for the defunct IRIX and OSF/1 OSes and Alpha CPU has been removed.

- `configure` checks that the compiler specified by '`$CXX $CXXFLAGS`' is able to compile C++ code.

- `configure` checks for the required header 'sys/select.h' (or 'sys/time.h' on legacy systems) and system call `select` and aborts if they are not found.

- If available, the POSIX 2008 system call `utimensat` will be used by `Sys.setFileTime()` and `file.copy(copy.date = TRUE)`. This may result in slightly more accurate file times. (It is available on Linux and FreeBSD but not macOS.)

- The minimum version requirement for `libcurl` has been reduced to 7.22.0, although at least 7.28.0 is preferred and earlier versions are little tested. (This is to support Debian 7 'Wheezy' LTS and Ubuntu 'Precise' 12.04 LTS, although the latter is close to end-of-life.)

- `configure` tests for a C++17-compliant compiler. The tests are experimental and subject to change in the future.

INCLUDED SOFTWARE

- (Windows only) Tcl/Tk version 8.6.4 is now included in the binary builds. The 'tcltk*.chm' help file is no longer included; please consult the online help at http://www.tcl.tk/man/ instead.

- The version of LAPACK included in the sources has been updated to 3.7.0: no new routines have been added to R.

PACKAGE INSTALLATION

- There is support for compiling C++14 or C++17 code in packages on suitable platforms: see 'Writing R Extensions' for how to request this.

- The order of flags when '`LinkingTo`' other packages has been changed so their include directories come earlier, before those specified in `CPPFLAGS`. This will only have an effect if non-system include directories are included with '`-I`' flags in `CPPFLAGS` (and so not the default `-I/usr/local/include` which is treated as a system include directory on most platforms).

- Packages which register native routines for `.C` or `.Fortran` need to be re-installed for this version (unless installed with R-devel SVN revision r72375 or later).

- Make variables with names containing `CXX1X` are deprecated in favour of those using `CXX11`, but for the time being are still made available *via* file 'etc/Makeconf'. Packages using them should be converted to the new forms and made dependent on '`R (>= 3.4.0)`'.

UTILITIES

- Running `R CMD check --as-cran` with `_R_CHECK_CRAN_INCOMING_REMOTE_` false now skips tests that require remote access. The remaining (local) tests typically run quickly compared to the remote tests.
- `R CMD build` will now give priority to vignettes produced from files in the 'vignettes' directory over those in the 'inst/doc' directory, with a warning that the latter are being ignored.
- `R CMD config` gains a '`--all`' option for printing names and values of all basic configure variables.

  It now knows about all the variables used for the C++98, C++11 and C++14 standards.
- `R CMD check` now checks that output files in 'inst/doc' are newer than the source files in 'vignettes'.
- For consistency with other package subdirectories, files named '*.r' in the 'tests' directory are now recognized as tests by `R CMD check`. (Wish of PR#17143.)
- `R CMD build` and `R CMD check` now use the *union* of `R_LIBS` and `.libPaths()`. They may not be equivalent, e.g., when the latter is determined by `R_PROFILE`.
- `R CMD build` now preserves dates when it copies files in preparing the tarball. (Previously on Windows it changed the dates on all files; on Unix, it changed some dates when installing vignettes.)
- The new option `R CMD check --no-stop-on-test-error` allows running the remaining tests (under 'tests/') even if one gave an error.
- Check customization *via* environment variables to detect side effects of `.Call()` and `.External()` calls which alter their arguments is described in §8 of the 'R Internals' manual.
- `R CMD check` now checks any '`BugReports`' field to be non-empty and a suitable single URL.
- `R CMD check --as-cran` now NOTEs if the package does not register its native routines or does not declare its intentions on (native) symbol search. (This will become a WARNING in due course.)

DEPRECATED AND DEFUNCT

- (Windows only) Function `setInternet2()` is defunct.
- Installation support for `readline` emulations based on `editline` (aka `libedit`) is deprecated.
- Use of the C/C++ macro '`NO_C_HEADERS`' is defunct and silently ignored.
- `unix.time()`, a traditional synonym for `system.time()`, has been deprecated.
- `structure(NULL,..)` is now deprecated as you cannot set attributes on `NULL`.
- Header 'Rconfig.h' no longer defines '`SUPPORT_OPENMP`'; instead use '`_OPENMP`' (as documented for a long time).
- (C-level Native routine registration.) The deprecated `styles` member of the `R_CMethodDef` and `R_FortranMethodDef` structures has been removed. Packages using these will need to be re-installed for R 3.4.0.
- The deprecated support for PCRE versions older than 8.20 will be removed in R 3.4.1. (Versions 8.20–8.31 will still be accepted but remain deprecated.)

BUG FIXES

- Getting or setting `body()` or `formals()` on non-functions for now signals a warning and may become an error for setting.
- `match(x,t)`, `duplicated(x)` and `unique(x)` work as documented for complex numbers with `NA`s or `NaN`s, where all those containing `NA` do match, whereas in the case of `NaN`'s both real and imaginary parts must match, compatibly with how `print()` and `format()` work for complex numbers.
- `deparse(<complex>,options = "digits17")` prints more nicely now, mostly thanks to a suggestion by Richie Cotton.
- Rotated symbols in plotmath expressions are now positioned correctly on `x11(type = "Xlib")`. (PR#16948)
- `as<-()` avoids an infinite loop when a virtual class is interposed between a subclass and an actual superclass.
- Fix level propagation in `unlist()` when the list contains zero-length lists or factors.
- Fix S3 dispatch on S4 objects when the **methods** package is not attached.
- Internal S4 dispatch sets `.Generic` in the method frame for consistency with `standardGeneric()`. (PR#16929)
- Fix `order(x,decreasing = TRUE)` when `x` is an integer vector containing `MAX_INT`. Ported from a fix Matt Dowle made to **data.table**.
- Fix caching by `callNextMethod()`, resolves PR#16973 and PR#16974.
- `grouping()` puts NAs last, to be consistent with the default behavior of `order()`.
- Point mass limit cases: `qpois(-2,0)` now gives `NaN` with a warning and `qgeom(1,1)` is `0`. (PR#16972)
- `table()` no longer drops an `"NaN"` factor level, and better obeys `exclude = <chr>`, thanks to Suharto Anggono's patch for PR#16936. Also, in the case of `exclude = NULL` and `NA`s, these are tabulated correctly (again).

  Further, `table(1:2,exclude = 1,useNA = "ifany")` no longer erroneously reports `<NA>` counts.

  Additionally, all cases of empty `exclude` are equivalent, and `useNA` is not overwritten when specified (as it was by `exclude = NULL`).
- `wilcox.test(x,conf.int=TRUE)` no longer errors out in cases where the confidence interval is not available, such as for `x = 0:2`.
- `droplevels(f)` now keeps <NA> levels when present.
- In integer arithmetic, `NULL` is now treated as `integer(0)` whereas it was previously treated as `double(0)`.
- The radix sort considers `NA_real_` and `NaN` to be equivalent in rank (like the other sort algorithms).
- When `index.return=TRUE` is passed to `sort.int()`, the radix sort treats `NA`s like `sort.list()` does (like the other sort algorithms).
- When in `tabulate(bin,nbin)` `length(bin)` is larger than the maximal integer, the result is now of type `double` and hence no longer silently overflows to wrong values. (PR#17140)

- `as.character.factor()` respects S4 inheritance when checking the type of its argument. (PR#17141)
- The `factor` method for `print()` no longer sets the class of the factor to `NULL`, which would violate a basic constraint of an S4 object.
- `formatC(x,flag = f)` allows two new flags, and signals an error for invalid flags also in the case of character formatting.
- Reading from `file("stdin")` now also closes the connection and hence no longer leaks memory when reading from a full pipe, thanks to Gábor Csárdi, see thread starting at https://stat.ethz.ch/pipermail/r-devel/2016-November/073360.html.
- Failure to create file in `tempdir()` for compressed `pdf()` graphics device no longer errors (then later segfaults). There is now a warning instead of error and compression is turned off for the device. Thanks to Alec Wysoker (PR#17191).
- Asking for `methods()` on `"|"` returns only S3 methods. See https://stat.ethz.ch/pipermail/r-devel/2016-December/073476.html.
- `dev.capture()` using Quartz Cocoa device (macOS) returned invalid components if the back-end chose to use ARGB instead of RGBA image format. (Reported by Noam Ross.)
- `seq("2","5")` now works too, equivalently to `"2":"5"` and `seq.int()`.
- `seq.int(to = 1,by = 1)` is now correct, other cases are integer (instead of `double`) when `seq()` is integer too, and the "non-finite" error messages are consistent between `seq.default()` and `seq.int()`, no longer mentioning `NaN` etc.
- `rep(x,times)` and `rep.int(x,times)` now work when `times` is larger than the largest value representable in an integer vector. (PR#16932)
- `download.file(method = "libcurl")` does not check for URL existence before attempting downloads; this is more robust to servers that do not support HEAD or range-based retrieval, but may create empty or incomplete files for aborted download requests.
- Bandwidth selectors `bw.ucv()`, `bw.bcv()` and `bw.SJ()` now avoid integer overflow for large sample sizes.
- `str()` no longer shows `"list output truncated"`, in cases that list was not shown at all. Thanks to Neal Fultz (PR#17219)
- Fix for `cairo_pdf()` (and `svg()` and `cairo_ps()`) when replaying a saved display list that contains a mix of **grid** and **graphics** output. (Report by Yihui Xie.)
- The `str()` and `as.hclust()` methods for `"dendrogram"` now also work for deeply nested dendrograms thanks to non-recursive implementations by Bradley Broom.
- `sample()` now uses two uniforms for added precision when the uniform generator is `Knuth-TAOCP`, `Knuth-TAOCP-2002`, or a user-defined generator and the population size is 2^{25} or greater.
- If a vignette in the 'vignettes' directory is listed in '.Rbuildignore', `R CMD build` would not include it in the tarball, but would include it in the vignette database, leading to a check warning. (PR#17246)
- `tools::latexToUtf8()` infinite looped on certain inputs. (PR#17138)
- `terms.formula()` ignored argument names when determining whether two terms were identical. (PR#17235)

- `callNextMethod()` was broken when called from a method that augments the formal arguments of a primitive generic.
- Coercion of an S4 object to a vector during sub-assignment into a vector failed to dispatch through the `as.vector()` generic (often leading to a segfault).
- Fix problems in command completion: Crash (PR#17222) and junk display in Windows, handling special characters in filenames on all systems.

CHANGES IN R 3.3.3

NEW FEATURES

- Changes when redirection of a 'http://' URL to a 'https://' URL is encountered:
  - The internal methods of `download.file()` and `url()` now report that they cannot follow this (rather than failing silently).
  - (Unix-alike) `download.file(method = "auto")` (the default) re-tries with `method = "libcurl"`.
  - (Unix-alike) `url(method = "default")` with an explicit `open` argument re-tries with `method = "libcurl"`. This covers many of the usages, e.g. `readLines()` with a URL argument.

INSTALLATION on a UNIX-ALIKE

- The `configure` check for the `zlib` version is now robust to versions longer than 5 characters, including `1.2.11`.

UTILITIES

- Environmental variable `_R_CHECK_TESTS_NLINES_` controls how `R CMD check` reports failing tests (see §8 of the 'R Internals' manual).

DEPRECATED AND DEFUNCT

- (C-level Native routine registration.) The undocumented `styles` field of the components of `R_CMethodDef` and `R_FortranMethodDef` is deprecated.

BUG FIXES

- `vapply(x,*)` now works with long vectors `x`. (PR#17174)
- `isS3method("is.na.data.frame")` and similar are correct now. (PR#17171)
- `grepRaw(<long>,<short>,fixed = TRUE)` now works, thanks to a patch by Mikko Korpela. (PR#17132)
- Package installation into a library where the package exists *via* symbolic link now should work wherever `Sys.readlink()` works, resolving PR#16725.
- `"Cincinnati"` was missing an `"n"` in the `precip` dataset.
- Fix buffer overflow vulnerability in `pdf()` when loading an encoding file. Reported by Talos (TALOS-2016-0227).
- `getDLLRegisteredRoutines()` now produces its warning correctly when multiple DLLs match, thanks to Matt Dowle's PR#17184.

- `Sys.timezone()` now returns non-NA also on platforms such as '`Ubuntu 14.04.5 LTS`', thanks to Mikko Korpela's PR#17186.
- `format(x)` for an illegal `"POSIXlt"` object `x` no longer segfaults.
- `methods(f)` now also works for `f "("` or `"{"`.
- (Windows only) `dir.create()` did not check the length of the path to create, and so could overflow a buffer and crash R. (PR#17206)
- On some systems, very small hexadecimal numbers in hex notation would underflow to zero. (PR#17199)
- `pmin()` and `pmax()` now work again for `ordered` factors and 0-length S3 classed objects, thanks to Suharto Anggono's PR#17195 and PR#17200.
- `bug.report()` did not do any validity checking on a package's '`BugReports`' field. It now ignores an empty field, removes leading whitespace and only attempts to open '`http://`' and '`https://`' URLs, falling back to emailing the maintainer.
- Bandwidth selectors `bw.ucv()` and `bw.SJ()` gave incorrect answers or incorrectly reported an error (because of integer overflow) for inputs longer than 46341. Similarly for `bw.bcv()` at length 5793.

  Another possible integer overflow is checked and may result in an error report (rather than an incorrect result) for much longer inputs (millions for a smooth distribution).
- `findMethod()` failed if the active signature had expanded beyond what a particular package used. (Example with packages **XR** and **XRJulia** on CRAN.)
- `qbeta()` underflowed too early in some very asymmetric cases. (PR#17178)
- `R CMD Rd2pdf` had problems with packages with non-ASCII titles in '`.Rd`' files (usually the titles were omitted).