locatr, a method finder for R

Amelia McNamara,Ted Kaehler

VPRI Research Note RN-2013-002

Viewpoints Research Institute, 1209 Grand Central Avenue, Glendale, CA 91201   t: (818) 332-3001 f: (818) 244-9761

locatr, a method finder for R

Amelia McNamara Ted Kaehler

November 25, 2013

Inspiration

The idea of `locatr` was to create a method finder in the style of Ted Kaehler's method finder for Squeak. Ideally, it will be useful to both novices and experts for the discovery of functions (methods) that are hard to remember.

For the novice, it can help remind them about when to use the `$` operator, when square braces are appropriate, whether rows or columns come first in indexing, etc. For the more experienced user, it can be used to help craft syntax for more complicated functions, such as `apply` or `ddply`. A user needs only to provide a set of data and a desired answer, and (hopefully) the method finder will discover the function they're looking for. In many cases, it finds the "expected" answer as well as a number of unexpected ones.

The package

The package,

```
library(locatr)
```

consists of three main functions:

```
isit
whatgives
locatr
```

Generally, users will interact with the titular `locatr` function. The other two functions are helpers for `locatr` or useful in specific use cases.

The function `isit` does a simple test to see if a given function with a particular set of arguments produces the desired result. It returns TRUE if there is a match (with some thresholding to prevent rounding error false negatives) and FALSE if there is not. For example,

```
isit("+", list(3, 4), 7)


## [1] TRUE


isit("+", list(c(3, 4, 5)), 3)


## [1] FALSE
```

Most of the time, users will not interact with `isit` directly. Instead, they will use `whatgives` or, most commonly, `locatr`.

The function `whatgives` will test the set of arguments and the desired result against all the R functions that have been declared safe to test, using `isit`. However, it tests the arguments in the exact order they were provided.

```
whatgives(list(3, 4), -1)


## [1] "3-4"           "regexpr(3, 4)"


whatgives(list(4, 3), -1)


## [1] "regexpr(4, 3)"
```

In the case where you don't know the particular order of your arguments (for example, if you're using apply and forget if the row/column flag comes before the function to be applied) `locatr` is the function you want. `locatr` tests all permutations of the arguments against all the safe functions to find the functions that provide the desired result.

```
locatr(list(3, 4), -1)


## Loading required package:  gtools


##           output
## 1            3-4
## 2 regexpr(3, 4)
## 3 regexpr(4, 3)


x <- cbind(x1 = 3, x2 = c(4:1, 2:5))
locatr(list(x, 2, mean), c(3, 3))
```

```
##              output
## 1 apply(x, 2, mean)
## 2   rep(x, mean, 2)
```

```
locatr(list(mean, x, 2), c(3, 3))
```

```
##              output
## 1   rep(x, mean, 2)
## 2 apply(x, 2, mean)
```

If you want to try the package yourself, follow the installation instructions in Appendix A. Again, the function to focus on when using the package is the `locatr` function.

Finding "safe" functions

The process to find the set of "safe" functions from the base R package was somewhat tedious. It is simple to list all the functions in a given package,

```
ls("package:base")
```

but the list can be more exhaustive than a person might like. After trying several examples on the entire list, it became clear that not all the functions in the base package made sense to test. Then began a process of elimination, so that the list would be as side-effect-free as possible. The functions that have been deemed "bad" fall into a few broad categories.

Some functions (although they are exported by the base package) are not intended to be utilized by end users, so they had surprising side effects when tried. This category includes functions like `asNamespace`, and many others that can be found by checking `?asNamespace`.

Others do not check that the arguments passed to them are of the expected class, so produce unexpected results. For example, `attach` and `detach` behave quite strangely in this context. If you pass a set of numeric arguments (as a list, for example), search paths will be changed for your R session that result in the entire session needing termination.

Similarly, functions that open connections and write files, like `write` and `gzcon`, do not check that they are being passed filenames or connections. If they are given numeric arguments they open strange connections and/or write files to your system with numeric filenames. This is obviously undesirable behavior, as we'd like the method finder to have no side effects.

Then, R has a number of generic functions, like `mean`, which behave differently depending on the class of object they are called on. Using the full list of functions from the package means that each "subsidiary function" is tested separately, e.g. `mean.Date`, `mean.difftime`, etc. These functions are especially bad about checking that the given arguments are of the proper class, because they were designed only to be called by the generic function, e.g. `mean`. Again, they produce strange results. So, all these so-called “subsidiary functions” were removed as well.

Finally, there are many functions in base R that include the assignment operator, `<-`. For example, `length<-`. These seemed dangerous in many situations, so they have all been removed from the list that is being tested. However, in some cases it may make sense to reinstitute them. For example, one might want to test

```
locatr(list(factor(c(1, 2, 3)), c("A", "B", "C")), factor(c("A", "B", "C")))
```

in which case `levels<-` would be the expected result.

The full list of functions that were deemed unsafe is listed in Appendix B. Again, this list was made with broad strokes, so it may make sense to slowly remove functions from the blacklist. This is a task for the future.

Further work

There is a lot more that needs to be done before `locatr` will be ready for real users. Some of these tasks include:

1. Include additional functions from other popular R packages. Eventually, I would like to support all of the top 100 R packages (as defined somewhere like `http://www.r-statistics.com/2013/06/top-100-r-packages-for-2013-jan-may/`) and all the default R packages (like utils, stats, etc). Given the difficulty of getting the base R package functions categorized as safe or unsafe, I’d probably start tackling the other 100 first.

2. Add support for matching attributes as well as objects themselves.

3. Test certain simple modifications to the input data. For example, try a vector as-is, then try it as a factor (especially if there were no results to begin with). Think about other examples that might also be relevant.

4. Work on a way to build up the list of functions programmatically, perhaps as an additional function? The idea would be that you’d get the package with the functions I’d defined as safe (hopefully, thousands of them!) but if you didn’t like my list, you could change it. For example, if you were a very novice user you could choose to just search a few packages, or if you were very experienced you could add additional package functions. This seems to be asking for some sort of database. For example, I’ve already discovered that one of the top 100 packages, `plyr`, has a strange function that we probably do not want to test: `progress_tk`. It would be great to be able to say, we’re going to use all the functions from `plyr` *except* `progress_tk`.

5. Include some more protections.

(a) A timer, so if trying a function is taking more than a second, you can kill it.

(b) Something that preserves the state of the environment and puts it back after you try a function, so you can't do too much damage to your R.

(c) Something that preserves the state of your data and answer objects so that functions can't change them behind the scenes, causing a trickledown effect with all the other functions tried after it.

A How to get the current (development) version of *locatr*

The instructions assume you have R running on your computer and understand how to execute R commands, but otherwise have minimal requirements.

To get the most up-to-date development version of `locatr`, you need the package `devtools`, which allows package installation straight from github.

```r
install.packages("devtools")
library(devtools)
install_github("locatr", "AmeliaMN")
```

B List of base R functions deemed unsafe or unnecessary for the package to test

```r
load("locatr/R/sysdata.rda")
badfunctions


##  [1] "cat"
##  [2] "kronecker"
##  [3] "l10n_info"
##  [4] "La.svd"
##  [5] "lockEnvironment"
##  [6] "lockBinding"
##  [7] "message"
##  [8] "packageStartupMessage"
##  [9] "print"
## [10] "quit"
## [11] "R.home"
## [12] "rawConnection"
## [13] "readRenviron"
## [14] "Recall"
## [15] "repeat"
## [16] "require"
## [17] "restartDescription"
## [18] "stop"
## [19] "stopifnot"
## [20] "substring<-"
## [21] "traceback"
## [22] "while"
##  [23] "asNamespace"
##  [24] "getNamespaceInfo"
##  [25] "importIntoEnv"
##  [26] "isBaseNamespace"
##  [27] "isNamespace"
##  [28] "namespaceExport"
##  [29] "namespaceImport"
##  [30] "namespaceImportFrom"
##  [31] "namespaceImportClasses"
##  [32] "namespaceImportMethods"
##  [33] "packageHasNamespace"
##  [34] "parseNamespaceFile"
##  [35] "registerS3method"
##  [36] "registerS3methods"
##  [37] "setNamespaceInfo"
##  [38] "loadingNamespaceInfo"
##  [39] "getNamespace"
##  [40] "file"
##  [41] "url"
##  [42] "gzfile"
##  [43] "bzfile"
##  [44] "xzfile"
##  [45] "unz"
##  [46] "pipe"
##  [47] "fifo"
##  [48] "socketConnection"
##  [49] "open"
##  [50] "close"
##  [51] "flush"
##  [52] "isOpen"
##  [53] "isIncomplete"
##  [54] "browser"
##  [55] "gc"
##  [56] "RNGkind"
##  [57] "trace"
##  [58] "tracemem"
##  [59] "tracingState"
##  [60] "warning"
##  [61] "lazyLoad"
##  [62] "library"
##  [63] "library.dynam"
##  [64] "library.dynam.unload"
##  [65] "license"
##  [66] "load"
##  [67] "attachNamespace"
##  [68] "loadNamespace"
##  [69] "requireNamespace"
##  [70] "loadedNamespaces"
##  [71] "unloadNamespace"
##  [72] "options"
##  [73] "remove"
##  [74] "rm"
##  [75] "save"
##  [76] "save.image"
##  [77] "setwd"
##  [78] "srcfile"
##  [79] "srcfilealias"
##  [80] "srcfilecopy"
##  [81] "srcref"
##  [82] "system"
##  [83] "system2"
##  [84] "tempdir"
##  [85] "tempfile"
##  [86] "socketSelect"
##  [87] "gc.time"
##  [88] "gcinfo"
##  [89] "gctorture"
##  [90] "gctorture2"
##  [91] "readline"
##  [92] "find.package"
##  [93] "path.package"
##  [94] "testPlatformEquivalence"
##  [95] "findPackageEnv"
##  [96] "lazyLoadDBfetch"
##  [97] "gzcon"
##  [98] "memCompress"
##  [99] "readLines"
## [100] "writeLines"
## [101] "sink"
## [102] "scan"
## [103] "read.dcf"
## [104] "dput"
## [105] "dump"
## [106] "readBin"
## [107] "readChar"
## [108] "writeBin"
## [109] "writeChar"
## [110] "textConnection"
## [111] "seek"
## [112] "pushBack"
## [113] "getConnection"
## [114] "stdin"
## [115] "stdout"
## [116] "stderr"
## [117] "isatty"
## [118] "setTimeLimit"
## [119] "setSessionTimeLimit"
## [120] "autoload"
## [121] "autoloader"
## [122] "readRDS"
## [123] "saveRDS"
## [124] "environment"
## [125] "environment<-"
## [126] "emptyenv"
## [127] "globalenv"
## [128] "baseenv"
## [129] "parent.env"
## [130] "parent.env<-"
## [131] "environmentName"
## [132] "env.profile"
## [133] "getwd"
## [134] "list.files"
## [135] "normalizePath"
## [136] "dir.create"
## [137] "basename"
## [138] "dirname"
## [139] "browserText"
## [140] "browserCondition"
## [141] "browserSetDebug"
## [142] "pos.to.env"
## [143] "gettext"
## [144] "ngettext"
## [145] "bindtextdomain"
## [146] "as.environment"
## [147] "environmentIsLocked"
## [148] "unlockBinding"
## [149] "bindingIsLocked"
## [150] "makeActiveBinding"
## [151] "bindingIsActive"
## [152] "dget"
## [153] "list2env"
## [154] "rawConnectionValue"
## [155] "search"
## [156] "searchpaths"
## [157] "seek.connection"
## [158] "standardGeneric"
## [159] "taskCallbackManager"
## [160] "attach"
## [161] "detach"
## [162] "serialize"
## [163] "unserialize"
## [164] "lazyLoadDBexec"
## [165] "prmatrix"
## [166] "getCallingDLL"
## [167] "getCallingDLLe"
## [168] "getDLLRegisteredRoutines"
## [169] "write"
## [170] "write.dcf"
## [171] "class<-"
## [172] "break"
## [173] "return"
## [174] "parse"
## [175] "debug"
## [176] "debugonce"
## [177] "undebug"
## [178] "isdebugged"
## [179] "-.Date"
## [180] "-.POSIXt"
## [181] "!.hexmode"
## [182] "!.octmode"
## [183] "[.AsIs"
## [184] "[.data.frame"
## [185] "[.Date"
## [186] "[.difftime"
## [187] "[.factor"
## [188] "[.hexmode"
## [189] "[.listof"
## [190] "[.noquote"
## [191] "[.numeric_version"
## [192] "[.octmode"
## [193] "[.POSIXct"
## [194] "[.POSIXlt"
## [195] "[.simple.list"
## [196] "[.warnings"
## [197] "[[.data.frame"
## [198] "[[.Date"
## [199] "[[.factor"
## [200] "[[.numeric_version"
## [201] "[[.POSIXct"
## [202] "[[<-.data.frame"
## [203] "[[<-.factor"
## [204] "[[<-.numeric_version"
## [205] "[<-.data.frame"
## [206] "[<-.Date"
## [207] "[<-.factor"
## [208] "[<-.POSIXct"
## [209] "[<-.POSIXlt"
## [210] "*.difftime"
## [211] "/.difftime"
## [212] "&.hexmode"
## [213] "&.octmode"
## [214] "+.Date"
## [215] "+.POSIXt"
## [216] "|.hexmode"
## [217] "|.octmode"
## [218] "$.DLLInfo"
## [219] "$.package_version"
## [220] "$<-.data.frame"
## [221] "all.equal.character"
## [222] "all.equal.default"
## [223] "all.equal.factor"
## [224] "all.equal.formula"
## [225] "all.equal.language"
## [226] "all.equal.list"
## [227] "all.equal.numeric"
## [228] "all.equal.POSIXct"
## [229] "all.equal.raw"
## [230] "anyDuplicated.array"
## [231] "anyDuplicated.data.frame"
## [232] "anyDuplicated.default"
## [233] "anyDuplicated.matrix"
## [234] "aperm.default"
## [235] "aperm.table"
## [236] "as.array.default"
## [237] "as.character.condition"
## [238] "as.character.Date"
## [239] "as.character.default"
## [240] "as.character.error"
## [241] "as.character.factor"
## [242] "as.character.hexmode"
## [243] "as.character.numeric_version"
## [244] "as.character.octmode"
## [245] "as.character.POSIXt"
## [246] "as.character.srcref"
## [247] "as.data.frame.array"
## [248] "as.data.frame.AsIs"
## [249] "as.data.frame.character"
## [250] "as.data.frame.complex"
## [251] "as.data.frame.data.frame"
## [252] "as.data.frame.Date"
## [253] "as.data.frame.default"
## [254] "as.data.frame.difftime"
## [255] "as.data.frame.factor"
## [256] "as.data.frame.integer"
## [257] "as.data.frame.list"
## [258] "as.data.frame.logical"
## [259] "as.data.frame.matrix"
## [260] "as.data.frame.model.matrix"
## [261] "as.data.frame.numeric"
## [262] "as.data.frame.numeric_version"
## [263] "as.data.frame.ordered"
## [264] "as.data.frame.POSIXct"
## [265] "as.data.frame.POSIXlt"
## [266] "as.data.frame.raw"
## [267] "as.data.frame.table"
## [268] "as.data.frame.ts"
## [269] "as.data.frame.vector"
## [270] "as.Date.character"
## [271] "as.Date.date"
## [272] "as.Date.dates"
## [273] "as.Date.default"
## [274] "as.Date.factor"
## [275] "as.Date.numeric"
## [276] "as.Date.POSIXct"
## [277] "as.Date.POSIXlt"
## [278] "as.double.difftime"
## [279] "as.double.POSIXlt"
## [280] "as.expression.default"
## [281] "as.list.data.frame"
## [282] "as.list.Date"
## [283] "as.list.default"
## [284] "as.list.environment"
## [285] "as.list.factor"
## [286] "as.list.function"
## [287] "as.list.numeric_version"
## [288] "as.list.POSIXct"
## [289] "as.logical.factor"
## [290] "as.matrix.data.frame"
## [291] "as.matrix.default"
## [292] "as.matrix.noquote"
## [293] "as.matrix.POSIXlt"
## [294] "as.null.default"
## [295] "as.POSIXct.date"
## [296] "as.POSIXct.Date"
## [297] "as.POSIXct.dates"
## [298] "as.POSIXct.default"
## [299] "as.POSIXct.numeric"
## [300] "as.POSIXct.POSIXlt"
## [301] "as.POSIXlt.character"
## [302] "as.POSIXlt.date"
## [303] "as.POSIXlt.Date"
## [304] "as.POSIXlt.dates"
## [305] "as.POSIXlt.default"
## [306] "as.POSIXlt.factor"
## [307] "as.POSIXlt.numeric"
## [308] "as.POSIXlt.POSIXct"
## [309] "as.single.default"
## [310] "as.table.default"
## [311] "as.vector.factor"
## [312] "by.data.frame"
## [313] "by.default"
## [314] "c.Date"
## [315] "c.noquote"
## [316] "c.numeric_version"
## [317] "c.POSIXct"
## [318] "c.POSIXlt"
## [319] "cbind.data.frame"
## [320] "chol.default"
## [321] "close.connection"
## [322] "close.srcfile"
## [323] "close.srcfilealias"
## [324] "conditionCall.condition"
## [325] "conditionMessage.condition"
## [326] "cut.Date"
## [327] "cut.default"
## [328] "cut.POSIXt"
## [329] "diff.Date"
## [330] "diff.default"
## [331] "diff.POSIXt"
## [332] "dim.data.frame"
## [333] "dimnames.data.frame"
## [334] "droplevels.data.frame"
## [335] "droplevels.factor"
## [336] "duplicated.array"
## [337] "duplicated.data.frame"
## [338] "duplicated.default"
## [339] "duplicated.matrix"
## [340] "duplicated.numeric_version"
## [341] "duplicated.POSIXlt"
## [342] "file.access"
## [343] "file.append"
## [344] "file.choose"
## [345] "file.copy"
## [346] "file.create"
## [347] "file.exists"
## [348] "file.info"
## [349] "file.link"
## [350] "file.path"
## [351] "file.remove"
## [352] "file.rename"
## [353] "file.show"
## [354] "file.symlink"
## [355] "format.AsIs"
## [356] "format.data.frame"
## [357] "format.Date"
## [358] "format.default"
## [359] "format.difftime"
## [360] "format.factor"
## [361] "format.hexmode"
## [362] "format.info"
## [363] "format.libraryIQR"
## [364] "format.numeric_version"
## [365] "format.octmode"
## [366] "format.packageInfo"
## [367] "format.POSIXct"
## [368] "format.POSIXlt"
## [369] "format.pval"
## [370] "format.summaryDefault"
## [371] "getDLLRegisteredRoutines.character"
## [372] "getDLLRegisteredRoutines.DLLInfo"
## [373] "is.na.data.frame"
## [374] "is.na.numeric_version"
## [375] "is.na.POSIXlt"
## [376] "is.na<-.default"
## [377] "is.na<-.factor"
## [378] "is.numeric.Date"
## [379] "is.numeric.difftime"
## [380] "is.numeric.POSIXt"
## [381] "julian.Date"
## [382] "julian.POSIXt"
## [383] "kappa.default"
## [384] "kappa.lm"
## [385] "kappa.qr"
## [386] "labels.default"
## [387] "length.POSIXlt"
## [388] "length<-.factor"
## [389] "levels.default"
## [390] "levels<-.factor"
## [391] "Math.data.frame"
## [392] "Math.Date"
## [393] "Math.difftime"
## [394] "Math.factor"
## [395] "Math.POSIXt"
## [396] "mean.Date"
## [397] "mean.default"
## [398] "mean.difftime"
## [399] "mean.POSIXct"
## [400] "mean.POSIXlt"
## [401] "merge.data.frame"
## [402] "merge.default"
## [403] "months.Date"
## [404] "months.POSIXt"
## [405] "names.POSIXlt"
## [406] "names<-.POSIXlt"
## [407] "open.connection"
## [408] "open.srcfile"
## [409] "open.srcfilealias"
## [410] "open.srcfilecopy"
## [411] "pmax.int"
## [412] "pmin.int"
## [413] "pretty.default"
## [414] "print.AsIs"
## [415] "print.by"
## [416] "print.condition"
## [417] "print.connection"
## [418] "print.data.frame"
## [419] "print.Date"
## [420] "print.default"
## [421] "print.difftime"
## [422] "print.DLLInfo"
## [423] "print.DLLInfoList"
## [424] "print.DLLRegisteredRoutines"
## [425] "print.factor"
## [426] "print.function"
## [427] "print.hexmode"
## [428] "print.libraryIQR"
## [429] "print.listof"
## [430] "print.NativeRoutineList"
## [431] "print.noquote"
## [432] "print.numeric_version"
## [433] "print.octmode"
## [434] "print.packageInfo"
## [435] "print.POSIXct"
## [436] "print.POSIXlt"
## [437] "print.proc_time"
## [438] "print.restart"
## [439] "print.rle"
## [440] "print.simple.list"
## [441] "print.srcfile"
## [442] "print.srcref"
## [443] "print.summary.table"
## [444] "print.summaryDefault"
## [445] "print.table"
## [446] "print.warnings"
## [447] "quarters.Date"
## [448] "quarters.POSIXt"
## [449] "range.default"
## [450] "rbind.data.frame"
## [451] "rep.Date"
## [452] "rep.factor"
## [453] "rep.int"
## [454] "rep.numeric_version"
## [455] "rep.POSIXct"
## [456] "rep.POSIXlt"
## [457] "rev.default"
## [458] "round.Date"
## [459] "round.POSIXt"
## [460] "row.names.data.frame"
## [461] "row.names.default"
## [462] "row.names<-.data.frame"
## [463] "row.names<-.default"
## [464] "rowsum.data.frame"
## [465] "rowsum.default"
## [466] "scale.default"
## [467] "seq.Date"
## [468] "seq.default"
## [469] "seq.int"
## [470] "seq.POSIXt"
## [471] "solve.default"
## [472] "solve.qr"
## [473] "sort.default"
## [474] "sort.int"
## [475] "sort.list"
## [476] "sort.POSIXlt"
## [477] "split.data.frame"
## [478] "split.Date"
## [479] "split.default"
## [480] "split.POSIXct"
## [481] "subset.data.frame"
## [482] "subset.default"
## [483] "subset.matrix"
## [484] "summary.connection"
## [485] "summary.data.frame"
## [486] "summary.Date"
## [487] "summary.default"
## [488] "summary.factor"
## [489] "summary.matrix"
## [490] "summary.POSIXct"
## [491] "summary.POSIXlt"
## [492] "summary.proc_time"
## [493] "summary.srcfile"
## [494] "summary.srcref"
## [495] "summary.table"
## [496] "Summary.data.frame"
## [497] "Summary.Date"
## [498] "Summary.difftime"
## [499] "Summary.factor"
## [500] "Summary.numeric_version"
## [501] "Summary.ordered"
## [502] "Summary.POSIXct"
## [503] "Summary.POSIXlt"
## [504] "sys.call"
## [505] "sys.calls"
## [506] "sys.frame"
## [507] "sys.frames"
## [508] "sys.function"
## [509] "sys.load.image"
## [510] "sys.nframe"
## [511] "sys.on.exit"
## [512] "sys.parent"
## [513] "sys.parents"
## [514] "sys.save.image"
## [515] "sys.source"
## [516] "sys.status"
## [517] "Sys.chmod"
## [518] "Sys.Date"
## [519] "Sys.getenv"
## [520] "Sys.getlocale"
## [521] "Sys.getpid"
## [522] "Sys.glob"
## [523] "Sys.info"
## [524] "Sys.localeconv"
## [525] "Sys.readlink"
## [526] "Sys.setenv"
## [527] "Sys.setFileTime"
## [528] "Sys.setlocale"
## [529] "Sys.sleep"
## [530] "Sys.time"
## [531] "Sys.timezone"
## [532] "Sys.umask"
## [533] "Sys.unsetenv"
## [534] "Sys.which"
## [535] "system.file"
## [536] "system.time"
## [537] "t.data.frame"
## [538] "t.default"
## [539] "transform.data.frame"
## [540] "transform.default"
## [541] "trunc.Date"
## [542] "trunc.POSIXt"
## [543] "unique.array"
## [544] "unique.data.frame"
## [545] "unique.default"
## [546] "unique.matrix"
## [547] "unique.numeric_version"
## [548] "unique.POSIXlt"
## [549] "units.difftime"
## [550] "units<-.difftime"
## [551] "weekdays.Date"
## [552] "weekdays.POSIXt"
## [553] "with.default"
## [554] "within.data.frame"
## [555] "within.list"
## [556] "xtfrm.AsIs"
## [557] "xtfrm.Date"
## [558] "xtfrm.default"
## [559] "xtfrm.difftime"
## [560] "xtfrm.factor"
## [561] "xtfrm.numeric_version"
## [562] "xtfrm.POSIXct"
## [563] "xtfrm.POSIXlt"
## [564] "xtfrm.Surv"
## [565] "[[<-"
## [566] "[<-"
## [567] "@<-"
## [568] "<-"
## [569] "<<-"
## [570] "$<-"
## [571] "attr<-"
## [572] "attributes<-"
## [573] "body<-"
## [574] "colnames<-"
## [575] "comment<-"
## [576] "diag<-"
## [577] "dim<-"
## [578] "dimnames<-"
## [579] "dimnames<-.data.frame"
## [580] "Encoding<-"
## [581] "formals<-"
## [582] "is.na<-"
## [583] "length<-"
## [584] "levels<-"
## [585] "mode<-"
## [586] "mostattributes<-"
## [587] "names<-"
## [588] "oldClass<-"
## [589] "regmatches<-"
## [590] "row.names<-"
## [591] "rownames<-"
## [592] "split<-"
## [593] "split<-.data.frame"
## [594] "split<-.default"
## [595] "storage.mode<-"
## [596] "substr<-"
## [597] "units<-"
```