Package ‘dang’

October 30, 2021

Type Package
Title 'Dang' Associated New Goodies
Version 0.0.15
Date 2021-10-26
Author Dirk Eddelbuettel with contributions by Brodie Gaslam, Kevin Denny, Kabira Namit, Colin Gillespie, R Core, Josh Ulrich, Jordan Mark Barbone, and others.
Maintainer Dirk Eddelbuettel <edd@debian.org>
Description A collection of utility functions.
URL https://github.com/eddelbuettel/dang,
    https://dirk.eddelbuettel.com/code/dang.html
BugReports https://github.com/eddelbuettel/dang/issues
Suggests anytime, data.table, zoo, xts, TTR, quantmod, rtweet
LinkingTo tidyCpp
License GPL (>= 2)
RoxygenNote 6.0.1
NeedsCompilation yes
Repository CRAN
Date/Publication 2021-10-29 23:10:02 UTC

R topics documented:

alphavantage ................................................. 2
as.data.table.xts ........................................... 3
assignFormals ................................................. 3
checkCRANStatus ............................................ 4
checkPackageAsciiCode ..................................... 5
demotivate .................................................... 5
getGitRoot .................................................... 6
googleFinanceData ......................................... 7
alphavantage

Retrieve real-time data from AlphaVantage

Description

Fetch a real-time market data series from AlphaVantage

Usage

alphavantage(sym)

Arguments

sym Character value for the ticker

Details

Several optional parameters could be set, but are not currently.

Value

A data.table object

Author(s)

Dirk Eddelbuettel
as.data.table.xts  Convert an xts object into a data.table

Description
Convert an xts object into a data.table

Usage

## S3 method for class 'xts'
as.data.table(x)

as.data.table(x)

## Default S3 method:
as.data.table(x, ...)

Arguments

x  An xts object
...
Catch-all arguments passed on to methods

Details
This is still experimental. Note that all four added columns are keys to the data.table object, and that setNumericRounding(0) is executed too,

Value
A data.table object with new columns date, time, micros, pt providing, respectively the data as IDate, time as ITime, microseconds (rounded) and numeric POSIXct.

Author(s)
Dirk Eddelbuettel

assignFormals  Assign formal arguments from function

Description
Debugging helper to assign formals from function

Usage
assignFormals(f, env = .GlobalEnv)
checkCRANStatus

Arguments

f          A function
env        An environment to assign the values to, defaults to global environment,

Details

(Beta) Attempt get default values from a given function, extracting its arguments and assigning
which should help in debugging via browser() and other helper functions.

Value

Nothing, but a side effect of assignment in global environment

Author(s)

Dirk Eddelbuettel

checkCRANStatus	Maintainer Status at CRAN

Description

Report Maintainer Status at CRAN

Usage

checkCRANStatus(email, cache, cache.life = 24 * 3600)

Arguments

eemail       A character variable with the maintainer email
ecache       A character variable with an optional cache file, default value is to use a file
icache.life  A numeric timeout, defaults to one day

Details

This function retrieves the maintainer status (given an email) at CRAN. Values are optionally
cached; the default cache location is inside the per-session temporary directory as R should not
write elsewhere.

Value

Nothing, the sideeffect of the display is the main effect

Author(s)

Brodie Gaslam (with minor modifications by Dirk Eddelbuettel)
**checkPackageAsciiCode**

Description

Check a package directory for non-ASCII characters in source files.

Usage

```r
checkPackageAsciiCode(dir, respect_quotes = FALSE)
```

Arguments

- `dir` Character variable with path to directory to be checked
- `respect_quotes` Logical variable whether quotes need to be checked

Details

The function is a renamed and slightly edited copy of the base R function `.check_package_ASCII_code`. It uses an unexported C function, also included, from base R, called as `_check_nonASCII`.

Value

A vector of things that are wrong per this function, also displayed on standard output

Author(s)

Dirk Eddelbuettel for this version, R Core for the underlying code

Examples

```r
## Not run:
checkPackageAsciiCode(".", FALSE)
## End(Not run)
```

---

**demotivate**

Display a demotivating quote

Description

Display a demotivating quote to remind the users of the harsh reality econometric and statistical practice.

Usage

```r
demotivate(x, width = NULL)
```
Arguments

- x: A index value chosen to select a quote; if not given a quote is chosen randomly.
- width: The desired display width.

Details

This function is a port of demotivate command for 'Stata' from the ado file at [http://fmwww.bc.edu/repec/bocode/d/demotivate.ado](http://fmwww.bc.edu/repec/bocode/d/demotivate.ado).

Value

The formatted is returned invisibly.

Author(s)

Dirk Eddelbuettel for this function, and Kevin Denny for the 'Stata' original.

See Also

[https://ideas.repec.org/c/boc/bocode/s458576.html](https://ideas.repec.org/c/boc/bocode/s458576.html)

Examples

demotivate()

gitRoot

getGitRoot

Are we in a git repository?

Description

The `getGitRoot()` function recursively ascends the filesystem tree from the given directory until it either finds a directory .git, or the top-level directory to abort the search. The root directory of the git repository is returned, with an empty string in the case of no repository. The `inGit()` function turns this into boolean predicate returning either TRUE or FALSE.

Usage

gitRoot(cwd = getwd())

inGit(cwd = getwd())

Arguments

- cwd: The start directory, default to the current working directory
**googleFinanceData**

**Value**

For `getGitRoot()`, the path of the directory containing the `.git` directory, i.e., the project root directory, or an empty string in case the search started outside a git directory. The `inGit()` function returns a boolean as to whether a git repository was found or not.

**Author(s)**

Dirk Eddelbuettel

**Examples**

`inGit()`

---

**googleFinanceData**  
*Historical Price Data Download from Google Finance*

**Description**

Download historical time series from Google Finance

**Usage**

```r
googleFinanceData(sym, current = TRUE, sy = 2005, sm = 1, sd = 1, ey, em, ed)
```

**Arguments**

- `sym`  
  A character string for a (tradeable) symbol
- `current`  
  A logical switch to indicate whether the current time is the endtime
- `sy`  
  An integer value for the start year, default is 2005
- `sm`  
  An integer value for the start month, default is 1
- `sd`  
  An integer value for the start date, default is 1
- `ey`  
  An optional integer value for the end year, required if `current` is false
- `em`  
  An optional integer value for the end month, required if `current` is false
- `ed`  
  An optional integer value for the end day, required if `current` is false

**Details**

The function uses an (unofficial) older CGI-style interface at Google to download historical data.

**Value**

A `data.table` object with a key on `date`
intradayMarketMonitor

Author(s)

Dirk Eddelbuettel

See Also

The post at https://chrisconlan.com/download-historical-stock-data-google-r-python/ provided the initial starting point

---

Description

Intra-day Market Monitor for Security Prices

Usage

intradayMarketMonitor(symbol = "^GSPC", defaultTZ = "America/Chicago")

Arguments

symbol A character variable with symbol understood by getQuote from package quantmod, default value is "^GSPC".
defaultTZ A character variable with the (local) timezone used for displaying the data, default value is "America/Chicago".

Details

This function periodically queries a public data source for a current price of given symbol and updates an intra-daily chart for that security. A working example is symbol "^GSPC" for the S&P500 index which can be obtained in real-time during (New York) trading hours. Other symbols may work.

An alternate version offering 24-hour coverage, for example for futures on Globex, is being prepared.

The function could be further generalized in numerous way and should be considered ‘alpha’. Current default values are a 15 second sleep, and fixed cut-off times for market open/close states corresponding to NYSE hours. The data history is reset to the two most recent days at the close, and the data is snapshot to file (with the filename derived from the symbol, and the path given by tools::R_user_dir). These parameters might become configuration parameters in the future.

Value

Nothing is returned, but a display of the current price and the recent history is updated, and the loops loops ‘forever’.
 isConnected

Author(s)
Dirk Eddelbuettel extending and refactoring the original code by Josh Ulrich

See Also
https://gist.github.com/joshuaulrich/ee11ef67b1461df399b84efd3c8f9f67#file-intraday-sp500-r

Examples
if (requireNamespace("quantmod", quietly)) {
  suppressMessages({
    library(xts); library(quantmod)
  })
  intradayMarketMonitor()
}

isConnected Is the current session (networked) and connected?

Description
Function to (heuristically) test for a network connection by attempting to connect to a given website.

Usage
isConnected(site = "https://www.google.com")

Arguments
site Character variable with site to try to connect to, defaults to http://www.google.com

Details
The main page of Google is used as a proxy for overall network connectivity as Google is generally 'network-close' and the page is relatively small. If a network is unavailable this fails generally already on domain name service resolution. Special thanks to Barry and Brodie for a very helpful discussion re-discovering this function.

Value
A boolean value indicating networking status

Author(s)
Dirk Eddelbuettel
**ls.objects**  
*List largest objects and show memory use*

### Description
List object and memory used by these objects

### Usage
```
ls.objects(pos = 1, pattern, order.by, decreasing = FALSE, head = FALSE,  
n = 5)
```
```
lsos(..., n = 10)
```
```
showMemoryUse(sort = "size", decreasing = FALSE, limit)
```

### Arguments
- **pos**: Position in search path, defaults to 1
- **pattern**: Pattern argument pass on to `ls`
- **order.by**: Optional sort order column
- **decreasing**: Optional switch for decreasing or increasing sort order
- **head**: Optional switch to show `head(..., n)`
- **n**: Number of elements to show, default to 5
- **...**: Passed through from `lsos` to `ls.objects`
- **sort**: Sort columns, defaults to `size`
- **limit**: Optional cap on displayed number of items

### Details
These helper functions have evolved over the years; some were also posted on StackOverflow in response to [https://stackoverflow.com/questions/1358003/tricks-to-manage-the-available-memory-in-an-r-s](https://stackoverflow.com/questions/1358003/tricks-to-manage-the-available-memory-in-an-r-s)

### Value
The displayed data.frame is returned to, the main purpose effect however is the displayed information

### Author(s)
Dirk Eddelbuettel, based on and extending code in mailing list posts by Petr Pikal and David Hinds
motivate

Display a motivating quote

Description
Display a motivating quote for users.

Usage
motivate(x, width = NULL)

Arguments
x
A index value chosen to select a quote; if not given a quote is chosen randomly.
width
The desired display width.

Details
This function is a port of motivate command for 'Stata' from the ado file at http://fmwww.bc.edu/repec/bocode/m/motivate.ado

Value
The formatted is returned invisibly.

Author(s)
Dirk Eddelbuettel for this function, and Kabira Namit for the 'Stata' original.

See Also
https://ideas.repec.org/c/boc/bocode/s458565.html

Examples
motivate()
muteTweeters  

*Mute Twitter Users with Excessive Hashtag Use*

Description

Mutes Twitter accounts using for than `ncrit` hashtags among `N` tweets in search of along with of `term`.

Usage

```
muteTweeters(term = "#rstats", N = 1000, ncrit = 10)
```

Arguments

- `term` A character variable to search for, defaults to `#rstats`
- `N` An number of tweets to fetch, defaults to 1000
- `ncrit` A number of hashtags after which use is deemed excessive, defaults to 10

Details

This is a modified version of the code in a wonderful tweet by Colin Gillespie (csgillespie) on 2020-August-26. It requires the *rtweet* and *data.table* packages.

Value

```
NULL, invisibly, but the function is invoked for the side effect of calling post_mute.
```

Author(s)

Dirk Eddelbuettel

Examples

```
## Not run:
## mute users with more than 10 hashtags among
## the 1000 most recent #rstats tweets
muteTweeters("#rstats", 1000, 10)

## End(Not run)
```
Description

Compute and display overbought and oversold regions

Usage

plotOBOS(symbol, n = 50, type = c("sma", "ema", "zlema", "hma"),
         years = 1, blue = TRUE, current = TRUE, title = symbol,
         ticks = TRUE, axes = TRUE)

Arguments

  symbol  A (required) character string for stock symbol, or alternatively a xts or zoo
           object with data
  n       An optional integer for the moving average length, defaults to 50
  type    An optional character string for the type of moving average; currently supported
           are "SMA", "EMA", "ZLEMA", and "HMA".
  years   An optional numeric or integer value for the number of years of data to display,
           defaults to one
  blue    An optional boolean determining whether blue or gray tones are used, defaults
           to true implying blue tones
  current An optional boolean determining whether the current date is the end date
  title   An optional character string for the plot title, defaults to the symbol
  ticks   An optional boolean indicating whether ticks are plotted, passed on to plot.xts,
           defaults to true
  axes    An optional boolean indicating whether axes are plotted, passed on to plot.xts,
           defaults to true

Details

This function computes a smoothed version of the price using a moving average (with one of several possible methods) as well as a standard deviation band, and displays one and two standard deviations around the smoothed price.

Value

  NULL as the function is invoked for the side effect of the plot

Author(s)

  Dirk Eddelbuettel
print.xts

Description
Print helper function for xts objects.

Usage
## S3 method for class 'xts'
print(x, ..., n = 10)

Arguments

- x: An xts object
- ...: Ignored, but needed for consistency with print generic.
- n: The default number of lines to display, default is 10.

Value
The return from print

Author(s)
Dirk Eddelbuettel

readAndConvert

Description
Read a file in a different encoding and return it as UTF-8 using iconv

Usage
readAndConvert(filename, encoding = "")

Arguments

- filename: Character variable with path a file with text in encoding
- encoding: Optional character variable with the encoding, if unset via the default empty string value no conversion is attempted.
rollMinMax

Details

The function is an adapted version of the one in the Rcpp Gallery post at https://gallery.rcpp.org/articles/iconv-via-r-header/ which is itself based on https://fishandwhistle.net/post/2021/using-rs-cross-platform-iconv-wrapper-from-cpp11/. It is however worth pointing out that 'iconv' results have been seen to vary across operating systems. While it is portable it does not guarantee identical outcomes across implementations: results on Windows have different from those on Unix OSs.

Value

A character variable with converted file content.

Usage

rollMinMax(x, window, min = TRUE)

Arguments

x A numeric vector.
window An interger with the size of the rolling window.
min A logical which, if true, selects minimum, else maximum

Details

The ichimoku variant is by Charlie Gao and credits Andrew Uhl for the initial implementation.

Value

A vector of the same length as 'x' with elements 1 to (length(window) - 1) containing NAs.

Author(s)

Dirk Eddelbuettel for this version, R Core for the underlying code
silent

Silently load a library

Description
Silently attach a library to the search path.

Usage
silent(...)

Arguments
... Passed though

Details
This function wraps suppressMessages around the call to library

Value
Nothing, but the desired library is attached

Author(s)
Dirk Eddelbuettel

wday

Return the day of the week as an integer

Usage
wday(date = Sys.Date())

Arguments
date A Date object, with the current date as the default

Details
This function simply wraps around the as.POSIXlt function and returns its wday field.

Value
A integer with the weekday component of the date
### ymd

**Description**

Format a Date(time) object as ymd

**Usage**

`ymd(pt)`

**Arguments**

- **pt** A POSIXt Datetime or a Date object

**Value**

A character object formatted as ‘YYYYMMDD’

**Author(s)**

Dirk Eddelbuettel

**Examples**

```r
if (requireNamespace("anytime", quietly=TRUE)) {
  ymd(anytime::anytime("2016-09-01 10:11:12.123456"))
  ymd(anytime::anydate("2016-Sep-01"))
}
```
Index

alphavantage, 2
as.data.table(as.data.table.xts), 3
as.data.table.xts, 3
assignFormals, 3

checkCRANstatus, 4
checkPackageAsciiCode, 5
demotivate, 5
getGitRoot, 6
googleFinanceData, 7

inGit(getGitRoot), 6
intradayMarketMonitor, 8
isConnected, 9

ls.objects, 10
lsos(ls.objects), 10

motivate, 11
muteTweeters, 12

plotOBOS, 13
print.xts, 14

readAndConvert, 14
rollMinMax, 15

showMemoryUse(ls.objects), 10
silent, 16

wday, 16

ymd, 17