

# Package ‘dang’

February 21, 2019

**Type** Package

**Title** 'Dang' Associated New Goodies

**Version** 0.0.10

**Date** 2019-02-11

**Author** Dirk Eddelbuettel with contributions by Brodie Gaslam, Kevin Denny,  
Kabira Namit, and others

**Maintainer** Dirk Eddelbuettel <edd@debian.org>

**Description** A collection of utility functions.

**Suggests** anytime, data.table, zoo, xts, TTR, quantmod

**License** GPL (>= 2)

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-02-21 15:20:03 UTC

## R topics documented:

alphavantage . . . . .	2
as.data.table.xts . . . . .	2
assignFormals . . . . .	3
checkCRANStatus . . . . .	4
demotivate . . . . .	4
googleFinanceData . . . . .	5
inGit . . . . .	6
ls.objects . . . . .	7
motivate . . . . .	8
plotOBOS . . . . .	9
print.xts . . . . .	10
silent . . . . .	10
ymd . . . . .	11
<b>Index</b>	<b>12</b>

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**

```
as.data.table.xts(x)
```

**Arguments**

x                      An xts object

**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	<i>Assign formal arguments from function</i>
---------------	--

---

**Description**

Debugging helper to assign formals from function

**Usage**

```
assignFormals(f, env = .GlobalEnv)
```

**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	<i>Maintainer Status at CRAN</i>
-----------------	----------------------------------

---

**Description**

Report Maintainer Status at CRAN

**Usage**

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

**Arguments**

email	A character variable with the maintainer email
cache	A character variable with an optional cache file, default value is to use a file inside the per-session temporary directory
cache.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)

---

demotivate	<i>Display a demotivating quote</i>
------------	-------------------------------------

---

**Description**

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

**Usage**

```
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>.

**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>

**Examples**

```
demotivate()
```

---

googleFinanceData      *Historical Price Data Download from Google Finance*

---

**Description**

Download historical time series from Google Finance

**Usage**

```
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

**Author(s)**

Dirk Eddelbuettel

**See Also**

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

---

inGit

*Are we in a git repository?*

---

**Description**

This 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.

**Usage**

```
inGit(cwd = getwd())
```

**Arguments**

`cwd`                   The start directory, default to the current working directory

**Details**

On Windows, `FALSE` is returned unconditionally.

**Value**

A boolean value indicating whether or not the given directory is, in fact, in a `git` repository.

**Author(s)**

Dirk Eddelbuettel

**Examples**

```
inGit()
```

---

ls.objects	<i>List largest objects and show memory use</i>
------------	---

---

**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 <http://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	<i>Display a motivating quote</i>
----------	-----------------------------------

---

**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()
```

---

plotOBOS

---

*Plot Overbought/Oversold Regions for A Stock or ETF*


---

**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`*Printing helper function for xts object*

---

**Description**

Print helper function for xts objects.

**Usage**

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

**Arguments**

<code>x</code>	An xts objects
<code>...</code>	Ignored, but needed for consistency with <code>print</code> generic.
<code>n</code>	The default number of lines to display, default is 10.

**Value**

The return from `print`

**Author(s)**

Dirk Eddelbuettel

---

`silent`*Silently load a library*

---

**Description**

Silently attach a library to the search path.

**Usage**

```
silent(...)
```

**Arguments**

<code>...</code>	Passed though
------------------	---------------

**Details**

This function wraps `suppressMessages` around the call to `library`

**Value**

Nothing, but the desired library is attached

**Author(s)**

Dirk Eddelbuettel

---

ymd

*Format a Date(time) object as 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**

```
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.xts`, [2](#)  
`assignFormals`, [3](#)  
  
`checkCRANStatus`, [4](#)  
  
`demotivate`, [4](#)  
  
`googleFinanceData`, [5](#)  
  
`inGit`, [6](#)  
  
`ls.objects`, [7](#)  
`lsos (ls.objects)`, [7](#)  
  
`motivate`, [8](#)  
  
`plotOBOS`, [9](#)  
`print.xts`, [10](#)  
  
`showMemoryUse (ls.objects)`, [7](#)  
`silent`, [10](#)  
  
`ymd`, [11](#)