

Package ‘DYM’

July 21, 2025

Type Package

Title Did You Mean?

Version 0.2

Date 2016-01-19

Author Kosei Abe [aut, cre], Richard Cotton [ctb]

Maintainer Kosei Abe <mail@recyclebin.jp>

Description Add a ``Did You Mean" feature to the R interactive. With this package, error messages for misspelled input of variable names or package names suggest what you really want to do in addition to notification of the mistake.

License BSD_3_clause + file LICENSE

BugReports <https://github.com/kos59125/DYM/issues>

Imports stats, utils

Suggests testthat

Encoding UTF-8

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-01-22 10:48:35

Contents

DYM	2
findSimilarName	2
getMissingVariable	3
getNames	4
isVariableName	4

Index	5
--------------	----------

DYM

DYM

Description

You might mistype an object name. The package suggests the correct spell of the object you meant.

If the function is called after an error of 'object not found', the function tries to tell you the name of the correct name that you meant.

Usage

```
DYM(threshold = 2, max_n = 10, ignoreCase = FALSE)
```

Arguments

threshold	The maximum distance between the misspell (x) and the correct answer (in name).
max_n	An integer limiting the number of results. Passed to head .
ignoreCase	A logical value indicating whether differences in case should be ignored when matching. Passed to adist .

Examples

```
## Not run:
options(error = DYM::DYM())
logg(10)

# For fewer or more suggestions, change threshold, max_n and ignoreCase
options(error = DYM::DYM(threshold = 3, max_n = 25, ignoreCase = TRUE))
logg(10)

## End(Not run)
```

findSimilarName

Looks for approximate matches to x (the first argument) within name (the second) argument.

Description

Looks for approximate matches to x (the first argument) within name (the second) argument.

Usage

```
findSimilarName(x, names, threshold = 2, max_n = 10, ignoreCase = FALSE)
```

Arguments

<code>x</code>	A string giving the (misspelt) name to search for.
<code>names</code>	A character vector of correct names to match against.
<code>threshold</code>	The maximum distance between the misspell (<code>x</code>) and the correct answer (in name).
<code>max_n</code>	An integer limiting the number of results. Passed to head .
<code>ignoreCase</code>	A logical value indicating whether differences in case should be ignored when matching. Passed to adist .

See Also

[adist](#) calculates the distance between strings. [agrep](#) and [stringdist-package](#) provide alternate metrics for these distances.

Examples

```
x <- "logg"
names <- DYM:::getNames(x)
# Increasing threshold increases the number of hits, upto max_n = 10
lapply(
  stats::setNames(0:4, 0:4),
  function(i) DYM:::findSimilarName(x, names, threshold = i)
)

# Use max_n = Inf to return all hits
DYM:::findSimilarName(x, names, threshold = 3, max_n = Inf)

# Negative max_n returns all hits except the last max_n
DYM:::findSimilarName(x, names, threshold = 3, max_n = -40)

# Set ignoreCase = TRUE to get more matches that differ by letter case
DYM:::findSimilarName(x, names, ignoreCase = TRUE)
```

`getMissingVariable` *Finds the misspelled object.*

Description

When this function is called after an error, it looks for the error message of missing value and returns the name of the mistype if it is found.

Usage

```
getMissingVariable()
```

getNames	<i>Retrieves available symbols in the specified environment.</i>
----------	--

Description

Retrieves available symbols in the specified environment.

Usage

```
getNames(mode, envir = .GlobalEnv)
```

Arguments

mode	The mode of misspelled name.
envir	The base environment to search variables.

isVariableName	<i>Checks if the given name is valid as a variable name for R.</i>
----------------	--

Description

Checks if the given name is valid as a variable name for R.

Usage

```
isVariableName(name)
```

Arguments

name	A character vector to check.
------	------------------------------

Index

adist, [2](#), [3](#)

agrep, [3](#)

DYM, [2](#)

DYM-package (DYM), [2](#)

findSimilarName, [2](#)

getMissingVariable, [3](#)

getNames, [4](#)

head, [2](#), [3](#)

isVariableName, [4](#)