

Package ‘dummies’

February 15, 2013

Type Package

Title Create dummy/indicator variables flexibly and efficiently

Version 1.5.6

Date 2012-06-14

Author Christopher Brown

Maintainer Christopher Brown <chris.brown@decisionpatterns.com>

Depends utils

Description Expands factors, characters and other eligible classes into dummy/indicator variables.

URL <http://www.decisionpatterns.com>

License GPL (>= 2)

LazyLoad no

Repository CRAN

Date/Publication 2012-06-14 18:17:00

NeedsCompilation no

R topics documented:

dummies-package	2
dummy	3
get.dummy	5
which.dummy	6

Index	8
--------------	----------

dummies-package

Create and manipulate dummy variables flexibly and efficiently

Description

Contains functions to create dummy variables flexibly using `model.matrix` returning them as either matrices or data frames for further analysis. Also, contains methods, for manipulating dummy variables.

Details

Package:	dummies
Type:	Package
Version:	1.5.6
Date:	2012-06-14
License:	GPL (>= 2)
LazyLoad:	no

Author(s)

Author: Christopher Brown

Maintainer: Christopher Brown

References

http://wiki.r-project.org/rwiki/doku.php?id=tips:data-manip:create_indicator

<http://blog.opendatagroup.com/2009/09/30/r-the-dummies-package>

See Also

[dummy](#)

Examples

```
# See examples in the documentation for dummy
```

 dummy

Flexible, efficient creation of dummy variables.

Description

This package flexibly and efficiently creates dummy variables for a variety of structures.

Usage

```
dummy(x, data = NULL, sep = "", drop = TRUE, fun = as.integer, verbose = FALSE)
```

```
dummy.data.frame(data, names = NULL, omit.constants=TRUE, dummy.classes = getOption("dummy.classes"))
```

Arguments

x	a single variable or variable <code>_name_</code>
data	an object such as a <code>data.frame</code> or matrix that has <code>colnames</code>
drop	Whether to drop (i.e. omit) dummy variables for unused levels. When <code>x</code> or <code>data[,x]</code> is a factor, this parameter variables for only the used levels. By default, dummies are created only for the used levels, i.e. <code>TRUE</code> .
sep	For the names of the created dummy variables, <code>sep</code> is the character used between the variable name and the value.
fun	Function used to coerce values in the resulting matrix or frame.
verbose	logical. Whether to <code>print(cat)</code> the number of dummy variables created Default: <code>FALSE</code> For <code>dummy.data.frame</code> only:
names	The names of the columns to expand to dummy variables. Takes precedent over <code>dummy.classes</code> parameter.
dummy.classes	(For <code>dummy.data.frame</code> only) A vector of classes names for which dummy variables are created -or- "ALL" to create dummy variables for all columns ir-regardless of type. By default, dummy variables are produced for factor and character class and be modified globally by <code>options('dummy.classes')</code> .
omit.constants	Whether to omit dummy variables that are constants, i.e. contain only one value. Overridden by <code>drop==FALSE</code> .
all	(For <code>dummy.data.frame</code> only). Whether to return columns that are not dummy classes. The default is <code>TRUE</code> and returns all classes. Non dummy classes are untouched.
...	arguments passed to dummy

Details

`dummy` takes a single variable OR the name of single variable and a data frame. It coerces the variable to a factor and returns a matrix of dummy variables using `model.matrix`. If the data has rownames, these are retained.

Optionally, the parameter `drop` indicates that that dummy variables will be created for only the expressed levels of factors. Setting it to false will produce dummy variables for all levels of all factors.

If there is only one level for the variable and `verbose == TRUE`, a warning is issued before creating the dummy variable. Each element of this dummy variable, will have the same value.

A separator, `sep`, can be specified for the separator between the variable name and the value for the construction of new variable names. The default is to provide no separator.

The type of values returned can be affected using the `fun` argument. `fun` is called on each of the resultant dummy variables. The only useful functions that the author has employed are `as.integer` (the default) or `as.logical`.

`dummy.data.frame` takes a `data.frame` or matrix and returns a `data.frame` in which all specified columns are expanded as dummy variables. Specific columns can be named with the `names` argument or specified on a class basis by the `dummy.classes` argument. Specified names take precedent over classes. The default is to expand dummy variables for character and factor classes, and can be controlled globally by `options('dummy.classes')`.

If the argument `all` is FALSE. The resulting `data.frame` will contain only the new dummy variables. By default, all columns of the object are returned in the order of the original frame. Dummy variables are expanded in place.

`omit.constants` indicates whether to omit dummy variables that assume only a single value. This is the default. If `drop==FALSE`, constant variables are retained regardless of the setting.

Value

`dummy` returns a matrix with the number of rows equal to the that of given variable. By default, the matrix contains integers, but the exact type can be affected by `fun` argument. Rownames are retained if the supplied variable has associated row names.

`dummy.data.frame` returns a `data.frame` in which variables are expanded to dummy variables if they are one of the dummy classes. The columns are returned in the same order as the input with dummy variable columns replacing the original column.

Author(s)

Christopher Brown

References

http://wiki.r-project.org/rwiki/doku.php?id=tips:data-manip:create_indicator
<http://tolstoy.newcastle.edu.au/R/help/00b/1199.html>
<http://tolstoy.newcastle.edu.au/R/help/03a/6409.html>
<http://tolstoy.newcastle.edu.au/R/help/01c/0580.html>
 Many other discussions on R-Help. Too many to list.

See Also

[model.frame](#), [model.matrix](#), [factor](#)

Examples

```

letters <- c( "a", "a", "b", "c", "d", "e", "f", "g", "h", "b", "b" )
dummy( as.character(letters) )
dummy( letters[1:6] )

l <- as.factor(letters)[ c(1:3,1:6,4:6) ]
dummy(l)
dummy(l, drop=FALSE)
dummy(l, sep=":")
dummy(l, sep="::", fun=as.logical)

# TESTING NAS
l <- c( NA, 1, NA)
dummy(l)
dummy(l, sep=":")

dummy(iris$Species)
dummy(iris$Species[ c(1:3,51:53,101:103) ] )
dummy(iris$Species[ c(1:3,51:53,101:103) ], sep=":" )
dummy(iris$Species[ c(1:3,51:53) ], sep=":", drop=FALSE )

# TESTING TRAP FOR ONE LEVEL
dummy( as.factor(letters)[c(1,1,1,1)] )
dummy( as.factor(letters)[c(1,1,2,2)] )
dummy( as.factor(letters)[c(1,1,1,1)] , drop = FALSE )

dummy.data.frame(iris)
dummy.data.frame(iris, all=FALSE)

dummy.data.frame(iris, dummy.class="numeric" )
dummy.data.frame(iris, dummy.class="ALL" )

```

get.dummy

Get the dummy variable columns from a data frame.

Description

Given a data.frame and an optional variables name, return only the columns that are dummy variables

Usage

```
get.dummy(data, name = NULL)
```

Arguments

data	A data.frame with an dummies attributes
name	Optional. The name of a variable.

Details

This uses [which.dummy](#) to identify the dummy columns.

Value

The subset of data that are dummy columns.

Author(s)

Christopher Brown

See Also

[which.dummy](#), [dummy.data.frame](#)

Examples

```
data( iris )
d <- dummy.data.frame( iris )
get.dummy( d, 'Species' )
```

which.dummy

Identify which columns are dummy variables on a data frame.

Description

Given a data frame and an optional variable name, `which.dummy` identifies which columns are dummy variables by the column index.

Usage

```
which.dummy(data, name=NULL)
```

Arguments

data	An object with a 'dummies' attribute
name	Optional. The name of a column that has been expanded to a dummy variable.

Details

Given a data frame and an optional variable name, returns the indices of the dummy columns.

Requires an `dummies` attribute, usually created by [dummy.data.frame](#).

Value

integer vector of column indices corresponding to the dummy variable(s)

Author(s)

Christopher Brown

See Also

[dummy.data.frame](#), [dummy](#)

Examples

```
data(iris)
dat <- dummy.data.frame(iris)
which.dummy(dat)
```

Index

- *Topic **attribute**
 - `which.dummy`, 6
- *Topic **manip**
 - `dummy`, 3
 - `get.dummy`, 5
 - `which.dummy`, 6
- *Topic **models**
 - `dummy`, 3
- *Topic **package**
 - `dummies-package`, 2

`dummies (dummies-package)`, 2

`dummies-package`, 2

`dummy`, 2, 3, 3, 7

`dummy.data.frame`, 6, 7

`factor`, 5

`get.dummy`, 5

`model.frame`, 5

`model.matrix`, 4, 5

`which.dummy`, 6, 6