Package ‘hypercube’

December 15, 2017

Type Package
Title Organizing Data in a Hypercube
Version 0.1.0
Author Michael Scholz
Maintainer Michael Scholz <michael.scholz@uni-passau.de>
Description Provides methods for organizing data in a hypercube
(i.e. a multi-dimensional cube). Cubes are generated from molten data frames.
Each cube can be manipulated with five operations: rotation (changeDimensionOrder()),
dicing and slicing (add.selection(), remove.selection()), drilling down (add.aggregation()),
and rolling up (remove.aggregation()).
License GPL-3
Encoding UTF-8
Depends R (>= 3.3.0), stats
Imports methods, stringr
LazyData TRUE
RoxygenNote 6.0.1
NeedsCompilation no
Repository CRAN
Date/Publication 2017-12-15 09:58:18 UTC

R topics documented:

hypercube-package .................................................. 2
add.aggregation ......................................................... 3
add.selection .......................................................... 4
as.data.frame.Cube ................................................... 5
changeDimensionOrder ............................................... 6
Cube-class .............................................................. 7
Dimension-class ....................................................... 7
generateCube .......................................................... 8
remove.aggregation .................................................... 9
hypercube-package

Provides methods for organizing data in a hypercube

Description

This package provides methods for organizing data in a hypercube. Each cube can be manipulated with five operations: rotation (changeDimensionOrder), dicing and slicing (add.selection, remove.selection), drilling down (add.aggregation), and rolling up (remove.aggregation).

Details

Package: hypercube
Type: Package
Version: 0.1.0
Date: 2017-12-14
License: GPL-3
Depends: R (>= 3.0), methods

Author(s)

Michael Scholz <michael.scholz@uni-passau.de>

Examples

# Simple example
data("sales")
cube = generateCube(sales, columns = list(time = c("month", "year"),
                               location = c("state"), product = "product"), valueColumn = "amount")
cube

# More sophisticated example
data("sales")
cube = generateCube(sales, columns = list(time = c("month", "year"),
                                           location = c("state"), product = "product"), valueColumn = "amount")
cube = add.selection(cube, criteria = list(state = c("AL", "TX")))
cube = add.aggregation(cube, dimensions = c("month", "year"), fun = "sum")
cube
add.aggregation

df = as.data.frame(cube)
df

add.aggregation

Adds an aggregation to a hypercube

Description
This function adds a further aggregation to a hypercube. The cube itself will not be changed. The aggregation only affect the data that will be shown when printing the cube. Note that selection criteria will be applied before the aggregating the data.

Usage
add.aggregation(cube, dimensions, fun = c("sum", "min", "max", "prod", "mean", "median", "sd", "count"))

Arguments
cube Hypercube for which the selection criteria will be defined.
dimensions A vector of dimensions that are used in the aggregation.
fun The function that is used for aggregation. Possible functions are sum, prod, min, max, mean, median, sd, and count.

Value
Returns a Cube object with the added aggregation.

Author(s)
Michael Scholz <michael.scholz@uni-passau.de>

See Also
Cube remove.aggregation add.selection

Examples

data("sales")
cube = generateCube(sales, columns = list(time = c("month", "year"), location = c("state"), product = "product"), valueColumn = "amount")
cube = add.aggregation(cube, dimensions = c("month", "year"), fun = "sum")
cube
add.selection  

Adds selection criteria to a hypercube

**Description**

This function adds further selection criteria to a hypercube. The cube itself will not be changed. The selection criteria only affect the data that will be shown when printing the cube. Note that selection criteria will be applied before the aggregating the data.

**Usage**

add.selection(cube, criteria)

**Arguments**

- **cube**: Hypercube for which the selection criteria will be defined.
- **criteria**: A list of selection criteria.

**Value**

Returns a Cube object with the added selection criteria.

**Author(s)**

Michael Scholz <michael.scholz@uni-passau.de>

**See Also**

Cube remove.selection add.aggregation

**Examples**

data("sales")
print(str(sales))
cube = generateCube(sales, columns = list(time = c("month", "year"),
    location = c("state"), product = "product"), valueColumn = "amount")
cube = add.selection(cube, criteria = list(state = c("CA", "FL")))
cube
cube = add.selection(cube, criteria = list(state = c("TX")))
cube
as.data.frame.Cube

Converts the actual view of a cube to a data frame

Description

Converts the actual view of a Cube object to a data frame. All added selections and aggregations will be regarded. Note that selection criteria will be applied before the aggregating the data.

Usage

```r
## S3 method for class 'Cube'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

- `x` The Cube object that will be converted to a data frame.
- `row.names` A character vector giving the row names for the data frame.
- `optional` Should setting row names and converting column names be optional?
- `...` Further parameters that are passed to `as.data.frame.table`.

Value

A molten data frame

Author(s)

Michael Scholz <michael.scholz@uni-passau.de>

See Also

`add.aggregation` `add.selection`

Examples

```r
data("sales")
cube = generateCube(sales, columns = list(time = c("month", "year"),
location = c("state"), product = "product"), valueColumn = "amount")
cube = changeDimensionOrder(cube, dimensions = c("product", "month", "year", "state"))
df = as.data.frame(cube)
df
```
changeDimensionOrder  Changes the order of the dimensions in a given cube

Description

Changes the order of the dimensions in a given cube

Usage

changeDimensionOrder(cube, dimensions)

Arguments

cube  Hypercube for which the dimension should be re-ordered.
dimensions  Vector of dimensions. The order of the dimensions in this vector defines the order of the dimensions in the cube.

Value

Returns a Cube object.

Author(s)

Michael Scholz <michael.scholz@uni-passau.de>

See Also

Cube

Examples

data("sales")
cube = generateCube(sales, columns = list(time = c("month", "year"),
    location = c("state"), product = "product"), valueColumn = "amount")
cube = changeDimensionOrder(cube, dimensions = c("product", "month", "year", "state"))
cube
Cube-class

Class "Cube"

Description

Class "Cube"

Slots

data (array) The data that are represented as hypercube.
structure (list) The structure of the dimensions of the hypercube.
view (list) Information about how to build a view for the hypercube. This information is stored in a list of Dimension-class objects.

Objects from the Class

Objects can be created by calls of the form new("Cube", ...). This S4 class describes Cube objects.

Author(s)

Michael Scholz <michael.scholz@uni-passau.de>

See Also

generateCube

Examples

# show Cube definition
showClass("Cube")

Dimension-class

Class "Cube"

Description

Class "Cube"

Slots

name (character) The name of the dimension.
values (vector) A vector of selected values for this dimension.
aggregation (vector) A vector of aggregation functions that will be applied to this dimension.
Objects from the Class

Objects can be created by calls of the form `new("Dimension", ...)`. This S4 class describes Dimension objects.

Author(s)

Michael Scholz <michael.scholz@uni-passau.de>

Examples

```r
# show Dimension definition
showClass("Dimension")
```

---

**generateCube**

*Generates a hypercube from a given dataframe*

Description

This function generates a hypercube from a given dataframe. The dimensions of the hypercube correspond to a set of selected columns from the dataframe.

Usage

```r
generateCube(data, columns, valueColumn)
```

Arguments

- `data`: A dataframe that is used as source for the hypercube.
- `columns`: A vector of column names that will form the dimensions of the hypercube.
- `valueColumn`: The name of the column that provides the values for the cells of the hypercube.

Value

Returns a Cube object.

Author(s)

Michael Scholz <michael.scholz@uni-passau.de>

See Also

Cube
remove.aggregation

Examples

data("sales")
cube = generateCube(sales, columns = list(time = c("month", "year"),
location = c("state"), product = "product"), valueColumn = "amount")

cube

cube = add.aggregation(cube, dimensions = c("month", "year"), fun = "sum")
cube

cube = add.aggregation(cube, dimensions = "year", fun = "sum")
cube

Description

This function removes aggregations from a hypercube. The cube itself will not be changed. The aggregation only affect the data that will be shown when printing the cube.

Usage

remove.aggregation(cube, dimensions = NA, last = FALSE)

Arguments

cube Hypercube from which the aggregation will be removed.
dimensions A vector of dimensions for which the aggregations will be removed.
last Should the last aggregation be removed? If this parameter is set TRUE, the dimension vector will be ignored.

Value

Returns a Cube object with the added aggregation.

Author(s)

Michael Scholz <michael.scholz@uni-passau.de>

See Also

Cube add.aggregation remove.selection

Examples

data("sales")
cube = generateCube(sales, columns = list(time = c("month", "year"),
location = c("state"), product = "product"), valueColumn = "amount")

cube

cube = add.aggregation(cube, dimensions = c("month", "year"), fun = "sum")
cube

cube = add.aggregation(cube, dimensions = "year", fun = "sum")
cube
remove.selection

Description

This function removes all selection criteria for the given dimensions. The cube itself will not be changed. The selection criteria only affect the data that will be shown when printing the cube.

Usage

```
remove.selection(cube, dimensions)
```

Arguments

- `cube`: Hypercube for which the selection criteria will be defined.
- `dimensions`: A vector of dimension names for which all selection criteria will be removed.

Value

Returns a Cube object with removed selection criteria.

Author(s)

Michael Scholz <michael.scholz@uni-passau.de>

See Also

Cube add.selection remove.aggregation

Examples

```r
cube = remove.aggregation(cube, dimensions = "year")
cube
```

```
remove.selection      Removes selection criteria from a hypercube
```

```r
cube = remove.selection(cube, dimensions = c("state"))
cube
```
### Description

A dataset containing 2,500 sales of 4 books in different states and countries.

### Usage

sales

### Format

A data frame with 2,500 rows and 7 variables:

- **month**: month as number
- **year**: year as number
- **state**: abbreviation of the state as character
- **country**: country as character
- **product**: name of the product as character
- **unit**: number of sold products
- **amount**: amount of sales

### Source

Synthetic dataset

---

### show, Cube-method

**Shows a Cube object**

### Description

Shows the actual view of a Cube object. All added selections and aggregations will be regarded. Note that selection criteria will be applied before the aggregating the data.

### Usage

```r
## S4 method for signature 'Cube'
show(object)
```

### Arguments

- **object**: The Cube object
Author(s)
Michael Scholz <michael.scholz@uni-passau.de>

See Also
Cube

Examples

data("sales")
cube = generateCube(sales, columns = list(time = c("month", "year"),
location = c("state"), product = "product"), valueColumn = "amount")
cube

data("sales")
cube@view[[1]]
Index

*Topic classes
  Cube-class, 7
  Dimension-class, 7
*Topic datasets
  sales, 11
*Topic data
  hypercube-package, 2
*Topic hypercube,
  hypercube-package, 2
*Topic hyperdimensional
  hypercube-package, 2

add.aggregation, 3, 4, 5, 9
add.selection, 3, 4, 5, 10
as.data.frame.Cube, 5
as.data.frame.table, 5

changeDimensionOrder, 6
Cube, 3, 4, 6, 8–10, 12
Cube-class, 7

Dimension-class, 7

generateCube, 7, 8

hypercube (hypercube-package), 2
hypercube-package, 2

remove.aggregation, 3, 9, 10
remove.selection, 4, 9, 10

sales, 11
show,Cube-method, 11
show,Dimension-method, 12