Package ‘cumplyr’

August 29, 2016

Type Package

Title Extends ddply to allow calculation of cumulative quantities.

Version 0.1-1

Date 2012-05-02

Author John Myles White

Maintainer John Myles White <`jmw@johnmyleswhite.com`>

Description Extends ddply to allow calculation of cumulative
quantities.

License MIT

Repository CRAN

Date/Publication 2012-05-14 15:58:39

NeedsCompilation no

R topics documented:

`cumplyr-package` *Extends ddply to allow calculation of cumulative quantities.*

Description

Extends ddply to allow calculation of cumulative quantities.

Details

Package:	cumplyr
Type:	Package
Version:	0.1-1
Date:	2012-05-02
License:	MIT

Author(s)

John Myles White

Maintainer: John Myles White <jmw@johnmyleswhite.com>

Examples

```
library('cumplyr')

data(rt.data)

results <- cumddply(rt.data,
                    c('Subject', 'Block'),
                    c('Trial'),
                    function (df) {with(df, mean(RT))})

print(results)
```

---

cartesian_product *Compute the Cartesian product of named variables.*

Description

Compute the Cartesian product of named variables.

Usage

```
cartesian_product(variable.names, envir = parent.frame())
```

Arguments

`variable.names`	Character vector of names of variables
`envir`	The environment in which to find names

Value

The Cartesian product of all variables

Examples

```
library('cumplyr')
x <- 1:2
y <- 10:11
cartesian_product(c('x', 'y'))

tmp.env <- new.env()
assign('x', 1:3, envir = tmp.env)
assign('y', 2:4, envir = tmp.env)
cartesian_product(c('x', 'y'), envir = tmp.env)
```

`cumddply` *Cumulative ddply*

Description

Cumulative ddply

Usage

```
cumddply(data, equality.variables, inequality.variables, func)
```

Arguments

`data`	Data to process
`equality.variables`	Character vector variables used to split data on equality
`inequality.variables`	Character vector variables used to split data on inequality
`func`	Function to call on each split of the data

Value

Data frame with cumulative results combined across splits

Examples

```
library('cumplyr')

data(rt.data)

results <- cumddply(rt.data,
                    c('Subject', 'Block'),
                    c('Trial'),
                    function (df) {with(df, mean(RT))})

print(results)
```

`iddply` *ddply with inequality constraints*

Description

ddply with inequality constraints

Usage

```
iddply(data,
       equality.variables,
       lower.bound.variables,
       upper.bound.variables,
       norm.ball.variables,
       func)
```

Arguments

`data`	Data to process
`equality.variables`	Character vector of variables used to split data on equality
`lower.bound.variables`	Character vector of variables used to split data on lower bound inequalities
`upper.bound.variables`	Character vector of variables used to split data on upper bound inequalities
`norm.ball.variables`	Character vector of variables used to split data on norm ball inequalities
`func`	Function to call on each split-out subset of the data

Value

Data frame with results combined across splits

Examples

```
library('cumplyr')

data <- data.frame(Time = 1:5, Value = seq(1, 9, by = 2))

iddply(data,
       equality.variables = c('Time'),
       lower.bound.variables = c(),
       upper.bound.variables = c(),
       norm.ball.variables = list(),
       func = function (df) {with(df, mean(Value))})

iddply(data,
       equality.variables = c(),
       lower.bound.variables = c('Time'),
       upper.bound.variables = c(),
       norm.ball.variables = list(),
       func = function (df) {with(df, mean(Value))})

iddply(data,
       equality.variables = c(),
       lower.bound.variables = c(),
       upper.bound.variables = c('Time'),
       norm.ball.variables = list(),
       func = function (df) {with(df, mean(Value))})

iddply(data,
       equality.variables = c(),
       lower.bound.variables = c(),
       upper.bound.variables = c(),
       norm.ball.variables = list('Time' = 1),
       func = function (df) {with(df, mean(Value))})

iddply(data,
       equality.variables = c(),
       lower.bound.variables = c(),
       upper.bound.variables = c(),
       norm.ball.variables = list('Time' = 2),
       func = function (df) {with(df, mean(Value))})

iddply(data,
       equality.variables = c(),
       lower.bound.variables = c(),
       upper.bound.variables = c(),
       norm.ball.variables = list('Time' = 5),
       func = function (df) {with(df, mean(Value))})
```

processed.rt.data *Processed rt.data*

Description

Processed rt.data

Usage

```
data(processed.rt.data)
```

Format

A data frame with 12 observations on the following 4 variables.

`Subject` a numeric vector

`Block` a numeric vector

`Trial` a numeric vector

`CumMeanRT` a numeric vector

Examples

```
data(processed.rt.data)
```

`rt.data` *rt.data*

Description

RT data

Usage

```
data(rt.data)
```

Format

A data frame with 12 observations on the following 4 variables.

`Subject` Subject ID number

`Block` Block ID

`Trial` Trial ID

`RT` RT measurement

Examples

```
data(rt.data)

with(rt.data, mean(RT))
```

Index