Package ‘zoocat’

September 11, 2018

Type Package
Title 'zoo' Objects with Column Attributes
Version 0.2.0.1
Date 2016-11-10
Author Ran-Ran He [aut, cre]
Maintainer Ran-Ran He <heranran2006@163.com>
Description Tools for manipulating multivariate time series data by extending 'zoo' class.
Depends R(>= 3.1.1), zoo, reshape2
Imports graphics, utils, stats, plyr, scales, grDevices
Suggests ggplot2, knitr, rmarkdown
License GPL-3
URL https://github.com/ran-ran/zoocat
RoxygenNote 5.0.1
VignetteBuilder knitr
NeedsCompilation no
Repository CRAN
Date/Publication 2018-09-11 09:37:51 UTC

R topics documented:

aggregate_col .......................... 2
apply_col .................................. 3
apply_core ................................ 4
as.gmon .................................. 5
as.zoo .................................. 5
as.zoocat ................................ 6
as.zoomly ................................ 7
cast2zoocat .............................. 8
cast2zoomly ............................... 9
aggregate_col

Aggregate columns of a "zoocat" object by given column attributes fields.

Usage

aggregate_col(x, by = colnames(cattr(x)), FUN = mean, ...)

Arguments

x
by
FUN
...  a "zoocat" object.
  a character string indicates the field of column attributes.
a function to be applied to all data subsets.
  additional arguments to be passed to the method.

Value

a "zoocat" object.
### Examples

```r
# This is the air quality example from package reshape2
names(airquality) <- tolower(names(airquality))
aqm <- melt(airquality, id = c("month", "day"), na.rm=TRUE)
zc <- cast2zoocat(aqm, index.var = 'month', value.var = 'value', fun.aggregate = mean)
aggregate_col(zc, by = 'variable', FUN = max)
aggregate_col(zc, by = 'variable', FUN = max, na.rm = TRUE)
```

---

### Description

Apply a function over each column of a "zoocat" object and return a data frame.

### Usage

```r
apply_col(x, ...)  
```

### Arguments

- `x` A object.
- `...` Additional arguments to be passed to or from methods.
- `FUN` The function apply for each column.
- `col.as` If vector, each column will be treated as a vector. If zoo, each column will be treated as a zoo object.

### Value

A data frame.

### Examples

```r
x <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = c(rep('xxx', 3), 'yyy'))
zc <- zoocat(x, order.by = 1991:1995, colattr = colAttr)
apply_col(zc, FUN = mean, col.as = 'vector')
apply_col(zc, FUN = max, col.as = 'vector')
```
**apply_core**

*Apply a function over the core data matrix*

**Description**

Apply a function over the core data matrix of the zoocat object, and bind the return data with `cattr` or `index`.

**Usage**

```r
apply_core(x, ...)
```

```r
## S3 method for class 'zoocat'
apply_core(x, FUN, bind, ...)
```

**Arguments**

- `x`: the object.
- `...`: other arguments for `FUN`.
- `FUN`: the function to apply. The `FUN` must return a matrix or a vector.
- `bind`: a vector of length 1 or 2 with element values to be `cattr` or `index` or NA to describe how to bind the return data with `cattr` or `index`. If `FUN` return a vector, set `bind` to be a scalar. If `FUN` return a matrix, set `bind` to be a vector of length 2. See details.

**Value**

a data frame, a "zoo" object, or a "zoocat" object.

**Examples**

```r
x <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = c(rep('xxx', 3), 'yyy'))
zc <- zoocat(x, order.by = 1991:1995, colattr = colAttr)

apply_core(zc, FUN = colMeans, bind = 'cattr')
apply_core(zc, FUN = rowMeans, bind = 'index')
apply_core(zc, FUN = function (x) {x*2}, bind = c('index', 'cattr'))
apply_core(zc, FUN = function (x) {t(x*2)}, bind = c('cattr', 'index'))
apply_core(zc, FUN = function (x) {x[3:4, ]}, bind = c('cattr'))
apply_core(zc, FUN = function (x)
    {r <- x[3:4, ]
     rownames(r) <- c('a', 'b')
     return(r)},
    bind = c('cattr', 'cattr'))

vec <- as.vector(zc[, 1])
```
apply_core(zc, FUN = function (x) {cor(x, vec), bind = 'cattr'}

**as.gmon**  
Coercion from and to gmon class

**Description**  
Coercion from and to gmon class

**Usage**  
```
as.gmon(x, ...)```

**Arguments**

- **x**
  - an numeric vector.
- **...**
  - unused.

**Value**

a "gmon" object.

**as.zoo**  
Coercion objects to class zoo

**Description**

Coercing objects to class zoo.

**Usage**

```
## S3 method for class 'zoocat'
as.zoo(x, add.colname = TRUE, ...)
```

**Arguments**

- **x**
  - an object.
- **add.colname**
  - logical. If TRUE, column names will be added automatically.
- **...**
  - further arguments.

**Details**

For `zoomly` and `zoocat`, the returned `zoo` object will be added column names automatically. Note that the result of `as.zoo` will be a `zoo` object if the input `x` is inhered from `zoo`. 

as.zoocat

Coercing objects to Class zoocat

Description

Coercing objects to class zoocat.

Usage

as.zoocat(x, ...)

## S3 method for class 'zoomly'
as.zoocat(x, ...)

## S3 method for class 'zoo'
as.zoocat(x, colattr = NULL, variable.name = "variable",
         index.name = "index", ...)

Arguments

x  the object.
...
other arguments passed to methods.
colattr a data frame the column attribute table for x.
variable.name the name of the field in the cattr of the output zoocat object to store the var-
               able name.
index.name   the name of the index variable.

Value

a "zoocat" object.

Examples

x <- matrix(1:20, nrow = 5)
zc <- zoocat(x, order.by = 1991:1995, colattr = data.frame(month=c(2, 3, 5, 6)))
z <- as.zoo(zc)
Examples

```r
x <- matrix(1:20, nrow = 5)
zm <- zoomly(x, order.by = 1991:1995, colattr = data.frame(month = c(2, 3, 5, 6)))
as.zoocat(zm)

zobj <- zoocat(matrix(1:10, nrow = 5), order.by = 11:15)
colnames(zobj) <- c('a', 'b')
as.zoocat(zobj)
```

Description

Coercion objects to class "zoomly". The index name of the object will be set to "year".

Usage

```r
as.zoomly(x, ...)
```

## S3 method for class 'zoocat'
```r
as.zoomly(x, ...)
```

Arguments

- `x`: an object.
- `...`: further arguments passed to methods.

Value

A "zoomly" object.

Examples

```r
x <- matrix(1:36, nrow = 3)
zc <- zoocat(x, order.by = 1991:1993,
             colattr = data.frame(month = 1:12))
as.zoomly(zc)

x <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = c('xx', 3), 'yy'))
zc <- zoocat(x, order.by = 1991:1995, colattr = colAttr)
as.zoomly(zc)
```
**cast2zoocat**  
*Cast a data frame to a zoocat object*

**Description**

Cast a data frame to a "zoocat" object. The data frame should be a long format data frame (for example, melted by `reshape2::melt`).

**Usage**

```r
cast2zoocat(x, index.var, value.var, attr.var = NULL, fun.aggregate = NULL, del.unique.cattr = TRUE)
```

**Arguments**

- `x`: a data frame.
- `index.var`: the name of the column to be treated as the index of the zoocat object.
- `value.var`: the name of the column which stored the values.
- `attr.var`: the name of the column which will be used as column attributes of the zoocat object. If NULL, all columns except `value.var` and `index.var` will be used.
- `fun.aggregate`: aggregation function needed if variables do not identify a single observation for each output cell. Defaults to `length` (with a message) if needed but not specified. See `dcast`.
- `del.unique.cattr`: logical. If TRUE, the column attributes with unique value will be deleted.

**Value**

a "zoocat" object.

**Examples**

```r
df <- data.frame(year = rep(1991:1995, each = 24), month = rep(1:12, 10), varname = rep(c('a', 'b'), each = 12), city = rep(1:3, each = 40), value = 1:120)
cast2zoocat(df, index.var = 'year', value.var = 'value')
cast2zoocat(df, index.var = 'year', value.var = 'value', attr.var = 'varname')

## This is the air quality example from package reshape2
names(airquality) <- tolower(names(airquality))
aqm <- melt(airquality, id = c("month", "day"), na.rm=TRUE)
head(aqm)
cast2zoocat(aqm, index.var = 'month', value.var = 'value', attr.var = 'variable')
cast2zoocat(aqm, index.var = 'month', value.var = 'value')
```
**cast2zoomly**

*Cast a data frame to a *zoomly* object*

**Description**

Cast a data frame to a "*zoomly*" object.

**Usage**

`cast2zoomly(x, ...)`

**Arguments**

- `x` a data frame containing a column named "month".
- `...` other arguments for `cast2zoocat`.

**Value**

a "*zoomly*" object.

**Examples**

```r
df <- data.frame(year = rep(1991:1995, each = 24), month = rep(1:12, 10),
                 varname = rep(c('a', 'b'), each = 12),
                 city = rep(1:3, each = 40),
                 value = 1:120)
cast2zoomly(df, index.var = 'year', value.var = 'value')
```

---

**cast_month**

*Cast month of a *zoomly* object*

**Description**

Cast month of a *zoomly object*

**Usage**

`cast_month(x)`

**Arguments**

- `x` a zoocat object.
**Examples**

```r
x <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = c(rep('xxx', 3), 'yyy'))
zc <- zoocat(x, order.by = 1991:1995, colattr = colAttr)
zmelt <- melt_month(as.zoomly(zc))
cast_month(zmelt)
```

---

**cattr**

*Getting or setting the column attributes table of a `zoocat` object.*

---

**Description**

Getting or setting the column attributes table of a "zoocat" object. Using `cattr(x) <- value` can translate a "zoo" object to a "zoocat" object.

**Usage**

```r
cattr(x) <- value
```

---

**Arguments**

- **x**: A zoocat or zoo object.
- **value**: The new value.

---

**Value**

"cattr(x)" return a data frame.

---

**Examples**

```r
x <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = 'x')
zc <- zoocat(x, order.by = 1991:1995, colattr = colAttr)
cattr(zc)[, 2] <- 'y'
```
cor

zobj <- zoo(x, order.by = 1991:1995)
cattr(zobj) <- colAttr

---

Correlation computing for zoo or zoocat objects

Description
This function is based on the function stats::cor. For cor.zoo and cor.zoocat, the equality of the index of x and y will be checked (if y is not NULL). For cor.zoocat, if y has one column, the result will be binded with the cattr table and a data frame will be returned.

Usage

cor(x, y = NULL, ...)

## S3 method for class 'zoo'
cor(x, y = NULL, ...)

## S3 method for class 'zoocat'
cor(x, y = NULL, ...)

Arguments

x A zoo or zoocat object.
y NULL or a zoo or zoocat object. If NULL, x will be used.
... Other arguments for function stats::cor.

Value

a vector, matrix or data frame.

Examples

x <- zoo(c(1, 3, 2, 4, 5))
y <- zoo(c(12, 30, 2, 46, 5))
cor(x, y)

x <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = 'x')
zc <- zoocat(x, order.by = 1991:1995, colattr = colAttr)
y <- zoo(c(3, 5, 4, 6, 3), order.by = 1991:1995)
cor(zc, y, method = 'kendall')
cor(zc, method = 'kendall')
filter_col

Return columns with matching conditions

Description

Return columns with matching conditions for the column attributes (cattr) table.

Usage

\[
\begin{align*}
\text{filter_col}_q(x, \ldots) \\
\text{filter_col}(x, \ldots) \\
\text{### S3 method for class 'zoocat'} \\
\text{filter_col}_q(x, \text{cond}, \ldots) \\
\text{### S3 method for class 'zoocat'} \\
\text{filter_col}(x, \text{cond}, \ldots) \\
\text{### S3 method for class 'zoomly'} \\
\text{filter_col}_q(x, \text{cond} = \text{NULL}, \text{mon.repro} = \text{NULL}, \ldots) \\
\text{### S3 method for class 'zoomly'} \\
\text{filter_col}(x, \text{cond} = \text{NULL}, \text{mon.repro} = \text{NULL}, \ldots)
\end{align*}
\]

Arguments

- **x**: the object.
- **\ldots**: other arguments.
- **cond**: logical predicates of conditions. Multiple conditions are combined with &. For filter_col, cond must be an expression, while for \text{filter_col}_q, cond must be a quoted expression.
- **mon.repro**: the reprocessing month vector, which is used for \text{reprocess_month}. See details.

Details

For filter_col, when the argument mon.repro is not NULL, \text{reprocess_month} will be called in the last step.

Value

a "zoocat" or "zoomly" object.
Examples

```r
x <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = c(rep('xxx', 3), 'yyy'))
zc <- zoocat(x, order.by = 1991:1995, colattr = colAttr)
filter_col(zc, month > 2)
filter_col(zc, month > 2)
filter_col(zc, month > 2 & name == 'yyy')

mat <- matrix(1:48, ncol = 12)
colAttr <- data.frame(month = rep(1:12))
zm <- zoomly(mat, order.by = 1991:1994, colattr = colAttr)
filter_col(zm, mon.repro = c(1:3))
filter_col(zm, mon.repro = c(-9:8))
filter_col(zm, cond = month %in% 1:3, mon.repro = c(-24:3))
```

gmon

A class for generalized month index

Description

In the "gmon" class, a integer number is used to indicate the month. The number from 1 to 12 is
used to indicate Jan to Dec of the reference year. The number from -11 to 0 is used to indicate Jan
to Dec of the previous year, and the number from 13 to 24 corresponds to the next year, and so on.

Usage

gmon(x)

Arguments

x a vector of integers.

Details

When print a "gmon" object, the suffix ".0" means the current year, ".1" means the next year and ".-1"
means the previous year, and so on. For example, Feb of the next year is printed as "Feb.1". The
methods scale_x_gmon and scale_y_gmon are provided to show "gmon" objects properly based on
ggplot.

Examples

```r
mvec <- gmon(-20:25)
print(mvec)
x <- 1:46
names(x) <- mvec
print(x)
```
gmon_trans  Tools for plotting zoocat objects with ggplot2

Description
Tools for plotting zoocat objects with ggplot2

Usage

```r
gmon_trans(n = 5)
scale_x_gmon(..., n = 5)
scale_y_gmon(..., n = 5)
```

Arguments

- `n` approximate number of axis ticks.
- `...` arguments for ggplot2::scale_x_continuous.

Examples

```r
library(ggplot2)
df <- data.frame(month = gmon(-5:5), value = 1:11)
ggplot(df, aes(month, value)) +
  geom_line() + geom_point() +
  scale_x_gmon()
```

index_detach  Detach index of a zoo object

Description

Detach the index of a zoo object to be separated values of year/month/day/hour/minute/second, and return a data frame containing these columns.

Usage

```r
index_detach(x, ...)
```

## S3 method for class 'zoo'
index_detach(x, nfield = 2, ...)
indname

Arguments

x  an object.

... further arguments.
nfield numeric. Number of fields of time to retain.

Value

a data frame.

Examples

```r
x.Date <- as.Date("2003-02-01") + c(1, 3, 7, 9, 14) - 1
x <- zoo(rnorm(5), x.Date)
index_detach(x, 3)

y.POSIXct <- ISOdatetime(2003, 02, c(1, 3, 7, 9, 14), 0, 0, 0)
y <- zoo(rnorm(5), y.POSIXct)
index_detach(y, 6)
```

Description

Get or set the name of the index variable

Usage

```r
indname(x)
indname(x) <- value

## S3 method for class 'zoocat'
indname(x)

## S3 replacement method for class 'zoocat'
indname(x) <- value
```

Arguments

x  the object.

value  the new value.
isvalid

Determine the validity of a zoocat Object

Description

Determine the validity of a "zoocat" object.

Usage

isvalid(x)

## S3 method for class 'zoocat'
isvalid(x)

Arguments

x a zoocat object.

Value

a logic variable.

Examples

x <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = c('xxx', 'yyy'))
zc <- zoocat(x, order.by = 1991:1995, colattr = colAttr)
indname(zc)
indname(zc) <- 'time'

isvalid
**melt**

*Melt a zoocat Object*

**Description**

Melt a zoocat to a data frame of the long table style, which is similar as in package reshape2.

**Usage**

```r
## S3 method for class 'zoocat'
melt(data, value.name = "value",
     index.name = indname(data), na.rm = FALSE, ...)
```

**Arguments**

- **data**: object to melt.
- **value.name**: name of the column used to store values. It is valid only when a data frame is returned.
- **index.name**: name of the column used to store the index of the zoocat object.
- **na.rm**: as `melt` in reshape2. Should NA values be removed from the data set?
- **...**: further arguments.

**Value**

a data frame.

**Examples**

```r
x <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = c('x', 'y', 'z', 'a'))
zc <- zoocat(x, order.by = 1991:1995, colattr = colAttr)
melt(zc)

zm <- as.zoomly(zc)
melt(zm)
```
melt_month

Translate zoomly objects to yearmon format

Description

Melt the month information of a zoomly object into the index, and return a "zoo" object with the index of "yearmon" class.

Usage

melt_month(x)

Arguments

x

a zoomly object.

Examples

```r
x <- matrix(1:20, nrow = 5)
zm <- zoomly(x, order.by = 1991:1995, 
  colattr = data.frame(month = c(2, 3, 5, 6))
melt_month(zm)

x <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = c(rep('xxx', 3), 'yyy'))
zc <- zoocat(x, order.by = 1991:1995, colattr = colAttr)
melt_month(as.zoomly(zc))
```

merge

Combine zoocat or zoomly Objects by Columns

Description

Combine "zoocat" or "zoomly" objects by columns.

Usage

```r
## S3 method for class 'zoocat'
merge(..., all = TRUE, fill = NA, suffixes = NULL, 
  check.names = FALSE, retclass = "zoocat", drop = TRUE)

cbind.zoocat(...)
Arguments

... zoocat or zoomly objects.
all, fill, suffixes, check.names, retclass, drop
See details.

Details

merge.zoocat and merge.zoomly are the extensions of merge.zoo. For merge.zoocat, when combining cattr, some NA will be filled in if it is necessary.
The arguments all and fill are used same as in merge.zoo. The arguments suffixes, check.names, retclass and drop are not used.

Value

merge.zoocat will return a "zoocat" object. merge.zoomly will return a "zoomly" object.

Examples

x1 <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = 'xxx')
zc1 <- zoocat(x1, order.by = 1991:1995, colattr = colAttr)
x2 <- x1 + 100
colAttr2 <- data.frame(modified = TRUE, month = c(4, 6, 7, 9))
zc2 <- zoocat(x2, order.by = 1991:1995, colattr = colAttr2)
merge(zc1, zc2)
merge(zc1, lag(zc2), fill = -999)
merge(zc1, lag(zc2), all = FALSE)
cbind(zc1, zc2)

---

normalize Normalize data

Description

Normalize each column of the object using different methods. See details.

Usage

normalize(x, ...)

## Default S3 method:
normalize(x, method = "sd1", base.period = 1:nrow(x), ...)

## S3 method for class 'zoo'
normalize(x, method = "sd1", base.period = index(x), ...)
Arguments

- **x**: a vector, matrix, data frame or zoo object.
- **...**: additional arguments to be passed to or from methods.
- **method**: a character string indicating which method to use. Must be "sd1" (default), "anomaly" or "perc". See details.
- **base.period**: a vector indicating the index or range of the base period. If NULL, base period is the all index range. For matrix, base.period means the row numbers.

Details

Three methods for normalization can be used:

1. "anomaly": Each column is normalized by \( x - \mu \), where \( \mu \) is the mean value based on the base.period.
2. "perc": Each column is normalized by \( 100(x - \mu)/\mu \). This is often called anomaly percentage.
3. "sd1": Each column is normalized by \( (x - \mu)/\sigma \), where \( \sigma \) is the standard deviation based on the base.period. The standard deviations of the results will be 1 if the base.period is set to be the whole time range.

Examples

```r
x <- matrix(1:20, nrow = 10)
colnames(x) <- c('a', 'b')
rownames(x) <- 1:10
normalize(x, method = 'anomaly')
normalize(x, method = 'perc')
normalize(x, method = 'sd1')

z <- zoo(x, order.by = 1991:2010)
normalize(z)

x <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = c(rep('xxx', 3), 'yyy'))
zc <- zoocat(x, order.by = 1991:1995, colattr = colAttr)
normalize(zc)
```

---

**order_col**

*Order a zoocat object by column*

Description

Order a "zoocat" object by column based on the values in the cattr table.

Usage

```r
order_col(x)
```
prcomp.zoocat

Arguments

x the object.

Examples

```r
x <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = c(rep('xxx', 3), 'yyy'))
zm <- zoomly(x, order.by = 1991:1995, colattr = colAttr)
order_col(zm)
zm <- zm[, c(4, 2, 3, 1)]
order_col(zm)
```

Description

Principal components analysis of a "zoocat" object. This function is a wrap of the `stats::prcomp`. The rotation returned by `stats::prcomp` is binded with the cattr table to be a data frame, and the x returned by `stats::prcomp` is binded with the index to be a "zoo" object.

Usage

```r
## S3 method for class 'zoocat'
prcomp(x, ...)
```

Arguments

x the zoocat object.

... other argument.

Value

a list with following elements: prcomp.obj (the object returned by `stats::prcomp`), rotation (the data frame containing information of the variable loadings), z (the zoo object containing the rotated data).

Examples

```r
x <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = c(rep('xxx', 3), 'yyy'))
zc <- zoocat(x, order.by = 1991:1995, colattr = colAttr)
pca <- prcomp(zc)
print(pca$rotation)
print(pca$z)
```
rbind.zoocat  
_Merge two or more zoocat objects by rows_

**Description**

Merge two or more zoocat objects by rows. Note that all the cattr tables must be the same.

**Usage**

```
rbind.zoocat(..., deparse.levels = 1)
```

**Arguments**

- `...` zoocat objects.
- `deparse.levels` Not used.

**Examples**

```
x <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = c(rep('xxx', 3), 'yyy'))
zc <- zoocat(x, order.by = 1991:1995, colattr = colAttr)
zcz <- zc
index(zcz) <- index(zc) + 6
rbind(zc, zcz)
```

---

rela_year  
_Get the relative years for a gmon object_

**Description**

Get the relative years for a "gmon" object.

**Usage**

```
rela_year(x)
```

**Arguments**

- `x` a gmon object.

**Value**

a vector.
**reprocess_month**

**Examples**

```r
t <- -12 : 3
num_mon <- gmon(t)
gm <- gmon(num_mon)
ry <- rela_year(gm)
df <- data.frame(num_mon, ry, as.character(gm))
print(df)
```

---

**reprocess_month**  
*Reprocess month of zoomly objects*

---

**Description**

Reprocess month of "zoomly" objects, make the objects contain the data corresponding to months of previous years and following years.

**Usage**

```r
reprocess_month(x, mon.repro)
```

**Arguments**

- `x`: a zoomly object.
- `mon.repro`: new setting month vector. Can be integers larger than 12 or less than 1.

**Details**

For example, if there is a data value corresponding to year of 1990 and month of Jan, the argument `month` for `reprocess_month` can be set to be 13, and we get data of "Jan.1" (means Jan of the following year, see `gmon`), and the year of that data value will be 1991.

**Value**

a "zoomly" object.

**Examples**

```r
mat <- matrix(1:48, ncol = 12)
c_table <- data.frame(month = rep(1:12))
zm <- zoomly(mat, order.by = 1991:1994, colattr = c_table)
reprocess_month(zm, mon.repro = -11:2)
reprocess_month(zm, mon.repro = -24:3)
```
**reset_index_var**

*Reset index variable of a zoocat object*

**Description**

Reset the index variable of a "zoocat" object to be one of the fields in the `cattr` table.

**Usage**

```r
reset_index_var(x, ...)
```

```r
## S3 method for class 'zoocat'
reset_index_var(x, index.var, ...)
```

**Arguments**

- `x`: the object.
- `...`: other arguments.
- `index.var`: the name of the variable to be set as the index. Must be one of the `cattr` field.

**Value**

a "zoocat" object.

**Examples**

```r
x <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = c(rep('xxx', 3), 'yyy'))
zc <- zoocat(x, order.by = 1991:1995, colattr = colAttr)
print(zc)
reset_index_var(zc, 'month')
reset_index_var(zc, 'name')
```

**rollcor**

*Calculate the rolling window correlations*

**Description**

Calculate the rolling window correlations.
Usage

rollcor(x, y, ...)

## Default S3 method:
rollcor(x, y, width, show = TRUE, ...)

## S3 method for class 'zoo'
rollcor(x, y, width, show = TRUE, ...)

Arguments

x, y
Two vectors or two zoo objects. For zoo objects, if their time ranges is different, intersection will be used.

... Other arguments for function cor.

width The width of the sliding window, which must be odd number.

show If TRUE, the result will be plotted.

Value

rollcor.default return a vector, and rollcor.zoo return a "zoo" object.

Examples

x <- 1:100
y <- 2 * x + rnorm(100, 0, 10)
rollcor(x, y, width = 21)

xz <- zoo(x)
yz <- zoo(y)
rollcor(xz, yz, width = 21)

rollcor(xz, yz, width = 21, show = FALSE)

x <- 1:100
y <- 2 * x + rnorm(100, 0, 10)
x <- zoo(x, order.by = 10:109)
y <- zoo(y, order.by = -3:96)
rollcor(x, y, width = 21, method = 'kendall')

---

**tropsST**

**SST Data of Tropical Pacific**

Description

SST Data of Tropical Pacific
Examples

```r
data(tropSST)
head(tropSST)
```

true_month  

*Get the true month of a gmon object.*

Description

Get the true month of a gmon object.

Usage

```r
true_month(x)
```

Arguments

- `x` a gmon object.

Value

a vector.

Examples

```r
num.mon <- -25 : 15
gm <- gmon(num.mon)
tm <- true_month(gm)
cbind(gm, tm)
```

zoocat  

*zoocat class*

Description

A class designed for "zoo" class with a column attribute (`cattr`) table.

Usage

```r
zoocat(x = NULL, order.by = index(x), colattr = NULL, index.name = "index", ...)
```
Arguments

- **x**: a matrix. If x is a data frame, it will be converted to a matrix.
- **order.by**: an index vector with unique entries by which the observations in x are ordered.
- **colattr**: the column attributes. Must be a data frame with column names.
- **index.name**: the name of the index variable.
- **...**: other arguments for `zoo`.

Details

"zoocat" is a S3 class based on the "zoo" class, which means "zoo" with column(C) attributes(AT). It is known that a "zoo" object is a vector or matrix with a index attribute. In a "zoocat" object, another attribute named "catattr" (a data frame with column names) is added to keeps the underlying attribute information of each column. So there are two attributes in a "zoocat" object: "index" and "catattr". The number of rows of the "catattr" table must be the same with ncol(x), where x is the core data. Each row in the "catattr" table is correspondent to each column of the core data matrix.

In summary, "zoocat" class can manage the underlying information of each column more conveniently than only using column names. It can be used to store time series data each column of which is corresponding to several underlying variables.

Two methods can be used to build a "zoocat" object. The first is to use function `zoocat`, the "catattr" table is specified by argument `colattr`. The second method is to use `cast2zoocat` to get the object from a data frame.

When printing "zoocat" objects, column names will be added automatically, but it should be noted that the column names do not exist. As "zoo" object, `coredata` can be used to extract the core data matrix from the object.

Many methods have been defined for the "zoocat" object. `filter_col` can be used to get columns which satisfy some conditions. `melt` can be used to melt the object, like what `melt` in the `reshape2` do. `normalize` can be used to normalized data using several methods. `apply_col` can be used to apply a function to each column and bind the results with the "catattr" table. `apply_core` can apply a function to the whole core data matrix, and bind the results with the "index" or "catattr" table.

It should be noted that all methods for "zoo" objects can be used for "zoocat" objects, such as `na.omit`, `na.approx`, `na.fill`, `na.trim`, `lag`. See the help page of `zoo`.

Value

A "zoocat" object.

Examples

```r
x <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = c(rep('xxx', 3), 'yyy'))
zc <- zoocat(x, order.by = 1991:1995, colattr = colAttr)
unclass(zc)
zc[1, 3]
zc[2, ]
zc[, '2_xxx']
coredata(zc)
```
as.matrix(zc)

x <- matrix(1:20, nrow = 5)
colAttr <- data.frame(month = c(2, 3, 5, 6), name = c(rep('xxx', 3), 'yyy'))
zc <- zoocat(x, order.by = 1991:1995, colattr = colAttr, frequency = 1)

---

zoomly | zoomly Class

---

**Description**

A class designed for monthly data. "zoomly" class inherits "zoocat" class, and there is one field "month" in the column attribute (colattr) table.

**Usage**

zoomly(x = NULL, order.by, colattr)

**Arguments**

- **x**: a matrix or a vector. For zoomly, if `x` is a matrix, each row will be treated as a year. If `x` is a vector, it will be treated as a matrix with only one column.
- **order.by**: a numeric vector representing years.
- **colattr**: a column attributes table contain a column "month".

**Value**

zoomly returns a zoomly object.

**Examples**

```r
x <- matrix(1:20, nrow = 5)
zm <- zoomly(x, order.by = 1991:1995,
             colattr = data.frame(month = c(2, 3, 5, 6)))
```
## Index

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>aggregate_col</td>
<td>2</td>
</tr>
<tr>
<td>apply_col</td>
<td>3, 27</td>
</tr>
<tr>
<td>apply_core</td>
<td>4, 27</td>
</tr>
<tr>
<td>as.gmon</td>
<td>5</td>
</tr>
<tr>
<td>as.zoo</td>
<td>5</td>
</tr>
<tr>
<td>as.zoocat</td>
<td>6</td>
</tr>
<tr>
<td>as.zoomly</td>
<td>7</td>
</tr>
<tr>
<td>castRzoocat</td>
<td>8, 27</td>
</tr>
<tr>
<td>castRzoomly</td>
<td>9</td>
</tr>
<tr>
<td>cast_month</td>
<td>9</td>
</tr>
<tr>
<td>cattr</td>
<td>10</td>
</tr>
<tr>
<td>cattr&lt;-(cattr)</td>
<td>10</td>
</tr>
<tr>
<td>cbindRzoocat(merge)</td>
<td>18</td>
</tr>
<tr>
<td>cor</td>
<td>11</td>
</tr>
<tr>
<td>coredata</td>
<td>27</td>
</tr>
<tr>
<td>dcast</td>
<td>8</td>
</tr>
<tr>
<td>filter_col</td>
<td>12, 27</td>
</tr>
<tr>
<td>filter_col_q(filter_col)</td>
<td>12</td>
</tr>
<tr>
<td>gmon</td>
<td>13, 23</td>
</tr>
<tr>
<td>gmon_trans</td>
<td>14</td>
</tr>
<tr>
<td>index_detach</td>
<td>14</td>
</tr>
<tr>
<td>indname</td>
<td>15</td>
</tr>
<tr>
<td>indname&lt;-(indname)</td>
<td>15</td>
</tr>
<tr>
<td>isvalid</td>
<td>16</td>
</tr>
<tr>
<td>melt</td>
<td>17, 27</td>
</tr>
<tr>
<td>melt_month</td>
<td>18</td>
</tr>
<tr>
<td>merge</td>
<td>18</td>
</tr>
<tr>
<td>na.approx</td>
<td>27</td>
</tr>
<tr>
<td>na.fill</td>
<td>27</td>
</tr>
<tr>
<td>na.trim</td>
<td>27</td>
</tr>
<tr>
<td>normalize</td>
<td>19, 27</td>
</tr>
<tr>
<td>order_col</td>
<td>20</td>
</tr>
<tr>
<td>prcomp.zoocat</td>
<td>21</td>
</tr>
<tr>
<td>rbind.zoocat</td>
<td>22</td>
</tr>
<tr>
<td>rela_year</td>
<td>22</td>
</tr>
<tr>
<td>reprocess_month</td>
<td>12, 23</td>
</tr>
<tr>
<td>reset_index_var</td>
<td>24</td>
</tr>
<tr>
<td>rollcor</td>
<td>24</td>
</tr>
<tr>
<td>scale_x_gmon(gmon_trans)</td>
<td>14</td>
</tr>
<tr>
<td>scale_y_gmon(gmon_trans)</td>
<td>14</td>
</tr>
<tr>
<td>tropSST</td>
<td>25</td>
</tr>
<tr>
<td>true_month</td>
<td>26</td>
</tr>
<tr>
<td>zoo</td>
<td>27</td>
</tr>
<tr>
<td>zoocat</td>
<td>26</td>
</tr>
<tr>
<td>zoomly</td>
<td>28</td>
</tr>
</tbody>
</table>