

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

cast_month	9
cattr	10
cor	11
filter_col	12
gmon	13
gmon_trans	14
index_detach	14
indname	15
isvalid	16
melt	17
melt_month	18
merge	18
normalize	19
order_col	20
prcomp.zoocat	21
rbind.zoocat	22
rela_year	22
reprocess_month	23
reset_index_var	24
rollcor	24
tropSST	25
true_month	26
zoocat	26
zoomly	28

Index **29**

aggregate_col	<i>Aggregate zoocat objects by column attributes</i>
---------------	--

Description

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

Usage

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

Arguments

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

Value

a "zoocat" object.

Examples

```
# 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)
```

 apply_col

Apply functions over each column

Description

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

Usage

```
apply_col(x, ...)

## S3 method for class 'zoocat'
apply_col(x, FUN, col.as = "vector", ...)
```

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

```
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

```
apply_core(x, ...)

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

Arguments

<code>x</code>	the object.
<code>...</code>	other arguments for FUN.
<code>FUN</code>	the function to apply. The FUN must return a matrix or a vector.
<code>bind</code>	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 <code>cattr</code> or <code>index</code> . If FUN return a vector, set <code>bind</code> to be a scalar. If FUN return a matrix, set <code>bind</code> to be a vector of length 2. See details.

Value

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

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)

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*2}, bind = c('index', NA))
apply_core(zc, FUN = function (x) {x[3:4, ]}, bind = c(NA, 'cattr'))
apply_core(zc, FUN = function (x)
  {r <- x[3:4, ]
  rownames(r) <- c('a', 'b')
  return(r)},
  bind = c(NA, 'cattr'))

vec <- as.vector(zc[, 1])
```

```
apply_core(zc, FUN = function (x) {cor(x, vec)}, bind = 'catr')
```

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 zooreg object if the input x is inherited from zooreg.

Value

A zoo 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)
```

as.zoocat	<i>Coercing objects to Class zoocat</i>
-----------	---

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 variable name.
index.name	the name of the index variable.

Value

a "zoocat" object.

Examples

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

as.zoomly	<i>Coercion objects to class zoomly</i>
-----------	---

Description

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

Usage

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

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

Arguments

x	an object.
...	further arguments passed to methods.

Value

A "zoomly" object.

Examples

```
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(rep('xx', 3), 'yy'))
zc <- zoocat(x, order.by = 1991 : 1995, colattr = colAttr)
as.zoomly(zc)
```

cast2zoocat	<i>Cast a data frame to a zoocat object</i>
-------------	---

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

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

Arguments

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

Value

a "zoocat" object.

Examples

```
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	<i>Cast a data frame to a zoomly object</i>
-------------	---

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

```
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	<i>Cast month of a zoomly object</i>
------------	--------------------------------------

Description

Cast month of a zoomly object

Usage

```
cast_month(x)
```

Arguments

x	a zoocat object.
---	------------------

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)
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

```
cattr(x) <- value

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

## S3 replacement method for class 'zoo'
cattr(x) <- value

cattr(x)

## S3 method for class 'zoocat'
cattr(x)
```

Arguments

x	A zoocat or zoo object.
value	The new value.

Value

"cattr(x)" return a data frame.

Examples

```
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'
```

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

cor

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 **catr** 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

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

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	<i>Return columns with matching conditions</i>
------------	--

Description

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

Usage

```
filter_col_q(x, ...)

filter_col(x, ...)

## S3 method for class 'zoocat'
filter_col_q(x, cond, ...)

## S3 method for class 'zoocat'
filter_col(x, cond, ...)

## S3 method for class 'zoomly'
filter_col_q(x, cond = NULL, mon.repro = NULL, ...)

## S3 method for class 'zoomly'
filter_col(x, cond = NULL, mon.repro = NULL, ...)
```

Arguments

x	the object.
...	other arguments.
cond	logical predicates of conditions. Multiple conditions are combined with &. For <code>filter_col</code> , cond must be an expression, while for <code>filter_col_q</code> , cond must be a quoted expression.
mon.repro	the reprocessing month vector, which is used for reprocess_month . See details.

Details

For `filter_col`, when the argument `mon.repro` is not `NULL`, `reprocess_month` will be called in the last step.

Value

a "zoocat" or "zoomly" object.

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)
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 = 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

```
mvec <- gmon(-20:25)
print(mvec)
x <- 1 : 46
names(x) <- mvec
print(x)
```

 gmon_trans

Tools for plottin zoocat objects with ggplot2

Description

Tools for plottin zoocat objects with ggplot2

Usage

```
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

```
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 seperated values of year/month/day/hour/minute/second, and return a data frame containing these columns.

Usage

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

```
## S3 method for class 'zoo'
```

```
index_detach(x, nfield = 2, ...)
```

Arguments

x a object.
 ... further arguments.
 nfield numeric. Number of fields of time to retain.

Value

a data frame.

Examples

```
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)
```

indname

Get or set the name of the index variable

Description

Get or set the name of the index variable

Usage

```
indname(x)

indname(x) <- value

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

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

Arguments

x the object.
 value the new value.

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)
indname(zc)
indname(zc) <- 'time'
```

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 = 'sst')
zc <- zoocat(x, order.by = 1991 : 1995, colattr = colAttr)
isvalid(zc)
```

melt	<i>Melt a zoocat Object</i>
------	-----------------------------

Description

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

Usage

```
## 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

```
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	<i>Translate zoomly objects to yearmon format</i>
------------	---

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

```
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	<i>Combine zoocat or zoomly Objects by Columns</i>
-------	--

Description

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

Usage

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

cbind.zoocat(...)
```

Arguments

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

Details

merge.zocat and merge.zoomly are the extensions of merge.zoo. For merge.zocat, 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.zocat will return a "zocat" 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 <- zocat(x1, order.by = 1991 : 1995, colattr = colAttr)
x2 <- x1 + 100
colAttr2 <- data.frame(modified = TRUE, month = c(4, 6, 7, 9))
zc2 <- zocat(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

<code>x</code>	a vector, matrix, data frame or zoo object.
<code>...</code>	additional arguments to be passed to or from methods.
<code>method</code>	a character string indicating which method to use. Must be 'sd1' (default), 'anomaly' or 'perc'. See details.
<code>base.period</code>	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 μ 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 σ 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

```
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

```
order_col(x)
```

Arguments

x the object.

Examples

```
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)
```

prcomp.zoocat

Principal components analysis of a zoocat object

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 `colattr` table to be a data frame, and the `x` returned by `stats::prcomp` is binded with the `index` to be a "zoo" object.

Usage

```
## 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

```
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	<i>Merge two or more zoocat objects by rows</i>
--------------	---

Description

Merge two or more zoocat objects by rows. Note that all the cat tr 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)
zc2 <- zc
index(zc2) <- index(zc) + 6
rbind(zc, zc2)
```

rela_year	<i>Get the relative years for a gmon object</i>
-----------	---

Description

Get the relative years for a "gmon" object.

Usage

```
rela_year(x)
```

Arguments

x a gmon object.

Value

a vector.

Examples

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

reprocess_month	<i>Reprocess month of zoomly objects</i>
-----------------	--

Description

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

Usage

```
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

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

reset_index_var	<i>Reset index variable of a zoocat object</i>
-----------------	--

Description

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

Usage

```
reset_index_var(x, ...)

## 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

```
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	<i>Calculate the rolling window correlations</i>
---------	--

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')
```

Description

SST Data of Tropical Pacific

Examples

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

true_month	<i>Get the true month of a gmon object.</i>
------------	---

Description

Get the true month of a gmon object.

Usage

```
true_month(x)
```

Arguments

x a gmon object.

Value

a vector.

Examples

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

zoocat	<i>zoocat class</i>
--------	---------------------

Description

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

Usage

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

Arguments

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

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 "**cattr**" (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 "cattr". The number of rows of the "cattr" table must be the same with `ncol(x)`, where `x` is the core data. Each row in the "cattr" 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 "cattr" 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 "cattr" table. `apply_core` can apply a function to the whole core data matrix, and bind the results with the "index" or "cattr" 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

```
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 (**cattr**) 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

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

Index

aggregate_col, 2
apply_col, 3, 27
apply_core, 4, 27
as.gmon, 5
as.zoo, 5
as.zoocat, 6
as.zoomly, 7

cast2zoocat, 8, 27
cast2zoomly, 9
cast_month, 9
cattr, 10
cattr<- (cattr), 10
cbind.zoocat (merge), 18
cor, 11
coredata, 27

dcast, 8

filter_col, 12, 27
filter_col_q (filter_col), 12

gmon, 13, 23
gmon_trans, 14

index_detach, 14
indname, 15
indname<- (indname), 15
isvalid, 16

melt, 17, 27
melt_month, 18
merge, 18

na.approx, 27
na.fill, 27
na.trim, 27
normalize, 19, 27

order_col, 20

prcomp.zoocat, 21

rbind.zoocat, 22
rela_year, 22
reprocess_month, 12, 23
reset_index_var, 24
rollcor, 24

scale_x_gmon (gmon_trans), 14
scale_y_gmon (gmon_trans), 14

tropSST, 25
true_month, 26

zoo, 27
zoocat, 26
zoomly, 28