

Package ‘its’

January 2, 2012

Version 1.1.8

Date 2009-09-06

Title Irregular Time Series

Author Portfolio & Risk Advisory Group, Commerzbank Securities

Maintainer Whit Armstrong <armstrong.whit@gmail.com>

Depends R (>= 2.0.0), methods, stats, Hmisc

Description The its package contains an S4 class for handling irregular time series

LazyLoad yes

License GPL-2

Repository CRAN

Date/Publication 2009-09-06 19:52:40

R topics documented:

core	2
dates	2
its-class	3
itsConvert	4
itsDef	5
itsFile	7
itsFin	9
itsInterp	10
itsJoin	11
itsLags	12
itsSubset	13
itsTimes	14
priceIts	15

Index	17
--------------	-----------

core	<i>Extract core from object or assign core to object</i>
------	--

Description

Extract core from objects. Particularly time series objects.

Usage

```
core(x)
core(x) <- value
```

Arguments

x	an object which contains core
value	new core to be assigned to object
...	further arguments to function

Value

a vector of core

Author(s)

Whit Armstrong

dates	<i>Extract dates from object or assign dates to object.</i>
-------	---

Description

Extract dates from objects. Particularly time series objects.

Usage

```
dates(x, ... )
dates(x) <- value
```

Arguments

x	an object which contains dates
value	new dates to be assigned to object
...	further arguments to function

Value

a vector of dates

Author(s)

Whit Armstrong

 its-class

 Class "its" Irregularly Spaced Time-Series

Description

S4 class implementation of irregularly spaced time series. The class implements Arith methods using date matching, union, intersection, and other time series methods.

Objects from the Class

Objects can be created by calls of the form `new("its", ...)`. ~~ describe objects here ~~

Slots

`.Data`: Object of class "matrix" the data portion of the time series

`dates`: Object of class "POSIXt" the date vector portion of the time series

Extends

Class "`matrix`", from data part. Class "`structure`", by class "matrix", distance 2. Class "`array`", by class "matrix", distance 2. Class "`vector`", by class "matrix", distance 3, with explicit coerce. Class "`vector`", by class "matrix", distance 4, with explicit coerce.

Methods

Arith signature(`e1 = "its"`, `e2 = "its"`): method for Arith of its and its

Arith signature(`e1 = "its"`, `e2 = "numeric"`): method for arith of its and numeric

Arith signature(`e1 = "numeric"`, `e2 = "its"`): method for arith of numeric and its

[signature(`x = "its"`): subset operator for its

[<- signature(`x = "its"`, `i = "ANY"`, `j = "ANY"`, `value = "its"`): subset assignment operator for its

coerce signature(`from = "its"`, `to = "data.frame"`): coerce its to data.frame

core signature(`x = "its"`): extract numeric portion of its

core<- signature(`x = "its"`, `value = "matrix"`): assignment to numeric portion of its

cumsum signature(`x = "its"`): cumulative sum of its series

dates signature(`x = "its"`): extract dates vector from its object

dates<- signature(`x = "its"`, `value = "POSIXt"`): assign dates into its object

diff signature(x = "its"): difference function for its
end signature(x = "its"): ...
intersect signature(x = "its", y = "its"): take date intersection of its object
intersect signature(x = "its", y = "NULL"): ...
intersect signature(x = "NULL", y = "its"): ...
names signature(x = "its"): extract column names from its object
names<- signature(x = "its", value = "character"): assign names into its object
plot signature(x = "its", y = "missing"): plot an its object
print signature(x = "its"): print an its object
start signature(x = "its"): ...
summary signature(object = "its"): summary of an its object
union signature(x = "its", y = "its"): combine its objects
union signature(x = "its", y = "NULL"): ...
union signature(x = "NULL", y = "its"): ...

Note

This package was written by Giles Heywood.

Author(s)

Whit Armstrong

References

<http://code.google.com/p/rits>

Examples

```
showClass("its")
```

 itsConvert

Conversion for its objects

Description

Conversions to basic types for "its".

Usage

```
as.list.its(x,...)
as.data.frame.its(x, row.names = NULL, optional = FALSE,...)
```

Arguments

x	an object of class "its"
...	further arguments
row.names	character vector giving the row names for the data frame.
optional	logical. If 'TRUE', setting row names and converting column names (to syntactic names) is optional.

Details

Conversion of basic types

Value

For `as.list.its` a list for `as.data.frame.its` a data frame

Author(s)

Gabor Grothendieck

itsDef

Irregularly Spaced Time-Series

Description

The function `its` is used to create irregular time-series objects from user-supplied data and time-stamps. `newIts` is used to create semi-regular time-series objects from user-supplied data, and rule for generating time-stamps. `as.its` coerces an object to an irregularly spaced time-series. `is.its` tests whether an object is an irregularly spaced time series.

Usage

```
its(x,
    dates=as.POSIXct(x=strptime(dimnames(x)[[1]], format=its.format()),
    names=dimnames(x)[[2]],
    format=its.format(),...)
```

```
newIts(x=NA, start=format(Sys.Date(), format=its.format()),
end, ncol=1, by="DSTday", extract=FALSE, format=its.format(), tz="",...)
```

```
as.its(x,...)
```

```
is.its(object)
```

```
its.format(formatDefault=NULL)
```

Arguments

dates	a vector of class "POSIXct" representing the time-stamps of the irregular time-series object. The elements of the numeric vector are construed as the number of seconds since the beginning of 1970, see POSIXct .
start, end	POSIXct or character representation of the start or end time-stamp, if character, then the format is as specified by the argument format
ncol	number of columns of synthetic sequence of dates
by	time increment for synthetic sequence of dates, see seq.POSIXt
extract	logical flag: if TRUE, a subset of the synthetic sequence of dates is extracted, see extractIts
x	a numeric matrix representing the values of the irregular time-series object. In the case of coercion in <code>as.its</code> , the first column is taken to be the time-stamps, in seconds since the beginning of 1970, see POSIXct .
object	an R object convertible to its
names	a vector of mode character
format	a formatting string, see format.POSIXct , defaults to <code>its.format()</code>
formatDefault	a formatting string, see format.POSIXct , defaults to "%Y-%m-%d" if formatDefault is not specified.
tz	time zone of dates of its object
...	further arguments passed to or from other methods: for <code>its</code> passed to format.POSIXct .; for <code>as.its</code> passed to its .; for <code>newIts</code> passed to extractIts .

Details

The function `its` is used to create irregular time-series objects, which have (S4) class "its". An object of class "its" is a matrix with rows indexed by a time-stamp of class "POSIXct". Unlike objects of class "ts", it can be used to represent irregularly spaced time-series. The object consists of a matrix, with a single slot, the named POSIX vector named "dates". An object of class "its" inherits matrix arithmetic methods. The matrix has dimnames: `dimnames[[1]]` is populated with a text representation of "dates", using a format which is defined by the function `its.format`. These dates are not used in computations - all computations use the POSIX representation. The dates are required to be in ascending order.

When matrix multiplication is applied to an "its", the result is of class matrix. It is possible to restore the "its" class (see examples) - `its()` is in this sense idempotent i.e. `its(mat)==its(its(mat))`. Note however that the dates will be taken from `dimnames[[1]]`, so the accuracy of this operation depends on the format of the dates.

`newIts` is a utility for creating a new "its" using a series of 'semi-regular' time-stamps, such as weekday, weekly, monthend etc. Conceptually the date sequence generation has two parts. The first part is the generation of a sequence using [seq.POSIXt](#) - the arguments from, to, and by are passed to this function. The second part (which is optional, and applies only if `extract=TRUE`) is an extraction, performed by `extractIts`. See [extractIts](#) for details of the arguments, which are passed via `'...'`.

The function `its.format` assigns a private variable and returns its value. The value of this default format persists in the session until reset. The purpose of the function is one of convenience: to

access and/or assign the default text format for dates in the "its" package, and hence reduce the need to define the format repeatedly in a session.

Value

For `its`, `newIts` and `as.its`, an object of class "its", inheriting from `matrix`, with a single slot named 'dates', which is a vector of class `POSIXct`

For `is.its`, a logical representing the result of a test for class membership

For `its.format`, a text representation of dates formatting to be used in the "its" package, see [format.POSIXct](#)

Author(s)

Giles Heywood

See Also

[ts](#), [POSIXct](#), [itsFile](#), [itsLags](#), [itsJoin](#), [itsTimes](#), [itsSubset](#), [itsFin](#), [itsInterp](#)

Examples

```
its.format("%Y-%m-%d") #defines text format of dates read from dimnames
mat <- structure(1:6,dim=c(2,3),dimnames=list(c("2003-01-01","2003-01-04"),letters[1:3]))
its(mat)
its.format("%Y%m%d") #defines text format of dates written to dimnames
times <- as.POSIXct(strptime(c("1999-12-31 01:00:00","2000-01-01 02:00:00"),format="%Y-%m-%d %X"))
its(mat,times)
its.format("%Y-%m-%d %X")
its(mat,times)
is.its(its(mat,times))
its.format("%Y%m%d %X") #defines text format of dates written to dimnames
as.its(mat)
its.format("%a %d %b %Y")
newIts(start="2003-09-30",end="2005-05-05",format="%Y-%m-%d",period="month",find="last",extract=TRUE,weekday=TRUE)
newIts(start=ISOdate(2003,12,24,0),end=ISOdate(2004,1,10),extract=TRUE,weekday=TRUE)
its.format("%Y%m%d")
as(newIts(),"data.frame")
```

itsFile

File Operations for Irregular Time-Series Objects

Description

File read and write operations for objects of class "its".

Usage

```
readcsvIts(filename,informat=its.format(),outformat=its.format(),tz="",usetz = FALSE,header=TRUE,...)
writescvIts(x,filename,format=its.format(),tz="",usetz = FALSE,col.names=NA,
sep=",",split=FALSE,...)
```

Arguments

filename	filename
x	an object of class "its"
format, informat, outformat	formatting related arguments, see format.POSIXct .
tz	what timezone the its is in
usetz	whether to include the tzone information in the saved file
header	see read.csv
col.names, sep	see write.table
split	when columns exceed 255 in number, flags for splitting into numbered subfiles
...	further arguments passed to or from other methods: for readcsvIts passed to read.csv ; for writecsvIts passed to write.table

Details

readcsvIts reads from a .csv file to a matrix. The first column is assumed to contain dates in text format specified by informat, which can optionally be reformatted into the text format outformat. Both of these formats default to the format specified by [its.format](#). To convert the matrix to an its, use [its](#) (see example)

writecsvIts write an irregular time-series object to a text file.

Value

For readcsvIts a matrix

Author(s)

Giles Heywood

See Also

[ts](#), [POSIXct](#), [itsFile](#), [itsLags](#) [itsJoin](#) [itsTimes](#) [itsSubset](#) [itsFin](#) [itsInterp](#)

Examples

```
## Not run:
b <- newIts(1:30,ncol=3)
fname <- tempfile()
# To write an irregular time-series object to a file one might use
writecsvIts(b,filename=fname)
# To read an irregular time-series object from a file one might use
its(readcsvIts(filename=fname))
unlink(fname)

## End(Not run)
```

Description

Financial functions for objects of class "its".

Usage

```
accrueIts(x, daysperyear=365)
```

Arguments

x	an object of class "its"
daysperyear	integer, representing days per year for accrual

Details

accrueIts Accrues an annual rate stored in x, expressed as a decimal (not a percentage), based on the difference of the time-stamps. The accrued value is not cumulated, it is a per-period return.

Value

An object of class "its"

Author(s)

Giles Heywood

See Also

[ts](#), [POSIXct](#), [itsFile](#), [itsLags](#) [itsJoin](#) [itsTimes](#) [itsSubset](#) [itsFin](#) [itsInterp](#)

Examples

```
a <- matrix(c(seq(by=24*60*60, length=20), 1:20, 41:60), nrow=20, ncol=3)
b <- as.its(a)
accrueIts(b[which(weekdayIts(b)), ], /100, daysperyear=365)
```

`itsInterp`*Interpolation Functions for Irregular Time-Series Objects*

Description

Interpolation functions for objects of class "its".

Usage

```
locf(x)
```

Arguments

`x` an object of class "its" or NULL

Details

`locf` 'Last Observation Carry Forward'. NAs are replaced by the last preceding valid observation within the series.

Value

An object of class "its".

Author(s)

Giles Heywood

See Also

[ts](#), [POSIXct](#), [itsFile](#), [itsLags](#), [itsJoin](#), [itsTimes](#), [itsSubset](#), [itsFin](#),

Examples

```
x <- newIts(11:40,ncol=3)
x[1:2,1] <- NA
x[3:4,2] <- NA
x[9:10,3] <- NA
print(x)
print(locf(x))
```

`itsJoin`*Join Functions for Irregular Time-Series Objects*

Description

Join functions for objects of class "its".

Usage

```
alignedIts(obj1, obj2, print=FALSE)
appendIts(obj1, obj2, but=TRUE, matchnames=TRUE)
collapseIts(x)
```

Arguments

<code>x</code>	an object of class "its" or NULL
<code>obj1, obj2</code>	object of class "its"
<code>print</code>	logical flag to display summary information
<code>but</code>	logical flag controls whether overlap is disallowed
<code>matchnames</code>	logical flag controls whether names must match

Details

`alignedIts` selects the rows from two inputs which have identical time-stamps.

`appendIts` appends one object to the other, removing overlapping data from the later object, optionally checking that the column names match. Overlapping data must match.

`collapseIts` checks for columns with the same names, using `dimnames(x)[[2]]`. columns with the same names are required to have the same numeric values in each row, but are allowed NAs. The numeric data is combined, and the resulting object has unique column names - this will in general result in a reduction in the number of columns.

Value

For `appendIts` an object of class "its".

For `alignedIts`, a list of two objects of class "its"

Author(s)

Giles Heywood

See Also

[ts](#), [POSIXct](#), [itsFile](#), [itsLags](#), [itsJoin](#), [itsTimes](#), [itsSubset](#), [itsFin](#), [itsInterp](#)

Examples

```

its.format("%Y-%m-%d")
b <- newIts(1:30, ncol=3, start="2003-01-01", end="2003-01-10")
union(b, diff(b))
intersect(b, diff(b))
alignedIts(b, diff(b))
b1 <- newIts(1:30, ncol=3, start="2003-01-11", end="2003-01-20")
appendIts(b, b1)
c <- union(b, b*NA)
c[2,4] <- 99
c[2,1] <- NA
print(c)
collapseIts(c)

```

itsLags

Lag Functions for Irregular Time-Series Objects

Description

Lag functions for objects of class "its".

Usage

```

lagIts(x, k=1)
lagdistIts(x, kmin, kmax)

```

Arguments

x an object of class "its"
k, kmin, kmax integer lag; positive value mean earlier data is assigned to a later timestamp

Details

lagIts returns an object with the same time-stamps, but with the data shifted. lagdistIts applies lagIts over a range of lags, and appends these columns

Value

An object of class "its".

Author(s)

Giles Heywood

See Also

[ts](#), [POSIXct](#), [itsFile](#), [itsLags](#) [itsJoin](#) [itsTimes](#) [itsSubset](#) [itsFin](#) [itsInterp](#)

Examples

```
b <- newIts(1:10)
lagIts(b)
lagdistIts(b[,1],1,3)
```

itsSubset

*Range and Extract for Irregular Time-Series Objects***Description**

Range and extract for objects of class "its".

Usage

```
rangeIts(x, start=dates(x)[1], end=dates(x)[nrow(x)], format=its.format(), ...)
extractIts(x, weekday=FALSE, find=c("all", "last", "first"), period=c("week", "month", "year"), partials=TRUE)
```

Arguments

x	an object of class "its"
start, end	POSIXct or character representation of the start or end time-stamp, if character, then the format is as specified by the argument format
format	format of "start" and "end" dates, see format.POSIXct .
dates	POSIX dates specifying the timestamps of rows to be extracted from the irregular time-series object
...	further arguments passed to as.POSIXct
weekday	logical, defines whether only weekdays are to be returned
find	to find the first, last, or all samples within each period
period	the period within which 'find' and/or 'select' operate
partials	defines whether the first (possibly incomplete) period is processed for find=first, and whether the last is processed for find=last
firstlast	if TRUE, the first and last observations are returned, in addition to those observations selected by other criteria
select	an integer vector defining one or more days to select. The integer specifies wday (for period=week) or mday (for period=month). See as.POSIXlt for details

Details

rangeIts selects a range of rows that fall between two times, specified in text format.

extractIts selects a subset of rows that obey some sort of semi-regular rule such as monthends, weekdays, and so on. The order of application is weekday, find, then select.

Value

An object of class "its".

Author(s)

Giles Heywood

See Also[ts](#), [POSIXct](#), [itsFile](#), [itsLags](#), [itsJoin](#), [itsTimes](#), [itsSubset](#), [itsFin](#), [itsInterp](#)**Examples**

```

its.format("%Y-%m-%d")
b <- newIts(start="2003-01-01",end="2003-01-20")
rangeIts(b,start = "2003-01-05" ,end= "2003-01-15")
rangeIts(b,start = ISOdate(2003,1,5,hour=0) ,end= ISOdate(2003,1,15,hour=0))
b[1:3,]
b[,1]
b[,dates=ISOdate(2003,1,1,hour=0,tz="")]
its.format("%a %d %b %Y")
c <- newIts()
extractIts(c,weekday=TRUE,period="month",find="last") #the last weekdays of the month in c

```

itsTimes

*Time Functions for Irregular Time-Series Objects***Description**

Time functions for objects of class "its".

Usage

```

daysecondIts(x,...)
weekdayIts(x,...)

```

Arguments

x an object of class "its"

... further arguments passed to [as.POSIXlt](#)

Details

daysecondIts returns the number of seconds since midnight.

weekdayIts tests which entries of an irregular time-series object are time-stamped with weekdays.

Value

For daysecondIts a vector of decimal numbers representing the number of seconds.

For weekdayIts a vector of "logical" representing the test results for each time.

Author(s)

Giles Heywood

See Also[ts](#), [POSIXct](#), [itsFile](#), [itsLags](#) [itsJoin](#) [itsTimes](#) [itsSubset](#) [itsFin](#) [itsInterp](#)**Examples**

```
## Not run:
its.format("%Y-%m-%d %A")
b <- newIts()
print(b)
daysecondIts(b)
weekdayIts(b)
its.format("%Y-%m-%d")

## End(Not run)
```

priceIts

Download Historical Finance Data

Description

Download historical financial data from a given data provider over the WWW.

Usage

```
priceIts(instruments = "^gdaxi", start, end,
         quote = c("Open", "High", "Low", "Close"),
         provider = "yahoo", method = "auto",
         origin = "1899-12-30", compression = "d", quiet=TRUE)
```

Arguments

<code>instruments</code>	a character string giving the name of the quote symbol to download. See the web page of the data provider for information about the available quote symbols.
<code>start</code>	an R object specifying the date of the start of the period to download. This must be in a form which is recognized by as.POSIXct , which includes R POSIX date/time objects, objects of class "date" (from package date) and "chron" and "dates" (from package chron), and character strings representing dates in ISO 8601 format. Defaults to 1992-01-02.
<code>end</code>	an R object specifying the end of the download period, see above. Defaults to yesterday.

quote	a character string or vector indicating whether to download opening, high, low, or closing quotes, or volume. For the default provider, this can be specified as "Open", "High", "Low", "Close", and "Volume", respectively. Abbreviations are allowed.
provider	a character string with the name of the data provider. Currently, only "yahoo" is implemented. See http://quote.yahoo.com/ for more information.
method	tool to be used for downloading the data. See download.file for the available download methods.
origin	an R object specifying the origin of the Julian dates, see above. Defaults to 1899-12-30 (Popular spreadsheet programs internally also use Julian dates with this origin).
quiet	a flag to suppress output
compression	Governs the granularity of the retrieved data; "d" for daily, "w" for weekly or "m" for monthly. Defaults to "d".

Details

This function is a direct copy from package `tseries`, and all credit must go to the author of that package.

Value

An "its" object with the requested data

Author(s)

A. Trapletti [**not** responsible for errors, adapted from his original idea by Giles Heywood]

See Also

[ts](#), [as.POSIXct](#), [download.file](#); <http://quote.yahoo.com/>

Examples

```
## Not run:
x1 <- priceIts(instrument = c("^ftse"), start = "1998-01-01",
              quote = "Close")
x2 <- priceIts(instrument = c("^gdax"), start = "1998-01-01",
              quote = "Close")
x <- union(x1,x2)
names(x) <- c("FTSE","DAX")
plot(x,lab=TRUE)

## End(Not run)
```

Index

- *Topic **classes**
 - its-class, 3
- *Topic **ts**
 - core, 2
 - dates, 2
 - itsConvert, 4
 - itsDef, 5
 - itsFile, 7
 - itsFin, 9
 - itsInterp, 10
 - itsJoin, 11
 - itsLags, 12
 - itsSubset, 13
 - itsTimes, 14
 - priceIts, 15
- [, its-method (its-class), 3
- [<-, its, ANY, ANY, its-method (its-class), 3
- accrueIts (itsFin), 9
- alignedIts (itsJoin), 11
- appendIts (itsJoin), 11
- Arith, its, its-method (its-class), 3
- Arith, its, numeric-method (its-class), 3
- Arith, numeric, its-method (its-class), 3
- array, 3
- as.data.frame.its (itsConvert), 4
- as.its (itsDef), 5
- as.list.its (itsConvert), 4
- as.POSIXct, 13, 15, 16
- as.POSIXlt, 13, 14
- coerce, its, data.frame-method (its-class), 3
- collapseIts (itsJoin), 11
- core, 2
- core, its-method (its-class), 3
- core<- (core), 2
- core<-, its, matrix-method (its-class), 3
- cumsum, its-method (its-class), 3
- dates, 2
- dates, its-method (its-class), 3
- dates<- (dates), 2
- dates<-, its, POSIXt-method (its-class), 3
- daysecondIts (itsTimes), 14
- diff, its-method (its-class), 3
- download.file, 16
- end, its-method (its-class), 3
- extractIts, 6
- extractIts (itsSubset), 13
- format.POSIXct, 6–8, 13
- intersect, its, its-method (its-class), 3
- intersect, its, NULL-method (its-class), 3
- intersect, NULL, its-method (its-class), 3
- is.its (itsDef), 5
- its, 6, 8
- its (itsDef), 5
- its-class, 3
- its.format, 8
- itsConvert, 4
- itsDef, 5
- itsFile, 7, 7–12, 14, 15
- itsFin, 7, 8, 9, 9–12, 14, 15
- itsInterp, 7–9, 10, 11, 12, 14, 15
- itsJoin, 7–10, 11, 11, 12, 14, 15
- itsLags, 7–11, 12, 12, 14, 15
- itsSubset, 7–12, 13, 14, 15
- itsTimes, 7–12, 14, 14, 15
- lagdistIts (itsLags), 12
- lagIts (itsLags), 12
- locf (itsInterp), 10
- matrix, 3
- names, its-method (its-class), 3
- names<-, its, character-method (its-class), 3

`newIts(itsDef)`, 5

`plot,its,missing-method(its-class)`, 3

`POSIXct`, 6–12, 14, 15

`priceIts`, 15

`print,its-method(its-class)`, 3

`rangeIts(itsSubset)`, 13

`read.csv`, 8

`readcsvIts(itsFile)`, 7

`seq.POSIXt`, 6

`start,its-method(its-class)`, 3

`structure`, 3

`summary,its-method(its-class)`, 3

`ts`, 7–12, 14–16

`union,its,its-method(its-class)`, 3

`union,its,NULL-method(its-class)`, 3

`union,NULL,its-method(its-class)`, 3

`vector`, 3

`weekdayIts(itsTimes)`, 14

`write.table`, 8

`writcsvIts(itsFile)`, 7