zoo: An S3 Class Providing Infrastructure for Totally Ordered Observations

Achim Zeileis
Wirtschaftsuniversität Wien

Gabor Grothendieck

Abstract

zoo provides infrastructure for ordered observations which are stored internally in a vector or matrix with an index attribute (with in principle arbitrary class, see below) which has to be of the same length as NROW(x).

Keywords: totally ordered observations, irregular time series, S3, R.

1. Introduction

zoo provides infrastructure for ordered observations which are stored internally in a vector or matrix with an index attribute (with in principle arbitrary class, see below) which has to be of the same length as NROW(x).

zoo is particularly aimed at irregular time series of numeric vectors/matrices. zoo's key design goals are independence of a particular index/date/time class and consistency (to the extent possible) with ts and base R by providing methods to standard generics. Therefore, standard functions can be used to work with "zoo" objects and no new set of commands has to be employed.

When creating a "zoo" object with the function zoo, the vector of indexes order.by can be of (a single) arbitrary class (if x is shorter or longer than order.by it is expanded accordingly), but it is essential that order(order.by) works. For other functions it is assumed that c(), length(), match() and subsetting [, work. If this is not the case for a particular index/date/time class, then methods for these generic functions should be created. Note, that to achieve this the non-generic base functions order and match are made generics in zoo with the base functions being the default (see order and match).

Methods to standard generics for "zoo" objects currently include: print (see above), summary, str, head, tail, [(subsetting), rbind, cbind, merge (see merge.zoo), aggregate (see aggregate.zoo), plot and lines (see plot.zoo).

Additionally, zoo provides several generic functions and methods to work (a) on the value or data contained in a "zoo" object, (b) the index (or time) attribute associated to it, and (c) on both data and index:

(a) The data contained in "zoo" objects can be extracted by value (strips off all "zoo"-specific attributes) and modified using value<-. Both are new generic functions with methods for "zoo" objects, see value.

(b) The index associated with a "zoo" object can be extracted by index and modified by index<-. As the interpretation of the index as "time" in time series applications is more natural, there are also synonymous methods time and time<-. The start and the end of the index/time vector can be queried by start and end. See index.

(c) To work on both data and index/time, zoo provides methods lag, diff (see lag.zoo) and window, window<- (see window.zoo).

In addition to standard group generic function (see Ops), the following mathematical operations are availabe as methods for "zoo" objects: transpose t which coerces to a matrix first, and cumsum,

`cumprod`, `cummin`, `cummax` which are applied column wise.

Coercion to and from `"zoo"` objects is available for objects of various classes, in particular `"ts"`, `"irts"` and `"its"` objects can be coerced to `"zoo"`, the reverse is available for `"its"` and for `"irts"` (the latter in package `tseries`). Furthermore, `"zoo"` objects can be coerced to vectors, matrices and data frames (dropping the index/time attribute). See `as.zoo`.

Two methods are available for `NA` handling in the data of `"zoo"` objects: `na.omit` which returns a `"zoo"` object with incomplete observations removed and `na.contiguous` which extracts the longest consecutive stretch of non-missing values in a `"zoo"` object. Note, that the letter function is made a generic in `zoo` with the base function being the default.

```
> library(zoo)
> x.date <- as.POSIXct(paste("2003-02-", c(1, 3, 7, 9, 14), sep = ""))
> x <- zoo(rnorm(5), x.date)
> plot(x)
> time(x)

[1] "2003-02-01 CET" "2003-02-03 CET" "2003-02-07 CET" "2003-02-09 CET"
[5] "2003-02-14 CET"

> x[1:3]

2003-02-01 2003-02-03 2003-02-07
 0.3021008  1.5694103 -0.5984546

> x.Date <- as.Date(paste("2003-02-", c(1, 3, 7, 9, 14), sep = ""))
> x <- zoo(rnorm(5), x.Date)
> plot(x)
> y.POSIXct <- ISOdatetime(2003, 2, c(1, 3, 7, 9, 14), 0, 0, 0)
> y <- zoo(rnorm(5), y.POSIXct)
> plot(y)
> z <- zoo(rnorm(5), runif(5))
> plot(z)
> z <- zoo(1, seq(4)[-2])
> z0 <- zoo(, 1:4)[, -1]
```

```
> lu <- get.hist.quote(instrument = "LU", start = "2001-01-01",
+     origin = "1970-01-01")

time series starts 2001-01-02
time series ends   2004-10-07

> LU <- zoo(value(lu), structure(time(lu) * 86400, class = c("POSIXt",
+     "POSIXct")))
> LU <- na.omit(LU)
> LU2 <- zoo(value(lu), structure(time(lu), class = "Date"))
> LU2 <- na.omit(LU2)
> plot(diff(log(LU)), col = list(High = 4, 2))
```