zoo FAQ

Gabor Grothendieck
GKX Associates Inc.

Achim Zeileis
Wirtschaftsuniversität Wien

Abstract

This is a collection of frequently asked questions (FAQ) about the **zoo** package together with their answers.

Keywords: irregular time series, daily data, weekly data, returns.

1. I know that duplicate times are not allowed but my data has them. What do I do?

In general, **zoo** functions will not accept **zoo** objects with duplicates times. `zoo` and `read.zoo` will issue warnings but will not prevent the creation of such objects in order to give the user a chance to fix them up – typically by using `aggregate.zoo` or `duplicated`. `merge.zoo` will issue an error message if such illegal operations are attempted with it. Since `merge` is the workhorse behind many **zoo** functions on that account alone many or most **zoo** functions will not work with duplicates among the times. Typically we eliminate duplicates by averaging over them, taking the last among each run of duplicates or interpolating the duplicates and deleting ones on the end that cannot be interpolated. These three approaches are shown here. Note that `force` is the identity function (i.e. it just returns its argument) and is an **R** core function:

```
> z <- suppressWarnings(zoo(1:8, c(1, 2, 2, 2, 3, 4, 5, 5)))
> z
```

```
1 2 2 2 3 4 5 5
1 2 3 4 5 6 7 8
```

```
> aggregate(z, force, mean)
```

```
  1   2   3   4   5
1.0 3.0 5.0 6.0 7.5
```

```
> aggregate(z, force, tail, 1)
```

```
1 2 3 4 5
1 4 5 6 8
```

```
> time(z) <- na.approx(ifelse(duplicated(time(z)), NA, time(z)),
+     na.rm = FALSE)
> z[!is.na(time(z))]
```

```
     1      2 2.3333 2.6667      3      4      5
     1      2      3      4      5      6      7
```

2. When I try to specify a log axis to plot.zoo a warning is issued. What is wrong?

Arguments that are part of `...` are passed to the `panel` function and the default `panel` function, `lines`, does not accept `log`. Either ignore the warning, use `suppressWarnings` (see `?suppressWarnings`) or create your own panel function which excludes the `log`:

```
> z <- zoo(1:100)
> plot(z, log = "y", panel = function(..., log) lines(...))
```

3. How do I create right and a left vertical axes in plot.zoo?

The following shows an example of creating a single panel plot using `plot.zoo` and both left and right axes.

```
> set.seed(1)
> z.Date <- as.Date(paste(2003, 2, c(1, 3, 7, 9, 14), sep = "-"))
> z <- zoo(cbind(left = rnorm(5), right = rnorm(5, sd = 0.2)),
+     z.Date)
> plot(z[, 1], xlab = "Time", ylab = "")
> opar <- par(usr = c(par("usr")[1:2], range(z[, 2])))
> lines(z[, 2], lty = 2)
> Axis(side = 4)
> legend("bottomright", lty = 1:2, legend = colnames(z))
> par(opar)
```

4. How do I create a monthly series but still keep track of the dates?

Create a **S3** subclass of `"yearmon"` called `"yearmon2"` that stores the dates as names on the time vector. It will be sufficient to create an `as.yearmon2` generic together with `as.Date.yearmon2` and `as.yearmon2.Date` methods. It will act the same as `"yearmon"` but we can recover the dates at any time using `as.Date` and `aggregate.zoo` as shown in the test at the end of this example:

```
> as.yearmon2 <- function(x, ...) UseMethod("as.yearmon2")
> as.yearmon2.Date <- function(x, ...) {
+     y <- as.yearmon(with(as.POSIXlt(x, tz = "GMT"), 1900 + year +
+         mon/12))
+     names(y) <- x
+     structure(y, class = c("yearmon2", class(y)))
+ }
> as.Date.yearmon2 <- function(x, frac = 0, ...) {
+     if (!is.null(names(x)))
+         return(as.Date(names(x)))
+     x <- unclass(x)
+     year <- floor(x + 0.001)
+     month <- floor(12 * (x - year) + 1 + 0.5 + 0.001)
+     dd.start <- as.Date(paste(year, month, 1, sep = "-"))
+     dd.end <- dd.start + 32 - as.numeric(format(dd.start + 32,
+         "%d"))
+     as.Date((1 - frac) * as.numeric(dd.start) + frac * as.numeric(dd.end),
+         origin = "1970-01-01")
+ }
> dd <- seq(as.Date("2000-01-01"), length = 5, by = 32)
> z <- zoo(1:5, as.yearmon2(dd))
> z
```

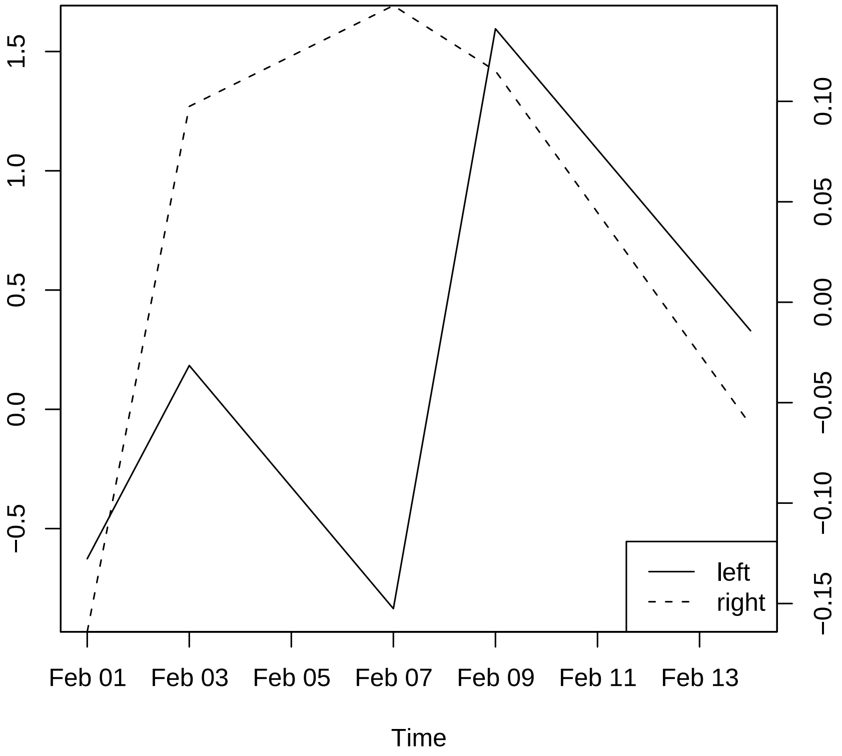

Figure 1: Left and right plot.zoo axes.

```
Jan 2000 Feb 2000 Mar 2000 Apr 2000 May 2000
       1        2        3        4        5

> aggregate(z, as.Date, force)

2000-01-01 2000-02-02 2000-03-05 2000-04-06 2000-05-08
         1          2          3          4          5
```

5. I have data frame with both numeric and factor columns. How do I convert that to a zoo object?

A `"zoo"` object may be a numeric vector or matrix or a factor but not both. You can convert it to two `"zoo"` variables:

```
> DF <- data.frame(time = 1:4, x = 1:4, f = factor(1:4))
> zx <- zoo(DF$x, DF$time)
> zf <- zoo(DF$f, DF$time)
```

or you can convert the factor to numeric and then create a single `"zoo"` series:

```
> z <- zoo(data.matrix(DF[-1]), DF$time)
```

6. Why does lag these give slightly different results on a zoo and a zooreg series which are otherwise the same?

To be definite let us consider the following examples:

```
> z <- zoo(11:15, as.Date("2008-01-01") + c(-4, 1, 2, 3, 6))
> zr <- as.zooreg(z)
> lag(z)

2007-12-28 2008-01-02 2008-01-03 2008-01-04
        12         13         14         15

> lag(zr)

2007-12-27 2008-01-01 2008-01-02 2008-01-03 2008-01-06
        11         12         13         14         15

> diff(log(z))

2008-01-02 2008-01-03 2008-01-04 2008-01-07
0.08701138 0.08004271 0.07410797 0.06899287

> diff(log(zr))

2008-01-03 2008-01-04
0.08004271 0.07410797
```

`lag.zoo` and `lag.zooreg` work differently. For `"zoo"` objects the lagged version is obtained by moving the time points to the adjacent time point that exists in the series but for `"zooreg"` objects the time is lagged by `deltat`, the time between adjacent regular times.

A key difference is that in "zooreg" one can lag a point to a time point that did not previously exist in the series and, in particular, can lag a series outside of the original time range whereas that is not possible in a "zoo" series.

Note that lag.zoo has an na.pad= argument which may be what you are looking for.

The difference between diff.zoo and diff.zooreg stems from the fact that diff(x) is defined in terms of lag like this: x-lag(x,-1).

7. How do I subtract the mean of each month from a zoo series?

To subtract the mean of Jan 2007 from each day in that month, subtract the mean of Feb 2007 from each day in that month, etc. try this:

```
> set.seed(123)
> z <- zoo(rnorm(100), as.Date("2007-01-01") + seq(0, by = 10,
+     length = 100))
> z.demean1 <- z - ave(z, as.yearmon(time(z)))
```

To subtract the mean of all Januaries from each January, etc. try this:

```
> z.demean2 <- z - ave(z, format(time(z), "%m"))
```