

# Can I perform autocorrelation / lag analysis on a zoo object in R with non-regular time steps? If so, how?

Asked 7 years, 7 months ago   Active 7 years, 7 months ago   Viewed 2k times



Can I perform autocorrelation / lag analysis on a zoo object in R with non-regular time steps? If so, how?

1



The only other post I could find here dealt with regular time series. I have a sequence of observations taken at irregular time steps. For example,  $(t, y) = (0, 2668), (36.62, 2723), (42, 2723), \dots$  where



1

- $t$  is the time in hours, and
- $y$  is the (categorical\*) observation. ... \*edited from original post

I would like to look for lag correlations daily ( $\text{lag} = 24$ ) and weekly ( $\text{lag} = 168$ ) to see whether certain categories of observation repeat at / near these lag intervals. Is there a way to do this in R? I created a zoo object for my data but have been unable to find any documentation concerning how to do this.

[r](#) [correlation](#) [lag](#) [zoo](#)

edited Jan 23 '12 at 15:28

asked Jan 23 '12 at 0:34



[occasionalUser](#)

34 8

1 I don't know much about `zoo` objects, but there is a method in the `nlme` package (`corCAR1`) for incorporating first-order autoregression in a model with unevenly spaced data (using `g[n]ls` or `[n]lme`). – [Ben Bolker](#) Jan 23 '12 at 13:33

Thanks! This looks like a great package that I can use on some other data I have to analyze. I didn't realize until after my initial post that I was being silly ... my observations are categorical. Nevertheless, while I don't think I can use this package on this data, I think I can use it later. – [occasionalUser](#) Jan 23 '12 at 17:11

## 1 Answer



You can use `aggregate` to convert your data into daily & weekly intervals, and then calculate the autocorrelation with whatever function does it for regular time series (say `acf`). e.g.:

2



```
# make a data set to play with
library(zoo)
ts <- sort(runif(100)*168*3) # 100 observations over 3 weeks
ys <- runif(100)           # y values
z <- zoo(ys, order.by=ts)
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```
# Now compute correlation, lag 1 (index in z.daily/weekly)
daily.acf <- acf(z.daily, lag.max=1)[1]
weekly.acf <- acf(z.weekly, lag.max=1)[1]
```

The `aggregate` converts `z` to daily or weekly data where you sum all occurrences for each day/week. It does the grouping by looking at `index(z) %/% 24` (or 168) which is the integer part of the hour of observation divided by 24 (ie, the day it occurs).

Then the `acf` function calculates autocorrelation (with the `lag` being on indices of the vector, not on time).

I don't really know much about statistics, and one thing I noticed was that if you do:

```
weekly.acf <- acf(z.daily, lag.max=7)[7]
```

you get a different answer from when you calculate autocorrelation from `z.weekly`, because it's doing autocorrelation on *daily* data with a lag of 7 as opposed to weekly data with a lag of 1 -- so I'm not sure if what I'm doing is actually what you want.

answered Jan 23 '12 at 1:39



[mathematical.coffee](#)

43.1k 8 109 160

---

Hi. Thank you for your help. I played around with this and then realized I was being silly because my `y` observations are categorical, not continuous. Thus, I can't sum the observations to get meaningful data. I was hoping to find if any categories repeat at / near certain lags but this may be difficult since I don't have data measurements at those regularly-spaced time intervals. If I am working with continuous data in the future, I will remember this `aggregate` though. Thanks for your help! – [occasionalUser](#) Jan 23 '12 at 15:22

---