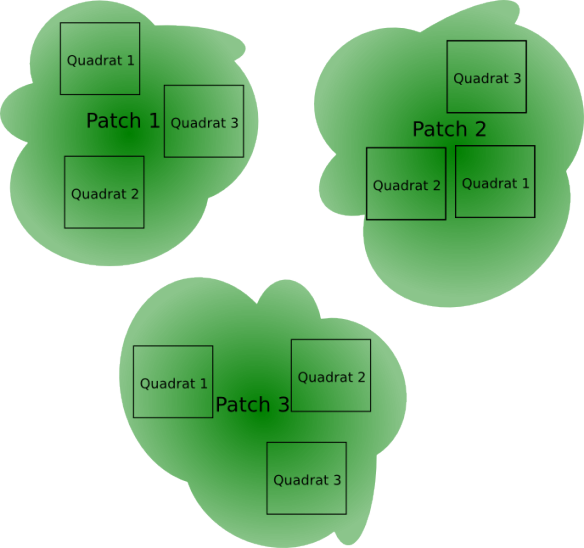
Reproduced from: <http://climateecology.wordpress.com/2012/07/17/r-for-ecologists-creating-a-site-x-species-matrix/>

**R for Ecologists: Creating a Site x Species Matrix**

Posted on [**July 17, 2012**](http://climateecology.wordpress.com/2012/07/17/r-for-ecologists-creating-a-site-x-species-matrix/) by [**Nathan Lemoine**](http://climateecology.wordpress.com/author/vcae/)

Today I’m going to address a fairly common problem in ecology that has been coming up frequently as of late. The issue is how to create a site x species matrix for community composition analysis (*i.e.* ordination). Moreover, ecologists have to do a good deal of averaging if data are collected on multiple levels, such as averaging quadrats within plots. R makes doing this easy. For example, suppose I’m collecting data on herbaceous plant cover in forest patches. I set up several quadrats within a patch, but the patch is my true replicate, so I have to average over the quadrats within each patch.

[](http://climateecology.files.wordpress.com/2012/07/forest.png)

Example of experimental layout. Here, there are three replicate patches, so I have to average quadrats within each patch.

Also, if you’re like most field ecologists, you probably don’t take a spreadsheet into the field with 200 columns, one for each species, so you can mark zeroes down for things that are absent. You probably have something like this:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Year | Patch | Quadrat | Species | Cover |
| 2012 | 1 | 1 | V. dentatum | 1 |
| 2012 | 1 | 1 | R. phoenicolasius | 20 |
| : | : | : | : | : |
| 2012 | 1 | 2 | V. dentatum | 2 |

Now you have two issues: Your data is not in a site x species format, and even if it were, you need to add zeroes in for the species that weren’t observed in the quadrats because you only recorded what you saw, not what you didn’t. This becomes an issue for averaging over quadrats. The zeros **must** be included when taking the average of the quadrats. Fortunately, the ‘*reshape*‘ package of R makes this a snap to do.

Let’s suppose the above data were imported as a dataframe called ‘Patch.Cover’ and you have, being forward-thinking, already loaded the ‘reshape’ package. The first thing to do is to ‘cast’ your data into a site by species matrix, including all sampling levels. We keep the quadrat level in for now:

|  |  |
| --- | --- |
| 1 | site.sp.quad <- cast(Patch.Cover, Year + Plot + Quad ~ Species, value='Cover',  FUN=mean) |

This code is pretty straightforward. The first argument is the dataframe. The second is a formula: on the left hand side are the factors you want to include and the right hand side is the factor you want to ‘transpose’, or convert into columns. FUN=mean tells it to take the mean value, which at this level, is just the observed cover because we’ve included all factors (*i.e.* it’s not going to aggregate or average over anything because each row has a unique identifier).

For whatever reason, ‘cast’ returns a funky output of class ‘cast’, so we need to make it a dataframe before we can mess with it.

|  |  |
| --- | --- |
| 1 | site.sp.quad <- as.data.frame(site.sp.quad) |

Now every year/plot/quadrat combination has been given a value for every species. This is great if zeroes had been recorded, but they were not. Instead, there is now an ‘NaN’ value where there was no cover entry*.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Year | Patch | Quadrat | R. phoenicoluasius | V. dentatum |
| 2012 | 1 | 1 | 20 | 1 |
| 2012 | 1 | 2 | NaN | 2 |

We convert those NaN’s to 0.

|  |  |
| --- | --- |
| 1 | site.sp.quad[is.na(site.sp.quad)] <- 0 |

We now have a full site x species matrix at the quadrat level that might look something like this.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Year | Patch | Quadrat | R. phoenicoluasius | V. dentatum |
| 2012 | 1 | 1 | 20 | 1 |
| 2012 | 1 | 2 | 0 | 2 |

But we’re not done, we now have to average quadrats within plots. After thinking about it some, the best way I came up with was to collapse the data back into column form and then recast it at the plot level.

First, ‘melt’ the data back into column form:

|  |  |
| --- | --- |
| 1 | column.quad <- melt(site.sp.quad, id=c('Year', 'Plot', 'Quad')) |

The ‘id’ argument is a vector of the factors assigned to each plot, ‘melt’ figures out the rest. Now you have a much bigger column dataframe because zeroes have been added in. Now we can properly average over the plots:

|  |  |
| --- | --- |
| 1 | site.sp.plot <- cast(column.quad, Year + Plot ~ variable, value='value', FUN=mean) |

We left ‘Quadrat’ out of the formula so now each row is identified only by year and plot. Thus, each year/plot combination has multiple entries (one for each quadrat, three in the patch example) and FUN=mean takes the average of those entries. Notice ‘variable’ and ‘value’. That’s because the melt( ) command made the column name of the species column ‘variable’ and the cover column was called ‘value’. You can change these to whatever you want, but I generally don’t see the purpose.

So there it is, you’ve now got a site x species matrix appropriately formatted and averaged for all of your ordination and multivariate needs (which are another headache altogether). For anyone interested, I wrote a custom function that collapses all of these steps into one function:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | site.sp <- function(data.frame, factors, obs, value, avg.over=''){  require(reshape)  fmla <- as.formula(paste(paste(factors, collapse='+'), obs, sep='~'))    full.data <- cast(data.frame, fmla, value=value, mean)  full.data <- as.data.frame(full.data)  full.data[is.na(full.data)] <- 0    col.full <- melt(full.data, id=factors)    factors.agg <- factors[which(factors!=avg.over)]  agg.fmla <- as.formula(paste(paste(factors.agg, collapse="+"), 'variable', sep='~'))  agg.data <- cast(col.full, agg.fmla, value=value, mean)    site.sp.data <<- as.data.frame(agg.data)   } |

It takes five arguments: data.frame (self explanatory), factors (a vector of ALL factors identifying a quadrat: Year, Plot, and Quadrat in the above example), obs (the name of the column including species names, ‘Species’ in the above example), value (the response variable, ‘Cover’ in the above example, but could be density or whatever), and avg.over (the factor you want to average out, ‘Quadrat’ in the above example). The default for avg.over is blank, so that if you don’t address this argument, you get a properly formatted non-averaged site x species matrix with all the factors.

I hope this helps. Feel free to critique the code and offer improvements.