Data Formatting

Nicholas J. Gotelli

February 2, 2017

This vignette gives some simple code for reformatting data sets into the long format.

```
library(reshape)
library(xtable)
set.seed=100 # for reproducible results
```

Community Ecology Abundance Matrices

A very commmon data format for community ecologists is an abundance or incidence matrix in which the rows are species or taxa, the columns are sites or samples, and the entries are abundance or incidence of a taxa in a site. Here is an example of a tiny data set in this format:

```
MyData <- matrix(rpois(50,lambda=1),nrow=5) # create a matrix of random integers, including some zeroes
rownames(MyData) <- paste("Species",as.character(1:5),sep="")
colnames(MyData) <-paste("Site",LETTERS[1:10],sep="")

print(xtable(MyData), type = "html",
     html.table.attributes = "align = 'center'") #demo for basic use of xtable package with html
```

	SiteA	SiteB	SiteC	SiteD	SiteE	SiteF	SiteG	SiteH	SiteI	SiteJ
Species1	0	1	0	3	0	1	0	1	0	0
Species2	3	0	0	2	0	2	0	1	2	0
Species3	0	1	0	2	0	0	1	1	2	0
Species4	2	1	1	2	1	0	2	3	1	1
Species5	1	0	2	0	0	1	3	0	0	1

Converting an Abundance Matrix into the Long Format

Here we will use the `reshape` package to turn this matrix into a data frame in the more usable long format:

```
MyData.Long <- melt(MyData,id=colnames(MyData))
print(MyData.Long)
```

```
##          X1    X2 value
## 1  Species1 SiteA     0
## 2  Species2 SiteA     3
## 3  Species3 SiteA     0
## 4  Species4 SiteA     2
## 5  Species5 SiteA     1
## 6  Species1 SiteB     1
## 7  Species2 SiteB     0
## 8  Species3 SiteB     1
## 9  Species4 SiteB     1
## 10 Species5 SiteB     0
## 11 Species1 SiteC     0
## 12 Species2 SiteC     0
## 13 Species3 SiteC     0
## 14 Species4 SiteC     1
## 15 Species5 SiteC     2
## 16 Species1 SiteD     3
## 17 Species2 SiteD     2
## 18 Species3 SiteD     2
## 19 Species4 SiteD     2
## 20 Species5 SiteD     0
## 21 Species1 SiteE     0
## 22 Species2 SiteE     0
## 23 Species3 SiteE     0
## 24 Species4 SiteE     1
## 25 Species5 SiteE     0
## 26 Species1 SiteF     1
## 27 Species2 SiteF     2
## 28 Species3 SiteF     0
## 29 Species4 SiteF     0
## 30 Species5 SiteF     1
## 31 Species1 SiteG     0
## 32 Species2 SiteG     0
## 33 Species3 SiteG     1
## 34 Species4 SiteG     2
## 35 Species5 SiteG     3
## 36 Species1 SiteH     1
## 37 Species2 SiteH     1
## 38 Species3 SiteH     1
## 39 Species4 SiteH     3
## 40 Species5 SiteH     0
## 41 Species1 SiteI     0
## 42 Species2 SiteI     2
## 43 Species3 SiteI     2
## 44 Species4 SiteI     1
## 45 Species5 SiteI     0
## 46 Species1 SiteJ     0
## 47 Species2 SiteJ     0
## 48 Species3 SiteJ     0
## 49 Species4 SiteJ     1
## 50 Species5 SiteJ     1
```

Let’s rename the variables and then check the structure:

```
MyData.Long <- rename(MyData.Long,c(X1="Species",X2="Site",value="Abundance"))
head(MyData.Long) # print the first 6 lines
```

```
##    Species  Site Abundance
## 1 Species1 SiteA         0
## 2 Species2 SiteA         3
## 3 Species3 SiteA         0
## 4 Species4 SiteA         2
## 5 Species5 SiteA         1
## 6 Species1 SiteB         1
```

```
str(MyData.Long)
```

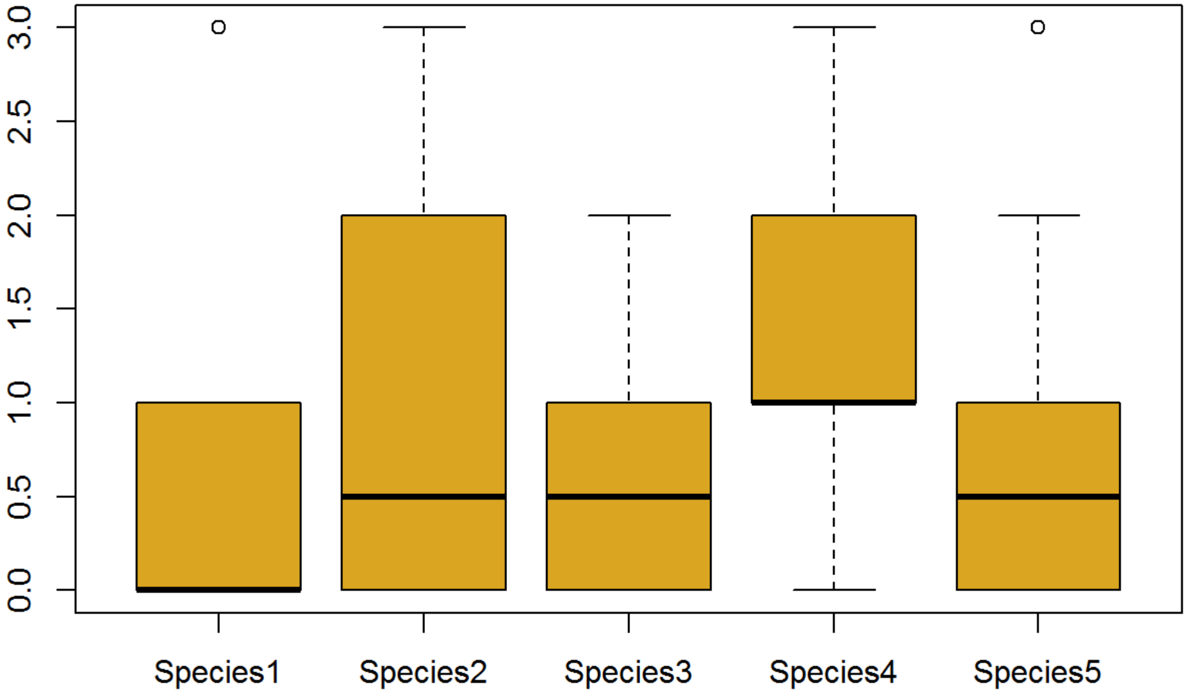
```
## 'data.frame':    50 obs. of  3 variables:
##  $ Species  : Factor w/ 5 levels "Species1","Species2",..: 1 2 3 4 5 1 2 3 4 5 ...
##  $ Site     : Factor w/ 10 levels "SiteA","SiteB",..: 1 1 1 1 1 2 2 2 2 2 ...
##  $ Abundance: int  0 3 0 2 1 1 0 1 1 0 ...
```

Notice how the `reshape` package has converted the data from a matrix to a data frame, with factors properly set up for `Species` and `Site`. Very nice! In this format, it is easy, for example, to conduct a one-way analysis of variance to test for differences in average abundance among species and then plot the results in a boxplot.

```
attach(MyData.Long)
Species.ANOVA <- aov(Abundance~Species)
summary(Species.ANOVA)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Species      4    4.0  1.0000   1.059  0.388
## Residuals   45   42.5  0.9444
```

```
boxplot(Abundance~Species,col="goldenrod")
```



Converting from the Long Format Back to a Species Abundance Matrix

Once your data are in the long format, it is easy to convert them to other structures. For example, to recreate the original data matrix we use the `cast` function:

```
Abundance.Matrix <- cast(MyData.Long,Species~Site,value="Abundance")
print(Abundance.Matrix)
```

```
##    Species SiteA SiteB SiteC SiteD SiteE SiteF SiteG SiteH SiteI SiteJ
## 1 Species1     0     1     0     3     0     1     0     1     0     0
## 2 Species2     3     0     0     2     0     2     0     1     2     0
## 3 Species3     0     1     0     2     0     0     1     1     2     0
## 4 Species4     2     1     1     2     1     0     2     3     1     1
## 5 Species5     1     0     2     0     0     1     3     0     0     1
```

Converting From the Long Format with Aggregation

Often when we convert the data from the long format, we will want to aggregate. For example, here is an aggregated table giving the mean abundances of each species:

```
Species.Means <- cast(MyData.Long,Species~.,value="Abundance",mean)
print(Species.Means)
```

```
##    Species (all)
## 1 Species1   0.6
## 2 Species2   1.0
## 3 Species3   0.7
## 4 Species4   1.4
## 5 Species5   0.8
```

Alternatively, we could use this function to calculate the total abundance in each of the samples

```
Site.Abundance <- cast(MyData.Long, Site~., value = "Abundance", sum)
print(Site.Abundance)
```

```
##     Site (all)
## 1  SiteA     6
## 2  SiteB     3
## 3  SiteC     3
## 4  SiteD     9
## 5  SiteE     1
## 6  SiteF     4
## 7  SiteG     6
## 8  SiteH     6
## 9  SiteI     5
## 10 SiteJ     2
```

The `aggregate` function works the same way (and maintains the variable label for abundance):

```
Species.Means2 <- aggregate(Abundance ~ Species, data = MyData.Long, mean)
print(Species.Means2)
```

```
##    Species Abundance
## 1 Species1       0.6
## 2 Species2       1.0
## 3 Species3       0.7
## 4 Species4       1.4
## 5 Species5       0.8
```