2. Data Screening for Multivariate Analysis

FISH 560: Applied Multivariate Statistics for Ecologists

Topics

- Checking for data irregularities
- Transforming and standardizing data

R Libraries: vegan
R Source: biostats

BACKGROUND

The development of an appropriate multivariate analytical strategy for a given data set should proceed as a careful sequence of steps, in which results obtained at a given step determine subsequent ones (Kenkel 2006). Multivariate data analysis is thus a process of adaptive learning, in which decisions made at a given analytical stage direct subsequent steps and strategies. Before proceeding with a formal multivariate analysis, it is critically important to complete a detailed exploratory analysis of the data. Exploratory analysis is undertaken to elucidate and summarize distributional properties and underlying trends of the data, which in turn direct the user to meaningful analyses and interpretations. In other words, data screening is an essential precursor to all multivariate analyses. Below we discuss how to search and correct data errors, treat missing data, detect and handle insufficiently sampled variables, conduct transformations and standardizations, and detect and handle outliers. This document is based largely on the functions (and help files) available in BIOSTATS written by Kevin McGarigal at the University of Massachusetts.

SET-UP

First, set-up your R work session by setting the current work directory to your folder of choice and load the VEGAN library and BIOSTATS "library" from the *File* pull-down menu. You can also do this using the functions **`setwd`**, **`library`** and **`source`** (see Chapter 1: Introduction to R)

Second, import the MAHA environment dataset and the MAHA species abundance dataset by typing:

```
envdata <- read.csv('MAHA_environment.csv',header=TRUE, row.names=1)
spedata <- read.csv('MAHA_speciesabu.csv',header=TRUE, row.names=1)
```

We can examine the structure of the data set by typing:

```
str(envdata)
```

CALCULATE SUMMARY STATISTICS AND SCREEN FOR DATA ERRORS

Let's explore the data and screen for errors by calculating a series of summary statistics. In particular, note the minimum and maximum values for obvious errors, and examine the summary statistics to gain a general understanding of the range of values observed and the frequency of zeros and missing values. There are two basic options here, summary by row (observation or record) and by column (variable).

Mixed data set option: If the data set consists of unrelated variables or variables on different scales of measurement (e.g., environmental variables), then only a column summary is meaningful and the following BIOSTATS function is available by typing:

col.summary(envdata)

This will return the following:

```
             Sinuosity  Slope WDRatio SubEmbed HabQual      Elev RoadDen
nobs            45.000 45.000  45.000   45.000  45.000    45.000  45.000
min              1.007  0.155   8.154   10.612   7.080   225.000   4.420
max              2.007  9.100  53.018   96.909  18.830  2720.000  82.580
mean             1.151  1.524  20.779   57.274  13.578  1425.133  14.959
. . .
```

Homogeneous data set option: If on the other hand the data set consists of a homogeneous set of variables measured on the same scale (e.g., species abundances), then both a row summary and column summary is meaningful.

To produce a column summary (i.e. species), type:

sum.stats(spedata)

To produce a column summary for just the 4 first species, type:

sum.stats(spedata,var='BANDDART:BLUECHUB')

To produce a row summary (i.e. sites), type:

sum.stats(spedata, margin='row')

This will return the following:

```
       nobs min     max    mean  median      sum      sd      cv  xeros
nobs     45  45  45.000  45.000  45.000   45.000  45.000  45.000 45.000
min      36   0   6.000   0.222   0.000    8.000   1.017 173.390 18.000
max      36   0 815.000  40.250   0.500 1449.000 147.177 594.962 34.000
mean     36   0  85.800   5.734   0.022  206.422  17.278 319.809 27.267
. . .
```

EVALUATE MISSING DATA AND TAKE CORRECTIVE ACTION

Review the column and row summary statistics above and note the amount and pattern of missing data. If missing data values exist and are sparsely distributed throughout the data frame, consider eliminating data with missing data or replacing the small amounts of missing values with either the median and mean of the column or using statistical interpolation. For count variables (e.g., species abundance data), the median is perhaps the more logical replacement, as it maintains the integer status of all values. To accomplish a 'median' replacement, type:

testdata <- replace.missing(spedata)

Note that testdata is identical to spedata because the original data contains no missing values.

Alternatively, with other types of numeric data the column mean may by considered a logical replacement as well. To accomplish a 'mean' replacement, type:

testdata <- replace.missing(spedata,method='mean')

If missing data values exist and are concentrated in one or more samples and/or one or more variables, consider dropping the offending samples and/or variables. For example, consider dropping variables with too many missing values, say >5 percent of observations missing, using the **drop.var()** function, by typing:

```
testdata <- drop.var(spedata, pct.missing=5)
```

Again, note that testdata is identical to spedata because the original data contains no missing values. What a wonderful dataset!

EXAMINE FREQUENCY OF OCCURRENCE AND ABUNDANCE GRAPHS AS APPROPRIATE

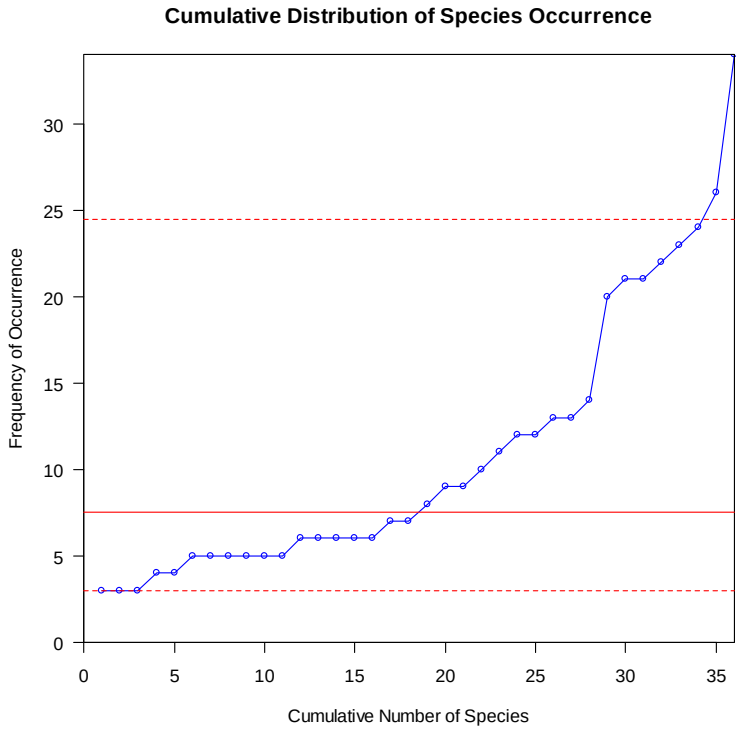
If you are analyzing community data (such as the MAHA data set) that contain samples by species abundances or occurrence, there are several things about the species and samples you may wish to know:

1. In how many sites does each species occur?
2. What is the mean abundance of each species when it occurs (not averaging zeros for sites where it is absent)?
3. Is the mean abundance of species correlated with the number of sites in which it occurs?
4. How many species occur in each site?
5. Is the total abundance of species in a site correlated with the number of species in a site?

Foa.plots() produces a series of 10 different plots that can help you answer these questions and more. To view the plots for the MAHA fish abundance data set (spedata), type:

```
foa.plots(spedata)
```

This will return the following in the Graphics window:

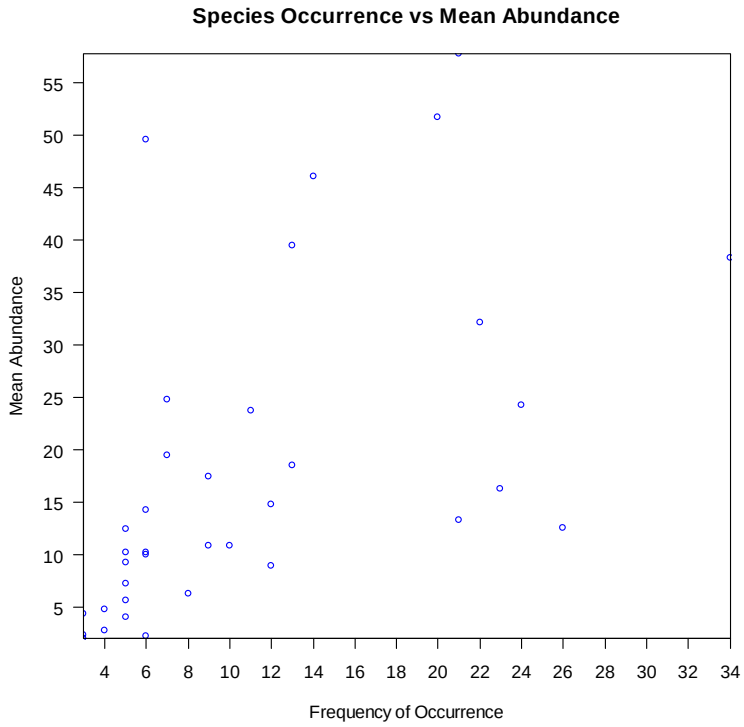


In how many sites does each species occur? The first four plots portray the species' frequency of occurrence among plots in a number of different ways – either as an empirical cumulative distribution function (ECDF) of species occurrence or as a histogram of species occurrence. Because some ecologists have suggested that species occurrence and abundance distributions sometimes follow a log-normal distribution, the fourth plot depicts a histogram of log-transformed species occurrence.

What is the mean abundance of each species when it occurs (not averaging zeros for sites where it is absent)? The fifth plot is an ECDF of species mean abundance.

Is the mean abundance of species correlated with the number of plots they occur in? The sixth plot is a scatter plot of frequency of occurrence against mean abundance. Is there any apparent relationship between the two? Are the widespread species also generally more abundant? Are there many widespread species that occur at low abundance? Conversely, are there species much less widespread, but abundant where they occur? To see which dot is which species, answer 'y' to the question on the console and then simply click on the point and the species acronym (variable name) will appear next to the point. When you are done identifying points, simply right click and choose 'stop'.

This will return the following in the Graphics window:



As before, it may be more meaningful to look at the log of mean abundance, as often abundance follows a log-normal distribution. The seventh plot is the same as above except that mean abundance has been log-transformed. Are the patterns the same?

Is the total abundance of species in a site correlated with the number of species in a site? To answer this question, first it is instructive to look at the number of species per site. The eighth plot depicts the ECDF of site richness. Are there any interesting patterns? For example, do most sites support an average number of species, while only a few sites supporting either very few or very many species? Or is the pattern different?

Second, what is the pattern in the distribution of site total abundance? The ninth plot is the ECDF of total site abundance. How does it compare to the ECDF of site richness?

Finally, to answer the question on the relation between total abundance and number of species/site, the last plot is a scatter plot of the two variables. Is there is relationship between the number of species per site and the total abundance? Do species-rich sites generally support a greater total abundance of species as well?

EXAMINE VARIABLES FOR "SUFFICIENCY" AND ELIMINATE IF NECESSARY

It is important to screen your data for insufficient variables (i.e., rare species, abundant species, variables with low variation: see lecture notes for a detailed discussion).

Review the column summary statistics above and note the number and percentage of zeros and the coefficient of variation of each variable. Consider dropping rare species having non-zero values in less than 5% of the sites, by typing:

```
testdata <- drop.var(spedata,min.po=5)
```

This should result in no change in the dataset (type **`str(testdata)`** to examine this).

Now, consider dropping variables with fewer than 5 non-zero values (i.e., sites), by typing:

```
testdata <- drop.var(spedata,min.fo=5)
str(spedata)
str(testdata)
```

The following will be returned and will indicate that 5 species have been deleted from the data set.

```
'data.frame':   45 obs. of  36 variables:
 $ BANDDART: int  0 0 0 0 0 0 0 0 5 0 ...
 $ BANDSCUL: int  0 0 0 0 0 2 0 0 0 0 ...
 $ BLACDACE: int  7 1 16 114 15 1 73 97 0 0 ...
. . .

'data.frame':   45 obs. of  31 variables:
 $ BANDSCUL: int  0 0 0 0 0 2 0 0 0 0 ...
 $ BLACDACE: int  7 1 16 114 15 1 73 97 0 0 ...
 $ BLUECHUB: int  372 91 1 0 3 0 0 0 0 0 ...
. . .
```

In community data sets, consider dropping abundant generalist species, for example those species occurring in more than 95% of the plots, by typing:

```
testdata <- drop.var(spedata,max.po=95)
```

Depending on the subsequent analysis, dropping abundant generalist species may or may not be desirable, so this decision should be made carefully.

Consider dropping variables with too little variation, for example those with cv<5, by typing:

```
testdata <- drop.var(spedata,min.cv=5)
```

Note: dropping variables with too little variation is unlikely to affect subsequent analyses, because they will not exert much influence over the data cloud.

EXAMINE THE DISTRIBUTIONAL PROPERTIES OF THE DATA

It is important to examine the distribution of each variable independently and, among other things, note the need for transformations to improve characteristics of the distribution (e.g., normality). Of course, there are many ways to examine distributions and we will not attempt an exhaustive review. Instead, we will focus on four common graphical approaches suitable for continuous variables. Note: categorical variables (or factors) are not suitable for these functions and will produce an error if they are not first removed from the input data set.

Empirical cumulative distribution functions: The empirical cumulative distribution function (ECDF) is a simple rank order distribution of increasing values of the variable. A variable with a perfectly uniform distribution of values within its range (minimum to maximum), so that no one value is more common than another, will have points that fall on a perfect diagonal straight line. Deviations from the diagonal indicate non-uniformity. Plot the ECDF for each species (in turn) by typing:

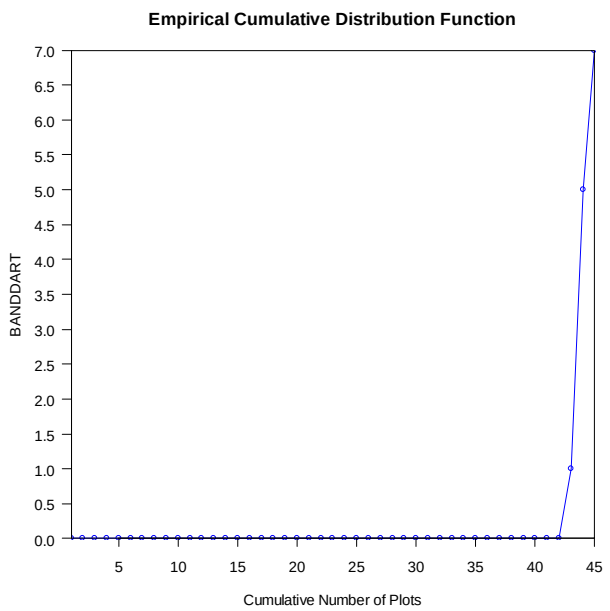
```
ecdf.plots(spedata)
```

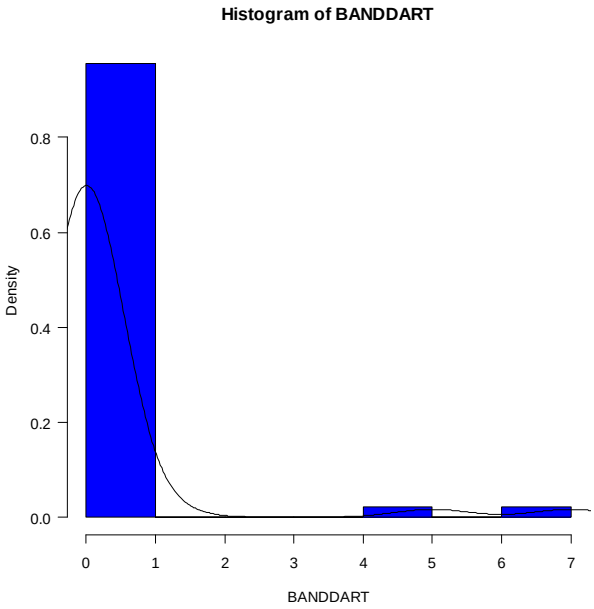
Histograms: Next, try the more conventional histogram, by typing:

```
hist.plots(spedata)
```

Ecological data are often highly skewed, so evaluate each distribution for skewness. You should also pay attention to the occurrence of extreme values in the distribution (i.e., potential outliers), as we will address this issue below.

The following two plots will be returned separately:





Box-and-whisker plots: An alternative way to examine the distribution of each variable is with a box-and-whisker plot, obtained by typing:

```
box.plots(spedata)
```

Box plots can depict the skewness of the distribution quite nicely and also can be used to identify extreme observations. The central box shows the data between the 'hinges' (roughly quartiles), with the median represented by a line. 'Whiskers' go out to the extremes of the data, and very extreme points (defined as samples that are farther than 1.5 (by default) times the inter-quartile range from the box) are shown by themselves.

Normal quantile-quantile plots: Another useful way of examining the distribution of each variable is to compare the empirical cumulative distribution function (ECDF) to the expected ECDF for a normal distribution. A normal quantile-quantile (or QQ) plot does just this. Try typing:

```
qqnorm.plots(spedata)
```

Note: the qqnorm plot depicts the sample quantiles on the x axis against the theoretical quantiles from a normal distribution of the same sample size on the y axis. If the data are from a perfectly normal distribution, the data will lie on a diagonal straight line. Departures from the diagonal indicate deviations from a normal distribution. Skewed distributions show up nicely as deviations from the line at the tails.

Four-in-one plots: The four plots described above can be depicted together in a single 4-part plot for each variable. The four-in-one plot is generated by typing:

```
uv.plots(spedata)
```

EVALUATE THE NEED FOR DATA TRANSFORMATION

Once you have thoroughly examined your data, you may deem it necessary or useful to transform your data. Ecological data are commonly skewed and/or range over several orders of magnitude, and as such can benefit from a transformation, such as the log or square-root transformations that compress large values. For community data sets involving species abundances, it is sometimes useful or more meaningful to transform the data to binary (presence/absence) data. Use the **`data.trans()`** function to transform variables using the log, power or arcsine square-root transformations.

Log (base 10) transformation: To log-transform the species abundances in the MAHA fish data set (spedata), type:

```
data.trans(spedata,method='log')
```

Examine the paired histograms comparing the raw (untransformed) and transformed distributions to see the effect of the transformation.

Power transformation (including binary transformation): To power-transform the species abundance data, try a squareroot transformation, by typing:

```
data.trans(spedata,method='power',exp=.5)
```

Note that the squareroot transformation is simply a special case of the power transformation where the exponent is equal to 0.5.

To transform the species abundances into presence/absence (i.e., binary transformation), use the power method with an exponent equal to zero, by typing:

```
data.trans(spedata,method='power',exp=0)
```

Arcsine square-root transformation: For proportional data (i.e., ranges 0-1), the arcsine square-root transformation is often recommended by statisticians. To accomplish this, type:

```
data.trans(spedata,method='asin')
```

Note: for any of these transformations, it is common to apply the same transformation to an entire set of related variables so that they are all on the same scale. For example, if some of the species variables benefit from a log transformation, it would generally be preferably to log-transform all species variables, not just the ones in need.

Once you have decided to use a particular transformation, you will need to save the transformed data into a new object by assigning the results, or you can use the 'outfile' argument in the **`data.trans()`** function to automatically save the transformed data set to a new permanent data set. You could also take a simple approach by using one of the many standard operations available in R. For example, if you decide on a natural log transformation, then type:

```
tradata <- data.trans(spedata,method='log',plot=F) or

tradata <- log10(spedata+1)
```

You will note that we included "+1" because the matrix contains many zeros and therefore performing a simple log transformation is not possible. The **`data.trans`** function automatically adds "1" to all cells to account for zero values.

EVALUATE THE NEED FOR DATA STANDARDIZATION

In many ecological data sets, especially community data sets involving species abundances, it is often quite useful to standardize (or relativize) the data before conducting subsequent analyses. Data "standardization" involves adjusting a data value relative to a specified standard derived from the corresponding row (sample) or column (variable) of the data frame. Keep in mind that standardizations can fundamentally alter the patterns in the data and can make the difference "between illusion and insight, fog and clarity" (McCune and Grace 2002).

If you are working with community data involving species abundances, such as in the MAHA data set, check the coefficient of variation (cv) in row and column totals for the species variables (see **`sum.stats()`** described previously). For the column summary, this is the cv in species' total abundances (i.e. the cv in column totals). The corresponding value in the row summary is the cv in site totals (i.e. the cv in row totals). If these values are small, say <50, it is unlikely that standardization will accomplish much. However, if these values are large, say >100, then it is likely that standardization will have a large effect on the results.

If you are working with environmental data involving variables measured in different units or on different scales, consider column standardization (by norm or standard deviates).

There are many possible row and column standardizations and they will not be reviewed here. Refer to the BIOSTATS help file for **`data.stand()`** for details on the available standardizations. Before applying any standardization, be sure to understand what the standardization does and whether the units for all the included variables are the same. Note: in some cases, standardization is built into the

subsequent analyses and therefore unnecessary – but to know this requires that you already understand the mechanics of the techniques you intend to use (which we haven't gotten to yet!).

At this point, you might simply explore what various standardizations do to the data so that you are ready and able to standardize your data as needed when you decide on a particular statistical procedure.

As an example, to conduct a 'row normalization' (i.e., to rescale each row so that the sums equal 1, or in other words to calculate relative species abundances), type the following:

```
data.stand(spedata,method='total',margin='row')
```

Examine the paired histograms that are produced by default to see the effect of the standardization on the distribution of each variable. Remember, if you intend to save the standardized results for use in a later analysis, you must save the results to a permanent data set. You can either assign the results to an object and then write that object to a file using the write.table() or write.csv() functions, or you can simply save the results directly with the data.stand() function by using the outfile='' argument.

Regardless of your decision to standardize or not, you should state your decision and justify briefly on ecological grounds.

EVALUATE THE DATA SET FOR OUTLIERS

Check your data set for potential outliers and eliminate them if justified. However, as a general rule, observations should not be dropped automatically just because they have extreme values. It is one thing to identify extreme observations that have high leverage on the results, but it is another thing altogether to delete these observations from the data set just because they have high leverage. What constitutes a true "outlier" depends on the question being asked and the analysis being conducted.

There is no general rule for deciding whether extreme observations should be considered "outliers" and deleted from the data set before proceeding with the analysis. Nevertheless, it is important to have an understanding of the number and pattern of extreme observations in order to gauge the robustness of the results. A good practice is to repeat the analyses with and without the suspect points and determine if the results are sensitive or robust to their inclusion. If the results are sensitive to the inclusion of these high-leverage points, you should probably carefully consider whether those points represent a meaningful ecological condition, and act on them accordingly.

There are several ways to identify extreme values, including both univariate and multivariate methods. It is always a good idea to begin with a univariate inspection. The uv.outliers() function computes the standardized values for each variable (i.e., z-scores) and returns a data frame with a list of samples and the variables that are greater than a specified number of standard deviations from the mean (default = 3). Let's do this for the MAHA environment data set. Try typing:

```
uv.outliers(envdata, id='Sinuosity:BasinAre', var='Elev', sd.limit=1)
```

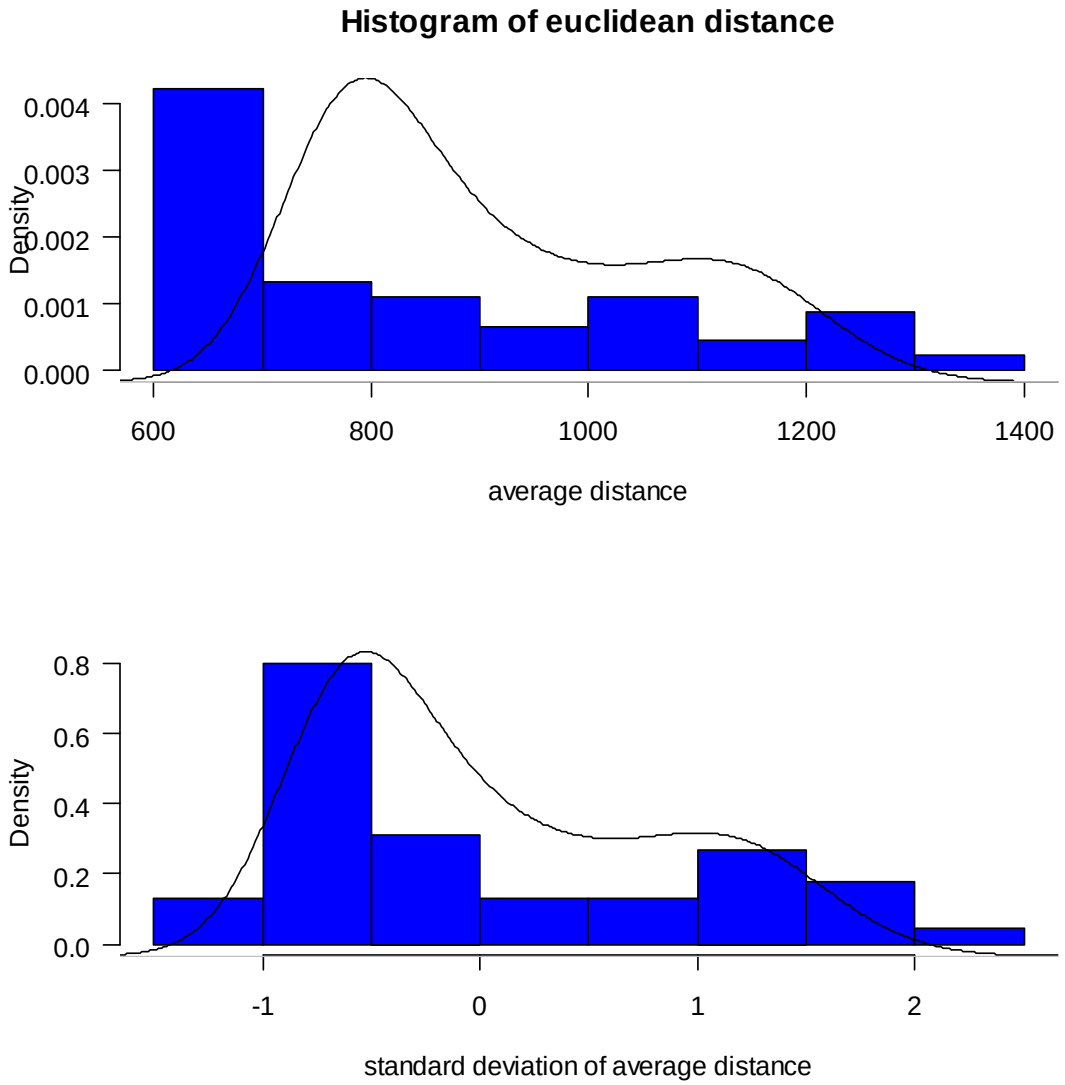
Note the number and distribution of extreme values based on 1 standard deviation. The default is sd.limit =3 based on the notion that 99.7% of all observations should fall within the mean +/- 3 SD according to a normal distribution. Are there observations with extreme values on several variables, or are the extreme values dispersed among samples and variables?

In the context of a multivariate data set, just because an observation is extreme on a single variable doesn't mean it is going to be a multivariate outlier. More importantly, an observation may not be a univariate outlier and yet still be an outlier when two or more variables are considered jointly. Thus, it is perhaps more instructive to determine if each observation is extreme in multivariate space.

To do this, use the **`multivar.outliers()`** function. Specifically, select an appropriate distance measure – ideally the distance measure to be used in subsequent analyses – and use this function to compute the average distance of each sample to all other samples. For example, for the MAHA environment data (envdata), try typing:

```
mv.outliers(envdata,method='euclidean', sd.limit=1)
```

This will produce a list of samples with an average Euclidean distance >1 standard deviations from the mean of average distances, in addition to paired histograms depicting the distribution of average distances and the distribution of standard deviates. Again, using the default of SD=3 is recommended (SD=1 is used here merely for illustrative purposes).



Another way of assessing multivariate outliers is to compute the Mahalanobis distance between each sample and the group of all other samples and to compare this against the expected distribution of Mahalanobis distances for a multivariate normal distribution, using a very conservative probability,

e.g., p<0.001 based on a chi-square distribution with degrees of freedom equal to the number of variables. Try typing:

```
mv.outliers(envdata, method='mahalanobis', sd.limit=1)
```

OPTIONAL READINGS (good ones in italics)

Hagaman, R.M. and M.E. Morbeck. 1984. Data transformations in multivariate morphometric analyses. Journal of Human Evolution 13: 225-245

Hinch, S. G., and K. M. Somers. 1987. An experimental evaluation of the effect of data centering, data standardization, and outlying observations on principal components analysis. Coenoses 2:19–23.

Jackson, D. A. 1993. Multivariate analysis of benthic invertebrate communities: the implication of choosing particular data standardizations, measures of association, and ordination methods. Hydrobiologia 268:9–26.

Kenkel, NC 2006. On selecting an appropriate multivariate analysis. Canadian Journal of Plant Sciences 86: 663-676.

Legendre, P. and E.D. Gallagher. 2001. Ecologically meaningful transformations for ordination of species data. Oecologia 129: 271-280.

Noy-Meir, I. 1973. Data transformations in ecological ordination. I. Some advantages of non-centering. Journal of Ecology 61:329–341.

Noy-Meir, I., D. Walker, and W. T. Williams. 1975. Data transformations in ecological ordination. II. On the meaning of data standardization. Journal of Ecology 63:779–800.

Cao, Y., D.D. Williams, and N.E. Williams. 1999. Data transformation and standardization in the multivariate analysis of river water quality. Ecological Applications 9: 669-677.

EXERCISE

Purpose

One of the first steps in multivariate analysis is data screening using numerical and graphical approaches. Why? It (1) suggests a plausible model for the data, (2) assesses validity of model assumptions, (3) detects outliers, and (4) suggests possible data transformations and standardizations. Many multivariate methods assume that the data are multivariate normally distributed. It this exercise you will exploring your own data using the techniques described above.

Tasks

1. Produce summary statistics for each variable in your dataset(s).
    a. Are there missing values, and if so how do you plan on accounting for this problem?
2. Examine the distributional properties of your variables.
    a. Do outliers exist? What statistical and ecological criteria are you using to base your decision?
    b. Is there a need for data transformation? What transformations are you considering and why? Again, is your decision based on statistical or ecological criteria, or both?
    c. Is there is a need for data standardization? What standardizations are you considering and why?

3. If transformations and/or standardizations are appropriate, then produce histograms for each of the transformed variables (the same way you did for the raw data).
    a. How effective was your transformation/standardization?
    b. Will you use the raw or transformed/standardized data in subsequent analyses? Why?