Ordination and Display of Multivariate Data
Edgar Barry Moser, Dept. Experimental Statistics, Louisiana State University, Baton Rouge, LA

Abstract
Graphical displays of multivariate data are a standard part of multivariate analysis and modeling. Multivariate ordinations and dimension reduction techniques permit low-dimensional views into higher dimensional space giving insight into model fitting and interpretation, and to the presentation of data and results. Minimum-spanning trees, biplots, and other devices further expand these capabilities. The PRINCOMP and CORRESP procedures of SAS/STAT®, and SAS/IML® and SAS/GRAPH® software, enable analysis and display of continuous and categorical multivariate data. Examples using real data are used to illustrate these techniques.

Introduction
Ordination is the typically 1, 2, or 3-dimensional ordering or display derived from a multidimensional set of data. Usually 2 or 3 dimensions are considered and the data are presented in a scatter-plot. These ordinations may then be correlated with auxiliary information on the observations or may be used in subsequent modelling. Some of the objectives of ordination include 1) a descriptive summary of the data, i.e., dimension reduction particularly for plotting, 2) a search for trends or relationships which may be useful in modelling or prediction, and 3) clustering or classification, as an ordering may be very useful for dividing the data into groups.

To order the observations we need a scale on which to measure the similarity or dissimilarity of pairs of observations and also some criterion by which the scale is used to produce an ordering. For ordinal, interval, or ratio scaled data, some measure of distance between observations may be used to determine how similar 2 observations are. Points with similar values on each variable should be near in distance, while those with very dissimilar values should be far apart. The usual definition of a distance metric is that it must obey 3 basic properties (see e.g., Borg and Lingoes, 1987):

1) \( d_{ij} \geq 0 \), distances are non-negative and \( d_{ii} = 0 \),
2) \( d_{ij} = d_{ji} \), distances are symmetric, and
3) \( d_{ij} \leq d_{ik} + d_{kj} \), the triangle inequality of distances.

Various measures of distance have been constructed with objectives of mathematical attractiveness and the proper measurement and resolution of similarity (Table 1). The city-block or Manhattan metric can be useful for dealing with ordinal and interval scaled data while various orders of the Minkowski metric and often the Mahalanobis metric have been used with ratio-scaled data, the latter particularly in hypothesis testing. Notice that the chi-square metric standardizes each variable to the same scale while the Mahalanobis metric further transforms the variables using information about their correlations. The chi-square metric is also commonly used for contingency tables as a measure of distance of each cell from its expected value.

For nominally scaled data, similarity measures may be used from which “distances” may be derived in the ordination process. As distances posses certain properties, similarities do as well:

1) \( 0 \leq s_{ij} \leq 1 \), similarities are between 0 and 1, inclusive,
2) \( s_{ii} = 1 \), an object is most similar to itself, and
3) \( s_{ij} = s_{ji} \), similarities are symmetric.

Note that the correlation coefficient, \( r \), does not fit this definition of

| TABLE 1. Distance measures commonly used in statistics and ordination. |
|-----------------------------|-----------------------------|
| Minkowski                   | \( d_{ij} = \left[ \sum_{k=1}^{p} (y_{ik} - y_{jk})^r \right]^{1/r} \) |
| City Block or Manhattan      | \( d_{ij} = \sum_{k=1}^{p} |y_{ik} - y_{jk}| \) |
| (Minkowski with \( r=1 \))   | \( d_{ij} = \sum_{k=1}^{p} (y_{ik} - y_{jk})^2 \) |
| EUclidean                    | \( d_{ij} = \left[ \sum_{k=1}^{p} \frac{(y_{ik} - y_{jk})^2}{\sigma_k^2} \right]^{1/2} \) |
| (Minkowski with \( r=2 \))   | \( d_{ij} = \left[ \sum_{k=1}^{p} \frac{(Y_i - Y_j)^2}{\Sigma_k Y_i} \right]^{1/2} \) |

| TABLE 2. Similarity measures for presence-absence data. |
|--------------|----------------|
| Let the cells of the contingency table be represented by \( a, b, c, \) and \( d \), where \( a \) is the number of joint presences and \( d \) is the number of joint absences. |
| Sokal Matching Coefficient | \( \frac{a + d}{a + b + c + d} \) |
| Jaccard Index | \( \frac{a}{a + b + c} \) |
| Dice Index | \( \frac{2a}{2a + b + c} \) |
| Ochiai Index | \( \frac{a}{\sqrt{(a + b)(a + c)}} \) |

a similarity measure, although \( r^2 \) may fit the definition depending upon one’s interpretation of \( r = 1 \) and \( r = -1 \).

In ecology or the behavioral sciences, the presence or absence of characteristics are often measured on each sampling unit or subject and similarities among units or subjects are constructed from these data. Various similarity measures for dichotomous outcomes have been devised and derive from the association in the contingency table of the 2 sets of presence-absences. Let the cross-classification of characteristics be represented in the following manner:

<table>
<thead>
<tr>
<th>Subject A</th>
<th>Presence</th>
<th>Absence</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>d</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Then the cell given as \( a \) contains the number of characteristics found present on both subjects, \( d \) contains the number absent on both, and \( b \) and \( c \) are the number of present on one, absent on the other subject. Some commonly used similarity measures for presence-absence data include Sokal's Matching Coefficient and Jaccard's Index (Table 2; see also Orlóci, 1978; Ludwig and Reynolds, 1986). There are many different measures of similarity because joint presence or joint absence of a characteristic may not imply similarity or dissimilarity between 2 observations. The distance between the 2 plots is similar, while their joint absence on 2 other plots probably implies nothing about those plots' similarity or dissimilarity.

It is always possible to compute a similarity measure from distances, however distances cannot always be computed from similarities. It is often desirable, however, to compute distances from similarities for the display of the similarity data. A commonly used conversion is \( d_{ij} = \sqrt{2(1-s_{ij})} \) (see Johnson and Wichern, 1988:551).

**Principal components analysis**

Principal components analysis (PCA) is a commonly used technique for ordinating and displaying continuous multivariate data. The technique finds an orthogonal basis for the data in such a way that the first axis is along the direction of greatest variation of the data and subsequent axes maximize variance explained given that they are orthogonal to previous axes. Since differences among observations generate variability in the data, a technique that summarizes the data according to variability should be informative and will often provide for an ordering of the data. Because of this property the technique is also often used to visually detect multivariate outliers. The PCA operates on the covariance matrix of the data and preserves the Euclidean distance among observations. The data may or may not be centered and/or scaled, leading to 4 types of PCA. The usual analysis operates either on the correlation matrix, which is a covariance matrix of centered and scaled (standardized) variables and is very useful when the units of measurement and scales of variables are different, or on the covariance matrix of centered variables. The additional 2 methods involve not centering the data prior to the computation of the "covariances". Sometimes these displays more clearly reveal the structure in the data as the first dimension will be summarizing variability of the data from the origin and for positive variables this is usually a good direction for ordering the data. Additional variations of the technique involve differential weighting or standardizing of the variables so that specific variables become more important in the analysis, since they would have greater variability than other variables and thus, exert more influence in determining the PCA solution.

In addition, the PCA can be an R-mode analysis which operates on the correlation or covariance matrix of response variables or characteristics, or the PCA can be a Q-mode analysis which operates on the correlation or covariance matrix of subjects or plots. The R-mode analysis provides an ordering of the subjects or plots in the component space, while the Q-mode analysis provides an ordering of the responses or characteristics in the component space. Later the biplot will be shown to connect the 2 analyses.

Let \( X \) be the \( (n\times p) \) matrix of data for \( n \) subjects on which \( p \) responses have been recorded. Further, let this matrix be centered and scaled as desired prior to analysis. The R-mode PCA solution is to perform a spectral decomposition on the \( (pxp) \) matrix \( X'X \).

\[
X'X = \sum_{j=1}^{p} \lambda_j e_j e_j' = V D_j V' \\
(\text{EQ} 1)
\]

where \( e_j \) and \( \lambda_j \) are the eigenvectors and eigenvalues, respectively, of \( X'X \), and \( V_j = e_j'X \) are the principal components with \( \text{Var}(Y_j) = \lambda_j \). Also \( V \) is the matrix of eigenvectors and \( D_j \) is the diagonal matrix of the eigenvalues. The Q-mode PCA is performed on the \( (nn) \) matrix \( X'X \). The mathematical relationship between the 2 solutions will be deferred until discussion of the biplot.

A data set containing enrollment, degree program offerings, and graduation data in 1989 for 67 university institutions classified as research I or II programs will be examined using the PCA techniques. The first ordination is based on the sizes of the universities. The 6 variables included in this analysis included total enrollment, total fall full time equivalent enrollment, total degrees, and total degrees per graduate, plus the enrollment, total fall full time equivalent enrollment, total degrees, and total degrees per graduate, plus the enrollment.

**Minimum spanning tree**

A spanning tree is a collection of straight line segments that connects each point on the graph in such a way that there are no closed loops and that each point is connected to every other point either directly or indirectly through this collection of line segments. A minimum spanning tree (MST) is a spanning tree with segments connected in such a way that the sum of the lengths of the individual line segments is a minimum (Gower and Ross, 1969). If the MST is computed using the full dimensionality of the data, then when plotted in the 2-dimensional space it gives insight into the dimensions not directly represented in the plot. Points may be located very near in the 2-space plot but the MST may indicate that they are actually further apart than each is to another point or points. The MST is also related to single linkage cluster analysis and has been used to divide the data into groups or clusters. If there are few major branches in the MST then it may also be used to order the observations. A SAS macro was written to construct and plot the minimum spanning tree for a set of points (Listing 2).

In Figure 2, the minimum spanning tree shows some relationships that would not be seen by inspection of the 2-dimensional point configuration alone. For example, LSU is located on the path connecting the schools on the left of the plot with those on the far right, such as Minnesota. This is not evident in the point configuration. In this analysis the MST is generally oriented along component 1 which is related to school size, but also has 2 major branches separated in the second dimension according to the degree programs offered. This interpretation will become clear with the use of the biplot.

**Biplot**

The matrix \( X \) of data values can be re-expressed in terms of the singular value decomposition (SVD) of a matrix,

\[
X = U D V' \\
(\text{EQ} 2)
\]

where the columns of \( U_{(pxp)} \) are orthogonal, \( V_{(pxp)} \) is orthogonal, and \( D \) is diagonal \( (pxp) \) with elements \( \Gamma_1 \geq \Gamma_2 \geq \ldots \geq \Gamma_p \).
which are called the singular values of X. A PCA was shown to operate on the matrix X'X or XX' for R or Q-mode analysis, respectively, and again where X is typically centered and scaled as appropriate. Using the SVD from above for the R-mode PCA results in

$$X'X = (UD_rV')' (UD_rV') = VD_r^2 V'$$

where the diagonal elements of $D_r^2$ are the eigenvalues of $X'X$ and $V$ contains the associated eigenvectors. For the Q-mode analysis the eigenvalues are the same as those for the R-mode approach, with the exception of the zeros for differences in matrix rank and dimension, while the eigenvectors of the Q-mode approach are contained in $U$. Thus, the SVD provides a mechanism for simultaneously exploring both the rows and columns of the data matrix $X$ such that they have a common basis for graphical display along with the properties of the PCA approach. The biplot re-expresses the data as

$$X = GH'$$

where $G$ is (nxr) and $H$ is (pxr) and r is the matrix rank of $X$. Using the SVD, $G$ and $H$ can be computed as

$$G = UD_r^0$$

and

$$H = VD_r^{-a}$$

where $\alpha$ is usually taken as 0, 1/2, or 1 to provide particular optimality conditions in the plot (Gabriel, 1971; see Gower and Digby, 1981). The biplot then becomes a scatter plot of the n + p vectors of $G$ and $H$ in r-space. When r is larger than 2, a 2-dimensional approximation to the biplot is to select the first 2 components associated with the 2 largest eigenvalues. A SAS macro was written to compute the biplot analysis and to plot the results (Listing 5). Macros to annotate the graphic output with a minimum spanning tree(s), label names (Listing 3), and rays from the origin to the variable points (Listing 4) were also used. The biplot macro uses the SAS/IML language to perform the analysis but a biplot can also be performed using the PRINCOMP procedure and data steps.

The research institutions data are oriented again (with UCSF omitted) and a biplot in 2-space is constructed (Figure 3). In the biplot distances between a row (school) and a column (variable) point are not meaningful, whereas directions of the points from the origin are. Depending upon the optimality conditions specified, distances among row points or among column points can be statistically interpreted, angles between variable vectors may be interpreted in terms of the correlations among the variables, and lengths of the variable rays may be proportional to the variables' standard deviations. In this plot $\alpha = 1/2$ and so is a compromise between optimality criteria to improve the appearance of the plot. The vectors for log total enrollment (LTLALL), log total degrees awarded (LTDAA), and log doctoral programs (LNDOTC) closely parallel component 1 indicating that the ordination along this axis is primarily one of student population size, while log number of master's (LNMAS) and log number of bachelor's (LNBAAC) programs are more in the direction of axis 2 suggesting that it serves as a contrast in the institutions' number of bachelor and master degree program offerings. Note that the lengths of the variable vectors are similar suggesting similar contributions of each variable to the 2-dimensional ordination.

A second analysis of the institutions data was performed on the composition of degrees awarded. Here, interest was about the number of degrees awarded relative to the other degrees, rather than about the absolute number of awards. Thus the degree awards are transformed to proportions before analysis. To preserve the sum-to-1 constraint of the proportions within an institution, the proportions were first transformed by an additive log ratio transformation (Aitchison, 1986)

$$Y_i = \pi_i / \pi_p$$

for $i = 1, 2, ..., p - 1$.}

Principal coordinates analysis

Very often we are able to determine the similarities among observations, again these may be vegetation plots in a study area or various teaching methods, and we wish to depict these relationships graphically. Also, we may have data for which a Euclidean distance measure is inappropriate or not desirable and we wish to use an alternative distance measure but still graphically represent the data. Thus an objective would be to depict the relationships among the observations as defined by distances or similarities in a few dimensions such that the observed distances or similarities are preserved as much as possible. This objective falls under the scope of multidimensional scaling (MDS). Principal coordinates analysis (PCO) is a form of classical or metric multidimensional scaling as it uses the actual distances or similarities in constructing the point configuration (Gower, 1987). Nonmetric MDS uses the ordinal ranking of the distances or similarities and has been generalized for many uses (see Borg and Lingoes, 1987).

PCO operates on the double centered matrix $\alpha$ derived from the matrix of similarities $\Lambda$ as

$$\alpha_{ij} = A_{ij} - \overline{A_i} - \overline{A_j} + \overline{A}$$
where the dot subscript indicates a marginal mean. Then as in PCA the spectral decomposition is used to extract the eigenvalues and eigenvectors of \( \alpha \).

\[
\alpha = \Gamma \Lambda \Gamma^T \tag{EQ 9}
\]

such that the columns of \( Z = \Gamma \Lambda^{1/2} \) form the principal coordinates solution (see Pielou, 1984). As in PCA, the eigenvalues indicate the amount of variability in the data explained by each dimension. Again, a SAS macro using the SAS/IML language was written to do the PCO computations and plot the resulting coordinates for the points (Listing 7). PROC MDS in SAS/STAT can also be used to compute the PCO solution.

A city block distance metric was used to compute distances among the research institutions for the number of degrees awarded in each of 9 major areas (professional, natural science, social science, humanities, engineering, business, education, health sciences, and math and computer sciences). A SAS macro written in SAS/IML was used to compute these distances (Listing 6). The distances were then converted to similarities (see Johnson and Wichern, 1988:551) as

\[
s_{ij} = \frac{1}{1 + d_{ij}} \tag{EQ 10}
\]

after appropriate scaling and were subjected to a PCO analysis (Figure 7). Again, a minimum spanning tree was also constructed for the data using all of the principal coordinates axes. When these data are viewed in 3 dimensions and especially when rotated by utilities found in SAS/INSIGHT™ the point cloud appears as a spiral and thus its projection into 2 space gives the appearance of the horseshoe for the MST. Without the MST the associations among the points could have been easily misinterpreted and the capabilities of SAS/INSIGHT permit further study of the point cloud. Here the associations identified are that the larger institutions are ordinated on the right most end of the data swarm while the smaller institutions are located at the left most end of the data swarm. The SAS code to perform this analysis was

```sas
/* Compute the Distance Matrix */
DISTMAX(VARLIST=dapr dans dass dahm daeg dabs daed
      CITY=l,ONLYONE=O,Stdize=1000);
/* Perform the PCO Analysis */
PCO(Data=Dist,Out=PCO);
/* Plot the Results */
PrinPlot(Dsn=PCO,Dim=2,prefix=PCO);
```

Correspondence analysis

Correspondence analysis (CA) can be used as a technique to graphically represent the rows and columns of a two-way contingency table (see Greenacre, 1984). It has been used extensively in ecology (see Moser, 1969; Moser et al., 1990) where it is often referred to as reciprocal averaging (see Pielou, 1984). Rather than operate on the covariance matrix of the data as in PCA, CA operates on the \( \chi^2 \) deviations of the cells from the centroid. The rows (columns) of the table can be viewed as points with coordinate values given by the columns (rows) of the table. The rows (columns) are divided by their marginal total to form profiles and the row (column) total is taken as the weight for the profile. CA then identifies an optimal subspace for representing both the weighted row and column profiles.

First the \((I \times J)\) contingency table \( N \) is converted to proportions \( P \) as \( P = (1/N)N \), where \( t \) is \( I \times N \) is the table total and \( I \) is a vector of ones. Then the \((I \times J)\) matrix \( Q \) is constructed as

\[
Q = D_r^{-1/2} (P - rc') D_c^{-1/2} \tag{EQ 11}
\]

where \( Q \) contains the square roots of the \( \chi^2 \) deviations of the row and column proportions under the assumption of row and column independence (or row or column homogeneity), and where \( r = P1 \) and \( c = P^T1 \) are row and column proportion totals, and \( D_r \) and \( D_c \) are diagonal matrices of the same \( r \) and \( c \) values.

As in PCA, the optimality criterion is to extract an orthogonal set of axes from the variation matrix (here \( Q \)) such that the first axis explains the greatest amount of variability possible, the second explains the greatest amount possible given that it is orthogonal to the first, and so on. Notice here, however, the summary is of \( \chi^2 \) variation.

The SAS/STAT procedure CORRESP can be used to accomplish the numerical analysis and construction of the CA axes. This procedure can also give information concerning the fit of the data to the CA solution including each points' contribution and fit in each dimension.

System-wide high school algebra I grades were reported for the parish of East Baton Rouge for the second semester of the 1987-1988 school year (Table 3; Lambert, 1988). A correspondence analysis and biplot of these data can illustrate the relationships between the grade distribution and the distribution of a particular grade across schools. Note that the CA preserves the constraint that the proportion of students in each grade category at each school must sum to 1.0. A PCA approach does not include such a constraint and assumes that the categories can act independently of one another. SAS program code to perform the correspondence analysis would be similar to

```sas
/* Compute the correspondence analysis on the summarized data with the variable COUNT containing the number of values per table cell. */
Proc Corresp Data=Algebra Observed Short
FreqOut Profile=both Out=Coords Dims=4;
Tables School,Grade;
Weight Count;
Run;

/* Now set up the plot with the desired annotation information. */
Data ORS VAR;
Set Coords;
If _Type_='OBS' then Output OBS;
Else If _Type_='VAR' then Output VAR;
Run;

MST(Data=Obs,Out=MST,Var=dim2 Dim3 Dim4);
```


<table>
<thead>
<tr>
<th>High School</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baker</td>
<td>29</td>
<td>25</td>
<td>60</td>
<td>107</td>
<td>170</td>
</tr>
<tr>
<td>BR Magnet</td>
<td>13</td>
<td>35</td>
<td>91</td>
<td>34</td>
<td>12</td>
</tr>
<tr>
<td>Belaire</td>
<td>15</td>
<td>34</td>
<td>55</td>
<td>147</td>
<td>169</td>
</tr>
<tr>
<td>Broadmoor</td>
<td>6</td>
<td>24</td>
<td>32</td>
<td>53</td>
<td>27</td>
</tr>
<tr>
<td>Capitol</td>
<td>20</td>
<td>31</td>
<td>42</td>
<td>61</td>
<td>37</td>
</tr>
<tr>
<td>Central</td>
<td>17</td>
<td>25</td>
<td>77</td>
<td>114</td>
<td>114</td>
</tr>
<tr>
<td>Glen Oaks</td>
<td>25</td>
<td>42</td>
<td>72</td>
<td>103</td>
<td>154</td>
</tr>
<tr>
<td>Istrouma</td>
<td>15</td>
<td>14</td>
<td>38</td>
<td>95</td>
<td>110</td>
</tr>
<tr>
<td>McKinley</td>
<td>2</td>
<td>8</td>
<td>39</td>
<td>71</td>
<td>45</td>
</tr>
<tr>
<td>Northdale Magnet</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>Northeast</td>
<td>3</td>
<td>10</td>
<td>17</td>
<td>36</td>
<td>32</td>
</tr>
<tr>
<td>Robert E. Lee</td>
<td>24</td>
<td>43</td>
<td>64</td>
<td>83</td>
<td>56</td>
</tr>
<tr>
<td>Scotlandville Magnet</td>
<td>30</td>
<td>05</td>
<td>66</td>
<td>31</td>
<td>11</td>
</tr>
<tr>
<td>Tara</td>
<td>12</td>
<td>32</td>
<td>68</td>
<td>78</td>
<td>168</td>
</tr>
<tr>
<td>Woodlawn</td>
<td>2</td>
<td>35</td>
<td>60</td>
<td>46</td>
<td>47</td>
</tr>
<tr>
<td>Zachary</td>
<td>39</td>
<td>33</td>
<td>63</td>
<td>81</td>
<td>94</td>
</tr>
</tbody>
</table>
The resulting ordination is generally with respect to increasing (or decreasing) grade scores, with schools such as Scotlandville having a large proportion of A and B grades (Figure 8).

Distortions and Non-linearities

In several of the plots the ordinations contained non-linear trends or shapes of the data as opposed to a nice linear ordering along the primary axis. This is not uncommon and can be the result of several things: (1) non-linearities in the data, (2) the data are represented in too few dimensions, (3) the distance function itself, and (4) the data might be inappropriate to the technique used. Note that a projection in too few dimensions is not a true distortion of the data but rather a misleading presentation of the data. Pieleu (1984;190-195) discusses these issues.

Pielou (1984:190-195) discusses these issues. Borg and Lingoes (1987:213-218) provide alternative explanations for observed non-linearities. The minimum spanning tree is a device that can be used to help reduce the interpretation problems of representing the ordination in too few dimensions. Other devices such as multi-way dynamic scatter plots with "brushing" capabilities and rotating scatter plots, all provided in SAS/INSIGHT software, should aid in the interpretation of complex projections of data.

particularly for presence-absence similarity measures. Both subjects moderately and widely separated along a gradient may have nothing in common and so would be scored as equally far apart or dissimilar. When the points are ordered, only the nearby subjects result in the directional ordering of points, and hence, the ordering must "bend" to represent both the nearby separation and the equality of moderate and far subjects to a particular point. This is very common in ecological applications where, for example, each plant species does not occupy the entire length of an environmental gradient being sampled, such as salinity, but rather has a limited range or optimum, and so may be absent from both ends of the gradient. Thus the gradient and distance measure induce a non-linearity in the representation of the data along the gradient.

Detrended correspondence analysis (Hill and Gauch, 1980) is a technique for linearizing the "horse-shoe" effect that commonly occurs in the correspondence analysis. Other ideas such as detrended FCA have been suggested, whereby one fits a quadratic linear model of component 2 on 1 to unfold the non-linear association (See Ludwig and Reynolds, 1988:257-274).

SAS, SAS/GRAzH, SAS/IML, SAS/STAT, and SAS/INSIGHT are registered trademarks or trademarks of SAS Institute Inc. in the USA or other countries. © indicates USA registration.

References


Lambert, M. 1988. More students fail algebra than any other subject in EBR. Sunday Advocate, Baton Rouge, Louisiana, June 12, 1988, pages 1B and 2B.


LISTING 1 A SAS macro to plot the ordination results using PROC PLOT and the SAS/GRAPH GPLOT procedures.

%AnLabel(data=VAR,Out=VLable,Position=2);
%Rays(data=VAR,Out=RAYS,X=Dirnl,Y=Dim2,Cl=Color);
Data Both;
Set OBS VAR;
Run;
Data Annotate;
Set MST RAYS VLabel;
Run;
Proc Gplot Data=Both Annotate=Annotate;
Plot Dim2=Dim2 _Type_ ;
Symbol CO=Red V=Dot 1=none F= RE;
Symbol CO=Yellow V=Star 1=none F= RE;
Run; Quit;

The resulting ordination is generally with respect to increasing (or decreasing) grade scores, with schools such as Scotlandville Magnet having a large proportion of A and B grades (Figure 8).
LISTING 2 A SAS macro to compute the minimum spanning tree for a set of points and to produce a SAS/GRAPH annotate data set that can annotate the tree onto a Gplot plot of the points.

* Macro: MstSAS -- Macro using SAS/IML to compute the minimum spanning tree for a set of points. Input to the macro includes the input data set containing coordinates, the variable names containing the coordinates, and the output data set containing the index and coordinates of the point to which each point will be joined. Alternatively, a SAS/GRAPH annotate data set can be constructed that contains the MOVE and DRAW commands to output data set containing the index and coordinates of the tree for a set of points. Requires an ID variable to set to label the points on a plot. Requires an ID variable to set to label the points on a plot.

LISTING 3 A SAS macro to annotate labels on a SAS/GRAPH Gplot graphic. Takes as input a SAS data set with a label variable and variables for the coordinates. Outputs a SAS/GRAPH annotate data set containing the labeling information.

%MACRO MstSAS(Data=LST, Output=OUTVAR, Var=Val, Summary=S, Annote=ANOT, LineType=1, LineSize=1, Color=White, HSys=2, XSys=2, YSys=2, ZSys=2, Position=Z, When=MeanX, MeanY, Si, Join=Simplex, Font=Sans-Serif);%If Position=Z Then %If Z=MeanX Then %Output OUT=OUTVAR, MeanX=MeanY, MeanZ=MeanX, Join=Si, Font=Simplex;%End;%If When<>"A" Then %Output OUT=OUTVAR, MeanX=MeanY, MeanZ=MeanX, Join=Si, Font=Simplex;%End;%Proc GPlot %Output OUT=OUTVAR, MeanX=MeanY, MeanZ=MeanX, Join=Si, Font=Simplex;%Run;%End;
LISTING 4  A SAS macro to annotate rays emanating from the origin to the specified points. Takes as input a data set with variables for coordinates and produces a SAS/GRAPH annotate data set for drawing the rays.

```sas
&=n; wr=ncol(G); run finish;
```

**LISTING 5**  A SAS macro to compute the biplot analysis of a rectangular matrix. It provides some flexibility in the choice of optimality criteria and optionally will plot and label the results.

```sas
proc iml;
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
NL 
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```

**PROCEDURE** MACRO computes G and H using alpha=.12

```sas
```
LISTING 6  A SAS macro to compute a distance matrix using either the Euclidean or the city block metric. Takes as input a data set and a list of variables from which the distances are computed. Optionally may be used to compute the distance from one point to all other points.

```sas
MACRO DISTHAT (VARLIST=, ID=, DATA=, NEWDATA=, CITY=, ONLYONE=, STDIZE=);
   PROC IML;
   USE &DATA;
   IF ONLYONE THEN
      N = NROW(DATAVARS);
      TEMP = J(N,1, 0);
      D = J(N, N, 0);
      DO I=1 TO N;
         IF I=1 THEN D[I,1]=D[I,1]; /* COMPUTE DISTANCE JUST TO FIRST.ONE */
      END;
      ELSE
         DO I=1 TO N-1;
            TEMP = ABS(DATAVARS[I,] - DATAVARS[J,]); /* I'TH AND J'TH ROWS */
            IF CITY THEN
               DIST[I,J] = 1/(1+TEMP);
               DIST[J,I] = DIST[I,J];
            ELSE
               DIST[I,J] = SQRT(TEMP); /* ASSUME SYMMETRIC DISTANCES */
               IF ONLYONE THEN D[I,J] = DIST[I,J];
            END;
         END;
         IF STDIZE > 0 THEN
            DIST = DIST-DIST[3]; /* SYMMETRIC */
      END;
   END;
   FINISH;
   QUIT;
   IF CITY THEN
      D = D + TEMP;
   END;
   Q=Q*(Q[i,j]+1)/(N*N);
   IF CITY THEN
      PRINT "Double centered Matrix w, alpha=", Q;
   CALL EIGEN(Q, L, E, ALPHA);
   L[n]=0;
   START:
      T = I:REPEAT(Q, 1, N);
      DO I=2 TO N;
         T[I] = T[I-1]+Q[I,]; /* SYMMETRIC */
      END;
   FINISH;
   TOT = T[N];
   PR = TOT/T;
   IF STDIZE THEN
      Q = Q*(Q[i,j]+1)/(N*N);
      PRINT "Principal Coordinates", Q;
   END;
   DATA &NEWDATA;
   SET &DATA;
   CALL SYMPUT("VARONE", &ID);
   IF NOT ONLYONE THEN
      DO I=1 TO N;
         CALL SYMPUT("VARLAST", &ID);
      END;
   END;
   RUN;
```

LISTING 7  A SAS macro to compute the principal coordinates analysis (PCA) using the SAS/IML language and optionally plot the results.

```sas
* PCO_MC -- Macro to Compute the Principal Coordinates; *
* solution given a similarity or distance matrix. *
* syntax: PCO(Data=LAST, Var=OUT=PCO, DimAtin=1, Summary=1, Complete=0); *
* $OUT="" When New Out=Data; *
* $Complete When Output SUMMARY=1; *
* proc iiml;
* reset log;
* use data;
* if "Out="" thenprint(read all var into A);
* else VSTR(read all var(Var) into A);
* close idata;
* ma; ncol=();
* start;
* if Summary Then print "Similarity Matrix", A,;
* aidot=A{.}J/n;
* adotj=A[+, J]/n;
* aadotdot=A[+, +]/(n*n);
* alpha=(A-repeat(avadotj, n, 1))-(repeat(avidot, n, 1)))+
* repeat(avadotj, n, n);
* if Complete Then print "Double centered matrix", alpha,;
* call eigen(1, alpha, l, e, alpha);
* l[n]=0;
* start:
* t=I:repeat(q, 1, n);
* do i=2 to n;
* t[i] = t[i-1]+q[i,]; /* SYMMETRIC */
* end;
* finish;
* run;
* if complete Then print "Eigenvalues and Cumulative Proportions",
* t[1],p; 0;
* end;
* if Summary Then print "Principal Co_ordinates", q,; /* 1 % prop */
* do i=1 to n;
* if [i,1]=0 then [i,1]=0;
* end;
* q=diag(q[1]);
* q=q[1];
* if complete Then
* print "Principal Co_ordinates", q,;
* reset log;
* n=ncol(q);
* index=char(n);
* names=names[1,:];
* if dimatin=1 then
* create_one_from_q("ColNames=names"); close_one;
* quit;
* data a4out;
* merge (data ,out); run;
```

1532
FIGURE 1. Scatter plot of research institutions with the first 2 principal components as axes. UCSF is noted as being far from the point cloud.

FIGURE 2. Scatter plot of research institutions in the principal component space with points connected by the minimum spanning tree.

FIGURE 3. Biplot of research institutions with schools identified as points and variables identified as rays from the origin.

FIGURE 4. Scatter plot of institutions in the principal component space of the centered and scaled log ratio variables for the numbers of degrees awarded. Points are connected by the minimum spanning tree.

FIGURE 5. Biplot of the centered and scaled log ratio variables for the numbers of degrees awarded, where points represent institutions and rays represent the log ratio variables.

FIGURE 6. Scatter plot of institutions in the principal components space of the uncentered but scaled log ratio variables for the numbers of degrees awarded. Points are connected by the minimum spanning tree.
FIGURE 7. Scatter plot of the institutions from the principal coordinates solution using the city block distance between institutions based upon the degrees awarded by major area variables.