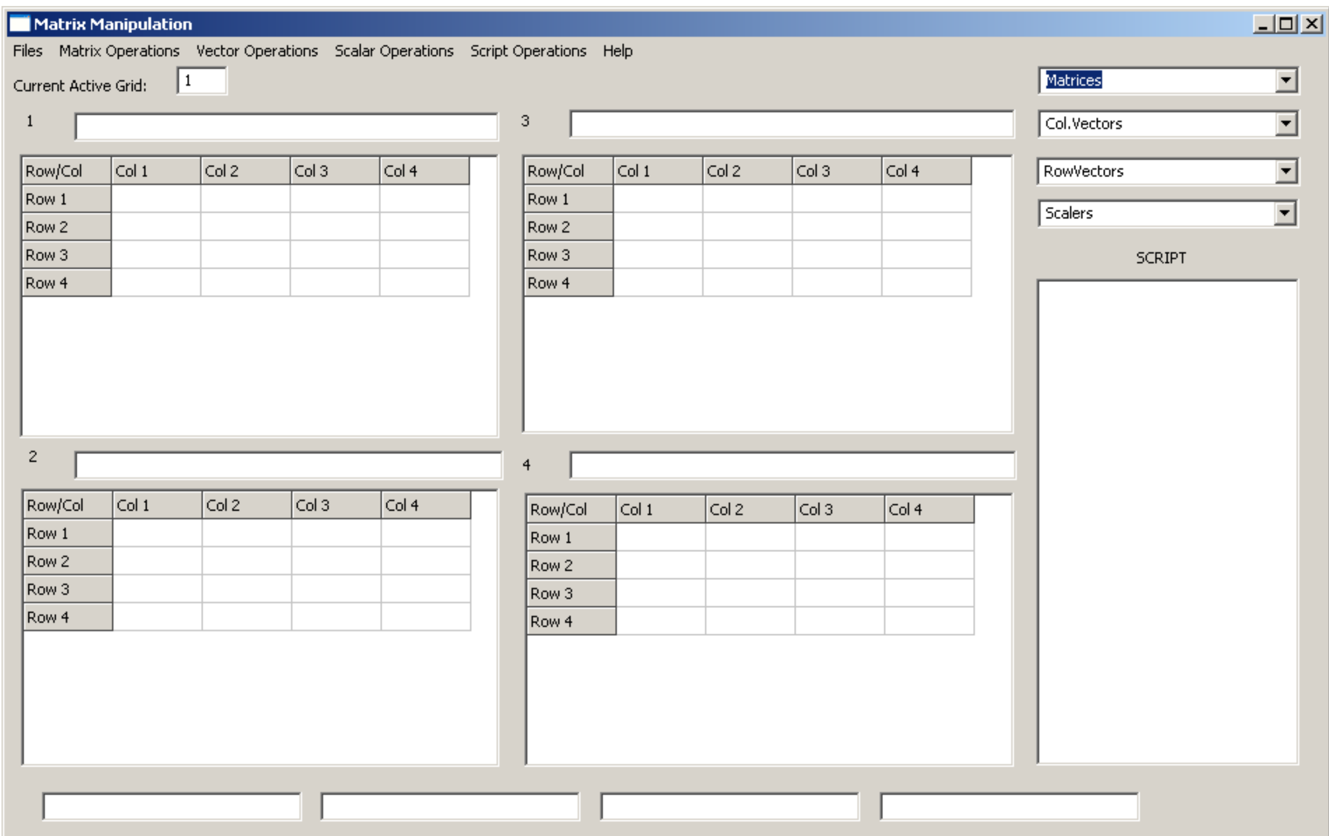
Purpose of MatMan

MatMan was written to provide a platform for performing common matrix and vector operations. It is designed to be helpful for the student learning matrix algebra and statistics as well as the researcher needing a tool for matrix manipulation. If you are already a user of the LazStats program, you can import files that you have saved with LazStats into a grid of MatMan.

Using MatMan

When you first start the MatMan program, you will see the main program form below{



This form displays four "grids" in which matrices, row or column vectors or scalars (single values) may be entered and saved. If a grid of data has already been saved, it can be retrieved into any one of the four grids. Once you have entered data into a grid, a number of operations can be performed depending on the type of data entered (matrix, vector or scalar.) Before performing an operation, you select the grid of data to analyze by clicking on the grid with the left mouse button. If the data in the selected grid is a matrix (file extension of .MAT) you can select any one of the matrix operations by clicking on the Matrix Operations "drop-down" menu at the top of the form. If the data is a row or column vector, select an operation option from the Vector Operations menu. If the data is a single value, select an operation from the Scalar Operations menu.

Scripts

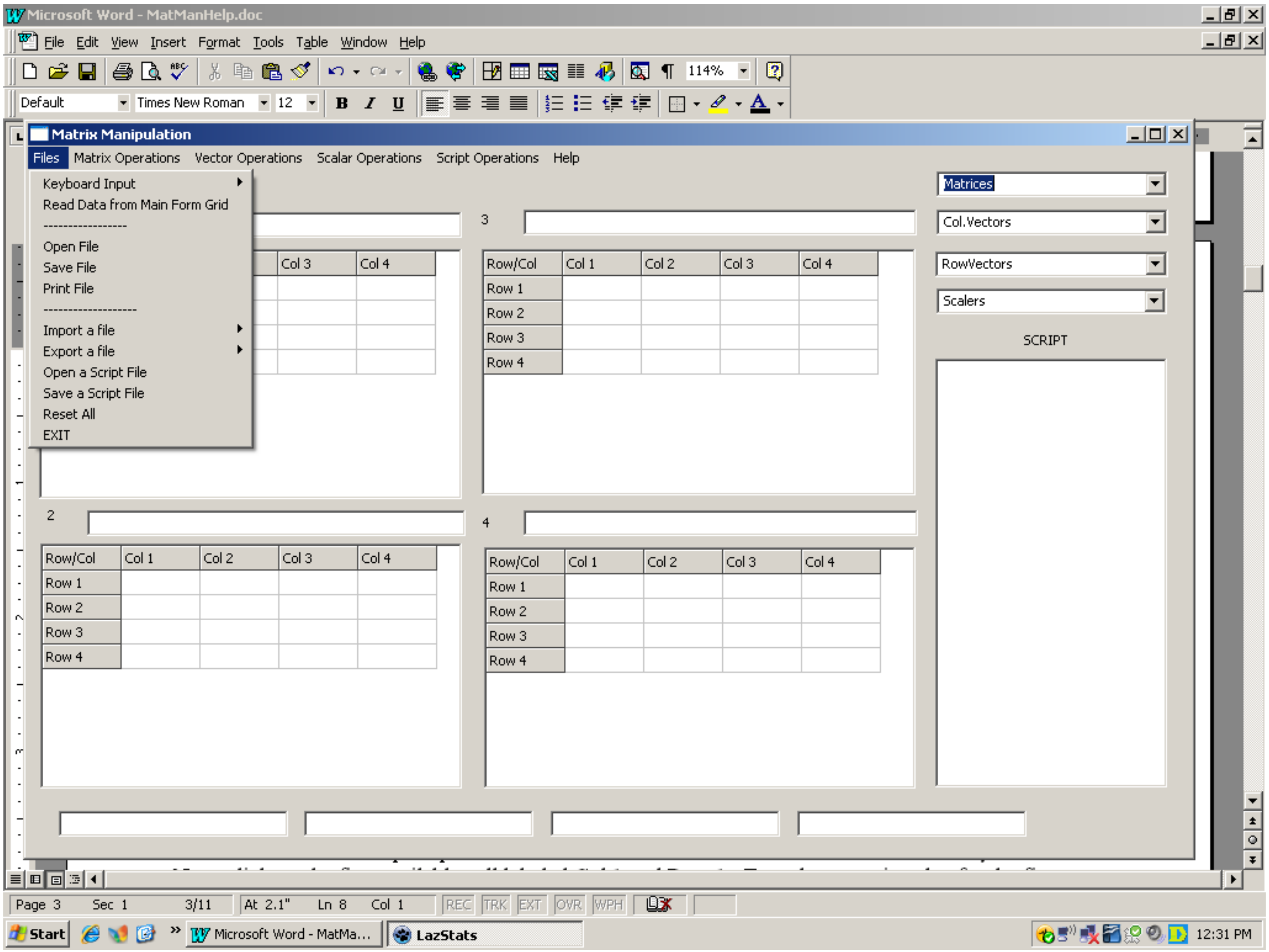
Each time an operation is performed on grid data, an entry is made in a "Script" list shown in the right-hand portion of the form. The operation may have one to three "operands" listed with it. For example, the operation of finding the eigenvalues and eigenvectors of a matrix will have an operation of SVDInverse followed by the name of the matrix being inverted, the name of the eigenvalues matrix and the name of the eigenvectors matrix. Each part of the script entry is preceded by a grid number followed by a hyphen (-). A colon separates the parts of the entry (:). Once a series of operations have been performed the script that is produced can be saved. Saved scripts can be loaded at a later time and re-executed as a group or each entry executed one at a time. Scripts can also be edited and re-saved. Shown below is an example script for obtaining multiple regression coefficients.

```
FileOpen:1-newcansas
1-ColAugment:newcansas:1-X
1-FileSave:1-X.MAT
1-MatTranspose:1-X:2-XT
2-FileSave:2-XT.MAT
2-PreMatxPostMat:2-XT:1-X:3-XTX
3-FileSave:3-XTX.MAT
3-SVDInverse:3-XTX.MAT:1-XTXINV
1-FileSave:1-XTXINV.MAT
FileOpen:1-XT.MAT
FileOpen:2-Y.CVE
1-PreMatxPostVec:1-XT.MAT:2-Y.CVE:3-XTY
3-FileSave:3-XTY.CVE
FileOpen:1-XTXINV.MAT
1-PreMatxPostVec:1-XTXINV.MAT:3-XTY:4-BETAS
4-FileSave:4-Bweights.CVE
```

Files

When MatMan is first started it searches the current directory of your disk for any matrices, column vectors, row vectors or scalars which have previously been saved. The current directory is the directory of data files identified in the Options menu of LazStats. The file names of each matrix, vector or scalar are entered into a drop-down list box corresponding to the type of data. These list boxes are located in the upper right portion of the main form. By first selecting one of the four grids with a click of the left mouse button and then clicking on one of the file names in a drop-down list, you can automatically load the file in the selected grid. Each time you save a grid of data with a new name, that file name is also added to the appropriate file list (Matrix, Column Vector, Row Vector or Scalar.)

At the top of the main form is a menu item labeled "Files". By clicking on the Files menu you will see a list of file options as shown in the picture below. In addition to saving or opening a file for a grid, you can also import a LazStats.LAZ file that has been opened or entered in the grid of the main form for LazStats.



Entering Grid Data

Grids are used to enter matrices, vectors or scalars. Select a grid for data by moving the mouse cursor to the one of the grids and click the left mouse button. Move your mouse to the Files menu at the top of the form and click it with the left mouse button. Bring your mouse down to the Keyboard Input option. For entry of a matrix of values, click on the Matrix option. You will then be asked to verify the grid for entry. Press return if the grid number shown is correct or enter a new grid number and press return. You will then be asked to enter the name of your matrix (or vector or scalar.) Enter a descriptive name but keep it fairly short. A default extension of .MAT will automatically be appended to matrix files, a .CVE will be appended to column vectors, a .RVE appended to row vectors and a .SCA appended to a scalar. You will then be prompted for the number of rows and the number of columns for your data. Next, click on the first available cell labeled Col.1 and Row 1. Type the numeric value for the first number of your data. Press the tab key to move to the next column in a row (if you have more than one column) and enter the next value. Each time you press the tab key you will be ready to enter a value in the next row cell of the grid. You can, of course, click on a particular cell to edit the value already entered or enter a new value. When you have entered the last data value, press the Enter key. A "Save" dialog box will appear with the name you previously chose. You can keep this name or enter a new name and click the OK button. If you later wish to edit values, load the saved file, make the changes desired and click on the Save option of the Files menu.

When a file is saved, an entry is made in the Script list indicating the action taken. If the file name is not already listed in one of the drop-down boxes (e.g. the matrix drop-down box), it will be added to that list.

Matrix Operations

Once a matrix of data has been entered into a grid you can elect to perform a number of matrix operations. The figure below illustrates the options under the Matrix Operations menu. Operations include:

Row Augment
Column Augment
Delete a Row
Delete a Column
Extract Col. Vector from Matrix
SVD Inverse
Tridiagonalize
Upper-Lower Decomposition
Diagonal to Vector
Determinant
Normalize Rows
Normalize Columns
Premultiply by : Row Vector; Matrix;Scaler
Postmultiply by : Column Vector; Matrix
Eigenvalues and Vectors
Transpose
Trace
Matrix A + Matrix B
Matrix A - Matrix B
Print

Printing

You may elect to print a matrix, vector, scalar or file. When you do, the output is placed on an "Output" form. On this form is a button labeled "Print" which, if clicked, will send the contents of the output form to the printer. Before printing this form, you may type in additional information, edit lines, cut and paste lines and in general edit the output to your liking. Edit operations are provided as icons at the top of the form. Note that you can also save the output to a disk file, load another output file and, in general, use the output form as a word processor. Show below is a typical output page for listing the inverse of a matrix.

SVDInverse.MAT From Grid Number 1

	Columns				
	Col.1	Col.2	Col.3	Col.4	Col.5
Rows					
1	5.656	-6.157	0.471	-0.730	-2.089
2	-6.157	9.075	-0.136	1.305	4.042

3	0.471	-0.136	1.187	0.029	-0.313
4	-0.730	1.305	0.029	2.172	-0.656
5	-2.089	4.042	-0.313	-0.656	4.929
6	1.881	-3.048	0.242	-0.567	-2.677

	Columns
	Col.6
Rows	
1	1.881
2	-3.048
3	0.242
4	-0.567
5	-2.677
6	2.923

Vector Operations

A number of vector operations may be performed on both row and column vectors. Shown below is the main form Vector Operations. The operations you may perform are:

Transpose
Multiply by Scalar
Square Root of Elements
Reciprocal of Elements
Print
Row Vec. x Col. Vec.
Col. Vec x Row Vec.

Scalar Operations

The operations available in the Scalar Operations menu are:

Square Root
Reciprocal
Scalar x Scalar
Print

Using the Combination Boxes

In the upper right portion of the MatMan main form, there are four "Combo Boxes". These boxes each contain a drop-down list of file names. The top box labeled "Matrix" contains the list of files containing matrices that have been created in the current disk directory and end with an extension of .MAT. The next two combo boxes contain similar lists of column or row vectors that have been created and are in the current disk directory. The last contains name of scalar files that have been saved in the current directory. These combo boxes provide documentation as to the names of current files already in use. In addition, they provide a "short-cut" method of opening a file and loading it into a selected grid.

Example Scripts:

Means
Students introduced to statistics are usually given the formula for the mean as the average of a set of scores or, in words, the sum of a set of values divided by the number of values. The formula may be written as:

$$\text{Mean} = \frac{\Sigma X}{N}$$

The above formula is then repeated for each variable for which a mean is sought. When we have a data matrix with multiple variables, the means can be obtained using matrix manipulation. As an example, let us assume we have a data matrix of 20 subjects (rows) and 6 variables (columns.) We will simply refer to that matrix as Matrix A. If we transpose this data matrix, the result will be a matrix of 6 rows and 20 columns. We will refer to this transpose matrix as AT. Next, we will create a column vector of 20 values, each value being 1.0. We will call this column vector simply Ones. If we now multiply our AT matrix times our column vectors of ones, we will obtain a column vector of 6 rows. Each element in that vector will be the sum, across the 20 cases, of the X values for a variable in our AT matrix. We save these results as the vector Asums. Now we must somehow divide each of those six sums by the number of cases. To do this we have several options. First, we could transpose our Ones column vector and obtain the OnesT transpose matrix. Multiplying this transpose times the original Ones matrix would yield a scalar of 20 since there were 20 1's in each vector. The resulting scalar can be saved as scalar N. The alternative method is to simply enter the value 20 from the keyboard and save it as the scalar N. The reciprocal of the scalar N can next be obtained and saved as the scalar RecipNSubjects. Multiply this scalar times our Asums vector and you have the AMeans vector!

Shown below is a script illustrating the above sequence of operations. We have imported a LazStats file with the label cansas.LAZ as our data file and saved it as the matrix A.

CURRENT LISTING FOR SCRIPT Means.SCP

```
1-FileOpen:1-cansas.MAT
1-FileSave:1-A.MAT
1-MatTranspose:1-A.MAT:2-AT
2-FileSave:2-AT.MAT
3-FileSave:3-Ones.CVE
2-PreMatxPostVec:2-AT:3-Ones.CVE:4-ASums
4-FileSave:4-ASums.CVE
1-FileSave:1-NSubjects.SCA
1-ScalerRecip:1-NSubjects.SCA:1-RecipNSubjects
1-FileSave:1-RecipNSubjects.SCA
4-ScalerxVector:1-RecipNSubjects:4-ASums:3-AMeans
3-FileSave:3-AMeans.CVE
```

Variances and Standard Deviations

The standard deviation of a variable is the positive root of the square root of the variable's variance. The formula for the (unbiased estimate of population) variance may be written:

$$S^2 = \frac{\Sigma X^2 / N - (\Sigma X)^2}{N - 1}$$

To obtain this result for a series of variables, we would repeat the above formula for each variable. You can see that we need to obtain the sums across the subjects (cases) for each variable, the sums of squared values across the subjects for each variable, the square of those sums for each variable and the number of subjects (N). Before we attempt to solve the above equation by matrix algebra, let us look at a more general equation for either the variance OR the covariance of two variables:

$$S^2_{XY} = \frac{\Sigma XY / N - (\Sigma XY)}{N - 1}$$

Notice that this is the same equation as before when $X_j = X_k$, that is, the two variables j and k refer to the same variable.

We introduced this formula for covariance since it is just as easy to obtain both the variances and the covariances in our matrix manipulation. Once we have the variance-covariance matrix, we note that the diagonal values of that matrix contains our desired variances for the variables. We can simply extract the diagonal and place it into a column vector. The standard deviations are then obtained by taking the square root of the elements of that vector of variances.

Cross-Products Matrix

Obtaining the Cross-Products matrix is quite simple with matrix algebra. The algebraic formula for the cross-product is:

where j and k represent any two variables and i represents the individual case. In matrix notation we can write:

[ATA] = [A]' [A]

If you start with a data matrix in which the rows correspond to subjects or objects measured, and the columns represent measurement observations, you have the basic matrix needed for the cross-products. Assume your data matrix is labeled "A". Obtain the transpose of that matrix by clicking on the transpose option under the Matrix Operations menu. Label the result AT. Now, if you multiply the transpose times the original matrix, you will obtain the cross-products. You might label this result ATA or ACP. The script produced may look like that below in which we used a file consisting of 20 cases with six variables (cansas.LAZ imported from LazStats:)

CURRENT LISTING FOR SCRIPT

```
1-FileOpen:1-A.MAT
1-MatTranspose:1-A.MAT:2-AT
```

2-FileSave:2-AT.MAT
2-PreMatxPostMat:2-AT:1-A.MAT:3-ATA
3-FileSave:3-ATA.MAT

Once you have obtained the above results, you might want to proceed with the calculation of the variance-covariance matrix and even the correlation matrix among the variables. See the script in the section Variance-Covariance Matrix and the Correlation Matrix.

Multiple Linear Regression Analysis

In multiple regression analysis, we want to obtain weights (regression coefficients) that when multiplied times original variable values and these weighted values are summed will yield a maximum linear relationship to another (dependent) variable. Using algebra we can write:

$$Y_i = B_1X_1 + B_2X_2 + \ldots + B_kX_k + B_0$$

where Y is the dependent variable observed for case i, the X's 1 through k are the independent variable values observed for case i, B's 1 through k are the coefficients to be obtained and B0 is the Y axis intercept (the line or plane which intersects the Y axis.) In obtaining the solution for the B weights, we use calculus to obtain derivatives that produce k equations to be solved simultaneously. Because the weights obtained are typically obtained from a sample of cases drawn at random from a population, we also will want to obtain the "standard errors" of these weights. In obtaining the solution for the weights, you can follow these steps:

1. Augment your original data matrix of independent variables with a column vector of 1's. This vector will provide the result for the intercept. Save the augmented matrix as, say, X.
1. Obtain the transpose of the matrix in 1 above. Save it with a label like XT.
1. Obtain the product of the transpose and the original matrix. Label the product as something like XTX.
1. Invert the product matrix obtained in 3 above. A label might read XTXINV.
1. Multiply the transpose matrix from 2 above times the column vector of independent variable values. A label to use could be XTY.

1. Multiply the inverse matrix from 4 above times the product matrix in 5 above. The result is a column vector of B weights (the last of which is the intercept.) Use a label like Bweights.

The script below illustrates obtaining the "least-squares" regression coefficients for a matrix imported from LazStats (cansas.LAZ.) After augmenting the imported file of independent variables (variables 1 through 5) it was saved as simply X. The Y values are obtained from the 6th variable of the original cansas.txt file that was imported.

CURRENT LISTING FOR SCRIPT Mreg.SCP

FileOpen:1-newcansas
1-ColAugment:newcansas:1-X
1-FileSave:1-X.MAT
1-MatTranspose:1-X:2-XT
2-FileSave:2-XT.MAT
2-PreMatxPostMat:2-XT:1-X:3-XTX

3-FileSave:3-XTX.MAT
3-SVDInverse:3-XTX.MAT:1-XTXINV
1-FileSave:1-XTXINV.MAT
FileOpen:1-XT.MAT
FileOpen:2-Y.CVE
1-PreMatxPostVec:1-XT.MAT:2-Y.CVE:3-XTY
3-FileSave:3-XTY.CVE
FileOpen:1-XTXINV.MAT
1-PreMatxPostVec:1-XTXINV.MAT:3-XTY:4-Bweights.CVE
4-FileSave:4-Bweights.CVE

Variance-Covariance Matrix

In the example below, we have used an imported file from LazStats named cansas.LAZ. It consists of 20 cases (subjects) each of which has been measured on 6 variables. This data matrix has been re-saved as matrix A. The transpose of matrix A has been saved as AT. In the example for obtaining the means for a set of variables (see Means) we obtain a vector labeled ASums. By multiplying this vector times its transpose, we can obtain the product of the sums. We saved this product in the matrix labeled: ASumxASumT. Multiplying this matrix by the scalar 1 / N yielded the matrix labeled ASumxASumTdivN. This is the second term of the numerator in the above formula. We also obtained the product of the transpose of our A data matrix with the data matrix A. This product was labeled ATA.MAT. From this matrix which represents the first term in the numerator of the above equation, we subtracted our previous result and saved the results as ADevCP.MAT. Multiplying this matrix which represents the numerator of the above equation (for all variables) by the reciprocal of our sample size N we obtained the variance-covariance matrix labeled AVarCov.Mat. We extracted the diagonal of this matrix (the variances) and placed them in the column vector labeled AVariances.CVE. Taking the square root of the elements of this vector gives the standard deviations for each variable (the AStdDevs.CVE vector.)

CURRENT LISTING FOR SCRIPT VarCovarMat.SCP

1-FileOpen:1-A.MAT
2-FileOpen:2-AT.MAT
2-PreMatxPostMat:2-AT.MAT:1-A.MAT:3-ATA
3-FileSave:3-ATA.MAT
1-FileOpen:1-ASums.CVE
1-VectorTranspose:1-ASums.CVE:2-ASumsT
2-FileSave:2-ASumsT.RVE
1-VecxVec:2-ASums.CVE:1-ASumsT::4-ASumxASumT
4-FileSave:4-ASumxASumT.MAT
1-FileOpen:1-OneOverN.SCA
4-ScalerxPostMat:1-OneOverN.SCA:4-ASumxASumT.MAT:2-ASumxASumTdivN
2-FileSave:2-ASumxASumTdivN.MAT
3-MatMinusMat:3-ATA.MAT:2-ASumxASumTdivN:1-ADevCP
1-FileSave:1-ADevCP.MAT
2-FileOpen:2-RecipNMinus1.SCA
2-ScalerxPostMat:2-RecipNMinus1.SCA:1-ADevCP:3-AVarCov
3-FileSave:3-AVarCov.MAT
3-DiagToVec:3-AVarCov:3-AVariances

4-FileSave:4-AVariances.CVE
4-sqrtvector:4-AVariances:4-AStdDevs

Principal Components Analysis

The principal components analysis is an attempt to transform a symmetric matrix into another matrix of the same size in which the columns are "orthogonal" (not correlated) with one another. If the original matrix represents the product-moment correlations among a set of variables, the product of the transformed matrix (with columns normalized) with its transpose should approximately reproduce the original correlation matrix. The process is rather easy. Simply obtain the roots and vectors (eigenvalues and eigenvectors) of the original matrix. Normalize the columns of the obtained matrix if the solution used does not already provide normalized vectors.

In addition to the transformation matrix (eigenvectors) you obtain the roots of the matrix. These roots may be interpreted as proportions of the total trace of the original matrix. If correlations are analyzed the diagonal values represent the (standardized) variance of the variables hence the trace of the matrix (sum of diagonal values) represents the total variance of the matrix. Typically, the solution of roots and vectors yields roots that are ordered in magnitude. You might observe that most of the matrix variance is "explained" by only a few roots (and corresponding vectors.)

Factor analysis frequently proceeds in a similar fashion although the diagonal values of the correlation matrix may be replaced by estimates of the "communality" that exists among the variables so that only common variance is reflected in the obtained transformation matrix. A common replacement for diagonal values is the squared multiple correlation of each variable with the remaining variables. In other factor analytic solutions, the variance-covariance matrix might be analyzed.

Shown below is the script for the principal component analysis of a set of data imported from LazStats (cansas.LAZ.) Correlations were obtained among the six variables and the roots and normalized vectors obtained (see Correlation Matrix.)

CURRENT LISTING FOR SCRIPT PrincComp.SCP

1-FileOpen:1-Acorrelations.MAT
1-MatrixRoots:1-Acorrelations.MAT:1-Aroots:2-Aeigenvectors
1-FileSave:1-Aroots.MAT
2-FileSave:2-Aeigenvectors.MAT

Standardized Scores

In the social sciences, most measurements are made with scales that do not have ratio scale properties, that is, that represent a scale of measurement where there is a "true" zero (absence of the attribute) and where the values such as 5 represents five times the amount of attribute that the value 1 represents. More commonly, the social scientist hopes to obtain a measurement scale that has interval scale properties. If these measurements are obtainable, the researcher may "standardize" the measurements observed for a group of subjects or cases. A common method for standardizing scores is to complete a linear transformation such that the resulting scores have a mean of zero and a standard deviation of one. The formula for converting a single observation to this new "z score" equivalent is:

$$Z = \frac{X - M}{S}$$

where the z is the transformed score, X is the original score, M is the mean of the scores and S is the standard deviation of the scores for the group.

We can obtain these transformed scores for an entire group of subjects and for each variable observed for those subjects using matrix algebra methods. The steps involved are:

1 Obtain the means of the variables. See Means.
2 Obtain the standard deviations of the variables. See Variances and Standard Deviations.
3 Create a row vector containing a 1 for each case.
4 Multiply the column vector of means times the row vector. The result will be a matrix with k variable rows and N cases columns.
5 Transpose the matrix in 4 above. The transpose is a N by k matrix containing means of the k variables in each row.
6 Subtract the matrix in 5 above from the original data matrix of X scores (an N by k matrix.) The results are "deviation" scores (X-M.)
7 Transpose the above matrix to obtain a k by N transpose matrix.
8 Multiply the transposed difference matrix obtained in 7 above by a diagonal matrix containing the reciprocal of standard deviations for the k variables (a k by k matrix.) The result will be a k by N matrix of standardized scores. You may want to transpose this matrix to obtain the N by k results in a matrix similar to the original matrix.

The above operations are summarized in the script below:

CURRENT LISTING FOR SCRIPT AStdScores.SCP

```
1-FileOpen:1-AMeans.CVE
2-FileSave:2-ARowVector.RVE
2-VecxVec:2-AMeans.CVE:1-ARowVector.RVE:3-Ameans
3-FileSave:3-Ameans.MAT
3-MatTranspose:3-Ameans.MAT:4-AmeansT
4-FileSave:4-AmeansT.MAT
1-FileOpen:1-A.MAT
2-MatMinusMat:1-A.MAT:2-AmeansT.MAT:3-ADeviations
3-FileSave:3-ADeviations.MAT
1-IDMAT:1-IDMAT
1-FileSave:1-IDMAT.MAT
2-FileOpen:2-AStdDev.CVE
2-VectorRecip:2-AStdDev.CVE:2-ASDRecips
2-FileSave:2-ASDRecips.CVE
2-VecToDiag:2-ASDRecips.CVE:1-C:\Projects\Delphi\MatMan\IDMAT.MAT
1-FileSave:1-ASDRecipMat.MAT
1-PreMatxPostMat:1-ASDRecipMat.MAT:3-ADeviations:4-AStdScores
4-FileSave:4-AStdScores.MAT
```

Correlation Matrix

The product-moment correlation between two variables X1 and X2 may be written:

$$R = \frac{C_{x_i x_j}}{S_i \; S_j}$$

where r is the correlation between two X variables, C is the covariance between two X variables and the S's are the standard deviations of the two variables.

To obtain the inter-correlations among variables we first obtain the covariance matrix [C] and pre and post multiply it with a diagonal matrix [D] in which the diagonal values represent the reciprocal of the standard deviations of each variable. The script below demonstrates the MatMan operations used to obtain the correlation matrix. The results from previous analyses are used in which a matrix A consists of variable measurements on 20 subjects (the cansas.LAZ data from LazStats.) See the sections Means, Variances and Standard Deviations and Variance-Covariance Matix.

CURRENT LISTING FOR SCRIPT CorrMat.SCP

```
1-IDMAT:1-IDMAT
1-FileSave:1-IDMAT.MAT
2-FileOpen:2-AStdDevs.CVE
2-VectorRecip:2-AStdDevs.CVE:2-ASDRecips
2-FileSave:2-ASDRecips.CVE
2-VecToDiag:2-ASDRecips.CVE:1-C:\Projects\Delphi\MatMan\IDMAT.MAT
1-FileSave:1-ASDRecipMat.MAT
2-FileOpen:2-AVarCov.MAT
2-PreMatxPostMat:1-ASDRecipMat.MAT:2-AVarCov.MAT:3-ASDRecipxVarCov
3-FileSave:3-ASDRecipxVarCov.MAT
3-PreMatxPostMat:3-ASDRecipxVarCov.MAT:1-ASDRecipMat.MAT:4-Acorrelations
4-FileSave:4-Acorrelations.MAT
```