THE MAGICAL MYSTERY MATRIX

Richard Q. Wendt, F.S.A.
Towers Perrin

Summary

The Choleski factor matrix has been well known for many years; it is a triangular matrix that satisfies the condition that the factor matrix multiplied by its transpose is equal to a symmetric matrix. The concept has been used since the early 1980's to generate asset returns with specified means and covariances. This paper develops some new uses of the factor matrix, including modeling serial correlation, adding asset classes with contingent returns, adjusting preliminary results to final targets and adjusting large, "unfactorable" matrices. Two appendices illustrate a numerical example and include listings of APL2/PC functions that implement the methodology.

I. INTRODUCTION

Given a symmetric, positive definite matrix, **M**, it is possible to determine an upper triangular factor matrix, **F**, such that

$$M=F^{T}\times F \qquad (1)$$

where the multiplication is the matrix inner product and F^{T} is the transpose of **F**. This mathematical definition plays a valuable role in the generation of data with specified covariances, an integral part of many simulation models.

Given an N row x M column matrix, **D**, containing N observations of M variables; if the mean of each column is zero, then the covariance matrix for **D** is defined by

$$COV(D)=D^{T}\times D \qquad (2)$$

Equations (1) and (2) lead to the following well known methodology of generating random data that satisfies specified mean and covariance requirements:

Given
- **C**, a specified covariance matrix for M variables
- **F**, the Choleski factor matrix of **C**
- **R**, an N x M matrix of random data, where each column follows a normal distribution with mean zero and the covariance matrix of **R** is the identity matrix

Then, let

$$R' = R \times F \qquad (3)$$

and it can be easily shown by matrix algebra that the covariance matrix of **R'** is **C**. By adding columnar means to **R'**, **R'** will be a data matrix with specified means and covariances.

In summary, the Choleski factor matrix can generate a data matrix with specified means and covariances, as long as

1. a normal random matrix, **R**, with a covariance matrix equal to the identity matrix can be supplied. This is generally done with a random number generator. A later section of this paper will show how to force the initial random numbers to exactly satisfy this requirement.

2. The specified covariance matrix is factorable. In simple terms, this means that the specified covariances and underlying correlations must be internally consistent. For example, if variable A is highly positively correlated to variables B and C, then variables B and C would be expected to also have a strong positive correlation. A specified covariance matrix with a strong negative correlation between variables B and C would not be factorable. For example, consider variables A, B and C, each with mean zero and unit standard deviation. If the specified correlation matrix for ABC is as follows:

	A	B	C
A	1.0	.9	.9
B	.9	1.0	-.9
C	.9	-.9	1.0

Then the covariance matrix would be identical to the correlation matrix and it is not factorable with the Choleski methodology. In order for the matrix to factor, the correlation between B and C must be about positive .65 or higher.

II. A SIMPLE ASSET SIMULATION MODEL

Some early asset simulation models that were popular in the early 1980's were based on specifying means and covariances for each asset class. As long as the specified covariances were

factorable, i.e., economically/statistically feasible, then the model could be developed in the following process:

Given M asset classes,
N simulations (scenarios) of each class in each year,
Y years for the simulation time horizon,
$R_1 \ldots R_Y$ N x M matrices of normally distributed random variables with mean zero and covariance matrix = I,
C a specified covariance matrix (M x M),
Then let F = the factor matrix of C
and $D_i = R_i \times F$, for i = 1...Y

Each D_i will satisfy the criteria for specified covariances; by adding the expected returns of each asset class to the appropriate columns, D_i will also have the specified mean returns for each asset class. By using simple variations of this methodology, both normal and lognormal distributions can be generated.

Chart One shows a visual example of the matrix calculation, where returns for three asset classes are generated.

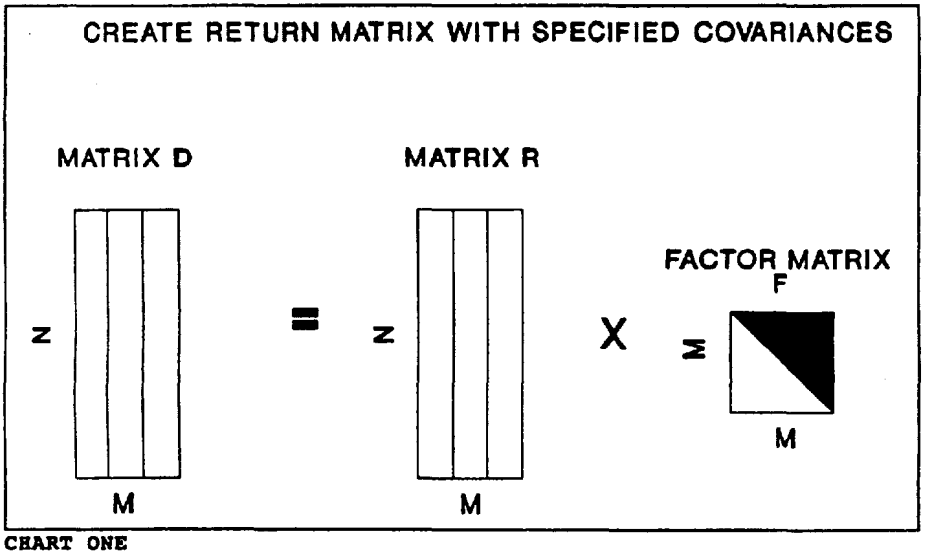

CHART ONE

The reader will note that although the simulated asset returns will exactly satisfy the criteria for means and covariances, each year of the simulation is independently determined and there is

no linkage between years. In recent years, developing economic theory has lead to a strong indications of linkages of asset returns between years.[1] More complex models are needed to develop linkage (i.e., serial correlation) between years; these will be explored in a later section of this paper.

III. ADJUSTING SIMULATED RESULTS TO TARGETS

In order to introduce linkages of asset returns across years, my firm has developed a sophisticated economic simulation model that first simulates yields and inflation and secondarily simulates asset returns based on the pattern of simulated yield and inflation. Since the asset class returns have a complex relationship with yields, inflation and other asset class returns, there is only indirect control over the simulated means and covariances of the asset class returns.

Where it is important to adjust the asset returns to meet specified targets, there are two possible solutions:

> The first approach is to "tweak" model parameters in an attempt to meet the targets. Even with expert knowledge of the model, this can be a difficult and time consuming process.
>
> The second, and more elegant, approach is to apply a matrix "filter" to force the simulated results to hit the specified targets for means and covariances.

As a simple one dimensional example, if V, a vector of length N, holds randomly simulated data with mean m and standard deviation s and if the target mean is M and the target standard deviation is Σ, then the following equation adjusts V to hit the targets:

$$V' = \frac{(V-m)}{s} \times \Sigma + M \qquad (4)$$

The adjustment for an N x M data matrix is analogous to the one dimensional example, but uses matrix calculations to control the covariances. Fortunately, the magical factor matrix can do the job.

---

[1] For example, see Poterba, James M. and Summers, Lawrence H., 1988, "Mean Reversion in Stock Prices -- Evidence and Implications, "Journal of Financial Economics 22, 27-59.

Let D = an N x M matrix of simulated data
M_D an N x M matrix, where each column contains the mean of the equivalent column of D
M'_D an N x M matrix of the target means for each variable
C covariance matrix of D,

$$C=(D-M_D)^T\times(D-M_D) \tag{5}$$

C′ target covariance matrix
F Choleski factor matrix of C
F′ Choleski factor matrix of C′

Then,

$$D'=M'_D+(D-M_D)\times(F'\times F^{-1}) \tag{6}$$

D′ will have the specified means and covariances and will tend to preserve the initial data patterns for small differences between C and C′.

One caveat is that the matrix filter works best for small changes in C. In the case where the targets are greatly different from the simulated covariances, there is first a question as to whether C′ will be factorable. If C′ is factorable, the means/covariances will be as desired, but other characteristics of the data could be distorted. When using this very powerful approach, the results should be carefully inspected to see if there are any undesirable side effects.

An interesting property of the matrix filter is that it can be set up so that the original pattern of returns for some asset classes can be preserved while the remaining classes are adjusted. This is accomplished by simply reordering the data matrix so that the asset class returns to be preserved are in the leftmost columns and the changeable classes are in the rightmost columns. This amazing property will be used in the next sections.

IV. CONTINGENT RETURNS

There are a number of situations where we want to add an additional asset class to a previously generated simulation. All the original data is to be preserved exactly as is and the additional asset class is to have a specified mean and standard deviation and specified correlations to the preexisting asset returns. These are called "contingent returns", since the return patterns of the new asset class depend on the existing classes.

For this case, we can use a variation of the matrix filter, using the previously mentioned technique to preserve the patterns of the original data and only adjust the added asset class. More

specifically, this technique is defined as follows:

Let D_0 = N x M matrix of simulated data with existing classes
M_0 N x M matrix of means of D_0
C_0 covariance matrix for D_0
R random vector with mean 0 and standard deviation 1
D_1 D_0 with R added as a rightmost column
M_1 M_0 with a column of zeroes added as a rightmost column
C_1 covariance matrix for D_1
C' target covariance matrix, equal to C_0, with new covariances appended in last row and column
M' matrix of target means, equal to M_0, with mean of added asset class in last column
F factor matrix for C_1
F' factor matrix for C'

Then,

$$D' = M' + (D1 - M1) \times (F' \times F^{-1}) \qquad (7)$$

Chart Two shows an example of adding contingent returns for an asset class to the returns for three pre-existing asset classes.

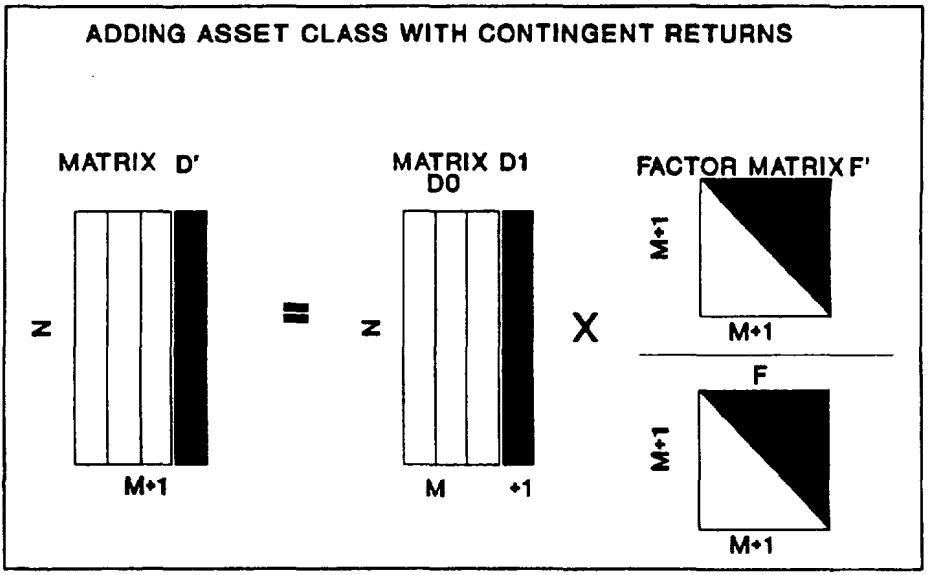

CHART TWO

One or more new asset classes can be added to the existing classes, as long as the covariances of the new classes to all

existing classes and to each new class can be specified. Appendix I shows a detailed numerical example of adding a new asset class to three pre-existing classes.

V. CONTROLLING SERIAL CORRELATION

The methodology for controlling serial correlation, the linkage between years, is very similar to the approach for creating contingent returns. There are two possible variations of this method - creating contingent returns with serial correlation and adjusting existing return patterns to specified serial correlations.

As mentioned in Section I, early economic simulation models assumed that returns were independent from year to year. More recent models assume that there is some negative serial correlation of returns between years. Serial correlation allows the modeler to directly control the shape of compound returns. Chart Three-A shows the results of a simulation of asset class returns with a mean of 13% and a standard deviation of 16%; the graph shows the distribution of compound returns of the "unlinked" yearly returns. By the tenth year, the standard deviation of the annualized compound return is 4.5%.

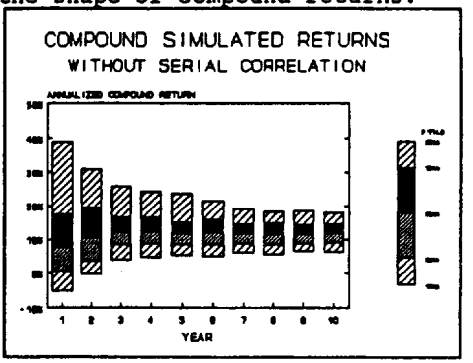

CHART THREE-A

Chart Three-B shows the results of a simulation with negative serial correlation; although the mean and standard deviation of all the asset returns is identical to the first simulation, the standard deviation of compound returns in the tenth year has dropped to 3.0%. Since the compound return patterns are relevant for forecasts with long time horizons, the ability to control the distribution of compound returns is critical to a successful simulation model.

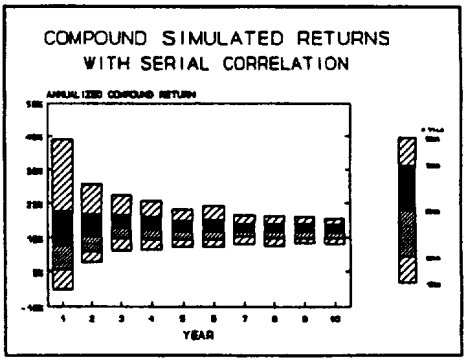

CHART THREE-B

To add contingent returns with specified serial correlation to the prior year's return for the new asset class, generate

the first year results by using the process explained in section IV.

For the second and later years, use the following methodology:

Let D_0 = N x M matrix of simulated data with existing classes
R_0 vector of simulated returns for new asset class in prior year
D_1 D_0 with R_0 added as leftmost column
M_1 N x M matrix of means of D_1
C_1 covariance matrix for D_1
R random vector with mean 0 and standard deviation 1
D_2 D_1 with R added as a rightmost column
M_2 M_1 with a column of zeroes added as a rightmost column
C_2 covariance matrix for D_2
C' target covariance matrix, equal to C_1, with new covariances appended in last row and column; the first row of the rightmost column has the covariance of the new asset class between years
M' matrix of target means, equal to M_1, with the mean of added asset class in last column
F factor matrix for C_2
F' factor matrix for C'

$$D'=M'+(D_2-M_2)\times(F'\times F^{-1}) \qquad (8)$$

Since C and C' only differ in the last column, all the other columns of returns will be preserved untouched. The last column of D' is the simulated returns for the new asset class, which will have the specified covariance, including the specified serial correlation to the prior year returns.

Chart FOUR gives a visual example of this process. For the example of three pre-existing classes, the prior year's data for the new class is added as the leftmost column and a random vector is added as the rightmost vector. After the matrix calculation, the rightmost column of the matrix product contains the returns for the new asset class in the current year. These returns are then used to generate the next year's returns.

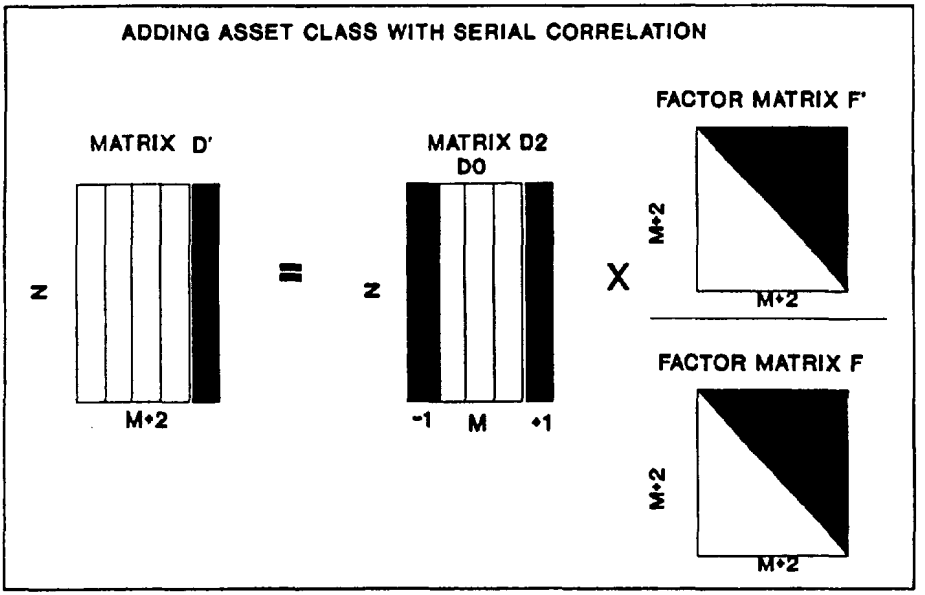

CHART FOUR

The process for adjusting existing results for serial correlation is similar. We want to preserve all existing means and covariances and only change the serial covariance, which is in the upper right and lower left cells of the augmented covariance matrix.

VI. "PURIFYING" RANDOM DATA

It's very easy to generate random data and force the means to zero and the standard deviations to unity; however, those changes do not necessarily remove the correlations from the data. Fortunately, the magical Choleski factor matrix can be used to remove all correlations. Simply generate a matrix of random data and use the matrix "filter" to force the covariance matrix to be equal to the Identity Matrix.

VII. FACTORING "UNFACTORABLE" MATRICES

As discussed above, some specified covariance matrices may not be factorable by the Choleski methodology. This may occur for two possible reasons. First, the specified correlations may not be economically or statistically sensible. (Note that a covariance matrix calculated from actual data will always be factorable, since the pattern of correlations of actual data is, by definition, sensible.) Second, the factor methodology may run into computational problems with large matrices, due to rounding error. Practitioners generally find that, as the number of asset classes increases, it becomes more difficult to factor the specified covariance matrix.

A possible solution to this problem is a special process to apply the matrix filter to data matrices with large numbers of variables. This process chooses random subsets of the variables and attempts to adjust each subset to the specified covariances for that subset. The next iteration chooses a different random subset from all the variables (including the results of all prior adjustments) and adjusts that subset to its specified covariances. If the specified covariances are sensible, this process will converge fairly quickly to the desired results.

VIII. CONCLUSION

The factor matrix approach has been used in economic models since the early 1980's. This paper presented some new uses of the factor matrix for simulating more complex asset models, controlling compound returns with specified serial correlations and for adjusting previously simulated results to meet specified means and covariances.

APPENDIX I
Numerical Example

This appendix gives a numerical example of the calculations to add an asset class with contingent returns to three pre-existing asset classes. For this simplified example, we will use 25 observations of each variable.

The methodology follows the process described in Section IV of the paper. For purposes of presentation, the data has been truncated to six decimals, or less. The actual calculations in APL2/PC use up to 12 digits of accuracy. Since a high degree of accuracy is needed to factor the covariance matrices, it is recommended that extended precision calculations be used.

The pre-existing data is as follows:

Observation	CLASS 1	CLASS 2	CLASS 3
1	0.0896	0.3824	0.2076
2	0.0656	0.2281	0.1181
3	0.0544	-0.0875	0.0605
4	0.0622	0.1076	0.0862
5	0.0395	0.1925	0.1189
6	0.0803	0.0174	0.0883
7	0.0188	-0.0538	0.0314
8	0.0464	0.1715	0.1202
9	0.0550	-0.1516	-0.0082
10	0.0679	-0.2269	-0.0085
11	0.1009	0.0300	0.0443
12	0.0551	-0.0614	0.0954
13	0.0780	0.0159	-0.0281
14	0.0471	0.2245	0.1059
15	0.0743	0.0135	0.0869
16	0.1041	-0.0344	0.0222
17	0.0712	0.3598	0.2150
18	0.0219	0.1108	0.0423
19	0.0585	0.0968	0.0750
20	0.0418	0.0463	0.0900
21	0.0959	0.4474	0.1702
22	0.0673	0.0027	-0.0449
23	0.0560	-0.0296	0.0235
24	0.0499	0.2466	0.1766
25	0.0534	-0.1291	0.0140
Mean	.0622	.0768	.0761
Std Dev[2]	.0213	.1677	.0683

[2] The standard deviation used here is the population standard deviation rather than the sample standard deviation (i.e., division by N rather than N-1). The population standard deviation is equal to the square root of the diagonals of the covariance matrix.

The correlation matrix for the data is the following:

	CLASS 1	CLASS 2	CLASS 3
CLASS 1	1.000		
CLASS 2	.171	1.000	
CLASS 3	.094	.840	1.000

Consistent with the standard deviations and correlations is the following covariance matrix:

	CLASS 1	CLASS 2	CLASS 3
CLASS 1	.000455		
CLASS 2	.000613	.028108	
CLASS 3	.000137	.009619	.004664

For the purposes of this example, we will add a fourth asset class, with the following characteristics:

Mean	.13
Std Dev.	.16
Correlations to Classes 1-3:	0, .2, .1

Based on these characteristics, in combination with the characteristics of the pre-existing classes, the target covariance matrix is the same as before, but with a row and column added. The reader can easily verify that the target covariance matrix is the following:

	CLASS 1	CLASS 2	CLASS 3	CLASS 4
CLASS 1	.000455			
CLASS 2	.000613	.028108		
CLASS 3	.000137	.009619	.004664	
CLASS 4	.000000	.005365	.001093	.025600

We will now append a column of 25 random numbers selected from a lognormal distribution with a mean and standard deviation of unity. The actual mean and standard deviation of the random sample will vary with sample error. The following table displays the numbers which were used for this example:

Random Numbers					
1-5	.488023	1.881652	.282318	.636009	2.307044
6-10	.812981	.239862	.399288	.582225	.127279
11-15	.322396	.536108	1.145508	.883130	1.047454
16-20	1.420966	.443956	1.270796	3.097224	.751726
21-25	.360330	.956060	.691400	1.622174	.188410

The Choleski factor matrix of the augmented data matrix in our example is shown in the following table:

	CLASS 1	CLASS 2	CLASS 3	CLASS 4
CLASS 1	.021331	.028737	.006406	-.070521
CLASS 2	0	.165174	.057123	.167693
CLASS 3	0	0	.036872	-.137580
CLASS 4	0	0	0	.664432

The Choleski factor matrix of the target covariance matrix is identical for the first three columns, corresponding to the pre-existing data, which is to be preserved. The target factor matrix is shown in the following table:

	CLASS 1	CLASS 2	CLASS 3	CLASS 4
CLASS 1	.021331	.028737	.006406	.000000
CLASS 2	0	.165174	.057123	.032481
CLASS 3	0	0	.036872	-.020686
CLASS 4	0	0	0	.155297

Then, if we multiply the target factor matrix by the inverse of the existing factor matrix, we get the following matrix:

	CLASS 1	CLASS 2	CLASS 3	CLASS 4
CLASS 1	1	0	0	.878989
CLASS 2	0	1	0	-.148231
CLASS 3	0	0	1	.311081
CLASS 4	0	0	0	.233729

Note that the first three columns and rows of this matrix form the Identity Matrix; this is the mechanism that preserves the original data.

After multiplying the augmented data matrix by the last matrix, we obtain the new data matrix, as shown in the following table:

	CLASS 1	CLASS 2	CLASS 3	CLASS 4
1	0.0896	0.3824	0.2076	0.0534
2	0.0656	0.2281	0.1181	0.3531
3	0.0544	-0.0875	0.0605	-0.0017
4	0.0622	0.1076	0.0862	0.0669
5	0.0395	0.1925	0.1189	0.4351
6	0.0803	0.0174	0.0883	0.1382
7	0.0188	-0.0538	0.0314	-0.0569
8	0.0464	0.1715	0.1202	-0.0012
9	0.0550	-0.1516	-0.0082	0.0571
10	0.0679	-0.2269	-0.0085	-0.0269
11	0.1009	0.0300	0.0443	0.0261
12	0.0551	-0.0614	0.0954	0.0652
13	0.0780	0.0159	-0.0281	0.1779
14	0.0471	0.2245	0.1059	0.1002
15	0.0743	0.0135	0.0869	0.1879
16	0.1041	-0.0344	0.0222	0.2884
17	0.0712	0.3598	0.2150	0.0326
18	0.0219	0.1108	0.0423	0.1657
19	0.0585	0.0968	0.0750	0.6370
20	0.0418	0.0463	0.0900	0.0863
21	0.0959	0.4474	0.1702	0.0079
22	0.0673	0.0027	-0.0449	0.1210
23	0.0560	-0.0296	0.0235	0.0752
24	0.0499	0.2466	0.1766	0.2941
25	0.0534	-0.1291	0.0140	-0.0328
MEAN	.0622	.0768	.0761	.1300
STD DEV	.0213	.1677	.0683	.1600

Note that the first three columns are identical to the starting data; the covariance matrix of the generated data is exactly equal to the target covariance matrix. Therefore, the generated asset class satisfies the requirements for mean, standard deviation and correlations.

APPENDIX II

This appendix contains source listings for several APL2/PC programs that facilitate the processes described in this paper. These functions are:

FACTOR	Calculate Choleski factor matrix for a given data matrix
CHANGE	Filter a given data matrix to meet specified covariances
CHANGEX	Filter a given data matrix with a large number of variables to meet specified covariances

```
[ 0]  B←FACTOR A;I;J;X
[ 1]  ⍝ FUNCTION TO CREATE CHOLESKI FACTOR MATRIX
[ 2]  ⍝ A IS SQUARE INPUT MATRIX
[ 3]  ⍝ B IS OUTPUT FACTOR MATRIX
[ 4]  B←(⍴A)⍴0
[ 5]  B[1;]←A[;1]÷A[1;1]*0.5
[ 6]  I←J←2
[ 7]  START:→(J=I)/DIAG
[ 8]  B[J;I]←(A[I;J]-(B[⍳I-1;I]+.×B[⍳I-1;J]))÷B[J;J]
[ 9]  →END
[10]  DIAG:→(0>X←A[I;J]-+/B[⍳I-1;I]*2)/ERROR
[11]  B[J;I]←X*0.5
[12]  END:→(I≥J←J+1)/START
[13]  J←2
[14]  →((1↑⍴A)≥I←I+1)/START
[15]  →0
[16]  ERROR:B←1⍴I          ⍝ERROR; RETURN ROW NUMBER OF ERROR
```

```
[ 0]  B←WANT CHANGE A;M
[ 1]  ⍝ FUNCTION TO CHANGE COVARIANCES OF A TO SPECIFIED COVARIANCES
[ 2]  ⍝ A IS INPUT DATA MATRIX
[ 3]  ⍝ WANT IS SPECIFIED COVARIANCE MATRIX
[ 4]  ⍝ B IS OUTPUT DATA MATRIX
[ 5]  ⍝ B WILL HAVE SPECIFIED COVARIANCE AND COLUMNAR MEANS OF A
[ 6]  B←FACTOR WANT
[ 7]  →(1=⍴,B)/NO              ⍝ TEST FOR FACTORABLE B MATRIX
[ 8]  M←(⍴A)⍴(+⌿A)÷1↑⍴A
[ 9]  B←M+(A-M)+.×B⌹FACTOR COVM A
[10]  NO:→0           ⍝IF ERROR, RETURN ROW WITH ERROR
```

```
[ 0]  B←WANT CHANGEX A;AA;BBB;COLS;EPS;LOW;MASK;MAXLOOP;NCOLS;NLOOP;R;T;W
[ 1]  ⍝ FUNCTION TO CHANGE LARGE DATA MATRICES TO
[ 2]  ⍝ SPECIFIED COVARIANCES
[ 3]  ⍝ A IS INPUT DATA MATRIX
[ 4]  ⍝ WANT IS SPECIFIED COVARIANCE MATRIX
[ 5]  ⍝ B IS OUTPUT DATA MATRIX
[ 6]  ⍝ B WILL HAVE SPECIFIED COVARIANCES AND COLUMNAR MEANS OF A
[ 7]  ⍝ THIS FUNCTION MAY WORK WHEN WANT WILL NOT FACTOR DUE TO ITS SIZE
[ 8]  ⍝ CHANGEX CALLS CHANGE ON SUBSETS OF THE DATA UNTIL CONVERGENCE
[ 9]  MAXLOOP←50
[10]  EPS←1E¯6
[11]  LOW←1000000
[12]  R←¯1↑⍴A
[13]  MASK←(⍳R)∘.≥⍳R
[14]  MASK←MASK÷+/,MASK
[15]  NLOOP←0
[16]  LOOP:NLOOP←NLOOP+1
[17]  →(NLOOP>MAXLOOP)/END
[18]  NCOLS←1+?¯2+R          ⍝ GET RANDOM NUMBER OF COLUMNS
[19]  COLS←(⍳R)[NCOLS?R]          ⍝ GET UNIQUE SAMPLE OF COLUMNS
[20]  COLS←COLS[⍋COLS]            ⍝ SORT INTO ORIGINAL ORDER
[21]  W←WANT[COLS;COLS]
[22]  BBB←W CHANGE A[;COLS]          ⍝ ITERATE THROUGH CHANGE
[23]  →(1=⍴,BBB)/LOOP        ⍝DIDN'T FACTOR, TRY AGAIN
[24]  A[;COLS]←BBB
[25]  →(LOW≥T←(+/,(MASK×WANT-COVM A)*2)*0.5)/NEXT    ⍝RMSE ERROR
[26]  AA←A                        ⍝SAVE LOWEST RESULT
[27]  ⍝ALTHOUGH ALGORITHM LETS ERROR GET HIGHER
[28]  ⍝CONVERGENCE IS ACTUALLY BETTER
[29]  LOW←T
[30]  NEXT:→(EPS≥□←T)/OUT       ⍝STOP IF ERROR ≤ EPS
[31]  →LOOP
[32]  END:□←'DOES NOT CONVERGE'              ⍝MAX LOOPS
[33]  OUT:B←AA                          ⍝GET BEST RESULT
```

ACTUARIAL EDUCATION AND RESEARCH FUND
1993 PRACTITIONERS AWARD PAPERS