ORIGIN ≡ 1

Methods of Matrix Decomposition

There are very many kids of matrix "decomposition" that is, expressing a matrix as a product of other matricies that are equivalent. Typically objective is simplification of the problem for graphical display, as in ordinations, or for other analytic reasons, such as solving systems of linear equations. By far the most commonly encountered kinds of decomposition in multivariate ordinations involve "spectral decomposition" and/or "singular value decomposition", both described here with examples. For extended discussion, see RA Johnson & DW Wichern *Applied Multivariate Statistical Analysis 4th Edition* 1998, or AC. Rencher *Methods of Multivariate Analysis* 1995. Also searching the web for a particular kind of decomposition often results in very useful information - especially in Wikipedia.

Spectral Decomposition of a Symmetric Matrix:

Spectral decomposition works with square symmetric matricies and underlies "Eigen" methods such as Principal Components Analysis (PCA) that analyze the symmetric covariance matrix S of a dataset.

Reading Data:

M := READPRN("c:/DATA/Multivariate/Isetosastand.txt")

n := rows(M) = (50)

p := cols(M) = (4)

i := 1 .. n j := 1 .. 4 **< index variables**

$l_{vec_i} := 1$

I := identity(n)

Covariance Matrix:

$$S := \frac{1}{n-1}\cdot M^T\cdot\left(I - \frac{1}{n}\cdot l_{vec}\cdot l_{vec}^T\right)\cdot M \qquad S = \begin{pmatrix} 1 & 0.7425467 & 0.2671757 & 0.2780984 \\ 0.7425467 & 1 & 0.1777 & 0.232752 \\ 0.2671757 & 0.1777 & 0.9999999 & 0.33163 \\ 0.2780984 & 0.232752 & 0.33163 & 1 \end{pmatrix}$$

Calculating Eigenvalues & Eigenvectors:

λ := eigenvals(S)

$$\lambda = \begin{pmatrix} 2.0585402 \\ 0.2514613 \\ 1.0221782 \\ 0.6678202 \end{pmatrix}$$

Λ := diag(λ)

$$\Lambda = \begin{pmatrix} 2.05854 & 0 & 0 & 0 \\ 0 & 0.251461 & 0 & 0 \\ 0 & 0 & 1.022178 & 0 \\ 0 & 0 & 0 & 0.66782 \end{pmatrix}$$

E := eigenvecs(S)

$$E = \begin{pmatrix} 0.6044164 & 0.7196698 & -0.3349908 & -0.0673598 \\ 0.5756194 & -0.6887064 & -0.4408461 & -0.0007138 \\ 0.3754348 & -0.0868399 & 0.6269717 & -0.6770628 \\ 0.4029788 & -0.014752 & 0.5480351 & 0.7328356 \end{pmatrix}$$

$$E\cdot E^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad E^T\cdot E = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

< Matrix E is orthogonal and composed of column vectors of unit length.

Definition of Spectral Decomposition:

A square symmetric matrix, such as matrix A, can be expressed in terms of "orthonormal" eigenvector matrix E (eigenvectors of S are orthogonal and of unit length) and "diagonal" matrix Λ (eigenvalues of S along the main diagonal).

$$S = \begin{pmatrix} 1 & 0.742547 & 0.267176 & 0.278098 \\ 0.742547 & 1 & 0.1777 & 0.232752 \\ 0.267176 & 0.1777 & 1 & 0.33163 \\ 0.278098 & 0.232752 & 0.33163 & 1 \end{pmatrix} \qquad E \cdot \Lambda \cdot E^{T} = \begin{pmatrix} 1 & 0.742547 & 0.267176 & 0.278098 \\ 0.742547 & 1 & 0.1777 & 0.232752 \\ 0.267176 & 0.1777 & 1 & 0.33163 \\ 0.278098 & 0.232752 & 0.33163 & 1 \end{pmatrix}$$

Diagonalizing a Symmetric Matrix:

Using identities for an orthonormal matrix, the diagonal matrix can be expressed in terms of symmetric matrix S and eigenvector matrix E: This expression is commonly termed "diagonalizing" matrix S, but is simply a restatement of the Eigenvector equation seen in MTB 050.

$$E^{T} \cdot S \cdot E = \begin{pmatrix} 2.059 & 0 & 0 & 0 \\ 0 & 0.251 & 0 & 0 \\ 0 & 0 & 1.022 & 0 \\ 0 & 0 & 0 & 0.668 \end{pmatrix} \qquad \Lambda = \begin{pmatrix} 2.05854 & 0 & 0 & 0 \\ 0 & 0.251461 & 0 & 0 \\ 0 & 0 & 1.022178 & 0 \\ 0 & 0 & 0 & 0.66782 \end{pmatrix}$$

Expression as Partial Sums:

The original matrix S can be restated as a sum of Partial Matrices P derived from the Eigenvalues and Eigenvectors of S. Summing the Partials, reconstitutes S:

$$P_j := \left[\left(E^{\langle j \rangle} \right) \cdot \lambda_j \cdot E^{\langle j \rangle^{T}} \right]$$

$$P = \left[\begin{pmatrix} 0.752 & 0.7162 & 0.4671 & 0.5014 \\ 0.7162 & 0.6821 & 0.4449 & 0.4775 \\ 0.4671 & 0.4449 & 0.2902 & 0.3114 \\ 0.5014 & 0.4775 & 0.3114 & 0.3343 \end{pmatrix} \begin{pmatrix} 0.1302 & -0.1246 & -0.0157 & -0.0027 \\ -0.1246 & 0.1193 & 0.015 & 0.0026 \\ -0.0157 & 0.015 & 0.0019 & 0.0003 \\ -0.0027 & 0.0026 & 0.0003 & 5.4724 \times 10^{-5} \end{pmatrix} \begin{pmatrix} 0.1147 & 0.151 & -0.2147 & -0.1877 \\ 0.151 & 0.1987 & -0.2825 & -0.247 \\ -0.2147 & -0.2825 & 0.4018 & 0.3512 \\ -0.1877 & -0.247 & 0.3512 & 0.307 \end{pmatrix} \begin{pmatrix} 0.003 & 3.211 \times 10^{-5} & 0.0305 & -0.033 \\ 3.211 \times 10^{-5} & 3.4027 \times 10^{-7} & 0.0003 & -0.0003 \\ 0.0305 & 0.0003 & 0.3061 & -0.3314 \\ -0.033 & -0.0003 & -0.3314 & 0.3587 \end{pmatrix} \right]$$

$$\sum_{j} P_j = \begin{pmatrix} 1 & 0.742547 & 0.267176 & 0.278098 \\ 0.742547 & 1 & 0.1777 & 0.232752 \\ 0.267176 & 0.1777 & 1 & 0.33163 \\ 0.278098 & 0.232752 & 0.33163 & 1 \end{pmatrix} \qquad S = \begin{pmatrix} 1 & 0.742547 & 0.267176 & 0.278098 \\ 0.742547 & 1 & 0.1777 & 0.232752 \\ 0.267176 & 0.1777 & 1 & 0.33163 \\ 0.278098 & 0.232752 & 0.33163 & 1 \end{pmatrix}$$

Prototype in R:

Note: reporting order of eigenvalues and eigenvectors in R are different than in MathCad but in both cases properly matched.

```
#METHODS OF MATRIX DECOMPOSITION:
#READ DATA:
M=read.table("c:/DATA/Multivariate/Isetosastand.txt",header=F)
M
attach(M)
#CALCULATION COVARIANCE MATRIX S:
S=cov(M)
S
```

```
> S
          V1        V2        V3        V4
V1 1.0000000 0.7425467 0.2671757 0.2780984
V2 0.7425467 1.0000000 0.1777000 0.2327520
V3 0.2671757 0.1777000 0.9999999 0.3316300
V4 0.2780984 0.2327520 0.3316300 1.0000000
```

```
#EIGENVALUES & EIGENVECTORS OF S:
lambda=eigen(S)$values
lambda
E=eigen(S)$vectors
E
```

```
> lambda
[1] 2.0585402 1.0221782 0.6678202 0.2514613
```

```
> E
          [,1]       [,2]          [,3]        [,4]
[1,] 0.6044164  0.3349908 -0.0673598259  0.71966982
[2,] 0.5756194  0.4408461 -0.0007138139 -0.68870645
[3,] 0.3754348 -0.6269717 -0.6770628473 -0.08683987
[4,] 0.4029788 -0.5480351  0.7328356195 -0.01475204
```

```
#SPECTRAL DECOMPOSITION:
S
E%*%diag(lambda)%*%t(E)
```

```
> E%*%diag(lambda)%*%t(E)
          [,1]      [,2]      [,3]      [,4]
[1,] 1.0000000 0.7425467 0.2671757 0.2780984
[2,] 0.7425467 1.0000000 0.1777000 0.2327520
[3,] 0.2671757 0.1777000 0.9999999 0.3316300
[4,] 0.2780984 0.2327520 0.3316300 1.0000000
```

```
#DIAGONALIZING S:
diag(lambda)
t(E)%*%S%*%E
```

```
> t(E)%*%S%*%E
              [,1]          [,2]          [,3]          [,4]
[1,]  2.058540e+00 -9.436896e-16  3.330669e-16 -2.550044e-16
[2,] -9.436896e-16  1.022178e+00 -5.551115e-16 -2.359224e-16
[3,]  2.775558e-16 -5.551115e-16  6.678202e-01 -5.204170e-17
[4,] -2.990230e-16 -2.506675e-16 -2.992398e-17  2.514613e-01
```

```
#EXPRESSION AS PARTIAL SUMS:
P1=(E[,1]*lambda[1])%*%t(E[,1])
P1
P2=(E[,2]*lambda[2])%*%t(E[,2])
P2
P3=(E[,3]*lambda[3])%*%t(E[,3])
P3
P4=(E[,4]*lambda[4])%*%t(E[,4])
P4
```

```
> P1
          [,1]      [,2]      [,3]      [,4]
[1,] 0.7520242 0.7161945 0.4671217 0.5013924
[2,] 0.7161945 0.6820719 0.4448660 0.4775039
[3,] 0.4671217 0.4448660 0.2901538 0.3114412
[4,] 0.5013924 0.4775039 0.3114412 0.3342902
> P2
           [,1]       [,2]       [,3]       [,4]
[1,]  0.1147076  0.1509546 -0.2146878 -0.1876583
[2,]  0.1509546  0.1986555 -0.2825280 -0.2469573
[3,] -0.2146878 -0.2825280  0.4018116  0.3512230
[4,] -0.1876583 -0.2469573  0.3512230  0.3070035
> P3
              [,1]          [,2]          [,3]          [,4]
[1,]  3.030132e-03  3.211039e-05  0.0304571674 -0.0329660639
[2,]  3.211039e-05  3.402746e-07  0.0003227554 -0.0003493423
[3,]  3.045717e-02  3.227554e-04  0.3061382091 -0.3313562176
[4,] -3.296606e-02 -3.493423e-04 -0.3313562176  0.3586515490
> P4
             [,1]          [,2]          [,3]          [,4]
[1,]  0.130238003 -0.124634589 -0.0157153327 -2.669664e-03
[2,] -0.124634589  0.119272259  0.0150391897  2.554803e-03
[3,] -0.015715333  0.015039190  0.0018963104  3.221383e-04
[4,] -0.002669664  0.002554803  0.0003221383  5.472369e-05
```

```
P1+P2+P3+P4
```

```
> P1+P2+P3+P4
          [,1]      [,2]      [,3]      [,4]
[1,] 1.0000000 0.7425467 0.2671757 0.2780984
[2,] 0.7425467 1.0000000 0.1777000 0.2327520
[3,] 0.2671757 0.1777000 0.9999999 0.3316300
[4,] 0.2780984 0.2327520 0.3316300 1.0000000
```

Singular Value Decomposition:

Singular Value Decomposition is an alternative "Eigen" method that works with a non-square matrix and is often used with data matrix M directly. Central to the method are calulations of two matrices U & V usually of different dimensions that are multiplied with diagonal matrix Γ of singular values that are the square root of Λ derived from S as defined above.

Example using small matrix A:

$A := \begin{pmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{pmatrix}$ **< Using the Matrix A defined in JW Example p. 102-103:**

Eigenvalues and Eigenvectors from Sums of Squares of M:

Note: Eigenvalues for the different products of A are designated λ (lambda) λ_U & λ_V whereas singular values are designated γ (gamma).

$\lambda_U := \text{reverse}(\text{sort}(\text{eigenvals}(A \cdot A^T)))$ $\lambda_U = \begin{pmatrix} 12 \\ 10 \end{pmatrix}$

$\lambda_V := \text{reverse}(\text{sort}(\text{eigenvals}(A^T \cdot A)))$ $\lambda_V = \begin{pmatrix} 12 \\ 10 \\ 0 \end{pmatrix}$ **< Only the minimum set of eigenvalues are used as defined by the smaller of A^TA or AA^T.**

$m := 1 .. \text{rows}(A \cdot A^T)$ $k := 1 .. \text{rows}(A^T \cdot A)$

Associated Eigenvectors are extracted:

$U^{\langle m \rangle} := \text{eigenvec}(A \cdot A^T, \lambda_{U_m})$ $U = \begin{pmatrix} 0.707 & -0.707 \\ 0.707 & 0.707 \end{pmatrix}$

$V^{\langle k \rangle} := \text{eigenvec}(A^T \cdot A, \lambda_{V_k})$ $V = \begin{pmatrix} 0.408 & -0.894 & -0.183 \\ 0.816 & 0.447 & -0.365 \\ 0.408 & 0 & 0.913 \end{pmatrix}$

Singular Values:

$m := \text{rows}(A)$ $k := \text{cols}(A)$

$\gamma := \sqrt{\lambda_U}$ $\gamma = \begin{pmatrix} 3.464 \\ 3.162 \end{pmatrix}$ **< Singular values γ are calculated.**

^ Again, only a minimum set are used as defined by the smaller of U or V.

$r := 1 .. k - m$ $k - m = 1$ $Z_{m,r} := 0$ $Z = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ **(k-m) Extra columns of zeros are needed for singular value matrix Γ**

$\Gamma := \text{augment}(\text{diag}(\gamma), Z)$ $\Gamma = \begin{pmatrix} 3.464 & 0 & 0 \\ 0 & 3.162 & 0 \end{pmatrix}$ **< Γ is a diagonal matrix of singular values augmented by zeros**

Singular Value Decomposition Defined:

$U \cdot \Gamma \cdot V^T = \begin{pmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{pmatrix}$ $A = \begin{pmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{pmatrix}$ **Singular Value Decomposition confirmed!**

Using MathCad's built-in functions:

$\gamma := \text{svds}(A^T)$ $\gamma = \begin{pmatrix} 3.464 \\ 3.162 \end{pmatrix}$ **< function svds() calculates singular values but requires the number of rows of matrix A to be equal or greater than number of columns. The transpose of A fixes this. Only non-zero singular values are reported.**

$\text{svd}(A^T) = \begin{pmatrix} 0.408 & 0.894 \\ 0.816 & -0.447 \\ 0.408 & 0 \\ 0.707 & 0.707 \\ 0.707 & -0.707 \end{pmatrix}$

$U = \begin{pmatrix} 0.707 & -0.707 \\ 0.707 & 0.707 \end{pmatrix}$ $V = \begin{pmatrix} 0.408 & -0.894 & -0.183 \\ 0.816 & 0.447 & -0.365 \\ 0.408 & 0 & 0.913 \end{pmatrix}$ **< calculated above**

< function svd() provides a stacked array of eigenvectors only showing the minimum required by the smaller of U or V. Note also that the order of U and V are reversed.

Example using data matrix M:

Eigenvalues:

$\lambda_U := \text{reverse}(\text{sort}(\text{eigenvals}(M \cdot M^T)))$

$\lambda_V := \text{reverse}(\text{sort}(\text{eigenvals}(M^T \cdot M)))$

$$\lambda_V = \begin{pmatrix} 100.868 \\ 50.087 \\ 32.723 \\ 12.322 \end{pmatrix}$$

< Only the minimum set of eigenvalues are used as defined by the smaller of M^TM or MM^T.

Singular Values:

$\gamma := \text{svds}(M)$

^ MathCad's built-in function svds()

$$\gamma = \begin{pmatrix} 10.04333 \\ 7.077198 \\ 5.720419 \\ 3.510214 \end{pmatrix} \qquad \sqrt{\lambda_V} = \begin{pmatrix} 10.043 \\ 7.077 \\ 5.72 \\ 3.51 \end{pmatrix}$$

< calculation by eigenvectors above is confirmed

Associated Eigenvector matrices U & V:

$m := 1 .. \text{rows}(M) \qquad k := 1 .. \text{cols}(M)$

$U^{\langle k \rangle} := \text{eigenvec}(M \cdot M^T, \lambda_{U_k})$

< U is a 50 X 4 matrix that is too large to show here

$V^{\langle k \rangle} := \text{eigenvec}(M^T \cdot M, \lambda_{V_k})$

$$V = \begin{pmatrix} 0.60442 & -0.33499 & 0.06736 & -0.71967 \\ 0.57562 & -0.44085 & 0.00071 & 0.68871 \\ 0.37543 & 0.62697 & 0.67706 & 0.08684 \\ 0.40298 & 0.54804 & -0.73284 & 0.01475 \end{pmatrix}$$

$UV := \text{svd}(M)$

^ MathCad's built-in function svd() provides values for U and V stacked in order.

U : rows 1-50
V: rows 51-54

$UV =$

	1	2	3	4
45	-0.22521	-0.26284	0.11457	0.20641
46	0.09267	-0.10605	-0.11492	-0.10839
47	-0.08449	0.03716	0.15323	0.15569
48	0.13465	-0.02656	0.00003	0.10747
49	-0.08199	0.09859	0.09173	-0.02664
50	0.05124	0.04359	0.01342	-0.07343
51	-0.60442	0.33499	0.06736	-0.71967
52	-0.57562	0.44085	0.00071	0.68871
53	-0.37543	-0.62697	0.67706	0.08684
54	-0.40298	-0.54804	-0.73284	0.01475

Same as V above >

Prototype in R:

```
#SINGULAR VALUE DECOMPOSITION:
#SINGULAR VALUES:
gamma=svd(M)$d
gamma

#LEFT MATRIX U:
U=svd(M)$u
U

#RIGHT MATRIX V:
V=svd(M)$v
V
```

^ Note: calculation error discovered here in MathCad (inconsistent directionality of Eigenvectors in V). So all calculations proceed from this point only with R.

```
> gamma
[1] 10.043330  7.077198  5.720419  3.510214

> V
           [,1]       [,2]          [,3]        [,4]
[1,] -0.6044164  0.3349908 -0.0673598259 -0.71966982
[2,] -0.5756194  0.4408461 -0.0007138139  0.68870645
[3,] -0.3754348 -0.6269717 -0.6770628473  0.08683987
[4,] -0.4029788 -0.5480351  0.7328356195  0.01475204
```

#IDENTITY DEMONSTRATED:
U%*%diag(gamma)%*%t(V)
M

> U%*%diag(gamma)%*%t(V)

```
            [,1]        [,2]       [,3]       [,4]
 [1,]  0.26667447  0.18994136 -0.3570112 -0.4364923
 [2,] -0.30071802 -1.12909583 -0.3570112 -0.4364923
 [3,] -0.86811050 -0.60148096 -0.9328358 -0.4364923
 [4,] -1.15180675 -0.86528840  0.2188133 -0.4364923
 [5,] -0.01702177  0.45374879 -0.3570112 -0.4364923
 [6,]  1.11776320  1.24517111  1.3704625  1.4613004
 [7,] -1.15180675 -0.07386608 -0.3570112  0.5124040
 [8,] -0.01702177 -0.07386608  0.2188133 -0.4364923
 [9,] -1.71919923 -1.39290327 -0.3570112 -0.4364923
...
[45,]  0.26667447  0.98136367  2.5221117  1.4613004
[46,] -0.58441426 -1.12909583 -0.3570112  0.5124040
[47,]  0.26667447  0.98136367  0.7946379 -0.4364923
[48,] -1.15180675 -0.60148096 -0.3570112 -0.4364923
[49,]  0.83406695  0.71755623  0.2188133 -0.4364923
[50,] -0.01702177 -0.33767352 -0.3570112 -0.4364923
```

> M

```
           V1          V2         V3         V4
1   0.26667447  0.18994136 -0.3570112 -0.4364923
2  -0.30071802 -1.12909583 -0.3570112 -0.4364923
3  -0.86811050 -0.60148096 -0.9328358 -0.4364923
4  -1.15180675 -0.86528840  0.2188133 -0.4364923
5  -0.01702177  0.45374879 -0.3570112 -0.4364923
6   1.11776320  1.24517111  1.3704625  1.4613004
7  -1.15180675 -0.07386608 -0.3570112  0.5124040
8  -0.01702177 -0.07386608  0.2188133 -0.4364923
9  -1.71919923 -1.39290327 -0.3570112 -0.4364923
...
45  0.26667447  0.98136367  2.5221117  1.4613004
46 -0.58441426 -1.12909583 -0.3570112  0.5124040
47  0.26667447  0.98136367  0.7946379 -0.4364923
48 -1.15180675 -0.60148096 -0.3570112 -0.4364923
49  0.83406695  0.71755623  0.2188133 -0.4364923
50 -0.01702177 -0.33767352 -0.3570112 -0.4364923
```

Positive Definite Matrix:

A matrix is "positive definite" if all eigenvalues of the matrix are positve. If some are zero, then the matrix is termed "positive semi-definite". The requirement that a covariance matrix S be positive definite is common in multivariate statistics. A positive definite matrix has additional properties including a matrix square root. and inverse square root.

Square Root Matrix:

A square and positive definite matrix A has only one square root matrix $A^{1/2}$, also known as the principal square root. A square root matrix is a square matrix when multiplied by itself reconstitutes matrix A. A common but by no means only way is by means of a variant of spectral decomposition.

Example Matrix Square Root:

Using covariance matrix of M calculated above with diagonal Eigenvector matrix Λ and Eigenvectors E:

$$S = \begin{pmatrix} 1 & 0.7425 & 0.2672 & 0.2781 \\ 0.7425 & 1 & 0.1777 & 0.2328 \\ 0.2672 & 0.1777 & 1 & 0.3316 \\ 0.2781 & 0.2328 & 0.3316 & 1 \end{pmatrix} \qquad \lambda = \begin{pmatrix} 2.059 \\ 0.251 \\ 1.022 \\ 0.668 \end{pmatrix} \qquad E = \begin{pmatrix} 0.604 & 0.72 & -0.335 & -0.067 \\ 0.576 & -0.689 & -0.441 & -0.001 \\ 0.375 & -0.087 & 0.627 & -0.677 \\ 0.403 & -0.015 & 0.548 & 0.733 \end{pmatrix}$$

Square root of the Eigenvalues:

$$\sqrt{\lambda} = \begin{pmatrix} 1.435 \\ 0.501 \\ 1.011 \\ 0.817 \end{pmatrix} \qquad \Lambda_{sqrt} := \operatorname{diag}\left(\sqrt{\lambda}\right) \qquad \Lambda_{sqrt} = \begin{pmatrix} 1.435 & 0 & 0 & 0 \\ 0 & 0.501 & 0 & 0 \\ 0 & 0 & 1.011 & 0 \\ 0 & 0 & 0 & 0.817 \end{pmatrix}$$

$$S_{sqrt} := E \cdot \Lambda_{sqrt} \cdot E^{T}$$

$$S_{sqrt} = \begin{pmatrix} 0.9010283 & 0.3999767 & 0.1191593 & 0.1181852 \\ 0.3999767 & 0.9097298 & 0.0610023 & 0.0932144 \\ 0.1191593 & 0.0610023 & 0.9780589 & 0.1596263 \\ 0.1181852 & 0.0932144 & 0.1596263 & 0.9756346 \end{pmatrix}$$

Identity demonstrated:

$$S_{sqrt} \cdot S_{sqrt} = \begin{pmatrix} 1 & 0.7425 & 0.2672 & 0.2781 \\ 0.7425 & 1 & 0.1777 & 0.2328 \\ 0.2672 & 0.1777 & 1 & 0.3316 \\ 0.2781 & 0.2328 & 0.3316 & 1 \end{pmatrix}$$

Prototype in R:

Since no off the shelf function exists for finding the matrix square root, so I built a function to do this. This is a very nice feature of R. This function was previously used in MTB 040.

```
#MATRIX SQUARE ROOT:

#FUNCTION FOR SQUARE ROOT OF A SYMMETRIC POSITIVE DEFINITE MATRIX:
matrix.square.root <- function(M) {
M.eig <- eigen(M)
M.square.root <- M.eig$vectors %*% diag(sqrt(M.eig$values)) %*% solve(M.eig$vectors)
return(M.square.root)
}

Ssqrt=matrix.square.root(S)
Ssqrt
```

```
> Ssqrt
          [,1]      [,2]      [,3]      [,4]
[1,] 0.9010283 0.39997674 0.11915933 0.11818522
[2,] 0.3999767 0.90972985 0.06100234 0.09321439
[3,] 0.1191593 0.06100234 0.97805887 0.15962631
[4,] 0.1181852 0.09321439 0.15962631 0.97563455
```

```
#IDENTITY DEMONSTRATED:
Ssqrt%*%Ssqrt
```

```
> Ssqrt%*%Ssqrt
          [,1]      [,2]      [,3]      [,4]
[1,] 1.0000000 0.7425467 0.2671757 0.2780984
[2,] 0.7425467 1.0000000 0.1777000 0.2327520
[3,] 0.2671757 0.1777000 0.9999999 0.3316300
[4,] 0.2780984 0.2327520 0.3316300 1.0000000
```

Matrix Inverse Square Root:

The inverse of a square root matrix $A^{-1/2}$ of matrix A may be calculated similarily.

$$\Lambda_{invsqrt} := \mathrm{diag}\left(\frac{1}{\sqrt{\lambda}}\right) \qquad \Lambda_{invsqrt} = \begin{pmatrix} 0.697 & 0 & 0 & 0 \\ 0 & 1.994 & 0 & 0 \\ 0 & 0 & 0.989 & 0 \\ 0 & 0 & 0 & 1.224 \end{pmatrix}$$

$$S_{invsqrt} := E \cdot \Lambda_{invsqrt} \cdot E^{T} \qquad S_{invsqrt} = \begin{pmatrix} 1.40400233 & -0.59978173 & -0.11840065 & -0.09339991 \\ -0.59978173 & 1.36903445 & -0.00290273 & -0.05767019 \\ -0.11840065 & -0.00290273 & 1.06303947 & -0.15930687 \\ -0.09339991 & -0.05767019 & -0.15930687 & 1.0678627 \end{pmatrix}$$

Identity demonstrated:

$$S_{invsqrt} \cdot S_{invsqrt} = \begin{pmatrix} 2.354 & -1.657 & -0.275 & -0.177 \\ -1.657 & 2.237 & 0.073 & -0.084 \\ -0.275 & 0.073 & 1.169 & -0.328 \\ -0.177 & -0.084 & -0.328 & 1.178 \end{pmatrix} \qquad S^{-1} = \begin{pmatrix} 2.354 & -1.657 & -0.275 & -0.177 \\ -1.657 & 2.237 & 0.073 & -0.084 \\ -0.275 & 0.073 & 1.169 & -0.328 \\ -0.177 & -0.084 & -0.328 & 1.178 \end{pmatrix}$$

Prototype in R:

```
#MATRIX INVERSE SQUARE ROOT:

#FUNCTION FOR INVERSE SQUARE ROOT OF A SYMMETRIC POSITIVE DEFINITE MATRIX:
matrix.inverse.square.root <- function(M) {
M.eig <- eigen(M)
M.square.root <- M.eig$vectors %*% diag(1/sqrt(M.eig$values)) %*% solve(M.eig$vectors)
return(M.square.root)
}

Sinvsqrt=matrix.inverse.square.root(S)
Sinvsqrt
```

```
> Sinvsqrt
             [,1]         [,2]         [,3]         [,4]
[1,]  1.40400233 -0.599781733 -0.118400648 -0.09339991
[2,] -0.59978173  1.369034455 -0.002902725 -0.05767019
[3,] -0.11840065 -0.002902725  1.063039465 -0.15930687
[4,] -0.09339991 -0.057670190 -0.159306865  1.06786270
```

Matrix Inverse:

The inverse of a matrix A^{-1} of matrix A, A^{-1}, may also be expressed in terms of spectral decomposition.

$$\Lambda_{inv} := \mathrm{diag}\left(\frac{1}{\lambda}\right) \qquad \Lambda_{inv} = \begin{pmatrix} 0.486 & 0 & 0 & 0 \\ 0 & 3.977 & 0 & 0 \\ 0 & 0 & 0.978 & 0 \\ 0 & 0 & 0 & 1.497 \end{pmatrix}$$

$$S_{inv} := E \cdot \Lambda_{inv} \cdot E^{T} \qquad S_{inv} = \begin{pmatrix} 2.35370294 & -1.65748674 & -0.2754791 & -0.1774204 \\ -1.65748674 & 2.23732774 & 0.07314216 & -0.08405434 \\ -0.2754791 & 0.07314216 & 1.16945872 & -0.32824133 \\ -0.1774204 & -0.08405434 & -0.32824133 & 1.17775881 \end{pmatrix}$$

Identity demonstrated:

$$S_{inv} = \begin{pmatrix} 2.354 & -1.657 & -0.275 & -0.177 \\ -1.657 & 2.237 & 0.073 & -0.084 \\ -0.275 & 0.073 & 1.169 & -0.328 \\ -0.177 & -0.084 & -0.328 & 1.178 \end{pmatrix} \qquad S^{-1} = \begin{pmatrix} 2.354 & -1.657 & -0.275 & -0.177 \\ -1.657 & 2.237 & 0.073 & -0.084 \\ -0.275 & 0.073 & 1.169 & -0.328 \\ -0.177 & -0.084 & -0.328 & 1.178 \end{pmatrix}$$

Prototype in R:

```
#MATRIX INVERSE:
Sinv=solve(S)
Sinv
```

```
> Sinv
           V1          V2          V3          V4
V1  2.3537029 -1.65748674 -0.27547910 -0.17742040
V2 -1.6574867  2.23732774  0.07314216 -0.08405434
V3 -0.2754791  0.07314216  1.16945872 -0.32824133
V4 -0.1774204 -0.08405434 -0.32824133  1.17775881
```