R Library Functions for Multivariate Analysis

Psychology 312

Dr. Steiger

The R Library Functions are in a file called "**Steiger R Library Functions.R**," which you can download from the course website in the "R Code and Support Materials" page.

Download the file and load the functions into R, either by using the clipboard or by placing them in your working directory and issuing the command

```
> source("Steiger R Library Functions.R")
```

Below I describe these functions and their use. These are not "commercial" functions and do not perform tests for improper input. Use them with care!

Function: **UnitVector(n)**

Input Description: **n** is an integer.

This function generates a column vector with n ones.

Example.

```
> UnitVector(5)
     [,1]
[1,]    1
[2,]    1
[3,]    1
[4,]    1
[5,]    1
```

Function: **P(X)**

Input Description: **X** is a $p \times q$ matrix of full column rank

This function creates the orthogonal projector for the column space of the matrix $\mathbf{X}$, defined as

$$\mathbf{P}_{\mathbf{X}} = \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'$$

Example.

```
> X<-matrix(c(2,4,2,1,3,1,3,1),4,2)
> P(X)
            [,1]        [,2]        [,3]       [,4]
[1,]  0.47867299 -0.01421801  0.47867299 0.14218009
[2,] -0.01421801  0.99052133 -0.01421801 0.09478673
[3,]  0.47867299 -0.01421801  0.47867299 0.14218009
[4,]  0.14218009  0.09478673  0.14218009 0.05213270
```

$\mathbf{P}_{\mathbf{X}}$ will project any $p \times 1$ vector into the column space of $\mathbf{X}$. For example,

```
> y<-matrix(c(3,2,2,2),4,1)
> P(X)%*%y
         [,1]
[1,] 2.649289
[2,] 2.099526
[3,] 2.649289
[4,] 1.004739
```

Any vector that is a linear combination of the columns of **X** will be left invariant by $\mathbf{P}_X$.

For example, suppose we take twice the first column of **X** and add it to the second column, using a weight vector **b**, i.e.,

```
> b <- matrix(c(2,1),2,1)
> b
     [,1]
[1,]    2
[2,]    1
> X %*% b
     [,1]
[1,]    7
[2,]    9
[3,]    7
[4,]    3
> w<-X %*% b
> w
     [,1]
[1,]    7
[2,]    9
[3,]    7
[4,]    3
> P(x) %*% w
     [,1]
[1,]    7
[2,]    9
[3,]    7
[4,]    3
```

Function: **Q(X)**

Input Description: **X** is a $p \times q$ matrix of full column rank

This function returns the complementary orthogonal projector for **X**, which projects any $p \times 1$ vector into the space orthogonal to the space of **X** in $\Re_p$

Example.

Take the vector $\mathbf{b} = \begin{bmatrix} 3 \\ 3 \\ 5 \\ 9 \end{bmatrix}$. Again let $\mathbf{X} = \begin{bmatrix} 2 & 3 \\ 4 & 1 \\ 2 & 3 \\ 1 & 1 \end{bmatrix}$.

We compute $\mathbf{w} = \mathbf{Q}_\mathbf{X}\mathbf{b}$ as

```
> b<-matrix(c(3,3,5,9),4,1)
> b
     [,1]
[1,]    3
[2,]    3
[3,]    5
[4,]    9
> w <- Q(X)%*%b
             [,1]
[1,] -2.06635071
[2,] -0.71090047
[3,] -0.06635071
[4,]  7.10900474
```

$\mathbf{w}$ is orthogonal to both columns of $\mathbf{X}$, as you can easily demonstrate by computing $\mathbf{w}'\mathbf{X}$.

Function: **CompleteSymmetricMatrix(x)**

Input Description: **x** is a vector containing the lower triangular elements of a symmetric matrix.

We frequently work with symmetric matrices, and it is a waste of time to type the redundant elements of the matrix twice. This function allows you to enter the lower triangular elements, and it constructs the full symmetric matrix for you.

Example.
Suppose you wish to input the correlation matrix

$$\mathbf{R} = \begin{bmatrix} 1 & .20 & .30 \\ .20 & 1 & .15 \\ .30 & .15 & 1 \end{bmatrix}$$

You can enter it as

```
> R<-CompleteSymmetricMatrix(c(1,.20,1,.30,.15,1))
> R
     [,1] [,2] [,3]
[1,]  1.0 0.20 0.30
[2,]  0.2 1.00 0.15
[3,]  0.3 0.15 1.00
```

Function: **CompleteCorrelationMatrix(x)**

Input Description: **x** is a vector containing the lower triangular elements of a correlation matrix, *excluding the 1's on the diagonal.*

This function is similar to **CompleteSymmmetricMatrix**. However, since there is no need to enter the 1's on the diagonal, this function allows you to save even more time and omit them.

Example.
Suppose you wish to input the correlation matrix

$$\mathbf{R} = \begin{bmatrix} 1 & .20 & .30 \\ .20 & 1 & .15 \\ .30 & .15 & 1 \end{bmatrix}$$

You can enter it as

```
> R<-CompleteCorrelationMatrix(c(.20,.30,.15))
> R
     [,1] [,2] [,3]
[1,]  1.0 0.20 0.30
[2,]  0.2 1.00 0.15
[3,]  0.3 0.15 1.00
```

Function: **MakeExactData(MeanVector, CovarianceMatrix, n, use.population=FALSE)**

Input Description: **MeanVector** is a $p \times 1$ vector of means for the p variables, **CovarianceMatrix** is a full rank $p \times p$ covariance matrix for the p variables, **n** is the sample size and must be greater than p. If **use.population** is **TRUE**, then the population equivalents (using n instead of $n–1$ as a denominator) are used to calculate sample variances and covariances instead of the normal sample statistics.

This function returns an $n \times p$ data matrix having sample statistics *exactly matching* the requested input specifications.

Example.

We wish 5 observations on 3 variables with sample means exactly equal to 2, 3, and 9, and sample covariance matrix $\begin{bmatrix} 12 & & \\ 4 & 13 & \\ 5 & 6 & 12 \end{bmatrix}$

Using **CompleteSymmetricMatrix** to assist entering the matrix, we have

```
> means <- c(2,3,9)
> covariances <- CompleteSymmetricMatrix(c(12,4,13,5,6,12))
> covariances
     [,1] [,2] [,3]
[1,]   12    4    5
[2,]    4   13    6
[3,]    5    6   12
> X<-MakeExactData(means,covariances,5)
> X
          [,1]       [,2]      [,3]
[1,] -3.641648  1.9305511  8.132055
[2,]  1.111600  0.6393700  3.764068
[3,]  3.303161  4.5955588 13.239266
[4,]  4.503564 -0.6134774  9.685397
```

Verification:

```
> cov(X)
     [,1] [,2] [,3]
```

```
[1,]   12    4    5
[2,]    4   13    6
[3,]    5    6   12
> apply(X,2,mean)
[1] 2 3 9
```

Function: **MultivariateNormalSample(mu,Sigma,n)**

Input Description: **mu** is a $p \times 1$ vector of means for the p variables,
Sigma is a full rank $p \times p$ covariance matrix for the p variables,
n is the sample size and must be greater than p.

This function simulates a sample of size n from a multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

Example.
We construct a sample of size 5 from a population having the same characteristics as the previous example.

```
> mu <- c(2,3,9)
> Sigma <- CompleteSymmetricMatrix(c(12,4,13,5,6,12))
> X<-MultivariateNormalSample(mu,Sigma,5)
> X
          [,1]      [,2]      [,3]
[1,] -1.664675 6.0176893 13.335378
[2,]  3.815069 3.1467965  5.077515
[3,] -5.470887 2.2097950  7.865798
[4,] -3.778831 0.8680326  5.051340
[5,] -5.315544 0.6632740  5.895257
```

Of course, the sample statistics will not match their population values because of random variation.

Function: **sympower(X, power)**

Input Description: **X** is a full rank, symmetric matrix, to be raised to **power.**

This function computes symmetric powers of a symmetric matrix. Symmetric powers $\mathbf{X}^k$ of $\mathbf{X}$ are symmetric matrices that operate like scalars, in that $\mathbf{X}^0 = \mathbf{I}$, $\mathbf{X}^1 = \mathbf{X}$ and $\mathbf{X}^p\mathbf{X}^q = \mathbf{X}^{p+q}$.

Example.

Consider the matrix

$$\mathbf{X} = \begin{bmatrix} 1 & .7 \\ .7 & 2 \end{bmatrix}$$

The "symmetric square root" of $\mathbf{X}$ is computed below:

```
> X<-CompleteSymmetricMatrix(c(1,.7,2))
> X
     [,1] [,2]
[1,]  1.0  0.7
[2,]  0.7  2.0
> Xhalf <- sympower(X,1/2)
> Xhalf
          [,1]      [,2]
[1,] 0.9540533 0.2996371
[2,] 0.2996371 1.3821062
```

We can verify that $\mathbf{X}^{1/2}\mathbf{X}^{1/2} = \mathbf{X}$

```
> Xhalf %*% Xhalf
     [,1] [,2]
[1,]  1.0  0.7
[2,]  0.7  2.0
```

Function: **MakeFactorCorrelationMatrix(F)**

Input Description: **F** is a $p \times m$ factor pattern of full column rank.

This function constructs the correlation matrix exactly corresponding to a common factor pattern $\mathbf{F}$ for m common factors.

Example.

$$\mathbf{F} = \begin{bmatrix} .6 & 0 \\ .5 & 0 \\ .7 & 0 \\ 0 & .6 \\ 0 & .6 \\ 0 & .7 \end{bmatrix}$$

```
> F <- matrix(c(.6,.5,.7,0,0,0,0,0,0,.6,.6,.7),6,2)
> F
     [,1] [,2]
[1,]  0.6  0.0
[2,]  0.5  0.0
[3,]  0.7  0.0
[4,]  0.0  0.6
[5,]  0.0  0.6
[6,]  0.0  0.7
> MakeFactorCorrelationMatrix(F)
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 1.00 0.30 0.42 0.00 0.00 0.00
[2,] 0.30 1.00 0.35 0.00 0.00 0.00
[3,] 0.42 0.35 1.00 0.00 0.00 0.00
[4,] 0.00 0.00 0.00 1.00 0.36 0.42
[5,] 0.00 0.00 0.00 0.36 1.00 0.42
[6,] 0.00 0.00 0.00 0.42 0.42 1.00
```

Function: **`MakeFactorData(F, n)`**

Input Description: **`F`** is a $p \times m$ factor pattern of full column rank, **`n`** is the number of independent observations.

This function simulates a sample of n independent observations from a multivariate normal distribution consonant with zero mean vector and covariance matrix generated by an orthogonal common factor model with factor pattern $\mathbf{F}$.

Function: **`MakeExactFactorData(F, n)`**

Input Description: **`F`** is a $p \times m$ factor pattern of full column rank, **`n`** is the number of independent observations.

This function creates a sample of n observations with exactly zero means and sample covariance matrix exactly equal to that generated by an orthogonal common factor model with factor pattern **F**.