Matrix Operations in SPSS and R

SPSS

```
matrix.
***** define a matrix with the compute command****.

      ***** A is a 2 X 2 matrix********.
compute A={3,2;
          4,0}.

      ***** B is a 2 X 5 matrix*********.
compute B={1,3,0,5,7;
           2,4,3,3,3}.

      ***** C is a 2 X 2 matrix*********.
compute C={0,8;
          9,1}.

***** transpose B******.
compute Btrans=transpos(B).

****** Add *****.
compute AplusC=A+C.

***** multiply****.
compute AB=A*B.

***** inverse*****.
compute inverseA=inv(A).

***** determinant ***.
compute detA=det(A).

*******print the results ********.
print A /title "Matrix A".
print B /title "Matrix B".
print Btrans /title "B transpose".
print AplusC /title "A + C".
print AB /title "AB".
print inverseA /title "A inverse".
print detA /title "Det A".

end matrix.
```

Results

```
Run MATRIX procedure:

Matrix A
  3  2
  4  0

Matrix B
  1  3  0  5  7
  2  4  3  3  3

B transpose

  1  2
  3  4
  0  3
  5  3
  7  3

A + C
   3  10
  13   1

AB
   7  17   6  21  27
   4  12   0  20  28

A inverse
   .0000000000   .2500000000
   .5000000000  -.3750000000

Det A
 -8
```

R

```
> #this method lists elements one column at a time
> #A=matrix(c(3,4,2,0),nrow=2)
>
> #alternatively, list elements one row at a time with this code (I prefer this method)
> A=matrix(c(3,2,4,0), nrow=2,byrow=TRUE)
> A
     [,1] [,2]
[1,]    3    2
[2,]    4    0
>
> B=matrix(c(1,3,0,5,7,2,4,3,3,3),nrow=2,byrow=T)
> B
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    0    5    7
[2,]    2    4    3    3    3
>
> C=matrix(c(0,8,9,1),nrow=2,byrow=TRUE)
> C
     [,1] [,2]
[1,]    0    8
[2,]    9    1
>
> #transpose
> Btrans=t(B)
> Btrans
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    0    3
[4,]    5    3
[5,]    7    3
>
> #adding
> AplusC=A+C
> AplusC
     [,1] [,2]
[1,]    3   10
[2,]   13    1
>
> #multiplying
> AB=A%*%B
> AB
     [,1] [,2] [,3] [,4] [,5]
[1,]    7   17    6   21   27
[2,]    4   12    0   20   28
>
> #inverse
> inverseA=solve(A)
> inverseA
     [,1]   [,2]
[1,]  0.0  0.250
[2,]  0.5 -0.375
>
> #determinant
> detA=det(A)
> detA
[1] -8
```

Computing Some Basic Statistical Values with Matrix Operations in SPSS and R

Matrix Algebra Equations

Mean

$$\bar{\mathbf{X}} = \left(\frac{1}{N}\right)\mathbf{1'X}$$

Centering **X**

$$\mathbf{X} = \mathbf{X}_{\mathbf{raw}} - \mathbf{1}\left(\frac{1}{N}\right)\mathbf{1'X}_{\mathbf{raw}}$$

Covariance matrix **S**

$$\mathbf{S} = \left(\frac{1}{n-1}\right)\mathbf{X'X}$$

where the matrix above called $\mathbf{X}$ is a matrix of X and Y variables with (one column each). In SPSS or R, then, you would want to specify just one matrix that contains both the X and Y variables. The $\mathbf{X}$ matrix here contains deviations scores, as shown in the prior equation.

Correlation Matrix

$$\mathbf{R} = \mathbf{D}^{-1/2}\mathbf{S}\,\mathbf{D}^{-1/2}$$

Where D is the diagonal matrix of variances. In other words, $\mathbf{D}^{-1/2}$ is a diagonal matrix with $1/sd$ for each variable as the diagonal elements.

Regression Coefficient

To obtain the regression coefficient, follow the formula below.

$$\mathbf{b} = \left(\mathbf{X'X}\right)^{-1}\mathbf{X'Y}$$

where $\mathbf{Y}$ is a column vector ($n \times 1$) and $\mathbf{X}$ is an $n \times k$ matrix with the first column as ones and as many additional columns as variables and as many rows as cases ($n \times k + 1$). By using ones in the first column of the matrix, you obtain the intercept as the first coefficient in the resulting matrix. Note: contrary to the covariance example above where $\mathbf{X}$ and $\mathbf{Y}$ were in the same matrix, the regression coefficient formula requires that X and Y variables have separate matrices.

SPSS

```
***** This file computes some basic statistical values using matrix algebra.
matrix.
*compute the mean of X1 and X2.
compute X={-2,-1;
            0,1;
            2,3;
            4,-2;
           -4,-1}.
compute ONE=make(5,1,1).  /* (5,1,1) generates a matrix with 5 rows, 1 column, and elements equal to 1.
compute transONE=transpos(ONE).
compute meanX=transONE*X*(1/5).
print meanX /title "mean of X1 and X2"
    /format="F5.2".
end matrix.

***for some equations, deviations scores are assumed, compute deviations this way.
*assuming means of 3 and 5 for Xraw, matrix X is same as the deviation matrix X used above and below.
*matrix.
*compute Xraw = {1,4;
                3,6;
                5,8;
                7,3;
               -1,4}.
*compute one=make(5,1,1).
*compute meanX=one*transpos(one)*Xraw*(1/5).
*compute X=Xraw - meanX.
*print X /title "X deviation scores".
*end matrix.

**** This section computes a variance-covariance matrix, s, for two variables, X1 and X2.
matrix.
compute X={-2,-1;
            0,1;
            2,3;
            4,-2;
           -4,-1}.
compute Xtrans=transpos(X).
compute Xprod=Xtrans*X.
compute S=(1/(5-1))*Xprod.      /*   1/(5-1) is 1/(N-1), the inverse of df  */
compute invS=inv(S).
print S /title="Covariance Matrix S".
print invS /title="Inverse of S".

*** This section computes the correlation matrix of X1 and X2.
compute rtdiagS=sqrt(diag(S)).  /* take the square root of the diagonal elements of S for standard deviations*/
compute rtD=mdiag(rtdiagS).  /* mdiag gives the full matrix diagonal with 0 off diagonal elements */
compute invrtD=inv(rtD).    /* invrtD is D raised to the negative one-half giving us the reciprocal of the
standard deviations*/
compute R=invrtD*S*invrtD.
print invrtD /title="invrtD: D raised to the negative one-half".
print R  /title="R, the correlation matrix".
end matrix.

***This section computes a simple regression with Y regressed on X. The first column of the X matrix must have
ones to obtain the intercept.

matrix.
** if you need to create an augmented matrix automatically use the following syntax--make sure number of rows in
one matrix is the same as in X.
*compute one=make(5,1,1).
*compute xaug={one,X}.
*print xaug.
*end matrix.

compute X={1,-2;
           1,0;
           1,2;
           1,4;
           1,-4}.
compute Y={-1;1;3;-2;-1}.
compute Xtrans=transpos(X).
compute XtransX=Xtrans*X.
compute invXX=inv(XtransX).
compute XtransY=Xtrans*Y.
compute b=invXX*XtransY.
print b /title="b".
end matrix.
```

Output

```
mean of X1 and X2
   .00   .00

Covariance Matrix S
  10   1
   1   4

Inverse of S
   .1025641026  -.0256410256
  -.0256410256   .2564102564

invrtD: D raised to the negative one-half
   .3162277660   .0000000000
   .0000000000   .5000000000

R, the correlation matrix
   1.000000000    .158113883
    .158113883   1.000000000

(The first element in the B vector below is the intercept. Here it is zero because I used deviations scores for
my X values.)
B
   .0000000000
   .1000000000
```

The SPSS illustrations above use small examples in which the matrices are manually specified, but it is also possible to create a matrix using variables in an existing (and active) data set (this will not be needed for the homework). For example,

```
get file='c:\data.sav'.  *data set would contain variables x1, x2, x3, and x4.
*note: no missing values are allowed, so eliminate them first.
matrix.
get x /variables x1 x2 x3 x4. *this creates a data set with four columns and N rows.
print x.
end matrix.
```

R *(output omitted)*

```
## This file computes some basic statistical values using matrix algebra

## Compute the mean of X1 and X2  (note: matrix names are case sensitive)
> X=matrix(c(-2,0,2,4,-4,-1,1,3,-2,-1), nrow=5)
> ONE=matrix(c(1,1,1,1,1), ncol=1)    # or, alternatively, ONE=rep(1,5,ncol=1) for a column vector
> transONE=t(ONE)
> meanX=transONE%*%X*(1/5)
> "mean of X1 and X2"                  # prints this text
> meanX                               # prints meanX matrix

#for deviations scores of X (assuming means 3 and 5 for Xraw)
#Xraw=matrix(c(1,4,3,6,5,8,7,3,-1,4), nrow=5,byrow=T)
#ONE=rep(1,5,ncol=1)
#meanX=ONE%*%transONE%*%Xraw*(1/5)
#X=Xraw - meanX
#X


## This section computes a variance-covariance matrix, s,  for two variables, X1 and X2.
> X=matrix(c(-2,0,2,4,-4,-1,1,3,-2,-1), ncol=2)
> Xtrans=t(X)
> Xprod=Xtrans%*%X
> S=(1/(5-1))*Xprod                   # 1/(5-1) is 1/(N-1), the inverse of df
> "Covariance Matrix of X"
> S
> invS=solve(S)
> "Inverse of S"
> invS

## This section computes the correlation matrix of X1 and X2
> rtdiagS=sqrt(diag(S))               # take the square root of the diagonal elements of S for standard
deviations
> rtD=diag(rtdiagS)                   # gives the full matrix diagonal with 0 off diagonal elements
> invrtD=solve(rtD)                          # invrtD is D raised to the negative one-half giving us the
reciprocal of the standard deviations
> "invrtD: D raised to the negative one-half"
> invrtD
> R=invrtD%*%S%*%invrtD
> "R, the correlation matrix"
> R

#This section computes a simple regression with Y regressed on X. The first column of the X matrix must have
ones to obtain the intercept.
##This section computes a simple regression with Y regressed on X. The first column of the X matrix must have
ones to obtain the intercept.
#if you need to create an augmented matrix automatically use the following code
#X=matrix(c(-2,0,2,4,-4), nrow=5)
#ONE=rep(1,5,ncol=1)
#ONE
#Xaug=cbind(ONE,X)
#Xaug

> X=matrix(c(1,1,1,1,1,-2,0,2,4,-4), nrow=5)
> Y=matrix(c(-1,1,3,-2,-1), nrow=5)
> Xtrans=t(X)
> XtransX=Xtrans%*%X
> invXX=solve(XtransX)
> XtransY=Xtrans%*%Y
> b=invXX%*%XtransY
> "B0 and B1"
> b
```

Creating a matrix using variables in an existing (and active) data frame in R (this will not be needed for the homework).

```
library(foreign)
d = read.spss("c:/data.sav", to.data.frame=TRUE)

library(dplyr)
d2 = d %>% select('x1','x2','x3','x4')

#listwise deletion
d2 = d2[complete.cases(d[,c('x1','x2','x3','x4')]),]

data.matrix(d2)
```