EdPsych 584/Psych 594
Applied Multivariate Analysis
C.J. Anderson

Useful commands in SAS PROC IML (last revised 2/12/2015).

Let

- $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{U}$, and $\boldsymbol{X}$ be rectangular matrices.
- $\boldsymbol{D}$ a diagonal matrix.
- $\boldsymbol{S}$, $\boldsymbol{W}$ and $\boldsymbol{R}$ be square symmetric matrices.
- $\boldsymbol{y}$, $\boldsymbol{z}$, $\boldsymbol{L}$ be vectors.
- c, n, p be scalars (constants).

Command	Example	Explanation/Desciription
$*$	$C = A * B;$	Matrix multiplication
$-$	$C = A - B;$	Subtraction
$+$	$C = A + B;$	Matrix or vector addition
/	$C = A/n;$	Divide elements of A by constant n
$-$	$C = -A;$	Reverses the sign of the elements
##	$C = A\#\#2$	Take the power of elements of A (element-wise)
sqrt	$C = sqrt(A);$	Take the square root of the elements of a matrix
‘	$C = A‘;$	Transpose operator
nrow()	$n =$nrow($\boldsymbol{X}$);	Returns the number of rows of matrix $\boldsymbol{X}$
ncol()	$p =$ncol($\boldsymbol{X}$);	Returns the number of columens of matrix $\boldsymbol{X}$
J(n,p,c)	$\boldsymbol{z} = J(n, p, c);$	Creates a matrix or vector with n rows, p columns and all elements equal to c. This is useful for creating a vector of ones (e.g., an $(n \times 1)$ vector of ones, one= $J(n, 1, 1)$).
I(n)	Ident=I(n);	Creates an identity matrix

<table>
<tr><th>Command</th><th>Example</th><th>Explanation/Desciription</th></tr>
<tr><td>||</td><td>$C = A||B$;</td><td>Horizontally concatenation of matrices</td></tr>
<tr><td>//</td><td>$C = A//B$;</td><td>Vertical concatenation of matrices</td></tr>
<tr><td>diag()</td><td>D =diag($\boldsymbol{S}$);

D =diag($\boldsymbol{z}$);</td><td>This creates a diagonal matrix where the argument is a square matrix or a vector (vector argument)</td></tr>
<tr><td>vecdiag()</td><td>$\boldsymbol{z}$ =vecdiag($\boldsymbol{S}$);</td><td>Function creates a column vector whose elements are the diagonal of the square matrix</td></tr>
<tr><td>[ ]</td><td>$y3 = \boldsymbol{X}[1:n,3]$;</td><td>Creates a vector $y3$ that is the 3rd column of matrix $\boldsymbol{X}$. The square brackets “[” and “]” can be used to indicate specific elements of a vector or matrix. The colon : is used to indicater series (e.g., Take rows 1 through n).</td></tr>
<tr><td>det()</td><td>c =det(A);</td><td>Computes the determinant of matrix</td></tr>
<tr><td>inv()</td><td>B = inv(A);</td><td>Finds the inverse of a square symmetric matrix.</td></tr>
<tr><td>eigen()</td><td>call eigen(L, U, S);</td><td>Finds of eigenvalues and vectors of the 3rd argument (e.g., S) and puts the eigenvalues into L and eigenvectors (as columns) in U.</td></tr>
<tr><td>svd()</td><td>call svd(U,q,V,A)</td><td>Finds the singular values and vectors of a matrix A ($m \times n$) where U $m \times n$) contain left singular vectors, U ($n \times n$) contains right singular vectors, and q ($n \times 1$) contains the singular values. Note that $m \geq n$.</td></tr>
<tr><td>use mydata</td><td>use mydata;</td><td>This statement indicates what sasdata set you want to access</td></tr>
<tr><td>read</td><td>read all var{ test1 test2 test3 test4} into X;</td><td>This statement reads data from an open sasdata set into a matrix (or vector)</td></tr>
<tr><td>print stuff</td><td>print X z;</td><td>Prints these on the same line of output (if possible).</td></tr>
<tr><td></td><td>print X, z;</td><td>The comma indicates to go to next line (i.e., print X and then on new line print z.</td></tr>
<tr><td></td><td>print ’text’ X</td><td>You can add text to the print command.</td></tr>
<tr><td></td><td>print x[format=5.3]</td><td>You can also indicate the format for printing numerical values. Note that ”format=5.3” means that there will be 5 numbers and at most 3 decimal places. You can play with the numbers to get what “looks” best.</td></tr>
</table>

An example of a module:

```
/* A module that compute the mean vector, SSCP matrix, covariance
   matrix and correlation matrix of an (n x p) matrix of observations */

start samplestats(X,Xbar,W,S,R,n);
  n = nrow(X);
  one = J(n,1);
  Xbar = X`*one/n;
  W =  (X - one*Xbar`)` * (X - one*Xbar`);
  S = W/(n-1);
  Dsqrt = sqrt(diag(S));
  R = inv(Dsqrt)*S*inv(Dsqrt);
Finish samplestats;

run samplestats(X,Xbar,W,S,R,n);
print 'Sample statistics for whole sample:',
      n Xbar, S, R;
```