```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  MATLAB DEMO 5-30-2013
%%  Basic Matlab commands
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%
%%  diary command
%%%%%%%%%%%%%%%%%%%%%%%%%

help diary
 DIARY Save text of MATLAB session.
    DIARY FILENAME causes a copy of all subsequent command window input
    and most of the resulting command window output to be appended to the
    named file.  If no file is specified, the file 'diary' is used.

    DIARY OFF suspends it.
    DIARY ON turns it back on.
    DIARY, by itself, toggles the diary state.

    Use the functional form of DIARY, such as DIARY('file'),
    when the file name is stored in a string.

    See also SAVE.

%%%%%%%%%%%%%%%%%%%%%%%%%
%% Inputting and manipulating matrices
%%%%%%%%%%%%%%%%%%%%%%%%%

A=[1 23 3;5 6 7;2 7 9;-3 5.6 12]

A =

    1.0000   23.0000    3.0000
    5.0000    6.0000    7.0000
    2.0000    7.0000    9.0000
   -3.0000    5.6000   12.0000

A=[1 23 3;5 6 7;2 7 9;-3 5.6 12];
B=[1,2,3,4;9,10,11,12]

B =

     1     2     3     4
     9    10    11    12

A*B
{??? Error using ==> mtimes
Inner matrix dimensions must agree.
}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  A*B is not defined since A is 4x3 and B is 2x4
%%
%%  Is B*A defined?  Yes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
B*A

ans =

    5.0000   78.4000   92.0000
   45.0000  411.2000  340.0000

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  We can pull out entries or portions of matrices
%%
%%  Find an individual entry
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

A(2,3)

ans =

     7

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Picking out an entire row or column
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

A(2,1:3)

ans =

     5     6     7

A(2,:)

ans =

     5     6     7

A(2:4,2:3)

ans =

    6.0000    7.0000
    7.0000    9.0000
    5.6000   12.0000

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Can add rows or columns to matrices
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

B

B =

     1     2     3     4
     9    10    11    12
```

```
B={B;15,16,17]
??? B={B;15,16,17]
                 |
{Error: Unbalanced or unexpected parenthesis or bracket.
}
B={B;15,16,17,18]
??? B={B;15,16,17,18]
                   |
{Error: Unbalanced or unexpected parenthesis or bracket.
}
B=[B;15 16 17 18]

B =

     1     2     3     4
     9    10    11    12
    15    16    17    18

B=[B;15,16,17,18]

B =

     1     2     3     4
     9    10    11    12
    15    16    17    18
    15    16    17    18

B=B(1:3,:)

B =

     1     2     3     4
     9    10    11    12
    15    16    17    18

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Now the product A*B is defined
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

A*B

ans =

  253.0000  280.0000  307.0000  334.0000
  164.0000  182.0000  200.0000  218.0000
  200.0000  218.0000  236.0000  254.0000
  227.4000  242.0000  256.6000  271.2000

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Lets add a column to B
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

B=[B [20; 21; 22]]
```

```
B =

     1     2     3     4    20
     9    10    11    12    21
    15    16    17    18    22

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Let's do Gauss-Jordan using MATLAB
%%
%%  This is #16 on page 61
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

C = [6/5 -1 12 -2/3;-1 0 0 5;0 2 -1 6]

C =

    1.2000   -1.0000   12.0000   -0.6667
   -1.0000         0         0    5.0000
         0    2.0000   -1.0000    6.0000

help format
 FORMAT Set output format.
    FORMAT with no inputs sets the output format to the default
appropriate
    for the class of the variable. For float variables, the default is
    FORMAT SHORT.

    FORMAT does not affect how MATLAB computations are done. Computations
    on float variables, namely single or double, are done in appropriate
    floating point precision, no matter how those variables are
displayed.
    Computations on integer variables are done natively in integer.
Integer
    variables are always displayed to the appropriate number of digits
for
    the class, for example, 3 digits to display the INT8 range -128:127.
    FORMAT SHORT and LONG do not affect the display of integer variables.

    FORMAT may be used to switch between different output display formats
    of all float variables as follows:
      FORMAT SHORT     Scaled fixed point format with 5 digits.
      FORMAT LONG      Scaled fixed point format with 15 digits for
double
                       and 7 digits for single.
      FORMAT SHORT E   Floating point format with 5 digits.
      FORMAT LONG E    Floating point format with 15 digits for double
and
                       7 digits for single.
      FORMAT SHORT G   Best of fixed or floating point format with 5
                       digits.
      FORMAT LONG G    Best of fixed or floating point format with 15
                       digits for double and 7 digits for single.
      FORMAT SHORT ENG Engineering format that has at least 5 digits
```

```
                    and a power that is a multiple of three
      FORMAT LONG ENG  Engineering format that has exactly 16 significant
                    digits and a power that is a multiple of three.

    FORMAT may be used to switch between different output display formats
    of all numeric variables as follows:
      FORMAT HEX     Hexadecimal format.
      FORMAT +       The symbols +, - and blank are printed
                     for positive, negative and zero elements.
                     Imaginary parts are ignored.
      FORMAT BANK    Fixed format for dollars and cents.
      FORMAT RAT     Approximation by ratio of small integers.  Numbers
                     with a large numerator or large denominator are
                     replaced by *.

    FORMAT may be used to affect the spacing in the display of all
    variables as follows:
      FORMAT COMPACT Suppresses extra line-feeds.
      FORMAT LOOSE   Puts the extra line-feeds back in.

    Example:
       format short, pi, single(pi)
    displays both double and single pi with 5 digits as 3.1416 while
       format long, pi, single(pi)
    displays pi as 3.141592653589793 and single(pi) as 3.1415927.

       format, intmax('uint64'), realmax
    shows these values as 18446744073709551615 and 1.7977e+308 while
       format hex, intmax('uint64'), realmax
    shows them as ffffffffffffffff and 7fefffffffffffff respectively.
    The HEX display corresponds to the internal representation of the
value
    and is not the same as the hexadecimal notation in the C programming
    language.

    See also DISP, DISPLAY, ISNUMERIC, ISFLOAT, ISINTEGER.

    Reference page in Help browser
       doc format

format rat
C

C =

       6/5           -1            12            -2/3
      -1              0             0             5
       0              2            -1             6

%%  first row operation:  Want a 1 in row 1 column 1.

C=[(5/6)*C(1,:);C(2,:);C(3,:)]

C =
```

```
       1             -5/6           10             -5/9
      -1              0              0              5
       0              2             -1              6

%%  Second row operation:  Want a 0 in row 2 column 1.

C=[C(1,:);1*C(1,:)+C(2,:);C(3,:)]

C =

       1             -5/6           10             -5/9
       0             -5/6           10             40/9
       0              2             -1              6

%%  Third row operation:  Want 1 in row 2 column 2

C=[C(1,:);(-6/5)*C(2,:);C(3,:)]

C =

       1             -5/6           10             -5/9
       0              1            -12            -16/3
       0              2             -1              6

%%  Fourth and fifth row operatinos:  Zeros in row 1 and row 3, column 2.

C=[(5/6)*C(2,:)+C(1,:);C(2,:);C(3,:)]

C =

       1              0              0             -5
       0              1            -12            -16/3
       0              2             -1              6

C=[C(1,:);C(2,:);-2*C(2,:)+C(3,:)]

C =

       1              0              0             -5
       0              1            -12            -16/3
       0              0             23             50/3

%%  Last row operations:  Get a 1 in row 3 column 3, and a 0 in row 2,
column 3.

C=[C(1,:);C(2,:);(1/23)*C(3,:)]

C =

       1              0              0             -5
       0              1            -12            -16/3
       0              0              1             50/69
```

```
C=[C(1,:);12*C(3,:)+C(2,:);C(3,:)]

C =

       1              0              0             -5
       0              1              0            232/69
       0              0              1             50/69

%%  Solution to the system is x=-5, y=232/69, z=50/69

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  All these calculations can be done in one step using rref.
%%  Stands for Row Reduced Echelon Form
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

help rref
 RREF   Reduced row echelon form.
    R = RREF(A) produces the reduced row echelon form of A.

    [R,jb] = RREF(A) also returns a vector, jb, so that:
        r = length(jb) is this algorithm's idea of the rank of A,
        x(jb) are the bound variables in a linear system, Ax = b,
        A(:,jb) is a basis for the range of A,
        R(1:r,jb) is the r-by-r identity matrix.

    [R,jb] = RREF(A,TOL) uses the given tolerance in the rank tests.

    Roundoff errors may cause this algorithm to compute a different
    value for the rank than RANK, ORTH and NULL.

    Class support for input A:
       float: double, single

    See also RANK, ORTH, NULL, QR, SVD.

    Overloaded methods:
       eml.rref

    Reference page in Help browser
       doc rref

C

C =

       1              0              0             -5
       0              1              0            232/69
       0              0              1             50/69

C = [6/5 -1 12 -2/3;-1 0 0 5;0 2 -1 6]

C =

       6/5           -1             12            -2/3
```

```
      -1             0              0              5
       0             2             -1              6

rref(C)

ans =

       1             0              0             -5
       0             1              0            232/69
       0             0              1             50/69

diary off
```