Topics: Two-dimensional array—matrix, file input/output (I/O)

Reading (ML): Sec 2.3, 2.7, review Sec 2.2, 2.4

Matrix creation, access, manipulation

```
m = [1 2 3 4; 5 6 7 8]  % 2-by-4 matrix
[nr,nc] = size(m)
m = [m; zeros(1,nc)]
m = [m m]
m = [m; m]
v = 1:6
newm = [m v']
newm = newm'
m1 = rand(4,3)  % 4-by-3 random matrix (uniform dist.)
tmp = m1(3,2)   % cell in 3rd row, 2nd column
tmp = m1(3:4,:)
tmp = m1(:,2)
tmp = m1([1 4],:)
tmp = m1(:, [1 3])
tmp = m1([1 4],[1 3])
load matrix.dat  % an ASCII data file called matrix.dat
load mmatrix  % a MATLAB data file called mmatrix.mat
help save
```

Processing data in a matrix

Write a program segment to sum all the values in matrix **m** without using any MATLAB predefined functions (other than `size`) or vectorized code.

```
[nr,nc] = size(m);

% initialize variable for accumulating the sum


% sum one element at a time

  % move along each row
  for i = 1:nr

    % move along each column
    for j = 1:nc


    end
  end
```

Pattern for traversing a matrix

```
for i = 1:nr
  for j = 1:nc
    % do something
  end
end
```

Note: Need to first assign values to `nr` (no. of rows), `nc` (no. of columns)

Example

Calculate the sum of the diagonal elements in a given square matrix **sm**.

```
for i = 1:nr
  for j = 1:nc
    % sum only the diagonal elements




  end
end
```

MATLAB predefined functions

Below are some useful functions that work on matrices. These functions "operate" on each column of the matrix and return a *row* vector.

`sum`, `mean`, `max`, `min`