Topics: Two-dimensional array, cont'd

Reading (ML): Sec 2.1–2.3, 2.7 (exclude matrix computation), Sec 6.3

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 1

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`

Example 2

Elevations of a rectangular area are stored in a matrix **elev**. Each cell contains the elevation at one location (x-y coordinate pair). All measurements in meter.

```
y=3       23   25   26   27
  2       23   23   24   26
  1       22   23   24   25

     x=    1    2    3    4
```

Print the elevation at location (x=3m,y=1m).

What if the x range is [110,140] and y range is [110,130]?

```
y=130     23   25   26   27
  120     23   23   24   26
  110     22   23   24   25

     x=   110  120  130  140
```

Print the elevation at location (x=130m,y=110m).

Arithmetic and logical operations

```
elev = rand(4,3)  % elevations on a map
elev = 8*elev
elev = elev + 10
elev > 16         % returns a logical array

% 1-d examples
  vec = elev(1,:)      % 1st row of matrix elev

  L = vec>16           % logical array indicating result from vec>16
  vec16 = vec(L)       % extract just the cells with values > 16

  vec16 = vec(vec>16)  % combine last two statements in one

  I = find(vec>16)     % get the indices where vec>16
  vec16 = vec(I)       % extract just the cells with values > 16


% 2-d examples
  L16 = elev>16            % logical array (matrix)
  elev16 = elev(elev>16)   % VECTOR!!!

  % extract values>16 and put back in matrix form
  [ri,ci] = find(elev>16)  % ri stores row index
                           % ci stores col index
  for i = 1:length(ri)
    elev16(ri(i),ci(i)) = elev(ri(i),ci(i));
  end
  elev16
```