Today’s Lecture:
- 2-d array—matrix
- Function & subfunction
- Details on for-loop (see blecture)

Announcements:
- Friday: lab session in Upson B7
- Assignment 1b due Tuesday 11:59pm
- Test 1 on Thursday in class; review on Tuesday.
- My Wednesday office hr on 9/16 moved to 10-11:30am
An array is a **named** collection of **like** data organized into rows and columns.

A 2-d array is a table, called a **matrix**.

Two **indices** identify the position of a value in a matrix, e.g.,

\[ \text{mat}(r,c) \]

refers to component in row \( r \), column \( c \) of matrix \( \text{mat} \).

Array index starts at \( 1 \).

**Rectangular**: all rows have the same \# of columns.
% What will M be?
M = [ones(1,3); 1:4]

A
  1 1 1 0
  1 2 3 4

B
  1 1 1
  1 2 3

C  Error – M not created
What will $\mathbf{A}$ be?

\[
\mathbf{A} = \begin{bmatrix} 0 & 0 \\ \end{bmatrix}
\]

\[
\mathbf{A} = [\mathbf{A}' \; \text{ones}(2,1)]
\]

\[
\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \mathbf{A} & \mathbf{A} \\ \end{bmatrix}
\]
Example: minimum value in a matrix

function val = minInMatrix(M)
% val is the smallest value in matrix M
minInMatrix.m
Pattern for traversing a matrix $M$

\[
[nr, nc] = \text{size}(M) \\
\text{for } r = 1:nr \\
\quad \% \text{At row } r \\
\quad \text{for } c = 1:nc \\
\quad \quad \% \text{At column } c \text{ (in row } r) \\
\quad \quad \% \\
\quad \quad \% \text{Do something with } M(r,c) \ldots \\
\quad \text{end} \\
\text{end}
\]
Matrix example: Random Web

- N web pages can be represented by an N-by-N Link Array $A$.
- $A(i,j)$ is 1 if there is a link on webpage $j$ to webpage $i$
- Generate a random link array and display the connectivity:
  - There is no link from a page to itself
  - If $i \neq j$ then $A(i,j) = 1$ with probability $\frac{1}{1+|i-j|}$
    There is more likely to be a link if $i$ is close to $j$
function A = RandomLinks(n)
% A is n-by-n matrix of 1s and 0s
% representing n webpages

A = zeros(n,n);
for i=1:n
    for j=1:n
        r = rand(1);
        if i~=j && r<= 1/(1 + abs(i-j));
            A(i,j) = 1;
        end
    end
end
Random web

N = 20
Represent the web pages graphically…

100 Web pages arranged in a circle. Next display the links….
Represent the web pages graphically...

Bidirectional links are blue. Unidirectional link is black as it leaves page j, red when it arrives at page i.

See ShowRandomLinks.m
for i = 1:n
    for j = 1:n
        if A(i,j) == 1 && A(j,i) == 1
            % Blue
            else if A(i,j) == 1
                % Black
                j \rightarrow mid
                mid \rightarrow i
    end
end

Is there another way? See ShowRandomLinks.m
Local minimum in a neighborhood

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>-1</th>
<th>.5</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-3</td>
<td>8.5</td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>81</td>
<td>.5</td>
<td>7</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Component (2,3)

Neighborhood of component (2,3)
Accessing a submatrix

Neighborhood of component (2,3)

\[ M(2,3) \]

\[ M(1:3, 2:4) \]
Local minimum in a neighborhood

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>-1</th>
<th>.5</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-3</td>
<td>8.5</td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>81</td>
<td>.5</td>
<td>7</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Component (3,5)

Neighborhood of component (3,5)
Local minimum in a neighborhood

- Write a function `minInNeighborhood`
- Input parameters:
  - \( M \): matrix of numeric values
  - \( \text{loc} \): location of the middle of the neighborhood
    - \( \text{loc}(1), \text{loc}(2) \) are the row, column numbers
- Output parameter: `minVal`
  - The minimum value of the neighborhood
Ask yourself questions!

- Can you find the min of a (sub)matrix?
  - Yes! Our function  \( \text{minInMatrix}(A) \)

- Given the indices \( r, c \) (representing element \( M(r,c) \)), is it easy to define the neighborhood?
  - Yes, for the general case the neighborhood is \( M(r-1:r+1, c-1:c+1) \)
  - But need to deal with the “border cases”
Local minimum in a neighborhood

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>0.5</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-1</td>
<td>0.5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>-3</td>
<td>8.5</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>52</td>
<td>81</td>
<td>0.5</td>
<td>7</td>
<td>2</td>
</tr>
</tbody>
</table>

Component (3,5)

Want to be able to use the general case, \( M(r-1:r+1,c-1:c+1) \)
Local minimum in a neighborhood

\[
\begin{array}{cccc}
2 & -1 & .5 & 0 & 1 \\
3 & 8 & 6 & 7 & 7 \\
5 & -3 & 8.5 & 9 & 10 \\
52 & 81 & .5 & 7 & 2 \\
\end{array}
\]

Want to be able to use the general case, \( m(r-1:r+1,c-1:c+1) \)
Local minimum in a neighborhood

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>-1</td>
<td>.5</td>
<td>0</td>
<td>1</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>-3</td>
<td>8.5</td>
<td>9</td>
<td>10</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>52</td>
<td>81</td>
<td>.5</td>
<td>7</td>
<td>2</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
</tbody>
</table>

Want to be able to use the general case, \( m(r-1:r+1, c-1:c+1) \)

Note: This is an exercise on manipulating a matrix. Method not suitable for a large matrix!
minInNeighborhood.m
minInNeighborhoodV2.m
minInNeighborhoodV3.m
Subfunction

- There can be more than one function in an M-file
- top function is the main function and has the name of the file
- remaining functions are subfunctions, accessible only by the functions in the same m-file
- Each (sub)function in the file begins with a function header
- Keyword end is not necessary at the end of a (sub)function
What will be displayed when you run the following script?

```matlab
for k = 4:6
    disp(k)
k = 9;
    disp(k)
end
```

- 4
- 9

or

- 4
- 4

or

Something else …
for \( k = 4:6 \) 
\[
\begin{align*}
\text{disp}(k) \\
k &= 9; \\
\text{disp}(k)
\end{align*}
\]
end

\textbf{Not} a condition (boolean expression) that checks whether \( k \leq 6 \).

It is an expression that specifies values:

\[4 \quad 5 \quad 6\]