Administrivia
Homework #12 is due Friday, Jan. 13.
Final examination will be held Jan. 20, Friday 8am-11am in A-0414.
To refer to multiple elements of an array, use the colon operator to specify a range of the form start:end.
- A(1:3,2) first 3 rows, 2nd column
- A(3,:) all columns in 3rd row

plot https://www.mathworks.com/help/matlab/ref/plot.html
plot(Y) creates a 2-D line plot of the data in Y versus the index of each value. If Y is a matrix, then the plot function plots the columns of Y versus their row number. The x-axis scale ranges from 1 to the number of rows in Y.
plot(X,Y) creates a 2-D line plot of the data in Y versus the corresponding values in X.
Warmup Questions
Which of the following could produce this plot?

A \( x = \text{rand}(10000,1) \);
B \( x = \text{randi}(10000,1) \);
C \( x = \text{randn}(10000,1) \);

\[ \text{plot}(x,'.') \];
Question #1

Which of the following could produce this plot?

A \[ x = \text{rand}(10000,1); \]
B \[ x = \text{randi}(10000,1); \]
C \[ x = \text{randn}(10000,1); \]

\[
\star \\
\text{plot}(x,'.');
\]
\[ \mathbf{A}\mathbf{x} = \mathbf{b} \]

Which is the preferred way to solve this matrix–vector equation?

A. \( \mathbf{A}\mathbf{x} = \text{inv}( \mathbf{A} ) \times \mathbf{b} \);
B. \( \mathbf{A}\mathbf{x} = \mathbf{A} \backslash \mathbf{b} \);
C. \( \mathbf{A}\mathbf{x} = \text{inv}( \mathbf{A} ) \times \mathbf{b} \);
D. \( \mathbf{A}\mathbf{x} = \mathbf{A} \div \mathbf{b} \);
Which is the preferred way to solve this matrix–vector equation?

A \* x = b

Why not this one? If A is a square matrix, A \ b is roughly equal to inv(A) * b, but MATLAB processes A \ b differently and more robustly.

https://www.mathworks.com/help/matlab/ref/mldivide.html
https://www.mathworks.com/help/matlab/ref/inv.html

B \* x = A \ b;
C \* x = inv(A) \* b;
D \* x = A / b;
A = \[ 5 \ 4 \ 1 \ -2 \ 2 \ ];
B = \[ 5 \ 4 \ 1 \ -2 \ 2 \ ];

Are A and B equal in value?

A Yes
B No
Question #3

\[
A = \begin{bmatrix}
5 & 4 & 1 & -2 & 2 \\
5 & 4 & 1 & -2 & 2 \\
\end{bmatrix};
\]
\[
B = \begin{bmatrix}
5 & 4 & 1 & -2 & 2 \\
5 & 4 & 1 & -2 & 2 \\
\end{bmatrix};
\]

Are \( A \) and \( B \) equal in value?

A Yes
B No ⭐
Example: Brexit polling

```
poll = csvread('brexit.csv');
% poll is a matrix.
% In matlab, you can use
% poll = importdata('brexit.csv');
% Then change below poll to be poll.data

plot( poll(:,2) );
plot( poll(:,3) );
% oh no! our plotted data disappeared!
```
Example: Brexit polling

```matlab
poll = csvread('brexit.csv');
hold on;  % make plots persistent until closed
plot( poll(:,2) );
plot( poll(:,3) );
plot( poll(:,4) );
```
n = numel(poll(:,2));

mean_r = mean( poll(:,2) ) * ones( n+1,1 );
stddev_r = std( poll(:,2) );
std_rp = mean_r+stddev_r;
std_rm = mean_r-stddev_r;
hold on
plot( poll(:,2), 'ro' );
plot( 0:n,mean_r, 'r-' );
plot( 0:n,std_rp, 'r--' );
plot( 0:n,std_rm, 'r--' );
n = numel(poll(:,2));
mean_r = rolling_mean( poll(:,2)', 25 );
stddev_r = rolling_std( poll(:,2)', 25 );
std_rp = mean_r+stddev_r;
std_rm = mean_r-stdev_r;
hold on
plot( poll(:,2), 'ro' );
plot( 0:n-1,mean_r, 'r-' );
plot( 0:n-1,std_rp, 'r--' );
plot( 0:n-1,std_rm, 'r--' );
Many operations are available:

- **mean (average), median, std**
- **max, min, range**
- **iqr (interquartile range), corrcoef** (the correlation coefficient of two random variables is a measure of their linear dependence) (not yet supported in Octave but supported by [https://octave.sourceforge.io/nan/function/corrcoef.html](https://octave.sourceforge.io/nan/function/corrcoef.html))
- **sort**
- **boxplot, hist**
Often we would like to fit a set of data to an equation.

We can then interpolate or extrapolate.

- interpolate: to estimate a value within two known values in a sequence of values.
- extrapolate: to infer something that is not explicitly stated from existing information.

This is called curve fitting or regression.
The simplest form of fitting is to a polynomial:

\[ f(x) = a_1 x^3 + a_2 x^2 + a_3 x + a_4 \]

(Note that the numbering is a bit odd!)

But first, we need to see how MATLAB represents polynomials.
Polynomials

\[ a_1x^3 + a_2x^2 + a_3x + a_4 \]

\[
\begin{bmatrix}
a_1 & a_2 & a_3 & a_4 \\
\end{bmatrix}
\]

\[ x^3 - 2x^2 + x + 8 \]

\[
\begin{bmatrix}
1 & -2 & 1 & 8 \\
\end{bmatrix}
\]
Polynomials

How would we write such an operation?

\[
\begin{bmatrix}
1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1
\end{bmatrix}
\]

\[x^3 + 2x^2 + 2x + 1\]
How can we evaluate a polynomial stored as an array?

\[ f(x) = x^3 + 2x^2 + 2x + 1 \]

\[ f(2) = 2^3 + 2 \cdot 2^2 + 2 \cdot 2 + 1 = 8 + 8 + 4 + 1 = 21 \]

`polyval([1 2 2 1], 2)`

`polyval(p,x)` returns the value of a polynomial of degree \( n \) evaluated at \( x \) where \( p \) is a vector of length \( n+1 \).
Example: Data fitting

\[ x = \text{linspace}( -1, 1, 11 ); \]
% linspace(x1,x2,n) generates n points
% The spacing between the points is \((x2-x1)/(n-1)\).

\[ y = [ 0.038 \ 0.058 \ 0.1 \ 0.2 \ 0.5 \ 1 \ 0.5 \ 0.2 \ 0.1 \ 0.058 \ 0.038 ]; \]

\[ \text{coefs} = \text{polyfit}( x,y,2 ); \]
\[ \text{yfit} = \text{polyval}( \text{coefs},x ); \]

\[ \text{plot}( x,y,'.', x,yfit,'-' ); \]
Example: Data fitting

```matlab
x = linspace( -1,1,11 );
y = [ 0.038 0.058 0.1 0.2 0.5 1 0.5 0.2 0.1 0.058 0.038 ];

coefs = polyfit( x,y,10 );
xfit = linspace( -1.5,1.5,101 );
yfit = polyval( coefs,xfit );

plot( x,y,'.', xfit,yfit,'-' );
ylim( [-1 1] );
```
Example: Data fitting

```matlab
x = linspace( 0,1,6 );
y = [ 1 0.5 0.2 0.1 0.058 0.038 ];

coefs = polyfit( x,y,3 );
yfit = polyval( coefs,x );

plot( x,y,'.', x,yfit,'-' );
```
Example: Brexit polling
Example: Brexit polling

```matlab
poll = csvread('brexit.csv');
% poll is a matrix.
% In matlab, you can use
% poll = importdata('brexit.csv');
% Then change below poll to be poll.data

n = numel(poll(:,3));
mean_l = rolling_mean( poll(:,3)', 25 );

fit_poly_l = polyfit( 13:167,mean_l(13:167),19 );
poly_l = polyval( fit_poly_l,1:n );

hold on
plot( poll(:,3), 'ro' );
plot( 1:n,mean_l, 'r-' );
plot( 1:n,poly_l, 'r:' );
```
Other equations are possible besides polynomials:

- See the “Nonlinear Least-Squares Curve Fitting in the Optimization Toolbox” for more information.
Example: Data fitting

```matlab
x = linspace( -2*pi,2*pi,21 );
y = sin( x );

figure; hold on;
plot( x,y,'.' );
for i = 2:9
    coefs = polyfit( x,y,i );
    xfit = linspace( -2*pi,2*pi,101 );
    yfit = polyval( coefs,xfit );
    plot( xfit,yfit,'-' );
end
```
Reminders

- Homework #12 is due Friday, Jan. 13.
- Final examination will be held Jan. 20, Friday 8am-11am in A-0414.