CALIFORNIA STATE UNIVERSITY
FULLERTON™

CPSC-440 Computer System Architecture

MATLAB Review

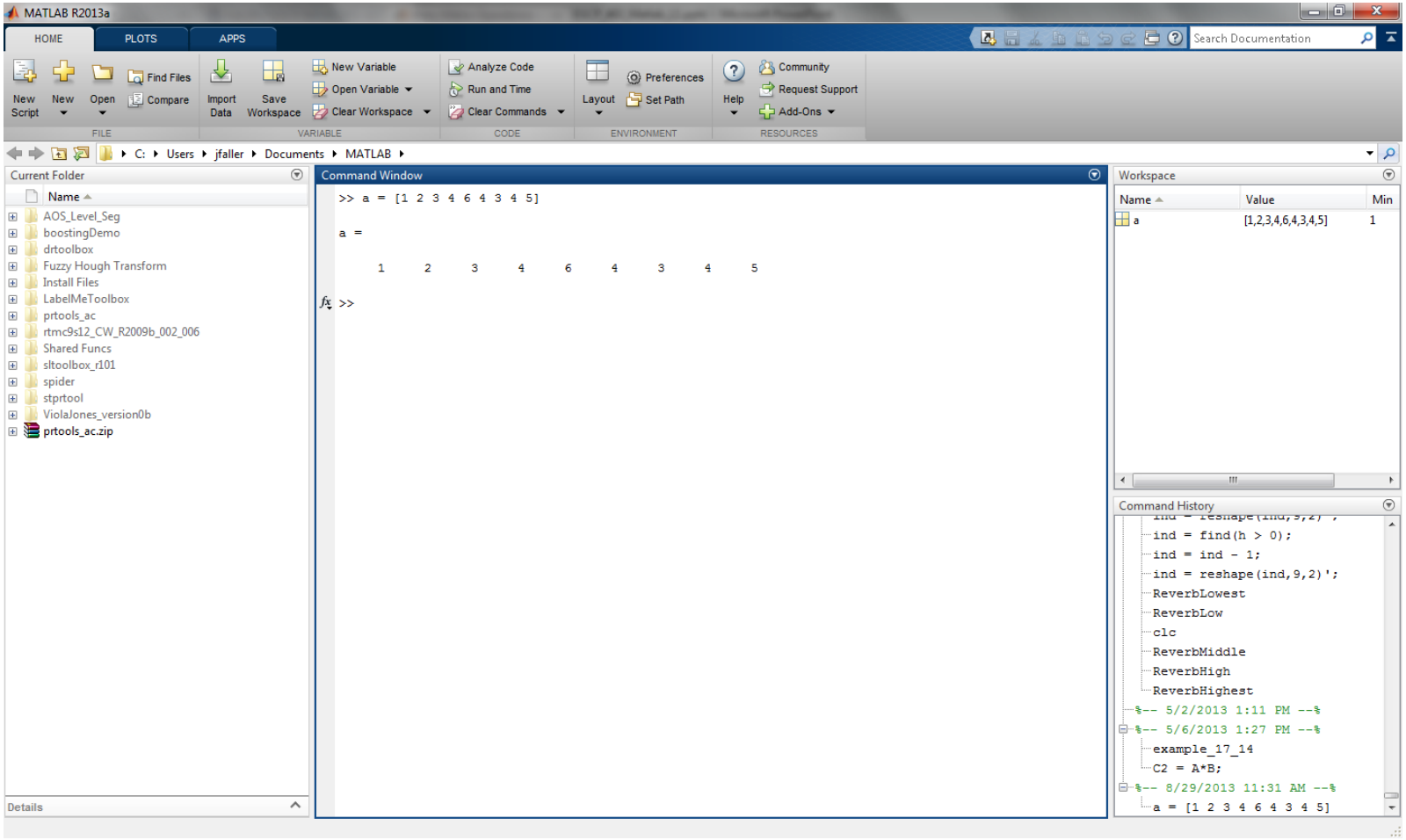
Matlab

- Is a numerical computing environment and 4th generation programming language
- Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran

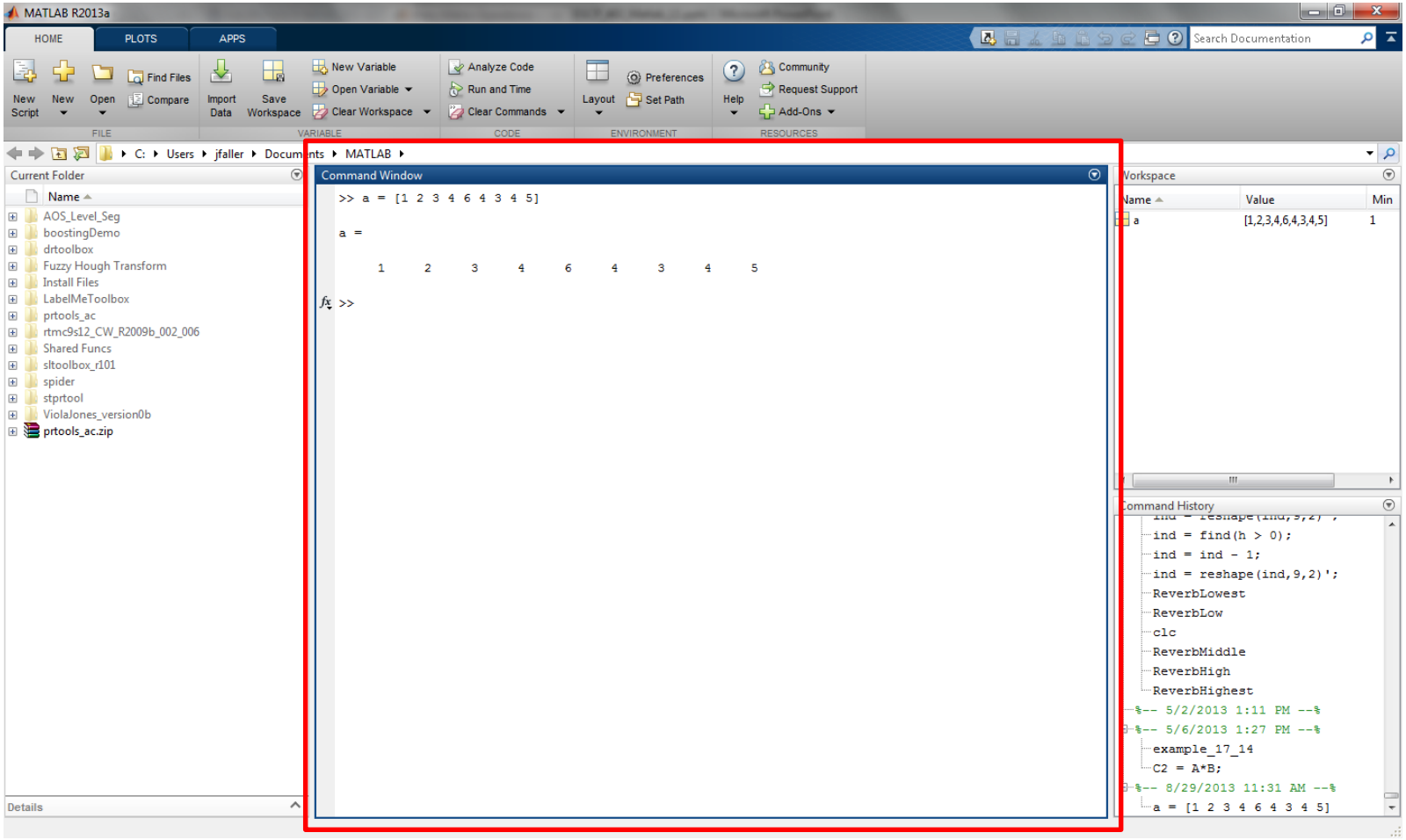
Free Matlab for Students

- Available at CSUF IT website:
  - http://www.fullerton.edu/it/students/software/matlab/

Matlab Default View

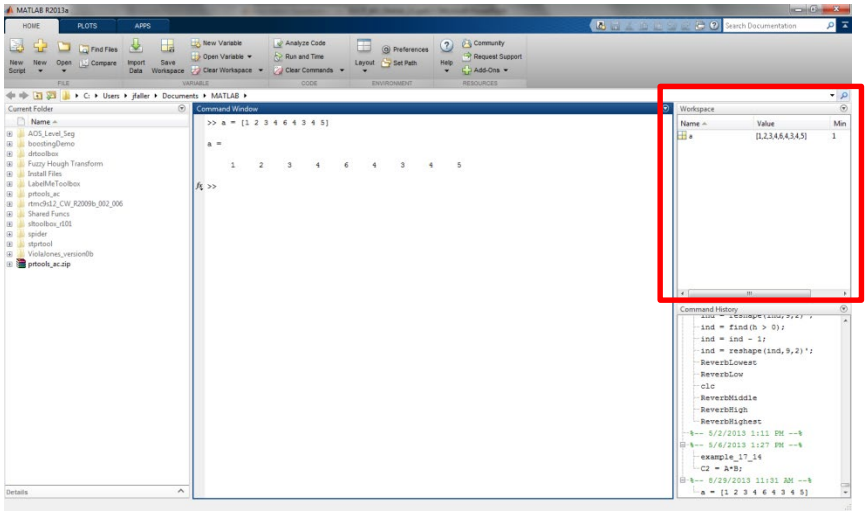


Command Window



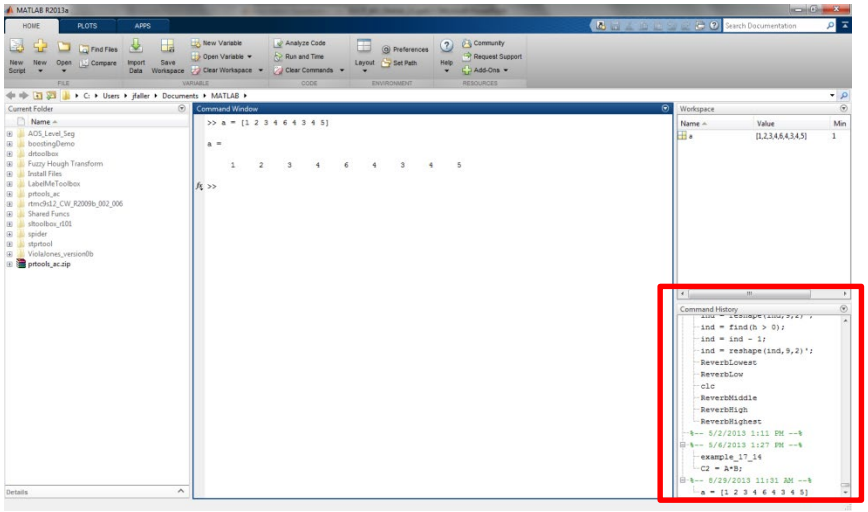
Workspace Window

- Shows the variables currently available to you



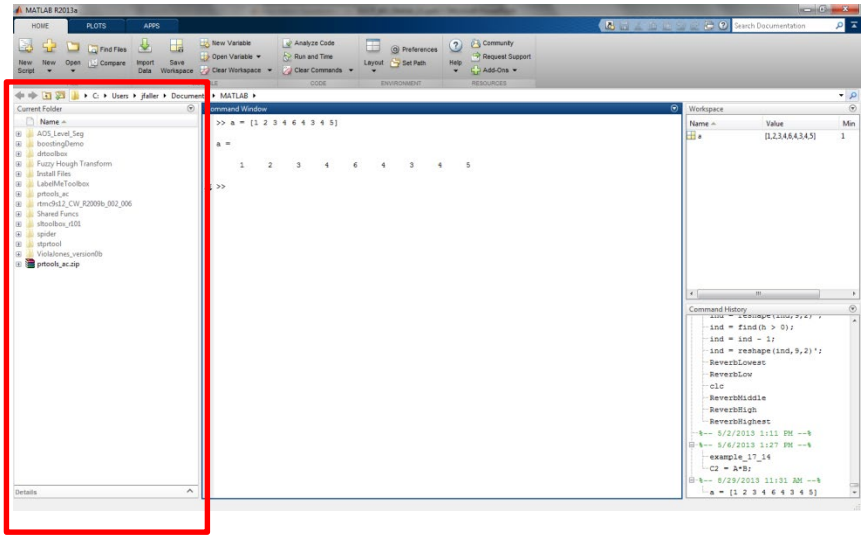
Command History Window

- Shows the commands you have entered
- Sorted by date



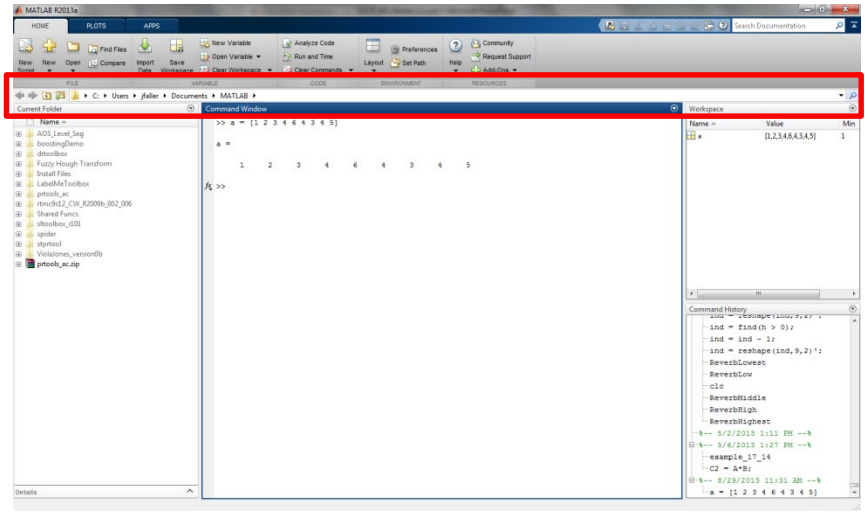
Current Folder Window

- Shows the folders for the present working directory



Present Working Directory

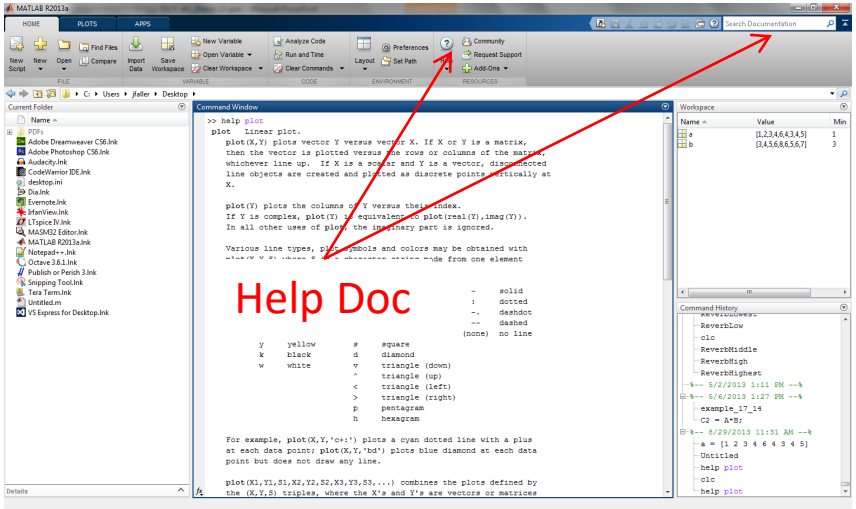
- Shows the current folder you are working in
- You can also use the command “pwd”



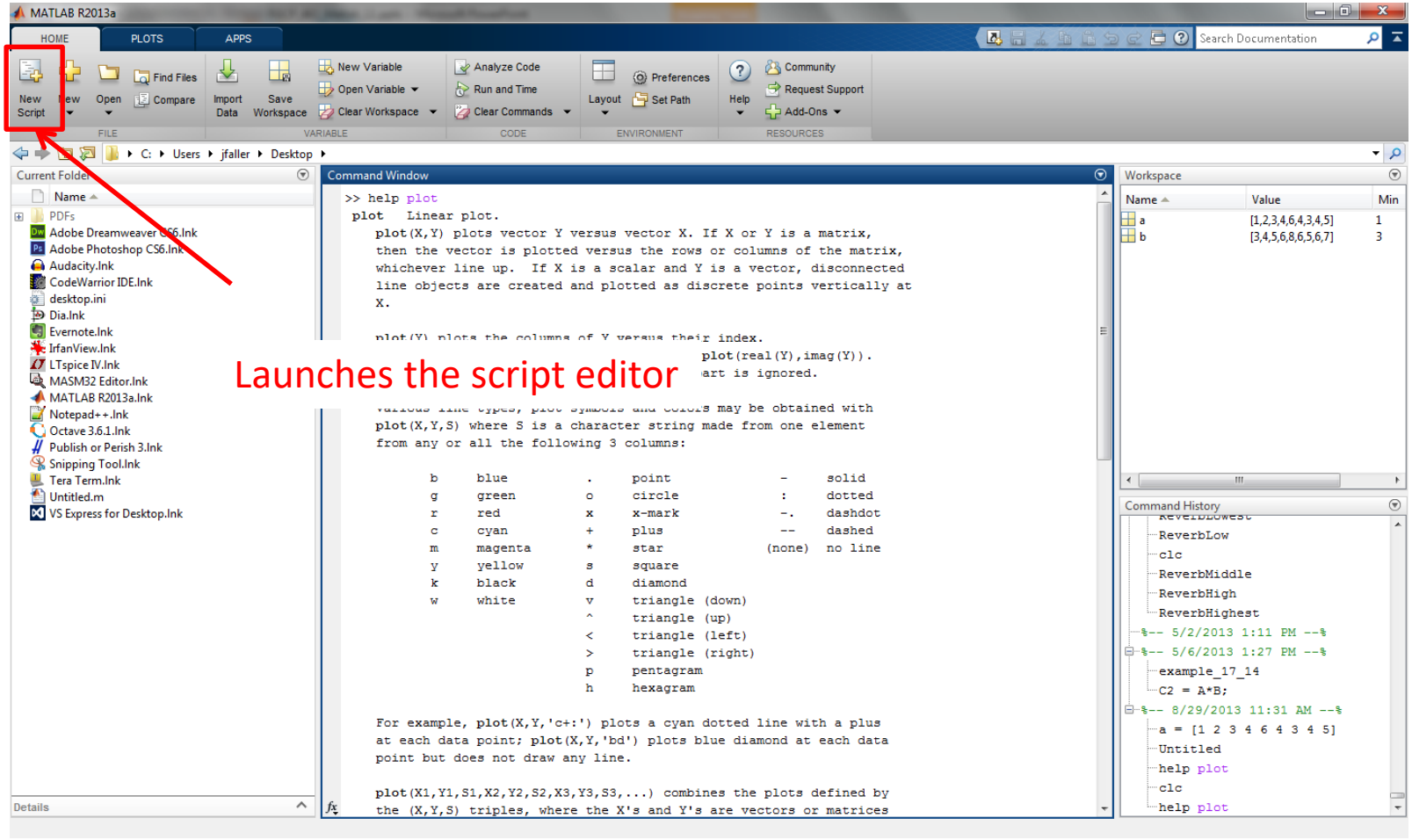
Help Docs

- Searchable help doc
- You can also use the "help" command
- Example:

```
help plot
```

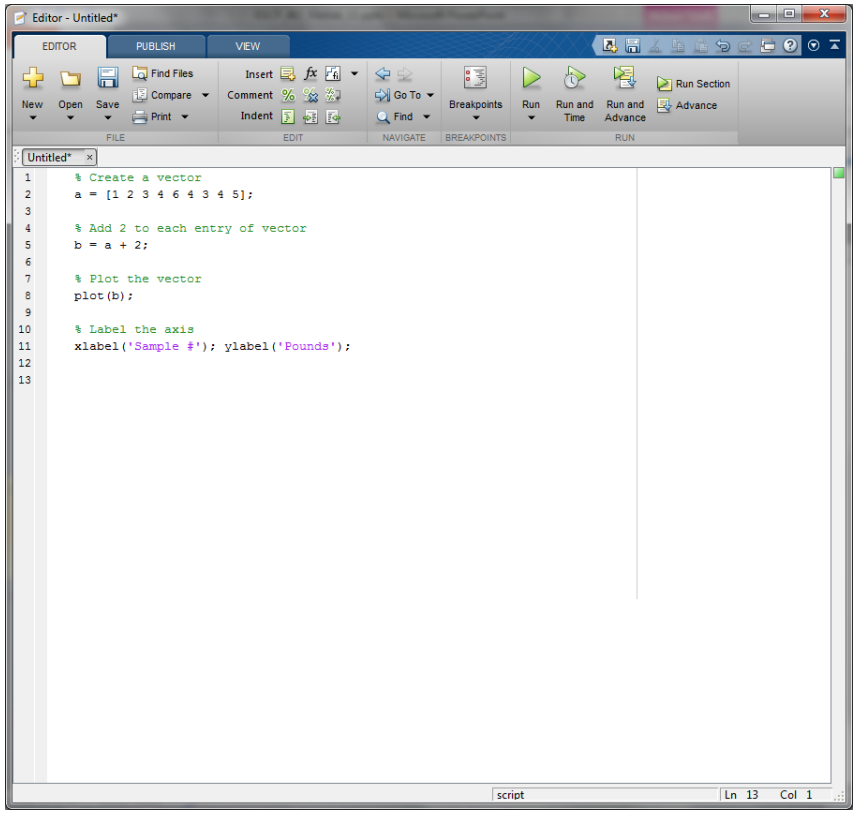


Creating Scripts

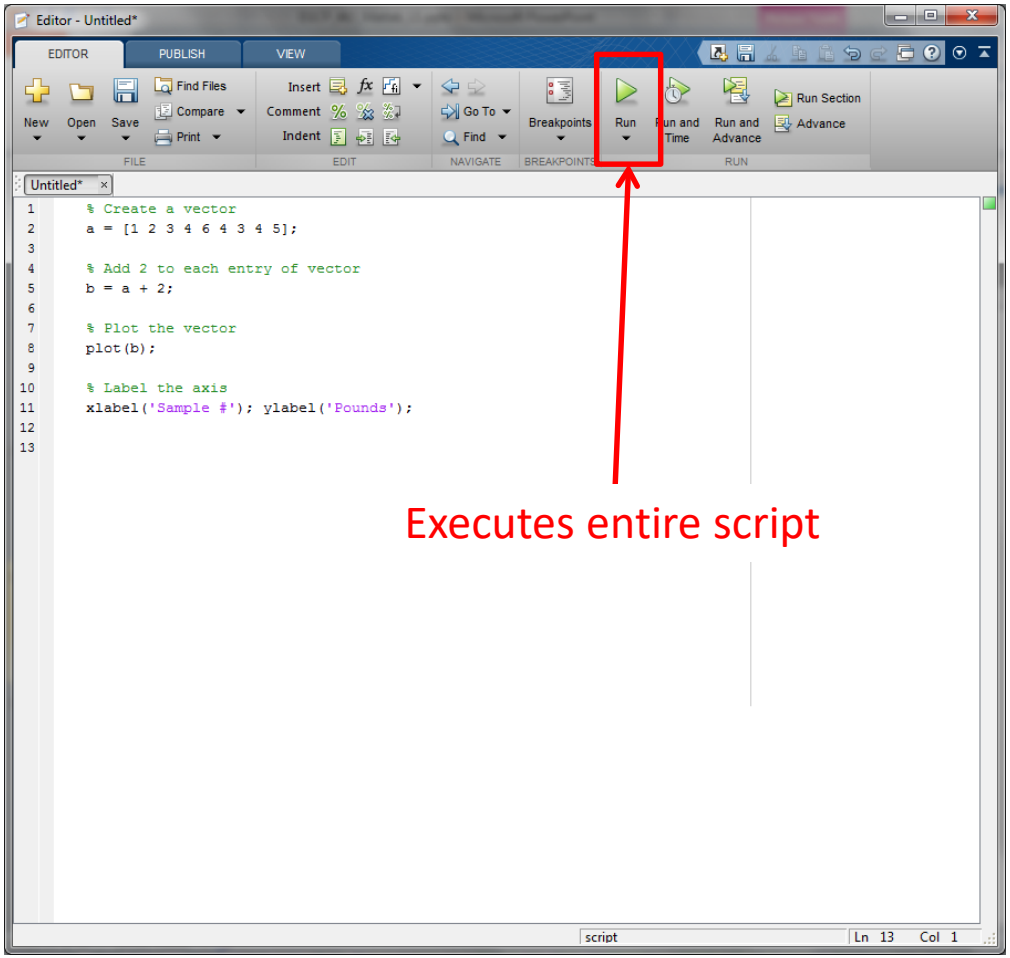


Script Editor

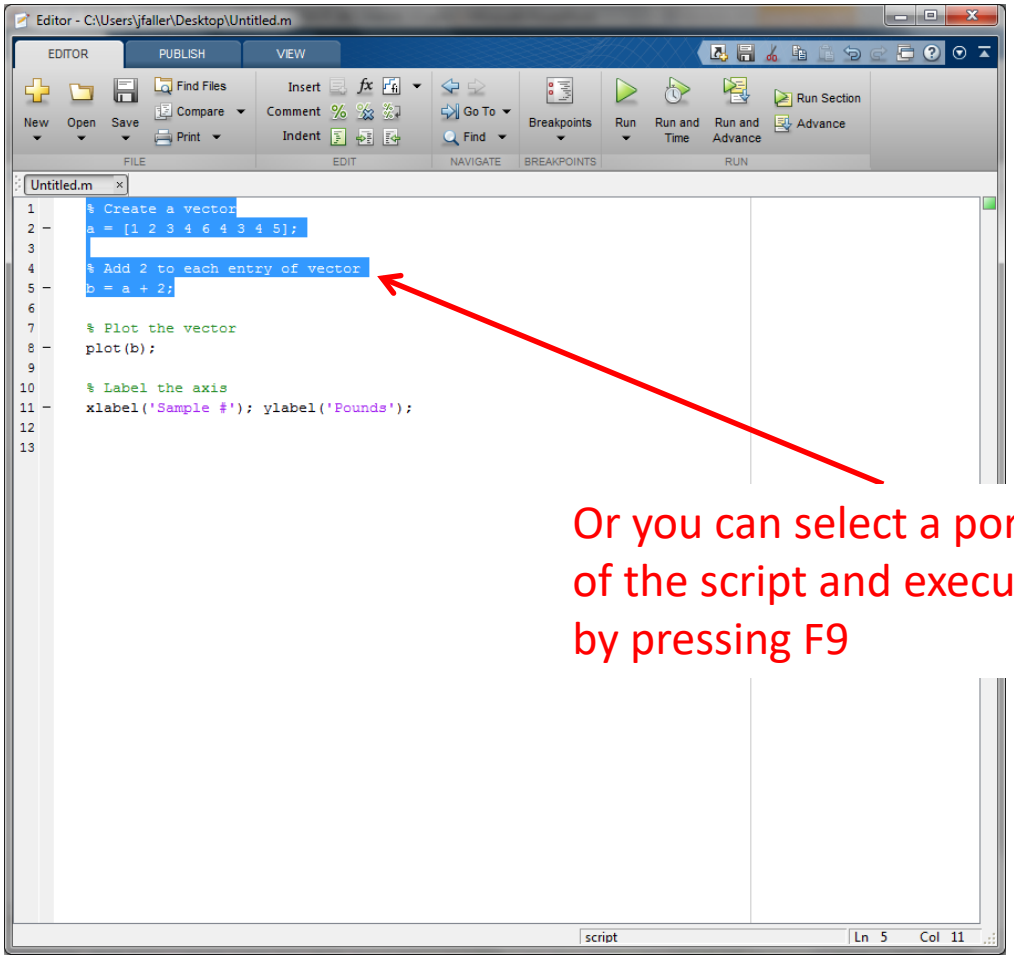
- Instead of entering in the command window directly, you can also enter commands in the script editor and save as a m-file



Script Editor



Script Editor



Or you can select a portion of the script and execute it only by pressing F9

Getting Started

```
>> a = [ 1 2; 2 1 ]

a =
     1     2
     2     1

>> a*a

ans =

     5     4
     4     5
```

- Example
  - Define a matrix “a” and computed its square
  - “a times a”
- Text in bold is what you type in the command window
- Ordinary text is what Matlab outputs

Matrices

- To enter the matrix:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

- and store it in a variable “a”, do this:

  >> **a = [1 2; 3 4];**

- To redisplay the matrix, just type its name:

  >> **a**

- Once you know how to enter and display matrices, it is easy to compute with them. First we will square the matrix “a”:

  >> **a * a**

Matrices

- Now we'll try something a little harder. First we define a matrix b:

  >> **b = [1 2; 0 1];**

- Then we compute the product ab:

  >> **a*b**

- Finally, we compute the product in the other order:

  >> **b*a**

Matrices

- Notice that the two products are different
  - Matrix multiplication is non-commmutative
- Of course, we can also add matrices:

  >> **a + b**
- Now let's store the result of this addition so that we can use it later:

  >> **s = a + b**

Matrices

- Matrices can sometimes be inverted:

  >> **inv(s)**

- To check that this is correct, we compute the product of s and its inverse:

  >> **s * inv(s)**

- The result is the unit, or identity matrix. We can also write the computation as

  >> **s/s**

- We can also write

  >> **s\s**

- which is the same as

  >> **inv(s) * s**

Matrices

- To see that these operations, left and right division, are really different, we do the following:

  >> **a/b**

  >> **a\b**

- Not all matrices can be inverted, or used as the denominator in matrix division:

  >> **c = [1 1; 1 1];**

  >> **inv(c);**

- A matrix can be inverted if and only if its determinant is nonzero:

  >> **det(a)**

  >> **det(c)**

Systems of Equations

- Now consider a linear equation

  ax + by = p

  cx + dy = q
- We can write this more compactly as

  AX = B
- where the coefficient matrix A is

  a b

  c d
- the vector of unknowns is

  x

  y
- and the vector on the right-hand side is

  p

  q
- If A is invertible, X = (1/A)B, or, using Matlab notation, X = A\B. Let's try this out by solving ax = b with a as before and b = [1; 0]. Note that b is a column vector.

  >> **b = [1; 0]**

  >> **a\b**

Loops

- Loop Example
  - We regard x as representing (for example) the population state of an island
  - The first entry (1) gives the fraction of the population in the west half of the island, the second entry (0) give the fraction in the east half
  - The state of the population T units of time later is given by the rule y = ax
  - This expresses the fact that an individual in the west half stays put with probability 0.8 and moves east with probability 0.2 (note 0.8 + 0.2 = 1), and the fact that in individual in the east stays put with probability 0.9 and moves west with probability 0.1
  - Thus, successive population states can be predicted/computed by repeated matrix multiplication

```
>> a = [ 0.8 0.1; 0.2 0.9 ]
>> x = [ 1; 0 ]
>> for i = 1:20, x = a*x, end
```

Graphing

Functions of One Variable

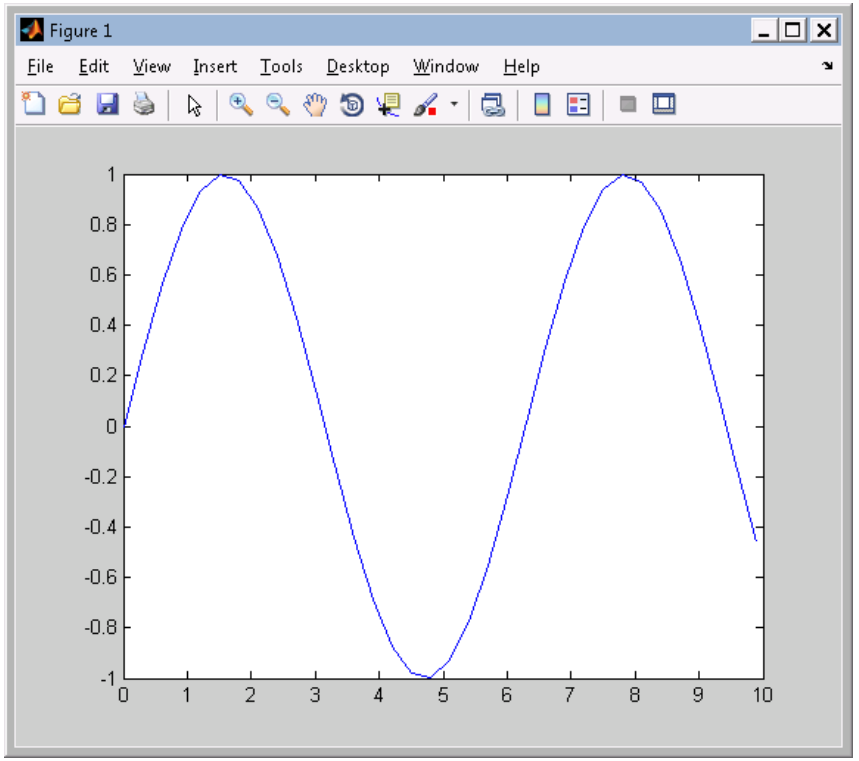
- To make a graph of y = sin(t) on the interval t = 0 to t = 10 we do the following:

  >> **t = 0:.3:10;**

  >> **y = sin(t);**

  >> **plot(t,y)**

- The command t = 0:.3:10; defines a vector with components ranging from 0 to 10 in steps of 0.3
- The y = sin(t); defines a vector whose components are sin(0), sin(0.3), sin(0.6), etc.
- Finally, plot(t,y) use the vector of t and y values to construct the graph



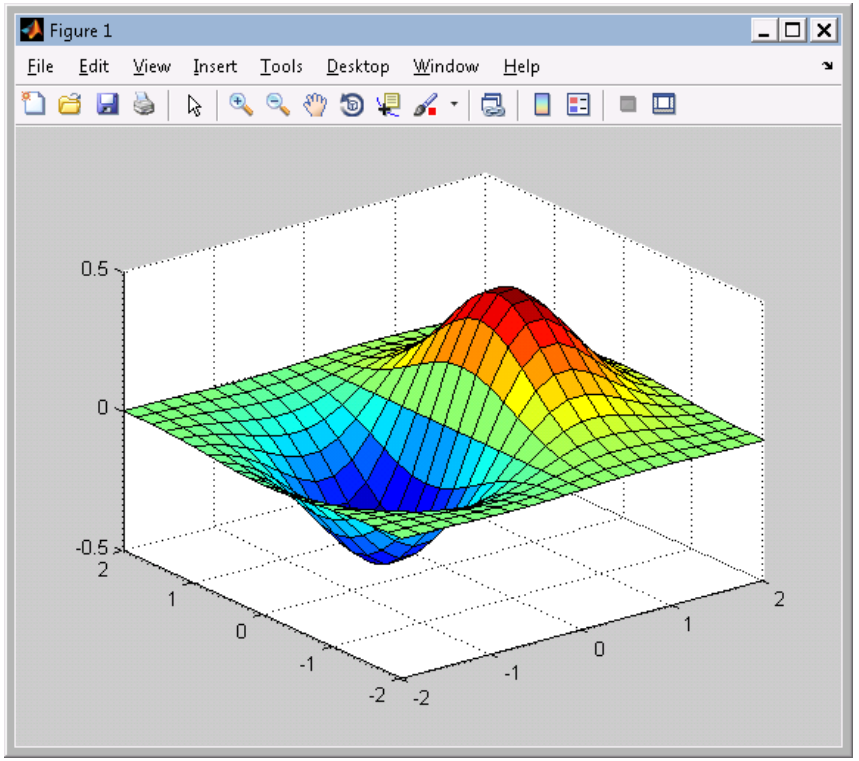
Graphing

Functions of Two Variable

- Here is how we graph the function
$$z(x, y) = xe^{(-x^2-y^2)}$$

```
>> [x,y] = meshgrid(-2:.2:2, -2:.2:2);
>> z = x .* exp(-x.^2 - y.^2);
>> surf(x,y,z)
```

- The first command creates a matrix whose entries are the points of a grid in the square -2 <= x <= 2, -2 <= y <= 2
- The small squares which make up the grid are 0.2 units wide and 0.2 unit tall
- The second command creates a matrix whose entries are the values of the function z(x,y) at the grid points
- The third command uses this information to construct the graph



Common Commands and Operators

- http://www.hkn.umn.edu/resources/files/matlab/MatlabCommands.pdf

Useful Tutorials

- Download MATLAB and do the following tutorials:
  - Basic Matric Operations
  - Getting Started with MATLAB
  - Matlab Overview Video
  - Analyzing and Visualizing Data with MATLAB
  - Programming and Developing Algorithms with MATLAB
  - Signal Related Videos