

# Math Tools: Lab I

MatLab I - variables, syntax, vectors, operations,  
basic graphics

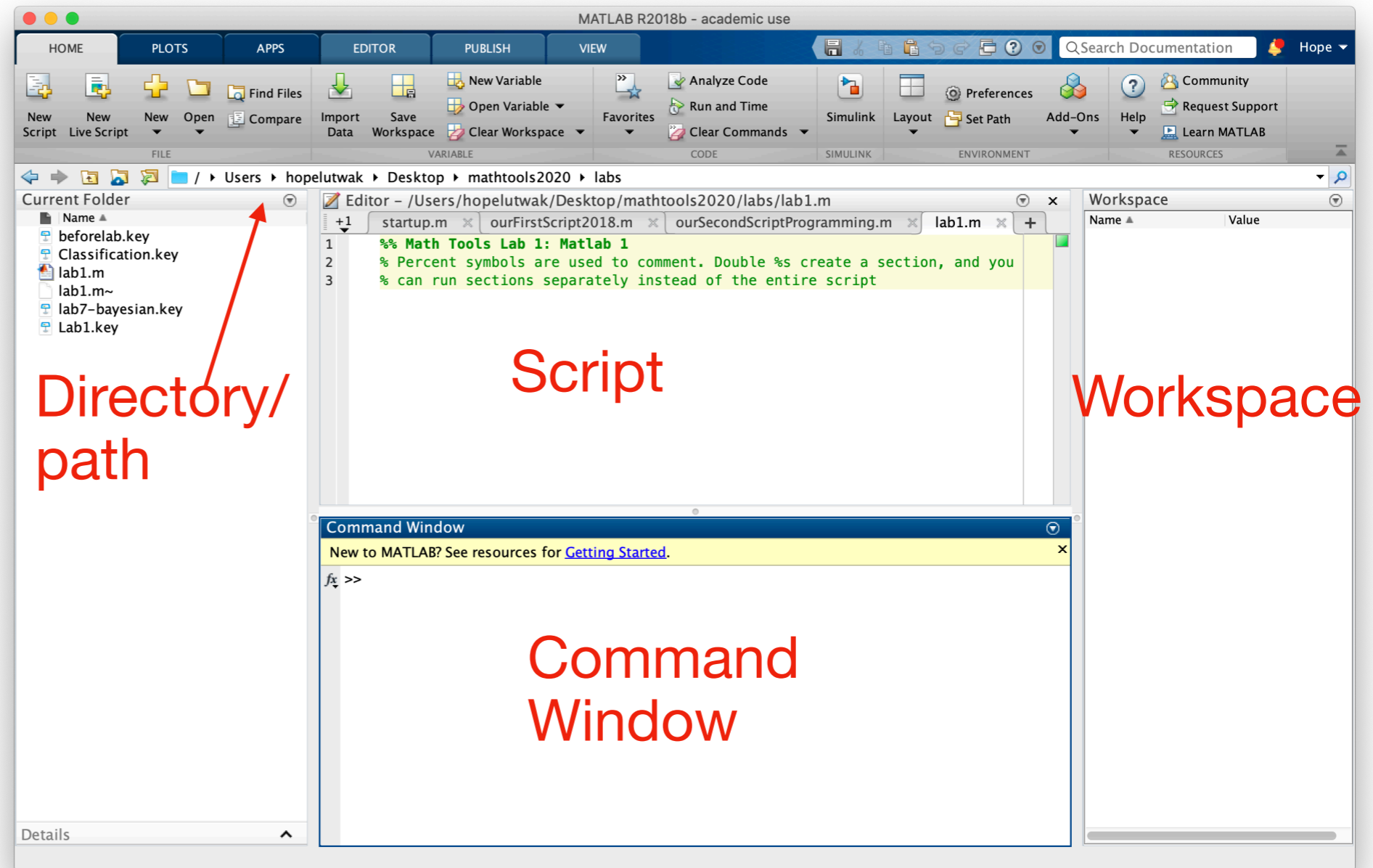
4 Sept 2020

# Matlab

- Installation

- Default display

- Download and open “lab1.m” from piazza/class website



# Basic operations

## Matlab as a calculator

- Using the command window we can use +, -, /, \* for addition, subtraction, division, multiplication
  - Exponential: ^, logarithm: log()

### Command window

```
>> 5+2^2
```

```
ans =
```

```
9
```

- **Try** it out in the command window! Be creative and have the answer be 42

# Variables

- What is a variable?
  - used to store information to be referenced and manipulated in a computer program.
    - Information can be in the form of a number, multiple numbers (vectors, matrices)
  - Provides a way of labeling data with a descriptive name, so our programs can be understood more clearly by the reader and ourselves.
- to create a new variable, enter the variable name in the command window, followed by an equal sign (=) and the value you want to assign to the variable.
  - **Try** creating a variable!

**Command window**

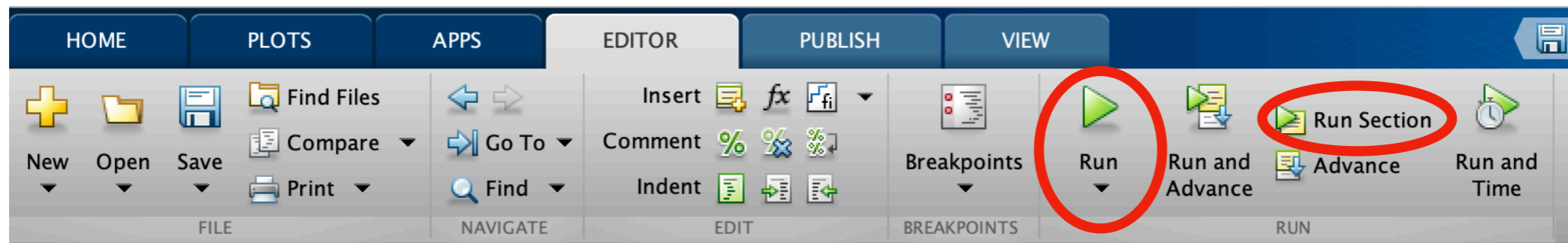
```
>> a = 5
```

a =

5
  - You can see the variable you created in the workspace
  - **Try** creating multiple variables, and using the basic operations
    - Where and how is the output saved?

# Scripts

- What is a script?
  - A script is a file that contains multiple sequential lines of MATLAB commands and function calls. You can run a script by typing its name at the command line.
  - **Try** creating multiple variables in the script, running the script by “Run” in the editor tab



- To create sections, start a line with %% followed by a space. Then you can run sections of the script without running the entire script
- keyboard shortcut for running a section: command(⌘) + enter on Mac or fn+F9 on Windows
- What do you notice in the command window?
  - Use a semicolon (;) in order to suppress the output. This will become very useful when you have lots of calculations in a script or a large variable

# Vectors and vector operations

## Command window

- Vector: an ordered list of numbers
- Column and row vector (this matters for operations!)
  - Row vector:  $v = [1 \ 2]$  (or separated by commas)
    - row vector  
`>> v = [1 2]`  
$$v = \begin{bmatrix} 1 & 2 \end{bmatrix}$$
    - column vector  
`>> v = [1; 2]`  
$$v = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$
  - Column vector:  $v = [1; 2]$
- Scalar multiplication:  $a = 2; a*v$
- Vector addition: make sure they're the same dimensions!
- **Try** creating the variables  $v = [1; 2]$  and  $w = [3; 4]$  in the script, adding them, saving the sum as  $u$ , and then running the section so that only  $u$  displays in the command window.

# Vectors and vector operations

- Retrieving an element of the vector
  - Matlab is 1-indexed so the first element is  $v(1)$ , the second is  $v(2)$
  - Last element:  $v(\text{end})$ , second to last element:  $v(\text{end}-1)$
- Unit vector: vector with length 1
  - **Try it:** What is the length of  $v = [1; 2]$ ? How can you create a unit vector in the direction of  $v$ ?
- recall: if  $v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$ , length of  $v$  is  $\sqrt{v_1^2 + v_2^2}$
- **Try it:** What is the length of  $a*v$ ?
- Inner product:  $v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}, w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$   
 $\langle v, w \rangle$  or  $v \cdot w = v_1 w_1 + v_2 w_2 = v^T w$
- **Try** with  $v = [1; 2]$ ,  $w = [3; 1]$

## Command window

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

```
v =
```

```
1  
2
```

```
>> v(1)
```

```
ans =
```

```
1
```

```
>> v(2)
```

```
ans =
```

```
2
```

# Plotting

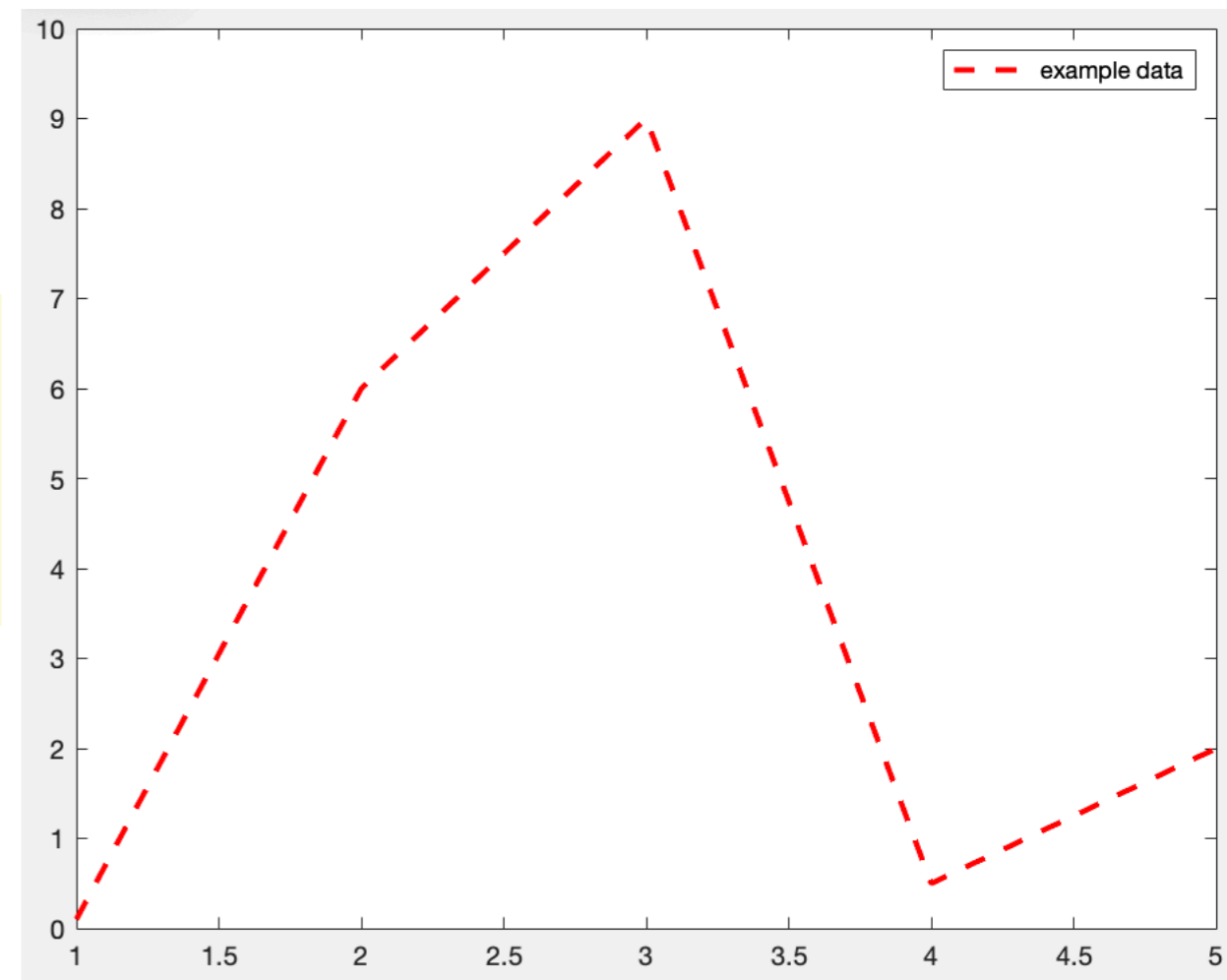
- The function `plot()` in Matlab takes in x-values, y-values, and display specifications.
  - To create a line plot, first set up the figure by typing "figure" to open a figure. On the next line we can then call the function: `plot(x-values,y-values, LineSpec)`
  - line color: preset matlab colors (red: 'r', green: 'g', blue: 'b') or 'Color' followed by the rgb values, eg red = [1 0 0]
  - line style: dashed line '--' , dotted line '.'
    - To make a dashed red line: '--r'
  - line width: `plot(x,y, 'LineWidth', 2)`
- Try visualizing the vectors  $v = [1; 2]$  in red and  $w = [3; 1]$  in blue
  - In the same figure plot:  $v+w$ , unit  $v$ , unit  $w$  (hint: use "hold on")
  - You can change the axis limits with `xlim([min, max])`, `ylim([min max])`
  - Add axis labels with `xlabel('your label')`, `ylabel('your label')`

# Plotting

- Legend: legend('first vector', 'second vector', ...etc) note: this will label data in the order you plotted them
- You can also add 'DisplayName', 'name here' within the plot function: plot(x,y, LineSpec, "DisplayName", 'name here') and then call 'legend'
- **Try** adding a legend to your plot

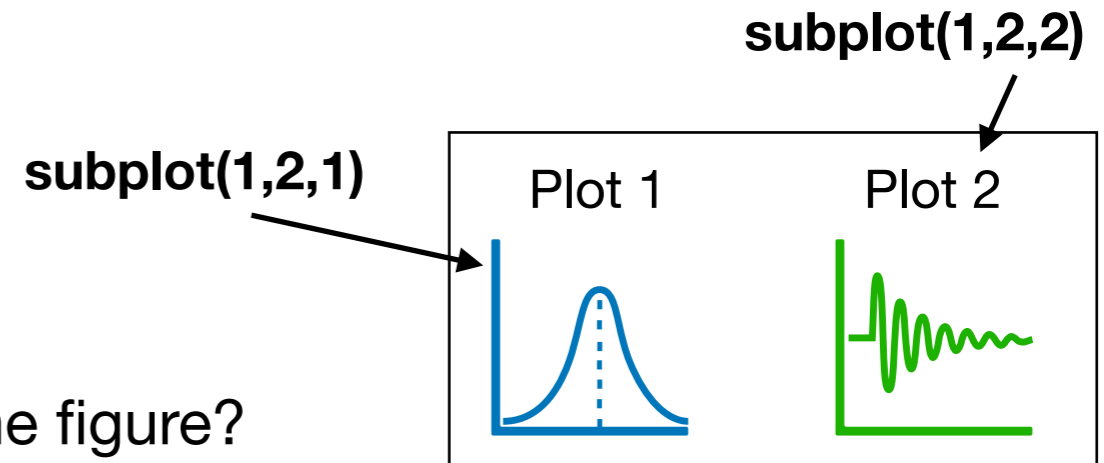
```
x = [1 2 3 4 5];  
y = [.1 6 9 .5 2];  
figure  
plot(x, y, '--r', 'LineWidth', 2, 'DisplayName', 'example data')  
xlim([1 5])  
ylim([0 10])  
legend
```

- Title: title('your title')
- **Try** adding a title

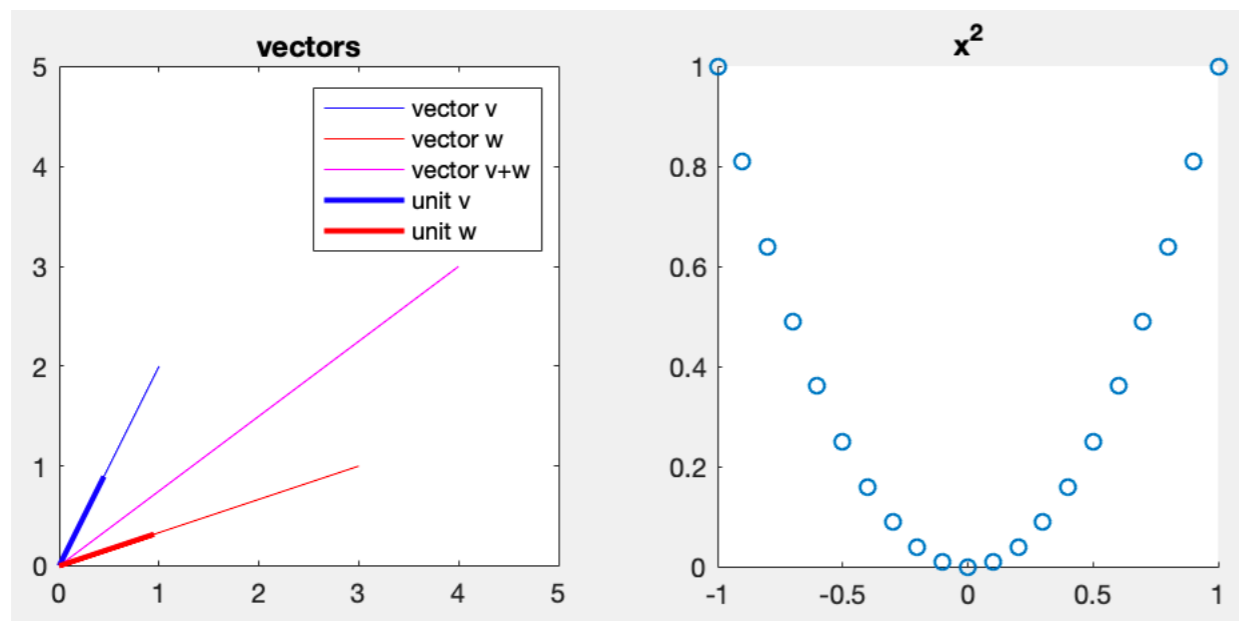


# Subplots

- Other types of plots: `bar()`, `hist()`, `scatter()`
- Subplots: what if you want multiple plots in one figure?



- call `subplot(# of rows, # columns, plot number)` then `plot(x,y)`
- **Try** plotting the vectors `v`, `w` in one subplot and a scatterplot of  $x^2$  from -1 to 1, with a step size of .1 in another subplot
  - To create a vector within a range with equal step size
    - `x = range minimum: step: range max`
    - `x = linspace(min, max, num of steps)`



Save the plot:

`saveas(fig, filename.filetype)`

# HWO

- Download from class website
  - Class website: <https://www.cns.nyu.edu/~eero/math-tools/>
- Submission instructions also on the class website