A Brief

Introduction to Script Files

- Up to this point, we've typed all our commands into MATLAB's Command Window and executed these commands by pressing the Enter Key.

- However, as you might begin to notice, using the command window to execute a series of sequential commands that are related to each other (a program), can be frustrating, difficult or even impossible.

- I like to think about the Command Window like a piece of scratch paper: it's great for working with simple ideas or recalling key features from projects I worked on previously. However, some features of the command Window make it a less-than ideal space to work on large projects.

☐ The command window is an inconvenient space to execute a series of related commands because:

A. Commands executed in the command window cannot be saved* to be executed again.

Note:

☐ While it is true that MATLAB tracks and records a log of the most recent 25,000 commands we enter in the command window (known as the command History), we have little control over this log (other than to search through our History)

☐ Sometimes we will want to save an ordered list of commands that we plan to run over and over again with the added benefit of being able to edit, organize, and save such commands. This is particularly true for sequential commands where output from one command is used as input for commands that follow.

B. The Command Window is not interactive. Every time we press enter, only the most recent command (the command most recently typed in the command prompt) will be executed. All commands executed prior to this point remain unchanged.*

* Note:

- Again, the line-by-line execution feature of the Command Window makes it hard to execute sequential commands. Imagine we make a mistake in our first command and then write 10 subsequent sequential commands that use our inaccurate data before we realize the error. If we go back to correct our initial error, we will need to re enter all subsequent commands in the correct order to get back on track. This can be quite frustrating and inefficient.

- A different (and often better) method to execute commands in MATLAB is to create a script file.

- A script file is a file in which we can type a list of commands (a program), save this list to a folder, and run (execute) the file at any time we'd like.

- When we execute a script file, MATLAB executes each command contained in that file, line-by-line, in the order that the commands are listed.

- Using this tool, we can correct, change, or edit any part of our script file, save our work and run the file again at any time.

☐ A useful analogy comes to mind:

- Let's use our experience writing essays in English class to get insight into the benefits and drawbacks of the Command Window versus script files.

- In this class, you will be asked to analyze the "world" by writing MATLAB code. In English class, your asked to analyze the "world" using written prose (essays and writing assignments).

- Think of the Command Window as writing ideas out with paper and pencil. This is a super effective mechanism to capture ideas "on the go", to brain storm, and to further develop your thoughts before refining them. However, paper and pencil is not super effective for the editing and refinement process, nor for sharing your work.

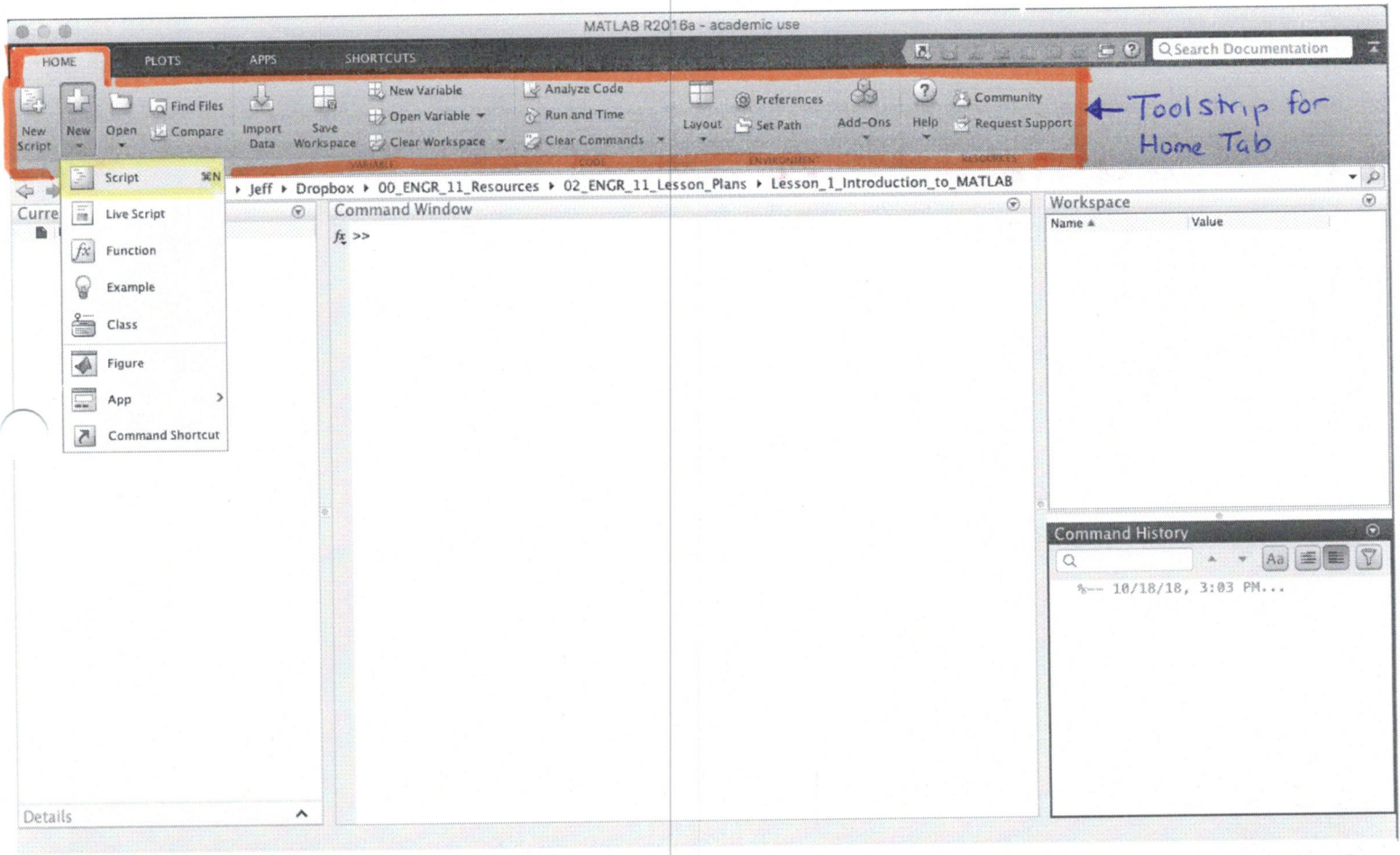
- Think of Script files as a Microsoft Word document. When preparing to write an essay, we don't always want to use MS word (since sometimes ideas come to us when we are away from our computer). But, if we are ready to compose, revise, edit and refine our ideas so that we can return to our work over a long period of time, MS word (script files) is a much more effective tool than paper and pencil.

In this brief intro to script files, we only cover ~~the~~ bare minimum required to create and run simple programs. We ~~will~~ do so to enable students to use script files to solve problems in lessons 1 -

However, we will defer a number of important features of script files for later. In particular, we will get a much deeper look at script files in a ~~Lesson~~ later lesson

Create a Script File

- To create a script file, toggle to the HOME tab of the MATLAB Toolstrip*. Then, click on the New (Create New Document) menu. Now highlight and click the Script option in this drop down menu.



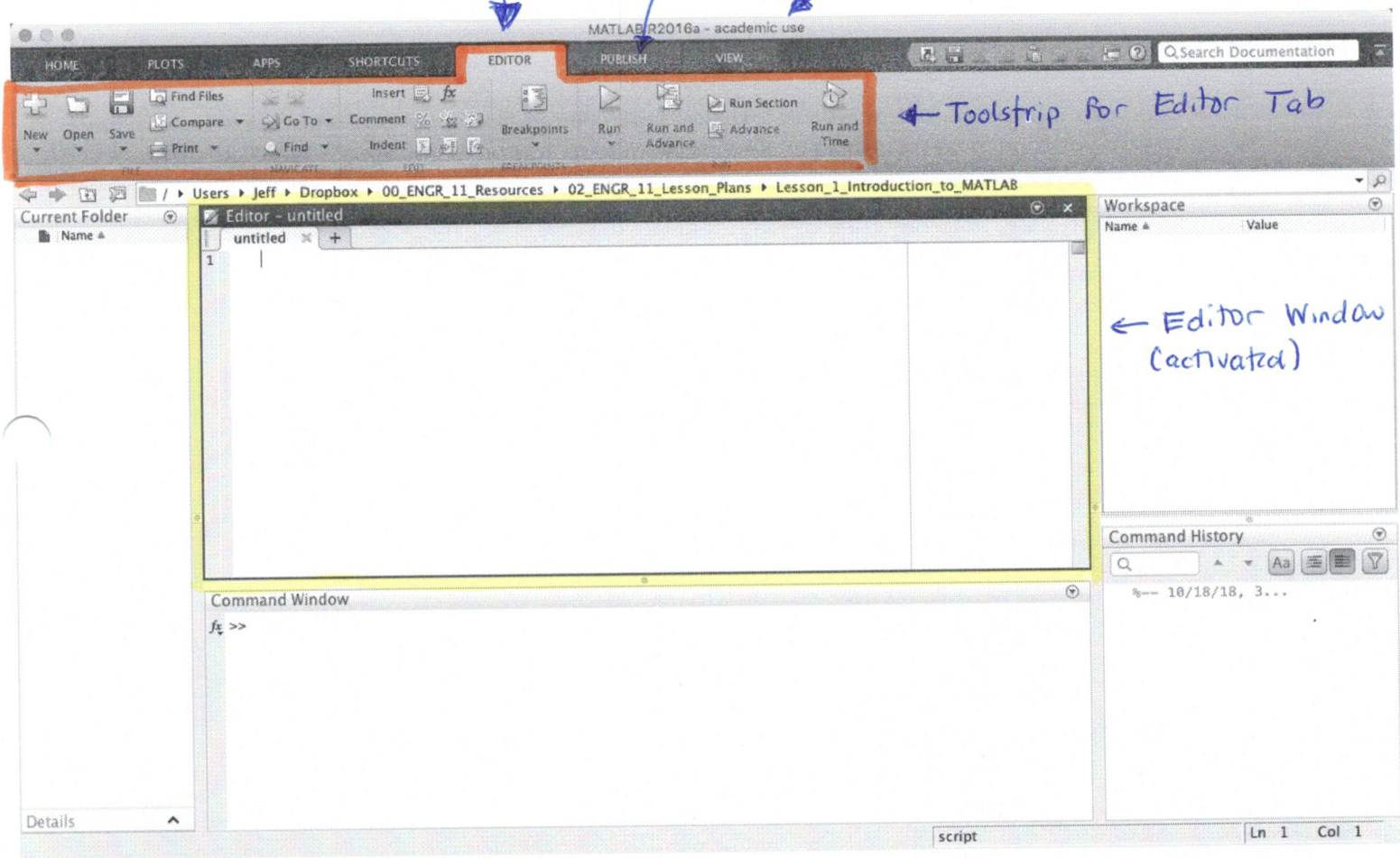
- Alternatively, you can also use the keyboard shortcut shown on screen. For example, on my Mac, I can create a script using Command + N.

* For more about the toolstrip, see

☐ Notice that a few interesting changes occured:

A. Our (unsaved) script file was created and opened in an Editor Window (which is docked in the MATLAB Desktop in the figure below)

B. Three new tabs appeared at the top of our Desktop: the Editor tab, the Publish tab, and the View tab



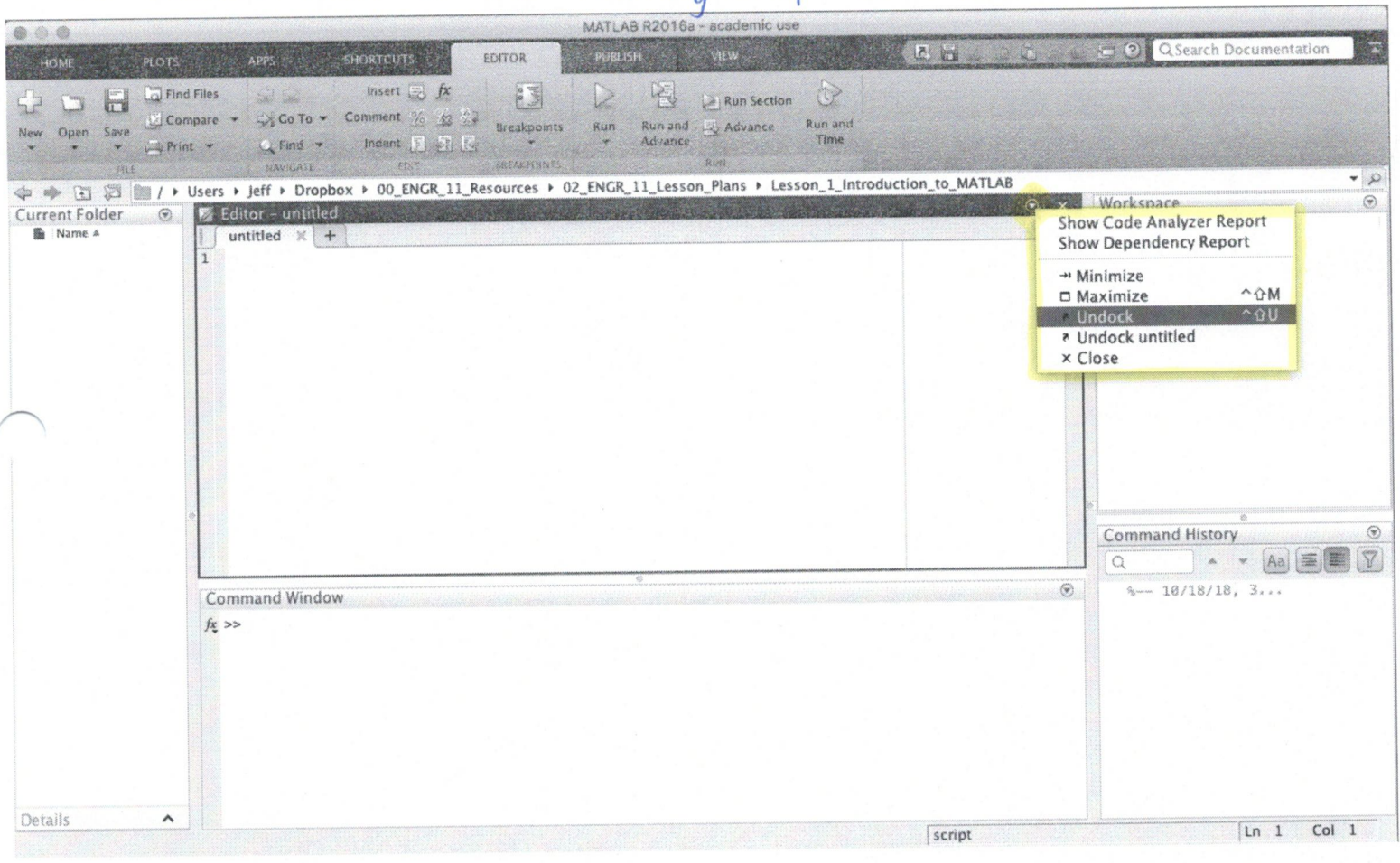
C. The Editor tab was automatically activated and we get access to the toolstrip for this Editor tab

D. The Editor Window was automatically activated.

Note: The major idea behind these Toolstrips in MATLAB is to give us access to clickable icons that provide the most prominent features we might need to use ...

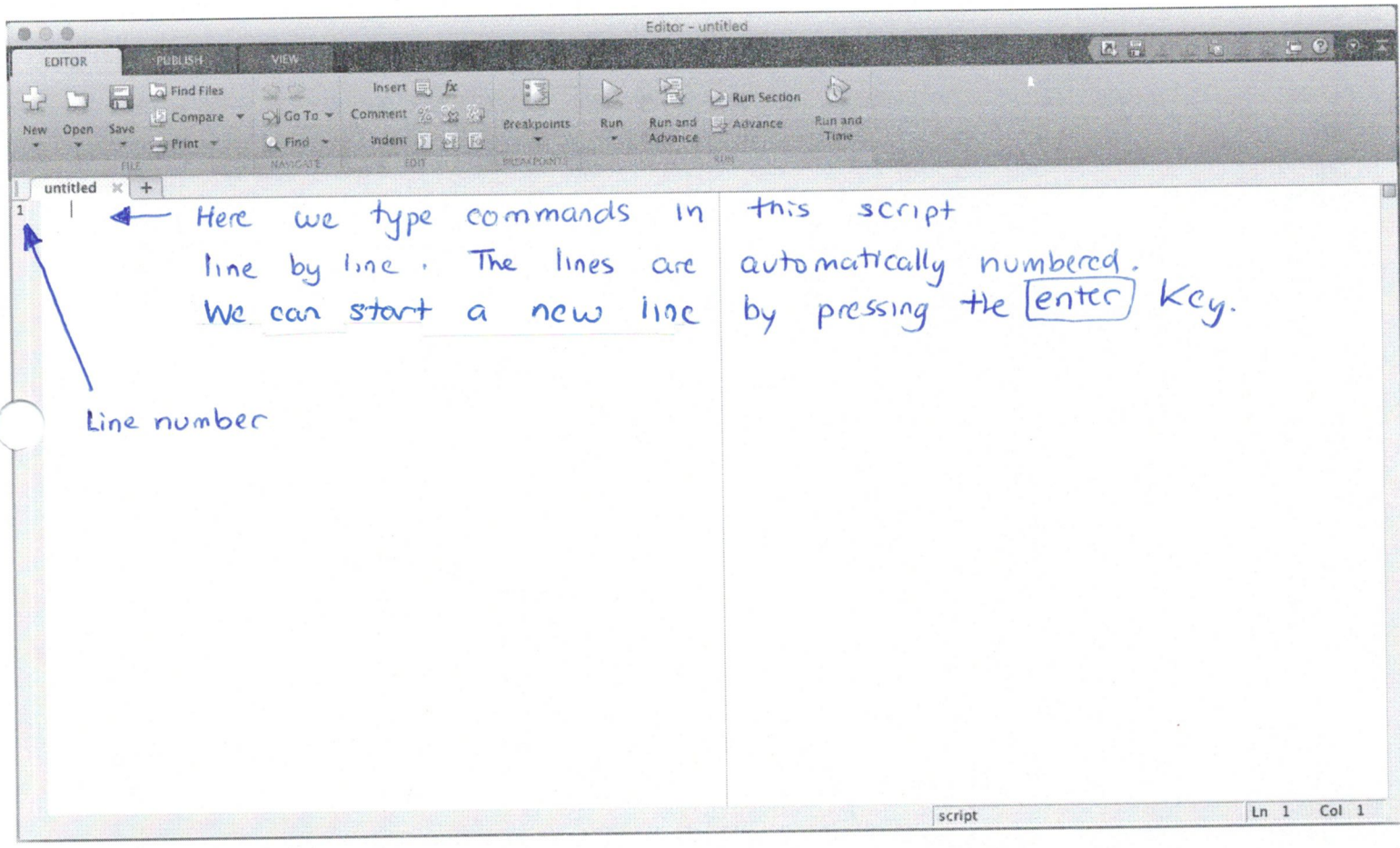
- Because Script files will often include a long list of commands, I like to work with my Editor window undocked. This gives me more screen space to view my script file(s).

- To undock the Editor Window, single click the little upside-down-triangle-in-a-circle button on top right of the editor window, and click on the undock item in the resulting drop-down menu.



- We can also use keyboard short cuts to accomplish docking and undocking:
  - CTRL-SHIFT-U (^⇧U) undocks an activated window
  - CTRL-SHIFT-D (^⇧D) docks an activated wind

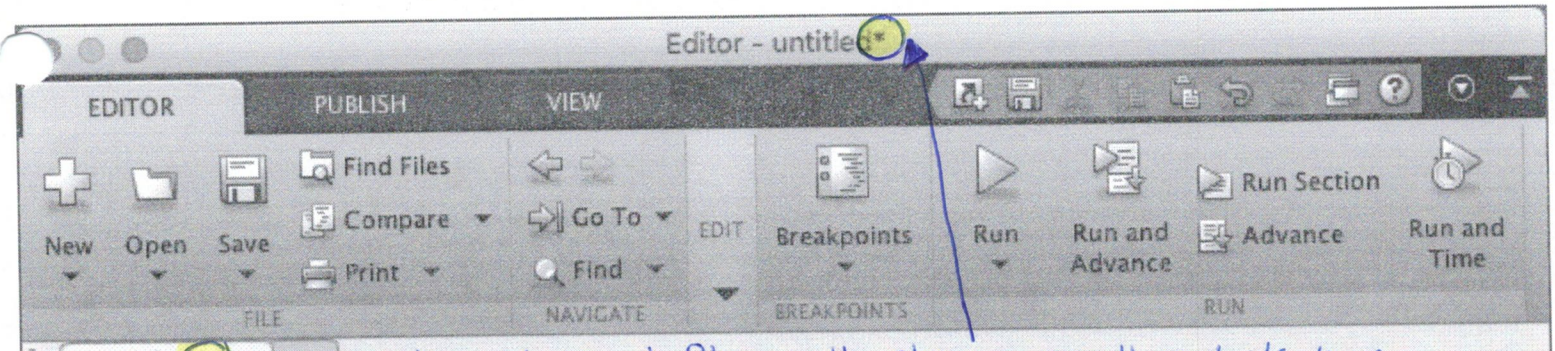
- Once we've undocked our Editor Window, notice that we now have a seperate window that contains just our (unsaved) script file. Moreover, the new (contextualized) tabs that popped up when we openned our script file (the Editor, Publish, and View tabs) came along with us to this new Editor Window.



- Once we've opened the Editor Window, we can type commands in our script file line by line.

- MATLAB Automatically numbers a new line everytime we press the Enter key.

☐ Let's write an Example script designed to solve general quadratic equations. Such a script is seen below



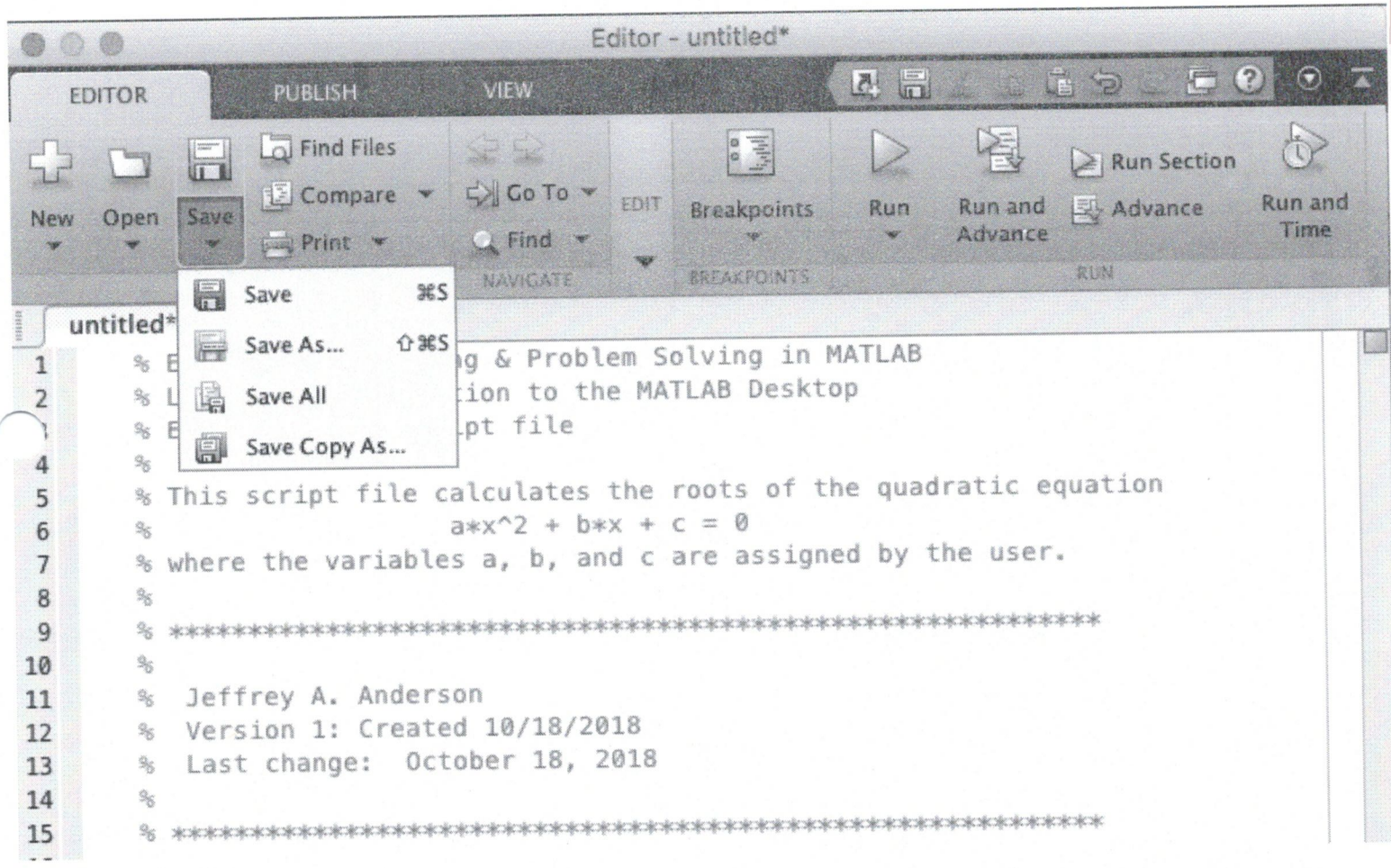
untitled* — Unsaved script files will show a small astrik here

```
1   % ENGR 11: Programming & Problem Solving in MATLAB
2   % Lesson 1: Introduction to the MATLAB Desktop
3   % Example 1 of a script file
4   %
5   % This script file calculates the roots of the quadratic equation
6   %                    a*x^2 + b*x + c = 0
7   % where the variables a, b, and c are assigned by the user.
8   %
9   % ************************************************************
10  %
11  %  Jeffrey A. Anderson
12  %  Version 1: Created 10/18/2018
13  %  Last change:  October 18, 2018
14  %
15  % ************************************************************
16
17  %Define quadratic coefficients
18  a =  2;     %Define degree-two coefficient
19  b = -5;     %Initialize degree-one coefficent
20  c = -3;     %Initialize degree-zero coefficient
21
22  %Use coefficents a, b, c to calculate the discriminant
23  dis = (b^2 - 4*a*c);
24
25  % The work below is based on the quadratic formula
26
27  % Left-hand root
28  % Calculate x-coordinate of x-intercept on left-hand side of vertex
29
30  x1 = (-b - sqrt(dis))/(2*a);
31
32  % Right-hand root
33  % Calculate x-coordinate of x-intercept on right-hand side of vertex
34  x2 = (-b + sqrt(dis))/(2*a);
```

script | Ln 19 Col 35

☐ Before a script file can be executed, we must save the file. We can tell this file is not saved based on the small astriks * highlighted in yellow above

☐ To save our .m script file, we single click on the Save icon in the Editor toolstrip and select "Save As..." item in the drop down menu (we can also use the keyboard shortcut CTRL-SHIFT-S (^ ⇧ S)).



☐ When we save our work, MATLAB will automatically add the extension .m to the file name.

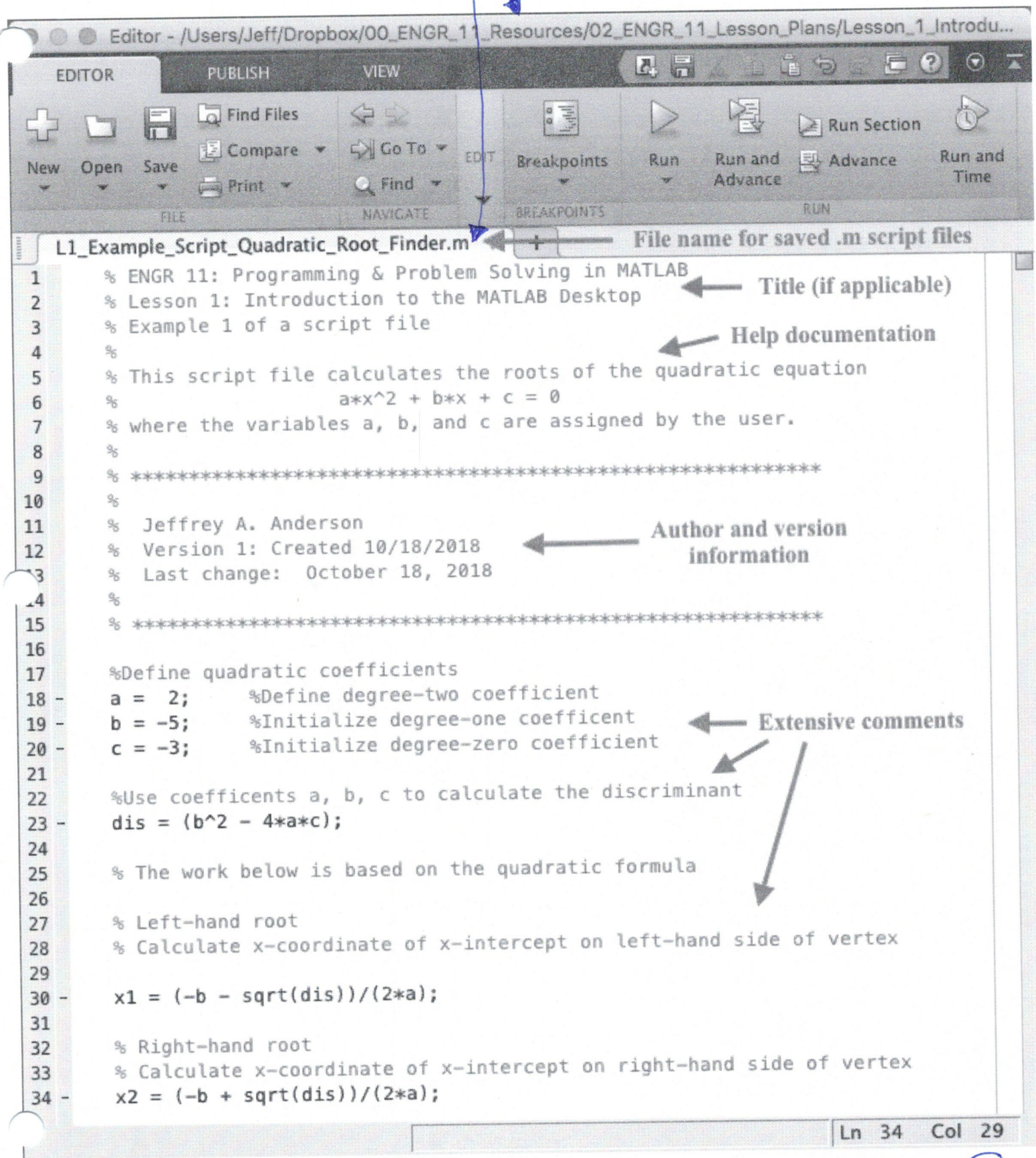
Rules for naming script files

Just as we had a list of rules for assigning variable names, so too we have a list of rules for the name of saved script files:

1. must begin with a letter
2. can be up to 63 characters long
3. Can contain upper case and lower case letters, decimal digits (0, 1, 2, 3, ... 9), and the underscore character
4. Cannot contain punctuation characters
5. Cannot contain spaces between characters

Also, the names of user-defined variables, predefined variables, and MATLAB commands or functions should not be used as names of script files.

☐ Once we've saved our file, we see the updates (and no astrik) in the appropriate places.




```
1     % ENGR 11: Programming & Problem Solving in MATLAB
2     % Lesson 1: Introduction to the MATLAB Desktop
3     % Example 1 of a script file
4     %
5     % This script file calculates the roots of the quadratic equation
6     %                    a*x^2 + b*x + c = 0
7     % where the variables a, b, and c are assigned by the user.
8     %
9     % ************************************************************
10    %
11    %  Jeffrey A. Anderson
12    %  Version 1: Created 10/18/2018
13    %  Last change:  October 18, 2018
14    %
15    % ************************************************************
16
17    %Define quadratic coefficients
18 -  a =  2;      %Define degree-two coefficient
19 -  b = -5;      %Initialize degree-one coefficent
20 -  c = -3;      %Initialize degree-zero coefficient
21
22    %Use coefficents a, b, c to calculate the discriminant
23 -  dis = (b^2 - 4*a*c);
24
25    % The work below is based on the quadratic formula
26
27    % Left-hand root
28    % Calculate x-coordinate of x-intercept on left-hand side of vertex
29
30 -  x1 = (-b - sqrt(dis))/(2*a);
31
32    % Right-hand root
33    % Calculate x-coordinate of x-intercept on right-hand side of vertex
34 -  x2 = (-b + sqrt(dis))/(2*a);
```

☐ Throughout this course, we will cultivate some useful habits when writing code, some of which are labeled in the file above.

Useful Habits for Script files

1. Choose a descriptive (and relevant) file name

2A. When appropriate, provide title information to provide context to your file

2B. Write detail help documentation that explains the background and purpose of the script file and gives you insight into the general utility of the file

2C. Provide version information w/ author name & date(s)

3. Build in error checking routines with fully descriptive error messages (not seen here)

4A. Write "good" code

4B. Avoid writing "bad" or lazy code

5. Provide extensive comments

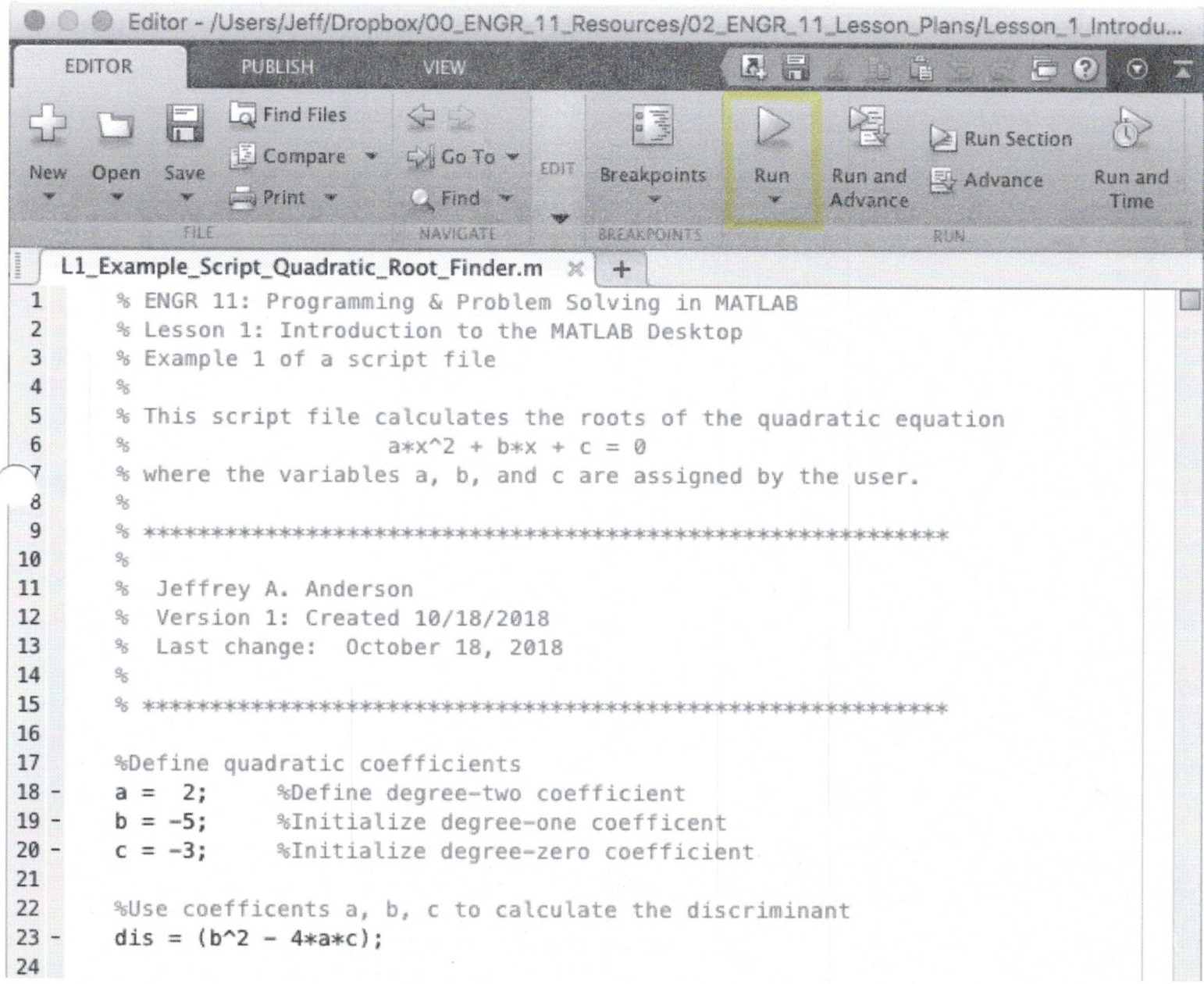
Talk about idea of writing code that outlives your tenure...

- (only cry once)...
- Don't make next person resolve your work
- kill our ego!

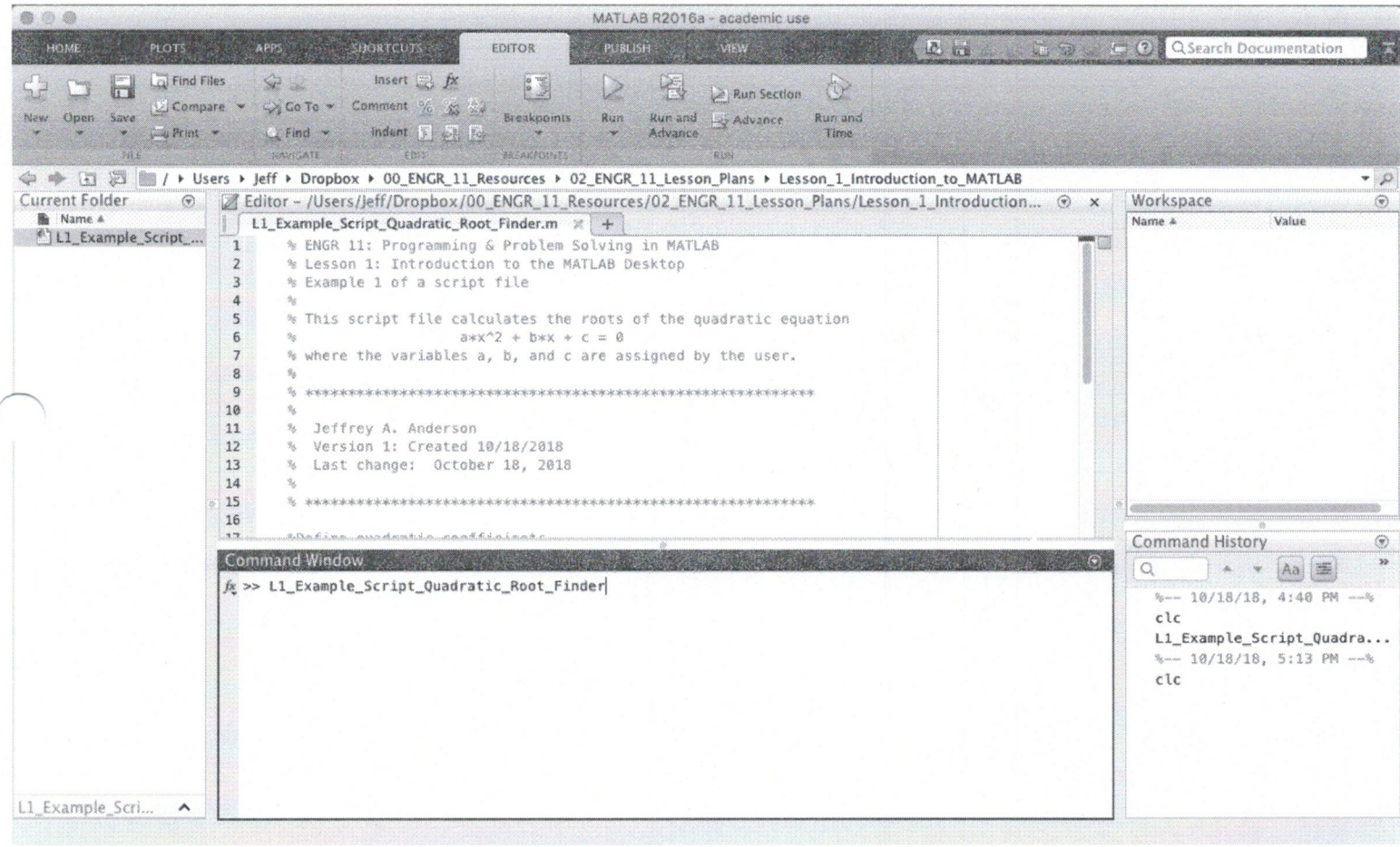
Our matras as Programmers:

- "an ounce of prevention is worth a pound of cure"
- "Bad habits are easy to form but hard to live with. Good habits are hard to form and easy to live with."

☐ We can run (execute) our saved script file directly from the editor window by clicking the Run icon as seen below



- We also may* be able to run our script by typing the file name in the command window (use autocomplete feature via [tab] key) and then pressing [enter]



Note:

- In order to execute our file using the command window, MATLAB needs to "know" where the file is saved.
- There are two ways to "teach" MATLAB how to find our script file:
  1. Set MATLAB's current folder pointer to be the folder where the file is saved
  2. Make sure the file is saved in a folder in MATLAB's search path

Overview of Script Files

- A script file is a sequence of MATLAB commands, also called a program, that can be saved and edited.

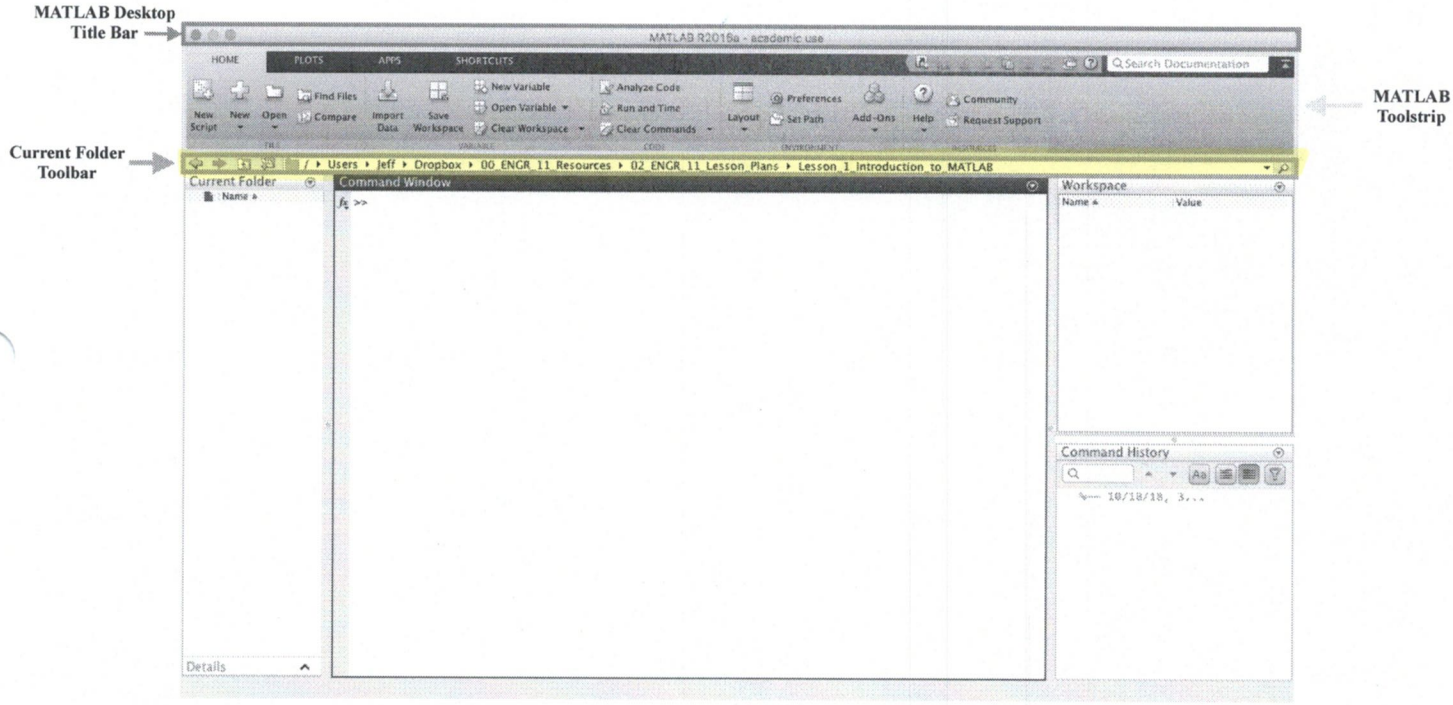
- Script files are also known as M-files because MATLAB uses the extension .m to save these files in our chosen directory.

- When we run (execute) a script file, MATLAB executes all the commands contained in this file in the order the commands are written. (just as if we typed these commands into the Command Window in that order).

- If we type a command into a script file that generates output (like assigning a variable without using a semicolon), the output is displayed in the Command Window.

Understand MATLAB's Current Folder (aka Working Directory)

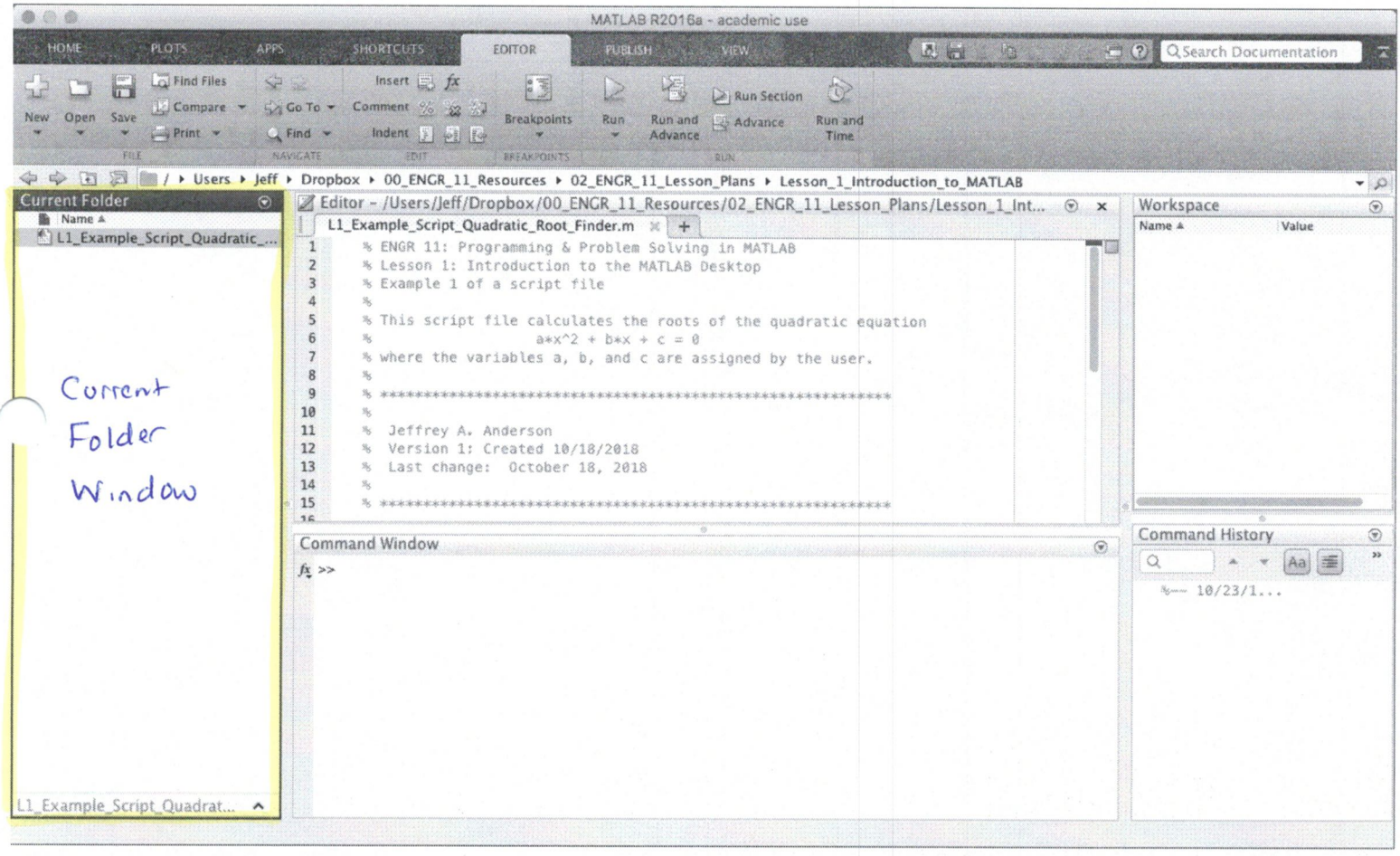
- The "current folder" is displayed in the Current Folder toolbar section of MATLAB's Desktop Window, as seen in the figure below.



- The "current folder" is a reference location that MATLAB uses to find files. This folder goes by many names including:

1. Current Folder
2. Current Directory
3. Current Working Folder
4. Present Working Directory

- MATLAB uses this Current Folder reference location to locate files that we want to use while programming (like script files, function files, data files, etc.).



Current Folder Window

- As long as we've saved our files in the same location as specified in MATLAB's Current Folder field, we can easily work with any file within this folder.
- To see a complete list of all files, we use the Current Folder window.

☐ The Working Directory Window (aka the Current Folder Window)

The Working Directory window, also known as the Current Folder window, displays a list of all files in our "current working directory" that we might use to execute code in MATLAB.

Some notes about the Working Directory:

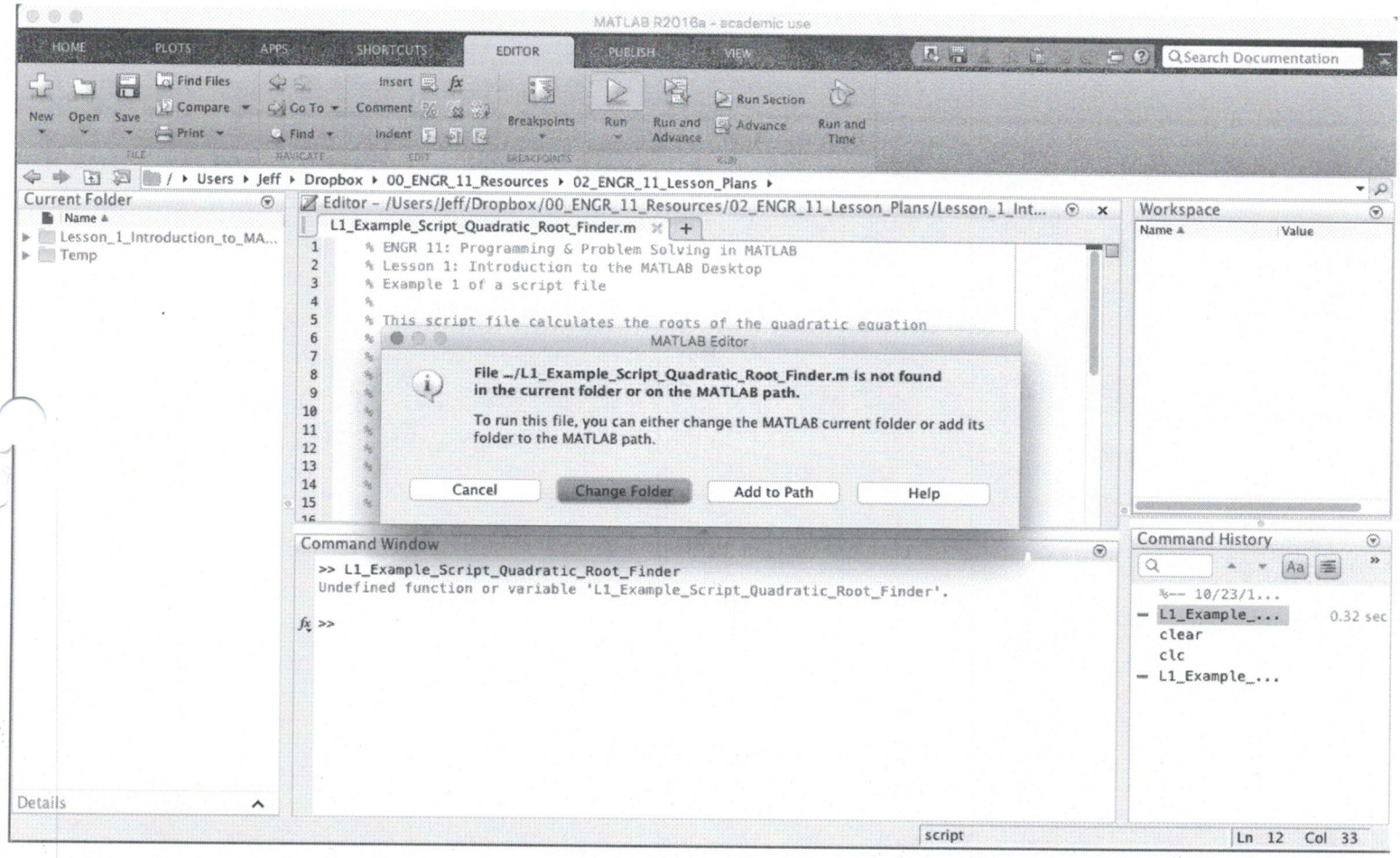
- ☐ A "directory" is a synonym for the word "folder". Because most contemperary operating systems use a hierarchical file system to organize files, we think of a directory as a folder in this organizational system.

- ☐ One of the features of the MATLAB environment is that we can modularize any code we write and thus break large programming tasks into a collection of smaller programming projects. To do so, we often save our work in multiple, different files.

- Notice in our Figure on the last page, our script file that we wrote to find the roots of a quadratic equation is saved in the Current Folder.

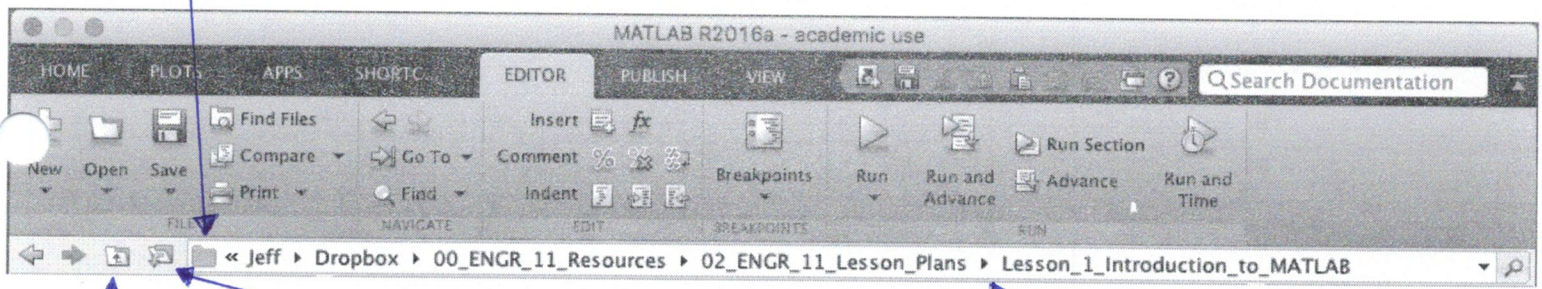
- Thus, when we run (execute) this file by clicking the Run icon in the Editor Toolstrip or by typing the name of the script file in the Command Window and then pressing Enter, MATLAB executes the file since the location of this script file is immediately on call.

☐ However, if we attempt to run a script file whose location is unknown to MATLAB (i.e. is not saved in the Current Folder then MATLAB will return an error prompt, as seen below.



☐ Of course, we as users can change our current folder. In fact, MATLAB provides many tools for navigating through our computer's filing system to eventually set our current folder to our desired location.

☐ One easy way to change the location of our Current Folder is to use the buttons on the Current Folder toolbar as seen below.



Path Name Button

Click on the miniature file-shaped button to produce a string of the path name for the current folder

Browse for folder Button

Click this button to browse your computer's file system and choose a location for the Current Folder

Up One Level Button

Click this button to set your current folder to be "one level up" in the computer's file system

Click on any of these small rightward pointing triangles to get a list of all folders that sit in the same location

- Another handy feature of MATLAB is that we can also type commands in the command window to change the current Folder and to gather more information about files and folders in our computer's file system

- Below is a list of useful commands to learn more about or change the location of the current folder

Command	Description
`pwd` *	displays the MATLAB current folder
`ls`	provides a brief list of the contents of the current folder
`dir`	provides a more comprehensive list of the contents of current folder
`what`	provides the path for the current folder and all the MATLAB files and folders found in the current folder
`cd`	change the current folder (using command line)

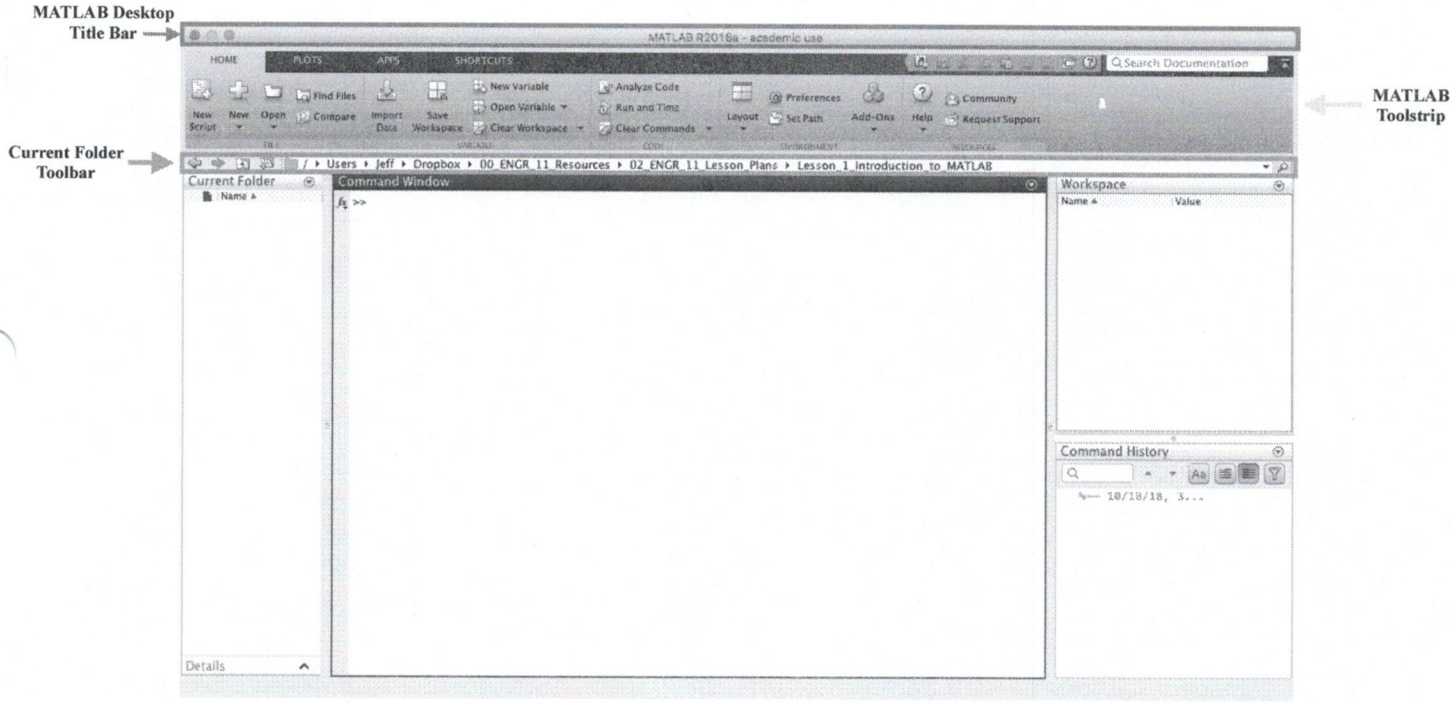
Note:

- One way to think about the current folder is that it is the exact folder on our computer that is retrieved when we call the `pwd` function in the command window.

Understand the MATLAB Toolstrip

- We end this lesson with a discussion of MATLAB's Toolstrip. The Toolstrip organizes many of MATLAB's most frequently used features into a series of context specific tabs.

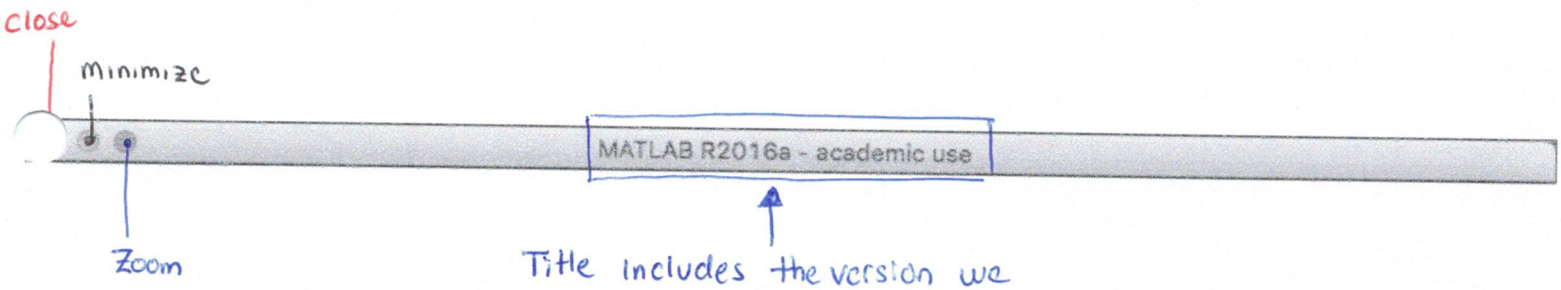
- In the Default Layout of the MATLAB Desktop, the Toolstrip sits between the Title Bar* and the Current Folder Toolbar, as seen in the figure below.



- The MATLAB Toolstrip organizes key features into a series of tabs each of which is divided into sections. Within each section we find a list of relevant controls that we can use to enhance our work.

Note:

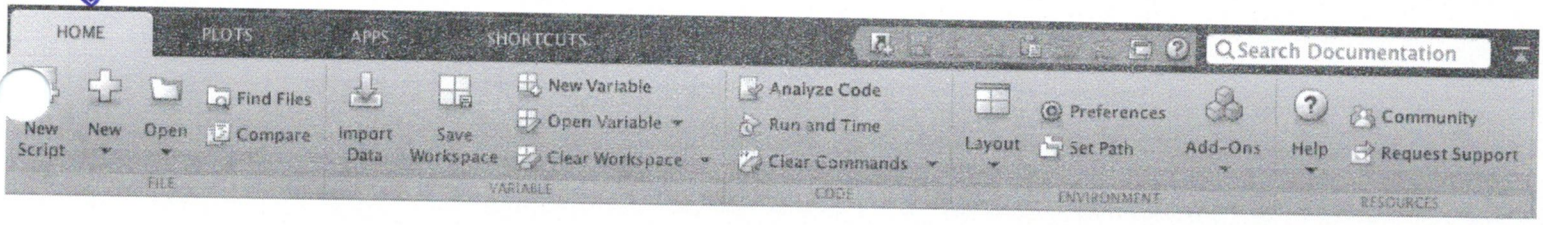
- The Title Bar of MATLAB's Desktop window is the frame at the very top of the desktop. The layout of this Title Bar depends on the operating system running on our computer (for the screen shot below, I was running MATLAB on a Mac with OS 10.13.2)



☐ We begin by considering the tabs. When we first open MATLAB, we will notice four tabs on the top of the toolstrip including

1. the Home tab
2. the Plots tab
3. the Apps tab
4. the Shortcuts tab

the Home tab is activated in this screen shot

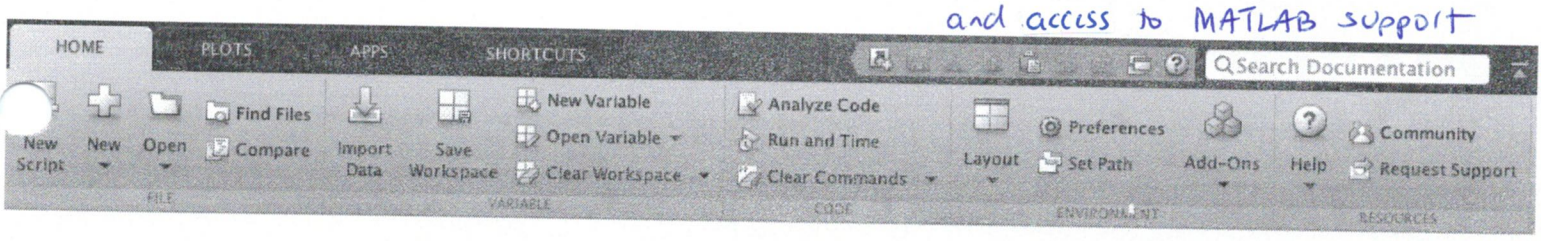


☐ These four tabs are always present in our desktop no matter what we are doing and thus are called Global tabs.

☐ The Home tab is where we go to find general purpose operations like creating new files, importing data, managing our workspace, and setting our Desktop layout.

☐ The Home tab is organized into five sections including

1. the File section: this section includes controls for file related operations
2. the Variable section: this section allows us to control manage our variables
3. the Code section: this section provides tools for managing and analyzing our code
4. the Environment section: this section provides controls for customizing the MATLAB Environm
5. the Resources section: this section provides support resources like documentation and access to MATLAB support

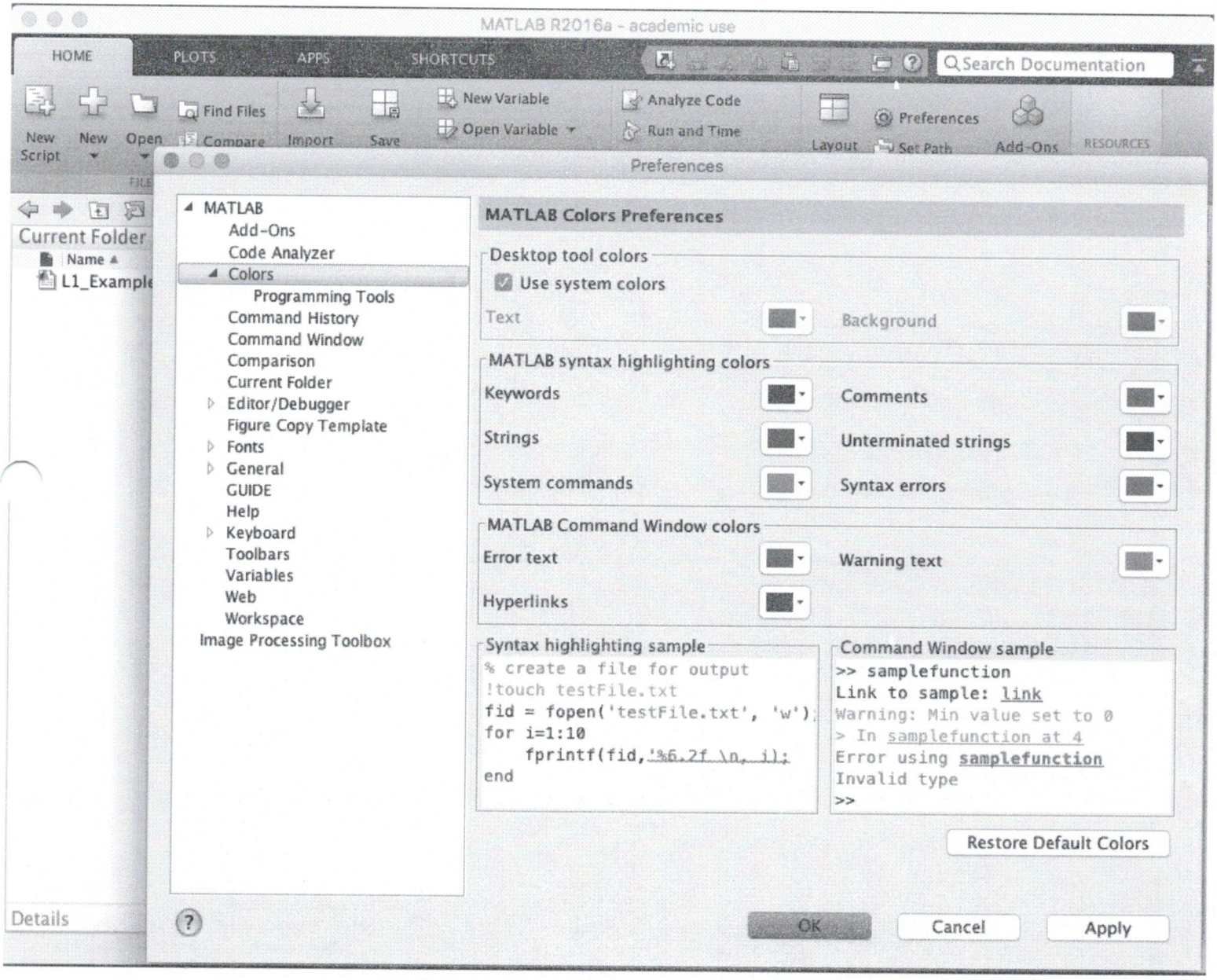


Note:

☐ The control buttons, drop down menus, and other user interface elements found in the toolstrip give us quick access to many of the most useful features of MATLAB we will need. By organizing these into tabs, we can search for relevant tools by context.

- Let's take a look at the Preferences control under the Environment section of the Home tab, as seen in the screen shot below.

- In this "Preferences" control button, we can customize many features of the MATLAB Enviroment



- In the Screen shot above, we see that we can control the colors that MATLAB uses to display an array of information