

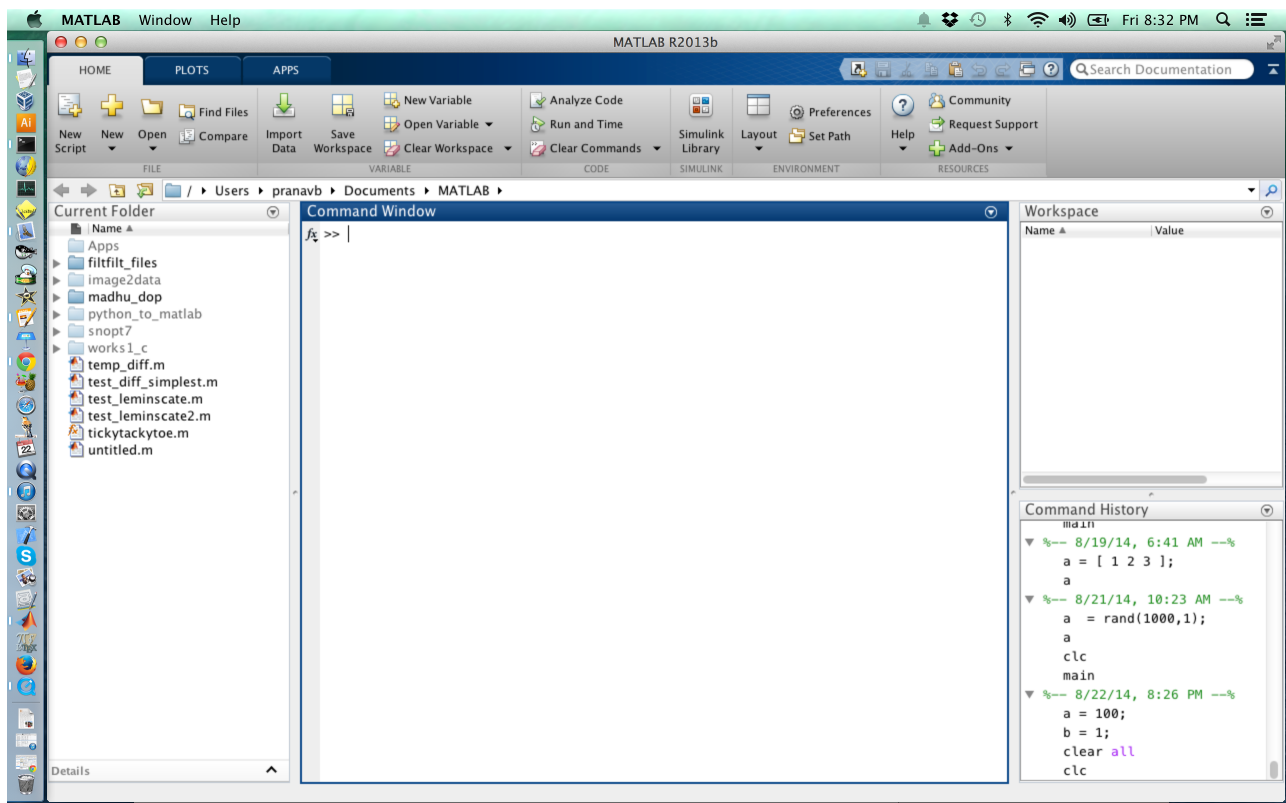
Pranav Bhounsule

MATLAB Basics. Using command line

1. Start MATLAB by navigating through your installed application or click the short cut icon for MATLAB shown next.

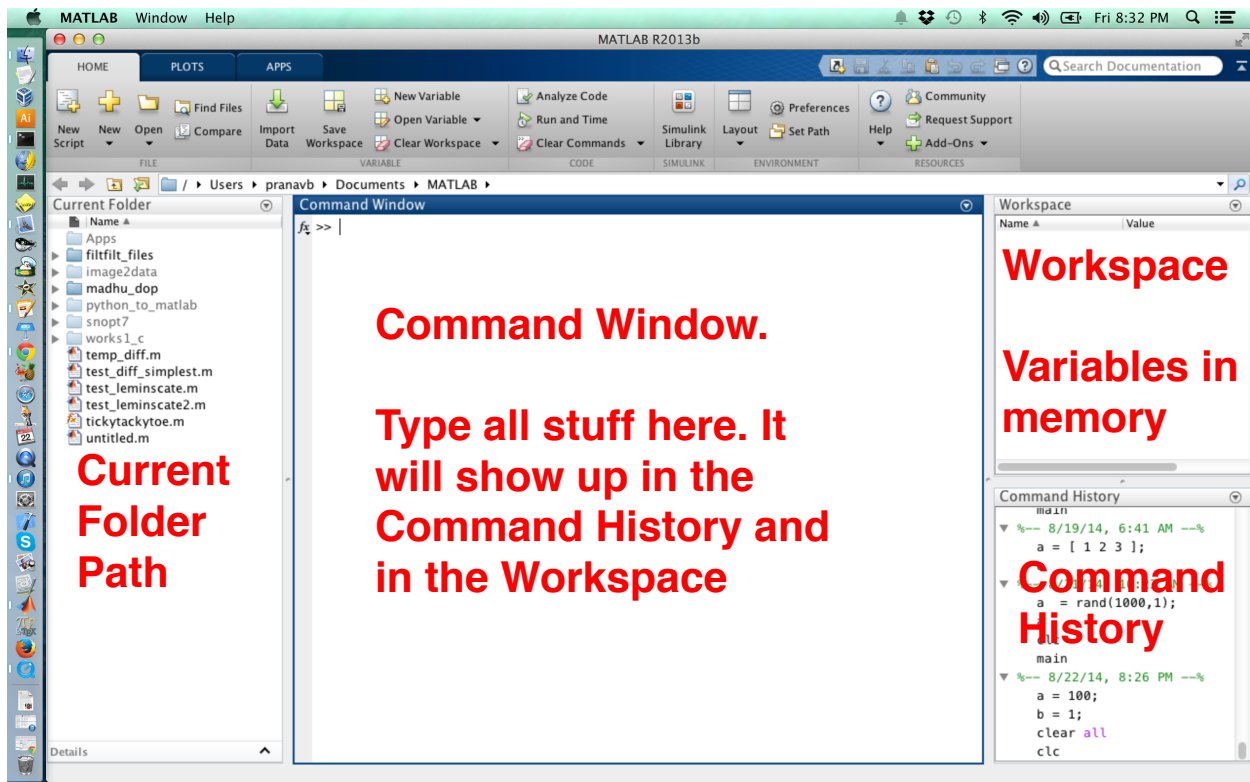


2. You will see the following screen once MATLAB loads up.



If you dont see something like that above, then go to Layout > Default Layout.

3. The different panels are marked below

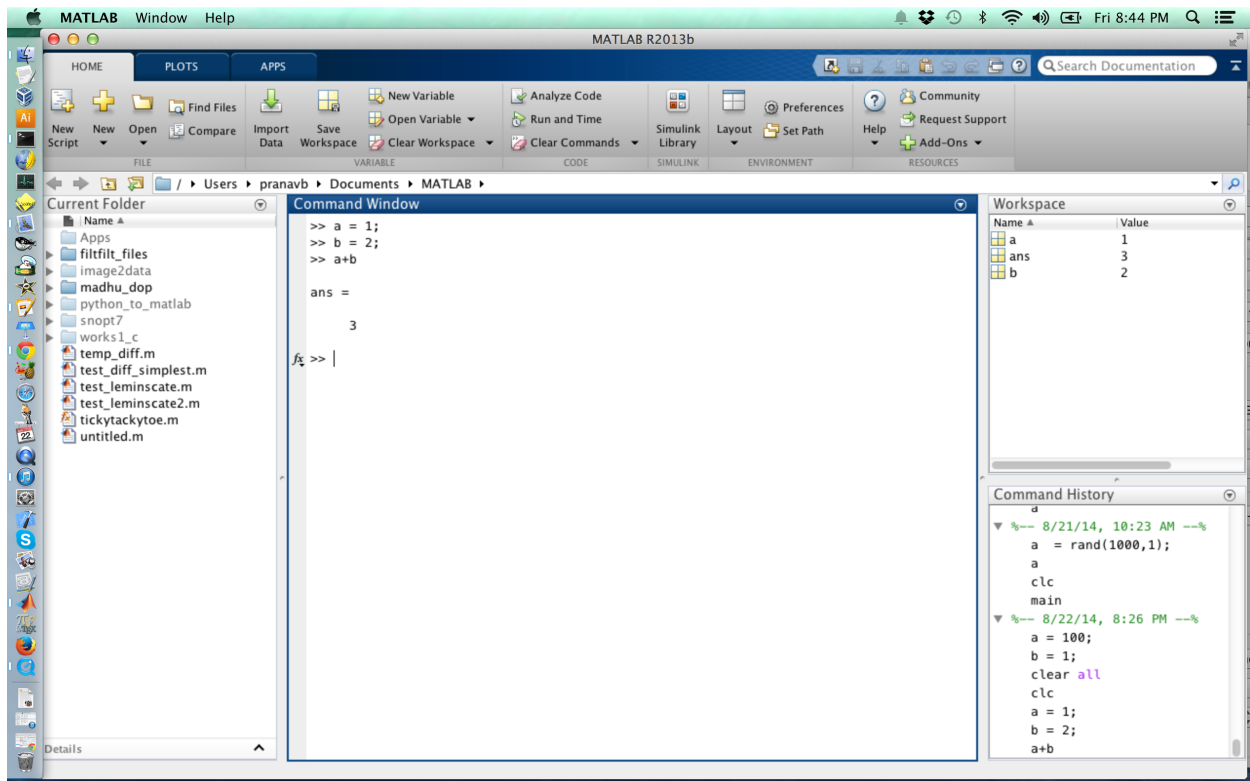


Note the Command Window has the symbol `>>`. This is where you will type. After typing you press ENTER to register the command.

4. For example.

```
>> a = 1;  
>> b = 2;  
>> a+b
```

The output is shown below.



NOTE a couple of things

- 1) The variables in memory appear in the workspace. (Top right panel)
- 2) The commands also appear in the Command History (Bottom right panel)
- 3) The sum $a+b$ is printed as 3.
- 4) Putting a semi colon suppresses the output. For example, if we put $a=1$ then we would get something like $a = 1$ displayed on the next line. Try it out. Similarly if we have put $a+b$; then we would not have got the answer.
- 5) We could also do something like

```
>> c = a+b;  
>> c
```

This would lead to something like

```
>> c =  
      3
```

From here on, I will only put things we type in the Command Window and the output.

6) Try these things

```
>> d = [1 2 3] %This is comment.
```

NOTES: Comments start with %. What you type after % is ignored by MATLAB. The above line produces the row matrix, i.e. one row and three column.

```
>> e = [1 2 3]'
```

This produces the column matrix, i.e. one column and three rows. The apostrophe here does the transpose of the matrix

Another way of writing the same matrix is

```
>> e = [1; 2; 3]
```

7)

One thing to note is that all variables entered in MATLAB are treated as matrices (hence the name MATrix LABoratory)

You can check the number of rows and columns by typing invoking the function size.

Try the following

```
>> size(a)
```

The answer will be 1 1, indicating this is a matrix with 1 row and 1 column.

Try these out.

```
>> size(d)
```

```
>> size(e)
```

8) If you need help on something just type help 'command name'

```
>> help size
```

9) Next, we turn to matrices.

Let us generate a random matrix of size 3 by 3.

```
>> A = rand(3,3)
```

This tells MATLAB to generate a random matrix of size 3 by 3. For more info you can type help rand

If you want the element in Row 1 and Column 1 you need to type A(1,1). For the element in Row 1 and Column 2 you type A(1,2).

10) Now find the inverse

```
>> B = inv(A)
```

To check the result try

```
>> A*B
```

As you know A into its own inverse will give the identity matrix.

11)

Now try this

```
>> A + d
```

You will get an error. MATLAB will tell say that Matrix dimensions must agree.

This is because you are adding a 3 by 3 matrix A with a 1 by 3 matrix d (you can check this by typing size(d) and size(A). (Note if you forgot what d is just type d in the command prompt to see what it is)

12)

Now try this

```
>> A*d
```

You will get an error. MATLAB will tell say that Matrix dimensions must agree.

This is because for multiplication to be compatible, the number of columns of A should be the same as the number of rows of d.

However something like

```
>> A*d'
```

will work because now the compatibility condition is met.

You will learn most by playing with MATLAB this way and searching for help on google. The book "Getting Started with MATLAB: A Quick Introduction for Scientists and Engineers" by Rudra Pratap is recommended. This book is available in the library.

13) Generating a list of numbers

```
>> t = linspace(0,6,100);
```

This generates the matrix t with 1 row and 100 columns in the range from 0 to 6.

14) Basic Plotting

```
>> x = sin(t)
```

x is also going to be a matrix with 1 row and 100 columns.

```
>> plot(t,x)
```

This generates a plot of x as a function of t

```
>> xlabel('x'); ylabel('t')
```

This puts labels on the x and y axis.

>> You can customize the plot. For e.g. if you want the plot to have x instead of continuous line try

```
>> plot(t,x,'rx');
```

```
>> figure(1)
```

14) Multiple Plots can be done using subplot. Try these commands

```
>> subplot(2,1,1); plot(t,x,'rx');
```

```
>> subplot(2,1,2); plot(t,y,'bo');
```

```
>> figure(1)
```

15) Vectors

```
>>a1 = [1 ; 2 ; 3];
```

```
>>b1 = [2 ;-1; 2];
```

Dot produce

```
>>c1 = dot(a1,b1);
```

```
>>d1 = cross(a1,b1);
```

16)

You will learn best by playing with MATLAB. Try these commands

```
>>eye(3)
```

```
>>ones(4,2)
```

```
>> clc
```

```
>>zeros(2,3)
```

```
>> clear all
```

The last command will clear variables in the workspace (see 2nd figure in this workout to locate the Workspace).

To learn more about a function just type help <function_name> at the command prompt.