

Hi Students,

Sorry in advance for the lengthy email!

Making plots and adding them to your homework:

You will be required to make plots from the given data for this homework. You may do this in Excel if you really want to, but I would recommend trying MATLAB (or another programming language of your choice, if you know any others). If you have never used MATLAB before, please refer to the instructions at the very end of this email. To get you started, I uploaded a template for a MATLAB script on the website as well. Please save your final plots as images and insert them into your written-up answers.

Quick update on LaTeX formatting:

1. Most of you wrote answers in red, probably because I put `\red{Your answer here.}` in the template. I just did that to highlight where your answers go. You can write out your answers without the `\red{}` command, which looks a bit more neat and might avoid some LaTeX complications. Anyways, either way works and as long as I can read your answers I won't take points off just because some formatting looks a bit "awkward".
2. In the overleaf template, you can upload an image you created (say 'yourPlotImage.png'), and then add it to the document with.

```
\begin{figure}
  \centering
  \includegraphics[width=0.48\textwidth]{yourPlotImage.png}
  \caption{Your caption of the plot.}
  \label{yourFigureLabel}
\end{figure}
```

You can then refer to your plot in the text with something like:

```
Figure \ref{yourFigureLabel} shows a few lines that look
very interesting.
```

Have a great long weekend everyone!

-Philipp

MATLAB basics (Only for help if you want to try MATLAB, but have never used it before!!!)

You can install MATLAB following the instructions at <https://matlab.ucsd.edu/>. The license is free for UCSD students.

Try to look at the examples below and re-create them in MATLAB. Then look at the template from the homework. You may want to play around a bit, and consult google about how to do a few things. I'll be around Wednesday after class (and the rest of the day) if you get stuck somewhere and need help.

You can define a variable like this: (the first line after >> is your input, the second line is MATLAB's output)

```
>> a = 3
```

```
a =
```

```
3
```

If you don't want the output to be printed, put a semicolon after your line.

```
>> b = 5;
```

You can always display variables by just typing them in the console. (All the variables you have created are also shown in the workspace window.)

```
>> b
```

```
b =
```

```
5
```

You can now do calculations with your saved variables. This is how you add variables.

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

```
c =
```

```
8
```

Similarly, you can multiply any variables.

```
>> d = a * c;
```

You can also display the value of a variable with the disp() command:

```
>> disp(d)
```

```
24
```

This is how you take exponents:

```
>> e = c^a
```

```
e =
```

```
512
```

This is how you do divisions:

```
>> f = e/d
```

```
f =
```

```
21.3333
```

You can add all these together to do more complex calculations in one line. Don't forget parentheses where you need them!

```
>> g = (a + b)^2 / d
```

```
g =
```

```
2.6667
```

You can create row vectors like this:

```
>> x = [a b c]
```

```
x =
```

```
3     5     8
```

And column vectors by putting a semicolon after each row:

```
>> y = [d; e; f]
```

```
y =
```

```
24.0000  
512.0000  
21.3333
```

You can convert a row vector to a column vector, and vice versa, using a ' after the vector. (This transposes the variable, i.e. switches rows and columns.)

```
>> z = y'
```

```
z =
```

```
24.0000  512.0000  21.3333
```

You can define a sequence of numbers with `start:end`

```
>> aa = 1:4
```

```
aa =
```

```
    1    2    3    4
```

You can also add a step size in the middle with `start:stepSize:end`

```
>> aaa = 1:0.5:4
```

```
aaa =
```

```
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000
    4.0000
```

You can multiply vectors with any scalar numbers, or add any scalars to vectors using the notation from above.

```
>> aa2 = aa * 2 - 1
```

```
aa2 =
```

```
    1    3    5    7
```

WATCH OUT: When you use multiplication or division on multiple vectors, then MATLAB assumes matrix operations!

```
>> bb = aa' * aa2
```

```
bb =
```

```
    1    3    5    7
    2    6   10   14
    3    9   15   21
    4   12   20   28
```

To do element-wise multiplication or division, you need a dot before your operator (so `.*` or `./`)

```
>> cc = aa .* aa2
```

```
cc =
```

```
    1    6   15   28
```

MATLAB also assumes matrix operations when using exponents. To apply an exponent to every entry individually, use `.^`

```
>> bb.^2
```

```
ans =
```

```
    50    150    250    350
   100    300    500    700
   150    450    750   1050
```

```
200      600      1000      1400
```

```
>> bb.^2
```

```
ans =
```

```
1     9     25     49
4    36    100    196
9    81    225    441
16   144    400    784
```

You can access entries in a vector by their indices using parentheses:

```
>> y(3)
```

```
ans =
```

```
21.3333
```

You can also access a slice of a vector by specifying the range of indices you want:

```
>> aa(1:3)
```

```
ans =
```

```
1     2     3
```

```
>> aa(2:end)
```

```
ans =
```

```
2     3     4
```

This also works in 2 dimensions for matrices:

```
>> bb(1:3,1:2)
```

```
ans =
```

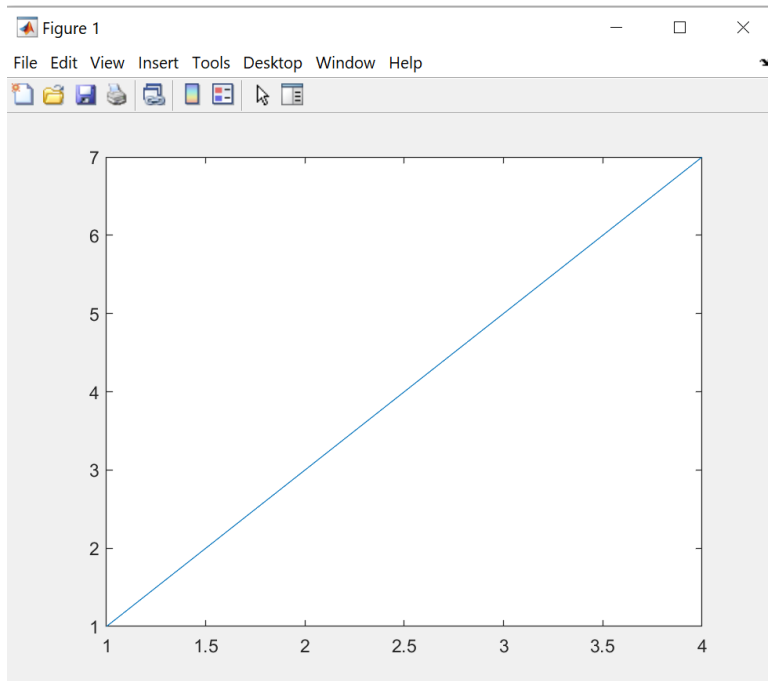
```
1     3
2     6
3     9
```

To create a figure, you can use the following. An empty figure window will open.

```
>> figure1 = figure(1);
```

Now, you can plot something. Here we plot the variable aa (these are our x-axis values) versus the variable aa2 (these are our y-axis values).

```
>> plot(aa,aa2)
```

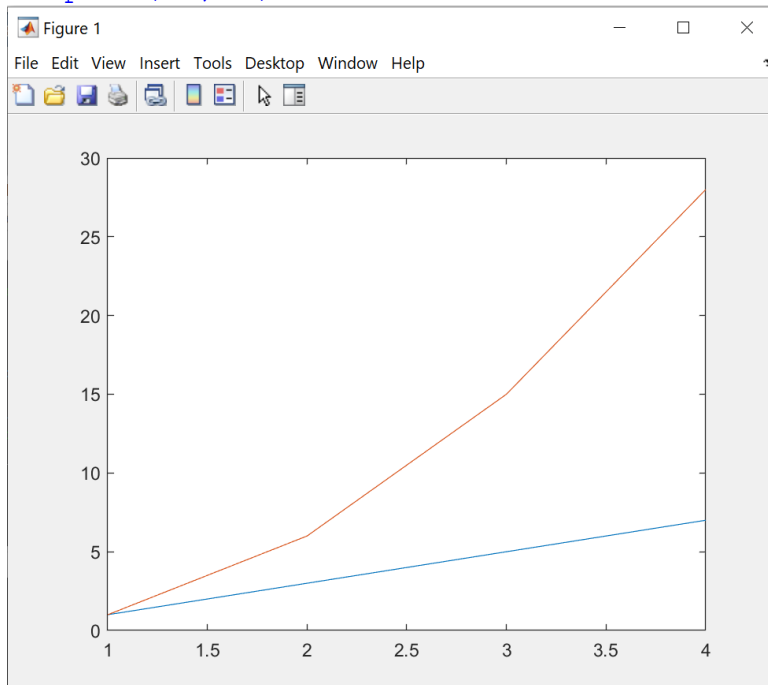


To plot a second line on the same figure, we need to tell MATLAB to not overwrite the previous plot with `hold on`.

```
>> hold on
```

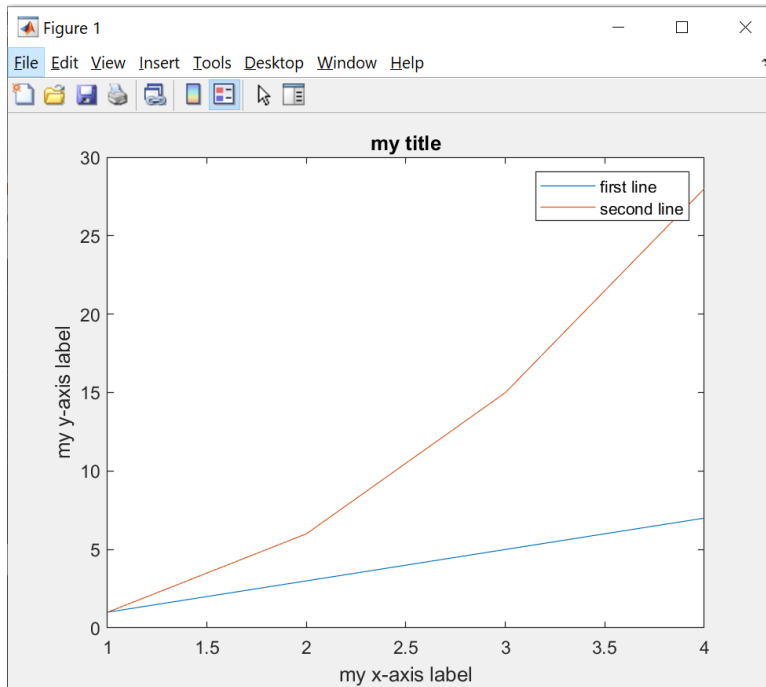
Now, we can plot another line

```
>> plot(aa,cc)
```



Now, you can add title, labels and a legend with the following:

```
>> xlabel('my x-axis label')
>> ylabel('my y-axis label')
>> title('my title')
>> legend('first line', 'second line')
```



Now you can save the plot as an image file with

```
>> saveas(figure1, 'myFigure.png')
```