

# CMPT 423/820

## Assignment 1 Question 1

- Seyedeh Mina Mousavifar
- 11279515
- sem311

### Ensuring your installation works correctly

Open this document in Jupyter Notebook. If you have opened this notebook, you should see a mixture of document and Python code. Near the bottom of the document, you should see a plot of a red line and a blue line.

```
In [4]: # As with a pure Python script, we can import modules.  
# This is a good one to know.  
  
import matplotlib.pyplot as plt
```

A notebook intermingles document and code. It's best to think of a notebook as a way of adding fancy internal documentation to a Python script (or other supported language). If you are new to Jupyter Notebooks, be sure to figure out the difference between a code cell, as immediately above, and a markdown cell, which is this one.

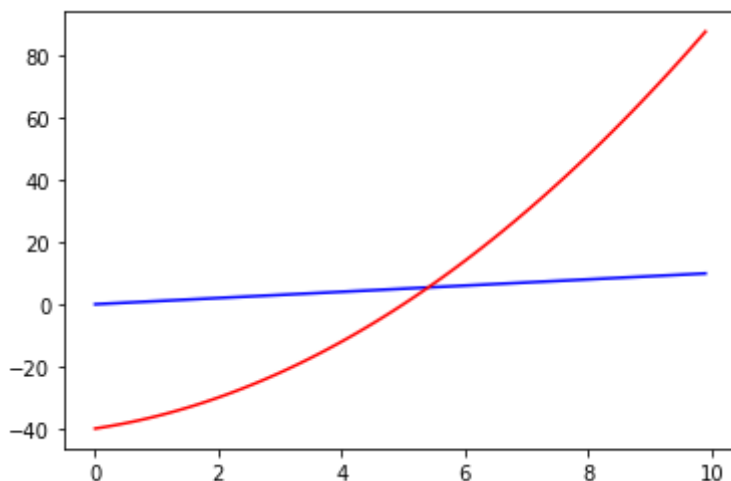
This is a `markdown` cell, which means it converts text to formatted display. The type of a cell can be determined by a drop down menu you can find near the top of this window. Go look. The Markdown language allows you a wide variety of formatting options. Check the Help menu for a link to external Markdown reference websites.

```
In [5]: # This is a CODE cell.  
# In this document, the code is Python, and all Python syntax rules apply.  
  
# Let's define a couple of variables using Python's List Comprehension notation  
  
xs = [i/10.0 for i in range(100)]  
ys = [x**2 + 3*x - 40 for x in xs]
```

The command to plot data comes from the MatPlotLib library. The `plot()` function adds the data to a figure. In plain Python, nothing appears before the `show()` is called, but it turns out that Jupyter can render the figure even if you don't call `show()`.

```
In [6]: # plot has a side-effect: to add the given data to an unseen set of axes
plt.plot(xs, xs, "b-")
plt.plot(xs, ys, "r-")

# Make the plot visible.
plt.show()
```



## Editing CODE cells

You can edit the code in any code cell, but for it to have an effect, you have to re-run the code. The keyboard command for that depends on your machine, but on my Mac, it's Ctrl-Enter. It's also available from the dropdown menu `Cell`, near the top of your window.

Note: If you re-run a single cell, the effect is limited to the current cell. You may rerun the whole notebook by finding an icon at the top of the window; the shape of the icon may be different depending on the computer you are using, but it may look like an arrow circling on itself, or a couple of triangles pointing to the right. This is also available in the Kernel menu as `Restart` and `Run All`.

Note: If you ever change a module that you import, or if you want to see the effects of a change to the whole notebook, use the `Restart` and `Run All` button.

## Producing a document to hand in

When using Jupyter Notebook, you'll want to keep a IPYN file. However, grading these files is awkward, so we will grade PDF files only. The best way to create a PDF is two steps:

1. Using the Jupyter Menu: `File > Download as... > HTML`
2. Open the downloaded HTML file with a Web-browser
3. Save the HTML document as PDF.

Jupyter does provide an option to download as PDF (via LaTeX) but this option is still too primitive; don't use it for submitting your work here.

# Task

1. Open this document in Jupyter Notebook.
2. Edit the first cell to add your name and student number
3. Re-run the entire document.
4. Following the instructions above, produce a document called A1Q1.pdf.
5. Hand the document in on Moodle Assignment 1.

## What to hand in

Your version of this notebook named A1Q1.pdf, containing your name and student number at the top.

## Evaluation:

1 mark: Your version of this notebook is named A1Q1.pdf, and it contains your name and student number at the top.