

CSCI 250 Visualization
Python File IO and Plotting Practice: L9
50 Points

Goal: Learn and practice concepts of plotting data and formatting the overall appearance using matplotlib and pyplot; to feel ready to provide graphs and plots in your Capstone project report.

You can create and run this from any device you prefer ... so you can use your Raspberry Pi, the Computers in the Classroom, your Laptop, etc.

Part 1 – Barebones:

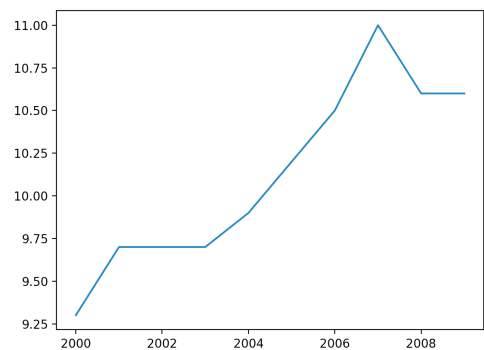
To start, we're going to make a simple plot with no labels or formatting. The sample data provided is the per capita consumption of mozzarella cheese (in pounds) each year from 2000 to 2009.

Load the Data

- a. Open up IDLE or something equivalent that you use, and import numpy as np and matplotlib.pyplot as plt.
- b. Go to Canvas and under Files, download "data.csv". For minimum pain, put it in the same directory as your python file. We're going to import that .csv file into our code and use it as our y axis.
- c. We've loaded a file into our code before in the Sound Lab (lab 3), but you can also find instructions on page 47 in your book or the lecture 9 slides in Canvas. Load the data, and make sure you specify how the data entries are separated (Hint: What does csv mean?)

Define your x and y axes, and plot!

- a. The .csv file's first row of data has the years and the second row has the actual cheese consumption. For practice slicing an array, create two variables and fill them with the contents from the above data (e.g., cheese and time).
- b. You can print this data to the screen to ensure you have correctly populated the variables.
- c. Lecture 9 and page 52 have information on how to plot, but I'll give a little info here too. What you want to use is plt.plot, where you put the x axis data in first and then the y, so plt.plot(x,y).
- d. If you try to run the code now, you won't get a plot to come out :(but that's just because you've basically defined the plot, you haven't asked the code to show the plot to you! Similar to how when you define a constant, table, or whatever you need to use print(), we need to use plt.show(). Just that, no stuff inside!
- e. Add the command plt.show() and you should end up with a blue line plot from 2000 to 2009 like the image to the right!



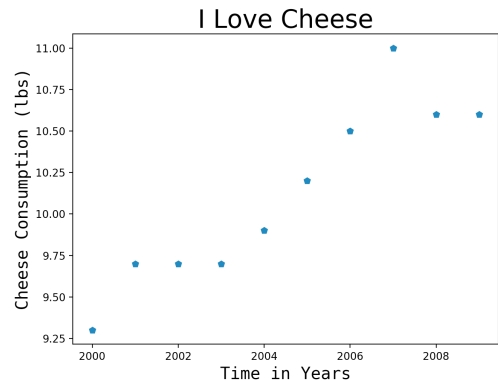
Part 2 – Make it Pretty!

Is this something that you would turn in with a lab report? Where are the axes labels, the title, why is the graph for some reason linear when I just gave you data points? So let's get to the formatting portion of your plots.

- Axes and Title:** There are multiple ways you can add axes and titles to a plot. The easiest way is to just use `plt.ylabel()`, `plt.xlabel()`, and `plt.title()`. However another way is to use `plt.gca()` and use its `set_xlabel()`, `set_ylabel()` and `set_title()`. The meaning of `gca` is **Get Current Axes** and it returns the object that controls a lot of the attributes of the plot currently used. Rather than use `plt.gca.set.blah`, just say `ax = plt.gca()`.

You can use either of these ways, but just make sure you give the axes and title the appropriate name, and that all of this goes BEFORE `plt.show()`!

- Color and Graph Format:** The graph shouldn't be a line, right? People don't eat mozzarella in a linear fashion throughout the year, and the data given is a representative of each year. So this should be a plot with points rather than a line! To fix this, put 'p' as the third input of your `plt.plot()`. If you want to change the color of the points as well, you can add another letter such as `r` for red to that string (ie. 'rp' instead of 'p').
- Extra Information:** If you want to know how to plot multiple graphs, make legends, put in error bars, and more, check out the book starting on page 52 :) or look around online! There are thousands of resources for you to choose from.
- Save the file:** explore the use of the command `plt.savefig('filename')`.



Part 3 – Practice with Dictionaries:

One popular data structure is the Python dictionary ... it works like a Map and has a key and a value pair. The goal for this part is to store the cheese data into a dictionary, sort it, and print the year and amount that the most pounds of cheese was consumed (imagine the power of this concept if our data set were much larger).

Be resourceful ... we briefly discussed this in class or you can look it up; for some reason, our book does not cover this fun data structure!

- Create a dictionary using keys = cheese and values = time
- Sort the dictionary in reverse order, such that the top element is the largest cheese value.
- Print a message to the screen similar to the below using the dictionary (not just hardcoded):
In x, x lbs of cheese was consumed!

Submission:

In canvas under the assignment for: Python File IO and Plotting Practice: L9
Submit Python file (.py file) and your final plot (image file or a screenshot)