PSYCH-GA.2211 / NEURL-GA.2201 – Fall 2026
Mathematical Tools for Cognitive and Neural Science

Homework 0

Due: 8 Sept 2026

Welcome to MathTools! This homework assignment is optional, but highly recommended. If you choose to submit answers, the TA's will give you feedback on code correctness, programming style, and submission format, which will help you perform better on later non-optional homeworks. [Click here for submission instructions]

All homeworks can be done in either MATLAB or Python. For Python, try to find equivalent numpy (often imported as `np`) and matplotlib.pyplot (often imported as `plt`) commands for the various MATLAB functions.

1. **Basics** Try the following quick exercises in your Jupyter Notebook or MATLAB command prompt.
   (a) Using Pythagorean theorem, calculate the length of a hypotenuse c of a right triangle given that the lengths of the other two sides a and b are both 1.
   (b) Calculate the perimeter p of a circle that has a diameter equal to the length of the hypotenuse we just got.
   (c) (MATLAB only) Press the up arrow key to get back the command you just entered. Now, add a semi-colon at the end of the line and press enter. What did this do?
   (d) (MATLAB only) Save what you just got into a MATLAB script. Name the variables. Run the script so it will output the hypotenuse length and the perimeter. Comment on what you just did.
2. **Vectors and plotting** Redo what we just did with a perspective of vectors.
   (a) Create two vectors that represent the two sides of a right triangle with lengths of 1 and 1 ($\vec{a}$ and $\vec{b}$). For simplicity, you may assume they are along the x and y axis and $\vec{b}$ starts where $\vec{a}$ ends (Note: we cannot strictly represent a shape with vectors since vectors don't care about their starting position. We will address this explicitly in 2d.). Then get the vector $\vec{c}$ for the hypotenuse. Calculate the length of $\vec{c}$. Check if it agrees with the previous calculation.
   (b) Bonus: Do the task in 2a at least 4 different ways, including (1) extracting the numbers and using the same procedure as previous problem, (2) element-wise exponentiation or element-wise multiplication, then `sum` function, (3) using `dot`, then `sum`, (4) try `norm`, (5) transpose the vector then product. (For Python, try `np.sum`, `np.dot` or `np.linalg.norm`. Note: you can get the transpose of an array `A` by `A.T`.)
   (c) Get the angle θ between $\vec{c}$ and $\vec{a}$. Get the unit vector of $\vec{c}$ in two ways (by shrinking the vector; by using the angle θ). Then create vectors that have the same length as $\vec{c}$ but with an angle that is 3.4 times the magnitude of θ.

(d) Plot all the vectors we have created. Check the documentation for `plot` in MATLAB or `plt.plot` in Python to see how you should modify the variables. `plotv` (MATLAB) is also a convenient tool for plotting vectors if you don't care about starting positions.

(e) Adjust the axis limits to leave some blank space in the figure. Add a title, legends, and axis labels.

(f) Save the plot.

3. **PhD financial life in NYC** Now let's learn to deal with a (slightly) bigger data set by modeling your financial life as a first year PhD student.

(a) Income from NYU: since you will have a steady income, create a column vector of 12 elements each with same amount of 3000. (Hint: use `ones` or `np.ones`).

(b) Assume you decide to TA from the second semester, therefore receiving an income of 4000 from the 7th month. Incorporate that into your income vector.

(c) Life expense: create a matrix of expenses with again 12 rows and each column representing a category (rent, grocery, clothing, fun). You can start with a constant expense as a first approximation. (Hint: use `repmat` in MATLAB or `np.tile` in Python).

(d) Calculate your savings vector for each month and calculate how much money you've got left after the whole year.

(e) Groceries and clothing will hopefully get cheaper as you gradually find the best options around. Suppose that by the end of the year your grocery expenses will only be 80 percent of the first month. And in between, your expense is linearly decreasing. Using `linspace(1,0.8,12)` in MATLAB or `np.linspace` in Python, create a "discount vector" and apply it to the grocery and clothing columns of the expense matrix.

(f) Use barplot (`bar` in MATLAB or `plt.bar` in Python) to show the effect of discounting on your monthly savings.

(g) Fun is always random! Add a vector of Gaussian noise with mean of 0 and standard deviation of 100 to your fun column using `randn` or `np.random.randn` in Python. Plot the savings column again to see the fluctuations. You can also check the shape of the Gaussian noise by using `histogram` in MATLAB or `plt.hist` in Python.

(h) Credit cards give reward points at different ratios for different categories of purchases. You want to compare 3 different companies to see which will give you the maximal reward. Specifically, their reward rate for different categories are as listed in table 1. Create a matrix representing all the credit systems and use matrix multiplication to get your monthly reward. Which card gives you the most reward for the first 6 months? Overall?

	rent	grocery	clothing	fun
card1	0	1%	2%	3%
card2	0	2%	2%	1%
card3	0.1%	1%	1%	2%

Table 1: see Problem 3(h)