Midterm review and Final Projects

Announcement:

No office hour this week due to midterm.

I will have **two office hours** next week with a signup sheet to have individual discussion about final projects.

Goals for today

- Explain the plan for the midterm
- Review for the midterm
- Discuss final projects
- Use matrix multiplication to simulate a recurrent network

Midterm Details

- Opens Wednesday at 10pm, closes Thursday at 10pm
- **We are NOT meeting for class on Thursday**.
- **Open notes, internet, and Python!** The only thing I ask is that you do not speak to others in the class about the midterm.
- One hour allotted, must be completed in one sitting.
- Includes ~20 multiple choice questions (~3 min/question)

Example question 1

Which of the following functions will return **True** if the integer **k** given as input is odd and **False** if it is even?

- A, B, C
- A, B
- A, C
- B
- B, C

```
## Function A
def is_odd(k):
    if k/2 != 0:
        return True
    else:
        return False

## Function B
def is_odd(k):
    if abs(round(k/2) - k/2) == 1:
        return True
    else:
        return False

## Function C
def is_odd(k):
    return bool(k%2)
```

Example question 2

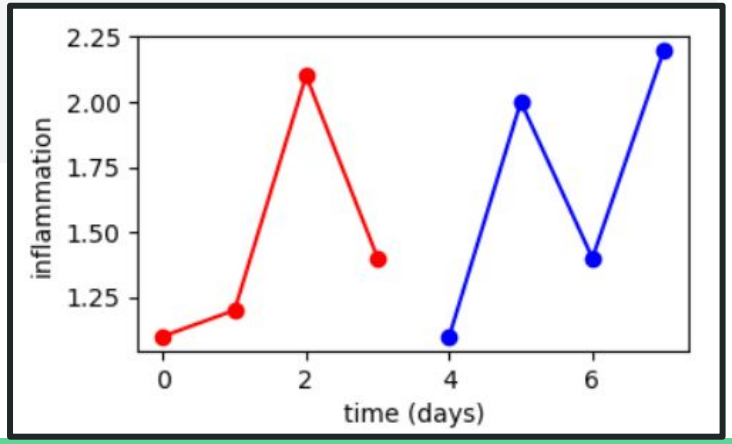
Which of the missing lines from the code below will create the following graph?

- Option A
- Option B
- Option C
- Option D

```
data = np.array([list(range(8)),[1.1,1.2,2.1,1.4,1.1,2.0,1.4,2.2]])
plt.figure(figsize=(4,2.3))

# Option A:
plt.plot(data[0,data[0,:]  >=4],data[1,data[0,:]  >=4] ,'o-b')
plt.plot(data[0,data[0,:]  < 4],data[1,data[0,:]  < 4] ,'o-r')
# Option B:
plt.plot(data[0,data[0,:]%2==1],data[1,data[0,:]%2==1] ,'o-b')
plt.plot(data[0,data[0,:]%2==0],data[1,data[0,:]%2==0] ,'o-r')
# # Option C:
plt.plot(data[0,data[0,:]  <=4],data[1,data[0,:]  <=4] ,'o-b')
# # Option D:
plt.plot(data[0,:],data[1,:] ,'o-b')

plt.xlabel('time (days)')
plt.ylabel('inflammation')
plt.show()
```

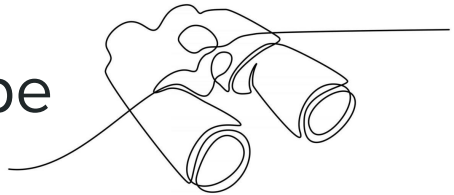


Objectives for the final project

- Choose a topic that you are interested and work on code related to that idea
- Plan out what is required for a computational solution to your chosen topic
- Follow best practices for coding style, documentation and code testing
- Work with different coding tools and packages
- Collaborate with two other students

The main goal of the project is to demonstrate that you can design, write, and implement code to fulfill some task that you choose.

Project Scope



- **There is not a specific amount of code that you have to write for the project.**
- **Your project must implement some new thing, that you design and write the code for.** To do so, you are expected to write new code that creates or adds some functionality.
- A project that appropriately responds to this call will have organized and documented code that:
  - Includes at least **three new functions (or methods)**
  - Uses code constructs such as loops and conditionals when needed
  - Imports code from available modules when needed
- **There is no need to perform a complex task!**

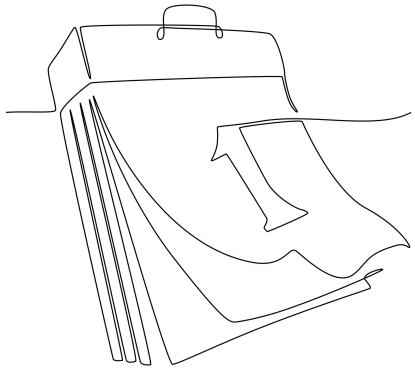
Project schedule

Weeks 7 & 8 — form teams and consider what you'd like to work on

Wednesday of Week 9 — one page proposal due

Weeks 9 & 10 — work on the project

Monday of finals week — turn in and share your project



Project topics

For your project, you can choose a topic that extends an application from one of the assignments, in-class notebooks, or propose your own topic.

Topics largely fall into two categories:

1. Write a program that will analyze a biological dataset and generate plots
2. Write a program that will simulate a simple model of a biological system

Example datasets:

- nytimes covid dataset https://github.com/nytimes/covid-19-data
- Human mortality/fertility/cause of death etc. https://www.demogr.mpg.de/en/publications_databases_6118/online_databases_6676/
- CDC / WHO / Allen Institute / Janelia Research Campus / NIH

Example models:

- Demographic model (Logistic map): https://en.wikipedia.org/wiki/Logistic_map
- Hopfield model https://en.wikipedia.org/wiki/Hopfield_network
- Random neural / gene-regulatory network

Sample projects

https://github.com/BILD62/StudentProjects

Modular design, like modular programming, is the approach of designing and building things as independent modules.

Project Organization

- **Notebooks**: good for interactive development
  - For when seeing the inputs and outputs of code running needs to be seen start to finish
  - file ends in .ipynb
- **Modules**: for storing mature Python code, that you can import
  - you don't *use* the functions in there, just define them
  - file ends in .py
- **Scripts**: a Python file for executing a particular task
  - this takes an input and does something start to finish
  - file ends in .py