Homework 3 **PSY4219/6219**
Due September 21 (in class) **Fall 2022**
16 points

Create a Jupyter notebook that answers each of the following questions and turn it in on Brightspace (you can use `Homework3.ipynb` as a starting point). You can answer all of these questions within one notebook.

Please do not turn in the `brain.mat` file with your homework (for Q1) – it is a large file.

Please make sure you comment (and/or use Jupyter markdown cells) in your code. And please use good programming style (especially points emphasized in class).

Q1. For this question, I want you to manipulate some brain imaging data (stored in a numpy array).

First, you need to load the data. I received these anatomical (structural MR) data from a colleague who processed them using Matlab, saving them as a .mat file. There is a module (part of scipy) that can load a `.mat` file (details are presented in the class slides).

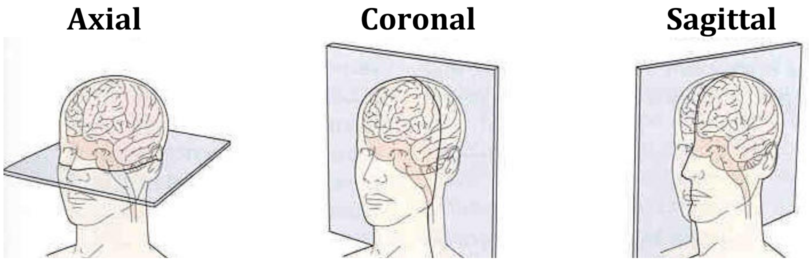
```
from scipy.io import loadmat

brainfile = loadmat('brain.mat')
brain = brainfile['data']
```

This file contains a 3D structural MR image of a human brain as a three-dimensional numpy array. The file is large.

(a) (4 points) Your first part of this assignment is to write a snippet of code to display a axial slice, a coronal slice, and a sagittal slice from the middle point of the 3D array (which will be roughly in the middle of the brain).

I want you to write code that will figure out the middle slice (of the axial, coronal, and sagittal slice directions) for any sized `brain` array (in other words, your calculations of the middle point should take place in the code, not with pencil and paper and hard-coded into your code). In other words, you need to check the shape of the numpy array and use those dimensions in your code.

Axial slice	slicing from the top of the head to the bottom of the head
Coronal slice	slicing from the front of the head to the back of the head
Sagittal slice	slicing from the left of the head to the right of the head



Axial Coronal Sagittal

You can display each 2D slice using the `imshow()` function from matplotlib (as described in class). If `axialslice` is a slice from the middle of the brain (from the `brain` numpy array) in the axial direction, then displaying that slice requires the following (we will talk about other tools for displaying and manipulating images later in the course – this is the simplest):

```
import matplotlib.pyplot as plt

plt.imshow(axialslice)
plt.title('axial')
```

Use `imshow()` to display an axial, coronal, and sagittal slice (from the middle of the `brain` numpy array). Remember that `imshow()` only takes a 2D array as input. Make sure you label (with a title) the slices as axial, coronal, and sagittal (if the figure on this page doesn't make that clear, feel free to Google around for some other example images of axial, coronal, and sagittal human brain slices).

You may recognize (even not being a neuroanatomy expert) that a couple of the slices are rotated (the bottom of the brain is not at the bottom of the image in coronal or sagittal images). Figure out how to rotate those numpy image slices by 90 degrees using `scipy.ndimage.rotate`.

(b) (4 points) In another code cell, I want you to write a bit of code that displays just a portion of the brain.

Assume you have an `(x, y, z)` coordinate that specifies one particular location in the 3D `brain` array and a value `N` that specifies the size of the subregion to show (using `imshow`).

For example, if `(x, y, z) = (100, 150, 50)` and N = 49, you would select a 49x49 portion of the brain centered at `(100, 150)` at `z` = 50 to display a 49x49 axial slice, a 49x49 portion of the brain centered at (150, 50) at `x` = 100 to display a 49x49 coronal slice, and a 49x49 portion of the brain centered at (100, 50) at `y`=150 to display a 49x49 sagittal slice.

Pulling out a portion of the `brain` array should be done using numpy array slicing. Display a portion of an axial, coronal, and sagittal slice given the values of `x`, `y`, `z`, and `N`. (Yeah, if `N` is even, the center point won't be centered, just do something sensible in that case.)

Your code should check that the combination of `(x, y, z)` and N produces a valid slice (in other words, the slice shouldn't try to index less than 0 and shouldn't try to index past the end of the array). You can just print out an error message if an invalid slice would result based on the values of `x`, `y`, `z`, and `N`.

Q2. Building off of Homework 1, I want you to now create a plot of the hemodynamic response function *HDR(t)* using basic `matplotlib`. Part (a) asks you to calculate the values of *HDR(t)* using a for loop and plot the function. Part (b) asks you to calculate the values of *HDR(t)* using the numpy vectorization techniques discussed in class and plot the function.

For both, make sure that you pick increments of *t* to make a smooth plot and that you pick a range of *t* to capture the full shape of the function.

$$HDR(t) = w\left(\frac{\beta_1^{\alpha_1} t^{\alpha_1} e^{-t\beta_1}}{\Gamma(\alpha_1)}\right) - (1-w)\left(\frac{\beta_2^{\alpha_2} t^{\alpha_2} e^{-t\beta_2}}{\Gamma(\alpha_2)}\right)$$

For the parameters, assume the following:
$\alpha_1 = 6$
$\beta_1 = 1$
$\alpha_2 = 16$
$\beta_2 = 1$
$w = 5/6$

Make sure that there is a plot title and that the axes are appropriately labeled.

(a) (4 points) Calculate the values of *HDR(t)* in a numpy array using a for loop and plot the function using the matplotlib techniques discussed in class.

(b) (4 points) Calculate the values of *HDR(t)* using the numpy vectorization techniques (in a single line of code) and plot the function.

Unexcused late assignments will be penalized 10% for every 24 hours late, starting from the time class ends, for a maximum of two days, after which they will earn a 0.