Homework 4 **PSY4219/6219**
Due Monday October 3 at 11:59pm **Fall 2022**
20 points

Q1. Using the code from Homework 3* for generating the *HDR(t)* function, I want you to save that *HDR(t)* function as a CSV file (using techniques from class), then load it back in, and redisplay the function using the information saved in the file (pasting in the code you wrote for the last assignment, but using the information loaded in from the file).

Your CSV file should have the follow structure:
1st line should be an informative note (< 50 characters) about what the file contains.
2nd line is the number of time steps in *HDR(t)*.
3rd line should be the names of the parameters of the HDR (from Homeworks 2 and 3), separated by commas.
4th line should be the values of the parameters (from Homeworks 1 and 2), separated by commas.
The remaining lines should be each value of *t* and its corresponding *HDR(t)*, separated by commas (in other words, if you had 1000 values of *t* and *HDR(t)* in your numpy arrays, these should be 1000 lines in the CSV file).

(a) (4 points) Save the file (write) as a CSV (using the structure described above).

(b) (4 points) Open the CSV file and load (read) its content into variables (for the number of time steps and the HDR parameters) and a numpy arrays for *t* and *HDR(t)*. Using your code from Q2 of Homework 3, display *HDR(t)* as a function of *t* (to demonstrate that what you saved to the file and read in from the file is the same as what you started with).

* *If you were unable to get your Homework 3 code to work, email Jason and we will supply code for you to use for this assignment.*

Q2. In class, we went over code for reading the data file `difdata.csv`. Here is example code that does that:

```
import csv
import numpy as np

with open('difdata.csv', 'r') as fp:
    csvreader = csv.reader(fp, delimiter=',')

    row = next(csvreader)
    Ntrials = int(row[0])

    icondition = np.zeros(Ntrials, dtype=int)
    ichoice    = np.zeros(Ntrials, dtype=int)
    iRT        = np.zeros(Ntrials, dtype=float)

    for i, row in enumerate(csvreader):
        icondition[i] = int(row[1])
        ichoice[i]    = int(row[2])
        iRT[i]        = float(row[3])
```

Recall that this code parses the data file into `icondition`, a numpy array which holds the condition (1 or 2) on each trial, `ichoice`, a numpy array which holds the choice accuracy (0=error or 1=correct) on each trial, and `iRT`, a numpy array which holds the response time (RT) on each trial.

(a) (6 points) Following the discussion from class, I want you to partition the data so that one two-dimensional array that holds the choices in condition 1 and the choices in condition 2 and another two-dimensional array that holds the RTs in condition 1 and the RTs in condition 2. **First**, do this using for loops. **Second**, do this using logical (Boolean) indexing.

(b) (6 points) Following the discussion from class, I want you to remove "outliers" based on RT, in this case trials where RT is outside some bound (RT<0.100 or RT>1.000). This will result in a list of arrays for the choices with outlier trials removed and a list of arrays for the RTs with the outlier trials removed. **First**, do this using for loops. **Second**, do this using logical (Boolean) indexing. Remember from discussion in class that here you will not be able to use a 2x500 numpy array because the number of resulting trials after removing outliers will be unequal (instead, use a list of numpy arrays).

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.