Homework 2 **PSY4219/6219**
Due September 14 in class **Fall 2022**
13 points

For this assignment, please do not use for loops or if/then/else statements, or define functions, or use any Python programming constructions that we have not yet discussed in class.

This assignment should be done within a single Jupyter notebook (`.ipynb`). Start with the `Homework2.ipynb` file that we have also posted online (and embed your answers where indicated).

Each question and subquestion should be answer in its own code cell and they should be proceeded by a properly-formatted markdown cell in your notebook.

Q1 (3 points). Research one of these modules that are part of The Python Standard Library (https://docs.python.org/3/library/index.html), choosing from the following: `datetime`, `pprint`, `fractions`, `statistics`, `time`, or `turtle` (note that while `turtle` is fun, it may not work on everyone's computer). Briefly describe what the module does in a markdown cell and demonstrate that you understand what it does in several lines of code in a code cell (describe in the markdown cell and show in a code cell at least three things the modules can do). Feel free to google around beyond the python.org web site to understand what the modules does (the python.org site can be terse and technical); just make sure you don't copy and paste code from a web site.

Q2 (3 points). fMRI (functional Magnetic Resonance Imaging) is used in neuroscience research to measure changes in local blood flow in the brain associated with local changes in neural activity associated with some manipulation.

For example, imagine a subject viewing a random display of dots while in an MR scanner. Those dots are stationary for a while. Then they move for a while. Then they are stationary again. Move for a while. Stationary for a while. And so on. We can imagine portraying this stimulus manipulation as a function of time as a "box car" (square wave) function that is low for some amount of time (stationary), high for some amount of time (moving), low for some amount of time (stationary), high for some amount of time (moving), and so on.

Ideally, we would like to find areas of the brain whose increasing and decreasing activity roughly matches a boxcar function time-locked with manipulated motion. In fact, there is such an area, discovered fifty years ago by Professor Jon Kaas in our department, called area MT. In fact, if we were measuring neural activity directly, using neurophysiological or electrophysiological techniques, we would see box car activity in the brain (with just some delay lag caused by the amount of time it takes for neural signals to propagate from the retina to area MT, plus a bit of time to integrate the visual signal to detect motion).

Things are different with fMRI. We are measuring changes in blood flow, which indirectly reflects changes in neural activity. An increase in neural activity leads to an increase in metabolic demand and this leads to an increase in blood flow. That change in blood flow is not instantaneous, but takes several seconds to unfold (whereas the change in neural activity unfolds over 10s to 100s of milliseconds). So finding areas in the brain that modulate activity with some experimental manipulation requires taking into account that hemodynamic lag. Doing so involves using what is called a hemodynamic response function.

We will talk later in the course about some of the signal processing that goes into doing such analyses. For now, I simply want you to calculate values from one common hemodynamic response function (there are more complex variants). (It is fine if you do not understand anything I wrote above about fMRI. I simply want you to calculate a function given below in Python code. All the above was just the scientific context.)

This function is called the double-gamma (because it is a weighted difference between two gammas) hemodynamic response function (HDR). Technically, the mathematics is the difference between two gamma distribution functions, but the hemodynamic response function is simply using these as functions having some shape, not as distributions per se.

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

The parameters of the first gamma distribution are α_1 and β_1, the parameters of the second gamma distribution are α_2 and β_2, and w is the weight. $\Gamma(x)$ is called a gamma function (not to be confused with the gamma distribution); it is a generalization of the factorial to real numbers, with $\Gamma(x) = (x\text{-}1)!$ when x is a positive integer. In Python, the gamma function is one of the special function in the `math` module.

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

Calculate (and print out) the value of the HDR function at t=0, t=1, and t=5 (these are in units of seconds).

Remember NOT to hard code the elements of the equation.

Q3 (4 points). For this question, I want you to demonstrate that you can use string literals / f-strings. I will often ask you to output results (on a Jupyter Notebook or as output from

Python script) and that output should be nicely formatted (readable). Furthermore, string literals / f-strings are also used to format output to text files. (As described in class, string literals / f-strings produce formatted strings without using the `.format` suffix.)

Produce a properly-formatted string literal and whatever other code is needed for each of the following (use a separate cell for each). Make sure your code prints the result using the `print()` function:

a) Given integer variables `a` and `b` (set to some value) compute a modulo division and print out the integer for the quotient followed by the remainder. So if `a=9` and `b=4`, the following would be printed out like this (you need to do the calculation in Python and then format the output with a string literal / f-string):

```
9 divided by 4 equals 2 with a remainder of 1
```

b) Get the value of pi from the math module. Using a formatted string literal (1) print it out as floating point number to 4 decimal places, (2) cast it as an integer and print it out taking up a total of 2 spaces, and (3) cast it as a string and print out pi <u>backwards</u>. (Printing the string backwards can be done in one step using the appropriate string slicing – do not use a for loop.)

Q5 (3 points). For this question, I want you to demonstrate that you can navigate strings, lists, and tuples.

a) Given the following string:

```
str = "The brown fox jumped over the lazy dog."
```

- Use a string slice to pull out just the word "`brown`" from the string and print it.
- Given the same string, use a string slice to pull out every other word from the string and print it.
- Given the same string, delete the word "`jumped`" from the string.

b) Given the following:

```
mylist = [["dog", (True, 17)], "house", [[3, 4], [5, 6], "dog"],
          ("dog", [1, "dog"])]
```

- Reference and print out (from the list) each instance of `"dog"`.
- Reference and print out the `"o"` from `"house"`.
- Reference and print out the `[5, 6]` list from the larger list.

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.