Homework 5

Due: Thursday Nov 21, 2019, 11:59pm

To submit:

- Save your work as a .ipynb file. Use the filename: **LASTNAME_HW5.ipynb**
- Upload the file to the *Homework 5* Activity in Moodle

Collaboration:

You may work with other students, but each student should write their own code for the assignment separately. In a comment at the top of your homework please indicate the people with whom you worked on the assignment.

Assignment goals:

In this assignment you will practice working with data from a recognition memory experiment, aggregating data across participants and creating a simple graph of the overall effect across the group. You will also add complexity to your recognition memory experiment code from last time, implementing methods to measure response times and collect confidence ratings. Finally, you'll practice presenting images using PsychoPy.

Problem 1: Calculate response time priming for all subjects in the in-class dataset (40 pts)

In class we reviewed how to calculate response time priming for consecutive pairs of items for a single subject, separately based on whether the items were presented as part of the same context (NonBoundary) or distinct contexts (Boundary). Update the code from class to calculate this measure for all of the subjects in the dataset. The data for all participants are in a folder on Moodle: PSYC027 INTEGRATE data.

Use the same exclusion criteria as we did in class: exclude pairs of trials where either item was incorrectly remembered or was a response time outlier. After calculating priming for all subjects, generate a bar graph (see `matplotlib.pyplot.bar`) showing the average response time priming effect across the group for the NonBoundary and Boundary conditions. Calculate a paired (also called *related*) samples t-test (`scipy.stats.ttest_rel`) between the NonBoundary and Boundary distributions. **Is there a significant difference in priming?**

Bonus (10 pts): Calculate priming for old items that were presented consecutively at test but were not consecutive at encoding--we reviewed this in class as a potential baseline measure. Compare priming separately for the NonBoundary and Boundary conditions to the Baseline condition. **Is there significant priming in either condition relative to the Baseline?**

Problem 2: Update your recognition memory experiment from HW4 (40 pts)
Update your recognition memory experiment from Homework 4 with the following improvements:

1) Code that will collect response times for each button press during the test phase (i.e. how long does it take the participant to respond Old/New (Target/Lure) after the word is presented on the screen?)
2) During the test phase, collect confidence on a 5 point scale:
    a) 1-Sure New / 2-Unsure New / 3-Don't Know / 4-Unsure Old / 5-Sure Old
3) Write code that will set the direction of the scale (1->5: New->Old vs. 1->5, Old->New) based on whether the subject number is even or odd.

If you haven't already done so, format your output from the test phase into a Pandas DataFrame, and then use Pandas functions to write out your DataFrame object to a file (e.g. `.to_csv` or `.to_excel`). Each row of the output should include the following data about the word that was presented:

- Subject Number
- Word
- Target Lure Status
- Response Button
- Response Old/New
- Confidence
- Response Time
- Hit/Miss/False Alarm/Correct Rejection

Problem 3: Use PsychoPy to present images (20 pts)
Up to this point we have focused on using PsychoPy to present word stimuli. Investigate the PsychoPy documentation to determine how you could present image stimuli instead. Write a simple loop to present the image stimuli contained in the `images.zip` archive on Moodle. Present each stimulus one at a time in the center of your window, for 1 second each (but make this timing parameter adjustable). Insert a 500 ms blank screen in between presenting each image. No need to collect any responses from the user. Your code should close the window once all images have been presented. Your code should be written so that it **reads** the filenames that are contained in the `images` folder and presents them--**do not hard code the image file names to be presented on each trial.**

Hint: there is a Python modulate called `os` that has a function `os.listdir()` that will probably be useful (there are many other ways to list the contents of a folder, so feel free to use another method if you find it).