Assignment 3: Learning and Memory PSY 306 (Winter 2024)

Name: Kunal Sharma

Roll Number: 2021331

PART B

1.

5 participants performed a task where they were presented with a list of words on a computer screen. The words were presented one at a time. There were a total of 16 words in each list. After the presentation of each word in one list, the participants were asked to recall that list. The given Excel file contains the data of the words recalled from each list, and each sheet contains data for a total of 198 trials per participant.
Each row in the 'LM_A3_Data1.xlsx' file represents one trial.
Each column represents the serial position of the recalled item/word. The data file contains a numerical identifier for each response made by the participant during the free recall period.
The integers in the data file correspond to the serial position of the recalled item. For example, integer 15 will correspond to the word which was presented at the 15th position in the list of 16 words.
-1 represents intrusions (word repeated from the previous list)
0 represents no item recalled at that position

Now do the following

A) Count the frequency of each item/word recalled by each participant. Plot a simple line and marker plot for each participant. Create a larger plot with 5 subplots, each containing the data for one participant. Identify and explain the effect associated with memory recall that can be observed from the data.

[3+1 Points]

Hint: Create a larger figure with five subplots (positioned as 5 rows x 1 columns);
Indicate the participant number on top of each subplot as the title.

Answer:

```
import pandas
import matplotlib.pyplot as plot

# Creates a figure with 5 subplots - Here No. of Rows = 5 and No. of Column = 1
Img, subImg = plot.subplots(5, 1, figsize=(8, 16))

i = 1

while i < 6:  # Loop to iterate on participant numbers

   # Reading the sheet
   Data_Frame = pandas.read_excel("LM_A3_Data1.xlsx", sheet_name=f"Sheet{i}", header=None)

   item_freq = [0] * 16  # List to store item frequencies
```

```python
index = 0
  # Loop to Iterate on each row in the DataFrame
  while index < len(Data_Frame):

      row = Data_Frame.iloc[index]

      col_index = 0

      # Loop to Iterate on each item in the row
      while col_index < len(row):

          item = row[col_index]

          if 0 < item <= 16:
              item_freq[item - 1] += 1  # Increment the corresponding frequency in item_freq

          col_index += 1

      index += 1

  # List of tuples for conversion to DataFrame
  freq_data = [(i + 1, freq) for i, freq in enumerate(item_freq)]

  # Converting list of tuples to DataFrame
  freq_df = pandas.DataFrame(freq_data, columns=['Item', 'Frequency'])

  # Sorting DataFrame by item number
  freq_df.sort_values(by='Item', inplace=True)

  # Plotting for each participant
  subImg[i - 1].plot(freq_df['Item'], freq_df['Frequency'], marker='o', linestyle='-')
  subImg[i - 1].grid(True)
  subImg[i - 1].set_xticks(range(1, 17))
  subImg[i - 1].set_title(f'Participant {i}')
  subImg[i - 1].set_xlabel('Item Position')
  subImg[i - 1].set_ylabel('Recall Frequency')
  i+=1

# Title
Img.suptitle("Participant's Frequency of Recalling Items")

plot.subplots_adjust(hspace=0.5,top=0.94)

plot.show()
```

Graph: To represent the participant's Frequency of Recalling Items.

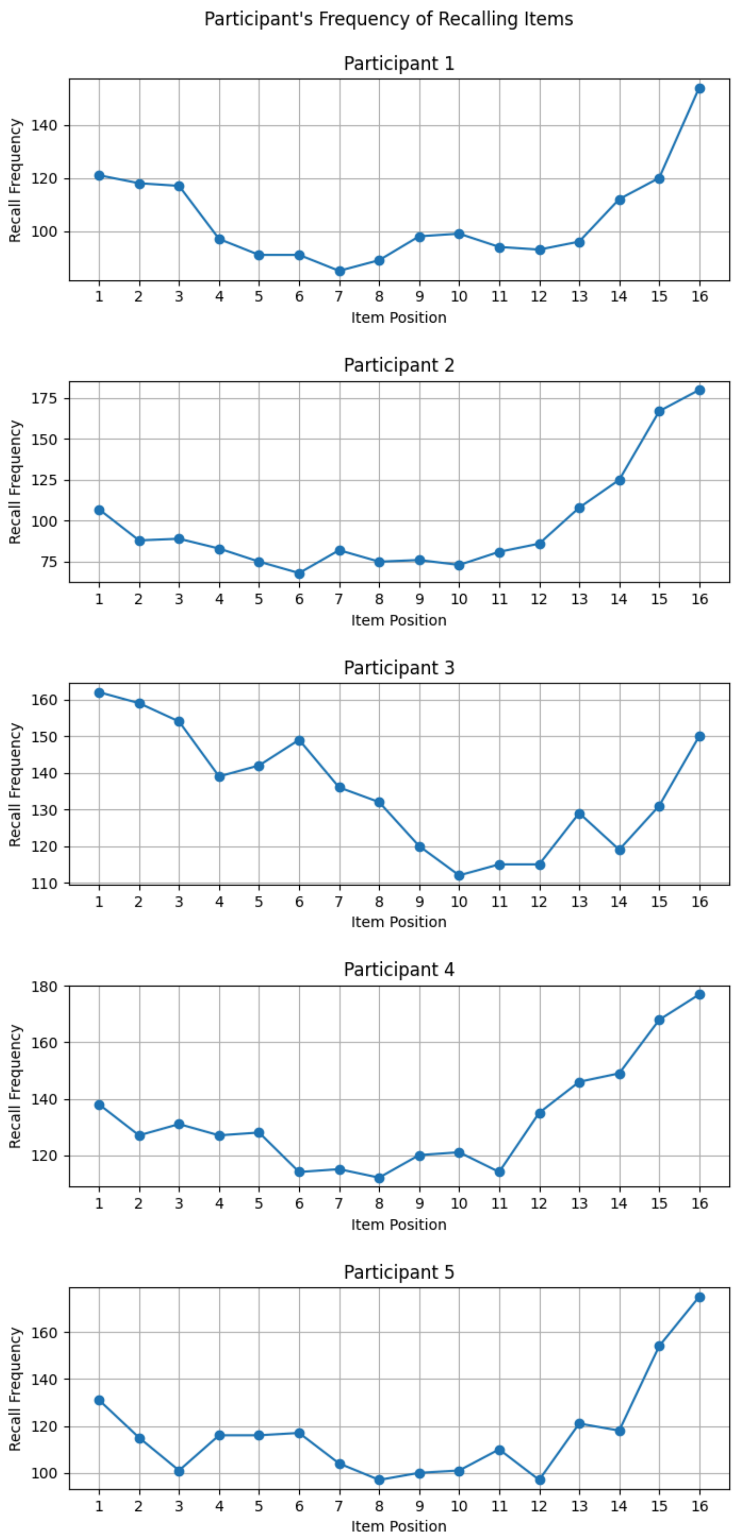

Figure 1.1 The Serial Position Curve depicts the relationship between the serial position of items and the frequency of their recall by participants. It illustrates that participants tend to remember words presented at the beginning of the list (primacy effect) and towards the end (recency effect) more effectively.

Observation:

The graph shows something called the serial position effect. It means that people remember things better if they're at the beginning or end of a list. Items at the start stick in our long-term memory longer, so we recall them better (that's the primacy effect). On the other hand, things at the end are still fresh in our short-term memory, so we can easily remember them (that's the recency effect). This happens because recent things are easier to recall, and they're not crowded out by other stuff yet.

B) For each participant, compute the distance from the last item recalled within each trial, excluding zeros and -1, by subtracting the item at the current index (ith) from the subsequent item at the adjacent index (ith+1). Count the frequency of each value and create a larger plot with five histograms (as subplots), each displaying the data for one participant.
What information can be drawn from the distribution (of the plot) with respect to the distance from the last recalled item?

[4+2 Points]

Hint: Create a larger figure with five subplots (positioned as 5 rows x 1 columns);
Indicate the participant number on top of each subplot as the title.

Answer:

```
import pandas
import numpy
import matplotlib.pyplot as plot

# Creates a figure with 5 subplots - Here No. of Rows = 5 and No. of Column = 1
Img, subImg = plot.subplots(5, 1, figsize=(8, 16))

i = 1

while i < 6:  # Loop to iterate on participant numbers

   # Reading the sheet
   Data_Frame = pandas.read_excel("LM_A3_Data1.xlsx", sheet_name=f"Sheet{i}", header=None)

   # Calculating the distance from the most recently recalled item for each trial
   dist = []

   j = 0

   while j < len(Data_Frame):

       row = Data_Frame.iloc[j]

       temp = 0

       k = 0

```

```
        while k < len(row):

            item = row[k]

            if item > 0:

                if temp > 0:

                    dist.append(abs(item - temp))

            temp = item

            k += 1

        j += 1
    # Data generation for the histogram.
    bins = numpy.arange(0, numpy.max(dist) + 2)

    Data, var = numpy.histogram(dist, bins=bins)

    # Plotting for each participant
    subImg[i - 1].bar(range(0, len(Data)), Data, align='center')

    subImg[i - 1].grid(True)

    subImg[i - 1].set_xticks(range(0, len(Data)))

    subImg[i - 1].set_title(f'Participant {i}')

    subImg[i - 1].set_xlabel('Distance from the most recently recalled item')

    subImg[i - 1].set_ylabel('Frequency')

    i+=1

# Title
Img.suptitle('Variation in the distribution of distances from the last recalled item among participants')

plot.subplots_adjust(hspace=0.5,top=0.94)

plot.show()
```

Graph: To represent the variation in the distribution of distances from the last recalled item among participants

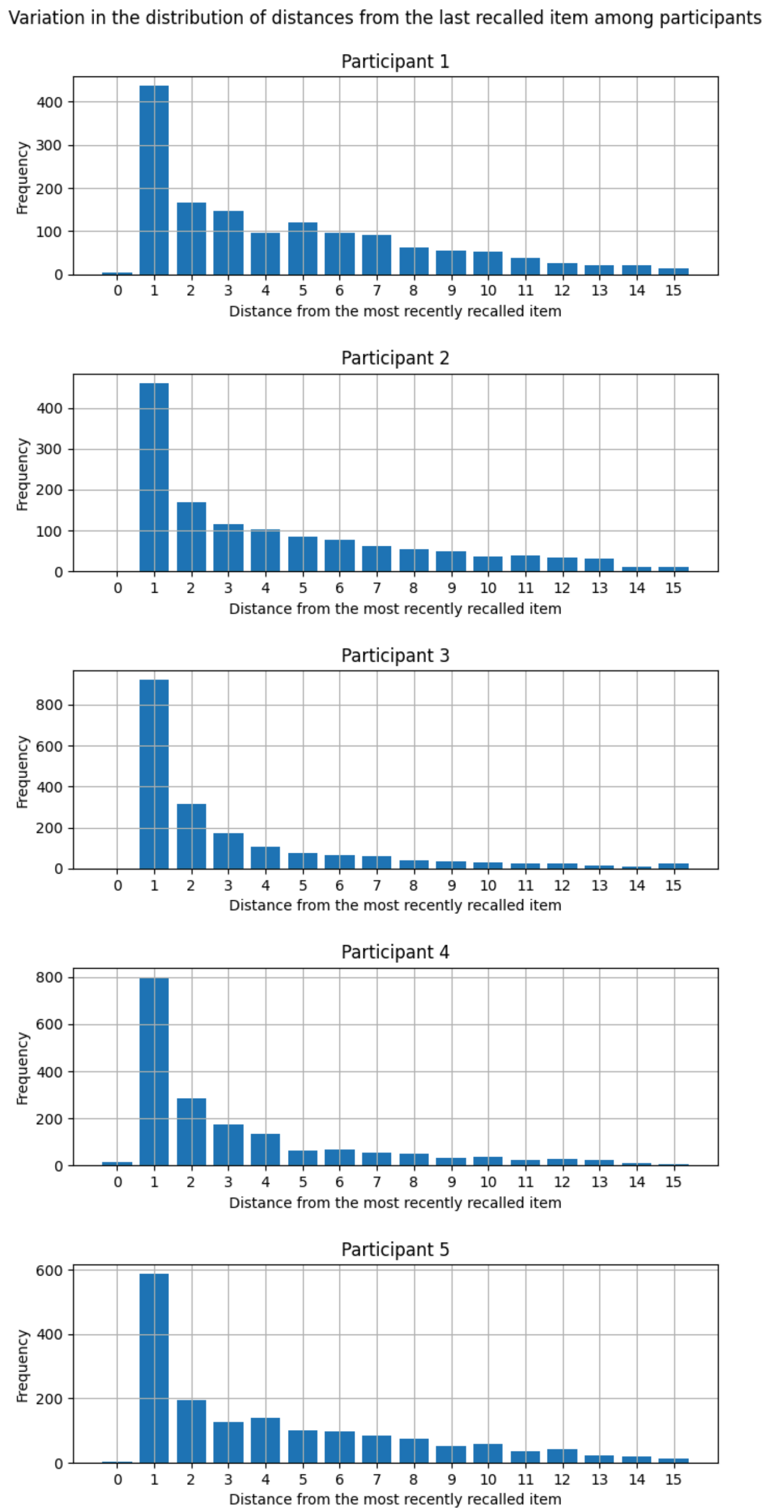

Figure 1.2 This graph shows that how frequently people remember things that are right beside each other in the list. You can notice that as the distance between the recalled items gets bigger, the number of times they're recalled next to each other. This means it's less likely for people to remember things that are far apart from each other. It indicates that people are inclined to remember things in groups, which is called the serial clustering effect.

Observation:

The highest frequency when the distance between items is 1 means that most often, people remember something right next to what they just recalled. This hints at a habit of remembering things in groups or chunks instead of scattered randomly throughout the list, known as the serial clustering effect. As the distance from the last remembered item increases, the chances of recalling items decrease. This decline in frequency as the distance grows also supports the idea of serial clustering, where items remembered recently tend to help recall nearby items. When items are farther apart, the connection between them weakens, making them harder to remember.

2.

Use the data given in LM_A3_Data2.xls | An experimenter recorded and pre-processed EEG data from 20 participants on an auditory oddball task playing them standard and deviant tones. The interstimulus intervals between the two tones were manipulated at levels – 0.75 s, 1.5s, 3s, 8s, and 9s as the EEG traces evoked by both standard and deviant tones were measured (1000 Hz sampling rate) from the participants' brains. Each Excel file sheet has data for both standard tone (beginning from cell 'B3') and deviant tone (beginning from cell 'B25'). For each of the above tones there is a 20 (participants) x 100 (time point) matrix in each sheet. Do the following…

[All figures/schematics should be properly labelled and should have accompanying captions/legends to provide all information necessary to interpret the same…]

A) Make a figure with five subplots – one for each interstimulus interval. In each subplot, graph the average EEG response (across 20 participants) from standard and deviant tones in blue and red, respectively.

[4 points]

Answer:

```
import pandas
import numpy
import matplotlib.pyplot as plot

# List containing interstimulus Interstimulus_Intervals
Interstimulus_Intervals = [0.75, 1.5, 3, 8, 9]

# Creates a figure with 5 subplots - Here No. of Rows = 5 and No. of Column = 1
Img, subImg = plot.subplots(5, 1, figsize=(8, 16))

# Loop to iterate on each interstimulus interval
i = 0

while i < len(Interstimulus_Intervals):

   interval = Interstimulus_Intervals[i]

   if i == 0:
       sheet_name = "0.75s"
```

```
    elif i == 1:
        sheet_name = "1.5s"
    elif i == 2:
        sheet_name = "3s"
    elif i == 3:
        sheet_name = "8s"
    else:
        sheet_name = "9s"

    # Reading the sheet
    Data_Frame = pandas.read_excel("LM_A3_Data2.xls", sheet_name=sheet_name, header=None)

    # Retrieving EEG data corresponding to standard and deviant auditory stimuli
    std_data = Data_Frame.iloc[2:22, 1:101].to_numpy()

    dev_data = Data_Frame.iloc[24:44, 1:101].to_numpy()

    # Calculating the average EEG response across participants
    Average_Standard = numpy.mean(std_data, axis=0)

    Average_Deviant = numpy.mean(dev_data, axis=0)

    # Plotting average EEG response for standard and deviant tones
    subImg[i].plot(Average_Standard, color='blue', label='Standard Tone')

    subImg[i].plot(Average_Deviant, color='red', label='Deviant Tone')

    subImg[i].set_title(f'Interval: {interval} s')

    subImg[i].set_xlabel('Time Point')

    subImg[i].set_ylabel('EEG Response')

    subImg[i].legend()

    i += 1

# Set figure title
Img.suptitle('Average EEG Response for Standard and Deviant Tones')

plot.subplots_adjust(hspace=0.5)

plot.show()
```

Graph: To represent average EEG Response for Standard and Deviant Tones

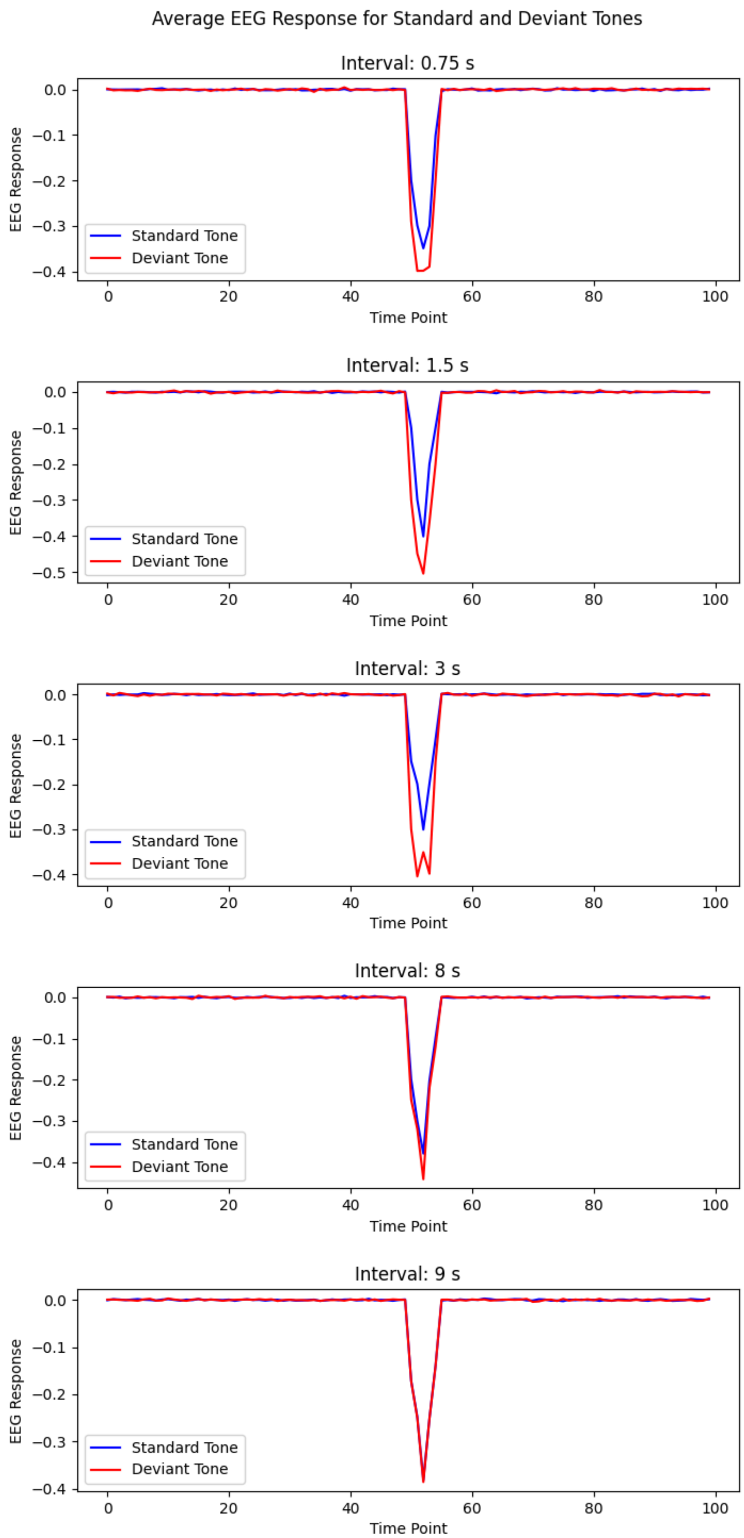

Figure 2.1 This graph shows how the brain's electrical activity changes when there's a gap between sounds. People listened to standard (blue) and deviant (red) tones and were asked to tell them apart. The V-shaped part in the graphs shows how the brain sorts out and recognizes sounds. When the gap between sounds gets longer, it becomes harder for people to tell the tones apart.

B) **Analyze the data from each interstimulus interval statistically and report the time scale of echoic memory. Explain the cognitive science consistent rationale behind the calculation and reported time scale.**

[4 + 2 points]

Hint: Carefully inspect the correctly created figure above for clues.

Answer:

```
import pandas
import numpy
from scipy import stats

# List containing interstimulus Interstimulus_Intervals
Interstimulus_Intervals = [0.75, 1.5, 3, 8, 9]

var=0
# Loop through each interstimulus interval
index = 0
prev_p_value = None
prev_interval = None
var = 0

while index < len(Interstimulus_Intervals):
   interval = Interstimulus_Intervals[index]
   df = pandas.read_excel("LM_A3_Data2.xls", sheet_name=f"{interval}s", header=None)
   # Extracting EEG data for standard tones
   std_data = df.iloc[2:22, 1:101].to_numpy()
   # Extracting EEG data for deviant tones
   dev_data = df.iloc[24:44, 1:101].to_numpy()
   # Average EEG response across participants for standard tones
   Average_Standard = numpy.mean(std_data, axis=0)
   # Average EEG response across participants for deviant tones
   Average_Deviant = numpy.mean(dev_data, axis=0)
   # Paired t-test for every individual time point
   t_statistic, p_values = stats.ttest_rel(Average_Standard, Average_Deviant, axis=0)
   print(f"Interstimulus Interval: {interval}s, p-value: {p_values}")

   # Finding range where the p-value transitions from significant to non-significant
   if p_values > 0.05 and var == 0:
       ans = f"Echoic Memory Time Scale: {interval}s"
       var=1
   index += 1

# Printing the Calculated Echoic Memory Time Scale
print(ans)
```

Output on the console:

Interstimulus Interval: 0.75s, p-value: 0.03637948493406719

Interstimulus Interval: 1.5s, p-value: 0.033057418772821534

Interstimulus Interval: 3s, p-value: 0.04053897882280644

Interstimulus Interval: 8s, p-value: 0.06741863112782308

Interstimulus Interval: 9s, p-value: 0.3523953527790259

Echoic Memory Time Scale: 8s

Observation:

Echoic memory, a component of sensory memory, briefly holds onto auditory information. Determining its duration involves identifying the interval where the brain no longer distinguishes between standard and deviant tones. Through paired t-tests at various time points within each interval, I assessed the discernibility of the tones. This analysis revealed that after 8 seconds, indicated by a p-value exceeding 0.05, the brain's ability to differentiate between tones diminishes. This investigation is rooted in cognitive science principles, seeking insights into auditory stimulus processing. Understanding the temporal constraints of echoic memory sheds light on how the brain processes auditory information.