**Assignment 2: Neuroscience of Decision Making PSY 3/507 (Monsoon 2023)**

**Name: Tushar Chandra**

**Roll Number: 2021211**

**Instructions:** Please write your own responses and do not copy or lift text/code from any source (including the paper). If you are referring to credible external sources other than the attached paper for your answers, please cite those sources (within the body of text and the provide a reference list at the end) in the APA citation format (<https://www.mendeley.com/guides/apa-citation-guide>). Word limits given are indicative and less than the indicated numbers may also be used.

**Please download this MS word question-cum-response template to TYPE your answers and feel free to add sheets as required. Convert this document to a PDF and rename the file: name\_roll no. before submitting. Please note that answers in this template only will be evaluated and hand-written or scanned answer sheets will not be evaluated.**

**[Strict deadline for submission: 28.10.2023 Saturday 10.00 PM]**

**Q1)**

**Fill the following form** [**https://forms.gle/BYotcMpVrtcVn1QQ9**](https://forms.gle/BYotcMpVrtcVn1QQ9)

**Q2)**

**A researcher conducted a random dot motion discrimination task with 2 different motion coherence levels as different conditions. 30 participants performed 100 trials in each condition and the evidence accumulation was recorded. Column 1 of cell array = Condition 1 and each cell of Column 1 has one participant’s data. Each cell has a 100 x 1000 matrix. Each row of the matrix = one trial for 1000 ms. The evidence accumulation starts from 300 and reaches the decision threshold at 600.The same convention applies to data from Column 2.**

**The data is attached herewith: Assignment2-NDM.mat’.**

**[ links about importing MATLAB data arrays into Python and R**

[**https://in.mathworks.com/help/matlab/matlab\_external/matlab-arrays-as-python-variables.html**](https://in.mathworks.com/help/matlab/matlab_external/matlab-arrays-as-python-variables.html)

[**https://stackoverflow.com/questions/11671883/importing-an-array-from-matlab-into-r**](https://stackoverflow.com/questions/11671883/importing-an-array-from-matlab-into-r) **]**

**Now solve the following. Insert a figure (wherever required) and paste the MATLAB/Python/R code for the same. Any figure must provide all information necessary to interpret it including axes labels, captions/legends (simple figure titles as captions are not enough).**

1. **i) Create two subplots for the two conditions separately. Plot the mean evidence accumulation (across 100 trials) of each participant in grey colour and the mean evidence accumulation (across 30 participants) in blue colour in each subplot.**



# Create a figure with two subplots

fig, (plt1, plt2) = plt.subplots(1, 2, *figsize*=(12, 6))

# Plot condition 1

c1Mean = np.mean(c1, *axis*=0)

totalc1Mean = np.mean(c1Mean, *axis*=0)

# Plot the mean of each participant's trials in grey

line1 = plt1.plot(c1Mean.T, *color*='grey', *alpha*=0.5, *label*='MEA of each participant')

# Plot the mean evidence accumulation of the 30 participants in blue

line2 = plt1.plot(totalc1Mean, *color*='blue', *linewidth*=2, *label*='MEA of 30 participants')

# Labels for the plot

handles = [line1[0], line2[0]]

labels = [

'MEA of each participant',

'MEA of 30 participants',

]

# Add labels to the plot

title1 = *f*'Condition 1 \nMean Accumulation Rate: {mean\_rate\_condition1*:.2f*} (unitless)'

plt1.set\_title(title1)

plt1.set\_xlabel('Time (ms)')

plt1.set\_ylabel('Mean Evidence Accumulation (ms)')

plt1.legend(*handles*=handles, *labels*=labels, *loc*='lower right')

# Plot condition 2

c2Mean = np.mean(c2, *axis*=0)

totalc2Mean = np.mean(c2Mean, *axis*=0)

# Plot the mean of each participant's trials in grey

line1 = plt2.plot(c2Mean.T, *color*='grey', *alpha*=0.5, *label*='MEA of each participant')

# Plot the mean evidence accumulation of the 30 participants in blue

line2 = plt2.plot(totalc2Mean, *color*='blue', *linewidth*=2, *label*='MEA of 30 participants')

# Labels for the plot

handles = [line1[0], line2[0]]

labels = [

'MEA of each participant',

'MEA of 30 participants',

]

# Add labels to the plot

title2 = *f*'Condition 2 \nMean Accumulation Rate: {mean\_rate\_condition2*:.2f*} (unitless)'

plt2.set\_title(title2)

plt2.set\_xlabel('Time (ms)')

plt2.set\_ylabel('Mean Evidence Accumulation (ms)')

plt2.legend(*handles*=handles, *labels*=labels, *loc*='lower right')

plt.tight\_layout()

plt.show()

**Calculate the rate of mean evidence accumulation (across participants) for each of the two conditions separately and report on the title of each subplot.**

**[Hint: Fit a straight line to the data between the starting point of mean evidence accumulation and the earliest point of reaching the threshold of maximum evidence to calculate the average rate.]**

**[4 + 3 points]**

****

# Create a figure with two subplots

fig, (plt1, plt2) = plt.subplots(1, 2, *figsize*=(12, 6))

# Plot condition 1

c1Mean = np.mean(c1, *axis*=0)

totalc1Mean = np.mean(c1Mean, *axis*=0)

# Plot the mean of each participant's trials in grey

line1 = plt1.plot(c1Mean.T, *color*='grey', *alpha*=0.5, *label*='MEA of each participant')

# Plot the mean evidence accumulation of the 30 participants in blue

line2 = plt1.plot(totalc1Mean, *color*='blue', *linewidth*=2, *label*='MEA of 30 participants')

fit\_x = np.arange(300, 601)

fit\_y = np.polyfit(np.arange(300, 601), totalc1Mean[300:601], 1)[0] \* fit\_x + np.polyfit(np.arange(300, 601), totalc1Mean[300:601], 1)[1]

# Calculate the rate of mean evidence accumulation for condition 1

mean\_rate\_condition1 = (totalc1Mean[600] - totalc1Mean[300]) / (fit\_x[-1] - fit\_x[0])

line3 = plt1.plot(fit\_x, fit\_y, *color*='red', *linewidth*=2, *label*='Fitted Line')

# Labels for the plot

handles = [line1[0], line2[0], line3[0]]

labels = [

'MEA of each participant',

*f*'MEA of 30 participants',

'Fitted Line',

]

# Add labels to the plot

title1 = *f*'Condition 1 \nMean Accumulation Rate: {mean\_rate\_condition1*:.2f*} (unitless)'

plt1.set\_title(title1)

plt1.set\_xlabel('Time (ms)')

plt1.set\_ylabel('Mean Evidence Accumulation (ms)')

plt1.legend(*handles*=handles, *labels*=labels, *loc*='lower right')

# Plot condition 2

c2Mean = np.mean(c2, *axis*=0)

totalc2Mean = np.mean(c2Mean, *axis*=0)

# Plot the mean of each participant's trials in grey

line1 = plt2.plot(c2Mean.T, *color*='grey', *alpha*=0.5, *label*='MEA of each participant')

# Plot the mean evidence accumulation of the 30 participants in blue

line2 = plt2.plot(totalc2Mean, *color*='blue', *linewidth*=2, *label*='MEA of 30 participants')

fit\_x = np.arange(300, 601)

fit\_y = np.polyfit(np.arange(300, 601), totalc2Mean[300:601], 1)[0] \* fit\_x + np.polyfit(np.arange(300, 601), totalc2Mean[300:601], 1)[1]

# Calculate the rate of mean evidence accumulation for condition 2

mean\_rate\_condition2 = (totalc2Mean[600] - totalc2Mean[300]) / (fit\_x[-1] - fit\_x[0])

line3 = plt2.plot(fit\_x, fit\_y, *color*='red', *linewidth*=2, *label*='Fitted Line')

# Define a list of handles and labels with colors to add to the plot

handles = [line1[0], line2[0], line3[0]]

labels = [

'MEA of each participant (Grey)',

*f*'MEA of 30 participants (Blue)',

'Fitted Line (Red)',

]

# Add labels to the plot

title2 = *f*'Condition 2 \nMean Accumulation Rate: {mean\_rate\_condition2*:.2f*} (unitless)'

plt2.set\_title(title2)

plt2.set\_xlabel('Time (ms)')

plt2.set\_ylabel('Mean Evidence Accumulation (ms)')

plt2.legend(*handles*=handles, *labels*=labels, *loc*='lower right')

common\_caption = (

'This graph represents the mean evidence accumulation in two different conditions.\n'

'In Condition 1, the mean evidence accumulation of each participant\'s trials is shown in grey,\n'

'while the mean evidence accumulation of all 30 participants is displayed in blue.\n'

'Additionally, a fitted line is plotted in red.\n'

'Condition 2 follows a similar pattern.\n'

'The rate of mean evidence accumulation is reported in the title of each subplot.'

)

fig.text(0.5, -0.2, common\_caption, *ha*='center', *fontsize*=12)

plt.tight\_layout()

plt.show()

**ii) To compare the mean rate of evidence accumulation (across participants) between the two conditions, conduct an appropriate statistical test and report the results with test statistics and p values. What can be concluded about the motion coherence in the two conditions from the results? [2+1 points]**

**(Hint: If the data in each of the two groups follow a normal distribution, use a parametric statistical test for testing the difference of two independent group means. Otherwise, use a suitable non-parametric counterpart of the parametric test.**

[**https://in.mathworks.com/help/stats/hypothesis-tests-1.html**](https://in.mathworks.com/help/stats/hypothesis-tests-1.html)**). Normality assumption can be checked using Lilliefors test.)**

Based on the Mann-Whitney U test results, which yielded a p-value of 1.0, we fail to reject the null hypothesis. This suggests that there is no significant difference in the mean rate of evidence accumulation between the two conditions. Therefore, we cannot conclude that there is a significant difference in motion coherence between the two conditions, as the evidence accumulation rates appear to be similar.

```

Mann-Whitney U Test Results:

Statistic: 0.0

p-value: 1.0

Fail to reject the null hypothesis. Since there is no significant difference in the mean rate of evidence accumulation between the two conditions.

```

import scipy.stats as stats

# Calculate the mean rate of evidence accumulation for each condition

meanRateC1 = (totalc1Mean[600] - totalc1Mean[300]) / (fit\_x[-1] - fit\_x[0])

meanRateC2 = (totalc2Mean[600] - totalc2Mean[300]) / (fit\_x[-1] - fit\_x[0])

# Perform the Mann-Whitney U test

stats, p\_value = stats.mannwhitneyu(meanRateC1, meanRateC2)

alpha = 0.05 # it is predetermined threshold used to determine the level of statistical significance in hypothesis testing

# Report the results

print(*f*'Mann-Whitney U Test Results:')

print(*f*'Statistic: {stats}')

print(*f*'p-value: {p\_value}')

# Check if the p-value is less than the significance level

if p\_value < alpha:

print(*f*'Reject the null hypothesis. Since there is a significant difference in the mean rate of evidence accumulation between the two conditions.')

else:

print(*f*'Fail to reject the null hypothesis. Since there is no significant difference in the mean rate of evidence accumulation between the two conditions.')

1. **i) From the data, calculate reaction time (RT) for all 100 trials of each participant. Divide the time axis in 20 bins and plot the mean RT for all 30 participants for both conditions separately. Plot one histogram of the reaction time distribution for each condition. Interpret the findings with respect to motion coherence. [8 points]**

****

import scipy.io

import numpy as np

import matplotlib.pyplot as plt

# Load the data from the .mat file

data = scipy.io.loadmat('NDM\_Assignment2.mat')['NDM\_Assignment2']

# Split the data into two conditions

c1 = data[:, 0]

c2 = data[:, 1]

# Create a figure with two subplots

fig, (plt1, plt2) = plt.subplots(1, 2, *figsize*=(12, 6))

# Calculate reaction time (RT) for each trial and participant in both conditions

rt\_condition1 = np.zeros((30, 100))

rt\_condition2 = np.zeros((30, 100))

threshold = 600 # Decision threshold

for participant in range(30):

for trial in range(100):

rt\_condition1[participant, trial] = np.argmax(c1[participant][trial][300:] >= threshold)

rt\_condition2[participant, trial] = np.argmax(c2[participant][trial][300:] >= threshold)

# Calculate mean RT for all 30 participants in each condition

mean\_rt\_condition1 = np.mean(rt\_condition1, *axis*=0)

mean\_rt\_condition2 = np.mean(rt\_condition2, *axis*=0)

# Plot histograms for Condition 1

plt1.hist(mean\_rt\_condition1, *bins*=20, *color*='green', *alpha*=0.7, *range*=(0, 20))

plt1.set\_title('Mean Reaction Time Distribution (Condition 1)')

plt1.set\_xlabel('Mean Reaction Time (ms)')

plt1.set\_ylabel('Frequency')

# Plot histograms for Condition 2

plt2.hist(mean\_rt\_condition2, *bins*=20, *color*='green', *alpha*=0.7, *range*=(50, 220))

plt2.set\_title('Mean Reaction Time Distribution (Condition 2)')

plt2.set\_xlabel('Mean Reaction Time (ms)')

plt2.set\_ylabel('Frequency')

# Set a detailed caption for the plot

fig.suptitle('Comparison of Mean Reaction Time Distribution in Two Conditions')

plt.show()

**Theory:** The histogram analysis of mean reaction times (RT) in conditions with differing motion coherence levels suggests that higher coherence leads to faster perceptual decisions, reflected in shorter RTs. Conversely, lower motion coherence in Condition 2 results in prolonged RTs, indicating the challenge of detecting less coherent motion. These findings underscore the critical role of motion coherence in influencing the speed and efficiency of visual decision-making.

**ii) From the data, calculate the median bias of evidence accumulation (across participants) separately for condition 1 and condition 2. Conduct the appropriate statistical test to compare both with a reasoning. Report relevant test statistics and p values. [Use the hint in the previous question to conduct the test] [2 points]**

import scipy.io

import numpy as np

import scipy.stats as stats

# Load the data from the .mat file

data = scipy.io.loadmat('NDM\_Assignment2.mat')['NDM\_Assignment2']

# Split the data into two conditions

c1 = data[:, 0]

c2 = data[:, 1]

# Calculate the median bias of evidence accumulation for each condition

median\_bias\_condition1 = np.median(c1, *axis*=(0, 2))

median\_bias\_condition2 = np.median(c2, *axis*=(0, 2))

# Perform an appropriate statistical test (e.g., Mann-Whitney U test) to compare both conditions

statistic, p\_value = stats.mannwhitneyu(median\_bias\_condition1, median\_bias\_condition2)

alpha = 0.05

# Report the results

print(*f*'Mann-Whitney U Test Results:')

print(*f*'Statistic: {statistic}')

print(*f*'p-value: {p\_value}')

# Check if the p-value is less than the significance level

if p\_value < alpha:

print(*f*'Reject the null hypothesis. There is a significant difference in median bias between the two conditions.')

else:

print(*f*'Fail to reject the null hypothesis. There is no significant difference in median bias between the two conditions.')

Theory: The median bias of evidence accumulation was calculated for both conditions, providing insights into decision-making tendencies. A Mann-Whitney U test compared these medians, revealing whether motion coherence levels significantly influence decision processes.