**Q2)**

**Two independent groups of participants (19 each) performed an Iowa Gambling task. Their**

**data are named Data1 and Data2 respectively. Each sheet in the attached excel file**

**(NDM\_Assignment3) contains data of one group. Each row represents one participant and**

**each column represents one trial. There are a total of 4 decks and 100 trials. Decks 1 and 2**

**yield immediate and steady rewards, but they are also characterised by unpredictable**

**occasional losses that can result in negative long-term outcomes. Decks 3 and 4 offer**

**relatively lower and steady immediate rewards, accompanied by even lower and less**

**unpredictable occasional losses, leading to favourable long-term outcomes. Solve the**

**following.**

**Insert a figure (wherever required) and paste the MATLAB/Python/R code for the same. All**

**figures should be properly labelled and MUST have accompanying captions to provide all**

**information necessary to interpret the figures**

**A. Divide the trials into five equal sized blocks and then calculate the mean proportion**

**(across participants) of advantageous cards and disadvantageous cards selected by the**

**participants. Create two subplots as part of a single figure. Each subplot should depict**

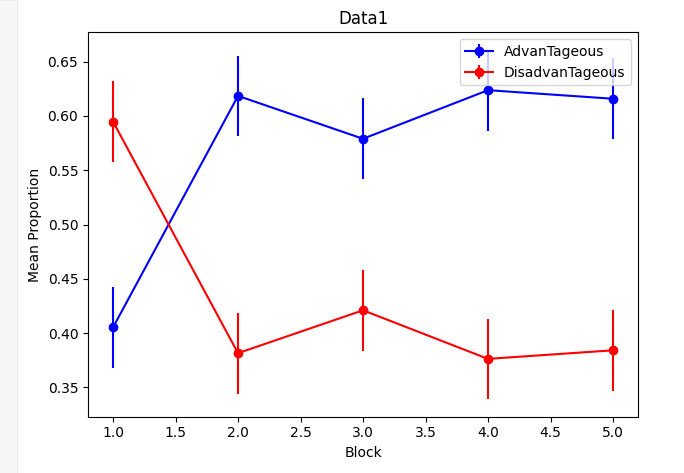
**the mean proportions across 5 blocks for each independent group using line and**

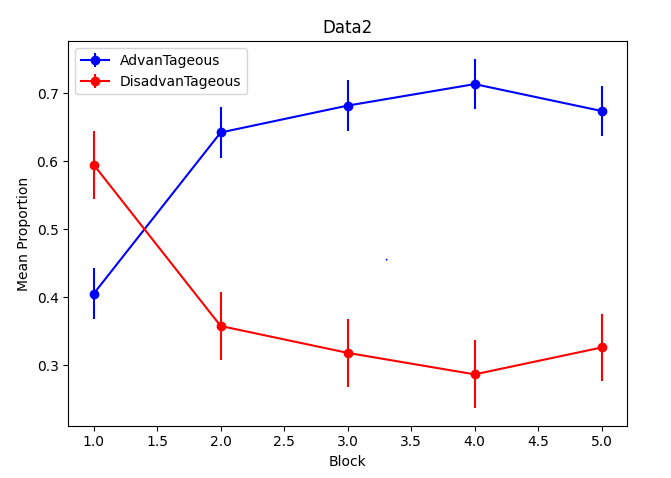
**marker plots. Mark advantageous cards with blue colour and disadvantageous cards**

**with red colour. Include standard errors of the mean as error bars at each marker**

**location in both subplots. Interpret the figures in the context of affective decision**

**making.**

****



**Code is:**

#!/usr/bin/env python

# coding: utf-8

# In[4]:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# In[15]:

# For data 1

data1 = pd.read\_excel('NDM\_Assignment3.xlsx', header=None)

data1 = data1.drop(0,axis =1)

data1

# In[14]:

# For data 2

data2 = pd.read\_excel('NDM\_Assignment3.xlsx', header=None,sheet\_name=1)

data2 = data2.drop(0,axis =1)

data2

# In[23]:

group = data1

AdvanTageous\_Data1 = []  # Change variable name from advantageous to AdvanTageous\_Data1

DisadvanTageous\_Data1 = []  # Change variable name from disadvantageous to DisadvanTageous\_Data1

# Divide the trials into five equal-sized blocks

for i in range(0, group.columns.size, group.columns.size // 5):

    block = group.iloc[:, i:i + group.columns.size // 5]

    # Calculate the mean proportion of AdvanTageous and DisadvanTageous cards

    AdvanTageous\_Data1.append((block.isin([3, 4])).sum().sum() / block.size)

    DisadvanTageous\_Data1.append((block.isin([1, 2])).sum().sum() / block.size)

# Calculate standard errors of the mean

AdvanTageous\_err\_Data1 = np.std(AdvanTageous\_Data1) / np.sqrt(len(AdvanTageous\_Data1))

DisadvanTageous\_err\_Data1 = np.std(DisadvanTageous\_Data1) / np.sqrt(len(DisadvanTageous\_Data1))

# Plotting

plt.errorbar(range(1, 6), AdvanTageous\_Data1, yerr=AdvanTageous\_err\_Data1, fmt='-o', color='blue', label='AdvanTageous')

plt.errorbar(range(1, 6), DisadvanTageous\_Data1, yerr=DisadvanTageous\_err\_Data1, fmt='-o', color='red', label='DisadvanTageous')

plt.title('Data1')

plt.xlabel('Block')

plt.ylabel('Mean Proportion')

plt.legend()

plt.tight\_layout()

plt.show()

group = data2

AdvanTageous\_Data2 = []  # Change variable name from AdvanTageous to AdvanTageous\_Data2

DisadvanTageous\_Data2 = []  # Change variable name from DisadvanTageous to DisadvanTageous\_Data2

i = 0

while i < group.columns.size:

    block = group.iloc[:, i:i + group.columns.size // 5]

    # Calculate the mean proportion of AdvanTageous and DisadvanTageous cards

    AdvanTageous\_Data2.append((block.isin([3, 4])).sum().sum() / block.size)

    DisadvanTageous\_Data2.append((block.isin([1, 2])).sum().sum() / block.size)

    i += group.columns.size // 5

# Calculate standard errors of the mean

AdvanTageous\_err\_Data2 = np.std(AdvanTageous\_Data2) / np.sqrt(len(AdvanTageous\_Data2))

DisadvanTageous\_err\_Data2 = np.std(DisadvanTageous\_Data2) / np.sqrt(len(DisadvanTageous\_Data2))

# Plotting

plt.errorbar(range(1, 6), AdvanTageous\_Data2, yerr=AdvanTageous\_err\_Data1, fmt='-o', color='blue', label='AdvanTageous')

plt.errorbar(range(1, 6), DisadvanTageous\_Data2, yerr=DisadvanTageous\_err\_Data2, fmt='-o', color='red', label='DisadvanTageous')

plt.title('Data2')

plt.xlabel('Block')

plt.ylabel('Mean Proportion')

plt.legend()

plt.tight\_layout()

plt.show()

# In[11]:

print("The the standard error of advantage for data 1 is {}".format(AdvanTageous\_err\_Data1))

print("The the standard error of disadvantage for data 1 is {}".format(DisadvanTageous\_err\_Data1))

# In[24]:

print("The the standard error of advantage for data 1 is {}".format(AdvanTageous\_err\_Data2))

print("The the standard error of disadvantage for data 1 is {}".format(DisadvanTageous\_err\_Data2))

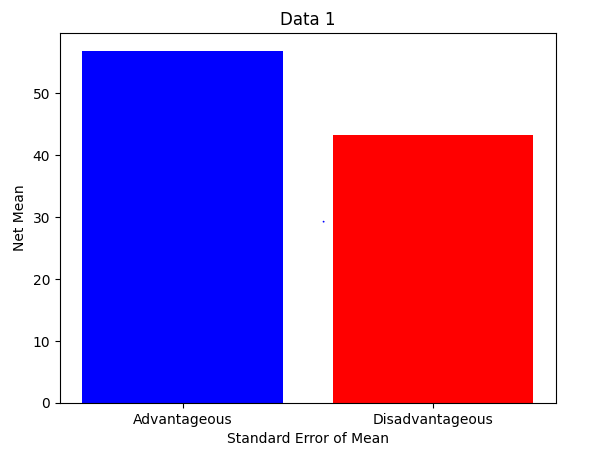
# In[ ]:

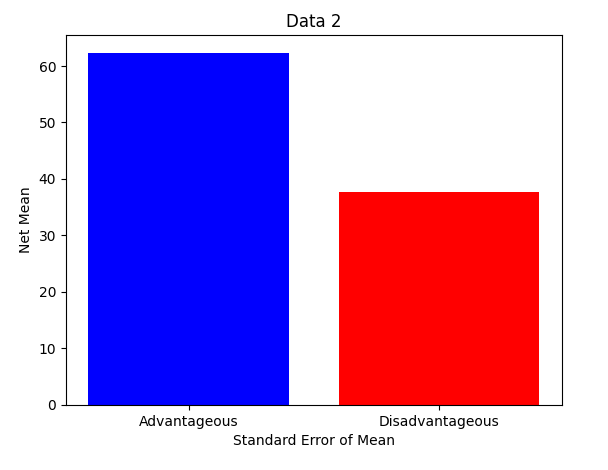
**B.** **Plot bar diagrams with error bars to show the net mean score/count/selections (across**

**participants; across advantageous vs disadvantageous cards respectively) and**

**standard error of the mean of advantageous cards versus disadvantageous cards**

**selected by participants of both groups.**





**Code is:**

#!/usr/bin/env python

# coding: utf-8

# # q2 B

# In[1]:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# In[2]:

data1 = pd.read\_excel('NDM\_Assignment3.xlsx', header=None)

data1 = data1.drop(0,axis =1)

data1

# In[3]:

data2 = pd.read\_excel('NDM\_Assignment3.xlsx', header=None,sheet\_name=1)

data2 = data2.drop(0,axis =1)

data2

# In[4]:

def calculateNetMean(data, decks):

    selected = data[data.isin(decks)].count(axis=1)

    net\_mean = selected.mean()

    return net\_mean

def calculate\_StandardError(data, decks):

    selected = data[data.isin(decks)].count(axis=1)

    stdErr = selected.sem()

    return stdErr

# In[5]:

# Calculate net mean and standard error for each group

advantageous\_decks = [3, 4]

disadvantageous\_decks = [1, 2]

netMeanData1Advantagous = calculateNetMean(data1, advantageous\_decks)

netMeanData2Advantageous = calculateNetMean(data2, advantageous\_decks)

netMeanData1Disadvantageous = calculateNetMean(data1, disadvantageous\_decks)

netMeanData2Disadvantageous = calculateNetMean(data2, disadvantageous\_decks)

# Calculate standard error of the mean for each group and deck type

semAdvantageousData1 = calculate\_StandardError(data1, advantageous\_decks)

semDisadvantageousData1 = calculate\_StandardError(data1, disadvantageous\_decks)

semAdvantageousData2 = calculate\_StandardError(data2, advantageous\_decks)

semDisadvantageousData2 = calculate\_StandardError(data2, disadvantageous\_decks)

# In[6]:

import matplotlib.pyplot as plt

# Assuming the following variables have been calculated

# sem\_advantageous\_data1, sem\_disadvantageous\_data1, net\_mean\_data1\_Advantagous, net\_mean\_data1\_Disadvantageous

labels = ['Advantageous', 'Disadvantageous']

x = [semAdvantageousData1, semDisadvantageousData1]

y = [netMeanData1Advantagous, netMeanData1Disadvantageous]

fig, axis = plt.subplots()

axis.bar(labels, y,color=['blue', 'red'])

axis.set\_ylabel('Net Mean')

axis.set\_xlabel('Standard Error of Mean')

# Set the title

axis.set\_title('Data 1')

# Display the plot

plt.show()

# In[7]:

import matplotlib.pyplot as plt

# Assuming the following variables have been calculated

# sem\_advantageous\_data1, sem\_disadvantageous\_data1, net\_mean\_data1\_Advantagous, net\_mean\_data1\_Disadvantageous

labels = ['Advantageous', 'Disadvantageous']

x = [semAdvantageousData2, semDisadvantageousData2]

y = [netMeanData2Advantageous, netMeanData2Disadvantageous]

fig, axis = plt.subplots()

axis.bar(labels, y,color=['blue', 'red'])

axis.set\_ylabel('Net Mean')

axis.set\_xlabel('Standard Error of Mean')

# Set the title

axis.set\_title('Data 2')

# Display the plot

plt.show()

# In[ ]:

**C.** **Carry out a statistical test to find if the means of the two different groups (calculated in**

**partB) from data1 and data2 differ from one another. Calculate the confidence interval**

**and effect size and report the results along with p-values, test statistics, and degrees of**

**freedom.**

For Advantageous Decks:

T-value: -4.957181970878901

P-value: 1.7123035593755243e-05

Confidence Interval: (50.96035037956482, 62.72386014675097)

Effect Size (Cohen's d): -1.6083222082395392

The degree of freedom :36

For Disadvantageous Decks:

T-value: 4.957181970878901

P-value: 1.7123035593755243e-05

Confidence Interval: (37.27613985324903, 49.03964962043518)

Effect Size (Cohen's d): 1.6083222082395392

The degree of freedom :36

**Interpret the results of the statistical test with regards to the decision making ability of**

**the participants in the two groups citing the key brain region(s) that might be linked to**

**the performance of both groups in the task.**

The t-test results for both the advantages and disadvantages show a significant difference between the two groups of participants.

For the Advantageous Decks:

* The negative t-value of this, which is -4.95  it tells that the mean score of the first group is less than the second group.
* Now, a small p-value of 1.71e-05 is less than the typical significance level (0.05), showing strong evidence against the null hypothesis.
* The confidence interval ranges from   50.96 to 62.72, which means that the 95% confident that the actual mean difference falls within this range.
* (Cohen’s d) is  -1.61 , indicating significant practical significance of the difference between the data1 and data2/

For the Disadvantageous Decks:

* t-value is 4.96  and is positive tells that the mean score of the data1 is greater than the second group.
* Now, a small p-value of 1.71e-05 is less than the typical significance level (0.05), showing strong evidence against the null hypothesis.
* The confidence interval ranges from   37.27 to 49.0, which means that the 95% confidence that the actual mean difference falls within this range.
* (Cohen’s d) is  -1.61 , indicating significant practical significance of the difference between the data1 and data2/

Here the key brain regions linked to decision-making ability are the prefrontal cortex and the amygdala, which are known to play crucial roles in decision-making processes.

Code is:

# # Q2c

# In[8]:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import scipy.stats as stats

# In[9]:

data1 = pd.read\_excel('NDM\_Assignment3.xlsx', header=None)

data1 = data1.drop(0,axis =1)

data1

# In[10]:

data2 = pd.read\_excel('NDM\_Assignment3.xlsx', header=None,sheet\_name=1)

data2 = data2.drop(0,axis =1)

data2

# In[15]:

n1 = len(data1)

n2 = len(data2)

# In[16]:

from scipy import stats

import numpy as np

t\_stat\_advanage, p\_val\_advadvantage = stats.ttest\_ind\_from\_stats(netMeanData1Advantagous, semAdvantageousData1, n1,

                                                   netMeanData2Advantageous, semAdvantageousData2, n2)

# Perform a t-test between the disadvantageous groups in data1 and data2

t\_stat\_disadvanatge, p\_val\_disadvantage = stats.ttest\_ind\_from\_stats(netMeanData1Disadvantageous, semDisadvantageousData1, n1,

                                                         netMeanData2Disadvantageous, semDisadvantageousData2, n2)

# Calculating the confidence interval for advantageous group

conf\_int\_advantage = stats.t.interval(0.95, df=n1-1, loc=netMeanData1Advantagous, scale=semAdvantageousData1)

# Calculating the confidence interval for disadvantageous group

conf\_int\_disadvantage = stats.t.interval(0.95, df=n1-1, loc=netMeanData1Disadvantageous, scale=semDisadvantageousData1)

# Calculating the   Cohen's d for advantageous groups

d\_advantage = (netMeanData1Advantagous - netMeanData2Advantageous) / np.sqrt((semAdvantageousData1\*\*2 + semAdvantageousData2\*\*2) / 2)

# Calculatinf the effect  Cohen's d for disadvantageous groups

d\_disadvantage = (netMeanData1Disadvantageous - netMeanData2Disadvantageous) / np.sqrt((semDisadvantageousData1\*\*2 + semDisadvantageousData2\*\*2) / 2)

# Calculating degree of freedom for advantage and disadvantage

degreeofFreedom\_advantage = n1 + n2 - 2

degreeofFreedom\_disadvantage = n1 + n2 - 2

# Print the results

print(" For Advantageous Decks:")

print("T-value: ", t\_stat\_advanage)

print("P-value: ", p\_val\_advadvantage)

print("Confidence Interval: ", conf\_int\_advantage)

print("Effect Size (Cohen's d): ", d\_advantage)

print("The degree of freedom :{}".format(degreeofFreedom\_advantage))

print("\n For  Disadvantageous Decks:")

print("T-value: ", t\_stat\_disadvanatge)

print("P-value: ", p\_val\_disadvantage)

print("Confidence Interval: ", conf\_int\_disadvantage)

print("Effect Size (Cohen's d): ", d\_disadvantage)

print("The degree of freedom :{}".format(degreeofFreedom\_disadvantage))

# In[ ]: