Assignment 3: Neuroscience of Decision Making PSY 307 (Monsoon 2024)

Name: Kunal Sharma

Roll Number: 2021331

---

2. Two independent groups of participants (63 each) performed an Iowa gambling task. 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. The file 'choice.xlsx' contains the deck chosen by the participant from four available options - deck 1, 2, 3, 4, 'win.xlsx' contains the gain associated with the deck chosen and 'loss.xlsx' contains the loss associated with the deck chosen. Each excel file contains two sheets representing the two groups. Each sheet contains 63 rows and 100 columns, each row representing one participant's data and each column represents one trial.

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).

A) Calculate the proportion of switches made after a loss and a gain trial for each participant. Switch refers to a change in the choice of the deck in the subsequent trial. A loss trial is where the money received is less than the amount lost, and a gain trial is where the money received is greater than the amount lost. Create a larger plot with two subplots—one subplot representing each group. Plot bar diagrams representing the mean proportion of switched responses for the gain and loss trial and the standard error of the mean for each group. [6+4 marks]
Conduct appropriate statistical tests to compare the

i) proportion of switched responses of gain/ loss trials between groups. Briefly explain the findings of the statistical analysis carried out.

ii) proportion of switched responses of gain trial and loss trials within each group. Briefly explain the findings of the statistical analysis carried out.

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

Answer:

```
import matplotlib.pyplot as plt
import pandas as pd
from scipy.stats import ttest_ind, ttest_rel, mannwhitneyu, wilcoxon
from scipy.stats import sem
import numpy as np
```

```python
from scipy.stats import shapiro

# Helper function to calculate proportion of switches after gain and loss trials
def helper(choices, wins, losses):
   n_participants, n_trials = choices.shape
   gain_switch_rates = np.zeros(n_participants)
   loss_switch_rates = np.zeros(n_participants)

   p_idx = 0
   while p_idx < n_participants:
       l_trials = 0
       g_switches = 0
       g_trials = 0
       l_switches = 0
       t_idx = 0
       while t_idx < n_trials - 1:
           if (wins.iloc[p_idx, t_idx]) + (losses.iloc[p_idx, t_idx]) > 0:
               g_trials += 1
               if (choices.iloc[p_idx, t_idx]) != (choices.iloc[p_idx, t_idx + 1]):
                   g_switches += 1

           elif (wins.iloc[p_idx, t_idx]) + (losses.iloc[p_idx, t_idx]) < 0:
               l_trials += 1
               if (choices.iloc[p_idx, t_idx]) != (choices.iloc[p_idx, t_idx + 1]):
                   l_switches += 1
           t_idx += 1

       if g_trials > 0:
           gain_switch_rates[p_idx] = g_switches / g_trials
       else:
           gain_switch_rates[p_idx] = 0


       if l_trials > 0:
           loss_switch_rates[p_idx] = l_switches / l_trials
       else:
           loss_switch_rates[p_idx] = 0


       p_idx += 1

   return gain_switch_rates, loss_switch_rates
```

```python
# Process data for both groups
grp1_choices = pd.ExcelFile('choice.xlsx').parse('group1')
grp2_choices = pd.ExcelFile('choice.xlsx').parse('group2')
grp1_wins = pd.ExcelFile('win.xlsx').parse('group1')
grp2_wins = pd.ExcelFile('win.xlsx').parse('group2')
grp1_losses = pd.ExcelFile('loss.xlsx').parse('group1')
grp2_losses = pd.ExcelFile('loss.xlsx').parse('group2')

grp1_gain_rates, grp1_loss_rates = helper(grp1_choices, grp1_wins, grp1_losses)
grp2_gain_rates, grp2_loss_rates = helper(grp2_choices, grp2_wins, grp2_losses)

# Calculate means and standard errors for plotting
mean_grp1 = [np.mean(grp1_gain_rates), np.mean(grp1_loss_rates)]
err_grp1 = [sem(grp1_gain_rates), sem(grp1_loss_rates)]
mean_grp2 = [np.mean(grp2_gain_rates), np.mean(grp2_loss_rates)]
err_grp2 = [sem(grp2_gain_rates), sem(grp2_loss_rates)]

# Plotting
fig, axes = plt.subplots(1, 2, figsize=(12, 6), sharey=True)
x_labels = ['Gain Trials', 'Loss Trials']

# Group 1
axes[0].bar(x_labels, mean_grp1, yerr=err_grp1, capsize=5, alpha=0.7)
axes[0].set_title('Group 1')
axes[0].set_ylabel('Mean Proportion of Switches')
axes[0].set_ylim(0, 1)

# Group 2
axes[1].bar(x_labels, mean_grp2, yerr=err_grp2, capsize=5, alpha=0.7)
axes[1].set_title('Group 2')

plt.tight_layout()
plt.show()

# Printing results for Shapiro-Wilk Normality Test
print("\nShapiro-Wilk Normality Test Results:")
print("-" * 35)
print(f"Group 1 Gain Trials: Statistic={shapiro(grp1_gain_rates).statistic},
P-value={shapiro(grp1_gain_rates).pvalue}")

print(f"Group 1 Loss Trials: Statistic={shapiro(grp1_loss_rates).statistic},
P-value={shapiro(grp1_loss_rates).pvalue}")
```

```python
print(f"Group 2 Gain Trials: Statistic={shapiro(grp2_gain_rates).statistic},
P-value={shapiro(grp2_gain_rates).pvalue}")

print(f"Group 2 Loss Trials: Statistic={shapiro(grp2_loss_rates).statistic},
P-value={shapiro(grp2_loss_rates).pvalue}")

# i) Between groups comparison
if shapiro(grp2_gain_rates).pvalue < 0.05:
   gain_test = mannwhitneyu(grp1_gain_rates, grp2_gain_rates)
else:
   gain_test = ttest_ind(grp1_gain_rates, grp2_gain_rates, equal_var=False)


if shapiro(grp2_loss_rates).pvalue < 0.05:
   loss_test = mannwhitneyu(grp1_loss_rates, grp2_loss_rates)
else:
   loss_test = ttest_ind(grp1_loss_rates, grp2_loss_rates, equal_var=False)

# ii) Within groups comparison (gain vs loss within each group)
if shapiro(grp1_gain_rates).pvalue > 0.05 and shapiro(grp1_loss_rates).pvalue > 0.05:
   grp1_gain_loss_test = ttest_rel(grp1_gain_rates, grp1_loss_rates)
else:
   grp1_gain_loss_test = wilcoxon(grp1_gain_rates, grp1_loss_rates)


if shapiro(grp2_gain_rates).pvalue < 0.05 or shapiro(grp2_loss_rates).pvalue < 0.05:
   grp2_gain_loss_test = wilcoxon(grp2_gain_rates, grp2_loss_rates)
else:
   grp2_gain_loss_test = ttest_rel(grp2_gain_rates, grp2_loss_rates)

# Printing results for Mann-Whitney U Statistical Test
print("\nMann-Whitney U Statistical Test Results:")
print("-" * 39)
print(f"Gain Trials Between Groups: Statistic={gain_test.statistic}, P-value={gain_test.pvalue}")
print(f"Loss Trials Between Groups: Statistic={loss_test.statistic}, P-value={loss_test.pvalue}")

# Printing results for Wilcoxon Signed-Rank Statistical Test
print("\nWilcoxon Signed-Rank Statistical Test Results:")
print("-" * 45)
print(f"Within Group 1 (Gain vs Loss): Statistic={grp1_gain_loss_test.statistic},
P-value={grp1_gain_loss_test.pvalue}")
print(f"Within Group 2 (Gain vs Loss): Statistic={grp2_gain_loss_test.statistic},
P-value={grp2_gain_loss_test.pvalue}\n")
```

Graph: Plot to represent the mean proportion of switched responses for the gain and loss trial and the standard error of the mean for each group.

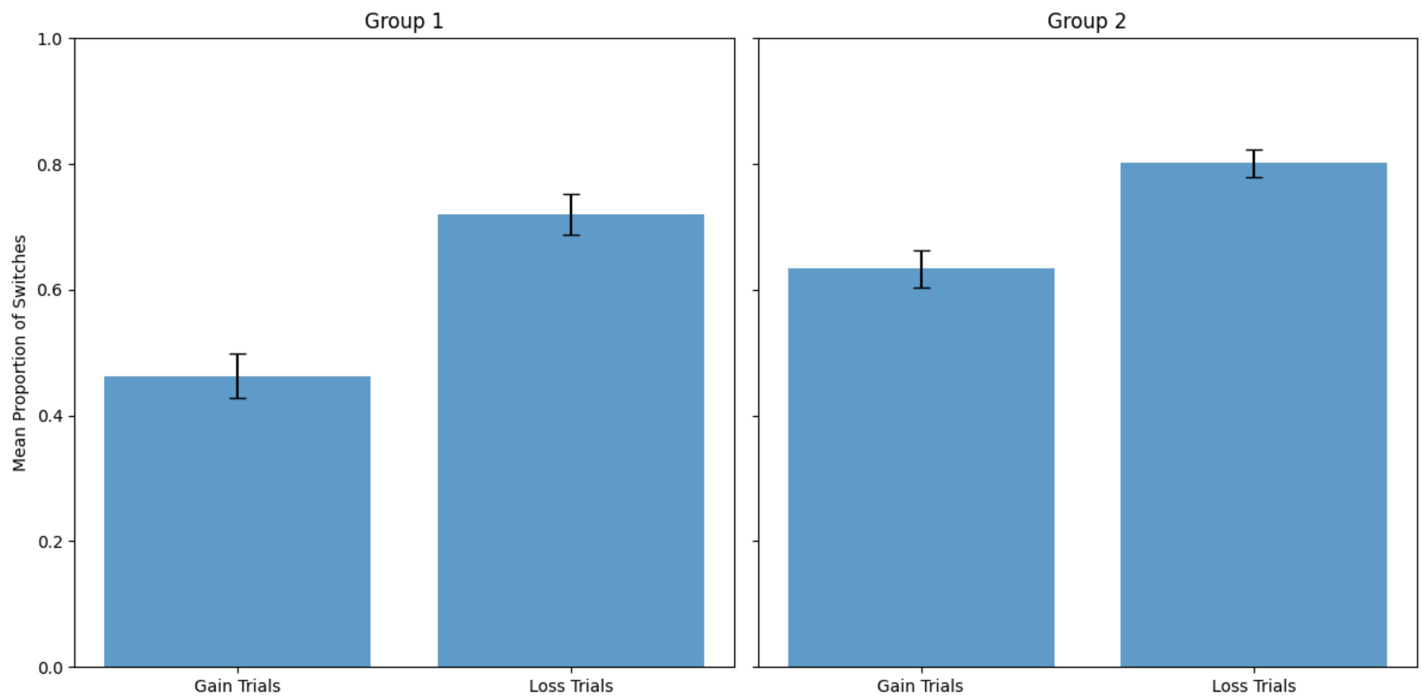

Figure 1.1 The mean percentage of switches in deck choice following gain and loss trials for two separate participant groups completing the Iowa Gambling Task is displayed in the bar graphs. Group 1 data is shown in the left subplot, whereas Group 2 data is shown in the right subplot. Participants in each group showed a greater percentage of switches after losing trials as opposed to winning trials. The standard error of the mean (SEM) for each condition is displayed by error bars, which show participant variation in switching behaviour.

Output on the console:

Shapiro-Wilk Normality Test Results:

Group 1 Gain Trials: Statistic=0.9608442783355713, P-value=0.04569971561431885

Group 1 Loss Trials: Statistic=0.9027128219604492, P-value=0.00012933777179569006

Group 2 Gain Trials: Statistic=0.9384575486183167, P-value=0.003854369977489114

Group 2 Loss Trials: Statistic=0.8878676891326904, P-value=3.7444133340613917e-05

Observation:

From the output we can observe that the data is not normally distributed (the null hypothesis is rejected) because the p-values for both groups gain and loss trials are less than 0.05.

(i)

For this part, I have used the nonparametric Mann-Whitney U Test, which is the counterpart of the independent t-test, as the two classes under comparison are independent and not normally distributed.

Output on the console:

Mann-Whitney U Statistical Test Results:

Gain Trials Between Groups: Statistic=1185.5, P-value=0.0002348745501814016

Loss Trials Between Groups: Statistic=1649.5, P-value=0.17311514947192652

Observation:

The findings indicate that while there is a significant difference ($p<0.05$) between group 1 and group 2's gain trials, there is not a significant difference ($p>0.05$) between their loss trials. This indicates that, in comparison to group 1 participants, group 2 individuals exchanged the decks more after receiving prizes. This indicates that, in contrast to group 2, group 1 was more often biased toward the decks that generated profits for them in the earlier trial. However, when changing decks after suffering losses, both groups behave similarly.

(ii)

For this part, I have used the nonparametric Wilcoxon Signed-Rank Test, which is the counterpart of the dependent t-test, as the two classes under comparison are dependent and not normally distributed.

Output on the console:

Wilcoxon Signed-Rank Statistical Test Results:

Within Group 1 (Gain vs Loss): Statistic=54.0, P-value=1.5188953034621485e-10

Within Group 2 (Gain vs Loss): Statistic=139.0, P-value=6.917007230063962e-09

Observation:

The findings indicate that there is a significant difference ($p<0.05$) between each group's gain and loss trials. This essentially indicates that after experiencing a loss, players in both groups often shuffle the decks more frequently than after experiencing a gain. This demonstrates the participant's mental bias toward selecting the deck that resulted in a profit in the prior trial as well as their bias against selecting the deck that resulted in a loss.

B) For each group, determine the deck chosen by each participant immediately before switching decks after encountering a loss trial. Subsequently, calculate the proportion of each deck chosen relative to the total number of loss trials for each participant. Create a larger plot with two subplots—one subplot representing each group. Plot the mean proportion as a bar diagram and the standard error of the mean for each of the four deck choices during loss trials. Rank the decks in decreasing order based on their mean proportions for each group. [4+ 1 marks]

Answer:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import sem

def helper(choice_df, loss_df):
   num_users, num_steps = choice_df.shape
   deck_data = {d: [] for d in range(1, 5)}  # Deck proportions
   i = 0
   while i < num_users:
       deck_count = {d: 0 for d in range(1, 5)}
       loss_count = 0
       j = 0
       while j < num_steps - 1:
           # Loss trial and switch occurred
           if (loss_df.iloc[i, j]) < 0 and (choice_df.iloc[i, j]) != choice_df.iloc[i, j + 1]:
               loss_count += 1
               deck_count[choice_df.iloc[i, j]] += 1
           j += 1
       # Calculating proportions for this user
       d = 1
       while d <= 4:
           if loss_count > 0:
               prop = deck_count[d] / loss_count
           else:
               prop = 0
           deck_data[d].append(prop)
           d += 1
       i += 1
   # Computing means
   mean_data = {d: np.mean(deck_data[d]) for d in deck_data}
   #computing SEMs
   sem_data = {d: sem(deck_data[d]) for d in deck_data}
   return mean_data, sem_data
# Process data for both group1 and group2
group1_choice = pd.ExcelFile('choice.xlsx').parse('group1')
```

```
group2_choice = pd.ExcelFile('choice.xlsx').parse('group2')
group1_loss = pd.ExcelFile('loss.xlsx').parse('group1')
group2_loss = pd.ExcelFile('loss.xlsx').parse('group2')

# Calculating proportions for group 1
group1_mean, group1_sem = helper(group1_choice, group1_loss)
# Calculating proportions for group 1
group2_mean, group2_sem = helper(group2_choice, group2_loss)

# Preparing data for plotting
group1_vals = [group1_mean[d] for d in [1, 2, 3, 4]]
group1_errs = [group1_sem[d] for d in [1, 2, 3, 4]]
group2_vals = [group2_mean[d] for d in [1, 2, 3, 4]]
group2_errs = [group2_sem[d] for d in [1, 2, 3, 4]]

# Plotting
fig, axes = plt.subplots(1, 2, figsize=(12, 6), sharey=True)
x_labels = ['Deck 1', 'Deck 2', 'Deck 3', 'Deck 4']

# plot for Group 1
axes[0].bar(x_labels, group1_vals, yerr=group1_errs, capsize=5, alpha=0.7)
axes[0].set_title('Group 1')
axes[0].set_ylabel('Mean Proportion')

# plot for Group 2
axes[1].bar(x_labels, group2_vals, yerr=group2_errs, capsize=5, alpha=0.7)
axes[1].set_title('Group 2')
plt.tight_layout()
plt.show()

# Ranking decks by mean proportions for group 1
group1_rank = sorted(group1_mean.items(), key=lambda x: x[1], reverse=True)
# Ranking decks by mean proportions for group 2
group2_rank = sorted(group2_mean.items(), key=lambda x: x[1], reverse=True)

# Print rankings for group 1
print("\nGroup 1 Rankings:")
for rank, (deck, mean) in enumerate(group1_rank, start=1):
   print(f"Rank {rank}: Deck {deck} with Mean Proportion {mean}")

# Print rankings for group 2
print("\nGroup 2 Rankings:")
for rank, (deck, mean) in enumerate(group2_rank, start=1):
   print(f"Rank {rank}: Deck {deck} with Mean Proportion {mean}")
```

Graph: Plot for the mean proportion and the standard error of the mean for each of the four deck choices during loss trials.

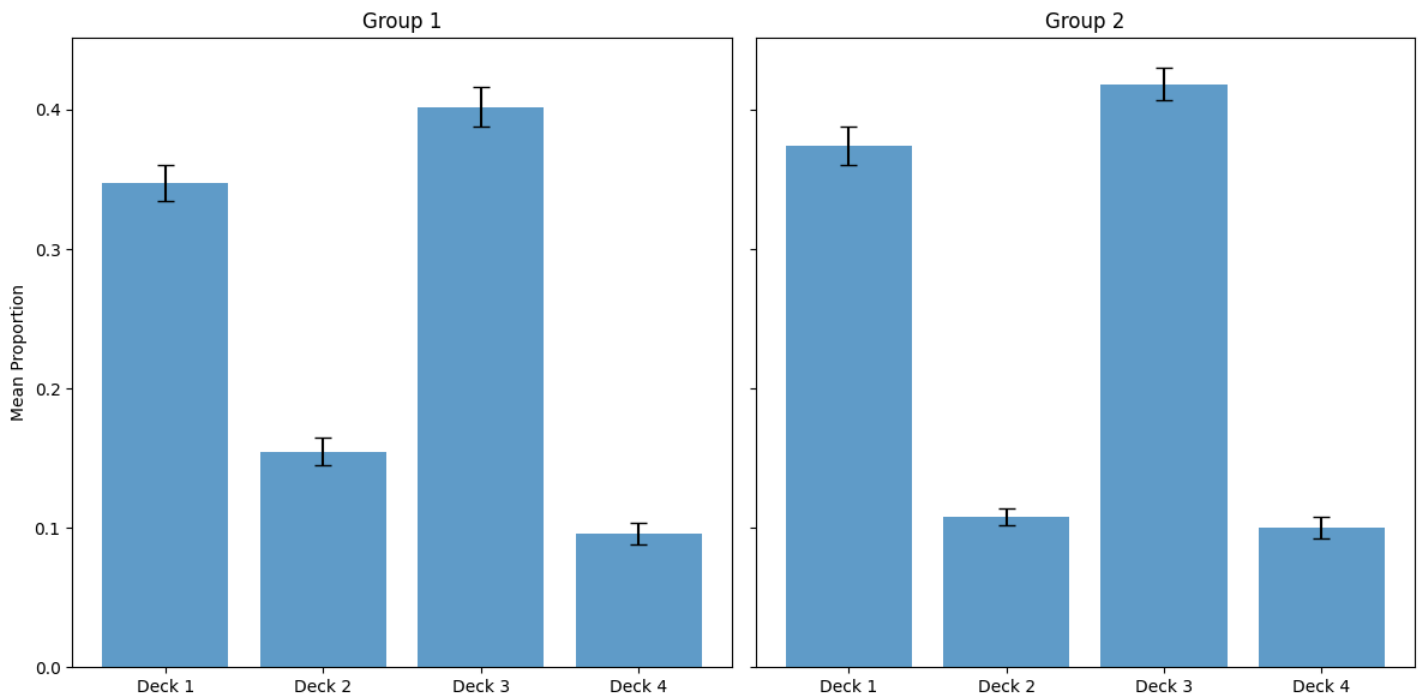

Figure 1.2 The mean percentage of each deck (Decks 1, 2, 3, and 4) selected right before swapping decks following a loss trial in the Iowa Gambling Task is shown in the bar graphs. Group 1 is represented by the left subplot, whereas Group 2 is represented by the right subplot. The standard error of the mean (SEM), which shows inter-participant variation in deck preferences during loss trials, is represented by error bars.

Output on the console:

Group 1 Rankings:

Rank 1: Deck 3 with Mean Proportion 0.40181232392805327

Rank 2: Deck 1 with Mean Proportion 0.34763136509987924

Rank 3: Deck 2 with Mean Proportion 0.15483243709121924

Rank 4: Deck 4 with Mean Proportion 0.09572387388084827

Group 2 Rankings:

Rank 1: Deck 3 with Mean Proportion 0.41814019344333664

Rank 2: Deck 1 with Mean Proportion 0.3740587316359752

Rank 3: Deck 2 with Mean Proportion 0.10767297358577349

Rank 4: Deck 4 with Mean Proportion 0.1001281013349146

C) For each group, determine the deck switched to by each participant immediately after encountering a loss trial. Subsequently, calculate the proportion of each deck chosen relative to the total number of loss trials for each participant. Create a larger plot with two subplots—one subplot representing each group. Plot the mean proportion as a bar diagram and the standard error of the mean for each of the four deck choices during loss trials. Rank the decks in decreasing order based on their mean proportions for each group. [4+1 marks]

Answer:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import sem

def helper(choice_df, loss_df):
   num_users, num_steps = choice_df.shape
   deck_data = {d: [] for d in range(1, 5)}  # Deck proportions
   i = 0
   while i < num_users:
       deck_count = {d: 0 for d in range(1, 5)}
       loss_count = 0
       j = 0
       while j < num_steps - 1:
           # Loss trial and switch occurred
           if (loss_df.iloc[i, j]) < 0 and (choice_df.iloc[i, j]) != choice_df.iloc[i, j + 1]:
               loss_count += 1
               deck_count[choice_df.iloc[i, j+1]] += 1
           j += 1
       # Calculating proportions for this user
       d = 1
       while d <= 4:
           if loss_count > 0:
               prop = deck_count[d] / loss_count
           else:
               prop = 0
           deck_data[d].append(prop)
           d += 1
       i += 1
   # Computing means
   mean_data = {d: np.mean(deck_data[d]) for d in deck_data}
   #computing SEMs
   sem_data = {d: sem(deck_data[d]) for d in deck_data}
   return mean_data, sem_data
# Process data for both group1 and group2
group1_choice = pd.ExcelFile('choice.xlsx').parse('group1')
```

```python
group2_choice = pd.ExcelFile('choice.xlsx').parse('group2')
group1_loss = pd.ExcelFile('loss.xlsx').parse('group1')
group2_loss = pd.ExcelFile('loss.xlsx').parse('group2')

# Calculating proportions for group 1
group1_mean, group1_sem = helper(group1_choice, group1_loss)
# Calculating proportions for group 1
group2_mean, group2_sem = helper(group2_choice, group2_loss)

# Preparing data for plotting
group1_vals = [group1_mean[d] for d in [1, 2, 3, 4]]
group1_errs = [group1_sem[d] for d in [1, 2, 3, 4]]
group2_vals = [group2_mean[d] for d in [1, 2, 3, 4]]
group2_errs = [group2_sem[d] for d in [1, 2, 3, 4]]

# Plotting
fig, axes = plt.subplots(1, 2, figsize=(12, 6), sharey=True)
x_labels = ['Deck 1', 'Deck 2', 'Deck 3', 'Deck 4']

# plot for Group 1
axes[0].bar(x_labels, group1_vals, yerr=group1_errs, capsize=5, alpha=0.7)
axes[0].set_title('Group 1')
axes[0].set_ylabel('Mean Proportion')

# plot for Group 2
axes[1].bar(x_labels, group2_vals, yerr=group2_errs, capsize=5, alpha=0.7)
axes[1].set_title('Group 2')
plt.tight_layout()
plt.show()

# Ranking decks by mean proportions for group 1
group1_rank = sorted(group1_mean.items(), key=lambda x: x[1], reverse=True)
# Ranking decks by mean proportions for group 2
group2_rank = sorted(group2_mean.items(), key=lambda x: x[1], reverse=True)

# Print rankings for group 1
print("\nGroup 1 Rankings:")
for rank, (deck, mean) in enumerate(group1_rank, start=1):
   print(f"Rank {rank}: Deck {deck} with Mean Proportion {mean}")

# Print rankings for group 2
print("\nGroup 2 Rankings:")
for rank, (deck, mean) in enumerate(group2_rank, start=1):
   print(f"Rank {rank}: Deck {deck} with Mean Proportion {mean}")
```

Graph: Plot for the mean proportion and the standard error of the mean for each of the four deck choices during loss trials.

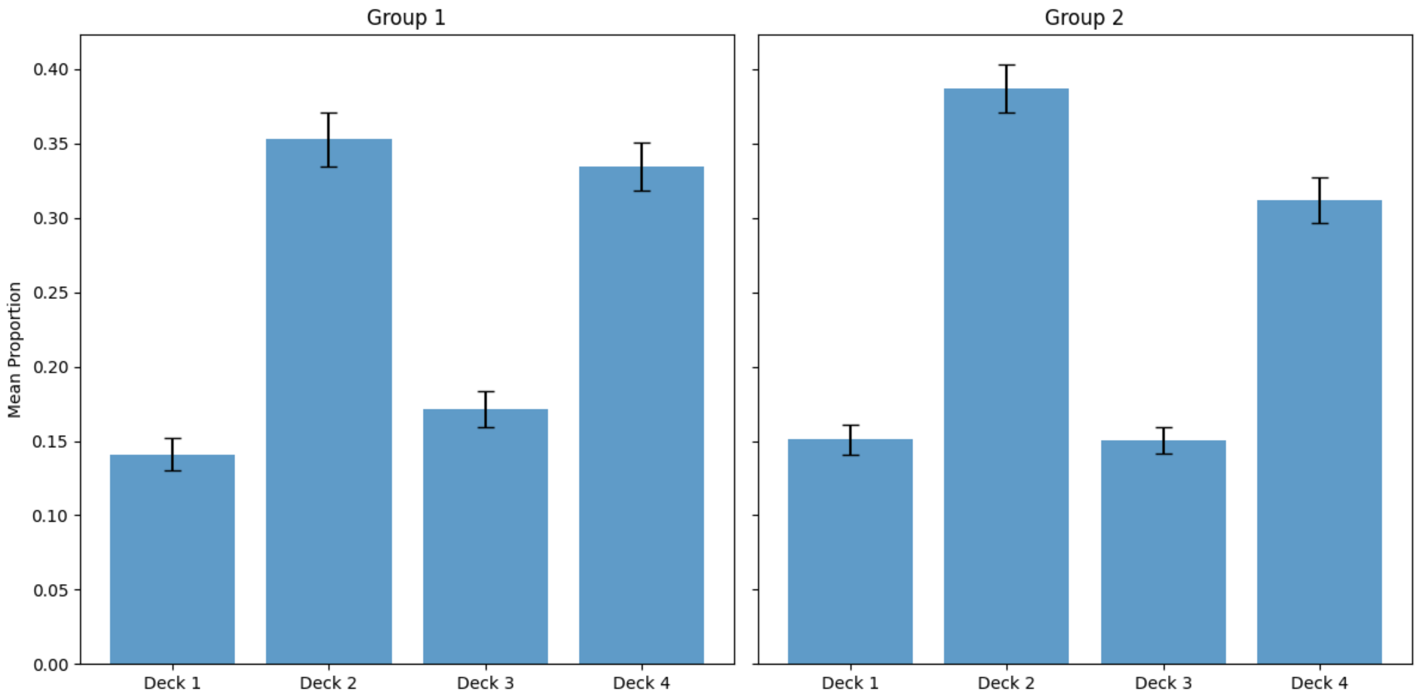

Figure 1.3 The mean percentage of each deck (Decks 1, 2, 3, and 4) selected right after switching decks following a loss trial in the Iowa Gambling Task is shown in the bar graphs. Group 1 is represented by the left subplot, whereas Group 2 is represented by the right subplot. The standard error of the mean (SEM), which shows inter-participant variation in deck preferences after loss trials, is represented by error bars.

Output on the console:

Group 1 Rankings:
Rank 1: Deck 2 with Mean Proportion 0.35286665385266264
Rank 2: Deck 4 with Mean Proportion 0.3345995398123223
Rank 3: Deck 3 with Mean Proportion 0.17141012280377985
Rank 4: Deck 1 with Mean Proportion 0.1411236835312352

Group 2 Rankings:
Rank 1: Deck 2 with Mean Proportion 0.3869435738387346
Rank 2: Deck 4 with Mean Proportion 0.3116253319427489
Rank 3: Deck 1 with Mean Proportion 0.1508303435836098
Rank 4: Deck 3 with Mean Proportion 0.1506007506349066