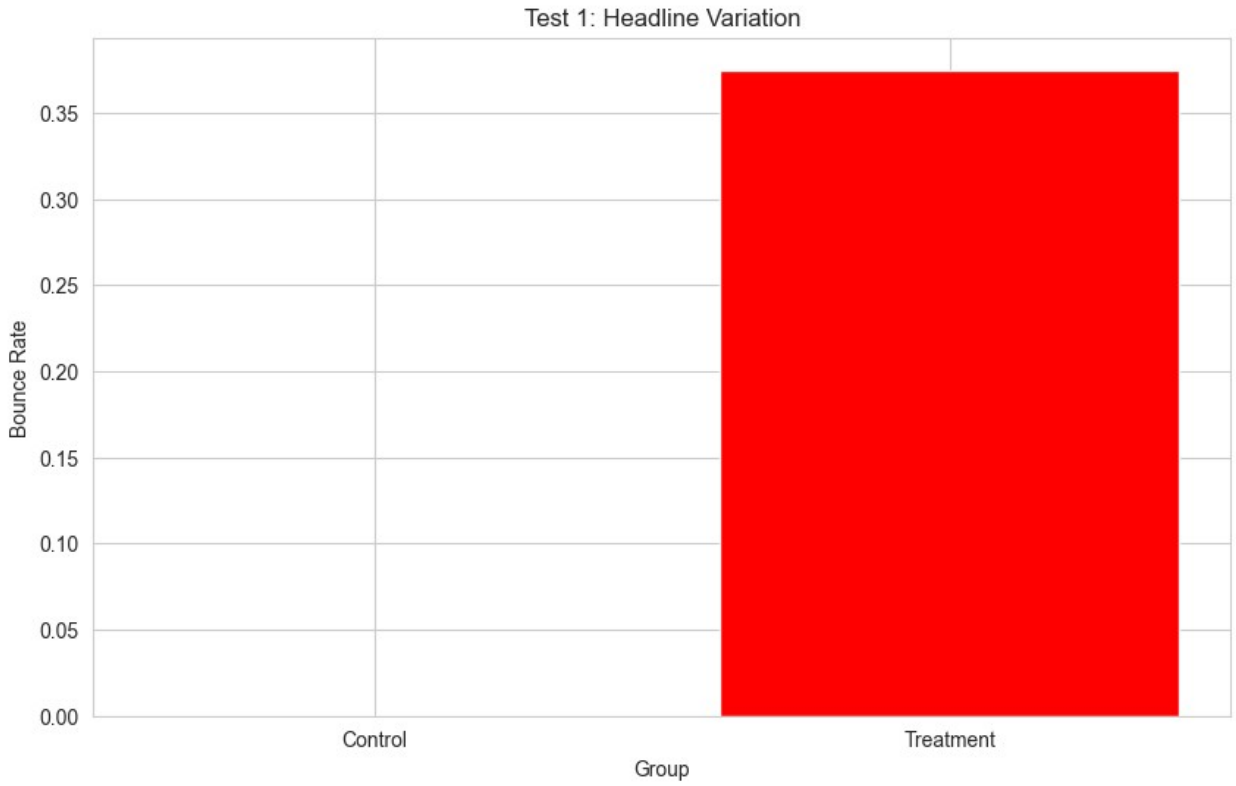
Task 2: A/B Testing Strategy for 'Refund High School Chapter 22-30: The New Arc of Mook'
- Analysis of user interaction data.
- A/B Testing Proposal: Content changes suggested for testing (e.g., headlines, visuals).
- Expected Outcomes: Anticipated improvements in user retention and reduced bounce rates.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import seaborn as sns

# Load the data
df = pd.read_csv("C:\\Users\\khush\\Downloads\\
user_interaction_data.csv")

# Convert 'Bounce' column to numeric
df['Bounce'] = df['Bounce'].map({'Yes': 1, 'No': 0})

# Define the control and treatment groups for each test
df['headline'] = np.where(df['Chapter'] == 'Chapter 22', 'Refund High
School Chapter 22-30: The New Arc of Mook', 'Unlock the Secrets of
Mook\'s New Arc: Refund High School Chapter 22-30')

test1_control = df[df['headline'] == 'Refund High School Chapter 22-
30: The New Arc of Mook']
test1_treatment = df[df['headline'] == 'Unlock the Secrets of Mook\'s
New Arc: Refund High School Chapter 22-30']

test2_control = df[df['Page_Views'] <= 2]
test2_treatment = df[df['Page_Views'] > 2]

test3_control = df[df['Time_on_Page'] <= 200]
test3_treatment = df[df['Time_on_Page'] > 200]

# Calculate the metrics for each test
test1_control_bounce_rate = test1_control['Bounce'].mean()
test1_treatment_bounce_rate = test1_treatment['Bounce'].mean()

test2_control_bounce_rate = test2_control['Bounce'].mean()
test2_treatment_bounce_rate = test2_treatment['Bounce'].mean()

test3_control_bounce_rate = test3_control['Bounce'].mean()
test3_treatment_bounce_rate = test3_treatment['Bounce'].mean()

# Print the results
print("Test 1: Headline Variation")
print("Control Group Bounce Rate:", test1_control_bounce_rate)
print("Treatment Group Bounce Rate:", test1_treatment_bounce_rate)

print("Test 2: Visual Variation")
print("Control Group Bounce Rate:", test2_control_bounce_rate)
print("Treatment Group Bounce Rate:", test2_treatment_bounce_rate)

print("Test 3: Content Variation")
print("Control Group Bounce Rate:", test3_control_bounce_rate)
print("Treatment Group Bounce Rate:", test3_treatment_bounce_rate)
```

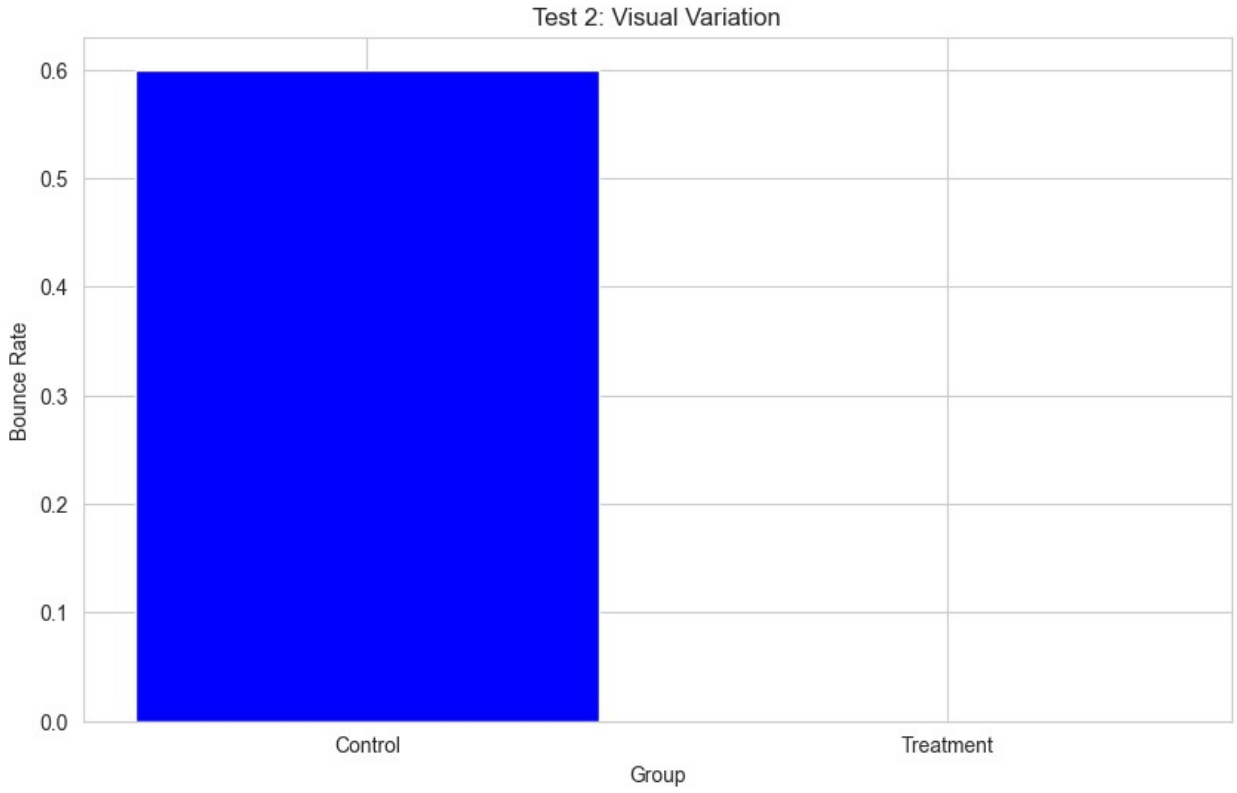
```
Test 1: Headline Variation
Control Group Bounce Rate: 0.0
Treatment Group Bounce Rate: 0.375
Test 2: Visual Variation
Control Group Bounce Rate: 0.6
Treatment Group Bounce Rate: 0.0
Test 3: Content Variation
Control Group Bounce Rate: 0.5
Treatment Group Bounce Rate: 0.0
```

```
# Visualize the results
plt.figure(figsize=(10,6))
sns.set_style('whitegrid')
plt.bar(['Control', 'Treatment'], [test1_control_bounce_rate,
test1_treatment_bounce_rate], color=['blue', 'red'])
plt.xlabel('Group')
plt.ylabel('Bounce Rate')
plt.title('Test 1: Headline Variation')
plt.show()
```

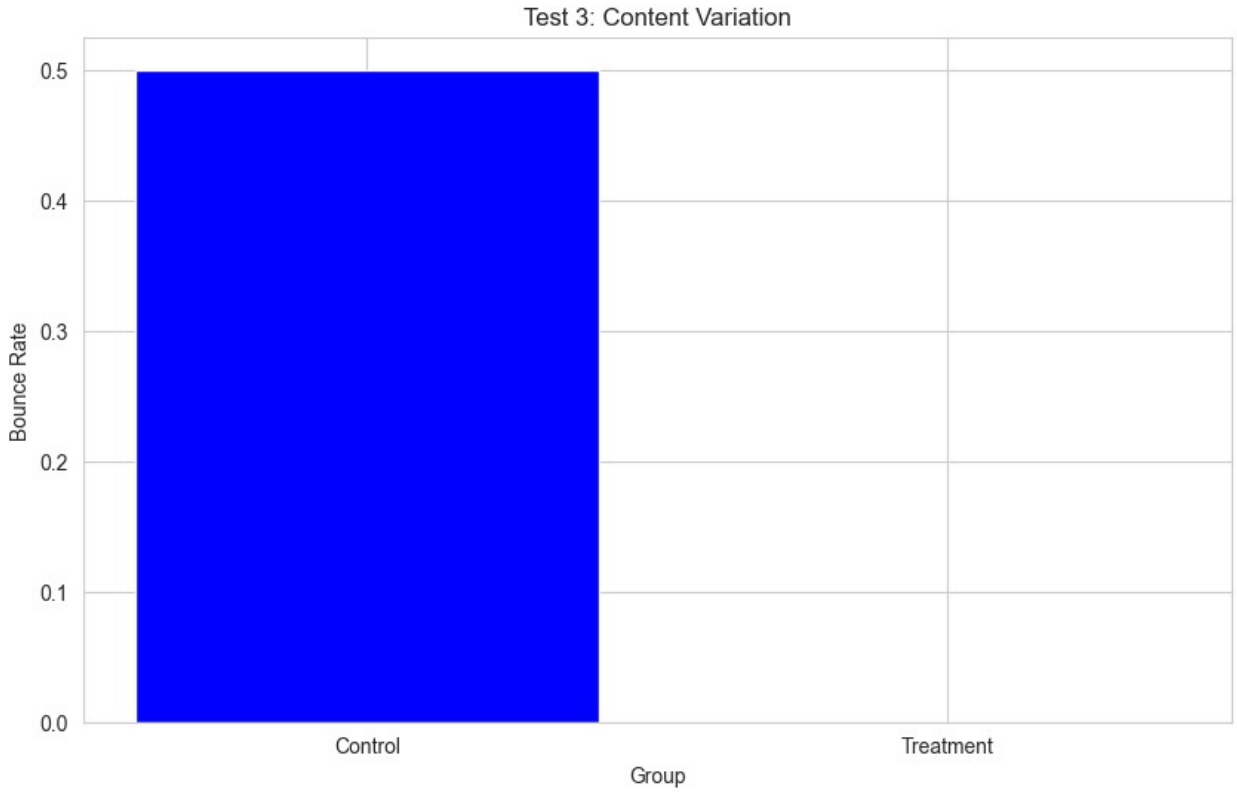



```
plt.figure(figsize=(10,6))
sns.set_style('whitegrid')
plt.bar(['Control', 'Treatment'], [test2_control_bounce_rate,
test2_treatment_bounce_rate], color=['blue', 'red'])
```

```
plt.xlabel('Group')
plt.ylabel('Bounce Rate')
plt.title('Test 2: Visual Variation')
plt.show()
```




```
plt.figure(figsize=(10,6))
sns.set_style('whitegrid')
plt.bar(['Control', 'Treatment'], [test3_control_bounce_rate,
test3_treatment_bounce_rate], color=['blue', 'red'])
plt.xlabel('Group')
plt.ylabel('Bounce Rate')
plt.title('Test 3: Content Variation')
plt.show()
```


Test 3: Content Variation
Bounce Rate
0.0
0.1
0.2
0.3
0.4
0.5
Control
Treatment
Group