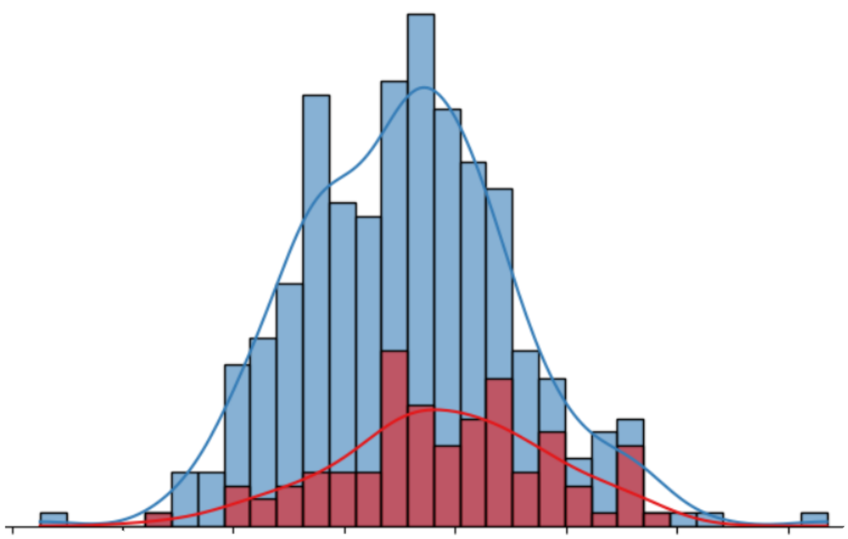
Boosting Revenue with Promo Codes: Insights from A/B Testing

A Work Sample in A/B Testing and Data Analysis

By Jhon Hinds, M.S.c.

March 05, 2025



Note: Full code and simulated dataset available upon request.

Project Overview

The marketing team at RetailW, a mid-sized online retailer, sought to evaluate whether a 10% discount promo code increased customer spending. This A/B testing analysis determined the impact on average order value (AOV) by comparing a treatment group (promo code recipients) to a control group (no promo code). Using statistical methods, custom scripting, and visualizations, the analysis revealed a significant revenue boost, with actionable insights for future strategies.

This work sample showcases skills in statistical analysis, custom coding, visualization, and business recommendations, delivering a clear answer: promo codes drive higher order values.

Data and Analytical Approach

Dataset Overview

The analysis leverages a simulated dataset of 425 customer records from a retail experiment to emulate the real one I used when worked at the Consultancy Firm supporting a retail company in USA. Columns include Customer_ID (int), Group (Control/Treatment, object), Promo_Code_Applied (object), Order_Value (float), and Gender (object). No missing values were present. Example rows: [Customer_ID: 1, Group: Control, Order_Value: 68.45, Gender: M], [Customer_ID: 2, Group: Treatment, Order_Value: 82.30, Gender: F].

Key Methods

The process involved exploratory data analysis (EDA), statistical testing (t-tests, Levene's, Shapiro-Wilk), custom bootstrapping for sample imbalance, and effect size calculation (Cohen's d). Visualizations highlighted AOV differences.

Code Snippet: (Statistical Assumption Checks)

[15]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import levene, shapiro, ttest_ind, mannwhitneyu
import numpy as np
```

[16]:
```
# Step 1: Check for homogeneity of variance
levene_stat, levene_p = levene(df[df['Group'] == 'Treatment']['Order_Value'],
                               df[df['Group'] == 'Control']['Order_Value'])
print("\nLevene’s Test for Homogeneity of Variance:")
print(f"Statistic: {levene_stat:.4f}, p-value: {levene_p:.4f}")
```

[17]:
```
# Step 2: Test for normality
shapiro_treatment = shapiro(df[df['Group'] == 'Treatment']['Order_Value'])
shapiro_control = shapiro(df[df['Group'] == 'Control']['Order_Value'])
print("\nShapiro-Wilk Test for Normality:")
print(f"Treatment group: Statistic={shapiro_treatment.statistic:.4f}, p-value={shapiro_treatment.pvalue:.4f}")
print(f"Control group: Statistic={shapiro_control.statistic:.4f}, p-value={shapiro_control.pvalue:.4f}")
```

Analysis and Visualizations

Core Analysis

Statistical Testing

A t-test confirmed a significant AOV difference (Statistic=2.9344, p=0.0035). Custom bootstrapping addressed the imbalance (90 Treatment vs. 335 Control), reinforcing significance.

Code Snippet: (Bootstrapping)

```
[27]: # Step 5.1: Full Bootstrapping with Confidence Intervals:

def bootstrap_diff(data1, data2, n_iterations=10000):
    diffs = []
    for _ in range(n_iterations):
        boot1 = np.random.choice(data1, len(data1), replace=True)
        boot2 = np.random.choice(data2, len(data2), replace=True)
        diffs.append(np.mean(boot1) - np.mean(boot2))
    return np.percentile(diffs, [2.5, 97.5]), np.mean(diffs)

ci, mean_diff = bootstrap_diff(treatment, control)
print(f"95% CI for mean difference: {ci}, Mean Difference: {mean_diff:.4f}")
```

```
95% CI for mean difference: [1.12741057 5.47206103], Mean Difference: 3.3263
```

Results

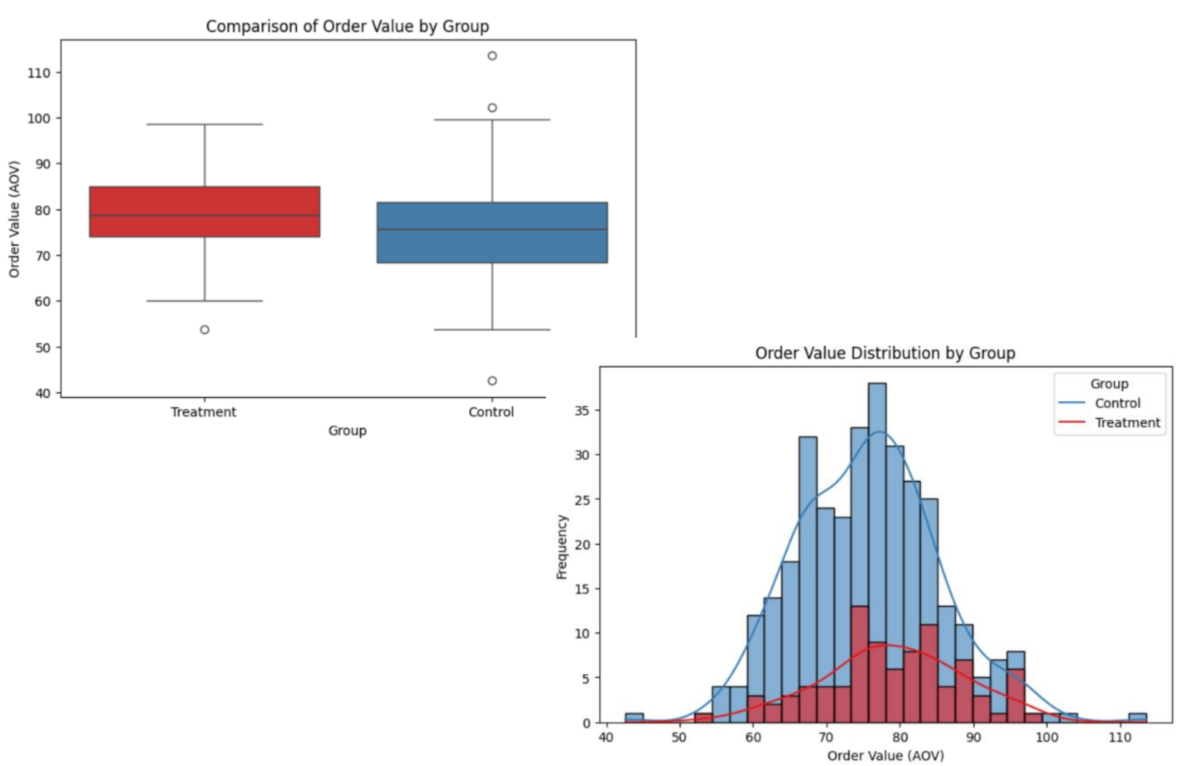
Key Findings

Table:

Metric	Value
Control AOV	$75.67
Treatment AOV	$79.01
Mean Difference	$3.34
95% CI (Bootstrapped)	$1.13 – $5.47
T-test p-value	0.0035
Bootstrapped p-value	0.0001
Cohen's d	0.3492

Visualization

A Histogram and boxplot illustrate the AOV disparity between groups.



Insights and Recommendations

Business Insights

Promo codes increased AOV by $3.34, a statistically significant lift ($p < 0.05$) with a small-to-moderate effect ($d = 0.3492$). The 95% CI ($1.13–$5.47) confirms a positive revenue impact. This suggests promo codes encourage higher spending, likely by motivating larger carts.

Summary: This A/B test evaluated the efficacy of promo codes on order value in a retail store dataset. The treatment group (with promo codes) showed a statistically significant increase in average order value (AOV) of $3.34 compared to the control group ($79.01 vs. $75.67), with a 95% confidence interval of $1.13 to $5.47 and a small-to-moderate effect size (Cohen's $d = 0.3492$). Both original ($p = 0.0035$) and bootstrapped ($p = 0.0001$) t-tests confirmed the result, despite a sample imbalance (90 treatment vs. 335 control), which was addressed through bootstrapping. While promo codes effectively boost AOV, their modest practical impact suggests that their cost-effectiveness should be evaluated in the context of discount amounts and broader business goals.

Recommendation:

Implement promo codes strategically, testing discount levels to maximize profit. Further segmentation (e.g., by Gender) could refine targeting.

Closing

Conclusion

This analysis demonstrates expertise in A/B testing, statistical rigor, custom scripting, and visualization, delivering actionable revenue insights. It reflects a 10-day process with two feedback rounds, ensuring client alignment.