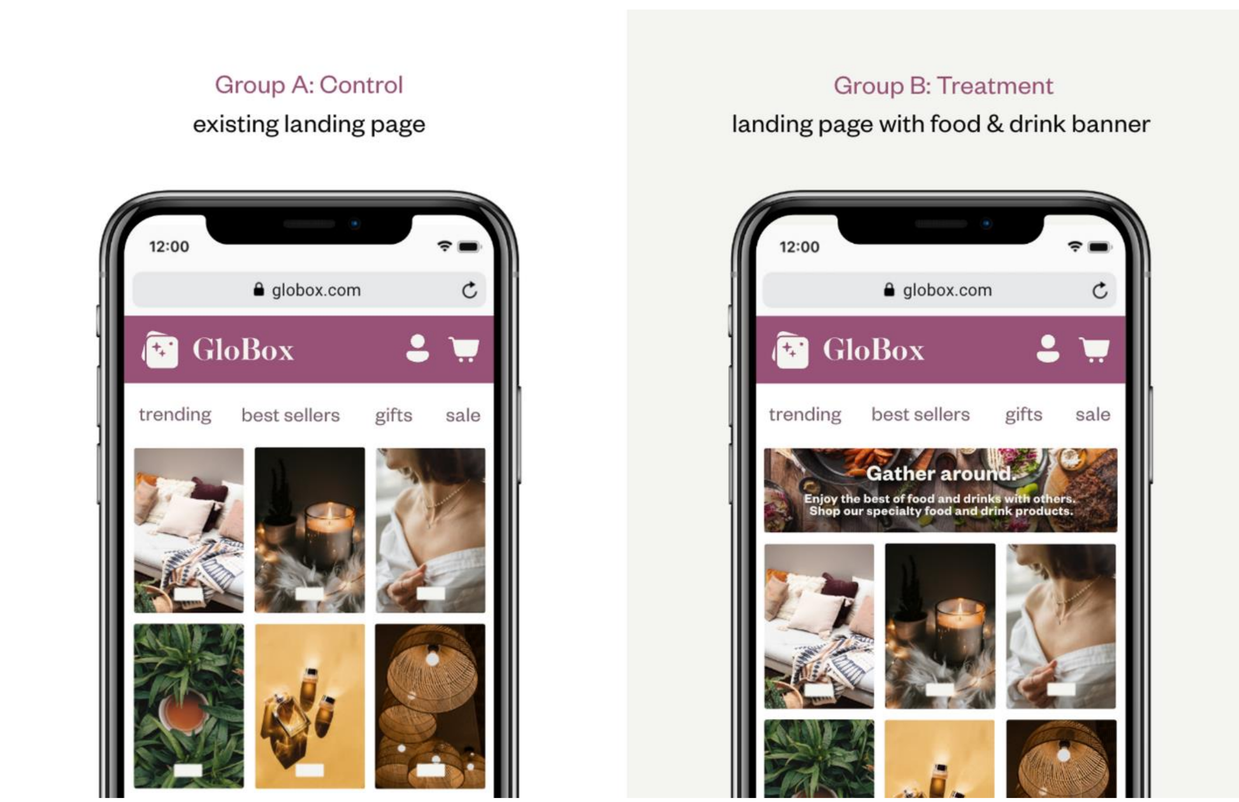
Report for A/B testing for GloBox(e-commerce)

Project Background

An A/B test is an experimentation technique used by businesses to compare two versions of a webpage, advertisement, or product feature to determine which one performs better. By randomly assigning customers or users to either the A or B version, the business can determine which version is more effective at achieving a particular goal.

The dataset provided is for an e-commerce company called GloBox. GloBox is an online marketplace that specializes in sourcing unique and high-quality products from around the world.
GloBox is primarily known amongst its customer base for boutique fashion items and high-end decor products. However, their food and drink offerings have grown tremendously in the last few months, and the company wants to bring awareness to this product category to increase revenue.
The Growth team decides to run an A/B test that highlights key products in the food and drink category as a banner at the top of the website. The control group does not see the banner, and the test group sees it as shown below:



The setup of the A/B test is as follows:

1. The experiment is only being run on the mobile website.
2. A user visits the GloBox main page and is randomly assigned to either the control or test group. This is the join date for the user.
3. The page loads the banner if the user is assigned to the test group, and does not load the banner if the user is assigned to the control group.

4. The user subsequently may or may not purchase products from the website. It could be on the same day they join the experiment, or days later. If they do make one or more purchases, this is considered a “conversion”.

My task as a data analyst team member is to analyse the results of the A/B test and provide a recommendation to your stakeholders about whether GloBox should launch the experience to all users and I have used Python and its libraries to study, analyze, explore, and visualize data as well as areas that present an opportunity to learn from successes. The A/B test is being conducted to increase the revenue in the food and drink category on the GloBox website by testing a banner highlighting key products. I have performed z test, t test, confidence interval.

Experiment Design

The metrics used for this test are revenue by treatment group, revenue by country and revenue by gender. The test is carried out at a 95% level of significance.

HYPOTHESIS:

Null Hypothesis: There is no significant difference in total revenue between the control and treatment group.

Alternate Hypothesis: The treatment group will have a significantly higher total revenue than the control group.

It is important to note if the p-value is less than 0.05, the null hypothesis will not be accepted.

Metric Choice

Conversion Rate

Average Order value

Exploratory Data Analysis

The metrics for control group showed that were 24,343 users who placed the total of 24, 313 orders resulting a total revenue 82,145.90 and an average order value 3.37.

The metrics for treatment group showed that were 24,600 users who placed the total of 24, 600 orders resulting a total revenue 83,415.30 and an average order value 3.39.

The analysis shows that the treatment group had a higher conversion rate than the control group. The conversion rate for the control group was 0.039 while treatment group was 0.046 represents 0.71% increase.

The difference in average order value between the two groups was 1.63% with version B having a higher average order value.

Statistical Analysis:

At a confidence interval of 95%, the alpha value of 0.05 and the probability of observing a result as extreme or more extreme than the one obtained if the null hypothesis is true is 0.0001 which is less than the alpha value suggesting strong evidence against the null hypothesis. This indicates results are statistically significant and the experiment is successful. A chi squared statistics of 14.76 the magnitude if this value indicates a moderate to strong degree of deviation from null hypothesis.

Results

Results of this A/B test provide evidence that version B of website is more effective in driving revenue than version A. Moreover, it is important to note other factors beyond website version may have influenced the results, such as time of day, gender of customers accessing the website or users from android or apple devices. I recommend performing the test over a longer period or with larger sample size.

The treatment group had a higher conversion rate, average order value, total revenue than the control group.

Appendix:

```
group_avg_purchased = data.groupby('group')['purchased_amount'].mean()

print("Average amount spent per user for the control group: ",
group_avg_purchased['A'])
print("Average amount spent per user for the control group: ",
group_avg_purchased['B'])

Average amount spent per user for the control group:  3.3663584952053114
Average amount spent per user for the control group:  3.379875202593193
```

What is the 95% confidence interval for the average amount spent per user in the control?

```
control_data = data[data['group'] == 'A']['purchased_amount']
# Calculate the sample mean and standard deviation

sample_mean = control_data.mean()
sample_std = control_data.std(ddof=1)

# Calculate the standard error of the mean
n = len(control_data)
sem = sample_std / (n ** 0.5)

# Calculate the t-value for a 95% confidence interval
t = stats.t.ppf(0.975, n - 1)

# Calculate the confidence interval
lower_ci = sample_mean - t * sem
upper_ci = sample_mean + t * sem
```

```
print("95% Confidence Interval for Average Amount Spent per User in Control:
[{:.2f}, {:.2f}]".format(lower_ci, upper_ci))
95% Confidence Interval for Average Amount Spent per User in Control: [3.07,
3.69]

## Conduct a hypothesis test to see whether there is a difference in the average
amount spent per user between the two groups. What are the resulting p-value and
conclusion? This question is required.*
##Use the t distribution and a 5% significance level. Assume unequal variance.##

# Calculate the sample means and standard deviations

control = data[data['group'] == 'A']
treatment = data[data['group'] == 'B']

ctrl_mean = control.mean()
trtm_mean = treatment.mean()
ctrl_std = control.std(ddof=1)
trtm_std = treatment.std(ddof=1)

# Calculate the pooled standard deviation
sp = ((n1-1)*ctrl_std**2 + (n2-1)*trtm_std**2)/(n1+n2-2)
sp = sp**0.5

# Calculate the t-value and p-value for a two-tailed test
n1 = len(control)
n2 = len(treatment)
t = (ctrl_mean - trtm_mean)/(sp*(1/n1 + 1/n2))**0.5
df = n1 + n2 - 2
p = stats.t.sf(abs(t), df)*2

print("t-value:", t)
print("p-value:", p)

# Test the hypothesis at the 5% significance level
alpha = 0.05
if p < alpha:
    print("Reject the null hypothesis - there is no difference in the mean amount
spent per user between the control and treatment")
else:
    print("Fail to reject the null hypothesis - there is not a significant
difference in average amount spent per user between the two groups.")
t-value: -0.059699361967889275
95% Confidence Interval: (-0.4572888372649684, 0.43025542248920506)
```

What is the user conversion rate for the control and treatment groups? This question is required.*

```
# Calculate the conversion rates for the control and treatment groups
control_data = data.loc[data['user_group'] == 'Control Group', ['id',
'purchased_amount', 'user_group']]
treatment_data = data.loc[data['user_group'] == 'Test Group', ['id',
'purchased_amount', 'user_group']]

control_converted = control_data.loc[control_data['purchased_amount'] > 0,
'id'].nunique()
treatment_converted = treatment_data.loc[treatment_data['purchased_amount'] > 0,
'id'].nunique()

control_total = control_data['id'].nunique()
treatment_total = treatment_data['id'].nunique()

control_conversion_rate = control_converted / control_total
treatment_conversion_rate = treatment_converted / treatment_total

print("Control conversion rate:", control_conversion_rate)
print("Treatment conversion rate:", treatment_conversion_rate)
```

```
Control conversion rate: 0.03923099042845993
Treatment conversion rate: 0.04630081300813008
```

```
# Calculate the conversion rate and standard error for the Treatment group
control_data = data.loc[data['user_group'] == 'Test Group', ['id',
'purchased_amount', 'user_group']]
control_converted = control_data.loc[control_data['purchased_amount'] > 0,
'id'].nunique()
control_total = control_data['id'].nunique()
control_p = control_converted / control_total
control_se = np.sqrt(control_p*(1-control_p)/control_total)

# Calculate the confidence interval for the control group
conf_level = 0.95  # 95% confidence interval
z_score = norm.ppf(1 - (1 - conf_level) / 2)
ci_lower = control_p - z_score * control_se
ci_upper = control_p + z_score * control_se

print("Treatment conversion rate:", control_p)
print("95% Confidence Interval for Treatment Conversion Rate: ({:.4f},
{:.4f})".format(ci_lower, ci_upper))
```

```
Treatment conversion rate: 0.04630081300813008
95% Confidence Interval for Treatment Conversion Rate: (0.0437, 0.0489)
```

Conduct a hypothesis test to see whether there is a difference in the conversion rate between the two groups. What are the resulting p-value and conclusion?

```
# Calculate the number of conversions and users in the control and treatment groups
control_conversions = data.loc[data['user_group'] == 'Control Group', 'purchased_amount'].sum()
control_users = data.loc[data['user_group'] == 'Control Group', 'id'].nunique()
treatment_conversions = data.loc[data['user_group'] == 'Test Group', 'purchased_amount'].sum()
treatment_users = data.loc[data['user_group'] == 'Test Group', 'id'].nunique()

# Calculate the conversion rates for the control and treatment groups
control_conversion_rate = control_conversions / control_users
treatment_conversion_rate = treatment_conversions / treatment_users

# Calculate the pooled proportion for the standard error
pooled_proportion = (control_conversions + treatment_conversions) / (control_users + treatment_users)

# Calculate the standard error of the difference in proportions using the pooled proportion
standard_error = np.sqrt(pooled_proportion * (1 - pooled_proportion) * ((1 / control_users) + (1 / treatment_users)))

# Calculate the test statistic (z-score)
z_score = (treatment_conversion_rate - control_conversion_rate) / standard_error

# Calculate the p-value
p_value = 2 * norm.sf(abs(z_score))
print("p-value:",p_value)

# Compare the p-value with the significance level and make a conclusion
significance_level = 0.05
if p_value < significance_level:
    print("Reject the null hypothesis and conclude that there is a difference in the conversion rate between the two groups.")
else:
    print("Fail to reject the null hypothesis and conclude that there is no difference in the conversion rate between the two groups.")
```

```
p-value: nan
Fail to reject the null hypothesis and conclude that there is no difference in
the conversion rate between the two groups.
```

```
from scipy import stats

## Null Hypotheses is no difference if p value < 0.05 -- reject null hypotheses
control_group = data[data['group']=='A']['purchased_amount']
test_group = data[data['group']=='B']['purchased_amount']




t_test, p_value = stats.ttest_ind(control_group,test_group , equal_var=False)
t_test
-0.0596916223061378
p_value
0.9524014875835265
```

```
import pandas as pd
import numpy as np
from scipy import stats

df = pd.read_csv('GloBox_Dataset.csv')
df['purchased_amount'] = pd.to_numeric(df['purchased_amount'], errors = 'coerce')
conversion_rate_control = control_group['conversion'].mean()
n_control = len(control_group)
n_converted_control = control_group['conversion'].sum()

conversion_rate_test = test_group['conversion'].mean()
n_test = len(test_group)
n_converted_test = test_group['conversion'].sum()
pooled_proportion = (n_converted_control + n_converted_test)/(n_control + n_test)
std_error = np.sqrt(pooled_proportion * (1 - pooled_proportion) * (1/n_control +
1/n_test))

#Calculate the z-score
z_score = (conversion_rate_test - conversion_rate_control) / std_error

p_value = 2 * (1 - stats.norm.cdf(abs(z_score)))
p_value
3.093173032975294e-05
```

```python
# separate the purchases made by the control and test groups
control_purchases = df[df['user_group'] == 'Control Group']['purchased_amount']
test_purchases = df[df['user_group'] == 'Test Group']['purchased_amount']

# calculate the sample size, mean, and standard deviation for each group
control_n = len(control_purchases)
control_mean = control_purchases.mean()
control_std = control_purchases.std(ddof=1)  # ddof=1 for sample standard
deviation

test_n = len(test_purchases)
test_mean = test_purchases.mean()
test_std = test_purchases.std(ddof=1)

# calculate the standard error of the difference
std_error = np.sqrt((control_std**2 / control_n) + (test_std**2 / test_n))

# set the desired confidence level
confidence_level = 0.95

# calculate the critical value (two-tailed test)
degrees_of_freedom = control_n + test_n - 2
critical_value = t.ppf((1 + confidence_level) / 2, df=degrees_of_freedom)

# calculate the margin of error
margin_of_error = critical_value * std_error

# calculate the confidence interval
lower_bound = (test_mean - control_mean) - margin_of_error
upper_bound = (test_mean - control_mean) + margin_of_error
print(f"Confidence Interval: [{lower_bound:.2f}, {upper_bound:.2f}]")
Confidence Interval: [-14.22, -10.94]
```