# Mastery Project 1 :

## Analyzing the impact of landing food and drink banner on conversion rate

In [1]:
```
import numpy as np
import pandas as pd
from scipy import stats
import statsmodels.api as sm
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm
```

In [2]:
```
df = pd.read_csv('/Users/siminjahankhah/Desktop/Master_School/Mastery_project_1/ABTest.csv')
```

In [3]:
```
df
```

Out[3]:

| | id | join_dt | group | country | gender | device | total_spent |
|---|---|---|---|---|---|---|---|
| **0** | 1000000 | 2023-01-28 | B | CAN | M | I | 0.0 |
| **1** | 1000001 | 2023-01-27 | A | BRA | M | A | 0.0 |
| **2** | 1000002 | 2023-02-01 | A | FRA | M | A | 0.0 |
| **3** | 1000003 | 2023-01-25 | B | BRA | M | I | 0.0 |
| **4** | 1000004 | 2023-02-04 | A | DEU | F | A | 0.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **48938** | 1049995 | 2023-02-03 | B | BRA | F | A | 0.0 |
| **48939** | 1049996 | 2023-01-29 | A | USA | F | A | 0.0 |
| **48940** | 1049997 | 2023-02-03 | B | BRA | M | A | 0.0 |
| **48941** | 1049998 | 2023-02-03 | B | CAN | M | I | 0.0 |
| **48942** | 1049999 | 2023-01-29 | B | GBR | M | I | 0.0 |

48943 rows × 7 columns

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48943 entries, 0 to 48942
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   id           48943 non-null  object 
 1   join_dt      48943 non-null  object 
 2   group        48943 non-null  object 
 3   country      48300 non-null  object 
 4   gender       42088 non-null  object 
 5   device       48649 non-null  object 
 6   total_spent  48943 non-null  float64
dtypes: float64(1), object(6)
memory usage: 2.6+ MB
```

In [6]:
```
# group by the 'group' column and compute the statistics for 'totel_spent'
grouped_stats = df.groupby('group')['total_spent'].agg(['mean', 'median', 'max', 'min'])

print(grouped_stats)
```

```
           mean  median     max  min
group                               
A      3.374518     0.0  1659.4  0.0
B      3.390867     0.0  1546.3  0.0
```

In [7]:
```
# Split data into control and treatment groups
control_df = df[df['group'] == 'A']
treatment_df = df[df['group'] == 'B']
```

## Conversion rate for Control and treatment groups:

## H0: there is no difference in the conversion rate between control and treatment groups.

## H1: there is a difference in the conversion rate between the two groups

In [8]:
```
import numpy as np
import scipy.stats as stats

# assuming df1 is the control group dataframe and df2 is the treatment group dataframe
control = df[df['group'] == 'A']
treatment = df[df['group'] == 'B']

# calculate the conversion rate for each group
n1 = len(control)
n2 = len(treatment)
c1 = sum(control['total_spent'] > 0)
c2 = sum(treatment['total_spent'] > 0)
p1 = c1 / n1
p2 = c2 / n2

# calculate the pooled proportion for the standard error
pooled_p = (c1 + c2) / (n1 + n2)

# calculate the standard error of the difference in proportions
std_error = np.sqrt(pooled_p * (1 - pooled_p) * (1/n1 + 1/n2))

# calculate the z-score
z_score = (p2 - p1) / std_error

# calculate the p-value
p_value = 2 * (1 - stats.norm.cdf(abs(z_score)))

print("The p-value for the hypothesis test is:", round(p_value,5))
```

```
The p-value for the hypothesis test is: 0.00011
```

## Conclusion:

p-value of 0.00011 indicates that there is strong evidence against the null hypothesis and suggests that there is a statistically significant difference in conversion rates between the control and treatment groups. Typically, a significance level of 0.05 (or 5%) is used, which means that we reject the null hypothesis

==========

## Country VS Groups

### H0 : There is no significant difference in the mean total_spent between the different levels of country and group and their interaction term.

### H1 : There is a significant difference in the mean total_spent between at least one pair of the levels of the country and group or their interaction term.

In [9]:
```
# Perform two-way ANOVA to test for interaction between country and group
model1 = ols('total_spent ~ C(country)*C(group)', data=df).fit()
anova_table1 = anova_lm(model1, typ=2)

# Print results
print(anova_table1)
```

```
                        sum_sq       df         F    PR(>F)
C(country)        1.766006e+04      9.0  2.960113  0.001607
C(group)          3.496624e-01      1.0  0.000527  0.981677
C(country):C(group)  7.555165e+03      9.0  1.266368  0.249488
Residual          3.200432e+07  48280.0       NaN       NaN
```

We can see that there is a significant difference in spending amounts for users from different countries (C(country)) with a p-value of 0.0016. The interaction between country and group (C(country):C(group)) is not significant with a p-value of 0.249.

Overall, this suggests that users from certain countries tend to spend more, regardless of whether they are in the control or treatment group.

## What about each counrty? (Total Spent amount)

In [10]:
```
import pandas as pd
from scipy.stats import ttest_ind
import statsmodels.stats.multitest as smm

# Initialize empty list to store p-values
p_values = []

# Loop through each country
for country in df['country'].unique():
    # Filter the data for the given country
    country_data = df[df['country'] == country]
    # Split the data into control and treatment groups
    control = country_data[country_data['group'] == 'A']
    treatment = country_data[country_data['group'] == 'B']
    # Conduct the t-test
    t, p = ttest_ind(control['total_spent'], treatment['total_spent'], equal_var=False)
    # Append the p-value to the list
    p_values.append(p)
    # Print the results
    print(f"Country: {country}")
    print(f"  Control group mean: {control['total_spent'].mean()}")
    print(f"  Treatment group mean: {treatment['total_spent'].mean()}")
    print(f"  p-value: {p}")
```

```
Country: CAN
  Control group mean: 3.6019038287706304
  Treatment group mean: 4.198567870485679
  p-value: 0.5365548986984023
Country: BRA
  Control group mean: 3.2139327391621166
  Treatment group mean: 3.0661169017829324
  p-value: 0.7917766506484263
Country: FRA
  Control group mean: 2.6778735894236507
  Treatment group mean: 2.268101311713363
  p-value: 0.4704874606700735
Country: DEU
  Control group mean: 3.4005885173465225
  Treatment group mean: 2.708085215452753
  p-value: 0.4983184937017483
Country: GBR
  Control group mean: 2.1087423297123675
  Treatment group mean: 4.49800401593943
  p-value: 0.07386538407069568
Country: ESP
  Control group mean: 2.17839518555667
  Treatment group mean: 3.234237312641462
  p-value: 0.18564672043509706
Country: USA
  Control group mean: 4.295363216828924
  Treatment group mean: 4.053451967455165
  p-value: 0.5737095115770742
Country: AUS
  Control group mean: 1.6683881578947368
  Treatment group mean: 2.0806250000000004
  p-value: 0.6034465766823374
Country: MEX
  Control group mean: 2.8119463174874166
  Treatment group mean: 3.345506017338939
  p-value: 0.39632400205822316
Country: TUR
  Control group mean: 3.6853754993400765
  Treatment group mean: 2.4889571850271714
  p-value: 0.1001421321766267
Country: nan
  Control group mean: nan
  Treatment group mean: nan
  p-value: nan
```

Based on these results, there is no strong evidence to suggest a significant difference in total spent between the control and treatment groups for most countries. However, in the United Kingdom and Turkey, there might be some weak evidence of a potential difference, but further investigation is needed.

## (Conversion Rate)

In [11]:
```
import numpy as np
import scipy.stats as stats

# Initialize empty list to store p-values
p_values = []

# Loop through each country
for country in df['country'].unique():
    # Filter the data for the given country
    country_data = df[df['country'] == country]
    # Split the data into control and treatment groups
    control = country_data[country_data['group'] == 'A']
    treatment = country_data[country_data['group'] == 'B']

    # Calculate the conversion rate for each group
    n1 = len(control)
    n2 = len(treatment)

    # Check for zero denominators
    if n1 == 0 or n2 == 0:
        print(f"Skipping country {country} due to missing data.")
        continue

    c1 = sum(control['total_spent'] > 0)
    c2 = sum(treatment['total_spent'] > 0)
    p1 = c1 / n1
    p2 = c2 / n2

    # Calculate the pooled proportion for the standard error
    pooled_p = (c1 + c2) / (n1 + n2)

    # Calculate the standard error of the difference in proportions
    std_error = np.sqrt(pooled_p * (1 - pooled_p) * (1/n1 + 1/n2))

    # Calculate the z-score
    z_score = (p2 - p1) / std_error

    # Calculate the p-value
    p_value = 2 * (1 - stats.norm.cdf(abs(z_score)))

    # Append the p-value to the list
    p_values.append(p_value)

    # Print the results
    print(f"Country: {country}")
    print(f"  Control group conversion rate: {p1}")
    print(f"  Treatment group conversion rate: {p2}")
    print(f"  p-value: {round(p_value, 5)}")
```

```
Country: CAN
  Control group conversion rate: 0.0469361147327249
  Treatment group conversion rate: 0.0647571606475716
  p-value: 0.1249
Country: BRA
  Control group conversion rate: 0.0372528616024974
  Treatment group conversion rate: 0.040613523439187726
  p-value: 0.39872
Country: FRA
  Control group conversion rate: 0.03125
  Treatment group conversion rate: 0.04182754182754183
  p-value: 0.11729
Country: DEU
  Control group conversion rate: 0.032004197271773345
  Treatment group conversion rate: 0.044147843942505136
  p-value: 0.04909
Country: GBR
  Control group conversion rate: 0.0288659793814433
  Treatment group conversion rate: 0.03681392235609103
  p-value: 0.22633
Country: ESP
  Control group conversion rate: 0.029087261785356068
  Treatment group conversion rate: 0.03614457831325301
  p-value: 0.37515
Country: USA
  Control group conversion rate: 0.05116979066903817
  Treatment group conversion rate: 0.05748358568940105
  p-value: 0.09061
Country: AUS
  Control group conversion rate: 0.02138157894736842
  Treatment group conversion rate: 0.030357142857142857
  p-value: 0.33269
Country: MEX
  Control group conversion rate: 0.029484902309058616
  Treatment group conversion rate: 0.04447485460143688
  p-value: 0.00268
Country: TUR
  Control group conversion rate: 0.04002163331530557
  Treatment group conversion rate: 0.03558151885289432
  p-value: 0.47691
Skipping country nan due to missing data.
```

## Conclusion:

There is a significant difference in conversion rates between the control group and treatment group for "Germany" and "Mexico".

## Gender VS Group

## H0: There is no significant difference in conversion rate between the control and treatment groups for gender.

## H1: There is a significant difference in conversion rate between each gender and treatment and control groups.

In [12]:
```
import numpy as np
import scipy.stats as stats

# Initialize empty list to store p-values
p_values = []

# Loop through each gender
for gender in df['gender'].unique():
    # Filter the data for the given gender
    gender_data = df[df['gender'] == gender]
    # Split the data into control and treatment groups
    control = gender_data[gender_data['group'] == 'A']
    treatment = gender_data[gender_data['group'] == 'B']

    # Calculate the conversion rate for each group
    n1 = len(control)
    n2 = len(treatment)

    # Check for zero denominators
    if n1 == 0 or n2 == 0:
        print(f"Skipping gender {gender} due to missing data.")
        continue

    c1 = sum(control['total_spent'] > 0)
    c2 = sum(treatment['total_spent'] > 0)
    p1 = c1 / n1
    p2 = c2 / n2

    # Calculate the pooled proportion for the standard error
    pooled_p = (c1 + c2) / (n1 + n2)

    # Calculate the standard error of the difference in proportions
    std_error = np.sqrt(pooled_p * (1 - pooled_p) * (1/n1 + 1/n2))

    # Calculate the z-score
    z_score = (p2 - p1) / std_error

    # Calculate the p-value
    p_value = 2 * (1 - stats.norm.cdf(abs(z_score)))

    # Append the p-value to the list
    p_values.append(p_value)

    # Print the results
    print(f"Gender: {gender}")
    print(f"  Control group conversion rate: {p1}")
    print(f"  Treatment group conversion rate: {p2}")
    print(f"  p-value: {round(p_value, 5)}")

# Adjust p-values for multiple comparisons using the Bonferroni correction
adjusted_pvalues = smm.multipletests(p_values, alpha=0.05, method='bonferroni')[1]

# Print adjusted p-values
print("Adjusted p-values:")
for gender, p_value in zip(df['gender'].unique(), adjusted_pvalues):
    print(f"Gender: {gender}")
    print(f"  Adjusted p-value: {round(p_value, 5)}")
```

```
Gender: M
  Control group conversion rate: 0.0262582056892779
  Treatment group conversion rate: 0.03790913531998046
  p-value: 0.0
Gender: F
  Control group conversion rate: 0.05144502929784487
  Treatment group conversion rate: 0.05436835304641686
  p-value: 0.35422
Skipping gender nan due to missing data.
Gender: O
  Control group conversion rate: 0.03217821782178218
  Treatment group conversion rate: 0.030197444831591175
  p-value: 0.81595
Adjusted p-values:
Gender: M
  Adjusted p-value: 1e-05
Gender: F
  Adjusted p-value: 1.0
Gender: nan
  Adjusted p-value: 1.0
```

## Conclusion:

There is a statistically significant difference in conversion rates between the control and treatment groups for males, while no significant differences are observed for females and the other gender category.

## Device VS Group

## H0: There is no significant difference in conversion rate between control and treatment groups for devices.

## H1: There is a significant difference in conversion rate between treatment and control groups for devices.

In [20]:
```
import numpy as np
import scipy.stats as stats
from statsmodels.stats.multitest import multipletests

# Initialize empty list to store p-values
p_values = []

# Loop through each device
for device in df['device'].unique():
    # Filter the data for the given device
    device_data = df[df['device'] == device]
    # Split the data into control and treatment groups
    control = device_data[device_data['group'] == 'A']
    treatment = device_data[device_data['group'] == 'B']

    # Calculate the conversion rate for each group
    n1 = len(control)
    n2 = len(treatment)

    # Check for zero denominators
    if n1 == 0 or n2 == 0:
        print(f"Skipping device {device} due to missing data.")
        continue

    c1 = sum(control['total_spent'] > 0)
    c2 = sum(treatment['total_spent'] > 0)
    p1 = c1 / n1
    p2 = c2 / n2

    # Calculate the pooled proportion for the standard error
    pooled_p = (c1 + c2) / (n1 + n2)

    # Calculate the standard error of the difference in proportions
    std_error = np.sqrt(pooled_p * (1 - pooled_p) * (1/n1 + 1/n2))

    # Calculate the z-score
    z_score = (p2 - p1) / std_error

    # Calculate the p-value
    p_value = 2 * (1 - stats.norm.cdf(abs(z_score)))

    # Append the p-value to the list
    p_values.append(p_value)

    # Print the results
    print(f"Device: {device}")
    print(f"  Control group conversion rate: {p1}")
    print(f"  Treatment group conversion rate: {p2}")
    print(f"  p-value: {round(p_value, 5)}")

# Adjust p-values for multiple comparisons using the Bonferroni correction
adjusted_pvalues = multipletests(p_values, alpha=0.05, method='bonferroni')[1]

# Print adjusted p-values
print("Adjusted p-values:")
for device, p_value in zip(df['device'].unique(), adjusted_pvalues):
    print(f"Device: {device}")
    print(f"  Adjusted p-value: {round(p_value, 5)}")
```

```
Device: I
  Control group conversion rate: 0.058521111354189456
  Treatment group conversion rate: 0.06465610761553482
  p-value: 0.08386
Device: A
  Control group conversion rate: 0.027700278995615783
  Treatment group conversion rate: 0.03524778470626846
  p-value: 0.00017
Skipping device nan due to missing data.
Adjusted p-values:
Device: I
  Adjusted p-value: 0.16771
Device: A
  Adjusted p-value: 0.00034
```

## Conclusion:

We conclude that there is no significant difference in conversion rate between the control and treatment groups for IOS device, while there is a significant difference for Android device.

## Some questions about data :

1. What is the average amount spent per user for the control and treatment groups?

In [13]:
```
df['total_spent'][df['group'] == 'A'].mean()
```

Out[13]:
```
3.374518467928841
```

In [14]:
```
df['total_spent'][df['group']== 'B'].mean()
```

Out[14]:
```
3.390866945885783
```

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

In [101]:
```
total_control = df.loc[df['group']== 'A','total_spent']

control_mean = total_control.mean()
control_std = total_control.std()

confidence_level = 0.95
degrees_of_freedom = len(total_control) - 1
t_value = stats.t.ppf((1 + confidence_level) / 2, degrees_of_freedom)

margin_error = t_value * (control_std/ np.sqrt(degrees_of_freedom))

low_control = control_mean - margin_error
high_control = control_mean + margin_error

print('95% Confidence Interval for the average amount of spent for control Group :  ' , low_control , ',' , high_contro
```

```
95% Confidence Interval for the average amount of spent for control Group :   3.048680945886285 , 3.7003559899713974
```

## 95% Confidence Interval for the average amount of spent for control Group : ( 3.05 , 3.70 )

3. What is the 95% confidence interval for the average amount spent per user in the treatment?

In [102]:
```
total_treat = df.loc[df['group']== 'B','total_spent']

treat_mean = total_treat.mean()
treat_std = total_treat.std()

confidence_level = 0.95
degrees_of_freedom = len(total_treat) - 1
t_value = stats.t.ppf((1 + confidence_level) / 2, degrees_of_freedom)

margin_error = t_value * (treat_std/ np.sqrt(degrees_of_freedom))

low_treat = treat_mean - margin_error
high_treat = treat_mean + margin_error

print('95% Confidence Interval for the average amount of spent for treatment Group :  ' , low_treat , ',' , high_treat)
```

```
95% Confidence Interval for the average amount of spent for treatment Group :   3.07326318772908 , 3.708470704042486
```

## 95% Confidence Interval for the average amount of spent for treatment Group : ( 3.073 , 3.71 )

4) 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? Use the t distribution and a 5% significance level. Assume unequal variance.

In [103]:
```
control = df[df['group'] == 'A']
treatment = df[df['group'] == 'B']

t_statistic, p_value = stats.ttest_ind(control['total_spent'], treatment['total_spent'])

print('t-statistic:', t_statistic)
print('p-value:', p_value)
```

```
t-statistic: -0.07043243220818624
p-value: 0.9438497659410893
```

## Conclusion : there is a 94% chance that the difference in average amount spent per user between the control and treatment groups is due to random chance, assuming that there is no true difference between the two groups.

6. What is the user conversion rate for the control and treatment groups?

9. 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? Use the normal distribution and a 5% significance level. Use the pooled proportion for the standard error.

In [109]:
```
import numpy as np
import scipy.stats as stats

# assuming df1 is the control group dataframe and df2 is the treatment group dataframe
control = df[df['group'] == 'A']
treatment = df[df['group'] == 'B']

# calculate the conversion rate for each group
n1 = len(control)
n2 = len(treatment)
c1 = sum(control['total_spent'] > 0)
c2 = sum(treatment['total_spent'] > 0)
p1 = c1 / n1
p2 = c2 / n2

# calculate the pooled proportion for the standard error
pooled_p = (c1 + c2) / (n1 + n2)

# calculate the standard error of the difference in proportions
std_error = np.sqrt(pooled_p * (1 - pooled_p) * (1/n1 + 1/n2))

# calculate the z-score
z_score = (p2 - p1) / std_error

# calculate the p-value
p_value = 2 * (1 - stats.norm.cdf(abs(z_score)))

print("The p-value for the hypothesis test is:", round(p_value,5))
```

```
The p-value for the hypothesis test is: 0.00011
```

## p-value of 0.00011 indicates that there is strong evidence against the null hypothesis and suggests that there is a statistically significant difference in conversion rates between the control and treatment groups. Typically, a significance level of 0.05 (or 5%) is used, which means that we reject the null hypothesis.

```
conversion_rate = n_converted / n_control
```

10. What is the 95% confidence interval for the difference in the conversion rate between the treatment and control (treatment-control)?

In [110]:
```
difference = p2 - p1
standard_error = np.sqrt(pooled_p * (1 - pooled_p) * (1/n2 + 1/n1))
dof = n2 + n1 - 2
margin_of_error = stats.t.ppf(0.975, dof) * standard_error
confidence_interval = (difference - margin_of_error, difference + margin_of_error)

print("The 95% confidence interval for the difference in conversion rates is ({:.2%}, {:.2%})".format(confidence_interv
```

```
The 95% confidence interval for the difference in conversion rates is (0.35%, 1.07%)
```

## The 95% confidence interval for the difference in conversion rates is (0.35%, 1.07%)

## Conclusion :

we are 95% confident that the true difference in conversion rates between the two groups lies between 0.35% and 1.07%.

```
n_treatment = len(treatment)

# calculate the number of converted users in the control group
n_converted = len(treatment[treatment['total_spent'] > 0])

# calculate the conversion rate in the control group
conversion_rate = n_converted / n_treatment

# calculate the standard error of the conversion rate
std_error = np.sqrt((conversion_rate * (1 - conversion_rate)) / n_treatment)

# calculate the 95% confidence interval for the conversion rate
ci_low, ci_high = stats.norm.interval(0.95, loc=conversion_rate, scale=std_error)

print("The 95% confidence interval for the conversion rate in the treatment group is ({:.2%}, {:.2%}).".format(ci_low,
```

```
The 95% confidence interval for the conversion rate in the treatment group is (4.37%, 4.89%).
```

## The 95% confidence interval for the conversion rate in the treatment group is (4.37%, 4.89%)