A/B TESTING MARKETING

DATA DESCRIPTION

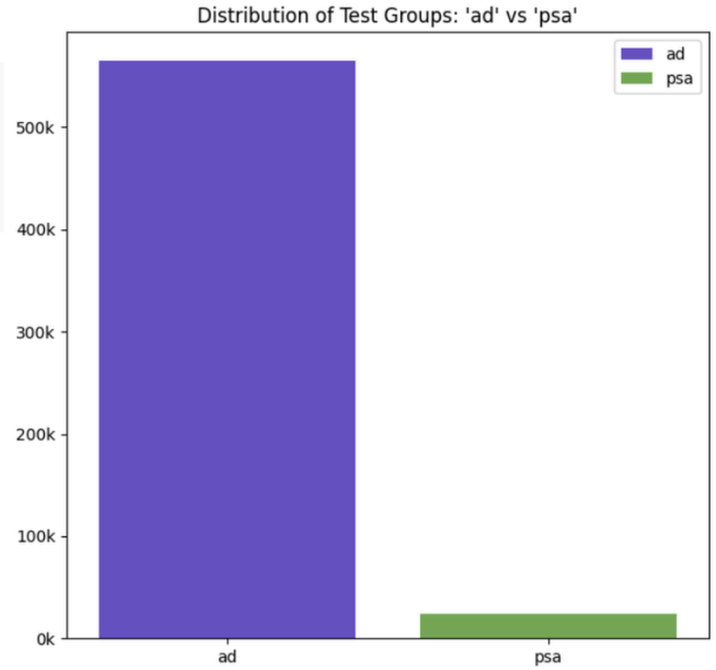
```
df = pd.read_csv('marketing_AB.csv')
df.head()
```

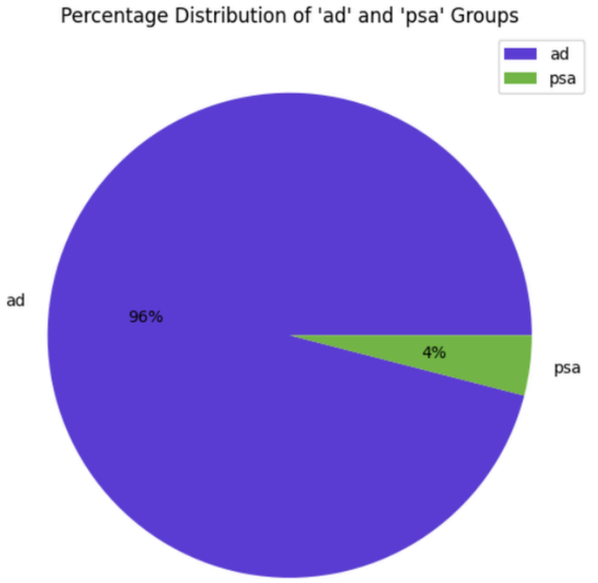
	Unnamed: 0	user id	test group	converted	total ads	most ads day	most ads hour
0	0	1069124	ad	False	130	Monday	20
1	1	1119715	ad	False	93	Tuesday	22
2	2	1144181	ad	False	21	Tuesday	18
3	3	1435133	ad	False	355	Tuesday	10
4	4	1015700	ad	False	276	Friday	14

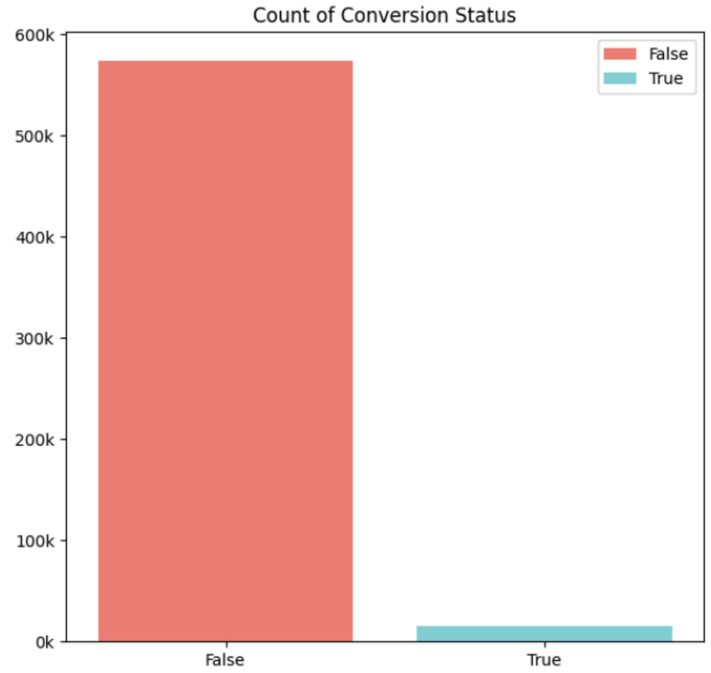
The idea of the dataset is to analyze the groups, find if the ads were successful, how much the company can make from the ads, and if the difference between the groups is statistically significant.

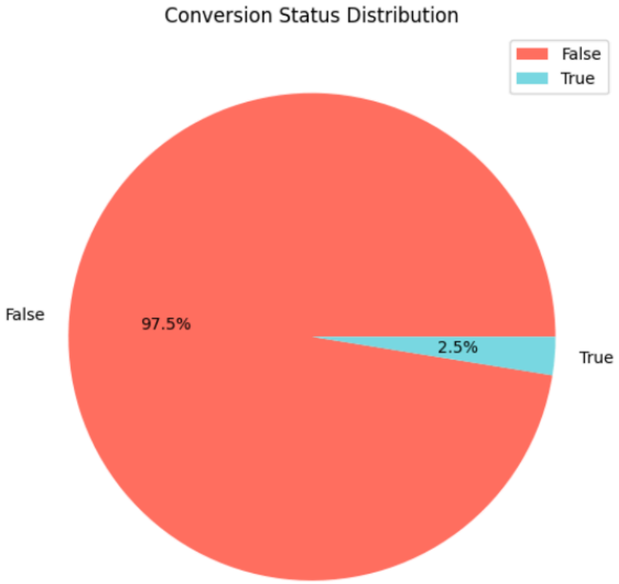
Data dictionary:

- Index: Row index
- user id: User ID (unique)
- test group: If "ad" the person saw the advertisement, if "psa" they only saw the public service announcement
- converted: If a person bought the product then True, else is False
- total ads: Amount of ads seen by person
- most ads day: Day that the person saw the biggest amount of ads
- most ads hour: Hour of day that the person saw the biggest amount of ads







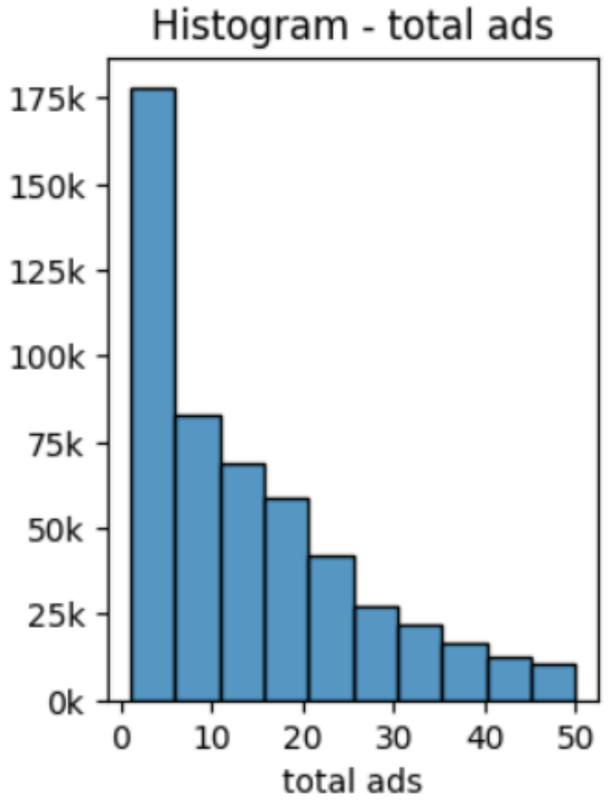


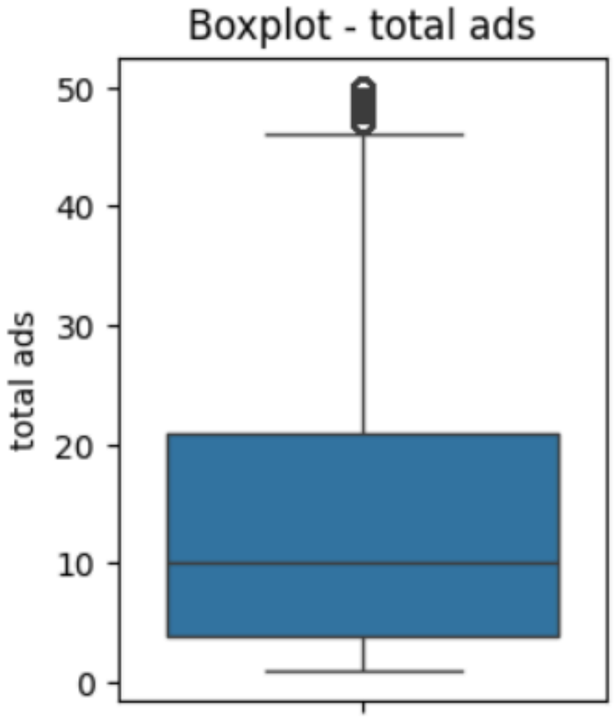
ANALYSIS OF DATASET CONTENT

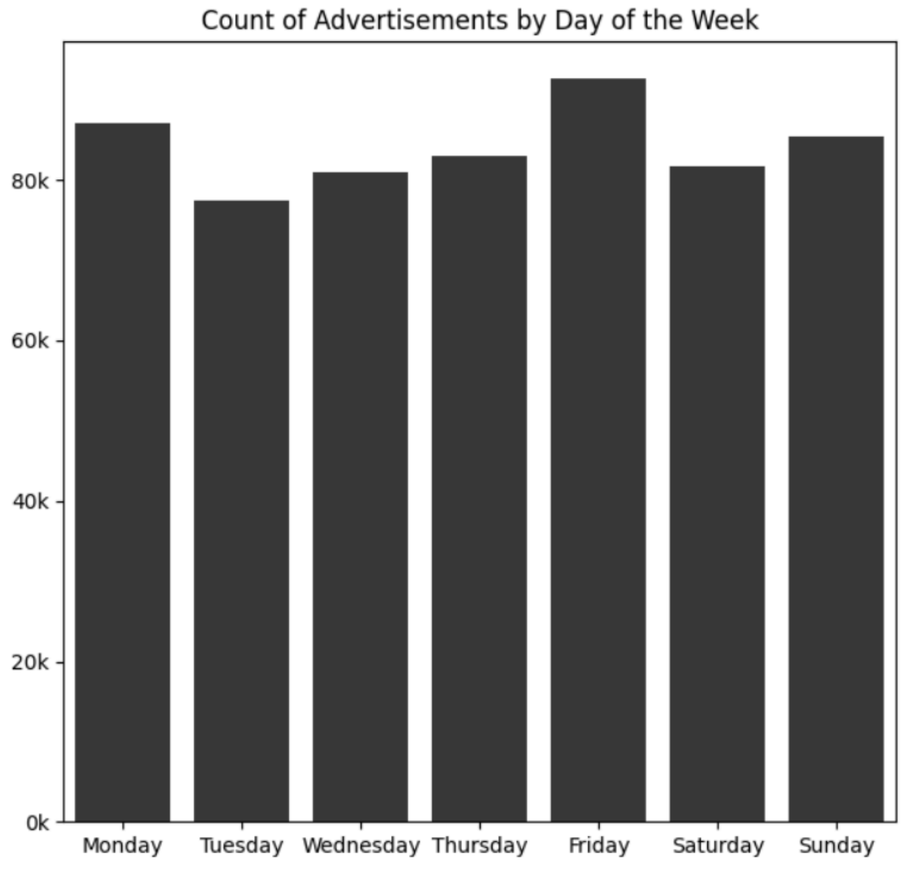
- ‘ad’ group dominated to ‘psa’ group
- users who didn't buy a product outnumber those who did

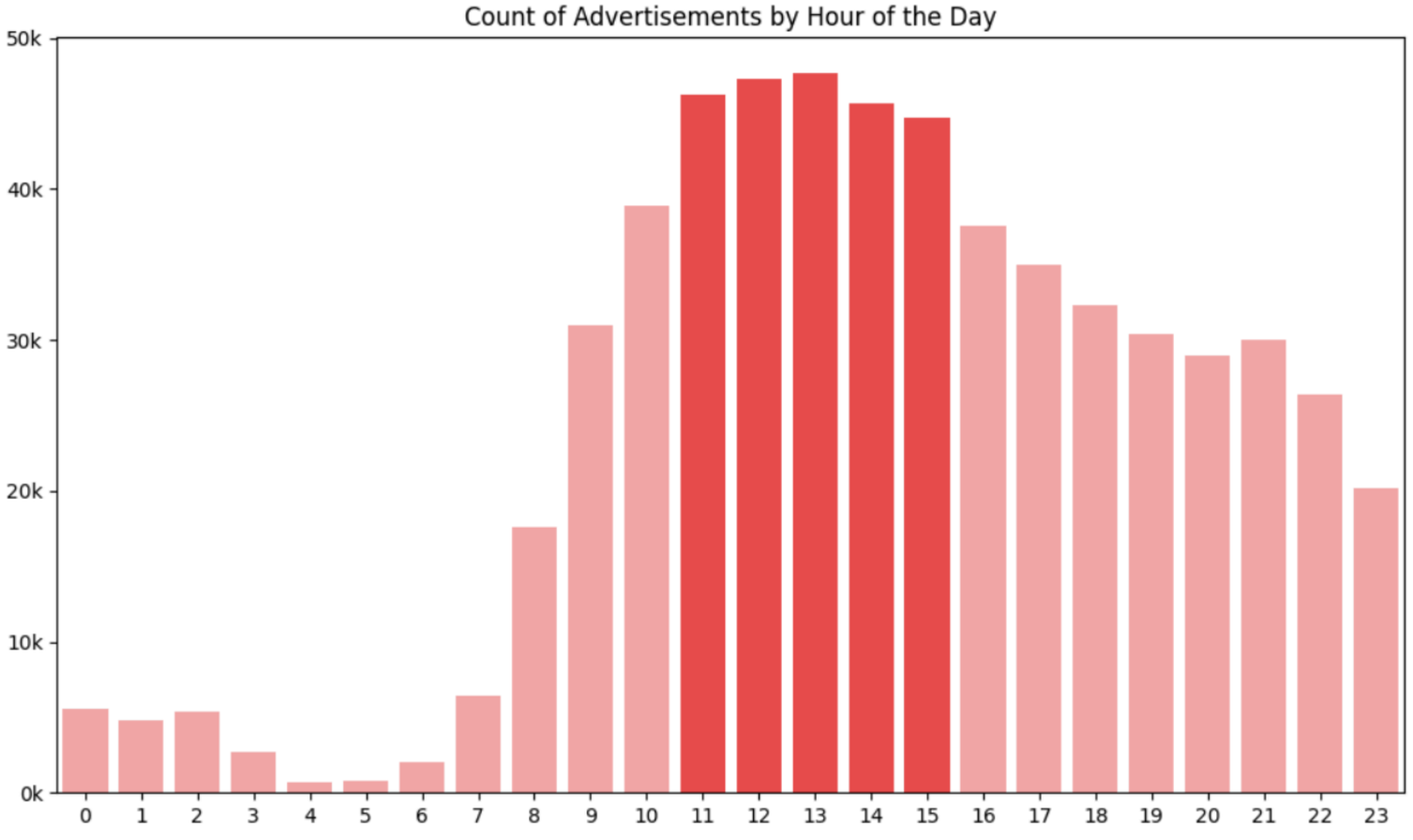
```
df['total ads'].describe()
```

```
count    588101.000000
mean         24.820876
std          43.715181
min           1.000000
25%           4.000000
50%          13.000000
75%          27.000000
max        2065.000000
```



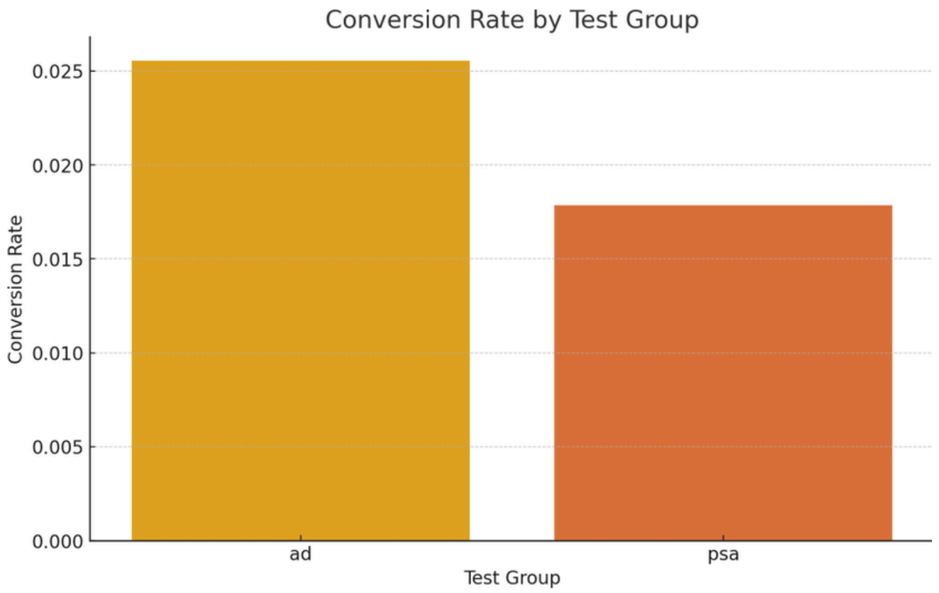



Count of Advertisements by Day of the Week
0k
20k
40k
60k
80k
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday


Count of Advertisements by Hour of the Day
0k
10k
20k
30k
40k
50k
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

Statistical Analysis: Z Test for Proportions

converted	False	True
test group		
ad	0.974453	0.025547
psa	0.982146	0.017854



Z-Test manualy

```
from scipy.stats import norm
group_ad = df[df['test group'] == 'ad']['converted']
group_psa = df[df['test group'] == 'psa']['converted']

n_ad = len(group_ad)
n_psa = len(group_psa)
success_ad = np.sum(group_ad)
success_psa = np.sum(group_psa)
# Proportions
p_ad = success_ad / n_ad
p_psa = success_psa / n_psa
# Combined proportion
p_combined = (success_ad + success_psa) / (n_ad + n_psa)
# Z-test for proportions formula
z_stat = (p_ad - p_psa) / np.sqrt(p_combined * (1 - p_combined) * (1/n_ad + 1/n_psa))
# Calculate p-value
p_value = 2 * (1 - norm.cdf(abs(z_stat)))
# Print results
print(f"Z-statistic: {z_stat:.4f}")
print(f"P-value: {p_value:.4f}")
alpha = 0.05  # significance level
if p_value < alpha:
    print("The difference in conversion rates is statistically significant.")
else:
    print("The difference in conversion rates is not statistically significant.")
```

```
Z-statistic: 7.3701
P-value: 0.0000
The difference in conversion rates is statistically significant.
```

Evaluating Statistical Significance of Differences

```
import numpy as np
import matplotlib.pyplot as plt

def cohen_d(x, y):
    return (np.mean(x) - np.mean(y)) / np.sqrt(((np.std(x) ** 2) + (np.std(y) ** 2)) / 2)

ad_conversions = df[df['test group'] == 'ad']['converted'].tolist()
psa_conversions = df[df['test group'] == 'psa']['converted'].tolist()
effect_size = cohen_d(ad_conversions, psa_conversions)
print(f"Effect size = {effect_size:.1}")
if abs(effect_size) < 0.2:
    print("Small effect: the difference between 'ad' and 'psa' conversions is negligible")
elif abs(effect_size) < 0.5:
    print("Medium effect: the difference between 'ad' and 'psa' conversions is moderate")
elif abs(effect_size) < 0.8:
    print("Large effect: the difference between 'ad' and 'psa' conversions is significant")
else:
    print("Very large effect: the difference between 'ad' and 'psa' conversions is very significant")
```

```
Effect size = 0.05
Small effect: the difference between 'ad' and 'psa' conversions is negligible
```

Independence Test for Categorical Variables

For group AD:

converted most ads day	False	True
Monday	0.966759	0.033241
Tuesday	0.969560	0.030440
Wednesday	0.974644	0.025356
Sunday	0.975380	0.024620
Friday	0.977535	0.022465
Thursday	0.978363	0.021637
Saturday	0.978693	0.021307

```python
# Create the crosstab dataframe
ct_conversion_day_ad = pd.crosstab(df_ad['most ads day'], df_ad['converted'])

# Perform chi-square test
chi2_stat, p_value, _, _ = stats.chi2_contingency(ct_conversion_day_ad)

print("This Test is for group AD\n")
print(f"Chi-square statistic: {chi2_stat:.4f}")
print(f"P-value: {p_value:.4f}")

# Interpret the result
alpha = 0.05  # significance level
if p_value < alpha:
    print("We reject the null hypothesis (H₀).\nThe variables 'most ads day' and 'converted' are dependent.")
else:
    print("We fail to reject the null hypothesis (H₀).\nThe variables 'most ads day' and 'converted' are independent.")
```

```
This Test is for group AD

Chi-square statistic: 412.7943
P-value: 0.0000
We reject the null hypothesis (H₀).
The variables 'most ads day' and 'converted' are dependent.
```

```
Test Group = AD

Conversion Rate (True) for Monday:
0.033241
Relative Percentage of Overall Conversion Rate compared to Monday:
-23.15%
Relative Percentage of Weekend Conversion Rate compared to Monday:
-30.92%
Relative Percentage of Weekday Conversion Rate compared to Monday:
-19.89%
```

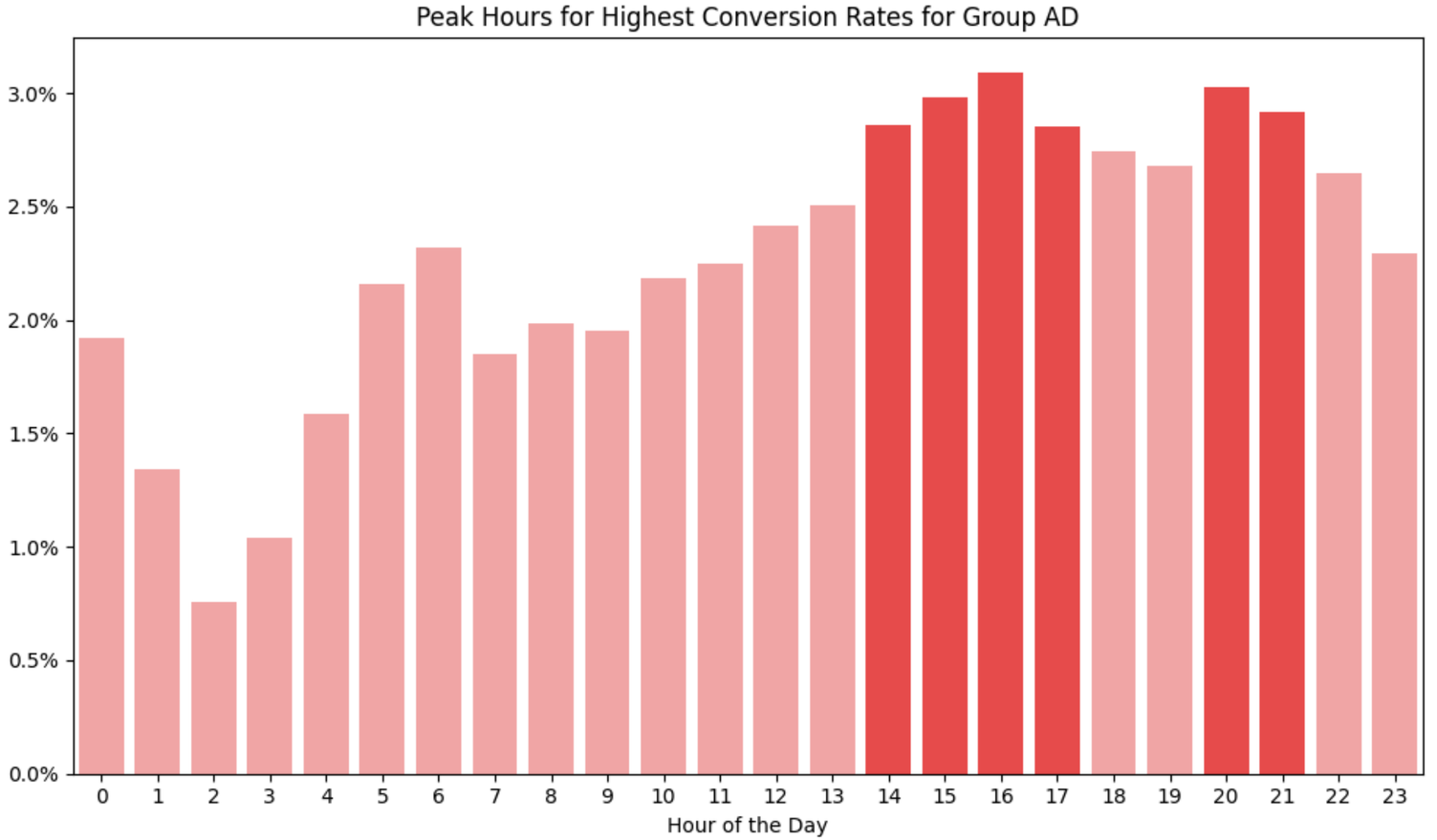
AD group
P-value matrix:

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Monday	1.000	0.002	0.000	0.000	0.000	0.000	0.000
Tuesday	0.002	1.000	0.000	0.000	0.000	0.000	0.000
Wednesday	0.000	0.000	1.000	0.000	0.000	0.000	0.354
Thursday	0.000	0.000	0.000	1.000	0.256	0.663	0.000
Friday	0.000	0.000	0.000	0.256	1.000	0.110	0.003
Saturday	0.000	0.000	0.000	0.663	0.110	1.000	0.000
Sunday	0.000	0.000	0.354	0.000	0.003	0.000	1.000

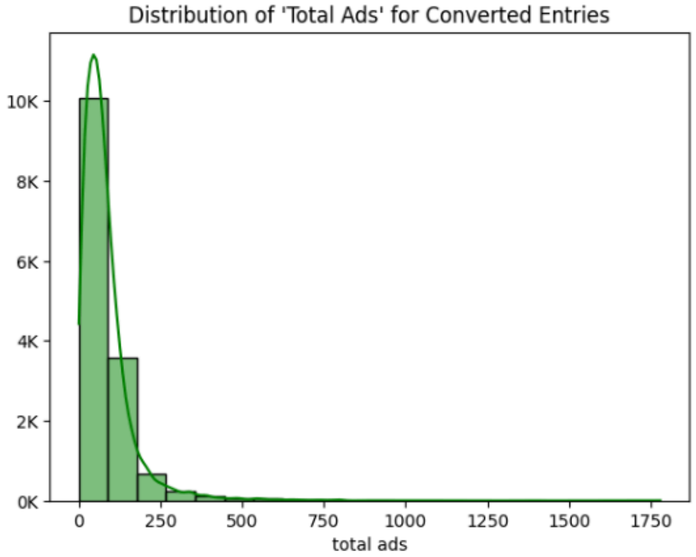
```
Test Group = AD

Combined Conversion Rate (True) for Monday and Tuesday:
0.031841

Relative Percentage of Overall Conversion Rate compared to Monday and Tuesday:
-19.77%

Relative Percentage of Weekend Conversion Rate compared to Monday and Tuesday:
-27.88%

Relative Percentage of Weekday Conversion Rate compared to Monday and Tuesday:
-16.37%
```

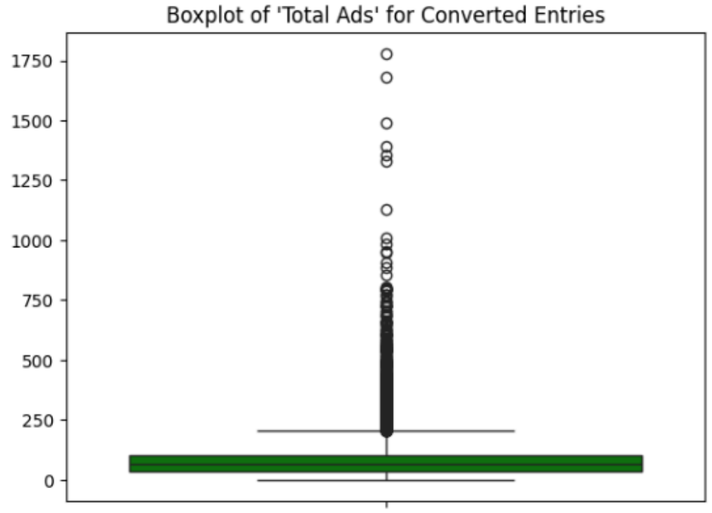
Peak Hours Analysis and Conversion Metrics

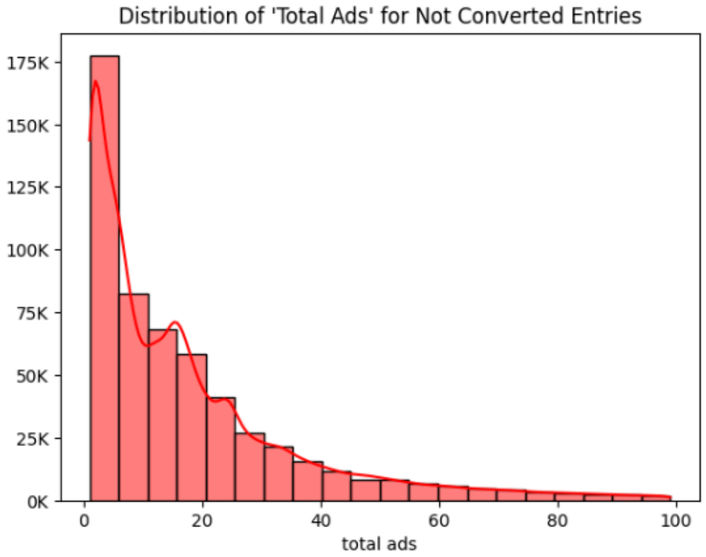
converted	False	True
most ads hour		
16	0.969107	0.030893
20	0.969726	0.030274
15	0.970155	0.029845
21	0.970825	0.029175
14	0.971425	0.028575
17	0.971463	0.028537
18	0.972530	0.027470
19	0.973191	0.026809
22	0.973545	0.026455
13	0.974937	0.025063
12	0.975861	0.024139

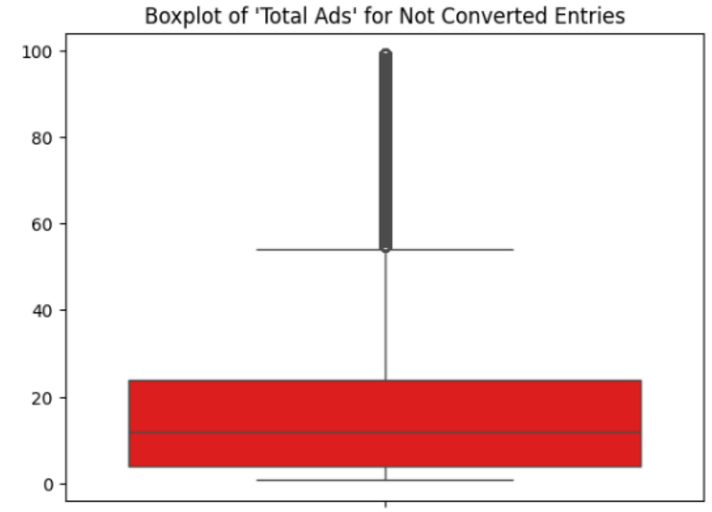


Distribution of Total Ads: Conversion Status & Normality Test










```python
import pandas as pd
from scipy.stats import shapiro

ads_converted = df_ad[df_ad['converted'] == True]['total ads']
ads_not_converted = df_ad[df_ad['converted'] == False]['total ads']

_,p_value_true = shapiro(df_ad[df_ad['converted'] == True]['total ads'])
_,p_value_false = shapiro(df_ad[df_ad['converted'] == False]['total ads'])

alpha = 0.05
if p_value_true < alpha:
    print(f"ads_converted group is not normally distributed")
else:
    print(f"ads_converted group is normally distributed")
if p_value_false < alpha:
    print(f"ads_not_converted group is not normally distributed")
else:
    print(f"ads_converted group is normally distributed")
```

```
ads_converted group is not normally distributed
ads_not_converted group is not normally distributed
```

Assessing the Relationship Between Conversion Status and Total Ads: Mann-Whitney U Test

```python
import pandas as pd
from scipy.stats import mannwhitneyu

# Perform the Mann-Whitney U test because the groups are not normally distributed
# Conducting the Mann-Whitney U test
u_stat, p_value = mannwhitneyu(ads_converted, ads_not_converted, alternative='two-sided')

print(f"U-statistic: {u_stat}")
print(f"P-value: {p_value}")

# Interpretation of the result
alpha = 0.05  # Significance level
if p_value < alpha:
    print("We reject the null hypothesis (H₀).\nThe number of ad views ('total ads') affects conversion ('converted').")
else:
    print("We cannot reject the null hypothesis (H₀).\nThe number of ad views ('total ads') does not affect conversion ('converted').")

median_converted = ads_converted.median()
median_not_converted = ads_not_converted.median()
q1_converted, q3_converted = ads_converted.quantile([0.25, 0.75])
q1_not_converted, q3_not_converted = ads_not_converted.quantile([0.25, 0.75])

# Recommendation based on median values and quartiles
# print(f"Median of converted = {median_converted}")
if median_converted > median_not_converted:
    print(f"\nIt is recommended to increase the number of ad views to the range {median_converted} - {q3_converted}.")
else:
    print(f"\nIt is recommended to decrease the number of ad views to the range {q1_converted} - {median_converted}.")
```

```
U-statistic: 6788295318.5
P-value: 0.0
We reject the null hypothesis (H₀).
The number of ad views ('total ads') affects conversion ('converted').

It is recommended to increase the number of ad views to the range 64.0 - 103.0.
```

THANK YOU