In [1]:
```python
# inserting neccesary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

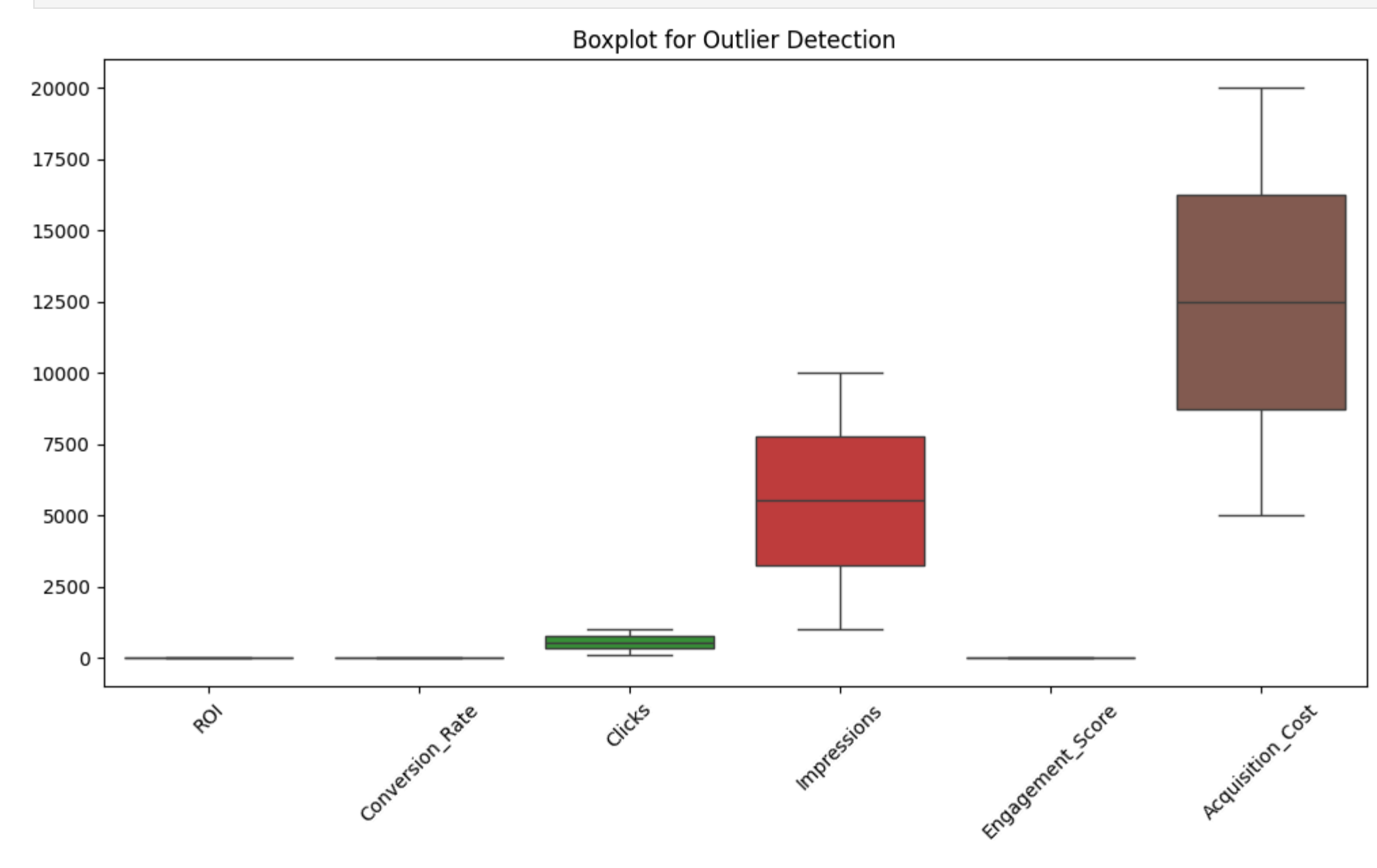
In [2]:
```python
# setting the dataframe
df = pd.read_csv(r"C:\Users\x\Downloads\marketing_campaign.csv")
# printing the first 5 rows
print(df.head())
# printing the information of the dataframe
df.info()
# checking for missing values
print(df.isnull().sum())
# checking for duplicate values
print(df.duplicated().sum())
# describes the dataframe
print(df.describe())
```

```
   Campaign_ID              Company Campaign_Type Target_Audience Duration  \
0            1  Innovate Industries         Email       Men 18-24  30 days
1            2       NexGen Systems         Email     Women 35-44  60 days
2            3    Alpha Innovations    Influencer       Men 25-34  30 days
3            4     DataTech Solutions       Display        All Ages  60 days
4            5       NexGen Systems         Email       Men 25-34  15 days

  Channel_Used  Conversion_Rate Acquisition_Cost   ROI     Location      Date  \
0   Google Ads             0.04       $16,174.00  6.29      Chicago  1/1/2021
1   Google Ads             0.12       $11,566.00  5.61     New York  1/2/2021
2      YouTube             0.07       $10,200.00  7.18  Los Angeles  1/3/2021
3      YouTube             0.11       $12,724.00  5.55        Miami  1/4/2021
4      YouTube             0.05       $16,452.00  6.50  Los Angeles  1/5/2021

   Clicks  Impressions  Engagement_Score     Customer_Segment
0     506         1922                 6     Health & Wellness
1     116         7523                 7          Fashionistas
2     584         7698                 1   Outdoor Adventurers
3     217         1820                 7     Health & Wellness
4     379         4201                 3     Health & Wellness
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200005 entries, 0 to 200004
Data columns (total 15 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   Campaign_ID       200005 non-null  int64
 1   Company           200005 non-null  object
 2   Campaign_Type     200005 non-null  object
 3   Target_Audience   200005 non-null  object
 4   Duration          200005 non-null  object
 5   Channel_Used      200005 non-null  object
 6   Conversion_Rate   200005 non-null  float64
 7   Acquisition_Cost  200005 non-null  object
 8   ROI               200005 non-null  float64
 9   Location          200005 non-null  object
 10  Date              200005 non-null  object
 11  Clicks            200005 non-null  int64
 12  Impressions       200005 non-null  int64
 13  Engagement_Score  200005 non-null  int64
 14  Customer_Segment  200005 non-null  object
dtypes: float64(2), int64(4), object(9)
memory usage: 22.9+ MB
Campaign_ID         0
Company             0
Campaign_Type       0
Target_Audience     0
Duration            0
Channel_Used        0
Conversion_Rate     0
Acquisition_Cost    0
ROI                 0
Location            0
Date                0
Clicks              0
Impressions         0
Engagement_Score    0
Customer_Segment    0
dtype: int64
0
         Campaign_ID  Conversion_Rate            ROI         Clicks  \
count  200005.000000    200005.000000  200005.000000  200005.000000
mean   100003.000000         0.080069       5.002416     549.774591
std     57736.614632         0.040602       1.734485     260.019354
min         1.000000         0.010000       2.000000     100.000000
25%     50002.000000         0.050000       3.500000     325.000000
50%    100003.000000         0.080000       5.010000     550.000000
75%    150004.000000         0.120000       6.510000     775.000000
max    200005.000000         0.150000       8.000000    1000.000000

         Impressions  Engagement_Score
count  200005.000000     200005.000000
mean     5507.307107          5.494673
std      2596.863794          2.872593
min      1000.000000          1.000000
25%      3266.000000          3.000000
50%      5518.000000          5.000000
75%      7753.000000          8.000000
max     10000.000000         10.000000
```

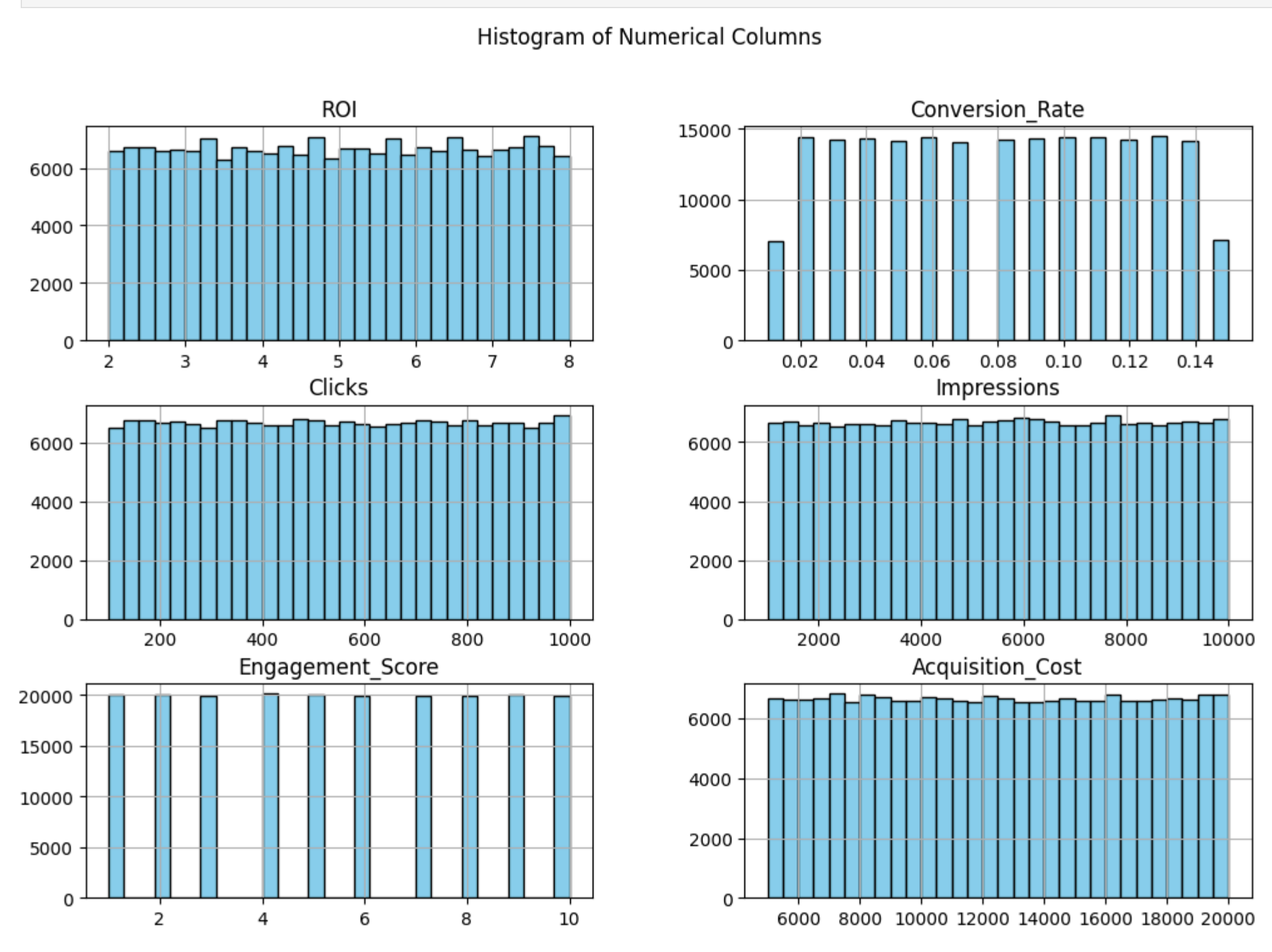
In [3]:
```python
# Remove dollar signs and commas from the 'Acquisition_Cost' column and convert it to float
df["Acquisition_Cost"] = df["Acquisition_Cost"].replace(r'[\$,]', '', regex=True).astype(float)

# List of numerical columns to check for outliers
numbers = ['ROI', 'Conversion_Rate', 'Clicks', 'Impressions', 'Engagement_Score', 'Acquisition_Cost']

# Display the median (50th percentile) of all numerical columns
print(df.select_dtypes(include=['number']).quantile())
```

```
Campaign_ID         100003.00
Conversion_Rate          0.08
Acquisition_Cost     12497.00
ROI                      5.01
Clicks                 550.00
Impressions           5518.00
Engagement_Score         5.00
Name: 0.5, dtype: float64
```

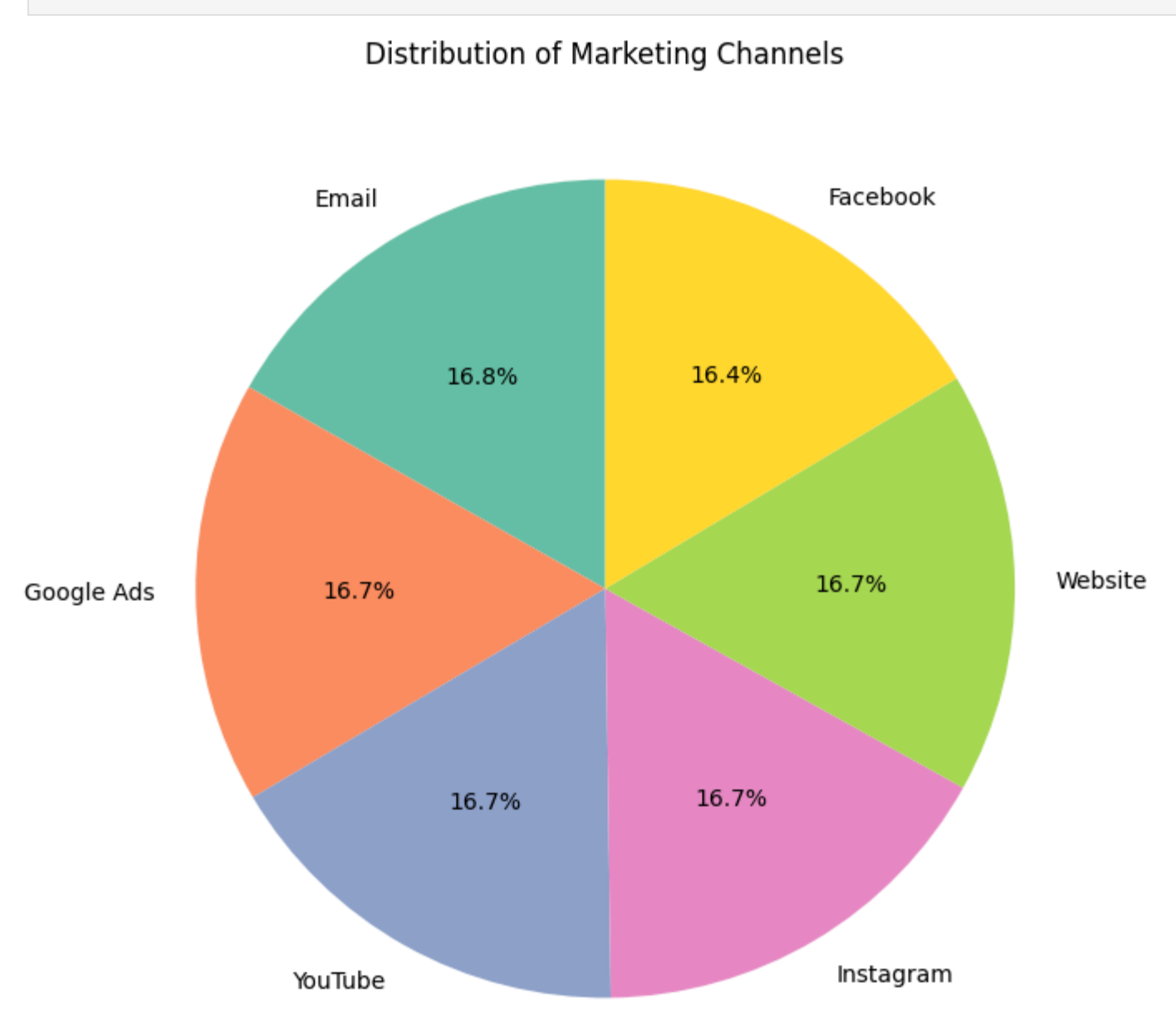
In [4]:
```python
# Plot for boxplots
plt.figure(figsize=(12, 6))
sns.boxplot(data=df[numbers])
plt.title("Boxplot for Outlier Detection")
plt.xticks(rotation=45)

# Display the plot
plt.show()
```



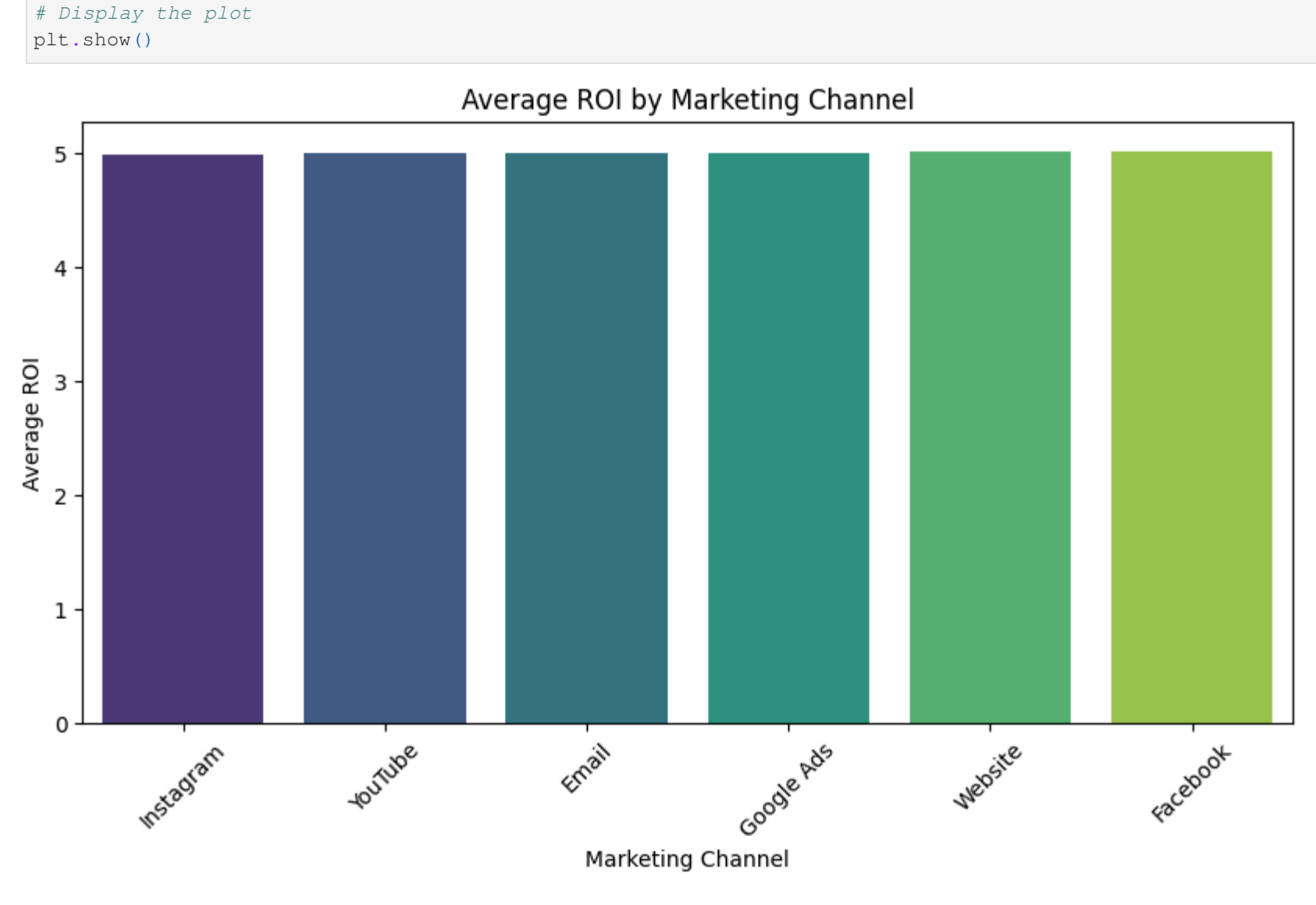
In [5]:
```python
#Plot for histogram
df[numbers].hist(figsize=(12, 8), bins=30, color='skyblue', edgecolor='black')
plt.suptitle("Histogram of Numerical Columns")

# Display the plot
plt.show()
```



In [7]:
```python
# plot for Pie Chart
plt.figure(figsize=(8, 8))
df['Channel_Used'].value_counts().plot.pie(autopct='%1.1f%%', colors=sns.color_palette('Set2'), startangle=90)
plt.title("Distribution of Marketing Channels")
plt.ylabel('')

# Display the plot
plt.show()
```



In [8]:
```python
# Group by marketing channel and calculate the average ROI, then sort the values
channel_performance = df.groupby('Channel_Used')['ROI'].mean().sort_values()

# a bar plot for average ROI by marketing channel
plt.figure(figsize=(10, 5))
sns.barplot(x=channel_performance.index, y=channel_performance.values, hue=channel_performance.index, legend=False, palette='viridis')

# plot title and axis labels
plt.title("Average ROI by Marketing Channel")
plt.xlabel("Marketing Channel")
plt.ylabel("Average ROI")
plt.xticks(rotation=45)

# Display the plot
plt.show()
```



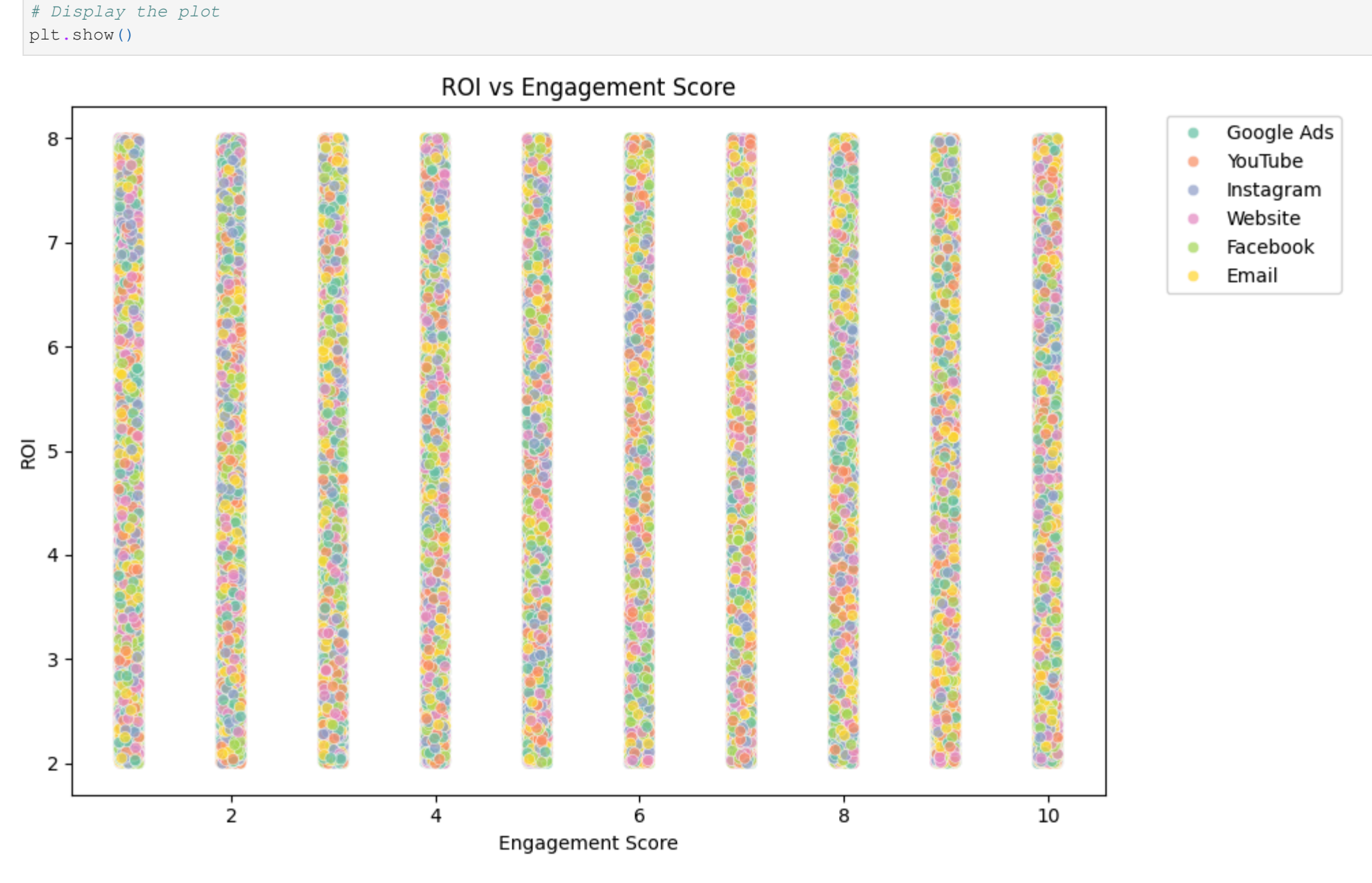
In [9]:
```python
# Calculating Click-Through Rate (CTR) (%)
df['CTR'] = (df['Clicks'] / df['Impressions']) * 100

# Calculating Cost Per Click (CPC)
df["CPC"] = df["Acquisition_Cost"] / df["Clicks"]

# print 5 values of CTR & CPC
print(df['CTR'].head())
print(df['CPC'].head())
```

```
0    26.326743
1     1.541938
2     7.586386
3    11.923077
4     9.021662
Name: CTR, dtype: float64
0    31.964427
1    99.706897
2    17.465753
3    58.635945
4    43.408971
Name: CPC, dtype: float64
```

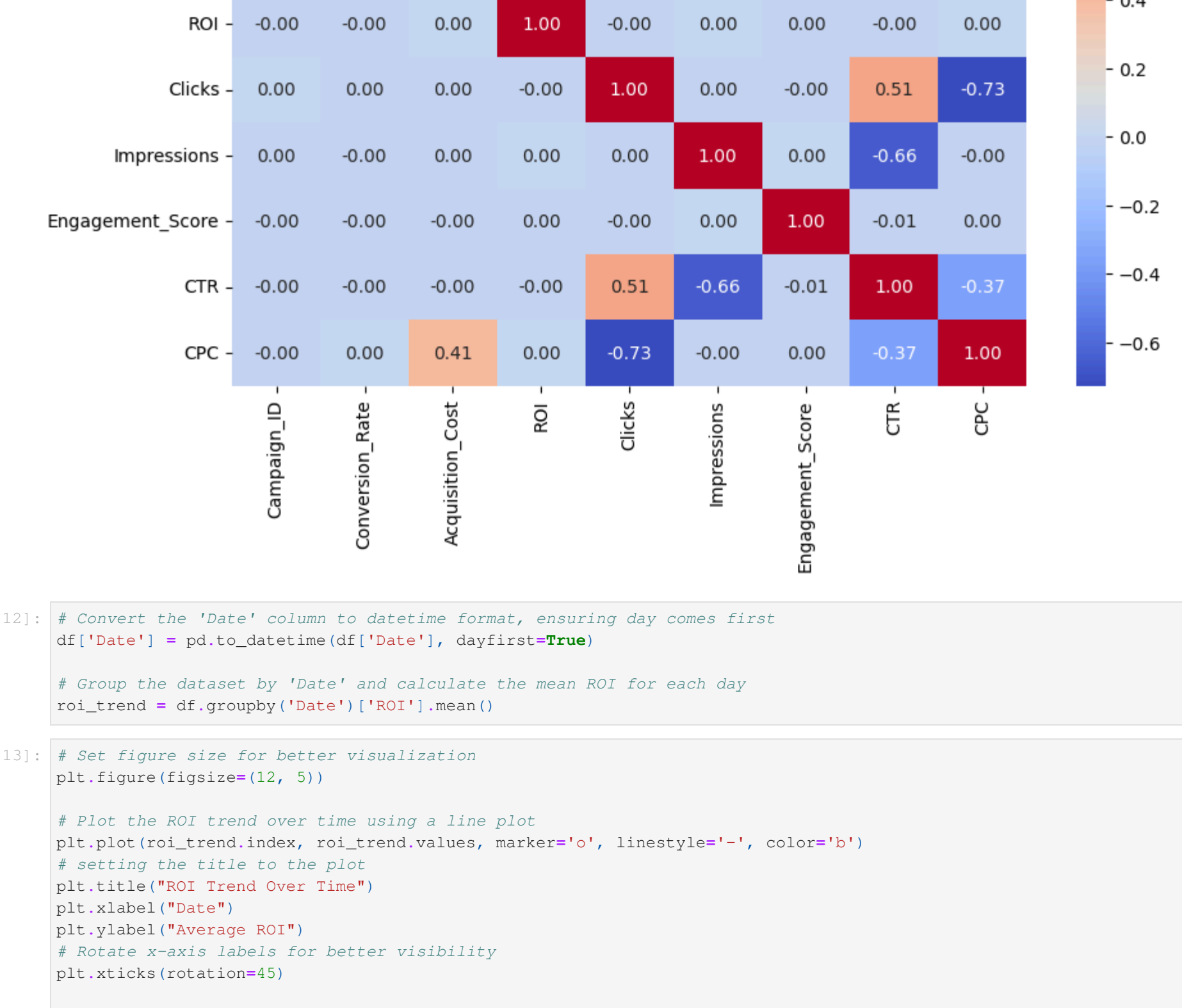
In [ ]:
```python
# print performance first 5 rows & columns
performance = df.groupby('Campaign_ID')[['CTR', 'CPC', 'Conversion_Rate', 'ROI']].mean().reset_index()
print(performance.head())
```

```
   Campaign_ID        CTR        CPC  Conversion_Rate   ROI
0            1  26.326743  31.964427             0.04  6.29
1            2   1.541938  99.706897             0.12  5.61
2            3   7.586386  17.465753             0.07  7.18
3            4  11.923077  58.635945             0.11  5.55
4            5   9.021662  43.408971             0.05  6.50
```

In [10]:
```python
# Set figure size for better visibility
plt.figure(figsize=(10, 6))

# Creating a scatter plot to visualize the relationship between Engagement Score and ROI
sns.scatterplot(
    x=df['Engagement_Score'] + np.random.uniform(-0.1, 0.1, df.shape[0]),
    y=df['ROI'],
    hue=df['Channel_Used'],
    palette='Set2', alpha=0.7
)
# Add plot title and axis labels
plt.title("ROI vs Engagement Score")
plt.xlabel("Engagement Score")
plt.ylabel("ROI")

# Move legend outside to avoid overlap
plt.legend(bbox_to_anchor=(1.05, 1), loc="upper left")
plt.tight_layout()

# Save without background
plt.savefig("ROI_vs_Engagement.png", transparent=True, dpi=300)

# Display the plot
plt.show()
```

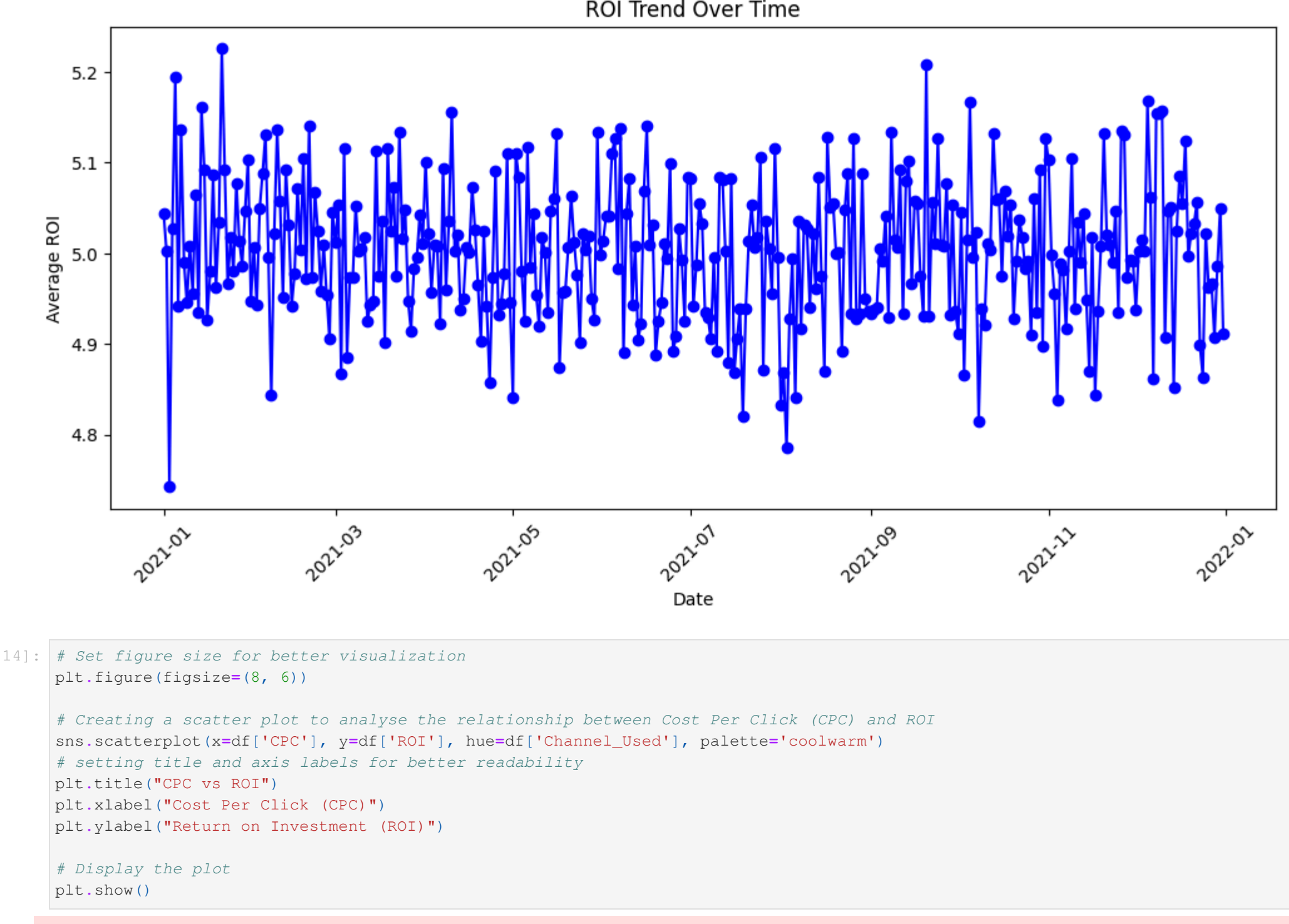


In [11]:
```python
# Set figure size for better readability
plt.figure(figsize=(10, 6))

# Creating a heatmap to visualize the correlation between numerical features
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap")

# Display the plot
plt.show()
```



In [12]:
```python
# Convert the 'Date' column to datetime format, ensuring day comes first
df['Date'] = pd.to_datetime(df['Date'], dayfirst=True)

# Group the dataset by 'Date' and calculate the mean ROI for each day
roi_trend = df.groupby('Date')['ROI'].mean()
```

In [13]:
```python
# Set figure size for better visualization
plt.figure(figsize=(12, 5))

# Plot the ROI trend over time using a line plot
plt.plot(roi_trend.index, roi_trend.values, marker='o', linestyle='-', color='b')
# setting the title to the plot
plt.title("ROI Trend Over Time")
plt.xlabel("Date")
plt.ylabel("Average ROI")
# Rotate x-axis labels for better visibility
plt.xticks(rotation=45)

# Display the plot
plt.show()
```



In [14]:
```python
# Set figure size for better visualization
plt.figure(figsize=(8, 6))

# Creating a scatter plot to analyse the relationship between Cost Per Click (CPC) and ROI
sns.scatterplot(x=df['CPC'], y=df['ROI'], hue=df['Channel_Used'], palette='coolwarm')
# setting title and axis labels for better readability
plt.title("CPC vs ROI")
plt.xlabel("Cost Per Click (CPC)")
plt.ylabel("Return on Investment (ROI)")

# Display the plot
plt.show()
```

```
c:\Users\x\AppData\Local\Programs\Python\Python313\Lib\site-packages\IPython\core\pylabtools.py:170: UserWarning: Creating legend with loc="best" can be slow with large amounts of data.
  fig.canvas.print_figure(bytes_io, **kw)
```

In [15]:
```python
# Define a list of categorical columns to analyze
categorical_cols = ['Company', 'Campaign_Type', 'Target_Audience', 'Channel_Used', 'Location', 'Customer_Segment']

# Loop through each categorical column and display unique values with their counts
for col in categorical_cols:
    print(f"\nUnique values in {col}:")
    print(df[col].value_counts())
```

```
Unique values in Company:
Company
TechCorp               40238
Alpha Innovations      40051
DataTech Solutions     40014
NexGen Systems         39991
Innovate Industries    39711
Name: count, dtype: int64

Unique values in Campaign_Type:
Campaign_Type
Influencer      40170
Search          40157
Display         39988
Email           39871
Social Media    39819
Name: count, dtype: int64

Unique values in Target_Audience:
Target_Audience
Men 18-24      40259
Men 25-34      40024
All Ages       40021
Women 25-34    40013
Women 35-44    39688
Name: count, dtype: int64

Unique values in Channel_Used:
Channel_Used
Email         33599
Google Ads    33440
YouTube       33393
Instagram     33392
Website       33361
Facebook      32820
Name: count, dtype: int64

Unique values in Location:
Location
Miami          40269
New York       40025
Chicago        40013
Los Angeles    39947
Houston        39751
Name: count, dtype: int64

Unique values in Customer_Segment:
Customer_Segment
Foodies                 40210
Tech Enthusiasts        40154
Outdoor Adventurers     40011
```

```
Health & Wellness      39888
Fashionistas           39742
Name: count, dtype: int64
```