EXPERIMENT No – 04

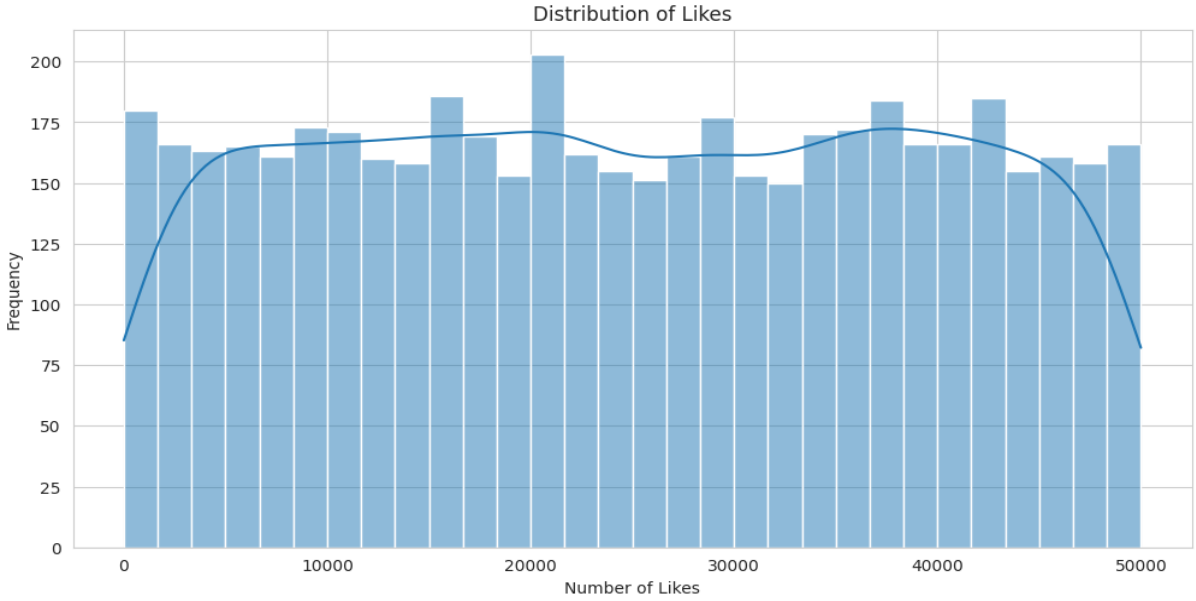
CODE :

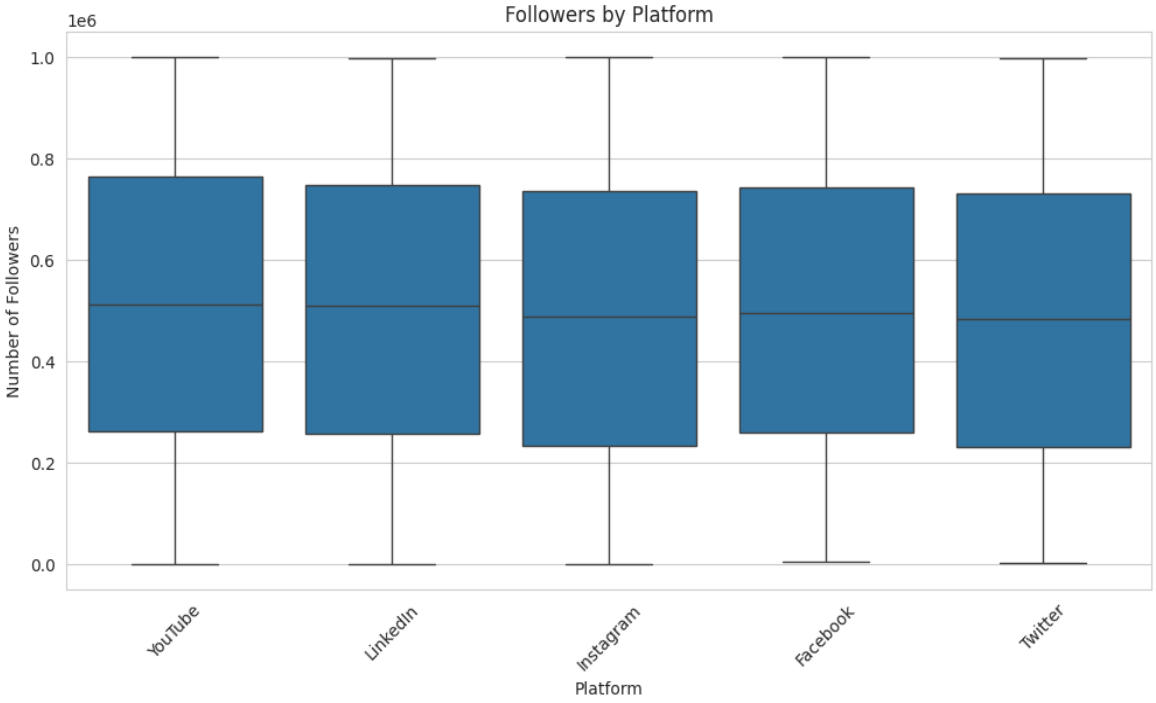
```
import pandas as pd  # Data manipulation and analysis
import numpy as np  # Numerical computing
import matplotlib.pyplot as plt  # Data visualization
import seaborn as sns  # Advanced data visualization

# Load the social media data into a pandas DataFrame
social_media_data = pd.read_csv('social_media_data.csv')

# Check for missing values and handle them
print("Missing values in dataset:")
print(social_media_data.isnull().sum())
social_media_data.fillna(0, inplace=True)  # Replacing missing values with 0

# Summary statistics
print("Summary statistics:")
print(social_media_data.describe())

# Set a consistent plot style
sns.set_style("whitegrid")

# Visualize distribution of likes
plt.figure(figsize=(12,6))
sns.histplot(social_media_data['likes'], kde=True, bins=30)
plt.title('Distribution of Likes')
plt.xlabel('Number of Likes')
plt.ylabel('Frequency')
plt.show()

# Box plot for followers by platform
plt.figure(figsize=(12,6))
sns.boxplot(x='platform', y='followers', data=social_media_data)
plt.title('Followers by Platform')
plt.xlabel('Platform')
plt.ylabel('Number of Followers')
plt.xticks(rotation=45)
plt.show()

# Compute correlation matrix and visualize as heatmap
# Select only numerical columns for correlation calculation
numerical_data = social_media_data.select_dtypes(include=np.number)
corr_matrix = numerical_data.corr()  # Calculate correlation for numerical columns only
plt.figure(figsize=(10,6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix')
```

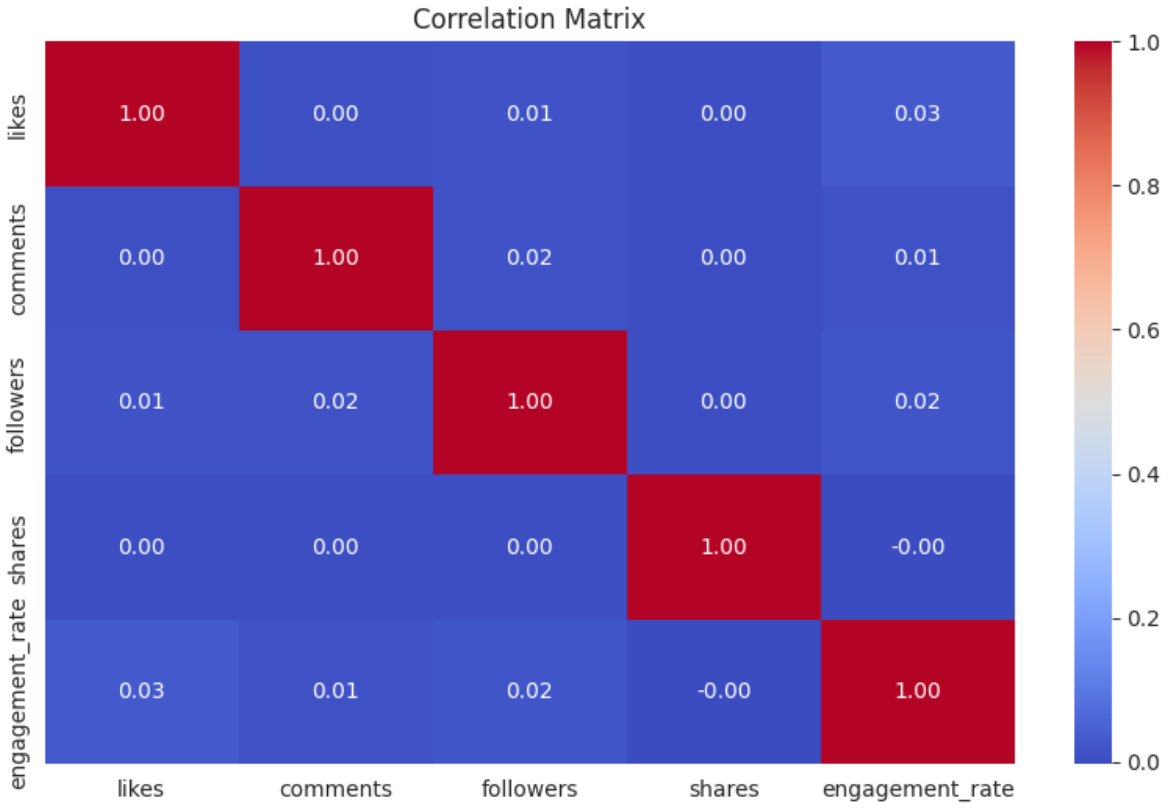
```
plt.show()

# Scatter plot for comments vs engagement rate
plt.figure(figsize=(12,6))
sns.scatterplot(x='comments', y='engagement_rate', data=social_media_data)
plt.title('Comments vs Engagement Rate')
plt.xlabel('Number of Comments')
plt.ylabel('Engagement Rate')
plt.show()

# Box plot for likes by platform
plt.figure(figsize=(12,6))
sns.boxplot(x='platform', y='likes', data=social_media_data)
plt.title('Likes by Platform')
plt.xlabel('Platform')
plt.ylabel('Number of Likes')
plt.xticks(rotation=45)
plt.show()

# Pairplot to visualize relationships between all numerical variables
sns.pairplot(data=social_media_data)
plt.show()
```

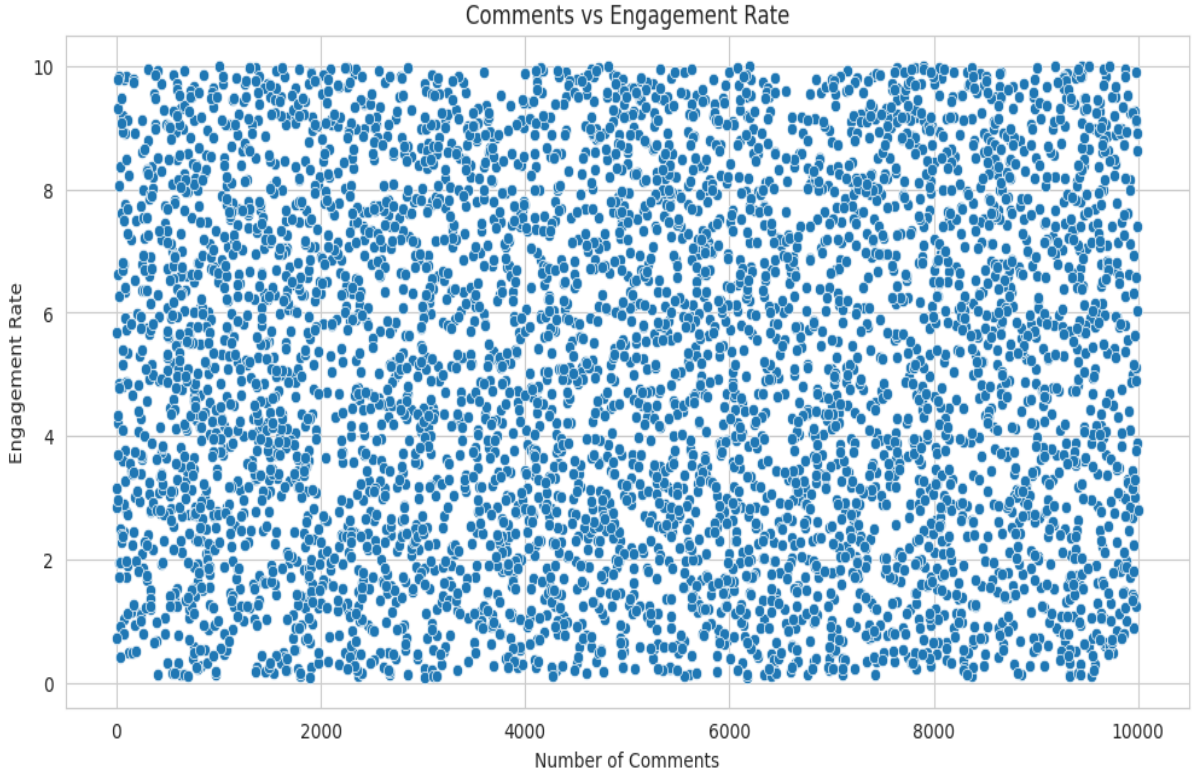
OUTPUT :

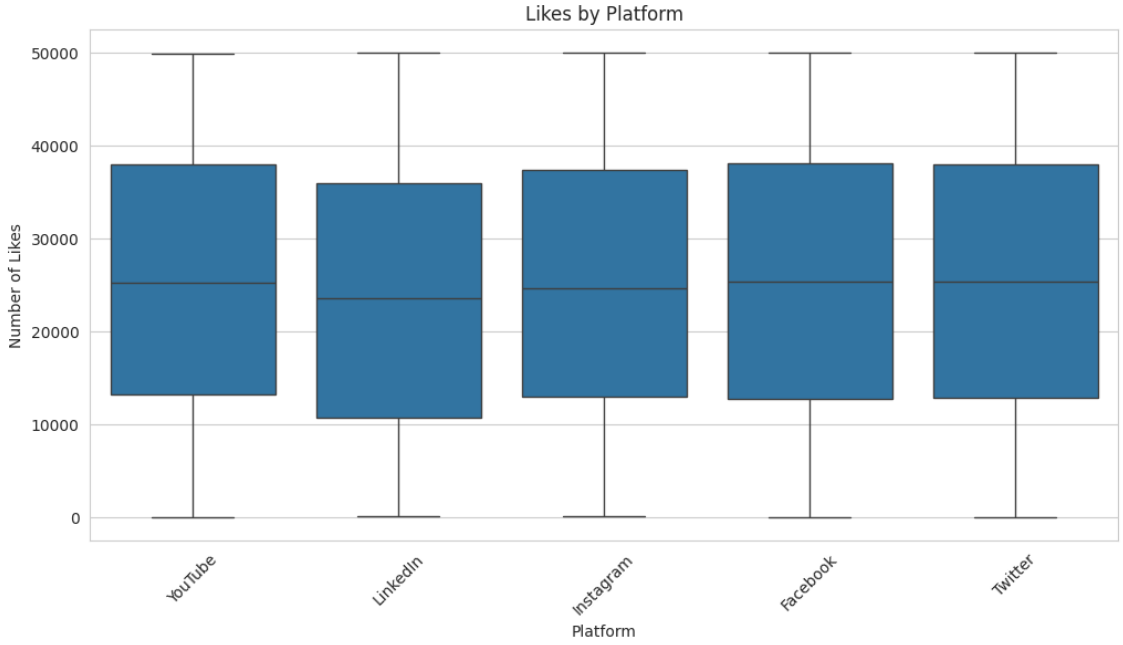
```
Missing values in dataset:
platform          0
likes             0
comments          0
followers         0
shares            0
engagement_rate   0
dtype: int64
Summary statistics:
              likes     comments      followers        shares  engagement_rate
count   5000.000000  5000.00000    5000.000000   5000.000000      5000.000000
mean   24920.195200  4978.69180  497436.485000  10254.957800         5.036538
std    14448.967788  2884.17361  288792.685097   5763.220002         2.862123
min        2.000000     4.00000     715.000000      3.000000         0.100000
25%    12402.250000  2449.75000  246364.250000   5257.000000         2.550000
50%    24713.000000  5007.50000  496087.000000  10481.000000         4.990000
75%    37420.500000  7454.00000  743184.500000  15299.500000         7.530000
max    49999.000000  9998.00000  999902.000000  19999.000000        10.000000
```

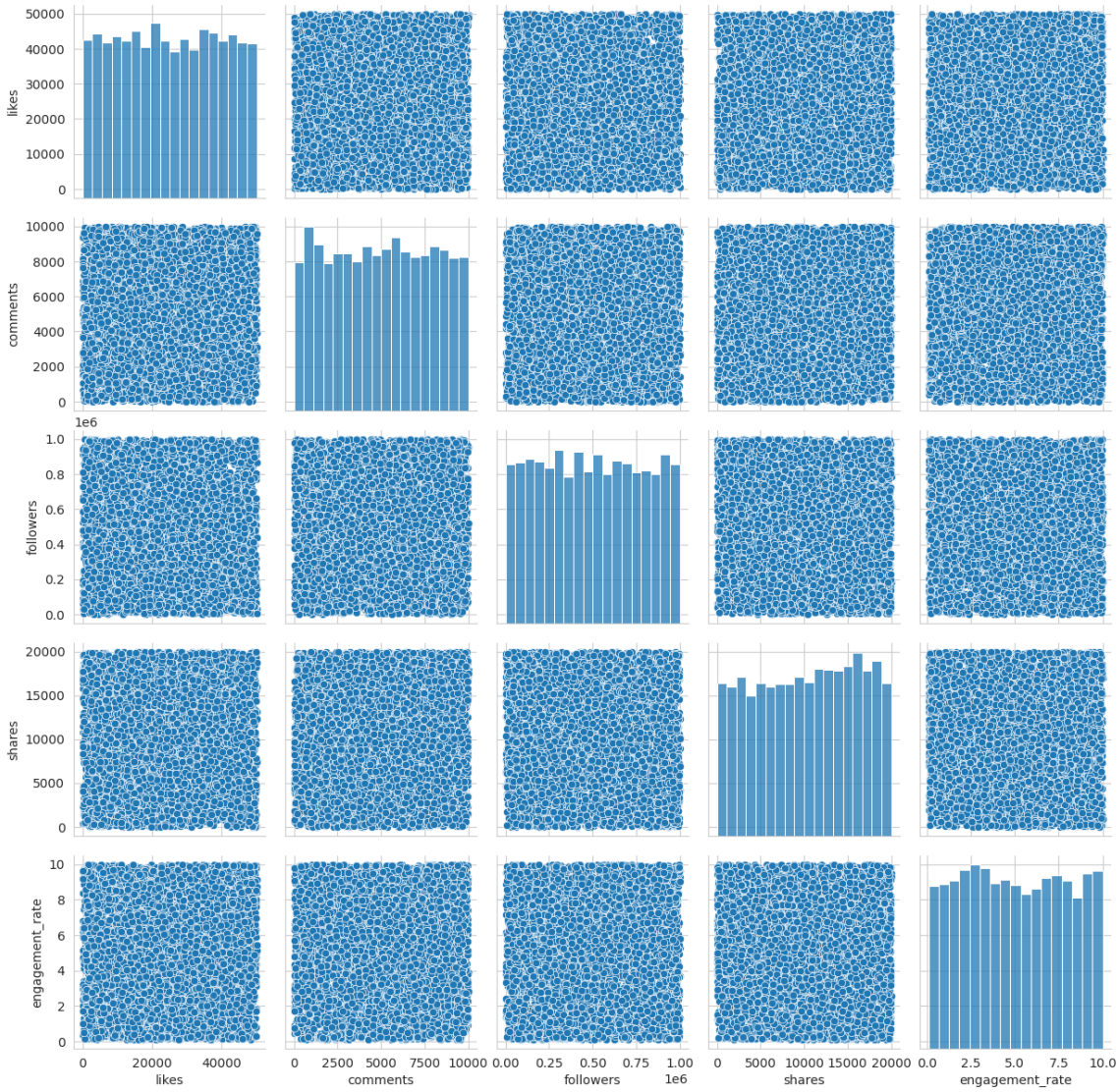

Distribution of Likes
Frequency
Number of Likes
0
25
50
75
100
125
150
175
200
0
10000
20000
30000
40000
50000


Followers by Platform
1e6
Number of Followers
Platform
0.0
0.2
0.4
0.6
0.8
1.0
YouTube
LinkedIn
Instagram
Facebook
Twitter


Correlation Matrix
likes
comments
followers
shares
engagement_rate
1.00
0.00
0.01
0.00
0.03
0.00
1.00
0.02
0.00
0.01
0.01
0.02
1.00
0.00
0.02
0.00
0.00
0.00
1.00
-0.00
0.03
0.01
0.02
-0.00
1.00
1.0
0.8
0.6
0.4
0.2
0.0


Comments vs Engagement Rate
Engagement Rate
Number of Comments
0
2
4
6
8
10
0
2000
4000
6000
8000
10000


Likes by Platform
Number of Likes
50000
40000
30000
20000
10000
0
YouTube
LinkedIn
Instagram
Facebook
Twitter
Platform


likes
comments
followers
1e6
shares
engagement_rate
likes
comments
followers
1e6
shares
engagement_rate