In [1]:
```
import pandas as pd
import random
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
#create categories

categories = ['Food', 'Travel', 'Fashion', 'Fitness', 'Music', 'Culture', 'Fami
```

In [3]:
```
# number of samples

n = 1000
```

In [8]:
```
#Generate the data dictionary

data = {
    'Date': pd.date_range('2021-01-01', periods=n),
    'Category': [random.choice(categories) for _ in range(n)],
    'Likes': np.random.randint(0, 10000, size=n)
}
```

In [13]:
```
# Cleaning, Remove the null data

df.dropna(inplace=True)
```

In [14]:
```
# Remove duplicate data

df.drop_duplicates(inplace=True)
```

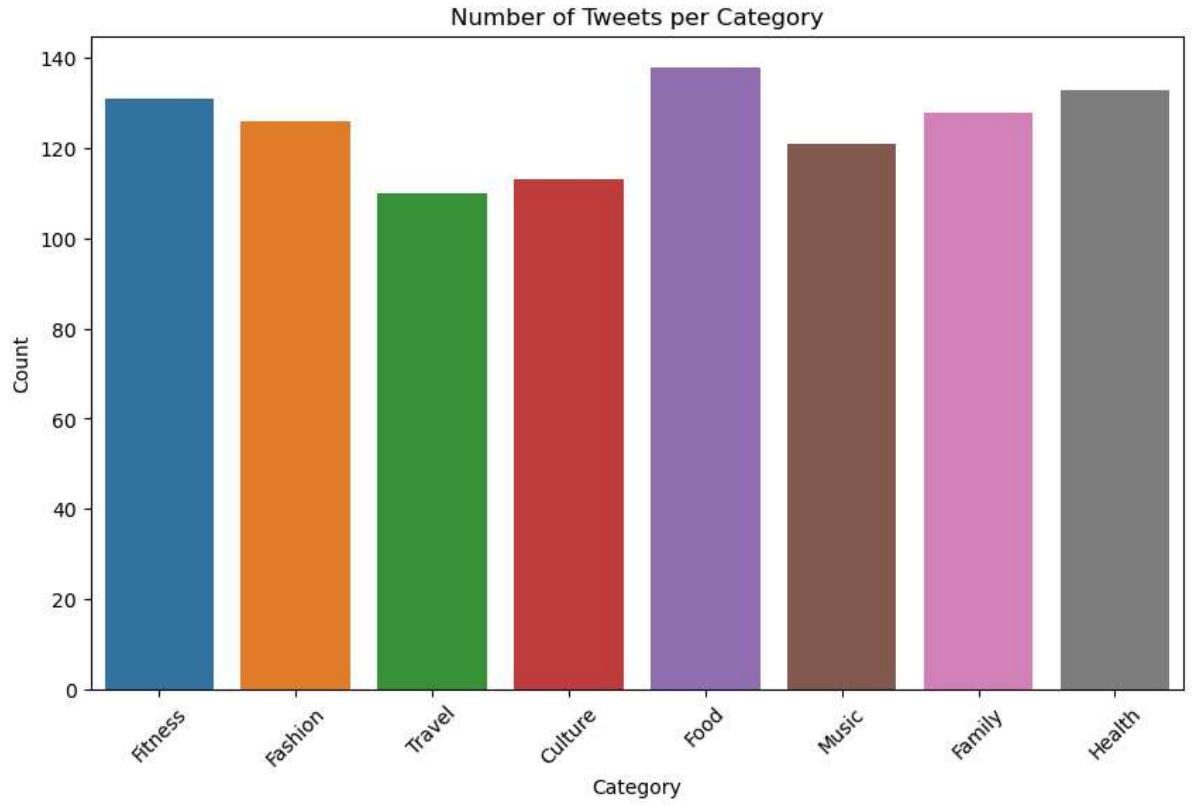
In [15]:
```
# Convert 'Date' to datetime format

df['Date'] = pd.to_datetime(df['Date'])
```

In [16]:
```
# Convert 'Likes' to integer

df['Likes'] = df['Likes'].astype(int)
```

In [17]:
```
print(df.head())
```

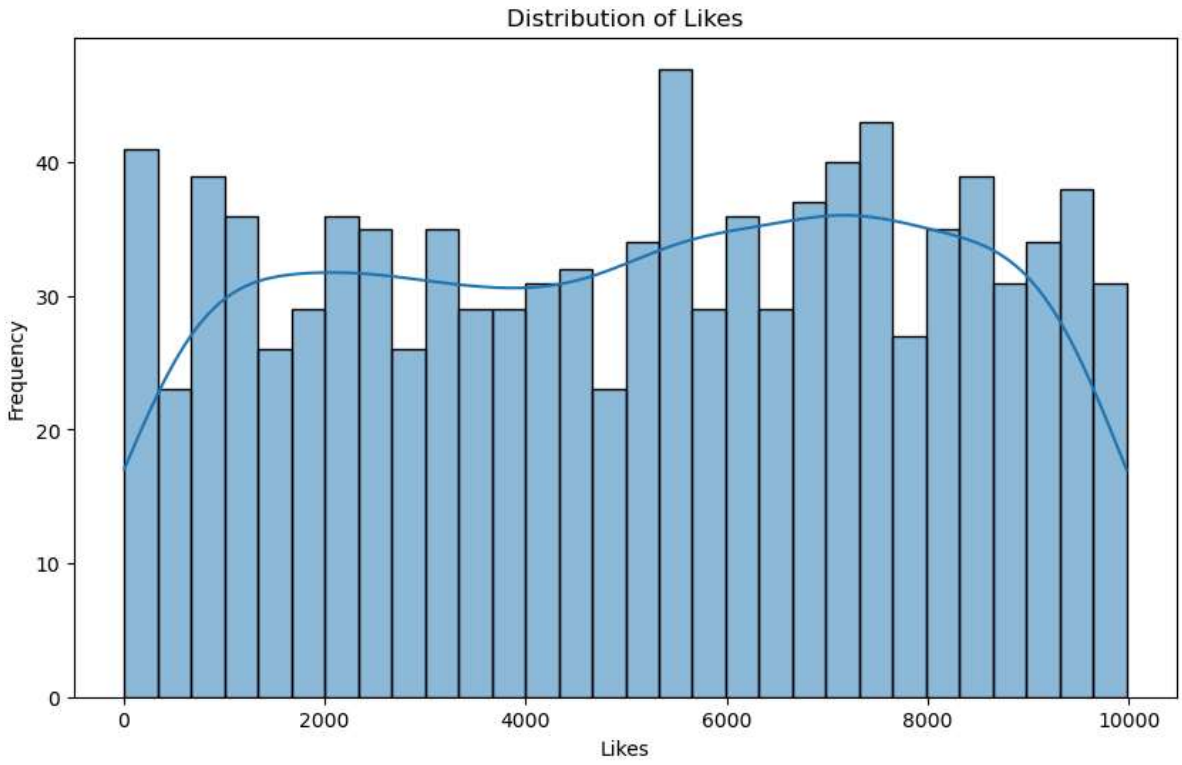
```
        Date Category  Likes
0 2021-01-01  Fitness   5355
1 2021-01-02  Fashion   8240
2 2021-01-03  Fitness   1966
3 2021-01-04   Travel    774
4 2021-01-05  Culture   4681
```

In [18]:
```
# Plot the data

plt.figure(figsize=(10,6))
sns.countplot(data=df, x='Category')
plt.title('Number of Tweets per Category')
plt.xlabel('Category')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

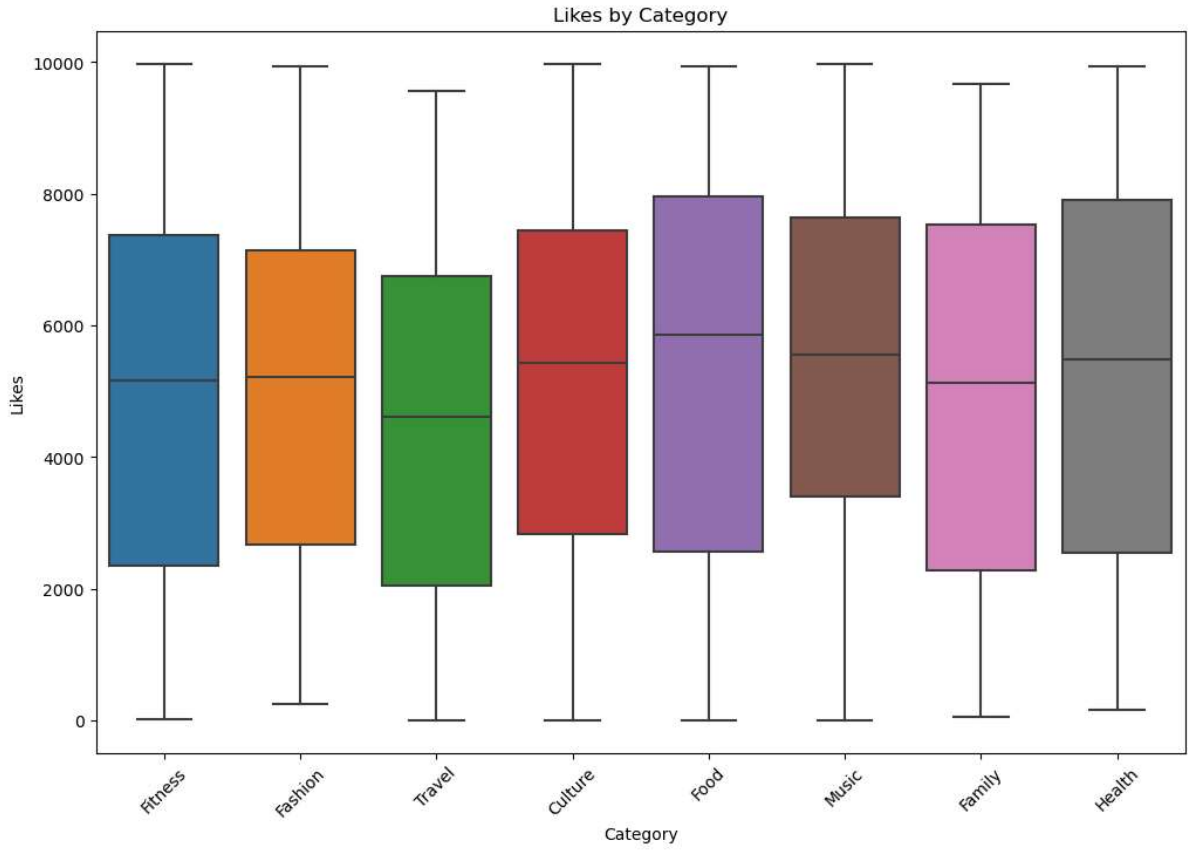


In [20]:
```
# Histogram plot of Likes

plt.figure(figsize=(10, 6))
sns.histplot(df['Likes'], bins=30, kde=True)
plt.title('Distribution of Likes')
plt.xlabel('Likes')
plt.ylabel('Frequency')
plt.show()
```

```
C:\Users\chuck\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWa
rning: use_inf_as_na option is deprecated and will be removed in a future ver
sion. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



In [21]:
```
# Boxplot of Likes by Category

plt.figure(figsize=(12, 8))
sns.boxplot(data=df, x="Category", y ='Likes')
plt.title('Likes by Category')
plt.xlabel('Category')
plt.ylabel('Likes')
plt.xticks(rotation=45)
plt.show()
```



In [23]:
```
#Mean of the 'Likes' category

mean_likes = df['Likes'].mean()
print(f'Mean of Likes: {mean_likes}')
```

```
Mean of Likes: 5077.703
```

In [27]:
```
# Mean of each Category 'Likes'

mean_likes_by_category = df.groupby('Category')['Likes'].mean()
print('Mean of Likes by Category:')
print(mean_likes_by_category)
```

```
Mean of Likes by Category:
Category
Culture    5283.247788
Family     4879.570312
Fashion    5002.666667
Fitness    5018.190840
Food       5303.463768
Health     5242.481203
Music      5398.570248
Travel     4418.518182
Name: Likes, dtype: float64
```

In [ ]: