SocialMediaDataAnalysis

December 27, 2024

1 Clean & Analyze Social Media

1.1 Introduction

Social media has become a ubiquitous part of modern life, with platforms such as Instagram, Twitter, and Facebook serving as essential communication channels. Social media data sets are vast and complex, making analysis a challenging task for businesses and researchers alike. In this project, we explore a simulated social media, for example Tweets, data set to understand trends in likes across different categories.

1.2 Prerequisites

To follow along with this project, you should have a basic understanding of Python programming and data analysis concepts. In addition, you may want to use the following packages in your Python environment:

- pandas
- Matplotlib
- …

These packages should already be installed in Coursera's Jupyter Notebook environment, however if you'd like to install additional packages that are not included in this environment or are working off platform you can install additional packages using `!pip install packagename` within a notebook cell such as:

- `!pip install pandas`
- `!pip install matplotlib`

1.3 Project Scope

The objective of this project is to analyze tweets (or other social media data) and gain insights into user engagement. We will explore the data set using visualization techniques to understand the distribution of likes across different categories. Finally, we will analyze the data to draw conclusions about the most popular categories and the overall engagement on the platform.

1.4 Step 1: Importing Required Libraries

As the name suggests, the first step is to import all the necessary libraries that will be used in the project. In this case, we need pandas, numpy, matplotlib, seaborn, and random libraries.

Pandas is a library used for data manipulation and analysis. Numpy is a library used for numerical computations. Matplotlib is a library used for data visualization. Seaborn is a library used for statistical data visualization. Random is a library used to generate random numbers.

[2]:
```
'/opt/conda/bin/python3 -m pip install --upgrade pip'
```

[2]:
```
'/opt/conda/bin/python3 -m pip install --upgrade pip'
```

[10]:
```
import pandas as pd
import random
import numpy as np

# Define the list of categories
categories = ['Food', 'Travel', 'Fashion', 'Fitness', 'Music', 'Culture',␣
 ↪'Family', 'Health']

# Generate random data with 500 entries
n = 500

# Create a dictionary to store the data
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)
}

# Create a pandas DataFrame from the dictionary
df = pd.DataFrame(data)

print(df.head())
```

```
        Date Category  Likes
0 2021-01-01  Fitness   2218
1 2021-01-02   Travel   6563
2 2021-01-03   Health    307
3 2021-01-04   Health   6653
4 2021-01-05  Fashion   1536
```

[11]:
```
df.describe()
#THis command ignores non numerial values
```

[11]:
```
              Likes
count   500.000000
```

```
mean   4960.546000
std    2750.711193
min       0.000000
25%    2680.000000
50%    5076.500000
75%    7191.500000
max    9981.000000
```

[12]:
```
df.describe(include='all')
```

[12]:
```
                       Date Category        Likes
count                   500      500   500.000000
unique                  500        8          NaN
top     2021-06-04 00:00:00    Music          NaN
freq                      1       71          NaN
first   2021-01-01 00:00:00      NaN          NaN
last    2022-05-15 00:00:00      NaN          NaN
mean                    NaN      NaN  4960.546000
std                     NaN      NaN  2750.711193
min                     NaN      NaN     0.000000
25%                     NaN      NaN  2680.000000
50%                     NaN      NaN  5076.500000
75%                     NaN      NaN  7191.500000
max                     NaN      NaN  9981.000000
```

[13]:
```
df.dtypes
#to check the data types in dataset
```

[13]:
```
Date        datetime64[ns]
Category            object
Likes                int64
dtype: object
```

[14]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Date      500 non-null    datetime64[ns]
 1   Category  500 non-null    object
 2   Likes     500 non-null    int64
dtypes: datetime64[ns](1), int64(1), object(1)
memory usage: 11.8+ KB
```

[15]:
```
df.loc[df['Category'] == "Health"]
```

[15]:

	Date	Category	Likes
2	2021-01-03	Health	307
3	2021-01-04	Health	6653
10	2021-01-11	Health	5686
35	2021-02-05	Health	332
47	2021-02-17	Health	4628
48	2021-02-18	Health	8074
59	2021-03-01	Health	915
65	2021-03-07	Health	2242
70	2021-03-12	Health	566
79	2021-03-21	Health	8383
89	2021-03-31	Health	8623
90	2021-04-01	Health	3586
108	2021-04-19	Health	2018
113	2021-04-24	Health	3231
122	2021-05-03	Health	5993
123	2021-05-04	Health	2566
124	2021-05-05	Health	3538
150	2021-05-31	Health	8875
152	2021-06-02	Health	5677
160	2021-06-10	Health	2896
173	2021-06-23	Health	8808
184	2021-07-04	Health	1491
186	2021-07-06	Health	7027
193	2021-07-13	Health	2152
200	2021-07-20	Health	3204
201	2021-07-21	Health	1115
207	2021-07-27	Health	9919
211	2021-07-31	Health	9981
214	2021-08-03	Health	6578
225	2021-08-14	Health	8334
226	2021-08-15	Health	3979
227	2021-08-16	Health	2106
232	2021-08-21	Health	1465
250	2021-09-08	Health	7388
257	2021-09-15	Health	1433
265	2021-09-23	Health	2586
276	2021-10-04	Health	2927
289	2021-10-17	Health	4511
303	2021-10-31	Health	4067
304	2021-11-01	Health	8468
311	2021-11-08	Health	4079
318	2021-11-15	Health	4615
342	2021-12-09	Health	3847
344	2021-12-11	Health	9547
360	2021-12-27	Health	6408
362	2021-12-29	Health	3975

```
405 2022-02-10   Health   2762
425 2022-03-02   Health   7481
427 2022-03-04   Health    457
430 2022-03-07   Health   3273
440 2022-03-17   Health   6587
441 2022-03-18   Health     93
448 2022-03-25   Health   4675
449 2022-03-26   Health   4682
453 2022-03-30   Health   3459
458 2022-04-04   Health   2188
481 2022-04-27   Health   7429
488 2022-05-04   Health   2786
489 2022-05-05   Health   2644
490 2022-05-06   Health    425
```

[16]:
```
df.isnull().sum()
#to check missing values.
```

[16]:
```
Date        0
Category    0
Likes       0
dtype: int64
```

[17]:
```
unique_categories = df['Category'].unique()
print(unique_categories)
```

```
['Fitness' 'Travel' 'Health' 'Fashion' 'Music' 'Family' 'Culture' 'Food']
```

[18]:
```
category_counts = df['Category'].value_counts()
print(category_counts)
#This code is to determine the occurrence of each category
```

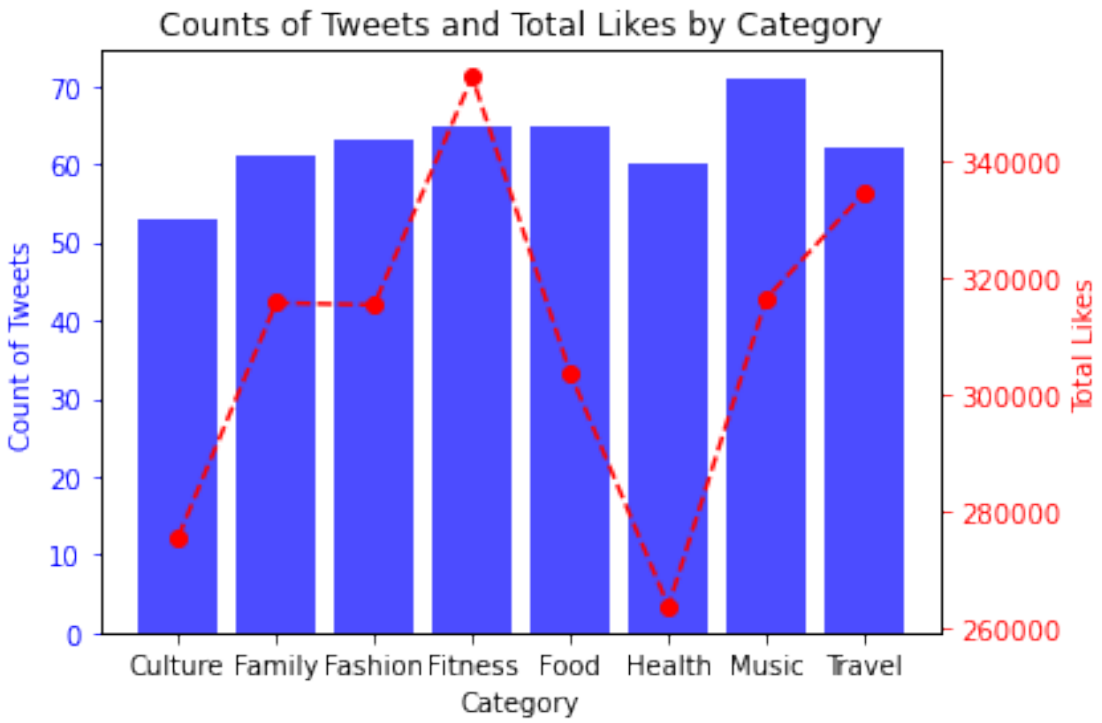
```
Music      71
Food       65
Fitness    65
Fashion    63
Travel     62
Family     61
Health     60
Culture    53
Name: Category, dtype: int64
```

[21]:
```
# Group by 'Category' and aggregate the counts and sum of likes
category_stats = df.groupby('Category').agg(
    count_of_tweets=('Category', 'count'),
    total_likes=('Likes', 'sum')
).reset_index()
```

```
# Display the result
print(category_stats)
```

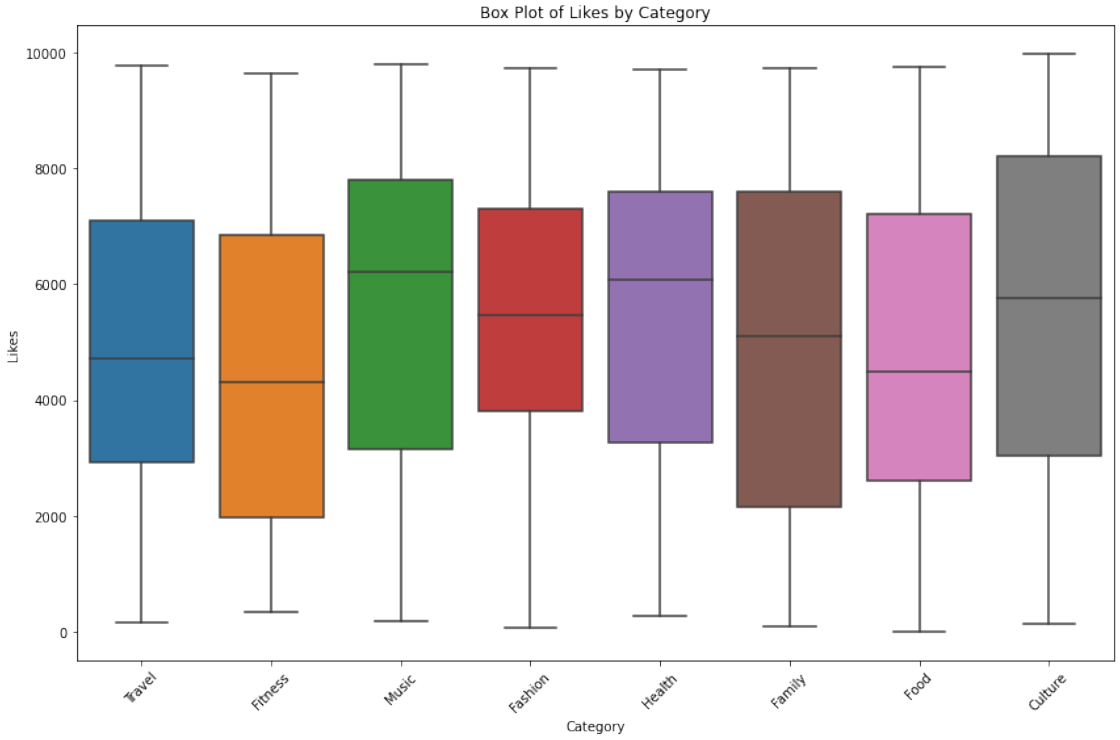
```
  Category  count_of_tweets  total_likes
0  Culture               53       275545
1   Family               61       315823
2  Fashion               63       315459
3  Fitness               65       354603
4     Food               65       303716
5   Health               60       263740
6    Music               71       316607
7   Travel               62       334780
```

[23]:
```
import matplotlib.pyplot as plt

# Plotting the bar chart
fig, ax1 = plt.subplots()

# Bar chart for count of tweets
ax1.bar(category_stats['Category'], category_stats['count_of_tweets'],␣
↪color='b', alpha=0.7)
ax1.set_xlabel('Category')
ax1.set_ylabel('Count of Tweets', color='b')
ax1.tick_params('y', colors='b')

# Create a second y-axis for the likes data
ax2 = ax1.twinx()
ax2.plot(category_stats['Category'], category_stats['total_likes'], color='r',␣
↪marker='o', linestyle='dashed')
ax2.set_ylabel('Total Likes', color='r')
ax2.tick_params('y', colors='r')

plt.title('Counts of Tweets and Total Likes by Category')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

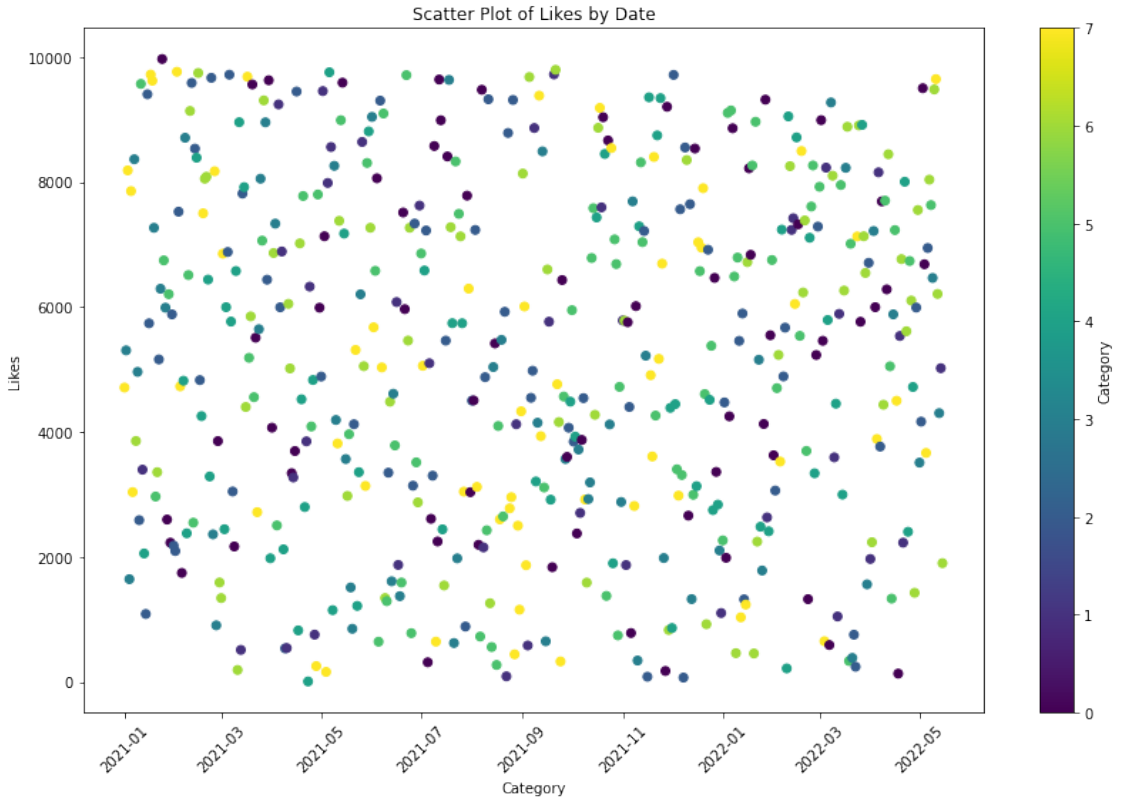


[ ]:
```
# Based on the barplot above,fitness has most likes with 65 tweets,travel is␣
 ↪second most likes with only 62 tweets
#compared to music with highest tweets but lesser likes.However the most␣
 ↪engaged category based on likes is fitness,followed by
#Travel category,third is Music and Family.
```

[115]:
```
plt.figure(figsize=(12, 8))
sns.boxplot(x='Category', y='Likes', data=df)
plt.title('Box Plot of Likes by Category')
plt.xlabel('Category')
plt.ylabel('Likes')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

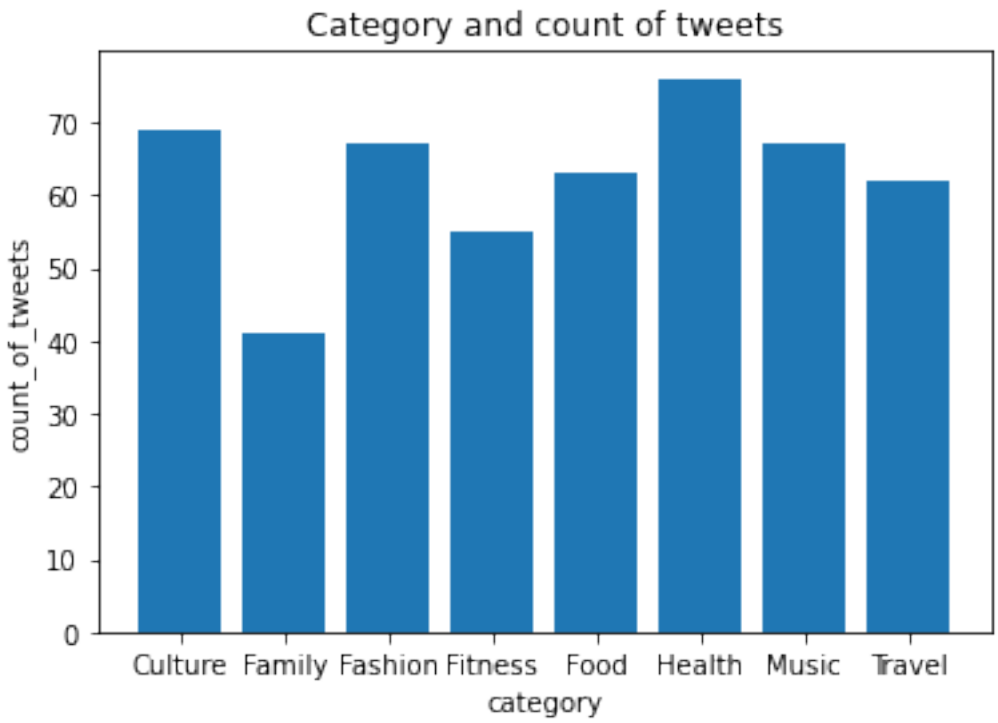


[118]:
```
plt.figure(figsize=(12, 8))
plt.scatter(df['Date'], df['Likes'], c=df['Category'].astype('category').cat.
 ↪codes, cmap='viridis')
plt.colorbar(label='Category')
plt.title('Scatter Plot of Likes by Date')
plt.xlabel('Date')
plt.ylabel('Likes')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



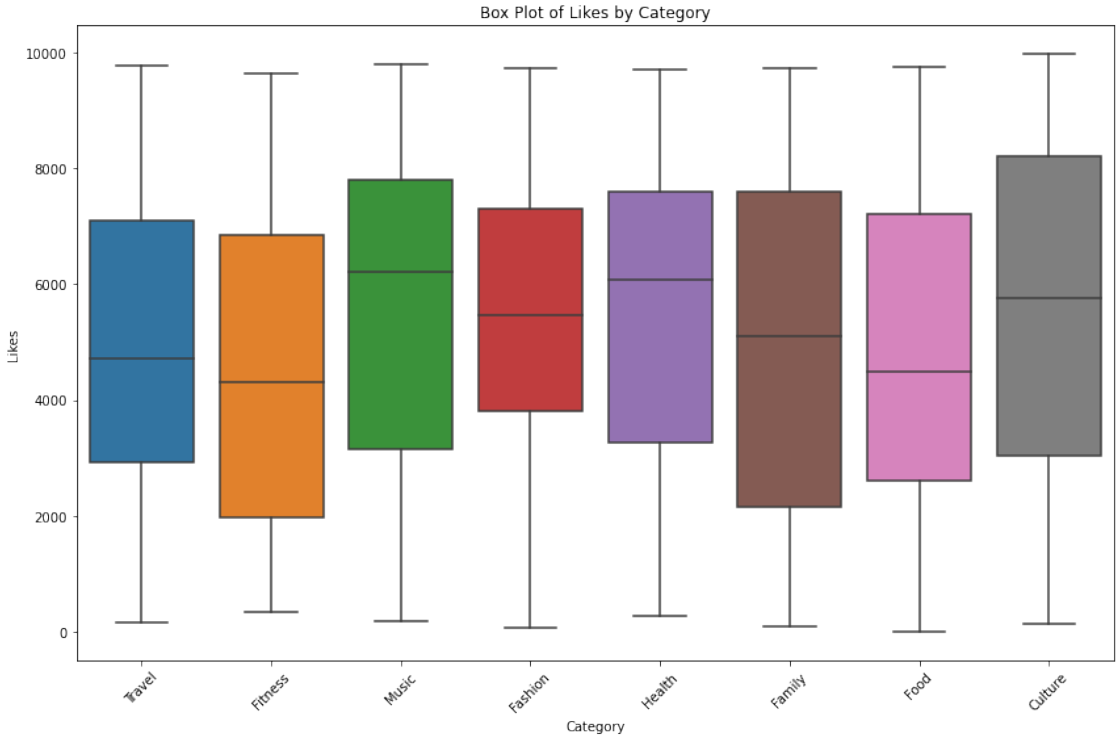
[ ]:
```
# This scatter plot shows a very weak relationship between category,likes and␣
↪date.All variables are independent of each other.
```

[28]:
```
Category␣
↪=['Culture','Family','Fashion','Fitness','Food','Health','Music','Travel']
count_of_tweets =[69,41,67,55,63,76,67,62]
plt.bar(Category,count_of_tweets)
plt.xlabel('category')
plt.ylabel('count_of_tweets')
plt.title('Category and count of tweets')
plt.show()
```



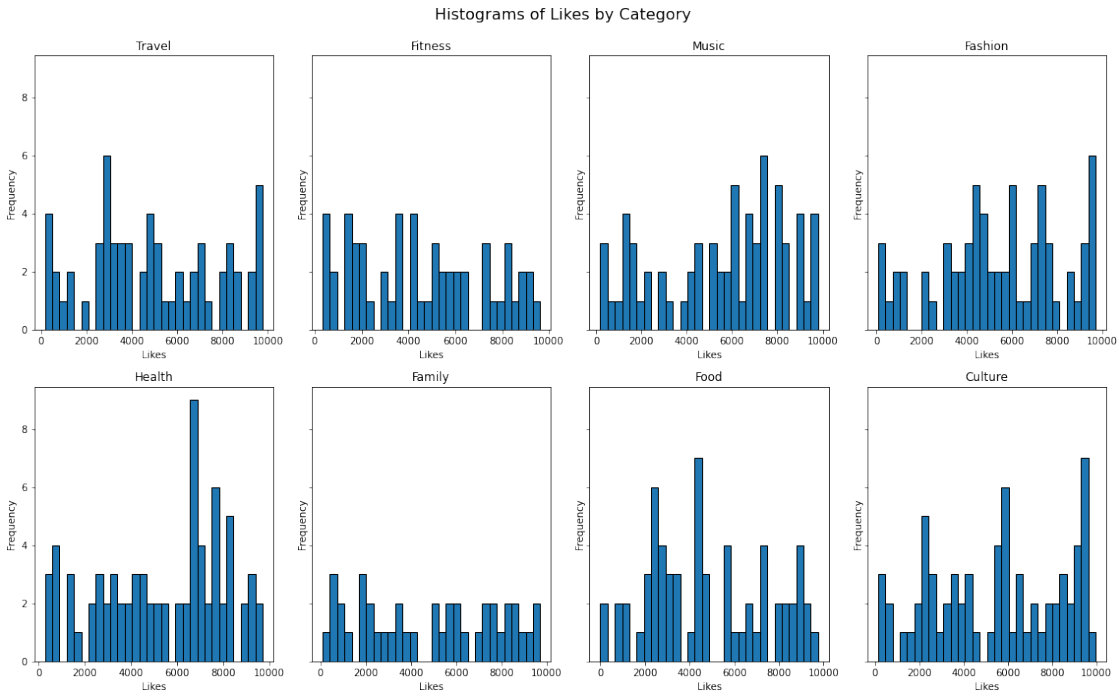
[ ]:
```
#Health is the most engaged category based on number of tweets,followed by␣
↪culture and fashion.
```

[121]:
```
plt.figure(figsize=(12, 8))
sns.boxplot(x='Category', y='Likes', data=df)
plt.title('Box Plot of Likes by Category')
plt.xlabel('Category')
plt.ylabel('Likes')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



[ ]:
```
#No outlier found in category and likes.
```

[126]:
```
categories = df['Category'].unique()

# Set up the figure and axes
fig, axes = plt.subplots(nrows=2, ncols=4, figsize=(16, 10), sharey=True)
axes = axes.flatten()

# Plot histograms for each category
for i, category in enumerate(categories):
    ax = axes[i]
    subset = df[df['Category'] == category]
    ax.hist(subset['Likes'], bins=30, edgecolor='black')
    ax.set_title(category)
    ax.set_xlabel('Likes')
    ax.set_ylabel('Frequency')

# Add a main title
fig.suptitle('Histograms of Likes by Category', fontsize=16)
plt.tight_layout(rect=[0, 0, 1, 0.95])  # Adjust layout to fit the main title

plt.show()
```



[57]:
```
df.loc[df['Category'] == "Health"].mean()
```

[57]:
```
total_like    5342.289474
dtype: float64
```

[60]:
```
round(5342.289474,0)
```

[60]:
```
5342.0
```

[58]:
```
df.loc[df['Category'] == "Family"].mean()
```

[58]:
```
total_like    4861.926829
dtype: float64
```

[68]:
```
round(4861.926829,0)
```

[68]:
```
4862.0
```

[63]:
```
df.loc[df['Category'] == "Fitness"].mean()
```

[63]:
```
total_like    4577.672727
dtype: float64
```

[64]:
```
round(4577.672727,0)
```

[64]: 4578.0

[66]:
```
df.loc[df['Category'] == "Food"].mean()
```

[66]:
```
total_like    4857.84127
dtype: float64
```

[67]:
```
round(4857.84127,0)
```

[67]: 4858.0

[69]:
```
df.loc[df['Category'] == "Culture"].mean()
```

[69]:
```
total_like    5481.101449
dtype: float64
```

[70]:
```
round(5481.101449,0)
```

[70]: 5481.0

[71]:
```
df.loc[df['Category'] == "Music"].mean()
```

[71]:
```
total_like    5538.253731
dtype: float64
```

[72]:
```
round(5538.253731,0)
```

[72]: 5538.0

[73]:
```
df.loc[df['Category'] == "Travel"].mean()
```

[73]:
```
total_like    4902.774194
dtype: float64
```

[74]:
```
round(4902.774194,0)
```

[74]: 4903.0

[75]:
```
df.loc[df['Category'] == "Fashion"].mean()
```

[75]:
```
total_like    5429.38806
dtype: float64
```

[76]:
```
round(5429.38806,0)
```

[76]: 5429.0

[ ]:
```
#Based on the average total likes of category,music has the most likes,followed␣
 ↪by Culture and Fashion with
#very little difference.Health has the fourth position of average total␣
 ↪likes,followed by Travel, Family,Food
#and fitness has least total average likes.
```

[ ]:
```
# To conclude from the above charts and statistical analysis,the number of␣
 ↪tweets does not guarantee or determine the likes
#of the tweet irrespective of the date.Therefore,I recommend for example heath␣
 ↪has the highest tweets but lesser likes compare
#to other categories,could be improve by focus on interesting topics or general␣
 ↪presentation of the tweet like design and colors.
```