SocialMediaDataAnalysis

December 16, 2023

1 Social Media Data Analysis

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

[2]:
```
#list of categories for social media experiment

categories = ['Food', 'Travel', 'Fashion', 'Fitness', 'Music', 'Culture',␣
↪'Family', 'Health']
categories
```

[2]:
```
['Food',
 'Travel',
 'Fashion',
 'Fitness',
 'Music',
 'Culture',
 'Family',
 'Health']
```

[3]:
```
n = 500
n
```

[3]:
```
500
```

[4]:
```
#random data dictionary
data = {
    'Date': pd.date_range('2021-01-01', periods=n),
    'Category': [random.choice(categories) for i in range(n)],
    'Likes': np.random.randint(0, 10000, size=n)
}
```

[5]:
```
data
```

[5]:
```
{'Date': DatetimeIndex(['2021-01-01', '2021-01-02', '2021-01-03', '2021-01-04',
                        '2021-01-05', '2021-01-06', '2021-01-07', '2021-01-08',
                        '2021-01-09', '2021-01-10',
                        …
                        '2022-05-06', '2022-05-07', '2022-05-08', '2022-05-09',
                        '2022-05-10', '2022-05-11', '2022-05-12', '2022-05-13',
                        '2022-05-14', '2022-05-15'],
                       dtype='datetime64[ns]', length=500, freq='D'),
 'Category': ['Fitness',
              'Travel',
              'Food',
              'Food',
              'Fitness',
              'Travel',
              'Fitness',
              'Music',
              'Health',
              'Health',
              'Family',
              'Music',
              'Health',
              'Culture',
              'Health',
              'Travel',
              'Fitness',
              'Music',
              'Family',
              'Family',
              'Fashion',
              'Food',
              'Music',
              'Culture',
              'Health',
```

```
'Travel',
'Family',
'Fashion',
'Family',
'Music',
'Culture',
'Fashion',
'Health',
'Fitness',
'Fashion',
'Culture',
'Fitness',
'Travel',
'Fitness',
'Family',
'Culture',
'Fitness',
'Fitness',
'Music',
'Fitness',
'Travel',
'Family',
'Travel',
'Food',
'Fashion',
'Health',
'Health',
'Health',
'Culture',
'Health',
'Family',
'Family',
'Travel',
'Music',
'Fashion',
'Culture',
'Food',
'Family',
'Fitness',
'Food',
'Culture',
'Health',
'Travel',
'Fashion',
'Travel',
'Music',
'Travel',
```

```
'Music',
'Fashion',
'Travel',
'Music',
'Fitness',
'Family',
'Family',
'Food',
'Fashion',
'Culture',
'Family',
'Fitness',
'Family',
'Health',
'Fitness',
'Culture',
'Culture',
'Culture',
'Food',
'Travel',
'Family',
'Fitness',
'Fashion',
'Culture',
'Family',
'Travel',
'Family',
'Culture',
'Fashion',
'Music',
'Family',
'Fitness',
'Music',
'Fitness',
'Culture',
'Food',
'Music',
'Food',
'Fashion',
'Music',
'Fitness',
'Music',
'Culture',
'Fashion',
'Health',
'Music',
'Family',
```

```
'Travel',
'Music',
'Health',
'Food',
'Health',
'Music',
'Family',
'Food',
'Travel',
'Culture',
'Fashion',
'Music',
'Food',
'Fashion',
'Family',
'Culture',
'Health',
'Travel',
'Health',
'Fitness',
'Health',
'Health',
'Fitness',
'Fitness',
'Health',
'Travel',
'Travel',
'Culture',
'Fitness',
'Music',
'Fashion',
'Health',
'Fashion',
'Fitness',
'Fashion',
'Food',
'Culture',
'Family',
'Fashion',
'Family',
'Family',
'Family',
'Family',
'Health',
'Health',
'Culture',
'Culture',
```

```
'Music',
'Travel',
'Fitness',
'Music',
'Fashion',
'Fashion',
'Fitness',
'Music',
'Fitness',
'Health',
'Food',
'Culture',
'Fitness',
'Fitness',
'Family',
'Food',
'Health',
'Fashion',
'Fashion',
'Fashion',
'Family',
'Food',
'Music',
'Fitness',
'Music',
'Culture',
'Travel',
'Culture',
'Fitness',
'Fashion',
'Food',
'Fashion',
'Travel',
'Culture',
'Music',
'Travel',
'Health',
'Health',
'Travel',
'Travel',
'Travel',
'Family',
'Culture',
'Family',
'Health',
'Music',
'Family',
```

```
'Travel',
'Music',
'Travel',
'Food',
'Travel',
'Travel',
'Fitness',
'Health',
'Music',
'Travel',
'Fashion',
'Culture',
'Fitness',
'Travel',
'Culture',
'Fashion',
'Health',
'Fashion',
'Culture',
'Music',
'Health',
'Culture',
'Food',
'Health',
'Fashion',
'Family',
'Food',
'Fashion',
'Food',
'Food',
'Culture',
'Travel',
'Music',
'Travel',
'Family',
'Music',
'Fitness',
'Fashion',
'Family',
'Travel',
'Health',
'Culture',
'Music',
'Music',
'Fashion',
'Culture',
'Fitness',
```

```
'Culture',
'Food',
'Food',
'Health',
'Food',
'Travel',
'Culture',
'Health',
'Health',
'Music',
'Music',
'Health',
'Family',
'Travel',
'Fitness',
'Travel',
'Music',
'Travel',
'Family',
'Health',
'Health',
'Health',
'Fashion',
'Travel',
'Travel',
'Health',
'Travel',
'Health',
'Food',
'Fitness',
'Fashion',
'Health',
'Fitness',
'Music',
'Fashion',
'Fashion',
'Family',
'Music',
'Music',
'Travel',
'Culture',
'Music',
'Food',
'Health',
'Travel',
'Music',
'Music',
```

```
'Food',
'Music',
'Health',
'Health',
'Family',
'Fashion',
'Family',
'Food',
'Health',
'Culture',
'Fitness',
'Family',
'Family',
'Fashion',
'Culture',
'Family',
'Health',
'Travel',
'Music',
'Fashion',
'Culture',
'Travel',
'Music',
'Food',
'Fashion',
'Food',
'Health',
'Food',
'Family',
'Music',
'Family',
'Fashion',
'Travel',
'Food',
'Travel',
'Culture',
'Culture',
'Music',
'Culture',
'Travel',
'Fashion',
'Music',
'Family',
'Fashion',
'Culture',
'Fashion',
'Fitness',
```

```
'Music',
'Health',
'Fitness',
'Fashion',
'Fashion',
'Music',
'Health',
'Travel',
'Health',
'Family',
'Culture',
'Culture',
'Music',
'Food',
'Family',
'Health',
'Culture',
'Food',
'Fashion',
'Culture',
'Health',
'Family',
'Culture',
'Family',
'Fitness',
'Family',
'Fitness',
'Food',
'Health',
'Fashion',
'Music',
'Culture',
'Health',
'Health',
'Family',
'Food',
'Health',
'Culture',
'Music',
'Health',
'Health',
'Family',
'Culture',
'Music',
'Culture',
'Fitness',
'Food',
```

```
'Health',
'Health',
'Health',
'Health',
'Food',
'Culture',
'Music',
'Fashion',
'Travel',
'Music',
'Travel',
'Fitness',
'Travel',
'Family',
'Food',
'Fitness',
'Food',
'Travel',
'Fashion',
'Fashion',
'Fashion',
'Music',
'Travel',
'Fitness',
'Music',
'Health',
'Food',
'Health',
'Fashion',
'Culture',
'Fashion',
'Health',
'Family',
'Family',
'Fitness',
'Culture',
'Culture',
'Culture',
'Fashion',
'Food',
'Health',
'Family',
'Fashion',
'Family',
'Travel',
'Music',
'Fitness',
```

```
'Health',
'Travel',
'Family',
'Culture',
'Health',
'Fitness',
'Culture',
'Fitness',
'Music',
'Fashion',
'Fashion',
'Fitness',
'Health',
'Food',
'Culture',
'Family',
'Music',
'Fashion',
'Health',
'Travel',
'Fitness',
'Music',
'Family',
'Food',
'Fitness',
'Health',
'Fitness',
'Food',
'Music',
'Family',
'Music',
'Culture',
'Travel',
'Fitness',
'Family',
'Food',
'Food',
'Health',
'Family',
'Health',
'Travel',
'Fashion',
'Food',
'Fashion',
'Music',
'Fitness',
'Music',
```

```
  'Music',
  'Family',
  'Fashion',
  'Fitness',
  'Food'],
 'Likes': array([5907, 8094,  294, 7937, 8801, 7511,  799, 2944,  197,   27,
100,
        4527, 1678, 7913, 2892, 9142, 5387, 4471, 5354, 7169, 3444, 2926,
        3165, 6365, 4524,  143, 6265, 2452,  562, 9300,  130, 8707, 3101,
        3081,  981, 9699, 4200, 9824, 8287, 2251,  916, 8879, 2897, 3211,
        2833, 3473, 5782, 8054, 6213, 3085, 1011, 5991, 1572, 6956, 9155,
        2861, 3969, 8897, 6641,  907, 9406, 4996, 6643, 6463, 2238, 8442,
        8965, 3354,  972, 7924, 3797,  841, 9482, 8811, 1510, 8626, 4220,
        2885, 5352,   36, 5412, 7453, 7139, 1092, 6809, 7977, 3230, 3495,
        6540, 6753, 2374,  448, 2841, 4620, 6863, 6863, 4042, 1983, 6800,
        7077, 5065, 9503, 4009, 9457, 8942, 8761,   23,  282, 7881, 4400,
        2617, 6176, 7838, 9546, 6647, 6074, 3467,  742, 1394, 6280, 9279,
        5825, 7671, 5044, 3370, 6779, 2804, 5068, 9501, 3784, 4770, 6343,
        6427,  507, 2469, 8237, 8382, 5230, 1858, 8906, 3890, 1679, 1821,
        3799, 5186, 3834, 3414, 9093,   77, 8316, 2871,  732, 3876, 3391,
        2329, 1620, 4731, 5012, 9918, 1295, 5563, 5971, 8846, 6655, 5296,
        1540, 3934, 6075, 1524, 8661, 3032, 5142, 2379, 5752, 9919, 2077,
        9744, 6349, 2231, 5171, 5499, 4462, 6612, 3686, 7524, 3199, 1864,
          13, 2706, 1698, 5906, 5172, 7286, 1100, 2672, 6362, 4948, 6001,
        6062, 7692, 3385, 7107, 7917, 2088, 1602, 9166, 2193,  804, 9681,
        8905, 5902,  733, 7016, 3781, 5199, 4017, 6012, 3497, 3409, 6730,
        2100, 3460, 6054, 7584, 9832, 7926,  740, 6186, 4483, 5487, 2553,
        9259, 6809, 6975,  323, 4183, 4958, 7238, 4392, 1615, 6821, 9794,
        8844, 6329, 1638, 1729, 8498, 7378, 4595, 3091, 7462, 5951, 2399,
        7461, 3022,  690, 1664, 5117, 8986, 7403, 8584, 7515, 5636, 8710,
        7068, 1756, 8170, 8528, 5229, 7102,  164,  328,  689, 8842, 8260,
        1331, 8586, 5736, 4256, 6824, 6212, 3921, 9448, 9451, 8013, 7994,
        2985, 2574, 3307, 1474, 9331, 3173, 2871, 2318, 8101, 4903, 7229,
         428, 1966, 7032, 7632, 5496, 1163, 7754, 7484, 8454, 8316, 1725,
        7824,  841, 6951, 1766, 1344, 1532, 9827, 5160, 8022, 8764, 1687,
        9204, 9654, 3343, 2136, 8443, 4951, 4444,  827, 7643, 9194,  146,
        6283, 1715, 3588, 7709, 5230,  318, 5092, 5896, 6826, 1648, 8897,
        2096, 9162, 3372,  481, 9672, 6889,  695, 2121, 5902, 6836, 9484,
        9312, 5128, 8951, 1324, 2707,  841, 7829, 8226, 9724, 5340, 3718,
         501, 7630, 7100, 4507, 8423, 3517, 3510, 8656,  928, 5969, 8662,
        5291,   43, 5474, 7587, 6139,  535, 6378, 8069,  997,  947, 9672,
        6545, 9160,  423, 3629, 6391, 9765, 1530, 9232, 2321, 8881, 7304,
        3151, 2879, 8829, 3112, 4186, 1661,  308, 1788, 9658, 5956,  724,
        9026, 9549, 1355, 2732, 6655,  899, 1900, 2403, 9432, 1446,  995,
        6721,  855, 2485, 4428, 1713, 7160, 6299, 2097,  677, 6153, 6461,
        9278, 5794, 5925, 2127, 2632, 9123, 5574, 2265,  382, 7437, 2790,
        8121,  475, 3741, 4010, 4261, 9936, 1755, 6152, 5163, 1089, 9891,
```

```
        9001, 8541, 8217, 2904, 8314, 3890, 7046, 1653, 9248, 1076, 2305,
        1607, 2942, 8228, 6456, 1355, 2625, 2103, 4551, 7433, 6356, 1836,
        5314, 4375, 8755, 2024, 1725,   26, 9343,  769, 1355, 5768, 2614,
        2693,  946, 3492, 8107,  738,   75, 5744, 2198,  829, 6097, 8393,
        5546, 9619,   56, 3408, 8085])}
```

[6]:
```
#creating dataframe from above dictionary

df = pd.DataFrame(data)
df
```

[6]:
```
           Date Category  Likes
0    2021-01-01  Fitness   5907
1    2021-01-02   Travel   8094
2    2021-01-03     Food    294
3    2021-01-04     Food   7937
4    2021-01-05  Fitness   8801
..          …       …    …
495  2022-05-11    Music   5546
496  2022-05-12   Family   9619
497  2022-05-13  Fashion     56
498  2022-05-14  Fitness   3408
499  2022-05-15     Food   8085

[500 rows x 3 columns]
```

[7]:
```
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
```

[8]:
```
df.describe()
```

[8]:
```
             Likes
count   500.000000
mean   4934.526000
std    2962.655724
min      13.000000
25%    2247.750000
```

```
50%    5122.500000
75%    7517.250000
max    9936.000000
```

[9]:
```
df['Category'].value_counts()
```

[9]:
```
Health     75
Music      68
Family     64
Fashion    63
Culture    62
Travel     60
Fitness    57
Food       51
Name: Category, dtype: int64
```

[10]:
```
df.head()
```

[10]:
```
        Date Category  Likes
0 2021-01-01  Fitness   5907
1 2021-01-02   Travel   8094
2 2021-01-03     Food    294
3 2021-01-04     Food   7937
4 2021-01-05  Fitness   8801
```

[11]:
```
#removed null values
df.dropna(inplace=True)
```

[12]:
```
df.drop_duplicates(inplace=True)
```

[13]:
```
df
```

[13]:
```
           Date Category  Likes
0    2021-01-01  Fitness   5907
1    2021-01-02   Travel   8094
2    2021-01-03     Food    294
3    2021-01-04     Food   7937
4    2021-01-05  Fitness   8801
..          …      …    …
495  2022-05-11    Music   5546
496  2022-05-12   Family   9619
497  2022-05-13  Fashion     56
498  2022-05-14  Fitness   3408
499  2022-05-15     Food   8085

[500 rows x 3 columns]
```

[14]:
```
df['Date'] = pd.to_datetime(df['Date'])
```

[15]:
```
df
```

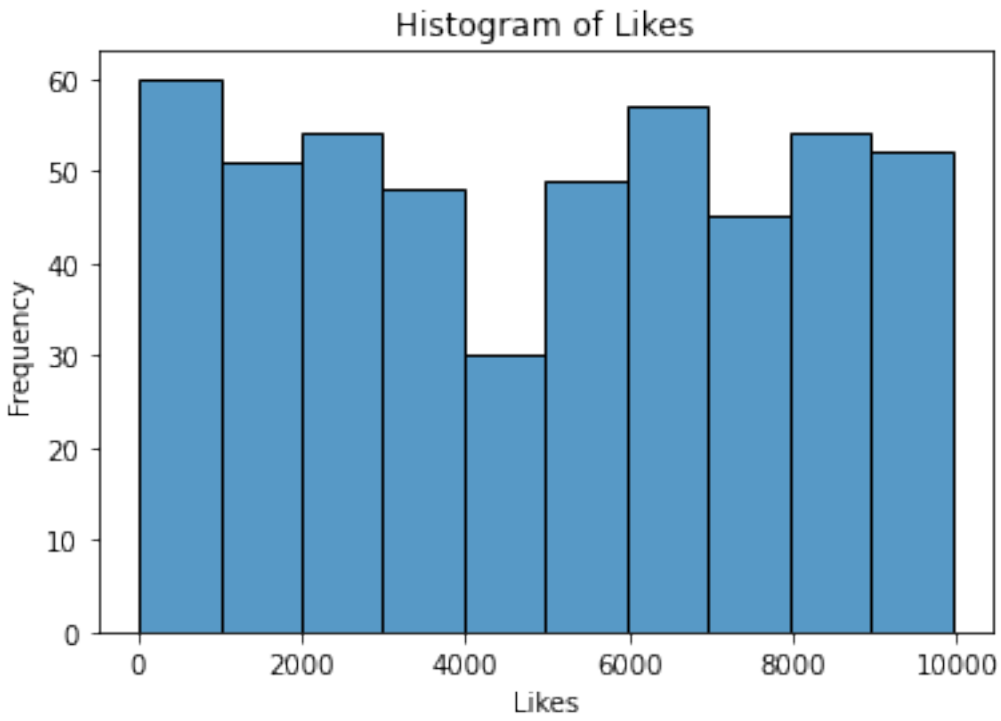
[15]:
```
           Date Category  Likes
0    2021-01-01  Fitness   5907
1    2021-01-02   Travel   8094
2    2021-01-03     Food    294
3    2021-01-04     Food   7937
4    2021-01-05  Fitness   8801
..          …       …    …
495  2022-05-11    Music   5546
496  2022-05-12   Family   9619
497  2022-05-13  Fashion     56
498  2022-05-14  Fitness   3408
499  2022-05-15     Food   8085

[500 rows x 3 columns]
```

[16]:
```
df['Likes'] = df['Likes'].astype(int)
```

[17]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 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: 15.6+ KB
```

[18]:
```
sns.histplot(df['Likes'])
plt.title('Histogram of Likes')
plt.xlabel('Likes')
plt.ylabel('Frequency')
plt.show()
```



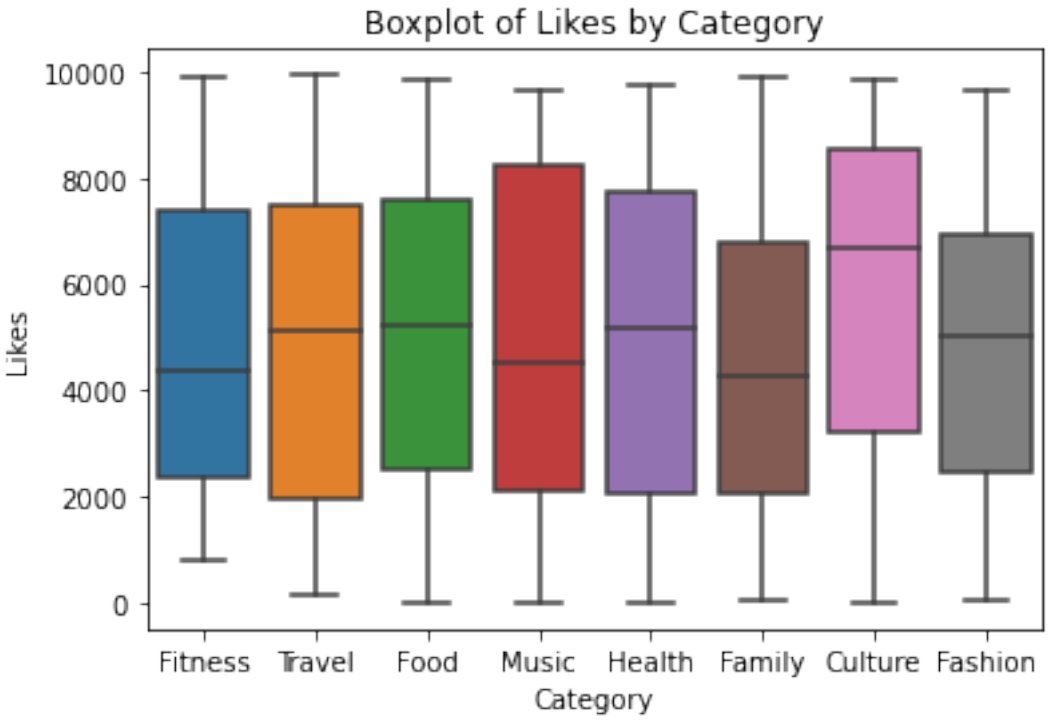
- The histogram of 'Likes' provides insights into the distribution of likes across the social media posts.
- The majority of posts seem to have a moderate number of likes, with a few outliers receiving significantly higher likes.

[19]:
```
pip install --upgrade seaborn
```

```
Requirement already satisfied: seaborn in /opt/conda/lib/python3.7/site-packages
(0.12.2)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in
/opt/conda/lib/python3.7/site-packages (from seaborn) (1.18.4)
Requirement already satisfied: pandas>=0.25 in /opt/conda/lib/python3.7/site-
packages (from seaborn) (1.0.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in
/opt/conda/lib/python3.7/site-packages (from seaborn) (3.2.1)
Requirement already satisfied: typing_extensions in
/opt/conda/lib/python3.7/site-packages (from seaborn) (3.7.4.2)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-
packages (from matplotlib!=3.6.1,>=3.1->seaborn) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn)
(1.2.0)
Requirement already satisfied: python-dateutil>=2.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn)
```

```
(2.8.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn)
(2.4.7)
Requirement already satisfied: pytz>=2017.2 in /opt/conda/lib/python3.7/site-
packages (from pandas>=0.25->seaborn) (2020.1)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages
(from cycler>=0.10->matplotlib!=3.6.1,>=3.1->seaborn) (1.14.0)
WARNING: You are using pip version 21.3.1; however, version 23.3.1 is
available.
You should consider upgrading via the '/opt/conda/bin/python -m pip install
--upgrade pip' command.
Note: you may need to restart the kernel to use updated packages.
```

[20]:
```
sns.boxplot(x='Category', y='Likes', data=df)
plt.title('Boxplot of Likes by Category')
plt.show()
```



- The boxplot of 'Likes' categorized by different social media categories reveals variations in engagement levels.
- Categories with wider interquartile ranges may have more diverse engagement levels, while those with narrower ranges might exhibit more consistent popularity.

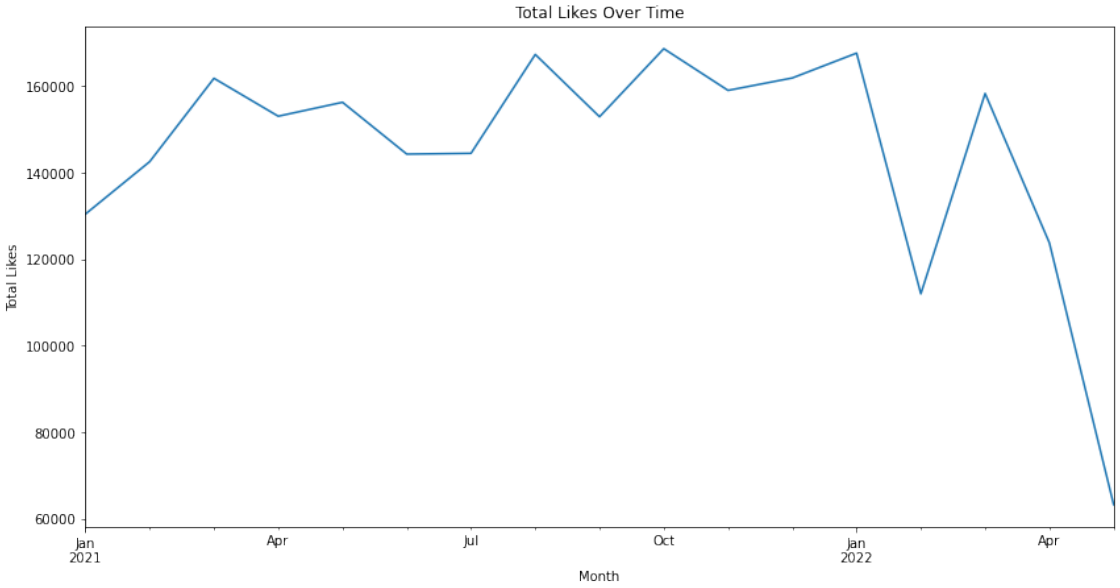
[22]:
```
print("\nMean of Likes:", df['Likes'].mean())
```

```
Mean of Likes: 4934.526
```

[23]:
```
#Mean Likes for Each Category
category_likes_mean = df.groupby('Category')['Likes'].mean()
print(category_likes_mean)
```

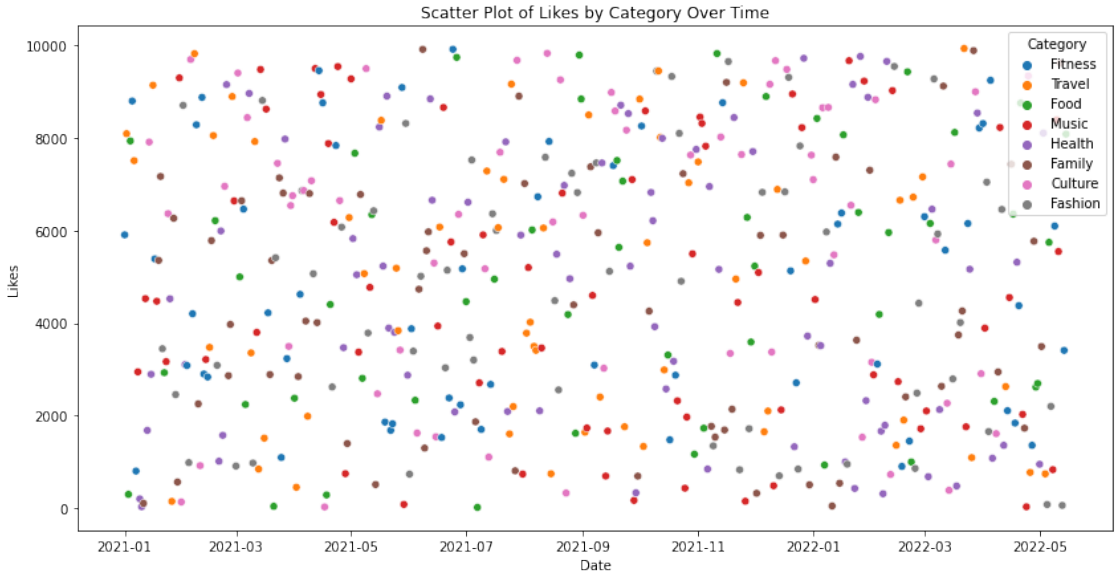
```
Category
Culture    5880.112903
Family     4461.890625
Fashion    4772.031746
Fitness    4841.736842
Food       5017.705882
Health     4807.453333
Music      4836.588235
Travel     4919.466667
Name: Likes, dtype: float64
```

[26]:
```
#likes over time
plt.figure(figsize=(14, 7))
df.groupby(df['Date'].dt.to_period('M'))['Likes'].sum().plot(kind='line')
plt.title('Total Likes Over Time')
plt.xlabel('Month')
plt.ylabel('Total Likes')
plt.show()
```



The line plot visually represents the total likes over time, with each point on the line corresponding to a specific month. The upward or downward movement of the line indicates the overall trend in user engagement.

[28]:
```
#category and likes over time
plt.figure(figsize=(14, 7))
sns.scatterplot(data=df, x='Date', y='Likes', hue='Category')
plt.title('Scatter Plot of Likes by Category Over Time')
plt.xlabel('Date')
plt.ylabel('Likes')
plt.legend(title='Category')
plt.show()
```



The scatter plot shows the spread of 'Likes' across different categories over time. Some categories may show consistent engagement, while others have varied performance. This could suggest the presence of seasonality or varying audience preferences. Identifying these patterns can assist in tailoring content to audience tastes and potentially increasing overall engagement.

These insights can guide strategic decisions for content creation, marketing campaigns, and audience engagement initiatives.

- The project demonstrated a comprehensive data analysis workflow for social media data using Python and key libraries.
- Visualizations provided a clear representation of engagement trends, aiding in strategic decision-making.
- Insights gained from the analysis can guide content creators and marketers to optimize their social media strategies.
- This project serves as a practical example of leveraging Python for social media data analysis, offering a foundation for more advanced analyses and real-world applications.

[ ]: