

Social media

June 27, 2023

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.

```
[7]: !pip install pandas
      !pip install matplotlib
```

```
Requirement already satisfied: pandas in /opt/conda/lib/python3.7/site-packages
(1.0.3)
```

```
Requirement already satisfied: numpy>=1.13.3 in /opt/conda/lib/python3.7/site-
packages (from pandas) (1.18.4)
```

```
Requirement already satisfied: pytz>=2017.2 in /opt/conda/lib/python3.7/site-
packages (from pandas) (2020.1)
```

```
Requirement already satisfied: python-dateutil>=2.6.1 in
/opt/conda/lib/python3.7/site-packages (from pandas) (2.8.1)
```

```
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.7/site-
packages (from python-dateutil>=2.6.1->pandas) (1.14.0)
```

```
WARNING: You are using pip version 21.3.1; however, version 23.1.2 is
available.
```

```
You should consider upgrading via the '/opt/conda/bin/python3 -m pip install
--upgrade pip' command.
```

```
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.7/site-
packages (3.2.1)
```

```
Requirement already satisfied: python-dateutil>=2.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib) (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) (2.4.7)
```

```
Requirement already satisfied: numpy>=1.11 in /opt/conda/lib/python3.7/site-
packages (from matplotlib) (1.18.4)
```

```
Requirement already satisfied: kiwisolver>=1.0.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib) (1.2.0)
```

```
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-
packages (from matplotlib) (0.10.0)
```

```
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages
(from cycler>=0.10->matplotlib) (1.14.0)
```

```
WARNING: You are using pip version 21.3.1; however, version 23.1.2 is
available.
```

```
You should consider upgrading via the '/opt/conda/bin/python3 -m pip install
--upgrade pip' command.
```

```
[37]: !install seaborn from
```

```
install: cannot stat 'seaborn': No such file or directory
```

```
[9]: !pip install seaborn
```

```
Requirement already satisfied: seaborn in /opt/conda/lib/python3.7/site-packages (0.10.1)
```

```
Requirement already satisfied: matplotlib>=2.1.2 in /opt/conda/lib/python3.7/site-packages (from seaborn) (3.2.1)
```

```
Requirement already satisfied: scipy>=1.0.1 in /opt/conda/lib/python3.7/site-packages (from seaborn) (1.4.1)
```

```
Requirement already satisfied: numpy>=1.13.3 in /opt/conda/lib/python3.7/site-packages (from seaborn) (1.18.4)
```

```
Requirement already satisfied: pandas>=0.22.0 in /opt/conda/lib/python3.7/site-packages (from seaborn) (1.0.3)
```

```
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.7/site-packages (from matplotlib>=2.1.2->seaborn) (1.2.0)
```

```
Requirement already satisfied: python-dateutil>=2.1 in /opt/conda/lib/python3.7/site-packages (from matplotlib>=2.1.2->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>=2.1.2->seaborn) (2.4.7)
```

```
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-packages (from matplotlib>=2.1.2->seaborn) (0.10.0)
```

```
Requirement already satisfied: pytz>=2017.2 in /opt/conda/lib/python3.7/site-packages (from pandas>=0.22.0->seaborn) (2020.1)
```

```
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages (from cycler>=0.10->matplotlib>=2.1.2->seaborn) (1.14.0)
```

```
WARNING: You are using pip version 21.3.1; however, version 23.1.2 is available.
```

```
You should consider upgrading via the '/opt/conda/bin/python3 -m pip install --upgrade pip' command.
```

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

Task 2

```
[11]: categories = ['Food', 'Travel', 'Fashion', 'Fitness', 'Music', 'Culture',
    ↪ 'Family', 'Health']
num_periods = 500

data_dict = {
```

```

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

print(data_dict)

```

```

{'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':
['Travel', 'Travel', 'Family', 'Fashion', 'Food', 'Fashion', 'Music', 'Music',
'Culture', 'Health', 'Fitness', 'Music', 'Fashion', 'Music', 'Family',
'Culture', 'Travel', 'Family', 'Culture', 'Fitness', 'Fitness', 'Fitness',
'Culture', 'Culture', 'Family', 'Fitness', 'Food', 'Family', 'Food', 'Travel',
'Health', 'Food', 'Travel', 'Culture', 'Family', 'Health', 'Health', 'Health',
'Travel', 'Travel', 'Health', 'Culture', 'Health', 'Fashion', 'Food', 'Fitness',
'Fitness', 'Family', 'Culture', 'Music', 'Health', 'Travel', 'Culture',
'Family', 'Fashion', 'Family', 'Music', 'Health', 'Health', 'Family', 'Fitness',
'Culture', 'Music', 'Fashion', 'Fitness', 'Fitness', 'Travel', 'Travel',
'Health', 'Culture', 'Travel', 'Fashion', 'Family', 'Food', 'Fashion', 'Music',
'Music', 'Family', 'Fashion', 'Food', 'Travel', 'Travel', 'Travel', 'Travel',
'Family', 'Health', 'Travel', 'Travel', 'Fitness', 'Food', 'Travel', 'Fashion',
'Health', 'Health', 'Health', 'Health', 'Health', 'Family', 'Travel', 'Family',
'Fashion', 'Music', 'Music', 'Culture', 'Music', 'Food', 'Music', 'Music',
'Food', 'Family', 'Family', 'Travel', 'Music', 'Family', 'Family', 'Food',
'Fitness', 'Fitness', 'Music', 'Fitness', 'Culture', 'Fitness', 'Food', 'Music',
'Food', 'Fitness', 'Fashion', 'Food', 'Family', 'Food', 'Food', 'Culture',
'Food', 'Music', 'Culture', 'Food', 'Fitness', 'Fitness', 'Fashion', 'Family',
'Fitness', 'Culture', 'Food', 'Culture', 'Fashion', 'Travel', 'Culture',
'Health', 'Health', 'Music', 'Fitness', 'Family', 'Music', 'Music', 'Culture',
'Music', 'Fashion', 'Food', 'Food', 'Fitness', 'Fitness', 'Food', 'Fashion',
'Culture', 'Music', 'Food', 'Health', 'Fitness', 'Food', 'Music', 'Travel',
'Food', 'Music', 'Health', 'Food', 'Travel', 'Fitness', 'Culture', 'Health',
'Family', 'Music', 'Health', 'Travel', 'Travel', 'Food', 'Culture', 'Family',
'Family', 'Fashion', 'Travel', 'Culture', 'Fitness', 'Health', 'Travel',
'Fitness', 'Family', 'Travel', 'Culture', 'Culture', 'Health', 'Travel',
'Fashion', 'Fitness', 'Fitness', 'Travel', 'Fashion', 'Fashion', 'Fitness',
'Family', 'Music', 'Food', 'Fashion', 'Food', 'Culture', 'Culture', 'Food',
'Fashion', 'Health', 'Music', 'Fashion', 'Fashion', 'Fitness', 'Health',
'Health', 'Family', 'Music', 'Family', 'Travel', 'Family', 'Family', 'Culture',
'Travel', 'Fitness', 'Travel', 'Travel', 'Music', 'Fitness', 'Culture', 'Food',
'Music', 'Music', 'Family', 'Travel', 'Health', 'Fitness', 'Food', 'Food',

```

'Travel', 'Culture', 'Family', 'Health', 'Travel', 'Family', 'Travel',
 'Fitness', 'Fashion', 'Health', 'Travel', 'Health', 'Fitness', 'Travel',
 'Culture', 'Fashion', 'Culture', 'Travel', 'Health', 'Travel', 'Food',
 'Culture', 'Food', 'Culture', 'Fashion', 'Family', 'Culture', 'Culture',
 'Travel', 'Family', 'Culture', 'Music', 'Health', 'Fitness', 'Health',
 'Fashion', 'Culture', 'Music', 'Travel', 'Fashion', 'Fitness', 'Fashion',
 'Health', 'Food', 'Music', 'Culture', 'Fashion', 'Culture', 'Culture', 'Travel',
 'Family', 'Culture', 'Food', 'Health', 'Food', 'Fitness', 'Culture', 'Music',
 'Food', 'Travel', 'Fitness', 'Music', 'Travel', 'Health', 'Culture', 'Music',
 'Fashion', 'Culture', 'Food', 'Food', 'Health', 'Travel', 'Fitness', 'Fashion',
 'Travel', 'Travel', 'Health', 'Health', 'Fashion', 'Family', 'Fitness',
 'Health', 'Fashion', 'Family', 'Fashion', 'Fitness', 'Fitness', 'Food',
 'Fitness', 'Fitness', 'Fitness', 'Culture', 'Health', 'Travel', 'Health',
 'Music', 'Food', 'Health', 'Fitness', 'Fitness', 'Music', 'Culture', 'Fashion',
 'Culture', 'Travel', 'Fashion', 'Food', 'Food', 'Culture', 'Culture', 'Culture',
 'Health', 'Culture', 'Music', 'Travel', 'Culture', 'Family', 'Health', 'Music',
 'Travel', 'Culture', 'Culture', 'Culture', 'Food', 'Fashion', 'Culture',
 'Music', 'Music', 'Culture', 'Travel', 'Food', 'Family', 'Travel', 'Fitness',
 'Family', 'Health', 'Culture', 'Food', 'Music', 'Travel', 'Family', 'Culture',
 'Fitness', 'Fitness', 'Family', 'Music', 'Travel', 'Family', 'Fashion',
 'Fashion', 'Music', 'Family', 'Health', 'Family', 'Food', 'Family', 'Family',
 'Food', 'Health', 'Food', 'Health', 'Fitness', 'Family', 'Food', 'Travel',
 'Health', 'Health', 'Fashion', 'Fitness', 'Culture', 'Family', 'Fitness',
 'Fashion', 'Food', 'Food', 'Travel', 'Travel', 'Music', 'Travel', 'Health',
 'Culture', 'Music', 'Culture', 'Food', 'Music', 'Food', 'Culture', 'Culture',
 'Fitness', 'Culture', 'Fashion', 'Family', 'Family', 'Fitness', 'Fitness',
 'Health', 'Travel', 'Health', 'Health', 'Fashion', 'Fitness', 'Family', 'Food',
 'Food', 'Culture', 'Food', 'Travel', 'Family', 'Fitness', 'Fitness', 'Music',
 'Fitness', 'Fashion', 'Fitness', 'Fitness', 'Culture', 'Food', 'Food', 'Travel',
 'Family', 'Family', 'Fashion', 'Fashion', 'Fitness', 'Food', 'Food', 'Music',
 'Health', 'Travel', 'Fitness', 'Culture', 'Culture', 'Fashion', 'Food', 'Music',
 'Culture', 'Food', 'Food', 'Fashion', 'Health', 'Family', 'Music', 'Health',
 'Fashion', 'Culture', 'Travel', 'Food', 'Music', 'Travel', 'Family', 'Music',
 'Food', 'Fashion'], 'Likes': array([9388, 361, 9161, 1374, 1707, 5604, 5865,
 768, 7917, 5444, 308,

734, 7817, 446, 827, 8007, 5818, 1497, 3130, 9569, 5391, 7603,
 910, 1618, 9173, 3156, 1631, 7429, 3550, 5438, 2373, 6939, 2055,
 3837, 6431, 7980, 3211, 3293, 5108, 368, 6907, 5746, 8464, 8395,
 6473, 5825, 4663, 2719, 420, 1861, 4398, 6440, 8125, 4709, 8026,
 9904, 5110, 5833, 3052, 7184, 6487, 1917, 2648, 9237, 4846, 5549,
 3099, 5333, 3721, 8775, 9406, 9787, 9241, 9182, 3496, 958, 3179,
 2777, 9313, 2799, 3319, 6227, 5732, 7426, 9784, 1729, 9391, 6101,
 1627, 5887, 20, 2892, 7844, 4801, 7046, 9036, 5369, 7634, 9651,
 9240, 2320, 6855, 5770, 6582, 8259, 9911, 1689, 7615, 5633, 1821,
 2358, 6124, 7477, 9348, 5850, 5990, 36, 8618, 6962, 7807, 4332,
 7539, 7062, 9862, 2281, 5282, 7135, 3448, 2121, 789, 9306, 9619,
 9656, 9403, 5411, 8345, 5739, 9558, 3430, 994, 8763, 6198, 5958,
 7780, 7398, 8490, 8123, 6604, 6913, 4452, 3327, 8375, 5819, 7446,

```

5809, 8404, 7508, 3031, 4509, 1336, 6186, 3258, 7697, 5359, 75,
9246, 1137, 9807, 4607, 5966, 1166, 8868, 973, 9446, 2467, 8353,
4417, 4044, 8683, 6351, 7919, 3123, 8428, 8681, 4978, 7171, 8018,
9351, 8966, 3195, 473, 2703, 8382, 3729, 6085, 4011, 3050, 3232,
6788, 3430, 3834, 9677, 2632, 6420, 6335, 717, 8821, 2749, 3452,
7082, 2219, 6908, 2469, 4326, 763, 7049, 7202, 8151, 9826, 1039,
4768, 1036, 453, 4147, 7587, 1732, 1161, 2148, 5321, 5384, 2368,
1234, 4246, 5206, 6720, 3073, 6281, 7113, 4310, 7669, 4540, 4564,
8704, 4445, 6534, 7943, 3052, 1949, 4684, 5940, 2749, 7641, 7371,
1342, 606, 9387, 898, 1466, 6460, 1823, 2945, 2688, 1340, 1355,
2623, 581, 5719, 8705, 2728, 9091, 7431, 6832, 8746, 1118, 9030,
5249, 8943, 8136, 9409, 9514, 7761, 803, 3507, 7944, 5003, 5048,
6029, 8355, 8290, 7392, 746, 3669, 5260, 6900, 7300, 4699, 7257,
43, 2859, 8078, 9557, 4928, 6523, 9286, 5931, 4429, 4091, 9648,
9735, 3332, 7963, 8381, 7009, 9315, 3892, 379, 4599, 408, 3825,
5598, 4648, 2213, 8879, 3669, 9892, 8185, 2796, 2068, 472, 1861,
1904, 7938, 435, 5688, 917, 7137, 3088, 831, 703, 6770, 8783,
4030, 7473, 5527, 1724, 1300, 6732, 9802, 3696, 9919, 4984, 4242,
8354, 6852, 9848, 701, 2117, 7722, 7570, 3445, 3236, 8435, 5078,
6049, 4239, 2043, 3636, 3660, 9740, 7621, 1271, 4658, 5338, 6036,
3468, 4166, 7884, 8600, 6233, 5184, 8364, 5795, 4714, 6757, 6375,
7693, 5031, 360, 7400, 1414, 2465, 955, 3735, 4093, 8121, 1827,
8502, 4705, 8506, 2815, 3933, 7572, 5177, 7929, 7201, 7167, 6393,
9132, 2635, 3553, 6159, 8411, 2606, 4116, 9344, 6004, 3252, 1452,
332, 250, 1648, 5634, 1651, 4199, 4690, 9353, 5881, 5815, 1663,
486, 221, 6329, 3900, 6008, 9128, 5308, 2851, 5449, 8572, 8457,
2361, 4668, 7209, 8792, 3115, 385, 8389, 9653, 7805, 8124, 6005,
3261, 1618, 2905, 8648, 9170, 8603, 3632, 981, 1292, 2996, 320,
775, 2442, 2890, 9206, 7638, 4401, 875, 7550, 319, 94, 1425,
5843, 8606, 622, 6722, 6734, 792, 6101, 5903, 6336, 6479, 4699,
414, 6723, 7711, 4869, 3480, 8388, 5060, 5877, 8366, 7260, 8829,
7751, 4380, 8149, 3733, 8475]})

```

Task 3

```

[12]: df=pd.DataFrame(data_dict)
      print(df)

```

	Date	Category	Likes
0	2021-01-01	Travel	9388
1	2021-01-02	Travel	361
2	2021-01-03	Family	9161
3	2021-01-04	Fashion	1374
4	2021-01-05	Food	1707
..
495	2022-05-11	Travel	7751
496	2022-05-12	Family	4380
497	2022-05-13	Music	8149

```
498 2022-05-14      Food    3733
499 2022-05-15  Fashion    8475
```

```
[500 rows x 3 columns]
```

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

```
[13]:      Date Category  Likes
0 2021-01-01   Travel  9388
1 2021-01-02   Travel   361
2 2021-01-03   Family  9161
3 2021-01-04  Fashion  1374
4 2021-01-05    Food  1707
```

```
[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.describe()
```

```
[15]:      Likes
count    500.000000
mean    5249.806000
std     2860.659344
min       20.000000
25%     2882.250000
50%     5488.000000
75%     7729.250000
max     9919.000000
```

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

```
[16]: Culture      72
      Food       68
      Travel     68
      Fitness    65
      Health     60
      Family     59
      Music      56
```

```
Fashion    52
Name: Category, dtype: int64
```

Task 4

```
[17]: df=df.dropna()
df
```

```
[17]:
```

	Date	Category	Likes
0	2021-01-01	Travel	9388
1	2021-01-02	Travel	361
2	2021-01-03	Family	9161
3	2021-01-04	Fashion	1374
4	2021-01-05	Food	1707
..
495	2022-05-11	Travel	7751
496	2022-05-12	Family	4380
497	2022-05-13	Music	8149
498	2022-05-14	Food	3733
499	2022-05-15	Fashion	8475

[500 rows x 3 columns]

```
[19]: df=df.drop_duplicates()
df
```

```
[19]:
```

	Date	Category	Likes
0	2021-01-01	Travel	9388
1	2021-01-02	Travel	361
2	2021-01-03	Family	9161
3	2021-01-04	Fashion	1374
4	2021-01-05	Food	1707
..
495	2022-05-11	Travel	7751
496	2022-05-12	Family	4380
497	2022-05-13	Music	8149
498	2022-05-14	Food	3733
499	2022-05-15	Fashion	8475

[500 rows x 3 columns]

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

```
[23]:
```

	Date	Category	Likes
0	2021-01-01	Travel	9388
1	2021-01-02	Travel	361

2	2021-01-03	Family	9161
3	2021-01-04	Fashion	1374
4	2021-01-05	Food	1707
..
495	2022-05-11	Travel	7751
496	2022-05-12	Family	4380
497	2022-05-13	Music	8149
498	2022-05-14	Food	3733
499	2022-05-15	Fashion	8475

[500 rows x 3 columns]

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

```
[27]:
```

	Date	Category	Likes
0	2021-01-01	Travel	9388
1	2021-01-02	Travel	361
2	2021-01-03	Family	9161
3	2021-01-04	Fashion	1374
4	2021-01-05	Food	1707
..
495	2022-05-11	Travel	7751
496	2022-05-12	Family	4380
497	2022-05-13	Music	8149
498	2022-05-14	Food	3733
499	2022-05-15	Fashion	8475

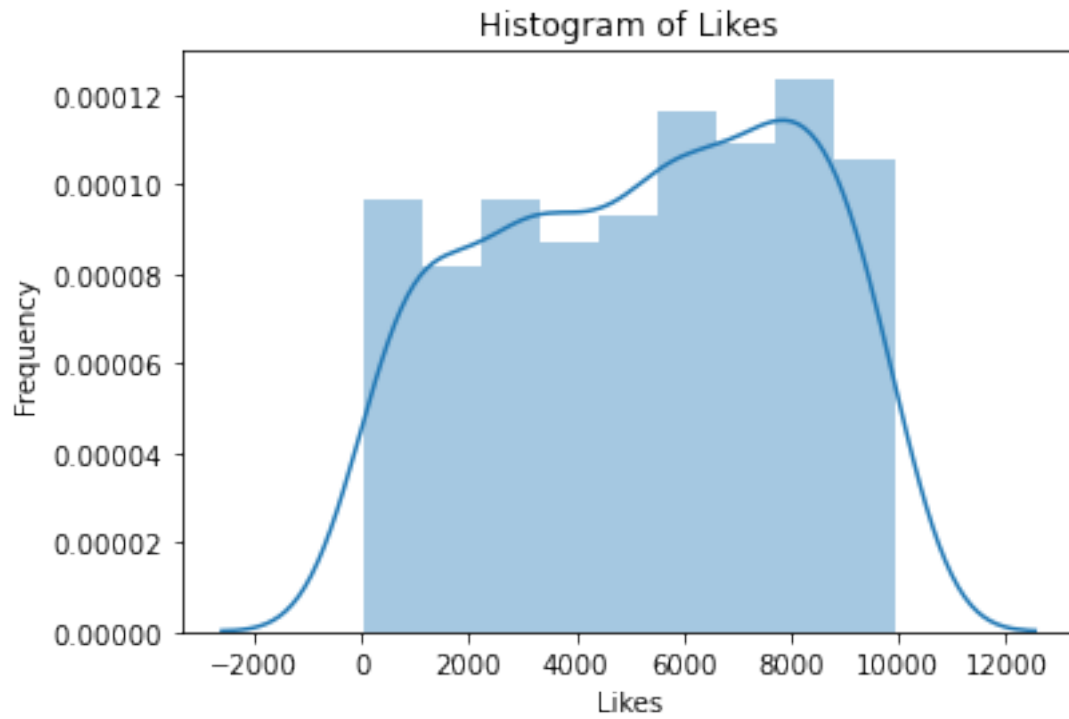
[500 rows x 3 columns]

Task 5

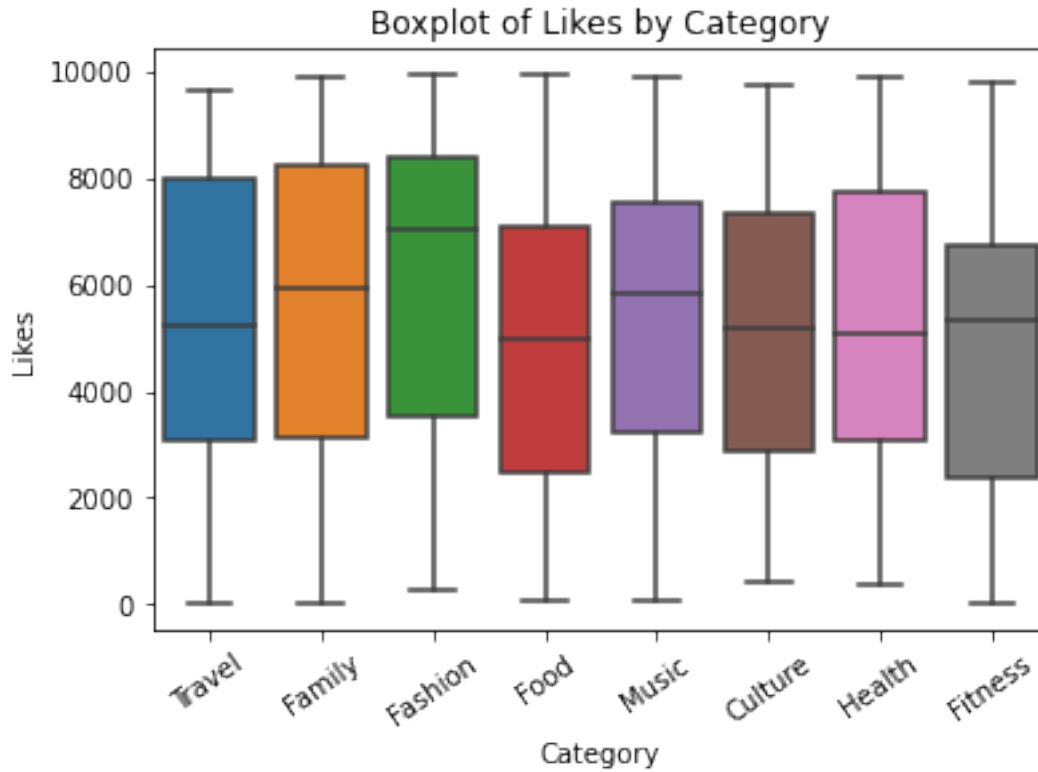
```
[53]: import seaborn as sns
import matplotlib.pyplot as plt

sns.distplot(df['Likes'])

plt.xlabel('Likes')
plt.ylabel('Frequency')
plt.title('Histogram of Likes')
plt.show()
```



```
[60]: sns.boxplot(data=df,y=df['Likes'],x=df['Category'])  
plt.xlabel('Category')  
plt.ylabel('Likes')  
plt.title('Boxplot of Likes by Category')  
plt.xticks(Rotation=35)  
plt.show()
```



```
[65]: Likes_mean=df['Likes'].mean()
Likes_mean
```

```
[65]: 5249.806
```

```
[66]: category_likes_mean = df.groupby('Category')['Likes'].mean()
category_likes_mean
```

```
[66]: Category
Culture    4962.083333
Family    5633.101695
Fashion    6087.057692
Fitness    4800.507692
Food       4974.014706
Health     5199.183333
Music      5327.410714
Travel     5267.661765
Name: Likes, dtype: float64
```

Task 5 Conclusion

The above project focused on analyzing social media data using Python and pandas.

Defined a list of categories for the social media experiment, such as Food, Travel, Fashion, Fitness, Music, Culture, Family, and Health.

Generated a Python DataFrame with fields 'Date', 'Category', and 'Likes' using random data. The 'Date' field was created using `pd.date_range()`, the 'Category' field was populated with random choices using `random.choice()`, and the 'Likes' field was filled with random integers using `np.random.randint()`.

Explored and analyzed the data by printing the count of each 'Category' element, dropping null values, and removing duplicate data using the DataFrame methods.

Converted the 'Date' column to a datetime format using `pd.to_datetime()` to facilitate appropriate display and analysis of the dates.

Converted the 'Likes' data to integer type using the `astype()` method to ensure numerical calculations and analysis.

Visualized the data by creating a histogram plot of the 'Likes' using `sns.histplot()` or `sns.distplot()` from the seaborn library.

Created a boxplot with the x-axis as 'Category' and the y-axis as 'Likes' using `sns.boxplot()` to examine the distribution and variability of 'Likes' across different categories.

Calculated and printed the mean of the 'Likes' category using the `mean()` method on the 'Likes' column.

[]: