SocialMediaDataAnalysis-Copy1

February 15, 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.

[4]:
```
# your code here
import numpy as np
import matplotlib
import seaborn
import random
import pandas as pd
```

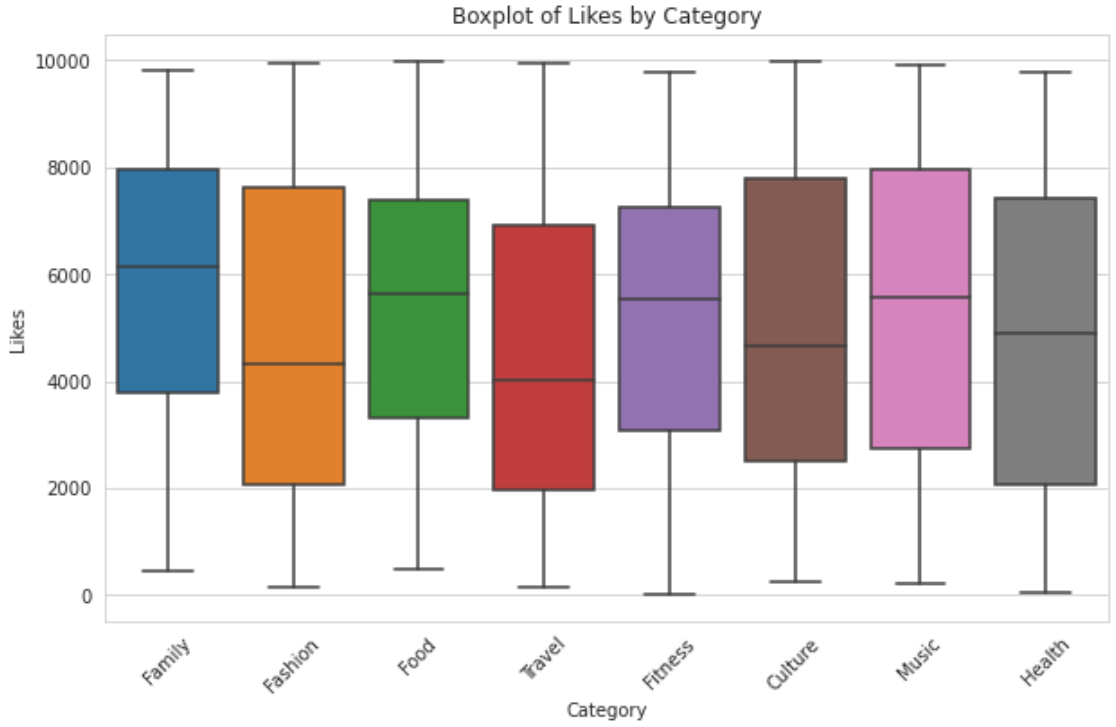
[5]:
```
# Define the list of categories
categories = ['Food', 'Travel', 'Fashion', 'Fitness', 'Music', 'Culture',␣
 ↪'Family', 'Health']

# Number of entries (n)
n = 500

# Generate random 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 DataFrame
df = pd.DataFrame(data)

# Print the first few rows of the DataFrame
print("DataFrame Head:")
print(df.head())

# Print DataFrame Information
print("\nDataFrame Information:")
print(df.info())

# Print DataFrame Description
print("\nDataFrame Description:")
print(df.describe())

# Print the count of each 'Category' element
category_counts = df['Category'].value_counts()
print("\nCount of each 'Category' element:")
```

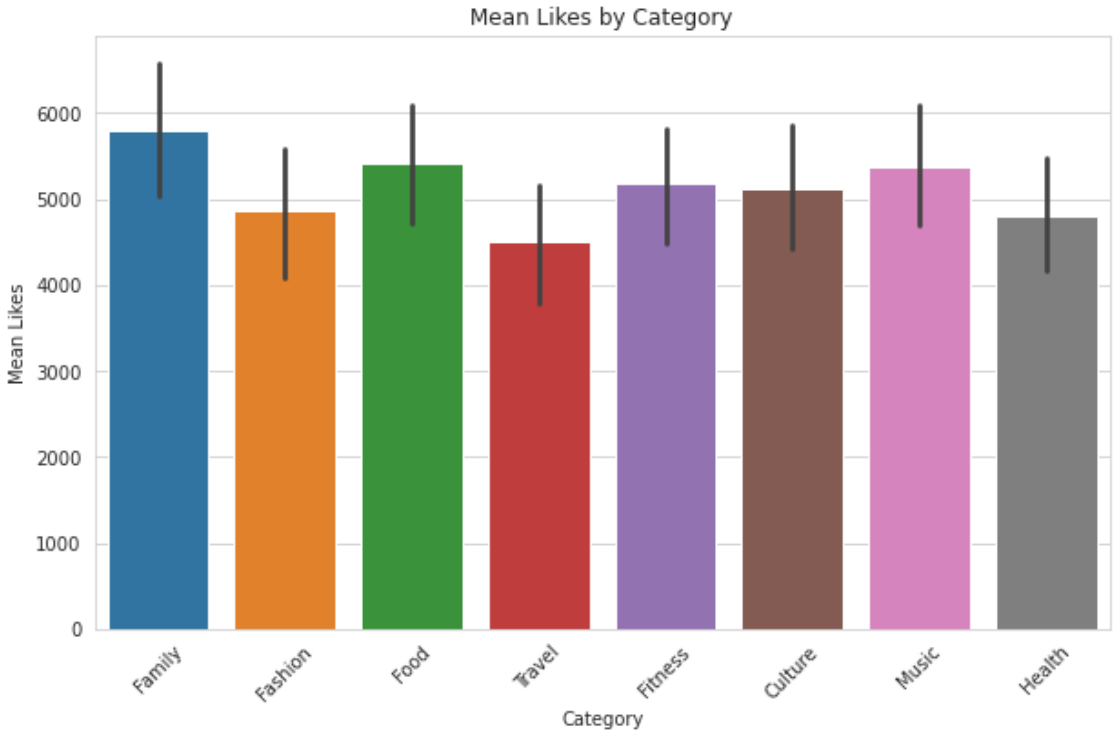
```
print(category_counts)
```

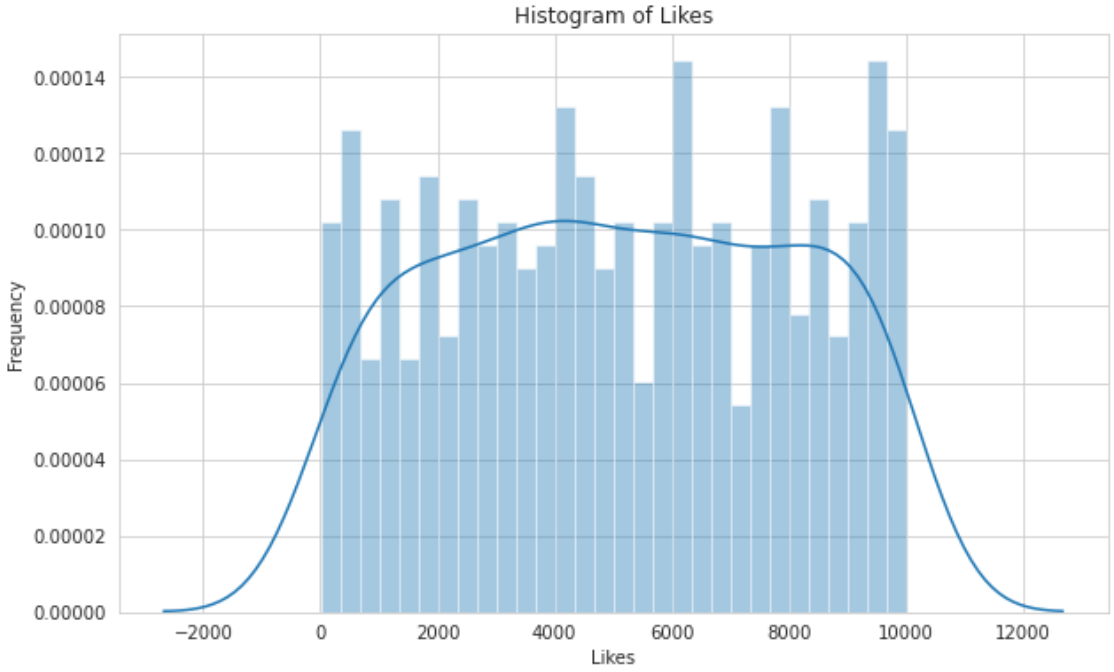
```
DataFrame Head:
        Date Category  Likes
0 2021-01-01   Family   9501
1 2021-01-02  Fashion    316
2 2021-01-03     Food   6489
3 2021-01-04   Travel   6307
4 2021-01-05  Fitness   5039

DataFrame Information:
<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
None

DataFrame Description:
             Likes
count   500.000000
mean   5104.464000
std    2931.690684
min      11.000000
25%    2627.500000
50%    5043.500000
75%    7680.500000
max    9996.000000

Count of each 'Category' element:
Health     75
Fitness    66
Travel     64
Fashion    64
Music      62
Culture    62
Food       54
Family     53
Name: Category, dtype: int64
```

[14]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Set the style of seaborn
sns.set_style("whitegrid")

# Create a boxplot of 'Likes' grouped by 'Category'
plt.figure(figsize=(10, 6))
sns.boxplot(x='Category', y='Likes', data=df)
plt.title('Boxplot of Likes by Category')
plt.xticks(rotation=45)
plt.show()



# Create a grouped barplot of the mean 'Likes' by 'Category'
plt.figure(figsize=(10, 6))
sns.barplot(x='Category', y='Likes', data=df, estimator=np.mean)
plt.title('Mean Likes by Category')
plt.xticks(rotation=45)
plt.ylabel('Mean Likes')
plt.show()


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


Boxplot of Likes by Category
Likes
10000
8000
6000
4000
2000
0
Family
Fashion
Food
Travel
Fitness
Culture
Music
Health
Category


Mean Likes by Category
Mean Likes
6000
5000
4000
3000
2000
1000
0
Family
Fashion
Food
Travel
Fitness
Culture
Music
Health
Category


Histogram of Likes
Frequency
0.00014
0.00012
0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
−2000
0
2000
4000
6000
8000
10000
12000
Likes

[ ]: