SocialMediaDataAnalysis

September 22, 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.

Task 1 – Import required libraries

[1]:
```
# your code here
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

[2]:
```
import random as rnd
plt.show()
```

Task 2 – Generate random data for the social media data

[3]:
```
cat_list = ['Fitness', 'Beauty', 'Food', 'Travel']

date = pd.date_range('2024-01-01', periods = 100)
cat = (rnd.choice(cat_list) for i in range (100))
likes = np.random.randint(0,10000,size = 100)
```

Task 3 – Load the data into a Pandas DataFrame and Explore the data

[4]:
```
dict = {
    'Date' : date,
    'Category' : cat,
    'Likes' : likes
}
df = pd.DataFrame(dict)
print(df.shape)
df.head()
```

```
(100, 3)
```

[4]:

	Date	Category	Likes
0	2024-01-01	Beauty	1703
1	2024-01-02	Fitness	229
2	2024-01-03	Beauty	7279
3	2024-01-04	Beauty	8492
4	2024-01-05	Food	2422

[5]:
```
df
```

[5]:
```
          Date Category  Likes
0   2024-01-01   Beauty   1703
1   2024-01-02  Fitness    229
2   2024-01-03   Beauty   7279
3   2024-01-04   Beauty   8492
4   2024-01-05     Food   2422
..         …      …    …
95  2024-04-05  Fitness   3723
96  2024-04-06  Fitness   9227
97  2024-04-07   Beauty   7657
98  2024-04-08  Fitness   5082
99  2024-04-09   Beauty   7057

[100 rows x 3 columns]
```

[6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Date      100 non-null    datetime64[ns]
 1   Category  100 non-null    object
 2   Likes     100 non-null    int64
dtypes: datetime64[ns](1), int64(1), object(1)
memory usage: 2.5+ KB
```

[7]:
```
df.describe()
```

[7]:
```
             Likes
count   100.000000
mean   4844.850000
std    2896.393715
min     113.000000
25%    2621.750000
50%    4709.500000
75%    7057.500000
max    9881.000000
```

Task 4 – Clean the data

[8]:
```
df = df.dropna()
df = df[~df.duplicated()]
print(df.shape)
df
```

```
(100, 3)
```

[8]:
```
          Date Category  Likes
0   2024-01-01   Beauty   1703
1   2024-01-02  Fitness    229
2   2024-01-03   Beauty   7279
3   2024-01-04   Beauty   8492
4   2024-01-05     Food   2422
..         …      …    …
95  2024-04-05  Fitness   3723
96  2024-04-06  Fitness   9227
97  2024-04-07   Beauty   7657
98  2024-04-08  Fitness   5082
99  2024-04-09   Beauty   7057

[100 rows x 3 columns]
```

[9]:
```
df['Date'] = pd.to_datetime(df['Date'], format = '%Y-%m-%d')
df['Likes'] = df['Likes'].astype('int64')
print(df.shape)
print(df.dtypes)
df
```

```
(100, 3)
Date        datetime64[ns]
Category            object
Likes                int64
dtype: object
```

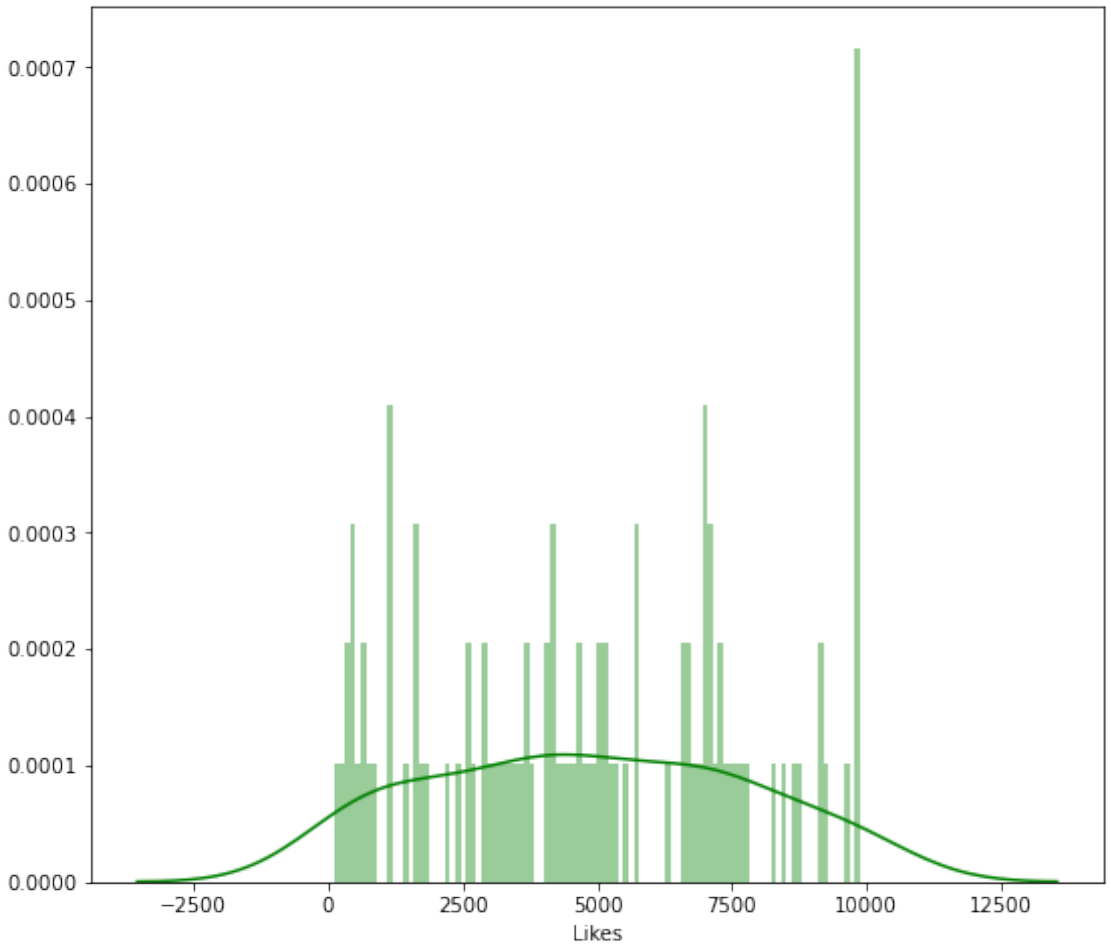
[9]:
```
          Date Category  Likes
0   2024-01-01   Beauty   1703
1   2024-01-02  Fitness    229
2   2024-01-03   Beauty   7279
3   2024-01-04   Beauty   8492
4   2024-01-05     Food   2422
..         …      …    …
95  2024-04-05  Fitness   3723
96  2024-04-06  Fitness   9227
97  2024-04-07   Beauty   7657
98  2024-04-08  Fitness   5082
99  2024-04-09   Beauty   7057

[100 rows x 3 columns]
```

Task 5– Visualize and Analyze the data

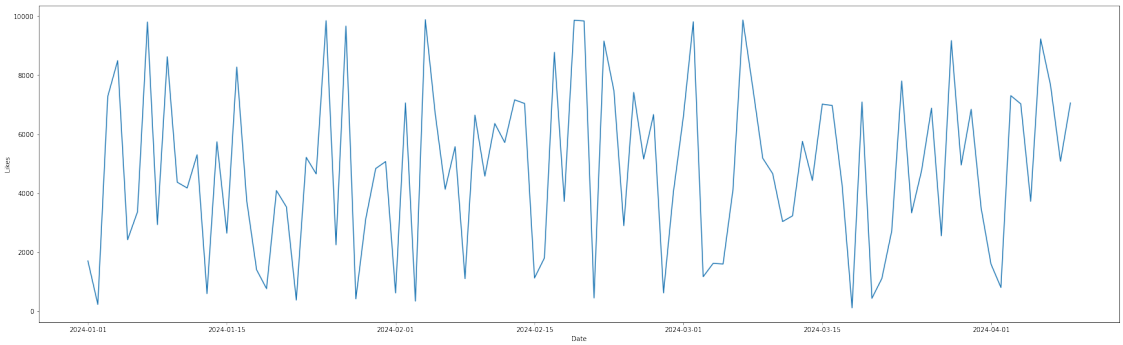
[10]:
```
print(df['Likes'].describe())
plt.figure(figsize=(9, 8))
sns.distplot(df['Likes'], color='g', bins=100, hist_kws={'alpha': 0.4})
```

```
count     100.000000
mean     4844.850000
std      2896.393715
min       113.000000
25%      2621.750000
50%      4709.500000
75%      7057.500000
max      9881.000000
Name: Likes, dtype: float64
```

[10]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd090e62ed0>

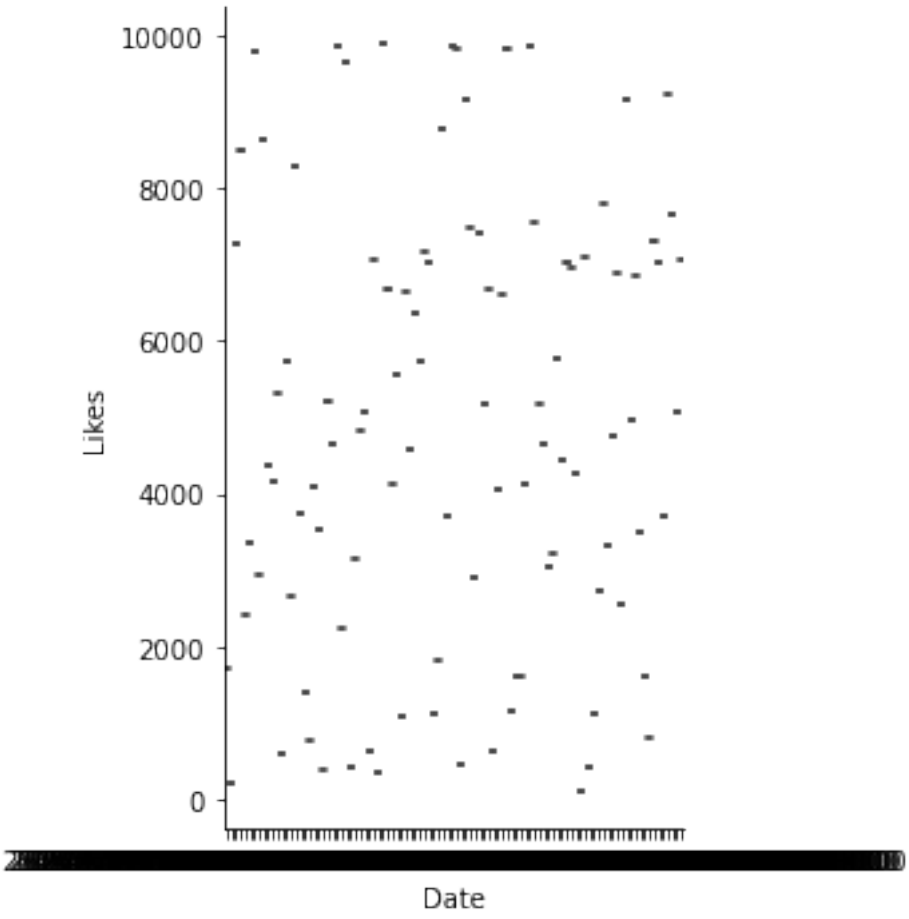


[11]:
```
# Like vs. Cat vs. Date
plt.figure(figsize=(30, 9))
sns.lineplot(x='Date', y = 'Likes', data = df)
plt.show()
```

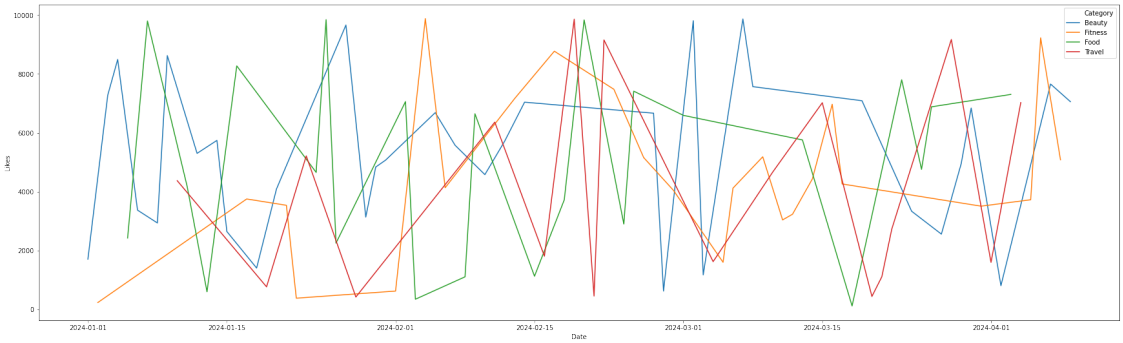


[12]:
```
plt.figure(figsize=(30, 9))
sns.catplot(x = 'Date', y = 'Likes', data = df, kind = "box")
plt.show()
```

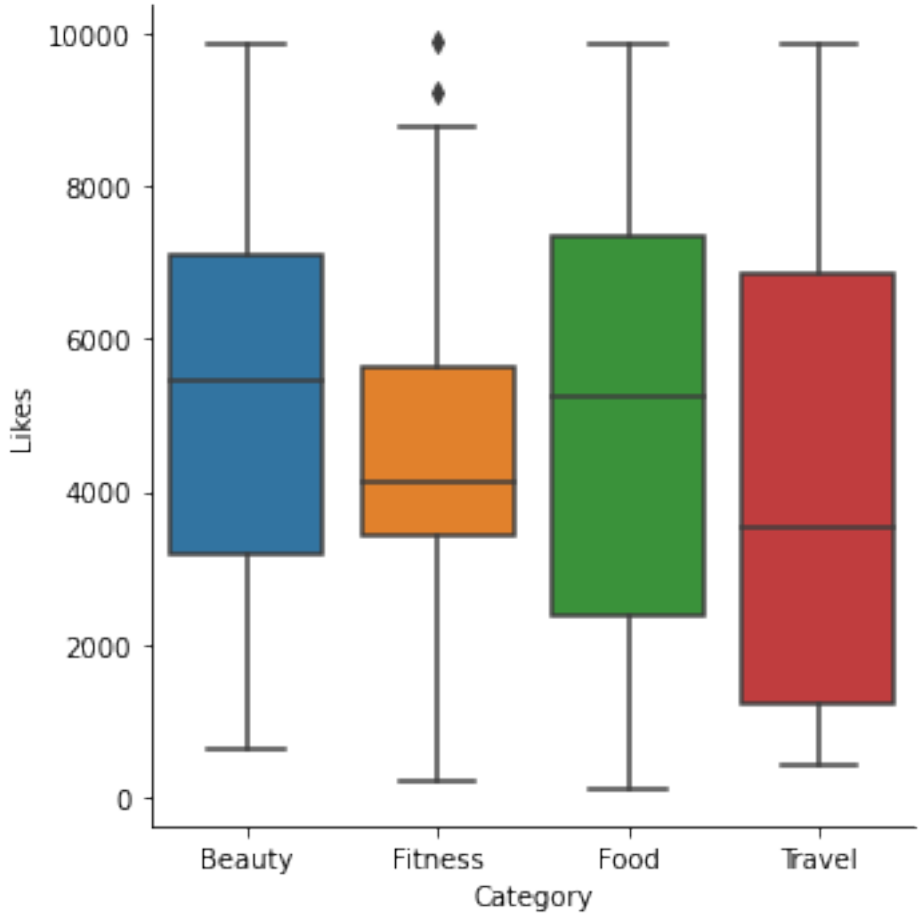
```
<Figure size 2160x648 with 0 Axes>
```



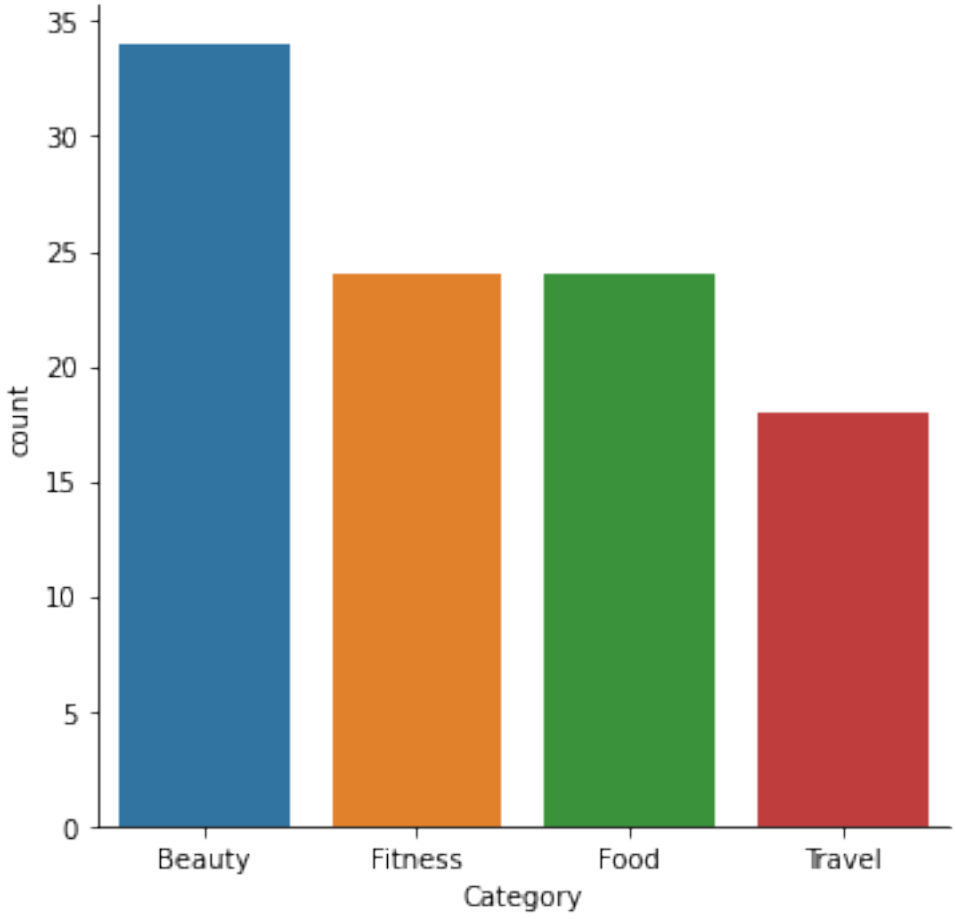
[13]:
```
# Like vs. Cat vs. Date
plt.figure(figsize=(30, 9))
sns.lineplot(x='Date', y = 'Likes', hue = 'Category', data = df)
plt.show()
```



[14]:
```
sns.catplot(x = "Category", y = 'Likes', data = df, kind = "box")      # create␣
↪a boxplot with the x axis as 'Category', and the y axis as 'Likes'
plt.show()
```



[15]:
```
sns.catplot(x = "Category", data = df, kind = "count")      # additional␣
↪visualization of counts of each category
plt.show()
```



[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: