

Name: Abhay Sharma
Div: C
RollNo: 28
Student ID: 20102065
Subject : SMA

▼ To perform social media action and location analysis on given dataset

```
from google.colab import files
import pandas as pd

print("Please upload your CSV file:")
uploaded = files.upload()

file_name = list(uploaded.keys())[0]

df = pd.read_csv(file_name)

df['brand_name'] = df['productTitle'].str.split().str[0]

df['reviewDescription'] = df['reviewDescription'] + " @" + df['brand_name']

df.head()
```

)- 143753 bytes, last modified: 27/03/2024 - 100% done
!.csv to flipkart_review_data_2022_02 (1).csv

productTitle	productPrice	averageRating	reviewTitle	reviewDescription	reviewAuthor	reviewAt	reviewLike
LG 108 cm (43 inch) Ultra HD (4K) LED Smart TV	₹36,499	4.4	Excellent	Very good product @LG	palakollu komali	6 months ago	
LG 108 cm (43 inch) Ultra HD (4K) LED Smart TV	₹36,499	4.4	Really Nice	Nice and super picture quality.and sound also ...	Yogesh Virkar	3 months ago	
LG 108 cm (43 inch) Ultra HD (4K) LED Smart TV	₹36,499	4.4	Super!	This is excellent picture qualityUltimate soun...	Sudhir Kumar Chaudhary	3 months ago	
LG 108 cm (43 inch) Ultra HD (4K) LED Smart TV	₹36,499	4.4	Just wow!	Nice @LG	Flipkart Customer	3 months ago	
LG 108 cm (43 inch) Ultra HD (4K) LED Smart TV	₹36,499	4.4	Highly recommended	Don't go with the negative reviews. Just go fo...	Ashok Mandial	3 months ago	

Next steps: [View recommended plots](#)

```
import pandas as pd
import re
import matplotlib.pyplot as plt
import seaborn as sns

def extract_mentions(text):
    mentions = re.findall(r'@(\w+)', text)
    return mentions

df['Mentions'] = df['reviewDescription'].apply(extract_mentions)

all_mentions = [mention for sublist in df['Mentions'].tolist() for mention in sublist]

mentions_df = pd.DataFrame(all_mentions, columns=['Mention'])
mention_counts = mentions_df['Mention'].value_counts().reset_index()
mention_counts.columns = ['Mention', 'Frequency']

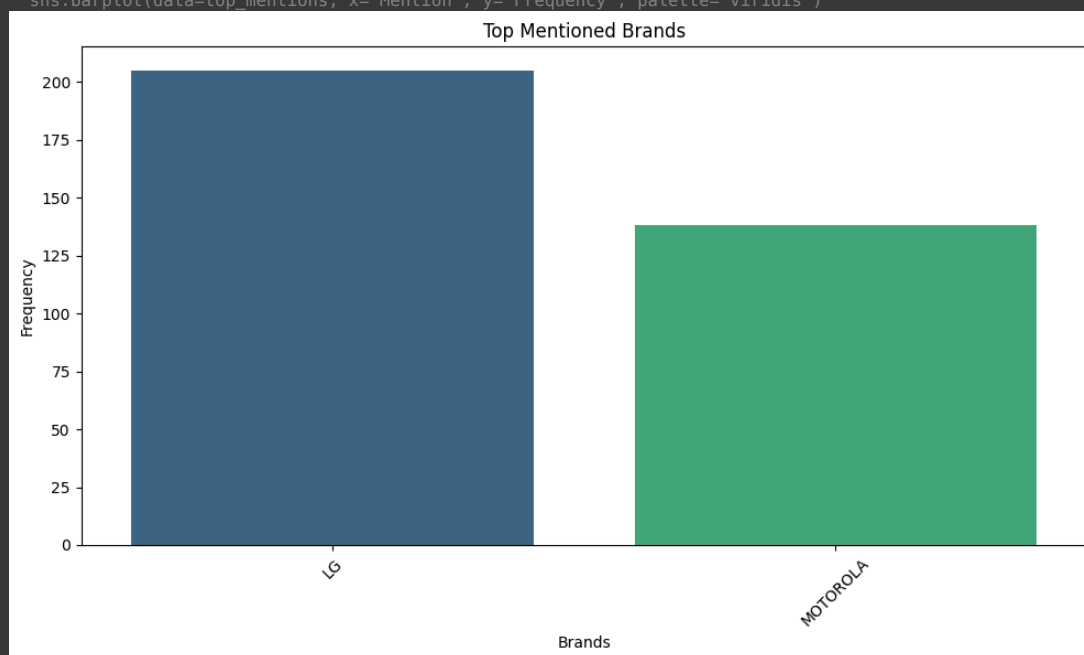
top_mentions = mention_counts.head(5)
plt.figure(figsize=(10, 6))
sns.barplot(data=top_mentions, x='Mention', y='Frequency', palette='viridis')
plt.title('Top Mentioned Brands')
plt.xlabel('Brands')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

print(mention_counts)
```

<ipython-input-4-bed43af9013a>:20: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` var

```
sns.barplot(data=top_mentions, x='Mention', y='Frequency', palette='viridis')
```



	Mention	Frequency
0	LG	205
1	MOTOROLA	138

```
import pandas as pd
import geopandas as gpd
from geopy.geocoders import Nominatim
import matplotlib.pyplot as plt
import seaborn as sns

geolocator = Nominatim(user_agent="my_geocoder")

def geocode_location(location_name):
    try:
        location = geolocator.geocode(location_name)
        if location:
            return location.latitude, location.longitude
        else:
            return None, None
    except Exception as e:
        print(f"Error geocoding {location_name}: {e}")
        return None, None

df['Latitude'], df['Longitude'] = zip(*df['reviewerLocation'].apply(geocode_location))

df = df.dropna(subset=['Latitude', 'Longitude'])

gdf = gpd.GeoDataFrame(df, geometry=gpd.points_from_xy(df['Longitude'], df['Latitude']))

world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
ax = world.plot(color='lightgrey', edgecolor='black', figsize=(10, 6))
gdf.plot(ax=ax, marker='o', color='blue', markersize=100, alpha=0.5)
plt.title('Product Reviews by Location')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.tight_layout()
plt.show()

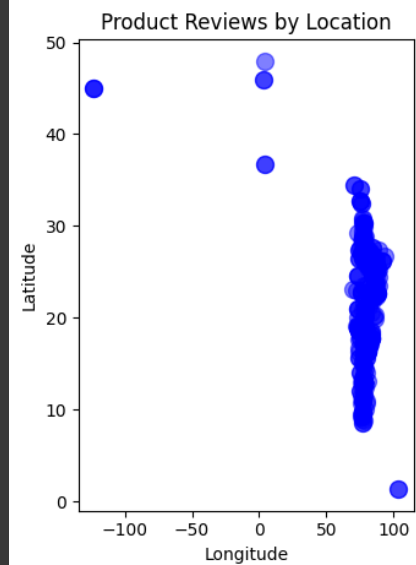
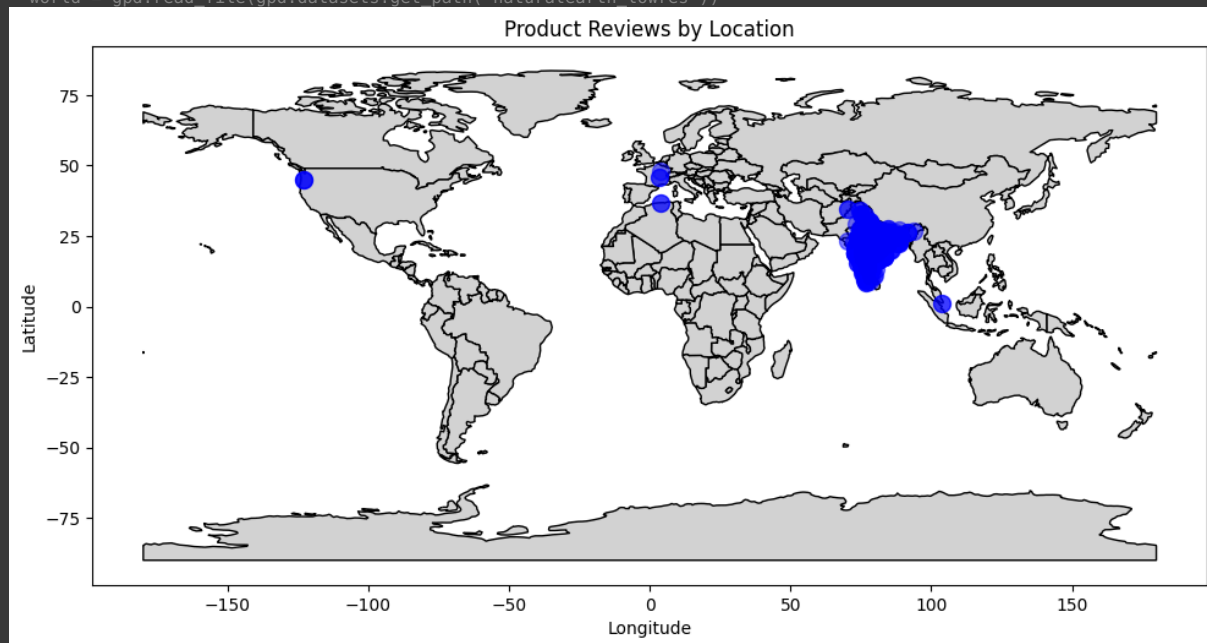
plt.subplot(1, 2, 2)
plt.scatter(df['Longitude'], df['Latitude'], color='blue', s=100, alpha=0.5)
plt.title('Product Reviews by Location')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.tight_layout()
plt.show()

print(df[['reviewerLocation', 'Latitude', 'Longitude']])
```

```

WARNING:urllib3.connectionpool:Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broke
WARNING:urllib3.connectionpool:Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broke
WARNING:urllib3.connectionpool:Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broke
WARNING:urllib3.connectionpool:Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broke
WARNING:urllib3.connectionpool:Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broke
WARNING:urllib3.connectionpool:Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broke
WARNING:urllib3.connectionpool:Retrying (Retry(total=0, connect=None, read=None, redirect=None, status=None)) after connection broke
WARNING:urllib3.connectionpool:Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broke
<ipython-input-8-be70d1a2ce35>:28: FutureWarning: The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0.
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))

```



	reviewerLocation	Latitude	Longitude
0	Tenali	16.237773	80.646422
1	Mumbai	19.078545	72.878176
2	Haridwar	29.938447	78.145298
3	Allahabad	25.438130	81.833800
4	Hamirpur District	25.750000	80.000000
..
338	Kottayam District	9.628738	76.645533
339	Kozhikode District	11.465594	75.891940
340	Jammu	32.718561	74.858092
341	Jabalpur	23.160894	79.949770
342	Balangir District	20.609518	83.167303

[333 rows x 3 columns]

```

import pandas as pd
from geopy.geocoders import Nominatim
import matplotlib.pyplot as plt
import seaborn as sns

# Load your DataFrame with 'reviewerLocation' column
# Assuming you have your DataFrame loaded as df with 'reviewerLocation' column

# Initialize geocoder
geolocator = Nominatim(user_agent="my_geocoder")

```

```
# Function to geocode location names into coordinates
def geocode_location(location_name):
    try:
        location = geolocator.geocode(location_name)
        if location:
            return location.latitude, location.longitude
        else:
            return None, None
    except Exception as e:
        print(f"Error geocoding {location_name}: {e}")
        return None, None

# Geocode reviewerLocation column
df['Latitude'], df['Longitude'] = zip(*df['reviewerLocation'].apply(geocode_location))

# Filter out rows with missing coordinates
df = df.dropna(subset=['Latitude', 'Longitude'])

# Count the number of reviews per location
location_counts = df['reviewerLocation'].value_counts().reset_index()
location_counts.columns = ['Location', 'Review Count']

# Plotting reviews on a bar chart
fig, ax = plt.subplots(1, 2, figsize=(14, 6))

sns.barplot(data=location_counts, x='Location', y='Review Count', palette='viridis', ax=ax[0])
ax[0].set_title('Number of Reviews per Location')
ax[0].set_xlabel('Location')
ax[0].set_ylabel('Review Count')
ax[0].tick_params(axis='x', rotation=45)

sns.barplot(data=location_counts, x='Location', y='Review Count', palette='viridis', ax=ax[1])
ax[1].set_title('Number of Reviews per Location')
ax[1].set_xlabel('Location')
ax[1].set_ylabel('Review Count')
ax[1].tick_params(axis='x', rotation=45)

plt.tight_layout()
plt.show()

# Display the DataFrame with review counts for each location
print(location_counts)
```

WARNING:urllib3.connectionpool:Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broke