

Socail media Analysis

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

df = pd.read_csv(r"C:\Users\Viren Rajendra Gamre\OneDrive\Desktop\
social media sentiments\archive\sentimentdataset.csv")

df.head
```

<bound method NDFrame.head of

	Text	Sentiment	
0	Enjoying a beautiful day at the park!	...	Positive
1	Traffic was terrible this morning.	...	Negative
2	Just finished an amazing workout! 🏋️	...	Positive
3	Excited about the upcoming weekend getaway!	...	Positive
4	Trying out a new recipe for dinner tonight.	...	Neutral
...
727	Collaborating on a science project that receiv...	...	Happy
728	Attending a surprise birthday party organized	Happy
729	Successfully fundraising for a school charity	Happy
730	Participating in a multicultural festival, cel...	...	Happy
731	Organizing a virtual talent show during challe...	...	Happy

	Platform	Timestamp	User
0	Twitter	15-01-2023 12:30	User123
1	Twitter	15-01-2023 08:45	CommuterX
2	Instagram	15-01-2023 15:45	FitnessFan
3	Facebook	15-01-2023 18:20	AdventureX
4	Instagram	15-01-2023 19:55	ChefCook
...
...
727	Facebook	18-08-2017 18:20	ScienceProjectSuccessHighSchool
728	Instagram	22-06-2018 14:15	BirthdayPartyJoyHighSchool
729	Twitter	05-04-2019 17:30	CharityFundraisingTriumphHighSchool
730	Facebook	29-02-2020 20:45	MulticulturalFestivalJoyHighSchool

731 15-11-2020 15:15 VirtualTalentShowSuccessHighSchool
Instagram

	Hashtags	Retweets	Likes	\
0	#Nature #Park	15	30	
1	#Traffic #Morning	5	10	
2	#Fitness #Workout	20	40	
3	#Travel #Adventure	8	15	
4	#Cooking #Food	12	25	
...	
727	#ScienceFairWinner #HighSchoolScience	20	39	
728	#SurpriseCelebration #HighSchoolFriendship	25	48	
729	#CommunityGiving #HighSchoolPhilanthropy	22	42	
730	#CulturalCelebration #HighSchoolUnity	21	43	
731	#VirtualEntertainment #HighSchoolPositivity	24	47	

	Country	Year	Month	Day	Hour
0	USA	2023	1	15	12
1	Canada	2023	1	15	8
2	USA	2023	1	15	15
3	UK	2023	1	15	18
4	Australia	2023	1	15	19
...
727	UK	2017	8	18	18
728	USA	2018	6	22	14
729	Canada	2019	4	5	17
730	UK	2020	2	29	20
731	USA	2020	11	15	15

[732 rows x 13 columns]>

df = df.drop_duplicates()

df.head()

	Text	Sentiment	\
0	Enjoying a beautiful day at the park!	...	Positive
1	Traffic was terrible this morning.	...	Negative
2	Just finished an amazing workout! 🏋️	...	Positive
3	Excited about the upcoming weekend getaway!	...	Positive
4	Trying out a new recipe for dinner tonight.	...	Neutral

	Timestamp	User	Platform	\
0	15-01-2023 12:30	User123	Twitter	
1	15-01-2023 08:45	CommuterX	Twitter	
2	15-01-2023 15:45	FitnessFan	Instagram	
3	15-01-2023 18:20	AdventureX	Facebook	
4	15-01-2023 19:55	ChefCook	Instagram	

Hashtags Retweets Likes

Country \			
0	#Nature #Park	15	30 USA
1	#Traffic #Morning	5	10
Canada			
2	#Fitness #Workout	20	40 USA
3	#Travel #Adventure	8	15 UK
4	#Cooking #Food	12	25
Australia			

	Year	Month	Day	Hour
0	2023	1	15	12
1	2023	1	15	8
2	2023	1	15	15
3	2023	1	15	18
4	2023	1	15	19

```
df["User"].count()
```

```
712
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 712 entries, 0 to 731
```

```
Data columns (total 13 columns):
```

#	Column	Non-Null Count	Dtype
0	Text	712 non-null	object
1	Sentiment	712 non-null	object
2	Timestamp	712 non-null	object
3	User	712 non-null	object
4	Platform	712 non-null	object
5	Hashtags	712 non-null	object
6	Retweets	712 non-null	int64
7	Likes	712 non-null	int64
8	Country	712 non-null	object
9	Year	712 non-null	int64
10	Month	712 non-null	int64
11	Day	712 non-null	int64
12	Hour	712 non-null	int64

```
dtypes: int64(6), object(7)
```

```
memory usage: 77.9+ KB
```

```
df["Timestamp"] = pd.to_datetime(df["Timestamp"])
```

```
C:\Users\Viren Rajendra Gamre\AppData\Local\Temp\
```

```
ipykernel_8740\2422418444.py:1: UserWarning: Parsing dates in %d-%m-%Y %H:%M format when dayfirst=False (the default) was specified. Pass
```

```
`dayfirst=True` or specify a format to silence this warning.  
df["Timestamp"] = pd.to_datetime(df["Timestamp"])  
C:\Users\Viren Rajendra Gamre\AppData\Local\Temp\  
ipykernel_8740\2422418444.py:1: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation:

https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df["Timestamp"] = pd.to_datetime(df["Timestamp"])
```

```
df.isnull().sum()
```

Text	0
Sentiment	0
Timestamp	0
User	0
Platform	0
Hashtags	0
Retweets	0
Likes	0
Country	0
Year	0
Month	0
Day	0
Hour	0

dtype: int64

```
df.nunique()
```

Text	707
Sentiment	279
Timestamp	683
User	685
Platform	4
Hashtags	697
Retweets	26
Likes	38
Country	115
Year	14
Month	12
Day	31
Hour	22

dtype: int64

#lets classify the sentiments into group so we can get a better idea for futhur analysis

```
categories = {  
    "Happiness": [  
        ]
```

```

        "Joy", "Love", "Amusement", "Admiration", "Excitement",
        "Contentment",
        "Serenity", "Gratitude", "Enthusiasm", "Playful",
        "Reflection",
        "Creativity", "Inspiration", "Celebration", "Charm",
        "Positivity",
        "Affection", "Adoration", "Kind", "Elation", "Euphoria",
        "Tenderness",
        "Radiance", "Rejuvenation", "Coziness", "Freedom", "Grandeur",

        "Exploration", "Harmony", "Zest"
    ],
    "Sadness": [
        "Despair", "Grief", "Loneliness", "Regret", "Melancholy",
        "Heartbreak",
        "Disappointed", "Nostalgia", "Numbness", "Ambivalence",
        "Isolation"
    ],
    "Anger": [
        "Resentment", "Frustration", "Bitterness", "Devastated",
        "Obstacles",
        "Pressure", "Dismissive"
    ],
    "Fear": [
        "Anxiety", "Intimidation", "Helplessness", "Overwhelmed"
    ],
    "Disgust": [
        "Disgust"
    ],
    "Surprise": [
        "Awe", "Surprise", "Curiosity", "Wonder", "Inspiration",
        "Enchantment"
    ],
    "Other Positive Emotions": [
        "Pride", "Hope", "Empowerment", "Compassion", "Determination",

        "Adventure", "Confidence", "Playful", "Reflection",
        "Celebration",
        "Positivity", "Fulfillment", "Reverence", "Satisfaction",
        "Charm",
        "Sympathy", "Gratitude", "Vibrancy"
    ],
    "Other Negative Emotions": [
        "Jealousy", "Indifference", "Confusion", "Boredom",
        "Devastated"
    ]
}

def map_to_category(sentiment):
    sentiment = sentiment.strip().lower()

```

```

for category, sentiments in categories.items():
    for s in sentiments:
        if sentiment == s.lower():
            return category
return "Unknown"

```

```
df['Category'] = df['Sentiment'].apply(map_to_category)
```

C:\Users\Viren Rajendra Gamre\AppData\Local\Temp\ipykernel_8740\435040092.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:

https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df['Category'] = df['Sentiment'].apply(map_to_category)
```

df

		Text	Sentiment \
0	Enjoying a beautiful day at the park!	...	Positive
1	Traffic was terrible this morning.	...	Negative
2	Just finished an amazing workout! 🏋️	...	Positive
3	Excited about the upcoming weekend getaway!	...	Positive
4	Trying out a new recipe for dinner tonight.	...	Neutral
...
727	Collaborating on a science project that receiv...	...	Happy
728	Attending a surprise birthday party organized	Happy
729	Successfully fundraising for a school charity	Happy
730	Participating in a multicultural festival, cel...	...	Happy
731	Organizing a virtual talent show during challe...	...	Happy
	Timestamp		User
Platform \			
0	2023-01-15 12:30:00		User123
Twitter			
1	2023-01-15 08:45:00		CommuterX
Twitter			
2	2023-01-15 15:45:00		FitnessFan
Instagram			
3	2023-01-15 18:20:00		AdventureX
Facebook			
4	2023-01-15 19:55:00		ChefCook
Instagram			
...
...			
727	2017-08-18 18:20:00		ScienceProjectSuccessHighSchool
Facebook			
728	2018-06-22 14:15:00		BirthdayPartyJoyHighSchool

```

Instagram
729 2019-04-05 17:30:00 CharityFundraisingTriumphHighSchool
Twitter
730 2020-02-29 20:45:00 MulticulturalFestivalJoyHighSchool
Facebook
731 2020-11-15 15:15:00 VirtualTalentShowSuccessHighSchool
Instagram

```

	Hashtags	Retweets	Likes	\
0	#Nature #Park	15	30	
1	#Traffic #Morning	5	10	
2	#Fitness #Workout	20	40	
3	#Travel #Adventure	8	15	
4	#Cooking #Food	12	25	
...		
727	#ScienceFairWinner #HighSchoolScience	20	39	
728	#SurpriseCelebration #HighSchoolFriendship	25	48	
729	#CommunityGiving #HighSchoolPhilanthropy	22	42	
730	#CulturalCelebration #HighSchoolUnity	21	43	
731	#VirtualEntertainment #HighSchoolPositivity	24	47	

	Country	Year	Month	Day	Hour	Category
0	USA	2023	1	15	12	Unknown
1	Canada	2023	1	15	8	Unknown
2	USA	2023	1	15	15	Unknown
3	UK	2023	1	15	18	Unknown
4	Australia	2023	1	15	19	Unknown
...
727	UK	2017	8	18	18	Unknown
728	USA	2018	6	22	14	Unknown
729	Canada	2019	4	5	17	Unknown
730	UK	2020	2	29	20	Unknown
731	USA	2020	11	15	15	Unknown

```
[712 rows x 14 columns]
```

```
df.nunique()
```

Text	707
Sentiment	279
Timestamp	683
User	685
Platform	4
Hashtags	697
Retweets	26
Likes	38
Country	115
Year	14
Month	12
Day	31

```
Hour          22
Category      9
dtype: int64
```

```
# lets categories more so we can identify a pattern
```

```
countries_by_continent = {
    "North America": [
        "USA", "Canada", "Jamaica", "Mexico", "Colombia"
    ],
    "South America": [
        "Brazil", "Peru", "Argentina", "Chile"
    ],
    "Europe": [
        "UK", "France", "Germany", "Italy", "Netherlands", "Spain",
        "Portugal",
        "Switzerland", "Austria", "Belgium", "Denmark", "Czech
        Republic", "Sweden",
        "Greece", "Ireland", "Norway", "Scotland"
    ],
    "Asia": [
        "India", "Japan", "China", "Cambodia", "Thailand", "Jordan",
        "Maldives"
    ],
    "Africa": [
        "South Africa", "Kenya"
    ],
    "Australia and Oceania": [
        "Australia", "New Zealand"
    ],
    "Antarctica": []
}
```

```
df
```

		Text	Sentiment \
0	Enjoying a beautiful day at the park!	...	Positive
1	Traffic was terrible this morning.	...	Negative
2	Just finished an amazing workout! 🏋️	...	Positive
3	Excited about the upcoming weekend getaway!	...	Positive
4	Trying out a new recipe for dinner tonight.	...	Neutral
...	
727	Collaborating on a science project that receiv...		Happy
728	Attending a surprise birthday party organized ...		Happy
729	Successfully fundraising for a school charity ...		Happy
730	Participating in a multicultural festival, cel...		Happy
731	Organizing a virtual talent show during challe...		Happy

	Timestamp	User
Platform \		

0	2023-01-15 12:30:00	User123
Twitter		
1	2023-01-15 08:45:00	CommuterX
Twitter		
2	2023-01-15 15:45:00	FitnessFan
Instagram		
3	2023-01-15 18:20:00	AdventureX
Facebook		
4	2023-01-15 19:55:00	ChefCook
Instagram		
..
...		
727	2017-08-18 18:20:00	ScienceProjectSuccessHighSchool
Facebook		
728	2018-06-22 14:15:00	BirthdayPartyJoyHighSchool
Instagram		
729	2019-04-05 17:30:00	CharityFundraisingTriumphHighSchool
Twitter		
730	2020-02-29 20:45:00	MulticulturalFestivalJoyHighSchool
Facebook		
731	2020-11-15 15:15:00	VirtualTalentShowSuccessHighSchool
Instagram		

	Hashtags	Retweets	Likes	\
0	#Nature #Park	15	30	
1	#Traffic #Morning	5	10	
2	#Fitness #Workout	20	40	
3	#Travel #Adventure	8	15	
4	#Cooking #Food	12	25	
..	
727	#ScienceFairWinner #HighSchoolScience	20	39	
728	#SurpriseCelebration #HighSchoolFriendship	25	48	
729	#CommunityGiving #HighSchoolPhilanthropy	22	42	
730	#CulturalCelebration #HighSchoolUnity	21	43	
731	#VirtualEntertainment #HighSchoolPositivity	24	47	

	Country	Year	Month	Day	Hour	Category
0	USA	2023	1	15	12	Unknown
1	Canada	2023	1	15	8	Unknown
2	USA	2023	1	15	15	Unknown
3	UK	2023	1	15	18	Unknown
4	Australia	2023	1	15	19	Unknown
..
727	UK	2017	8	18	18	Unknown
728	USA	2018	6	22	14	Unknown
729	Canada	2019	4	5	17	Unknown
730	UK	2020	2	29	20	Unknown
731	USA	2020	11	15	15	Unknown

[712 rows x 14 columns]

```
def map_to_continent(country):
    country = country.strip().lower()
    for continent, countries in countries_by_continent.items():
        for c in countries:
            if country == c.lower():
                return continent
    return "Unknown"
```

```
df['Continents'] = df['Country'].apply(map_to_continent)
```

C:\Users\Viren Rajendra Gamre\AppData\Local\Temp\ipykernel_8740\1263603132.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:

https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df['Continents'] = df['Country'].apply(map_to_continent)
```

```
df
```

		Text	Sentiment \
0	Enjoying a beautiful day at the park!	...	Positive
1	Traffic was terrible this morning.	...	Negative
2	Just finished an amazing workout! 🏋️	...	Positive
3	Excited about the upcoming weekend getaway!	...	Positive
4	Trying out a new recipe for dinner tonight.	...	Neutral
..	
727	Collaborating on a science project that receiv...		Happy
728	Attending a surprise birthday party organized ...		Happy
729	Successfully fundraising for a school charity ...		Happy
730	Participating in a multicultural festival, cel...		Happy
731	Organizing a virtual talent show during challe...		Happy

	Timestamp	User
Platform \		
0	2023-01-15 12:30:00	User123
Twitter		
1	2023-01-15 08:45:00	CommuterX
Twitter		
2	2023-01-15 15:45:00	FitnessFan
Instagram		
3	2023-01-15 18:20:00	AdventureX
Facebook		
4	2023-01-15 19:55:00	ChefCook
Instagram		
..
...		

727	2017-08-18 18:20:00	ScienceProjectSuccessHighSchool
	Facebook	
728	2018-06-22 14:15:00	BirthdayPartyJoyHighSchool
	Instagram	
729	2019-04-05 17:30:00	CharityFundraisingTriumphHighSchool
	Twitter	
730	2020-02-29 20:45:00	MulticulturalFestivalJoyHighSchool
	Facebook	
731	2020-11-15 15:15:00	VirtualTalentShowSuccessHighSchool
	Instagram	

	Hashtags	Retweets	Likes	\
0	#Nature #Park	15	30	
1	#Traffic #Morning	5	10	
2	#Fitness #Workout	20	40	
3	#Travel #Adventure	8	15	
4	#Cooking #Food	12	25	
..	
727	#ScienceFairWinner #HighSchoolScience	20	39	
728	#SurpriseCelebration #HighSchoolFriendship	25	48	
729	#CommunityGiving #HighSchoolPhilanthropy	22	42	
730	#CulturalCelebration #HighSchoolUnity	21	43	
731	#VirtualEntertainment #HighSchoolPositivity	24	47	

	Country	Year	Month	Day	Hour	Category	
Continents							
0	USA	2023	1	15	12	Unknown	North
America							
1	Canada	2023	1	15	8	Unknown	North
America							
2	USA	2023	1	15	15	Unknown	North
America							
3	UK	2023	1	15	18	Unknown	
Europe							
4	Australia	2023	1	15	19	Unknown	Australia and
Oceania							
..	
...							
727	UK	2017	8	18	18	Unknown	
Europe							
728	USA	2018	6	22	14	Unknown	North
America							
729	Canada	2019	4	5	17	Unknown	North
America							
730	UK	2020	2	29	20	Unknown	
Europe							
731	USA	2020	11	15	15	Unknown	North
America							

[712 rows x 15 columns]

```
df.nunique()
```

```
Text          707
Sentiment     279
Timestamp     683
User          685
Platform       4
Hashtags      697
Retweets      26
Likes         38
Country       115
Year          14
Month         12
Day           31
Hour          22
Category       9
Continents     6
dtype: int64
```

```
# Assuming df["Hashtags"] contains hashtags and df["continents"]
contains the continent names
```

```
# Function to calculate top 5 hashtags for a specific group
```

```
def get_top_hashtags(group):
    hashtags = group.str.split().explode().tolist() # Split and
flatten hashtags
    hashtag_counts = Counter(hashtags) # Count occurrences
    return hashtag_counts.most_common(5) # Return top 5 hashtags
```

```
# Group by continent and calculate top hashtags for each group
```

```
top_hashtags_by_continent = df.groupby("Continents")
["Hashtags"].apply(get_top_hashtags)
```

```
# Display results
```

```
for continent, top_hashtags in top_hashtags_by_continent.items():
    print(f"\nTop 5 hashtags in {continent}:")
    for hashtag, count in top_hashtags:
        print(f"{hashtag}: {count}")
```

```
Top 5 hashtags in Africa:
```

```
#Wonder: 2
#StargazingAdventure: 2
#Despair: 1
#AbyssOfHeartache: 1
#Isolation: 1
```

```
Top 5 hashtags in Asia:
```

```
#Gratitude: 5
#Injustice: 3
```

```
#Contentment: 3
#Fulfillment: 3
#Curiosity: 3
```

Top 5 hashtags in Australia and Oceania:

```
#Acceptance: 4
#Confusion: 3
#Love: 2
#SpreadLove: 2
#Awe: 2
```

Top 5 hashtags in Europe:

```
#Serenity: 5
#Inspiration: 5
#Confusion: 4
#Regret: 4
#Nostalgia: 4
```

Top 5 hashtags in North America:

```
#Serenity: 8
#Excitement: 6
#Elation: 6
#Enthusiasm: 5
#Joy: 5
```

Top 5 hashtags in South America:

```
#Enchantment: 2
#Gratitude: 1
#LifeLessons: 1
#CosmicFireflies: 1
#FireflyField: 1
```

```
n_continents
```

```
6
```

```
5//3
```

```
1
```

```
n_continents = len(top_hashtags_by_continent)
```

```
ncols = 3 # Number of columns in the grid
```

```
nrows = (n_continents + ncols - 1) // ncols # Calculate rows needed
```

```
# Initialize a figure for subplots
```

```
fig, axes = plt.subplots(nrows=nrows, ncols=ncols, figsize=(15, 5 *
nrows))
```

```
# Flatten axes for easy iteration (if there are multiple rows and
columns)
```

```
axes = axes.flatten()
```

```

# Plot each continent's top hashtags
for ax, (continent, top_hashtags) in zip(axes,
top_hashtags_by_continent.items()):
    hashtags, counts = zip(*top_hashtags) # Unpack hashtags and their
counts
    sns.barplot(x=counts, y=hashtags, ax=ax, palette="viridis")
    ax.set_title(f"Top 5 Hashtags in {continent}", fontsize=14)
    ax.set_xlabel("Count")
    ax.set_ylabel("Hashtags")

# Remove any unused subplots
for ax in axes[len(top_hashtags_by_continent):]:
    ax.remove()

# Adjust layout for better visualization
plt.tight_layout()
plt.show()

```

C:\Users\Viren Rajendra Gamre\AppData\Local\Temp\ipykernel_8740\3144524082.py:14: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=counts, y=hashtags, ax=ax, palette="viridis")
```

C:\Users\Viren Rajendra Gamre\AppData\Local\Temp\ipykernel_8740\3144524082.py:14: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=counts, y=hashtags, ax=ax, palette="viridis")
```

C:\Users\Viren Rajendra Gamre\AppData\Local\Temp\ipykernel_8740\3144524082.py:14: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=counts, y=hashtags, ax=ax, palette="viridis")
```

C:\Users\Viren Rajendra Gamre\AppData\Local\Temp\ipykernel_8740\3144524082.py:14: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=counts, y=hashtags, ax=ax, palette="viridis")
```

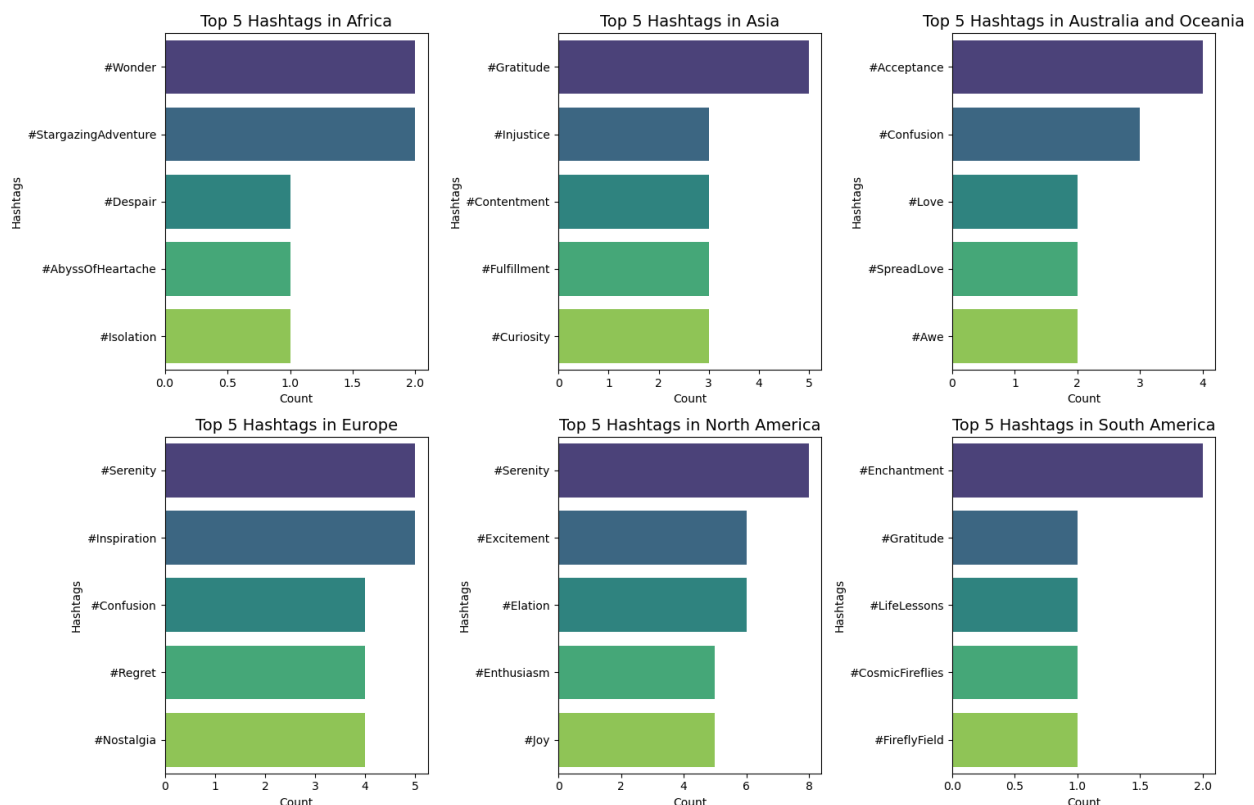
```
C:\Users\Viren Rajendra Gamre\AppData\Local\Temp\
ipykernel_8740\3144524082.py:14: FutureWarning:
```

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=counts, y=hashtags, ax=ax, palette="viridis")
C:\Users\Viren Rajendra Gamre\AppData\Local\Temp\
ipykernel_8740\3144524082.py:14: FutureWarning:
```

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=counts, y=hashtags, ax=ax, palette="viridis")
```



#finding any corr in category and time

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 712 entries, 0 to 731
```

```
Data columns (total 15 columns):
```

```
#   Column      Non-Null Count  Dtype
```

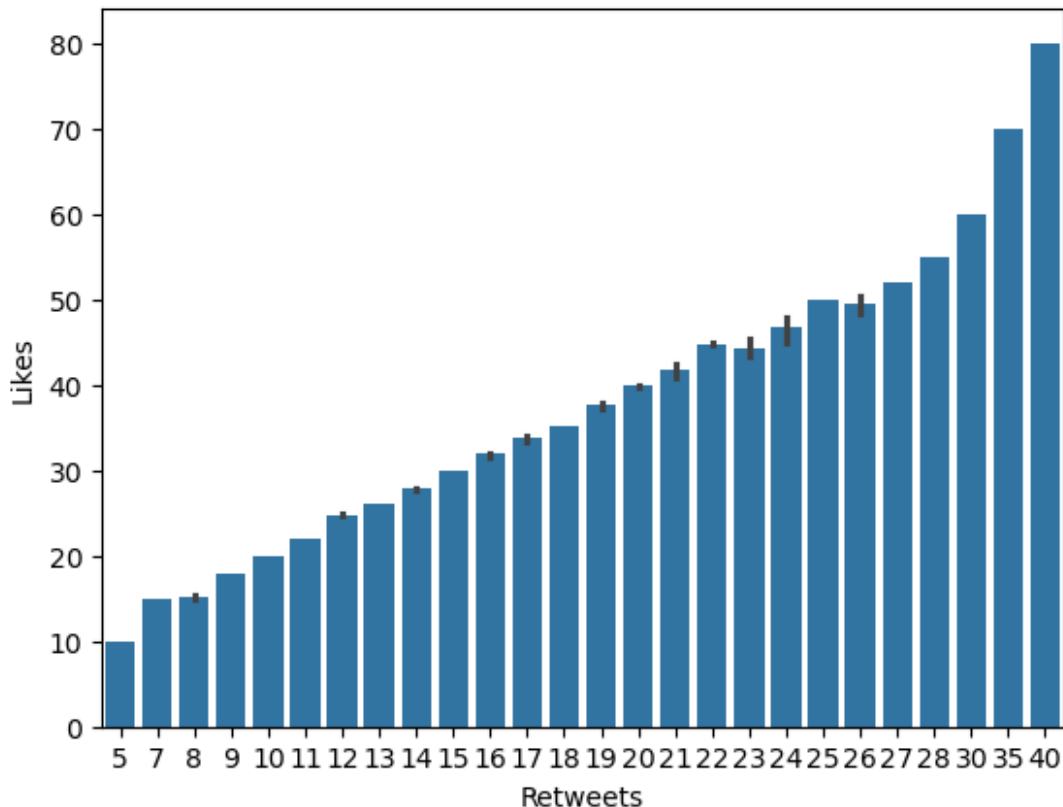
```
---
0   Text      712 non-null object
1   Sentiment 712 non-null object
2   Timestamp 712 non-null datetime64[ns]
3   User      712 non-null object
4   Platform  712 non-null object
5   Hashtags  712 non-null object
6   Retweets  712 non-null int64
7   Likes     712 non-null int64
8   Country   712 non-null object
9   Year      712 non-null int64
10  Month     712 non-null int64
11  Day       712 non-null int64
12  Hour      712 non-null int64
13  Category  712 non-null object
14  Continents 712 non-null object
dtypes: datetime64[ns](1), int64(6), object(8)
memory usage: 89.0+ KB
```

```
df["Retweets"].corr(df["Likes"])
```

```
0.9984750254879579
```

```
sns.barplot(x=df["Retweets"], y= df["Likes"], data= df, )
```

```
<Axes: xlabel='Retweets', ylabel='Likes'>
```

here we can see the the higher the retweets the duble the likes

```
df["Hour"].corr(df["Day"])
```

```
0.040422404173935125
```

```
df["Platform"]=df["Platform"].str.strip()
```

C:\Users\Viren Rajendra Gamre\AppData\Local\Temp\ipykernel_8740\344770620.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:

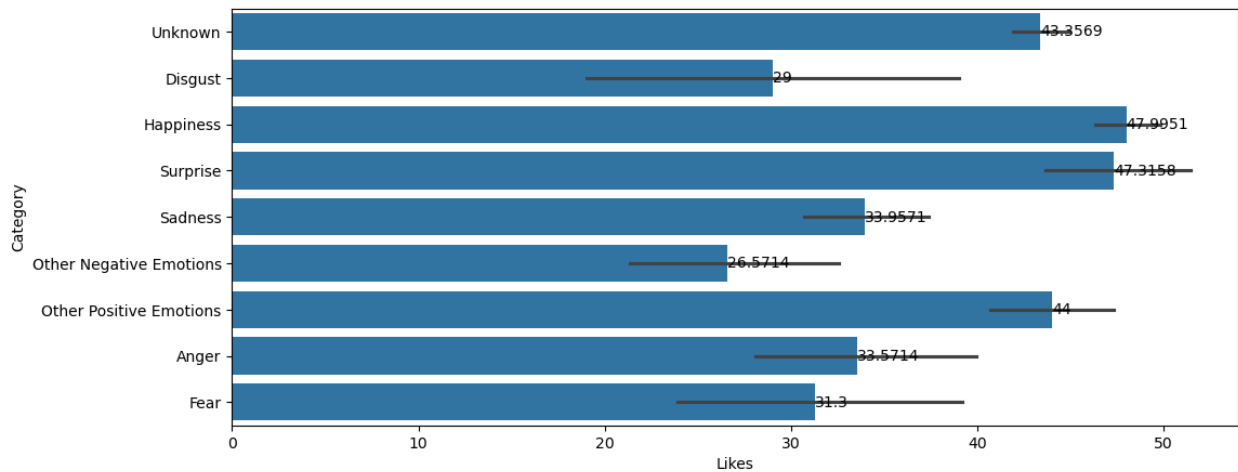
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df["Platform"]=df["Platform"].str.strip()
```

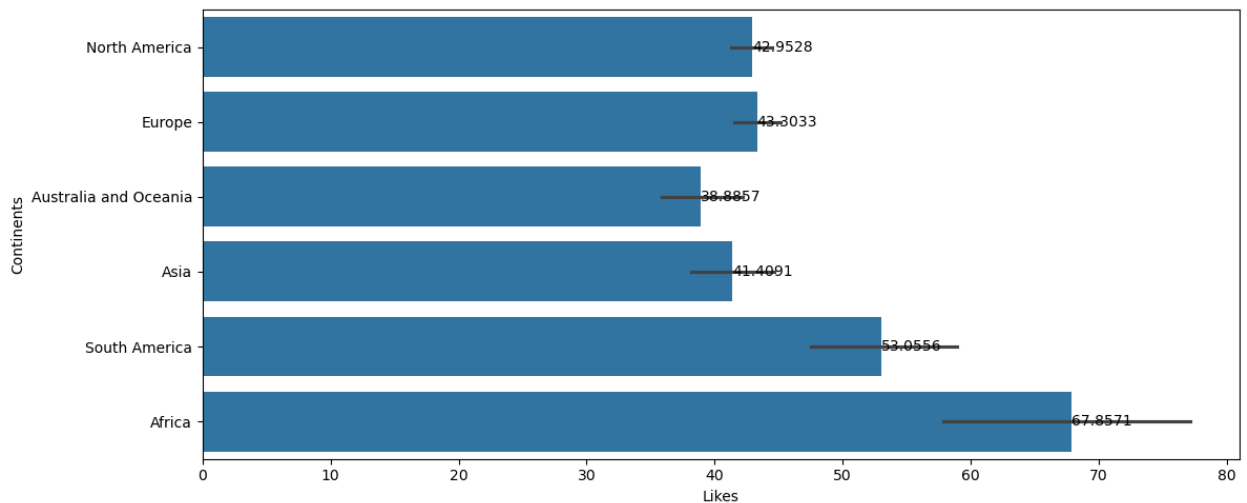
```
plt.figure(figsize=(12,5))
ax = sns.barplot(y=df["Category"], x=df["Likes"])
plt.bar_label(ax.containers[0])
```

```
[Text(0, 0, '43.3569'),
 Text(0, 0, '29'),
```

```
Text(0, 0, '47.9951'),
Text(0, 0, '47.3158'),
Text(0, 0, '33.9571'),
Text(0, 0, '26.5714'),
Text(0, 0, '44'),
Text(0, 0, '33.5714'),
Text(0, 0, '31.3')]
```



```
plt.figure(figsize=(12,5))
ax = sns.barplot(y=df["Continents"], x=df["Likes"])
plt.bar_label(ax.containers[0])
plt.tight_layout()
```



#more exgagement is seen in africa with around 678571 as the figure express clear with the ratio of likes

```
#lets find the active user base in the continents.
```

```
df.groupby("Continents")["Retweets"].sum()
```

```
Continents
```

```
Africa                237
```

```
Asia                  1819
```

```
Australia and Oceania 1362
```

```
Europe                4592
```

```
North America         6852
```

```
South America          476
```

```
Name: Retweets, dtype: int64
```

```
# this shows the like ratio might be higer in african region but not  
america has more active partisepents on social media
```

```
df["time"] = df["Timestamp"].dt.strftime("%H-%M-%S")
```

```
C:\Users\Viren Rajendra Gamre\AppData\Local\Temp\
```

```
ipykernel_8740\2235344181.py:1: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation:
```

```
https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#  
returning-a-view-versus-a-copy
```

```
df["time"] = df["Timestamp"].dt.strftime("%H-%M-%S")
```

```
df.groupby("Category")["Hour"].mean()
```

```
Category
```

```
Anger                14.571429
```

```
Disgust               15.000000
```

```
Fear                 13.900000
```

```
Happiness            16.107317
```

```
Other Negative Emotions 15.190476
```

```
Other Positive Emotions 15.688889
```

```
Sadness              16.242857
```

```
Surprise             15.105263
```

```
Unknown              15.299663
```

```
Name: Hour, dtype: float64
```

```
from statistics import mode
```

```
df.groupby("Continents"),mode(df["Timestamp"])
```

```
(<pandas.core.groupby.generic.DataFrameGroupBy object at  
0x000002AEB66EBBF0>,
```

```
Timestamp('2019-04-05 17:30:00'))
```

```
mode(df["Hour"])
```

```
df.head()
```

		Text	Sentiment \
0	Enjoying a beautiful day at the park!	...	Positive
1	Traffic was terrible this morning.	...	Negative
2	Just finished an amazing workout! 🏋️	...	Positive
3	Excited about the upcoming weekend getaway!	...	Positive
4	Trying out a new recipe for dinner tonight.	...	Neutral

	Timestamp	User	Platform \
0	2023-01-15 12:30:00	User123	Twitter
1	2023-01-15 08:45:00	CommuterX	Twitter
2	2023-01-15 15:45:00	FitnessFan	Instagram
3	2023-01-15 18:20:00	AdventureX	Facebook
4	2023-01-15 19:55:00	ChefCook	Instagram

	Hashtags	Retweets	Likes	Country \
0	#Nature #Park	15	30	USA
1	#Traffic #Morning	5	10	Canada
2	#Fitness #Workout	20	40	USA
3	#Travel #Adventure	8	15	UK
4	#Cooking #Food	12	25	Australia

	Year	Month	Day	Hour	Category	Continents	time
0	2023	1	15	12	Unknown	North America	12-30-00
1	2023	1	15	8	Unknown	North America	08-45-00
2	2023	1	15	15	Unknown	North America	15-45-00
3	2023	1	15	18	Unknown	Europe	18-20-00
4	2023	1	15	19	Unknown	Australia and Oceania	19-55-00

```
df.info()
```

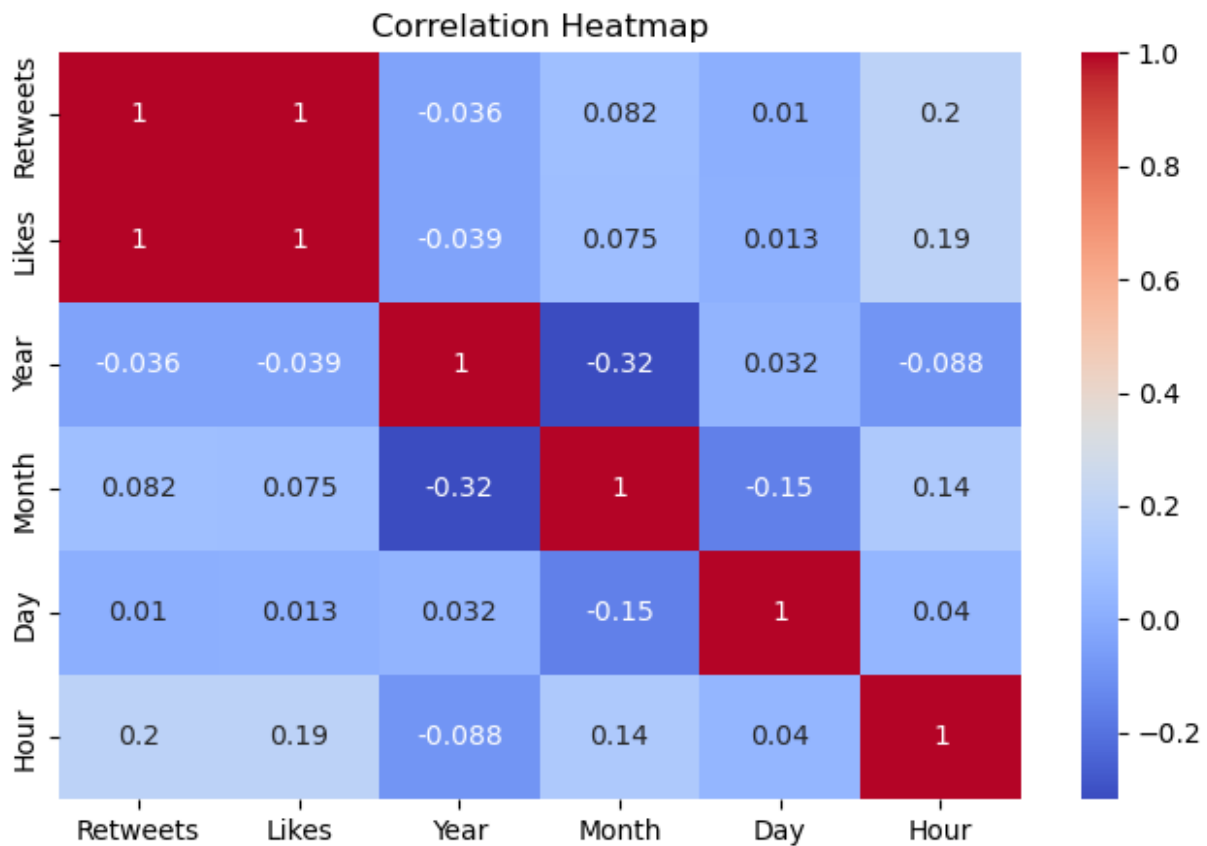
```
<class 'pandas.core.frame.DataFrame'>
Index: 712 entries, 0 to 731
Data columns (total 16 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Text            712 non-null   object
1   Sentiment       712 non-null   object
2   Timestamp       712 non-null   datetime64[ns]
3   User            712 non-null   object
4   Platform        712 non-null   object
5   Hashtags        712 non-null   object
6   Retweets        712 non-null   int64
7   Likes           712 non-null   int64
```

```

8   Country      712 non-null object
9   Year         712 non-null int64
10  Month        712 non-null int64
11  Day          712 non-null int64
12  Hour         712 non-null int64
13  Category     712 non-null object
14  Continents   712 non-null object
15  time         712 non-null object
dtypes: datetime64[ns](1), int64(6), object(9)
memory usage: 110.7+ KB

numeric_df = df.select_dtypes(include=[np.number])
plt.figure(figsize=(8, 5))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()

```



Summary

From this correlation heatmap, I can interpret the relationships between the variables in my dataset as follows:

1. Correlation Coefficient: Range: The correlation values range from -1 to 1:

1: A perfect positive correlation means that as one variable increases, the other also increases. -

1: A perfect negative correlation means that as one variable increases, the other decreases. 0: No correlation indicates the variables are independent. Color Map:

Red: Represents a high positive correlation. Blue: Indicates a negative correlation. Light Shades: Show values close to zero, indicating a weak or no correlation.

1. Key Observations I Can Make: Strong Positive Correlation:

Retweets and Likes have a perfect correlation of 1, meaning that more retweets are strongly associated with more likes. Weak or No Correlation:

Variables like Year, Month, Day, and Hour have very low correlation values with Likes and Retweets, which suggests minimal linear relationships. Negative Correlation:

Year has a negative correlation with Month (-0.32), which might mean there's an inverse relationship between these variables. Moderate Correlation:

Hour shows a moderate positive correlation with Retweets (0.2), which might suggest that the time of posting has some influence on engagement.

1. What This Means for Me: Focusing on Strong Relationships:

I should explore why Retweets and Likes are perfectly correlated—perhaps they represent similar engagement behavior or are directly linked. Understanding Low Correlations:

Variables like Day or Month might not significantly impact Likes or Retweets based on their weak correlations. However, I need to check for potential nonlinear relationships. Gaining Actionable Insights:

If my goal is to optimize engagement, I should look at patterns in variables with stronger correlations, like Hour or Retweets. Let me know if I should explore or refine any of these observations further!

You said: write it this way that the first person which is myself explaining to u ChatGPT said: ChatGPT Here's how I interpret this correlation heatmap:

1. Correlation Coefficient: The values in this heatmap range from -1 to 1. A value of 1 means a perfect positive correlation, where both variables increase together. A value of -1 means a perfect negative correlation, where one variable increases as the other decreases. A value of 0 means no correlation or independence between the variables.

The color map helps me visualize this: red indicates a strong positive correlation, blue indicates a negative correlation, and light shades represent weak or no correlation.

2. What I Observe: Strong Positive Correlation:

Retweets and Likes are perfectly correlated with a value of 1, meaning that the more retweets a post gets, the more likes it tends to receive. This makes sense because they're both engagement metrics. Weak or No Correlation:

Variables like Year, Month, Day, and Hour show very low correlations with Likes and Retweets (close to 0). This tells me there isn't a strong linear relationship between these variables and engagement metrics. Negative Correlation:

I notice a negative correlation of -0.32 between Year and Month. This might mean there's some inverse relationship, though I'd need to look closer to understand why. Moderate Correlation:

Hour has a moderate positive correlation with Retweets (0.2). This suggests that the time of day might have some influence on how many retweets a post gets.

1. How I Interpret This: I should focus on the strong correlation between Retweets and Likes since they appear to be directly linked. It could help me understand engagement trends better. For the weaker correlations (like Day or Month), I might want to explore whether there are nonlinear relationships or other factors at play that the correlation metric isn't capturing. If I'm trying to optimize engagement, I should pay attention to variables with moderate correlations like Hour to see if posting at certain times can improve results.

