Social Media Engagement Analysis

1. Import Libraries and Load Data

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('/Viral_Social_Media_Trends.csv')
print(df.head())
```

```
  Post_ID   Platform     Hashtag Content_Type     Region    Views   Likes  \
0  Post_1     TikTok  #Challenge        Video         UK  4163464  339431
1  Post_2  Instagram  #Education       Shorts      India  4155940  215240
2  Post_3    Twitter  #Challenge        Video     Brazil  3666211  327143
3  Post_4    YouTube  #Education       Shorts  Australia   917951  127125
4  Post_5     TikTok      #Dance         Post     Brazil    64866  171361

   Shares  Comments Engagement_Level
0   53135     19346             High
1   65860     27239           Medium
2   39423     36223           Medium
3   11687     36806              Low
4   69581      6376           Medium
```

+ Code + Text

2. Basic Info and Cleanup

```
print(df.info())           # Check columns & data types
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Post_ID           5000 non-null   object
 1   Platform          5000 non-null   object
 2   Hashtag           5000 non-null   object
 3   Content_Type      5000 non-null   object
 4   Region            5000 non-null   object
 5   Views             5000 non-null   int64
 6   Likes             5000 non-null   int64
 7   Shares            5000 non-null   int64
 8   Comments          5000 non-null   int64
 9   Engagement_Level  5000 non-null   object
dtypes: int64(4), object(6)
memory usage: 390.8+ KB
None
```

```
print(df.isnull().sum())   # See missing values
```

```
Post_ID             0
Platform            0
Hashtag             0
Content_Type        0
Region              0
Views               0
Likes               0
Shares              0
Comments            0
Engagement_Level    0
dtype: int64
```

```
# Fill or drop missing values if needed
df.dropna(inplace=True)
print(df.head())
```

```
  Post_ID   Platform     Hashtag Content_Type     Region    Views   Likes  \
0  Post_1     TikTok  #Challenge        Video         UK  4163464  339431
1  Post_2  Instagram  #Education       Shorts      India  4155940  215240
2  Post_3    Twitter  #Challenge        Video     Brazil  3666211  327143
3  Post_4    YouTube  #Education       Shorts  Australia   917951  127125
4  Post_5     TikTok      #Dance         Post     Brazil    64866  171361

   Shares  Comments Engagement_Level
0   53135     19346             High
1   65860     27239           Medium
2   39423     36223           Medium
3   11687     36806              Low
```

```
4   69581      6376           Medium
```

```
# Convert numerical columns to int
df[['Views', 'Likes', 'Shares', 'Comments']] = df[['Views', 'Likes', 'Shares', 'Comments']].astype(int)
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Post_ID           5000 non-null   object
 1   Platform          5000 non-null   object
 2   Hashtag           5000 non-null   object
 3   Content_Type      5000 non-null   object
 4   Region            5000 non-null   object
 5   Views             5000 non-null   int64 
 6   Likes             5000 non-null   int64 
 7   Shares            5000 non-null   int64 
 8   Comments          5000 non-null   int64 
 9   Engagement_Level  5000 non-null   object
dtypes: int64(4), object(6)
memory usage: 390.8+ KB
None
```

3. Calculate Engagement Rate

```
df['Engagement_Rate'] = (df['Likes'] + df['Shares'] + df['Comments']) / df['Views']
df['Engagement_Rate'] = df['Engagement_Rate'].fillna(0)
print(df.Engagement_Rate.head())
```

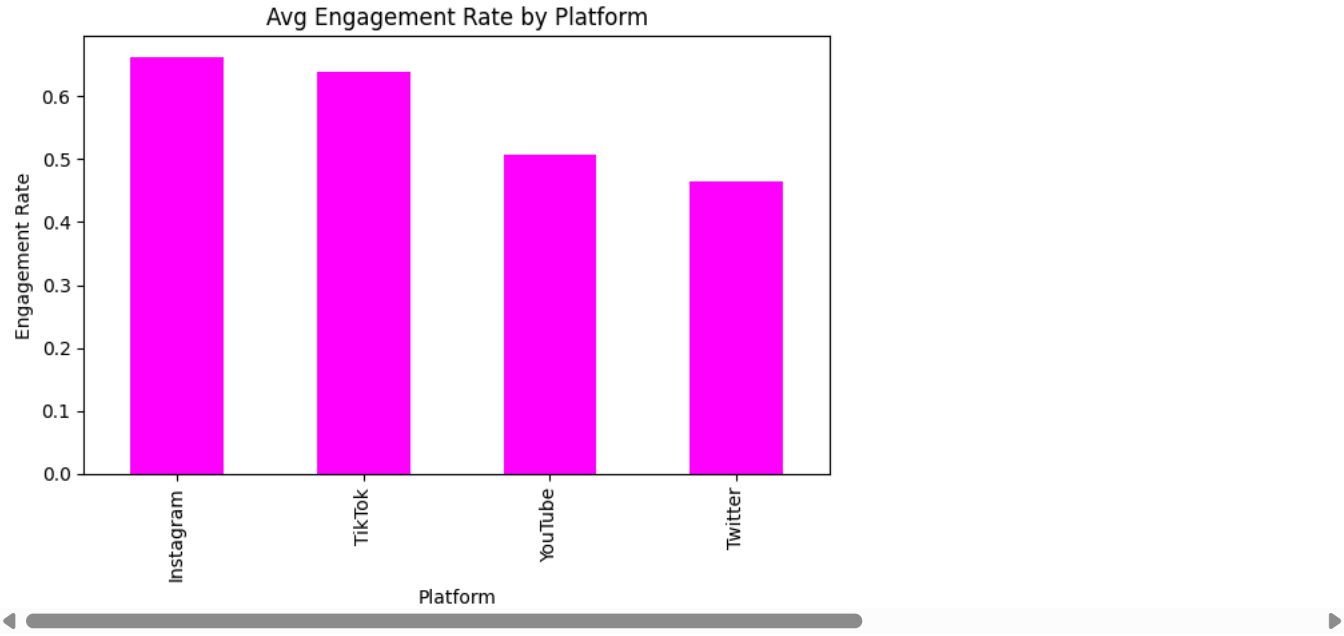
```
0    0.098935
1    0.074192
2    0.109865
3    0.191315
4    3.812752
Name: Engagement_Rate, dtype: float64
```

4. Top Performing Platforms

```
Platform_Avg = df.groupby('Platform')['Engagement_Rate'].mean().sort_values(ascending=False)
df["Platform_Avg"]= Platform_Avg
print(df.Platform_Avg.head())
```

```
0   NaN
1   NaN
2   NaN
3   NaN
4   NaN
Name: Platform_Avg, dtype: float64
```

```
#represent this Platform_Avg on a graph


platform_avg.plot(kind='bar', color='magenta', title='Avg Engagement Rate by Platform')
plt.ylabel('Engagement Rate')
plt.tight_layout()
plt.show()
```

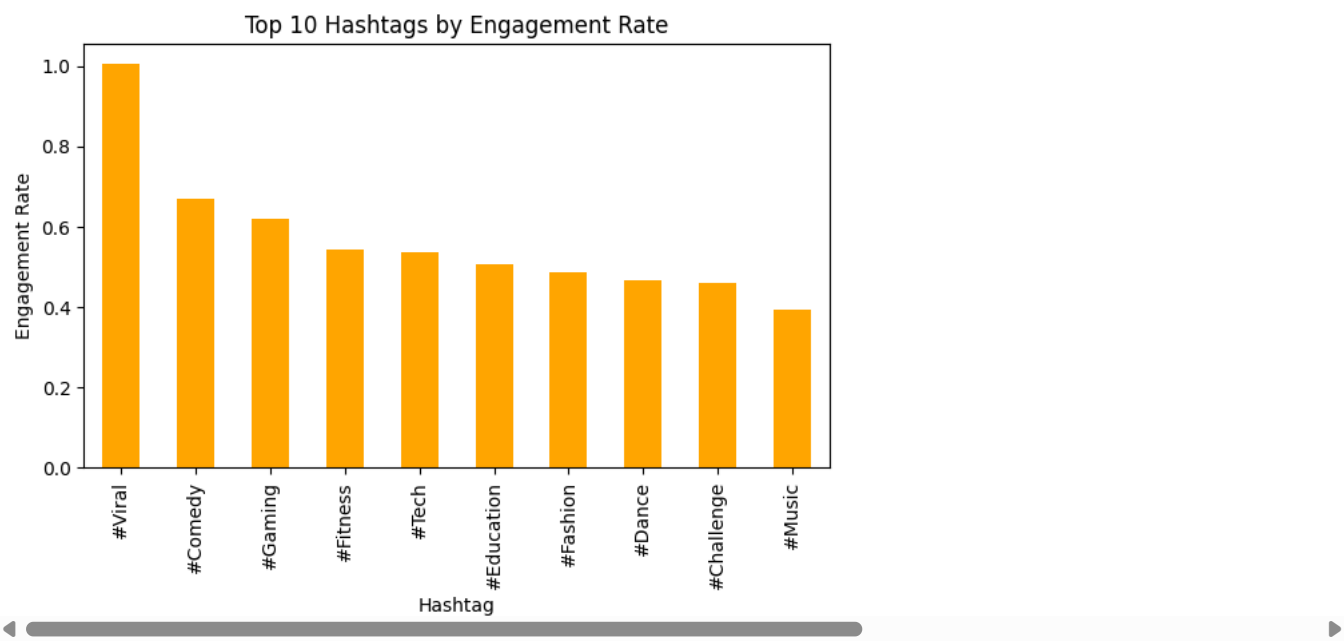


5. Most Engaging Hashtags

```
Top_Hashtags = df.groupby('Hashtag')['Engagement_Rate'].mean().sort_values(ascending=False).head(10)
df["Top_Hashtags"]= Top_Hashtags
print(df.Top_Hashtags.head())
```

```
0   NaN
1   NaN
2   NaN
3   NaN
4   NaN
Name: Top_Hashtags, dtype: float64
```

```
#represent this Top_Hashtags on a graph


Top_Hashtags.plot(kind='bar', title='Top 10 Hashtags by Engagement Rate', color='orange')
plt.ylabel('Engagement Rate')
plt.tight_layout()
plt.show()
```



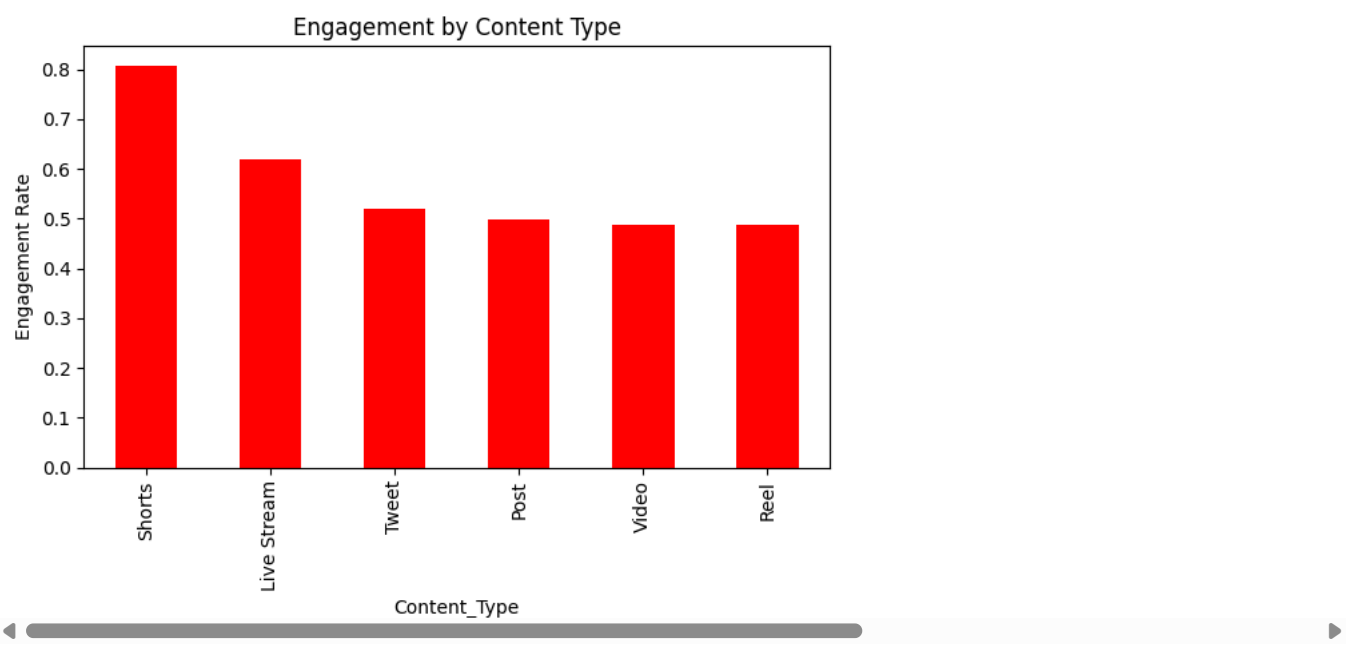
6. Engagement by Content Type

```
Content_Avg = df.groupby('Content_Type')['Engagement_Rate'].mean().sort_values(ascending=False)
df["Content_Avg"]=Content_Avg
```

```
print(df.Content_Avg.head())
```

```
0   NaN
1   NaN
2   NaN
3   NaN
4   NaN
Name: Content_Avg, dtype: float64
```

```
#represent this Content_Avg on a graph


Content_Avg.plot(kind='bar', color='red', title='Engagement by Content Type')
plt.ylabel('Engagement Rate')
plt.tight_layout()
plt.show()
```



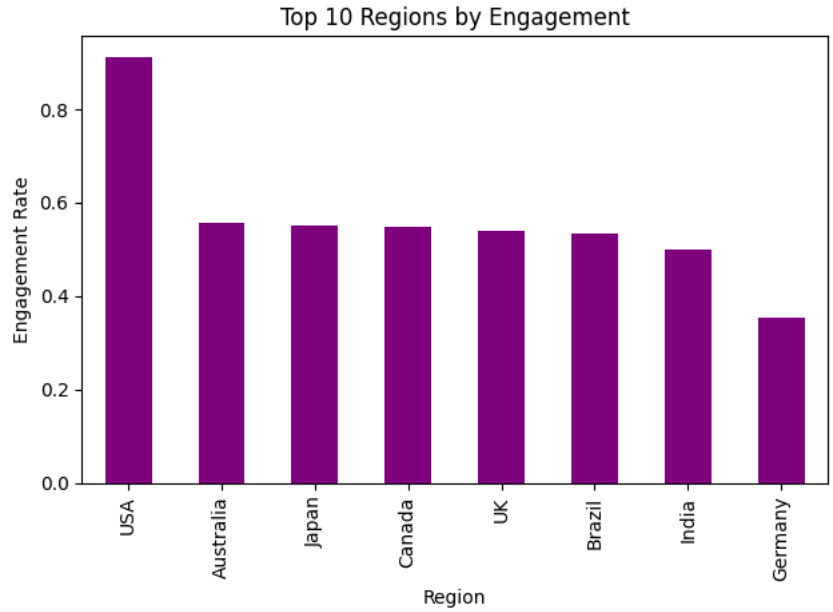
Double-click (or enter) to edit

7. Engagement by Region

```
Region_Avg = df.groupby('Region')['Engagement_Rate'].mean().sort_values(ascending=False).head(10)
df["Region_Avg"]=Region_Avg
print(df.Region_Avg.head())
```

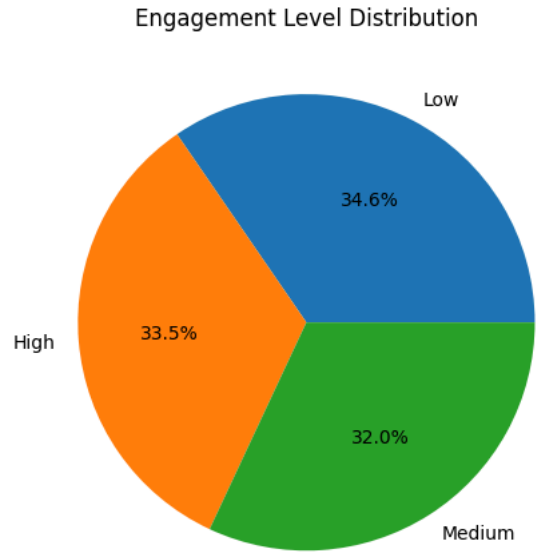
```
0   NaN
1   NaN
2   NaN
3   NaN
4   NaN
Name: Region_Avg, dtype: float64
```

```
#represent this Region_Avg on a graph


Region_Avg.plot(kind='bar', color='purple', title='Top 10 Regions by Engagement')
plt.ylabel('Engagement Rate')
plt.tight_layout()
plt.show()
```



8. Engagement Level Distribution

```
#represent Engagement_Level this  on a graph


df['Engagement_Level'].value_counts().plot(kind='pie',autopct='%1.1f%%', title='Engagement Level Distribution')  #autopct is a parameter
plt.ylabel('')
plt.tight_layout()
plt.show()
```



Summary Table

```
summary = df.groupby('Platform')[['Views', 'Likes', 'Shares', 'Comments']].sum()
summary['Engagement Rate'] = (summary['Likes'] + summary['Shares'] + summary['Comments']) / summary['Views']
print(summary.sort_values(by='Engagement Rate', ascending=False))
```

```
               Views      Likes    Shares  Comments  Engagement Rate
Platform
Instagram  2913744812  311627280  60976822  30249234         0.138260
YouTube    3370438480  342007739  66296773  33525521         0.131090
Twitter    3017229522  296039663  60474212  29446056         0.127919
TikTok     3168919406  307700467  64850003  31221158         0.127416
```

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

data = {
    'Platform': ['Instagram', 'YouTube', 'Twitter', 'TikTok'],
    'Views': [2913744812, 3370438480, 3017229522, 3168919406],
    'Likes': [311627280, 342007739, 296039663, 307700467],
    'Shares': [60976822, 66296773, 60474212, 64850003],
    'Comments': [30249234, 33525521, 29446056, 31221158]
}

df = pd.DataFrame(data)

bar_width = 0.15
index = np.arange(len(df))

fig, ax = plt.subplots(figsize=(16, 8))

# Create bars for each metric
ax.bar(index - 1.5 * bar_width, df['Views'], width=bar_width, label='Views', color='#1f77b4')
ax.bar(index - 0.5 * bar_width, df['Likes'], width=bar_width, label='Likes', color='#ff7f0e')
ax.bar(index + 0.5 * bar_width, df['Shares'], width=bar_width, label='Shares', color='#2ca02c')
ax.bar(index + 1.5 * bar_width, df['Comments'], width=bar_width, label='Comments', color='#d62728')

# Set labels and title
ax.set_xlabel('Platform', fontsize=20, fontweight='bold')
ax.set_ylabel('Count', fontsize=20, fontweight='bold')
ax.set_xticks(index)
ax.set_xticklabels(df['Platform'], fontsize=18, fontweight='bold')
ax.tick_params(axis='y', labelsize=16)

# Add legend
ax.legend(loc='upper left', bbox_to_anchor=(1, 1), fontsize=14, title='Metrics', title_fontsize=16)

# Set title and grid
plt.title('Social Media Metrics Comparison by Platform', fontsize=24, fontweight='bold')
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Show the plot
plt.tight_layout()  # Adjust layout to make room for the legend
plt.show()
```