Instagram Analytics

PYTHON CODE:

```
from google.colab import files

uploaded = files.upload()

# Instagram Analytics Dashboard using Python

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read_csv("/content/Instagram_data_by_Bhanu.csv", encoding='ISO-8859-1')

print("Dataset Shape:", df.shape)

print("\nColumns:", df.columns.tolist())

print("\nSummary Statistics:\n", df.describe())

top_likes = df.nlargest(10, 'Likes')

plt.figure(figsize=(10,6))

sns.barplot(x='Likes', y='Caption', data=top_likes, palette='coolwarm')

plt.title("Top 10 Instagram Posts by Likes")

plt.xlabel("Number of Likes")

plt.ylabel("Post Caption (truncated)")

plt.tight_layout()

plt.show()

plt.figure(figsize=(8,5))

sns.histplot(df['Impressions'], kde=True, color='skyblue')

plt.title("Distribution of Post Impressions")

plt.xlabel("Impressions")

plt.ylabel("Frequency")

plt.show()

numeric_cols = df.select_dtypes(include='number')
```

```
plt.figure(figsize=(10,6))

sns.heatmap(numeric_cols.corr(), annot=True, cmap='Blues', fmt=".2f")

plt.title("Correlation Between Instagram Metrics")

plt.show()

df['Engagement_Rate'] = ((df['Likes'] + df['Comments'] + df['Saves'] + df['Shares']) / df['Impressions']) * 100

plt.figure(figsize=(8,5))

sns.histplot(df['Engagement_Rate'], kde=True, color='green')

plt.title("Distribution of Engagement Rate (%)")

plt.xlabel("Engagement Rate (%)")

plt.ylabel("Number of Posts")

plt.show()

df['Primary_Hashtag'] = df['Hashtags'].apply(lambda x: x.split()[0] if isinstance(x, str) else None)

hashtag_engagement = df.groupby('Primary_Hashtag')['Engagement_Rate'].mean().sort_values(ascending=False).head(5)

plt.figure(figsize=(8,5))

sns.barplot(x=hashtag_engagement.values, y=hashtag_engagement.index, palette='mako')

plt.title("Top 5 Hashtags by Average Engagement Rate")

plt.xlabel("Average Engagement Rate (%)")

plt.ylabel("Hashtag")

plt.show()

plt.figure(figsize=(8,5))

sns.scatterplot(x='Likes', y='Comments', data=df, color='purple')

plt.title("Relationship Between Likes and Comments")

plt.xlabel("Likes")

plt.ylabel("Comments")
```

```
plt.show()

print("\n✅ Dashboard Visualizations Generated Successfully!")
```

OUTPUT:

```
Saving Instagram_data_by_Bhanu.csv to Instagram_data_by_Bhanu (1).csv
Dataset Shape: (119, 13)

Columns: ['Impressions', 'From Home', 'From Hashtags', 'From Explore', 'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visits', 'Follows', 'Caption', 'Hashtags']

Summary Statistics:
       Impressions     From Home  From Hashtags  From Explore   From Other  \
count   119.000000    119.000000    119.000000    119.000000   119.000000
mean   5703.991597   2475.789916   1887.512605   1078.100840   171.092437
std    4843.780105   1489.386348   1884.361443   2613.026132   289.431031
min    1941.000000   1133.000000    116.000000     0.000000     9.000000
25%    3467.000000   1945.000000    726.000000    157.500000    38.000000
50%    4289.000000   2207.000000   1278.000000    326.000000    74.000000
75%    6138.000000   2602.500000   2363.500000    689.500000   196.000000
max   36919.000000  13473.000000  11817.000000  17414.000000  2547.000000

          Saves    Comments     Shares      Likes  Profile Visits  \
count   119.000000  119.000000  119.000000  119.000000     119.000000
mean    153.310924    6.663866    9.361345  173.781513      50.621849
std     156.317731    3.544576   10.089205   82.378947      87.088402
min      22.000000    0.000000    0.000000   72.000000       4.000000
```

25% 65.000000 4.000000 3.000000 121.500000 15.000000

50% 109.000000 6.000000 6.000000 151.000000 23.000000

75% 169.000000 8.000000 13.500000 204.000000 42.000000

max 1095.000000 19.000000 75.000000 549.000000 611.000000

Follows

count 119.000000

mean 20.756303

std 40.921580

min 0.000000

25% 4.000000

50% 8.000000

75% 18.000000

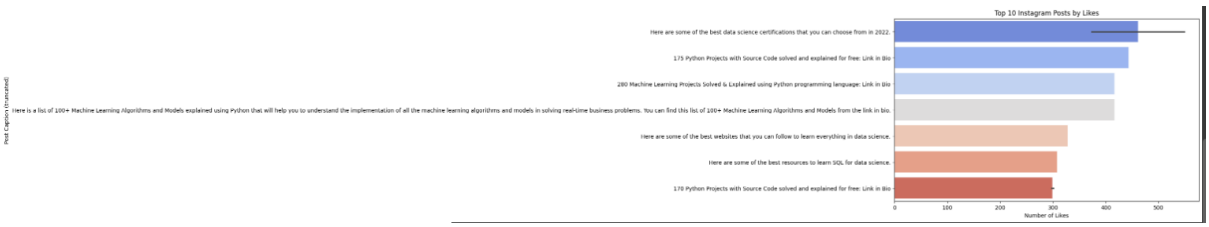
max 260.000000

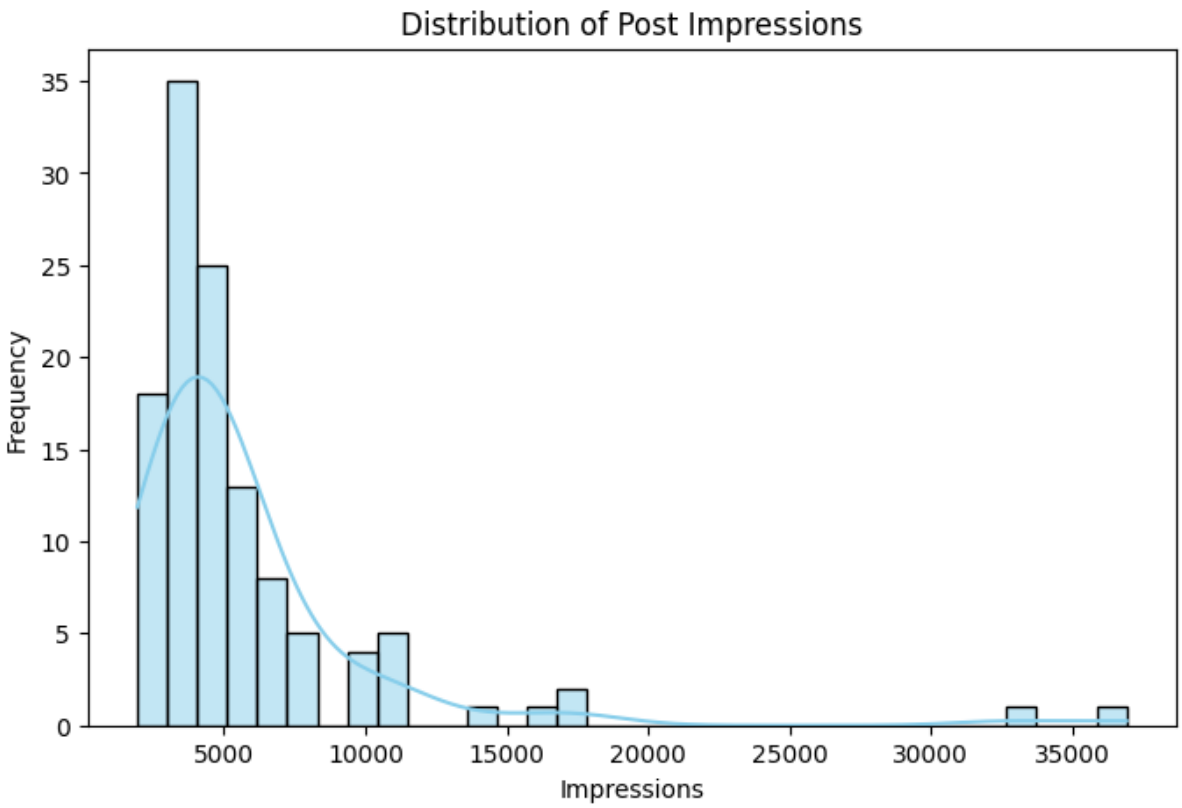
/tmp/ipython-input-3412741208.py:19: FutureWarning:

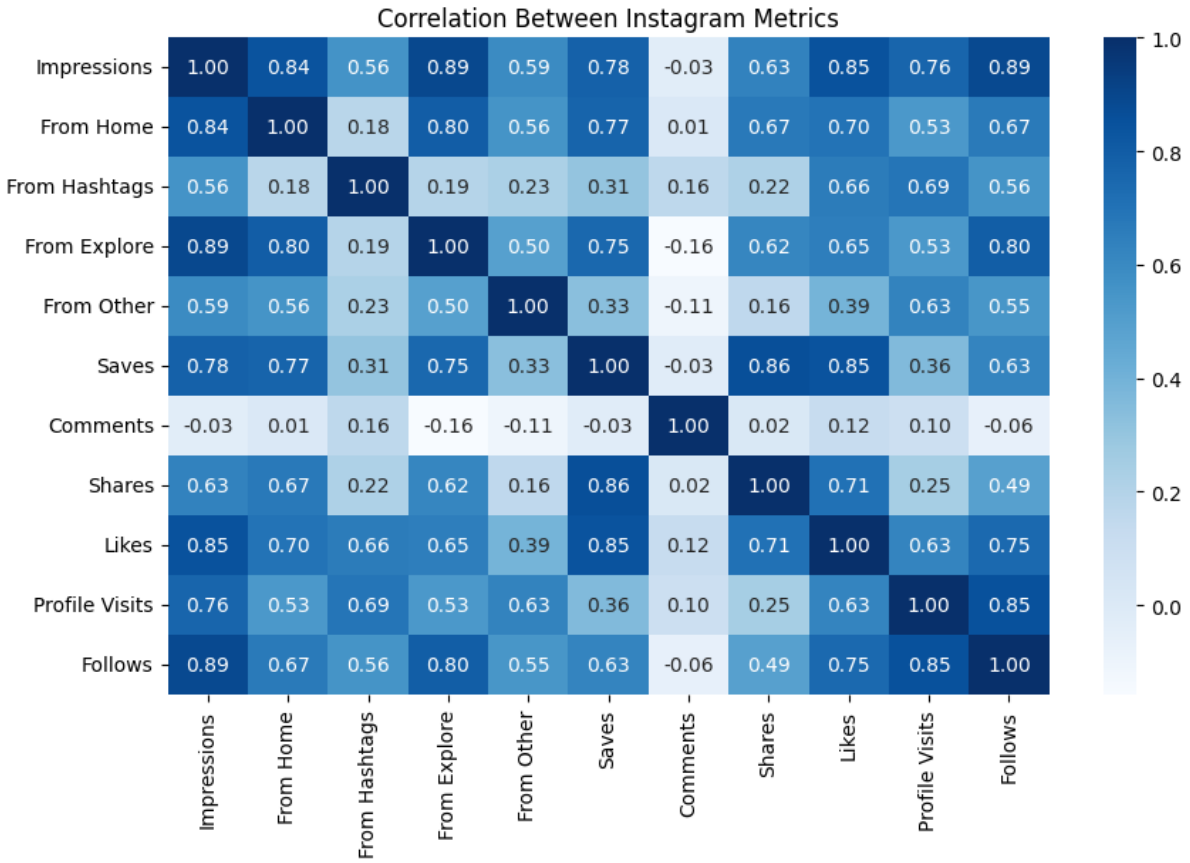
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

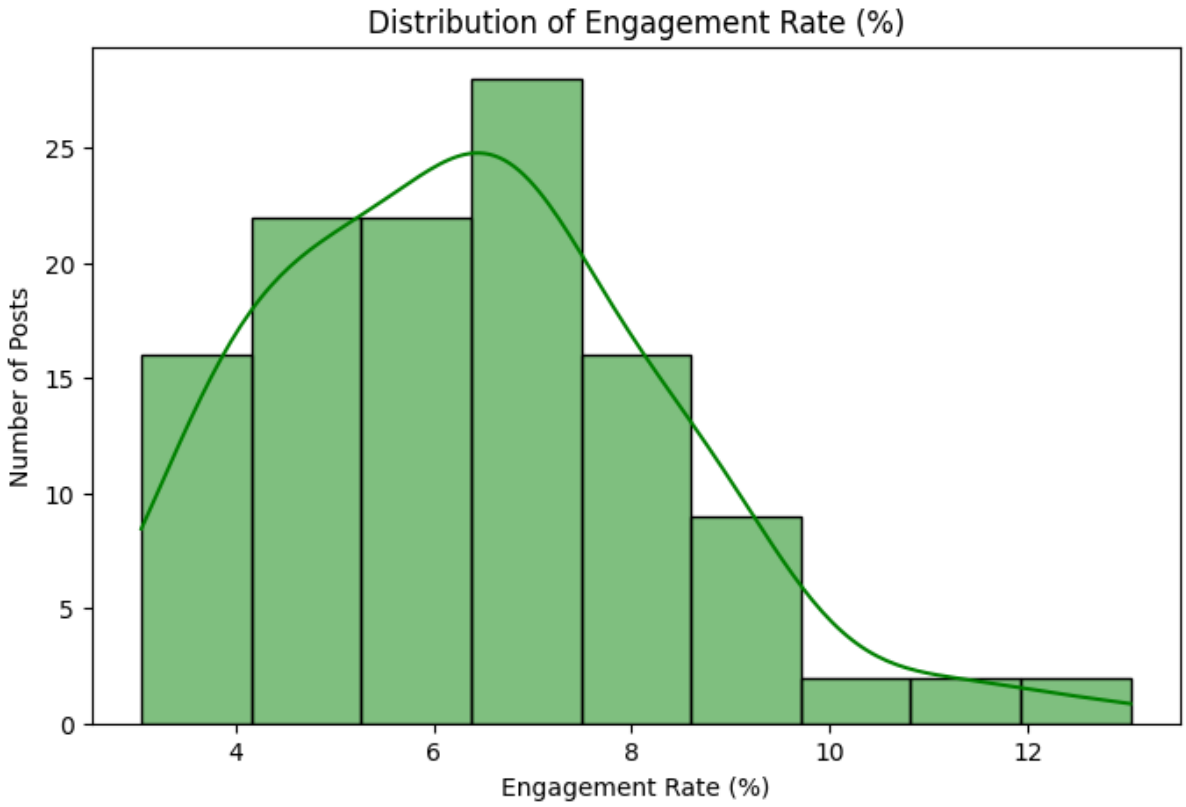
sns.barplot(x='Likes', y='Caption', data=top_likes, palette='coolwarm')

/tmp/ipython-input-3412741208.py:23: UserWarning: Tight layout not applied. The left and right margins cannot be made large enough to accommodate all Axes decorations.




Distribution of Post Impressions
Frequency
Impressions
0
5
10
15
20
25
30
35
5000
10000
15000
20000
25000
30000
35000

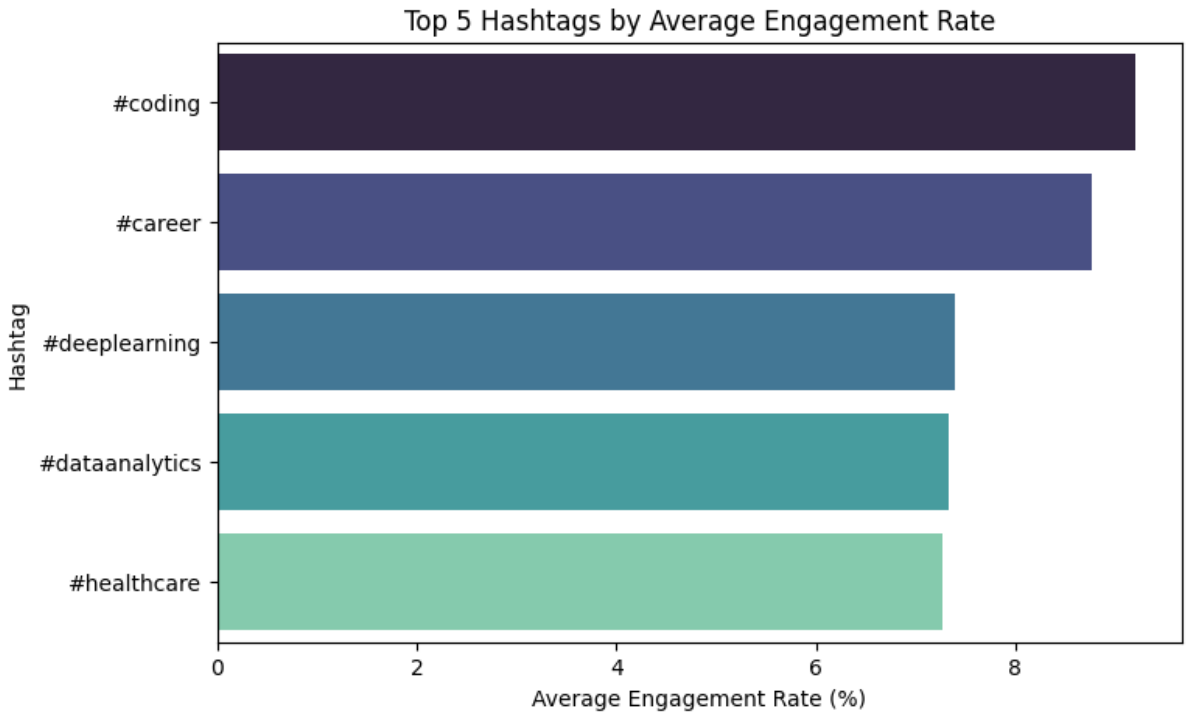

Correlation Between Instagram Metrics
Impressions 1.00 0.84 0.56 0.89 0.59 0.78 -0.03 0.63 0.85 0.76 0.89
From Home 0.84 1.00 0.18 0.80 0.56 0.77 0.01 0.67 0.70 0.53 0.67
From Hashtags 0.56 0.18 1.00 0.19 0.23 0.31 0.16 0.22 0.66 0.69 0.56
From Explore 0.89 0.80 0.19 1.00 0.50 0.75 -0.16 0.62 0.65 0.53 0.80
From Other 0.59 0.56 0.23 0.50 1.00 0.33 -0.11 0.16 0.39 0.63 0.55
Saves 0.78 0.77 0.31 0.75 0.33 1.00 -0.03 0.86 0.85 0.36 0.63
Comments -0.03 0.01 0.16 -0.16 -0.11 -0.03 1.00 0.02 0.12 0.10 -0.06
Shares 0.63 0.67 0.22 0.62 0.16 0.86 0.02 1.00 0.71 0.25 0.49
Likes 0.85 0.70 0.66 0.65 0.39 0.85 0.12 0.71 1.00 0.63 0.75
Profile Visits 0.76 0.53 0.69 0.53 0.63 0.36 0.10 0.25 0.63 1.00 0.85
Follows 0.89 0.67 0.56 0.80 0.55 0.63 -0.06 0.49 0.75 0.85 1.00
Impressions
From Home
From Hashtags
From Explore
From Other
Saves
Comments
Shares
Likes
Profile Visits
Follows
1.0
0.8
0.6
0.4
0.2
0.0

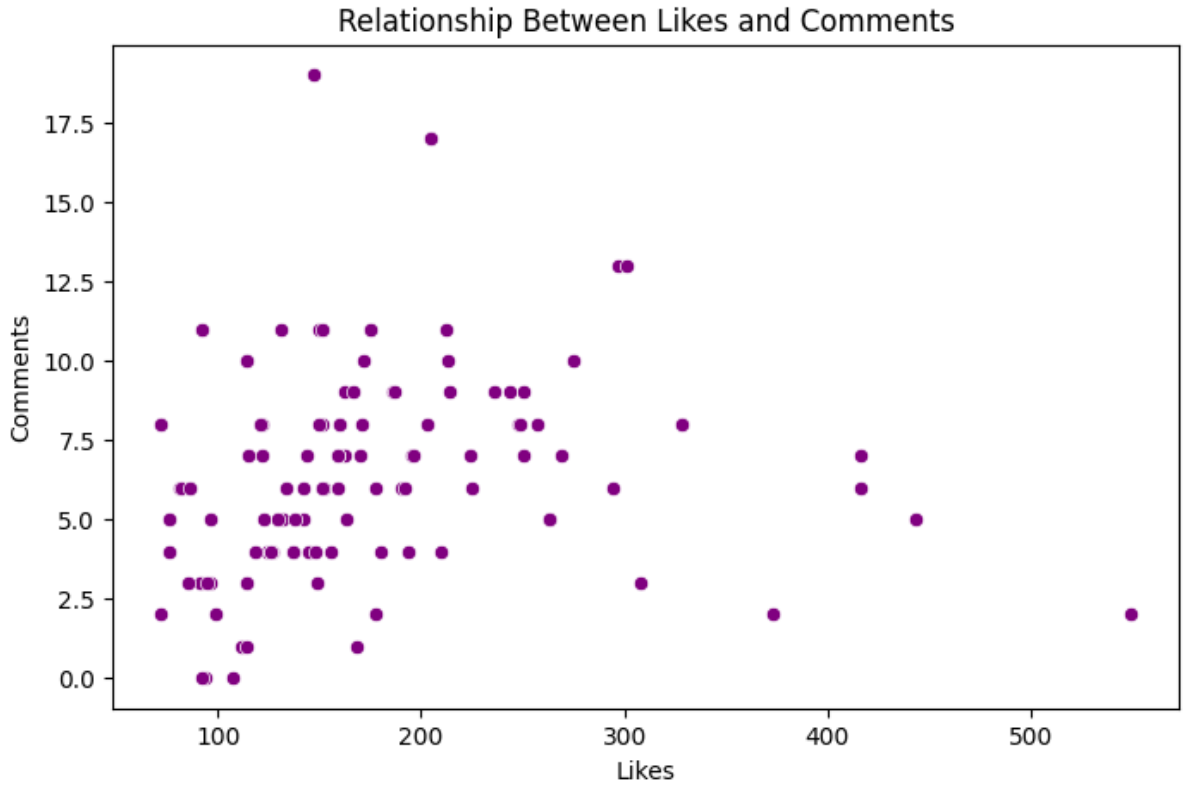


/tmp/ipython-input-3412741208.py:56: FutureWarning:

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=hashtag_engagement.values, y=hashtag_engagement.index, palette='mako')





✅ Dashboard Visualizations Generated Successfully!