Simppl assignment - social media analysis (instagram)

In [ ]:

```
# src/data_collection.py
import requests
from bs4 import BeautifulSoup
import pandas as pd
import os

def fetch_sharechat_data(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')

    # Debug print to check the structure of the page
    print(soup.prettify())

    posts = []
    for post in soup.find_all('div', class_='post'):
        account_tag = post.find('a', class_='account-name')
        content_tag = post.find('div', class_='post-content')
        likes_tag = post.find('span', class_='like-count')
        comments_tag = post.find('span', class_='comment-count')
        shares_tag = post.find('span', class_='share-count')

        if account_tag and content_tag and likes_tag and comments_tag and shares_ta
            post_data = {
                'account': account_tag.text.strip(),
                'content': content_tag.text.strip(),
                'likes': int(likes_tag.text.strip()),
                'comments': int(comments_tag.text.strip()),
                'shares': int(shares_tag.text.strip()),
            }
            posts.append(post_data)

    return pd.DataFrame(posts)

if __name__ == '__main__':
    url = 'https://www.instagram.com/explore/'
    data = fetch_sharechat_data(url)

    # Ensure the 'data' directory exists
    os.makedirs('data', exist_ok=True)

    # Check if data is fetched properly
    if not data.empty:
        print("Data fetched successfully!")
        print(data.head())
    else:
        print("No data fetched.")
    # Save the collected data to a CSV file
    data.to_csv('data/collected_data.csv', index=False)
```

In [ ]:

```
# Create a sample CSV for testing
import pandas as pd
import os

# Sample data
data = {
    'account': ['account1', 'account2', 'account3'],
    'content': ['content1', 'content2', 'content3'],
    'likes': [10, 20, 30],
    'comments': [5, 10, 15],
    'shares': [2, 3, 4]
}

# Create a DataFrame
df = pd.DataFrame(data)

# Ensure the 'data' directory exists
os.makedirs('data', exist_ok=True)

# Save the DataFrame to a CSV file
df.to_csv('data/collected_data.csv', index=False)
```

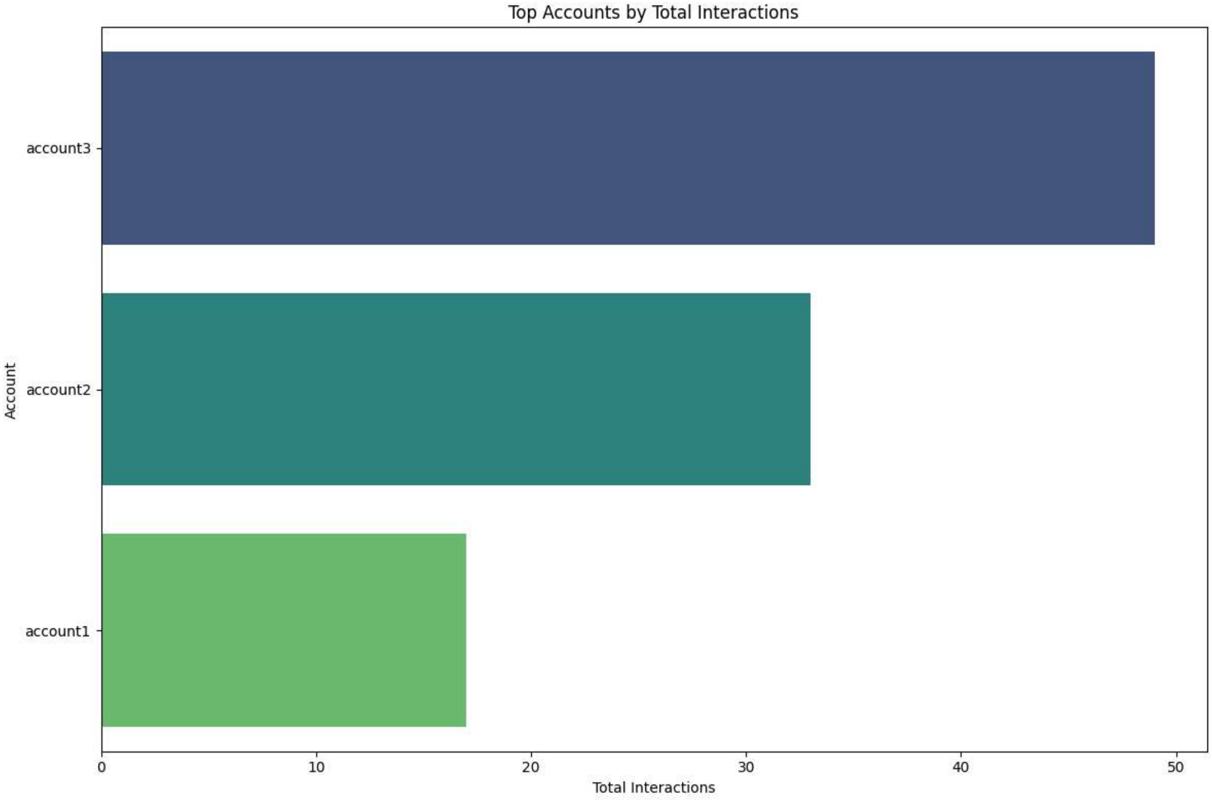
In [1]:

```
# src/data_analysis.py
import pandas as pd
import os

def analyze_data(file_path):
    # Check if the file exists and is not empty
    if not os.path.exists(file_path):
        raise FileNotFoundError(f"No such file: '{file_path}'")
    if os.stat(file_path).st_size == 0:
        raise ValueError(f"File is empty: '{file_path}'")

    data = pd.read_csv(file_path)

    # Check if the data has the expected columns
    expected_columns = {'account', 'content', 'likes', 'comments', 'shares'}
    if not expected_columns.issubset(data.columns):
        raise ValueError(f"File does not contain the expected columns. Found column

    # Calculate total interactions
    data['total_interactions'] = data['likes'] + data['comments'] + data['shares']

    # Identify top accounts by interactions
    top_accounts = data.groupby('account')['total_interactions'].sum().reset_index(
    top_accounts = top_accounts.sort_values(by='total_interactions', ascending=Fals

    return data, top_accounts

if __name__ == '__main__':
    try:
        data, top_accounts = analyze_data('data/collected_data.csv')
        data.to_csv('data/analyzed_data.csv', index=False)
        top_accounts.to_csv('data/top_accounts.csv', index=False)
        print("Data analysis completed successfully!")
    except (FileNotFoundError, ValueError, pd.errors.EmptyDataError) as e:
        print(f"Error: {e}")
```

```
Data analysis completed successfully!
```

In [2]:

```
# src/visualization.py
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os

def plot_top_accounts(file_path):
    # Ensure the 'plots' directory exists
    os.makedirs('plots', exist_ok=True)

    top_accounts = pd.read_csv(file_path)

    plt.figure(figsize=(12, 8))
    sns.barplot(x='total_interactions', y='account', data=top_accounts, hue='accoun
    plt.title('Top Accounts by Total Interactions')
    plt.xlabel('Total Interactions')
    plt.ylabel('Account')
    plt.legend([],[], frameon=False)  # Disable the legend
    plt.tight_layout()
    plt.savefig('plots/top_accounts.png')
    plt.show()

if __name__ == '__main__':
    plot_top_accounts('data/top_accounts.csv')
```

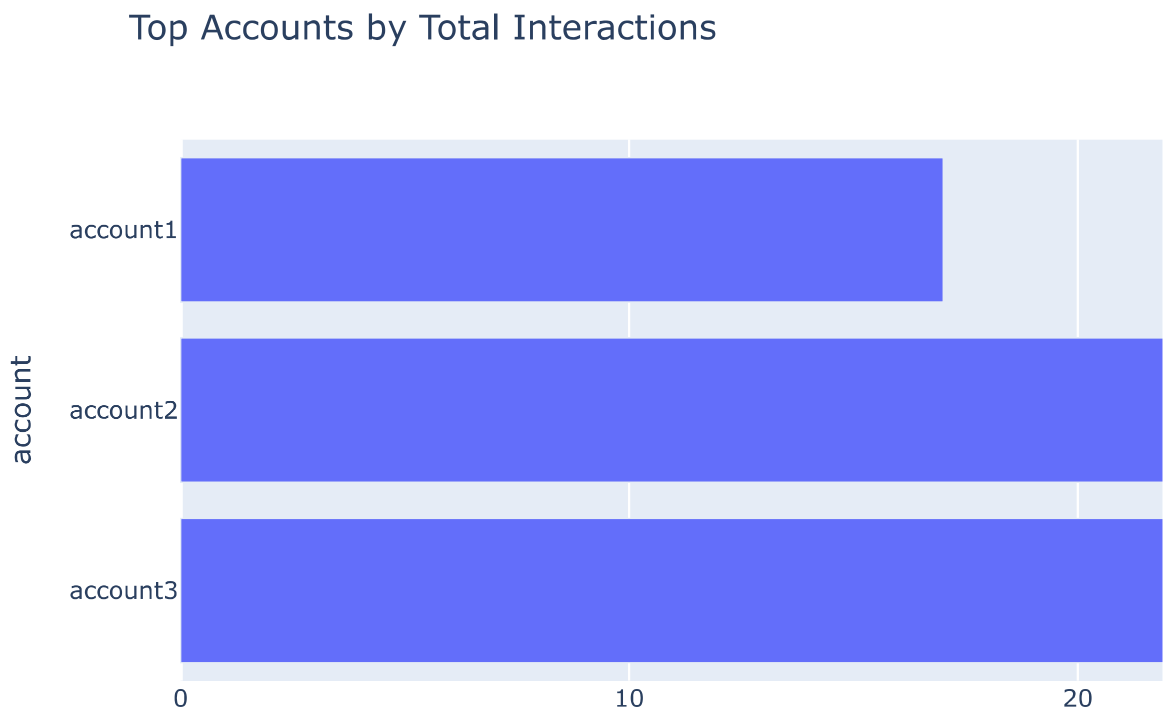


In [3]:

```
# src/dashboard.py
import pandas as pd
from dash import Dash, dcc, html
import plotly.express as px

app = Dash(__name__)

# Load data
data = pd.read_csv('data/analyzed_data.csv')
top_accounts = pd.read_csv('data/top_accounts.csv')

# Create plots
fig = px.bar(top_accounts, x='total_interactions', y='account', title='Top Accounts

app.layout = html.Div(children=[
    html.H1(children='Social Media Analytics Dashboard'),

    html.Div(children='''
        A dashboard to visualize social media data.
    '''),

    dcc.Graph(
        id='top-accounts-bar',
        figure=fig
    ),

    # Add more components as needed
])

if __name__ == '__main__':
    app.run_server(debug=True)
```

Social Media Analytics Dashboard

A dashboard to visualize social media data.



```
In [ ]:
```