Drugs Side Effect And Medical Condition

```
# Step 1: Imports
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import re
from sklearn.preprocessing import LabelEncoder, StandardScaler
import warnings
warnings.filterwarnings('ignore')

try:
    # Step 2: Load the dataset
    print("Loading dataset...")
    df = pd.read_csv("drugs_side_effects_drugs_com.csv")
    print(f"Dataset loaded successfully. Shape: {df.shape}")
    print("\nColumns in the dataset:")
    print(df.columns.tolist())

    # Step 3: Basic Cleaning
    print("\nCleaning data...")
    # Check if columns exist before cleaning
    required_columns = ['alcohol', 'side_effects', 'related_drugs', 'generic_name',
                        'drug_classes', 'rx_otc', 'pregnancy_category', 'rating',
                        'no_of_reviews', 'medical_condition', 'csa']

    missing_columns = [col for col in required_columns if col not in df.columns]
```

```
if missing_columns:
    raise ValueError(f"Missing required columns: {missing_columns}")

# Clean the data
df['alcohol'] = df['alcohol'].replace({np.nan: 0, 'X': 1})
df['side_effects'] = df['side_effects'].fillna('Unknown')
df['related_drugs'] = df['related_drugs'].fillna('Unknown')
df['generic_name'] = df['generic_name'].fillna('Unknown')
df['drug_classes'] = df['drug_classes'].fillna('Unknown')
df['rx_otc'] = df['rx_otc'].fillna('Unknown')
df['pregnancy_category'] = df['pregnancy_category'].fillna('Unknown')
df['rating'] = pd.to_numeric(df['rating'], errors='coerce').fillna(0)
df['no_of_reviews'] = pd.to_numeric(df['no_of_reviews'], errors='coerce').fillna(0)

# Step 4: Label Encoding
print("\nPerforming label encoding...")
cols_to_encode = ['generic_name', 'medical_condition', 'side_effects',
                  'drug_classes', 'rx_otc', 'pregnancy_category', 'csa']

encoder = LabelEncoder()
for col in cols_to_encode:
    if col in df.columns:
        df[col] = encoder.fit_transform(df[col].astype(str))

# Step 5: Normalize numeric columns
print("\nNormalizing numeric columns...")
scale_cols = ['generic_name', 'medical_condition', 'no_of_reviews', 'side_effects',
              'rating', 'csa', 'pregnancy_category', 'rx_otc', 'alcohol']
```

```
scaler = StandardScaler()
df_scaled = pd.DataFrame(scaler.fit_transform(df[scale_cols]), columns=scale_cols)

# Step 6: EDA - Side Effects Frequency
print("\nAnalyzing side effects...")
def extract_side_effects(text):
    if pd.isna(text) or text == 'Unknown':
        return []
    return [s.strip().lower() for s in re.split(r'[;,]', str(text))]

side_effects_series = df['side_effects'].astype(str).apply(extract_side_effects).explode()
side_effect_counts = side_effects_series.value_counts().head(10)

plt.figure(figsize=(12,6))
side_effect_counts.plot(kind='bar', color='salmon')
plt.title("Top 10 Side Effects")
plt.xlabel("Side Effect")
plt.ylabel("Frequency")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

# Step 7: Medical Condition Frequency
print("\nAnalyzing medical conditions...")
plt.figure(figsize=(12,6))
df['medical_condition'].value_counts().head(10).plot(kind='bar', color='skyblue')
plt.title("Top 10 Medical Conditions")
```

```
    plt.xlabel("Condition")
    plt.ylabel("Frequency")
    plt.xticks(rotation=45, ha='right')
    plt.tight_layout()
    plt.show()

    # Step 8: Rating Distribution
    print("\nAnalyzing rating distribution...")
    plt.figure(figsize=(10,6))
    sns.histplot(df['rating'], bins=20, kde=True)
    plt.title("Distribution of Drug Ratings")
    plt.xlabel("Rating")
    plt.ylabel("Count")
    plt.tight_layout()
    plt.show()

    # Step 9: Save cleaned dataset
    print("\nSaving cleaned dataset...")
    df.to_csv("cleaned_drug_dataset.csv", index=False)
    print("Analysis complete! Cleaned dataset saved as 'cleaned_drug_dataset.csv'")

except FileNotFoundError:
    print("Error: drugs_side_effects_drugs_com.csv file not found. Please make sure the file exists in the same directory as this script.")
except Exception as e:
    print(f"An error occurred: {str(e)}")
    print("\nPlease check if:")
    print("1. The CSV file is properly formatted")
```

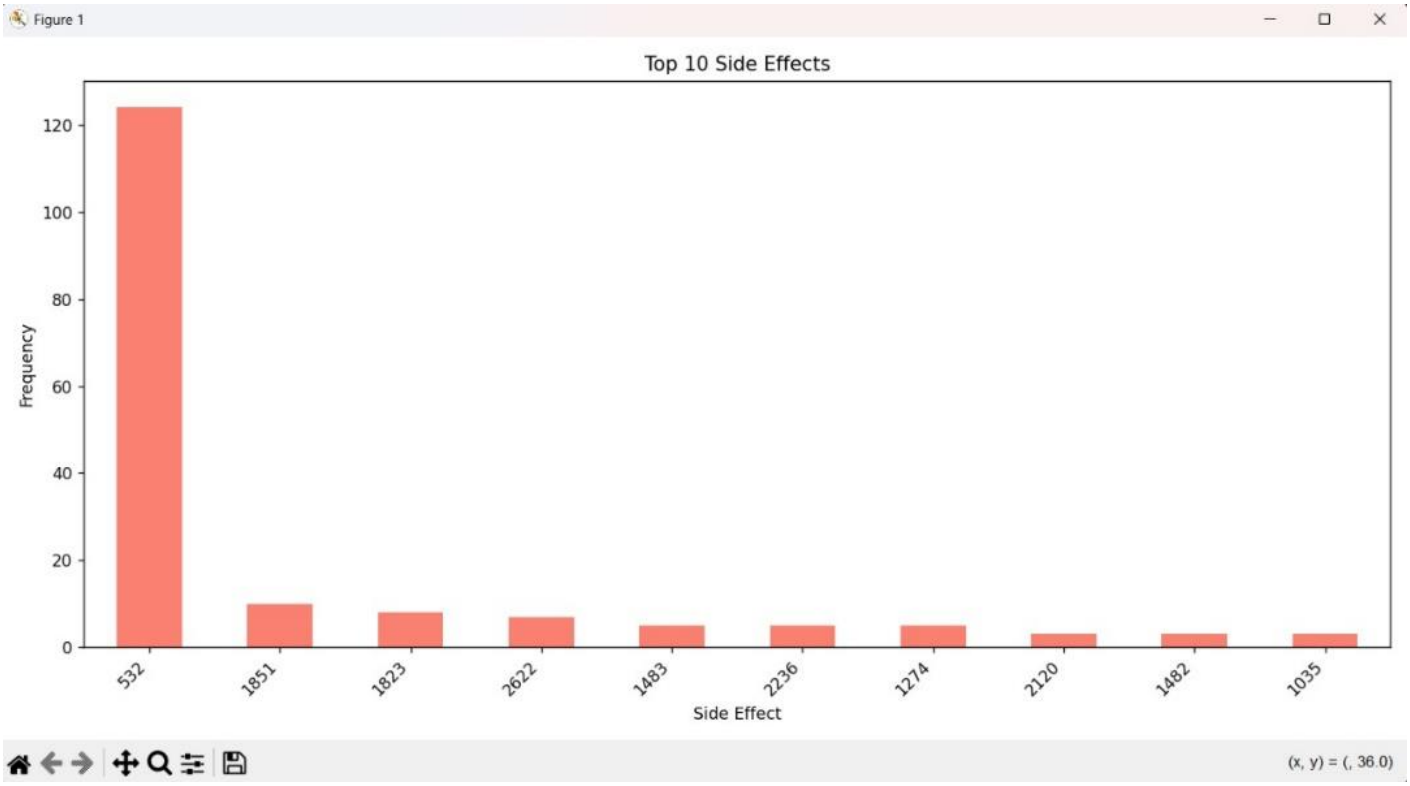
print("2. All required columns are present")

print("3. The data types are correct")

```python
# Step 1: Imports
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import re
from sklearn.preprocessing import LabelEncoder, StandardScaler
import warnings
warnings.filterwarnings('ignore')

try:
    # Step 2: Load the dataset
    print("Loading dataset...")
    df = pd.read_csv("drugs_side_effects_drugs_com.csv")
    print(f"Dataset loaded successfully. Shape: {df.shape}")
    print("\nColumns in the dataset:")
    print(df.columns.tolist())

    # Step 3: Basic Cleaning
    print("\nCleaning data...")
    # Check if columns exist before cleaning
    required_columns = ['alcohol', 'side_effects', 'related_drugs', 'generic_name',
                        'drug_classes', 'rx_otc', 'pregnancy_category', 'rating',
                        'no_of_reviews', 'medical_condition', 'csa']

    missing_columns = [col for col in required_columns if col not in df.columns]
    if missing_columns:
        raise ValueError(f"Missing required columns: {missing_columns}")

    # Clean the data
    df['alcohol'] = df['alcohol'].replace({np.nan: 0, 'X': 1})
    df['side_effects'] = df['side_effects'].fillna('Unknown')
    df['related_drugs'] = df['related_drugs'].fillna('Unknown')
    df['generic_name'] = df['generic_name'].fillna('Unknown')
    df['drug_classes'] = df['drug_classes'].fillna('Unknown')
    df['rx_otc'] = df['rx_otc'].fillna('Unknown')
    df['pregnancy_category'] = df['pregnancy_category'].fillna('Unknown')
    df['rating'] = pd.to_numeric(df['rating'], errors='coerce').fillna(0)
    df['no_of_reviews'] = pd.to_numeric(df['no_of_reviews'], errors='coerce').fillna(0)

    # Step 4: Label Encoding
    print("\nPerforming label encoding...")
    cols_to_encode = ['generic_name', 'medical_condition', 'side_effects',
                      'drug_classes', 'rx_otc', 'pregnancy_category', 'csa']
```

```python
    encoder = LabelEncoder()
    for col in cols_to_encode:
        if col in df.columns:
            df[col] = encoder.fit_transform(df[col].astype(str))

    # Step 5: Normalize numeric columns
    print("\nNormalizing numeric columns...")
    scale_cols = ['generic_name', 'medical_condition', 'no_of_reviews', 'side_effects',
                  'rating', 'csa', 'pregnancy_category', 'rx_otc', 'alcohol']

    scaler = StandardScaler()
    df_scaled = pd.DataFrame(scaler.fit_transform(df[scale_cols]), columns=scale_cols)

    # Step 6: EDA - Side Effects Frequency
    print("\nAnalyzing side effects...")
    def extract_side_effects(text):
        if pd.isna(text) or text == 'Unknown':
            return []
        return [s.strip().lower() for s in re.split(r'[;,]', str(text))]

    side_effects_series = df['side_effects'].astype(str).apply(extract_side_effects).explode()
    side_effect_counts = side_effects_series.value_counts().head(10)

    plt.figure(figsize=(12,6))
    side_effect_counts.plot(kind='bar', color='salmon')
    plt.title("Top 10 Side Effects")
    plt.xlabel("Side Effect")
    plt.ylabel("Frequency")
    plt.xticks(rotation=45, ha='right')
    plt.tight_layout()
    plt.show()

    # Step 7: Medical Condition Frequency
    print("\nAnalyzing medical conditions...")
    plt.figure(figsize=(12,6))
    df['medical_condition'].value_counts().head(10).plot(kind='bar', color='skyblue')
    plt.title("Top 10 Medical Conditions")
    plt.xlabel("Condition")
    plt.ylabel("Frequency")
    plt.xticks(rotation=45, ha='right')
    plt.tight_layout()
    plt.show()

    # Step 8: Rating Distribution
    print("\nAnalyzing rating distribution...")
```

```python
    # Step 8: Rating Distribution
    print("\nAnalyzing rating distribution...")
    plt.figure(figsize=(10,6))
    sns.histplot(df['rating'], bins=20, kde=True)
    plt.title("Distribution of Drug Ratings")
    plt.xlabel("Rating")
    plt.ylabel("Count")
    plt.tight_layout()
    plt.show()

    # Step 9: Save cleaned dataset
    print("\nSaving cleaned dataset...")
    df.to_csv("cleaned_drug_dataset.csv", index=False)
    print("Analysis complete! Cleaned dataset saved as 'cleaned_drug_dataset.csv'")

except FileNotFoundError:
    print("Error: drugs_side_effects_drugs_com.csv file not found. Please make sure the file exists in the same directory as this script.")
except Exception as e:
    print(f"An error occurred: {str(e)}")
    print("\nPlease check if:")
    print("1. The CSV file is properly formatted")
    print("2. All required columns are present")
    print("3. The data types are correct")
```



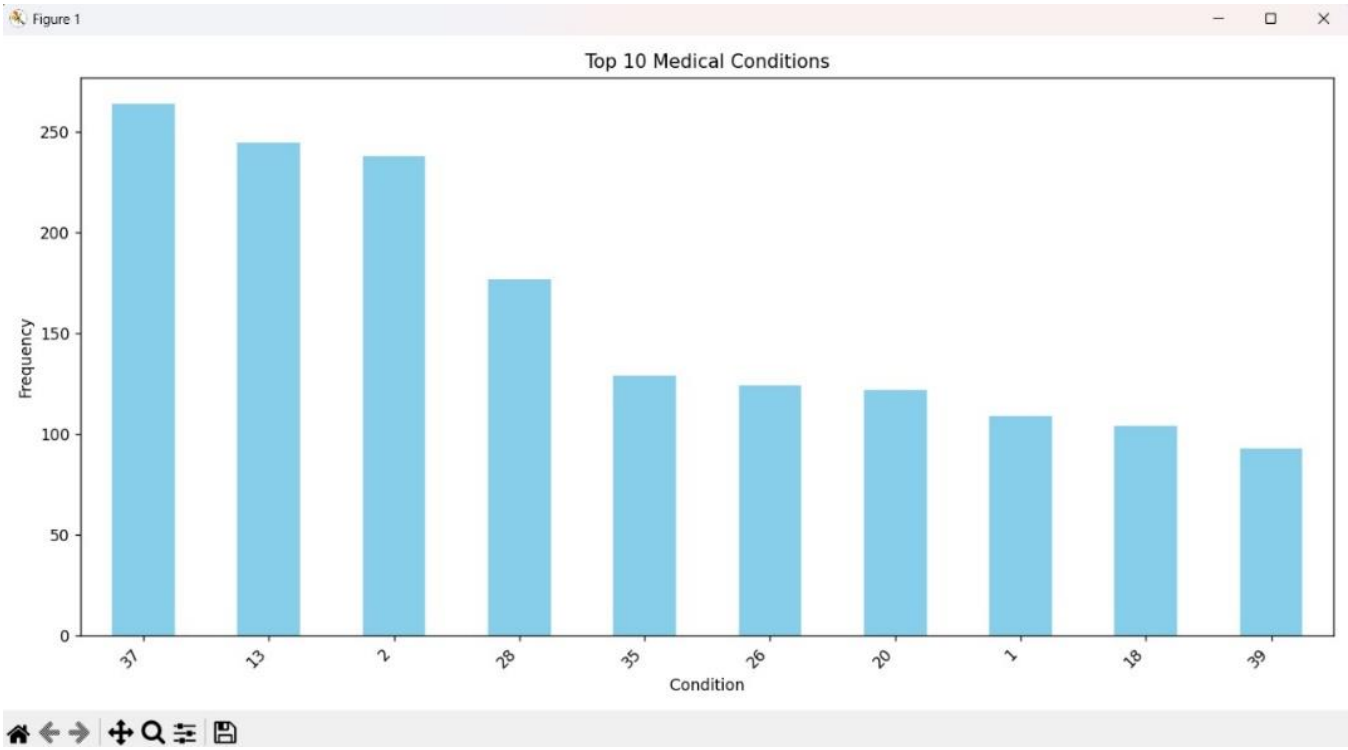


Figure 1
Top 10 Medical Conditions
Frequency
0
50
100
150
200
250
37
13
2
28
35
26
20
1
18
39
Condition

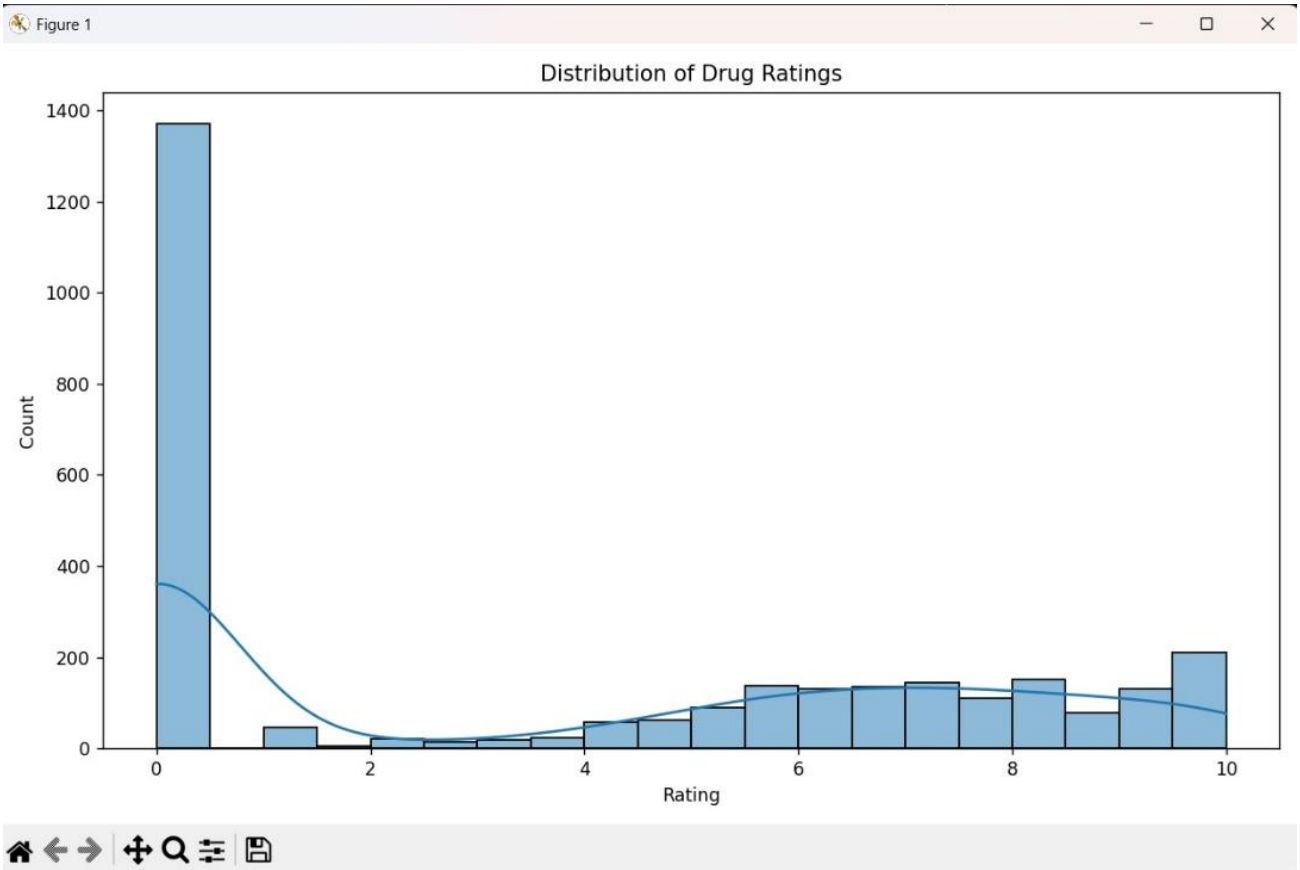


Figure 1
Distribution of Drug Ratings
Count
0
200
400
600
800
1000
1200
1400
0
2
4
6
8
10
Rating