In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('covid_clinical_trials.csv')

print(df.shape)

df.head()
```

```
(5783, 27)
```

Out[1]:

	Rank	NCT Number	Title	Acronym	Status	Study Results	Conditions
0	1	NCT04785898	Diagnostic Performance of the ID Now™ COVID-19...	COVID-IDNow	Active, not recruiting	No Results Available	Covid19
1	2	NCT04595136	Study to Evaluate the Efficacy of COVID19-0001...	COVID-19	Not yet recruiting	No Results Available	SARS-CoV-2 Infection
2	3	NCT04395482	Lung CT Scan Analysis of SARS-CoV2 Induced Lun...	TAC-COVID19	Recruiting	No Results Available	covid19
3	4	NCT04416061	The Role of a Private Hospital in Hong Kong Am...	COVID-19	Active, not recruiting	No Results Available	COVID
4	5	NCT04395924	Maternal-foetal Transmission of SARS-Cov-2	TMF-COVID-19	Recruiting	No Results Available	Maternal Fetal Infection Transmission|COVID-19...

5 rows × 27 columns

In [2]:
```
# Check missing values
```

In [3]:
```
df.isnull().sum()
```

Out[3]:
```
Rank                          0
NCT Number                    0
Title                         0
Acronym                    3303
Status                        0
Study Results                 0
Conditions                    0
Interventions               886
Outcome Measures             35
Sponsor/Collaborators         0
Gender                       10
Age                           0
Phases                     2461
Enrollment                   34
Funded Bys                    0
Study Type                    0
Study Designs                35
Other IDs                     1
Start Date                   34
Primary Completion Date      36
Completion Date              36
First Posted                  0
Results First Posted       5747
Last Update Posted            0
Locations                   585
Study Documents            5601
URL                           0
dtype: int64
```

In [4]:
```
df.drop(['Results First Posted', 'Study Documents'], axis=1, inplace=True)
```

In [5]:
```
# Handle missing categorical data
```

In [6]:
```
categorical_cols = df.select_dtypes(include='object').columns

for col in categorical_cols:
    if df[col].isnull().sum() > 0:
        df[col] = df[col].fillna(f"Missing {col}")
```

In [7]:
```
# Verify cleaning
```

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
Rank                          0
NCT Number                    0
Title                         0
Acronym                       0
Status                        0
Study Results                 0
Conditions                    0
Interventions                 0
Outcome Measures              0
Sponsor/Collaborators         0
Gender                        0
Age                           0
Phases                        0
Enrollment                   34
Funded Bys                    0
Study Type                    0
Study Designs                 0
Other IDs                     0
Start Date                    0
Primary Completion Date       0
Completion Date               0
First Posted                  0
Last Update Posted            0
Locations                     0
URL                           0
dtype: int64
```

In [9]:
```
# Filling missing numeric data in Enrollment with median
```

In [10]:
```
median_enrollment = df['Enrollment'].median()
df['Enrollment'] = df['Enrollment'].fillna(median_enrollment)
```

In [11]:
```
df.isnull().sum()
```

Out[11]:
```
Rank                        0
NCT Number                  0
Title                       0
Acronym                     0
Status                      0
Study Results               0
Conditions                  0
Interventions               0
Outcome Measures            0
Sponsor/Collaborators       0
Gender                      0
Age                         0
Phases                      0
Enrollment                  0
Funded Bys                  0
Study Type                  0
Study Designs               0
Other IDs                   0
Start Date                  0
Primary Completion Date     0
Completion Date             0
First Posted                0
Last Update Posted          0
Locations                   0
URL                         0
dtype: int64
```

In [12]:
```
# Extract country name from Locations column
```

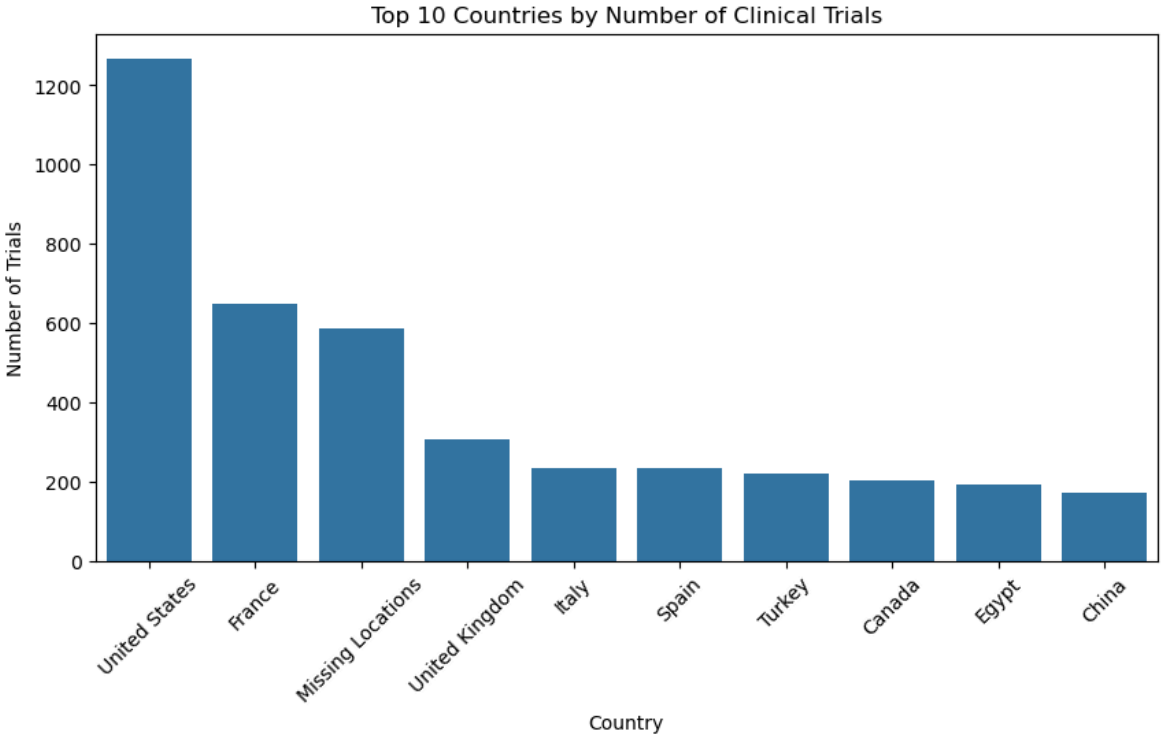
In [13]:
```
df['Country'] = df['Locations'].apply(lambda x: str(x).split(',')[-1].strip())
```

In [14]:
```
df['Country'].value_counts().head(10)
```

Out[14]:
```
Country
United States        1267
France                647
Missing Locations     585
United Kingdom        306
Italy                 235
Spain                 234
Turkey                219
Canada                202
Egypt                 192
China                 171
Name: count, dtype: int64
```

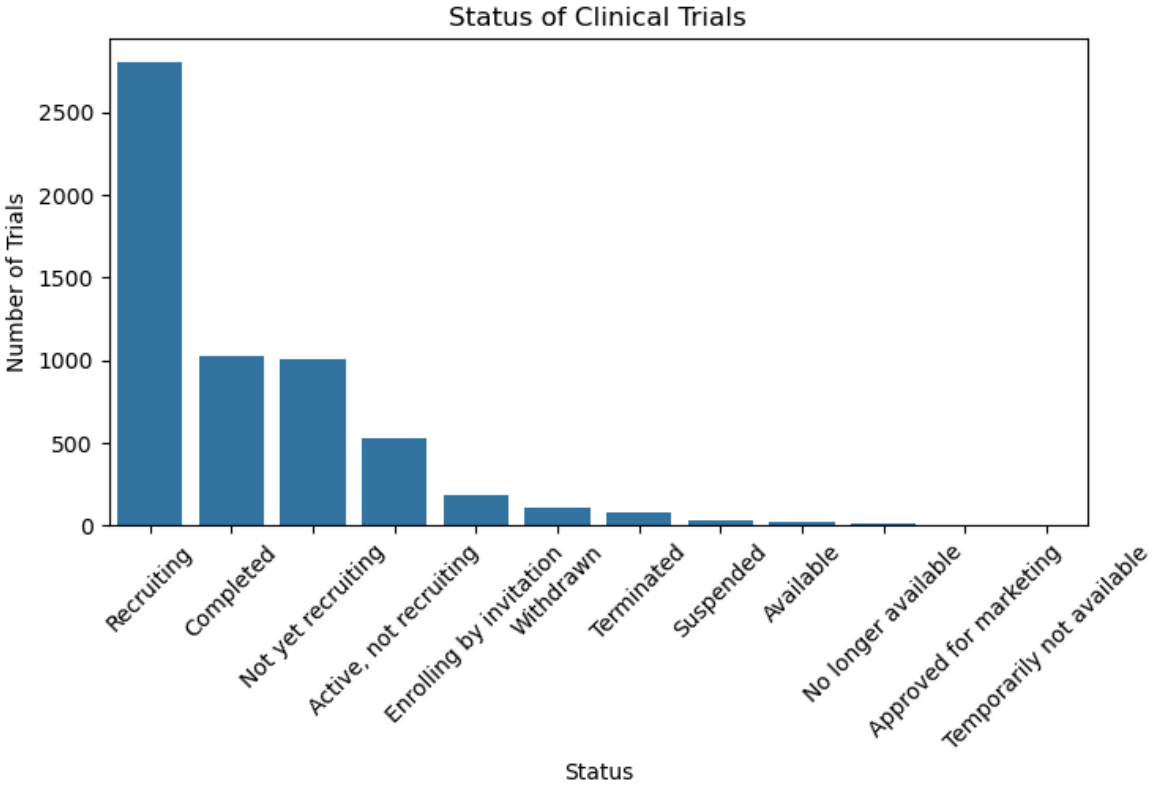
In [15]:
```
# Univariate Analysis
```

In [16]:
```
import matplotlib.pyplot as plt
import seaborn as sns

top_countries = df['Country'].value_counts().head(10)

plt.figure(figsize=(10,5))
sns.barplot(x=top_countries.index, y=top_countries.values)
plt.title('Top 10 Countries by Number of Clinical Trials')
plt.ylabel('Number of Trials')
plt.xticks(rotation=45)
plt.show()
```



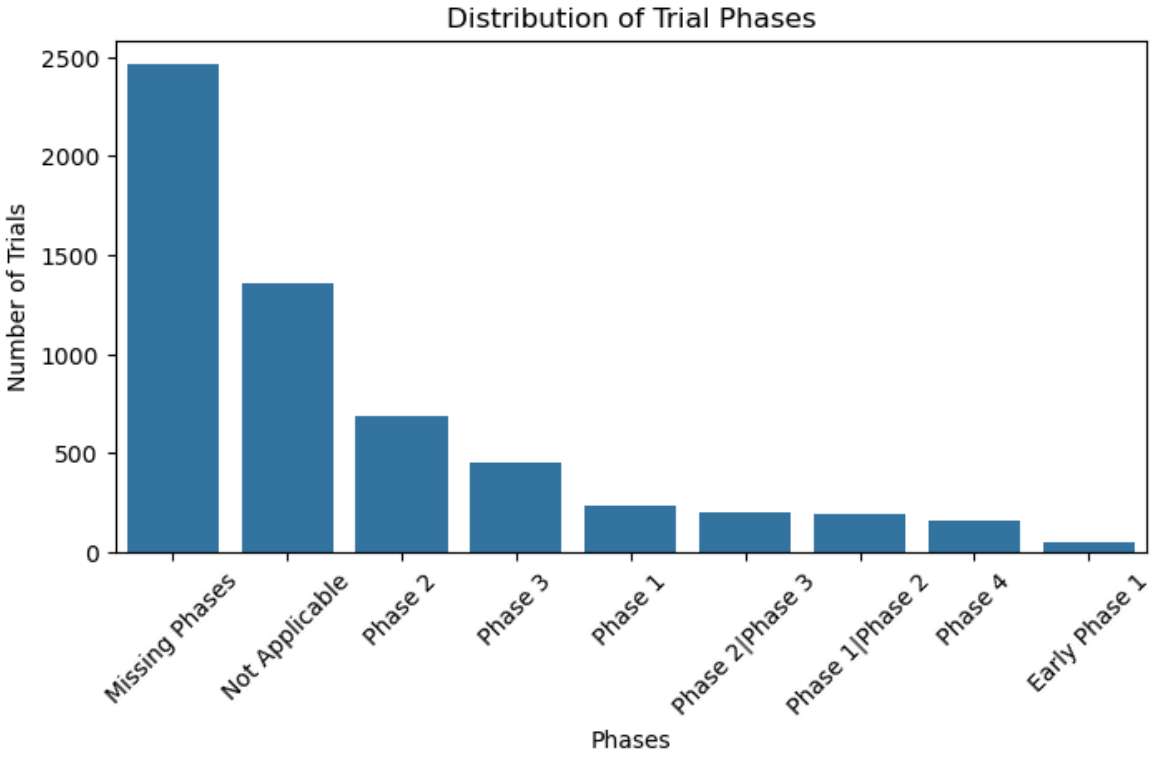
In [17]:
```
# Status distribution of trials
```

In [18]:
```
status_counts = df['Status'].value_counts()

plt.figure(figsize=(8,4))
sns.barplot(x=status_counts.index, y=status_counts.values)
plt.title('Status of Clinical Trials')
plt.ylabel('Number of Trials')
plt.xticks(rotation=45)
plt.show()
```



In [19]:
```
# Phases distributing
```

In [20]:
```
phase_counts = df['Phases'].value_counts()

plt.figure(figsize=(8,4))
sns.barplot(x=phase_counts.index, y=phase_counts.values)
plt.title('Distribution of Trial Phases')
plt.ylabel('Number of Trials')
plt.xticks(rotation=45)
plt.show()
```

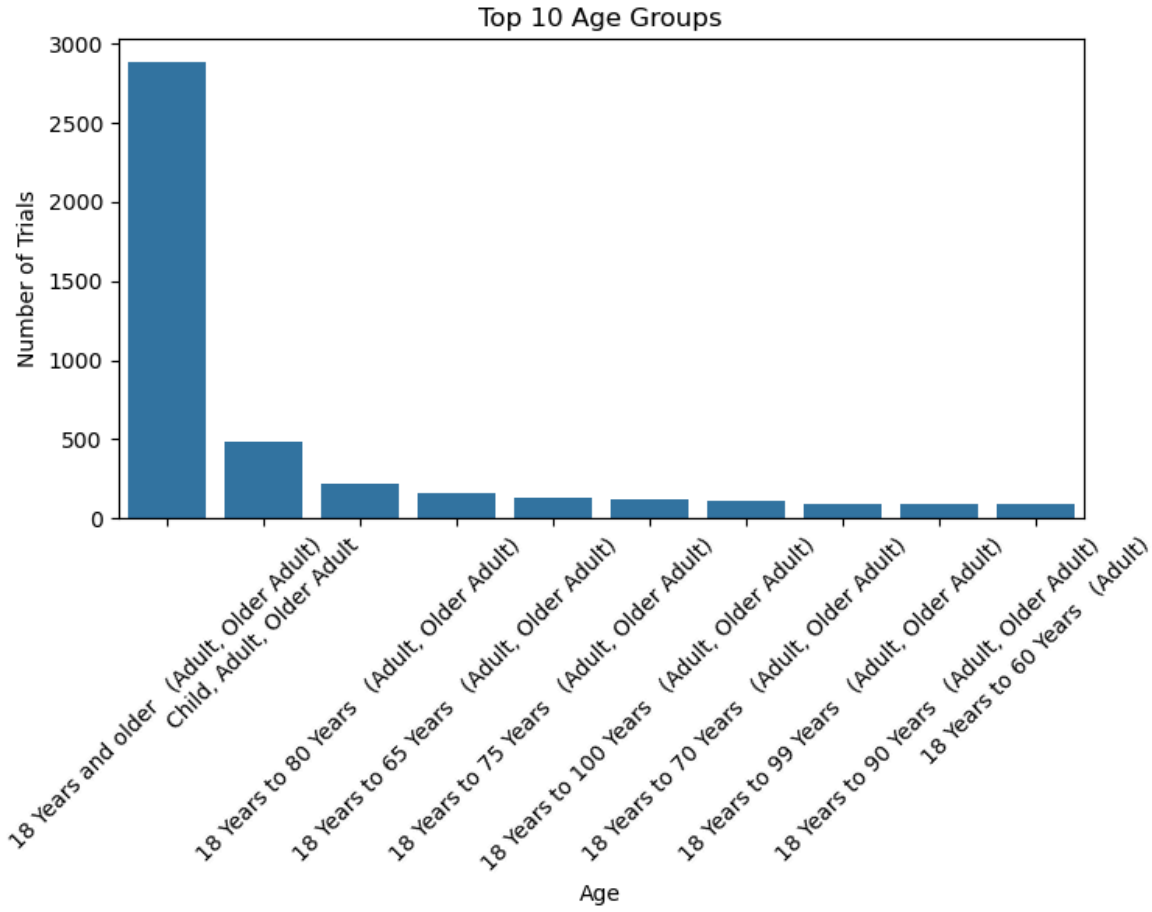


In [21]:
```
Age group distribution
```

```
  Cell In[21], line 1
    Age group distribution
        ^
SyntaxError: invalid syntax
```

In [22]:
```
# Age group distribution
```

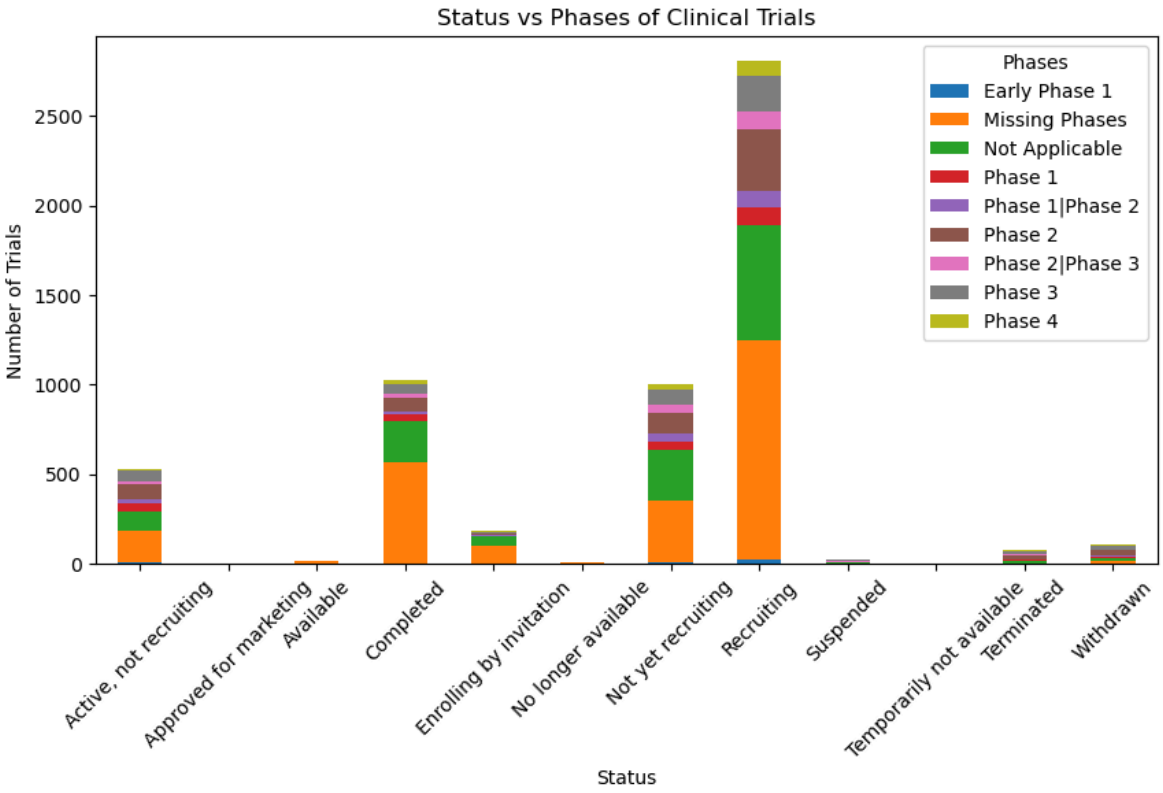
In [23]:
```
age_counts = df['Age'].value_counts().head(10)

plt.figure(figsize=(8,4))
sns.barplot(x=age_counts.index, y=age_counts.values)
plt.title('Top 10 Age Groups')
plt.ylabel('Number of Trials')
plt.xticks(rotation=45)
plt.show()
```



In [24]:
```
# Status vs. Phases
```

In [25]:
```
status_phase = pd.crosstab(df['Status'], df['Phases'])

plt.figure(figsize=(10,5))
status_phase.plot(kind='bar', stacked=True, figsize=(10,5))
plt.title('Status vs Phases of Clinical Trials')
plt.ylabel('Number of Trials')
plt.xticks(rotation=45)
plt.show()
```

```
<Figure size 1000x500 with 0 Axes>
```



In [26]:
```
conditions_outcomes = df.groupby('Conditions')['Outcome Measures'] \
                        .apply(lambda x: ', '.join(x.astype(str))) \
                        .reset_index()

conditions_outcomes.head()
```

Out[26]:

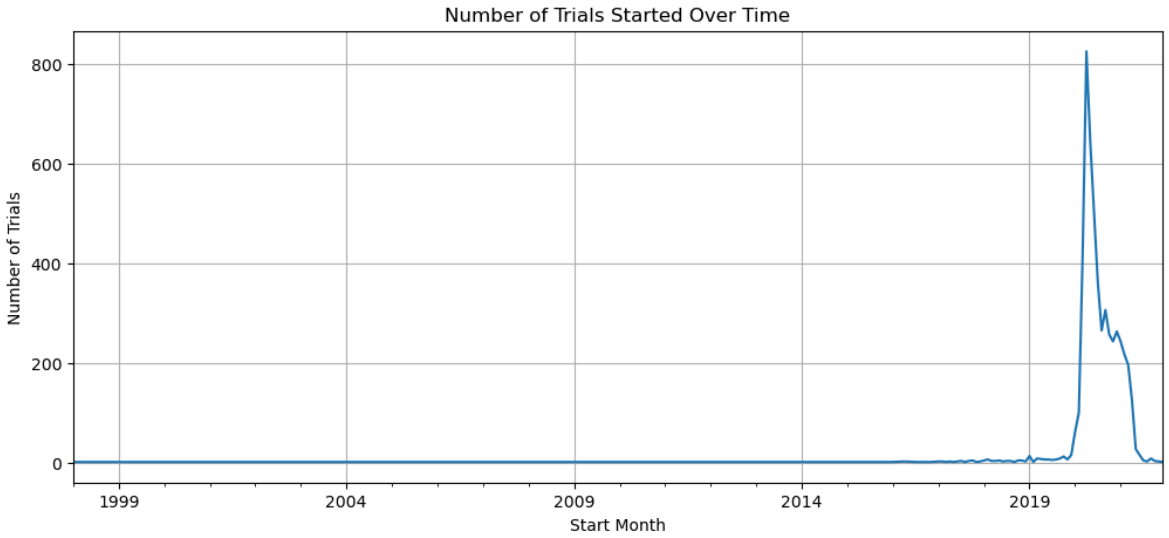
	Conditions	Outcome Measures
0	2019 Novel Coronavirus	Proportion of participants who improve by at l...
1	2019 Novel Coronavirus Infection	new-onset COVID-19|Number of Participants with...
2	2019 Novel Coronavirus Infection|COVID-19 Viru...	Number of participants with treatment emergent...
3	2019 Novel Coronavirus Pneumonia	Clinical recovery time|Complete fever time|Cou...
4	2019 Novel Coronavirus Pneumonia|COVID-19	Pneumonia severity index|Oxygenation index (Pa...

In [27]:
```
# Time Series Analysis
```

In [28]:
```
df['Start Date'] = pd.to_datetime(df['Start Date'], errors='coerce')
```

In [29]:
```
trials_over_time = df['Start Date'].dt.to_period('M').value_counts().sort_index(

plt.figure(figsize=(12,5))
trials_over_time.plot(kind='line')
plt.title('Number of Trials Started Over Time')
plt.ylabel('Number of Trials')
plt.xlabel('Start Month')
```

```
plt.grid(True)
plt.show()
```



In [30]:
```
trials_over_time.sort_values(ascending=False).head(10)
```

Out[30]:
```
Start Date
2020-04    825
2020-05    645
2020-06    502
2020-03    417
2020-07    361
2020-09    306
2020-08    265
2020-12    263
2020-10    257
2020-11    243
Freq: M, Name: count, dtype: int64
```

In [31]:
```
df.to_csv('cleaned_covid_Nidhi.csv', index=False)
print("Cleaned dataset saved successfully!")
```

```
Cleaned dataset saved successfully!
```

In [ ]: