```
import pandas as pd

# Load the data
df = pd.read_csv('COVID clinical trials.csv')

# Inspect the data
print(df.info())
print(df.head())

# Check unique values and distributions
print("Unique Phases counts:")
print(df['Phases'].value_counts(dropna=False))

print("\nUnique Status counts:")
print(df['Status'].value_counts(dropna=False))

# Look at Conditions
print("\nTop 10 Conditions:")
print(df['Conditions'].value_counts().head(10))

# Sample values for Start Date to check format
print("\nSample Start Dates:")
print(df['Start Date'].head())
```

```
import matplotlib.pyplot as plt
import seaborn as sns

# Convert Start Date to datetime
df['Start Date'] = pd.to_datetime(df['Start Date'], errors='coerce')
df['Start Year'] = df['Start Date'].dt.year

# Distribution of Status
status_dist = df['Status'].value_counts()

# Distribution of Phases
phase_dist = df['Phases'].fillna('Missing').value_counts()

# Conditions analysis
# Flatten conditions and normalize
all_conditions = df['Conditions'].str.split('|').explode().str.strip()
# Standardize common COVID names
covid_variants = ['COVID-19', 'Covid19', 'COVID', 'Covid-19', 'SARS-CoV-2',
'Coronavirus Infection',
                'Coronavirus', 'COVID 19', 'COVID19', 'Corona Virus Infection', 'SARS-
CoV2', 'SARS-CoV 2']
all_conditions_norm = all_conditions.apply(lambda x: 'COVID-19' if x in
covid_variants else x)
top_conditions = all_conditions_norm.value_counts().head(15)

# 1. Plot Phase Distribution
plt.figure(figsize=(10, 6))
sns.barplot(x=phase_dist.index, y=phase_dist.values, palette='viridis')
plt.title('Distribution of Clinical Trials by Phase')
plt.xticks(rotation=45)
plt.ylabel('Count')
plt.savefig('phase_distribution.png')

# 2. Plot Status Distribution
plt.figure(figsize=(10, 6))
sns.barplot(x=status_dist.index, y=status_dist.values, palette='magma')
plt.title('Distribution of Clinical Trials by Status')
plt.xticks(rotation=45)
plt.ylabel('Count')
plt.savefig('status_distribution.png')

# 3. Plot Top Conditions
plt.figure(figsize=(10, 6))
sns.barplot(x=top_conditions.values, y=top_conditions.index, palette='rocket')
plt.title('Top 15 Conditions (Therapeutic Areas)')
plt.xlabel('Count')
plt.savefig('top_conditions.png')

# 4. Evolution over time
# We'll focus on the count of trials starting each year
evolution_year = df.groupby('Start Year').size().reset_index(name='Trial Count')
evolution_year = evolution_year[evolution_year['Start Year'] >= 2019] # Focus
on the pandemic era

plt.figure(figsize=(10, 6))
sns.lineplot(data=evolution_year, x='Start Year', y='Trial Count', marker='o')
plt.title('Evolution of Clinical Trials Over Time (Start Year)')
plt.xticks([2019, 2020, 2021, 2022, 2023, 2024, 2025])
plt.grid(True)
plt.savefig('evolution_time.png')

# 5. Evolution of Phase by Year
phase_evolution = df[df['Start Year'] >= 2019].groupby(['Start Year',
'Phases']).size().unstack().fillna(0)
```

```
# We might want to filter or group some phases to make the chart clearer
plt.figure(figsize=(12, 7))
phase_evolution.plot(kind='bar', stacked=True, ax=plt.gca())
plt.title('Evolution of Trial Phases Over Time')
plt.ylabel('Count')
plt.legend(title='Phase', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.savefig('phase_evolution.png')

print("Charts generated.")
```