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

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

# Function to extract country from location string
def extract_countries(location_str):
    if pd.isna(location_str):
        return []
    # Split by '|' for multiple locations
    locs = location_str.split('|')
    countries = []
    for loc in locs:
        parts = loc.split(',')
        if len(parts) > 0:
            country = parts[-1].strip()
            countries.append(country)
    return list(set(countries)) # unique countries per trial

# Apply extraction
df['Countries'] = df['Locations'].apply(extract_countries)

# Explode to get one row per country per trial
df_geo = df.explode('Countries')

# 1. Global Distribution by Country
country_counts = df_geo['Countries'].value_counts().head(20)

# 2. Regional Specializations
# Standardize conditions as before
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']

def standardize_condition(cond):
    if pd.isna(cond): return 'Missing'
    conds = [c.strip() for c in cond.split('|')]
    std_conds = ['COVID-19' if c in covid_variants else c for c in conds]
    return std_conds

df_geo['Std_Conditions'] = df['Conditions'].apply(standardize_condition)
```

```
df_geo_cond = df_geo.explode('Std_Conditions')

# Filter for top 5 countries to see specializations
top_5_countries = country_counts.head(5).index.tolist()
specialization = df_geo_cond[df_geo_cond['Countries'].isin(top_5_countries)]

# Visualizations
plt.figure(figsize=(12, 8))
sns.barplot(x=country_counts.values, y=country_counts.index,
palette='plasma')
plt.title('Top 20 Countries by Number of Clinical Trials')
plt.xlabel('Number of Trials')
plt.savefig('trials_by_country.png')

# Specialized heatmap for Top 5 Countries vs Top Non-COVID Conditions
# Get top conditions excluding the generic COVID-19 ones
top_conditions = df_geo_cond[df_geo_cond['Std_Conditions'] != 'COVID-19']
['Std_Conditions'].value_counts().head(10).index
pivot_specialization =
specialization[specialization['Std_Conditions'].isin(top_conditions)].pivot_table(
    index='Std_Conditions', columns='Countries', aggfunc='size', fill_value=0
)

plt.figure(figsize=(12, 8))
sns.heatmap(pivot_specialization, annot=True, fmt='d', cmap='YlGnBu')
plt.title('Specialization: Top Non-COVID Conditions in Top 5 Countries')
plt.tight_layout()
plt.savefig('geo_specialization_heatmap.png')

print("Top 10 Countries:")
print(country_counts.head(10))
```