In [1]:
```
!pip install pandas matplotlib seaborn
```

```
Requirement already satisfied: pandas in c:\users\asus\miniconda3\lib\site-packages
(2.3.1)
Requirement already satisfied: matplotlib in c:\users\asus\miniconda3\lib\site-packa
ges (3.10.3)
Collecting seaborn
  Using cached seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Requirement already satisfied: numpy>=1.26.0 in c:\users\asus\miniconda3\lib\site-pa
ckages (from pandas) (2.3.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\asus\miniconda3\li
b\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\asus\miniconda3\lib\site-pac
kages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\asus\miniconda3\lib\site-p
ackages (from pandas) (2025.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\asus\miniconda3\lib\site
-packages (from matplotlib) (1.3.2)
Requirement already satisfied: cycler>=0.10 in c:\users\asus\miniconda3\lib\site-pac
kages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\asus\miniconda3\lib\sit
e-packages (from matplotlib) (4.59.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\asus\miniconda3\lib\sit
e-packages (from matplotlib) (1.4.8)
Requirement already satisfied: packaging>=20.0 in c:\users\asus\miniconda3\lib\site-
packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in c:\users\asus\miniconda3\lib\site-packag
es (from matplotlib) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\asus\miniconda3\lib\site
-packages (from matplotlib) (3.2.3)
Requirement already satisfied: six>=1.5 in c:\users\asus\miniconda3\lib\site-package
s (from python-dateutil>=2.8.2->pandas) (1.17.0)
Using cached seaborn-0.13.2-py3-none-any.whl (294 kB)
Installing collected packages: seaborn
Successfully installed seaborn-0.13.2
```

In [ ]:
```
```

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

# Set Seaborn style for plots
sns.set(style="whitegrid")
```

In [4]:
```
# Read the file and skip the first 4 rows which contain metadata
df = pd.read_csv("API_SP.POP.TOTL_DS2_en_csv_v2_38144.csv", skiprows=4)

# Show the first few rows
df.head()
```

Out[4]:

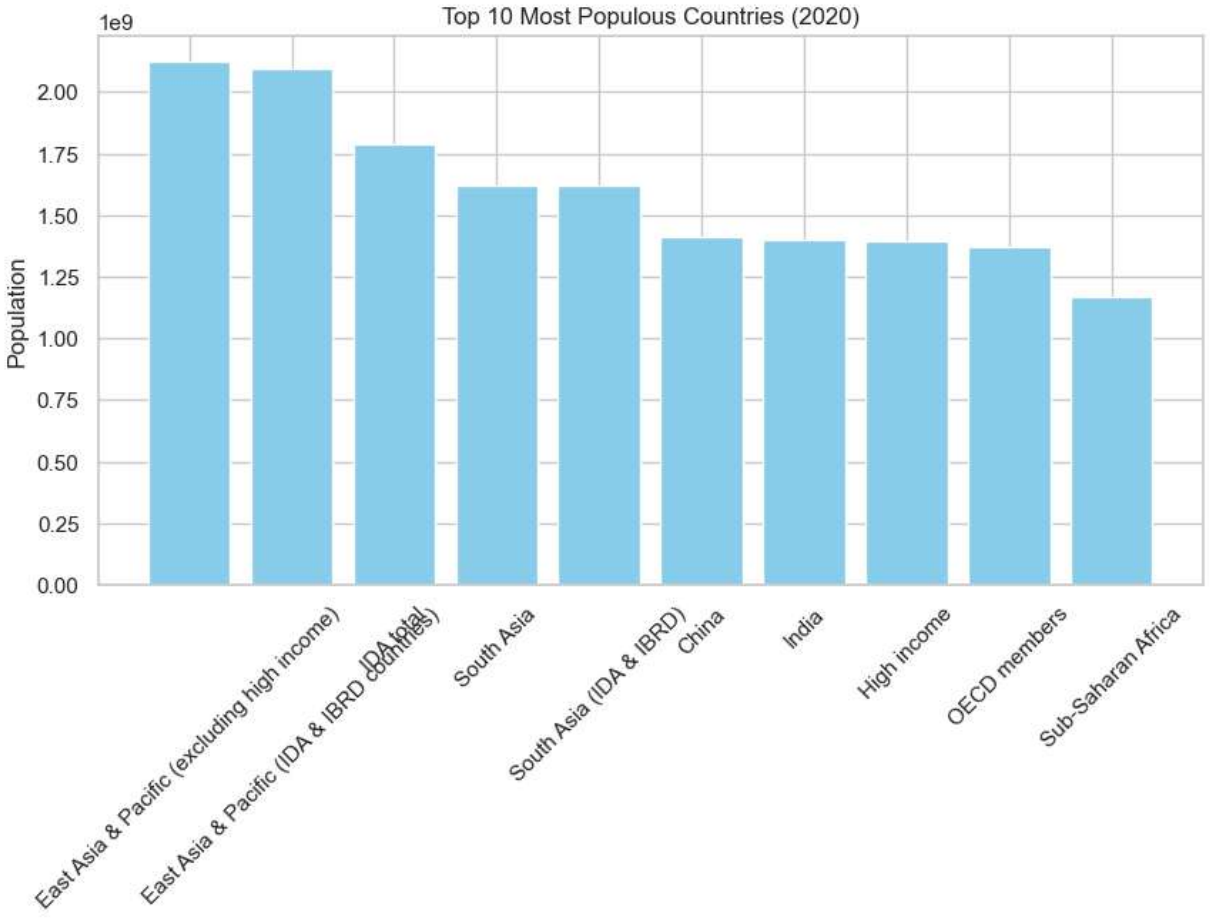
| | Country Name | Country Code | Indicator Name | Indicator Code | 1960 | 1961 | 1962 | |
|---|---|---|---|---|---|---|---|---|
| **0** | Aruba | ABW | Population, total | SP.POP.TOTL | 54922.0 | 55578.0 | 56320.0 | |
| **1** | Africa Eastern and Southern | AFE | Population, total | SP.POP.TOTL | 130075728.0 | 133534923.0 | 137171659.0 | 1 |
| **2** | Afghanistan | AFG | Population, total | SP.POP.TOTL | 9035043.0 | 9214083.0 | 9404406.0 | |
| **3** | Africa Western and Central | AFW | Population, total | SP.POP.TOTL | 97630925.0 | 99706674.0 | 101854756.0 | 1 |
| **4** | Angola | AGO | Population, total | SP.POP.TOTL | 5231654.0 | 5301583.0 | 5354310.0 | |

5 rows × 70 columns

In [14]:
```
# Keep only the columns we need
df_simple = df[["Country Name", "2020"]].dropna()
df_simple.columns = ["Country", "Population"]

# Convert population to integer
df_simple["Population"] = df_simple["Population"].astype(int)
```

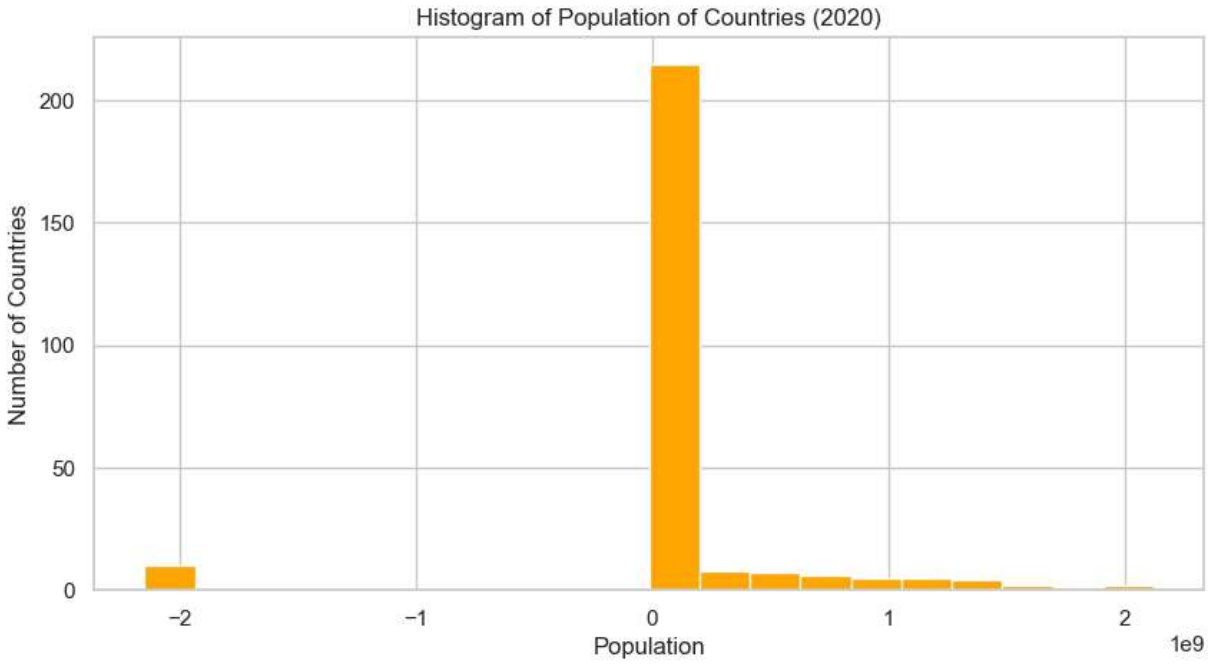
# 1. Bar Chart: Top 10 populous countries

In [15]:
```
top10 = df_simple.sort_values(by="Population", ascending=False).head(10)
plt.figure(figsize=(10,5))
plt.bar(top10["Country"], top10["Population"], color='skyblue')
plt.title("Top 10 Most Populous Countries (2020)")
plt.xticks(rotation=45)
plt.ylabel("Population")
plt.show()
```



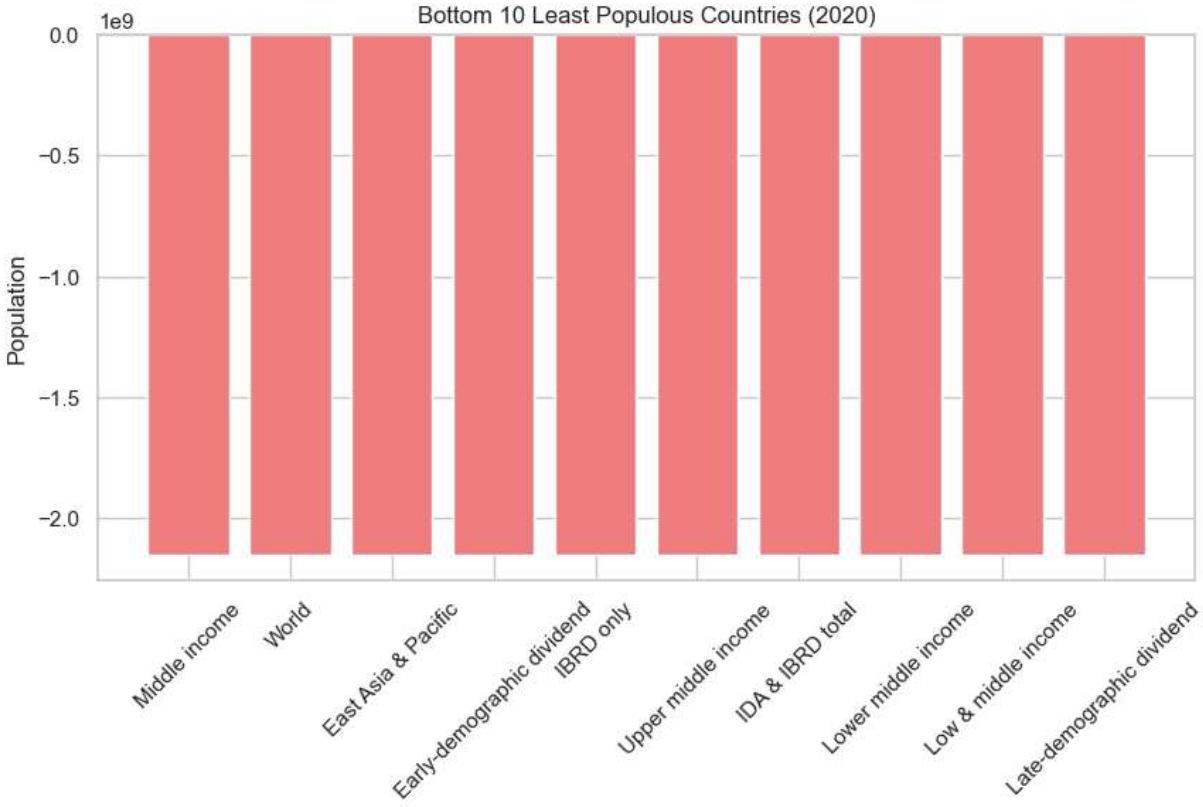

# 2. Histogram: Distribution of country populations

In [16]:
```
plt.figure(figsize=(10,5))
plt.hist(df_simple["Population"], bins=20, color='orange')
plt.title("Histogram of Population of Countries (2020)")
plt.xlabel("Population")
plt.ylabel("Number of Countries")
plt.show()
```



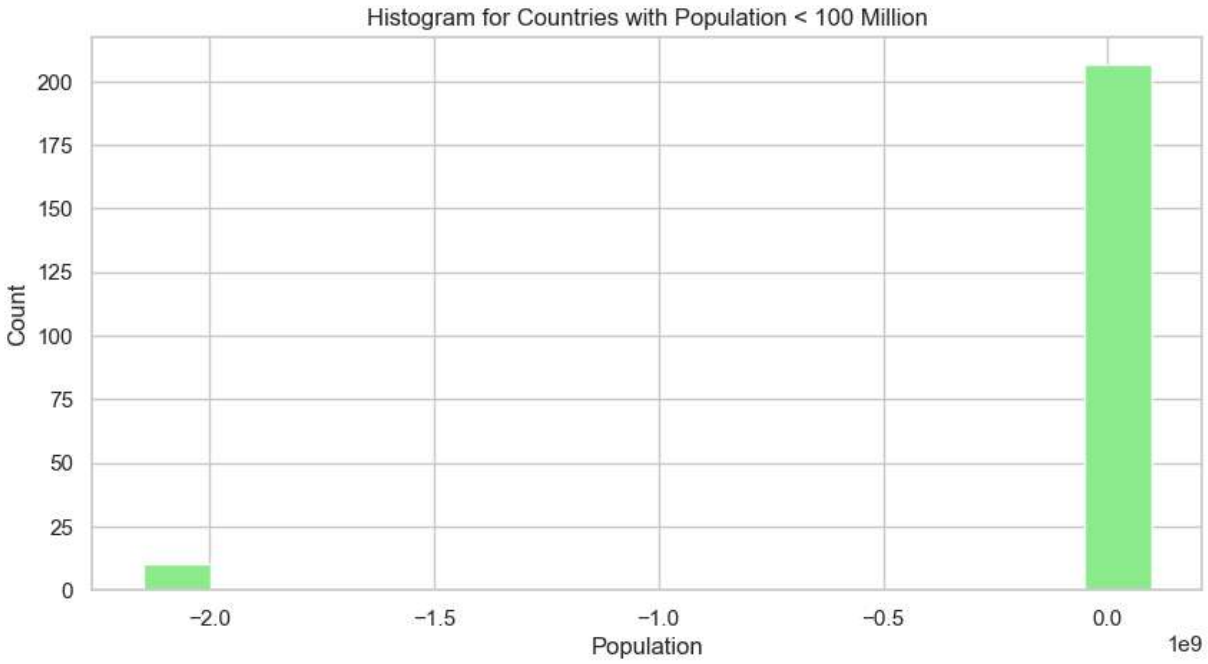

# 3. Bar Chart: Bottom 10 least populous countries

In [17]:
```
bottom10 = df_simple.sort_values(by="Population").head(10)
plt.figure(figsize=(10,5))
plt.bar(bottom10["Country"], bottom10["Population"], color='lightcoral')
plt.title("Bottom 10 Least Populous Countries (2020)")
plt.xticks(rotation=45)
plt.ylabel("Population")
plt.show()
```



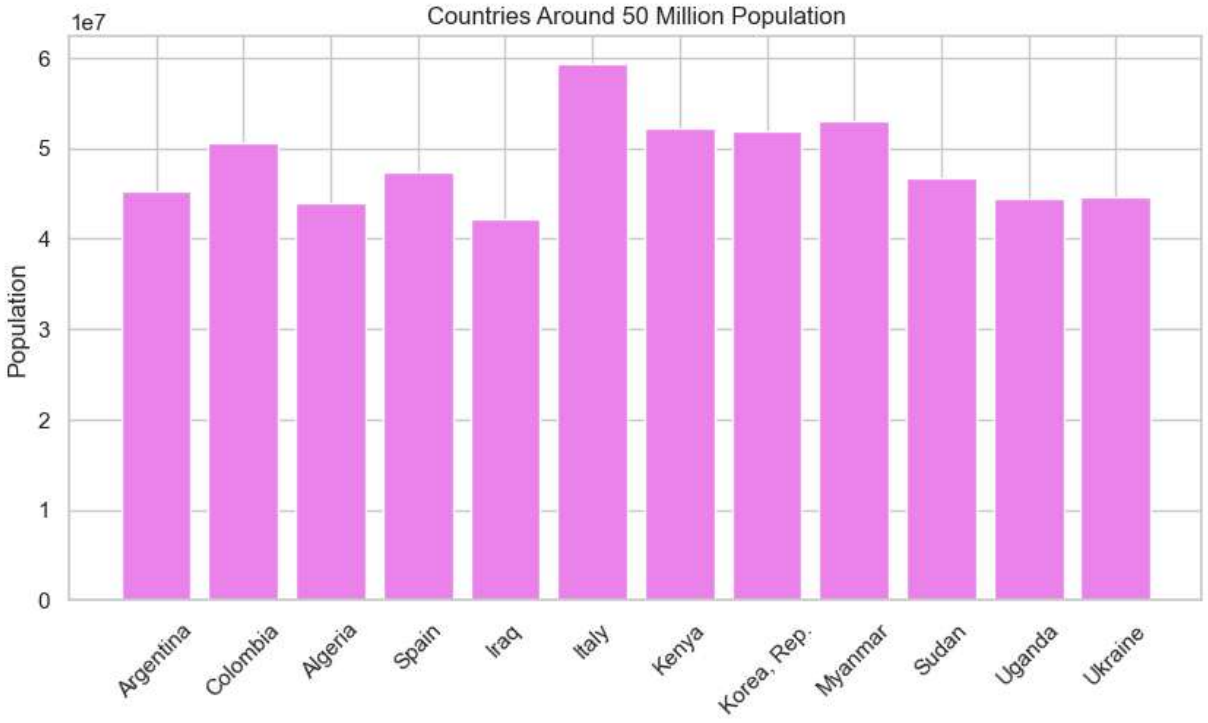

# 4. Histogram: Countries with population < 100 million

In [18]:
```
small_pop = df_simple[df_simple["Population"] < 100_000_000]
plt.figure(figsize=(10,5))
plt.hist(small_pop["Population"], bins=15, color='lightgreen')
plt.title("Histogram for Countries with Population < 100 Million")
plt.xlabel("Population")
plt.ylabel("Count")
plt.show()
```



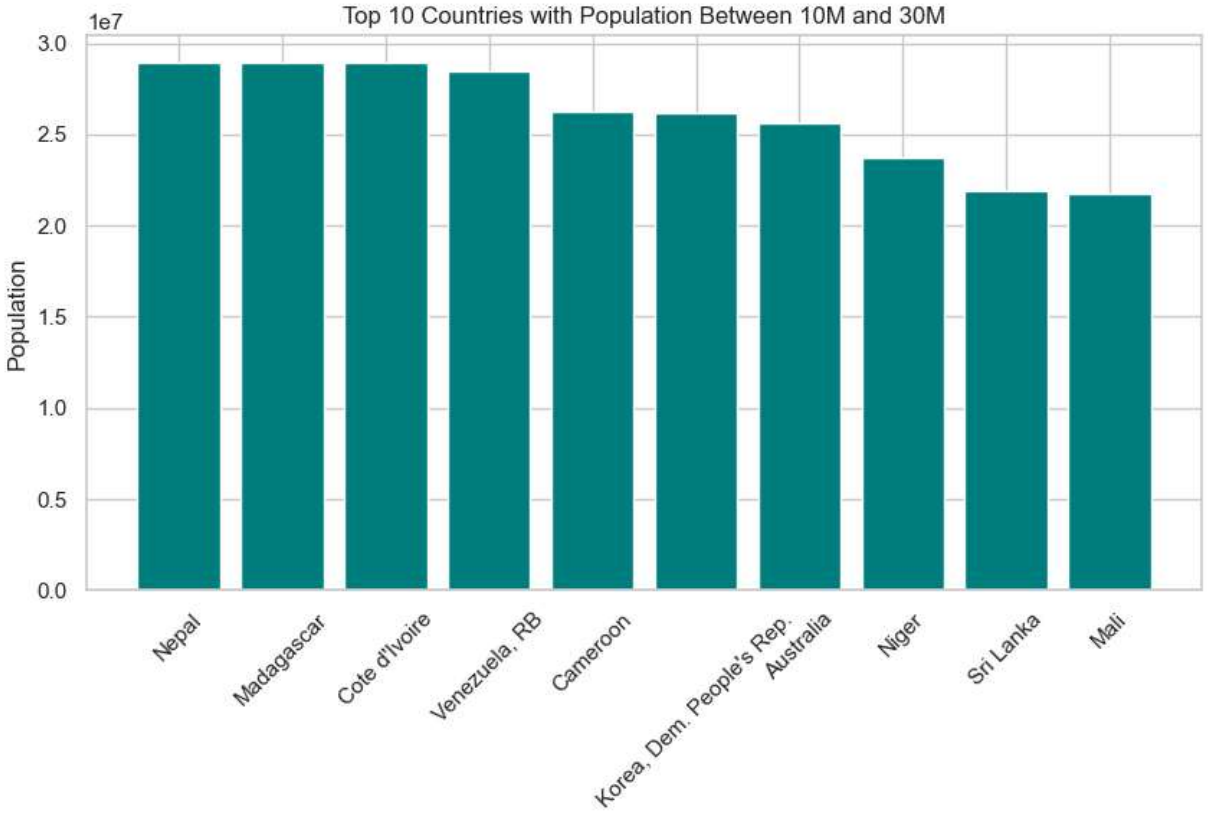

# 5. Bar Chart: Countries with ~50 million population

In [19]:
```
mid_pop = df_simple[(df_simple["Population"] > 40_000_000) & (df_simple["Population
plt.figure(figsize=(10,5))
plt.bar(mid_pop["Country"], mid_pop["Population"], color='violet')
plt.title("Countries Around 50 Million Population")
plt.xticks(rotation=45)
plt.ylabel("Population")
plt.show()
```



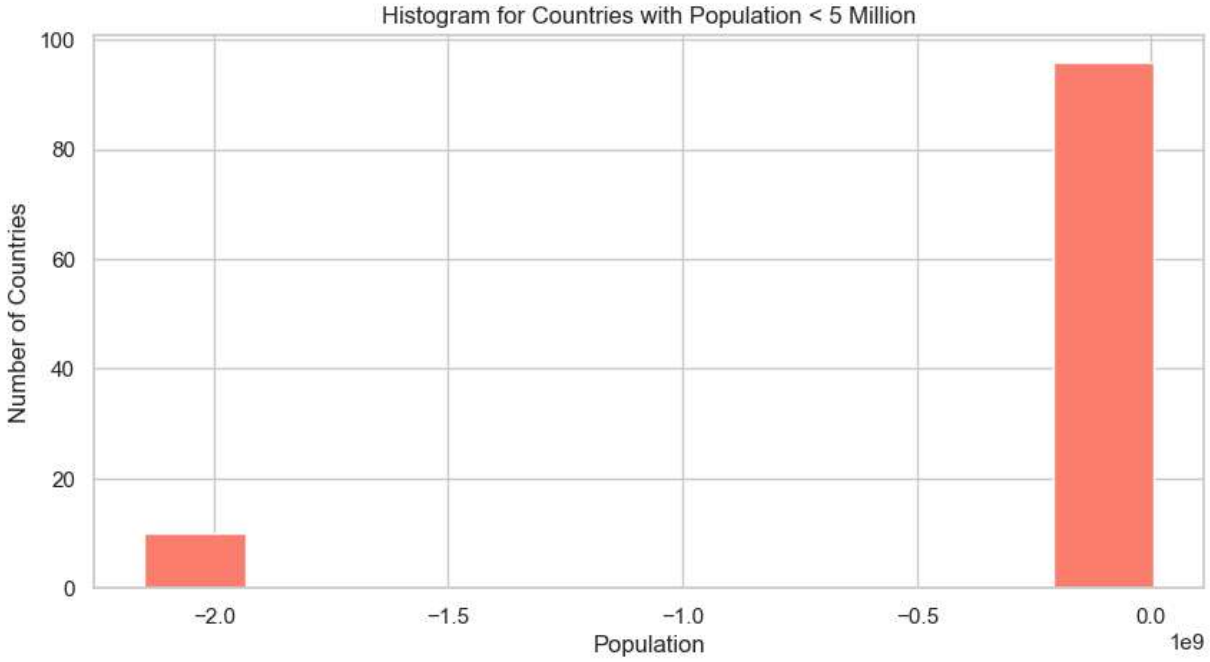

# 6. Bar Chart: Top 10 countries with population between 10M and 30M

In [23]:
```
range_pop = df_simple[(df_simple["Population"] >= 10_000_000) & (df_simple["Populat
top10_range = range_pop.sort_values(by="Population", ascending=False).head(10)
plt.figure(figsize=(10,5))
plt.bar(top10_range["Country"], top10_range["Population"], color='teal')
plt.title("Top 10 Countries with Population Between 10M and 30M")
plt.xticks(rotation=45)
plt.ylabel("Population")
plt.show()
```



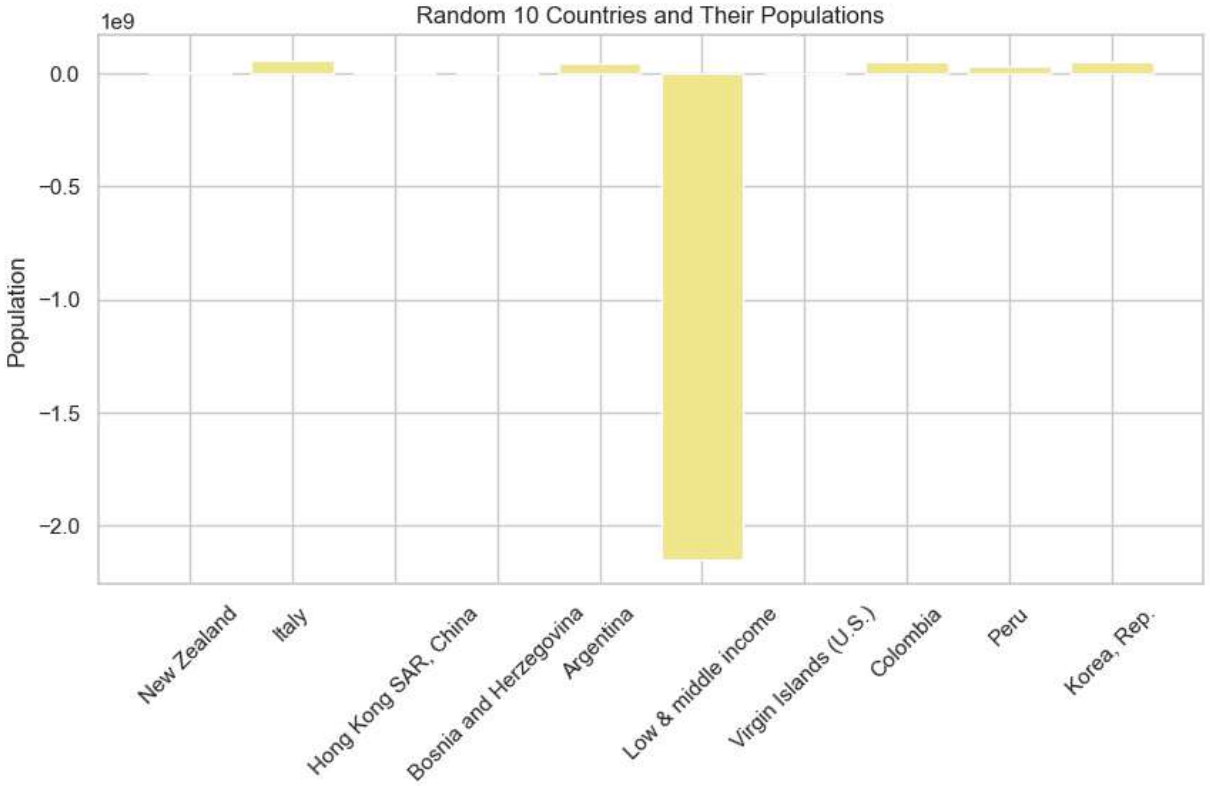

# 7. Histogram: Countries with < 5 million population

In [24]:
```
low_pop = df_simple[df_simple["Population"] < 5_000_000]
plt.figure(figsize=(10,5))
plt.hist(low_pop["Population"], bins=10, color='salmon')
plt.title("Histogram for Countries with Population < 5 Million")
plt.xlabel("Population")
plt.ylabel("Number of Countries")
plt.show()
```




# 8. Bar Chart: Random 10 countries' populations

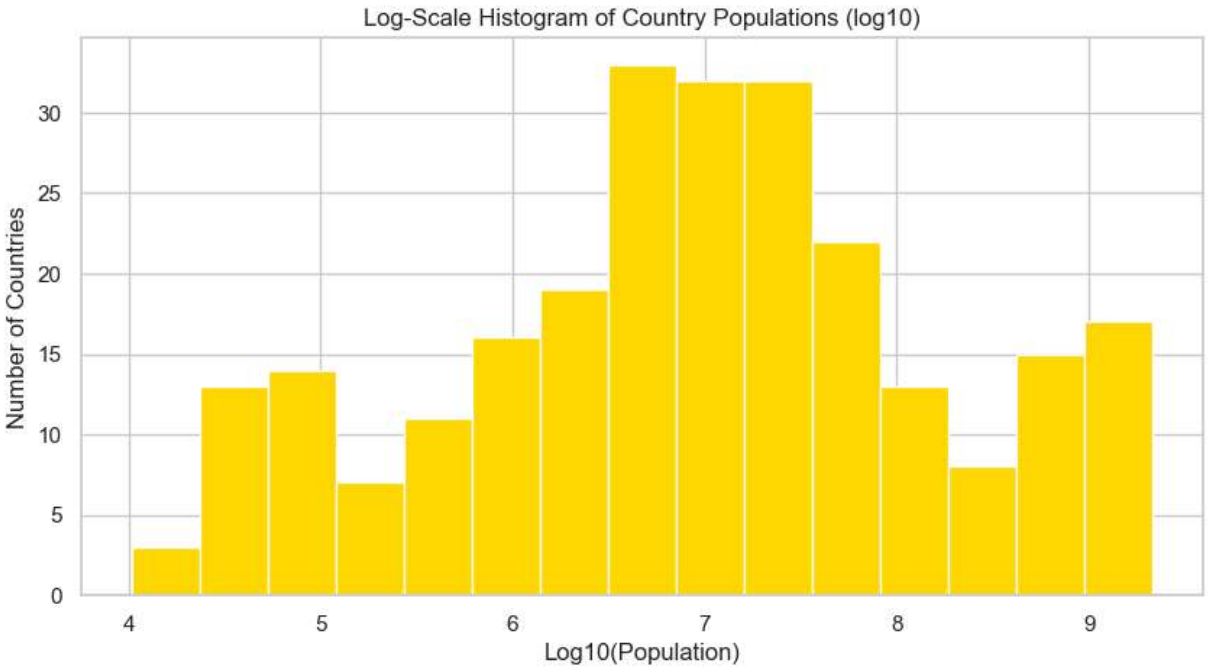
In [26]:
```
# 8. Histogram: Countries with population between 10 million and 200 million
pop_filtered = df_simple[(df_simple["Population"] >= 10_000_000) & (df_simple["Popu

plt.figure(figsize=(10, 5))
plt.hist(pop_filtered["Population"], bins=10, color='cyan', edgecolor='black')
plt.title("Histogram for Countries with Population Between 10M and 200M (2020)")
plt.xlabel("Population")
plt.ylabel("Number of Countries")
plt.grid(True)
plt.tight_layout()
plt.show()
```




# 9. Histogram: Log-scale population to show global spread clearly

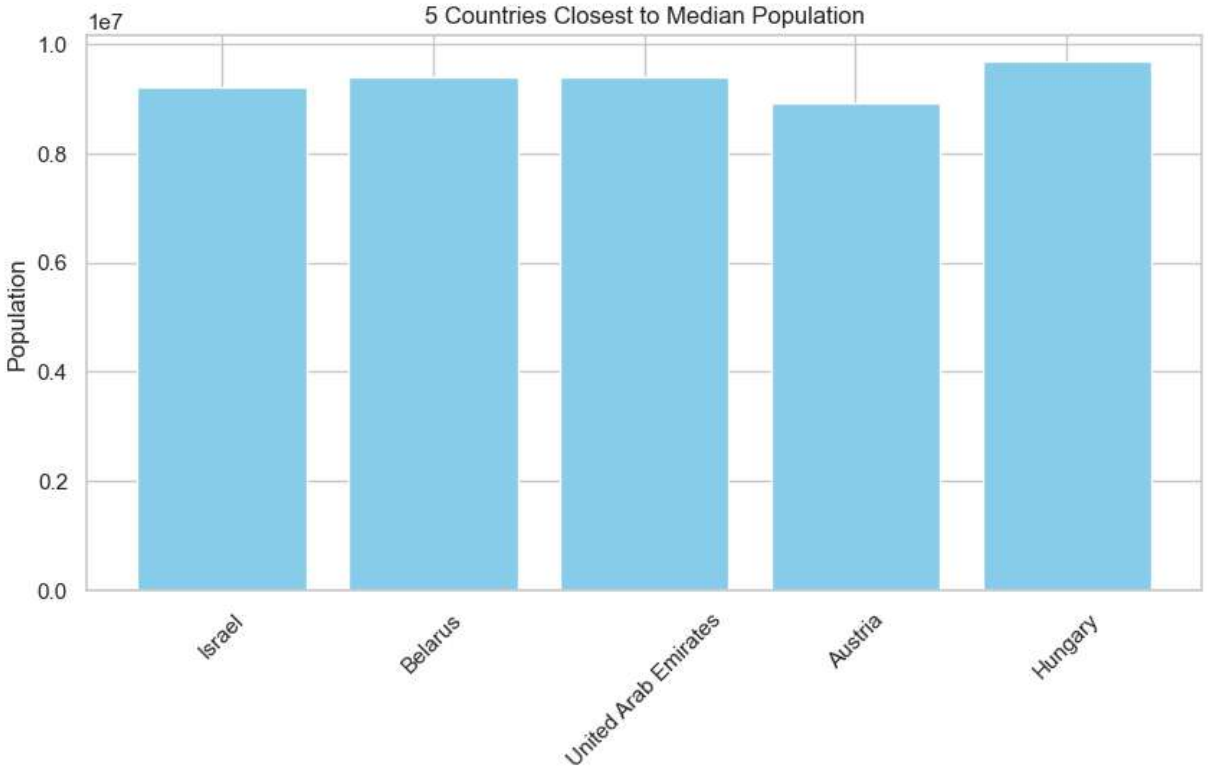
In [29]:
```
import numpy as np
plt.figure(figsize=(10,5))
plt.hist(np.log10(df_simple["Population"]), bins=15, color='gold')
plt.title("Log-Scale Histogram of Country Populations (log10)")
plt.xlabel("Log10(Population)")
plt.ylabel("Number of Countries")
plt.show()
```

```
C:\Users\ASUS\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core
\arraylike.py:399: RuntimeWarning: invalid value encountered in log10
  result = getattr(ufunc, method)(*inputs, **kwargs)
```




# 10. Bar Chart: 5 countries closest to median population

In [28]:
```
median_val = df_simple["Population"].median()
df_simple["diff_from_median"] = abs(df_simple["Population"] - median_val)
closest_to_median = df_simple.sort_values(by="diff_from_median").head(5)
plt.figure(figsize=(10,5))
plt.bar(closest_to_median["Country"], closest_to_median["Population"], color='skybl
plt.title("5 Countries Closest to Median Population")
plt.xticks(rotation=45)
plt.ylabel("Population")
plt.show()
```


5 Countries Closest to Median Population
1e7
1.0
0.8
0.6
0.4
0.2
0.0
Population
Israel
Belarus
United Arab Emirates
Austria
Hungary