1. Aim of the project

1.1 Introduction

In the light of significant developments within the Catholic Church, such as the recent approval of Carlo Acutis as the first millennial saint and the controversy surrounding Pope Francis's approval in the blessing of individuals in a same-sex relationships, there seems to be a palpable sense of change and also transformation within the Catholic Church. Hence the decesion to explore this topic was enhanced by a series of personal experiences that has sparked in me a profound curiosity about the evolving landscape in the field of catholic demographics.

During a recent visit to vietnam, i was baffled at the number of catholics present at a sunday mass.The visit not only sparked curiosity but also provided me with a firsthand exposure to the vastness and diversity of Catholic communities, highlighting the global reach of the Church and complex interplay of cultural, social and religious factors that shape its demographic makeup. Witnessing this grace and vibrancy of Catholism in a country with a rich tapestry of traditions and histories served as a catalyst for deeper exploration into the demographic shifts occuring within the church.

Through this project, i will be able to also explore various concepts such as statistical analysis and geospatial analysis to create rubrics to see how varying factors play a vital role in the evolving nature of Catholism in the mordern world.

These recent developments, coupled with personal observations and reflections, underscore the relevance and timeliness of investigating Catholic demographics. By delving into the demographic changes within the Catholic Church, this project aims to unravel the intricate dynamics at play, shed light on emerging trends, and contribute to a deeper understanding of the changes in the catholic heritage in the modern world.

1.2 Aim and Objectives

Aim: The aim of this project is to conduct a comprehensive analysis of demographic changes within the global Catholic Church over the past 50 years.

Objectives:

1. Examine Historical Trends: Analyze historical data to understand how the demographics of the Catholic population have evolved over the past five decades.
2. Identify Regional Variations: Identify regions that have experienced significant growth or decline in Catholic adherents and explore the factors contributing to these trends.
3. Investigate Economic, Social and political Correlations: Investigate the correlation between various factors and changes in the Catholic population to understand how broader societal changes influence religious demographics.

4. Explore Recent Developments: Explore recent developments within the Catholic Church, such as the canonization of Carlo Acutis and statements by Pope Francis regarding same-sex relationships, to understand their potential impact on demographic trends.

5. Utilize Data Analysis Techniques: Apply data analysis techniques such as exploratory data analysis, statistical analysis, and geospatial analysis to uncover insights and patterns within the demographic data.

6. Provide Insights for Policy and Planning: Provide valuable insights and implications for policymakers, religious leaders, and researchers to inform strategic planning and policy decisions within the Catholic Church.

1.3 Research Question

1. How have the demographics of the Catholic population changed over the 70 years?
2. What regions have seen the most significant growth or decline in Catholic adherents?

1.4 Data

1.4.1 Data Requirements and Choices (Referenced Below)

I chose 3 different datasets to conduct my research on being able to explore the changes to the demographics of the Catholism over time. These data sets help me to retreive information that would be useful to understand the shifts in demographics

Dataset 1: Changes in population numbers and percentages of various religions from 1945 to 2010 at ten-year intervals .

- Provides longitudinal perspective for trend analysis.
- Enables examination of demographic shifts over time.

Dataset 2: Changes in population numbers and percentages of various religions from 1945 to 2010 at ten-year intervals in individual countries.

- Facilitates analysis of regional variations in Catholic demographics.
- Allows comparison of Catholic populations across different countries.

Dataset 3: Population numbers of each religion in various countries from 1945 to 2010 at ten-year intervals in continents.

- Enables comparative analysis between Catholicism and other religions.
- Provides insights into religious demographics on a continental scale.

The selection of these datasets is guided by the need to comprehensively analyze the demographic changes within the global Catholic Church. Dataset 1, tracking changes in population numbers and percentages of various religions from 1945 to 2010, offers a historical lens through which to examine the evolution of Catholic demographics over several decades. By including Catholic population trends with those of other religions, this dataset allows for a nuanced understanding of the Catholic Church's position within the broader religious landscape.

Similarly, Dataset 2, which provides the percentage of the Catholic population in each country along with corresponding population numbers, enables an exploration of regional variations in Catholic demographics. Understanding the geographic distribution of Catholicism and its relative prevalence in different parts of the world is essential for contextualizing demographic trends and identifying regions where the Catholic Church wields significant influence.

Dataset 3 complements this analysis by offering population numbers of each religion in various continents from 1945 to 2010. This dataset facilitates comparative analyses between Catholicism and other religions, shedding light on competitive or collaborative dynamics within the religious sphere. By examining changes in religious demographics on a continetnal scale, researchers can gain insights into the relative growth or decline of the Catholic population, thereby enhancing our understanding of the Catholic Church's broader societal impact.

1.4.2 Limitations and shortfalls of data (Data is referenced below)

Temporal Scope:

- These datasets primarily oversee the period from 1945 to 2010, potentially overlooking recent demographic trends within the Catholic Church.
- This limitation may then lead to an incomplete understanding of the current dynamics and hinder the ability to draw robust conclusions about contemporary Catholic demographics.

Limited Contextual Information:

- The datasets may lack contextual information necessary to fully understand the demographic trends observed within the Catholic Church.
- Factors such as social, cultural, political and economic dynamics influencing religious affiliation may not be adequately captured, making interpretation challenging.

Data Accuracy and Reliability:

- The accuracy and reliability of the data sources used in the datasets may vary, leading to potential inaccuracies or inconsistencies in the demographic information.
- Issues such as data collection methods,reporting biases and discrepancies between sources could impact the validity of the findings and reliability of conclusions.

Geographic Coverage:

- Certain regions or countries with significant Catholic populations may be underrepresented or omitted entirely in the various datasets.
- This limitation could introduce biases in the analysis and affect the generalizability of findings regarding global Catholic demographics

Categorical Representation:

- The datasets may categorize religious affiliation in broad terms, without capturing the diversity of beliefs and practices within the Catholic Church.
- This oversimplification could obsure important nuances in Catholic demographics, limiting the depth of the analysis.

2. Scrapping Information

2.1 Catholic Demographics from 1940 to 2010

In this section, I am analyzing demographic data related to the Roman Catholic population from the years 1940 to 2010. The data, sourced from a CSV file, provides insights into the changes in the Roman Catholic population over these decades.

- By examining trends over time, I aim to understand the dynamics and fluctuations within the Catholic demographic landscape during this period.
- Data will be transferred to a new CSV file where the analysis will be done.

```
import numpy as np
import pandas as pd

# Read the CSV file
demographic_df = pd.read_csv('global.csv')

# Filtering out the specified columns and arranging them accordingly
Catholic_demo_df = demographic_df[['year',
'christianity_romancatholic', 'world_population',
'romancatholic_percent', 'total_percent']]

Catholic_demo_df.columns = ['Year', 'Roman_Catholic_nbr',
'World_Population_nbr', 'Roman_Catholic_Percent', 'Total_Percent']

Catholic_demo_df.to_csv('filtered_global.csv', index=False)

# verifying the data frame
print(Catholic_demo_df)
```

```
    Year  Roman_Catholic_nbr  World_Population_nbr
Roman_Catholic_Percent  \
0   1945           391332035            2250000000
0.2434
1   1950           401935856            2780296616
0.1810
2   1955           474378130            3039433944
0.1871
3   1960           541957872            3345409879
0.1856
4   1965           614115021            3706601448
0.1873
5   1970           671006540            4086387665
0.1836
6   1975           748325898            4453863820
0.1860
7   1980           798834729            4850224998
```

```
0.1873
8   1985           865316946            5277725410
0.1853
9   1990           928527756            5687011326
0.1748
10  1995           962614859            5687011326
0.1698
11  2000           978633933            6081002937
0.1608
12  2005          1013883916            6462181426
0.1575
13  2010          1049709823            6840423256
0.1537

    Total_Percent
0        0.718667
1        0.802432
2        0.834366
3        0.873137
4        0.884908
5        0.894433
6        0.903261
7        0.909855
8        0.915925
9        0.934234
10       0.997009
11       0.999342
12       0.995949
13       0.999792
```

2.2 Reshaping data from various continents and then merging them together with the world data numbers and percentages through the use of two (regional and global) CSV files.

- With regards to this section i will be looking at the regional.csv file which shows me all the data with regards to the continental data in 5 different continents of Africa, Asia, Europe, Mideast, West.Hem.
- From the file i will filter out the numbers for other religion and only focus on the Catholic demographic Population numbers and place them in my cleaned filtered regional data
- After the data has been cleaned, it will be merged together with the cleaned global data to form the main_catholic.csv file for analysis

```
# Reading the regional.csv file
regional_ext_df = pd.read_csv('regional.csv')

# Filter out the specified columns and rearrange the data required
```

```python
regional_df = regional_ext_df[['year', 'christianity_romancatholic',
'region']]
regional_df.columns = ['Year', 'Roman_Catholic_nbr', 'Region']
regional_df_cleaned = regional_df.drop_duplicates(subset=['Year',
'Region'])
regional_df_cleaned = regional_df_cleaned.pivot(index='Year',
columns='Region', values='Roman_Catholic_nbr').reset_index()

# Save this data
regional_df_cleaned.to_csv('filtered_regional.csv', index=False)

# Print the filtered DataFrame to verify
#print(regional_df_cleaned)

# Read data from both CSV files
df_regional = pd.read_csv('filtered_regional.csv')
df_global = pd.read_csv('filtered_global.csv')

# Merging the DataFrames based on the 'Year' column that is present
df_combined = pd.merge(df_global, df_regional, on='Year')
df_combined = df_combined[['Year', 'Africa', 'Asia', 'Europe',
'Mideast', 'West. Hem', 'Roman_Catholic_nbr', 'World_Population_nbr',
'Roman_Catholic_Percent', 'Total_Percent']]
df_combined.columns = ['Year', 'Africa', 'Asia', 'Europe', 'Middle
east', 'Western Hemisphere', 'Roman_Catholic_nbr',
'World_Population_nbr', 'Roman_Catholic_Percent', 'Total_Percent']
# Print the rearranged DataFrame

# Print the combined DataFrame
df_combined.to_csv('main_catholic.csv', index=False)
print(df_combined)
```

```
    Year     Africa       Asia     Europe  Middle east  Western
Hemisphere  \
0   1945     672337    2222908  216041422       635929
171759439
1   1950     736193   25948053  182146023       946089
192159498
2   1955     802667   34806115  223576835      1740329
213452184
3   1960   14074508   41601765  232373678      1627382
252280539
4   1965   28626956   48324442  243240445      2864431
291058747
5   1970   38089508   54128278  248209651      2696173
327882930
6   1975   52803680   62221703  254101668      2331935
376866912
7   1980   63116530   70791416  259260904      2429312
403236567
```

```
8   1985   73386870   79653394  260592119      2614480
449070083
9   1990   86230560   97213932  288130151      4910719
452042394
10  1995  105118187  109255711  258950188      5202572
484088201
11  2000  119734890  111594438  252170341      5009970
490124294
12  2005  143240680  121326959  249539088      4861634
494915555
13  2010  165943467  129766537  244246870      5126980
504625969

    Roman_Catholic_nbr  World_Population_nbr
Roman_Catholic_Percent  \
0            391332035            2250000000                  0.2434

1            401935856            2780296616                  0.1810

2            474378130            3039433944                  0.1871

3            541957872            3345409879                  0.1856

4            614115021            3706601448                  0.1873

5            671006540            4086387665                  0.1836

6            748325898            4453863820                  0.1860

7            798834729            4850224998                  0.1873

8            865316946            5277725410                  0.1853

9            928527756            5687011326                  0.1748

10           962614859            5687011326                  0.1698

11           978633933            6081002937                  0.1608

12          1013883916            6462181426                  0.1575

13          1049709823            6840423256                  0.1537


    Total_Percent
0        0.718667
1        0.802432
2        0.834366
3        0.873137
4        0.884908
5        0.894433
```

```
6        0.903261
7        0.909855
8        0.915925
9        0.934234
10       0.997009
11       0.999342
12       0.995949
13       0.999792
```

2.3 Catholic demographic changes in Countries

In this section i will be filtering the data necessary with repect to the changes to the catholic demographics in the various countires and then we will be comparing this data with the data from the various continents (Africa, Asia, Europe, Mideast, West.Hem).

I will also arrange the countries through the continents that they are in to allow us to be able to figure out the change in the catholic demographic in each continent (Africa, Asia, Europe, Mideast, West.Hem).

```
# Read the CSV file
country_ext_df = pd.read_csv('national.csv')

# Select and rename the necessary columns
country_df = country_ext_df[['year', 'state',
'christianity_romancatholic']]
country_df.columns = ['Year', 'Country', 'Roman_Catholic_nbr']

# Reorder the columns
country_df = country_df[['Year', 'Roman_Catholic_nbr', 'Country']]

# Pivot the DataFrame
country_df = country_df.pivot(index='Year', columns='Country',
values='Roman_Catholic_nbr')
country_df = country_df.dropna(axis=1)

# Calculate the net difference from 1945 to 2010
#difference = country_df.loc[2010] - country_df.loc[1945]
#difference.name = 'Net Difference'

# Add the net difference as a new row to the DataFrame using pd.concat
#country_df = pd.concat([country_df, difference.to_frame().T])

# Save the DataFrame to a CSV file
country_df.to_csv('filtered_national.csv')

# Print the DataFrame and the difference
#print(country_df)
#print("\nNet Difference:\n", difference)
```

```python
# Read the updated CSV file
country_ext_df = pd.read_csv('filtered_national.csv', index_col=0)

# Define continents for each country
continents = {
    'Africa': ['Ethiopia', 'Egypt'],
    'Asia': ['China', 'India', 'Iran', 'Iraq', 'Japan','Thailand',
'Turkey'],
    'Europe': ['Albania', 'Belgium', 'Bulgaria', 'Denmark', 'Finland',
'France', 'Greece', 'Hungary', 'Iceland', 'Ireland', 'Italy',
'Luxembourg', 'Norway', 'Poland', 'Portugal', 'Romania', 'Russia',
'Spain', 'Sweden', 'Switzerland', 'United Kingdom'],
    'Middle East': ['Jordan', 'Lebanon', 'Saudi Arabia'],
    'Western Hemisphere': ['Argentina', 'Australia', 'Bolivia',
'Brazil', 'Canada', 'Chile', 'Colombia', 'Costa Rica', 'Cuba',
'Dominican Republic', 'Ecuador', 'El Salvador', 'Guatemala', 'Haiti',
'Honduras', 'Mexico', 'New Zealand', 'Nicaragua', 'Panama',
'Paraguay', 'Peru', 'United States of America', 'Uruguay',
'Venezuela']
}

# Create a dictionary to store the columns for each continent
continent_columns = {continent: [] for continent in continents.keys()}

# Iterate through the continents and assign columns to each
for continent, countries in continents.items():
    for country in countries:
        if country in country_ext_df.columns:
            continent_columns[continent].append(country)

# Reorder the columns of the DataFrame
ordered_columns = []
for continent in ['Africa', 'Asia', 'Europe', 'Middle East', 'Western
Hemisphere']:
    ordered_columns.extend(continent_columns[continent])

# Select only the necessary columns
filtered_country_df = country_ext_df[ordered_columns]

# Reset the index to include the 'Year' column in the output
filtered_country_df.reset_index(inplace=True)

# Rename the index column to 'Year'
filtered_country_df.rename(columns={'index': 'Year'}, inplace=True)

# Save the DataFrame to a CSV file, ensuring the index (Year) is
included
filtered_country_df.to_csv('filtered_national.csv', index=False)
```

```python
# Print the updated DataFrame
print(filtered_country_df)
```

```
    Year  Ethiopia     Egypt       China     Iran      Iraq      Japan  \
0   1945   46436.0  163560.0    129525.0      0.0   45819.0   111856.0

1   1950   47758.0  204610.0    623359.0      0.0  103960.0   100000.0

2   1955   49117.0  177305.0   3000000.0      0.0  107998.0   164961.0

3   1960   50475.0  150000.0   3000000.0      0.0  101639.0   229921.0

4   1965   51834.0  293890.0   3000000.0      0.0  232406.0   294882.0

5   1970   53192.0  120500.0    407861.0      0.0  266600.0   359842.0

6   1975  131596.0   98000.0   2354167.0      0.0  326007.0   392421.0

7   1980  210000.0   42500.0   4300474.0  29300.0  385415.0   425000.0

8   1985  245000.0   55159.0   6246780.0      0.0  125000.0   432000.0

9   1990  280000.0  200000.0   8193086.0   5600.0  272000.0   439000.0

10  1995  351469.0  208603.0  10139393.0   2800.0  231150.0   477363.0

11  2000  434043.0  120000.0  10220464.0      0.0  235000.0  1276702.0

12  2005  450000.0  100000.0  10429759.0      0.0  279630.0  1287372.0

13  2010  578379.0   98000.0  10761394.0      0.0  265000.0  1072810.0


    Thailand   Turkey   Albania  ...   Honduras      Mexico  New
Zealand  \
0    50032.0  21950.0  143053.0  ...  1174269.0  20673900.0
231153.0
1    90517.0      0.0  135000.0  ...  1373000.0  25329498.0
280000.0
2   163761.0  14891.0  127400.0  ...  1583700.0  25329000.0
341760.0
3   237006.0   6193.0  119200.0  ...  1854000.0  33871000.0
393000.0
4   310250.0      0.0  112200.0  ...  1952293.0  36228500.0
420480.0
5   151663.0  19133.0  163400.0  ...  2412601.0  47028524.0
424768.0
6   187331.0  28066.0  214400.0  ...  2889835.0  61914000.0
493280.0
7   223000.0  37000.0  265000.0  ...  3367070.0  64138880.0
```

```
526220.0
8   231500.0  30000.0  325270.0  ...  4156000.0  72253000.0
474000.0
9   240000.0  29200.0  423280.0  ...  4750300.0  77770000.0
470000.0
10  252929.0  30146.0  505260.0  ...  5245150.0  84970000.0
531300.0
11  261436.0  31000.0  249040.0  ...  5740000.0  92770000.0
582951.0
12  153713.0  33548.0  402519.0  ...  5740000.0  93248365.0
637907.0
13  172023.0  35747.0  419535.0  ...  4561162.0  92924489.0
673936.0

    Nicaragua     Panama   Paraguay        Peru  United States of
America  \
0    873843.0   657800.0  1100000.0   6779700.0
38716742.0
1   1012000.0   739000.0  1234000.0   7668000.0
42635882.0
2   1172000.0   889240.0  1507711.0   8170818.0
46402368.0
3   1332000.0  1058440.0  1711115.0   9561000.0
50587880.0
4   1450081.0  1187430.0  1914520.0  10893650.0
64761783.0
5   1780995.0  1316421.0  2185000.0  12769151.0
69119143.0
6   2090547.0  1502390.0  2526050.0  14402950.0
77181399.0
7   2400100.0  1741766.0  2958300.0  16357861.0
75149580.0
8   2974000.0  1941000.0  3368000.0  18360700.0
82463100.0
9   3275300.0  2004500.0  3835000.0  20091050.0
65530789.0
10  3787650.0  2367900.0  4345200.0  21805388.0
70308695.0
11  4300000.0  2330000.0  4924416.0  20529600.0
75086601.0
12  3556125.0  2550120.0  4771111.0  23639350.0
76742245.0
13  3798734.0  3361450.0  5680657.0  23904351.0
78397889.0

      Uruguay   Venezuela
0   1368000.0   3710000.0
1   1537000.0   4466000.0
2   1756900.0   5648283.0
3   1976800.0   6835000.0
```

```
4   1954310.0   8111460.0
5   2115337.0   9567080.0
6   2067646.0  11473000.0
7   2019955.0  13378920.0
8   1985280.0  16418773.0
9   2119600.0  18331880.0
10  2202760.0  20012640.0
11  2311080.0  18852600.0
12  1557126.0  20959900.0
13  1580995.0  23067200.0

[14 rows x 57 columns]

/var/folders/_g/rrqlgdwn22xgr2gty3pkq_0m0000gn/T/
ipykernel_60377/3191907937.py:34: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  filtered_country_df.rename(columns={'index': 'Year'}, inplace=True)
```

2.4 Crafting Our Main Csv File

- Main_Catholic.csv which consists of the world population numbers and the continental population data for catholics will be combined together with the filtered_national.csv that consists of the Catholic demogrpahic population numbers in each country
- This will produce the cleaned main_catholic.csv file that we will use to analyise our data and reserach about.

```
# Loading all the datasets we need
catholic_data = pd.read_csv('main_catholic.csv')
national_data = pd.read_csv('filtered_national.csv')

# Merging the datasets based of the 'Year' column
combined_data = pd.merge(catholic_data, national_data, on='Year')
#df_combined = df_combined[['Year', 'Africa', 'Asia', 'Europe',
'Middle east', 'Western Hemisphere', 'Roman_Catholic_nbr',
'World_Population_nbr', 'Roman_Catholic_Percent', 'Total_Percent',
'Ethiopia', 'Egypt', 'China', 'Iran', 'Iraq', 'Japan', 'Jordan',
'Lebanon', 'Saudi Arabia', 'Thailand', 'Turkey', 'Albania', 'Belgium',
'Bulgaria', 'Denmark', 'Finland', 'France', 'Greece', 'Hungary',
'Iceland', 'Ireland', 'Italy', 'Luxembourg', 'Norway', 'Poland',
'Portugal', 'Romania', 'Russia', 'Spain', 'Sweden', 'Switzerland',
'United Kingdom', 'Iraq.1', 'Iran.1', 'Jordan.1', 'Lebanon.1', 'Saudi
Arabia.1', 'Turkey.1', 'Argentina', 'Australia', 'Bolivia', 'Brazil',
'Canada', 'Chile', 'Colombia', 'Costa Rica', 'Cuba', 'Dominican
```

```
Republic', 'Ecuador', 'El Salvador', 'Guatemala', 'Haiti', 'Honduras',
'Mexico', 'New Zealand', 'Nicaragua', 'Panama', 'Paraguay', 'Peru',
'United States of America', 'Uruguay', 'Venezuela']]
#df_combined.columns = ['Year','Ethiopia', 'Egypt', 'Africa','China',
'Iran', 'Iraq', 'Japan', 'Jordan', 'Lebanon', 'Thailand', 'Turkey',
'Asia','Albania', 'Belgium', 'Bulgaria', 'Denmark', 'Finland',
'France', 'Greece', 'Hungary','Iceland', 'Ireland', 'Italy',
'Luxembourg', 'Norway', 'Poland', 'Portugal', 'Romania','Russia',
'Spain', 'Sweden', 'Switzerland', 'United Kingdom', 'Europe','Saudi
Arabia', 'Middle east','Argentina', 'Australia', 'Bolivia', 'Brazil',
'Canada', 'Chile', 'Colombia', 'Costa Rica','Cuba', 'Dominican
Republic', 'Ecuador', 'El Salvador', 'Guatemala', 'Haiti',
'Honduras','Mexico', 'New Zealand', 'Nicaragua', 'Panama', 'Paraguay',
'Peru', 'United States of America','Uruguay', 'Venezuela', 'Western
Hemisphere','Roman_Catholic_nbr', 'World_Population_nbr',
'Roman_Catholic_Percent', 'Total_Percent']

# Saving the combined dataset to the main_catholic.csv file
combined_data.to_csv('main_catholic.csv', index=False)

# Loading the main_catholic.csv file
df = pd.read_csv('main_catholic.csv')
# Rearranging all the columns to get my desired column input based on
countries and continents
columns_to_move = ['Roman_Catholic_nbr', 'World_Population_nbr',
'Roman_Catholic_Percent', 'Total_Percent']

remaining_columns = [col for col in df.columns if col not in
columns_to_move]

df_reordered = df[remaining_columns + columns_to_move]

# Save the reordered DataFrame to a new CSV file
df_reordered.to_csv('main_catholic.csv', index=False)

print(df_reordered)
```

```
    Year     Africa       Asia     Europe  Middle east  Western
Hemisphere  \
0   1945     672337    2222908  216041422       635929
171759439
1   1950     736193   25948053  182146023       946089
192159498
2   1955     802667   34806115  223576835      1740329
213452184
3   1960   14074508   41601765  232373678      1627382
252280539
4   1965   28626956   48324442  243240445      2864431
291058747
5   1970   38089508   54128278  248209651      2696173
```

```
327882930
6   1975   52803680   62221703  254101668      2331935
376866912
7   1980   63116530   70791416  259260904      2429312
403236567
8   1985   73386870   79653394  260592119      2614480
449070083
9   1990   86230560   97213932  288130151      4910719
452042394
10  1995  105118187  109255711  258950188      5202572
484088201
11  2000  119734890  111594438  252170341      5009970
490124294
12  2005  143240680  121326959  249539088      4861634
494915555
13  2010  165943467  129766537  244246870      5126980
504625969

    Ethiopia     Egypt        China      Iran  ...      Panama   Paraguay
\
0     46436.0  163560.0    129525.0       0.0  ...   657800.0  1100000.0

1     47758.0  204610.0    623359.0       0.0  ...   739000.0  1234000.0

2     49117.0  177305.0   3000000.0       0.0  ...   889240.0  1507711.0

3     50475.0  150000.0   3000000.0       0.0  ...  1058440.0  1711115.0

4     51834.0  293890.0   3000000.0       0.0  ...  1187430.0  1914520.0

5     53192.0  120500.0    407861.0       0.0  ...  1316421.0  2185000.0

6    131596.0   98000.0   2354167.0       0.0  ...  1502390.0  2526050.0

7    210000.0   42500.0   4300474.0   29300.0  ...  1741766.0  2958300.0

8    245000.0   55159.0   6246780.0       0.0  ...  1941000.0  3368000.0

9    280000.0  200000.0   8193086.0    5600.0  ...  2004500.0  3835000.0

10   351469.0  208603.0  10139393.0    2800.0  ...  2367900.0  4345200.0

11   434043.0  120000.0  10220464.0       0.0  ...  2330000.0  4924416.0

12   450000.0  100000.0  10429759.0       0.0  ...  2550120.0  4771111.0

13   578379.0   98000.0  10761394.0       0.0  ...  3361450.0  5680657.0


          Peru  United States of America    Uruguay   Venezuela  \
0    6779700.0                38716742.0  1368000.0   3710000.0
```

```
1    7668000.0                42635882.0  1537000.0   4466000.0
2    8170818.0                46402368.0  1756900.0   5648283.0
3    9561000.0                50587880.0  1976800.0   6835000.0
4   10893650.0                64761783.0  1954310.0   8111460.0
5   12769151.0                69119143.0  2115337.0   9567080.0
6   14402950.0                77181399.0  2067646.0  11473000.0
7   16357861.0                75149580.0  2019955.0  13378920.0
8   18360700.0                82463100.0  1985280.0  16418773.0
9   20091050.0                65530789.0  2119600.0  18331880.0
10  21805388.0                70308695.0  2202760.0  20012640.0
11  20529600.0                75086601.0  2311080.0  18852600.0
12  23639350.0                76742245.0  1557126.0  20959900.0
13  23904351.0                78397889.0  1580995.0  23067200.0

    Roman_Catholic_nbr  World_Population_nbr
Roman_Catholic_Percent  \
0            391332035            2250000000                  0.2434

1            401935856            2780296616                  0.1810

2            474378130            3039433944                  0.1871

3            541957872            3345409879                  0.1856

4            614115021            3706601448                  0.1873

5            671006540            4086387665                  0.1836

6            748325898            4453863820                  0.1860

7            798834729            4850224998                  0.1873

8            865316946            5277725410                  0.1853

9            928527756            5687011326                  0.1748

10           962614859            5687011326                  0.1698

11           978633933            6081002937                  0.1608

12          1013883916            6462181426                  0.1575

13          1049709823            6840423256                  0.1537


    Total_Percent
0        0.718667
1        0.802432
2        0.834366
3        0.873137
4        0.884908
```

```
5        0.894433
6        0.903261
7        0.909855
8        0.915925
9        0.934234
10       0.997009
11       0.999342
12       0.995949
13       0.999792

[14 rows x 66 columns]
```

3. Detailed Analysis of Data

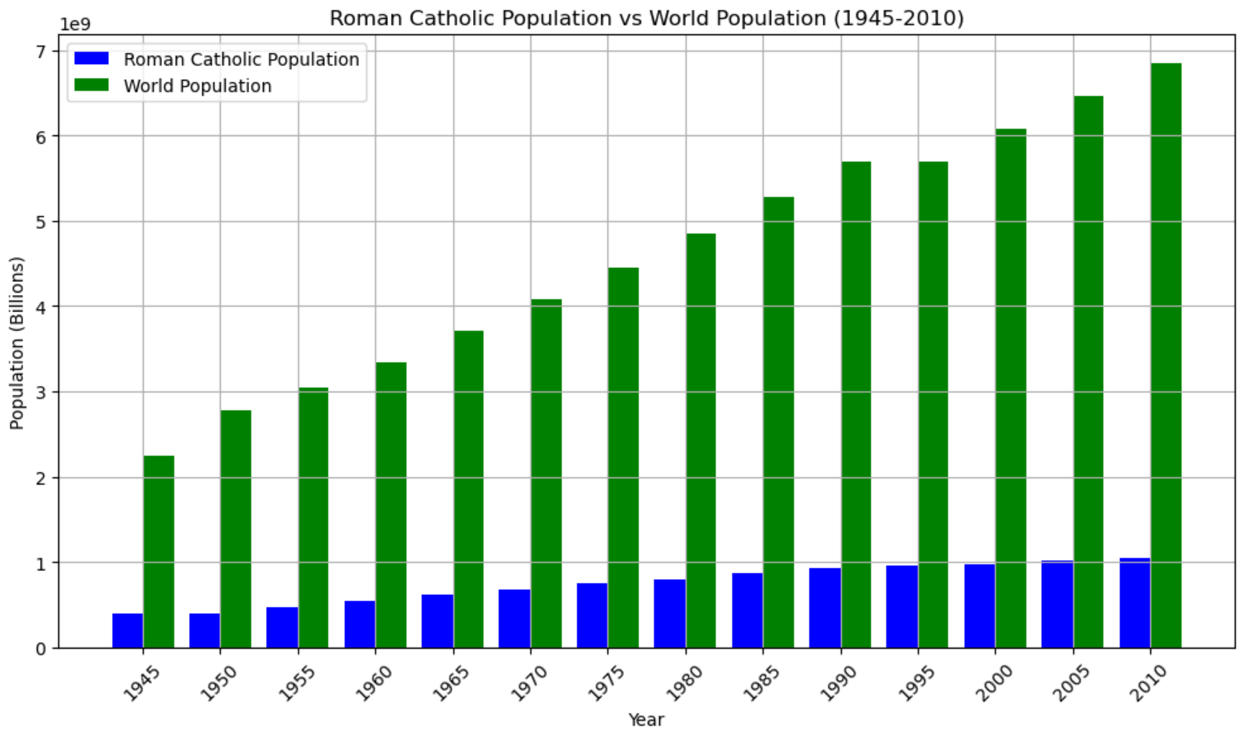
3.1 Looking at the bigger picture - analysis with the world population

3.1.1 World vs Catholic Population

From the data that we have collected we can see there is a change in the world population and the Roman Catholic population. Through the use of the bar graph we can then identify and seek a clearer clarification in the sizeable difference in increase of the population numbers.

```python
import matplotlib.pyplot as plt
# Read data from CSV file
df = pd.read_csv('filtered_global.csv')

# Plotting
plt.figure(figsize=(10, 6))

# Bar width
bar_width = 0.4

# Positions of the bars on the x-axis
r1 = range(len(df['Year']))
r2 = [x + bar_width for x in r1]

# Plot Roman Catholic population
plt.bar(r1, df['Roman_Catholic_nbr'], width=bar_width, color='blue',
label='Roman Catholic Population')

# Plot World population
plt.bar(r2, df['World_Population_nbr'], width=bar_width,
color='green', label='World Population')

plt.title('Roman Catholic Population vs World Population (1945-2010)')
```

```
plt.xlabel('Year')
plt.ylabel('Population (Billions)')
plt.grid(True)
plt.xticks([r + bar_width / 2 for r in range(len(df['Year']))],
df['Year'], rotation=45)
plt.legend()
plt.tight_layout()
plt.show()
```

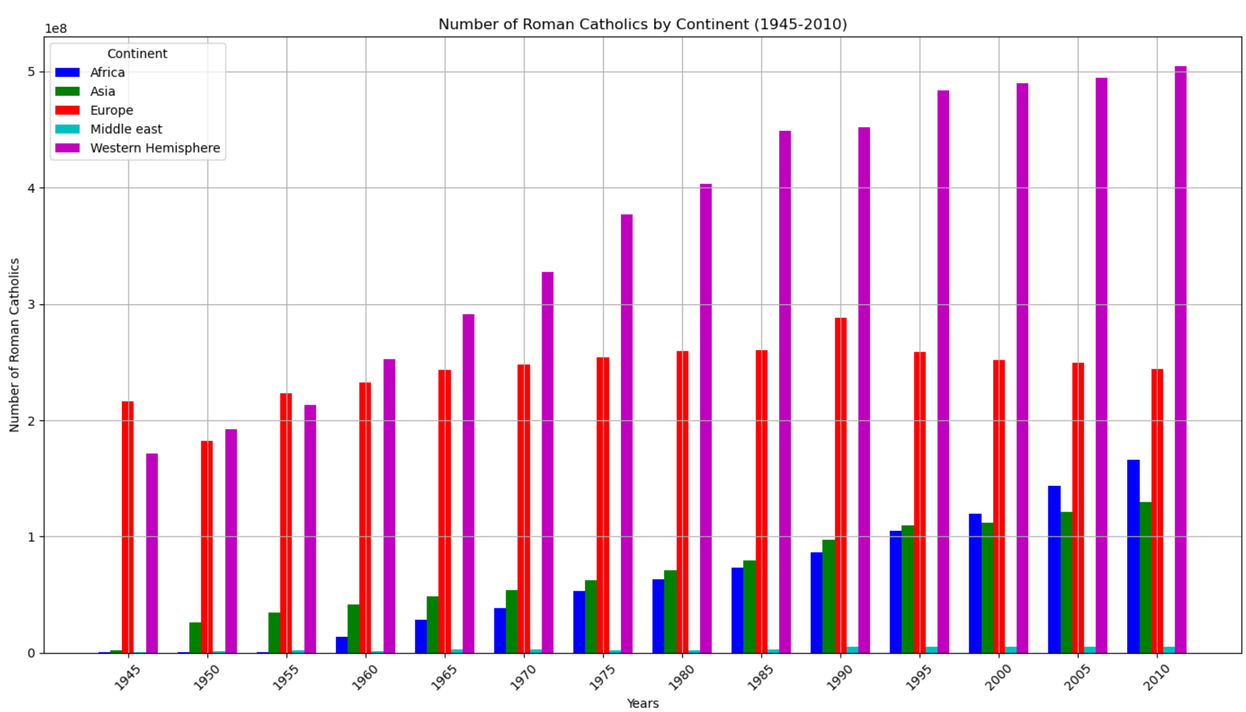


3.1.2 Growth in Continents

With the use of the bar graph again we will be diving into the various continents and seeing how the population numbers have changed over time from the 1940-2010. The bar graph will display the Catholic Population numbers of each continent (Africa, Asia, Europe, Mideast, West.Hem).

```
# Read the combined CSV file
df_combined = pd.read_csv('main_catholic.csv')

# Plotting the bar graph for each continent
plt.figure(figsize=(14, 8))

# Define a color palette
colors = ['b', 'g', 'r', 'c', 'm']

# Define the width of each bar
bar_width = 0.15
```

```
# Define the positions of the bars on the x-axis
years = df_combined['Year']
bar_positions = [np.arange(len(years))]
for i in range(1, len(df_combined.columns) - 1):
    bar_positions.append([x + bar_width for x in bar_positions[i -
1]])

# Plot each continent's data
for i, continent in enumerate(['Africa', 'Asia', 'Europe', 'Middle
east', 'Western Hemisphere']):
    plt.bar(bar_positions[i], df_combined[continent], width=bar_width,
color=colors[i], label=continent)

# Add labels, title, and legend
plt.xlabel('Years')
plt.ylabel('Number of Roman Catholics')
plt.title('Number of Roman Catholics by Continent (1945-2010)')
plt.xticks([r + bar_width*2 for r in range(len(years))], years,
rotation=45)
plt.legend(title='Continent')
plt.grid(True)
plt.tight_layout()

# Show the plot
plt.show()
```



3.1.3 Findings with relation to Bar Graphs

Through the use of the Bar graphs i have been able to understand trend in the shifts in the catholic demographics between the years 1940 and 2010.

Firstly in the graph between the 1940 and 2010 we can see the exponential shift in the growth of the world population. It is known that through the medical, scientific and economic advances we have seen the significant increase in demographic numbers. (https://www.iberdrola.com/sustainability/world-population-evolution#:~:text=In%20the%2019th%20century%2C%20the,led%20to%20this%20exponential%20growth.)

However we can also see that even though the increase in the world population numbers is exponential, the catholic demographic numbers did not increase exponentially and only experienced a gradual increase in numbers and this could be due to a plethora of reasons.

As we dive a little bit deeper in trying to find which parts of the world that might have resulted in the slow increase in number we can see that different parts grew at a very different rate.

- The western Hemisphere saw an exponential growth in their numbers while Asia and Africa also saw some growth in their numbers
- On the other hand, regions of europe and middle east did not see much growth from 1940 and 2010 and pretty much stayed constant. However the difference was that europe saw fluctuations in the Catholic demographics indicating various socio-economic, political, religious factors could have played a significant role.

3.2 Time Series Decomposition

With the completion of scraping for the necessary data from the various csv files i will now undergo analysis of the demographic data collected. This analysis will help us to understand the evolution of the catholic church and its global presense over time.

Firstly, we will conduct a time series decomposition to find out more about the trends across different continents and how the demographic of the catholic church changes accordingly. By decomposing the data into various components-trend,seasonality and residuals we aim to uncover the various underlying patterns and insights that contribute to the changes in Catholic populations worldwide!

The data that we are comparing here specifically spans the historical catholic demographic figures that span from 1940 - 2010 across continents of Africa,Asia,Europe,Middle East and Western Western Hemisphere. This data also includes the number of catholics as a percentave of the total population for each continent over specific years.

3.2.1 Decomposition of Continental Data

```
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose

# Load the data
df = pd.read_csv('main_catholic.csv')
```

```python
# Convert 'Year' column to datetime format and set as index
df['Year'] = pd.to_datetime(df['Year'], format='%Y')
df.set_index('Year', inplace=True)

# Select relevant columns for analysis (continents)
continents = ['Africa', 'Asia', 'Europe', 'Middle east', 'Western
Hemisphere']
df_continents = df[continents]
```

The components that we are focusing are Actual Series, Trend, Seasonality and Residuals :

Actual : This displays the actual data points that reflect the Catholic Demographic percentages over time

Trend : Focuses on the long-term movement or direction of how the data moves by filtering out the various short-term fluctuations

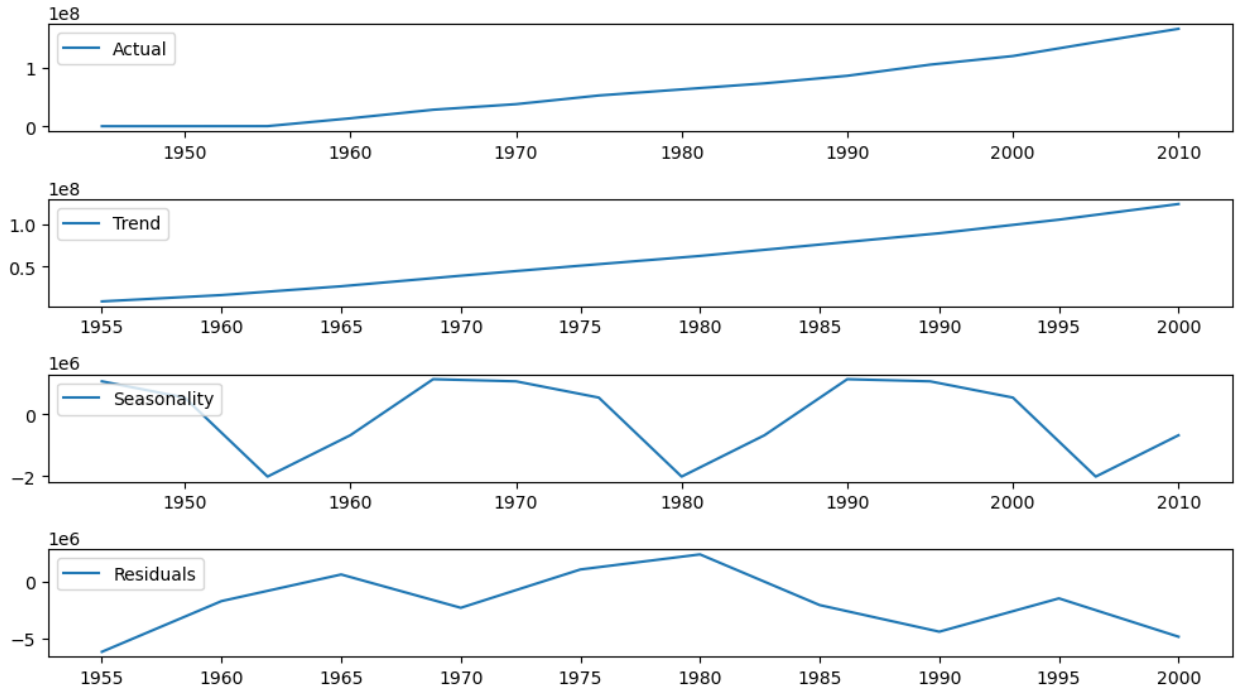
Seasonality : Helps to highlight the periodic fluctuations (Might not be accurate due to the type of data set)

Residuals : Represents the random variability that will remain after we remove the seasonality and trend components from the actual

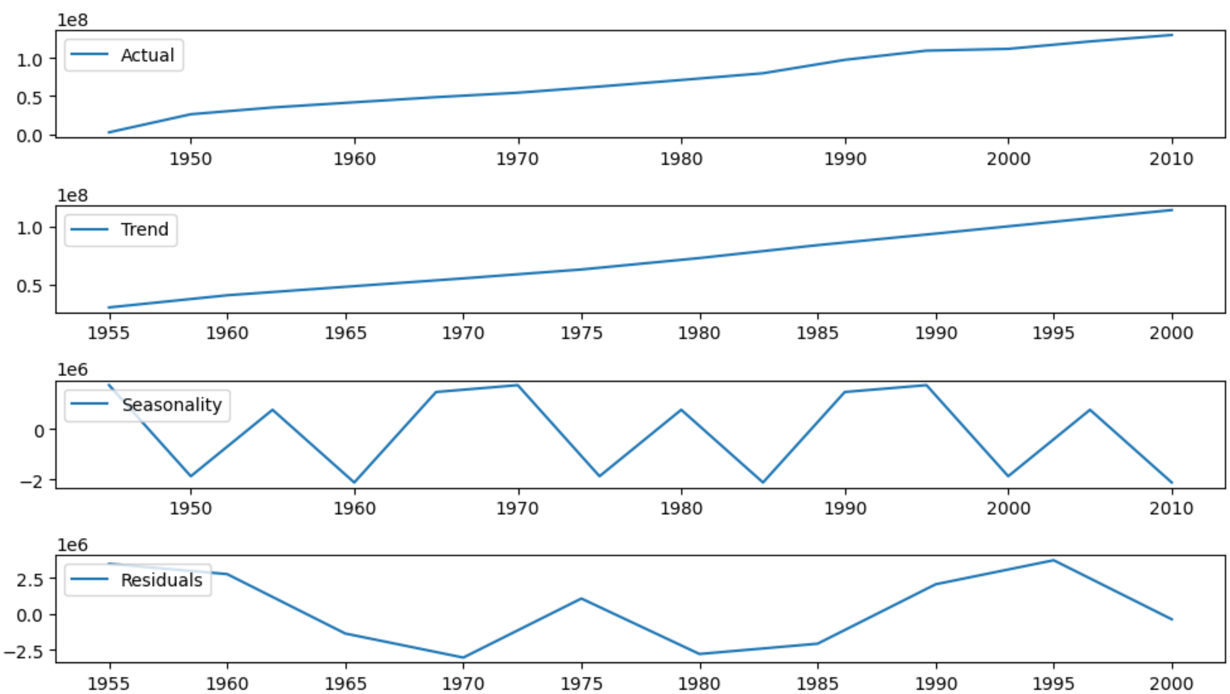
Using the continental data from the main_catholic.csv file we will be performing a decomposition for each continent based on the components listed above

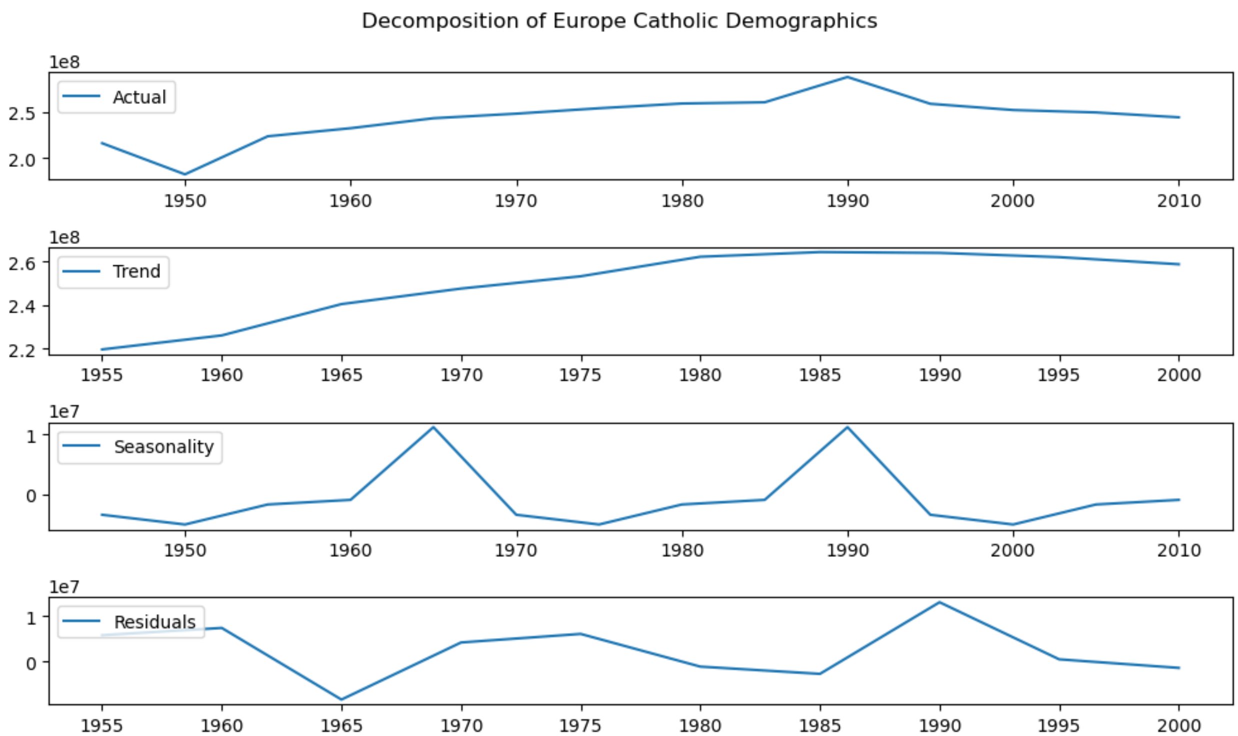
```python
# Perform decomposition for each continent
for continent in continents:
    decomposition = seasonal_decompose(df_continents[continent],
model='additive', period=5)  # adjust period as needed

    # Plotting decomposition
    plt.figure(figsize=(10, 6))
    plt.subplot(411)
    plt.plot(df_continents[continent], label='Actual')
    plt.legend(loc='upper left')
    plt.subplot(412)
    plt.plot(decomposition.trend, label='Trend')
    plt.legend(loc='upper left')
    plt.subplot(413)
    plt.plot(decomposition.seasonal, label='Seasonality')
    plt.legend(loc='upper left')
    plt.subplot(414)
    plt.plot(decomposition.resid, label='Residuals')
    plt.legend(loc='upper left')
    plt.suptitle(f'Decomposition of {continent} Catholic
Demographics')
    plt.tight_layout()
    plt.show()
```

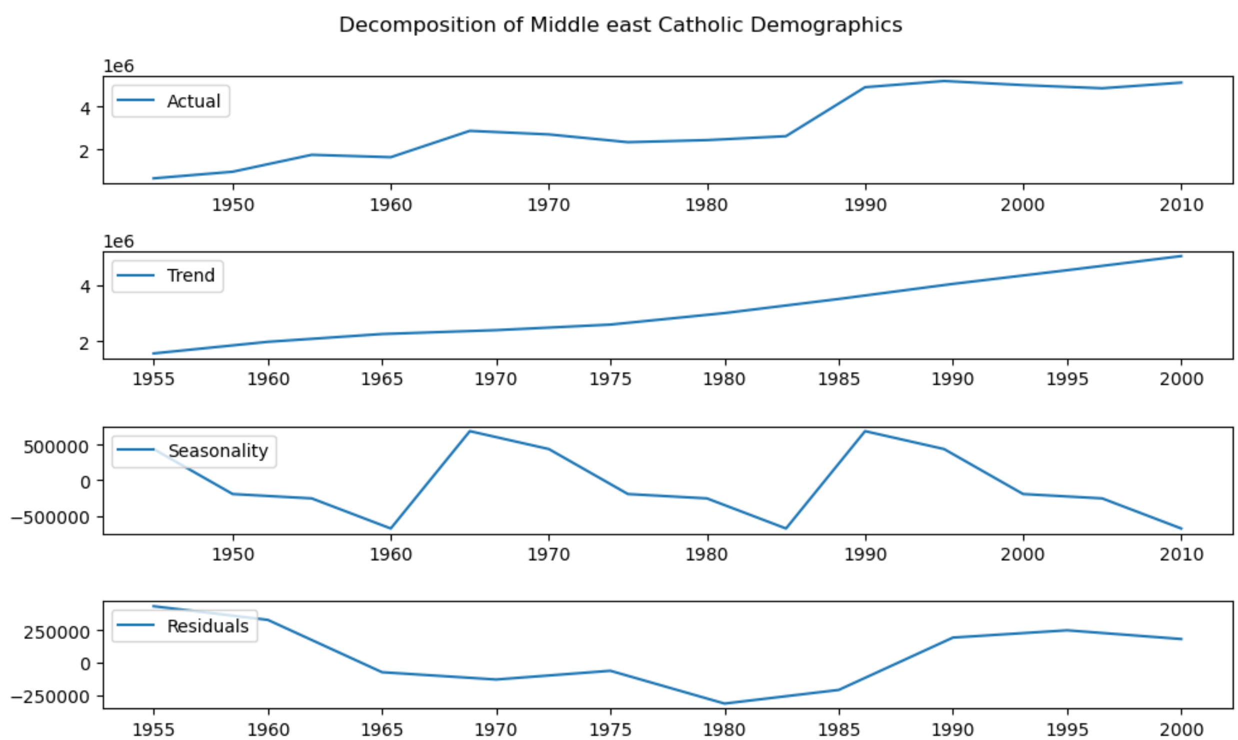
Decomposition of Africa Catholic Demographics

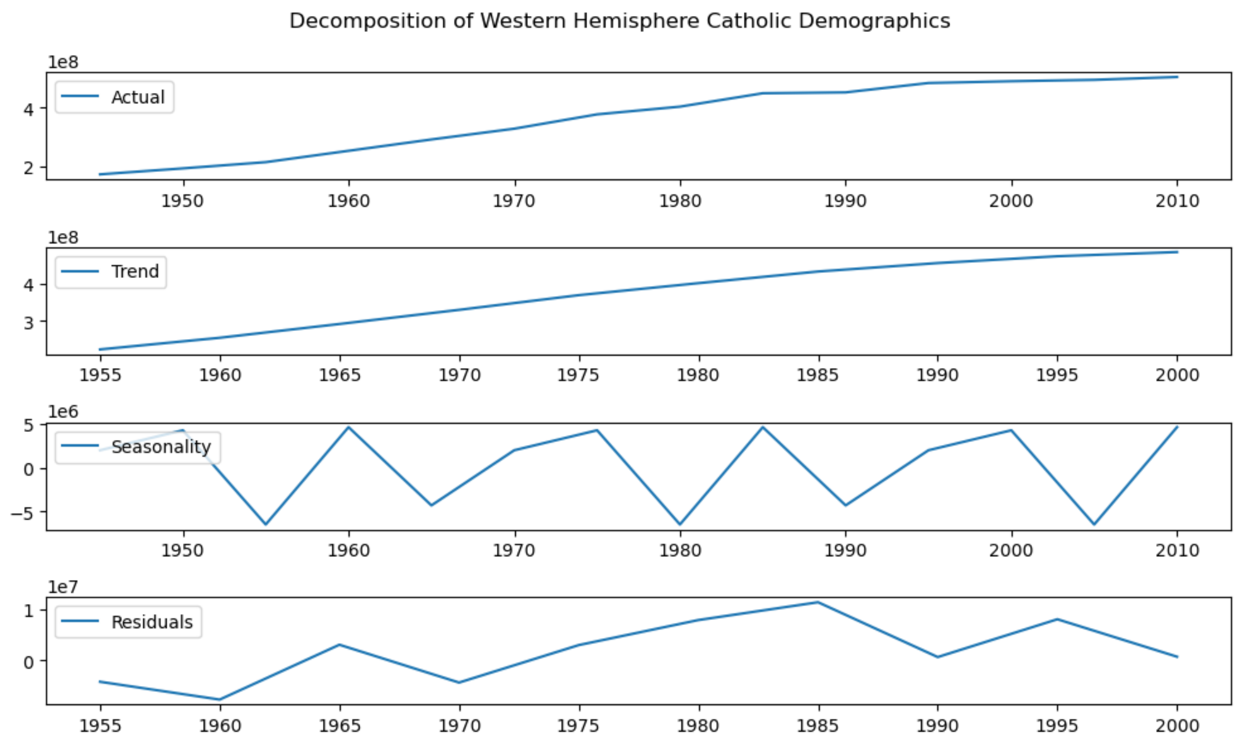


Decomposition of Asia Catholic Demographics

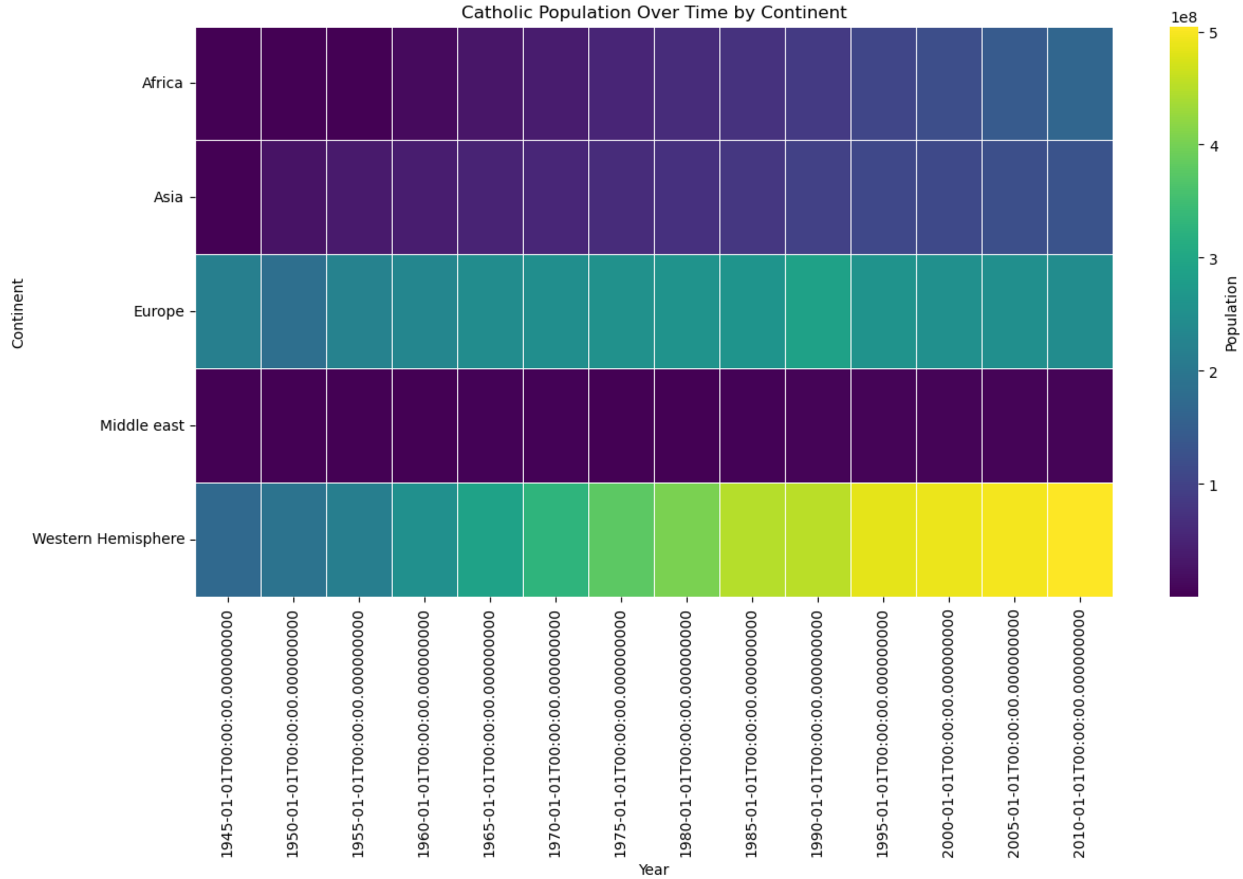



Decomposition of Europe Catholic Demographics
1e8
Actual
2.5
2.0
1950
1960
1970
1980
1990
2000
2010
1e8
Trend
2.6
2.4
2.2
1955
1960
1965
1970
1975
1980
1985
1990
1995
2000
1e7
Seasonality
1
0
1e7
Residuals


Decomposition of Middle east Catholic Demographics
1e6
Actual
4
2
1e6
Trend
Seasonality
500000
0
−500000
Residuals
250000
0
−250000




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

# Load the data
df = pd.read_csv('main_catholic.csv')

# Convert 'Year' column to datetime format and set as index
df['Year'] = pd.to_datetime(df['Year'], format='%Y')
df.set_index('Year', inplace=True)

# Select relevant columns for analysis (continents)
continents = ['Africa', 'Asia', 'Europe', 'Middle east', 'Western
Hemisphere']
df_continents = df[continents]

# Plot the data as heatmap
plt.figure(figsize=(14, 7))
sns.heatmap(df_continents.T, cmap='viridis', cbar_kws={'label':
'Population'}, linewidths=.5)

plt.title('Catholic Population Over Time by Continent')
plt.xlabel('Year')
plt.ylabel('Continent')
plt.show()
```



3.2.2 Insights from Decomposition of Continental Data

Continental/Regional Data

AFRICA : Africa as a region showed steady increase in Catholic demographics, with pronounced seasonality.

ASIA : Demonstrated a small and steady growth rate

EUROPE : Displayed either a stable or decling trends in the Catholic Demographics, with subtle seasonal variations

MIDDLE EAST : Displayed only a slight increase and was consistently the lowest in terms of catholic population numbers

WESTERN HEMISPHERE: Showed a very steady growth in Catholic demographics and also had a very high number of catholic population indicating high amount of regional events and cultural practices.

4. Decomposition of Country Data

After focusing on the decomposition of the continental data, i will be looking into the decomposition of the country data on the Catholic population to get a better perspective into the global changes

```
#Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
import random

#Load the data
df = pd.read_csv('main_catholic.csv')
# Transpose the dataframe for easier manipulation
df_t = df.set_index('Year').transpose()
# Extract country names
countries = df_t.index.to_list()
ignore_list = ['Africa','Asia','Europe','Middle east','Western
Hemisphere','Roman_Catholic_nbr', 'World_Population_nbr',
'Roman_Catholic_Percent', 'Total_Percent']
countries = [country for country in countries if country not in
ignore_list]
```

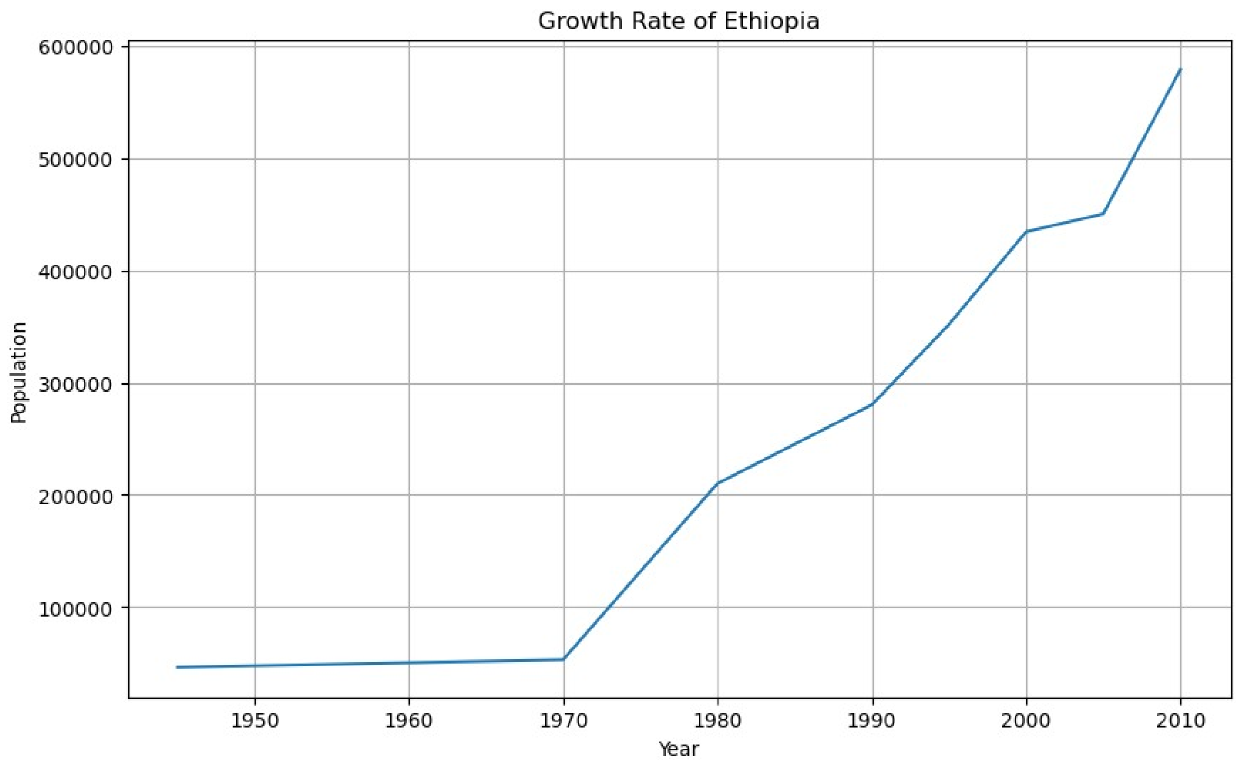
4.1 Growth rate in Countries

In this section we will be exploring the growth rate of the catholic population in the various countries and also some of the factors that might have resulted in them having such growth rates. The first function is used to plot individual growth rates such that

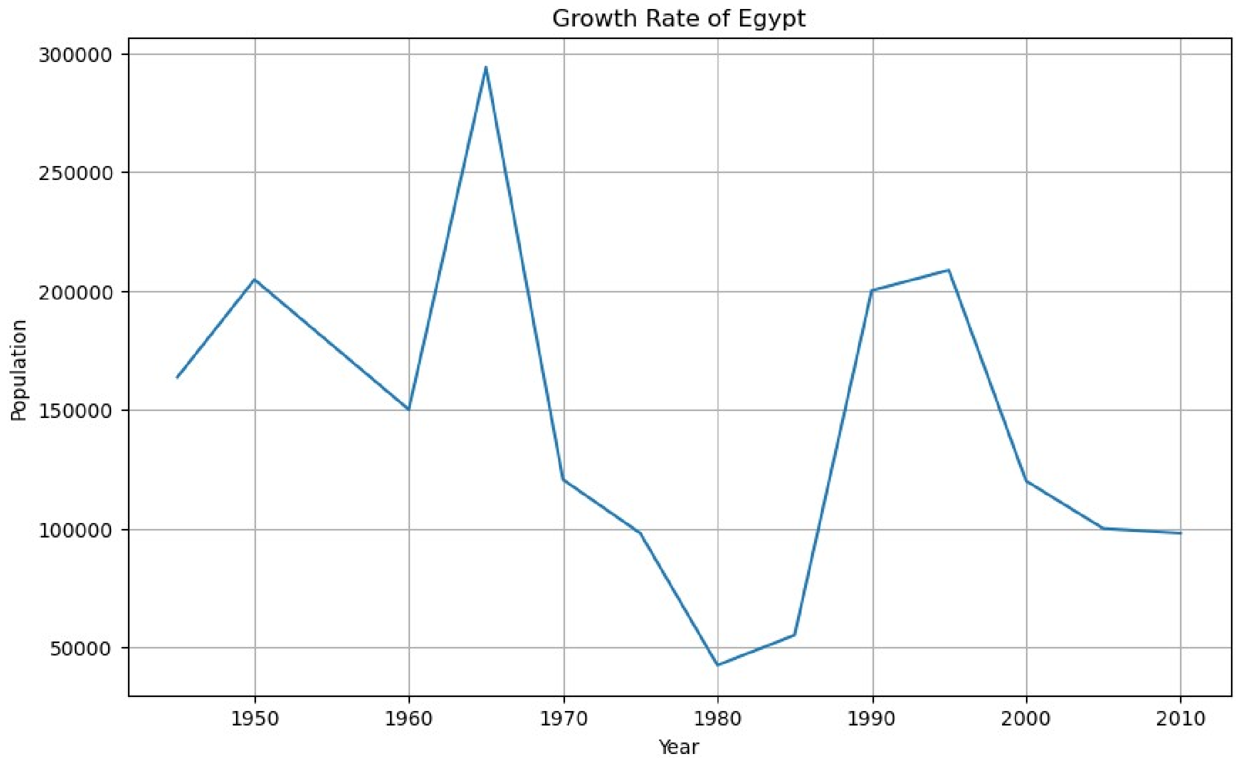
```
# Function to plot time series for each country
def growth_rate(country):
    plt.figure(figsize=(10, 6))
    sns.lineplot(x=df['Year'], y=df[country])
    plt.title(f'Growth Rate of {country}')
    plt.xlabel('Year')
    plt.ylabel('Population')
    plt.grid(True)
    plt.show()

for country in countries:
    growth_rate(country)
```

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
```

```
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

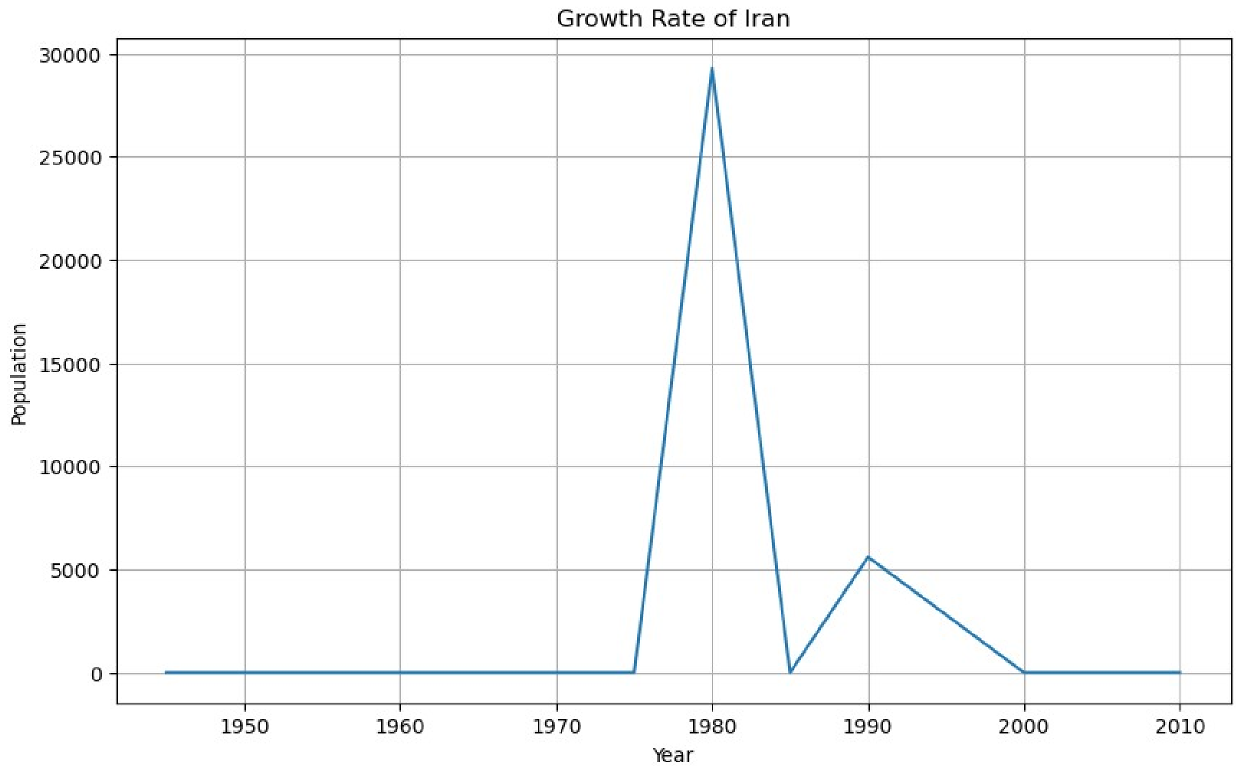



```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

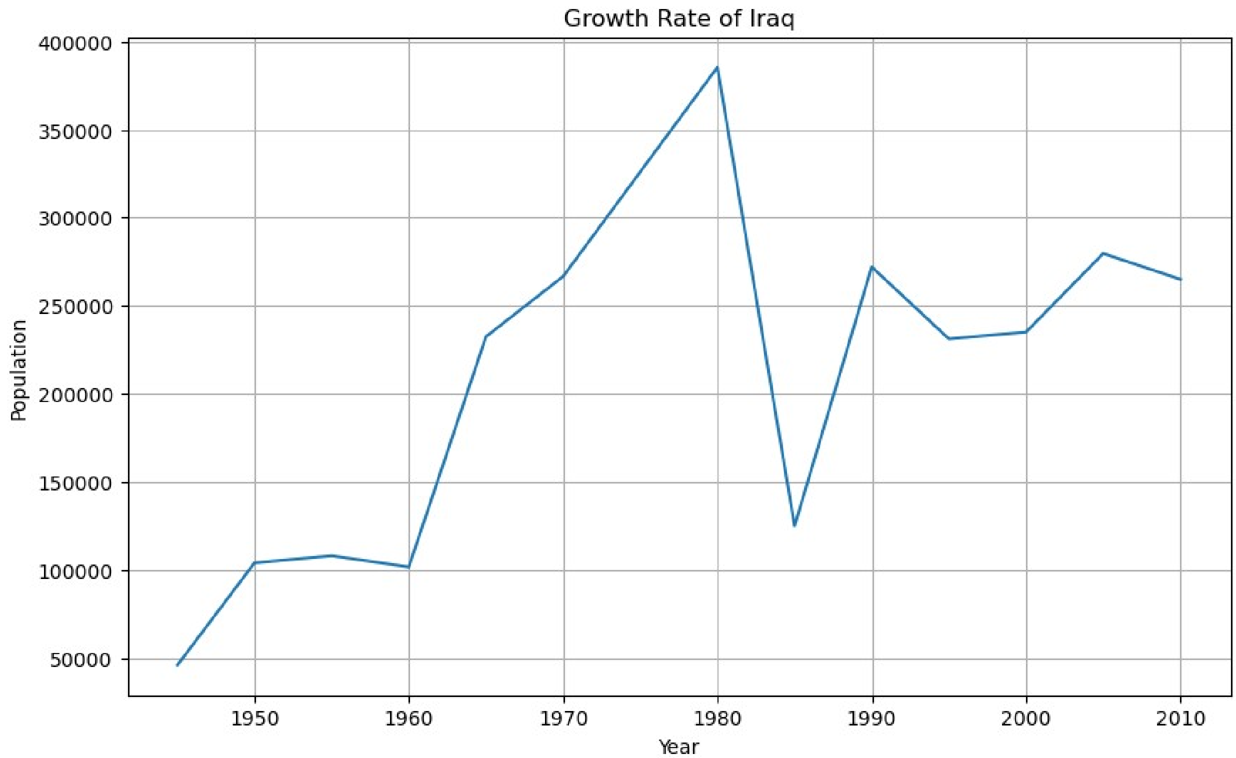



```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

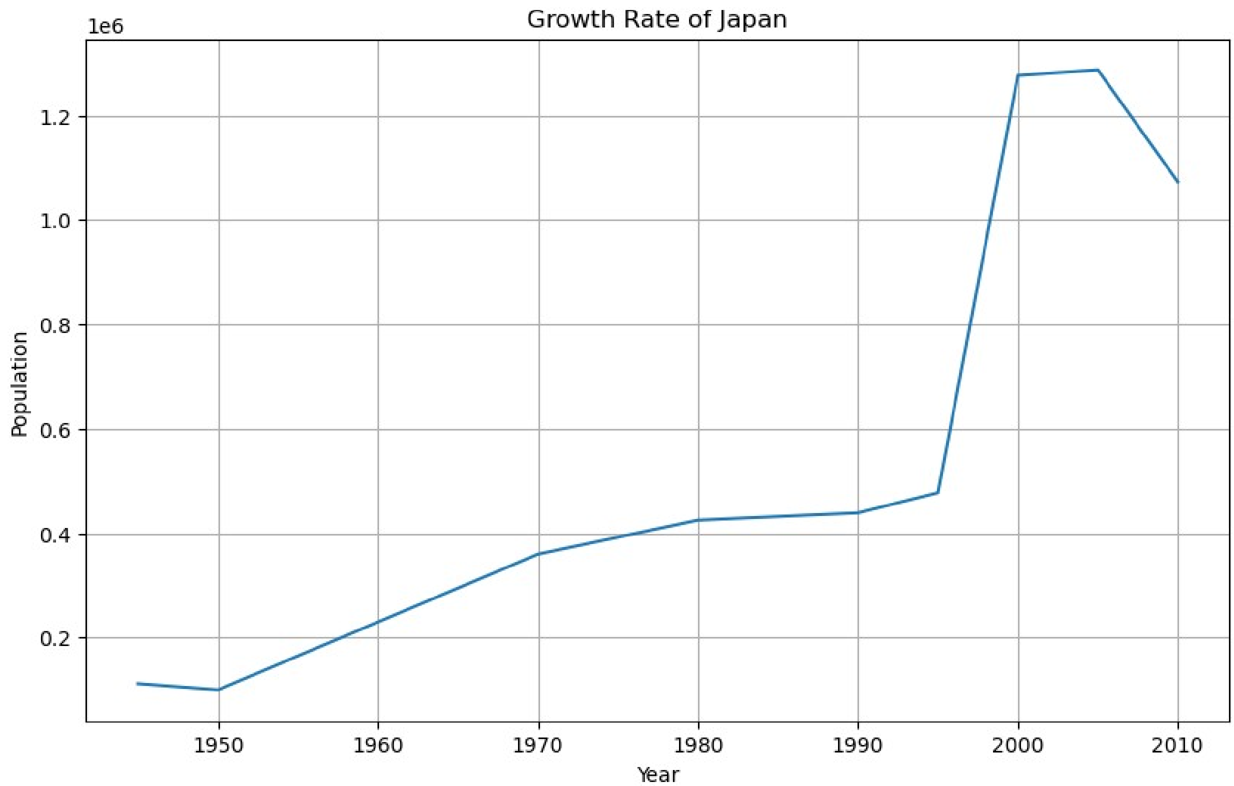

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```




```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

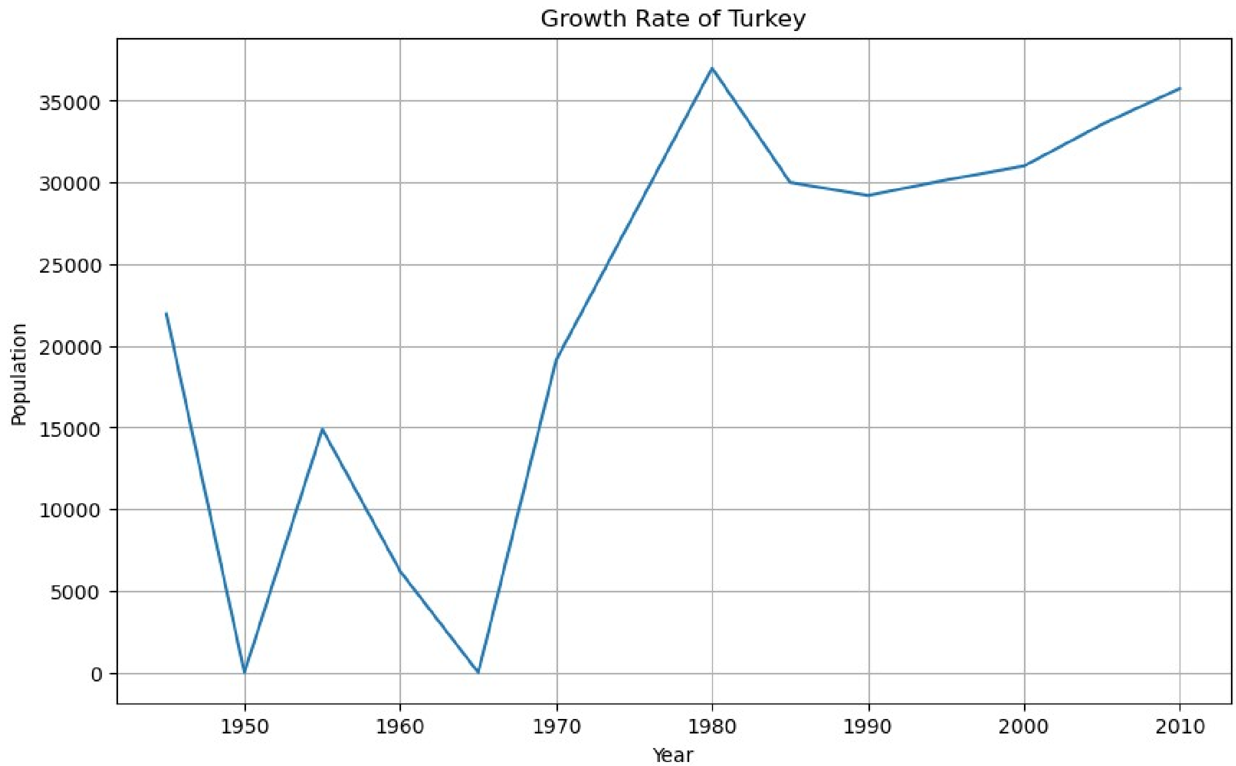



```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

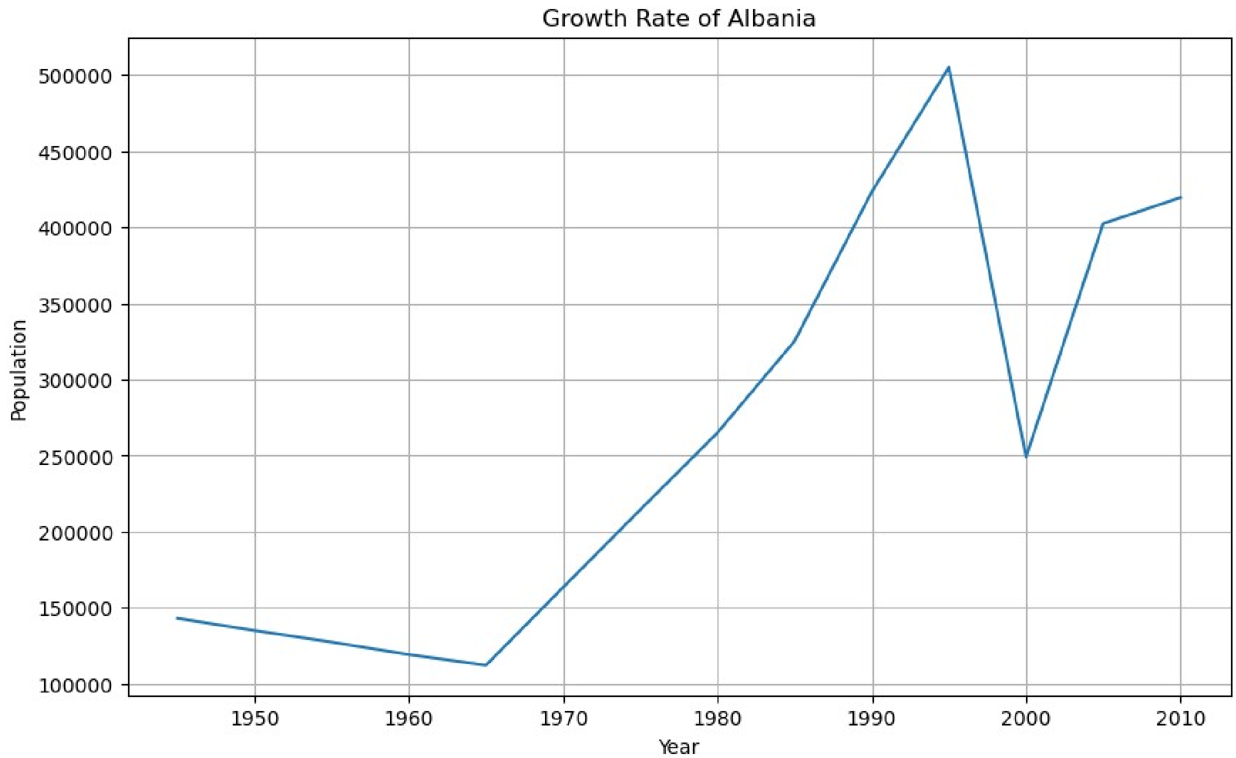



```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

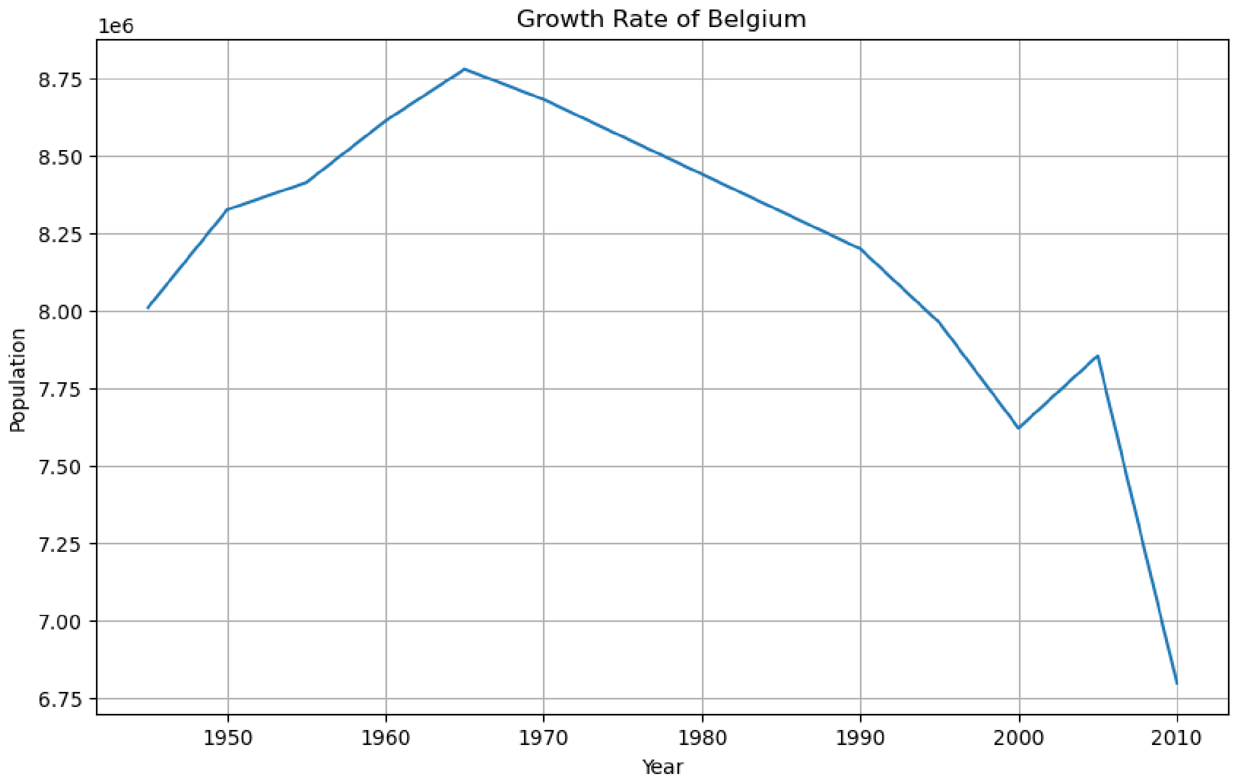

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



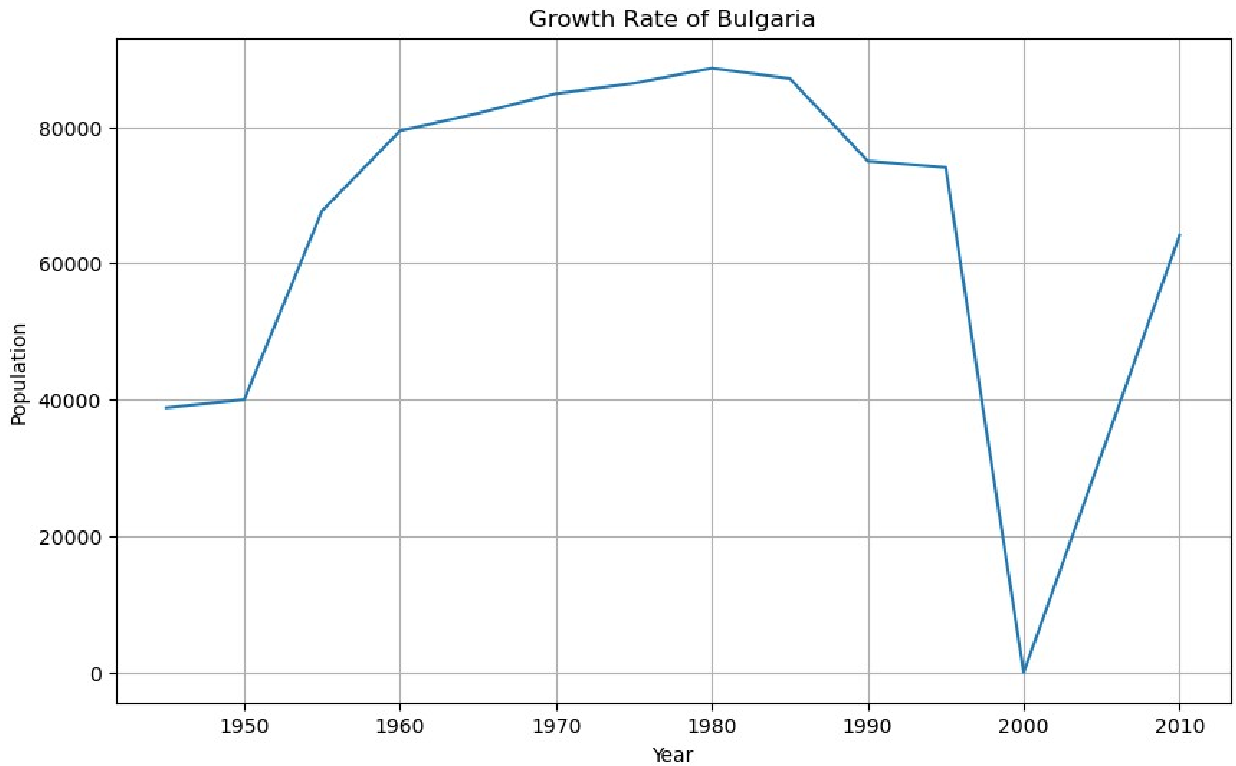

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



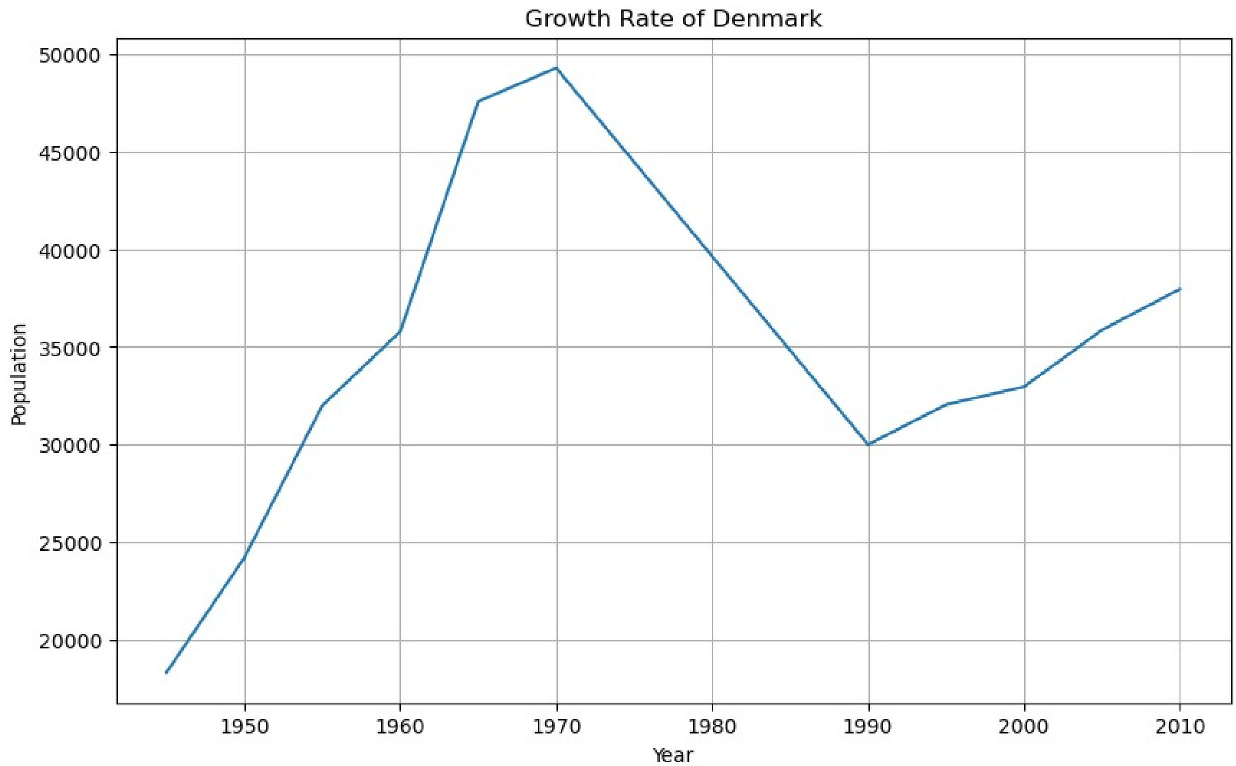

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



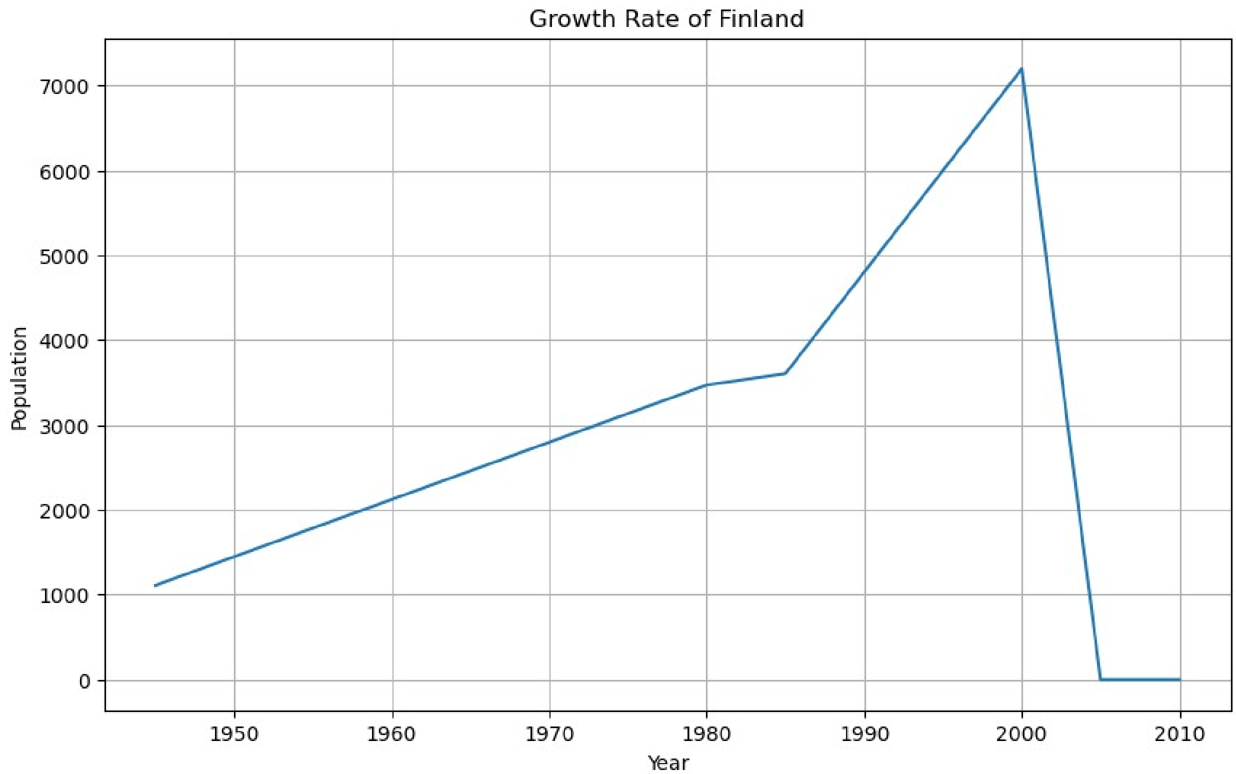

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



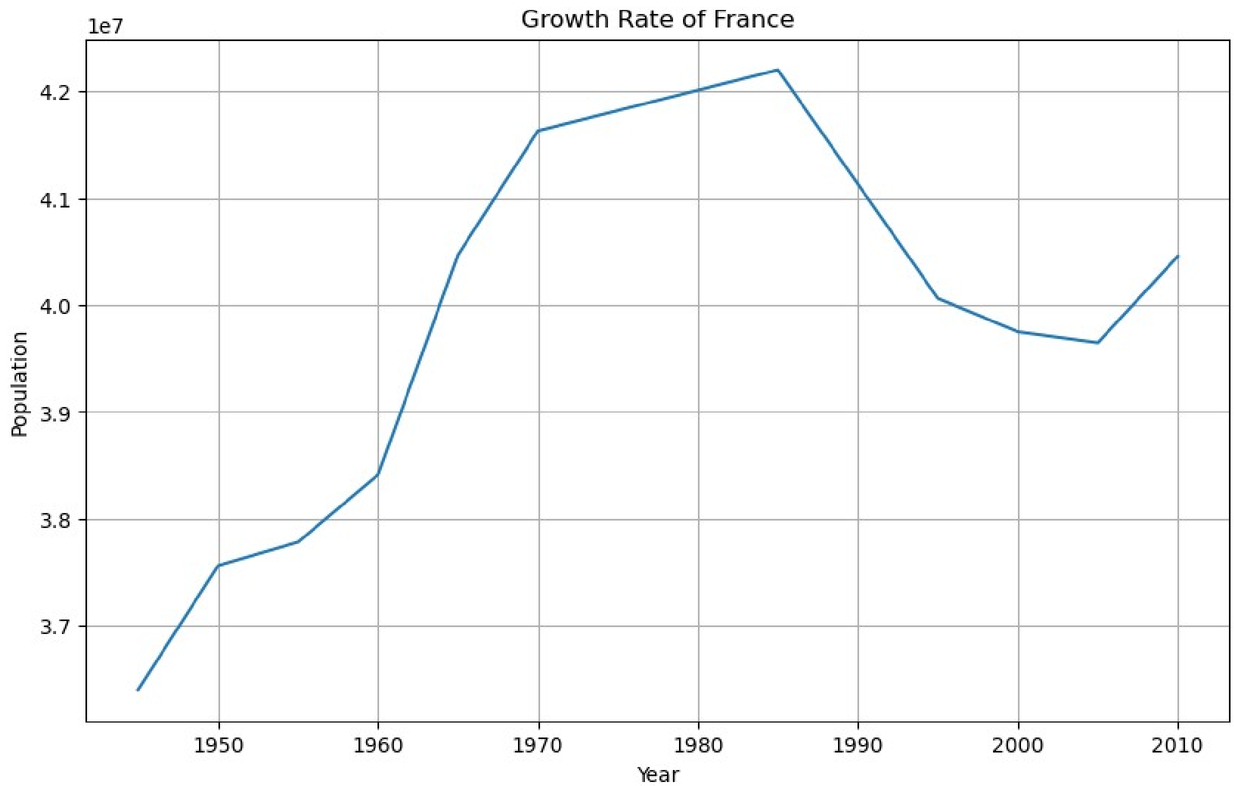

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



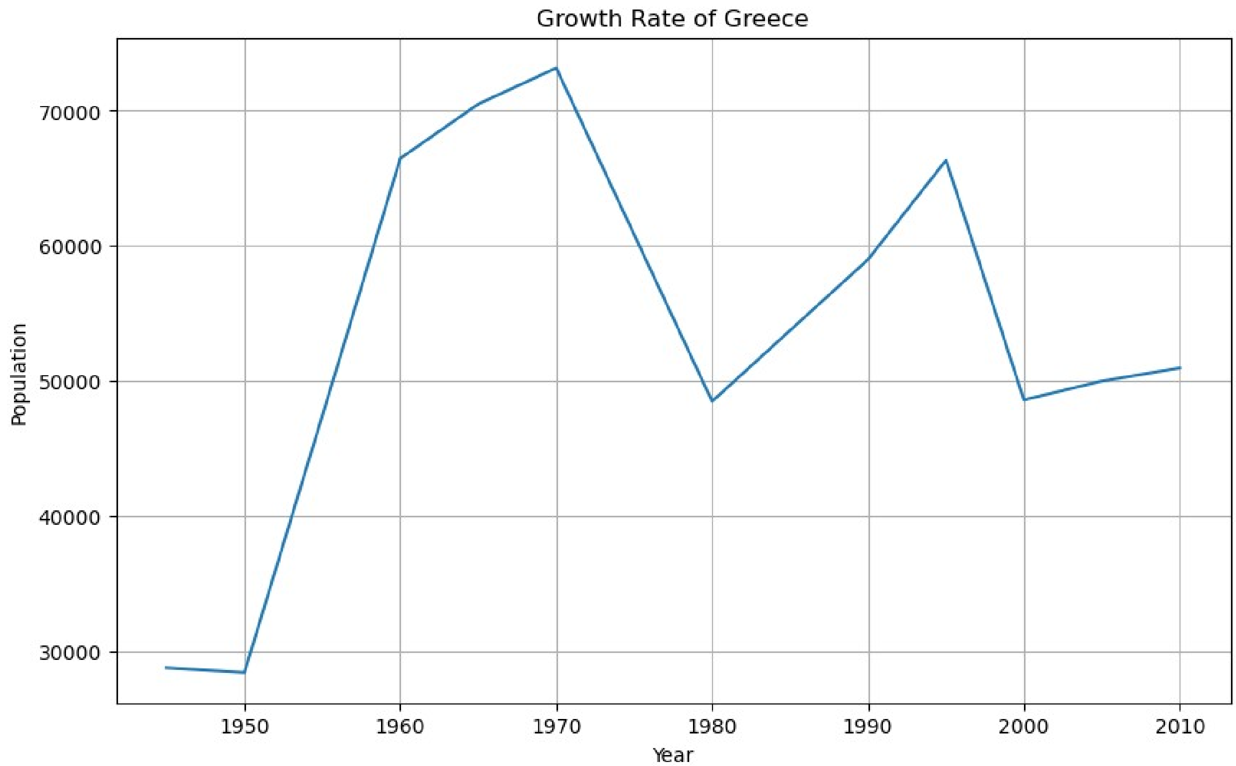

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



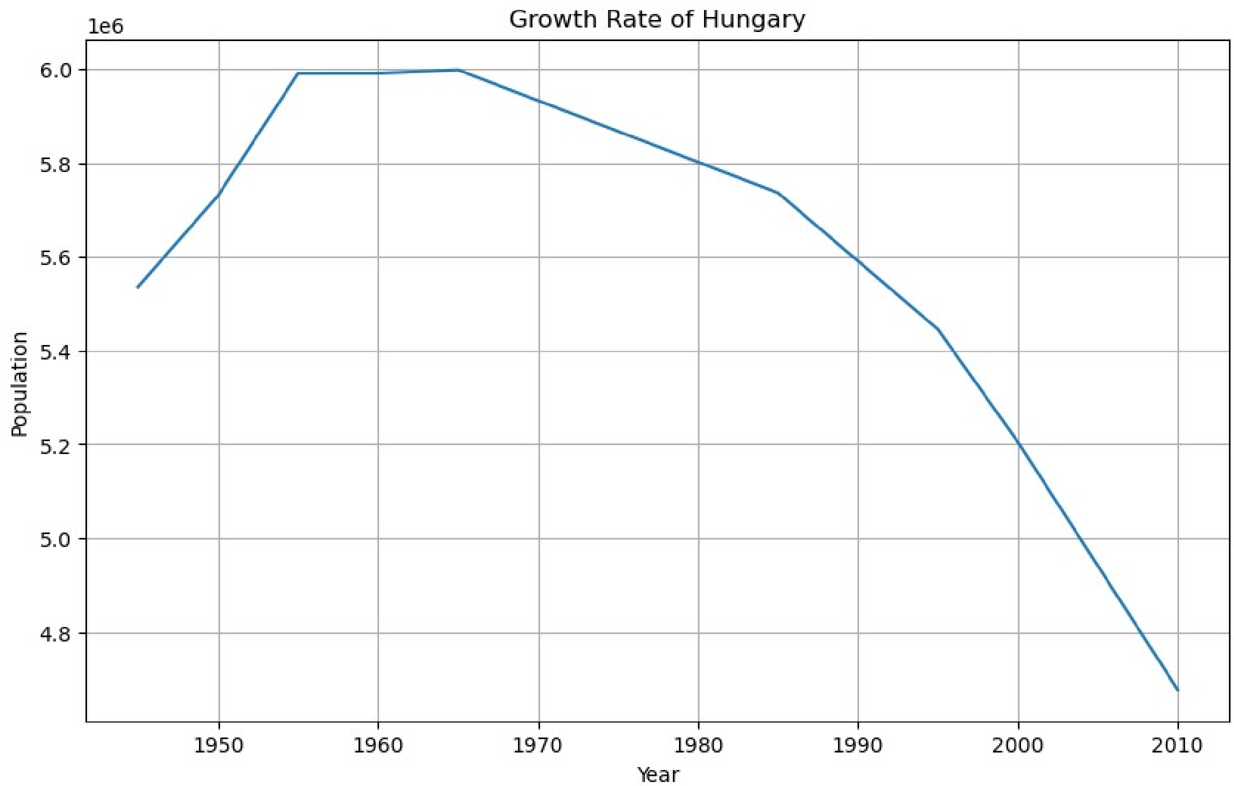

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



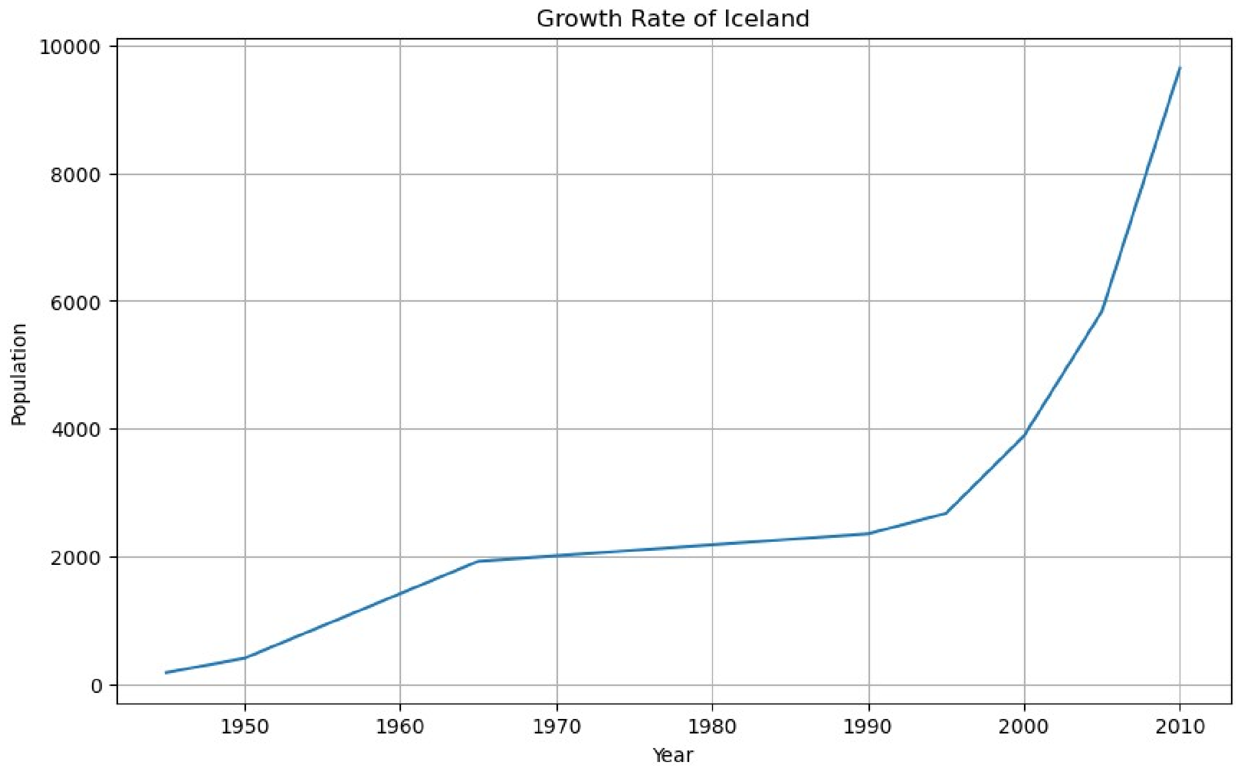

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



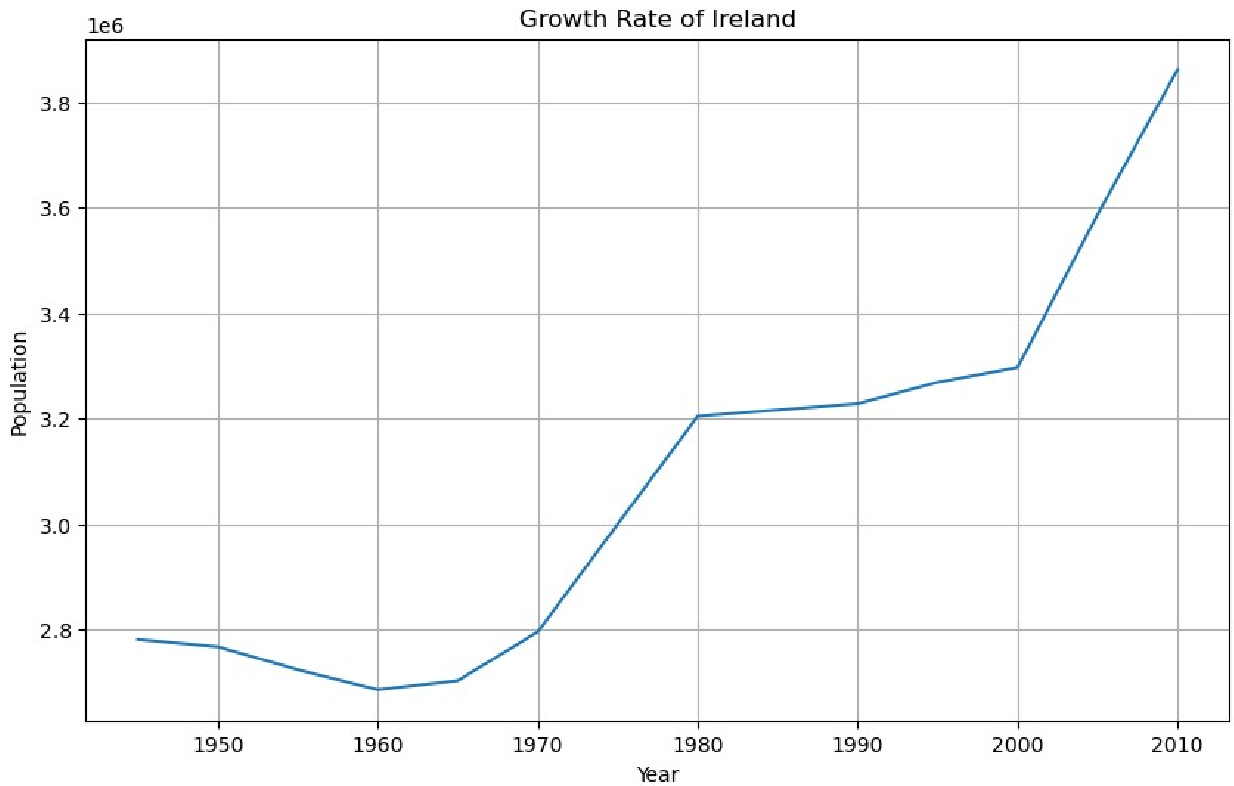

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



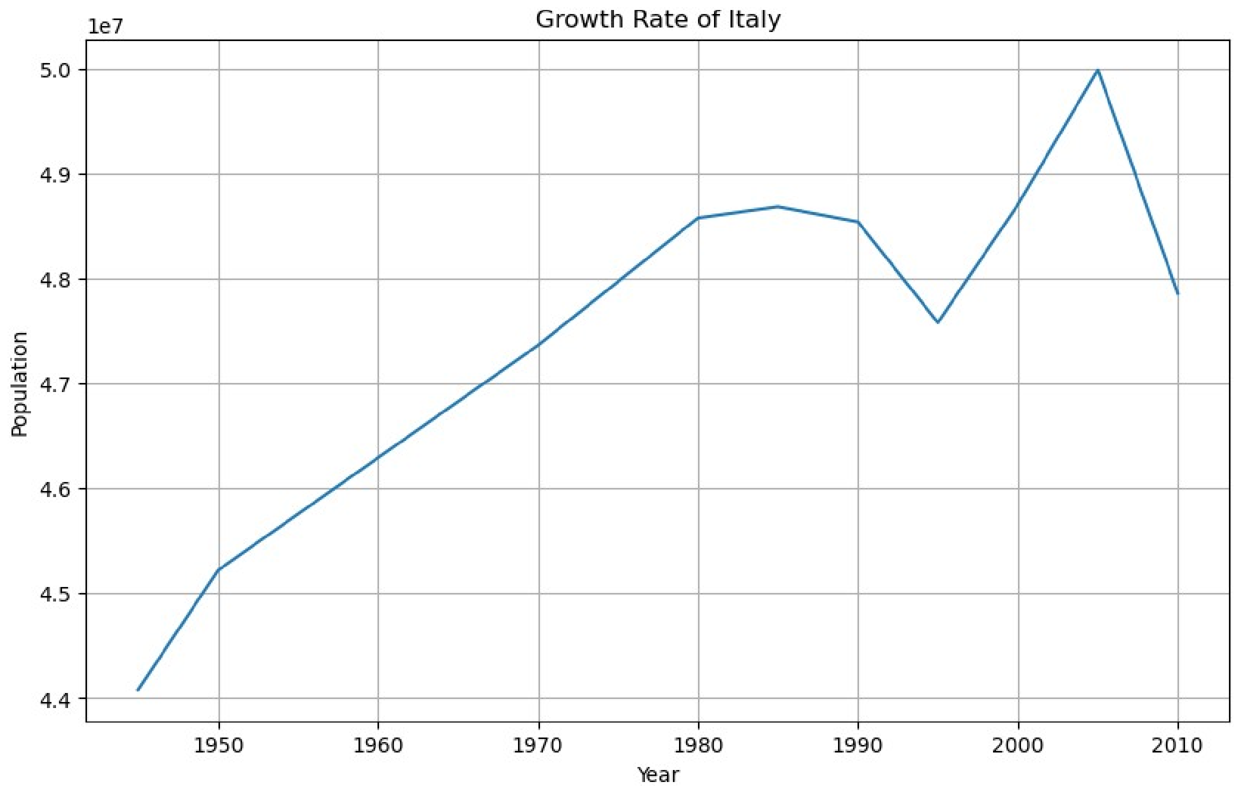

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



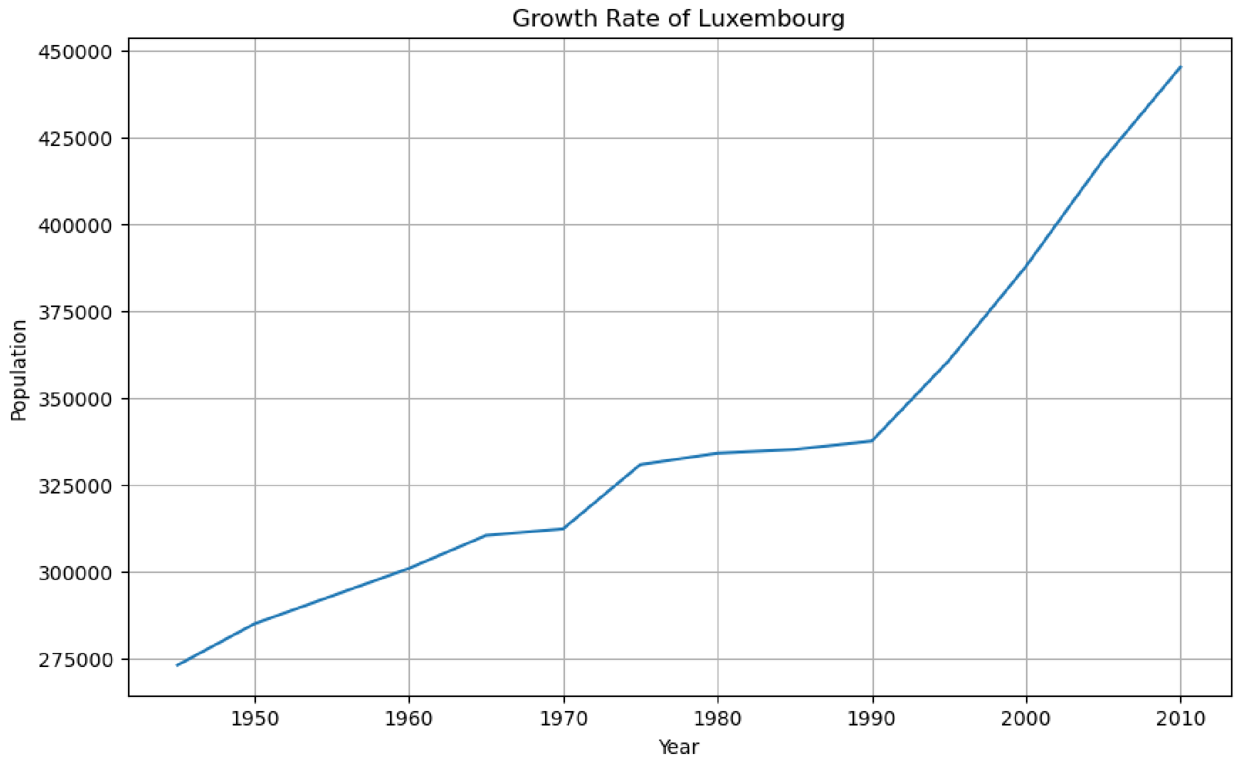

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



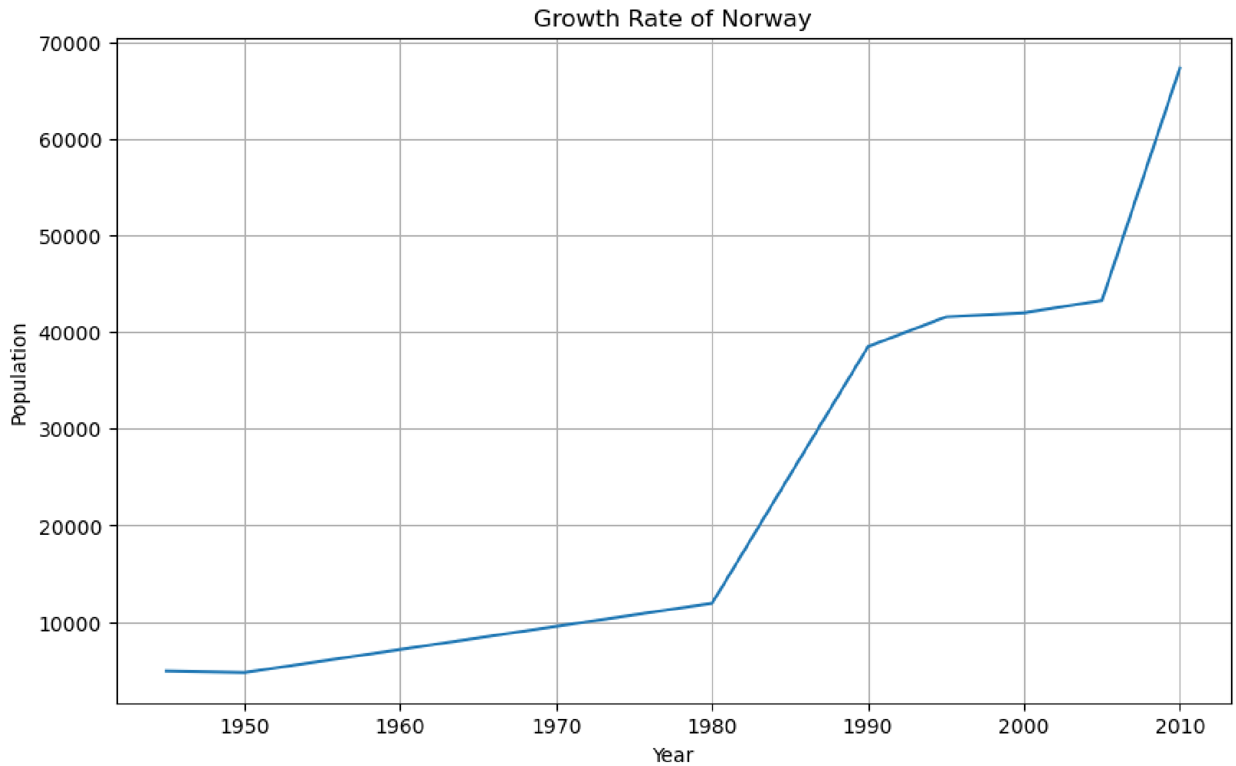

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



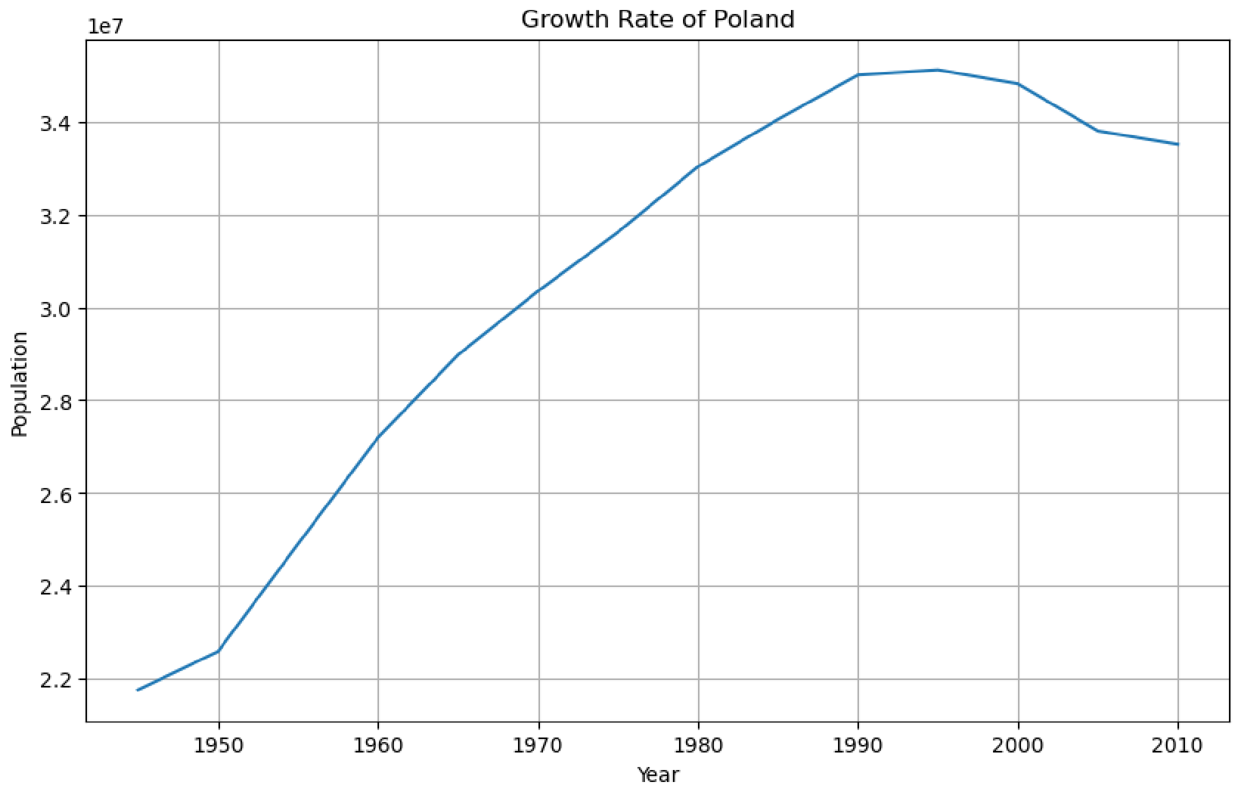

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



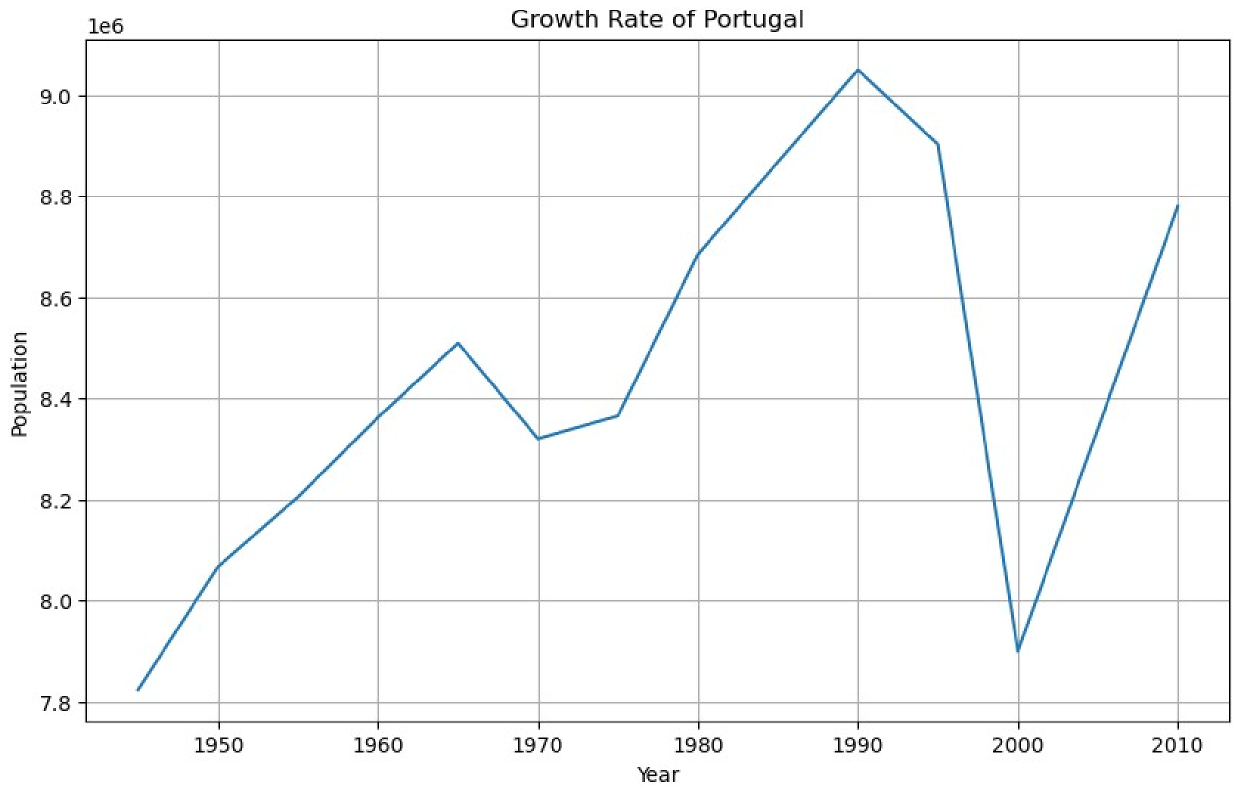

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



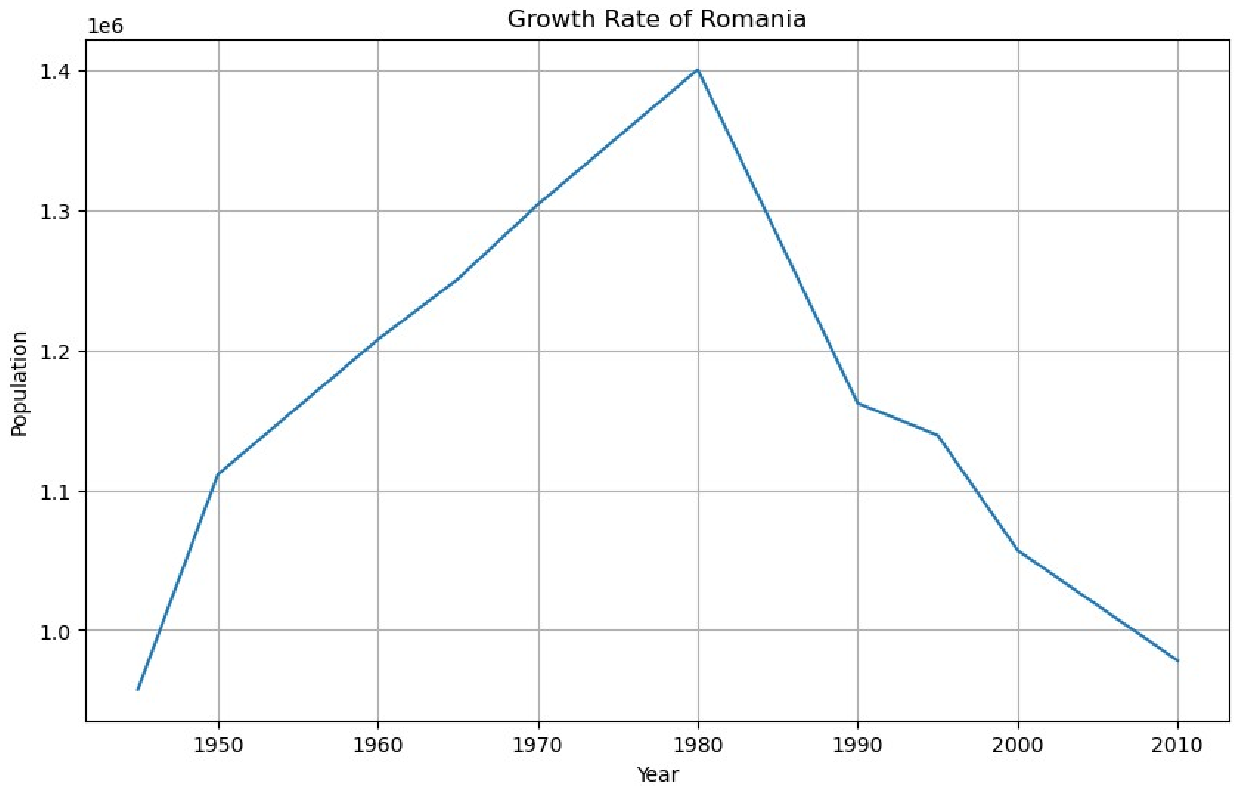

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```




```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

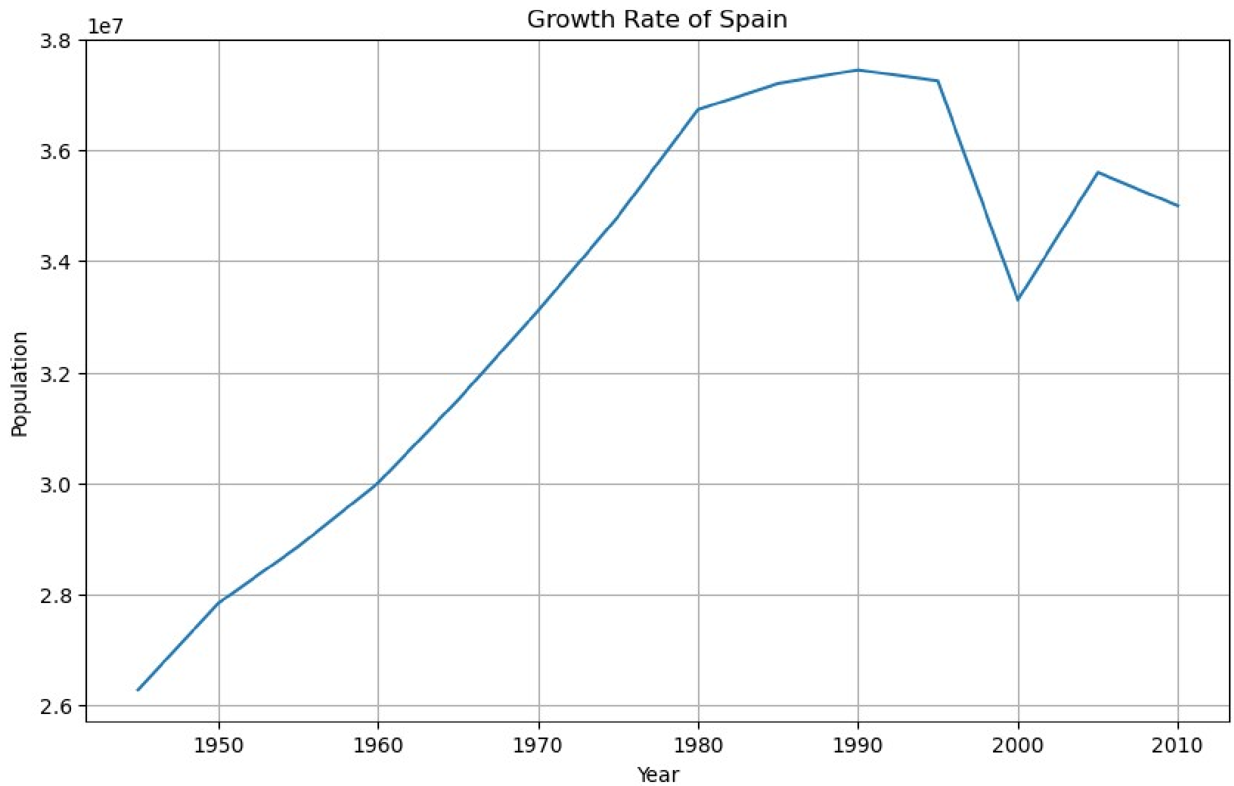



```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

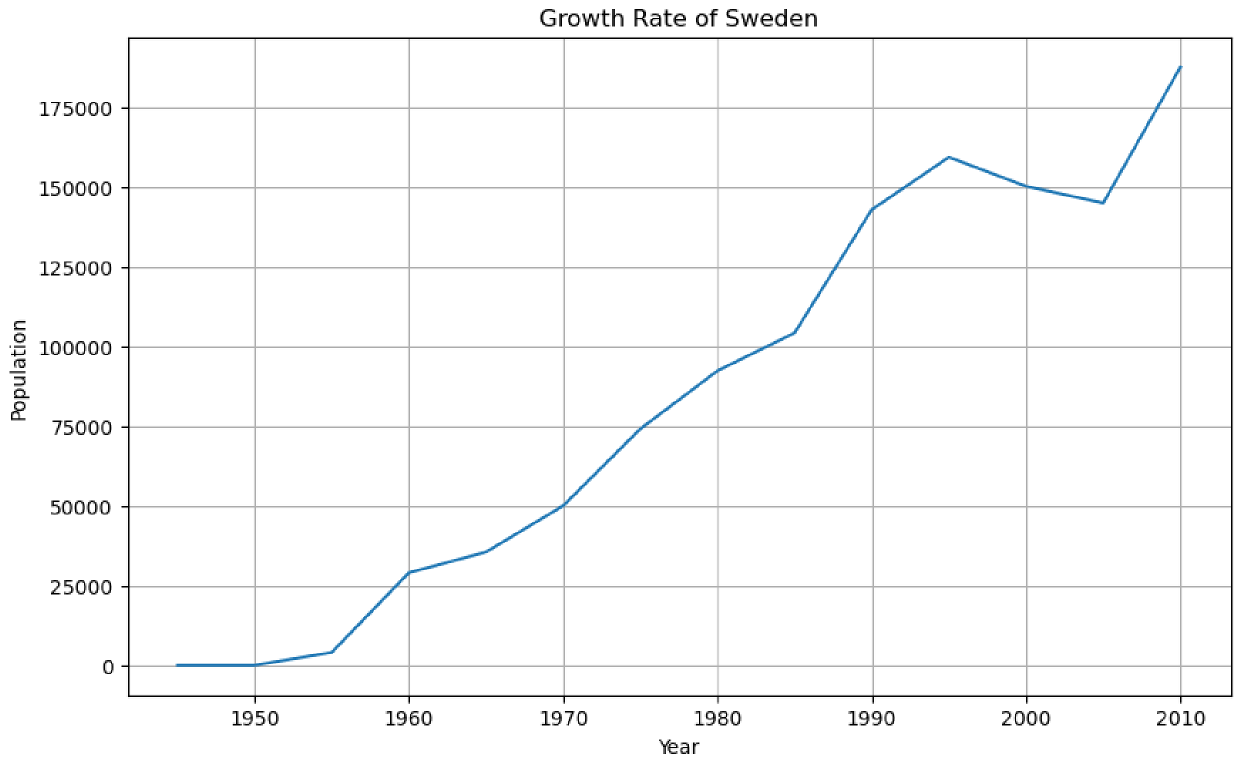



```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

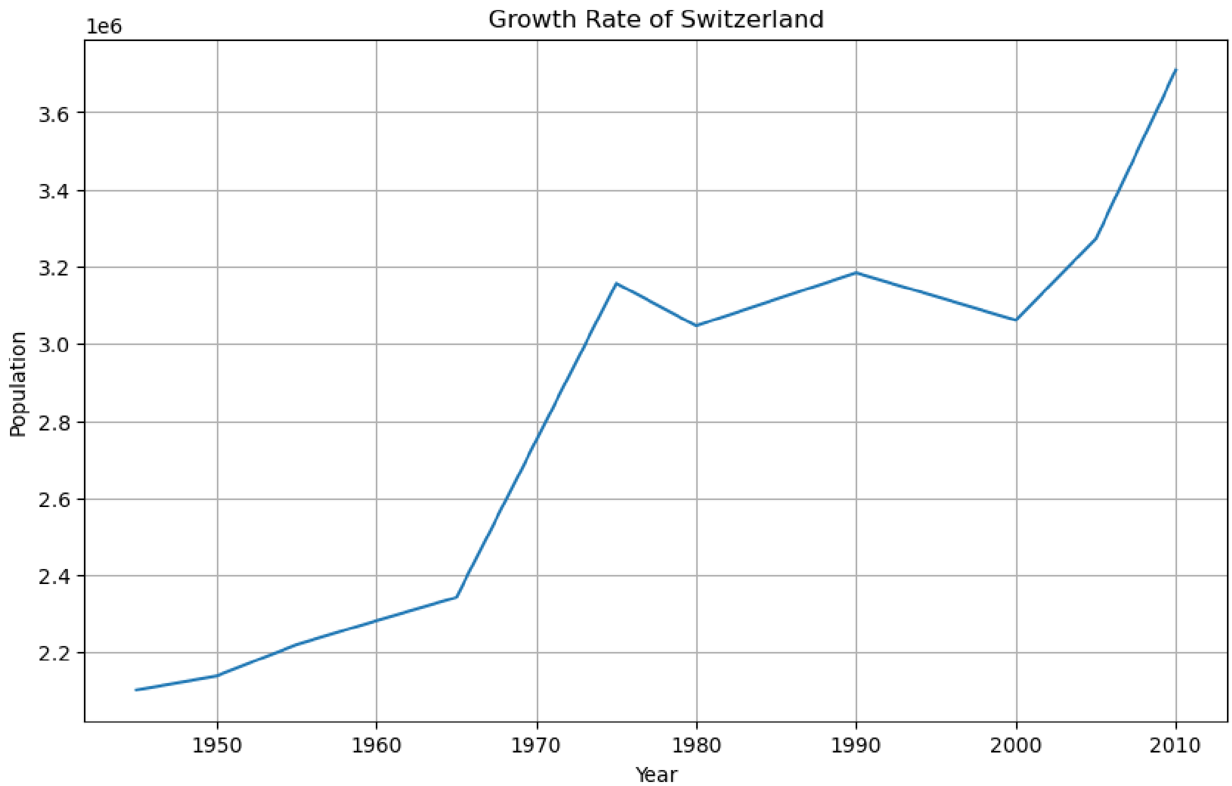

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



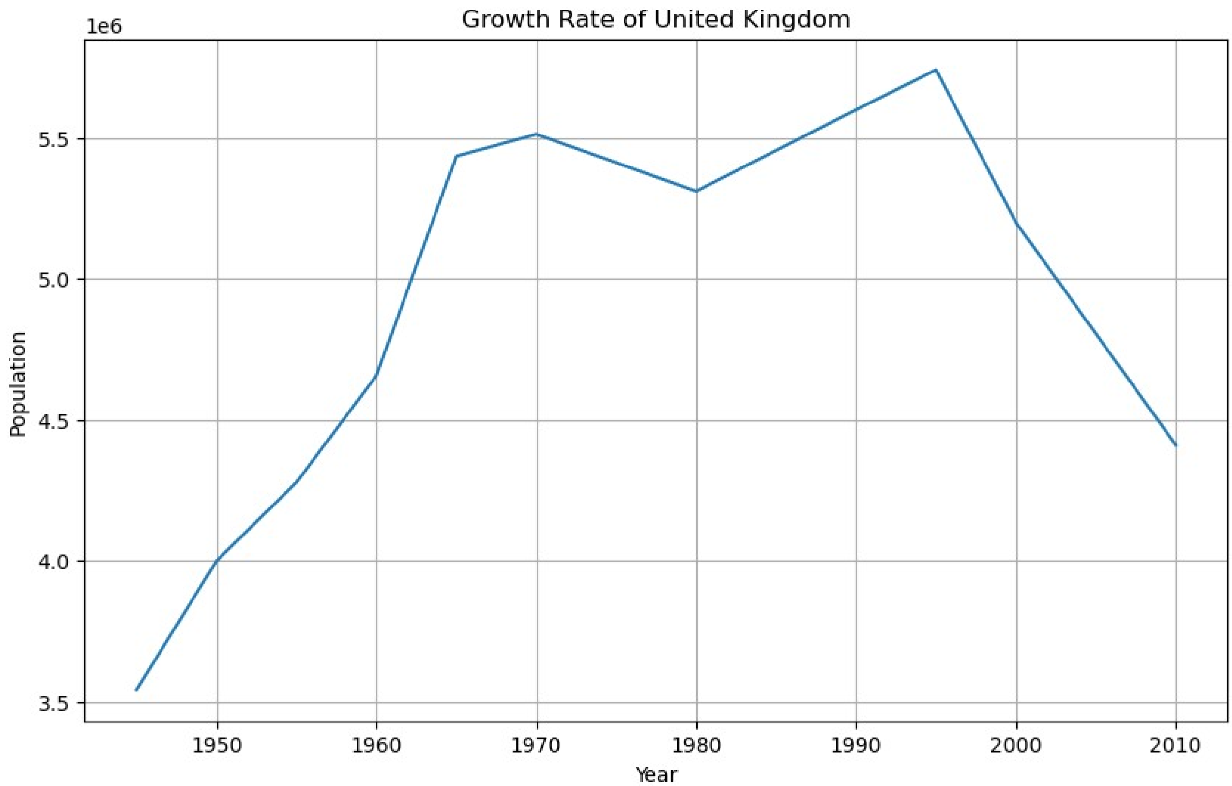

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



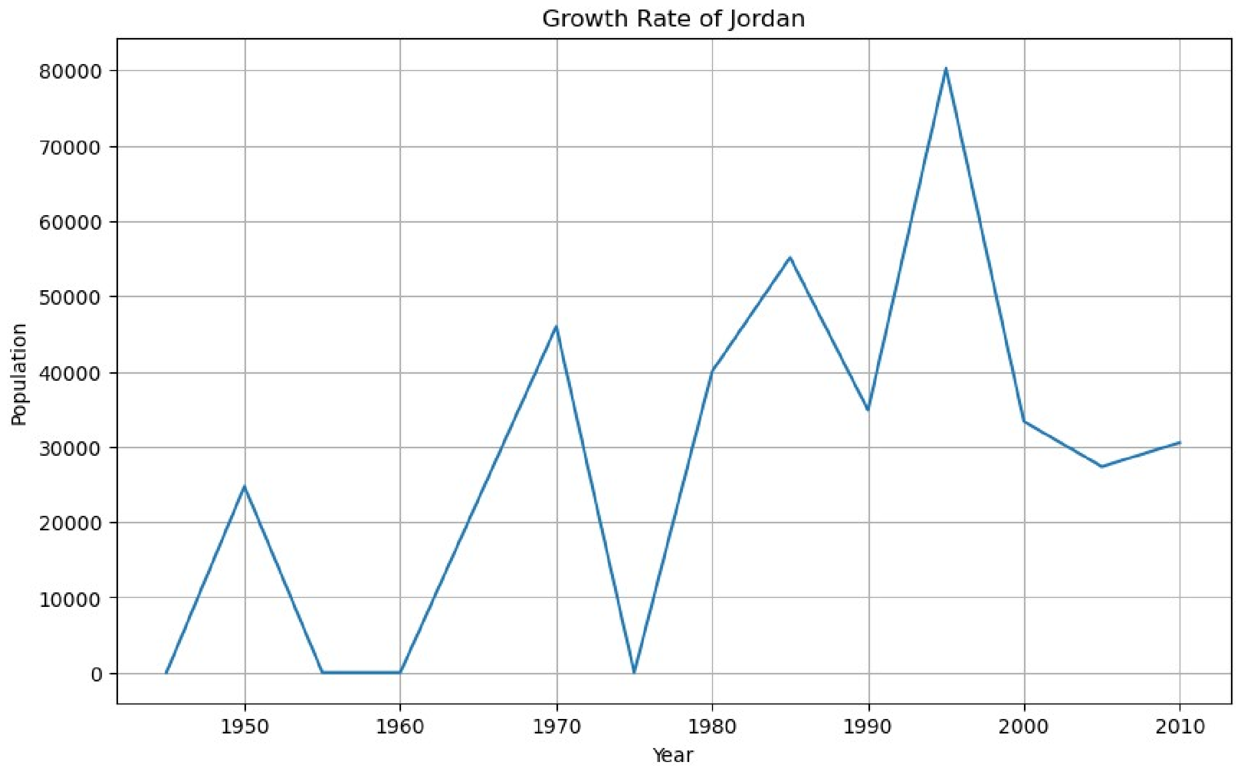

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



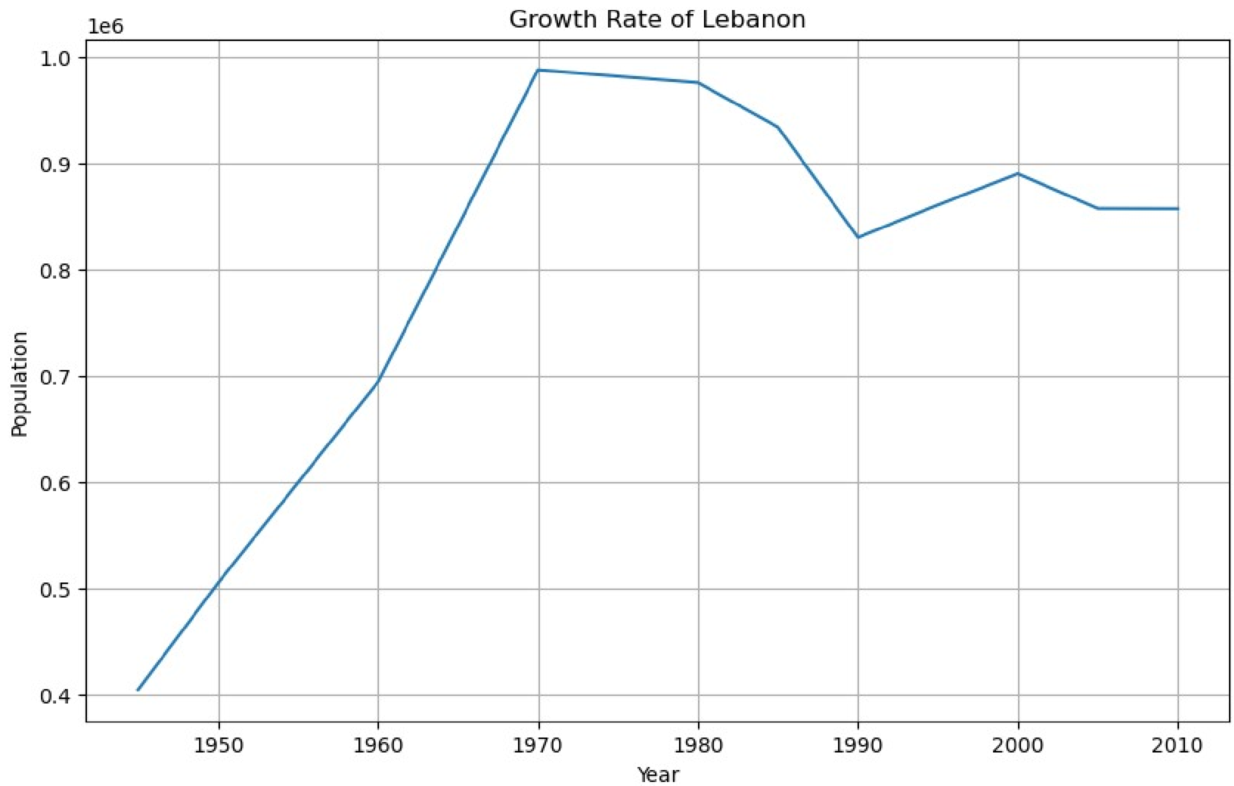

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



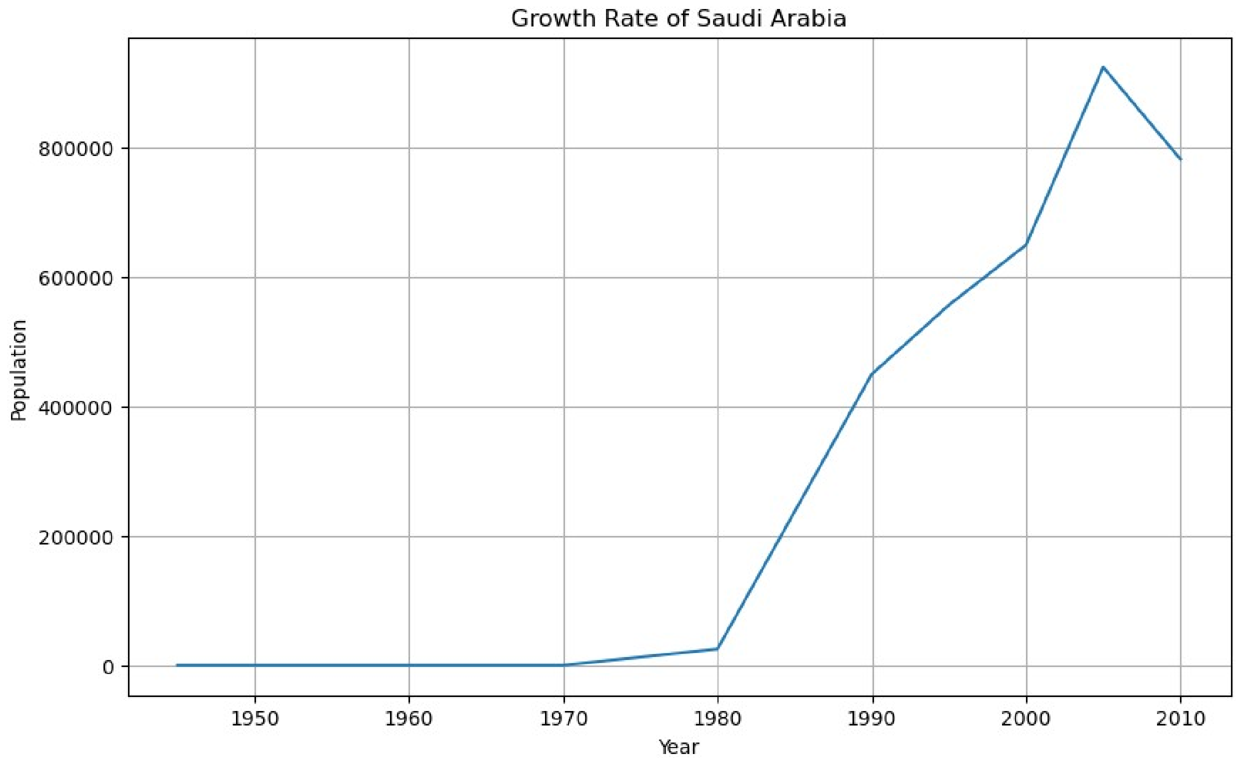

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



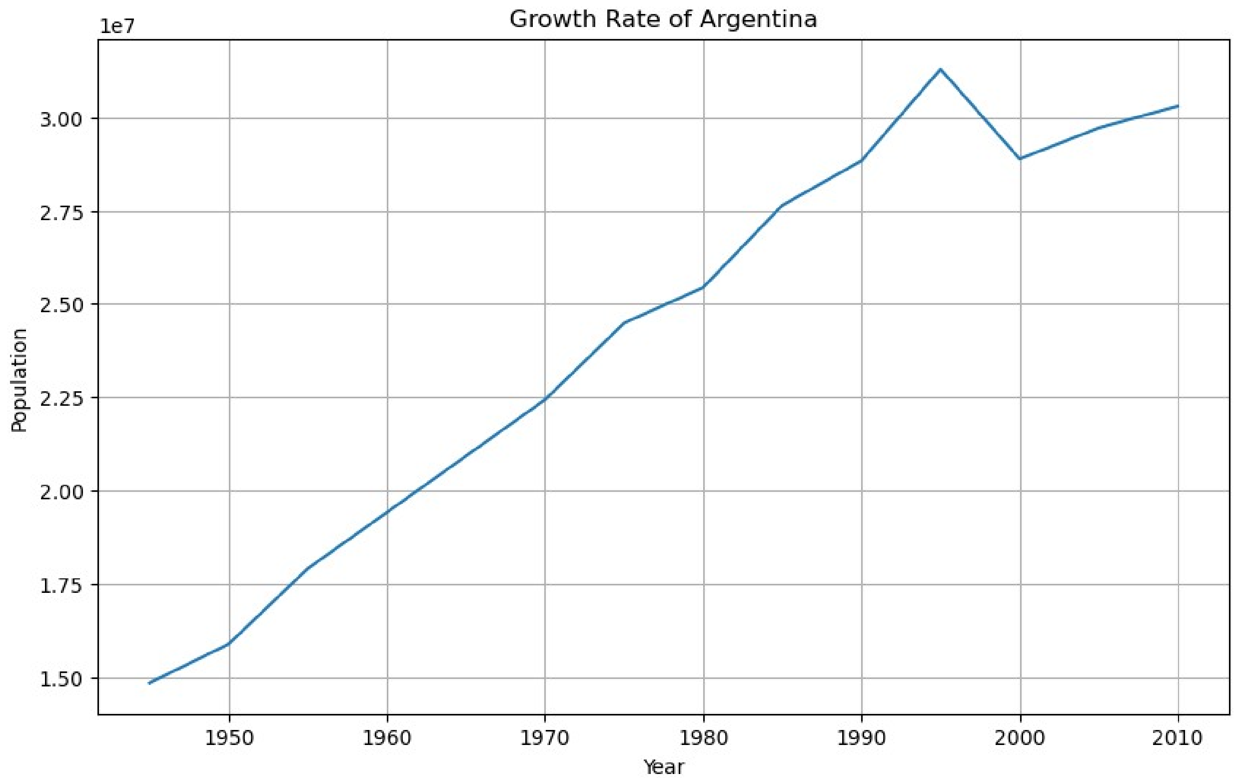

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



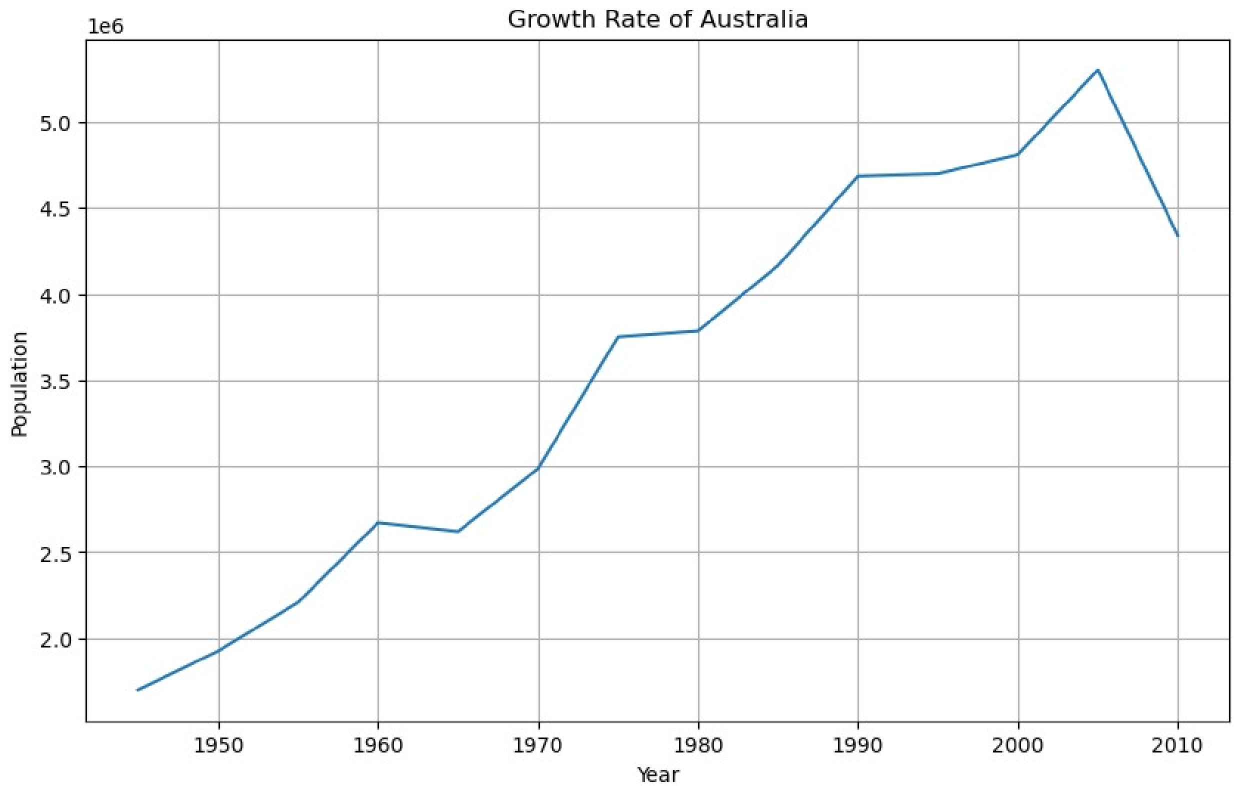

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



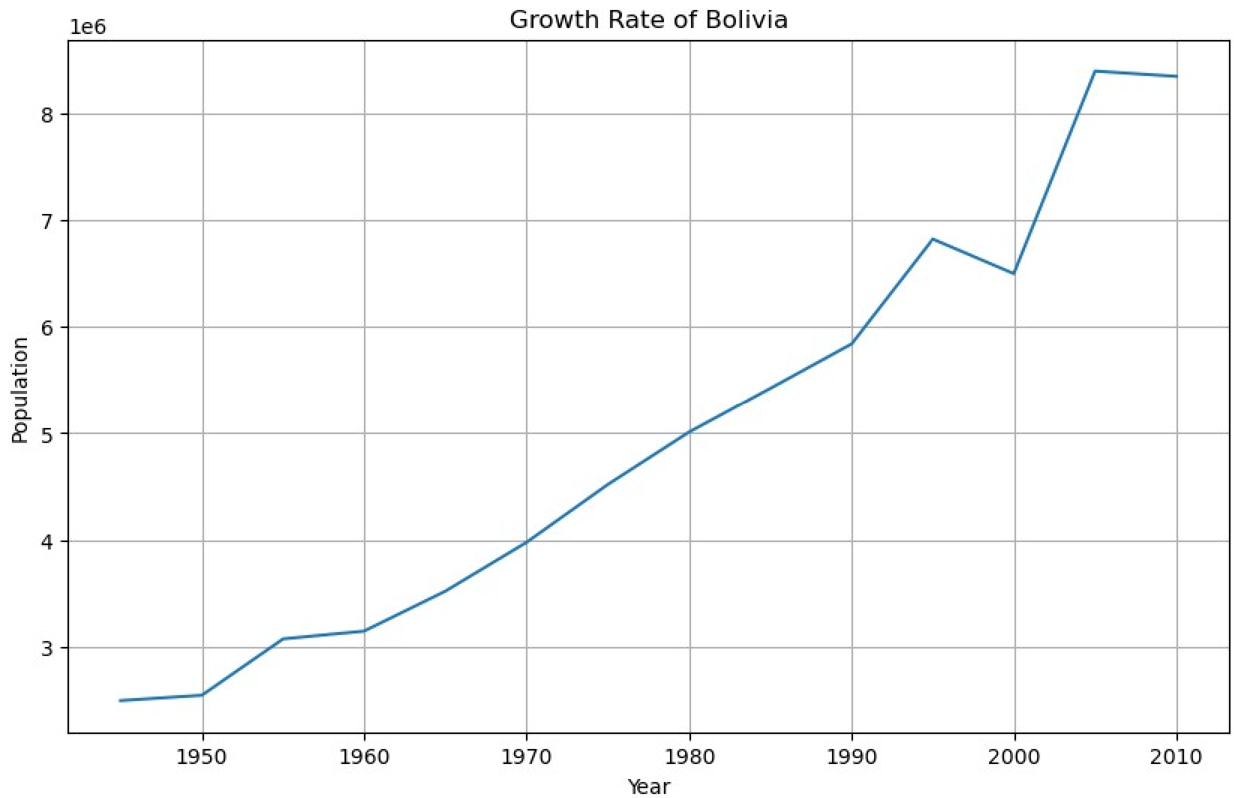

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



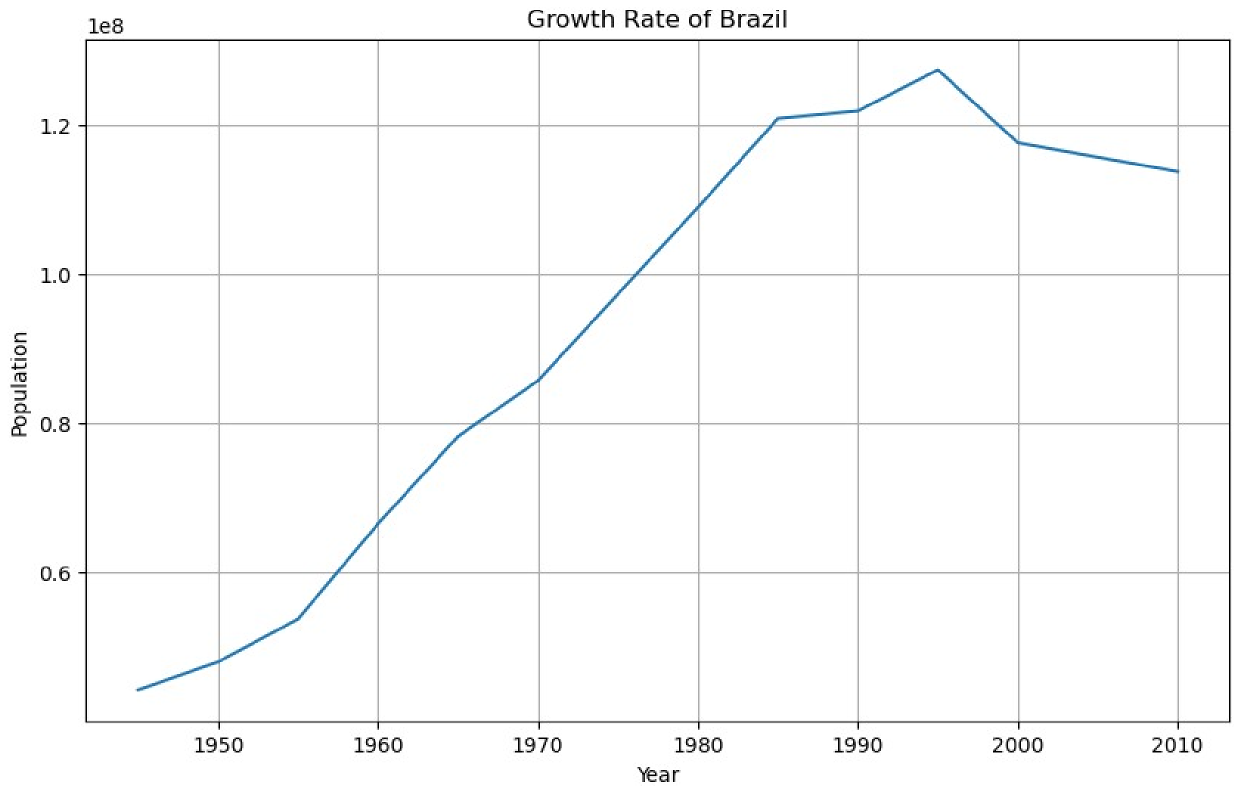

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



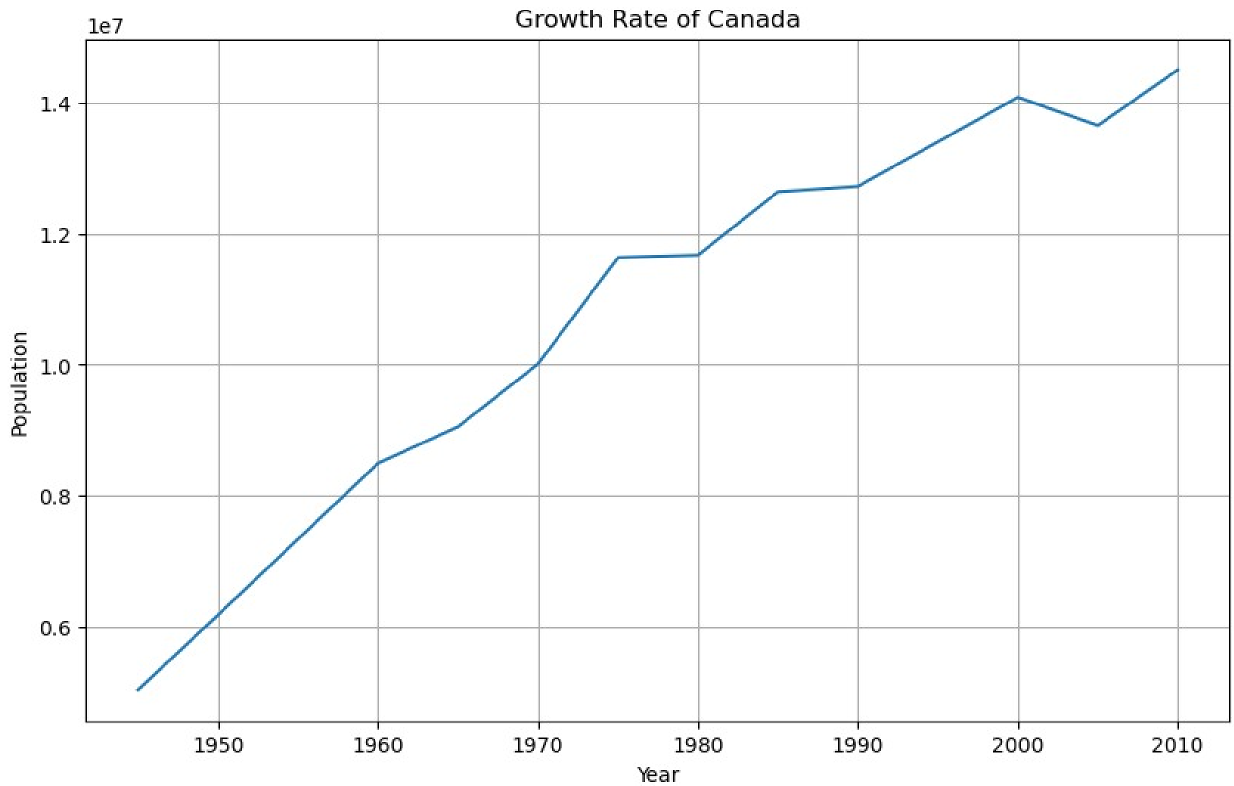

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



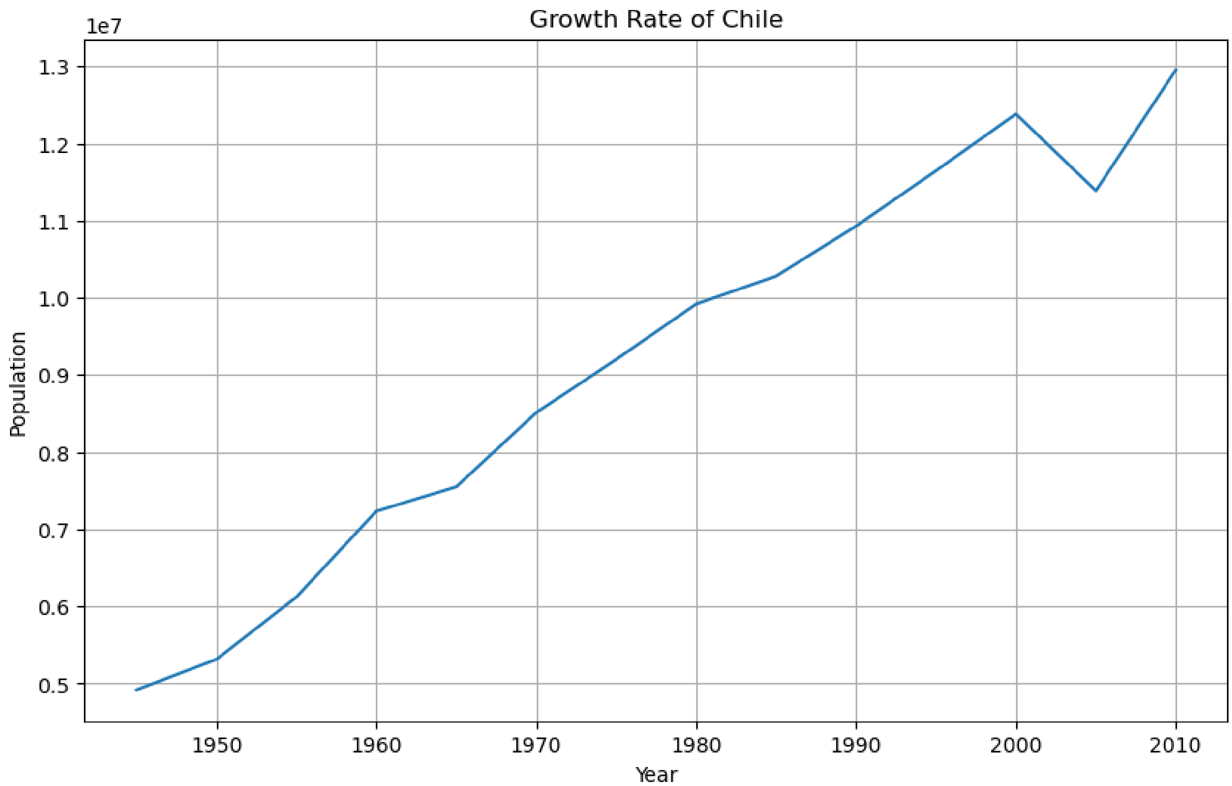

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



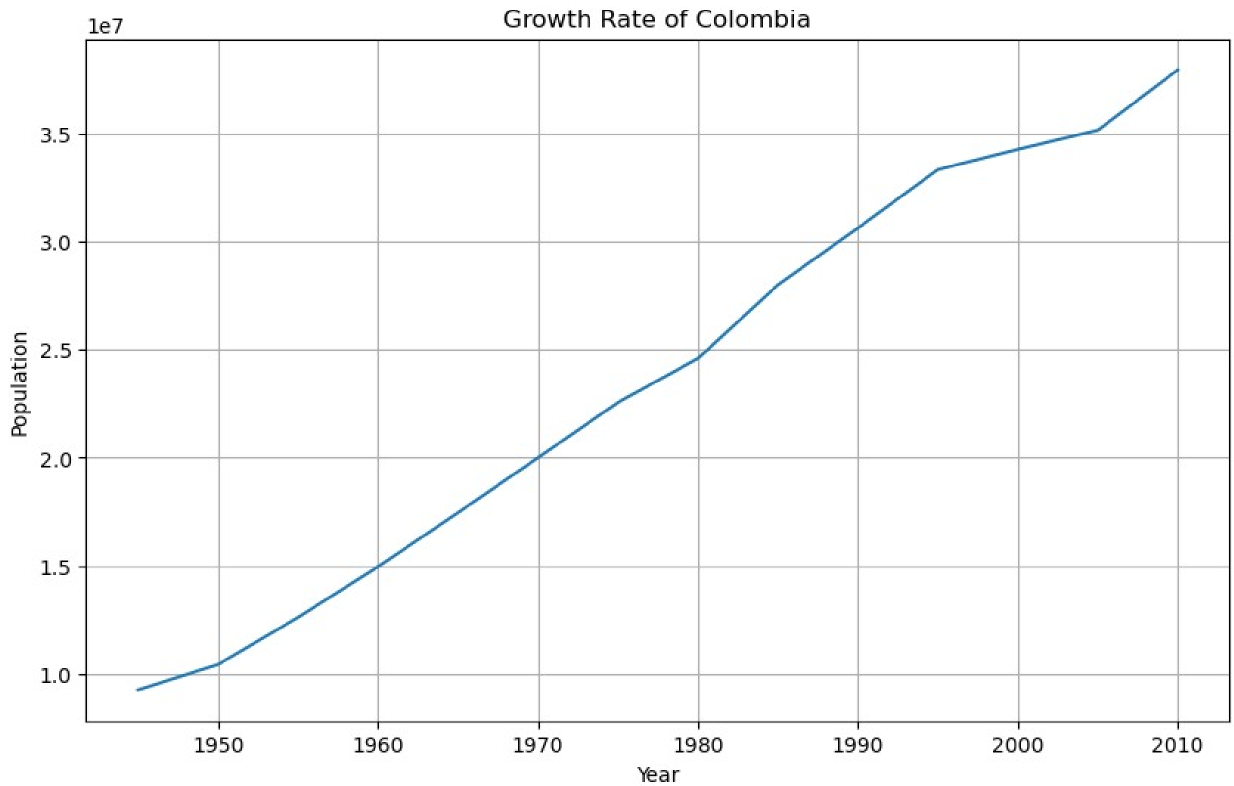

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



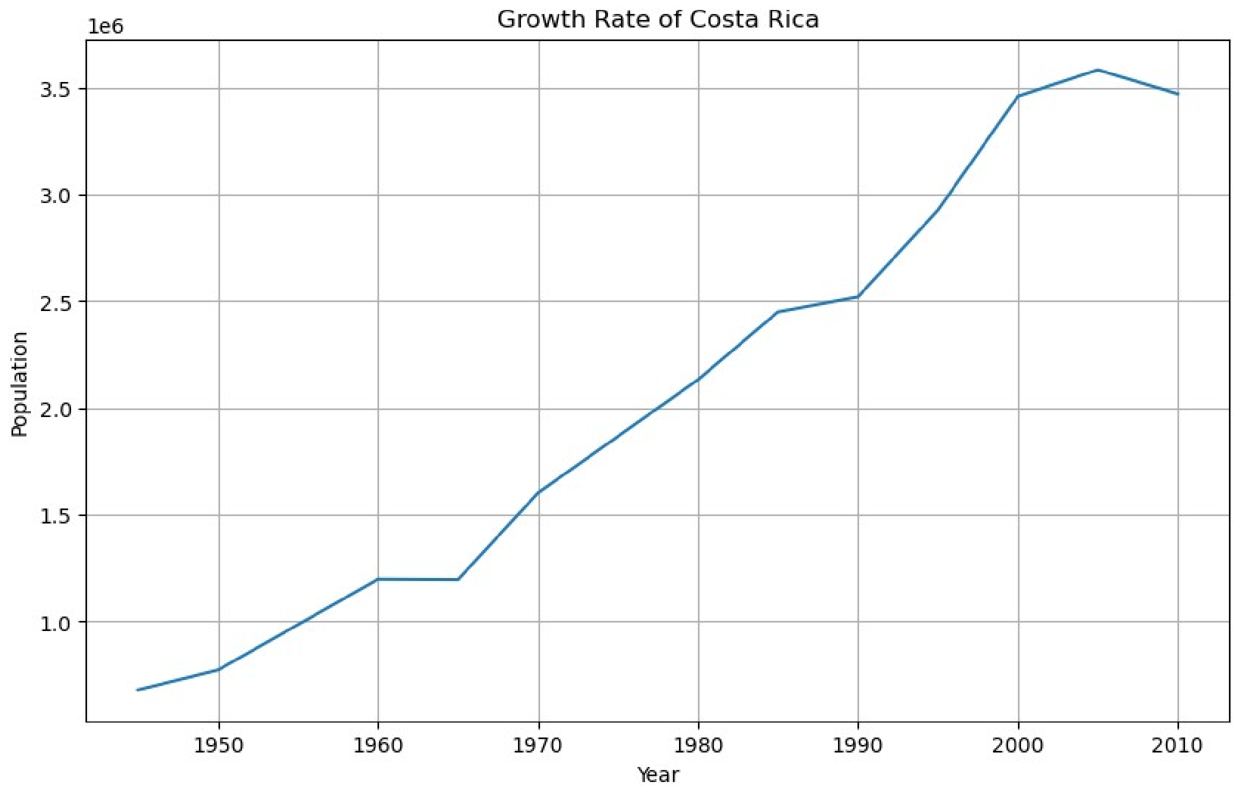

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



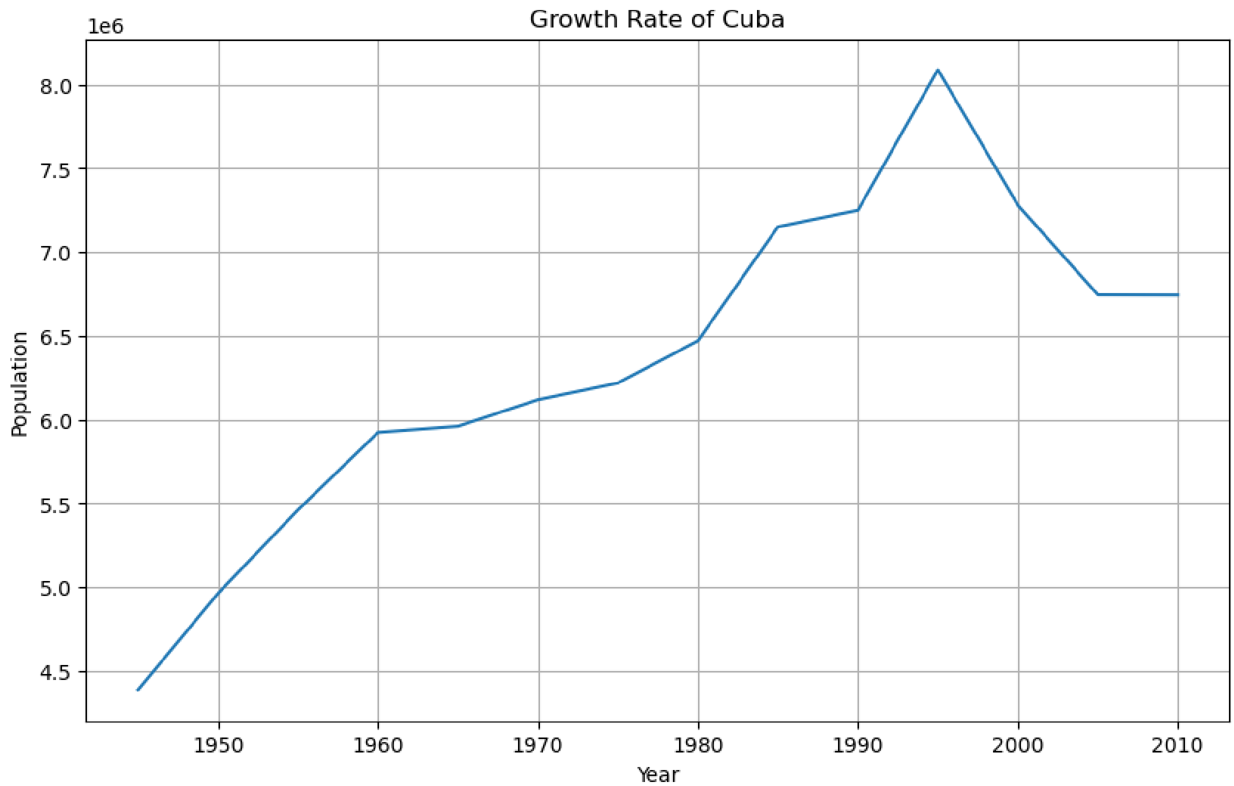

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



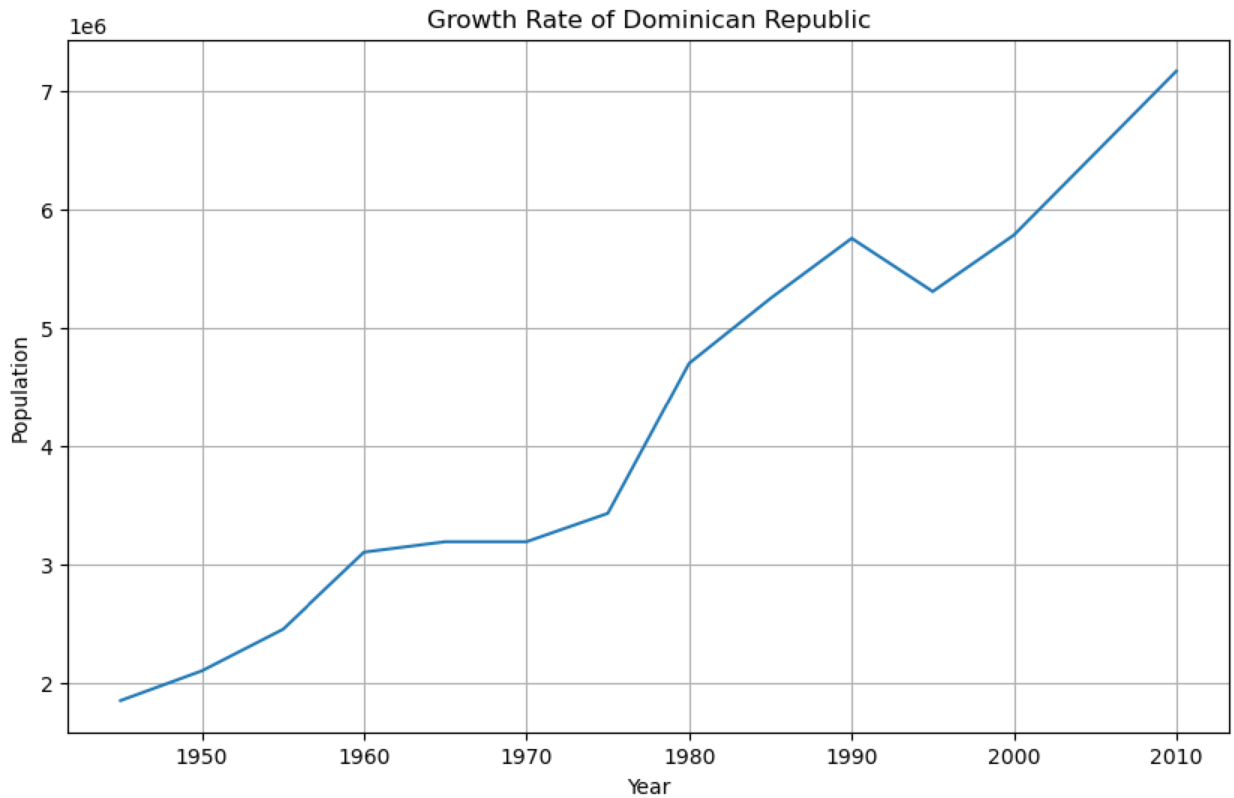

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



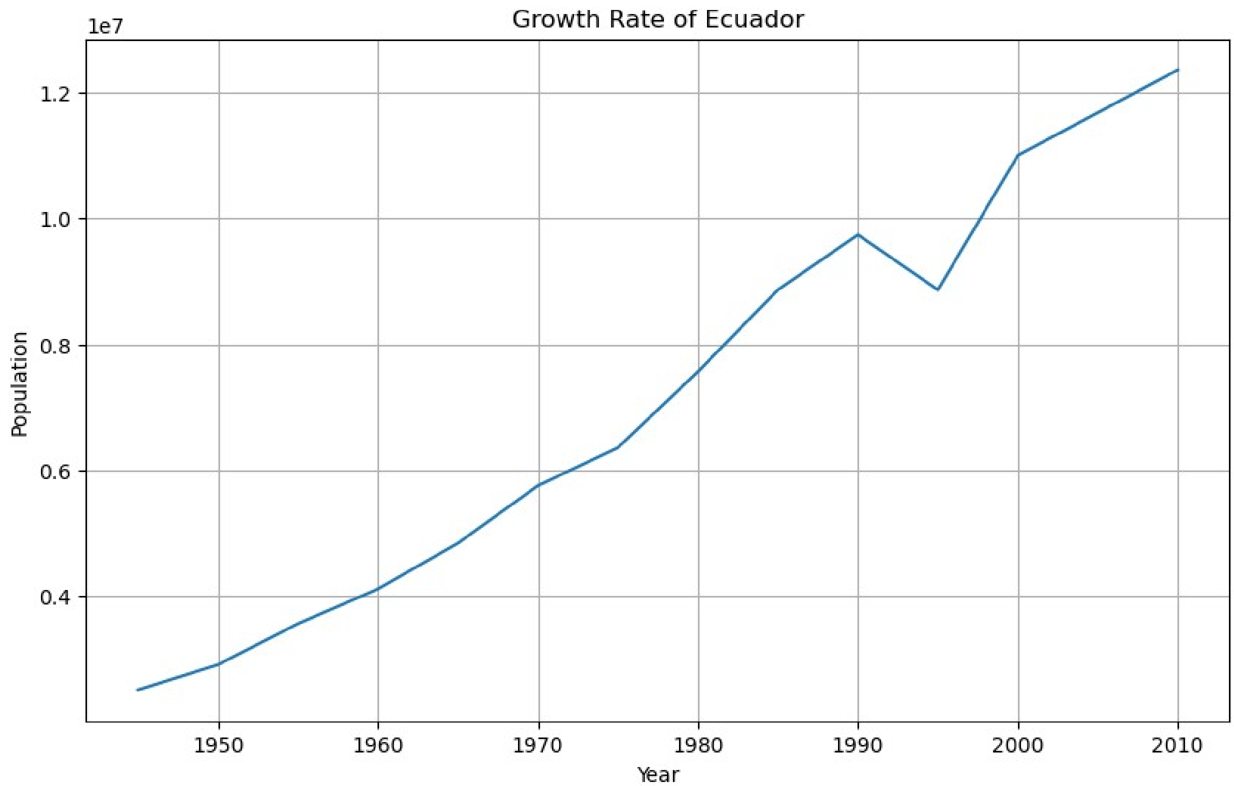

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: 
FutureWarning: use_inf_as_na option is deprecated and will be removed 
in a future version. Convert inf values to NaN before operating 
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: 
FutureWarning: use_inf_as_na option is deprecated and will be removed 
in a future version. Convert inf values to NaN before operating 
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



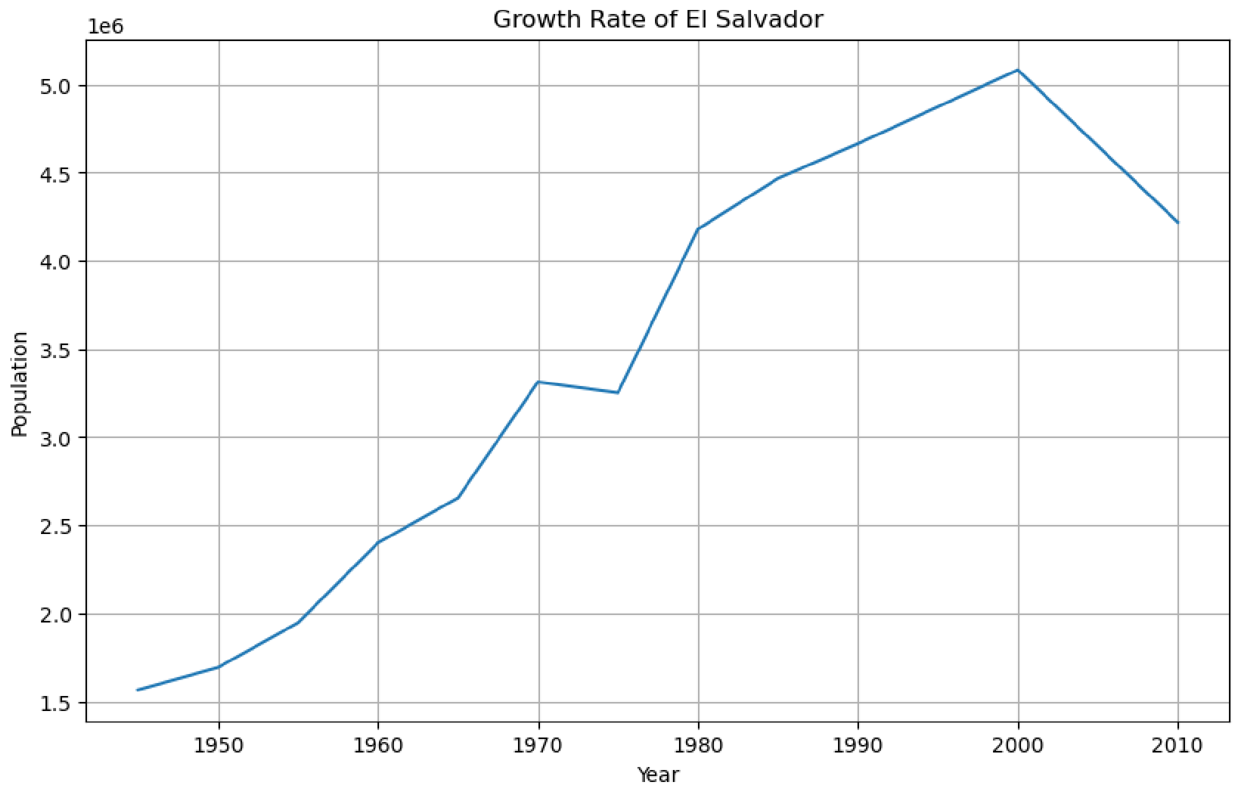

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



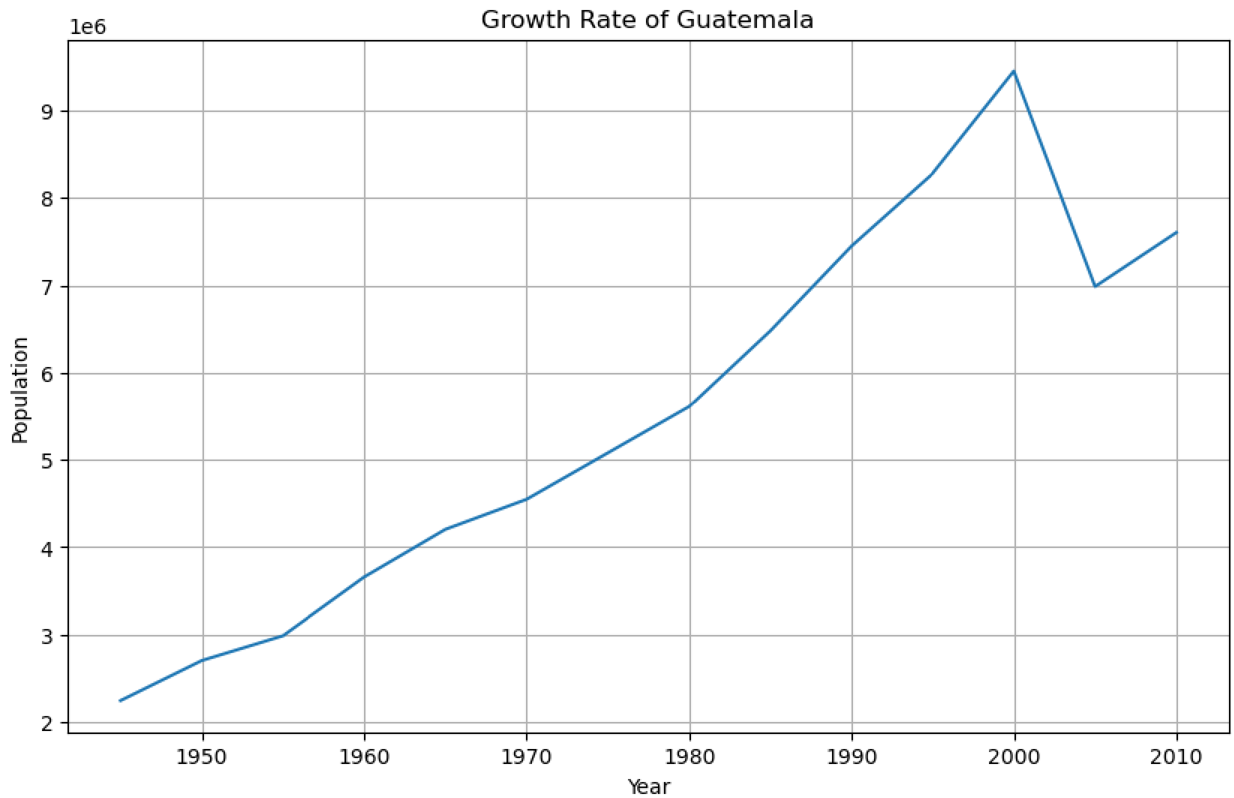

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



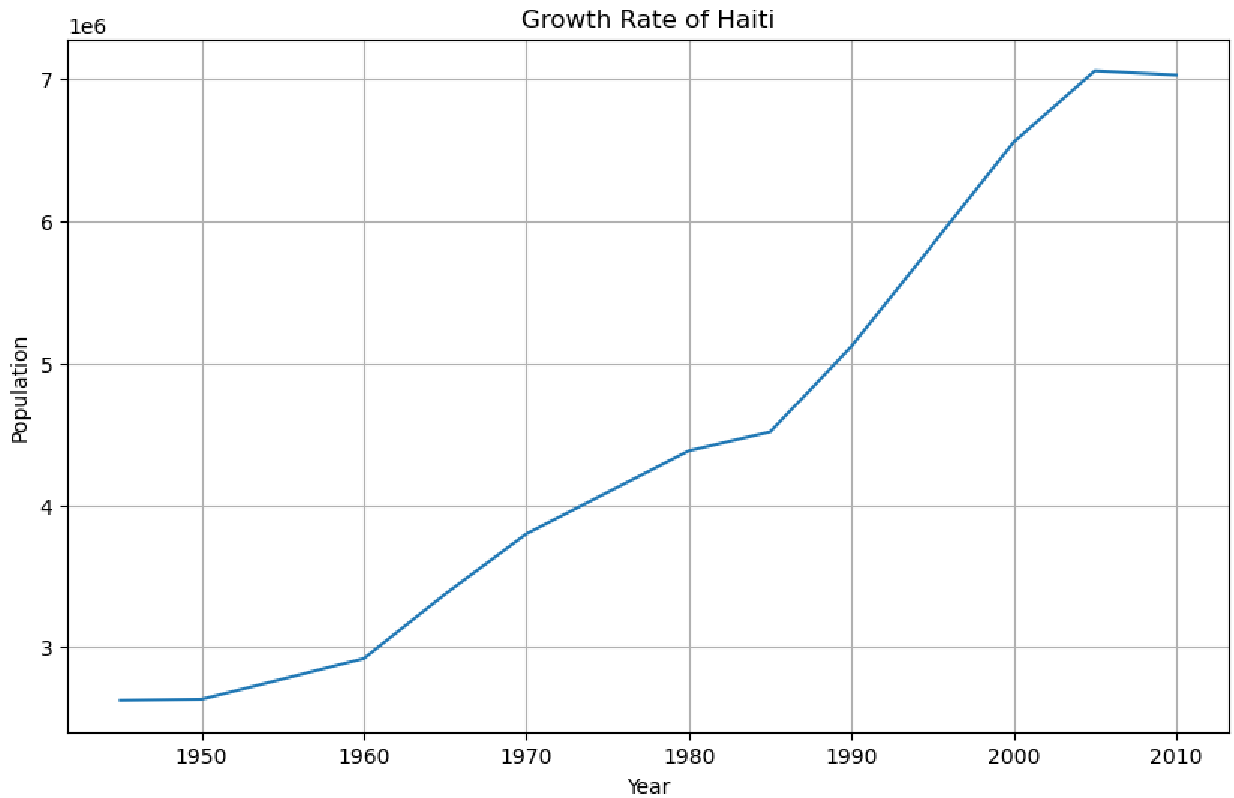

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



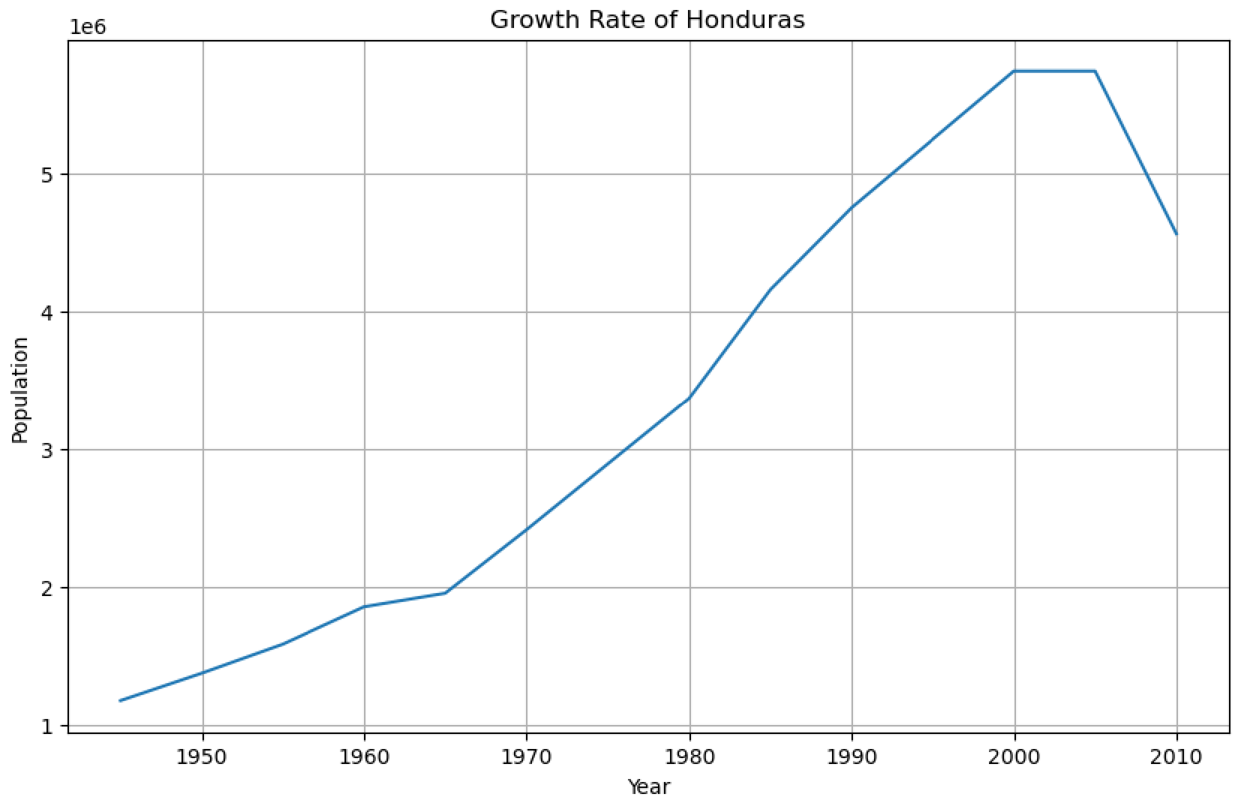

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



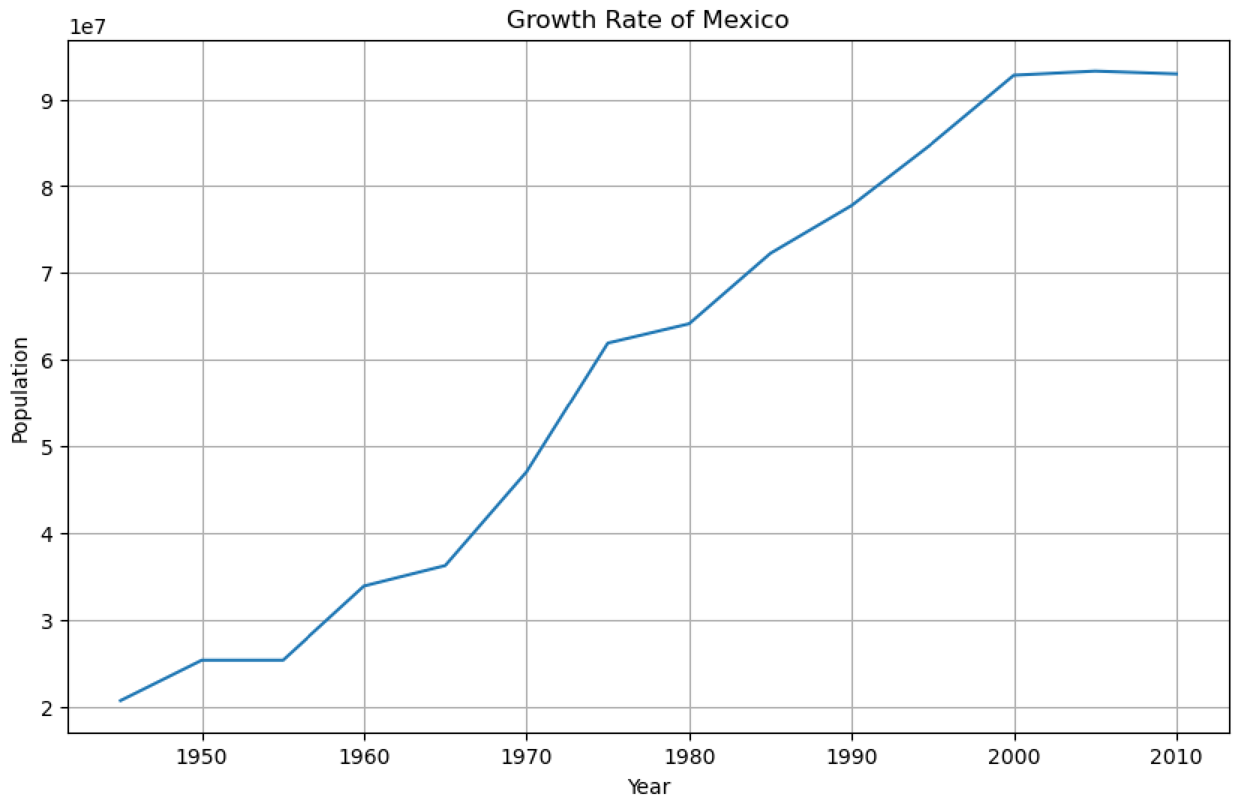

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



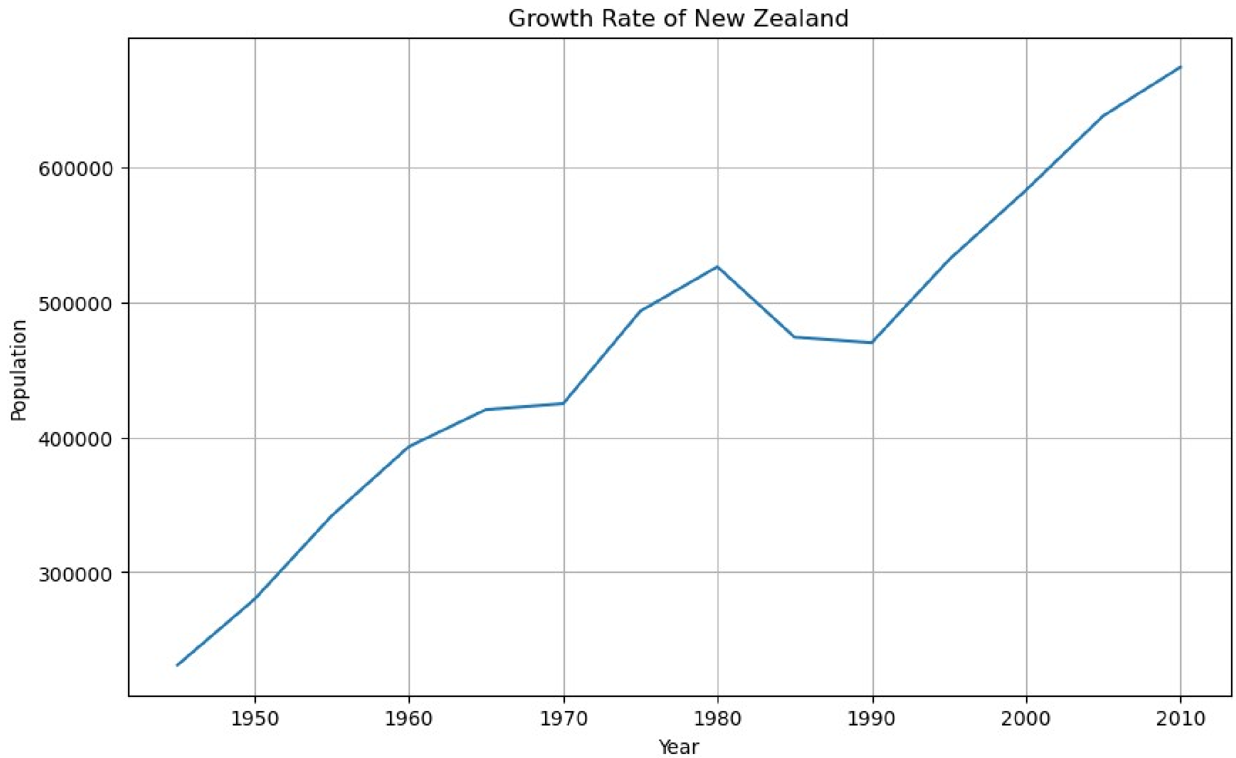

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



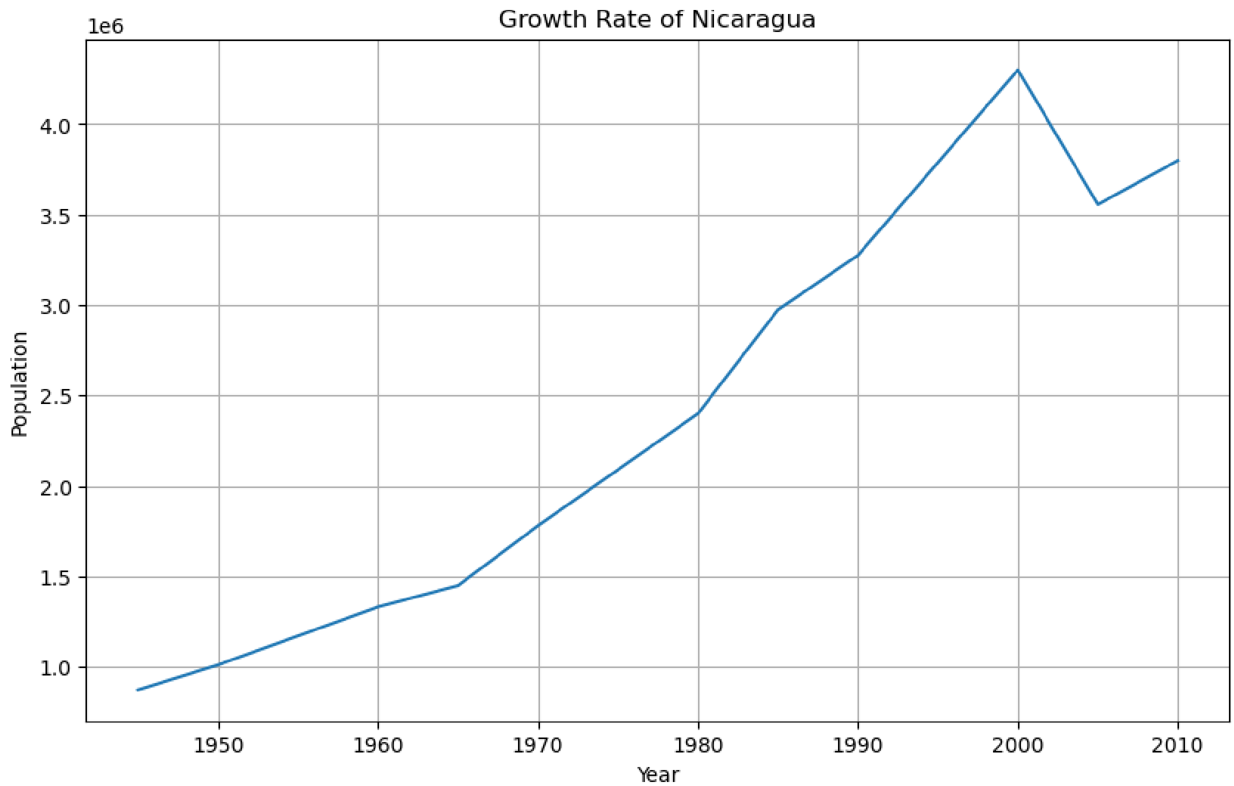

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



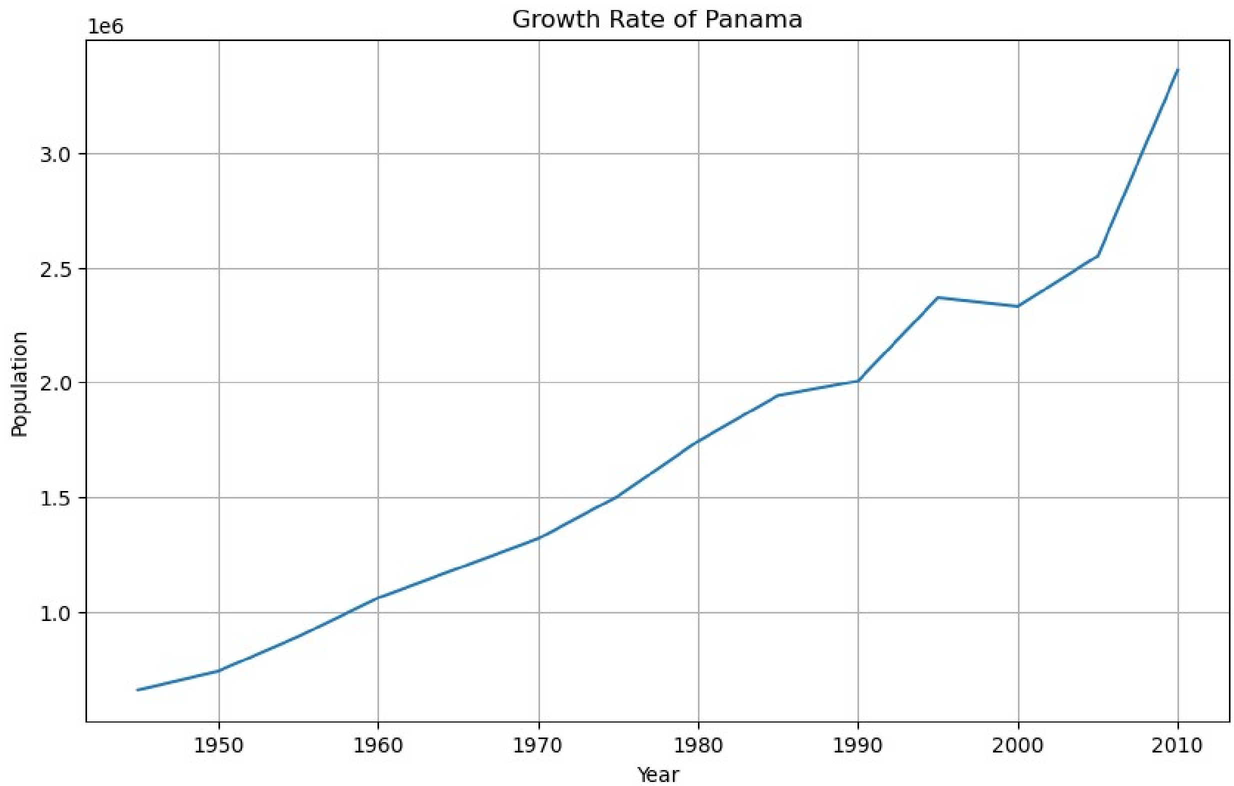

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



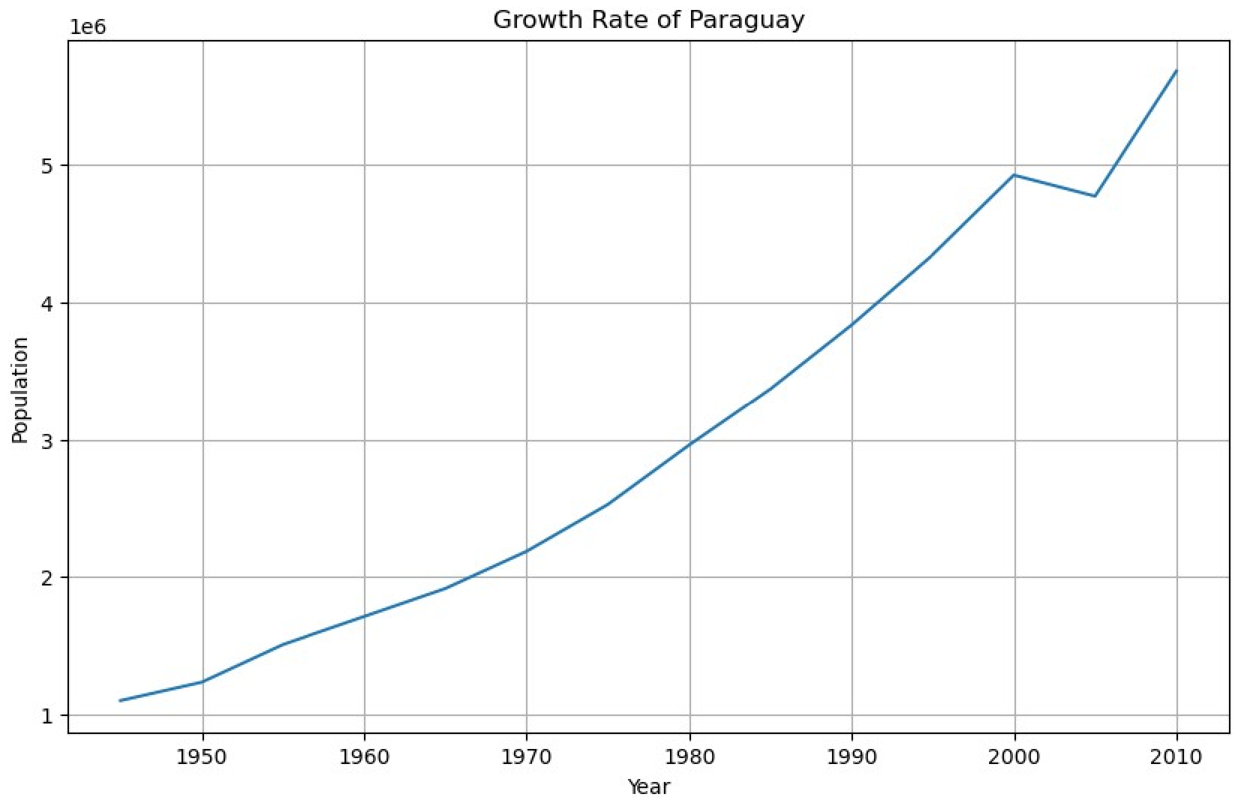

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



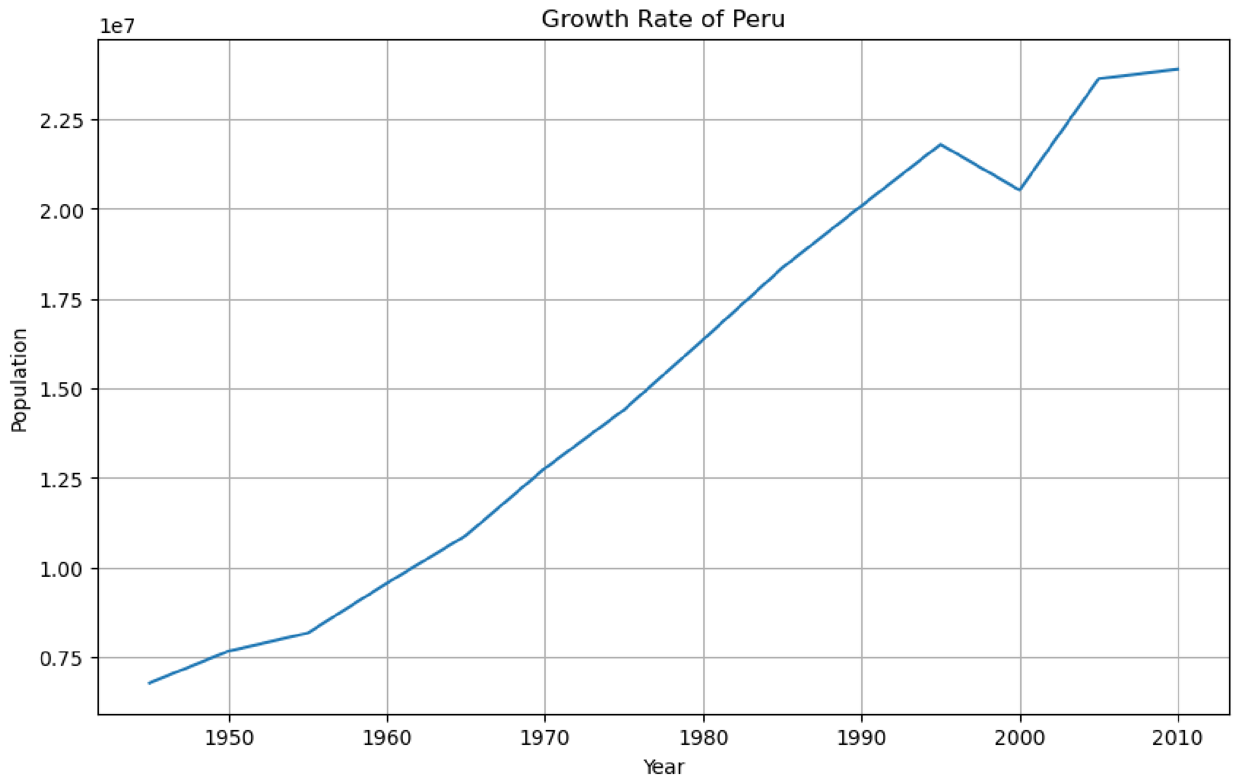

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



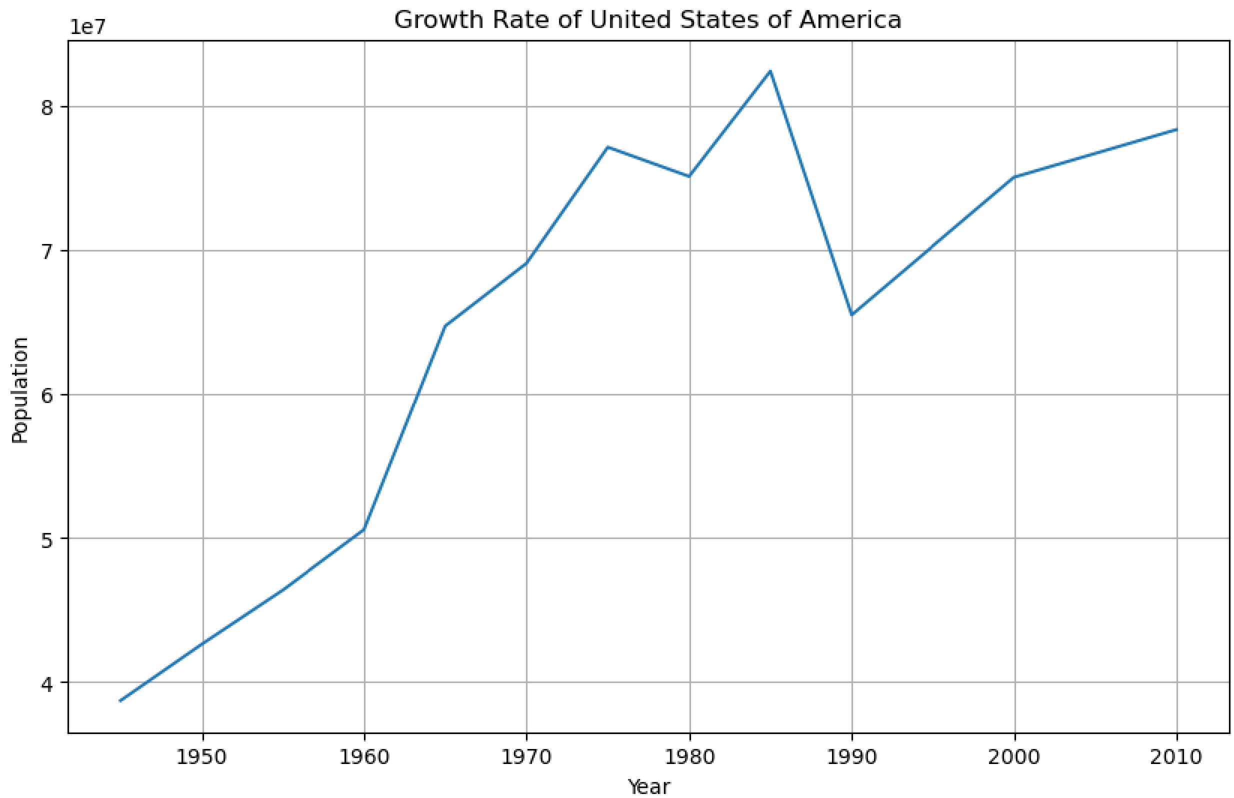

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



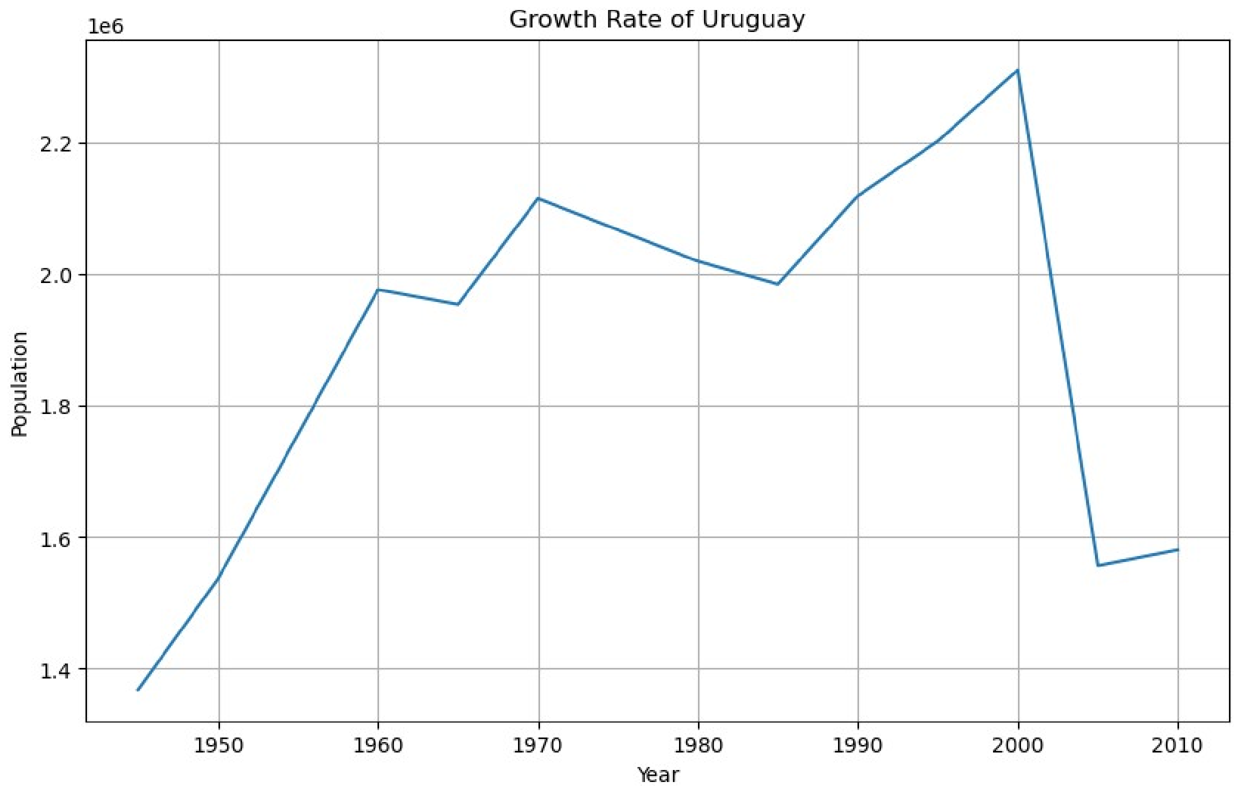

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



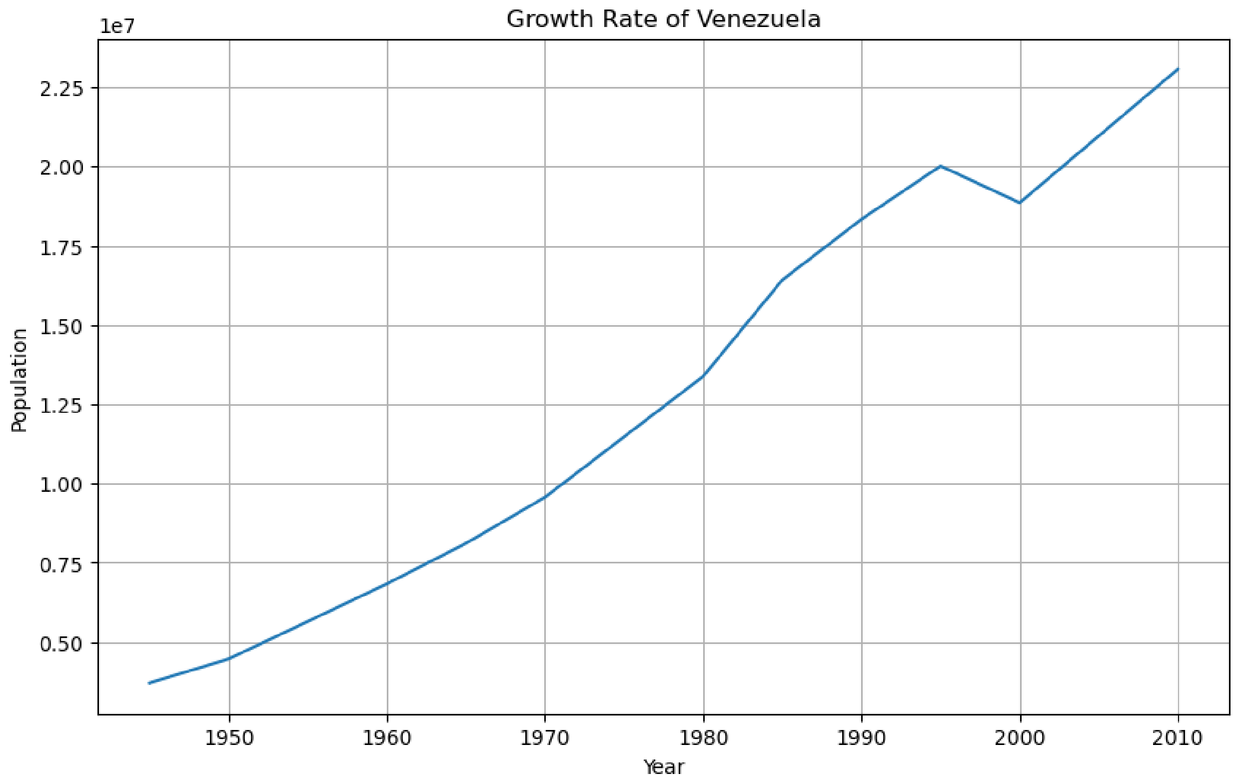

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```




```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```




```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



4.2 Growth rates in a Random Countries

A function is then used to randomly select one country from each region. The selected countries are printed and used for further analysis. Another function calculates the annual growth rate for each selected country by computing the percentage change year-over-year, excluding any NaN values. Finally, the growth rates of the selected countries are plotted on a single graph, showing a comparison of their growth rates over time.

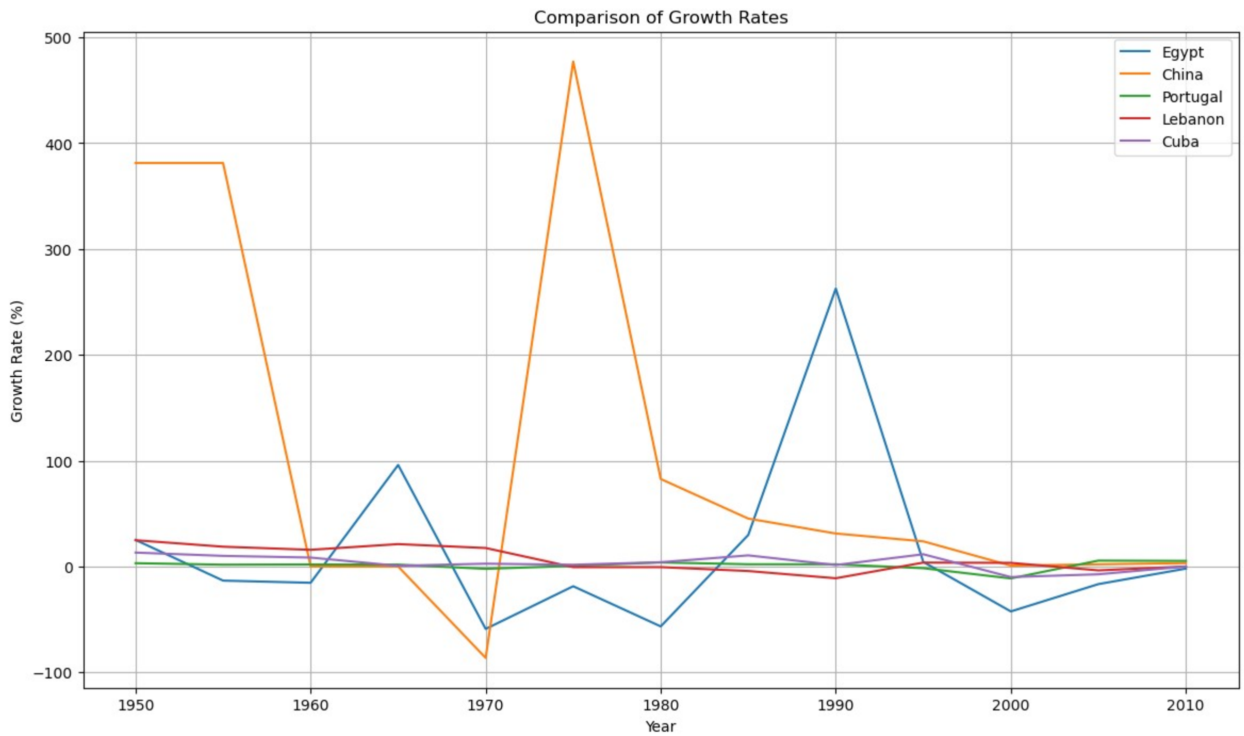
```
# Define regions and countries
regions = {
    'Africa': ['Ethiopia', 'Egypt'],
    'Asia': ['China', 'India', 'Iran', 'Iraq', 'Japan','Thailand',
'Turkey'],
    'Europe': ['Albania', 'Belgium', 'Bulgaria', 'Denmark', 'Finland',
'France', 'Greece', 'Hungary', 'Iceland', 'Ireland', 'Italy',
'Luxembourg', 'Norway', 'Poland', 'Portugal', 'Romania', 'Russia',
'Spain', 'Sweden', 'Switzerland', 'United Kingdom'],
    'Middle East': ['Jordan', 'Lebanon', 'Saudi Arabia'],
    'Western Hemisphere': ['Argentina', 'Australia', 'Bolivia',
'Brazil', 'Canada', 'Chile', 'Colombia', 'Costa Rica', 'Cuba',
'Dominican Republic', 'Ecuador', 'El Salvador', 'Guatemala', 'Haiti',
'Honduras', 'Mexico', 'New Zealand', 'Nicaragua', 'Panama',
'Paraguay', 'Peru', 'United States of America', 'Uruguay',
'Venezuela']
}
```

```
# Function to select a random country from each region
def select_random_country(region):
    if region in regions:
        return random.choice(regions[region])
    else:
        return None

# Select one random country from each region
selected_countries = [select_random_country(region) for region in
regions.keys()]
selected_countries = [country for country in selected_countries if
country is not None]  # Remove any None values

print("Selected countries:", selected_countries)

# Function to calculate growth rate
def calculate_growth_rate(country):
    if country in df.columns:
        df_country = df[['Year', country]].copy()
        df_country['Growth_Rate'] = df_country[country].pct_change() *
100
        return df_country.dropna()
    else:
        return None

# Plot growth rates of selected countries
plt.figure(figsize=(14, 8))

for country in selected_countries:
    df_country = calculate_growth_rate(country)
    if df_country is not None:
        sns.lineplot(x=df_country['Year'],
y=df_country['Growth_Rate'], label=country)

plt.title('Comparison of Growth Rates')
plt.xlabel('Year')
plt.ylabel('Growth Rate (%)')
plt.legend()
plt.grid(True)
plt.show()
```

```
Selected countries: ['Egypt', 'China', 'Portugal', 'Lebanon', 'Cuba']
```

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
```

```
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



From the analysis that i have concluded with regards to the catholic demographic population numbers these are the findings:

- Ethiopia and Egypt: Both countries have seen significant population growth.Ethiopia, starting with a smaller number has shown rapid increases, while Eygpt has also seen substantial growth, reflecting broader regional trends.
- China: China's population saw a rapid increase post-1950, particularly during the mid-20th century, despite later policies that was aimed in controlling birth.
- Iran: Iran's population data, although incomplete in the early years, has shown a growing trend aligning with regional growth patterns.
- Latin American Countries (e.g. Panama, Paraguay, Peru, Venezula), have shown consistent population growth driven by the higher birth rates and improvements in health and living standards.
- Uruguay : Uruguay has shown a steady but slower growth rate compared to other Latin American countries

Projected Growth Trends:

- High Growth Areas: Parts of Asia and Africa are projected to continue ecperiencing high growth rates due to higher fertility rates and the younger populations.
- Moderate Growth Areas: The Middle East and Latin America are expected to see moderate growth, driven by improving economic conditions and healthcare.

- Low Growth/Stagnant Areas: Europe and also certain high-income countries like the United states may see slower growth rates or even stagnation due to the aging populations and lower birth rates.

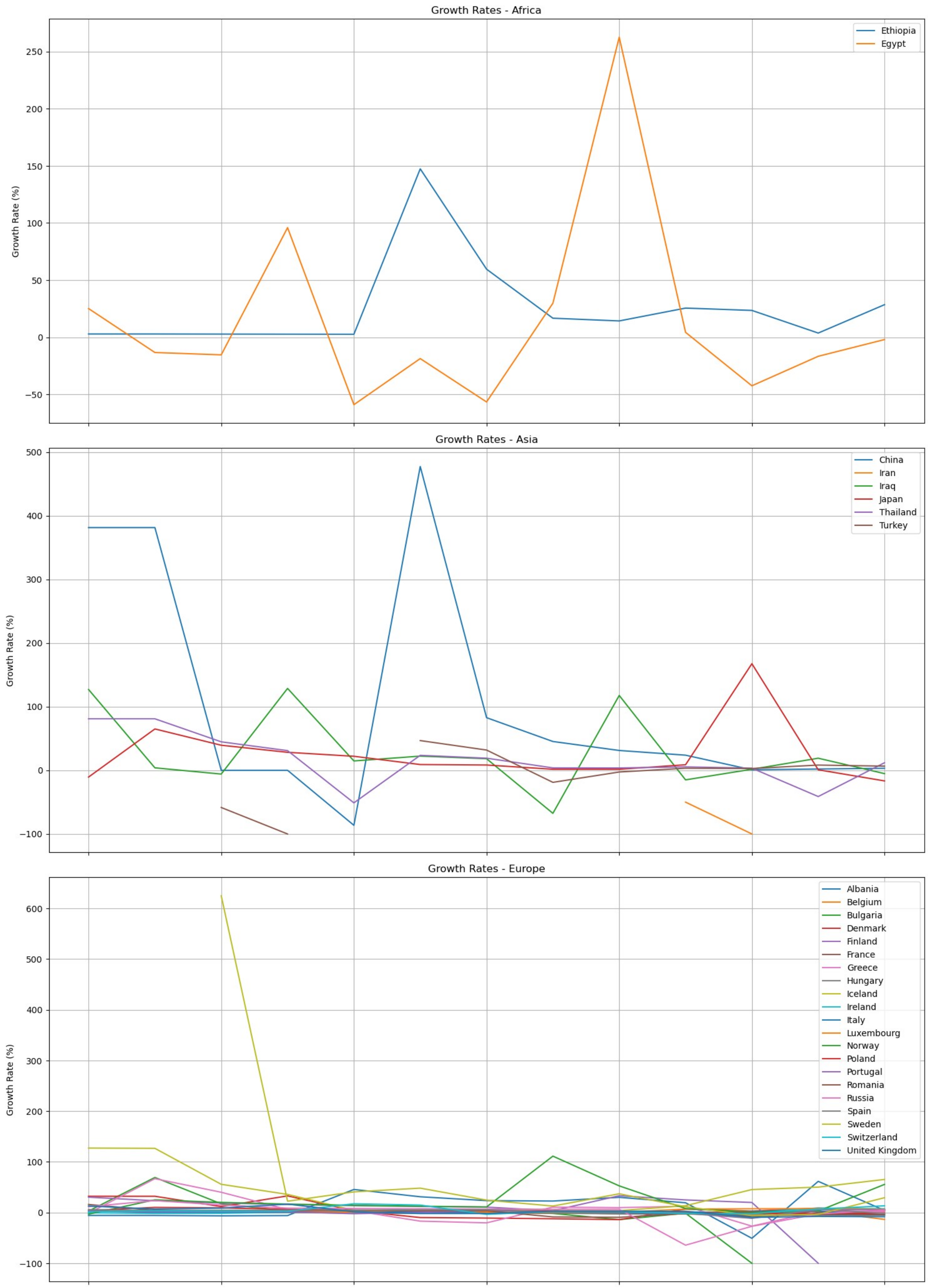
In a nutshell, the data indicates the diverse growth trajectories across the different regions, reflecting varying demographic economic , political and social factors influencing population trends.

4.3 Country representation through their Continents

We are defining a dictionary that categorizes various countries into their respective regions. A function, calculate_growth_rate, is defined to compute the annual growth rate of data for each country by calculating the percentage change year-over-year and return this data, excluding any NaN values resulting from the calculation. The script proceeds to create a subplot for each region, plotting the growth rates of all countries within each region on separate graphs. The graphs are titled by region, labeled with the growth rate on the y-axis, and include legends for clarity.

```
# Load data from CSV into a DataFrame
df = pd.read_csv('main_catholic.csv')
df.set_index('Year', inplace=True)  # Assuming 'Year' is the index
column

# Define regions and countries (based on the given regions)
regions = {
    'Africa': ['Ethiopia', 'Egypt'],
    'Asia': ['China', 'India', 'Iran', 'Iraq', 'Japan', 'Thailand',
'Turkey'],
    'Europe': ['Albania', 'Belgium', 'Bulgaria', 'Denmark', 'Finland',
'France', 'Greece', 'Hungary', 'Iceland', 'Ireland', 'Italy',
'Luxembourg', 'Norway', 'Poland', 'Portugal', 'Romania', 'Russia',
'Spain', 'Sweden', 'Switzerland', 'United Kingdom'],
    'Middle East': ['Jordan', 'Lebanon', 'Saudi Arabia'],
    'Western Hemisphere': ['Argentina', 'Australia', 'Bolivia',
'Brazil', 'Canada', 'Chile', 'Colombia', 'Costa Rica', 'Cuba',
'Dominican Republic', 'Ecuador', 'El Salvador', 'Guatemala', 'Haiti',
'Honduras', 'Mexico', 'New Zealand', 'Nicaragua', 'Panama',
'Paraguay', 'Peru', 'United States of America', 'Uruguay',
'Venezuela']
}

# Function to calculate growth rate for a country
def calculate_growth_rate(country):
    if country in df.columns:
        df_country = df[[country]].copy()
        df_country['Growth_Rate'] = df_country[country].pct_change() *
100
        return df_country.dropna()
    else:
```

```
        return None

# Plot growth rates of all countries in each region in separate graphs
fig, axs = plt.subplots(len(regions), 1, figsize=(15, 35),
sharex=True)  # Adjusted figsize

for i, (region, countries) in enumerate(regions.items()):
    for country in countries:
        df_country = calculate_growth_rate(country)
        if df_country is not None:
            axs[i].plot(df_country.index, df_country['Growth_Rate'],
label=country)
    axs[i].set_title(f'Growth Rates - {region}')
    axs[i].set_ylabel('Growth Rate (%)')
    axs[i].legend()
    axs[i].grid(True)

plt.xlabel('Year')
plt.tight_layout()
plt.show()
```



Growth Rates - Africa
Growth Rate (%)
Ethiopia
Egypt
250
200
150
100
50
0
−50
Growth Rates - Asia
Growth Rate (%)
China
Iran
Iraq
Japan
Thailand
Turkey
500
400
300
200
100
0
−100
Growth Rates - Europe
Growth Rate (%)
Albania
Belgium
Bulgaria
Denmark
Finland
France
Greece
Hungary
Iceland
Ireland
Italy
Luxembourg
Norway
Poland
Portugal
Romania
Russia
Spain
Sweden
Switzerland
United Kingdom
600
500
400
300
200
100
0
−100

From the Country Representation we can understand the changes and how they relate to the issues that these places are facing or went through that resulted in the catholic population numbers changing :

Africa - Africa has seen a remarkable Catholic population surge since 1945.Beginning with modest numbers, the continent's population has grown dramatically due to high fertility rates and improved healthcare and living conditions. This growth represents one of the highest regional population expansions globally.

Asia - Asia Catholic population numbers have shown a steady upward trend. Countries like China have experienced substantial growth, particularly after the 1950s, driven by economic reforms, improved living standards and advancements in healthcare. This therefore underscores Asia's significant demographic transformation over the decades

Europe - Europe's Catholic population growth has been relatively stable but slower than other regions. Following World War II, the continent experienced modest growth rates, which have stagnated in recent decades due to lower birth rates, increased urbanization, and an aging population. This trend reflects a shift towards an older demographic.

Middle East - Middle East has experienced a moderate to high population growth. Political stability in certain areas, combined with higher birth rates and improved access to healthcare and education, has significantly contributed to this trend, leading to a notable demographic changes in the region over the past few decades.

Western Hemisphere - Has seen a mixture in the Catholic population increase with some countries having a increase and some countries having a decrease in population numbers and this could be resulted due to the region's demographic dynamism.

5. DISCUSSIONS:

Continental Shifts:

Africa and Asia have experienced significant growth in their Catholic populations due to both natural population growth and missionary activities. Europe’s Catholic population has remained stable but shows signs of aging and stagnation. The Western Hemisphere continues to maintain strong Catholic populations, particularly in Latin America.

Country-Specific Trends:

Ethiopia and Egypt show modest Catholic population growth. China has seen a significant increase in its Catholic population, likely due to the relaxation of religious restrictions. Brazil and Argentina have experienced substantial growth, reflecting their cultural and historical ties to Catholicism.

Global Trends:

The overall global Catholic population has increased steadily over the years. There has been a shift in the Catholic demographic distribution from Europe to Africa and Asia.

Socio-Economic Influences:

Economic growth, urbanization, and migration have significantly influenced Catholic demographics, especially in developing regions. Socio-economic factors like urbanization and economic migration continue to shape the demographic trends in Latin America.

Future Outlook:

Continued growth in the Catholic population is expected in Africa and Asia due to higher birth rates and ongoing evangelization efforts. Adaptive strategies are required to maintain the Catholic faith's influence in increasingly secular and economically diverse societies.

6. CONCLUSION

In Conclusion -

Changes in Catholic Population Demographics over 70 Years Over the past 70 years, the Catholic population has shifted from being predominantly in Europe and the Western Hemisphere to seeing significant growth in Africa and Asia. While Europe has experienced stagnation or decline due to the secularization and lower birth rates, Africa and Asia have seen rapid increases due to higher birth rates, improved healthcare and active missionary efforts.

Significant Growth and Decline in Catholic Adherents Africa and Asia have experienced the most significant growth in Catholic adherents, with countries like Nigeria, Ethiopia, and the Philippines showing substantial increases. Conversely, Europe has seen a decline or stagnation in its Catholic population, reflecting broader secularization trends. The Western Hemisphere remains a major center of Catholicism but has experienced slower growth compared to the dynamic increase in Africa and Asia

7. REFERENCES

[1]Iberdrola. (n.d.). World population evolution. Retrieved from https://www.iberdrola.com/sustainability/world-population-evolution

[2] Lu, A. (2018). Factors contributing to Catholic population growth and decline. Res Publica - Journal of Undergraduate Research, 23(1). Retrieved from https://digitalcommons.iwu.edu/cgi/viewcontent.cgi?article=1039&context=respublica

[3] Kaggle. (n.d.). Roman Catholic analysis: World population data. Retrieved from https://www.kaggle.com/code/donyoe/roman-catholic-analysis/input?select=global.csv

[4] Kaggle. (n.d.). Roman Catholic analysis: Country data. Retrieved from https://www.kaggle.com/code/donyoe/roman-catholic-analysis/input?select=national.csv

[5] Kaggle. (n.d.). World religions: Continental data. Retrieved from https://www.kaggle.com/code/gulsahdemiryurek/world-religions/input?select=regional.csv