

```

In [24]: import pandas as pd
import matplotlib.pyplot as plt

# CSV dosyasından veriyi okuma
df = pd.read_csv('country_birth_rate.csv')

# Z-Skoru hesaplama
mean = df['1000 Kişi Başına Canlı Doğumlar'].mean()
std = df['1000 Kişi Başına Canlı Doğumlar'].std()
df['z_score'] = (df['1000 Kişi Başına Canlı Doğumlar'] - mean) / std

# Z-Skoru 3'ten büyük olan aykırı değerleri çıkarma
df_no_outliers = df[df['z_score'].abs() <= 3]

# Veriyi sıralama
df_sorted = df_no_outliers.sort_values(by="1000 Kişi Başına Canlı Doğumlar", ascending=False)

# Kategorilere ayırma
category_1 = df_sorted[df_sorted["1000 Kişi Başına Canlı Doğumlar"] >= 40]
category_2 = df_sorted[(df_sorted["1000 Kişi Başına Canlı Doğumlar"] >= 35) & (df_sorted["1000 Kişi Başına Canlı Doğumlar"] < 40)]
category_3 = df_sorted[(df_sorted["1000 Kişi Başına Canlı Doğumlar"] >= 30) & (df_sorted["1000 Kişi Başına Canlı Doğumlar"] < 35)]
category_4 = df_sorted[df_sorted["1000 Kişi Başına Canlı Doğumlar"] < 30]

# Box plot verilerini hazırlama
data_to_plot = [
    category_1["1000 Kişi Başına Canlı Doğumlar"],
    category_2["1000 Kişi Başına Canlı Doğumlar"],
    category_3["1000 Kişi Başına Canlı Doğumlar"],
    category_4["1000 Kişi Başına Canlı Doğumlar"]
]

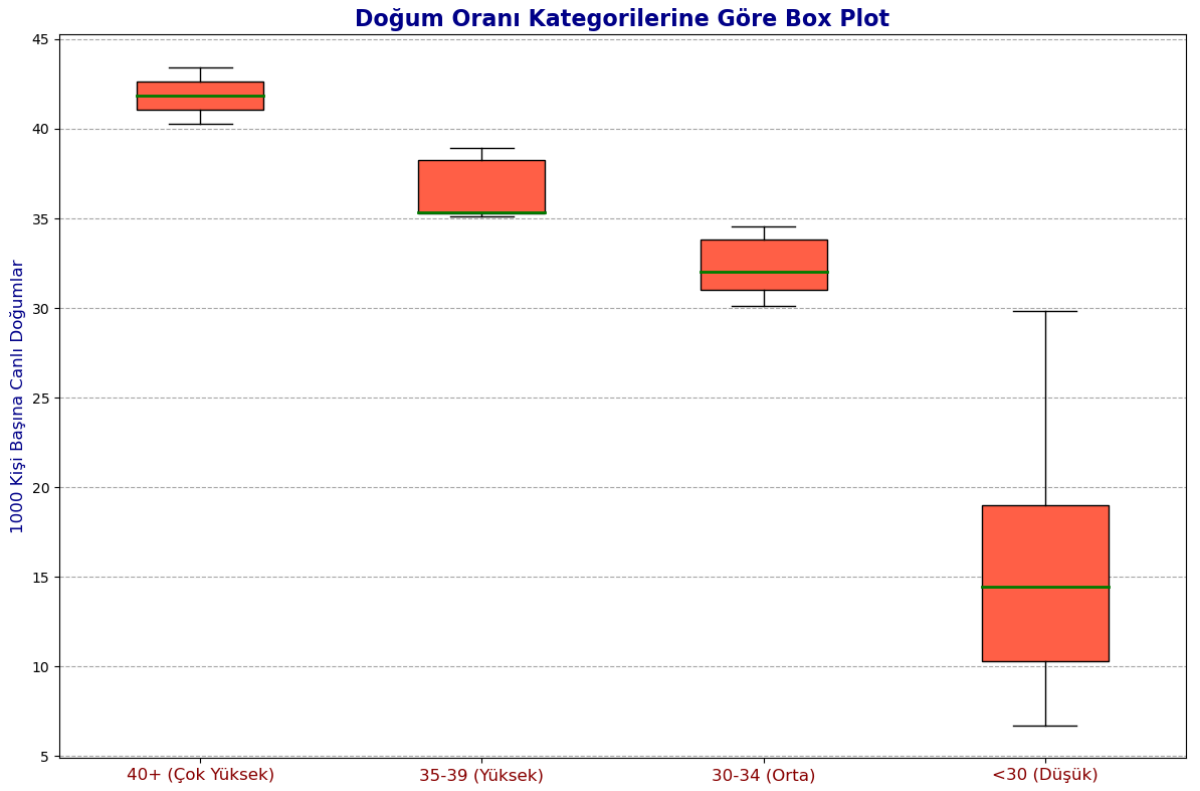
# Grafik oluşturma
plt.figure(figsize=(12, 8))
plt.boxplot(data_to_plot, patch_artist=True,
            boxprops=dict(facecolor='#FF6347', color='black'), # Canlı kırmızı tor
            whiskerprops=dict(color='black'),
            capprops=dict(color='black'),
            medianprops=dict(color='green', linewidth=2), # Medyan çizgisi yeşil
            flierprops=dict(markerfacecolor='yellow', marker='o', markersize=6, linestyle='none'))

# Başlık ve etiketler
plt.title('Doğum Oranı Kategorilerine Göre Box Plot', fontsize=16, fontweight='bold')
plt.xticks([1, 2, 3, 4], ['40+ (Çok Yüksek)', '35-39 (Yüksek)', '30-34 (Orta)', '<30 (Düşük)'])
plt.ylabel('1000 Kişi Başına Canlı Doğumlar', fontsize=12, color='darkblue')
plt.grid(axis='y', linestyle='--', alpha=0.7, color='gray')

# Grafik kaydetme
plt.savefig('C:/Users/Lenovo/Documents/dogum_orani_box_plot_no_outliers.png', dpi=300)

# Gösterim
plt.tight_layout()
plt.show()

```



```
In [33]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Manuel veri girişi
data = [
    ["Finland", 100.00, 5545475],
    ["Norway", 100.00, 5474360],
    ["Luxembourg", 100.00, 654768],
    ["Andorra", 100.00, 80088],
    ["Greenland", 100.00, 56643],
    ["Liechtenstein", 100.00, 39584],
    ["Uzbekistan", 100.00, 35163944],
    ["Latvia", 99.89, 1830211],
    ["Estonia", 99.82, 1322765],
    ["Lithuania", 99.82, 2718352],
    ["Azerbaijan", 99.81, 10412651],
    ["Guam", 99.79, 172952],
    ["Kazakhstan", 99.79, 19606633],
    ["Poland", 99.79, 41026067],
    ["Tajikistan", 99.78, 10143543],
    ["Armenia", 99.77, 2777970],
    ["Ukraine", 99.76, 36744634],
    ["Georgia", 99.76, 3728282],
    ["Belarus", 99.72, 9498238],
    ["Russia", 99.72, 144444359],
    ["Slovenia", 99.71, 2119675],
    ["Cuba", 99.71, 11194449],
    ["Barbados", 99.70, 281995],
    ["Turkmenistan", 99.69, 6516100],
    ["Slovakia", 99.60, 5795199],
    ["Palau", 99.52, 18058],
    ["Kyrgyzstan", 99.50, 6735347],
    ["Tonga", 99.40, 107773],
    ["Hungary", 99.38, 10156239],
    ["Maldives", 99.32, 521021],
    ["Croatia", 99.27, 4008617],
    ["Moldova", 99.24, 3435931],
```

```

["Cyprus", 99.06, 1260138],
["Italy", 99.02, 58870762],
["Samoa", 99.02, 225681],
["United States", 99.00, 339996563],
["Japan", 99.00, 123294513],
["Germany", 99.00, 83294633],
["United Kingdom", 99.00, 67736802],
["France", 99.00, 64756584],
["Canada", 99.00, 38781291],
["Australia", 99.00, 26439111],
["Netherlands", 99.00, 17618299],
["Belgium", 99.00, 11686140],
["Sweden", 99.00, 10612086],
["Switzerland", 99.00, 8796669],
["Denmark", 99.00, 5910913],
["New Zealand", 99.00, 5228100],
["Ireland", 99.00, 5056935],
["Iceland", 99.00, 375318],
["Monaco", 99.00, 36297],
["Trinidad And Tobago", 98.97, 1534937],
["Antigua And Barbuda", 98.95, 94298],
["Cayman Islands", 98.87, 69310] # Düzeltilmiş sayı
]

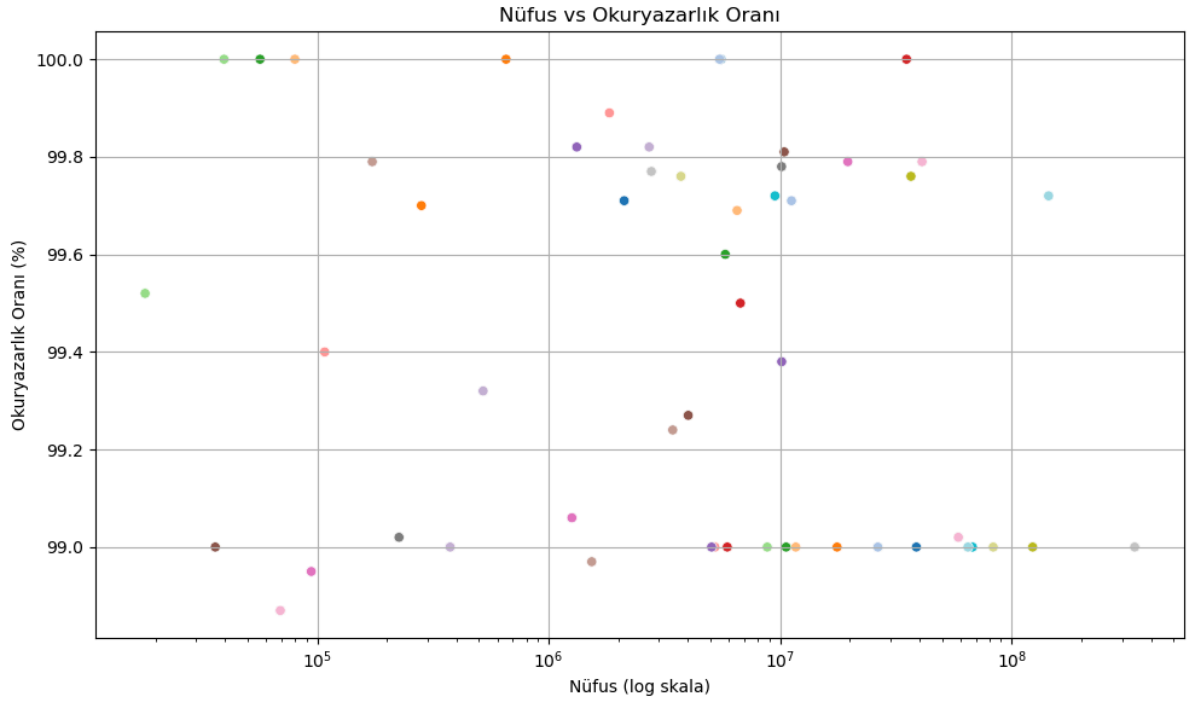
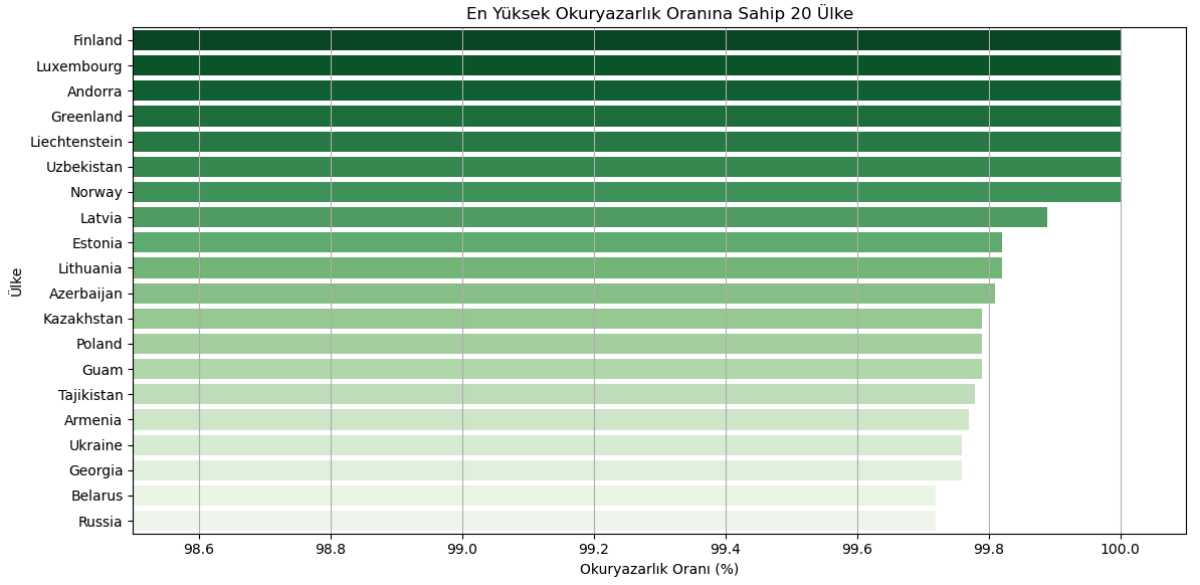
# DataFrame oluştur
df = pd.DataFrame(data, columns=["Country", "Literacy Rate", "Population"])

# --- Grafik 1: En Yüksek Okuryazarlık Oranı İlk 20 Ülke ---
top20 = df.sort_values(by="Literacy Rate", ascending=False).head(20)

plt.figure(figsize=(12, 6))
sns.barplot(data=top20, x="Literacy Rate", y="Country", palette="Greens_r")
plt.title("En Yüksek Okuryazarlık Oranına Sahip 20 Ülke")
plt.xlabel("Okuryazarlık Oranı (%)")
plt.ylabel("Ülke")
plt.xlim(98.5, 100.1)
plt.grid(True, axis="x")
plt.tight_layout()
plt.show()

# --- Grafik 2: Nüfus vs Okuryazarlık Oranı (Log Skala) ---
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x="Population", y="Literacy Rate", hue="Country", palette=
plt.xscale("log")
plt.title("Nüfus vs Okuryazarlık Oranı")
plt.xlabel("Nüfus (log skala)")
plt.ylabel("Okuryazarlık Oranı (%)")
plt.grid(True)
plt.tight_layout()
plt.show()

```



```
In [34]: import pandas as pd
import matplotlib.pyplot as plt

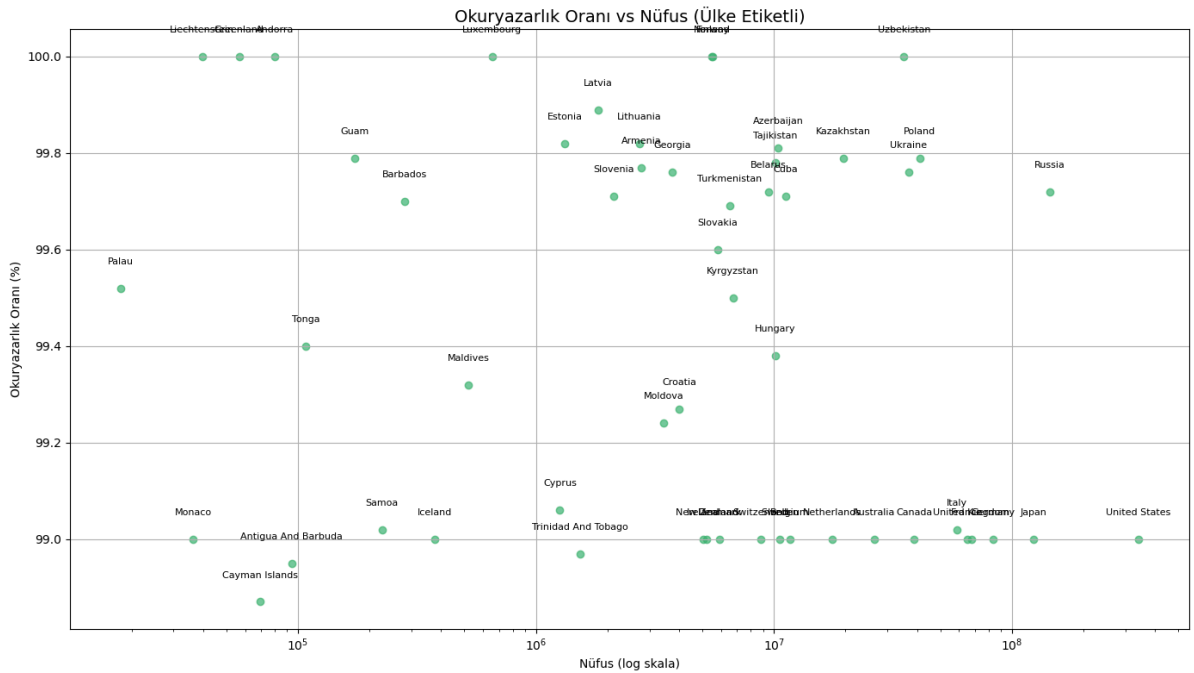
# Veri çerçevesi
df = pd.DataFrame(data, columns=["Ülke", "Okuryazarlık Oranı", "Nüfus"])

# Grafik oluşturma
plt.figure(figsize=(14, 8))
plt.scatter(df["Nüfus"], df["Okuryazarlık Oranı"], color="mediumseagreen", alpha=0.5)

# Her noktanın üzerine ülke adını yazma
for i in range(len(df)):
    plt.text(
        df["Nüfus"][i],
        df["Okuryazarlık Oranı"][i] + 0.05, # Yazıyı biraz yukarı taşı
        df["Ülke"][i],
        fontsize=8,
        ha='center'
    )

# Log skala (nüfus çok geniş aralıkta olduğu için)
plt.xscale("log")
```

```
# Başlık ve eksenler
plt.title("Okuryazarlık Oranı vs Nüfus (Ülke Etiketli)", fontsize=14)
plt.xlabel("Nüfus (log skala)")
plt.ylabel("Okuryazarlık Oranı (%)")
plt.grid(True)
plt.tight_layout()
plt.show()
```



```
In [2]: import matplotlib.pyplot as plt
import pandas as pd
import matplotlib.ticker as ticker

# Veriyi yükleyin
file_path = r"C:\Users\Lenovo\Downloads\ada ülkeler.xlsx"
df = pd.read_excel(file_path)

# İlk 50 ülkeyi seçelim
df = df.head(50)

# Grafik oluşturmak için
fig, ax = plt.subplots(figsize=(12, 8))

# Çubuk grafik çizimi
bar_width = 0.35
index = range(len(df))

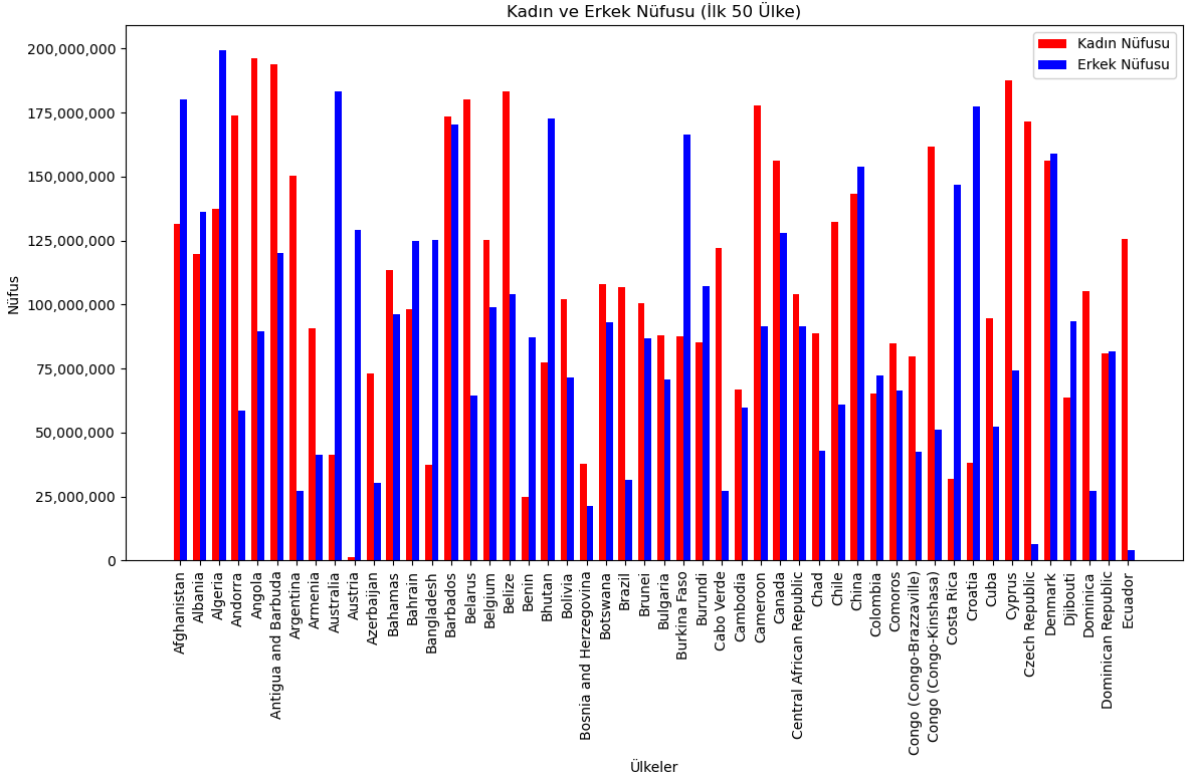
bar1 = ax.bar(index, df['Female Population'], bar_width, label='Kadın Nüfusu', color='blue')
bar2 = ax.bar([i + bar_width for i in index], df['Male Population'], bar_width, label='Erkek Nüfusu', color='red')

# Grafik ayarları
ax.set_xlabel('Ülkeler')
ax.set_ylabel('Nüfus')
ax.set_title('Kadın ve Erkek Nüfusu (İlk 50 Ülke)')
ax.set_xticks([i + bar_width / 2 for i in index])
ax.set_xticklabels(df['Country'], rotation=90)
ax.legend()

# Y eksenini daha anlaşılır yapmak için büyük sayıları formatlayalım
ax.yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, _: f'{int(x):,}'))

# Grafik gösterimi
```

```
plt.tight_layout()
plt.show()
```



```
In [3]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# CSV dosyasından veriyi okuyoruz
df = pd.read_csv('C:/Users/Lenovo/Documents/gsyih_verileri.csv')

# Ülkeleri index olarak ayarlıyoruz
df.set_index('Ülke', inplace=True)

# Grafik için bir figure ayarlıyoruz
fig, ax = plt.subplots(figsize=(14, 8))

# Üç yılın çubuklarını yanyana yerleştirebilmek için bar_width ve index ayarları yapılıyor
bar_width = 0.25
index = np.arange(len(df))

# Her yıl için çubukları çiziyoruz
ax.bar(index, df['2025'], bar_width, label='2025')
ax.bar(index + bar_width, df['2023'], bar_width, label='2023')
ax.bar(index + 2 * bar_width, df['2022'], bar_width, label='2022')

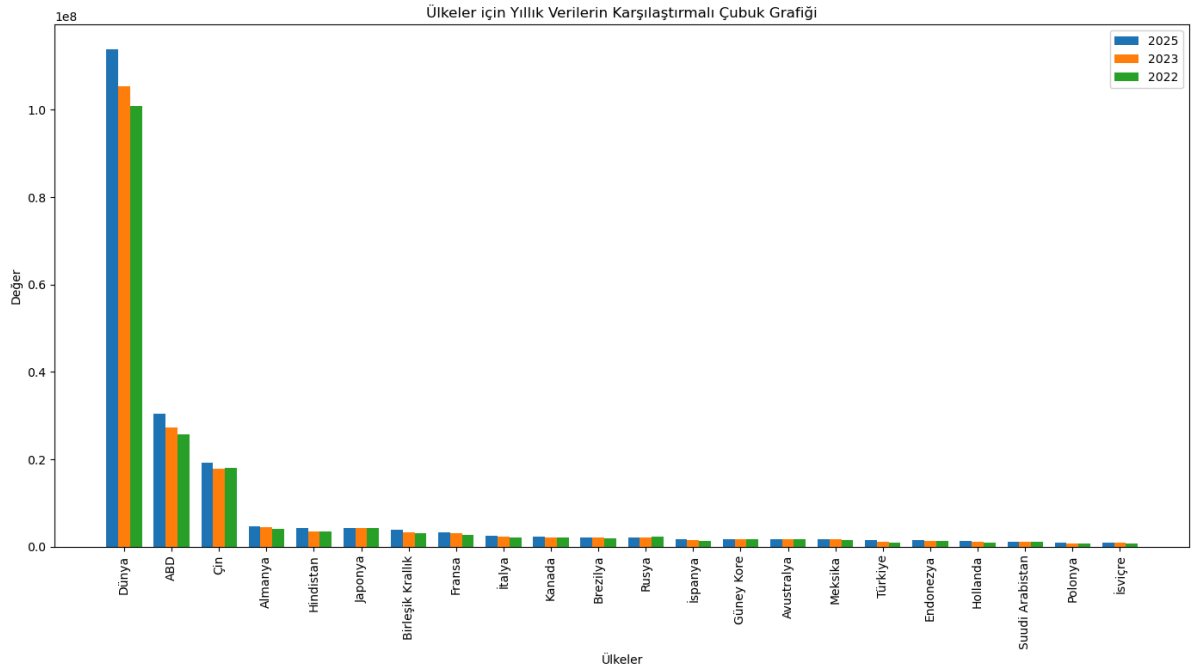
# Başlık ve etiketler
ax.set_title('Ülkeler için Yıllık Verilerin Karşılaştırmalı Çubuk Grafiği')
ax.set_xlabel('Ülkeler')
ax.set_ylabel('Değer')
ax.set_xticks(index + bar_width)
ax.set_xticklabels(df.index, rotation=90) # Ülkeleri yanyana sıralıyoruz

# Legend ekliyoruz
ax.legend()

# Dosyayı Görsel (PNG) olarak kaydediyoruz
plt.savefig('C:/Users/Lenovo/Documents/gsyih_comparison_bar_chart.png', dpi=300)

# Grafiği gösteriyoruz
```

```
plt.tight_layout()
plt.show()
```



```
In [8]: print("GSYİH verisindeki ülkeler:")
print(gdp_df['Country'].unique())
```

GSYİH verisindeki ülkeler:

```
['dünya' 'abd' 'çin' 'almanya' 'hindistan' 'japonya' 'birleşik krallık'
 'fransa' 'italya' 'kanada' 'brezilya' 'rusya' 'ispanya' 'güney kore'
 'avustralya' 'meksika' 'türkiye' 'endonezya' 'hollanda' 'suudi arabistan'
 'polonya' 'isviçre']
```

```
In [9]: print("Yaşam beklentisi verisindeki ülkeler:")
print(life_df['Country'].unique())
```

Yaşam beklentisi verisindeki ülkeler:

['aruba' 'africa eastern and southern' 'afghanistan'
'africa western and central' 'angola' 'albania' 'andorra' 'arab world'
'united arab emirates' 'argentina' 'armenia' 'american samoa'
'antigua and barbuda' 'australia' 'austria' 'azerbaijan' 'burundi'
'belgium' 'benin' 'burkina faso' 'bangladesh' 'bulgaria' 'bahrain'
'bahamas, the' 'bosnia and herzegovina' 'belarus' 'belize' 'bermuda'
'bolivia' 'brazil' 'barbados' 'brunei darussalam' 'bhutan' 'botswana'
'central african republic' 'canada' 'central europe and the baltics'
'switzerland' 'channel islands' 'chile' 'china' 'cote d'ivoire'
'cameroon' 'congo, dem. rep.' 'congo, rep.' 'colombia' 'comoros'
'cabo verde' 'costa rica' 'caribbean small states' 'cuba' 'curacao'
'cayman islands' 'cyprus' 'czechia' 'germany' 'djibouti' 'dominica'
'denmark' 'dominican republic' 'algeria'
'east asia & pacific (excluding high income)'
'early-demographic dividend' 'east asia & pacific'
'europe & central asia (excluding high income)' 'europe & central asia'
'ecuador' 'egypt, arab rep.' 'euro area' 'eritrea' 'spain' 'estonia'
'ethiopia' 'european union' 'fragile and conflict affected situations'
'finland' 'fiji' 'france' 'faroe islands' 'micronesia, fed. sts.' 'gabon'
'united kingdom' 'georgia' 'ghana' 'gibraltar' 'guinea' 'gambia, the'
'guinea-bissau' 'equatorial guinea' 'greece' 'grenada' 'greenland'
'guatemala' 'guam' 'guyana' 'high income' 'hong kong sar, china'
'honduras' 'heavily indebted poor countries (hipc)' 'croatia' 'haiti'
'hungary' 'ibrd only' 'ida & ibrd total' 'ida total' 'ida blend'
'indonesia' 'ida only' 'isle of man' 'india' 'not classified' 'ireland'
'iran, islamic rep.' 'iraq' 'iceland' 'israel' 'italy' 'jamaica' 'jordan'
'japan' 'kazakhstan' 'kenya' 'kyrgyz republic' 'cambodia' 'kiribati'
'st. kitts and nevis' 'korea, rep.' 'kuwait'
'latin america & caribbean (excluding high income)' 'lao pdr' 'lebanon'
'liberia' 'libya' 'st. lucia' 'latin america & caribbean'
'least developed countries: un classification' 'low income'
'liechtenstein' 'sri lanka' 'lower middle income' 'low & middle income'
'lesotho' 'late-demographic dividend' 'lithuania' 'luxembourg' 'latvia'
'macao sar, china' 'st. martin (french part)' 'morocco' 'monaco'
'moldova' 'madagascar' 'maldives' 'middle east & north africa' 'mexico'
'marshall islands' 'middle income' 'north macedonia' 'mali' 'malta'
'myanmar' 'middle east & north africa (excluding high income)'
'montenegro' 'mongolia' 'northern mariana islands' 'mozambique'
'mauritania' 'mauritius' 'malawi' 'malaysia' 'north america' 'namibia'
'new caledonia' 'niger' 'nigeria' 'nicaragua' 'netherlands' 'norway'
'nepal' 'nauru' 'new zealand' 'oecd members' 'oman' 'other small states'
'pakistan' 'panama' 'peru' 'philippines' 'palau' 'papua new guinea'
'poland' 'pre-demographic dividend' 'puerto rico'
'korea, dem. people's rep.' 'portugal' 'paraguay' 'west bank and gaza'
'pacific island small states' 'post-demographic dividend'
'french polynesia' 'qatar' 'romania' 'russian federation' 'rwanda'
'south asia' 'saudi arabia' 'sudan' 'senegal' 'singapore'
'solomon islands' 'sierra leone' 'el salvador' 'san marino' 'somalia'
'serbia' 'sub-saharan africa (excluding high income)' 'south sudan'
'sub-saharan africa' 'small states' 'sao tome and principe' 'suriname'
'slovak republic' 'slovenia' 'sweden' 'eswatini'
'sint maarten (dutch part)' 'seychelles' 'syrian arab republic'
'turks and caicos islands' 'chad'
'east asia & pacific (ida & ibrd countries)'
'europe & central asia (ida & ibrd countries)' 'togo' 'thailand'
'tajikistan' 'turkmenistan'
'latin america & the caribbean (ida & ibrd countries)' 'timor-leste'
'middle east & north africa (ida & ibrd countries)' 'tonga'
'south asia (ida & ibrd)' 'sub-saharan africa (ida & ibrd countries)'
'trinidad and tobago' 'tunisia' 'turkiye' 'tuvalu' 'tanzania' 'uganda'
'ukraine' 'upper middle income' 'uruguay' 'united states' 'uzbekistan'
'st. vincent and the grenadines' 'venezuela, rb' 'british virgin islands'


```
'virgin islands (u.s.)' 'viet nam' 'vanuatu' 'world' 'samoa' 'kosovo'
'yemen, rep.' 'south africa' 'zambia' 'zimbabwe']
```

```
In [6]: import pandas as pd

# Veriyi bir sözlük olarak tanımlıyoruz
data = {
    'Year': [i for i in range(1950, 2026)],
    'Population': [
        2499322157, 2543130380, 2590270899, 2640278797, 2691979339, 2746072141, 280
        2857866857, 2916108097, 2970292188, 3019233434, 3068370609, 3126686743, 319
        3267212338, 3337111983, 3406417036, 3475448166, 3546810808, 3620655275, 369
        3770163092, 3844800885, 3920251504, 3995517077, 4069437231, 4142505882, 421
        4289657708, 4365582871, 4444007706, 4524627658, 4607984871, 4691884238, 477
        4861730613, 4950063339, 5040984495, 5132293974, 5223704308, 5316175862, 540
        5492686093, 5577433523, 5660727993, 5743219454, 5825145298, 5906481261, 598
        6067758458, 6148898975, 6230746982, 6312407360, 6393898365, 6475751478, 655
        6641416218, 6725948544, 6811597272, 6898305908, 6985603105, 7073125425, 716
        7250593370, 7339013419, 7426597537, 7513474238, 7599822404, 7683789828, 776
        7840952880, 7909295151, 7975105156, 8045311447, 8118835999, 8191988453
    ]
}

# DataFrame'e dönüştürme
df = pd.DataFrame(data)

# Veriyi CSV dosyasına kaydetme
df.to_csv('world_population_data.csv', index=False)
```

```
In [7]: # CSV dosyasını okuma
df = pd.read_csv('world_population_data.csv')

# Veriyi görselleştirme veya analiz etme
print(df.head()) # İlk 5 satırı yazdırır

# Grafik veya diğer analizleri burada yapabilirsiniz
```

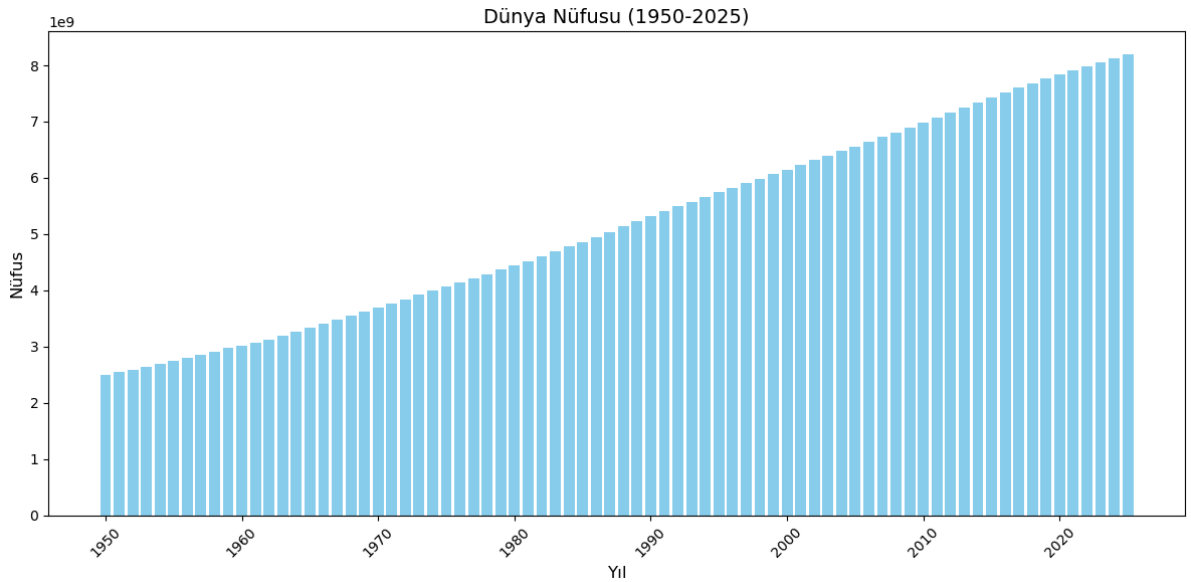
	Year	Population
0	1950	2499322157
1	1951	2543130380
2	1952	2590270899
3	1953	2640278797
4	1954	2691979339

```
In [8]: # Nüfus verisini bar grafiğiyle görselleştirme
import matplotlib.pyplot as plt

plt.figure(figsize=(12,6))
plt.bar(df['Year'], df['Population'], color='skyblue')

plt.title('Dünya Nüfusu (1950-2025)', fontsize=14)
plt.xlabel('Yıl', fontsize=12)
plt.ylabel('Nüfus', fontsize=12)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



```
In [9]: import matplotlib.pyplot as plt

# Ülke ve şehirleşme oranı verilerini tanımlayalım
countries = [
    'Arjantin', 'Kolombiya', 'Tayland', 'Güney Afrika', 'Mısır', 'Pakistan', 'Bangla',
    'Polonya', 'Romanya', 'Macaristan', 'Çekya', 'Portekiz', 'Yunanistan', 'Finlandiya',
    'Belçika', 'Hollanda', 'Japonya', 'ABD', 'Türkiye', 'Kanada', 'Almanya', 'İspanya',
    'Fransa', 'İsveç', 'Norveç', 'Avustralya', 'İtalya', 'Güney Kore', 'Rusya', 'Brazilya',
    'Endonezya', 'Malezya', 'Arnavutluk', 'Çili', 'Peru', 'Uruguay', 'Estonya', 'Letonya',
    'Kazakistan', 'Azerbaycan'
]

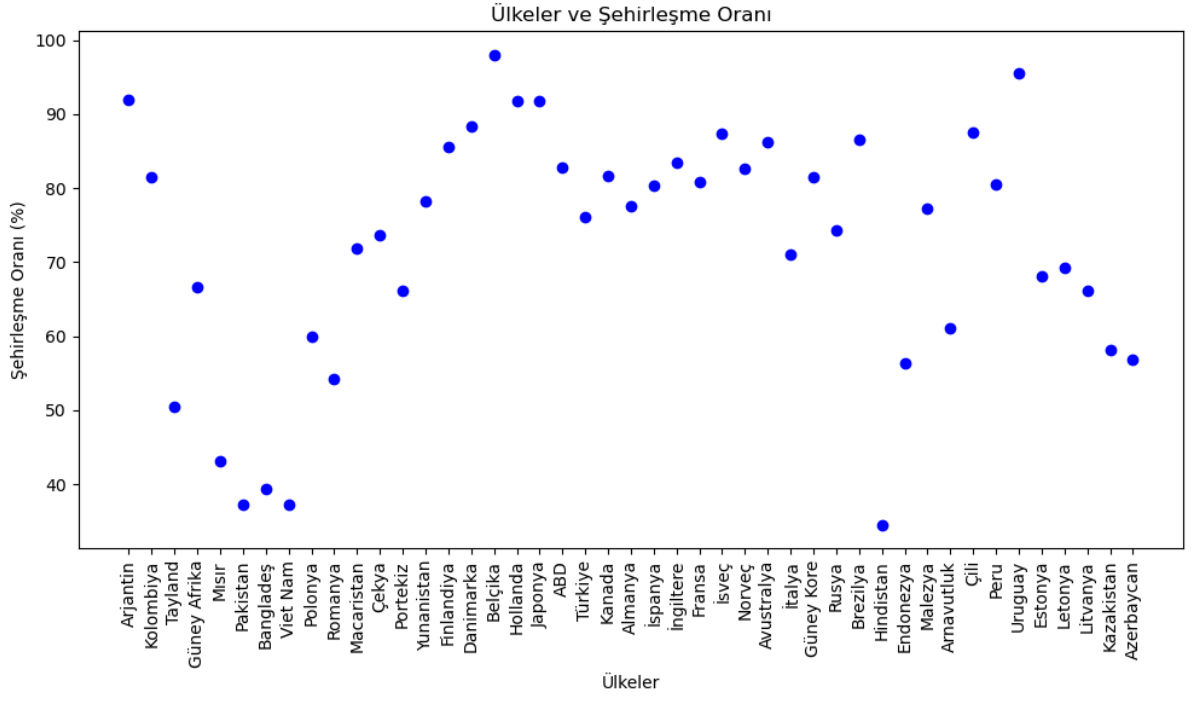
urbanization_rate = [
    92.0, 81.4, 50.4, 66.7, 43.1, 37.2, 39.4, 37.3, 60.0, 54.2, 71.9, 73.6, 66.1, 71.9,
    98.0, 91.8, 91.8, 82.8, 76.1, 81.6, 77.5, 80.3, 83.4, 80.8, 87.3, 82.7, 86.2, 71.9,
    86.6, 34.5, 56.4, 77.2, 61.0, 87.5, 80.5, 95.5, 68.1, 69.2, 66.1, 58.2, 56.8
]

# Grafik oluşturma
plt.figure(figsize=(10,6))
plt.scatter(countries, urbanization_rate, color='blue', marker='o')

# Grafiği etiketleme
plt.title('Ülkeler ve Şehirleşme Oranı')
plt.xlabel('Ülkeler')
plt.ylabel('Şehirleşme Oranı (%)')

# X eksenini döndürme (okunabilirliği artırmak için)
plt.xticks(rotation=90)

# Grafik gösterimi
plt.tight_layout()
plt.show()
```



In []:

In []:

In []:

In []:

In []:

In []:

In []: