

Данные о численности населения мира

https://www.kaggle.com/datasets/sazidthe1/world-population-data?select=world_population_data.csv



О наборе данных Контекст

Население мира претерпело значительный рост, превысив 7,5 миллиарда к середине 2019 года и продолжая расти сверх предыдущих оценок. Примечательно, что Китай и Индия стоят как две самые густонаселенные страны, при этом население Китая потенциально сталкивается с сокращением, в то время как траектория Индии намекает на то, чтобы превзойти ее к

2030 году. Этот значительный демографический сдвиг является лишь одним из аспектов глобального ландшафта, где такие страны, как Соединенные Штаты, Индонезия, Бразилия, Нигерия и другие, каждая из которых имеет население более 100 миллионов человек, играют ключевую роль.

Однако устойчивое снижение темпов роста изменит прогнозы. В то время как ожидается, что население мира превысит 8 миллиардов к 2030 году, рост заметно замедлится по сравнению с предыдущими десятилетиями. Конкретные страны, такие как Индия, Нигерия и несколько африканских стран, внесут значительный вклад в этот рост, потенциально удвоив свое население до плато ставок.

Содержание

Этот набор данных предоставляет исчерпывающие исторические данные о численности населения для стран и территорий во всем мире, предлагая представление о различных параметрах, таких как размер площади, континент, темпы роста населения, рейтинги и процент населения мира. С 1970 по 2023 год он включает в себя данные о численности населения за разные годы, что позволяет детально изучить демографические тенденции и изменения с течением времени.

Набор данных

Этот набор данных, структурированный с тщательной детализацией, предлагает широкий спектр информации в формате, способствующем анализу и исследованию. Благодаря таким параметрам, как население по годам, рейтинг стран, географические детали и темпы роста, он служит ценным ресурсом для исследователей, политиков и аналитиков. Кроме того, включение темпов роста и процентов населения мира дает четкое понимание того, как страны вносят свой вклад в глобальные демографические сдвиги.

Этот набор данных бесценен для тех, кто заинтересован в понимании исторических тенденций в области народонаселения, прогнозировании будущих демографических моделей и проведении углубленного анализа для информирования политики в различных секторах, таких как экономика, городское планирование, общественное здравоохранение и многое другое.

Структура

Этот набор данных, охватывающий период с 1970 по 2023 год, включает в себя следующие столбцы:

Имя Столбца описание

Rank Ранг по численности населения

CCA3 3-значный код страны/территории

Country Название страны

Continent Название континента

2023 Population Население страны в 2023 году

2022 Population Население страны в 2022 году

2020 Population Население страны в 2020 году

2015 Population Население страны в 2015 году

2010 Population Население страны в 2010 году

2000 Population Население страны в 2000 году

1990 Population Население страны в 1990 году

1980 Population Население страны в 1980 году

1970 Population Население страны в 1970 году

Area (km²) Размер территории страны/территории в квадратных километрах

Density (km²) Плотность населения на квадратный километр

Growth Rate Темпы роста населения по странам

World Population Percentage Процент населения по каждой стране

```
In [61]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
from matplotlib.ticker import MultipleLocator
```

```
In [62]: # переменные для размеров графиков
```

```
a = 20
```

```
In [63]: # Загрузка данных из CSV файла в датафрейм
file_path = '1_Данные_о_численности_населения_мира.csv'
world_population_data = pd.read_csv(file_path)

world_population_data.head(1000000)
```

Out [63]:

	rank	cca3	country	continent	2023 population	2022 population	2020 population	2015 population	2010 population	2000 population	1990 population	popu
0	1	IND	India	Asia	1428627663	1417173173	1396387127	1322866505	1240613620	1059633675	870452165	6968:
1	2	CHN	China	Asia	1425671352	1425887337	1424929781	1393715448	1348191368	1264099069	1153704252	9823:
2	3	USA	United States	North America	339996563	338289857	335942003	324607776	311182845	282398554	248083732	2231
3	4	IDN	Indonesia	Asia	277534122	275501339	271857970	259091970	244016173	214072421	182159874	1481:
4	5	PAK	Pakistan	Asia	240485658	235824862	227196741	210969298	194454498	154369924	115414069	806:
...
229	230	MSR	Montserrat	North America	4386	4390	4500	5059	4938	5138	10805	
230	231	FLK	Falkland Islands	South America	3791	3780	3747	3408	3187	3080	2332	
231	232	NIU	Niue	Oceania	1935	1934	1942	1847	1812	2074	2533	
232	233	TKL	Tokelau	Oceania	1893	1871	1827	1454	1367	1666	1669	
233	234	VAT	Vatican City	Europe	518	510	520	564	596	651	700	

234 rows x 17 columns

In [64]:

```
world_population_data.describe()
```

Out [64]:

	rank	2023 population	2022 population	2020 population	2015 population	2010 population	2000 population	1990 population	1970 population
count	234.000000	2.340000e+02	2.340000e+02	2.340000e+02	2.340000e+02	2.340000e+02	2.340000e+02	2.340000e+02	2.340000e+02
mean	117.500000	3.437442e+07	3.407441e+07	3.350107e+07	3.172996e+07	2.984524e+07	2.626947e+07	2.271022e+07	1.898462e+07
std	67.694165	1.373864e+08	1.367664e+08	1.355899e+08	1.304050e+08	1.242185e+08	1.116982e+08	9.783217e+07	8.178519e+07
min	1.000000	5.180000e+02	5.100000e+02	5.200000e+02	5.640000e+02	5.960000e+02	6.510000e+02	7.000000e+02	7.330000e+02
25%	59.250000	4.225982e+05	4.197385e+05	4.152845e+05	4.046760e+05	3.931490e+05	3.272420e+05	2.641158e+05	2.296142e+05
50%	117.500000	5.643895e+06	5.559944e+06	5.493074e+06	5.307400e+06	4.942770e+06	4.292907e+06	3.825410e+06	3.141146e+06
75%	175.750000	2.324537e+07	2.247650e+07	2.144798e+07	1.973085e+07	1.915957e+07	1.576230e+07	1.186923e+07	9.826054e+06
max	234.000000	1.428628e+09	1.425887e+09	1.424930e+09	1.393715e+09	1.348191e+09	1.264099e+09	1.153704e+09	9.823725e+08

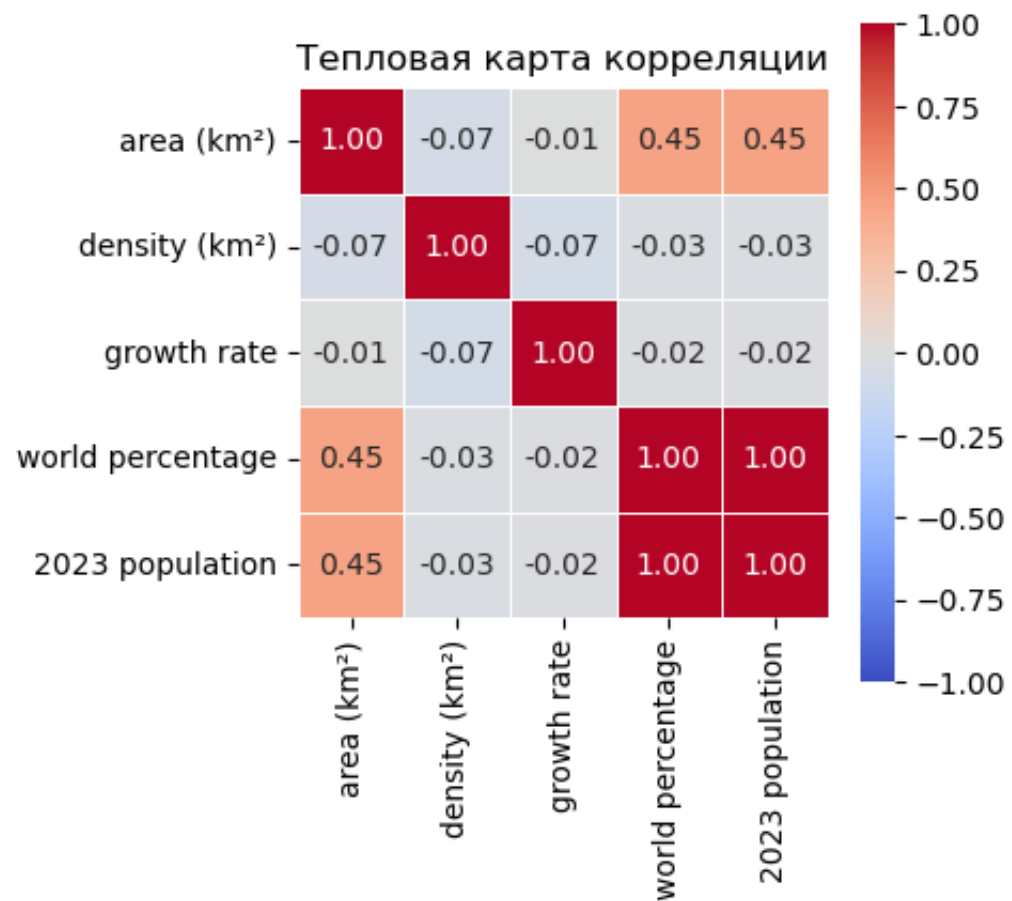
In [65]:

```
world_population_data_copy = world_population_data.copy(deep=True)

world_population_data_copy['growth rate'] = pd.to_numeric(world_population_data_copy['growth rate'].str.replace(' ', ''))
world_population_data_copy['world percentage'] = pd.to_numeric(world_population_data_copy['world percentage'].str.replace(' ', ''))

# Построение тепловой карты корреляции
correlation_matrix = world_population_data_copy[['area (km²)', 'density (km²)', 'growth rate', 'world percentage']]

plt.figure(figsize=(a * 0.2, a * 0.2))
cmap = sns.color_palette("coolwarm", as_cmap=True)
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap=cmap, vmin=-1, vmax=1, center=0, linewidths=.5, square=True)
plt.title('Тепловая карта корреляции')
plt.show()
```

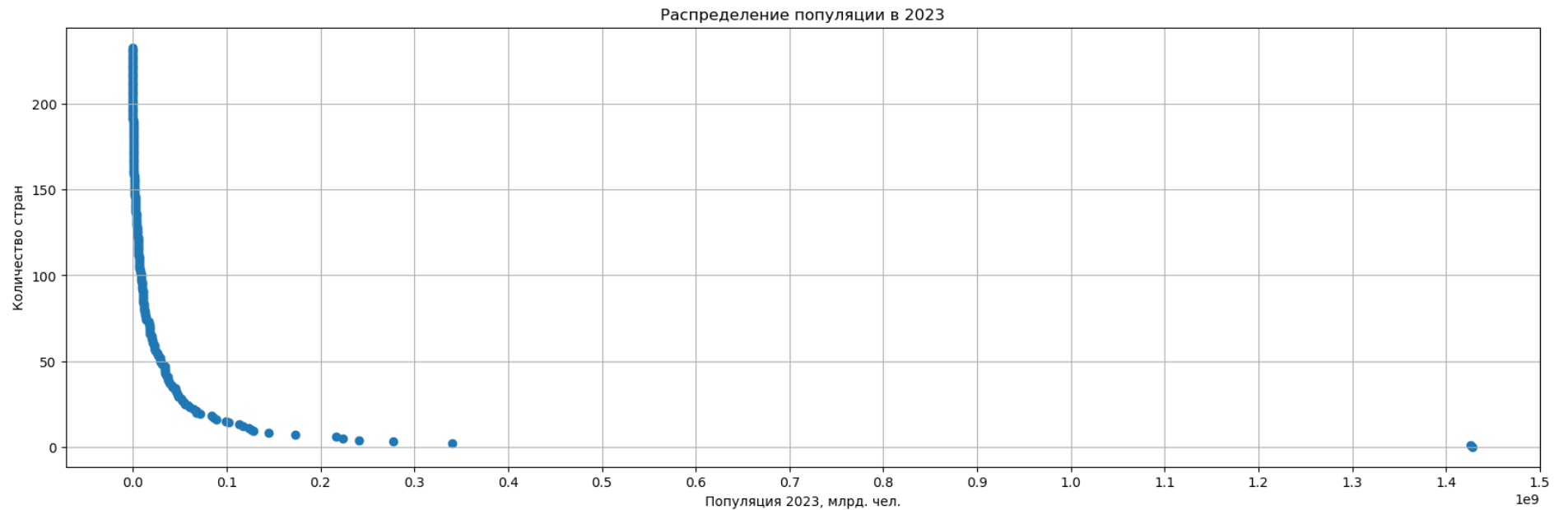


Прослеживается зависимость размера популяции и процента населения от площади страны.

In [66]: `#world_population_data.dtypes`

```
In [67]: world_population_data_copy = world_population_data.copy(deep = True)

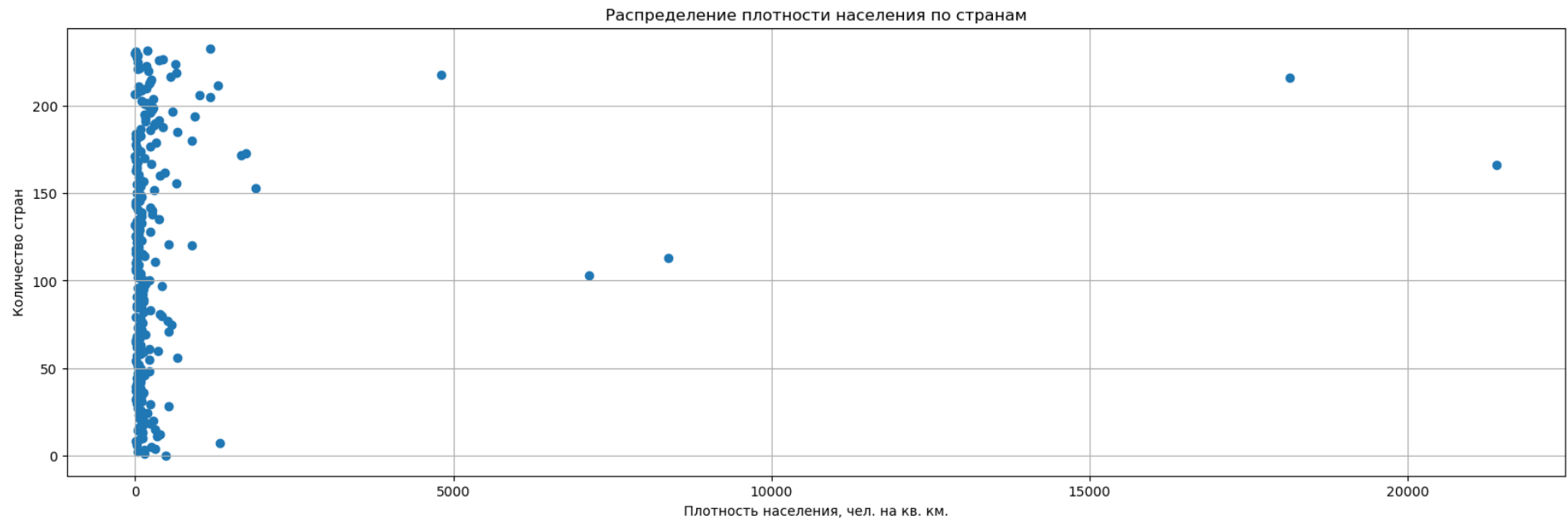
plt.figure(figsize=(a, a * 0.3))
plt.scatter(world_population_data_copy['2023 population'], range(len(world_population_data_copy)))
plt.title('Распределение популяции в 2023')
plt.xlabel('Популяция 2023, млрд. чел.')
plt.ylabel('Количество стран')
plt.grid(True)
plt.gca().xaxis.set_major_locator(MultipleLocator(100000000))
plt.show()
```



В большинстве стран популяция меньше 100 млн. человек.


```
In [68]: world_population_data_copy = world_population_data.copy(deep = True)

plt.figure(figsize=(a, a * 0.3))
plt.scatter(world_population_data_copy['density (km²)'], range(len(world_population_data_copy)))
plt.title('Распределение плотности населения по странам')
plt.xlabel('Плотность населения, чел. на кв. км.')
plt.ylabel('Количество стран')
plt.grid(True)
#plt.gca().xaxis.set_major_locator(MultipleLocator(100000000))
plt.show()
```



Средняя плотность населения 450 чел. на кв. км.

```
In [69]: x_columns = ['2023 population', '2022 population', '2020 population', '2015 population', '2010 population', '2000 population']
y_columns = ['2023 population', '2022 population', '2020 population', '2015 population', '2010 population', '2000 population']

# Инвертируйте порядок колонок
x_columns = x_columns[::-1]
y_columns = y_columns[::-1]

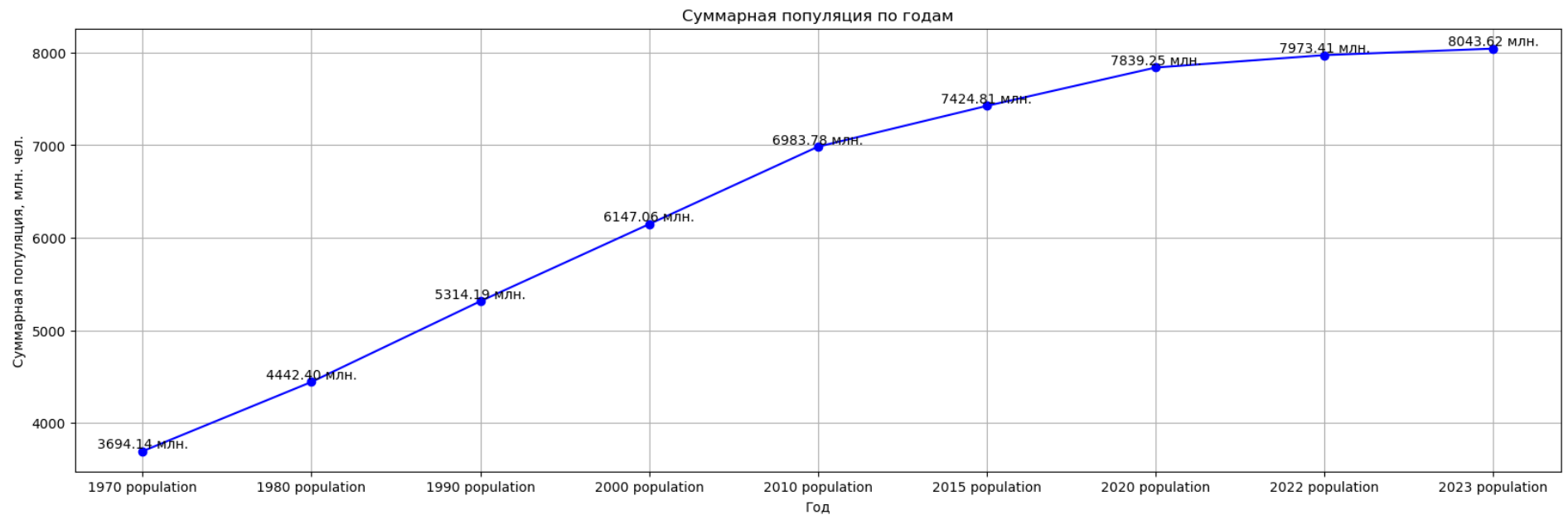
# Создайте новый датафрейм с суммарными значениями по выбранным колонкам
sum_df = world_population_data[y_columns].sum()

# Конвертируйте значения в миллионы
sum_df = sum_df / 1e6

# Постройте линейный график
plt.figure(figsize=(a, a * 0.3))
plt.plot(x_columns, sum_df, marker='o', linestyle='-', color='b')

# Добавление значений над точками
for x, y in zip(x_columns, sum_df):
    plt.text(x, y, f'{y:.2f} млн.', ha='center', va='bottom')

plt.title('Суммарная популяция по годам')
plt.xlabel('Год')
plt.ylabel('Суммарная популяция, млн. чел.')
plt.grid(True)
plt.show()
```



```
In [70]: # Установка размера графика
plt.figure(figsize=(a, a * 0.3))

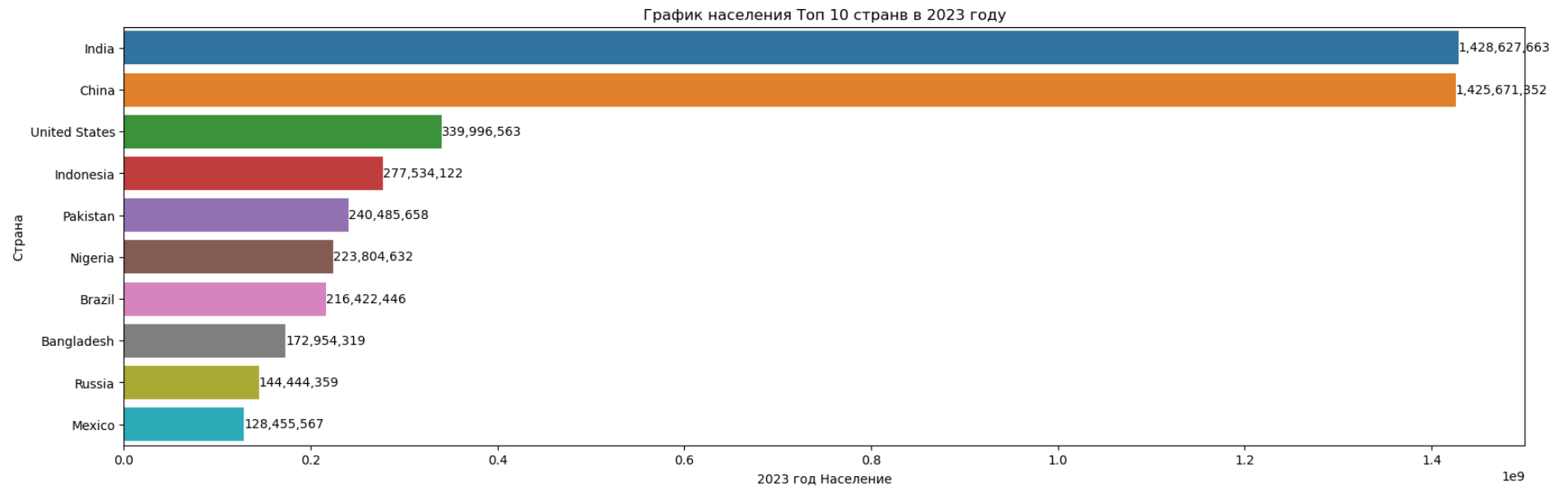
# Выбор только первых 10 стран
top_10_countries = world_population_data.head(10)

# Построение горизонтального столбчатого графика
plot = sns.barplot(x='2023 population', y='country', data=top_10_countries)

# Добавление значений над столбцами
for index, value in enumerate(top_10_countries['2023 population']):
    plot.text(value, index, f'{value:,.0f}', ha='left', va='center', fontsize=10)

# Настройка заголовка и меток осей
plt.title('График населения Топ 10 стран в 2023 году')
plt.xlabel('2023 год Население')
plt.ylabel('Страна')

# Отображение графика
plt.show()
```



```
In [71]: # Установка размера графика
plt.figure(figsize=(a, a * 0.3))

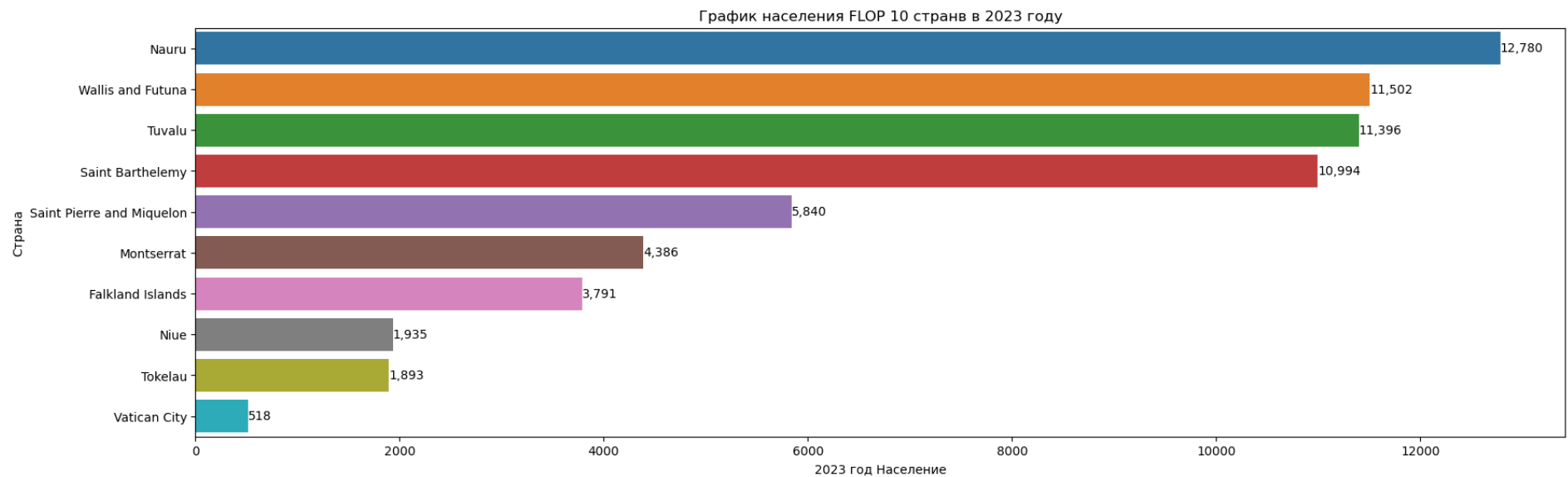
# Выбор только последних 10 стран
bottom_10_countries = world_population_data.tail(10)

# Построение горизонтального столбчатого графика
plot = sns.barplot(x='2023 population', y='country', data=bottom_10_countries)

# Добавление значений над столбцами
for index, value in enumerate(bottom_10_countries['2023 population']):
    plot.text(value, index, f'{value:,}', ha='left', va='center', fontsize=10)

# Настройка заголовка и меток осей
plt.title('График населения Топ 10 стран в 2023 году')
plt.xlabel('2023 год Население')
plt.ylabel('Страна')

# Отображение графика
plt.show()
```



```
In [72]: # Группировка данных по континентам и суммирование населения для 2023 года
continent_population = world_population_data.groupby('continent')['2023 population'].sum()

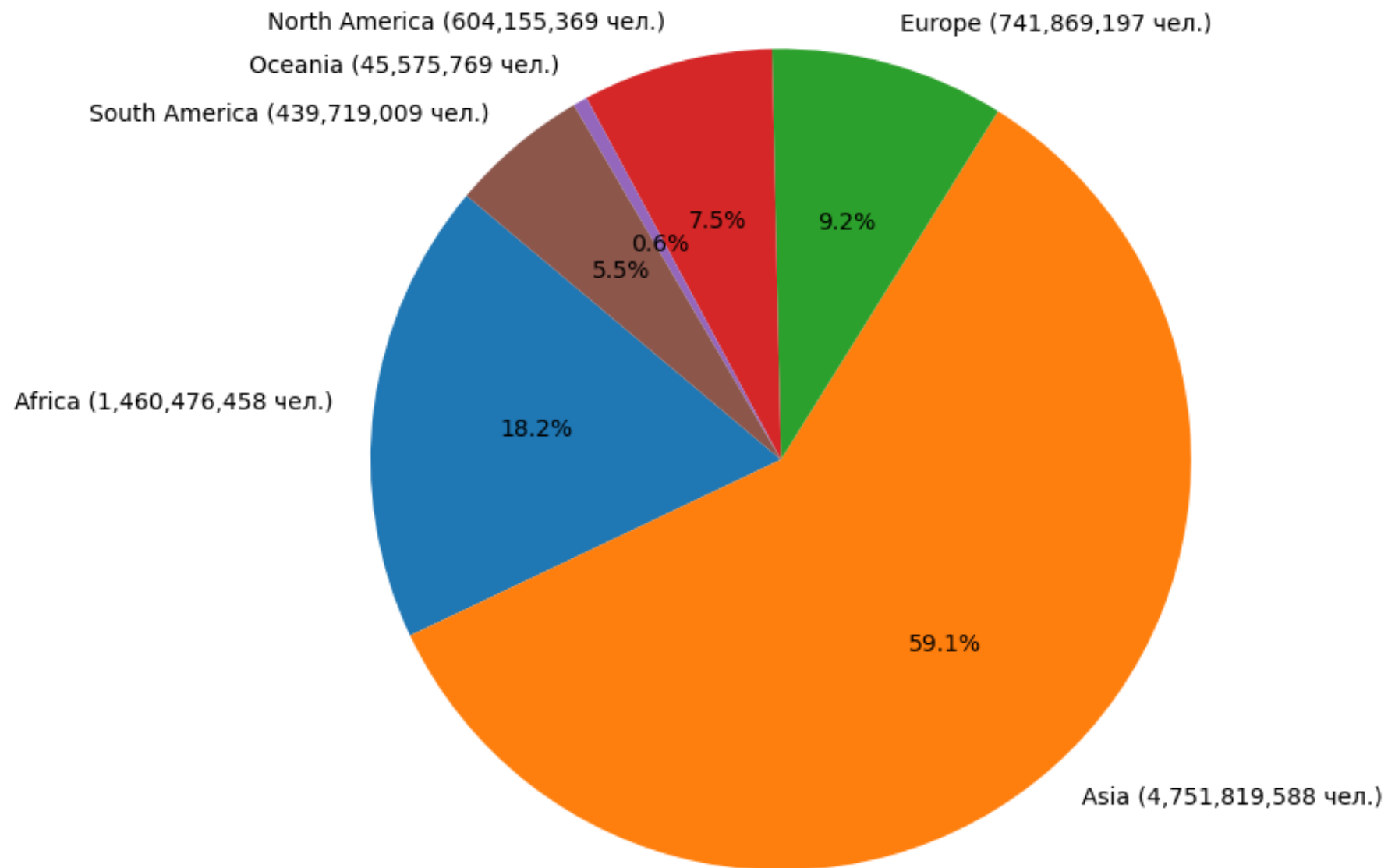
# Создание круговой диаграммы
plt.figure(figsize=(a * 0.5, a * 0.4))
patches, texts, autotexts = plt.pie(continent_population, labels=continent_population.index, autopct='%1.1f%%', s

# Добавление аннотаций (количество жителей) на круговую диаграмму
for text, autotext, value in zip(texts, autotexts, continent_population):
    text.set_text(f'{text.get_text()} ({value:,} чел.)')

# Настройка заголовка
plt.title('Распределение населения по континентам в 2023 году')

# Отображение круговой диаграммы
plt.show()
```

Распределение населения по континентам в 2023 году



```
In [73]: # Выбираем первые 10 стран
top_10_countries = world_population_data.head(10)

# Отсортируем столбцы с годами
sorted_columns = sorted(['2023 population', '2022 population', '2020 population', '2015 population',
                        '2010 population', '2000 population', '1990 population', '1980 population', '1970 popul

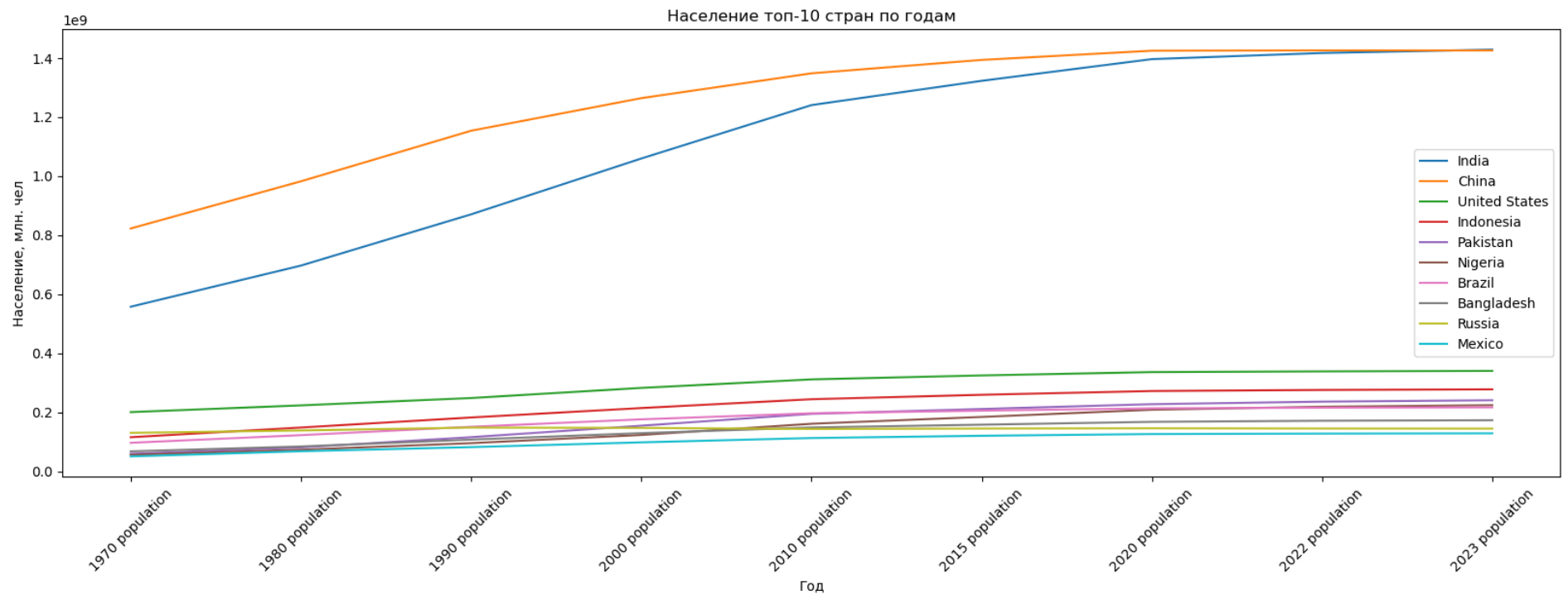
# Строим линейные графики для каждой страны
plt.figure(figsize=(a, a * 0.3))

for index, row in top_10_countries.iterrows():
    country_name = row['country']
    population_data = row[sorted_columns]

    plt.plot(population_data, label=country_name)

# Настройка осей и легенды
plt.xlabel('Год')
plt.ylabel('Население, млн. чел')
plt.title('Население топ-10 стран по годам')
plt.legend()
plt.xticks(rotation=45) # Поворот названий годов для лучшей читаемости

# Отображение графика
plt.show()
```



```
In [74]: # Создаем DataFrame down_Population
down_Population = world_population_data[['rank', 'сca3', 'country', 'continent', '2023 population', '1970 population']]

# Добавляем колонку down_Population как разницу между 2023 population и 1970 population
down_Population['down_Population'] = down_Population['2023 population'] - down_Population['1970 population']

# Сортируем по возрастанию по колонке down_Population
down_Population = down_Population.sort_values(by='down_Population')

# Отфильтровываем значения меньше или равно нулю
down_Population = down_Population[down_Population['down_Population'] >= 0]

# Выводим результат
down_Population.head(1000000)
```


Out [74]:

	rank	cca3	country	continent	2023 population	1970 population	down_Population
40	41	UKR	Ukraine	Europe	36744634	47279086	-10534452
109	110	BGR	Bulgaria	Europe	6687717	8582950	-1895233
131	132	GEO	Georgia	Asia	3728282	4800426	-1072144
136	137	BIH	Bosnia and Herzegovina	Europe	3210847	3815561	-604714
150	151	LVA	Latvia	Europe	1830211	2397414	-567203
141	142	LTU	Lithuania	Europe	2718352	3210147	-491795
129	130	HRV	Croatia	Europe	4008617	4492638	-484021
133	134	MDA	Moldova	Europe	3435931	3711140	-275209
93	94	HUN	Hungary	Europe	10156239	10315366	-159127
104	105	SRB	Serbia	Europe	7149077	7193533	-44456
155	156	EST	Estonia	Europe	1322765	1361999	-39234
63	64	ROU	Romania	Europe	19892812	19922618	-29806
229	230	MSR	Montserrat	North America	4386	11402	-7016
222	223	COK	Cook Islands	Oceania	17044	20470	-3426
231	232	NIU	Niue	Oceania	1935	5185	-3250
233	234	VAT	Vatican City	Europe	518	752	-234

In [75]:

```
# Создаем DataFrame down_Population
down_Population = world_population_data.copy()

# Добавляем колонку down_Population как разницу между 2023 population и 1970 population
down_Population['down_Population'] = down_Population['2023 population'] - down_Population['1970 population']

# Сортируем по возрастанию по колонке down_Population
down_Population = down_Population.sort_values(by='down_Population')

# Отфильтровываем значения меньше или равно нулю
down_Population = down_Population[down_Population['down_Population'] <= 0]
```

```
# Определение списка колонок с данными о населении
population_columns = ['2023 population', '2022 population', '2020 population',
                     '2015 population', '2010 population', '2000 population',
                     '1990 population', '1980 population', '1970 population']

# Переворачивание порядка колонок по оси x
population_columns.reverse()

# Построение графиков для каждой страны
for country in down_Population['country'].unique():
    country_data = down_Population[down_Population['country'] == country]

    # Сортировка данных по выбранным годам
    sorted_data = country_data[population_columns]

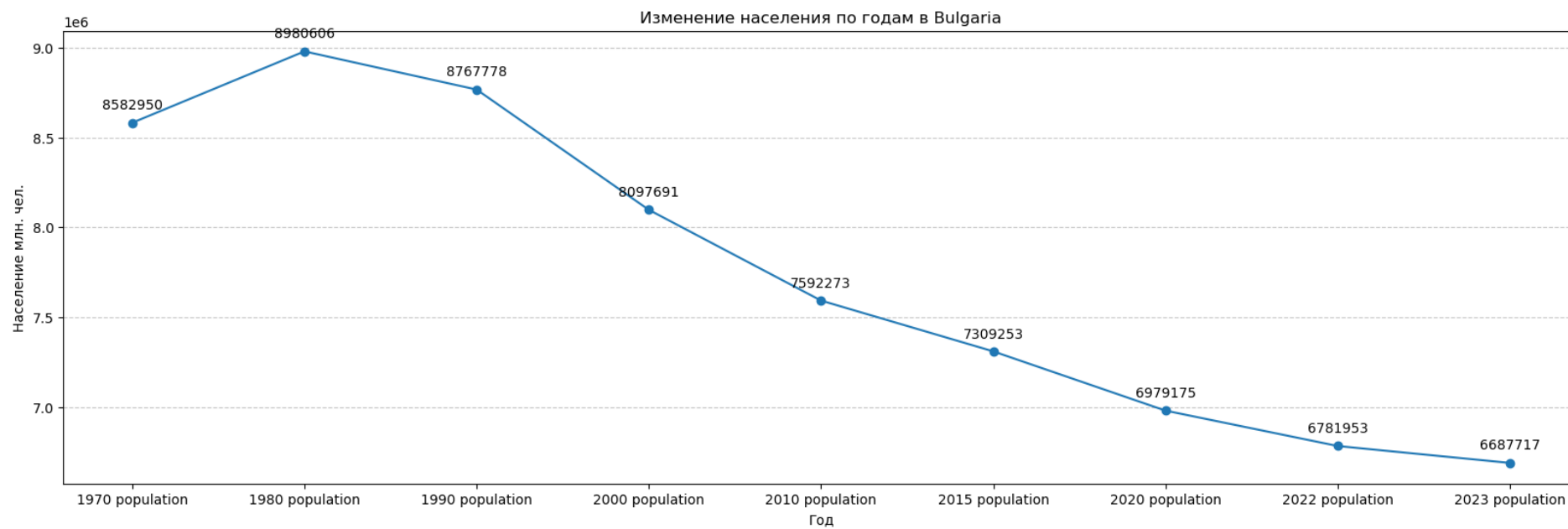
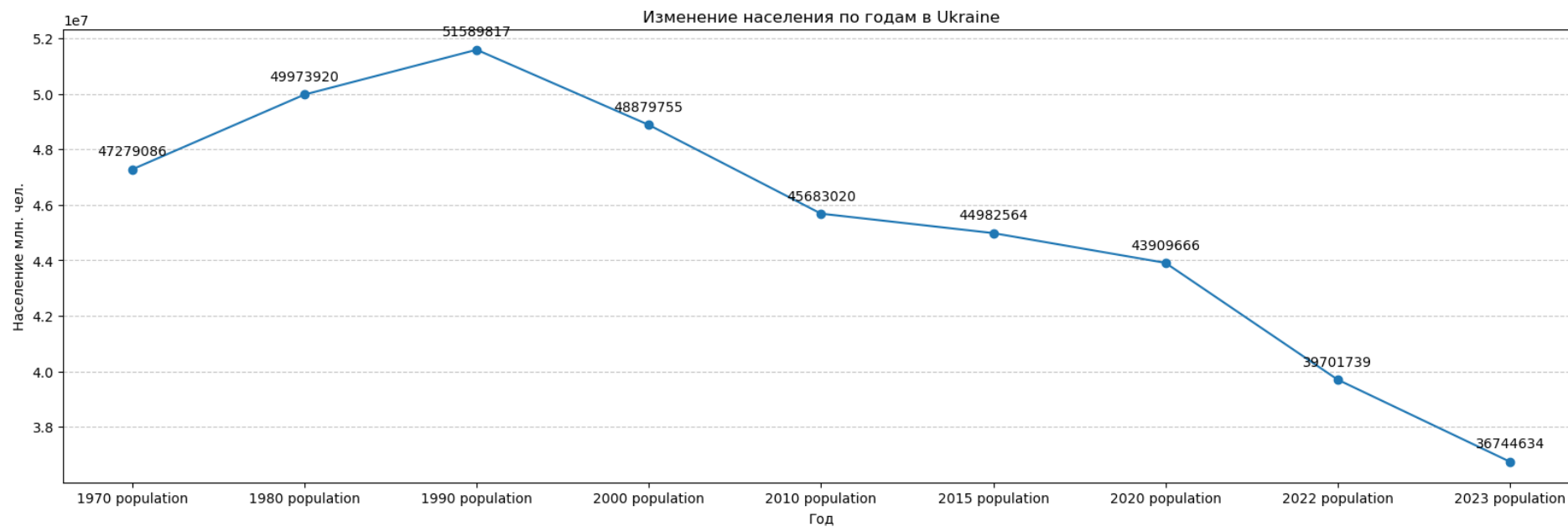
    # Построение линейного графика
    plt.figure(figsize=(a, a * 0.3))
    plt.plot(sorted_data.columns, sorted_data.iloc[0], marker='o', label=country)

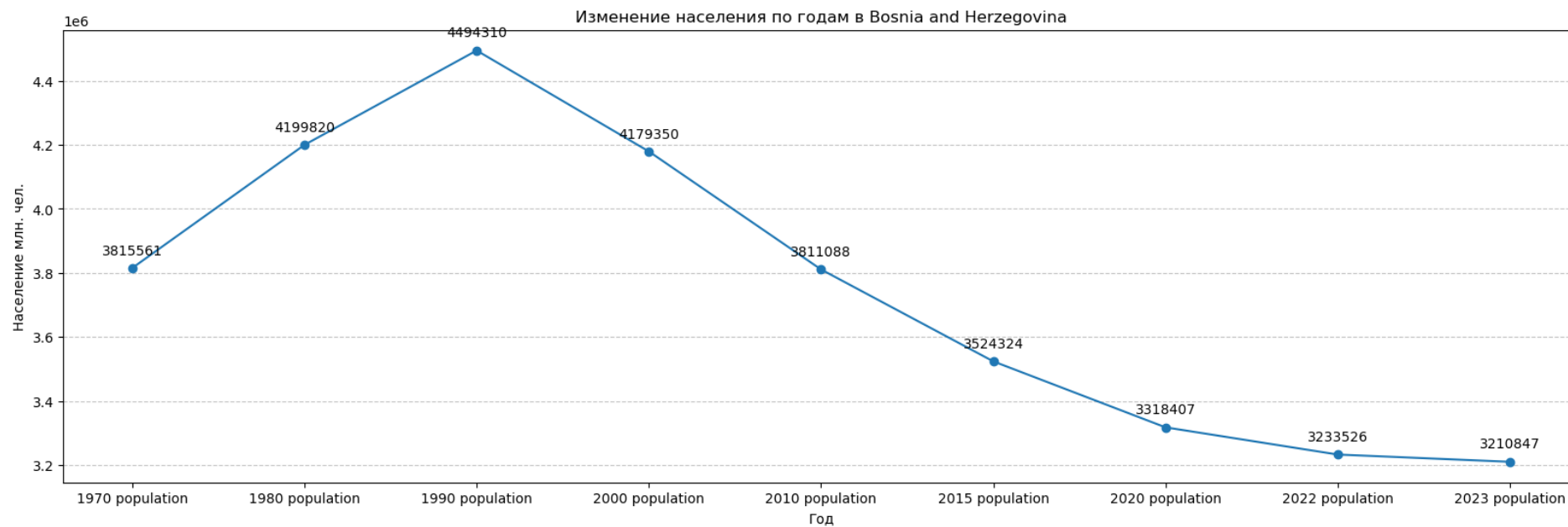
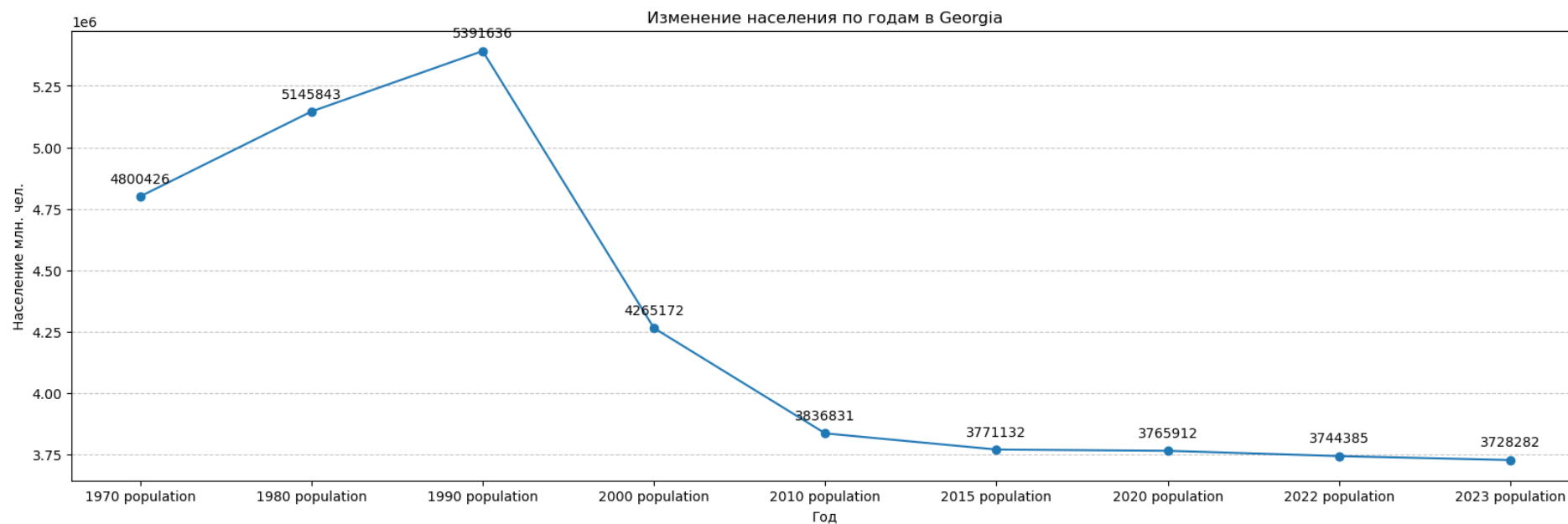
    # Добавление значений к точкам на графике
    for i, txt in enumerate(sorted_data.iloc[0]):
        plt.annotate(txt, (sorted_data.columns[i], sorted_data.iloc[0, i]), textcoords="offset points", xytext=(0, 10))

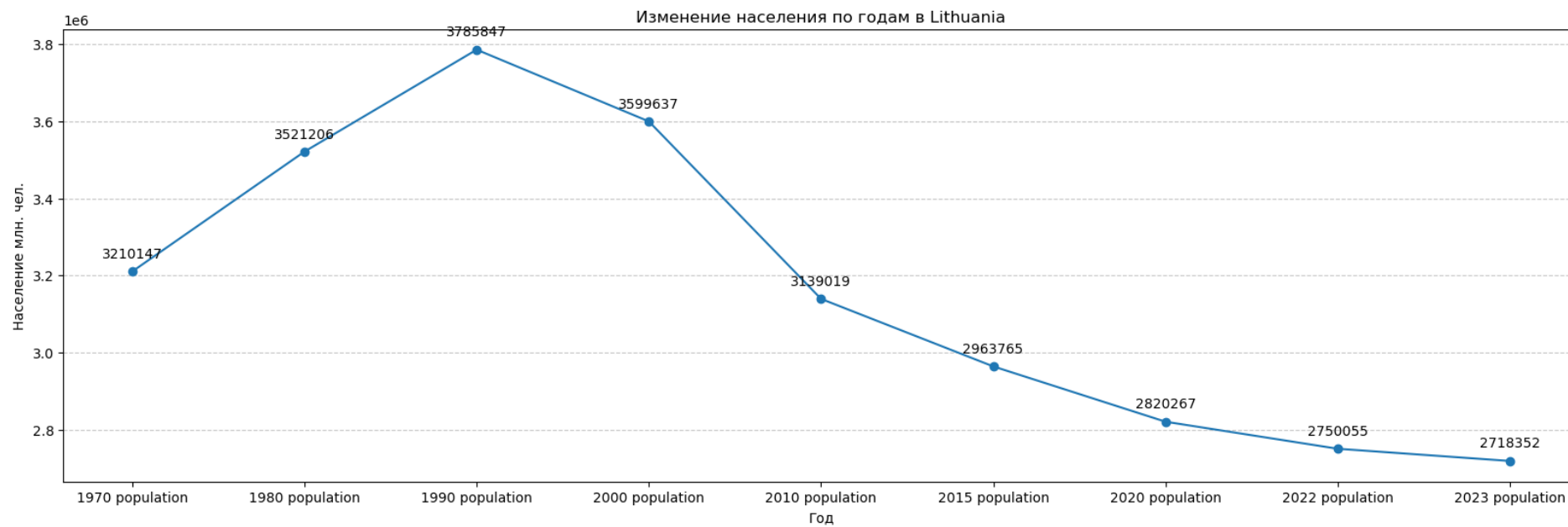
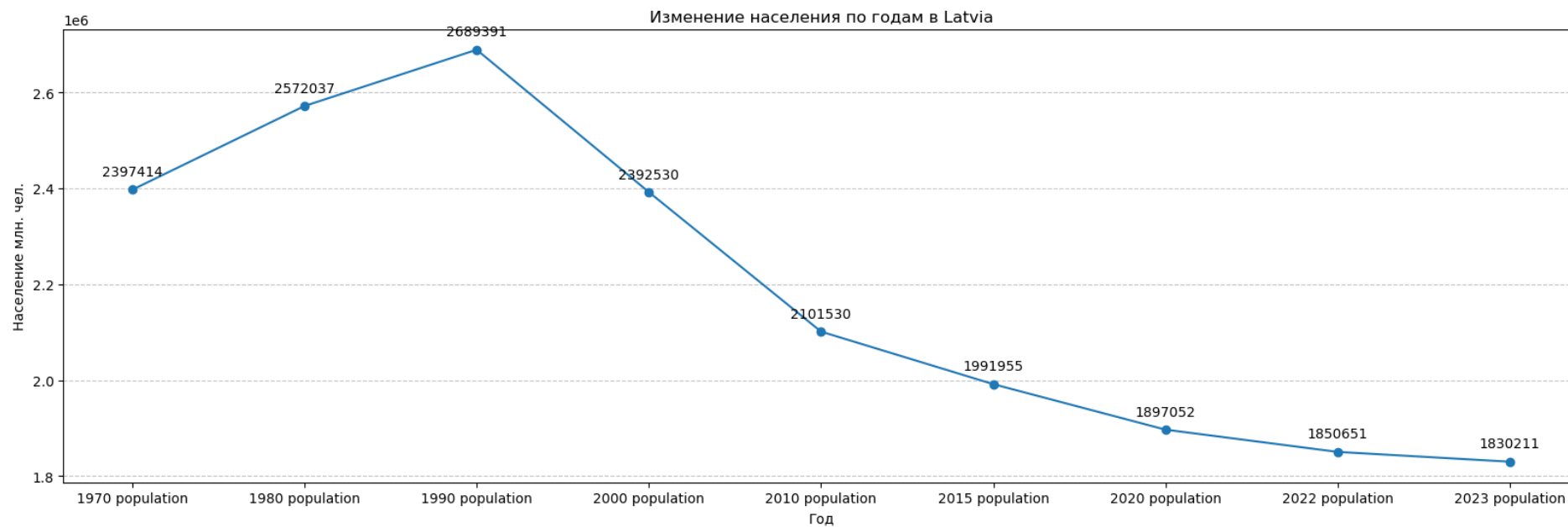
    plt.title(f'Изменение населения по годам в {country}')
    plt.xlabel('Год')
    plt.ylabel('Население млн. чел.')

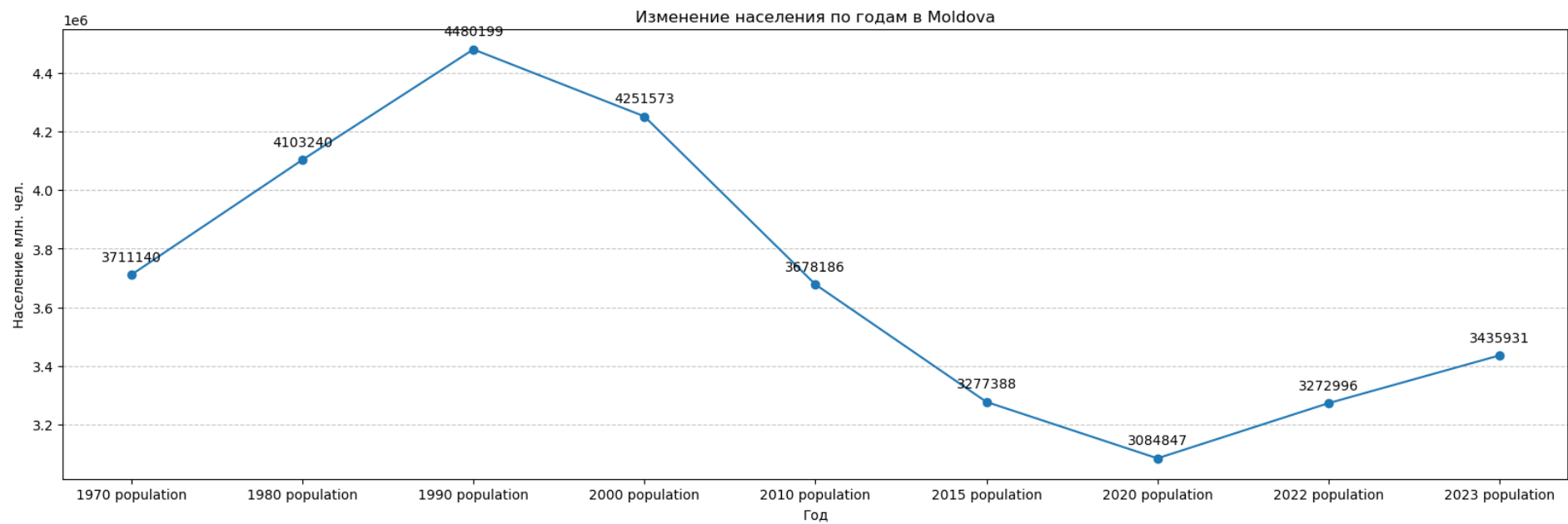
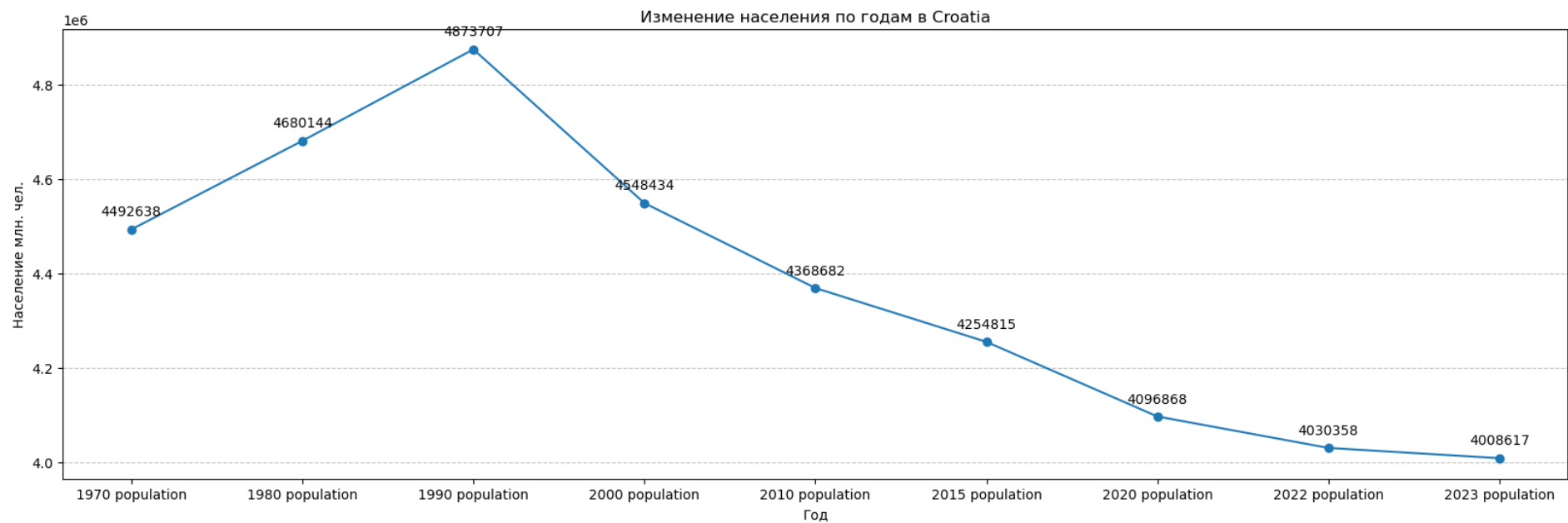
    # Добавление горизонтальных линий сетки
    plt.grid(axis='y', linestyle='--', alpha=0.7)

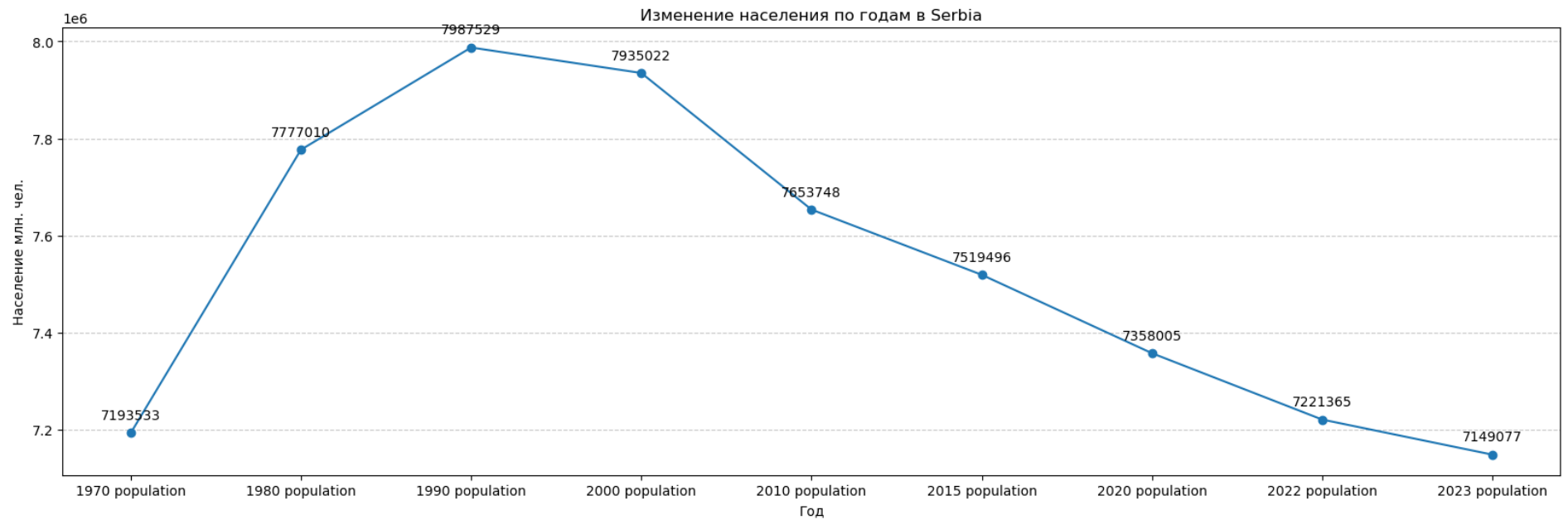
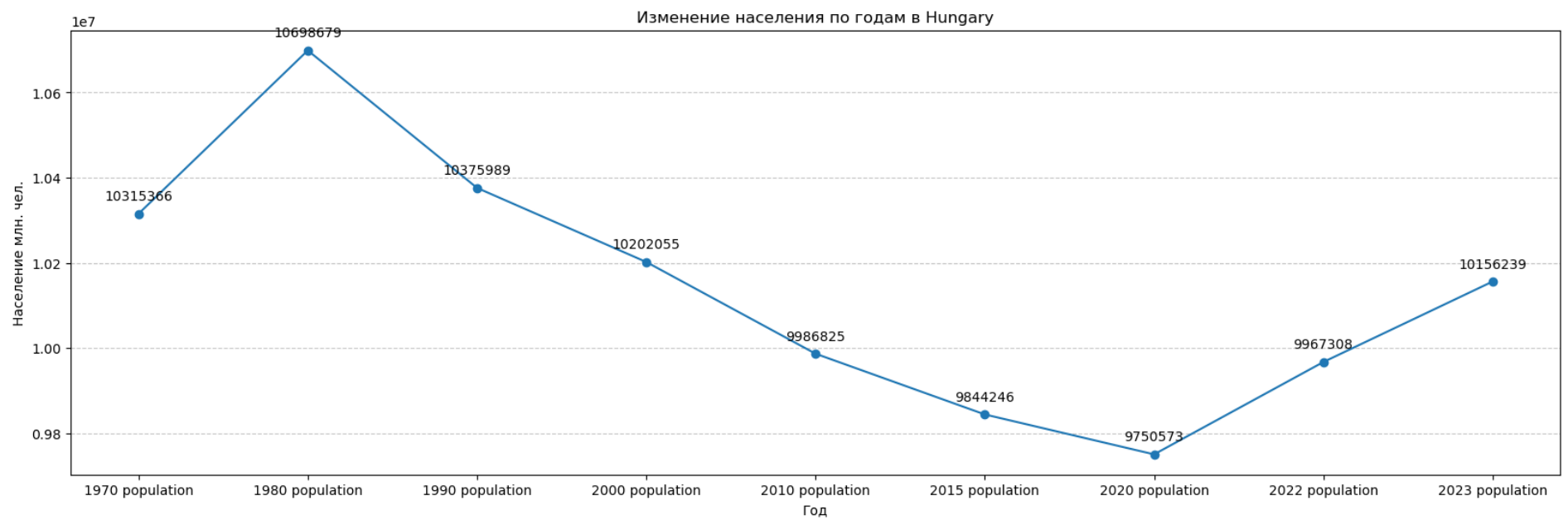
plt.show()
```

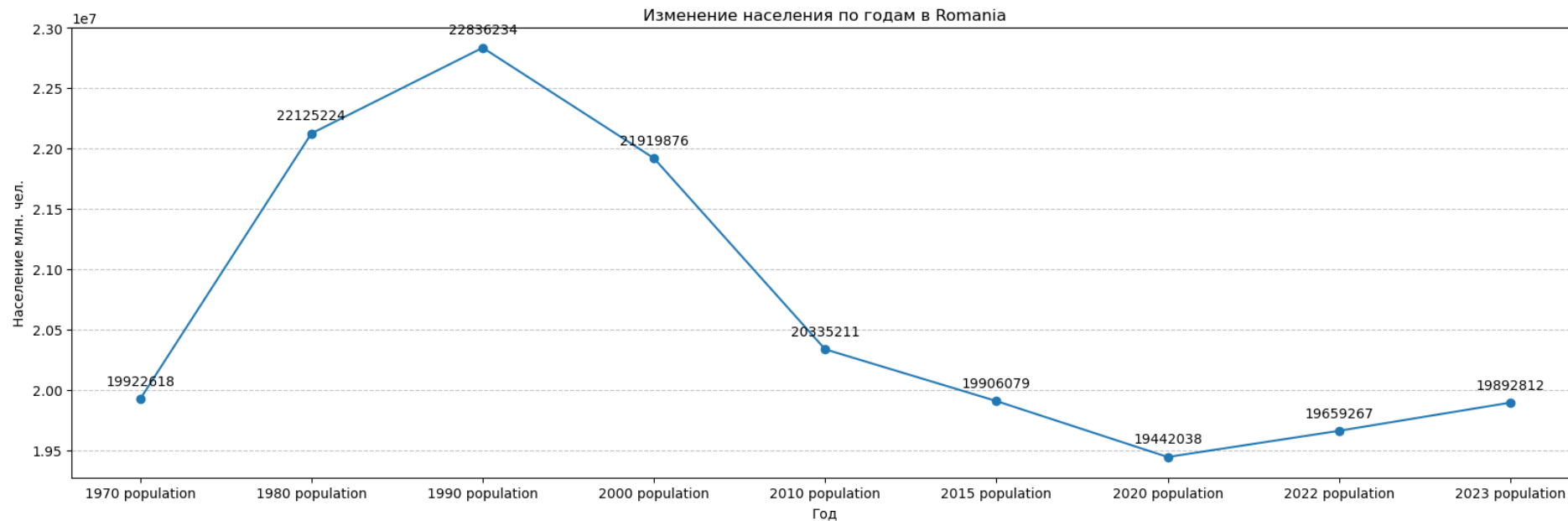
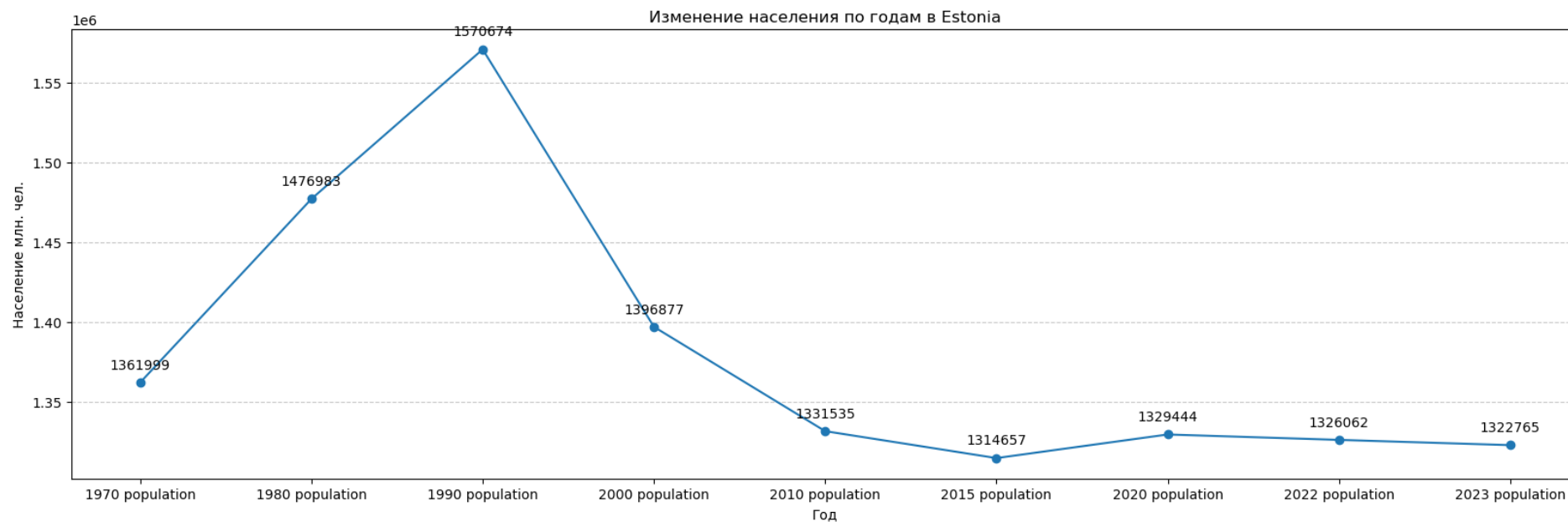


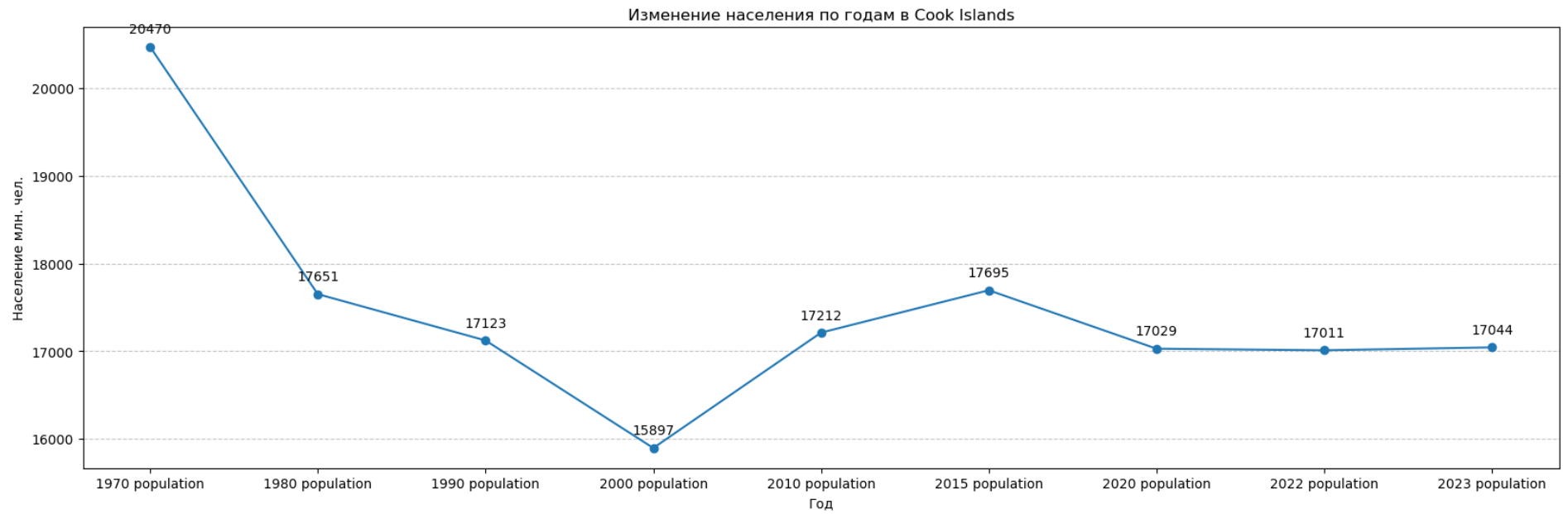
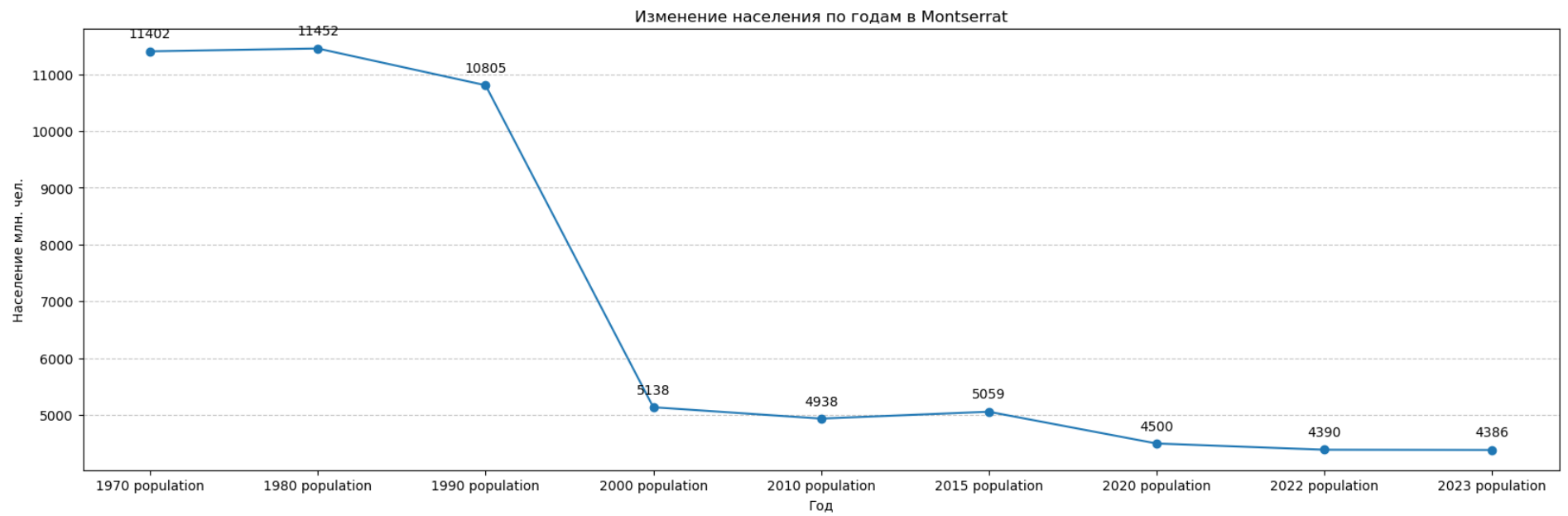


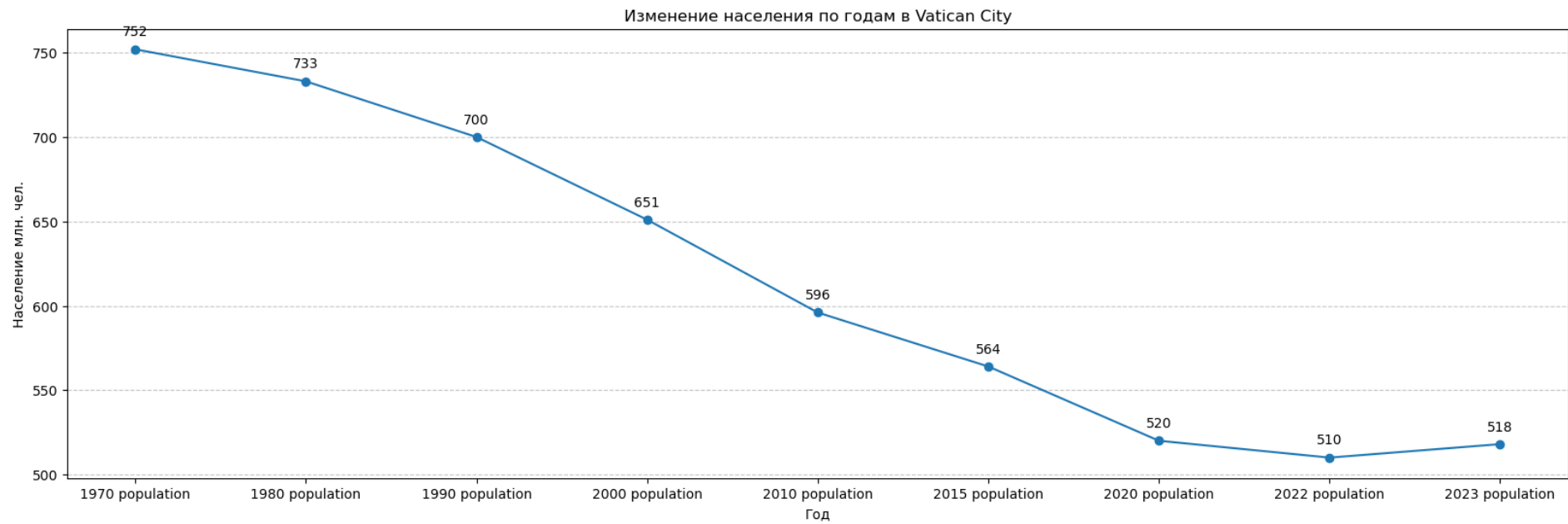
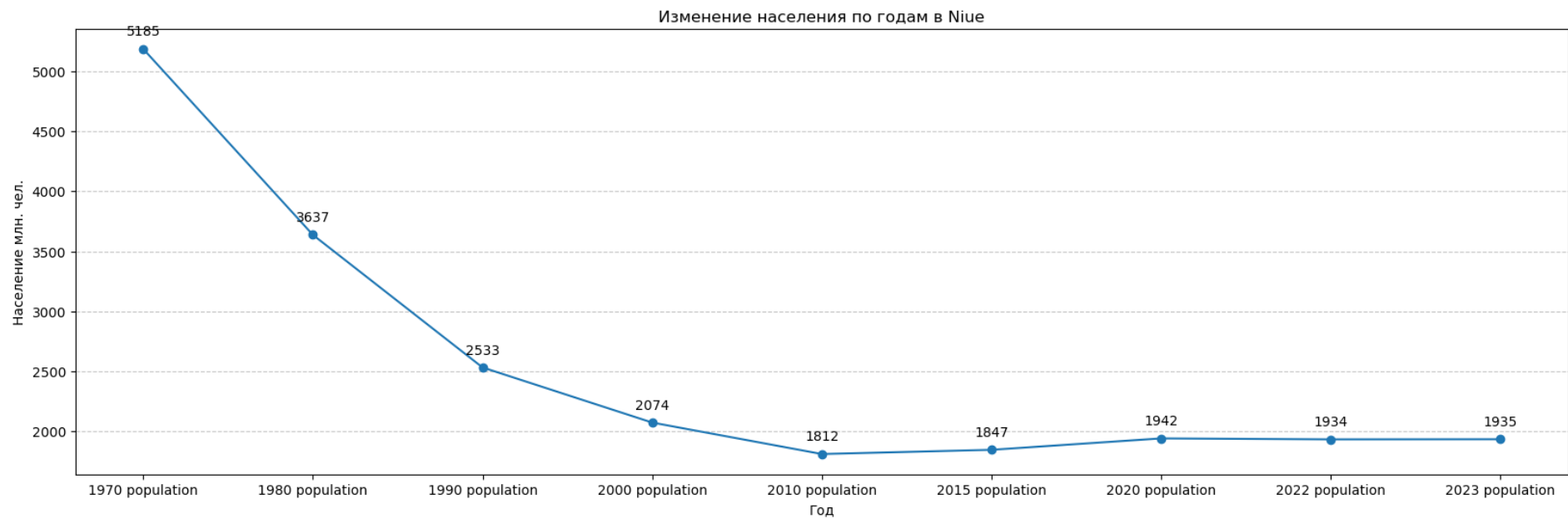












```
In [76]: # Сортировка по убыванию density (km²)
sorted_data = world_population_data.sort_values(by='density (km²)', ascending=False).copy()

# Построение горизонтальных столбчатых диаграмм для первых 10 стран
plt.figure(figsize=(a, a * 0.3))
bars = plt.barh(sorted_data['country'].head(10), sorted_data['density (km²)'].head(10), color='skyblue')
plt.xlabel('Плотность населения чел на km²')
plt.title('Топ 10 стран по плотности населения')
plt.gca().invert_yaxis() # чтобы страны были в порядке убывания

# Добавление значений на столбцы
for bar in bars:
    yval = bar.get_width() # теперь берем значение из ширины столбца
    plt.text(yval, bar.get_y() + bar.get_height()/2, round(yval, 2), ha='left', va='center')

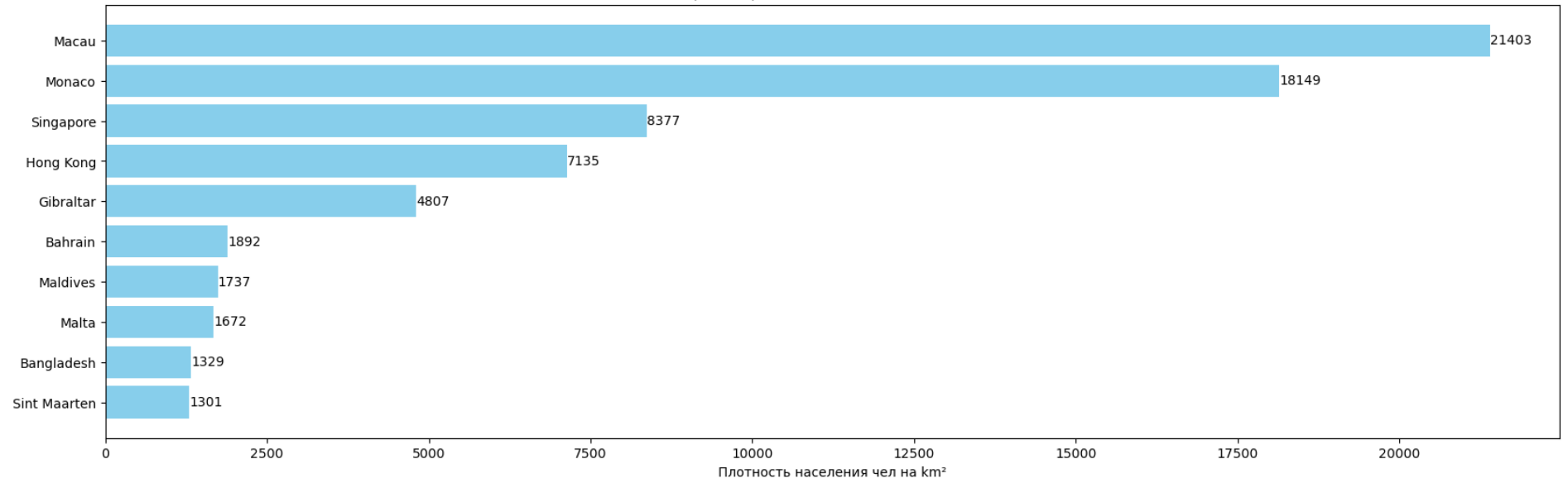
plt.show()

# Построение горизонтальных столбчатых диаграмм для последних 10 стран
plt.figure(figsize=(a, a * 0.3))
bars_bottom = plt.barh(sorted_data['country'].tail(10), sorted_data['density (km²)'].tail(10), color='skyblue')
plt.xlabel('Плотность населения чел на km²')
plt.title('БЛОП 10 стран по плотности населения')
plt.gca().invert_yaxis() # чтобы страны были в порядке убывания

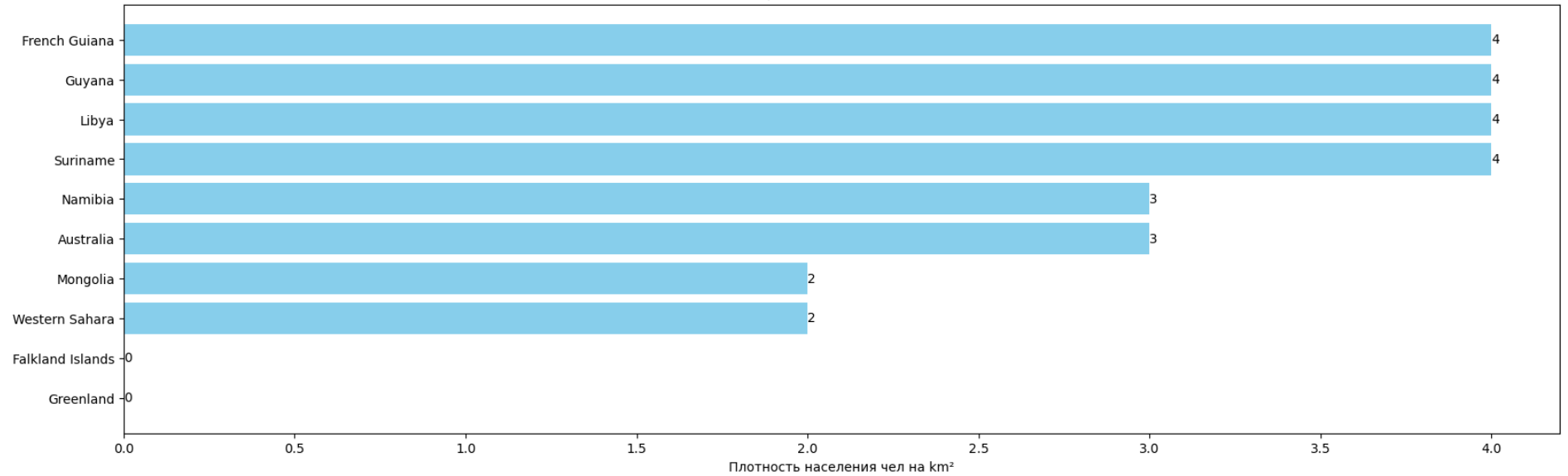
# Добавление значений на столбцы
for bar in bars_bottom:
    yval = bar.get_width() # теперь берем значение из ширины столбца
    plt.text(yval, bar.get_y() + bar.get_height()/2, round(yval, 2), ha='left', va='center')

plt.show()
```

Топ 10 стран по плотности населения



FLOP 10 стран по плотности населения



```
In [77]: continent_metriks = world_population_data.copy(deep=True)

pivot_table = pd.pivot_table(continent_metriks, values=['2023 population', 'area (km²)'], index='continent', aggfunc='sum')

pivot_table['density (km²)'] = pivot_table['2023 population'] / pivot_table['area (km²)']

pivot_table = pivot_table.reset_index()

# Сортировка по увеличению '2023 population'
pivot_table = pivot_table.sort_values(by='2023 population', ascending=True)

# Выводим DataFrame
pivot_table.head(1000000)
```

```
Out[77]:
```

	continent	2023 population	area (km²)	density (km²)
4	Oceania	45575769	8515218.00	5.352273
5	South America	439719009	17833382.00	24.657073
3	North America	604155369	24244178.00	24.919606
2	Europe	741869197	23010411.26	32.240588
0	Africa	1460476458	30317963.00	48.171985
1	Asia	4751819588	32138143.90	147.856068

```
In [78]: # Сортируем DataFrame по каждому столбцу
sorted_by_population = pivot_table.sort_values(by='2023 population', ascending=False)
sorted_by_area = pivot_table.sort_values(by='area (km²)', ascending=False)
sorted_by_density = pivot_table.sort_values(by='density (km²)', ascending=False)

# Создаем три горизонтальных столбчатых графика
plt.figure(figsize=(a, a * 0.5))

# График для '2023 population'
plt.subplot(3, 1, 1)
bars_population = plt.barh(sorted_by_population['continent'], sorted_by_population['2023 population'], color='blue')
plt.xlabel('2023 Население, млрд. чел.')
plt.title('2023 Распределение населения по континентам')
```

```
# Добавляем значения внутри столбцов
for bar in bars_population:
    yval = bar.get_width()
    plt.text(yval, bar.get_y() + bar.get_height()/2, round(yval, 2), ha='left', va='center')

# Добавляем вертикальные линии сетки
plt.grid(axis='x', linestyle='--', alpha=0.6)

# График для 'area (km²)'
plt.subplot(3, 1, 2)
bars_area = plt.barh(sorted_by_area['continent'], sorted_by_area['area (km²)'], color='green')
plt.xlabel('Площадь км²')
plt.title('Площадь континента км²')

# Добавляем значения внутри столбцов
for bar in bars_area:
    yval = bar.get_width()
    plt.text(yval, bar.get_y() + bar.get_height()/2, round(yval, 2), ha='left', va='center')

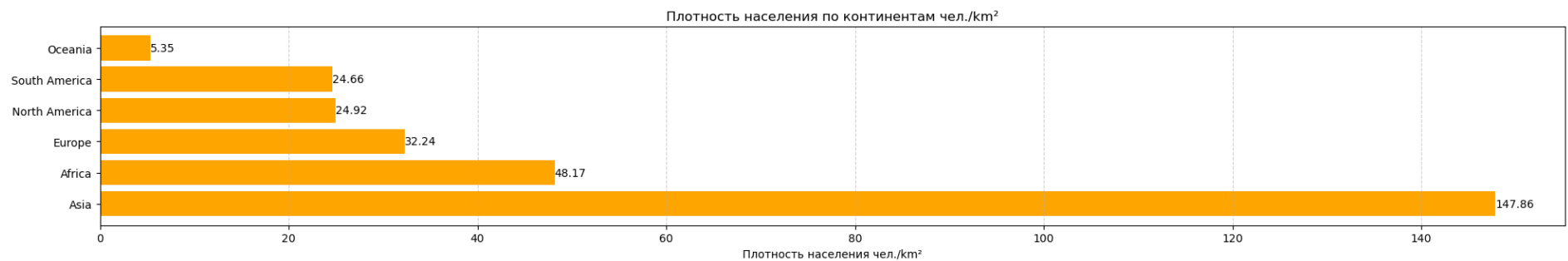
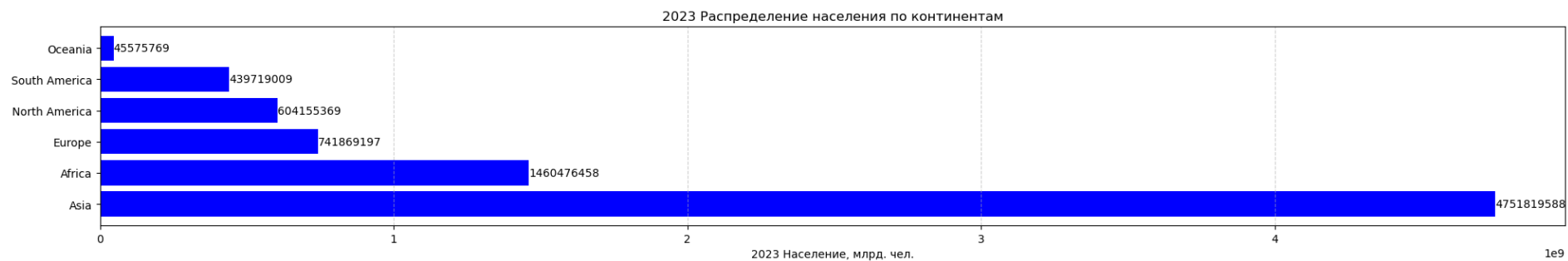
# Добавляем вертикальные линии сетки
plt.grid(axis='x', linestyle='--', alpha=0.6)

# График для 'density (km²)'
plt.subplot(3, 1, 3)
bars_density = plt.barh(sorted_by_density['continent'], sorted_by_density['density (km²)'], color='orange')
plt.xlabel('Плотность населения чел./км²')
plt.title('Плотность населения по континентам чел./км²')

# Добавляем значения внутри столбцов
for bar in bars_density:
    yval = bar.get_width()
    plt.text(yval, bar.get_y() + bar.get_height()/2, round(yval, 2), ha='left', va='center')

# Добавляем вертикальные линии сетки
plt.grid(axis='x', linestyle='--', alpha=0.6)

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



```
In [79]: fig = px.choropleth(
    world_population_data,
    locations="cca3",
    color="2023 population",
    hover_name="country",
    color_continuous_scale="RdYlGn", # Красно-зелёная цветовая схема
    #color_continuous_scale=px.colors.sequential.Plasma,
    title="Численность населения по странам в 2023 году",
    labels={"2023 population": "Население 2023"},
    projection="natural earth"
)

# Увеличение размера карты
fig.update_layout(width=1000, height=500)

# Отображение карты
fig.show()
```