Лучшая страна для жизни в 2024 году

https://www.kaggle.com/datasets/rafsunahmad/best-country-to-live-in-2024

Лучшая страна для жизни в соответствии с отчетом по индексу человеческого развития

О наборе данных Этот набор данных содержит данные разных стран. Этот набор данных о лучшей стране для жизни в 2024 году. Этот набор данных лучше всего подходит для исследовательского анализа данных.

population_2024 - Общая численность населения в 2024 году

population_growthRate - Темпы роста населения

land_area - Общая Площадь Страны

country - Название страны

region - Название региона

unMember - Является ли страна членом Организации Объединенных Наций или нет

population_density - Плотность Населения На КМ

population_densityMi - Плотность населения на милю

share_borders - Границы с другой страной

Hdi2021 - Индекс человеческого развития, является метрикой, составленной Программой развития Организации Объединенных Наций и используемой для количественной оценки "средних достижений страны в трех основных измерениях развития человека: долгая и здоровая жизнь, знания и достойный уровень жизни.

Hdi2020 - Индекс человеческого развития, является метрикой, составленной Программой развития Организации Объединенных Наций и используемой для количественной оценки "средних достижений страны в трех основных измерениях развития человека: долгая и здоровая жизнь, знания и достойный уровень жизни.

WorldHappiness2022 - Индекс счастья

In [101…

```
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
import numpy as np
from sklearn.linear_model import LinearRegression
from scipy.stats import linregress
from sklearn.model_selection import train_test_split
```

In [101…

```
#Переменные

#переменая для head
a = 1000000

#переменая для размера ерафиков
b = 20
```

In [101…

```
# Загрузка CSV-файла в датафрейм
input_raw = pd.read_csv('2_Лучшая_страна_для_жизни_в_2024 году.csv')

# Отображение первых нескольких строк датафрейма, чтобы убедиться, что данные были успешно загружены
input_raw.head(a)
```

Out[1014]:

	population_2024	population_growthRate	land_area	country	region	unMember	population_density	population_densityMi
0	1441719852	0.00916	3287590	India	Asia	True	484.9067	1255.9084
1	1425178782	-0.00035	9706961	China	Asia	True	151.2174	391.6530
2	341814420	0.00535	9372610	United States	North America	True	37.3673	96.7813
3	279798049	0.00816	1904569	Indonesia	Asia	True	149.0254	385.9758
4	245209815	0.01964	881912	Pakistan	Asia	True	318.0908	823.8551
...	...	...	...	...	...	...	...	...
136	867605	0.01823	1862	Comoros	Africa	True	466.2037	1207.4675
137	661594	0.01043	2586	Luxembourg	Europe	True	256.9796	665.5772
138	626102	-0.00061	13812	Montenegro	Europe	True	46.5503	120.5654
139	536740	0.00313	316	Malta	Europe	True	1677.3125	4344.2394
140	377689	0.00632	103000	Iceland	Europe	True	3.7458	9.7016

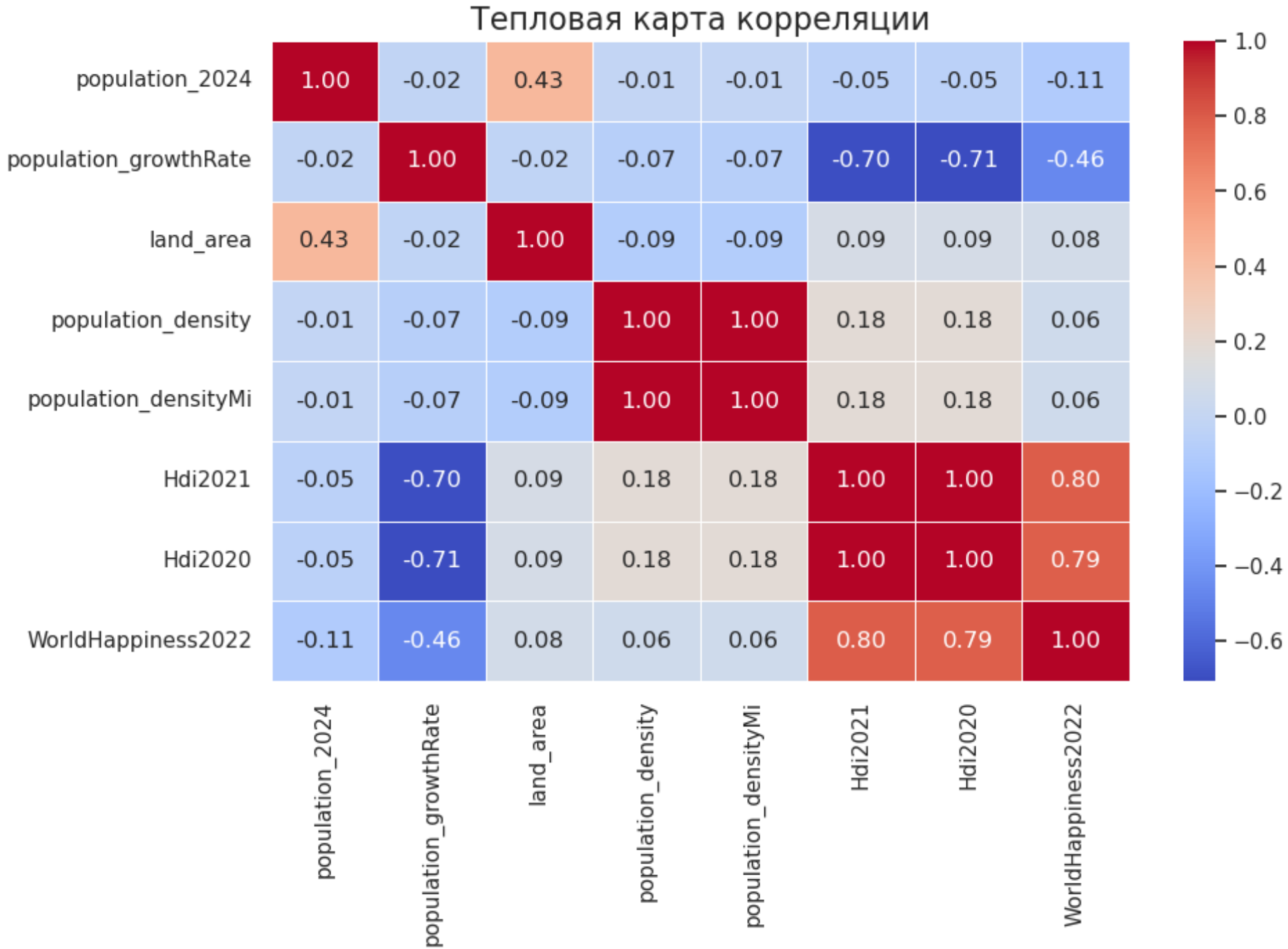
141 rows × 12 columns

In [101…

```
#input_raw.dtypes
```

In [101…

```
input_raw_copy = input_raw.copy(deep = True)

# Создаем подмножество DataFrame с нужными столбцами
selected_columns = ['population_2024', 'population_growthRate', 'land_area',
                    'population_density', 'population_densityMi', 'Hdi2021', 'Hdi2020', 'WorldHappiness2022']

subset_df = input_raw_copy[selected_columns]

# Создаем тепловую карту корреляции
correlation_matrix = subset_df.corr()

# Настраиваем размер фигуры
plt.figure(figsize=(b * 0.5, b * 0.3))

# Рисуем тепловую карту с использованием seaborn
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)

# Настраиваем заголовок
plt.title('Тепловая карта корреляции')

# Показываем график
plt.show()
```


Тепловая карта корреляции
population_2024 1.00 -0.02 0.43 -0.01 -0.01 -0.05 -0.05 -0.11
population_growthRate -0.02 1.00 -0.02 -0.07 -0.07 -0.70 -0.71 -0.46
land_area 0.43 -0.02 1.00 -0.09 -0.09 0.09 0.09 0.08
population_density -0.01 -0.07 -0.09 1.00 1.00 0.18 0.18 0.06
population_densityMi -0.01 -0.07 -0.09 1.00 1.00 0.18 0.18 0.06
Hdi2021 -0.05 -0.70 0.09 0.18 0.18 1.00 1.00 0.80
Hdi2020 -0.05 -0.71 0.09 0.18 0.18 1.00 1.00 0.79
WorldHappiness2022 -0.11 -0.46 0.08 0.06 0.06 0.80 0.79 1.00
population_2024
population_growthRate
land_area
population_density
population_densityMi
Hdi2021
Hdi2020
WorldHappiness2022
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

Вывод:

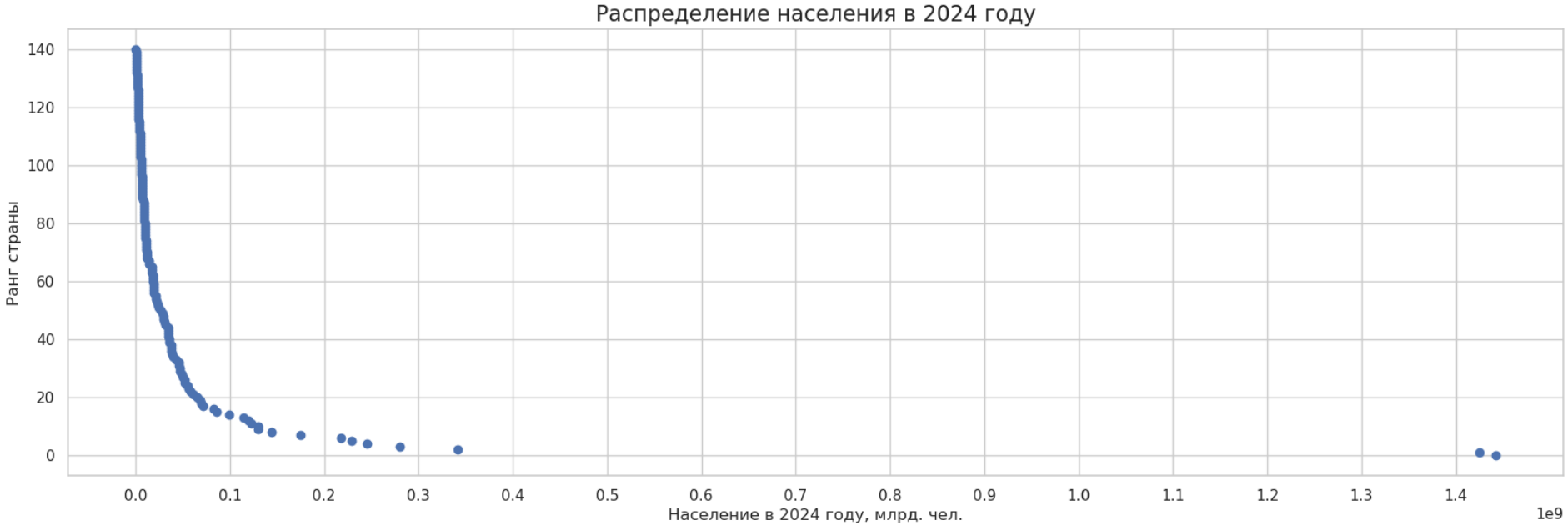
Есть небольшая зависимость популяции населения от площади страны.

Чем выше уровень человеческого развития, тем выше индекс счастья

Рост популяции зависит обратно от индекса развития и индекса счастья, чем ниже индекс развития и индекс счастья, тем выше рост популяции.

In [101…

```
input_raw_copy = input_raw.copy(deep = True)

plt.figure(figsize=(b, b * 0.3))

# Построение точечного графика
plt.scatter(input_raw_copy['population_2024'], range(len(input_raw_copy)))

# Добавление сетки
plt.grid(True)

# Настройка оси x с шагом 1,000,000,000
plt.xticks(range(0, int(max(input_raw_copy['population_2024'])) + 1, 100000000))

# Наименование осей и графика на русском языке
plt.xlabel('Население в 2024 году, млрд. чел.')
plt.ylabel('Ранг страны')
plt.title('Распределение населения в 2024 году')

# Отображение графика
plt.show()
```



В большинстве стран население меньше 100 млн. чел.

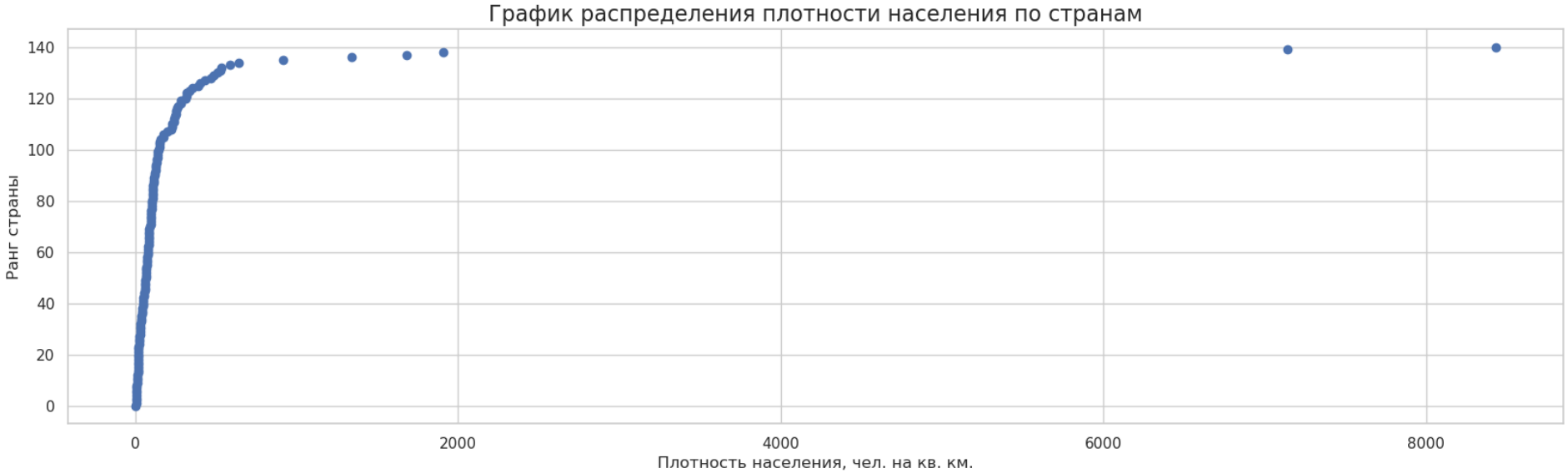
In [101…

```
input_raw_copy = input_raw.copy(deep = True)

fig, ax = plt.subplots(figsize=(b, b * 0.3))

# Сортировка
input_raw_copy = input_raw_copy.sort_values(by='population_growthRate')

# Создадим точечный график
ax.scatter(input_raw_copy['population_growthRate'], range(len(input_raw_copy)))

# Настроим оси и сетку
ax.set_xlabel('Темп роста населения')
ax.set_ylabel('Ранг страны')
ax.grid(True)

# Добавим название графика
ax.set_title('График темпа роста населения по странам')

# Отобразим график
plt.show()
```

В большинстве стран население растёт

In [101…

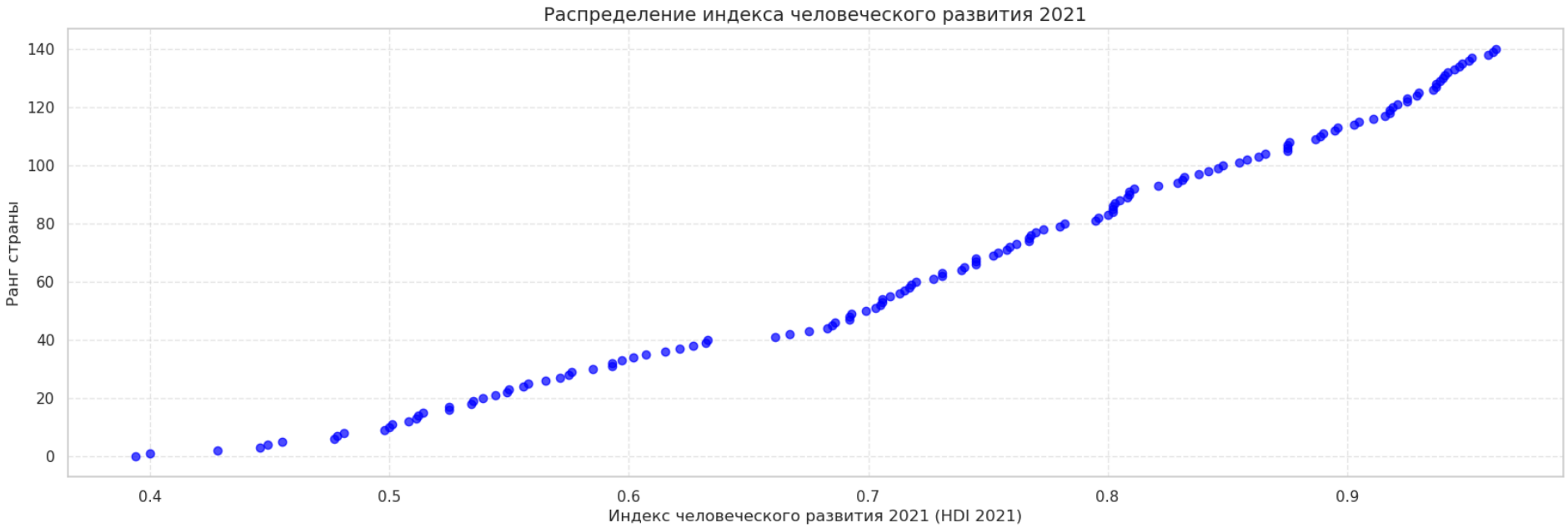
```
input_raw_copy = input_raw.copy(deep = True)

# Создаем фигуру и оси
fig, ax = plt.subplots(figsize=(b, b *0.3))

# Сортировка
input_raw_copy = input_raw_copy.sort_values(by='population_density')

# Строим точечный график
ax.scatter(input_raw_copy['population_density'], range(len(input_raw_copy)))

# Добавляем сетку
ax.grid(True)

# Устанавливаем подписи осей на русском языке
ax.set_xlabel('Плотность населения, чел. на кв. км.')
ax.set_ylabel('Ранг страны')

# Устанавливаем название графика
ax.set_title('График распределения плотности населения по странам')

# Устанавливаем размер графика
plt.gcf().subplots_adjust(bottom=0.2)

# Отображаем график
plt.show()
```



Плотность населения в большинстве стран меньше 450 человек на кв. км.

In [102…

```
input_raw_copy = input_raw.copy(deep = True)

# Создание точечного графика
plt.figure(figsize=(b, b * 0.3))  # Установка размера графика

# Сортировка
input_raw_copy = input_raw_copy.sort_values(by='Hdi2021')

# Построение точечного графика
plt.scatter(input_raw_copy['Hdi2021'], range(len(input_raw_copy)), color='blue', marker='o', alpha=0.7)

# Добавление сетки
plt.grid(True, linestyle='--', alpha=0.5)

# Настройка осей и подписей
plt.xlabel('Индекс человеческого развития 2021 (HDI 2021)', fontsize=12)
plt.ylabel('Ранг страны', fontsize=12)
plt.title('Распределение индекса человеческого развития 2021', fontsize=14)

# Показать график
plt.show()
```



Индекс человечечкого развития распределён равномерно по странам

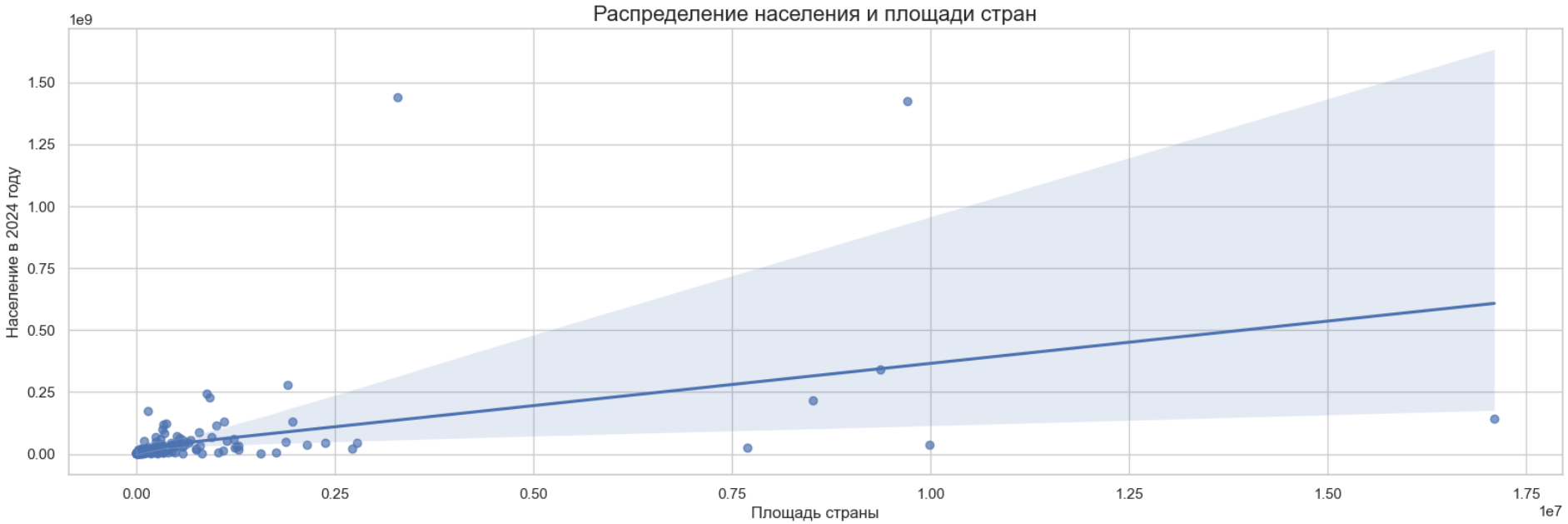
In [102…

```
input_raw_copy = input_raw.copy(deep = True)

# Сортировка
input_raw_copy = input_raw_copy.sort_values(by='WorldHappiness2022')

# Создание точечного графика
plt.figure(figsize=(b, b * 0.3))  # Размер графика (b, b * 0.3)
plt.scatter(input_raw_copy['WorldHappiness2022'], range(len(input_raw_copy)), color='blue', marker='o', alpha=0.7

# Настройка осей и сетки
plt.xlabel('Уровень счастья в 2022 году')
plt.ylabel('Ранг страны')
plt.title('Распределение WorldHappiness2022')
plt.grid(True)

# Отображение графика
plt.show()
```

Уровень счастья распределён равномерно по странам

In [102…

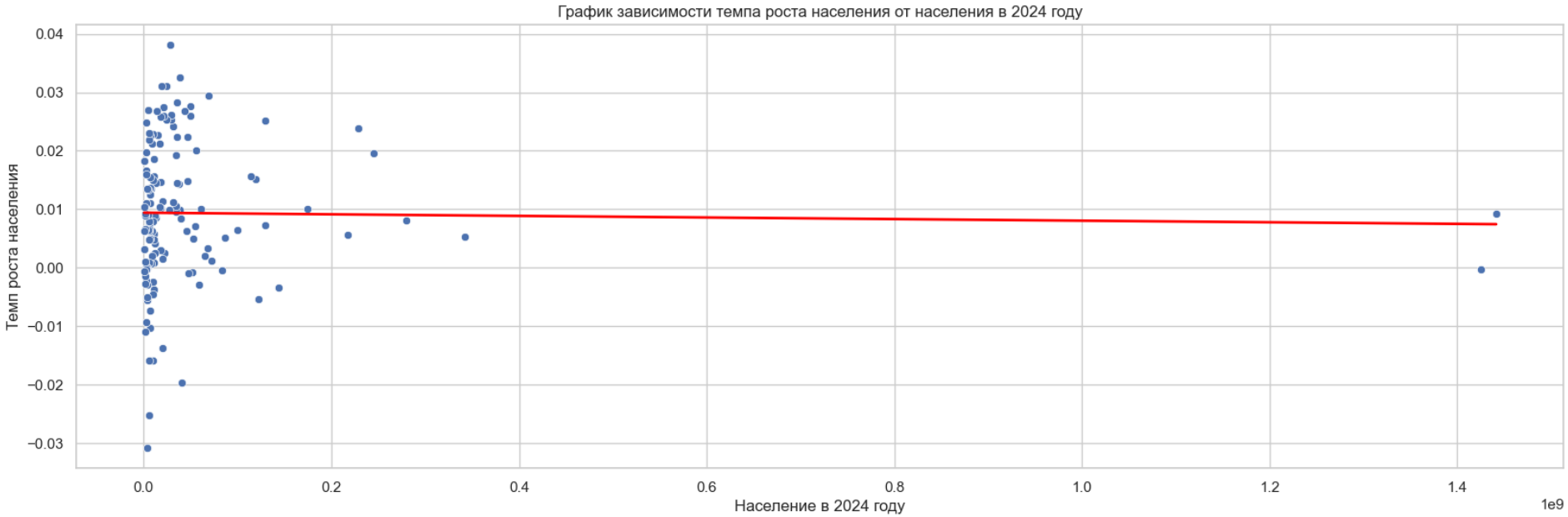
```
input_raw_copy = input_raw.copy(deep = True)

# Установка стиля seaborn для красивого оформления графиков
sns.set(style="whitegrid")

input_raw_copy = input_raw.copy(deep=True)

# Размер графика
plt.figure(figsize=(b, b * 0.3))  # Поменяли местами размеры

# Построение точечного графика с линией регрессии
sns.regplot(data=input_raw_copy, y='population_2024', x='land_area', scatter_kws={'alpha': 0.7})  # Поменяли мест

# Настройка подписей осей и заголовка
plt.title("Распределение населения и площади стран", fontsize=16)
plt.xlabel("Площадь страны", fontsize=12)  # Поменяли подпись оси x
plt.ylabel("Население в 2024 году", fontsize=12)  # Поменяли подпись оси y

# Добавление сетки
plt.grid(True)

# Отображение графика
plt.show()
```



В целом зависимость чем больше площадь страны, тем больше население

In [102…

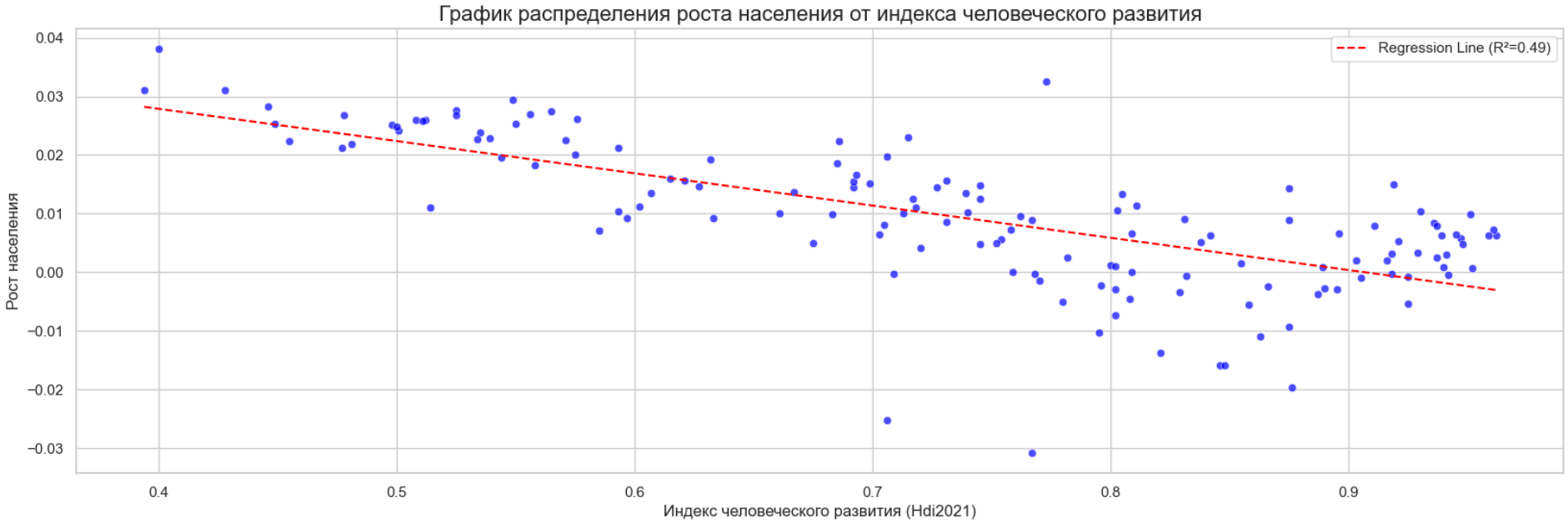
```
input_raw_copy = input_raw.copy(deep = True)

# Создаем точечный график
plt.figure(figsize=(b, b *0.3))  # Размер графика (b, b * 0.3)

# Отображаем точки
sns.scatterplot(x='population_2024', y='population_growthRate', data=input_raw_copy)

# Добавляем сетку
plt.grid(True)

# Добавляем подписи на русском языке
plt.xlabel('Население в 2024 году')
plt.ylabel('Темп роста населения')

# Добавляем линии регрессии
X = input_raw_copy['population_2024'].values.reshape(-1, 1)
y = input_raw_copy['population_growthRate'].values

regressor = LinearRegression()
regressor.fit(X, y)
y_pred = regressor.predict(X)

plt.plot(X, y_pred, color='red', linewidth=2)  # Линия регрессии

# Добавляем название графика
plt.title('График зависимости темпа роста населения от населения в 2024 году')

# Отображаем график
plt.show()
```



Темпы роста населения снижаются

In [102…

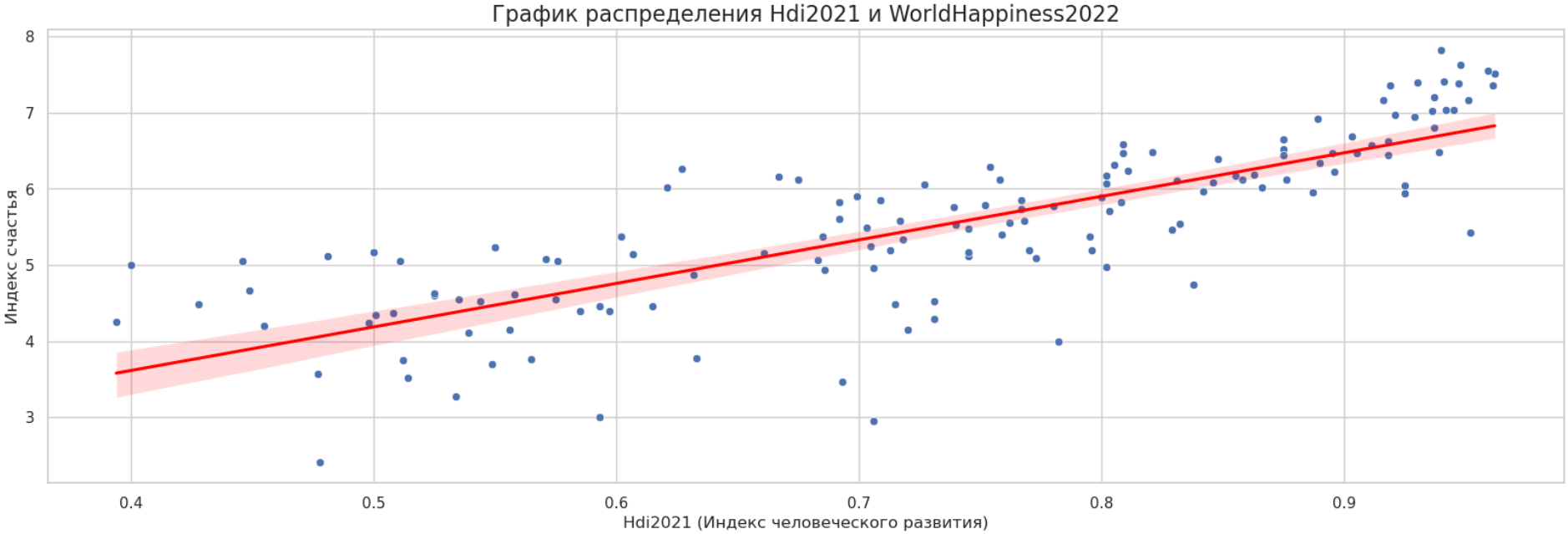
```
# Поменять местами оси x и y в input_raw_copy
input_raw_copy = input_raw.copy(deep=True)

# Установка стиля seaborn для более красивого вида графиков
sns.set(style="whitegrid")

# Создание графика
plt.figure(figsize=(b, b * 0.3))  # Замените b на ваш желаемый размер
sns.scatterplot(x='Hdi2021', y='population_growthRate', data=input_raw_copy, color='blue', alpha=0.7)

# Добавление линии регрессии
slope, intercept, r_value, p_value, std_err = linregress(input_raw_copy['Hdi2021'], input_raw_copy['population_gr
x_values = np.linspace(input_raw_copy['Hdi2021'].min(), input_raw_copy['Hdi2021'].max(), 100)
y_values = slope * x_values + intercept
plt.plot(x_values, y_values, color='red', linestyle='--', label=f'Regression Line (R²={r_value**2:.2f})')

# Добавление подписей
plt.title('График распределения роста населения от индекса человеческого развития', fontsize=16)
plt.xlabel('Индекс человеческого развития (Hdi2021)', fontsize=12)
plt.ylabel('Рост населения', fontsize=12)
plt.legend()

# Отображение сетки
plt.grid(True)

# Отображение графика
plt.show()
```



Темпы роста населения меньше в странах где высокий уровень человеческого развития

In [102…

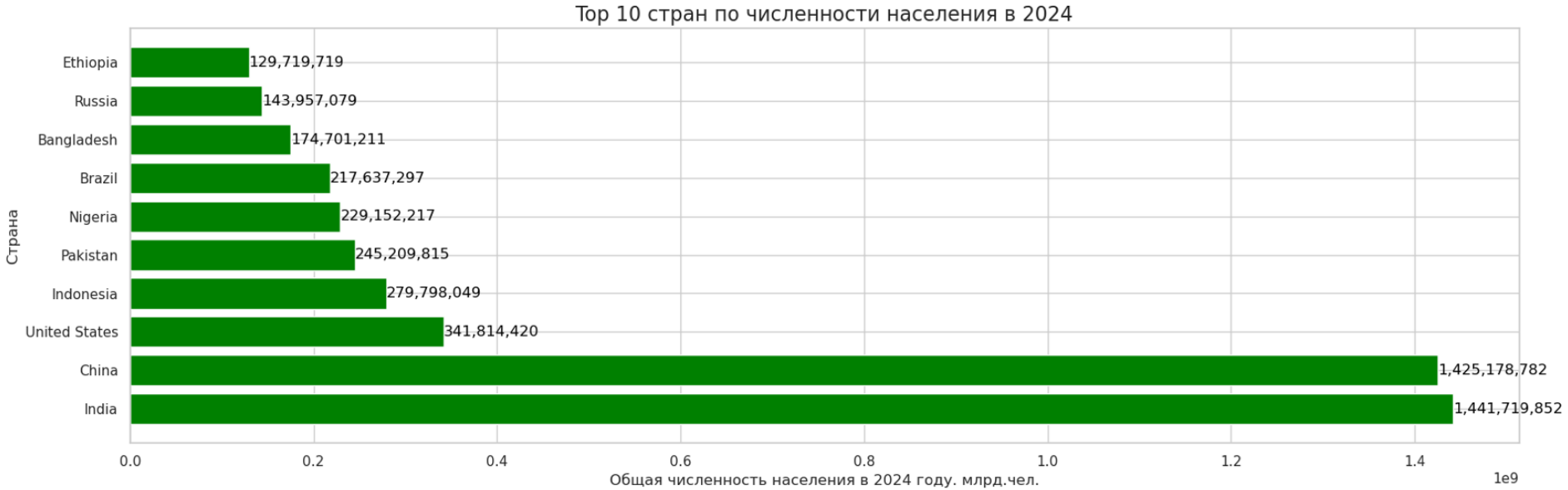
```
input_raw_copy = input_raw.copy(deep = True)

# Настройки для русских подписей
plt.rcParams['font.family'] = 'DejaVu Sans'
plt.rcParams['font.size'] = 12
plt.rcParams['axes.titlesize'] = 16

# Создаем точечный график
plt.figure(figsize=(b, b * 0.3))
sns.scatterplot(x='Hdi2021', y='WorldHappiness2022', data=input_raw_copy)

# Добавляем линию регрессии
sns.regplot(x='Hdi2021', y='WorldHappiness2022', data=input_raw_copy, scatter=False, color='red')

# Добавляем сетку и подписи
plt.grid(True)
plt.title('График распределения Hdi2021 и WorldHappiness2022')
plt.xlabel('Hdi2021 (Индекс человеческого развития)')
plt.ylabel('Индекс счастья')

# Показываем график
plt.show()
```



Чем выше индекс человекческого развития, тем выше индекс счастья

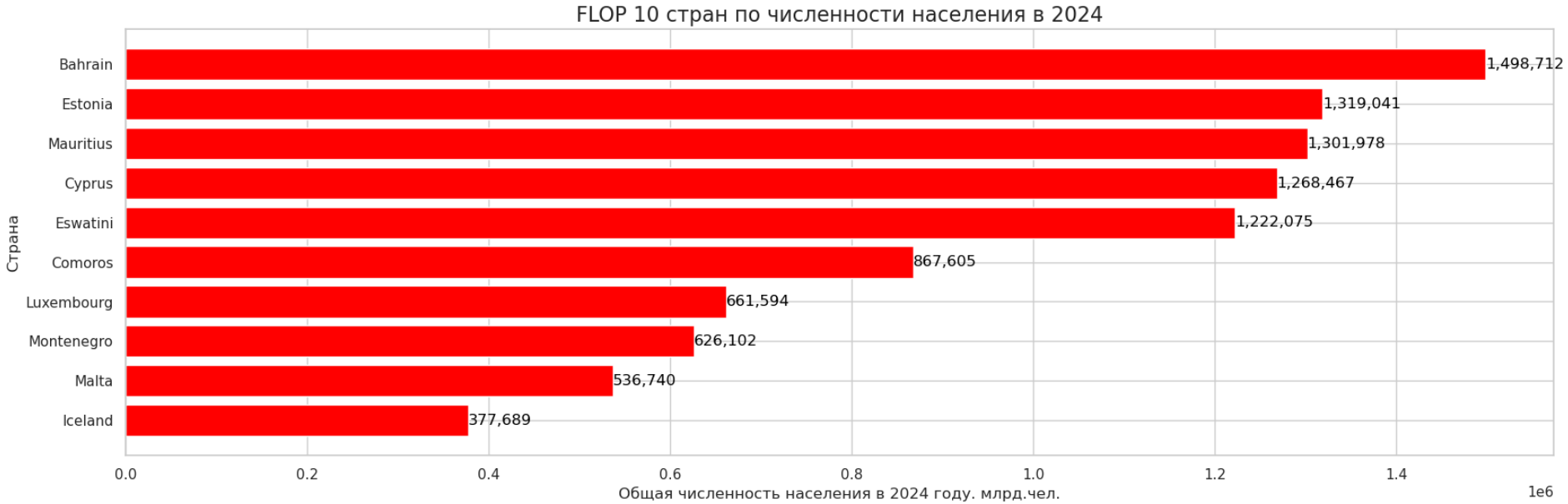
In [102…

```python
input_raw_copy = input_raw.copy(deep = True)

# Сортировка DataFrame по столбцу 'population_2024' и выбор топ-10
top_countries = input_raw_copy.sort_values(by='population_2024', ascending=False).head(10)

# Размер графика
fig, ax = plt.subplots(figsize=(b, b * 0.3))

# Построение горизонтальной столбчатой диаграммы
bars = ax.barh(top_countries['country'], top_countries['population_2024'], color='green')

# Настройка осей и меток
ax.set_xlabel('Общая численность населения в 2024 году. млрд.чел.')
ax.set_ylabel('Страна')
ax.set_title('Top 10 стран по численности населения в 2024')

# Отображение значений над столбцами
for bar in bars:
    plt.text(bar.get_width(), bar.get_y() + bar.get_height() / 2,
             f'{bar.get_width():,.0f}',
             va='center', ha='left', color='black')

plt.show()
```


Top 10 стран по численности населения в 2024
Страна
Ethiopia
Russia
Bangladesh
Brazil
Nigeria
Pakistan
Indonesia
United States
China
India
129,719,719
143,957,079
174,701,211
217,637,297
229,152,217
245,209,815
279,798,049
341,814,420
1,425,178,782
1,441,719,852
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1e9
Общая численность населения в 2024 году. млрд.чел.

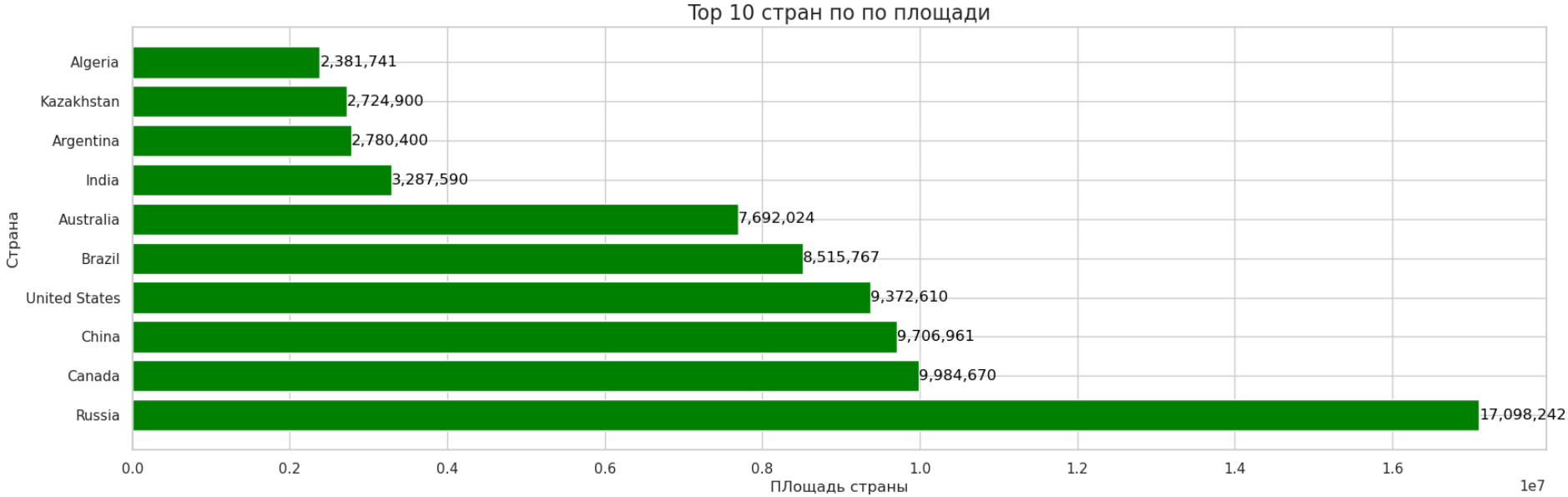
In [102…

```python
input_raw_copy = input_raw.copy(deep = True)

# Сортировка DataFrame по столбцу 'population_2024' и выбор топ-10
top_countries = input_raw_copy.sort_values(by='population_2024', ascending=True).head(10)

# Размер графика
fig, ax = plt.subplots(figsize=(b, b * 0.3))

# Построение горизонтальной столбчатой диаграммы
bars = ax.barh(top_countries['country'], top_countries['population_2024'], color='red')

# Настройка осей и меток
ax.set_xlabel('Общая численность населения в 2024 году. млрд.чел.')
ax.set_ylabel('Страна')
ax.set_title('FLOP 10 стран по численности населения в 2024')

# Отображение значений над столбцами
for bar in bars:
    plt.text(bar.get_width(), bar.get_y() + bar.get_height() / 2,
             f'{bar.get_width():,.0f}',
             va='center', ha='left', color='black')

plt.show()
```


FLOP 10 стран по численности населения в 2024
Страна
Bahrain
Estonia
Mauritius
Cyprus
Eswatini
Comoros
Luxembourg
Montenegro
Malta
Iceland
1,498,712
1,319,041
1,301,978
1,268,467
1,222,075
867,605
661,594
626,102
536,740
377,689
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1e6
Общая численность населения в 2024 году. млрд.чел.

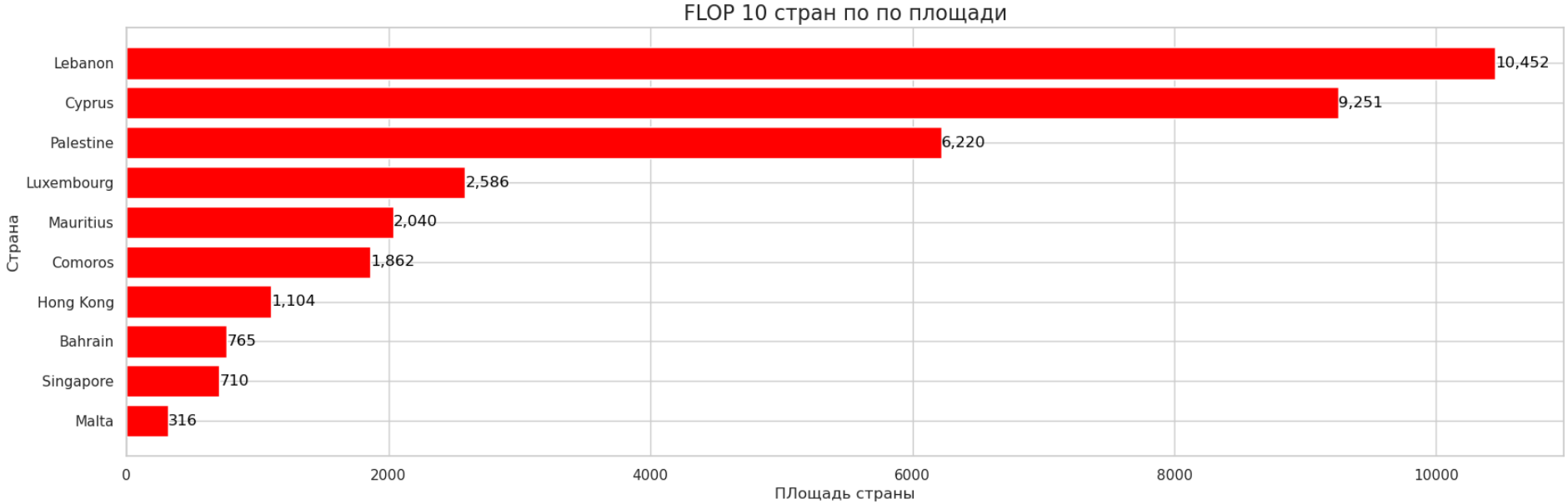
In [102…

```python
input_raw_copy = input_raw.copy(deep = True)

# Сортировка DataFrame по столбцу 'population_2024' и выбор топ-10
top_countries = input_raw_copy.sort_values(by='land_area', ascending=False).head(10)

# Размер графика
fig, ax = plt.subplots(figsize=(b, b * 0.3))

# Построение горизонтальной столбчатой диаграммы
bars = ax.barh(top_countries['country'], top_countries['land_area'], color='green')

# Настройка осей и меток
ax.set_xlabel('ПЛощадь страны')
ax.set_ylabel('Страна')
ax.set_title('Top 10 стран по по площади')

# Отображение значений над столбцами
for bar in bars:
    plt.text(bar.get_width(), bar.get_y() + bar.get_height() / 2,
             f'{bar.get_width():,.0f}',
             va='center', ha='left', color='black')

plt.show()
```


Top 10 стран по по площади
Страна
Algeria
Kazakhstan
Argentina
India
Australia
Brazil
United States
China
Canada
Russia
2,381,741
2,724,900
2,780,400
3,287,590
7,692,024
8,515,767
9,372,610
9,706,961
9,984,670
17,098,242
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
ПЛощадь страны
1e7

In [102…

```python
input_raw_copy = input_raw.copy(deep = True)

# Сортировка DataFrame по столбцу 'population_2024' и выбор топ-10
top_countries = input_raw_copy.sort_values(by='land_area', ascending=True).head(10)

# Размер графика
fig, ax = plt.subplots(figsize=(b, b * 0.3))

# Построение горизонтальной столбчатой диаграммы
bars = ax.barh(top_countries['country'], top_countries['land_area'], color='red')

# Настройка осей и меток
ax.set_xlabel('ПЛощадь страны')
ax.set_ylabel('Страна')
ax.set_title('FLOP 10 стран по по площади')

# Отображение значений над столбцами
for bar in bars:
    plt.text(bar.get_width(), bar.get_y() + bar.get_height() / 2,
             f'{bar.get_width():,.0f}',
             va='center', ha='left', color='black')

plt.show()
```



Страны которые не входят в ООО

In [103…

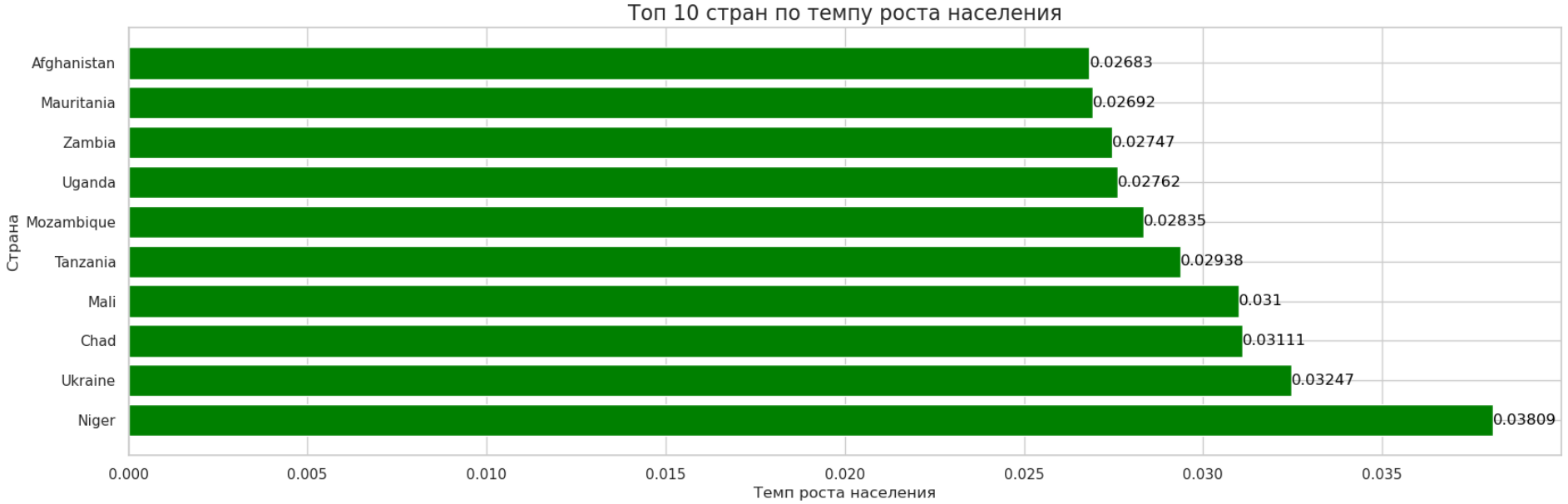
```
input_raw_copy = input_raw.copy(deep = True)
# Фильтрация стран, не входящих в ООН
non_un_members = input_raw_copy[input_raw['unMember'] == False]

# Вывод результатов
non_un_members_table = non_un_members[[
    'population_2024',
    'population_growthRate',
    'land_area',
    'country',
    'region',
    'unMember',
    'population_density',
    'population_densityMi',
    'share_borders',
    'Hdi2021',
    'Hdi2020',
    'WorldHappiness2022'
]]

non_un_members_table
```

Out[1030]:

	population_2024	population_growthRate	land_area	country	region	unMember	population_density	population_densityMi	share_
89	7496681	0.00068	1104	Hong Kong	Asia	False	7139.6962	18491.8131	

In [103…

```python
# Создайте копию датафрейма
input_raw_copy = input_raw.copy(deep=True)

# Отсортируйте датафрейм по столбцу population_growthRate и возьмите топ 10
top_10_countries = input_raw_copy.sort_values(by='population_growthRate', ascending=False).head(10)

# Создайте горизонтальную столбчатую диаграмму
fig, ax = plt.subplots(figsize=(b, b * 0.3))
bars = ax.barh(top_10_countries['country'], top_10_countries['population_growthRate'], color='green')

# Настройте подписи осей и заголовок
ax.set_xlabel('Темп роста населения')
ax.set_ylabel('Страна')
ax.set_title('Топ 10 стран по темпу роста населения')

# Добавьте значения на график
for bar in bars:
    yval = bar.get_width()  # Используйте get_width() вместо get_height()
    plt.text(yval, bar.get_y() + bar.get_height()/2, round(yval, 5), ha='left', va='center', color='black')

# Отобразите график
plt.show()
```



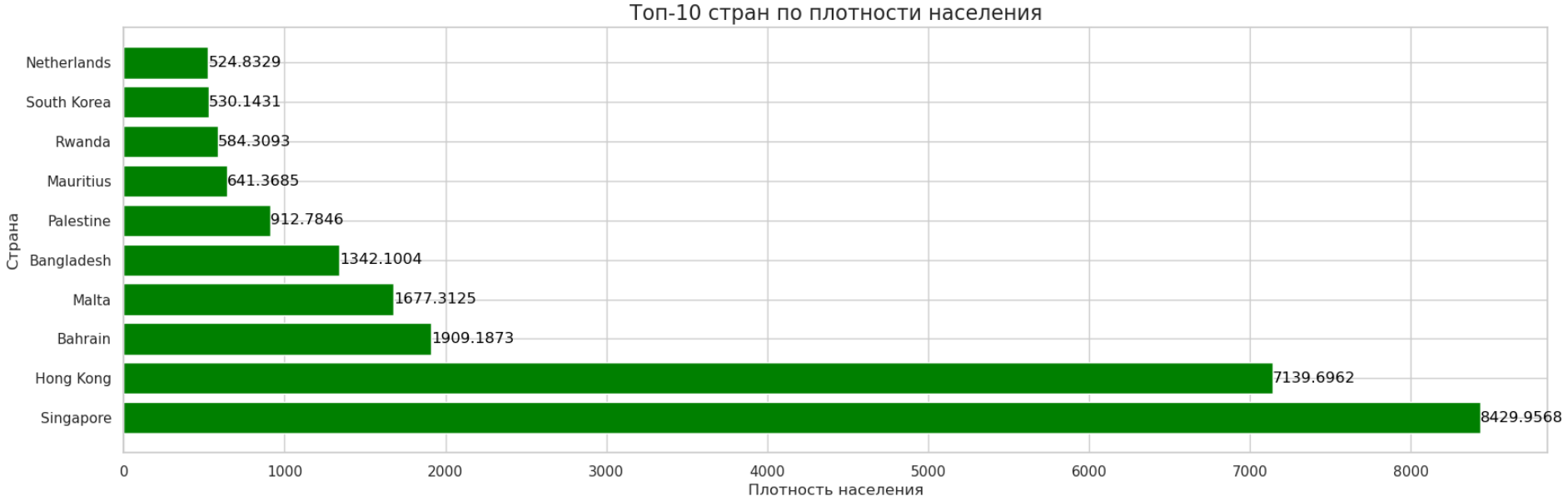
In [103…

```python
# Создайте копию датафрейма
input_raw_copy = input_raw.copy(deep=True)

# Отсортируйте датафрейм по столбцу population_growthRate и возьмите топ 10
top_10_countries = input_raw_copy.sort_values(by='population_growthRate', ascending=True).head(10)

# Создайте горизонтальную столбчатую диаграмму
fig, ax = plt.subplots(figsize=(b, b * 0.3))
bars = ax.barh(top_10_countries['country'], top_10_countries['population_growthRate'], color='red')

# Настройте подписи осей и заголовок
ax.set_xlabel('Темп убывания населения')
ax.set_ylabel('Страна')
ax.set_title('TOP 10 стран по темпу убывания населения')

# Добавьте значения на график
for bar in bars:
    yval = bar.get_width()  # Используйте get_width() вместо get_height()
    plt.text(yval, bar.get_y() + bar.get_height()/2, round(yval, 5), ha='left', va='center', color='black')

# Отобразите график
plt.show()
```


TOP 10 стран по темпу убывания населения
Страна
Serbia
Lithuania
Bulgaria
Latvia
Romania
Hungary
Slovakia
Poland
Lebanon
Moldova
-0.00728
-0.0094
-0.01033
-0.01091
-0.01377
-0.01588
-0.01594
-0.01961
-0.02519
-0.03087
−0.030
−0.025
−0.020
−0.015
−0.010
−0.005
0.000
Темп убывания населения

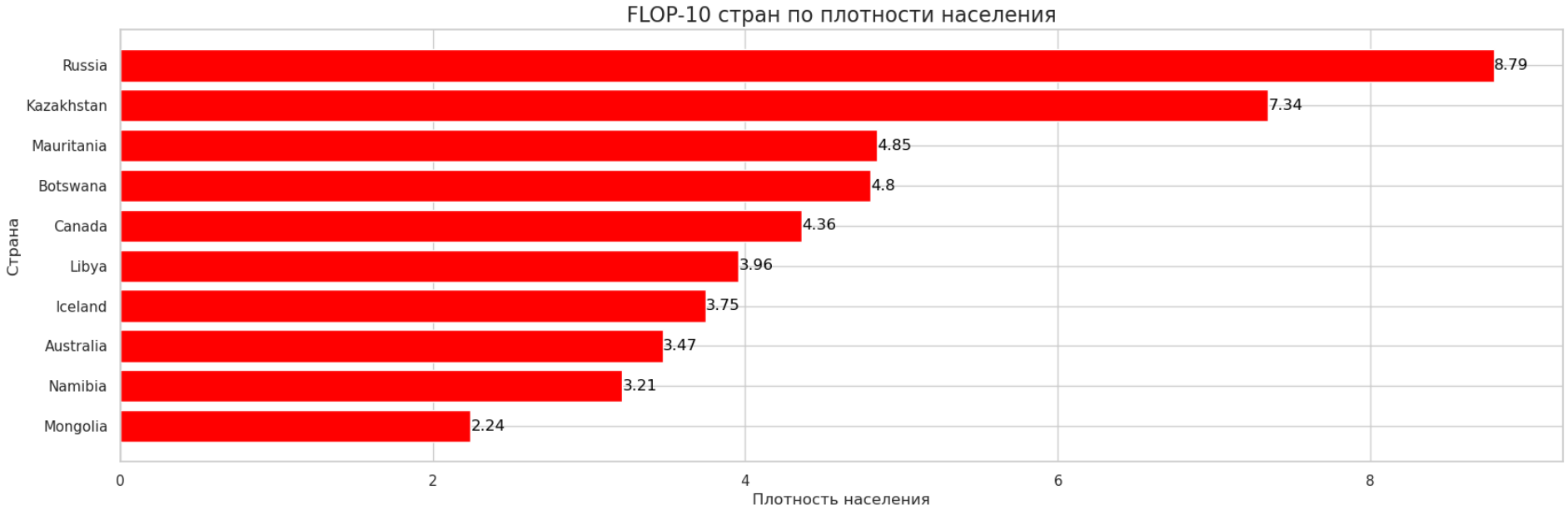
In [103…

```
input_raw_copy = input_raw.copy(deep = True)

# Сортировка по плотности населения и выбор топ-10 стран
top_10_countries = input_raw.sort_values(by='population_density', ascending=False).head(10)

# Создание графика
fig, ax = plt.subplots(figsize=(b, b * 0.3))

# Построение горизонтальной столбчатой диаграммы
bars = ax.barh(top_10_countries['country'], top_10_countries['population_density'], color='green')

# Настройка осей и заголовка
ax.set_xlabel('Плотность населения')
ax.set_ylabel('Страна')
ax.set_title('Топ-10 стран по плотности населения')

# Добавление значения над каждым столбцом
# Добавьте значения на график
for bar in bars:
    yval = bar.get_width()  # Используйте get_width() вместо get_height()
    plt.text(yval, bar.get_y() + bar.get_height()/2, round(yval, 5), ha='left', va='center', color='black')

# Отображение графика
plt.show()
```


Топ-10 стран по плотности населения
Netherlands
South Korea
Rwanda
Mauritius
Palestine
Bangladesh
Malta
Bahrain
Hong Kong
Singapore
524.8329
530.1431
584.3093
641.3685
912.7846
1342.1004
1677.3125
1909.1873
7139.6962
8429.9568
Страна
0
1000
2000
3000
4000
5000
6000
7000
8000
Плотность населения

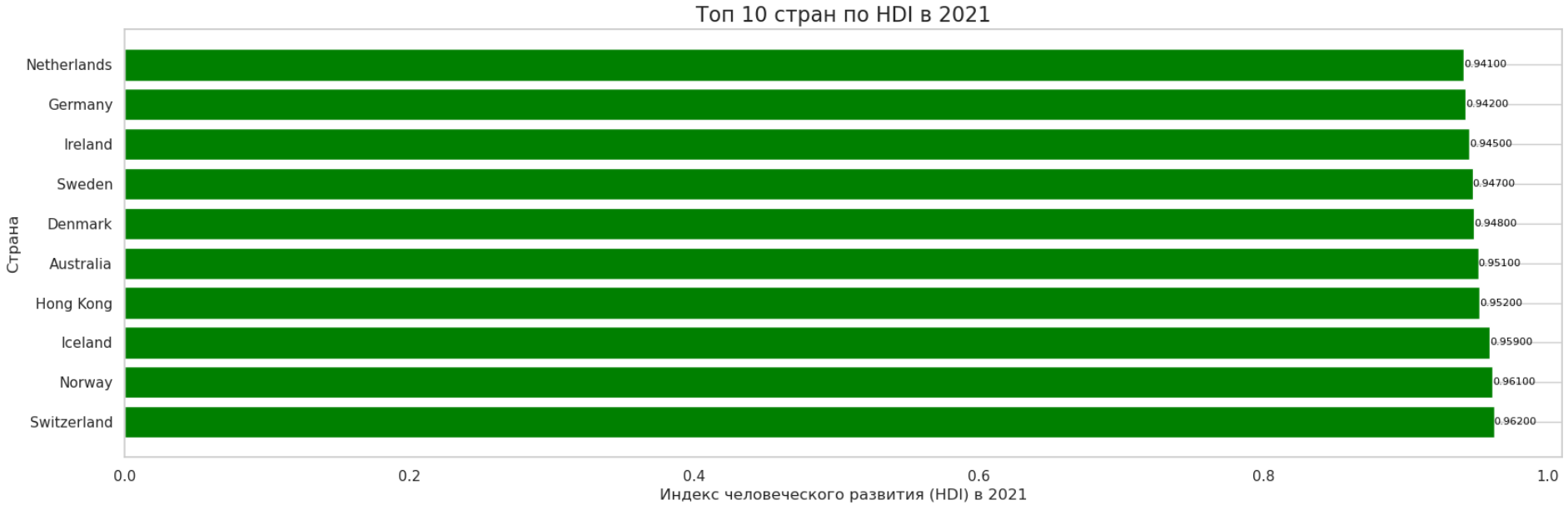
In [103…

```python
input_raw_copy = input_raw.copy(deep = True)

# Сортировка по плотности населения и выбор топ-10 стран
top_10_countries = input_raw.sort_values(by='population_density', ascending=True).head(10)

# Создание графика
fig, ax = plt.subplots(figsize=(b, b * 0.3))

# Построение горизонтальной столбчатой диаграммы
bars = ax.barh(top_10_countries['country'], top_10_countries['population_density'], color='red')

# Настройка осей и заголовка
ax.set_xlabel('Плотность населения')
ax.set_ylabel('Страна')
ax.set_title('FLOP-10 стран по плотности населения')

# Добавление значения над каждым столбцом
for bar in bars:
    yval = bar.get_width()
    plt.text(yval, bar.get_y() + bar.get_height()/2, round(yval, 2), va='center', ha='left', color='black')

# Отображение графика
plt.show()
```



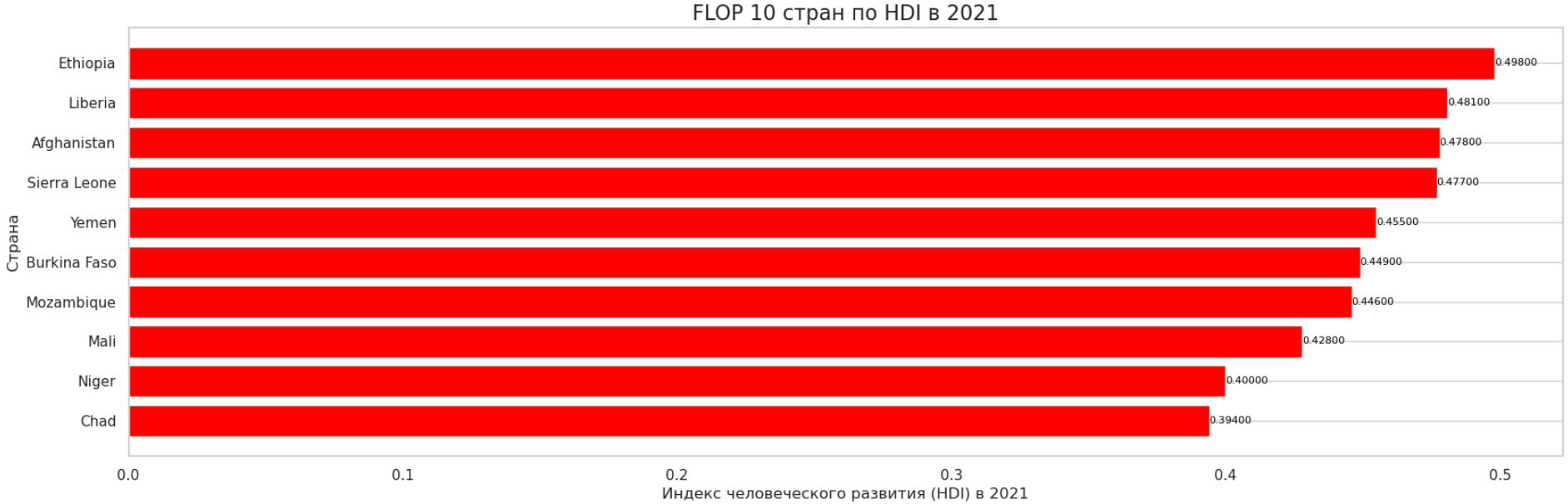
In [103…

```
input_raw_copy = input_raw.copy(deep = True)


# Сортировка датафрейма по столбцу Hdi2021 и выбор топ 10
top_countries = input_raw_copy.sort_values(by='Hdi2021', ascending=False).head(10)

# Создание горизонтальной столбчатой диаграммы
plt.figure(figsize=(b, b * 0.3))
bars = plt.barh(top_countries['country'], top_countries['Hdi2021'], color='green')
plt.xlabel('Индекс человеческого развития (HDI) в 2021')
plt.ylabel('Страна')
plt.title('Топ 10 стран по HDI в 2021')
plt.grid(axis='x')

# Добавление значений на график
for bar, value in zip(bars, top_countries['Hdi2021']):
    plt.text(bar.get_width(), bar.get_y() + bar.get_height() / 2, f'{value:.5f}',
             va='center', ha='left', fontsize=8, color='black')

# Показать график
plt.show()
```



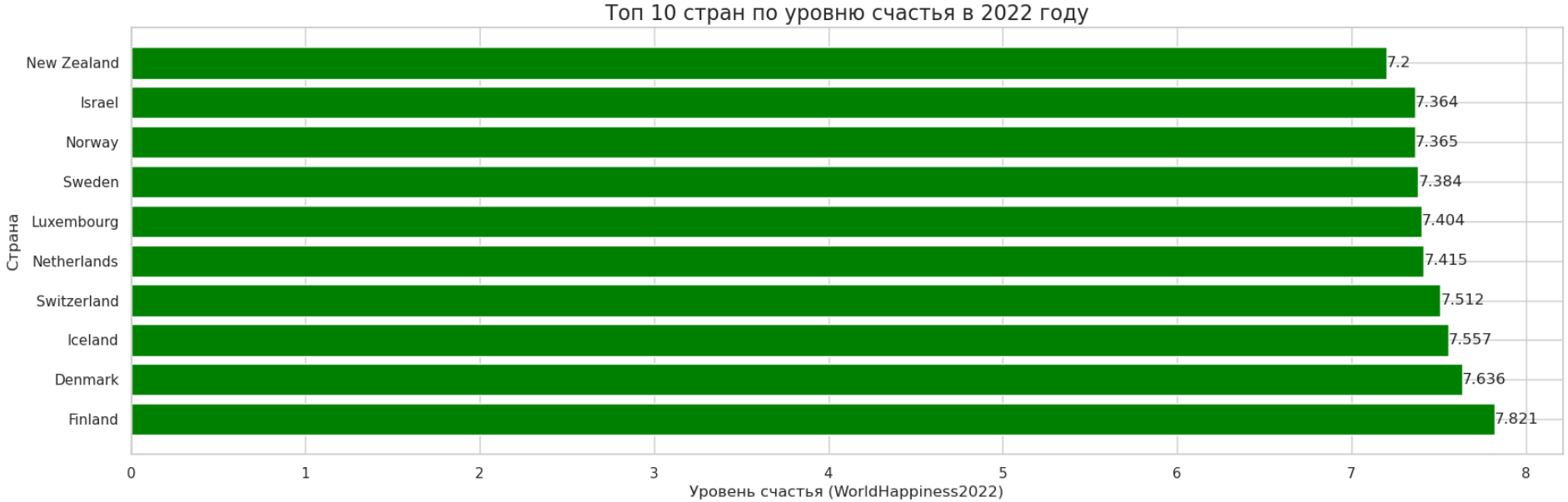
In [103…

```python
input_raw_copy = input_raw.copy(deep = True)

# Сортировка датафрейма по столбцу Hdi2021 и выбор топ 10
top_countries = input_raw_copy.sort_values(by='Hdi2021', ascending=True).head(10)

# Создание горизонтальной столбчатой диаграммы
plt.figure(figsize=(b, b * 0.3))  # Размер графика (8, 2.4), замените на ваш b и b * 0.3
bars = plt.barh(top_countries['country'], top_countries['Hdi2021'], color='red')
plt.xlabel('Индекс человеческого развития (HDI) в 2021')
plt.ylabel('Страна')
plt.title('FLOP 10 стран по HDI в 2021')
plt.grid(axis='x')

# Добавление значений на график
for bar, value in zip(bars, top_countries['Hdi2021']):
    plt.text(bar.get_width(), bar.get_y() + bar.get_height() / 2, f'{value:.5f}',
             va='center', ha='left', fontsize=8, color='black')

# Показать график
plt.show()
```


FLOP 10 стран по HDI в 2021
Страна
Ethiopia
Liberia
Afghanistan
Sierra Leone
Yemen
Burkina Faso
Mozambique
Mali
Niger
Chad
0.49800
0.48100
0.47800
0.47700
0.45500
0.44900
0.44600
0.42800
0.40000
0.39400
0.0
0.1
0.2
0.3
0.4
0.5
Индекс человеческого развития (HDI) в 2021

In [103…

```
input_raw_copy = input_raw.copy(deep = True)

# Сортировка по столбцу WorldHappiness2022 и выбор топ 10
top_countries = input_raw_copy.sort_values(by='WorldHappiness2022', ascending=False).head(10)

# Установка размеров графика
b = 10  # Ширина графика
fig, ax = plt.subplots(figsize=(b *2, b * 0.6))

# Построение горизонтальной столбчатой диаграммы
bars = ax.barh(top_countries['country'], top_countries['WorldHappiness2022'], color='green')

# Настройка осей и заголовка
ax.set_xlabel('Уровень счастья (WorldHappiness2022)')
ax.set_ylabel('Страна')
ax.set_title('Топ 10 стран по уровню счастья в 2022 году')

# Добавление значения на каждом столбце
for bar in bars:
    xval = bar.get_width()
    ax.text(xval, bar.get_y() + bar.get_height()/2, round(xval, 5), va='center', ha='left', fontsize=12)

# Отображение графика
plt.show()
```


Топ 10 стран по уровню счастья в 2022 году
New Zealand
Israel
Norway
Sweden
Luxembourg
Netherlands
Switzerland
Iceland
Denmark
Finland
7.2
7.364
7.365
7.384
7.404
7.415
7.512
7.557
7.636
7.821
Страна
0
1
2
3
4
5
6
7
8
Уровень счастья (WorldHappiness2022)

In [103…

```
input_raw_copy = input_raw.copy(deep = True)

# Сортировка по столбцу WorldHappiness2022 и выбор топ 10
top_countries = input_raw_copy.sort_values(by='WorldHappiness2022', ascending=True).head(10)

# Установка размеров графика
b = 10  # Ширина графика
fig, ax = plt.subplots(figsize=(b *2, b * 0.6))

# Построение горизонтальной столбчатой диаграммы
bars = ax.barh(top_countries['country'], top_countries['WorldHappiness2022'], color='red')

# Настройка осей и заголовка
ax.set_xlabel('Уровень счастья (WorldHappiness2022)')
ax.set_ylabel('Страна')
ax.set_title('FLOP 10 стран по уровню счастья в 2022 году')

# Добавление значения на каждом столбце
for bar in bars:
    xval = bar.get_width()
    ax.text(xval, bar.get_y() + bar.get_height()/2, round(xval, 5), va='center', ha='left', fontsize=12)

# Отображение графика
plt.show()
```

In [103…

```python
input_raw_copy = input_raw.copy(deep = True)

fig = px.choropleth(
    input_raw_copy,
    locations="country",
    locationmode="country names",
    color="WorldHappiness2022",
    hover_name="country",
    title="World Happiness 2022",
    color_continuous_scale="RdYlGn",  # Выберите нужную цветовую схему
    projection="natural earth"
)

fig.update_layout(
    height=600,  # Указываете желаемую высоту графика
    width=1000    # Указываете желаемую ширину графика
)

fig.show()
```