1. Data Importation and Cleaning

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from scipy.stats import pearsonr
import matplotlib
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
plt.style.use("ggplot")
import datetime as dt
import plotly.express as px
import plotly.graph_objects as go
world = pd.read_csv("C:\\Users\\njoku\\Documents\\DATA_ANALYSIS_CLASS\PYTHON_CLASS\\Pand
world
```

Out[1]:

	Country	Density\n(P/Km2)	Abbreviation	Agricultural Land(%)	Land Area(Km2)	Armed Forces size	Birth Rate	Calling Code	Capital/Major City
0	Afghanistan	60	AF	58.10%	652,230	323,000	32.49	93.0	Kabul
1	Albania	105	AL	43.10%	28,748	9,000	11.78	355.0	Tirana
2	Algeria	18	DZ	17.40%	2,381,741	317,000	24.28	213.0	Algiers
3	Andorra	164	AD	40.00%	468	NaN	7.20	376.0	Andorra la Vella
4	Angola	26	AO	47.50%	1,246,700	117,000	40.73	244.0	Luanda
...	...	...	...	...	...	...	...	...	...
190	Venezuela	32	VE	24.50%	912,050	343,000	17.88	58.0	Caracas
191	Vietnam	314	VN	39.30%	331,210	522,000	16.75	84.0	Hanoi
192	Yemen	56	YE	44.60%	527,968	40,000	30.45	967.0	Sanaa
193	Zambia	25	ZM	32.10%	752,618	16,000	36.19	260.0	Lusaka
194	Zimbabwe	38	ZW	41.90%	390,757	51,000	30.68	263.0	Harare

195 rows × 35 columns

In [2]:
```
world = world.rename({'Density\n(P/Km2)':'Density',
                      'Agricultural Land( %)': 'Agricultural_Land( %)',
                      'Land Area(Km2)': 'Land_Area(Km2)',
                      'Armed Forces size': 'Armed_Forces_size',
                      'Birth Rate': 'Birth_Rate',
                      'Calling Code': 'Calling_Code',
                      'Capital/Major City': 'Capital_Major_City',
                      'Co2-Emissions': 'Co2_Emissions',
                      'CPI Change (%) ': 'CPI_Change(%)',
                      'Currency-Code': 'Currency_Code',
                      'Fertility Rate': 'Fertility_Rate',
                      'Forested Area (%)': 'Forested_Area(%)',
                      'Gasoline Price': 'Gasoline_Price',
                      'Gross primary education enrollment (%)': 'Gross_primary_education
                     'Gross tertiary education enrollment (%)': 'Gross_tertiary_educatio
```

```
                          'Infant mortality':'Infant_mortality',
                          'Largest city':'Largest_city',
                          'Life expectancy':'Life_expectancy',
                          'Maternal mortality ratio':'Maternal_mortality_ratio',
                          'Minimum wage':'Minimum_wage',
                          'Official language':'Official_language',
                          'Out of pocket health expenditure':'Out_of_pocket_health_expenditur
                          'Physicians per thousand':'Physicians_per_thousand',
                          'Population: Labor force participation (%)':'Population_Labor_force
                          'Tax revenue (%)':'Tax_revenue(%)',
                          'Total tax rate':'Total_tax_rate',
                          'Unemployment rate':'Unemployment_rate',}, axis = 'columns')
```

In [3]:

```
world = world.replace({'%':'', ',':'', '$':''},regex = True)
```

In [4]:

```
world['Gasoline_Price'] = world['Gasoline_Price'].str.replace('$','')
world['GDP'] = world['GDP'].str.replace('$','')
world['Minimum_wage'] = world['Minimum_wage'].str.replace('$','')
```

```
C:\Users\njoku\AppData\Local\Temp\ipykernel_27324\3044282255.py:1: FutureWarning: The de
fault value of regex will change from True to False in a future version. In addition, si
ngle character regular expressions will *not* be treated as literal strings when regex=T
rue.
  world['Gasoline_Price'] = world['Gasoline_Price'].str.replace('$','')
C:\Users\njoku\AppData\Local\Temp\ipykernel_27324\3044282255.py:2: FutureWarning: The de
fault value of regex will change from True to False in a future version. In addition, si
ngle character regular expressions will *not* be treated as literal strings when regex=T
rue.
  world['GDP'] = world['GDP'].str.replace('$','')
C:\Users\njoku\AppData\Local\Temp\ipykernel_27324\3044282255.py:3: FutureWarning: The de
fault value of regex will change from True to False in a future version. In addition, si
ngle character regular expressions will *not* be treated as literal strings when regex=T
rue.
  world['Minimum_wage'] = world['Minimum_wage'].str.replace('$','')
```

In [5]:

```
pd.set_option('display.max.columns', None)
world
```

Out[5]:

	Country	Density	Abbreviation	Agricultural_Land(%)	Land_Area(Km2)	Armed_Forces_size	Birth_Rate	Callir
0	Afghanistan	60	AF	58.10	652230	323000	32.49	
1	Albania	105	AL	43.10	28748	9000	11.78	
2	Algeria	18	DZ	17.40	2381741	317000	24.28	
3	Andorra	164	AD	40.00	468	NaN	7.20	
4	Angola	26	AO	47.50	1246700	117000	40.73	
...	...	...	...	...	...	...	...	
190	Venezuela	32	VE	24.50	912050	343000	17.88	
191	Vietnam	314	VN	39.30	331210	522000	16.75	
192	Yemen	56	YE	44.60	527968	40000	30.45	
193	Zambia	25	ZM	32.10	752618	16000	36.19	
194	Zimbabwe	38	ZW	41.90	390757	51000	30.68	

195 rows × 35 columns

In [6]:
```
world['Density'] = pd.to_numeric(world['Density'])
world['Agricultural_Land( %)'] = pd.to_numeric(world['Agricultural_Land( %)'])
world['Land_Area(Km2)'] = pd.to_numeric(world['Land_Area(Km2)'])
world['Armed_Forces_size'] = pd.to_numeric(world['Armed_Forces_size'])
world['Birth_Rate'] = pd.to_numeric(world['Birth_Rate'])
world['Calling_Code'] = pd.to_numeric(world['Calling_Code'])
world['Co2_Emissions'] = pd.to_numeric(world['Co2_Emissions'])
world['CPI'] = pd.to_numeric(world['CPI'])
world['CPI Change (%)'] = pd.to_numeric(world['CPI Change (%)'])
world['Fertility_Rate'] = pd.to_numeric(world['Fertility_Rate'])
world['Forested_Area(%)'] = pd.to_numeric(world['Forested_Area(%)'])
world['Gasoline_Price'] = pd.to_numeric(world['Gasoline_Price'])
world['GDP'] = pd.to_numeric(world['GDP'])
world['Gross_primary_education_enrollment(%)'] = pd.to_numeric(world['Gross_primary_educ
world['Gross_tertiary_education_enrollment(%)'] = pd.to_numeric(world['Gross_tertiary_ed
world['Infant_mortality'] = pd.to_numeric(world['Infant_mortality'])
world['Life_expectancy'] = pd.to_numeric(world['Life_expectancy'])
world['Maternal_mortality_ratio'] = pd.to_numeric(world['Maternal_mortality_ratio'])
world['Minimum_wage'] = pd.to_numeric(world['Minimum_wage'])
world['Out_of_pocket_health_expenditure'] = pd.to_numeric(world['Out_of_pocket_health_ex
world['Physicians_per_thousand'] = pd.to_numeric(world['Physicians_per_thousand'])
world['Population'] = pd.to_numeric(world['Population'])
world['Population_Labor_force_participation(%)'] = pd.to_numeric(world['Population_Labor
world['Tax_revenue(%)'] = pd.to_numeric(world['Tax_revenue(%)'])
world['Total_tax_rate'] = pd.to_numeric(world['Total_tax_rate'])
world['Unemployment_rate'] = pd.to_numeric(world['Unemployment_rate'])
world['Urban_population'] = pd.to_numeric(world['Urban_population'])
world['Longitude'] = pd.to_numeric(world['Longitude'])
```

In [7]:
```
world['Density']=world['Density'].fillna(356.76)
world['Agricultural_Land( %)']=world['Agricultural_Land( %)'].fillna(39.11)
world['Land_Area(Km2)']=world['Land_Area(Km2)'].fillna(6.89)
world['Armed_Forces_size']=world['Armed_Forces_size'].fillna(6.89)
world['Birth_Rate']=world['Birth_Rate'].fillna(20.21)
world['Calling_Code']=world['Calling_Code'].fillna(360.54)
world['Co2_Emissions']=world['Co2_Emissions'].fillna(1.77)
world['CPI']=world['CPI'].fillna(190.46)
world['CPI Change (%)']=world['CPI Change (%)'].fillna(6.72)
world['Fertility_Rate']=world['Fertility_Rate'].fillna(2.69)
world['Forested_Area(%)']=world['Forested_Area(%)'].fillna(32.01)
world['Gasoline_Price']=world['Gasoline_Price'].fillna(32.01)
world['GDP']=world['GDP'].fillna(4.77)
world['Gross_primary_education_enrollment(%)']=world['Gross_primary_education_enrollment
world['Gross_tertiary_education_enrollment(%)']=world['Gross_tertiary_education_enrollme
world['Infant_mortality']=world['Infant_mortality'].fillna(21.33)
world['Life_expectancy']=world['Life_expectancy'].fillna(72.27)
world['Maternal_mortality_ratio']=world['Maternal_mortality_ratio'].fillna(160.39)
world['Minimum_wage']=world['Minimum_wage'].fillna(4.77)
world['Out_of_pocket_health_expenditure']=world['Out_of_pocket_health_expenditure'].fill
world['Physicians_per_thousand']=world['Physicians_per_thousand'].fillna(1.83)
world['Population']=world['Population'].fillna(3.93)
world['Population_Labor_force_participation(%)']=world['Population_Labor_force_participa
world['Tax_revenue(%)']=world['Tax_revenue(%)'].fillna(16.57)
world['Total_tax_rate']=world['Total_tax_rate'].fillna(40.82)
world['Unemployment_rate']=world['Unemployment_rate'].fillna(6.88)
world['Urban_population']=world['Urban_population'].fillna(2.23)
world['Latitude']=world['Latitude'].fillna(19.09)
world['Longitude']=world['Longitude'].fillna(20.23)
```

In [8]:
```
world.loc[world['Country'] == 'Republic of the Congo', 'Abbreviation'] = 'RC'
world.loc[world['Country'] == 'Eswatini', 'Abbreviation'] = 'SZ'
world.loc[world['Country'] == 'Vatican City', 'Abbreviation'] = 'VA'
world.loc[world['Country'] == 'Republic of Ireland', 'Abbreviation'] = 'IS'
```

```
world.loc[world['Country'] == 'Namibia', 'Abbreviation'] = 'NA'
world.loc[world['Country'] == 'North Macedonia', 'Abbreviation'] = 'MK'
world.loc[world['Country'] == 'Palestinian National Authority', 'Abbreviation'] = 'PA'
```

In [9]:
```
world.loc[150, 'Official_language'] = 'English'
```

In [10]:
```
world.loc[11, 'Currency_Code'] = 'BSD'
world.loc[19, 'Currency_Code'] = 'BTN'
world.loc[30, 'Currency_Code'] = 'KHR'
world.loc[33, 'Currency_Code'] = 'XAF'
world.loc[52, 'Currency_Code'] = 'USD'
world.loc[56, 'Currency_Code'] = 'SZL'
world.loc[85, 'Currency_Code'] = 'JPY'
world.loc[95, 'Currency_Code'] = 'LSL'
world.loc[96, 'Currency_Code'] = 'LRD'
world.loc[104, 'Currency_Code'] = 'MVR'
world.loc[119, 'Currency_Code'] = 'NAD'
world.loc[122, 'Currency_Code'] = 'EUR'
world.loc[133, 'Currency_Code'] = 'ILS'
world.loc[134, 'Currency_Code'] = 'PAB'
world.loc[194, 'Currency_Code'] = 'ZWL'
```

In [11]:
```
world.loc[97, 'Capital_Major_City'] = 'Tripoli'
world.loc[133, 'Capital_Major_City'] = 'Ramallah'
world.loc[156, 'Capital_Major_City'] = 'Singapore'
```

In [12]:
```
world.loc[24, 'Largest_city'] = 'Bandar Seri Begawan'
world.loc[43, 'Largest_city'] = 'Nicosia'
world.loc[73, 'Largest_city'] = 'Vatican City'
world.loc[97, 'Largest_city'] = 'Tripoli'
world.loc[120, 'Largest_city'] = 'Yaren'
world.loc[133, 'Largest_city'] = 'Gaza'
world.loc[156, 'Largest_city'] = 'Singapore'
world.loc[168, 'Largest_city'] = 'Stockholm'
world.loc[169, 'Largest_city'] = 'Zurich'
```

In [13]:
```
world = world.replace({'S�����������':'South Africa'},regex = True)
world = world.replace({'Yaound�':'Yaounde'},regex = True)
world = world.replace({'Lom�':'Lome'},regex = True)
world = world.replace({'Chi����':'Chisinau'},regex = True)
world = world.replace({'Mal�':'Male'},regex = True)
world = world.replace({'Bras���':'Brasilia'},regex = True)
world = world.replace({'Bogot�':'Bogota'},regex = True)
world = world.replace({'San Jos������':'San Jose'},regex = True)
world = world.replace({'Reykjav��':'Reykjavik'},regex = True)
world = world.replace({'Asunci��':'Asuncion'},regex = True)
world = world.replace({'Nuku����':'Nukualofa'},regex = True)
world = world.replace({'NaN':'Tripoli'},regex = True)
world = world.replace({'NaN':'Ramallah'},regex = True)
world = world.replace({'NaN':'Singapore'},regex = True)
world = world.replace({'S����':'São Tomé'},regex = True)
world = world.replace({'NaN':'United Arab Emirates'}, regex = True)
```

In [ ]:

2. INTRODUCTION

In this dataset, diverse range of information pertaining to 195 countries were presented. This collection of data encompasses various attributes, statistics, and metrics that shed light on the socio-economic, demographic, and geographic aspects of each country. From land area and population to economic indicators and healthcare metrics, this dataset offers a rich tapestry of insights for anyone interested in understanding the global landscape.As you navigate through this dataset, you'll uncover valuable insights that can inform research, decision-making, and policy formulation across various domains. The data provided here can facilitate comprehensive analyses and drive discussions on topics ranging from sustainable development to geopolitical dynamics.

Let this dataset be a compass that guides you through the complexities of our diverse world, allowing you to explore, learn, and contribute to the global conversation.

3. OBJECTIVES

The dataset containing information about 195 countries offers a world of possibilities for exploration and discovery. As someone who thrives on data-driven insights, I see the potential to uncover a multitude of valuable perspectives by delving into this dataset. By immersing myself in its depths, I can envision achieving the following objectives:

1. Understanding Our Economies: By examining attributes like GDP, taxes, and wages, I aim to decode the economic tapestry of different countries. My goal is to uncover hidden patterns, understand how wealth is shared, and pinpoint factors that drive economic growth.

2. People at the Heart: I am eager to dive into birth rates, life expectancies, and more, to truly grasp the diverse dynamics of human populations. By connecting these dots, I strive to unravel the intricate relationship between healthcare spending, education, and people's well-being.

3. Planet and Progress: The dataset provides a chance to gauge the environmental impact of nations through CO_2 emissions, forests, and energy costs. My objective is to analyze trends that reveal how countries are contributing to sustainability and how they can do better.

4. Health Insights: Through the lens of healthcare metrics, I want to unravel the story of each country's well-being. By dissecting data on doctors, maternal health, and healthcare spending, I aim to highlight disparities, emphasize quality, and recommend improvements.

5. Mapping the World: I intend to chart the geographical trends of urbanization and population distribution. By studying locations and densities, I hope to paint a vivid picture of where societies are heading and where opportunities lie.

6. Stability and Society: By analyzing metrics such as corruption perceptions, employment, and military presence, I aim to understand the foundations of each country's stability and prosperity, and how these factors are intertwined.

7. Educational Journeys: My objective is to journey through education data to uncover opportunities and challenges in different countries. By connecting education with economic growth, I seek to illuminate paths for improvement.

8. Workforce Dynamics: Through the lens of labor participation, unemployment, and taxation, I aspire to unravel the intricate story of livelihoods. By identifying patterns, I want to influence policies that empower workforces.

9. Cultural Kaleidoscope: I am excited to explore the diverse linguistic and cultural landscapes by studying official languages and capital cities. I believe these insights can lead to a deeper understanding of societies' uniqueness.

10. Learning from Each Other: Through comparative analysis, I envision showcasing success stories, highlighting challenges, and suggesting strategies for growth. By facilitating knowledge sharing, I hope to contribute to a more connected global society.

4. Data Overview and Analysis

Here are some insight questions you could explore using the dataset. Also, I have classified the dataset into 10 sections to gain insights from

1. Population Analysis

In [14]:
```
## Question 1: Which countries have the highest and lowest populations?
world_sorted = world.sort_values(by='Population', ascending=False)
```

In [15]:
```
highest_population_country = world_sorted.iloc[0]['Country']
highest_population_country
```

Out[15]:
```
'China'
```

In [16]:
```
lowest_population_country = world_sorted.iloc[-1]['Country']
lowest_population_country
```

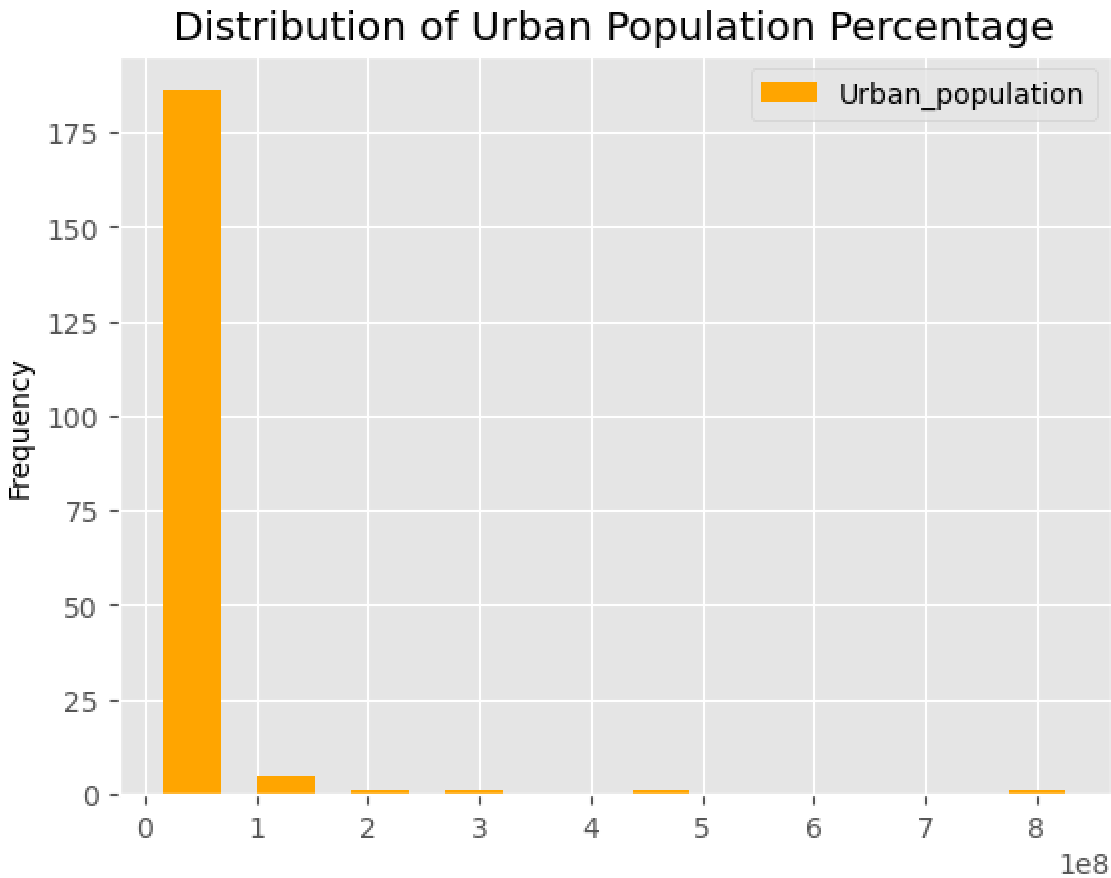
Out[16]:
```
'Palestinian National Authority'
```

In [17]:
```
## Question 2: Is there a correlation between population and density?
correlation = world['Population'].corr(world['Density'])
correlation
```

Out[17]:
```
-0.017946157448559417
```

In [18]:
```
## Question 3: How does the urban population percentage vary across countries?
world.plot(kind='hist', y='Urban_population', x='Country', color='orange', title='Distri
```

Out[18]:
```
<Axes: title={'center': 'Distribution of Urban Population Percentage'}, ylabel='Frequenc
y'>
```

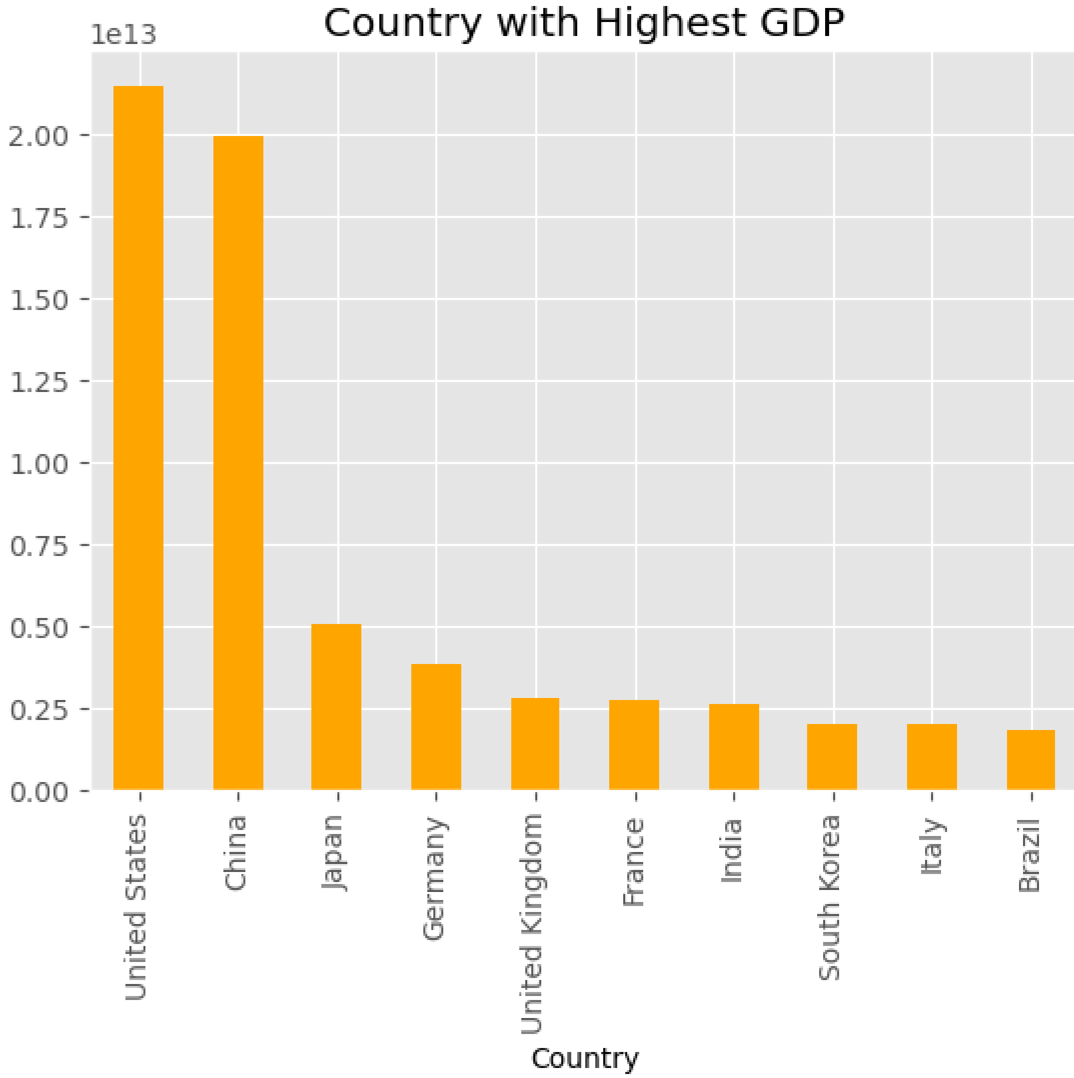


2. Economic Analysis

In [19]:
```
## Question 4: Which countries have the highest GDP values?
highest=world.groupby ('Country')['GDP'].mean().sort_values(ascending=False).head(10)
```

In [20]:
```
highest.plot(kind ='bar', y='GDP', x='Country', color='orange', title='Country with High
```

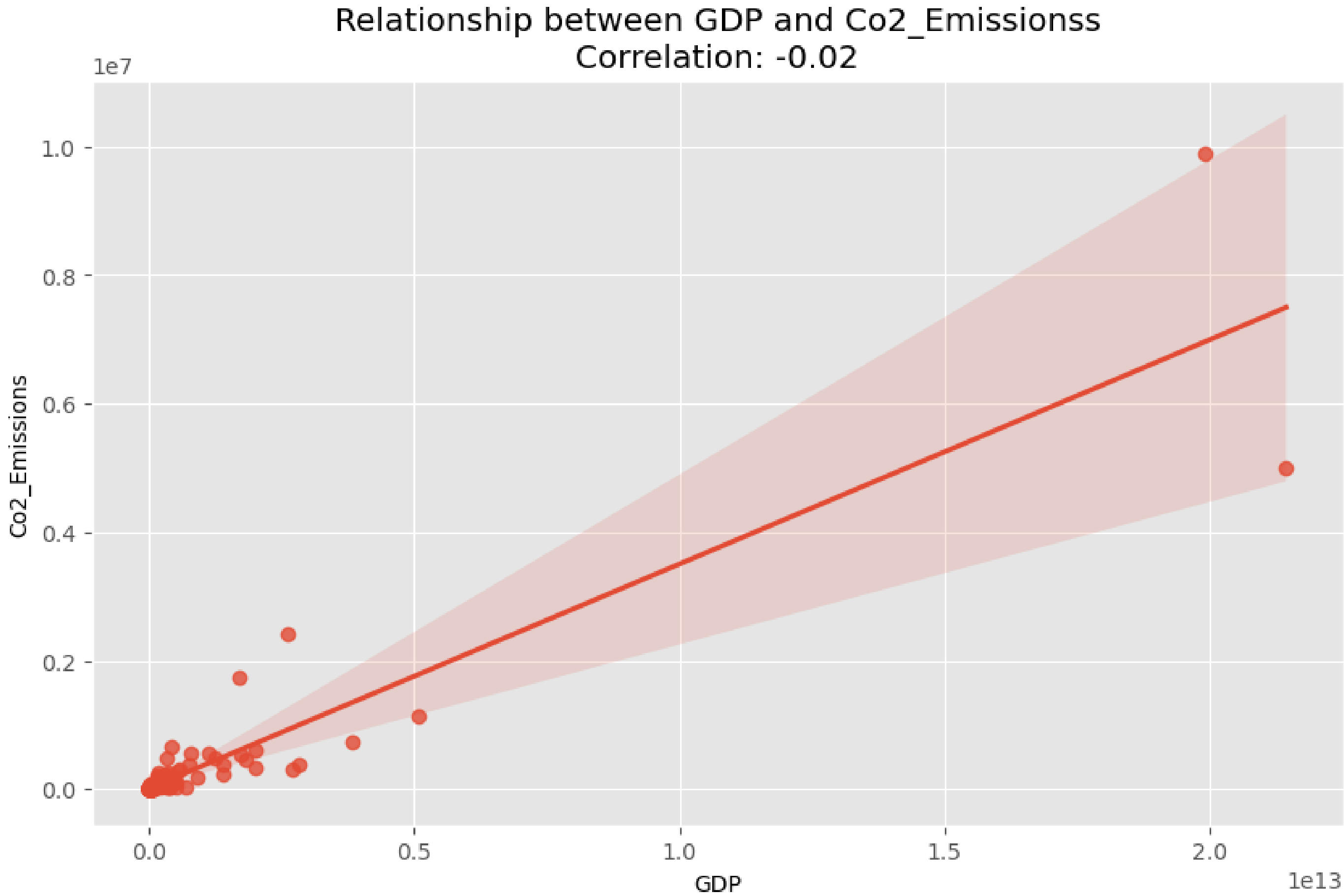
Out[20]:
```
<Axes: title={'center': 'Country with Highest GDP'}, xlabel='Country'>
```



In [21]:
```
# Question 5: What is the relationship between GDP and CO2 emissions?
relationship_GDP_CO2_Emission = world['GDP'].corr(world['Co2_Emissions'])
relationship_GDP_CO2_Emission
```

Out[21]:
```
0.9169960708699614
```

In [22]:
```
plt.figure(figsize=(10, 6))
sns.regplot(x='GDP', y='Co2_Emissions', data=world)
plt.title(f"Relationship between GDP and Co2_Emissionss\nCorrelation: {correlation:.2f}"
plt.xlabel('GDP')
plt.ylabel('Co2_Emissions')
plt.grid(True)
plt.show()
```



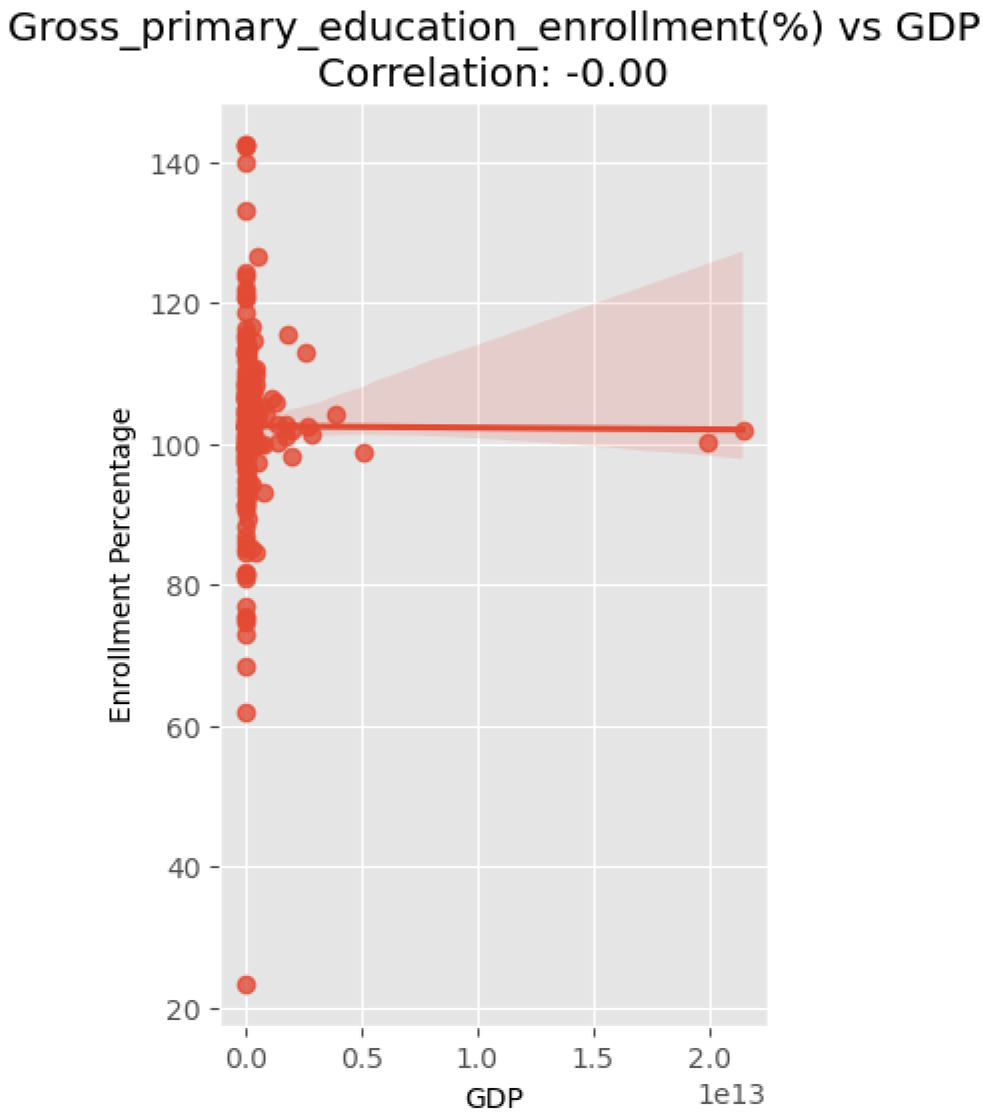
In [23]:
```
# Question 6: Is there a correlation between GDP and various education enrollment percen
correlation_primary = world['GDP'].corr(world['Gross_primary_education_enrollment(%)'])
correlation_tertiary = world['GDP'].corr(world['Gross_tertiary_education_enrollment(%)']
```

In [24]:
```
print(correlation_primary)
print(correlation_tertiary)
```

```
-0.004058638282026252
0.2115579171736697
```

In [25]:
```
plt.figure(figsize=(12, 6))

plt.subplot(1, 3, 1)
sns.regplot(x='GDP', y='Gross_primary_education_enrollment(%)', data=world)
plt.title(f"Gross_primary_education_enrollment(%) vs GDP\nCorrelation: {correlation_prim
plt.xlabel('GDP')
plt.ylabel('Enrollment Percentage')
```

Out[25]:
```
Text(0, 0.5, 'Enrollment Percentage')
```

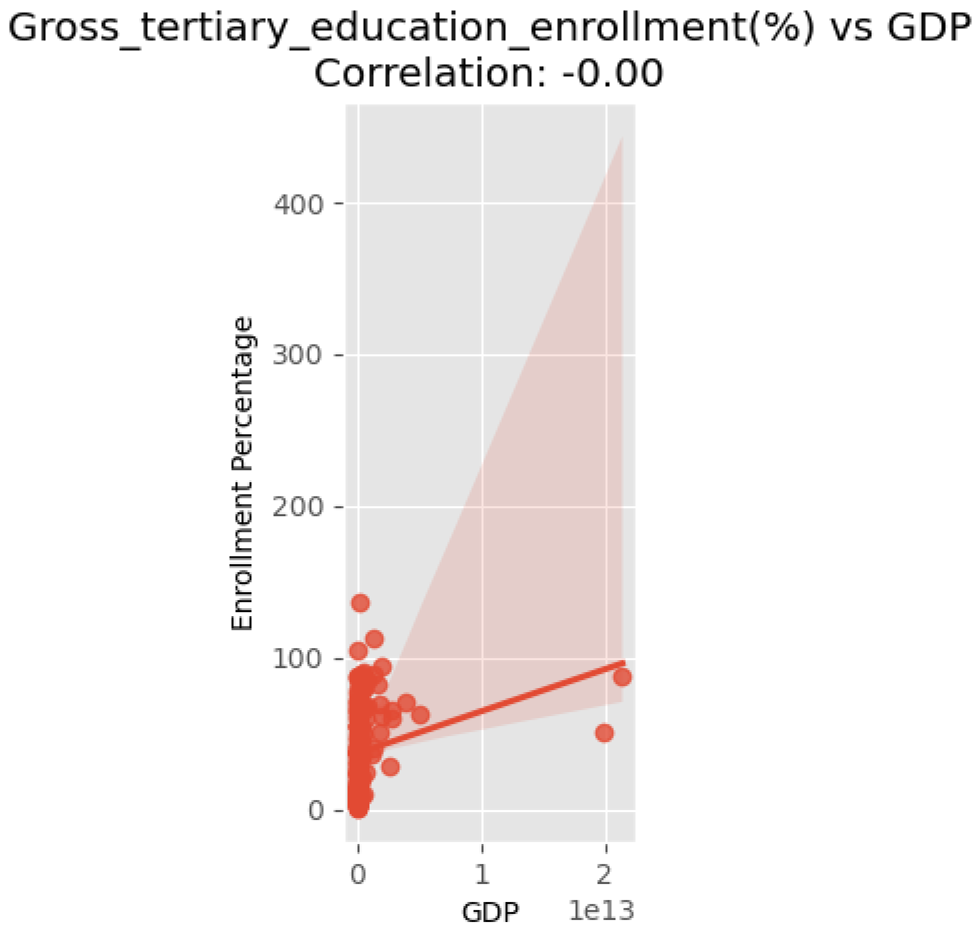


In [26]:

```
plt.subplot(1, 3, 1)
sns.regplot(x='GDP', y='Gross_tertiary_education_enrollment(%)', data=world)
plt.title(f"Gross_tertiary_education_enrollment(%) vs GDP\nCorrelation: {correlation_pri
plt.xlabel('GDP')
plt.ylabel('Enrollment Percentage')
```

Out[26]:

```
Text(0, 0.5, 'Enrollment Percentage')
```



3. Health and Social Indicators

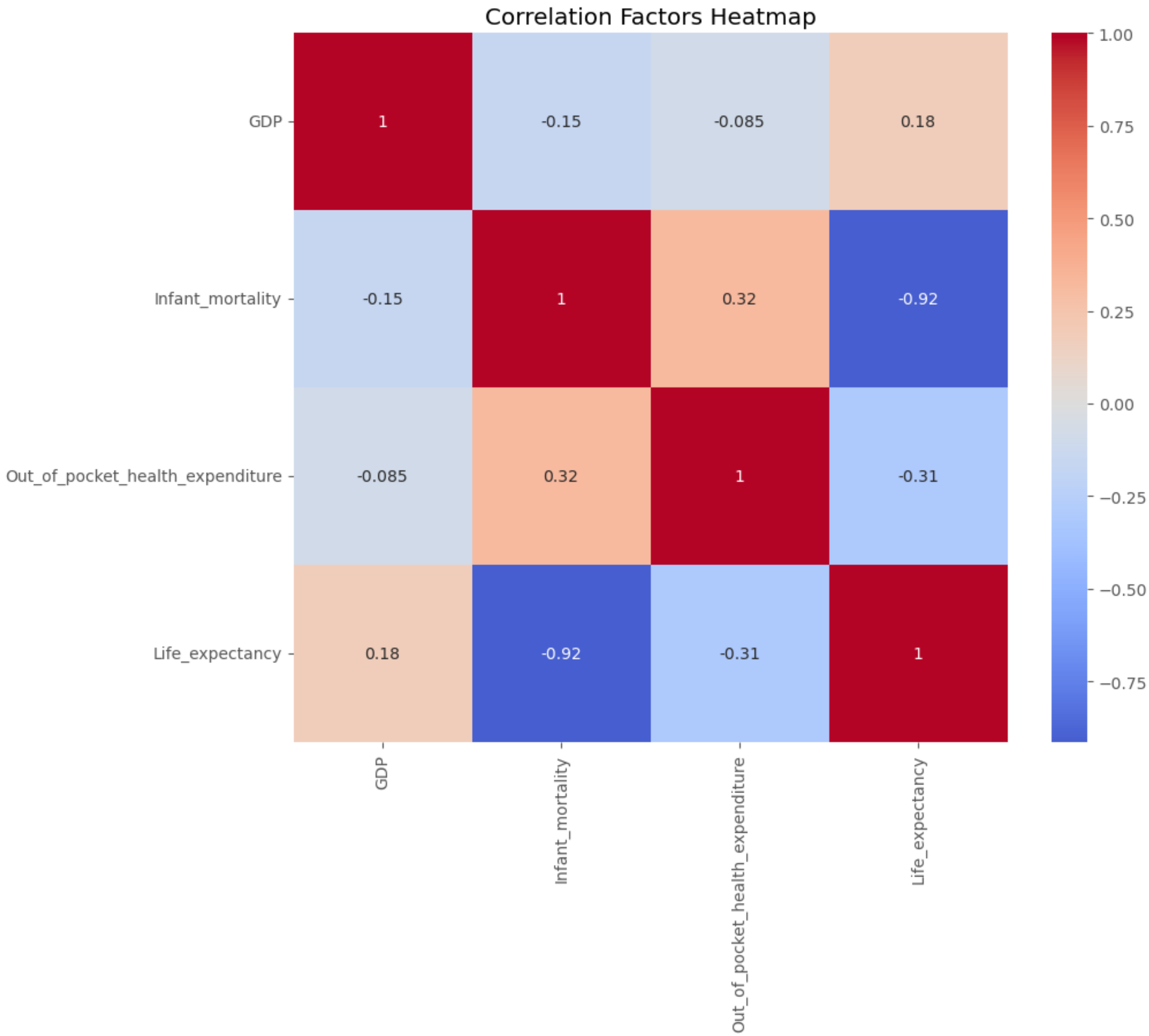
In [27]:
```
# Question 7:  How does life expectancy correlate with factors like GDP, infant mortalit
correlation_matrix = world[['GDP', 'Infant_mortality', 'Out_of_pocket_health_expenditure
correlation_matrix
```

Out[27]:

	GDP	Infant_mortality	Out_of_pocket_health_expenditure	Life_expectancy
GDP	1.000000	-0.152818	-0.085188	0.175589
Infant_mortality	-0.152818	1.000000	0.318019	-0.915068
Out_of_pocket_health_expenditure	-0.085188	0.318019	1.000000	-0.305220
Life_expectancy	0.175589	-0.915068	-0.305220	1.000000

In [28]:
```
plt.figure(figsize=(10, 8))
plt.title('Correlation Factors Heatmap')
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', center=0)
```

Out[28]:
```
<Axes: title={'center': 'Correlation Factors Heatmap'}>
```



In [29]:

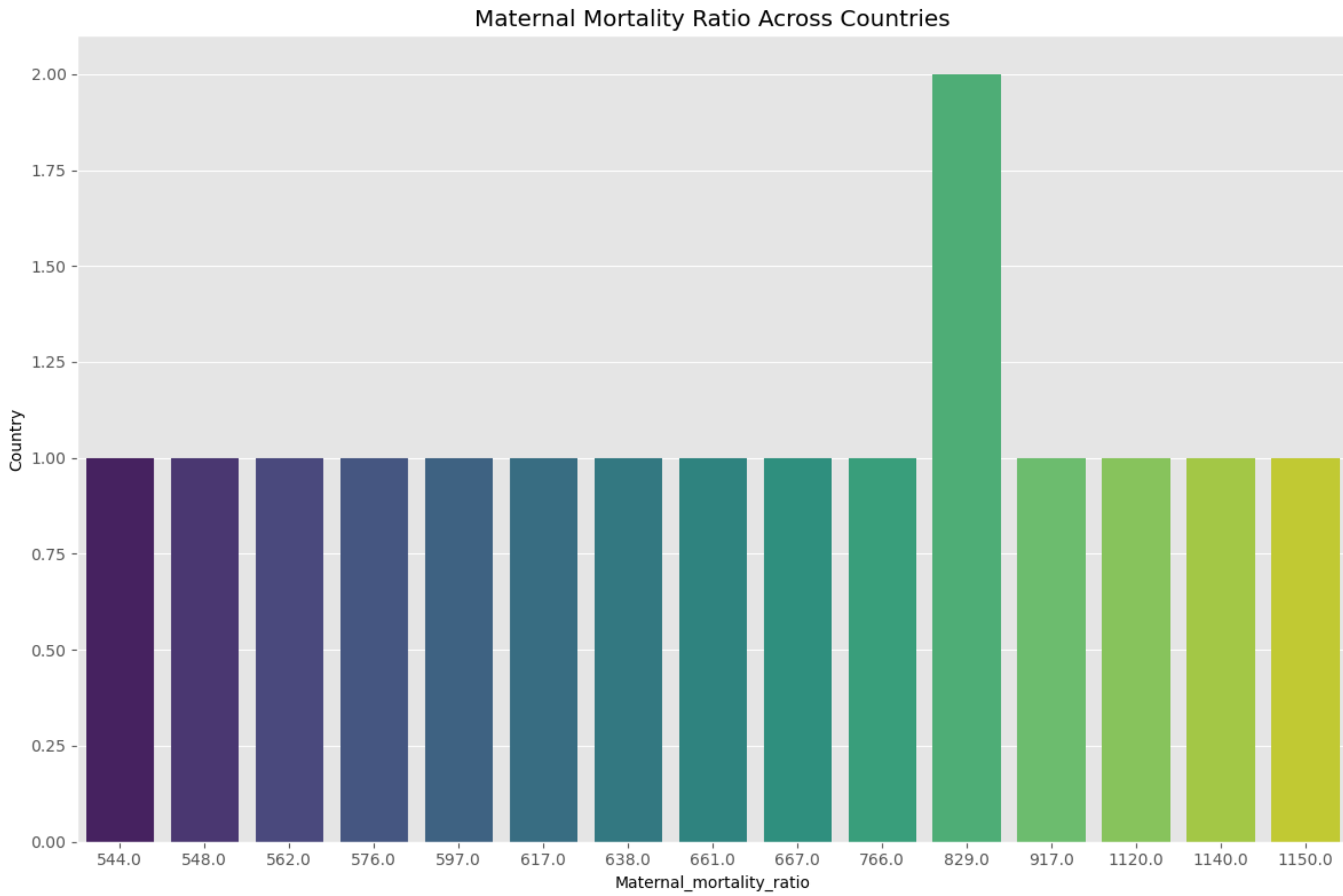
```
# Question 8: What is the maternal mortality ratio like across different countries?
sorted_world = world.groupby('Maternal_mortality_ratio')['Country'].count().reset_index(
maternal_mortality_country = sorted_world.sort_values(by='Maternal_mortality_ratio', asc
maternal_mortality_country
```

Out[29]:

	Maternal_mortality_ratio	Country
114	1150.0	1
113	1140.0	1
112	1120.0	1
111	917.0	1
110	829.0	2
109	766.0	1
108	667.0	1
107	661.0	1
106	638.0	1
105	617.0	1
104	597.0	1

103	576.0	1
102	562.0	1
101	548.0	1
100	544.0	1

In [30]:
```
plt.figure(figsize=(12, 8))
plt.title('Maternal Mortality Ratio Across Countries')

# Create a bar plot using Seaborn
sns.barplot(data=maternal_mortality_country, x='Maternal_mortality_ratio', y='Country',

# Show the plot
plt.xlabel('Maternal_mortality_ratio')
plt.ylabel('Country')
plt.tight_layout()
plt.show()
```



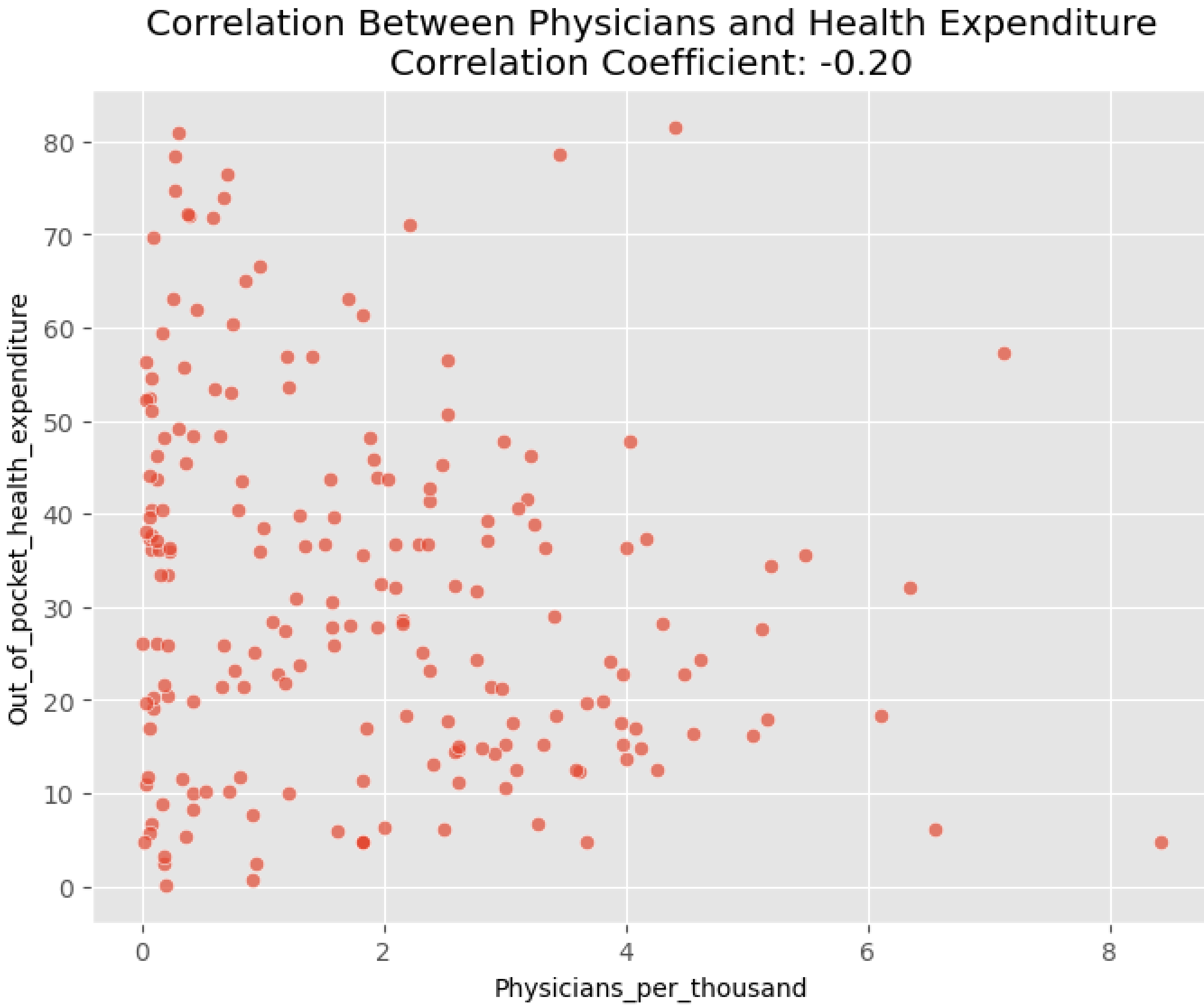
In [31]:
```
# Question 9: Is there a correlation between physicians per thousand people and overall
correlation_coefficient = np.corrcoef(world['Physicians_per_thousand'], world['Out_of_po
correlation_coefficient
```

Out[31]:
```
-0.19875144618065713
```

In [32]:
```
# Set up the plot
plt.figure(figsize=(8, 6))
plt.title(f'Correlation Between Physicians and Health Expenditure\nCorrelation Coefficie

# Create a scatter plot using Seaborn
sns.scatterplot(data=world, x='Physicians_per_thousand', y='Out_of_pocket_health_expendi
```

Out[32]:
```
<Axes: title={'center': 'Correlation Between Physicians and Health Expenditure\nCorrelat
ion Coefficient: -0.20'}, xlabel='Physicians_per_thousand', ylabel='Out_of_pocket_health
```

_expenditure'>

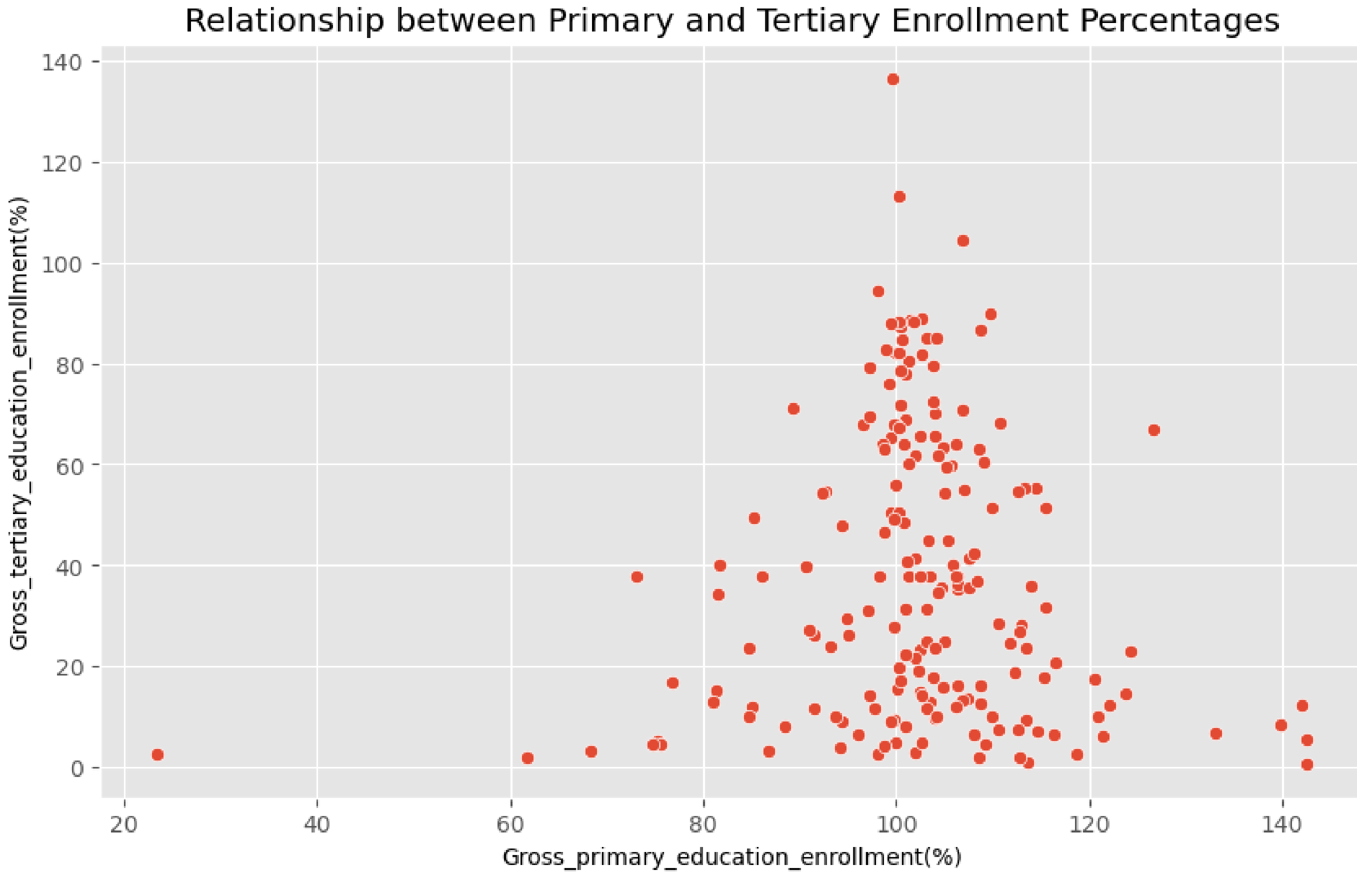


4. Education and Development

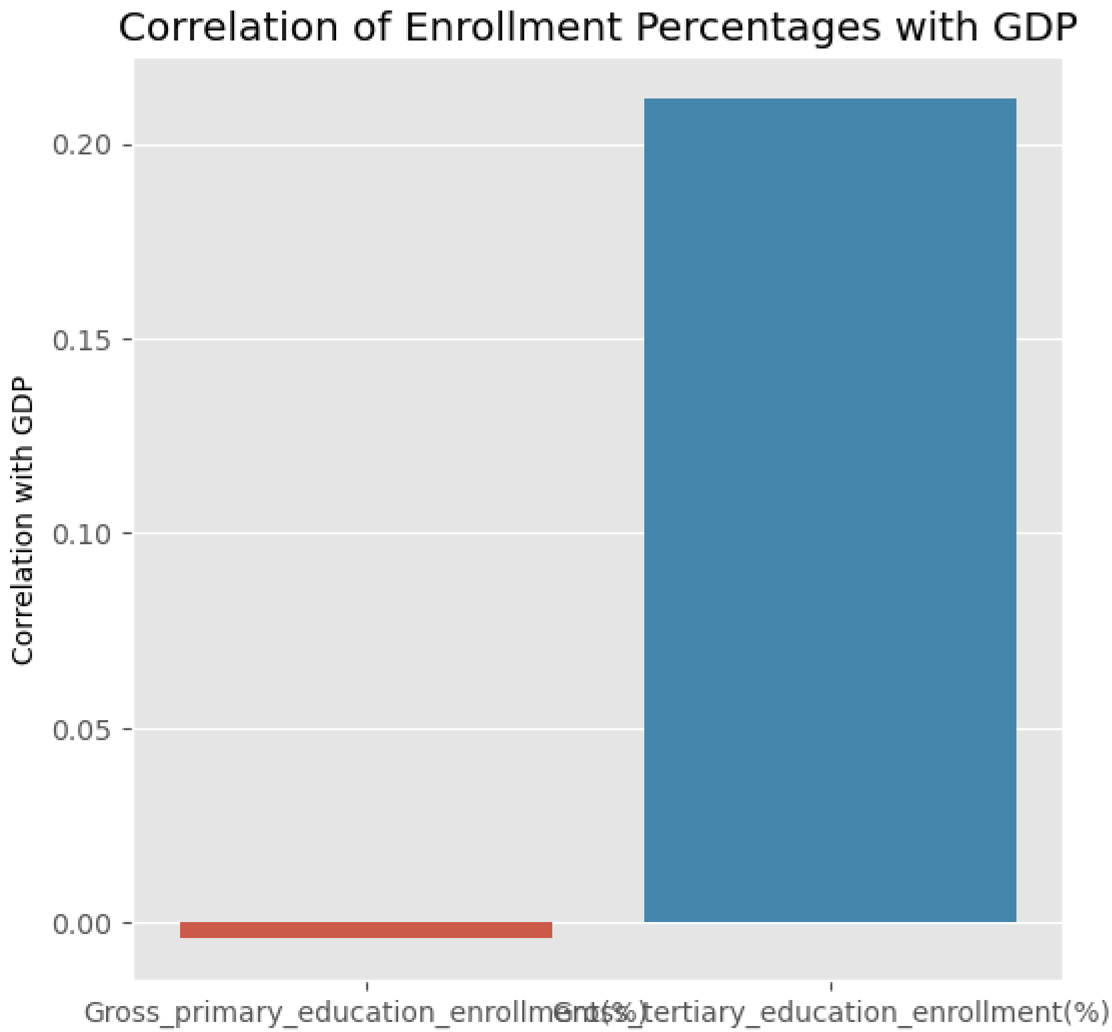
In [33]:
```
# Question 10: How do different education enrollment percentages (primary and tertiary)
correlation_primary = world['Gross_primary_education_enrollment(%)'].corr(world['GDP'])
correlation_tertiary = world['Gross_tertiary_education_enrollment(%)'].corr(world['GDP']

print(correlation_primary)
print(correlation_tertiary)
```

```
-0.004058638282026252
0.21155791717366973
```

In [34]:
```
# visualizing the relationship between Primary Enrollment and Tertiary Enrollment
plt.figure(figsize=(10, 6))

sns.scatterplot(data=world, x='Gross_primary_education_enrollment(%)', y='Gross_tertiary
plt.xlabel('Gross_primary_education_enrollment(%)')
plt.ylabel('Gross_tertiary_education_enrollment(%)')

plt.title('Relationship between Primary and Tertiary Enrollment Percentages')

plt.show()
```



In [35]:
```
# visualizing the correlations with GDP
plt.figure(figsize=(6, 6))

sns.barplot(x=['Gross_primary_education_enrollment(%)', 'Gross_tertiary_education_enroll
plt.ylabel('Correlation with GDP')
plt.title('Correlation of Enrollment Percentages with GDP')
plt.show()
```

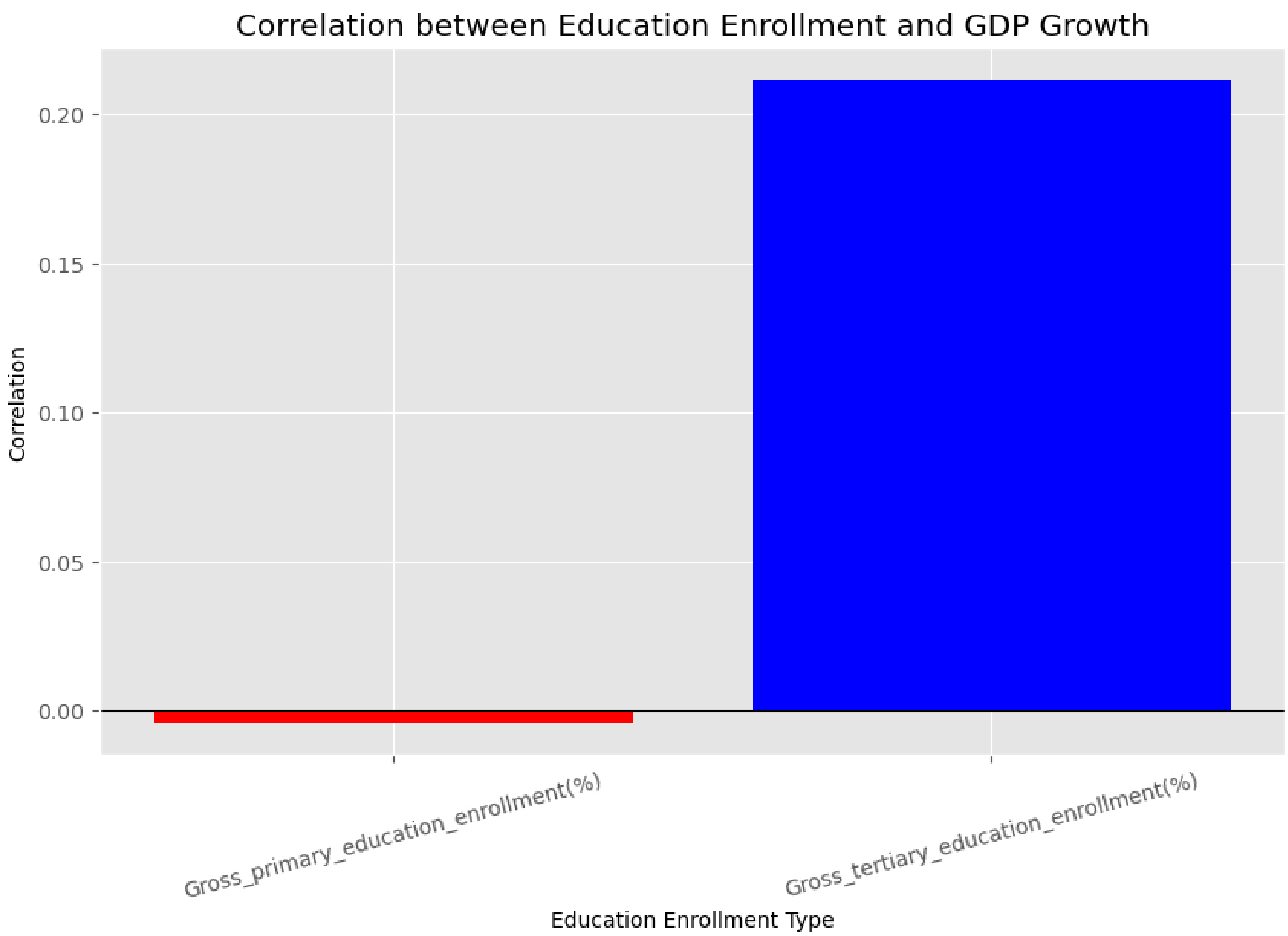


In [36]:
```python
# Question 11: Is there a relationship between education enrollment and GDP growth?

# Calculate correlations for primary and tertiary education enrollments
correlation_primary = world['Gross_primary_education_enrollment(%)'].corr(world['GDP'])
correlation_tertiary = world['Gross_tertiary_education_enrollment(%)'].corr(world['GDP']

# Determine the strength and direction of the relationship for primary education
if correlation_primary > 0:
    primary_relationship = "positive"
elif correlation_primary < 0:
    primary_relationship = "negative"
else:
    primary_relationship = "no"

# Determine the strength and direction of the relationship for tertiary education
if correlation_tertiary > 0:
    tertiary_relationship = "positive"
elif correlation_tertiary < 0:
    tertiary_relationship = "negative"
else:
    tertiary_relationship = "no"

print(f"The correlation between Primary Education Enrollment and GDP Growth is {correlat
print(f"There is a {primary_relationship} relationship between them.")

print(f"The correlation between Tertiary Education Enrollment and GDP Growth is {correla
print(f"There is a {tertiary_relationship} relationship between them.")
```

```
The correlation between Primary Education Enrollment and GDP Growth is -0.00.
There is a negative relationship between them.
```

```
The correlation between Tertiary Education Enrollment and GDP Growth is 0.21.
There is a positive relationship between them.
```

In [37]:
```
# Visualizing the correlation between education enrolment and GDP growth.
categories = ['Gross_primary_education_enrollment(%)', 'Gross_tertiary_education_enrollm
correlation_values = [correlation_primary, correlation_tertiary]
colors = ['blue' if c > 0 else 'red' for c in correlation_values]

plt.figure(figsize=(10, 6))
plt.bar(categories, correlation_values, color=colors)
plt.axhline(y=0, color='black', linewidth=0.8)
plt.title('Correlation between Education Enrollment and GDP Growth')
plt.ylabel('Correlation')
plt.xlabel('Education Enrollment Type')
plt.xticks(rotation=15)
plt.show()
```



5. Geographical Analysis

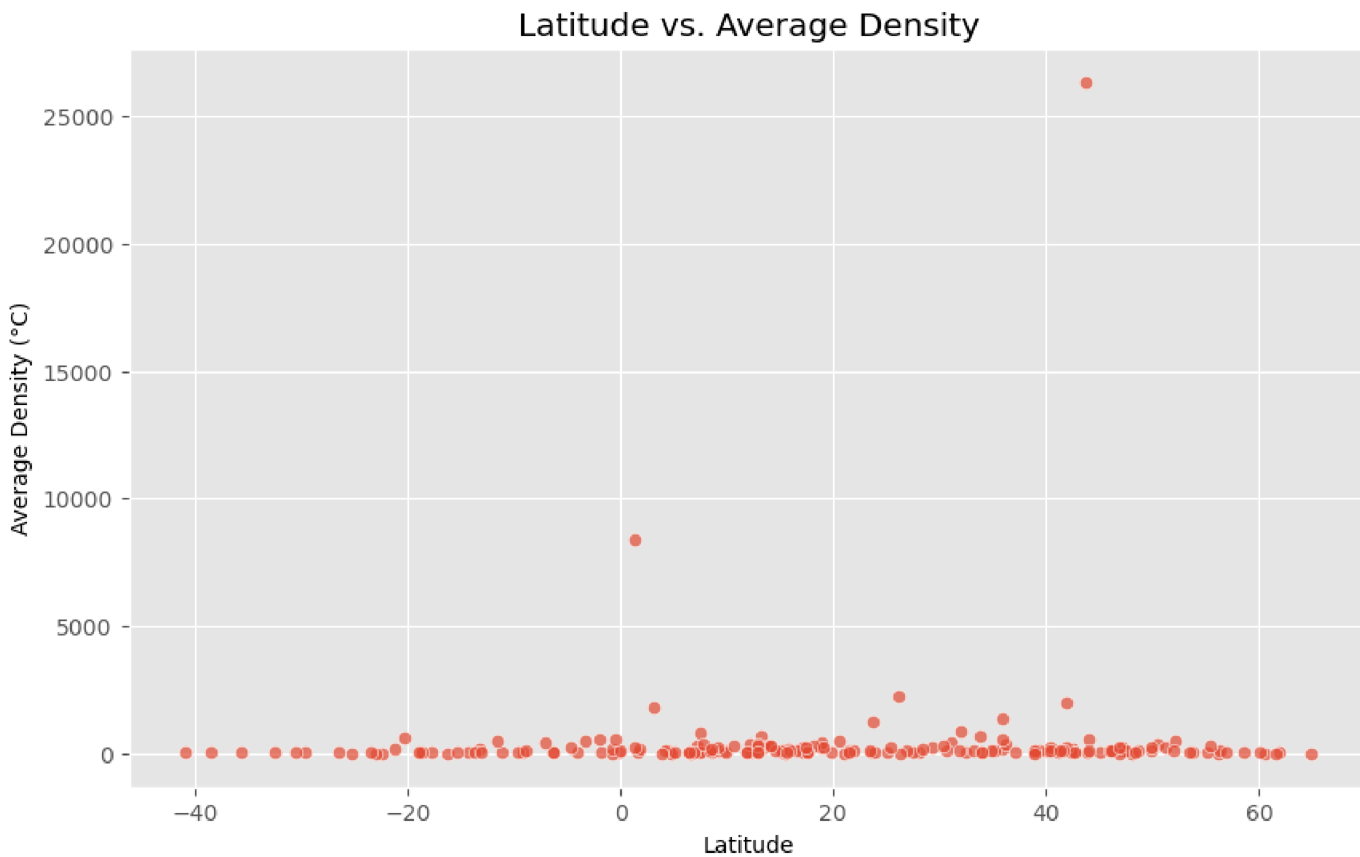
In [38]:
```
# Question 12: Calculate the average density for each latitude
average_temps_by_latitude = world.groupby('Latitude')['Density'].mean().head(10)
average_temps_by_latitude
```

Out[38]:
```
Latitude
-40.900557    18.0
-38.416097    17.0
-35.675147    26.0
-32.522779    20.0
-30.559482    49.0
-29.609988    71.0
-26.522503    67.0
-25.274398     3.0
```

```
-23.442503    18.0
-22.957640     3.0
Name: Density, dtype: float64
```

In [39]:

```python
# Create a scatter plot to visualize the relationship between latitude and average densi
plt.figure(figsize=(10, 6))
sns.scatterplot(x=world['Latitude'], y=world['Density'], alpha=0.7)
plt.title('Latitude vs. Average Density')
plt.xlabel('Latitude')
plt.ylabel('Average Density (°C)')
plt.grid(True)
plt.show()
```



In [40]:

```python
# Question 13: Are there any noticeable trends when comparing data across different coun
# Calculate mean values for each country and column
country_means = world.groupby('Country').mean()

# Calculate total counts for each region
country_counts = world['Country'].value_counts()
country_counts
```

```
C:\Users\njoku\AppData\Local\Temp\ipykernel_27324\2245185099.py:3: FutureWarning: The de
fault value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only columns w
hich should be valid for the function.
  country_means = world.groupby('Country').mean()
```

Out[40]:

```
South Africa                    2
Afghanistan                     1
Palestinian National Authority  1
Nicaragua                       1
Niger                           1
                               ..
Grenada                         1
Guatemala                       1
Guinea                          1
Guinea-Bissau                   1
```

```
Zimbabwe                              1
Name: Country, Length: 194, dtype: int64
```

6. Economic Stability

In [41]:
```
# Question 14: How do the unemployment rate and minimum wage relate to each other?

# Calculate the correlation coefficient between Unemployment_rate and Minimum_wage
Unemployment_mini_wage_relate = world["Unemployment_rate"].corr(world["Minimum_wage"])
Unemployment_mini_wage_relate
```

Out[41]:
```
-0.031379955140272656
```

In [42]:
```
# Question 15: Is there a correlation between the unemployment rate and GDP?
# Calculate the correlation between unemployment rate and GDP
unemployment_GDP_corr = world['Unemployment_rate'].corr(world['GDP'])
unemployment_GDP_corr
```

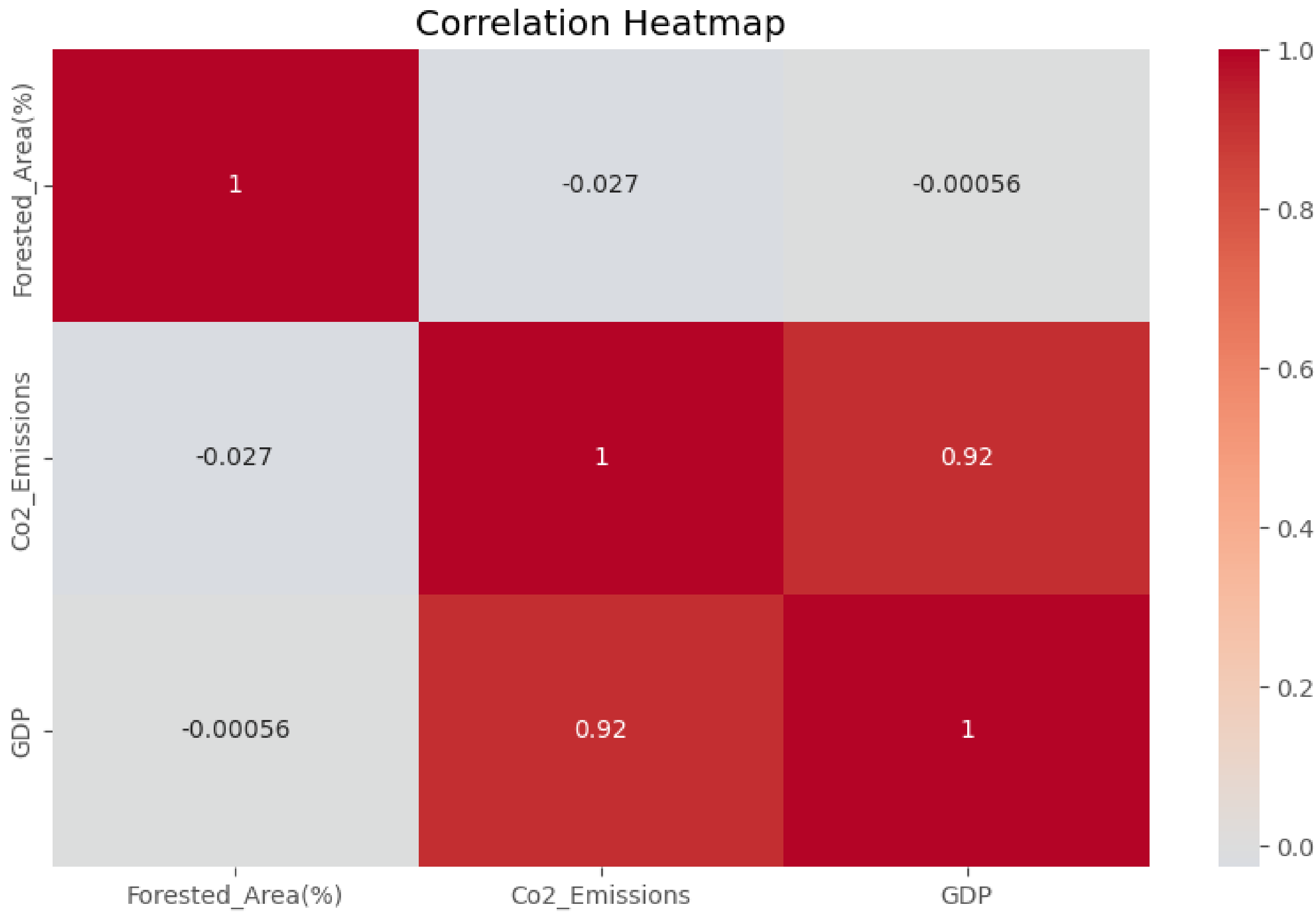
Out[42]:
```
0.031168460585031293
```

7. Environmental Factors

In [43]:
```
# Question 16: How does the percentage of forested area correlate with CO2 emissions and
# Calculate correlation coefficients
correlation_forest_co2 = world["Forested_Area(%)"].corr(world["Co2_Emissions"])
correlation_forest_gdp = world["Forested_Area(%)"].corr(world["GDP"])
correlation_co2_gdp = world["Co2_Emissions"].corr(world["GDP"])
print("Correlation between Forested Area (%) and CO2 Emissions:", correlation_forest_co2
print("Correlation between Forested Area (%) and GDP:", correlation_forest_gdp)
print("Correlation between CO2 Emissions and GDP:", correlation_co2_gdp)
```

```
Correlation between Forested Area (%) and CO2 Emissions: -0.027101448695144918
Correlation between Forested Area (%) and GDP: -0.0005623297969089521
Correlation between CO2 Emissions and GDP: 0.9169960708699614
```

In [44]:
```
# Calculate the correlation matrix
correlation_matrix = world[["Forested_Area(%)", "Co2_Emissions", "GDP"]].corr()

# Create a heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", center=0)
plt.title("Correlation Heatmap")
plt.show()
```



In [45]:

```
# Question 17: What is the relationship between agricultural land percentage and foreste
# Calculate the correlation coefficient
agric_forest_area_corr = world['Agricultural_Land( %)'].corr(world['Forested_Area(%)'])

print("agric_forest_area_cor:", correlation)
```

```
agric_forest_area_cor: -0.01794615744859417
```

8. Taxation and Revenue

In [46]:

```
world
```

Out[46]:

	Country	Density	Abbreviation	Agricultural_Land(%)	Land_Area(Km2)	Armed_Forces_size	Birth_Rate	Callir
0	Afghanistan	60	AF	58.1	652230.0	323000.00	32.49	
1	Albania	105	AL	43.1	28748.0	9000.00	11.78	
2	Algeria	18	DZ	17.4	2381741.0	317000.00	24.28	
3	Andorra	164	AD	40.0	468.0	6.89	7.20	
4	Angola	26	AO	47.5	1246700.0	117000.00	40.73	
...	...	...	...	...	...	...	...	
190	Venezuela	32	VE	24.5	912050.0	343000.00	17.88	
191	Vietnam	314	VN	39.3	331210.0	522000.00	16.75	
192	Yemen	56	YE	44.6	527968.0	40000.00	30.45	

193	Zambia	25	ZM	32.1	752618.0	16000.00	36.19
194	Zimbabwe	38	ZW	41.9	390757.0	51000.00	30.68

195 rows × 35 columns

In [47]:
```
# Question 18: How does the total tax rate impact tax revenue percentage?
# Calculate the correlation coefficient
correlation = world['Total_tax_rate'].corr(world['Tax_revenue(%)'])

print("Correlation coefficient:", correlation)
```

```
Correlation coefficient: -0.08068023658110621
```

In [48]:
```
# Question 19: Are there countries where the tax revenue percentage is significantly dif
# Calculate the average tax rate
average_tax_rate = world["Tax_revenue(%)"].mean()
average_tax_rate
```

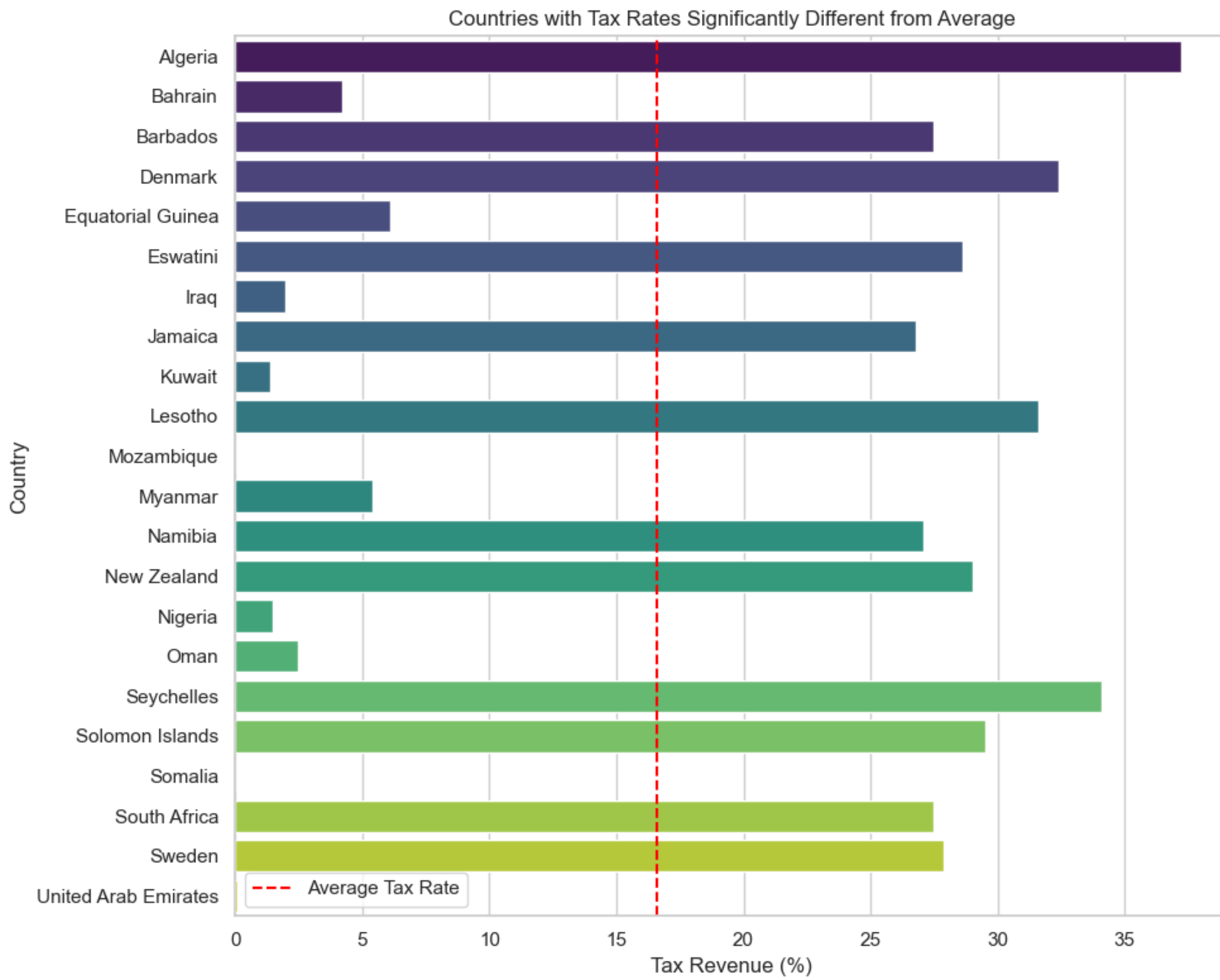
Out[48]:
```
16.573435897435896
```

In [49]:
```
# average_tax_rate = 16.573435897435896

# Define a threshold for significant difference
significant_difference_threshold = 10  # For example, 10%

# Find countries with tax rates significantly different from the average
countries_with_significant_difference = world[
    (world["Tax_revenue(%)"] > average_tax_rate + significant_difference_threshold) |
    (world["Tax_revenue(%)"] < average_tax_rate - significant_difference_threshold)]

# Print the results
print("Average Tax Rate:", average_tax_rate)
print("Countries with Tax Rates Significantly Different from Average:")
print(countries_with_significant_difference[["Country", "Tax_revenue(%)"]])
```

```
Average Tax Rate: 16.573435897435896
Countries with Tax Rates Significantly Different from Average:
                  Country  Tax_revenue(%)
2                 Algeria            37.2
12                Bahrain             4.2
14               Barbados            27.5
46                Denmark            32.4
53      Equatorial Guinea             6.1
56               Eswatini            28.6
80                   Iraq             2.0
84                Jamaica            26.8
90                 Kuwait             1.4
95                Lesotho            31.6
117            Mozambique             0.0
118               Myanmar             5.4
119               Namibia            27.1
123           New Zealand            29.0
126               Nigeria             1.5
130                  Oman             2.5
154            Seychelles            34.1
159       Solomon Islands            29.5
160               Somalia             0.0
161          South Africa            27.5
168                Sweden            27.9
184  United Arab Emirates             0.1
```

In [50]:
```
# visualizing the counytries with significant different with the average tax rate
sns.set(style="whitegrid")
```

```python
# Plot the data
plt.figure(figsize=(10, 8))
sns.barplot(x="Tax_revenue(%)", y="Country", data=countries_with_significant_difference,
plt.axvline(average_tax_rate, color="red", linestyle="--", label="Average Tax Rate")
plt.xlabel("Tax Revenue (%)")
plt.ylabel("Country")
plt.title("Countries with Tax Rates Significantly Different from Average")
plt.legend()
plt.tight_layout()

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



9. Cultural and Linguistic Analysis

In [51]:

```python
# Question 20: How do the official languages of different countries correlate with their

# Calculate the correlation between GDP and Gross tertiary education enrollment
gdp_tertiary_corr = world[['GDP', 'Gross_tertiary_education_enrollment(%)']].corr().iloc

# Calculate the correlation between GDP and Gross primary education enrollment
gdp_primary_corr = world[['GDP', 'Gross_primary_education_enrollment(%)']].corr().iloc[0

# Calculate the correlation between GDP and Official language
gdp_lang_corr = world[['GDP', 'Official_language']].groupby('Official_language').mean()

# Calculate the correlation between Gross tertiary education enrollment and Official lan
tertiary_lang_corr = world[['Gross_tertiary_education_enrollment(%)', 'Official_language
```

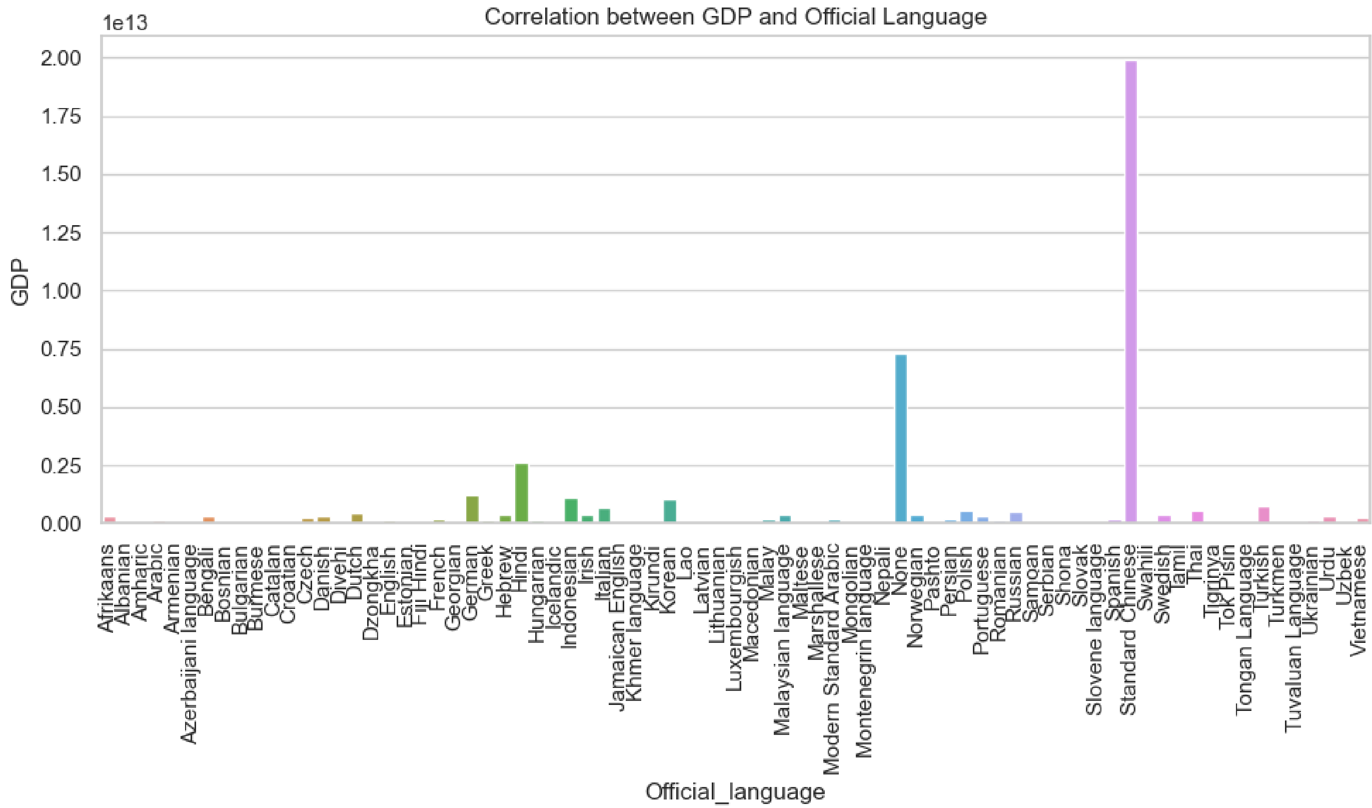
```
# Calculate the correlation between Gross primary education enrollment and Official lang
primary_lang_corr = world[['Gross_primary_education_enrollment(%)', 'Official_language']

print("Correlation between GDP and Gross tertiary education enrollment:", gdp_tertiary_c
print("Correlation between GDP and Gross primary education enrollment:", gdp_primary_cor
print("Correlation between GDP and Official language:\n", gdp_lang_corr)
print("Correlation between Gross tertiary education enrollment and Official language:\n"
print("Correlation between Gross primary education enrollment and Official language:\n",
```

```
Correlation between GDP and Gross tertiary education enrollment: 0.21155791717366976
Correlation between GDP and Gross primary education enrollment: -0.004058638282026142
Correlation between GDP and Official language:
                              GDP
Official_language
Afrikaans          3.514316e+11
Albanian           1.527808e+10
Amharic            9.610766e+10
Arabic             1.290256e+11
Armenian           1.367280e+10
...                          ...
Tuvaluan Language  4.727146e+07
Ukrainian          1.537811e+11
Urdu               3.044000e+11
Uzbek              5.792129e+10
Vietnamese         2.619212e+11

[77 rows x 1 columns]
Correlation between Gross tertiary education enrollment and Official language:
                    Gross_tertiary_education_enrollment(%)
Official_language
Afrikaans                                        22.400000
Albanian                                         55.000000
Amharic                                           8.100000
Arabic                                           32.236667
Armenian                                         54.600000
...                                                    ...
Tuvaluan Language                                37.960000
Ukrainian                                        82.700000
Urdu                                              9.000000
Uzbek                                            10.100000
Vietnamese                                       28.500000

[77 rows x 1 columns]
Correlation between Gross primary education enrollment and Official language:
                    Gross_primary_education_enrollment(%)
Official_language
Afrikaans                                      100.900000
Albanian                                       107.000000
Amharic                                        101.000000
Arabic                                          95.898333
Armenian                                        92.700000
...                                                   ...
Tuvaluan Language                               86.000000
Ukrainian                                       99.000000
Urdu                                            94.300000
Uzbek                                          104.200000
Vietnamese                                     110.600000

[77 rows x 1 columns]
```

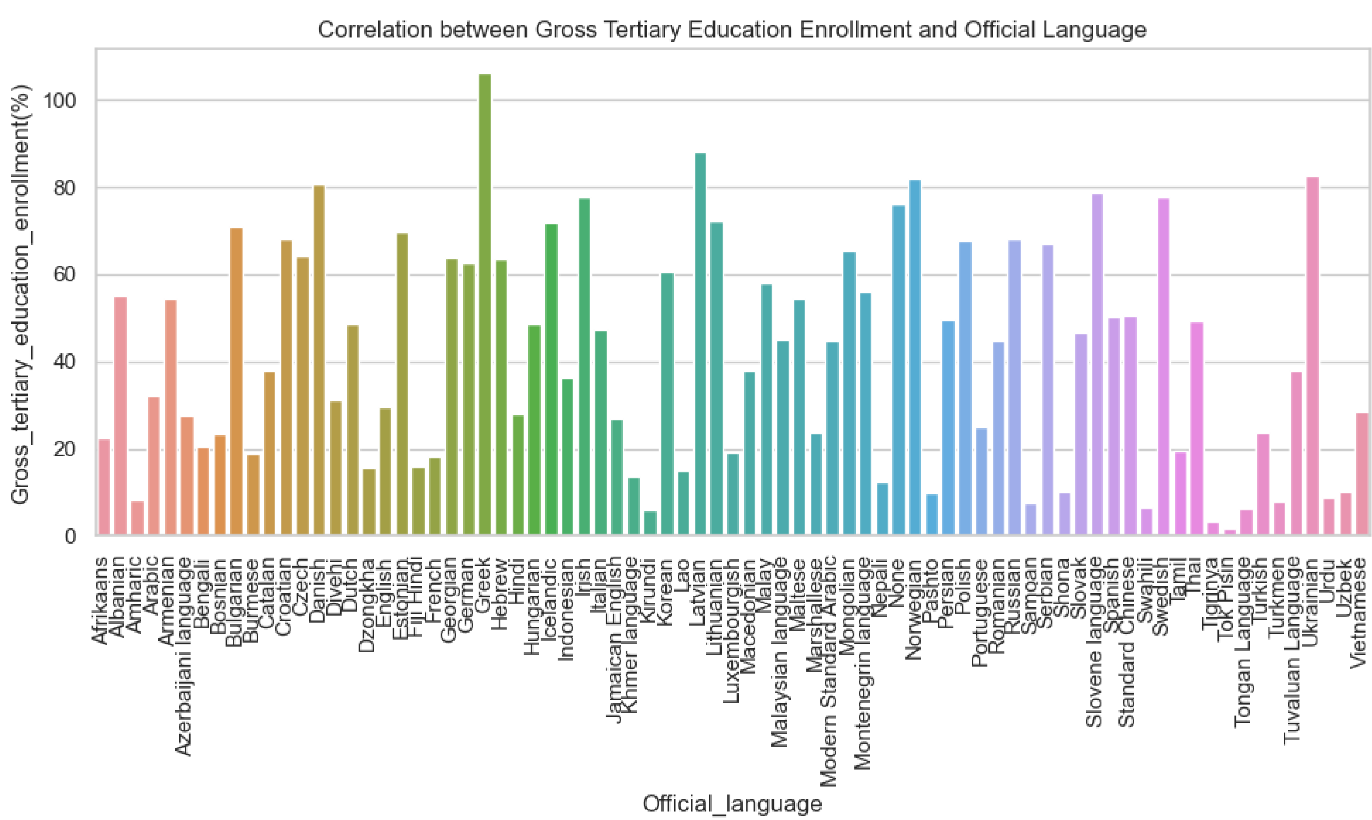
In [52]:

```
# Create a bar plot for the correlation between GDP and Official language
gdp_lang_corr = world[['GDP', 'Official_language']].groupby('Official_language').mean()
plt.figure(figsize=(10, 6))
sns.barplot(data=gdp_lang_corr.reset_index(), x='Official_language', y='GDP')
plt.xticks(rotation=90)
```

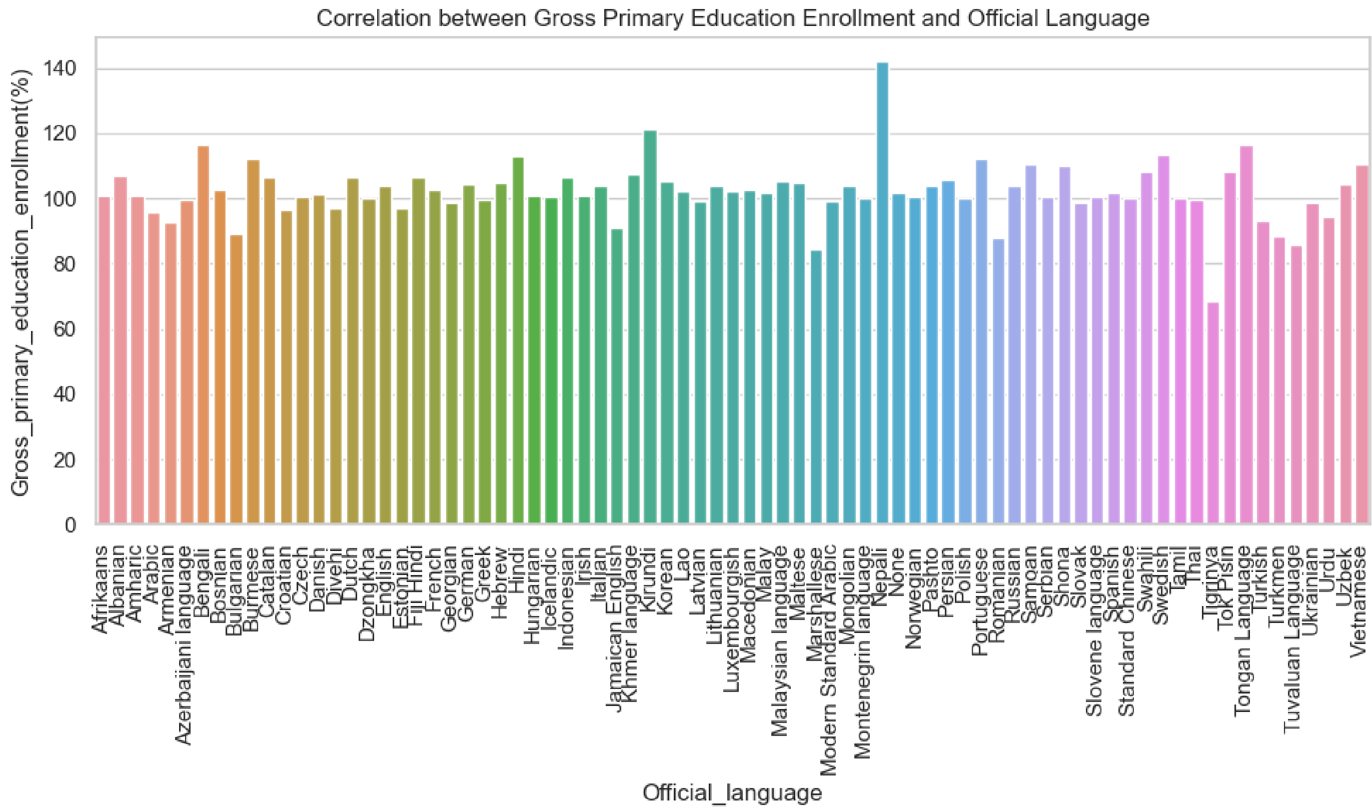
```
plt.title('Correlation between GDP and Official Language')
plt.tight_layout()
plt.show()
```



In [53]:
```
# Create a bar plot for the correlation between Gross tertiary education enrollment and
tertiary_lang_corr = world[['Gross_tertiary_education_enrollment(%)', 'Official_language
plt.figure(figsize=(10, 6))
sns.barplot(data=tertiary_lang_corr.reset_index(), x='Official_language', y='Gross_terti
plt.xticks(rotation=90)
plt.title('Correlation between Gross Tertiary Education Enrollment and Official Language
plt.tight_layout()
plt.show()
```



In [54]:
```
# Create a bar plot for the correlation between Gross primary education enrollment and O
primary_lang_corr = world[['Gross_primary_education_enrollment(%)', 'Official_language']
plt.figure(figsize=(10, 6))
sns.barplot(data=primary_lang_corr.reset_index(), x='Official_language', y='Gross_primar
plt.xticks(rotation=90)
plt.title('Correlation between Gross Primary Education Enrollment and Official Language'
plt.tight_layout()
plt.show()
```



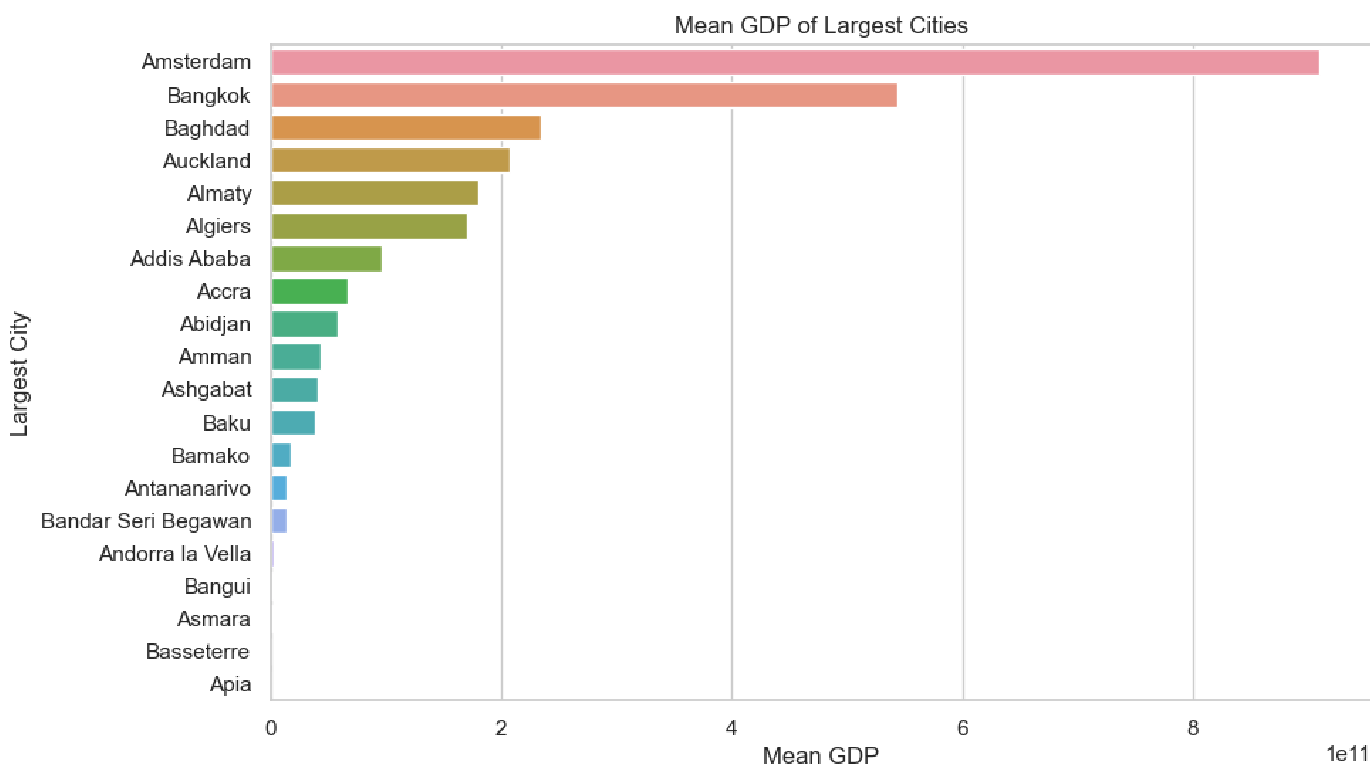
In [55]:
```
# Question 21: Are there any patterns in the largest cities of countries and their econo

largest_city_gdp = world[['Largest_city', 'GDP']]
city_mean_gdp = largest_city_gdp.groupby('Largest_city').mean().reset_index().head(20)

city_mean_gdp = city_mean_gdp.sort_values(by='GDP', ascending=False).head(20)

largest_city_gdp
city_mean_gdp
```

Out[55]:

	Largest_city	GDP
6	Amsterdam	9.090704e+11
17	Bangkok	5.436500e+11
13	Baghdad	2.340940e+11
12	Auckland	2.069288e+11
4	Almaty	1.801617e+11
3	Algiers	1.699882e+11
2	Addis Ababa	9.610766e+10
1	Accra	6.698363e+10
0	Abidjan	5.879221e+10
5	Amman	4.374366e+10

10	Ashgabat	4.076114e+10
14	Baku	3.920700e+10
15	Bamako	1.751014e+10
8	Antananarivo	1.408391e+10
16	Bandar Seri Begawan	1.346942e+10
7	Andorra la Vella	3.154058e+09
18	Bangui	2.220307e+09
11	Asmara	2.065002e+09
19	Basseterre	1.050993e+09
9	Apia	8.506550e+08

In [56]:
```
# Visualizing the Mean GDP of Largest Cities

plt.figure(figsize=(10, 6))
sns.barplot(x='GDP', y='Largest_city', data=city_mean_gdp)
plt.xlabel('Mean GDP')
plt.ylabel('Largest City')
plt.title('Mean GDP of Largest Cities')
plt.show()
```



10. Armed Forces and Security

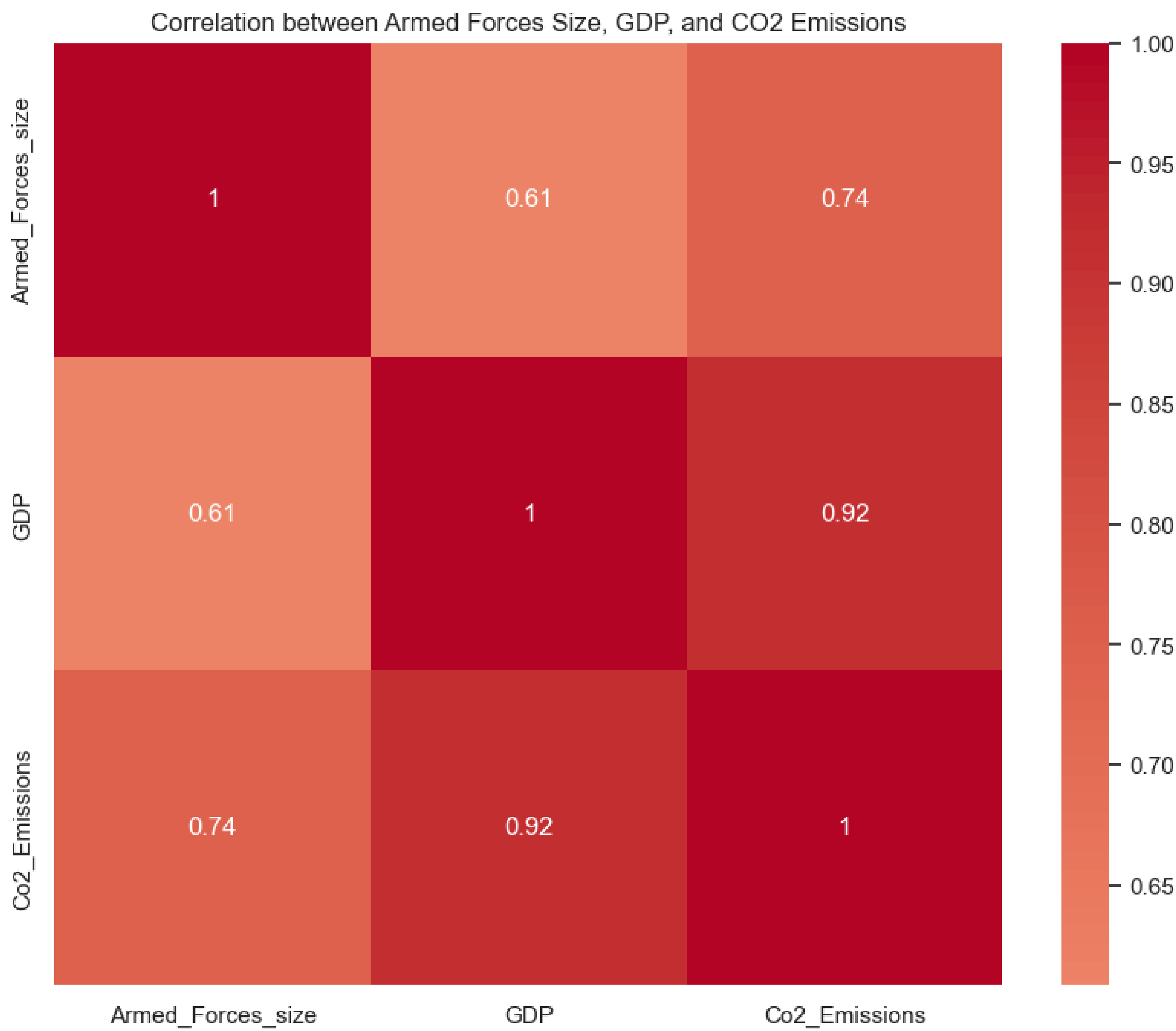
In [57]:
```
# Question 22: Is there a correlation between the size of armed forces and indicators li

# Calculate the correlation matrix
correlation_matrix = world[["Armed_Forces_size", "GDP", "Co2_Emissions"]].corr()

correlation_matrix
```

Out[57]:

	Armed_Forces_size	GDP	Co2_Emissions

Armed_Forces_size	1.000000	0.608933	0.742100
GDP	0.608933	1.000000	0.916996
Co2_Emissions	0.742100	0.916996	1.000000

In [58]:
```
# Visualize the correlation matrix using a heatmap:

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', center=0)
plt.title('Correlation between Armed Forces Size, GDP, and CO2 Emissions')
plt.show()
```

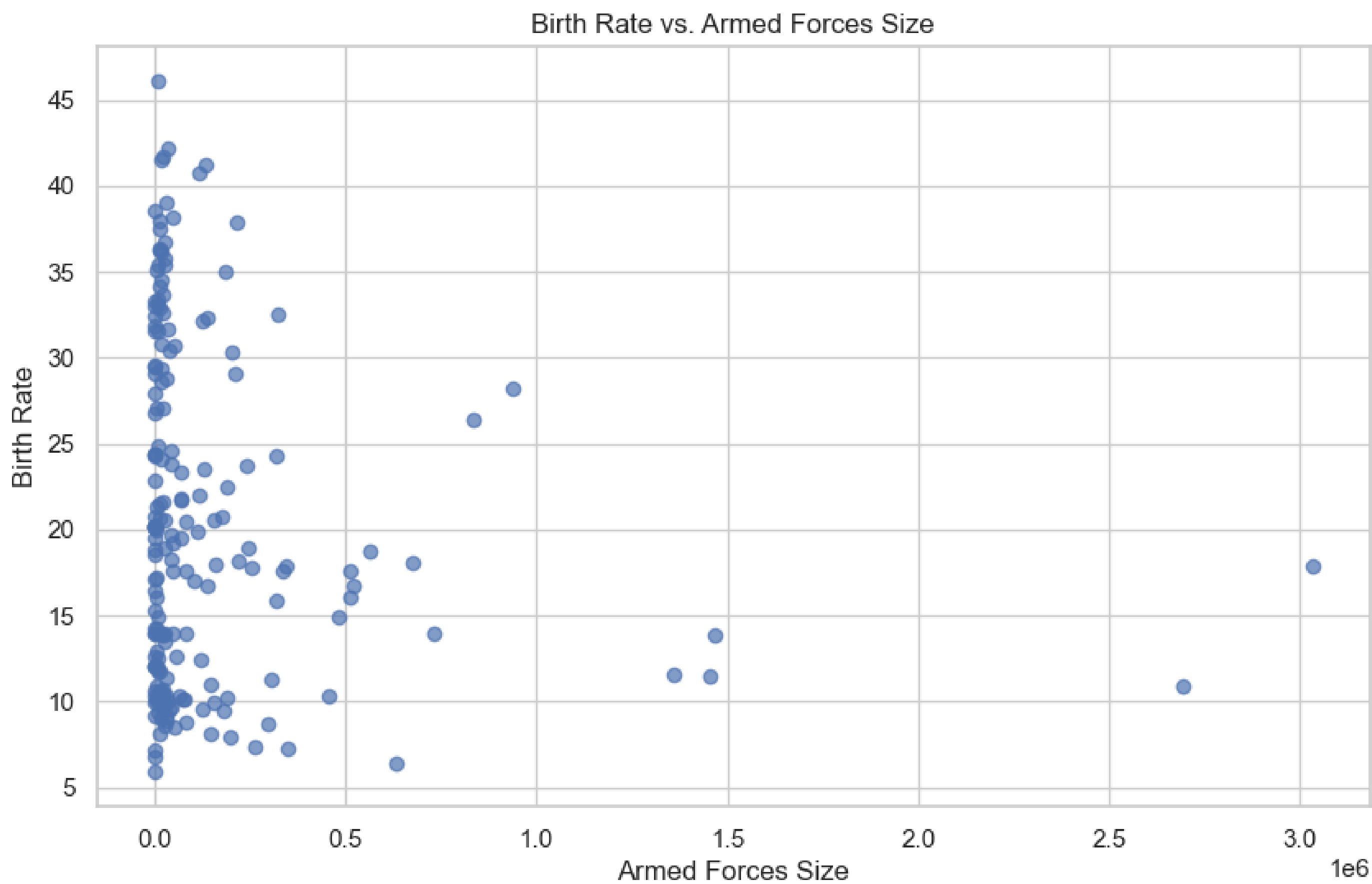


In [59]:
```
# Question 23: How does the birth rate relate to the size of the armed forces?

# Calculate the correlation between Birth_Rate and Armed_Forces_size
correlation = world['Birth_Rate'].corr(world['Armed_Forces_size'])

print(f"Correlation between Birth Rate and Armed Forces Size: {correlation}")
```

```
Correlation between Birth Rate and Armed Forces Size: -0.1302758272337991
```

In [60]:
```
# Create a scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(world['Armed_Forces_size'], world['Birth_Rate'], alpha=0.7)
plt.title('Birth Rate vs. Armed Forces Size')
```

```
plt.xlabel('Armed Forces Size')
plt.ylabel('Birth Rate')
plt.grid(True)
plt.show()
```



Conclusion / Insights

- China is the highest population country in the World
- Palestinian National Authority is the lowest population country in the world
- There is a slight correlation between "Population" and "Density" with -0.01794615744859417
- The relationship between "GDP" and "Co2 Emission" is 0.9169960708699614

```
In [ ]:
```