Military Expenditure Analysis

Welcome to my exploratory data analysis (EDA) project on global military expenditures. In this notebook, I will explore various trends and patterns in military spending across different countries and time periods using Python. The objectives are to analyze historical trends in military expenditures, compare military spending across countries, and investigate the correlation between GDP and military expenditure. This analysis will utilize tools and libraries such as Pandas, Matplotlib, and Seaborn to provide insights and visualizations.

Import Library

In [5]:

```
import pandas as pd
```

In [7]:

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

Uploading Csv fle

In [9]:

```
df = pd.read_csv(r"C:\Users\Syed Arif\OneDrive\Desktop\Military expenditure.
```

Data Preprocessing

head()

head is used show to the By default = 5 rows in the dataset

In [14]:

```
df.head()
```

Loading [MathJax]/extensions/Safe.js

Out[14]:

	country	iso3c	iso2c	year	Military expenditure (current USD)	Military expenditure (% of general government expenditure)	Military expenditure (% of GDP)	a
0	Afghanistan	AFG	AF	1970	2.939586e+06	NaN	1.629606	
1	Afghanistan	AFG	AF	1971	NaN	NaN	NaN	
2	Afghanistan	AFG	AF	1972	NaN	NaN	NaN	
3	Afghanistan	AFG	AF	1973	3.341272e+06	NaN	1.868910	
4	Afghanistan	AFG	AF	1974	3.581366e+06	NaN	1.610825	

.tail()

tail is used to show rows by Descending order

In [18]:

```
df.tail()
```

Out[18]:

	country	iso3c	iso2c	year	Military expenditure (current USD)	Military expenditure (% of general government expenditure)	Military expenditure (% of GDP)
13561	Zimbabwe	ZWE	ZW	2016	358065000.0	7.363048	1.742494
13562	Zimbabwe	ZWE	ZW	2017	340522000.0	5.519890	1.544948
13563	Zimbabwe	ZWE	ZW	2018	420364000.0	5.607096	1.222795
13564	Zimbabwe	ZWE	ZW	2019	546939000.0	2.099816	0.698601
13565	Zimbabwe	ZWE	ZW	2020	NaN	NaN	NaN

shape

It show the total no of rows & Column in the dataset

```
In [22]: df.shape
```

```
Out[22]: (13566, 9)
```

Columns

It show the no of each Column

```
In [26]: df.columns
```

```
Out[26]: Index(['country', 'iso3c', 'iso2c', 'year',
                'Military expenditure (current USD)',
                'Military expenditure (% of general government expenditure)',
                'Military expenditure (% of GDP)', 'adminregion', 'incomeLevel'],
               dtype='object')
```

.dtypes

This Attribute show the data type of each column

```
In [30]: df.dtypes
```

```
Out[30]: country                                                        object
         iso3c                                                          object
         iso2c                                                          object
         year                                                            int64
         Military expenditure (current USD)                            float64
         Military expenditure (% of general government expenditure)    float64
         Military expenditure (% of GDP)                               float64
         adminregion                                                    object
         incomeLevel                                                    object
         dtype: object
```

.unique()

In a column, It show the unique value of specific column.

```
In [36]: df["country"].unique()
```

```
Out[36]: array(['Afghanistan', 'Africa Eastern and Southern',
                'Africa Western and Central', 'Albania', 'Algeria',
                'American Samoa', 'Andorra', 'Angola', 'Antigua and Barbuda',
                'Arab World', 'Argentina', 'Armenia', 'Aruba', 'Australia',
                'Austria', 'Azerbaijan', 'Bahamas, The', 'Bahrain', 'Bangladesh',
                'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin', 'Bermuda',
                'Bhutan', 'Bolivia', 'Bosnia and Herzegovina', 'Botswana',
                'Brazil', 'British Virgin Islands', 'Brunei Darussalam',
                'Bulgaria', 'Burkina Faso', 'Burundi', 'Cabo Verde', 'Cambodia',
                'Cameroon', 'Canada', 'Caribbean small states', 'Cayman Islands',
                'Central African Republic', 'Central Europe and the Baltics',
                'Chad', 'Channel Islands', 'Chile', 'China', 'Colombia', 'Comoros',
                'Congo, Dem. Rep.', 'Congo, Rep.', 'Costa Rica', "Cote d'Ivoire",
                'Croatia', 'Cuba', 'Curacao', 'Cyprus', 'Czech Republic',
                'Denmark', 'Djibouti', 'Dominica', 'Dominican Republic',
                'Early-demographic dividend', 'East Asia & Pacific',
                'East Asia & Pacific (IDA & IBRD countries)',
                'East Asia & Pacific (excluding high income)', 'Ecuador',
                'Egypt, Arab Rep.', 'El Salvador', 'Equatorial Guinea', 'Eritrea',
                'Estonia', 'Eswatini', 'Ethiopia', 'Euro area',
                'Europe & Central Asia',
                'Europe & Central Asia (IDA & IBRD countries)',
                'Europe & Central Asia (excluding high income)', 'European Union',
                'Faroe Islands', 'Fiji', 'Finland',
                'Fragile and conflict affected situations', 'France',
                'French Polynesia', 'Gabon', 'Gambia, The', 'Georgia', 'Germany',
                'Ghana', 'Gibraltar', 'Greece', 'Greenland', 'Grenada', 'Guam',
                'Guatemala', 'Guinea', 'Guinea-Bissau', 'Guyana', 'Haiti',
                'Heavily indebted poor countries (HIPC)', 'High income',
                'Honduras', 'Hong Kong SAR, China', 'Hungary', 'IBRD only',
                'IDA & IBRD total', 'IDA blend', 'IDA only', 'IDA total',
                'Iceland', 'India', 'Indonesia', 'Iran, Islamic Rep.', 'Iraq',
                'Ireland', 'Isle of Man', 'Israel', 'Italy', 'Jamaica', 'Japan',
                'Jordan', 'Kazakhstan', 'Kenya', 'Kiribati',
                "Korea, Dem. People's Rep.", 'Korea, Rep.', 'Kosovo', 'Kuwait',
                'Kyrgyz Republic', 'Lao PDR', 'Late-demographic dividend',
                'Latin America & Caribbean',
                'Latin America & Caribbean (excluding high income)',
                'Latin America & the Caribbean (IDA & IBRD countries)', 'Latvia',
                'Least developed countries: UN classification', 'Lebanon',
                'Lesotho', 'Liberia', 'Libya', 'Liechtenstein', 'Lithuania',
                'Low & middle income', 'Low income', 'Lower middle income',
                'Luxembourg', 'Macao SAR, China', 'Madagascar', 'Malawi',
                'Malaysia', 'Maldives', 'Mali', 'Malta', 'Marshall Islands',
                'Mauritania', 'Mauritius', 'Mexico', 'Micronesia, Fed. Sts.',
                'Middle East & North Africa',
                'Middle East & North Africa (IDA & IBRD countries)',
                'Middle East & North Africa (excluding high income)',
                'Middle income', 'Moldova', 'Monaco', 'Mongolia', 'Montenegro',
                'Morocco', 'Mozambique', 'Myanmar', 'Namibia', 'Nauru', 'Nepal',
                'Netherlands', 'New Caledonia', 'New Zealand', 'Nicaragua',
                'Niger', 'Nigeria', 'North America', 'North Macedonia',
                'Northern Mariana Islands', 'Norway', 'Not classified',
                'OECD members', 'Oman', 'Other small states',
                'Pacific island small states', 'Pakistan', 'Palau', 'Panama',
                 apua New Guinea', 'Paraguay', 'Peru', 'Philippines', 'Poland',
```

Loading [MathJax]/extensions/Safe.js

```
        'Portugal', 'Post-demographic dividend',
        'Pre-demographic dividend', 'Puerto Rico', 'Qatar', 'Romania',
        'Russian Federation', 'Rwanda', 'Samoa', 'San Marino',
        'Sao Tome and Principe', 'Saudi Arabia', 'Senegal', 'Serbia',
        'Seychelles', 'Sierra Leone', 'Singapore',
        'Sint Maarten (Dutch part)', 'Slovak Republic', 'Slovenia',
        'Small states', 'Solomon Islands', 'Somalia', 'South Africa',
        'South Asia', 'South Asia (IDA & IBRD)', 'South Sudan', 'Spain',
        'Sri Lanka', 'St. Kitts and Nevis', 'St. Lucia',
        'St. Martin (French part)', 'St. Vincent and the Grenadines',
        'Sub-Saharan Africa', 'Sub-Saharan Africa (IDA & IBRD countries)',
        'Sub-Saharan Africa (excluding high income)', 'Sudan', 'Suriname',
        'Sweden', 'Switzerland', 'Syrian Arab Republic', 'Tajikistan',
        'Tanzania', 'Thailand', 'Timor-Leste', 'Togo', 'Tonga',
        'Trinidad and Tobago', 'Tunisia', 'Turkey', 'Turkmenistan',
        'Turks and Caicos Islands', 'Tuvalu', 'Uganda', 'Ukraine',
        'United Arab Emirates', 'United Kingdom', 'United States',
        'Upper middle income', 'Uruguay', 'Uzbekistan', 'Vanuatu',
        'Venezuela, RB', 'Vietnam', 'Virgin Islands (U.S.)',
        'West Bank and Gaza', 'World', 'Yemen, Rep.', 'Zambia', 'Zimbabwe'],
       dtype=object)
```

.nuique()

It will show the total no of unque value from whole data frame

```
In [40]: df.nunique()
```

```
Out[40]: country                                                     266
         iso3c                                                       264
         iso2c                                                       263
         year                                                         51
         Military expenditure (current USD)                         8407
         Military expenditure (% of general government expenditure)    5041
         Military expenditure (% of GDP)                            8428
         adminregion                                                   6
         incomeLevel                                                   6
         dtype: int64
```

.describe()

It show the Count, mean , median etc

```
In [44]: df.describe()
```

Out[44]:

	year	Military expenditure (current USD)	Military expenditure (% of general government expenditure)	Military expenditure (% of GDP)
count	13566.000000	8.769000e+03	5282.000000	8777.000000
mean	1995.000000	3.676065e+10	7.658514	2.722603
std	14.720144	1.422587e+11	6.048938	2.921869
min	1970.000000	0.000000e+00	0.000000	0.000000
25%	1982.000000	1.200000e+08	3.665614	1.327207
50%	1995.000000	1.148446e+09	6.033675	2.013005
75%	2008.000000	8.533090e+09	9.996249	3.218261
max	2020.000000	1.928852e+12	57.478098	117.349823

.value_counts

It Shows all the unique values with their count

In [54]:

```
df["incomeLevel"].value_counts()
```

Out[54]:

```
incomeLevel
High income            4029
Upper middle income    2805
Lower middle income    2805
Aggregates             2397
Low income             1377
Not classified           51
Name: count, dtype: int64
```

isnull()

It shows the how many null values

In [52]:

```
df.isnull()
```

Out[52]:

	country	iso3c	iso2c	year	Military expenditure (current USD)	Military expenditure (% of general government expenditure)	Military expenditure (% of GDP)	a
0	False	False	False	False	False	True	False	
1	False	False	False	False	True	True	True	
2	False	False	False	False	True	True	True	
3	False	False	False	False	False	True	False	
4	False	False	False	False	False	True	False	
...	...	...	...	...	...	...	...	
13561	False	False	False	False	False	False	False	
13562	False	False	False	False	False	False	False	
13563	False	False	False	False	False	False	False	
13564	False	False	False	False	False	False	False	
13565	False	False	False	False	True	True	True	

13566 rows × 9 columns

.info()

To Show Data type of each column

In [42]:

```
df.info()
```

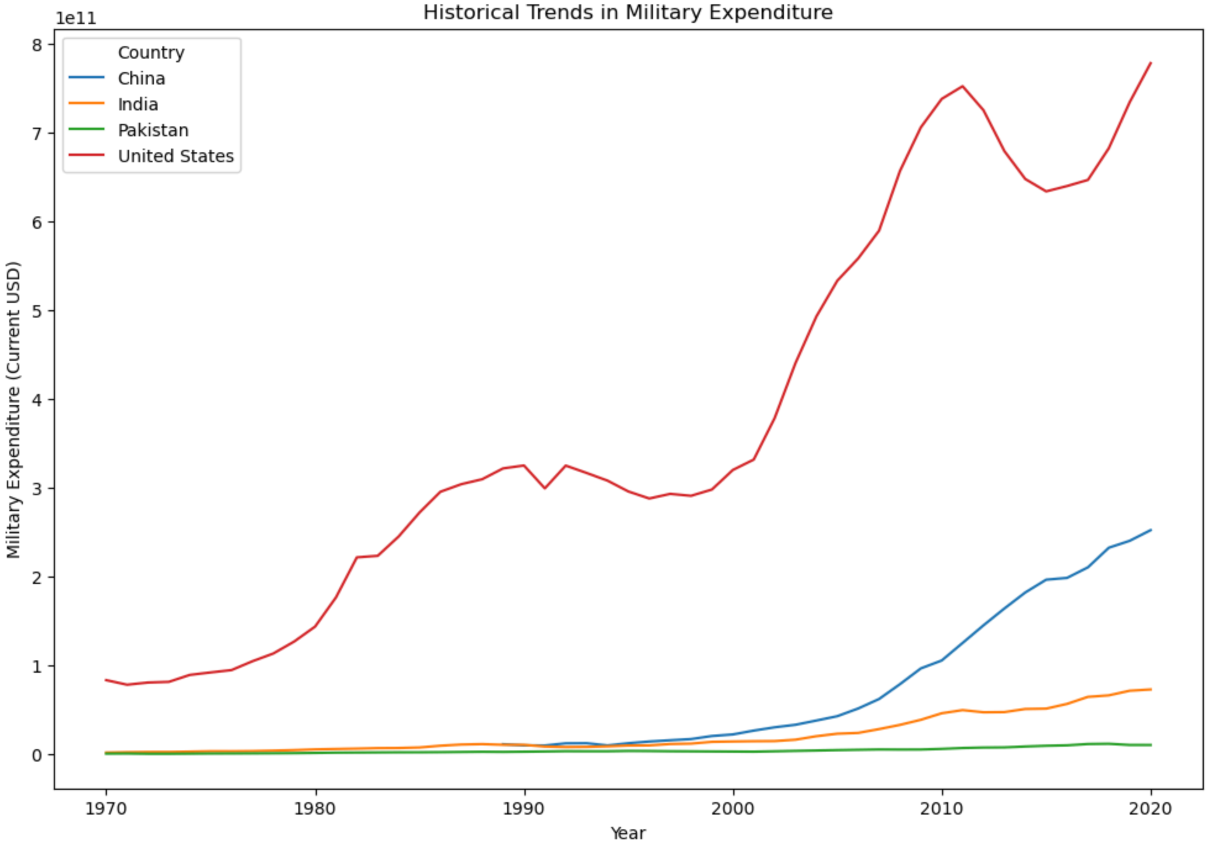
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44335 entries, 0 to 44334
Data columns (total 8 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   date       44335 non-null  object 
 1   home_team  44335 non-null  object 
 2   away_team  44335 non-null  object 
 3   team       44335 non-null  object 
 4   scorer     44286 non-null  object 
 5   minute     44076 non-null  float64
 6   own_goal   44335 non-null  bool   
 7   penalty    44335 non-null  bool   
dtypes: bool(2), float64(1), object(5)
memory usage: 2.1+ MB
```

What are the historical trends in military expenditures for different

countries?

In [69]:
```
# Filter data for a few example countries
countries = ['United States', 'China', 'Russia', 'India','Pakistan']
filtered_data = df[df['country'].isin(countries)]

# Plot historical trends
plt.figure(figsize=(12, 8))
sns.lineplot(data=filtered_data, x='year', y='Military expenditure (current
plt.title('Historical Trends in Military Expenditure')
plt.xlabel('Year')
plt.ylabel('Military Expenditure (Current USD)')
plt.legend(title='Country')
plt.show()
```



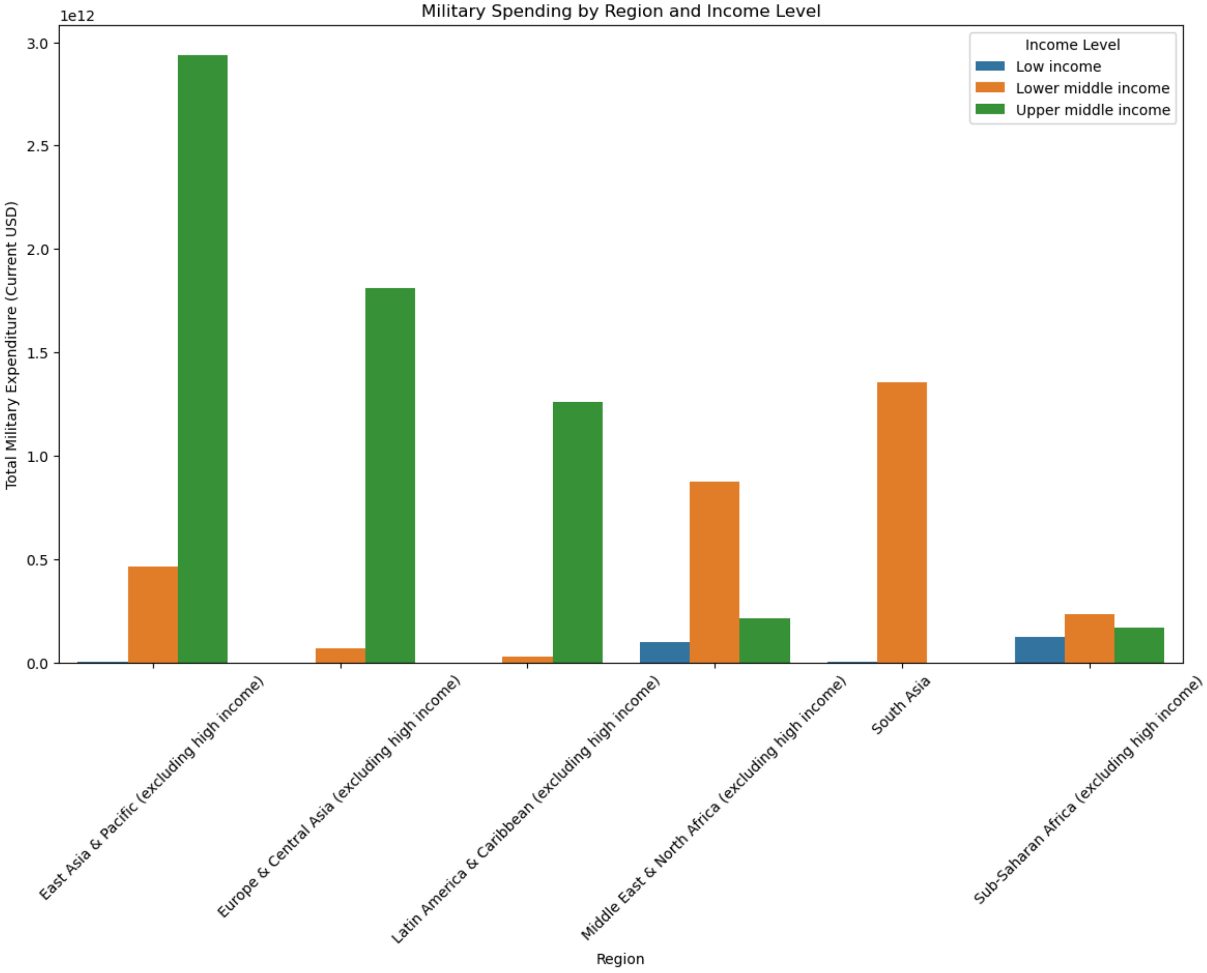
How does military spending vary across different regions and income levels?

In [74]:
```
# Aggregate data by region and income level
region_income_data = df.groupby(['adminregion', 'incomeLevel'])['Military ex

# Plot the aggregated data
plt.figure(figsize=(14, 8))
sns.barplot(data=region_income_data, x='adminregion', y='Military expenditur
```

Loading [MathJax]/extensions/Safe.js

```
plt.title('Military Spending by Region and Income Level')
plt.xlabel('Region')
plt.ylabel('Total Military Expenditure (Current USD)')
plt.legend(title='Income Level')
plt.xticks(rotation=45)
plt.show()
```



Countries with the largest increase in military spending:

In [97]:

```
df.groupby('country').max()
max_min = df.groupby('country').max()['Military expenditure (current USD)']
top_10 = max_min.nlargest(10)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[97], line 2
      1 df.groupby('country').max()
----> 2 max_min = df.groupby('country').max()['Military expenditure (current 
USD)'] - rf.groupby('country').min()['Military expenditure (current USD)']
      3 top_10 = max_min.nlargest(10)

NameError: name 'rf' is not defined
```

Loading [MathJax]/extensions/Safe.js

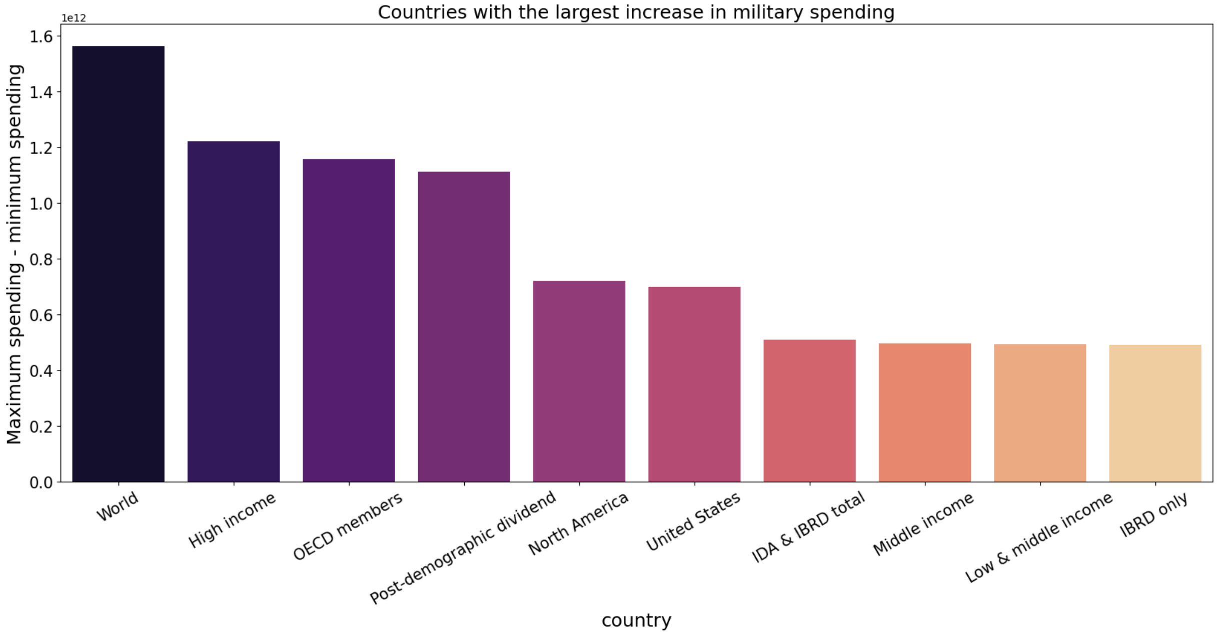
```
In [ ]: top_10.index
```

```
In [92]:
```

```
C:\Users\Syed Arif\AppData\Local\Temp\ipykernel_5100\2267883564.py:4: Future
Warning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.barplot(top_10,palette='magma')
```

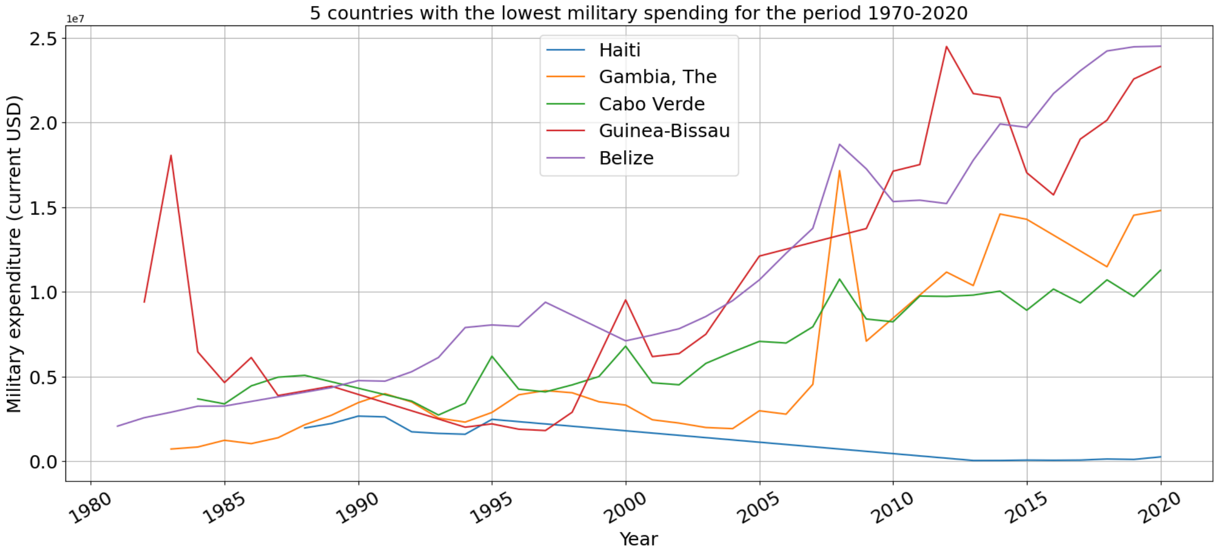


5 countries with the lowest military spending for the period 1970-2020 ?

```
In [103…
tf = df.loc[df['Military expenditure (current USD)'] > 0, :]
# Find the 5 countries with the smallest total military expenditure
smallest_5 = tf.groupby('country').sum()['Military expenditure (current USD)
smallest_5 = list(smallest_5.index)

def plotter_10(smallest_5):
    plt.figure(figsize=(20, 8))
    plt.title('5 countries with the lowest military spending for the period
    YEARS = [i for i in range(df['year'].min(), df['year'].max() + 1)]
    for country in smallest_5:
        country_data = df[df['country'] == country]
        sns.lineplot(x=country_data['year'], y=country_data['Military expenc
    plt.legend(fontsize=18)
    plt.xticks(rotation=30, fontsize=18)
    plt.yticks(fontsize=18)
    plt.xlabel('Year', fontsize=18)
    plt.ylabel('Military expenditure (current USD)', fontsize=18)
    plt.grid()

plotter_10(smallest_5)
plt.show()
```


5 countries with the lowest military spending for the period 1970-2020
1e7
Military expenditure (current USD)
Year
Haiti
Gambia, The
Cabo Verde
Guinea-Bissau
Belize