



```
In [ ]: #https://www.al texsoft.com/blog/data-collection-machine-learning/  
#https://pynative.com/python-sqlite-insert-into-table/  
#https://Labelyourdata.com/articles/data-collection-methods-AI
```

### Project Context:

Hi Yedou,

I remember you mentioned that you had some programming knowledge with Python and SQL. I have a small project in preparation for our Nonprofit Organization and would appreciate your help.

Could you use your Python and SQL skills to create a database for managing and analyzing our activities?

Best, Sydney

**Conversation:**

Yedou (Developer): "Hello Sydney! Have you collected or saved your activity data into a CSV file or database that I can use to extract the data needed for your project?"

Sydney (Client): "No, we don't have data stored this way."

Yedou (Developer): "Okay, no problem. I can pull the data I need from the API to complete the project for you. What are the objectives of the project?"

Sydney (Client): "Currently, our budgets are limited, so we want to identify which countries should be our priorities for humanitarian aid and determine the priority sectors within these countries."

## Data collection for data analysis and machine learning

**Data Extraction:** Data extraction is the process retrieving or pulling data from various sources and converting it into a usable and meaningful format for further analysis, reporting, or storage. **It is one of the most crucial steps in data management** that allows you to feed data into applications or analytics downstream.

### API Integration

---

---

Using Python for ETL Processes Because of the programming language's versatility, developers and data engineers can use Python to code nearly any ETL process. Including data aggregation. Python can easily handle important components of ETL operations, including indexed data structures and dictionaries

In [ ]:

# VOLUNTEER DATA MANAGEMENT USING PYTHON AND SQLITE DATABASE

In [217...]

```
import requests as r
import pandas as pd
import json
import numpy as np
import ast
import re
import sqlite3
!pip install country_converter
!pip install pycountry
!pip install geopy
!pip install iso3
!pip install Countrydetails
!pip install pycountry-convert
!pip show pycountry_convert
!pip install --upgrade pycountry_convert
!pip install shapely
!pip install pygwalker --upgrade
!pip install dash
!pip3 install geoplotlib
!pip install altair_viewer
!from shapely.geometry import Point
!pip install scikit-plot

# import pycountry
import ast
# import pycountry
def get_data(endpoint):
    """
    Function to make a GET request to the specified endpoint and return the response as pandas DataFrame.
    """
    try:
        response = r.get(base_url + endpoint)
        response.raise_for_status() # Raise an exception for HTTP errors
        return pd.DataFrame(response.json())
    except r.exceptions.RequestException as e:
        print("Error:", e)
```

```
return None

# Base URL for the UNICEF API
# https://rdmapi.unicef.org/api/doc/index.html
base_url = "https://rdmapi.unicef.org"

# Endpoint: /api/countries
df_countries = get_data("/api/countries")
if df_countries is not None:
    #print("\nDataFrame from /api/countries:")
    df_countries

df_languages = get_data("/api/languages")
if df_languages is not None:
    #print("\nDataFrame from /api/countries:")
    df_languages

# Endpoint: /api/countries/current
df_current = get_data("/api/countries/current")
if df_current is not None:
    #print("\nDataFrame from /api/countries/current:")
    df_current

# Endpoint: /api/countries/organizations
df_organizations = get_data("/api/countries/organizations")
if df_organizations is not None:
    #print("\nDataFrame from /api/countries/organizations:")
    df_organizations
```

```
Requirement already satisfied: country_converter in /usr/local/lib/python3.10/dist-packages (1.2)
Requirement already satisfied: pandas>=1.0 in /usr/local/lib/python3.10/dist-packages (from country_converter) (2.0.3)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0->country_converter) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0->country_converter) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0->country_converter) (2024.1)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0->country_converter) (1.25.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas>=1.0->country_converter) (1.16.0)
Requirement already satisfied: pycountry in /usr/local/lib/python3.10/dist-packages (24.6.1)
Requirement already satisfied: geopy in /usr/local/lib/python3.10/dist-packages (2.3.0)
Requirement already satisfied: geographiclib<3,>=1.52 in /usr/local/lib/python3.10/dist-packages (from geopy) (2.0)
ERROR: Could not find a version that satisfies the requirement iso3 (from versions: none)
ERROR: No matching distribution found for iso3
Requirement already satisfied: Countrydetails in /usr/local/lib/python3.10/dist-packages (1.0.8)
Requirement already satisfied: pycountry-convert in /usr/local/lib/python3.10/dist-packages (0.7.2)
Requirement already satisfied: pprintpp>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from pycountry-convert) (0.4.0)
Requirement already satisfied: pycountry>=16.11.27.1 in /usr/local/lib/python3.10/dist-packages (from pycountry-convert) (24.6.1)
Requirement already satisfied: pytest>=3.4.0 in /usr/local/lib/python3.10/dist-packages (from pycountry-convert) (7.4.4)
Requirement already satisfied: pytest-mock>=1.6.3 in /usr/local/lib/python3.10/dist-packages (from pycountry-convert) (3.14.0)
Requirement already satisfied: pytest-cov>=2.5.1 in /usr/local/lib/python3.10/dist-packages (from pycountry-convert) (5.0.0)
Requirement already satisfied: repoze.lru>=0.7 in /usr/local/lib/python3.10/dist-packages (from pycountry-convert) (0.7)
Requirement already satisfied: wheel>=0.30.0 in /usr/local/lib/python3.10/dist-packages (from pycountry-convert) (0.43.0)
Requirement already satisfied: iniconfig in /usr/local/lib/python3.10/dist-packages (from pytest>=3.4.0->pycountry-convert) (2.0.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from pytest>=3.4.0->pycountry-convert) (24.1)
Requirement already satisfied: pluggy<2.0,>=0.12 in /usr/local/lib/python3.10/dist-packages (from pytest>=3.4.0->pycountry-convert) (1.5.0)
Requirement already satisfied: exceptiongroup>=1.0.0rc8 in /usr/local/lib/python3.10/dist-packages (from pytest>=3.4.0->pycountry-convert) (1.2.1)
Requirement already satisfied: tomli>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from pytest>=3.4.0->pycountry-convert) (2.0.1)
Requirement already satisfied: coverage[toml]>=5.2.1 in /usr/local/lib/python3.10/dist-packages (from pytest-cov>=2.5.1->pycountry-convert) (7.5.4)
```

Name: pycountry-convert  
Version: 0.7.2  
Summary: Extension of Python package pycountry providing conversion functions.  
Home-page: <https://github.com/TuneLab/pycountry-convert>  
Author: TUNE Inc.  
Author-email: jefft@tune.com  
License: MIT License  
Location: /usr/local/lib/python3.10/dist-packages  
Requires: pprintpp, pycountry, pytest, pytest-cov, pytest-mock, repoze.lru, wheel  
Required-by:  
Requirement already satisfied: pycountry\_convert in /usr/local/lib/python3.10/dist-packages (0.7.2)  
Requirement already satisfied: pprintpp>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from pycountry\_convert) (0.4.0)  
Requirement already satisfied: pycountry>=16.11.27.1 in /usr/local/lib/python3.10/dist-packages (from pycountry\_convert) (24.6.1)  
Requirement already satisfied: pytest>=3.4.0 in /usr/local/lib/python3.10/dist-packages (from pycountry\_convert) (7.4.4)  
Requirement already satisfied: pytest-mock>=1.6.3 in /usr/local/lib/python3.10/dist-packages (from pycountry\_convert) (3.14.0)  
Requirement already satisfied: pytest-cov>=2.5.1 in /usr/local/lib/python3.10/dist-packages (from pycountry\_convert) (5.0.0)  
Requirement already satisfied: repoze.lru>=0.7 in /usr/local/lib/python3.10/dist-packages (from pycountry\_convert) (0.7)  
Requirement already satisfied: wheel>=0.30.0 in /usr/local/lib/python3.10/dist-packages (from pycountry\_convert) (0.43.0)  
Requirement already satisfied: iniconfig in /usr/local/lib/python3.10/dist-packages (from pytest>=3.4.0->pycountry\_convert) (2.0.0)  
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from pytest>=3.4.0->pycountry\_convert) (24.1)  
Requirement already satisfied: pluggy<2.0,>=0.12 in /usr/local/lib/python3.10/dist-packages (from pytest>=3.4.0->pycountry\_convert) (1.5.0)  
Requirement already satisfied: exceptiongroup>=1.0.0rc8 in /usr/local/lib/python3.10/dist-packages (from pytest>=3.4.0->pycountry\_convert) (1.2.1)  
Requirement already satisfied: tomli>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from pytest>=3.4.0->pycountry\_convert) (2.0.1)  
Requirement already satisfied: coverage[toml]>=5.2.1 in /usr/local/lib/python3.10/dist-packages (from pytest-cov>=2.5.1->pycountry\_convert) (7.5.4)  
Requirement already satisfied: shapely in /usr/local/lib/python3.10/dist-packages (2.0.4)  
Requirement already satisfied: numpy<3,>=1.14 in /usr/local/lib/python3.10/dist-packages (from shapely) (1.25.2)  
Requirement already satisfied: pygwalker in /usr/local/lib/python3.10/dist-packages (0.4.8.10)  
Requirement already satisfied: appdirs in /usr/local/lib/python3.10/dist-packages (from pygwalker) (1.4.4)  
Requirement already satisfied: arrow in /usr/local/lib/python3.10/dist-packages (from pygwalker) (1.3.0)  
Requirement already satisfied: astor in /usr/local/lib/python3.10/dist-packages (from pygwalker) (0.8.1)  
Requirement already satisfied: cachetools in /usr/local/lib/python3.10/dist-packages (from pygwalker) (5.3.3)  
Requirement already satisfied: duckdb<0.11.0,>=0.10.1 in /usr/local/lib/python3.10/dist-packages (from pygwalker) (0.1

0.3)  
Requirement already satisfied: gw-dsl-parser==0.1.49 in /usr/local/lib/python3.10/dist-packages (from pygwalker) (0.1.49)  
Requirement already satisfied: ipylab<=1.0.0 in /usr/local/lib/python3.10/dist-packages (from pygwalker) (1.0.0)  
Requirement already satisfied: ipython in /usr/local/lib/python3.10/dist-packages (from pygwalker) (7.34.0)  
Requirement already satisfied: ipywidgets in /usr/local/lib/python3.10/dist-packages (from pygwalker) (7.7.1)  
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from pygwalker) (3.1.4)  
Requirement already satisfied: kanaries-track==0.0.5 in /usr/local/lib/python3.10/dist-packages (from pygwalker) (0.0.5)  
Requirement already satisfied: numpy<2.0.0 in /usr/local/lib/python3.10/dist-packages (from pygwalker) (1.25.2)  
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from pygwalker) (24.1)  
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from pygwalker) (2.0.3)  
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from pygwalker) (5.9.5)  
Requirement already satisfied: pyarrow in /usr/local/lib/python3.10/dist-packages (from pygwalker) (14.0.2)  
Requirement already satisfied: pydantic in /usr/local/lib/python3.10/dist-packages (from pygwalker) (2.7.4)  
Requirement already satisfied: pytz in /usr/local/lib/python3.10/dist-packages (from pygwalker) (2023.4)  
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from pygwalker) (2.31.0)  
Requirement already satisfied: segment-analytics-python==2.2.3 in /usr/local/lib/python3.10/dist-packages (from pygwalker) (2.2.3)  
Requirement already satisfied: sqlalchemy in /usr/local/lib/python3.10/dist-packages (from pygwalker) (2.0.31)  
Requirement already satisfied: sqlglot>=23.15.8 in /usr/local/lib/python3.10/dist-packages (from pygwalker) (25.3.0)  
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from pygwalker) (4.12.2)  
Requirement already satisfied: wasmtime>=12.0.0 in /usr/local/lib/python3.10/dist-packages (from gw-dsl-parser==0.1.49->pygwalker) (22.0.0)  
Requirement already satisfied: backoff>=2.2.1 in /usr/local/lib/python3.10/dist-packages (from kanaries-track==0.0.5->pygwalker) (2.2.1)  
Requirement already satisfied: dateutils>=0.6.12 in /usr/local/lib/python3.10/dist-packages (from kanaries-track==0.0.5->pygwalker) (0.6.12)  
Requirement already satisfied: monotonic~1.5 in /usr/local/lib/python3.10/dist-packages (from segment-analytics-python==2.2.3->pygwalker) (1.6)  
Requirement already satisfied: python-dateutil~2.2 in /usr/local/lib/python3.10/dist-packages (from segment-analytics-python==2.2.3->pygwalker) (2.8.2)  
Requirement already satisfied: ipykernel>=4.5.1 in /usr/local/lib/python3.10/dist-packages (from ipywidgets->pygwalker) (5.5.6)  
Requirement already satisfied: ipython-genutils~0.2.0 in /usr/local/lib/python3.10/dist-packages (from ipywidgets->pygwalker) (0.2.0)  
Requirement already satisfied: traitlets>=4.3.1 in /usr/local/lib/python3.10/dist-packages (from ipywidgets->pygwalker) (5.7.1)  
Requirement already satisfied: widgetsnbextension~3.6.0 in /usr/local/lib/python3.10/dist-packages (from ipywidgets->pygwalker) (3.6.6)  
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from ipywidgets->pygwalker) (3.0.11)  
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.10/dist-packages (from ipython->pygwalker) (67.7.2)  
Requirement already satisfied: jedi>=0.16 in /usr/local/lib/python3.10/dist-packages (from ipython->pygwalker) (0.19.1)

Requirement already satisfied: decorator in /usr/local/lib/python3.10/dist-packages (from ipython->pygwalker) (4.4.2)  
Requirement already satisfied: pickleshare in /usr/local/lib/python3.10/dist-packages (from ipython->pygwalker) (0.7.5)  
Requirement already satisfied: prompt-toolkit!=3.0.0,!>=3.0.1,<3.1.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from ipython->pygwalker) (3.0.47)  
Requirement already satisfied: pygments in /usr/local/lib/python3.10/dist-packages (from ipython->pygwalker) (2.16.1)  
Requirement already satisfied: backcall in /usr/local/lib/python3.10/dist-packages (from ipython->pygwalker) (0.2.0)  
Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.10/dist-packages (from ipython->pygwalker) (0.1.7)  
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.10/dist-packages (from ipython->pygwalker) (4.9.0)  
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->pygwalker) (3.3.2)  
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->pygwalker) (3.7)  
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->pygwalker) (2.0.7)  
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->pygwalker) (2024.6.2)  
Requirement already satisfied: types-python-dateutil>=2.8.10 in /usr/local/lib/python3.10/dist-packages (from arrow->pygwalker) (2.9.0.20240316)  
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->pygwalker) (2.1.5)  
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas->pygwalker) (2024.1)  
Requirement already satisfied: annotated-types>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from pydantic->pygwalker) (0.7.0)  
Requirement already satisfied: pydantic-core==2.18.4 in /usr/local/lib/python3.10/dist-packages (from pydantic->pygwalker) (2.18.4)  
Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/python3.10/dist-packages (from sqlalchemy->pygwalker) (3.0.3)  
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.10/dist-packages (from ipykernel>=4.5.1->ipywidgets->pygwalker) (6.1.12)  
Requirement already satisfied: tornado>=4.2 in /usr/local/lib/python3.10/dist-packages (from ipykernel>=4.5.1->ipywidgits->pygwalker) (6.3.3)  
Requirement already satisfied: parso<0.9.0,>=0.8.3 in /usr/local/lib/python3.10/dist-packages (from jedi>=0.16->ipython->pygwalker) (0.8.4)  
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.10/dist-packages (from pexpect>4.3->ipython->pygwalker) (0.7.0)  
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-packages (from prompt-toolkit!=3.0.0,!>3.0.1,<3.1.0,>=2.0.0->ipython->pygwalker) (0.2.13)  
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil~2.2->segment-analytics-python==2.2.3->pygwalker) (1.16.0)  
Requirement already satisfied: importlib-resources>=5.10 in /usr/local/lib/python3.10/dist-packages (from wasmtime>=12.0.0->gw-dsl-parser==0.1.49->pygwalker) (6.4.0)  
Requirement already satisfied: notebook>=4.4.1 in /usr/local/lib/python3.10/dist-packages (from widgetsnbextension~3.6.0->ipywidgits->pygwalker) (6.5.5)  
Requirement already satisfied: pyzmq<25,>=17 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension)

```
bextension~=3.6.0->ipywidgets->pygwalker) (24.0.1)
Requirement already satisfied: argon2-cffi in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (23.1.0)
Requirement already satisfied: jupyter-core>=4.6.1 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (5.7.2)
Requirement already satisfied: nbformat in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (5.10.4)
Requirement already satisfied: nbconvert>=5 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (6.5.4)
Requirement already satisfied: nest-asyncio>=1.5 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (1.6.0)
Requirement already satisfied: Send2Trash>=1.8.0 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (1.8.3)
Requirement already satisfied: terminado>=0.8.3 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (0.18.1)
Requirement already satisfied: prometheus-client in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (0.20.0)
Requirement already satisfied: nbclassic>=0.4.7 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (1.1.0)
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.10/dist-packages (from jupyter-core>=4.6.1->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (4.2.2)
Requirement already satisfied: notebook-shim>=0.2.3 in /usr/local/lib/python3.10/dist-packages (from nbclassic>=0.4.7->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (0.2.4)
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (4.9.4)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (4.12.3)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (6.1.0)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (0.7.1)
Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (0.4)
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (0.3.0)
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (0.8.4)
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (0.10.0)
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (1.5.1)
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (1.3.0)
Requirement already satisfied: fastjsonschema>=2.15 in /usr/local/lib/python3.10/dist-packages (from nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (2.20.0)
```

```
Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/python3.10/dist-packages (from nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (4.19.2)
Requirement already satisfied: argon2-cffi-bindings in /usr/local/lib/python3.10/dist-packages (from argon2-cffi->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (21.2.0)
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=2.6->nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (23.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=2.6->nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (2023.12.1)
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=2.6->nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (0.35.1)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=2.6->nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (0.18.1)
Requirement already satisfied: jupyter-server<3,>=1.8 in /usr/local/lib/python3.10/dist-packages (from notebook-shim>=0.2.3->nbclassic>=0.4.7->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (1.24.0)
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from argon2-cffi-bindings->argon2-cffi->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (1.16.0)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4->nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (2.5)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach->nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (0.5.1)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-packages (from cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (2.22)
Requirement already satisfied: anyio<4,>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from jupyter-server<3,>=1.8->notebook-shim>=0.2.3->nbclassic>=0.4.7->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (3.7.1)
Requirement already satisfied: websocket-client in /usr/local/lib/python3.10/dist-packages (from jupyter-server<3,>=1.8->notebook-shim>=0.2.3->nbclassic>=0.4.7->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (1.8.0)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-packages (from anyio<4,>=3.1.0->jupyter-server<3,>=1.8->notebook-shim>=0.2.3->nbclassic>=0.4.7->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (1.3.1)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio<4,>=3.1.0->jupyter-server<3,>=1.8->notebook-shim>=0.2.3->nbclassic>=0.4.7->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pygwalker) (1.2.1)
Requirement already satisfied: dash in /usr/local/lib/python3.10/dist-packages (2.17.1)
Requirement already satisfied: Flask<3.1,>=1.0.4 in /usr/local/lib/python3.10/dist-packages (from dash) (2.2.5)
Requirement already satisfied: Werkzeug<3.1 in /usr/local/lib/python3.10/dist-packages (from dash) (3.0.3)
Requirement already satisfied: plotly>=5.0.0 in /usr/local/lib/python3.10/dist-packages (from dash) (5.15.0)
Requirement already satisfied: dash-html-components==2.0.0 in /usr/local/lib/python3.10/dist-packages (from dash) (2.0.0)
Requirement already satisfied: dash-core-components==2.0.0 in /usr/local/lib/python3.10/dist-packages (from dash) (2.0.0)
Requirement already satisfied: dash-table==5.0.0 in /usr/local/lib/python3.10/dist-packages (from dash) (5.0.0)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.10/dist-packages (from dash) (7.2.0)
Requirement already satisfied: typing-extensions>=4.1.1 in /usr/local/lib/python3.10/dist-packages (from dash) (4.12.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from dash) (2.31.0)
Requirement already satisfied: retrying in /usr/local/lib/python3.10/dist-packages (from dash) (1.3.4)
```

```
Requirement already satisfied: nest-asyncio in /usr/local/lib/python3.10/dist-packages (from dash) (1.6.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from dash) (67.7.2)
Requirement already satisfied: Jinja2>=3.0 in /usr/local/lib/python3.10/dist-packages (from Flask<3.1,>=1.0.4->dash) (3.1.4)
Requirement already satisfied: itsdangerous>=2.0 in /usr/local/lib/python3.10/dist-packages (from Flask<3.1,>=1.0.4->dash) (2.2.0)
Requirement already satisfied: click>=8.0 in /usr/local/lib/python3.10/dist-packages (from Flask<3.1,>=1.0.4->dash) (8.1.7)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly>=5.0.0->dash) (8.4.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from plotly>=5.0.0->dash) (24.1)
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from Werkzeug<3.1->dash) (2.1.5)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.10/dist-packages (from importlib-metadata->dash) (3.19.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->dash) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->dash) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->dash) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->dash) (2024.6.2)
Requirement already satisfied: six>=1.7.0 in /usr/local/lib/python3.10/dist-packages (from retrying->dash) (1.16.0)
Collecting geoplotlib
  Using cached geoplotlib-0.3.2.tar.gz (22 kB)
error: subprocess-exited-with-error

  × python setup.py egg_info did not run successfully.
    | exit code: 1
    ↴ See above for output.

note: This error originates from a subprocess, and is likely not a problem with pip.
Preparing metadata (setup.py) ... error
error: metadata-generation-failed

  × Encountered error while generating package metadata.
    ↴ See above for output.

note: This is an issue with the package mentioned above, not pip.
hint: See above for details.
Requirement already satisfied: altair_viewer in /usr/local/lib/python3.10/dist-packages (0.4.0)
Requirement already satisfied: altair in /usr/local/lib/python3.10/dist-packages (from altair_viewer) (4.2.2)
Requirement already satisfied: altair-data-server>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from altair_viewer) (0.4.1)
Requirement already satisfied: portpicker in /usr/local/lib/python3.10/dist-packages (from altair-data-server>=0.4.0->a
```

```
ltair_viewer) (1.5.2)
Requirement already satisfied: tornado in /usr/local/lib/python3.10/dist-packages (from altair-data-server>=0.4.0->altair_viewer) (6.3.3)
Requirement already satisfied: entrypoints in /usr/local/lib/python3.10/dist-packages (from altair->altair_viewer) (0.4)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from altair->altair_viewer) (3.1.4)
Requirement already satisfied: jsonschema>=3.0 in /usr/local/lib/python3.10/dist-packages (from altair->altair_viewer) (4.19.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from altair->altair_viewer) (1.25.2)
Requirement already satisfied: pandas>=0.18 in /usr/local/lib/python3.10/dist-packages (from altair->altair_viewer) (2.0.3)
Requirement already satisfied: toolz in /usr/local/lib/python3.10/dist-packages (from altair->altair_viewer) (0.12.1)
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair->altair_viewer) (23.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair->altair_viewer) (2023.12.1)
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair->altair_viewer) (0.35.1)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair->altair_viewer) (0.18.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.18->altair->altair_viewer) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.18->altair->altair_viewer) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.18->altair->altair_viewer) (2024.1)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->altair->altair_viewer) (2.1.5)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from portpicker->altair-data-server>=0.4.0->altair_viewer) (5.9.5)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas>=0.18->altair->altair_viewer) (1.16.0)
/bin/bash: line 1: from: command not found
Requirement already satisfied: scikit-plot in /usr/local/lib/python3.10/dist-packages (0.3.7)
Requirement already satisfied: matplotlib>=1.4.0 in /usr/local/lib/python3.10/dist-packages (from scikit-plot) (3.7.1)
Requirement already satisfied: scikit-learn>=0.18 in /usr/local/lib/python3.10/dist-packages (from scikit-plot) (1.2.2)
Requirement already satisfied: scipy>=0.9 in /usr/local/lib/python3.10/dist-packages (from scikit-plot) (1.11.4)
Requirement already satisfied: joblib>=0.10 in /usr/local/lib/python3.10/dist-packages (from scikit-plot) (1.4.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=1.4.0->scikit-plot) (1.2.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=1.4.0->scikit-plot) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=1.4.0->scikit-plot) (4.53.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=1.4.0->scikit-plot) (1.4.2)
```

```
ikit-plot) (1.4.5)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=1.4.0->scikit-plot) (1.25.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=1.4.0->scikit-plot) (24.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=1.4.0->scikit-plot) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=1.4.0->scikit-plot) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=1.4.0->scikit-plot) (2.8.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.18->scikit-plot) (3.5.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib>=1.4.0->scikit-plot) (1.16.0)
```

In [218...]: df\_current.head(2)

	countryId	iso3	m49	isCurrent	isSOWC	isCountDown	isPublished	cndcountryCode	regions	language2Name	sectorPriorities
0	1	ABW	533	True	False	False	True	ABW	[{"regionCndCode": "MDG_DEVPING"}, {"regionCnd..."]]	{"languageName": "English", "value": "Aruba", ...}	[]
1	2	AFG	4	True	False	True	True	AFG	[{"regionCndCode": "MDG_DEVPING"}, {"regionCnd..."]]	{"languageName": "English", "value": "Afghani...", ...}	[{"sectorId": 7}, {"sectorId": 18}]

In [218...]:

```
def extract_regions_values(data):
    regions_lists = []
    for row in data:
        regions_list = []
        for entry in row:
            if isinstance(entry, dict) and 'regionCndCode' in entry:
                split_list = entry['regionCndCode'].split('_')
                if len(split_list) > 1:
                    regions_list.append(split_list[1])
        regions_lists.append(regions_list)
    return regions_lists
regions_lists = extract_regions_values(df_current['regions'])
print(regions_lists)
```



[[ 'DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME'], [ 'DEVPING', 'LDC', 'LLDC', 'SOUTHCEASIA', 'ASIA', 'GLOBAL', 'SOUTHASIA', 'ROSA', 'SA', 'EMRO'], [ 'MIDDLEAFR', 'DEVPING', 'SSA', 'LDC', 'SSA', 'AFRICA', 'GLOBAL', 'ESARO', 'SSA', 'AFRO', 'ESA'], [ 'DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME', 'NC', 'LAC', 'LACRO'], [ 'NORTHEUR', 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL'], [ 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'SOUTHEUR', 'CEECIS', 'ECA', 'ECARO', 'EURO'], [ 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'SOUTHEUR', 'IND', 'ECA', 'EURO'], [ 'DEVPING', 'WESTASI', 'WASINA', 'ASIA', 'GLOBAL', 'WESTASIA', 'MENA', 'MENARO', 'EMRO'], [ 'SOUTHAME', 'DEVPING', 'AMERICAS', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], [ 'CAUCCEN', 'DEVPING', 'LLDC', 'WASINA', 'ASIA', 'GLOBAL', 'WESTASIA', 'CEECIS', 'ECA', 'ECARO', 'EURO'], [ 'OCEAN', 'DEVPING', 'SIDS', 'GLOBAL', 'OCEANIA', 'POLYNES'], [ 'DEVPING', 'SSA', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL'], [ 'DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], [ 'DEV', 'AUSNZ', 'GLOBAL', 'OCEANIA', 'IND', 'EAP', 'WPRO'], [ 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'WESTEUR', 'IND', 'ECA', 'EURO'], [ 'CAUCCEN', 'DEVPING', 'LLDC', 'WASINA', 'ASIA', 'GLOBAL', 'WESTASIA', 'CEECIS', 'ECA', 'ECARO', 'EURO'], [ 'SSA', 'LDC', 'LLDC', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL', 'ESARO', 'SSA', 'AFRO', 'ESA'], [ 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'WESTEUR', 'IND', 'ECA', 'EURO'], [ 'WESTAFR', 'DEVPING', 'SSA', 'LDC', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO'], [ 'SSA', 'AFRO'], [ 'DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME'], [ 'WESTAFR', 'DEVPING', 'SSA', 'LD', 'LLDC', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'SSA', 'AFRO'], [ 'DEVPING', 'LDC', 'SOUTHCEASIA', 'ASIA', 'GLOBAL', 'SOUTHASIA', 'ROSA', 'SA', 'SEARO'], [ 'DEV', 'NORTHAMEUR', 'EASTEUR', 'EUROPE', 'GLOBAL', 'CEECIS', 'ECA', 'ECARO', 'EURO'], [ 'DEVPING', 'WESTASI', 'WASINA', 'ASIA', 'GLOBAL', 'WESTASIA', 'MENA', 'MENARO', 'EMRO'], [ 'DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], [ 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'SOUTHEUR', 'CEECIS', 'ECA', 'ECARO', 'EURO'], [ 'DEVPING', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME'], [ 'DEV', 'NORTHAMEUR', 'EASTEUR', 'EUROPE', 'GLOBAL', 'CEECIS', 'ECA', 'ECARO', 'EURO'], [ 'DEVPING', 'SIDS', 'AMERICAS', 'CENTRALAME', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], [ 'DEV', 'NORTHAMEUR', 'AMERICAS', 'GLOBAL', 'NORTHAME'], [ 'SOUTHAME', 'DEVPING', 'LLDC', 'AMERICA', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], [ 'DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], [ 'DEVPING', 'EASOUTASI', 'ASIA', 'GLOBAL', 'SOUTHEASTASIA', 'EAPRO', 'EAP', 'WPRO'], [ 'DEVPING', 'LDC', 'LLDC', 'SOUTHCEASIA', 'ASIA', 'GLOBAL', 'SOUTHASIA', 'ROSA', 'SA', 'SEARO'], [ 'DEVPING', 'SSA', 'LLDC', 'SSA', 'AFRICA', 'GLOBAL', 'SOUTHAFR', 'ESARO', 'SSA', 'AFRO'], [ 'SSA', 'EPA'], [ 'MIDDLEAFR', 'DEVPING', 'SSA', 'LDC', 'LLDC', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'SSA', 'AFRO'], [ 'DEV', 'NORTHAMEUR', 'AMERICAS', 'GLOBAL', 'NORTHAME', 'IND', 'NA'], [ 'DEV', 'AUSNZ', 'GLOBAL', 'OCEANIA'], [ 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'WESTEUR', 'IND', 'ECA', 'EURO'], [ 'GLOBAL'], [ 'SOUTHAME', 'DEV', 'DEVPING', 'NORTHAMEUR', 'AMERICAS', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], [ 'DEVPING', 'EASTASIA', 'EASOUTASI', 'ASIA', 'EASTASIA', 'GLOBAL', 'EAPRO', 'EAP', 'WPRO'], [ 'WESTAFR', 'DEVPING', 'SSA', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'SSA', 'AFRO'], [ 'MIDDLEAFR', 'DEVPING', 'SSA', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'SSA', 'AFRO'], [ 'MIDDLEAFR', 'DEVPING', 'SSA', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'SSA', 'AFRO'], [ 'OCEAN', 'DEVPING', 'SIDS', 'GLOBAL', 'OCEANIA', 'POLYNES', 'EAPRO', 'EAP', 'WPRO'], [ 'SOUTHAME', 'DEVPING', 'AMERICAS', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], [ 'DEVPING', 'SSA', 'LDC', 'SIDS', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL', 'ESARO', 'SSA', 'AFRO', 'ESA'], [ 'WESTAFR', 'DEVPING', 'SSA', 'SIDS', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'SSA', 'AFRO'], [ 'DEVPING', 'AMERICAS', 'CENTRALAME', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], [ 'DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME'], [ 'DEV', 'AUSNZ', 'GLOBAL', 'OCEANIA'], [ 'DEVPING', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME'], [ 'DEV', 'WASINA', 'ASIA', 'GLOBAL', 'WESTASIA', 'IND', 'ECA', 'EURO'], [ 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'WESTEUR', 'IND', 'ECA', 'EURO'], [ 'DEVPING', 'SSA', 'LDC', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL', 'MENARO', 'SSA', 'EMRO'], [ 'DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], [ 'NORTHEUR', 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'IND', 'ECA', 'EURO'], [ 'DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], [ 'DEVPING', 'NORTHAFR', 'WASINA', 'AFRICA', 'GLOBAL', 'NORTHAFR', 'MENA', 'MENARO', 'AFRO'], [ 'SOUTHAME', 'DEVPING', 'AMERICAS', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], [ 'DEVPING', 'NORTHAFR', 'WASINA', 'AFRICA', 'GLOBAL', 'NORTHAFR', 'MENA', 'MENARO', 'EMRO'], [ 'DEVPING', 'SSA', 'LDC', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL']]

AFR', 'GLOBAL', 'ESARO', 'AFRO', 'ESA'], ['DEVPING', 'NORTHAFR', 'WASINAFR', 'AFRICA', 'GLOBAL', 'NORTHAFR'], ['DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'SOUTHEUR', 'IND', 'ECA', 'EURO'], ['NORTHEUR', 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBA L', 'IND', 'ECA', 'EURO'], ['DEVPING', 'SSA', 'LDC', 'LLDC', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL', 'ESARO', 'SSA', 'AFR O', 'ESA'], ['NORTHEUR', 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'IND', 'ECA', 'EURO'], ['OCEAN', 'DEVPING', 'SIDS', 'GLOBAL', 'MELANESIA', 'OCEANIA', 'EAPRO', 'EAP', 'WPRO'], ['SOUTHAME', 'DEVPING', 'AMERICAS', 'GLOBAL', 'LATINAME'], ['DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'WESTEUR', 'IND', 'ECA', 'EURO'], ['NORTHEUR', 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL'], ['OCEAN', 'DEVPING', 'SIDS', 'GLOBAL', 'MICRONESIA', 'OCEANIA', 'EAPRO', 'EAP', 'WPRO'], ['MIDDLEAFR', 'DEVP ING', 'SSA', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'SSA', 'AFRO'], ['NORTHEUR', 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'IND', 'ECA', 'EURO'], ['CAUCCEN', 'DEVPING', 'WASINAFR', 'ASIA', 'GLOBAL', 'WESTASIA', 'CEECIS', 'ECA', 'ECARO', 'EUR O'], ['NORTHEUR', 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL'], ['WESTAFR', 'DEVPING', 'SSA', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'AFRO'], ['DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'SOUTHEUR'], ['WESTAFR', 'DEVPING', 'SSA', 'SSA', 'LDC', 'SSA', 'AF RICA', 'GLOBAL', 'WCARO', 'SSA', 'AFRO'], ['DEVPING', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME'], ['WESTAFR', 'DEVPIN G', 'SSA', 'LDC', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'AFRO'], ['WESTAFR', 'DEVPING', 'SSA', 'LDC', 'SIDS', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'AFRO'], ['DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'SOUTHEUR', 'IND', 'ECA', 'EURO'], ['DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBA L', 'LATINAME', 'LACRO', 'LAC'], ['DEV', 'NORTHAMEUR', 'AMERICAS', 'GLOBAL', 'NORTHAME'], ['DEVPING', 'AMERICAS', 'CENT RALAME', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], ['SOUTHAME', 'DEVPING', 'AMERICAS', 'GLOBAL', 'LATINAME'], ['OCEAN', 'DE VPING', 'SIDS', 'GLOBAL', 'MICRONESIA', 'OCEANIA'], ['SOUTHAME', 'DEVPING', 'SIDS', 'AMERICAS', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], ['DEVPING', 'EASTASIA', 'EASOUTASI', 'ASIA', 'EASTASIA', 'GLOBAL'], ['DEV', 'AUSNZ', 'GLOBAL', 'OCEANI A'], ['DEVPING', 'AMERICAS', 'CENTRALAME', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], ['DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'SOUTHEUR', 'CEECIS', 'ECA', 'ECARO', 'EURO'], ['DEVPING', 'LDC', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAM E', 'LACRO', 'LAC'], ['DEV', 'NORTHAMEUR', 'EASTEUR', 'EUROPE', 'GLOBAL', 'IND', 'ECA', 'EURO'], ['DEVPING', 'EASOUTAS I', 'ASIA', 'GLOBAL', 'SOUTHEASTASIA', 'EAPRO', 'EAP', 'SEARO'], ['NORTHEUR', 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL'], ['DEVPING', 'SOUTHCEASIA', 'ASIA', 'GLOBAL', 'SOUTHASIA', 'ROSA', 'SA', 'SEARO'], ['DEVPING', 'SSA', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL'], ['NORTHEUR', 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'IND', 'ECA', 'EURO'], ['DEVPING', 'SOUTHCE ASIA', 'ASIA', 'GLOBAL', 'SOUTHASIA', 'MENA', 'MENARO', 'EMRO'], ['DEVPING', 'WESTASI', 'WASINAFR', 'ASIA', 'GLOBAL', 'WESTASIA', 'MENA', 'MENARO', 'EMRO'], ['NORTHEUR', 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'IND', 'ECA', 'EURO'], ['DEV', 'WASINAFR', 'ASIA', 'GLOBAL', 'WESTASIA', 'IND', 'MENA', 'EURO'], ['DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'SOUTH EUR', 'IND', 'ECA', 'EURO'], ['DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], ['NORTHEUR', 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL'], ['DEVPING', 'WESTASI', 'WASINAFR', 'ASIA', 'GLOBAL', 'WESTASIA', 'MENA', 'MENARO', 'EMRO'], ['DEV', 'EASOUTASI', 'ASIA', 'EASTASIA', 'GLOBAL', 'IND', 'EAP', 'WPRO'], ['CAUCCEN', 'DEVPING', 'LL DC', 'SOUTHCEASIA', 'ASIA', 'CENTRALASIA', 'GLOBAL', 'CEECIS', 'ECA', 'ECARO', 'EURO'], ['DEVPING', 'SSA', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL', 'ESARO', 'AFRO', 'ESA'], ['CAUCCEN', 'DEVPING', 'LLDC', 'SOUTHCEASIA', 'ASIA', 'CENTRALASI A', 'GLOBAL', 'CEECIS', 'ECA', 'ECARO', 'EURO'], ['DEVPING', 'EASOUTASI', 'LDC', 'ASIA', 'GLOBAL', 'SOUTHEASTASIA', 'EA PRO', 'EAP', 'WPRO'], ['OCEAN', 'DEVPING', 'LDC', 'SIDS', 'GLOBAL', 'MICRONESIA', 'OCEANIA', 'EAPRO', 'EAP', 'WPRO'], ['DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], ['DEVPING', 'EASTASIA', 'EASOUTASI', 'ASIA', 'GLOBAL', 'IND', 'EAP', 'WPRO'], ['DEVPING', 'WESTASI', 'WASINAFR', 'ASIA', 'GLOBAL', 'WESTASIA', 'MENA', 'MENARO', 'EMRO'], ['WESTAFR', 'DEVPIN G', 'SSA', 'LDC', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'AFRO'], ['DEVPING', 'NORTHAFR', 'WASINAFR', 'AFRICA', 'GLOBAL', 'NORTHAFR', 'MENA', 'MENARO', 'SSA', 'EMRO'], ['DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], ['DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'WESTEUR', 'IND', 'ECA'], ['DEVPING', 'SOUTHCEASIA', 'ASIA', 'GLOBAL', 'SOUTHASIA', 'ROSA', 'SA', 'SEARO'], ['DEVPING', 'SSA', 'LDC', 'LLDC', 'SSA', 'AFRICA', 'GLOBAL', 'SOUTHAFR', 'ESARO', 'SSA', 'AFRO', 'ESA'], ['NORTHEUR', 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'IND', 'ECA', 'EURO'], ['DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'WESTEUR', 'IND', 'ECA', 'EURO'], ['NORTHEUR', 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'IND', 'ECA', 'EURO']]

D', 'ECA', 'EURO'], ['DEVPING', 'EASTASIA', 'ASIA', 'EASTASIA', 'GLOBAL'], ['DEVPING', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME'], ['DEVPING', 'NORTHAFR', 'WASINAFR', 'AFRICA', 'GLOBAL', 'NORTHAFR', 'MENA', 'MENARO', 'EMRO'], ['DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'WESTEUR', 'IND', 'ECA', 'EURO'], ['DEV', 'LLDC', 'NORTHAMEUR', 'EASTEUR', 'EUROPE', 'GLOBAL', 'CEECIS', 'ECA', 'ECARO', 'EURO'], ['DEVPING', 'SSA', 'LDC', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL', 'ESARO', 'SSA', 'AFRO', 'ESA'], ['DEVPING', 'SIDS', 'SOUTHCEASIA', 'ASIA', 'GLOBAL', 'SOUTHASIA', 'ROSA', 'SA', 'SEARO'], ['DEVPING', 'AMERICAS', 'CENTRALAME', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], ['OCEAN', 'DEVPING', 'SIDS', 'GLOBAL', 'MICRONESIA', 'OCEANIA'], ['DEV', 'LLDC', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'SOUTHEUR', 'CEECIS', 'ECA', 'ECARO', 'EURO'], ['WESTAFR', 'DEVPING', 'SSA', 'LDC', 'LLDC', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'SSA', 'AFRO'], ['DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'SOUTHEUR', 'IND', 'ECA', 'EURO'], ['DEVPING', 'EASOUTASI', 'LD', 'ASIA', 'GLOBAL', 'SOUTHEASTASIA', 'EAPRO', 'EAP', 'SEARO'], ['DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'SOUTHEUR', 'CEECIS', 'ECA', 'ECARO', 'EURO'], ['DEVPING', 'EASTASIA', 'EASOUTASI', 'LLDC', 'ASIA', 'EASTASIA', 'GLOBAL', 'EAPRO', 'WPRO'], ['OCEAN', 'DEVPING', 'SIDS', 'GLOBAL', 'MICRONESIA', 'OCEANIA'], ['DEVPING', 'SSA', 'LDC', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL', 'ESARO', 'SSA', 'AFRO'], ['DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], ['DEVPING', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME'], ['DEVPING', 'SSA', 'SIDS', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL', 'ESARO', 'AFRO', 'ESA'], ['DEVPING', 'EASOUTASI', 'ASIA', 'GLOBAL', 'SOUTHEASTASIA', 'EAPRO', 'EAP', 'WPRO'], ['DEVPING', 'SSA', 'SSA', 'AFRIC', 'EASTAFR', 'GLOBAL'], ['DEVPING', 'SSA', 'AFRICA', 'GLOBAL', 'SOUTHAFR', 'ESARO', 'SSA', 'AFRO', 'ESA'], ['OCEAN', 'DEVPING', 'SIDS', 'GLOBAL', 'MELANESIA', 'OCEANIA'], ['WESTAFR', 'DEVPING', 'SSA', 'LDC', 'LLDC', 'SSA', 'AFRIC', 'GLOBAL', 'WCARO', 'SSA', 'AFRO'], ['DEV', 'AUSNZ', 'GLOBAL', 'OCEANIA'], ['WESTAFR', 'DEVPING', 'SSA', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'SSA', 'AFRO'], ['DEVPING', 'AMERICAS', 'CENTRALAME', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], ['OCEAN', 'DEVPING', 'SIDS', 'GLOBAL', 'OCEANIA', 'POLYNES'], ['EAPRO', 'EAP', 'WPRO'], ['DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'WESTEUR', 'IND', 'ECA', 'EURO'], ['NORTHEUR', 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'IND', 'ECA', 'EUR'], ['DEVPING', 'LDC', 'LLDC', 'SOUTHCEASIA', 'ASIA', 'GLOBAL', 'SOUTHASIA', 'ROSA', 'SA', 'SEARO'], ['OCEAN', 'DEVPING', 'SIDS', 'GLOBAL', 'MICRONESIA', 'OCEANIA'], ['EAPRO', 'EAP', 'WPRO'], ['DEV', 'AUSNZ', 'GLOBAL', 'OCEANIA', 'IND', 'EAP', 'WPRO'], ['DEVPING', 'WESTASI', 'WASINAFR', 'ASIA', 'GLOBAL', 'WESTASIA', 'MENA', 'MENARO', 'EMRO'], ['DEVPING', 'SOUTHCEASIA', 'ASIA', 'GLOBAL', 'SOUTHASIA', 'ROSA', 'SA', 'EMRO'], ['DEVPING', 'AMERICAS', 'CENTRALAME', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], ['OCEAN', 'DEVPING', 'GLOBAL', 'OCEANIA', 'POLYNES'], ['SOUTHAME', 'DEVPING', 'AMERICAS', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], ['DEVPING', 'EASOUTASI', 'ASIA', 'GLOBAL', 'SOUTHEASTASIA', 'EAPRO', 'EAP', 'WPRO'], ['OCEAN', 'DEVPING', 'SIDS', 'GLOBAL', 'MICRONESIA', 'OCEANIA'], ['EAPRO', 'EAP', 'WPRO'], ['OCEAN', 'DEVPING', 'SIDS', 'GLOBAL', 'MELANESIA', 'OCEANIA'], ['EAPRO', 'EAP', 'WPRO'], ['DEV', 'NORTHAMEUR', 'EASTEUR', 'EUROPE', 'GLOBAL', 'IND', 'ECA', 'EURO'], ['DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME'], ['DEVPING', 'EASTASIA', 'EASOUTASI', 'ASIA', 'GLOBAL', 'EAPRO', 'SEARO'], ['DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'SOUTHEUR', 'IND', 'ECA', 'EURO'], ['SOUTHAME', 'DEVPING', 'LLDC', 'AMERICAS', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], ['DEVPING', 'WESTASI', 'WASINAFR', 'ASIA', 'GLOBAL', 'WESTASIA', 'MENA', 'MENARO'], ['OCEAN', 'DEVPING', 'SIDS', 'GLOBAL', 'OCEANIA'], ['POLYNES'], ['DEVPING', 'WESTASI', 'WASINAFR', 'ASIA', 'GLOBAL', 'WESTASIA', 'MENA', 'MENARO', 'EMRO'], ['DEVPING', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL'], ['DEV', 'NORTHAMEUR', 'EASTEUR', 'EUROPE', 'GLOBAL', 'CEECIS', 'ECA', 'ECARO'], ['DEV', 'NORTHAMEUR', 'EASTEUR', 'EUROPE', 'GLOBAL', 'CEECIS', 'ECA', 'ECARO', 'EURO'], ['DEVPING', 'SSA', 'LDC', 'LLDC', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL', 'ESARO', 'AFRO'], ['DEVPING', 'WESTASI', 'WASINAFR', 'ASIA', 'GLOBAL', 'WESTASIA', 'MENA', 'MENARO', 'EMRO'], ['DEVPING', 'SSA', 'LDC', 'WASINAFR', 'AFRICA', 'GLOBAL', 'NORTHAFR', 'MENARO', 'SSA', 'EMRO'], ['WESTAFR', 'DEVPING', 'SSA', 'LDC', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'SSA', 'AFRO'], ['DEVPING', 'EASOUTASI', 'SIDS', 'ASIA', 'GLOBAL', 'SOUTHEASTASIA', 'EAPRO', 'EAP', 'WPRO'], ['SOUTHAME', 'DEVPING', 'AMERICAS', 'GLOBAL', 'LATINAME'], ['WESTAFR', 'DEVPING', 'SSA', 'SSA', 'AFRICA', 'GLOBAL'], ['NORTHEUR', 'DEV', 'NORTHAM', 'EUROPE', 'GLOBAL'], ['OCEAN', 'DEVPING', 'LDC', 'SIDS', 'GLOBAL', 'MELANESIA', 'OCEANIA'], ['EAPRO', 'EAP', 'WPRO'], ['WESTAFR', 'DEVPING', 'SSA', 'LDC', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'SSA', 'AFRO'], ['DEVPING', 'AMERICAS', 'GLOBAL', 'LATINAME'], ['WESTAFR', 'DEVPING', 'SSA', 'SSA', 'AFRICA', 'GLOBAL'], ['NORTHUR', 'DEV', 'NORTHAM', 'EUROPE', 'GLOBAL'], ['OCEAN', 'DEVPING', 'LDC', 'SIDS', 'GLOBAL', 'MELANESIA', 'OCEANIA'], ['EAPRO', 'EAP', 'WPRO'], ['WESTAFR', 'DEVPING', 'SSA', 'LDC', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'SSA', 'AFRO'], ['DEVPING', 'AMERICAS', 'GLOBAL', 'LATINAME']]

```
'CENTRALAME', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], ['DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'SOUTHEUR', 'IND', 'ECA', 'EURO'], ['DEVPING', 'SSA', 'LDC', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL', 'ESARO', 'AFRO', 'EMRO', 'ESA'], ['DEV', 'NORTHAMEUR', 'AMERICAS', 'GLOBAL', 'NORTHAME'], ['DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'SOUTHEUR', 'CEECIS', 'ECA', 'ECARO', 'EURO'], ['DEVPING', 'SSA', 'LDC', 'LLDC', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL', 'ESARO', 'SSA', 'EMRO', 'ESA'], ['MIDDLEAFR', 'DEVPING', 'SSA', 'LDC', 'SIDS', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'SSA', 'AFRO'], ['SOUTHAME', 'DEV', 'NORTHAMEUR', 'EASTEUR', 'EUROPE', 'GLOBA', 'IND', 'ECA', 'EURO'], ['DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'SOUTHEUR', 'IND', 'ECA', 'EURO'], ['NORTHEUR', 'DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'IND', 'ECA', 'EURO'], ['DEVPING', 'SSA', 'LLDC', 'SSA', 'AFRICA', 'GLOBAL', 'SOUTHAFR', 'ESARO', 'SSA', 'AFRO', 'ESA'], ['DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME'], ['DEVPING', 'SSA', 'SIDS', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL', 'ESARO', 'SSA', 'AFRO', 'ESA'], ['DEVPING', 'WESTASI', 'WASINAFR', 'ASIA', 'GLOBAL', 'WESTASIA', 'MENA', 'MENARO', 'EMRO'], ['DEVPING', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME', 'NC', 'LAC', 'LACRO'], ['MIDDLEAFR', 'DEVPING', 'SSA', 'LDC', 'LLDC', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'SSA', 'AFRO'], ['WESTAFR', 'DEVPING', 'SSA', 'LDC', 'SSA', 'AFRICA', 'GLOBAL', 'WCARO', 'SSA', 'AFRO'], ['DEVPING', 'EASOUTASI', 'ASIA', 'GLOBAL', 'SOUTHEASTASIA', 'EAPRO', 'EAP', 'SEARO'], ['CAUCCEN', 'DEVPING', 'LLDC', 'SOUTHCEASIA', 'ASIA', 'CENTRALASI', 'GLOBAL', 'CEECIS', 'ECA', 'ECARO', 'EURO'], ['OCEAN', 'DEVPING', 'GLOBAL', 'OCEANIA', 'POLYNES', 'EAPRO', 'EAP'], ['DEVPING', 'LLDC', 'SOUTHCEASIA', 'ASIA', 'CENTRALASIA', 'GLOBAL', 'CEECIS', 'ECA', 'ECARO', 'EURO'], ['DEVPING', 'EASOUTASI', 'LDC', 'SIDS', 'ASIA', 'GLOBAL', 'SOUTHEASTASIA', 'EAPRO', 'EAP', 'SEARO'], ['OCEAN', 'DEVPING', 'SIDS', 'GLOBAL', 'OCEANIA', 'POLYNES', 'EAPRO', 'EAP', 'WPRO'], ['DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], ['DEVPING', 'NORTHAFR', 'WASINAFR', 'AFRICA', 'GLOBAL', 'NORTHAFR', 'MENA', 'MENARO', 'EMRO'], ['DEVPING', 'WESTASI', 'WASINAFR', 'ASIA', 'GLOBAL', 'WESTASIA', 'CEECIS', 'ECA', 'ECARO', 'EURO'], ['OCEAN', 'DEVPING', 'LDC', 'SSA', 'AFRIC', 'EASTAFR', 'GLOBAL', 'ESARO', 'SSA', 'AFRO', 'ESA'], ['DEVPING', 'SSA', 'LDC', 'LLDC', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL', 'ESARO', 'SSA', 'AFRO', 'ESA'], ['DEV', 'NORTHAMEUR', 'EASTEUR', 'EUROPE', 'GLOBAL', 'CEECIS', 'ECA', 'ECARO', 'EURO'], ['OCEAN', 'DEVPING', 'GLOBAL', 'MICRONESIA', 'OCEANIA'], ['SOUTHAME', 'DEVPING', 'AMERICAS', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], ['DEV', 'NORTHAMEUR', 'AMERICAS', 'GLOBAL', 'NORTHAME', 'IND', 'NA'], ['CAUCCEN', 'DEVPING', 'LLDC', 'SOUTHCEASIA', 'ASIA', 'CENTRALASIA', 'GLOBAL', 'CEECIS', 'ECA', 'ECARO', 'EURO'], ['DEV', 'NORTHAMEUR', 'EUROPE', 'GLOBAL', 'SOUTHEUR', 'IND', 'ECA'], ['DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], ['SOUTHAME', 'DEVPING', 'AMERICAS', 'GLOBAL', 'LATINAME', 'LACRO', 'LAC'], ['DEVPING', 'SIDS', 'AMERICAS', 'CARIBB', 'GLOBAL', 'LATINAME'], ['DEVPING', 'EASOUTASI', 'ASIA', 'GLOBAL', 'SOUTHEASTASIA', 'EAPRO', 'EAP', 'WPRO'], ['OCEAN', 'DEVPING', 'LDC', 'SIDS', 'GLOBAL', 'MELANESIA', 'OCEANIA', 'EAPRO', 'EAP', 'WPRO'], ['OCEAN', 'DEVPING', 'GLOBAL', 'OCEANIA', 'POLYNES', 'EAPRO', 'EAP', 'WPRO'], ['GLOBAL', 'DEVPING', 'WESTASI', 'LDC', 'WASINAFR', 'ASIA', 'GLOBAL', 'WESTASIA', 'MENA', 'MENARO', 'EMRO'], ['DEVPING', 'SSA', 'SSA', 'AFRICA', 'GLOBAL', 'SOU', 'ESARO', 'SSA', 'AFRO', 'ESA'], ['DEVPING', 'SSA', 'LDC', 'LLDC', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL', 'ESARO', 'SSA', 'AFRO', 'ESA'], ['DEVPING', 'SSA', 'LLDC', 'SSA', 'AFRICA', 'EASTAFR', 'GLOBAL', 'ESARO', 'SSA', 'AFRO', 'ESA'], ['GLOBAL']]
```

In [220...]

```
regions_lists = pd.DataFrame(regions_lists)

# Concatenate lists for all rows, excluding NaN values
Regions_Lists = []
for index, row in regions_lists.iterrows():
    row_list = []
    for value in row:
        if not pd.isnull(value):
```

```
        row_list.append(value)
Regions_Lists.append(row_list)

# Create DataFrame
df_regions_lists = pd.DataFrame({'Regions_Lists': Regions_Lists})

# Print DataFrame
#print(df_regions_lists)
df_regions_lists.head()
```

Out[220]:

**Regions\_Lists**

0	[DEVPING, SIDS, AMERICAS, CARIBB, GLOBAL, LATI...
1	[DEVPING, LDC, LLDC, SOUTHCENASIA, ASIA, GLOBA...
2	[MIDDLEAFR, DEVPING, SSA, LDC, SSA, AFRICA, GL...
3	[DEVPING, SIDS, AMERICAS, CARIBB, GLOBAL, LATI...
4	[NORTHEUR, DEV, NORTHAMEUR, EUROPE, GLOBAL]

In [221...]

# df\_regions\_lists AND df\_sector\_lists

In [222...]

```
def extract_sector_values(data):
    sector_lists = []
    for row in data:
        sector_list = []
        for entry in row:
            if isinstance(entry, dict) and 'sectorId' in entry:
                sector_list.append(entry['sectorId'])
        sector_lists.append(sector_list)
    return sector_lists
sector_lists = extract_sector_values(df_current['sectorPriorities'])
print(sector_lists)
```

```
[], [7, 18], [], [], [], [], [], [], [], [2, 7], [], [7, 18], [], [7, 18], [7, 18], [],
[], [], [], [], [], [], [7, 18], [], [7, 18], [], [], [], [2, 7, 18], [2, 7, 18], [2,
7, 18], [7], [], [7], [], [], [], [2, 7, 18], [], [], [2, 7, 18], [2, 7, 18], [2, 7, 18],
[], [7], [], [2, 7, 18], [], [7, 18], [], [7, 18], [7, 18], [7, 18], [7, 18], [7, 18], [7, 18],
[], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18],
[], [7], [], [7], [], [2, 7, 18], [], [7, 18], [], [7, 18], [7, 18], [7, 18], [7, 18], [7, 18],
[], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18],
[], [7], [], [7], [], [2, 7, 18], [], [7, 18], [], [7, 18], [7, 18], [7, 18], [7, 18], [7, 18],
[], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18],
[], [7], [], [7], [], [2, 7, 18], [], [7, 18], [], [7, 18], [7, 18], [7, 18], [7, 18], [7, 18],
[], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18],
[], [7], [], [7], [], [2, 7, 18], [], [7, 18], [], [7, 18], [7, 18], [7, 18], [7, 18], [7, 18],
[], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18], [2, 7, 18]
```

In [223...]

```
# Concatenate lists for all rows, excluding NaN values
Sector_Lists = []
for row in sector_lists:
    row_list = [value for value in row if not pd.isnull(value)]
    Sector_Lists.append(row_list)

# Create DataFrame
df_sector_lists = pd.DataFrame({'Sector_Lists': Sector_Lists})

# Print DataFrame
df_sector_lists.head()
```

Out[223]:

	Sector_Lists
0	[]
1	[7, 18]
2	[]
3	[]
4	[]

In [223...]

In [224...]

```
# Convert the list of dictionaries into a string
df_current['language2Name'] = df_current['language2Name'].astype(str)

# Split the string where " " occurs
df_current['language2Name'] = df_current['language2Name'].str.split(":")
```

```
# Expand the splitted column into separate columns
coltransdf_current = df_current['language2Name'].apply(pd.Series)

coltransdf_current.head(2)
```

Out[224]:

	0	1	2	3	4	5	6	7	8	9	10
0	[{"languageName": "English", "value": "Aruba", "organization": "United Nations Children's Fund"}, {"language..."]]			"United Nations Children's Fund", {"language..."]}	'English', 'value' 'Aruba', 'organization'		'United Nations Statistical Division']	NaN	NaN	NaN	NaN
1	[{"languageName": "English", "value": "Afghanistan", "organization": "United Nations Children's Fund"}, {"language..."]]			"United Nations Children's Fund", {"language..."]}	'French', 'value' 'Afghanistan', 'organization'	"United Nations Children's Fund", {"language..."]}	'Spanish', 'value' 'Afganistán', 'organization'	"United Nations Children's Fund", {"language..."]}	'English', 'value' 'Afghanistan', 'organization'		

In [225...]

```
### Remove Special Characters from Column.
```

In [226...]

```
coltransdf_current = coltransdf_current.drop(columns=[0])
coltransdf_current = coltransdf_current.applymap(lambda x: x.replace(' ', '')).replace('value', '').replace('organization', '')
df_current = df_current.applymap(lambda x: x.strip().replace(' ', '').replace(',', '').replace('[', '').replace(']', '') if isinstance(x, str) else x)
coltransdf_current.applymap(lambda x: x.strip("''").strip("'''")) if isinstance(x, str) else x)
coltransdf_current.head(2)
```

Out[226]:

	1	2	3	4	5	6	7	8	9	10	11	12	13
0	'English' "	'Aruba' "	"United Nations Children's Fund"	'English' "	'Aruba' "	'United Nations Statistical Division'	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	'English' "	'Afghanistan' "	"United Nations Children's Fund"	'French' "	'Afghanistan' "	"United Nations Children's Fund"	'Spanish' "	'Afganistán' "	"United Nations Children's Fund"	'English' "	'Afghanistan' "	"United Nations Statistical Division'	

In [227...]

```
# Remove single quotes (') from the beginning and end of each string in columns 1 through 15
for col in range(1, 16):
    coltransdf_current[col] = coltransdf_current[col].apply(lambda x: x.strip(" '") if isinstance(x, str) else x)
    coltransdf_current[col] = coltransdf_current[col].apply(lambda x: x.strip("'''") if isinstance(x, str) and "'''" in x else x)
```

In [228...]

```
coltransdf_current.head(2)
```

Out[228]:

	1	2	3	4	5	6	7	8	9	10	11	12	13
0	English	Aruba	United Nations Children's Fund	English	Aruba	United Nations Statistical Division	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	English	Afghanistan	United Nations Children's Fund	French	Afghanistan	United Nations Children's Fund	Spanish	Afganistán	United Nations Children's Fund	English	Afghanistan	United Nations Statistical Division	English

In [229...]

```
# Language2Name   rename by Language1,Value1,Organization1,Language2,Value2,Organization2,Language3,Value3,Organization3,Language4,Value4
```

In [230...]

```
# Define the new column names
new_column_names = [f'Language{((col-1)//3+1)}' if col % 3 == 1 else
                    f'Value{((col-1)//3+1)}' if col % 3 == 2 else
                    f'Organization{((col-1)//3+1)}'
                    for col in range(1, 16)]
# Rename the columns of the DataFrame
coltransdf_current.columns = new_column_names
coltransdf_current.head(2)
```

Out[230]:

	Language1	Value1	Organization1	Language2	Value2	Organization2	Language3	Value3	Organization3	Language4	Value4
0	English	Aruba	United Nations Children's Fund	English	Aruba	United Nations Statistical Division	NaN	NaN	NaN	NaN	NaN
1	English	Afghanistan	United Nations Children's Fund	French	Afghanistan	United Nations Children's Fund	Spanish	Afganistán	United Nations Children's Fund	English	Afghanistan

In [231...]

```
# List of columns to drop
columns_to_drop = ['regions', 'language2Name', 'sectorPriorities']
# Drop the columns
df_current_new = df_current.drop(columns=columns_to_drop)
```

In [232...]

```
# Concatenate the DataFrames along the columns axis
CurrentcountriesTable = pd.concat([df_current_new, df_regions_lists, coltransdf_current, df_sector_lists], axis=1, sort=True)
# Print the resulting DataFrame
CurrentcountriesTable.head(2)
```

Out[232]:

	countryId	iso3	m49	isCurrent	isSOWC	isCountDown	isPublished	cndcountryCode	Regions_Lists	Language1	...	Language3	Val
0	1	ABW	533	True	False	False	True	ABW	[DEVPING, SIDS, AMERICAS, CARIBB, GLOBAL, LATI...]	English	...	NaN	
1	2	AFG	4	True	False	True	True	AFG	[DEVPING, LDC, LLDC, SOUTHCEASIA, ASIA, GLOBA...]	English	...	Spanish Afgani	

2 rows × 25 columns

In [232...]

```
#Select row where Regions_Lists contain DEVPING
CurrentcountriesTable[CurrentcountriesTable.Regions_Lists.apply(lambda x: 'DEVPING' in x)].head(2)
```

Out[233]:	countryId	iso3	m49	isCurrent	isSOWC	isCountDown	isPublished	cndcountryCode	Regions_Lists	Language1	...	Language3	Val
0	1	ABW	533	True	False	False	True	ABW	[DEVPING, SIDS, AMERICAS, CARIBB, GLOBAL, LATI...]	English	...	Nan	
1	2	AFG	4	True	False	True	True	AFG	[DEVPING, LDC, LLDC, SOUTHCEANASIA, ASIA, GLOBA...]	English	...	Spanish Afgani	

2 rows × 25 columns

In [233...]	
In [234...]	#####

## Fact Table and Dimension Tables of Unicef DW

**DataFrame to SQL Database Insertion for Data Analysts, BI Engineers, and Data Scientists Using Panda to\_sql**

In [235...]	<pre>import sqlite3 try:     conn = sqlite3.connect('unicef.db')     print("Opened database successfully") except sqlite3.Error as e:     print("Error connecting to SQLite database:", e)</pre>
	Opened database successfully
In [235...]	
In [236...]	DB ="/content/unicef.db"
In [237...]	<pre>DB ="/content/unicef.db" def create_and_load_table(df, table_name, db_name):</pre>

```

try:
    # Connect to SQLite database
    conn = sqlite3.connect(db_name)
    print("Opened database successfully")

    # Convert lists to strings
    df_copy = df.copy() # Create a copy to avoid modifying the original DataFrame
    df_copy['Regions_Lists'] = df_copy['Regions_Lists'].apply(json.dumps)
    df_copy['Sector_Lists'] = df_copy['Sector_Lists'].apply(json.dumps)

    # Write DataFrame to SQLite database
    df_copy.to_sql(table_name, conn, if_exists='replace', index=False)
    print("Table created and loaded successfully")

except sqlite3.Error as e:
    print("Error connecting to SQLite database:", e)
finally:
    # Close connection
    conn.close()

create_and_load_table(CurrentcountriesTable, "Actualcountries", DB)

```

Opened database successfully  
Table created and loaded successfully

In [238...]

```

#Fact Table of Unicef DW (Fact Table serves as the foundation for analytical queries and reporting.)
def select_table(db_name, table_name):
    try:
        # Connect to SQLite database
        conn = sqlite3.connect(db_name)
        print("Opened database successfully")

        # Create a cursor object to execute SQL queries
        cursor = conn.cursor()

        # Execute the SQL query to select the table
        cursor.execute(f"SELECT * FROM {table_name}")

        # Fetch all rows from the result
        rows = cursor.fetchall()

        # Print the selected table (or do further processing)
        for row in rows:
            print(row)
    
```

```
except sqlite3.Error as e:  
    print("Error connecting to SQLite database:", e)  
finally:  
    # Close connection  
    conn.close()  
select_table("unicef.db", "Actualcountries") ## Fact Table of Unicef DW (Fact Table serves as the foundation for analy
```

Opened database successfully

- (1, 'ABW', '533', 1, 0, 0, 1, 'ABW', '[{"DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME"]]', 'English', 'Aruba', "United Nations Children's Fund", 'English', 'Aruba', 'United Nations Statistical Division', None, None, None, None, None, None, None, None, None, '[]')
- (2, 'AFG', '4', 1, 0, 1, 1, 'AFG', '[{"DEVPING", "LDC", "LLDC", "SOUTHCEASIA", "ASIA", "GLOBAL", "SOUTHASIA", "ROSA", "SA", "EMRO"]]', 'English', 'Afghanistan', "United Nations Children's Fund", 'French', 'Afghanistan', "United Nations Children's Fund", 'Spanish', 'Afganistán', "United Nations Children's Fund", 'English', 'Afghanistan', 'United Nations Statistical Division', 'English', 'Afghanistan', 'World Bank', '[7, 18]')
- (3, 'AGO', '24', 1, 1, 1, 1, 'AGO', '[{"MIDDLEAFR", "DEVPING", "SSA", "LDC", "SSA", "AFRICA", "GLOBAL", "ESARO", "SSA", "AFRO", "ESA"]]', 'English', 'Angola', "United Nations Children's Fund", 'French', 'Angola', "United Nations Children's Fund", 'Spanish', 'Angola', "United Nations Children's Fund", 'English', 'Angola', 'United Nations Statistical Division', 'English', 'Angola', 'World Bank', '[]')
- (4, 'AIA', '660', 1, 1, 0, 1, 'AIA', '[{"DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME", "NC", "LAC", "LACRO"]]', 'English', 'Anguilla', "United Nations Children's Fund", 'French', 'Anguilla', "United Nations Children's Fund", 'Spanish', 'Anguilla', "United Nations Children's Fund", 'English', 'Anguilla', 'United Nations Statistical Division', None, None, None, '[]')
- (5, 'ALA', '248', 1, 0, 0, 0, 'ALA', '[{"NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "GLOBAL"]]', 'English', 'Åland Islands', "United Nations Children's Fund", 'French', "Îles d'Åland", "United Nations Children's Fund", 'Spanish', 'Islas Åland', "United Nations Children's Fund", 'English', 'Åland Islands', 'United Nations Statistical Division', None, None, None, '[]')
- (6, 'ALB', '8', 1, 1, 0, 1, 'ALB', '[{"DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEUR", "CEECIS", "ECA", "ECARO", "EURO"]]', 'English', 'Albania', "United Nations Children's Fund", 'French', 'Albanie', "United Nations Children's Fund", 'Spanish', 'Albania', "United Nations Children's Fund", 'English', 'Albania', 'United Nations Statistical Division', 'English', 'Albania', 'World Bank', '[]')
- (7, 'AND', '20', 1, 1, 0, 1, 'AND', '[{"DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEUR", "IND", "ECA", "EURO"]]', 'English', 'Andorra', "United Nations Children's Fund", 'French', 'Andorre', "United Nations Children's Fund", 'Spanish', 'Andorra', "United Nations Children's Fund", 'English', 'Andorra', 'United Nations Statistical Division', 'English', 'Andorra', 'World Bank', '[]')
- (8, 'ARE', '784', 1, 1, 0, 1, 'ARE', '[{"DEVPING", "WESTASIA", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "MENA", "MENARO", "EMRO"]]', 'English', 'United Arab Emirates', "United Nations Children's Fund", 'French', 'Émirats arabes unis', "United Nations Children's Fund", 'Spanish', 'Emiratos Árabes Unidos', "United Nations Children's Fund", 'English', 'United Arab Emirates', "United Nations Statistical Division", 'English', 'United Arab Emirates', 'World Bank', '[]')
- (9, 'ARG', '32', 1, 1, 0, 1, 'ARG', '[{"SOUTHAME", "DEVPING", "AMERICAS", "GLOBAL", "LATINAME", "LACRO", "LAC"]]', 'English', 'Argentina', "United Nations Children's Fund", 'French', 'Argentine', "United Nations Children's Fund", 'Spanish', 'Argentina', "United Nations Children's Fund", 'English', 'Argentina', 'United Nations Statistical Division', 'English', 'Argentina', 'World Bank', '[]')
- (10, 'ARM', '51', 1, 1, 0, 1, 'ARM', '[{"CAUCCEN", "DEVPING", "LLDC", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "CEECIS", "ECA", "ECARO", "EURO"]]', 'English', 'Armenia', "United Nations Children's Fund", 'French', 'Arménie', "United Nations Children's Fund", 'Spanish', 'Armenia', "United Nations Children's Fund", 'English', 'Armenia', 'United Nations Statistical Division', 'English', 'Armenia', 'World Bank', '[]')
- (11, 'ASM', '16', 1, 0, 0, 1, 'ASM', '[{"OCEAN", "DEVPING", "SIDS", "GLOBAL", "OCEANIA", "POLYNES"]]', 'English', 'American Samoa', "United Nations Children's Fund", 'English', 'American Samoa', 'United Nations Statistical Division', None, None, None, None, None, None, None, None, '[]')
- (12, 'ATF', '260', 1, 0, 0, 0, 'ATF', '[{"DEVPING", "SSA", "SSA", "AFRICA", "EASTAFR", "GLOBAL"]]', 'English', 'French Southern and Antarctic Territories', "United Nations Children's Fund", 'English', 'French Southern and Antarctic Territories', '[]')

```

ies', 'United Nations Statistical Division', None, None, None, None, None, None, None, None, None, '[]')
(13, 'ATG', '28', 1, 1, 0, 1, 'ATG', '[{"DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME", "LAC"}]', 'English', 'Antigua and Barbuda', "United Nations Children's Fund", 'French', 'Antigua-et-Barbuda', "United Nations Children's Fund", 'Spanish', 'Antigua y Barbuda', "United Nations Children's Fund", 'English', 'Antigua and Barbuda', 'United Nations Statistical Division', 'English', 'Antigua and Barbuda', 'World Bank', '[]')
(14, 'AUS', '36', 1, 1, 0, 1, 'AUS', '[{"DEV", "AUSNZ", "GLOBAL", "OCEANIA", "IND", "EAP", "WPRO"}]', 'English', 'Australia', "United Nations Children's Fund", 'French', 'Australie', "United Nations Children's Fund", 'Spanish', 'Australia', "United Nations Children's Fund", 'English', 'Australia', 'United Nations Statistical Division', 'English', 'Australia', 'World Bank', '[]')
(15, 'AUT', '40', 1, 1, 0, 1, 'AUT', '[{"DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "WESTEUR", "IND", "ECA", "EURO"}]', 'English', 'Austria', "United Nations Children's Fund", 'French', 'Autriche', "United Nations Children's Fund", 'Spanish', 'Austria', "United Nations Children's Fund", 'English', 'Austria', 'United Nations Statistical Division', 'English', 'Austria', 'World Bank', '[]')
(16, 'AZE', '31', 1, 1, 1, 1, 'AZE', '[{"CAUCCEN", "DEVPING", "LLDC", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "CEECIS", "ECA", "ECARO", "EURO"}]', 'English', 'Azerbaijan', "United Nations Children's Fund", 'French', 'Azerbaïdjan', "United Nations Children's Fund", 'Spanish', 'Azerbaiyán', "United Nations Children's Fund", 'English', 'Azerbaijan', 'United Nations Statistical Division', 'English', 'Azerbaijan', 'World Bank', '[]')
(17, 'BDI', '108', 1, 1, 1, 1, 'BDI', '[{"DEVPING", "SSA", "LDC", "LLDC", "SSA", "AFRICA", "EASTAFR", "GLOBAL", "ESARO", "SSA", "AFRO", "ESA"}]', 'English', 'Burundi', "United Nations Children's Fund", 'French', 'Burundi', "United Nations Children's Fund", 'Spanish', 'Burundi', "United Nations Children's Fund", 'English', 'Burundi', 'United Nations Statistical Division', 'English', 'Burundi', 'World Bank', '[2, 7]')
(18, 'BEL', '56', 1, 1, 0, 1, 'BEL', '[{"DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "WESTEUR", "IND", "ECA", "EURO"}]', 'English', 'Belgium', "United Nations Children's Fund", 'French', 'Belgique', "United Nations Children's Fund", 'Spanish', 'Bélgica', "United Nations Children's Fund", 'English', 'Belgium', 'United Nations Statistical Division', 'English', 'Belgium', 'World Bank', '[]')
(19, 'BEN', '204', 1, 1, 1, 1, 'BEN', '[{"WESTAFR", "DEVPING", "SSA", "LDC", "SSA", "AFRICA", "GLOBAL", "WCARO", "SSA", "AFRO"}]', 'English', 'Benin', "United Nations Children's Fund", 'French', 'Bénin', "United Nations Children's Fund", 'Spanish', 'Benin', "United Nations Children's Fund", 'English', 'Benin', 'United Nations Statistical Division', 'English', 'Benin', 'World Bank', '[7, 18]')
(20, 'BES', '535', 1, 0, 0, 1, 'BES', '[{"DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME"}]', 'English', 'Bonaire Sint Eustatius and Saba', "United Nations Children's Fund", 'English', 'Bonaire Sint Eustatius and Saba', 'United Nations Statistical Division', None, None, None, None, None, None, None, None, '[]')
(21, 'BFA', '854', 1, 1, 1, 1, 'BFA', '[{"WESTAFR", "DEVPING", "SSA", "LDC", "LLDC", "SSA", "AFRICA", "GLOBAL", "WCARO", "SSA", "AFRO"}]', 'English', 'Burkina Faso', "United Nations Children's Fund", 'French', 'Burkina Faso', "United Nations Children's Fund", 'Spanish', 'Burkina Faso', "United Nations Children's Fund", 'English', 'Burkina Faso', 'United Nations Statistical Division', 'English', 'Burkina Faso', 'World Bank', '[7, 18]')
(22, 'BGD', '50', 1, 1, 1, 1, 'BGD', '[{"DEVPING", "LDC", "SOUTHCEANASIA", "ASIA", "GLOBAL", "SOUTHASIA", "ROSA", "SA", "SEARO"}]', 'English', 'Bangladesh', "United Nations Children's Fund", 'French', 'Bangladesh', "United Nations Children's Fund", 'Spanish', 'Bangladesh', "United Nations Children's Fund", 'English', 'Bangladesh', 'United Nations Statistical Division', 'English', 'Bangladesh', 'World Bank', '[7, 18]')
(23, 'BGR', '100', 1, 1, 0, 1, 'BGR', '[{"DEV", "NORTHAMEUR", "EASTEUR", "EUROPE", "GLOBAL", "CEECIS", "ECA", "ECARO", "EURO"}]', 'English', 'Bulgaria', "United Nations Children's Fund", 'French', 'Bulgarie', "United Nations Children's Fund", 'Spanish', 'Bulgaria', "United Nations Children's Fund", 'English', 'Bulgaria', 'United Nations Statistical Division', 'English', 'Bulgaria', 'World Bank', '[]')
(24, 'BHR', '48', 1, 1, 0, 1, 'BHR', '[{"DEVPING", "WESTASI", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "MENA", "MENAR

```

0", "EMRO"]]', 'English', 'Bahrain', "United Nations Children's Fund", 'French', 'Bahreïn', "United Nations Children's Fund", 'Spanish', 'Bahreïn', "United Nations Children's Fund", 'English', 'Bahrain', 'United Nations Statistical Division', 'English', 'Bahrain', 'World Bank', '[]')

(25, 'BHS', '44', 1, 1, 0, 1, 'BHS', '[["DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME", "LACRO", "LAC"]]', 'English', 'Bahamas', "United Nations Children's Fund", 'French', 'Bahamas', "United Nations Children's Fund", 'Spanish', 'Bahamas', "United Nations Children's Fund", 'English', 'Bahamas', 'United Nations Statistical Division', 'English', 'Bahamas The', 'World Bank', '[]')

(26, 'BIH', '70', 1, 1, 0, 1, 'BIH', '[["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEUR", "CEECIS", "ECA", "ECARO", "EURO"]]', 'English', 'Bosnia and Herzegovina', "United Nations Children's Fund", 'French', 'Bosnie-Herzégovine', "United Nations Children's Fund", 'Spanish', 'Bosnia y Herzegovina', "United Nations Children's Fund", 'English', 'Bosnia and Herzegovina', 'United Nations Statistical Division', 'English', 'Bosnia and Herzegovina', 'World Bank', '[]')

(27, 'BLM', '652', 1, 0, 0, 1, 'BLM', '[["DEVPING", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME"]]', 'English', 'Saint Barthélemy', "United Nations Children's Fund", 'English', 'Saint Barthélemy', 'United Nations Statistical Division', None, None, None, None, None, None, None, '[]')

(28, 'BLR', '112', 1, 1, 0, 1, 'BLR', '[["DEV", "NORTHAMEUR", "EASTEUR", "EUROPE", "GLOBAL", "CEECIS", "ECA", "ECARO", "EURO"]]', 'English', 'Belarus', "United Nations Children's Fund", 'French', 'Bélarus', "United Nations Children's Fund", 'Spanish', 'Belarús', "United Nations Children's Fund", 'English', 'Belarus', 'United Nations Statistical Division', 'English', 'Belarus', 'World Bank', '[]')

(29, 'BLZ', '84', 1, 1, 0, 1, 'BLZ', '[["DEVPING", "SIDS", "AMERICAS", "CENTRALAME", "GLOBAL", "LATINAME", "LACRO", "LAC"]]', 'English', 'Belize', "United Nations Children's Fund", 'French', 'Belize', "United Nations Children's Fund", 'Spanish', 'Belice', "United Nations Children's Fund", 'English', 'Belize', 'United Nations Statistical Division', 'English', 'Belize', 'World Bank', '[]')

(30, 'BMU', '60', 1, 0, 0, 1, 'BMU', '[["DEV", "NORTHAMEUR", "AMERICAS", "GLOBAL", "NORTHAME"]]', 'English', 'Bermuda', "United Nations Children's Fund", 'English', 'Bermuda', 'United Nations Statistical Division', None, None, None, None, None, None, None, '[]')

(31, 'BOL', '68', 1, 1, 1, 1, 'BOL', '[["SOUTHAME", "DEVPING", "LLDC", "AMERICAS", "GLOBAL", "LATINAME", "LACRO", "LAC"]]', 'English', 'Bolivia (Plurinational State of)', "United Nations Children's Fund", 'French', 'Bolivie (État plurinational de)', "United Nations Children's Fund", 'Spanish', 'Bolivia (Estado Plurinacional de)', "United Nations Children's Fund", 'English', 'Bolivia (Plurinational State of)', 'United Nations Statistical Division', 'English', 'Bolivia', 'World Bank', '[]')

(32, 'BRA', '76', 1, 1, 1, 1, 'BRA', '[["SOUTHAME", "DEVPING", "AMERICAS", "GLOBAL", "LATINAME", "LACRO", "LAC"]]', 'English', 'Brazil', "United Nations Children's Fund", 'French', 'Brésil', "United Nations Children's Fund", 'Spanish', 'Brazil', "United Nations Children's Fund", 'English', 'Brazil', 'United Nations Statistical Division', 'English', 'Brazil', 'World Bank', '[]')

(33, 'BRB', '52', 1, 1, 0, 1, 'BRB', '[["DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME", "LACRO", "LAC"]]', 'English', 'Barbados', "United Nations Children's Fund", 'French', 'Barbade', "United Nations Children's Fund", 'Spanish', 'Barbados', "United Nations Children's Fund", 'English', 'Barbados', 'United Nations Statistical Division', 'English', 'Barbados', 'World Bank', '[]')

(34, 'BRN', '96', 1, 1, 0, 1, 'BRN', '[["DEVPING", "EASOUTASI", "ASIA", "GLOBAL", "SOUTHEASTASIA", "EAPRO", "EAP", "WPRO"]]', 'English', 'Brunei Darussalam', "United Nations Children's Fund", 'French', 'Brunei Darussalam', "United Nations Children's Fund", 'Spanish', 'Brunei Darussalam', "United Nations Children's Fund", 'English', 'Brunei Darussalam', 'United Nations Statistical Division', 'English', 'Brunei Darussalam', 'World Bank', '[]')

(35, 'BTN', '64', 1, 1, 0, 1, 'BTN', '[["DEVPING", "LDC", "LLDC", "SOUTHENASIA", "ASIA", "GLOBAL", "SOUTHASIA", "ROSA", "SA", "SEARO"]]', 'English', 'Bhutan', "United Nations Children's Fund", 'French', 'Bhoutan', "United Nations Children's Fund", 'Spanish', 'Bhután', "United Nations Children's Fund", 'English', 'Bhutan', 'United Nations Statistical Division', 'English', 'Bhutan', 'World Bank', '[]')

```

n', 'English', 'Bhutan', 'World Bank', '[7, 18]')
(36, 'BWA', '72', 1, 1, 1, 'BWA', '["DEVPING", "SSA", "LLDC", "SSA", "AFRICA", "GLOBAL", "SOUTHAFR", "ESARO", "SSA", "AFRO", "ESA"]', 'English', 'Botswana', 'United Nations Children's Fund', 'French', 'Botswana', 'United Nations Children's Fund', 'Spanish', 'Botswana', 'United Nations Children's Fund', 'English', 'Botswana', 'United Nations Statistical Division', 'English', 'Botswana', 'World Bank', '[]')
(37, 'CAF', '140', 1, 1, 1, 'CAF', '["MIDDLEAFR", "DEVPING", "SSA", "LDC", "LLDC", "SSA", "AFRICA", "GLOBAL", "WCARO", "SSA", "AFRO"]', 'English', 'Central African Republic', 'United Nations Children's Fund', 'French', 'République centrafricaine', 'United Nations Children's Fund', 'Spanish', 'República Centroafricana', 'United Nations Children's Fund', 'English', 'Central African Republic', 'United Nations Statistical Division', 'English', 'Central African Republic', 'World Bank', '[7, 18]')
(38, 'CAN', '124', 1, 1, 0, 1, 'CAN', '["DEV", "NORTHAMEUR", "AMERICAS", "GLOBAL", "NORTHAME", "IND", "NA"]', 'English', 'Canada', 'United Nations Children's Fund', 'French', 'Canada', 'United Nations Children's Fund', 'Spanish', 'Canada', 'United Nations Children's Fund', 'English', 'Canada', 'United Nations Statistical Division', 'English', 'Canada', 'World Bank', '[]')
(39, 'CCK', '166', 1, 0, 0, 0, 'CCK', '["DEV", "AUSNZ", "GLOBAL", "OCEANIA"]', 'English', 'Cocos (Keeling) Islands', 'United Nations Children's Fund', 'English', 'Cocos (Keeling) Islands', 'United Nations Statistical Division', None, None, None, None, None, None, None, None, '[]')
(40, 'CHE', '756', 1, 1, 0, 1, 'CHE', '["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "WESTEUR", "IND", "ECA", "EURO"]', 'English', 'Switzerland', 'United Nations Children's Fund', 'French', 'Suisse', 'United Nations Children's Fund', 'Spanish', 'Suiza', 'United Nations Children's Fund', 'English', 'Switzerland', 'United Nations Statistical Division', 'English', 'Switzerland', 'World Bank', '[]')
(41, 'CHI', '830', 1, 0, 0, 1, 'CHI', '["GLOBAL"]', 'English', 'Channel Islands', 'United Nations Children's Fund', 'English', 'Channel Islands', 'United Nations Statistical Division', None, None, None, None, None, None, None, None, None, '[]')
(42, 'CHL', '152', 1, 1, 0, 1, 'CHL', '["SOUTHAME", "DEV", "DEVPING", "NORTHAMEUR", "AMERICAS", "GLOBAL", "LATINAME", "LACRO", "LAC"]', 'English', 'Chile', 'United Nations Children's Fund', 'French', 'Chili', 'United Nations Children's Fund', 'Spanish', 'Chile', 'United Nations Children's Fund', 'English', 'Chile', 'United Nations Statistical Division', 'English', 'Chile', 'World Bank', '[]')
(43, 'CHN', '156', 1, 1, 1, 1, 'CHN', '["DEVPING", "EASTASIA", "EASOUTASI", "ASIA", "EASTASIA", "GLOBAL", "EAPRO", "EAP", "WPRO"]', 'English', 'China', 'United Nations Children's Fund', 'French', 'Chine', 'United Nations Children's Fund', 'Spanish', 'China', 'United Nations Children's Fund', 'English', 'China', 'United Nations Statistical Division', 'English', 'China', 'World Bank', '[]')
(44, 'CIV', '384', 1, 1, 1, 1, 'CIV', '["WESTAFR", "DEVPING", "SSA", "SSA", "AFRICA", "GLOBAL", "WCARO", "SSA", "AFRO"]', 'English', 'Côte d'Ivoire', 'United Nations Children's Fund', 'French', 'Côte d'Ivoire', 'United Nations Children's Fund', 'Spanish', 'Côte d'Ivoire', 'United Nations Children's Fund', 'English', 'Côte d'Ivoire', 'United Nations Statistical Division', 'English', 'Côte d'Ivoire', 'World Bank', '[2, 7, 18]')
(45, 'CMR', '120', 1, 1, 1, 1, 'CMR', '["MIDDLEAFR", "DEVPING", "SSA", "SSA", "AFRICA", "GLOBAL", "WCARO", "SSA", "AFRO"]', 'English', 'Cameroon', 'United Nations Children's Fund', 'French', 'Cameroun', 'United Nations Children's Fund', 'Spanish', 'Camerún', 'United Nations Children's Fund', 'English', 'Cameroon', 'United Nations Statistical Division', 'English', 'Cameroon', 'World Bank', '[2, 7, 18]')
(46, 'COD', '180', 1, 1, 1, 1, 'COD', '["MIDDLEAFR", "DEVPING", "SSA", "LDC", "SSA", "AFRICA", "GLOBAL", "WCARO", "SSA", "AFRO"]', 'English', 'Democratic Republic of the Congo', 'United Nations Children's Fund', 'French', 'République démocratique du Congo', 'United Nations Children's Fund', 'Spanish', 'República Democrática del Congo', 'United Nations Children's Fund', 'English', 'Democratic Republic of the Congo', 'United Nations Statistical Division', 'English', 'Congo Dem. Rep.', 'World Bank', '[2, 7, 18]')

```

(47, 'COG', '178', 1, 1, 1, 1, 'COG', '["MIDDLEAFR", "DEVPING", "SSA", "SSA", "AFRICA", "GLOBAL", "WCARO", "SSA", "AFRO"]', 'English', 'Congo', "United Nations Children's Fund", 'French', 'Congo', "United Nations Children's Fund", 'Spanish', 'Congo', "United Nations Children's Fund", 'English', 'Congo', 'United Nations Statistical Division', 'English', 'Congo Rep.', 'World Bank', '[7]')

(48, 'COK', '184', 1, 1, 0, 1, 'COK', '["OCEAN", "DEVPING", "SIDS", "GLOBAL", "OCEANIA", "POLYNES", "EAPRO", "EAP", "WPRO"]', 'English', 'Cook Islands', "United Nations Children's Fund", 'French', 'Îles Cook', "United Nations Children's Fund", 'Spanish', 'Islas Cook', "United Nations Children's Fund", 'English', 'Cook Islands', 'United Nations Statistical Division', None, None, None, '[]')

(49, 'COL', '170', 1, 1, 0, 1, 'COL', '["SOUTHAME", "DEVPING", "AMERICAS", "GLOBAL", "LATINAME", "LACRO", "LAC"]', 'English', 'Colombia', "United Nations Children's Fund", 'French', 'Colombie', "United Nations Children's Fund", 'Spanish', 'Colombia', "United Nations Children's Fund", 'English', 'Colombia', 'United Nations Statistical Division', 'English', 'Colombia', 'World Bank', '[]')

(50, 'COM', '174', 1, 1, 1, 1, 'COM', '["DEVPING", "SSA", "LDC", "SIDS", "SSA", "AFRICA", "EASTAFR", "GLOBAL", "ESARO", "SSA", "AFRO", "ESA"]', 'English', 'Comoros', "United Nations Children's Fund", 'French', 'Comores', "United Nations Children's Fund", 'Spanish', 'Comoras', "United Nations Children's Fund", 'English', 'Comoros', 'United Nations Statistical Division', 'English', 'Comoros', 'World Bank', '[7]')

(51, 'CPV', '132', 1, 1, 0, 1, 'CPV', '["WESTAFR", "DEVPING", "SSA", "SIDS", "SSA", "AFRICA", "GLOBAL", "WCARO", "SSA", "AFRO"]', 'English', 'Cabo Verde', "United Nations Children's Fund", 'French', 'Cabo Verde', "United Nations Children's Fund", 'Spanish', 'Cabo Verde', "United Nations Children's Fund", 'English', 'Cabo Verde', 'United Nations Statistical Division', 'English', 'Cabo Verde', 'World Bank', '[]')

(52, 'CRI', '188', 1, 1, 0, 1, 'CRI', '["DEVPING", "AMERICAS", "CENTRALAME", "GLOBAL", "LATINAME", "LACRO", "LAC"]', 'English', 'Costa Rica', "United Nations Children's Fund", 'French', 'Costa Rica', "United Nations Children's Fund", 'Spanish', 'Costa Rica', "United Nations Children's Fund", 'English', 'Costa Rica', 'United Nations Statistical Division', 'English', 'Costa Rica', 'World Bank', '[]')

(53, 'CUB', '192', 1, 1, 0, 1, 'CUB', '["DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME", "LACRO", "LAC"]', 'English', 'Cuba', "United Nations Children's Fund", 'French', 'Cuba', "United Nations Children's Fund", 'Spanish', 'Cuba', "United Nations Children's Fund", 'English', 'Cuba', 'United Nations Statistical Division', 'English', 'Cuba', 'World Bank', '[]')

(54, 'CUW', '531', 1, 0, 0, 1, 'CUW', '["DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME"]', 'English', 'Curaçao', "United Nations Children's Fund", 'English', 'Curaçao', 'United Nations Statistical Division', None, None, None, None, None, None, None, None, '[]')

(55, 'CXR', '162', 1, 0, 0, 0, 'CXR', '["DEV", "AUSNZ", "GLOBAL", "OCEANIA"]', 'English', 'Christmas Island', "United Nations Children's Fund", 'English', 'Christmas Island', 'United Nations Statistical Division', None, None, None, None, None, None, None, '[]')

(56, 'CYM', '136', 1, 0, 0, 1, 'CYM', '["DEVPING", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME"]', 'English', 'Cayman Islands', "United Nations Children's Fund", 'English', 'Cayman Islands', 'United Nations Statistical Division', None, None, None, None, None, None, None, '[]')

(57, 'CYP', '196', 1, 1, 0, 1, 'CYP', '["DEV", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "IND", "ECA", "EURO"]', 'English', 'Cyprus', "United Nations Children's Fund", 'French', 'Chypre', "United Nations Children's Fund", 'Spanish', 'Chipre', "United Nations Children's Fund", 'English', 'Cyprus', 'United Nations Statistical Division', 'English', 'Cyprus', 'World Bank', '[]')

(58, 'CZE', '203', 1, 1, 0, 1, 'CZE', '["DEV", "NORTHAMEUR", "EASTEUR", "EUROPE", "GLOBAL", "IND", "ECA", "EURO"]', 'English', 'Czechia', "United Nations Children's Fund", 'French', 'République tchèque', "United Nations Children's Fund", 'Spanish', 'República Checa', "United Nations Children's Fund", 'English', 'Czech Republic', 'United Nations Statistical Division', 'English', 'Czech Republic', 'World Bank', '[]')

(59, 'DEU', '276', 1, 1, 0, 1, 'DEU', '["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "WESTEUR", "IND", "ECA", "EURO"]', 'English', 'Germany', 'United Nations Children's Fund', 'French', 'Allemagne', 'United Nations Children's Fund', 'Spanish', 'Alemania', 'United Nations Children's Fund', 'English', 'Germany', 'United Nations Statistical Division', 'English', 'Germany', 'World Bank', '[]')

(60, 'DJI', '262', 1, 1, 1, 1, 'DJI', '["DEVPING", "SSA", "LDC", "SSA", "AFRICA", "EASTAFR", "GLOBAL", "MENARO", "SSA", "EMRO"]', 'English', 'Djibouti', 'United Nations Children's Fund', 'French', 'Djibouti', 'United Nations Children's Fund', 'Spanish', 'Djibouti', 'United Nations Children's Fund', 'English', 'Djibouti', 'United Nations Statistical Division', 'English', 'Djibouti', 'World Bank', '[2, 7, 18]')

(61, 'DMA', '212', 1, 1, 0, 1, 'DMA', '["DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME", "LACRO", "LAC"], 'English', 'Dominica', 'United Nations Children's Fund', 'French', 'Dominique', 'United Nations Children's Fund', 'Spanish', 'Dominica', 'United Nations Children's Fund', 'English', 'Dominica', 'United Nations Statistical Division', 'English', 'Dominica', 'World Bank', '[]')

(62, 'DNK', '208', 1, 1, 0, 1, 'DNK', '["NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "IND", "ECA", "EURO"]', 'English', 'Denmark', 'United Nations Children's Fund', 'French', 'Danemark', 'United Nations Children's Fund', 'Spanish', 'Dinamarca', 'United Nations Children's Fund', 'English', 'Denmark', 'United Nations Statistical Division', 'English', 'Denmark', 'World Bank', '[]')

(63, 'DOM', '214', 1, 1, 0, 1, 'DOM', '["DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME", "LACRO", "LAC"], 'English', 'Dominican Republic', 'United Nations Children's Fund', 'French', 'République dominicaine', 'United Nations Children's Fund', 'Spanish', 'República Dominicana', 'United Nations Children's Fund', 'English', 'Dominican Republic', 'United Nations Statistical Division', 'English', 'Dominican Republic', 'World Bank', '[]')

(64, 'DZA', '12', 1, 1, 0, 1, 'DZA', '["DEVPING", "NORTHAFR", "WASINAFR", "AFRICA", "GLOBAL", "NORTHAFR", "MENA", "MENARO", "AFRO"]', 'English', 'Algeria', 'United Nations Children's Fund', 'French', 'Algérie', 'United Nations Children's Fund', 'Spanish', 'Argelia', 'United Nations Children's Fund', 'English', 'Algeria', 'United Nations Statistical Division', 'English', 'Algeria', 'World Bank', '[]')

(65, 'ECU', '218', 1, 1, 0, 1, 'ECU', '["SOUTHAME", "DEVPING", "AMERICAS", "GLOBAL", "LATINAME", "LACRO", "LAC"]', 'English', 'Ecuador', 'United Nations Children's Fund', 'French', 'Équateur', 'United Nations Children's Fund', 'Spanish', 'Ecuador', 'United Nations Children's Fund', 'English', 'Ecuador', 'United Nations Statistical Division', 'English', 'Ecuador', 'World Bank', '[]')

(66, 'EGY', '818', 1, 1, 1, 1, 'EGY', '["DEVPING", "NORTHAFR", "WASINAFR", "AFRICA", "GLOBAL", "NORTHAFR", "MENA", "MENARO", "EMRO"]', 'English', 'Egypt', 'United Nations Children's Fund', 'French', 'Égypte', 'United Nations Children's Fund', 'Spanish', 'Egipto', 'United Nations Children's Fund', 'English', 'Egypt', 'United Nations Statistical Division', 'English', 'Egypt Arab Rep.', 'World Bank', '[]')

(67, 'ERI', '232', 1, 1, 1, 1, 'ERI', '["DEVPING", "SSA", "LDC", "SSA", "AFRICA", "EASTAFR", "GLOBAL", "ESARO", "AFRO", "ESA"]', 'English', 'Eritrea', 'United Nations Children's Fund', 'French', 'Érythrée', 'United Nations Children's Fund', 'Spanish', 'Eritrea', 'United Nations Children's Fund', 'English', 'Eritrea', 'United Nations Statistical Division', 'English', 'Eritrea', 'World Bank', '[7]')

(68, 'ESH', '732', 1, 0, 0, 0, 'ESH', '["DEVPING", "NORTHAFR", "WASINAFR", "AFRICA", "GLOBAL", "NORTHAFR"]', 'English', 'Western Sahara', 'United Nations Children's Fund', 'English', 'Western Sahara', 'United Nations Statistical Division', None, None, None, None, None, None, None, '[]')

(69, 'ESP', '724', 1, 1, 0, 1, 'ESP', '["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEUR", "IND", "ECA", "EURO"]', 'English', 'Spain', 'United Nations Children's Fund', 'French', 'Espagne', 'United Nations Children's Fund', 'Spanish', 'España', 'United Nations Children's Fund', 'English', 'Spain', 'United Nations Statistical Division', 'English', 'Spain', 'World Bank', '[]')

(70, 'EST', '233', 1, 1, 0, 1, 'EST', '["NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "IND", "ECA", "EURO"]', 'English', 'Estonia', 'United Nations Children's Fund', 'French', 'Estonie', 'United Nations Children's Fund', 'Spanish',

'Estonia', "United Nations Children's Fund", 'English', 'Estonia', 'United Nations Statistical Division', 'English', 'Estonia', 'World Bank', '[]')

(71, 'ETH', '231', 1, 1, 1, 1, 'ETH', '[["DEVPING", "SSA", "LDC", "LLDC", "SSA", "AFRICA", "EASTAFR", "GLOBAL", "ESARO", "SSA", "AFRO", "ESA"]]', 'English', 'Ethiopia', "United Nations Children's Fund", 'French', 'Éthiopie', "United Nations Children's Fund", 'Spanish', 'Etiopía', "United Nations Children's Fund", 'English', 'Ethiopia', 'United Nations Statistical Division', 'English', 'Ethiopia', 'World Bank', '[2, 7, 18]')

(72, 'FIN', '246', 1, 1, 0, 1, 'FIN', '[["NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "IND", "ECA", "EURO"]]', 'English', 'Finland', "United Nations Children's Fund", 'French', 'Finlande', "United Nations Children's Fund", 'Spanish', 'Finlandia', "United Nations Children's Fund", 'English', 'Finland', 'United Nations Statistical Division', 'English', 'Finland', 'Finland', 'World Bank', '[]')

(73, 'FJI', '242', 1, 1, 0, 1, 'FJI', '[["OCEAN", "DEVPING", "SIDS", "GLOBAL", "MELANESIA", "OCEANIA", "EAPRO", "EAP", "WPRO"]]', 'English', 'Fiji', "United Nations Children's Fund", 'French', 'Fidji', "United Nations Children's Fund", 'Spanish', 'Fiji', "United Nations Children's Fund", 'English', 'Fiji', 'United Nations Statistical Division', 'English', 'Fiji', 'World Bank', '[7]')

(74, 'FLK', '238', 1, 0, 0, 1, 'FLK', '[["SOUTHAME", "DEVPING", "AMERICAS", "GLOBAL", "LATINAME"]]', 'English', 'Falkland Islands (Malvinas)', "United Nations Children's Fund", 'English', 'Falkland Islands (Malvinas)', 'United Nations Statistical Division', None, '[]')

(75, 'FRA', '250', 1, 1, 0, 1, 'FRA', '[["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "WESTEUR", "IND", "ECA", "EURO"]]', 'English', 'France', "United Nations Children's Fund", 'French', 'France', "United Nations Children's Fund", 'Spanish', 'Francia', "United Nations Children's Fund", 'English', 'France', 'United Nations Statistical Division', 'English', 'France', 'World Bank', '[]')

(76, 'FRO', '234', 1, 0, 0, 1, 'FRO', '[["NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "GLOBAL"]]', 'English', 'Faroe Islands', "United Nations Children's Fund", 'English', 'Faroe Islands', 'United Nations Statistical Division', None, '[]')

(77, 'FSM', '583', 1, 1, 0, 1, 'FSM', '[["OCEAN", "DEVPING", "SIDS", "GLOBAL", "MICRONESIA", "OCEANIA", "EAPRO", "EAP", "WPRO"]]', 'English', 'Micronesia (Federated States of)', "United Nations Children's Fund", 'French', 'Micronésie (États fédérés de)', "United Nations Children's Fund", 'Spanish', 'Micronesia (Estados Federados de)', "United Nations Children's Fund", 'English', 'Micronesia (Fed. States of)', 'United Nations Statistical Division', 'English', 'Micronesia Fed. Sts.', 'World Bank', '[]')

(78, 'GAB', '266', 1, 1, 1, 1, 'GAB', '[["MIDDLEAFR", "DEVPING", "SSA", "SSA", "AFRICA", "GLOBAL", "WCARO", "SSA", "AFRO"]]', 'English', 'Gabon', "United Nations Children's Fund", 'French', 'Gabon', "United Nations Children's Fund", 'Spanish', 'Gabón', "United Nations Children's Fund", 'English', 'Gabon', 'United Nations Statistical Division', 'English', 'Gabon', 'World Bank', '[]')

(79, 'GBR', '826', 1, 1, 0, 1, 'GBR', '[["NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "IND", "ECA", "EURO"]]', 'English', 'United Kingdom', "United Nations Children's Fund", 'French', 'Royaume-Uni', "United Nations Children's Fund", 'Spanish', 'Reino Unido', "United Nations Children's Fund", 'English', 'United Kingdom of Great Britain and Northern Ireland', "United Nations Statistical Division", 'English', 'United Kingdom', 'World Bank', '[]')

(80, 'GEO', '268', 1, 1, 0, 1, 'GEO', '[["CAUCCEN", "DEVPING", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "CEECIS", "ECA", "ECARO", "EURO"]]', 'English', 'Georgia', "United Nations Children's Fund", 'French', 'Géorgie', "United Nations Children's Fund", 'Spanish', 'Georgia', "United Nations Children's Fund", 'English', 'Georgia', 'United Nations Statistical Division', 'English', 'Georgia', 'World Bank', '[]')

(81, 'GGY', '831', 1, 0, 0, 0, 'GGY', '[["NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "GLOBAL"]]', 'English', 'Guernsey', "United Nations Children's Fund", 'English', 'Guernsey', 'United Nations Statistical Division', None, None, None, None, None, None, None, None, None, '[]')

(82, 'GHA', '288', 1, 1, 1, 1, 'GHA', '[["WESTAFR", "DEVPING", "SSA", "SSA", "AFRICA", "GLOBAL", "WCARO", "AFRO"]]', 'Eng

lish', 'Ghana', "United Nations Children's Fund", 'French', 'Ghana', "United Nations Children's Fund", 'Spanish', 'Ghana', "United Nations Children's Fund", 'English', 'Ghana', 'United Nations Statistical Division', 'English', 'Ghana', 'World Bank', '[2, 7, 18]')

(83, 'GIB', '292', 1, 0, 0, 1, 'GIB', '["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEUR"]', 'English', 'Gibraltar', "United Nations Children's Fund", 'English', 'Gibraltar', 'United Nations Statistical Division', None, '[]')

(84, 'GIN', '324', 1, 1, 1, 1, 'GIN', '["WESTAFR", "DEVPING", "SSA", "LDC", "SSA", "AFRICA", "GLOBAL", "WCARO", "SSA", "AFRO"]', 'English', 'Guinea', "United Nations Children's Fund", 'French', 'Guinée', "United Nations Children's Fund", 'Spanish', 'Guinea', "United Nations Children's Fund", 'English', 'Guinea', 'United Nations Statistical Division', 'English', 'Guinea', 'World Bank', '[7, 18]')

(85, 'GLP', '312', 1, 0, 0, 1, 'GLP', '["DEVPING", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME"]', 'English', 'Guadeloupe', "United Nations Children's Fund", 'English', 'Guadeloupe', 'United Nations Statistical Division', None, None, None, None, None, None, None, None, None, '[]')

(86, 'GMB', '270', 1, 1, 1, 1, 'GMB', '["WESTAFR", "DEVPING", "SSA", "LDC", "SSA", "AFRICA", "GLOBAL", "WCARO", "AFR0"]', 'English', 'Gambia', "United Nations Children's Fund", 'French', 'Gambie', "United Nations Children's Fund", 'Spanish', 'Gambia', "United Nations Children's Fund", 'English', 'Gambia', 'United Nations Statistical Division', 'English', 'Gambia The', 'World Bank', '[7, 18]')

(87, 'GNB', '624', 1, 1, 1, 1, 'GNB', '["WESTAFR", "DEVPING", "SSA", "LDC", "SIDS", "SSA", "AFRICA", "GLOBAL", "WCARO", "AFRO"]', 'English', 'Guinea-Bissau', "United Nations Children's Fund", 'French', 'Guinée-Bissau', "United Nations Children's Fund", 'Spanish', 'Guinea-Bissau', "United Nations Children's Fund", 'English', 'Guinea-Bissau', 'United Nations Statistical Division', 'English', 'Guinea-Bissau', 'World Bank', '[7, 18]')

(88, 'GNQ', '226', 1, 1, 1, 1, 'GNQ', '["MIDDLEAFR", "DEVPING", "SSA", "SSA", "AFRICA", "GLOBAL", "WCARO", "AFRO"]', 'English', 'Equatorial Guinea', "United Nations Children's Fund", 'French', 'Guinée équatoriale', "United Nations Children's Fund", 'Spanish', 'Guinea Ecuatorial', "United Nations Children's Fund", 'English', 'Equatorial Guinea', 'United Nations Statistical Division', 'English', 'Equatorial Guinea', 'World Bank', '[]')

(89, 'GRC', '300', 1, 1, 0, 1, 'GRC', '["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEUR", "IND", "ECA", "EURO"]', 'English', 'Greece', "United Nations Children's Fund", 'French', 'Grèce', "United Nations Children's Fund", 'Spanish', 'Grecia', "United Nations Children's Fund", 'English', 'Greece', 'United Nations Statistical Division', 'English', 'Greece', 'World Bank', '[]')

(90, 'GRD', '308', 1, 1, 0, 1, 'GRD', '["DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME", "LACRO", "LAC"]', 'English', 'Grenada', "United Nations Children's Fund", 'French', 'Grenade', "United Nations Children's Fund", 'Spanish', 'Granada', "United Nations Children's Fund", 'English', 'Grenada', 'United Nations Statistical Division', 'English', 'Grenada', 'World Bank', '[]')

(91, 'GRL', '304', 1, 0, 0, 1, 'GRL', '["DEV", "NORTHAMEUR", "AMERICAS", "GLOBAL", "NORTHAME"]', 'English', 'Greenland', "United Nations Children's Fund", 'English', 'Greenland', 'United Nations Statistical Division', None, None, None, None, None, None, None, None, None, '[]')

(92, 'GTM', '320', 1, 1, 1, 1, 'GTM', '["DEVPING", "AMERICAS", "CENTRALAME", "GLOBAL", "LATINAME", "LACRO", "LAC"]', 'English', 'Guatemala', "United Nations Children's Fund", 'French', 'Guatemala', "United Nations Children's Fund", 'Spanish', 'Guatemala', "United Nations Children's Fund", 'English', 'Guatemala', 'United Nations Statistical Division', 'English', 'Guatemala', 'World Bank', '[]')

(93, 'GUF', '254', 1, 0, 0, 1, 'GUF', '["SOUTHAME", "DEVPING", "AMERICAS", "GLOBAL", "LATINAME"]', 'English', 'French Guiana', "United Nations Children's Fund", 'English', 'French Guiana', 'United Nations Statistical Division', None, None, None, None, None, None, None, None, None, '[]')

(94, 'GUM', '316', 1, 0, 0, 1, 'GUM', '["OCEAN", "DEVPING", "SIDS", "GLOBAL", "MICRONESIA", "OCEANIA"]', 'English', 'Guam', "United Nations Children's Fund", 'English', 'Guam', 'United Nations Statistical Division', None, None, None, Non

```

e, None, None, None, None, None, '[]')
(95, 'GUY', '328', 1, 1, 0, 1, 'GUY', '["SOUTHAME", "DEVPING", "SIDS", "AMERICAS", "GLOBAL", "LATINAME", "LACRO", "LA C"]', 'English', 'Guyana', 'United Nations Children's Fund', 'French', 'Guyana', 'United Nations Children's Fund', 'Spanish', 'Guyana', 'United Nations Children's Fund', 'English', 'Guyana', 'United Nations Statistical Division', 'English', 'Guyana', 'World Bank', '[]')
(96, 'HKG', '344', 1, 0, 0, 1, 'HKG', '["DEVPING", "EASTASIA", "EASOUTASI", "ASIA", "EASTASIA", "GLOBAL"]', 'English', 'China Hong Kong Special Administrative Region', 'United Nations Children's Fund', 'French', 'Chine région administrative spéciale de Hong Kong', 'United Nations Children's Fund', 'Spanish', 'China región administrativa especial de Hong Kong', 'United Nations Children's Fund', 'English', 'China Hong Kong Special Administrative Region', 'United Nations Statistical Division', None, None, None, '[]')
(97, 'HMD', '334', 1, 0, 0, 0, 'HMD', '["DEV", "AUSNZ", "GLOBAL", "OCEANIA"]', 'English', 'Heard Island and McDonald Islands', 'United Nations Children's Fund', 'English', 'Heard Island and McDonald Islands', 'United Nations Statistical Division', None, None, None, None, None, None, None, '[]')
(98, 'HND', '340', 1, 1, 0, 1, 'HND', '["DEVPING", "AMERICAS", "CENTRALAME", "GLOBAL", "LATINAME", "LACRO", "LAC"]', 'English', 'Honduras', 'United Nations Children's Fund', 'French', 'Honduras', 'United Nations Children's Fund', 'Spanish', 'Honduras', 'United Nations Children's Fund', 'English', 'Honduras', 'United Nations Statistical Division', 'English', 'Honduras', 'World Bank', '[]')
(99, 'HRV', '191', 1, 1, 0, 1, 'HRV', '["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEUR", "CEECIS", "ECA", "ECARO", "EURO"]', 'English', 'Croatia', 'United Nations Children's Fund', 'French', 'Croatie', 'United Nations Children's Fund', 'Spanish', 'Croacia', 'United Nations Children's Fund', 'English', 'Croatia', 'United Nations Statistical Division', 'English', 'Croatia', 'World Bank', '[]')
(100, 'HTI', '332', 1, 1, 1, 1, 'HTI', '["DEVPING", "LDC", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME", "LACRO", "LAC"]', 'English', 'Haiti', 'United Nations Children's Fund', 'French', 'Haïti', 'United Nations Children's Fund', 'Spanish', 'Haiti', 'United Nations Children's Fund', 'English', 'Haiti', 'United Nations Statistical Division', 'English', 'Haiti', 'World Bank', '[2, 7]')
(101, 'HUN', '348', 1, 1, 0, 1, 'HUN', '["DEV", "NORTHAMEUR", "EASTEUR", "EUROPE", "GLOBAL", "IND", "ECA", "EURO"]', 'English', 'Hungary', 'United Nations Children's Fund', 'French', 'Hongrie', 'United Nations Children's Fund', 'Spanish', 'Hungria', 'United Nations Children's Fund', 'English', 'Hungary', 'United Nations Statistical Division', 'English', 'Hungary', 'World Bank', '[]')
(102, 'IDN', '360', 1, 1, 1, 1, 'IDN', '["DEVPING", "EASOUTASI", "ASIA", "GLOBAL", "SOUTHEASTASIA", "EAPRO", "EAP", "SE ARO"]', 'English', 'Indonesia', 'United Nations Children's Fund', 'French', 'Indonésie', 'United Nations Children's Fund', 'Spanish', 'Indonesia', 'United Nations Children's Fund', 'English', 'Indonesia', 'United Nations Statistical Division', 'English', 'Indonesia', 'World Bank', '[2, 7, 18]')
(103, 'IMN', '833', 1, 0, 0, 1, 'IMN', '["NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "GLOBAL"]', 'English', 'Isle of Man', 'United Nations Children's Fund', 'English', 'Isle of Man', 'United Nations Statistical Division', None, None, None, None, None, None, '[]')
(104, 'IND', '356', 1, 1, 1, 1, 'IND', '["DEVPING", "SOUTHCENASIA", "ASIA", "GLOBAL", "SOUTHASIA", "ROSA", "SA", "SEAR O"]', 'English', 'India', 'United Nations Children's Fund', 'French', 'Inde', 'United Nations Children's Fund', 'Spanish', 'India', 'United Nations Children's Fund', 'English', 'India', 'United Nations Statistical Division', 'English', 'India', 'World Bank', '[2, 7, 18]')
(105, 'IOT', '86', 1, 0, 0, 0, 'IOT', '["DEVPING", "SSA", "SSA", "AFRICA", "EASTAFR", "GLOBAL"]', 'English', 'British Indian Ocean Territory', 'United Nations Children's Fund', 'English', 'British Indian Ocean Territory', 'United Nations Statistical Division', None, None, None, None, None, None, None, '[]')
(106, 'IRL', '372', 1, 1, 0, 1, 'IRL', '["NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "IND", "ECA", "EURO"]', 'English', 'Ireland', 'United Nations Children's Fund', 'French', 'Irlande', 'United Nations Children's Fund', 'Spanish'

```

```

h', 'Irlanda', "United Nations Children's Fund", 'English', 'Ireland', 'United Nations Statistical Division', 'English',
h', 'Ireland', 'World Bank', '[]')
(107, 'IRN', '364', 1, 1, 0, 1, 'IRN', '["DEVPING", "SOUTHCEASIA", "ASIA", "GLOBAL", "SOUTHASIA", "MENA", "MENARO", "EMRO"]', 'English', 'Iran (Islamic Republic of)', "United Nations Children's Fund", 'French', 'Iran (République islamique d')', "United Nations Children's Fund", 'Spanish', 'Irán (República Islámica de)', "United Nations Children's Fund", 'English', 'Iran (Islamic Republic of)', 'United Nations Statistical Division', 'English', 'Iran Islamic Rep.', 'World Bank', '[]')
(108, 'IRQ', '368', 1, 1, 1, 1, 'IRQ', '["DEVPING", "WESTASI", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "MENA", "MENARO", "EMRO"]', 'English', 'Iraq', "United Nations Children's Fund", 'French', 'Iraq', "United Nations Children's Fund", 'Spanish', 'Iraq', "United Nations Children's Fund", 'English', 'Iraq', 'United Nations Statistical Division', 'English', 'Iraq', 'World Bank', '[7, 18]')
(109, 'ISL', '352', 1, 1, 0, 1, 'ISL', '["NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "IND", "ECA", "EURO"]', 'English', 'Iceland', "United Nations Children's Fund", 'French', 'Islande', "United Nations Children's Fund", 'Spanish', 'Islandia', "United Nations Children's Fund", 'English', 'Iceland', 'United Nations Statistical Division', 'English', 'Iceland', 'World Bank', '[]')
(110, 'ISR', '376', 1, 1, 0, 1, 'ISR', '["DEV", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "IND", "MENA", "EURO"]', 'English', 'Israel', "United Nations Children's Fund", 'French', 'Israël', "United Nations Children's Fund", 'Spanish', 'Israel', "United Nations Children's Fund", 'English', 'Israel', 'United Nations Statistical Division', 'English', 'Israel', 'World Bank', '[]')
(111, 'ITA', '380', 1, 1, 0, 1, 'ITA', '["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEUR", "IND", "ECA", "EURO"]', 'English', 'Italy', "United Nations Children's Fund", 'French', 'Italie', "United Nations Children's Fund", 'Spanish', 'Italia', "United Nations Children's Fund", 'English', 'Italy', 'United Nations Statistical Division', 'English', 'Italy', 'World Bank', '[]')
(112, 'JAM', '388', 1, 1, 0, 1, 'JAM', '["DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME", "LACRO", "LAC"], 'English', 'Jamaica', "United Nations Children's Fund", 'French', 'Jamaïque', "United Nations Children's Fund", 'Spanish', 'Jamaica', "United Nations Children's Fund", 'English', 'Jamaica', 'United Nations Statistical Division', 'English', 'Jamaica', 'World Bank', '[]')
(113, 'JEY', '832', 1, 0, 0, 0, 'JEY', '["NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "GLOBAL"]', 'English', 'Jersey', "United Nations Children's Fund", 'English', 'Jersey', 'United Nations Statistical Division', None, None, None, None, None, None, None, None, None, '[]')
(114, 'JOR', '400', 1, 1, 0, 1, 'JOR', '["DEVPING", "WESTASI", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "MENA", "MENARO", "EMRO"]', 'English', 'Jordan', "United Nations Children's Fund", 'French', 'Jordanie', "United Nations Children's Fund", 'Spanish', 'Jordania', "United Nations Children's Fund", 'English', 'Jordan', 'United Nations Statistical Division', 'English', 'Jordan', 'World Bank', '[]')
(115, 'JPN', '392', 1, 1, 0, 1, 'JPN', '["DEV", "EASOUTASI", "ASIA", "EASTASIA", "GLOBAL", "IND", "EAP", "WPRO"]', 'English', 'Japan', "United Nations Children's Fund", 'French', 'Japon', "United Nations Children's Fund", 'Spanish', 'Japón', "United Nations Children's Fund", 'English', 'Japan', 'United Nations Statistical Division', 'English', 'Japan', 'World Bank', '[]')
(116, 'KAZ', '398', 1, 1, 0, 1, 'KAZ', '["CAUCCEN", "DEVPING", "LLDC", "SOUTHCEASIA", "ASIA", "CENTRALASIA", "GLOBAL", "CEECIS", "ECA", "ECARO", "EURO"]', 'English', 'Kazakhstan', "United Nations Children's Fund", 'French', 'Kazakhstan', "United Nations Children's Fund", 'Spanish', 'Kazajstán', "United Nations Children's Fund", 'English', 'Kazakhstan', 'United Nations Statistical Division', 'English', 'Kazakhstan', 'World Bank', '[]')
(117, 'KEN', '404', 1, 1, 1, 1, 'KEN', '["DEVPING", "SSA", "SSA", "AFRICA", "EASTAFR", "GLOBAL", "ESARO", "AFRO", "ESA"]', 'English', 'Kenya', "United Nations Children's Fund", 'French', 'Kenya', "United Nations Children's Fund", 'Spanish', 'Kenya', "United Nations Children's Fund", 'English', 'Kenya', 'United Nations Statistical Division', 'English',

```

'Kenya', 'World Bank', '[2, 7, 18]')

(118, 'KGZ', '417', 1, 1, 1, 1, 'KGZ', '[{"CAUCCEN", "DEVPING", "LLDC", "SOUTHCENASIA", "ASIA", "CENTRALASIA", "GLOBAL", "CEECIS", "ECA", "ECARO", "EURO"]}', 'English', 'Kyrgyzstan', "United Nations Children's Fund", 'French', 'Kirghizistan', "United Nations Children's Fund", 'Spanish', 'Kirguistán', "United Nations Children's Fund", 'English', 'Kyrgyzstan', 'United Nations Statistical Division', 'English', 'Kyrgyz Republic', 'World Bank', '[7]')

(119, 'KHM', '116', 1, 1, 1, 1, 'KHM', '[{"DEVPING", "EASOUTASI", "LDC", "ASIA", "GLOBAL", "SOUTHEASTASIA", "EAPRO", "EAP", "WPRO"]]', 'English', 'Cambodia', "United Nations Children's Fund", 'French', 'Cambodge', "United Nations Children's Fund", 'Spanish', 'Camboya', "United Nations Children's Fund", 'English', 'Cambodia', 'United Nations Statistical Division', 'English', 'Cambodia', 'World Bank', '[7]')

(120, 'KIR', '296', 1, 1, 0, 1, 'KIR', '[{"OCEAN", "DEVPING", "LDC", "SIDS", "GLOBAL", "MICRONESIA", "OCEANIA", "EAPRO", "EAP", "WPRO"]]', 'English', 'Kiribati', "United Nations Children's Fund", 'French', 'Kiribati', "United Nations Children's Fund", 'Spanish', 'Kiribati', "United Nations Children's Fund", 'English', 'Kiribati', 'United Nations Statistical Division', 'English', 'Kiribati', 'World Bank', '[]')

(121, 'KNA', '659', 1, 1, 0, 1, 'KNA', '[{"DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME", "LACRO", "LAC"], "English", 'Saint Kitts and Nevis', "United Nations Children's Fund", 'French', 'Saint-Kitts-et-Nevis', "United Nations Children's Fund", 'Spanish', 'Saint Kitts y Nevis', "United Nations Children's Fund", 'English', 'Saint Kitts and Nevis', 'United Nations Statistical Division', 'English', 'St. Kitts and Nevis', 'World Bank', '[]')

(122, 'KOR', '410', 1, 1, 0, 1, 'KOR', '[{"DEVPING", "EASTASIA", "EASOUTASI", "ASIA", "EASTASIA", "GLOBAL", "EAPRO", "EAP", "WPRO"]]', 'English', 'Republic of Korea', "United Nations Children's Fund", 'French', 'République de Corée', "United Nations Children's Fund", 'Spanish', 'República de Corea', "United Nations Children's Fund", 'English', 'Republic of Korea', 'United Nations Statistical Division', 'English', 'Korea Rep.', 'World Bank', '[]')

(123, 'KWT', '414', 1, 1, 0, 1, 'KWT', '[{"DEVPING", "WESTASI", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "MENA", "MENARO", "EMRO"]]', 'English', 'Kuwait', "United Nations Children's Fund", 'French', 'Koweït', "United Nations Children's Fund", 'Spanish', 'Kuwait', "United Nations Children's Fund", 'English', 'Kuwait', 'United Nations Statistical Division', 'English', 'Kuwait', 'World Bank', '[]')

(124, 'LAO', '418', 1, 1, 1, 1, 'LAO', '[{"DEVPING", "EASOUTASI", "LDC", "LLDC", "ASIA", "GLOBAL", "SOUTHEASTASIA", "EAPRO", "EAP", "WPRO"]]', 'English', 'Lao People's Democratic Republic', "United Nations Children's Fund", 'French', 'République démocratique populaire lao', "United Nations Children's Fund", 'Spanish', 'República Democrática Popular Lao', "United Nations Children's Fund", 'English', 'Lao People's Democratic Republic', 'United Nations Statistical Division', 'English', 'Lao PDR', 'World Bank', '[7]')

(125, 'LBN', '422', 1, 1, 0, 1, 'LBN', '[{"DEVPING", "WESTASI", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "MENA", "MENARO", "EMRO"]]', 'English', 'Lebanon', "United Nations Children's Fund", 'French', 'Liban', "United Nations Children's Fund", 'Spanish', 'Líbano', "United Nations Children's Fund", 'English', 'Lebanon', 'United Nations Statistical Division', 'English', 'Lebanon', 'World Bank', '[]')

(126, 'LBR', '430', 1, 1, 1, 1, 'LBR', '[{"WESTAFR", "DEVPING", "SSA", "LDC", "SSA", "AFRICA", "GLOBAL", "WCARO", "AFRO"]]', 'English', 'Liberia', "United Nations Children's Fund", 'French', 'Libéria', "United Nations Children's Fund", 'Spanish', 'Liberia', "United Nations Children's Fund", 'English', 'Liberia', 'United Nations Statistical Division', 'English', 'Liberia', 'World Bank', '[7, 18]')

(127, 'LBY', '434', 1, 1, 0, 1, 'LBY', '[{"DEVPING", "NORTHAFR", "WASINAFR", "AFRICA", "GLOBAL", "NORTHAFR", "MENA", "MENARO", "SSA", "EMRO"]]', 'English', 'Libya', "United Nations Children's Fund", 'French', 'Libye', "United Nations Children's Fund", 'Spanish', 'Libia', "United Nations Children's Fund", 'English', 'Libya', 'United Nations Statistical Division', 'English', 'Libya', 'World Bank', '[]')

(128, 'LCA', '662', 1, 1, 0, 1, 'LCA', '[{"DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME", "LACRO", "LAC"], "English", 'Saint Lucia', "United Nations Children's Fund", 'French', 'Sainte-Lucie', "United Nations Children's Fund", 'Spanish', 'Santa Lucía', "United Nations Children's Fund", 'English', 'Saint Lucia', 'United Nations Statistical

```
l Division', 'English', 'St. Lucia', 'World Bank', '[]')
(129, 'LIE', '438', 1, 1, 0, 1, 'LIE', '["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "WESTEUR", "IND", "ECA"]', 'English',
'Liechtenstein', "United Nations Children's Fund", 'French', 'Liechtenstein', "United Nations Children's Fund", 'Spanish',
'Liechtenstein', "United Nations Children's Fund", 'English', 'Liechtenstein', 'United Nations Statistical Division',
'English', 'Liechtenstein', 'World Bank', '[]')
(130, 'LKA', '144', 1, 1, 0, 1, 'LKA', '["DEVPING", "SOUTHCEASIA", "ASIA", "GLOBAL", "SOUTHASIA", "ROSA", "SA", "SEAR
O"]', 'English', 'Sri Lanka', "United Nations Children's Fund", 'French', 'Sri Lanka', "United Nations Children's Fun
d", 'Spanish', 'Sri Lanka', "United Nations Children's Fund", 'English', 'Sri Lanka', 'United Nations Statistical Divis
ion', 'English', 'Sri Lanka', 'World Bank', '[7, 18]')
(131, 'LSO', '426', 1, 1, 1, 'LSO', '["DEVPING", "SSA", "LDC", "LLDC", "SSA", "AFRICA", "GLOBAL", "SOUTHAFR", "ESAR
O", "SSA", "AFRO", "ESA"]', 'English', 'Lesotho', "United Nations Children's Fund", 'French', 'Lesotho', "United Nation
s Children's Fund", 'Spanish', 'Lesotho', "United Nations Children's Fund", 'English', 'Lesotho', 'United Nations Stati
stical Division', 'English', 'Lesotho', 'World Bank', '[2, 7, 18]')
(132, 'LTU', '440', 1, 1, 0, 1, 'LTU', '["NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "IND", "ECA", "EURO"]',
'English', 'Lithuania', "United Nations Children's Fund", 'French', 'Lituanie', "United Nations Children's Fund", 'Span
ish', 'Lituania', "United Nations Children's Fund", 'English', 'Lithuania', 'United Nations Statistical Division', 'Eng
lish', 'Lithuania', 'World Bank', '[]')
(133, 'LUX', '442', 1, 1, 0, 1, 'LUX', '["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "WESTEUR", "IND", "ECA", "EURO"]',
'English', 'Luxembourg', "United Nations Children's Fund", 'French', 'Luxembourg', "United Nations Children's Fund", 'Spa
nish', 'Luxemburgo', "United Nations Children's Fund", 'English', 'Luxembourg', 'United Nations Statistical Division',
'English', 'Luxembourg', 'World Bank', '[]')
(134, 'LVA', '428', 1, 1, 0, 1, 'LVA', '["NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "IND", "ECA", "EURO"]',
'English', 'Latvia', "United Nations Children's Fund", 'French', 'Lettonie', "United Nations Children's Fund", 'Spanis
h', 'Letonia', "United Nations Children's Fund", 'English', 'Latvia', 'United Nations Statistical Division', 'English',
'Latvia', 'World Bank', '[]')
(135, 'MAC', '446', 1, 0, 0, 1, 'MAC', '["DEVPING", "EASTASIA", "EASOUTASI", "ASIA", "EASTASIA", "GLOBAL"]', 'English',
'China Macao Special Administrative Region', "United Nations Children's Fund", 'English', 'China Macao Special Administ
rative Region', 'United Nations Statistical Division', None, None,
'[]')
(136, 'MAF', '663', 1, 0, 0, 1, 'MAF', '["DEVPING", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME"]', 'English', 'Saint Mar
tin (French part)', "United Nations Children's Fund", 'English', 'Saint Martin (French Part)', 'United Nations Statisti
cal Division', None, '[]')
(137, 'MAR', '504', 1, 1, 1, 1, 'MAR', '["DEVPING", "NORTHAFR", "WASINAFR", "AFRICA", "GLOBAL", "NORTHAFR", "MENA",
"ME
NARO", "EMRO"]', 'English', 'Morocco', "United Nations Children's Fund", 'French', 'Maroc', "United Nations Children's
Fund", 'Spanish', 'Marruecos', "United Nations Children's Fund", 'English', 'Morocco', 'United Nations Statistical Divi
sion', 'English', 'Morocco', 'World Bank', '[]')
(138, 'MCO', '492', 1, 1, 0, 1, 'MCO', '["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "WESTEUR", "IND", "ECA", "EURO"]',
'English', 'Monaco', "United Nations Children's Fund", 'French', 'Monaco', "United Nations Children's Fund", 'Spanish',
'Mónaco', "United Nations Children's Fund", 'English', 'Monaco', 'United Nations Statistical Division', 'English', 'Mon
aco', 'World Bank', '[]')
(139, 'MDA', '498', 1, 1, 0, 1, 'MDA', '["DEV", "LLDC", "NORTHAMEUR", "EASTEUR", "EUROPE", "GLOBAL", "CEECIS", "ECA",
"ECARO", "EURO"]', 'English', 'Republic of Moldova', "United Nations Children's Fund", 'French', 'République de Moldov
a', "United Nations Children's Fund", 'Spanish', 'República de Moldova', "United Nations Children's Fund", 'English',
'Republic of Moldova', 'United Nations Statistical Division', 'English', 'Moldova', 'World Bank', '[]')
(140, 'MDG', '450', 1, 1, 1, 1, 'MDG', '["DEVPING", "SSA", "LDC", "SSA", "AFRICA", "EASTAFR", "GLOBAL", "ESARO", "SSA",
"AFRO", "ESA"]', 'English', 'Madagascar', "United Nations Children's Fund", 'French', 'Madagascar', "United Nations Chi
```

ldren's Fund", 'Spanish', 'Madagascar', "United Nations Children's Fund", 'English', 'Madagascar', 'United Nations Statistical Division', 'English', 'Madagascar', 'World Bank', '[7]')

(141, 'MDV', '462', 1, 1, 0, 1, 'MDV', '[["DEVPING", "SIDS", "SOUTHCENASIA", "ASIA", "GLOBAL", "SOUTHASIA", "ROSA", "S A", "SEARO"]]', 'English', 'Maldives', "United Nations Children's Fund", 'French', 'Maldives', "United Nations Children's Fund", 'Spanish', 'Maldivas', "United Nations Children's Fund", 'English', 'Maldives', 'United Nations Statistical Division', 'English', 'Maldives', 'World Bank', '[]')

(142, 'MEX', '484', 1, 1, 1, 1, 'MEX', '[["DEVPING", "AMERICAS", "CENTRALAME", "GLOBAL", "LATINAME", "LACRO", "LAC"]]', 'English', 'Mexico', "United Nations Children's Fund", 'French', 'Mexique', "United Nations Children's Fund", 'Spanish', 'México', "United Nations Children's Fund", 'English', 'Mexico', 'United Nations Statistical Division', 'English', 'Mexico', 'World Bank', '[]')

(143, 'MHL', '584', 1, 1, 0, 1, 'MHL', '[["OCEAN", "DEVPING", "SIDS", "GLOBAL", "MICRONESIA", "OCEANIA", "EAPRO", "EAP", "WPRO"]]', 'English', 'Marshall Islands', "United Nations Children's Fund", 'French', 'Îles Marshall', "United Nations Children's Fund", 'Spanish', 'Islas Marshall', "United Nations Children's Fund", 'English', 'Marshall Islands', 'United Nations Statistical Division', 'English', 'Marshall Islands', 'World Bank', '[]')

(144, 'MKD', '807', 1, 1, 0, 1, 'MKD', '[["DEV", "LLDC", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEUR", "CEECIS", "ECA", "ECARO", "EURO"]]', 'English', 'North Macedonia', "United Nations Children's Fund", 'French', 'Macédoine du Nord', "United Nations Children's Fund", 'English', 'North Macedonia', 'World Bank', None, None, None, None, None, None, '[]')

(145, 'MLI', '466', 1, 1, 1, 1, 'MLI', '[["WESTAFR", "DEVPING", "SSA", "LDC", "LLDC", "SSA", "AFRICA", "GLOBAL", "WCAR O", "SSA", "AFRO"]]', 'English', 'Mali', "United Nations Children's Fund", 'French', 'Mali', "United Nations Children's Fund", 'Spanish', 'Malí', "United Nations Children's Fund", 'English', 'Mali', 'United Nations Statistical Division', 'English', 'Mali', 'World Bank', '[7, 18]')

(146, 'MLT', '470', 1, 1, 0, 1, 'MLT', '[["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEUR", "IND", "ECA", "EURO"]]', 'English', 'Malta', "United Nations Children's Fund", 'French', 'Malte', "United Nations Children's Fund", 'Spanish', 'Malta', "United Nations Children's Fund", 'English', 'Malta', 'United Nations Statistical Division', 'English', 'Malta', 'World Bank', '[]')

(147, 'MMR', '104', 1, 1, 1, 1, 'MMR', '[["DEVPING", "EASOUTASI", "LDC", "ASIA", "GLOBAL", "SOUTHEASTASIA", "EAPRO", "EA P", "SEARO"]]', 'English', 'Myanmar', "United Nations Children's Fund", 'French', 'Myanmar', "United Nations Children's Fund", 'Spanish', 'Myanmar', "United Nations Children's Fund", 'English', 'Myanmar', 'United Nations Statistical Division', 'English', 'Myanmar', 'World Bank', '[2, 7, 18]')

(148, 'MNE', '499', 1, 1, 0, 1, 'MNE', '[["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEUR", "CEECIS", "ECA", "ECARO", "EURO"]]', 'English', 'Montenegro', "United Nations Children's Fund", 'French', 'Monténégro', "United Nations Children's Fund", 'Spanish', 'Montenegro', "United Nations Children's Fund", 'English', 'Montenegro', 'United Nations Statistical Division', 'English', 'Montenegro', 'World Bank', '[]')

(149, 'MNG', '496', 1, 1, 0, 1, 'MNG', '[["DEVPING", "EASTASIA", "EASOUTASI", "LLDC", "ASIA", "EASTASIA", "GLOBAL", "EAP RO", "EAP", "WPRO"]]', 'English', 'Mongolia', "United Nations Children's Fund", 'French', 'Mongolie', "United Nations Children's Fund", 'Spanish', 'Mongolia', "United Nations Children's Fund", 'English', 'Mongolia', 'United Nations Statistical Division', 'English', 'Mongolia', 'World Bank', '[]')

(150, 'MNP', '580', 1, 0, 0, 1, 'MNP', '[["OCEAN", "DEVPING", "SIDS", "GLOBAL", "MICRONESIA", "OCEANIA"]]', 'English', 'Northern Mariana Islands', "United Nations Children's Fund", 'English', 'Northern Mariana Islands', 'United Nations Statistical Division', None, None, None, None, None, None, None, None, None, '[]')

(151, 'MOZ', '508', 1, 1, 1, 1, 'MOZ', '[["DEVPING", "SSA", "LDC", "SSA", "AFRICA", "EASTAFR", "GLOBAL", "ESARO", "SSA", "AFRO", "ESA"]]', 'English', 'Mozambique', "United Nations Children's Fund", 'French', 'Mozambique', "United Nations Children's Fund", 'Spanish', 'Mozambique', "United Nations Children's Fund", 'English', 'Mozambique', 'United Nations Statistical Division', 'English', 'Mozambique', 'World Bank', '[2, 7, 18]')

(152, 'MRT', '478', 1, 1, 1, 1, 'MRT', '[["WESTAFR", "DEVPING", "SSA", "LDC", "SSA", "AFRICA", "GLOBAL", "WCARO", "SSA",



nish', 'Nicaragua', "United Nations Children's Fund", 'English', 'Nicaragua', 'United Nations Statistical Division', 'English', 'Nicaragua', 'World Bank', '[7]')

(165, 'NIU', '570', 1, 1, 0, 1, 'NIU', '["OCEAN", "DEVPING", "SIDS", "GLOBAL", "OCEANIA", "POLYNES", "EAPRO", "EAP", "WPRO"]', 'English', 'Niue', "United Nations Children's Fund", 'French', 'Nioué', "United Nations Children's Fund", 'Spanish', 'Niue', "United Nations Children's Fund", 'English', 'Niue', 'United Nations Statistical Division', None, None, None, '[]')

(166, 'NLD', '528', 1, 1, 0, 1, 'NLD', '["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "WESTEUR", "IND", "ECA", "EURO"]', 'English', 'Netherlands (Kingdom of the)', "United Nations Children's Fund", 'French', 'Pays-Bas', "United Nations Children's Fund", 'Spanish', 'Países Bajos', "United Nations Children's Fund", 'English', 'Netherlands', 'United Nations Statistical Division', 'English', 'Netherlands', 'World Bank', '[]')

(167, 'NOR', '578', 1, 1, 0, 1, 'NOR', '["NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "IND", "ECA", "EURO"]', 'English', 'Norway', "United Nations Children's Fund", 'French', 'Norvège', "United Nations Children's Fund", 'Spanish', 'Noruega', "United Nations Children's Fund", 'English', 'Norway', 'United Nations Statistical Division', 'English', 'Norway', 'World Bank', '[]')

(168, 'NPL', '524', 1, 1, 1, 1, 'NPL', '["DEVPING", "LDC", "LLDC", "SOUTHCEASIA", "ASIA", "GLOBAL", "SOUTHASIA", "ROSA", "SA", "SEARO"]', 'English', 'Nepal', "United Nations Children's Fund", 'French', 'Népal', "United Nations Children's Fund", 'Spanish', 'Nepal', "United Nations Children's Fund", 'English', 'Nepal', 'United Nations Statistical Division', 'English', 'Nepal', 'World Bank', '[7, 18]')

(169, 'NRU', '520', 1, 1, 0, 1, 'NRU', '["OCEAN", "DEVPING", "SIDS", "GLOBAL", "MICRONESIA", "OCEANIA", "EAPRO", "EAP", "WPRO"]', 'English', 'Nauru', "United Nations Children's Fund", 'French', 'Nauru', "United Nations Children's Fund", 'Spanish', 'Nauru', "United Nations Children's Fund", 'English', 'Nauru', 'United Nations Statistical Division', 'English', 'Nauru', 'World Bank', '[]')

(170, 'NZL', '554', 1, 1, 0, 1, 'NZL', '["DEV", "AUSNZ", "GLOBAL", "OCEANIA", "IND", "EAP", "WPRO"]', 'English', 'New Zealand', "United Nations Children's Fund", 'French', 'Nouvelle-Zélande', "United Nations Children's Fund", 'Spanish', 'Nueva Zelanda', "United Nations Children's Fund", 'English', 'New Zealand', 'United Nations Statistical Division', 'English', 'New Zealand', 'World Bank', '[]')

(171, 'OMN', '512', 1, 1, 0, 1, 'OMN', '["DEVPING", "WESTASI", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "MENA", "MENARO", "EMRO"]', 'English', 'Oman', "United Nations Children's Fund", 'French', 'Oman', "United Nations Children's Fund", 'Spanish', 'Omán', "United Nations Children's Fund", 'English', 'Oman', 'United Nations Statistical Division', 'English', 'Oman', 'World Bank', '[]')

(172, 'PAK', '586', 1, 1, 1, 1, 'PAK', '["DEVPING", "SOUTHCEASIA", "ASIA", "GLOBAL", "SOUTHASIA", "ROSA", "SA", "EMRO"]', 'English', 'Pakistan', "United Nations Children's Fund", 'French', 'Pakistan', "United Nations Children's Fund", 'Spanish', 'Pakistán', "United Nations Children's Fund", 'English', 'Pakistan', 'United Nations Statistical Division', 'English', 'Pakistan', 'World Bank', '[7, 18]')

(173, 'PAN', '591', 1, 1, 0, 1, 'PAN', '["DEVPING", "AMERICAS", "CENTRALAME", "GLOBAL", "LATINAME", "LACRO", "LAC"]', 'English', 'Panama', "United Nations Children's Fund", 'French', 'Panama', "United Nations Children's Fund", 'Spanish', 'Panamá', "United Nations Children's Fund", 'English', 'Panama', 'United Nations Statistical Division', 'English', 'Panama', 'World Bank', '[]')

(174, 'PCN', '612', 1, 0, 0, 0, 'PCN', '["OCEAN", "DEVPING", "GLOBAL", "OCEANIA", "POLYNES"]', 'English', 'Pitcairn', "United Nations Children's Fund", 'English', 'Pitcairn', 'United Nations Statistical Division', None, None, None, None, None, None, None, '[]')

(175, 'PER', '604', 1, 1, 1, 1, 'PER', '["SOUTHAME", "DEVPING", "AMERICAS", "GLOBAL", "LATINAME", "LACRO", "LAC"]', 'English', 'Peru', "United Nations Children's Fund", 'French', 'Pérou', "United Nations Children's Fund", 'Spanish', 'Perú', "United Nations Children's Fund", 'English', 'Peru', 'United Nations Statistical Division', 'English', 'Peru', 'World Bank', '[]')

(176, 'PHL', '608', 1, 1, 1, 1, 'PHL', '[{"DEVPING", "EASOUTASI", "ASIA", "GLOBAL", "SOUTHEASTASIA", "EAPRO", "EAP", "WPRO"]]', 'English', 'Philippines', 'United Nations Children's Fund', 'French', 'Philippines', 'United Nations Children's Fund', 'Spanish', 'Filipinas', 'United Nations Children's Fund', 'English', 'Philippines', 'United Nations Statistical Division', 'English', 'Philippines', 'World Bank', '[2, 7, 18]')

(177, 'PLW', '585', 1, 1, 0, 1, 'PLW', '[{"OCEAN", "DEVPING", "SIDS", "GLOBAL", "MICRONESIA", "OCEANIA", "EAPRO", "EAP", "WPRO"]]', 'English', 'Palau', 'United Nations Children's Fund', 'French', 'Palaos', 'United Nations Children's Fund', 'Spanish', 'Palau', 'United Nations Children's Fund', 'English', 'Palau', 'United Nations Statistical Division', 'English', 'Palau', 'World Bank', '[]')

(178, 'PNG', '598', 1, 1, 1, 1, 'PNG', '[{"OCEAN", "DEVPING", "SIDS", "GLOBAL", "MELANESIA", "OCEANIA", "EAPRO", "EAP", "WPRO"]]', 'English', 'Papua New Guinea', 'United Nations Children's Fund', 'French', 'Papouasie-Nouvelle-Guinée', 'United Nations Children's Fund', 'Spanish', 'Papúa Nueva Guinea', 'United Nations Children's Fund', 'English', 'Papua New Guinea', 'United Nations Statistical Division', 'English', 'Papua New Guinea', 'World Bank', '[7]')

(179, 'POL', '616', 1, 1, 0, 1, 'POL', '[{"DEV", "NORTHAMEUR", "EASTEUR", "EUROPE", "GLOBAL", "IND", "ECA", "EURO"]]', 'English', 'Poland', 'United Nations Children's Fund', 'French', 'Pologne', 'United Nations Children's Fund', 'Spanish', 'Polonia', 'United Nations Children's Fund', 'English', 'Poland', 'United Nations Statistical Division', 'English', 'Poland', 'World Bank', '[]')

(180, 'PRI', '630', 1, 0, 0, 1, 'PRI', '[{"DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME"]]', 'English', 'Puerto Rico', 'United Nations Children's Fund', 'English', 'Puerto Rico', 'United Nations Statistical Division', None, None, None, None, None, None, '[]')

(181, 'PRK', '408', 1, 1, 1, 1, 'PRK', '[{"DEVPING", "EASTASIA", "EASOUTASI", "ASIA", "GLOBAL", "EAPRO", "EAP", "SEARO"]]', 'English', 'Democratic People's Republic of Korea', 'United Nations Children's Fund', 'French', 'République populaire démocratique de Corée', 'United Nations Children's Fund', 'Spanish', 'República Popular Democrática de Corea', 'United Nations Children's Fund', 'English', 'Dem. People's Republic of Korea', 'United Nations Statistical Division', 'English', 'Korea Dem. Rep.', 'World Bank', '[7]')

(182, 'PRT', '620', 1, 1, 0, 1, 'PRT', '[{"DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEUR", "IND", "ECA", "EURO"]]', 'English', 'Portugal', 'United Nations Children's Fund', 'French', 'Portugal', 'United Nations Children's Fund', 'Spanish', 'Portugal', 'United Nations Children's Fund', 'English', 'Portugal', 'United Nations Statistical Division', 'English', 'Portugal', 'World Bank', '[]')

(183, 'PRY', '600', 1, 1, 0, 1, 'PRY', '[{"SOUTHAME", "DEVPING", "LLDC", "AMERICAS", "GLOBAL", "LATINAME", "LACRO", "LACRO"]]', 'English', 'Paraguay', 'United Nations Children's Fund', 'French', 'Paraguay', 'United Nations Children's Fund', 'Spanish', 'Paraguay', 'United Nations Children's Fund', 'English', 'Paraguay', 'United Nations Statistical Division', 'English', 'Paraguay', 'World Bank', '[]')

(184, 'PSE', '275', 1, 1, 0, 1, 'PSE', '[{"DEVPING", "WESTASI", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "MENA", "MENARO"]]', 'English', 'State of Palestine', 'United Nations Children's Fund', 'French', 'État de Palestine\\xa0 \\xa0', 'United Nations Children's Fund', 'Spanish', 'Estado de Palestina\\xa0', 'United Nations Children's Fund', 'English', 'State of Palestine', 'United Nations Statistical Division', 'English', 'West Bank and Gaza', 'World Bank', '[]')

(185, 'PYF', '258', 1, 0, 0, 1, 'PYF', '[{"OCEAN", "DEVPING", "SIDS", "GLOBAL", "OCEANIA", "POLYNES"]]', 'English', 'French Polynesia', 'United Nations Children's Fund', 'English', 'French Polynesia', 'United Nations Statistical Division', None, None, None, None, None, None, None, '[]')

(186, 'QAT', '634', 1, 1, 0, 1, 'QAT', '[{"DEVPING", "WESTASI", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "MENA", "MENARO", "EMRO"]]', 'English', 'Qatar', 'United Nations Children's Fund', 'French', 'Qatar', 'United Nations Children's Fund', 'Spanish', 'Qatar', 'United Nations Children's Fund', 'English', 'Qatar', 'United Nations Statistical Division', 'English', 'Qatar', 'World Bank', '[]')

(187, 'REU', '638', 1, 0, 0, 1, 'REU', '[{"DEVPING", "SSA", "SSA", "AFRICA", "EASTAFR", "GLOBAL"]]', 'English', 'Réunion', 'United Nations Children's Fund', 'English', 'Réunion', 'United Nations Statistical Division', None, None, None, No

ne, None, None, None, None, None, '[]')

(188, 'ROU', '642', 1, 1, 0, 1, 'ROU', '["DEV", "NORTHAMEUR", "EASTEUR", "EUROPE", "GLOBAL", "CEECIS", "ECA", "ECARO", "EURO"]]', 'English', 'Romania', 'United Nations Children's Fund', 'French', 'Roumanie', 'United Nations Children's Fund', 'Spanish', 'Rumania', 'United Nations Children's Fund', 'English', 'Romania', 'United Nations Statistical Division', 'English', 'Romania', 'World Bank', '[]')

(189, 'RUS', '643', 1, 1, 0, 1, 'RUS', '["DEV", "NORTHAMEUR", "EASTEUR", "EUROPE", "GLOBAL", "CEECIS", "ECA", "ECARO", "EURO"]]', 'English', 'Russian Federation', 'United Nations Children's Fund', 'French', 'Fédération de Russie', 'United Nations Children's Fund', 'Spanish', 'Federación de Rusia', 'United Nations Children's Fund', 'English', 'Russian Federation', 'United Nations Statistical Division', 'English', 'Russian Federation', 'World Bank', '[]')

(190, 'RWA', '646', 1, 1, 1, 1, 'RWA', '["DEVPING", "SSA", "LDC", "LLDC", "SSA", "AFRICA", "EASTAFR", "GLOBAL", "ESARO", "AFRO", "ESA"]]', 'English', 'Rwanda', 'United Nations Children's Fund', 'French', 'Rwanda', 'United Nations Children's Fund', 'Spanish', 'Rwanda', 'United Nations Children's Fund', 'English', 'Rwanda', 'United Nations Statistical Division', 'English', 'Rwanda', 'World Bank', '[2, 7]')

(191, 'SAU', '682', 1, 1, 0, 1, 'SAU', '["DEVPING", "WESTASI", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "MENA", "MENARO", "EMRO"]]', 'English', 'Saudi Arabia', 'United Nations Children's Fund', 'French', 'Arabie saoudite', 'United Nations Children's Fund', 'Spanish', 'Arabia Saudita', 'United Nations Children's Fund', 'English', 'Saudi Arabia', 'United Nations Statistical Division', 'English', 'Saudi Arabia', 'World Bank', '[]')

(192, 'SDN', '729', 1, 1, 1, 1, 'SDN', '["DEVPING", "SSA", "LDC", "WASINAFR", "AFRICA", "GLOBAL", "NORTHAFR", "MENARO", "SSA", "EMRO"]]', 'English', 'Sudan', 'United Nations Children's Fund', 'French', 'Soudan', 'United Nations Children's Fund', 'Spanish', 'Sudán', 'United Nations Children's Fund', 'English', 'Sudan', 'United Nations Statistical Division', 'English', 'Sudan', 'World Bank', '[7, 18]')

(193, 'SEN', '686', 1, 1, 1, 1, 'SEN', '["WESTAFR", "DEVPING", "SSA", "LDC", "SSA", "AFRICA", "GLOBAL", "WCARO", "SSA", "AFRO"]]', 'English', 'Senegal', 'United Nations Children's Fund', 'French', 'Sénégal', 'United Nations Children's Fund', 'Spanish', 'Senegal', 'United Nations Children's Fund', 'English', 'Senegal', 'United Nations Statistical Division', 'English', 'Senegal', 'World Bank', '[7, 18]')

(194, 'SGP', '702', 1, 1, 0, 1, 'SGP', '["DEVPING", "EASOUTASI", "SIDS", "ASIA", "GLOBAL", "SOUTHEASTASIA", "EAPRO", "EAP", "WPRO"]]', 'English', 'Singapore', 'United Nations Children's Fund', 'French', 'Singapour', 'United Nations Children's Fund', 'Spanish', 'Singapur', 'United Nations Children's Fund', 'English', 'Singapore', 'United Nations Statistical Division', 'English', 'Singapore', 'World Bank', '[]')

(195, 'SGS', '239', 1, 0, 0, 0, 'SGS', '["SOUTHAME", "DEVPING", "AMERICAS", "GLOBAL", "LATINAME"]]', 'English', 'South Georgia and the South Sandwich Islands', 'United Nations Children's Fund', 'English', 'South Georgia and the South Sandwich Islands', 'United Nations Statistical Division', None, '[]')

(196, 'SHN', '654', 1, 0, 0, 1, 'SHN', '["WESTAFR", "DEVPING", "SSA", "SSA", "AFRICA", "GLOBAL"]]', 'English', 'Saint Helena Ascension and Tristan da Cunha', 'United Nations Children's Fund', 'English', 'Saint Helena', 'United Nations Statistical Division', None, None, None, None, None, None, None, None, None, '[]')

(197, 'SJM', '744', 1, 0, 0, 0, 'SJM', '["NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "GLOBAL"]]', 'English', 'Svalbard and Jan Mayen Islands', 'United Nations Children's Fund', 'English', 'Svalbard and Jan Mayen Islands', 'United Nations Statistical Division', None, '[]')

(198, 'SLB', '90', 1, 1, 1, 1, 'SLB', '["OCEAN", "DEVPING", "LDC", "SIDS", "GLOBAL", "MELANESIA", "OCEANIA", "EAPRO", "EAP", "WPRO"]]', 'English', 'Solomon Islands', 'United Nations Children's Fund', 'French', 'îles Salomon', 'United Nations Children's Fund', 'Spanish', 'Islas Salomón', 'United Nations Children's Fund', 'English', 'Solomon Islands', 'United Nations Statistical Division', 'English', 'Solomon Islands', 'World Bank', '[7]')

(199, 'SLE', '694', 1, 1, 1, 1, 'SLE', '["WESTAFR", "DEVPING", "SSA", "LDC", "SSA", "AFRICA", "GLOBAL", "WCARO", "SSA", "AFRO"]]', 'English', 'Sierra Leone', 'United Nations Children's Fund', 'French', 'Sierra Leone', 'United Nations Children's Fund', 'Spanish', 'Sierra Leona', 'United Nations Children's Fund', 'English', 'Sierra Leone', 'United Nations Sta

tistical Division', 'English', 'Sierra Leone', 'World Bank', '[7, 18]')  
(200, 'SLV', '222', 1, 1, 0, 1, 'SLV', '["DEVPING", "AMERICAS", "CENTRALAME", "GLOBAL", "LATINAME", "LACRO", "LAC"]',  
'English', 'El Salvador', "United Nations Children's Fund", 'French', 'El Salvador', "United Nations Children's Fund",  
'Spanish', 'El Salvador', "United Nations Children's Fund", 'English', 'El Salvador', 'United Nations Statistical Division',  
'English', 'El Salvador', 'World Bank', '[]')  
(201, 'SMR', '674', 1, 1, 0, 1, 'SMR', '["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEUR", "IND", "ECA", "EURO"]',  
'English', 'San Marino', "United Nations Children's Fund", 'French', 'Saint-Marin', "United Nations Children's Fund",  
'Spanish', 'San Marino', "United Nations Children's Fund", 'English', 'San Marino', 'United Nations Statistical Division',  
'English', 'San Marino', 'World Bank', '[]')  
(202, 'SOM', '706', 1, 1, 1, 1, 'SOM', '["DEVPING", "SSA", "LDC", "SSA", "AFRICA", "EASTAFR", "GLOBAL", "ESARO", "AFR  
O", "EMRO", "ESA"]', 'English', 'Somalia', "United Nations Children's Fund", 'French', 'Somalie', "United Nations Children's Fund",  
'Spanish', 'Somalia', "United Nations Children's Fund", 'English', 'Somalia', 'United Nations Statistical Division',  
'English', 'Somalia', 'World Bank', '[7, 18]')  
(203, 'SPM', '666', 1, 0, 0, 1, 'SPM', '["DEV", "NORTHAMEUR", "AMERICAS", "GLOBAL", "NORTHAME"]', 'English', 'Saint Pierre and Miquelon',  
"United Nations Children's Fund", 'English', 'Saint Pierre and Miquelon', 'United Nations Statistical Division',  
None, None, None, None, None, None, None, None, None, '[]')  
(204, 'SRB', '688', 1, 1, 0, 1, 'SRB', '["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEUR", "CEECIS", "ECA", "ECARO",  
"EURO"]', 'English', 'Serbia', "United Nations Children's Fund", 'French', 'Serbie', "United Nations Children's Fund",  
'Spanish', 'Serbia', "United Nations Children's Fund", 'English', 'Serbia', 'United Nations Statistical Division', 'Eng  
lish', 'Serbia', 'World Bank', '[]')  
(205, 'SSD', '728', 1, 1, 1, 1, 'SSD', '["DEVPING", "SSA", "LDC", "LLDC", "SSA", "AFRICA", "EASTAFR", "GLOBAL", "ESAR  
O", "SSA", "EMRO", "ESA"]', 'English', 'South Sudan', "United Nations Children's Fund", 'French', 'Soudan du Sud', "United Nations Children's Fund",  
'Spanish', 'Sudán del Sur', "United Nations Children's Fund", 'English', 'South Sudan', 'United Nations Statistical Division',  
'English', 'South Sudan', 'World Bank', '[7, 18]')  
(206, 'STP', '678', 1, 1, 1, 1, 'STP', '["MIDDLEAFR", "DEVPING", "SSA", "LDC", "SIDS", "SSA", "AFRICA", "GLOBAL", "WCAR  
O", "SSA", "AFRO"]', 'English', 'Sao Tome and Principe', "United Nations Children's Fund", 'French', 'Sao Tomé-et-Princ  
ipe', "United Nations Children's Fund", 'Spanish', 'Santo Tomé y Príncipe', "United Nations Children's Fund", 'Englis  
h', 'Sao Tome and Principe', 'United Nations Statistical Division', 'English', 'São Tomé and Príncipe', 'World Bank',  
'[7]')  
(207, 'SUR', '740', 1, 1, 0, 1, 'SUR', '["SOUTHAME", "DEVPING", "SIDS", "AMERICAS", "GLOBAL", "LATINAME", "LACRO", "LA  
C"]', 'English', 'Suriname', "United Nations Children's Fund", 'French', 'Suriname', "United Nations Children's Fund",  
'Spanish', 'Surinam', "United Nations Children's Fund", 'English', 'Suriname', 'United Nations Statistical Division',  
'English', 'Suriname', 'World Bank', '[]')  
(208, 'SVK', '703', 1, 1, 0, 1, 'SVK', '["DEV", "NORTHAMEUR", "EASTEUR", "EUROPE", "GLOBAL", "IND", "ECA", "EURO"]', 'E  
nglish', 'Slovakia', "United Nations Children's Fund", 'French', 'Slovaquie', "United Nations Children's Fund", 'Spanis  
h', 'Eslovaquia', "United Nations Children's Fund", 'English', 'Slovakia', 'United Nations Statistical Division', 'Eng  
lish', 'Slovak Republic', 'World Bank', '[]')  
(209, 'SVN', '705', 1, 1, 0, 1, 'SVN', '["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEUR", "IND", "ECA", "EURO"]',  
'English', 'Slovenia', "United Nations Children's Fund", 'French', 'Slovénie', "United Nations Children's Fund", 'Spanis  
h', 'Eslovenia', "United Nations Children's Fund", 'English', 'Slovenia', 'United Nations Statistical Division', 'Engl  
ish', 'Slovenia', 'World Bank', '[]')  
(210, 'SWE', '752', 1, 1, 0, 1, 'SWE', '["NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "IND", "ECA", "EURO"]',  
'English', 'Sweden', "United Nations Children's Fund", 'French', 'Suède', "United Nations Children's Fund", 'Spanish',  
'Suecia', "United Nations Children's Fund", 'English', 'Sweden', 'United Nations Statistical Division', 'English', 'Swe  
den', 'World Bank', '[]')

(211, 'SWZ', '748', 1, 1, 1, 1, 'SWZ', '[{"DEVPING", "SSA", "LLDC", "SSA", "AFRICA", "GLOBAL", "SOUTHAFR", "ESARO", "SSA", "AFRO", "ESA"]]', 'English', 'Eswatini', 'United Nations Children's Fund', 'French', 'Eswatini (l')', 'United Nations Children's Fund', 'Spanish', 'Swazilandia', 'United Nations Children's Fund', 'English', 'Eswatini', 'United Nations Statistical Division', 'English', 'Swaziland', 'World Bank', '[]')

(212, 'SXM', '534', 1, 0, 0, 1, 'SXM', '[{"DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME"]]', 'English', 'Sint Maarten', 'United Nations Children's Fund', 'English', 'Sint Maarten (Dutch part)', 'United Nations Statistical Division', None, None, None, None, None, None, None, None, '[]')

(213, 'SYC', '690', 1, 1, 0, 1, 'SYC', '[{"DEVPING", "SSA", "SIDS", "SSA", "AFRICA", "EASTAFR", "GLOBAL", "ESARO", "SSA", "AFRO", "ESA"]]', 'English', 'Seychelles', 'United Nations Children's Fund', 'French', 'Seychelles', 'United Nations Children's Fund', 'Spanish', 'Seychelles', 'United Nations Children's Fund', 'English', 'Seychelles', 'United Nations Statistical Division', 'English', 'Seychelles', 'World Bank', '[]')

(214, 'SYR', '760', 1, 1, 0, 1, 'SYR', '[{"DEVPING", "WESTASI", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "MENA", "MENARO", "EMRO"]]', 'English', 'Syrian Arab Republic', 'United Nations Children's Fund', 'French', 'République arabe syrienne', 'United Nations Children's Fund', 'Spanish', 'República Árabe Siria', 'United Nations Children's Fund', 'English', 'Syrian Arab Republic', 'United Nations Statistical Division', 'English', 'Syrian Arab Republic', 'World Bank', '[7, 18]')

(215, 'TCA', '796', 1, 1, 0, 1, 'TCA', '[{"DEVPING", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME", "NC", "LAC", "LACRO"]]', 'English', 'Turks and Caicos Islands', 'United Nations Children's Fund', 'French', 'Îles Turques-et-Caïques', 'United Nations Children's Fund', 'Spanish', 'Islas Turcas y Caicos', 'United Nations Children's Fund', 'English', 'Turks and Caicos Islands', 'United Nations Statistical Division', None, None, None, '[]')

(216, 'TCD', '148', 1, 1, 1, 1, 'TCD', '[{"MIDDLEAFR", "DEVPING", "SSA", "LDC", "LLDC", "SSA", "AFRICA", "GLOBAL", "WCARO", "SSA", "AFRO"]]', 'English', 'Chad', 'United Nations Children's Fund', 'French', 'Tchad', 'United Nations Children's Fund', 'Spanish', 'Chad', 'United Nations Children's Fund', 'English', 'Chad', 'United Nations Statistical Division', 'English', 'Chad', 'World Bank', '[2, 7, 18]')

(217, 'TGO', '768', 1, 1, 1, 1, 'TGO', '[{"WESTAFR", "DEVPING", "SSA", "LDC", "SSA", "AFRICA", "GLOBAL", "WCARO", "SSA", "AFRO"]]', 'English', 'Togo', 'United Nations Children's Fund', 'French', 'Togo', 'United Nations Children's Fund', 'Spanish', 'Togo', 'United Nations Children's Fund', 'English', 'Togo', 'United Nations Statistical Division', 'English', 'Togo', 'World Bank', '[7, 18]')

(218, 'THA', '764', 1, 1, 0, 1, 'THA', '[{"DEVPING", "EASOUTASI", "ASIA", "GLOBAL", "SOUTHEASTASIA", "EAPRO", "EAP", "SEASRO"]]', 'English', 'Thailand', 'United Nations Children's Fund', 'French', 'Thaïlande', 'United Nations Children's Fund', 'Spanish', 'Tailandia', 'United Nations Children's Fund', 'English', 'Thailand', 'United Nations Statistical Division', 'English', 'Thailand', 'World Bank', '[]')

(219, 'TJK', '762', 1, 1, 1, 1, 'TJK', '[{"CAUCCEN", "DEVPING", "LLDC", "SOUTHCEASIA", "ASIA", "CENTRALASIA", "GLOBAL", "CEECIS", "ECA", "ECARO", "EURO"]]', 'English', 'Tajikistan', 'United Nations Children's Fund', 'French', 'Tadjikistan', 'United Nations Children's Fund', 'Spanish', 'Tayikistán', 'United Nations Children's Fund', 'English', 'Tajikistan', 'United Nations Statistical Division', 'English', 'Tajikistan', 'World Bank', '[7]')

(220, 'TKL', '772', 1, 1, 0, 1, 'TKL', '[{"OCEAN", "DEVPING", "GLOBAL", "OCEANIA", "POLYNES", "EAPRO", "EAP"]]', 'English', 'Tokelau', 'United Nations Children's Fund', 'French', 'Tokélaou', 'United Nations Children's Fund', 'Spanish', 'Tokelau', 'United Nations Children's Fund', 'English', 'Tokelau', 'United Nations Statistical Division', None, None, None, '[]')

(221, 'TKM', '795', 1, 1, 1, 1, 'TKM', '[{"CAUCCEN", "DEVPING", "LLDC", "SOUTHCEASIA", "ASIA", "CENTRALASIA", "GLOBAL", "CEECIS", "ECA", "ECARO", "EURO"]]', 'English', 'Turkmenistan', 'United Nations Children's Fund', 'French', 'Turkménistan', 'United Nations Children's Fund', 'Spanish', 'Turkmenistán', 'United Nations Children's Fund', 'English', 'Turkmenistan', 'United Nations Statistical Division', 'English', 'Turkmenistan', 'World Bank', '[]')

(222, 'TLS', '626', 1, 1, 0, 1, 'TLS', '[{"DEVPING", "EASOUTASI", "LDC", "SIDS", "ASIA", "GLOBAL", "SOUTHEASTASIA", "EAP"]]', 'English', 'Timor-Leste', 'United Nations Children's Fund', 'French', 'Timor-Leste', 'United Nations Statistical Division', 'English', 'Timor-Leste', 'World Bank', '[]')

RO", "EAP", "SEARO"]]', 'English', 'Timor-Leste', "United Nations Children's Fund", 'French', 'Timor-Leste', "United Nations Children's Fund", 'Spanish', 'Timor-Leste', "United Nations Children's Fund", 'English', 'Timor-Leste', 'United Nations Statistical Division', 'English', 'Timor-Leste', 'World Bank', '[]')  
(223, 'TON', '776', 1, 1, 0, 1, 'TON', '["OCEAN", "DEVPING", "SIDS", "GLOBAL", "OCEANIA", "POLYNES", "EAPRO", "EAP", "WPRO"]', 'English', 'Tonga', "United Nations Children's Fund", 'French', 'Tonga', "United Nations Children's Fund", 'Spanish', 'Tonga', "United Nations Children's Fund", 'English', 'Tonga', 'United Nations Statistical Division', 'English', 'Tonga', 'World Bank', '[]')  
(224, 'TTO', '780', 1, 1, 0, 1, 'TTO', '["DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME", "LACRO", "LAC"]', 'English', 'Trinidad and Tobago', "United Nations Children's Fund", 'French', 'Trinité-et-Tobago', "United Nations Children's Fund", 'Spanish', 'Trinidad y Tabago', "United Nations Children's Fund", 'English', 'Trinidad and Tobago', 'United Nations Statistical Division', 'English', 'Trinidad and Tobago', 'World Bank', '[]')  
(225, 'TUN', '788', 1, 1, 0, 1, 'TUN', '["DEVPING", "NORTHAFR", "WASINAFR", "AFRICA", "GLOBAL", "NORTHAFR", "MENA", "MENARO", "EMRO"]', 'English', 'Tunisia', "United Nations Children's Fund", 'French', 'Tunisie', "United Nations Children's Fund", 'Spanish', 'Túnez', "United Nations Children's Fund", 'English', 'Tunisia', 'United Nations Statistical Division', 'English', 'Tunisia', 'World Bank', '[]')  
(226, 'TUR', '792', 1, 1, 0, 1, 'TUR', '["DEVPING", "WESTASI", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "CEECIS", "ECA", "ECARO", "EURO"]', 'English', 'Republic of Türkiye', "United Nations Children's Fund", 'French', 'Turquie', "United Nations Children's Fund", 'Spanish', 'Turquía', "United Nations Children's Fund", 'English', 'Turkey', 'United Nations Statistical Division', 'English', 'Turkey', 'World Bank', '[]')  
(227, 'TUV', '798', 1, 1, 0, 1, 'TUV', '["OCEAN", "DEVPING", "LDC", "SIDS", "GLOBAL", "OCEANIA", "POLYNES", "EAPRO", "EAP", "WPRO"]', 'English', 'Tuvalu', "United Nations Children's Fund", 'French', 'Tuvalu', "United Nations Children's Fund", 'Spanish', 'Tuvalu', "United Nations Children's Fund", 'English', 'Tuvalu', 'United Nations Statistical Division', 'English', 'Tuvalu', 'World Bank', '[]')  
(228, 'TZA', '834', 1, 1, 1, 1, 'TZA', '["DEVPING", "SSA", "LDC", "SSA", "AFRICA", "EASTAFR", "GLOBAL", "ESARO", "SSA", "AFRO", "ESA"]', 'English', 'United Republic of Tanzania', "United Nations Children's Fund", 'French', 'République-Unie de Tanzanie', "United Nations Children's Fund", 'Spanish', 'República Unida de Tanzania', "United Nations Children's Fund", 'English', 'United Republic of Tanzania', 'United Nations Statistical Division', 'English', 'Tanzania', 'World Bank', '[2, 7, 18]')  
(229, 'UGA', '800', 1, 1, 1, 1, 'UGA', '["DEVPING", "SSA", "LDC", "LLDC", "SSA", "AFRICA", "EASTAFR", "GLOBAL", "ESARO", "SSA", "AFRO", "ESA"]', 'English', 'Uganda', "United Nations Children's Fund", 'French', 'Ouganda', "United Nations Children's Fund", 'Spanish', 'Uganda', "United Nations Children's Fund", 'English', 'Uganda', 'United Nations Statistical Division', 'English', 'Uganda', 'World Bank', '[2, 7, 18]')  
(230, 'UKR', '804', 1, 1, 0, 1, 'UKR', '["DEV", "NORTHAMEUR", "EASTEUR", "EUROPE", "GLOBAL", "CEECIS", "ECA", "ECARO", "EURO"]', 'English', 'Ukraine', "United Nations Children's Fund", 'French', 'Ukraine', "United Nations Children's Fund", 'Spanish', 'Ucrania', "United Nations Children's Fund", 'English', 'Ukraine', 'United Nations Statistical Division', 'English', 'Ukraine', 'World Bank', '[2, 7]')  
(231, 'UMI', '581', 1, 0, 0, 0, 'UMI', '["OCEAN", "DEVPING", "GLOBAL", "MICRONESIA", "OCEANIA"]', 'English', 'United States minor outlying islands', "United Nations Children's Fund", 'English', 'United States minor outlying islands', 'United Nations Statistical Division', None, '[]')  
(232, 'URY', '858', 1, 1, 0, 1, 'URY', '["SOUTHAME", "DEVPING", "AMERICAS", "GLOBAL", "LATINAME", "LACRO", "LAC"]', 'English', 'Uruguay', "United Nations Children's Fund", 'French', 'Uruguay', "United Nations Children's Fund", 'Spanish', 'Uruguay', "United Nations Children's Fund", 'English', 'Uruguay', 'United Nations Statistical Division', 'English', 'Uruguay', 'World Bank', '[]')  
(233, 'USA', '840', 1, 1, 0, 1, 'USA', '["DEV", "NORTHAMEUR", "AMERICAS", "GLOBAL", "NORTHAME", "IND", "NA"]', 'English', 'United States', "United Nations Children's Fund", 'French', 'États-Unis', "United Nations Children's Fund", 'Spani

sh', 'Estados Unidos', "United Nations Children's Fund", 'English', 'United States of America', 'United Nations Statistical Division', 'English', 'United States', 'World Bank', '[]')

(234, 'UZB', '860', 1, 1, 1, 'UZB', '[["CAUCCEN", "DEVPING", "LLDC", "SOUTHCEASIA", "ASIA", "CENTRALASIA", "GLOBAL", "CEECLIS", "ECA", "ECARO", "EURO"]]', 'English', 'Uzbekistan', "United Nations Children's Fund", 'French', 'Ouzbékistan', "United Nations Children's Fund", 'Spanish', 'Uzbekistán', "United Nations Children's Fund", 'English', 'Uzbekistan', 'United Nations Statistical Division', 'English', 'Uzbekistan', 'World Bank', '[]')

(235, 'VAT', '336', 1, 1, 0, 1, 'VAT', '[["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEUR", "IND", "ECA"]]', 'English', 'Holy See', "United Nations Children's Fund", 'French', 'Saint-Siège', "United Nations Children's Fund", 'Spanish', 'Santa Sede', "United Nations Children's Fund", 'English', 'Holy See', 'United Nations Statistical Division', None, None, None, '[]')

(236, 'VCT', '670', 1, 1, 0, 1, 'VCT', '[["DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME", "LACRO", "LAC"]]', 'English', 'Saint Vincent and the Grenadines', "United Nations Children's Fund", 'French', 'Saint-Vincent-et-les Grenadines', "United Nations Children's Fund", 'Spanish', 'San Vicente y las Granadinas', "United Nations Children's Fund", 'English', 'Saint Vincent and the Grenadines', 'United Nations Statistical Division', 'English', 'St. Vincent and the Grenadines', 'World Bank', '[]')

(237, 'VEN', '862', 1, 1, 0, 1, 'VEN', '[["SOUTHAME", "DEVPING", "AMERICAS", "GLOBAL", "LATINAME", "LACRO", "LAC"]]', 'English', 'Venezuela (Bolivarian Republic of)', "United Nations Children's Fund", 'French', 'Venezuela (République bolivarienne du)', "United Nations Children's Fund", 'Spanish', 'Venezuela (República Bolivariana de)', "United Nations Children's Fund", 'English', 'Venezuela (Bolivarian Republic of)', 'United Nations Statistical Division', 'English', 'Venezuela RB', 'World Bank', '[]')

(238, 'VGB', '92', 1, 1, 0, 1, 'VGB', '[["DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME", "NC", "LAC", "LACRO"]]', 'English', 'British Virgin Islands', "United Nations Children's Fund", 'French', 'Îles Vierges britanniques', "United Nations Children's Fund", 'Spanish', 'Islas Vírgenes Británicas', "United Nations Children's Fund", 'English', 'British Virgin Islands', 'United Nations Statistical Division', 'English', 'British Virgin Islands', 'World Bank', '[]')

(239, 'VIR', '850', 1, 0, 0, 1, 'VIR', '[["DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLOBAL", "LATINAME"]]', 'English', 'Virgin Islands U.S.', "United Nations Children's Fund", 'English', 'United States Virgin Islands', 'United Nations Statistical Division', None, None, None, None, None, None, None, None, '[]')

(240, 'VNM', '704', 1, 1, 1, 1, 'VNM', '[["DEVPING", "EASOUTASI", "ASIA", "GLOBAL", "SOUTHEASTASIA", "EAPRO", "EAP", "WPRO"]]', 'English', 'Viet Nam', "United Nations Children's Fund", 'French', 'Viet Nam', "United Nations Children's Fund", 'Spanish', 'Viet Nam', "United Nations Children's Fund", 'English', 'Viet Nam', 'United Nations Statistical Division', 'English', 'Vietnam', 'World Bank', '[7, 18]')

(241, 'VUT', '548', 1, 1, 0, 1, 'VUT', '[["OCEAN", "DEVPING", "LDC", "SIDS", "GLOBAL", "MELANESIA", "OCEANIA", "EAPRO", "EAP", "WPRO"]]', 'English', 'Vanuatu', "United Nations Children's Fund", 'French', 'Vanuatu', "United Nations Children's Fund", 'Spanish', 'Vanuatu', "United Nations Children's Fund", 'English', 'Vanuatu', 'United Nations Statistical Division', 'English', 'Vanuatu', 'World Bank', '[]')

(242, 'WLF', '876', 1, 0, 0, 1, 'WLF', '[["OCEAN", "DEVPING", "GLOBAL", "OCEANIA", "POLYNES"]]', 'English', 'Wallis and Futuna', "United Nations Children's Fund", 'English', 'Wallis and Futuna Islands', 'United Nations Statistical Division', None, None, None, None, None, None, None, None, '[]')

(243, 'WSM', '882', 1, 1, 0, 1, 'WSM', '[["OCEAN", "DEVPING", "SIDS", "GLOBAL", "OCEANIA", "POLYNES", "EAPRO", "EAP", "WPRO"]]', 'English', 'Samoa', "United Nations Children's Fund", 'French', 'Samoa', "United Nations Children's Fund", 'Spanish', 'Samoa', "United Nations Children's Fund", 'English', 'Samoa', 'United Nations Statistical Division', 'English', 'Samoa', 'World Bank', '[]')

(244, 'XKX', '', 1, 0, 0, 1, 'XKX', '[["GLOBAL"]]', 'English', 'Kosovo', "United Nations Children's Fund", None, None, None, None, None, None, None, None, '[]')

```
(245, 'YEM', '887', 1, 1, 1, 1, 'YEM', '[{"DEVPING", "WESTASIA", "LDC", "WASINAFR", "ASIA", "GLOBAL", "WESTASIA", "MENA", "MENARO", "EMRO"]]', 'English', 'Yemen', 'United Nations Children's Fund', 'French', 'Yémen', 'United Nations Children's Fund', 'Spanish', 'Yemen', 'United Nations Children's Fund', 'English', 'Yemen', 'United Nations Statistical Division', 'English', 'Yemen Rep.', 'World Bank', '[7, 18]')
(246, 'ZAF', '710', 1, 1, 1, 1, 'ZAF', '[{"DEVPING", "SSA", "SSA", "AFRICA", "GLOBAL", "SOUTHAFR", "ESARO", "SSA", "AFR0", "ESA"]]', 'English', 'South Africa', 'United Nations Children's Fund', 'French', 'Afrique du Sud', 'United Nations Children's Fund', 'Spanish', 'Sudáfrica', 'United Nations Children's Fund', 'English', 'South Africa', 'United Nations Statistical Division', 'English', 'South Africa', 'World Bank', '[]')
(247, 'ZMB', '894', 1, 1, 1, 1, 'ZMB', '[{"DEVPING", "SSA", "LDC", "LLDC", "SSA", "AFRICA", "EASTAFR", "GLOBAL", "ESARO", "SSA", "AFRO", "ESA"]]', 'English', 'Zambia', 'United Nations Children's Fund', 'French', 'Zambie', 'United Nations Children's Fund', 'Spanish', 'Zambia', 'United Nations Children's Fund', 'English', 'Zambia', 'United Nations Statistical Division', 'English', 'Zambia', 'World Bank', '[2, 7]')
(248, 'ZWE', '716', 1, 1, 1, 1, 'ZWE', '[{"DEVPING", "SSA", "LLDC", "SSA", "AFRICA", "EASTAFR", "GLOBAL", "ESARO", "SSA", "AFRO", "ESA"]]', 'English', 'Zimbabwe', 'United Nations Children's Fund', 'French', 'Zimbabwe', 'United Nations Children's Fund', 'Spanish', 'Zimbabwe', 'United Nations Children's Fund', 'English', 'Zimbabwe', 'United Nations Statistical Division', 'English', 'Zimbabwe', 'World Bank', '[2, 7, 18]')
(249, 'TWN', None, 1, 0, 0, 1, 'TWN', '[{"GLOBAL"}]', 'English', 'Other non specified', 'United Nations Children's Fund', None, '[]')
```

In [239...]

```
def select_table(db_name, table_name):
    try:
        # Connect to SQLite database
        conn = sqlite3.connect(db_name)
        print("Opened database successfully")

        # Query to select all rows from the table
        query = f"SELECT * FROM {table_name}"

        # Read the selected table into a pandas DataFrame
        df = pd.read_sql_query(query, conn)

        return df

    except sqlite3.Error as e:
        print("Error connecting to SQLite database:", e)
        return None
    finally:
        # Close connection
        conn.close()

df = select_table("unicef.db", "Actualcountries")
df.head()
```

Opened database successfully

Out[239]:

	countryId	iso3	m49	isCurrent	isSOWC	isCountDown	isPublished	cndcountryCode	Regions_Lists	Language1	...	Language3	V
0	1	ABW	533	1	0	0	1	ABW	["DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLO..."]	English	...	None	
1	2	AFG	4	1	0	1	1	AFG	["DEVPING", "LDC", "LLDC", "SOUTHCEASIA", "AS..."]	English	...	Spanish	Afga
2	3	AGO	24	1	1	1	1	AGO	["MIDDLEAFR", "DEVPING", "SSA", "LDC", "SSA", ...]	English	...	Spanish	A
3	4	AIA	660	1	1	0	1	AIA	["DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLO..."]	English	...	Spanish	A
4	5	ALA	248	1	0	0	0	ALA	["NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "G..."]	English	...	Spanish	Islas

5 rows × 25 columns

In [240...]: `dfcopy=df.copy()`

In [241...]: `# Calculate the sum of non-null values across 'Organization1' to 'Organization5' columns`  
`dfcopy['Total_Organization_By_country'] = dfcopy[['Organization1', 'Organization2', 'Organization3', 'Organization4', 'Organization5']].sum(axis=1)`  
`dfcopy.head(1)`

Out[241]:

	countryId	iso3	m49	isCurrent	isSOWC	isCountDown	isPublished	cndcountryCode	Regions_Lists	Language1	...	Value3	Organization
0	1	ABW	533	1	0	0	1	ABW	["AMERICAS", "CARIBB", "GLO..."]	English	...	None	Nor

1 rows × 26 columns

In [242...]

```
# Define a Lambda function to count the total number of elements in a List, excluding empty Lists and treating each comma-separated string as a single element
count_elements = lambda x: len(x.split(',')) if x.strip('[]') else 0
# Apply the Lambda function to each row in the 'Sector_Lists' column and assign the result to a new column 'Total_Sector_By_country'
dfcopy['Total_Sector_By_country'] = dfcopy['Sector_Lists'].apply(count_elements)
dfcopy.head(2)
```

Out[242]:

	countryId	iso3	m49	isCurrent	isSOWC	isCountDown	isPublished	cndcountryCode	Regions_Lists	Language1	...	Organization3	L
0	1	ABW	533	1	0	0	1	ABW	["AMERICAS", "CARIBB", "GLO..."]	English	...	None	
1	2	AFG	4	1	0	1	1	AFG	["SOUTHCEASIA", "AS..."]	English	...	United Nations Children's Fund	

2 rows × 27 columns

In [243...]

```
dfcopy2 = dfcopy.copy()
```

In [244...]

```
# Check for duplicate rows
duplicate_rows = dfcopy2.duplicated()

# Count the number of duplicate rows
num_duplicate_rows = duplicate_rows.sum()

if num_duplicate_rows > 0:
    print("There are duplicate rows in the DataFrame.")
```

```

    print("Number of duplicate rows:", num_duplicate_rows)
else:
    print("There are no duplicate rows in the DataFrame.")

```

There are no duplicate rows in the DataFrame.

In [245...]

```

# Check for None values in "Value1" column
none_values_present = dfcopy2["Value1"].isna().any()

if none_values_present:
    print("Column 'Value1' contains None values.")
else:
    print("Column 'Value1' does not contain None values.")

# Check for duplicate values in "Value1" column
duplicate_values_present = dfcopy2["Value1"].nunique() != dfcopy2.shape[0]

if duplicate_values_present:
    print("Column 'Value1' contains duplicate values.")
else:
    print("Column 'Value1' does not contain duplicate values.")

```

Column 'Value1' does not contain None values.

Column 'Value1' does not contain duplicate values.

In [246...]

```
# Drop columns except the specified ones
dfcopy2 = dfcopy2[['Value1', "Sector_Lists", "Total_Organization_By_country", "Total_Sector_By_country"]]
```

In [247...]

```
dfcopy2.head()
```

Out[247]:

	Value1	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country
0	Aruba	[]	2	0
1	Afghanistan	[7, 18]	5	2
2	Angola	[]	5	0
3	Anguilla	[]	4	0
4	Åland Islands	[]	4	0

In [248...]

```

dfcopy2.rename(columns={"Value1": "Nations"}, inplace=True)
dfcopy2.head()

```

Out[248]:

	Nations	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country
0	Aruba	[]	2	0
1	Afghanistan	[7, 18]	5	2
2	Angola	[]	5	0
3	Anguilla	[]	4	0
4	Åland Islands	[]	4	0

In [249...]

```
# Define the mapping of values to be replaced
mapping = {
    "French Southern and Antarctic Territories": "French Southern and Antarctic Lands",
    "Bonaire Sint Eustatius and Saba": "Bonaire",
    "Bolivia (Plurinational State of)": "Bolivia",
    "Channel Islands": "Channel Islands",
    "Democratic Republic of the Congo": "Democratic Republic of Congo",
    "Micronesia (Federated States of)": " Federated States of Micronesia",
    "China Hong Kong Special Administrative Region": "Hong Kong",
    "Iran (Islamic Republic of)": "Iran",
    "Republic of Korea": "Republic of South Korea",
    "China Macao Special Administrative Region": "Macao",
    "Netherlands (Kingdom of the)": "Kingdom of the Netherlands",
    "State of Palestine": "Palestine",
    "Saint Helena Ascension and Tristan da Cunha": "Saint Helena",
    "Svalbard and Jan Mayen Islands": "Svalbard and Jan Mayen",
    "Sint Maarten": "Saint Martin",
    "Holy See": "Vatican",
    "Venezuela (Bolivarian Republic of)": "Bolivia",
    "Virgin Islands U.S.": "Virgin Islands",
    "Kosovo": "Kosovo"
}

# Replace values in the "Nations" column based on the mapping
dfcopy2["Nations"].replace(mapping, inplace=True)
# Print the DataFrame to verify changes
dfcopy2.head()
```

Out[249]:

	Nations	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country
0	Aruba	[]	2	0
1	Afghanistan	[7, 18]	5	2
2	Angola	[]	5	0
3	Anguilla	[]	4	0
4	Åland Islands	[]	4	0

In [249...]

In [250...]

```

import pycountry
import pandas as pd

def get_country_info(country_name):
    try:
        country_info = pycountry.countries.lookup(country_name)
        return {
            'name': country_info.name,
            'alpha_2': country_info.alpha_2,
            'alpha_3': country_info.alpha_3,
            # Add more attributes as needed
            'population': None # Placeholder since pycountry doesn't have population data
        }
    except LookupError:
        return {
            'name': country_name,
            'alpha_2': None,
            'alpha_3': None,
            'population': None,
            # Add more attributes as needed
        }

# Assuming dfcopy2 is already defined and contains a column 'Nations'
dfcopy2[['name', 'alpha_2', 'alpha_3', 'population']] = dfcopy2['Nations'].apply(get_country_info).apply(pd.Series)
# Handle missing population data
dfcopy2['population'].fillna(0, inplace=True)
# Display the result
dfcopy2.head()

```

Out[250]:

	Nations	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country		name	alpha_2	alpha_3	population
0	Aruba	[]	2	0	Aruba	AW	ABW	0	
1	Afghanistan	[7, 18]	5	2	Afghanistan	AF	AFG	0	
2	Angola	[]	5	0	Angola	AO	AGO	0	
3	Anguilla	[]	4	0	Anguilla	AI	AIA	0	
4	Åland Islands	[]	4	0	Åland Islands	AX	ALA	0	

In [250...]

In [251...]

```
# Define the mapping of Nations to their corresponding alpha_3 codes
mapping = {
    "French Southern and Antarctic Lands": "ATF",
    "Bonaire": "BES",
    "Democratic Republic of Congo": "COD",
    "Federated States of Micronesia": "FSM",
    "Republic of South Korea": "KOR",
    "Palestine": "PSE",
    "Saint Helena": "SHN",
    "Saint Martin": "SXM",
    "Vatican": "VAT",
    "Kosovo": "XKX",
    "Virgin Islands": "VIR"
}

# Function to update alpha_3 based on conditions
def update_alpha_3(row):
    nation = row['Nations']
    if row['alpha_3'] in [None, '', np.nan]:
        for key in mapping:
            if key in nation:
                return mapping[key]
    return row['alpha_3']

# Apply the function to the DataFrame
dfcopy2['alpha_3'] = dfcopy2.apply(update_alpha_3, axis=1)

# Print the DataFrame to verify changes
dfcopy2.head()
```

Out[251]:

	Nations	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country		name	alpha_2	alpha_3	population
0	Aruba	[]	2	0	Aruba	AW	ABW	0	
1	Afghanistan	[7, 18]	5	2	Afghanistan	AF	AFG	0	
2	Angola	[]	5	0	Angola	AO	AGO	0	
3	Anguilla	[]	4	0	Anguilla	AI	AIA	0	
4	Åland Islands	[]	4	0	Åland Islands	AX	ALA	0	

In [252...]

copytoconcat = dfcopy2.copy()

In [253...]

dfcopy2['Total\_Sector\_By\_country'].value\_counts()

Out[253]:

Total\_Sector\_By\_country

0	185
2	31
3	19
1	14

Name: count, dtype: int64

In [254...]

```
# Specify the columns to check for None values
columns_to_check = ['Nations', 'Sector_Lists', 'Total_Organization_By_country', 'Total_Sector_By_country', 'name', 'population']

# Check for None values in each column
for column in columns_to_check:
    if dfcopy2[column].isna().any():
        print(f"Column '{column}' contains None values.")
    else:
        print(f"Column '{column}' does not contain None values.)
```

Column 'Nations' does not contain None values.  
 Column 'Sector\_Lists' does not contain None values.  
 Column 'Total\_Organization\_By\_country' does not contain None values.  
 Column 'Total\_Sector\_By\_country' does not contain None values.  
 Column 'name' does not contain None values.  
 Column 'population' does not contain None values.  
 Column 'alpha\_2' contains None values.  
 Column 'alpha\_3' contains None values.

In [255...]

```
#####
dfcopy2badformat = dfcopy2[dfcopy2.isnull().any(axis=1)]
```

```
dfcopy2badformatoremove = dfcopy2badformat.copy()  
#####
```

In [256]:

```
# Find common rows based on all columns  
df_common = pd.merge(dfcopy2, dfcopy2badformat, on=list(dfcopy2.columns))  
# Drop the common rows from the original DataFrame  
dfcopy2_cleaned = dfcopy2[~dfcopy2.apply(tuple, 1).isin(df_common.apply(tuple, 1))]  
# Drop the 'alpha_2' and 'population' columns  
dfcopy2_cleaned.drop(columns=['alpha_2', 'population'], inplace=True)  
# Rename the 'alpha_3' column to 'Country'  
dfcopy2_cleaned.rename(columns={'name': 'Country'}, inplace=True)  
dfcopy2_cleaned.rename(columns={'alpha_3': 'iso_a3'}, inplace=True)  
# Display the cleaned DataFrame  
dfcopy2_cleaned.head()
```

```
<ipython-input-256-63f1e5b38357>:6: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
<ipython-input-256-63f1e5b38357>:8: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
<ipython-input-256-63f1e5b38357>:9: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Out[256]:

	Nations	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country	Country	iso_a3
0	Aruba	[]	2	0	Aruba	ABW
1	Afghanistan	[7, 18]	5	2	Afghanistan	AFG
2	Angola	[]	5	0	Angola	AGO
3	Anguilla	[]	4	0	Anguilla	AIA
4	Åland Islands	[]	4	0	Åland Islands	ALA

In [257...]

```
# Specify the columns to check for None values
columns_to_check = ['Nations', 'Sector_Lists', 'Total_Organization_By_country', 'Total_Sector_By_country']

# Check for None values in each column
for column in columns_to_check:
    if dfcopy2_cleaned[column].isna().any():
        print(f"Column '{column}' contains None values.")
    else:
        print(f"Column '{column}' does not contain None values.")

Column 'Nations' does not contain None values.
Column 'Sector_Lists' does not contain None values.
Column 'Total_Organization_By_country' does not contain None values.
Column 'Total_Sector_By_country' does not contain None values.
```

In [258...]

```
import geopandas as gpd
import pandas as pd
# Read the world countries GeoJSON file
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
# Create an empty DataFrame to store the country names and areas
area_data = []
# Iterate over the rows of the GeoDataFrame
for index, row in world.iterrows():
    country_name = row['name']
    area = row.geometry.area # Area in square units of the GeoDataFrame's coordinate reference system
    area_data.append({'Country': country_name, 'Area': area})
# Create a DataFrame from the list of dictionaries
area_df = pd.DataFrame(area_data)
# Print the DataFrame with country names and areas
area_df.head()
```

```
<ipython-input-258-a64f9852c9b8>:4: FutureWarning:
```

The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth\_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.

Out[258]:

	Country	Area
0	Fiji	1.639511
1	Tanzania	76.301964
2	W. Sahara	8.603984
3	Canada	1712.995228
4	United States of America	1122.281921

In [259...]

```
# https://www.geeksforgeeks.org/python-pandas-dataframe-append/
```

In [260...]

```
import geopandas as gpd
# Load the world dataset from geopandas
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
# Merge the world dataset with Geodfcopy DataFrame based on country names
merged_data1 = world.merge(dfcopy2_cleaned, how='left', left_on='name', right_on='Country')
# Print the DataFrame with merged data
merged_data1.head()
```

```
<ipython-input-260-b2854ce8e785>:3: FutureWarning:
```

The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth\_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.

Out[260]:

	pop_est	continent	name	iso_a3_x	gdp_md_est	geometry	Nations	Sector_Lists	Total_Organization_By_country	Total_Sector_I
0	889953.0	Oceania	Fiji	FJI	5496	MULTIPOLYGON ((180.00000 -16.06713, 180.00000...)	Fiji	[7]		5.0
1	58005463.0	Africa	Tanzania	TZA	63177	POLYGON ((33.90371 -0.95000, 34.07262 -1.05982...)	NaN	NaN		NaN
2	603253.0	Africa	W. Sahara	ESH	907	POLYGON ((-8.66559 27.65643, -8.66512 27.58948...)	NaN	NaN		NaN
3	37589262.0	North America	Canada	CAN	1736425	MULTIPOLYGON ((( -122.84000 49.00000, -122.9742...))	Canada	[]		5.0
4	328239523.0	North America	United States of America	USA	21433226	MULTIPOLYGON ((( -122.84000 49.00000, -120.0000...))	NaN	NaN		NaN

In [260...]

In [261...]

```
# prompt: value count Total_Organization_By_country
merged_data1['Total_Organization_By_country'].value_counts()
```

Out[261]:

```
Total_Organization_By_country
5.0    142
2.0     3
3.0     1
Name: count, dtype: int64
```

In [262...]

```
# https://www.capitalOne.com/tech/open-source/pandas-dataframe-concat/
```

In [262...]

In [263...]

```
import geopandas as gpd

# Load the world dataset
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))

# Filter the population information
population_data = world[['name', 'pop_est']]

# Merge population data with dfcopy2_cleaned based on country names
dfcopy2_cleaned = pd.merge(dfcopy2_cleaned, population_data, left_on='Country', right_on='name', how='left')

# Drop redundant columns and rename the population column
dfcopy2_cleaned.drop(columns=['name'], inplace=True)
dfcopy2_cleaned.rename(columns={'pop_est': 'Population'}, inplace=True)

# Display the resulting DataFrame
print("Updated DataFrame with new column 'Population':")
dfcopy2_cleaned.head()
```

Updated DataFrame with new column 'Population':

<ipython-input-263-59dbe9280ded>:4: FutureWarning:

The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth\_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.

Out[263]:

	Nations	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country	Country	iso_a3	Population
0	Aruba	[]		2		ABW	NaN
1	Afghanistan	[7, 18]		5		AFG	38041754.0
2	Angola	[]		5		AGO	31825295.0
3	Anguilla	[]		4		AIA	NaN
4	Åland Islands	[]		4		ALA	NaN

In [263...]

In [264...]

```
# Specify the columns to check for None values
columns_to_check = ['Nations', 'Sector_Lists', 'Total_Organization_By_country', 'Total_Sector_By_country', 'Country', 'iso_a3']

# Check for None values in each column
```

```

for column in columns_to_check:
    if dfcopy2_cleaned[column].isna().any():
        print(f"Column '{column}' contains None values.")
    else:
        print(f"Column '{column}' does not contain None values.")

```

Column 'Nations' does not contain None values.  
 Column 'Sector\_Lists' does not contain None values.  
 Column 'Total\_Organization\_By\_country' does not contain None values.  
 Column 'Total\_Sector\_By\_country' does not contain None values.  
 Column 'Country' does not contain None values.  
 Column 'iso\_a3' does not contain None values.  
 Column 'Population' contains None values.

In [264]:

In [265]:

```

# https://www.w3schools.com/python/pandas/ref_df_join.asp
# https://www.w3schools.com/python/pandas/ref_df_merge.asp

```

In [266]:

```

import pandas as pd
# Dropping the row with index 248
dfcopy2badformat = dfcopy2badformat.drop([248], errors='ignore')
# Resetting the index
dfcopy2badformat = dfcopy2badformat.reset_index(drop=True)
# Printing the modified DataFrame
dfcopy2badformat.head()

```

Out[266]:

	Nations	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country		name	alpha_2	alpha_3	population
0	French Southern and Antarctic Lands	[]		2	0	French Southern and Antarctic Lands	None	ATF	0
1	Bonaire	[]		2	0	Bonaire	None	BES	0
2	Channel Islands	[]		2	0	Channel Islands	None	None	0
3	Democratic Republic of Congo	[2, 7, 18]		5	3	Democratic Republic of Congo	None	COD	0
4	Federated States of Micronesia	[]		5	0	Federated States of Micronesia	None	FSM	0

In [267...]

`dfcopy2badformat.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12 entries, 0 to 11
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
---  --  
 0   Nations          12 non-null    object  
 1   Sector_Lists     12 non-null    object  
 2   Total_Organization_By_country 12 non-null    int64  
 3   Total_Sector_By_country      12 non-null    int64  
 4   name              12 non-null    object  
 5   alpha_2            0 non-null    object  
 6   alpha_3            11 non-null    object  
 7   population        12 non-null    int64  
dtypes: int64(3), object(5)
memory usage: 896.0+ bytes
```

In [268...]

```
# https://www.LearnDataSci.com/tutorials/geospatial-data-python-geopandas-shapely/
#https://datascience.quantecon.org/tools/maps.html
world_gdf = gpd.read_file(
    gpd.datasets.get_path('naturalearth_lowres')
)
world_gdf.head()
```

<ipython-input-268-41fc787cb852>:4: FutureWarning:

The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth\_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.

Out[268]:

	<b>pop_est</b>	<b>continent</b>	<b>name</b>	<b>iso_a3</b>	<b>gdp_md_est</b>	<b>geometry</b>
<b>0</b>	889953.0	Oceania	Fiji	FJI	5496	MULTIPOLYGON (((180.00000 -16.06713, 180.00000...
<b>1</b>	58005463.0	Africa	Tanzania	TZA	63177	POLYGON ((33.90371 -0.95000, 34.07262 -1.05982...
<b>2</b>	603253.0	Africa	W. Sahara	ESH	907	POLYGON ((-8.66559 27.65643, -8.66512 27.58948...
<b>3</b>	37589262.0	North America	Canada	CAN	1736425	MULTIPOLYGON (((-122.84000 49.00000, -122.9742...
<b>4</b>	328239523.0	North America	United States of America	USA	21433226	MULTIPOLYGON (((-122.84000 49.00000, -120.0000...

In [269...]

`WordGeoPandasBase = world_gdf.copy()`

In [270...]

```
world_gdf['pop_density'] = world_gdf.pop_est / world_gdf.area * 10**6
world_gdf1 = world_gdf.sort_values(by='pop_density', ascending=False)
world_gdf1.head()
```

<ipython-input-270-ec25e7ed7208>:1: UserWarning:

Geometry is in a geographic CRS. Results from 'area' are likely incorrect. Use 'GeoSeries.to\_crs()' to re-project geometries to a projected CRS before this operation.

Out[270]:

	pop_est	continent	name	iso_a3	gdp_md_est	geometry	pop_density
99	163046161.0	Asia	Bangladesh	BGD	302571	POLYGON ((92.67272 22.04124, 92.65226 21.32405...,	1.375084e+13
79	4685306.0	Asia	Palestine	PSE	16276	POLYGON ((35.39756 31.48909, 34.92741 31.35344...,	9.754682e+12
140	23568378.0	Asia	Taiwan	TWN	1127000	POLYGON ((121.77782 24.39427, 121.17563 22.790...,	7.744572e+12
77	6855713.0	Asia	Lebanon	LBN	51991	POLYGON ((35.82110 33.27743, 35.55280 33.26427...,	6.968534e+12
169	12626950.0	Africa	Rwanda	RWA	10354	POLYGON ((30.41910 -1.13466, 30.81613 -1.69891...,	6.647863e+12

In [270...]

## Web Data Collection

---



---

I used an online data collection method to correct country names that did not match the GeoPandas format and to fill in missing values in my dataset.

In [271...]

```
import pandas as pd

# Create the dataframe from the transformed table
data = {
    'pop_est': [300, 2409, 170, 170, 99010212, 115021, 51709098, 5051953, 4439, 38666, 825, 104, 104, 1810366],
    'continent': ['Antarctica', 'South America', 'Europe', 'Europe', 'Africa', 'Oceania', 'Asia', 'Asia', 'Africa', 'No',
    'name': [
        'French Southern and Antarctic Territories',
        'Caribbean Netherlands',
        'Jersey',
        'Guernsey',
```

```

'Democratic Republic of the Congo',
'Micronesia, Federated States of',
'Korea, Republic of',
'State of Palestine',
'Saint Helena, Ascension and Tristan da Cunha',
'Saint Martin (French part)',
'Holy See',
'British Virgin Islands',
'United States Virgin Islands',
'Kosovo'
],
'gdp_md_est': [31005, 553, 3446, 3446, 69474, 424, 1761000, 15562, 42, 561, 801, 4200, 4200, 7358],
'area_km2': [16978, 288, 350, 350, 905567, 271, 100210, 6020, 121, 87, 0.49, 346, 346, 10887]
}

dfbadformatAftercorrection = pd.DataFrame(data)
# Dropping the 3rd (index 2) and 11th (index 10) rows
dfbadformatAftercorrection = dfbadformatAftercorrection.drop([2, 11])
# Resetting the index
dfbadformatAftercorrection = dfbadformatAftercorrection.reset_index(drop=True)
dfbadformatAftercorrection = dfbadformatAftercorrection[['name', 'continent', 'area_km2', 'pop_est', 'gdp_md_est']]
dfbadformatAftercorrection.head()

```

Out[271]:

	name	continent	area_km2	pop_est	gdp_md_est
0	French Southern and Antarctic Territories	Antarctica	16978.0	300	31005
1	Caribbean Netherlands	South America	288.0	2409	553
2	Guernsey	Europe	350.0	170	3446
3	Democratic Republic of the Congo	Africa	905567.0	99010212	69474
4	Micronesia, Federated States of	Oceania	271.0	115021	424

In [272...]

```

# Select the specified columns from dfcopy2badformat
selected_columns = dfcopy2badformat[['Sector_Lists', 'Total_Organization_By_country', 'Total_Sector_By_country']]
# Concatenate the selected columns to the right of dfbadformatAftercorrection
dfbadformatAftercorrection = pd.concat([dfbadformatAftercorrection, selected_columns], axis=1)
# Printing the modified DataFrame
dfbadformatAftercorrection.head()

```

Out[272]:

	<b>name</b>	<b>continent</b>	<b>area_km2</b>	<b>pop_est</b>	<b>gdp_md_est</b>	<b>Sector_Lists</b>	<b>Total_Organization_By_country</b>	<b>Total_Sector_By_country</b>
<b>0</b>	French Southern and Antarctic Territories	Antarctica	16978.0	300	31005	[]	2	0
<b>1</b>	Caribbean Netherlands	South America	288.0	2409	553	[]	2	0
<b>2</b>	Guernsey	Europe	350.0	170	3446	[]	2	0
<b>3</b>	Democratic Republic of the Congo	Africa	905567.0	99010212	69474	[2, 7, 18]	5	3
<b>4</b>	Micronesia, Federated States of	Oceania	271.0	115021	424	[]	5	0

In [273...]

```
# Merge the DataFrames on 'name' and 'Country'
merged_df = pd.merge(merged_data1, area_df, left_on='name', right_on='Country')

# Select and reorder the columns
final_df = merged_df[['name', 'continent', 'Area', 'pop_est', 'gdp_md_est']]

# Print the final DataFrame
final_df.head()
```

Out[273]:

	<b>name</b>	<b>continent</b>	<b>Area</b>	<b>pop_est</b>	<b>gdp_md_est</b>
<b>0</b>	Fiji	Oceania	1.639511	889953.0	5496
<b>1</b>	Tanzania	Africa	76.301964	58005463.0	63177
<b>2</b>	W. Sahara	Africa	8.603984	603253.0	907
<b>3</b>	Canada	North America	1712.995228	37589262.0	1736425
<b>4</b>	United States of America	North America	1122.281921	328239523.0	21433226

In [274...]

```
# Let's say you want to select only the required columns from dfcopy2_cleaned
dfcopy2_selected = copytoconcat[['Sector_Lists', 'Total_Organization_By_country', 'Total_Sector_By_country', 'name']]
# Merge final_df with dfcopy2_selected based on the condition
merged_df = pd.merge(final_df, dfcopy2_selected, left_on='name', right_on='name', how='left')
```

In [275...]

`merged_df.head()`

Out[275]:

	<b>name</b>	<b>continent</b>	<b>Area</b>	<b>pop_est</b>	<b>gdp_md_est</b>	<b>Sector_Lists</b>	<b>Total_Organization_By_country</b>	<b>Total_Sector_By_country</b>
<b>0</b>	Fiji	Oceania	1.639511	889953.0	5496	[7]	5.0	1.0
<b>1</b>	Tanzania	Africa	76.301964	58005463.0	63177	NaN	NaN	NaN
<b>2</b>	W. Sahara	Africa	8.603984	603253.0	907	NaN	NaN	NaN
<b>3</b>	Canada	North America	1712.995228	37589262.0	1736425	[]	5.0	0.0
<b>4</b>	United States of America	North America	1122.281921	328239523.0	21433226	NaN	NaN	NaN

In [276...]

`#####
#####`

In [277...]

`merged_dftodrop = merged_df[merged_df.isnull().any(axis=1)]`

In [278...]

```
# Find indexes of rows to drop
indexes_to_drop = merged_df[merged_df['name'].isin(merged_dftodrop['name'])].index

# Drop rows from merged_df
Afterdfdropped = merged_df.drop(indexes_to_drop)
```

In [279...]

`Afterdfdropped.head()`

Out[279]:

	<b>name</b>	<b>continent</b>	<b>Area</b>	<b>pop_est</b>	<b>gdp_md_est</b>	<b>Sector_Lists</b>	<b>Total_Organization_By_country</b>	<b>Total_Sector_By_country</b>
<b>0</b>	Fiji	Oceania	1.639511	889953.0	5496	[7]	5.0	1.0
<b>3</b>	Canada	North America	1712.995228	37589262.0	1736425	[]	5.0	0.0
<b>5</b>	Kazakhstan	Asia	330.586822	18513930.0	181665	[]	5.0	0.0
<b>6</b>	Uzbekistan	Asia	49.958821	33580650.0	57921	[]	5.0	0.0
<b>7</b>	Papua New Guinea	Oceania	37.991091	8776109.0	24829	[7]	5.0	1.0

In [280...]

Afterdfdropped.info()

```
<class 'pandas.core.frame.DataFrame'>
Index: 148 entries, 0 to 175
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
---  --  
 0   name              148 non-null    object  
 1   continent         148 non-null    object  
 2   Area               148 non-null    float64 
 3   pop_est            148 non-null    float64 
 4   gdp_md_est         148 non-null    int64  
 5   Sector_Lists       148 non-null    object  
 6   Total_Organization_By_country 148 non-null    float64 
 7   Total_Sector_By_country     148 non-null    float64 
dtypes: float64(4), int64(1), object(3)
memory usage: 10.4+ KB
```

After collecting the missing information from the web, I have now created a new table to replace the dellete row of the Afterdfdropped table

In [281...]

```
# Data for the new DataFrame
data = {
    'index': [
        1, 2, 4, 11, 17, 18, 20, 23, 30, 66, 69, 73, 92, 94, 95, 96, 107, 108, 116, 124, 135, 149, 159, 160, 167, 170,
    ],
    'name': [
        'United Republic of Tanzania', 'Western Sahara', 'United States', 'Democratic Republic of Congo',
        'Dominican Republic', 'Russian Federation', 'Falkland Islands (Malvinas)', 'French Southern and Antarctic Lands',
        'Bolivia', 'Central African Republic', 'Equatorial Guinea', 'Eswatini',
        "Lao People's Democratic Republic", 'Viet Nam', "Democratic People's Republic of Korea",
        'Republic of South Korea', 'Iran', 'Syrian Arab Republic', 'Moldova', 'Republic of Türkiye',
        'Solomon Islands', 'Brunei', 'Antarctica', 'Cyprus', 'Somalia', 'Bosnia and Herzegovina', 'South Sudan'
    ],
    'continent': [
        'Africa', 'Africa', 'North America', 'Africa', 'North America', 'Europe', 'South America', 'Oceania',
        'South America', 'Africa', 'Africa', 'Africa', 'Asia', 'Asia', 'Asia', 'Asia', 'Asia', 'Europe',
        'Asia', 'Oceania', 'Asia', 'Antarctica', 'Asia', 'Africa', 'Europe', 'Africa'
    ],
    'Area': [
        76.30196359087157, 8.603984207472143, 1122.2819207780806, 189.5152324776122, 4.129206498675565,
        2931.8319455265946, 2.1287499999999984, 1.4329281249999886, 92.07717331220415, 50.861075547061496,
        2.204018821470195, 1.6399831040728134, 19.60219942489319, 28.552628470049843, 13.277813493481911,
        9.954819746830225, 155.53114761158264, 18.293965039175614, 3.837657629321802, 83.15462647945547,
```

```
2.0326265689987473, 0.8720531008584439, 6028.836194274539, 0.3746440631902616, 13.800261642538429,
5.696665861833908, 51.196105909569184
],
'pop_est': [
    58005463.0, 603253.0, 328239523.0, 86790567.0, 10738958.0, 144373535.0, 3398.0, 140.0, 11513100.0,
    4745185.0, 1355986.0, 1148130.0, 7169455.0, 96462106.0, 25666161.0, 51709098.0, 82913906.0, 17070135.0,
    2657637.0, 83429615.0, 669823.0, 433285.0, 4490.0, 326000.0, 5096159.0, 3301000.0, 11062113.0
],
'gdp_md_est': [
    63177, 907, 21433226, 50400, 88941, 1699876, 282, 16, 40895, 2220, 11026, 4471, 18173, 261921, 40000,
    1646739, 453996, 98830, 11968, 761425, 1589, 13469, 898, 3600, 17836, 20164, 11998
],
'Sector_Lists': [
    [2, 7, 18], [], [], [2, 7, 18], [], [], [], [7, 18], [], [7], [7, 18], [7], [], [7, 18], [7, 18]
    [7], [], [], [7, 18], [], [7, 18]
],
'Total_Organization_By_country': [
    5, 2, 5, 5, 5, 5, 2, 2, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5
],
'Total_Sector_By_country': [
    3, 0, 0, 3, 0, 0, 0, 0, 0, 2, 0, 0, 1, 2, 1, 0, 0, 2, 0, 0, 1, 0, 0, 0, 2, 0, 2
]
}
}

# Create the DataFrame
new_df = pd.DataFrame(data)

# Set the index
new_df.set_index('index', inplace=True)

# Display the new DataFrame
new_df.head()
```

Out[281]:

	name	continent	Area	pop_est	gdp_md_est	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country
index								
1	United Republic of Tanzania	Africa	76.301964	58005463.0	63177	[2, 7, 18]	5	3
2	Western Sahara	Africa	8.603984	603253.0	907	[]	2	0
4	United States	North America	1122.281921	328239523.0	21433226	[]	5	0
11	Democratic Republic of Congo	Africa	189.515232	86790567.0	50400	[2, 7, 18]	5	3
17	Dominican Republic	North America	4.129206	10738958.0	88941	[]	5	0

In [282...]: new\_df['Total\_Sector\_By\_country'].value\_counts()

Out[282]:

Total_Sector_By_country	count
0	17
2	5
1	3
3	2

Name: count, dtype: int64

In [282...]:

Append DataFrames Afterdfdropped and new\_df DataFrames

In [283...]:

```
import pandas as pd
appended_df = pd.concat([Afterdfdropped, new_df], axis=0)
# Reorder the DataFrame by index in ascending order
appended_df = appended_df.sort_index()
```

In [284...]: appended\_df.head()

Out[284]:

	name	continent	Area	pop_est	gdp_md_est	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country
0	Fiji	Oceania	1.639511	889953.0	5496	[7]	5.0	1.0
1	United Republic of Tanzania	Africa	76.301964	58005463.0	63177	[2, 7, 18]	5.0	3.0
2	Western Sahara	Africa	8.603984	603253.0	907	[]	2.0	0.0
3	Canada	North America	1712.995228	37589262.0	1736425	[]	5.0	0.0
4	United States	North America	1122.281921	328239523.0	21433226	[]	5.0	0.0

In [285...]

```
# # Perform the merge operation
appended_df = appended_df.merge(world_gdf1[['name', 'pop_density']], on='name', how='left')
#appended_df = appended_df.drop(columns=['pop_density_y'])
#appended_df = appended_df.drop(columns=['pop_density'])
#appended_df = appended_df.drop(columns=['pop_density_y'])
appended_df.head()
```

Out[285]:

	name	continent	Area	pop_est	gdp_md_est	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country	pop_dens
0	Fiji	Oceania	1.639511	889953.0	5496	[7]	5.0	1.0	5.428161e+
1	United Republic of Tanzania	Africa	76.301964	58005463.0	63177	[2, 7, 18]	5.0	3.0	N
2	Western Sahara	Africa	8.603984	603253.0	907	[]	2.0	0.0	N
3	Canada	North America	1712.995228	37589262.0	1736425	[]	5.0	0.0	2.194359e+
4	United States	North America	1122.281921	328239523.0	21433226	[]	5.0	0.0	N

In [286...]

```
import pandas as pd
import numpy as np

# Provided data
data = {
    'index': [2, 93, 4, 1, 107, 174, 134, 18, 123, 95, 91, 23, 20, 72, 68, 17, 11, 94, 65, 168],
    'Area': [8.603984207472143, 28.552628470049843, 1122.2819207780806, 76.30196359087157,
              18.293965039175614, 51.196105909569184, 2.0326265689987473, 2931.8319455265946,
              83.15462647945547, 9.954819746830225, 19.60219942489319, 1.4329281249999886,
              2.1287499999999984, 1.6399831040728134, 2.204018821470195, 4.129206498675565,
              189.5152324776122, 13.277813493481911, 50.861075547061496, 5.696665861833908],
    'pop_est': [603253.0, 96462106.0, 328239523.0, 58005463.0, 17070135.0, 11062113.0, 669823.0,
                144373535.0, 83429615.0, 51709098.0, 7169455.0, 140.0, 3398.0, 1148130.0, 1355986.0,
                10738958.0, 86790567.0, 25666161.0, 4745185.0, 3301000.0],
    'pop_density_x': [np.nan] * 20
}

# Creating the DataFrame
dfnewdensite = pd.DataFrame(data)

# Calculate population density where pop_density_x is NaN
dfnewdensite['pop_density_x'] = dfnewdensite.apply(
    lambda row: row['pop_est'] / row['Area'] if np.isnan(row['pop_density_x']) else row['pop_density_x'],
    axis=1
)

# Rename the 'index' column to 'Number'
#dfnewdensite.rename(columns={'index': 'Number'}, inplace=True)

# Display the updated DataFrame with formatted values
dfnewdensite.set_index('index', inplace=True)
dfnewdensite['Area'] = dfnewdensite['Area'].map('{:.6f}'.format)
dfnewdensite['pop_density_x'] = dfnewdensite['pop_density_x'].map('{:.6e}'.format)
dfnewdensite.head()
```

Out[286]:

Area	pop_est	pop_density_x
------	---------	---------------

index	Area	pop_est	pop_density_x
2	8.603984	603253.0	7.011322e+04
93	28.552628	96462106.0	3.378397e+06
4	1122.281921	328239523.0	2.924751e+05
1	76.301964	58005463.0	7.602093e+05
107	18.293965	17070135.0	9.331020e+05

In [286...]

In [287...]

```
NANCOLUMN = appended_df[appended_df.isnull().any(axis=1)]
```

In [288...]

```
# Merge the two DataFrames on the 'index' column
appended_df2= appended_df.merge(dfornewdensity[['pop_est', 'pop_density_x']], on='pop_est', how='left', suffixes=('', '_'))
# Fill NaN values in 'pop_density_x' with values from 'pop_density_x_new'
appended_df2['pop_density_x'] = appended_df2['pop_density_x'].combine_first(appended_df2['pop_density_x_new'])
# Drop the 'pop_density_x_new' column as it is no longer needed
appended_df2 = appended_df2.drop(columns=['pop_density_x_new'])
# Display the result
appended_df2.head()
```

Out[288]:

	name	continent	Area	pop_est	gdp_md_est	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country	pop_dens
0	Fiji	Oceania	1.639511	889953.0	5496	[7]	5.0	1.0	5.428161e+
1	United Republic of Tanzania	Africa	76.301964	58005463.0	63177	[2, 7, 18]	5.0	3.0	N
2	Western Sahara	Africa	8.603984	603253.0	907	[]	2.0	0.0	N
3	Canada	North America	1712.995228	37589262.0	1736425	[]	5.0	0.0	2.194359e+
4	United States	North America	1122.281921	328239523.0	21433226	[]	5.0	0.0	N

In [289...]: dfornewdensite.head(1)

Out[289]:

	Area	pop_est	pop_density_x
index			
2	8.603984	603253.0	7.011322e+04

In [289...]:

```
# Merge DataFrames based on 'pop_est'
merged_df = pd.merge(appended_df2, dfornewdensite[['pop_est', 'pop_density_x']], on='pop_est', how='left')

# Fill NaN values in 'pop_density' column with values from 'pop_density_x'
merged_df['pop_density'] = merged_df.apply(lambda row: row['pop_density_x'] if pd.isna(row['pop_density']) else row['pop_density'])

# Drop the auxiliary column
merged_df.drop(columns=['pop_density_x'], inplace=True)

# Check if 'index' column exists before setting it as the index
if 'index' in merged_df.columns:
    merged_df.set_index('index', inplace=True)
else:
```

```
print("The 'index' column does not exist in the DataFrame.")
merged_df.head()
```

The 'index' column does not exist in the DataFrame.

Out[290]:

	<b>name</b>	<b>continent</b>	<b>Area</b>	<b>pop_est</b>	<b>gdp_md_est</b>	<b>Sector_Lists</b>	<b>Total_Organization_By_country</b>	<b>Total_Sector_By_country</b>	<b>pop_dens</b>
<b>0</b>	Fiji	Oceania	1.639511	889953.0	5496	[7]		5.0	1.0 5.428161e+
<b>1</b>	United Republic of Tanzania	Africa	76.301964	58005463.0	63177	[2, 7, 18]		5.0	3.0 7.602093e+
<b>2</b>	Western Sahara	Africa	8.603984	603253.0	907	[]		2.0	0.0 7.011322e+
<b>3</b>	Canada	North America	1712.995228	37589262.0	1736425	[]		5.0	0.0 2.194359e+
<b>4</b>	United States	North America	1122.281921	328239523.0	21433226	[]		5.0	0.0 2.924751e+

In [291...]

```
# Define GDP thresholds for categorization
poor_threshold = merged_df['gdp_md_est'].quantile(0.25)
rich_threshold = merged_df['gdp_md_est'].quantile(0.75)
# Categorize countries based on GDP
merged_df['GDP_Category'] = pd.cut(merged_df['gdp_md_est'], bins=[0, poor_threshold, rich_threshold, float('inf')], labels=['Poor', 'Rich'])
# Display the DataFrame with GDP categories
merged_df[['gdp_md_est', 'GDP_Category']]
# Create new column 'Priority'
merged_df['Priority'] = merged_df['Total_Sector_By_country'].apply(lambda x: 'No' if x == 0 else 'Yes')
# Create new column 'Organization'
merged_df['Organization'] = merged_df['Total_Organization_By_country'].apply(lambda x: 'No' if x == 0 else 'Yes')
merged_df.head()
```

Out[291]:

	name	continent	Area	pop_est	gdp_md_est	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country	pop_dens
0	Fiji	Oceania	1.639511	889953.0	5496	[7]	5.0	1.0	5.428161e+
1	United Republic of Tanzania	Africa	76.301964	58005463.0	63177	[2, 7, 18]	5.0	3.0	7.602093e+
2	Western Sahara	Africa	8.603984	603253.0	907	[]	2.0	0.0	7.011322e+
3	Canada	North America	1712.995228	37589262.0	1736425	[]	5.0	0.0	2.194359e+
4	United States	North America	1122.281921	328239523.0	21433226	[]	5.0	0.0	2.924751e+

In [292...]

```
# Create the data for the DataFrame
data = {
    "index": [100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120,
    "name": ["Nepal", "Pakistan", "Afghanistan", "Tajikistan", "Kyrgyzstan", "Turkmenistan", "Iran", "Syrian Arab Repub
    "Armed_Conflict": ["No", "Yes", "Yes", "No", "No", "No", "Yes", "Yes", "No", "No", "Yes", "No", "No", "No", "No", ""
}
# Create the DataFrame
dfConflict1 = pd.DataFrame(data)
# Display the DataFrame
dfConflict1.head()
```

Out[292]:

	index	name	Armed_Conflict
0	100	Nepal	No
1	101	Pakistan	Yes
2	102	Afghanistan	Yes
3	103	Tajikistan	No
4	104	Kyrgyzstan	No

In [292...]

In [293...]

```

import pandas as pd

# Create the data
data = {
    "name": ["Fiji", "United Republic of Tanzania", "Western Sahara", "Canada", "United States", "Kazakhstan", "Uzbekis",
    "armed_conflict_list" : ["Yes", "No", "Yes", "No", "Yes", "No", "No", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes",
    "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "No", "Yes", "No", "Yes", "No", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes",
    "No", "No", "No", "No"]
}
}

# Get the number of elements in each list
num_name_elements = len(data["name"])
num_conflict_elements = len(data["armed_conflict_list"])

# Print the number of elements in each list
print("Number of elements in 'name' list:", num_name_elements)
print("Number of elements in 'armed_conflict_list' list:", num_conflict_elements)

```

Number of elements in 'name' list: 99  
 Number of elements in 'armed\_conflict\_list' list: 99

In [294...]

```

import pandas as pd
data = {
    "name": ["Fiji", "United Republic of Tanzania", "Western Sahara", "Canada", "United States", "Kazakhstan", "Uzbekis",
    "Armed_Conflict": ["Yes", "No", "Yes", "No", "No", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes",
    "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "No", "Yes", "No", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes",
    "No", "No", "No", "No"]
}
dfConflict2 = pd.DataFrame(data)
dfConflict2.head()

```

Out[294]:

	name	Armed_Conflict
0	Fiji	Yes
1	United Republic of Tanzania	No
2	Western Sahara	Yes
3	Canada	No
4	United States	No

In [295...]

```
# Concatenate the dataframes along rows
# Select the desired columns from each dataframe
dfConflict1_selected = dfConflict1[["name", "Armed_Conflict"]]
dfConflict2_selected = dfConflict2[["name", "Armed_Conflict"]]

# Concatenate the selected dataframes along rows
Conflictcombined_df = pd.concat([dfConflict1_selected, dfConflict2_selected], ignore_index=True)
# Display the combined dataframe
Conflictcombined_df.head()
```

Out[295]:

	name	Armed_Conflict
0	Nepal	No
1	Pakistan	Yes
2	Afghanistan	Yes
3	Tajikistan	No
4	Kyrgyzstan	No

In [296...]

```
Conflictcombined_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 174 entries, 0 to 173
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   name            174 non-null    object 
 1   Armed_Conflict  174 non-null    object 
dtypes: object(2)
memory usage: 2.8+ KB
```

In [297...]

```
merged_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 175 entries, 0 to 174
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   name             175 non-null    object  
 1   continent        175 non-null    object  
 2   Area              175 non-null    float64 
 3   pop_est           175 non-null    float64 
 4   gdp_md_est        175 non-null    int64  
 5   Sector_Lists      175 non-null    object  
 6   Total_Organization_By_country 175 non-null    float64 
 7   Total_Sector_By_country     175 non-null    float64 
 8   pop_density        175 non-null    float64 
 9   GDP_Category       175 non-null    category 
 10  Priority           175 non-null    object  
 11  Organization       175 non-null    object  
dtypes: category(1), float64(5), int64(1), object(5)
memory usage: 15.5+ KB
```

In [298...]

```
# Merge the "Armed_Conflict" column to the left of merged_df based on the "name" column
merged_df = pd.merge(merged_df, Conflitcombined_df[["name", "Armed_Conflict"]], on="name", how="left")
```

In [299...]

```
merged_df.head()
```

Out[299]:

	name	continent	Area	pop_est	gdp_md_est	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country	pop_dens
0	Fiji	Oceania	1.639511	889953.0	5496	[7]		5.0	1.0 5.428161e+
1	United Republic of Tanzania	Africa	76.301964	58005463.0	63177	[2, 7, 18]		5.0	3.0 7.602093e+
2	Western Sahara	Africa	8.603984	603253.0	907	[]		2.0	0.0 7.011322e+
3	Canada	North America	1712.995228	37589262.0	1736425	[]		5.0	0.0 2.194359e+
4	United States	North America	1122.281921	328239523.0	21433226	[]		5.0	0.0 2.924751e+

In [300...]

```
#https://geneva-academy.ch/galleries/today-s-armed-conflicts
#https://fr.statista.com/infographie/25635/pays-affrontements-armes-guerres-en-cours/
```

In [300...]

In [301...]

```
# Define a function to convert Sector_Lists values
def convert_sector_lists(value):
    if isinstance(value, list):
        return value
    elif value == '':
        return []
    elif value == '2,7,18':
        return [2, 7, 18]
    elif value == '7,18':
        return [7, 18]
    elif value == '7':
        return [7]
    else:
        # Remove square brackets and split the string by commas
        value = value.strip('[]').strip()
        if not value:
            return []
        # Enclose each element in brackets and return as a list
        return [int(x) for x in value.split(',')]

# Apply the function to the Sector_Lists column
merged_df['Sector_Lists'] = merged_df['Sector_Lists'].apply(convert_sector_lists)

# Check the result
print(merged_df[['Sector_Lists']])
```

```
Sector_Lists
0      [7]
1      [2, 7, 18]
2      []
3      []
4      []
..
172     ...
173     []
174     []
175     []
176     [7, 18]
```

[177 rows x 1 columns]

In [302...]

```
# Get unique lists from the 'Sector_Lists' column
unique_lists = set(tuple(lst) for lst in merged_df['Sector_Lists'])

# Iterate over unique lists and print them
for lst in unique_lists:
    print(lst)
```

```
(2, 7)
(2, 7, 18)
(7,)
()
(7, 18)
```

In [303...]

```
merged_df.head()
```

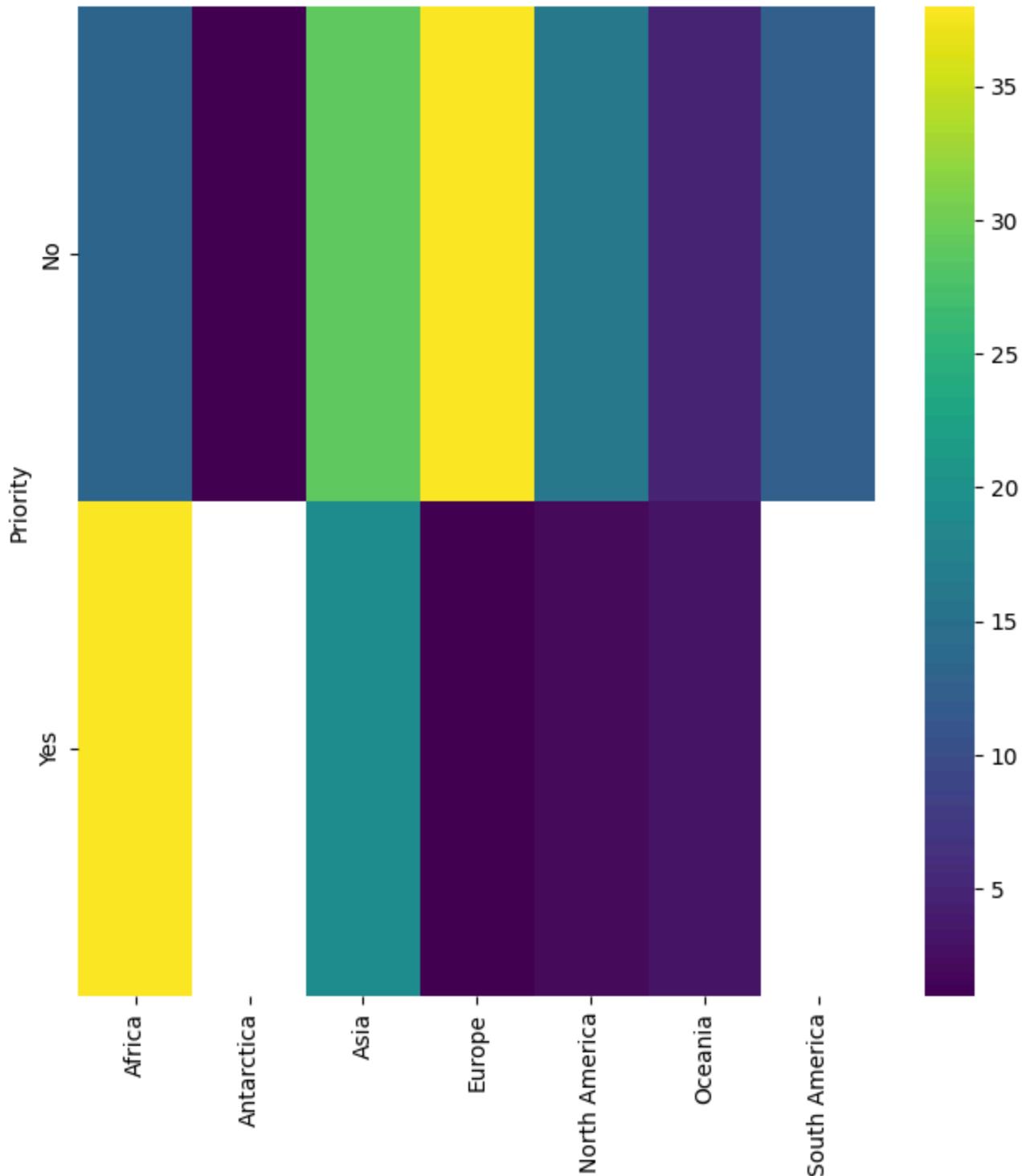
Out[303]:

	name	continent	Area	pop_est	gdp_md_est	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country	pop_dens
0	Fiji	Oceania	1.639511	889953.0	5496	[7]	5.0	1.0	5.428161e+
1	United Republic of Tanzania	Africa	76.301964	58005463.0	63177	[2, 7, 18]	5.0	3.0	7.602093e+
2	Western Sahara	Africa	8.603984	603253.0	907	[]	2.0	0.0	7.011322e+
3	Canada	North America	1712.995228	37589262.0	1736425	[]	5.0	0.0	2.194359e+
4	United States	North America	1122.281921	328239523.0	21433226	[]	5.0	0.0	2.924751e+

In [304...]: DataCleanAndPrepareForML = merged\_df.copy()

In [304...]:

```
# continent vs Priority
from matplotlib import pyplot as plt
import seaborn as sns
import pandas as pd
plt.subplots(figsize=(8, 8))
df_2dhist = pd.DataFrame({
    x_label: grp['Priority'].value_counts()
    for x_label, grp in merged_df.groupby('continent')
})
sns.heatmap(df_2dhist, cmap='viridis')
plt.xlabel('continent')
_ = plt.ylabel('Priority')
```



## continent

**Summary:** High Priority: Africa (bright yellow). No Priority or Low Priority: Antarctica, Europe, Oceania, South America (all represented mostly by dark purple). Based on this interpretation, Africa has a high priority for humanitarian aid, whereas Antarctica, Europe, Oceania, and South America have no or low priority. Asia and North America have mixed levels of priority.

In [306...]

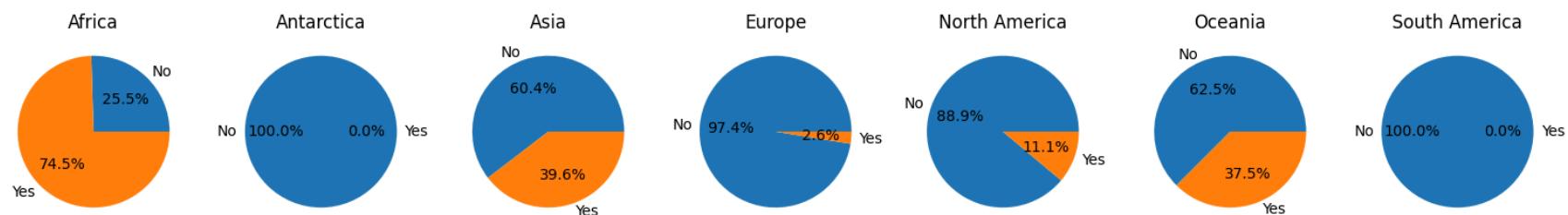
```
import matplotlib.pyplot as plt

# Group data by continent and Priority
grouped_df = merged_df.groupby(['continent', 'Priority']).size().unstack().fillna(0)

# Create a pie chart for each continent
fig, axes = plt.subplots(nrows=1, ncols=len(grouped_df.index), figsize=(15, 5))

for i, continent in enumerate(grouped_df.index):
    ax = axes[i]
    ax.pie(grouped_df.loc[continent], labels=grouped_df.columns, autopct="%1.1f%%")
    ax.set_title(continent)

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



**Summary:** High Priority: Africa (bright yellow). No Priority or Low Priority: Antarctica, Europe, Oceania, South America (all represented mostly by dark purple). Based on this interpretation, Africa has a high priority for humanitarian aid, whereas Antarctica, Europe, Oceania, and South America have no or low priority. Asia and North America have mixed levels of priority.

In [307...]

```
# Count occurrences of each value in the 'Priority' column for countries categorized as "Poor"
poor_priority_counts = merged_df['GDP_Category'].value_counts()

poor_priority_counts
```

```
Out[307]: GDP_Category
Middle-income    88
Poor            45
Rich             44
Name: count, dtype: int64
```

```
In [308... import matplotlib.pyplot as plt

# Group data by continent and GDP_Category
grouped_df = merged_df.groupby(['continent', 'GDP_Category']).size().unstack().fillna(0)

# Calculate the number of rows needed
n_continents = len(grouped_df.index)
nrows = (n_continents // 4) + (n_continents % 4 > 0)

# Create a pie chart for each continent
fig, axes = plt.subplots(nrows=nrows, ncols=4, figsize=(35, 20)) # Adjust figsize for better readability

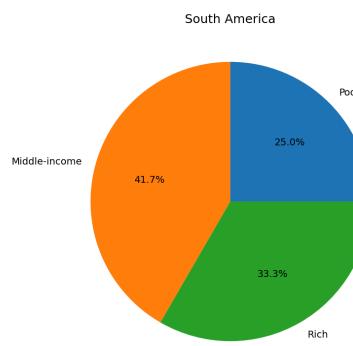
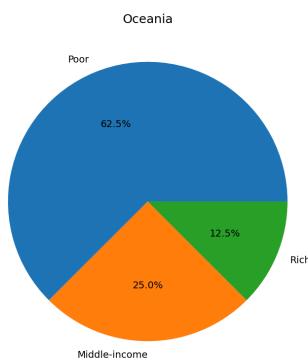
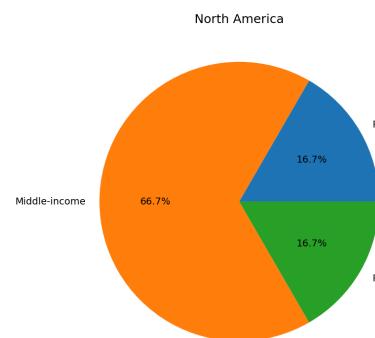
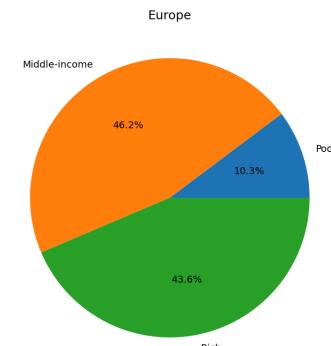
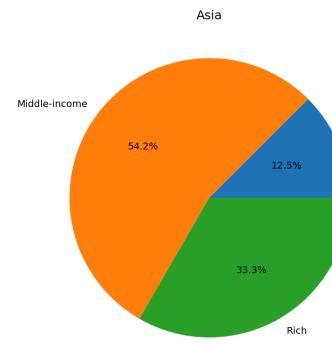
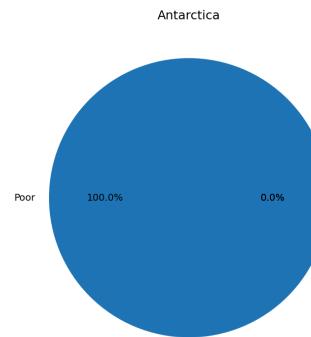
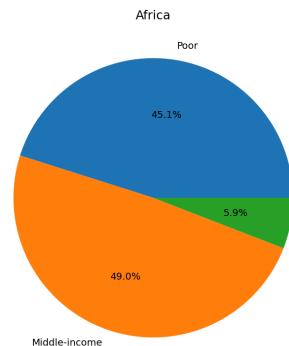
# Flatten the axes array for easier indexing
axes = axes.flatten()

for i, continent in enumerate(grouped_df.index):
    ax = axes[i]
    wedges, texts, autotexts = ax.pie(grouped_df.loc[continent], labels=grouped_df.columns, autopct="%1.1f%%", textprops={'color': 'white'})
    ax.set_title(continent, fontsize=18)

# Hide any unused subplots
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

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

## FromAPI\_SQLDatabase\_VizPowerBI



**Summary:** Antarctica and Oceania stand out with the highest percentage of poor countries, while Europe and Asia have the lowest. Africa has a significant portion of poor countries but also has a substantial middle-income population.

In [309...]

```
#####
#####
```

In [310...]

```
#####
#####
```

In [311...]

```
# Define a function to parse the string and convert it to a list
def parse_list_str(s):
    # Check if the string is empty or contains only whitespace
    if not s.strip('[]'):
        return []
    else:
        # Remove brackets and split by comma, then strip whitespace from each element
        return [x.strip() for x in s.strip('[]').split(',') if x.strip()]
```

```
# Apply the custom function to convert the strings in 'Regions_Lists' column to lists
dfcopy['Regions_Lists'] = df['Regions_Lists'].apply(parse_list_str)
dfcopy.head(1)
```

Out[311]:

	countryId	iso3	m49	isCurrent	isSOWC	isCountDown	isPublished	cndcountryCode	Regions_Lists	Language1	...	Organization3	Lang
0	1	ABW	533	1	0	0	1	ABW	["DEVPING", "SIDS", "AMERICAS", "CARIBB", "GLO..."]	English	...	None	

1 rows × 27 columns

In [312...]

```
import geopandas as gpd
import pandas as pd

# Load the country boundaries dataset
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))

# Join the datasets based on country codes
merged_data = pd.merge(dfcopy, world, how='left', left_on='cndcountryCode', right_on='iso_a3')

# Fix the population column (replace NaN values with zeros)
merged_data['Population'] = merged_data['pop_est'].fillna(0)

# Check the column names in the merged DataFrame
print(merged_data.columns)

# Append the fixed population column to dfcopy
dfcopy['Population'] = merged_data['Population']

# Print out the merged DataFrame
#print(dfcopy[['cndcountryCode', 'Population']])
Geodfcopy = dfcopy[['iso3', 'cndcountryCode', 'Population']]
```

```
Index(['countryId', 'iso3', 'm49', 'isCurrent', 'isSOWC', 'isCountDown',
       'isPublished', 'cndcountryCode', 'Regions_Lists', 'Language1', 'Value1',
       'Organization1', 'Language2', 'Value2', 'Organization2', 'Language3',
       'Value3', 'Organization3', 'Language4', 'Value4', 'Organization4',
       'Language5', 'Value5', 'Organization5', 'Sector_Lists',
       'Total_Organization_By_country', 'Total_Sector_By_country', 'pop_est',
       'continent', 'name', 'iso_a3', 'gdp_md_est', 'geometry', 'Population'],
      dtype='object')
```

```
<ipython-input-312-9d35cdb59372>:5: FutureWarning:
```

The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth\_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.

In [313...]

```
Geodfcopy.head(1)
```

Out[313]:

	iso3	cndcountryCode	Population
0	ABW	ABW	0.0

In [314...]

```
#import iso3
import pycountry

# Define the function to get country name based on ISO3 code
def get_country_name(iso3_code):
    """Returns the name of the country corresponding to the given ISO3 code."""
    try:
        country = pycountry.countries.get(alpha_3=iso3_code)
        return country.name
    except AttributeError:
        return None

# Create a new column "Countries" and populate it with country names
Geodfcopy['Countries'] = Geodfcopy['iso3'].apply(get_country_name)

# Print the DataFrame to verify the new column
Geodfcopy.head(1)
```

```
<ipython-input-314-7140c906c2e4>:14: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Out[314]:

	iso3	cndcountryCode	Population	Countries
0	ABW	ABW	0.0	Aruba

In [315...]

```
from Countrydetails import country
import geopandas as gpd

# Load the world dataset
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))

# Merge the world dataset with your Geodfcopy DataFrame based on ISO3 code
merged_data = world.merge(Geodfcopy, how='right', left_on='iso_a3', right_on='iso3')

# Calculate the area for each country in square kilometers
#merged_data['area'] = merged_data.geometry.area / 10**6 # Convert to square kilometers
merged_data['area'] = merged_data.geometry.area
# Add the 'area' column to Geodfcopy
Geodfcopy['area'] = merged_data['area']

# Print the updated DataFrame
Geodfcopy.head(1)
```

```
<ipython-input-315-ca095e2ffb51>:5: FutureWarning:
```

The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth\_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.

```
<ipython-input-315-ca095e2ffb51>:12: UserWarning:
```

Geometry is in a geographic CRS. Results from 'area' are likely incorrect. Use 'GeoSeries.to\_crs()' to re-project geometries to a projected CRS before this operation.

```
<ipython-input-315-ca095e2ffb51>:14: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
Out[315]:    iso3  cndcountryCode  Population  Countries  area
0   ABW          ABW        0.0      Aruba    NaN
```

```
In [316...  from pycountry_convert import country_alpha2_to_continent_code, country_name_to_country_alpha2

# Define a function to get the continent for a country
def get_country_continent(country_name):
    try:
        country_alpha2 = country_name_to_country_alpha2(country_name)
        continent_code = country_alpha2_to_continent_code(country_alpha2)
        return continent_code if continent_code is not None else None
    except Exception as e:
        print(f"Error retrieving continent for {country_name}: {e}")
        return None

# Add a new column for continent to Geodfcopy DataFrame
Geodfcopy['Continent'] = Geodfcopy['Countries'].apply(get_country_continent)

# Print the updated DataFrame with the continent column
Geodfcopy[['Continent', 'iso3', 'cndcountryCode', 'Countries', 'area', 'Population']]
Geodfcopy.head(1)
```

```
<ipython-input-316-a02649d33815>:14: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
Error retrieving continent for French Southern Territories: "Invalid Country Alpha-2 code: 'TF'"  
Error retrieving continent for None: "Invalid Country Name: 'None'"  
Error retrieving continent for Western Sahara: "Invalid Country Alpha-2 code: 'EH'"  
Error retrieving continent for Pitcairn: "Invalid Country Alpha-2 code: 'PN'"  
Error retrieving continent for Sint Maarten (Dutch part): "Invalid Country Alpha-2 code: 'SX'"  
Error retrieving continent for Timor-Leste: "Invalid Country Alpha-2 code: 'TL'"  
Error retrieving continent for United States Minor Outlying Islands: "Invalid Country Alpha-2 code: 'UM'"  
Error retrieving continent for Holy See (Vatican City State): "Invalid Country Alpha-2 code: 'VA'"  
Error retrieving continent for None: "Invalid Country Name: 'None'"
```

Out[316]:

	iso3	cndcountryCode	Population	Countries	area	Continent
0	ABW	ABW	0.0	Aruba	NaN	NA

In [317...]

## Geospatial Data Processing GIS:

### Collecting and Managing Geospatial Data for Creating Geomaps Using GeoPandas

In [318...]

```
import geopandas as gpd  
  
# Load the world dataset from geopandas  
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))  
  
# Merge the world dataset with Geodfcopy DataFrame based on country names  
merged_data = world.merge(Geodfcopy, how='left', left_on='name', right_on='Countries')  
  
# Print the DataFrame with merged data  
print(merged_data)
```

```
<ipython-input-318-cf06567191ff>:4: FutureWarning:
```

```
The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from https://www.naturalearthdata.com/downloads/110m-cultural-vectors/.
```

	pop_est	continent		name	iso_a3	gdp_md_est	\
0	889953.0	Oceania		Fiji	FJI	5496	
1	58005463.0	Africa		Tanzania	TZA	63177	
2	603253.0	Africa		W. Sahara	ESH	907	
3	37589262.0	North America		Canada	CAN	1736425	
4	328239523.0	North America	United States of America	USA	USA	21433226	
..	...	...		...	...	...	
172	6944975.0	Europe		Serbia	SRB	51475	
173	622137.0	Europe		Montenegro	MNE	5542	
174	1794248.0	Europe		Kosovo	-99	7926	
175	1394973.0	North America	Trinidad and Tobago	TTO	TTO	24269	
176	11062113.0	Africa		S. Sudan	SSD	11998	
				geometry	iso3	cndcountryCode	\
0	MULTIPOLYGON (((180.00000 -16.06713, 180.00000... FJI				FJI		
1	POLYGON ((33.90371 -0.95000, 34.07262 -1.05982... NaN				NaN		
2	POLYGON ((-8.66559 27.65643, -8.66512 27.58948... NaN				NaN		
3	MULTIPOLYGON (((-122.84000 49.00000, -122.9742... CAN				CAN		
4	MULTIPOLYGON (((-122.84000 49.00000, -120.0000... NaN				NaN		
..	...	...		...	...	...	
172	POLYGON ((18.82982 45.90887, 18.82984 45.90888... SRB				SRB		
173	POLYGON ((20.07070 42.58863, 19.80161 42.50009... MNE				MNE		
174	POLYGON ((20.59025 41.85541, 20.52295 42.21787... NaN				NaN		
175	POLYGON ((-61.68000 10.76000, -61.10500 10.890... TTO				TTO		
176	POLYGON ((30.83385 3.50917, 29.95350 4.17370, ... NaN				NaN		
	Population	Countries	area	Continent			
0	889953.0	Fiji	1.639511	OC			
1	NaN	NaN	NaN	NaN			
2	NaN	NaN	NaN	NaN			
3	37589262.0	Canada	1712.995228	NA			
4	NaN	NaN	NaN	NaN			
..	...	...	...	...			
172	6944975.0	Serbia	8.604719	EU			
173	622137.0	Montenegro	1.479321	EU			
174	NaN	NaN	NaN	NaN			
175	1394973.0	Trinidad and Tobago	0.639000	NA			
176	NaN	NaN	NaN	NaN			

[177 rows x 12 columns]

In [319...]

```
import numpy as np
from sklearn.impute import SimpleImputer

# Initialize SimpleImputer
```

```

imp = SimpleImputer(missing_values=np.nan, strategy='median')
# Reshape the 'area' column to a 2D array as SimpleImputer expects
area_column = merged_data['area'].values.reshape(-1, 1)
# Fit and transform the data
merged_data['area'] = imp.fit_transform(area_column)

# Now, merged_data['area'] will contain the imputed values

```

In [320...]

```
#merged_data.head(10)
merged_data.head() # columns of merged_data are pop_est, continent, name, iso_a3, gdp_md_est, geometry
```

Out[320]:

	pop_est	continent	name	iso_a3	gdp_md_est	geometry	iso3	cndcountryCode	Population	Countries	area	Continer
0	889953.0	Oceania	Fiji	FJI	5496	MULTIPOLYGON (((180.00000 -16.06713, 180.00000...))	FJI	FJI	889953.0	Fiji	1.639511	O
1	58005463.0	Africa	Tanzania	TZA	63177	POLYGON ((33.90371 -0.95000, 34.07262 -1.05982...))	NaN	NaN	NaN	NaN	19.879816	Na
2	603253.0	Africa	W. Sahara	ESH	907	POLYGON ((-8.66559 27.65643, -8.66512 27.58948...))	NaN	NaN	NaN	NaN	19.879816	Na
3	37589262.0	North America	Canada	CAN	1736425	MULTIPOLYGON (((-122.84000 49.00000, -122.9742...)))	CAN	CAN	37589262.0	Canada	1712.995228	N
4	328239523.0	North America	United States of America	USA	21433226	MULTIPOLYGON (((-122.84000 49.00000, -120.0000...)))	NaN	NaN	NaN	NaN	19.879816	Na

In [321...]

```
GisDataWithImputationTable = merged_data.copy()
```

In [322...]

```
# From GisDataWithImputation drop columns contain NAN

GisDataWithImputationTable.dropna(axis=1, inplace=True)
GisDataWithImputationTable.head()
```

Out[322]:

	<b>pop_est</b>	<b>continent</b>	<b>name</b>	<b>iso_a3</b>	<b>gdp_md_est</b>	<b>geometry</b>	<b>area</b>
<b>0</b>	889953.0	Oceania	Fiji	FJI	5496	MULTIPOLYGON (((180.00000 -16.06713, 180.00000...	1.639511
<b>1</b>	58005463.0	Africa	Tanzania	TZA	63177	POLYGON ((33.90371 -0.95000, 34.07262 -1.05982...	19.879816
<b>2</b>	603253.0	Africa	W. Sahara	ESH	907	POLYGON ((-8.66559 27.65643, -8.66512 27.58948...	19.879816
<b>3</b>	37589262.0	North America	Canada	CAN	1736425	MULTIPOLYGON (((-122.84000 49.00000, -122.9742...	1712.995228
<b>4</b>	328239523.0	North America	United States of America	USA	21433226	MULTIPOLYGON (((-122.84000 49.00000, -120.0000...	19.879816

In [323...]

```
merged_data["area"].describe()
```

Out[323]:

```
count      177.000000
mean       62.802542
std        173.583415
min        0.301516
25%        8.029139
50%        19.879816
75%        48.033195
max       1712.995228
Name: area, dtype: float64
```

In [323...]

In [324...]

```
import geopandas as gpd
import matplotlib.pyplot as plt

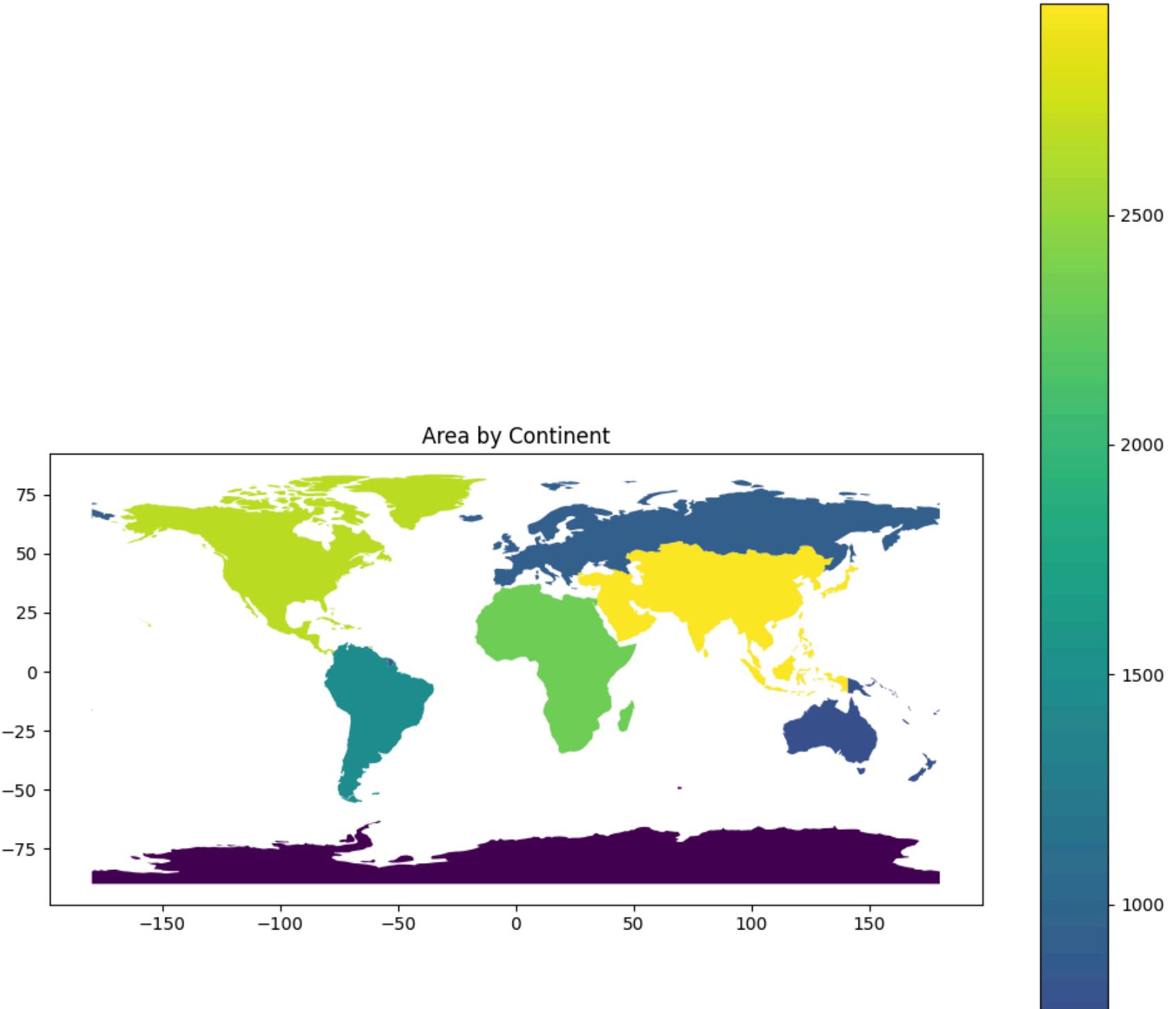
# Group data by continent and sum the Areas
area_population = merged_data.dissolve(by='continent', aggfunc='sum')

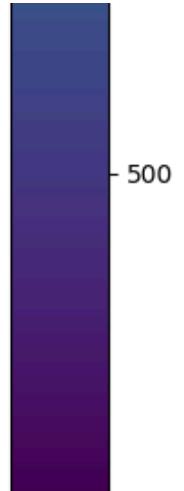
# Create a figure and axes object
fig, ax = plt.subplots(figsize=(12, 14))

# Plot the continent_population data on the axes
area_population.plot(ax=ax, column='area', legend=True)

# Set the title and show the figure
```

```
plt.title('Area by Continent')
plt.show()
```





### Summary:

Largest Areas: Asia (yellow), North America (light green), Africa (green).

Moderate Areas: South America (teal), Europe (yellow).

Smallest Areas: Oceania (blue), Antarctica (dark purple).

This map helps to visualize and compare the areas of different continents, giving a sense of their relative sizes on the world map.

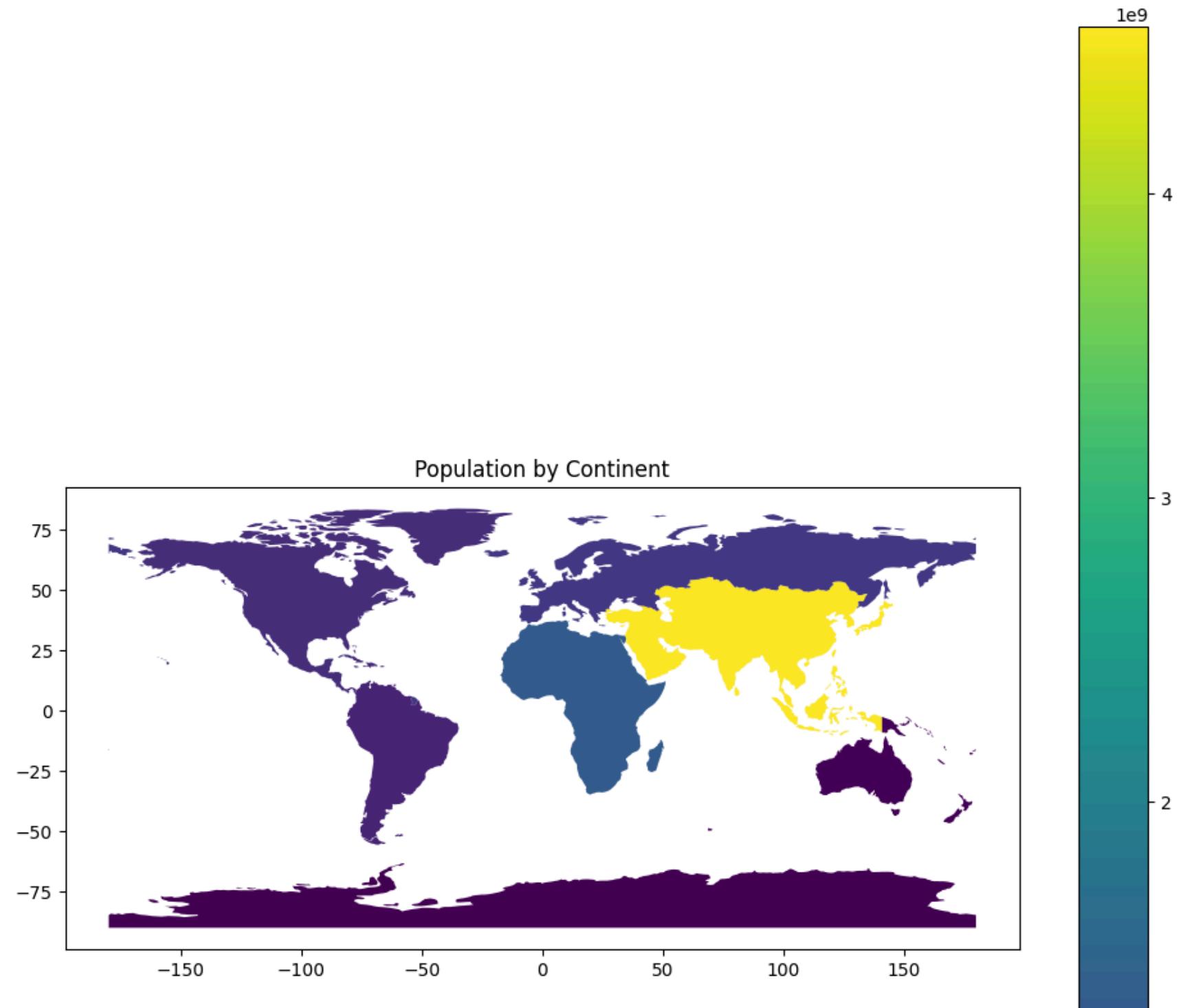
In [325...]

```
import geopandas as gpd
import matplotlib.pyplot as plt
# Group data by continent and sum the population
continent_population = merged_data.dissolve(by='continent', aggfunc='sum')

# Create a figure and axes object
fig, ax = plt.subplots(figsize=(12, 14))

# Plot the continent_population data on the axes
continent_population.plot(ax=ax, column='pop_est', legend=True)

# Set the title and show the figure
plt.title('Population by Continent')
plt.show()
```





In [326...]

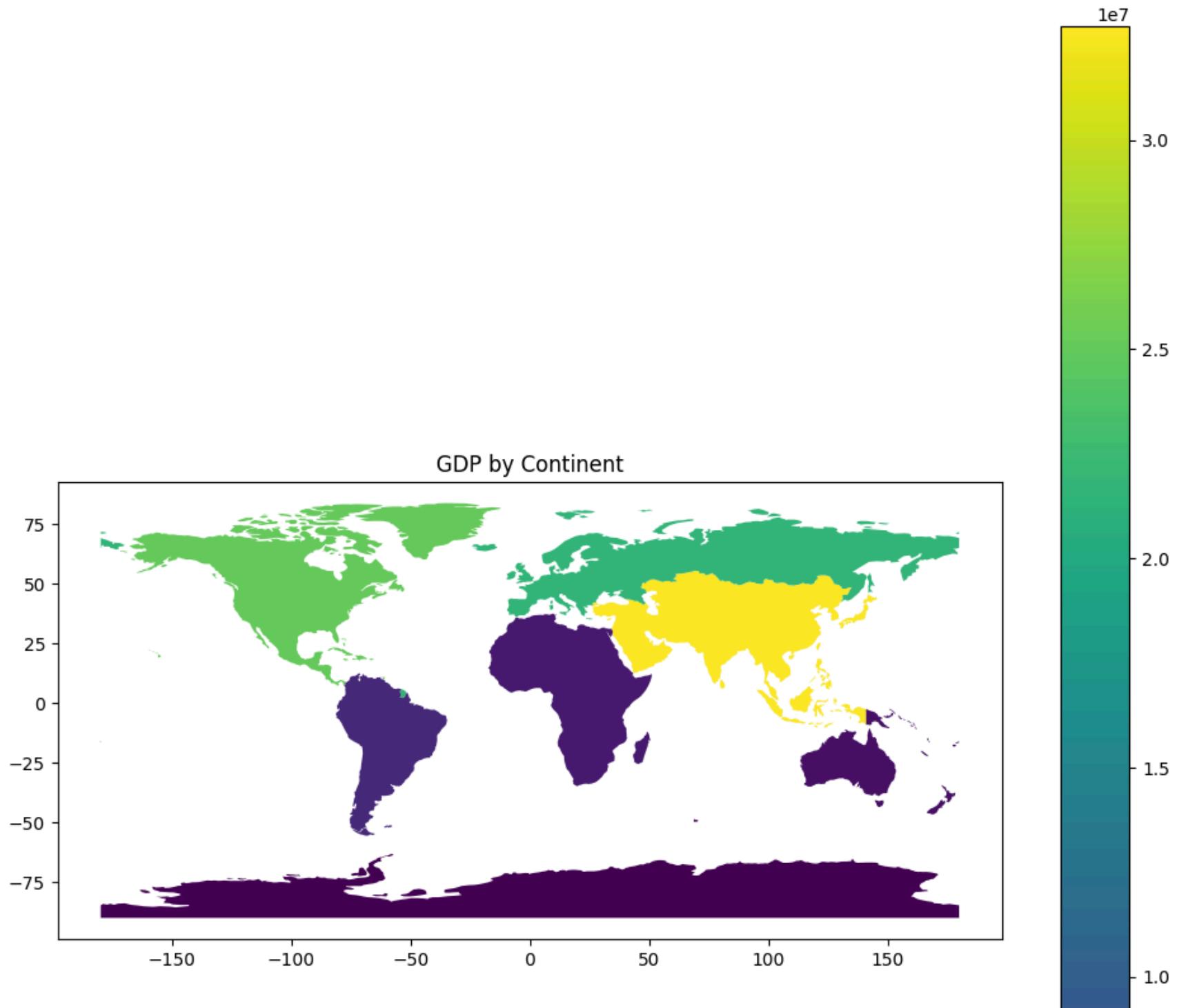
```
import geopandas as gpd
import matplotlib.pyplot as plt

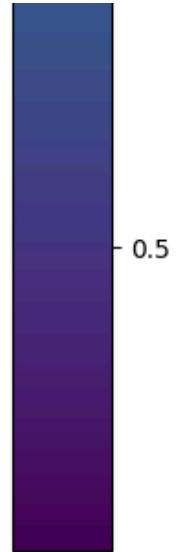
# Aggregate GDP data by continent
gdp_continent = merged_data.dissolve(by='continent', aggfunc='sum')

# Create a figure and axes object
fig, ax = plt.subplots(figsize=(12, 14))

# Plot the dissolved data on the axes
gdp_continent.plot(ax=ax, column='gdp_md_est', legend=True)

# Set the title and show the figure
plt.title('GDP by Continent')
plt.show()
```





## GIS and Regional Economic Development Planning

---

---

In [327...]

```
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
# Sum population for each continent
aggregated_data = merged_data.groupby('continent', as_index=False)[['Population']].sum()

# Sort the data by Population in descending order
sorted_data = aggregated_data.sort_values(by='Population', ascending=False)

# Create a pie chart with repeated labels using Plotly Express
fig1 = px.pie(merged_data, values='Population', names='continent', title='Population by Continent (Repeated Labels)')
fig1.update_traces(textposition='outside', textinfo='label+percent', marker=dict(colors=['#1f77b4', '#ff7f0e', '#2ca02c']))
fig1.update_layout(title='Population by Continent')
fig1.show()

# Create a bar chart with different colors for each bar
fig2 = go.Figure(data=[go.Bar(
    x=sorted_data['continent'],
    y=sorted_data['Population'],
    text=sorted_data['Population'],
    textposition='auto',
    marker=dict(color=['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#8c564b']) # Different colors for each bar
)])
```

```
)])  
  
# Update Layout  
fig2.update_layout(  
    title='Population by Continent',  
    xaxis_title='Continent',  
    yaxis_title='Population',  
    uniformtext_minsize=8,  
    uniformtext_mode='hide'  
)  
  
# Show the figure  
fig2.show()
```



In [328...]

```
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
# Create a pie chart with repeated labels using Plotly Express
fig1 = px.pie(merged_data, values='gdp_md_est', names='continent', title='GDP by Continent (Repeated Labels)')
fig1.update_traces(textposition='outside', textinfo='label+percent', marker=dict(colors=['#1f77b4', '#ff7f0e', '#2ca02c']))
fig1.update_layout(title='GDP by Continent')
fig1.show()

# Sum area for each continent
aggregated_data = merged_data.groupby('continent', as_index=False)[['area']].sum()
```

```
# Sort the data by area in descending order
sorted_data = aggregated_data.sort_values(by='area', ascending=False)

# Create a bar chart with go.Bar
fig2 = go.Figure(data=[go.Bar(
    x=sorted_data['continent'],
    y=sorted_data['area'],
    text=sorted_data['area'],
    textposition='auto',
    marker=dict(color=['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#8c564b']) # Different colors for each continent
)])

# Update layout
fig2.update_layout(
    title='Area by Continent',
    xaxis_title='Continent',
    yaxis_title='Area',
    uniformtext_minsize=8,
    uniformtext_mode='hide'
)

# Show the figure
fig2.show()
```



In [329...]

```
import dash
from dash import dcc, html
import pandas as pd

# Group data by continent for GDP, Population, and Area
grouped_data = merged_data.groupby('continent').agg({'gdp_md_est': 'sum', 'pop_est': 'sum', 'area': 'sum'}).reset_index
grouped_data = grouped_data.sort_values(by='gdp_md_est', ascending=False)

# Initialize the Dash app
app = dash.Dash(__name__)
```

```

app.title = "Global Economic Dashboard - Table" # Set the title for the browser tab

# Define the layout of the dashboard
app.layout = html.Div(children=[
    html.H1(children='Global Economic Dashboard - Table', style={'textAlign': 'center'}),

    # Table of Recapitulation
    html.Div(children=[
        html.H2(children='Table of Recapitulation', style={'textAlign': 'center'}),
        html.Table([
            html.Tr([html.Th('Continent'), html.Th('Area'), html.Th('Population'), html.Th('GDP')]),
            *[[
                html.Tr([html.Td(continent), html.Td(area), html.Td(pop_est), html.Td(gdp)])
                for continent, area, pop_est, gdp in zip(grouped_data['continent'], grouped_data['area'], grouped_data[
                ],
                ], style={'border': '1px solid black', 'borderCollapse': 'collapse', 'width': '100%', 'margin': '0 auto'})
            ], style={'width': '80%', 'margin': '0 auto'})
        ])
    ])

# Run the Dash app
if __name__ == '__main__':
    app.run_server(debug=True, port=8050)

```

In [330...]

```

import dash
from dash import dcc, html
import plotly.express as px
import pandas as pd
# Group data by continent for GDP, Population, and Area
grouped_data = merged_data.groupby('continent').agg({'gdp_md_est': 'sum', 'pop_est': 'sum', 'area': 'sum'}).reset_index
grouped_data = grouped_data.sort_values(by='gdp_md_est', ascending=False)

# Identify top 3 continents by GDP
top_3_gdp_continents = grouped_data.head(3)['continent']

# Initialize the Dash app
app = dash.Dash(__name__)
app.title = "Global Economic Dashboard - Charts" # Set the title for the browser tab

# Define the layout of the dashboard
app.layout = html.Div(children=[
    html.H1(children='Global Economic Dashboard - Charts', style={'textAlign': 'center'}),

    # Bar Chart for top 3 continents by GDP
    html.Div([

```

```

html.H2(children='Bar Chart for Top 3 Continents by GDP', style={'textAlign': 'center'}),
dcc.Graph(
    id='bar-chart',
    figure=px.bar(
        grouped_data.head(3),
        x='continent', y='gdp_md_est', title='Top 3 Continents by GDP', color='continent',
        labels={'gdp_md_est': 'GDP (in million dollars)', 'continent': 'Continent'},
        template='plotly_dark',
        color_discrete_sequence=['#1f77b4', '#ff7f0e', '#2ca02c']
    )
),
], style={'width': '48%', 'display': 'inline-block}),

# Pie Chart for top 3 continents by GDP
html.Div([
    html.H2(children='Pie Chart for Top 3 Continents by GDP', style={'textAlign': 'center'}),
    dcc.Graph(
        id='pie-chart',
        figure=px.pie(
            grouped_data.head(3), values='gdp_md_est', names='continent', title='Top 3 Continents by GDP',
            template='plotly_dark',
            color_discrete_sequence=['#1f77b4', '#ff7f0e', '#2ca02c']
        )
    ),
], style={'width': '48%', 'display': 'inline-block'})
])

# Run the Dash app
if __name__ == '__main__':
    app.run_server(debug=True, port=8051)

```

### Interpretation:

Asia, with the largest population among the listed continents, also boasts the highest GDP. Similarly, North America and Europe, which have substantial populations, exhibit significant GDP figures. Conversely, continents with smaller populations, such as Oceania and Antarctica, have lower GDPs.

While land area may play a role in economic development, as evidenced by Asia's vast expanse, it does not seem to be the sole determinant of GDP. This is illustrated by the fact that continents like Africa, despite having extensive land areas, have relatively lower GDPs compared to continents with smaller landmasses but larger populations.

Therefore, population size appears to have a more pronounced effect on a continent's GDP compared to land area.

In [331...]

```
#####
```

In [332...]

```
#####
```

In [333...]

```
# Dimension Tables of Unicef DW(Dimension Tables provide unique features that improve data processing and analytics, such as Descriptive attributes: Attributes in the Dimension Table offer context to numerical data in Fact Tables.)
```

In [334...]

```
df_languages.head(1)
```

Out[334]:

	languageID	name	isO6392	editedBy	editedDate
0	1	English	eng	fcelli@unicef.org	

In [335...]

```
DimTablelanguages = df_languages.drop(['languageID', 'editedBy', 'editedDate'], axis=1)
```

In [336...]

```
DimTablelanguages.head(1) #Dimension Tables of Unicef DW
```

Out[336]:

	name	isO6392
0	English	eng

In [337...]

```
df_organizations.head(1)
```

Out[337]:

	id	organization	editedBy	editedDate	code	indicators
0	1	United Nations Children's Fund	yjaques@unicef.org	2020-05-15T13:24:09.807	UNICEF	[]

In [338...]

```
DimTableOrganizations = df_organizations.drop(['id', 'editedBy', 'editedDate', 'indicators'], axis=1)
```

In [339...]

```
DimTableOrganizations.head(1) #Dimension Tables of Unicef DW
```

Out[339]:

	organization	code
0	United Nations Children's Fund	UNICEF

In [340...]

## Working with SQLite Databases using Python and Pandas

### DataFrame to SQL Database Insertion for Data Analysts, BI Engineers, and Data Scientists Using Pandas to\_sql

#### Connect to SQLite Database

In [341...]

```
import sqlite3
try:
    conn = sqlite3.connect('unicef.db')
    print("Opened database successfully")
except sqlite3.Error as e:
    print("Error connecting to SQLite database:", e)
```

Opened database successfully

"Table 'Actualcountries' created and loaded from 'CurrentcountriesTable' DataFrame."

In [342...]

```
DB ="/content/unicef.db"
def create_and_load_table(df, table_name, db_name):
    try:
        # Connect to SQLite database
        conn = sqlite3.connect(db_name)
        print("Opened database successfully")

        # Convert lists to strings
        df_copy = df.copy() # Create a copy to avoid modifying the original DataFrame
        df_copy['Regions_Lists'] = df_copy['Regions_Lists'].apply(json.dumps)
        df_copy['Sector_Lists'] = df_copy['Sector_Lists'].apply(json.dumps)

        # Write DataFrame to SQLite database
        df_copy.to_sql(table_name, conn, if_exists='replace', index=False)
        print("Table created and loaded successfully")

    except sqlite3.Error as e:
        print("Error connecting to SQLite database:", e)
```

```
    finally:
        # Close connection
        conn.close()

create_and_load_table(CurrentcountriesTable, "Actualcountries", DB)
```

Opened database successfully  
Table created and loaded successfully

In [342...]

In [343...]

```
# Define the database file path
DB = "/content/unicef.db"

def create_and_load_table(df, table_name, db_name):
    try:
        # Connect to SQLite database
        conn = sqlite3.connect(db_name)
        print("Opened database successfully")

        # Write DataFrame to SQLite database
        df.to_sql(table_name, conn, if_exists='replace', index=False)
        print("Table created and loaded successfully")

    except sqlite3.Error as e:
        print("Error connecting to SQLite database:", e)
    finally:
        # Close connection
        conn.close()

# Example usage to Load DimTableLanguages
create_and_load_table(DimTableLanguages, "DimTableLanguages", DB)

# Example usage to Load DimTableOrganizations
create_and_load_table(DimTableOrganizations, "DimTableOrganizations", DB)
#####
# Example usage to Load GisDataWithImputationTable table
#create_and_load_table(GisDataWithImputationTable, "GisDataWithImputationTable", DB)

# Example usage to Load DataCleanAndPrepareForML table
#create_and_load_table(DataCleanAndPrepareForML, "DataCleanAndPrepareForML", DB)

# Example usage to Load WordGeoPandasBase table
#create_and_load_table(WordGeoPandasBase, "WordGeoPandasBase", DB)
```

```
Opened database successfully
Table created and loaded successfully
Opened database successfully
Table created and loaded successfully
```

In [343...]

In [344...]

```
import sqlite3
import pandas as pd
import json

# Define the database file path
DB = "/content/unicef.db"

def create_and_load_table(df, table_name, db_name):
    try:
        # Connect to SQLite database
        conn = sqlite3.connect(db_name)
        print("Opened database successfully")

        # Convert unsupported types to strings
        df_copy = df.copy() # Create a copy to avoid modifying the original DataFrame
        for column in df_copy.columns:
            if df_copy[column].dtype == 'object':
                # Convert lists, dicts, or other unsupported types to JSON strings
                df_copy[column] = df_copy[column].apply(lambda x: json.dumps(x) if isinstance(x, (list, dict)) else x)

        # Write DataFrame to SQLite database
        df_copy.to_sql(table_name, conn, if_exists='replace', index=False)
        print(f"Table '{table_name}' created and loaded successfully")

    except sqlite3.Error as e:
        print(f"Error connecting to SQLite database: {e}")
    finally:
        # Close connection
        conn.close()

# Create and Load tables
create_and_load_table(DataCleanAndPrepareForML, "DataCleanAndPrepareForML", DB)
```

```
Opened database successfully
Table 'DataCleanAndPrepareForML' created and loaded successfully
```

In [ ]:

```
import sqlite3
import pandas as pd
```

```

import geopandas as gpd

# Define the database file path
DB = "/content/unicef.db"

def create_and_load_table(df, table_name, db_name):
    try:
        # Connect to SQLite database
        conn = sqlite3.connect(db_name)
        print("Opened database successfully")

        # Convert GeoPandas geometries to WKT (Well-Known Text)
        df_copy = df.copy() # Create a copy to avoid modifying the original DataFrame
        if isinstance(df_copy, gpd.GeoDataFrame):
            df_copy['geometry'] = df_copy['geometry'].apply(lambda geom: geom.wkt)

        # Write DataFrame to SQLite database
        df_copy.to_sql(table_name, conn, if_exists='replace', index=False)
        print(f"Table '{table_name}' created and loaded successfully")

    except sqlite3.Error as e:
        print(f"Error connecting to SQLite database: {e}")
    finally:
        # Close connection
        conn.close()

# Example usage
# Assuming WordGeoPandasBase and GisDataWithImputationTable are GeoPandas GeoDataFrames
create_and_load_table(WordGeoPandasBase, "WordGeoPandasBase", DB)
create_and_load_table(GisDataWithImputationTable, "GisDataWithImputationTable", DB)

```

In [345...]

## Load verification

In [346...]

```

import sqlite3

def get_all_tables(db_name):
    try:
        # Connect to SQLite database
        conn = sqlite3.connect(db_name)

        # Create a cursor object
        cursor = conn.cursor()

```

```
# Execute the query to fetch all tables
cursor.execute("SELECT name FROM sqlite_master WHERE type='table';")

# Fetch all the tables
tables = cursor.fetchall()

# Print the tables
print("Tables in the database:")
for table in tables:
    print(table[0])

except sqlite3.Error as e:
    print("Error connecting to SQLite database:", e)
finally:
    # Close connection
    conn.close()

get_all_tables("unicef.db")
```

Tables in the database:  
Actualcountries  
DimTableLanguages  
DimTableOrganizations  
DataCleanAndPrepareForML  
WordGeoPandasBase  
GisDataWithImputationTable

In [347...]

```
# Define the database file path
DB = "/content/unicef.db"

# Connect to SQLite database
conn = sqlite3.connect(DB)

# Read data from DimTableLanguages into a DataFrame
df_languages = pd.read_sql_query("SELECT * FROM DimTableLanguages", conn)

# Read data from DimTableOrganizations into a DataFrame
df_organizations = pd.read_sql_query("SELECT * FROM DimTableOrganizations", conn)

# Close the connection
conn.close()

# Print the DataFrames
print("Data from DimTableLanguages:")
```

```
print(df_languages)

print("\nData from DimTableOrganizations:")
print(df_organizations)
```

Data from DimTablelanguages:

```
    name isO6392
0 English eng
1 French fre
2 Spanish spa
3 Arabic ara
```

Data from DimTableOrganizations:

	organization	code
0	United Nations Children's Fund	UNICEF
1	United Nations Statistical Division	UNSD
2	World Bank	WB
3	UNICEF Private Fundraising and Partnerships Di...	PPF
4	UNICEF Europe and Central Asia Regional Office	ECARO
5	United Nations Children's Fund Programme Division	UNICEF_PD
6	United Nations Population Fund	UNFPA
7	World Health Organization	WHO
8	UNICEF Middle East and North Africa Office	MENARO

In [348]: df\_languages.head(1)

Out[348]: name isO6392

```
0 English eng
```

In [349]: df\_organizations.head(1)

Out[349]: organization code

```
0 United Nations Children's Fund UNICEF
```

## Data scientist's responsibilities: Datasets and Models

- 
- 
- 
- 1.Data preparation and cleaning.
-

Collecting data from transactional databases(The data scientist writes queries to extract relevant data and proceeds to prepare the data by cleaning and preprocessing it.)

In [350...]

```
# Define the database file path
DB = "/content/unicef.db"
# Connect to SQLite database
conn = sqlite3.connect(DB)
# Read data from DataCleanAndPrepareForML into a DataFrame
df_DataCleanAndPrepareForML = pd.read_sql_query("SELECT * FROM DataCleanAndPrepareForML", conn)
# Close the connection
conn.close()
# Print the DataFrames
print("Data from DataCleanAndPrepareForML:")
df_DataCleanAndPrepareForML.head()
#print("\nData from DataCleanAndPrepareForML:")
```

Data from DataCleanAndPrepareForML:

Out[350]:

	name	continent	Area	pop_est	gdp_md_est	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country	pop_dens
0	Fiji	Oceania	1.639511	889953.0	5496	[7]	5.0	1.0	5.428161e+
1	United Republic of Tanzania	Africa	76.301964	58005463.0	63177	[2, 7, 18]	5.0	3.0	7.602093e+
2	Western Sahara	Africa	8.603984	603253.0	907	[]	2.0	0.0	7.011322e+
3	Canada	North America	1712.995228	37589262.0	1736425	[]	5.0	0.0	2.194359e+
4	United States	North America	1122.281921	328239523.0	21433226	[]	5.0	0.0	2.924751e+

Evaluating for missing data

In [351...]

```
df_DataCleanAndPrepareForML.isnull().sum()
```

```
Out[351]: name          0  
continent      0  
Area           0  
pop_est        0  
gdp_md_est     0  
Sector_Lists   0  
Total_Organization_By_country 0  
Total_Sector_By_country    0  
pop_density     0  
GDP_Category    0  
Priority        0  
Organization     0  
Armed_Conflict  0  
dtype: int64
```

Check the data type

```
In [352... df_DataCleanAndPrepareForML.dtypes
```

```
Out[352]: name          object  
continent      object  
Area           float64  
pop_est        float64  
gdp_md_est     int64  
Sector_Lists   object  
Total_Organization_By_country float64  
Total_Sector_By_country    float64  
pop_density     float64  
GDP_Category    object  
Priority        object  
Organization     object  
Armed_Conflict  object  
dtype: object
```

```
In [352... ]:
```

## Exploratory Data Analysis

Create boxplots to identify points that may be outliers.

---

---

```
In [353... import pandas as pd  
import seaborn as sns
```

```
import matplotlib.pyplot as plt

# Assuming cleaned_df is your DataFrame containing the data
# Numerical columns to plot
numerical_cols = ['Area', 'pop_est', 'gdp_md_est']

# Number of plots per row
plots_per_row = 3

# Calculate the number of rows needed
num_rows = (len(numerical_cols) + plots_per_row - 1) // plots_per_row

# Create a figure and axes for subplots
fig, axes = plt.subplots(num_rows, plots_per_row, figsize=(12, 6 * num_rows))

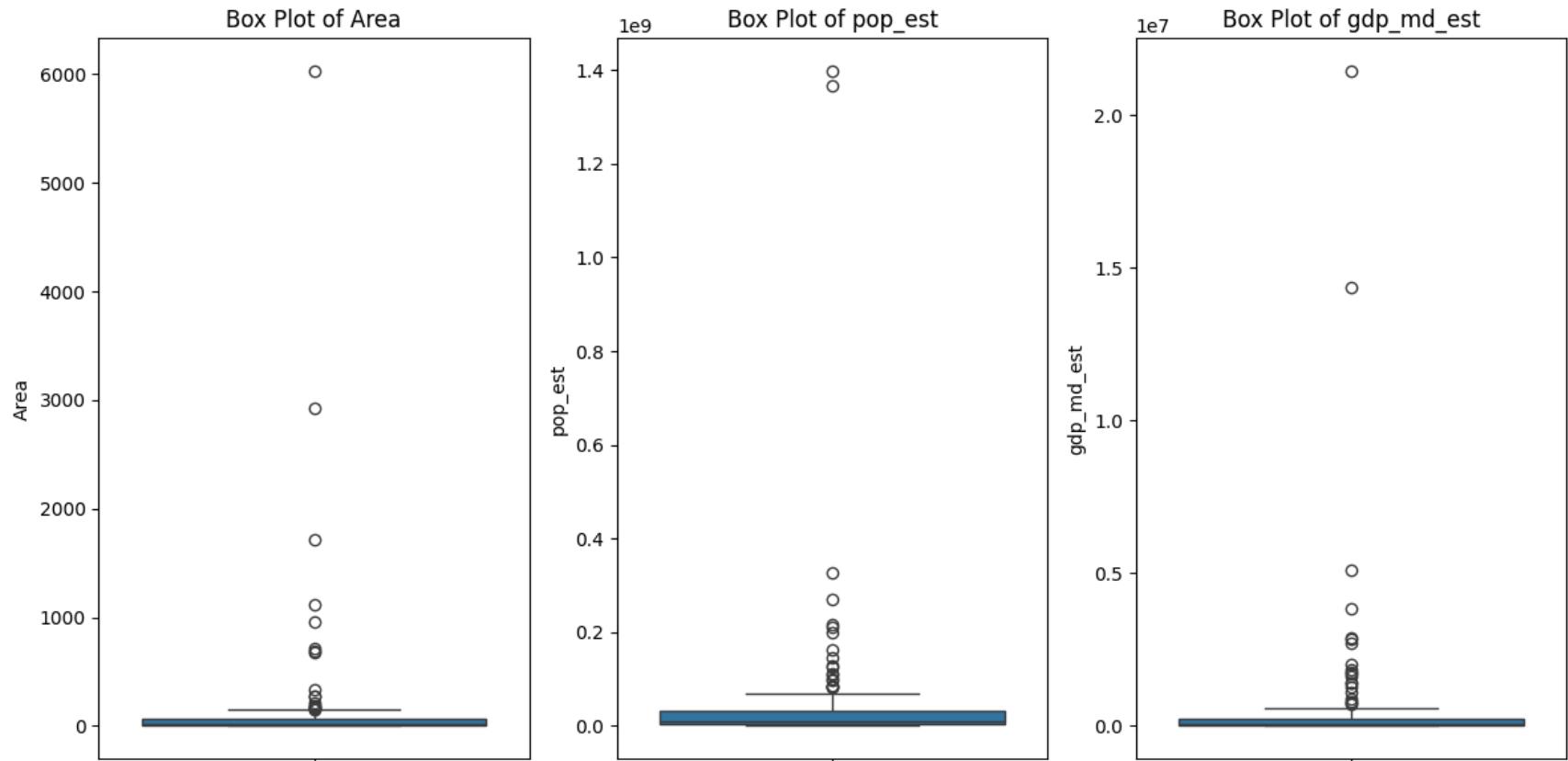
# Flatten the axes array for easy iteration
axes = axes.flatten()

# Plot each numerical column as a box plot
for i, col in enumerate(numerical_cols):
    sns.boxplot(data=df_DataCleanAndPrepareForML, y=col, ax=axes[i])
    axes[i].set_title(f'Box Plot of {col}')

# Remove any unused axes if the number of columns is not a multiple of plots_per_row
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

# Adjust layout to prevent overlapping
plt.tight_layout()

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



In [354...]

```

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

# Assuming cleaned_df is your DataFrame containing the data
# Numerical columns to plot
numerical_cols = ['Total_Organization_By_country', 'Total_Sector_By_country', 'pop_density']

# Number of plots per row
plots_per_row = 3

# Calculate the number of rows needed
num_rows = (len(numerical_cols) + plots_per_row - 1) // plots_per_row

# Create a figure and axes for subplots
fig, axes = plt.subplots(num_rows, plots_per_row, figsize=(12, 6 * num_rows))

# Flatten the axes array for easy iteration

```

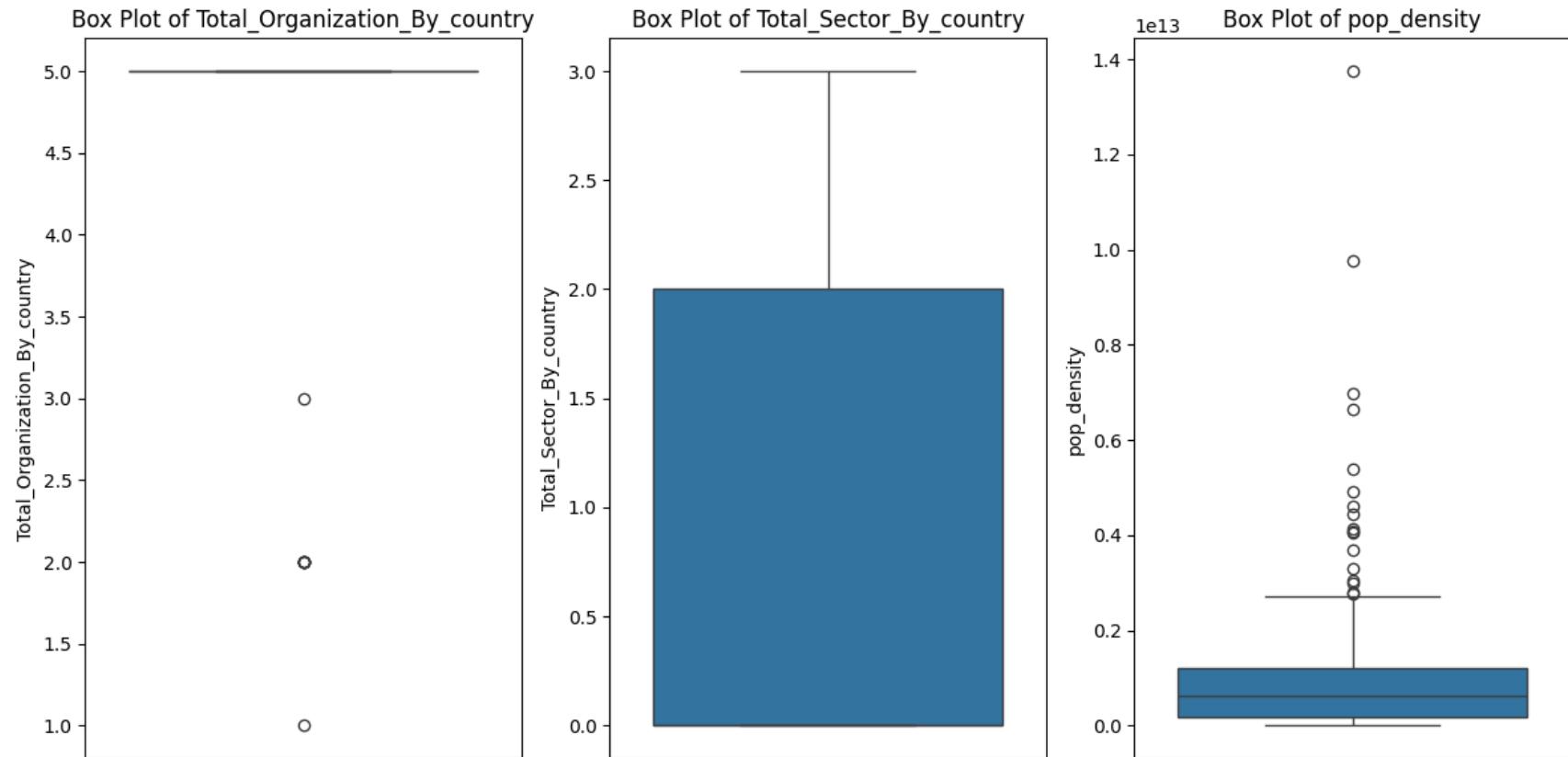
```
axes = axes.flatten()

# Plot each numerical column as a box plot
for i, col in enumerate(numerical_cols):
    sns.boxplot(data=df_DataCleanAndPrepareForML, y=col, ax=axes[i])
    axes[i].set_title(f'Box Plot of {col}')

# Remove any unused axes if the number of columns is not a multiple of plots_per_row
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

# Adjust layout to prevent overlapping
plt.tight_layout()

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



In [355...]

```

import pandas as pd
# Select numerical columns
numerical_cols = ['Area', 'pop_est', 'gdp_md_est', 'Total_Organization_By_country', 'Total_Sector_By_country', 'pop_den
# Function to count outliers using the IQR method
def count_outliers(series):
    Q1 = series.quantile(0.25)
    Q3 = series.quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = series[(series < lower_bound) | (series > upper_bound)]
    return outliers.count()

# Dictionary to store outlier counts
outlier_counts = {}

# Calculate outlier counts for each numerical column
for col in numerical_cols:
    outlier_counts[col] = count_outliers(DataCleanAndPrepareForML[col])

# Display the outlier counts
for col, count in outlier_counts.items():
    print(f'{col}: {count} outliers')

```

Area: 18 outliers  
 pop\_est: 19 outliers  
 gdp\_md\_est: 20 outliers  
 Total\_Organization\_By\_country: 8 outliers  
 Total\_Sector\_By\_country: 0 outliers  
 pop\_density: 17 outliers

In [356...]

```

import pandas as pd
# Select numerical columns
numerical_cols = ['Area', 'pop_est', 'gdp_md_est', 'Total_Organization_By_country', 'Total_Sector_By_country', 'pop_den
# Function to remove outliers using the IQR method
def remove_outliers(df, columns):
    for col in columns:
        Q1 = df[col].quantile(0.25)
        Q3 = df[col].quantile(0.75)
        IQR = Q3 - Q1
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR
        df = df[(df[col] >= lower_bound) & (df[col] <= upper_bound)]
    return df

```

```
# Remove outliers from the DataFrame
cleaned_df = remove_outliers(DataCleanAndPrepareForML, numerical_cols)

# Display the shape of the original and cleaned DataFrame
print(f'Original DataFrame shape: {DataCleanAndPrepareForML.shape}')
print(f'Cleaned DataFrame shape: {cleaned_df.shape}')

# Save the cleaned DataFrame to a new CSV file (optional)
# cleaned_df.to_csv('cleaned_data.csv', index=False)
```

Original DataFrame shape: (177, 13)

Cleaned DataFrame shape: (101, 13)

In [ ]: # # Convert 'Sector\_Lists' to string type  
cleaned\_df['Sector\_Lists'] = cleaned\_df['Sector\_Lists'].astype(str)

In [358...]: cleaned\_df.head()

Out[358]:

	name	continent	Area	pop_est	gdp_md_est	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country	pop_dens
0	Fiji	Oceania	1.639511	889953.0	5496	[7]	5.0	1.0	5.428161e+
6	Uzbekistan	Asia	49.958821	33580650.0	57921	[]	5.0	0.0	6.721666e+
7	Papua New Guinea	Oceania	37.991091	8776109.0	24829	[7]	5.0	1.0	2.310044e+
12	Somalia	Africa	39.528186	10192317.3	4719	[7, 18]	5.0	2.0	2.578494e+
13	Kenya	Africa	48.033195	52573973.0	95503	[2, 7, 18]	5.0	3.0	1.094534e+

Checking the data After deleting the outliers

In [359...]:

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

# Assuming cleaned_df is your DataFrame containing the data
# Numerical columns to plot
numerical_cols = ['Total_Organization_By_country', 'Total_Sector_By_country', 'pop_density']
```

```
# Number of plots per row
plots_per_row = 3

# Calculate the number of rows needed
num_rows = (len(numerical_cols) + plots_per_row - 1) // plots_per_row

# Create a figure and axes for subplots
fig, axes = plt.subplots(num_rows, plots_per_row, figsize=(12, 6 * num_rows))

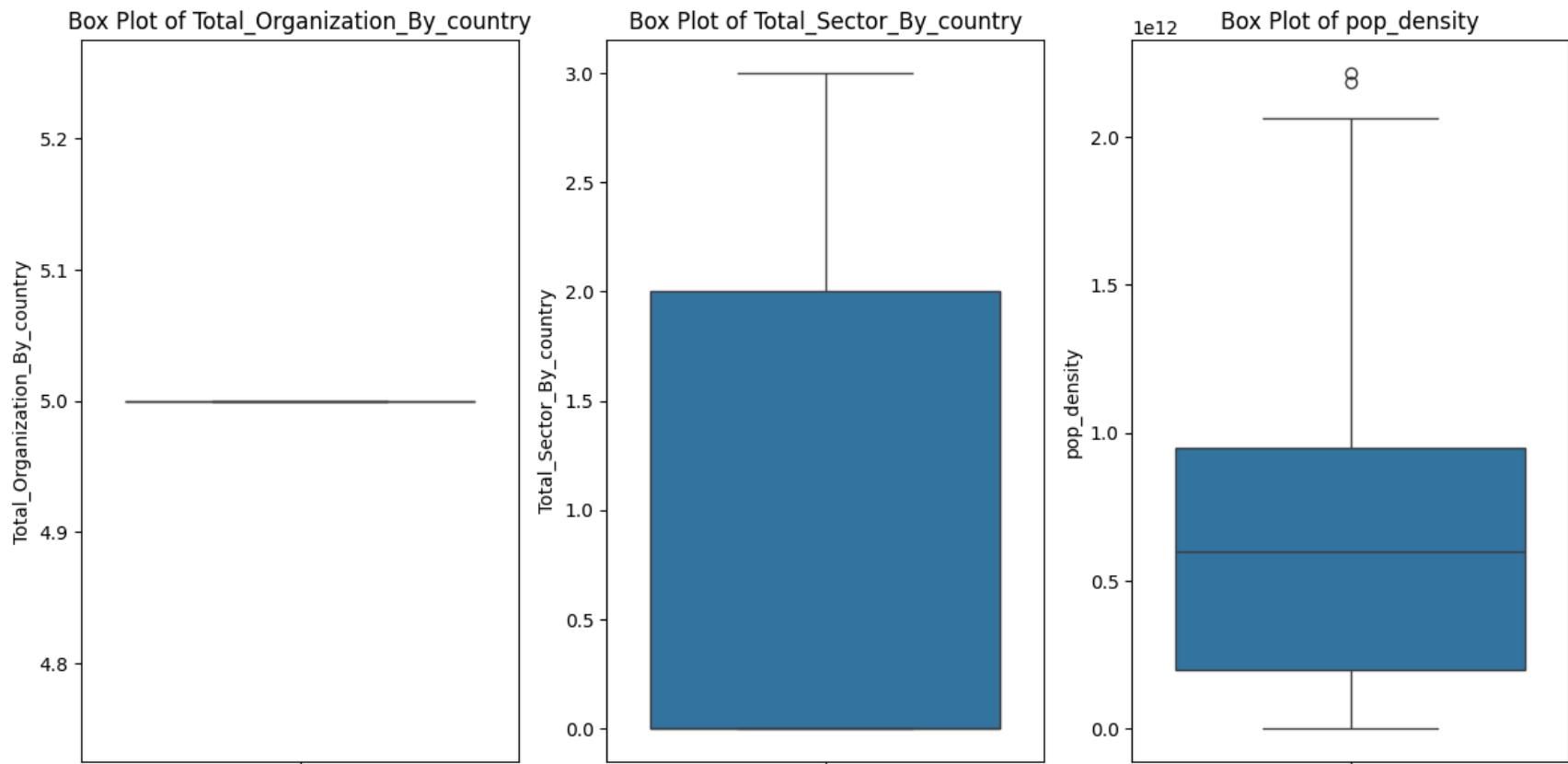
# Flatten the axes array for easy iteration
axes = axes.flatten()

# Plot each numerical column as a box plot
for i, col in enumerate(numerical_cols):
    sns.boxplot(data=cleaned_df, y=col, ax=axes[i])
    axes[i].set_title(f'Box Plot of {col}')

# Remove any unused axes if the number of columns is not a multiple of plots_per_row
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

# Adjust layout to prevent overlapping
plt.tight_layout()

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



### Analyzing Bivariate Data:Categorical

In [360...]

```

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
# Numerical columns to plot
numerical_cols = ['Area', 'pop_est', 'gdp_md_est']

# Number of plots per row
plots_per_row = 3

# Calculate the number of rows needed
num_rows = (len(numerical_cols) + plots_per_row - 1) // plots_per_row

# Create a figure and axes for subplots
fig, axes = plt.subplots(num_rows, plots_per_row, figsize=(12, 6 * num_rows))

```

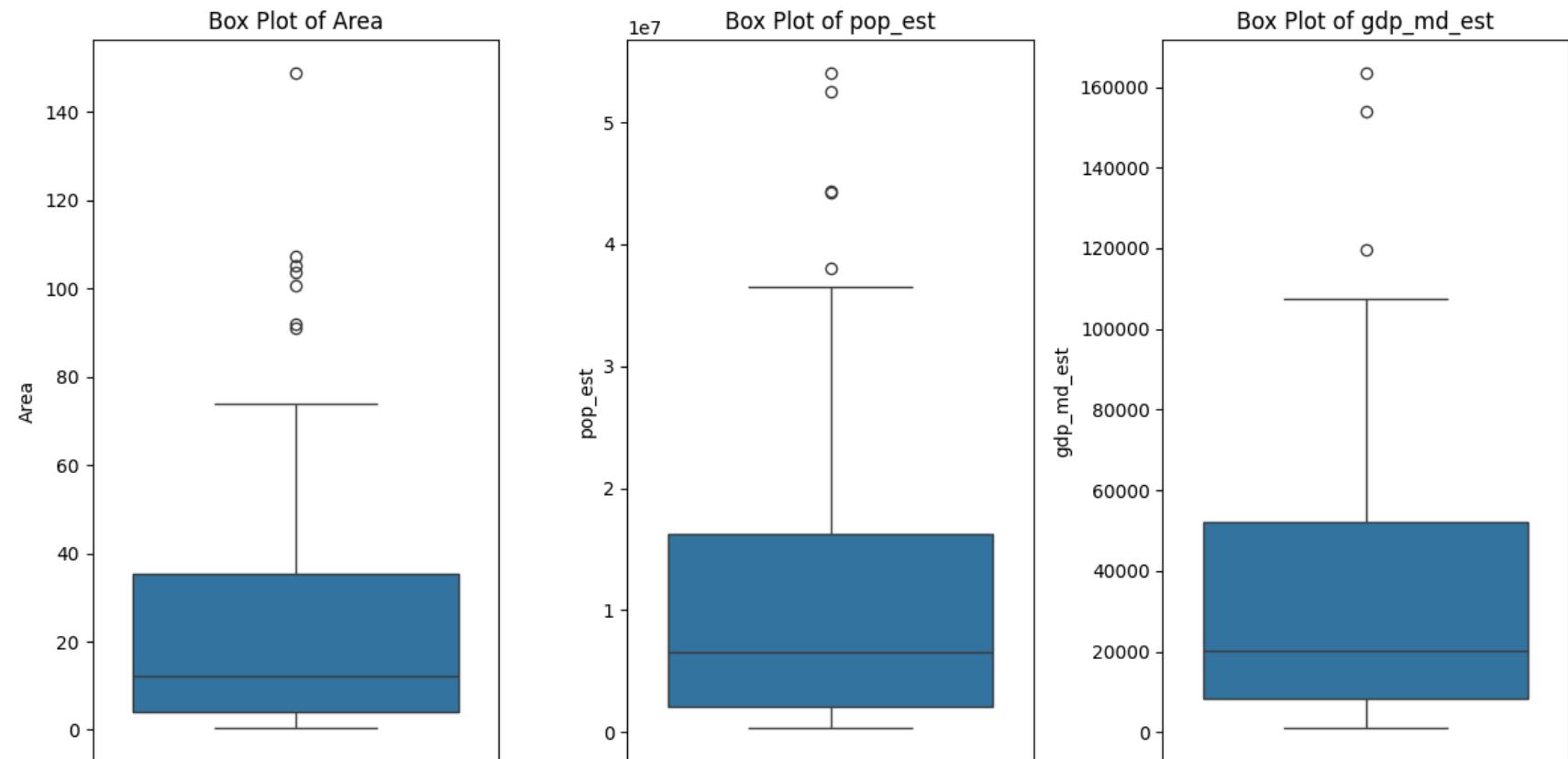
```
# Flatten the axes array for easy iteration
axes = axes.flatten()

# Plot each numerical column as a box plot
for i, col in enumerate(numerical_cols):
    sns.boxplot(data=cleaned_df, y=col, ax=axes[i])
    axes[i].set_title(f'Box Plot of {col}')

# Remove any unused axes if the number of columns is not a multiple of plots_per_row
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

# Adjust Layout to prevent overlapping
plt.tight_layout()

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



In [361...]

```
import pandas as pd
# Selecting the columns of interest
selected_cols = ['Priority', 'continent']

# Creating a subset of the DataFrame with the selected columns
subset_df = cleaned_df[selected_cols]

# Performing a cross-tabulation (contingency table)
cross_tab = pd.crosstab(subset_df['Priority'], subset_df['continent'])
# Calculate row-wise percentages
row_sum = cross_tab.sum(axis=1)
cross_tab_percent = cross_tab.div(row_sum, axis=0) * 100
# Displaying the cross-tabulation
cross_tab
#cross_tab_percent
```

Out[361]: continent Africa Asia Europe North America Oceania South America

Priority						
No	9	13	16	9	1	6
Yes	31	11	1	1	3	0

In [362...]

cross\_tab\_percent

Out[362]: continent Africa Asia Europe North America Oceania South America

Priority						
No	16.666667	24.074074	29.62963	16.666667	1.851852	11.111111
Yes	65.957447	23.404255	2.12766	2.127660	6.382979	0.000000

In [363...]

```
import pandas as pd

# Assuming cleaned_df is your DataFrame containing the data
# Selecting the columns of interest
selected_cols = ['Priority', 'Sector_Lists']

# Creating a subset of the DataFrame with the selected columns
subset_df = cleaned_df[selected_cols]
```

```
# Convert 'Sector_Lists' to object type
subset_df['Sector_Lists'] = subset_df['Sector_Lists'].astype(str)

# Performing a cross-tabulation (contingency table)
cross_tab = pd.crosstab(subset_df['Priority'], subset_df['Sector_Lists'])

# Calculate row-wise percentages
row_sum = cross_tab.sum(axis=1)
cross_tab_percent = cross_tab.div(row_sum, axis=0) * 100

# Displaying the cross-tabulation with percentages
print("Contingency Table with Percentages for Sector_Lists:")
cross_tab
```

Contingency Table with Percentages for Sector\_Lists:

<ipython-input-363-28372af627ae>:11: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Out[363]: Sector\_Lists [2, 7, 18] [2, 7] [7, 18] [7] []

Priority					
No	0	0	0	0	54
Yes	12	2	21	12	0

In [364...]: cross\_tab\_percent

Out[364]: Sector\_Lists [2, 7, 18] [2, 7] [7, 18] [7] []

Priority					
No	0.000000	0.000000	0.000000	0.000000	100.0
Yes	25.531915	4.255319	44.680851	25.531915	0.0

In [365...]

```
import pandas as pd

# Assuming cleaned_df is your DataFrame containing the data
# Selecting the columns of interest
selected_cols = ['Priority', 'GDP_Category']

# Creating a subset of the DataFrame with the selected columns
subset_df = cleaned_df[selected_cols]

# Convert 'GDP_Category' to object type
subset_df['GDP_Category'] = subset_df['GDP_Category'].astype(str)

# Performing a cross-tabulation (contingency table)
cross_tab = pd.crosstab(subset_df['Priority'], subset_df['GDP_Category'])
cross_tab
```

<ipython-input-365-d5d4b66298d5>:11: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Out[365]: GDP\_Category Middle-income Poor

Priority		
No	42	12
Yes	24	23

In [366...]

```
# Calculate row-wise percentages
row_sum = cross_tab.sum(axis=1)
cross_tab_percent = cross_tab.div(row_sum, axis=0) * 100

# Displaying the cross-tabulation with percentages
print("Contingency Table with Percentages for GDP_Category:")
cross_tab_percent
```

Contingency Table with Percentages for GDP\_Category:

Out[366]: GDP\_Catagory Middle-income Poor

**Priority**

No	77.777778	22.222222
Yes	51.063830	48.936170

In [367...]

```
import pandas as pd
# Selecting the columns of interest
selected_cols = ['Priority', 'Organization']
# Creating a subset of the DataFrame with the selected columns
subset_df = cleaned_df[selected_cols]
# Convert 'Organization' to object type
subset_df['Organization'] = subset_df['Organization'].astype(str)
# Performing a cross-tabulation (contingency table)
cross_tab = pd.crosstab(subset_df['Priority'], subset_df['Organization'])
cross_tab
```

<ipython-input-367-3f43316d9478>:7: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Out[367]: Organization Yes

**Priority**

No	54
Yes	47

In [368...]

```
# Calculate row-wise percentages
row_sum = cross_tab.sum(axis=1)
cross_tab_percent = cross_tab.div(row_sum, axis=0) * 100

# Displaying the cross-tabulation with percentages
```

```
print("Contingency Table with Percentages for Organization:")
cross_tab_percent
```

Contingency Table with Percentages for Organization:

Out[368]: Organization Yes

Priority	
No	100.0
Yes	100.0

In [369...]

```
import pandas as pd

# Assuming cleaned_df is your DataFrame containing the data
# Selecting the columns of interest
selected_cols = ['Priority', 'Armed_Conflict']

# Creating a subset of the DataFrame with the selected columns
subset_df = cleaned_df[selected_cols]

# Convert 'Armed_Conflict' to object type
subset_df['Armed_Conflict'] = subset_df['Armed_Conflict'].astype(str)

# Performing a cross-tabulation (contingency table)
cross_tab = pd.crosstab(subset_df['Priority'], subset_df['Armed_Conflict'])
cross_tab
```

<ipython-input-369-7338c5ecd5ce>:11: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Out[369]: Armed\_Conflict No Yes

Priority		
No	34	20
Yes	24	23

In [370...]

```
# Calculate row-wise percentages
row_sum = cross_tab.sum(axis=1)
cross_tab_percent = cross_tab.div(row_sum, axis=0) * 100

# Displaying the cross-tabulation with percentages
print("Contingency Table with Percentages for Armed_Conflict:")
cross_tab_percent
```

Contingency Table with Percentages for Armed\_Conflict:

Out[370]: Armed\_Conflict No Yes

Priority		
No	62.962963	37.037037
Yes	51.063830	48.936170

## Analysis of Bivariate Quantitative Data

In [370...]

In [371...]

```
import pandas as pd

# Assuming subset_df is your DataFrame containing the data
# Selecting the columns of interest
selected_cols = ['Priority', 'Area', 'pop_est', 'gdp_md_est', 'Total_Organization_By_country', 'Total_Sector_By_country']

# Creating a subset of the DataFrame with the selected columns
subset_df = cleaned_df[selected_cols]

# Convert 'Priority' to object type
subset_df['Priority'] = subset_df['Priority'].astype(str)

# Dictionary to store pivot tables for each quantitative variable
```

```
pivot_tables = {}

# Calculate pivot table with mean for each quantitative variable
for col in ['Area', 'pop_est', 'gdp_md_est', 'Total_Organization_By_country', 'Total_Sector_By_country', 'pop_density']:
    pivot_table_mean = pd.pivot_table(subset_df, index='Priority', values=col, aggfunc='mean')
    pivot_tables[col] = pivot_table_mean

# Displaying the pivot tables
for col, pivot_table in pivot_tables.items():
    print(f"Pivot Table with Mean for {col}:")
    print(pivot_table)
    print()
```

<ipython-input-371-25db2ca10bb5>:11: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Pivot Table with Mean for Area:

	Area
Priority	
No	18.692531
Yes	31.420377

Pivot Table with Mean for pop\_est:

	pop_est
Priority	
No	6.194047e+06
Yes	1.652706e+07

Pivot Table with Mean for gdp\_md\_est:

	gdp_md_est
Priority	
No	42226.37037
Yes	23724.06383

Pivot Table with Mean for Total\_Organization\_By\_country:

	Total_Organization_By_country
Priority	
No	5.0
Yes	5.0

Pivot Table with Mean for Total\_Sector\_By\_country:

	Total_Sector_By_country
Priority	
No	0.0
Yes	2.0

Pivot Table with Mean for pop\_density:

	pop_density
Priority	
No	6.724859e+11
Yes	6.799405e+11

In [372...]

```
from scipy.stats import chi2_contingency

# Assuming subset_df is your DataFrame containing the data
# Selecting the columns of interest
selected_cols = ['Priority', 'continent', 'Sector_Lists', 'GDP_Category', 'Organization', 'Armed_Conflict']

# Creating a subset of the DataFrame with the selected columns
subset_df = cleaned_df[selected_cols]
```

```
# Convert 'Sector_Lists' to object type
subset_df['Sector_Lists'] = subset_df['Sector_Lists'].astype(str)

# Creating a contingency table for each qualitative variable
contingency_tables = {}
for col in ['continent', 'Sector_Lists', 'GDP_Category', 'Organization', 'Armed_Conflict']:
    contingency_table = pd.crosstab(subset_df['Priority'], subset_df[col])
    contingency_tables[col] = contingency_table

# Performing the chi-square test for each contingency table
chi2_results = {}
for col, contingency_table in contingency_tables.items():
    chi2, p, dof, expected = chi2_contingency(contingency_table)
    chi2_results[col] = {'chi2': chi2, 'p-value': p, 'dof': dof, 'expected': expected}

# Displaying the results
for col, result in chi2_results.items():
    #print(f"Chi-square Test Results for {col}:")
    #print(f" Chi-square value: {result['chi2']} ")
    print(f" p-value: {result['p-value']} ")
    #print(f" Degrees of Freedom: {result['dof']} ")
    print("")
```

<ipython-input-372-f1c6d01ef189>:11: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

p-value: 2.855378162200539e-07

p-value: 6.024706476186923e-21

p-value: 0.009201122844906106

p-value: 1.0

p-value: 0.31507671744681276

In [372...]

The p-value for 'continent' is approximately 2.86e-07 (very small), indicating a significant association with 'Priority'.

The p-value for 'Sector\_Lists' is approximately 6.02e-21 (very small), indicating a significant association with 'Priority'.

The p-value for 'GDP\_Category' is approximately 0.0092 (less than 0.05), indicating a significant association with 'Priority'.

The p-value for 'Organization' is 1.0 (equal to or greater than 0.05), indicating no significant association with 'Priority'.

The p-value for 'Armed\_Conflict' is approximately 0.315 (greater than 0.05), indicating no significant association with 'Priority'.

## Shapiro-Wilk Test for Normality

In [373...]

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import shapiro, skew

# Assuming cleaned_df is your DataFrame containing the data
# Filter DataFrame for Priority == 'Yes'
filtered_df = cleaned_df[cleaned_df['Priority'] == 'Yes']
# Numerical columns to test
numerical_cols = ['Area', 'pop_est', 'gdp_md_est']
# Function to perform Shapiro-Wilk Test and create histograms with skewness lines
def shapiro_test_and_histogram(df, cols):
    results = {}
    num_plots = len(cols)
    plots_per_row = 3
    num_rows = (num_plots + plots_per_row - 1) // plots_per_row

    fig, axes = plt.subplots(num_rows, plots_per_row, figsize=(15, 5 * num_rows))
    axes = axes.flatten()

    for i, col in enumerate(cols):
        # Shapiro-Wilk Test
        stat, p = shapiro(df[col])
        results[col] = {'Shapiro-Wilk Statistic': stat, 'p-value': p}

        # Determine normality
        normality = "Normal" if p > 0.05 else "Not Normal"
```

```
# Plot histogram
sns.histplot(df[col], kde=True, ax=axes[i])
axes[i].set_title(f'Histogram of {col} with Skewness')
axes[i].axvline(df[col].mean(), color='k', linestyle='dashed', linewidth=1)

# Add skewness line
skewness = skew(df[col])
axes[i].text(0.95, 0.95, f'Skewness: {skewness:.2f}\n{normality}',
            verticalalignment='top', horizontalalignment='right',
            transform=axes[i].transAxes, color='red', fontsize=12)

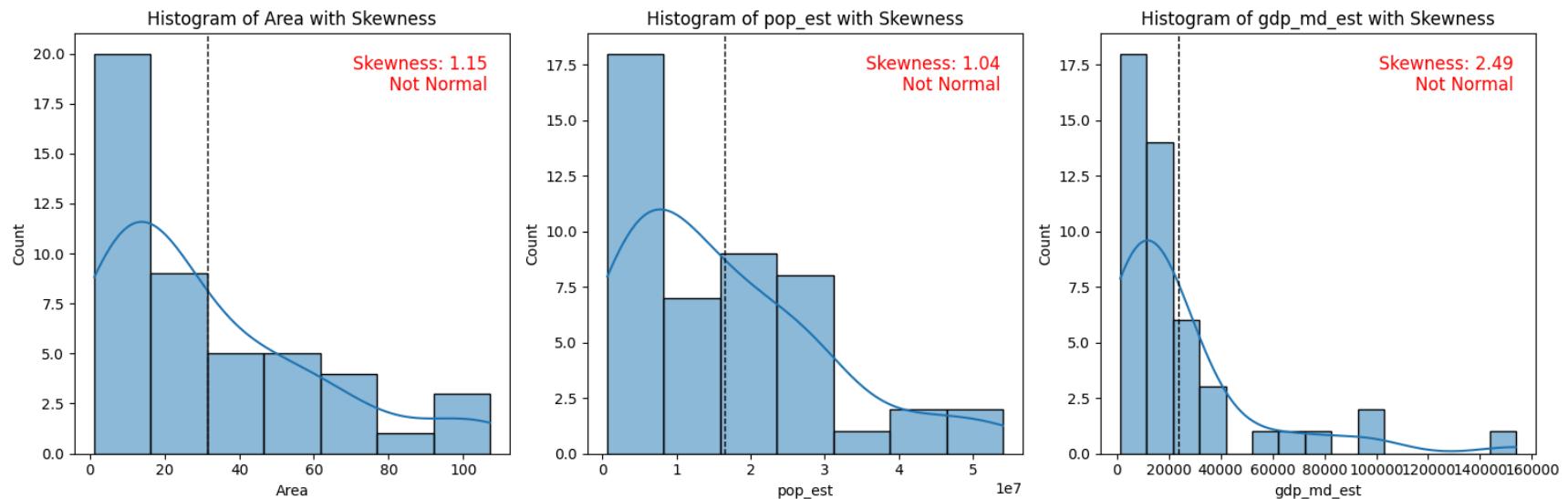
# Remove any unused axes if the number of columns is not a multiple of plots_per_row
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

plt.tight_layout()
plt.show()

return results

# Perform Shapiro-Wilk Test and plot histograms for filtered DataFrame
results = shapiro_test_and_histogram(filtered_df, numerical_cols)

# Print the Shapiro-Wilk Test results
print("Shapiro-Wilk Test Results:")
for col, result in results.items():
    print(f'{col}:')
    print(f'  Shapiro-Wilk Statistic: {result['Shapiro-Wilk Statistic']}')
    print(f'  p-value: {result['p-value']}')
    print(f'  Normality: {"Normal" if result['p-value'] > 0.05 else "Not Normal"}')
    print()
```



#### Shapiro-Wilk Test Results:

Area:

Shapiro-Wilk Statistic: 0.8564444184303284  
p-value: 3.8319463783409446e-05  
Normality: Not Normal

pop\_est:

Shapiro-Wilk Statistic: 0.8935688734054565  
p-value: 0.0004519100475590676  
Normality: Not Normal

gdp\_md\_est:

Shapiro-Wilk Statistic: 0.6850902438163757  
p-value: 9.336237560830796e-09  
Normality: Not Normal

```
In [ ]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import shapiro, skew

# Assuming cleaned_df is your DataFrame containing the data
# Filter DataFrame for Priority == 'Yes'
filtered_df = cleaned_df[cleaned_df['Priority'] == 'Yes']

# Numerical columns to test
numerical_cols = ['Total_Sector_By_country', 'pop_density']
```

```
# Function to perform Shapiro-Wilk Test and create histograms with skewness Lines
def shapiro_test_and_histogram(df, cols):
    results = {}
    num_plots = len(cols)
    plots_per_row = 3
    num_rows = (num_plots + plots_per_row - 1) // plots_per_row

    fig, axes = plt.subplots(num_rows, plots_per_row, figsize=(15, 5 * num_rows))
    axes = axes.flatten()

    for i, col in enumerate(cols):
        # Shapiro-Wilk Test
        stat, p = shapiro(df[col])
        results[col] = {'Shapiro-Wilk Statistic': stat, 'p-value': p}

        # Determine normality
        normality = "Normal" if p > 0.05 else "Not Normal"

        # Plot histogram
        sns.histplot(df[col], kde=True, ax=axes[i])
        axes[i].set_title(f'Histogram of {col} with Skewness')
        axes[i].axvline(df[col].mean(), color='k', linestyle='dashed', linewidth=1)

        # Add skewness line
        skewness = skew(df[col])
        axes[i].text(0.95, 0.95, f'Skewness: {skewness:.2f}\n{normality}',
                    verticalalignment='top', horizontalalignment='right',
                    transform=axes[i].transAxes, color='red', fontsize=12)

    # Remove any unused axes if the number of columns is not a multiple of plots_per_row
    for j in range(i + 1, len(axes)):
        fig.delaxes(axes[j])

    plt.tight_layout()
    plt.show()

    return results

# Perform Shapiro-Wilk Test and plot histograms for filtered DataFrame
results = shapiro_test_and_histogram(filtered_df, numerical_cols)

# Print the Shapiro-Wilk Test results
print("Shapiro-Wilk Test Results:")
for col, result in results.items():
```

```

print(f'{col}:')
print(f" Shapiro-Wilk Statistic: {result['Shapiro-Wilk Statistic']}")
print(f" p-value: {result['p-value']}")
print(f" Normality: {'Normal' if result['p-value'] > 0.05 else 'Not Normal'}")
print()

```

## Non-parametric statistical: The Mann-Whitney U Test(Wilcoxon Rank Sum Test)

"Because the variable doesn't meet the assumptions of normal distribution required for parametric tests, a nonparametric test, the Mann-Whitney U test, will be used."

In [375...]

```

# Change the priority variable to integer
DatasiteForMannWhitney= cleaned_df.copy()
DatasiteForMannWhitney['Priority'][DatasiteForMannWhitney['Priority']=='No']=0
DatasiteForMannWhitney['Priority'][DatasiteForMannWhitney['Priority']=='Yes']=1
DatasiteForMannWhitney['Priority']=DatasiteForMannWhitney['Priority'].astype('int64')

```

<ipython-input-375-7ed53632766f>:3: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

<ipython-input-375-7ed53632766f>:4: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

In [ ]:

```

import pandas as pd
from scipy.stats import mannwhitneyu

# Assuming DatasiteForMannWhitney is your DataFrame
# Columns to test
columns_to_test = ['Area', 'pop_est', 'gdp_md_est', 'Total_Sector_By_country', 'pop_density']

# Split the DataFrame based on the 'Priority' column
group1 = DatasiteForMannWhitney[DatasiteForMannWhitney['Priority'] == 1]
group2 = DatasiteForMannWhitney[DatasiteForMannWhitney['Priority'] == 0]

```

```
# Define significance level
alpha = 0.05

# Perform the Mann-Whitney U test
results = {}
for col in columns_to_test:
    stat, p_value = mannwhitneyu(group1[col], group2[col])
    results[col] = {'Mann-Whitney U Statistic': stat, 'p-value': p_value}

# Print the results
for col, result in results.items():
    print(f"{col}:")
    print(f"  Mann-Whitney U Statistic: {result['Mann-Whitney U Statistic']}")
    print(f"  p-value: {result['p-value']}")
    if result['p-value'] < alpha:
        print("  Result: Significant")
    else:
        print("  Result: Not Significant")
    print()
```

## Feature Engineering: Feature Selection For Machine Learning

---

---

### Converting Categorical Data into Numerical Form

```
In [377]: DatasiteForML = DatasiteForMannWhitney.copy()
```

```
In [378]: DatasiteForML
```

Out[378]:

	name	continent	Area	pop_est	gdp_md_est	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country	pop_de
0	Fiji	Oceania	1.639511	889953.0	5496	[7]	5.0	1.0	5.428161
6	Uzbekistan	Asia	49.958821	33580650.0	57921	[]	5.0	0.0	6.721666
7	Papua New Guinea	Oceania	37.991091	8776109.0	24829	[7]	5.0	1.0	2.310042
12	Somalia	Africa	39.528186	10192317.3	4719	[7, 18]	5.0	2.0	2.578494
13	Kenya	Africa	48.033195	52573973.0	95503	[2, 7, 18]	5.0	3.0	1.094534
...	...	...	...	...	...	...	...	...	...
170	Bosnia and Herzegovina	Europe	5.696666	3301000.0	20164	[]	5.0	0.0	5.794618
172	Serbia	Europe	8.604719	6944975.0	51475	[]	5.0	0.0	8.071124
173	Montenegro	Europe	1.479321	622137.0	5542	[]	5.0	0.0	4.205559
175	Trinidad and Tobago	North America	0.639000	1394973.0	24269	[]	5.0	0.0	2.183056
176	South Sudan	Africa	51.196106	11062113.0	11998	[7, 18]	5.0	2.0	2.160733

101 rows × 13 columns

In [379...]

```
# Columns to keep and reorder
columns_to_keep = [
    'continent',
    'Area',
    'pop_est',
    'gdp_md_est',
    'Sector_Lists',
    'Total_Sector_By_country',
    'GDP_Category',
    'Armed_Conflict',
    'Priority'
]
# Creating the new DataFrame with the selected columns
DatasiteForML = DatasiteForML[columns_to_keep]
# Resetting the index to start from 0 to 101
```

```
DatasiteForML.reset_index(drop=True, inplace=True)
DatasiteForML.index = pd.RangeIndex(start=0, stop=len(DatasiteForML))
# Displaying the new DataFrame to verify the changes
DatasiteForML.head()
```

Out[379]:

	continent	Area	pop_est	gdp_md_est	Sector_Lists	Total_Sector_By_country	GDP_Category	Armed_Conflict	Priority
0	Oceania	1.639511	889953.0	5496	[7]	1.0	Poor	Yes	1
1	Asia	49.958821	33580650.0	57921	[]	0.0	Middle-income	No	0
2	Oceania	37.991091	8776109.0	24829	[7]	1.0	Middle-income	Yes	1
3	Africa	39.528186	10192317.3	4719	[7, 18]	2.0	Poor	Yes	1
4	Africa	48.033195	52573973.0	95503	[2, 7, 18]	3.0	Middle-income	Yes	1

## Convert categorical data to binary data

In [380...]

```
# Creating dummy variables (converting to binary data)
DatasiteForMLDummies = pd.get_dummies(DatasiteForML, columns=['continent', 'Sector_Lists', 'GDP_Category', 'Armed_Conflict'])
# Convert all dummy variables to integer type
DatasiteForMLDummies = DatasiteForMLDummies.astype(int)
```

In [381...]

```
DatasiteForMLDummies.head()
```

Out[381]:

	Area	pop_est	gdp_md_est	Total_Sector_By_country	Priority	continent_Asia	continent_Europe	continent_North_America	continent_Oceania	cont
0	1	889953	5496		1	1	0	0	0	1
1	49	33580650	57921		0	0	1	0	0	0
2	37	8776109	24829		1	1	0	0	0	1
3	39	10192317	4719		2	1	0	0	0	0
4	48	52573973	95503		3	1	0	0	0	0

In [382...]

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, roc_curve, roc_auc_score
```

```
In [383]: # Prepare data: Split features (X) and target variable (y)
X = DatasiteForMLDummies.drop(columns=['Priority'])
y = DatasiteForMLDummies['Priority']
```

```
In [384]: # Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [385]: # Choose a regression model (Logistic Regression)
model = LogisticRegression()
```

```
In [386]: # Train the model
model.fit(X_train, y_train)
```

```
Out[386]: ▾ LogisticRegression
LogisticRegression()
```

```
In [387]: # Make predictions
y_pred = model.predict(X_test)
```

```
In [388]: # Make predictions
#y_pred = model.predict(X_test)
```

```
In [389]: # Evaluate the model
accuracy = accuracy_score(y_test, y_pred).round(4)*100
report = classification_report(y_test, y_pred)
```

```
In [390]: print("Accuracy:", accuracy)
print("Classification Report:")
print(report)
```

Accuracy: 80.95

Classification Report:

	precision	recall	f1-score	support
0	0.64	1.00	0.78	7
1	1.00	0.71	0.83	14
accuracy			0.81	21
macro avg	0.82	0.86	0.81	21
weighted avg	0.88	0.81	0.81	21

In [391...]

```
# Create a DataFrame to compare actual and predicted values
comparison_df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
comparison_df
```

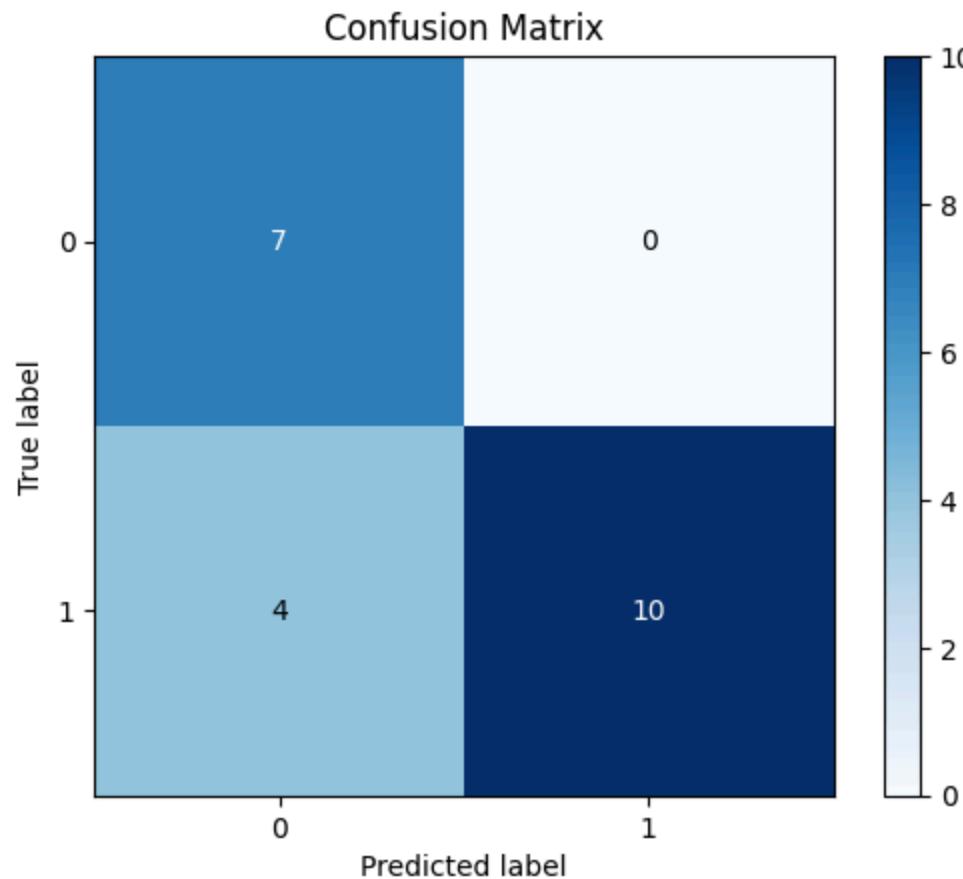
Out[391]:

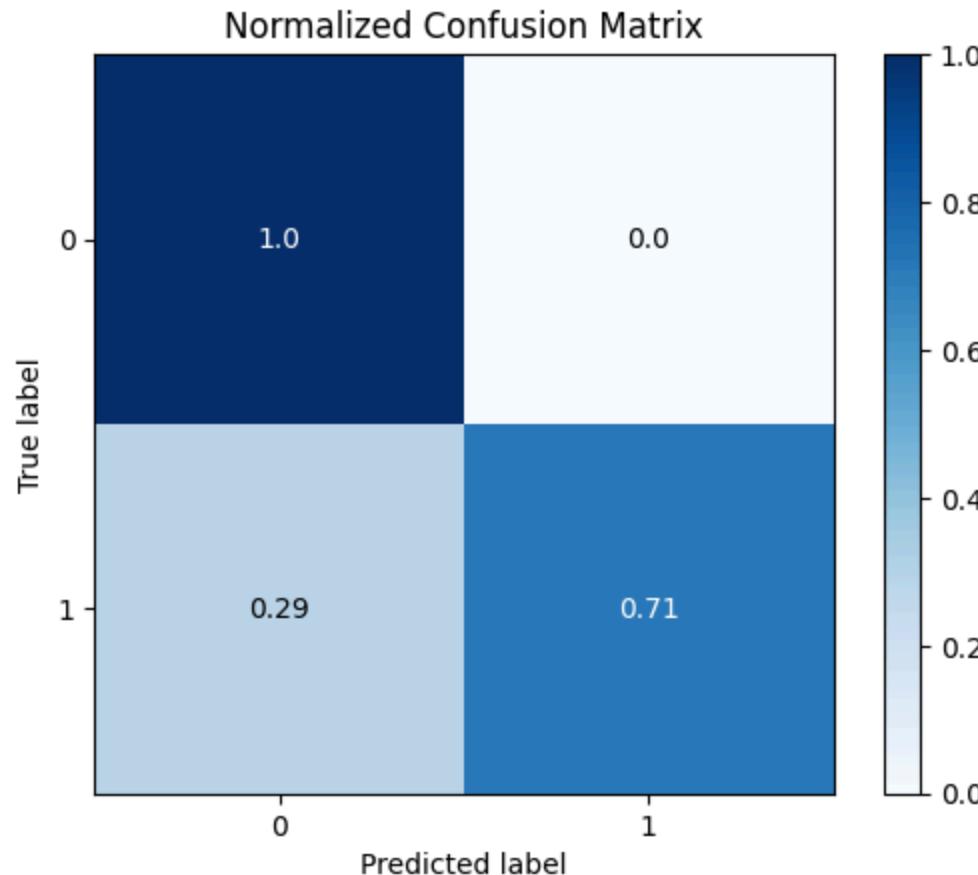
	Actual	Predicted
84	1	1
55	1	0
66	0	0
67	1	0
45	1	1
39	1	1
22	1	1
44	1	1
10	0	0
0	1	0
18	0	0
30	1	1
97	0	0
33	1	1
77	1	1
4	1	1
93	1	0
78	0	0
12	0	0
31	1	1
76	0	0

In [392...]

```
# Create a the confusion matrix
import scikitplot as skplt
skplt.metrics.plot_confusion_matrix(y_test, y_pred)
plt.show()
```

```
skplt.metrics.plot_confusion_matrix(y_test, y_pred, normalize=True) # in percentage  
plt.show()
```





#### Breakdown of the Confusion Matrix:

True Negatives (TN): The number of instances that were correctly predicted as the negative class (Priority = 0). In this case, 7.

True Positives (TP): The number of instances that were correctly predicted as the positive class (Priority = 1). In this case, 10.

False Positives (FP): The number of instances that were incorrectly predicted as the positive class (Priority = 1) but were actually negative (Priority = 0).

In this case, 0.

False Negatives (FN): The number of instances that were incorrectly predicted as the negative class (Priority = 0) but were actually positive (Priority = 1). In this case, 4.

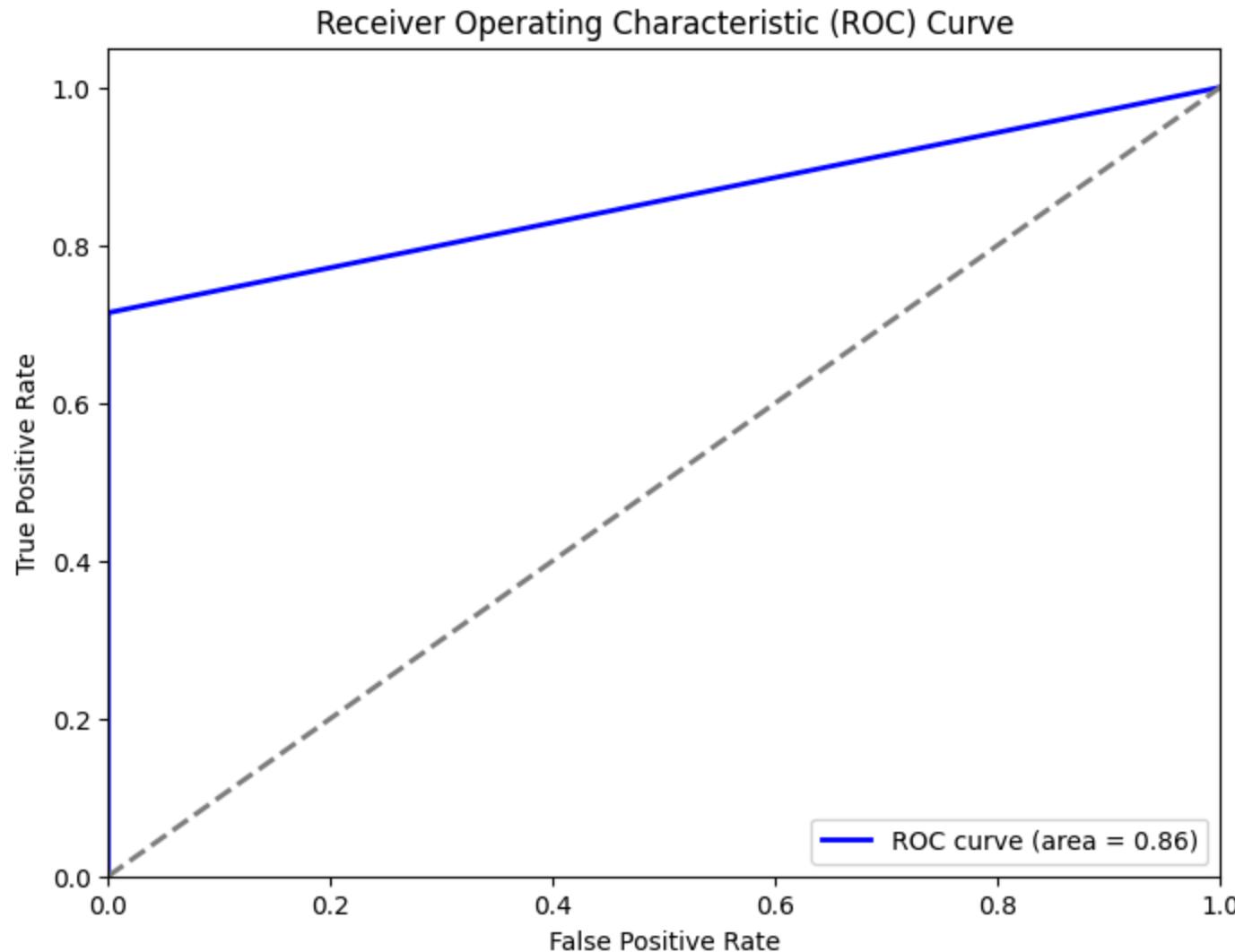
#### Interpretation:

True Negatives (TN = 7): 7 samples were correctly predicted as Priority = 0. False Positives (FP = 0): 0 samples were incorrectly predicted as Priority = 1, but their true label is Priority = 0. False Negatives (FN = 4): 4 samples were incorrectly predicted as Priority = 0, but their true label is Priority = 1. True Positives (TP = 10): 10 samples were correctly predicted as Priority = 1.

In [393...]

```
# Generate the ROC curve
fpr, tpr, thresholds = roc_curve(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_pred)

# Plot the ROC curve
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='blue', lw=2, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='gray', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")
plt.show()
```



The model performs well with an AUC of 0.86, indicating it can effectively distinguish between the classes. The ROC curve shows a good balance between sensitivity and specificity, especially at the initial thresholds. The classification threshold can be adjusted to find an optimal balance between TPR and FPR based on the specific requirements of the application. *Overall, the ROC curve and AUC provide a clear indication that the logistic regression model is effective for the given classification task.*

In [394...]

```
# Calculate AUC
auc = roc_auc_score(y_test, y_pred).round(2)
print("AUC:", auc)
```

AUC: 0.86

"The AUC value of 0.86 indicates that the model performs well in distinguishing between classes. This suggests that the model has a high true positive rate and a low false positive rate, making it effective in classification tasks."

In [395...]

```
import pandas as pd

# Assuming data_clean and geo_base are your DataFrames
# data_clean = pd.read_csv('data_clean.csv')
# geo_base = pd.read_csv('geo_base.csv')

# Perform the left join
merged_df = pd.merge(DataCleanAndPrepareForML, WordGeoPandasBase, on='name', how='left')

# Display the result
merged_df
```

Out[395]:

		name	continent_x	Area	pop_est_x	gdp_md_est_x	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country
0		Fiji	Oceania	1.639511	889953.0	5496	[7]	5.0	1.0
1		United Republic of Tanzania	Africa	76.301964	58005463.0	63177	[2, 7, 18]	5.0	3.0
2		Western Sahara	Africa	8.603984	603253.0	907	[]	2.0	0.0
3		Canada	North America	1712.995228	37589262.0	1736425	[]	5.0	0.0
4		United States	North America	1122.281921	328239523.0	21433226	[]	5.0	0.0
...		...	...	...	...	...	...	...	...
172		Serbia	Europe	8.604719	6944975.0	51475	[]	5.0	0.0
173		Montenegro	Europe	1.479321	622137.0	5542	[]	5.0	0.0
174		Kosovo	Europe	1.231641	1794248.0	7926	[]	1.0	0.0
175		Trinidad and Tobago	North America	0.639000	1394973.0	24269	[]	5.0	0.0

	name	continent_x	Area	pop_est_x	gdp_md_est_x	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country	
176	South Sudan	Africa	51.196106	11062113.0	11998	[7, 18]	5.0	2.0	2

177 rows × 18 columns

In [396...]

```
# Define a dictionary mapping country names to ISO3 codes
iso3_mapping = {
    'Bosnia and Herzegovina': 'BIH',
    'Central African Republic': 'CAF',
    'Democratic People\'s Republic of Korea (North Korea)': 'PRK',
    'Democratic Republic of Congo (Congo - Kinshasa)': 'COD',
    'Dominican Republic': 'DOM',
    'Equatorial Guinea': 'GNQ',
    'Eswatini (Swaziland)': 'SWZ',
    'Falkland Islands (Malvinas)': 'FLK',
    'French Southern and Antarctic Lands': 'ATF',
    'Lao People\'s Democratic Republic (Laos)': 'LAO',
    'Republic of South Korea (South Korea)': 'KOR',
    'Republic of Türkiye (Turkey)': 'TUR',
    'Russian Federation (Russia)': 'RUS',
    'Solomon Islands': 'SLB',
    'South Sudan': 'SSD',
    'Syrian Arab Republic (Syria)': 'SYR',
    'United Republic of Tanzania (Tanzania)': 'TZA',
    'United States': 'USA',
    'Viet Nam (Vietnam)': 'VNM',
    'Western Sahara': 'ESH'
}

# Fill NaN values in iso_a3 column based on the name matching
merged_df['iso_a3'] = merged_df.apply(lambda row: iso3_mapping.get(row['name'], row['iso_a3']), axis=1)
```

In [397...]

merged\_df

Out[397]:

	name	continent_x	Area	pop_est_x	gdp_md_est_x	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country
0	Fiji	Oceania	1.639511	889953.0	5496	[7]	5.0	1.0
1	United Republic of Tanzania	Africa	76.301964	58005463.0	63177	[2, 7, 18]	5.0	3.0
2	Western Sahara	Africa	8.603984	603253.0	907	[]	2.0	0.0
3	Canada	North America	1712.995228	37589262.0	1736425	[]	5.0	0.0
4	United States	North America	1122.281921	328239523.0	21433226	[]	5.0	0.0
...	...	...	...	...	...	...	...	...
172	Serbia	Europe	8.604719	6944975.0	51475	[]	5.0	0.0
173	Montenegro	Europe	1.479321	622137.0	5542	[]	5.0	0.0
174	Kosovo	Europe	1.231641	1794248.0	7926	[]	1.0	0.0
175	Trinidad and Tobago	North America	0.639000	1394973.0	24269	[]	5.0	0.0

	name	continent_x	Area	pop_est_x	gdp_md_est_x	Sector_Lists	Total_Organization_By_country	Total_Sector_By_country	
176	South Sudan	Africa	51.196106	11062113.0	11998	[7, 18]	5.0	2.0	2

177 rows × 18 columns

In [398...]

`merged_df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 177 entries, 0 to 176
Data columns (total 18 columns):
 #   Column           Non-Null Count  Dtype  
---  --  
 0   name             177 non-null    object  
 1   continent_x     177 non-null    object  
 2   Area              177 non-null    float64 
 3   pop_est_x        177 non-null    float64 
 4   gdp_md_est_x    177 non-null    int64  
 5   Sector_Lists      177 non-null    object  
 6   Total_Organization_By_country  177 non-null    float64 
 7   Total_Sector_By_country    177 non-null    float64 
 8   pop_density       177 non-null    float64 
 9   GDP_Category      177 non-null    category 
 10  Priority          177 non-null    object  
 11  Organization       177 non-null    object  
 12  Armed_Conflict     177 non-null    object  
 13  pop_est_y         157 non-null    float64 
 14  continent_y       157 non-null    object  
 15  iso_a3            167 non-null    object  
 16  gdp_md_est_y     157 non-null    float64 
 17  geometry           157 non-null    geometry 
```

dtypes: category(1), float64(7), geometry(1), int64(1), object(8)  
memory usage: 23.9+ KB

In [400...]

```
# print merged_df as excel file to drive

import pandas as pd
from google.colab import drive
# Convert the DataFrame to an Excel file
merged_df.to_excel('merged_df.xlsx', index=False)

# Mount Google Drive
drive.mount('/content/drive')
```

```
# Define the destination path on Google Drive
destination_path = '/content/drive/MyDrive/merged_df.xlsx'

# Copy the Excel file to Google Drive
!cp merged_df.xlsx '{destination_path}'
```

Mounted at /content/drive