DSC540-T301_2245_1 Data Preparation

Assignment Week 9 & 10 Term Project Milstone 4;

Author: Zemelak Goraga;

Date: 5/18/2024

Milstone 4: Connecting to an API/Pulling in the Data and Cleaning/Formatting

The dataset:

In this milstone 4 of the term project, the FAOSTAT historical dataset on LiveAnimals was obtained from Kaggle using this API Command (kaggle datasets download -d unitednations/global-food-agriculture-statistics) following the undermentioned procedures. The dataset represent over 200 countries with more than 25 primary products and inputs that were collected in between 1961 to 2013 years. Key variables include in the dataset are Area or Country, Item (Agricult.Products, Cattle, Sheep, Chicken, Crops, etc), Element (Import Quantity, Export Quantity, Import Value, Export Value), Year (1961 – 2013), and Value.

Step 1: Connecting to an API/Pulling in the Data and Cleaning/Formatting

In [1]:
```
# Import required libraries
import subprocess
import os
import zipfile
import pandas as pd
from zipfile import ZipFile
import warnings
warnings.filterwarnings('ignore')
```

In [14]:
```
# Execute the Kaggle API command to download the dataset
command = "kaggle datasets download -d unitednations/global-food-agriculture-statis
subprocess.run(command.split())
```

Out[14]:
```
CompletedProcess(args=['kaggle', 'datasets', 'download', '-d', 'unitednations/glob
al-food-agriculture-statistics'], returncode=0)
```

In [108…
```
# Step 2: Check if the download was successful
if os.path.exists("global-food-agriculture-statistics.zip"):
    print("Dataset downloaded successfully!")
```

```
Dataset downloaded successfully!
```

In [109…
```
# Step 3: Unzip the downloaded file
with zipfile.ZipFile("global-food-agriculture-statistics.zip", "r") as zip_ref:
```

```
    zip_ref.extractall("data")
```

In [110…

```
# Step 4: Optionally, list the contents of the extracted directory
extracted_files = os.listdir("data")
print("Extracted files:", extracted_files)
```

```
Extracted files: ['car_price8.json', 'countries.csv', 'countries.json', 'current_F
AO', 'fao_data_crops_data.csv', 'fao_data_fertilizers_data.csv', 'fao_data_forest_
data.csv', 'fao_data_land_data.csv', 'fao_data_production_indices_data.csv', 'LICE
NSE.md', 'nutrients.json', 'original', 'README.md', 'README.txt', 'test', 'train_v
alidate', 'us-usda-gov-3d6e899c-ce46-4998-ab3f-ad228d7f3888', 'words.txt', 'words_
alpha.txt', 'words_dictionary.json']
```

In [111…

```
    # Step 5: Download a specific table to work with
    # Specify the CSV file to read from the ZIP archive
    csv_file_to_read = "current_FAO/raw_files/Trade_LiveAnimals_E_All_Data_(Normali

    # Read the ZIP archive
    with ZipFile("global-food-agriculture-statistics.zip", 'r') as zip_file:
        # List the files within the ZIP archive (to double-check paths)
        print(zip_file.namelist())

        # Read the CSV file from the ZIP archive with the specified encoding and de
        with zip_file.open(csv_file_to_read) as csv_file:
            df = pd.read_csv(csv_file, encoding='ISO-8859-1')
```

['current_FAO/__MACOSX/raw_files/._ASTI_Research_Spending_E_All_Data_(Norm).csv',
'current_FAO/__MACOSX/raw_files/._ASTI_Researchers_E_All_Data_(Norm).csv', 'curren
t_FAO/__MACOSX/raw_files/._CommodityBalances_Crops_E_All_Data_(Normalized).csv',
'current_FAO/__MACOSX/raw_files/._CommodityBalances_LivestockFish_E_All_Data_(Norm
alized).csv', 'current_FAO/__MACOSX/raw_files/._ConsumerPriceIndices_E_All_Data_(N
ormalized).csv', 'current_FAO/__MACOSX/raw_files/._Deflators_E_All_Data_(Normalize
d).csv', 'current_FAO/__MACOSX/raw_files/._Emissions_Agriculture_Agriculture_total
_E_All_Data_(Norm).csv', 'current_FAO/__MACOSX/raw_files/._Emissions_Agriculture_B
urning_Savanna_E_All_Data_(Norm).csv', 'current_FAO/__MACOSX/raw_files/._Emissions
_Agriculture_Burning_crop_residues_E_All_Data_(Norm).csv', 'current_FAO/__MACOSX/r
aw_files/._Emissions_Agriculture_Crop_Residues_E_All_Data_(Norm).csv', 'current_FA
O/__MACOSX/raw_files/._Emissions_Agriculture_Cultivated_Organic_Soils_E_All_Data_
(Norm).csv', 'current_FAO/__MACOSX/raw_files/._Emissions_Agriculture_Energy_E_All_
Data_(Norm).csv', 'current_FAO/__MACOSX/raw_files/._Emissions_Agriculture_Enteric_
Fermentation_E_All_Data_(Norm).csv', 'current_FAO/__MACOSX/raw_files/._Emissions_A
griculture_Manure_Management_E_All_Data_(Norm).csv', 'current_FAO/__MACOSX/raw_fil
es/._Emissions_Agriculture_Manure_applied_to_soils_E_All_Data_(Norm).csv', 'curren
t_FAO/__MACOSX/raw_files/._Emissions_Agriculture_Manure_left_on_pasture_E_All_Data
_(Norm).csv', 'current_FAO/__MACOSX/raw_files/._Emissions_Agriculture_Rice_Cultiva
tion_E_All_Data_(Norm).csv', 'current_FAO/__MACOSX/raw_files/._Emissions_Agricultu
re_Synthetic_Fertilizers_E_All_Data_(Norm).csv', 'current_FAO/__MACOSX/raw_files/.
_Emissions_Land_Use_Burning_Biomass_E_All_Data_(Norm).csv', 'current_FAO/__MACOSX/
raw_files/._Emissions_Land_Use_Cropland_E_All_Data_(Norm).csv', 'current_FAO/__MAC
OSX/raw_files/._Emissions_Land_Use_Forest_Land_E_All_Data_(Norm).csv', 'current_FA
O/__MACOSX/raw_files/._Emissions_Land_Use_Grassland_E_All_Data_(Norm).csv', 'curre
nt_FAO/__MACOSX/raw_files/._Emissions_Land_Use_Land_Use_Total_E_All_Data_(Norm).cs
v', 'current_FAO/__MACOSX/raw_files/._Employment_Indicators_E_All_Data_(Norm).cs
v', 'current_FAO/__MACOSX/raw_files/._Environment_AirClimateChange_E_All_Data.cs
v', 'current_FAO/__MACOSX/raw_files/._Environment_Emissions_by_Sector_E_All_Data_
(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Environment_Emissions_intensi
ties_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Environment_E
nergy_E_All_Data.csv', 'current_FAO/__MACOSX/raw_files/._Environment_Fertilizers_E
_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Environment_LandCov
er_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Environment_Lan
dUse_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Environment_L
ivestockPatterns_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._E
nvironment_Livestock_E_All_Data.csv', 'current_FAO/__MACOSX/raw_files/._Environmen
t_Pesticides_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Envir
onment_Soil_E_All_Data.csv', 'current_FAO/__MACOSX/raw_files/._Environment_Tempera
ture_change_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Enviro
nment_Water_E_All_Data.csv', 'current_FAO/__MACOSX/raw_files/._Exchange_rate_E_All
Data(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._FoodSupply_Crops_E_All_
Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._FoodSupply_LivestockFish
_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Food_Aid_Shipment
s_WFP_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Food_Securit
y_Data_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Forestry_E_
All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Forestry_Trade_Flows
_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Indicators_from_H
ousehold_Surveys_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._I
nputs_FertilizersTradeValues_E_All_Data_(Norm).csv', 'current_FAO/__MACOSX/raw_fil
es/._Inputs_Fertilizers_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_fi
les/._Inputs_Land_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._
Inputs_Pesticides_Trade_E_All_Data_(Norm).csv', 'current_FAO/__MACOSX/raw_files/._
Inputs_Pesticides_Use_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_file
s/._Investment_CapitalStock_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/ra
w_files/._Investment_CountryInvestmentStatisticsProfile__E_All_Data_(Normalized).c
sv', 'current_FAO/__MACOSX/raw_files/._Investment_CreditAgriculture_E_All_Data_(No
rmalized).csv', 'current_FAO/__MACOSX/raw_files/._Investment_ForeignDirectInvestme
nt_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Investment_Gove
rnmentExpenditure_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._
Investment_MachineryArchive_E_All_Data.csv', 'current_FAO/__MACOSX/raw_files/._Inv
estment_Machinery_E_All_Data.csv', 'current_FAO/__MACOSX/raw_files/._Macro-Statist
ics_Key_Indicators_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/.
_Population_E_All_Data_(Norm).csv', 'current_FAO/__MACOSX/raw_files/._Population_E

_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Price_Indices_E_All
Data(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._PricesArchive_E_All_Dat
a.csv', 'current_FAO/__MACOSX/raw_files/._Prices_E_All_Data_(Normalized).csv', 'cu
rrent_FAO/__MACOSX/raw_files/._Prices_Monthly_E_All_Data_(Normalized).csv', 'curre
nt_FAO/__MACOSX/raw_files/._Production_CropsProcessed_E_All_Data_(Normalized).cs
v', 'current_FAO/__MACOSX/raw_files/._Production_Crops_E_All_Data_(Normalized).cs
v', 'current_FAO/__MACOSX/raw_files/._Production_Indices_E_All_Data_(Normalized).c
sv', 'current_FAO/__MACOSX/raw_files/._Production_LivestockPrimary_E_All_Data_(Nor
malized).csv', 'current_FAO/__MACOSX/raw_files/._Production_LivestockProcessed_E_A
ll_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Production_Livestock_
E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Resources_Fertiliz
ersArchive_E_All_Data.csv', 'current_FAO/__MACOSX/raw_files/._Trade_Crops_Livestoc
k_E_All_Data_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Trade_Indices_E_
All_Data_(Norm).csv', 'current_FAO/__MACOSX/raw_files/._Trade_LiveAnimals_E_All_Da
ta_(Normalized).csv', 'current_FAO/__MACOSX/raw_files/._Value_of_Production_E_All_
Data_(Normalized).csv', 'current_FAO/raw_files/ASTI_Research_Spending_E_All_Data_
(Norm).csv', 'current_FAO/raw_files/ASTI_Researchers_E_All_Data_(Norm).csv', 'curr
ent_FAO/raw_files/CommodityBalances_Crops_E_All_Data_(Normalized).csv', 'current_F
AO/raw_files/CommodityBalances_LivestockFish_E_All_Data_(Normalized).csv', 'curren
t_FAO/raw_files/ConsumerPriceIndices_E_All_Data_(Normalized).csv', 'current_FAO/ra
w_files/Deflators_E_All_Data_(Normalized).csv', 'current_FAO/raw_files/Emissions_A
griculture_Agriculture_total_E_All_Data_(Norm).csv', 'current_FAO/raw_files/Emissi
ons_Agriculture_Burning_Savanna_E_All_Data_(Norm).csv', 'current_FAO/raw_files/Emi
ssions_Agriculture_Burning_crop_residues_E_All_Data_(Norm).csv', 'current_FAO/raw_
files/Emissions_Agriculture_Crop_Residues_E_All_Data_(Norm).csv', 'current_FAO/raw
_files/Emissions_Agriculture_Cultivated_Organic_Soils_E_All_Data_(Norm).csv', 'cur
rent_FAO/raw_files/Emissions_Agriculture_Energy_E_All_Data_(Norm).csv', 'current_F
AO/raw_files/Emissions_Agriculture_Enteric_Fermentation_E_All_Data_(Norm).csv', 'c
urrent_FAO/raw_files/Emissions_Agriculture_Manure_Management_E_All_Data_(Norm).cs
v', 'current_FAO/raw_files/Emissions_Agriculture_Manure_applied_to_soils_E_All_Dat
a_(Norm).csv', 'current_FAO/raw_files/Emissions_Agriculture_Manure_left_on_pasture
_E_All_Data_(Norm).csv', 'current_FAO/raw_files/Emissions_Agriculture_Rice_Cultiva
tion_E_All_Data_(Norm).csv', 'current_FAO/raw_files/Emissions_Agriculture_Syntheti
c_Fertilizers_E_All_Data_(Norm).csv', 'current_FAO/raw_files/Emissions_Land_Use_Bu
rning_Biomass_E_All_Data_(Norm).csv', 'current_FAO/raw_files/Emissions_Land_Use_Cr
opland_E_All_Data_(Norm).csv', 'current_FAO/raw_files/Emissions_Land_Use_Forest_La
nd_E_All_Data_(Norm).csv', 'current_FAO/raw_files/Emissions_Land_Use_Grassland_E_A
ll_Data_(Norm).csv', 'current_FAO/raw_files/Emissions_Land_Use_Land_Use_Total_E_Al
l_Data_(Norm).csv', 'current_FAO/raw_files/Employment_Indicators_E_All_Data_(Nor
m).csv', 'current_FAO/raw_files/Environment_AirClimateChange_E_All_Data.csv', 'cur
rent_FAO/raw_files/Environment_Emissions_by_Sector_E_All_Data_(Normalized).csv',
'current_FAO/raw_files/Environment_Emissions_intensities_E_All_Data_(Normalized).c
sv', 'current_FAO/raw_files/Environment_Energy_E_All_Data.csv', 'current_FAO/raw_f
iles/Environment_Fertilizers_E_All_Data_(Normalized).csv', 'current_FAO/raw_files/
Environment_LandCover_E_All_Data_(Normalized).csv', 'current_FAO/raw_files/Environ
ment_LandUse_E_All_Data_(Normalized).csv', 'current_FAO/raw_files/Environment_Live
stockPatterns_E_All_Data_(Normalized).csv', 'current_FAO/raw_files/Environment_Liv
estock_E_All_Data.csv', 'current_FAO/raw_files/Environment_Pesticides_E_All_Data_
(Normalized).csv', 'current_FAO/raw_files/Environment_Soil_E_All_Data.csv', 'curre
nt_FAO/raw_files/Environment_Temperature_change_E_All_Data_(Normalized).csv', 'cur
rent_FAO/raw_files/Environment_Water_E_All_Data.csv', 'current_FAO/raw_files/Excha
nge_rate_E_All_Data_(Normalized).csv', 'current_FAO/raw_files/FoodSupply_Crops_E_A
ll_Data_(Normalized).csv', 'current_FAO/raw_files/FoodSupply_LivestockFish_E_All_D
ata_(Normalized).csv', 'current_FAO/raw_files/Food_Aid_Shipments_WFP_E_All_Data_(N
ormalized).csv', 'current_FAO/raw_files/Food_Security_Data_E_All_Data_(Normalize
d).csv', 'current_FAO/raw_files/Forestry_E_All_Data_(Normalized).csv', 'current_FA
O/raw_files/Forestry_Trade_Flows_E_All_Data_(Normalized).csv', 'current_FAO/raw_fi
les/Indicators_from_Household_Surveys_E_All_Data_(Normalized).csv', 'current_FAO/r
aw_files/Inputs_FertilizersTradeValues_E_All_Data_(Norm).csv', 'current_FAO/raw_fi
les/Inputs_Fertilizers_E_All_Data_(Normalized).csv', 'current_FAO/raw_files/Inputs
_Land_E_All_Data_(Normalized).csv', 'current_FAO/raw_files/Inputs_Pesticides_Trade
_E_All_Data_(Norm).csv', 'current_FAO/raw_files/Inputs_Pesticides_Use_E_All_Data_
(Normalized).csv', 'current_FAO/raw_files/Investment_CapitalStock_E_All_Data_(Norm
alized).csv', 'current_FAO/raw_files/Investment_CountryInvestmentStatisticsProfile

```
__E_All_Data_(Normalized).csv', 'current_FAO/raw_files/Investment_CreditAgricultur
e_E_All_Data_(Normalized).csv', 'current_FAO/raw_files/Investment_ForeignDirectInv
estment_E_All_Data_(Normalized).csv', 'current_FAO/raw_files/Investment_Government
Expenditure_E_All_Data_(Normalized).csv', 'current_FAO/raw_files/Investment_Machin
eryArchive_E_All_Data.csv', 'current_FAO/raw_files/Investment_Machinery_E_All_Dat
a.csv', 'current_FAO/raw_files/Macro-Statistics_Key_Indicators_E_All_Data_(Normali
zed).csv', 'current_FAO/raw_files/Population_E_All_Data_(Norm).csv', 'current_FAO/
raw_files/Population_E_All_Data_(Normalized).csv', 'current_FAO/raw_files/Price_In
dices_E_All_Data_(Normalized).csv', 'current_FAO/raw_files/PricesArchive_E_All_Dat
a.csv', 'current_FAO/raw_files/Prices_E_All_Data_(Normalized).csv', 'current_FAO/r
aw_files/Prices_Monthly_E_All_Data_(Normalized).csv', 'current_FAO/raw_files/Produ
ction_CropsProcessed_E_All_Data_(Normalized).csv', 'current_FAO/raw_files/Producti
on_Crops_E_All_Data_(Normalized).csv', 'current_FAO/raw_files/Production_Indices_E
_All_Data_(Normalized).csv', 'current_FAO/raw_files/Production_LivestockPrimary_E_
All_Data_(Normalized).csv', 'current_FAO/raw_files/Production_LivestockProcessed_E
_All_Data_(Normalized).csv', 'current_FAO/raw_files/Production_Livestock_E_All_Dat
a_(Normalized).csv', 'current_FAO/raw_files/Resources_FertilizersArchive_E_All_Dat
a.csv', 'current_FAO/raw_files/Trade_Crops_Livestock_E_All_Data_(Normalized).csv',
'current_FAO/raw_files/Trade_Indices_E_All_Data_(Norm).csv', 'current_FAO/raw_file
s/Trade_LiveAnimals_E_All_Data_(Normalized).csv', 'current_FAO/raw_files/Value_of_
Production_E_All_Data_(Normalized).csv', 'fao_data_crops_data.csv', 'fao_data_fert
ilizers_data.csv', 'fao_data_forest_data.csv', 'fao_data_land_data.csv', 'fao_data
_production_indices_data.csv']
```

In [119…

```
# Print the first few rows of the dataset
df.head()
```

Out[119]:

	Area Code	Area	Item Code	Item	Element Code	Element	Year Code	Year	Unit	Value	Flag
0	2	Afghanistan	866	Cattle	5608	Import Quantity	1961	1961	Head	NaN	M
1	2	Afghanistan	866	Cattle	5608	Import Quantity	1962	1962	Head	NaN	M
2	2	Afghanistan	866	Cattle	5608	Import Quantity	1963	1963	Head	NaN	M
3	2	Afghanistan	866	Cattle	5608	Import Quantity	1964	1964	Head	NaN	M
4	2	Afghanistan	866	Cattle	5608	Import Quantity	1965	1965	Head	NaN	M

In [120…

```
# Print the last few rows of the dataset
df.tail()
```

Out[120]:

	Area Code	Area	Item Code	Item	Element Code	Element	Year Code	Year	Unit	Value	Flag
662953	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2009	2009	1000 US$	456293.0	A
662954	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2010	2010	1000 US$	421311.0	A
662955	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2011	2011	1000 US$	649321.0	A
662956	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2012	2012	1000 US$	778317.0	A
662957	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2013	2013	1000 US$	1038636.0	A

In [121…

```
# Copying 'df' to 'df2' # keep 'df' as the original version and here on use df2
df2 = df.copy()

df2
```

Out[121]:

	Area Code	Area	Item Code	Item	Element Code	Element	Year Code	Year	Unit	Value	Flag
0	2	Afghanistan	866	Cattle	5608	Import Quantity	1961	1961	Head	NaN	M
1	2	Afghanistan	866	Cattle	5608	Import Quantity	1962	1962	Head	NaN	M
2	2	Afghanistan	866	Cattle	5608	Import Quantity	1963	1963	Head	NaN	M
3	2	Afghanistan	866	Cattle	5608	Import Quantity	1964	1964	Head	NaN	M
4	2	Afghanistan	866	Cattle	5608	Import Quantity	1965	1965	Head	NaN	M
...	...	...	...	...	...	...	...	...	...	...	...
662953	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2009	2009	1000 US$	456293.0	A
662954	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2010	2010	1000 US$	421311.0	A
662955	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2011	2011	1000 US$	649321.0	A
662956	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2012	2012	1000 US$	778317.0	A
662957	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2013	2013	1000 US$	1038636.0	A

662958 rows × 11 columns

In [122…

```
df2.tail()
```

Out[122]:

	Area Code	Area	Item Code	Item	Element Code	Element	Year Code	Year	Unit	Value	Flag
662953	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2009	2009	1000 US$	456293.0	A
662954	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2010	2010	1000 US$	421311.0	A
662955	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2011	2011	1000 US$	649321.0	A
662956	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2012	2012	1000 US$	778317.0	A
662957	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2013	2013	1000 US$	1038636.0	A

Perform at least 5 data transformation and/or cleansing steps to your API data.

In [123…

```
# Step 1: Replace Headers
new_headers = ["area_code","area", "item_code", "item", "element_code", "element",
df2.columns = new_headers
df2
```

Out[123]:

	area_code	area	item_code	item	element_code	element	year_code	year	unit
0	2	Afghanistan	866	Cattle	5608	Import Quantity	1961	1961	Heac
1	2	Afghanistan	866	Cattle	5608	Import Quantity	1962	1962	Heac
2	2	Afghanistan	866	Cattle	5608	Import Quantity	1963	1963	Heac
3	2	Afghanistan	866	Cattle	5608	Import Quantity	1964	1964	Heac
4	2	Afghanistan	866	Cattle	5608	Import Quantity	1965	1965	Heac
...	...	...	...	...	...	...	...	...	..
662953	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2009	2009	1000 US$
662954	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2010	2010	1000 US$
662955	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2011	2011	1000 US$
662956	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2012	2012	1000 US$
662957	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2013	2013	1000 US$

662958 rows × 11 columns

In [124…

```
# renaming 'area' and 'item' columns

# Renaming columns 'area' to 'country' and 'item' to 'animal_category'
df2 = df2.rename(columns={'area': 'country', 'item': 'animal_category'})

df2.head()
```

Out[124]:

	area_code	country	item_code	animal_category	element_code	element	year_code	year
0	2	Afghanistan	866	Cattle	5608	Import Quantity	1961	1961
1	2	Afghanistan	866	Cattle	5608	Import Quantity	1962	1962
2	2	Afghanistan	866	Cattle	5608	Import Quantity	1963	1963
3	2	Afghanistan	866	Cattle	5608	Import Quantity	1964	1964
4	2	Afghanistan	866	Cattle	5608	Import Quantity	1965	1965

In [125…

```
df2.tail()
```

Out[125]:

	area_code	country	item_code	animal_category	element_code	element	year_code	y
662953	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2009	2
662954	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2010	2
662955	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2011	2
662956	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2012	2
662957	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2013	2

In [126…

```
# data types
print(df2.dtypes)
```

```
area_code           int64
country            object
item_code           int64
animal_category    object
element_code        int64
element            object
year_code           int64
year                int64
unit               object
value             float64
flag               object
dtype: object
```

In [127…

```
# Step 2: Handling Missing Values
missing_values = df2.isnull().sum()
print("Missing values:\n", missing_values)
```

```
Missing values:
 area_code               0
country                  0
item_code                0
animal_category          0
element_code             0
element                  0
year_code                0
year                     0
unit                     0
value               135190
flag                203064
dtype: int64
```

In [128…

```
# Step 3:Data Transformation: creating new variables using the 'gdp' variable

import pandas as pd  # Importing the pandas library and aliasing it as 'pd'
import numpy as np  # Importing the numpy library and aliasing it as 'np'

# Step 3: Data Transformation: creating new variables using the 'value' variable

# Convert 'value' column to numeric
df2['value'] = pd.to_numeric(df2['value'], errors='coerce')

# Create new variables
df2['value_squared'] = df2['value'] ** 2  # Creating a new column 'value_squared' w
df2['value_square_root'] = np.sqrt(df2['value'])  # Creating a new column 'value_sq
df2['value_log'] = np.log(df2['value'])  # Creating a new column 'value_log' which
df2['value_zscore'] = (df2['value'] - df2['value'].mean()) / df2['value'].std()  #

# Min-max normalization
min_value = df2['value'].min()  # Finding the minimum value of the 'value' column
max_value = df2['value'].max()  # Finding the maximum value of the 'value' column
df2['value_normalized'] = (df2['value'] - min_value) / (max_value - min_value)  # C

#print(df2[['country', 'animal_category', 'year', 'value', 'value_squared', 'value_

df2[['country', 'animal_category', 'element', 'year', 'value', 'value_squared', 'va
```

Out[128]:

	country	animal_category	element	year	value	value_squared	value_square_root
0	Afghanistan	Cattle	Import Quantity	1961	NaN	NaN	NaN
1	Afghanistan	Cattle	Import Quantity	1962	NaN	NaN	NaN
2	Afghanistan	Cattle	Import Quantity	1963	NaN	NaN	NaN
3	Afghanistan	Cattle	Import Quantity	1964	NaN	NaN	NaN
4	Afghanistan	Cattle	Import Quantity	1965	NaN	NaN	NaN
...	...	...	...	...	...	...	...
662953	Net Food Importing Developing Countries	Sheep and Goats	Export Value	2009	456293.0	2.082033e+11	675.494634
662954	Net Food Importing Developing Countries	Sheep and Goats	Export Value	2010	421311.0	1.775030e+11	649.084740
662955	Net Food Importing Developing Countries	Sheep and Goats	Export Value	2011	649321.0	4.216178e+11	805.804567
662956	Net Food Importing Developing Countries	Sheep and Goats	Export Value	2012	778317.0	6.057774e+11	882.222761
662957	Net Food Importing Developing Countries	Sheep and Goats	Export Value	2013	1038636.0	1.078765e+12	1019.134927

662958 rows × 10 columns

In [129…

```
import pandas as pd

# Convert 'value' and its derived new variables to numeric using the given formula
df2['value'] = pd.to_numeric(df2['value'], errors='coerce')
df2['value_squared'] = pd.to_numeric(df2['value_squared'], errors='coerce')
df2['value_square_root'] = pd.to_numeric(df2['value_square_root'], errors='coerce')
df2['value_log'] = pd.to_numeric(df2['value_log'], errors='coerce')
df2['value_zscore'] = pd.to_numeric(df2['value_zscore'], errors='coerce')
df2['value_normalized'] = pd.to_numeric(df2['value_normalized'], errors='coerce')
df2
```

Out[129]:

	area_code	country	item_code	animal_category	element_code	element	year_code	y
0	2	Afghanistan	866	Cattle	5608	Import Quantity	1961	1
1	2	Afghanistan	866	Cattle	5608	Import Quantity	1962	1
2	2	Afghanistan	866	Cattle	5608	Import Quantity	1963	1
3	2	Afghanistan	866	Cattle	5608	Import Quantity	1964	1
4	2	Afghanistan	866	Cattle	5608	Import Quantity	1965	1
...	...	...	...	...	...	...	...	
662953	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2009	2
662954	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2010	2
662955	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2011	2
662956	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2012	2
662957	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2013	2

662958 rows × 16 columns

In [130…

```
df2.tail()
```

Out[130]:

	area_code	country	item_code	animal_category	element_code	element	year_code	y
662953	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2009	2
662954	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2010	2
662955	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2011	2
662956	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2012	2
662957	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2013	2

In [ ]:

In [131…

```
# Step 6: There are still some some 'NaN' and 'None' values in the dataset, let rem

# Replace 'None' values with NaN
df2.replace('None', np.nan, inplace=True)

# Remove rows with NaN values
df2.dropna(inplace=True)

# Reset index after dropping rows
df2.reset_index(drop=True, inplace=True)

# Display the cleaned DataFrame
print("DataFrame after removing NaN and None values:")
df2
```

```
DataFrame after removing NaN and None values:
```

Out[131]:

	area_code	country	item_code	animal_category	element_code	element	year_code	y
0	2	Afghanistan	866	Cattle	5608	Import Quantity	2000	2
1	2	Afghanistan	866	Cattle	5608	Import Quantity	2001	2
2	2	Afghanistan	866	Cattle	5608	Import Quantity	2002	2
3	2	Afghanistan	866	Cattle	5608	Import Quantity	2003	2
4	2	Afghanistan	866	Cattle	5608	Import Quantity	2004	2
...	...	...	...	...	...	...	...	
324699	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2009	2
324700	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2010	2
324701	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2011	2
324702	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2012	2
324703	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2013	2

In [132...

```
df2.tail()
```

Out[132]:

	area_code	country	item_code	animal_category	element_code	element	year_code	y
324699	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2009	2
324700	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2010	2
324701	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2011	2
324702	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2012	2
324703	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2013	2

In [ ]:

In [133…

```
# Step 5: Format Data

# Format 'value' columns into a readable format (e.g., adding commas for thousands
df2['value'] = df2['value'].apply(lambda x: '{:,.2f}'.format(x) if isinstance(x, (f
df2['value_squared'] = df2['value_squared'].apply(lambda x: '{:,.2f}'.format(x) if
df2['value_square_root'] = df2['value_square_root'].apply(lambda x: '{:,.2f}'.forma
df2['value_log'] = df2['value_log'].apply(lambda x: '{:,.2f}'.format(x) if isinstar
df2['value_zscore'] = df2['value_zscore'].apply(lambda x: '{:,.2f}'.format(x) if is
df2['value_normalized'] = df2['value_normalized'].apply(lambda x: '{:,.2f}'.format(

df2
```

Out[133]:

	area_code	country	item_code	animal_category	element_code	element	year_code	y
0	2	Afghanistan	866	Cattle	5608	Import Quantity	2000	2
1	2	Afghanistan	866	Cattle	5608	Import Quantity	2001	2
2	2	Afghanistan	866	Cattle	5608	Import Quantity	2002	2
3	2	Afghanistan	866	Cattle	5608	Import Quantity	2003	2
4	2	Afghanistan	866	Cattle	5608	Import Quantity	2004	2
...	...	...	...	...	...	...	...	
324699	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2009	2
324700	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2010	2
324701	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2011	2
324702	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2012	2
324703	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2013	2

324704 rows × 16 columns

In [134…

```
df2.tail()
```

Out[134]:

	area_code	country	item_code	animal_category	element_code	element	year_code	y
324699	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2009	2
324700	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2010	2
324701	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2011	2
324702	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2012	2
324703	5817	Net Food Importing Developing Countries	1922	Sheep and Goats	5922	Export Value	2013	2

In [67]:

```
# Outlier detection

import pandas as pd

# Step 7: Converting each variable to numeric

# Convert 'value' and its derived new variables to numeric using the given formula
df2['value'] = pd.to_numeric(df2['value'].str.replace(',', ''), errors='coerce')
df2['value_squared'] = pd.to_numeric(df2['value_squared'].str.replace(',', ''), err
df2['value_square_root'] = pd.to_numeric(df2['value_square_root'], errors='coerce')
df2['value_log'] = pd.to_numeric(df2['value_log'], errors='coerce')
df2['value_zscore'] = pd.to_numeric(df2['value_zscore'], errors='coerce')
df2['value_normalized'] = pd.to_numeric(df2['value_normalized'], errors='coerce')

# Calculate z-score for the 'value' column
z_scores_value = ((df2['value'] - df2['value'].mean()) / df2['value'].std()).abs()
outliers_value = z_scores_value > 3

# Calculate z-score for the 'value_squared' column
z_scores_value_squared = ((df2['value_squared'] - df2['value_squared'].mean()) / df
outliers_value_squared = z_scores_value_squared > 3

# Print outliers for each variable
print("Outliers for 'value':")
print(outliers_value)

print("Outliers for 'value_squared':")
print(outliers_value_squared)
```

```
Outliers for 'value':
0          True
1         False
2         False
3         False
4         False
          ...
111851    False
111852    False
111853    False
111854    False
111855    False
Name: value, Length: 111856, dtype: bool
Outliers for 'value_squared':
0          True
1         False
2         False
3         False
4         False
          ...
111851    False
111852    False
111853    False
111854    False
111855    False
Name: value_squared, Length: 111856, dtype: bool
```

In [ ]:

In [69]:
```
# Step 8: Fix Inconsistent Values: convert all strings to lowercase to address inco

df2['country'] = df2['country'].str.lower()

df2
```

Out[69]:

	area_code	country	item_code	animal_category	element_code	element	year_code	y
0	2	afghanistan	866	Cattle	5622	Import Value	2005	2
1	2	afghanistan	1057	Chickens	5609	Import Quantity	1997	1
2	2	afghanistan	1057	Chickens	5609	Import Quantity	1998	1
3	2	afghanistan	1057	Chickens	5609	Import Quantity	1999	1
4	2	afghanistan	1057	Chickens	5609	Import Quantity	2000	2
...	...	...	...	...	...	...	...	
111851	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1986	1
111852	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1987	1
111853	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1988	1
111854	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1989	1
111855	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1992	1

111856 rows × 16 columns

In [71]:
```
# Step 9: Replace Inconsistent Values with Standardized Ones
# For example, replacing 'united states' with 'United States of America'
df2['country'].replace({'united states': 'United States of America'}, inplace=True)

df2
```

Out[71]:

	area_code	country	item_code	animal_category	element_code	element	year_code	y
0	2	afghanistan	866	Cattle	5622	Import Value	2005	2
1	2	afghanistan	1057	Chickens	5609	Import Quantity	1997	1
2	2	afghanistan	1057	Chickens	5609	Import Quantity	1998	1
3	2	afghanistan	1057	Chickens	5609	Import Quantity	1999	1
4	2	afghanistan	1057	Chickens	5609	Import Quantity	2000	2
...	...	...	...	...	...	...	...	
111851	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1986	1
111852	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1987	1
111853	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1988	1
111854	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1989	1
111855	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1992	1

111856 rows × 16 columns

In [73]:
```
# Step 9: Replace Inconsistent Values with Standardized Ones
# For example, replacing 'united states' with 'United States of America'
df2['country'].replace({'afghanistan': 'Afghanistan'}, inplace=True)

df2
```

Out[73]:

	area_code	country	item_code	animal_category	element_code	element	year_code	y
0	2	Afghanistan	866	Cattle	5622	Import Value	2005	2
1	2	Afghanistan	1057	Chickens	5609	Import Quantity	1997	1
2	2	Afghanistan	1057	Chickens	5609	Import Quantity	1998	1
3	2	Afghanistan	1057	Chickens	5609	Import Quantity	1999	1
4	2	Afghanistan	1057	Chickens	5609	Import Quantity	2000	2
...	...	...	...	...	...	...	...	
111851	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1986	1
111852	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1987	1
111853	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1988	1
111854	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1989	1
111855	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1992	1

111856 rows × 16 columns

In [74]:
```
df2.tail()
```

Out[74]:

	area_code	country	item_code	animal_category	element_code	element	year_code	y
111851	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1986	1
111852	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1987	1
111853	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1988	1
111854	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1989	1
111855	5817	net food importing developing countries	1079	Turkeys	5922	Export Value	1992	1

In [78]:

```
# Step 10: Making countries names start with capital letter, except preposition
# List of common prepositions to be converted to lowercase
prepositions = ['on', 'and', 'in', 'to', 'with', 'by', 'at', 'for', 'of', 'from']

# Function to capitalize each word in a string, except for prepositions
def capitalize_country_name(country):
    words = country.split()  # Split the country name into words
    capitalized_words = [word.capitalize() if word.lower() not in prepositions else
    return ' '.join(capitalized_words)

# Apply the function to the 'country' column
df2['country'] = df2['country'].apply(capitalize_country_name)

# Print the updated DataFrame
df2.head()
```

Out[78]:

	area_code	country	item_code	animal_category	element_code	element	year_code	year
0	2	Afghanistan	866	Cattle	5622	Import Value	2005	2005
1	2	Afghanistan	1057	Chickens	5609	Import Quantity	1997	1997
2	2	Afghanistan	1057	Chickens	5609	Import Quantity	1998	1998
3	2	Afghanistan	1057	Chickens	5609	Import Quantity	1999	1999
4	2	Afghanistan	1057	Chickens	5609	Import Quantity	2000	2000

In [77]:
```
df2.tail()
```

Out[77]:

	area_code	country	item_code	animal_category	element_code	element	year_code	y
111851	5817	Net Food Importing Developing Countries	1079	Turkeys	5922	Export Value	1986	1
111852	5817	Net Food Importing Developing Countries	1079	Turkeys	5922	Export Value	1987	1
111853	5817	Net Food Importing Developing Countries	1079	Turkeys	5922	Export Value	1988	1
111854	5817	Net Food Importing Developing Countries	1079	Turkeys	5922	Export Value	1989	1
111855	5817	Net Food Importing Developing Countries	1079	Turkeys	5922	Export Value	1992	1

In [79]:
```
pip install fuzzywuzzy
```

```
Requirement already satisfied: fuzzywuzzy in c:\users\mariastella\anaconda3\lib\si
te-packages (0.18.0)Note: you may need to restart the kernel to use updated packag
es.
```

In [80]:
```
# Step 11: Conduct Fuzzy Matching

import pandas as pd
from fuzzywuzzy import fuzz

# Assuming df2 is your DataFrame
# If you're reading it from a CSV or Excel file, use pd.read_csv() or pd.read_excel
# For example:
# df2 = pd.read_csv('your_file.csv')

# Function to find fuzzy matches for a given input string in the 'country' column c
def fuzzy_match(input_string, choices, threshold=70):
    """
    Find fuzzy matches for input_string in choices list.

    Args:
        input_string (str): Input string to match.
        choices (list): List of strings to search for matches.
        threshold (int, optional): Fuzzy matching threshold (0-100). Defaults to 70

    Returns:
        list: List of tuples containing (matched_string, similarity_score).
    """
    matches = []
    for choice in choices:
        similarity = fuzz.partial_ratio(input_string, choice)
```

```
        if similarity >= threshold:
            matches.append((choice, similarity))
    return matches

# List of 5 countries to perform fuzzy matching for
countries = ['United States', 'France', 'Germany', 'United Kingdom', 'Japan']

# Iterate through each country and find fuzzy matches
for country in countries:
    matches = fuzzy_match(country, df2['country'].tolist())
    if matches:
        print(f"Fuzzy matches for '{country}':")
        for match, similarity in matches:
            print(f"{match} (Similarity: {similarity}%)")
    else:
        print(f"No fuzzy matches found for '{country}'.")
```

```
Fuzzy matches for 'United States':
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
United States of America (Similarity: 100%)
Fuzzy matches for 'France':
```

```
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
```

```
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
France (Similarity: 100%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
```

```
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
```

```
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
```

```
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Ukraine (Similarity: 73%)
Fuzzy matches for 'Germany':
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
```

```
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
```

```
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
```

```
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
```

```
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
```

```
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Germany (Similarity: 100%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
```

```
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
```

```
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
```

```
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
```

```
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
```

```
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Oman (Similarity: 75%)
Fuzzy matches for 'United Kingdom':
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
```

```
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
```

United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)

```
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
```

```
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
United Kingdom (Similarity: 100%)
Fuzzy matches for 'Japan':
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
```

```
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
```

```
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
```

```
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
```

```
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
```

```
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
```

```
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
```

```
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
```

```
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
Japan (Similarity: 100%)
```

In [81]:

```
# Step 12: Cleaned Dataset: Print the cleaned dataset

# Cleaned Dataset: Print the cleaned dataset
print("Cleaned Dataset:")
df2
```

Cleaned Dataset:

Out[81]:

	area_code	country	item_code	animal_category	element_code	element	year_code	y
0	2	Afghanistan	866	Cattle	5622	Import Value	2005	2
1	2	Afghanistan	1057	Chickens	5609	Import Quantity	1997	1
2	2	Afghanistan	1057	Chickens	5609	Import Quantity	1998	1
3	2	Afghanistan	1057	Chickens	5609	Import Quantity	1999	1
4	2	Afghanistan	1057	Chickens	5609	Import Quantity	2000	2
...	...	...	...	...	...	...	...	
111851	5817	Net Food Importing Developing Countries	1079	Turkeys	5922	Export Value	1986	1
111852	5817	Net Food Importing Developing Countries	1079	Turkeys	5922	Export Value	1987	1
111853	5817	Net Food Importing Developing Countries	1079	Turkeys	5922	Export Value	1988	1
111854	5817	Net Food Importing Developing Countries	1079	Turkeys	5922	Export Value	1989	1
111855	5817	Net Food Importing Developing Countries	1079	Turkeys	5922	Export Value	1992	1

111856 rows × 16 columns

In [ ]:

In [219…

```
# Step 13: Save the DataFrame as a CSV file in the current directory
df2.to_csv("df2.csv", index=False)

# Print a message indicating successful saving
print("df2 dataset saved as df2.csv in the current directory.")
```

```
df2 dataset saved as df2.csv in the current directory.
```

In [221…

```
# Step 14: Export the clean dataset to local computer

import shutil

# Source file path (current directory)
source_path = "df2.csv"
```

```
# Destination directory
destination_dir = "C:\\Users\\MariaStella\\Downloads"

# Move the file to the destination directory
shutil.move(source_path, destination_dir)

# Print the path of the moved file
print("df2 dataset moved to:", destination_dir)
```

```
df2 dataset moved to: C:\Users\MariaStella\Downloads
```

In [ ]:

1 paragraph of the ethical implications of data wrangling specific to the datasource

Ethical Implications of Data Wrangling for the FAOSTAT Live Animals Dataset:

In the process of data wrangling the FAOSTAT Live Animals dataset, ethical implications arise concerning data privacy, integrity, and transparency. Given the extensive historical data spanning over 50 years and covering over 200 countries, it is crucial to ensure that the data remains anonymized and that no sensitive information about individuals or specific entities is exposed. The integrity of the data must be maintained by accurately handling outliers and inconsistencies without introducing biases or errors that could misrepresent trends or patterns. Transparency in the data wrangling process is also essential, as stakeholders must be able to trust the methods used to clean and format the data. Clear documentation of the transformations and decisions made during data wrangling ensures that the dataset can be reliably used for further analysis and that the findings derived from it are credible and ethically sound.

In [ ]:

Responses to the questions based on Milestone 4 project activities:

What changes were made to the data?

Various data transformation and cleansing steps were applied to the FAOSTAT Live Animals dataset. These steps included replacing headers with more descriptive names, reformatting the data into a more readable structure, identifying and handling outliers and erroneous values, removing duplicate entries, and standardizing inconsistent casing or values across the dataset.

Are there any legal or regulatory guidelines for your data or project topic?

The FAOSTAT Live Animals dataset is publicly available and provided by the United Nations, which ensures compliance with international data sharing standards. However, the use of this data must still adhere to general data privacy regulations such as the General Data

Protection Regulation (GDPR) in Europe, ensuring that no personal data is inadvertently disclosed or misused.

What risks could be created based on the transformations done?

The transformations could introduce risks such as data distortion if not executed carefully. For instance, improper handling of outliers could lead to the loss of significant data points, and mismanagement of duplicates could result in inflated or deflated statistical analyses. These risks could potentially lead to erroneous conclusions and misinformed decisions based on the dataset.

Did you make any assumptions in cleaning/transforming the data?

Yes, certain assumptions were made during the data cleaning process. For example, it was assumed that missing values in specific fields should be treated as nulls rather than zeroes to avoid skewing the data. Additionally, it was assumed that certain inconsistencies in casing and value formatting were unintentional errors rather than meaningful distinctions.

How was your data sourced/verified for credibility?

The dataset was sourced from FAOSTAT, a reputable and reliable provider of global food and agriculture statistics managed by the United Nations. The credibility of the data was further verified by cross-referencing key statistics with other authoritative sources in agricultural research to ensure consistency and accuracy.

Was your data acquired in an ethical way?

Yes, the data was acquired ethically, following the proper procedures for accessing and using publicly available datasets. The data is openly shared by FAOSTAT for research and analysis purposes, ensuring that the use of this data adheres to ethical standards and promotes transparency.

How would you mitigate any of the ethical implications you have identified?

To mitigate ethical implications, several measures can be taken. First, maintaining transparency in the data wrangling process through thorough documentation of all transformations ensures accountability. Second, implementing rigorous validation checks during the data cleaning process helps maintain data integrity. Third, ensuring that any sensitive or potentially identifiable information is anonymized protects privacy. Lastly, adhering to legal and regulatory guidelines while engaging in continuous ethical reviews ensures that the data handling process remains compliant with ethical standards.

In [ ]:

Short Report on Term Project Milstone 4:

Milestone 4: Connecting to an API/Pulling in the Data and Cleaning/Formatting

Summary:

Milestone 4 involves connecting to an API, pulling in the data, and performing at least five data transformation and cleansing steps to ensure a clean dataset at the end of the milestone. The Kaggle API data was subjected to various transformation techniques such as replacing headers, formatting data into a more readable format, identifying outliers, finding duplicates, and fixing casing or inconsistent values.

Introduction:

Data obtained from APIs often require cleaning and formatting to prepare them for analysis. In this milestone 4, I connected to Kaggle API and pulled the LiveAnimals Dataset by applying various transformation techniques to ensure the dataset is clean, consistent, and formatted correctly. By performing these steps, I aimed to improve the quality and reliability of the data for subsequent analysis.

Statement of the Problem:

The data obtained from the API may contain errors, inconsistencies, or missing values that could affect the accuracy of the analysis. Therefore, it is essential to perform data cleaning and formatting to ensure the dataset is reliable and suitable for analysis. Ethical considerations such as data privacy, integrity, and transparency must be upheld throughout the data wrangling process.

About the Dataset:

In this Milestone 4 of the term project, the FAOSTAT historical dataset on Live Animals was obtained from Kaggle using this API Command: kaggle datasets download -d unitednations/global-food-agriculture-statistics. The dataset represents over 200 countries with more than 25 primary products and inputs collected between 1961 and 2013. Key variables in the dataset include Area or Country, Item (Agricultural Products, Cattle, Sheep, Chicken, Crops, etc.), Element (Import Quantity, Export Quantity, Import Value, Export Value), Year (1961 – 2013), and Value.

Methodology:

More than 10 data transformation and cleansing steps were applied to the API data:

Replacing Headers: Standardized the headers to ensure they are descriptive and consistent. Formatting Data: Reformatted data into a more readable and consistent structure. Identifying Outliers: Detected and handled outliers to maintain data integrity. Finding Duplicates: Identified and removed duplicate entries to ensure data accuracy. Fixing Casing/Inconsistent Values: Corrected casing issues and ensured uniformity in data values. & others.

Each step was clearly labeled and described to provide transparency and clarity in the data wrangling process.

Results:

Upon completion of the data transformation and cleansing steps, the API dataset was clean, consistent, and formatted correctly. The human-readable dataset printed at the end of the milestone demonstrated the effectiveness of the applied transformations, with improved

readability and accuracy. The dataset is now ready for further analysis in subsequent milestones.

Discussion:

The data transformation and cleansing steps are essential for ensuring the integrity and reliability of the dataset. By addressing issues such as outliers, duplicates, and inconsistent values, I improved the quality of the data and minimized the risk of erroneous conclusions. Ethical implications such as data privacy and integrity were carefully considered throughout the process to uphold ethical standards and maintain trust in the analysis.

Conclusion:

In conclusion, the data transformation and cleansing steps applied in this milestone played a crucial role in preparing the API dataset for analysis. By identifying and addressing issues such as outliers, duplicates, and inconsistent values, I ensured the dataset is clean, consistent, and ready for further exploration. Moving forward, the clean dataset will serve as the foundation for subsequent analyses and insights in the project.

The Way Forward:

The next steps will involve conducting exploratory data analysis (EDA) and visualization to gain insights into the relationships and patterns within the dataset. By analyzing the cleaned dataset, I can uncover trends, correlations, and anomalies that provide valuable insights into the subject area. Ethical considerations will continue to be prioritized, with a focus on maintaining data privacy, integrity, and transparency throughout the analysis process.

In [ ]: