

**Project Report**

Introduction to Data Science

Semester – 2

“Global Trade Analysis: 34 Years of Export-Import Data (1988–2021)”

By

Prem Panchal

Reg no: 2411021240011

GitHub link: [Prem2309/DATA-SCIENCE](https://github.com/Prem2309/DATA-SCIENCE)

Department of Computer Application

Alliance University Chandapura — Anekal Main Road,

Anekal Bengaluru — 562 106

April 2025

# Project Overview

This project focuses on analysing a dataset covering 34 years of global trade. The data includes export and import values for multiple countries from 1988 to 2021. By cleaning, transforming, visualizing, and modelling this dataset, we aim to uncover patterns, trends, and insights about the global economy and trade relationships.

The study also uses machine learning techniques like Linear Regression to predict imports based on export values, providing practical forecasting tools for policy makers and economists**.**

#### Project Goals

* To understand long-term trends in global exports and imports
* To identify top trading countries and their trade evolution
* To compute and analyse trade balances
* To visualize patterns using graphs, heatmaps, and plots
* To apply regression modelling for prediction of trade values
* To create a well-structured analytical report for academic and industry relevance

#### Challenges Faced

* Data Cleaning: Inconsistent column names, missing values, and formatting issues
* Large Dataset Size: Aggregating 34 years of data across many countries required careful optimization
* Visualization Complexity: Representing trends clearly without cluttered visuals
* Model Accuracy: Achieving a good R² value while avoiding overfitting
* Report Design: Balancing technical depth with visual clarity for a ~20-page PDF

#### Tools and Libraries Used

* **Pandas** – Data manipulation and transformation
* **Matplotlib & Seaborn** – Graphs and visualizations
* **Scikit-learn** – Linear regression modelling and evaluation
* **Jupyter Notebook** – Interactive development
* **MS Word / LaTeX / PDF Export** – Final report generation

import pandas as pd  
df=pd.read\_csv(r"C:\Users\Prem Panchal\Downloads\34\_years\_world\_export\_import\_dataset.csv")

df

Partner Name Year Export (US$ Thousand) \  
0 Aruba 1988 3.498100e+03   
1 Afghanistan 1988 2.130304e+05   
2 Angola 1988 3.755279e+05   
3 Anguila 1988 3.669800e+02   
4 Albania 1988 3.010356e+04   
... ... ... ...   
8091 Latin America & Caribbean 2021 1.330557e+09   
8092 Middle East & North Africa 2021 1.196712e+09   
8093 North America 2021 3.823319e+09   
8094 South Asia 2021 6.991380e+08   
8095 Sub-Saharan Africa 2021 4.951000e+08   
  
 Import (US$ Thousand) Export Product Share (%) \  
0 3.284900e+02 100.0   
1 5.445952e+04 100.0   
2 3.707028e+05 100.0   
3 4.000000e+00 100.0   
4 4.770930e+04 100.0   
... ... ...   
8091 1.310305e+09 100.0   
8092 1.088471e+09 100.0   
8093 2.219849e+09 100.0   
8094 4.723832e+08 100.0   
8095 4.350468e+08 100.0   
  
 Import Product Share (%) Revealed comparative advantage \  
0 100 NaN   
1 100 NaN   
2 100 NaN   
3 100 NaN   
4 100 NaN   
... ... ...   
8091 100 NaN   
8092 100 NaN   
8093 100 NaN   
8094 100 NaN   
8095 100 NaN   
  
 World Growth (%) Country Growth (%) AHS Simple Average (%) ... \  
0 NaN NaN 2.80 ...   
1 NaN NaN 0.88 ...   
2 NaN NaN 2.02 ...   
3 NaN NaN 3.71 ...   
4 NaN NaN 1.84 ...   
... ... ... ... ...   
8091 NaN NaN 3.84 ...   
8092 NaN NaN 4.63 ...   
8093 NaN NaN 6.45 ...   
8094 NaN NaN 5.09 ...   
8095 NaN NaN 3.22 ...   
  
 MFN Total Tariff Lines MFN Dutiable Tariff Lines Share (%) \  
0 1152.0 63.54   
1 4142.0 69.41   
2 5438.0 76.00   
3 322.0 66.15   
4 5684.0 66.87   
... ... ...   
8091 1509731.0 67.97   
8092 1507304.0 67.98   
8093 1533357.0 67.86   
8094 1508290.0 67.94   
8095 1513840.0 67.92   
  
 MFN Duty Free Tariff Lines Share (%) \  
0 22.74   
1 15.64   
2 16.27   
3 22.05   
4 19.19   
... ...   
8091 25.91   
8092 25.91   
8093 26.04   
8094 26.02   
8095 25.93   
  
 MFN Specific Tariff Lines Share (%) MFN AVE Tariff Lines Share (%) \  
0 70.32 31.61   
1 72.45 40.51   
2 41.55 24.80   
3 78.79 36.36   
4 57.93 48.52   
... ... ...   
8091 0.39 5.08   
8092 0.46 5.92   
8093 0.59 7.58   
8094 0.68 8.71   
8095 0.70 8.99   
  
 MFN MaxRate (%) MFN MinRate (%) \  
0 352.69 0.0   
1 2029.66 0.0   
2 451.15 0.0   
3 100.00 0.0   
4 3000.00 0.0   
... ... ...   
8091 3000.00 0.0   
8092 3000.00 0.0   
8093 3000.00 0.0   
8094 3000.00 0.0   
8095 3000.00 0.0   
  
 MFN SpecificDuty Imports (US$ Thousand) \  
0 2.186000e+03   
1 7.843691e+04   
2 7.277420e+05   
3 9.400000e+01   
4 3.790409e+04   
... ...   
8091 4.193397e+08   
8092 5.809079e+08   
8093 6.433132e+08   
8094 8.497724e+07   
8095 2.234618e+08   
  
 MFN Dutiable Imports (US$ Thousand) \  
0 3.128020e+03   
1 9.419150e+04   
2 9.214815e+05   
3 5.830000e+02   
4 1.011959e+05   
... ...   
8091 1.346740e+09   
8092 1.117309e+09   
8093 2.055953e+09   
8094 5.123190e+08   
8095 4.915992e+08   
  
 MFN Duty Free Imports (US$ Thousand)   
0 0.00   
1 0.00   
2 0.00   
3 0.00   
4 0.00   
... ...   
8091 1632.05   
8092 0.00   
8093 23691.96   
8094 0.05   
8095 2.83   
  
[8096 rows x 33 columns]

#Understanding the data  
df.head()

Partner Name Year Export (US$ Thousand) Import (US$ Thousand) \  
0 Aruba 1988 3498.10 328.49   
1 Afghanistan 1988 213030.40 54459.52   
2 Angola 1988 375527.89 370702.76   
3 Anguila 1988 366.98 4.00   
4 Albania 1988 30103.56 47709.30   
  
 Export Product Share (%) Import Product Share (%) \  
0 100.0 100   
1 100.0 100   
2 100.0 100   
3 100.0 100   
4 100.0 100   
  
 Revealed comparative advantage World Growth (%) Country Growth (%) \  
0 NaN NaN NaN   
1 NaN NaN NaN   
2 NaN NaN NaN   
3 NaN NaN NaN   
4 NaN NaN NaN   
  
 AHS Simple Average (%) ... MFN Total Tariff Lines \  
0 2.80 ... 1152.0   
1 0.88 ... 4142.0   
2 2.02 ... 5438.0   
3 3.71 ... 322.0   
4 1.84 ... 5684.0   
  
 MFN Dutiable Tariff Lines Share (%) MFN Duty Free Tariff Lines Share (%) \  
0 63.54 22.74   
1 69.41 15.64   
2 76.00 16.27   
3 66.15 22.05   
4 66.87 19.19   
  
 MFN Specific Tariff Lines Share (%) MFN AVE Tariff Lines Share (%) \  
0 70.32 31.61   
1 72.45 40.51   
2 41.55 24.80   
3 78.79 36.36   
4 57.93 48.52   
  
 MFN MaxRate (%) MFN MinRate (%) MFN SpecificDuty Imports (US$ Thousand) \  
0 352.69 0.0 2186.00   
1 2029.66 0.0 78436.91   
2 451.15 0.0 727741.99   
3 100.00 0.0 94.00   
4 3000.00 0.0 37904.09   
  
 MFN Dutiable Imports (US$ Thousand) MFN Duty Free Imports (US$ Thousand)   
0 3128.02 0.0   
1 94191.50 0.0   
2 921481.52 0.0   
3 583.00 0.0   
4 101195.95 0.0   
  
[5 rows x 33 columns]

df.tail()

Partner Name Year Export (US$ Thousand) \  
8091 Latin America & Caribbean 2021 1.330557e+09   
8092 Middle East & North Africa 2021 1.196712e+09   
8093 North America 2021 3.823319e+09   
8094 South Asia 2021 6.991380e+08   
8095 Sub-Saharan Africa 2021 4.951000e+08   
  
 Import (US$ Thousand) Export Product Share (%) \  
8091 1.310305e+09 100.0   
8092 1.088471e+09 100.0   
8093 2.219849e+09 100.0   
8094 4.723832e+08 100.0   
8095 4.350468e+08 100.0   
  
 Import Product Share (%) Revealed comparative advantage \  
8091 100 NaN   
8092 100 NaN   
8093 100 NaN   
8094 100 NaN   
8095 100 NaN   
  
 World Growth (%) Country Growth (%) AHS Simple Average (%) ... \  
8091 NaN NaN 3.84 ...   
8092 NaN NaN 4.63 ...   
8093 NaN NaN 6.45 ...   
8094 NaN NaN 5.09 ...   
8095 NaN NaN 3.22 ...   
  
 MFN Total Tariff Lines MFN Dutiable Tariff Lines Share (%) \  
8091 1509731.0 67.97   
8092 1507304.0 67.98   
8093 1533357.0 67.86   
8094 1508290.0 67.94   
8095 1513840.0 67.92   
  
 MFN Duty Free Tariff Lines Share (%) \  
8091 25.91   
8092 25.91   
8093 26.04   
8094 26.02   
8095 25.93   
  
 MFN Specific Tariff Lines Share (%) MFN AVE Tariff Lines Share (%) \  
8091 0.39 5.08   
8092 0.46 5.92   
8093 0.59 7.58   
8094 0.68 8.71   
8095 0.70 8.99   
  
 MFN MaxRate (%) MFN MinRate (%) \  
8091 3000.0 0.0   
8092 3000.0 0.0   
8093 3000.0 0.0   
8094 3000.0 0.0   
8095 3000.0 0.0   
  
 MFN SpecificDuty Imports (US$ Thousand) \  
8091 4.193397e+08   
8092 5.809079e+08   
8093 6.433132e+08   
8094 8.497724e+07   
8095 2.234618e+08   
  
 MFN Dutiable Imports (US$ Thousand) \  
8091 1.346740e+09   
8092 1.117309e+09   
8093 2.055953e+09   
8094 5.123190e+08   
8095 4.915992e+08   
  
 MFN Duty Free Imports (US$ Thousand)   
8091 1632.05   
8092 0.00   
8093 23691.96   
8094 0.05   
8095 2.83   
  
[5 rows x 33 columns]

df.describe()

Year Export (US$ Thousand) Import (US$ Thousand) \  
count 8096.000000 8.096000e+03 8.096000e+03   
mean 2004.908226 1.421192e+08 1.305216e+08   
std 9.707831 9.928417e+08 9.073802e+08   
min 1988.000000 0.000000e+00 3.000000e-02   
25% 1997.000000 4.274264e+05 1.601335e+05   
50% 2005.000000 3.719683e+06 2.053967e+06   
75% 2013.000000 2.585514e+07 2.102937e+07   
max 2021.000000 2.422743e+10 2.193121e+10   
  
 Export Product Share (%) Import Product Share (%) \  
count 8076.0 8096.0   
mean 100.0 100.0   
std 0.0 0.0   
min 100.0 100.0   
25% 100.0 100.0   
50% 100.0 100.0   
75% 100.0 100.0   
max 100.0 100.0   
  
 Revealed comparative advantage World Growth (%) Country Growth (%) \  
count 4712.0 4410.000000 4410.000000   
mean 1.0 3.986016 3.986016   
std 0.0 10.004221 10.004221   
min 1.0 -62.280000 -62.280000   
25% 1.0 -1.437500 -1.437500   
50% 1.0 3.830000 3.830000   
75% 1.0 9.407500 9.407500   
max 1.0 174.000000 174.000000   
  
 AHS Simple Average (%) AHS Weighted Average (%) ... \  
count 8080.000000 8080.000000 ...   
mean 6.656053 6.082833 ...   
std 3.837668 6.765269 ...   
min 0.000000 0.000000 ...   
25% 4.030000 2.100000 ...   
50% 6.120000 4.700000 ...   
75% 8.480000 8.050000 ...   
max 46.980000 197.760000 ...   
  
 MFN Total Tariff Lines MFN Dutiable Tariff Lines Share (%) \  
count 8.081000e+03 8081.000000   
mean 4.291884e+05 72.722564   
std 4.337735e+05 5.245883   
min 1.700000e+01 37.110000   
25% 6.481600e+04 69.400000   
50% 2.669150e+05 71.340000   
75% 7.071260e+05 75.260000   
max 1.538248e+06 98.210000   
  
 MFN Duty Free Tariff Lines Share (%) \  
count 8081.000000   
mean 20.893574   
std 4.554328   
min 1.790000   
25% 18.320000   
50% 22.030000   
75% 24.100000   
max 62.890000   
  
 MFN Specific Tariff Lines Share (%) MFN AVE Tariff Lines Share (%) \  
count 8080.000000 8080.000000   
mean 52.018722 133.145250   
std 84.848859 161.189034   
min 0.000000 0.000000   
25% 10.617500 33.375000   
50% 28.180000 88.050000   
75% 55.807500 176.242500   
max 1800.000000 3860.000000   
  
 MFN MaxRate (%) MFN MinRate (%) \  
count 8081.000000 8081.0   
mean 2710.981771 0.0   
std 840.575331 0.0   
min 25.000000 0.0   
25% 3000.000000 0.0   
50% 3000.000000 0.0   
75% 3000.000000 0.0   
max 5000.000000 0.0   
  
 MFN SpecificDuty Imports (US$ Thousand) \  
count 8.081000e+03   
mean 5.454060e+07   
std 4.137543e+08   
min 0.000000e+00   
25% 4.682219e+04   
50% 9.012839e+05   
75% 8.704754e+06   
max 1.649501e+10   
  
 MFN Dutiable Imports (US$ Thousand) \  
count 8.081000e+03   
mean 1.097283e+08   
std 7.691260e+08   
min 9.400000e-01   
25% 1.785371e+05   
50% 2.048332e+06   
75% 1.717688e+07   
max 1.854956e+10   
  
 MFN Duty Free Imports (US$ Thousand)   
count 8.081000e+03   
mean 1.058794e+05   
std 4.178933e+06   
min 0.000000e+00   
25% 0.000000e+00   
50% 0.000000e+00   
75% 0.000000e+00   
max 3.495531e+08   
  
[8 rows x 32 columns]

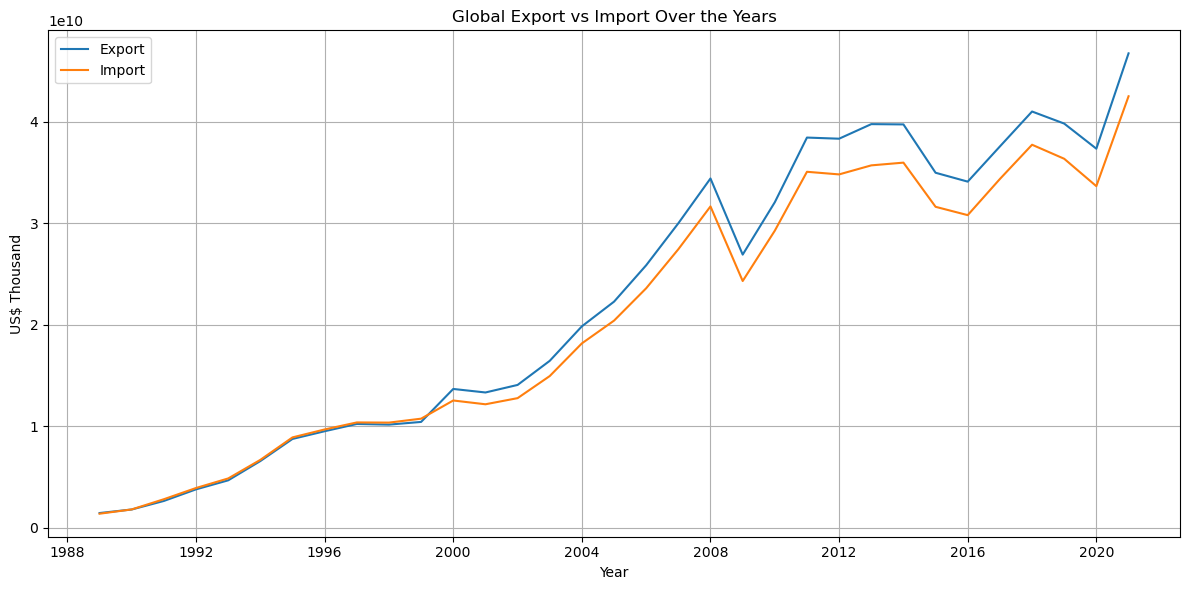
df.info()

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 8096 entries, 0 to 8095  
Data columns (total 33 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 Partner Name 8096 non-null object   
 1 Year 8096 non-null int64   
 2 Export (US$ Thousand) 8096 non-null float64  
 3 Import (US$ Thousand) 8096 non-null float64  
 4 Export Product Share (%) 8076 non-null float64  
 5 Import Product Share (%) 8096 non-null int64   
 6 Revealed comparative advantage 4712 non-null float64  
 7 World Growth (%) 4410 non-null float64  
 8 Country Growth (%) 4410 non-null float64  
 9 AHS Simple Average (%) 8080 non-null float64  
 10 AHS Weighted Average (%) 8080 non-null float64  
 11 AHS Total Tariff Lines 8080 non-null float64  
 12 AHS Dutiable Tariff Lines Share (%) 8080 non-null float64  
 13 AHS Duty Free Tariff Lines Share (%) 8080 non-null float64  
 14 AHS Specific Tariff Lines Share (%) 8080 non-null float64  
 15 AHS AVE Tariff Lines Share (%) 8080 non-null float64  
 16 AHS MaxRate (%) 8080 non-null float64  
 17 AHS MinRate (%) 8080 non-null float64  
 18 AHS SpecificDuty Imports (US$ Thousand) 8081 non-null float64  
 19 AHS Dutiable Imports (US$ Thousand) 8081 non-null float64  
 20 AHS Duty Free Imports (US$ Thousand) 8081 non-null float64  
 21 MFN Simple Average (%) 8081 non-null float64  
 22 MFN Weighted Average (%) 8081 non-null float64  
 23 MFN Total Tariff Lines 8081 non-null float64  
 24 MFN Dutiable Tariff Lines Share (%) 8081 non-null float64  
 25 MFN Duty Free Tariff Lines Share (%) 8081 non-null float64  
 26 MFN Specific Tariff Lines Share (%) 8080 non-null float64  
 27 MFN AVE Tariff Lines Share (%) 8080 non-null float64  
 28 MFN MaxRate (%) 8081 non-null float64  
 29 MFN MinRate (%) 8081 non-null float64  
 30 MFN SpecificDuty Imports (US$ Thousand) 8081 non-null float64  
 31 MFN Dutiable Imports (US$ Thousand) 8081 non-null float64  
 32 MFN Duty Free Imports (US$ Thousand) 8081 non-null float64  
dtypes: float64(30), int64(2), object(1)  
memory usage: 2.0+ MB

print(df.shape)

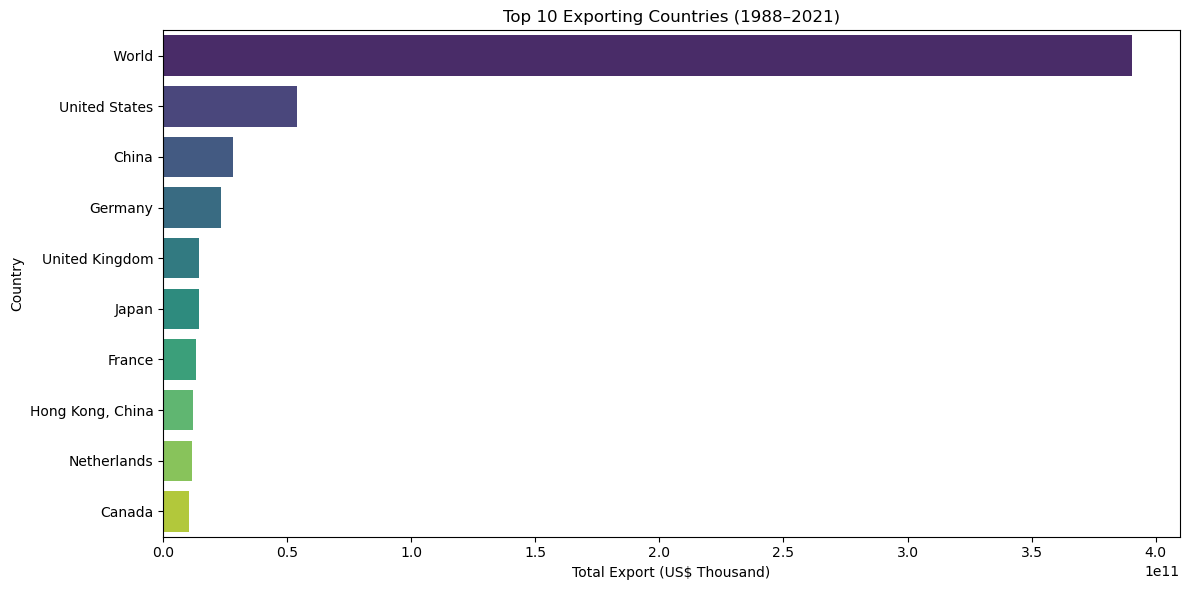
(4410, 33)

import matplotlib.pyplot as plt  
import seaborn as sns  
import numpy as np  
  
# Clean column names  
df.columns = df.columns.str.strip()  
  
# Create new features  
df['Trade Balance'] = df['Export (US$ Thousand)'] - df['Import (US$ Thousand)']  
df['Total Trade'] = df['Export (US$ Thousand)'] + df['Import (US$ Thousand)']  
  
# Plot 1: Export and Import trends over time (world total)  
yearly\_data = df.groupby('Year')[['Export (US$ Thousand)', 'Import (US$ Thousand)']].sum().reset\_index()  
  
plt.figure(figsize=(12, 6))  
sns.lineplot(data=yearly\_data, x='Year', y='Export (US$ Thousand)', label='Export')  
sns.lineplot(data=yearly\_data, x='Year', y='Import (US$ Thousand)', label='Import')  
plt.title('Global Export vs Import Over the Years')  
plt.ylabel('US$ Thousand')  
plt.grid(True)  
plt.legend()  
plt.tight\_layout()  
plt.show()

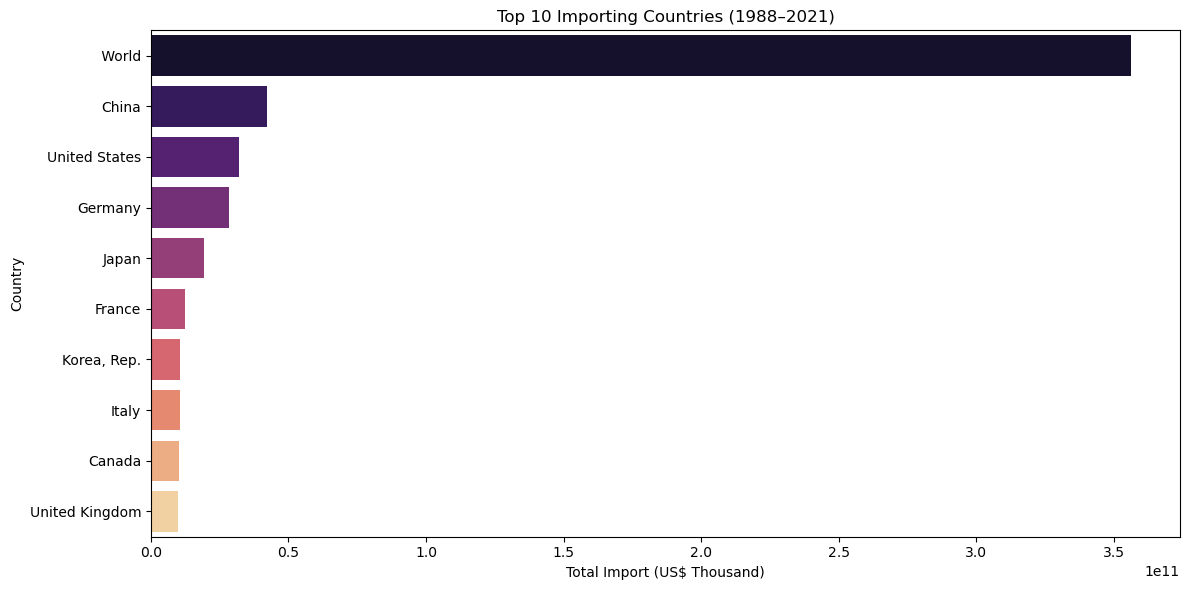


# Top 10 countries by total export and import over all years  
top\_exporters = df.groupby('Partner Name')['Export (US$ Thousand)'].sum().sort\_values(ascending=False).head(10)  
top\_importers = df.groupby('Partner Name')['Import (US$ Thousand)'].sum().sort\_values(ascending=False).head(10)  
  
# Plot 2: Top 10 Exporting Countries  
plt.figure(figsize=(12, 6))  
sns.barplot(x=top\_exporters.values, y=top\_exporters.index, palette='viridis')  
plt.title('Top 10 Exporting Countries (1988–2021)')  
plt.xlabel('Total Export (US$ Thousand)')  
plt.ylabel('Country')  
plt.tight\_layout()  
plt.show()  
  
# Plot 3: Top 10 Importing Countries  
plt.figure(figsize=(12, 6))  
sns.barplot(x=top\_importers.values, y=top\_importers.index, palette='magma')  
plt.title('Top 10 Importing Countries (1988–2021)')  
plt.xlabel('Total Import (US$ Thousand)')  
plt.ylabel('Country')  
plt.tight\_layout()  
plt.show()

C:\Users\Prem Panchal\AppData\Local\Temp\ipykernel\_6580\3041476904.py:7: FutureWarning:   
  
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.  
  
 sns.barplot(x=top\_exporters.values, y=top\_exporters.index, palette='viridis')



C:\Users\Prem Panchal\AppData\Local\Temp\ipykernel\_6580\3041476904.py:16: FutureWarning:   
  
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.  
  
 sns.barplot(x=top\_importers.values, y=top\_importers.index, palette='magma')



import pandas as pd  
from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LinearRegression  
from sklearn.metrics import mean\_squared\_error, r2\_score  
import matplotlib.pyplot as plt

df = pd.read\_csv(r"C:\Users\Prem Panchal\Downloads\34\_years\_world\_export\_import\_dataset.csv")  
  
# Clean column names  
df.columns = df.columns.str.strip()  
  
# Filter relevant columns  
df = df[['Year', 'Export (US$ Thousand)', 'Import (US$ Thousand)']].dropna()  
  
# Features and Target  
X = df[['Export (US$ Thousand)']]  
y = df['Import (US$ Thousand)']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

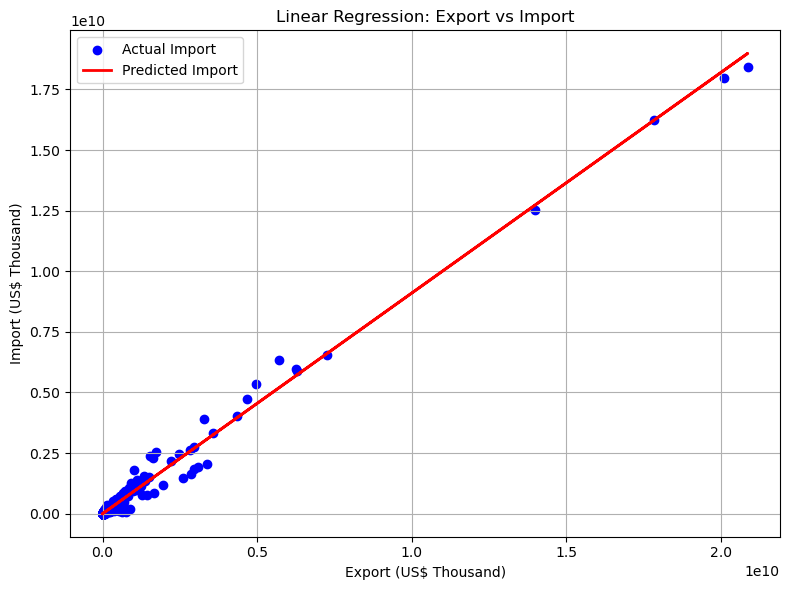
model = LinearRegression()  
model.fit(X\_train, y\_train)

LinearRegression()

y\_pred = model.predict(X\_test)  
  
# Evaluation  
print("R² Score:", r2\_score(y\_test, y\_pred))  
print("Mean Squared Error:", mean\_squared\_error(y\_test, y\_pred))

R² Score: 0.9883133868632338  
Mean Squared Error: 1.0136915205133322e+16

plt.figure(figsize=(8, 6))  
plt.scatter(X\_test, y\_test, color='blue', label='Actual Import')  
plt.plot(X\_test, y\_pred, color='red', linewidth=2, label='Predicted Import')  
plt.xlabel('Export (US$ Thousand)')  
plt.ylabel('Import (US$ Thousand)')  
plt.title('Linear Regression: Export vs Import')  
plt.legend()  
plt.grid(True)  
plt.tight\_layout()  
plt.show()



#Playing with Missing Data   
# importing libraries  
import pandas as pd  
import numpy as np

# loading dataset  
dataset = pd.read\_csv(r"C:\Users\Prem Panchal\Downloads\34\_years\_world\_export\_import\_dataset.csv")  
dataset.head()

Partner Name Year Export (US$ Thousand) Import (US$ Thousand) \  
0 Aruba 1988 3498.10 328.49   
1 Afghanistan 1988 213030.40 54459.52   
2 Angola 1988 375527.89 370702.76   
3 Anguila 1988 366.98 4.00   
4 Albania 1988 30103.56 47709.30   
  
 Export Product Share (%) Import Product Share (%) \  
0 100.0 100   
1 100.0 100   
2 100.0 100   
3 100.0 100   
4 100.0 100   
  
 Revealed comparative advantage World Growth (%) Country Growth (%) \  
0 NaN NaN NaN   
1 NaN NaN NaN   
2 NaN NaN NaN   
3 NaN NaN NaN   
4 NaN NaN NaN   
  
 AHS Simple Average (%) ... MFN Total Tariff Lines \  
0 2.80 ... 1152.0   
1 0.88 ... 4142.0   
2 2.02 ... 5438.0   
3 3.71 ... 322.0   
4 1.84 ... 5684.0   
  
 MFN Dutiable Tariff Lines Share (%) MFN Duty Free Tariff Lines Share (%) \  
0 63.54 22.74   
1 69.41 15.64   
2 76.00 16.27   
3 66.15 22.05   
4 66.87 19.19   
  
 MFN Specific Tariff Lines Share (%) MFN AVE Tariff Lines Share (%) \  
0 70.32 31.61   
1 72.45 40.51   
2 41.55 24.80   
3 78.79 36.36   
4 57.93 48.52   
  
 MFN MaxRate (%) MFN MinRate (%) MFN SpecificDuty Imports (US$ Thousand) \  
0 352.69 0.0 2186.00   
1 2029.66 0.0 78436.91   
2 451.15 0.0 727741.99   
3 100.00 0.0 94.00   
4 3000.00 0.0 37904.09   
  
 MFN Dutiable Imports (US$ Thousand) MFN Duty Free Imports (US$ Thousand)   
0 3128.02 0.0   
1 94191.50 0.0   
2 921481.52 0.0   
3 583.00 0.0   
4 101195.95 0.0   
  
[5 rows x 33 columns]

# missing values in the dataset  
missing\_values = dataset.isnull().sum()  
missing\_values

Partner Name 0  
Year 0  
Export (US$ Thousand) 0  
Import (US$ Thousand) 0  
Export Product Share (%) 20  
Import Product Share (%) 0  
Revealed comparative advantage 3384  
World Growth (%) 3686  
Country Growth (%) 3686  
AHS Simple Average (%) 16  
AHS Weighted Average (%) 16  
AHS Total Tariff Lines 16  
AHS Dutiable Tariff Lines Share (%) 16  
AHS Duty Free Tariff Lines Share (%) 16  
AHS Specific Tariff Lines Share (%) 16  
AHS AVE Tariff Lines Share (%) 16  
AHS MaxRate (%) 16  
AHS MinRate (%) 16  
AHS SpecificDuty Imports (US$ Thousand) 15  
AHS Dutiable Imports (US$ Thousand) 15  
AHS Duty Free Imports (US$ Thousand) 15  
MFN Simple Average (%) 15  
MFN Weighted Average (%) 15  
MFN Total Tariff Lines 15  
MFN Dutiable Tariff Lines Share (%) 15  
MFN Duty Free Tariff Lines Share (%) 15  
MFN Specific Tariff Lines Share (%) 16  
MFN AVE Tariff Lines Share (%) 16  
MFN MaxRate (%) 15  
MFN MinRate (%) 15  
MFN SpecificDuty Imports (US$ Thousand) 15  
MFN Dutiable Imports (US$ Thousand) 15  
MFN Duty Free Imports (US$ Thousand) 15  
dtype: int64

missing\_values[missing\_values > 0]

Export Product Share (%) 20  
Revealed comparative advantage 3384  
World Growth (%) 3686  
Country Growth (%) 3686  
AHS Simple Average (%) 16  
AHS Weighted Average (%) 16  
AHS Total Tariff Lines 16  
AHS Dutiable Tariff Lines Share (%) 16  
AHS Duty Free Tariff Lines Share (%) 16  
AHS Specific Tariff Lines Share (%) 16  
AHS AVE Tariff Lines Share (%) 16  
AHS MaxRate (%) 16  
AHS MinRate (%) 16  
AHS SpecificDuty Imports (US$ Thousand) 15  
AHS Dutiable Imports (US$ Thousand) 15  
AHS Duty Free Imports (US$ Thousand) 15  
MFN Simple Average (%) 15  
MFN Weighted Average (%) 15  
MFN Total Tariff Lines 15  
MFN Dutiable Tariff Lines Share (%) 15  
MFN Duty Free Tariff Lines Share (%) 15  
MFN Specific Tariff Lines Share (%) 16  
MFN AVE Tariff Lines Share (%) 16  
MFN MaxRate (%) 15  
MFN MinRate (%) 15  
MFN SpecificDuty Imports (US$ Thousand) 15  
MFN Dutiable Imports (US$ Thousand) 15  
MFN Duty Free Imports (US$ Thousand) 15  
dtype: int64

dataset.shape

(8096, 33)

# total data in the dataset  
total\_dataset = np.product(dataset.shape)  
total\_dataset

267168

# total missing values  
total\_missing = missing\_values.sum()  
total\_missing

11147

# total percentage of missing values  
percent = (total\_missing / total\_dataset) \* 100  
percent

4.172281111510361

y = dataset['Export (US$ Thousand)']  
y.head()

0 3498.10  
1 213030.40  
2 375527.89  
3 366.98  
4 30103.56  
Name: Export (US$ Thousand), dtype: float64

drop = dataset.drop(['Export (US$ Thousand)'], axis = 1)  
X = drop.select\_dtypes(exclude = 'object')  
X.head()

Year Import (US$ Thousand) Export Product Share (%) \  
0 1988 328.49 100.0   
1 1988 54459.52 100.0   
2 1988 370702.76 100.0   
3 1988 4.00 100.0   
4 1988 47709.30 100.0   
  
 Import Product Share (%) Revealed comparative advantage World Growth (%) \  
0 100 NaN NaN   
1 100 NaN NaN   
2 100 NaN NaN   
3 100 NaN NaN   
4 100 NaN NaN   
  
 Country Growth (%) AHS Simple Average (%) AHS Weighted Average (%) \  
0 NaN 2.80 2.92   
1 NaN 0.88 1.83   
2 NaN 2.02 3.89   
3 NaN 3.71 1.09   
4 NaN 1.84 2.38   
  
 AHS Total Tariff Lines ... MFN Total Tariff Lines \  
0 155.0 ... 1152.0   
1 548.0 ... 4142.0   
2 633.0 ... 5438.0   
3 33.0 ... 322.0   
4 744.0 ... 5684.0   
  
 MFN Dutiable Tariff Lines Share (%) MFN Duty Free Tariff Lines Share (%) \  
0 63.54 22.74   
1 69.41 15.64   
2 76.00 16.27   
3 66.15 22.05   
4 66.87 19.19   
  
 MFN Specific Tariff Lines Share (%) MFN AVE Tariff Lines Share (%) \  
0 70.32 31.61   
1 72.45 40.51   
2 41.55 24.80   
3 78.79 36.36   
4 57.93 48.52   
  
 MFN MaxRate (%) MFN MinRate (%) MFN SpecificDuty Imports (US$ Thousand) \  
0 352.69 0.0 2186.00   
1 2029.66 0.0 78436.91   
2 451.15 0.0 727741.99   
3 100.00 0.0 94.00   
4 3000.00 0.0 37904.09   
  
 MFN Dutiable Imports (US$ Thousand) MFN Duty Free Imports (US$ Thousand)   
0 3128.02 0.0   
1 94191.50 0.0   
2 921481.52 0.0   
3 583.00 0.0   
4 101195.95 0.0   
  
[5 rows x 31 columns]

from sklearn.model\_selection import train\_test\_split  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, train\_size= 0.8, test\_size= 0.2, random\_state= 0)

from sklearn.metrics import mean\_absolute\_error  
from sklearn.ensemble import RandomForestRegressor  
  
def score\_data(X\_train, X\_test, y\_train, y\_test):  
 model = RandomForestRegressor(n\_estimators=100, random\_state = 0)  
 model.fit(X\_train, y\_train)  
 pred = model.predict(X\_test)  
 return mean\_absolute\_error(y\_test, pred)

missing\_with\_columns = [col for col in X\_train.columns if X\_train[col].isnull().sum()]  
  
drop\_X\_train = X\_train.drop(missing\_with\_columns, axis = 1)  
drop\_X\_test = X\_test.drop(missing\_with\_columns, axis = 1)  
  
print(score\_data(drop\_X\_train, drop\_X\_test, y\_train, y\_test))

25236166.30298803

from sklearn.impute import SimpleImputer  
  
impute = SimpleImputer()  
imputed\_X\_train = pd.DataFrame(impute.fit\_transform(X\_train))  
imputed\_X\_test = pd.DataFrame(impute.fit\_transform(X\_test))  
  
imputed\_X\_train.columns = X\_train.columns  
imputed\_X\_test.columns = X\_test.columns  
  
print(score\_data(imputed\_X\_train, imputed\_X\_test, y\_train, y\_test))

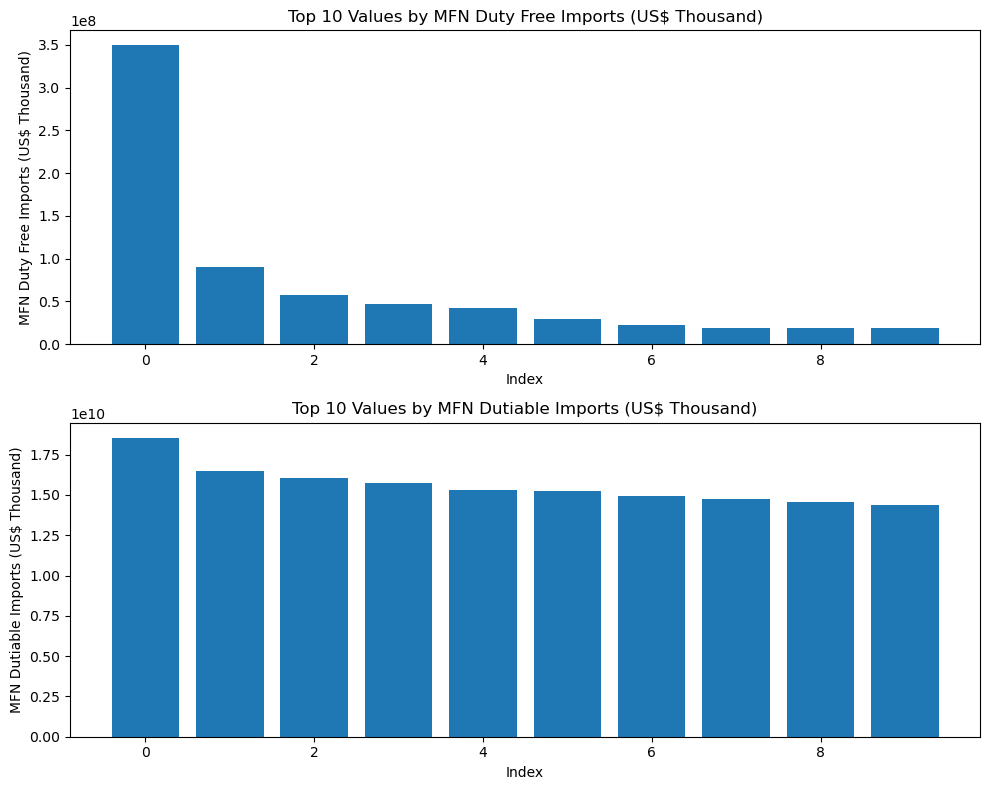
15968448.128519729

X\_train\_plus = X\_train.copy()  
X\_test\_plus = X\_test.copy()  
  
for col in missing\_with\_columns:  
 X\_train\_plus[col + "\_was\_missing"] = X\_train\_plus[col].isnull()  
 X\_test\_plus[col + "\_was\_missing"] = X\_test\_plus[col].isnull()  
  
imputed\_X\_train\_plus = pd.DataFrame(impute.fit\_transform(X\_train\_plus))  
imputed\_X\_test\_plus = pd.DataFrame(impute.fit\_transform(X\_test\_plus))  
  
imputed\_X\_train\_plus.columns = X\_train\_plus.columns  
imputed\_X\_test\_plus.columns = X\_test\_plus.columns  
  
print(score\_data(imputed\_X\_train\_plus, imputed\_X\_test\_plus, y\_train, y\_test))

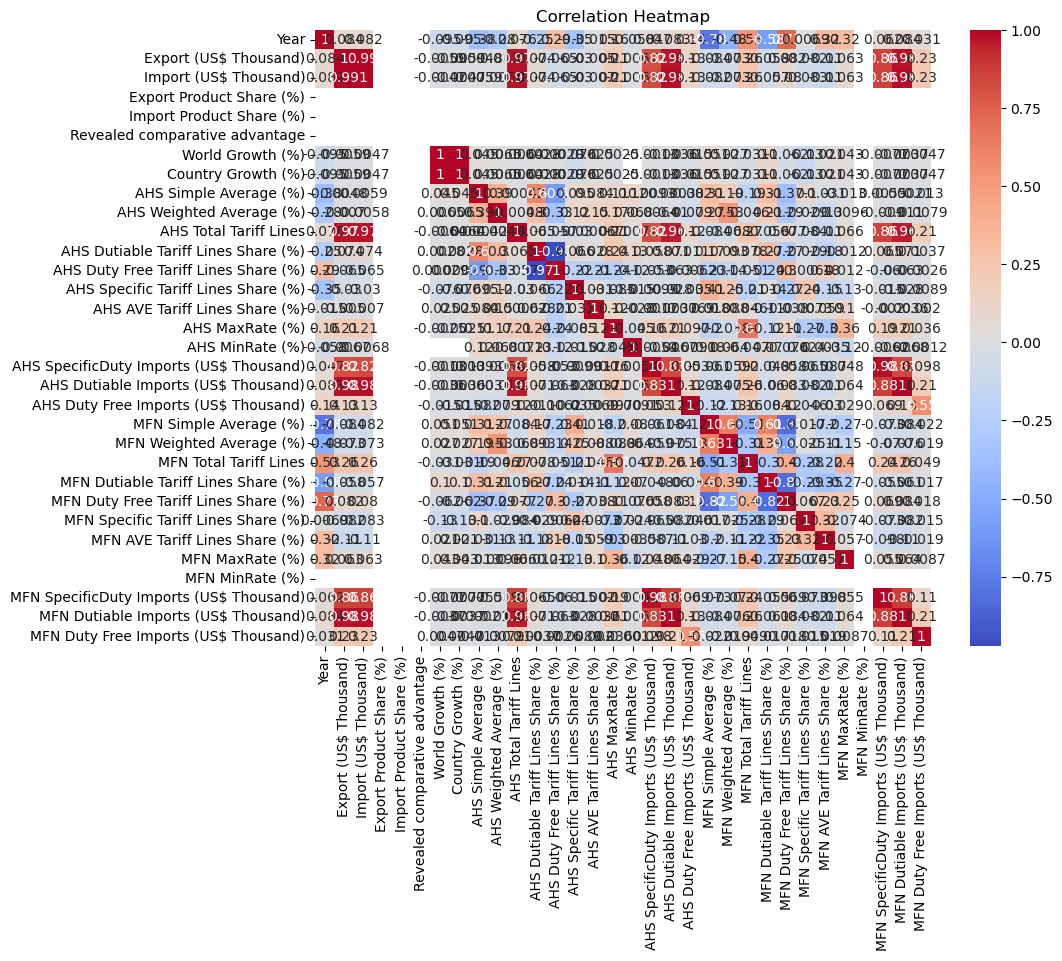
15855818.724038702

#Bar Graph  
  
import pandas as pd  
import matplotlib.pyplot as plt  
  
# Load the dataset  
df = pd.read\_csv(r"C:\Users\Prem Panchal\Downloads\34\_years\_world\_export\_import\_dataset.csv")  
  
# Get an overview of the data  
print(df.head())  
print(df.columns)  
  
# Let's select the last two numeric columns for visualization (assuming they are exports and imports)  
numeric\_cols = df.select\_dtypes(include=['int64', 'float64']).columns  
if len(numeric\_cols) >= 2:  
 exports\_col = numeric\_cols[-1] # Assuming the last column is exports  
 imports\_col = numeric\_cols[-2] # Assuming the second last column is imports  
  
 # Group by a specific column (if exists) or use the entire dataframe  
 if 'Country' in df.columns:  
 country\_exports = df.groupby('Country')[exports\_col].sum().sort\_values(ascending=False).head(10)  
 country\_imports = df.groupby('Country')[imports\_col].sum().sort\_values(ascending=False).head(10)  
 else:  
 country\_exports = df[exports\_col].sort\_values(ascending=False).head(10)  
 country\_imports = df[imports\_col].sort\_values(ascending=False).head(10)  
  
 # Create bar charts  
 fig, ax = plt.subplots(2, figsize=(10, 8))  
  
 if 'Country' in df.columns:  
 ax[0].bar(country\_exports.index, country\_exports.values)  
 ax[0].set\_title(f'Top 10 Countries by Total {exports\_col}')  
 ax[0].set\_xlabel('Country')  
 ax[0].set\_ylabel(exports\_col)  
  
 ax[1].bar(country\_imports.index, country\_imports.values)  
 ax[1].set\_title(f'Top 10 Countries by Total {imports\_col}')  
 ax[1].set\_xlabel('Country')  
 ax[1].set\_ylabel(imports\_col)  
 else:  
 ax[0].bar(range(len(country\_exports)), country\_exports.values)  
 ax[0].set\_title(f'Top 10 Values by {exports\_col}')  
 ax[0].set\_xlabel('Index')  
 ax[0].set\_ylabel(exports\_col)  
  
 ax[1].bar(range(len(country\_imports)), country\_imports.values)  
 ax[1].set\_title(f'Top 10 Values by {imports\_col}')  
 ax[1].set\_xlabel('Index')  
 ax[1].set\_ylabel(imports\_col)  
  
 plt.tight\_layout()  
 plt.show()  
else:  
 print("Not enough numeric columns for visualization.")

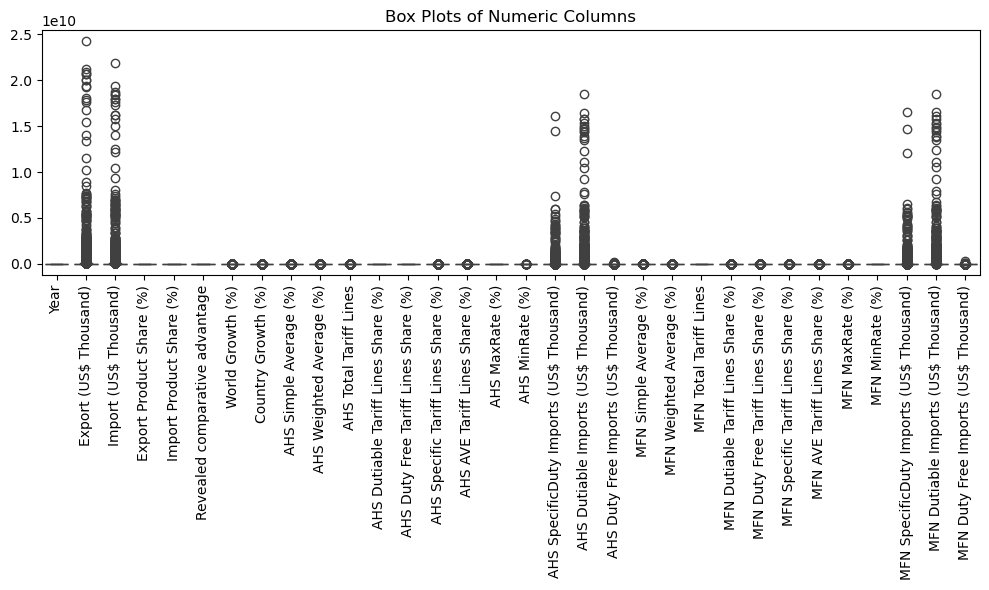
Partner Name Year Export (US$ Thousand) Import (US$ Thousand) \  
0 Aruba 1988 3498.10 328.49   
1 Afghanistan 1988 213030.40 54459.52   
2 Angola 1988 375527.89 370702.76   
3 Anguila 1988 366.98 4.00   
4 Albania 1988 30103.56 47709.30   
  
 Export Product Share (%) Import Product Share (%) \  
0 100.0 100   
1 100.0 100   
2 100.0 100   
3 100.0 100   
4 100.0 100   
  
 Revealed comparative advantage World Growth (%) Country Growth (%) \  
0 NaN NaN NaN   
1 NaN NaN NaN   
2 NaN NaN NaN   
3 NaN NaN NaN   
4 NaN NaN NaN   
  
 AHS Simple Average (%) ... MFN Total Tariff Lines \  
0 2.80 ... 1152.0   
1 0.88 ... 4142.0   
2 2.02 ... 5438.0   
3 3.71 ... 322.0   
4 1.84 ... 5684.0   
  
 MFN Dutiable Tariff Lines Share (%) MFN Duty Free Tariff Lines Share (%) \  
0 63.54 22.74   
1 69.41 15.64   
2 76.00 16.27   
3 66.15 22.05   
4 66.87 19.19   
  
 MFN Specific Tariff Lines Share (%) MFN AVE Tariff Lines Share (%) \  
0 70.32 31.61   
1 72.45 40.51   
2 41.55 24.80   
3 78.79 36.36   
4 57.93 48.52   
  
 MFN MaxRate (%) MFN MinRate (%) MFN SpecificDuty Imports (US$ Thousand) \  
0 352.69 0.0 2186.00   
1 2029.66 0.0 78436.91   
2 451.15 0.0 727741.99   
3 100.00 0.0 94.00   
4 3000.00 0.0 37904.09   
  
 MFN Dutiable Imports (US$ Thousand) MFN Duty Free Imports (US$ Thousand)   
0 3128.02 0.0   
1 94191.50 0.0   
2 921481.52 0.0   
3 583.00 0.0   
4 101195.95 0.0   
  
[5 rows x 33 columns]  
Index(['Partner Name', 'Year', 'Export (US$ Thousand)',  
 'Import (US$ Thousand)', 'Export Product Share (%)',  
 'Import Product Share (%)', 'Revealed comparative advantage',  
 'World Growth (%)', 'Country Growth (%)', 'AHS Simple Average (%)',  
 'AHS Weighted Average (%)', 'AHS Total Tariff Lines',  
 'AHS Dutiable Tariff Lines Share (%)',  
 'AHS Duty Free Tariff Lines Share (%)',  
 'AHS Specific Tariff Lines Share (%)', 'AHS AVE Tariff Lines Share (%)',  
 'AHS MaxRate (%)', 'AHS MinRate (%)',  
 'AHS SpecificDuty Imports (US$ Thousand)',  
 'AHS Dutiable Imports (US$ Thousand)',  
 'AHS Duty Free Imports (US$ Thousand)', 'MFN Simple Average (%)',  
 'MFN Weighted Average (%)', 'MFN Total Tariff Lines',  
 'MFN Dutiable Tariff Lines Share (%)',  
 'MFN Duty Free Tariff Lines Share (%)',  
 'MFN Specific Tariff Lines Share (%)', 'MFN AVE Tariff Lines Share (%)',  
 'MFN MaxRate (%)', 'MFN MinRate (%)',  
 'MFN SpecificDuty Imports (US$ Thousand)',  
 'MFN Dutiable Imports (US$ Thousand)',  
 'MFN Duty Free Imports (US$ Thousand)'],  
 dtype='object')



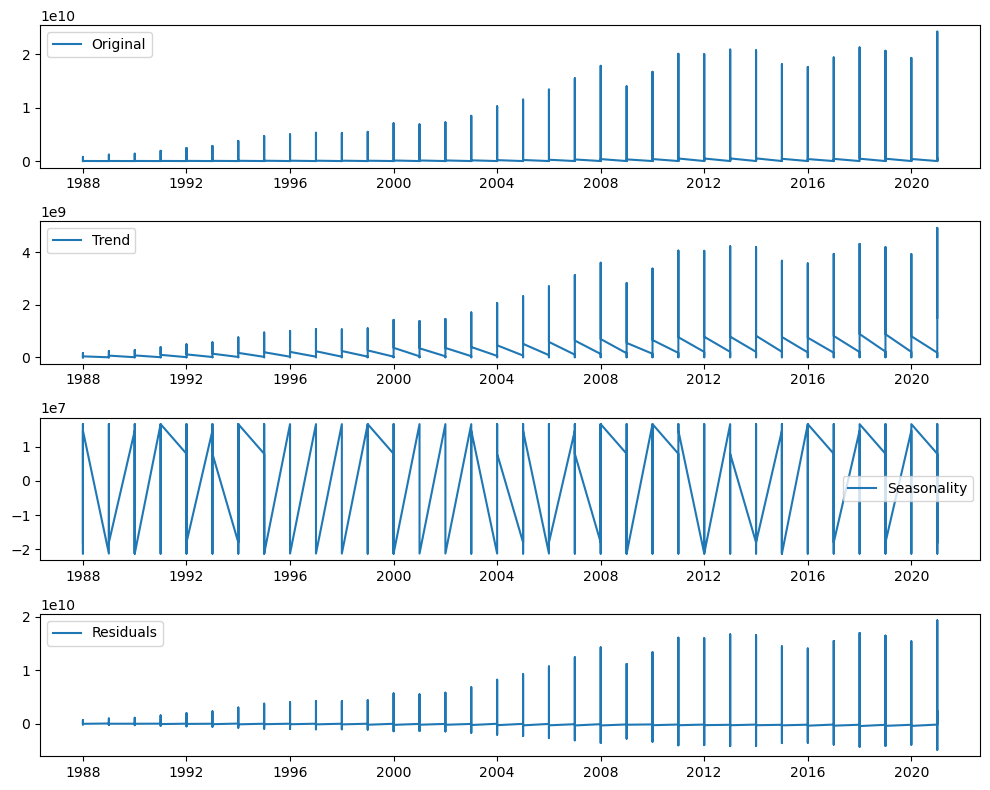
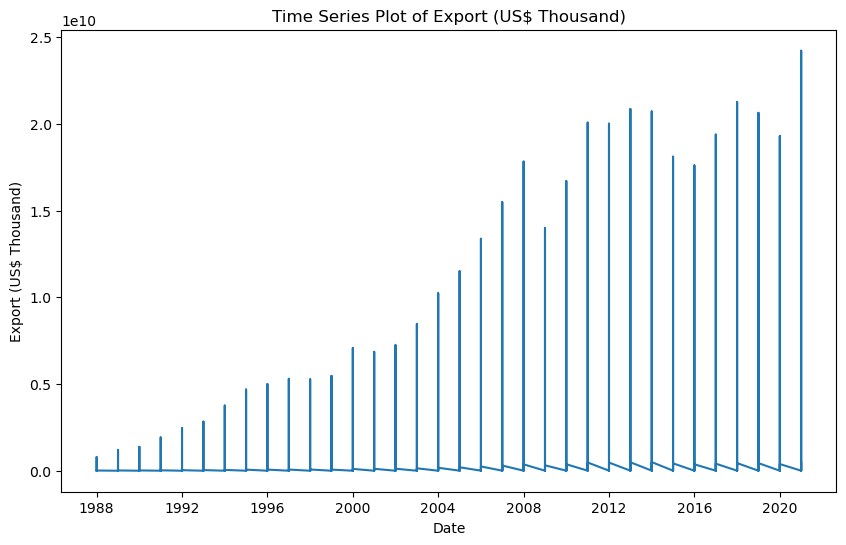
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
# Load the dataset  
df = pd.read\_csv(r"C:\Users\Prem Panchal\Downloads\34\_years\_world\_export\_import\_dataset.csv")  
  
# Select numeric columns  
numeric\_df = df.select\_dtypes(include=['int64', 'float64'])  
  
# Calculate correlation matrix  
corr\_matrix = numeric\_df.corr()  
  
# Create a heatmap  
plt.figure(figsize=(10, 8))  
sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm', square=True)  
plt.title('Correlation Heatmap')  
plt.show()



#Box Plot   
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
# Load the dataset  
df = pd.read\_csv(r"C:\Users\Prem Panchal\Downloads\34\_years\_world\_export\_import\_dataset.csv")  
  
# Select numeric columns  
numeric\_df = df.select\_dtypes(include=['int64', 'float64'])  
  
# Create box plots  
fig, ax = plt.subplots(figsize=(10, 6))  
sns.boxplot(data=numeric\_df)  
plt.title('Box Plots of Numeric Columns')  
plt.xticks(rotation=90)  
plt.tight\_layout()  
plt.show()



#Time series Analysis   
import pandas as pd  
import matplotlib.pyplot as plt  
from statsmodels.tsa.seasonal import seasonal\_decompose  
from statsmodels.tsa.stattools import adfuller  
  
# Load the dataset  
df = pd.read\_csv(r"C:\Users\Prem Panchal\Downloads\34\_years\_world\_export\_import\_dataset.csv")  
  
# Convert date column to datetime format (if it exists)  
if 'Year' in df.columns:  
 df['Year'] = pd.to\_datetime(df['Year'], format='%Y')  
 df.set\_index('Year', inplace=True)  
  
 # Select a numeric column for analysis  
 numeric\_col = df.select\_dtypes(include=['int64', 'float64']).columns[0]  
 ts = df[numeric\_col]  
  
 # Plot the time series  
 plt.figure(figsize=(10, 6))  
 plt.plot(ts)  
 plt.title(f'Time Series Plot of {numeric\_col}')  
 plt.xlabel('Date')  
 plt.ylabel(numeric\_col)  
 plt.show()  
  
 # Perform seasonal decomposition with a specified period  
 period = 5 # Adjust this value based on your data  
 decomposition = seasonal\_decompose(ts, model='additive', period=period)  
 trend = decomposition.trend  
 seasonal = decomposition.seasonal  
 residual = decomposition.resid  
  
 plt.figure(figsize=(10, 8))  
 plt.subplot(411)  
 plt.plot(ts, label='Original')  
 plt.legend(loc='best')  
 plt.subplot(412)  
 plt.plot(trend, label='Trend')  
 plt.legend(loc='best')  
 plt.subplot(413)  
 plt.plot(seasonal,label='Seasonality')  
 plt.legend(loc='best')  
 plt.subplot(414)  
 plt.plot(residual, label='Residuals')  
 plt.legend(loc='best')  
 plt.tight\_layout()  
 plt.show()  
  
 # Perform Augmented Dickey-Fuller test for stationarity  
 result = adfuller(ts)  
 print('ADF Statistic:', result[0])  
 print('p-value:', result[1])  
else:  
 print("No date column found in the dataset.")



ADF Statistic: -13.057911428738935  
p-value: 2.076297118730091e-24