World Trade is Good, World tariffs are bad

Module 10 Individual Project 1

Tony Chan

chanto@oregonstate.edu

Candidate for Bachelor of Science in Computer Science

CS332 Applied Data Science

3/14/2025

Table of Contents

Table of Contents

Introduction: The Story of Trade: How the Flow of Goods Shapes the World

Note: 1 Global trade connects nations through the exchange of goods and services.

Trade is one of the oldest human activities, dating back thousands of years to ancient civilizations that exchanged spices, textiles, and precious metals across vast distances. Today, trade is the engine that drives the global economy. From the smartphones in our pockets to the coffee in our cups, nearly everything we use has traveled across borders before reaching us. Countries engage in trade to access products they don't produce, to sell goods they specialize in, and to strengthen economic ties with their neighbors. Trade fosters cooperation, fuels economic growth, and ultimately shapes the prosperity of nations.

But trade doesn't come without challenges. Governments use tariffs—taxes on imported goods—to protect domestic industries, control market competition, or respond to political conflicts.

While tariffs may serve a purpose, they can also slow down trade, increase prices for consumers, and disrupt industries that rely on global supply chains. The history of tariffs shows how they have influenced economies, from the Great Depression to modern trade wars between major powers like the United States and China. Understanding tariffs is not just about economics—it's about how policies affect everyday life, from the cost of groceries to the availability of new technology.

The importance of trade has grown in recent years as the world becomes more interconnected. Inflation, economic downturns, and supply chain disruptions make trade a hot topic for policymakers and businesses alike. Countries that impose high tariffs risk reducing the flow of goods, which can lead to price hikes and economic instability. On the other hand, free trade agreements and low tariffs have allowed economies to flourish. As global markets shift, it's crucial to ask: Who benefits from trade? Who suffers? And what role do tariffs play in shaping the global economy?

This data story aims to explore the role of trade and tariffs by analyzing patterns over time, comparing different countries, and uncovering the real-world impact of trade policies. How do tariffs affect inflation? What happens when trade between countries slows down? By looking at historical trends, we can gain insights into the future of trade and understand the balance between protecting national industries and promoting economic growth. Through this journey, we'll see how trade shapes economies and, in turn, the lives of people around the world.

As global markets evolve, understanding trade and tariffs becomes essential. In this project, we will explore how trade policies impact economies, consumers, and businesses, using real-world data to uncover patterns and trends.

Introduction: Why This Study Matters

topic introduction

Despite its significance, world trade is often overlooked in daily conversations. Yet, it plays a crucial role in shaping economies and livelihoods. The largest economies in the world—like the United States, China, and Japan—became so big because of trade with other countries. By buying and selling goods and services across borders, they have grown their industries and created jobs. World trade helps countries share resources and benefit from each other's strengths.

why you chose the topic

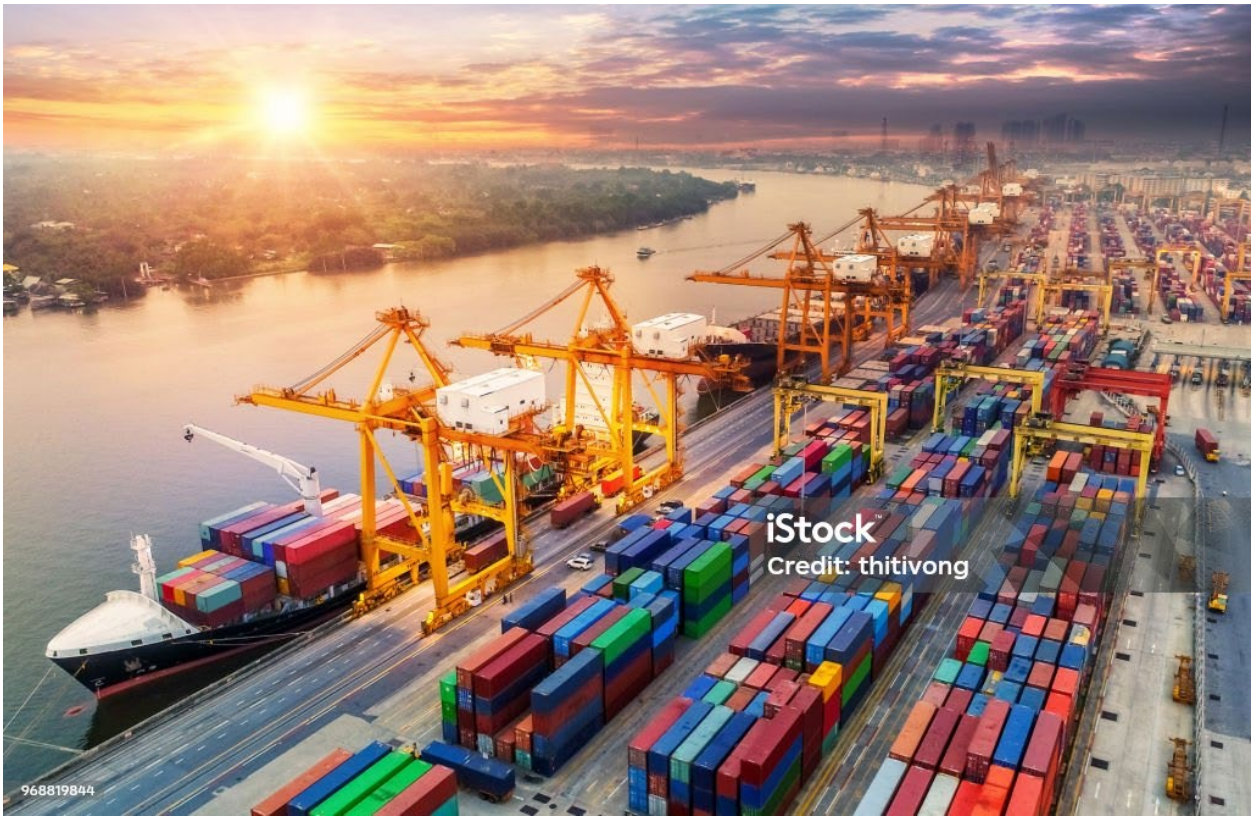
Tariffs are going to be an important topic in global politics over the next few years and probably beyond. While people often talk about business and companies, trade between countries isn't something you usually hear about at the dinner table. I want to learn more about trade facts before any trade conflicts arise. Information about trade between countries is stored in government databases, but it takes some effort to find and understand it. Knowing these facts can help us better understand how trade impacts our lives and the economy.

We can look back in time to see when there were hi tariff levels to see if inflation increased in reaction. Inflation has been on the minds of Americans this past election and will continue to linger. The tariff level for each country and product could exacerbate inflation.

It is not talked about much in the US but other countries can impose higher tariffs on US products which could cause an escalation of tensions.

at least one image to support the topic

Figure 1 Cargo Container Loading Represents World Trade



Note: 2 Logistics and transportation of Container Cargo ship and Cargo plane... (2018, June 5). iStock. https://www.istockphoto.com/photo/logistics-and-transportation-of-container-cargo-ship-and-cargo-plane-with-working-gm968819844-264102201

For each question, give initial ideas about how you might answer the question

For all the questions below, the dataset has column data spanning 40 years. The data can be sliced into even more detail. We will be able to compare different tariffs by categories of products, by country and by time. The dataset comes with column labels and an explanation of the variable names.

10 questions that you intend to research about your topic

1. Which country is has the largest trade?

2. Which countries have been increasing and decreasing imports for the last 40 years?

3. Which countries have been increasing and decreasing exports for the last 40 years?

4. What is the percentage change in the total value of a country's trade (exports and imports) compared to the previous year.

5. What is the simple average tariff rate applied by the country based on the Harmonized System (AHS) classification of products.

6. What is the highest tariff rate applied by the country on any product category in the Harmonized System.

7. What is the total value of imports that are duty-free in thousands of US dollars.

8. What is the simple average most-favored-nation (MFN) tariff rate applied by the country.

9. What is the total number of tariff lines or product categories for which MFN tariff rates are applied.

10. What is the percentage of MFN tariff lines with specific (fixed) duty rates.

Data Gathering

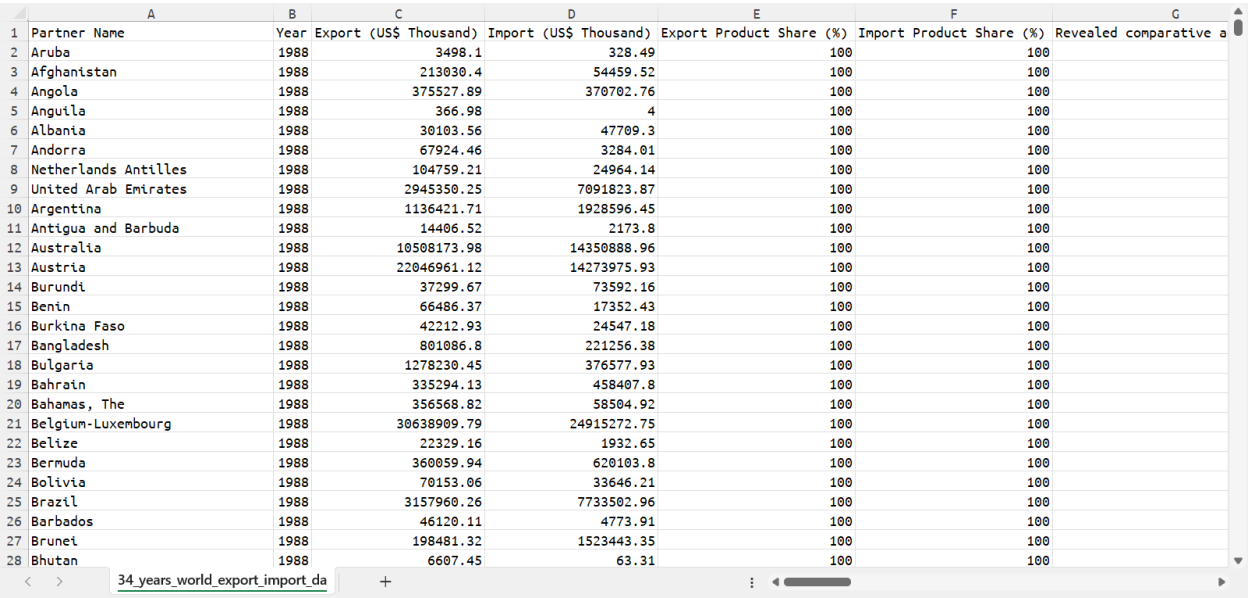
The dataset you gathered

The "World Export & Import Dataset (1989 - 2023)" is a comprehensive collection of data related to international trade and trade policies. This dataset is designed to provide insights and analysis for researchers, policymakers, and analysts interested in understanding global trade dynamics. It covers various aspects of trade, including trade values, tariff rates, and trade policy indicators for numerous countries over multiple years.

The dataset may not be in good condition to use right away. In order to clean up the dataset, I will use data cleaning methods to standardize data types, exploratory data methods to understand the data, feature engineering to prepare the data for models, statistical models to spot any trends, machine learning models to prepare data for advanced analysis and basic programming libraries for easier Python programming.

a screen image of the dataset

Figure 2 Screenshot of Import / Export Dataset from Kaggle



	A	B	C	D	E	F	G
1	Partner Name	Year	Export (US$ Thousand)	Import (US$ Thousand)	Export Product Share (%)	Import Product Share (%)	Revealed comparative a
2	Aruba	1988	3498.1	328.49	100	100	
3	Afghanistan	1988	213030.4	54459.52	100	100	
4	Angola	1988	375527.89	370702.76	100	100	
5	Anguila	1988	366.98	4	100	100	
6	Albania	1988	30103.56	47709.3	100	100	
7	Andorra	1988	67924.46	3284.01	100	100	
8	Netherlands Antilles	1988	104759.21	24964.14	100	100	
9	United Arab Emirates	1988	2945350.25	7091823.87	100	100	
10	Argentina	1988	1136421.71	1928596.45	100	100	
11	Antigua and Barbuda	1988	14406.52	2173.8	100	100	
12	Australia	1988	10508173.98	14350888.96	100	100	
13	Austria	1988	22046961.12	14273975.93	100	100	
14	Burundi	1988	37299.67	73592.16	100	100	
15	Benin	1988	66486.37	17352.43	100	100	
16	Burkina Faso	1988	42212.93	24547.18	100	100	
17	Bangladesh	1988	801086.8	221256.38	100	100	
18	Bulgaria	1988	1278230.45	376577.93	100	100	
19	Bahrain	1988	335294.13	458407.8	100	100	
20	Bahamas, The	1988	356568.82	58504.92	100	100	
21	Belgium-Luxembourg	1988	30638909.79	24915272.75	100	100	
22	Belize	1988	22329.16	1932.65	100	100	
23	Bermuda	1988	360059.94	620103.8	100	100	
24	Bolivia	1988	70153.06	33646.21	100	100	
25	Brazil	1988	3157960.26	7733502.96	100	100	
26	Barbados	1988	46120.11	4773.91	100	100	
27	Brunei	1988	198481.32	1523443.35	100	100	
28	Bhutan	1988	6607.45	63.31	100	100	

34_years_world_export_import_da

a link to the dataset

The dataset is from Kaggle.

https://www.kaggle.com/datasets/muhammadtalhaawan/world-export-and-import-dataset

an explanation of the variables in the dataset

The main data fields are

1. Partner Name: This feature represents the name of the trading partner or country with which import and export data is being measured.

2. Year: This feature specifies the specific year for which the import and export data is recorded.

3. Export (US$ Thousand): This feature represents the total value of goods and products exported by the country in thousands of US dollars.

4. Import (US$ Thousand): Similar to exports, this feature represents the total value of goods and products imported by the country in thousands of US dollars.

5. Export Product Share (%): This feature indicates the percentage of the country's total exports that a specific product category represents.

6. Import Product Share (%): Like export product share, this feature represents the percentage of the country's total imports that a specific product category represents.

7. Revealed Comparative Advantage: This feature is likely a qualitative measure of whether a country has a comparative advantage in a particular product category.

8. World Growth (%): This feature indicates the percentage change in the total value of world trade (exports and imports) compared to the previous year.

Direct Data Downloads

Examples of how and where direct download datasets, datasets through an API and datasets through web scraping come from the following.

Example. Dataweb (dataweb.usitc.gov)

Description

Dataweb is an official website of the United States government. The USITC DataWeb provides public access to the official U.S. import and export statistics of the U.S. Department of Commerce in a user-friendly web interface. Using the DataWeb querying tool, users can build custom queries and access these data in a spreadsheet or a web-based format. The tool enables users to query data by key parameters such as the trade flow (e.g., imports, exports, trade balance), various measures (e.g., value, quantity), timeframes (e.g., monthly, quarterly, annual, year-to-date, even custom time periods), trading partners, product classifications (e.g., HTS, SITC, or NAICS), product details (up to the most granular basis available, or the 10-digit HTS or Schedule B statistical reporting number level), and additional data

descriptors like duty-rate provision codes, special import programs (e.g., unilateral preference programs, free trade agreements), customs districts, etc.

Visualize with screen images

Figure 3

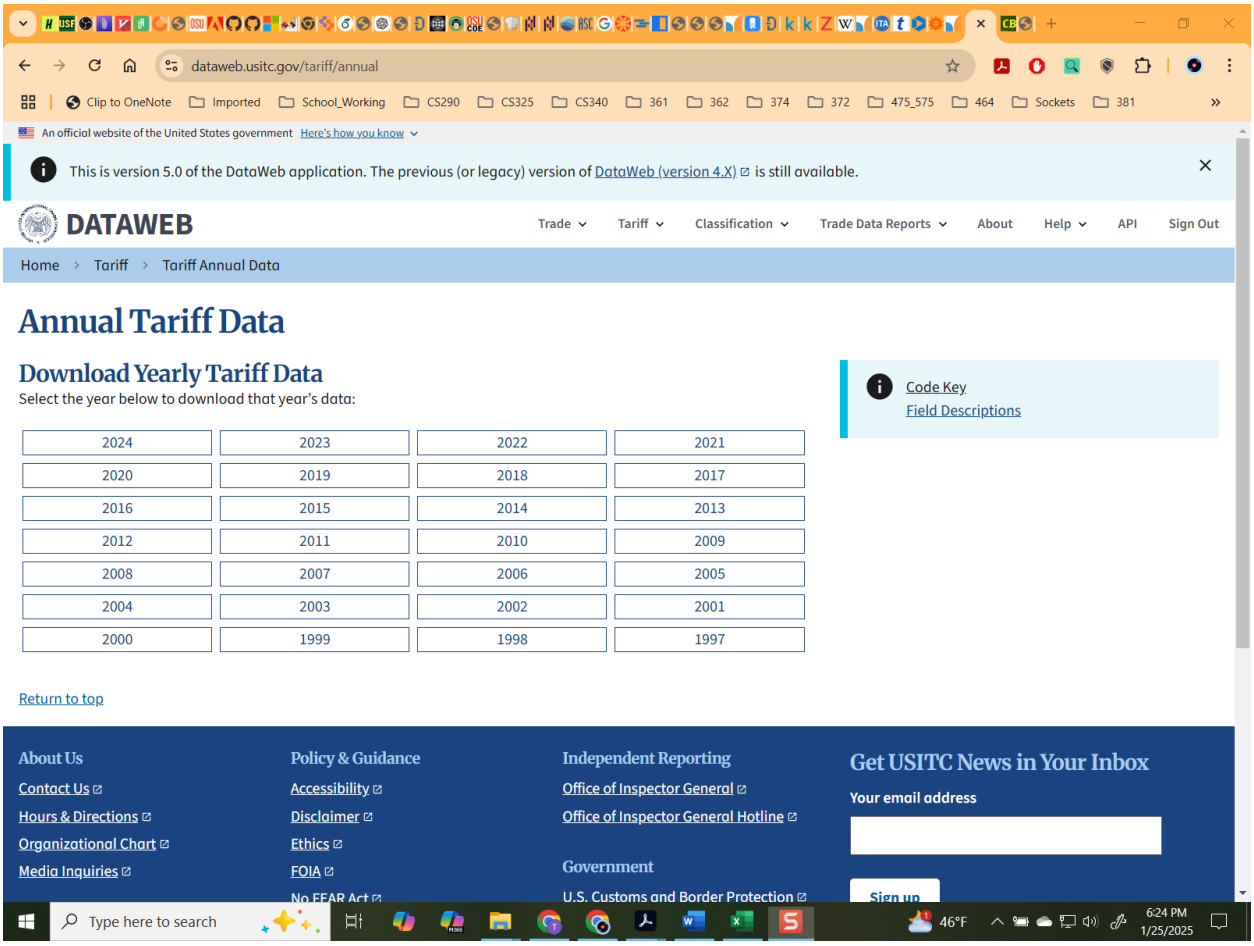


Figure 4

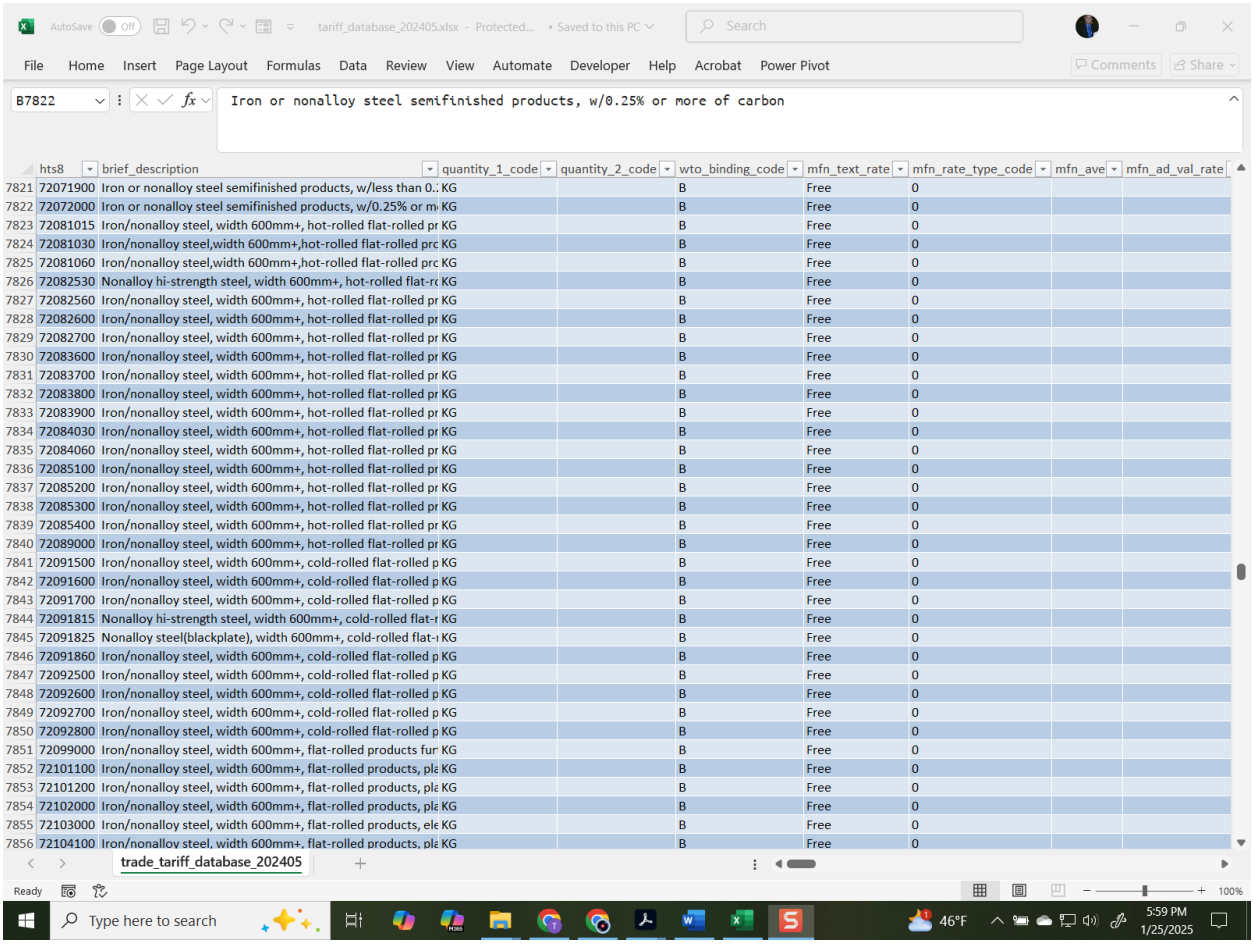
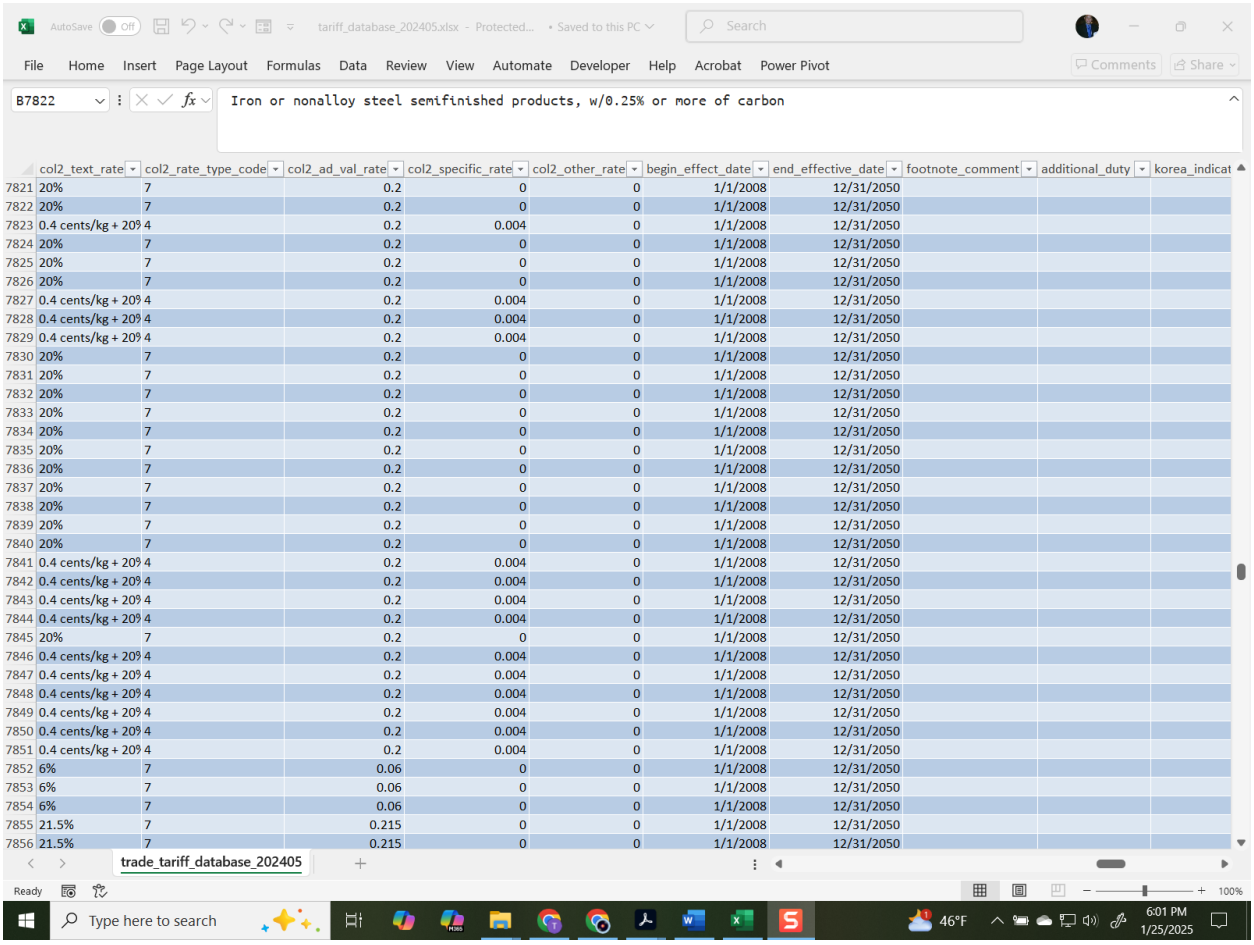

tariff_database_202405.xlsx - Protected... • Saved to this PC
Search
File Home Insert Page Layout Formulas Data Review View Automate Developer Help Acrobat Power Pivot
Comments Share
B7822
Iron or nonalloy steel semifinished products, w/0.25% or more of carbon
hts8 brief_description quantity_1_code quantity_2_code wto_binding_code mfn_text_rate mfn_rate_type_code mfn_ave mfn_ad_val_rate
7821 72071900 Iron or nonalloy steel semifinished products, w/less than 0.: KG B Free 0
7822 72072000 Iron or nonalloy steel semifinished products, w/0.25% or m KG B Free 0
7823 72081015 Iron/nonalloy steel, width 600mm+, hot-rolled flat-rolled pr KG B Free 0
7824 72081030 Iron/nonalloy steel,width 600mm+,hot-rolled flat-rolled prc KG B Free 0
7825 72081060 Iron/nonalloy steel,width 600mm+,hot-rolled flat-rolled prc KG B Free 0
7826 72082530 Nonalloy hi-strength steel, width 600mm+, hot-rolled flat-rc KG B Free 0
7827 72082560 Iron/nonalloy steel, width 600mm+, hot-rolled flat-rolled pr KG B Free 0
7828 72082600 Iron/nonalloy steel, width 600mm+, hot-rolled flat-rolled pr KG B Free 0
7829 72082700 Iron/nonalloy steel, width 600mm+, hot-rolled flat-rolled pr KG B Free 0
7830 72083600 Iron/nonalloy steel, width 600mm+, hot-rolled flat-rolled pr KG B Free 0
7831 72083700 Iron/nonalloy steel, width 600mm+, hot-rolled flat-rolled pr KG B Free 0
7832 72083800 Iron/nonalloy steel, width 600mm+, hot-rolled flat-rolled pr KG B Free 0
7833 72083900 Iron/nonalloy steel, width 600mm+, hot-rolled flat-rolled pr KG B Free 0
7834 72084030 Iron/nonalloy steel, width 600mm+, hot-rolled flat-rolled pr KG B Free 0
7835 72084060 Iron/nonalloy steel, width 600mm+, hot-rolled flat-rolled pr KG B Free 0
7836 72085100 Iron/nonalloy steel, width 600mm+, hot-rolled flat-rolled pr KG B Free 0
7837 72085200 Iron/nonalloy steel, width 600mm+, hot-rolled flat-rolled pr KG B Free 0
7838 72085300 Iron/nonalloy steel, width 600mm+, hot-rolled flat-rolled pr KG B Free 0
7839 72085400 Iron/nonalloy steel, width 600mm+, hot-rolled flat-rolled pr KG B Free 0
7840 72089000 Iron/nonalloy steel, width 600mm+, hot-rolled flat-rolled pr KG B Free 0
7841 72091500 Iron/nonalloy steel, width 600mm+, cold-rolled flat-rolled p KG B Free 0
7842 72091600 Iron/nonalloy steel, width 600mm+, cold-rolled flat-rolled p KG B Free 0
7843 72091700 Iron/nonalloy steel, width 600mm+, cold-rolled flat-rolled p KG B Free 0
7844 72091815 Nonalloy hi-strength steel, width 600mm+, cold-rolled flat-r KG B Free 0
7845 72091825 Nonalloy steel(blackplate), width 600mm+, cold-rolled flat-r KG B Free 0
7846 72091860 Iron/nonalloy steel, width 600mm+, cold-rolled flat-rolled p KG B Free 0
7847 72092500 Iron/nonalloy steel, width 600mm+, cold-rolled flat-rolled p KG B Free 0
7848 72092600 Iron/nonalloy steel, width 600mm+, cold-rolled flat-rolled p KG B Free 0
7849 72092700 Iron/nonalloy steel, width 600mm+, cold-rolled flat-rolled p KG B Free 0
7850 72092800 Iron/nonalloy steel, width 600mm+, cold-rolled flat-rolled p KG B Free 0
7851 72099000 Iron/nonalloy steel, width 600mm+, flat-rolled products fur KG B Free 0
7852 72101100 Iron/nonalloy steel, width 600mm+, flat-rolled products, pla KG B Free 0
7853 72101200 Iron/nonalloy steel, width 600mm+, flat-rolled products, pla KG B Free 0
7854 72102000 Iron/nonalloy steel, width 600mm+, flat-rolled products, pla KG B Free 0
7855 72103000 Iron/nonalloy steel, width 600mm+, flat-rolled products, ele KG B Free 0
7856 72104100 Iron/nonalloy steel, width 600mm+, flat-rolled products, pla KG B Free 0
trade_tariff_database_202405
Ready
100%
Type here to search
46°F
5:59 PM
1/25/2025

Figure 5



Link to dataset

https://dataweb.usitc.gov/tariff/annual

Discussion Qualitative, Quantitative, Mixed, Labeled and more

There are thousands of lines in Figure 4 and 5 but above screenshots are of tariffs for steel. The average person does not know how complicated international trade is. The Harmonized System code labeled in first column is a glimpse at all the codes, and the description describes how detailed steel is divided into many categories. Steel can be broken down by how much carbon it contains and different tariff rates can apply to the amount of carbon in your steel. The data is both qualitative and quantitative. It is labeled data in the form of a csv file.

Example. The United States Census Bureau (census.gov)

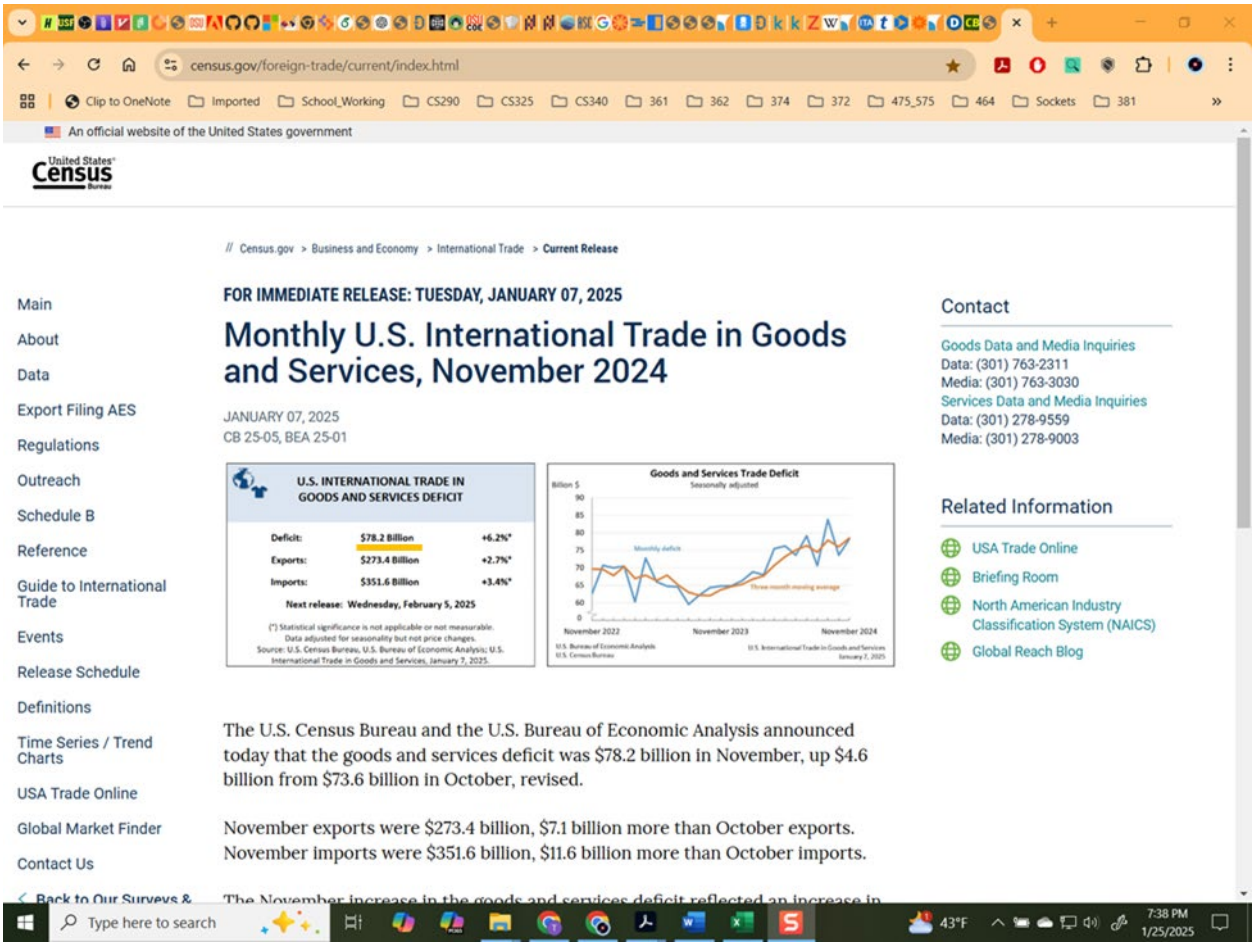
Description

This publication is intended to serve as a guide to the various sources of foreign trade statistics and to inform users of the content and general arrangement of the data. The foreign trade statistics program is unique among the Census Bureau's economic statistics programs in that the information is not collected from forms sent to respondents soliciting responses as in the case of surveys. Rather, the information is compiled from automated forms and reports.

The Census Bureau offers a mass amount of data but will focus on economic indicators and international trade data. The data as seen on the website can also be downloaded as an excel spreadsheet or text file. I will probably focus on overview data such as trade deficit, export totals and import totals for now.

Visualize with screen images

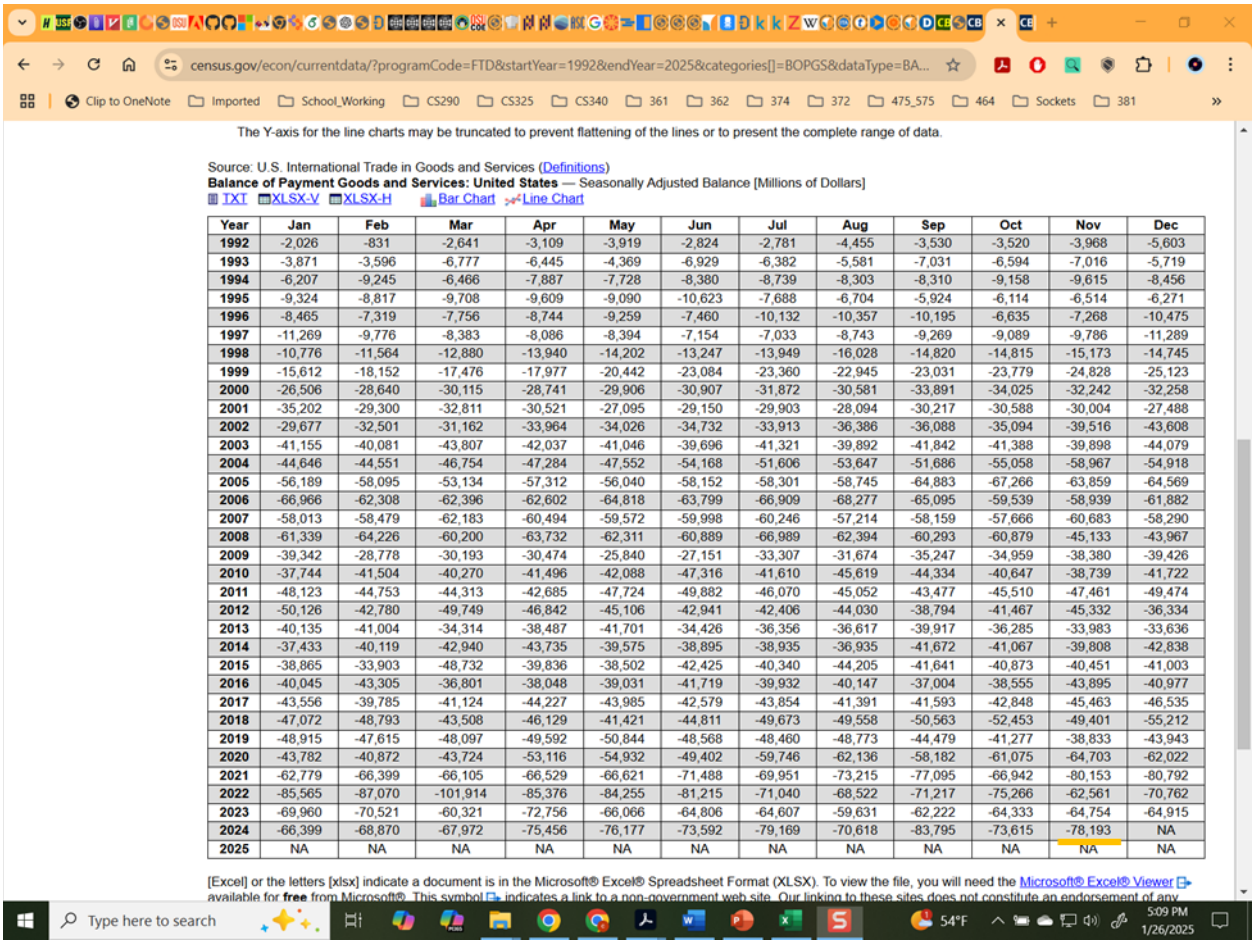
Figure 6



Link to dataset

https://www.census.gov/foreign-trade/current/index.html

Figure 7



census.gov/econ/currentdata/?programCode=FTD&startYear=1992&endYear=2025&categories[]=BOPGS&dataType=BA...

The Y-axis for the line charts may be truncated to prevent flattening of the lines or to present the complete range of data.

Source: U.S. International Trade in Goods and Services (Definitions)

Balance of Payment Goods and Services: United States — Seasonally Adjusted Balance [Millions of Dollars]

TXT XLSX-V XLSX-H Bar Chart Line Chart

Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1992	-2,026	-831	-2,641	-3,109	-3,919	-2,824	-2,781	-4,455	-3,530	-3,520	-3,968	-5,603
1993	-3,871	-3,596	-6,777	-6,445	-4,369	-6,929	-6,382	-5,581	-7,031	-6,594	-7,016	-5,719
1994	-6,207	-9,245	-6,466	-7,887	-7,728	-8,380	-8,739	-8,303	-8,310	-9,158	-9,615	-8,456
1995	-9,324	-8,817	-9,708	-9,609	-9,090	-10,623	-7,688	-6,704	-5,924	-6,114	-6,514	-6,271
1996	-8,465	-7,319	-7,756	-8,744	-9,259	-7,460	-10,132	-10,357	-10,195	-6,635	-7,268	-10,475
1997	-11,269	-9,776	-8,383	-8,086	-8,394	-7,154	-7,033	-8,743	-9,269	-9,089	-9,786	-11,289
1998	-10,776	-11,564	-12,880	-13,940	-14,202	-13,247	-13,949	-16,028	-14,820	-14,815	-15,173	-14,745
1999	-15,612	-18,152	-17,476	-17,977	-20,442	-23,084	-23,360	-22,945	-23,031	-23,779	-24,828	-25,123
2000	-26,506	-28,640	-30,115	-28,741	-29,906	-30,907	-31,872	-30,581	-33,891	-34,025	-32,242	-32,258
2001	-35,202	-29,300	-32,811	-30,521	-27,095	-29,150	-29,903	-28,094	-30,217	-30,588	-30,004	-27,488
2002	-29,677	-32,501	-31,162	-33,964	-34,026	-34,732	-33,913	-36,386	-36,088	-35,094	-39,516	-43,608
2003	-41,155	-40,081	-43,807	-42,037	-41,046	-39,696	-41,321	-39,892	-41,842	-41,388	-39,898	-44,079
2004	-44,646	-44,551	-46,754	-47,284	-47,552	-54,168	-51,606	-53,647	-51,686	-55,058	-58,967	-54,918
2005	-56,189	-58,095	-53,134	-57,312	-56,040	-58,152	-58,301	-58,745	-64,883	-67,266	-63,859	-64,569
2006	-66,966	-62,308	-62,396	-62,602	-64,818	-63,799	-66,909	-68,277	-65,095	-59,539	-58,939	-61,882
2007	-58,013	-58,479	-62,183	-60,494	-59,572	-59,998	-60,246	-57,214	-58,159	-57,666	-60,683	-58,290
2008	-61,339	-64,226	-60,200	-63,732	-62,311	-60,889	-66,989	-62,394	-60,293	-60,879	-45,133	-43,967
2009	-39,342	-28,778	-30,193	-30,474	-25,840	-27,151	-33,307	-31,674	-35,247	-34,959	-38,380	-39,426
2010	-37,744	-41,504	-40,270	-41,496	-42,088	-47,316	-41,610	-45,619	-44,334	-40,647	-38,739	-41,722
2011	-48,123	-44,753	-44,313	-42,685	-47,724	-49,882	-46,070	-45,052	-43,477	-45,510	-47,461	-49,474
2012	-50,126	-42,780	-49,749	-46,842	-45,106	-42,941	-42,406	-44,030	-38,794	-41,467	-45,332	-36,334
2013	-40,135	-41,004	-34,314	-38,487	-41,701	-34,426	-36,356	-36,617	-39,917	-36,285	-33,983	-33,636
2014	-37,433	-40,119	-42,940	-43,735	-39,575	-38,895	-38,935	-36,935	-41,672	-41,067	-39,808	-42,838
2015	-38,865	-33,903	-48,732	-39,836	-38,502	-42,425	-40,340	-44,205	-41,641	-40,873	-40,451	-41,003
2016	-40,045	-43,305	-36,801	-38,048	-39,031	-41,719	-39,932	-40,147	-37,004	-38,555	-43,895	-40,977
2017	-43,556	-39,785	-41,124	-44,227	-43,985	-42,579	-43,854	-41,391	-41,593	-42,848	-45,463	-46,535
2018	-47,072	-48,793	-43,508	-46,129	-41,421	-44,811	-49,673	-49,558	-50,563	-52,453	-49,401	-55,212
2019	-48,915	-47,615	-48,097	-49,592	-50,844	-48,568	-48,460	-48,773	-44,479	-41,277	-38,833	-43,943
2020	-43,782	-40,872	-43,724	-53,116	-54,932	-49,402	-59,746	-62,136	-58,182	-61,075	-64,703	-62,022
2021	-62,779	-66,399	-66,105	-66,529	-66,621	-71,488	-69,951	-73,215	-77,095	-66,942	-80,153	-80,792
2022	-85,565	-87,070	-101,914	-85,376	-84,255	-81,215	-71,040	-68,522	-71,217	-75,266	-62,561	-70,762
2023	-69,960	-70,521	-60,321	-72,756	-66,066	-64,806	-64,607	-59,631	-62,222	-64,333	-64,754	-64,915
2024	-66,399	-68,870	-67,972	-75,456	-76,177	-73,592	-79,169	-70,618	-83,795	-73,615	-78,193	NA
2025	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

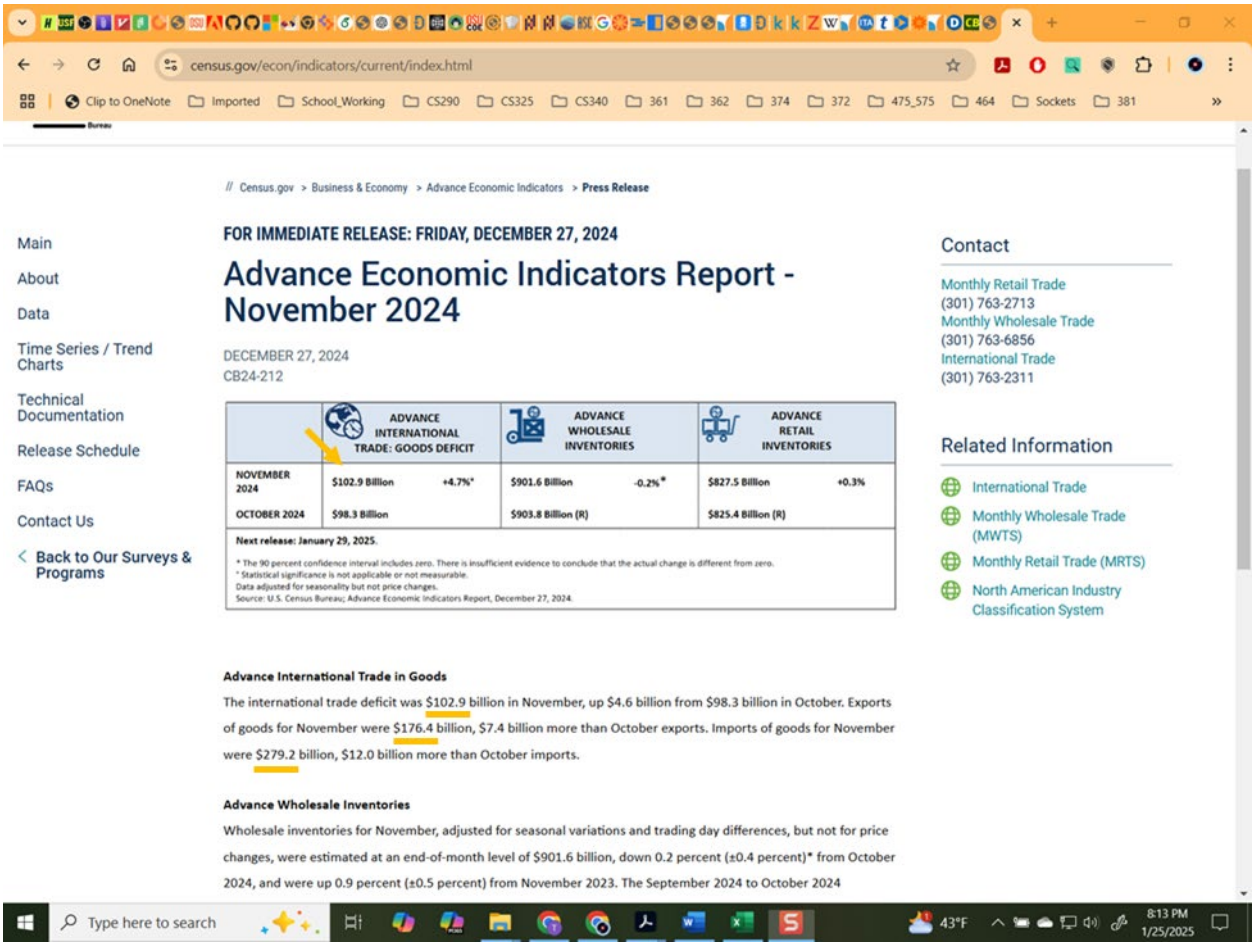
[Excel] or the letters [xlsx] indicate a document is in the Microsoft® Excel® Spreadsheet Format (XLSX). To view the file, you will need the Microsoft® Excel® Viewer available for **free** from Microsoft®. This symbol indicates a link to a non-government web site. Our linking to these sites does not constitute an endorsement of any

Type here to search 54°F 5:09 PM 1/26/2025

Link to dataset

https://www.census.gov/econ/currentdata/?programCode=FTD&startYear=1992&endYear=2025&categories[]=BOPGS&dataType=BAL&geoLevel=US&adjusted=1&notAdjusted=0&errorData=0

Figure 8



Link to dataset

https://www.census.gov/econ/indicators/current/index.html

Figure 9

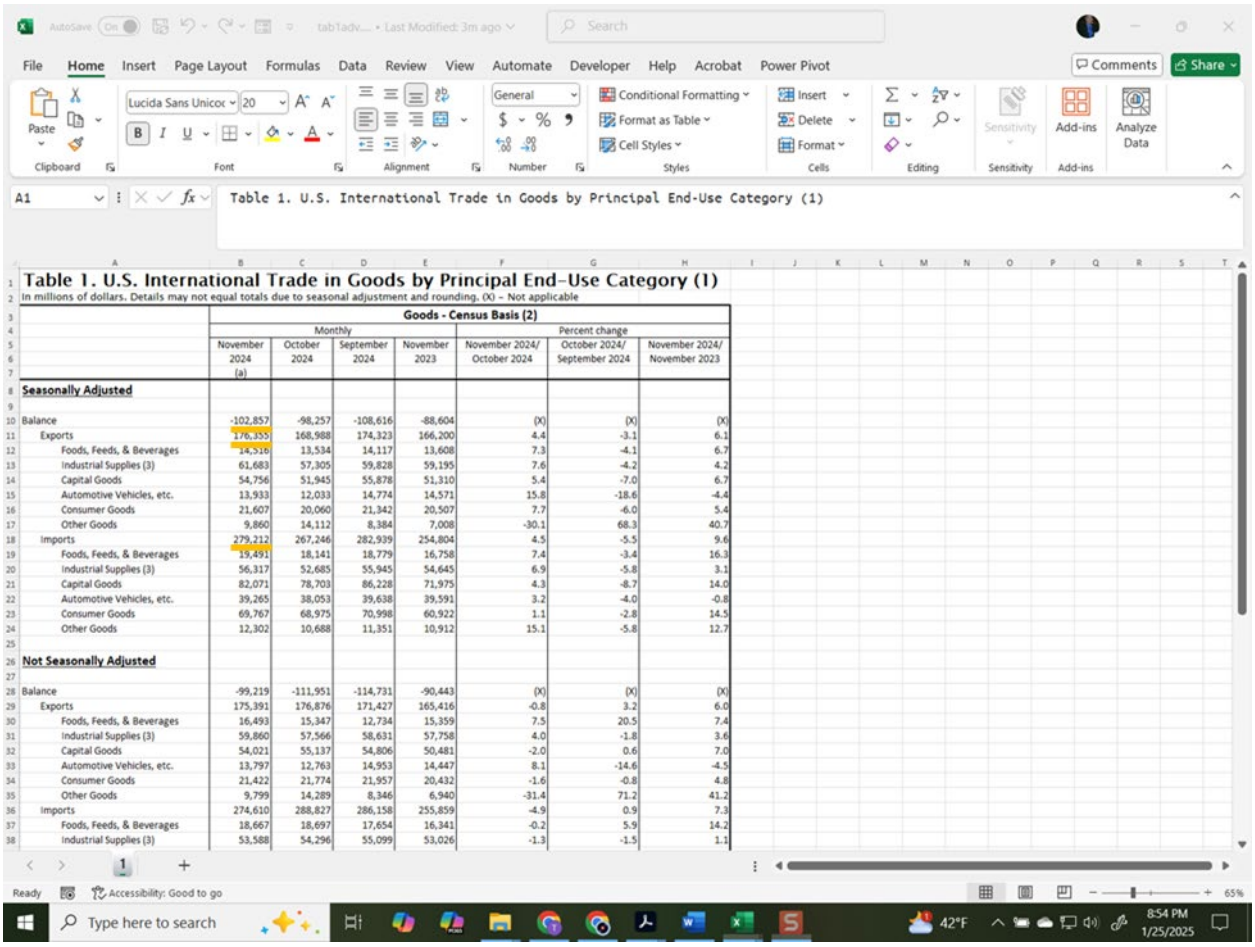


Table 1. U.S. International Trade in Goods by Principal End-Use Category (1)

In millions of dollars. Details may not equal totals due to seasonal adjustment and rounding. (X) - Not applicable

	Goods - Census Basis (2)						
	Monthly				Percent change		
	November 2024 (a)	October 2024	September 2024	November 2023	November 2024/ October 2024	October 2024/ September 2024	November 2024/ November 2023
Seasonally Adjusted							
Balance	-102,857	-98,257	-108,616	-88,604	(X)	(X)	(X)
Exports	176,355	168,988	174,323	166,200	4.4	-3.1	6.1
Foods, Feeds, & Beverages	14,516	13,534	14,117	13,608	7.3	-4.1	6.7
Industrial Supplies (3)	61,683	57,305	59,828	59,195	7.6	-4.2	4.2
Capital Goods	54,756	51,945	55,878	51,310	5.4	-7.0	6.7
Automotive Vehicles, etc.	13,933	12,033	14,774	14,571	15.8	-18.6	-4.4
Consumer Goods	21,607	20,060	21,342	20,507	7.7	-6.0	5.4
Other Goods	9,860	14,112	8,384	7,008	-30.1	68.3	40.7
Imports	279,212	267,246	282,939	254,804	4.5	-5.5	9.6
Foods, Feeds, & Beverages	19,491	18,141	18,779	16,758	7.4	-3.4	16.3
Industrial Supplies (3)	56,317	52,685	55,945	54,645	6.9	-5.8	3.1
Capital Goods	82,071	78,703	86,228	71,975	4.3	-8.7	14.0
Automotive Vehicles, etc.	39,265	38,053	39,638	39,591	3.2	-4.0	-0.8
Consumer Goods	69,767	68,975	70,998	60,922	1.1	-2.8	14.5
Other Goods	12,302	10,688	11,351	10,912	15.1	-5.8	12.7
Not Seasonally Adjusted							
Balance	-99,219	-111,951	-114,731	-90,443	(X)	(X)	(X)
Exports	175,391	176,876	171,427	165,416	-0.8	3.2	6.0
Foods, Feeds, & Beverages	16,493	15,347	12,734	15,359	7.5	20.5	7.4
Industrial Supplies (3)	59,860	57,566	58,631	57,758	4.0	-1.8	3.6
Capital Goods	54,021	55,137	54,806	50,481	-2.0	0.6	7.0
Automotive Vehicles, etc.	13,797	12,763	14,953	14,447	8.1	-14.6	-4.5
Consumer Goods	21,422	21,774	21,957	20,432	-1.6	-0.8	4.8
Other Goods	9,799	14,289	8,346	6,940	-31.4	71.2	41.2
Imports	274,610	288,827	286,158	255,859	-4.9	0.9	7.3
Foods, Feeds, & Beverages	18,667	18,697	17,654	16,341	-0.2	5.9	14.2
Industrial Supplies (3)	53,588	54,296	55,099	53,026	-1.3	-1.5	1.1

Discussion Qualitative, Quantitative, Mixed, Labeled and more

In Figures 6 and 7 the monthly reports are heavily looked upon and could cause a big movement in the stock market. The data has both qualitative and quantitative values. It is also labeled data in the form of a csv file. Some of the key pieces of data come from Figures 6 and 7 on the trade deficit which many economists, investors and politicians follow. The US's biggest trading partners are Canada, China and Mexico. I will try to include some information on them too as soon as I figure out what all the numbers mean.

Example. World Bank WITS World Integrated Trade Solution (https://wits.worldbank.org)

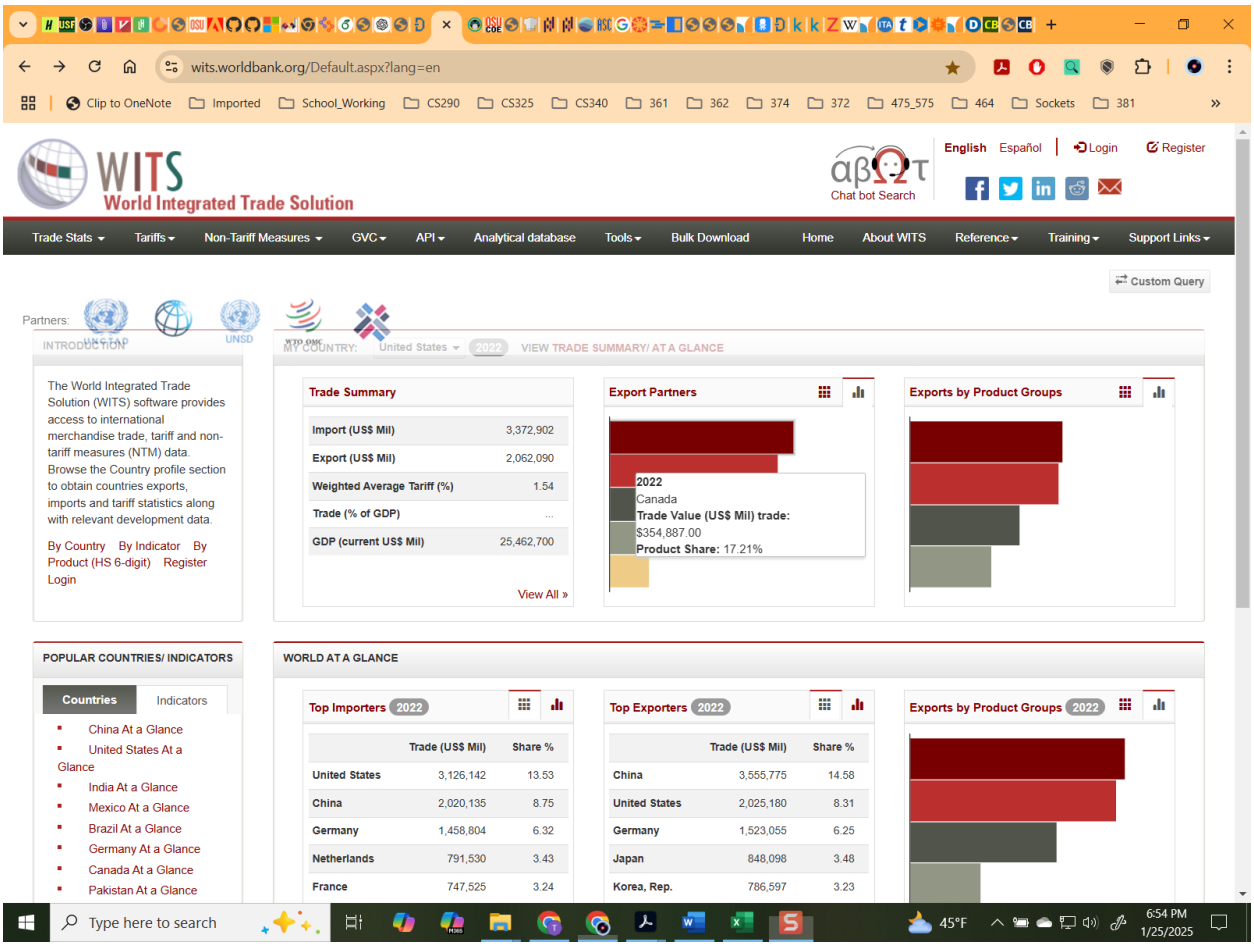
Description

The World Bank — in collaboration with the United Nations Conference on Trade and Development (UNCTAD) and in consultation with organizations such as International Trade Center, United Nations Statistical Division (UNSD) and the World Trade Organization (WTO) — developed the World Integrated Trade Solution (WITS). This software allows users to access and retrieve information on trade and tariffs. WITS is useful to anyone involved in merchandise trade and/or tariff related work specifically researchers, policy makers, governments, academia.

The World Integrated Trade Solution (WITS) software provides access to international merchandise trade, tariff and non-tariff measures (NTM) data. Tariff by country which is what I need.

Visualize with screen images

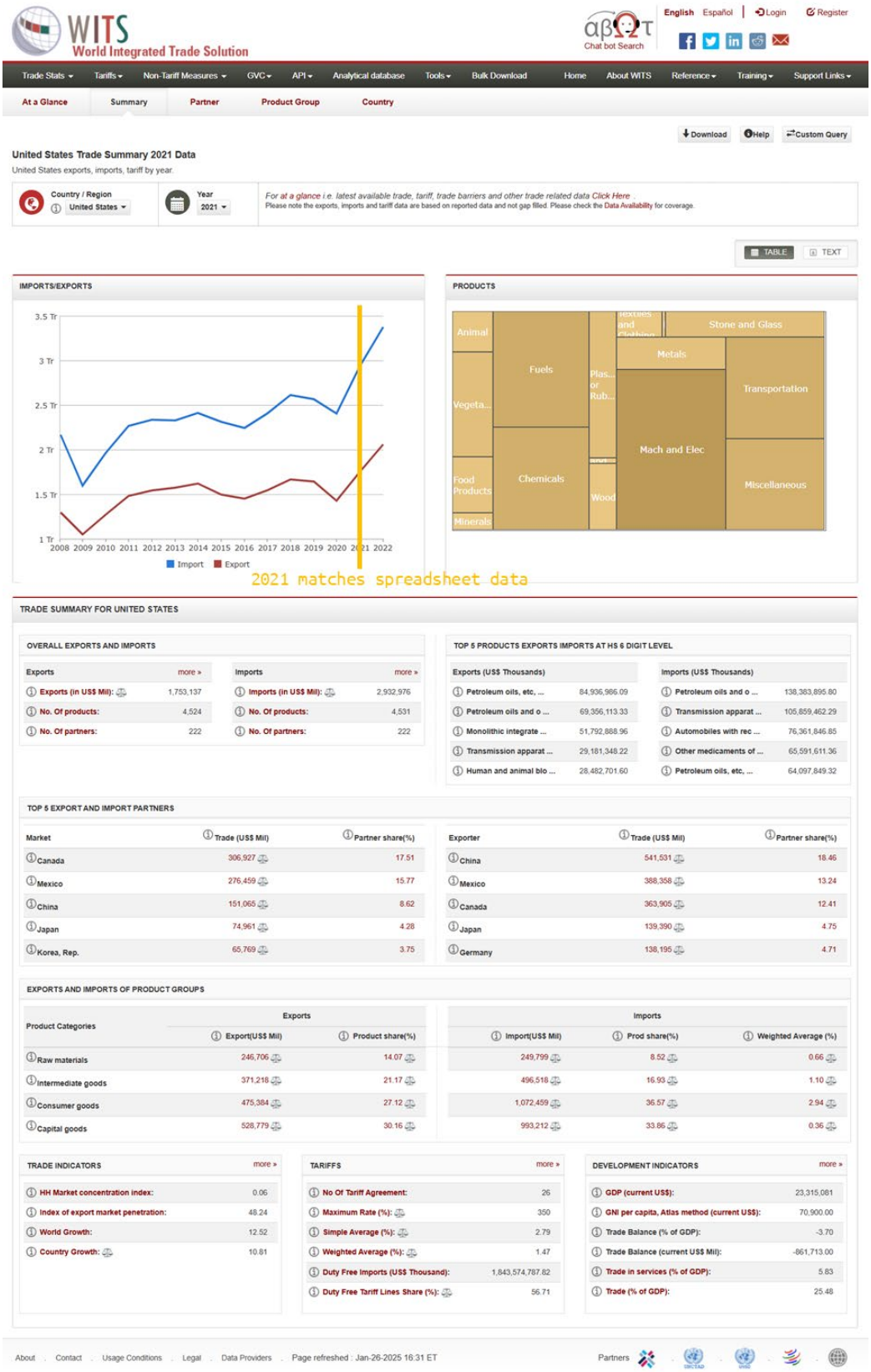
Figure 10



Link to dataset

https://wits.worldbank.org/Default.aspx?lang=en

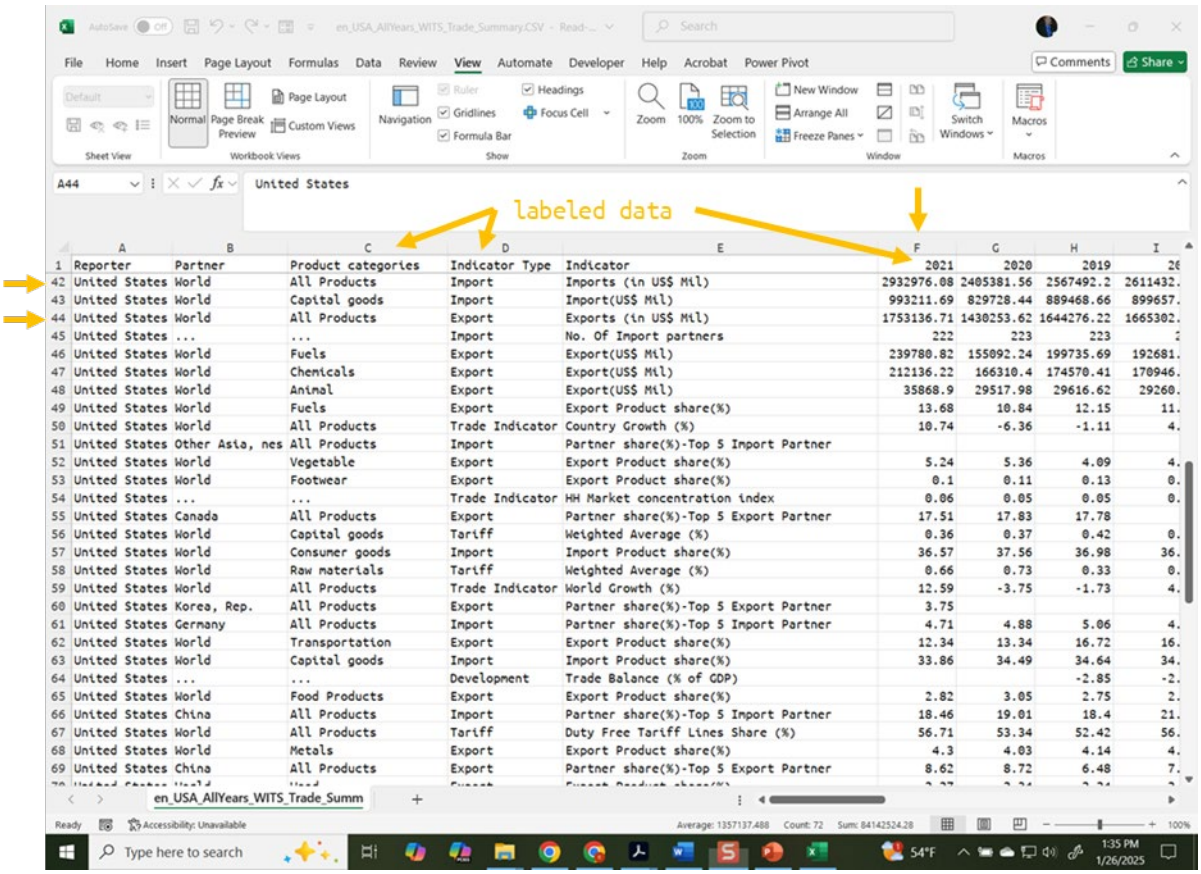
Figure 11


WITS
World Integrated Trade Solution
Chat bot Search
English Español Login Register
Trade Stats Tariffs Non-Tariff Measures GVC API Analytical database Tools Bulk Download Home About WITS Reference Training Support Links
At a Glance Summary Partner Product Group Country
Download Help Custom Query
United States Trade Summary 2021 Data
United States exports, imports, tariff by year.
Country / Region United States
Year 2021
For at a glance i.e. latest available trade, tariff, trade barriers and other trade related data Click Here
Please note the exports, imports and tariff data are based on reported data and not gap filled. Please check the Data Availability for coverage.
TABLE TEXT
IMPORTS/EXPORTS
3.5 Tr
3 Tr
2.5 Tr
2 Tr
1.5 Tr
1 Tr
2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022
Import Export
PRODUCTS
Animal
Fuels
Stone and Glass
Metals
Vegeta...
Plas... or Rub...
Transportation
Mach and Elec
Food Products
Chemicals
Miscellaneous
Wood
Minerals
2021 matches spreadsheet data
TRADE SUMMARY FOR UNITED STATES
OVERALL EXPORTS AND IMPORTS
Exports more »
Exports (in US$ Mil): 1,753,137
No. Of products: 4,524
No. Of partners: 222
Imports more »
Imports (in US$ Mil): 2,932,976
No. Of products: 4,531
No. Of partners: 222
TOP 5 PRODUCTS EXPORTS IMPORTS AT HS 6 DIGIT LEVEL
Exports (US$ Thousands)
Petroleum oils, etc, ... 84,936,986.09
Petroleum oils and o ... 69,356,113.33
Monolithic integrate ... 51,792,888.96
Transmission apparat ... 29,181,348.22
Human and animal blo ... 28,482,701.60
Imports (US$ Thousands)
Petroleum oils and o ... 138,383,895.80
Transmission apparat ... 105,859,462.29
Automobiles with rec ... 76,361,846.85
Other medicaments of ... 65,591,611.36
Petroleum oils, etc, ... 64,097,849.32
TOP 5 EXPORT AND IMPORT PARTNERS
Market | Trade (US$ Mil) | Partner share(%)
Canada 306,927 17.51
Mexico 276,459 15.77
China 151,065 8.62
Japan 74,961 4.28
Korea, Rep. 65,769 3.75
Exporter | Trade (US$ Mil) | Partner share(%)
China 541,531 18.46
Mexico 388,358 13.24
Canada 363,905 12.41
Japan 139,390 4.75
Germany 138,195 4.71
EXPORTS AND IMPORTS OF PRODUCT GROUPS
Product Categories
Exports: Export(US$ Mil) | Product share(%)
Imports: Import(US$ Mil) | Prod share(%) | Weighted Average (%)
Raw materials 246,706 14.07 249,799 8.52 0.66
Intermediate goods 371,218 21.17 496,518 16.93 1.10
Consumer goods 475,384 27.12 1,072,459 36.57 2.94
Capital goods 528,779 30.16 993,212 33.86 0.36
TRADE INDICATORS more »
HH Market concentration index: 0.06
Index of export market penetration: 48.24
World Growth: 12.52
Country Growth: 10.81
TARIFFS more »
No Of Tariff Agreement: 26
Maximum Rate (%): 350
Simple Average (%): 2.79
Weighted Average (%): 1.47
Duty Free Imports (US$ Thousand): 1,843,574,787.82
Duty Free Tariff Lines Share (%): 56.71
DEVELOPMENT INDICATORS more »
GDP (current US$): 23,315,081
GNI per capita, Atlas method (current US$): 70,900.00
Trade Balance (% of GDP): -3.70
Trade Balance (current US$ Mil): -861,713.00
Trade in services (% of GDP): 5.83
Trade (% of GDP): 25.48
About Contact Usage Conditions Legal Data Providers Page refreshed : Jan-26-2025 16:31 ET
Partners

Link to dataset

https://wits.worldbank.org/CountryProfile/en/Country/USA/Year/2022/Summary

Figure 12



	A	B	C	D	E	F	G	H	I
1	Reporter	Partner	Product categories	Indicator Type	Indicator	2021	2020	2019	2018
42	United States	World	All Products	Import	Imports (in US$ Mil)	2932976.08	2405381.56	2567492.2	2611432.
43	United States	World	Capital goods	Import	Import(US$ Mil)	993211.69	829728.44	889468.66	899657.
44	United States	World	All Products	Export	Exports (in US$ Mil)	1753136.71	1430253.62	1644276.22	1665302.
45	United States	...	...	Import	No. Of Import partners	222	223	223	2
46	United States	World	Fuels	Export	Export(US$ Mil)	239780.82	155092.24	199735.69	192681.
47	United States	World	Chemicals	Export	Export(US$ Mil)	212136.22	166310.4	174570.41	170946.
48	United States	World	Animal	Export	Export(US$ Mil)	35868.9	29517.98	29616.62	29260.
49	United States	World	Fuels	Export	Export Product share(%)	13.68	10.84	12.15	11.
50	United States	World	All Products	Trade Indicator	Country Growth (%)	10.74	-6.36	-1.11	4.
51	United States	Other Asia, nes	All Products	Import	Partner share(%)-Top 5 Import Partner				
52	United States	World	Vegetable	Export	Export Product share(%)	5.24	5.36	4.09	4.
53	United States	World	Footwear	Export	Export Product share(%)	0.1	0.11	0.13	0.
54	United States	...	...	Trade Indicator	HH Market concentration index	0.06	0.05	0.05	0.
55	United States	Canada	All Products	Export	Partner share(%)-Top 5 Export Partner	17.51	17.83	17.78	
56	United States	World	Capital goods	Tariff	Weighted Average (%)	0.36	0.37	0.42	0.
57	United States	World	Consumer goods	Import	Import Product share(%)	36.57	37.56	36.98	36.
58	United States	World	Raw materials	Tariff	Weighted Average (%)	0.66	0.73	0.33	0.
59	United States	World	All Products	Trade Indicator	World Growth (%)	12.59	-3.75	-1.73	4.
60	United States	Korea, Rep.	All Products	Export	Partner share(%)-Top 5 Export Partner	3.75			
61	United States	Germany	All Products	Import	Partner share(%)-Top 5 Import Partner	4.71	4.88	5.06	4.
62	United States	World	Transportation	Export	Export Product share(%)	12.34	13.34	16.72	16.
63	United States	World	Capital goods	Import	Import Product share(%)	33.86	34.49	34.64	34.
64	United States	...	...	Development	Trade Balance (% of GDP)			-2.85	-2.
65	United States	World	Food Products	Export	Export Product share(%)	2.82	3.05	2.75	2.
66	United States	China	All Products	Import	Partner share(%)-Top 5 Import Partner	18.46	19.01	18.4	21.
67	United States	World	All Products	Tariff	Duty Free Tariff Lines Share (%)	56.71	53.34	52.42	56.
68	United States	World	Metals	Export	Export Product share(%)	4.3	4.03	4.14	4.
69	United States	China	All Products	Export	Partner share(%)-Top 5 Export Partner	8.62	8.72	6.48	7.

Row lines 42 and 44 for all products match the above charts for trade debt US vs World on imports and exports for 2021. The data for the charts above can also be downloaded as csv files.

Discussion Qualitative, Quantitative, Mixed, Labeled and more

As of this morning the US enacted an emergency tariff on the Country of Columbia because they refused to accept over one hundred plane loads of returned Columbian refugees. Tariffs are a focus on this website and will be able to see the exact amount of tariffs imposed onto different countries.

The website data has both qualitative and quantitative values. I can also download the data as an excel csv spreadsheet or text file. All the data is labeled in the csv spreadsheet. In the dataset, there are over

20 files depending on the country. As you can see there can be many categories. I have just started to understand some of the terms used.

Data Gathering Using an API

A Python program was written called census.py that uses the API made by Census Bureau website and is included in the submission.

Example. The United States Census Bureau (census.gov)

Description

The Census Bureau is dedicated to providing current facts and figures about America's people, places, and economy.

In the API examples from their website, There are separate API's to do different data gathering tasks. It gathers export data for many countries around the world. What is gathered in this API is export data using these parameters.

"CTY_CODE",

"CTY_NAME",

"ALL_VAL_MO",

"ALL_VAL_YR",

"time"

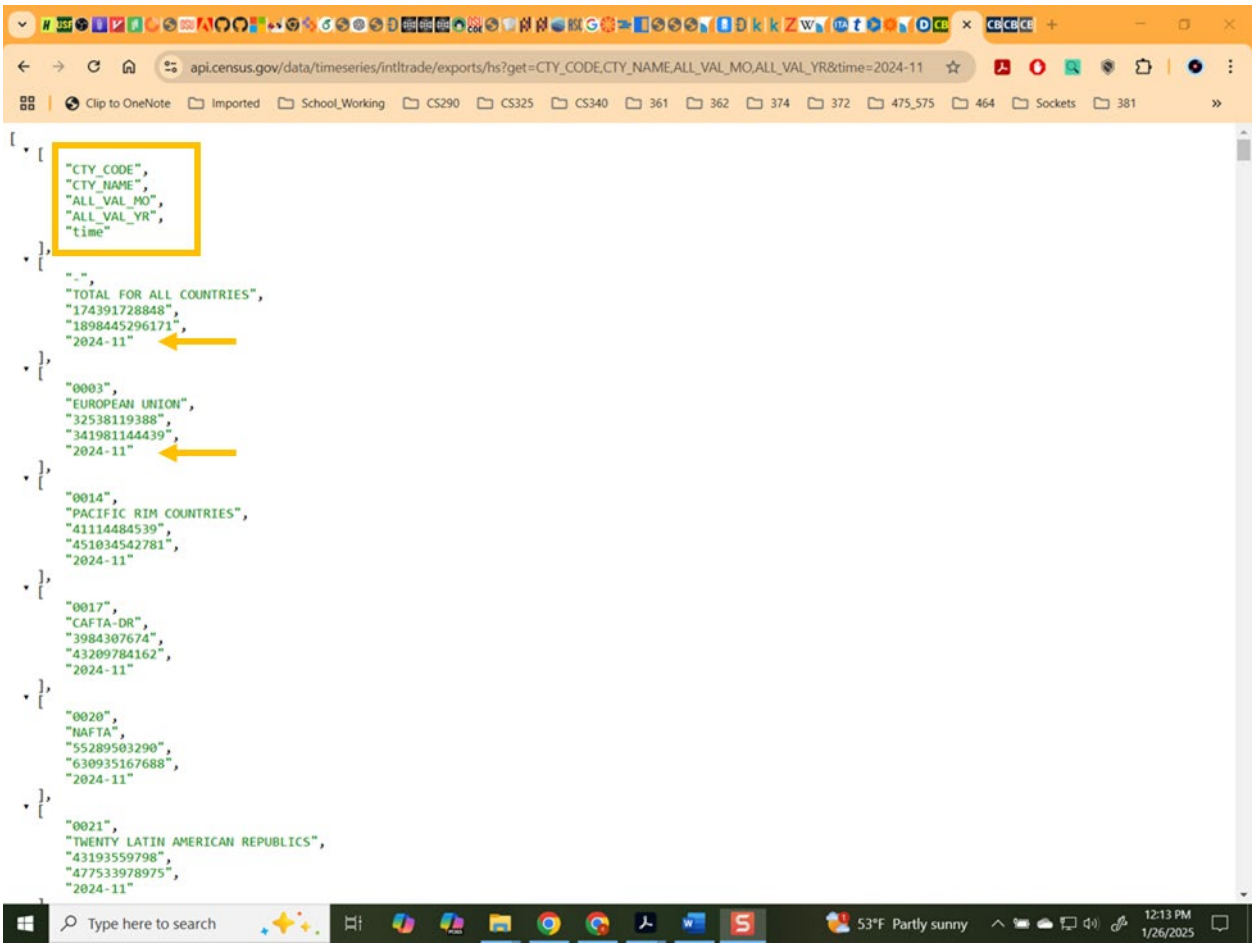
Link to the API

Census Bureau example APIs are located at

https://www.census.gov/data/developers/data-sets/international-trade.html

API query

Figure 13



An actual query using the following API is demonstrated to work in a web browser. The data is returned as JSON.

https://api.census.gov/data/timeseries/intltrade/exports/hs?

get=CTY_CODE,

CTY_NAME,

ALL_VAL_MO,

ALL_VAL_YR&

time=2024-11

Visualize with screen images

Figure 14

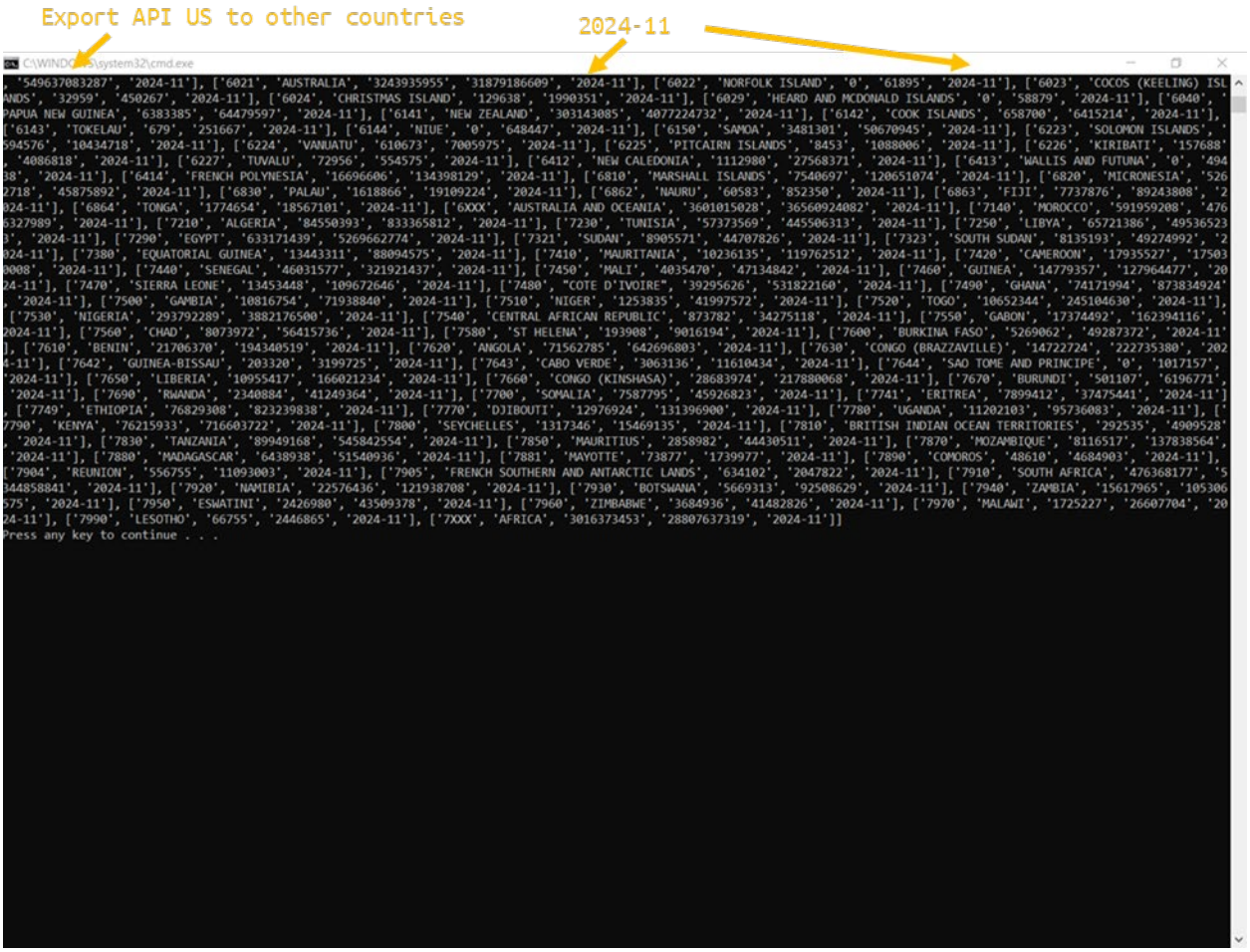

Export API US to other countries
2024-11
C:\WINDOWS\system32\cmd.exe
, '549637083287', '2024-11'], ['6021', 'AUSTRALIA', '3243935955', '31879186609', '2024-11'], ['6022', 'NORFOLK ISLAND', '0', '61895', '2024-11'], ['6023', 'COCOS (KEELING) ISL
ANDS', '32959', '450267', '2024-11'], ['6024', 'CHRISTMAS ISLAND', '129638', '1990351', '2024-11'], ['6029', 'HEARD AND MCDONALD ISLANDS', '0', '58879', '2024-11'], ['6040', '
PAPUA NEW GUINEA', '6383385', '64479597', '2024-11'], ['6141', 'NEW ZEALAND', '303143085', '4077224732', '2024-11'], ['6142', 'COOK ISLANDS', '658700', '6415214', '2024-11'],
['6143', 'TOKELAU', '679', '251667', '2024-11'], ['6144', 'NIUE', '0', '648447', '2024-11'], ['6150', 'SAMOA', '3481301', '50670945', '2024-11'], ['6223', 'SOLOMON ISLANDS', '
594576', '10434718', '2024-11'], ['6224', 'VANUATU', '610673', '7005975', '2024-11'], ['6225', 'PITCAIRN ISLANDS', '8453', '1088006', '2024-11'], ['6226', 'KIRIBATI', '157688'
, '4086818', '2024-11'], ['6227', 'TUVALU', '72956', '554575', '2024-11'], ['6412', 'NEW CALEDONIA', '1112980', '27568371', '2024-11'], ['6413', 'WALLIS AND FUTUNA', '0', '494
38', '2024-11'], ['6414', 'FRENCH POLYNESIA', '16696606', '134398129', '2024-11'], ['6810', 'MARSHALL ISLANDS', '7540697', '12065074', '2024-11'], ['6820', 'MICRONESIA', '526
2718', '45875892', '2024-11'], ['6830', 'PALAU', '1618866', '19109224', '2024-11'], ['6862', 'NAURU', '60583', '852350', '2024-11'], ['6863', 'FIJI', '7737876', '89243808', '2
024-11'], ['6864', 'TONGA', '1774654', '18567101', '2024-11'], ['6XXX', 'AUSTRALIA AND OCEANIA', '3601015028', '36560924082', '2024-11'], ['7140', 'MOROCCO', '591959208', '476
6327989', '2024-11'], ['7210', 'ALGERIA', '84550393', '833365812', '2024-11'], ['7230', 'TUNISIA', '57373569', '445506313', '2024-11'], ['7250', 'LIBYA', '65721386', '49536523
3', '2024-11'], ['7290', 'EGYPT', '633171439', '5269662774', '2024-11'], ['7321', 'SUDAN', '8905571', '44707826', '2024-11'], ['7323', 'SOUTH SUDAN', '8135193', '49274992', '2
024-11'], ['7380', 'EQUATORIAL GUINEA', '13443311', '88094575', '2024-11'], ['7410', 'MAURITANIA', '10236135', '119762512', '2024-11'], ['7420', 'CAMEROON', '17935527', '17503
9008', '2024-11'], ['7440', 'SENEGAL', '46031577', '321921437', '2024-11'], ['7450', 'MALI', '4035470', '47134842', '2024-11'], ['7460', 'GUINEA', '14779357', '127964477', '20
24-11'], ['7470', 'SIERRA LEONE', '13453448', '109672646', '2024-11'], ['7480', 'COTE D'IVOIRE', '39295626', '531822160', '2024-11'], ['7490', 'GHANA', '74171994', '873834924'
, '2024-11'], ['7500', 'GAMBIA', '10816754', '71938840', '2024-11'], ['7510', 'NIGER', '1253835', '41997572', '2024-11'], ['7520', 'TOGO', '10652344', '245104630', '2024-11'],
['7530', 'NIGERIA', '293792289', '3882176500', '2024-11'], ['7540', 'CENTRAL AFRICAN REPUBLIC', '873782', '34275118', '2024-11'], ['7550', 'GABON', '17374492', '162394116', '
2024-11'], ['7560', 'CHAD', '8073972', '56415736', '2024-11'], ['7580', 'ST HELENA', '193908', '9016194', '2024-11'], ['7600', 'BURKINA FASO', '5269062', '49287372', '2024-11'
], ['7610', 'BENIN', '21706370', '194340519', '2024-11'], ['7620', 'ANGOLA', '71562785', '642696803', '2024-11'], ['7630', 'CONGO (BRAZZAVILLE)', '14722724', '222735380', '202
4-11'], ['7642', 'GUINEA-BISSAU', '203320', '3199725', '2024-11'], ['7643', 'CABO VERDE', '3063136', '11610434', '2024-11'], ['7644', 'SAO TOME AND PRINCIPE', '0', '1017157',
'2024-11'], ['7650', 'LIBERIA', '10955417', '166021234', '2024-11'], ['7660', 'CONGO (KINSHASA)', '28683974', '217880068', '2024-11'], ['7670', 'BURUNDI', '501107', '6196771',
'2024-11'], ['7690', 'RWANDA', '2340884', '41249364', '2024-11'], ['7700', 'SOMALIA', '7587795', '45926823', '2024-11'], ['7741', 'ERITREA', '7899412', '37475441', '2024-11']
, ['7749', 'ETHIOPIA', '76829308', '823239838', '2024-11'], ['7770', 'DJIBOUTI', '12976924', '131396900', '2024-11'], ['7780', 'UGANDA', '11202103', '95736083', '2024-11'], ['
7790', 'KENYA', '76215933', '716603722', '2024-11'], ['7800', 'SEYCHELLES', '1317346', '15469135', '2024-11'], ['7810', 'BRITISH INDIAN OCEAN TERRITORIES', '292535', '4909528'
, '2024-11'], ['7830', 'TANZANIA', '89949168', '545842554', '2024-11'], ['7850', 'MAURITIUS', '2858982', '44430511', '2024-11'], ['7870', 'MOZAMBIQUE', '8116517', '13738564',
'2024-11'], ['7880', 'MADAGASCAR', '6438938', '51540936', '2024-11'], ['7881', 'MAYOTTE', '73877', '1739977', '2024-11'], ['7890', 'COMOROS', '48610', '4684903', '2024-11'],
['7904', 'REUNION', '556755', '11093003', '2024-11'], ['7905', 'FRENCH SOUTHERN AND ANTARCTIC LANDS', '634102', '2047822', '2024-11'], ['7910', 'SOUTH AFRICA', '476368177', '5
44858841', '2024-11'], ['7920', 'NAMIBIA', '22576436', '121938708', '2024-11'], ['7930', 'BOTSWANA', '5669313', '92508629', '2024-11'], ['7940', 'ZAMBIA', '15617965', '105306
575', '2024-11'], ['7950', 'ESWATINI', '2426980', '43509378', '2024-11'], ['7960', 'ZIMBABWE', '3684936', '41482826', '2024-11'], ['7970', 'MALAWI', '1725227', '26607704', '20
24-11'], ['7990', 'LESOTHO', '66755', '2446865', '2024-11'], ['7XXX', 'AFRICA', '3016373453', '28807637319', '2024-11']]
Press any key to continue . . .

Figure 15

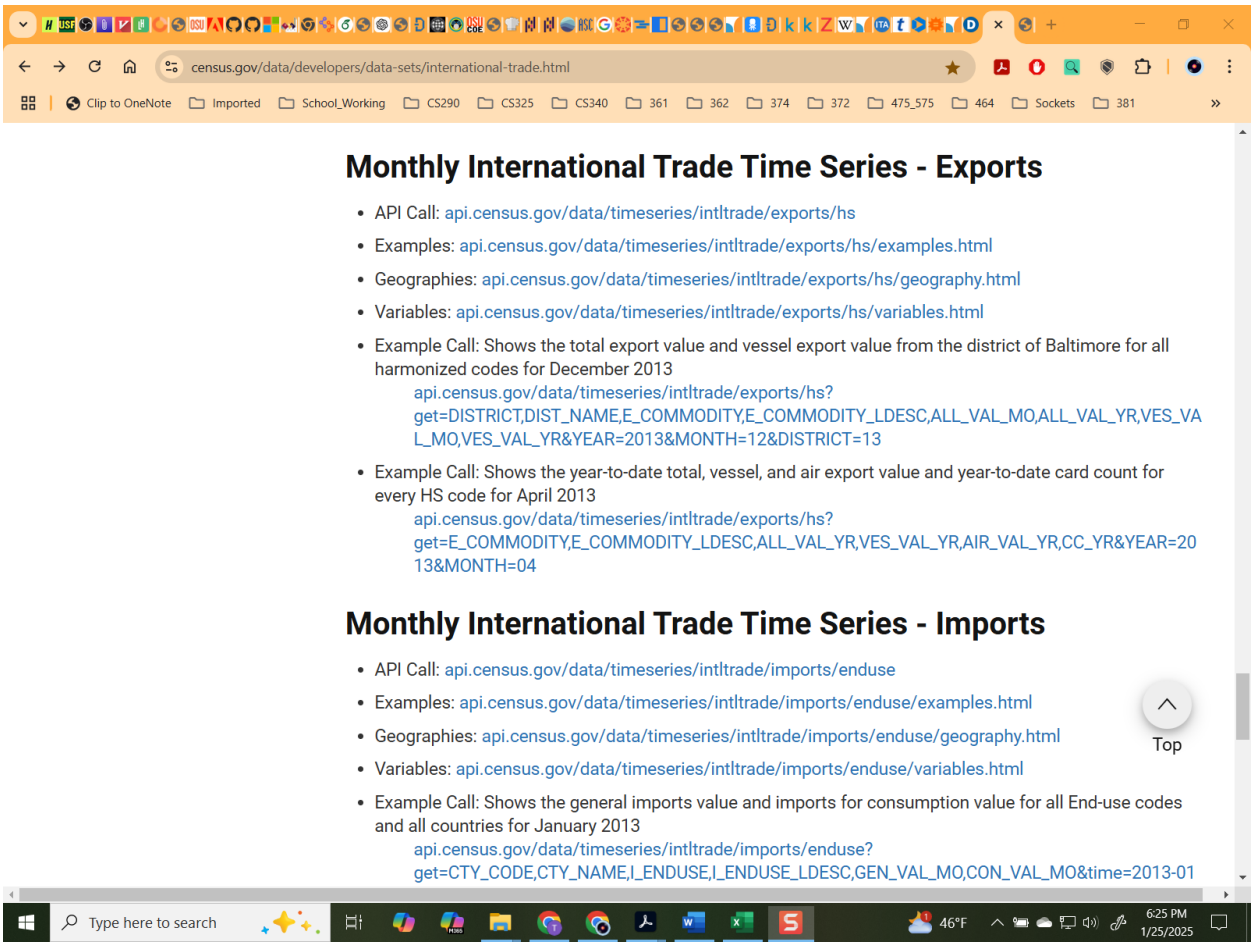
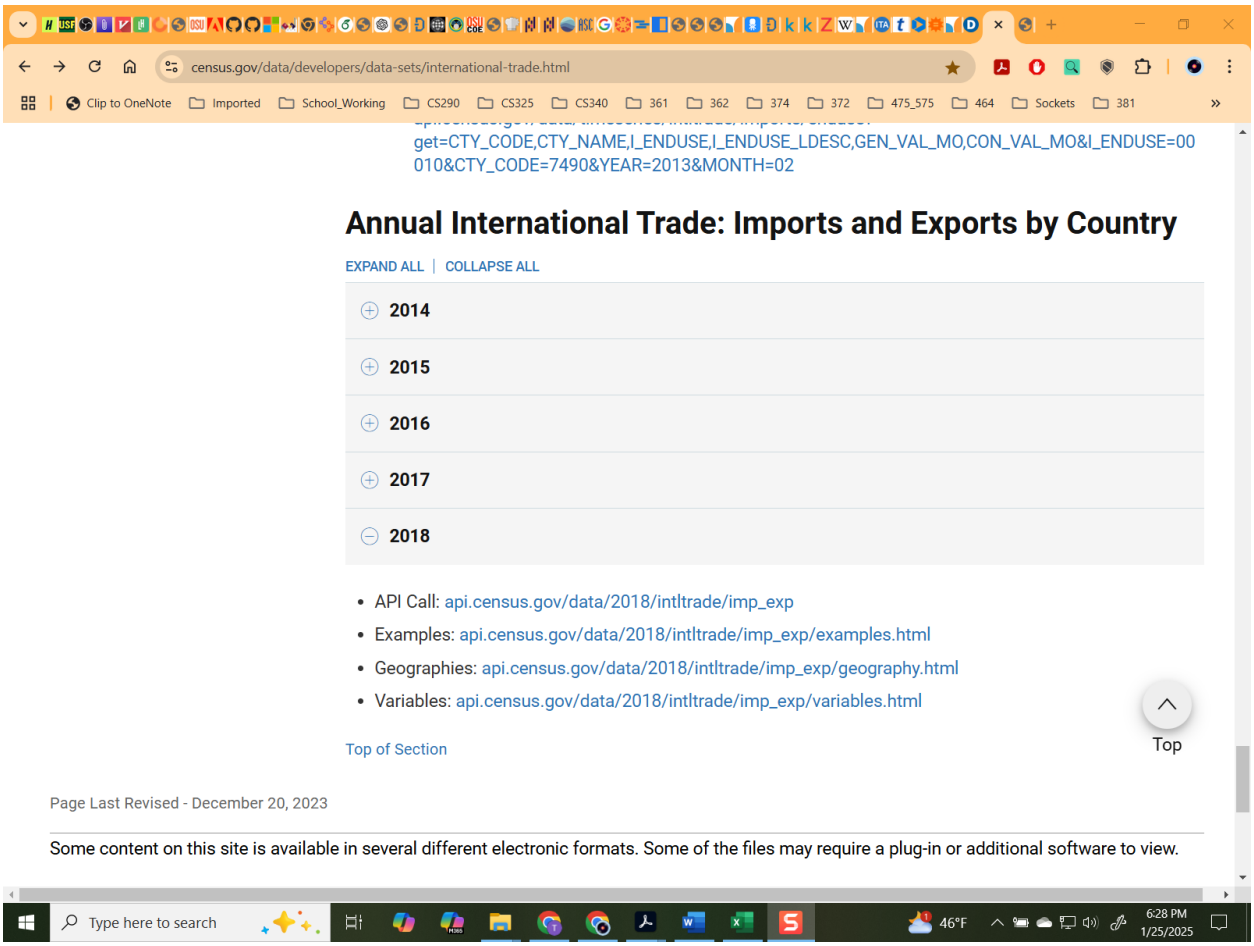

census.gov/data/developers/data-sets/international-trade.html
Monthly International Trade Time Series - Exports
API Call: api.census.gov/data/timeseries/intltrade/exports/hs
Examples: api.census.gov/data/timeseries/intltrade/exports/hs/examples.html
Geographies: api.census.gov/data/timeseries/intltrade/exports/hs/geography.html
Variables: api.census.gov/data/timeseries/intltrade/exports/hs/variables.html
Example Call: Shows the total export value and vessel export value from the district of Baltimore for all harmonized codes for December 2013
api.census.gov/data/timeseries/intltrade/exports/hs?get=DISTRICT,DIST_NAME,E_COMMODITY,E_COMMODITY_LDESC,ALL_VAL_MO,ALL_VAL_YR,VES_VAL_MO,VES_VAL_YR&YEAR=2013&MONTH=12&DISTRICT=13
Example Call: Shows the year-to-date total, vessel, and air export value and year-to-date card count for every HS code for April 2013
api.census.gov/data/timeseries/intltrade/exports/hs?get=E_COMMODITY,E_COMMODITY_LDESC,ALL_VAL_YR,VES_VAL_YR,AIR_VAL_YR,CC_YR&YEAR=2013&MONTH=04
Monthly International Trade Time Series - Imports
API Call: api.census.gov/data/timeseries/intltrade/imports/enduse
Examples: api.census.gov/data/timeseries/intltrade/imports/enduse/examples.html
Geographies: api.census.gov/data/timeseries/intltrade/imports/enduse/geography.html
Variables: api.census.gov/data/timeseries/intltrade/imports/enduse/variables.html
Example Call: Shows the general imports value and imports for consumption value for all End-use codes and all countries for January 2013
api.census.gov/data/timeseries/intltrade/imports/enduse?get=CTY_CODE,CTY_NAME,I_ENDUSE,I_ENDUSE_LDESC,GEN_VAL_MO,CON_VAL_MO&time=2013-01
Top
Type here to search
46°F
6:25 PM
1/25/2025

Figure 16



Discussion Qualitative, Quantitative, Mixed, Labeled and more

The webpage screenshot is using the export API from US to other countries for 2024-11. The data is in JSON format which is just a data format. Labeled vs. unlabeled depends on whether each record in the data includes a known target or outcome variable. There is no category label to go with these values but the labels are defined at the top of the JSON file. Since the label is not included with each value then I consider it unlabeled. The same data structure format is used for 2024-11 and for different countries not shown. It has both qualitative and quantitative data. The city name is qualitative but there is a matching quantitative city code that is included.

Figure 15 shows what all the different APIs can do.

Data Gathering using Web Scraping

Example. World Trade Organization WTO (wto.org)

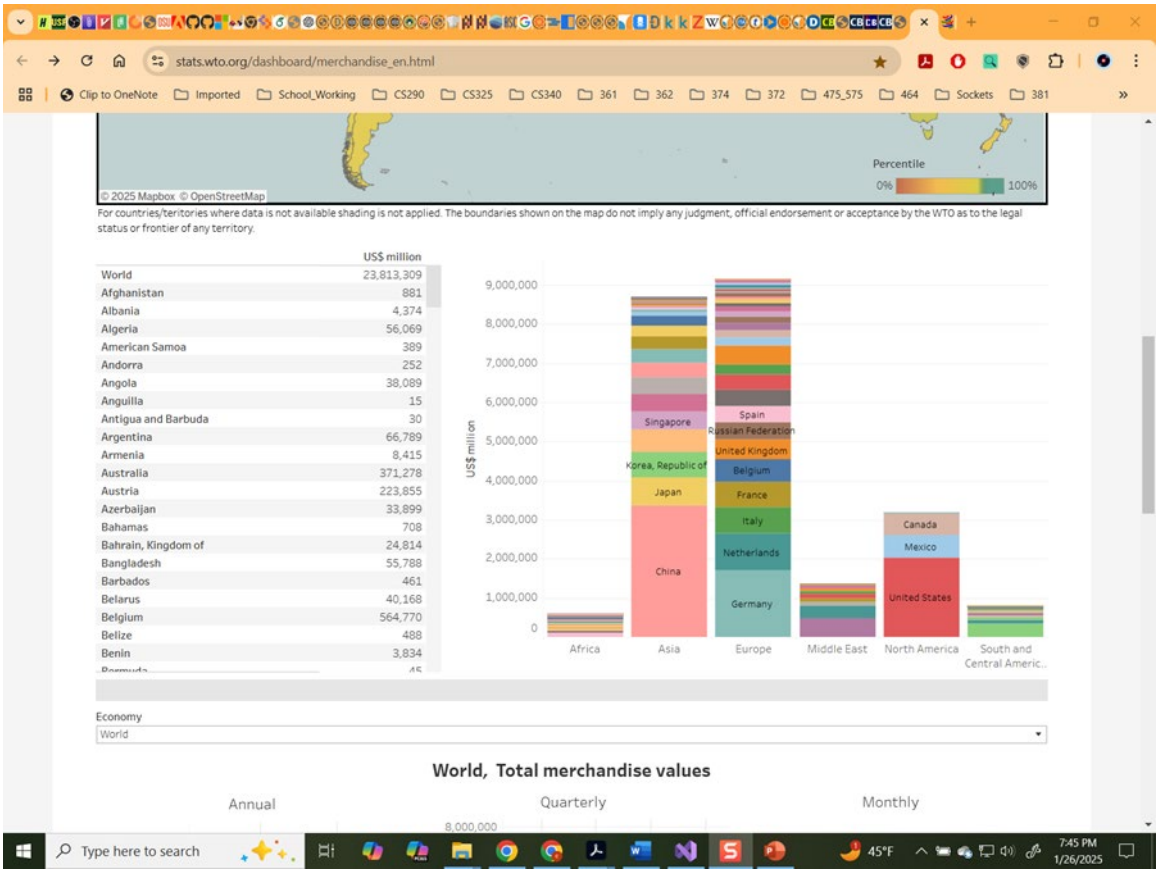
Description

The World Trade Organization — the WTO — is the international organization whose primary purpose is to open trade for the benefit of all.

The WTO Stats Dashboard shows the total merchandise exports for countries around the world and by year.

Visualize with screen images

Figure 17

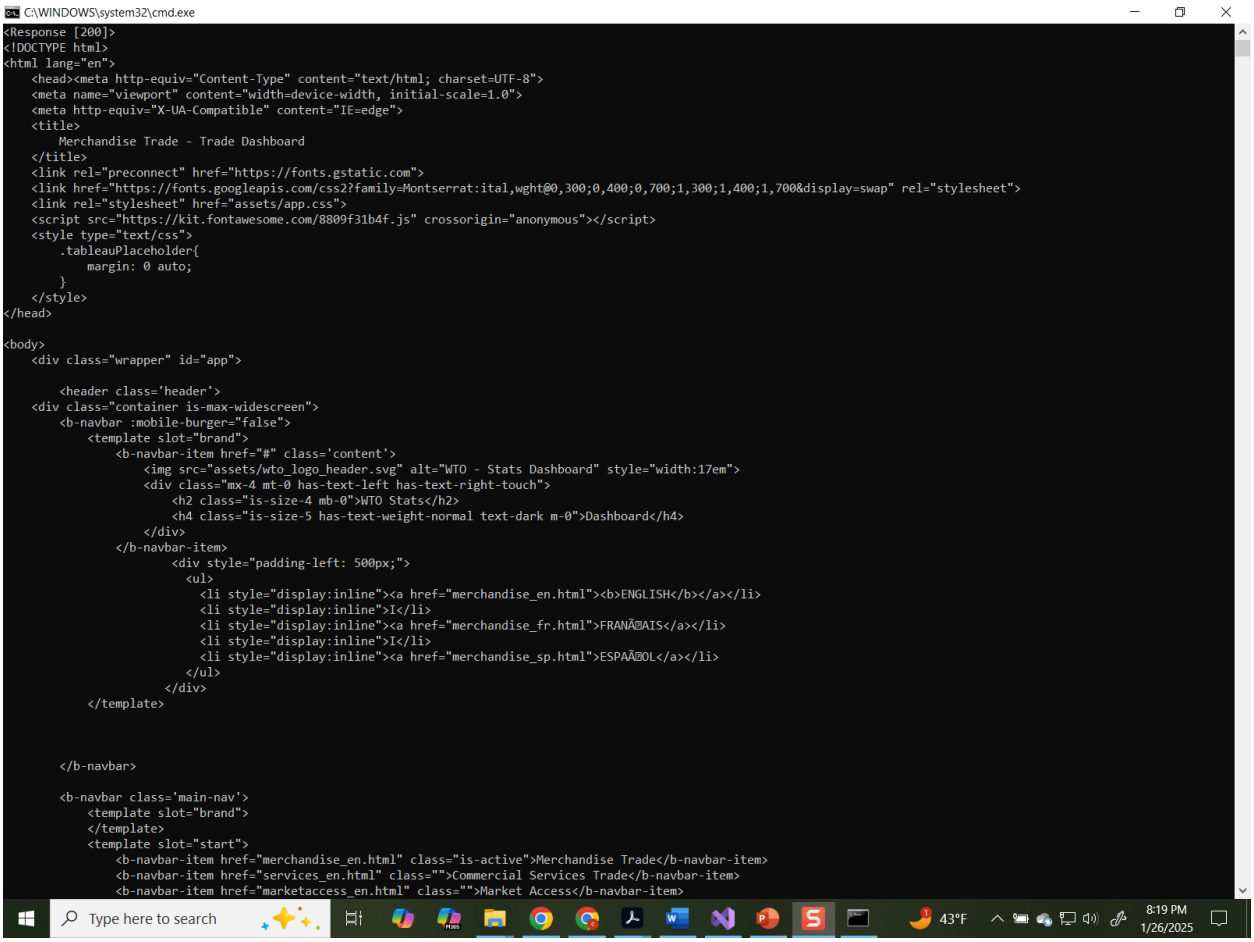


Link to webpage

https://stats.wto.org/dashboard/merchandise_en.html

Results of webpage scrape

Figure 18



Discussion Qualitative, Quantitative, Mixed, Labeled and more

A python file named wto_scrape.py was created to scrape the webpage and is included in the submission. The web scraping will extract the targeted webpage source code but does not get the value from running the webpage. In order to run the webpage, programs like Selenium and Beautiful Soup need to be used also.

The WTO Stats Dashboard, Figure 17, shows the total merchandise exports for countries around the world and by year. The web scraping does not get the actual dollar amount of merchandise exports yet. It will need more data processing, but the web scraping did occur in Figure 18.

Data Cleaning

Data cleaning for 3 datasets which are all direct data download.

Dataset 1 World Export & Import Dataset (1989 - 2023)

Dataset is from Kaggle

AHS (Applied Tariff Rates) and MFN (Most Favored Nation) tariff rates are two different types of tariff rates used in international trade:

1. AHS (Applied Tariff Rates):

These are the actual tariff rates that a country applies to imports from other countries.

AHS rates can vary depending on trade agreements, preferential trade arrangements, or specific trade policies.

They reflect the real-world tariffs that importers face when bringing goods into a country.

2. MFN (Most Favored Nation) Tariff Rates:

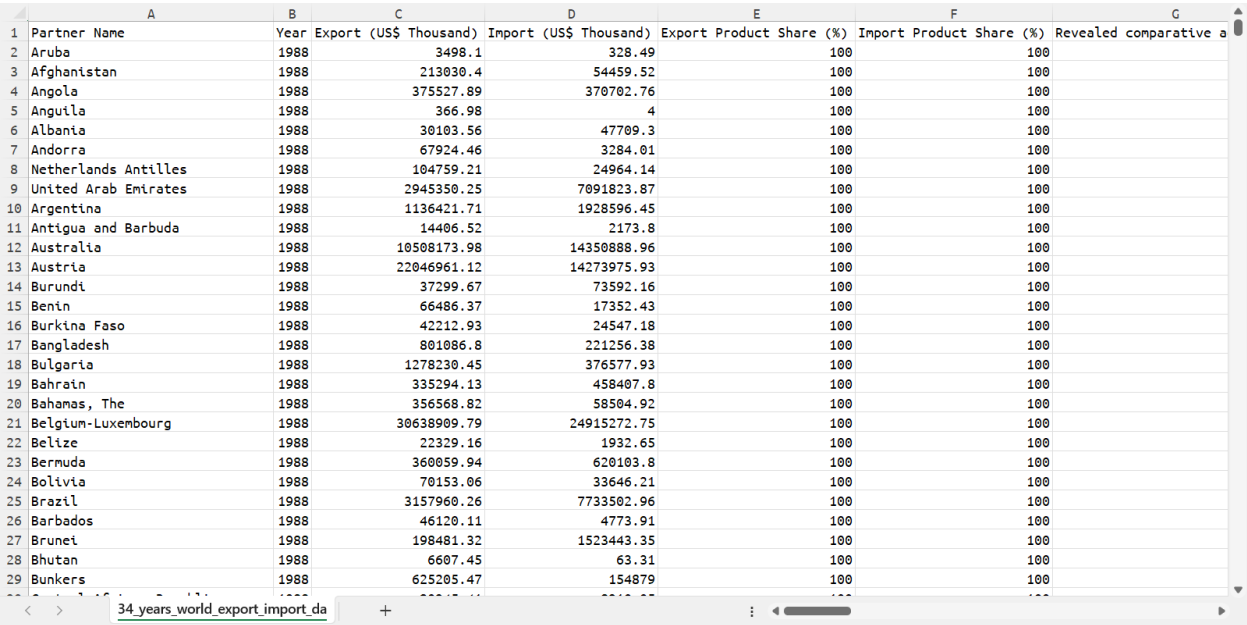
These are the standard tariff rates that a country applies to imports from all other World Trade Organization (WTO) member countries, unless a preferential trade agreement is in place.

The MFN principle ensures that WTO members do not discriminate between their trading partners. If a country grants a lower tariff rate to one WTO member, it must extend the same rate to all other WTO members.

MFN rates are often used as a baseline for trade negotiations and are generally lower than the maximum tariff rates a country might apply.

(a) Write Python code to assure that your datasets are in record format so that they are structured as rows and columns, where each column has a variable name.

Before



	A	B	C	D	E	F	G
1	Partner Name	Year	Export (US$ Thousand)	Import (US$ Thousand)	Export Product Share (%)	Import Product Share (%)	Revealed comparative a
2	Aruba	1988	3498.1	328.49	100	100	
3	Afghanistan	1988	213030.4	54459.52	100	100	
4	Angola	1988	375527.89	370702.76	100	100	
5	Anguila	1988	366.98	4	100	100	
6	Albania	1988	30103.56	47709.3	100	100	
7	Andorra	1988	67924.46	3284.01	100	100	
8	Netherlands Antilles	1988	104759.21	24964.14	100	100	
9	United Arab Emirates	1988	2945350.25	7091823.87	100	100	
10	Argentina	1988	1136421.71	1928596.45	100	100	
11	Antigua and Barbuda	1988	14406.52	2173.8	100	100	
12	Australia	1988	10508173.98	14350888.96	100	100	
13	Austria	1988	22046961.12	14273975.93	100	100	
14	Burundi	1988	37299.67	73592.16	100	100	
15	Benin	1988	66486.37	17352.43	100	100	
16	Burkina Faso	1988	42212.93	24547.18	100	100	
17	Bangladesh	1988	801086.8	221256.38	100	100	
18	Bulgaria	1988	1278230.45	376577.93	100	100	
19	Bahrain	1988	335294.13	458407.8	100	100	
20	Bahamas, The	1988	356568.82	58504.92	100	100	
21	Belgium-Luxembourg	1988	30638909.79	24915272.75	100	100	
22	Belize	1988	22329.16	1932.65	100	100	
23	Bermuda	1988	360059.94	620103.8	100	100	
24	Bolivia	1988	70153.06	33646.21	100	100	
25	Brazil	1988	3157960.26	7733502.96	100	100	
26	Barbados	1988	46120.11	4773.91	100	100	
27	Brunei	1988	198481.32	1523443.35	100	100	
28	Bhutan	1988	6607.45	63.31	100	100	
29	Bunkers	1988	625205.47	154879	100	100	

34_years_world_export_import_da

Big csv file.

After

```
Data loaded successfully
                   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

      AHS Weighted Average (%)  AHS Total Tariff Lines  \
0                         2.92                   155.0
1                         1.83                   548.0
2                         3.89                   633.0
3                         1.09                    33.0
4                         2.38                   744.0
```

There are 8096 rows.

(b) Write Python code to check and print the data types of the variables in your dataset.

Before

```
Partner Name                                object
Year                                         int64
Export (US$ Thousand)                      float64
Import (US$ Thousand)                      float64
Export Product Share (%)                   float64
Import Product Share (%)                     int64
Revealed comparative advantage             float64
World Growth (%)                           float64
Country Growth (%)                         float64
AHS Simple Average (%)                     float64
AHS Weighted Average (%)                   float64
AHS Total Tariff Lines                     float64
AHS Dutiable Tariff Lines Share (%)        float64
AHS Duty Free Tariff Lines Share (%)       float64
AHS Specific Tariff Lines Share (%)        float64
AHS AVE Tariff Lines Share (%)             float64
AHS MaxRate (%)                            float64
AHS MinRate (%)                            float64
AHS SpecificDuty Imports (US$ Thousand)    float64
AHS Dutiable Imports (US$ Thousand)        float64
AHS Duty Free Imports (US$ Thousand)       float64
MFN Simple Average (%)                     float64
MFN Weighted Average (%)                   float64
MFN Total Tariff Lines                     float64
MFN Dutiable Tariff Lines Share (%)        float64
MFN Duty Free Tariff Lines Share (%)       float64
MFN Specific Tariff Lines Share (%)        float64
MFN AVE Tariff Lines Share (%)             float64
MFN MaxRate (%)                            float64
MFN MinRate (%)                            float64
MFN SpecificDuty Imports (US$ Thousand)    float64
MFN Dutiable Imports (US$ Thousand)        float64
MFN Duty Free Imports (US$ Thousand)       float64
dtype: object
```

Write code to correct any data types. For example, if Python reads in a categorical variable as a number, you will need to update this to a category.

After

```
Partner Name                                object
Year                                         int64
Export (US$ Thousand)                      float64
Import (US$ Thousand)                      float64
Export Product Share (%)                     int32
Import Product Share (%)                     int64
Revealed comparative advantage             float64
World Growth (%)                           float64
Country Growth (%)                         float64
AHS Simple Average (%)                       int32
AHS Weighted Average (%)                   float64
AHS Total Tariff Lines                     float64
AHS Dutiable Tariff Lines Share (%)        float64
AHS Duty Free Tariff Lines Share (%)       float64
AHS Specific Tariff Lines Share (%)        float64
AHS AVE Tariff Lines Share (%)             float64
AHS MaxRate (%)                            float64
AHS MinRate (%)                            float64
AHS SpecificDuty Imports (US$ Thousand)    float64
AHS Dutiable Imports (US$ Thousand)        float64
AHS Duty Free Imports (US$ Thousand)       float64
MFN Simple Average (%)                     float64
MFN Weighted Average (%)                   float64
MFN Total Tariff Lines                     float64
MFN Dutiable Tariff Lines Share (%)        float64
MFN Duty Free Tariff Lines Share (%)       float64
MFN Specific Tariff Lines Share (%)        float64
MFN AVE Tariff Lines Share (%)             float64
MFN MaxRate (%)                            float64
MFN MinRate (%)                            float64
MFN SpecificDuty Imports (US$ Thousand)    float64
MFN Dutiable Imports (US$ Thousand)        float64
MFN Duty Free Imports (US$ Thousand)       float64
dtype: object
```

The Python code changed the datatype of Simple Average from float to integer which is one of the variables that is being analyzed. All the other datatypes do not need to be changed.

(c) Write Python code to find, count, report, and then clean any missing values.

Before

```
Data loaded successfully
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
```

The above chart shows how many Null or missing values for each column variable.

```
<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
None
Press any key to continue . . .
```

There are 8096 total rows. The Non Null column is a count of numeric values in the column.

After

```
Partner Name                                0
Year                                        0
Export (US$ Thousand)                       0
Import (US$ Thousand)                       0
Export Product Share (%)                    0
Import Product Share (%)                    0
Revealed comparative advantage           3384
World Growth (%)                         3686
Country Growth (%)                       3686
AHS Simple Average (%)                      0
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 (%)                             0
AHS MinRate (%)                             0
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
```

The Python code replaces Null or missing values with the mean or mode. All the Export and Import column variables and Simple Average Null count has been reduced to zero. I am not using the other column variables yet and missing values were not fixed.

```
<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 (%)               8096 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 (%)                 8096 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 (%)                        8096 non-null   float64
 17  AHS MinRate (%)                        8096 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
None
```

The Python code fixed the missing values by replacing them with a number. That is why the Non Null columns have more 8096 values which is the maximum rows.

(d) Write Python code to find, report, and correct any incorrect values. You can use visual methods here. For example, you can "report" incorrect values visually.

Before

	Partner Name	Year	Export (US$ Thousand)	Import (US$ Thousand)	\
183	United States	1988	1.499114e+08	8.942467e+07	
392	United States	1989	2.761864e+08	2.117782e+08	
604	United States	1990	3.096113e+08	2.489308e+08	
816	United States	1991	3.226421e+08	2.714174e+08	
1048	United States	1992	3.907638e+08	3.375038e+08	
1282	United States	1993	4.893048e+08	3.922621e+08	
1516	United States	1994	6.091563e+08	4.879011e+08	
1750	United States	1995	6.899775e+08	5.838828e+08	
1980	United States	1996	7.459441e+08	6.374394e+08	
2210	United States	1997	8.337250e+08	7.181061e+08	
2440	United States	1998	8.747538e+08	7.138633e+08	
2671	United States	1999	9.758553e+08	7.259631e+08	
2914	United States	2000	1.361770e+09	8.193953e+08	
3159	United States	2001	1.279330e+09	7.623074e+08	
3404	United States	2002	1.329803e+09	7.325223e+08	
3649	United States	2003	1.439066e+09	7.639996e+08	
3892	United States	2004	1.662595e+09	8.569998e+08	
4134	United States	2005	1.832823e+09	9.256655e+08	
4378	United States	2006	2.061697e+09	1.046340e+09	
4622	United States	2007	2.168243e+09	1.182116e+09	
4866	United States	2008	2.291350e+09	1.333130e+09	
5110	United States	2009	1.735722e+09	1.063351e+09	
5356	United States	2010	2.054287e+09	1.233995e+09	
5604	United States	2011	2.349498e+09	1.435719e+09	
5850	United States	2012	2.420653e+09	1.496188e+09	
6097	United States	2013	2.442815e+09	1.503086e+09	
6343	United States	2014	2.573329e+09	1.536625e+09	
6590	United States	2015	2.495721e+09	1.423376e+09	
6836	United States	2016	2.438626e+09	1.369816e+09	
7083	United States	2017	2.617923e+09	1.486206e+09	
7331	United States	2018	2.865909e+09	1.620239e+09	
7578	United States	2019	2.873984e+09	1.584392e+09	
7826	United States	2020	2.702693e+09	1.389319e+09	
8073	United States	2021	3.291675e+09	1.703893e+09	

Export and Import data looks good.

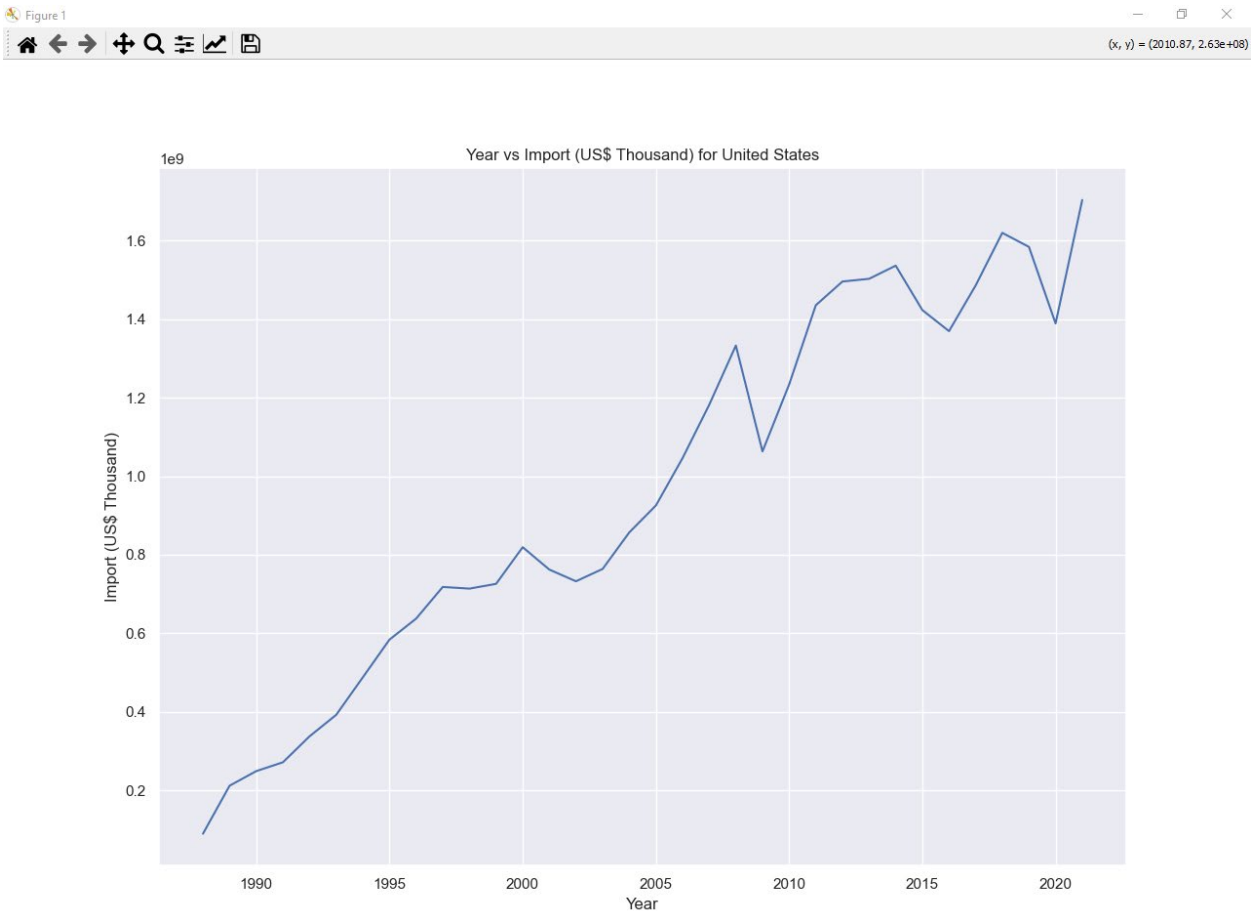


Figure 1
(x, y) = (2010.87, 2.63e+08)
Year vs Import (US$ Thousand) for United States
1e9
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
Import (US$ Thousand)
1990
1995
2000
2005
2010
2015
2020
Year

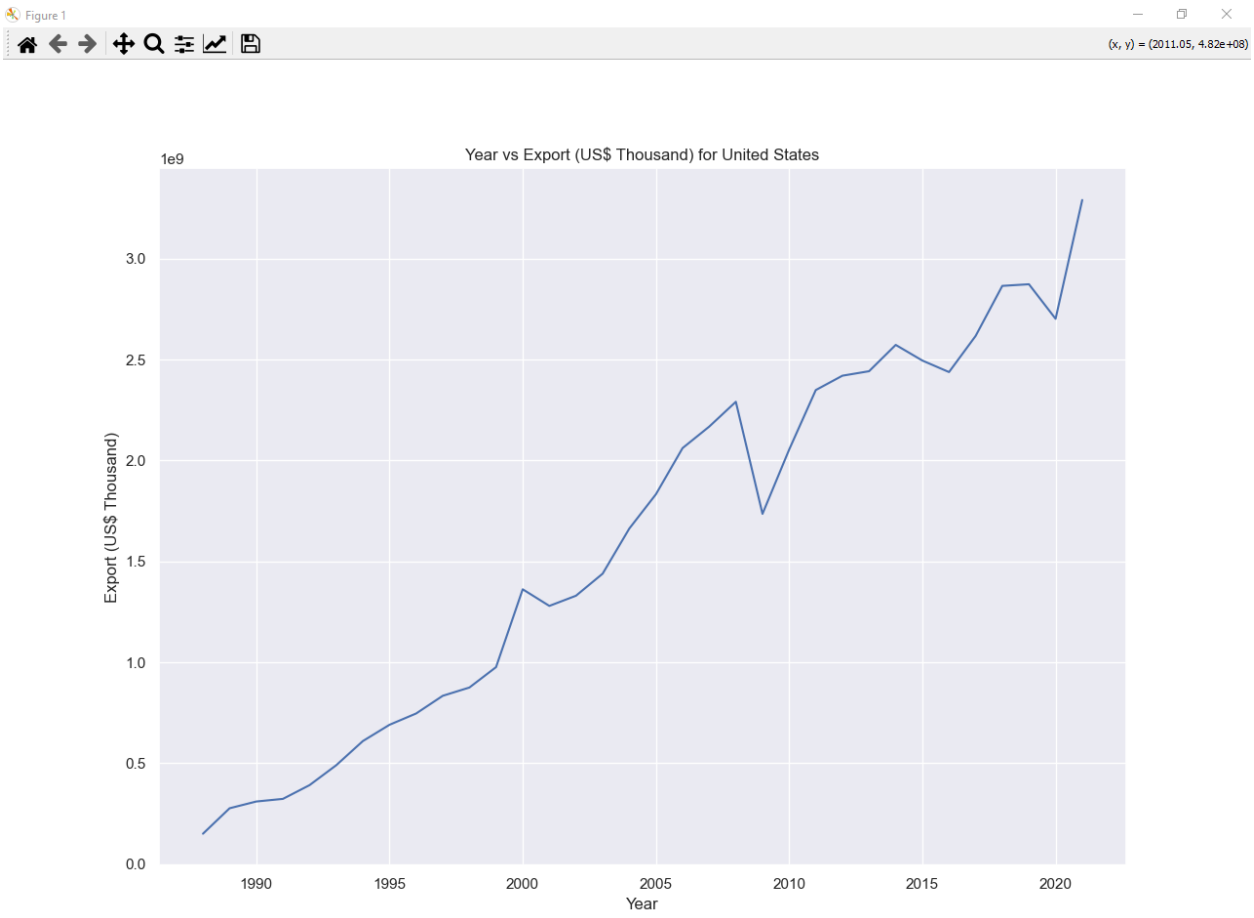
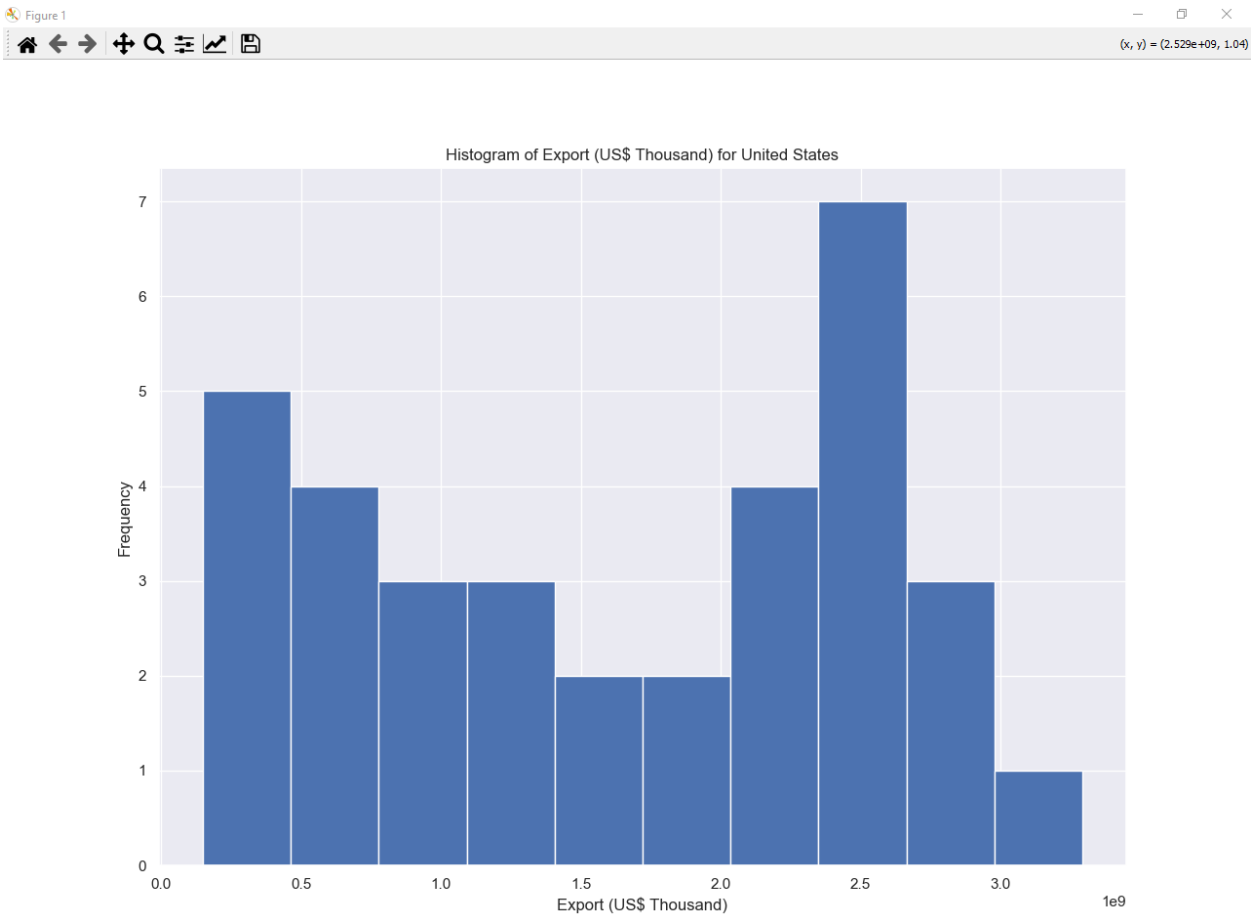
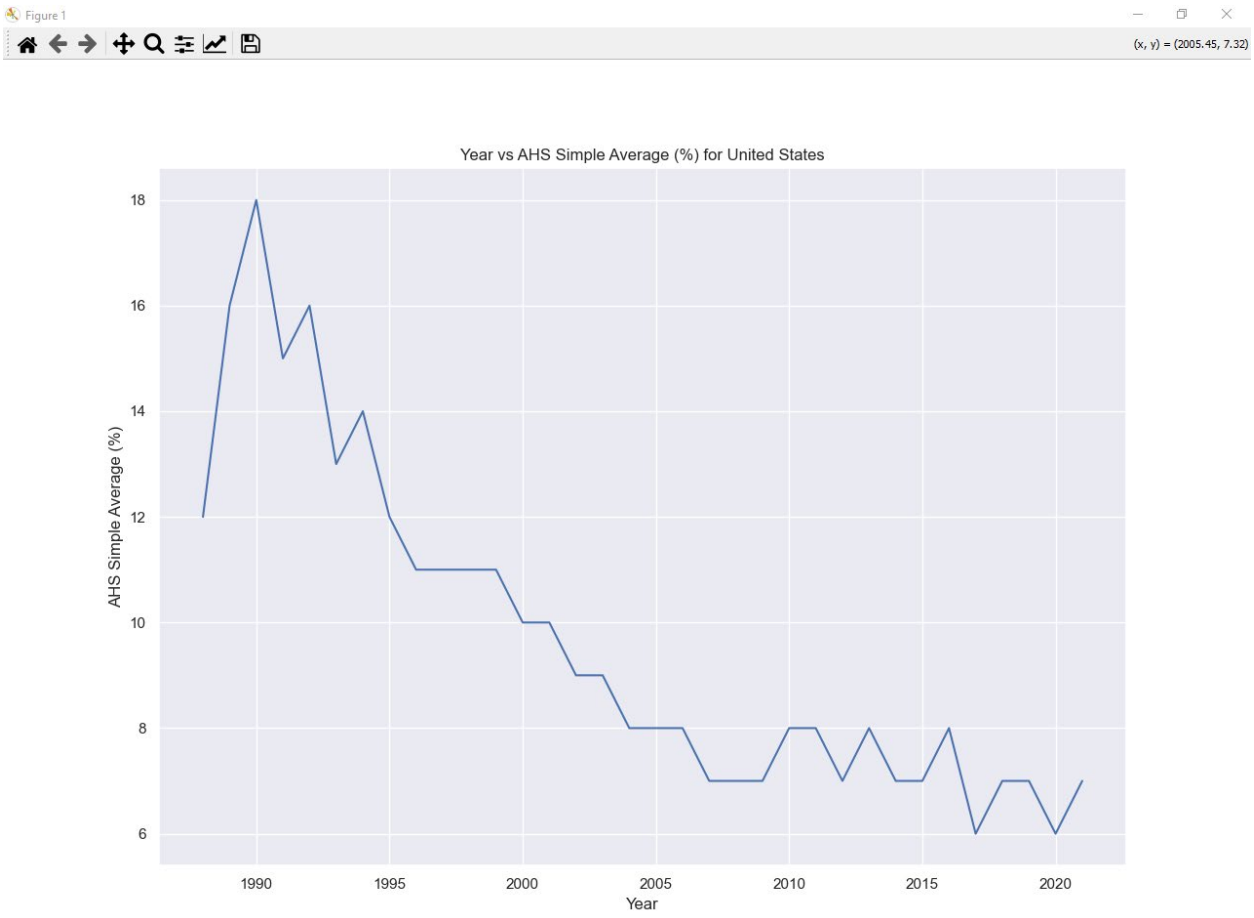


Figure 1
(x, y) = (2011.05, 4.82e+08)
Year vs Export (US$ Thousand) for United States
1e9
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Export (US$ Thousand)
1990
1995
2000
2005
2010
2015
2020
Year



The Python code shows visualizations in order to detect any anomalies. There are no anomalies showing in these visualizations.



The Python code is detecting missing values, zero values or negative values. None were found.

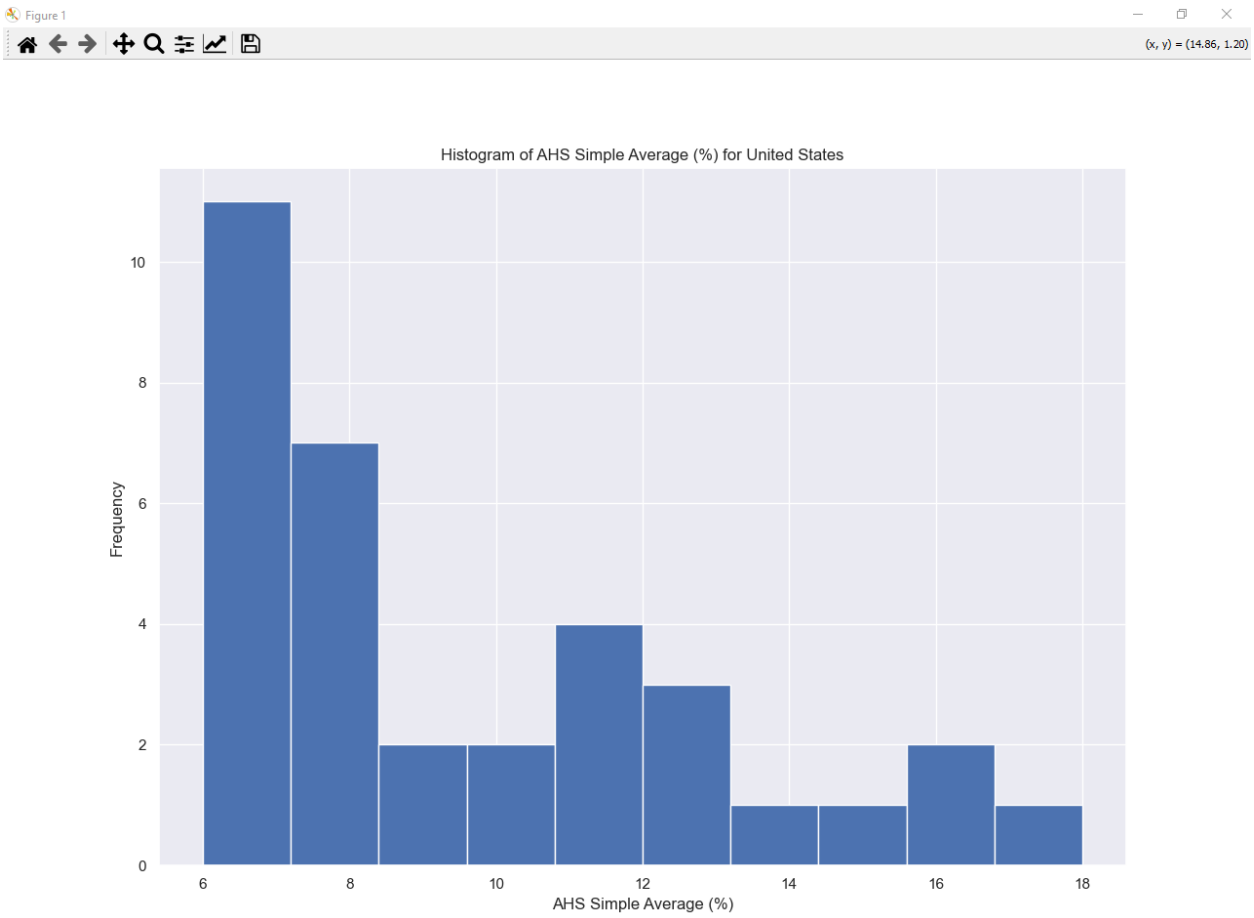
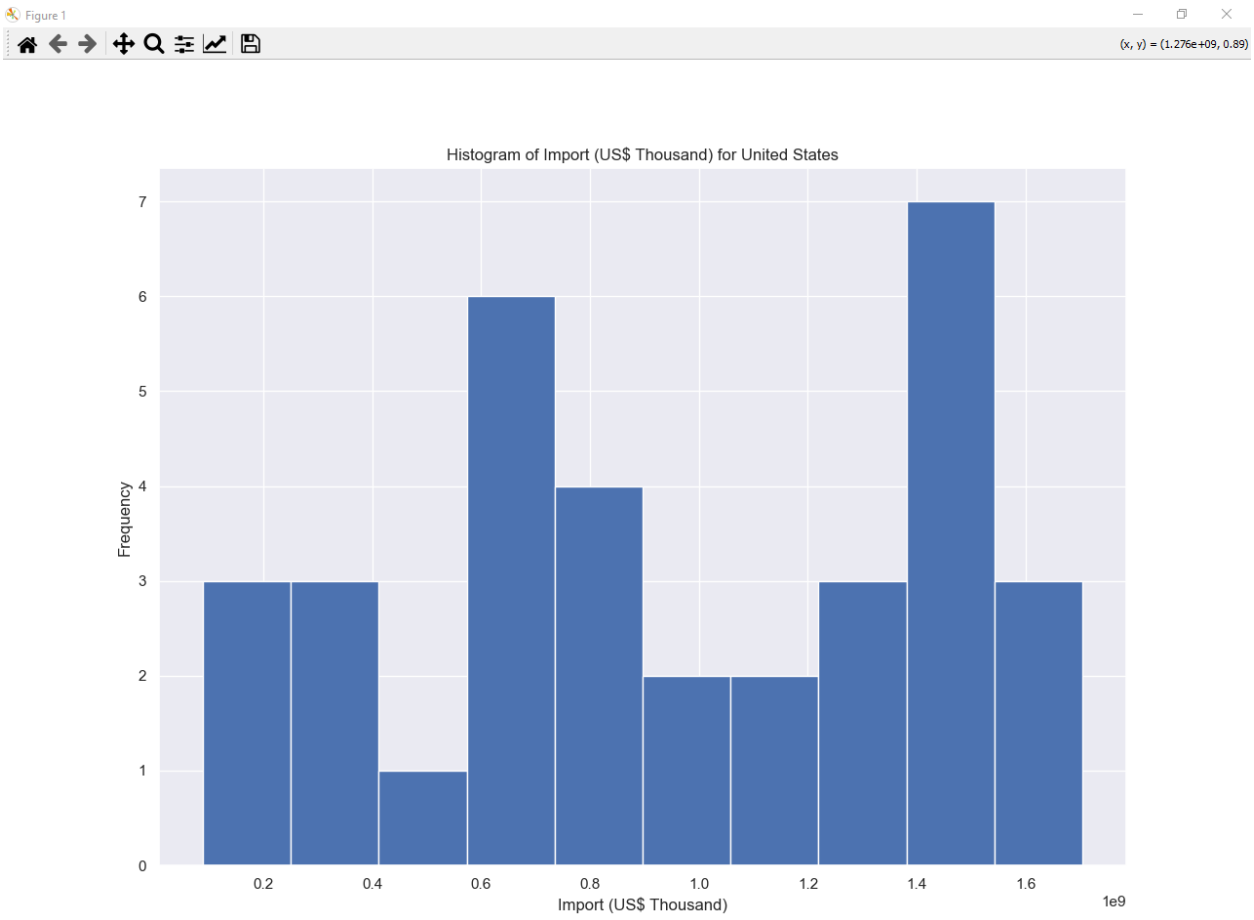


Figure 1
(x, y) = (14.86, 1.20)
Histogram of AHS Simple Average (%) for United States
Frequency
0
2
4
6
8
10
6
8
10
12
14
16
18
AHS Simple Average (%)



No zeros or negative values. It is not a bell shaped curve because the imports are constantly growing and do not trade in a range.

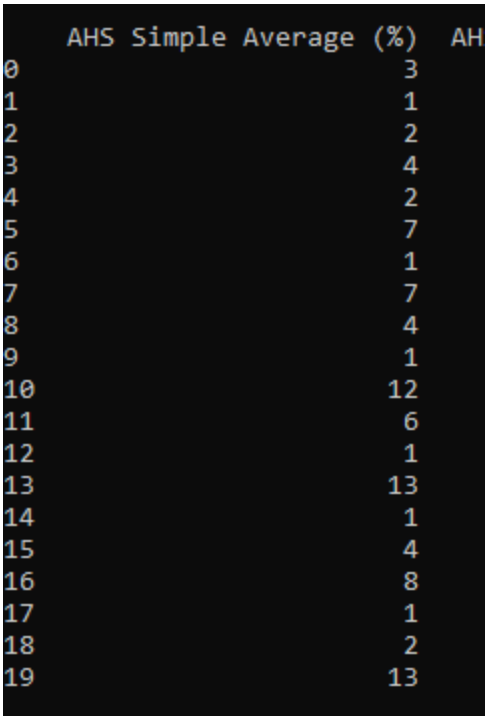
After

Some characters were removed. No outliers were found. Looked for non integers, nulls, zeros and negative numbers. The charts did not change.

(e) Write Python code to find, report, and correct any values in the dataset(s) that might be correct but in the wrong format. For example, suppose you have a variable called "State" and the

values can be state abbreviations like FL, OR, CA, etc. However, one of the entries is Fla. We know that this is FL and it needs to be updated to be the right (expected) format as prescribed by the dataset. Again, there are an infinite number of possibilities because all datasets are different. Take your time, explore your data, and determine how best to clean it.

Before



	AHS Simple Average (%)	AH
0	3	
1	1	
2	2	
3	4	
4	2	
5	7	
6	1	
7	7	
8	4	
9	1	
10	12	
11	6	
12	1	
13	13	
14	1	
15	4	
16	8	
17	1	
18	2	
19	13	

	AHS MaxRate (%)	AHS MinRate (%)	A
0	50.00	0.0	
1	35.00	0.0	
2	40.00	0.0	
3	35.00	0.0	
4	25.00	0.0	
5	43.96	0.0	
6	40.00	0.0	
7	352.69	0.0	
8	1296.17	0.0	
9	30.00	0.0	
10	3000.00	0.0	
11	2029.66	0.0	
12	50.00	0.0	
13	110.00	0.0	
14	30.00	0.0	
15	50.00	0.0	
16	472.46	0.0	
17	50.00	0.0	
18	45.00	0.0	
19	3000.00	0.0	

After

	AHS Simple Average (%)	A
0	3	
1	1	
2	2	
3	4	
4	2	
5	7	
6	1	
7	7	
8	4	
9	1	
10	12	
11	6	
12	1	
13	13	
14	1	
15	4	
16	8	
17	1	
18	2	
19	13	

	AHS MaxRate (%)	AHS MinRate (%)	AH
0	50.0	0.0	
1	35.0	0.0	
2	40.0	0.0	
3	35.0	0.0	
4	25.0	0.0	
5	44.0	0.0	
6	40.0	0.0	
7	352.7	0.0	
8	1296.2	0.0	
9	30.0	0.0	
10	3000.0	0.0	
11	2029.7	0.0	
12	50.0	0.0	
13	110.0	0.0	
14	30.0	0.0	
15	50.0	0.0	
16	472.5	0.0	
17	50.0	0.0	
18	45.0	0.0	
19	3000.0	0.0	

The Python code changes the format to 1 decimal places.

(f) Write Python code to find, visualize, and correct any outliers.

Before

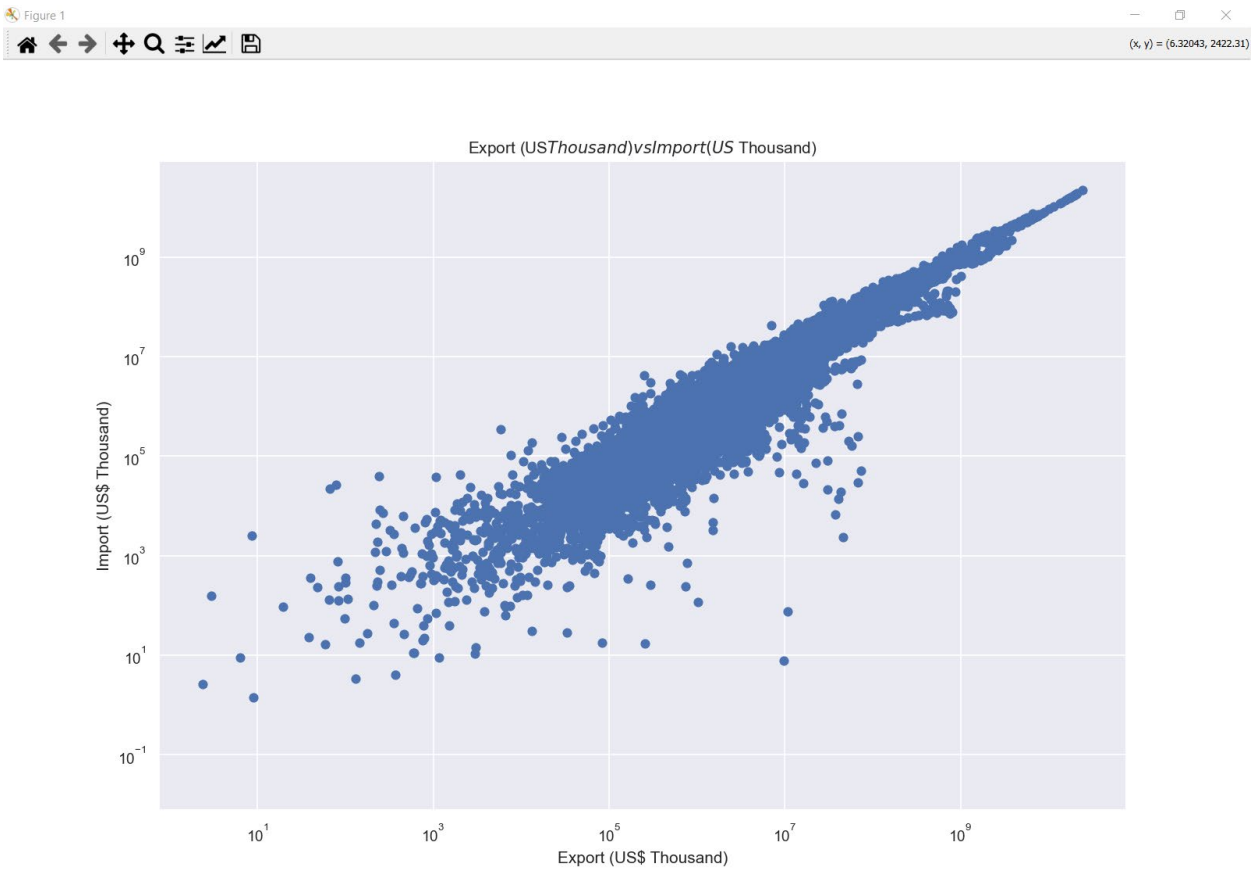
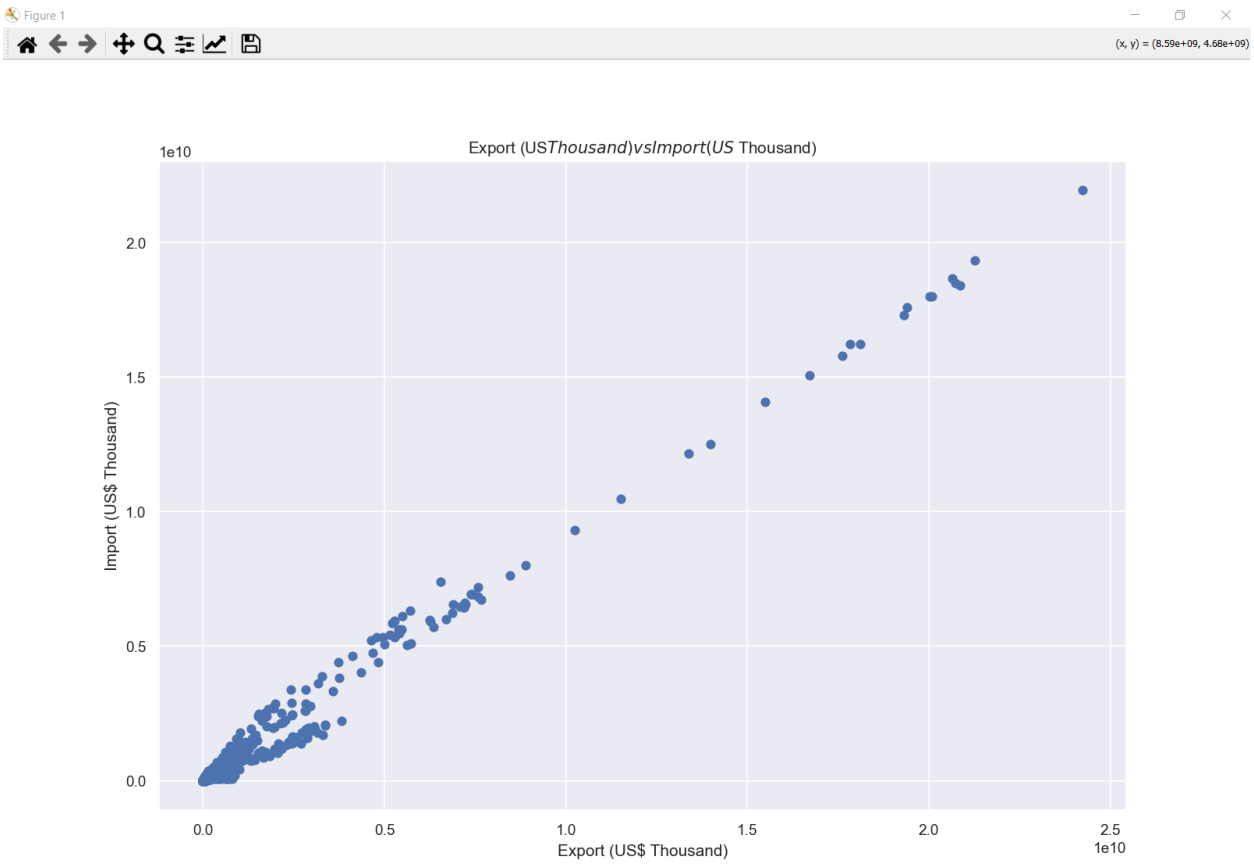


Figure 1
(x, y) = (6.32043, 2422.31)
Export (US*Thousand*)*vs*Import(*US* Thousand)
Import (US$ Thousand)
Export (US$ Thousand)
10^{9}
10^{7}
10^{5}
10^{3}
10^{1}
10^{-1}
10^{1}
10^{3}
10^{5}
10^{7}
10^{9}

Log scale



XY scatter plot of Export vs Import for US. No outliers. No zero or negative values. Values are in a channel as expected.

Linear scale

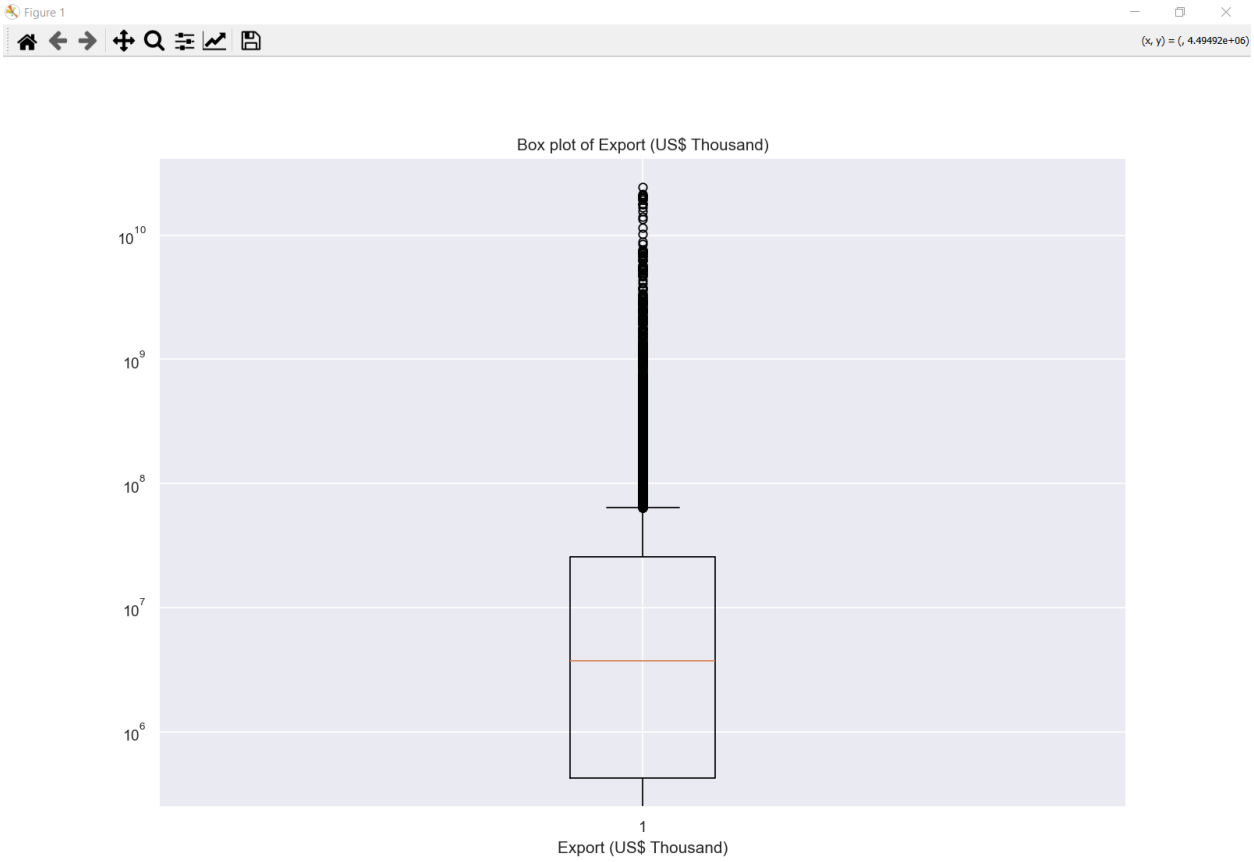
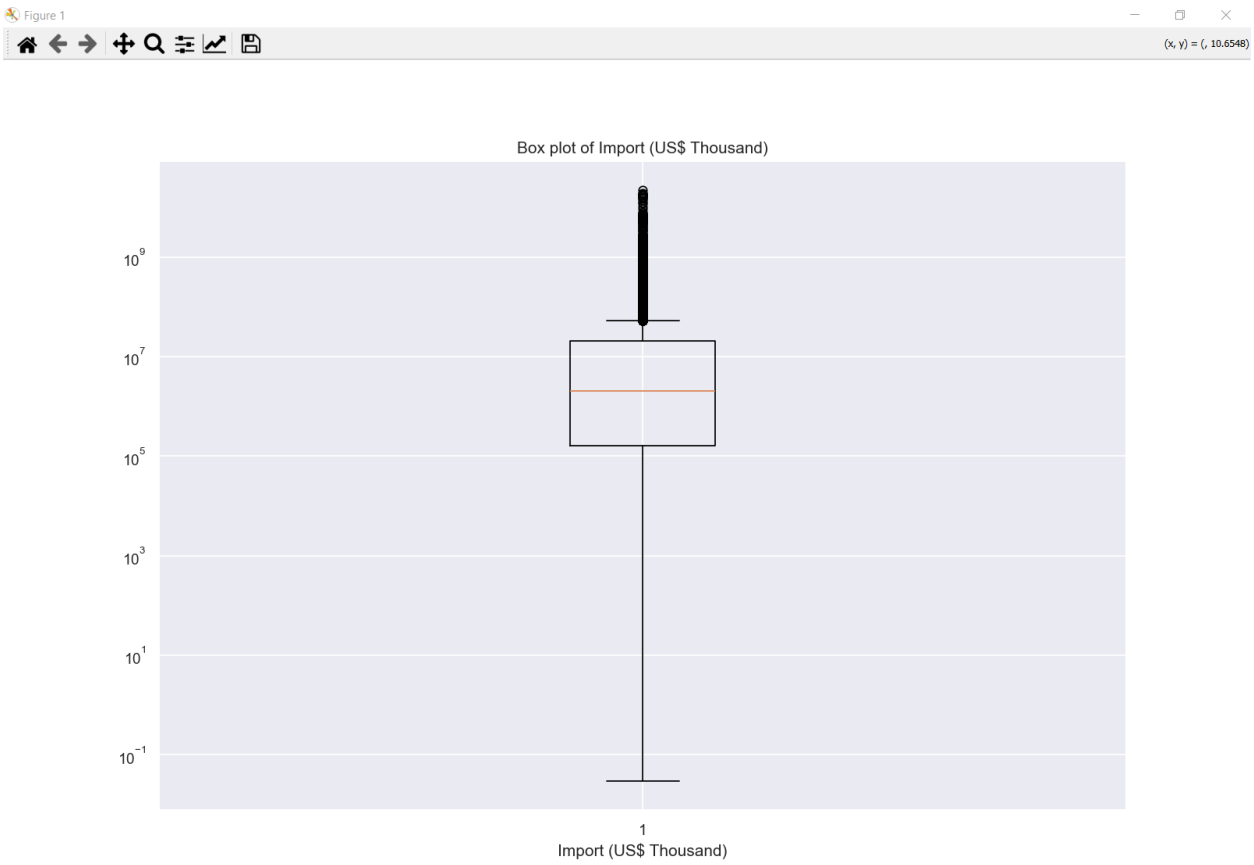
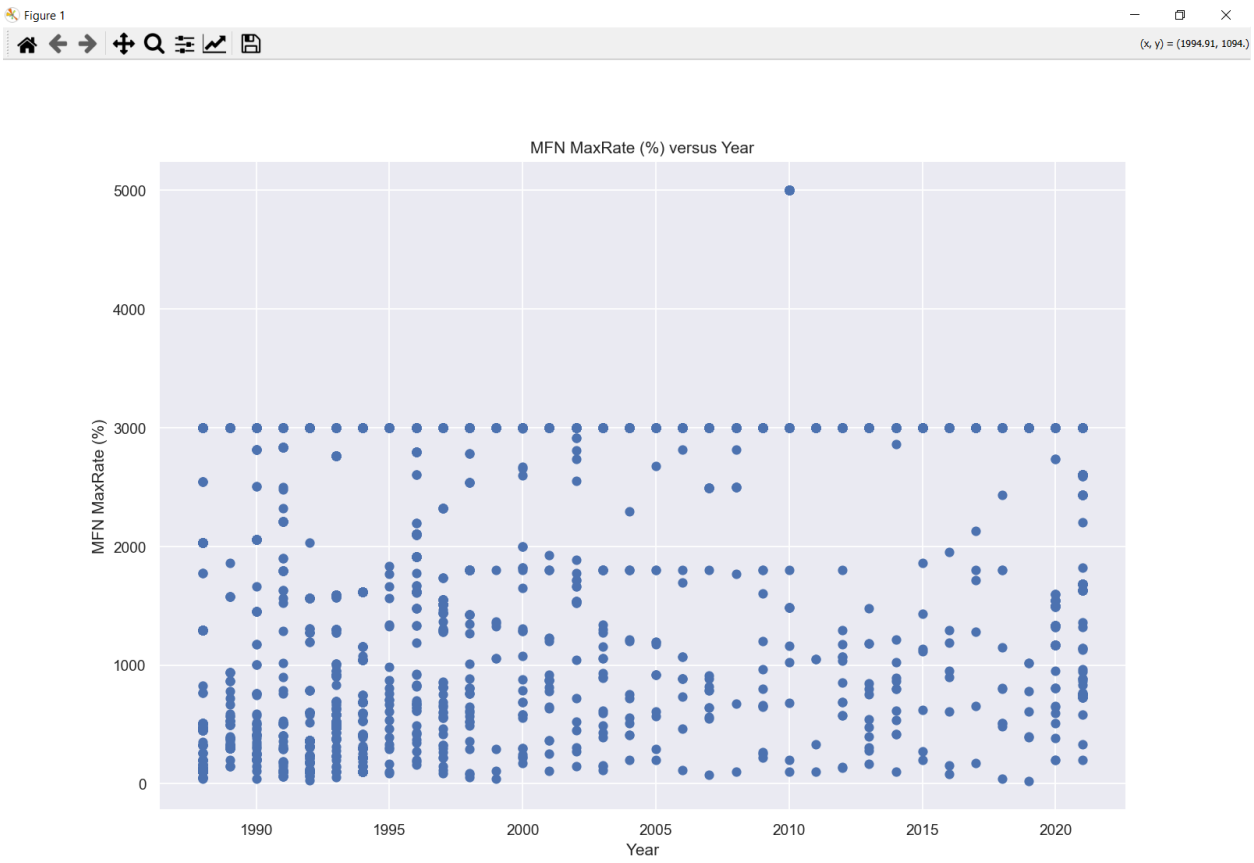


Figure 1
(x, y) = (, 4.49492e+06)
Box plot of Export (US$ Thousand)
10^{10}
10^{9}
10^{8}
10^{7}
10^{6}
1
Export (US$ Thousand)

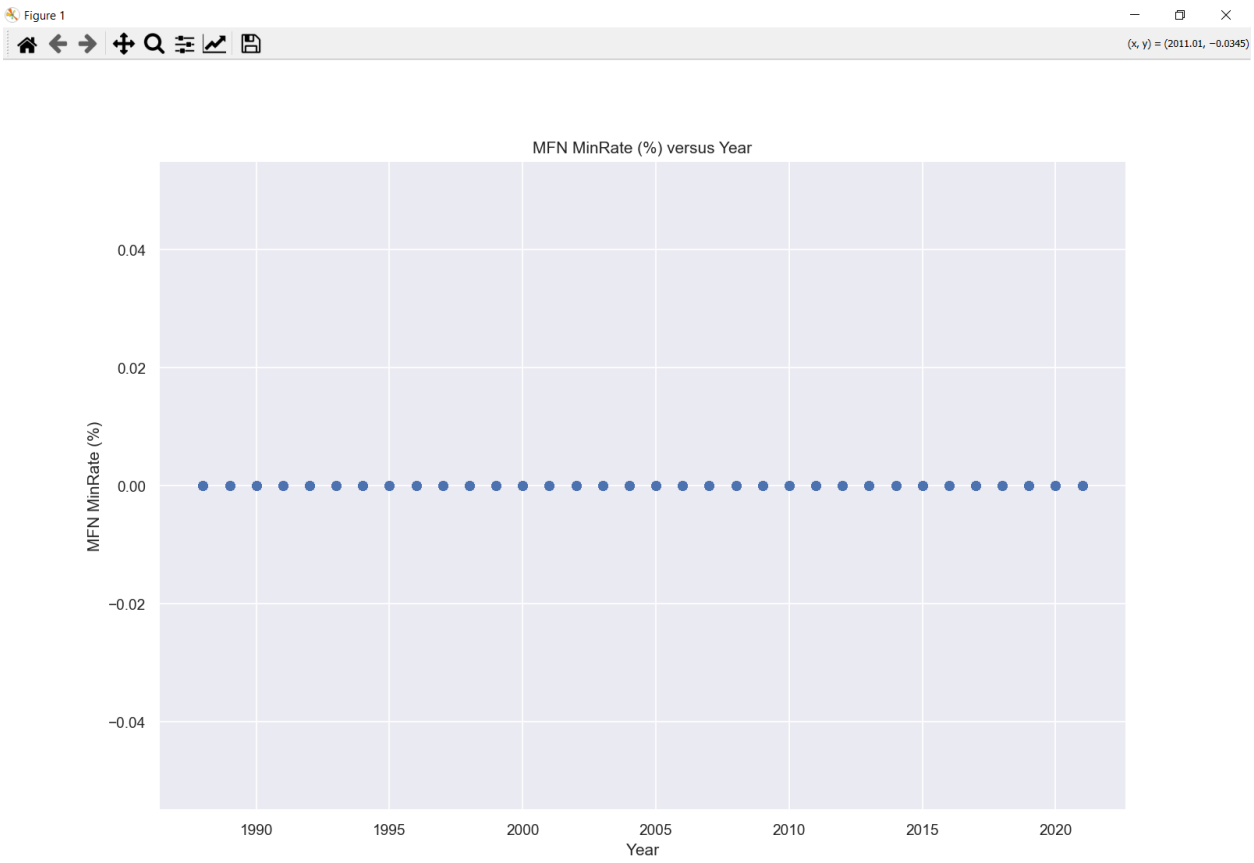


The Python code made boxplots but were harder to detect outliers because the data trends upward.

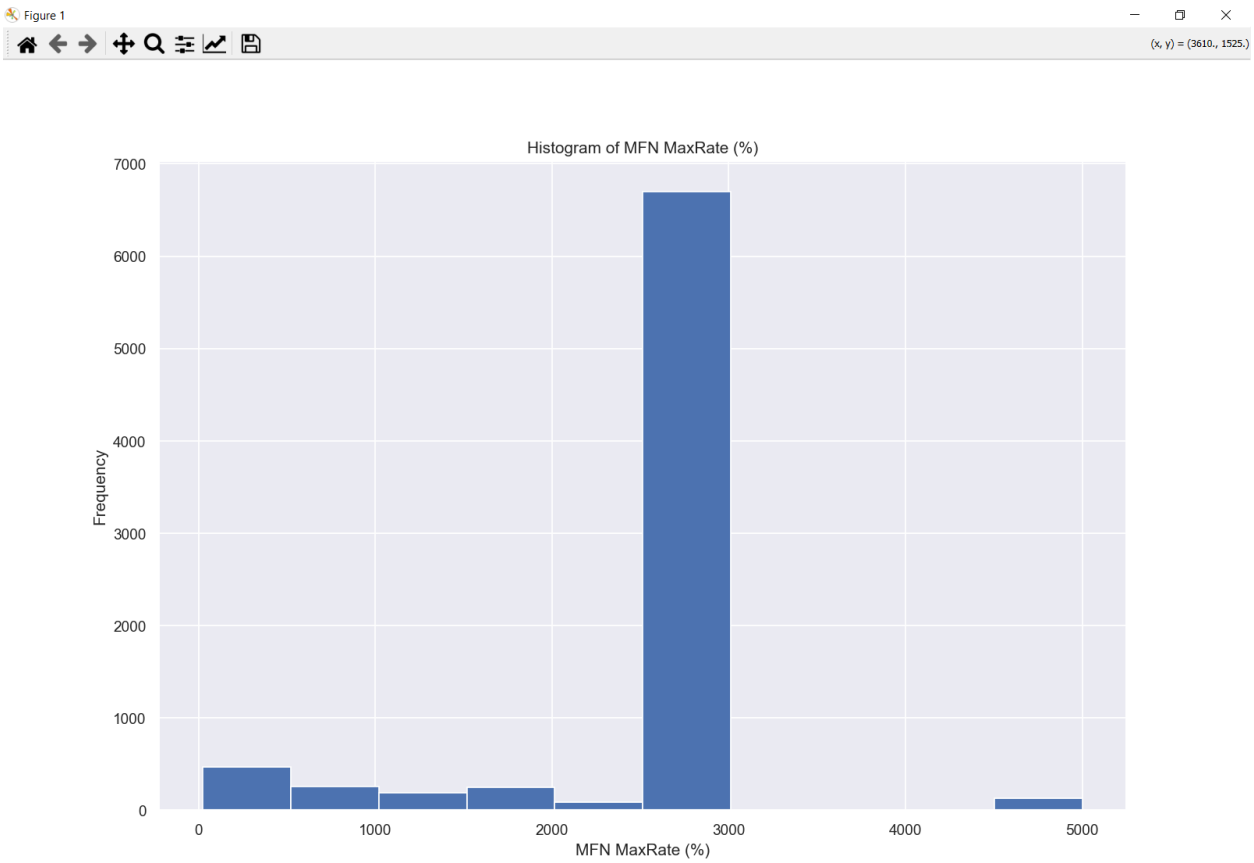
MFN MaxRate (%) and MFN MinRate (%) for all countries and years



The 5000 values look like an outlier and is removed. The Most Favored Nation Status fixes the maximum and minimum tariff for its members.

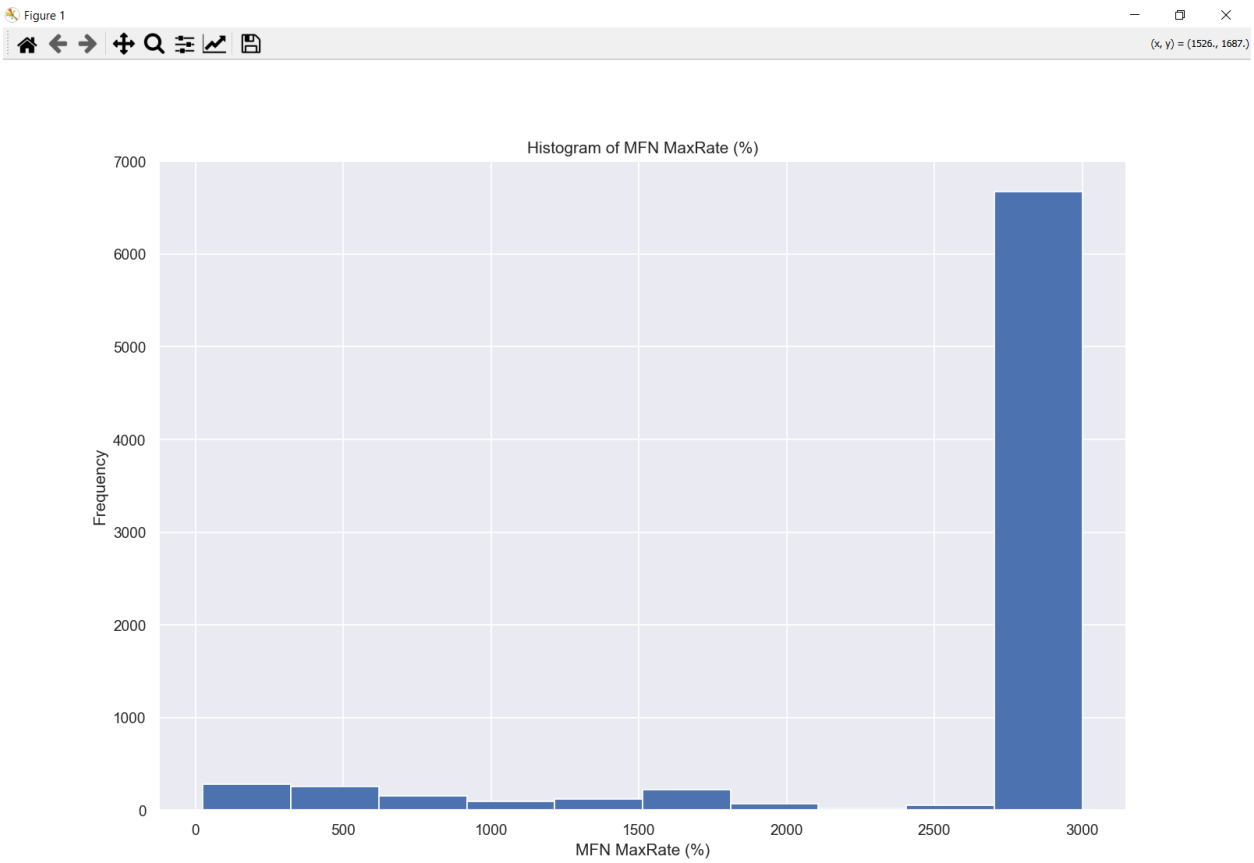


Zero is not an outlier. By joining the WTO, Most Favored Nation status gives the privilege of low tariffs.



The 5000 values looks like an outlier which is detected by this visual. The 3000 value is the maximum rate for MFN members.

After



The Python code deleted the 8000 value and any values below 0.

(g) Write Python code to create a new dataframe from one of your datasets such that the new dataframe is normalized using min-max. Be sure to include a screen image of the normalized dataset in your report.

The columns "Export (US$ Thousand)" and "Import (US$ Thousand)" are chosen because they represent continuous numerical data, which is suitable for min-max normalization.

Before

	Export (US$ Thousand)	Import (US$ Thousand)
0	3498.10	328.49
1	213030.40	54459.52
2	375527.89	370702.76
3	366.98	4.00
4	30103.56	47709.30
5	67924.46	3284.01
6	104759.21	24964.14
7	2945350.25	7091823.87
8	1136421.71	1928596.45
9	14406.52	2173.80
10	10508173.98	14350888.96
11	22046961.12	14273975.93
12	37299.67	73592.16
13	66486.37	17352.43
14	42212.93	24547.18

This code will output a normalized dataframe where the values of "Export (US$ Thousand)" and "Import (US$ Thousand)" are scaled to the range [0, 1].

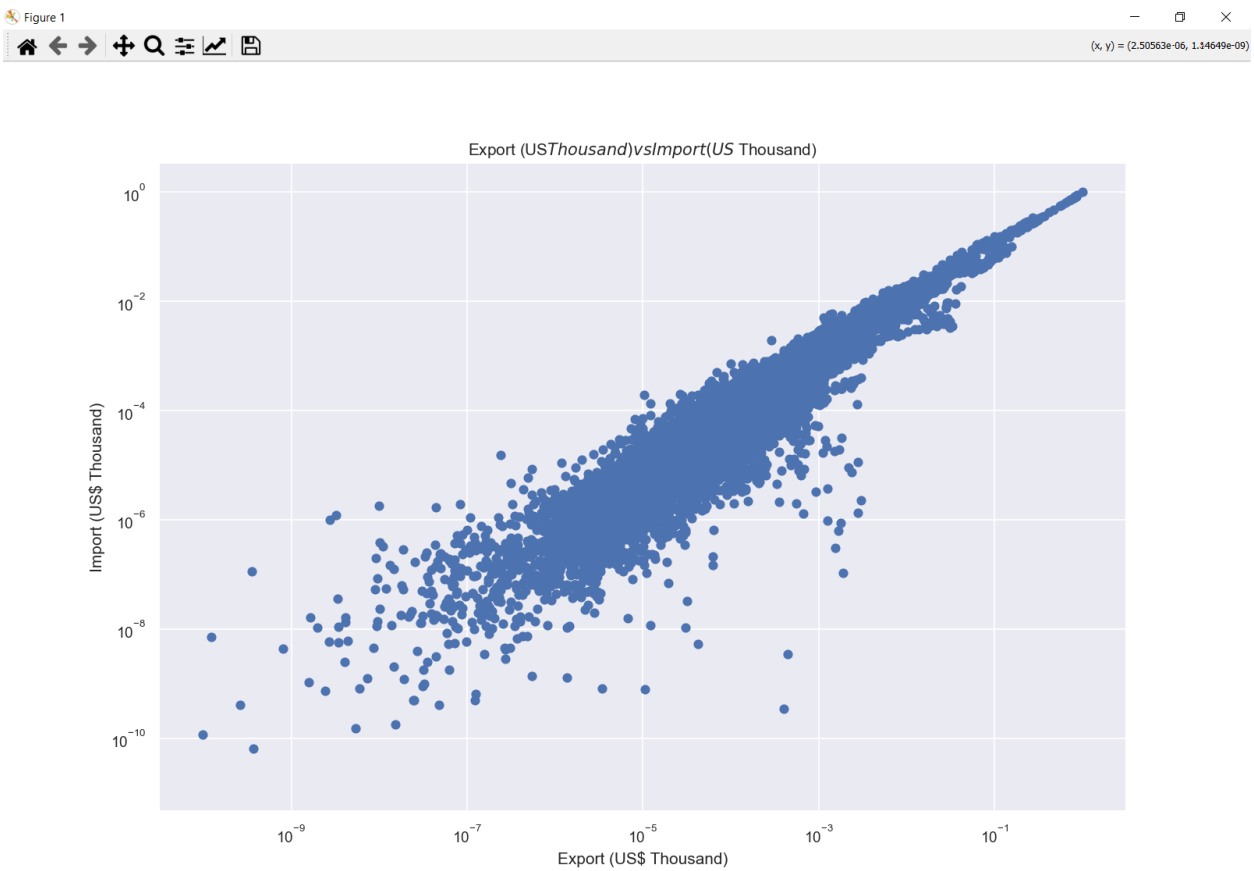
After

```
new dataset normalized_Export and normalized_Import created
```

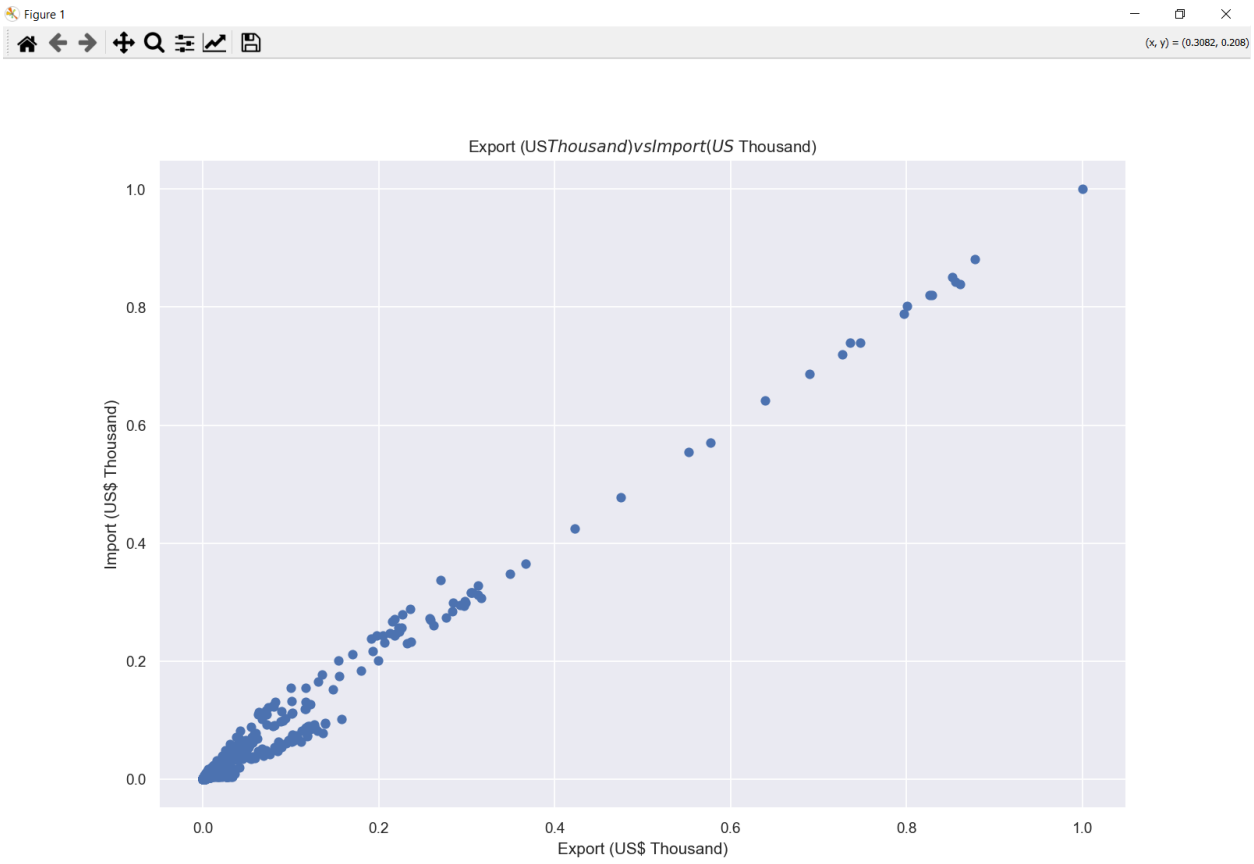
A new dataset name stores the normalized Import and Export data. It can be printed and charted from new dataset.

	Export (US$ Thousand)	Import (US$ Thousand)
0	1.443859e-07	1.497683e-08
1	8.792942e-06	2.483196e-06
2	1.550011e-05	1.690297e-05
3	1.514729e-08	1.810205e-10
4	1.242540e-06	2.175405e-06
5	2.803618e-06	1.497400e-07
6	4.323991e-06	1.138291e-06
7	1.215709e-04	3.233667e-04
8	4.690640e-05	8.793843e-05
9	5.946367e-07	9.911764e-08
10	4.337304e-04	6.543591e-04
11	9.099999e-04	6.508521e-04
12	1.539563e-06	3.355589e-06
13	2.744260e-06	7.912193e-07
14	1.742361e-06	1.119279e-06

The Python code normalizes the values ranges between 0 and 1. Now both sets of columns can be compared because it is comparing oranges to oranges.



Looks the same except axis ranges are 0 to 1. log scale



Linear scale

(h) Write Python code to create a new dataframe that contains only unlabeled and quantitative data. Be sure to include a screen image in your report.

"country" column is qualitative. So, "country" column will not be included in new dataset.

New dataset just contains numerical columns.

Before

	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	
5	Andorra	1988	67924.46	3284.01	
6	Netherlands Antilles	1988	104759.21	24964.14	
7	United Arab Emirates	1988	2945350.25	7091823.87	
8	Argentina	1988	1136421.71	1928596.45	
9	Antigua and Barbuda	1988	14406.52	2173.80	
10	Australia	1988	10508173.98	14350888.96	
11	Austria	1988	22046961.12	14273975.93	
12	Burundi	1988	37299.67	73592.16	
13	Benin	1988	66486.37	17352.43	
14	Burkina Faso	1988	42212.93	24547.18	

Column Partner Name is the only quantitative data and will be deleted by the Python code.

After

```
new dataset MyData_quant created
```

The new numeric dataset was assigned a new name. Printing and charting can now be done with this new dataset.

	Year	Export (US$ Thousand)	Import (US$ Thousand)	\
0	1988	3498.10	328.49	
1	1988	213030.40	54459.52	
2	1988	375527.89	370702.76	
3	1988	366.98	4.00	
4	1988	30103.56	47709.30	
5	1988	67924.46	3284.01	
6	1988	104759.21	24964.14	
7	1988	2945350.25	7091823.87	
8	1988	1136421.71	1928596.45	
9	1988	14406.52	2173.80	
10	1988	10508173.98	14350888.96	
11	1988	22046961.12	14273975.93	
12	1988	37299.67	73592.16	
13	1988	66486.37	17352.43	
14	1988	42212.93	24547.18	

It uses a Panda command to look at the values in each column to see whether it is numeric or non-numeric values. The Python code deleted the Partner Name column.

(i) Finally, take any further steps you feel are needed to clean up your data such as discretization and feature generation.

No further action. All the previous steps have checked for negative values, zero values, non numeric values, missing values, null vales and outlier values.

Dataset 2 Exports and Imports of India(1997-July 2022)

Dataset is from Kaggle. India was the eighth largest exporter of commercial services in the world in 2016, accounting for 3.4 % of global trade in services. India recorded a 5.7% growth in services trade in 2016.

(a) Write Python code to assure that your datasets are in record format so that they are structured as rows and columns, where each column has a variable name.

Before

	A	B	C	D	E	F	G	H	I	J
1	Country	Export	Import	Total Trade	Trade Balance	Financial Year(start)	Financial Year(end)			
2	AFGHANISTAN	21.25	10.7	31.95	10.55	1997	1998			
3	AFGHANISTAN	12.81	28.14	40.95	-15.33	1998	1999			
4	AFGHANISTAN	33.2	21.06	54.26	12.15	1999	2000			
5	AFGHANISTAN	25.86	26.59	52.45	-0.73	2000	2001			
6	AFGHANISTAN	24.37	17.52	41.89	6.85	2001	2002			
7	AFGHANISTAN	60.77	18.46	79.23	42.31	2002	2003			
8	AFGHANISTAN	145.47	40.51	185.98	104.96	2003	2004			
9	AFGHANISTAN	165.44	47.01	212.44	118.43	2004	2005			
10	AFGHANISTAN	142.67	58.42	201.09	84.24	2005	2006			
11	AFGHANISTAN	182.11	34.37	216.48	147.73	2006	2007			
12	AFGHANISTAN	249.21	109.97	359.18	139.24	2007	2008			
13	AFGHANISTAN	394.23	126.24	520.47	268	2008	2009			
14	AFGHANISTAN	463.55	125.19	588.74	338.36	2009	2010			
15	AFGHANISTAN	422.41	146.03	568.44	276.38	2010	2011			
16	AFGHANISTAN	510.9	132.5	643.41	378.4	2011	2012			
17	AFGHANISTAN	472.63	159.55	632.18	313.07	2012	2013			
18	AFGHANISTAN	474.34	208.77	683.1	265.57	2013	2014			
19	AFGHANISTAN	422.56	261.91	684.47	160.65	2014	2015			
20	AFGHANISTAN	526.6	307.9	834.5	218.7	2015	2016			
21	AFGHANISTAN	506.34	292.9	799.24	213.44	2016	2017			
22	AFGHANISTAN	709.75	433.78	1,143.53	275.97	2017	2018			
23	AFGHANISTAN	715.44	435.44	1,150.89	280	2018	2019			
24	AFGHANISTAN	997.58	529.84	1,527.42	467.74	2019	2020			
25	AFGHANISTAN	825.78	509.49	1,335.27	316.29	2020	2021			
26	AFGHANISTAN	554.47	510.93	1,065.40	43.54	2021	2022			
27	AFGHANISTAN	147.56	94.03	241.58	53.53	2022	till now			
28	ALBANIA	0.55	0.03	0.59	0.52	1997	1998			
29	ALBANIA	0.8	0.18	0.97	0.62	1998	1999			

exports and imports of india(19

Big csv file. The Excel program gave the text file a lot of column order.

After

```
Data loaded successfully
         Country  Export Import Total Trade Trade Balance  \
0     AFGHANISTAN   21.25   10.7       31.95         10.55
1     AFGHANISTAN   12.81  28.14       40.95        -15.33
2     AFGHANISTAN    33.2  21.06       54.26         12.15
3     AFGHANISTAN   25.86  26.59       52.45         -0.73
4     AFGHANISTAN   24.37  17.52       41.89          6.85
...           ...     ...    ...         ...           ...
5989     ZIMBABWE  181.72   7.79      189.51        173.93
5990     ZIMBABWE  161.13  13.59      174.72        147.54
5991     ZIMBABWE  175.72   5.71      181.42        170.01
5992     ZIMBABWE  200.49   7.77      208.27        192.72
5993     ZIMBABWE   50.17   0.61       50.78         49.57

      Financial Year(start) Financial Year(end)
0                      1997                1998
1                      1998                1999
2                      1999                2000
3                      2000                2001
4                      2001                2002
...                     ...                 ...
5989                   2018                2019
5990                   2019                2020
5991                   2020                2021
5992                   2021                2022
5993                   2022            till now
```

The Python code read in the csv file and Panda converted it into a dataframe. It prints out “data loaded successfully.”

(b) Write Python code to check and print the data types of the variables in your dataset. Write code to correct any data types. For example, if Python reads in a categorical variable as a number, you will need to update this to a category.

Before

```
Data loaded successfully
<class 'pandas.core.frame.DataFrame'>


Country                 object
Export                  object
Import                  object
Total Trade             object
Trade Balance           object
Financial Year(start)    int64
Financial Year(end)     object
dtype: object
```

We need to do part (c) before we can change the column objects to what we want. Missing values prevents the Python code from converting one datatype to another datatype.

After

```
Data saved to MyNew_CleanFile.csv
Country                 0
Export                  0
Import                  0
Total Trade             0
Trade Balance           0
Financial Year(start)   0
Financial Year(end)     0
dtype: int64


<class 'pandas.core.frame.DataFrame'>
Index: 5949 entries, 0 to 5993
Data columns (total 7 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Country               5949 non-null   object
 1   Export                5949 non-null   float64
 2   Import                5949 non-null   float64
 3   Total Trade           5949 non-null   float64
 4   Trade Balance         5949 non-null   float64
 5   Financial Year(start) 5949 non-null   int64
 6   Financial Year(end)   5949 non-null   object
dtypes: float64(4), int64(1), object(2)
memory usage: 371.8+ KB
None
```

After all the missing values were filled in, the Python code converted four columns from object to float. It needs to be float because algebra may need to be performed on them.

(c) Write Python code to find, count, report, and then clean any missing values.

Before

```
Data loaded successfully
Country                    0
Export                     8
Import                   552
Total Trade              585
Trade Balance            586
Financial Year(start)      0
Financial Year(end)        0
dtype: int64


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5994 entries, 0 to 5993
Data columns (total 7 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Country               5994 non-null   object
 1   Export                5986 non-null   object
 2   Import                5442 non-null   object
 3   Total Trade           5409 non-null   object
 4   Trade Balance         5408 non-null   object
 5   Financial Year(start) 5994 non-null   int64
 6   Financial Year(end)   5994 non-null   object
dtypes: int64(1), object(6)
memory usage: 327.9+ KB
None
```

In Export column there are commas separating values. Import column has many null values and missing values. Export, Import, Total Trade and Trade Balance all need to be changed to a floating real number. Zero and negative values need to be replaced with the mean.

There are a couple countries that do not have any Total Trade numbers. So, a mean cannot be determined for them. They were deleted. For example countries, SAHARWI A.DM RP, SINT MAARTEN (DUTCH PART), NEUTRAL ZONE

After

```
Data saved to MyNew_CleanFile.csv
Country                 0
Export                  0
Import                  0
Total Trade             0
Trade Balance           0
Financial Year(start)   0
Financial Year(end)     0
dtype: int64


<class 'pandas.core.frame.DataFrame'>
Index: 5949 entries, 0 to 5993
Data columns (total 7 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Country               5949 non-null   object
 1   Export                5949 non-null   float64
 2   Import                5949 non-null   float64
 3   Total Trade           5949 non-null   float64
 4   Trade Balance         5949 non-null   float64
 5   Financial Year(start) 5949 non-null   int64
 6   Financial Year(end)   5949 non-null   object
dtypes: float64(4), int64(1), object(2)
memory usage: 371.8+ KB
None
```

Clean dataset now. all the missing values and null values were filled in with the column mean. We will use clean file for outlier discovery. All the numbers are numeric types now.

(d) Write Python code to find, report, and correct any incorrect values. You can use visual methods here. For example, you can "report" incorrect values visually.

Before

```
Data loaded successfully
Country                    0
Export                     8
Import                   552
Total Trade              585
Trade Balance            586
Financial Year(start)      0
Financial Year(end)        0
dtype: int64


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5994 entries, 0 to 5993
Data columns (total 7 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Country                5994 non-null   object
 1   Export                 5986 non-null   object
 2   Import                 5442 non-null   object
 3   Total Trade            5409 non-null   object
 4   Trade Balance          5408 non-null   object
 5   Financial Year(start)  5994 non-null   int64
 6   Financial Year(end)    5994 non-null   object
dtypes: int64(1), object(6)
memory usage: 327.9+ KB
None
```

There are missing values, null values and incorrect values in the data.

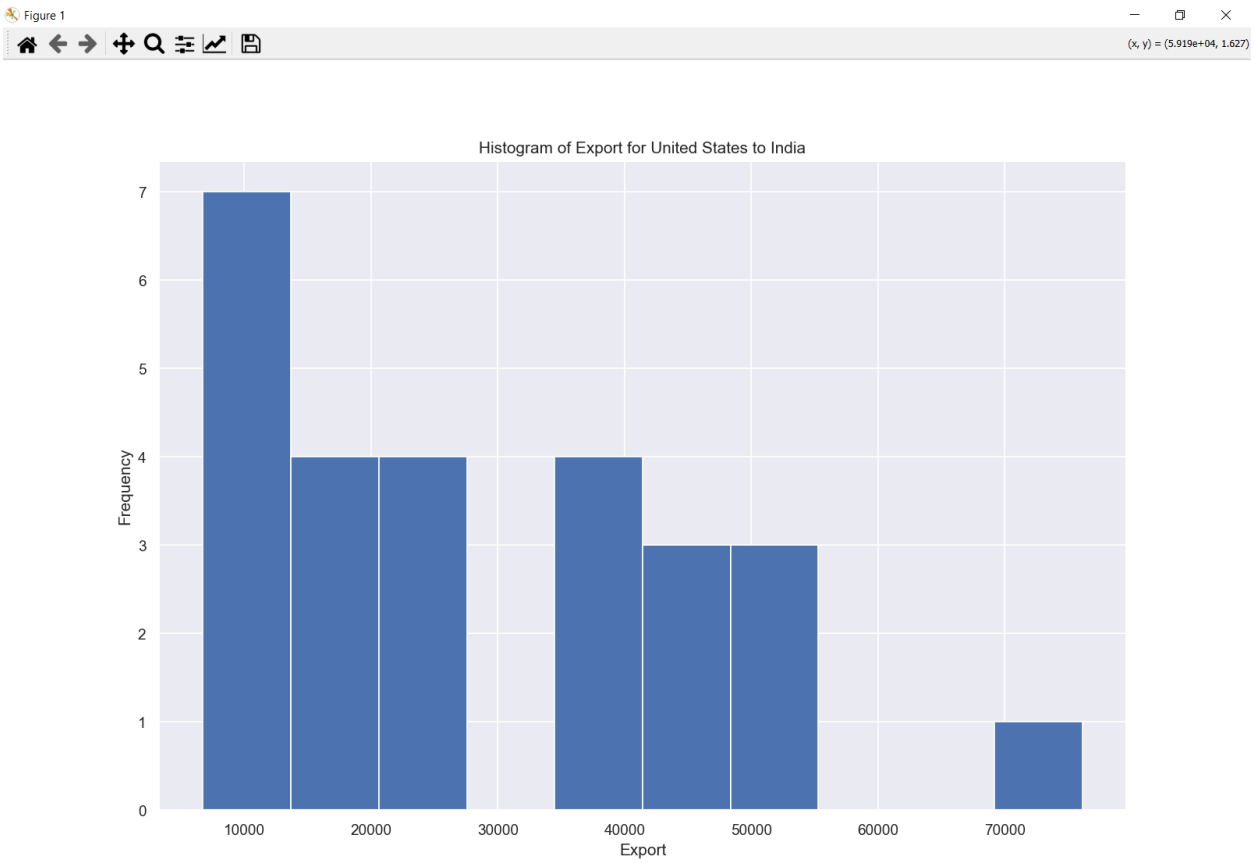


Figure 1
(x, y) = (5.919e+04, 1.627)
Histogram of Export for United States to India
Frequency
0
1
2
3
4
5
6
7
10000
20000
30000
40000
50000
60000
70000
Export

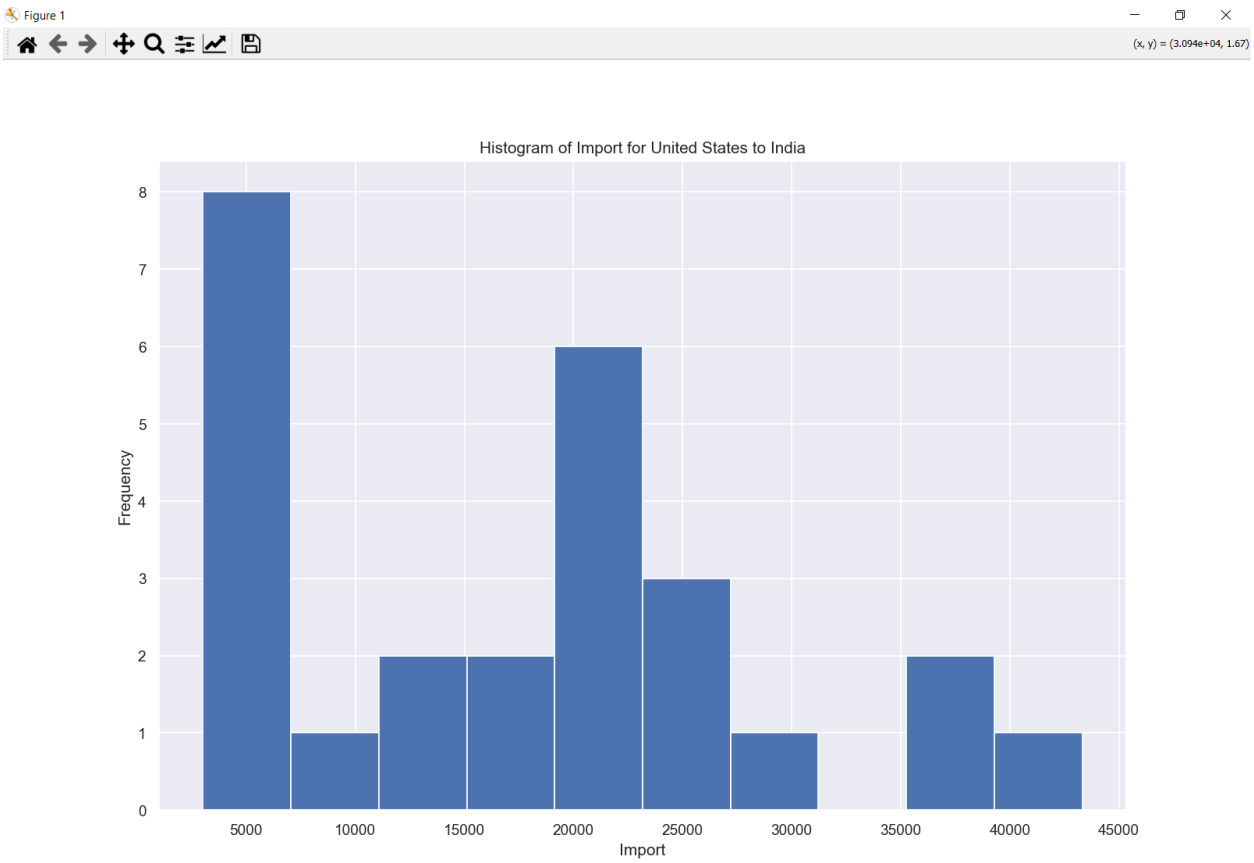
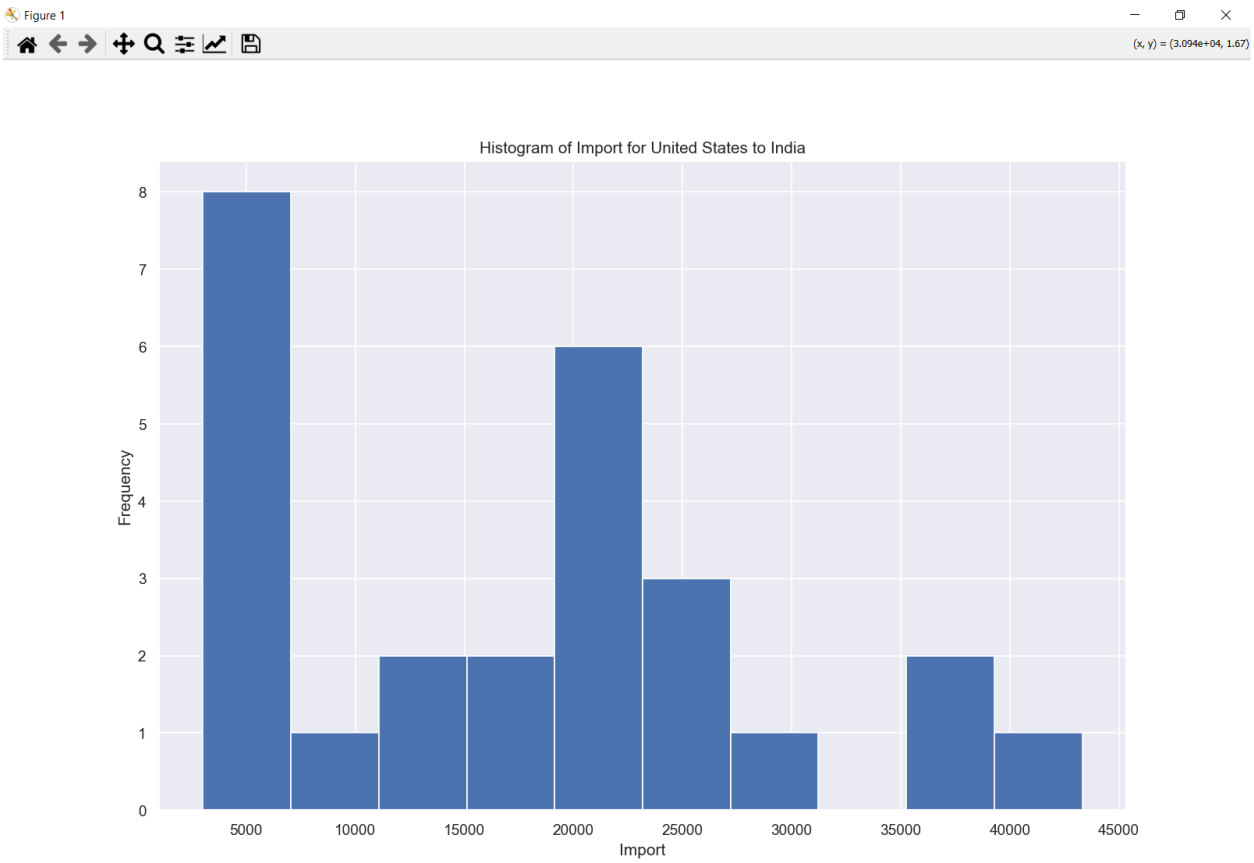


Figure 1
(x, y) = (3.094e+04, 1.67)
Histogram of Import for United States to India
Frequency
Import
0
1
2
3
4
5
6
7
8
5000
10000
15000
20000
25000
30000
35000
40000
45000



The boxplots, histograms and XY charts are visualizations meant to detect anomalies in the data. The histograms are not bell shaped because the data is not random.

After

```
Data saved to MyNew_CleanFile.csv
Country                 0
Export                  0
Import                  0
Total Trade             0
Trade Balance           0
Financial Year(start)   0
Financial Year(end)     0
dtype: int64


<class 'pandas.core.frame.DataFrame'>
Index: 5949 entries, 0 to 5993
Data columns (total 7 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Country               5949 non-null   object
 1   Export                5949 non-null   float64
 2   Import                5949 non-null   float64
 3   Total Trade           5949 non-null   float64
 4   Trade Balance         5949 non-null   float64
 5   Financial Year(start) 5949 non-null   int64
 6   Financial Year(end)   5949 non-null   object
dtypes: float64(4), int64(1), object(2)
memory usage: 371.8+ KB
None
```

The Python code cleaned up all the problems and as can be seen above the Null values are all zero.

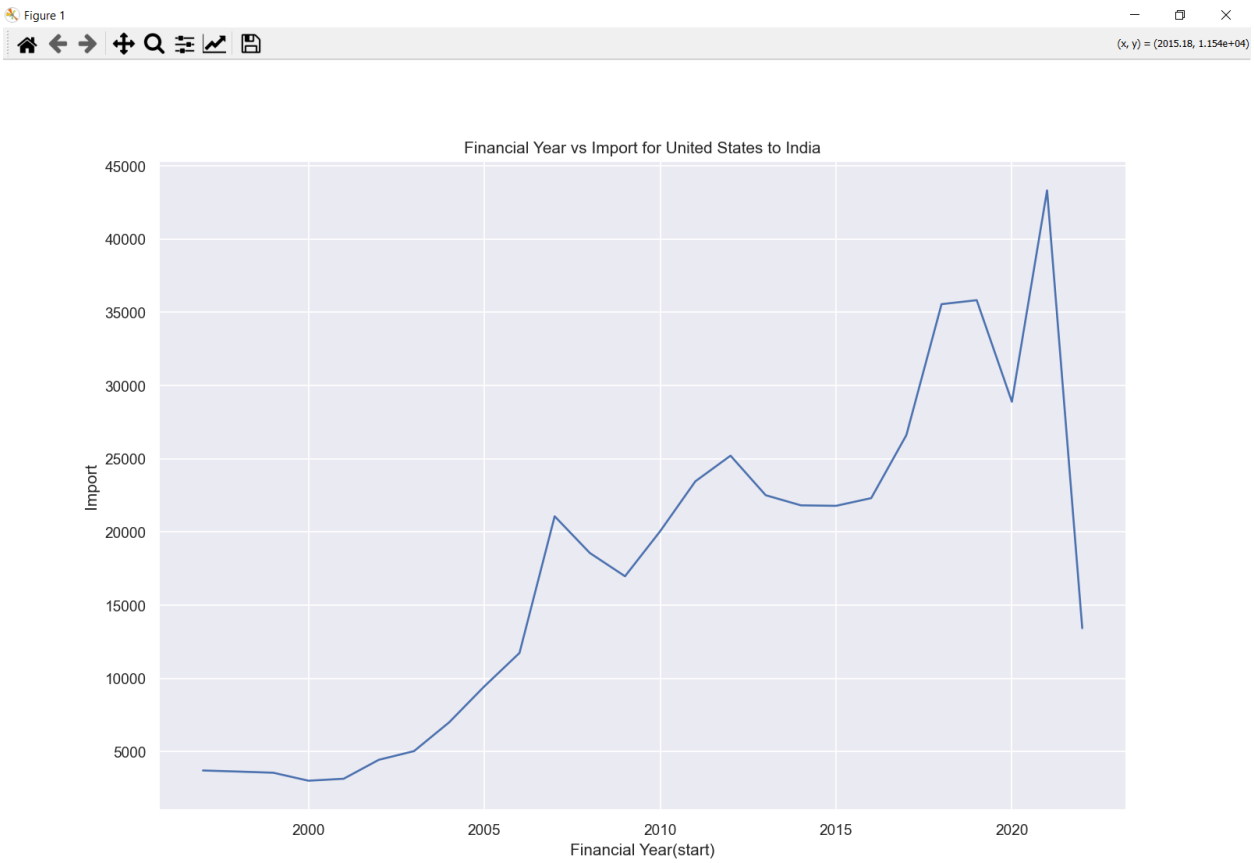


Figure 1
(x, y) = (2015.18, 1.154e+04)
Financial Year vs Import for United States to India
Import
45000
40000
35000
30000
25000
20000
15000
10000
5000
2000
2005
2010
2015
2020
Financial Year(start)

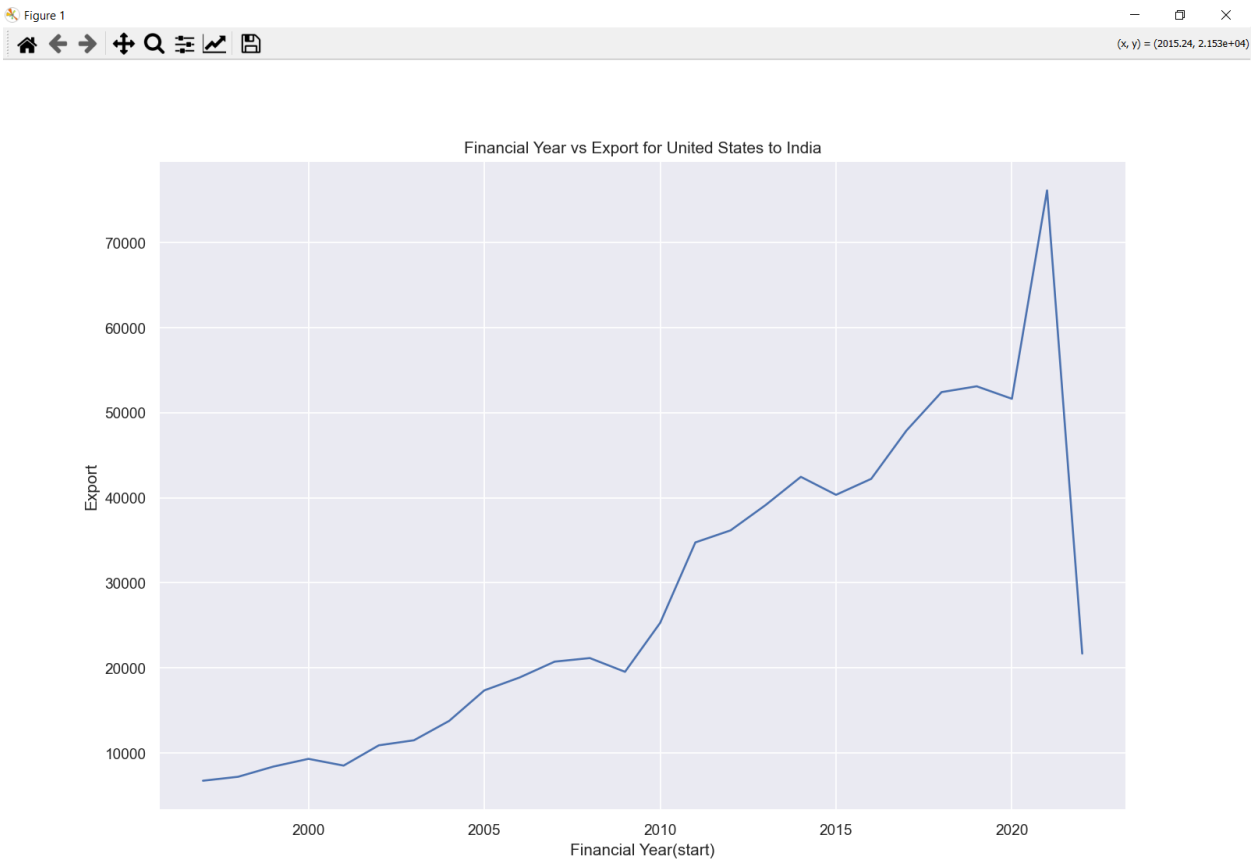


Figure 1
(x, y) = (2015.24, 2.153e+04)
Financial Year vs Export for United States to India
Export
70000
60000
50000
40000
30000
20000
10000
2000
2005
2010
2015
2020
Financial Year(start)

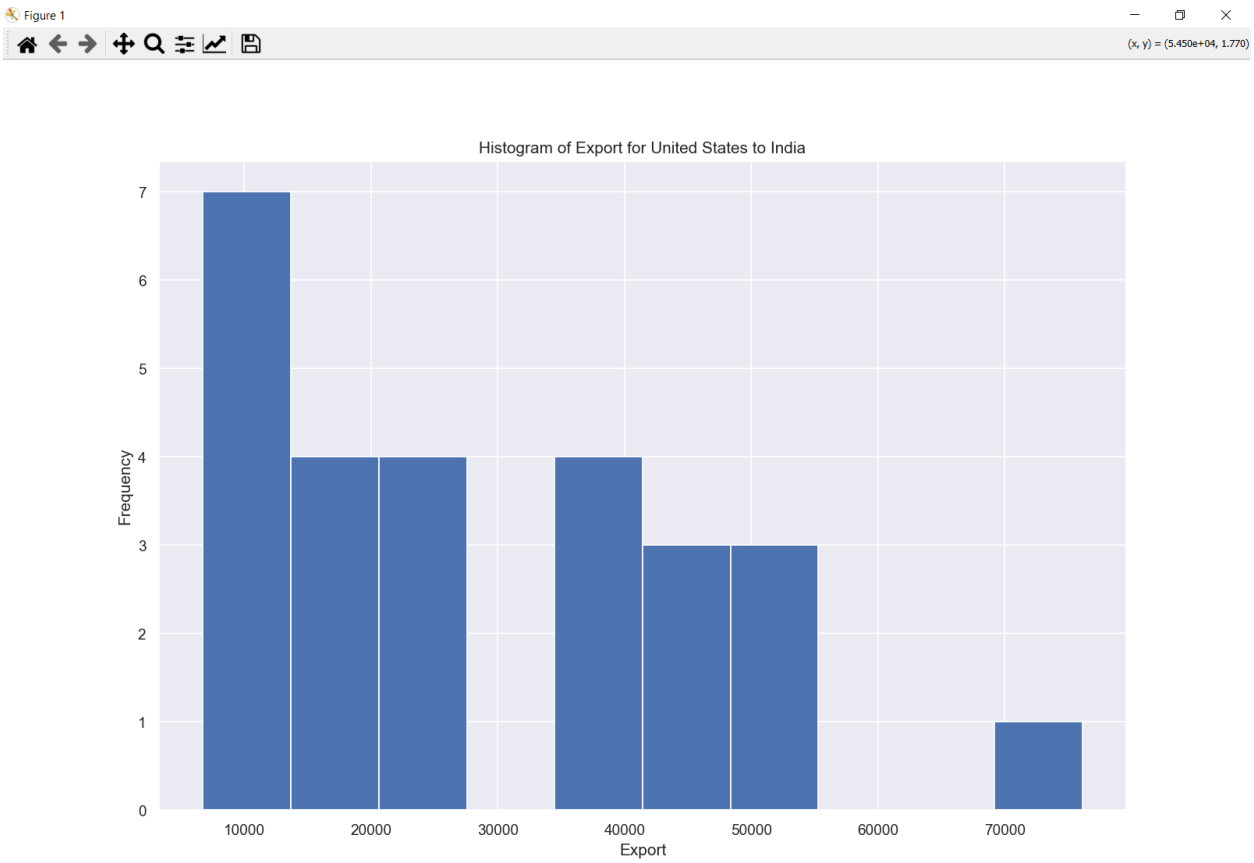


Figure 1
(x, y) = (5.450e+04, 1.770)
Histogram of Export for United States to India
Frequency
0
1
2
3
4
5
6
7
10000
20000
30000
40000
50000
60000
70000
Export

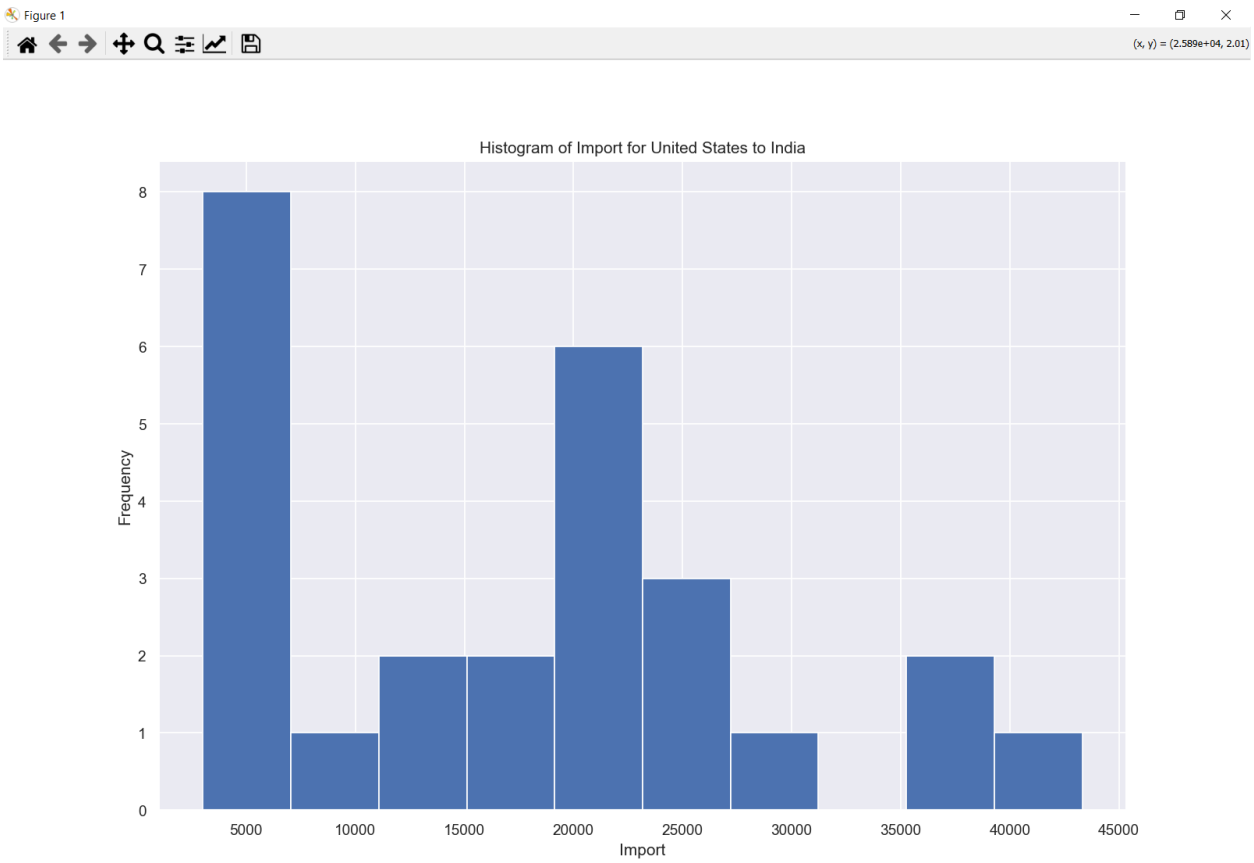


Figure 1
(x, y) = (2.589e+04, 2.01)
Histogram of Import for United States to India
Frequency
Import
0
1
2
3
4
5
6
7
8
5000
10000
15000
20000
25000
30000
35000
40000
45000

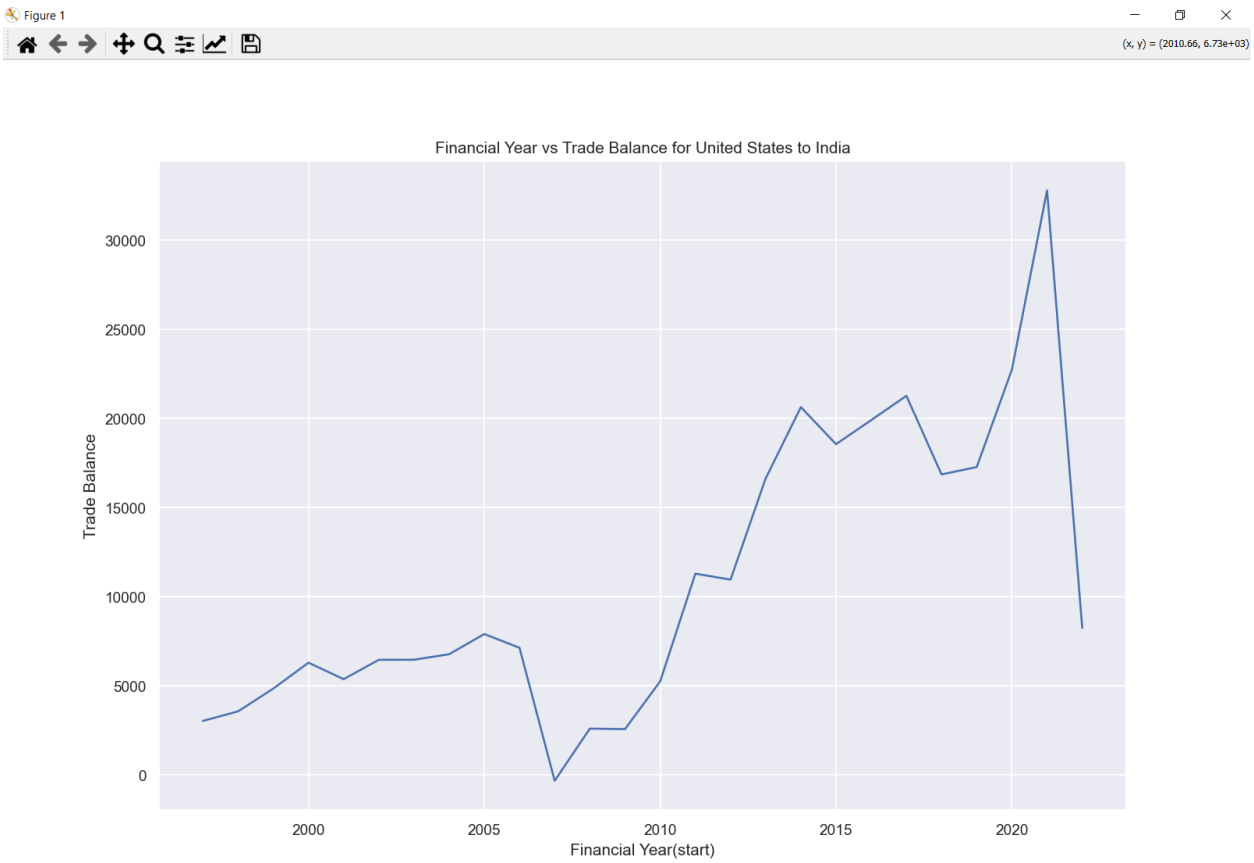
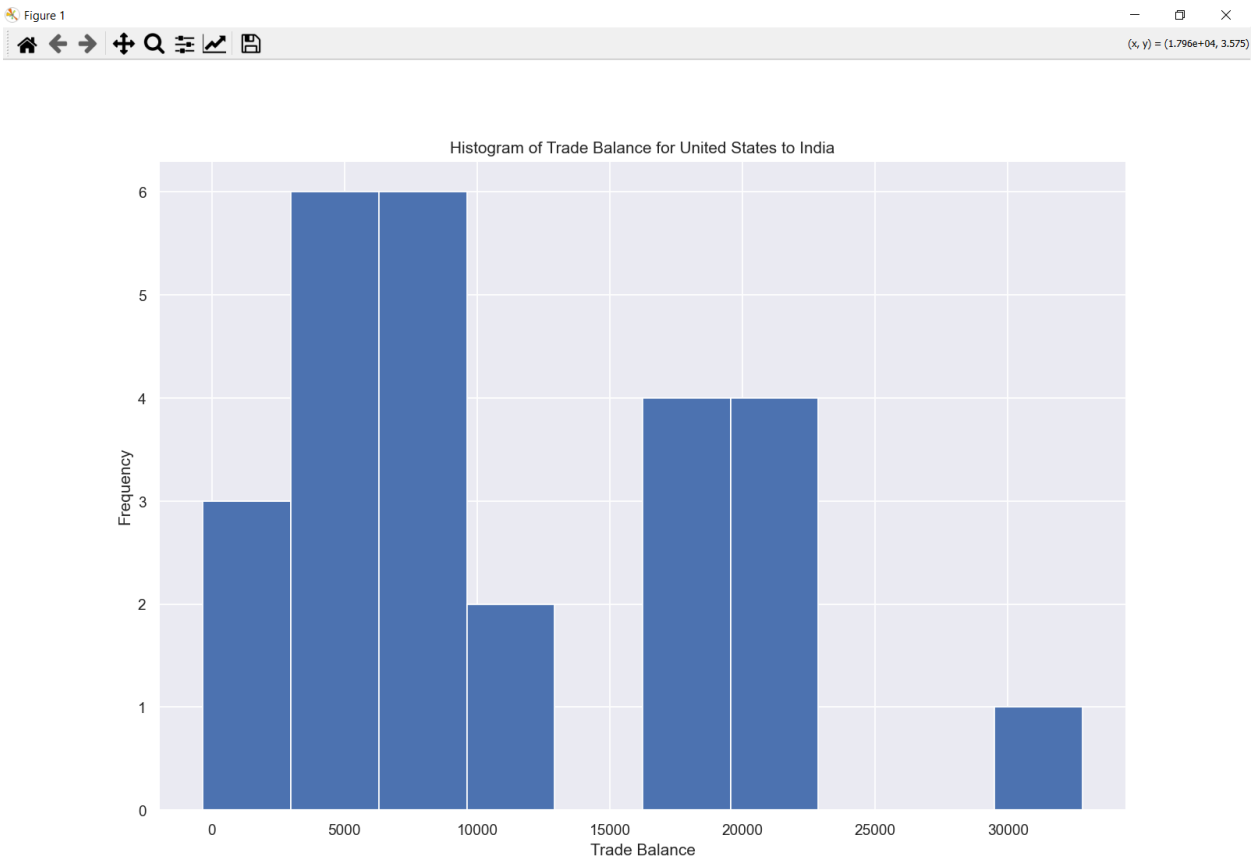


Figure 1
(x, y) = (2010.66, 6.73e+03)
Financial Year vs Trade Balance for United States to India
Trade Balance
30000
25000
20000
15000
10000
5000
0
2000
2005
2010
2015
2020
Financial Year(start)



After the cleanup by the Python code, all the histogram, XY charts look tightly bound.

(e) Write Python code to find, report, and correct any values in the dataset(s) that might be correct but in the wrong format. For example, suppose you have a variable called "State" and the values can be state abbreviations like FL, OR, CA, etc. However, one of the entries is Fla. We know that this is FL and it needs to be updated to be the right (expected) format as prescribed by the dataset. Again, there are an infinite number of possibilities because all datasets are different. Take your time, explore your data, and determine how best to clean it.

Before

```
Data loaded successfully
       Country  Export  Import  Total Trade  Trade Balance  \
0   AFGHANISTAN   21.25   10.70        31.95          10.55
1   AFGHANISTAN   12.81   28.14        40.95         -15.33
2   AFGHANISTAN   33.20   21.06        54.26          12.15
3   AFGHANISTAN   25.86   26.59        52.45          -0.73
4   AFGHANISTAN   24.37   17.52        41.89           6.85
5   AFGHANISTAN   60.77   18.46        79.23          42.31
6   AFGHANISTAN  145.47   40.51       185.98         104.96
7   AFGHANISTAN  165.44   47.01       212.44         118.43
8   AFGHANISTAN  142.67   58.42       201.09          84.24
9   AFGHANISTAN  182.11   34.37       216.48         147.73
10  AFGHANISTAN  249.21  109.97       359.18         139.24
11  AFGHANISTAN  394.23  126.24       520.47         268.00
12  AFGHANISTAN  463.55  125.19       588.74         338.36
13  AFGHANISTAN  422.41  146.03       568.44         276.38
14  AFGHANISTAN  510.90  132.50       643.41         378.40
15  AFGHANISTAN  472.63  159.55       632.18         313.07
16  AFGHANISTAN  474.34  208.77       683.10         265.57
17  AFGHANISTAN  422.56  261.91       684.47         160.65
18  AFGHANISTAN  526.60  307.90       834.50         218.70
19  AFGHANISTAN  506.34  292.90       799.24         213.44
```

	A	B	C	D	E	F	G	H	I	J
136	ANGOLA	25.02	0.01	25.03	25.01	2001	2002			
137	ANGOLA	37.31	7.2	44.51	30.12	2002	2003			
138	ANGOLA	70.55				2003	2004			
139	ANGOLA	72.89	0.91	73.8	71.99	2004	2005			
140	ANGOLA	151.66	3.25	154.91	148.41	2005	2006			
141	ANGOLA	201.89	244.71	446.6	-42.83	2006	2007			
142	ANGOLA	261.47	1,024.74	1,286.21	-763.27	2007	2008			
143	ANGOLA	370.45	1,386.25	1,756.70	-1,015.80	2008	2009			
144	ANGOLA	635.07	4,242.79	4,877.85	-3,607.72	2009	2010			
145	ANGOLA	675.44	5,112.12	5,787.56	-4,436.68	2010	2011			
146	ANGOLA	454.33	6,625.07	7,079.40	-6,170.75	2011	2012			
147	ANGOLA	488.79	7,157.54	7,646.33	-6,668.75	2012	2013			
148	ANGOLA	536.03	5,992.31	6,528.34	-5,456.28	2013	2014			
149	ANGOLA	552.64	4,617.64	5,170.29	-4,065.00	2014	2015			
150	ANGOLA	223.19	2,766.81	2,990.00	-2,543.62	2015	2016			
151	ANGOLA	154.63	2,596.49	2,751.12	-2,441.86	2016	2017			
152	ANGOLA	234.92	4,323.85	4,558.77	-4,088.93	2017	2018			
153	ANGOLA	282.36	4,027.49	4,309.86	-3,745.13	2018	2019			
154	ANGOLA	285.1	3,649.02	3,934.11	-3,363.92	2019	2020			
155	ANGOLA	259.6	1,879.74	2,139.34	-1,620.14	2020	2021			
156	ANGOLA	452.45	2,725.08	3,177.53	-2,272.63	2021	2022			
157	ANGOLA	128.68	941.83	1,070.51	-813.15	2022	till now			
158	ANGUILLA	0.47				2003	2004			
159	ANGUILLA	0.27				2004	2005			
160	ANGUILLA	0.12				2005	2006			
161	ANGUILLA	0.87				2006	2007			
162	ANGUILLA	0.18				2007	2008			
163	ANGUILLA	0.08				2008	2009			
164	ANGUILLA	0.05				2009	2010			

exports and imports of india(19

The commas in the numbers are in a wrong format and made them into a string instead of a float. They need to be removed.

Some countries have very little trade and show many missing values.

The decimal places were rounded to 1 place.

After

	Country	Export	Import	Total Trade	Trade Balance	\
0	AFGHANISTAN	21.2	10.7	32.0	10.6	
1	AFGHANISTAN	12.8	28.1	41.0	-15.3	
2	AFGHANISTAN	33.2	21.1	54.3	12.2	
3	AFGHANISTAN	25.9	26.6	52.4	-0.7	
4	AFGHANISTAN	24.4	17.5	41.9	6.8	
5	AFGHANISTAN	60.8	18.5	79.2	42.3	
6	AFGHANISTAN	145.5	40.5	186.0	105.0	
7	AFGHANISTAN	165.4	47.0	212.4	118.4	
8	AFGHANISTAN	142.7	58.4	201.1	84.2	
9	AFGHANISTAN	182.1	34.4	216.5	147.7	
10	AFGHANISTAN	249.2	110.0	359.2	139.2	
11	AFGHANISTAN	394.2	126.2	520.5	268.0	
12	AFGHANISTAN	463.6	125.2	588.7	338.4	
13	AFGHANISTAN	422.4	146.0	568.4	276.4	
14	AFGHANISTAN	510.9	132.5	643.4	378.4	
15	AFGHANISTAN	472.6	159.6	632.2	313.1	
16	AFGHANISTAN	474.3	208.8	683.1	265.6	
17	AFGHANISTAN	422.6	261.9	684.5	160.6	
18	AFGHANISTAN	526.6	307.9	834.5	218.7	
19	AFGHANISTAN	506.3	292.9	799.2	213.4	

The Python code changed the format for these floating numbers to 1 decimal places for easier reading.

All the commas were removed making them floating numbers.

(f) Write Python code to find, visualize, and correct any outliers.

Before

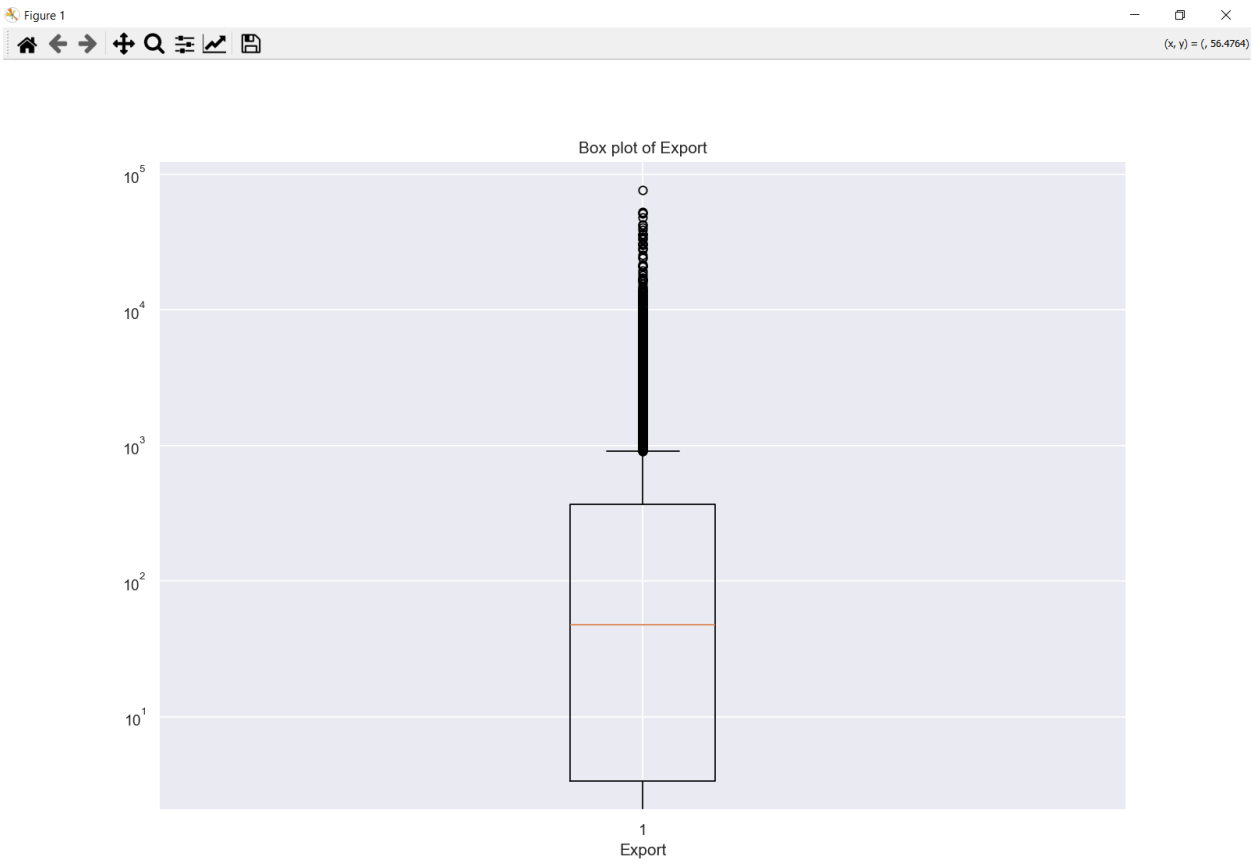


Figure 1
(x, y) = (, 56.4764)
Box plot of Export
10^5
10^4
10^3
10^2
10^1
1
Export

log scale

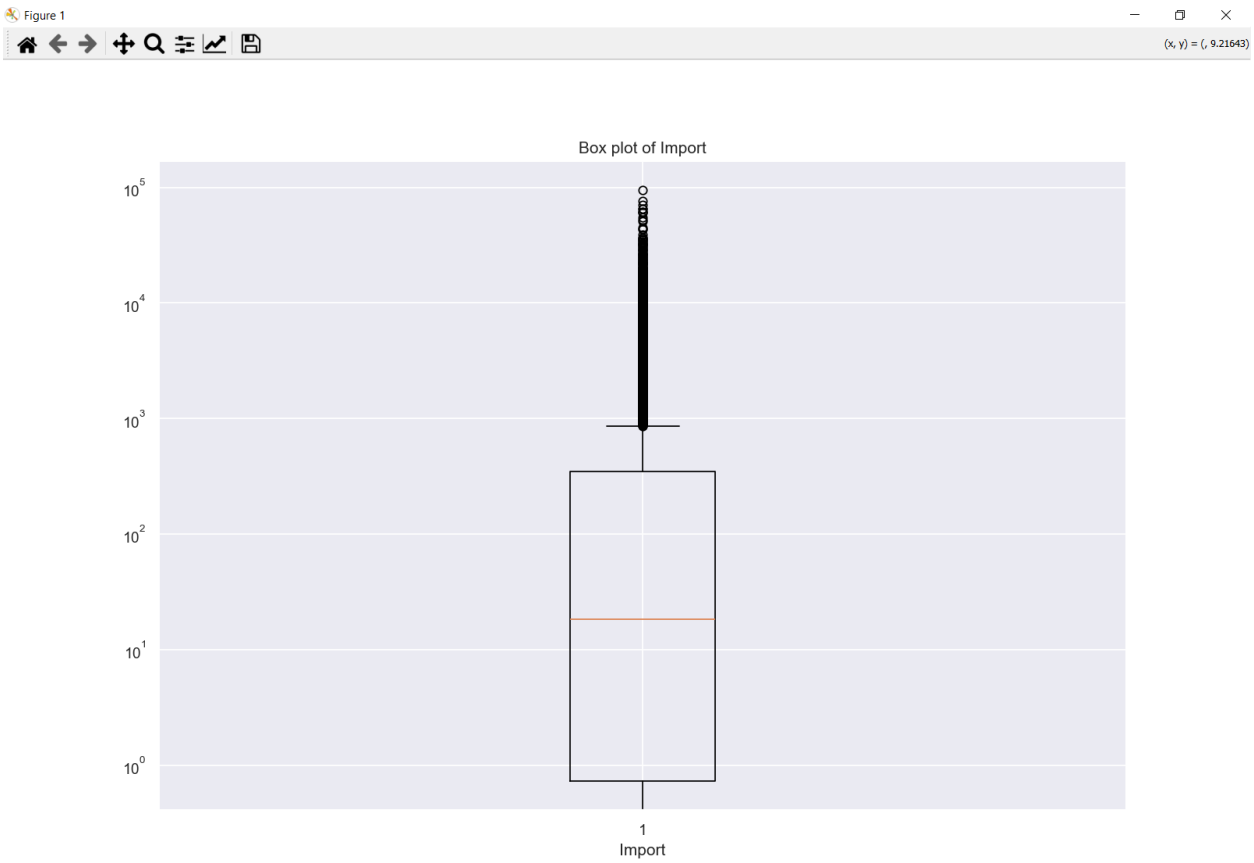


Figure 1
(x, y) = (, 9.21643)
Box plot of Import
10^5
10^4
10^3
10^2
10^1
10^0
1
Import

log scale

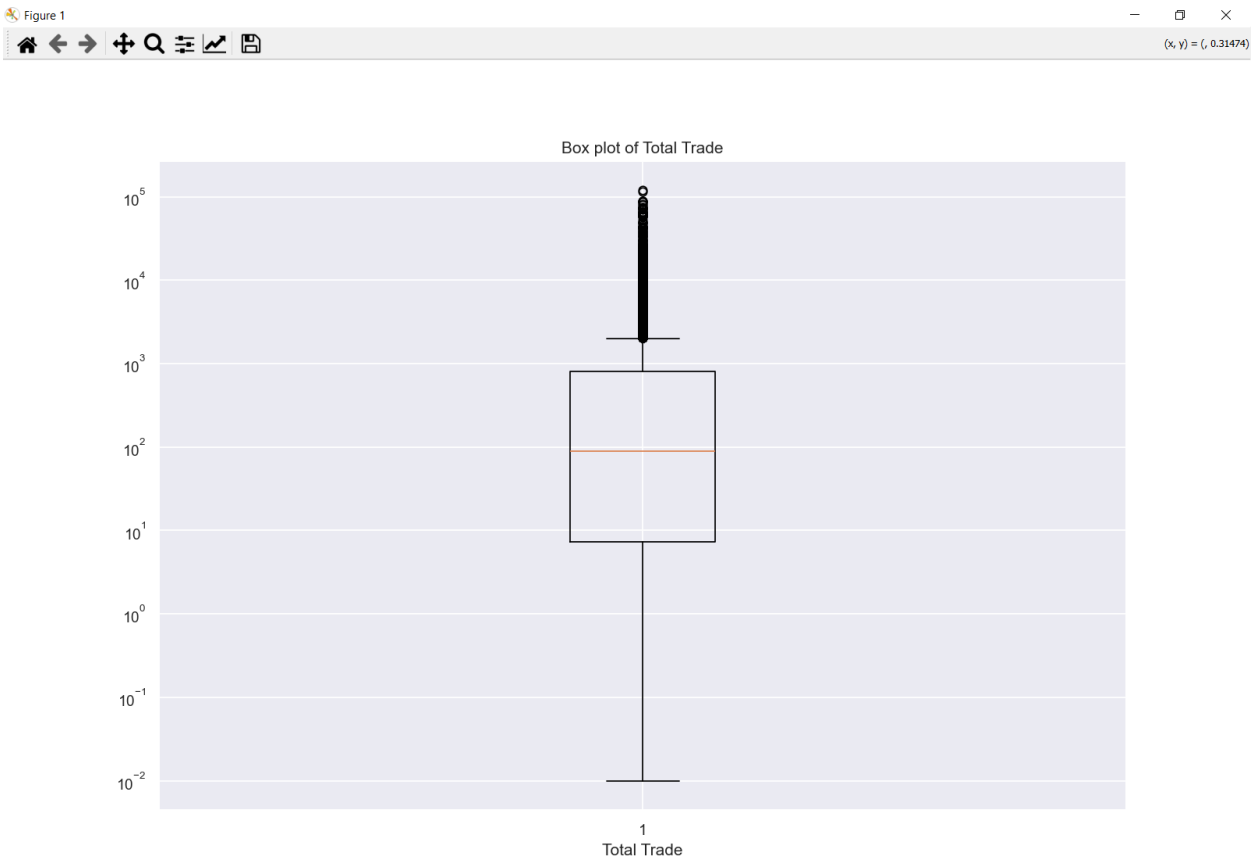


Figure 1
(x, y) = (, 0.31474)
Box plot of Total Trade
10^{5}
10^{4}
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
1
Total Trade

log scale

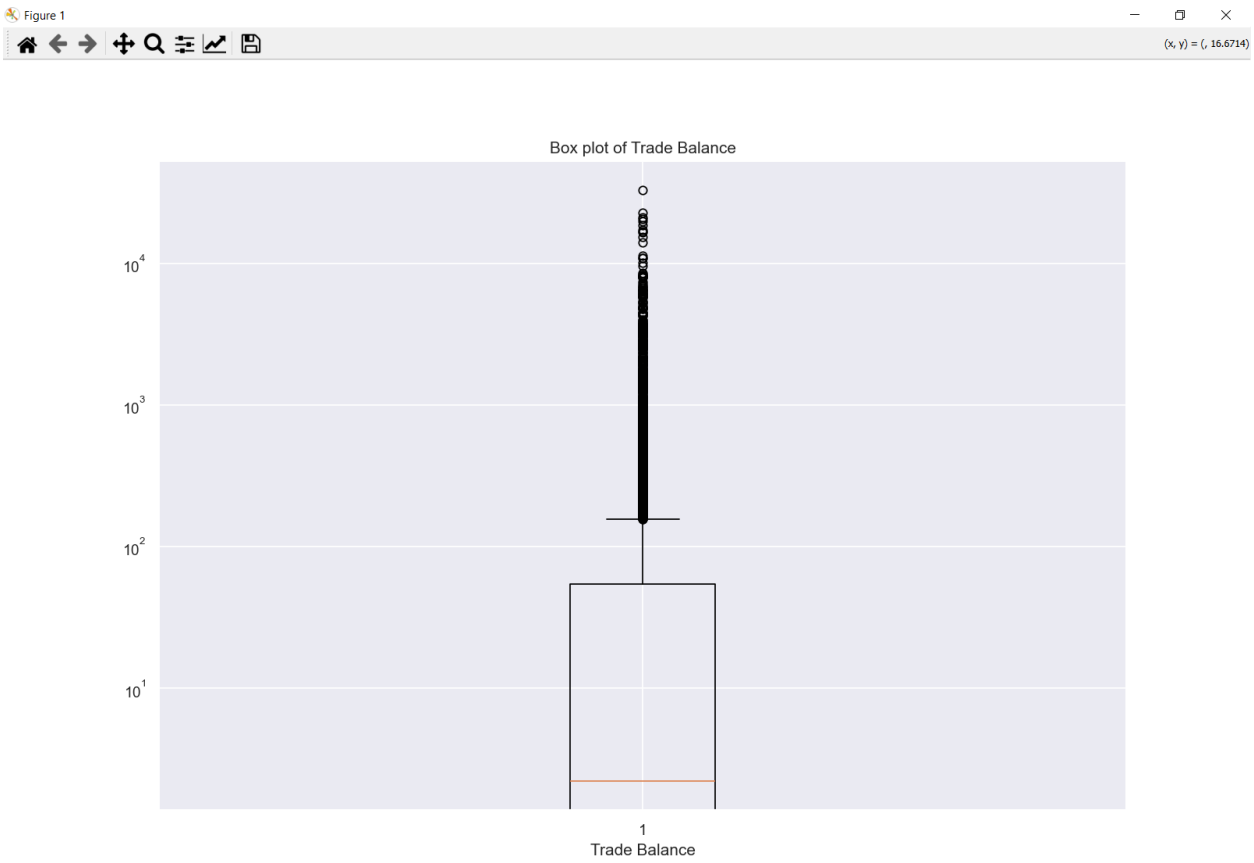


Figure 1
(x, y) = (, 16.6714)
Box plot of Trade Balance
10^4
10^3
10^2
10^1
1
Trade Balance

log scale

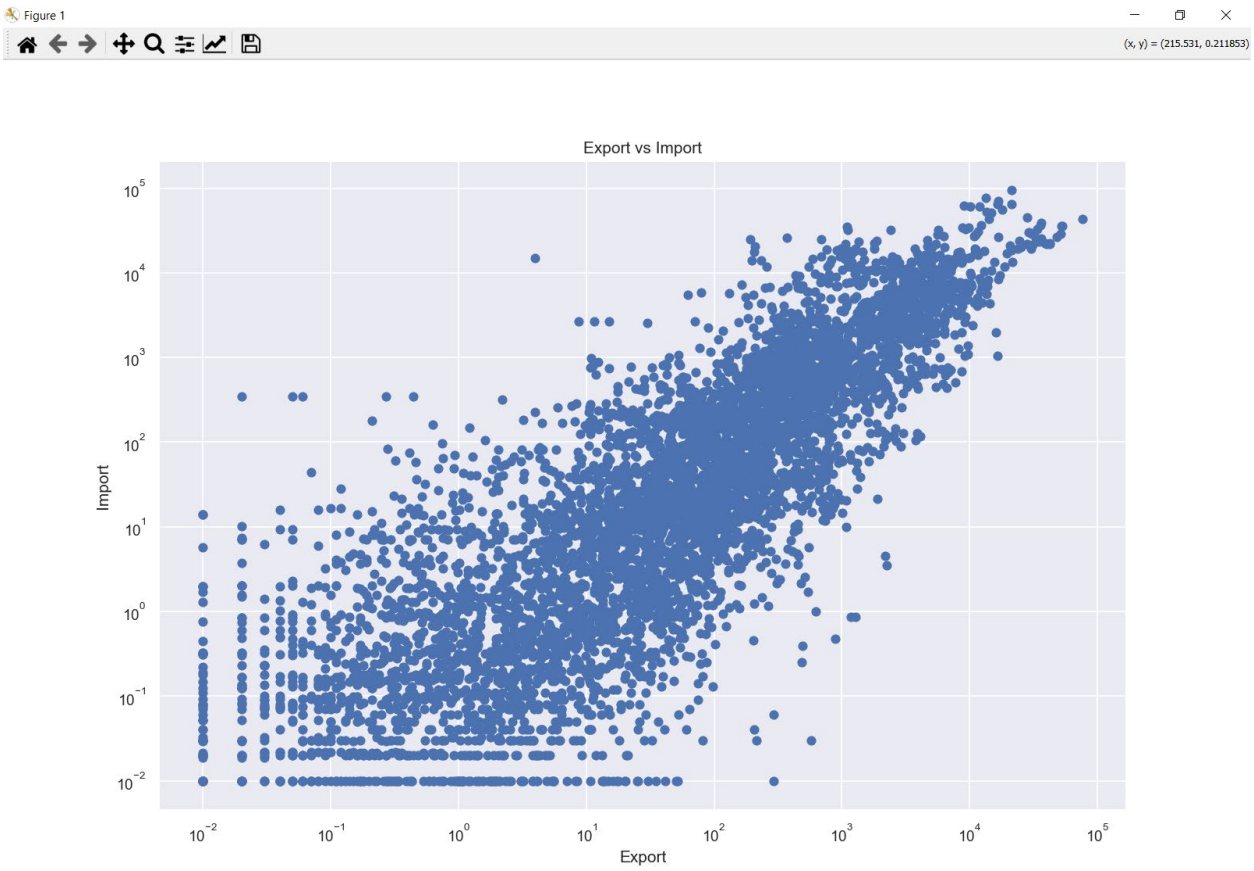
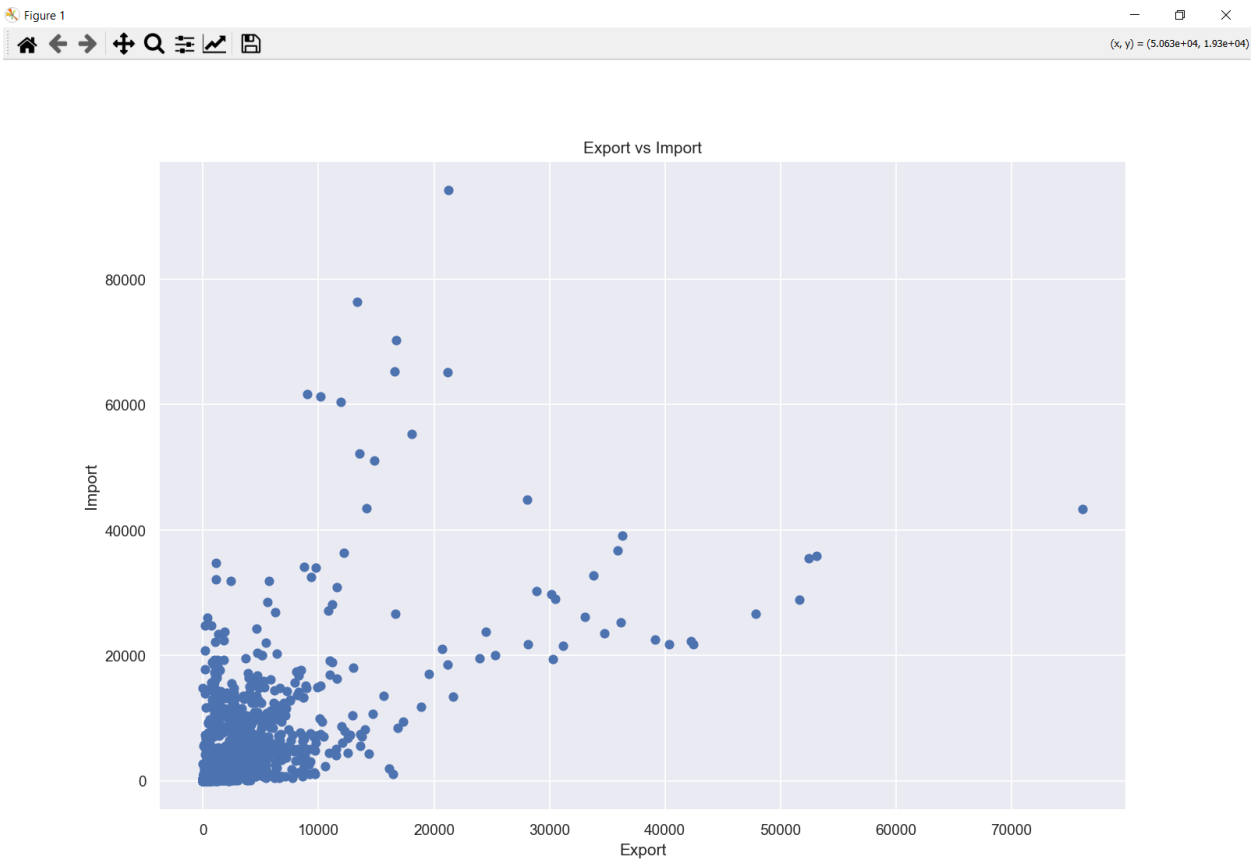


Figure 1
(x, y) = (215.531, 0.211853)
Export vs Import
Import
Export
10^{5}
10^{4}
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}

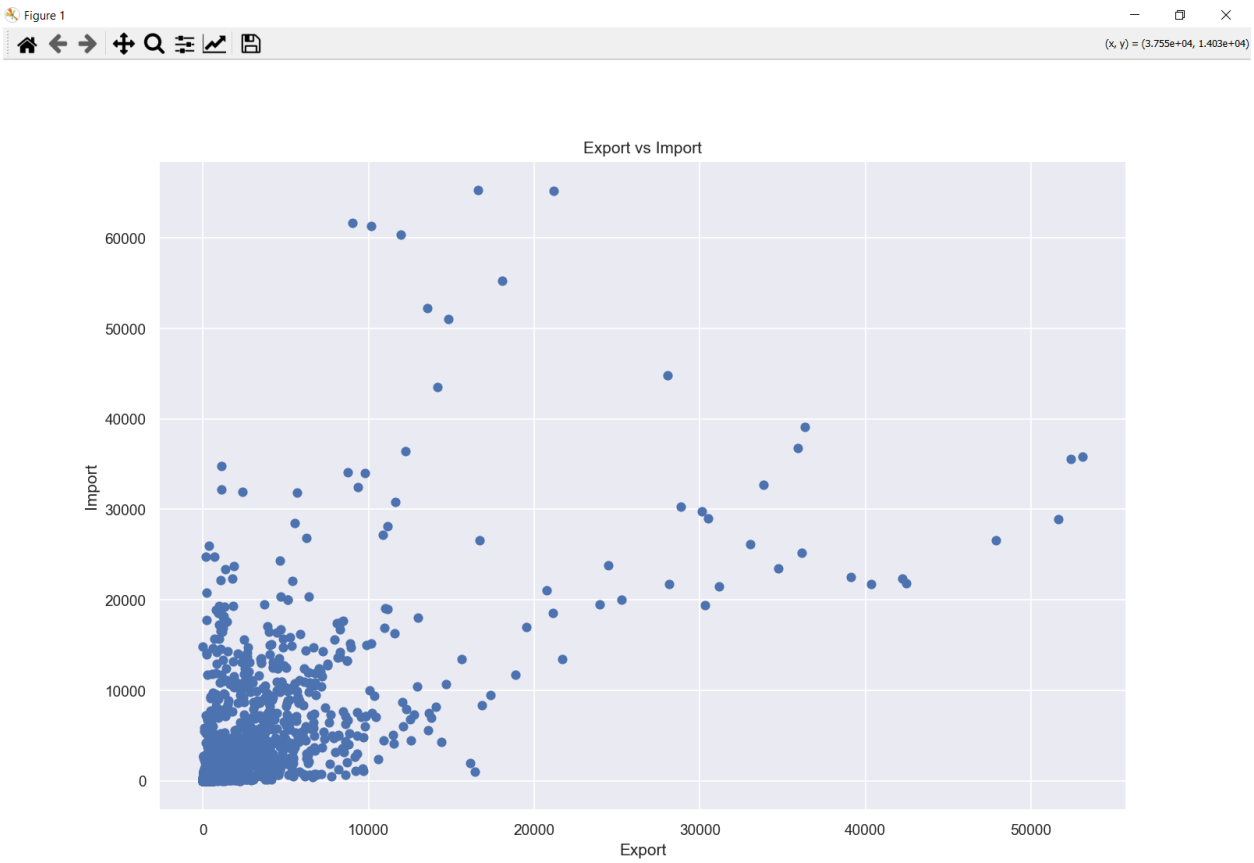
log scale



linear scale

Export values greater than 70000 look like outliers.

After



linear scale

The Python code analyzed Export and Import values greater than 70000 and were deleted.

(g) Write Python code to create a new dataframe from one of your datasets such that the new dataframe is normalized using min-max. Be sure to include a screen image of the normalized dataset in your report.

Before

```
Data loaded successfully
    Export  Import
0    21.25   10.70
1    12.81   28.14
2    33.20   21.06
3    25.86   26.59
4    24.37   17.52
5    60.77   18.46
6   145.47   40.51
7   165.44   47.01
8   142.67   58.42
9   182.11   34.37
10  249.21  109.97
11  394.23  126.24
12  463.55  125.19
13  422.41  146.03
14  510.90  132.50
```

Export and Import amounts are a good choice for min max normalization. They have a broad range.

After

```
new dataset normalized_Export and normalized_Import created
```

A new dataset name stores the normalized import and export data. It is easier to print and chart the minimized data.

```
      Export    Import
0   0.000400  0.000164
1   0.000241  0.000431
2   0.000625  0.000323
3   0.000487  0.000407
4   0.000459  0.000268
5   0.001145  0.000283
6   0.002740  0.000621
7   0.003116  0.000720
8   0.002687  0.000895
9   0.003430  0.000527
10  0.004694  0.001685
11  0.007426  0.001934
12  0.008732  0.001918
13  0.007957  0.002238
14  0.009624  0.002030
```

The Python code normalized the Export and Import data to a range of 0 to 1.

(h) Write Python code to create a new dataframe that contains only unlabeled and quantitative data. Be sure to include a screen image in your report.

Before

```
Data loaded successfully
       Country  Export  Import  Total Trade  Trade Balance  \
0   AFGHANISTAN   21.25   10.70        31.95          10.55
1   AFGHANISTAN   12.81   28.14        40.95         -15.33
2   AFGHANISTAN   33.20   21.06        54.26          12.15
3   AFGHANISTAN   25.86   26.59        52.45          -0.73
4   AFGHANISTAN   24.37   17.52        41.89           6.85
5   AFGHANISTAN   60.77   18.46        79.23          42.31
6   AFGHANISTAN  145.47   40.51       185.98         104.96
7   AFGHANISTAN  165.44   47.01       212.44         118.43
8   AFGHANISTAN  142.67   58.42       201.09          84.24
9   AFGHANISTAN  182.11   34.37       216.48         147.73
10  AFGHANISTAN  249.21  109.97       359.18         139.24
11  AFGHANISTAN  394.23  126.24       520.47         268.00
12  AFGHANISTAN  463.55  125.19       588.74         338.36
13  AFGHANISTAN  422.41  146.03       568.44         276.38
14  AFGHANISTAN  510.90  132.50       643.41         378.40
```

The column label Country is the only non numeric column.

After

```
new dataset MyData_quant created
```

The remaining numeric dataset was assigned a new name. Printing and charting can done using new dataset.

```
    Export  Import  Total Trade  Trade Balance  Financial Year(start)
0    21.25   10.70        31.95          10.55                   1997
1    12.81   28.14        40.95         -15.33                   1998
2    33.20   21.06        54.26          12.15                   1999
3    25.86   26.59        52.45          -0.73                   2000
4    24.37   17.52        41.89           6.85                   2001
5    60.77   18.46        79.23          42.31                   2002
6   145.47   40.51       185.98         104.96                   2003
7   165.44   47.01       212.44         118.43                   2004
8   142.67   58.42       201.09          84.24                   2005
9   182.11   34.37       216.48         147.73                   2006
10  249.21  109.97       359.18         139.24                   2007
11  394.23  126.24       520.47         268.00                   2008
12  463.55  125.19       588.74         338.36                   2009
13  422.41  146.03       568.44         276.38                   2010
14  510.90  132.50       643.41         378.40                   2011
Press any key to continue . . .
```

The Country column has been deleted because it was an unlabeled data. The new dataset is all numeric.

(i) Finally, take any further steps you feel are needed to clean up your data such as discretization and feature generation.

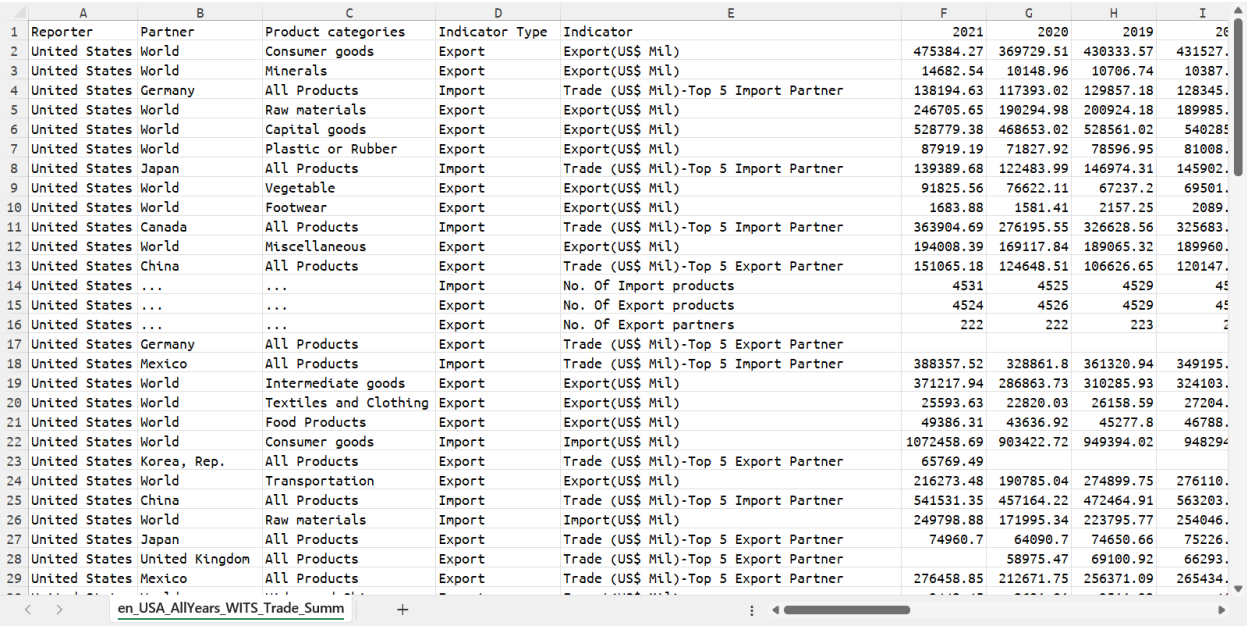
No more changes. All the previous steps has cleaned up all the anomalies. There were basic issues that needed to be taken care of.

Dataset 3 wits_en_trade_summary_allcountries_allyears

Dataset from International Trade Administration. This dataset has been harder to process. The data is not in columns but in the rows. It requires a different technique to extract sub data from rows.

(a) Write Python code to assure that your datasets are in record format so that they are structured as rows and columns, where each column has a variable name.

Before



	A	B	C	D	E	F	G	H	I
1	Reporter	Partner	Product categories	Indicator Type	Indicator	2021	2020	2019	26
2	United States	World	Consumer goods	Export	Export(US$ Mil)	475384.27	369729.51	430333.57	431527.
3	United States	World	Minerals	Export	Export(US$ Mil)	14682.54	10148.96	10706.74	10387.
4	United States	Germany	All Products	Import	Trade (US$ Mil)-Top 5 Import Partner	138194.63	117393.02	129857.18	128345.
5	United States	World	Raw materials	Export	Export(US$ Mil)	246705.65	190294.98	200924.18	189985.
6	United States	World	Capital goods	Export	Export(US$ Mil)	528779.38	468653.02	528561.02	540285
7	United States	World	Plastic or Rubber	Export	Export(US$ Mil)	87919.19	71827.92	78596.95	81008.
8	United States	Japan	All Products	Import	Trade (US$ Mil)-Top 5 Import Partner	139389.68	122483.99	146974.31	145902.
9	United States	World	Vegetable	Export	Export(US$ Mil)	91825.56	76622.11	67237.2	69501.
10	United States	World	Footwear	Export	Export(US$ Mil)	1683.88	1581.41	2157.25	2089.
11	United States	Canada	All Products	Import	Trade (US$ Mil)-Top 5 Import Partner	363904.69	276195.55	326628.56	325683.
12	United States	World	Miscellaneous	Export	Export(US$ Mil)	194008.39	169117.84	189065.32	189960.
13	United States	China	All Products	Export	Trade (US$ Mil)-Top 5 Export Partner	151065.18	124648.51	106626.65	120147.
14	United States	...	...	Import	No. Of Import products	4531	4525	4529	45
15	United States	...	...	Export	No. Of Export products	4524	4526	4529	45
16	United States	...	...	Export	No. Of Export partners	222	222	223	2
17	United States	Germany	All Products	Export	Trade (US$ Mil)-Top 5 Export Partner				
18	United States	Mexico	All Products	Import	Trade (US$ Mil)-Top 5 Import Partner	388357.52	328861.8	361320.94	349195.
19	United States	World	Intermediate goods	Export	Export(US$ Mil)	371217.94	286863.73	310285.93	324103.
20	United States	World	Textiles and Clothing	Export	Export(US$ Mil)	25593.63	22820.03	26158.59	27204.
21	United States	World	Food Products	Export	Export(US$ Mil)	49386.31	43636.92	45277.8	46788.
22	United States	World	Consumer goods	Import	Import(US$ Mil)	1072458.69	903422.72	949394.02	948294
23	United States	Korea, Rep.	All Products	Export	Trade (US$ Mil)-Top 5 Export Partner	65769.49			
24	United States	World	Transportation	Export	Export(US$ Mil)	216273.48	190785.04	274899.75	276110.
25	United States	China	All Products	Import	Trade (US$ Mil)-Top 5 Import Partner	541531.35	457164.22	472464.91	563203.
26	United States	World	Raw materials	Import	Import(US$ Mil)	249798.88	171995.34	223795.77	254046.
27	United States	Japan	All Products	Export	Trade (US$ Mil)-Top 5 Export Partner	74960.7	64090.7	74650.66	75226.
28	United States	United Kingdom	All Products	Export	Trade (US$ Mil)-Top 5 Export Partner		58975.47	69100.92	66293.
29	United States	Mexico	All Products	Export	Trade (US$ Mil)-Top 5 Export Partner	276458.85	212671.75	256371.09	265434.

en_USA_AllYears_WITS_Trade_Summ

Big csv file.

After

```
Data loaded successfully
<class 'pandas.core.frame.DataFrame'>
        Reporter  Partner Product categories Indicator Type  \
0   United States    World     Consumer goods         Export
1   United States    World           Minerals         Export
2   United States  Germany       All Products         Import
3   United States    World      Raw materials         Export
4   United States    World      Capital goods         Export
..            ...      ...                ...            ...
95  United States    Japan       All Products         Import
96  United States      ...                ...    Development
97  United States    World     Consumer goods         Export
98  United States    World             Animal         Export
99  United States    World      Raw materials         Import

                             Indicator       2021       2020       2019  \
0                      Export(US$ Mil)  475384.27  369729.51  430333.57
1                      Export(US$ Mil)   14682.54   10148.96   10706.74
2    Trade (US$ Mil)-Top 5 Import Partner  138194.63  117393.02  129857.18
3                      Export(US$ Mil)  246705.65  190294.98  200924.18
4                      Export(US$ Mil)  528779.38  468653.02  528561.02
..                                 ...        ...        ...        ...
95  Partner share(%)-Top 5 Import Partner       4.75       5.09       5.72
96                     Trade (% of GDP)        NaN        NaN      26.31
97               Export Product share(%)      27.12      25.85      26.17
98               Export Product share(%)       2.05       2.06       1.80
99               Import Product share(%)       8.52       7.15       8.72
```

The Python code read in the csv file and Panda converted it into a dataframe.

(b) Write Python code to check and print the data types of the variables in your dataset. Write code to correct any data types. For example, if Python reads in a categorical variable as a number, you will need to update this to a category.

Before

```
Data loaded successfully
<class 'pandas.core.frame.DataFrame'>
Reporter               0
Partner                0
Product categories     0
Indicator Type         0
Indicator              0
2021                  12
2020                  10
2019                   6
2018                   6
2017                   6
2016                   6
2015                   6
2014                   6
2013                   6
2012                   6
2011                   6
2010                   6
2009                   6
2008                   6
2007                   6
2006                   6
2005                   6
2004                   6
2003                   6
2002                   6
2001                   6
2000                   6
1999                   6
1998                   6
1997                   6
1996                   6
1995                   6
1994                  16
1993                   6
1992                   6
1991                   6
1990                  94
1989                  94
1988                  94
dtype: int64
```

The column years are numbers but they have missing values. The data in this dataset is in the rows not the columns.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 39 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Reporter            100 non-null    object
 1   Partner             100 non-null    object
 2   Product categories  100 non-null    object
 3   Indicator Type      100 non-null    object
 4   Indicator           100 non-null    object
 5   2021                88 non-null     float64
 6   2020                90 non-null     float64
 7   2019                94 non-null     float64
 8   2018                94 non-null     float64
 9   2017                94 non-null     float64
 10  2016                94 non-null     float64
 11  2015                94 non-null     float64
 12  2014                94 non-null     float64
 13  2013                94 non-null     float64
 14  2012                94 non-null     float64
 15  2011                94 non-null     float64
 16  2010                94 non-null     float64
 17  2009                94 non-null     float64
 18  2008                94 non-null     float64
 19  2007                94 non-null     float64
 20  2006                94 non-null     float64
 21  2005                94 non-null     float64
 22  2004                94 non-null     float64
 23  2003                94 non-null     float64
 24  2002                94 non-null     float64
 25  2001                94 non-null     float64
 26  2000                94 non-null     float64
 27  1999                94 non-null     float64
 28  1998                94 non-null     float64
 29  1997                94 non-null     float64
 30  1996                94 non-null     float64
 31  1995                94 non-null     float64
 32  1994                84 non-null     float64
 33  1993                94 non-null     float64
 34  1992                94 non-null     float64
 35  1991                94 non-null     float64
 36  1990                6 non-null      float64
 37  1989                6 non-null      float64
 38  1988                6 non-null      float64
dtypes: float64(34), object(5)
memory usage: 30.6+ KB
None
```

There are 100 rows but many columns are missing values.

After

```
Reporter                0
Partner                 0
Product categories      0
Indicator Type          0
Indicator               0
2021                    0
2020                    0
2019                    0
2018                    0
2017                    0
2016                    0
2015                    4
2014                    4
2013                    4
2012                    4
2011                    4
2010                    4
2009                    4
2008                    4
2007                    4
2006                    4
2005                    4
2004                    4
2003                    4
2002                    4
2001                    4
2000                    4
1999                    4
1998                    4
1997                    4
1996                    4
1995                    4
1994                   14
1993                    4
1992                    4
1991                    6
dtype: int64
```

The Python code only cleaned up the years 2016 to 2021 because the analyst does not need this data. We are primarily looking at the United states which the Python code has cleaned up already in the row data.

```
<class 'pandas.core.frame.DataFrame'>
Index: 98 entries, 0 to 99
Data columns (total 36 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Reporter            98 non-null     object
 1   Partner             98 non-null     object
 2   Product categories  98 non-null     object
 3   Indicator Type      98 non-null     object
 4   Indicator           98 non-null     object
 5   2021                98 non-null     int32
 6   2020                98 non-null     float64
 7   2019                98 non-null     float64
 8   2018                98 non-null     float64
 9   2017                98 non-null     float64
 10  2016                98 non-null     float64
 11  2015                94 non-null     float64
 12  2014                94 non-null     float64
 13  2013                94 non-null     float64
 14  2012                94 non-null     float64
 15  2011                94 non-null     float64
 16  2010                94 non-null     float64
 17  2009                94 non-null     float64
 18  2008                94 non-null     float64
 19  2007                94 non-null     float64
 20  2006                94 non-null     float64
 21  2005                94 non-null     float64
 22  2004                94 non-null     float64
 23  2003                94 non-null     float64
 24  2002                94 non-null     float64
 25  2001                94 non-null     float64
 26  2000                94 non-null     float64
 27  1999                94 non-null     float64
 28  1998                94 non-null     float64
 29  1997                94 non-null     float64
 30  1996                94 non-null     float64
 31  1995                94 non-null     float64
 32  1994                84 non-null     float64
 33  1993                94 non-null     float64
 34  1992                94 non-null     float64
 35  1991                92 non-null     float64
dtypes: float64(30), int32(1), object(5)
memory usage: 27.9+ KB
None
```

All the datatypes are correct now. I did make a change from float to integer for 2021 just to show this question has been fulfilled. In order to change datatypes two rows were deleted because they still had missing values which have a Null datatype. The column datatype cannot be changed if it has mixed

datatypes. So, the two Null rows were deleted. Now the entire column datatype can be changed to integer for demonstration.

(c) Write Python code to find, count, report, and then clean any missing values.

Before

```
Data loaded successfully
<class 'pandas.core.frame.DataFrame'>
Reporter               0
Partner                0
Product categories     0
Indicator Type         0
Indicator              0
2021                  12
2020                  10
2019                   6
2018                   6
2017                   6
2016                   6
2015                   6
2014                   6
2013                   6
2012                   6
2011                   6
2010                   6
2009                   6
2008                   6
2007                   6
2006                   6
2005                   6
2004                   6
2003                   6
2002                   6
2001                   6
2000                   6
1999                   6
1998                   6
1997                   6
1996                   6
1995                   6
1994                  16
1993                   6
1992                   6
1991                   6
1990                  94
1989                  94
1988                  94
dtype: int64
```

There are some missing values. We need to do a clean up.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 39 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Reporter            100 non-null    object
 1   Partner             100 non-null    object
 2   Product categories  100 non-null    object
 3   Indicator Type      100 non-null    object
 4   Indicator           100 non-null    object
 5   2021                88 non-null     float64
 6   2020                90 non-null     float64
 7   2019                94 non-null     float64
 8   2018                94 non-null     float64
 9   2017                94 non-null     float64
 10  2016                94 non-null     float64
 11  2015                94 non-null     float64
 12  2014                94 non-null     float64
 13  2013                94 non-null     float64
 14  2012                94 non-null     float64
 15  2011                94 non-null     float64
 16  2010                94 non-null     float64
 17  2009                94 non-null     float64
 18  2008                94 non-null     float64
 19  2007                94 non-null     float64
 20  2006                94 non-null     float64
 21  2005                94 non-null     float64
 22  2004                94 non-null     float64
 23  2003                94 non-null     float64
 24  2002                94 non-null     float64
 25  2001                94 non-null     float64
 26  2000                94 non-null     float64
 27  1999                94 non-null     float64
 28  1998                94 non-null     float64
 29  1997                94 non-null     float64
 30  1996                94 non-null     float64
 31  1995                94 non-null     float64
 32  1994                84 non-null     float64
 33  1993                94 non-null     float64
 34  1992                94 non-null     float64
 35  1991                94 non-null     float64
 36  1990                6 non-null      float64
 37  1989                6 non-null      float64
 38  1988                6 non-null      float64
dtypes: float64(34), object(5)
memory usage: 30.6+ KB
None
```

After

```
Reporter                0
Partner                 0
Product categories      0
Indicator Type          0
Indicator               0
2021                    0
2020                    0
2019                    0
2018                    0
2017                    0
2016                    0
2015                    4
2014                    4
2013                    4
2012                    4
2011                    4
2010                    4
2009                    4
2008                    4
2007                    4
2006                    4
2005                    4
2004                    4
2003                    4
2002                    4
2001                    4
2000                    4
1999                    4
1998                    4
1997                    4
1996                    4
1995                    4
1994                   14
1993                    4
1992                    4
1991                    6
dtype: int64
```

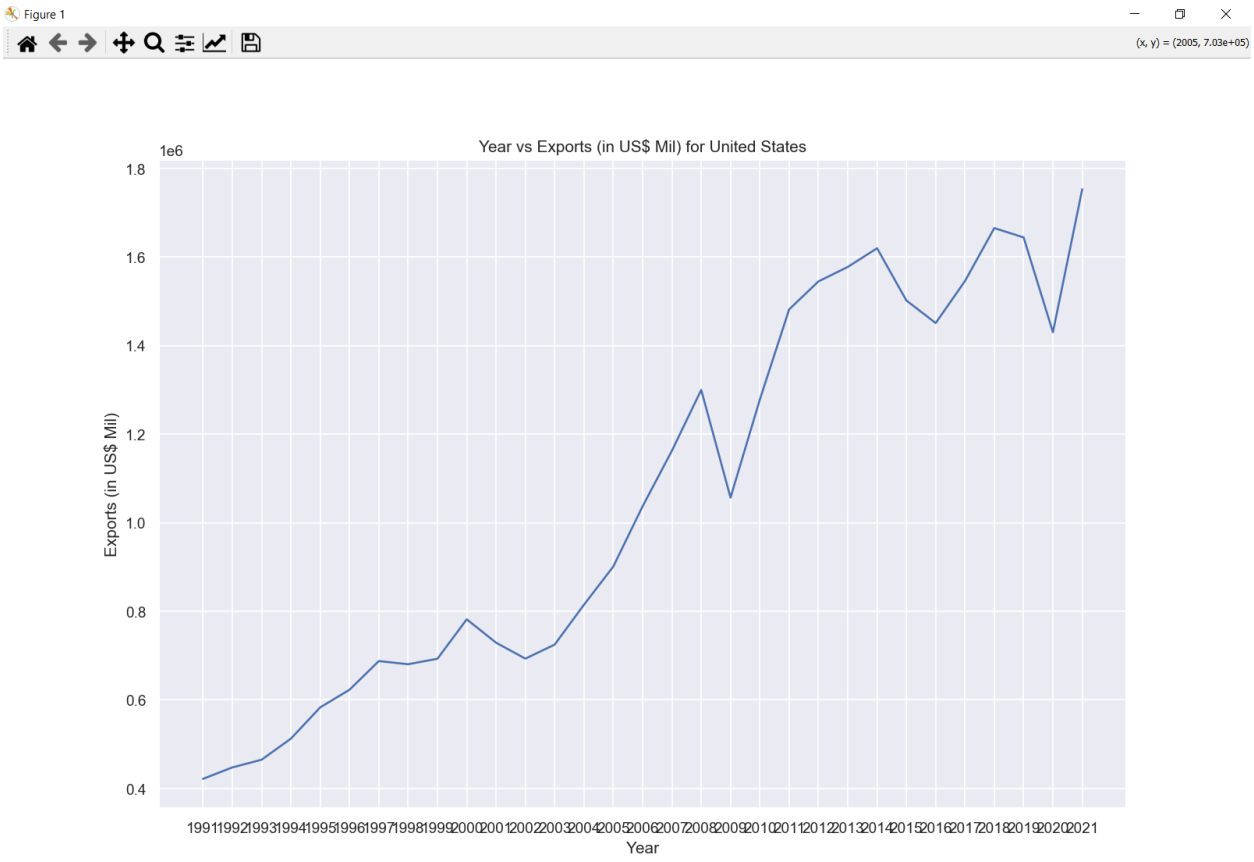
The Python code replaced the missing values with its column mean. The columns from 2016 through 2021 had their missing values replaced. The other column years were not needed. The data is in the rows for this dataset and missing values were replaced with the row mean.

```
<class 'pandas.core.frame.DataFrame'>
Index: 98 entries, 0 to 99
Data columns (total 36 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Reporter            98 non-null     object
 1   Partner             98 non-null     object
 2   Product categories  98 non-null     object
 3   Indicator Type      98 non-null     object
 4   Indicator           98 non-null     object
 5   2021                98 non-null     int32
 6   2020                98 non-null     float64
 7   2019                98 non-null     float64
 8   2018                98 non-null     float64
 9   2017                98 non-null     float64
 10  2016                98 non-null     float64
 11  2015                94 non-null     float64
 12  2014                94 non-null     float64
 13  2013                94 non-null     float64
 14  2012                94 non-null     float64
 15  2011                94 non-null     float64
 16  2010                94 non-null     float64
 17  2009                94 non-null     float64
 18  2008                94 non-null     float64
 19  2007                94 non-null     float64
 20  2006                94 non-null     float64
 21  2005                94 non-null     float64
 22  2004                94 non-null     float64
 23  2003                94 non-null     float64
 24  2002                94 non-null     float64
 25  2001                94 non-null     float64
 26  2000                94 non-null     float64
 27  1999                94 non-null     float64
 28  1998                94 non-null     float64
 29  1997                94 non-null     float64
 30  1996                94 non-null     float64
 31  1995                94 non-null     float64
 32  1994                84 non-null     float64
 33  1993                94 non-null     float64
 34  1992                94 non-null     float64
 35  1991                92 non-null     float64
dtypes: float64(30), int32(1), object(5)
memory usage: 27.9+ KB
None
```

Some missing values are still in rows that I do not use the data. The total number of rows has been reduced from 100 to 98 because a couple rows had no data or just one data point which could not calculate an average to fill in the rest of the row. So, the rows were deleted.

(d) Write Python code to find, report, and correct any incorrect values. You can use visual methods here. For example, you can "report" incorrect values visually.

Before



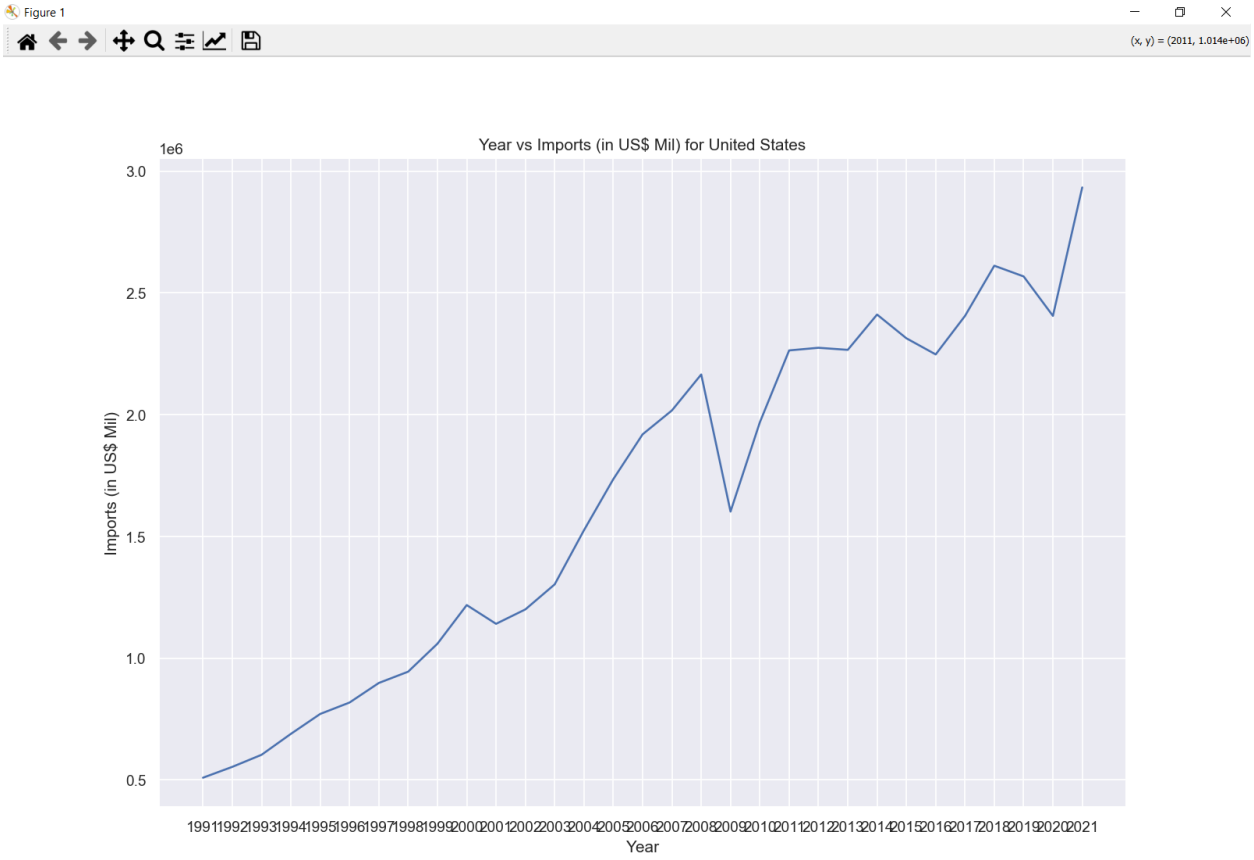
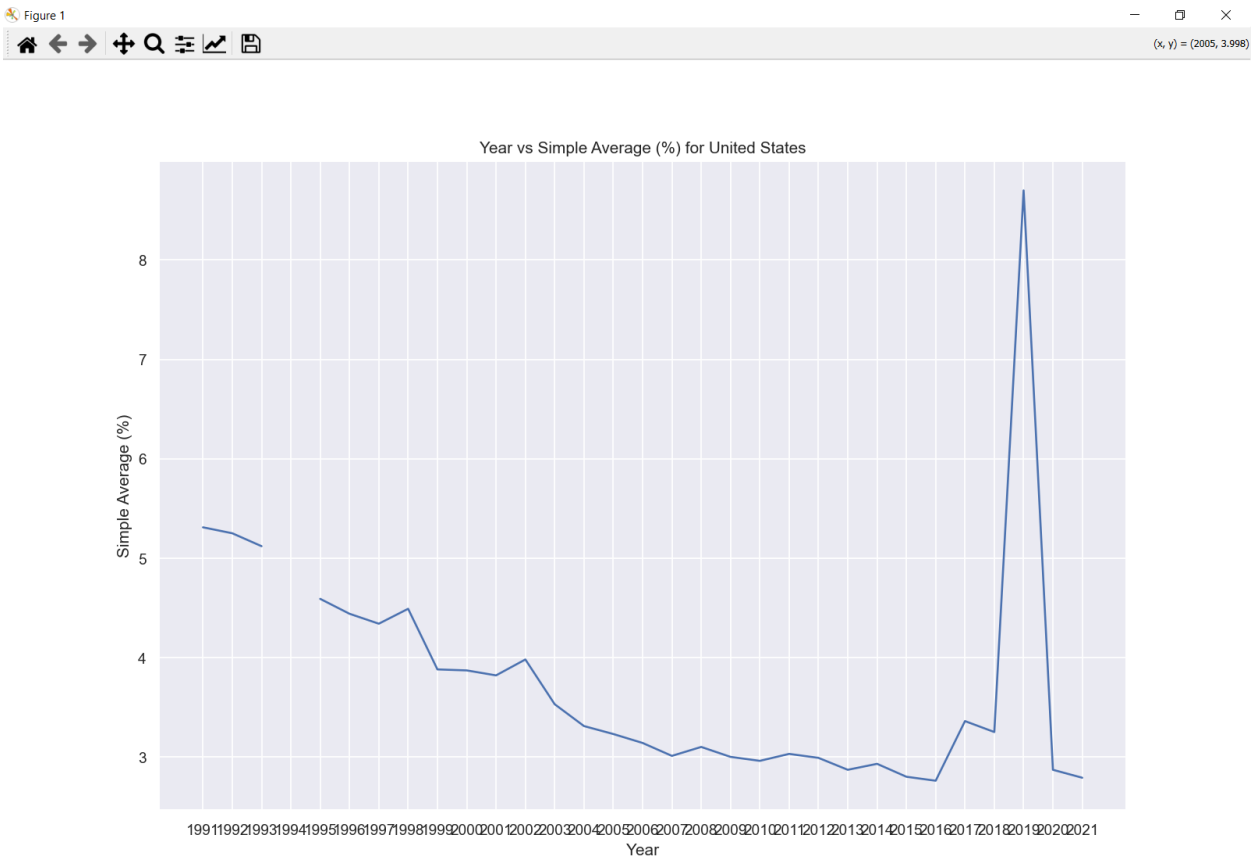


Figure 1
(x, y) = (2011, 1.014e+06)
Year vs Imports (in US$ Mil) for United States
1e6
3.0
2.5
2.0
1.5
1.0
0.5
Imports (in US$ Mil)
1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021
Year



Incorrect values and a big spike were detected by this visual.

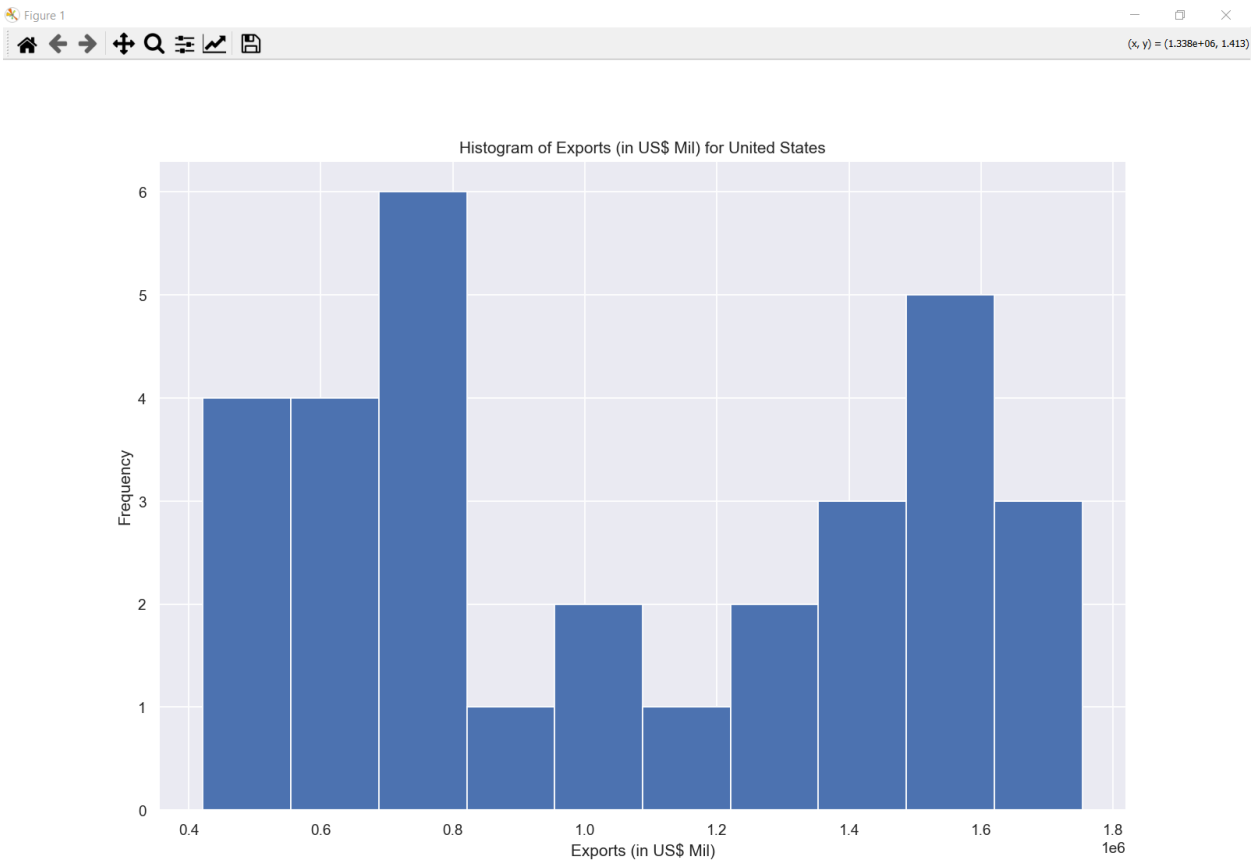


Figure 1
(x, y) = (1.338e+06, 1.413)
Histogram of Exports (in US$ Mil) for United States
Frequency
0
1
2
3
4
5
6
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
1e6
Exports (in US$ Mil)

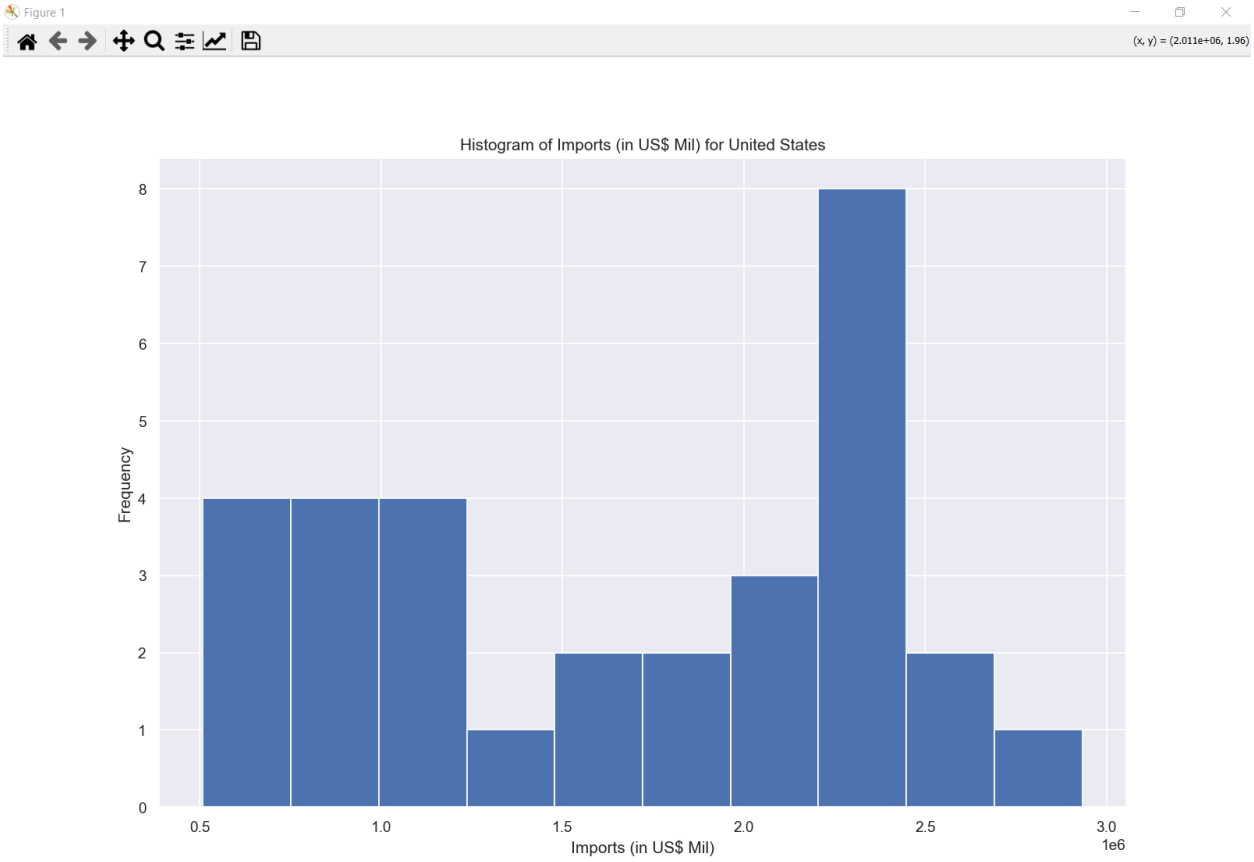
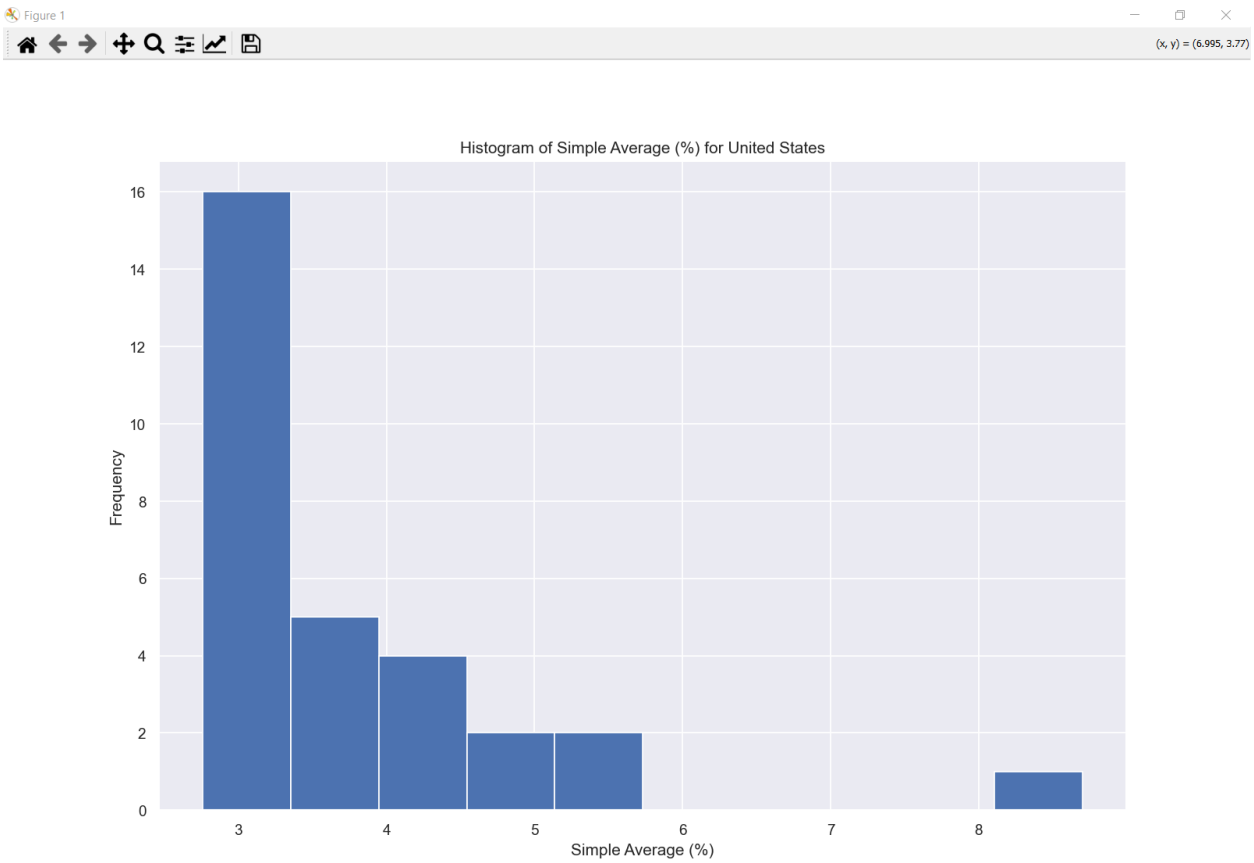


Figure 1
(x, y) = (2.011e+06, 1.96)
Histogram of Imports (in US$ Mil) for United States
Frequency
0
1
2
3
4
5
6
7
8
0.5
1.0
1.5
2.0
2.5
3.0
1e6
Imports (in US$ Mil)



The 8 value maybe an outlier. It is rather high but could be true, during those years the US imposing higher tariffs on the rest of the world was the trend. I will treat it as an outlier and make the correction as a demonstration of removing outliers.

After

Missing value for 1994 and 8 value for 2019 have been replaced with some local mean values. Now the graph looks in range. Import and Export graphs look fine. There were no corrections.

(e) Write Python code to find, report, and correct any values in the dataset(s) that might be correct but in the wrong format. For example, suppose you have a variable called "State" and the values can be state abbreviations like FL, OR, CA, etc. However, one of the entries is Fla. We know that this is FL and it needs to be updated to be the right (expected) format as prescribed by the dataset. Again, there are an infinite number of possibilities because all datasets are different. Take your time, explore your data, and determine how best to clean it.

Before

```
Data loaded successfully
<class 'pandas.core.frame.DataFrame'>
2021    1753136.71
2020    1430253.62
2019    1644276.22
2018    1665302.94
2017     1545809.6
2016    1450906.27
2015    1501845.86
2014    1619742.86
2013    1577587.25
2012    1544932.01
2011     1481682.2
2010    1278099.19
2009    1056712.08
2008    1299898.88
2007    1162538.15
2006    1037029.25
2005     901041.41
2004     814844.39
2003     724736.58
2002     693068.31
2001     729080.42
2000     781830.67
1999     692783.81
1998      680434.6
1997     687532.54
1996     622784.15
1995     582964.67
1994     512336.86
1993     464757.16
1992     447330.09
1991      421555.4
Name: 41, dtype: object
```

Exports

The decimal places need to be reduced to 1 decimal place.

```
2021    2932976.08
2020    2405381.56
2019     2567492.2
2018    2611432.49
2017    2405276.63
2016    2247167.25
2015    2313424.57
2014    2410855.48
2013    2265911.27
2012    2274461.87
2011    2263619.06
2010     1968259.9
2009    1601895.82
2008    2164834.03
2007    2017120.78
2006    1918997.09
2005    1734849.14
2004    1525304.22
2003    1302833.51
2002    1200095.83
2001    1140900.16
2000    1217932.97
1999    1059220.09
1998      944350.1
1997     898025.46
1996     817627.14
1995     770821.46
1994     689029.91
1993     603153.56
1992     553496.48
1991     508944.02
Name: 39, dtype: object
```

Imports

The decimal places need to be reduced to 1 decimal place.

```
2021    2.79
2020    2.87
2019    2.76
2018    3.25
2017    3.36
2016    2.76
2015     2.8
2014    2.93
2013    2.87
2012    2.99
2011    3.03
2010    2.96
2009     3.0
2008     3.1
2007    3.01
2006    3.14
2005    3.23
2004    3.31
2003    3.53
2002    3.98
2001    3.82
2000    3.87
1999    3.88
1998    4.49
1997    4.34
1996    4.44
1995    4.59
1994     4.6
1993    5.12
1992    5.25
1991    5.31
Name: 76, dtype: object
```

Simple Average

The decimal place needs to be reduced to 1 decimal place.

After

```
2021    1753136.7
2020    1430253.6
2019    1644276.2
2018    1665302.9
2017    1545809.6
2016    1450906.3
2015    1501845.9
2014    1619742.9
2013    1577587.2
2012    1544932.0
2011    1481682.2
2010    1278099.2
2009    1056712.1
2008    1299898.9
2007    1162538.2
2006    1037029.2
2005     901041.4
2004     814844.4
2003     724736.6
2002     693068.3
2001     729080.4
2000     781830.7
1999     692783.8
1998     680434.6
1997     687532.5
1996     622784.2
1995     582964.7
1994     512336.9
1993     464757.2
1992     447330.1
1991     421555.4
Name: 41, dtype: object
```

Exports

```
2021    2932976.1
2020    2405381.6
2019    2567492.2
2018    2611432.5
2017    2405276.6
2016    2247167.2
2015    2313424.6
2014    2410855.5
2013    2265911.3
2012    2274461.9
2011    2263619.1
2010    1968259.9
2009    1601895.8
2008    2164834.0
2007    2017120.8
2006    1918997.1
2005    1734849.1
2004    1525304.2
2003    1302833.5
2002    1200095.8
2001    1140900.2
2000    1217933.0
1999    1059220.1
1998     944350.1
1997     898025.5
1996     817627.1
1995     770821.5
1994     689029.9
1993     603153.6
1992     553496.5
1991     508944.0
Name: 39, dtype: object
```

Imports

```
2021    2.8
2020    2.9
2019    2.8
2018    3.2
2017    3.4
2016    2.8
2015    2.8
2014    2.9
2013    2.9
2012    3.0
2011    3.0
2010    3.0
2009    3.0
2008    3.1
2007    3.0
2006    3.1
2005    3.2
2004    3.3
2003    3.5
2002    4.0
2001    3.8
2000    3.9
1999    3.9
1998    4.5
1997    4.3
1996    4.4
1995    4.6
1994    4.6
1993    5.1
1992    5.2
1991    5.3
Name: 76, dtype: object
```

Simple Average

The Python code changed the decimal places using Panda command.

(f) Write Python code to find, visualize, and correct any outliers.

Before

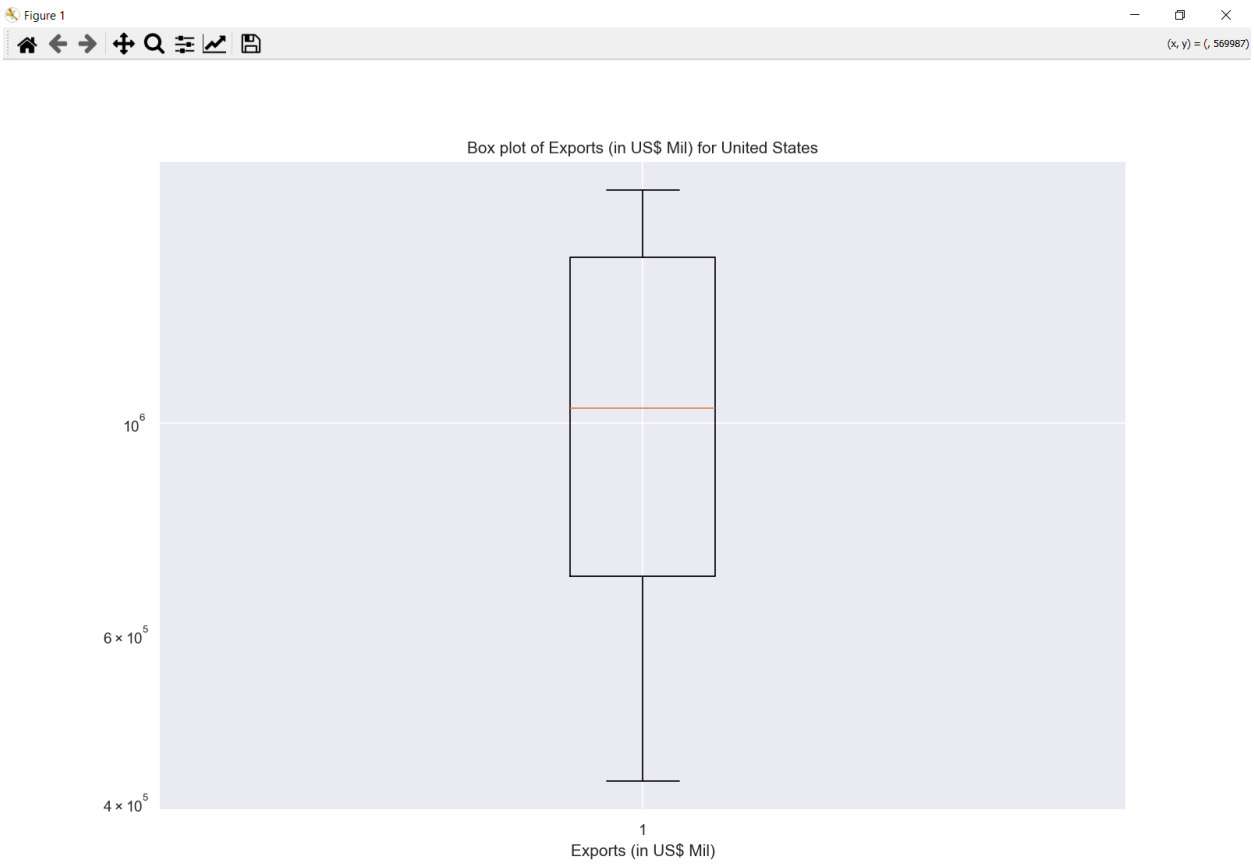


Figure 1
(x, y) = (, 569987)
Box plot of Exports (in US$ Mil) for United States
10^6
6 × 10^5
4 × 10^5
1
Exports (in US$ Mil)

Exports

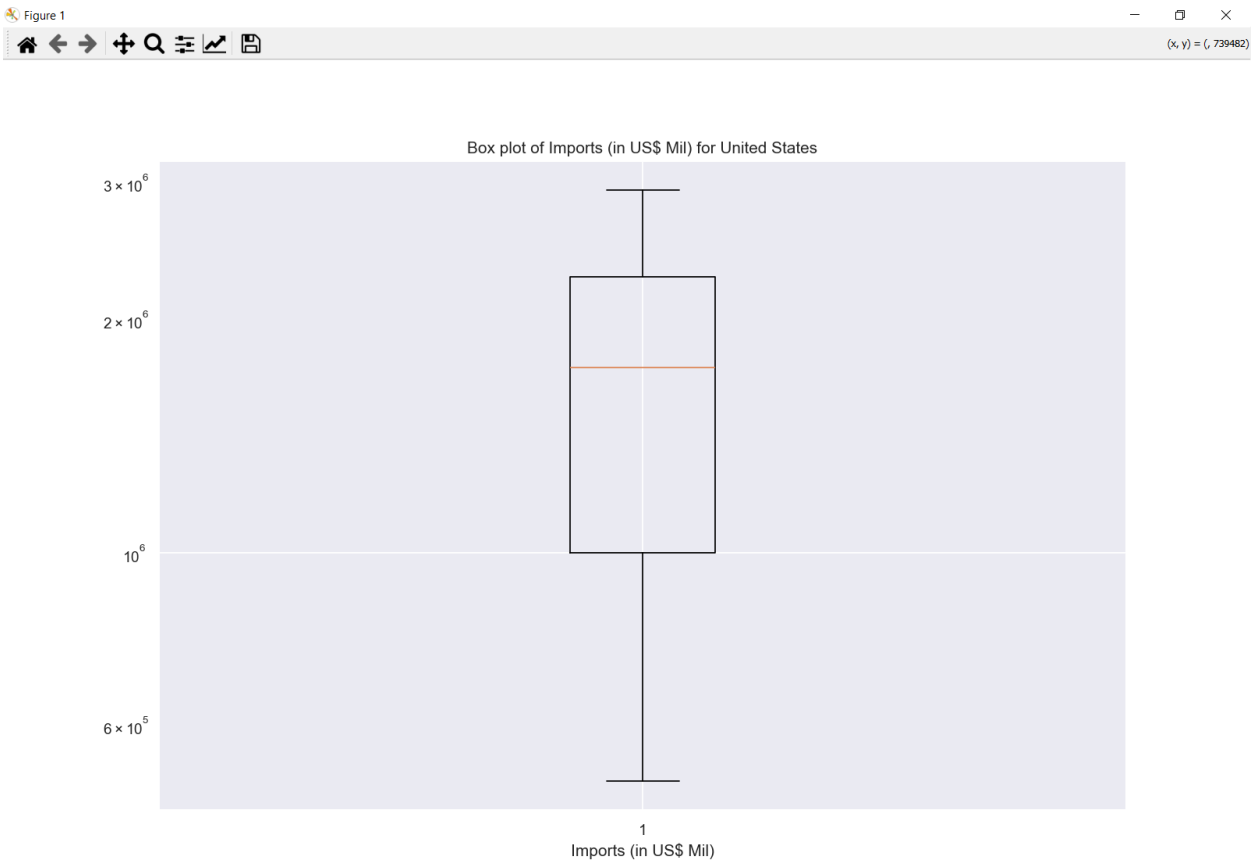
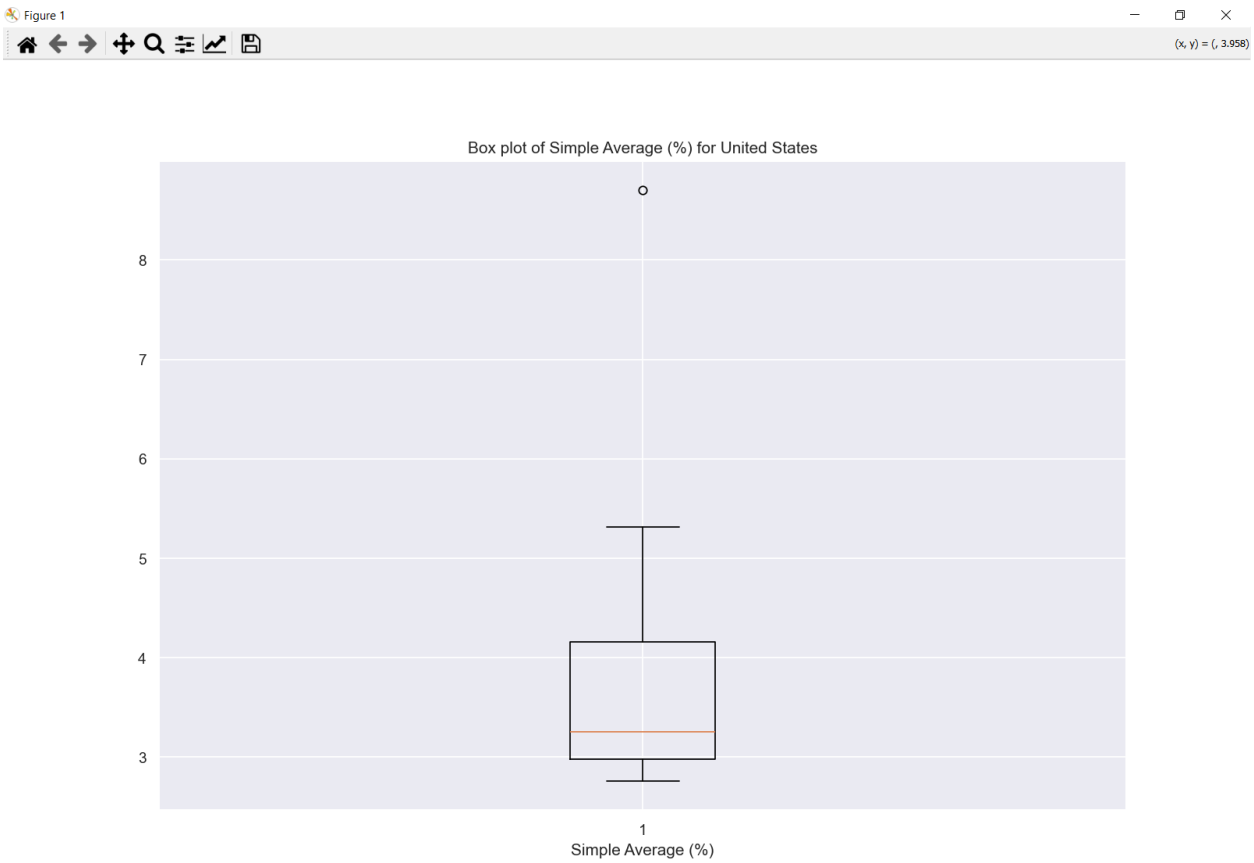


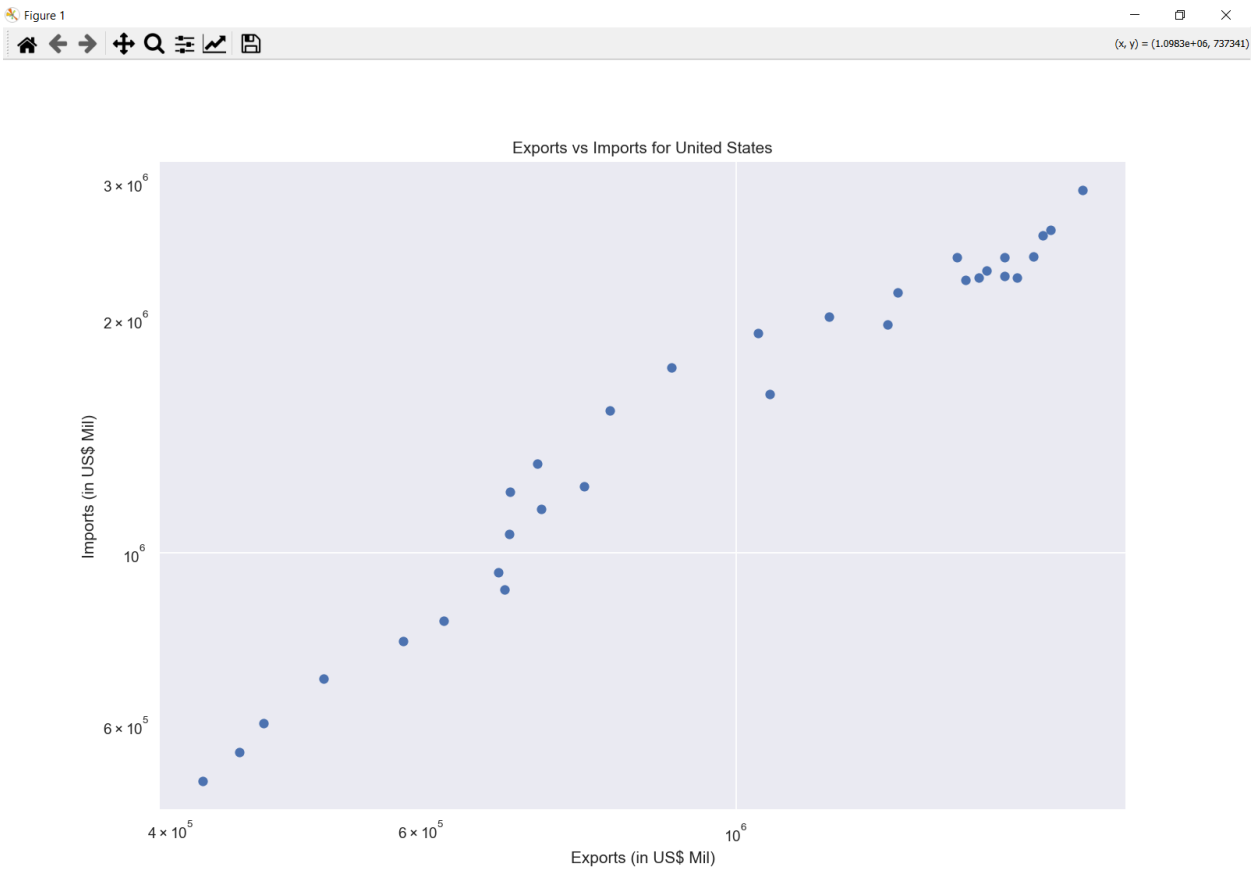
Figure 1
(x, y) = (, 739482)
Box plot of Imports (in US$ Mil) for United States
3 × 10^6
2 × 10^6
10^6
6 × 10^5
1
Imports (in US$ Mil)

Imports



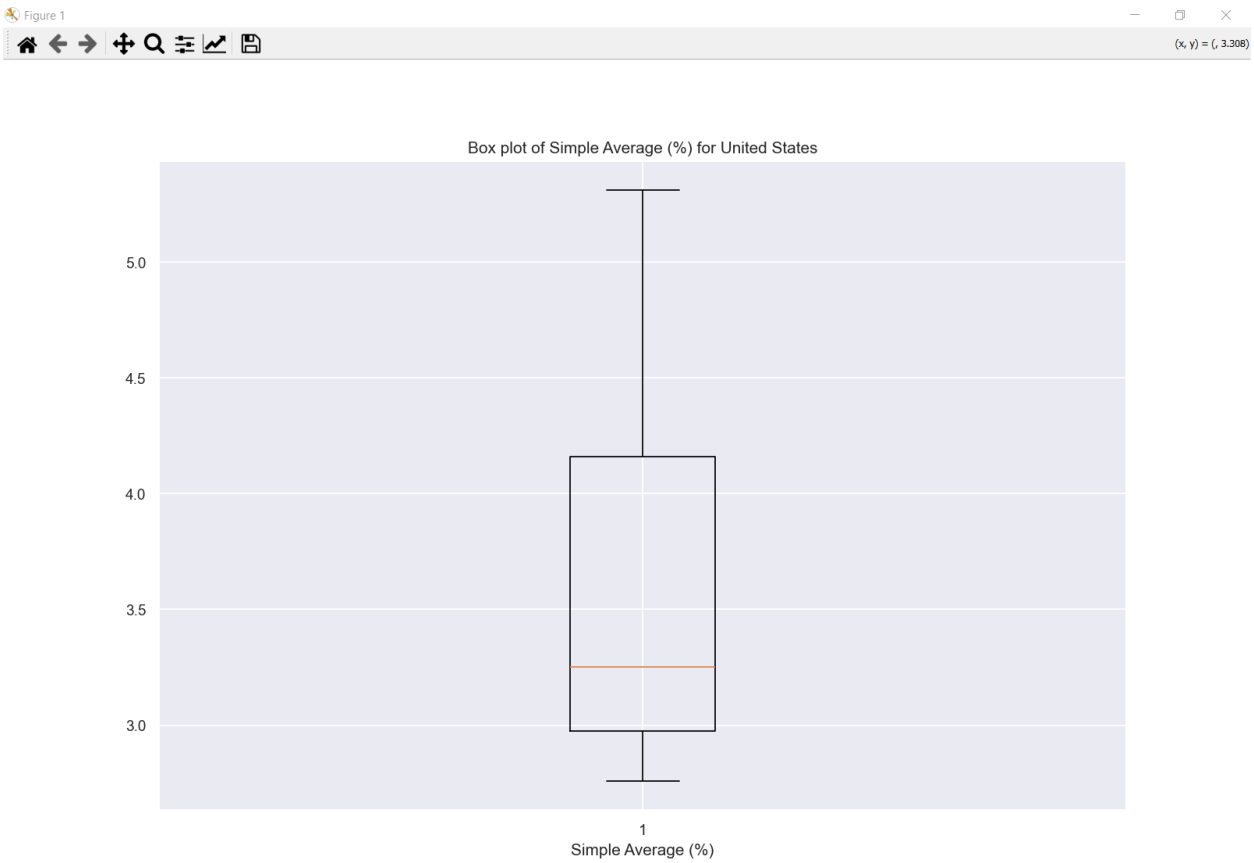
Simple Average

There is an outlier 8 value that sticks out at the top.



Log scale

After



Linear scale.

No more 8 outlier value in dataset.

(g) Write Python code to create a new dataframe from one of your datasets such that the new dataframe is normalized using min-max. Be sure to include a screen image of the normalized dataset in your report.

Before

```
Data loaded successfully
<class 'pandas.core.frame.DataFrame'>
        Reporter Partner Product categories Indicator Type  \
39  United States   World       All Products         Import

             Indicator        2021        2020       2019        2018  \
39  Imports (in US$ Mil)  2932976.08  2405381.56  2567492.2  2611432.49

         2017        2016        2015        2014        2013        2012  \
39  2405276.63  2247167.25  2313424.57  2410855.48  2265911.27  2274461.87

         2011        2010        2009        2008        2007        2006  \
39  2263619.06  1968259.9  1601895.82  2164834.03  2017120.78  1918997.09

         2005        2004        2003        2002        2001        2000  \
39  1734849.14  1525304.22  1302833.51  1200095.83  1140900.16  1217932.97

         1999      1998       1997       1996       1995       1994  \
39  1059220.09  944350.1  898025.46  817627.14  770821.46  689029.91

        1993       1992       1991
39  603153.56  553496.48  508944.02
```

```
        Reporter Partner Product categories Indicator Type  \
41  United States   World       All Products         Export

             Indicator        2021        2020        2019        2018  \
41  Exports (in US$ Mil)  1753136.71  1430253.62  1644276.22  1665302.94

        2017        2016        2015        2014        2013        2012  \
41  1545809.6  1450906.27  1501845.86  1619742.86  1577587.25  1544932.01

        2011        2010        2009        2008        2007        2006  \
41  1481682.2  1278099.19  1056712.08  1299898.88  1162538.15  1037029.25

        2005       2004       2003       2002       2001       2000  \
41  901041.41  814844.39  724736.58  693068.31  729080.42  781830.67

        1999      1998       1997       1996       1995       1994  \
41  692783.81  680434.6  687532.54  622784.15  582964.67  512336.86

        1993       1992      1991
41  464757.16  447330.09  421555.4
```

Export and Import are chosen for min max normalization because they have a broad range of values.

After

```
new dataset normalized_Export and normalized_Import created
```

A new dataset with normalized import and export data is stored. It is easier to print and chart this data.

```
Normalized Export Data:
2021         1.0
2020    0.757519
2019    0.918247
2018    0.934038
2017      0.8443
2016    0.773029
2015    0.811284
2014    0.899823
2013    0.868165
2012    0.843641
2011    0.796141
2010    0.643253
2009    0.476994
2008    0.659624
2007    0.556468
2006    0.462213
2005    0.360088
2004    0.295355
2003    0.227685
2002    0.203903
2001    0.230947
2000    0.270562
1999    0.203689
1998    0.194415
1997    0.199745
1996     0.15112
1995    0.121216
1994    0.068176
1993    0.032444
1992    0.019356
1991         0.0
Name: 41, dtype: object
```

```
Normalized Import Data:
2021         1.0
2020    0.782348
2019    0.849225
2018    0.867352
2017    0.782305
2016    0.717079
2015    0.744413
2014    0.784607
2013    0.724812
2012    0.728339
2011    0.723866
2010     0.60202
2009    0.450882
2008    0.683114
2007    0.622177
2006    0.581697
2005     0.50573
2004    0.419285
2003    0.327508
2002    0.285125
2001    0.260705
2000    0.292483
1999    0.227009
1998    0.179621
1997     0.16051
1996    0.127343
1995    0.108034
1994    0.074292
1993    0.038865
1992    0.018379
1991         0.0
Name: 39, dtype: object
Press any key to continue . . .
```

The Python code normalized Export and Import data to a range of 0 to 1.

(h) Write Python code to create a new dataframe that contains only unlabeled and quantitative data. Be sure to include a screen image in your report.

The new dataframe will only be numeric.

Before

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 98 entries, 0 to 97
Data columns (total 36 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Reporter            98 non-null     object
 1   Partner             98 non-null     object
 2   Product categories  98 non-null     object
 3   Indicator Type      98 non-null     object
 4   Indicator           98 non-null     object
 5   2021                98 non-null     float64
 6   2020                98 non-null     float64
 7   2019                98 non-null     float64
 8   2018                98 non-null     float64
 9   2017                98 non-null     float64
 10  2016                98 non-null     float64
 11  2015                94 non-null     float64
 12  2014                94 non-null     float64
 13  2013                94 non-null     float64
 14  2012                94 non-null     float64
 15  2011                94 non-null     float64
 16  2010                94 non-null     float64
 17  2009                94 non-null     float64
 18  2008                94 non-null     float64
 19  2007                94 non-null     float64
 20  2006                94 non-null     float64
 21  2005                94 non-null     float64
 22  2004                94 non-null     float64
 23  2003                94 non-null     float64
 24  2002                94 non-null     float64
 25  2001                94 non-null     float64
 26  2000                94 non-null     float64
 27  1999                94 non-null     float64
 28  1998                94 non-null     float64
 29  1997                94 non-null     float64
 30  1996                94 non-null     float64
 31  1995                94 non-null     float64
 32  1994                85 non-null     float64
 33  1993                94 non-null     float64
 34  1992                94 non-null     float64
 35  1991                92 non-null     float64
dtypes: float64(31), object(5)
memory usage: 27.7+ KB
None
```

	Reporter	Partner	Product categories	Indicator Type	\
0	United States	World	Consumer goods	Export	
1	United States	World	Minerals	Export	
2	United States	Germany	All Products	Import	
3	United States	World	Raw materials	Export	
4	United States	World	Capital goods	Export	
5	United States	World	Plastic or Rubber	Export	
6	United States	Japan	All Products	Import	
7	United States	World	Vegetable	Export	
8	United States	World	Footwear	Export	
9	United States	Canada	All Products	Import	
10	United States	World	Miscellaneous	Export	
11	United States	China	All Products	Export	
12	United States	...	...	Import	
13	United States	...	...	Export	
14	United States	...	...	Export	

	Indicator	2021	2020	2019	\
0	Export(US$ Mil)	475384.27	369729.51	430333.57	
1	Export(US$ Mil)	14682.54	10148.96	10706.74	
2	Trade (US$ Mil)-Top 5 Import Partner	138194.63	117393.02	129857.18	
3	Export(US$ Mil)	246705.65	190294.98	200924.18	
4	Export(US$ Mil)	528779.38	468653.02	528561.02	
5	Export(US$ Mil)	87919.19	71827.92	78596.95	
6	Trade (US$ Mil)-Top 5 Import Partner	139389.68	122483.99	146974.31	
7	Export(US$ Mil)	91825.56	76622.11	67237.20	
8	Export(US$ Mil)	1683.88	1581.41	2157.25	
9	Trade (US$ Mil)-Top 5 Import Partner	363904.69	276195.55	326628.56	
10	Export(US$ Mil)	194008.39	169117.84	189065.32	
11	Trade (US$ Mil)-Top 5 Export Partner	151065.18	124648.51	106626.65	
12	No. Of Import products	4531.00	4525.00	4529.00	
13	No. Of Export products	4524.00	4526.00	4529.00	
14	No. Of Export partners	222.00	222.00	223.00	

There are five qualitative columns which will be deleted. Reporter, Partner, Product categories, Indicator Type and Indicator.

After

```
new dataset MyData_quant created
```

The Python code looks at row data and columns 1991 through 2021 and are extracted and placed in new dataset for printing and charting.

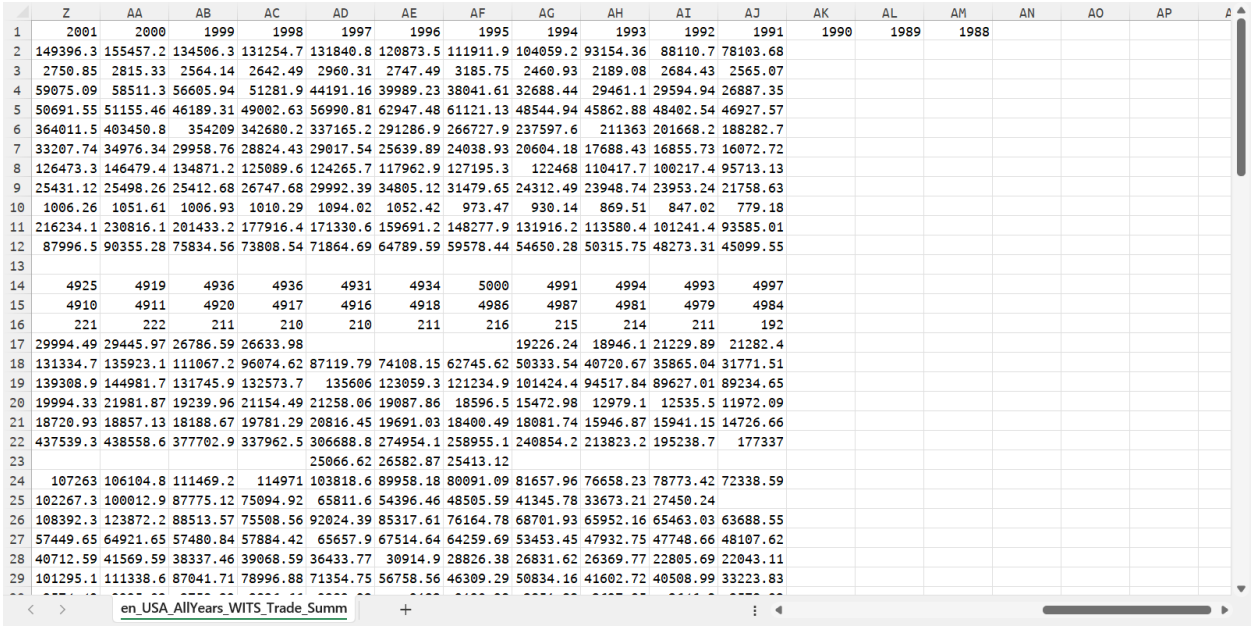
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 98 entries, 0 to 97
Data columns (total 31 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   2021    98 non-null     float64
 1   2020    98 non-null     float64
 2   2019    98 non-null     float64
 3   2018    98 non-null     float64
 4   2017    98 non-null     float64
 5   2016    98 non-null     float64
 6   2015    94 non-null     float64
 7   2014    94 non-null     float64
 8   2013    94 non-null     float64
 9   2012    94 non-null     float64
 10  2011    94 non-null     float64
 11  2010    94 non-null     float64
 12  2009    94 non-null     float64
 13  2008    94 non-null     float64
 14  2007    94 non-null     float64
 15  2006    94 non-null     float64
 16  2005    94 non-null     float64
 17  2004    94 non-null     float64
 18  2003    94 non-null     float64
 19  2002    94 non-null     float64
 20  2001    94 non-null     float64
 21  2000    94 non-null     float64
 22  1999    94 non-null     float64
 23  1998    94 non-null     float64
 24  1997    94 non-null     float64
 25  1996    94 non-null     float64
 26  1995    94 non-null     float64
 27  1994    85 non-null     float64
 28  1993    94 non-null     float64
 29  1992    94 non-null     float64
 30  1991    92 non-null     float64
dtypes: float64(31)
memory usage: 23.9 KB
None
```

The Python code deleted all the qualitative columns and all the numeric columns are left.

```
        2021       2020       2019       2018       2017       2016  \
0   475384.27  369729.51  430333.57  431527.79  400719.11  370436.52
1    14682.54   10148.96   10706.74   10387.62    9407.67    8118.80
2   138194.63  117393.02  129857.18  128345.62  119962.97  116259.43
3   246705.65  190294.98  200924.18  189985.11  158069.68  136670.47
4   528779.38  468653.02  528561.02  540285.30  518558.06  498512.90
5    87919.19   71827.92   78596.95   81008.39   75825.19   71655.24
6   139389.68  122483.99  146974.31  145902.25  139733.00  135071.04
7    91825.56   76622.11   67237.20   69501.02   71553.04   72138.16
8     1683.88    1581.41    2157.25    2089.32    1924.89    1868.50
9   363904.69  276195.55  326628.56  325683.55  305647.66  282919.22
10  194008.39  169117.84  189065.32  189960.05  177167.04  168832.29
11  151065.18  124648.51  106626.65  120147.87  129797.52  115594.77
12    4531.00    4525.00    4529.00    4531.00    4527.00    4558.00
13    4524.00    4526.00    4529.00    4533.00    4523.00    4563.00
14     222.00     222.00     223.00     224.00     224.00     225.00

        2015       2014       2013       2012       2011       2010  \
0   388367.79  442287.78  427244.98  404361.66  376012.37  308042.05
1     9489.88   11218.75    9917.61    9636.59   10201.93    7823.45
2   127170.57  125532.37  114338.45  109216.64  100675.93   84129.52
3   136449.47  162869.32  153991.59  158800.07  164758.98  135180.76
4   517758.98  536096.70  519829.22  518805.16  492760.97  445798.52
5    74344.87   78350.75   76125.37   75065.07   73995.19   66075.81
6   135023.80  137503.84  138574.36  146431.64  132558.80  123762.73
7    67976.11   78192.80   73667.52   76186.80   74074.60   63841.69
8     1934.20    1888.77    1805.56    1748.20    1655.94    1416.81
9   301942.71  354171.83  331016.25  323026.40  317921.22  280426.52
10  290291.52  285865.55  271546.42  256846.61  229990.09  215120.30
11  116071.71  123675.62  121721.08  110516.54  104121.38   91910.98
12    4561.00    4555.00    4558.00    4562.00    4582.00    4577.00
13    4559.00    4560.00    4569.00    4568.00    4600.00    4602.00
14     224.00     225.00     225.00     226.00     225.00     223.00
```

(i) Finally, take any further steps you feel are needed to clean up your data such as discretization and feature generation.

Before



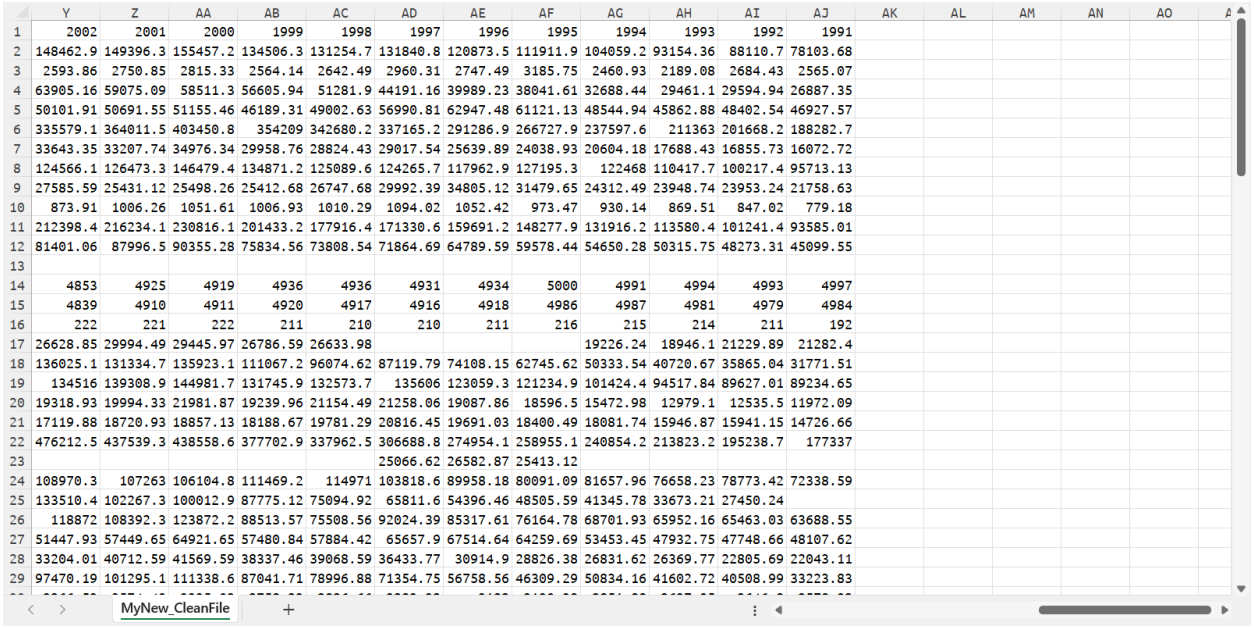
	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP
1	2001	2000	1999	1998	1997	1996	1995	1994	1993	1992	1991	1990	1989	1988			
2	149396.3	155457.2	134506.3	131254.7	131840.8	120873.5	111911.9	104059.2	93154.36	88110.7	78103.68						
3	2750.85	2815.33	2564.14	2642.49	2960.31	2747.49	3185.75	2460.93	2189.08	2684.43	2565.07						
4	59075.09	58511.3	56605.94	51281.9	44191.16	39989.23	38041.61	32688.44	29461.1	29594.94	26887.35						
5	50691.55	51155.46	46189.31	49002.63	56990.81	62947.48	61121.13	48544.94	45862.88	48402.54	46927.57						
6	364011.5	403450.8	354209	342680.2	337165.2	291286.9	266727.9	237597.6	211363	201668.2	188282.7						
7	33207.74	34976.34	29958.76	28824.43	29017.54	25639.89	24038.93	20604.18	17688.43	16855.73	16072.72						
8	126473.3	146479.4	134871.2	125089.6	124265.7	117962.9	127195.3	122468	110417.7	100217.4	95713.13						
9	25431.12	25498.26	25412.68	26747.68	29992.39	34805.12	31479.65	24312.49	23948.74	23953.24	21758.63						
10	1006.26	1051.61	1006.93	1010.29	1094.02	1052.42	973.47	930.14	869.51	847.02	779.18						
11	216234.1	230816.1	201433.2	177916.4	171330.6	159691.2	148277.9	131916.2	113580.4	101241.4	93585.01						
12	87996.5	90355.28	75834.56	73808.54	71864.69	64789.59	59578.44	54650.28	50315.75	48273.31	45099.55						
13																	
14	4925	4919	4936	4936	4931	4934	5000	4991	4994	4993	4997						
15	4910	4911	4920	4917	4916	4918	4986	4987	4981	4979	4984						
16	221	222	211	210	210	211	216	215	214	211	192						
17	29994.49	29445.97	26786.59	26633.98				19226.24	18946.1	21229.89	21282.4						
18	131334.7	135923.1	111067.2	96074.62	87119.79	74108.15	62745.62	50333.54	40720.67	35865.04	31771.51						
19	139308.9	144981.7	131745.9	132573.7	135606	123059.3	121234.9	101424.4	94517.84	89627.01	89234.65						
20	19994.33	21981.87	19239.96	21154.49	21258.06	19087.86	18596.5	15472.98	12979.1	12535.5	11972.09						
21	18720.93	18857.13	18188.67	19781.29	20816.45	19691.03	18400.49	18081.74	15946.87	15941.15	14726.66						
22	437539.3	438558.6	377702.9	337962.5	306688.8	274954.1	258955.1	240854.2	213823.2	195238.7	177337						
23					25066.62	26582.87	25413.12										
24	107263	106104.8	111469.2	114971	103818.6	89958.18	80091.09	81657.96	76658.23	78773.42	72338.59						
25	102267.3	100012.9	87775.12	75094.92	65811.6	54396.46	48505.59	41345.78	33673.21	27450.24							
26	108392.3	123872.2	88513.57	75508.56	92024.39	85317.61	76164.78	68701.93	65952.16	65463.03	63688.55						
27	57449.65	64921.65	57480.84	57884.42	65657.9	67514.64	64259.69	53453.45	47932.75	47748.66	48107.62						
28	40712.59	41569.59	38337.46	39068.59	36433.77	30914.9	28826.38	26831.62	26369.77	22805.69	22043.11						
29	101295.1	111338.6	87041.71	78996.88	71354.75	56758.56	46309.29	50834.16	41602.72	40508.99	33223.83						

en_USA_AllYears_WITS_Trade_Summ

```
Index(['Reporter', 'Partner', 'Product categories', 'Indicator Type',
       'Indicator', '2021', '2020', '2019', '2018', '2017', '2016', '2015',
       '2014', '2013', '2012', '2011', '2010', '2009', '2008', '2007', '2006',
       '2005', '2004', '2003', '2002', '2001', '2000', '1999', '1998', '1997',
       '1996', '1995', '1994', '1993', '1992', '1991', '1990', '1989', '1988'],
      dtype='object')
```

There are some empty year columns that have a lot of missing data. There was not enough data to make a mean for replacement. So they were deleted.

After



	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO
1	2002	2001	2000	1999	1998	1997	1996	1995	1994	1993	1992	1991					
2	148462.9	149396.3	155457.2	134506.3	131254.7	131840.8	120873.5	111911.9	104059.2	93154.36	88110.7	78103.68					
3	2593.86	2750.85	2815.33	2564.14	2642.49	2960.31	2747.49	3185.75	2460.93	2189.08	2684.43	2565.07					
4	63905.16	59075.09	58511.3	56605.94	51281.9	44191.16	39989.23	38041.61	32688.44	29461.1	29594.94	26887.35					
5	50101.91	50691.55	51155.46	46189.31	49002.63	56990.81	62947.48	61121.13	48544.94	45862.88	48402.54	46927.57					
6	335579.1	364011.5	403450.8	354209	342680.2	337165.2	291286.9	266727.9	237597.6	211363	201668.2	188282.7					
7	33643.35	33207.74	34976.34	29958.76	28824.43	29017.54	25639.89	24038.93	20604.18	17688.43	16855.73	16072.72					
8	124566.1	126473.3	146479.4	134871.2	125089.6	124265.7	117962.9	127195.3	122468	110417.7	100217.4	95713.13					
9	27585.59	25431.12	25498.26	25412.68	26747.68	29992.39	34805.12	31479.65	24312.49	23948.74	23953.24	21758.63					
10	873.91	1006.26	1051.61	1006.93	1010.29	1094.02	1052.42	973.47	930.14	869.51	847.02	779.18					
11	212398.4	216234.1	230816.1	201433.2	177916.4	171330.6	159691.2	148277.9	131916.2	113580.4	101241.4	93585.01					
12	81401.06	87996.5	90355.28	75834.56	73808.54	71864.69	64789.59	59578.44	54650.28	50315.75	48273.31	45099.55					
13																	
14	4853	4925	4919	4936	4936	4931	4934	5000	4991	4994	4993	4997					
15	4839	4910	4911	4920	4917	4916	4918	4986	4987	4981	4979	4984					
16	222	221	222	211	210	210	211	216	215	214	211	192					
17	26628.85	29994.49	29445.97	26786.59	26633.98				19226.24	18946.1	21229.89	21282.4					
18	136025.1	131334.7	135923.1	111067.2	96074.62	87119.79	74108.15	62745.62	50333.54	40720.67	35865.04	31771.51					
19	134516	139308.9	144981.7	131745.9	132573.7	135606	123059.3	121234.9	101424.4	94517.84	89627.01	89234.65					
20	19318.93	19994.33	21981.87	19239.96	21154.49	21258.06	19087.86	18596.5	15472.98	12979.1	12535.5	11972.09					
21	17119.88	18720.93	18857.13	18188.67	19781.29	20816.45	19691.03	18400.49	18081.74	15946.87	15941.15	14726.66					
22	476212.5	437539.3	438558.6	377702.9	337962.5	306688.8	274954.1	258955.1	240854.2	213823.2	195238.7	177337					
23						25066.62	26582.87	25413.12									
24	108970.3	107263	106104.8	111469.2	114971	103818.6	89958.18	80091.09	81657.96	76658.23	78773.42	72338.59					
25	133510.4	102267.3	100012.9	87775.12	75094.92	65811.6	54396.46	48505.59	41345.78	33673.21	27450.24						
26	118872	108392.3	123872.2	88513.57	75508.56	92024.39	85317.61	76164.78	68701.93	65952.16	65463.03	63688.55					
27	51447.93	57449.65	64921.65	57480.84	57884.42	65657.9	67514.64	64259.69	53453.45	47932.75	47748.66	48107.62					
28	33204.01	40712.59	41569.59	38337.46	39068.59	36433.77	30914.9	28826.38	26831.62	26369.77	22805.69	22043.11					
29	97470.19	101295.1	111338.6	87041.71	78996.88	71354.75	56758.56	46309.29	50834.16	41602.72	40508.99	33223.83					

MyNew_CleanFile

```
Index(['Reporter', 'Partner', 'Product categories', 'Indicator Type',
       'Indicator', '2021', '2020', '2019', '2018', '2017', '2016', '2015',
       '2014', '2013', '2012', '2011', '2010', '2009', '2008', '2007', '2006',
       '2005', '2004', '2003', '2002', '2001', '2000', '1999', '1998', '1997',
       '1996', '1995', '1994', '1993', '1992', '1991'],
      dtype='object')
```

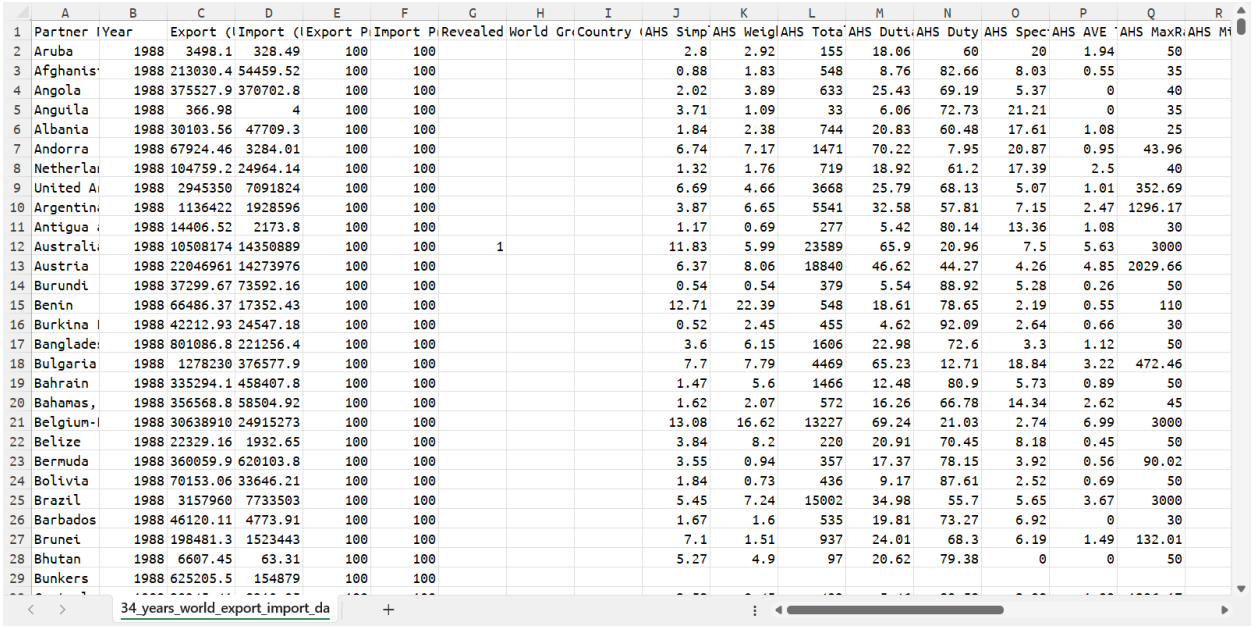
The Python code deleted columns 1990, 1989, 1988. There was no data to salvage.

Unsupervised Learning with KMeans Clustering.

From Module 7 Exercise subsections. The overall goal here is to both clean and prepare this data for KMeans analysis.

(a) Format the Data (to apply k means clustering.

First, include a "before" image of the data that you plan to use.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Partner I	Year	Export (I	Import (I	Export P	Import P	Revealed	World Gr	Country (	AHS Simp	AHS Weig	AHS Tota	AHS Duti	AHS Duty	AHS Spec	AHS AVE	AHS MaxR	AHS M
2	Aruba	1988	3498.1	328.49	100	100				2.8	2.92	155	18.06	60	20	1.94	50	
3	Afghanis	1988	213030.4	54459.52	100	100				0.88	1.83	548	8.76	82.66	8.03	0.55	35	
4	Angola	1988	375527.9	370702.8	100	100				2.02	3.89	633	25.43	69.19	5.37	0	40	
5	Anguila	1988	366.98	4	100	100				3.71	1.09	33	6.06	72.73	21.21	0	35	
6	Albania	1988	30103.56	47709.3	100	100				1.84	2.38	744	20.83	60.48	17.61	1.08	25	
7	Andorra	1988	67924.46	3284.01	100	100				6.74	7.17	1471	70.22	7.95	20.87	0.95	43.96	
8	Netherla	1988	104759.2	24964.14	100	100				1.32	1.76	719	18.92	61.2	17.39	2.5	40	
9	United A	1988	2945350	7091824	100	100				6.69	4.66	3668	25.79	68.13	5.07	1.01	352.69	
10	Argentin	1988	1136422	1928596	100	100				3.87	6.65	5541	32.58	57.81	7.15	2.47	1296.17	
11	Antigua	1988	14406.52	2173.8	100	100				1.17	0.69	277	5.42	80.14	13.36	1.08	30	
12	Australi	1988	10508174	14350889	100	100	1			11.83	5.99	23589	65.9	20.96	7.5	5.63	3000	
13	Austria	1988	22046961	14273976	100	100				6.37	8.06	18840	46.62	44.27	4.26	4.85	2029.66	
14	Burundi	1988	37299.67	73592.16	100	100				0.54	0.54	379	5.54	88.92	5.28	0.26	50	
15	Benin	1988	66486.37	17352.43	100	100				12.71	22.39	548	18.61	78.65	2.19	0.55	110	
16	Burkina	1988	42212.93	24547.18	100	100				0.52	2.45	455	4.62	92.09	2.64	0.66	30	
17	Banglade	1988	801086.8	221256.4	100	100				3.6	6.15	1606	22.98	72.6	3.3	1.12	50	
18	Bulgaria	1988	1278230	376577.9	100	100				7.7	7.79	4469	65.23	12.71	18.84	3.22	472.46	
19	Bahrain	1988	335294.1	458407.8	100	100				1.47	5.6	1466	12.48	80.9	5.73	0.89	50	
20	Bahamas,	1988	356568.8	58504.92	100	100				1.62	2.07	572	16.26	66.78	14.34	2.62	45	
21	Belgium-	1988	30638910	24915273	100	100				13.08	16.62	13227	69.24	21.03	2.74	6.99	3000	
22	Belize	1988	22329.16	1932.65	100	100				3.84	8.2	220	20.91	70.45	8.18	0.45	50	
23	Bermuda	1988	360059.9	620103.8	100	100				3.55	0.94	357	17.37	78.15	3.92	0.56	90.02	
24	Bolivia	1988	70153.06	33646.21	100	100				1.84	0.73	436	9.17	87.61	2.52	0.69	50	
25	Brazil	1988	3157960	7733503	100	100				5.45	7.24	15002	34.98	55.7	5.65	3.67	3000	
26	Barbados	1988	46120.11	4773.91	100	100				1.67	1.6	535	19.81	73.27	6.92	0	30	
27	Brunei	1988	198481.3	1523443	100	100				7.1	1.51	937	24.01	68.3	6.19	1.49	132.01	
28	Bhutan	1988	6607.45	63.31	100	100				5.27	4.9	97	20.62	79.38	0	0	50	
29	Bunkers	1988	625205.5	154879	100	100												

34_years_world_export_import_da

This is a csv file shown in Excel. There are over 30 columns but all are not shown.

- Removing any columns you plan not to use for k means.

All the unused columns for KMeans were removed, leaving just 5 columns. Partner Name, Year, Export, Import, Simple Ave.

- Clean the dataset as needed. Missing values, NaN values, Null values,

All Missing values, NaN values, Null values were replaced with its mean of the column and has same Partner Name in the row.

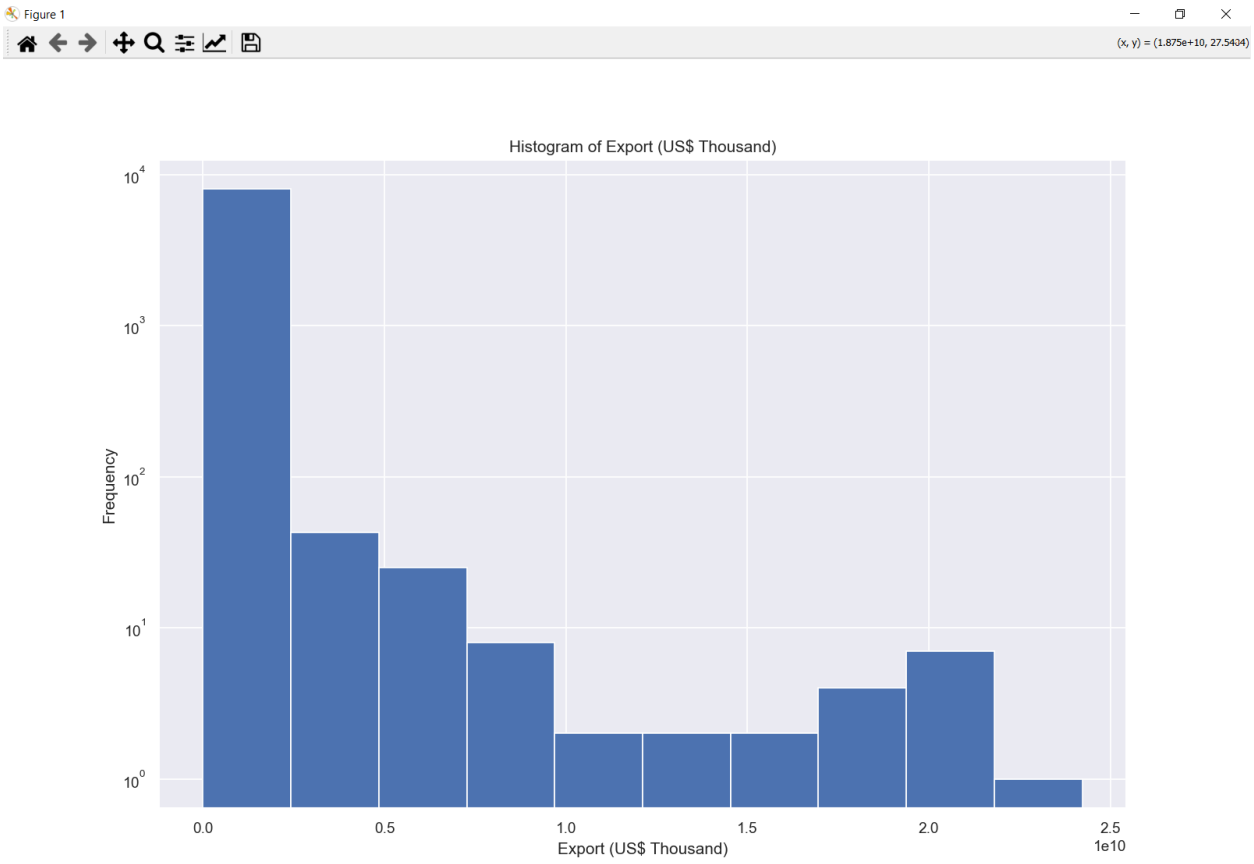
incorrect values, outliers

We used histograms, line charts, XY charts, boxplots to spot negative values, zero values or high values. The ranges were also checked for each column. The outliers were either deleted or replaced with a value within its column range.

any other issues you find. incorrect datatypes

The datatypes were checked and changed to either integer or floating depending on any possible calculations to be performed.

- Use at least 5 visualizations as part of your cleaning methods.



Log scale

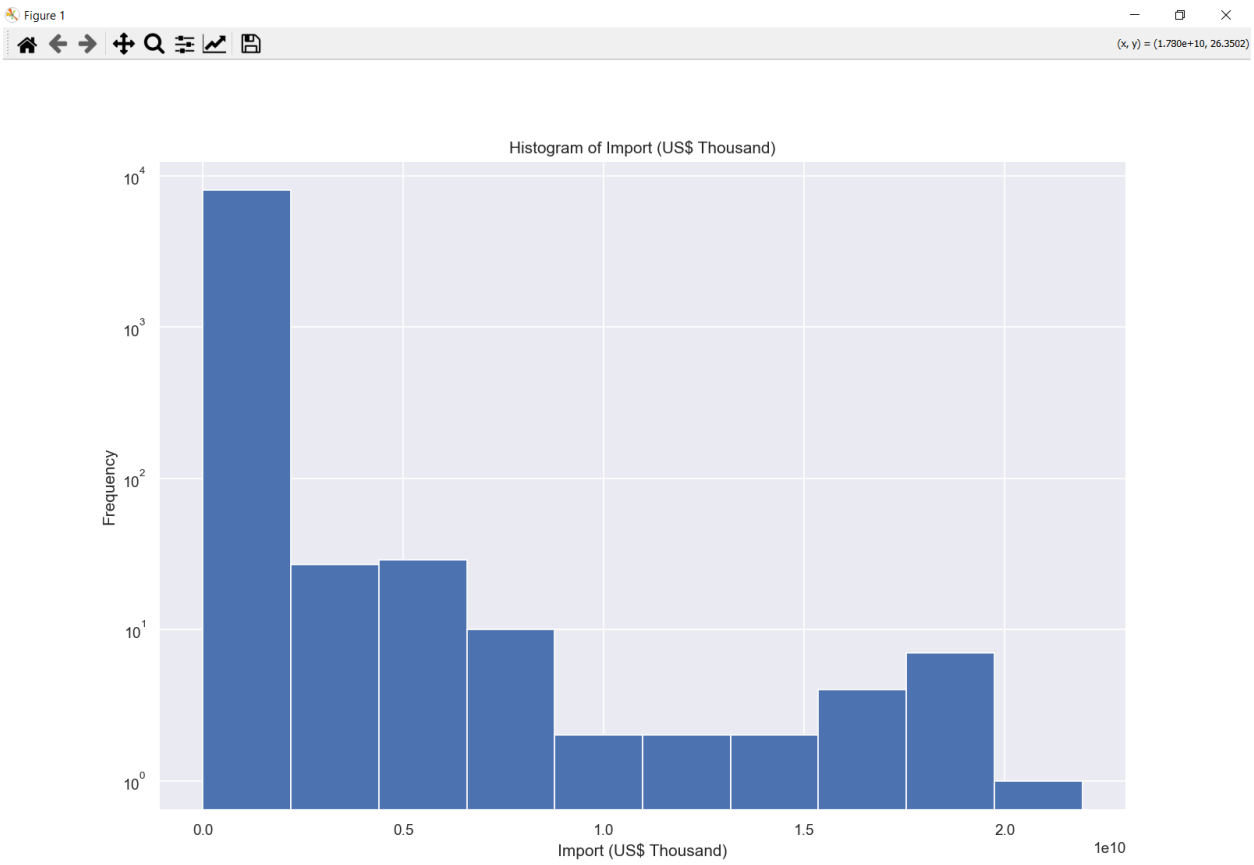
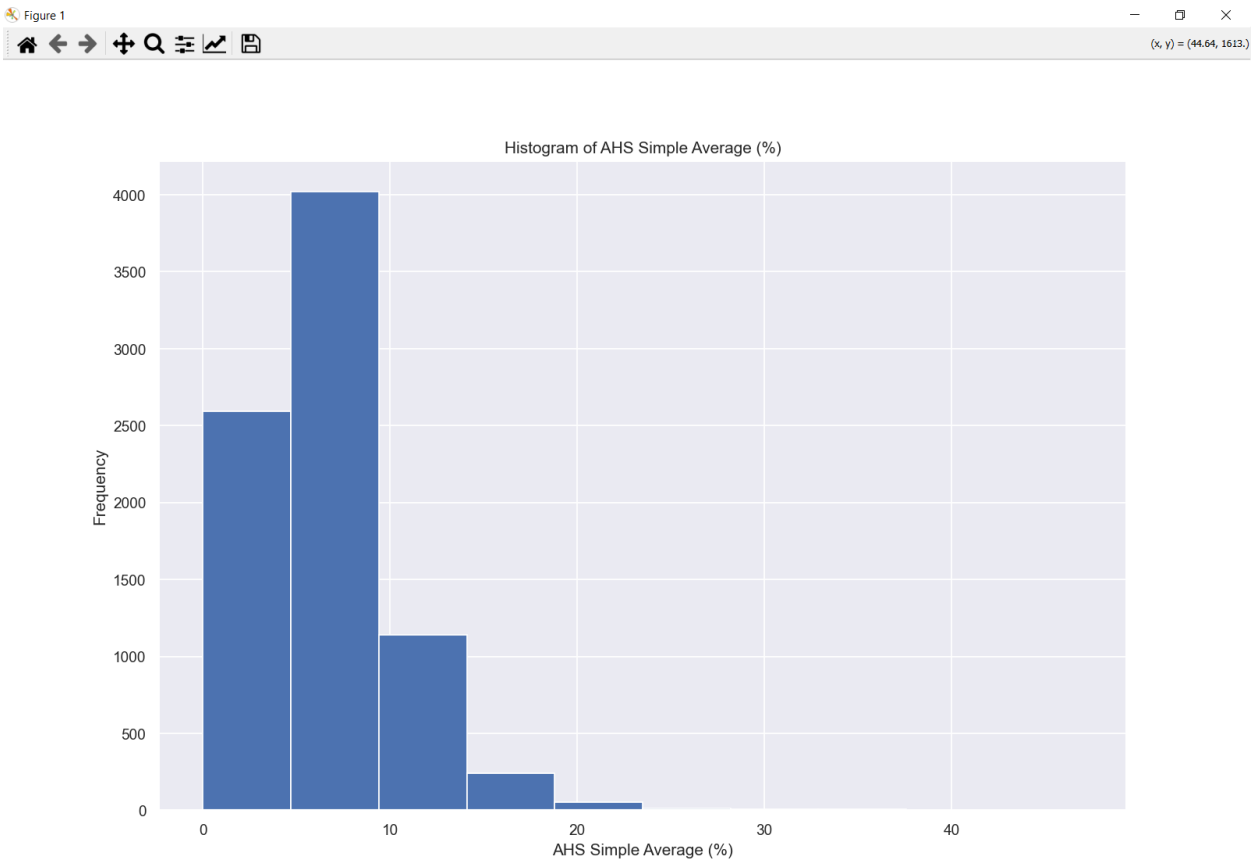
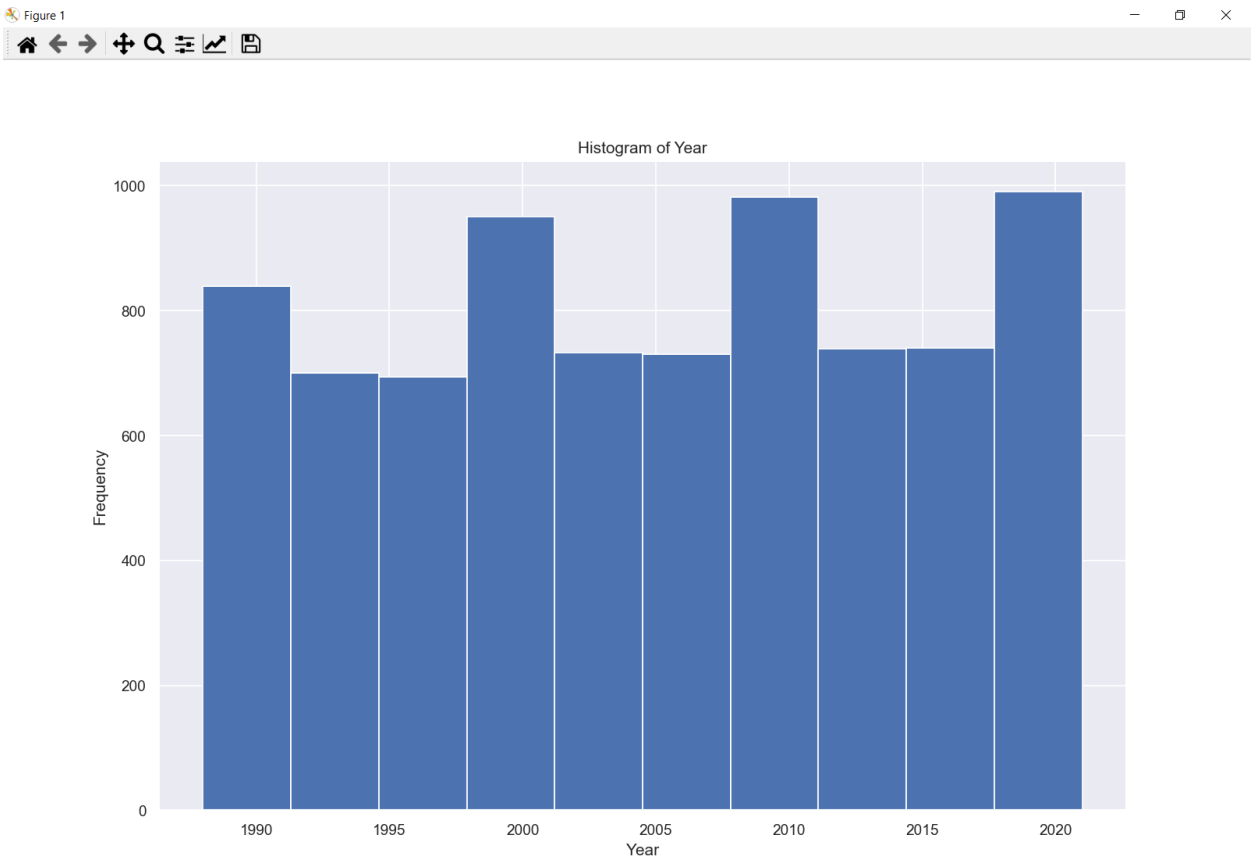


Figure 1
(x, y) = (1.730e+10, 26.3502)
Histogram of Import (US$ Thousand)
Frequency
10^4
10^3
10^2
10^1
10^0
0.0
0.5
1.0
1.5
2.0
Import (US$ Thousand)
1e10

Log scale

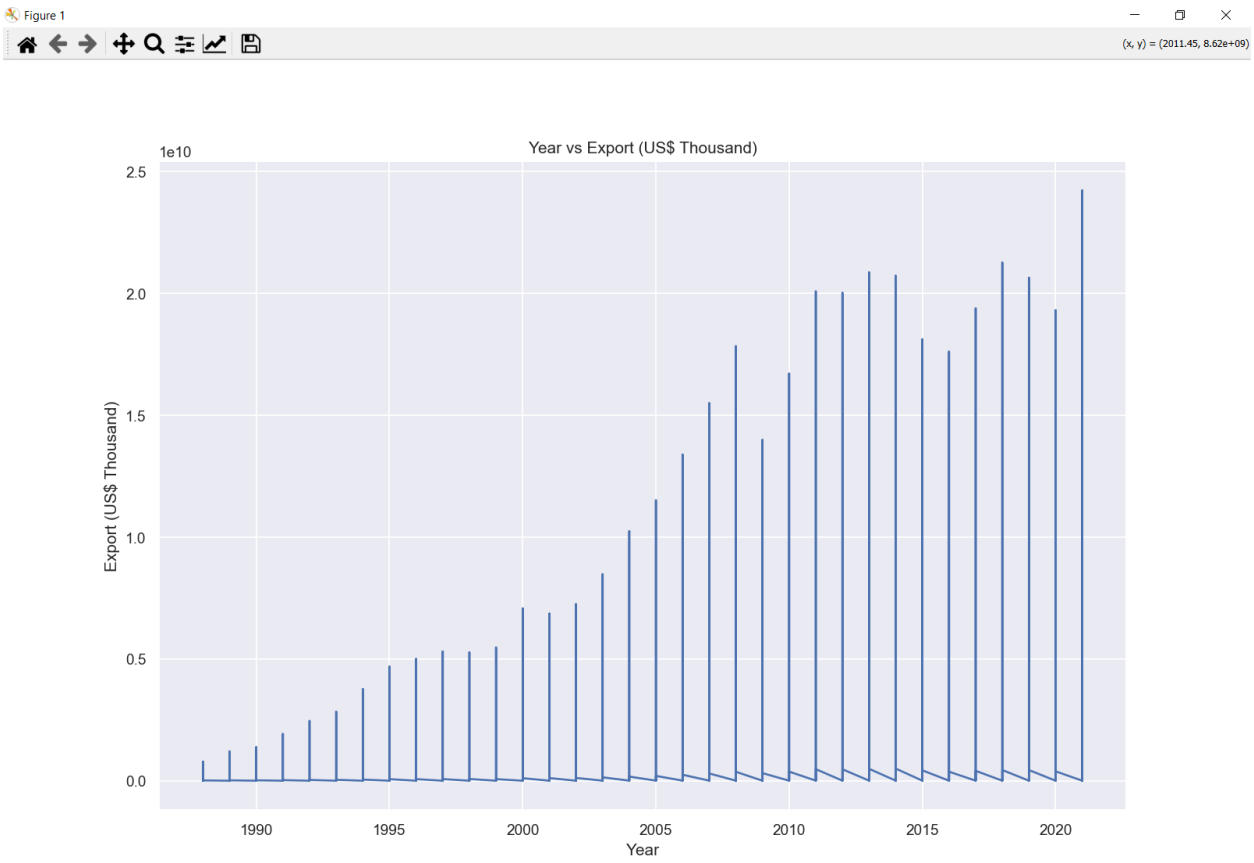


Linear scale

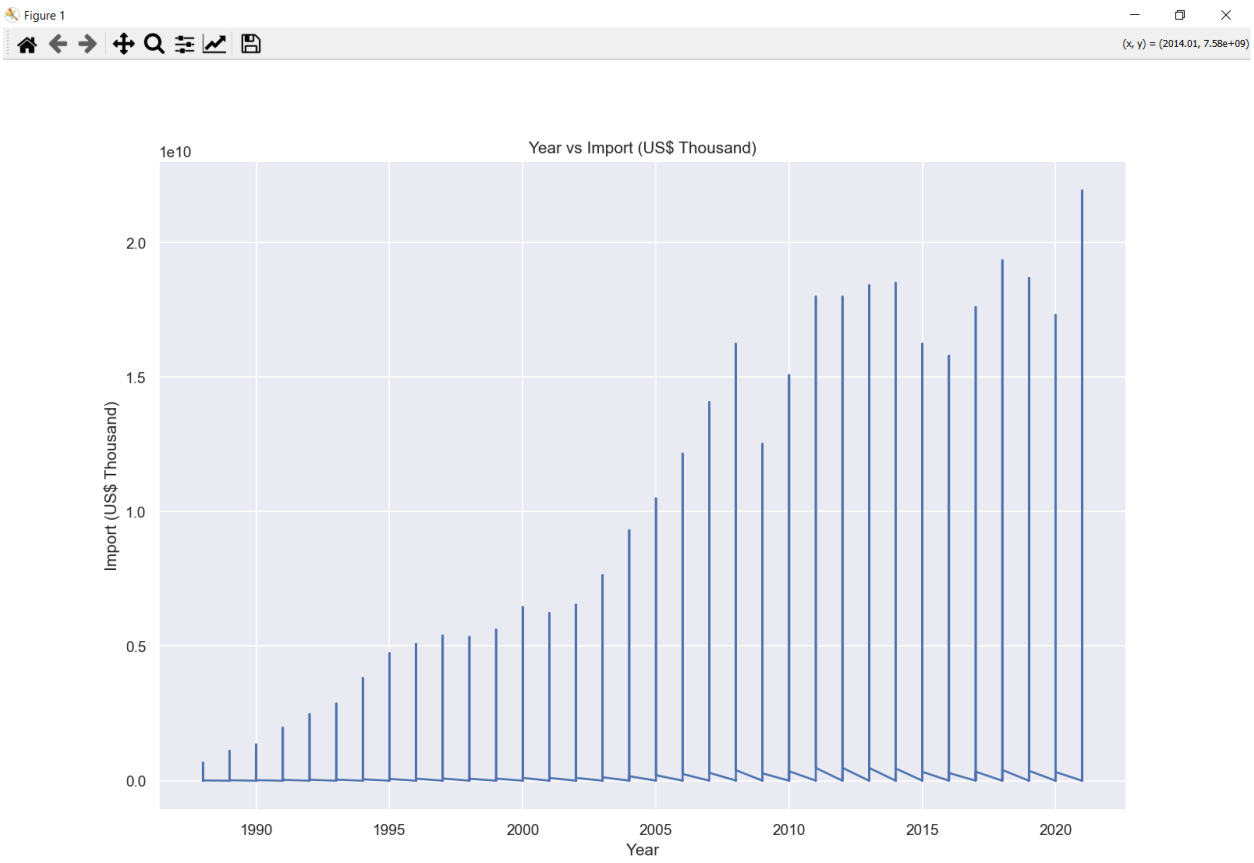


Linear scale

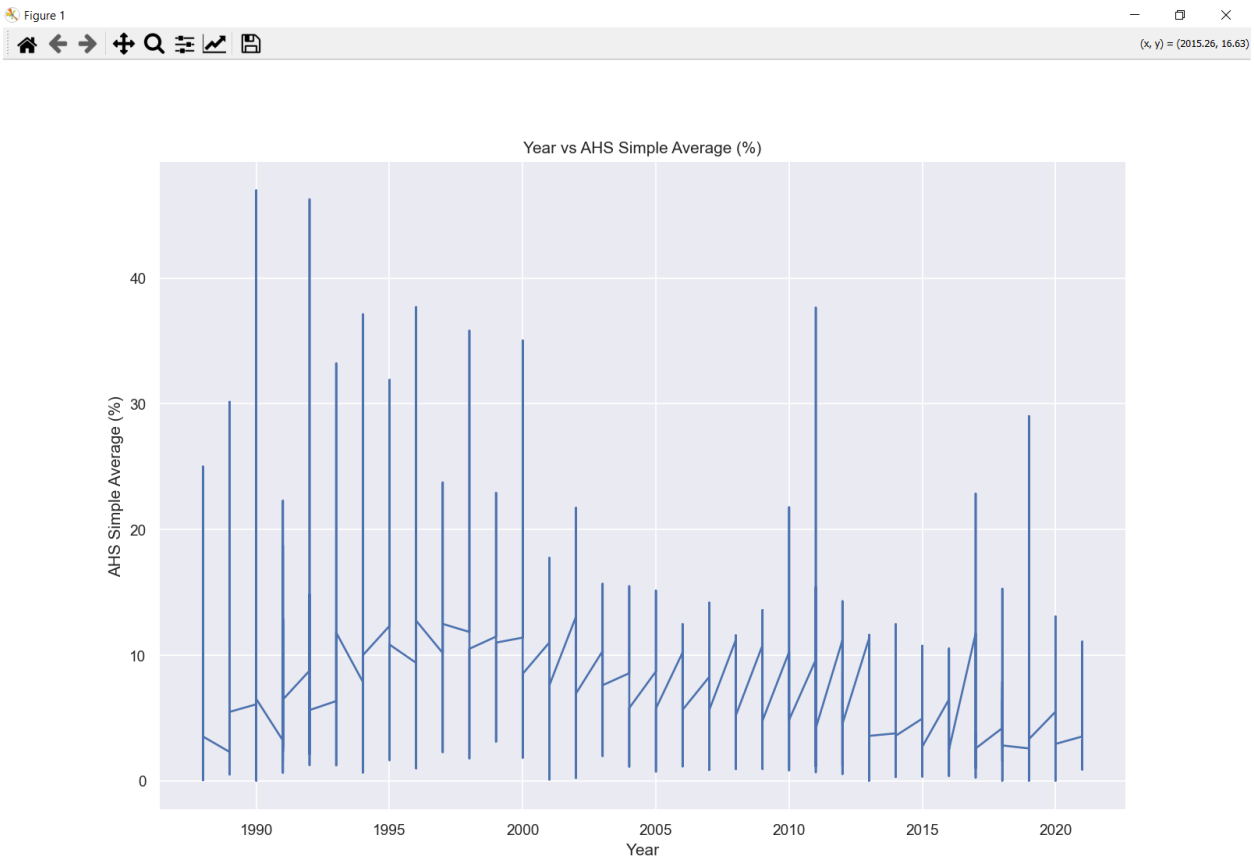
It looks like there is more data on the decade marks than other years.



Linear scale



Linear scale



Linear scale

- Once the dataset is cleaned and ready to use on KMeans, then remove the "Partner Name" column and save it as a LIST.

(Hint - do not yet remove the "label" column (Survived) as you will clean this first and then remove but SAVE it later. You cannot run k means with labeled data, but you can compare your labels to the clusters that you discover using k means. Therefore, keep the label on the dataset until the dataset is completely cleaned. Then remove the cleaned label and save it.

Include an "after" image of the data once it is ready to apply to k means.

```
Partner Name column has been saved as a list


    Year  Export (US$ Thousand)  Import (US$ Thousand)
0   1988                3498.10                328.49
1   1988              213030.40              54459.52
2   1988              375527.89             370702.76
3   1988                 366.98                  4.00
4   1988               30103.56              47709.30
5   1988               67924.46               3284.01
6   1988              104759.21              24964.14
7   1988             2945350.25             7091823.87
8   1988             1136421.71             1928596.45
9   1988               14406.52               2173.80
10  1988            10508173.98            14350888.96
11  1988            22046961.12            14273975.93
12  1988               37299.67               73592.16
13  1988               66486.37               17352.43
14  1988               42212.93               24547.18
15  1988              801086.80              221256.38
16  1988             1278230.45              376577.93
17  1988              335294.13              458407.80
18  1988              356568.82               58504.92
19  1988            30638909.79            24915272.75
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8096 entries, 0 to 8095
Data columns (total 3 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Year                   8096 non-null   int64
 1   Export (US$ Thousand)  8096 non-null   float64
 2   Import (US$ Thousand)  8096 non-null   float64
dtypes: float64(2), int64(1)
memory usage: 189.9 KB
None
```

The cleaned dataset has 3 columns. Year, Export and Import. There are 34 years of data for countries around the world.

(b) Visualize the Data

Read your cleaned and prepared dataset into Python and print the DF. Include an image of this in your report with a description.

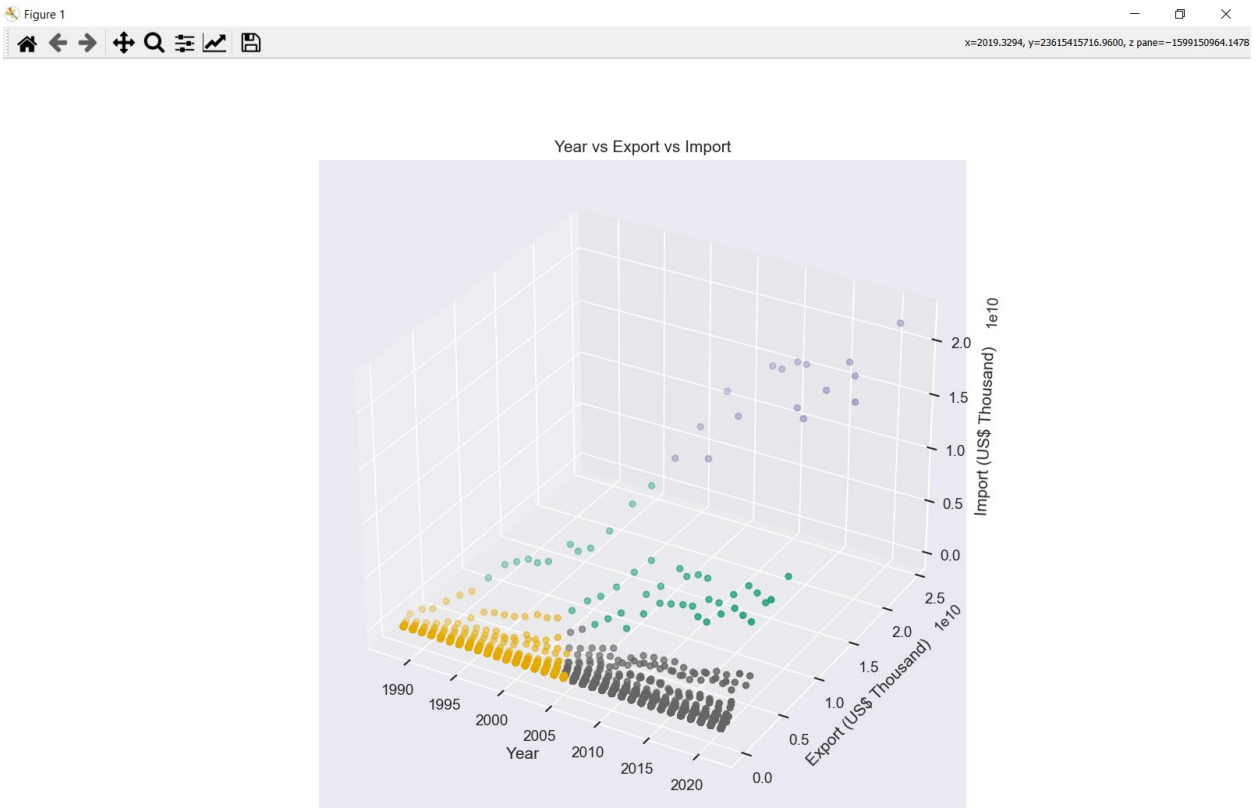
```
Partner Name column has been saved as a list


    Year  Export (US$ Thousand)  Import (US$ Thousand)
0   1988                3498.10                328.49
1   1988              213030.40              54459.52
2   1988              375527.89             370702.76
3   1988                 366.98                  4.00
4   1988               30103.56              47709.30
5   1988               67924.46               3284.01
6   1988              104759.21              24964.14
7   1988             2945350.25            7091823.87
8   1988             1136421.71            1928596.45
9   1988               14406.52               2173.80
10  1988            10508173.98           14350888.96
11  1988            22046961.12           14273975.93
12  1988               37299.67              73592.16
13  1988               66486.37              17352.43
14  1988               42212.93              24547.18
15  1988              801086.80             221256.38
16  1988             1278230.45             376577.93
17  1988              335294.13             458407.80
18  1988              356568.82              58504.92
19  1988            30638909.79           24915272.75
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8096 entries, 0 to 8095
Data columns (total 3 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Year                   8096 non-null   int64
 1   Export (US$ Thousand)  8096 non-null   float64
 2   Import (US$ Thousand)  8096 non-null   float64
dtypes: float64(2), int64(1)
memory usage: 189.9 KB
None
```

We start our KMeans analysis with 3 column clean dataset.

Create a visualization of the dataset in 3D using Python and include it in the report. Use any method you wish to place the centroids on to the visualization.

Figure 19 3D Plot of Year vs Export vs Import



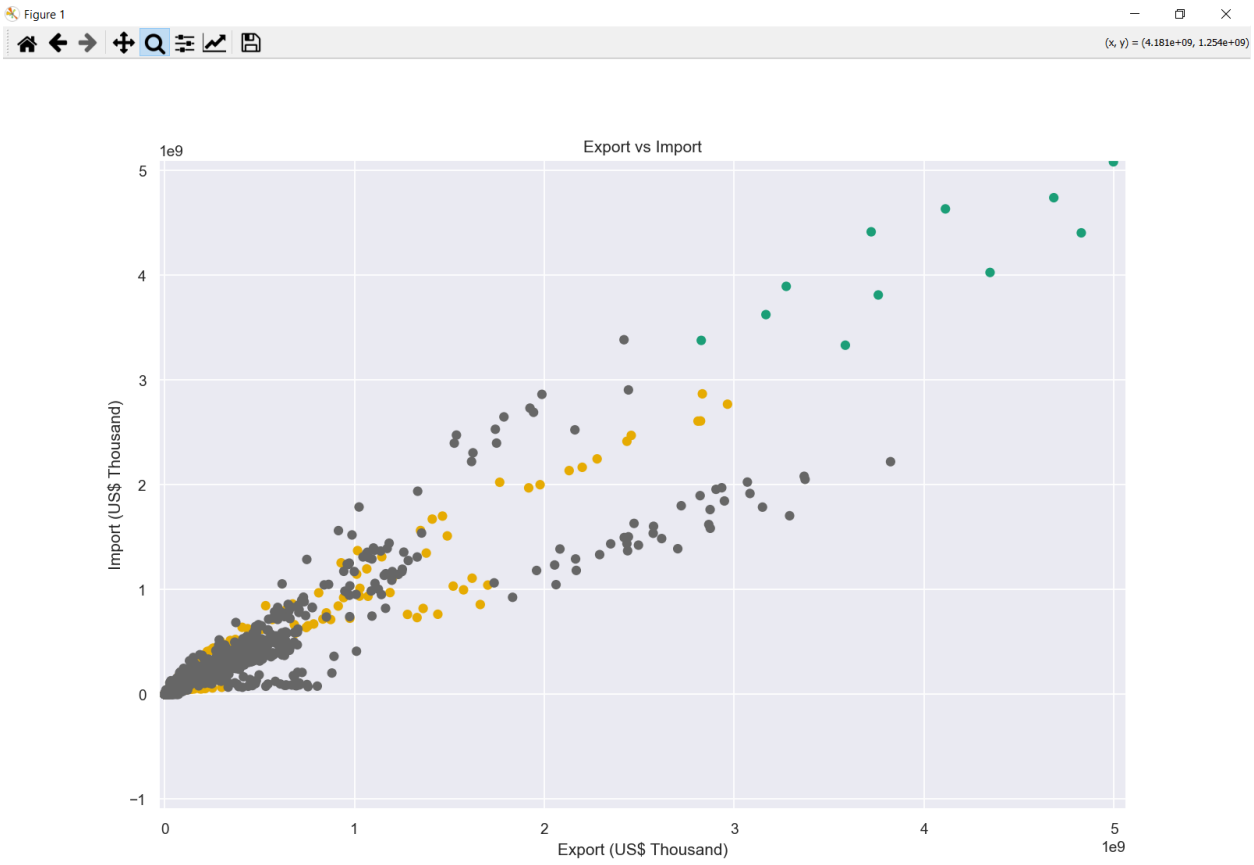
Note: 3

Cleaned dataset from Kaggle

Linear scale

4 clusters are shown in different colors.

Figure 20 Import vs Export



Note: 4

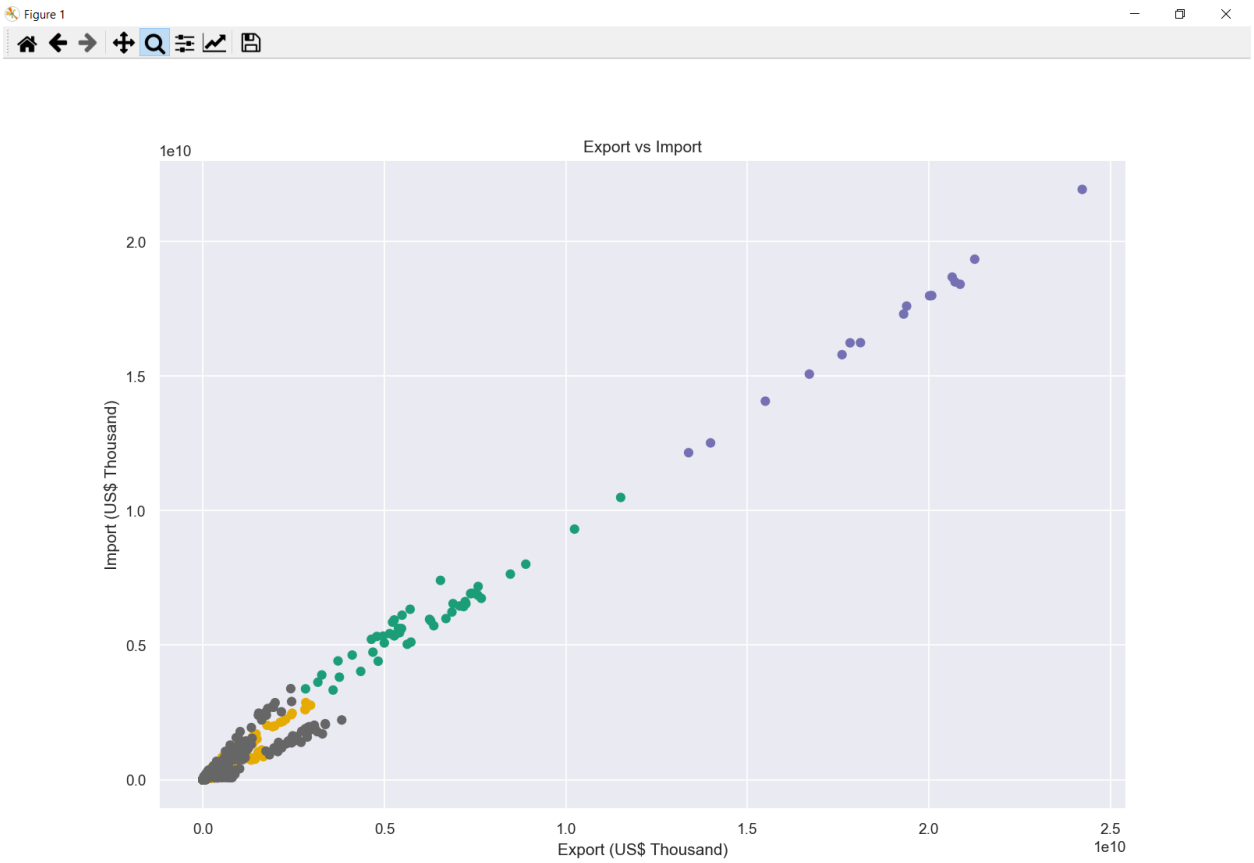
Cleaned dataset from Kaggle

Linear scale

Linear scale

4 clusters are shown in different colors.

Figure 21 Import vs Export



Note: 5

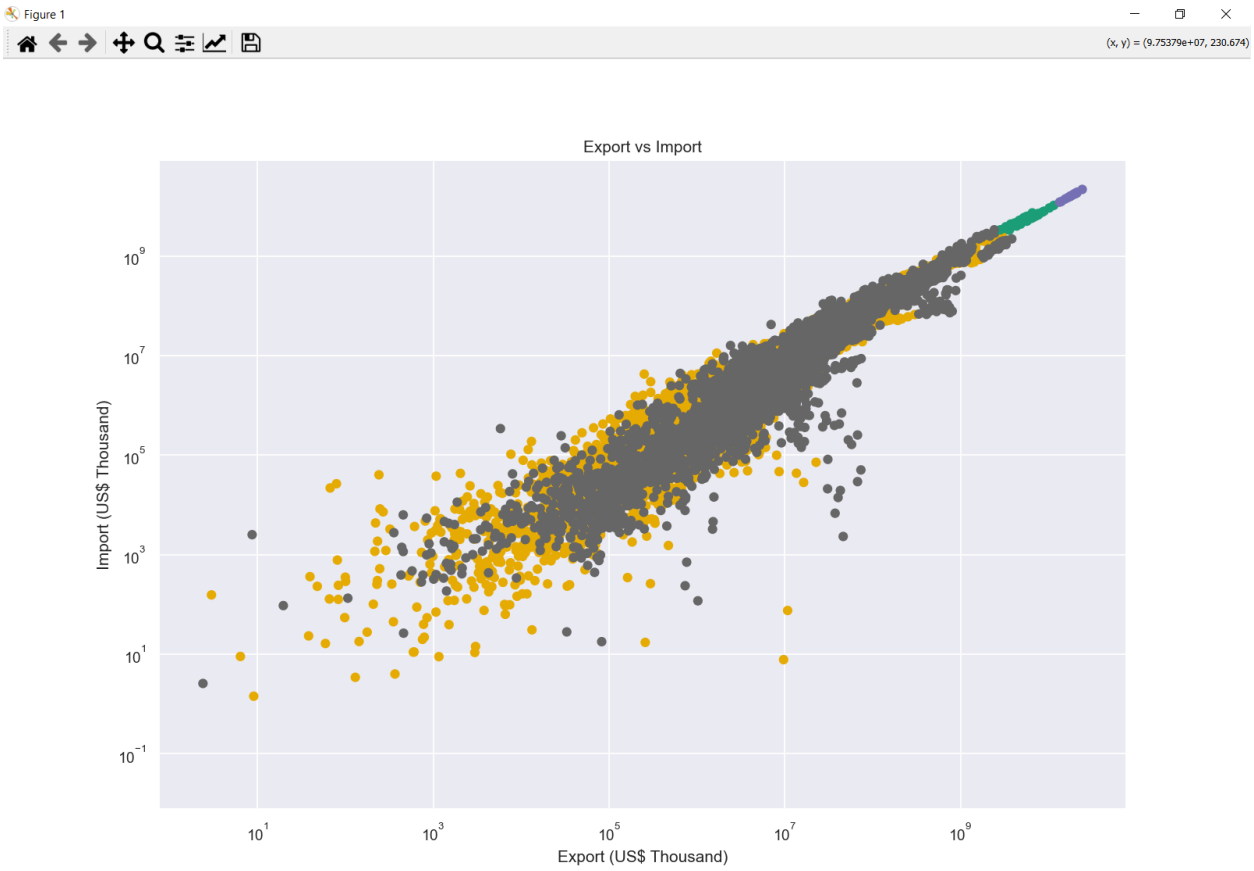
Cleaned dataset from Kaggle

Linear scale

Linear scale

4 clusters are shown in different colors.

Figure 22 Import vs Export



Note: 6

Cleaned dataset from Kaggle

Log scale

Log scale

4 clusters are shown in different colors.

What do you see?

In Figure 19, it is a 3D plot of United States Year versus Exports versus Imports to other countries. The Year axis spans 34 years. Import and Export cover all the countries around the world. We are getting a time lapse of how much trade, import and export, has the United States achieved in the last 30 years to versus other countries around the world. In the cluster with purple data points, a few countries have the

fastest trade growth with the US. The 3D plot does not include the name of the countries and therefore cannot tell which are those fastest trade countries. Much of the trade is on the lower trade numbers. Over time, the trade among countries with the United States has steadily grown which accounts for the US prosperity.

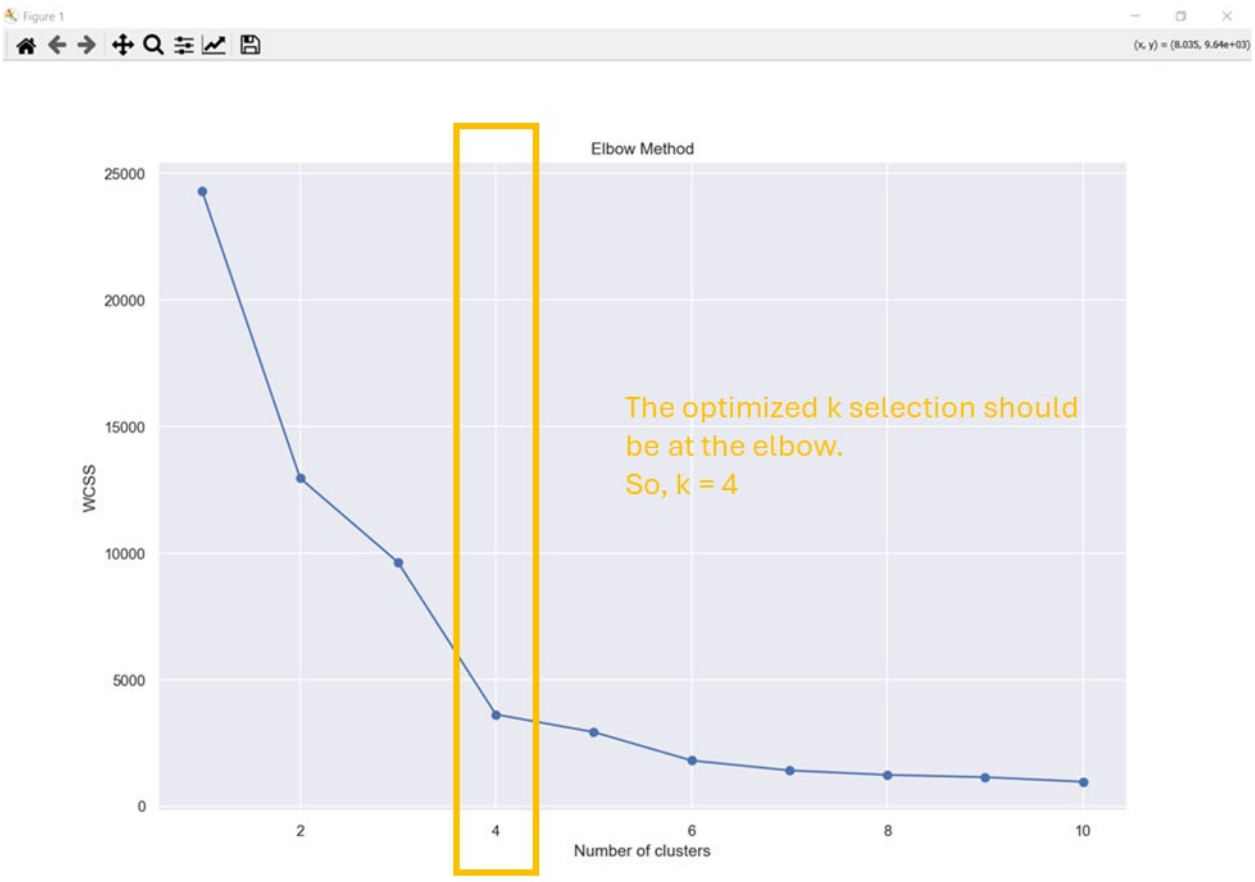
In the 3D plot there are 4 clusters shown in different colors. It broke down into how much trade different countries are doing with the US across a 30 year span. The more trade in exports seem to influence a higher trade in imports and vice versa. The countries with purple color have embraced trade with the US. While the other countries are smaller trade partners, they are growing in trade as well but at a slower pace.

In the 2D plots there are also 4 clusters shown in different colors. A 2D chart is easier to understand than a 3D chart sometimes. From Figures 20 through 22, the Export versus Import is pretty linear. I would of thought countries would try to export more than import but is not true. The slope of growth looks like a 45 degree angle. So, imports and exports are growing one to one. Once again the purple colored data points are at far right top which is a cluster.

The 3D plot may not need to be in 3D. Exports and Imports do not overlap each other, which is why there are no data points floating in space. The data points are all in one plane for Year, Exports and Imports.

(c) Apply KMeans Using Sklearn in Python

Perform, illustrate, and discuss your k means results using k = 2, k = 3, and k = 4



Elbow Method chart for optimal k

Seek out the plot's "elbow" point. This number represents the ideal number of clusters. Several techniques can be used to determine the appropriate value of (k) for KMeans clustering. The Elbow Method is one popular strategy. The within-cluster sum of squares (WCSS) is plotted against the number of clusters using this method, which entails running KMeans for a range of values of (k). The ideal (k) is usually thought to be the "elbow" point, where the rate of decrease abruptly slows.

Looking at the chart above, I choose k = 4.

For each value of k. print and include the cluster centers(centroids), the cluster labels.

```
Cluster centroids:
 [[-1.74267555e-03 -3.97568625e-02 -3.91874649e-02]
 [ 7.82074060e-01  1.78419964e+01  1.75864634e+01]]
```

k = 2

```
Cluster summary:
                Year  Export (US$ Thousand)  Import (US$ Thousand)
Cluster
0        2004.89131           1.026493e+08           9.496591e+07
1        2012.50000           1.785530e+10           1.608714e+10
```

k = 2

See examples of using your clusters to predict a new data vector in the section below.

--

```
Cluster centroids:
 [[ 5.03510423e-01  5.77087862e+00  6.17282619e+00]
 [ 8.85090047e-01  1.87210225e+01  1.84390350e+01]
 [-4.83544409e-03 -7.25077092e-02 -7.43983368e-02]]
```

k = 3

```
Cluster summary:
                Year  Export (US$ Thousand)  Import (US$ Thousand)
Cluster
0        2009.795918           5.871335e+09           5.731276e+09
1        2013.500000           1.872798e+10           1.686070e+10
2        2004.861288           7.013495e+07           6.301824e+07
```

k = 3

#--

```
Cluster centroids:
 [[ 0.45439555  5.88874539  6.31022499]
 [ 0.88509005 18.72102252 18.43903496]
 [-0.84761882 -0.10472115 -0.10315921]
 [ 0.88117239 -0.03710564 -0.04265058]]
```

k = 4

```
Cluster summary:
                Year  Export (US$ Thousand)  Import (US$ Thousand)
Cluster
0        2009.319149           5.988350e+09           5.855941e+09
1        2013.500000           1.872798e+10           1.686070e+10
2        1996.680194           3.815408e+07           3.692280e+07
3        2013.461970           1.052814e+08           9.182374e+07
```

k = 4

#--

examples of using your clusters to predict a new data vector.


```
Cluster summary:
                Year  Export (US$ Thousand)  Import (US$ Thousand)
Cluster
0        2009.319149           5.988350e+09           5.855941e+09
1        2013.500000           1.872798e+10           1.686070e+10
2        1996.680194           3.815408e+07           3.692280e+07
3        2013.461970           1.052814e+08           9.182374e+07
```


We are given a cluster such as 4 clusters above. Generate a new vector within the given cluster. The new vector just needs to fit in the centroid on the cluster. We can generate a new data vector that belongs to a specific cluster by sampling within the range of the cluster's centroid and its standard deviation.

#--

k = 0

```
Generated new data vector for cluster 0: [1.99668019e+03 3.81540800e+07 3.69228000e+07]
```

--

k = 1

```
Generated new data vector for cluster 1: [2.013500e+03 1.872798e+10 1.686070e+10]
```

--

k =2

```
Generated new data vector for cluster 2: [2.00931915e+03 5.98835000e+09 5.85594100e+09]
```

--

k=3

```
Generated new data vector for cluster 3: [2.01346197e+03 1.05281400e+08 9.18237400e+07]
```

With this exercise, given a cluster, we can now predict a new vector inside the cluster centroid. Also, from previous assignment, given a new vector, we can predict which cluster the new vector belongs in.

(d) Technical Results

The 3D plot in Figure 19 shows the trade relationship between the United States and other countries over 34 years. The three axes represent Year, Exports, and Imports. The chart tracks how much the United States has traded with different countries worldwide over time. The trade data is grouped into

four color-coded clusters. Each cluster represents different levels of trade activity between the US and other countries.

The purple cluster represents countries with the fastest trade growth with the US. However, the chart does not label specific country names, so it is unclear which countries have the fastest-growing trade. Most trade activity falls within lower trade values, but trade has steadily increased over time. This steady increase in trade has contributed to US economic prosperity.

The 3D plot shows that when exports increase, imports also tend to increase. Countries that trade more with the US in exports also import more from the US. The purple-colored countries have embraced trade with the US the most. Other countries trade at a lower level but are also growing, just at a slower pace.

Figures 20 through 22 present 2D plots, also using four color-coded clusters. A 2D chart is sometimes easier to understand than a 3D chart. These 2D plots show a clear trend in trade between the US and other countries. The relationship between Exports and Imports appears linear. The trade growth follows a 45-degree slope, meaning that exports and imports are increasing at the same rate. Many people might expect countries to export more than they import, but the data shows that imports and exports grow together.

The purple cluster is positioned at the top right of the 2D charts, showing that these countries have the highest trade volume with the US. The overall trend suggests that as trade increases, both exports and imports rise together. This balanced growth in trade supports economic stability and international business relations.

Supervised Learning with Decision Trees

(a) Format the Labeled Data you plan to use with Decision Tree Modeling

Before image of United States Import Export dataset

	A	B	C	D	E
1	Partner Name	Year	Export (US$ Thousand)	Import (US$ Thousand)	AHS Simple Average (%)
2	Aruba	1988	3498.1	328.49	2.8
3	Afghanistan	1988	213030.4	54459.52	0.88
4	Angola	1988	375527.89	370702.76	2.02
5	Anguila	1988	366.98	4	3.71
6	Albania	1988	30103.56	47709.3	1.84
7	Andorra	1988	67924.46	3284.01	6.74
8	Netherlands Antilles	1988	104759.21	24964.14	1.32
9	United Arab Emirates	1988	2945350.25	7091823.87	6.69
10	Argentina	1988	1136421.71	1928596.45	3.87
11	Antigua and Barbuda	1988	14406.52	2173.8	1.17
12	Australia	1988	10508173.98	14350888.96	11.83
13	Austria	1988	22046961.12	14273975.93	6.37
14	Burundi	1988	37299.67	73592.16	0.54
15	Benin	1988	66486.37	17352.43	12.71
16	Burkina Faso	1988	42212.93	24547.18	0.52
17	Bangladesh	1988	801086.8	221256.38	3.6
18	Bulgaria	1988	1278230.45	376577.93	7.7
19	Bahrain	1988	335294.13	458407.8	1.47
20	Bahamas, The	1988	356568.82	58504.92	1.62
21	Belgium-Luxembourg	1988	30638909.79	24915272.75	13.08
22	Belize	1988	22329.16	1932.65	3.84
23	Bermuda	1988	360059.94	620103.8	3.55
24	Bolivia	1988	70153.06	33646.21	1.84
25	Brazil	1988	3157960.26	7733502.96	5.45
26	Barbados	1988	46120.11	4773.91	1.67
27	Brunei	1988	198481.32	1523443.35	7.1
28	Bhutan	1988	6607.45	63.31	5.27
29	Bunkers	1988	625205.47	154879	9.831290323
30	Central African Republic	1988	20345.41	8310.05	2.58
31	Canada	1988	12211696.04	13379048.71	8.94
32	Cocos (Keeling) Islands	1988	2728.03	295.1	25
33	Switzerland	1988	23739550.61	17088873.08	8.86
34	Chile	1988	987036.13	2117305.65	3.65
35	China	1988	13575267.56	14251274.37	8.72
36	Cote d'Ivoire	1988	208122.47	463490.14	3.25
37	Cameroon	1988	167900.07	209905.15	1.97
38	Congo, Rep.	1988	39611.91	123629.56	3.25
39	Cook Islands	1988	5870.56	1144.8	7.47
40	Colombia	1988	1011212.56	1210433.39	2.37
41	Comoros	1988	2432.41	2617.6	2.25

My labeled dataset has previously been cleaned. There are over 8000 rows of data.

Here is a step by step explanation of how decision trees work

We will perform these tasks below.

1. Dataset Preparation: You start with a dataset that you want to use for training and evaluating your model.

My labeled dataset has been reduced to 5 columns above. Both qualitative and quantitative data are present. In order to do a decision tree, it needs to be a label or target which can be any of the existing columns or from a new column. To see any patterns in the data, four targets were created such as

1.1. Classifies trade balance into 'Trade Surplus', 'Trade Deficit', or 'Balanced Trade'

1.1.1. 1. Trade Balance Category

Based on Net Trade Balance (Export - Import)

Labels:

"Trade Surplus" (Exports > Imports)

"Trade Deficit" (Exports < Imports)

"Balanced Trade" (Exports ≈ Imports)

1.2. Categorizes trade intensity into 'Low Trade', 'Medium Trade', and 'High Trade'.

1.2.1. 2. Trade Intensity

Based on Total Trade Volume (Exports + Imports)

Labels:

"Low Trade" (Bottom 33% of total trade volume)

"Medium Trade" (Middle 33%)

"High Trade" (Top 33%)

1.3. Categorizes AHS Simple Average Tariff into 'Low Tariff', 'Moderate Tariff', and 'High Tariff'.

1.3.1. Tariff Impact Level

Based on AHS Simple Average (%)

Labels:

"Low Tariff" (Bottom 33% of tariff values)

"Moderate Tariff" (Middle 33%)

"High Tariff" (Top 33%)

1.4. Categorizes export dependence as 'Export-Driven', 'Balanced', or 'Import-Driven'.

1.4.1. Export Dependence

Based on Export Share (Export / (Export + Import))

Labels:

"Export-Driven" (Export Share > 60%)

"Balanced" (Export Share 40%-60%)

"Import-Driven" (Export Share < 40%)

We can easily switch between targets to see the outcome and determine how parameters effect results.

All the non-numeric columns will be removed and labels saved, and remaining dataset is all numeric for the next steps.

We will use Export Dependence as our first target.

```
create target classifications in dataset

      Year  Export (US$ Thousand)  Import (US$ Thousand)  \
0     1988           3.498100e+03           3.284900e+02
1     1988           2.130304e+05           5.445952e+04
2     1988           3.755279e+05           3.707028e+05
3     1988           3.669800e+02           4.000000e+00
4     1988           3.010356e+04           4.770930e+04
...    ...                    ...                    ...
8091  2021           1.330557e+09           1.310305e+09
8092  2021           1.196712e+09           1.088471e+09
8093  2021           3.823319e+09           2.219849e+09
8094  2021           6.991380e+08           4.723832e+08
8095  2021           4.951000e+08           4.350468e+08

      AHS Simple Average (%) Export Dependence
0                       2.80     Export-Driven
1                       0.88     Export-Driven
2                       2.02          Balanced
3                       3.71     Export-Driven
4                       1.84     Import-Driven
...                      ...               ...
8091                    3.84          Balanced
8092                    4.63          Balanced
8093                    6.45     Export-Driven
8094                    5.09          Balanced
8095                    3.22          Balanced

[8096 rows x 5 columns]


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8096 entries, 0 to 8095
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Year                    8096 non-null   int64
 1   Export (US$ Thousand)   8096 non-null   float64
 2   Import (US$ Thousand)   8096 non-null   float64
 3   AHS Simple Average (%)  8096 non-null   float64
 4   Export Dependence       8096 non-null   object
dtypes: float64(3), int64(1), object(1)
memory usage: 316.4+ KB
None
```

2. Train-Test Split: You split this dataset into two subsets:

The dataset is all numeric numbers.

Training Set: Used to train the model.

Training Data

```
The Training Data is
      Year  Export (US$ Thousand)  Import (US$ Thousand)  \
761   1991           1.055063e+07           1.386355e+07
4657  2008           7.638157e+04           8.212800e+03
5867  2012           7.224239e+09           6.617274e+09
2976  2001           1.420871e+05           1.745497e+05
1424  1994           3.760479e+06           2.869588e+06
...    ...                    ...                    ...
2780  2000           2.224496e+05           6.085805e+04
3677  2004           3.166244e+06           1.167673e+05
6452  2015           4.248666e+05           2.978247e+05
4960  2009           1.140880e+05           1.452443e+05
1804  1996           4.136071e+06           2.829998e+06

      AHS Simple Average (%) Export Dependence
761                     8.22          Balanced
4657                    5.70     Export-Driven
5867                    5.74          Balanced
2976                    6.42          Balanced
1424                    9.46          Balanced
...                      ...               ...
2780                   13.96     Export-Driven
3677                    2.86     Export-Driven
6452                    4.44          Balanced
4960                    3.20          Balanced
1804                    4.17          Balanced

[5667 rows x 5 columns]
```

Training Label

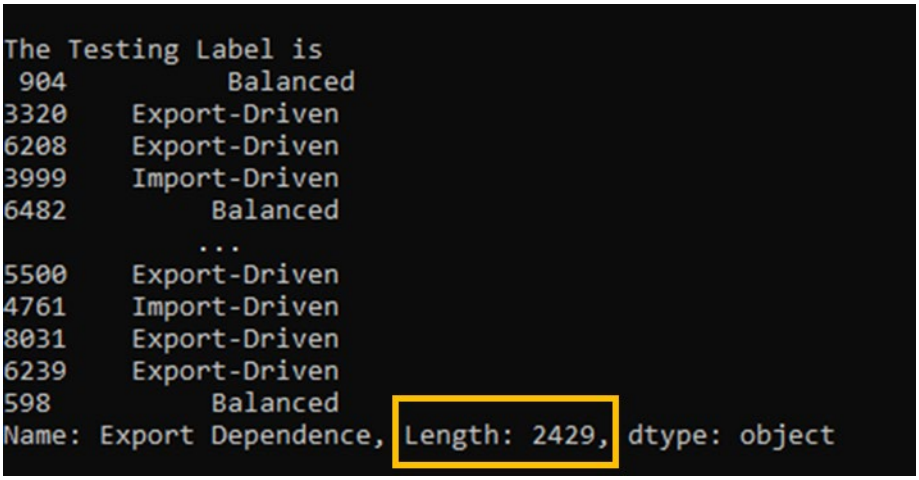
```
The Training Label is
 761          Balanced
4657    Export-Driven
5867          Balanced
2976          Balanced
1424          Balanced
             ...
2780    Export-Driven
3677    Export-Driven
6452          Balanced
4960          Balanced
1804          Balanced
Name: Export Dependence, Length: 5667, dtype: object
```

Test Set: Used to evaluate the model's performance.

Testing Data

```
The Testing Data is
       Year  Export (US$ Thousand)  Import (US$ Thousand)  \
904    1992            11271395.49            14982434.01
3320   2002             4668033.78             2657724.41
6208   2014              307999.13               42475.21
3999   2005             1604980.84             6628318.11
6482   2015              184950.70              129290.65
...     ...                    ...                    ...
5500   2011            24148857.44             1124883.11
4761   2008            29729382.25            86226572.91
8031   2021            18305116.32             3400026.67
6239   2014             7412382.73             4664523.42
598    1990             7518817.68             5044444.91

      AHS Simple Average (%) Export Dependence
904                    12.51          Balanced
3320                   11.33     Export-Driven
6208                    7.57     Export-Driven
3999                   10.66     Import-Driven
6482                    6.46          Balanced
...                      ...               ...
5500                    7.37     Export-Driven
4761                    4.36     Import-Driven
8031                    1.89     Export-Driven
6239                    2.04     Export-Driven
598                     9.72          Balanced

[2429 rows x 5 columns]
```

Testing Label



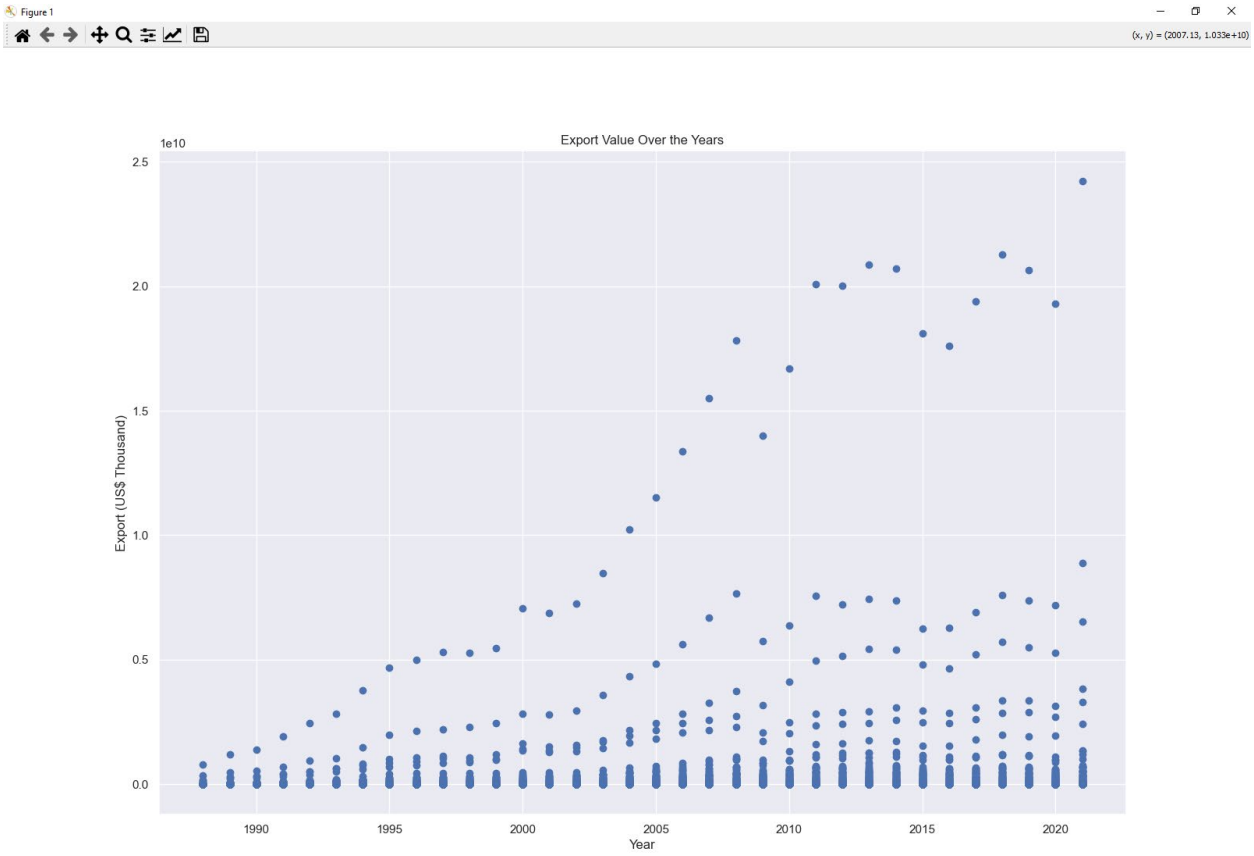
(b) Visualize the Data

Five visualizations

--

visual 1

Figure 23



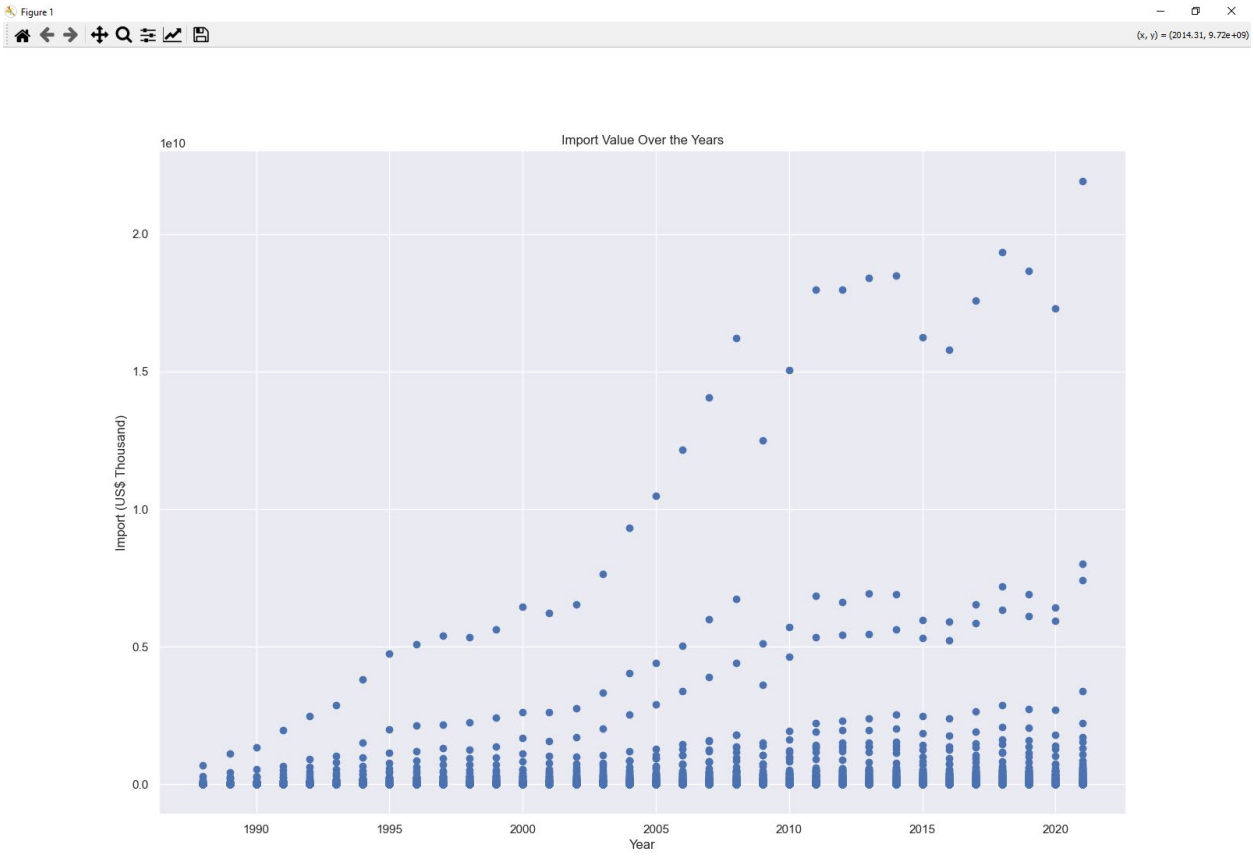
A scatter plot of the data with the x-axis representing the year and the y-axis representing the export value.

This visualization will show how the export value has changed over the years. Over time the export amounts have increased for all countries. It looks like the countries with the largest economies have increased their export rates more than the smaller countries. The largest countries with the higher trade also have a larger population which could be a demographic factor.

--

visual 2

Figure 24



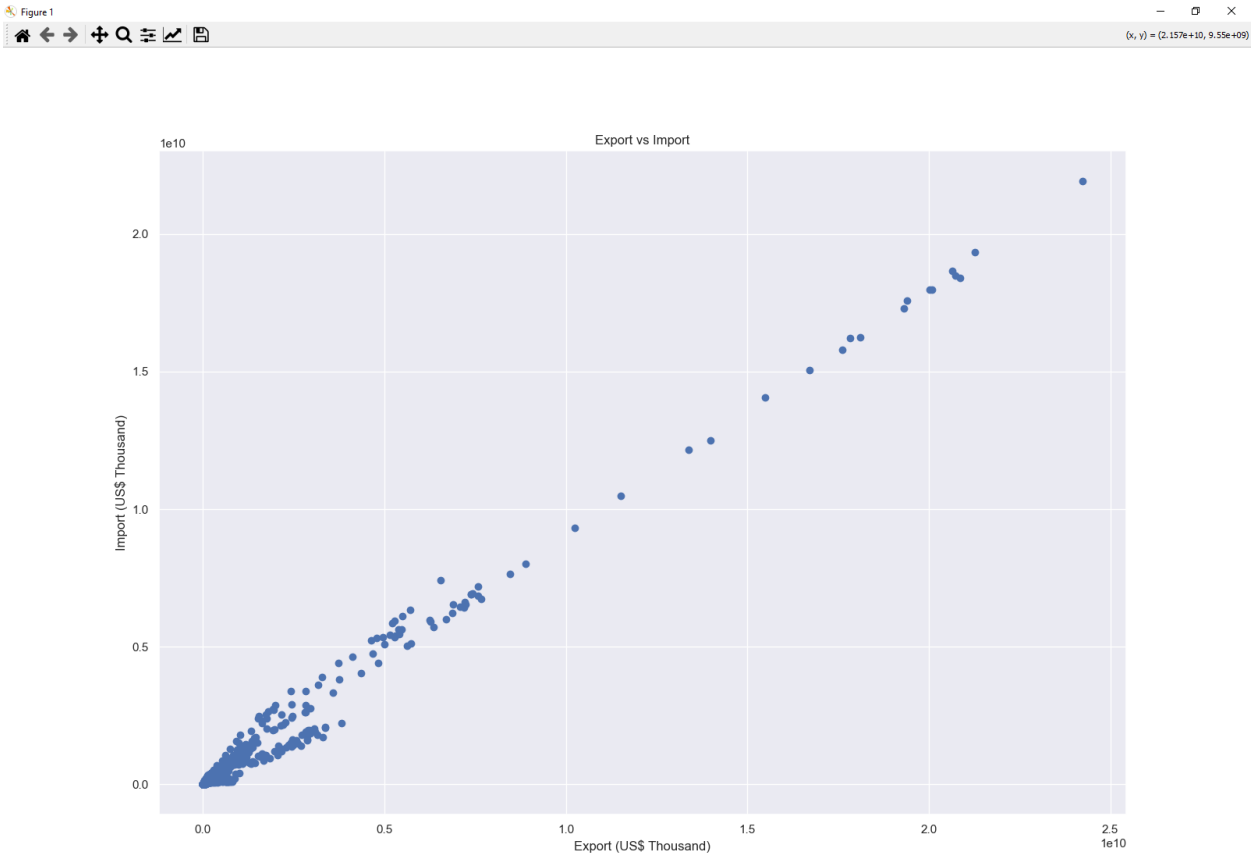
A scatter plot of the data with the x-axis representing the year and the y-axis representing the import value.

This visualization will show how the import value has changed over the years. Over time the import amounts have increased for all countries. It looks like the countries with the largest economies have increased their import rates more than the smaller countries.

--

visual 3

Figure 25



A scatter plot of the data with the x-axis representing the export value and the y-axis representing the import value.

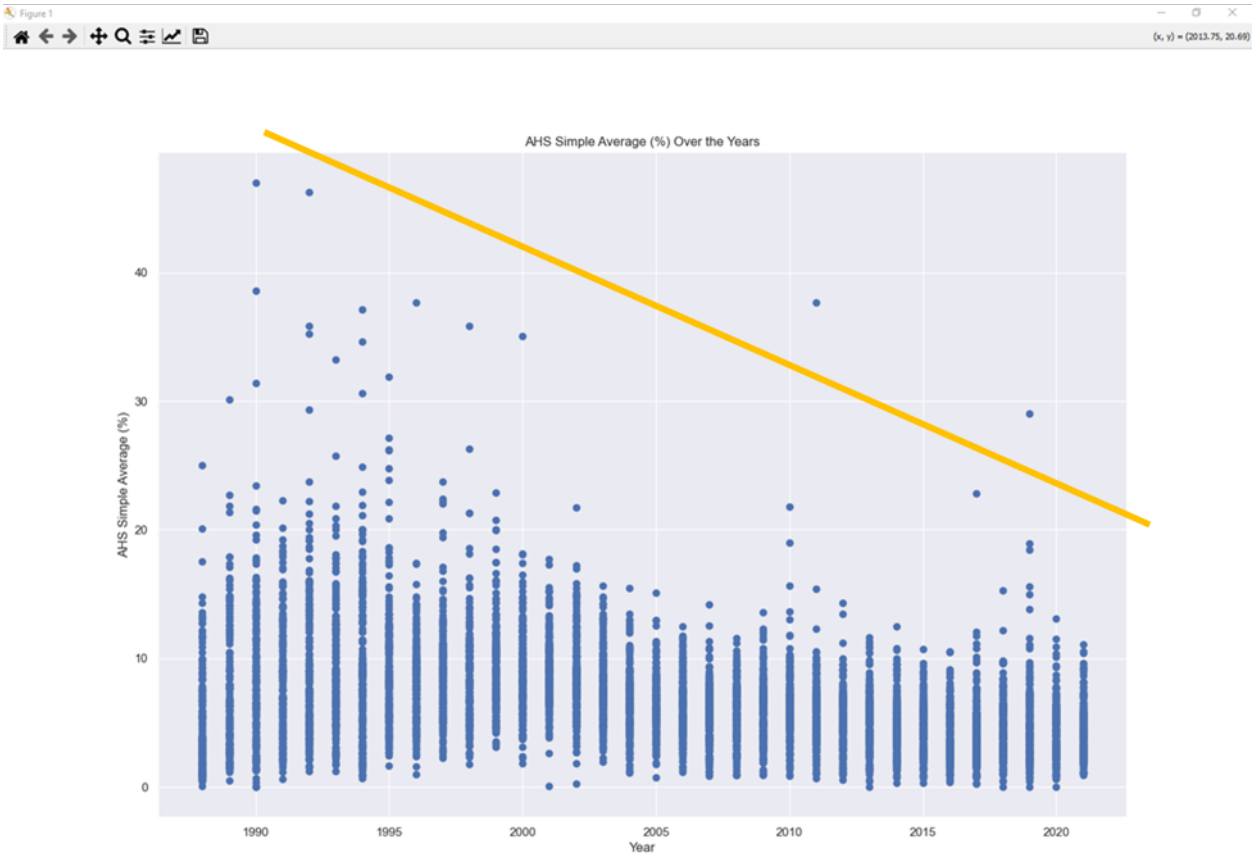
This visualization will show the relationship between the export and import values. Imports and Exports have a linear relationship. As exports increase so do imports and vice versa. For a country to increase its economy then it should start either importing or exporting more. Eventually, the other side of international trade will catch up as Figure 25 shows.

--

visual 4

Figure 26

The average tarif rate for other countries has been going down.
With the lowering of tariff rates all over the world, both imports and exports have been increasing too. It is an inverse correlation.



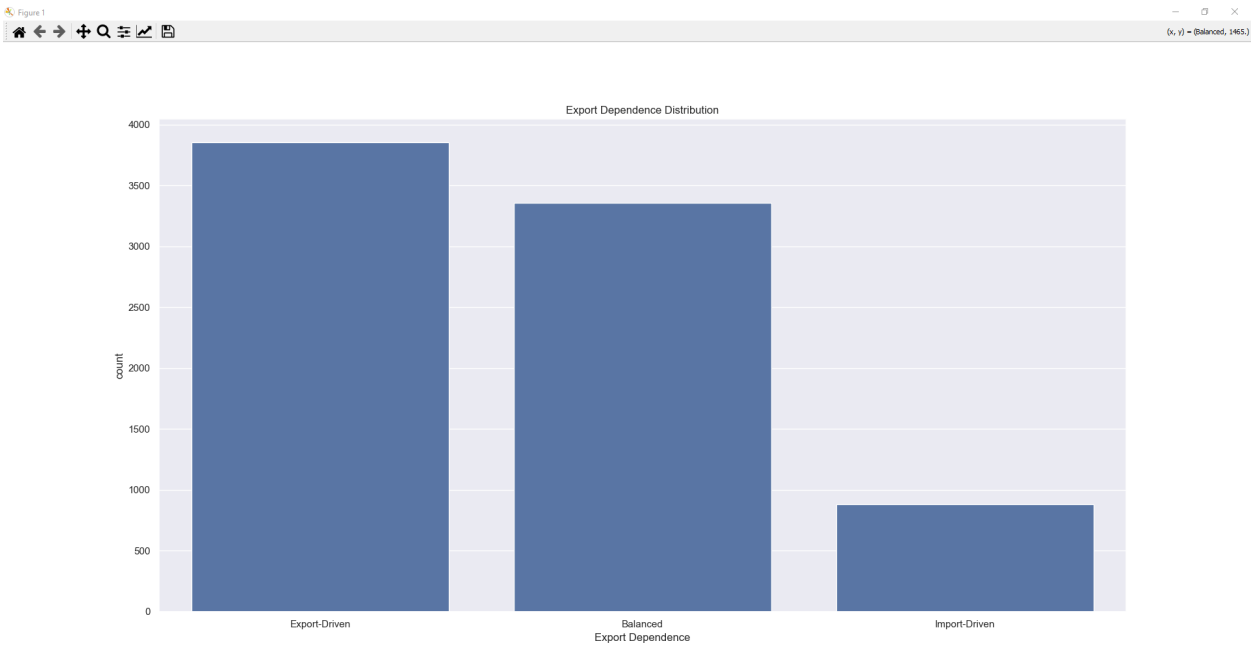
A scatter plot of the data with the x-axis representing the year and the y-axis representing the AHS Simple Average (%).

The average tariff rate imposed on other countries by the United States has be declining over the years. The lower tariff rates have caused an increase in imports and exports for all countries. Even small countries have benefitted from a lowering of tariff rates.

--

visual 5

Figure 27



Export Dependence

Based on Export Share (Export / (Export + Import))

Labels:

"Export-Driven" (Export Share > 60%)

"Balanced" (Export Share 40%-60%)

"Import-Driven" (Export Share < 40%)

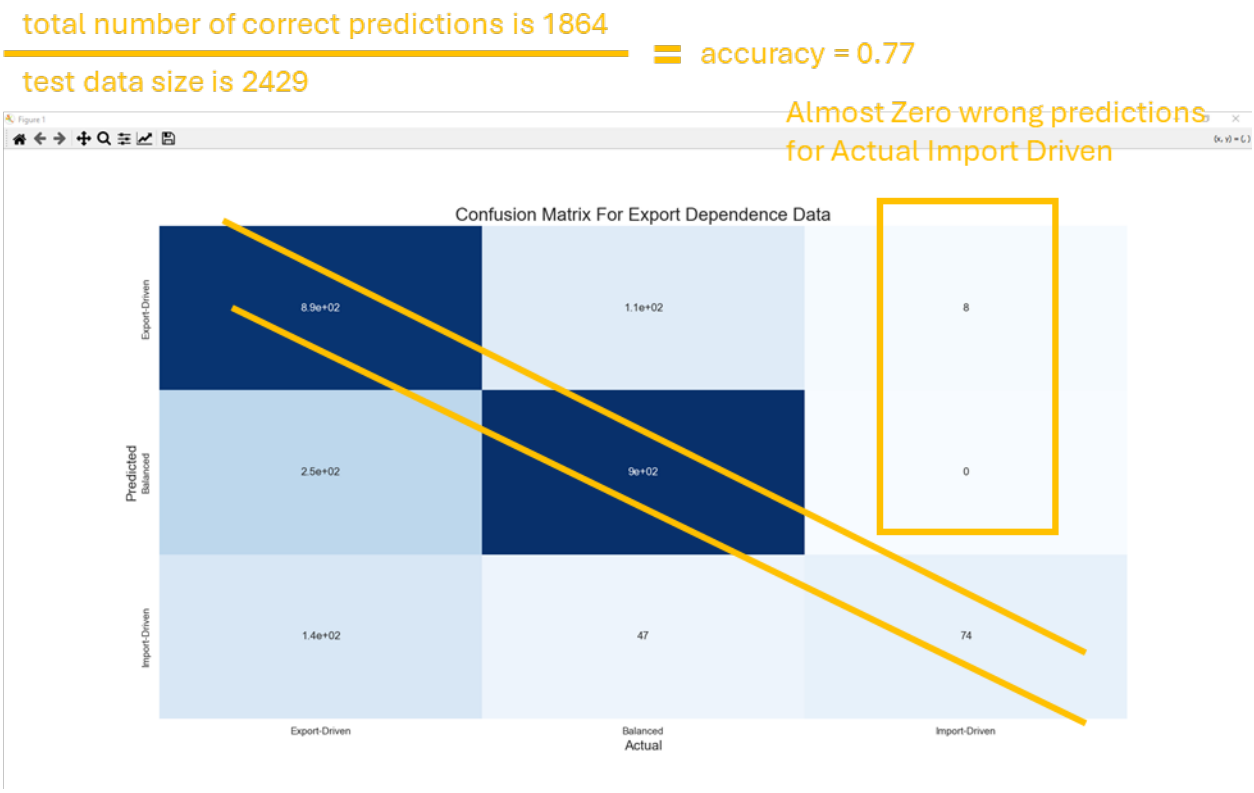
Most of the countries are export driven or balanced driven. They all want to sell their goods to someone else rather than buy other countries' goods. Export dependence probably has more overlap with balanced export dependence which is why both dominate.

(c) Apply Decision Tree modeling Using Sklearn in Python

3. Model Training: You use the training set to train your decision tree model.

4. Model Evaluation: After training, you use the test set to evaluate how well your model performs on unseen data.

Figure 28



The fewest bad predictions are in the Import driven category. 74 predictions matched the actual data.

(d) Create a Decision Tree Visualization

Figure 29

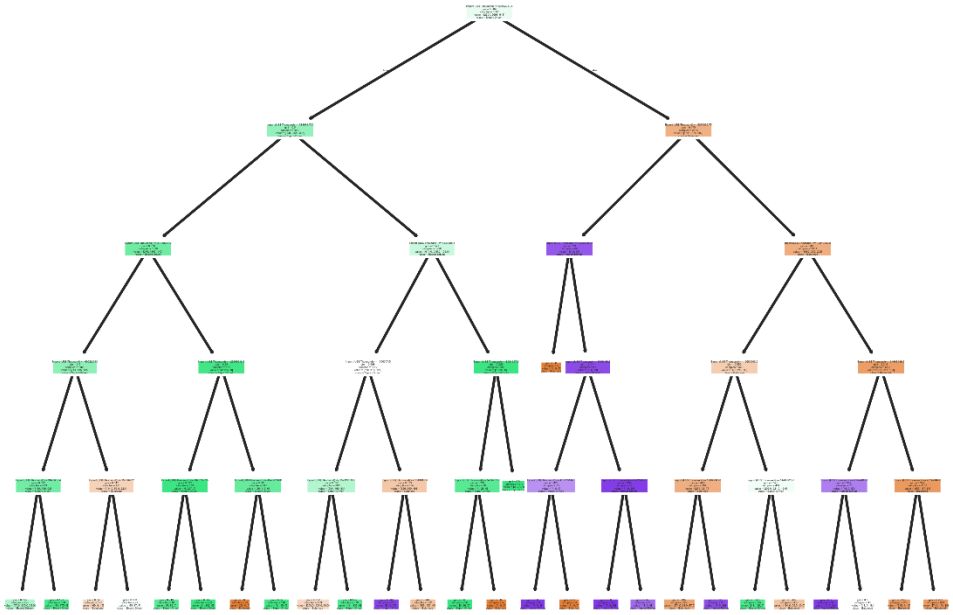


Figure 30

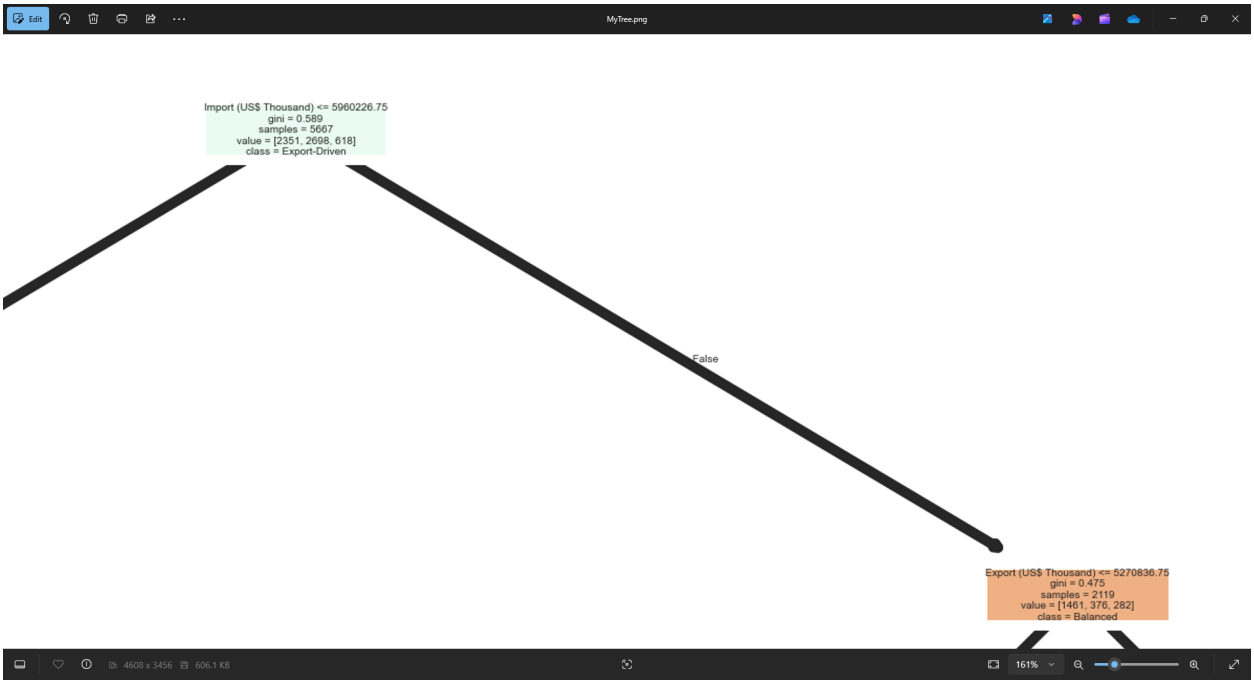
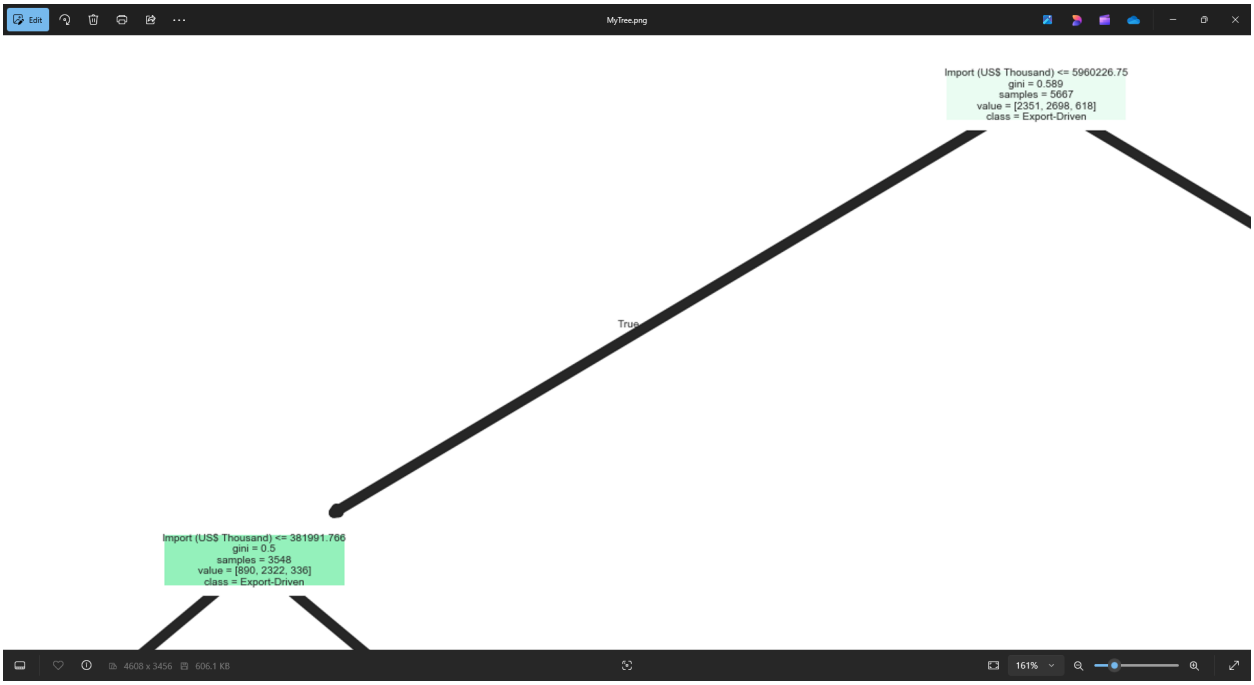
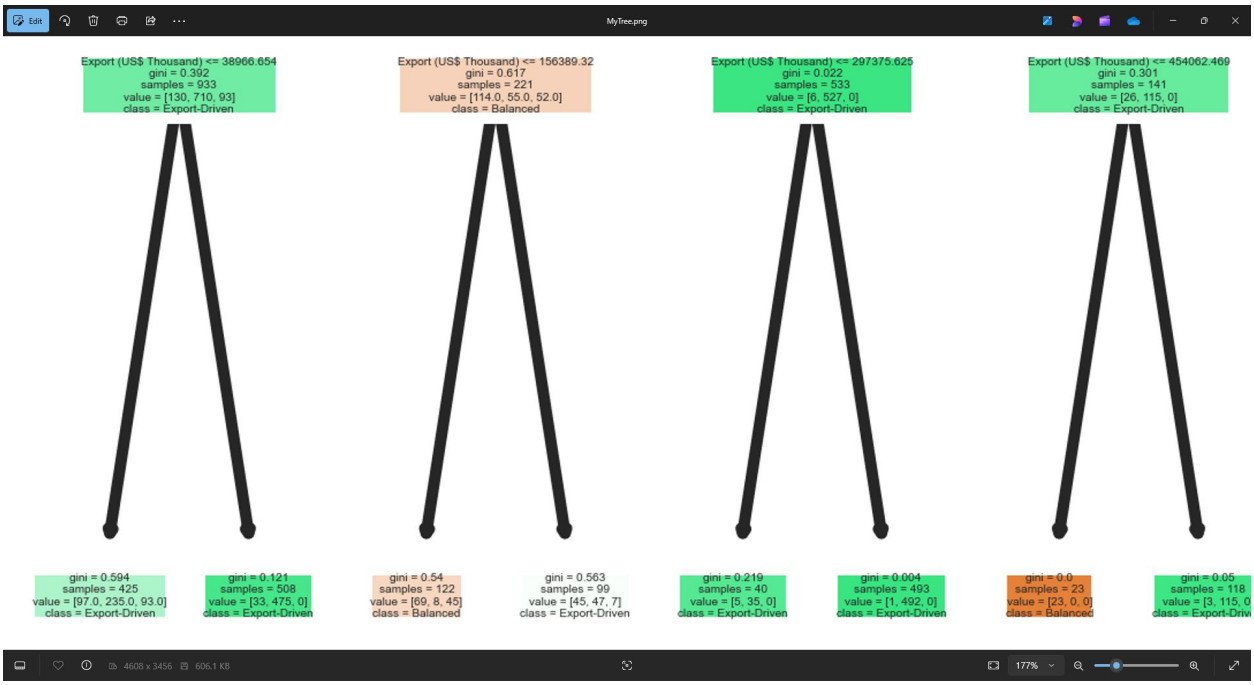


Figure 31





Depth is 5, less than 100 node Decision Tree.

The Gini impurity is a metric used in decision trees to measure how pure or impure a node is. It determines how mixed the labels are in a given dataset split.

(e) Technical Results

In Figures 23 through 26, for the last 34 years the import and export trade of the United States in respect to other countries has increased which has increased the size of the US economy and benefitted its people. Over the years, the average tariff rate imposed on other countries has decreased which is an inverse correlation to the amount of international trade between the US and other countries. The largest 3 or 4 countries have benefitted most from international trade. Their import and export trade amounts grew at a faster pace than smaller countries. With an increasing trade rate, these larger countries can leverage an even lower tariff rate which in turn increases more trading volume.

In Figure 26, almost all the countries have some tariff rates but over time as more international trade is done, the tariff rates have been decreasing which is beneficial for smaller countries to access the large US market. Small countries would not be able to sell as many goods just in their home market.

Figure 27 is the Export Dependence for all the countries in dataset and the target label. Most countries want to export their goods in order to get money for their economy, which is even more important for developing countries. The balanced driven category is almost equal to the export driven category. As a country gets more currency, they want to spend it on foreign goods that they do not usually have.

Figure 28 is the Confusion Matrix for Export Dependence. All the Export Dependence predictions do not match the actual Export Dependence but still matches the most. The accuracy for Export Dependence, Balanced Dependence and Import Dependence is 0.77 using test data. Export Dependence was one of the hardest to predict accurately while Import Dependence prediction is good. Balanced Dependence prediction is similar to Export Dependence.

Figure 29 is the Decision Tree. Depth was preset to 5 and nodes less than 100 in order to keep it manageable. The root node is Import amount, has gini number and number of samples used. The Gini impurity is a metric used in decision trees to measure how pure or impure a node is. It determines how mixed the labels are in a given dataset split. The next level row uses Export Amount to make a decision. To find a better accuracy rate or more ideal tree node, the number of rows can be varied to determine a decision tree for a better answer.

In conclusion, international trade has been beneficial to countries around the world. Their economies have grown which can then support the people. Over the last 34 years, international trade has increased and simple average tariff rates have decreased which has been a benefit for people all around the world.

Conclusion: The Impact of Trade and Tariffs on Everyday Life

Figure 32 Almost everyone will use a product today that was imported from another country

Trade Shapes the Global Economy

Trade is one of the most powerful forces driving prosperity worldwide. It allows countries to access products they do not produce, strengthens relationships between nations, and creates economic opportunities for businesses and workers. Over time, international trade has expanded, bringing more goods and services to people at lower costs. This interconnected economy benefits consumers by providing a greater variety of products, often at more competitive prices.

Tariffs Can Have Unintended Consequences

While tariffs are meant to protect domestic industries, they can also create challenges. When governments impose tariffs on imports, businesses and consumers often face higher prices. These additional costs can lead to inflation, making everyday essentials more expensive. History has shown that high tariffs can slow economic growth and lead to trade disputes between nations, affecting jobs and industries that rely on imports and exports.

Free Trade Encourages Growth and Innovation

Countries with lower tariffs and open trade policies tend to experience faster economic growth. Businesses can access new markets, and consumers benefit from lower prices and better-quality products. Over the past few decades, as many countries have reduced tariffs, global trade has increased, lifting millions of people out of poverty. International cooperation in trade agreements has helped businesses grow, encouraged investment, and supported economic stability.

Looking Ahead: Making Trade Work for Everyone

As the world continues to evolve, policymakers and businesses must find a balance between protecting local industries and fostering global trade. Understanding how tariffs impact everyday life helps people make informed decisions about economic policies. Countries that embrace fair trade policies can ensure that businesses, workers, and consumers all benefit from a stable and prosperous global economy.

References

Logistics and transportation of Container Cargo ship and Cargo plane... (2018, June 5). iStock. https://www.istockphoto.com/photo/logistics-and-transportation-of-container-cargo-ship-and-cargo-plane-with-working-gm968819844-264102201

World export & import dataset(1989—2023). (n.d.). Retrieved January 12, 2025, from https://www.kaggle.com/datasets/muhammadtalhaawan/world-export-and-import-dataset

Made germany germany flag ribbon circle stock vector (Royalty free) 2446788409. (n.d.). Shutterstock. Retrieved March 11, 2025, from https://www.shutterstock.com/image-vector/made-germany-flag-ribbon-circle-silver-2446788409

Made italy badge vector emblem italy stock vector (Royalty free) 395719213. (n.d.). Shutterstock. Retrieved March 11, 2025, from https://www.shutterstock.com/image-vector/made-italy-badge-vector-emblem-flag-395719213

Made japan stamps stock vector (Royalty free) 411150793. (n.d.). Shutterstock. Retrieved March 11, 2025, from https://www.shutterstock.com/image-vector/made-japan-stamps-411150793

Vector stamp flag lebanon lettering made stock vector (Royalty free) 1723901476. (n.d.). Shutterstock. Retrieved March 11, 2025, from https://www.shutterstock.com/image-vector/vector-stamp-flag-lebanon-lettering-made-1723901476