Global Impact on Food Prices

Mackenzie Kreutzer
Naz Shalamo
Sofiri Afonya
Kishankumar Patel

University of Calgary DATA 604 Winter 2024: Working with Data at Scale

1.0 Abstract

This project explored the intricate and delicate web of factors that could impact food prices globally. Focusing on five various data sets the project set forth a set of guiding questions aimed to answer what impacts food prices.

Through the five various datasets obtained from open source data sets, the project explores oil prices, food waste, country GDP, and food imports and compares them to the dependent variable of food prices. Important to note is the risk of causation bias due to the fact that food prices are a very nuanced topic. There are many factors to food prices increasing or decreasing, and due to project timelines are unable to explore to the full extent. The goal of this project is to see if any correlations between these datasets may impact the price of food, while still being aware that the project can not definitely define a reason for food prices to increase or decrease, but instead can only provide further exploration points of what may impact these prices.

Through data wrangling, this project was prepared in the Jupyter Python environment connecting to a SQL database. The data was filtered, cleaned and joined together in new tables to answer the guiding questions in hopes of revealing relationships between the various datasets.

Within the scope of the datasets used for this research we were able to find complex relationships not only in relation to the food prices but also amongst the datasets themselves. The conclusions made on relationships are found to depend on factors such as the countries themselves, with differences that are significant enough to even inverse the supposed correlation that is discovered in other countries. The information we find allows for an even broader investigation to be had, when taking into account the amount of externalities present within the datasets.

2.1 Context

Food prices are something that everyone has been talking about recently. From personal experience, we have all noticed the hike in food prices after the global COVID-19 pandemic. Research shows that Canadians have had to pay significantly more for food, and food prices have increased nearly 22% between February 2021 and February 2024 (CTV News, 2024). More people than ever have had to rely on food banks over the past year, which is alarming to hear as young students (CTV News, 2024). This trend inspired the project, as we wanted to explore why food prices may be increasing, and if these trends have been seen globally, or just in Canada.

2.2 Dataset

To answer guiding questions, we have set for this project, we will be using five datasets which are obtained from their respective websites. The first and most important dataset to begin our investigation is Global Food Prices. This dataset is taken from Kaggle, and it is compiled by the World Food Program and distributed by HDX (The Humanitarian Data Exchange). This dataset contains food prices for 76 countries covering food such as maize, rice, beans, fish, and sugar (Global Food Prices Database, 2020). This dataset goes back to the year 1992 for a few countries and 2003 or thereafter for other countries as many countries did not start reporting food prices before 2003 (Global Food Prices Database, 2020). The data is in a CSV format containing over 1.04 million rows for the column country, locality, market, goods purchased, price & currency used, quantity exchanged, and month/year of purchase (Global Food Prices Database, 2020).

The second source of data we will need to answer our set of guiding questions would be the GDPs of different countries corresponding to different points in time. The dataset we plan to use is in excel format and has 270 rows corresponding to each country like the previous set (World GDP by Country, 2023). The slight difference in number seems to be due to duplicates or extra information. The dataset contains the column for countries, country code, then the GDP (in US currency) by years from 1960 to 2022 (World GDP by Country, 2023). This data set was retrieved from data.worldbank.org. This is a website that conducts extensive research and collects data on global economic and social trends, with open access to information that continues to be updated and verified by multiple collaborators to create a better understanding of development issues worldwide.

The third dataset is in an excel format and it is sourced from the World Integrated Trade Solution (WITS) website. This is a global trade collaboration website that has aided in the development of WITS, a trusted source that provides multiple ways to access trade, tariff, and nontariff data. This food import dataset is a collection of information that provides insights into the import product shares of various countries over different years (Food Products Imports by Country, n.d.). This data is recorded from 1988 to 2021 (Food Products Imports by Country, n.d.). The dataset has 266 rows for the same number of different countries, but to make a more concise project, we plan to filter the data to include only certain countries that will be compared across the board. The dataset is sourced from the United Nations Conference on Trade and Development (UNCTAD) in collaboration with organizations such as the International Trade Center, United Nations Statistical Division (UNSD) and the World Trade Organization (WTO) which ensures its accuracy and reliability (Food Products Imports by Country, n.d.).

The fourth dataset is sourced from Food Loss and Waste Database from the Food and Agriculture Organization of the United States (FAO), FAO provides free access to data which can be used by anyone. This dataset was compiled by FAO through an in-depth review of publicly available literature, which included data and information from almost 500 articles, reports, and studies from a variety of sources such as World Bank, The Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ), FAO, International Food Policy Research Institute (IFPRI), and more (Food Loss and Waste Database, n.d.). The database gives information into food loss and waste percentages of several food products with their respective countries and years (Food Loss and Waste Database, n.d.). The data goes back to 1966 for very few countries and thereafter for other countries to 2020. We are interested to see what impact food loss and waste have on global food prices.

The fifth dataset is retrieved from ourworldindata.org. with two crucial columns showing the cost of crude oil for every year in 1861 in the global world measured in US dollars per cubic meter. The production and transportation of commodities is crucial to the price of goods. We believe that this importance should be accounted for as a variable in the fluctuations of the consumed goods. The data is in CSV format, containing 163 rows with four columns named ‘Entity’, ‘Code’, ‘Year’, and ‘Oil price - Crude prices since 1861 (current US$)’. In this set Entity only refers to the world as its a global representation, and because there is only one entity the code will always be “OWID_WRL”.

2.3 Data Limitations and Assumptions

The dataset being collected from different sources can result in many of the entries that we plan to join on, such as countries, won't be as easy as initially planned as there will be different names for the same values. An example can be the country, such as how the food price data labels India as Bassas da India while the other datasets don't. Or even how the name of a comedy such as wheat, will be named with categorized specifications such as "Wheat - Retail". The countries also don't have the same amount of entries as some data only starts from more recent years in comparison to others, however this can be solved by focusing on averages between years and being selective as to which entries are being compared.

Another important thing we keep in mind through this investigation is that 4 datasets are not enough to make conclusions on something as complex as food prices, but we do expect to see some sort of correlations in specific areas for the sake of interest. We are keeping the scope of this analysis small, but will be as complex as possible with the queries made for getting as much information as possible. Also since our global food prices dataset only contains the countries that are developing and not the developed countries, therefore our analysis results will apply to the developing countries only.

3.1 Data Preparation

The first step to answer our guiding questions was to shorten our data set of global food prices. It had more than 1.04 million rows, we shortlisted our commodities based on what we had in food waste and our global food price dataset.

```python
import pandas as pd
import numpy as np

# Read the first CSV file
food1 = pd.read_csv("wfp_food_prices_database.csv")
country = food1['adm0_name'].unique()

# Read the second CSV file
waste = pd.read_csv("FoodWaste_updated.csv")
commodities = waste['Item'].unique()

# Get unique commodities from the first CSV file
commodities1 = food1['cm_name'].unique()

# Split the commodities into individual lowercase words
commodities_words = set([word.lower() for commodity in commodities for word in str(commodity).split()])
commodities1_words = set([word.lower() for commodity in commodities1 for word in str(commodity).split()])

# Find the common words
common_words = commodities_words.intersection(commodities1_words)

# Convert to numpy array
common_words_array = np.array(list(common_words))

print(common_words_array)
```

The second most important thing we did before fully comparing datasets was figuring out a means to address the currency within the food price dataset, especially since the other datasets are based in USD while the food price has over 100 different currencies for different countries. This problem was addressed by creating an exchange rate table, that could be used for conversions everytime the food price dataset is meant to be compared or even merged:

```
Exchange = pd.read_csv("exchange_rater.csv")

Exchange = Exchange.rename(columns={
        'Country':'Country',
        '2017':'exchange_rate',
        'Exchange rates': 'code',
    })

Exchange.to_sql("exchange", engine, index=False, if_exists='replace')
Exchange_table_df = pd.read_sql_table("exchange", engine)
Exchange_table_df
```

	Country	exchange_rate	code
0	Columbia	2951.5	COP
1	Afghanistan	68.03	AFN
2	Bolivia	6.9	BOB
3	Chad	580.6	XAF
4	Egypt	17.78	EGP
5	India	65.1	INR
6	Costa Rica	567.5	CRC
7	Indonesia	13380.8	IDR
8	Bangladesh	81.2	BDT
9	Ethiopia	23.87	ETB
10	Sri lanka	152.5	LKR
11	Guatemala	7.35	GTQ
12	Iraq	1184	IQD
13	Kenya	103.41	KES
14	Syrian Arab Republic	492.61	SYP
15	Turkey	3.65	TRY
16	Myanmar	1360.36	MMK
17	Niger	580.66	XOF
18	Nigeria	305.79	NGN
19	Pakistan	105.46	PKR
20	Panama	1	USD
21	Peru	3.26	PEN
22	China	6.76	RMB
23	Argentina	16.56	ARS
24	South Africa	13.32	ZAR
25	Mexico	18.93	MXN
26			None
27			None

After this was completed, we started creating our tables and inserting the data into the tables. It is important to note that since we are using Jupyter, we had to do all of this pre work separately, and you may notice variances in the variable names going through this report. We started by creating the food waste table as shown below.

```python
#Food Waste Database

# CREATE TABLE STATEMENT
create_statement = '''create table mackenzie_kreutzer.foodwaste1 (
    domain_code varchar(5),
    domain varchar(100),
    area_code varchar(5),
    area varchar(500),
    element_code int,
    element varchar(50),
    item_code varchar(10),
    item varchar(100),
    year_code int,
    year int,
    unit varchar(10),
    value int,
    flag varchar(5),
    flag_description varchar(50)
    );'''

# creates table and lets me know if it was created successfully
create_cursor = myconnection.cursor()
try:
    create_cursor.execute(create_statement)
except mysql.connector.Error as err:
    if err.errno == errorcode.ER_TABLE_EXISTS_ERROR:
        print("Ooops! We already have that table")
    else:
        print(err.msg)
else:
    print("Table created successfully!")

create_cursor.close()
```

We then inserted the data into the datatable as shown here.

```python
#Insert data into the database

insert_into_foodwaste1 = """
INSERT INTO foodwaste1
(domain_code, domain, area_code, area, element_code, element, item_code, item, year_code, year, unit, value, flag, flag_description)
VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
"""
with myconnection.cursor() as cursor:
    for i, row in waste.iterrows():
        cursor.execute(insert_into_foodwaste1, tuple(row))
    myconnection.commit()
    print("Data inserted successfully")
```

```
Ooops! We already have that table
Data inserted successfully
```

After, we then created the food price datatable, and inserted the data.

```python
#Food Price Database

# CREATE TABLE STATEMENT
create_statement = '''create table mackenzie_kreutzer.foodprice2 (
    adm0_id int,
    adm0_name varchar(100),
    adm1_id int,
    adm1_name varchar(500),
    mkt_id int,
    mkt_name varchar(100),
    cm_id int,
    cm_name varchar(100),
    cur_id int,
    cur_name varchar(100),
    pt_id int,
    pt_name varchar(100),
    um_id int,
    um_name varchar(100),
    mp_month int,
    mp_year int,
    mp_price int,
    mp_commoditysource varchar(10)
    );'''

# creates table and lets me know if it was created successfully
create_cursor = myconnection.cursor()
try:
    create_cursor.execute(create_statement)
except mysql.connector.Error as err:
    if err.errno == errorcode.ER_TABLE_EXISTS_ERROR:
        print("Ooops! We already have that table")
    else:
        print(err.msg)
else:
    print("Table created successfully!")

create_cursor.close()
```

```
#Insert data into the database

insert_into_foodprice2 = """
INSERT INTO foodprice2
(adm0_id, adm0_name, adm1_id, adm1_name, mkt_id, mkt_name, cm_id, cm_name, cur_id, cur_name, pt_id, pt_name, um_id, um_name, mp_month, mp_year,
VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
"""
with myconnection.cursor() as cursor:
    for i, row in price.iterrows():
        cursor.execute(insert_into_foodprice2, tuple(row))
    myconnection.commit()
    print("Data inserted successfully")
```

```
Table created successfully!
Data inserted successfully
```

We then created the GDP datatable. As this dataset has so many various columns, we had to create and insert the data in a different way so as not to overload the fragile Jupyter system.

```
# Prepare the DataFrame by renaming year columns
year_columns = [f'Year{year}' for year in range(1960, 2023)]
gdp_data.columns = ['Country', 'Country_Code'] + year_columns

# SQL to create the table
create_statement = '''
CREATE TABLE IF NOT EXISTS mackenzie_kreutzer.gdp (
    Country VARCHAR(100),
    Country_Code VARCHAR(10),
''' + ',\n'.join([f'{year} DOUBLE' for year in year_columns]) + '''
);'''

# Connect to the database and create the table
try:
    connection = mysql.connector.connect(**config)
    cursor = connection.cursor()
    cursor.execute(create_statement)
    print("Table created successfully or already exists.")
except mysql.connector.Error as err:
    print("Error occurred: ", err)
finally:
    if connection.is_connected():
        cursor.close()
        connection.close()
```

```python
# Connect again to insert data
try:
    connection = mysql.connector.connect(**config)
    cursor = connection.cursor()
    insert_statement = f'''
    INSERT INTO mackenzie_kreutzer.gdp (Country, Country_Code, {', '.join(year_columns)})
    VALUES ({', '.join(['%s']*len(gdp_data.columns))});
    '''

    # Insert each row from the DataFrame
    for index, row in gdp_data.iterrows():
        cursor.execute(insert_statement, tuple(row))
    connection.commit()
    print("Data inserted successfully.")
except mysql.connector.Error as err:
    print("Error occurred during data insertion: ", err)
finally:
    if connection.is_connected():
        cursor.close()
        connection.close()
```

After creating our GDP table we created oil prices and food imports.

Food Imports:

```python
foodi = pd.read_csv("food_imports.csv")

foodi = foodi.rename(columns={
        'Country':'Country',
        'Year_Name':'Year',
        'Year_Value': 'Import(%)',
    })

foodi.to_sql("foodimp", engine, index=False, if_exists='replace')
foodi_table_df = pd.read_sql_table("foodimp", engine)
foodi_table_df.head()
```

Oil Prices:

```python
Oil = pd.read_csv("crude-oil-prices.csv")
Oil = Oil.rename(columns={
'Entity':'World',
'Code':'Code',
'Year':'Year',
'Oil price - Crude prices since 1861 (current US$)':'Oil Price',
})

Oil.to_sql("oil_prices", engine, index=False, if_exists='replace')
Oil_table_df = pd.read_sql_table("oil_prices", engine)
Oil_table_df.tail()
```

3.2 Data Cleaning, Wrangling and Uploading

The data that we imported was filtered so that only necessary columns were needed for the sake of space and organization. The food price dataset is the main one we performed these manipulations on. Many commodities are specific to certain countries, so we manually got rid of rows with these entries in the excel file. Through pandas we were able to filter out columns that weren't needed for our investigation such as market name and area codes:

```python
price_df.drop(columns=["adm0_id","adm1_id","adm1_name","mkt_id","pt_id","pt_name","um_id","um_name","mp_commoditysource"], inplace=True)
#change names to make more understandable
```

Also the renaming of entries, so that the merges would be cohesive names for variables across datasets:

```
price.rename(columns={"adm0_name": "country",  "cm_name":"commodity", "cur_name":"currency",
                      "um_name":"unit", "mp_month": "month", "mp_year":"year", "mp_price":"price" })
```

We needed to fix the fact that this dataset didn't have a price based on just years but months as well. We manipulated the data for it to only be the year column as average price:

	Country	Commodity	Currency_name	Quantity	Quantity_unit	Month	Year	Price
0	Afghanistan	Wheat - Retail	AFN	5	KG	1	2003	7.0
1	Afghanistan	Wheat - Retail	AFN	5	KG	2	2003	6.0
2	Afghanistan	Wheat - Retail	AFN	5	KG	3	2003	7.0
3	Afghanistan	Wheat - Retail	AFN	5	KG	4	2003	6.0
4	Afghanistan	Wheat - Retail	AFN	5	KG	5	2003	6.0

```
sql_query =  """
          SELECT country, currency_name, Commodity, Quantity, Quantity_unit, Year, AVG(Price) AS average_price
          FROM food_prices
          GROUP BY country, currency_name, Commodity, Quantity, Quantity_unit, Year
          """
df = pd.read_sql_query(sql_query, engine)
df
```

	country	currency_name	Commodity	Quantity	Quantity_unit	Year	average_price
0	Afghanistan	AFN	Rice (low quality) - Retail	5	KG	2007	27.249533
1	Afghanistan	AFN	Rice (low quality) - Retail	5	KG	2008	40.336394
2	Afghanistan	AFN	Rice (low quality) - Retail	5	KG	2009	34.720331
3	Afghanistan	AFN	Rice (low quality) - Retail	5	KG	2010	31.726013
4	Afghanistan	AFN	Rice (low quality) - Retail	5	KG	2011	32.855337
...	...	...	...	...	...	...	...
1782	Turkey	TRY	Wheat flour - Retail	5	KG	2014	2.565082
1783	Turkey	TRY	Wheat flour - Retail	5	KG	2015	2.672583
1784	Turkey	TRY	Wheat flour - Retail	5	KG	2016	2.848333
1785	Turkey	TRY	Wheat flour - Retail	5	KG	2017	3.132718
1786	Turkey	TRY	Wheat flour - Retail	5	KG	2018	3.167391

We also needed to fix the measurement issue, where different commodities prices are measured in different amounts. For consistency sake, we get rid of the Quantity column and create a price per unit column:

```
sql_query = """
            SELECT country, currency_name, commodity, Quantity_unit, Year, AVG(price) / Quantity AS price_per_unit
            FROM food_prices
            GROUP BY country, currency_name, commodity, Quantity_unit, Year;
            """
final = pd.read_sql_query(sql_query, engine)
final
```

	country	currency_name	commodity	Quantity_unit	Year	price_per_unit
0	Afghanistan	AFN	Rice (low quality) - Retail	KG	2007	5.449907
1	Afghanistan	AFN	Rice (low quality) - Retail	KG	2008	8.067279
2	Afghanistan	AFN	Rice (low quality) - Retail	KG	2009	6.944066
3	Afghanistan	AFN	Rice (low quality) - Retail	KG	2010	6.345203
4	Afghanistan	AFN	Rice (low quality) - Retail	KG	2011	6.571067
...	...	...	...	...	...	...
1782	Turkey	TRY	Wheat flour - Retail	KG	2014	0.513016
1783	Turkey	TRY	Wheat flour - Retail	KG	2015	0.534517
1784	Turkey	TRY	Wheat flour - Retail	KG	2016	0.569667
1785	Turkey	TRY	Wheat flour - Retail	KG	2017	0.626544
1786	Turkey	TRY	Wheat flour - Retail	KG	2018	0.633478

1787 rows × 6 columns

The food waste database had a lot of useless columns that were not relevant to the analysis that we wanted to complete. We created some code that dropped the columns to make the processing time more manageable. Removing these columns allowed for less data to be processed at one time, making the processing time faster.

```
#Data Cleanup of Food Waste
#Removing Columns
# Columns to remove
columns_to_remove = ['domain_code', 'domain', 'element_code', 'element', 'year_code', 'flag', 'flag_description']

cursor = myconnection.cursor(buffered=True, dictionary=True)

# Loop through the columns and execute ALTER TABLE query to remove each one
for column in columns_to_remove:
    try:
        alter_query = f"ALTER TABLE foodwaste1 DROP COLUMN {column}"
        cursor.execute(alter_query)
        print(f"Column '{column}' successfully removed.")
    except mysql.connector.Error as error:
        print(f"Failed to remove column '{column}': {error}")

# Commit the changes and close the connection
connection.commit()
connection.close()
```

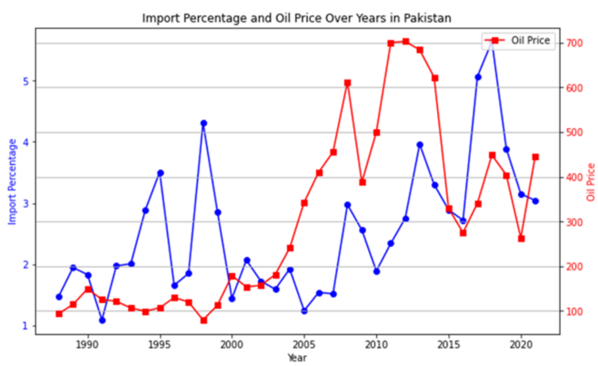
4.1 Exploratory Analysis

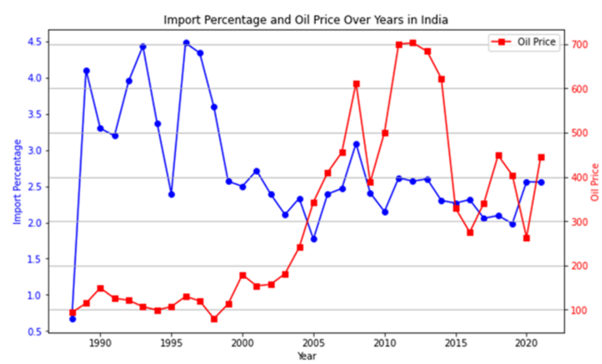
4.1.1 Is there a correlation between oil prices and food imports?

In order to answer this question for our future exploratory analysis, we joined tables of oil prices and food imports. Following is the query and the results for that.

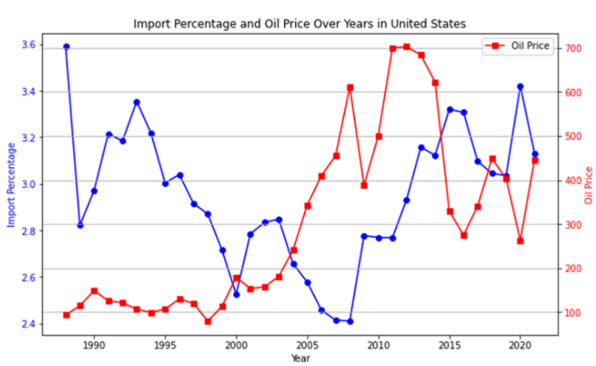
```
# SQL query to merge the two tables
query = """
SELECT
    CASE
        WHEN o.World = 'World' THEN f.Country
        ELSE o.World
    END AS Country,
    f.Year AS Year,
    f.`Import(%)` AS Import_Percentage,
    o.`Oil Price` AS Oil_Price
FROM
    oil_prices o
INNER JOIN
    foodimp f ON o.Year = f.Year AND (o.World = f.Country OR o.World = 'World');

"""

# Execute the query and fetch the results
merged_df = pd.read_sql_query(query, engine)
merged_df.to_csv("oil_price_and_food_imports.csv", index=False)

merged_df.head()
```

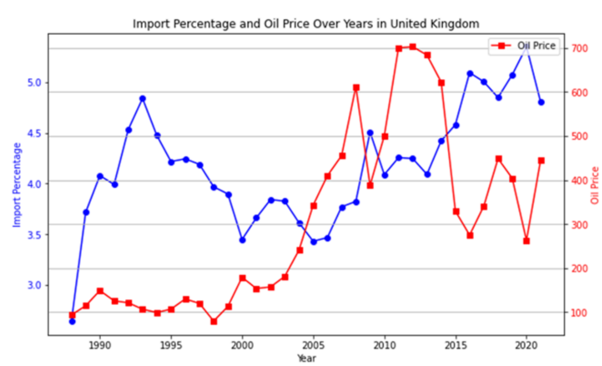
	Country	Year	Import_Percentage	Oil_Price
0	World	1988	3.2619	93.86798
1	World	1989	3.0612	114.63860
2	World	1990	3.1452	149.23067
3	World	1991	3.0808	125.80175
4	World	1992	3.2705	121.52420

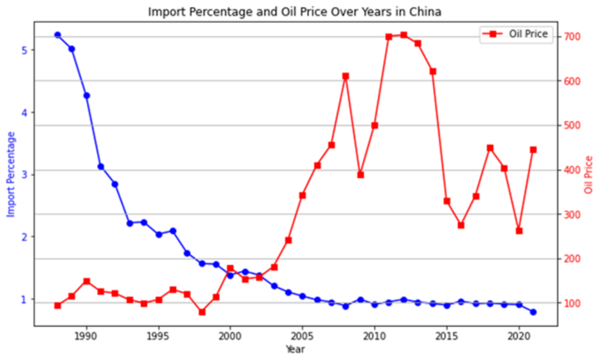


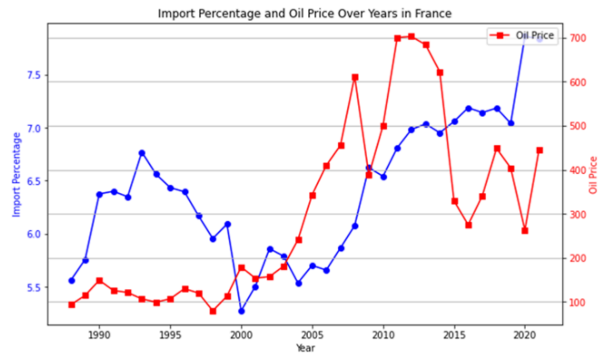


From the above four and below 2 plots we got to know that China has decreased their food imports by 5 times from 1988-2021. But we do see some correlation between oil prices and food imports. As seen from all of the plots, after the year 2000 all of the peaks and troughs are somewhat inversely identical. With the increase in oil prices we see a decrease in food imports and vice versa, there is clear evidence of this in France, USA, and UK. One thing to note is that the USA, India, and China have reduced their food imports but Pakistan, France and UK have increased in the last two decades.









We will use the following queries to get the average food import from 1988-2021 to get top 7 countries that import food. And the same way we will find the bottom 7 countries that import food.

```
countries_to_keep = [
    'India', 'China', 'Indonesia', 'Bolivia', 'Guatemala', 'Costa Rica',
    'Panama', 'Nigeria', 'Kenya', 'Ethiopia', 'Egypt', 'Iran',
    'Turkey', 'Syria', 'Iraq', 'Pakistan', 'Afghanistan', 'Sri Lanka',
    'Bangladesh', 'Vietnam'
]

# Convert the list of countries to a string format for SQL query
country_list = ', '.join([f"'{country}'" for country in countries_to_keep])

sql_query = f"""
SELECT
    Country,
    AVG(`Import(%)`) AS avg_import_percentage
FROM
    foodimp
WHERE
    Country IN ({country_list})
GROUP BY
    Country
ORDER BY
    avg_import_percentage DESC
LIMIT 7;
"""

# Execute the query and read the results into a DataFrame
top_average_import_percentage = pd.read_sql_query(sql_query, engine)
```

Top 7:

	Country	avg_import_percentage
0	Yemen Democratic	0.000500
1	Qatar	0.010197
2	Libya	0.012062
3	South Sudan	0.014500
4	Brunei	0.015215
5	Angola	0.032167
6	Iraq	0.117239

Bottom 7:

	Country	avg_import_percentage
0	Malawi	70.441094
1	Reunion	60.456175
2	Martinique	56.170725
3	Cote d'Ivoire	48.686950
4	Sao Tome and Principe	48.441409
5	Guadeloupe	47.462775
6	Cuba	45.188524

From these results, the countries that least imported food is from the Middle East and Africa. Countries that exported the most amount of food are mostly island nations and some African countries.

4.1.2 Is there a correlation between GDP and food imports?

Before I began my analysis on the correlation between GDP and food imports, I used SQL queries to gather and reshape data on GDP and food imports for select countries from 2010 to 2018. The SQL queries were run within a Jupyter Notebook, aiming to transform the wide-format GDP and food imports tables into a long format that aligns with analysis requirements. This transformation was essential to facilitate the merging of datasets based on the 'Country' and 'Year' columns, allowing me to examine the trends and potential correlations between a country's GDP and its food imports. The data extracted and processed from these queries served as a foundation for generating visualizations and conducting a correlation analysis.

```
gdp_query = "SELECT * FROM gdp_data"
gdp_data = pd.read_sql(gdp_query, engine)

food_imports_query = "SELECT * FROM food_imports"
food_imports_data = pd.read_sql(food_imports_query, engine)

years = [str(year) for year in range(2010, 2018)]
countries_of_interest = ['Afghanistan', 'Ethiopia', 'Myanmar', 'Pakistan', 'Guatemala', 'Sri Lanka']

gdp_long = pd.melt(gdp_data, id_vars=['Country'], value_vars=years, var_name='Year', value_name='GDP')
gdp_long = gdp_long[gdp_long['Country'].isin(countries_of_interest)]

food_imports_long = pd.melt(food_imports_data, id_vars=['Partner Name'], value_vars=years, var_name='Year', value_name='Import Share')
food_imports_long = food_imports_long[food_imports_long['Partner Name'].isin(countries_of_interest)]

gdp_long['Year'] = gdp_long['Year'].astype(int)
food_imports_long['Year'] = food_imports_long['Year'].astype(int)

merged_data = pd.merge(gdp_long, food_imports_long, left_on=['Country', 'Year'], right_on=['Partner Name', 'Year'])

# Drop the 'Partner Name' column as it is redundant
merged_data.drop(columns=['Partner Name'], inplace=True)
print(merged_data.head())
```

```
       Country  Year           GDP  Import Share
0  Afghanistan  2010  1.563384e+10        0.5451
1    Guatemala  2010  4.067658e+10       19.1166
2    Sri Lanka  2010  5.863616e+10        2.4728
3      Myanmar  2010  3.779605e+10        0.7003
4     Pakistan  2010  1.771655e+11        1.8903
```

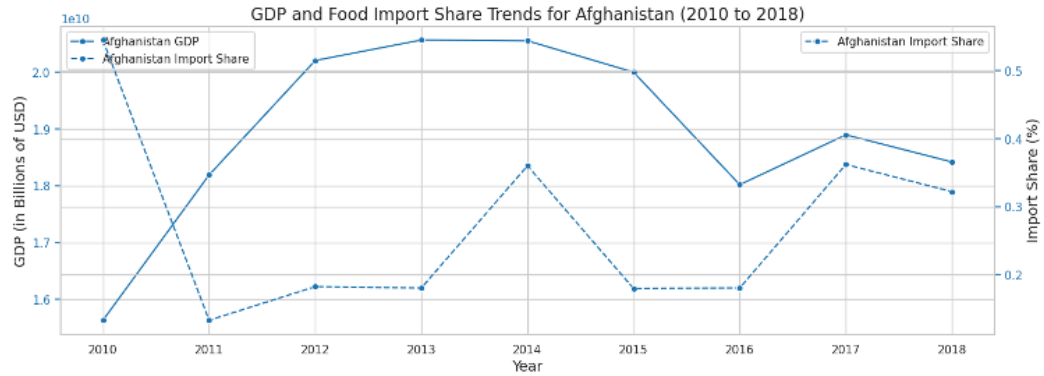
The assessment of GDP and food imports across the selected countries illustrates the nuanced interplay of economic indicators. In Afghanistan, the erratic pattern of GDP against the backdrop of food import share points towards an inverse relationship during certain years. For instance, from 2011 to 2012, a downturn in GDP is met with an uptick in food import share, highlighting a potential dependency on imports when domestic economic conditions weaken. Conversely, a stark decline in food import share in 2018 parallels a drop in GDP, reinforcing this inverse trend.

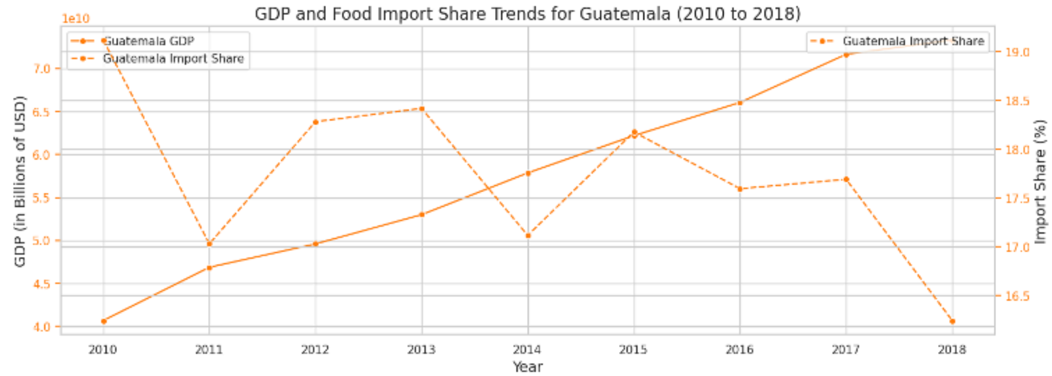
Guatemala's economic narrative is characterized by steady GDP growth interspersed with fluctuations in food import share, most notably the spike in 2015 followed by a decline. This variability suggests a disconnection between overall economic health and the reliance on food imports, as the two do not exhibit a consistent direct relationship.

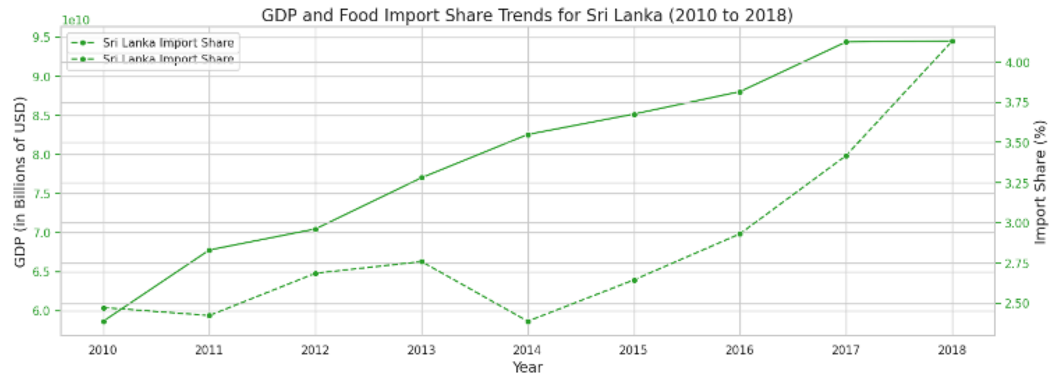
Contrastingly, Sri Lanka shows a congruent rise in both GDP and food import share over the observed period, hinting at a sustained economic expansion alongside an increased integration of imported food in national consumption. Despite minor variations, the import share consistently trends upwards, possibly indicating a growing dependence on external food sources alongside economic development.

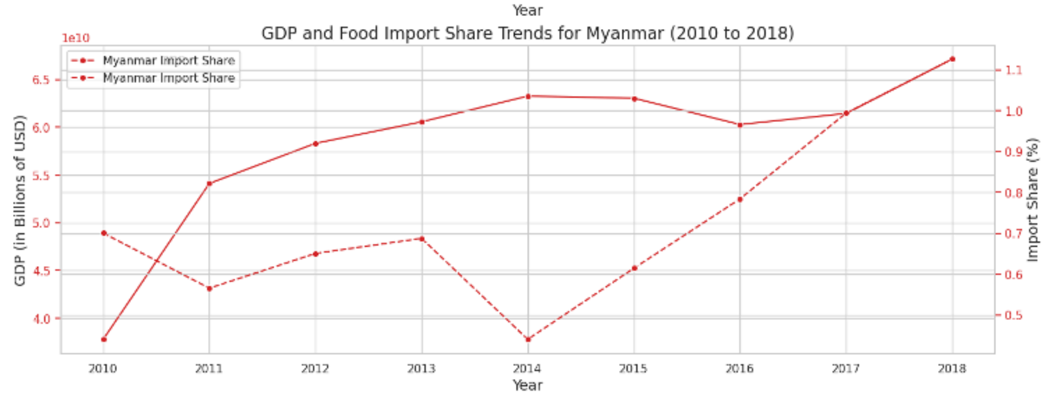
In Myanmar, post-2012 GDP growth and a subsequent fall in food import share post-2013 may signal a boost in domestic food production capabilities or a strategic shift in food importation policies, leading to reduced reliance on foreign food sources despite economic improvements.

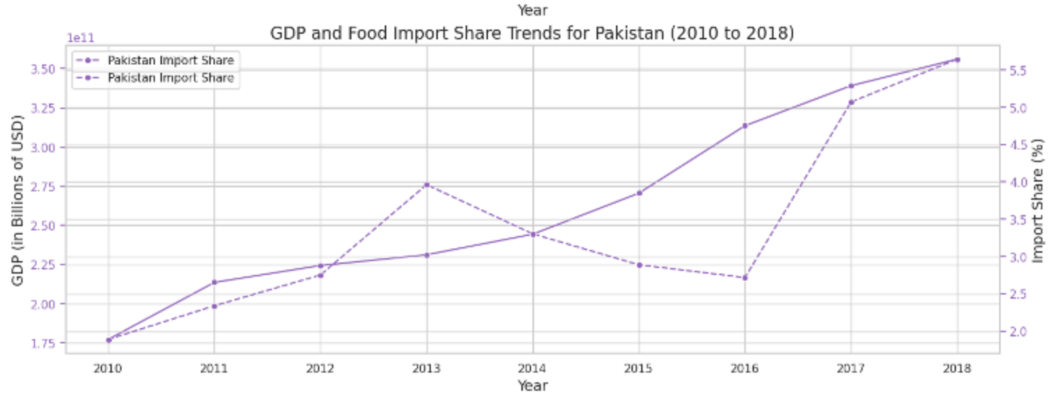
Pakistan's trajectory depicts a positive correlation, with both GDP and food import share ascending over time, albeit with periodic GDP fluctuations. This upward movement may reflect a burgeoning economic environment coupled with an escalating dependence on imported food, a trend that could have wide-ranging implications for national food security and trade policy.


GDP and Food Import Share Trends for Afghanistan (2010 to 2018)
1e10
Afghanistan GDP
Afghanistan Import Share
Afghanistan Import Share
GDP (in Billions of USD)
Import Share (%)
Year


GDP and Food Import Share Trends for Guatemala (2010 to 2018)
1e10
Guatemala GDP
Guatemala Import Share
Guatemala Import Share
GDP (in Billions of USD)
Import Share (%)
Year


GDP and Food Import Share Trends for Sri Lanka (2010 to 2018)
1e10
Sri Lanka Import Share
Sri Lanka Import Share
GDP (in Billions of USD)
Import Share (%)
Year


GDP and Food Import Share Trends for Myanmar (2010 to 2018)
1e10
Myanmar Import Share
Myanmar Import Share
GDP (in Billions of USD)
Import Share (%)
Year


GDP and Food Import Share Trends for Pakistan (2010 to 2018)
1e11
Pakistan Import Share
Pakistan Import Share
GDP (in Billions of USD)
Import Share (%)
Year

The divergent trends observed underscore the complexity of the relationship between GDP and food import share. While the upward trends in countries like Sri Lanka and Pakistan could imply a growing reliance on imported food in the context of economic growth, the intricate patterns seen in Afghanistan and Myanmar necessitate a comprehensive analysis that goes beyond GDP figures and import statistics.

Year

```
•[49]: # Calculate the Pearson correlation coefficient for each country
for country in merged_data['Country'].unique():
    country_data = merged_data[merged_data['Country'] == country]
    correlation = country_data[['GDP', 'Import Share']].corr().iloc[0, 1]
    print(f"The Pearson correlation coefficient for {country} is: {correlation:.2f}")

The Pearson correlation coefficient for Afghanistan is: -0.56
The Pearson correlation coefficient for Guatemala is: -0.60
The Pearson correlation coefficient for Sri Lanka is: 0.72
The Pearson correlation coefficient for Myanmar is: 0.25
The Pearson correlation coefficient for Pakistan is: 0.80
```

I employed Pearson correlation coefficients as a supplementary tool to enhance the insights gleaned from the plots, aiming to unravel the dynamics between GDP and food import shares from 2010 to 2018. I found that Afghanistan and Guatemala exhibit a moderate negative correlation, with coefficients of -0.56 and -0.60, respectively, suggesting that increases in GDP correlate with declines in food import share. Contrarily, Sri Lanka and Pakistan display strong positive correlations of 0.72 and 0.80, signifying that as their economies expanded, so did their reliance on imported food. Myanmar presented a more nuanced picture with a weak positive correlation of 0.25, hinting at other underlying factors influencing its food import shares. These coefficients provided valuable context to my visual assessments, indicating a complex and country-specific relationship between economic growth and food imports.

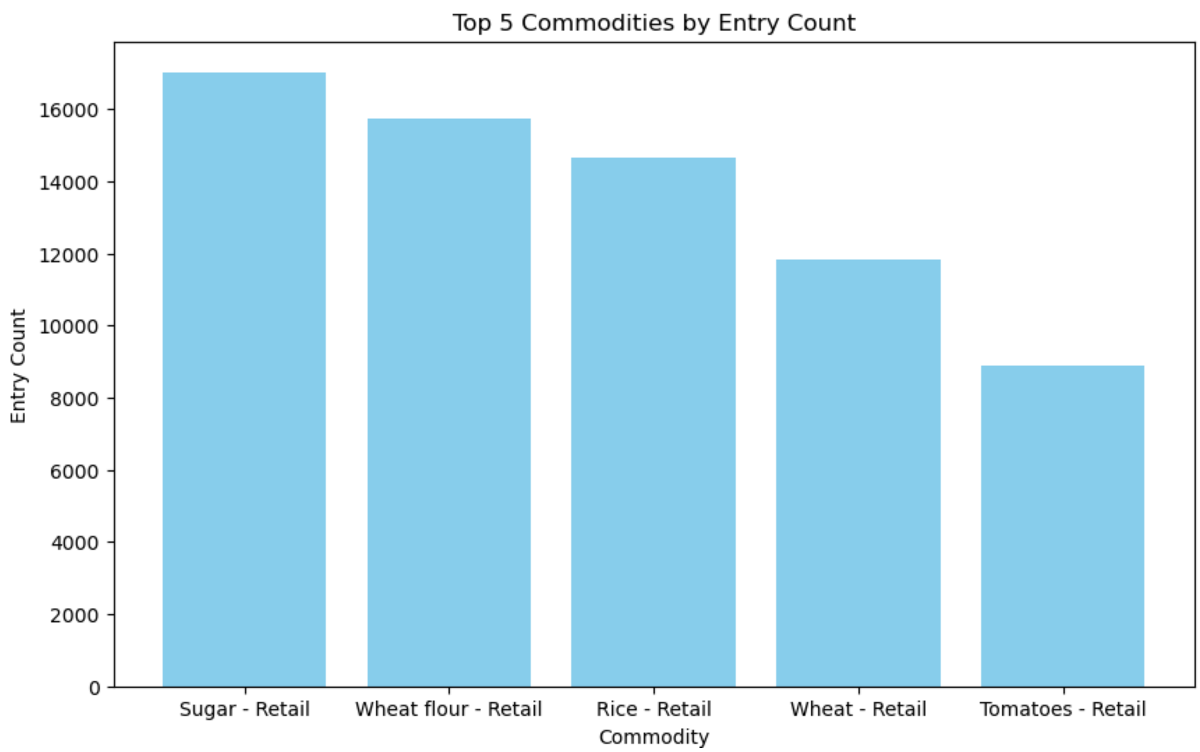
4.1.3 Have food prices changed over the years?

The plan for this section is to add the cost of selecting the most common commodities, get the average for each year across countries, then create a histogram for said commodity in each year. We start by finding which commodities are actually the most common.

```
SELECT
    commodity,
    COUNT(*) AS entry_count
FROM
    food_prices
GROUP BY
    commodity
ORDER BY
    entry_count DESC
LIMIT 5
"""

query_result = pd.read_sql_query(sql_query, engine)
query_result
```

	commodity	entry_count
0	Sugar - Retail	17025
1	Wheat flour - Retail	15749
2	Rice - Retail	14661
3	Wheat - Retail	11824
4	Tomatoes - Retail	8877

```
#making the histogram
plt.figure(figsize=(10, 6))
plt.bar(query_result['commodity'], query_result['entry_count'], color='skyblue')
plt.xlabel('Commodity')
plt.ylabel('Entry Count')
plt.title('Top 5 Commodities by Entry Count')
plt.show()
```



We these results, we now need to get the average of each commodity for each year

```
sql_query = """
WITH ranked_commodities AS (
    SELECT
        commodity,
        COUNT(*) AS entry_count,
        ROW_NUMBER() OVER (ORDER BY COUNT(*) DESC) AS rank
    FROM
        food_prices
    GROUP BY
        commodity)
SELECT
    year,
    commodity,
    AVG(price_per_unit_usd) AS price_per_unit_usd FROM (
    SELECT
        fp.Year,
        AVG(fp.price) / ex.exchange_rate AS price_per_unit_usd,
        fp.commodity
    FROM
        food_prices fp
    JOIN
        exchange ex ON fp.currency_name = ex.code
    JOIN
        ranked_commodities rc ON fp.commodity = rc.commodity
    WHERE
        rc.rank <= 5
        AND fp.commodity IN ('Sugar - Retail', 'Wheat flour - Retail', 'Rice - Retail', 'Wheat - Retail', 'Tomatoes - Retail')
    GROUP BY
        fp.Year, fp.commodity
) AS subquery
GROUP BY
    year, commodity
ORDER BY
    year"""
query = pd.read_sql_query(sql_query, engine)
query
```

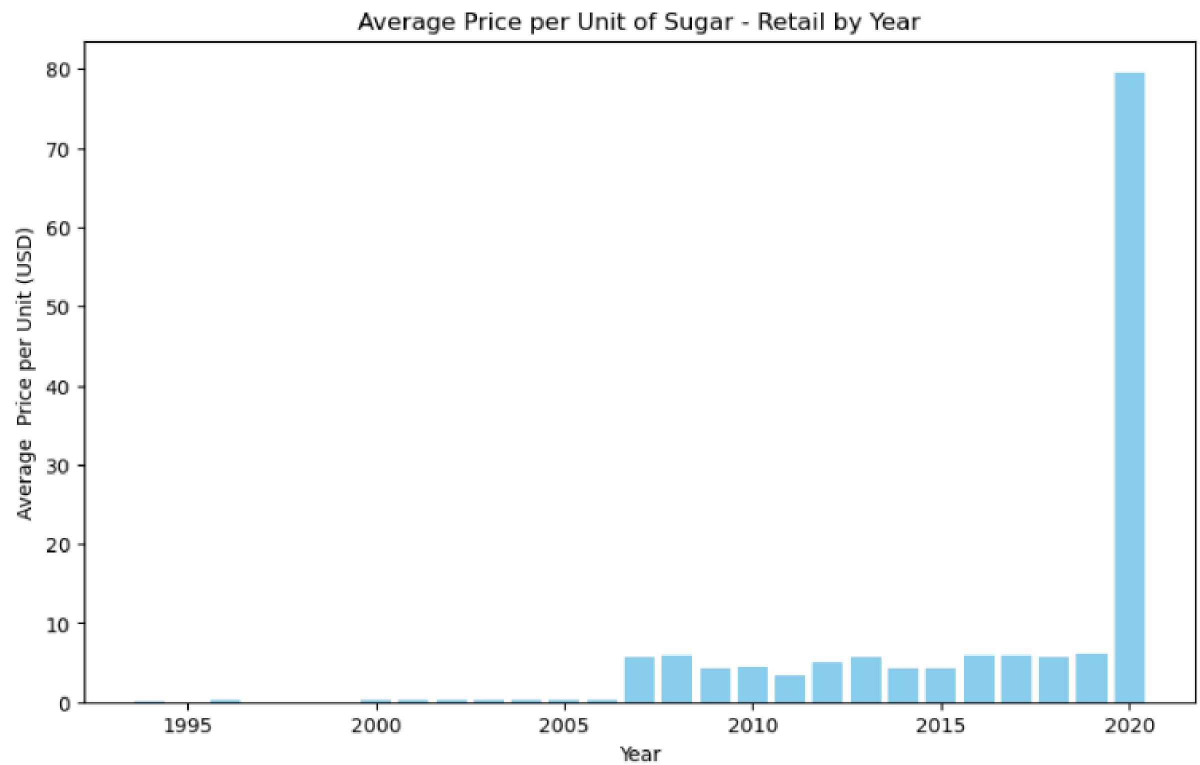
	year	commodity	price_per_unit_usd
0	1994	Rice - Retail	0.120137
1	1994	Sugar - Retail	0.220278
2	1994	Wheat - Retail	0.092489
3	1995	Rice - Retail	0.123400
4	1995	Wheat - Retail	0.088933
...	...	...	...
105	2020	Rice - Retail	75.316781
106	2020	Sugar - Retail	79.576457
107	2020	Tomatoes - Retail	4.578759
108	2020	Wheat - Retail	2.163293
109	2020	Wheat flour - Retail	6.132012

10 rows × 3 columns

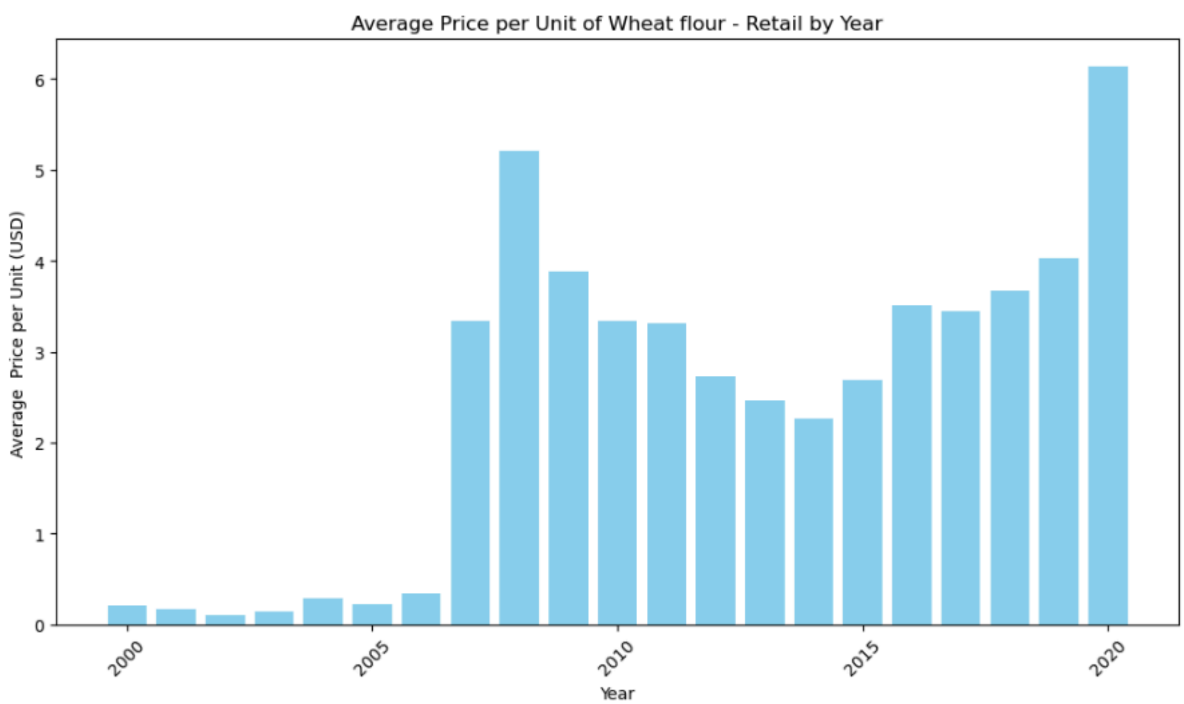
Now we start getting the histogram for each of the common commodities:

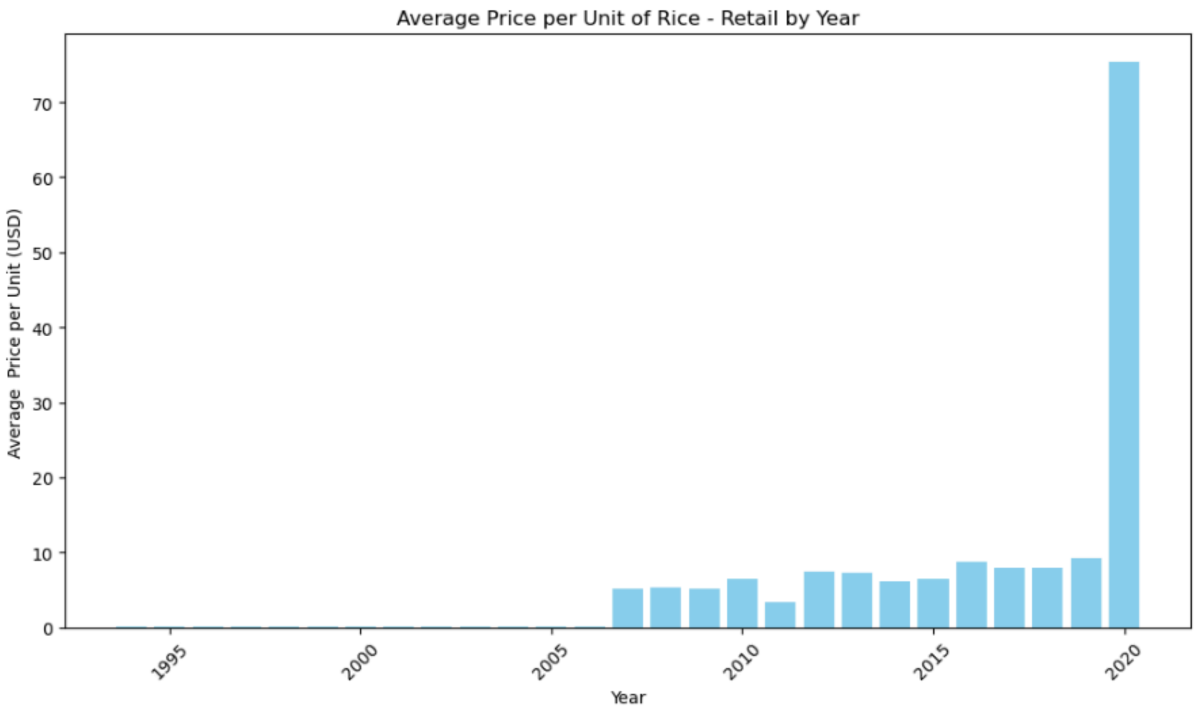
```
#Sugar Query
#OVER CLAUSE CITATION: "Database by Doug." (2017, Oct 12). SQL Ranking Functions: Part 1 The Over Clause [Video]. YouTube. https://www.youtube.com/watch?v=YdxyTMjpMMs
sql_query = """
WITH ranked_commodities AS (
    SELECT
        commodity,
        COUNT(*) AS entry_count,
        ROW_NUMBER() OVER (ORDER BY COUNT(*) DESC) AS rank
        food_prices
    GROUP BY
        commodity
)
SELECT
    year,
    commodity,
    AVG(price_per_unit_usd) AS price_per_unit_usd
FROM (
    SELECT
        fp.Year,
        AVG(fp.price) / ex.exchange_rate AS price_per_unit_usd,
        fp.commodity
    FROM
        food_prices fp
    JOIN
        exchange ex ON fp.currency_name = ex.code
    JOIN
        ranked_commodities rc ON fp.commodity = rc.commodity
    WHERE
 fp.commodity = 'Sugar - Retail'
    GROUP BY
        fp.Year, fp.commodity
) AS subquery
GROUP BY
    year, commodity
ORDER BY
    year
"""
```

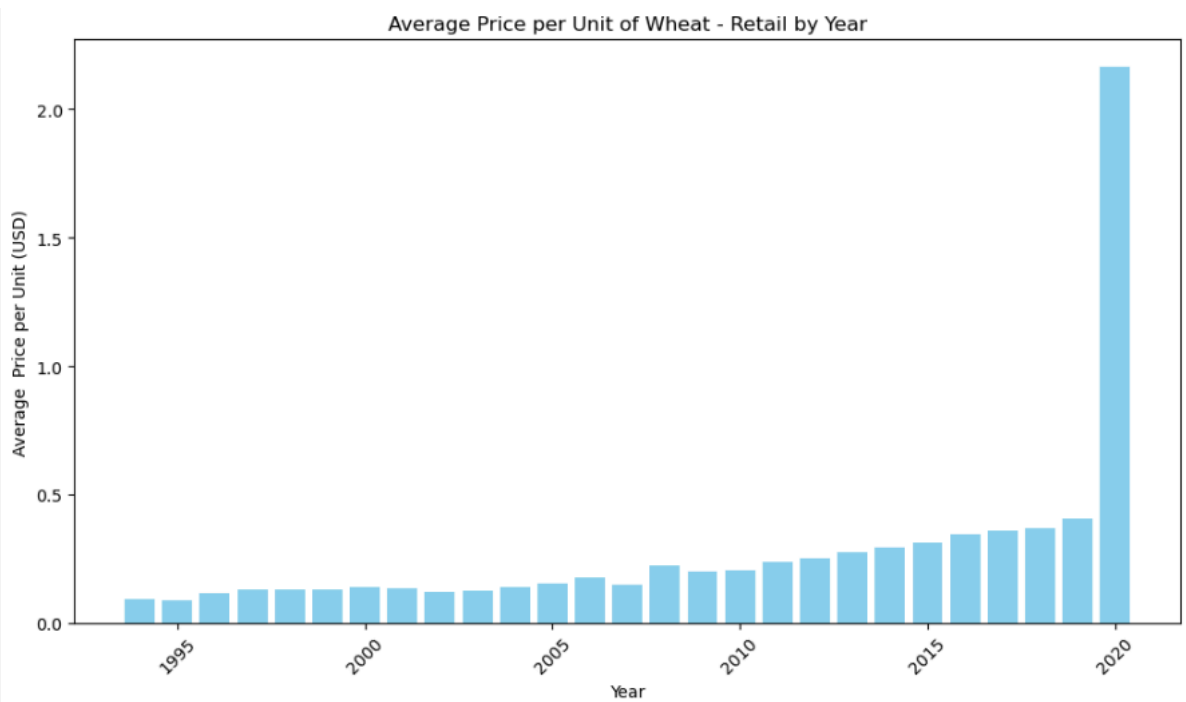
```
query_result = pd.read_sql_query(sql_query, engine)

plt.figure(figsize=(10, 6))
plt.bar(query_result['year'], query_result['price_per_unit_usd'], color='skyblue')
plt.xlabel('Year')
plt.ylabel('Average  Price per Unit (USD)')
plt.title('Average Price per Unit of Sugar - Retail by Year')
plt.show()
```

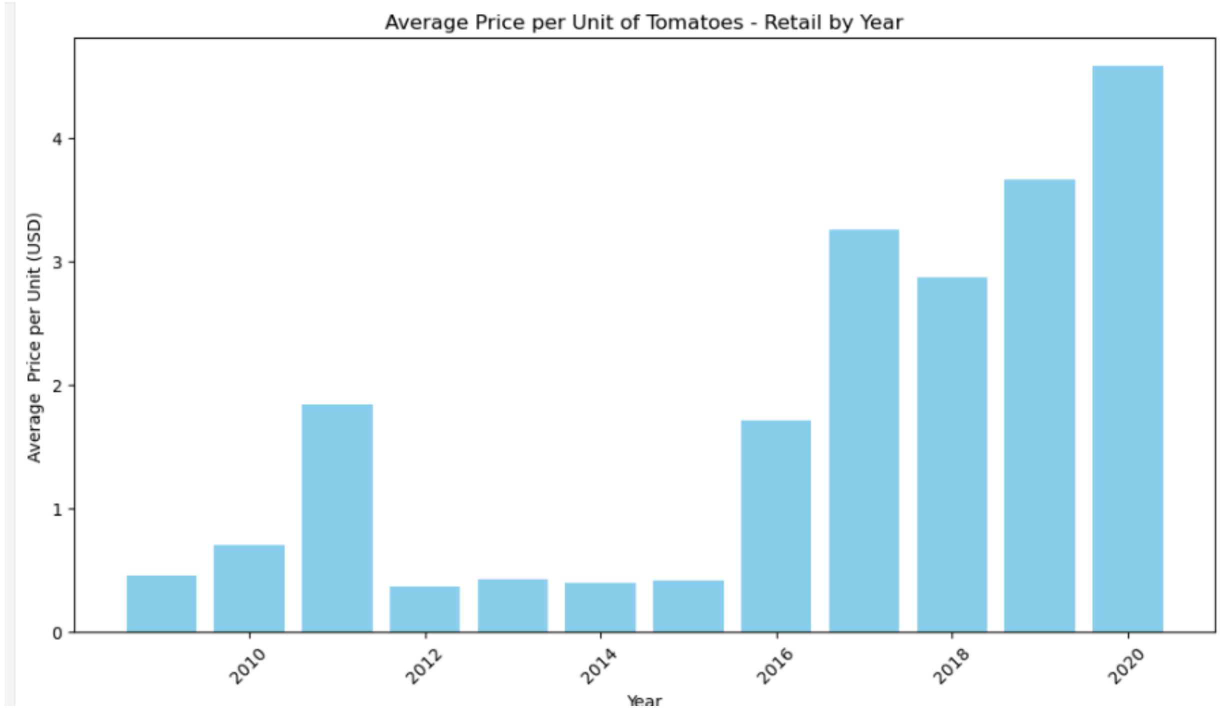


The query is the same for all the commodities, so I'll continue to show the visualizations.




Average Price per Unit of Rice - Retail by Year
Average Price per Unit (USD)
Year
0
10
20
30
40
50
60
70
1995
2000
2005
2010
2015
2020


Average Price per Unit of Wheat - Retail by Year
Average Price per Unit (USD)
Year
0.0
0.5
1.0
1.5
2.0
1995
2000
2005
2010
2015
2020

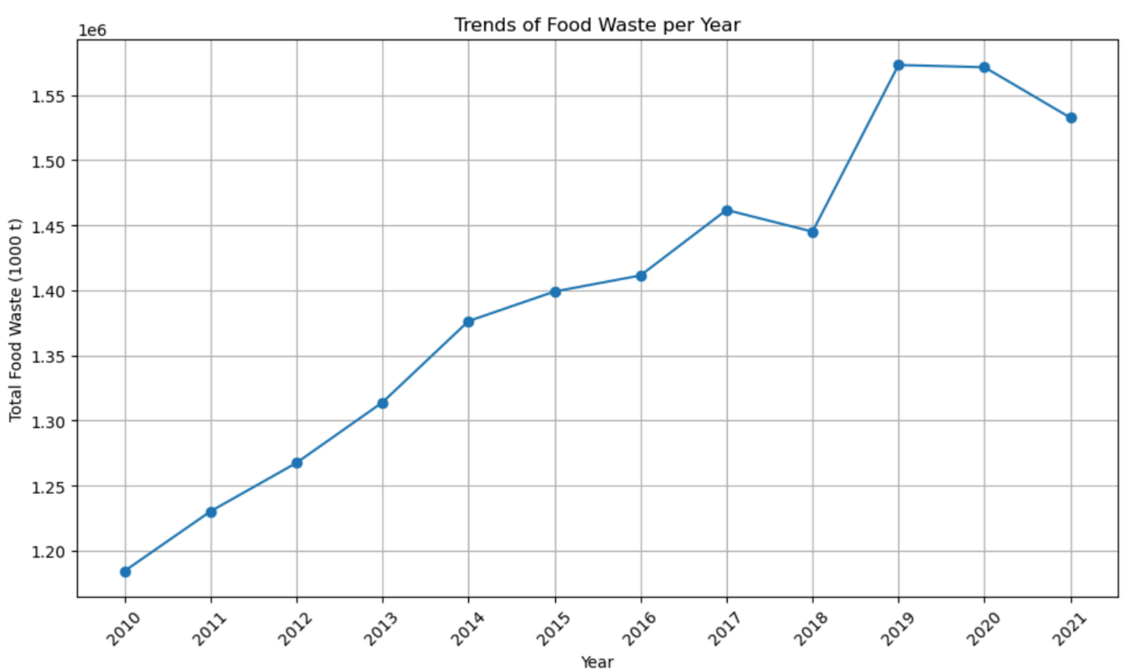


As expected for a continuation of our abstract, while there can be ups and downs, in general across the board the price of commodities are increasing. The common trend between these histograms is the massive spike in food price in 2020. The natural conclusion of this would be to suspect it had to do with the pandemic, so we made sure to research if this was the cause and get information from multiple sources. One of these was from the USDA ERS website. Reporting that in 2020, food-at-home prices increased 3.5 percent, while food-away-from-home prices increased 3.4 percent. This was a continuous growth in the year, as the Food and Agriculture Organization (FAO) of the United Nations reported global food prices continued to rise especially in the months June to August(USDA ERS - Summary Findings, 2021). The FAO also mentioned two other causes, where the seasonal demands were outpacing low inventory levels, and measures taken to contain the spread of coronavirus having an impact("Global food prices have been rising during the coronavirus pandemic, hitting food security - CNBC, 2020). While this can explain some reason for the spikes in this year, the number is still an extreme jump. We aren't ruling out the possibility that there could have been some sort of error in data extraction, given the strange consequences of the pandemic.

4.1.4 Have food waste changed over the years?

The following graph was built to explore the sum of food waste globally and see if there were any trends. This graph will show food waste in 1000 tons, and shows a steady increase globally over the years. One thing to note with this graph is that it does not include any household waste, this is all commercial losses that were recorded. Since food waste is increasing, there is a strong likelihood that it could have an impact on food prices.

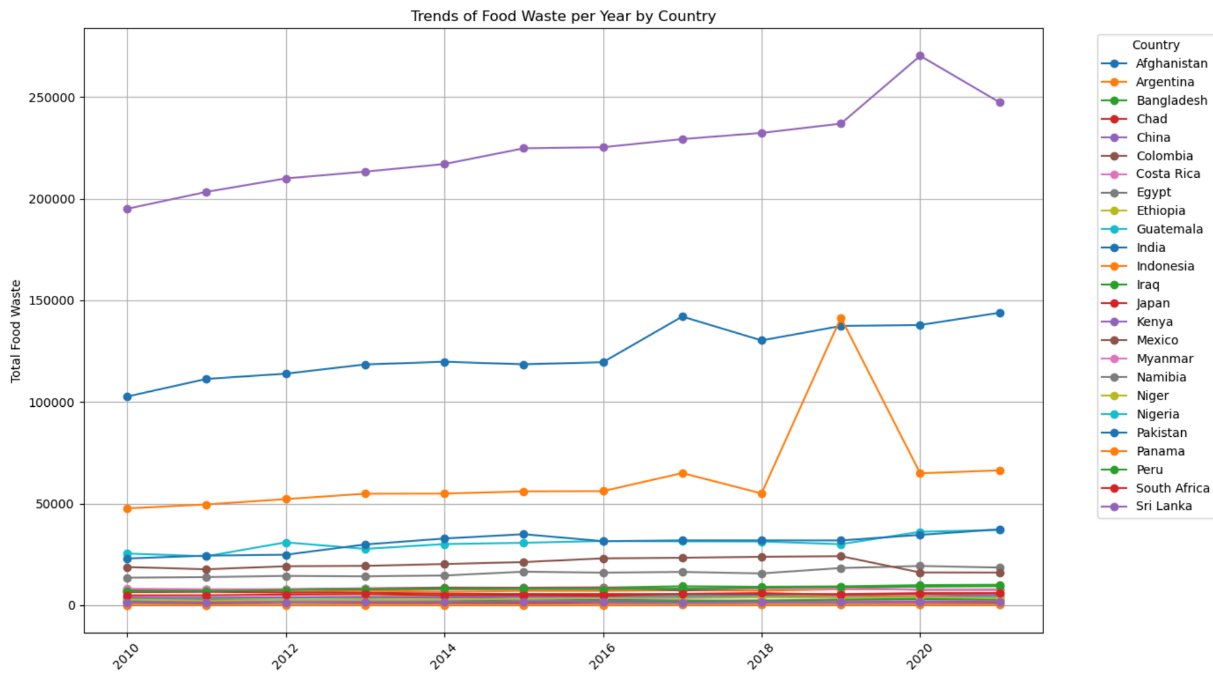
```
import matplotlib.pyplot as plt

#Query for sum of food waste globally
query = """
SELECT year, SUM(value) AS total_waste
FROM foodwaste1
GROUP BY year
ORDER BY year;
"""

with myconnection.cursor(dictionary=True) as cursor:
    cursor.execute(query)
    result = cursor.fetchall()

#Extracting years and total waste values
years = [row['year'] for row in result]
total_waste = [row['total_waste'] for row in result]

#Plotting the data
plt.figure(figsize=(10, 6))
plt.plot(years, total_waste, marker='o', linestyle='-')
plt.title('Trends of Food Waste per Year')
plt.xlabel('Year')
plt.ylabel('Total Food Waste (1000 t)')
plt.grid(True)
plt.xticks(years, rotation=45)
plt.tight_layout()

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



We wanted to see if there were any specific countries that had large trends in food waste over the years. As there is over 150+ countries in this dataset, we narrowed it down to countries of interest trying to hit as many of the continents as possible. We used these countries of interest to make our graph more manageable and to see if there were any trends. There is not many trends with this graph, there are slight increases with most of the countries, however a majority of the countries are too low to see any changes. China is the leading country with the most food waste, and there are various spikes in China, Indonesia and India, however the majority of the countries have not seen large trends.

```python
#Making the graph smaller by focusing on specific countries
countries_of_interest = [
    # Asia
    'India', 'China', 'Japan', 'Indonesia', 'Russia', 'Turkey', 'Syria',
    'Iraq', 'Pakistan', 'Afghanistan', 'Sri Lanka', 'Bangladesh',
    'Myanmar', 'Vietnam', 'Singapore',
    # Africa
    'Nigeria', 'Chad', 'Niger', 'South Africa', 'Namibia', 'Kenya',
    'Ethiopia', 'Egypt',
    # North America
    'Mexico',
    # South America
    'Argentina', 'Colombia', 'Peru', 'Bolivia', 'Guatemala',
    'Costa Rica', 'Panama'
]

# Country graph
query = """
SELECT area, year, SUM(value) AS TotalWaste
FROM foodwaste1
GROUP BY area, year
ORDER BY area, year;
"""

with myconnection.cursor(dictionary=True) as cursor:
    cursor.execute(query)
    result = cursor.fetchall()

# Organizing data by country
data_by_country = {}
for row in result:
    country = row['area']
    if country not in data_by_country:
        data_by_country[country] = {'year': [], 'TotalWaste': []}
    data_by_country[country]['year'].append(row['year'])
    data_by_country[country]['TotalWaste'].append(row['TotalWaste'])
```

```python
# Creating a plot for each country
plt.figure(figsize=(14, 8))

for country, data in data_by_country.items():
    if country in countries_of_interest:
        plt.plot(data['year'], data['TotalWaste'], marker='o', linestyle='-', label=country)

plt.title('Trends of Food Waste per Year by Country')
plt.xlabel('Year')
plt.ylabel('Total Food Waste')
plt.legend(title='Country', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()

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



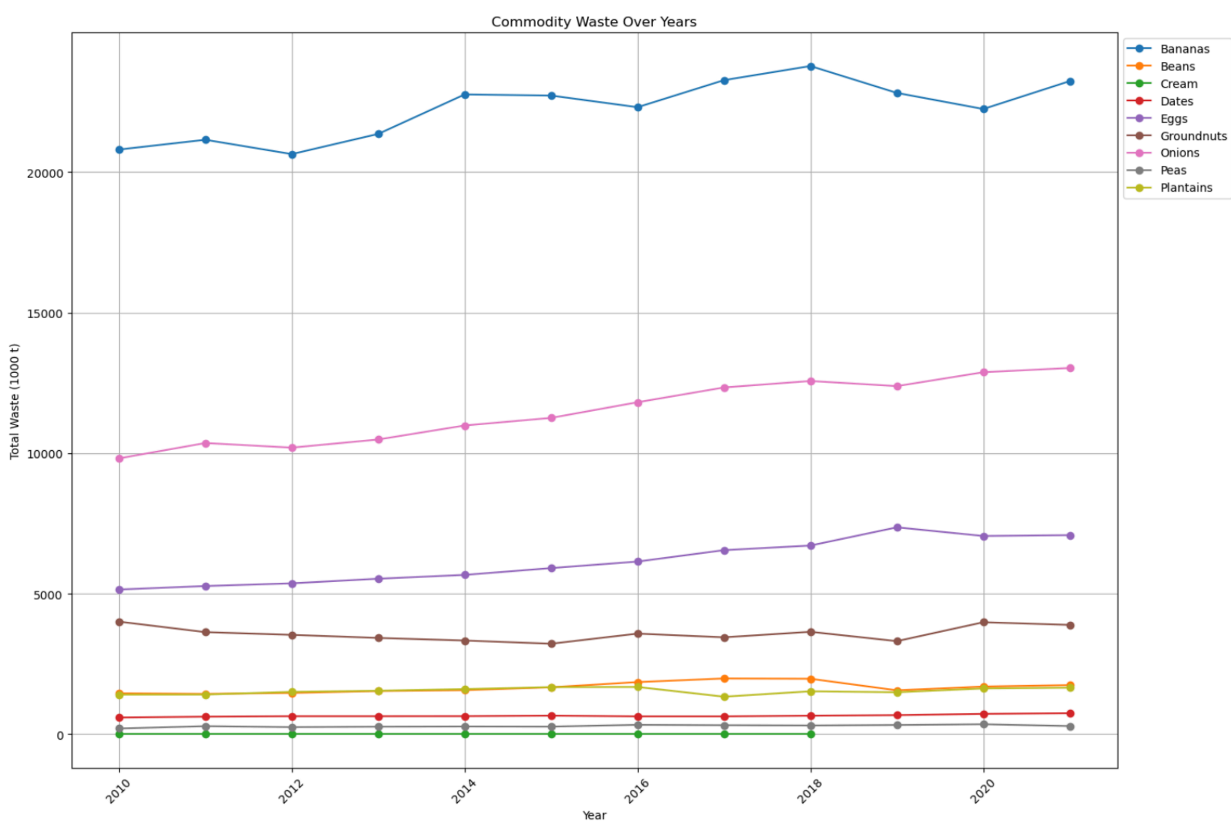
4.2 Guiding questions

4.2.1 How does Food waste impact Food prices?

We aimed to look at food prices and compare it to food waste globally. As food prices increase we wanted to see if there were any relationships between food prices and food waste, if food prices go up, is it because food waste goes up? Or does food waste decrease when food prices go up as people want to waste less food? As the dataset is quite large, we chose a few countries to examine this relationship between food waste and food prices.

We started with doing a simple analysis for food waste by commodity. We used a SQL query to create a graph outlining the commodities and summing the food waste.

```python
#Query for fetching aggregate food waste
query_waste = """
SELECT Year, Item, SUM(value) AS TotalWaste
FROM foodwaste1
WHERE area IN ('{}') AND item IN ('{}')
GROUP BY year, item
ORDER BY year, item;
""".format("', '".join(countries_of_interest), "', '".join(commodities_of_interest))

cursor.execute(query_waste)
waste_data = cursor.fetchall()

# Prepare the data
waste_by_commodity = {}
for row in waste_data:
    if row['Item'] not in waste_by_commodity:
        waste_by_commodity[row['Item']] = {'years': [], 'total_waste': []}
    waste_by_commodity[row['Item']]['years'].append(row['Year'])
    waste_by_commodity[row['Item']]['total_waste'].append(row['TotalWaste'])

# Plotting
plt.figure(figsize=(15, 10))

# Plot each commodity
for commodity, data in waste_by_commodity.items():
    plt.plot(data['years'], data['total_waste'], marker='o', linestyle='-', label=commodity)

plt.title('Commodity Waste Over Years')
plt.xlabel('Year')
plt.ylabel('Total Waste (1000 t)')
plt.legend(loc='upper left', bbox_to_anchor=(1, 1))  # Move the legend out of the plot
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()

plt.show()
```

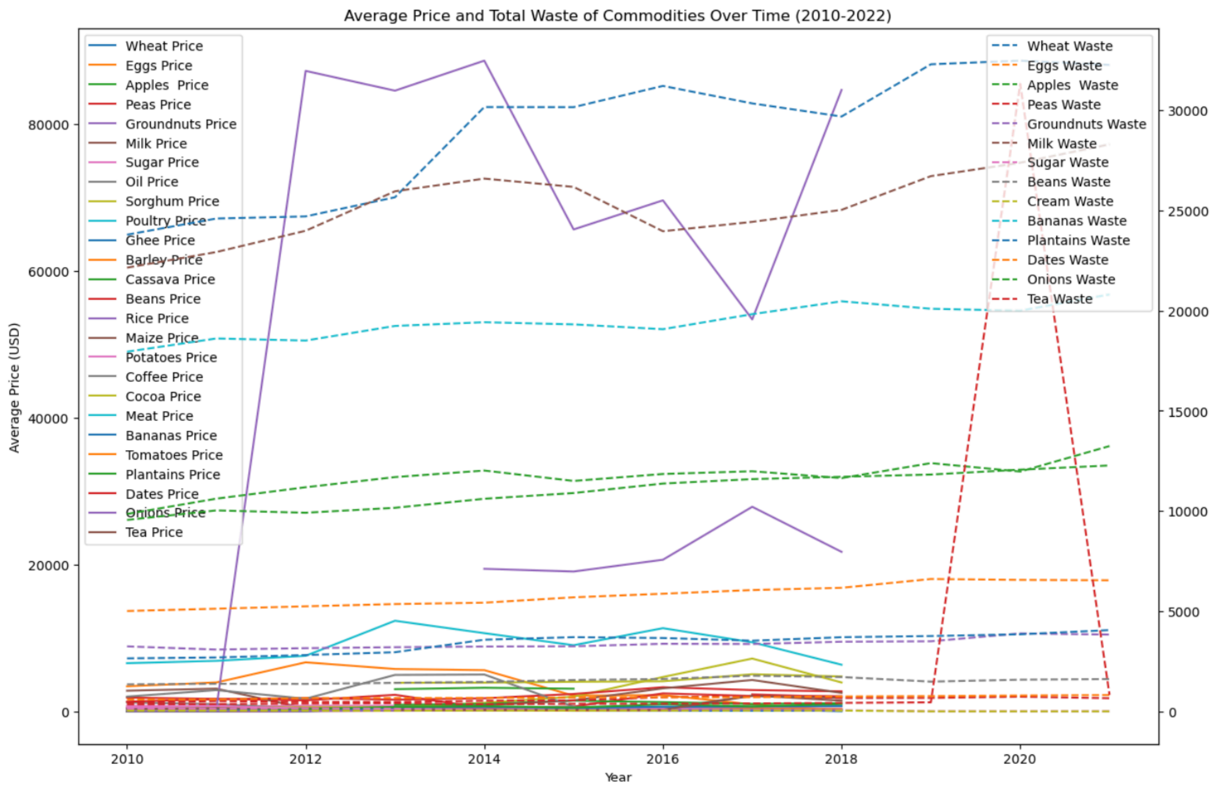


As you can see above, bananas are the highest recorded food that is wasted, followed by onions. Like the countries we graphed in the exploratory analysis section, we are able to see that there is a small but steady increase across most of the commodities looked at. Besides bananas, there are no trends that stand out commodity-wise.

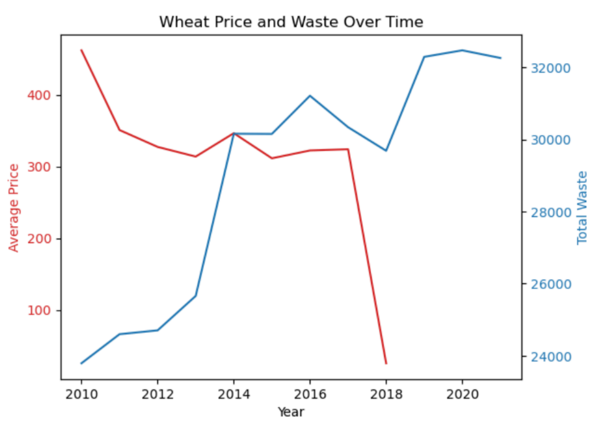
Once we saw that there were no trends that stood out, we wanted to look at commodity price vs food waste.

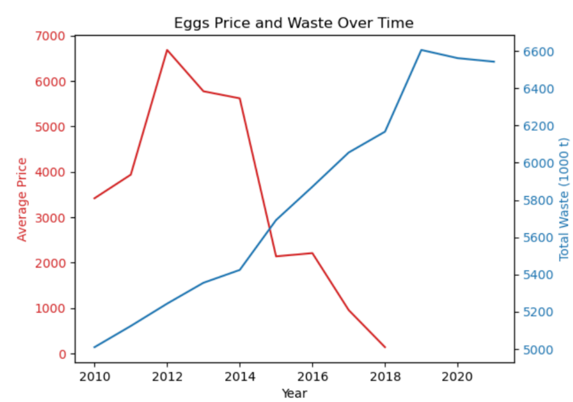
```python
# List of countries and commodities of interest
countries_of_interest = [
    'India', 'China', 'Japan', 'Indonesia', 'Russia', 'Turkey', 'Syria',
    'Iraq', 'Pakistan', 'Afghanistan', 'Sri Lanka', 'Bangladesh',
    'Myanmar', 'Vietnam', 'Singapore', 'Nigeria', 'Chad', 'Niger',
    'South Africa', 'Namibia', 'Kenya', 'Ethiopia', 'Egypt', 'Mexico',
    'Argentina', 'Colombia', 'Peru', 'Bolivia', 'Guatemala',
    'Costa Rica', 'Panama'
]

# Creating a cursor with dictionary output
cursor = myconnection.cursor(dictionary=True)

# Query to fetch combined data of waste and prices
query_combined = f"""
SELECT
    fw.year, SUM(fw.value) AS TotalWaste, AVG(fp.mp_price) AS AveragePrice
FROM
    foodwaste1 fw
JOIN
    foodprice2 fp ON fw.year = fp.mp_year
WHERE
    fw.area IN ({", ".join(f"'{country}'" for country in countries_of_interest)})
GROUP BY
    fw.year, fw.item
ORDER BY
    fw.year, fw.item;
"""

# Execute the query
cursor.execute(query_combined)
result_data = cursor.fetchall()

# Close the cursor and the database connection
cursor.close()
myconnection.close()
```

We then plotted it by commodity to see a visual of this query.



This graph has too much information on it for us to discern anything useful. We had to break it down into smaller clearer graphs that would better show us the relationship between food waste and food price. We took the sum of the food waste across all of our countries of interest and calculated the average food price and plotted a couple of the commodities that we were interested in on our graph.





As we can see, surprisingly, there doesn't seem to be much influence on food prices based on food waste. There are various fluctuations in the prices while there is a steady increase of food waste over the years. This could be due to some variance in the data that we are unaware of, but overall there doesn't seem to be much correspondence between the two.

4.2.2 How does GDP impact Food Prices?

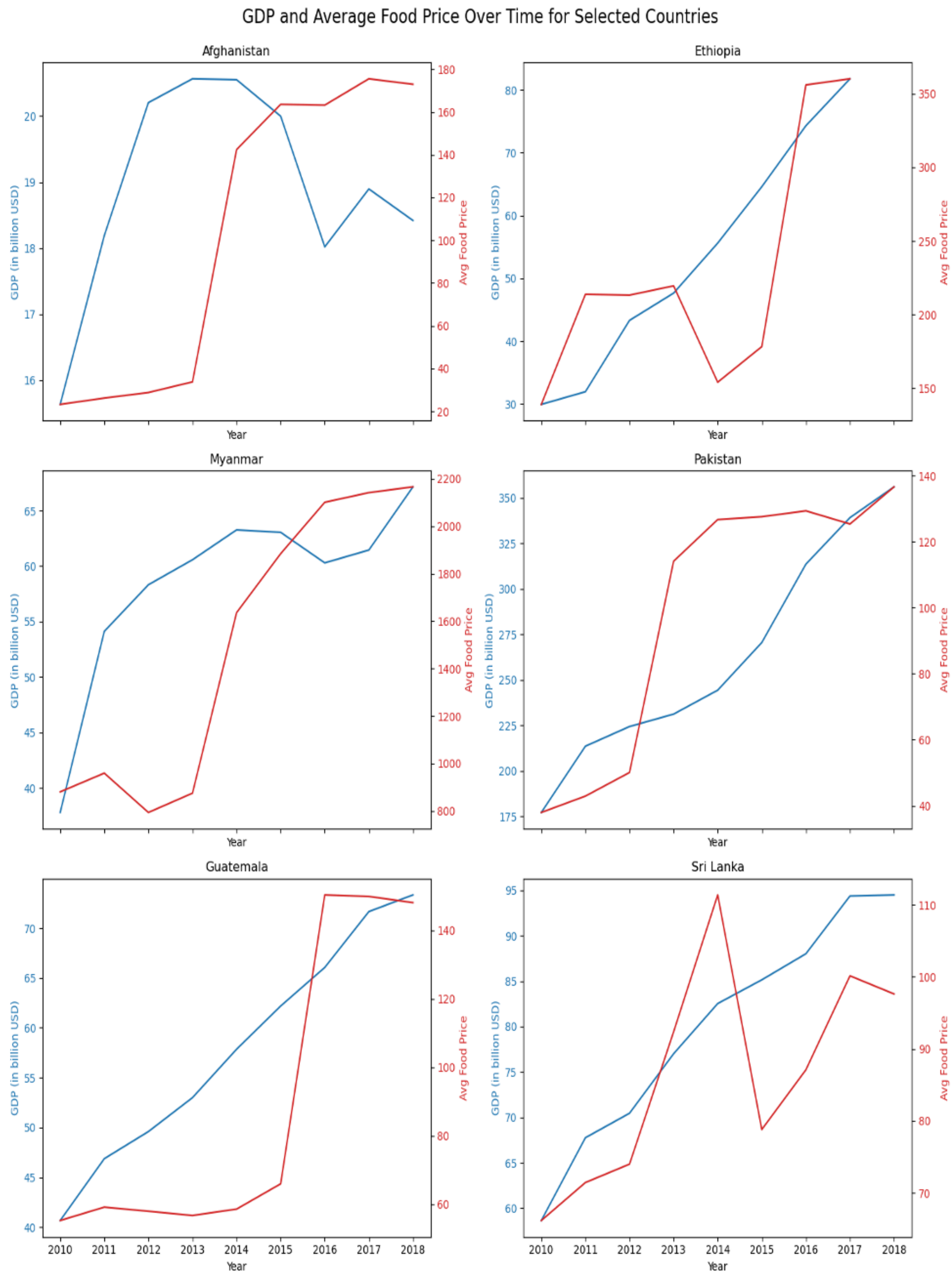
To begin my analysis, I aimed to extract and analyze data on GDP and average food prices for specific countries from 2010 to 2018. Using SQL queries, I first filtered the GDP data to include only the countries of interest that are common to both the GDP and food price datasets. The resulting filtered data was then written into a temporary table within the database for ease of access and manipulation.

Subsequently, I crafted an SQL query to join the transformed GDP data with the selected food price data on the basis of country and year. This query was designed to yield average food prices and GDP figures for each country for each year within the specified range. The JOIN operation was crucial for consolidating the two datasets, enabling me to compare and contrast the economic indicators directly.

The final output of this process provides a comprehensive view of the average food prices alongside GDP values. This joined data was then ready to be utilized for visual analysis and to explore potential correlations, trends, and patterns that could shed light on the economic and food security situation across the selected nations during the given timeframe.

```python
# Filtering for Countries we're interested in
countries_of_interest = ['India', 'China', 'Japan', 'Indonesia', 'Russia', 'Mexico',
                         'Argentina', 'Colombia', 'Peru', 'Bolivia', 'Guatemala',
                         'Costa Rica', 'Panama', 'Nigeria', 'Chad', 'Niger', 'South Africa',
                         'Namibia', 'Kenya', 'Ethiopia', 'Egypt', 'Iran', 'Turkey', 'Syria',
                         'Iraq', 'Pakistan', 'Afghanistan', 'Sri Lanka', 'Bangladesh',
                         'Myanmar', 'Vietnam', 'Singapore']
gdp_long_filtered = gdp_long[(gdp_long['Country'].isin(countries_of_interest)) & (gdp_long['Country'].isin(price_data_countries['adm0_name']))]
gdp_long_filtered.to_sql('gdp_long_filtered', con=engine, if_exists='replace', index=False)

sql_query = """
SELECT
    g.Country AS Country,
    g.Year AS Year,
    AVG(p.mp_price) AS Avg_Food_Price,
    g.GDP AS GDP
FROM
    gdp_long_filtered g
INNER JOIN
    price_data_selected_columns p
ON
    g.Country = p.adm0_name AND g.Year = p.mp_year
GROUP BY
    g.Country, g.Year;
"""

result = pd.read_sql(sql_query, engine)
print(result)
```

```
         Country  Year  Avg_Food_Price           GDP
0    Afghanistan  2010       23.258976  1.563384e+10
1    Afghanistan  2011       26.214087  1.819041e+10
2    Afghanistan  2012       28.804487  2.020357e+10
3    Afghanistan  2013       33.791324  2.056449e+10
4    Afghanistan  2014      142.392612  2.055058e+10
..           ...   ...             ...           ...
148    Sri Lanka  2014      111.372111  8.252854e+10
149    Sri Lanka  2015       78.785998  8.514096e+10
150    Sri Lanka  2016       87.077641  8.801228e+10
151    Sri Lanka  2017      100.131072  9.437624e+10
152    Sri Lanka  2018       97.613000  9.449387e+10

[153 rows x 4 columns]
```



The visual data presented across six countries from 2010 to 2018 reveal varied interactions between GDP and average food prices. In Afghanistan, GDP and food prices fluctuate without a clear pattern of correlation, suggesting that external factors, possibly socio-political or environmental, are at play influencing food prices. Ethiopia's trend lines ascend together, with food prices climbing sharply towards the latter years, which could signify economic growth coupled with rising inflation in food costs.

Myanmar's graphs depict a steady economic rise with sporadic food price changes, peaking in 2015, hinting at unique market or policy impacts on food prices.

In Pakistan, there's a strong congruence of increasing GDP and food prices, although the latter rises more steeply, indicating a potential inflationary trend or increased demand outpacing supply. Guatemala's data shows continuous economic growth, while food prices exhibit significant instability, particularly the surge around 2014-2015, which may reflect temporary market disruptions or policy shifts. Lastly, Sri Lanka's trends initially appear to move in tandem with notable increases in both metrics, despite the food prices showing considerable fluctuations and a dramatic decrease in 2015-2016, implying a complex interplay of domestic and global factors that affect food costs.

Each country's economic narrative, as told by these trends, underscores the multifaceted relationship between national economic health measured by GDP and the food prices experienced by consumers, driven by a myriad of factors beyond mere economic output. The complexity of these relationships indicates the necessity for a nuanced analysis that incorporates broader economic, policy, and environmental elements to fully understand and interpret the dynamics affecting food prices in the context of national economic performance.

In general, these visualizations provide an overview of the trends but do not establish causation. They suggest that while there is often a positive correlation between GDP and food prices, the relationship is not uniform across countries or years. The impact of GDP on food prices is complex and can be influenced by various factors, including monetary policy, import tariffs, subsidies, domestic agricultural production, and international commodity prices.

4.2.3 How does food imports impact Food prices?

In this part of the question, we will be trying to get a correlation between the food prices of India, Afghanistan, China, Indonesia, Pakistan, and Turkey with respect to the food imports made by the country. The years will vary from country to country because that is how the data was collected. I was unable to join the food imports data with global food data as my kernel was crashing due to having a lot of data. Following are the respective queries shown for India, where we got the results for Wheat prices in India from 1994 - 2020, later we created another query to get food imports for those years and then we plot the food imports and food prices with respect to years to see for visual correlation. We followed the same steps for the countries listed above.

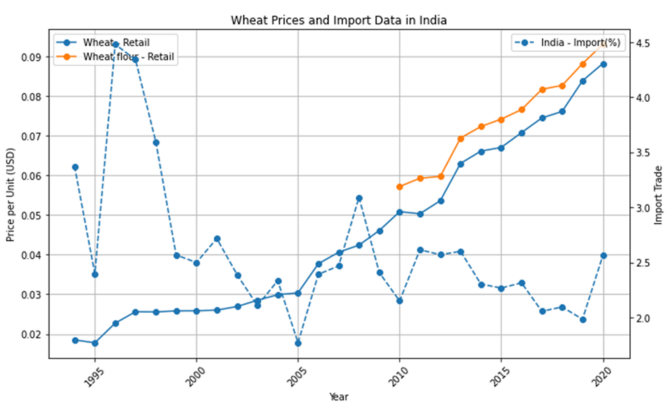
```
sql_query = """
-- Query to get wheat prices for India
SELECT
    fp.country,
    fp.currency_name,
    fp.commodity,
    fp.Quantity_unit,
    fp.Year,
    AVG(fp.price) / fp.Quantity / ex.exchange_rate AS price_per_unit_usd
FROM
    food_prices fp
JOIN
    exchange ex ON fp.currency_name = ex.code
WHERE
    fp.country = 'Bassas da India'
    AND fp.commodity LIKE '%wheat%'
GROUP BY
    fp.country, fp.currency_name, fp.commodity, fp.Quantity_unit, fp.Year;
"""

India = pd.read_sql_query(sql_query, engine)
India
```

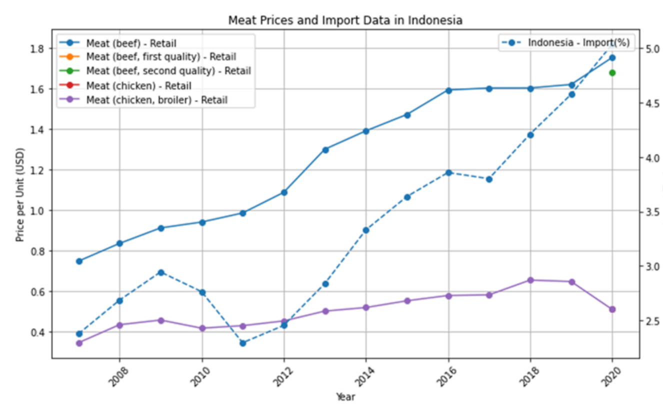
```
sql_query = """
SELECT *
FROM foodimp
WHERE Country = 'India'
AND Year BETWEEN 1994 AND 2020;
"""

India_import = pd.read_sql_query(sql_query, engine)
India_import
```

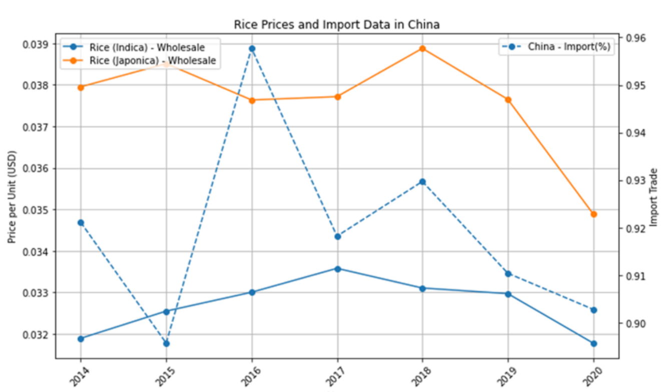
	country	currency_name	commodity	Quantity_unit	Year	avg_price_per_unit_usd
0	Bassas da India	INR	Wheat - Retail	KG	1994	0.018498
1	Bassas da India	INR	Wheat - Retail	KG	1995	0.017787
2	Bassas da India	INR	Wheat - Retail	KG	1996	0.022723
3	Bassas da India	INR	Wheat - Retail	KG	1997	0.025616
4	Bassas da India	INR	Wheat - Retail	KG	1998	0.025563
5	Bassas da India	INR	Wheat - Retail	KG	1999	0.025822
6	Bassas da India	INR	Wheat - Retail	KG	2000	0.025860
7	Bassas da India	INR	Wheat - Retail	KG	2001	0.026023
8	Bassas da India	INR	Wheat - Retail	KG	2002	0.026916
9	Bassas da India	INR	Wheat - Retail	KG	2003	0.028513
10	Bassas da India	INR	Wheat - Retail	KG	2004	0.029902
11	Bassas da India	INR	Wheat - Retail	KG	2005	0.030383
12	Bassas da India	INR	Wheat - Retail	KG	2006	0.037725
13	Bassas da India	INR	Wheat - Retail	KG	2007	0.040616
14	Bassas da India	INR	Wheat - Retail	KG	2008	0.042419
15	Bassas da India	INR	Wheat - Retail	KG	2009	0.046109
16	Bassas da India	INR	Wheat - Retail	KG	2010	0.050841
17	Bassas da India	INR	Wheat - Retail	KG	2011	0.050342
18	Bassas da India	INR	Wheat - Retail	KG	2012	0.053609
19	Bassas da India	INR	Wheat - Retail	KG	2013	0.063039
20	Bassas da India	INR	Wheat - Retail	KG	2014	0.066131
21	Bassas da India	INR	Wheat - Retail	KG	2015	0.067058

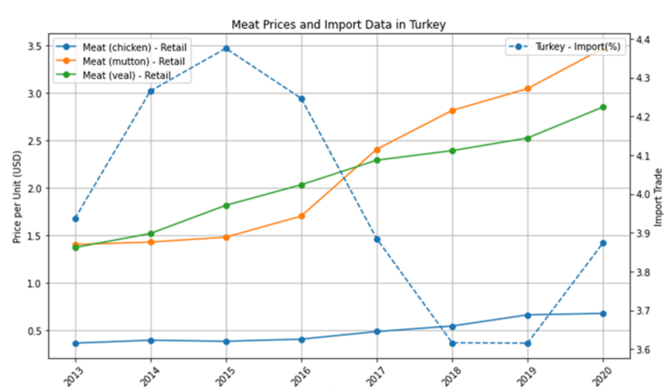
As can be seen from the below graphs, countries like India (Wheat) and Turkey (Meat) do not show correlation between food imports and food prices, one of the reasons for this anomaly for India is that it is the second highest producer of wheat in the world (Wheat). Countries like China (Rice), Indonesia (Meat), Pakistan (Wheat) and Afghanistan (Wheat) are following the food imports trend linearly as seen from the following graphs. We can say that to some extent food imports in a country do have an impact on food prices. When food imports increase, food prices increase, they have a positive linear growth.

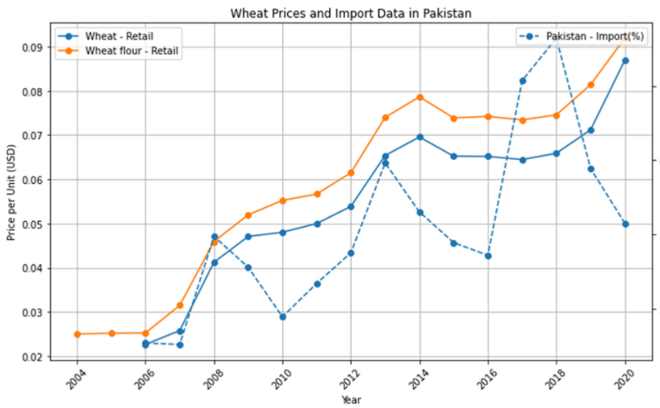
```python
import matplotlib.pyplot as plt

# Plot wheat prices and import data on the same graph
fig, ax1 = plt.subplots(figsize=(10, 6))

# Plot wheat prices
for wheat_type in India['commodity'].unique():
    wheat_data = India[India['commodity'] == wheat_type]
    ax1.plot(wheat_data['Year'], wheat_data['avg_price_per_unit_usd'], marker='o', label=wheat_type)

ax1.set_xlabel('Year')
ax1.set_ylabel('Price per Unit (USD)')
ax1.tick_params(axis='x', rotation=45)
ax1.legend(loc='upper left')

# Create a twin axis for import data
ax2 = ax1.twinx()
for column in India_import.columns[2:]:
    ax2.plot(India_import['Year'], India_import[column], marker='o', label='India - ' + column, linestyle='--')

ax2.set_ylabel('Import Trade')
ax2.tick_params(axis='y')
ax2.legend(loc='upper right')  # Add legend for import data

# Add title and grid
plt.title('Wheat Prices and Import Data in India')
ax1.grid(True)
plt.tight_layout()
plt.show()
```

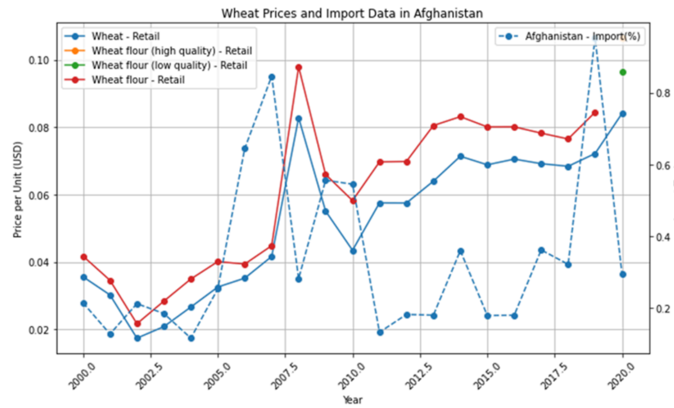












4.2.4 How does oil impact Food prices?

This question has been made to understand and make interpretations based on the correlation the crude oil dataset could possibly have with the main Food Prices dataset. Our interest here an interval of years

```
#In the year of 2019, The year before the pandemic(year before covid). For the common commodity of sugar, which were the top 5 countries in terms of cost
sql_query = """
SELECT
    fp.country,
    AVG(fp.price) / ex.exchange_rate AS price_per_unit_usd
FROM
    food_prices fp
JOIN
    exchange ex ON fp.currency_name = ex.code
WHERE
    fp.commodity = 'Sugar - Retail'
    AND fp.Year = 2019
GROUP BY
    fp.country
ORDER BY
    price_per_unit_usd DESC
LIMIT 5
"""
result = pd.read_sql_query(sql_query, engine)
result
```

	country	price_per_unit_usd
0	Nigeria	1.585008
1	Turkey	1.355898
2	Ethiopia	1.087488
3	Indonesia	0.928027
4	Iraq	0.830429

before the pandemic in this case:

For each of these countries, we want to pick a common commodity to use to see their growth in these 5 respective countries throughout the year. We start with looking at sugar.

```
#For the following top countries, show the trends for the sugar commodity from 2000-2019
sql_query = """

SELECT
    fp.country,
    fp.Year,
    AVG(fp.price) / ex.exchange_rate AS price_per_unit_usd
FROM
    food_prices fp
JOIN
    exchange ex ON fp.currency_name = ex.code
WHERE
    fp.commodity = 'Sugar - Retail'
    AND fp.Year BETWEEN 2000 AND 2019
    AND fp.country IN ('Nigeria', 'Turkey', 'Ethiopia', 'Indonesia', 'Iraq')
GROUP BY
    fp.country, fp.Year
ORDER BY
    fp.country, fp.Year DESC, price_per_unit_usd DESC
"""

result = pd.read_sql_query(sql_query, engine)
result
```

	country	Year	price_per_unit_usd
0	Ethiopia	2019	1.087488
1	Ethiopia	2018	1.149081
2	Ethiopia	2017	1.151026
3	Ethiopia	2016	0.927820
4	Ethiopia	2015	0.941503
5	Ethiopia	2014	0.746434
6	Indonesia	2019	0.928027
7	Indonesia	2018	0.926147
8	Indonesia	2017	1.010682
9	Indonesia	2016	1.084875
10	Indonesia	2015	0.941330
11	Indonesia	2014	0.880559
12	Indonesia	2013	0.937238
13	Indonesia	2012	0.893924
14	Indonesia	2011	0.808341
15	Indonesia	2010	0.811337
16	Indonesia	2009	0.640721
17	Indonesia	2008	0.488523
18	Indonesia	2007	0.490882
19	Iraq	2019	0.830429
20	Iraq	2018	0.920520
21	Iraq	2017	0.930681
22	Iraq	2016	0.983377
23	Iraq	2015	0.910564
24	Iraq	2014	1.120160
25	Iraq	2013	1.165844
26	Iraq	2012	1.153828
27	Nigeria	2019	1.585008
28	Nigeria	2018	2.123482
29	Nigeria	2017	2.703575
30	Turkey	2019	1.355898
31	Turkey	2018	1.275463
32	Turkey	2017	1.172027
33	Turkey	2016	1.144064
34	Turkey	2015	1.055525
35	Turkey	2014	1.036633
36	Turkey	2013	0.973740

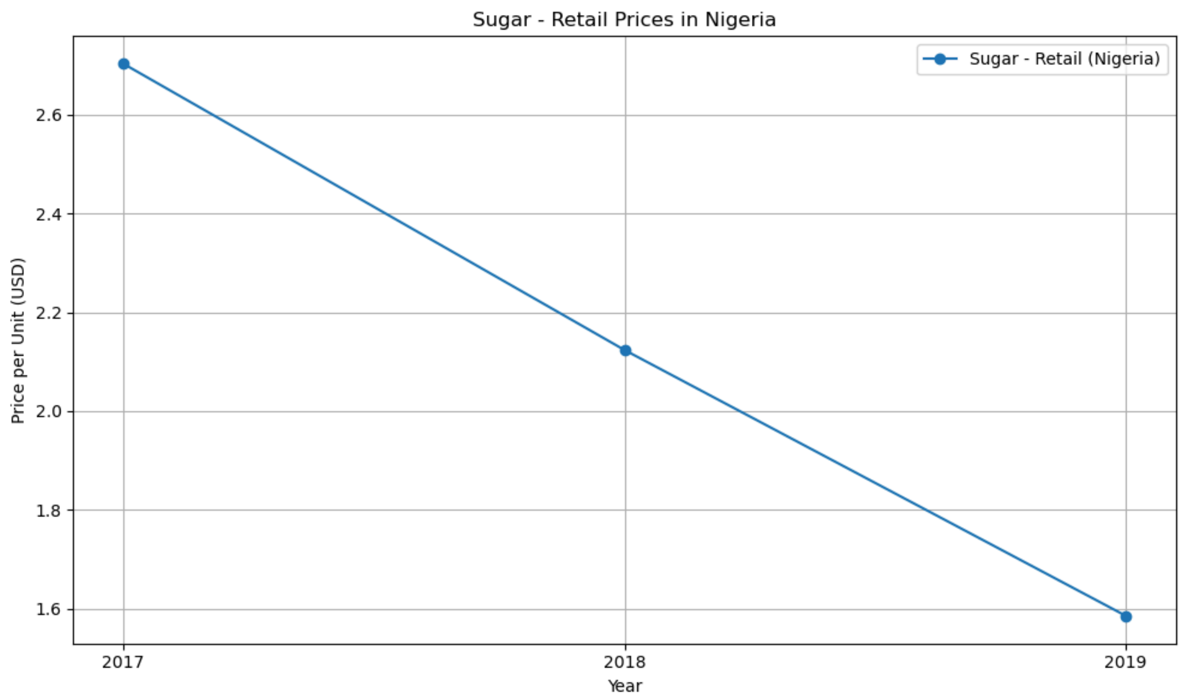
Next is to see how this information has trended with the crude oil price dataset.

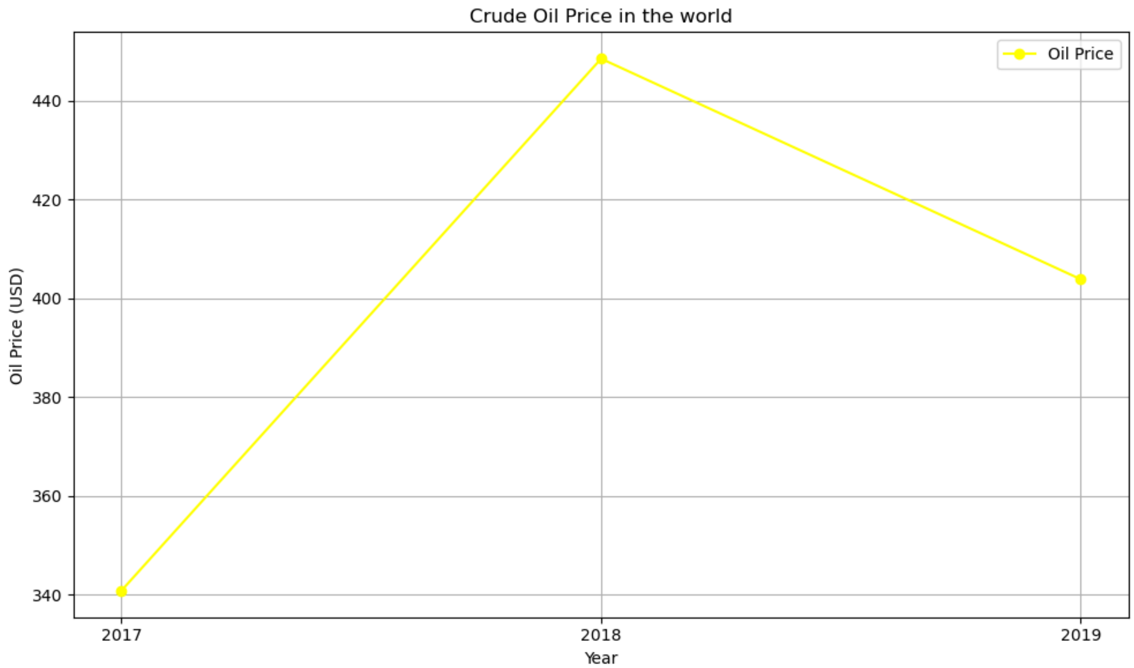
```
sql_query = """

SELECT
    fp.country,
    fp.Year,
    AVG(fp.price) / ex.exchange_rate AS sugar_price_per_unit_usd,
    op.`Oil Price`
FROM
    food_prices fp
JOIN
    exchange ex ON fp.currency_name = ex.code
LEFT JOIN
    oil_prices op ON fp.Year = op.Year
WHERE
    fp.commodity = 'Sugar - Retail'
    AND fp.Year BETWEEN 2000 AND 2019
    AND fp.country IN ('Nigeria', 'Turkey', 'Ethiopia', 'Indonesia', 'Iraq')
GROUP BY
    fp.country, fp.Year, op.`Oil Price`
ORDER BY
    fp.country, fp.Year DESC, sugar_price_per_unit_usd DESC
"""

result3 = pd.read_sql_query(sql_query, engine)
result3
```

	country	Year	sugar_price_per_unit_usd	Oil Price
0	Ethiopia	2019	1.087488	403.87164
1	Ethiopia	2018	1.149081	448.52600
2	Ethiopia	2017	1.151026	340.85962
3	Ethiopia	2016	0.927820	275.07916
4	Ethiopia	2015	0.941503	329.50226
5	Ethiopia	2014	0.746434	622.35060
6	Indonesia	2019	0.928027	403.87164
7	Indonesia	2018	0.926147	448.52600
8	Indonesia	2017	1.010682	340.85962
9	Indonesia	2016	1.084875	275.07916
10	Indonesia	2015	0.941330	329.50226
11	Indonesia	2014	0.880559	622.35060
12	Indonesia	2013	0.937238	683.44030
13	Indonesia	2012	0.893924	702.38007
14	Indonesia	2011	0.808341	699.77545
15	Indonesia	2010	0.811337	500.01102
16	Indonesia	2009	0.640721	387.89993
17	Indonesia	2008	0.488523	611.72064
18	Indonesia	2007	0.490882	455.31280
19	Iraq	2019	0.830429	403.87164
20	Iraq	2018	0.920520	448.52600
21	Iraq	2017	0.930681	340.85962
22	Iraq	2016	0.983377	275.07916
23	Iraq	2015	0.910564	329.50226
24	Iraq	2014	1.120160	622.35060
25	Iraq	2013	1.165844	683.44030
26	Iraq	2012	1.153828	702.38007
27	Nigeria	2019	1.585008	403.87164
28	Nigeria	2018	2.123482	448.52600
29	Nigeria	2017	2.703575	340.85962
30	Turkey	2019	1.355898	403.87164
31	Turkey	2018	1.275463	448.52600
32	Turkey	2017	1.172027	340.85962
33	Turkey	2016	1.144064	275.07916
34	Turkey	2015	1.055525	329.50226
35	Turkey	2014	1.036633	622.35060
36	Turkey	2013	0.973740	683.44030

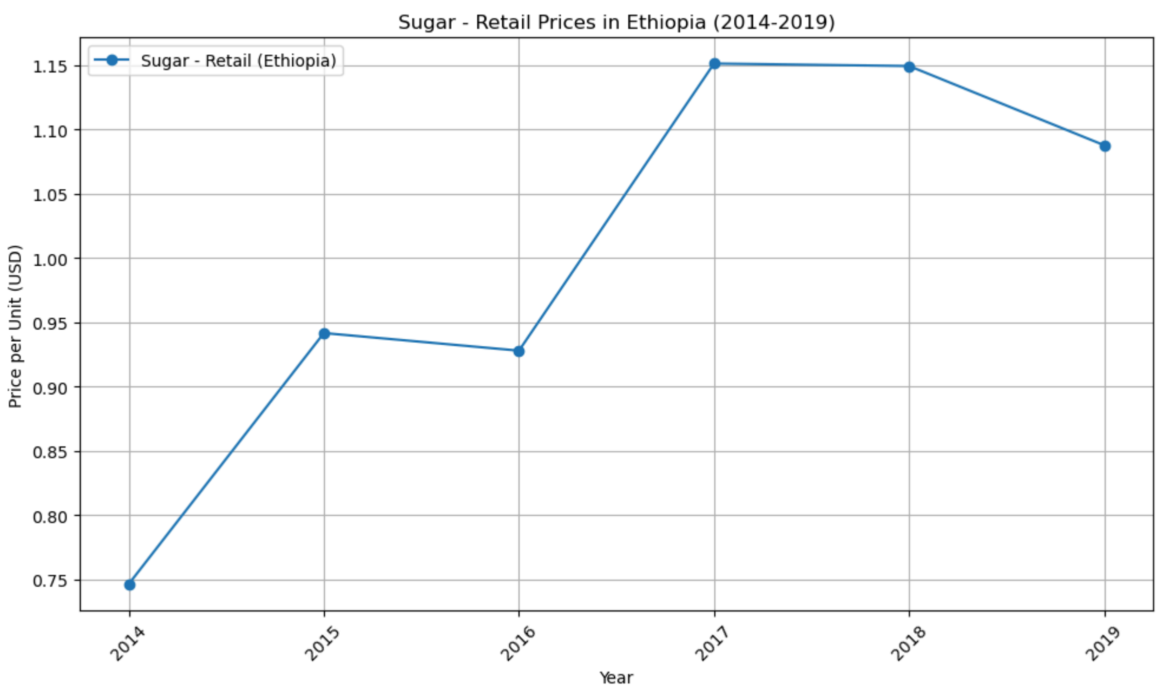
We then continue to visualize this data for each of the countries. Since Nigeria was initially at the top, we will start with it.

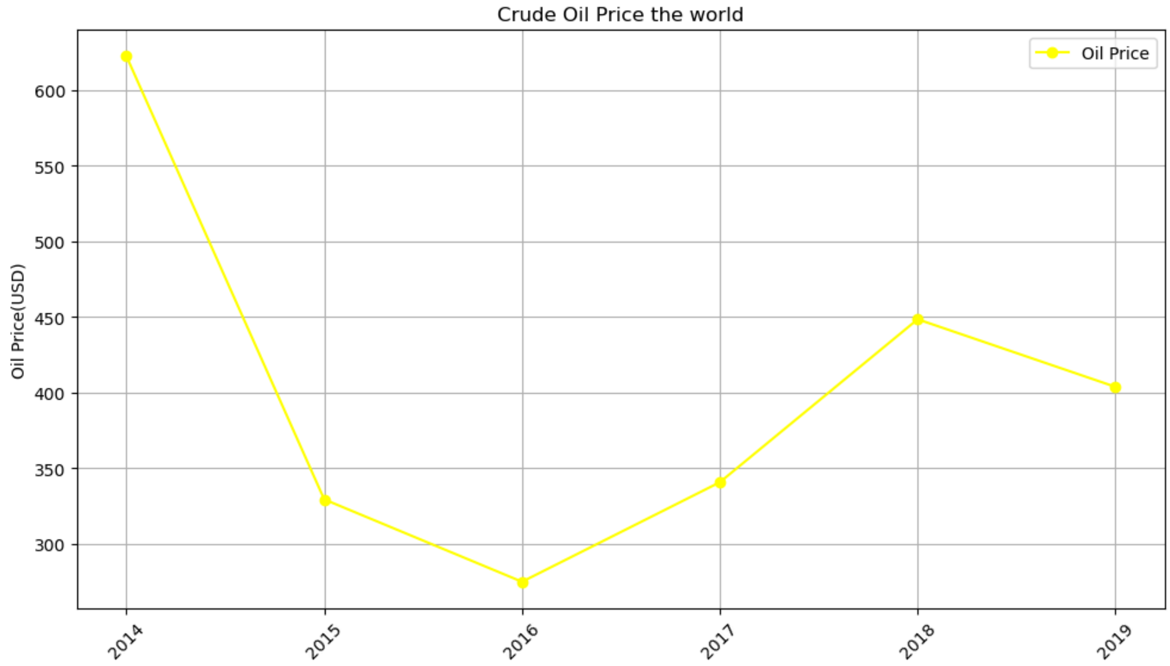
```
#Graph the results for Nigeria
import matplotlib.pyplot as plt
Nigeria_data = result3[result3['country'] == 'Nigeria']
plt.figure(figsize=(10, 6))
plt.plot(Nigeria_data['Year'], Nigeria_data['sugar_price_per_unit_usd'], marker='o', label='Sugar - Retail (Nigeria)')
plt.title('Sugar - Retail Prices in Nigeria')
plt.xlabel('Year')
plt.ylabel('Price per Unit (USD)')
plt.xticks(range(int(min(Nigeria_data['Year'])), int(max(Nigeria_data['Year'])) + 1))
plt.legend()
plt.grid(True)
plt.show()

#plot for oil
plt.figure(figsize=(10, 6))
plt.plot(Nigeria_data['Year'], Nigeria_data['Oil Price'], marker='o', label='Oil Price', color='yellow')
plt.title('Crude Oil Price in the world')
plt.xlabel('Year')
plt.ylabel('Oil Price (USD)')
plt.xticks(range(int(min(Nigeria_data['Year'])), int(max(Nigeria_data['Year'])) + 1))
plt.legend()
plt.grid(True)
plt.show()
```



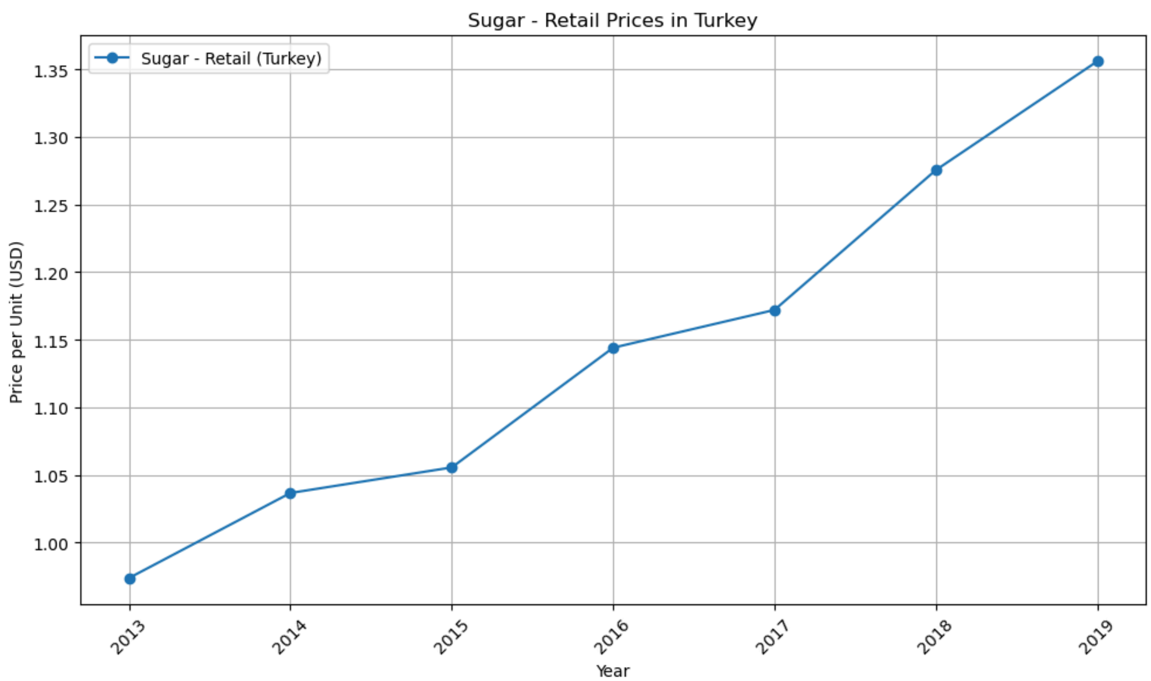


I will proceed to the following countries. They all use the same code with the name being the only difference, so I will show the visualizations.

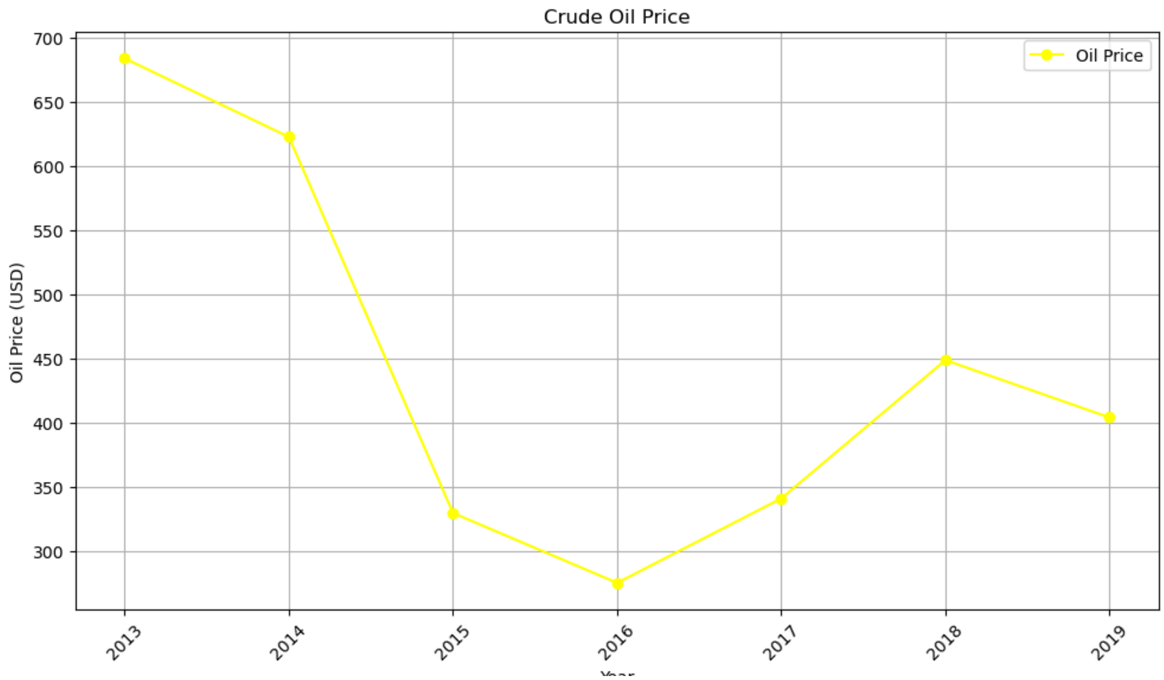


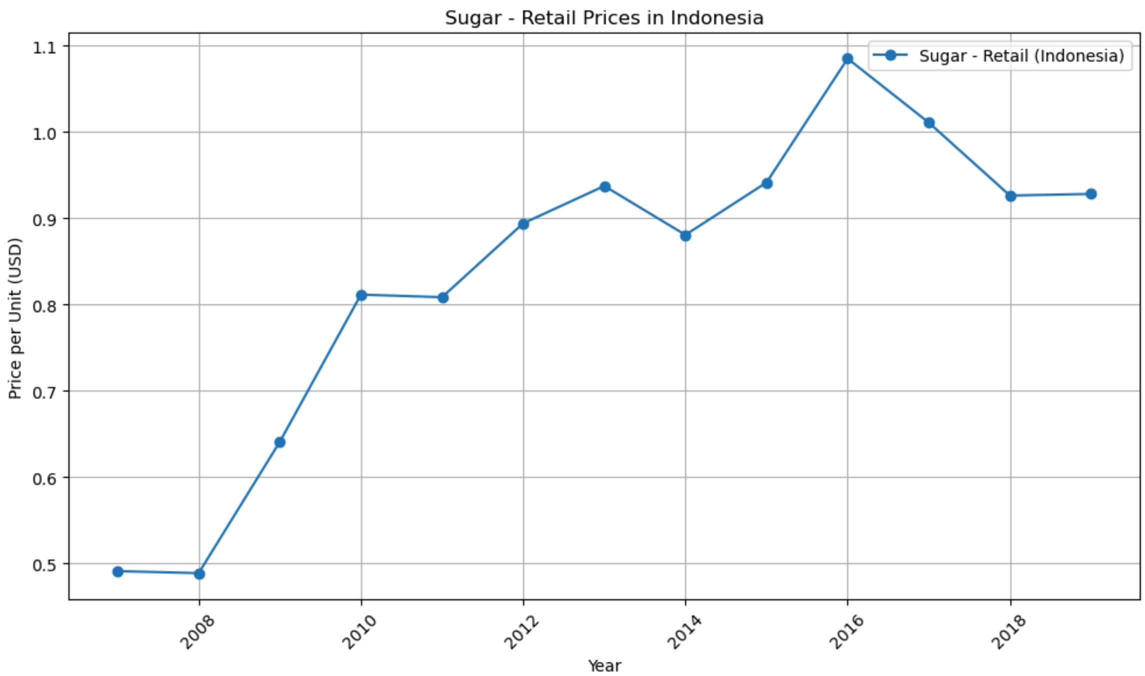


Crude Oil Price the world
Oil Price
Oil Price(USD)
600
550
500
450
400
350
300
2014
2015
2016
2017
2018
2019

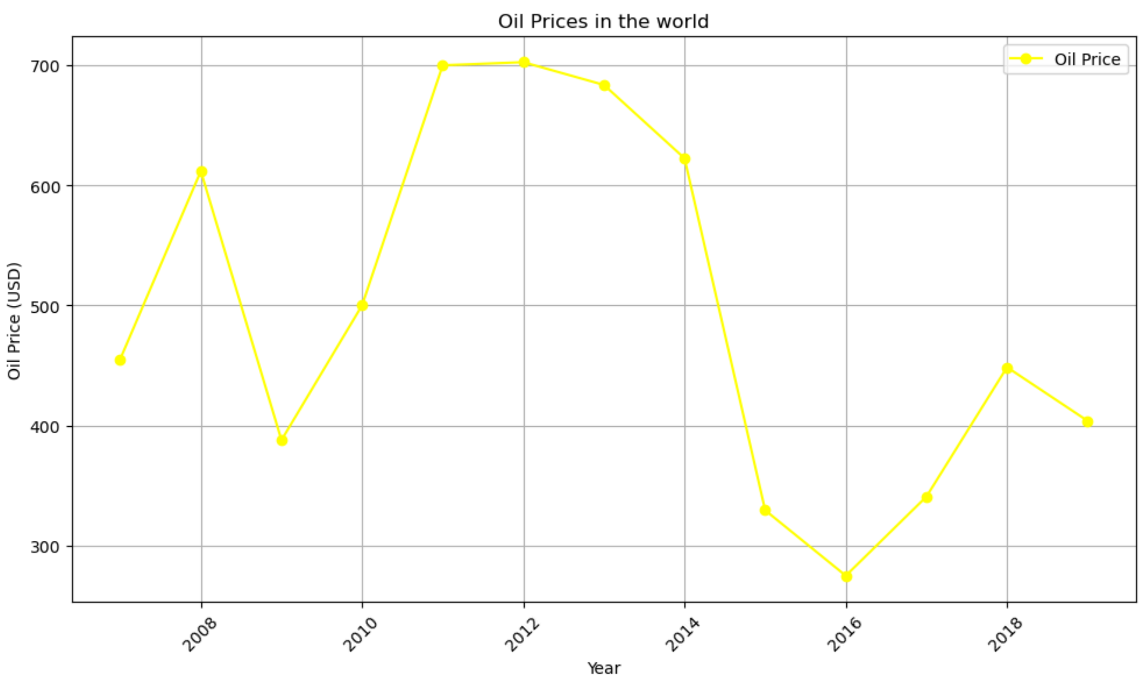


Sugar - Retail Prices in Turkey
Sugar - Retail (Turkey)
Price per Unit (USD)
1.35
1.30
1.25
1.20
1.15
1.10
1.05
1.00
2013
2014
2015
2016
2017
2018
2019
Year

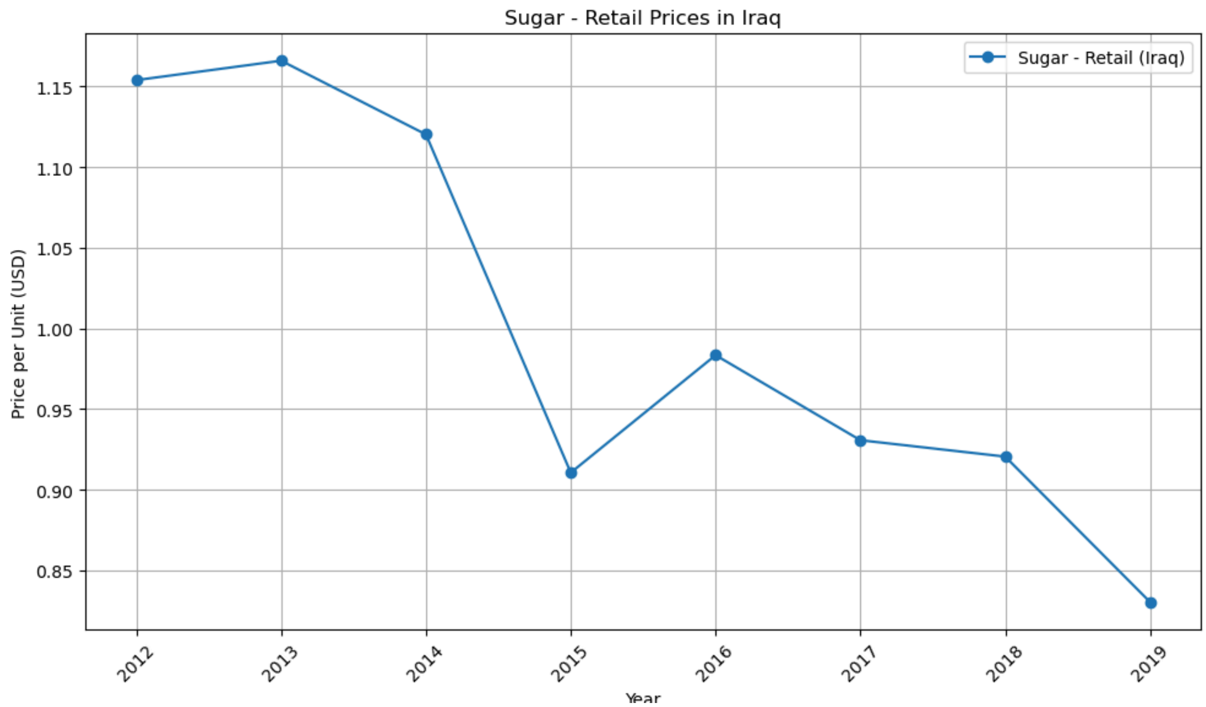

Crude Oil Price
Oil Price
Oil Price (USD)
700
650
600
550
500
450
400
350
300
2013
2014
2015
2016
2017
2018
2019

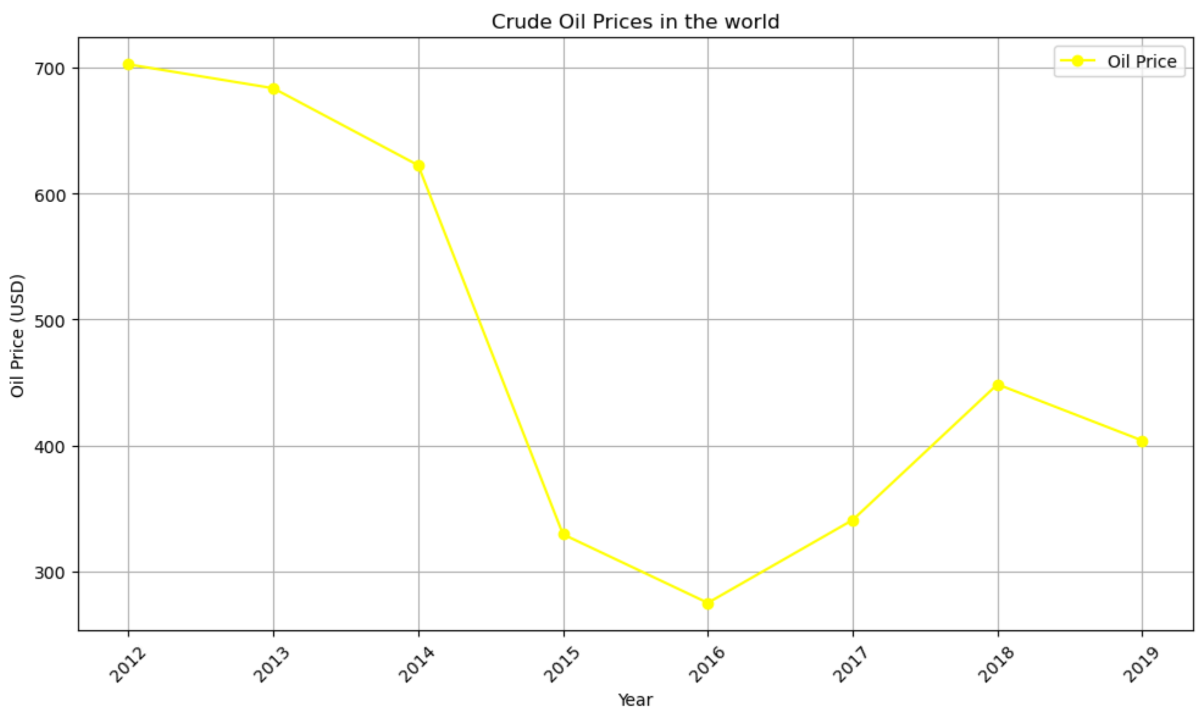


Sugar - Retail Prices in Indonesia
Sugar - Retail (Indonesia)
Price per Unit (USD)
1.1
1.0
0.9
0.8
0.7
0.6
0.5
2008
2010
2012
2014
2016
2018
Year



Oil Prices in the world
Oil Price
Oil Price (USD)
700
600
500
400
300
2008
2010
2012
2014
2016
2018
Year





Discussion: There are many mixed results analyzing the corresponding visualizations. All of these countries seem to follow random trends amongst these two variables, and in doing so it would seem we can conclude that there is zero correlation between them. Interestingly enough though, while it's not one to one Iraq seems to have the most visually prominent relationship in some areas. Both the sugar price and oil follow similar trends up until 2016, and the pattern is evident as well from 2018 to 2019. It is possible that this can't be a broad assumption, in that some countries may be more sensitive to oil prices than others. The Iraq data isn't enough to make this conclusion as there are many external factors that could be involved but it is still important to note the familiarity when we see it.

5. Conclusion

After thoroughly analyzing the global prices of food from multiple datasets, the findings reveal a complex tapestry of factors that influence global food pricing dynamics. Our investigation incorporated diverse datasets. The analysis of these datasets demonstrated significant variations in how different factors interplay to affect food prices across various countries. For instance, the correlation between GDP and food imports varied, with some nations showing positive correlations, where rises in GDP coincided with increased food imports, suggesting that economic growth could lead to greater dependence on imported food. Conversely, other countries displayed negative correlations, indicating that higher GDP might reduce reliance on food imports due to enhanced domestic production or other economic factors. These findings underscore the complexity of global food markets and suggest that multiple factors significantly influence food prices. Oil prices did seem to have an inverse correlation with food imports in general, especially in countries such as the UK, USA and France, along with exceptions such as China. In regards to food price being the benchmark, similar events occur with all variables. GDP seems to have a strong positive correlation to food price in places like Pakistan, while others it is not as consistent. Oil price has some sort of positive correlation with Iraq but is almost non-existent amongst the other countries that were visualized. Food waste once again, the country creates mixed results.

Moreover, the project explored the impact of external commodities like crude oil on food prices, reflecting on how global economic conditions, such as oil price fluctuations, can have cascading effects on food markets. This aspect of the analysis highlights the interconnectedness of global economies. While the trend can appear somewhat random at times due to varying external factors, there is a strong suggestion that increases in oil prices tend to coincide with higher food prices. However, this relationship is not consistent across all nations or time periods, highlighting the complex nature of food price dynamics.

In conclusion, this project shows the intricate relationships among various economic indicators and their impact on food prices. While it provides valuable insights into potential trends and correlations, it also underscores the complexity of attributing changes in food prices to specific economic factors.

6. Reference

CTV News. (2024, April 8). 32 per cent of Canadians blame grocery stores for rising food prices more than any other reason: Nanos. CTV News. https://www.ctvnews.ca/business/32-per-cent-of-canadians-blame-grocery-stores-for-rising-food-prices-more-than-any-other-reason-nanos-1.6834573

Python Data Analysis Library. (2024). pandas vX.X.X [Computer software]. Retrieved from https://pandas.pydata.org/

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Retrieved from https://matplotlib.org/

Oracle. (2024). MySQL Connector/Python [Computer software]. Retrieved from https://dev.mysql.com/downloads/connector/python/

Food Products Imports by country | WITS Data. (n.d.). https://wits.worldbank.org/CountryProfile/en/Country/WLD/Year/LTST/TradeFlow/Import/Partner/by-country/Product/16-24_FoodProd

Crude oil prices.(2022).ourworldindata https://ourworldindata.org/grapher/crude-oil-prices

Global Food Prices Database. (2020, September 20). Kaggle. https://www.kaggle.com/datasets/salehahmedrony/global-food-prices

World GDP by Country: 1960-2022. (2023, August 24). World Bank Open Data https://data.worldbank.org/indicator/NY.GDP.MKTP.CD

"Database by Doug." (2017, Oct 12). SQL Ranking Functions: Part 1 The Over Clause [Video]. YouTube. https://www.youtube.com/watch?v=YdxyTMjpMMs

Food Loss and Waste Database | Food and Agriculture Organization of the United Nations. (n.d.). https://www.fao.org/faostat/en/#data/FBS

US Official exchange rate year 2017 | The World Bank Datac https://data.worldbank.org/indicator/PA.NUS.FCRF?end=2022&start=2022&view=map&year=2017

Summary findings. USDA ERS - Summary Findings. (n.d.). https://www.ers.usda.gov/data-products/food-price-outlook/summary-findings/

Global food prices have been rising during the coronavirus pandemic, hitting food security(2020, Sep 9).CNBC. https://www.cnbc.com/2020/09/09/global-food-prices-have-been-rising-during-pandemic-hitting-food-security.html

Wheat | World Population Review. (n.d.).
https://worldpopulationreview.com/country-rankings/wheat-production-by-country