REPORT ON ANALYSIS OF US CRUDE OIL IMPORTS

Exploratory Data Analysis (EDA) with Python:

1.1) Load data from the database into Pandas DataFrames:

We can load a CSV file containing US crude oil import data into Pandas DataFrame by using the following code, as shown in the image.

```
df=pd.read_csv("US crude oil Import.csv")
df
```

	year	month	originName	originTypeName	destinationName	destinationTypeName	gradeName	quantity
0	2009	1	Belize	Country	EXXONMOBIL REFINING & SPLY CO / BEAUMONT / TX	Refinery	Light Sour	61
1	2009	1	Belize	Country	FLINT HILLS RESOURCES LP / WEST / TX	Refinery	Light Sour	62
2	2009	1	Algeria	Country	SHELL OIL PRODUCTS US / ST ROSE / LA	Refinery	Light Sweet	10
3	2009	1	Algeria	Country	OIL TANKING PL INC / HOUSTON (GULF) / TX	Refinery	Light Sweet	381
4	2009	1	Algeria	Country	UNKNOWN PROCESSOR-TX / UNKNOWN PROCESSOR-TX / TX	Refinery	Light Sweet	851
...	...	...	...	...	...	...	...	...
483048	2024	1	World	World	United States	United States	Heavy Sour	120942
483049	2024	1	World	World	United States	United States	Heavy Sweet	8859
483050	2024	1	World	World	United States	United States	Light Sour	7811
483051	2024	1	World	World	United States	United States	Light Sweet	12553
483052	2024	1	World	World	United States	United States	Medium	55237

483053 rows × 8 columns

1.2) Perform extensive EDA:

Libraries such as Pandas, NumPy, Matplotlib, and Seaborn were imported for both numerical analysis and data visualization purposes.
To confirm the dimensions of the dataset, use **df.shape**, which reveals that there are a total of 483,053 rows and 8 columns.
To examine the first five rows used **df.head()** and to examine the last five rows used **df.tail()**.
Used **df.columns** to access the column names.
Used **info()** in E.D.A to completely gain the insights including datatypes,non null counts,and memory usage.

1.3) Statistical summaries:

Used **df.describe()** for statistical analysis to retrieve key summary statistics including count, mean, maximum, minimum, standard deviation, variance, and interquartile range (IQR) which is shown below in the image.

```
df.describe() #gives the statistical summary
```

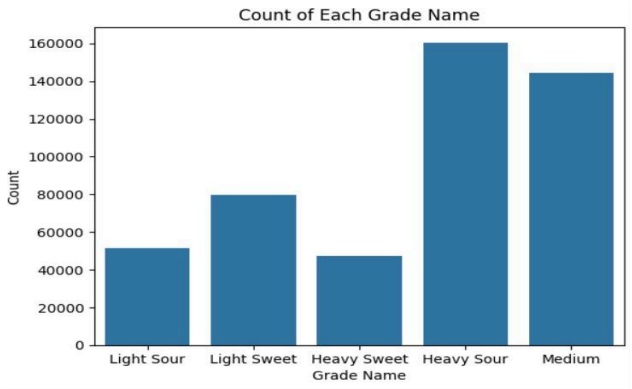
	quantity
count	483018.000000
mean	2425.353745
std	6367.915241
min	1.000000
25%	359.000000
50%	804.000000
75%	2008.000000
max	141016.000000

1.4) Missing value analysis:

Utilized **df.isna().sum()** to determine the count of missing values in the DataFrame. In this case, the count returned zero, indicating no missing values were found within the dataset.
Utilized **df.duplicated().sum()** to determine the count of duplicate values in the DataFrame. In this case, the count returned zero, indicating no duplicate values were found within the dataset.

1.5) Data Cleaning and Preparation:

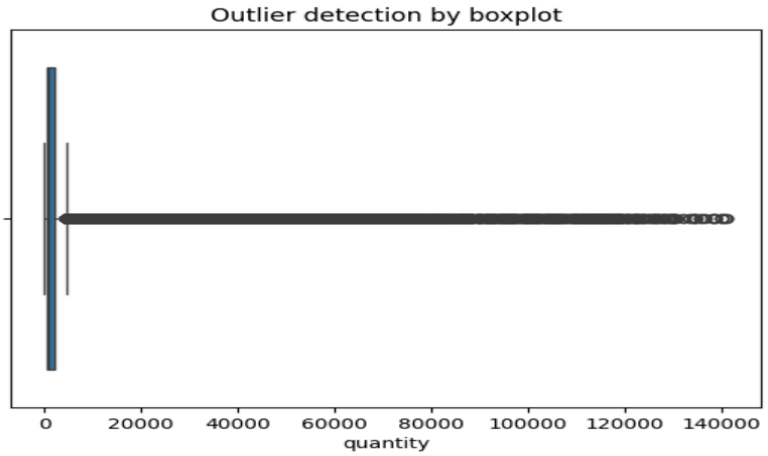
Performed typecasting to change the data types of the **‘year’** and **‘month’** columns from integer to object.
Removed **‘Country not known’** rows from the **'originname'** column where the country name was not known, as these entries did not match valid country names.
Replaced rows containing the name **'The Bahamas**' with **'Bahamas'** and **'South Sudan'** with **'Sudan'** in the 'origin_name' column.
Plotted a count plot to visualize the value counts of the 'Grade name' column.



From the above graph, it is evident that **'Heavy Sour'** has the highest amount of imports.

1.6) Outlier detection and treatment:

Detection of outliers visually using boxplot on the column **‘quantity’**



From the above plot we can see that a huge number of outliers are present above the upper limit.

1.6.1) Getting outliers using Tukey method (IQR method):

Since we don't have any outliers visible at lower bound we have taken the quantile for Q1 as **0.0** and we learned that complete outliers are removed from the dataset when the quantile for Q3 is **0.506**.
So the total number of outliers present in the dataset is **116432** which is a very huge value.

1.6.2) Approaches to deal with outliers:

1) Trimming: Removing the outliers form the dataset
Quantile used for Q3 is **0.506**
Before Trimming: **483018**
After Trimming: **366586**
Percentage of data loss: **24.11 %**
Therefore, there is huge loss for data if outliers are removed.

2) Capping: Sets strict upper and lower limits beyond which values cannot go.
Quantile used for Q3 is **0.55**
Before Capping: **483018**
After Capping: **483018**
Percentage of data loss: **0 %**
Therefore,the outliers are replaced by its maximum value and not removed.

3) Transformation by log: When data is highly skewed or has a long tail, applying a logarithmic transformation can help make the distribution more symmetric.
Here we create a new column known as '**quantity_log**' that has transformed data.
Quantile used for Q3 is **0.55**
Before Transforming: **483018**
After Transforming: **483018**
Percentage of data loss: **0 %**
Therefore,the outliers are replaced by its log and not removed,still few are left.

4) Winsorization: Is a data transformation technique used to mitigate the impact of extreme values (outliers) in a dataset by replacing them with less extreme values within a specified range, typically determined by percentiles.
Upper Percentile used is **88**
Before Transforming: **483018**
After Transforming: **483018**
Percentage of data loss: **0 %**
Therefore,the outliers are replaced with less extreme values and not removed.

Let us see the statistical values of each approach:

Original

count	483018.000000
mean	2425.353745
std	6367.915241
min	1.000000
25%	359.000000
50%	804.000000
75%	2008.000000
max	141016.000000

Trimming

count	366586.000000
mean	683.935311
std	529.568813
min	1.000000
25%	269.000000
50%	524.000000
75%	999.000000
max	2071.000000

Capping

count	483018.000000
mean	1091.037261
std	857.727212
min	1.000000
25%	359.000000
50%	804.000000
75%	2008.000000
max	2393.500000

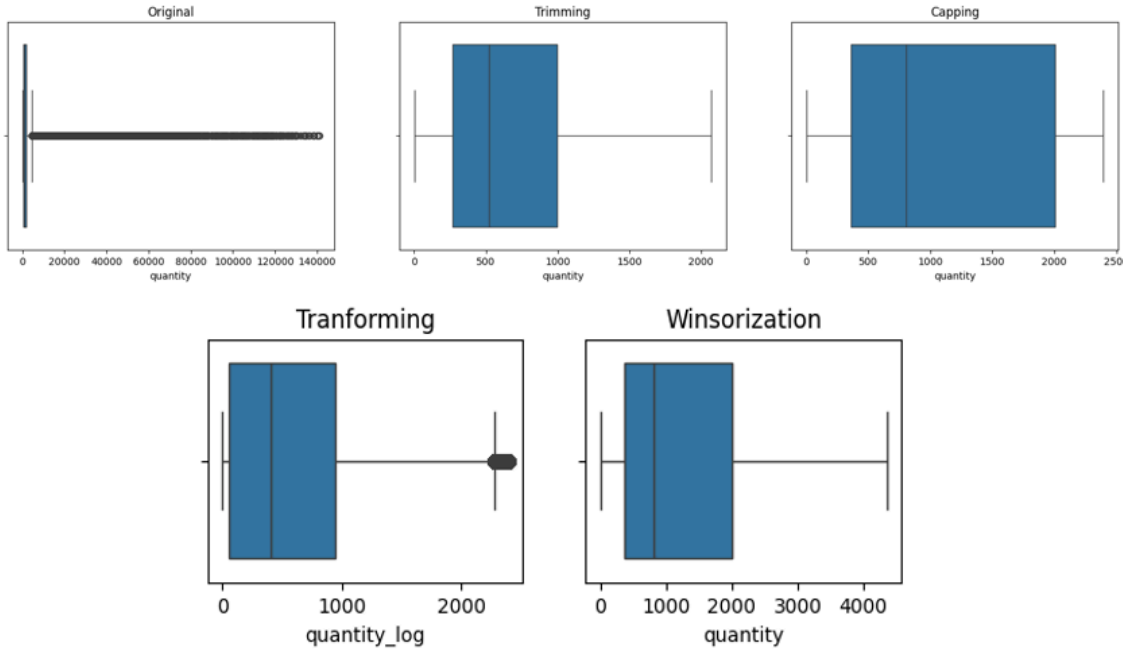
Transform by log

count	483018.000000
mean	582.387845
std	604.165513
min	1.000000
25%	54.000000
50%	413.000000
75%	941.000000
max	2393.000000

Winsorization

count	483018.000000
mean	1401.104334
std	1412.704087
min	1.000000
25%	359.000000
50%	804.000000
75%	2008.000000
max	4349.000000

Let us see Boxplot for each approach:

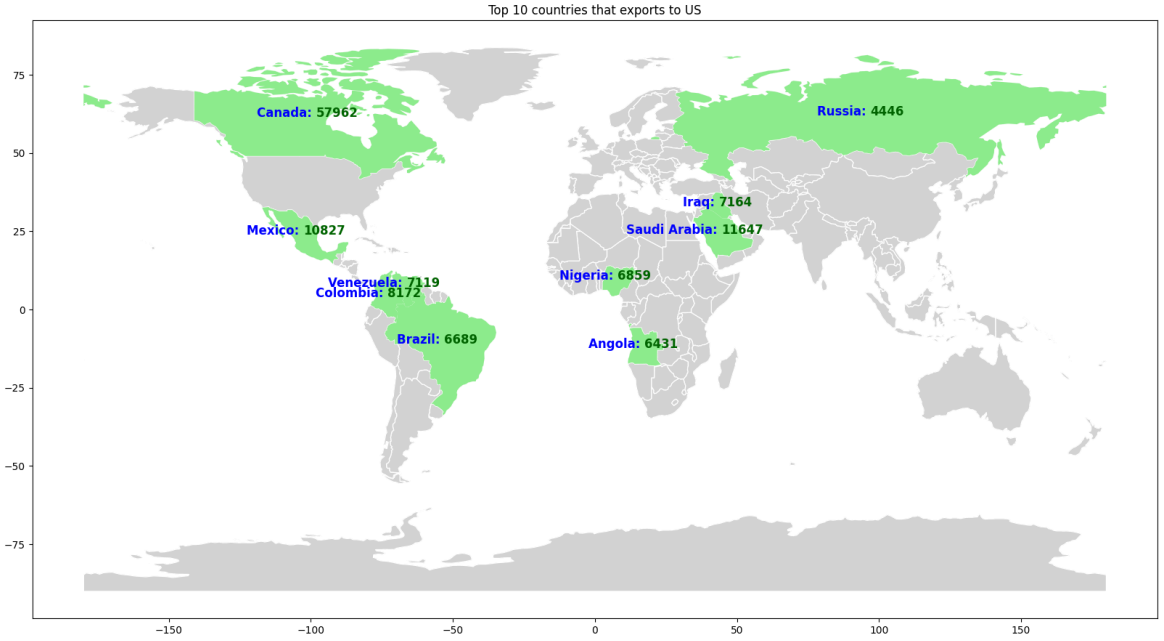


So by looking at the statistical values and the boxplot we can conclude that the winsorization method best suits for handling the outliers.

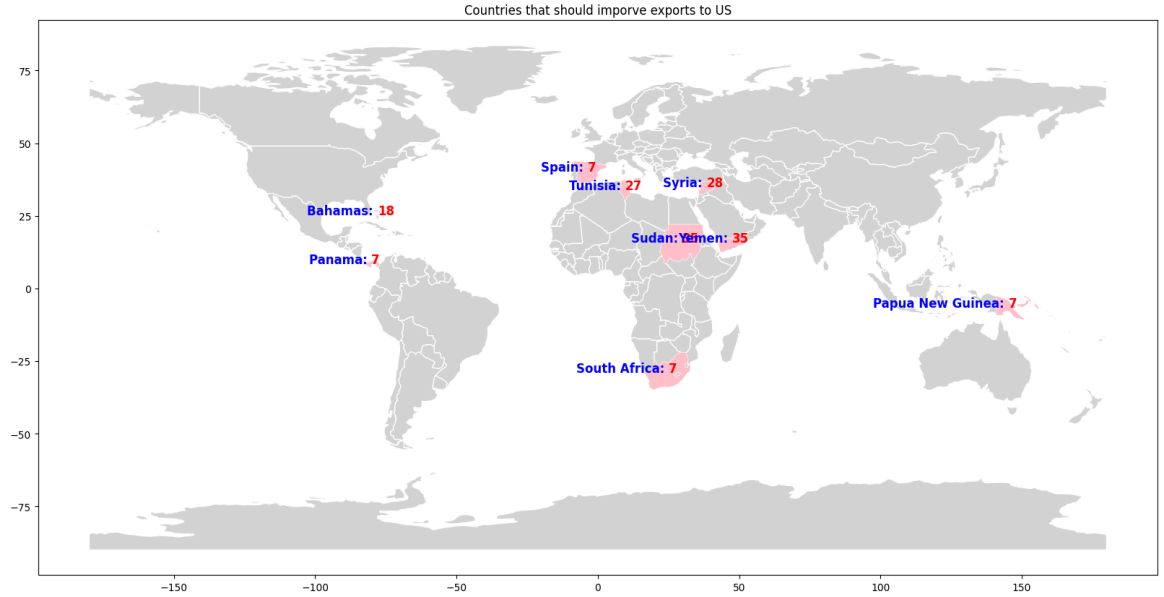
Geospatial Analysis using Python:

We need to install the geopandas library in VScode **pip install geopandas.**
We load the world shapefile using GeoPandas. The shapefile is sourced from the Natural Earth dataset, which provides public domain map datasets available at various resolutions.
gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))

1) Let us see the top 10 countries that have exported crude oil highest number of times to US:

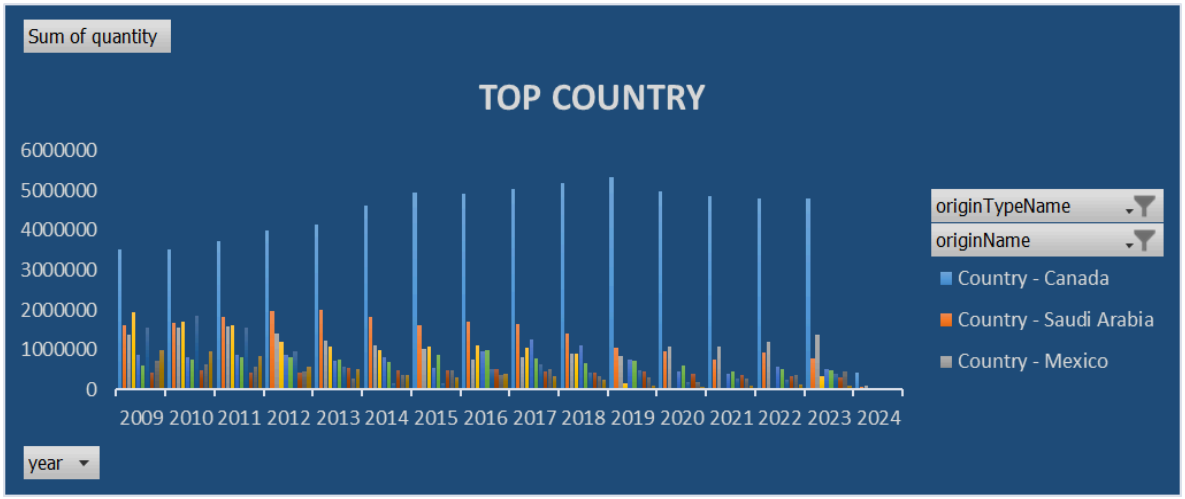


2) Now, the countries exported least number of times to US and which should increase its export are:



Data Analysis with Excel:

1. How has the quantity of crude oil imports varied over the years and which countries are the top origins of crude oil imports to the U.S?



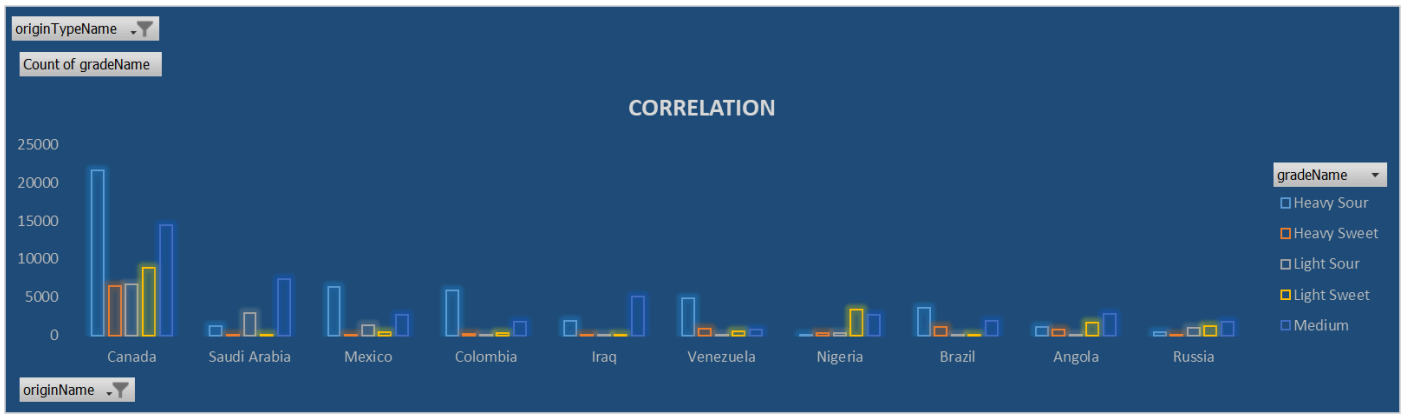
The quantity of crude oil imports increased until **2019**, followed by a slight decrease in imports to the U.S. **Canada** remains the top origin for crude oil imports, with the highest quantity observed in 2019, followed by Saudi Arabia and Mexico.

2. Are there any seasonal trends in crude oil imports?

year	2024
Row Labels	
Jan	
Grand Total	

Since there's only data available for January 2024, we're restricting our analysis to the years between 2009 and 2023. This ensures that the plot isn't influenced by additional January data from 2024, which could distort the results.The above plot includes only data from 2009-2023.
We also get to know from the plot that **July has highest** and **February has least** sum of quantity.

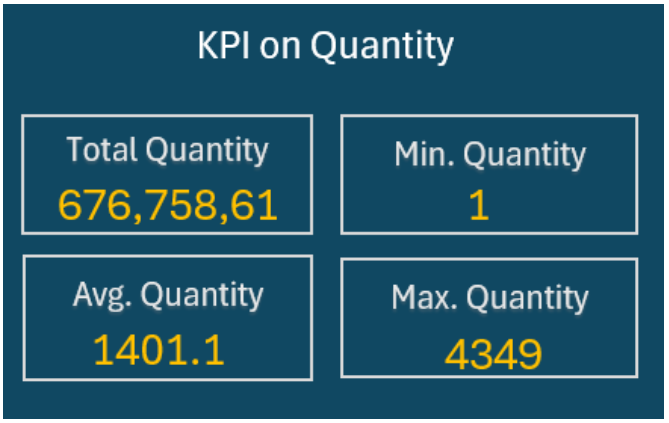
3. Is there a correlation between the origin of crude oil and the grade imported?



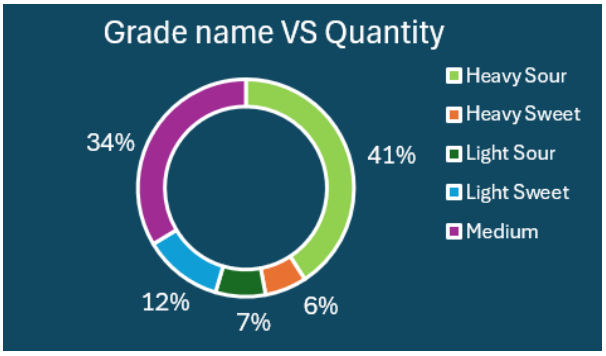
There is a correlation between the origin of crude oil and its grade imported. **Canada** has the highest count of imports, particularly in the **Heavy Sour** grade.

Excel Dashboard Development:

1) Created dynamic Excel dashboards and reports to visualize key performance indicators (KPIs).
- Used **Buttons** to navigate to different sheets i.e the main dashboard and the sheet with map.
- Made four **KPIs** using the column **'quantity'** which is total quantity, maximum quantity, minimum quantity and average quantity.

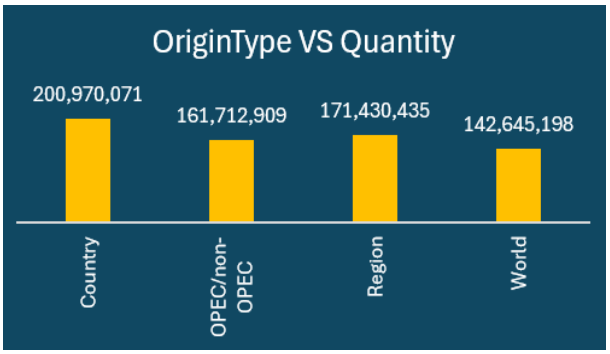


2) Used different charts to represent insights gained.
2.1) Grade Name VS Quantity:



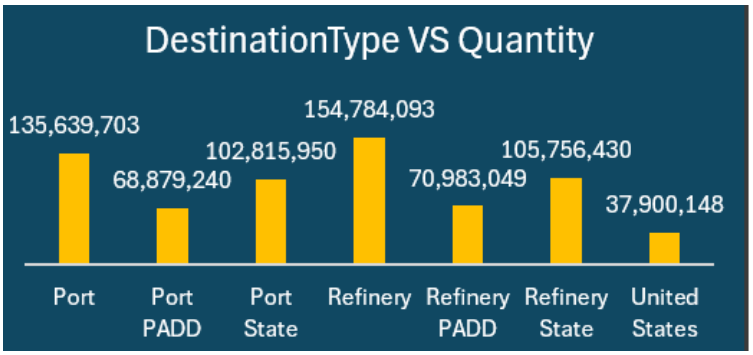
From this donut plot we get to know that **Heavy Sour** has the **highest** quantity imports of **41%** and **Heavy Sweet** has **less** quantity imports of **6%**.

2.2) Origin Type Name VS Quantity:



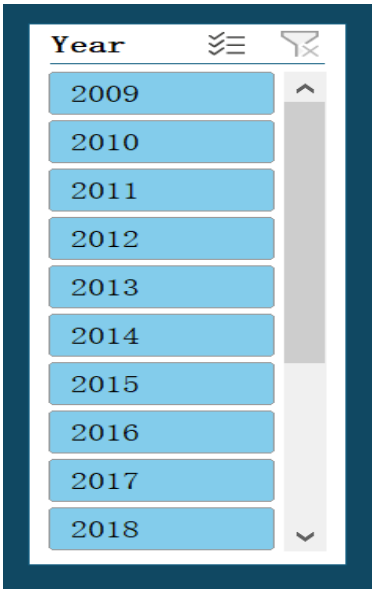
From this bar plot we get to know that from the originType **Country** it's the **highest** quantity imports and from the **World** its **less** quantity imports.

2.3) Destination Type Name VS Quantity:

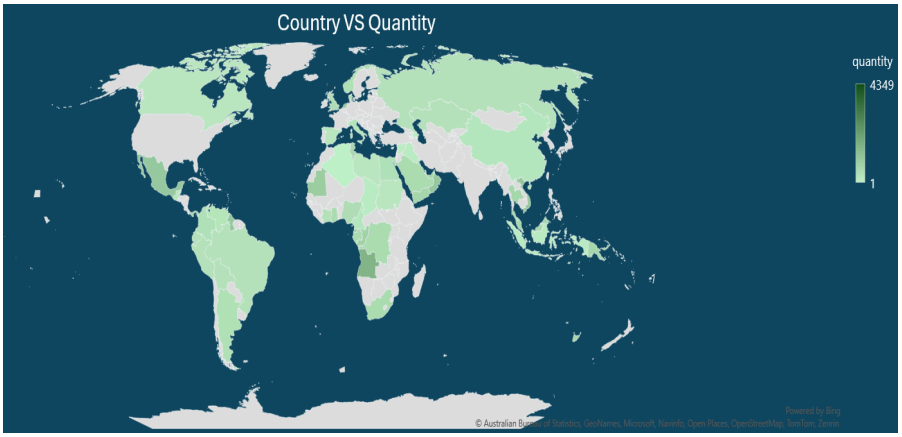


From this bar plot we get to know that the 'destinationTypeName' **Refinery** it's the **highest** quantity imports and from the **United States** its **less** quantity imports.

3) Added an interactive feature by **adding slicers** on the **'year'** where we can choose the year to be applied on all other charts.

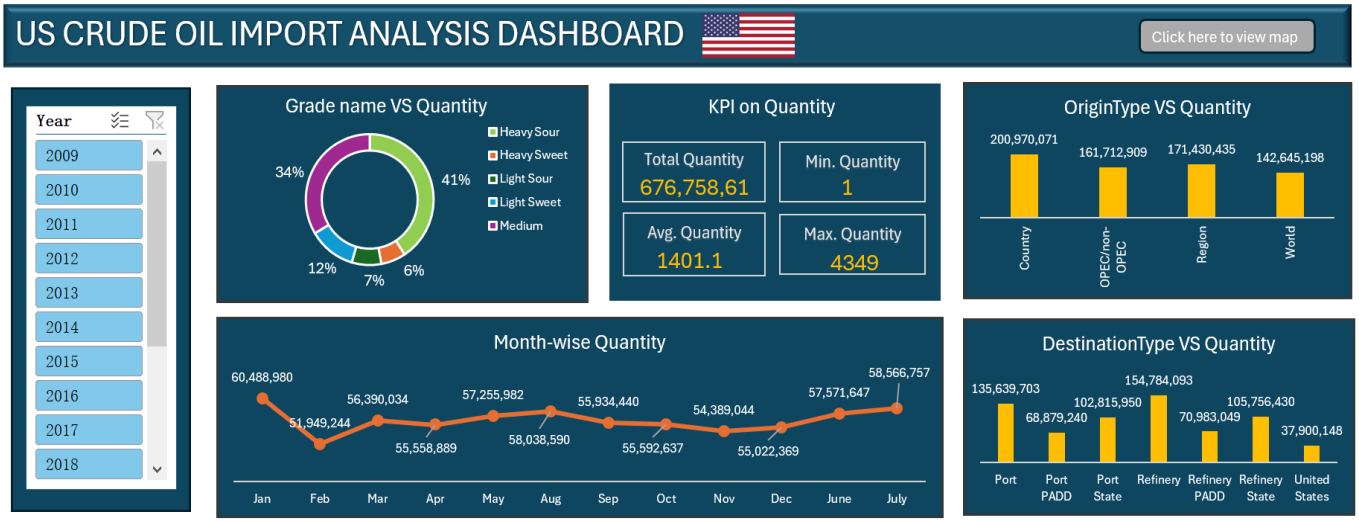


4) Used Excel's charting tools to create a map visualization.



This is the plot on the **'originName'** and **'quantity'** ,here we had to filter out the **'originName'** by selecting **Country** value in **'originTypeName'** to plot the map. Here the dark green shows the area that has exported the highest quantity to the US.

5) Final look of interactive Excel dashboard.

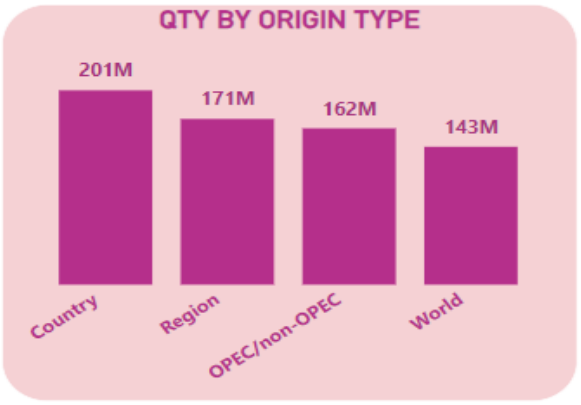


Power BI Dashboard development :

1) Created a dynamic Power BI dashboard and reports to visualize key performance indicators (KPIs).

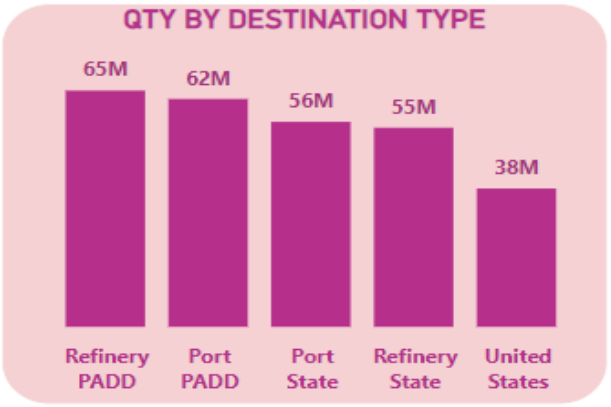
Made five **KPIs** using the column **'quantity'** which is total quantity, maximum quantity, minimum quantity, average quantity and grade of the oil.

2) Used different charts to represent insights gained.
2.1) Quantity By Origin Type:



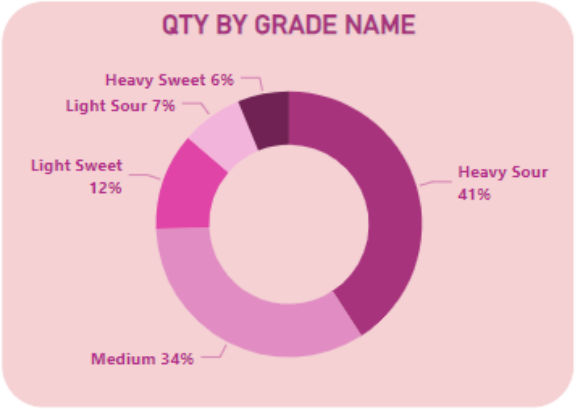
From this bar plot we get to know that 'originTypeName' **Country** has the highest quantity and 'originTypeName' **World** has the lowest quantity of imports.

2.2) Quantity by Destination Type:



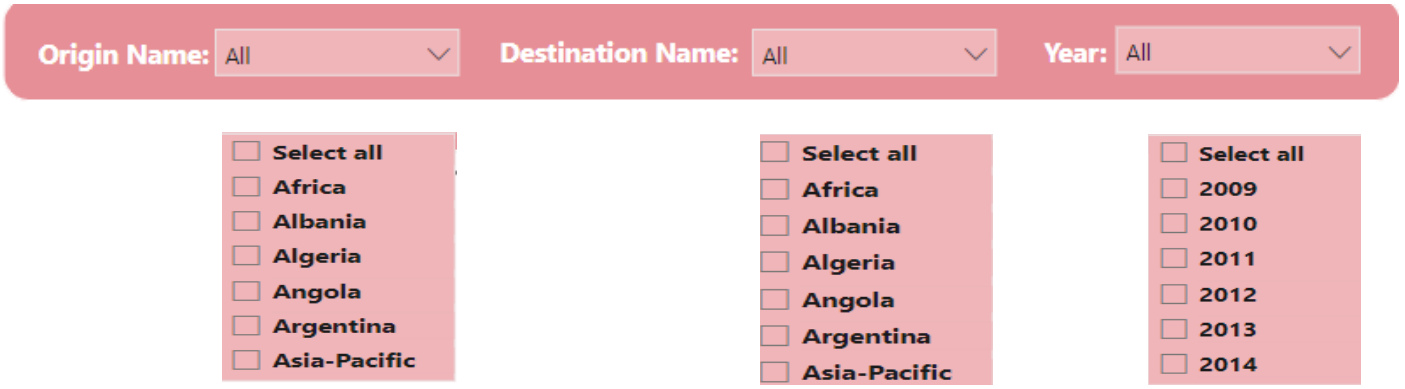
Here ,from the Bar plot we can get to know that **Refinery PADD** has the highest quantity and the **United States** has the lowest quantity of imports.

2.3) Quantity by Grade Name:

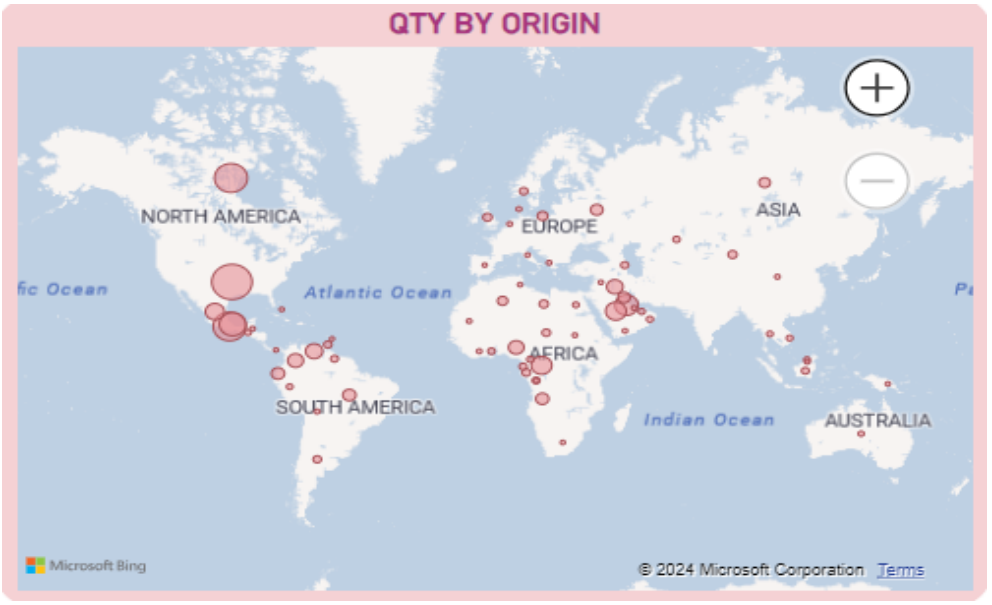


Here ,from this Donut plot we can get to know that **Heavy Sour of 41%** has the highest quantity and **Heavy Sweet of 6%** has the lowest quantity of imports.

3) Added an interactive feature by **adding slicers** on the **'originName'**,**'destinationName'**and **'year'** so where we can apply the graphs to be performed on all the other charts.



4) Used Power BI charting tools to create a map visualization.



In this map visualization plots on the **'originName'** and **'quantity'.** Here the Pink bubble shows the area that has exported the quantity to the US. Based on the selection of origin areas the graph performs according to the origins selected on the map.

5) Final look of interactive Power BI dashboard.