

Name: K.Vinay Balaji Reddy

Rollno : 2211cs010276

Dataset : Crude oil import and petroleum product import/export by Oil companies

description :

Crude oil import involves the purchase and transportation of unrefined crude oil from one country to another. Countries that do not have enough domestic oil production rely on imports to meet their demand for refining into petroleum products. Major importing countries include the United States, China, India, and many European nations. Oil companies play a key role in facilitating these imports, as they manage the logistical, financial, and refining processes involved.

```
In [225... import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("oil.csv")
```

```
In [226... df
```

Out[226...

	Month	Year	PRODUCTS	TRADE	Quantity (000 Metric Tonnes)	Value in Rupees (Crore)	Value in Dollars (Million US dollar)	date_updated
0	April	2024	CRUDE OIL	Import	21439.18	108423.53	12992.53	2025-01-21 04:00:07
1	April	2024	LPG	Import	1570.75	8005.70	959.87	2025-01-21 04:00:07
2	April	2024	MS	Import	29.99	224.85	26.97	2025-01-21 04:00:07
3	April	2024	Naphtha	Import	121.22	735.58	88.19	2025-01-21 04:00:07
4	April	2024	ATF	Import	0.00	0.02	0.00	2025-01-21 04:00:07
...
449	March	2024	Bitumen	Export	2.47	10.02	1.21	2025-01-21 04:00:07
450	March	2024	Petcoke / CBFS	Export	NaN	NaN	NaN	2025-01-21 04:00:07
451	March	2024	Others%	Export	300.89	1625.65	196.37	2025-01-21 04:00:07
452	March	2024	TOTAL PRODUCT EXPORT	Export	5672.60	35623.12	4288.69	2025-01-21 04:00:07
453	March	2024	NET IMPORT	Export	19494.93	80932.31	9723.56	2025-01-21 04:00:07

454 rows × 8 columns

In [227...

df.shape

Out[227...

(454, 8)

In [228...

a = df[df.select_dtypes(exclude=["category"]).columns] = df.select_dtypes(exclud

In [229...

a

Out [229...

	Month	Year	PRODUCTS	TRADE	Quantity (000 Metric Tonnes)	Value in Rupees (Crore)	Value in Dollars (Million US dollar)	date_updated
0	April	2024	CRUDE OIL	Import	21439.18	108423.53	12992.53	2025-01-21 04:00:07
1	April	2024	LPG	Import	1570.75	8005.70	959.87	2025-01-21 04:00:07
2	April	2024	MS	Import	29.99	224.85	26.97	2025-01-21 04:00:07
3	April	2024	Naphtha	Import	121.22	735.58	88.19	2025-01-21 04:00:07
4	April	2024	ATF	Import	0.00	0.02	0.00	2025-01-21 04:00:07
...
449	March	2024	Bitumen	Export	2.47	10.02	1.21	2025-01-21 04:00:07
450	March	2024	Petcoke / CBFS	Export	0.00	0.00	0.00	2025-01-21 04:00:07
451	March	2024	Others%	Export	300.89	1625.65	196.37	2025-01-21 04:00:07
452	March	2024	TOTAL PRODUCT EXPORT	Export	5672.60	35623.12	4288.69	2025-01-21 04:00:07
453	March	2024	NET IMPORT	Export	19494.93	80932.31	9723.56	2025-01-21 04:00:07

454 rows × 8 columns

In [230...

```
a.info
```

```

Out[230... <bound method DataFrame.info of
Quantity (000 Metric Tonnes) \
0 April 2024 CRUDE OIL Import 21439.18
1 April 2024 LPG Import 1570.75
2 April 2024 MS Import 29.99
3 April 2024 Naphtha Import 121.22
4 April 2024 ATF Import 0.00
.. ... ..
449 March 2024 Bitumen Export 2.47
450 March 2024 Petcoke / CBFS Export 0.00
451 March 2024 Others% Export 300.89
452 March 2024 TOTAL PRODUCT EXPORT Export 5672.60
453 March 2024 NET IMPORT Export 19494.93

Value in Rupees (Crore) Value in Dollars (Million US dollar) \
0 108423.53 12992.53
1 8005.70 959.87
2 224.85 26.97
3 735.58 88.19
4 0.02 0.00
.. ... ..
449 10.02 1.21
450 0.00 0.00
451 1625.65 196.37
452 35623.12 4288.69
453 80932.31 9723.56

date_updated
0 2025-01-21 04:00:07
1 2025-01-21 04:00:07
2 2025-01-21 04:00:07
3 2025-01-21 04:00:07
4 2025-01-21 04:00:07
.. ... ..
449 2025-01-21 04:00:07
450 2025-01-21 04:00:07
451 2025-01-21 04:00:07
452 2025-01-21 04:00:07
453 2025-01-21 04:00:07

[454 rows x 8 columns]>

```

In [231...

```
print(df.head())
```

	Month	Year	PRODUCTS	TRADE	Quantity (000 Metric Tonnes)	\
0	April	2024	CRUDE OIL	Import	21439.18	
1	April	2024	LPG	Import	1570.75	
2	April	2024	MS	Import	29.99	
3	April	2024	Naphtha	Import	121.22	
4	April	2024	ATF	Import	0.00	

	Value in Rupees (Crore)	Value in Dollars (Million US dollar)	\
0	108423.53	12992.53	
1	8005.70	959.87	
2	224.85	26.97	
3	735.58	88.19	
4	0.02	0.00	

	date_updated
0	2025-01-21 04:00:07
1	2025-01-21 04:00:07
2	2025-01-21 04:00:07
3	2025-01-21 04:00:07
4	2025-01-21 04:00:07

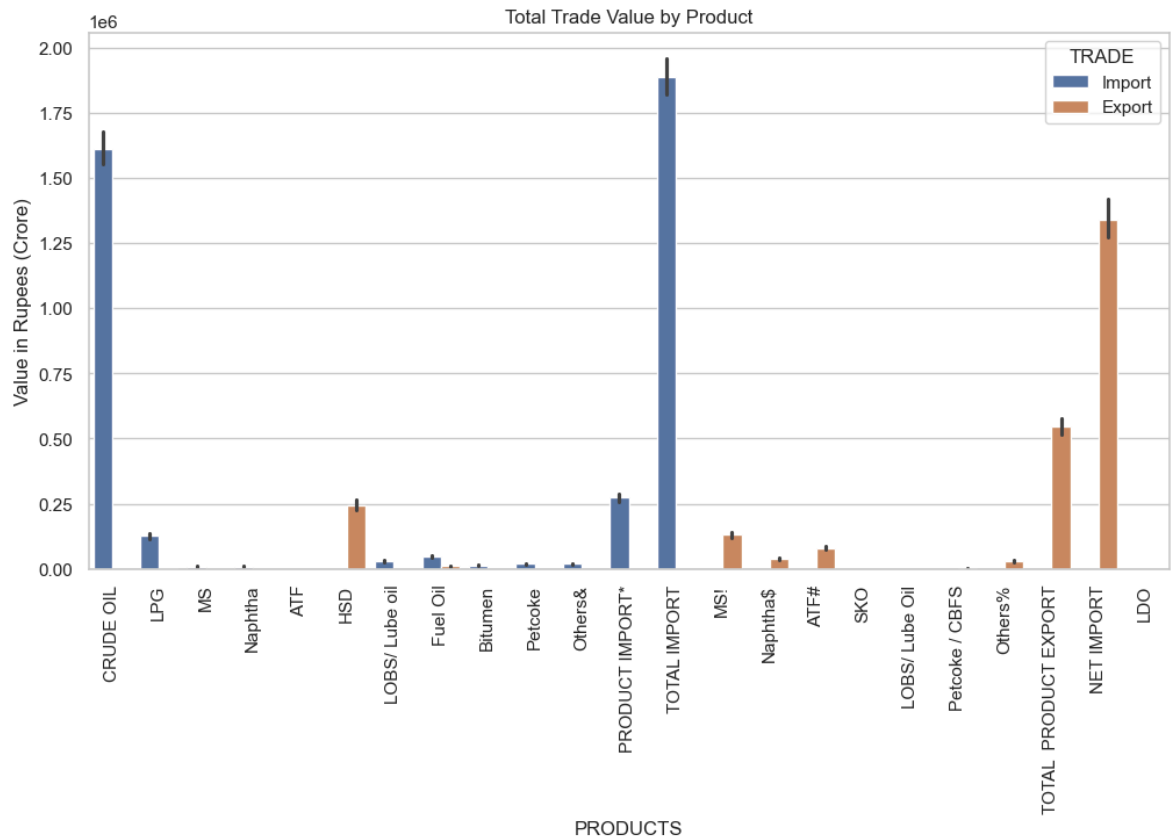
```
In [232... df["Quantity (000 Metric Tonnes)"] = pd.to_numeric(df["Quantity (000 Metric Tonnes)"])
df["Value in Rupees (Crore)"] = pd.to_numeric(df["Value in Rupees (Crore)"], errors="coerce")
df["Value in Dollars (Million US dollar)"] = pd.to_numeric(df["Value in Dollars (Million US dollar)"], errors="coerce")
```

```
In [233... month_order = ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"]
df["Month"] = pd.Categorical(df["Month"], categories=month_order, ordered=True)
df_sorted = df.sort_values(["Year", "Month"])
```

```
In [234... sns.set(style="whitegrid")
```

1.Total Trade Value by Product

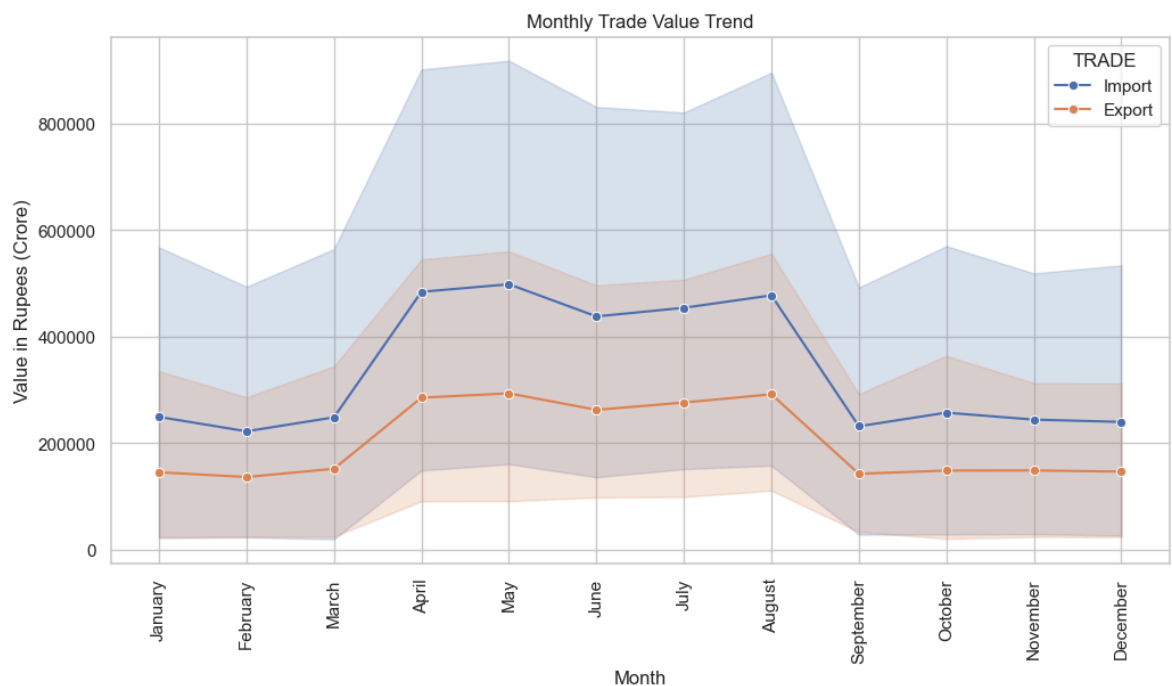
```
In [236... fig, ax = plt.subplots(figsize=(12, 6))
sns.barplot(data=df, x="PRODUCTS", y="Value in Rupees (Crore)", hue="TRADE", estimator=None)
plt.xticks(rotation=90)
plt.title("Total Trade Value by Product")
plt.show()
```



2.Monthly Trade Value Trend

In [238...

```
fig, ax = plt.subplots(figsize=(12, 6))
sns.lineplot(data=df_sorted, x="Month", y="Value in Rupees (Crore)", hue="TRADE")
plt.title("Monthly Trade Value Trend")
plt.xticks(rotation=90)
plt.show()
```

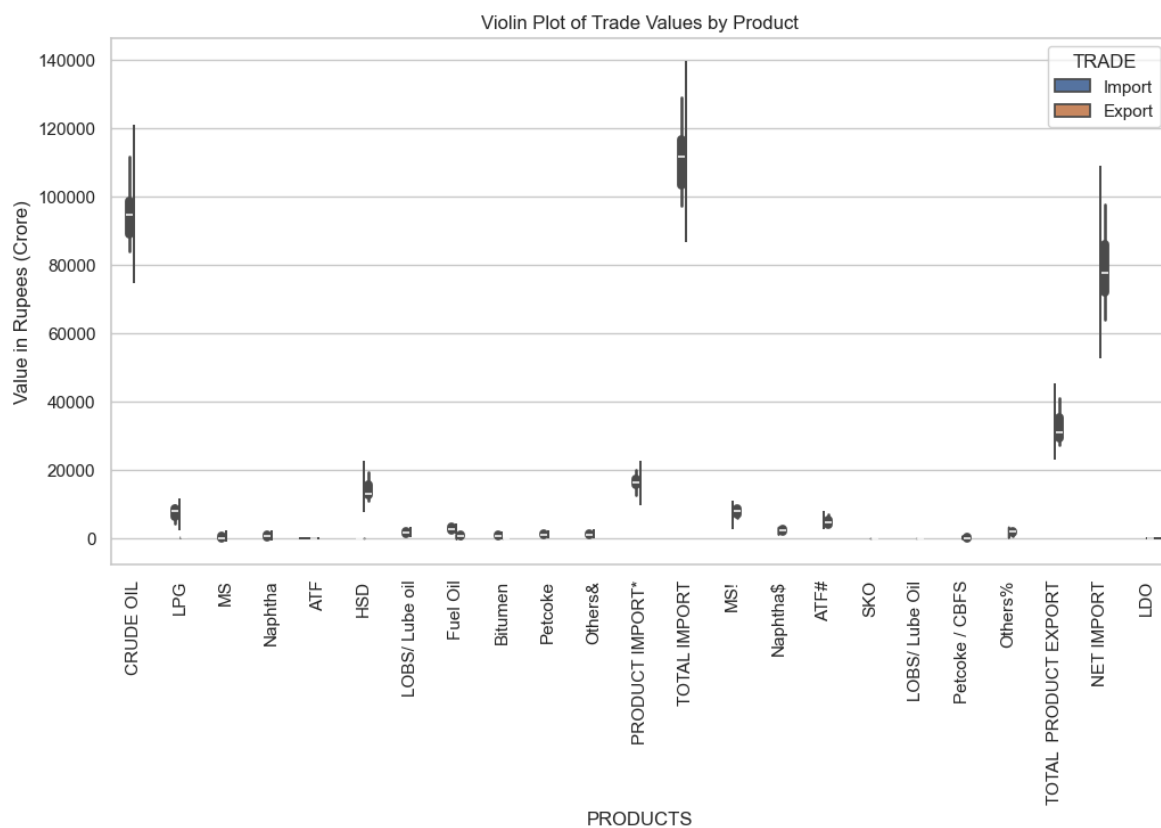


3.Violin Plot of Trade Values by Product

In [240...

```
fig, ax = plt.subplots(figsize=(12, 6))
sns.violinplot(data=df, x="PRODUCTS", y="Value in Rupees (Crore)", hue="TRADE",

plt.xticks(rotation=90)
plt.title("Violin Plot of Trade Values by Product")
plt.show()
```



4.Scatter Plot: Quantity vs Trade Value

In [242...

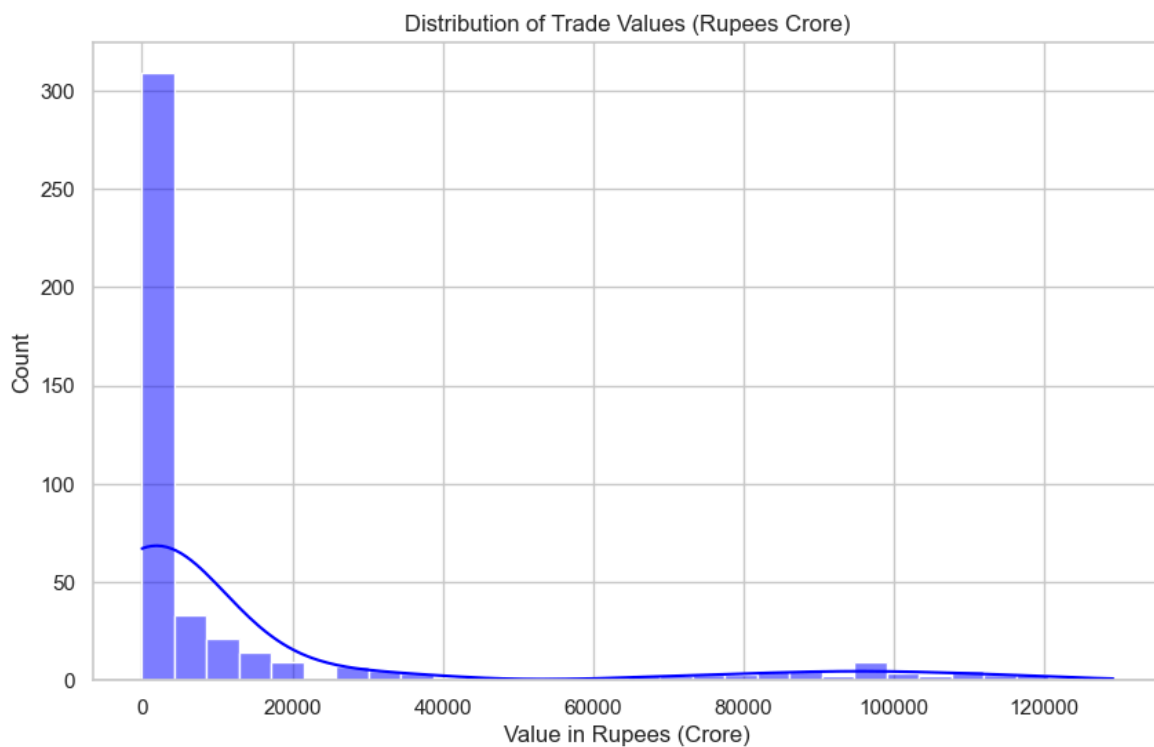
```
fig, ax = plt.subplots(figsize=(10, 6))
sns.scatterplot(data=df, x="Quantity (000 Metric Tonnes)", y="Value in Rupees (C
plt.title("Quantity vs Trade Value (Rupees)")
plt.show()
```



5. Distribution of Trade Values (Rupees Crore)

In [244...

```
fig, ax = plt.subplots(figsize=(10, 6))
sns.histplot(df["Value in Rupees (Crore)"], bins=30, kde=True, ax=ax, color="blue")
plt.title("Distribution of Trade Values (Rupees Crore)")
plt.show()
```



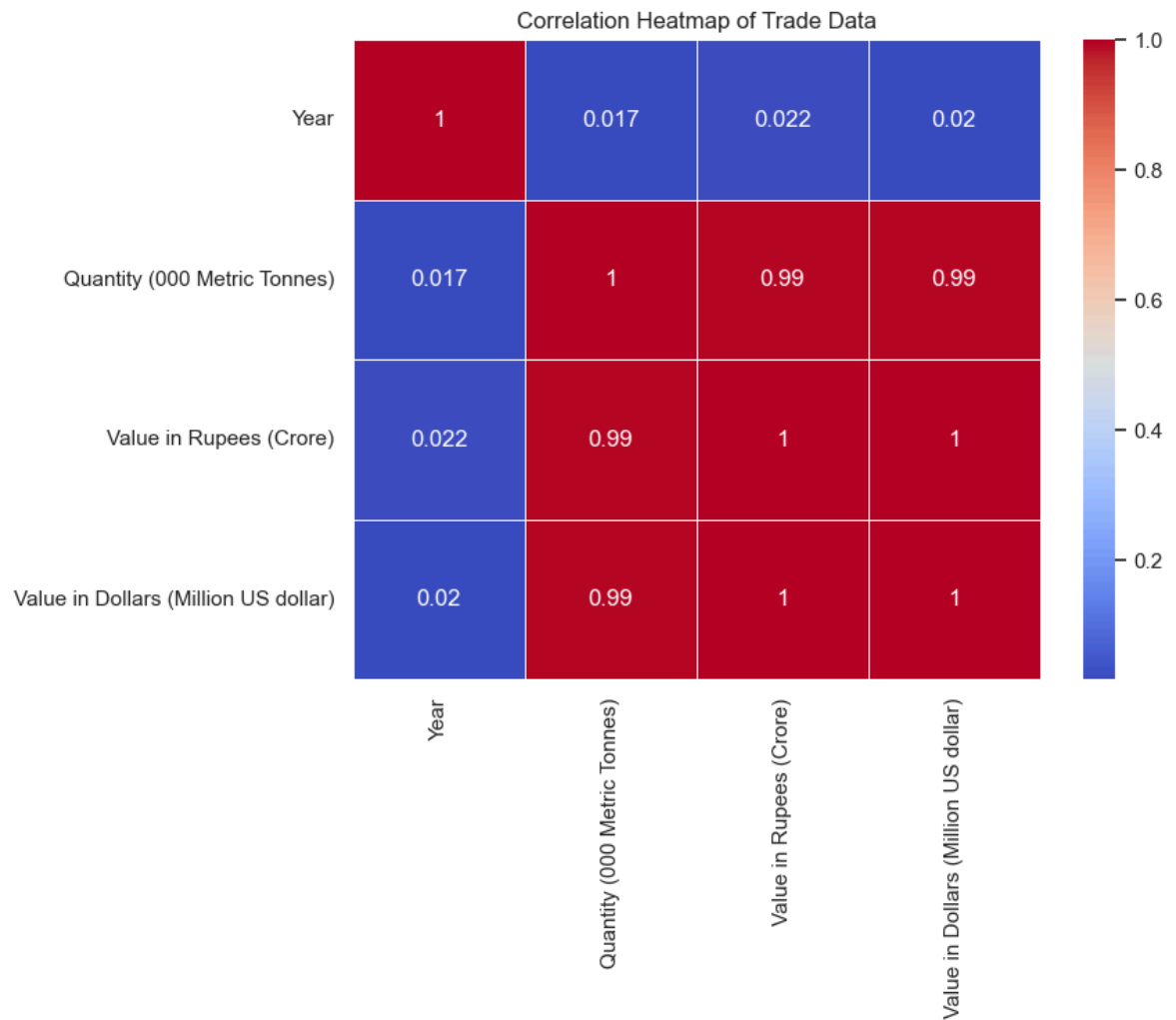
6. Correlation Heatmap of Trade Data

In [246...

```
fig, ax = plt.subplots(figsize=(8, 6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap="coolwarm", linewidths=
```

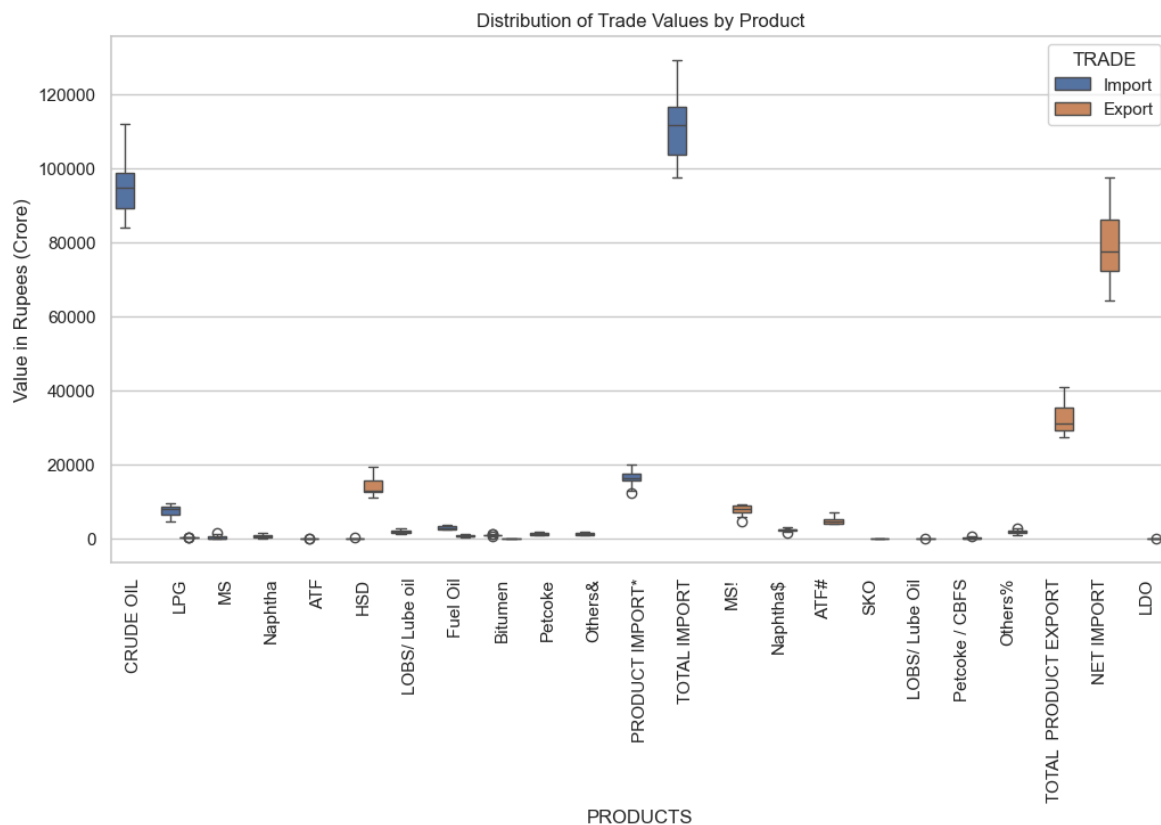


```
plt.title("Correlation Heatmap of Trade Data")
plt.show()
```



7. Distribution of Trade Values by Product

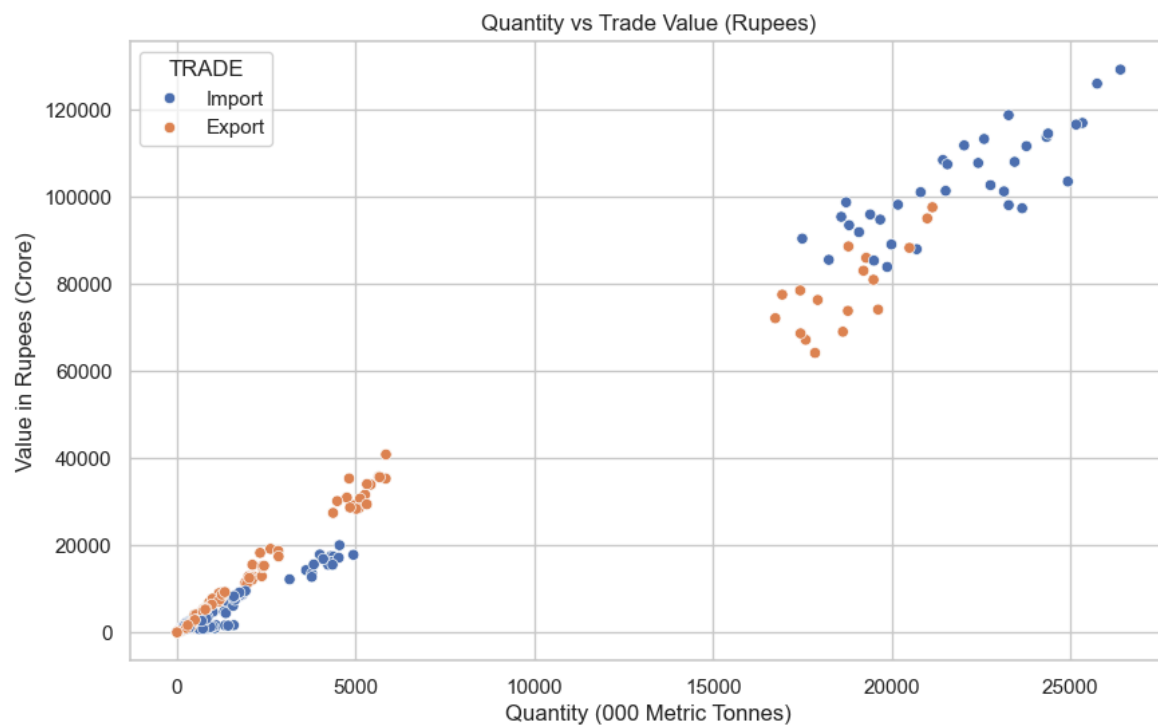
```
In [248... fig, ax = plt.subplots(figsize=(12, 6))
sns.boxplot(data=df, x="PRODUCTS", y="Value in Rupees (Crore)", hue="TRADE", ax=
plt.xticks(rotation=90)
plt.title("Distribution of Trade Values by Product")
plt.show()
```



8.Quantity vs Trade Value (Rupees)

In [250...

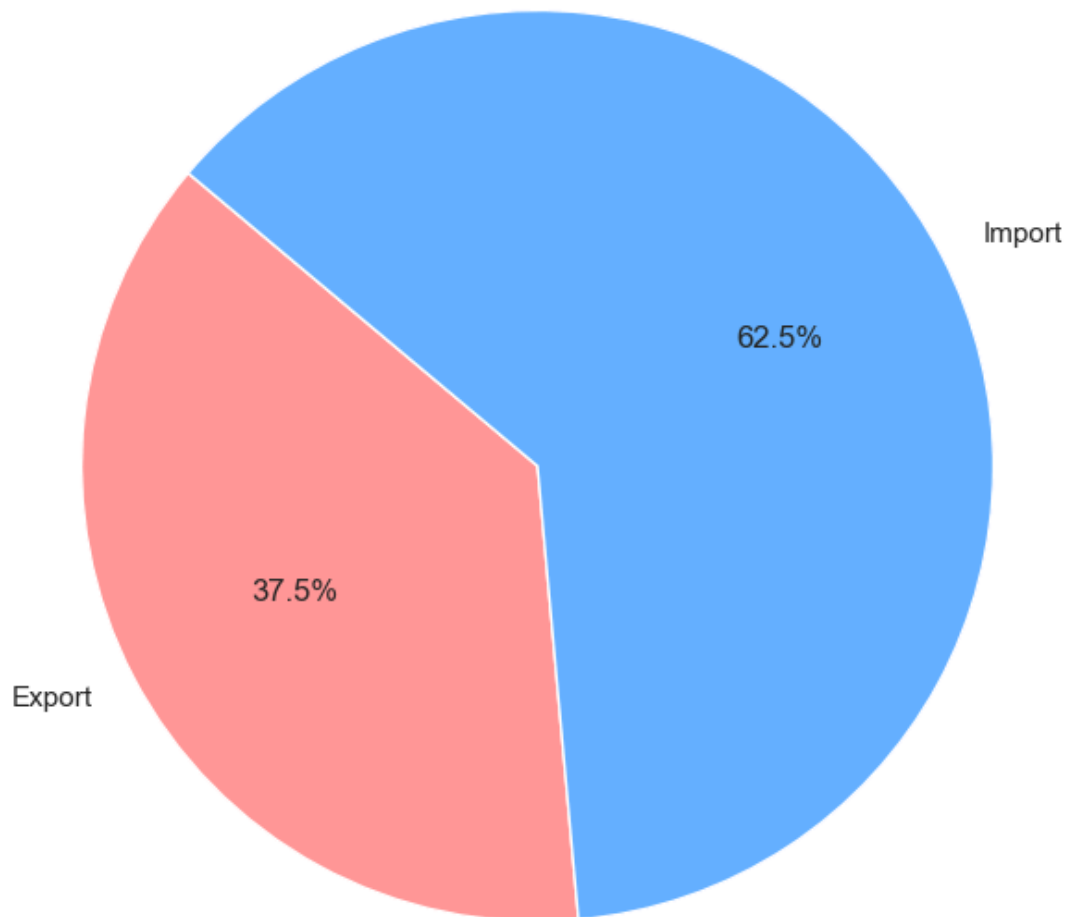
```
fig, ax = plt.subplots(figsize=(10, 6))
sns.scatterplot(data=df, x="Quantity (000 Metric Tonnes)", y="Value in Rupees (C
plt.title("Quantity vs Trade Value (Rupees)")
plt.show()
```



9.Trade Value Distribution (Import vs Export)

```
In [252... trade_distribution = df.groupby("TRADE")["Value in Rupees (Crore)"].sum()  
fig, ax = plt.subplots(figsize=(8, 8))  
ax.pie(trade_distribution, labels=trade_distribution.index, autopct="%1.1f%", c  
plt.title("Trade Value Distribution (Import vs Export)")  
plt.show()
```

Trade Value Distribution (Import vs Export)

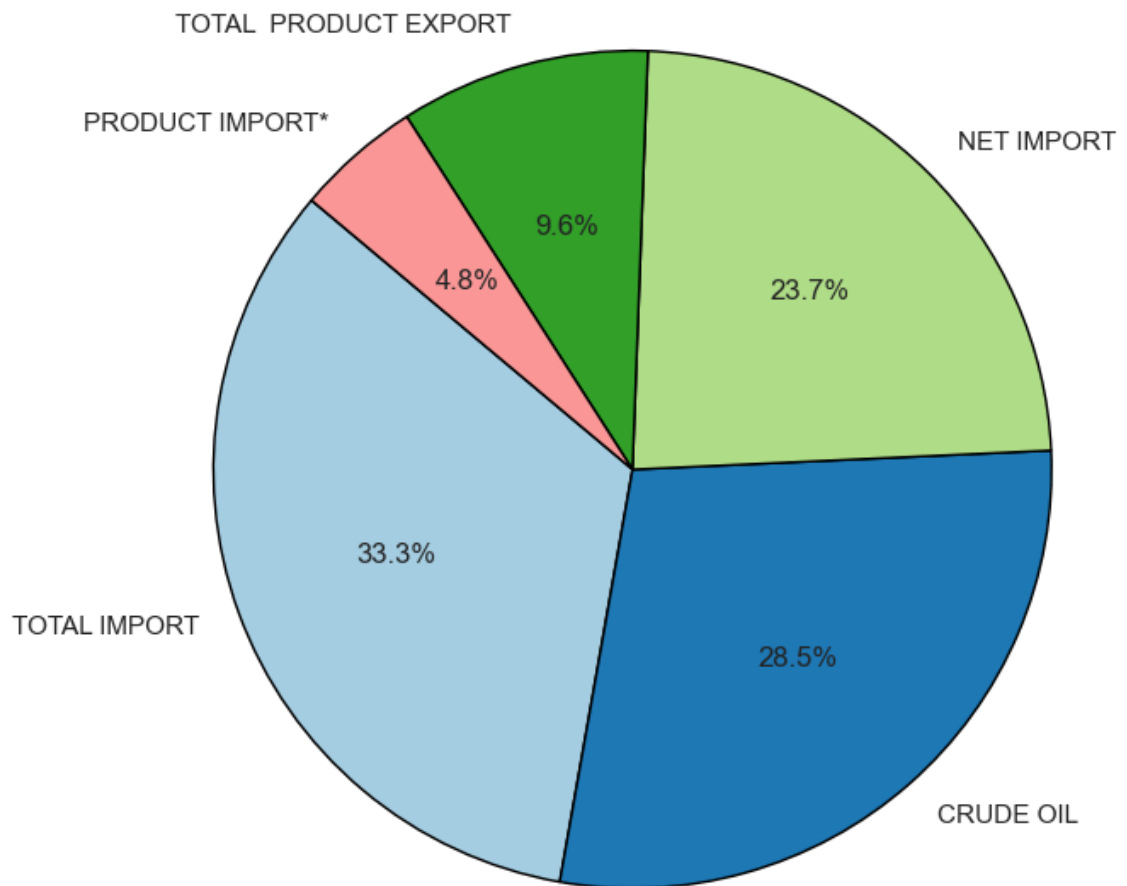


10.Top 5 Products by Trade Value

```
In [254... top_products = df.groupby("PRODUCTS")["Value in Rupees (Crore)"].sum().nlargest(
```

```
In [255... fig, ax = plt.subplots(figsize=(8, 8))  
ax.pie(top_products, labels=top_products.index, autopct="%1.1f%", colors=plt.cm  
plt.title("Top 5 Products by Trade Value")  
plt.show()
```

Top 5 Products by Trade Value



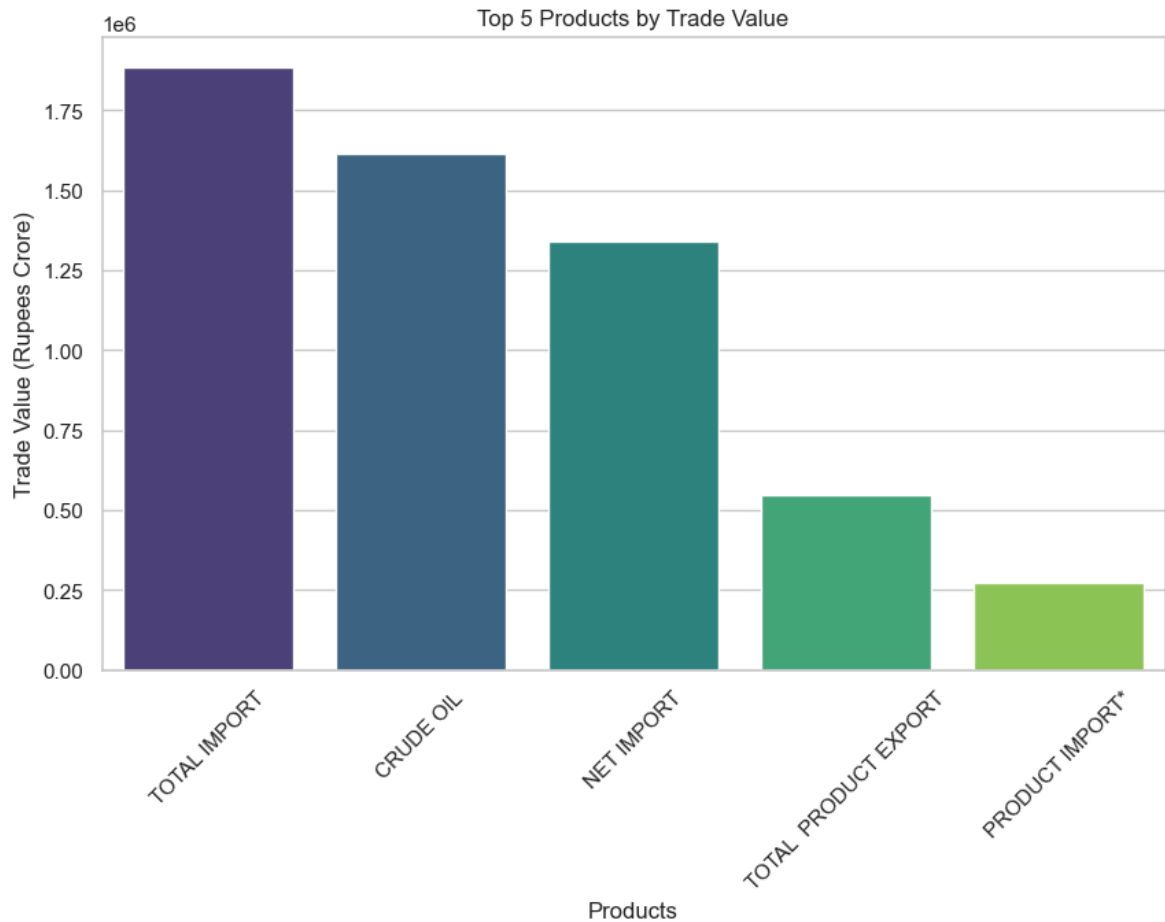
11.Top 5 Products by Trade Value

```
In [257... fig, ax = plt.subplots(figsize=(10, 6))
sns.barplot(x=top_products.index, y=top_products.values, palette="viridis", ax=ax)
plt.ylabel("Trade Value (Rupees Crore)")
plt.xlabel("Products")
plt.title("Top 5 Products by Trade Value")
plt.xticks(rotation=45)
plt.show()
```

C:\Users\vinay\AppData\Local\Temp\ipykernel_18904\3566439296.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

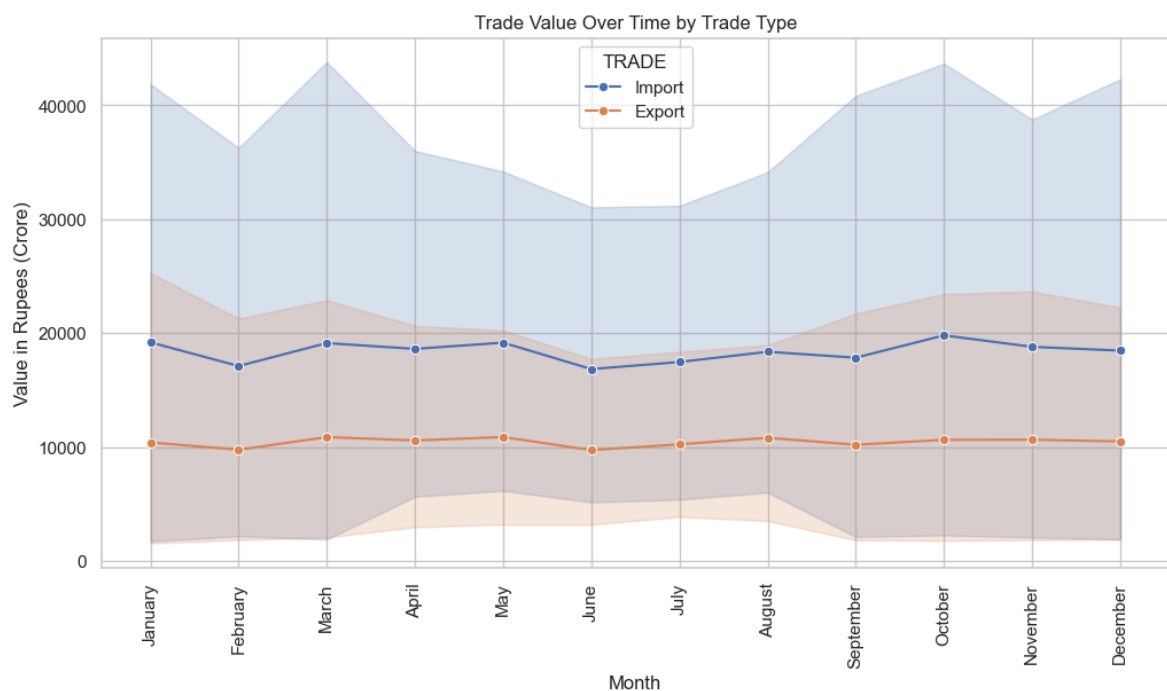
```
sns.barplot(x=top_products.index, y=top_products.values, palette="viridis", ax=ax)
```



12.Trade Value Over Time by Trade Type

In [259...

```
fig, ax = plt.subplots(figsize=(12, 6))
sns.lineplot(data=df_sorted, x="Month", y="Value in Rupees (Crore)", hue="TRADE")
plt.title("Trade Value Over Time by Trade Type")
plt.xticks(rotation=90)
plt.show()
```



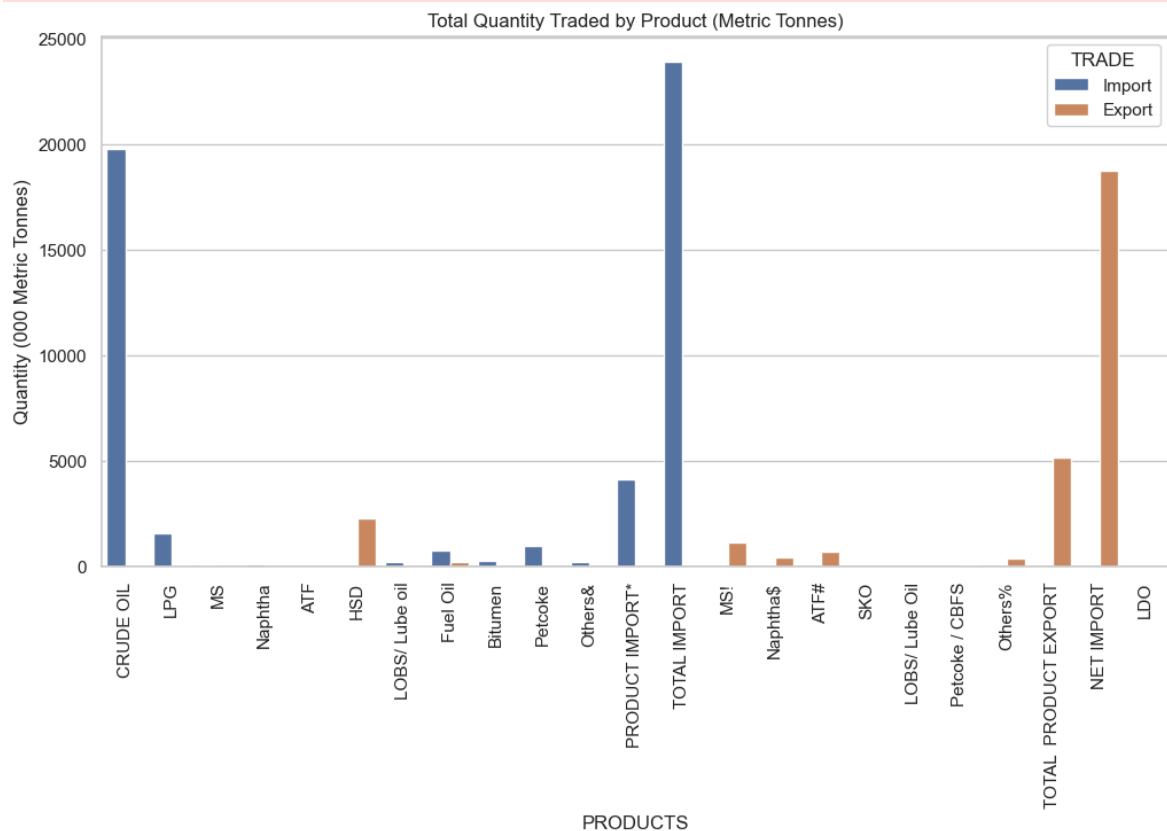
13.Total Quantity Traded by Product (Metric Tonnes)

```
In [261... fig, ax = plt.subplots(figsize=(12, 6))
sns.barplot(data=df, x="PRODUCTS", y="Quantity (000 Metric Tonnes)", hue="TRADE")
plt.xticks(rotation=90)
plt.title("Total Quantity Traded by Product (Metric Tonnes)")
plt.show()
```

C:\Users\vinay\AppData\Local\Temp\ipykernel_18904\1446733767.py:2: FutureWarning:

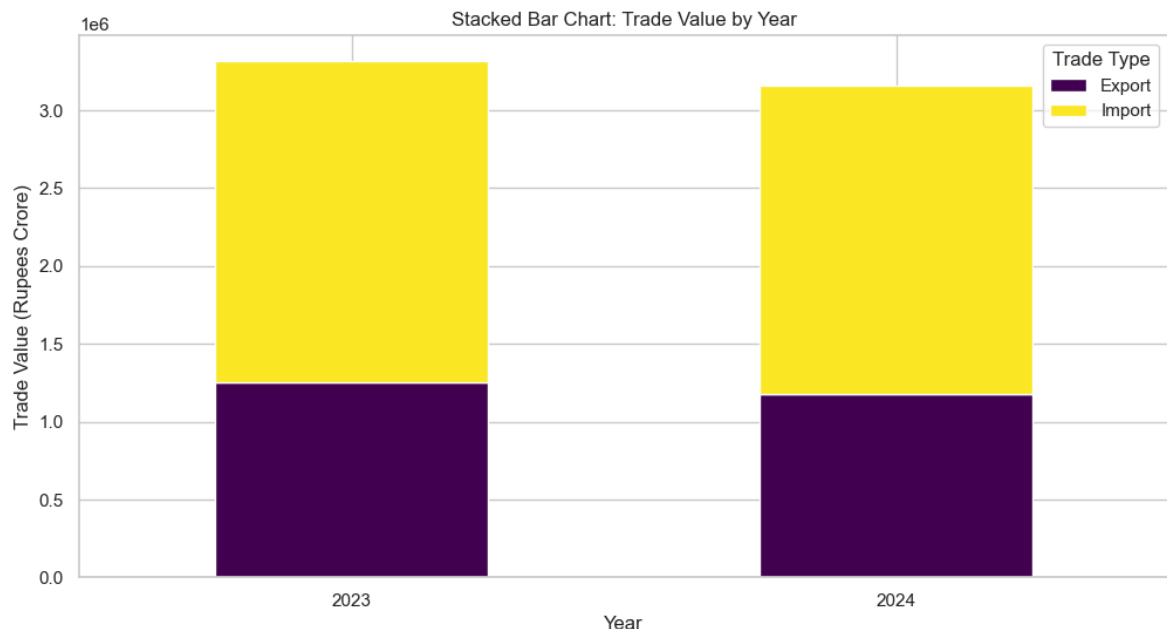
The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

```
sns.barplot(data=df, x="PRODUCTS", y="Quantity (000 Metric Tonnes)", hue="TRADE", ax=ax, ci=None)
```



14.Stacked Bar Chart: Trade Value by Year

```
In [263... fig, ax = plt.subplots(figsize=(12, 6))
df_pivot = df.pivot_table(values="Value in Rupees (Crore)", index="Year", columns="Trade Type")
df_pivot.plot(kind="bar", stacked=True, colormap="viridis", ax=ax)
plt.ylabel("Trade Value (Rupees Crore)")
plt.title("Stacked Bar Chart: Trade Value by Year")
plt.xticks(rotation=0)
plt.legend(title="Trade Type")
plt.show()
```



15. Yearly Trend of Trade Value

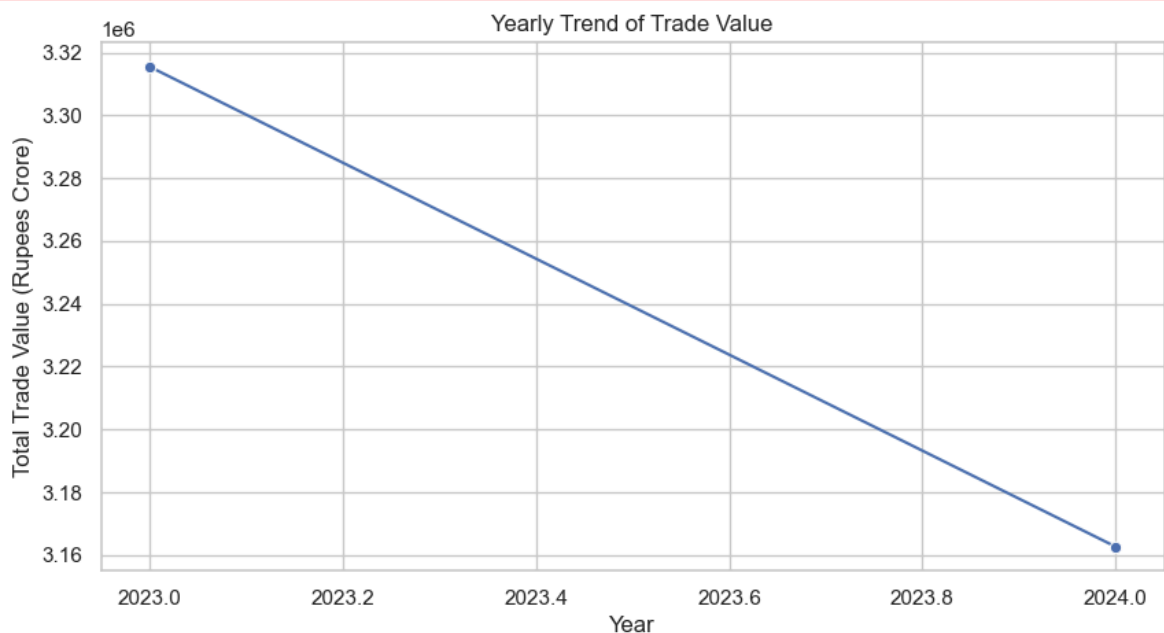
In [265...

```
fig, ax = plt.subplots(figsize=(10, 5))
sns.lineplot(data=df, x="Year", y="Value in Rupees (Crore)", estimator="sum", ci
plt.ylabel("Total Trade Value (Rupees Crore)")
plt.title("Yearly Trend of Trade Value")
plt.grid(True)
plt.show()
```

C:\Users\vinay\AppData\Local\Temp\ipykernel_18904\3037182024.py:2: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

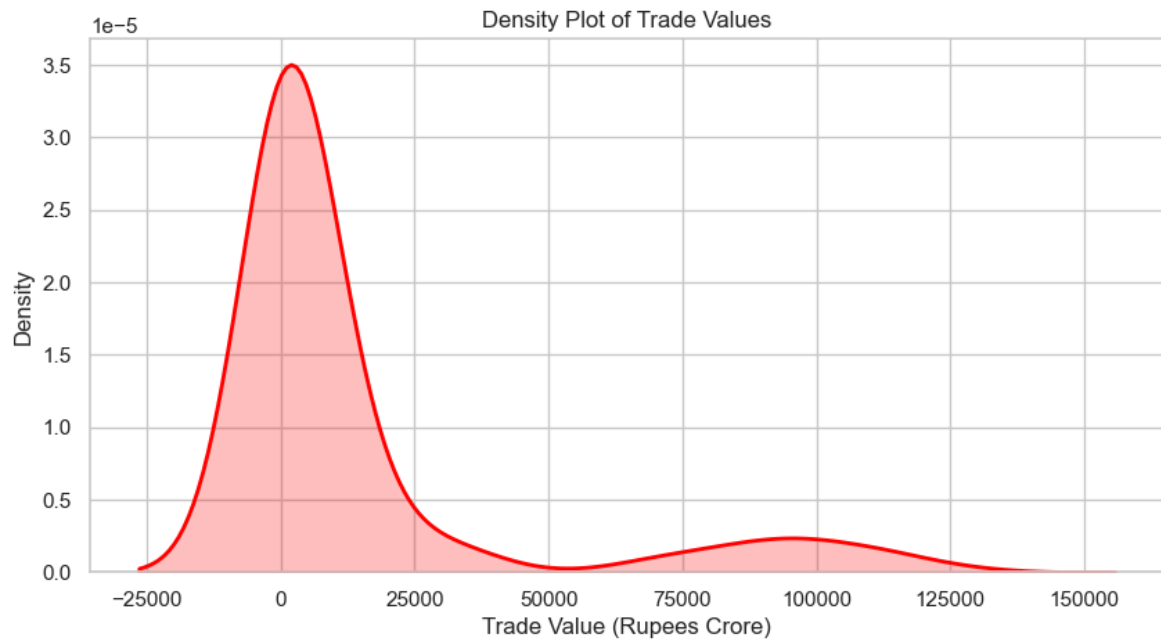
```
sns.lineplot(data=df, x="Year", y="Value in Rupees (Crore)", estimator="sum", c
i=None, marker="o", color="b", ax=ax)
```



16. Density Plot of Trade Values

In [267...

```
fig, ax = plt.subplots(figsize=(10, 5))
sns.kdeplot(df["Value in Rupees (Crore)"], fill=True, color="red", linewidth=2,
plt.title("Density Plot of Trade Values")
plt.xlabel("Trade Value (Rupees Crore)")
plt.show()
```



17.Top 10 Products by Trade Value"

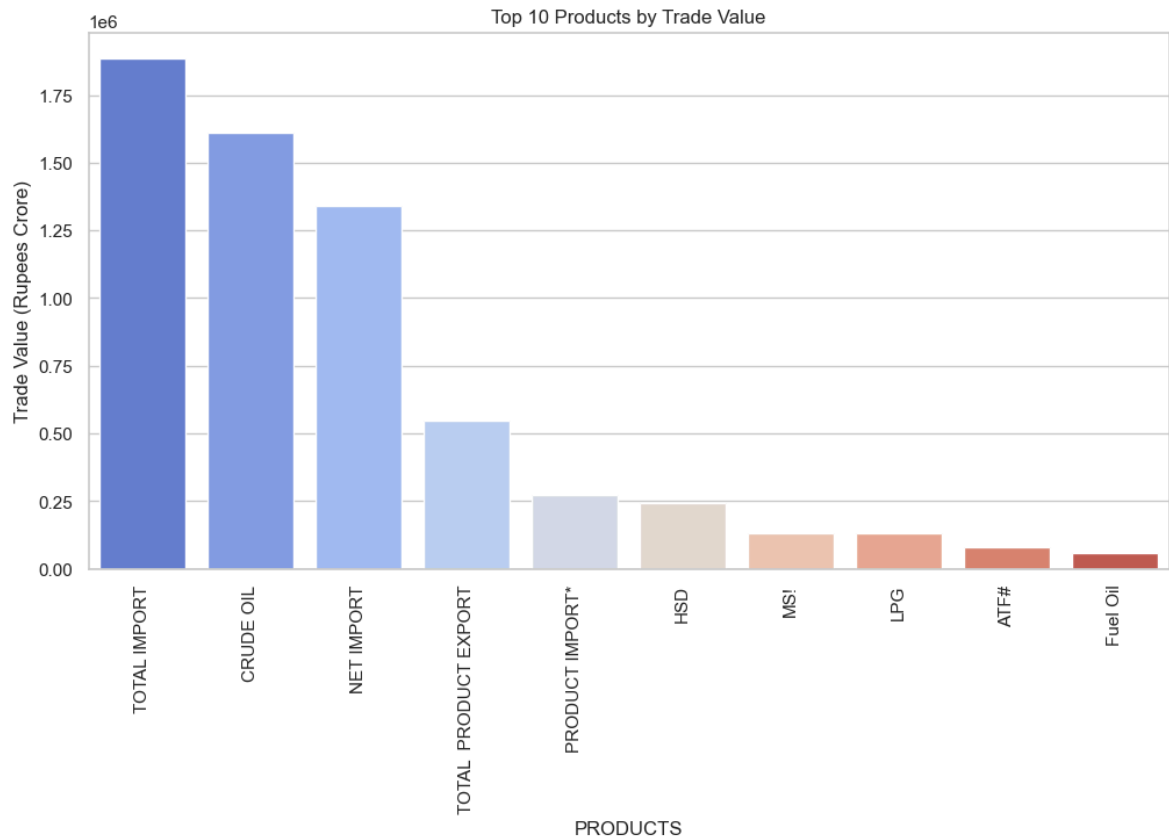
In [269...

```
top_10_products = df.groupby("PRODUCTS")["Value in Rupees (Crore)"].sum().nlarge
fig, ax = plt.subplots(figsize=(12, 6))
sns.barplot(x=top_10_products.index, y=top_10_products.values, palette="coolwarm")
plt.ylabel("Trade Value (Rupees Crore)")
plt.title("Top 10 Products by Trade Value")
plt.xticks(rotation=90)
plt.show()
```

C:\Users\vinay\AppData\Local\Temp\ipykernel_18904\1045635650.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v 0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=top_10_products.index, y=top_10_products.values, palette="coolwarm", ax=ax)
```

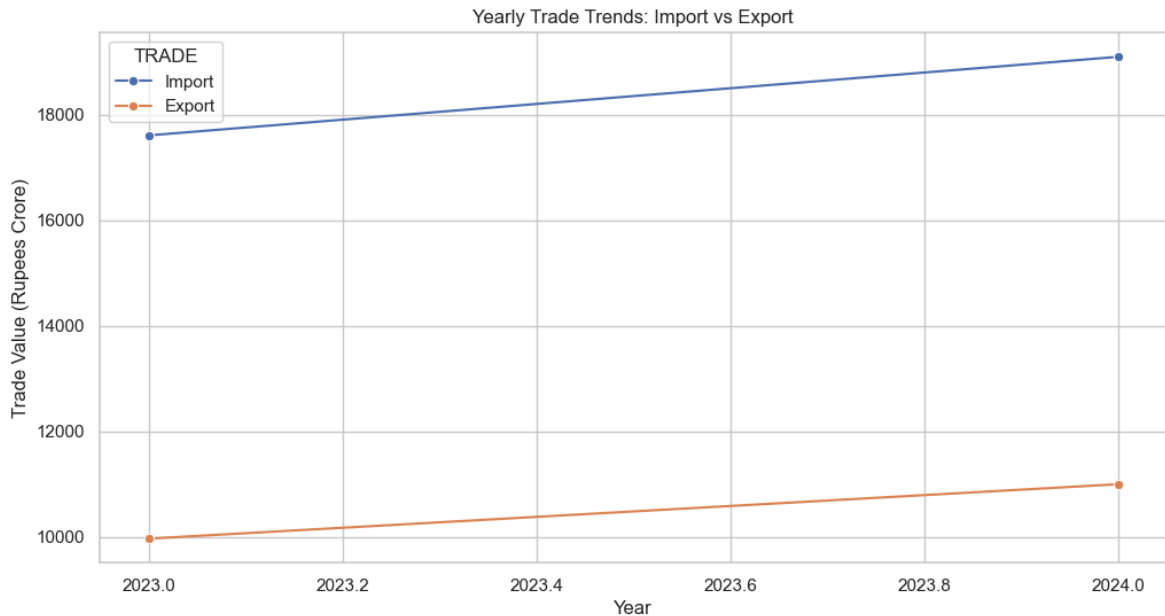
18. Yearly Trade Trends: Import vs Export

```
In [271... fig, ax = plt.subplots(figsize=(12, 6))
sns.lineplot(data=df, x="Year", y="Value in Rupees (Crore)", hue="TRADE", marker=
plt.title("Yearly Trade Trends: Import vs Export")
plt.ylabel("Trade Value (Rupees Crore)")
plt.grid(True)
plt.show()
```

C:\Users\vinay\AppData\Local\Temp\ipykernel_18904\2042977563.py:2: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

```
sns.lineplot(data=df, x="Year", y="Value in Rupees (Crore)", hue="TRADE", marker="o", ci=None, ax=ax)
```



19. Monthly Trade Value Distribution

In [273...

```
fig, ax = plt.subplots(figsize=(12, 6))
sns.barplot(data=df, x="Month", y="Value in Rupees (Crore)", hue="TRADE", ci=None)
plt.title("Monthly Trade Value Distribution")
plt.xticks(rotation=90)
plt.ylabel("Trade Value (Rupees Crore)")
plt.show()
```

C:\Users\vinay\AppData\Local\Temp\ipykernel_18904\1700642794.py:2: FutureWarning:

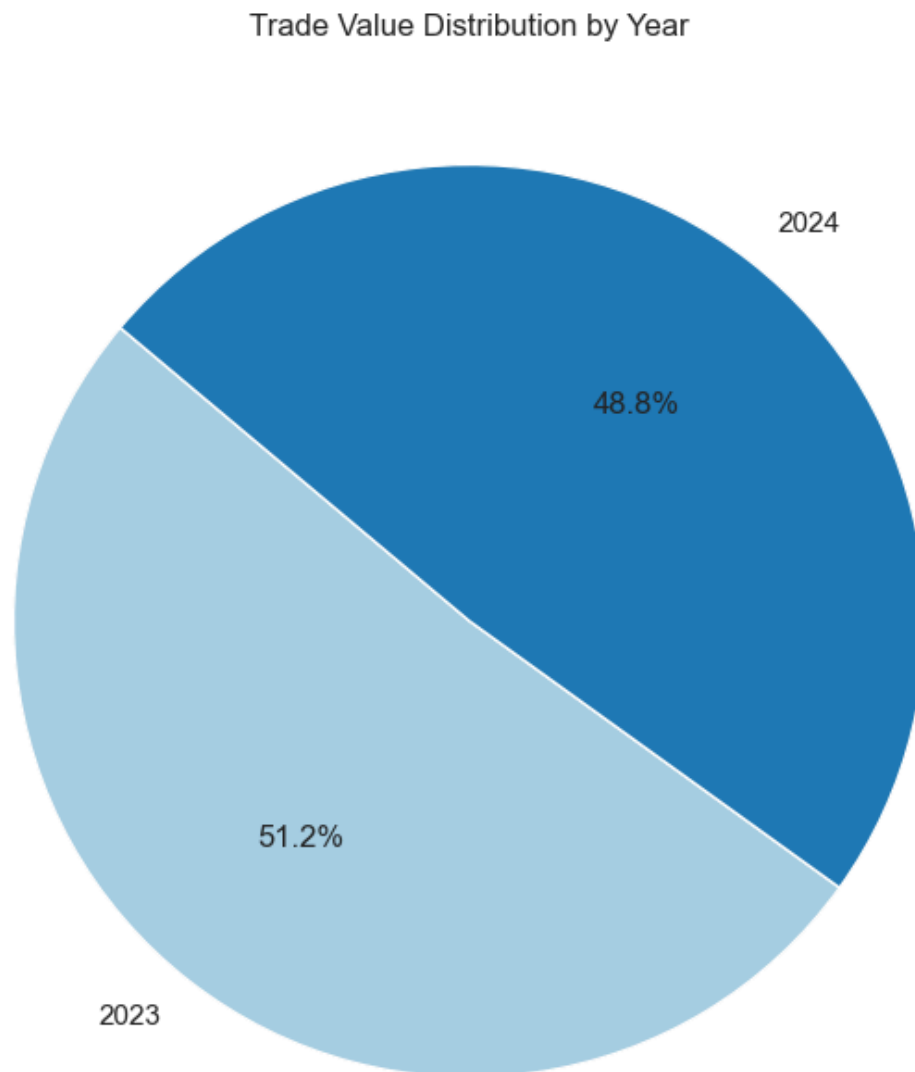
The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

```
sns.barplot(data=df, x="Month", y="Value in Rupees (Crore)", hue="TRADE", ci=None, ax=ax, palette="coolwarm")
```



20.Trade Value Distribution by Year

```
In [275... yearly_trade = df.groupby("Year")["Value in Rupees (Crore)"].sum()
fig, ax = plt.subplots(figsize=(8, 8))
ax.pie(yearly_trade, labels=yearly_trade.index, autopct="%1.1f%%", colors=plt.cm
plt.title("Trade Value Distribution by Year")
plt.show()
```

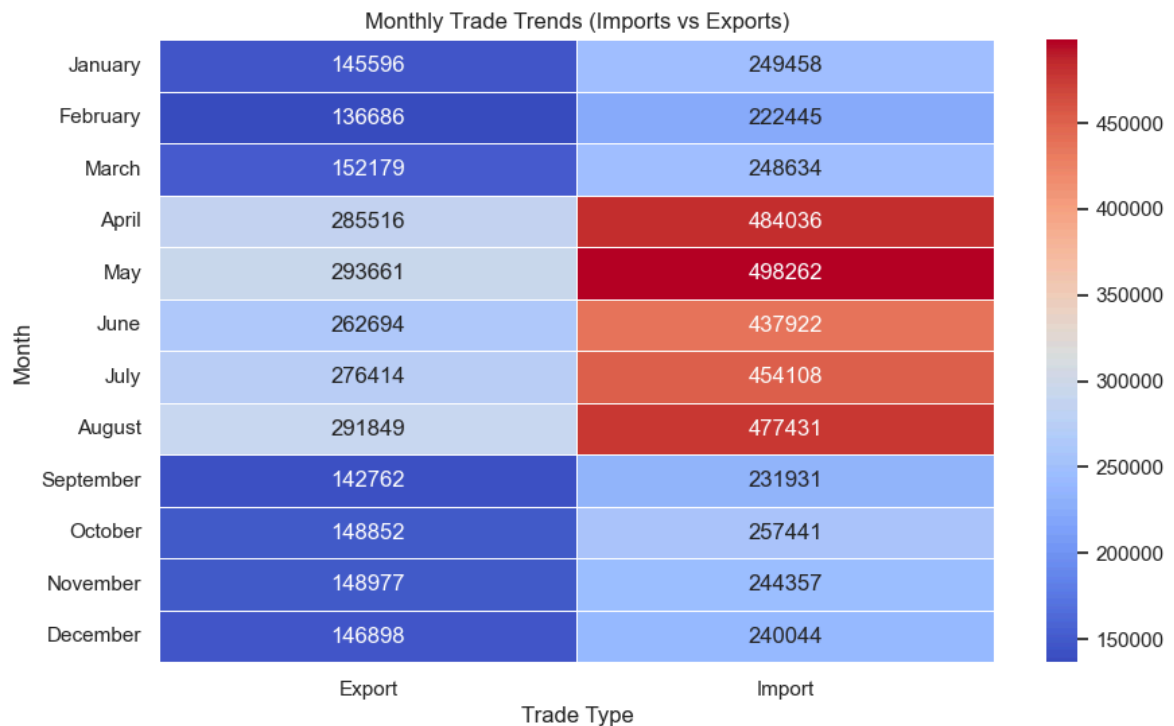


21.Monthly Trade Trends (Imports vs Exports)

```
In [277... pivot_table = df.pivot_table(values="Value in Rupees (Crore)", index="Month", co
fig, ax = plt.subplots(figsize=(10, 6))
sns.heatmap(pivot_table, cmap="coolwarm", annot=True, fmt=".0f", linewidths=0.5,
plt.title("Monthly Trade Trends (Imports vs Exports)")
plt.ylabel("Month")
plt.xlabel("Trade Type")
plt.show()
```

C:\Users\vinay\AppData\Local\Temp\ipykernel_18904\1820503796.py:1: FutureWarning: The default value of observed=False is deprecated and will change to observed=True in a future version of pandas. Specify observed=False to silence this warning and retain the current behavior

```
pivot_table = df.pivot_table(values="Value in Rupees (Crore)", index="Month", columns="TRADE", aggfunc="sum")
```



Observations:

1)Crude Oil Dominates Imports:

- *) Crude oil has the highest import quantity and trade value among all products.
- *) The total import value of crude oil is significantly higher compared to other commodities.

2)LPG and MS (Motor Spirit) Are Major Imports:

- *) LPG is the second-highest imported product after crude oil.
- *) MS (Motor Spirit) is imported in smaller quantities but has a substantial trade value.

3. Trade Value Trends Over Time:

- *) There is a seasonal pattern in trade, with some months showing higher imports.
- *) The value of imports fluctuates based on global oil prices and demand.

4. Export Trends Are Lower Compared to Imports:

- *) The dataset primarily consists of import transactions, with fewer export records.

*) Some petroleum products are exported, but the value is relatively lower compared to imports.

5. Correlation Between Quantity and Value:

*) A strong positive correlation exists between the quantity imported and its trade value.

*) Products with higher metric tonnes generally contribute more to the total trade value.

6) Price Fluctuations in Different Months:

*) The cost per metric tonne varies across different months, indicating changes in market prices.

*) Some months see a sharp increase in trade value even when quantity remains similar.

In []:

In []: