PWP ETE

submitted by Harish G D (2327422)

Introduction:

PEPFAR: The U.S. President's Emergency Plan for AIDS Relief (PEPFAR)is the largest commitment by any nation to address a single disease in history, enabled by strong bipartisan support across ten U.S. congresses and four presidential administrations, and through the American people's generosity. PEPFAR shows the power of what is possible through compassionate, cost-effective, accountable, and transparent American foreign assistance.

PEPFAR is managed and overseen by the U.S. Department of State's Office of the U.S. Global AIDS Coordinator and Health Diplomacy. PEPFAR leverages the power of a whole-of-government approach to controlling the global HIV/AIDS epidemic, implemented by seven other U.S. government departments and agencies: the U.S Agency for International Development; the U.S. Department of Health and Human Services and its agencies, including the Centers for Disease Control and Prevention, Health Resources and Service Administration and the National Institutes of Health; the U.S. Department of Defense; the Peace Corps; the U.S. Department of Labor; the U.S. Department of Commerce; and the U.S. Department of the Treasury.

PEPFAR's investments also have strengthened the systems that drive effective, efficient, and sustainable health care. These investments create a lasting health system for partner countries to confront other current and future health challenges and enhance global health security.

PEPFAR's transformative, lifesaving impact is unassailable, but our mission is not yet finished. The HIV pandemic continues to evolve in every community and country, and PEPFAR constantly adapts to address new risk groups, new health challenges, and persistent gaps.

PEPFAR Supply Chain Management System (SCMS) Delivery History: About the Dataset:

The PEPFAR Supply Chain Management System (SCMS) Delivery History dataset provides valuable insights into the distribution and delivery of essential healthcare commodities facilitated by the President's Emergency Plan For AIDS Relief (PEPFAR). PEPFAR aims to combat the global HIV/AIDS epidemic by providing critical resources, including antiretroviral drugs, laboratory supplies, and other medical commodities, to countries in need.Overall, the PEPFAR SCMS Delivery History dataset serves as a valuable resource for stakeholders involved in global health supply chain management, including policymakers, healthcare providers, non-governmental organizations (NGOs), and researchers. By analyzing this data, stakeholders can identify trends, optimize supply chain processes, address challenges, and ultimately improve access to life-saving healthcare commodities for populations affected by HIV/AIDS.

Source: https://catalog.data.gov/dataset/pepfar-supply-chain-management-system-scms-delivery-history

BUSINESS PROBLEM

The analysis of the PEPFAR Supply Chain Management System (SCMS) Delivery History dataset aims to address various critical aspects of the supply chain for delivering essential healthcare commodities to populations affected by HIV/AIDS. The overarching business problem involves optimizing logistical efficiency, vendor performance, inventory management, geographical

accessibility, quality assurance, and cost optimization. By improving these facets, stakeholders can enhance the delivery of healthcare commodities, ensure equitable access to resources, maintain high-quality standards, and maximize the cost-effectiveness of HIV/AIDS treatment and prevention efforts.

OBJECTIVES

1.Distribution of Durations and Line Item Quantities

2.Lead Time Analysis and On-Time Delivery Analysis

3.Insurance Coverage and Shipment Mode Analysis

4.Procurement Process Analysis

5.Manufacturing Site Analysis and country wise Demand analysis

6.Vendor performance Analysis

7.Additional Correlation and Cluster Analysis

Approach To Data (Systematic Approach)

DATA DICTIONARY

ID	FieldName	FieldDescription	FieldNotes	DataType
1	ID	Primary key indentifer of the line of data in our analytical tool		Number
2	Project Code	Project code	Only includes PEPFAR project codes	Text
3	PQ #	Price quote (PQ) number	"Pre-PQ Process" indicates deliveries that occurred before the PQ process was put in place in mid-2009.	Text
4	PO #	Order number: Purchase order (PO) for Direct Drop deliveries, or Sales Order (SO) for from Regional Delivery Center (RDC) deliveries	PO # is not applicable for from RDC deliveries ("NA - From RDC")	Text
5	ASN/DN #	Shipment number: Advanced Shipment Note (ASN) for Direct Drop deliveries, or Delivery Note (DN) for from RDC deliveries		Text
6	Country	Destination country		Text
7	Managed By	SCMS managing office: either the Program Management Office (PMO) in the U.S. or the relevant SCMS field office		Text
8	Fulfill Via	Method through which the shipment was fulfilled: via Direct Drop from vendor or from stock available in the RDCs		Text
9	Vendor INCO Term	The vendor INCO term (also known as International Commercial Terms) for Direct Drop deliveries	Not applicable for from RDC deliveries ("NA - From RDC")	Text
10	Shipment Mode	Method by which commodities are shipped		Text
11	PQ First Sent to Client Date	Date the PQ is first sent to the client	"Pre-PQ Process" indicates deliveries that occurred before the PQ process was put in place in mid-2009. "Date Not Captured" where date was not captured.	Date/Time
12	PO Sent to Vendor Date	Date the PO is first sent to the vendor	Not applicable for from RDC deliveries ("NA - From RDC"). "Date Not Captured" where date was not captured.	Date/Time
13	Scheduled Delivery Date	Current anticipated delivery date	This date is not equivalent to the client promised delivery date and should not be used to determine on-time perform.	Date/Time
14	Delivered to Client Date	Date of delivery to client	Transactions are included in the dataset only after the goods have been delivered to the client	Date/Time
15	Delivery Recorded Date	Date on which delivery to client was recorded in SCMS information systems	This date is used for official SCMS reporting. Deliveries are only recorded in SCMS systems once all necessary documentation has been received. Due to documentation delays there can be a lag between the time goods are	

			physically delivered to the client and the date on which all necessary documentation has been received.	
16	Product Group	Product group for item, i.e. ARV, HRDT	ACT, ANTM, ARV, HRDT, MRDT only	Text
17	Sub Classification	Identifies relevant product sub classifications, such as whether ARVs are pediatric or adult, whether a malaria product is an artemisinin-based combination therapy (ACT), etc.		Text
18	Vendor	Vendor name	SCMS is the vendor for from RDC deliveries (product can be from multiple manufacturers, based on available stock)	Text
19	Item Description	Product name and formulation from Partnership for Supply Chain Management (PFSCM) Item Master		Text
20	Molecule/Test Type	Active drug(s) or test kit type		Text
21	Brand	Generic or branded name for the item		Text
22	Dosage	Item dosage and unit		Text
23	Dosage Form	Dosage form for the item (tablet, oral solution, injection, etc.).	"FDC" denotes if the item contains a fixed-dose combination (FDC) formulation. "Blister" denotes if the item is presented in blister packaging. "Co-blister" denotes when the item contains more than one product packaged together in blister packaging.	Text
24	Unit of Measure (Per Pack)	Pack quantity (pills or test kits) used to compute unit price		Number
25	Line Item Quantity	Total quantity (packs) of commodity per line item		Number
26	Line Item Value	Total value of commodity per line item		Currency (USD)
27	Pack Price	Cost per pack (i.e. month's supply of ARVs, pack of 60 test kits)		Currency (USD)
28	Unit Price	Cost per pill (for drugs) or per test (for test kits)		Currency (USD)
29	Manufacturing Site	Identifies manufacturing site for the line item for direct drop and from RDC deliveries		Text
30	First Line Designation	Designates if the line in question shows the aggregated freight costs and weight associated with all items on the ASN/DN	There may or may not be other associated lines with each ASN/DN	Binary
31	Weight (Kilograms)	Weight for all lines on an ASN/DN	Present only for FirstLine designated lines	Number

32	Freight Cost (USD)	Freight charges associated with all lines on the respective ASN/DN	Present only for FirstLine designated lines. For C- and D-vendor INCO term deliveries, freight costs may be included in the unit price for the commodities as indicated by "Freight Included in Commodity Price". All other lines are "Invoiced Separately"	Currency (USD)
33	Line Item Insurance (USD)	Line item cost of insurance, created by applying an annual flat rate (%) to commodity cost	Pre 6/1/2009 lines are still under analysis for correct rates because they do not have PQs and cannot be computed in the same way that they currently are across the partnership	Currency (USD)

Importing Needed libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
```

Data Loading and Preprocessing:

Problem Statement: Loading the delivery history dataset, cleaning the data by removing rows with missing values, and parse date columns.

```
df = pd.read_excel(r"C:\Users\haris\OneDrive\Desktop\
SCMS_Delivery_History_Dataset_20150929.xlsx")
df.head(5)
#Dropping N/A values
df.dropna(inplace=True)
#Creating a new frame with date columns for pre processing
date_columns = ['PO Sent to Vendor Date', 'PQ First Sent to Client
Date', 'Scheduled Delivery Date', 'Delivered to Client Date','Delivery
Recorded Date']
for column in date_columns:
    #The to_datetime function in pandas is used to convert argument to
datetime.
    #When set to 'coerce', any errors encountered during conversion
will be set as NaT (Not a Time) values.
    df[column] = pd.to_datetime(df[column], errors='coerce')

# Removing rows with NaT values after date parsing
df.dropna(subset=date_columns, inplace=True)


C:\Users\haris\AppData\Local\Temp\ipykernel_18268\3060285880.py:10:
UserWarning: Could not infer format, so each element will be parsed
individually, falling back to `dateutil`. To ensure parsing is
consistent and as-expected, please specify a format.
  df[column] = pd.to_datetime(df[column], errors='coerce')
```

Basic EDA using describe() Function.

```
print(df.describe())
print(df.info())
```

```
                 ID   PQ First Sent to Client Date  \
count   2673.000000                           2673
mean   47471.971942  2012-08-29 17:43:58.383838464
min    12935.000000            2009-02-09 00:00:00
25%    29957.000000            2010-10-08 00:00:00
50%    46942.000000            2013-03-05 00:00:00
75%    65263.000000            2014-04-30 00:00:00
max    82256.000000            2015-12-01 00:00:00
std    20095.863042                            NaN

              PO Sent to Vendor Date        Scheduled Delivery Date  \
count                           2673                           2673
mean   2012-10-09 00:37:42.626262528  2013-01-30 08:13:28.080807936
min              2009-02-11 00:00:00            2009-08-05 00:00:00
25%              2010-11-17 00:00:00            2011-04-08 00:00:00
50%              2013-02-13 00:00:00            2013-08-07 00:00:00
75%              2014-08-13 00:00:00            2014-09-12 00:00:00
max              2015-12-02 00:00:00            2015-09-14 00:00:00
std                              NaN                            NaN

            Delivered to Client Date         Delivery Recorded Date  \
count                           2673                           2673
mean   2013-01-31 18:50:46.464646656  2013-02-02 06:51:02.626262784
min              2009-08-05 00:00:00            2009-08-05 00:00:00
25%              2011-04-16 00:00:00            2011-04-16 00:00:00
50%              2013-08-07 00:00:00            2013-08-07 00:00:00
75%              2014-09-12 00:00:00            2014-09-19 00:00:00
max              2015-09-14 00:00:00            2015-09-14 00:00:00
std                              NaN                            NaN

       Unit of Measure (Per Pack)  Line Item Quantity  Line Item Value
\
count                 2673.000000         2673.000000     2.673000e+03

mean                    89.079312        17738.256640     1.131113e+05

min                      5.000000            1.000000     1.000000e-02

25%                     30.000000          135.000000     2.160000e+03

50%                     60.000000         2000.000000     1.827324e+04

75%                     90.000000        17952.000000     1.310265e+05

max                   1000.000000       515000.000000     2.801262e+06

std                    101.193801        39791.922648     2.233653e+05
```

```
        Pack Price   Unit Price  Line Item Insurance (USD)
count  2673.000000  2673.000000                2673.000000
mean     18.283345     0.285230                 179.256607
min       0.000000     0.000000                   0.000000
25%       4.600000     0.090000                   2.740000
50%       9.760000     0.150000                  26.400000
75%      23.000000     0.380000                 190.410000
max     306.880000    14.040000                3487.130000
std      24.651571     0.404332                 357.974695
<class 'pandas.core.frame.DataFrame'>
Index: 2673 entries, 2682 to 6579
Data columns (total 33 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   ID                              2673 non-null   int64
 1   Project Code                    2673 non-null   object
 2   PQ #                            2673 non-null   object
 3   PO / SO #                       2673 non-null   object
 4   ASN/DN #                        2673 non-null   object
 5   Country                         2673 non-null   object
 6   Managed By                      2673 non-null   object
 7   Fulfill Via                     2673 non-null   object
 8   Vendor INCO Term                2673 non-null   object
 9   Shipment Mode                   2673 non-null   object
 10  PQ First Sent to Client Date    2673 non-null   datetime64[ns]
 11  PO Sent to Vendor Date          2673 non-null   datetime64[ns]
 12  Scheduled Delivery Date         2673 non-null   datetime64[ns]
 13  Delivered to Client Date        2673 non-null   datetime64[ns]
 14  Delivery Recorded Date          2673 non-null   datetime64[ns]
 15  Product Group                   2673 non-null   object
 16  Sub Classification              2673 non-null   object
 17  Vendor                          2673 non-null   object
 18  Item Description                2673 non-null   object
 19  Molecule/Test Type              2673 non-null   object
 20  Brand                           2673 non-null   object
 21  Dosage                          2673 non-null   object
 22  Dosage Form                     2673 non-null   object
 23  Unit of Measure (Per Pack)      2673 non-null   int64
 24  Line Item Quantity              2673 non-null   int64
 25  Line Item Value                 2673 non-null   float64
 26  Pack Price                      2673 non-null   float64
 27  Unit Price                      2673 non-null   float64
 28  Manufacturing Site              2673 non-null   object
 29  First Line Designation          2673 non-null   object
 30  Weight (Kilograms)              2673 non-null   object
 31  Freight Cost (USD)              2673 non-null   object
 32  Line Item Insurance (USD)       2673 non-null   float64
dtypes: datetime64[ns](5), float64(4), int64(3), object(21)
```

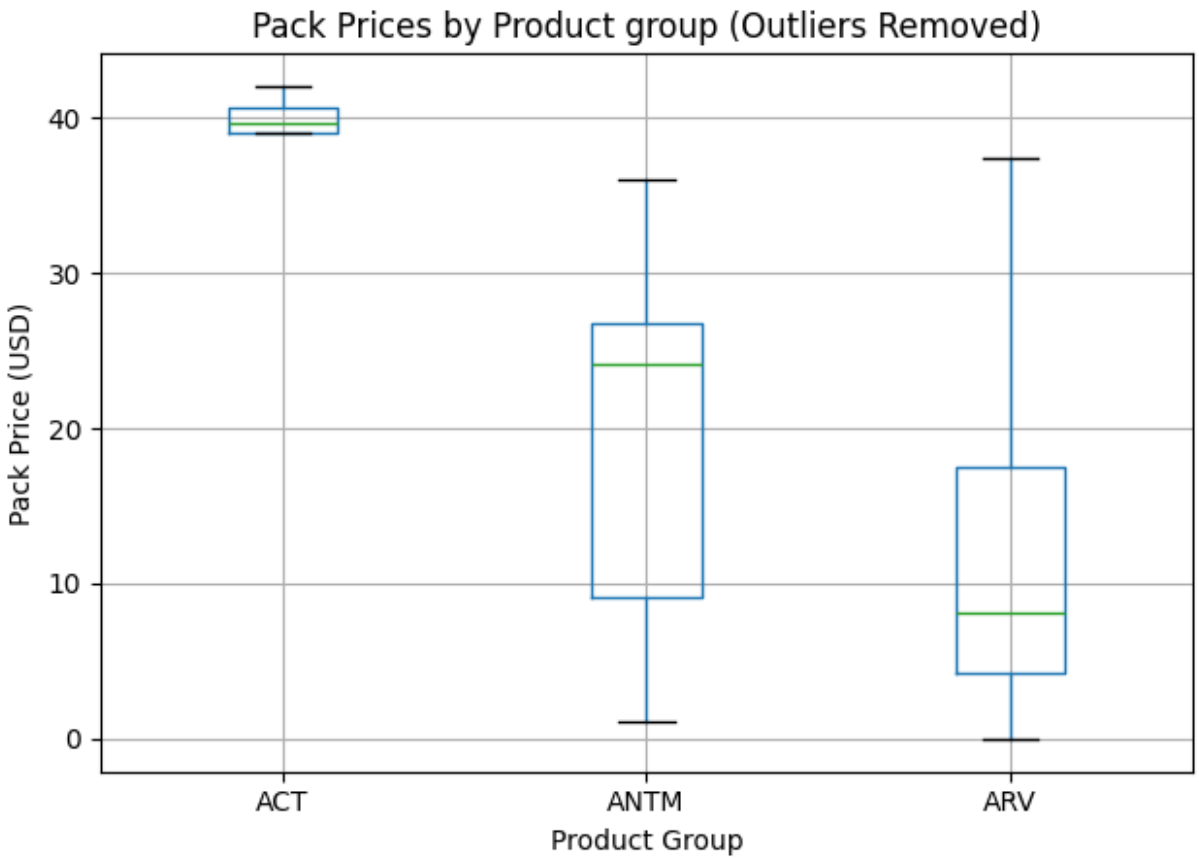
```
memory usage: 710.0+ KB
None
```

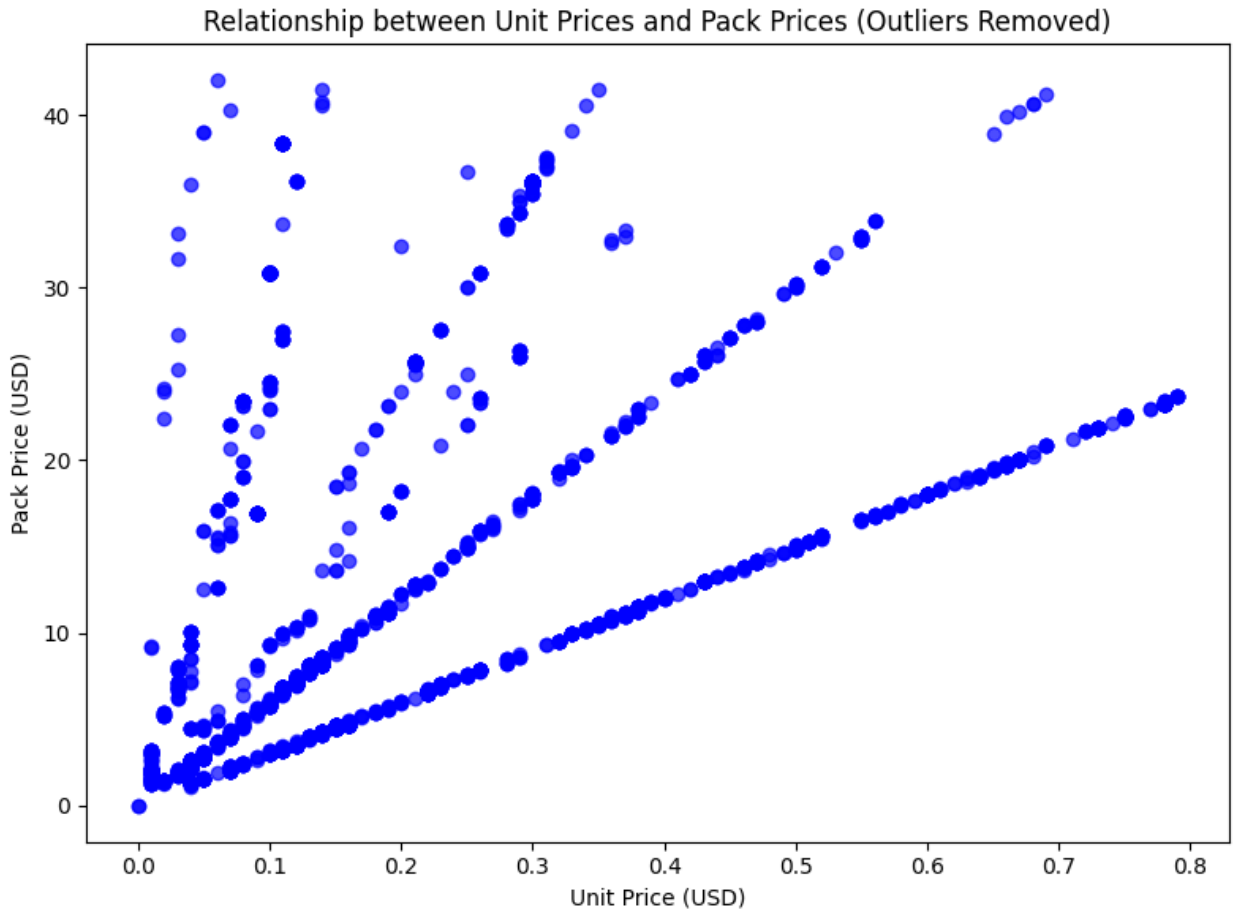
Outlier Removal and Visualization:

Problem Statement: Removing outliers from unit prices and pack prices and visualize the distribution.

```
#Creating a func to reduce outliers
def remove_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return df[(df[column] >= lower_bound) & (df[column] <=
upper_bound)]

# Removing outliers from both unit prices and pack prices
df_cleaned = remove_outliers(df, 'Unit Price')
df_cleaned = remove_outliers(df_cleaned, 'Pack Price')

# Visualization - Box plot to compare unit prices and pack prices
across different suppliers (after removing outliers)
plt.figure(figsize=(10, 6))

df_cleaned.boxplot(column='Pack Price', by='Product
Group',showfliers=False)
plt.title('Pack Prices by Product group (Outliers Removed)')
plt.ylabel('Pack Price (USD)')
plt.xlabel('Product Group')
plt.suptitle('')
plt.tight_layout()
plt.show()

# Visualization - Scatter plot to examine the relationship between
unit prices and pack prices (after removing outliers)
plt.figure(figsize=(8, 6))
plt.scatter(df_cleaned['Unit Price'], df_cleaned['Pack Price'],
color='blue', alpha=0.7)
plt.title('Relationship between Unit Prices and Pack Prices (Outliers
Removed)')
plt.xlabel('Unit Price (USD)')
plt.ylabel('Pack Price (USD)')
plt.tight_layout()
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```


Pack Prices by Product group (Outliers Removed)
Pack Price (USD)
40
30
20
10
0
ACT
ANTM
ARV
Product Group



The median price appears to be lowest for product group ACT, followed by ARV and then ANTM. The box shows the spread of the data around the median price. The box extends from the first quartile (Q1) to the third quartile (Q3). The interquartile range (IQR) is the difference between Q3 and Q1 and is a measure of the dispersion of the data. The larger the IQR, the greater the spread of the data. The box for product group ANTM has the largest IQR, which means that the prices in this group are more spread out than the prices in the other two groups. The median price is lowest for product group ACT, and the prices in product group ANTM are more spread out than the prices in the other two groups.

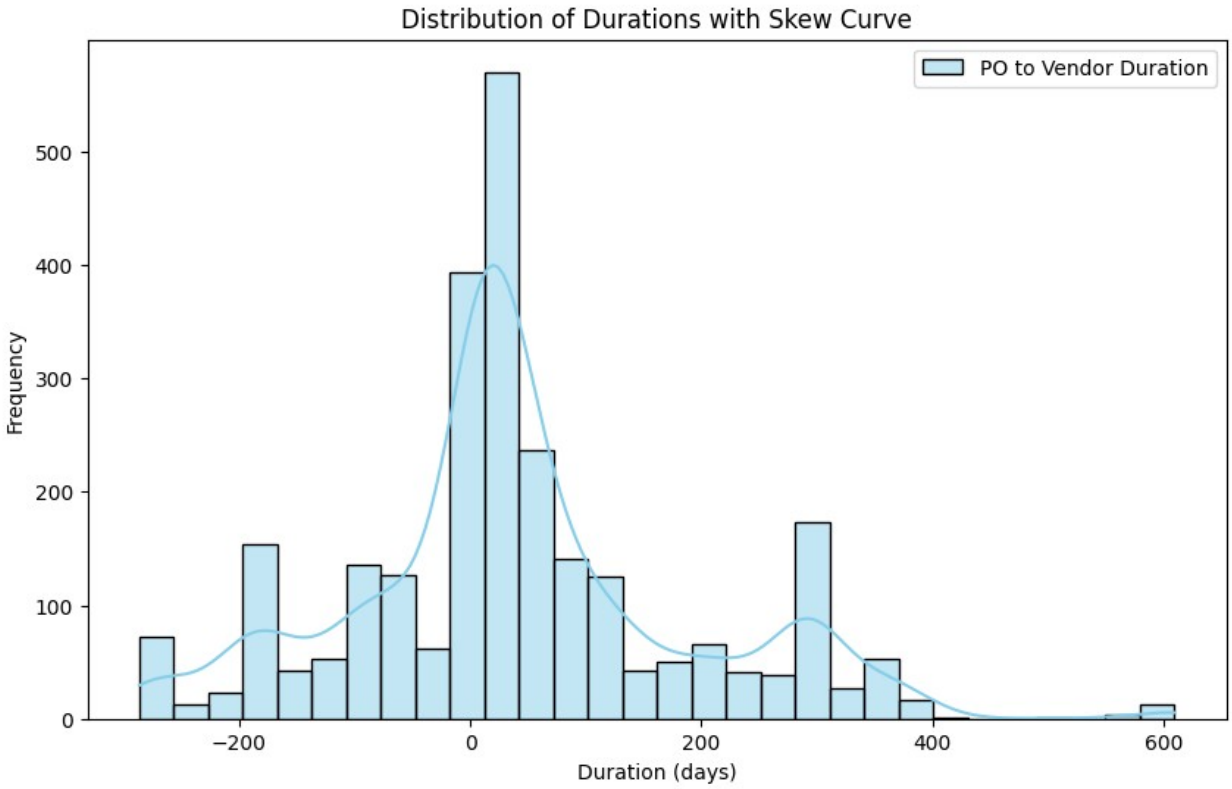
There is a positive correlation between unit price and pack price in the scatter plot. This means that as the unit price increases, the pack price also tends to increase. However, the data points are scattered, so there is not a perfect linear relationship between the two variables.

1. Distribution of Durations:

Problem Statement: Analyzing and visualizing the distribution of durations for sending POs to vendors and delivery to clients.

```
#Creating a new column named po_to_vendor_duration
#dt.days is an accessor used to extract the number of days from a
Timedelta object.
```

```
df['PO_to_Vendor_Duration'] = (df['PO Sent to Vendor Date'] - df['PQ
First Sent to Client Date']).dt.days

# Ploting histograms for distributions of durations with skew
curve(kde) using seaborn.
plt.figure(figsize=(10, 6))
sns.histplot(df['PO_to_Vendor_Duration'].dropna(), bins=30,
color='skyblue', kde=True, label='PO to Vendor Duration')
plt.title('Distribution of Durations with Skew Curve')
plt.xlabel('Duration (days)')
plt.ylabel('Frequency')
plt.legend()
plt.show()
```



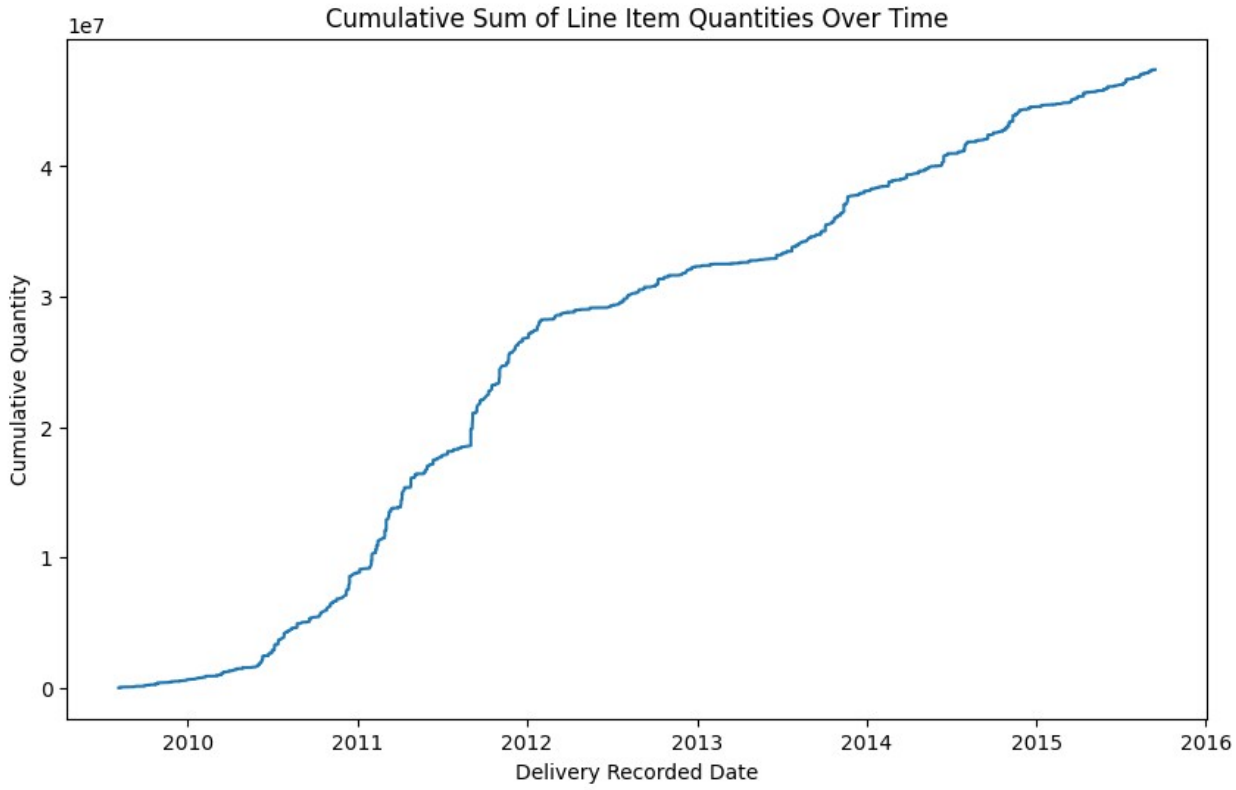
The graph displays a right-skewed distribution, meaning most durations fall on the shorter side (left side of the graph) with a longer tail extending towards the right side. This indicates that there are more instances of shorter durations than longer durations. The median (the center point) is likely lower than the mean (the average) due to the skew.

Cumulative Sum of Line Item Quantities Over Time:

Problem Statement: Visualizing the cumulative sum of line item quantities over time.

```
# Sorting DataFrame by delivery dates
df = df.sort_values(by='Delivery Recorded Date')
```

```
# Calculating cumulative sum of line item quantities
df['Cumulative Quantity'] = df['Line Item Quantity'].cumsum()

# Plotting cumsum of line quantities
plt.figure(figsize=(10, 6))
plt.plot(df['Delivery Recorded Date'], df['Cumulative Quantity'])
plt.title('Cumulative Sum of Line Item Quantities Over Time')
plt.xlabel('Delivery Recorded Date')
plt.ylabel('Cumulative Quantity')
plt.show()
```



The line starts at a cumulative quantity of zero and increases steadily over time. This means that the number of line items being shipped is increasing over time. There are also some fluctuations in the line, which means that the number of items being shipped is not always increasing at a constant rate. For example, there appears to be a significant increase in shipments in 2012. The graph can be used to compare the cumulative quantity of items shipped at different points in time. For example, the graph shows that the cumulative quantity of items shipped in 2016 was much higher than the cumulative quantity of items shipped in 2010.

2. Lead Time Analysis:

Problem Statement: Calculating and analyzing lead times, including mean lead time, standard deviation, reorder point, and safety stock level.

```
#Lead time refers to the amount of time it takes for an order to be
fulfilled from the moment it is placed until it is delivered to the
customer
# Calculating lead times (in days)
df['Lead Time'] = (df['Delivered to Client Date'] - df['Scheduled
Delivery Date']).dt.days

# Displaying lead times
print("Lead Times (in days):")
print(df['Lead Time'])

# Calculating basic statistics on lead times
mean_lead_time = df['Lead Time'].mean()
std_lead_time = df['Lead Time'].std()

# Analyzing lead time variability
print("\nLead Time Statistics:")
print("Mean Lead Time:", mean_lead_time)
print("Standard Deviation of Lead Time:", std_lead_time)

#The reorder point is the inventory level at which a new order should
be placed to replenish stock before it runs out.
# Determining reorder point (example: 95th percentile of lead times)
#In the context of inventory management or supply chain analysis,
quantile(0.95) is often used to calculate the reorder point. By
determining the 95th percentile of lead times,
#for example, a company can establish the reorder point at a level
that ensures 95% of orders are received before the inventory runs out.
reorder_point = df['Lead Time'].quantile(0.95)
print("\nReorder Point (95th percentile of lead times):",
reorder_point)

#Safety stock, also known as buffer stock, is the extra inventory held
by a company to mitigate the risk of stockouts due to unexpected
fluctuations in demand or supply lead times
# Calculating safety stock level.The value 1.64 corresponds to the Z-
score associated with a 95% service level in a standard normal
distribution.
safety_stock = 1.64 * std_lead_time
print("Safety Stock Level (for 95% service level):", safety_stock)
```

```
Lead Times (in days):
3055    0
4754    0
2806    0
3894    0
3422    0
       ..
4005    0
6575    0
```

```
3810    0
5846    0
3749    0
Name: Lead Time, Length: 2673, dtype: int64

Lead Time Statistics:
Mean Lead Time: 1.4425738870183316
Standard Deviation of Lead Time: 11.372132050965742

Reorder Point (95th percentile of lead times): 0.40000000000009095
Safety Stock Level (for 95% service level): 18.650296563583815
```

The lead time analysis reveals the following insights:

Lead Time Distribution: The majority of deliveries have a lead time of 0 days, indicating that they were delivered on the scheduled delivery date.

Mean Lead Time: The average lead time across all deliveries is approximately 1.44 days.

Standard Deviation of Lead Time: The standard deviation of lead times is relatively high at around 11.37 days, indicating considerable variability in lead times.

Reorder Point: The 95th percentile of lead times is approximately 0.4 days, suggesting that only a small percentage of deliveries experience longer lead times.

Safety Stock Level: Based on a 95% service level, the calculated safety stock level is approximately 18.65 days. This safety stock helps mitigate the risk of stockouts due to variability in lead times, ensuring a high level of service to clients.

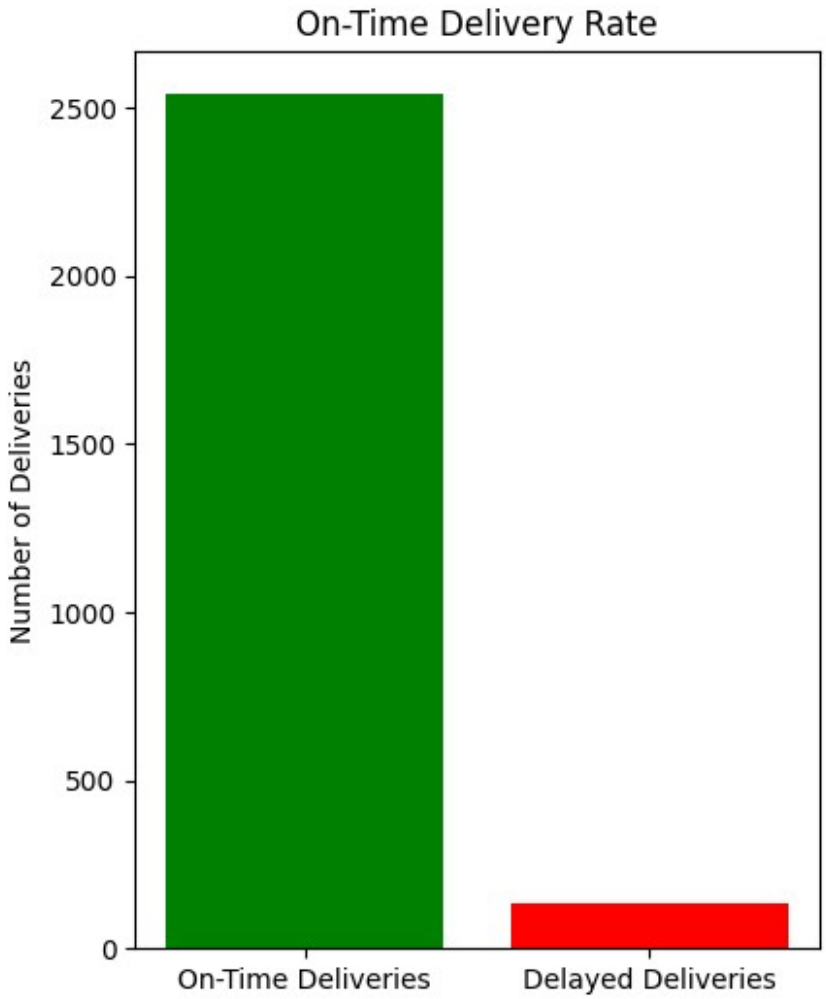
Overall, while the mean lead time is relatively short, the high standard deviation indicates variability in delivery times. Maintaining an appropriate safety stock level can help buffer against this variability and ensure timely delivery to clients.

On-Time Delivery Analysis:

Problem Statement: Calculating on-time delivery rate and average delivery delay.

```
df['Delivery Delay'] = (df['Delivered to Client Date'] - df['Scheduled
Delivery Date']).dt.days

# Calculating on-time delivery rate
#Shape[0] extracts the number of rows from the filtered DataFrame,
effectively counting the number of on-time deliveries.
on_time_deliveries = df[df['Delivery Delay'] <= 0].shape[0]
total_deliveries = df.shape[0]
on_time_delivery_rate = on_time_deliveries / total_deliveries

# Calculating average delivery delay
average_delivery_delay = df['Delivery Delay'].mean()
print("Average Delivery Delay........",average_delivery_delay,'Days')
```

```
# Visualization
plt.figure(figsize=(10, 6))
plt.subplot(1, 2, 1)
plt.bar(['On-Time Deliveries', 'Delayed Deliveries'],
[on_time_deliveries, total_deliveries - on_time_deliveries],
color=['green', 'red'])
plt.title('On-Time Delivery Rate')
plt.ylabel('Number of Deliveries')
plt.show()

Average Delivery Delay........ 1.4425738870183316 Days
```

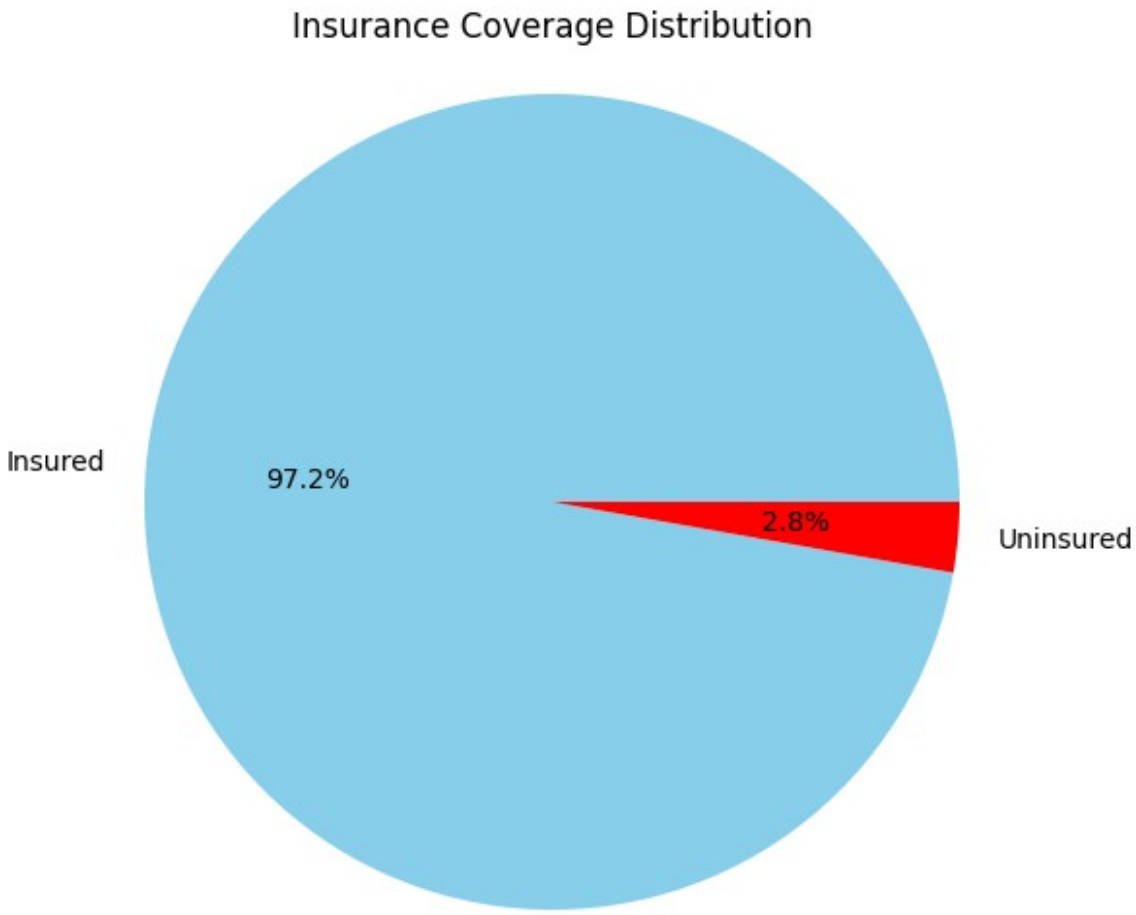


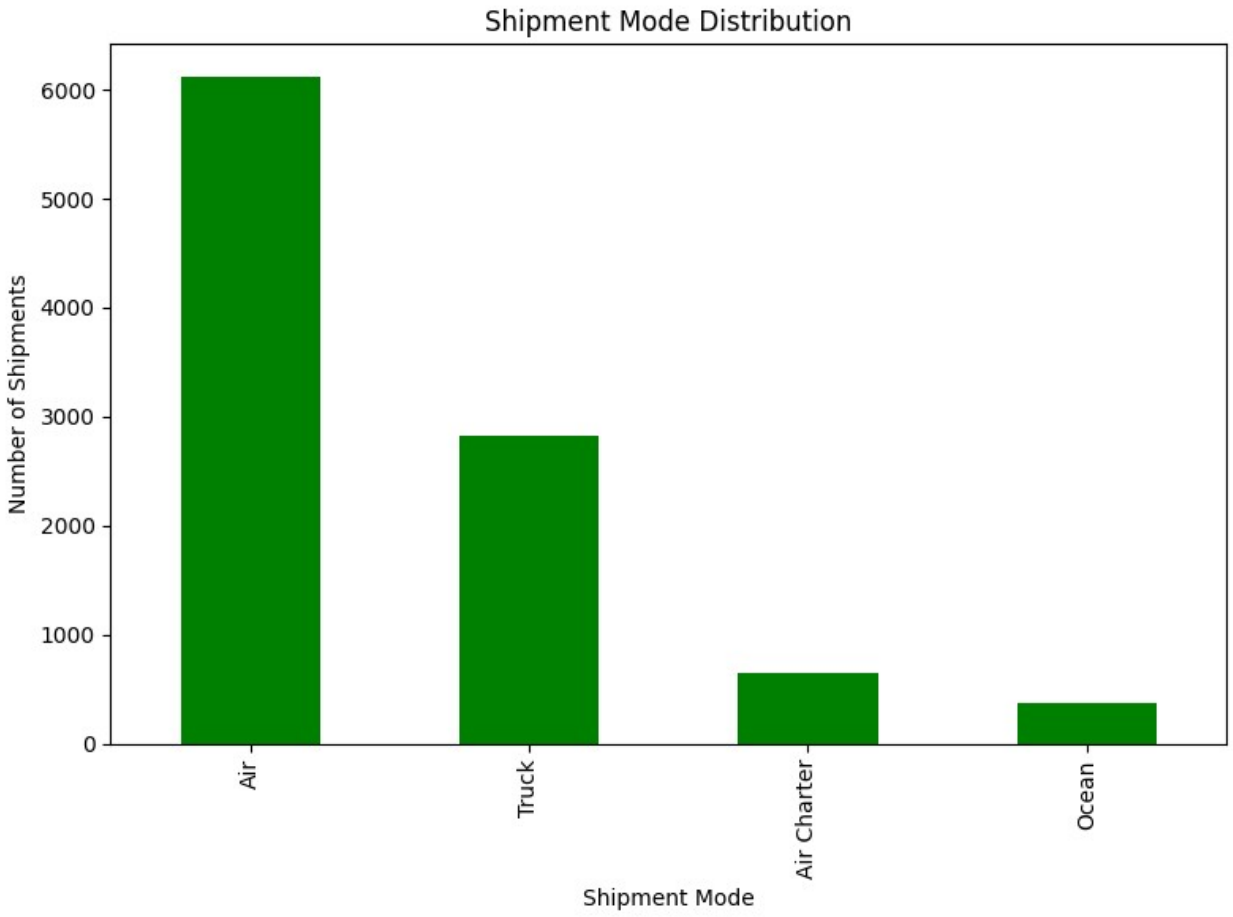
on-time deliveries are more frequent than delayed deliveries throughout the period shown in the graph. There are spikes in both on-time deliveries and delayed deliveries, but the peaks for on-time deliveries are consistently higher than the peaks for delayed deliveries. The average delivery delay is stated at the top of the graph as 1.4425738870183316. Overall, the graph suggests that the Organisation has a good on-time delivery rate.

3. Insurance Coverage and Shipment Mode Analysis:

Problem Statement: Analyzing insurance coverage distribution and shipment mode distribution.

```
df2= pd.read_excel(r"C:\Users\haris\OneDrive\Desktop\
SCMS_Delivery_History_Dataset_20150929.xlsx")
df2.dropna()
total_shipments = len(df2)
insured_shipments = df2['Line Item Insurance (USD)'].notnull().sum()
uninsured_shipments = total_shipments - insured_shipments


# Calculating shipment mode distribution
shipment_mode_counts = df2['Shipment Mode'].value_counts()

# Visualization - Pie chart for insurance coverage
#autopct='%1.1f%%' indicates that the percentage labels on the pie
chart should display the percentage values with one digit before the
decimal point and one digit after the decimal point, followed by a
percentage sign.
plt.figure(figsize=(8, 6))
plt.pie([insured_shipments, uninsured_shipments], labels=['Insured',
'Uninsured'], autopct='%1.1f%%', colors=['skyblue', 'red'])
plt.title('Insurance Coverage Distribution')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a
circle
plt.show()

# Visualization - Bar plot for shipment mode distribution
plt.figure(figsize=(8, 6))
shipment_mode_counts.plot(kind='bar', color='green')
plt.title('Shipment Mode Distribution')
plt.xlabel('Shipment Mode')
plt.ylabel('Number of Shipments')
plt.tight_layout()
plt.show()
```


Insurance Coverage Distribution
Insured
97.2%
2.8%
Uninsured



Insured: Pie chart represents 97.2% of the total shipments in the dataset. These are the shipments where insurance coverage information is available, indicating that the majority of shipments are insured against potential risks during transit or delivery. Only 2.8% indicates a smaller proportion of shipments where insurance coverage status is unknown or not provided.

The graph shows the number of shipments for four different shipment modes: Air, Truck, Air Charter, and Ocean. Here's a breakdown of the number of shipments for each mode:

Air: 6000 shipments

Truck: 3,000 shipments

Air Charter: 800 shipments

Ocean: 500 shipments

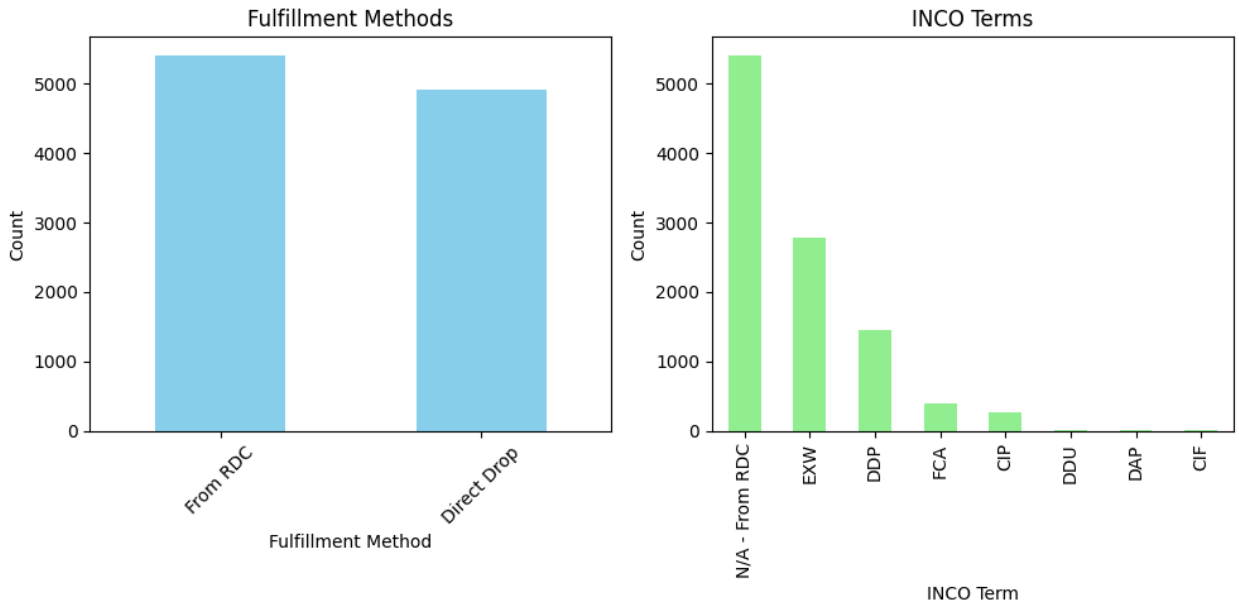
The graph shows that air is the most common shipment mode, followed by truck, air charter, and then ocean. This suggests that the Organisation ships most of its goods by air. This could be because the company needs to ship goods quickly or because the goods are perishable.

4.Procurement Process Analysis

Distribution of fulfillment methods and INCO:

The Objective is to analyze and visualize the distribution of fulfillment methods and INCO (International Commercial) terms used in procurement processes. Understanding these distributions is crucial for optimizing procurement strategies, assessing vendor performance, and ensuring compliance with contractual agreements.

```
fulfillment_counts = df2['Fulfill Via'].value_counts()
inco_counts = df2['Vendor INCO Term'].value_counts()

# Visualization - Bar plot for fulfillment methods
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
fulfillment_counts.plot(kind='bar', color='skyblue')
plt.title('Fulfillment Methods')
plt.xlabel('Fulfillment Method')
plt.ylabel('Count')
plt.xticks(rotation=45)

# Visualization - Bar plot for INCO terms
plt.subplot(1, 2, 2)
inco_counts.plot(kind='bar', color='lightgreen')
plt.title('INCO Terms')
plt.xlabel('INCO Term')
plt.ylabel('Count')


plt.tight_layout()
plt.show()
```



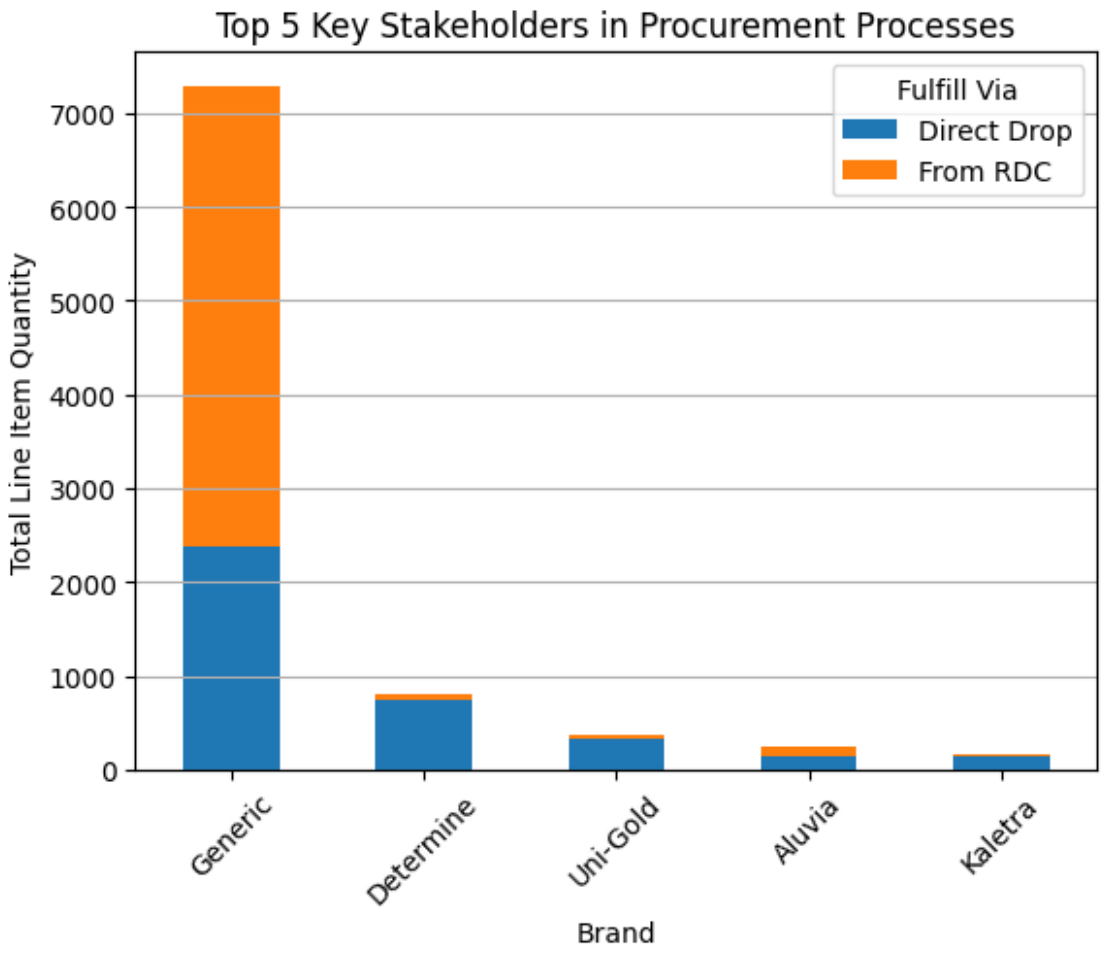
The interpretation of the fulfillment method counts suggests that the majority of instances, over 5000, involve the fulfillment method "From RDC." This indicates a significant portion of the supply chain involves items sourced from a Regional Distribution Center (RDC). On the other

hand, "From Direct Drop" accounts for approximately 4500 instances, signifying another substantial portion of the supply chain where items are directly dropped without passing through an intermediary distribution center.

The interpretation of the INCO terms distribution indicates that a vast majority, over 5000 instances, are categorized as "N/A from RDC." This suggests that a significant portion of transactions either does not have specified INCO terms or pertains to items sourced directly from a Regional Distribution Center (RDC) where INCO terms might not be applicable in the traditional sense. Additionally, there are approximately 3000 instances categorized under "EXW" (Ex Works) and 2500 instances categorized under "DDP" (Delivered Duty Paid)

```
stakeholders = df2.groupby(['Brand', 'Fulfill
Via']).size().unstack(fill_value=0)

# Select the top 5 stakeholders based on total line item quantity
top_5_stakeholders = stakeholders.sum(axis=1).nlargest(5)

# Plotting the analysis of top 10 key stakeholders
plt.figure(figsize=(10, 6))
stakeholders.loc[top_5_stakeholders.index].plot(kind='bar',
stacked=True)
plt.title('Top 5 Key Stakeholders in Procurement Processes')
plt.xlabel('Brand')
plt.ylabel('Total Line Item Quantity')
plt.xticks(rotation=45)
plt.legend(title='Fulfill Via')
plt.grid(axis='y')
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```



Top 5 key stakeholders in procurement process are listed. Out of this Generic is the top brand in the process. Over 7000 distributions and 2500 via direct drop.

5.Manufacturing Site Analysis:

The objective is to assess the compliance status for manufacturing sites based on the first line designation. This analysis aims to understand the distribution of compliance status across different manufacturing sites and first line designations, providing insights into the adherence to regulatory standards and quality control measures.

```
compliance_status_manufacturing = df.groupby(['First Line
Designation', 'Manufacturing Site']).size().unstack(fill_value=0)

# Display compliance status for manufacturing
print("\nCompliance Status for Manufacturing:")
print(compliance_status_manufacturing)

no_counts = compliance_status_manufacturing.loc['No']

# Get the manufacturing sites with the highest number of 'No'
occurrences
```

```
top_no_sites = no_counts.nlargest(5)
print("\nManufacturing Sites with the Highest Number of 'No' First
Line Designation:")
print(top_no_sites)
```

```
Compliance Status for Manufacturing:
Manufacturing Site      ABBVIE (Abbott) France  ABBVIE (Abbott) Logis.
UK  \
First Line Designation

No                                           1
2
Yes                                          8
145

Manufacturing Site      ABBVIE (Abbott) St. P'burg USA  \
First Line Designation
No                                                   0
Yes                                                  3

Manufacturing Site      ABBVIE Labs North Chicago US  \
First Line Designation
No                                                 0
Yes                                                1

Manufacturing Site      ABBVIE Ludwigshafen Germany  \
First Line Designation
No                                                28
Yes                                              193

Manufacturing Site      Aspen-OSD, Port Elizabeth, SA  \
First Line Designation
No                                                 12
Yes                                                21

Manufacturing Site      Aurobindo Unit III, India  Aurobindo Unit VII,
IN  \
First Line Designation

No                                            433
0
Yes                                           504
26

Manufacturing Site      BMS Evansville, US  BMS Meymac, France  ...  \
First Line Designation                                          ...
No                                         7                 17  ...
Yes                                        9                 21  ...
```

```
Manufacturing Site      Micro labs, Verna, Goa, India  \
First Line Designation
No                                                 10
Yes                                                25

Manufacturing Site      Mylan (formerly Matrix) Nashik  \
First Line Designation
No                                                  80
Yes                                                228

Manufacturing Site      Mylan,  H-12 & H-13, India  \
First Line Designation
No                                                2
Yes                                               6

Manufacturing Site      Novartis Pharma AG, Switzerland  \
First Line Designation
No                                                   0
Yes                                                  1

Manufacturing Site      Novartis Pharma Suffern, USA  \
First Line Designation
No                                                 0
Yes                                                5

Manufacturing Site      Ranbaxy per Shasun Pharma Ltd  \
First Line Designation
No                                                  0
Yes                                                 1

Manufacturing Site      Remedica, Limassol, Cyprus  Roche Basel  Roche
Madrid  \
First Line Designation

No                                                 0             2
3
Yes                                                1            14
8

Manufacturing Site      Strides, Bangalore, India.
First Line Designation
No                                                 7
Yes                                               79

[2 rows x 47 columns]

Manufacturing Sites with the Highest Number of 'No' First Line
Designation:
Manufacturing Site
Aurobindo Unit III, India         433
```

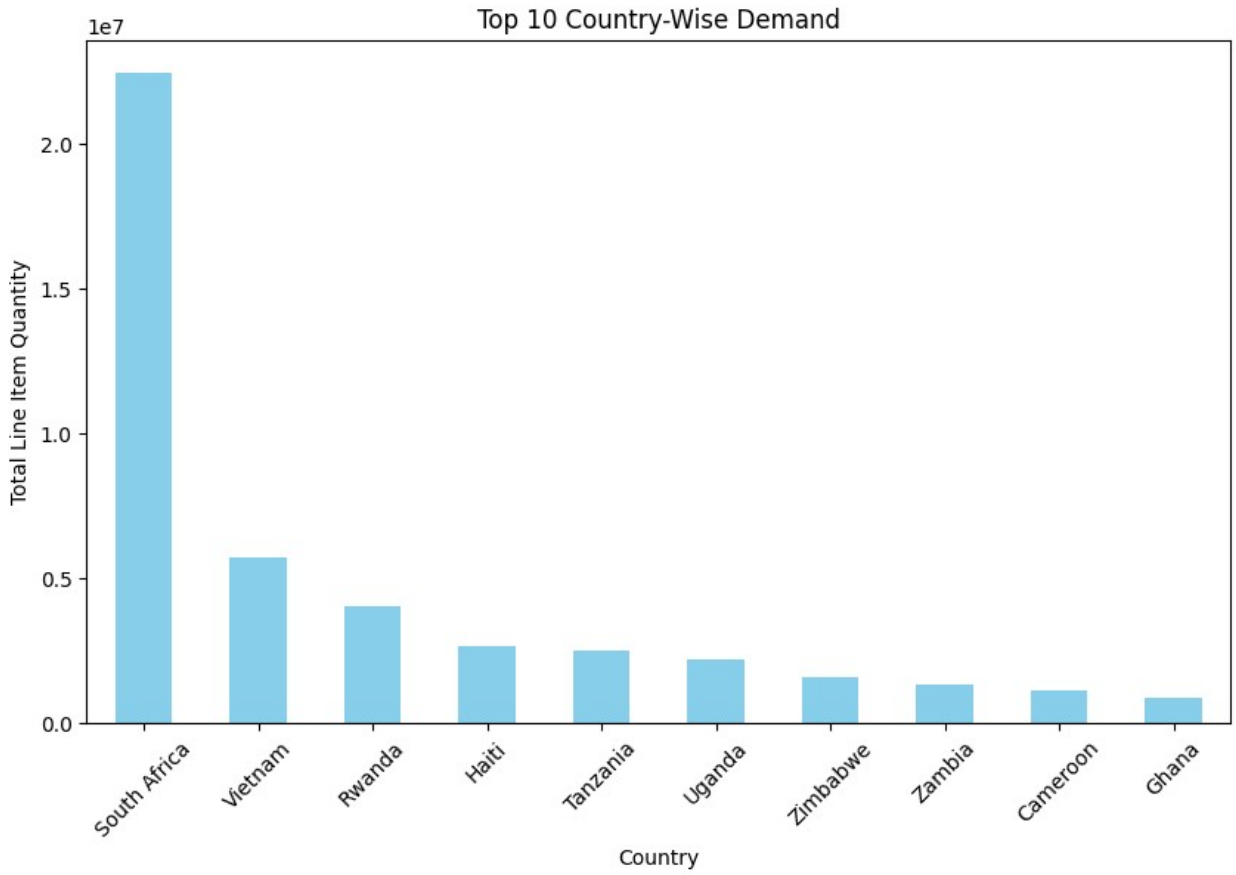
```
Hetero Unit III Hyderabad IN      109
Mylan (formerly Matrix) Nashik     80
ABBVIE Ludwigshafen Germany        28
Cipla, Goa, India                  23
Name: No, dtype: int64
```

The compliance status for manufacturing sites reveals a disparity between sites designated as "Yes" and "No" for the first-line designation. Among the manufacturing sites designated as "Yes," the majority demonstrate higher compliance, as evidenced by the larger counts. For instance, ABBVIE (Abbott) Logistics UK and ABBVIE Ludwigshafen Germany have significantly higher counts of compliant designations compared to non-compliant ones.Manufacturing Sites with the Highest Number of 'No' First Line Designation are listed.

Plotting country-wise demand for the top 10 countries:

The objective is to analyze the demand for products across different countries based on the aggregated line item quantities. This analysis aids in identifying the top countries with the highest demand for products, providing valuable insights for supply chain management, inventory planning, and market expansion strategies.

```
# Group by country and aggregate line item quantities
country_demand = df.groupby('Country')['Line Item
Quantity'].sum().sort_values(ascending=False)


top_10_countries = country_demand.head(10)

# Plotting country-wise demand for the top 10 countries
top_10_countries.plot(kind='bar', color='skyblue', figsize=(10, 6))
plt.title('Top 10 Country-Wise Demand')
plt.xlabel('Country')
plt.ylabel('Total Line Item Quantity')
plt.xticks(rotation=45)
plt.show()
```



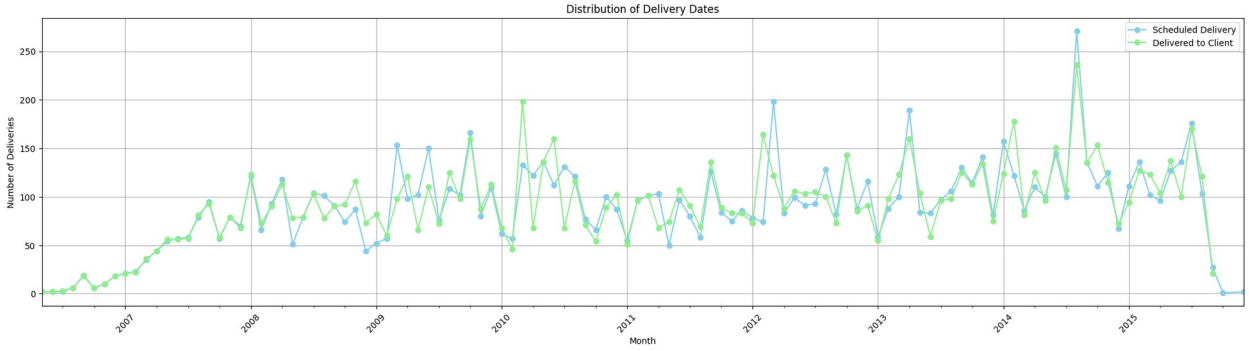
The observation that South Africa exhibits the highest demand among all countries in the dataset is crucial for supply chain management. Addressing this issue involves prioritizing and allocating more supplies to meet the demand effectively in South Africa. By recognizing South Africa's significant demand, supply chain managers can strategize procurement, production, and distribution processes to ensure adequate stock levels and timely deliveries to this region. This proactive approach helps in maintaining customer satisfaction and meeting market demands.

Distribution of Delivery Dates (Demand Forecast):

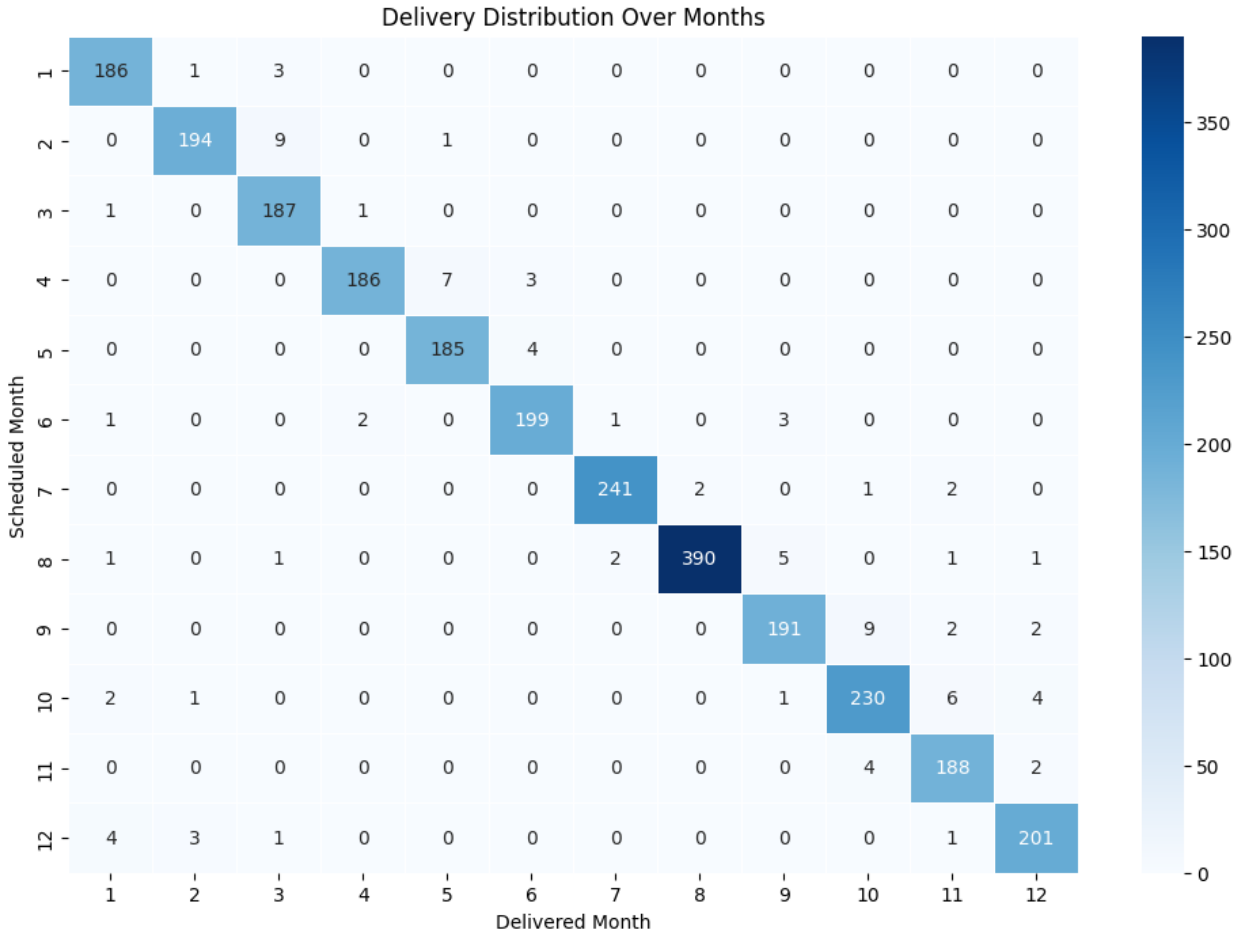
The objective is to analyze the distribution of delivery dates over time, comparing scheduled delivery dates with actual delivery dates to clients. This analysis aims to provide insights into the timeliness of deliveries and potential trends in delivery performance over different months.

```
df2['Scheduled Delivery Date'] = pd.to_datetime(df2['Scheduled
Delivery Date'])
df2['Delivered to Client Date'] = pd.to_datetime(df2['Delivered to
Client Date'])

# Count the number of deliveries per month
#Converting the 'Scheduled Delivery Date' and 'Delivered to client'
column to periods of month granularity and assigns the result to a new
column named 'Scheduled Month' and 'delivered month'.
df2['Scheduled Month'] = df2['Scheduled Delivery
```

```
Date'].dt.to_period('M')
df2['Delivered Month'] = df2['Delivered to Client
Date'].dt.to_period('M')
scheduled_counts = df2['Scheduled Month'].value_counts().sort_index()
delivered_counts = df2['Delivered Month'].value_counts().sort_index()

# Plotting the distribution of delivery dates as a line chart
plt.figure(figsize=(25, 6))
scheduled_counts.plot(kind='line', marker='o', color='skyblue',
label='Scheduled Delivery')
delivered_counts.plot(kind='line', marker='o', color='lightgreen',
label='Delivered to Client')
plt.title('Distribution of Delivery Dates')
plt.xlabel('Month')
plt.ylabel('Number of Deliveries')
plt.legend()
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```




```
# Extract month and year from the date columns
df['Scheduled Month'] = df['Scheduled Delivery Date'].dt.month
df['Delivered Month'] = df['Delivered to Client Date'].dt.month

# Group by month and count the number of deliveries
delivery_counts = df.groupby(['Scheduled Month', 'Delivered
Month']).size().unstack(fill_value=0)

# Create a heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(delivery_counts, cmap='Blues', annot=True, fmt='d',
linewidths=.5)
plt.title('Delivery Distribution Over Months')
plt.xlabel('Delivered Month')
plt.ylabel('Scheduled Month')
plt.show()
```



6.Vendor Performance Analysis:

Problem Statement: Analyzing vendor performance based on line item quantity and freight cost

Average Vendor duration:

The objective is to analyze the average duration for each stage of the delivery process, specifically the duration from the placement of purchase orders (POs) to delivery and the duration from vendor confirmation to delivery. This analysis helps in understanding the efficiency of the procurement and delivery processes, identifying potential bottlenecks, and improving overall supply chain management.

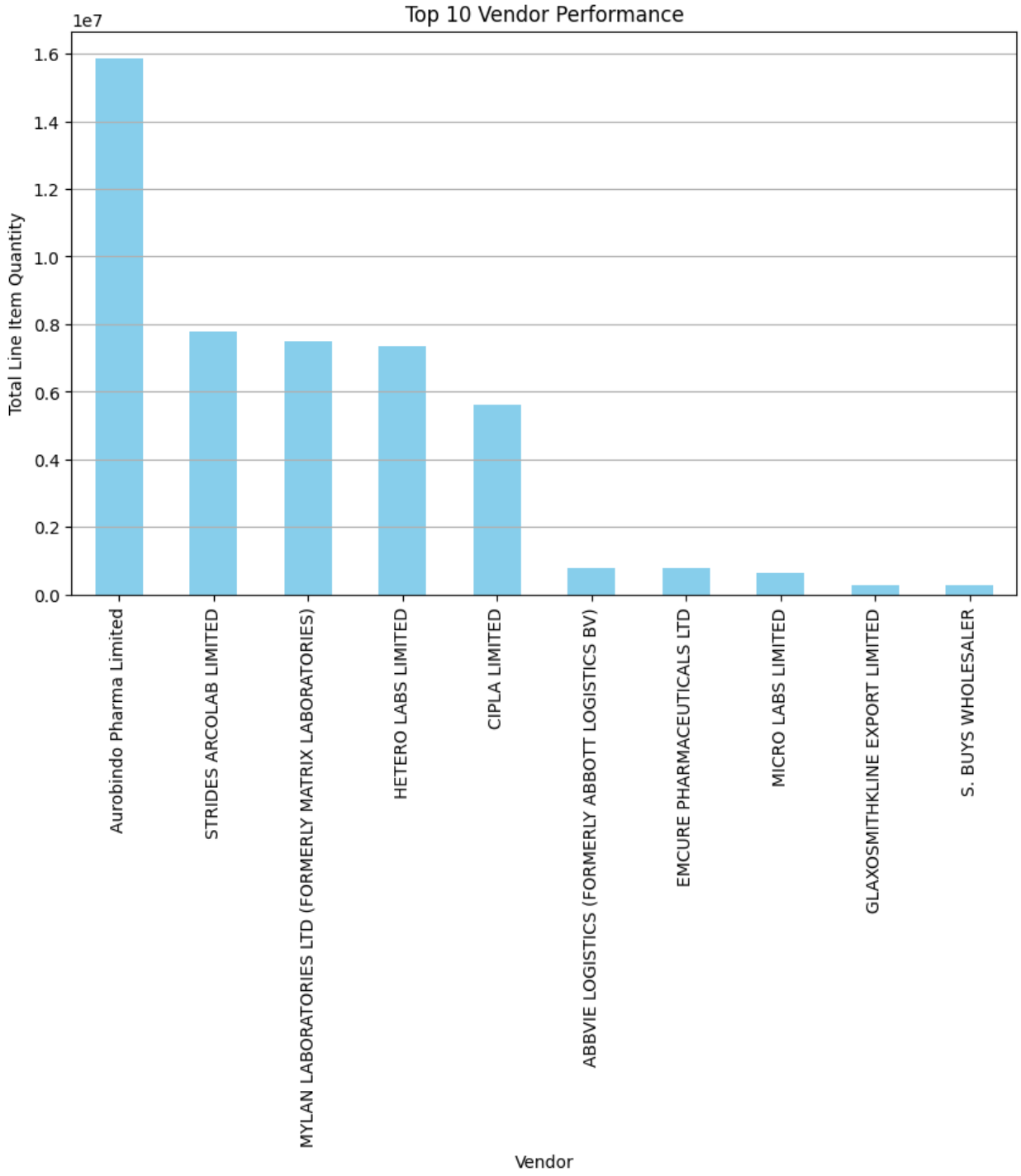
```
df.replace(["N/A - From RDC", "Date Not Captured"], np.nan,
inplace=True)

# Calculating duration for each stage in days, ignoring NaN values
df['PO to Delivery Duration'] = (df['Delivered to Client Date'] -
df['PO Sent to Vendor Date']).dt.days
df['Vendor to Delivery Duration'] = (df['Delivered to Client Date'] -
df['Scheduled Delivery Date']).dt.days

# Calculating average duration for each stage, ignoring NaN values
```

```
avg_po_to_delivery_duration = df['PO to Delivery
Duration'].mean(skipna=True)
avg_vendor_to_delivery_duration = df['Vendor to Delivery
Duration'].mean(skipna=True)

# Display average duration for each stage
print("Average PO to Delivery Duration:", avg_po_to_delivery_duration,
"days")
print("Average Vendor to Delivery Duration:",
avg_vendor_to_delivery_duration, "days")
```

```
Average PO to Delivery Duration: 114.75907220351665 days
Average Vendor to Delivery Duration: 1.4425738870183316 days
```

The average duration from purchase order (PO) to delivery is approximately 114.76 days, indicating the typical time it takes from initiating a purchase order to receiving the delivery of goods. On the other hand, the average duration from the vendor to delivery is much shorter, at approximately 1.44 days, suggesting a relatively quick turnaround time from the vendor sending the goods to their delivery to the client.

```
vendor_performance = df.groupby('Vendor')['Line Item Quantity'].sum()

# Select the top 10 vendors based on performance
top_10_vendors = vendor_performance.nlargest(10)

# Plotting the analysis of top 10 vendor performance
plt.figure(figsize=(10, 6))
top_10_vendors.plot(kind='bar', color='skyblue')
plt.title('Top 10 Vendor Performance')
plt.xlabel('Vendor')
plt.ylabel('Total Line Item Quantity')
plt.grid(axis='y')
plt.show()
```




```
vendor_performance_df = df[['Vendor', 'Line Item Quantity', 'Freight
Cost (USD)', 'Delivered to Client Date']]
# Converting 'Freight Cost (USD)' column to numeric type
vendor_performance_df['Freight Cost (USD)'] =
pd.to_numeric(vendor_performance_df['Freight Cost (USD)'],
errors='coerce')
```

```
# Grouping by vendor and aggregate metrics
vendor_performance_summary =
vendor_performance_df.groupby('Vendor').agg({
    'Line Item Quantity': 'sum',
    'Freight Cost (USD)': 'sum',
    'Delivered to Client Date': 'count'  # Counting number of
deliveries
}).reset_index()

# Droping rows with missing values (if any)
vendor_performance_summary.dropna(inplace=True)

# Renaming columns for clarity
vendor_performance_summary.columns = ['Vendor', 'Total Line Item
Quantity', 'Total Freight Cost (USD)', 'Number of Deliveries']

# Calculating average line item quantity per delivery
vendor_performance_summary['Average Line Item Quantity per Delivery']
= vendor_performance_summary['Total Line Item Quantity'] /
vendor_performance_summary['Number of Deliveries']

# Calculating average freight cost per delivery
vendor_performance_summary['Average Freight Cost per Delivery (USD)']
= vendor_performance_summary['Total Freight Cost (USD)'] /
vendor_performance_summary['Number of Deliveries']

# Writing the vendor performance summary to a CSV file
vendor_performance_summary.to_csv('vendor_performance_summary.csv',
index=False)

# Displaying the vendor performance summary
print(vendor_performance_summary)
```

```
                                                Vendor  \
0                                ABBOTT LOGISTICS B.V.
1     ABBVIE LOGISTICS (FORMERLY ABBOTT LOGISTICS BV)
2   ABBVIE, SRL (FORMALLY ABBOTT LABORATORIES INTE...
3                                  ACTION MEDEOR E.V.
4                                    AMSTELFARMA B.V.
5                                    ASPEN PHARMACARE
6                      AUROBINDO PHARAM (SOUTH AFRICA)
7                             Aurobindo Pharma Limited
8                                       B&C GROUP S.A.
9                                 BRISTOL-MYERS SQUIBB
10                                      CIPLA LIMITED
11                           EMCURE PHARMACEUTICALS LTD
12                                ETHNOR DEL ISTMO S.A.
13                        GLAXOSMITHKLINE EXPORT LIMITED
14                                  HETERO LABS LIMITED
15                             Hoffmann-La Roche ltd Basel
```

```
16                                      IDA FOUNDATION
17                                        IDIS LIMITED
18                                          IMRES B.V.
19              INTERNATIONAL HEALTHCARE DISTRIBUTORS
20  JANSSEN SCIENCES IRELAND UC (FORMERLY JANSSEN ...
21                            JSI R&T INSTITUTE, INC.
22  LAWRENCE LABORATORIES (SUBSIDIARY OF BRISTOL M...
23  MERCK SHARP & DOHME IDEA GMBH (FORMALLY MERCK ...
24                                   MICRO LABS LIMITED
25     MSD LATIN AMERICA SERVICES, S. DE R.L. DE C.V.
26  MYLAN LABORATORIES LTD (FORMERLY MATRIX LABORA...
27                         NOVARTIS PHARMA SERVICES AG
28                                     PHARMACY DIRECT
29                   PUETRO RICO PHARMACEUTICAL, INC.
30                            RAININ INSTRUMENT, LLC.
31                                 S. BUYS WHOLESALER
32                            STRIDES ARCOLAB LIMITED
33  SUN PHARMACEUTICAL INDUSTRIES LTD (RANBAXY LAB...
34                                 SWORDS LABORATORIES
35                                  SYSMEX AMERICA INC
36                           THE MEDICAL EXPORT GROUP BV

    Total Line Item Quantity  Total Freight Cost (USD)  Number of
Deliveries  \
0                       2426                      0.00
1
1                     796016                 1787679.56
305
2                       4970                    666.66
4
3                        415                    1731.89
1
4                       5388                   24810.04
7
5                     193443                   46171.17
36
6                      16500                    3908.56
1
7                   15876033                 4361108.97
561
8                        582                    1200.58
2
9                      72188                   16030.99
24
10                   5612146                 1573460.44
147
11                    790891                  177920.98
41
12                       142                    240.00
```

```
2
13                    280996                  119057.01
17
14                   7364059                 2496888.88
274
15                      8917                   30184.92
19
16                      1830                    9567.92
4
17                      1205                     180.00
6
18                      2179                   13339.00
4
19                     12400                    1100.20
6
20                      1778                    2364.26
7
21                      1130                       0.00
1
22                     40151                   44133.34
18
23                     78912                   42138.02
57
24                    629383                  194250.10
35
25                       820                       0.00
2
26                   7474413                 2915334.36
290
27                     10129                   47138.58
5
28                     31654                       0.00
326
29                      4900                    2748.16
1
30                         2                       0.00
1
31                    270346                       0.00
376
32                   7775162                  966543.98
86
33                     39960                   28778.76
1
34                      1678                    2494.72
3
35                       216                    4059.85
1
36                     11000                   37744.27
1
```

	Average Line Item Quantity per Delivery	Average Freight Cost per Delivery (USD)
0	2426.000000	0.000000
1	2609.888525	5861.244459
2	1242.500000	166.665000
3	415.000000	1731.890000
4	769.714286	3544.291429
5	5373.416667	1282.532500
6	16500.000000	3908.560000
7	28299.524064	7773.812781
8	291.000000	
9	3007.833333	
10	38177.863946	
11	19290.024390	
12	71.000000	
13	16529.176471	
14	26876.127737	
15	469.315789	
16	457.500000	
17	200.833333	
18	544.750000	
19	2066.666667	
20	254.000000	
21	1130.000000	
22	2230.611111	
23	1384.421053	
24	17982.371429	
25	410.000000	
26	25773.837931	
27	2025.800000	
28	97.098160	
29	4900.000000	
30	2.000000	
31	719.005319	
32	90408.860465	
33	39960.000000	
34	559.333333	
35	216.000000	
36	11000.000000	

```
8                                  600.290000
9                                  667.957917
10                               10703.812517
11                                4339.536098
12                                 120.000000
13                                7003.353529
14                                9112.733139
15                                1588.680000
16                                2391.980000
17                                  30.000000
18                                3334.750000
19                                 183.366667
20                                 337.751429
21                                   0.000000
22                                2451.852222
23                                 739.263509
24                                5550.002857
25                                   0.000000
26                               10052.877103
27                                9427.716000
28                                   0.000000
29                                2748.160000
30                                   0.000000
31                                   0.000000
32                               11238.883488
33                               28778.760000
34                                 831.573333
35                                4059.850000
36                               37744.270000

C:\Users\haris\AppData\Local\Temp\ipykernel_18268\4242267466.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  vendor_performance_df['Freight Cost (USD)'] =
pd.to_numeric(vendor_performance_df['Freight Cost (USD)'],
errors='coerce')
```

The output is given as a CSV file with the vendor performance summary. It can be further Analysed to Stream line vendor operations.

```
# Calculating duration for sending POs to vendors
df['PO_to_Vendor_Duration'] = df['PO Sent to Vendor Date'] - df['PQ
First Sent to Client Date']
```

```
# Calculating duration for delivery to clients
df['Delivery_to_Client_Duration'] = df['Delivered to Client Date'] -
df['Scheduled Delivery Date']

# Displaying the processed DataFrame
print(df[['PO Sent to Vendor Date', 'PQ First Sent to Client Date',
'PO_to_Vendor_Duration',
          'Scheduled Delivery Date', 'Delivered to Client Date',
'Delivery_to_Client_Duration']])

print("Mean PO to Vendor Duration:",
df['PO_to_Vendor_Duration'].mean())
print("Mean Delivery to Client Duration:",
df['Delivery_to_Client_Duration'].mean())
```

```
     PO Sent to Vendor Date PQ First Sent to Client Date  \
3055             2009-06-15                   2009-05-29
4754             2009-07-21                   2009-07-05
2806             2009-07-21                   2009-07-05
3894             2009-07-21                   2009-07-05
3422             2009-07-21                   2009-07-05
...                     ...                          ...
4005             2015-04-08                   2015-07-07
6575             2015-04-08                   2015-07-07
3810             2015-04-08                   2015-07-07
5846             2015-04-03                   2015-02-13
3749             2015-04-03                   2015-02-13

     PO_to_Vendor_Duration Scheduled Delivery Date Delivered to Client
Date  \
3055               17 days              2009-08-05                2009-
08-05
4754               16 days              2009-08-05                2009-
08-05
2806               16 days              2009-08-05                2009-
08-05
3894               16 days              2009-08-05                2009-
08-05
3422               16 days              2009-08-05                2009-
08-05
...                    ...                     ...
...
4005              -90 days              2015-09-02                2015-
09-02
6575              -90 days              2015-08-31                2015-
08-31
3810              -90 days              2015-09-02                2015-
09-02
5846               49 days              2015-09-14                2015-
09-14
```

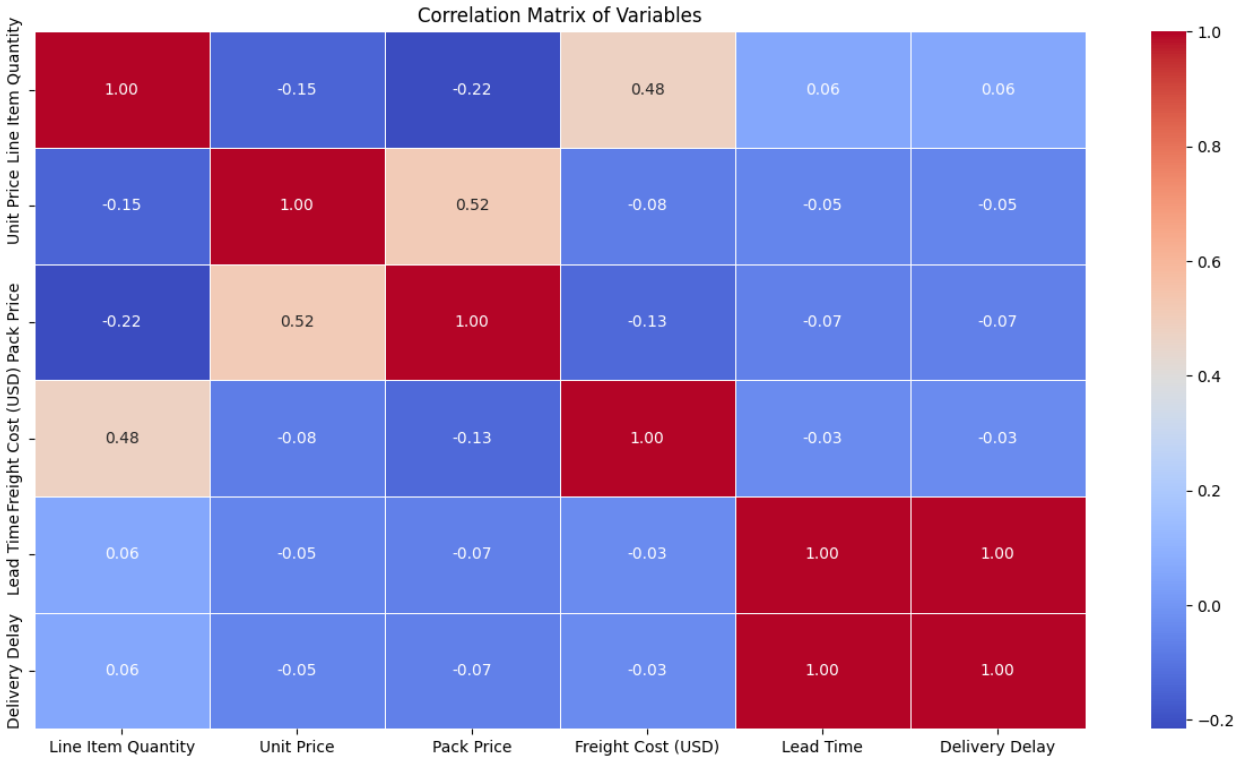
```
3749                49 days               2015-09-14                2015-
09-14

     Delivery_to_Client_Duration
3055                      0 days
4754                      0 days
2806                      0 days
3894                      0 days
3422                      0 days
...                          ...
4005                      0 days
6575                      0 days
3810                      0 days
5846                      0 days
3749                      0 days

[2673 rows x 6 columns]
Mean PO to Vendor Duration: 40 days 06:53:44.242424242
Mean Delivery to Client Duration: 1 days 10:37:18.383838383
```

For most entries, the duration between sending the PO to the vendor and sending the PQ to the client is 16 or 17 days. The delivery to the client occurred on the scheduled delivery date for most entries, resulting in a delivery duration of 0 days. There are also instances where the delivery occurred before the scheduled delivery date (e.g., negative values in "PO_to_Vendor_Duration"), indicating early delivery.

7.Additional Correlation and Cluster Analysis.

Correlation Analysis:

```
relevant_columns = ['Line Item Quantity', 'Unit Price', 'Pack Price',
'Freight Cost (USD)', 'Lead Time', 'Delivery Delay']

# Convert non-numeric values to NaN
df[relevant_columns] = df[relevant_columns].apply(pd.to_numeric,
errors='coerce')

# Creating a correlation matrix
correlation_matrix = df[relevant_columns].corr()

# Plotting the correlation matrix as a heatmap
plt.figure(figsize=(15, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
fmt=".2f", linewidths=.5)
plt.title('Correlation Matrix of Variables')
plt.show()
```



Intepretation: There is a weak positive linear relationship between line item quantity and unit price.

There is a moderate positive linear relationship between unit price and pack price.

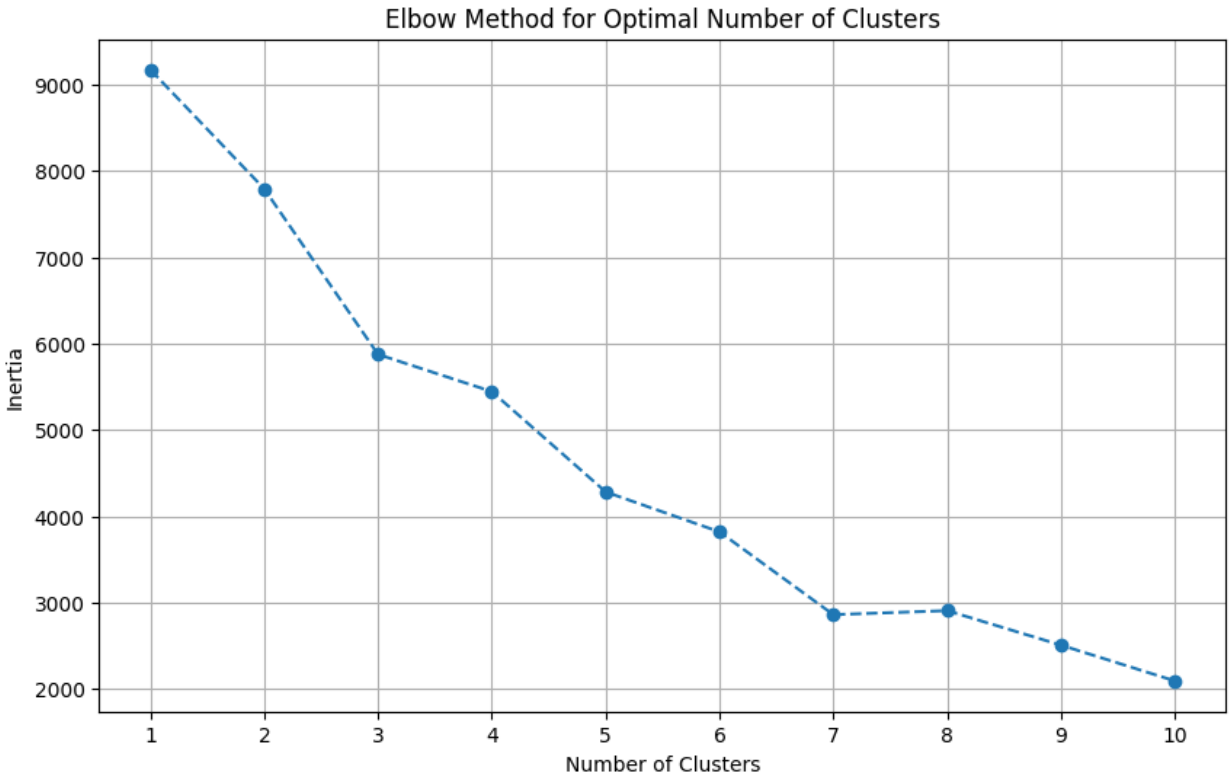
There is a weak negative linear relationship between line item quantity and freight cost.

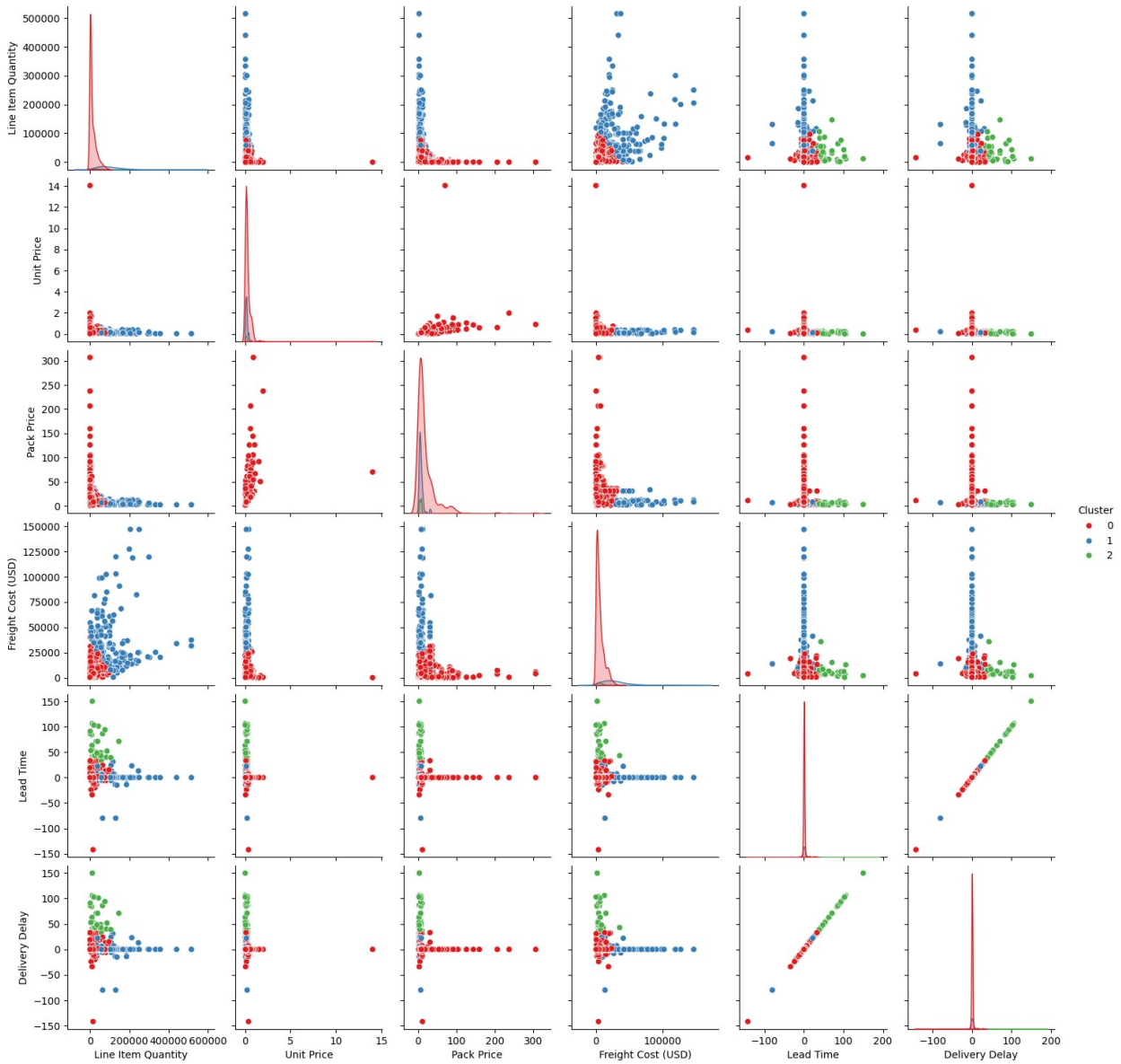
There is a very strong negative linear relationship between lead time and delivery delay.

Cluster Analysis

```
relevant_columns = ['Line Item Quantity', 'Unit Price', 'Pack Price',
'Freight Cost (USD)', 'Lead Time', 'Delivery Delay']

# Dropping NaN values if any
df_cleaned = df[relevant_columns].dropna()

#The selected data is normalized to ensure that each feature
contributes equally to the distance computations during clustering.
normalized_data = (df_cleaned - df_cleaned.mean()) / df_cleaned.std()

# The Elbow method is used to determine the optimal number of
clusters. The inertia values (sum of squared distances of samples to
their closest cluster center)
#are calculated for different values of k (number of clusters) and
plotted to identify the "elbow point," which indicates the optimal
number of clusters.
inertia_values = []
```

```
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(normalized_data)
    inertia_values.append(kmeans.inertia_)

# Plotting the Elbow curve
plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), inertia_values, marker='o', linestyle='--')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.title('Elbow Method for Optimal Number of Clusters')
plt.xticks(range(1, 11))
plt.grid(True)
plt.show()

#Based on the Elbow curve, a specific value of k (in this example,
k=3) is chosen, and K-means clustering is performed using that value.
k = 3
kmeans = KMeans(n_clusters=k, random_state=42)
kmeans.fit(normalized_data)

# Adding cluster labels to the DataFrame
df_cleaned['Cluster'] = kmeans.labels_

# Pairplot is used to visualize the clusters in the selected feature
space. Each pair of features is plotted against each other, with
different colors representing different clusters.
sns.pairplot(df_cleaned, hue='Cluster', palette='Set1',
diag_kind='kde')
plt.show()
```


Elbow Method for Optimal Number of Clusters
Inertia
Number of Clusters
9000
8000
7000
6000
5000
4000
3000
2000
1
2
3
4
5
6
7
8
9
10



The pairplot visualization allows us to visually inspect the clusters and understand how data points are distributed across different features. This can provide insights into the structure of the data and any underlying patterns or relationships between variables.The clustering results can be used for various purposes such as customer segmentation, anomaly detection, or targeted marketing strategies. By understanding the distinct groups present in the data, businesses can tailor their approaches and strategies accordingly to better serve their customers or optimize their operations.

Findings and Recommendations

Duration Distribution:

The right-skewed distribution suggests that most procurement and delivery durations are shorter, indicating efficient processing and delivery for the majority of orders.

The longer tail on the right side indicates occasional instances of longer durations, which could be due to various factors such as delays in vendor response, shipping issues, or logistical challenges.

Cumulative Quantity of Items Shipped Over Time:

The upward trend in the cumulative quantity of items shipped over time indicates overall growth or increasing demand for the products.

Fluctuations in the trend, such as the significant increase in shipments in 2012, may reflect seasonal variations, changes in market demand, or specific business initiative

Lead Time and On-time Deliveries

The average delivery delay is approximately 1.44 days. This suggests that, on average, deliveries are completed within a short timeframe of the scheduled delivery date. Overall, the analysis indicates that the organization maintains a good on-time delivery rate, with most deliveries being fulfilled promptly. However, there is some variability in lead times, highlighting the importance of continuous monitoring and process improvement initiatives to ensure consistent and reliable delivery performance

Insurance Coverage and Shipment Mode:

Overall, the analysis suggests a strategic approach to shipping, with a predominant focus on air transportation to meet delivery deadlines and ensure the efficient movement of goods. However, the organization also leverages other modes such as truck, air charter, and ocean, indicating a diversified shipping strategy tailored to different logistical requirements and priorities.

Consider the environmental impact of different shipment modes and prioritize sustainable transportation options where feasible. This could involve optimizing routes to minimize fuel consumption, adopting eco-friendly packaging materials, or exploring carbon offset programs for shipping emissions.

Continue to prioritize insurance coverage for shipments to mitigate risks associated with transit and delivery. However, ensure that insurance policies are cost-effective and tailored to the specific needs and value of goods being shipped.

Procurement Process:

Implement standardized INCO terms across procurement processes to ensure clarity and consistency in international trade transactions. This will help mitigate risks, streamline logistics, and enhance communication between parties involved.

Foster stronger partnerships with key stakeholders, particularly top brands like Generic, to improve collaboration and communication throughout the procurement process. This can involve regular meetings, joint planning sessions, and shared performance metrics to align goals and objectives.

Manufacturing Site and country wise Demand:

To improve compliance levels at manufacturing sites with low first line designation, it's recommended to conduct thorough audits, provide targeted training programs, implement

robust quality management systems, and foster a culture of compliance and accountability across all levels of the organization. Additionally, establishing clear guidelines and protocols for regulatory compliance and regularly monitoring performance metrics can help drive continuous improvement in compliance standards.

Recognizing South Africa's substantial demand in the dataset is pivotal for effective supply chain management. It necessitates prioritizing and allocating additional resources to meet the heightened demand efficiently in this region. August emerges as the month with the highest number of orders, indicating potentially heightened demand or specific market dynamics during that period. Conversely, July and September exhibit moderate order volumes. Understanding these patterns enables supply chain managers to anticipate demand fluctuations, optimize resource allocation, and streamline operational efficiency to meet varying demand levels across different months effectively.

Vendor performance

The average duration from purchase order (PO) to delivery, standing at approximately 114.76 days, signifies the typical timeline from initiating a purchase order to receiving the goods.

Conversely, the average duration from the vendor to delivery is notably shorter, approximately 1.44 days, indicating a swift turnaround from the vendor dispatching the goods to their delivery to the client. Aurobindo Pharma Limited emerges as the entity with the highest line quantity, underscoring its significance within the procurement process.

Thorough Vendor Analysis can be conducted trough the given csv output file.

Correlation Performance:

Line Item Quantity vs. Unit Price: The weak positive linear relationship suggests that, on average, as the quantity of items ordered increases, there is a slight tendency for the unit price to increase as well. This could indicate potential volume discounts or economies of scale in pricing.

Unit Price vs. Pack Price: The moderate positive linear relationship between unit price and pack price indicates that higher unit prices tend to correspond to higher pack prices. This relationship could be attributed to factors such as product quality or packaging specifications, where higher-priced items may come in larger or more specialized packaging.

Line Item Quantity vs. Freight Cost: The weak negative linear relationship suggests that, on average, as the quantity of items ordered increases, there is a slight tendency for the freight cost per item to decrease. This could be due to economies of scale in shipping or bulk shipping discounts.

Lead Time vs. Delivery Delay: The very strong negative linear relationship between lead time and delivery delay indicates that shorter lead times are associated with fewer delivery delays. This highlights the critical importance of efficient lead time management in ensuring timely deliveries and minimizing delays in the supply chain.