About the Dataset

In this data science project, you will build a machine learning system that will be able to predict the cost of the shipment or package by using machine learning algorithms. This project will be very useful for logistics companies, where on a day-to-day basis a lot of couriers, packages, or goods are transported via different modes of transport. The main concern with these logistics companies is trying to deliver these goods in an efficient and cost-efficient way possible, so the pricing of the shipment is tricky and involves a lot of variables to consider while the pricing of the shipment. There might be scenarios where the shipment might be delayed due to some external reasons, leading to a loss for the company and a delay in delivery of the shipment. So logistics companies need to use dynamic pricing based on several factors and variables to price the shipment in such a way that there are no losses to the company and the price of the shipment is as less as possible so that customers can use their services more due to effective pricing rates.

In [ ]:

```

```

Problem Statement:

The market for supply chain analytics is expected to develop at a CAGR of 17.3 percent from 2019 to 2024, more than doubling in size. This data demonstrates how supply chain organizations are understanding the advantages of being able to predict what will happen in the future with a decent degree of certainty. Supply chain leaders may use this data to address supply chain difficulties, cut costs, and enhance service levels all at the same time. The main goal is to predict the supply chain shipment pricing based on the available factors in the dataset. Approach: The classical machine learning tasks like Data Exploration, Data Cleaning, Feature Engineering, Model Building and Model Testing. Try out different machine learning algorithms that’s best fit for the above case.

Import libraries

In [ ]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Import dataset

In [ ]:

```
!git clone https://github.com/alinegorischf/Shipment-Price-Prediction
```

```
Cloning into 'Shipment-Price-Prediction'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 47 (delta 18), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (47/47), 2.14 MiB | 858.00 KiB/s, done.
Resolving deltas: 100% (18/18), done.
```

In [ ]:

In [ ]:

```
# Display all the dataset
pd.pandas.set_option('display.max_columns', None)
```

In [ ]:

```
data = '/content/Shipment-Price-Prediction/dataset/SCMS_Delivery_History_Da
read = pd.read_csv(data)
read
```

Out[5]:

	ID	Project Code	PQ #	PO / SO #	ASN/DN #	Country	Managed By	Fulfill Via	Vendor INCO Term	Shipm Mo
0	1	100-CI-T01	Pre-PQ Process	SCMS-4	ASN-8	Côte d'Ivoire	PMO - US	Direct Drop	EXW	
1	3	108-VN-T01	Pre-PQ Process	SCMS-13	ASN-85	Vietnam	PMO - US	Direct Drop	EXW	
2	4	100-CI-T01	Pre-PQ Process	SCMS-20	ASN-14	Côte d'Ivoire	PMO - US	Direct Drop	FCA	
3	15	108-VN-T01	Pre-PQ Process	SCMS-78	ASN-50	Vietnam	PMO - US	Direct Drop	EXW	
4	16	108-VN-T01	Pre-PQ Process	SCMS-81	ASN-55	Vietnam	PMO - US	Direct Drop	EXW	
...	...	...	...	...	...	...	...	...	...	
10319	86818	103-ZW-T30	FPQ-15197	SO-50020	DN-4307	Zimbabwe	PMO - US	From RDC	N/A - From RDC	Tr
10320	86819	104-CI-T30	FPQ-15259	SO-50102	DN-4313	Côte d'Ivoire	PMO - US	From RDC	N/A - From RDC	Tr
10321	86821	110-ZM-T30	FPQ-14784	SO-49600	DN-4316	Zambia	PMO - US	From RDC	N/A - From RDC	Tr
10322	86822	200-ZW-T30	FPQ-16523	SO-51680	DN-4334	Zimbabwe	PMO - US	From RDC	N/A - From RDC	Tr
10323	86823	103-ZW-T30	FPQ-15197	SO-50022	DN-4336	Zimbabwe	PMO - US	From RDC	N/A - From RDC	Tr

10324 rows × 33 columns

Check the data

```
In [ ]: df = pd.read_csv('/content/Shipment-Price-Prediction/dataset/SCMS_Delivery_
        df.head()
```

Out[6]:

	ID	Project Code	PQ #	PO / SO #	ASN/DN #	Country	Managed By	Fulfill Via	Vendor INCO Term	Shipment Mode	Fi Sent Cli D
0	1	100-CI-T01	Pre-PQ Process	SCMS-4	ASN-8	Côte d'Ivoire	PMO - US	Direct Drop	EXW	Air	Pre-Proc
1	3	108-VN-T01	Pre-PQ Process	SCMS-13	ASN-85	Vietnam	PMO - US	Direct Drop	EXW	Air	Pre-Proc
2	4	100-CI-T01	Pre-PQ Process	SCMS-20	ASN-14	Côte d'Ivoire	PMO - US	Direct Drop	FCA	Air	Pre-Proc
3	15	108-VN-T01	Pre-PQ Process	SCMS-78	ASN-50	Vietnam	PMO - US	Direct Drop	EXW	Air	Pre-Proc
4	16	108-VN-T01	Pre-PQ Process	SCMS-81	ASN-55	Vietnam	PMO - US	Direct Drop	EXW	Air	Pre-Proc

Data Cleaning

In [ ]:

```
# Verify the data variables and the data type an if there is null data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10324 entries, 0 to 10323
Data columns (total 33 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   ID                            10324 non-null  int64  
 1   Project Code                  10324 non-null  object 
 2   PQ #                          10324 non-null  object 
 3   PO / SO #                     10324 non-null  object 
 4   ASN/DN #                      10324 non-null  object 
 5   Country                       10324 non-null  object 
 6   Managed By                    10324 non-null  object 
 7   Fulfill Via                   10324 non-null  object 
 8   Vendor INCO Term              10324 non-null  object 
 9   Shipment Mode                 9964 non-null   object 
 10  PQ First Sent to Client Date  10324 non-null  object 
 11  PO Sent to Vendor Date        10324 non-null  object 
 12  Scheduled Delivery Date       10324 non-null  object 
 13  Delivered to Client Date      10324 non-null  object 
 14  Delivery Recorded Date        10324 non-null  object 
 15  Product Group                 10324 non-null  object 
 16  Sub Classification            10324 non-null  object 
 17  Vendor                        10324 non-null  object 
 18  Item Description              10324 non-null  object 
 19  Molecule/Test Type            10324 non-null  object 
 20  Brand                         10324 non-null  object 
 21  Dosage                        8588 non-null   object 
 22  Dosage Form                   10324 non-null  object 
 23  Unit of Measure (Per Pack)    10324 non-null  int64  
 24  Line Item Quantity            10324 non-null  int64  
 25  Line Item Value               10324 non-null  float64
 26  Pack Price                    10324 non-null  float64
 27  Unit Price                    10324 non-null  float64
 28  Manufacturing Site            10324 non-null  object 
 29  First Line Designation        10324 non-null  object 
 30  Weight (Kilograms)            10324 non-null  object 
 31  Freight Cost (USD)            10324 non-null  object 
 32  Line Item Insurance (USD)     10037 non-null  float64
dtypes: float64(4), int64(3), object(26)
memory usage: 2.6+ MB
```

- it is indicated the there are total 33 columns, 4 are float columns , 3 are integer columns and 26 are object columns

In [ ]:

```
# check the columns of dataset
df.columns
```

Out[8]:

```
Index(['ID', 'Project Code', 'PQ #', 'PO / SO #', 'ASN/DN #', 'Country',
       'Managed By', 'Fulfill Via', 'Vendor INCO Term', 'Shipment Mode',
       'PQ First Sent to Client Date', 'PO Sent to Vendor Date',
       'Scheduled Delivery Date', 'Delivered to Client Date',
       'Delivery Recorded Date', 'Product Group', 'Sub Classification',
       'Vendor', 'Item Description', 'Molecule/Test Type', 'Brand', 'Dosag
e',
       'Dosage Form', 'Unit of Measure (Per Pack)', 'Line Item Quantity',
       'Line Item Value', 'Pack Price', 'Unit Price', 'Manufacturing Sit
e',
       'First Line Designation', 'Weight (Kilograms)', 'Freight Cost (US
D)',
       'Line Item Insurance (USD)'],
      dtype='object')
```

In [ ]:

In [ ]:

```
# check the shape of datasets
df.shape
```

Out[9]:

```
(10324, 33)
```

```
In [ ]: # check the missing value

        df.isnull().sum()
```

```
Out[10]: ID                                 0
Project Code                       0
PQ #                               0
PO / SO #                          0
ASN/DN #                           0
Country                            0
Managed By                         0
Fulfill Via                        0
Vendor INCO Term                   0
Shipment Mode                    360
PQ First Sent to Client Date       0
PO Sent to Vendor Date             0
Scheduled Delivery Date            0
Delivered to Client Date           0
Delivery Recorded Date             0
Product Group                      0
Sub Classification                 0
Vendor                             0
Item Description                   0
Molecule/Test Type                 0
Brand                              0
Dosage                          1736
Dosage Form                        0
Unit of Measure (Per Pack)         0
Line Item Quantity                 0
Line Item Value                    0
Pack Price                         0
Unit Price                         0
Manufacturing Site                 0
First Line Designation             0
Weight (Kilograms)                 0
Freight Cost (USD)                 0
Line Item Insurance (USD)        287
dtype: int64
```

- it is indicated the three columns are missing value

In [ ]:

```
# check the missing value of percentage

(df.isnull().mean()*100).sort_values(ascending=False)
```

Out[11]:

```
Dosage                          16.815188
Shipment Mode                    3.487021
Line Item Insurance (USD)        2.779930
Molecule/Test Type               0.000000
Brand                            0.000000
Dosage Form                      0.000000
Unit of Measure (Per Pack)       0.000000
Line Item Quantity               0.000000
Line Item Value                  0.000000
Vendor                           0.000000
Pack Price                       0.000000
Unit Price                       0.000000
Manufacturing Site               0.000000
First Line Designation           0.000000
Weight (Kilograms)               0.000000
Freight Cost (USD)               0.000000
Item Description                 0.000000
ID                               0.000000
Project Code                     0.000000
Product Group                    0.000000
Delivery Recorded Date           0.000000
Delivered to Client Date         0.000000
Scheduled Delivery Date          0.000000
PO Sent to Vendor Date           0.000000
PQ First Sent to Client Date     0.000000
Vendor INCO Term                 0.000000
Fulfill Via                      0.000000
Managed By                       0.000000
Country                          0.000000
ASN/DN #                         0.000000
PO / SO #                        0.000000
PQ #                             0.000000
Sub Classification               0.000000
dtype: float64
```

In [ ]:

```
# check the total missing value
df.isnull().sum().sum()
```

Out[12]:

```
2383
```

In [ ]:

```
# drop the columns
df = df.drop('ID',axis=1)
```

In [ ]:

```
# check the unique value
df.nunique()
```

Out[14]:

```
Project Code                    142
PQ #                           1237
PO / SO #                      6233
ASN/DN #                       7030
Country                          43
Managed By                        4
Fulfill Via                       2
Vendor INCO Term                  8
Shipment Mode                     4
PQ First Sent to Client Date    765
PO Sent to Vendor Date          897
Scheduled Delivery Date        2006
Delivered to Client Date       2093
Delivery Recorded Date         2042
Product Group                     5
Sub Classification                6
Vendor                           73
Item Description                184
Molecule/Test Type               86
Brand                            48
Dosage                           54
Dosage Form                      17
Unit of Measure (Per Pack)       31
Line Item Quantity             5065
Line Item Value                8741
Pack Price                     1175
Unit Price                      183
Manufacturing Site               88
First Line Designation            2
Weight (Kilograms)             4688
Freight Cost (USD)             6733
Line Item Insurance (USD)      6722
dtype: int64
```

In [ ]:

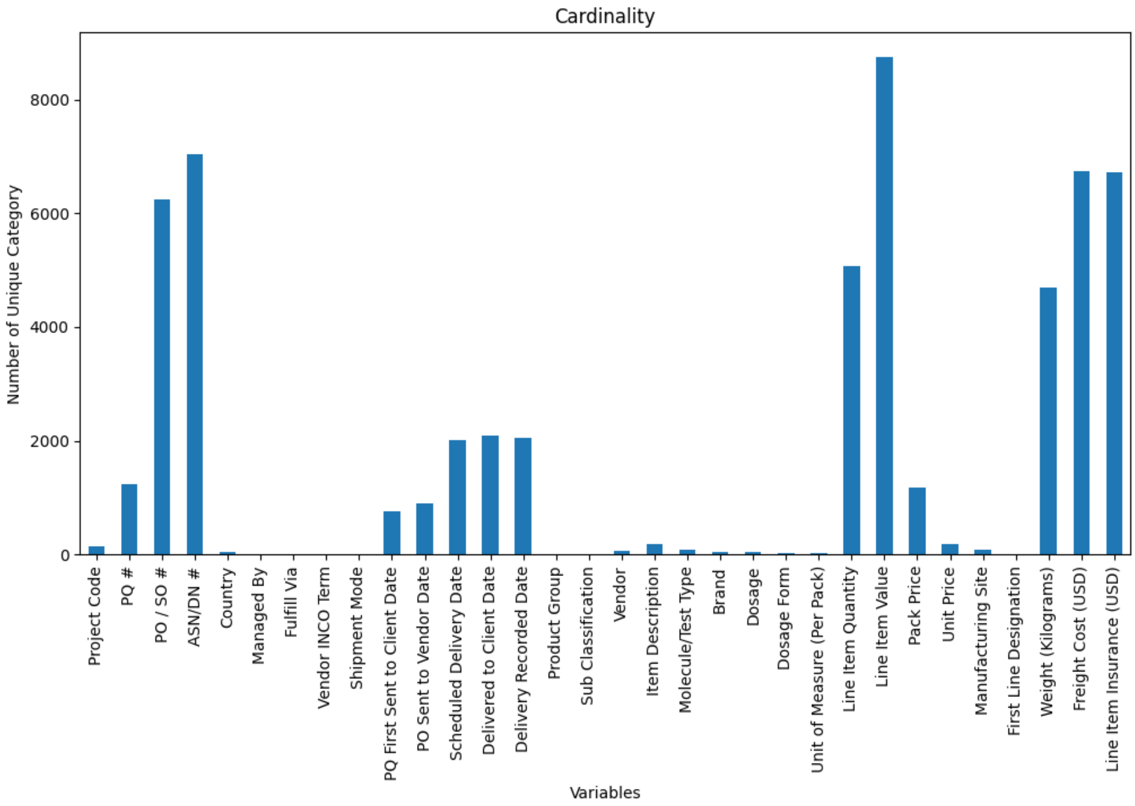
```
# check the unique value in ascending order
df.nunique().sort_values(ascending=False)
```

Out[15]:

```
Line Item Value                 8741
ASN/DN #                        7030
Freight Cost (USD)              6733
Line Item Insurance (USD)       6722
PO / SO #                       6233
Line Item Quantity              5065
Weight (Kilograms)              4688
Delivered to Client Date        2093
Delivery Recorded Date          2042
Scheduled Delivery Date         2006
PQ #                            1237
Pack Price                      1175
PO Sent to Vendor Date           897
PQ First Sent to Client Date     765
Item Description                 184
Unit Price                       183
Project Code                     142
Manufacturing Site                88
Molecule/Test Type                86
Vendor                            73
Dosage                            54
Brand                             48
Country                           43
Unit of Measure (Per Pack)        31
Dosage Form                       17
Vendor INCO Term                   8
Sub Classification                 6
Product Group                      5
Shipment Mode                      4
Managed By                         4
Fulfill Via                        2
First Line Designation             2
dtype: int64
```

```
In [ ]: # show the unique value in graph
        df.nunique().plot.bar(figsize=(12,6))
        plt.ylabel('Number of Unique Category')
        plt.xlabel('Variables')
        plt.title('Cardinality')

Out[16]: Text(0.5, 1.0, 'Cardinality')
```



```
In [ ]: # check the duplicated values
        df.duplicated().sum()

Out[17]: 4
```

In [ ]:
```
df[df.duplicated]
```

Out[18]:

	Project Code	PQ #	PO / SO #	ASN/DN #	Country	Managed By	Fulfill Via	Vendor INCO Term	Shipment Mode	F Ser Cl D
1520	125-HT-T01	Pre-PQ Process	SO-1291	DN-686	Haiti	PMO - US	From RDC	N/A - From RDC	Air	Pre Proc
2135	100-ZW-T01	Pre-PQ Process	SO-710	DN-488	Zimbabwe	PMO - US	From RDC	N/A - From RDC	Air	Pre Proc
2577	100-ZW-T01	Pre-PQ Process	SO-716	DN-770	Zimbabwe	PMO - US	From RDC	N/A - From RDC	Air	Pre Proc
5781	105-SS-T30	FPQ-12623	SCMS-200920	ASN-21751	South Sudan	PMO - US	Direct Drop	EXW	Air	7/1

- it is indicated the duplicated value

In [ ]:
```
# drop the duplicated values
df = df.drop_duplicates()
```

In [ ]:
```
# check the shape of dataset after drop the duplicated value

df.shape
```

Out[20]:
```
(10320, 32)
```

```
In [ ]: # Getting the count of each category from data
        for feature in df.columns:
            print(df[feature].value_counts())
```

```
116-ZA-T30    768
104-CI-T30    729
151-NG-T30    628
114-UG-T30    596
108-VN-T30    522
             ...
100-SN-T01      1
201-UG-T30      1
100-GN-T30      1
A02-SN-T50      1
104-SZ-T30      1
Name: Project Code, Length: 142, dtype: int64
Pre-PQ Process    2678
FPQ-14942          205
FPQ-12522          154
FPQ-13973          110
FPQ-4537            98
                  ...
FPQ-12933            1
```

```
In [ ]: # print the unique values in each column name
        for feature in df.columns:

            print(f"Unique values in  '{feature}' column: {df[feature].unique()}")
```

```
Unique values in  'Project Code' column: ['100-CI-T01' '108-VN-T01' '11
2-NG-T01' '110-ZM-T01' '109-TZ-T01'
 '102-NG-T01' '107-RW-T01' '106-HT-T01' '113-ZW-T01' '104-CI-T01'
 '100-HT-T01' '117-ET-T01' '116-ZA-T01' '123-NG-T01' '125-HT-T01'
 '102-GY-T01' '119-NA-T01' '131-NG-T01' '102-BW-T01' '111-MZ-T01'
 '144-BW-T01' '102-KE-T01' '133-NG-T01' '100-KZ-T01' '141-NA-T01'
 '114-UG-T01' '105-GY-T01' '139-NA-T01' '129-KG-T01' '100-SN-T01'
 '128-BJ-T01' '102-LS-T01' '130-NG-T01' '100-BW-T01' '100-ZW-T01'
 '100-PK-T01' '126-NG-T01' '151-NG-T01' '100-SZ-T01' '100-GH-T01'
 '120-AO-T01' '132-NG-T01' '153-NG-T01' '100-LB-T01' '151-NG-T30'
 '127-KE-T01' '510-KE-T01' '100-SL-T01' '136-RW-T01' '102-KE-T30'
 '108-VN-T30' '110-ZM-T30' '106-HT-T30' '105-SS-T30' '111-MZ-T30'
 '102-BI-T30' '122-HT-T30' '161-ZA-T30' '116-ZA-T30' '133-NG-T30'
 '103-DO-T30' '104-CI-T30' '107-RW-T30' '103-MW-T30' '101-CD-T30'
 '102-SZ-T30' '114-UG-T30' '105-DO-T30' '113-ZW-T30' '103-CM-T30'
 '109-TZ-T30' '800-CM-T30' '100-BJ-T30' '117-ET-T30' '900-TZ-T30'
 '112-NG-T30' '110-PK-T30' '102-SS-T30' '105-GY-T30' '102-SD-T30'
 '102-ML-T30' 'A01-CM-T50' '901-CM-T30' '123-NG-T30' '103-KE-T30'
 '152-HT-T30' '901-NA-T30' '103-ZW-T30' '105-GH-T30' '202-GT-T30'
```

- it is indicated the Six Columns are some numeric and string value. Columns name

1. PQ First Sent to Client Date
2. PO Sent to Vendor Date
3. Item Description
4. Dosage Columns are null value
5. Weight (Kilograms)
6. Freight Cost (USD) We need a remove unique value

In [ ]:

```

```

it is indicated the some columns are specifice charcter value. we need a clean it and convert the date time formate.

In [ ]:

```
df['PQ First Sent to Client Date']
```

Out[23]:

```
0         Pre-PQ Process
1         Pre-PQ Process
2         Pre-PQ Process
3         Pre-PQ Process
4         Pre-PQ Process
               ...
10319           10/16/14
10320           10/24/14
10321            8/12/14
10322             7/1/15
10323           10/16/14
Name: PQ First Sent to Client Date, Length: 10320, dtype: object
```

In [ ]:

```
df['PO Sent to Vendor Date']
```

Out[24]:

```
0         Date Not Captured
1         Date Not Captured
2         Date Not Captured
3         Date Not Captured
4         Date Not Captured
                ...
10319        N/A - From RDC
10320        N/A - From RDC
10321        N/A - From RDC
10322        N/A - From RDC
10323        N/A - From RDC
Name: PO Sent to Vendor Date, Length: 10320, dtype: object
```

In [ ]:

```
# converting dates into datetimes formate

date_time = ['PQ First Sent to Client Date','PO Sent to Vendor Date','Sched

for columns in date_time:
  df[columns] = pd.to_datetime(df[columns],errors='coerce')
```

- We are convert the 5 columns are date and time formate

In [ ]:

```
# Replace NAN with mode in Dosage column
df['Dosage']
```

Out[26]:

```
0                  NaN
1              10mg/ml
2                  NaN
3                150mg
4                 30mg
              ...
10319       30/50/60mg
10320        150/300mg
10321    600/300/300mg
10322        150/300mg
10323          30/60mg
Name: Dosage, Length: 10320, dtype: object
```

In [ ]:

```
df['Dosage'] = df['Dosage'].fillna(df['Dosage'].mode()[0])
```

In [ ]:

```
df['Weight (Kilograms)']
```

Out[28]:

```
0                               13
1                              358
2                              171
3                             1855
4                             7590
                     ...
10319       See DN-4307 (ID#:83920)
10320       See DN-4313 (ID#:83921)
10321    Weight Captured Separately
10322                         1392
10323    Weight Captured Separately
Name: Weight (Kilograms), Length: 10320, dtype: object
```

In [ ]:

```
 # Tackling Weight (Kilograms) missing values and convert the numeric data
df['Weight (Kilograms)'] = df['Weight (Kilograms)'].replace('Weight Capture

df['Weight (Kilograms)'] = pd.to_numeric(df['Weight (Kilograms)'], errors =

# filling the missing value with mean
df['Weight (Kilograms)'] = df['Weight (Kilograms)'].fillna(df['Weight (Kilo
```

In [ ]:

```
df['Freight Cost (USD)']
```

Out[30]:

```
0                                  780.34
1                                  4521.5
2                                 1653.78
3                                16007.06
4                                45450.08
                          ...
10319             See DN-4307 (ID#:83920)
10320             See DN-4313 (ID#:83921)
10321    Freight Included in Commodity Cost
10322    Freight Included in Commodity Cost
10323    Freight Included in Commodity Cost
Name: Freight Cost (USD), Length: 10320, dtype: object
```

In [ ]:

```
df['Freight Cost (USD)'] = pd.to_numeric(df['Freight Cost (USD)'], errors =

# filling the missing value with the help of mean()
df['Freight Cost (USD)'] = df['Freight Cost (USD)'].fillna(df['Freight Cost
```

In [ ]:

```
df['Line Item Insurance (USD)']
```

Out[32]:

```
0              NaN
1              NaN
2              NaN
3              NaN
4              NaN
             ...
10319       705.79
10320       161.71
10321      5284.04
10322       134.03
10323        85.82
Name: Line Item Insurance (USD), Length: 10320, dtype: float64
```

In [ ]:

```
# remove rows with NaN values
df.dropna(subset=['Line Item Insurance (USD)'], inplace=True)

# convert column to float type
df['Line Item Insurance (USD)'] = df['Line Item Insurance (USD)'].astype(fl

print(df)
```

```
      Project Code            PQ # PO / SO # ASN/DN #        Country  \
16      102-NG-T01  Pre-PQ Process  SCMS-354  ASN-608        Nigeria
19      102-NG-T01  Pre-PQ Process  SCMS-592  ASN-485        Nigeria
21      104-CI-T01  Pre-PQ Process  SCMS-698  ASN-727  Côte d'Ivoire
22      108-VN-T01  Pre-PQ Process  SCMS-753  ASN-781        Vietnam
23      108-VN-T01  Pre-PQ Process  SCMS-759  ASN-632        Vietnam
...            ...             ...       ...      ...            ...
10319   103-ZW-T30       FPQ-15197  SO-50020  DN-4307       Zimbabwe
10320   104-CI-T30       FPQ-15259  SO-50102  DN-4313  Côte d'Ivoire
10321   110-ZM-T30       FPQ-14784  SO-49600  DN-4316         Zambia
10322   200-ZW-T30       FPQ-16523  SO-51680  DN-4334       Zimbabwe
10323   103-ZW-T30       FPQ-15197  SO-50022  DN-4336       Zimbabwe

      Managed By  Fulfill Via Vendor INCO Term Shipment Mode  \
16      PMO - US  Direct Drop              CIP           NaN
19      PMO - US  Direct Drop              EXW           Air
21      PMO - US  Direct Drop              CIP           Air
22      PMO - US  Direct Drop              EXW           Air
23      PMO - US  Direct Drop              FCA           Air
```

Exploratory Data Analysis (EDA)

In [ ]:

```
# it is indicate the data type of columns
df.dtypes
```

Out[34]:

```
Project Code                                object
PQ #                                        object
PO / SO #                                   object
ASN/DN #                                    object
Country                                     object
Managed By                                  object
Fulfill Via                                 object
Vendor INCO Term                            object
Shipment Mode                               object
PQ First Sent to Client Date        datetime64[ns]
PO Sent to Vendor Date              datetime64[ns]
Scheduled Delivery Date             datetime64[ns]
Delivered to Client Date            datetime64[ns]
Delivery Recorded Date              datetime64[ns]
Product Group                               object
Sub Classification                          object
Vendor                                      object
Item Description                            object
Molecule/Test Type                          object
Brand                                       object
Dosage                                      object
Dosage Form                                 object
Unit of Measure (Per Pack)                   int64
Line Item Quantity                           int64
Line Item Value                            float64
Pack Price                                 float64
Unit Price                                 float64
Manufacturing Site                          object
First Line Designation                      object
Weight (Kilograms)                         float64
Freight Cost (USD)                         float64
Line Item Insurance (USD)                  float64
dtype: object
```

In [ ]:

```
# drop the columns

df = df.drop(['PQ #', 'PO / SO #', 'ASN/DN #'], axis = 1)
```

In [ ]:

```
# after cleaning the data we are again analysis data


df.head()
```

Out[36]:

	Project Code	Country	Managed By	Fulfill Via	Vendor INCO Term	Shipment Mode	PQ First Sent to Client Date	PO Sent to Vendor Date	Scheduled Delivery Date	Deliver to Cli Da
16	102-NG-T01	Nigeria	PMO - US	Direct Drop	CIP	NaN	NaT	NaT	2007-05-07	2007-
19	102-NG-T01	Nigeria	PMO - US	Direct Drop	EXW	Air	NaT	2007-05-13	2007-06-19	2007-
21	104-CI-T01	Côte d'Ivoire	PMO - US	Direct Drop	CIP	Air	NaT	2007-07-13	2007-10-02	2007-
22	108-VN-T01	Vietnam	PMO - US	Direct Drop	EXW	Air	NaT	2007-07-04	2007-10-15	2007-
23	108-VN-T01	Vietnam	PMO - US	Direct Drop	FCA	Air	NaT	2007-07-04	2007-08-27	2007-

In [ ]:
```
df.isnull().sum()
```

Out[37]:
```
Project Code                       0
Country                            0
Managed By                         0
Fulfill Via                        0
Vendor INCO Term                   0
Shipment Mode                    254
PQ First Sent to Client Date    2391
PO Sent to Vendor Date          5482
Scheduled Delivery Date            0
Delivered to Client Date           0
Delivery Recorded Date             0
Product Group                      0
Sub Classification                 0
Vendor                             0
Item Description                   0
Molecule/Test Type                 0
Brand                              0
Dosage                             0
Dosage Form                        0
Unit of Measure (Per Pack)         0
Line Item Quantity                 0
Line Item Value                    0
Pack Price                         0
Unit Price                         0
Manufacturing Site                 0
First Line Designation             0
Weight (Kilograms)                 0
Freight Cost (USD)                 0
Line Item Insurance (USD)          0
dtype: int64
```

In [ ]:
```
df.shape
```

Out[38]:
```
(10033, 29)
```

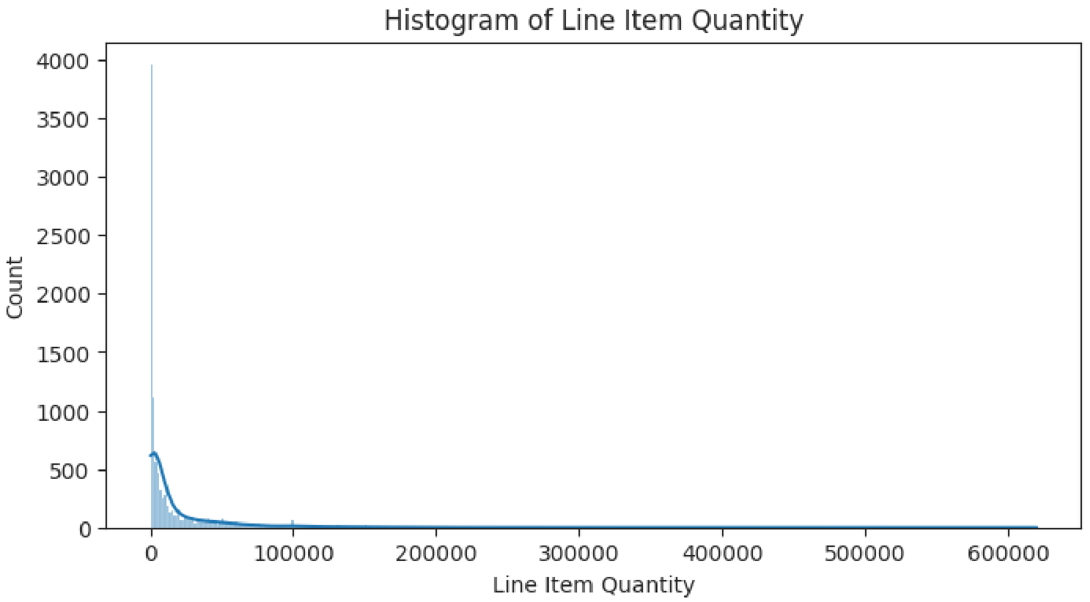
In [ ]:
```
df.columns
```

Out[39]:
```
Index(['Project Code', 'Country', 'Managed By', 'Fulfill Via',
       'Vendor INCO Term', 'Shipment Mode', 'PQ First Sent to Client Dat
e',
       'PO Sent to Vendor Date', 'Scheduled Delivery Date',
       'Delivered to Client Date', 'Delivery Recorded Date', 'Product Grou
p',
       'Sub Classification', 'Vendor', 'Item Description',
       'Molecule/Test Type', 'Brand', 'Dosage', 'Dosage Form',
       'Unit of Measure (Per Pack)', 'Line Item Quantity', 'Line Item Valu
e',
       'Pack Price', 'Unit Price', 'Manufacturing Site',
       'First Line Designation', 'Weight (Kilograms)', 'Freight Cost (US
D)',
       'Line Item Insurance (USD)'],
      dtype='object')
```

#Univariate Analysis

In [ ]:

```
# Univariate Analysis
# Histograms for numerical columns
numerical_cols = ['Line Item Quantity', 'Line Item Value', 'Pack Price', 'W
for col in numerical_cols:
    plt.figure(figsize=(8, 4))
    sns.histplot(df[col], kde=True)
    plt.title(f'Histogram of {col}')
    plt.show()
```



In [ ]:

Segregrate the data in NUmerical and Categorical Columns

In [ ]:

```
num_columns = [feature for feature in df.columns if df[feature].dtypes=='Ob
print(" Numerical columns :", len(num_columns))
print((num_columns))
```

```
 Numerical columns : 0
[]
```

In [ ]:

```
float_columns = [feature for feature in df.columns if df[feature].dtypes=='
print("Number of  coloumns :" , len(float_columns))
print((float_columns))
```

```
Number of  coloumns : 6
['Line Item Value', 'Pack Price', 'Unit Price', 'Weight (Kilograms)', 'Fre
ight Cost (USD)', 'Line Item Insurance (USD)']
```

In [ ]:

```
cat_columns = [feature for feature in df.columns if df[feature].dtype!='Obj
print("Number of Columns: " , len(cat_columns))
print(cat_columns)
```

```
Number of Columns:  29
['Project Code', 'Country', 'Managed By', 'Fulfill Via', 'Vendor INCO Ter
m', 'Shipment Mode', 'PQ First Sent to Client Date', 'PO Sent to Vendor Da
te', 'Scheduled Delivery Date', 'Delivered to Client Date', 'Delivery Reco
rded Date', 'Product Group', 'Sub Classification', 'Vendor', 'Item Descrip
tion', 'Molecule/Test Type', 'Brand', 'Dosage', 'Dosage Form', 'Unit of Me
asure (Per Pack)', 'Line Item Quantity', 'Line Item Value', 'Pack Price',
'Unit Price', 'Manufacturing Site', 'First Line Designation', 'Weight (Kil
ograms)', 'Freight Cost (USD)', 'Line Item Insurance (USD)']
```

In [ ]:

```
df.head()
```

Out[43]:

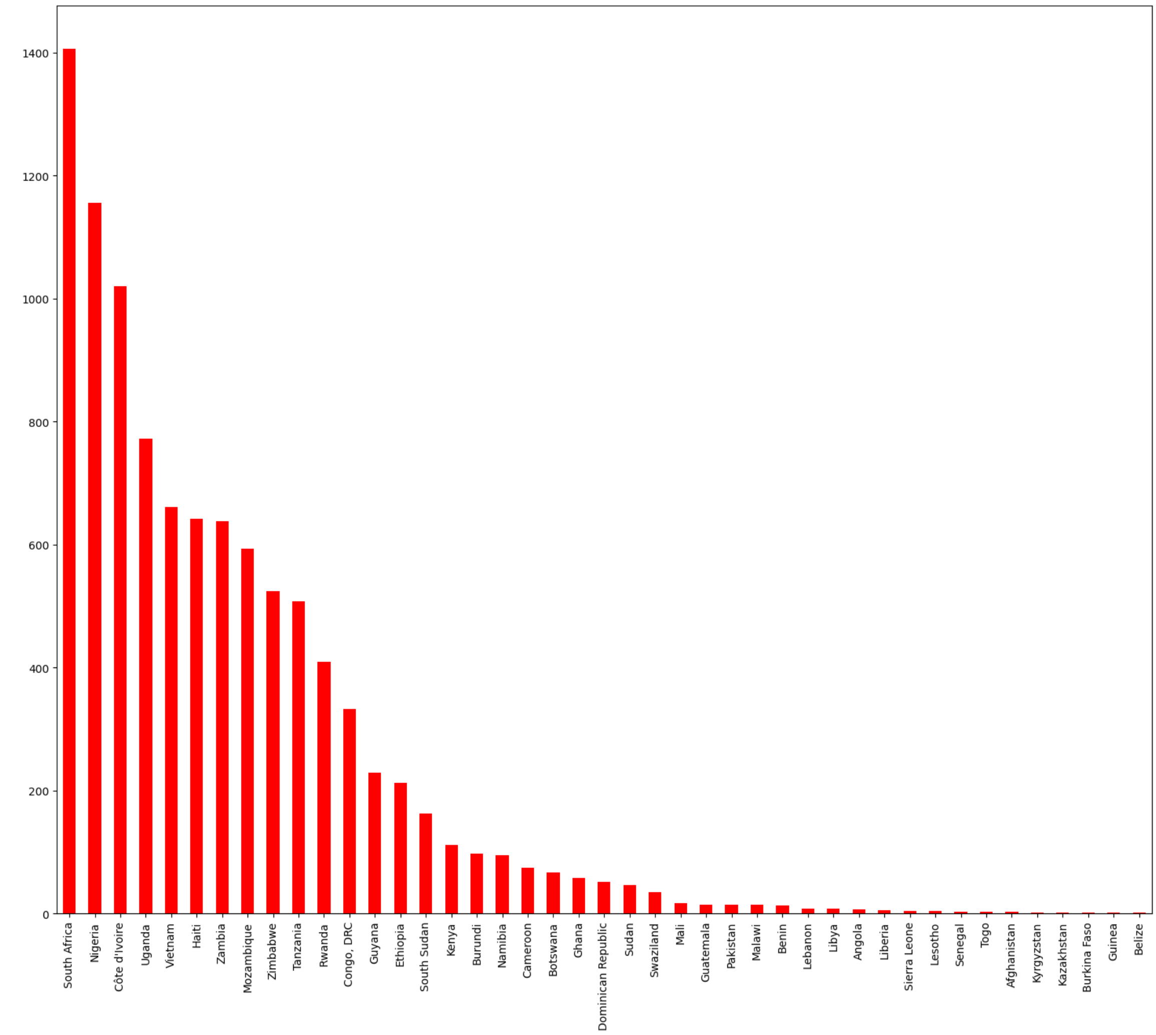
	Project Code	Country	Managed By	Fulfill Via	Vendor INCO Term	Shipment Mode	PQ First Sent to Client Date	PO Sent to Vendor Date	Scheduled Delivery Date	Deliver to Cli Da
16	102-NG-T01	Nigeria	PMO - US	Direct Drop	CIP	NaN	NaT	NaT	2007-05-07	2007-(
19	102-NG-T01	Nigeria	PMO - US	Direct Drop	EXW	Air	NaT	2007-05-13	2007-06-19	2007-(
21	104-CI-T01	Côte d'Ivoire	PMO - US	Direct Drop	CIP	Air	NaT	2007-07-13	2007-10-02	2007-'
22	108-VN-T01	Vietnam	PMO - US	Direct Drop	EXW	Air	NaT	2007-07-04	2007-10-15	2007-'
23	108-VN-T01	Vietnam	PMO - US	Direct Drop	FCA	Air	NaT	2007-07-04	2007-08-27	2007-(

In [ ]:

```
import seaborn as sns
```

```
In [ ]: # top 10 country
        plt.figure(figsize=(18,15))
        df['Country'].value_counts().plot(kind="bar", color='red')
```

```
Out[45]: <Axes: >
```

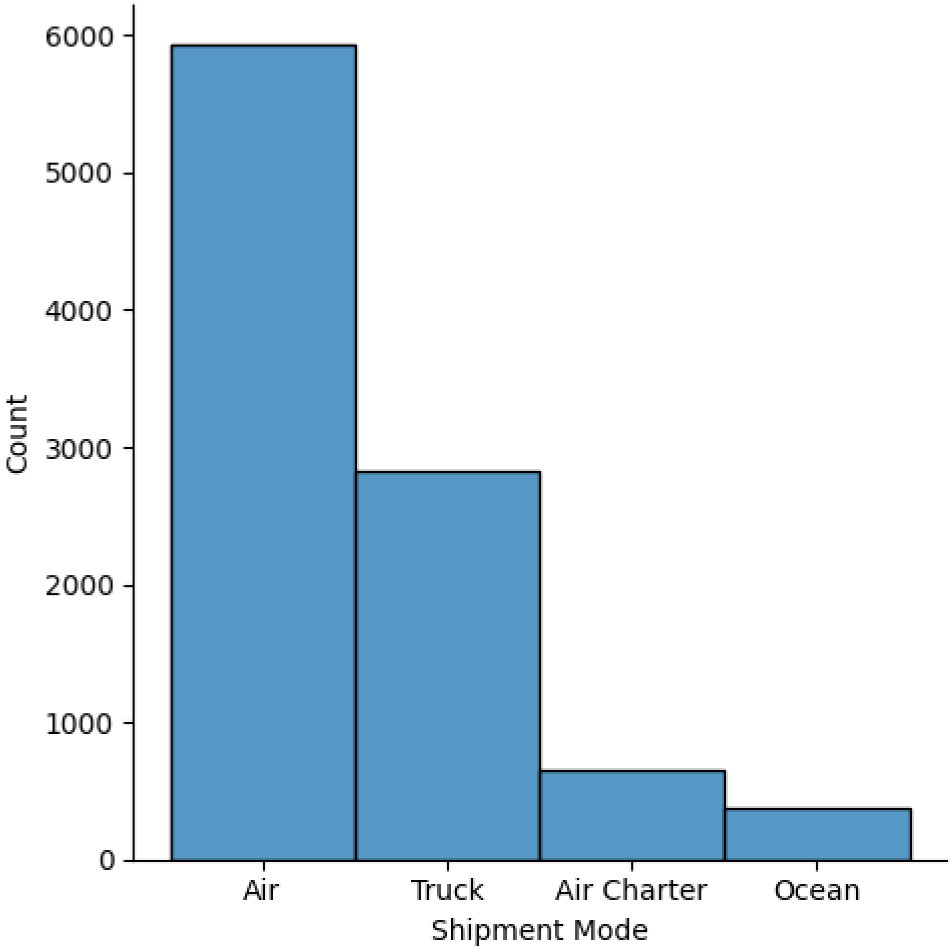


```
In [ ]: df['Shipment Mode'].value_counts()
```

```
Out[46]: Air            5928
         Truck          2830
         Air Charter     650
         Ocean           371
         Name: Shipment Mode, dtype: int64
```

```
In [ ]: sns.displot(df['Shipment Mode'])
```

```
Out[47]: <seaborn.axisgrid.FacetGrid at 0x7d5ab85c98d0>
```



```
In [ ]: 
```

- it is indicate the by Air Shipment MOde is too much demand

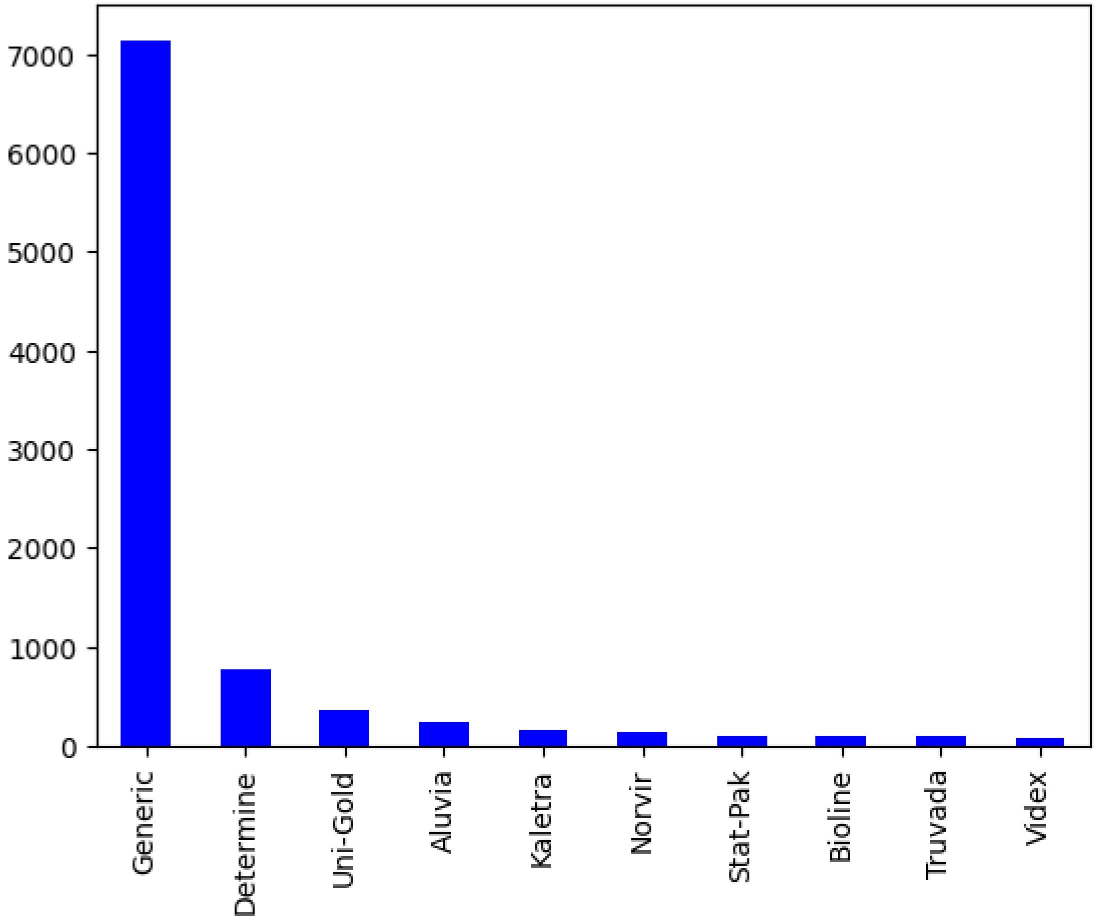
find out the top 10 brand

```
In [ ]: 
top10 = df['Brand'].value_counts().sort_values(ascending=False).head(10)
top10
```

```
Out[48]: Generic       7135
Determine      775
Uni-Gold       359
Aluvia         242
Kaletra        161
Norvir         135
Stat-Pak       108
Bioline        107
Truvada         92
Videx           78
Name: Brand, dtype: int64
```

In [ ]:
```
top10.plot(kind='bar', color ='blue')
```

Out[49]:
```
<Axes: >
```



find out the how many product group category

In [ ]:
```
df['Product Group'].value_counts()
```

Out[50]:
```
ARV     8339
HRDT    1648
ANTM      22
ACT       16
MRDT       8
Name: Product Group, dtype: int64
```

In [ ]:

```
# check the Item Description name
df['Item Description'].value_counts()
```

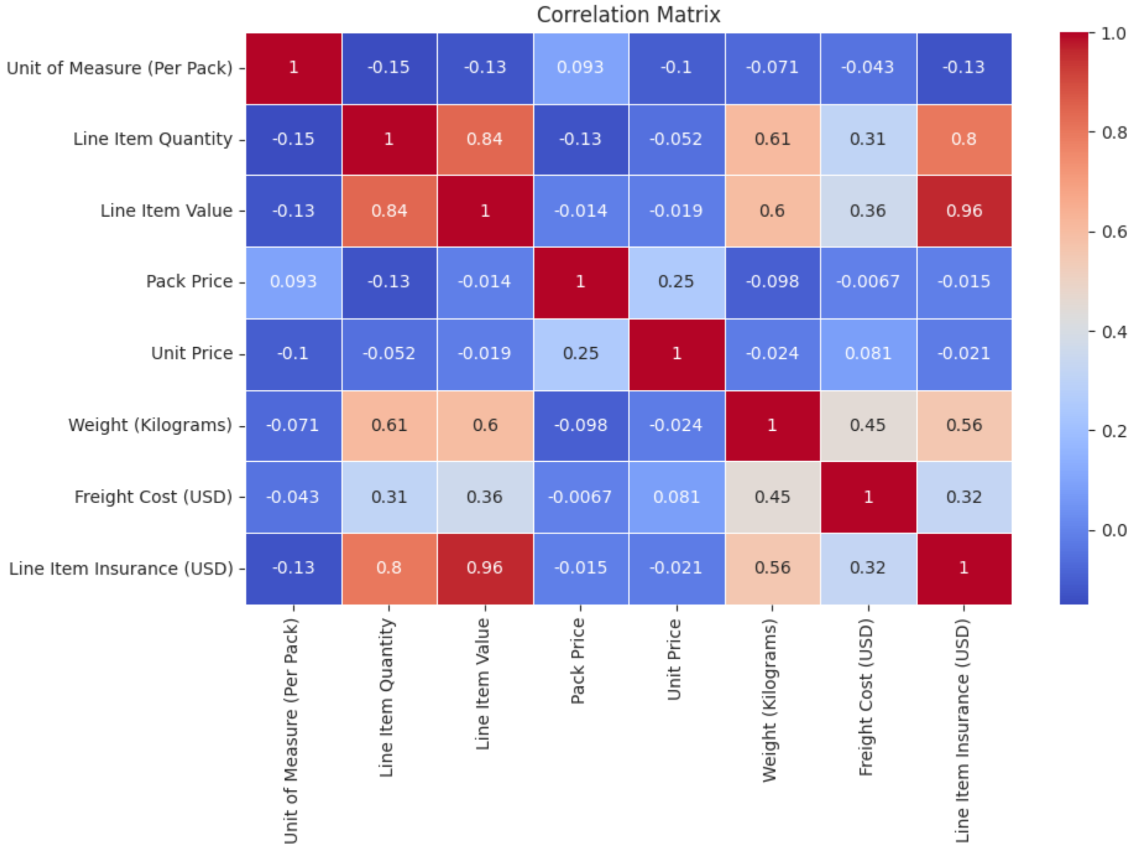
Out[51]:

```
Efavirenz 600mg, tablets, 30 Tabs
726
Nevirapine 200mg, tablets, 60 Tabs
614
Lamivudine/Nevirapine/Zidovudine 150/200/300mg, tablets, 60 Tabs
578
Lamivudine/Zidovudine 150/300mg, tablets, 60 Tabs
576
HIV 1/2, Determine Complete HIV Kit, 100 Tests
554

...
HIV 1/2, ImmunoComb II BiSpot EIA Kit, 36 Tests
1
Malaria Antigen P.f Kit, 30 x 1 Test
1
Lopinavir/Ritonavir 80/20mg/ml [Kaletra], oral solution, cool, Bottle, 160
ml      1
HIV 1/2, InstantChek HIV 1+2 Kit, 100 Tests
1
Lopinavir/Ritonavir 200/50mg, [DON] tablets, 120 Tabs
1
Name: Item Description, Length: 182, dtype: int64
```

#Correlation Analysis

```
In [ ]: # Correlation Analysis
        correlation_matrix = df.corr()
        plt.figure(figsize=(10, 6))
        sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5
        plt.title('Correlation Matrix')
        plt.show()
```



```
In [ ]: # finf out top 10 unit price
        df['Unit Price'].value_counts().sort_values(ascending=False).head(10)

Out[52]: 0.04    708
         0.01    482
         0.12    450
         0.14    439
         0.11    396
         0.80    385
         1.60    358
         0.05    340
         0.16    340
         0.19    319
         Name: Unit Price, dtype: int64
```

find out top 10 brand and unit price

In [ ]:
```
df.groupby('Brand')['Pack Price'].sum()
```

Out[53]:
```
Brand
Aluvia             10386.54
Atripla              806.40
Bioline             2206.32
Bundi                 75.00
Capillus            3687.69
CareStart             23.40
Clearview           1505.00
Coartem              500.63
Colloidal Gold      1686.00
Combivir              71.61
Crixivan            1549.32
Determine          57434.56
DoubleCheck          267.56
Epivir               284.41
First Response       217.90
Generic            59774.17
Genie               2225.71
Hexagon              334.74
INSTi                 94.01
ImmunoComb           295.00
InstantCHEK           75.00
Intelence           1358.02
Invirase            4135.41
Isentress           2945.38
Kaletra             5460.78
LAV                 1674.80
Multispot           6323.23
Norvir              4712.31
OraQuick           15015.25
Paramax              187.50
Pepti-LAV            238.65
Prezista            2769.12
Retrovir             564.39
Reveal                51.00
Reyataz              729.93
Stat-Pak            3034.79
Stocrin/Sustiva      956.54
Trizivir             987.55
Truvada             2652.52
Uni-Gold           11492.16
Videx               1024.06
Videx EC             894.37
Viracept             322.36
Viramune             469.58
Viread               823.22
Zerit                178.79
Ziagen               781.99
Name: Pack Price, dtype: float64
```

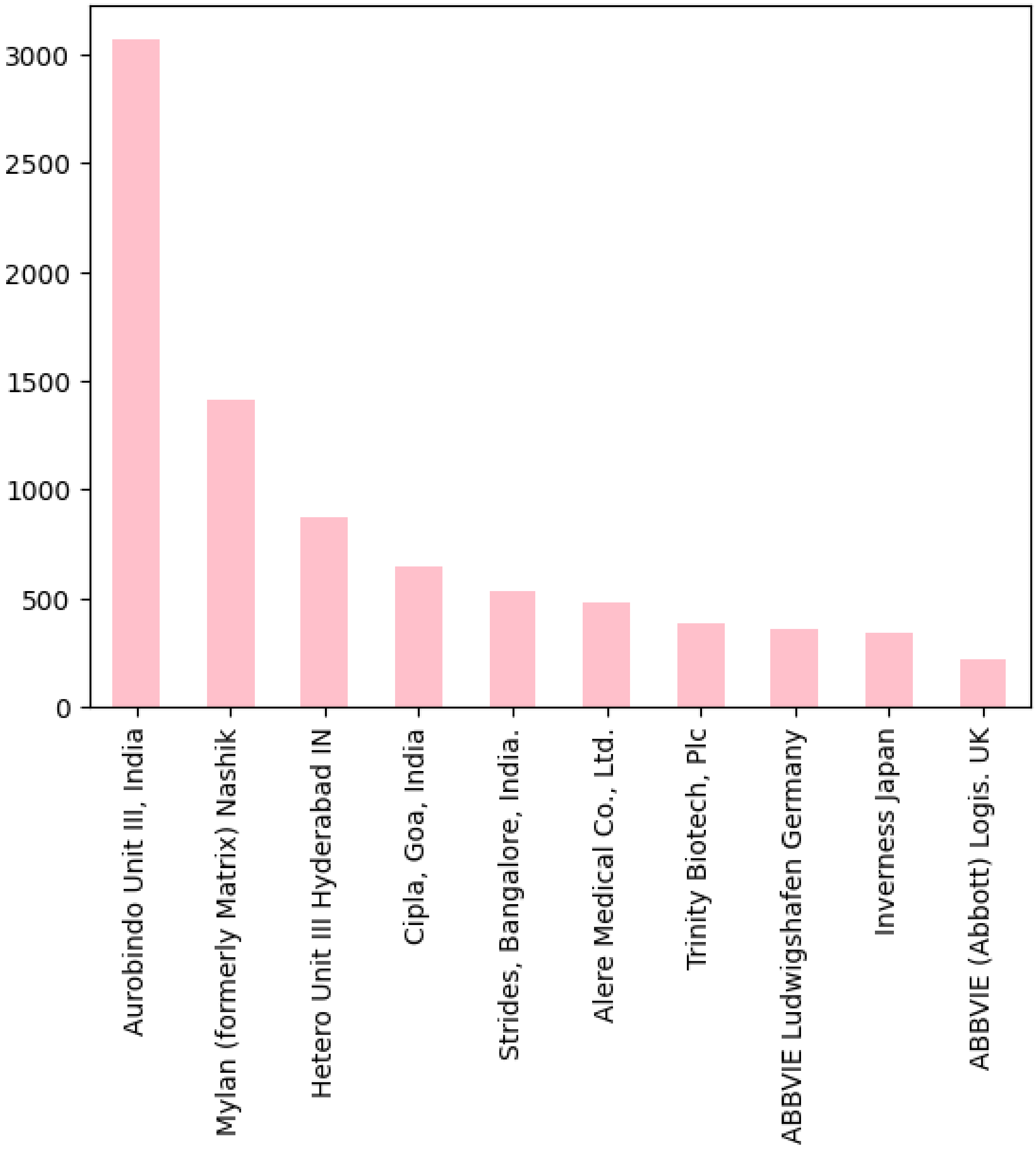
find out top 10 Manufacturing Site

```
In [ ]: manu_fact = df['Manufacturing Site'].value_counts().sort_values(ascending=F
        manu_fact
```

```
Out[54]: Aurobindo Unit III, India        3070
         Mylan (formerly Matrix) Nashik    1415
         Hetero Unit III Hyderabad IN       869
         Cipla, Goa, India                  644
         Strides, Bangalore, India.         534
         Alere Medical Co., Ltd.            481
         Trinity Biotech, Plc               385
         ABBVIE Ludwigshafen Germany        361
         Inverness Japan                    344
         ABBVIE (Abbott) Logis. UK          216
         Name: Manufacturing Site, dtype: int64
```

```
In [ ]: manu_fact.plot(kind='bar', color='pink' )
```

```
Out[55]: <Axes: >
```



In [ ]:

```python
# find out top 10 brand and company name , where is Manufacturing site
```

In [ ]:

```python
df.groupby('Country')
```

Out[57]:

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7d5ab82ff640>
```

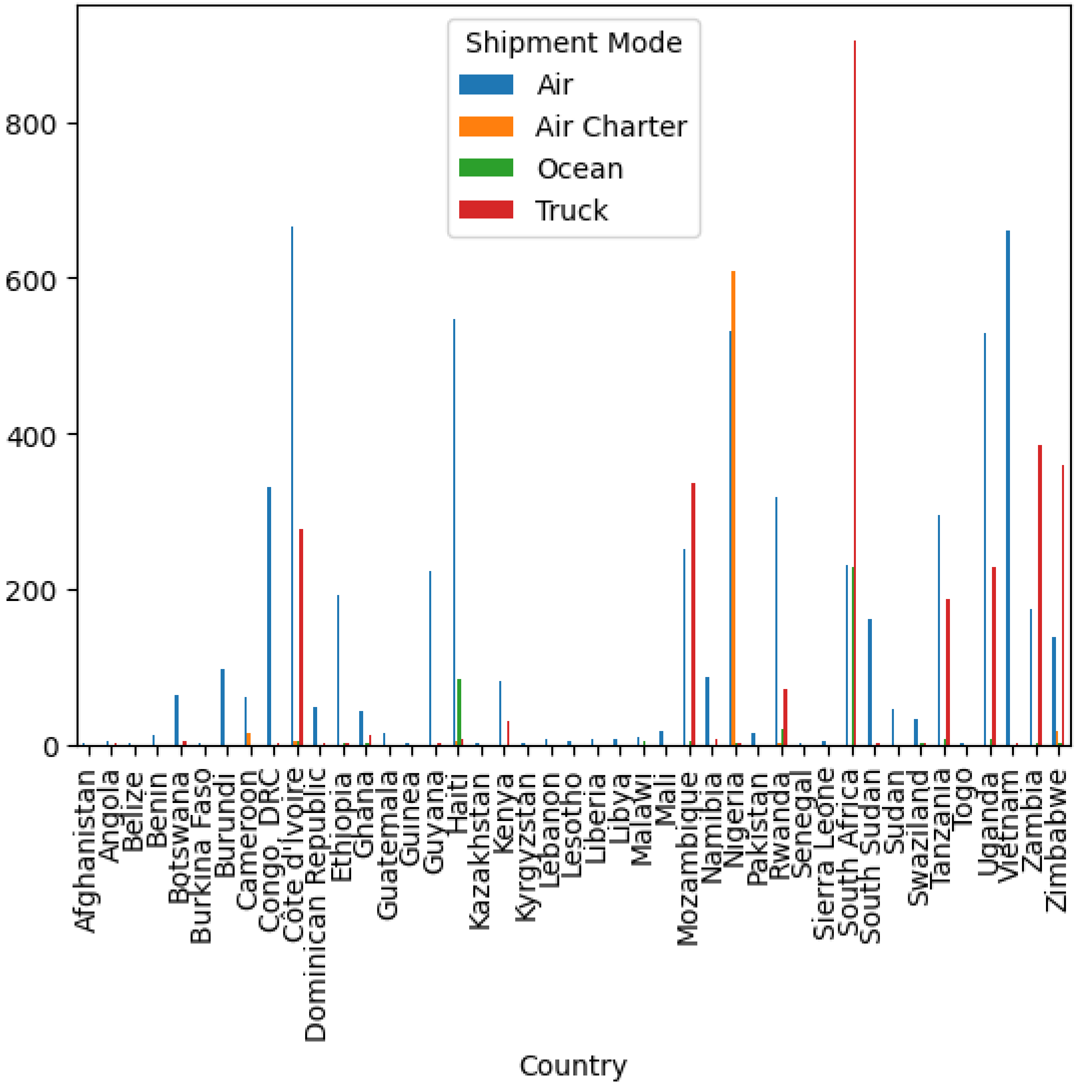
In [ ]:

```python
# check the Country and Shipment Mode
plt.figure(figsize=(18,15))
resume=pd.crosstab(df['Country'],df['Shipment Mode'])
resume.plot(kind='bar')
```

Out[58]:

```
<Axes: xlabel='Country'>
```

```
<Figure size 1800x1500 with 0 Axes>
```



In [ ]:

```
# check the Product Group item
df['Product Group'].value_counts()
```

Out[59]:

```
ARV     8339
HRDT    1648
ANTM      22
ACT       16
MRDT       8
Name: Product Group, dtype: int64
```

Find out top 25 Country, which Manufacturing Site is situated ?

In [ ]:

```

df.groupby(['Country'])['Manufacturing Site'].value_counts().sort_values(as
```

Out[60]:

```
Country        Manufacturing Site
South Africa   Aurobindo Unit III, India         703
Nigeria        Aurobindo Unit III, India         408
Côte d'Ivoire  Aurobindo Unit III, India         353
Haiti          Aurobindo Unit III, India         262
Nigeria        Mylan (formerly Matrix) Nashik    211
Uganda         Aurobindo Unit III, India         203
Côte d'Ivoire  Mylan (formerly Matrix) Nashik    171
Zambia         Aurobindo Unit III, India         169
Vietnam        Mylan (formerly Matrix) Nashik    161
Uganda         Mylan (formerly Matrix) Nashik    158
Vietnam        Aurobindo Unit III, India         151
               Hetero Unit III Hyderabad IN      146
Mozambique     Aurobindo Unit III, India         140
Tanzania       Aurobindo Unit III, India         127
Zimbabwe       Cipla, Goa, India                 124
Zambia         Mylan (formerly Matrix) Nashik    122
South Africa   Cipla, Goa, India                 121
Tanzania       Mylan (formerly Matrix) Nashik    105
Guyana         Aurobindo Unit III, India         105
Nigeria        Alere Medical Co., Ltd.           103
Rwanda         Aurobindo Unit III, India         101
Uganda         Hetero Unit III Hyderabad IN       93
Mozambique     Hetero Unit III Hyderabad IN       89
Zambia         Hetero Unit III Hyderabad IN       88
Vietnam        ABBVIE Ludwigshafen Germany        84
Name: Manufacturing Site, dtype: int64
```

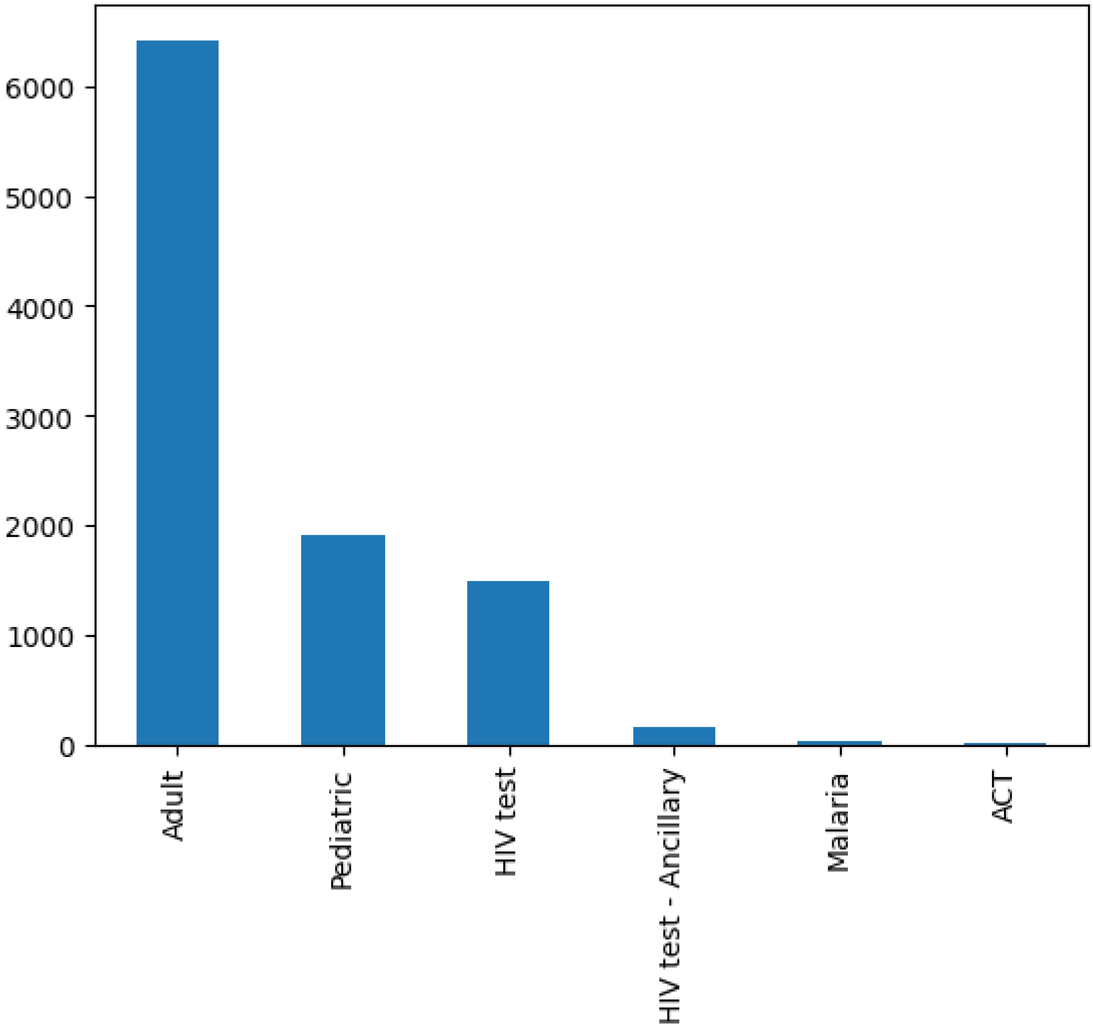
check the realtion ship between Vendor and Item Description

```
In [ ]: df.groupby(['Vendor'])['Item Description'].value_counts().sort_values(ascen
```

```
Out[61]: Vendor                Item Description
         Orgenics, Ltd         HIV 1/2, Determine Complete HIV Kit, 100 Tests
         505
         SCMS from RDC         Efavirenz 600mg, tablets, 30 Tabs
         482
                               Lamivudine/Nevirapine/Zidovudine 150/200/300mg, tabl
         ets, 60 Tabs        473
                               Lamivudine/Zidovudine 150/300mg, tablets, 60 Tabs
         454
                               Nevirapine 200mg, tablets, 60 Tabs
         445
         Trinity Biotech, Plc  HIV 1/2, Uni-Gold HIV Kit, 20 Tests
         321
         SCMS from RDC         Lamivudine/Tenofovir Disoproxil Fumarate 300/300mg,
         tablets, 30 Tabs    238
                               Lamivudine 150mg, tablets, 60 Tabs
         213
                               Lamivudine/Nevirapine/Stavudine 150/200/30mg, tablet
         s, 60 Tabs          204
                               Zidovudine 300mg, tablets, 60 Tabs
         198
         Name: Item Description, dtype: int64
```

```
In [ ]: df["Sub Classification"].value_counts().plot(kind='bar')
```

```
Out[62]: <Axes: >
```

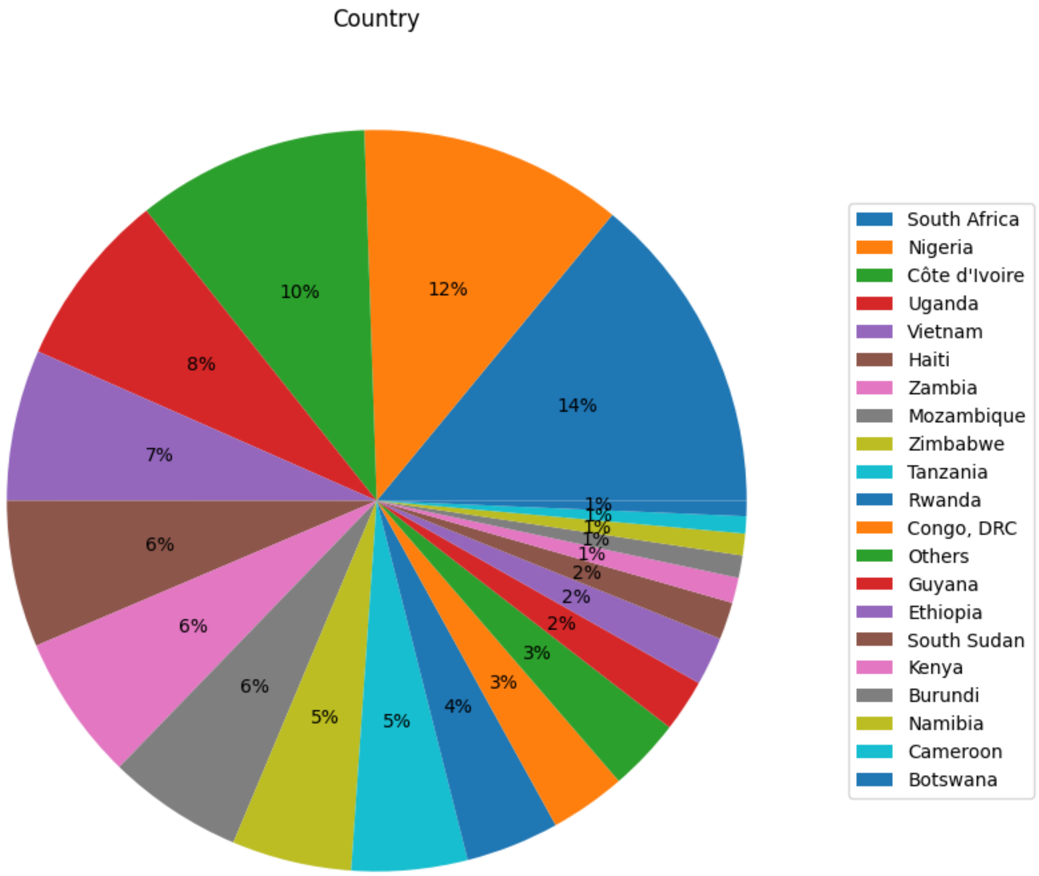


```
In [ ]: df['Fulfill Via'].value_counts()
```

```
Out[63]: From RDC       5232
         Direct Drop    4801
         Name: Fulfill Via, dtype: int64
```

Check the percentage of bussine occupaid by country wise

```
In [ ]: counts = df['Country'].value_counts()
        idx = counts[counts.lt(60)].index
        df.loc[df['Country'].isin(idx), 'Country'] = 'Others'
        df["Country"].value_counts().plot.pie(label='',title="Country",legend=True,
        plt.legend(loc='center left', bbox_to_anchor=(1.0, 0.5))
        plt.show()
```



In [ ]:
```
df.head()
```

Out[65]:

	Project Code	Country	Managed By	Fulfill Via	Vendor INCO Term	Shipment Mode	PQ First Sent to Client Date	PO Sent to Vendor Date	Scheduled Delivery Date	Deliver to Cli Da
16	102-NG-T01	Nigeria	PMO - US	Direct Drop	CIP	NaN	NaT	NaT	2007-05-07	2007-
19	102-NG-T01	Nigeria	PMO - US	Direct Drop	EXW	Air	NaT	2007-05-13	2007-06-19	2007-
21	104-CI-T01	Côte d'Ivoire	PMO - US	Direct Drop	CIP	Air	NaT	2007-07-13	2007-10-02	2007-
22	108-VN-T01	Vietnam	PMO - US	Direct Drop	EXW	Air	NaT	2007-07-04	2007-10-15	2007-
23	108-VN-T01	Vietnam	PMO - US	Direct Drop	FCA	Air	NaT	2007-07-04	2007-08-27	2007-

Find out heighest price of Shipment Mode

In [ ]:
```
df.groupby('Shipment Mode')['Freight Cost (USD)'].sum()
```

Out[66]:
```
Shipment Mode
Air            6.359307e+07
Air Charter    1.143544e+07
Ocean          4.578917e+06
Truck          3.034147e+07
Name: Freight Cost (USD), dtype: float64
```

- it is indiacted the Heighest Price of Shipment Mode in Air 6.359307e+07

Find out the lowest price of shipment mode

In [ ]:
```
df.groupby('Shipment Mode')['Freight Cost (USD)'].min()
```

Out[67]:
```
Shipment Mode
Air              0.75
Air Charter    134.00
Ocean          146.50
Truck           22.29
Name: Freight Cost (USD), dtype: float64
```

Find the total number of barnd

In [ ]:
```
len(df.groupby('Brand'))
```

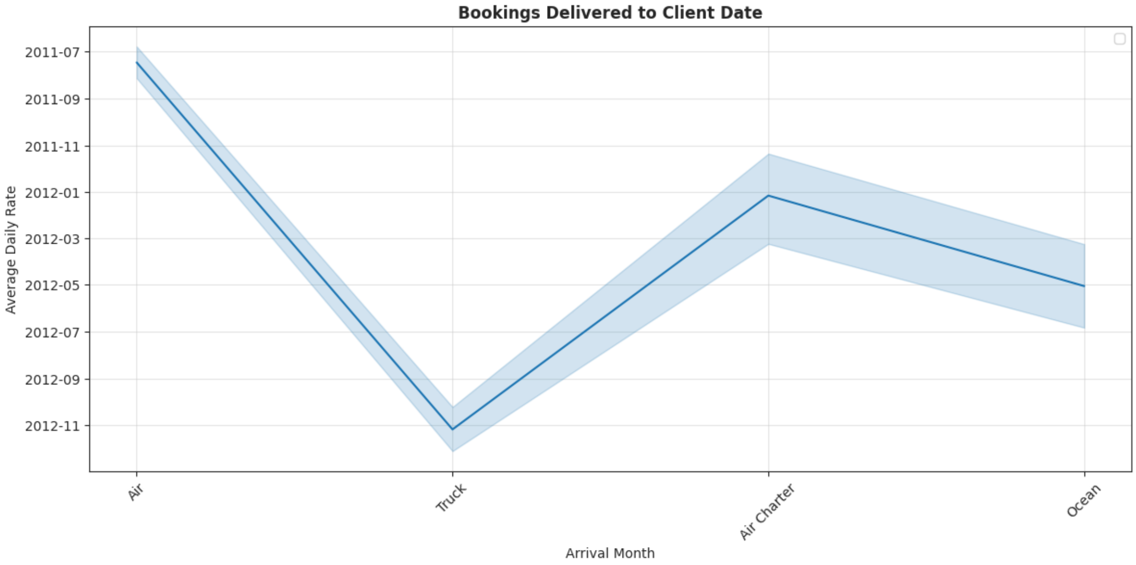
Out[68]:
```
47
```

In [ ]:
```
# Arrange the month for plotting Delivered to Client Date

df['Delivered to Client Date']=pd.Categorical(df['Delivered to Client Date'

#plot line chart
plt.figure(figsize=(14,6))
sns.set_style('ticks')
sns.lineplot(x='Shipment Mode',y='Delivered to Client Date', data=df)
plt.title('Bookings Delivered to Client Date' , weight='bold')
plt.xlabel('Arrival Month')
plt.ylabel('Average Daily Rate')
plt.xticks(rotation=45)
plt.legend(loc='upper right')
plt.grid(alpha=0.5)
```

```
WARNING:matplotlib.legend:No artists with labels found to put in legend.
Note that artists whose label start with an underscore are ignored when le
gend() is called with no argument.
```



In [ ]:
```
# Check Total freight Cost (USD)
total_freight_cost = df['Freight Cost (USD)'].sum()
total_freight_cost
```

Out[70]:
```
112353922.2273072
```

it is indicated the top Country by total expence

In [ ]:
```
country_summary = df.groupby('Country').sum().reset_index()
country_summary
```

Out[71]:

	Country	Unit of Measure (Per Pack)	Line Item Quantity	Line Item Value	Pack Price	Unit Price	Weight (Kilograms)	Freight (U
0	Botswana	3619	118902	1.596899e+06	6540.54	257.81	3.066091e+04	2.238884
1	Burundi	8031	203212	3.351580e+06	2051.81	55.47	1.311839e+05	7.783782
2	Cameroon	4342	1790405	1.462917e+07	3864.62	55.66	2.486705e+05	2.003313
3	Congo, DRC	30445	518546	5.772336e+06	6032.41	171.91	6.681158e+05	3.241724
4	Côte d'Ivoire	91487	11637154	1.174490e+08	20256.87	615.41	2.384526e+06	9.738285
5	Ethiopia	13628	2554695	1.872480e+07	3418.42	836.97	4.321305e+05	2.444096
6	Guyana	23496	182767	4.134950e+06	5306.95	101.17	2.346828e+05	1.384353
7	Haiti	66550	5223263	4.323458e+07	16700.30	550.18	1.160608e+06	6.826143
8	Kenya	8249	570631	3.393156e+07	6349.14	128.40	1.927046e+05	1.651076
9	Mozambique	45102	19073498	1.787870e+08	12896.17	246.86	2.941291e+06	5.982037
10	Namibia	4780	613658	5.857024e+06	3777.44	107.76	1.036720e+05	6.263879
11	Nigeria	81530	33842564	3.486618e+08	24496.51	516.64	4.791052e+06	1.872887
12	Others	24562	2780199	2.633626e+07	11975.10	305.05	4.887200e+05	2.706812
13	Rwanda	29709	8708314	6.895871e+07	6870.57	585.14	1.059129e+06	6.522661
14	South Africa	126293	22995781	1.086701e+08	24318.90	442.81	1.742075e+06	1.489590
15	South Sudan	9461	190158	2.132357e+06	4368.73	202.52	2.697745e+05	1.404555
16	Tanzania	36000	12387823	1.280563e+08	10718.24	203.58	1.628875e+06	5.953967
17	Uganda	48518	11883640	9.597446e+07	15606.63	277.42	1.810668e+06	7.836618
18	Vietnam	44786	6532326	5.305512e+07	11015.11	149.83	1.146425e+06	5.002666
19	Zambia	43190	28058534	2.387675e+08	10410.32	190.66	3.111783e+06	7.738134
20	Zimbabwe	35654	17384535	1.040576e+08	6309.89	132.41	2.227336e+06	6.664060

We are compare the Country wise Freight Cost (USD) price

In [ ]:

```
country_summary = country_summary[['Country','Freight Cost (USD)']]
country_summary
```

Out[72]:

	Country	Freight Cost (USD)
0	Botswana	2.238884e+05
1	Burundi	7.783782e+05
2	Cameroon	2.003313e+06
3	Congo, DRC	3.241724e+06
4	Côte d'Ivoire	9.738285e+06
5	Ethiopia	2.444096e+06
6	Guyana	1.384353e+06
7	Haiti	6.826143e+06
8	Kenya	1.651076e+06
9	Mozambique	5.982037e+06
10	Namibia	6.263879e+05
11	Nigeria	1.872887e+07
12	Others	2.706812e+06
13	Rwanda	6.522661e+06
14	South Africa	1.489590e+07
15	South Sudan	1.404555e+06
16	Tanzania	5.953967e+06
17	Uganda	7.836618e+06
18	Vietnam	5.002666e+06
19	Zambia	7.738134e+06
20	Zimbabwe	6.664060e+06

We are compare Country wise , Shipment Mode and Freight Cost (USD) in list

In [ ]:

```
country_summary = df.groupby(['Country', 'Shipment Mode']).sum().reset_inde
country_summary
```

Out[73]:

	Country	Shipment Mode	Unit of Measure (Per Pack)	Line Item Quantity	Line Item Value	Pack Price	Unit Price	Weight (Kilograms)
0	Botswana	Air	3299	117497	1.546999e+06	5736.54	248.49	2.342900e+04
1	Botswana	Truck	320	1405	4.990000e+04	804.00	9.32	7.231911e+03
2	Burundi	Air	8031	203212	3.351580e+06	2051.81	55.47	1.311839e+05
3	Cameroon	Air	3742	1201005	1.064419e+07	3775.48	53.31	1.512044e+05
4	Cameroon	Air Charter	600	589400	3.984977e+06	89.14	2.35	9.746610e+04
5	Congo, DRC	Air	30344	513546	5.465586e+06	5909.71	163.25	6.681158e+05
6	Congo, DRC	Truck	101	5000	3.067500e+05	122.70	8.66	0.000000e+00
7	Côte d'Ivoire	Air	64396	5057167	6.109787e+07	15189.66	282.92	1.259278e+06
8	Côte d'Ivoire	Air Charter	240	79898	7.662445e+05	78.74	1.85	1.014946e+04
9	Côte d'Ivoire	Ocean	600	68973	2.107216e+06	153.28	1.29	1.126600e+04
10	Côte d'Ivoire	Truck	19140	5838421	4.781743e+07	2331.12	48.12	9.645200e+05
11	Ethiopia	Air	10712	2166796	1.727690e+07	2982.11	830.95	3.496495e+05
12	Ethiopia	Ocean	120	13703	4.604208e+05	33.60	0.28	3.476000e+03
13	Ethiopia	Truck	536	72	3.477480e+03	143.42	3.14	2.794456e+03
14	Guyana	Air	22861	181225	4.106339e+06	5159.31	97.48	2.260614e+05
15	Guyana	Truck	390	803	1.826660e+03	11.14	0.12	5.648911e+03
16	Haiti	Air	58750	3219641	2.750965e+07	15287.00	529.75	8.717761e+05
17	Haiti	Air Charter	740	635	7.936880e+03	85.32	0.47	8.440367e+03
18	Haiti	Ocean	6360	1997851	1.508941e+07	490.48	11.58	2.802549e+05
19	Haiti	Truck	600	4536	5.843750e+05	765.50	7.66	0.000000e+00
20	Kenya	Air	6424	463278	3.036991e+07	5548.40	109.16	1.536600e+05
21	Kenya	Truck	1825	107353	3.561651e+06	800.74	19.24	3.904464e+04
22	Mozambique	Air	21038	2847789	3.689066e+07	10386.31	192.91	5.553244e+05
23	Mozambique	Ocean	160	29647	1.327040e+06	176.00	5.60	1.070000e+04
24	Mozambique	Truck	23904	16196062	1.405693e+08	2333.86	48.35	2.375267e+06
25	Namibia	Air	4360	500744	5.048938e+06	3630.57	104.55	8.041919e+04
26	Namibia	Truck	300	112314	7.744857e+05	34.87	0.81	2.031137e+04
27	Nigeria	Air	46419	5795644	1.092045e+08	18904.48	360.23	1.119189e+06
28	Nigeria	Air Charter	33510	28012687	2.388676e+08	5288.92	114.95	3.648565e+06
29	Nigeria	Ocean	60	13334	1.520076e+05	11.40	0.19	1.626000e+03
30	Nigeria	Truck	300	1858	3.249900e+03	3.47	0.04	4.040000e+02
31	Others	Air	21538	1774883	1.655269e+07	11321.80	296.90	3.554139e+05
32	Others	Ocean	510	825734	8.308996e+06	78.28	2.26	9.190600e+04
33	Others	Truck	2514	179582	1.474571e+06	575.02	5.89	4.140010e+04
34	Rwanda	Air	22809	4353055	4.118450e+07	6208.88	571.10	5.831234e+05

	Country	Shipment Mode	Unit of Measure (Per Pack)	Line Item Quantity	Line Item Value	Pack Price	Unit Price	Weight (Kilograms)
35	Rwanda	Air Charter	240	1800	3.330000e+03	1.85	0.01	7.180000e+02
36	Rwanda	Ocean	1170	1854991	1.117828e+07	190.35	4.20	1.620775e+05
37	Rwanda	Truck	5490	2498468	1.659260e+07	469.49	9.83	3.132107e+05
38	South Africa	Air	23219	4477793	2.438328e+07	2137.60	42.86	3.725663e+05
39	South Africa	Ocean	16410	18004326	7.616652e+07	1146.29	26.92	1.355662e+06
40	South Africa	Truck	82740	498237	7.905571e+06	20195.85	357.84	1.384728e+04
41	South Sudan	Air	9341	189693	2.130103e+06	4359.20	202.36	2.669240e+05
42	South Sudan	Truck	120	465	2.253300e+03	9.53	0.16	2.850456e+03
43	Tanzania	Air	20924	5077126	6.045069e+07	8131.33	155.22	7.600916e+05
44	Tanzania	Ocean	210	490568	5.900185e+06	80.12	2.61	5.528246e+04
45	Tanzania	Truck	13120	6547864	5.786909e+07	1921.02	37.24	7.602817e+05
46	Uganda	Air	35018	5853268	4.981260e+07	13713.58	238.77	9.922569e+05
47	Uganda	Ocean	360	549043	3.211445e+06	44.00	0.94	5.415446e+04
48	Uganda	Truck	12690	5472415	4.285740e+07	1770.58	35.96	7.578957e+05
49	Vietnam	Air	44756	6530358	5.301441e+07	10994.42	149.14	1.146425e+06
50	Vietnam	Truck	30	1968	4.071792e+04	20.69	0.69	0.000000e+00
51	Zambia	Air	14300	2781096	3.672319e+07	6180.32	100.84	3.839441e+05
52	Zambia	Ocean	510	94528	9.492215e+05	15.70	0.41	1.504800e+04
53	Zambia	Truck	20790	23375265	1.812618e+08	2885.10	67.60	2.498826e+06
54	Zimbabwe	Air	13984	1281431	1.420968e+07	3941.24	78.84	2.935205e+05
55	Zimbabwe	Air Charter	690	795000	2.742166e+06	71.03	2.18	7.851546e+04
56	Zimbabwe	Ocean	180	289053	1.327196e+06	17.58	0.29	1.982600e+04
57	Zimbabwe	Truck	20530	15010859	8.570585e+07	2243.40	50.23	1.832073e+06

```
In [ ]: country_summary = country_summary[['Country','Shipment Mode', 'Freight Cost
        country_summary
```

Out[74]:

	Country	Shipment Mode	Freight Cost (USD)
0	Botswana	Air	1.893580e+05
1	Botswana	Truck	3.453043e+04
2	Burundi	Air	7.783782e+05
3	Cameroon	Air	1.546904e+06
4	Cameroon	Air Charter	4.564088e+05
5	Congo, DRC	Air	3.219517e+06
6	Congo, DRC	Truck	2.220647e+04
7	Côte d'Ivoire	Air	5.772273e+06
8	Côte d'Ivoire	Air Charter	4.255951e+04
9	Côte d'Ivoire	Ocean	1.274085e+05
10	Côte d'Ivoire	Truck	3.259164e+06
11	Ethiopia	Air	2.153580e+06
12	Ethiopia	Ocean	1.172982e+04
13	Ethiopia	Truck	2.271147e+04
14	Guyana	Air	1.346082e+06
15	Guyana	Truck	2.538447e+04
16	Haiti	Air	5.890579e+06
17	Haiti	Air Charter	4.163883e+04
18	Haiti	Ocean	8.255730e+05
19	Haiti	Truck	6.661941e+04
20	Kenya	Air	1.475429e+06
21	Kenya	Truck	1.756469e+05
22	Mozambique	Air	2.678513e+06
23	Mozambique	Ocean	8.349039e+04
24	Mozambique	Truck	3.220033e+06
25	Namibia	Air	5.542231e+05
26	Namibia	Truck	6.037275e+04
27	Nigeria	Air	8.226885e+06
28	Nigeria	Air Charter	1.035219e+07
29	Nigeria	Ocean	1.709624e+04
30	Nigeria	Truck	9.352630e+03
31	Others	Air	2.397750e+06
32	Others	Ocean	1.420389e+05
33	Others	Truck	1.670233e+05
34	Rwanda	Air	5.074776e+06
35	Rwanda	Air Charter	4.370800e+03
36	Rwanda	Ocean	6.507976e+05
37	Rwanda	Truck	7.927164e+05
38	South Africa	Air	2.019066e+06

	Country	Shipment Mode	Freight Cost (USD)
39	South Africa	Ocean	2.377234e+06
40	South Africa	Truck	1.003326e+07
41	South Sudan	Air	1.391811e+06
42	South Sudan	Truck	1.274423e+04
43	Tanzania	Air	4.149984e+06
44	Tanzania	Ocean	1.227775e+05
45	Tanzania	Truck	1.494453e+06
46	Uganda	Air	4.907013e+06
47	Uganda	Ocean	9.337122e+04
48	Uganda	Truck	2.807341e+06
49	Vietnam	Air	5.001265e+06
50	Vietnam	Truck	1.401490e+03
51	Zambia	Air	2.830421e+06
52	Zambia	Ocean	6.394006e+04
53	Zambia	Truck	4.078471e+06
54	Zimbabwe	Air	1.989263e+06
55	Zimbabwe	Air Charter	5.382669e+05
56	Zimbabwe	Ocean	6.345907e+04
57	Zimbabwe	Truck	4.058035e+06

In [ ]:

Statistical Analysis

In [ ]:

```
# check the summary of statistical
df.describe()
```

Out[75]:

	Unit of Measure (Per Pack)	Line Item Quantity	Line Item Value	Pack Price	Unit Price	Weigh (Kilograms
count	10033.000000	10033.000000	1.003300e+04	10033.000000	10033.000000	10033.00000
mean	77.686833	18663.471046	1.596869e+05	21.258315	0.611349	2671.59225
std	76.650711	40482.366445	3.490771e+05	44.459721	3.320426	5672.61310
min	1.000000	1.000000	0.000000e+00	0.000000	0.000000	0.00000
25%	30.000000	407.000000	4.267000e+03	4.120000	0.070000	100.00000
50%	60.000000	3056.000000	3.044840e+04	8.820000	0.160000	1454.00000
75%	90.000000	17600.000000	1.687635e+05	23.000000	0.450000	2769.45551
max	1000.000000	619999.000000	5.951990e+06	1345.640000	238.650000	154780.00000

In [ ]:
```
# check the transpose value
df.describe().T
```

Out[76]:

	count	mean	std	min	25%	50%	7
Unit of Measure (Per Pack)	10033.0	77.686833	76.650711	1.00	30.00	60.000000	90.0000
Line Item Quantity	10033.0	18663.471046	40482.366445	1.00	407.00	3056.000000	17600.0000
Line Item Value	10033.0	159686.941312	349077.069994	0.00	4267.00	30448.400000	168763.5400
Pack Price	10033.0	21.258315	44.459721	0.00	4.12	8.820000	23.0000
Unit Price	10033.0	0.611349	3.320426	0.00	0.07	0.160000	0.4500
Weight (Kilograms)	10033.0	2671.592257	5672.613103	0.00	100.00	1454.000000	2769.4555
Freight Cost (USD)	10033.0	11198.437379	12344.983985	0.75	4454.62	11103.234819	11103.2348
Line Item Insurance (USD)	10033.0	240.205776	500.270659	0.00	6.51	47.110000	252.4000

In [ ]:
```
# check the correlation
df.corr()
```

Out[77]:

	Unit of Measure (Per Pack)	Line Item Quantity	Line Item Value	Pack Price	Unit Price	Weight (Kilograms)	Freight Cost (USD)	Lir Insu
Unit of Measure (Per Pack)	1.000000	-0.150273	-0.127548	0.092973	-0.103052	-0.071029	-0.043027	-0.1
Line Item Quantity	-0.150273	1.000000	0.839380	-0.131729	-0.051906	0.606994	0.311752	0.7
Line Item Value	-0.127548	0.839380	1.000000	-0.014006	-0.019387	0.598238	0.358078	0.9
Pack Price	0.092973	-0.131729	-0.014006	1.000000	0.251254	-0.097732	-0.006715	-0.0
Unit Price	-0.103052	-0.051906	-0.019387	0.251254	1.000000	-0.023980	0.080606	-0.0
Weight (Kilograms)	-0.071029	0.606994	0.598238	-0.097732	-0.023980	1.000000	0.450246	0.5
Freight Cost (USD)	-0.043027	0.311752	0.358078	-0.006715	0.080606	0.450246	1.000000	0.3
Line Item Insurance (USD)	-0.131912	0.798646	0.961350	-0.015350	-0.021423	0.557945	0.324064	1.0

```
In [ ]: # check the skewness
        df.skew()
```

```
Out[78]: Unit of Measure (Per Pack)      4.377980
         Line Item Quantity              4.988691
         Line Item Value                 5.790676
         Pack Price                     13.916055
         Unit Price                     40.100685
         Weight (Kilograms)              8.720364
         Freight Cost (USD)              6.051483
         Line Item Insurance (USD)       4.826275
         dtype: float64
```

```
In [ ]: # check the quantile value
        df.quantile()
```

```
Out[79]: Unit of Measure (Per Pack)         60.000000
         Line Item Quantity               3056.000000
         Line Item Value                 30448.400000
         Pack Price                          8.820000
         Unit Price                          0.160000
         Weight (Kilograms)               1454.000000
         Freight Cost (USD)              11103.234819
         Line Item Insurance (USD)          47.110000
         Name: 0.5, dtype: float64
```

```
In [ ]: # check the  covarrience

        df.cov()
```

Out[80]:

	Unit of Measure (Per Pack)	Line Item Quantity	Line Item Value	Pack Price	Unit Price	(K
Unit of Measure (Per Pack)	5.875332e+03	-4.662964e+05	-3.412804e+06	316.841230	-26.228153	-3.08
Line Item Quantity	-4.662964e+05	1.638822e+09	1.186167e+10	-237090.582886	-6977.151369	1.39
Line Item Value	-3.412804e+06	1.186167e+10	1.218548e+11	-217377.600310	-22470.848775	1.18
Pack Price	3.168412e+02	-2.370906e+05	-2.173776e+05	1976.666782	37.091369	-2.46
Unit Price	-2.622815e+01	-6.977151e+03	-2.247085e+04	37.091369	11.025229	-4.5
Weight (Kilograms)	-3.088431e+04	1.393905e+08	1.184618e+09	-24648.324902	-451.670765	3.2
Freight Cost (USD)	-4.071472e+04	1.557995e+08	1.543084e+09	-3685.608148	3304.088604	3.15
Line Item Insurance (USD)	-5.058314e+03	1.617429e+07	1.678834e+08	-341.417215	-35.585619	1.58

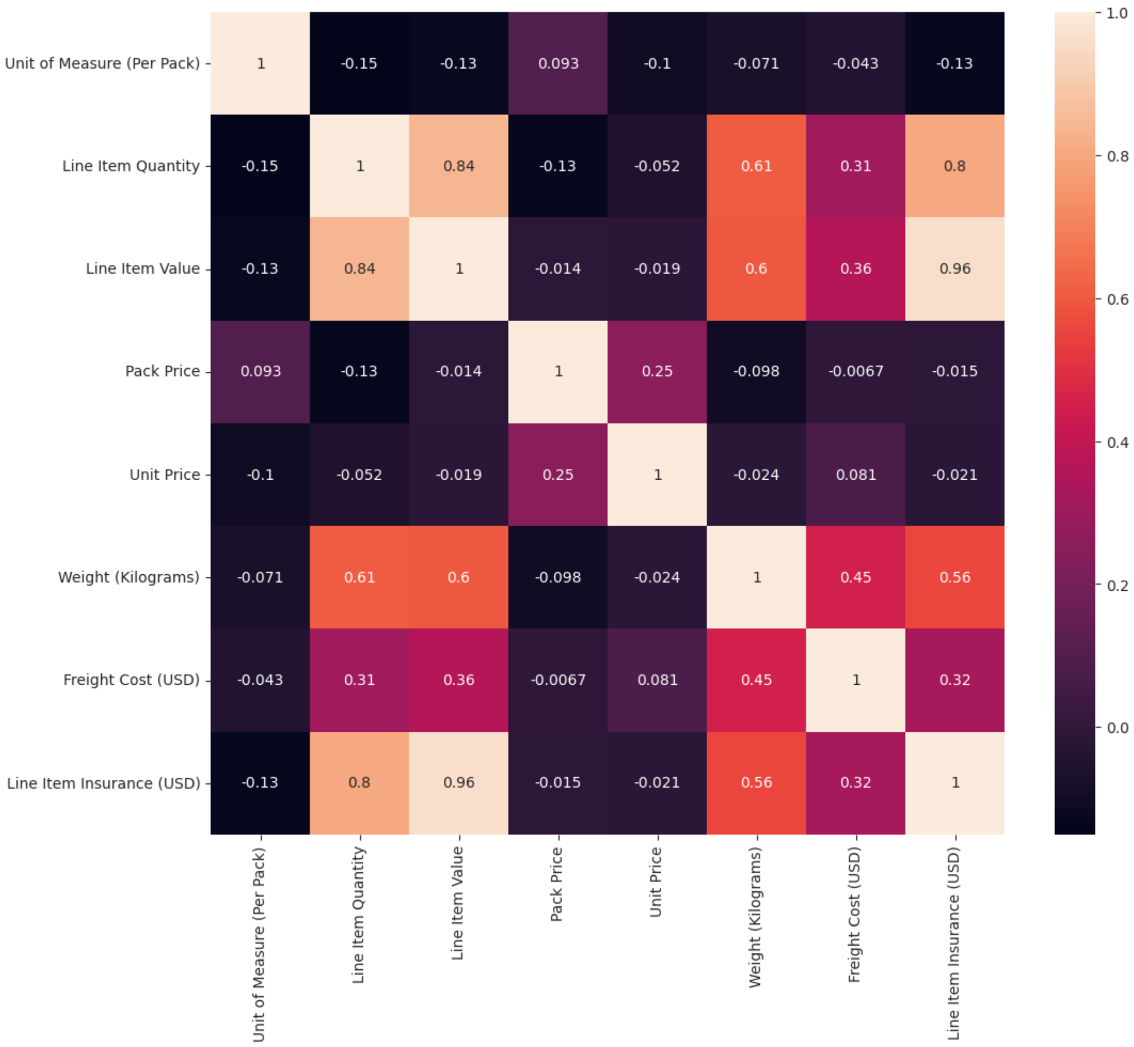
Heatmap

In [ ]:

```
plt.figure(figsize=(12,10))
sns.heatmap(df.corr(), annot = True)
```

Out[81]:

```
<Axes: >
```



In [ ]:

```
# check the Standard Deviation
df.std()
```

Out[82]:

```
PQ First Sent to Client Date     618 days 05:04:00.432302272
PO Sent to Vendor Date           833 days 21:11:21.117641600
Scheduled Delivery Date          866 days 09:01:07.647410512
Delivery Recorded Date           870 days 18:28:03.287990336
Unit of Measure (Per Pack)                         76.650711
Line Item Quantity                              40482.366445
Line Item Value                                349077.069994
Pack Price                                         44.459721
Unit Price                                          3.320426
Weight (Kilograms)                               5672.613103
Freight Cost (USD)                              12344.983985
Line Item Insurance (USD)                         500.270659
dtype: object
```

In [ ]:

```
# check the minimum number
df.min()
```

Out[83]:

```
Project Code                                                           100
-BJ-T30
Country                                                                  B
otswana
Managed By                                                  Ethiopia Field
Office
Fulfill Via                                                           Dire
ct Drop
Vendor INCO Term
CIF
PQ First Sent to Client Date                                  2009-01-04 0
0:00:00
PO Sent to Vendor Date                                        2007-02-07 0
0:00:00
Scheduled Delivery Date                                       2007-05-07 0
0:00:00
Delivered to Client Date                                      2007-01-24 0
0:00:00
Delivery Recorded Date                                        2007-05-07 0
0:00:00
Product Group
ACT
Sub Classification
ACT
Vendor                                          ABBOTT LABORATORIES (PUERT
O RICO)
Item Description                #102198**Didanosine 200mg [Videx], tablet
s, 60...
Molecule/Test Type                                                       A
bacavir
Brand
Aluvia
Dosage                                                                   1
00/25mg
Dosage Form
Capsule
Unit of Measure (Per Pack)
1
Line Item Quantity
1
Line Item Value
0.0
Pack Price
0.0
Unit Price
0.0
Manufacturing Site                                         ABBVIE (Abbott)
France
First Line Designation
No
Weight (Kilograms)
0.0
Freight Cost (USD)
0.75
Line Item Insurance (USD)
0.0
dtype: object
```

In [ ]:

```
# check the maximum number
df.max()
```

Out[84]:

```
Project Code                                              A02-SN-T50
Country                                                     Zimbabwe
Managed By                                South Africa Field Office
Fulfill Via                                                 From RDC
Vendor INCO Term                                      N/A - From RDC
PQ First Sent to Client Date                     2015-07-07 00:00:00
PO Sent to Vendor Date                           2015-08-24 00:00:00
Scheduled Delivery Date                          2015-12-31 00:00:00
Delivered to Client Date                         2015-09-14 00:00:00
Delivery Recorded Date                           2015-09-14 00:00:00
Product Group                                                   MRDT
Sub Classification                                         Pediatric
Vendor                                            ZEPHYR BIOMEDICALS
Item Description                 Zidovudine 300mg, tablets, 60 Tabs
Molecule/Test Type                                        Zidovudine
Brand                                                         Ziagen
Dosage                                                       80mg/ml
Dosage Form                                      Test kit - Ancillary
Unit of Measure (Per Pack)                                      1000
Line Item Quantity                                            619999
Line Item Value                                            5951990.4
Pack Price                                                   1345.64
Unit Price                                                    238.65
Manufacturing Site                           bioLytical Laboratories
First Line Designation                                           Yes
Weight (Kilograms)                                          154780.0
Freight Cost (USD)                                          289653.2
Line Item Insurance (USD)                                    7708.44
dtype: object
```

In [ ]:

```
# check the mean()
df.mean()
```

Out[85]:

```
Unit of Measure (Per Pack)        77.686833
Line Item Quantity             18663.471046
Line Item Value               159686.941312
Pack Price                        21.258315
Unit Price                         0.611349
Weight (Kilograms)              2671.592257
Freight Cost (USD)             11198.437379
Line Item Insurance (USD)        240.205776
dtype: float64
```

In [ ]:

```
# df.median()
df.median()
```

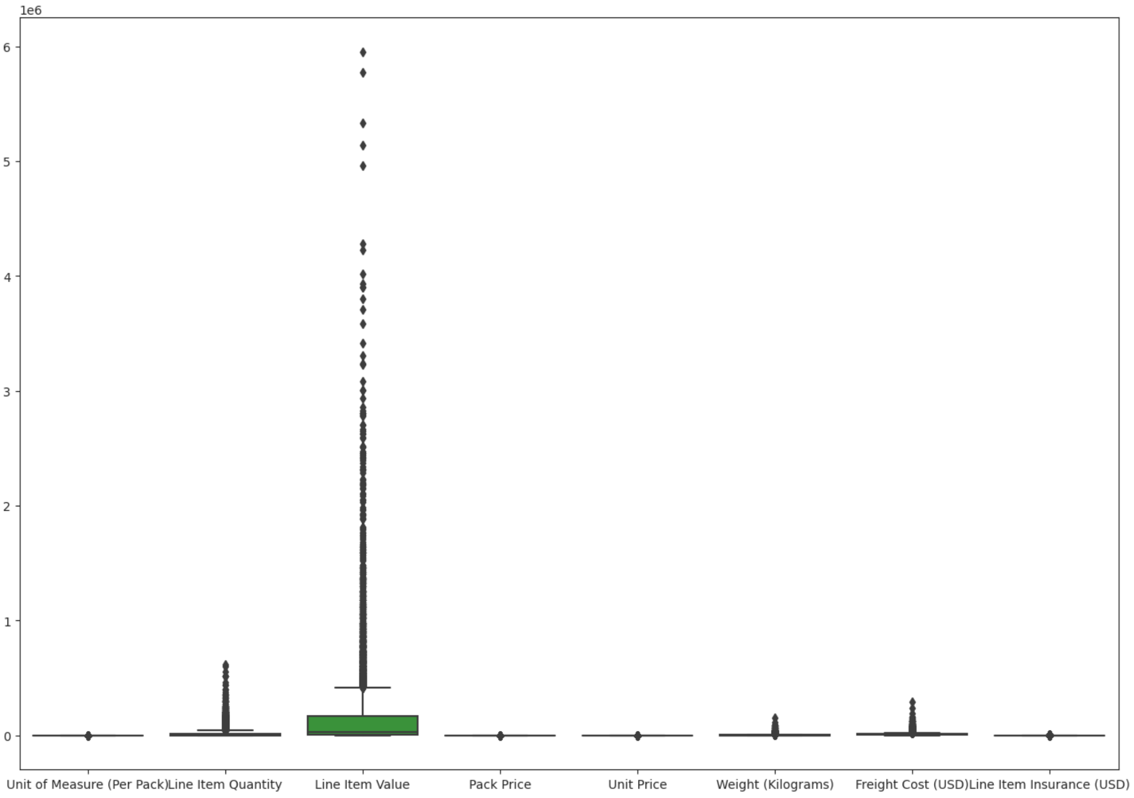
Out[86]:

```
Unit of Measure (Per Pack)       60.000000
Line Item Quantity             3056.000000
Line Item Value               30448.400000
Pack Price                        8.820000
Unit Price                        0.160000
Weight (Kilograms)             1454.000000
Freight Cost (USD)            11103.234819
Line Item Insurance (USD)        47.110000
dtype: float64
```

Box Plot

```
In [ ]: plt.figure(figsize=(16,11))
        sns.boxplot(data=df , orient = "v")
```

```
Out[87]: <Axes: >
```



- it is indicated the outliers in Line Item Quantity , Line Item Value , Weight(kilograme) , Freight Cost (USD)

Handle outliers

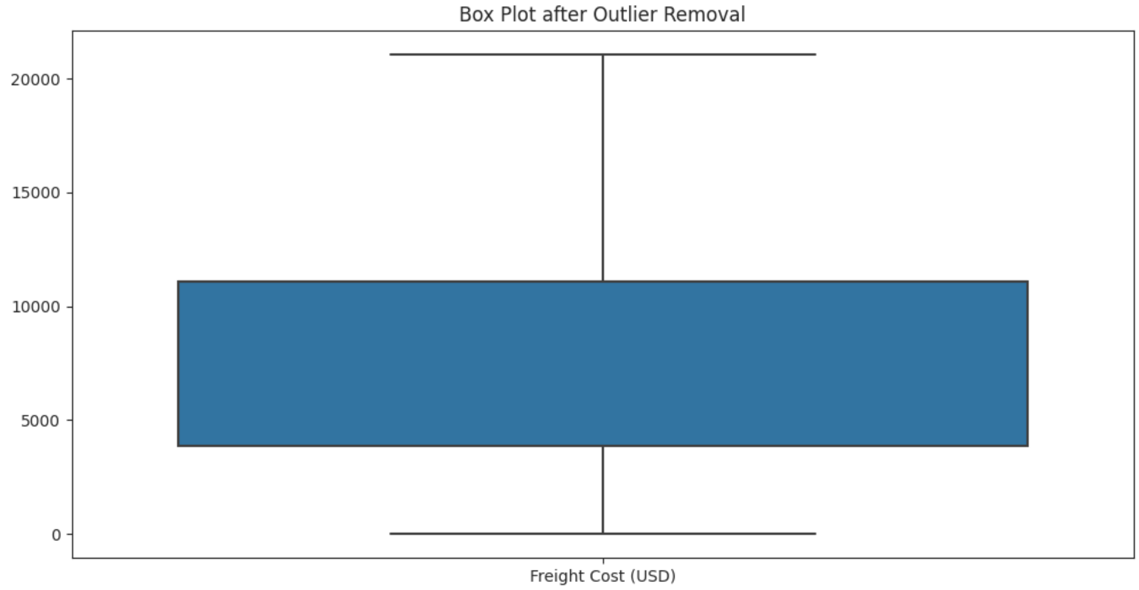
```
In [ ]: # Identify outliers using the IQR method
        Q1 = df['Freight Cost (USD)'].quantile(0.25)
        Q3 = df['Freight Cost (USD)'].quantile(0.75)
        IQR = Q3 - Q1
        lower_threshold = Q1 - 1.5 * IQR
        upper_threshold = Q3 + 1.5 * IQR
```

```
In [ ]: # Remove outliers
        data_no_outliers = df[(df['Freight Cost (USD)'] >= lower_threshold) & (df['
```

```
In [ ]: #Cap and floor outliers
        data_capped_floored = df.copy()
        data_capped_floored['Freight Cost (USD)'] = data_capped_floored['Freight Co
```

In [ ]:

```
# Visualize the distribution after outlier handling
plt.figure(figsize=(12, 6))
sns.boxplot(data=data_no_outliers[['Freight Cost (USD)']])
plt.title('Box Plot after Outlier Removal')
plt.show()
```



#After performing EDA and handling outliers now processing dataset for Machine learning model

#Train -Test split

In [ ]:

```
from sklearn.model_selection import train_test_split

X = df.drop('Freight Cost (USD)', axis=1)  # Features
y = df['Freight Cost (USD)']  # Target variable

# Split the data into training, validation, and test sets
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.3, ra
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0
```

#Encode Categorical Features

In [ ]:

```
from sklearn.preprocessing import OneHotEncoder

encoder = OneHotEncoder(sparse=False, drop='first')  # drop='first' avoids
country_encoded = encoder.fit_transform(X_train[['Country']])
```

In [ ]:

```
from sklearn.preprocessing import LabelEncoder

encoder = LabelEncoder()
country_encoded = encoder.fit_transform(X_train['Country'])
```

#Data Scaling

In [ ]:

```
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.linear_model import LinearRegression
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline

# Define preprocessing steps for numeric and categorical features
numeric_features = ['Line Item Quantity', 'Line Item Value', 'Pack Price',
categorical_features = ['Country', 'Vendor', 'Product Group']

# Create transformers
numeric_transformer = StandardScaler()
categorical_transformer = Pipeline(steps=[
    ('onehot', OneHotEncoder(sparse=False, drop='first'))
])

preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_features),
        ('cat', categorical_transformer, categorical_features)
    ])

# Create a pipeline with preprocessing and modeling steps
pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                           ('model', LinearRegression())])

# Fit the pipeline to your training data
pipeline.fit(X_train, y_train)
```

Out[109]:

```
Pipeline(steps=[('preprocessor',
                 ColumnTransformer(transformers=[('num', StandardScaler(),
                                                  ['Line Item Quantity',
                                                   'Line Item Value',
                                                   'Pack Price',
                                                   'Weight (Kilograms)']),
                                                 ('cat',
                                                  Pipeline(steps=[('oneho
t',
                                                                   OneHotE
ncoder(drop='first',

sparse=False))]),
                                                  ['Country', 'Vendor',
                                                   'Product Group'])])),
                ('model', LinearRegression())])
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

#Explanation: We set up a machine learning pipeline for building a linear regression model to predict shipment costs based on our dataset. It includes data preprocessing steps like feature scaling and one-hot encoding for both numeric and categorical features.