DATA ANALYSIS

SUPPLY CHAIN

PROJECT

وزارة الاتصـــــالات وتكنولوجيا المعلومات

Digital Egypt Pioneers

مبــادرة رواد مصــر الرقميـــة

Presented By:

Amal Soliman
Aya Gomaa
Esraa Mahmoud
Sara Gamal
Abd elwahab Abo Elkasem
Ahmed Ellabban

PRESENTATION TABLE OF CONTENTS

1

Project Overview & Data Understanding

Digital Egypt Pioneers

وزارة الاتصالات
وتكنولوجيا المعلومات

DATA ANALYSIS – SUPPLY CHAIN PROJECT

PROJECT OVERVIEW

USAID Global Health Supply Chain Program

The USAID Global Health Supply Chain Program (GHSC) is a collection of 8 complementary projects working globally to achieve stronger, more resilient health supply chains.

Data Source :

https://www.kaggle.com/datasets/divyeshardeshana/supply-chain-shipment-pricing-data

PROJECT OVERVIEW

Digital Egypt Pioneers

وزارة الاتصـــــالات وتكنولوجيا المعلومات

This project involves the analysis of **Global Health Supply Chain** data to track procurement, delivery, and logistics performance for various **pharmaceutical and diagnostic products** across multiple countries.

The data consists of 10324 rows and 33 columns.
The dataset comprises a mix of categorical, numerical, and date columns.

PROJECT OVERVIEW

Time Interval

2015

↑

2006

142 Projects

5 Products

73 Vendors

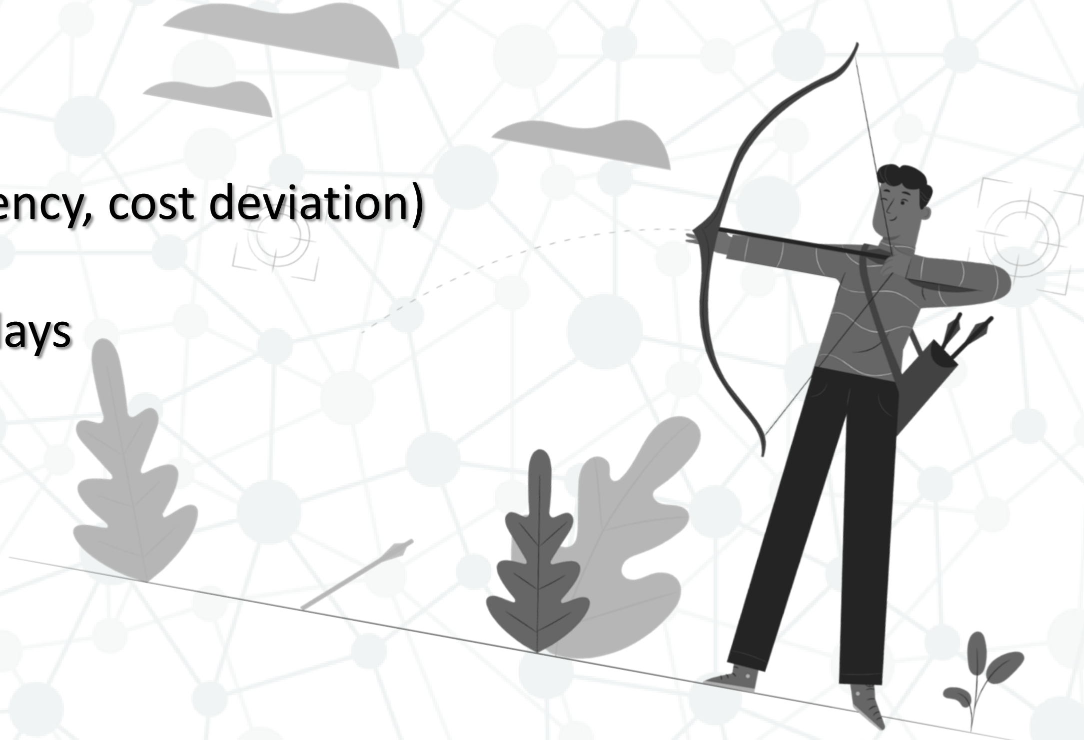
43 Countries

4 Shipments

OBJECTIVE

To Uncover operational inefficiencies.

- Delivery Delays: Analyze late shipments by vendor/product/region
- Cost Overruns: High-cost shipments vs. weight
- Vendor Issues: Performance scores (on-time %, cost efficiency, cost deviation)
- Product Bottlenecks: Items with recurring problems or delays



2

Data Cleaning & Transformation

Digital Egypt Pioneers

وزارة الاتصـــــالات وتكنولوجيا المعلومات

DATA ANALYSIS – SUPPLY CHAIN PROJECT

IMPORT DATA

Import Libraries: pandas, numpy, matplotlib, seaborn

```python
#Import Packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Load Data: Read dataset (XLS)

```python
# importing supply chain data
df= pd.read_excel(r"D:\aml\project\last choice\Supply_Chain_Shipment_Pricing.xlsx")
```

Generate + Code + Markdown

OVERVIEW

Quick Overview: .head(), .info(), .shape()

```
# Strating EDA
df.shape
```

```
(10324, 33)
```

```
df.columns
```

```
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', 'dosage',
       'dosage form', 'unit of measure (per pack)', 'line item quantity',
       'line item value', 'pack price', 'unit price', 'manufacturing site',
       'first line designation', 'weight (kilograms)', 'freight cost (usd)',
       'line item insurance (usd)'],
      dtype='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  datetime64[ns]
 13  delivered to client date      10324 non-null  datetime64[ns]
 14  delivery recorded date        10324 non-null  datetime64[ns]
 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  bool
 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: bool(1), datetime64[ns](3), float64(4), int64(2), object(23)
```

DATA CLEANING

وزارة الاتـــــــصالات
وتكنولوجيا المعلومات

Digital Egypt Pioneers

Handle missing values, fix types

Data Cleaning Summary Plan

Missing Values:

- `Shipment Mode: ~360 missing → consider imputation or flagging`
- `Dosage: Nulls expected for HIV kits → keep as-is; filter when needed >>>>> df_filtered = df[df['dosage'].notnull()]`
- `Line Item Insurance (USD): Review based on analysis needs`

Data Type Fixes:

- `Dates (object → datetime): PQ First Sent to Client Date, PO Sent to Vendor Date`
- `Numerics (object → numeric): Weight (Kilograms), Freight Cost (USD)`

DATA CLEANING

Digital Egypt Pioneers

وزارة الاتصالات وتكنولوجيا المعلومات

Handle missing values

```
#make a copy of the original data before processing
#Now sc will be processed

sc =df.copy()
```

```
#replaces all NaN (missing) values in the shipment mode column with the string 'Missing'
#To make sure no value is NaN, especially before grouping, counting, or visualizing
sc['shipment mode'] = sc['shipment mode'].fillna('Missing')
```

```
# the most common shipment mode for each vendor
mode_mapping = sc.dropna().groupby('vendor')['shipment mode'].agg(lambda x: x.mode()[0]).to_dict()

# Fill missing shipment modes using the mapping
sc['shipment mode'] = sc.apply(lambda row: mode_mapping.get(row['vendor'], 'Unknown') if pd.isna(row['shipment mode
sc['shipment mode'].isnull().sum()
```

```
np.int64(0)
```

```
# So, we will fill the missing values with the most common shipment mode in the dataset.
# Replace "Unknown" shipment modes with the most common mode

most_common_shipment_mode =sc["shipment mode"].mode()[0]

sc["shipment mode"] = sc["shipment mode"].replace(["Missing", None], most_common_shipment_mode)

# Verify that "Unknown" is now gone
print(sc["shipment mode"].value_counts())
```

```
shipment mode
Air            6473
Truck          2830
Air Charter     650
Ocean           371
Name: count, dtype: int64
```

DATA CLEANING

fix types

```
#2 numerical columns are of object type ( weight (kilograms),freight cost (usd) )
# change to numeric and convert text values by Nan
sc['weight (kilograms)'] = pd.to_numeric(sc['weight (kilograms)'], errors='coerce')
sc['freight cost (usd)'] = pd.to_numeric(sc['freight cost (usd)'], errors='coerce')
sc[['freight cost (usd)', 'weight (kilograms)']].head(10)
```

	freight cost (usd)	weight (kilograms)
0	780.34	13.0
1	4521.50	358.0
2	1653.78	171.0

```
def convert_to_datetime(df, columns, date_format="%Y-%m-%d"):
    for col in columns:
        df[col] = pd.to_datetime(df[col], format=date_format, errors='coerce')
    return df
```

```
date_cols = [
    "pq first sent to client date",
    "po sent to vendor date",
    "scheduled delivery date",
    "delivered to client date",
    "delivery recorded date"
]

sc = convert_to_datetime(sc, date_cols)
```

DATA CLEANING

Profile Data: Duplicates, unique values, value counts

Data Profiling & validation : Checking for Duplicates, Unique Values, Accuracy, and co

```
missing_values = sc.isnull().sum()
missing_values
```

```
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                   0
pq first sent to client date 2681
po sent to vendor date       1931
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
...
freight cost (usd)_original     0
freight_included_flag           0
freight_cost_imputed            0
weight_per_unit              3952
dtype: int64
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```
### 1. Checking for Duplicates
duplicate= sc_cleaned.duplicated().sum()
print(f"Number od duplicate roes: {duplicate}")
```

```
Number od duplicate roes: 0
```

```
sc_cleaned["country"].unique()
```

```
array(["CÃ´TE D'IVOIRE", 'VIETNAM', 'NIGERIA', 'ZAMBIA', 'TANZANIA',
       'RWANDA', 'HAITI', 'ZIMBABWE', 'ETHIOPIA', 'SOUTH AFRICA',
       'GUYANA', 'NAMIBIA', 'BOTSWANA', 'MOZAMBIQUE', 'KENYA',
       'KAZAKHSTAN', 'UGANDA', 'KYRGYZSTAN', 'SENEGAL', 'BENIN',
       'LESOTHO', 'PAKISTAN', 'SWAZILAND', 'GHANA', 'ANGOLA', 'LEBANON',
       'SIERRA LEONE', 'CAMEROON', 'SOUTH SUDAN', 'BURUNDI',
       'DOMINICAN REPUBLIC', 'MALAWI', 'CONGO, DRC', 'SUDAN', 'MALI',
       'GUATEMALA', 'TOGO', 'AFGHANISTAN', 'LIBERIA', 'BURKINA FASO',
       'GUINEA', 'LIBYA', 'BELIZE'], dtype=object)
```

```
sc_cleaned["country"] = sc_cleaned["country"].replace("Cã▯Te D'Ivoire", "Côte d'Ivoire")
```

```
sc_cleaned["country"].unique()
```

```
array(["Côte d'Ivoire", 'Vietnam', 'Nigeria', 'Zambia', 'Tanzania',
       'Rwanda', 'Haiti', 'Zimbabwe', 'Ethiopia', 'South Africa',
```

DATA CLEANING

Finally the data is ready for analysis

```
# Save the cleaned data to a new CSV file (USE encoding ="utf-8-sig" to avoid issues with special characters)
sc_cleaned.to_csv(r"D:\aml\project\last choice\NEW\cleaned_data.csv", index=False, encoding="utf-8-sig")
```

3

Exploratory Data Analysis

Digital Egypt Pioneers

وزارة الاتصـــــالات
وتكنولوجيا المعلومات

DATA ANALYSIS – SUPPLY CHAIN PROJECT

EXPLORATORY DATA ANALYSIS

Import Libraries-Read Dataset

```
# Step 1: Importing necessary libraries
# Pandas and NumPy are used for data manipulation.
# Matplotlib and Seaborn are used for data visualization.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[ ] # Step 2: Reading the supply chain dataset into a DataFrame.
    # This step is essential to load the data into memory for further exploration.
    # Read the dataset
    df = pd.read_csv('/content/Supply_chain_cleaned_data.csv')
```

```
# Step 3: Displaying the first five rows of the dataset.
# This preview helps to confirm that the data was loaded correctly and gives an initial look at the structure.
df.head()
```

	id	project code	pq #	po / so #	asn/dn #	country	managed by	fulfill via	vendor inco term	shipment mode	...	manufacturing site	first line designation	weight (kilograms)	freight cost (usd)	line item insurance (usd)
0	1	100-CI-T01	Pre-PQ Process	SCMS-4	ASN-8	Côte d'Ivoire	PMO - US	Direct Drop	EXW	Air	...	Ranbaxy Fine Chemicals LTD	True	13.0	780.34	38.45
1	3	108-VN-T01	Pre-PQ Process	SCMS-13	ASN-85	Vietnam	PMO - US	Direct Drop	EXW	Air	...	Aurobindo Unit III, India	True	358.0	4521.50	38.45
2	4	100-CI-T01	Pre-PQ Process	SCMS-20	ASN-14	Côte d'Ivoire	PMO - US	Direct Drop	FCA	Air	...	ABBVIE GmbH & Co.KG Wiesbaden	True	171.0	1653.78	38.45
3	15	108-VN-T01	Pre-PQ Process	SCMS-78	ASN-50	Vietnam	PMO - US	Direct Drop	EXW	Air	...	Ranbaxy, Paonta Shahib, India	True	1855.0	16007.06	38.45
4	16	108-VN-T01	Pre-PQ Process	SCMS-81	ASN-55	Vietnam	PMO - US	Direct Drop	EXW	Air	...	Aurobindo Unit III, India	True	7590.0	45450.08	38.45

Check for Duplicates-Explore Unique Values in Categorical Columns

```
# Detect and count duplicate rows in the dataset
df.duplicated().sum()
```

```
np.int64(0)
```

Result:

The `df.duplicated().sum()` function was used to check for duplicated rows in the dataset.

- The result shows **0 duplicated rows.**
- This confirms that the dataset is clean and free of duplicate entries, which is essential before proceeding to the analysis stage.

```
# Display unique values for selected categorical columns
for col in df.select_dtypes(include='object').columns:
    print(f"Unique values in {col}:")
    print(df[col].unique())
    print('-'*50)
```

Show hidden output

Result:

This loop was used to explore the **unique values** of all categorical columns.

- The output provided an overview of the distinct values in each text-based column.
- This step helps in understanding the structure, detecting unexpected or inconsistent entries, and planning for future encoding or grouping in the analysis phase.

EXPLORATORY DATA ANALYSIS

Summary Statistics

```
# Generate descriptive statistics for numerical columns
df.describe()
```

	id	unit of measure (per pack)	line item quantity	line item value	pack price	unit price	weight (kilograms)	freight cost (usd)	line item insurance (usd)	pq_to_po_days	weight_per_unit
count	10324.000000	10324.000000	10324.000000	1.032400e+04	10324.000000	10324.000000	10324.000000	8870.000000	10324.000000	7643.000000	6372.000000
mean	51098.968229	77.990895	18332.534870	1.576506e+05	21.910241	0.611701	2699.627517	11673.565415	234.582169	39.704697	4.671812
std	31944.332496	76.579764	40035.302961	3.452921e+05	45.609223	3.275808	10937.883547	23341.216397	494.270775	65.393815	121.268278
min	1.000000	1.000000	1.000000	0.000000e+00	0.000000	0.000000	0.000000	0.000000	0.000000	-224.000000	0.000000
25%	12795.750000	30.000000	408.000000	4.314593e+03	4.120000	0.080000	75.000000	1570.462958	7.147500	24.000000	0.088684
50%	57540.500000	60.000000	3000.000000	3.047147e+04	9.300000	0.160000	575.135144	4957.905000	42.815000	24.000000	0.140542
75%	83648.250000	90.000000	17039.750000	1.664471e+05	23.592500	0.470000	2523.000000	13583.888640	241.750000	24.500000	0.367643
max	86823.000000	1000.000000	619999.000000	5.951990e+06	1345.640000	238.650000	857354.000000	826058.981376	7708.440000	414.000000	8529.750000

High variability: Metrics like line item quantity, value, weight, and freight cost show large spreads → expected due to commodity diversity, including RDC items.

Zeros in key fields:

- Freight cost = 0 is valid because cost is bundled into commodity price or RDC shipments.
- Unit price = 0 could result from low-value items priced via pack price.
- Insurance = 0 may relate to Incoterms or RDC transactions.

Negative values in pq_to_po_days:

Likely due to RDC lead time setups or system timestamps.

Missing data: Fields like freight cost and pq_to_po_days have nulls —>> RDC transactions.

Outliers: Large values (e.g., weight > 800,000 kg) are acceptable for specific bulk commodities.

EXPLORATORY DATA ANALYSIS

Digital Egypt Pioneers

وزارة الاتصـــــــالات وتكنولوجيا المعلومات

Summary Statistics

```
df.describe( include= object)
```

	project code	pq #	po / so #	asn/dn #	country	managed by	fulfill via	vendor inco term	shipment mode	pq first sent to client date	...	product group	sub classification	vendor	item description	molecule/test type	brand	dosage	dosage form
count	10324	10324	10324	10324	10324	10324	10324	10324	10324	7643	...	10324	10324	10324	10324	10324	10324	8588	10324
unique	142	1237	6233	7030	43	4	2	8	4	763	...	5	6	73	184	86	48	54	17
top	116-ZA-T30	Pre-PQ Process	SCMS-199289	ASN-19166	South Africa	PMO - US	From RDC	N/A - FROM RDC	Air	2014-09-11	...	ARV	Adult	Scms From Rdc	Efavirenz 600mg, tablets, 30 Tabs	Efavirenz	Generic	300mg	Tablet
freq	768	2681	67	54	1406	10265	5404	5404	6473	205	...	8550	6595	5404	755	1125	7285	990	3532

4 rows × 24 columns

Top 10 Most Frequent Categorical Values

```python
# Display the top 10 most frequent values for selected categorical columns
categorical_cols = ['country', 'managed by', 'shipment mode', 'vendor inco term',
                    'product group', 'sub classification', 'vendor', 'brand', 'dosage form']

for col in categorical_cols:
    print(f"\nColumn: {col}")
    print(df[col].value_counts().head(10))
```

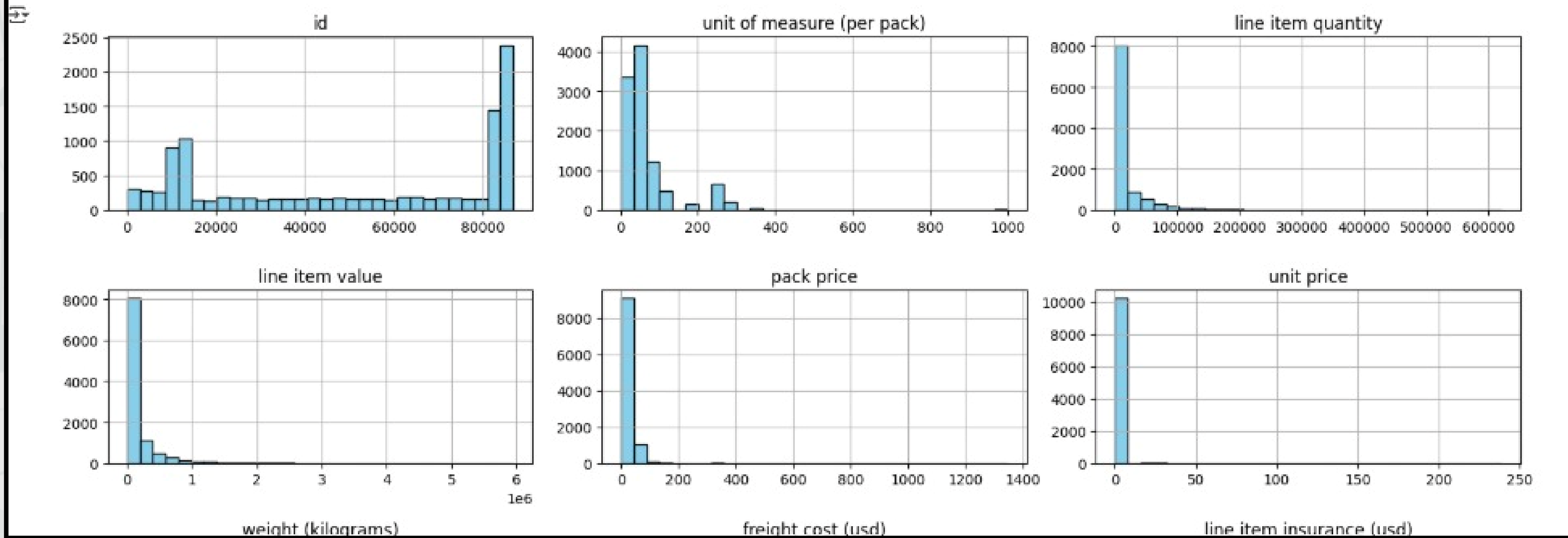
```
ARV     8550
HRDT    1728
ANTM      22
ACT       16
MRDT       8
Name: count, dtype: int64

Column: sub classification
sub classification
Adult                   6595
Pediatric               1955
HIV test                1567
HIV test - Ancillary     161
Malaria                   30
ACT                       16
Name: count, dtype: int64
```

EXPLORATORY DATA ANALYSIS

Plotting Distribution of Numerical Features

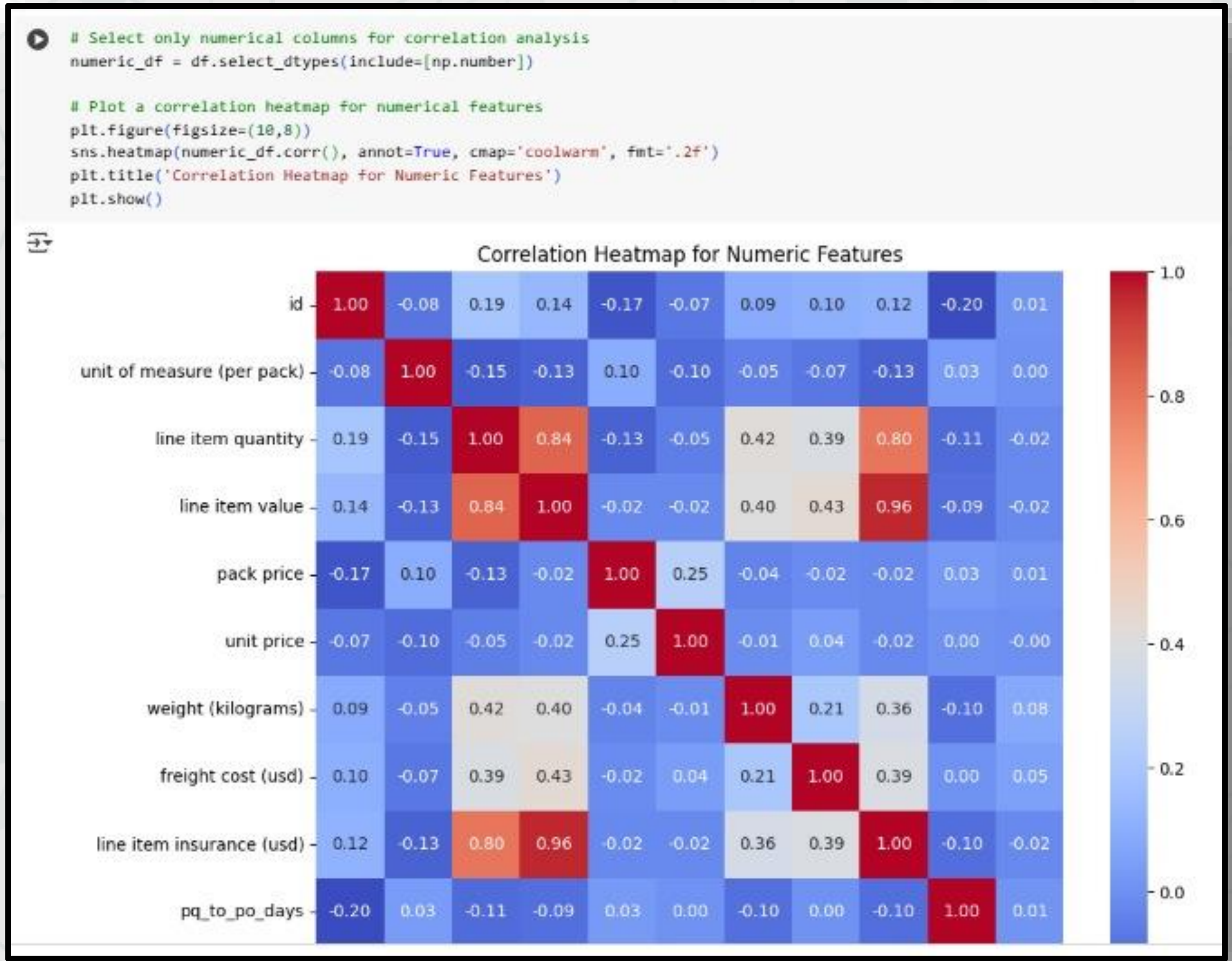
The histograms reveal that most variables are highly skewed to the right (positively skewed)

```
# Plot histograms for all numerical columns to visualize distribution
df.hist(bins=30, figsize=(15,10), color='skyblue', edgecolor='black')
plt.tight_layout()
plt.show()
```



EXPLORATORY DATA ANALYSIS

Visualizing Correlation Between Numeric Variables



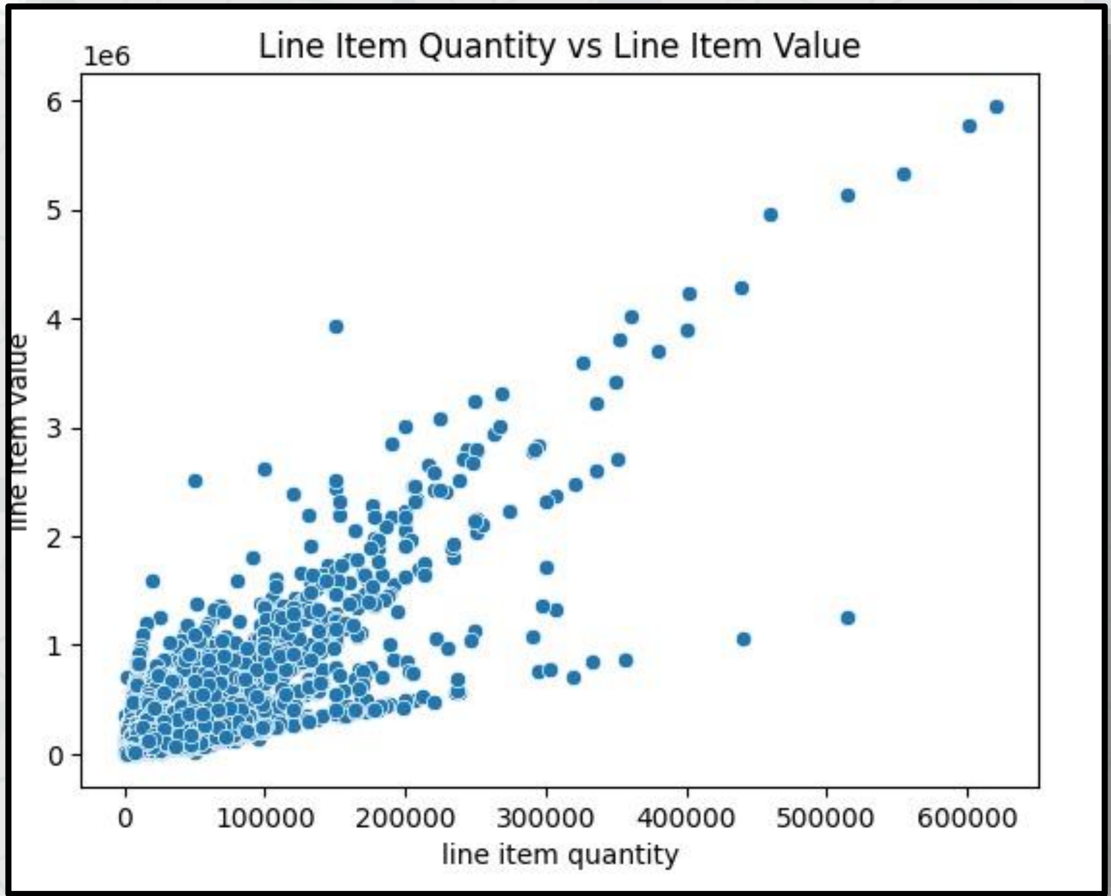
- line item quantity & line item value (0.82) as expected, **larger** quantities often increase the total value.
- pack price & unit price (0.49) **moderate** relationship due to unit scaling.

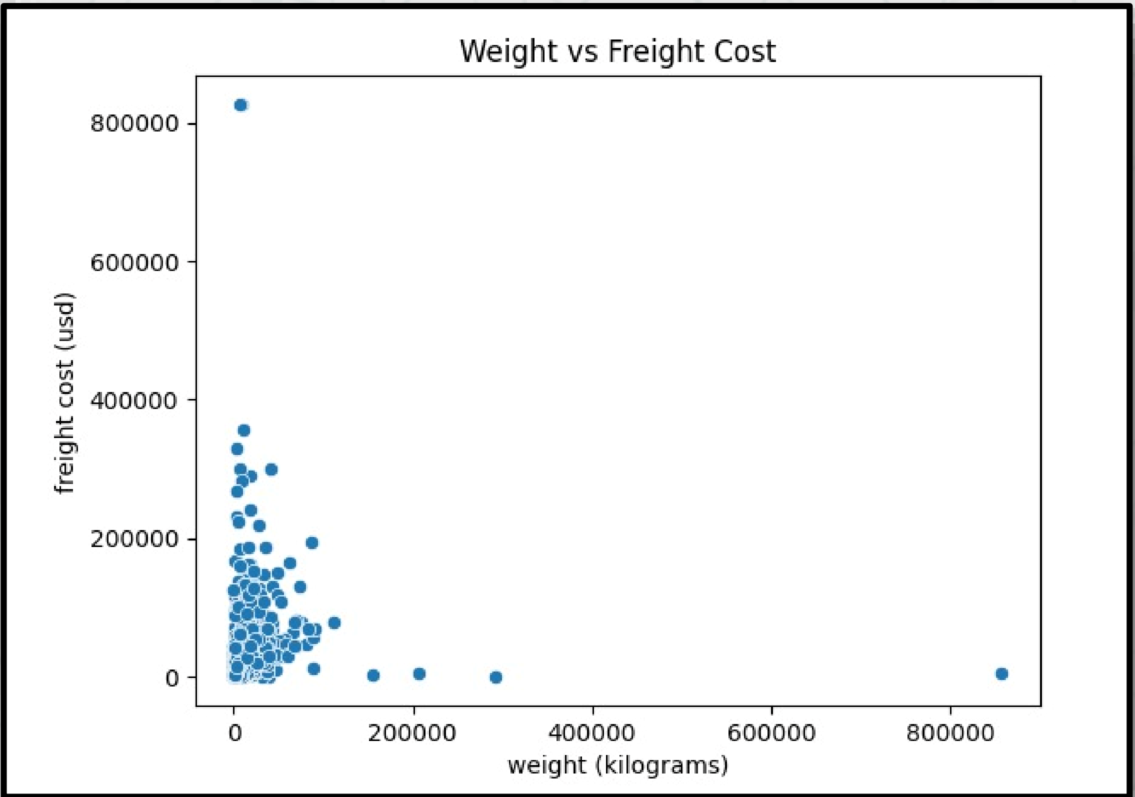
EXPLORATORY DATA ANALYSIS

وزارة الاتصالات وتكنولوجيا المعلومات

Digital Egypt Pioneers

Scatter Plot for Relationship Exploration





DATA ANALYSIS – SUPPLY CHAIN PROJECT

4

Visualization

DATA ANALYSIS – SUPPLY CHAIN PROJECT

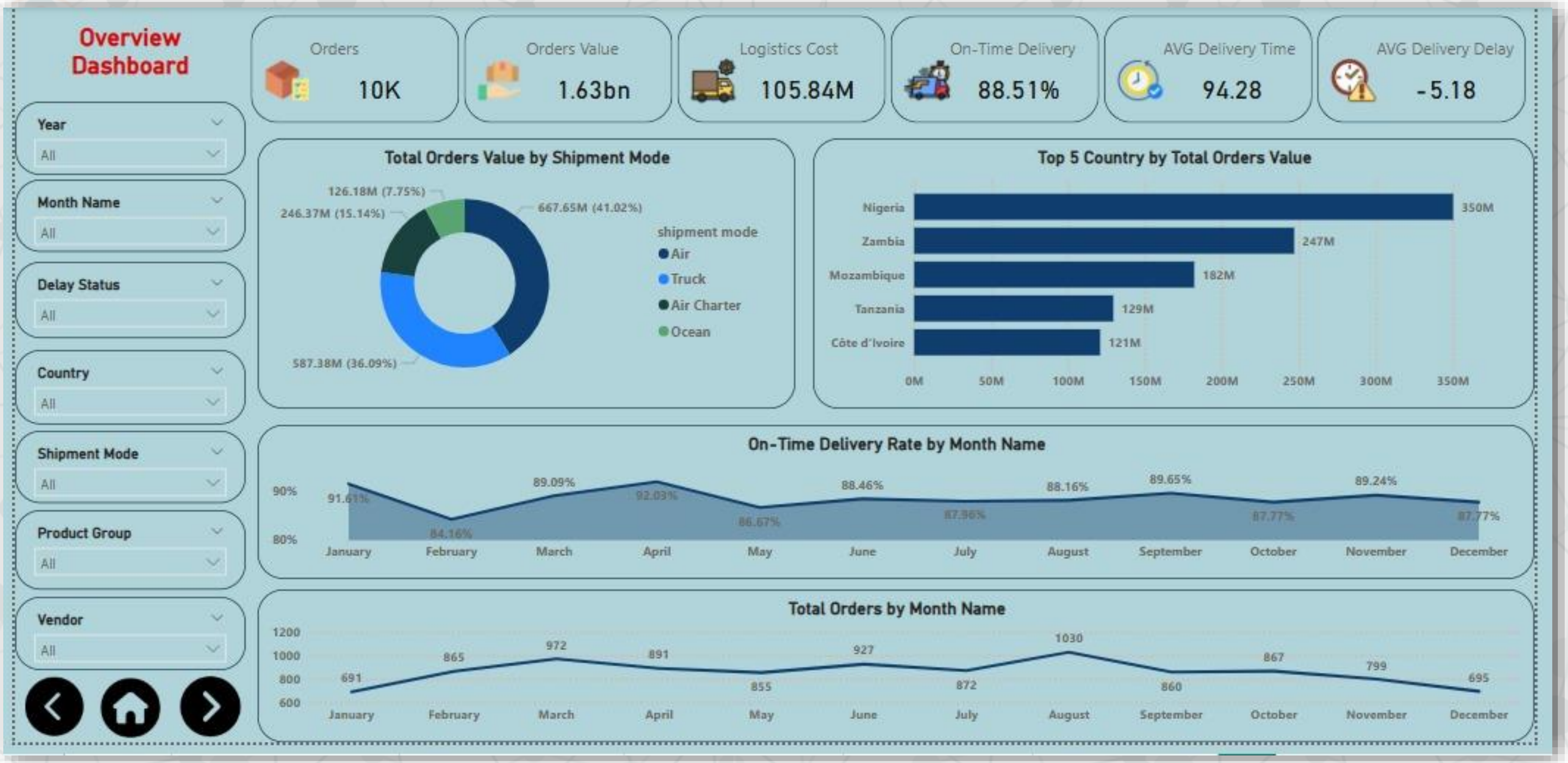
VISUALIZATION

Digital Egypt Pioneers

وزارة الاتصـــالات وتكنولوجيا المعلومات

Power BI

VISUALIZATION

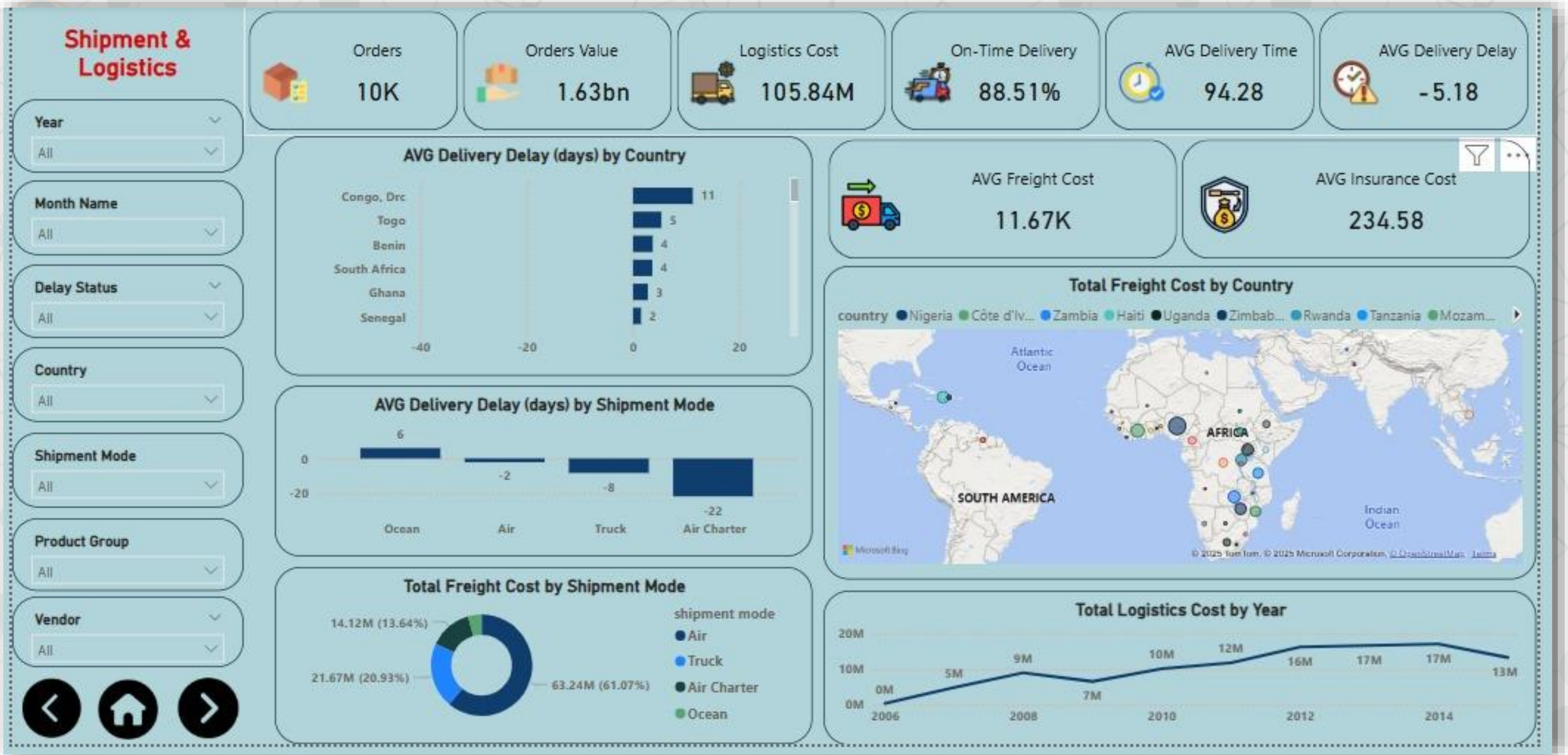


DATA ANALYSIS – SUPPLY CHAIN PROJECT

VISUALIZATION

Digital Egypt Pioneers

وزارة الاتصـــــالات وتكنولوجيا المعلومات

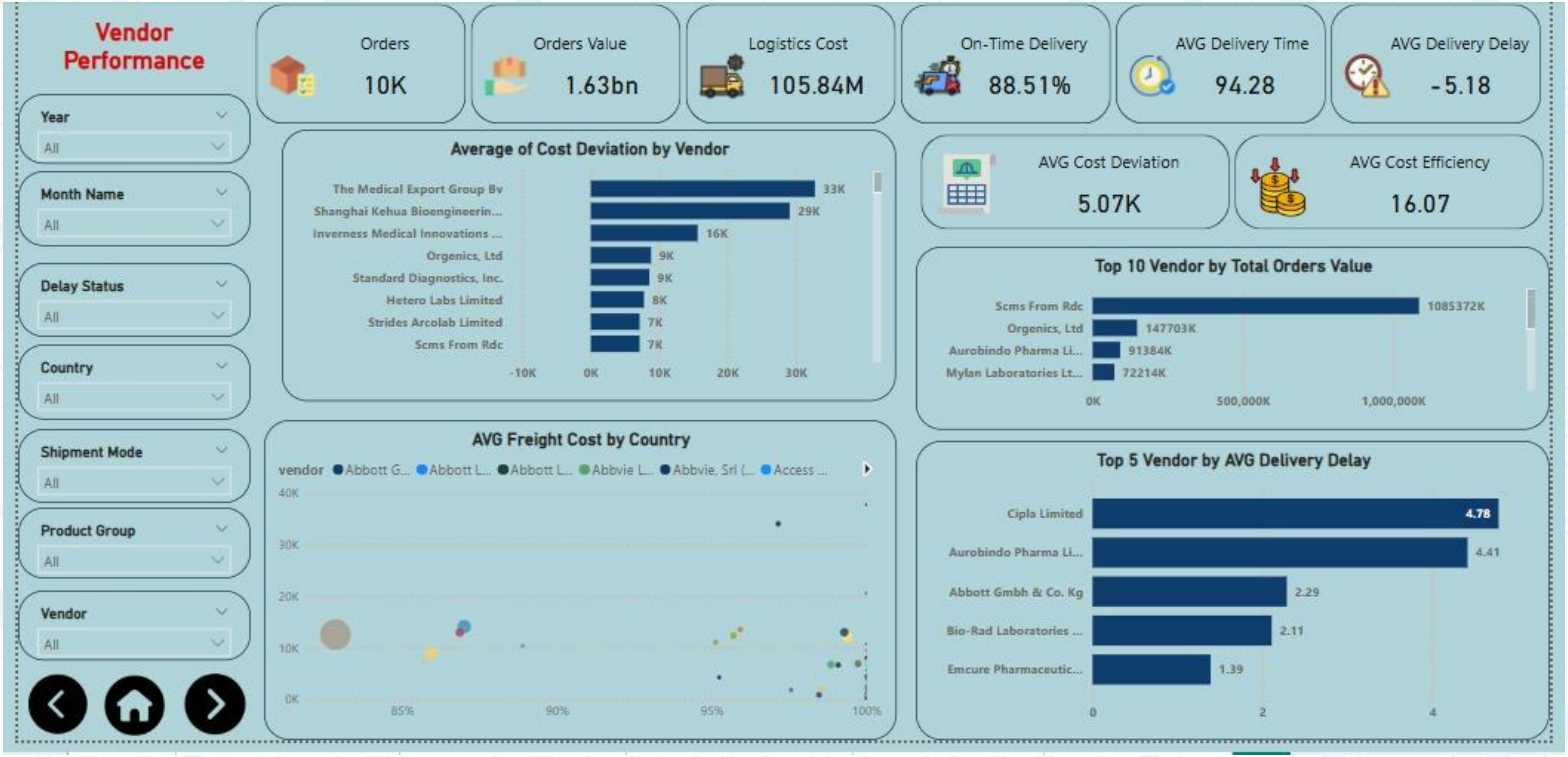


DATA ANALYSIS – SUPPLY CHAIN PROJECT

VISUALIZATION

Digital Egypt Pioneers

وزارة الاتصـــالات
وتكنولوجيا المعلومات

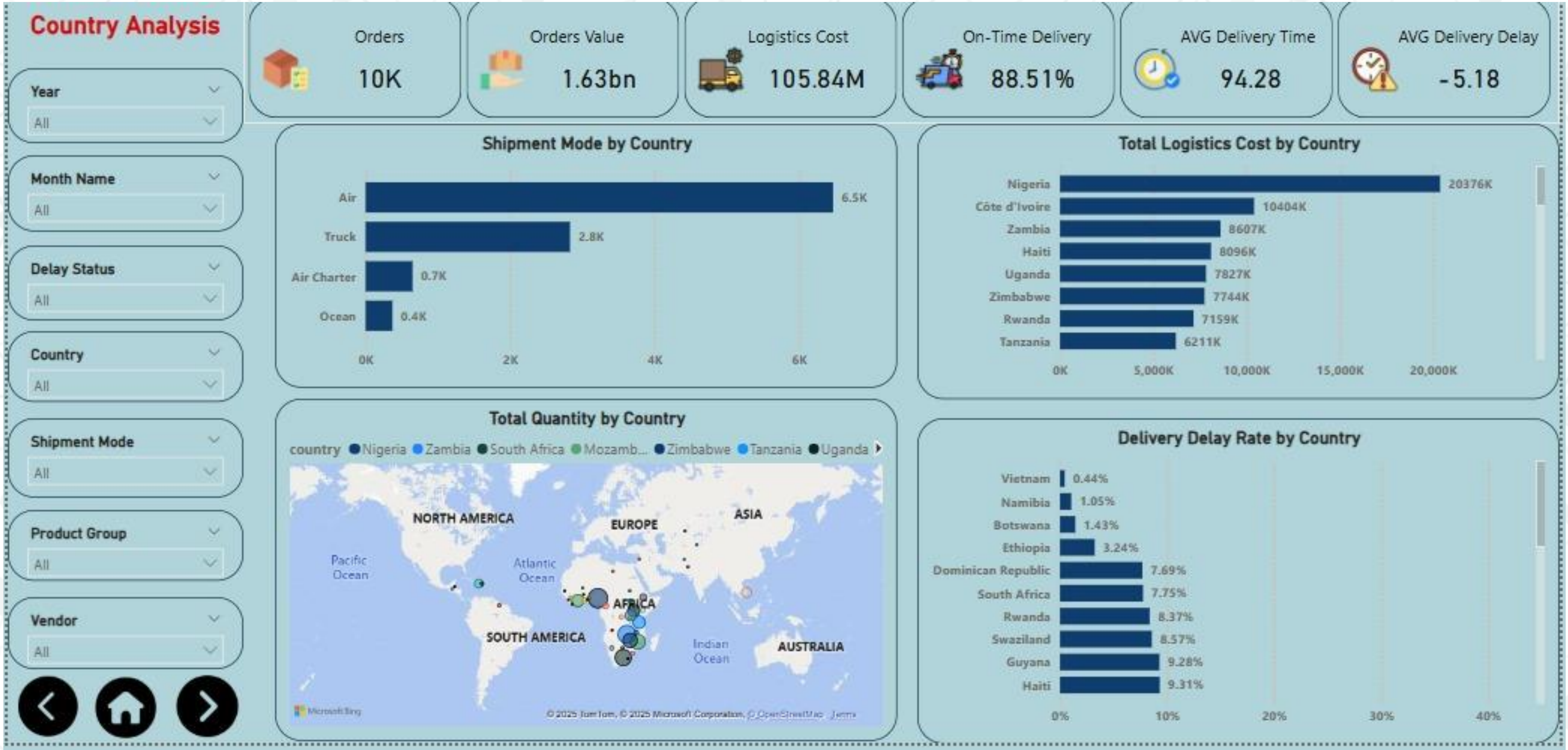


DATA ANALYSIS – SUPPLY CHAIN PROJECT

VISUALIZATION

Digital Egypt Pioneers

وزارة الاتصالات
وتكنولوجيا المعلومات



DATA ANALYSIS – SUPPLY CHAIN PROJECT

VISUALIZATION

Digital Egypt Pioneers

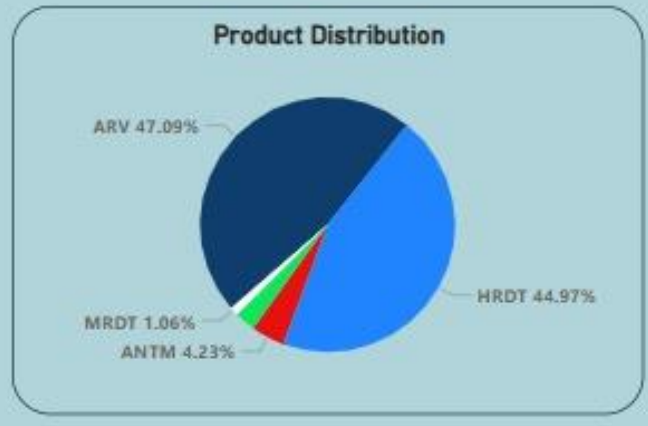
وزارة الاتصـــــالات
وتكنولوجيا المعلومات

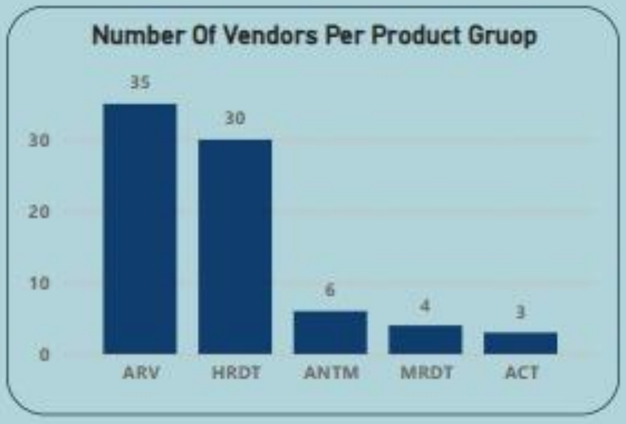
DATA ANALYSIS – SUPPLY CHAIN PROJECT

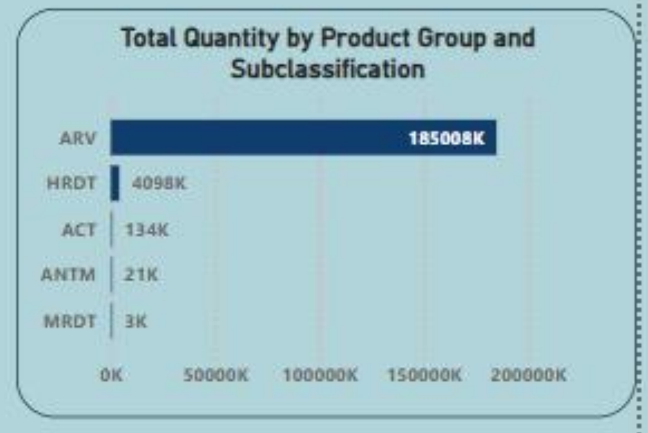
Product Insights

Year: All
Delay Status: All
Freight Included: All
Country: All
Shipment Mode: All
Product Group: All
Vendor: All

Orders	Orders Value	Logistics Cost	On-Time Delivery	AVG Delivery Time	AVG Delivery Delay
10K	1.63bn	105.84M	88.51%	94.28	-5.18







product group	sub classification	Total Orders Value	Total Quantity	Average of Cost Deviation	On-Time Delivery Rate	AVG Delivery Lead Time (days)	AVG Cost Efficiency
ACT	ACT	664,380.94	134286	4,066.85	100.00%	35.63	0.67
ANTM	Malaria	274,176.56	21023	2,978.34	100.00%	76.86	0.28
ARV	Adult	1,360,122,113.46	172279581	6,590.13	87.43%	102.19	6.34
ARV	Pediatric	53,097,959.42	12728808	-594.51	87.52%	101.71	17.31
HRDT	HIV test	212,961,651.72	3928054	6,695.07	93.24%	73.75	0.41
HRDT	HIV test - Ancillary	383,110.09	170360	-3,297.93	95.65%	75.99	516.52
MRDT	Malaria	81,065.10	2978	-3,916.80	100.00%	13.00	0.53
Total		**1,627,584,457.29**	**189265090**	**5,071.59**	**88.51%**	**97.15**	**16.07**

VISUALIZATION

Power BI

5

Key Insights & Conclusion

Digital Egypt Pioneers

وزارة الاتصـــــالات
وتكنولوجيا المعلومات

DATA ANALYSIS – SUPPLY CHAIN PROJECT

KEY INSIGHTS

Shipping Costs Vary Significantly by Country

Countries like *Nigeria*, *Zambia*, and *Côte d'Ivoire* tend to have the highest freight costs, possibly due to distance, logistics infrastructure, or vendor preferences.

Air Freight is Faster but Much More Expensive

Air shipments significantly reduce delivery time compared to ocean freight, but they come at a much higher cost. This method is best suited for urgent or sensitive medical supplies.

Supplier Performance Shows Clear Variation

Some vendors, such as *Aspen Pharmacare*, have consistently high on-time delivery performance, while others show frequent delays.

Wide Differences in Insurance and Shipping Charges Among Suppliers

Shipping and insurance costs vary even for similar products, indicating a need for cost benchmarking and negotiation.

CONCLUSION

Identify Gaps

We discovered major differences in shipping costs, delivery times, and supplier performance across countries and vendors.

Understand the Impact

These inefficiencies can lead to higher expenses, delays in delivery, and risk of stockouts for critical health commodities.

Learn from the Data

Reliable vendors and efficient freight methods (like ocean shipping for non-urgent goods) can significantly improve performance and cut costs.

Act Strategically

With data-driven decisions, we can optimize procurement, improve supplier selection, and enhance logistics planning.

Deliver Better Results

By following this approach, organizations can ensure faster, more cost-effective, and reliable delivery of life-saving health products where they’re needed most.

RECOMMENDATION

وزارة الاتصــــــالات
وتكنولوجيا المعلومات

Digital Egypt Pioneers

Work with High-Performing Suppliers

Focus on vendors with proven reliability and timely deliveries to reduce delays and disruptions.

Use Ocean Freight When Time Allows

To minimize costs, ocean shipping should be prioritized for non-urgent deliveries.

RECOMMENDATION

Implement a Supplier Rating System

Use KPIs like delivery time, shipping cost, and quality to evaluate and select suppliers.

Regularly Analyze Country-Specific Shipping Costs

This can uncover inefficiencies and help renegotiate rates or explore alternative logistics options.

Invest in Local Supply Chain Infrastructure

Improve warehousing and transportation networks in recipient countries to shorten delivery times and reduce reliance on expensive shipping methods.

THANK YOU

Digital Egypt Pioneers

وزارة الاتصــــــالات
وتكنولوجيا المعلومات

DATA ANALYSIS – SUPPLY CHAIN PROJECT