Supply Chain

MAFA TEAM Alaa Oraby -Asmaa Gamil- Mariam Ebrahem- Farah Mohamed

Under The supervision of: Eng Ahmed Radwan

DEPI- Final Project

Supply Chain

The project aims to analyze shipping performance, customer purchasing patterns, and profitability,

SUPPLY CHAIN

OVERVIEW

PROBLEM

SCOPE OF DATA

ENTITY RELATIONSHIP DIAGRAM

ANALYZING DATA

RECOMMENDATION

SUPPLY CHAIN

PROBLEM

+180.519,54

Rows, Columns

This project focuses on analyzing various aspects of the supply chain to assess performance and optimize operations. Key metrics include shipping days (real vs. scheduled), benefit per order, sales per customer, delivery status, and late delivery risk. The data also covers customer demographics, product details, and order performance, providing insights into logistics efficiency, customer behavior, and profitability. The goal is to identify areas for improvement and make data-driven decisions to enhance overall supply chain performance.

SUPPLY CHAIN

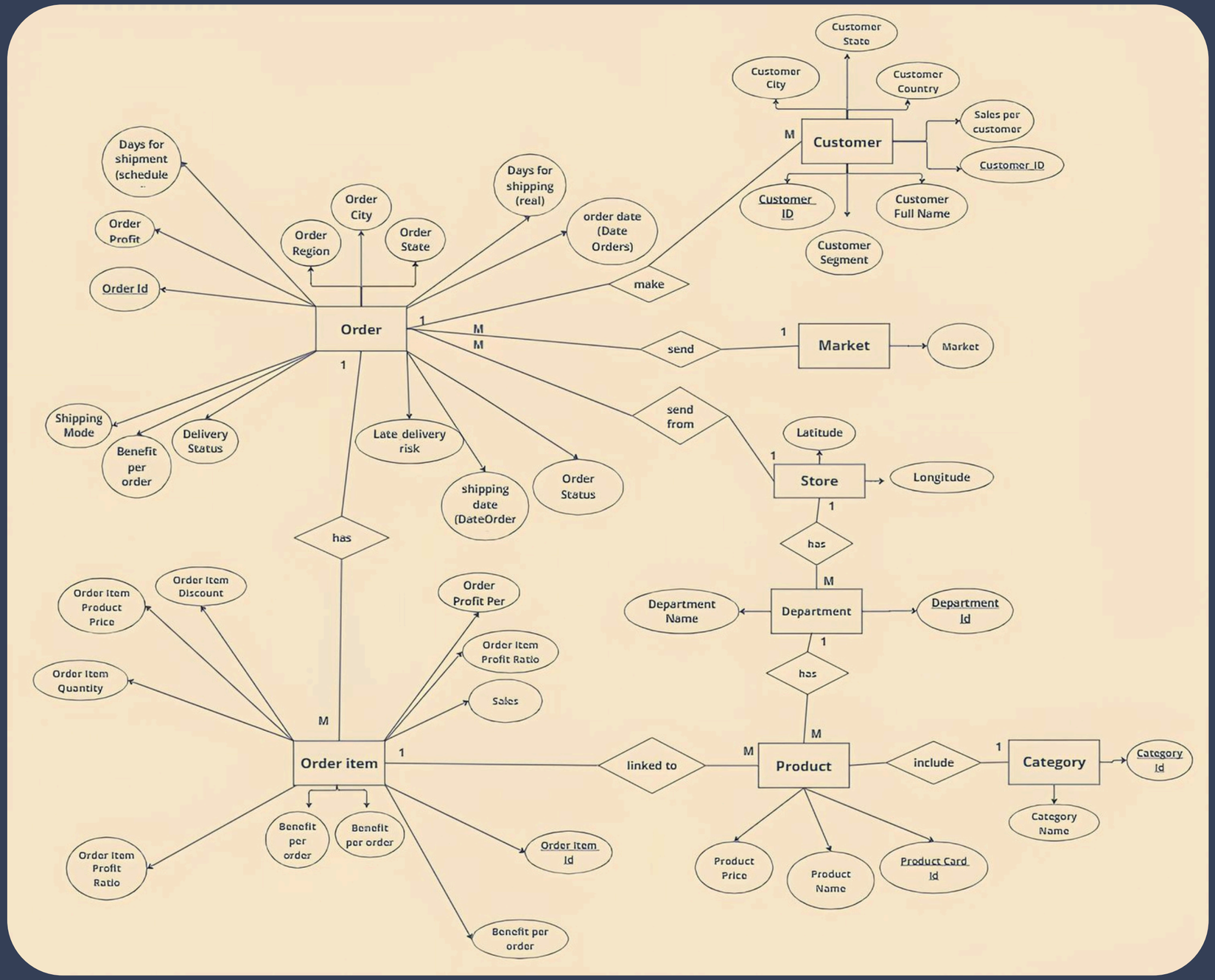
SCOPE OF DATA

- Customer
- Sales
- Products
- Market
- Order
- Delivery Status

Supply Chain

Entity Relationship Diagram

An Entity-Relationship Diagram (ERD) visualizes the relationships between entities (objects or concepts) in a system. It's commonly used in database design to illustrate how data is structured and connected. ERDs help to model the logical structure of databases by depicting entities, attributes and relationships.

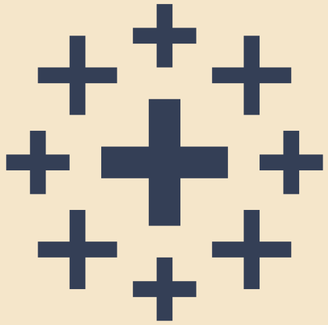


Entity Relationship Diagram of W Supply Chain Company

Analyzing Data

01 PYTHON

TABLEAU



01 PYTHON

```
read_supply_chain= pd.read_excel('/content/drive/MyDrive/DEPI/final_project/CoSupplyChainDataset.xlsx')
read_supply_chain.head()
```

OutPut

	Type	Days for shipping (real)	Days for shipment (scheduled)	Benefit per order	Sales per customer	Delivery Status	Late_delivery_risk	Category Id	Category Name	Customer City
0	DEBIT	3	4	91.250000	314.640015	Advance shipping	0	73	Sporting Goods	Caguas
1	TRANSFER	5	4	-249.089996	311.359985	Late delivery	1	73	Sporting Goods	Caguas
2	CASH	4	4	-247.779999	309.720001	Shipping on time	0	73	Sporting Goods	San Jose
3	DEBIT	3	4	22.860001	304.809998	Advance shipping	0	73	Sporting Goods	Los Angeles
4	PAYMENT	2	4	134.210007	298.250000	Advance shipping	0	73	Sporting Goods	Caguas

SUPPLY CHAIN

01 PYTHON

Columns

```
read_supply_chain.columns
```

OutPut

```
Index(['Type', 'Days for shipping (real)', 'Days for shipment (scheduled)',
       'Benefit per order', 'Sales per customer', 'Delivery Status',
       'Late_delivery_risk', 'Category Id', 'Category Name', 'Customer City',
       'Customer Country', 'Customer Email', 'Customer Fname', 'Customer Id',
       'Customer Lname', 'Customer Password', 'Customer Segment',
       'Customer State', 'Customer Street', 'Customer Zipcode',
       'Department Id', 'Department Name', 'Latitude', 'Longitude', 'Market',
       'Order City', 'Order Country', 'Order Customer Id',
       'order date (DateOrders)', 'Order Id', 'Order Item Cardprod Id',
       'Order Item Discount', 'Order Item Discount Rate', 'Order Item Id',
       'Order Item Product Price', 'Order Item Profit Ratio',
       'Order Item Quantity', 'Sales', 'Order Item Total',
       'Order Profit Per Order', 'Order Region', 'Order State', 'Order Status',
       'Order Zipcode', 'Product Card Id', 'Product Category Id',
       'Product Description', 'Product Image', 'Product Name', 'Product Price',
       'Product Status', 'shipping date (DateOrders)', 'Shipping Mode'],
      dtype='object')
```

SUPPLY CHAIN

01 PYTHON

Drop Columns

```
read_supply_chain.drop(columns=['Product Description', 'Product Image',"Order Zipcode","Product
Status","Customer Zipcode"], inplace=True)
read_supply_chain.columns
```

OutPut

```
Index(['Type', 'Days for shipping (real)', 'Days for shipment (scheduled)',
       'Benefit per order', 'Sales per customer', 'Delivery Status',
       'Late_delivery_risk', 'Category Id', 'Category Name', 'Customer City',
       'Customer Country', 'Customer Email', 'Customer Fname', 'Customer Id',
       'Customer Lname', 'Customer Password', 'Customer Segment',
       'Customer State', 'Customer Street', 'Department Id', 'Department Name',
       'Latitude', 'Longitude', 'Market', 'Order City', 'Order Country',
       'Order Customer Id', 'order date (DateOrders)', 'Order Id',
       'Order Item Cardprod Id', 'Order Item Discount',
       'Order Item Discount Rate', 'Order Item Id', 'Order Item Product Price',
       'Order Item Profit Ratio', 'Order Item Quantity', 'Sales',
       'Order Item Total', 'Order Profit Per Order', 'Order Region',
       'Order State', 'Order Status', 'Product Card Id', 'Product Category Id',
       'Product Name', 'Product Price', 'shipping date (DateOrders)',
       'Shipping Mode'],
      dtype='object')
```

SUPPLY CHAIN

01 PYTHON

Importing Libraries

```python
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Unifying Color to be consistent with the identity of W supply chain company

```python
custom_palette = ['#343F56', '#A29D7B', '#F5E6CA', '#B9B9B9', '#959595', '#676767', '#323232', '#EAEAEA']

# Set the palette for matplotlib
plt.rcParams['axes.facecolor'] = '#343F56'  # Background of the plot itself
plt.rcParams['figure.facecolor'] = '#343F56'  # Background of the entire figure
plt.rcParams['savefig.facecolor'] = '#343F56'  # Background of saved figures

# # Set the palette for Seaborn
sns.set_style("darkgrid", {"axes.facecolor": "#343F56"})
```

W
SUPPLY CHAIN

01 PYTHON

Info About Dataset

```
read_supply_chain.info()
```

OutPut

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180519 entries, 0 to 180518
Data columns (total 51 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   Type                           180519 non-null  object
 1   Days for shipping (real)       180519 non-null  int64
 2   Days for shipment (scheduled)  180519 non-null  int64
 3   Benefit per order              180519 non-null  float64
 4   Sales per customer             180519 non-null  float64
 5   Delivery Status                180519 non-null  object
 6   Late_delivery_risk             180519 non-null  int64
 7   Category Id                    180519 non-null  int64
 8   Category Name                  180519 non-null  object
 9   Customer City                  180519 non-null  object
 10  Customer Country               180519 non-null  object
 11  Customer Email                 180519 non-null  object
 12  Customer Fname                 180519 non-null  object
 13  Customer Id                    180519 non-null  int64
 14  Customer Lname                 180511 non-null  object
 15  Customer Password              180519 non-null  object
 16  Customer Segment               180519 non-null  object
```

Describe Dataset

```
read_supply_chain.describe()
```

OutPut

	Days for shipping (real)	Days for shipment (scheduled)	Benefit per order	Sales per customer
count	180519.000000	180519.000000	180519.000000	180519.000000
mean	3.497654	2.931847	21.974989	183.107609
std	1.623722	1.374449	104.433526	120.043670
min	0.000000	0.000000	-4274.979980	7.490000
25%	2.000000	2.000000	7.000000	104.379997
50%	3.000000	4.000000	31.520000	163.990005
75%	5.000000	4.000000	64.800003	247.399994
max	6.000000	4.000000	911.799988	1939.989990

SUPPLY CHAIN

01 PYTHON

Null Values

```
read_supply_chain.isnull().sum()
```

OutPut

Late_delivery_risk	0
Category Id	0
Category Name	0
Customer City	0
Customer Country	0
Customer Email	0
Customer Fname	0
Customer Id	0
Customer Lname	8

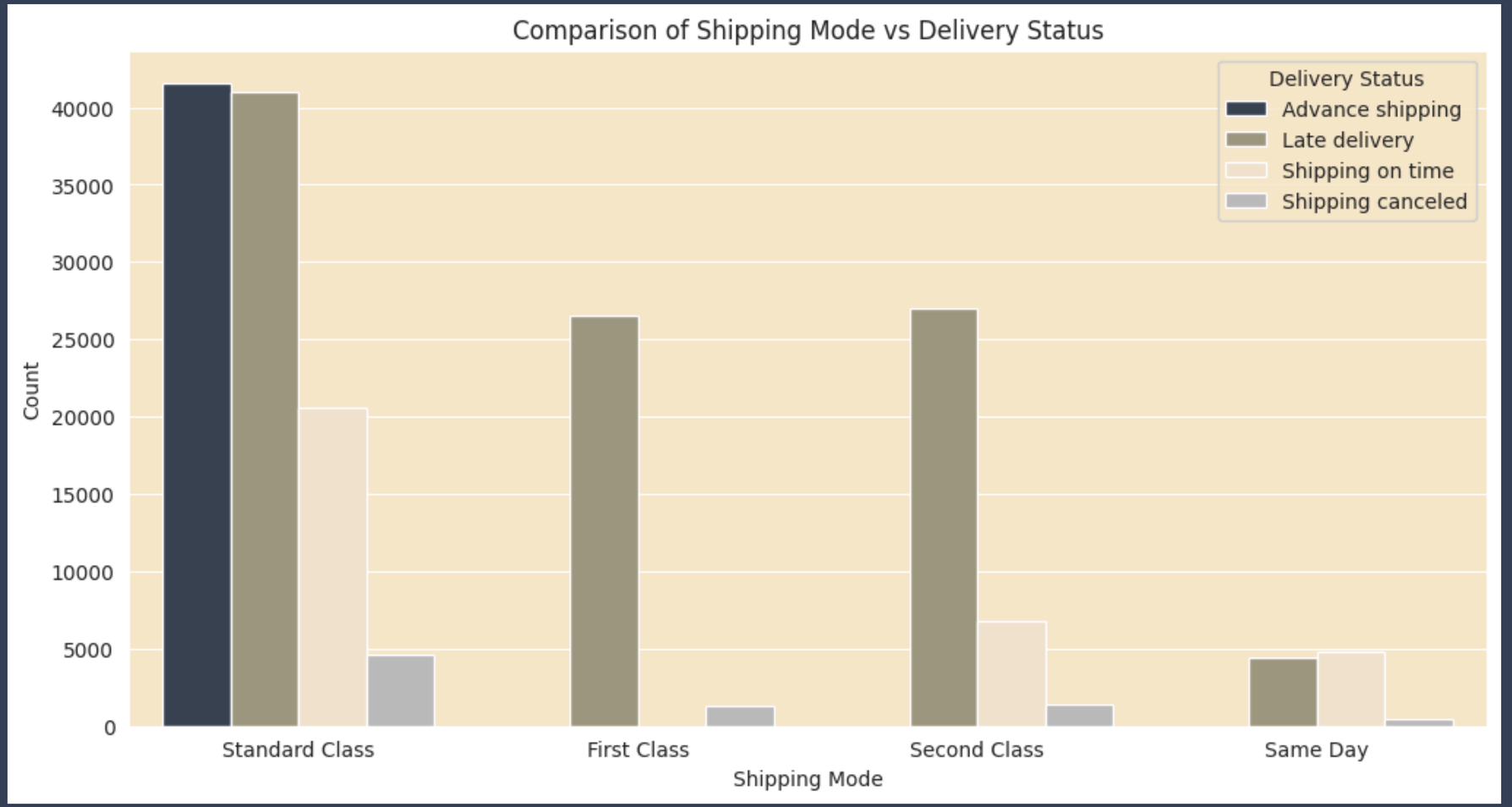
SUPPLY CHAIN

01 PYTHON

Delivery Status& Shipping Mode

OutPut

```
plt.figure(figsize=(12, 6))
sns.countplot(x='Shipping Mode', hue='Delivery Status', data=read_supply_chain)
plt.title('Comparison of Shipping Mode vs Delivery Status')
plt.xlabel('Shipping Mode')
plt.ylabel('Count')
plt.show()
```



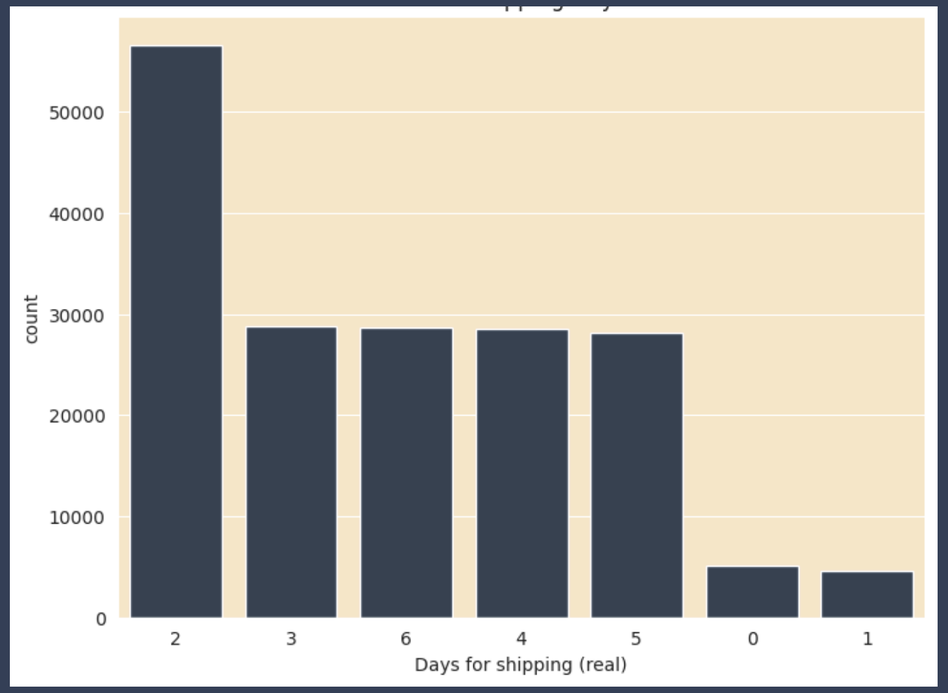
SUPPLY CHAIN

01 PYTHON

Shipping Days: Real vs. Scheduled:

```
plt.figure(figsize=(8, 6))
sns.countplot(x='Days for shipment (scheduled)',data=read_supply_chain)
plt.title('Scheduled Shipping Days')
plt.show()
```


```
plt.figure(figsize=(8, 6))
sns.countplot(x='Days for shipping (real)',data=read_supply_chain,order=read_supply_chain['Days for
shipping (real)'].value_counts().index)
plt.title('Real Shipping Days')
plt.show()
```

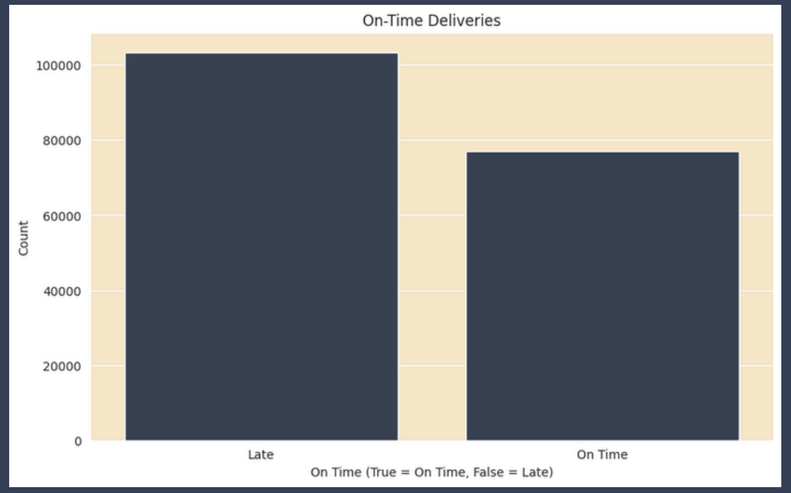


SUPPLY CHAIN

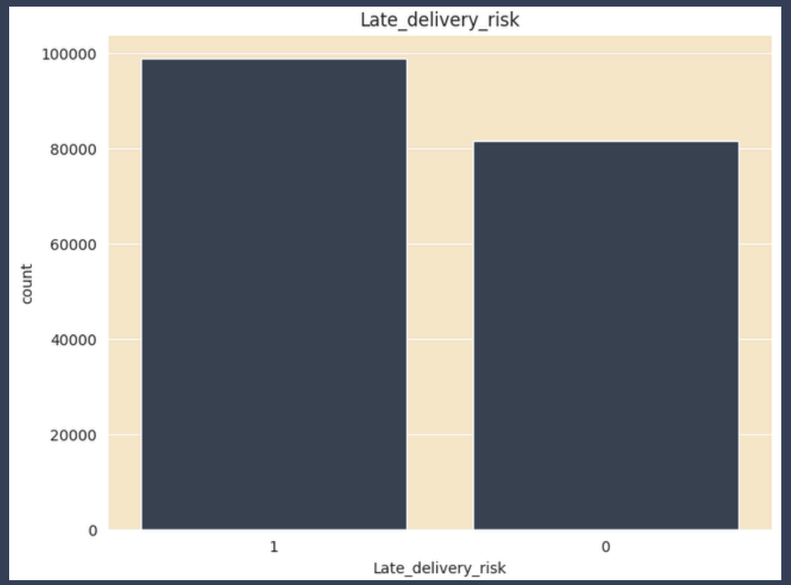
01 PYTHON

Shipping Days: Real vs. Scheduled:

```
read_supply_chain['On_Time'] = read_supply_chain['Days for shipping (real)'] <= read_supply_chain['Days
for shipment (scheduled)']
plt.figure(figsize=(10, 6))
sns.countplot(data=read_supply_chain, x='On_Time')
plt.title('On-Time Deliveries')
plt.xlabel('On Time (True = On Time, False = Late)')
plt.ylabel('Count')
plt.xticks(ticks=[0, 1], labels=['Late', 'On Time'])
plt.show()
```



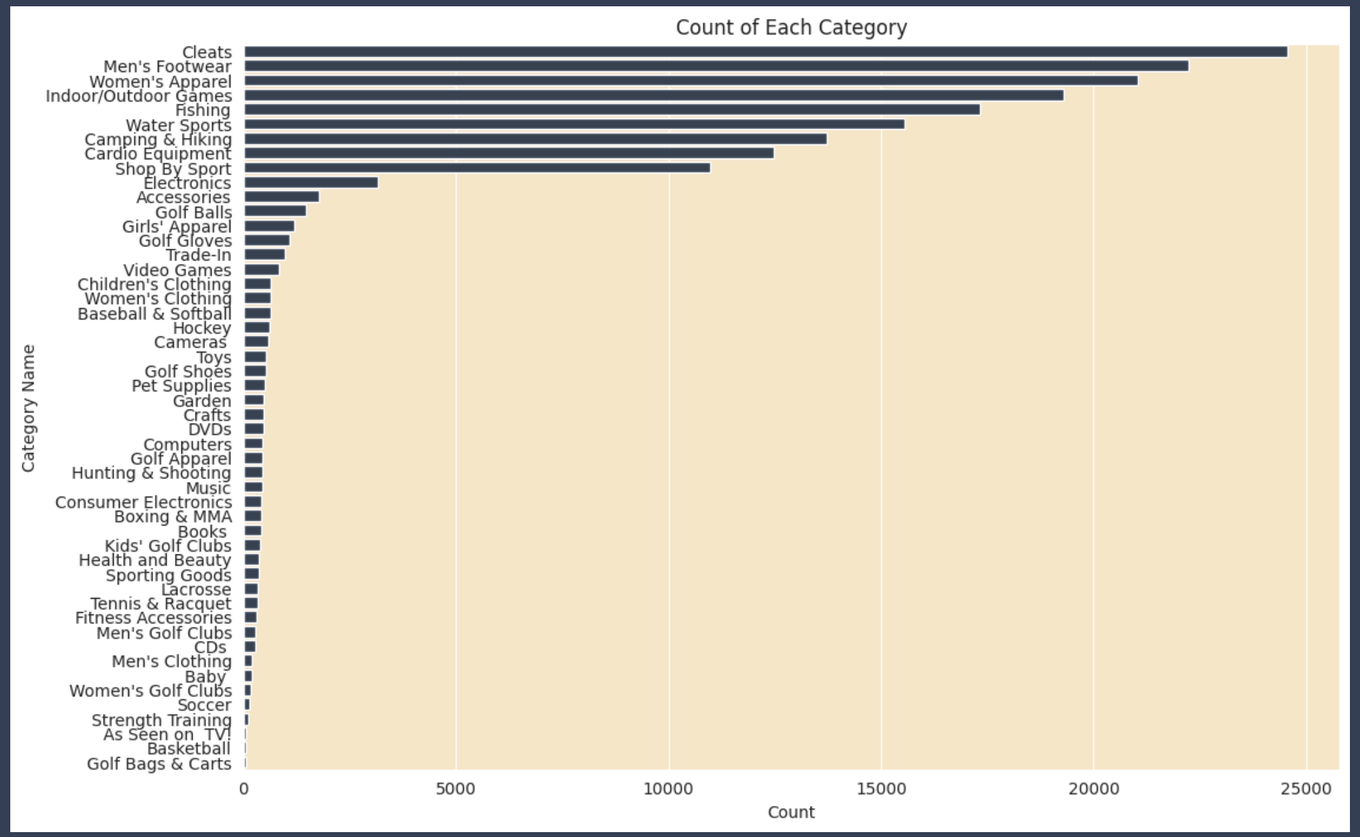

```
plt.figure(figsize=(8, 6))
sns.countplot(x='Late_delivery_risk',data=read_supply_chain,order=read_supply_chain['Late_delivery_risk']
.value_counts().index)
plt.title('Late_delivery_risk')
plt.show()
```



SUPPLY CHAIN

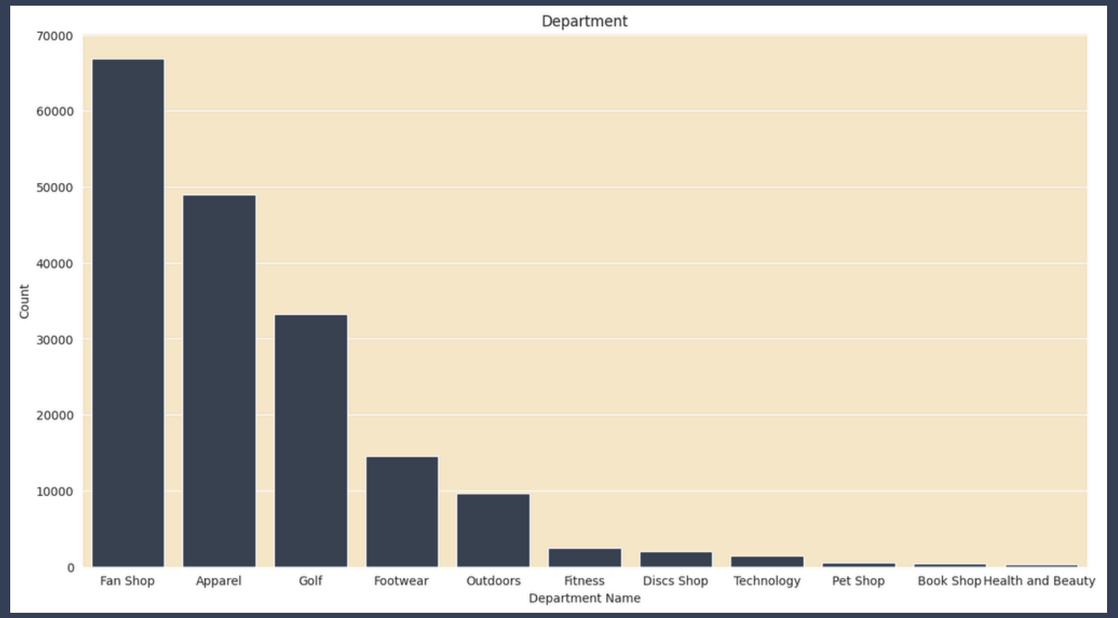
01 PYTHON

Products



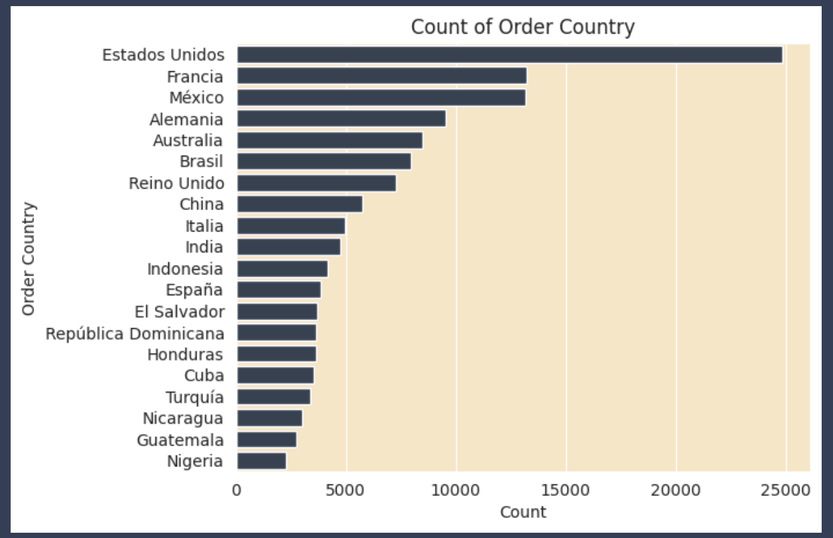
Sample of Code (categorical data)

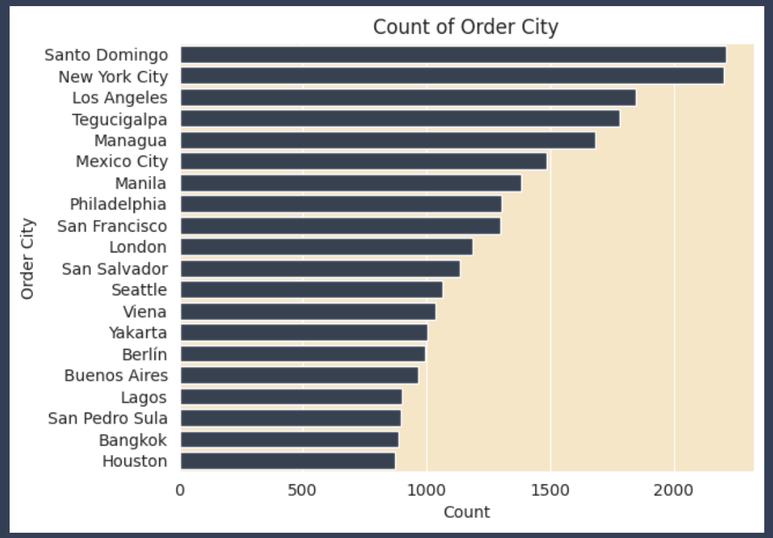
```
plt.figure(figsize=(12, 8))
sns.countplot(y='Category Name', data=read_supply_chain, order=read_supply_chain['Category
Name'].value_counts().index)
plt.title('Count of Each Category')
plt.xlabel('Count')
plt.ylabel('Category Name')
plt.show()
```

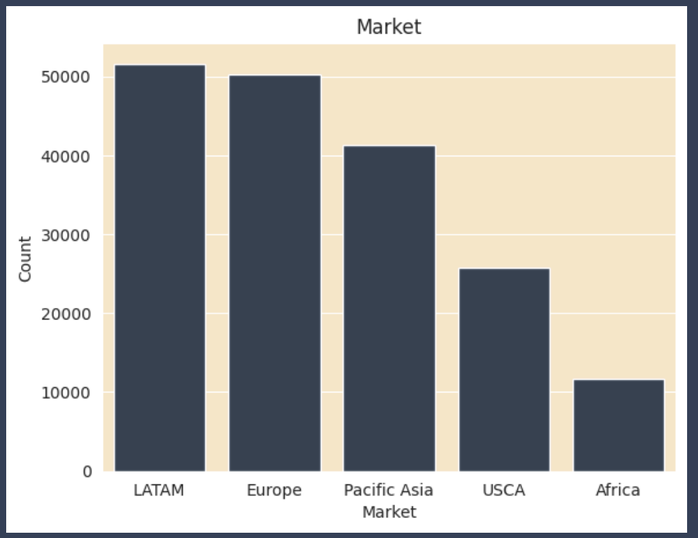


SUPPLY CHAIN

01 PYTHON





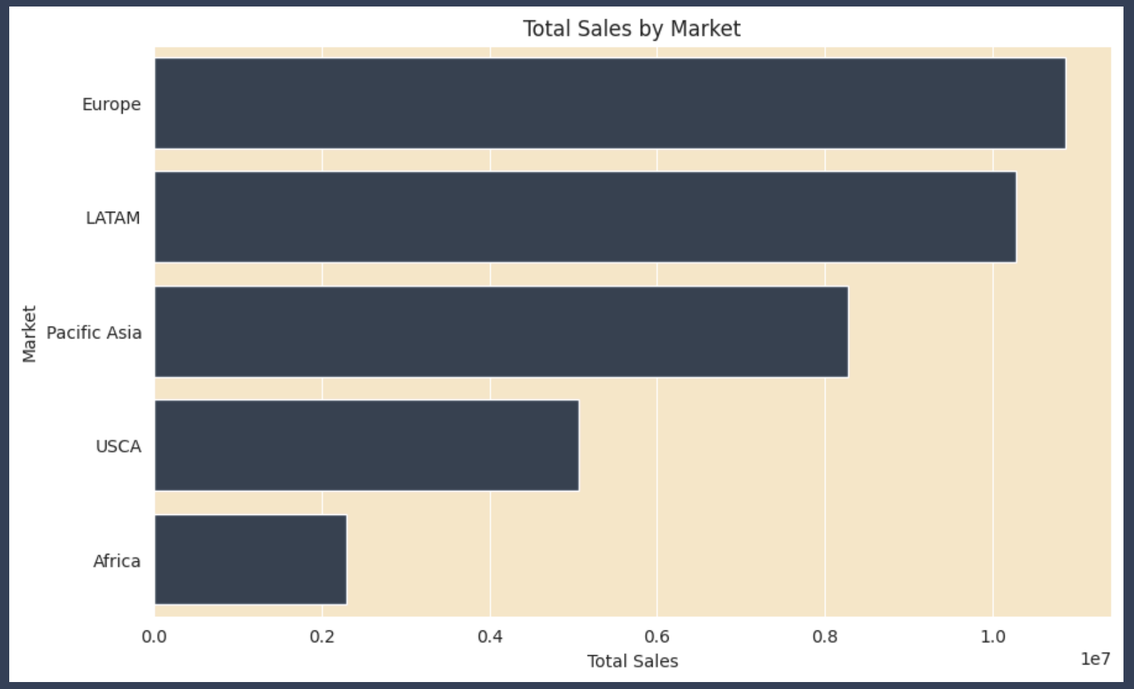


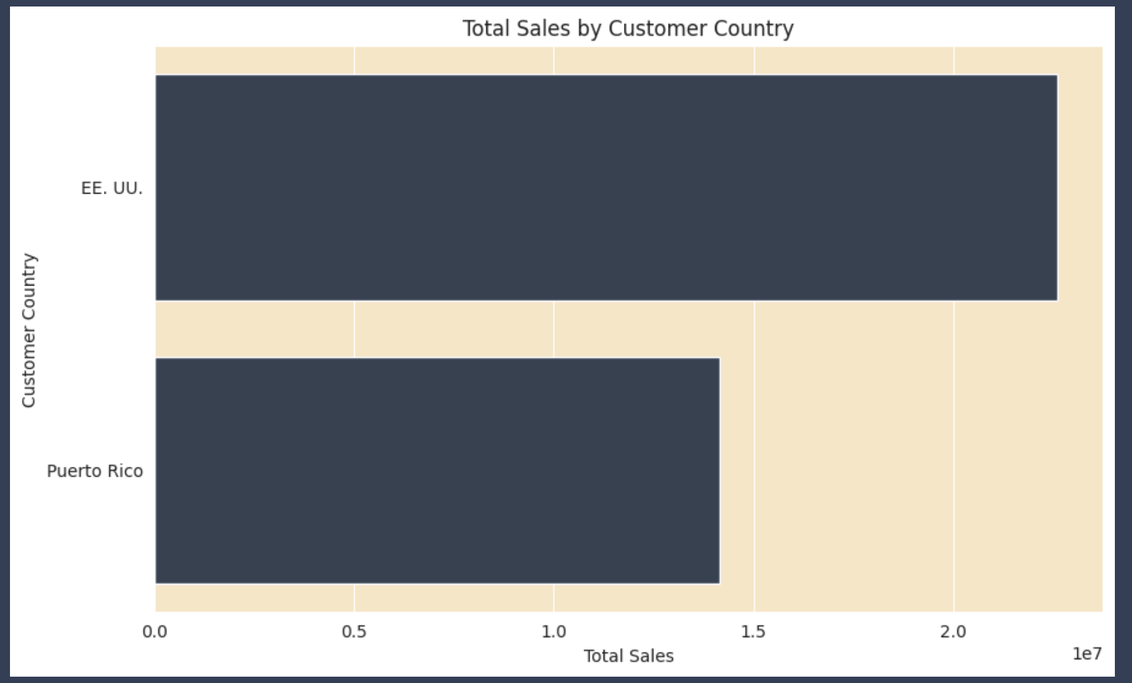
Geographics

SUPPLY CHAIN

01 PYTHON

Sales& Geographics

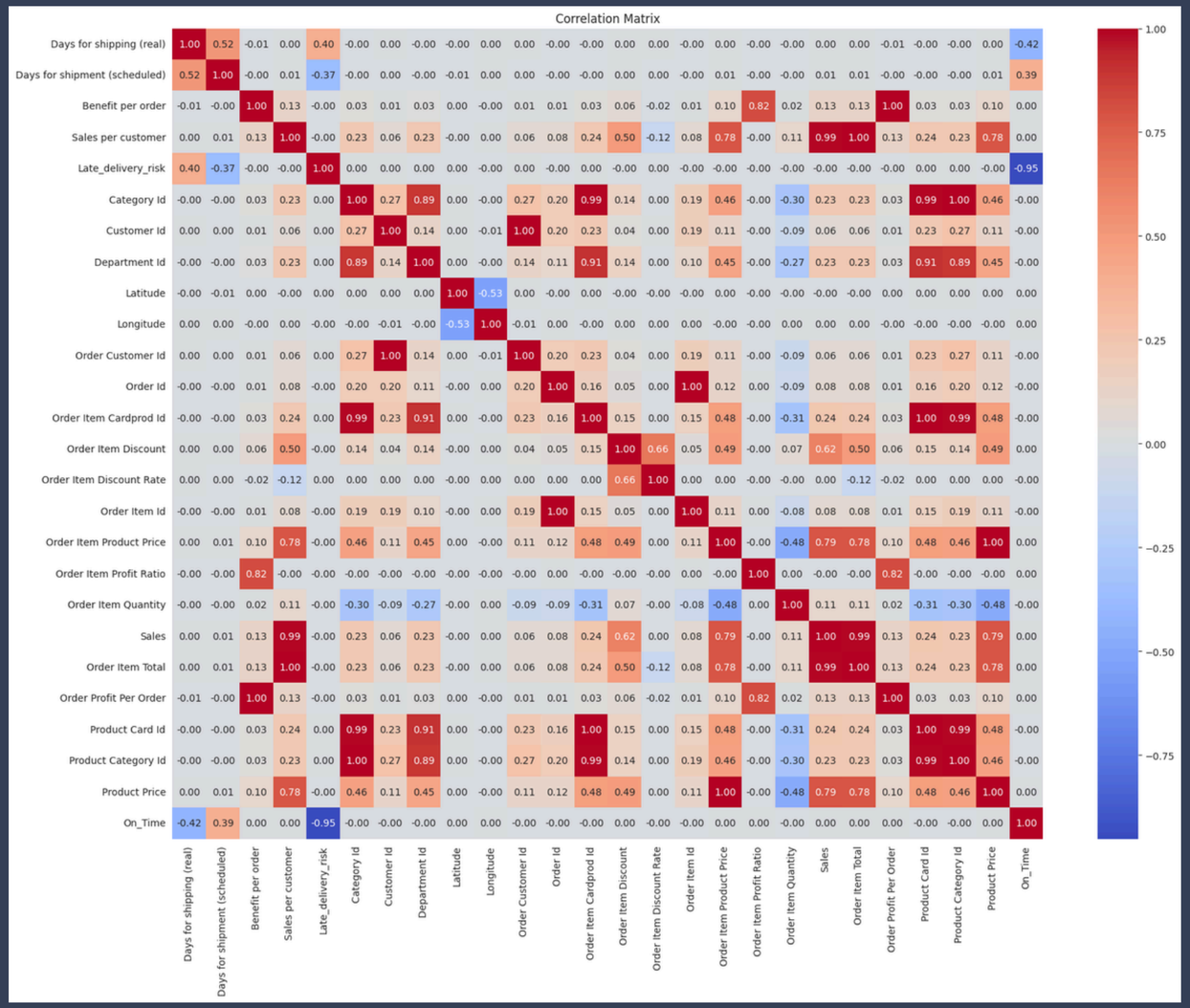




SUPPLY CHAIN

01 PYTHON

Relations




```
correlation_matrix = read_supply_chain.corr(numeric_only=True)
plt.figure(figsize=(20, 15))
sns.heatmap(correlation_matrix, annot=True, fmt='.2f', cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```

Strong Positive Correlations (Above 0.5)

- Sales and Order Item Total:
  Correlation: 0.99
- Order Item Profit Ratio and Sales:
  Correlation: 0.82
- Sales and Sales per Customer:
  Correlation: 0.78
- Benefit per Order and Sales:
  Correlation: 0.78

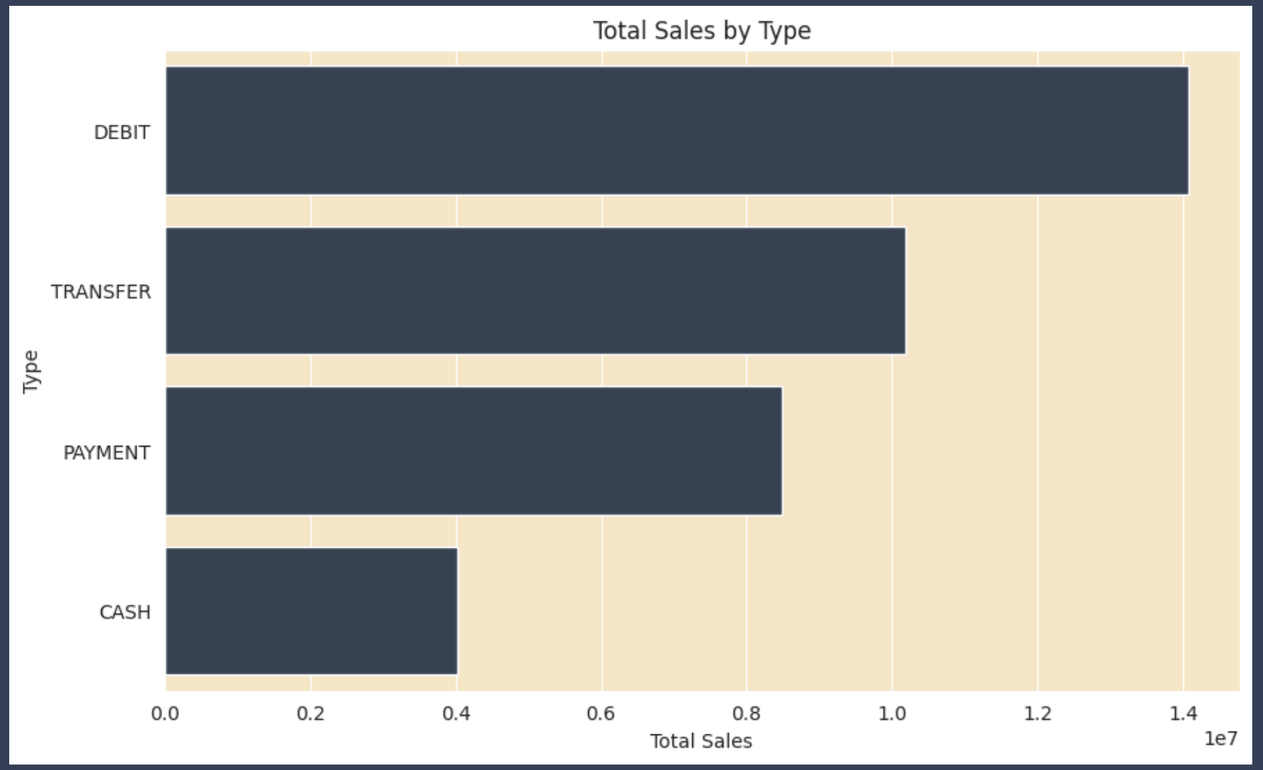
Strong Negative Correlations (Below -0.5)

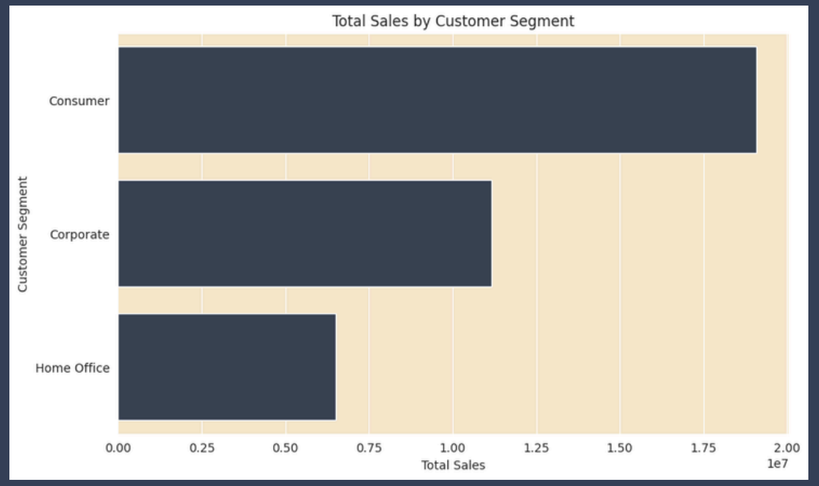
- Late Delivery Risk and On Time:
  Correlation: -0.95
- Days for Shipping (Real) and On Time:
  Correlation: -0.42
- Days for Shipment (Scheduled) and Days for Shipping (Real):
  Correlation: -0.37

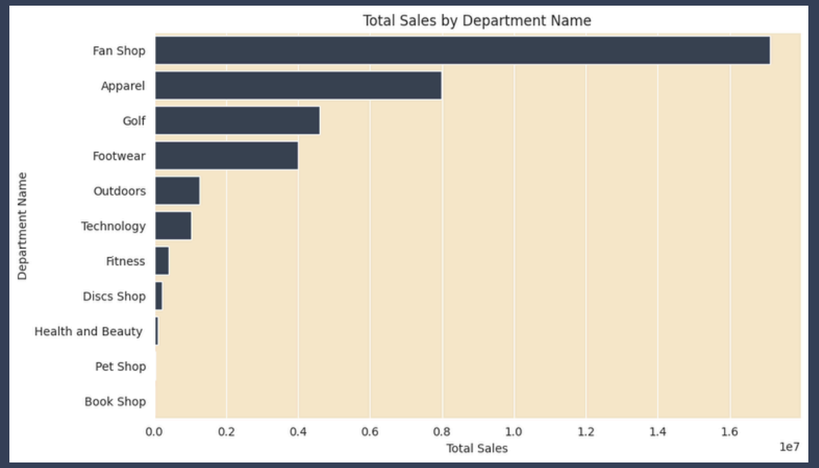
SUPPLY CHAIN

01 PYTHON

Sales & Customers



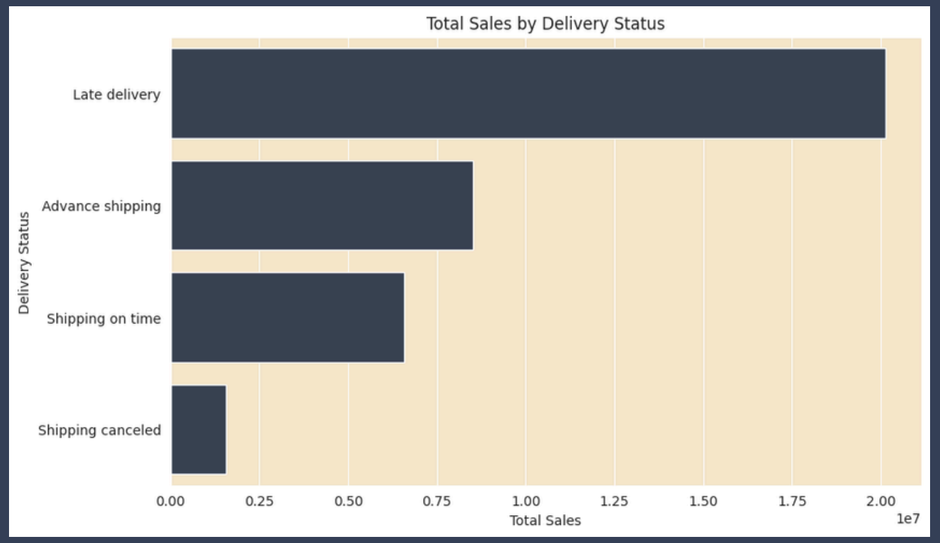


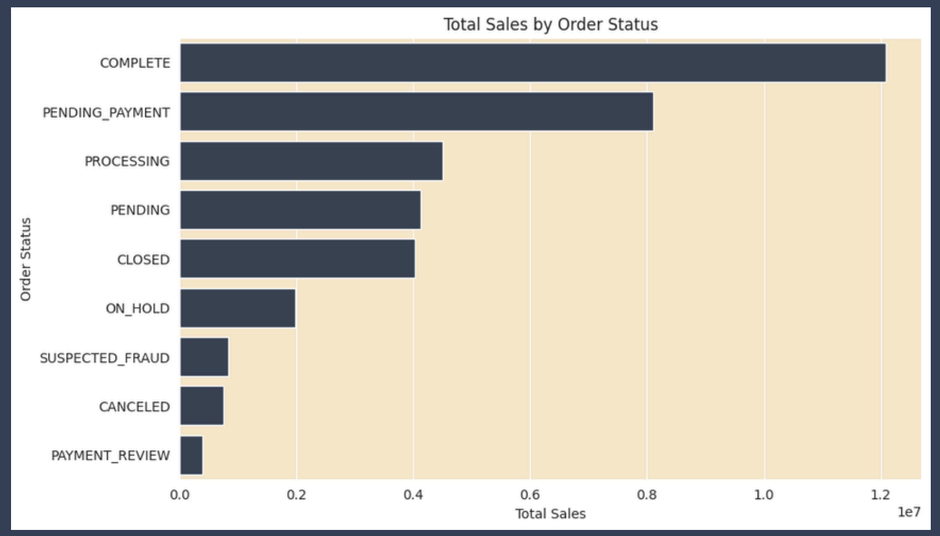


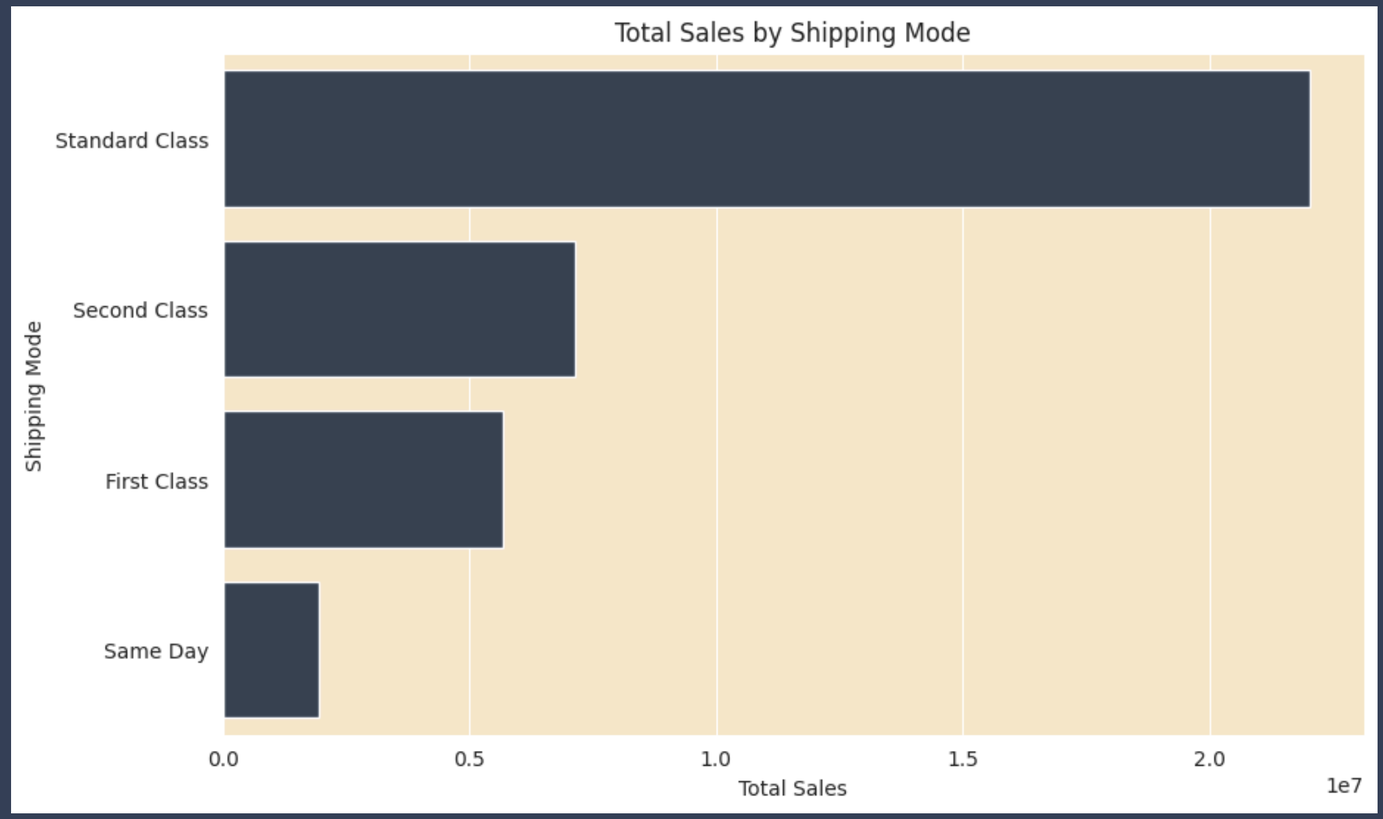
SUPPLY CHAIN

01 PYTHON

Sales & Shipping







SUPPLY CHAIN

01 PYTHON

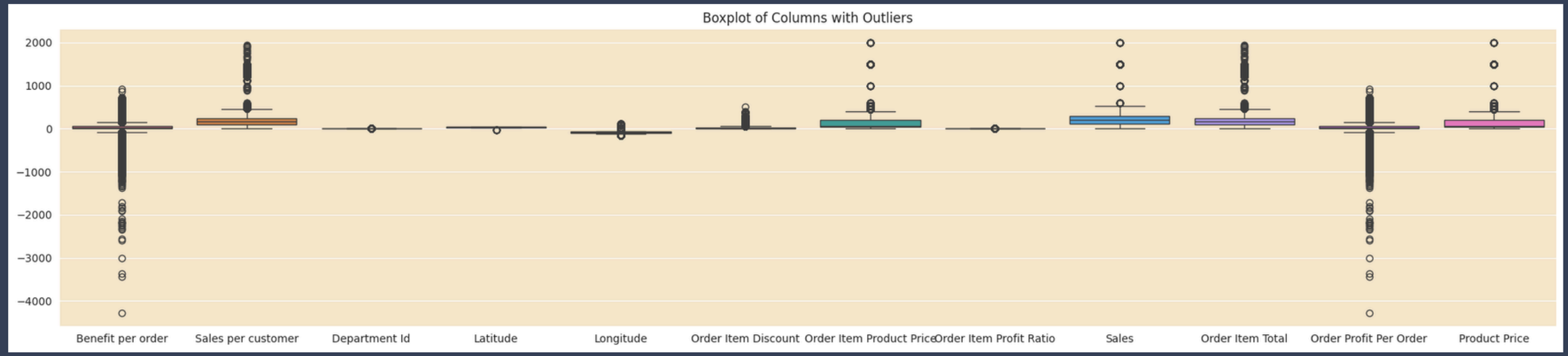
Prediction of Sales Based on shipping

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
df_sample = read_supply_chain.sample(n=50000, random_state=42)
X = df_sample[['Product Price', 'Days for shipping (real)', 'Late_delivery_risk']]
y = df_sample['Sales per customer']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

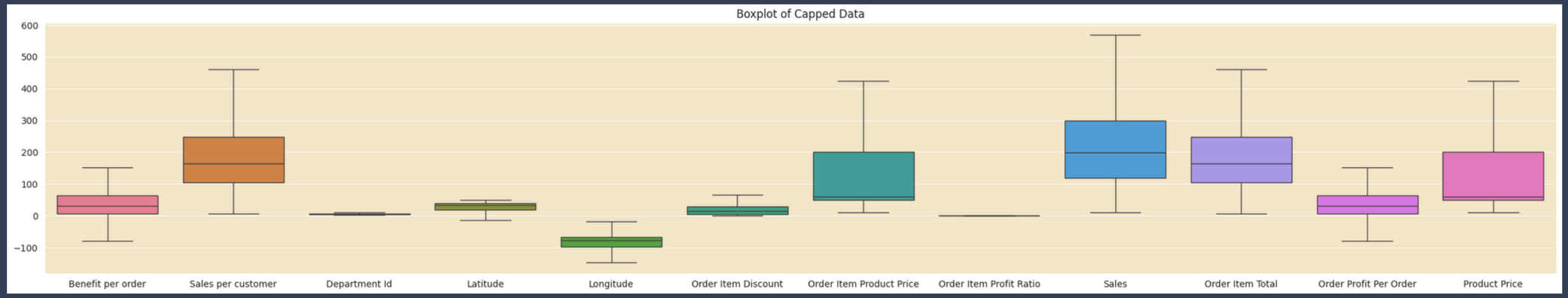
SUPPLY CHAIN

01 PYTHON

BEFORE



AFTER



SUPPLY CHAIN

01 PYTHON

Prediction of Sales Based on shipping

```python
#Linear Regression
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
mse, r2
```

```
(5617.034731656759, 0.6000595538631259)
```

```python
#Random Forest
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
mse, r2
```

```
(3512.910287104074 5, 0.7498760512294673)
```

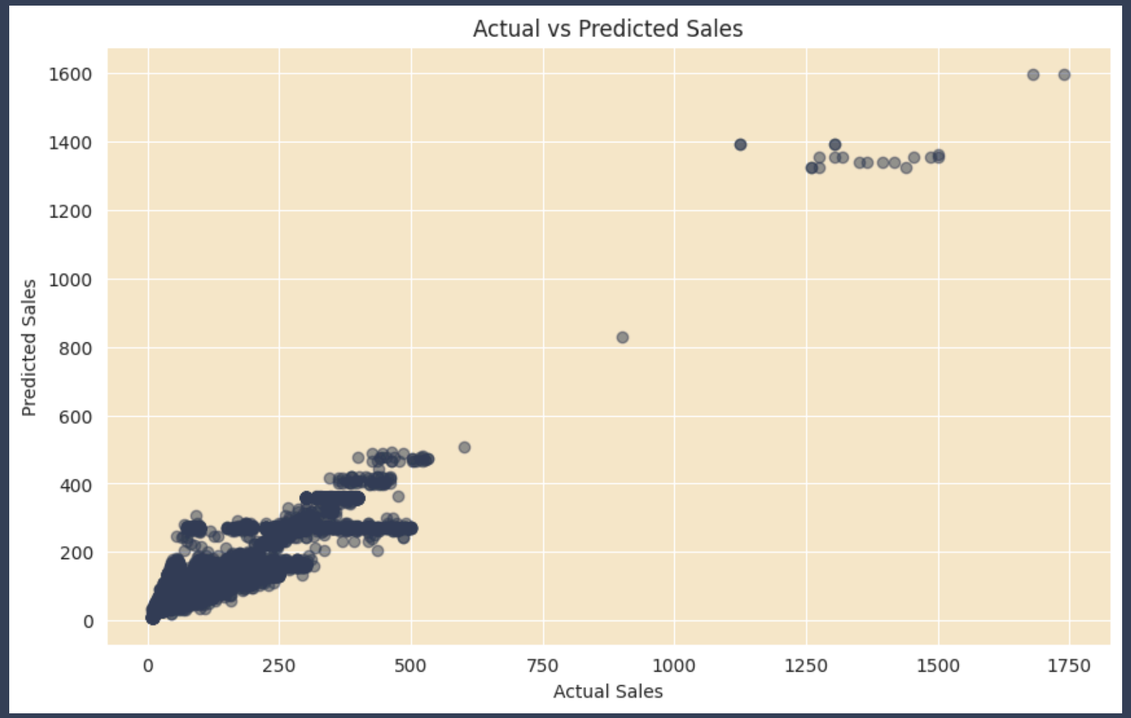
SUPPLY CHAIN

01 PYTHON

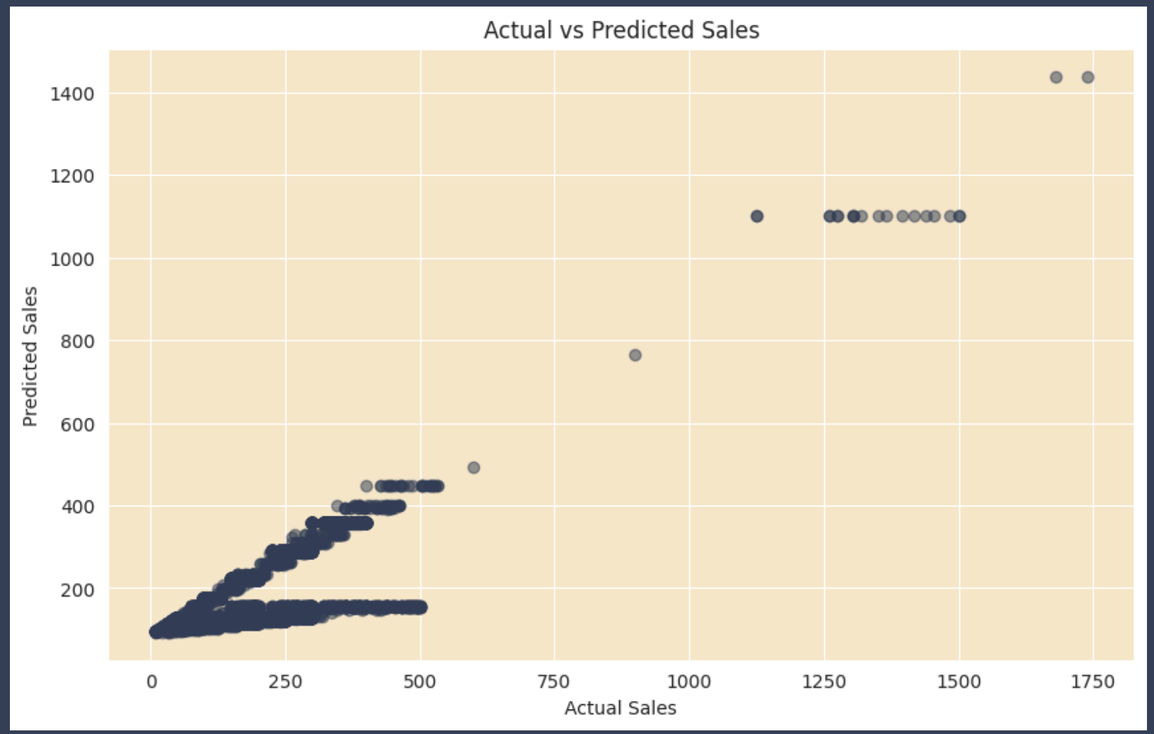
```
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, alpha=0.5)
plt.xlabel("Actual Sales")
plt.ylabel("Predicted Sales")
plt.title("Actual vs Predicted Sales")
plt.show()
```

Prediction of Sales Based on shipping

Random Forest



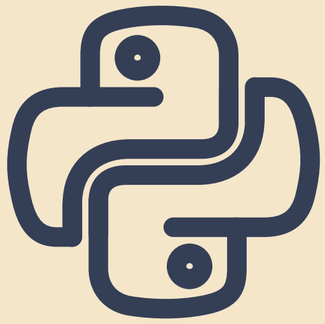
Linear Regression



SUPPLY CHAIN

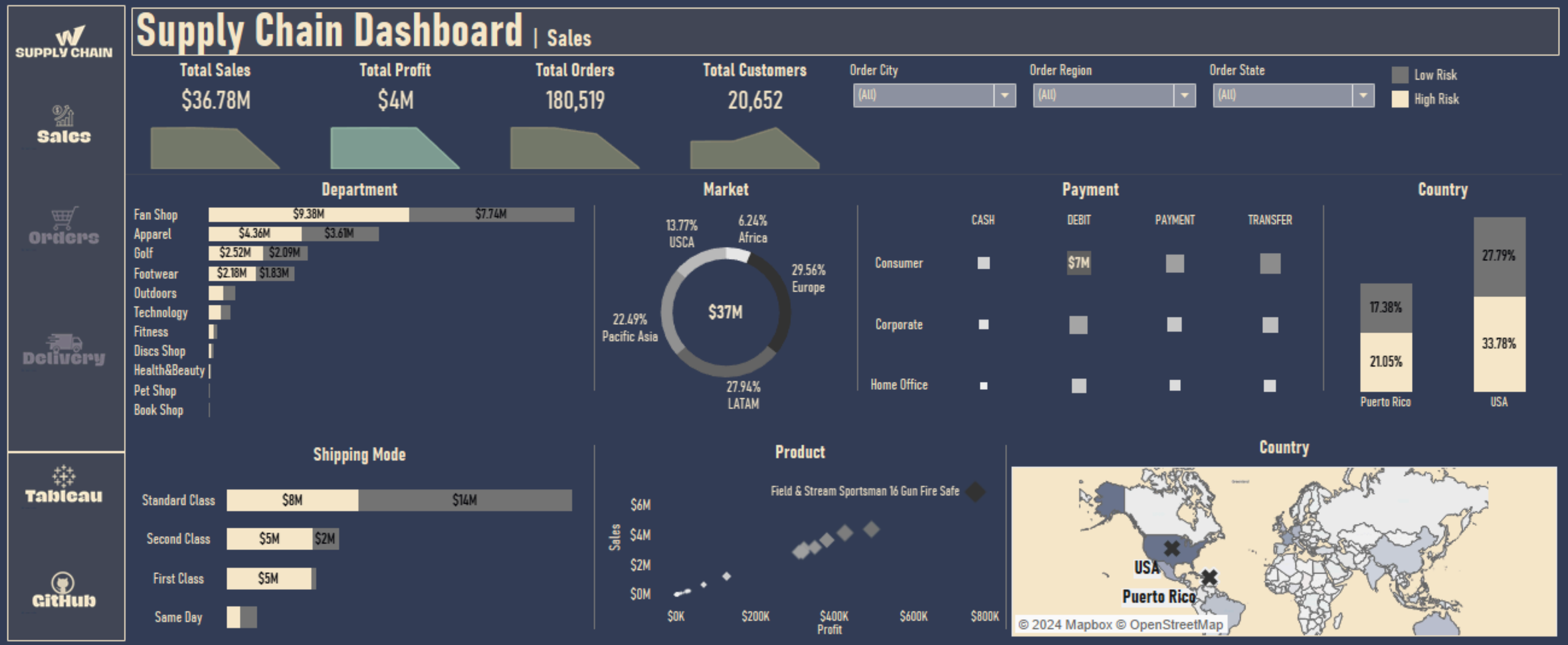
Analyzing Data

01 PYTHON



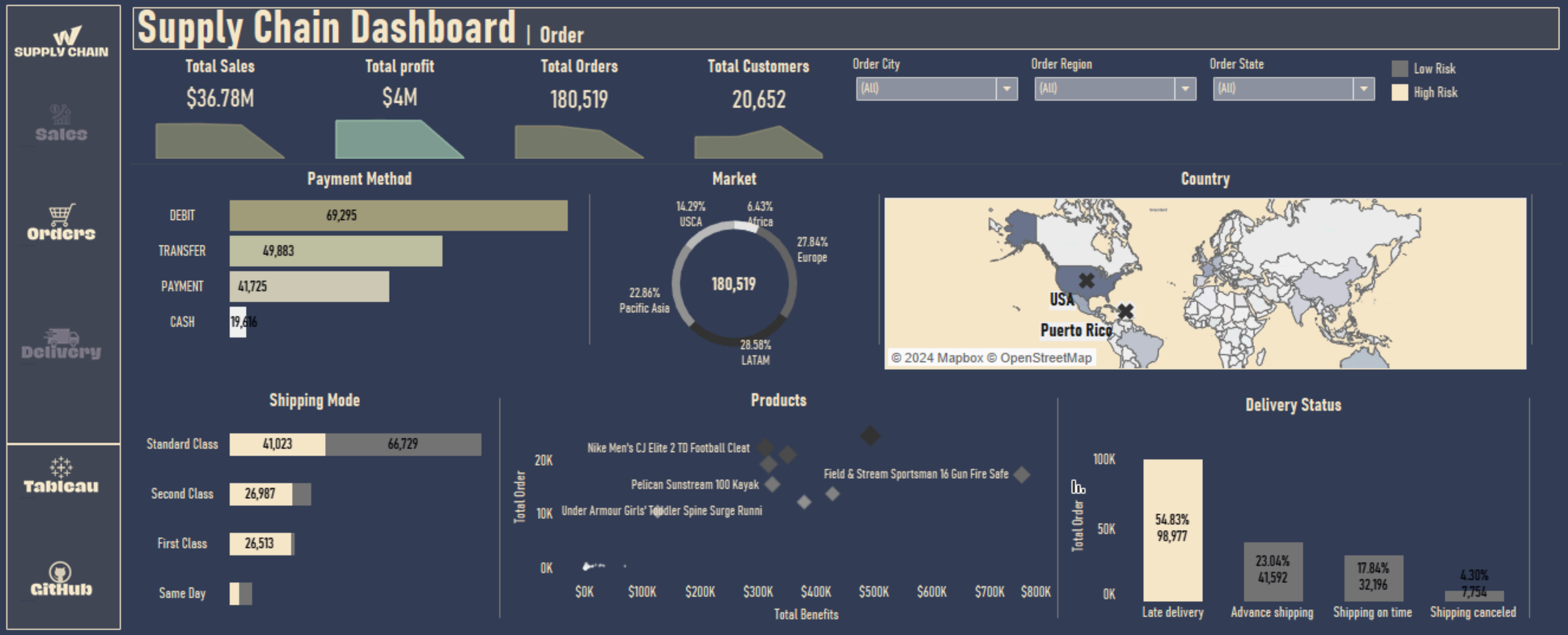
02 TABLEAU

02 TABLEAU



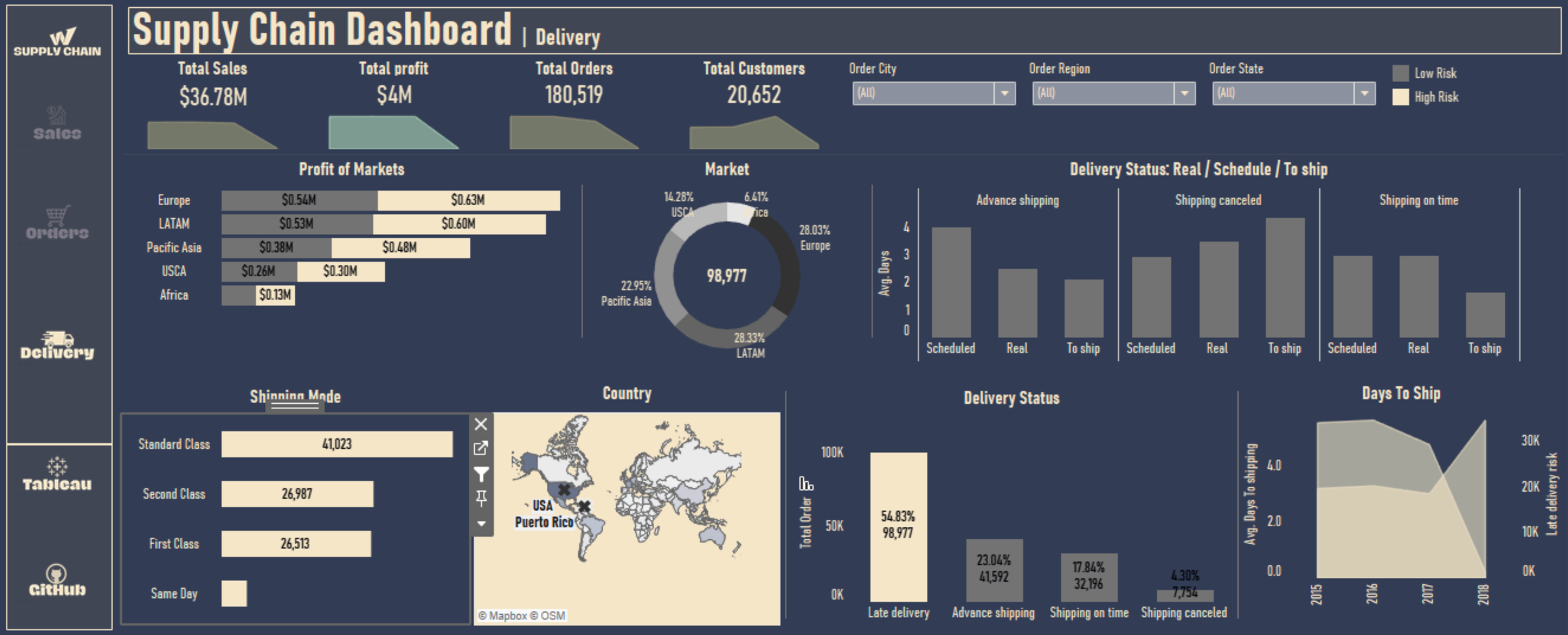
SUPPLY CHAIN

02 TABLEAU



SUPPLY CHAIN

02 TABLEAU



SUPPLY CHAIN

SUPPLY CHAIN

RECOMMENDATIONS

SUPPLY CHAIN

RECOMMENDATION 01

Top Fan department not only generates the highest sales but also faces a significant risk of delayed deliveries, which could impact customer satisfaction and business performance. Focusing on optimizing delivery strategies for this department would help mitigate risks and sustain its strong sales performance.

RECOMMENDATION 02

As approximately 55% of orders experience delays. By targeting this area for improvement, we can enhance overall customer satisfaction and optimize shipping processes, leading to better performance outcomes for the project.

RECOMMENDATION 03

Europe has the highest rate of late deliveries, along with a significant increase in shipping times after orders are placed. By prioritizing efforts to address these challenges in this key market, we can enhance shipping efficiency and improve overall customer satisfaction.

THANK YOU