# WEEK 1

Build Data Model, Data Cleaning and Preprocessing

## 1. Introduction

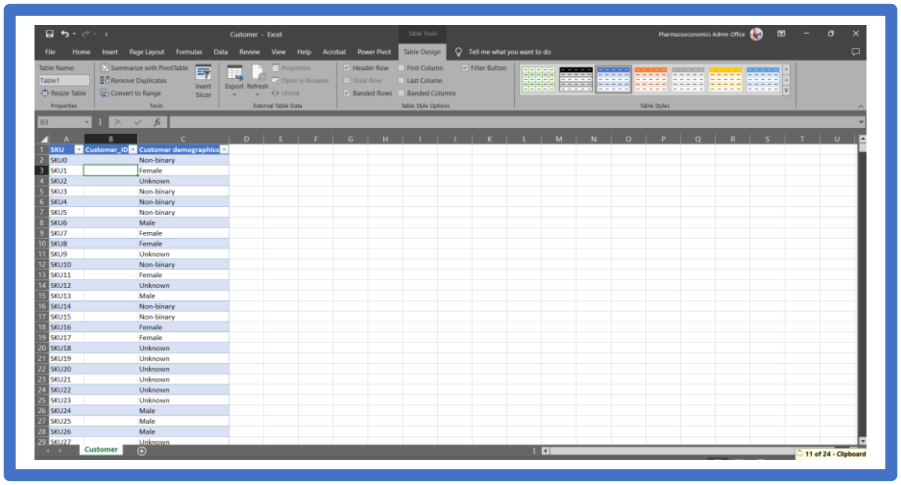
This document outlines the process of building a data model from an Excel file, cleaning the data, and preprocessing it for analysis. The project involves the creation of a database in SQL Server, the separation of data into different tables, and data cleaning using Python's Pandas library.

## 2. Data Preparation

### 2.1 Initial Data Extraction

The initial data was extracted from an Excel file with 24 columns. This data was separated into 7 main CSV files, each representing a specific aspect of the supply chain:

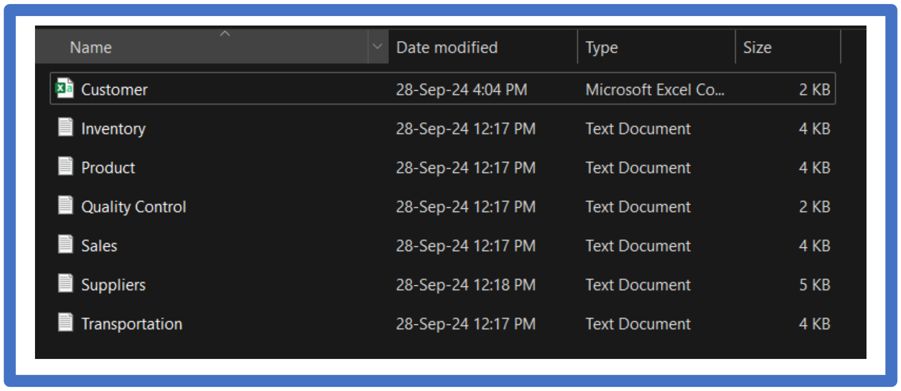
- Customers
- Inventory
- Products
- Quality Control
- Sales
- Suppliers
- Transportation



| SKU | Customer_ID | Customer demographics |
|---|---|---|
| SKU0 | | Non-binary |
| SKU1 | | Female |
| SKU2 | | Unknown |
| SKU3 | | Non-binary |
| SKU4 | | Non-binary |
| SKU5 | | Non-binary |
| SKU6 | | Male |
| SKU7 | | Female |
| SKU8 | | Female |
| SKU9 | | Unknown |
| SKU10 | | Non-binary |
| SKU11 | | Female |
| SKU12 | | Unknown |
| SKU13 | | Male |
| SKU14 | | Non-binary |
| SKU15 | | Non-binary |
| SKU16 | | Female |
| SKU17 | | Female |
| SKU18 | | Unknown |
| SKU19 | | Unknown |
| SKU20 | | Unknown |
| SKU21 | | Unknown |
| SKU22 | | Unknown |
| SKU23 | | Unknown |
| SKU24 | | Male |
| SKU25 | | Male |
| SKU26 | | Male |
| SKU27 | | Unknown |

## 2.2 CSV File Creation

The separated data was saved as tab-delimited CSV files. Below is an example of how to create a Transportation table in SQL Server:



| Name | Date modified | Type | Size |
|---|---|---|---|
| Customer | 28-Sep-24 4:04 PM | Microsoft Excel Co... | 2 KB |
| Inventory | 28-Sep-24 12:17 PM | Text Document | 4 KB |
| Product | 28-Sep-24 12:17 PM | Text Document | 4 KB |
| Quality Control | 28-Sep-24 12:17 PM | Text Document | 2 KB |
| Sales | 28-Sep-24 12:17 PM | Text Document | 4 KB |
| Suppliers | 28-Sep-24 12:18 PM | Text Document | 5 KB |
| Transportation | 28-Sep-24 12:17 PM | Text Document | 4 KB |

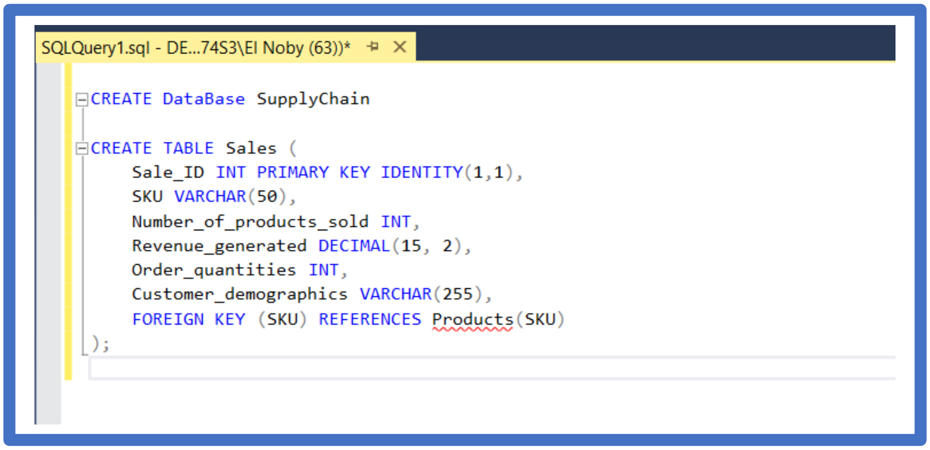
# 3. Database Creation in SQL Server

## 3.1 Creating the Database

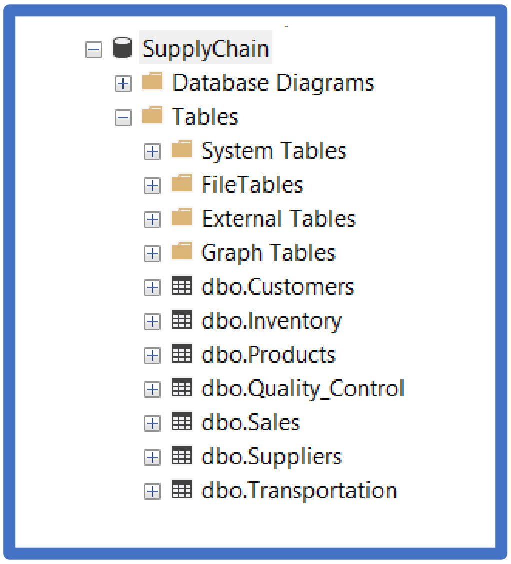
A new database called Supply Chain was created in SQL Server.

## 3.2 Creating Tables

For each of the CSV files, corresponding tables were created in the SQL Server database. For example, the Products table was created as follows:



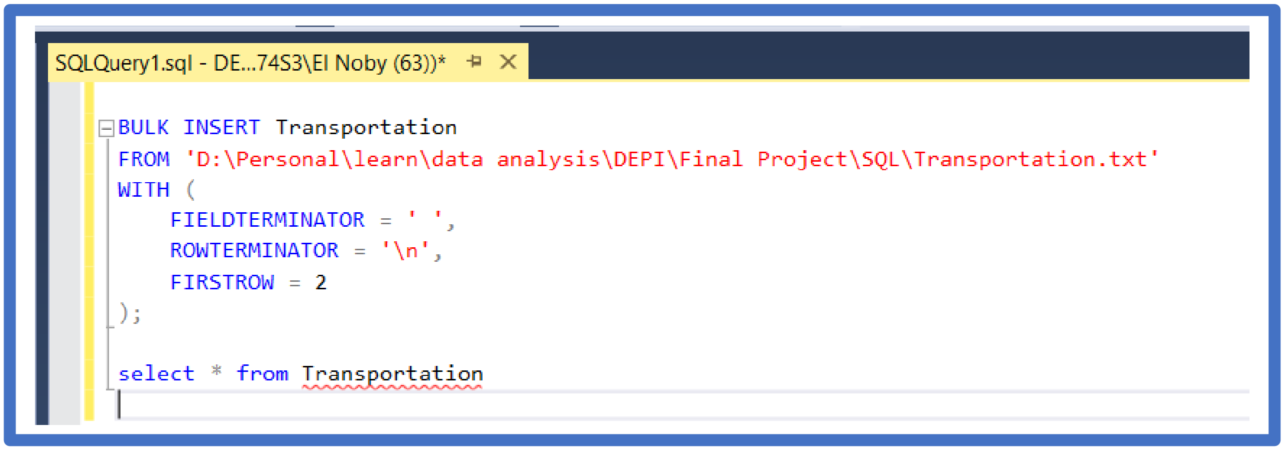
```
CREATE DataBase SupplyChain

CREATE TABLE Sales (
    Sale_ID INT PRIMARY KEY IDENTITY(1,1),
    SKU VARCHAR(50),
    Number_of_products_sold INT,
    Revenue_generated DECIMAL(15, 2),
    Order_quantities INT,
    Customer_demographics VARCHAR(255),
    FOREIGN KEY (SKU) REFERENCES Products(SKU)
);
```



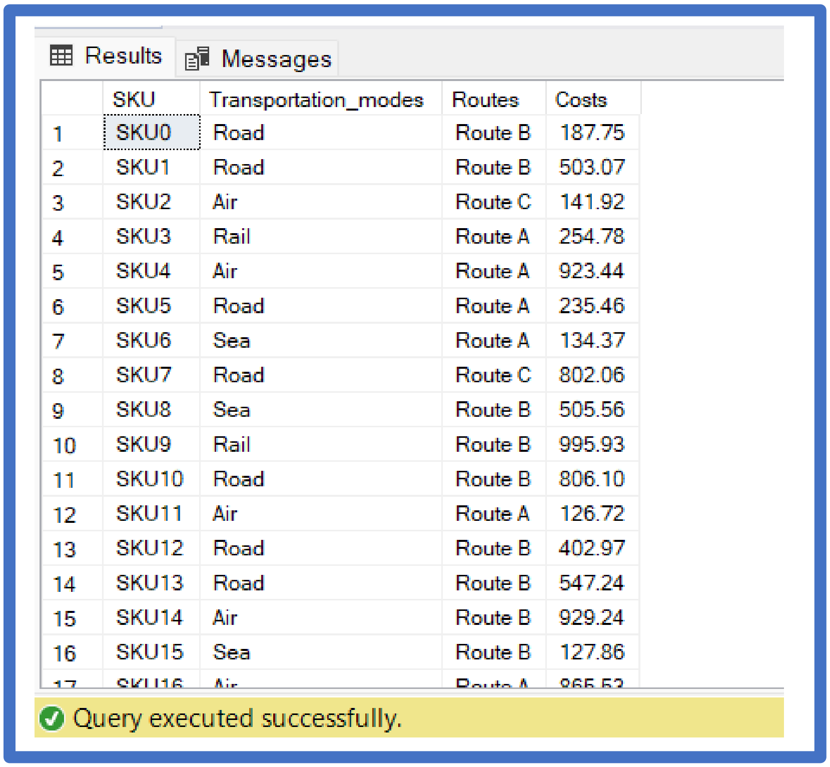
```
⊟ SupplyChain
   ⊞ Database Diagrams
   ⊟ Tables
      ⊞ System Tables
      ⊞ FileTables
      ⊞ External Tables
      ⊞ Graph Tables
      ⊞ dbo.Customers
      ⊞ dbo.Inventory
      ⊞ dbo.Products
      ⊞ dbo.Quality_Control
      ⊞ dbo.Sales
      ⊞ dbo.Suppliers
      ⊞ dbo.Transportation
```

## 4. Data Ingestion

Data was imported into each table using the BULK INSERT command. Below is an example for the Transportation table:



```
BULK INSERT Transportation
FROM 'D:\Personal\learn\data analysis\DEPI\Final Project\SQL\Transportation.txt'
WITH (
    FIELDTERMINATOR = ' ',
    ROWTERMINATOR = '\n',
    FIRSTROW = 2
);

select * from Transportation
```



| | SKU | Transportation_modes | Routes | Costs |
|---|---|---|---|---|
| 1 | SKU0 | Road | Route B | 187.75 |
| 2 | SKU1 | Road | Route B | 503.07 |
| 3 | SKU2 | Air | Route C | 141.92 |
| 4 | SKU3 | Rail | Route A | 254.78 |
| 5 | SKU4 | Air | Route A | 923.44 |
| 6 | SKU5 | Road | Route A | 235.46 |
| 7 | SKU6 | Sea | Route A | 134.37 |
| 8 | SKU7 | Road | Route C | 802.06 |
| 9 | SKU8 | Sea | Route B | 505.56 |
| 10 | SKU9 | Rail | Route B | 995.93 |
| 11 | SKU10 | Road | Route B | 806.10 |
| 12 | SKU11 | Air | Route A | 126.72 |
| 13 | SKU12 | Road | Route B | 402.97 |
| 14 | SKU13 | Road | Route B | 547.24 |
| 15 | SKU14 | Air | Route B | 929.24 |
| 16 | SKU15 | Sea | Route B | 127.86 |
| 17 | SKU16 | Air | Route A | 865.53 |

✓ Query executed successfully.

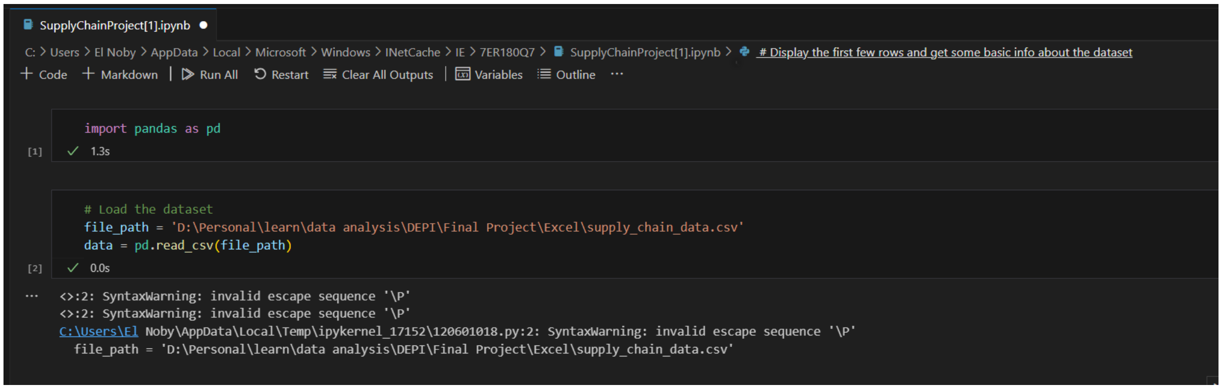
## 5. Data Relationships

Once the tables were populated, relationships were established among them, leveraging foreign keys to ensure referential integrity

# 6. Data Cleaning and Preprocessing in Python

## 6.1 Loading Data into Python

The CSV files were loaded into Python for data cleaning and preprocessing using the Pandas package:


SupplyChainProject[1].ipynb ●
C: > Users > El Noby > AppData > Local > Microsoft > Windows > INetCache > IE > 7ER180Q7 > SupplyChainProject[1].ipynb > # Display the first few rows and get some basic info about the dataset
+ Code
+ Markdown
Run All
Restart
Clear All Outputs
Variables
Outline
···
[1]
✓ 1.3s
[2]
✓ 0.0s

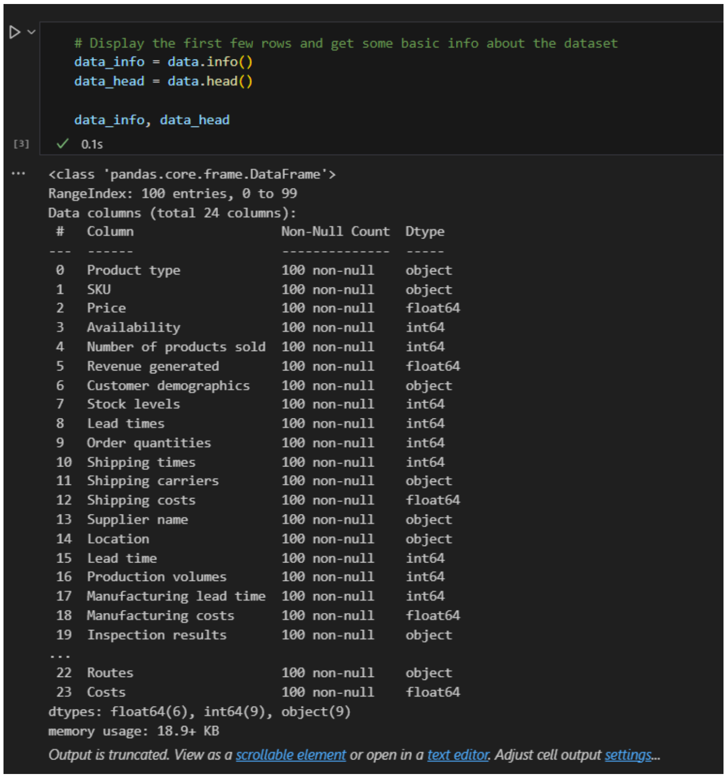
```
import pandas as pd
```

```
# Load the dataset
file_path = 'D:\Personal\learn\data analysis\DEPI\Final Project\Excel\supply_chain_data.csv'
data = pd.read_csv(file_path)
```

```
<>:2: SyntaxWarning: invalid escape sequence '\P'
<>:2: SyntaxWarning: invalid escape sequence '\P'
C:\Users\El Noby\AppData\Local\Temp\ipykernel_17152\120601018.py:2: SyntaxWarning: invalid escape sequence '\P'
  file_path = 'D:\Personal\learn\data analysis\DEPI\Final Project\Excel\supply_chain_data.csv'
```

## 6.2 Data Discovering

Data Discovering find the data types and columns heads:

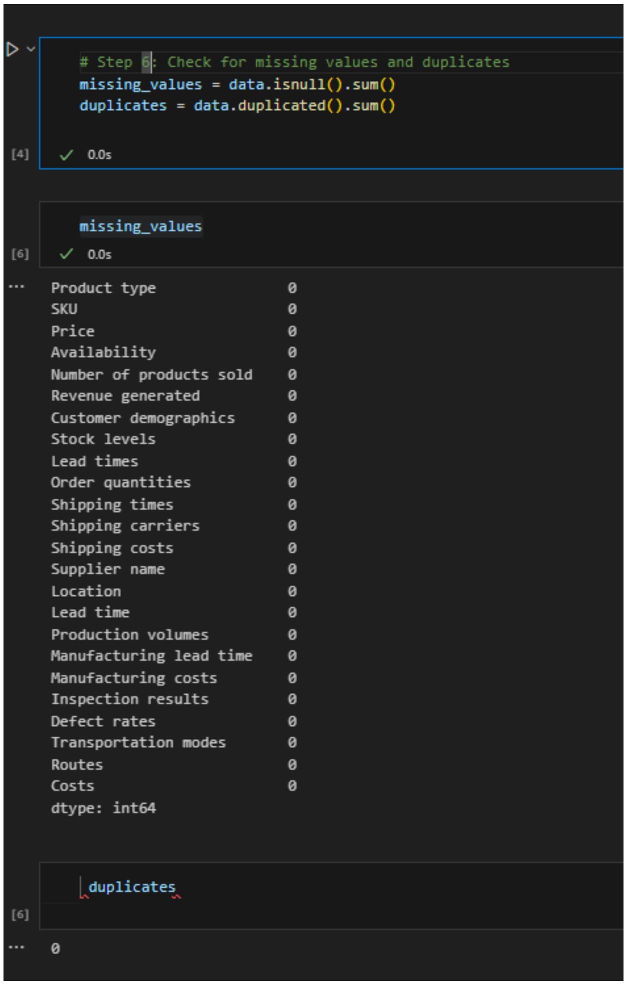


```
# Display the first few rows and get some basic info about the dataset
data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 24 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Product type             100 non-null    object
 1   SKU                      100 non-null    object
 2   Price                    100 non-null    float64
 3   Availability             100 non-null    int64
 4   Number of products sold  100 non-null    int64
 5   Revenue generated        100 non-null    float64
 6   Customer demographics    100 non-null    object
 7   Stock levels             100 non-null    int64
 8   Lead times               100 non-null    int64
 9   Order quantities         100 non-null    int64
 10  Shipping times           100 non-null    int64
 11  Shipping carriers        100 non-null    object
 12  Shipping costs           100 non-null    float64
 13  Supplier name            100 non-null    object
 14  Location                 100 non-null    object
 15  Lead time                100 non-null    int64
 16  Production volumes       100 non-null    int64
 17  Manufacturing lead time  100 non-null    int64
 18  Manufacturing costs      100 non-null    float64
 19  Inspection results       100 non-null    object
...
 22  Routes                   100 non-null    object
 23  Costs                    100 non-null    float64
dtypes: float64(6), int64(9), object(9)
memory usage: 18.9+ KB
```

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

Data Discovering find Nulls and Duplicated Values:



```
[4]
# Step 6: Check for missing values and duplicates
missing_values = data.isnull().sum()
duplicates = data.duplicated().sum()
✓ 0.0s
```

```
[6]
missing_values
✓ 0.0s
```

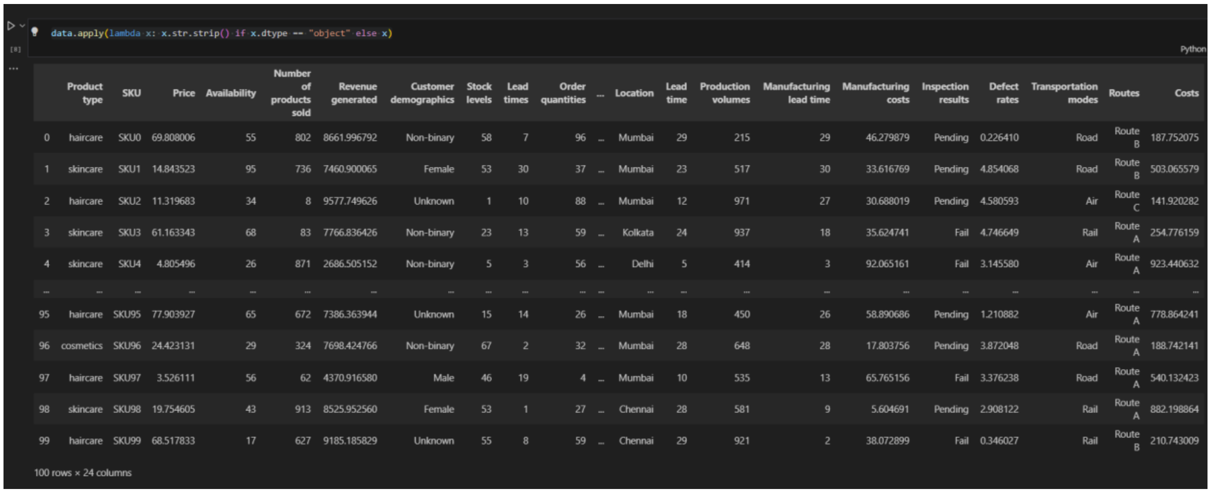
```
Product type                0
SKU                         0
Price                       0
Availability                0
Number of products sold     0
Revenue generated           0
Customer demographics       0
Stock levels                0
Lead times                  0
Order quantities            0
Shipping times              0
Shipping carriers           0
Shipping costs              0
Supplier name               0
Location                    0
Lead time                   0
Production volumes          0
Manufacturing lead time     0
Manufacturing costs         0
Inspection results          0
Defect rates                0
Transportation modes        0
Routes                      0
Costs                       0
dtype: int64
```

```
[6]
duplicates
```

```
0
```

## 6.3 Data Cleaning

Data cleaning involved stripping whitespace from string fields:



```
[8]
data.apply(lambda x: x.str.strip() if x.dtype == "object" else x)
```

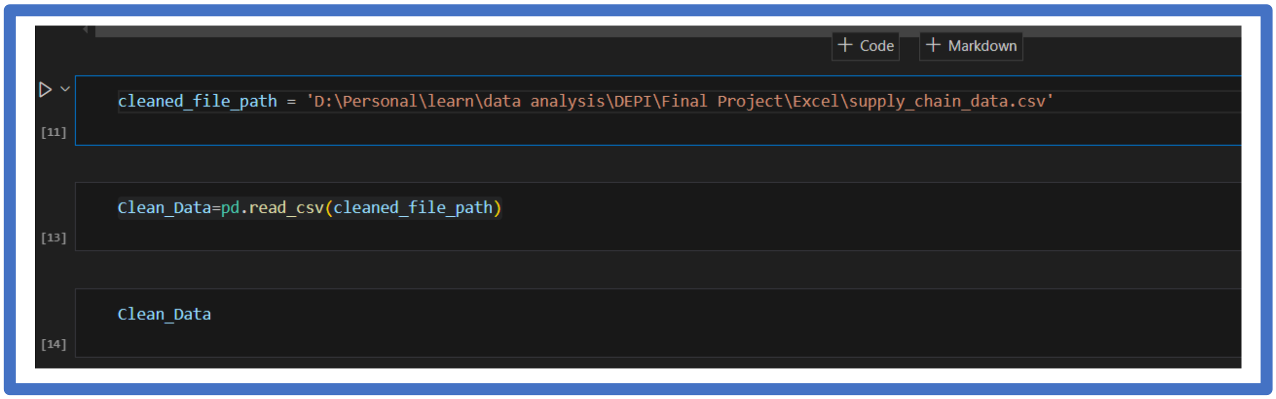
Python

|  | Product type | SKU | Price | Availability | Number of products sold | Revenue generated | Customer demographics | Stock levels | Lead times | Order quantities | ... | Location | Lead time | Production volumes | Manufacturing lead time | Manufacturing costs | Inspection results | Defect rates | Transportation modes | Routes | Costs |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | haircare | SKU0 | 69.808006 | 55 | 802 | 8661.996792 | Non-binary | 58 | 7 | 96 | ... | Mumbai | 29 | 215 | 29 | 46.279879 | Pending | 0.226410 | Road | Route B | 187.752075 |
| 1 | skincare | SKU1 | 14.843523 | 95 | 736 | 7460.900065 | Female | 53 | 30 | 37 | ... | Mumbai | 23 | 517 | 30 | 33.616769 | Pending | 4.854068 | Road | Route B | 503.065579 |
| 2 | haircare | SKU2 | 11.319683 | 34 | 8 | 9577.749626 | Unknown | 1 | 10 | 88 | ... | Mumbai | 12 | 971 | 27 | 30.688019 | Pending | 4.580593 | Air | Route C | 141.920282 |
| 3 | skincare | SKU3 | 61.163343 | 68 | 83 | 7766.836426 | Non-binary | 23 | 13 | 59 | ... | Kolkata | 24 | 937 | 18 | 35.624741 | Fail | 4.746649 | Rail | Route A | 254.776159 |
| 4 | skincare | SKU4 | 4.805496 | 26 | 871 | 2686.505152 | Non-binary | 5 | 3 | 56 | ... | Delhi | 5 | 414 | 3 | 92.065161 | Fail | 3.145580 | Air | Route A | 923.440632 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 95 | haircare | SKU95 | 77.903927 | 65 | 672 | 7386.363944 | Unknown | 15 | 14 | 26 | ... | Mumbai | 18 | 450 | 26 | 58.890686 | Pending | 1.210882 | Air | Route A | 778.864241 |
| 96 | cosmetics | SKU96 | 24.423131 | 29 | 324 | 7698.424766 | Non-binary | 67 | 2 | 32 | ... | Mumbai | 28 | 648 | 28 | 17.803756 | Pending | 3.872048 | Road | Route A | 188.742141 |
| 97 | haircare | SKU97 | 3.526111 | 56 | 62 | 4370.916580 | Male | 46 | 19 | 4 | ... | Mumbai | 10 | 535 | 13 | 65.765156 | Fail | 3.376238 | Road | Route A | 540.132423 |
| 98 | skincare | SKU98 | 19.754605 | 43 | 913 | 8525.952560 | Female | 53 | 1 | 27 | ... | Chennai | 28 | 581 | 9 | 5.604691 | Pending | 2.908122 | Rail | Route A | 882.198864 |
| 99 | haircare | SKU99 | 68.517833 | 17 | 627 | 9185.185829 | Unknown | 55 | 8 | 59 | ... | Chennai | 29 | 921 | 2 | 38.072899 | Fail | 0.346027 | Rail | Route B | 210.743009 |

100 rows × 24 columns

### 6.4 Saving the Cleaned Data

Finally, the cleaned data was saved to a new CSV file:



```
[11]
cleaned_file_path = 'D:\Personal\learn\data analysis\DEPI\Final Project\Excel\supply_chain_data.csv'

[13]
Clean_Data=pd.read_csv(cleaned_file_path)

[14]
Clean_Data
```

## 7. Conclusion

This documentation outlines the complete process from data extraction to cleaning and preprocessing. The structured approach ensures that data integrity is maintained, and the cleaned data is ready for analysis.