Part 1: Generating Dataset

Installing Required Modules...

Let's install the required modules inside the notebook kernel:

1. **Pandas**; 2. **NumPy**; 3. **Faker**; 4. **Matplotlib** and 5. **Seaborn**

```
#!pip install numpy pandas faker matplotlib seaborn
```

Step-1: Creating the Dataset...

As we've no predefined dataset, I'll create a dummy dataset at first step. So, let's import the required modules like **faker**, **pandas**, **numpy**, **random** and datetime to create the dataset.

```
from faker import Faker
import pandas as pd
import numpy as np
import random
from datetime import datetime
```

Step-2: Setting Random Seeds for Reproducibility...

We set a fixed `seed` for **Faker**, **NumPy** and **Random** to make our random data `reproducible`. This means we'll get the same fake names, prices, and values every time we run the code — super helpful for consistency, debugging, and sharing work with others.

Step-3: Initializing **Faker** and Preparing Options...

```
Faker.seed(42)
np.random.seed(42)
random.seed(42)
# 42 is just a fun, common choice – feel free to use any number!
```

We'll initialize the **Faker** object as **`fake`** and link some dummy product names to their categories and provide some realistic payment options and cities.

```
fake = Faker('bn_BD')  # Initializing Faker object with Bangla locale

# Initializing lists to hold the generated data
products = ["Laptop", "Smartphone", "T-shirt", "Jeans", "Detergent",
            "Toothpaste", "Electric Kettle", "Rice Cooker", "Notebook
", "Pen"]

categories = {
    "Laptop": ["Electronics", np.random.randint(50000, 150000)],
    "Smartphone": ["Electronics", np.random.randint(10000, 100000)],
    "T-shirt": ["Clothing", np.random.randint(500, 3000)],
    "Jeans": ["Clothing", np.random.randint(1000, 5000)],
    "Detergent": ["Household", random.choice([50, 80, 150, 700])],
    "Toothpaste": ["Household", random.choice([30, 50, 100, 200])],
    "Electric Kettle": ["Appliances", np.random.randint(1000, 5000)],
    "Rice Cooker": ["Appliances", np.random.randint(2000, 10000)],
    "Notebook": ["Stationery", np.random.randint(50, 500)],
    "Pen": ["Stationery", np.random.randint(10, 100)],
}   # dictionary to map products to categories and prices in BDT

payment_methods = ["Credit Card", "Debit Card", "Mobile Payment", "Cas
h on Delivery"]
cities = ["Dhaka", "Chittagong", "Khulna", "Rajshahi", "Sylhet", "Myme
nsing", "Rangpur", "Barishal"]

# print(categories)  # Displaying the categories dictionary
```

Here are the list to visualize as human:

Products We'll Be Using To make our fake sales dataset feel realistic, we're including a variety of common items people actually buy. Here's our selection of 10 products:

`Laptop, Smartphone` **`--> Electronics`**

`T-Shirt, Jeans` **`--> Clothings`**

`Detergent, Toothpaste` **`--> Household`**

`Rice Cooker, Electric Kettle` **`--> Appliances`**

`Notebook, Pen` **`--> Stationary`**

Payment methods available: **`Credit Card, Debit Card, Mobile Payment and COD.`**

To the all the **`divisional cities`** in Bangladesh.

Step-4: Creating 1000 of Entries...

Now we will create 1000 entries of random customer data using **`for`** `loop` and store those in a **list** named **`data`**.

```
data = [] # Initializing variable to hold the generated data

for _ in range(1000): # Generating 1000 records
    product = random.choice(products)
    category, price = categories[product]

    entry = {
        'Invoice ID': fake.uuid4(),
        'Date': fake.date_between(start_date='-1y', end_date='today'),
        'Customer Name': fake.name(),
        'Customer Email': fake.email(),
        'Product' : product,
        'Category': category,
        'Quantity': random.randint(1, 24),
        'Price Per Unit (BDT)': price,
        'Payment Method': random.choice(payment_methods),
        'City': random.choice(cities),
    }

    # Adding the entry to the data list
    data.append(entry)
```

Now, we'll use the **`pandas`** module to create a **dataframe** of **`data`** (previously created) called `df`. Then we'll calculate the **`Total Price`** by multiplying **`Quantity`** and **`Price per Unit (BDT)`** values for each entry and get them in a column. Then, we'll apply some cleaning processes to remove the blank data. Though we're using **`Faker`** module to create dataset, but it's a good practice to apply cleaning for real-life datasets. Finally, we'll write the dataframe to the **CSV** file called **`sales_data.csv`** and save it for further use.

```python
df = pd.DataFrame(data)  # Creating a DataFrame from the generated data
df['Total Price'] = df['Quantity'] * df['Price Per Unit (BDT)'] # Calculating total price

# Applyiing Cleaning Operations
df.dropna(inplace=True)  # Dropping any rows with missing values
df.drop_duplicates(inplace=True)  # Dropping duplicate rows
df['Date'] = pd.to_datetime(df['Date'])  # Converting 'Date' column to datetime format

df.to_csv('sales_data.csv', index=False)                                    # Saving the DataFrame to a CSV file

# Displaying the first few rows of the DataFrame
print("First 20 rows of the DataFrame:")
df.head(20)
```

0	bdd640fb-0667-4ad1-9c80-317fa3b1799d	2024-09-27	আশীষ চন্দ্র	debaaphiphaa@example.com	Detergent	Household	8	50	Debit Card	Khulna	400
1	37f8a88b-17fc-495a-87a0-ca6e0822e8f3	2024-09-11	অদৃতা সরকার	shaarminkhaanm@example.org	Smartphone	Electronics	22	10860	Credit Card	Rangpur	238920
2	cf36d58b-4737-4190-96da-1dac72ff5d2a	2025-05-16	চঞ্চল মোড়ল	ekraamul23@example.com	Laptop	Electronics	1	65795	Credit Card	Rajshahi	65795
3	18c26797-6142-4a7d-97be-31111a2a73ed	2024-10-30	মুনতাকিম হক	tnmy78@example.org	Jeans	Clothing	17	2130	Credit Card	Rajshahi	36210
4	142c3fe8-60e7-4113-ac1b-8ca1f91e1d4c	2025-01-13	মোস্তাফিজ সিনহা	priymkumaar@example.net	Notebook	Stationery	14	70	Debit Card	Barishal	980
5	fc377a4c-4a15-444d-85e7-ce8a3a578a8e	2024-07-13	কাফি জাহান	daacaaryy@example.net	Pen	Stationery	9	92	Credit Card	Khulna	828
6	5ec42e08-29a3-42e9-9d65-a441d58842de	2024-10-28	মনোজ পাণ্ডে	raayaashaaltaa@example.net	Electric Kettle	Appliances	11	2095	Mobile Payment	Khulna	23045
7	6123fdf7-7656-4f72-a9d4-beef3eabedcb	2024-09-25	হৈমন্তী দাশগুপ্তা	nyn30@example.com	Jeans	Clothing	11	2130	Credit Card	Chittagong	23430
8	3602f8ac-10f1-4c81-848a-aa9e66b2bc5b	2025-06-02	প্রিয়াঙ্কা দে	psaahaa@example.com	Electric Kettle	Appliances	4	2095	Mobile Payment	Mymensing	8380
9	3f22faf8-23be-401d-83cf-2fde24933b83	2025-03-29	হৈমন্তী মৃধা	xsin@example.org	Pen	Stationery	9	92	Credit Card	Barishal	828
10	827050a8-2369-4584-bf5e-9ff0ff50bde4	2024-12-21	সৌমিক চৌধুরী	shuklaaphaarihaa@example.org	Notebook	Stationery	4	70	Cash on Delivery	Chittagong	280
11	98ae4334-6c12-4ce8-ae34-0454cac5b68c	2024-07-07	মুশফিক হাকিম	kaaberi01@example.net	Notebook	Stationery	10	70	Mobile Payment	Rajshahi	700
12	444ea7c8-c039-4710-8976-e334e2817efd	2025-04-07	ঋষিতা চট্টোপাধ্যায়	priiti27@example.org	Smartphone	Electronics	2	10860	Debit Card	Sylhet	21720
13	b83cfe0b-e037-45ed-b8db-0672f42d47cc	2024-09-23	আরাধ্যা সরকার	ghossmaalihaa@example.net	Smartphone	Electronics	8	10860	Credit Card	Rangpur	86880
14	5fb8d16c-2720-497d-b2eb-d6899be578c7	2025-04-05	অর্ক বন্দ্যোপাধ্যায়	hmrdhaa@example.com	Detergent	Household	15	50	Mobile Payment	Khulna	750
15	ce88cb2d-d4e8-4839-bc3e-058be0f3eab0	2024-10-10	অশোক পাল	daasrinaa@example.org	Toothpaste	Household	12	30	Debit Card	Sylhet	360
16	c40db9b4-885f-4e66-82b6-d2c5fa5d3100	2024-07-31	রাজু দত্ত	aarunni96@example.org	Smartphone	Electronics	20	10860	Debit Card	Rajshahi	217200
17	badcc32a-c159-4f53-8a0f-4efbedcd465e	2025-03-07	শেখর ব্যানার্জি	cbaagcii@example.net	T-shirt	Clothing	15	1794	Cash on Delivery	Sylhet	26910
18	3985c3cf-3f76-4e1d-9efa-21977394988f	2024-07-07	আকাশ কাদের	aakaashcndr@example.net	Notebook	Stationery	8	70	Mobile Payment	Dhaka	560
19	114125c6-3a9b-4dd4-8f12-59e0a18ff6b6	2025-05-30	মিলন তালুকদার	ekraamulhosaain@example.org	Jeans	Clothing	2	2130	Mobile Payment	Rangpur	4260

Some info about the data-set.

```
df.describe()  # Displaying summary statistics of the DataFrame
```

	Date	Quantity	Price Per Unit (BDT)	Total Price
count	1000	1000.000000	1000.000000	1.000000e+03
mean	2024-12-17 18:50:24	12.652000	8943.027000	1.050734e+05
min	2024-06-13 00:00:00	1.000000	30.000000	3.000000e+01
25%	2024-09-12 00:00:00	7.000000	70.000000	9.200000e+02
50%	2024-12-20 00:00:00	13.000000	2095.000000	1.731600e+04
75%	2025-03-24 00:00:00	19.000000	5772.000000	6.349200e+04
max	2025-06-13 00:00:00	24.000000	65795.000000	1.579080e+06
std	NaN	6.773535	19061.654886	2.549634e+05

Hurray! Data-set creation has been done. `sales_data.csv` appeared in the same directory of this file. Check it out...

Now let's move on to part 2 to analyze the data.

Part 2: Analyzing the Data

Step-1: Importing Required Module…

```
import pandas as pd
import matplotlib.pypot as plt
```

Step-2: loading the .csv File...

Loading the `sales_data.csv` file using `pandas` module and saving it to a variable `df`.

```
df = pd.read_csv('sales_data.csv')
df.head(10)
```

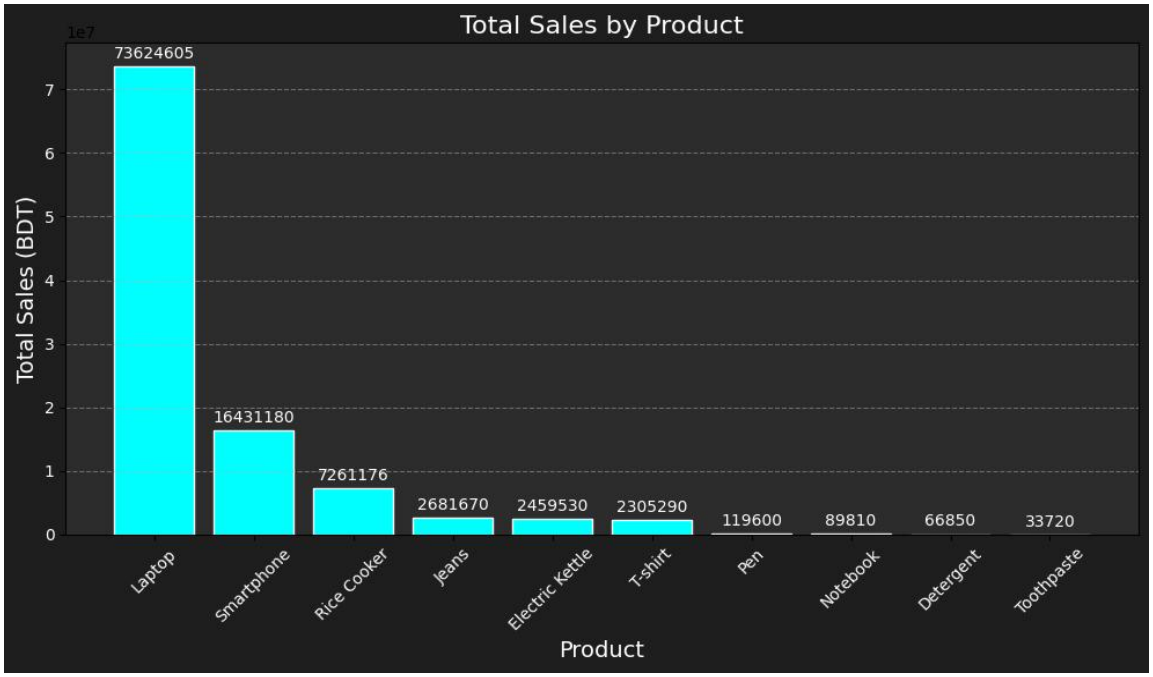
	Invoice ID	Date	Customer Name	Customer Email	Product	Category	Quantity	Price Per Unit (BDT)	Payment Method	City	Total Price
0	bdd640fb-0667-4ad1-9c80-317fa3b1799d	2024-09-27	আশীষ চন্দ্র	debaaphiphaa@example.com	Detergent	Household	8	50	Debit Card	Khulna	400
1	37f8a88b-17fc-495a-87a0-ca6e0822e8f3	2024-09-11	অদৃতা সরকার	shaarminkhaanm@example.org	Smartphone	Electronics	22	10860	Credit Card	Rangpur	238920
2	cf36d58b-4737-4190-96da-1dac72ff5d2a	2025-05-16	চঞ্চল মোড়ল	ekraamul23@example.com	Laptop	Electronics	1	65795	Credit Card	Rajshahi	65795
3	18c26797-6142-4a7d-97be-31111a2a73ed	2024-10-30	মুনতাকিম হক	tnmy78@example.org	Jeans	Clothing	17	2130	Credit Card	Rajshahi	36210
4	142c3fe8-60e7-4113-ac1b-8ca1f91e1d4c	2025-01-13	মোস্তাফিজ সিনহা	priymkumaar@example.net	Notebook	Stationery	14	70	Debit Card	Barishal	980
5	fc377a4c-4a15-444d-85e7-ce8a3a578a8e	2024-07-13	কাফি জাহান	daacaaryy@example.net	Pen	Stationery	9	92	Credit Card	Khulna	828
6	5ec42e08-29a3-42e9-9d65-a441d58842de	2024-10-28	মনোজ পাণ্ডে	raayaashaaltaa@example.net	Electric Kettle	Appliances	11	2095	Mobile Payment	Khulna	23045
7	6123fdf7-7656-4f72-a9d4-beef3eabedcb	2024-09-25	হৈমন্তী দাশগুপ্ত	nyn30@example.com	Jeans	Clothing	11	2130	Credit Card	Chittagong	23430
8	3602f8ac-10f1-4c81-848a-aa9e66b2bc5b	2025-06-02	প্রিয়াঙ্কা দে	psaahaa@example.com	Electric Kettle	Appliances	4	2095	Mobile Payment	Mymensing	8380
9	3f22faf8-23be-401d-83cf-2fde24933b83	2025-03-29	হৈমন্তী মৃধা	xsin@example.org	Pen	Stationery	9	92	Credit Card	Barishal	828

Step-3: Analyzing the Data

(a) Total Sales per Unit:

```
ax.set_facecolor('#2c2c2c')
```

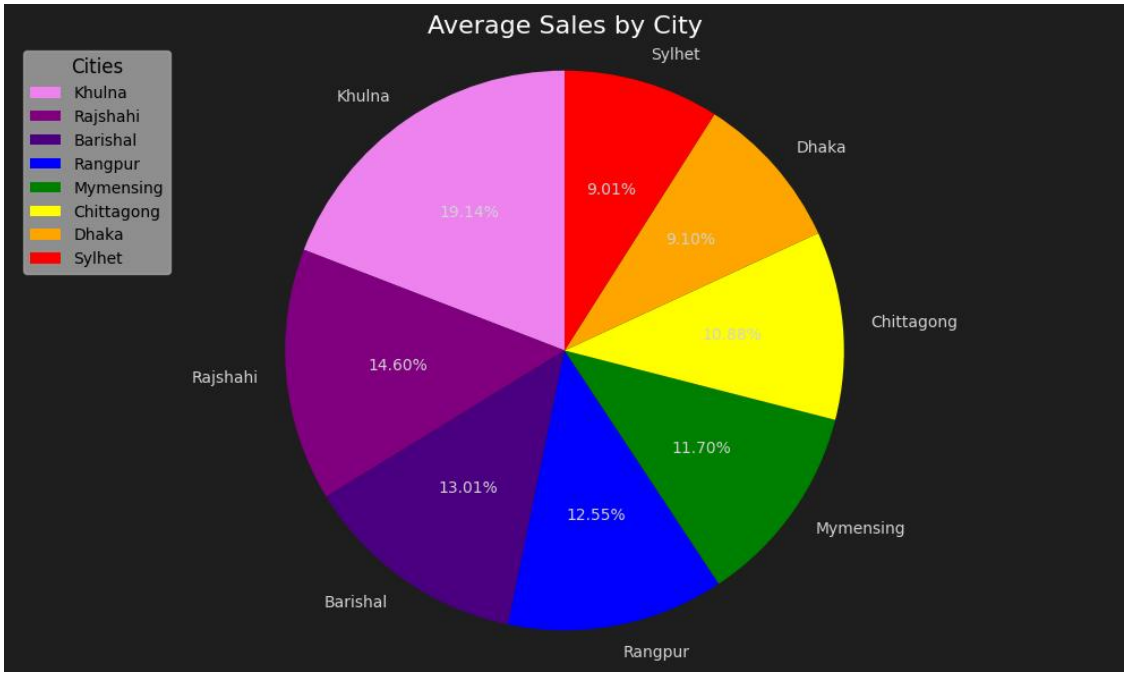
```
product_vs_total_sales = df.groupby('Product')['Total
Price'].sum().sort_values(ascending=False)

# Plotting the total sales for each product
fig, ax = plt.subplots(figsize=(10, 6))
fig.patch.set_facecolor('#1e1e1e')
fig.fontcolor = 'white'

# Set the style for the plot
bars = ax.bar(product_vs_total_sales.index,
product_vs_total_sales.values, color='cyan', edgecolor='white')
ax.bar_label(bars, fmt='%d', padding=3, fontsize=10, color='white')

# Customizing the plot
plt.title('Total Sales by Product', fontsize = 16, color = 'white')
plt.xlabel('Product', fontsize = 14, color = 'white')
plt.ylabel('Total Sales (BDT)', fontsize = 14, color = 'white')
plt.xticks(rotation = 45, color = 'white')
plt.yticks(color = 'white')
plt.grid(axis = 'y', linestyle = '--', alpha = 0.5)
plt.tight_layout()
plt.show()
```



(b) Average Quantities Sold per City:

```
city_vs_avg_sales = df.groupby('City')['Total
Price'].mean().round(2).sort_values(ascending=False)
print(city_vs_avg_sales)

# Plotting the average sales for each city
fig, ax = plt.subplots(figsize=(10, 6))
fig.patch.set_facecolor('#1e1e1e')
fig.fontcolor = 'white'

# Set the style for the plot
pie = ax.pie(city_vs_avg_sales, labels = city_vs_avg_sales.index,
            autopct = '%.2f%%', colors = ['violet', 'purple',
            'indigo', 'blue', 'green', 'yellow', 'orange', 'red'],
            startangle = 90, textprops={'color': 'lightgrey'})


ax.set_title('Average Sales by City', fontsize = 16, color='white')

# Customizing the plot
plt.axis('equal')
plt.legend(title='Cities', loc='upper left', fontsize=10,
title_fontsize='12', facecolor='white', framealpha=0.5)
plt.tight_layout()
plt.show()
```

```
City
Khulna       161337.47
Rajshahi     123090.68
Barishal     109632.34
Rangpur      105822.19
Mymensing     98650.09
Chittagong    91694.05
Dhaka         76723.02
Sylhet        75955.21
Name: Total Price, dtype: float64
```



(c) Categorizing with Pricing Weight:

```
df['Order Type'] = df['Total Price'].apply(
                        lambda x:
'                             High' if x > 100000
                              else 'Medium' if x > 50000
                              else 'Low' if x > 10000 else 'Very Low'
                        )
df['Order Type'].value_counts().sort_values(ascending = True)
```

```
Order Type
Medium       75
High        199
Low         286
Very Low    440
Name: count, dtype: int64
```

(d) Extracting Top 20 Buyers Info:

```
info = df.sort_values(by='Total Price', ascending=False)

df.reset_index(drop=True, inplace=True)
# Resetting index after sorting

info.head(20) # Displaying top 20 orders by total price
```

	Invoice ID	Date	Customer Name	Customer Email	Product	Category	Quantity	Price Per Unit (BDT)	Payment Method	City	Total Price	Order Type
681	699fdb02-bcdd-4b48-bcb3-c9fa191abb3d	2025-05-16	জয়দীপ দাশগুপ্তা	kaaderaaphiyaa@example.com	Laptop	Electronics	24	65795	Debit Card	Barishal	1579080	High
891	5b9e75f4-12e0-4b9c-8288-49d95ce195be	2024-06-24	অদিতি প্রামানিক	priyti39@example.com	Laptop	Electronics	23	65795	Cash on Delivery	Chittagong	1513285	High
977	aa546c79-bf3f-41d5-a557-c82c488c6ea4	2025-05-08	অহনা হক	muntaasir81@example.com	Laptop	Electronics	22	65795	Credit Card	Chittagong	1447490	High
567	953fcfd4-6069-4b6d-84b5-1c12ab132032	2025-05-30	মাধু কান্ত	jynt04@example.org	Laptop	Electronics	22	65795	Cash on Delivery	Dhaka	1447490	High
519	71e15b9f-8c6d-468e-85bc-f3baef52d36e	2024-08-04	সাব্বির প্রামানিক	hbegm@example.net	Laptop	Electronics	22	65795	Debit Card	Rajshahi	1447490	High
931	833334f6-6a6a-4149-8b44-676d2ac39be9	2024-07-24	বাসির আলম	phaujiyaakhaan@example.net	Laptop	Electronics	21	65795	Cash on Delivery	Sylhet	1381695	High
954	84bed7ed-1491-451d-a6a7-a6b0f9619ae2	2025-06-01	জগদীশ দত্ত	sbishbaas@example.com	Laptop	Electronics	21	65795	Debit Card	Rajshahi	1381695	High
630	d21ce315-8f5e-448e-b40f-040e0037447f	2024-10-30	আয়ুষ চৌধুরী	nishitaashuklaa@example.net	Laptop	Electronics	21	65795	Mobile Payment	Barishal	1381695	High
388	5fcd859e-d6e1-4109-97e1-74e6f76b9723	2025-01-17	আয়ুশি ব্যানার্জি	minyaapritm@example.com	Laptop	Electronics	21	65795	Cash on Delivery	Mymensing	1381695	High
221	1cd66b09-cf0e-4d2b-b15c-167a45d8a6ad	2024-10-19	রাজ পোদ্দার	aadipaal@example.com	Laptop	Electronics	21	65795	Debit Card	Khulna	1381695	High
274	905881df-ee9b-43e1-9d27-c09a5d71a898	2024-08-29	শ্রাবন্তী দাস	islaammnoj@example.com	Laptop	Electronics	20	65795	Mobile Payment	Rajshahi	1315900	High
933	3677f5b4-63a1-4caa-8311-362650f51e97	2024-12-08	অজিত পাণ্ডে	tnmy13@example.com	Laptop	Electronics	19	65795	Debit Card	Mymensing	1250105	High
596	5e5a90c1-52a4-43f9-8c9e-903ac4bd338e	2025-05-27	আনিস ছরাইয়া	gaanggulimisstti@example.org	Laptop	Electronics	19	65795	Debit Card	Khulna	1250105	High
183	65e58f34-5df0-4e8c-878a-aed9233ffc82	2024-10-13	আশ্বিনী পাল	moddlprdiip@example.com	Laptop	Electronics	19	65795	Mobile Payment	Barishal	1250105	High
259	af36b8cc-1308-4974-ac95-b3ac0beb7c34	2025-06-07	সাব্বির সিনহা	aalimuntaakim@example.org	Laptop	Electronics	19	65795	Credit Card	Barishal	1250105	High
962	fca54576-8523-43d7-9a5d-376e602ba256	2025-04-04	শাহজাহান মন্ডল	jobaaydaa65@example.com	Laptop	Electronics	19	65795	Debit Card	Chittagong	1250105	High
500	7e1f48c4-a9d6-490d-b4ad-fa22ad0205f5	2025-02-06	আরিয়ান মন্ডল	nusraat57@example.net	Laptop	Electronics	19	65795	Cash on Delivery	Khulna	1250105	High
64	4fdc6e1b-edcb-4cb6-8692-dc639424aed5	2025-01-22	কাবেরি মুখার্জি	prtyytaalukdaar@example.org	Laptop	Electronics	19	65795	Credit Card	Mymensing	1250105	High
485	fd0ee0b6-40f7-424b-bbd7-b4bf8e0eabb7	2024-10-10	বাঁধন মুখার্জি	sumaaiyaabndopaadhyaay@example.net	Laptop	Electronics	19	65795	Debit Card	Mymensing	1250105	High
777	f83bf029-b6ac-4048-8430-8e0c3fe604f4	2025-04-07	ইউসুফ হক	cdtt@example.com	Laptop	Electronics	18	65795	Cash on Delivery	Rajshahi	1184310	High

(d) Listing Products According to Their Quantities:

```
df.groupby('Product')['Quantity'].sum().sort_values(ascending=False)
```

```
Product
Smartphone      1513
Detergent       1337
Pen             1300
T-shirt         1285
Notebook        1283
```

```
Jeans             1259
Rice Cooker        1258
Electric Kettle    1174
Toothpaste         1124
Laptop            1119
Name: Quantity, dtype: int64
```

We can also do many more operations using the `pandas` and `matplotlib` modules and play with the dataset.