

task7

August 15, 2025

```
[1]: import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
```

```
[3]: import pandas as pd

csv_file = "Superstore.csv" # Replace with your file's actual name
df = pd.read_csv(csv_file, encoding='latin1')

df.head()
```

```
[3]:
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	\
0	1	CA-2013-152156	09-11-2013	12-11-2013	Second Class	CG-12520	
1	2	CA-2013-152156	09-11-2013	12-11-2013	Second Class	CG-12520	
2	3	CA-2013-138688	13-06-2013	17-06-2013	Second Class	DV-13045	
3	4	US-2012-108966	11-10-2012	18-10-2012	Standard Class	SO-20335	
4	5	US-2012-108966	11-10-2012	18-10-2012	Standard Class	SO-20335	

	Customer Name	Segment	Country	City	...	\
0	Claire Gute	Consumer	United States	Henderson	...	
1	Claire Gute	Consumer	United States	Henderson	...	
2	Darrin Van Huff	Corporate	United States	Los Angeles	...	
3	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	
4	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	

	Postal Code	Region	Product ID	Category	Sub-Category	\
0	42420	South	FUR-BO-10001798	Furniture	Bookcases	
1	42420	South	FUR-CH-10000454	Furniture	Chairs	
2	90036	West	OFF-LA-10000240	Office Supplies	Labels	
3	33311	South	FUR-TA-10000577	Furniture	Tables	
4	33311	South	OFF-ST-10000760	Office Supplies	Storage	

	Product Name	Sales	Quantity	\
0	Bush Somerset Collection Bookcase	261.9600	2	
1	Hon Deluxe Fabric Upholstered Stacking Chairs,...	731.9400	3	
2	Self-Adhesive Address Labels for Typewriters b...	14.6200	2	
3	Bretford CR4500 Series Slim Rectangular Table	957.5775	5	

4 Eldon Fold 'N Roll Cart System 22.3680 2

	Discount	Profit
0	0.00	41.9136
1	0.00	219.5820
2	0.00	6.8714
3	0.45	-383.0310
4	0.20	2.5164

[5 rows x 21 columns]

```
[4]: conn = sqlite3.connect("sales_data.db")
df.to_sql("sales", conn, if_exists="replace", index=False)
```

[4]: 9994

```
[5]: import pandas as pd

# See the first few rows to inspect column names
print(df.head())

# Or if you're querying directly from DB:
print(pd.read_sql_query("PRAGMA table_info(sales);", conn))
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	\
0	1	CA-2013-152156	09-11-2013	12-11-2013	Second Class	CG-12520	
1	2	CA-2013-152156	09-11-2013	12-11-2013	Second Class	CG-12520	
2	3	CA-2013-138688	13-06-2013	17-06-2013	Second Class	DV-13045	
3	4	US-2012-108966	11-10-2012	18-10-2012	Standard Class	SO-20335	
4	5	US-2012-108966	11-10-2012	18-10-2012	Standard Class	SO-20335	

	Customer Name	Segment	Country	City	...	\
0	Claire Gute	Consumer	United States	Henderson	...	
1	Claire Gute	Consumer	United States	Henderson	...	
2	Darrin Van Huff	Corporate	United States	Los Angeles	...	
3	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	
4	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	

	Postal Code	Region	Product ID	Category	Sub-Category	\
0	42420	South	FUR-BO-10001798	Furniture	Bookcases	
1	42420	South	FUR-CH-10000454	Furniture	Chairs	
2	90036	West	OFF-LA-10000240	Office Supplies	Labels	
3	33311	South	FUR-TA-10000577	Furniture	Tables	
4	33311	South	OFF-ST-10000760	Office Supplies	Storage	

	Product Name	Sales	Quantity	\
0	Bush Somerset Collection Bookcase	261.9600	2	

1	Hon Deluxe Fabric Upholstered Stacking Chairs,...	731.9400	3
2	Self-Adhesive Address Labels for Typewriters b...	14.6200	2
3	Bretford CR4500 Series Slim Rectangular Table	957.5775	5
4	Eldon Fold 'N Roll Cart System	22.3680	2

	Discount	Profit
0	0.00	41.9136
1	0.00	219.5820
2	0.00	6.8714
3	0.45	-383.0310
4	0.20	2.5164

[5 rows x 21 columns]

	cid	name	type	notnull	dflt_value	pk
0	0	Row ID	INTEGER	0	None	0
1	1	Order ID	TEXT	0	None	0
2	2	Order Date	TEXT	0	None	0
3	3	Ship Date	TEXT	0	None	0
4	4	Ship Mode	TEXT	0	None	0
5	5	Customer ID	TEXT	0	None	0
6	6	Customer Name	TEXT	0	None	0
7	7	Segment	TEXT	0	None	0
8	8	Country	TEXT	0	None	0
9	9	City	TEXT	0	None	0
10	10	State	TEXT	0	None	0
11	11	Postal Code	INTEGER	0	None	0
12	12	Region	TEXT	0	None	0
13	13	Product ID	TEXT	0	None	0
14	14	Category	TEXT	0	None	0
15	15	Sub-Category	TEXT	0	None	0
16	16	Product Name	TEXT	0	None	0
17	17	Sales	REAL	0	None	0
18	18	Quantity	INTEGER	0	None	0
19	19	Discount	REAL	0	None	0
20	20	Profit	REAL	0	None	0

```
[6]: query = """
SELECT Product_Name AS product,
       SUM(Qty) AS total_qty,
       SUM(Qty * Unit_Price) AS revenue
FROM sales
GROUP BY Product_Name
"""
```

```
[7]: print(df.head())
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	\
0	1	CA-2013-152156	09-11-2013	12-11-2013	Second Class	CG-12520	

1	2	CA-2013-152156	09-11-2013	12-11-2013	Second Class	CG-12520
2	3	CA-2013-138688	13-06-2013	17-06-2013	Second Class	DV-13045
3	4	US-2012-108966	11-10-2012	18-10-2012	Standard Class	SO-20335
4	5	US-2012-108966	11-10-2012	18-10-2012	Standard Class	SO-20335

	Customer Name	Segment	Country	City	...	\
0	Claire Gute	Consumer	United States	Henderson	...	
1	Claire Gute	Consumer	United States	Henderson	...	
2	Darrin Van Huff	Corporate	United States	Los Angeles	...	
3	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	
4	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	

	Postal Code	Region	Product ID	Category	Sub-Category	\
0	42420	South	FUR-BO-10001798	Furniture	Bookcases	
1	42420	South	FUR-CH-10000454	Furniture	Chairs	
2	90036	West	OFF-LA-10000240	Office Supplies	Labels	
3	33311	South	FUR-TA-10000577	Furniture	Tables	
4	33311	South	OFF-ST-10000760	Office Supplies	Storage	

	Product Name	Sales	Quantity	\
0	Bush Somerset Collection Bookcase	261.9600	2	
1	Hon Deluxe Fabric Upholstered Stacking Chairs,...	731.9400	3	
2	Self-Adhesive Address Labels for Typewriters b...	14.6200	2	
3	Bretford CR4500 Series Slim Rectangular Table	957.5775	5	
4	Eldon Fold 'N Roll Cart System	22.3680	2	

	Discount	Profit
0	0.00	41.9136
1	0.00	219.5820
2	0.00	6.8714
3	0.45	-383.0310
4	0.20	2.5164

[5 rows x 21 columns]

```
[14]: # 3. Run SQL query to get total quantity & revenue per product
query = """
SELECT "Product Name" AS product,
       SUM(Quantity) AS total_qty,
       SUM(Quantity * Sales) AS revenue
FROM sales
GROUP BY "Product Name"
"""
summary_df = pd.read_sql_query(query, conn)
```

```
[15]: # 4. Print results
print("Sales Summary:")
```

```
print(summary_df)
```

Sales Summary:

	product	total_qty	revenue
0	"While you Were Out" Message Book, One Form pe...	8	68.264
1	#10 Gummed Flap White Envelopes, 100/Box	11	125.552
2	#10 Self-Seal White Envelopes	10	463.562
3	#10 White Business Envelopes, 4 1/8 x 9 1/2	32	2820.600
4	#10- 4 1/8" x 9 1/2" Recycled Envelopes	37	1366.936
...
1836	iKross Bluetooth Portable Keyboard + Cell Phon...	24	3268.200
1837	iOttie HLCRI0102 Car Mount	12	879.560
1838	iOttie XL Car Mount	14	1695.152
1839	invisibleSHIELD by ZAGG Smudge-Free Screen Pro...	29	1975.302
1840	netTALK DUO VoIP Telephone Service	26	5668.920

[1841 rows x 3 columns]

```
[16]: # 5. Plot simple bar chart for revenue per product
plt.figure(figsize=(8, 5))
plt.bar(summary_df['product'], summary_df['revenue'], color='skyblue')
plt.xlabel("Product")
plt.ylabel("Revenue")
plt.title("Revenue by Product")
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-1654925963.py:7: UserWarning: Glyph 148 (\x94) missing from
font(s) DejaVu Sans.
```

```
plt.tight_layout()
```

```
/tmp/ipython-input-1654925963.py:7: UserWarning: Glyph 147 (\x93) missing from
font(s) DejaVu Sans.
```

```
plt.tight_layout()
```

```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151:
```

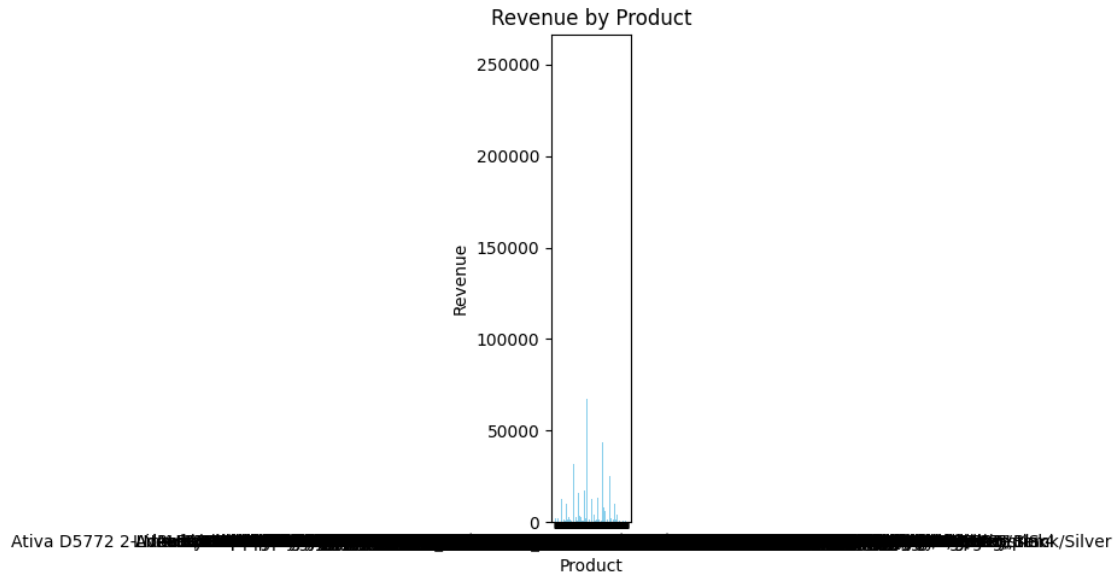
```
UserWarning: Glyph 148 (\x94) missing from font(s) DejaVu Sans.
```

```
fig.canvas.print_figure(bytes_io, **kw)
```

```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151:
```

```
UserWarning: Glyph 147 (\x93) missing from font(s) DejaVu Sans.
```

```
fig.canvas.print_figure(bytes_io, **kw)
```



[18]: *# Execute the SQL query and store the result in a DataFrame*

```
query = """
SELECT "Product Name" AS product,
       SUM(Quantity) AS total_qty,
       SUM(Quantity * Sales) AS revenue
FROM sales
GROUP BY "Product Name"
"""
summary_df = pd.read_sql_query(query, conn)
```

```
# Display the resulting DataFrame
display(summary_df)
```

	product	total_qty	revenue
0	"While you Were Out" Message Book, One Form pe...	8	68.264
1	#10 Gummed Flap White Envelopes, 100/Box	11	125.552
2	#10 Self-Seal White Envelopes	10	463.562
3	#10 White Business Envelopes, 4 1/8 x 9 1/2	32	2820.600
4	#10- 4 1/8" x 9 1/2" Recycled Envelopes	37	1366.936
...
1836	iKross Bluetooth Portable Keyboard + Cell Phon...	24	3268.200
1837	iOttie HLCRI0102 Car Mount	12	879.560
1838	iOttie XL Car Mount	14	1695.152
1839	invisibleSHIELD by ZAGG Smudge-Free Screen Pro...	29	1975.302
1840	netTALK DUO VoIP Telephone Service	26	5668.920

[1841 rows x 3 columns]

```
[20]: import sqlite3
import pandas as pd

# The column 'discounted_price' does not exist in the DataFrame.
# If you intended to perform an operation on another column, please
# specify the column name and the desired operation.

# df['discounted_price'] = df['discounted_price'].str.replace(' ', '',
    ↪ regex=False)
# df['discounted_price'] = pd.to_numeric(df['discounted_price'],
    ↪ errors='coerce')
```

```
[21]: # Assume each row = 1 quantity sold
df['quantity'] = 1
```

```
[23]: # Remove the duplicate 'quantity' column before writing to SQL
if 'quantity' in df.columns:
    df = df.drop(columns=['quantity'])

conn = sqlite3.connect("sales_data.db")
df.to_sql("sales", conn, if_exists="replace", index=False)
```

```
[23]: 9994
```

```
[25]: # SQL query with your actual column names
query = """
SELECT "Product Name" AS product,
       SUM(Quantity) AS total_qty,
       SUM(Quantity * Sales) AS revenue
FROM sales
GROUP BY "Product Name"
"""
summary_df = pd.read_sql_query(query, conn)
```

```
[26]: print(summary_df.head())
```

	product	total_qty	revenue
0	"While you Were Out" Message Book, One Form pe...	8	68.264
1	#10 Gummed Flap White Envelopes, 100/Box	11	125.552
2	#10 Self-Seal White Envelopes	10	463.562
3	#10 White Business Envelopes, 4 1/8 x 9 1/2	32	2820.600
4	#10- 4 1/8" x 9 1/2" Recycled Envelopes	37	1366.936

```
[28]: display(summary_df)
```

	product	total_qty	revenue
0	"While you Were Out" Message Book, One Form pe...	8	68.264

1	#10 Gummed Flap White Envelopes, 100/Box	11	125.552
2	#10 Self-Seal White Envelopes	10	463.562
3	#10 White Business Envelopes, 4 1/8 x 9 1/2	32	2820.600
4	#10- 4 1/8" x 9 1/2" Recycled Envelopes	37	1366.936
...
1836	iKross Bluetooth Portable Keyboard + Cell Phon...	24	3268.200
1837	iOttie HLCRI0102 Car Mount	12	879.560
1838	iOttie XL Car Mount	14	1695.152
1839	invisibleSHIELD by ZAGG Smudge-Free Screen Pro...	29	1975.302
1840	netTALK DUO VoIP Telephone Service	26	5668.920

[1841 rows x 3 columns]

```
[29]: import matplotlib.pyplot as plt
```

```
# Plot revenue by product
summary_df.plot(kind='bar', x='product', y='revenue', figsize=(10,5),
↳ legend=False)
plt.xlabel("Product")
plt.ylabel("Revenue ( )")
plt.title("Revenue by Product")
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-2387937019.py:8: UserWarning: Glyph 148 (\x94) missing from
font(s) DejaVu Sans.
```

```
plt.tight_layout()
```

```
/tmp/ipython-input-2387937019.py:8: UserWarning: Glyph 147 (\x93) missing from
font(s) DejaVu Sans.
```

```
plt.tight_layout()
```

```
/tmp/ipython-input-2387937019.py:8: UserWarning: Tight layout not applied. The
bottom and top margins cannot be made large enough to accommodate all Axes
decorations.
```

```
plt.tight_layout()
```

```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151:
```

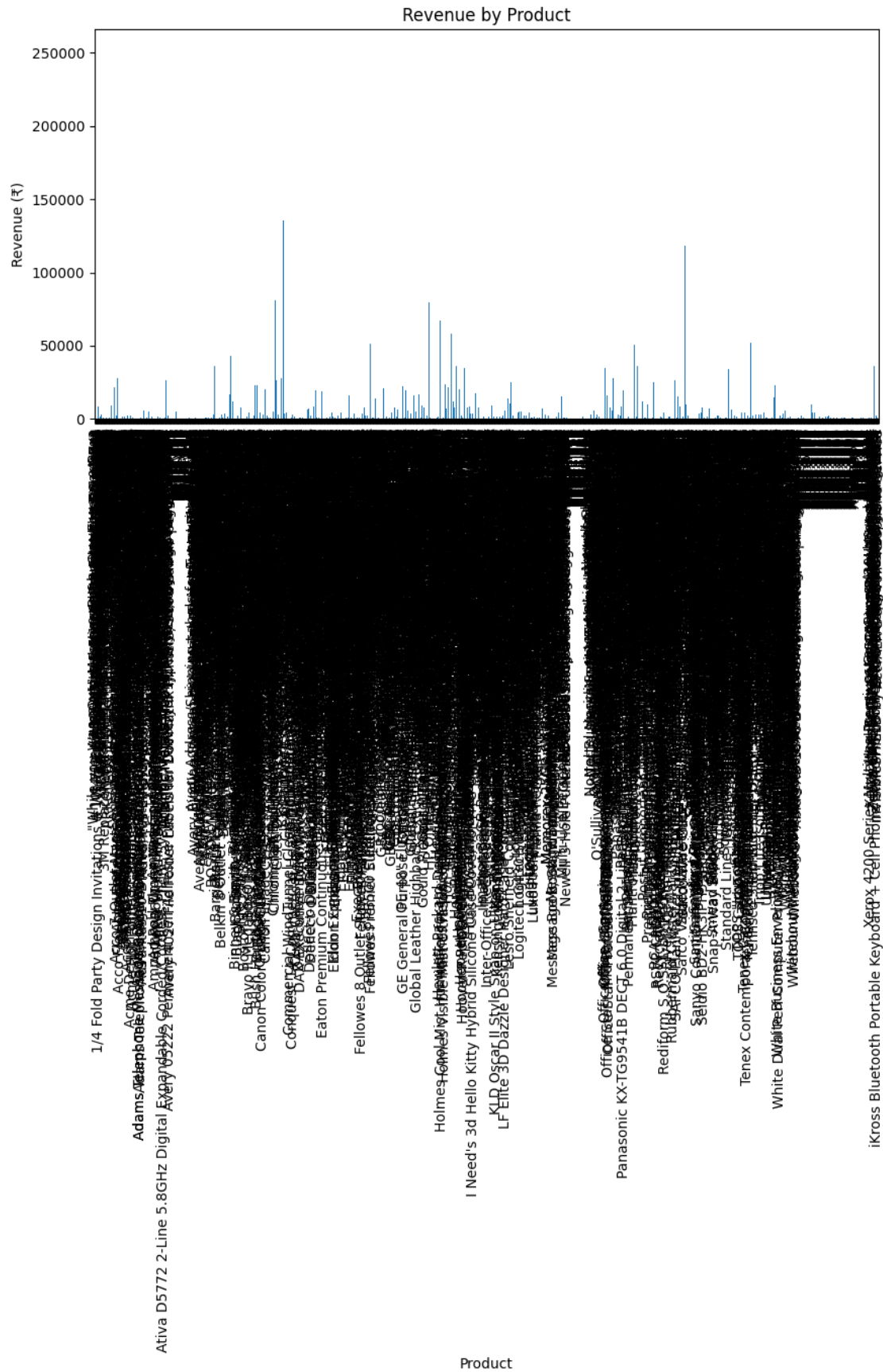
```
UserWarning: Glyph 148 (\x94) missing from font(s) DejaVu Sans.
```

```
fig.canvas.print_figure(bytes_io, **kw)
```

```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151:
```

```
UserWarning: Glyph 147 (\x93) missing from font(s) DejaVu Sans.
```

```
fig.canvas.print_figure(bytes_io, **kw)
```

```
[30]: import matplotlib.pyplot as plt
```

```
# Plot revenue by product
summary_df.plot(kind='bar', x='product', y='revenue', figsize=(10,5),
    ↪legend=False)
plt.xlabel("Product")
plt.ylabel("Revenue ( )")
plt.title("Revenue by Product")
plt.tight_layout()

# Save chart to a PNG file
plt.savefig("sales_chart.png") # Saved in Colab's /content directory
plt.show()
```

```
/tmp/ipython-input-274942349.py:8: UserWarning: Glyph 148 (\x94) missing from
font(s) DejaVu Sans.
```

```
plt.tight_layout()
```

```
/tmp/ipython-input-274942349.py:8: UserWarning: Glyph 147 (\x93) missing from
font(s) DejaVu Sans.
```

```
plt.tight_layout()
```

```
/tmp/ipython-input-274942349.py:8: UserWarning: Tight layout not applied. The
bottom and top margins cannot be made large enough to accommodate all Axes
decorations.
```

```
plt.tight_layout()
```

```
/tmp/ipython-input-274942349.py:11: UserWarning: Glyph 148 (\x94) missing from
font(s) DejaVu Sans.
```

```
plt.savefig("sales_chart.png") # Saved in Colab's /content directory
```

```
/tmp/ipython-input-274942349.py:11: UserWarning: Glyph 147 (\x93) missing from
font(s) DejaVu Sans.
```

```
plt.savefig("sales_chart.png") # Saved in Colab's /content directory
```

```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151:
```

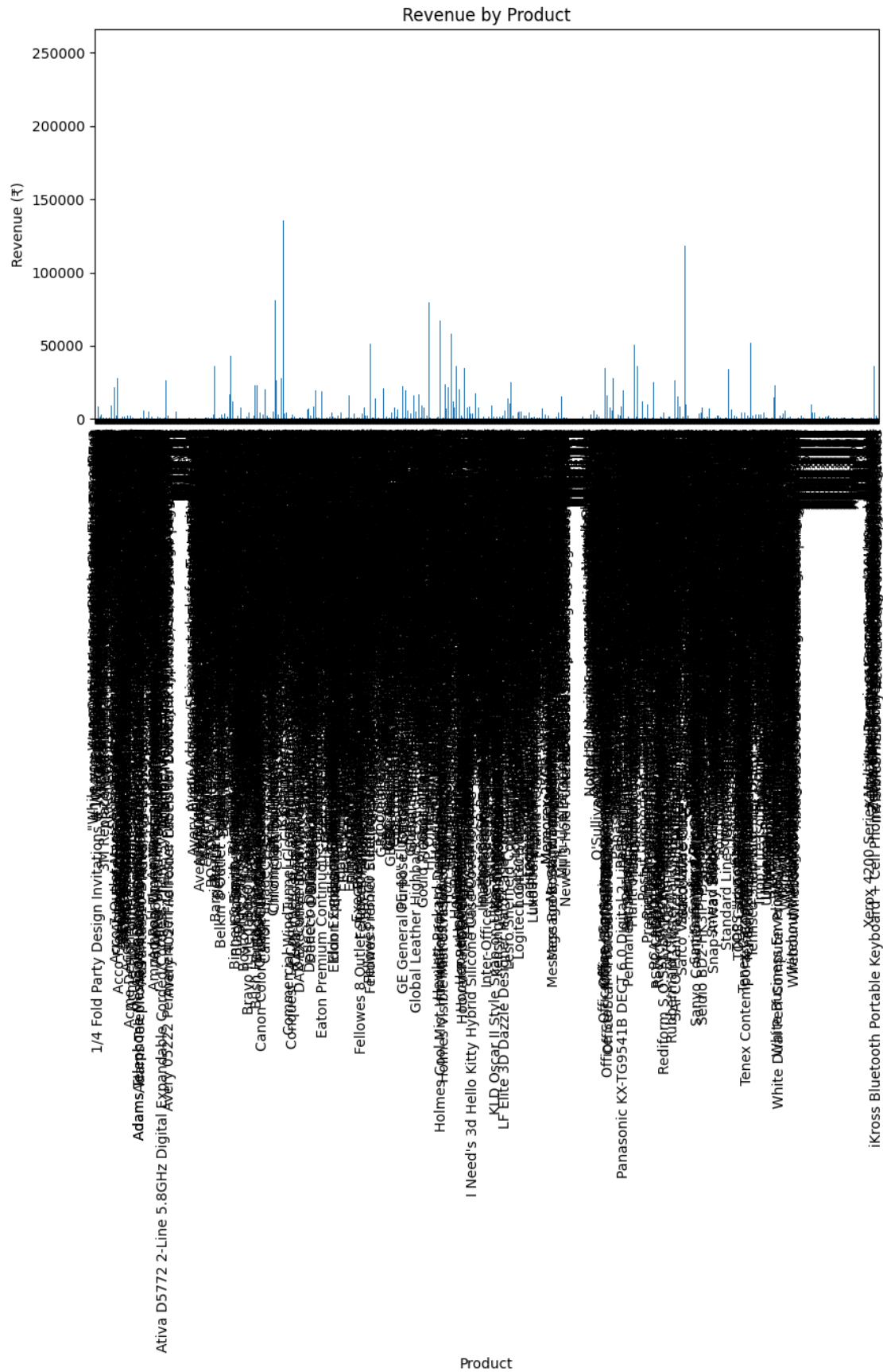
```
UserWarning: Glyph 148 (\x94) missing from font(s) DejaVu Sans.
```

```
fig.canvas.print_figure(bytes_io, **kw)
```

```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151:
```

```
UserWarning: Glyph 147 (\x93) missing from font(s) DejaVu Sans.
```

```
fig.canvas.print_figure(bytes_io, **kw)
```



```
[31]: from google.colab import files  
files.download("sales_chart.png")
```

<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>

```
[32]: conn.close()
```