

Quantum-task1.ipynb • Untitled-1.ipynb •

D: > Data\_Analyst\_Quantium > Quantum-task1.ipynb > transaction\_data.head()

+ Code + Markdown | ▶ Run All ↺ Restart 🗑 Clear All Outputs | 📄 Variables 📖 Outline ...

Python 3.12.4

```
import pandas as pd
import numpy as np
import seaborn as sns
```

[1] ✓ 1.5s Python

```
file_path = "D:/Data_Analyst_Quantium/"
transaction_data = pd.read_excel(file_path + "QVI_transaction_data.xlsx")
```

[2] ✓ 17.8s Python

```
transaction_data.head()
```

[3] ✓ 0.0s Python

...

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES
0	43390	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0
1	43599	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3
2	43605	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9
3	43329	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0
4	43330	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8

Quantum-task1.ipynb • Untitled-1.ipynb •

D: > Data\_Analyst\_Quantium > Quantum-task1.ipynb > customer\_data.head()

+ Code + Markdown | ▶ Run All ↺ Restart 🗑 Clear All Outputs | 📄 Variables 📖 Outline ...

Python 3.12.4

```
customer_data = pd.read_csv(file_path + "QVI_purchase_behaviour.csv")
```

[4] ✓ 0.0s Python

```
customer_data.head()
```

[5] ✓ 0.0s Python

...

	LYLTY_CARD_NBR	LIFESTAGE	PREMIUM_CUSTOMER
0	1000	YOUNG SINGLES/COUPLES	Premium
1	1002	YOUNG SINGLES/COUPLES	Mainstream
2	1003	YOUNG FAMILIES	Budget
3	1004	OLDER SINGLES/COUPLES	Mainstream
4	1005	MIDAGE SINGLES/COUPLES	Mainstream

```
[6] #SUMMARIZE DATASET
✓ 0.0s Python
```

```
[7] transaction_data.describe()
✓ 0.0s Python
```

```
...
```

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_QTY	TOT_SALES
count	264836.000000	264836.000000	2.648360e+05	2.648360e+05	264836.000000	264836.000000	264836.000000
mean	43464.036260	135.08011	1.355495e+05	1.351583e+05	56.583157	1.907309	7.304200
std	105.389282	76.78418	8.057998e+04	7.813303e+04	32.826638	0.643654	3.083226
min	43282.000000	1.00000	1.000000e+03	1.000000e+00	1.000000	1.000000	1.500000
25%	43373.000000	70.00000	7.002100e+04	6.760150e+04	28.000000	2.000000	5.400000
50%	43464.000000	130.00000	1.303575e+05	1.351375e+05	56.000000	2.000000	7.400000
75%	43555.000000	203.00000	2.030942e+05	2.027012e+05	85.000000	2.000000	9.200000
max	43646.000000	272.00000	2.373711e+06	2.415841e+06	114.000000	200.000000	650.000000

[+ Code](#) [+ Markdown](#)

```
[8] transaction_data.isnull().sum()
✓ 0.0s Python
```

```
...
```

DATE	0
STORE_NBR	0
LYLTY_CARD_NBR	0
TXN_ID	0
PROD_NBR	0
PROD_NAME	0
PROD_QTY	0
TOT_SALES	0

dtype: int64

```
[9] data_types = transaction_data.dtypes
print(data_types)
✓ 0.0s Python
```

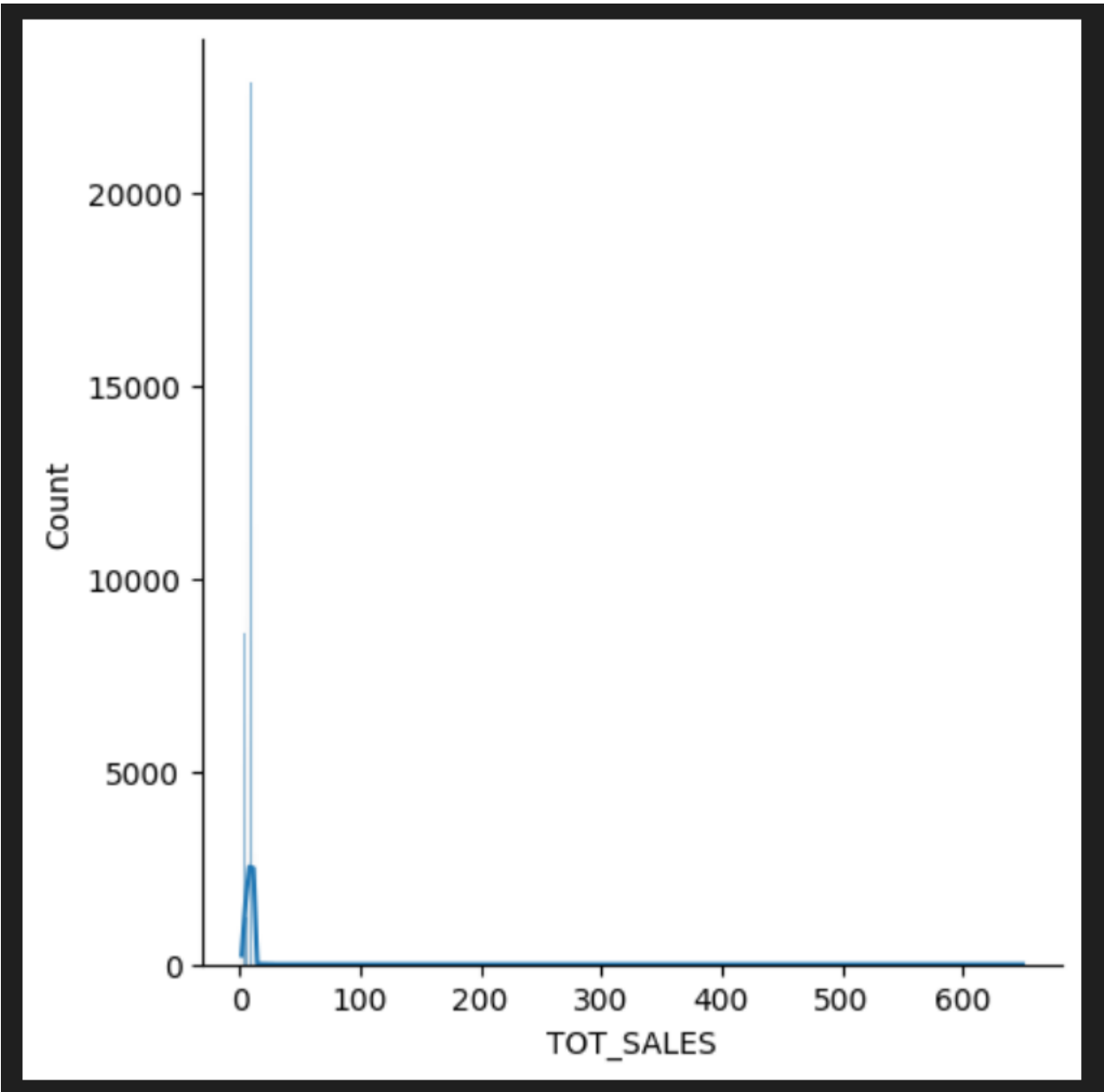
```
...
```

DATE	int64
STORE_NBR	int64
LYLTY_CARD_NBR	int64
TXN_ID	int64
PROD_NBR	int64
PROD_NAME	object
PROD_QTY	int64
TOT_SALES	float64

Spaces: 4 CRLF Cell 4 of 16

```
import matplotlib.pyplot as plt
import seaborn as sns
✓ 0.0s Python
```

```
sns.displot(transaction_data.TOT_SALES, kde = True)
✓ 4.7s Python
```



```
numericdata = transaction_data.select_dtypes(['float', 'int'])
numericdata.head()
```

13]

✓ 0.0s

Python

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_QTY	TOT_SALES
0	43390	1	1000	1	5	2	6.0
1	43599	1	1307	348	66	3	6.3
2	43605	1	1343	383	61	2	2.9
3	43329	2	2373	974	69	5	15.0
4	43330	2	2426	1038	108	3	13.8

```
x = numericdata[numericdata['TOT_SALES'] < 8.000]
```

14]

✓ 0.0s

Python

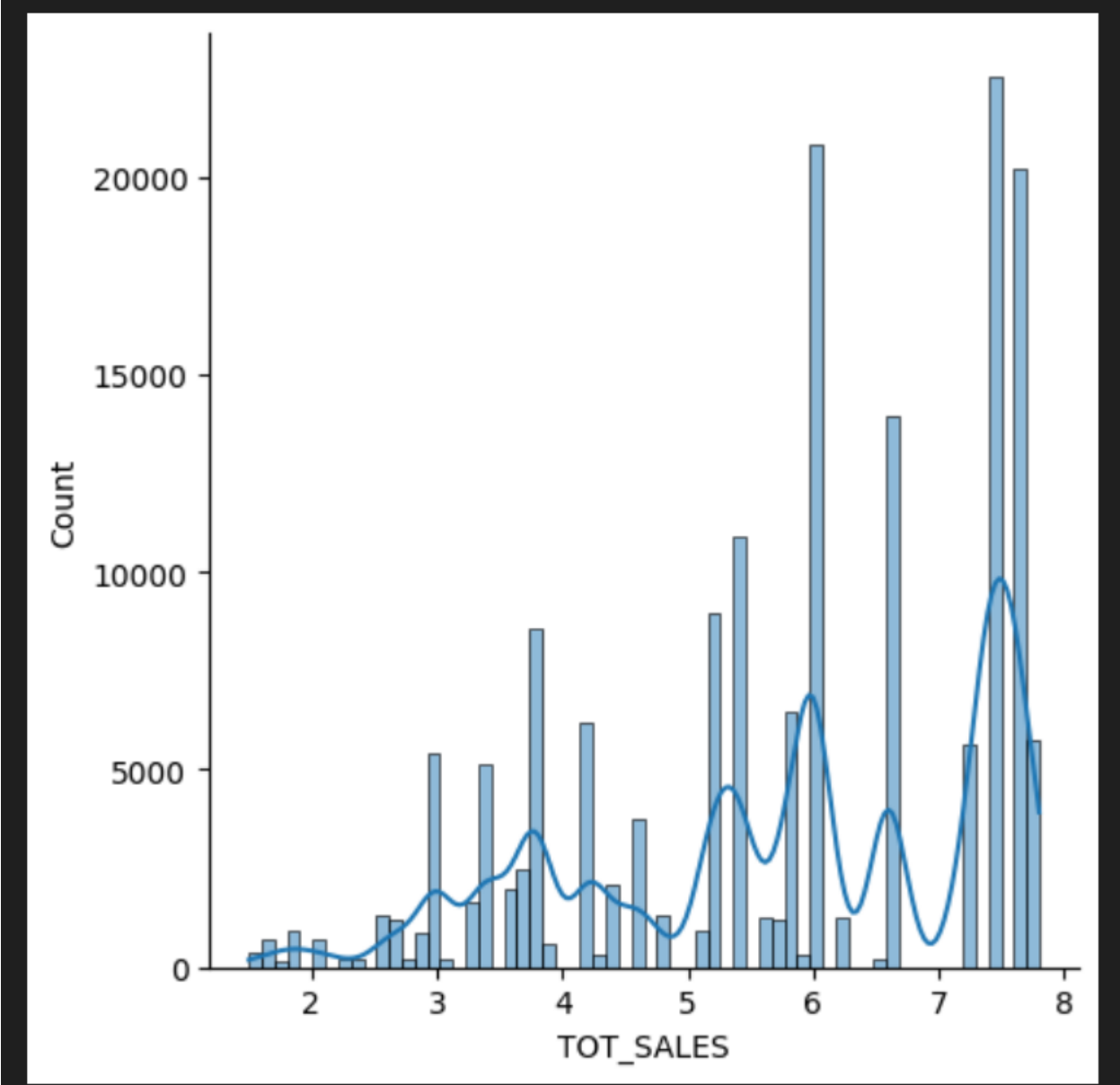
```
sns.displot(x.TOT_SALES, kde = True)
```

15]

✓ 1.7s

Python

<seaborn.axisgrid.FacetGrid at 0x19e90ac3980>



```
sns.boxplot(x.TOT_SALES)
```

✓ 0.2s

Py

<Axes: ylabel='TOT\_SALES'>

