Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Academic Year (2024-2025)

Name:	Anika Shah
Roll No:	C020
SAP ID:	60004230001

Experiment No: 2

AIM: Cleaning and Visualizing a dataset using Python (Matplotlib).

Attributes of the Dataset (List the attributes):

1. Invoice ID
2. Branch
3. City
4. Customer type
5. Gender
6. Product line
7. Unit price
8. Quantity
9. Tax 5%
10. Total
11. Date
12. Time
13. Payment
14. cogs
15. gross margin percentage
16. gross income
17. Rating

Academic Year (2024-2025)

DATA CLEANING:

```
[1]: import pandas as pd
[2]: df=pd.read_csv('supermarket_sales _e2_1.csv')
[3]: df
```

[3]:

Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	cogs	gross margin percentage	gross income	Rating
750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7.0	26.1415	548.9715	01-05-2019	13:08	Ewallet	522.83	4.761905	26.1415	9.1
226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5.0	3.8200	80.2200	03-08-2019	10:29	Cash	76.40	4.761905	3.8200	9.6
631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7.0	16.2155	340.5255	03-03-2019	13:23	Credit card	324.31	4.761905	16.2155	7.4
123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8.0	23.2880	489.0480	1/27/2019	20:33	Ewallet	465.76	4.761905	23.2880	8.4
373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7.0	30.2085	634.3785	02-08-2019	10:37	Ewallet	604.17	4.761905	30.2085	5.3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
233-67-5758	C	Naypyitaw	Normal	Male	Health and beauty	40.35	1.0	2.0175	42.3675	1/29/2019	13:46	Ewallet	40.35	4.761905	2.0175	6.2
303-96-2227	B	Mandalay	Normal	Female	Home and lifestyle	97.38	10.0	48.6900	1022.4900	03-02-2019	17:16	Ewallet	973.80	4.761905	48.6900	4.4
727-02-1313	A	Yangon	Member	Male	Food and beverages	31.84	1.0	1.5920	33.4320	02-09-2019	13:22	Cash	31.84	4.761905	1.5920	7.7
347-56-2442	A	Yangon	Normal	Male	Home and lifestyle	65.82	1.0	3.2910	69.1110	2/22/2019	15:33	Cash	65.82	4.761905	3.2910	4.1
849-09-3807	A	Yangon	Member	Female	Fashion accessories	88.34	7.0	30.9190	649.2990	2/18/2019	13:28	Cash	618.38	4.761905	30.9190	6.6

ws × 17 columns

Academic Year (2024-2025)

1. Describe

```
[5]: df.describe()
```

[5]:

	Unit price	Quantity	Tax 5%	Total	cogs	gross margin percentage	gross income	Rating
count	1001.000000	998.000000	1006.000000	1006.000000	1006.000000	1.006000e+03	1005.000000	1006.000000
mean	55.674386	5.506012	15.376767	322.912105	307.535338	4.761905e+00	15.380388	6.972465
std	26.431465	2.929027	11.700532	245.711172	234.010640	6.220341e-14	11.705793	1.716315
min	10.080000	1.000000	0.508500	10.678500	10.170000	4.761905e+00	0.508500	4.000000
25%	32.900000	3.000000	5.864625	123.157125	117.292500	4.761905e+00	5.834500	5.500000
50%	55.500000	5.000000	12.088000	253.848000	241.760000	4.761905e+00	12.096000	6.950000
75%	77.680000	8.000000	22.563750	473.838750	451.275000	4.761905e+00	22.588000	8.475000
max	99.960000	10.000000	49.650000	1042.650000	993.000000	4.761905e+00	49.650000	10.000000

2. Sort Values

```
[56]: df.sort_values('Rating', ascending=False)
```

[56]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	...	Time	Payment	cogs	gross margin percentage	gross income	Rating
66	347-34-2234	B	Mandalay	Member	Female	Sports and travel	55.07	9.0	24.7815	520.4115	...	13:40	Ewallet	495.63	4.761905	24.7815	10.0
64	285-68-5083	C	Naypyitaw	Member	Female	Sports and travel	24.74	3.0	3.7110	77.9310	...	17:47	Credit card	74.22	4.761905	3.7110	10.0
859	866-70-2814	B	Mandalay	Normal	Female	Electronic accessories	52.79	10.0	26.3950	554.2950	...	11:58	Ewallet	527.90	4.761905	26.3950	10.0
393	725-56-0833	A	Yangon	Normal	Female	Health and beauty	32.32	10.0	16.1600	339.3600	...	16:49	Credit card	323.20	4.761905	16.1600	10.0
165	423-57-2993	B	Mandalay	Normal	Male	Sports and travel	93.39	6.0	28.0170	588.3570	...	19:18	Ewallet	560.34	4.761905	28.0170	10.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
796	651-96-5970	A	Yangon	Normal	Male	Fashion accessories	46.41	1.0	2.3205	48.7305	...	20:06	Credit card	46.41	4.761905	2.3205	4.0
77	510-95-6347	B	Mandalay	Member	Female	Food and beverages	48.52	NaN	7.2780	152.8380	...	18:17	Ewallet	145.56	4.761905	7.2780	4.0
678	576-31-4774	B	Mandalay	Normal	Female	Health and beauty	73.41	3.0	11.0115	231.2415	...	13:10	Ewallet	220.23	4.761905	11.0115	4.0
232	836-82-5858	B	Mandalay	Member	Male	Health and beauty	69.37	9.0	31.2165	655.5465	...	19:14	Ewallet	624.33	4.761905	31.2165	4.0
385	182-69-8360	B	Mandalay	Normal	Female	Electronic accessories	23.65	4.0	4.7300	99.3300	...	13:32	Credit card	94.60	4.761905	4.7300	4.0

006 rows × 21 columns

3. Display first 5 rows and last 5 rows.

```
[58]: df.head()
```

[58]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	...	Time	Payment	cogs	gross margin percentage	gross income	Rating
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7.0	26.1415	548.9715	...	13:08	Ewallet	522.83	4.761905	26.1415	9.
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5.0	3.8200	80.2200	...	10:29	Cash	76.40	4.761905	3.8200	9.8
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7.0	16.2155	340.5255	...	13:23	Credit card	324.31	4.761905	16.2155	7.4
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8.0	23.2880	489.0480	...	20:33	Ewallet	465.76	4.761905	23.2880	8.4
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7.0	30.2085	634.3785	...	10:37	Ewallet	604.17	4.761905	30.2085	5.3

5 rows × 21 columns

```
[60]: df.tail()
```

[60]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	...	Time	Payment	cogs	gross margin percentage	gross income	R
1001	233-67-5758	C	Naypyitaw	Normal	Male	Health and beauty	40.35	1.0	2.0175	42.3675	...	13:46	Ewallet	40.35	4.761905	2.0175	
1002	303-96-2227	B	Mandalay	Normal	Female	Home and lifestyle	97.38	10.0	48.6900	1022.4900	...	17:16	Ewallet	973.80	4.761905	48.6900	
1003	727-02-1313	A	Yangon	Member	Male	Food and beverages	31.84	1.0	1.5920	33.4320	...	13:22	Cash	31.84	4.761905	1.5920	
1004	347-56-2442	A	Yangon	Normal	Male	Home and lifestyle	65.82	1.0	3.2910	69.1110	...	15:33	Cash	65.82	4.761905	3.2910	
1005	849-09-3807	A	Yangon	Member	Female	Fashion accessories	88.34	7.0	30.9190	649.2990	...	13:28	Cash	618.38	4.761905	30.9190	

5 rows × 21 columns

4. Display Headers

```
[68]: print(df.columns)
```

```
Index(['Invoice ID', 'Branch', 'City', 'Customer type', 'Gender',
       'Product line', 'Unit price', 'Quantity', 'Tax 5%', 'Total', 'Date',
       'Time', 'Payment', 'cogs', 'gross margin percentage', 'gross income',
       'Rating', 'Product Price', 'day', 'month', 'year'],
      dtype='object')
```

5. Display columns

```
[6]: df.columns

[6]: Index(['Invoice ID', 'Branch', 'City', 'Customer type', 'Gender',
       'Product line', 'Unit price', 'Quantity', 'Tax 5%', 'Total', 'Date',
       'Time', 'Payment', 'cogs', 'gross margin percentage', 'gross income',
       'Rating'],
      dtype='object')
```

6. Read Row (To find the particular row use Integer Location).

```
[70]: df.iloc[4]

[70]: Invoice ID                      373-73-7910
      Branch                                 A
      City                              Yangon
      Customer type                     Normal
      Gender                              Male
      Product line             Sports and travel
      Unit price                         86.31
      Quantity                             7.0
      Tax 5%                           30.2085
      Total                           634.3785
      Date                 2019-02-08 00:00:00
      Time                               10:37
      Payment                          Ewallet
      cogs                              604.17
      gross margin percentage         4.761905
      gross income                     30.2085
      Rating                               5.3
      Product Price                     604.17
      day                                    8
      month                                  2
      year                                2019
      Name: 4, dtype: object
```

7. Read a specific Location.

```
[7]: df.iloc[2,3]

[7]: 'Normal'
```

8. Iterate through columns (for City)

```
[72]: for city in df['City'].unique():
          print(city)

      Yangon
      Naypyitaw
      Mandalay
```

9. Finding data from the dataset which is based on textual data not row index number.

```
[8]: df.loc[df.City == 'Yangon']
```

[8]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	cogs	gross margin percentage	gross income
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7.0	26.1415	548.9715	01-05-2019	13:08	Ewallet	522.83	4.761905	26.141
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7.0	16.2155	340.5255	03-03-2019	13:23	Credit card	324.31	4.761905	16.215
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8.0	23.2880	489.0480	1/27/2019	20:33	Ewallet	465.76	4.761905	23.288
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7.0	30.2085	634.3785	02-08-2019	10:37	Ewallet	604.17	4.761905	30.208
6	355-53-5943	A	Yangon	Member	Female	Electronic accessories	68.84	6.0	20.6520	433.6920	2/25/2019	14:36	Ewallet	413.04	4.761905	20.652
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
996	886-18-2897	A	Yangon	Normal	Female	Food and beverages	56.56	5.0	14.1400	296.9400	3/22/2019	19:06	Credit card	282.80	4.761905	14.140
998	745-74-0715	A	Yangon	Normal	Male	Electronic accessories	58.03	2.0	5.8030	121.8630	03-10-2019	20:46	Ewallet	116.06	4.761905	5.803
1003	727-02-1313	A	Yangon	Member	Male	Food and beverages	31.84	1.0	1.5920	33.4320	02-09-2019	13:22	Cash	31.84	4.761905	1.592
1004	347-56-2442	A	Yangon	Normal	Male	Home and lifestyle	65.82	1.0	3.2910	69.1110	2/22/2019	15:33	Cash	65.82	4.761905	3.291
1005	849-09-3807	A	Yangon	Member	Female	Fashion accessories	88.34	7.0	30.9190	649.2990	2/18/2019	13:28	Cash	618.38	4.761905	30.919

341 rows × 17 columns

10. Create a new calculated column (Product Price)

[12]:
```
df['Product Price']=df['Unit price']*df['Quantity']
```

[77]:
```
df.head(10)
```

[77]:

Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	...	Time	Payment	cogs	gross margin percentage	gross income	Rating	Product Price	day	month	year
Member	Female	Health and beauty	74.69	7.0	26.1415	548.9715	...	13:08	Ewallet	522.83	4.761905	26.1415	9.1	522.83	5	1	2019
Normal	Female	Electronic accessories	15.28	5.0	3.8200	80.2200	...	10:29	Cash	76.40	4.761905	3.8200	9.6	76.40	8	3	2019
Normal	Male	Home and lifestyle	46.33	7.0	16.2155	340.5255	...	13:23	Credit card	324.31	4.761905	16.2155	7.4	324.31	3	3	2019
Member	Male	Health and beauty	58.22	8.0	23.2880	489.0480	...	20:33	Ewallet	465.76	4.761905	23.2880	8.4	465.76	27	1	2019
Normal	Male	Sports and travel	86.31	7.0	30.2085	634.3785	...	10:37	Ewallet	604.17	4.761905	30.2085	5.3	604.17	8	2	2019
Normal	Male	Electronic accessories	85.39	7.0	29.8865	627.6165	...	18:30	Ewallet	597.73	4.761905	29.8865	4.1	597.73	25	3	2019
Member	Female	Electronic accessories	68.84	6.0	20.6520	433.6920	...	14:36	Ewallet	413.04	4.761905	20.6520	5.8	413.04	25	2	2019
Normal	Female	Home and lifestyle	73.56	10.0	36.7800	772.3800	...	11:38	Ewallet	735.60	4.761905	36.7800	8.0	735.60	24	2	2019
Member	Female	Health and beauty	36.26	2.0	3.6260	76.1460	...	17:15	Credit card	72.52	4.761905	3.6260	7.2	72.52	10	1	2019
Member	Female	Food and beverages	54.84	3.0	8.2260	172.7460	...	13:27	Credit card	164.52	4.761905	8.2260	5.9	164.52	20	2	2019

11. Drop a column.

[85]:
```
df['Discounted Price'] = df['Product Price'] * 0.9
df.columns
```

[85]:
```
Index(['Invoice ID', 'Branch', 'City', 'Customer type', 'Gender',
       'Product line', 'Unit price', 'Quantity', 'Tax 5%', 'Total', 'Date',
       'Time', 'Payment', 'cogs', 'gross margin percentage', 'gross income',
       'Rating', 'Product Price', 'day', 'month', 'year', 'Discounted Price'],
      dtype='object')
```

[87]:
```
df = df.drop(columns=['Discounted Price'])
df.columns
```

[87]:
```
Index(['Invoice ID', 'Branch', 'City', 'Customer type', 'Gender',
       'Product line', 'Unit price', 'Quantity', 'Tax 5%', 'Total', 'Date',
       'Time', 'Payment', 'cogs', 'gross margin percentage', 'gross income',
       'Rating', 'Product Price', 'day', 'month', 'year'],
      dtype='object')
```

12. List duplicated rows.

```
[21]: df[df.duplicated()]
```

[21]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	...	Time	Payment	cogs	gross margin percentage	gross income	Rating
17	656-95-9349	A	Yangon	Member	Female	Health and beauty	68.93	7.0	24.1255	506.6355	...	11:03	Credit card	482.51	4.761905	24.1255	4.6
43	232-16-2483	C	Naypyitaw	Member	Female	Sports and travel	68.12	1.0	3.4060	71.5260	...	12:28	Ewallet	68.12	4.761905	3.4060	6.8
46	232-16-2483	C	Naypyitaw	Member	Female	Sports and travel	68.12	1.0	3.4060	71.5260	...	12:28	Ewallet	68.12	4.761905	3.4060	6.8
56	370-41-7321	B	Mandalay	Member	Male	Health and beauty	56.69	9.0	25.5105	535.7205	...	17:24	Credit card	510.21	4.761905	25.5105	8.4
67	370-41-7321	B	Mandalay	Member	Male	Health and beauty	56.69	9.0	25.5105	535.7205	...	17:24	Credit card	510.21	4.761905	25.5105	8.4
107	480-63-2856	C	Naypyitaw	Normal	Male	Food and beverages	19.25	8.0	7.7000	161.7000	...	18:37	Ewallet	154.00	4.761905	7.7000	6.6

6 rows × 21 columns

13. Drop duplicate rows.

```
[22]: df.drop_duplicates()
```

[22]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	...	Time	Payment	cogs	gross margin percentage	gross income	R
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7.0	26.1415	548.9715	...	13:08	Ewallet	522.83	4.761905	26.1415	
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5.0	3.8200	80.2200	...	10:29	Cash	76.40	4.761905	3.8200	
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7.0	16.2155	340.5255	...	13:23	Credit card	324.31	4.761905	16.2155	
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8.0	23.2880	489.0480	...	20:33	Ewallet	465.76	4.761905	23.2880	
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7.0	30.2085	634.3785	...	10:37	Ewallet	604.17	4.761905	30.2085	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1001	233-67-5758	C	Naypyitaw	Normal	Male	Health and beauty	40.35	1.0	2.0175	42.3675	...	13:46	Ewallet	40.35	4.761905	2.0175	
1002	303-96-2227	B	Mandalay	Normal	Female	Home and lifestyle	97.38	10.0	48.6900	1022.4900	...	17:16	Ewallet	973.80	4.761905	48.6900	
1003	727-02-1313	A	Yangon	Member	Male	Food and beverages	31.84	1.0	1.5920	33.4320	...	13:22	Cash	31.84	4.761905	1.5920	
1004	347-56-2442	A	Yangon	Normal	Male	Home and lifestyle	65.82	1.0	3.2910	69.1110	...	15:33	Cash	65.82	4.761905	3.2910	
1005	849-09-3807	A	Yangon	Member	Female	Fashion accessories	88.34	7.0	30.9190	649.2990	...	13:28	Cash	618.38	4.761905	30.9190	

1000 rows × 21 columns

14. Change the datatype of the Date column.

```
[14]: df['Date']=df['Date'].str.replace('/','-')
```

```
[15]: df.head()
```

[15]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	cogs	gross margin percentage	gross income	Rat
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7.0	26.1415	548.9715	01-05-2019	13:08	Ewallet	522.83	4.761905	26.1415	
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5.0	3.8200	80.2200	03-08-2019	10:29	Cash	76.40	4.761905	3.8200	
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7.0	16.2155	340.5255	03-03-2019	13:23	Credit card	324.31	4.761905	16.2155	
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8.0	23.2880	489.0480	1-27-2019	20:33	Ewallet	465.76	4.761905	23.2880	
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7.0	30.2085	634.3785	02-08-2019	10:37	Ewallet	604.17	4.761905	30.2085	

```
[16]: df['Date']=pd.to_datetime(df['Date'])
```

```
[17]: df['day']=df['Date'].dt.day
```

```
[18]: df['month']=df['Date'].dt.month
```

```
[19]: df['year']=df['Date'].dt.year
```

```
[20]: df.head()
```

[20]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	...	Time	Payment	cogs	gross margin percentage	gross income	Rating
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7.0	26.1415	548.9715	...	13:08	Ewallet	522.83	4.761905	26.1415	9.
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5.0	3.8200	80.2200	...	10:29	Cash	76.40	4.761905	3.8200	9.8
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7.0	16.2155	340.5255	...	13:23	Credit card	324.31	4.761905	16.2155	7.4
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8.0	23.2880	489.0480	...	20:33	Ewallet	465.76	4.761905	23.2880	8.4
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7.0	30.2085	634.3785	...	10:37	Ewallet	604.17	4.761905	30.2085	5.3

5 rows × 21 columns

15. Deal with missing data and NA data.

[23]:
```
missing_values=df.isnull()
```

[24]:
```
print(missing_values)
```

```
      Invoice ID  Branch   City  Customer type  Gender  Product line  \
0          False   False  False          False   False         False
1          False   False  False          False   False         False
2          False   False  False          False   False         False
3          False   False  False          False   False         False
4          False   False  False          False   False         False
...          ...     ...    ...            ...     ...           ...
1001       False   False  False          False   False         False
1002       False   False  False          False   False         False
1003       False   False  False          False   False         False
1004       False   False  False          False   False         False
1005       False   False  False          False   False         False

      Unit price  Quantity  Tax 5%  Total  ...   Time  Payment   cogs  \
0          False     False   False  False  ...  False    False  False
1          False     False   False  False  ...  False    False  False
2          False     False   False  False  ...  False    False  False
3          False     False   False  False  ...  False    False  False
4          False     False   False  False  ...  False    False  False
...          ...       ...     ...    ...  ...    ...      ...    ...
1001       False     False   False  False  ...  False    False  False
1002       False     False   False  False  ...  False    False  False
1003       False     False   False  False  ...  False    False  False
1004       False     False   False  False  ...  False    False  False
1005       False     False   False  False  ...  False    False  False

      gross margin percentage  gross income  Rating  Product Price    day  \
0                       False         False   False          False  False
1                       False         False   False          False  False
2                       False         False   False          False  False
3                       False         False   False          False  False
4                       False         False   False          False  False
...                       ...           ...     ...            ...    ...
1001                    False         False   False          False  False
1002                    False         False   False          False  False
1003                    False         False   False          False  False
1004                    False         False   False          False  False
1005                    False         False   False          False  False

      month   year
0     False  False
1     False  False
2     False  False
3     False  False
4     False  False
...     ...    ...
1001  False  False
1002  False  False
1003  False  False
1004  False  False
1005  False  False

[1006 rows x 21 columns]
```

[91]:
```
df = df.fillna(0)
```

16. Saving the data as csv.

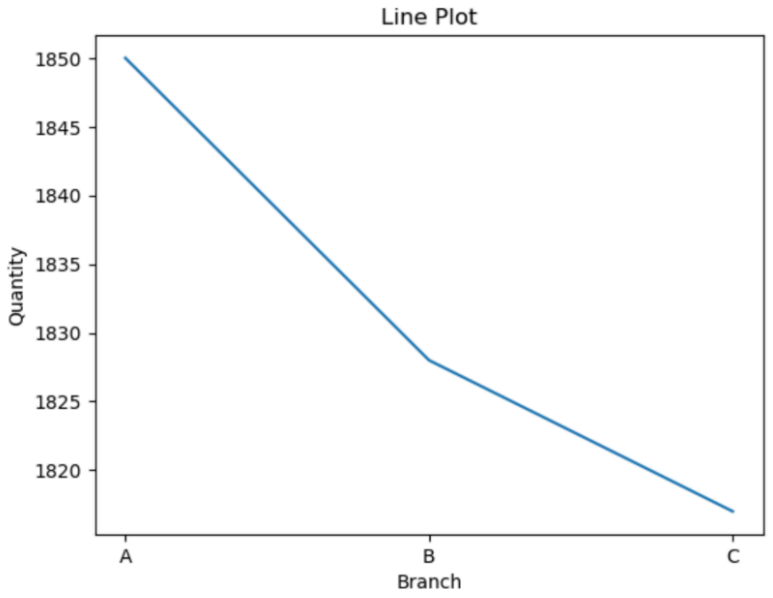
```
[97]: df.to_csv("Updated.csv")
```

DATA VISUALIZATION:

1. Draw a line chart for sum of quantity and branch using a line chart.

```
[44]: branch_quantity_sum=df.groupby('Branch')['Quantity'].sum()
      branch_quantity_name=df.groupby('Branch')
```

```
[46]: plt.plot(branch_quantity_sum)
      plt.title('Line Plot')
      plt.xlabel('Branch')
      plt.ylabel('Quantity')
      plt.show()
```

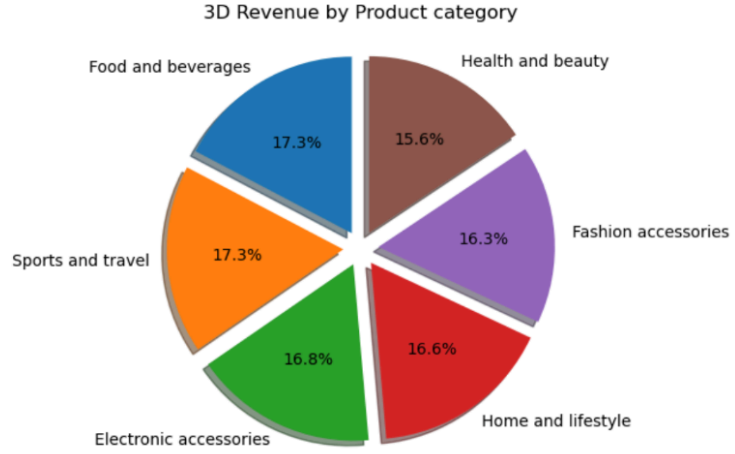


2. What are the top-performing product categories in terms of revenue? (Pie Chart)

[8]:
```
#Create a new column for revenue
df['Revenue']=df['Unit price']*df['Quantity']
#Group by product line then calculate sum then sort in desc
revenue_by_category=df.groupby('Product line')['Revenue'].sum().sort_values(ascending=False)
print(revenue_by_category)
```

```
Product line
Food and beverages        52705.25
Sports and travel         52634.17
Electronic accessories    51100.53
Home and lifestyle        50641.23
Fashion accessories       49763.28
Health and beauty         47608.54
Name: Revenue, dtype: float64
```

[9]:
```
#Create pie chart
plt.figure(figsize=(5,5)) #Adjusts size for all charts
explode=(0.1,0.1,0.1,0.1,0.1,0.1) #adds spaces between slices
plt.pie(revenue_by_category, labels=revenue_by_category.index, autopct='%1.1f%%', startangle=90, explode=explode, shadow=True)
plt.title('3D Revenue by Product category')
plt.show()
```

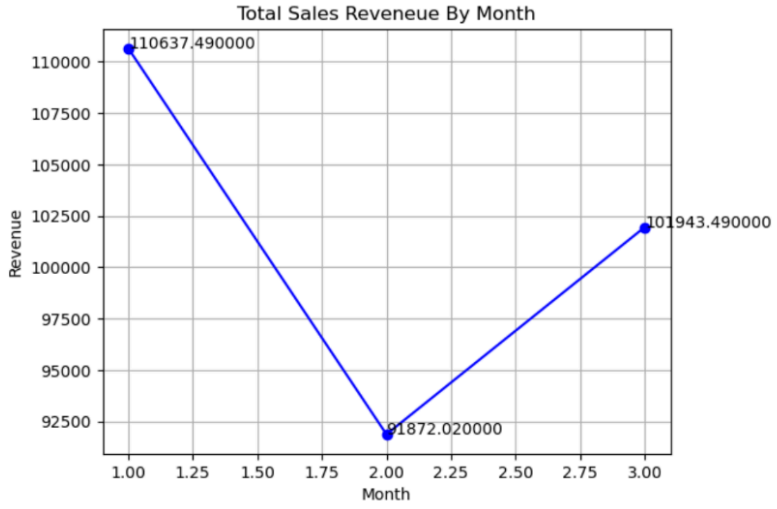


3. What is the total sales revenue by month? (line chart)

[10]:
```
#Create total sales revenue by month
df['Revenue']=df['Unit price']*df['Quantity']
revenue_by_month=df.groupby('month')['Revenue'].sum()
print(revenue_by_month)
```

```
month
1    110637.49
2     91872.02
3    101943.49
Name: Revenue, dtype: float64
```

[11]:
```
plt.plot(revenue_by_month.index, revenue_by_month.values, marker='o', color='b') #Marker highlights point ha=center, left, right va=
for i,values in enumerate(revenue_by_month.values):
    plt.text(revenue_by_month.index[i],values, f'{values:2f}')
plt.title('Total Sales Reveneue By Month')
plt.xlabel('Month')
plt.ylabel('Revenue')
plt.grid(True)
plt.show()
```

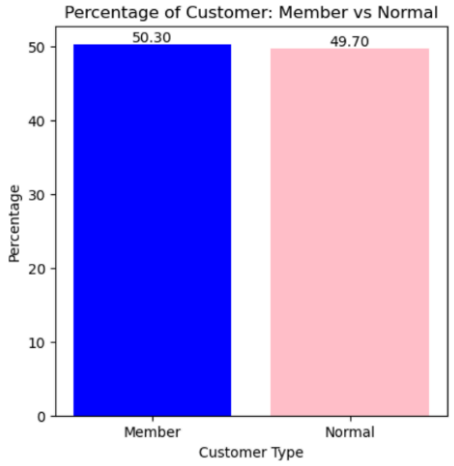


4. What percentage of customers are 'Members' vs 'Normal'? (Stacked bar chart)

[12]:
```
#Percent of customers are Members vs Normal
customer_count=df['Customer type'].value_counts() #calculates no of normal vs members separately
total_customer=customer_count.sum() #calculates all customers

#percentage
customer_percentage=(customer_count/total_customer)*100
print(customer_percentage)
```

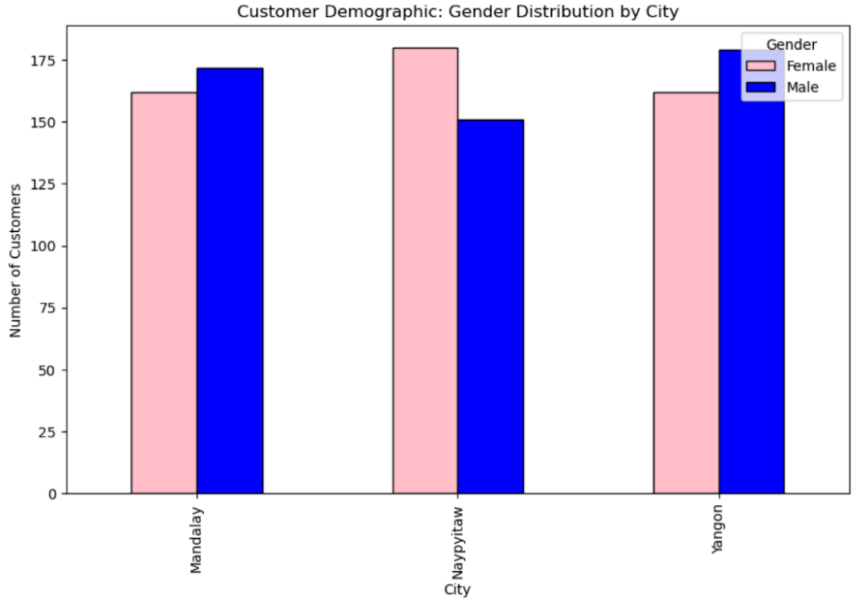
```
Customer type
Member    50.298211
Normal    49.701789
Name: count, dtype: float64
```

[58]:
```
plt.figure(figsize=(5,5))
plt.bar(customer_percentage.index, customer_percentage.values, color=['blue','pink'])

plt.title('Percentage of Customer: Member vs Normal')
plt.xlabel('Customer Type')
plt.ylabel('Percentage')

for i,values in enumerate(customer_percentage.values):
    plt.text(customer_percentage.index[i],values, f'{values:.2f}',ha='center', va='bottom')

plt.show()
```



5. What are the demographics (gender, location) of the customer base? (Stacked bar chart)

[62]:
```
#what are the customer demographics (gender,location) of the customer base?
gender_city_count=df.groupby(['City','Gender']).size().unstack()
gender_city_count.plot(kind='bar', figsize=(10,6), color=['pink','blue'],edgecolor='black')

plt.title('Customer Demographic: Gender Distribution by City')
plt.xlabel('City')
plt.ylabel('Number of Customers')
plt.legend(title='Gender')
plt.show()
```

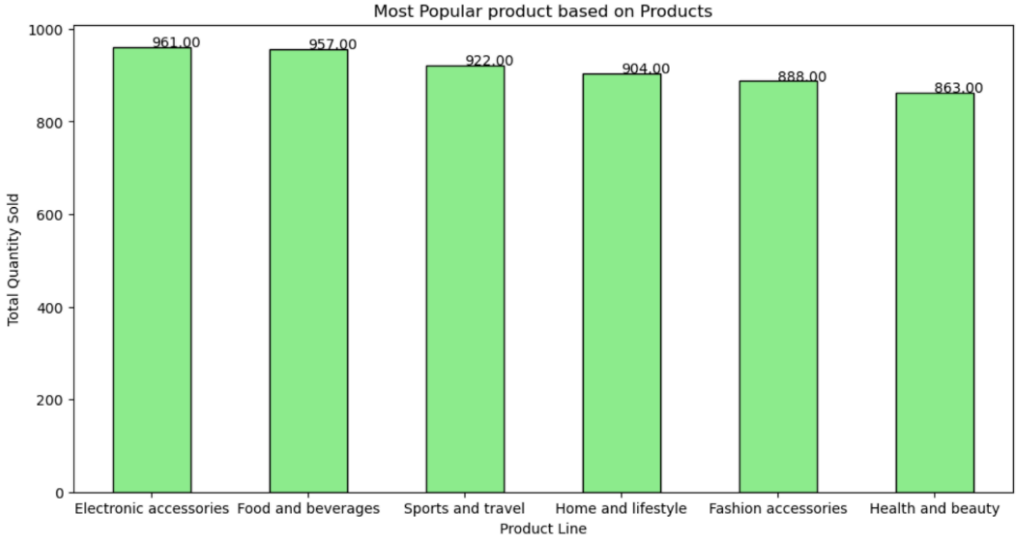


6. Which product lines are the most popular based on quantity sold? (Bar Chart)

[15]:
```
#which product lines are the most popular
product_line_quantity=df.groupby('Product line')['Quantity'].sum().sort_values(ascending=False)
print(product_line_quantity)
```

```
Product line
Electronic accessories    961.0
Food and beverages        957.0
Sports and travel         922.0
Home and lifestyle        904.0
Fashion accessories       888.0
Health and beauty         863.0
Name: Quantity, dtype: float64
```

[64]:
```
product_line_quantity.plot(kind='bar', figsize=(12,6), color='lightgreen',edgecolor='black')
for i,values in enumerate(product_line_quantity.values):
    plt.text(i,values, f'{values:.2f}')
plt.title('Most Popular product based on Products')
plt.xlabel('Product Line')
plt.ylabel('Total Quantity Sold')
plt.xticks(rotation=360, ha='center')
plt.show()
```

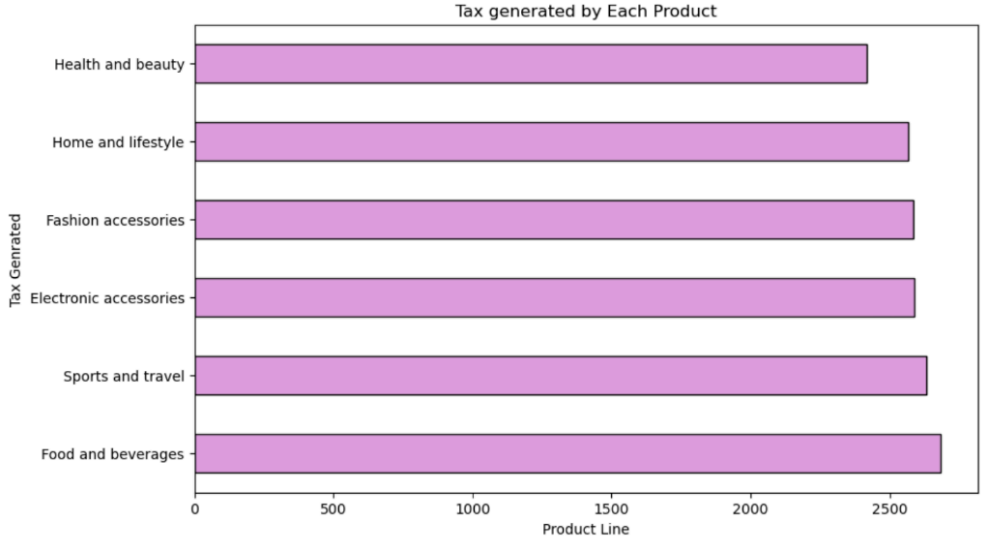


7. What is the tax amount generated by each product line? (horizontal bar chart)

```
[17]: #what is the total tax generated by each product
tax_product_line=df.groupby('Product line')['Tax 5%'].sum()
tax_product_line.sort_values(ascending=False, inplace=True)
print(tax_product_line)
```

```
Product line
Food and beverages        2681.2640
Sports and travel         2631.7085
Electronic accessories    2587.5015
Fashion accessories       2585.9950
Home and lifestyle        2564.8530
Health and beauty         2417.7055
Name: Tax 5%, dtype: float64
```

```
[18]: tax_product_line.plot(kind='barh', figsize=(10,6), color='plum',edgecolor='black')
plt.title('Tax generated by Each Product')
plt.xlabel('Product Line')
plt.ylabel('Tax Genrated')
plt.show()
```




```
[ ]:
```

8. What is the total sales revenue by payment method? (Pie Chart)

```
[110]: plt.figure(figsize=(6,6))
df.groupby('Payment')['Product Price'].sum().plot(kind='pie', autopct='%1.1f%%', startangle=90, colormap='Set2')
plt.title('Revenue by Payment Method')
plt.ylabel('')
plt.show()
```