

Assignment_2_filtering_sorting(Questions)

In [114...]:

```
import pandas as pd
```

1.1.1 Q1. Load the (SuperMarket Analysis.csv) dataset and display the first 5 rows.

In [115...]:

```
df = pd.read_csv('SuperMarketAnalysis.csv', sep=',')
df.head()
```

Out[115...]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Sales	Date	Time	Payment
0	750-67-8428	Alex	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	1/5/2019	1:08:00 PM	Ewallet
1	226-31-3081	Giza	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	3/8/2019	10:29:00 AM	Cash
2	631-41-3108	Alex	Yangon	Normal	Female	Home and lifestyle	46.33	7	16.2155	340.5255	3/3/2019	1:23:00 PM	Credit card
3	123-19-1176	Alex	Yangon	Member	Female	Health and beauty	58.22	8	23.2880	489.0480	1/27/2019	8:33:00 PM	Ewallet
4	373-73-7910	Alex	Yangon	Member	Female	Sports and travel	86.31	7	30.2085	634.3785	2/8/2019	10:37:00 AM	Ewallet

1.1.2 Q2. Display the dataset shape (rows and columns).

In [116...]:

```
df.shape
```

```
Out[116... (1000, 17)
```

1.1.3 Q3. List all column names.

```
In [117... df.columns
```

```
Out[117... Index(['Invoice ID', 'Branch', 'City', 'Customer type', 'Gender',
       'Product line', 'Unit price', 'Quantity', 'Tax 5%', 'Sales', 'Date',
       'Time', 'Payment', 'cogs', 'gross margin percentage', 'gross income',
       'Rating'],
      dtype='object')
```

1.1.4 Q4. Identify categorical and numerical columns.

```
In [118... categorical = df.select_dtypes(include='object').columns
numerical = df.select_dtypes(exclude='object').columns
print(categorical)
print(numerical)
```

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

1.1.5 Q5. Check for missing values in each column.

```
In [119... df.isnull().sum()
```

```
Out[119...]: Invoice ID      0  
Branch          0  
City            0  
Customer type   0  
Gender          0  
Product line    0  
Unit price     0  
Quantity        0  
Tax 5%          0  
Sales           0  
Date            0  
Time            0  
Payment         0  
cogs            0  
gross margin percentage 0  
gross income    0  
Rating          0  
dtype: int64
```

1.1.6 Q6. Display the data types of each column.

```
In [120...]: df.dtypes
```

```
Out[120... Invoice ID          object
Branch            object
City              object
Customer type    object
Gender            object
Product line     object
Unit price       float64
Quantity          int64
Tax 5%           float64
Sales             float64
Date              object
Time              object
Payment           object
cogs              float64
gross margin percentage float64
gross income     float64
Rating            float64
dtype: object
```

1.1.7 Q7. Show summary statistics for numerical columns (use pandas method).

```
In [121... numerical = df.select_dtypes(exclude='object').describe()
print(numerical)
```

	Unit price	Quantity	Tax 5%	Sales	cogs	\
count	1000.00000	1000.00000	1000.00000	1000.00000	1000.00000	
mean	55.672130	5.510000	15.379369	322.966749	307.58738	
std	26.494628	2.923431	11.708825	245.885335	234.17651	
min	10.080000	1.000000	0.508500	10.678500	10.17000	
25%	32.875000	3.000000	5.924875	124.422375	118.49750	
50%	55.230000	5.000000	12.088000	253.848000	241.76000	
75%	77.935000	8.000000	22.445250	471.350250	448.90500	
max	99.960000	10.000000	49.650000	1042.650000	993.00000	

	gross margin percentage	gross income	Rating
count	1000.00000	1000.00000	1000.00000
mean	4.761905	15.379369	6.97270
std	0.000000	11.708825	1.71858
min	4.761905	0.508500	4.00000
25%	4.761905	5.924875	5.50000
50%	4.761905	12.088000	7.00000
75%	4.761905	22.445250	8.50000
max	4.761905	49.650000	10.00000

1.1.8 Q8. Filter rows where Sales > 500.

```
In [122]: # df[df['Sales'] > 500]
df[df.Sales > 500]
```

Out[122...]

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Sales	Date	Time	Payme
0	750-67-8428	Alex	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	1/5/2019	1:08:00 PM	Ewal
4	373-73-7910	Alex	Yangon	Member	Female	Sports and travel	86.31	7	30.2085	634.3785	2/8/2019	10:37:00 AM	Ewal
5	699-14-3026	Giza	Naypyitaw	Member	Female	Electronic accessories	85.39	7	29.8865	627.6165	3/25/2019	6:30:00 PM	Ewal
7	315-22-5665	Giza	Naypyitaw	Member	Female	Home and lifestyle	73.56	10	36.7800	772.3800	2/24/2019	11:38:00 AM	Ewal
14	829-34-3910	Alex	Yangon	Member	Female	Health and beauty	71.38	10	35.6900	749.4900	3/29/2019	7:21:00 PM	Ca
...
988	267-62-7380	Giza	Naypyitaw	Member	Male	Electronic accessories	82.34	10	41.1700	864.5700	3/29/2019	7:12:00 PM	Ewal
989	430-53-4718	Cairo	Mandalay	Member	Male	Health and beauty	75.37	8	30.1480	633.1080	1/28/2019	3:46:00 PM	Cre ca
991	602-16-6955	Cairo	Mandalay	Normal	Female	Sports and travel	76.60	10	38.3000	804.3000	1/24/2019	6:10:00 PM	Ewal
996	303-96-2227	Cairo	Mandalay	Normal	Female	Home and lifestyle	97.38	10	48.6900	1022.4900	3/2/2019	5:16:00 PM	Ewal

Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Sales	Date	Time	Payme	
999	849-09-3807	Alex	Yangon	Member	Female	Fashion accessories	88.34	7	30.9190	649.2990	2/18/2019	1:28:00 PM	Ca

227 rows × 17 columns

1.1.9 Q9. Filter sales in City = "Yangon" and Sales > 200.

```
In [123...]: df[(df['Sales']>200) & (df['City'] == 'Yangon')]
```

Out[123...]

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Sales	Date	Time	Payment
0	750-67-8428	Alex	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	1/5/2019	1:08:00 PM	Ewallet
2	631-41-3108	Alex	Yangon	Normal	Female	Home and lifestyle	46.33	7	16.2155	340.5255	3/3/2019	1:23:00 PM	Credit card
3	123-19-1176	Alex	Yangon	Member	Female	Health and beauty	58.22	8	23.2880	489.0480	1/27/2019	8:33:00 PM	Ewallet
4	373-73-7910	Alex	Yangon	Member	Female	Sports and travel	86.31	7	30.2085	634.3785	2/8/2019	10:37:00 AM	Ewallet
6	355-53-5943	Alex	Yangon	Member	Female	Electronic accessories	68.84	6	20.6520	433.6920	2/25/2019	2:36:00 PM	Ewallet
...
976	221-25-5073	Alex	Yangon	Normal	Female	Food and beverages	74.66	4	14.9320	313.5720	3/4/2019	10:39:00 AM	Cash
981	809-46-1866	Alex	Yangon	Normal	Male	Health and beauty	58.15	4	11.6300	244.2300	1/23/2019	5:44:00 PM	Cash
982	139-32-4183	Alex	Yangon	Member	Female	Sports and travel	97.48	9	43.8660	921.1860	3/14/2019	2:19:00 PM	Ewallet
990	886-18-2897	Alex	Yangon	Normal	Female	Food and beverages	56.56	5	14.1400	296.9400	3/22/2019	7:06:00 PM	Credit card

Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Sales	Date	Time	Payment	
999	849-09-3807	Alex	Yangon	Member	Female	Fashion accessories	88.34	7	30.9190	649.2990	2/18/2019	1:28:00 PM	Cash

204 rows × 17 columns

1.1.10 Q10. Sort all orders by Sales in descending order.

```
In [124...]: df.sort_values(by='Sales', ascending=False)
```

Out[124...]

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Sales	Date	Time	Payme	
	350	860-79-0874	Giza	Naypyitaw	Member	Female	Fashion accessories	99.30	10	49.6500	1042.6500	2/15/2019	2:53:00 PM	Credit ca
	167	687-47-8271	Alex	Yangon	Normal	Male	Fashion accessories	98.98	10	49.4900	1039.2900	2/8/2019	4:20:00 PM	Credit ca
	557	283-26-5248	Giza	Naypyitaw	Member	Female	Food and beverages	98.52	10	49.2600	1034.4600	1/30/2019	8:23:00 PM	Ewal
	699	751-41-9720	Giza	Naypyitaw	Normal	Male	Home and lifestyle	97.50	10	48.7500	1023.7500	1/12/2019	4:18:00 PM	Ewal
	996	303-96-2227	Cairo	Mandalay	Normal	Female	Home and lifestyle	97.38	10	48.6900	1022.4900	3/2/2019	5:16:00 PM	Ewal
	
	402	236-86-3015	Giza	Naypyitaw	Member	Male	Home and lifestyle	13.98	1	0.6990	14.6790	2/4/2019	1:38:00 PM	Ewal
	443	192-98-7397	Giza	Naypyitaw	Normal	Male	Fashion accessories	12.78	1	0.6390	13.4190	1/8/2019	2:11:00 PM	Ewal
	223	279-62-1445	Giza	Naypyitaw	Member	Female	Fashion accessories	12.54	1	0.6270	13.1670	2/21/2019	12:38:00 PM	Ca
	629	308-39-1707	Alex	Yangon	Normal	Female	Fashion accessories	12.09	1	0.6045	12.6945	1/26/2019	6:19:00 PM	Credit ca

Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Sales	Date	Time	Payme	
822	784-21-9238	Giza	Naypyitaw	Member	Male	Sports and travel	10.17	1	0.5085	10.6785	2/7/2019	2:15:00 PM	Ca

1000 rows × 17 columns

1.1.11 Q11. Sort by Date (ascending) and then Time (ascending).

```
In [125...]: df.sort_values(by=['Date', 'Time'])
```

Out[125...]

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Sales	Date	Time	Payment
17	765-26-6951	Alex	Yangon	Member	Female	Sports and travel	72.61	6	21.783	457.443	1/1/2019	10:39:00 AM	Credit card
970	746-04-1077	Cairo	Mandalay	Member	Female	Food and beverages	84.63	10	42.315	888.615	1/1/2019	11:36:00 AM	Credit card
839	271-77-8740	Giza	Naypyitaw	Member	Female	Sports and travel	29.22	6	8.766	184.086	1/1/2019	11:40:00 AM	Ewallet
523	133-14-7229	Giza	Naypyitaw	Normal	Male	Health and beauty	62.87	2	6.287	132.027	1/1/2019	11:43:00 AM	Cash
567	651-88-7328	Alex	Yangon	Normal	Female	Fashion accessories	65.74	9	29.583	621.243	1/1/2019	1:55:00 PM	Cash
...
122	219-22-9386	Cairo	Mandalay	Member	Female	Sports and travel	99.96	9	44.982	944.622	3/9/2019	5:26:00 PM	Credit card
45	132-32-9879	Cairo	Mandalay	Member	Female	Electronic accessories	93.96	4	18.792	394.632	3/9/2019	6:00:00 PM	Cash
73	841-35-6630	Giza	Naypyitaw	Member	Female	Electronic accessories	75.91	6	22.773	478.233	3/9/2019	6:21:00 PM	Cash
234	157-13-5295	Alex	Yangon	Member	Male	Health and beauty	51.94	10	25.970	545.370	3/9/2019	6:24:00 PM	Ewallet

Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Sales	Date	Time	Payment	
326	815-11-1168	Alex	Yangon	Member	Male	Food and beverages	99.78	5	24.945	523.845	3/9/2019	7:09:00 PM	Cash

1000 rows × 17 columns

1.1.12 Q12. Sort by Unit price and Quantity.

```
In [126...]: df.sort_values(by=['Unit price', 'Quantity'])
```

Out[126...]

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Sales	Date	Time	Payer
944	333-23-2632	Alex	Yangon	Member	Male	Health and beauty	10.08	7	3.5280	74.0880	3/28/2019	8:14:00 PM	Cash
572	239-48-4278	Alex	Yangon	Member	Male	Food and beverages	10.13	7	3.5455	74.4555	3/10/2019	7:35:00 PM	Ewallet
784	516-77-6464	Giza	Naypyitaw	Member	Female	Health and beauty	10.16	5	2.5400	53.3400	2/24/2019	1:08:00 PM	Ewallet
822	784-21-9238	Giza	Naypyitaw	Member	Male	Sports and travel	10.17	1	0.5085	10.6785	2/7/2019	2:15:00 PM	Cash
881	115-38-7388	Giza	Naypyitaw	Member	Female	Fashion accessories	10.18	8	4.0720	85.5120	3/30/2019	12:51:00 PM	Credit card
...
283	667-92-0055	Alex	Yangon	Member	Male	Health and beauty	99.83	6	29.9490	628.9290	3/4/2019	3:02:00 PM	Ewallet
494	437-53-3084	Cairo	Mandalay	Normal	Male	Fashion accessories	99.89	2	9.9890	209.7690	2/26/2019	11:48:00 AM	Ewallet
930	641-62-7288	Cairo	Mandalay	Normal	Male	Home and lifestyle	99.92	6	29.9760	629.4960	3/24/2019	1:33:00 PM	Ewallet
983	148-41-7930	Giza	Naypyitaw	Normal	Male	Health and beauty	99.96	7	34.9860	734.7060	1/23/2019	10:33:00 AM	Cash

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Sales	Date	Time	Payer
122	219-22-9386	Cairo	Mandalay	Member	Female	Sports and travel	99.96	9	44.9820	944.6220	3/9/2019	5:26:00 PM	Crec cai

1000 rows × 17 columns

1.1.13 Q13. Calculate the total sales in "Sales" per Branch.

```
In [127...]: df.groupby('Branch')['Sales'].sum()
```

```
Out[127...]: Branch
Alex      106200.3705
Cairo    106197.6720
Giza     110568.7065
Name: Sales, dtype: float64
```

1.1.14 Q14. Calculate average Sales per City.

```
In [128...]: df.groupby('City')['Sales'].mean()
```

```
Out[128...]: City
Mandalay    319.872506
Naypyitaw   337.099715
Yangon      312.354031
Name: Sales, dtype: float64
```

1.1.15 Q15. Find the quantities sold per product line.

```
In [129...]: df.groupby('Product line')['Quantity'].sum()
```

```
Out[129... Product line
Electronic accessories    971
Fashion accessories       902
Food and beverages        952
Health and beauty         854
Home and lifestyle        911
Sports and travel         920
Name: Quantity, dtype: int64
```

1.1.16 Q16. Calculate average gross income per Gender.

```
In [130... df.groupby('Gender')['gross income'].mean()
```

```
Out[130... Gender
Female      16.234829
Male        14.240749
Name: gross income, dtype: float64
```

1.1.17 Q17. Count number of sales per Payment method.

```
In [131... df.groupby('Payment')['Sales'].count()
```

```
Out[131... Payment
Cash          344
Credit card   311
Ewallet        345
Name: Sales, dtype: int64
```

1.1.18 Q18. Find maximum Sales per Branch.

```
In [132... df.groupby('Branch')['Sales'].max()
```

```
Out[132... Branch
Alex        1039.29
Cairo       1022.49
Giza        1042.65
Name: Sales, dtype: float64
```

1.1.19 Q19. Find minimum Unit price per Product line.

```
In [133]: df.groupby('Product line')['Unit price'].min()
```

```
Out[133]: Product line
Electronic accessories    10.56
Fashion accessories      10.18
Food and beverages        10.13
Health and beauty         10.08
Home and lifestyle        10.53
Sports and travel         10.17
Name: Unit price, dtype: float64
```

1.1.20 Q20. Find the sum of gross income per Product line and Branch.

```
In [134]: df.groupby(['Product line', 'Branch'])['gross income'].sum()
```

```
Out[134]: Product line      Branch
Electronic accessories   Alex     872.2435
                           Cairo    811.9735
                           Giza    903.2845
Fashion accessories      Alex     777.7385
                           Cairo    781.5865
                           Giza    1026.6700
Food and beverages       Alex     817.2905
                           Cairo    724.5185
                           Giza    1131.7550
Health and beauty        Alex     599.8930
                           Cairo    951.4600
                           Giza    791.2060
Home and lifestyle       Alex     1067.4855
                           Cairo    835.6745
                           Giza    661.6930
Sports and travel        Alex     922.5095
                           Cairo    951.8190
                           Giza    750.5680
Name: gross income, dtype: float64
```

1.1.21 21. What is the total quantities sold in Product line: "Electronic accessories"?

```
In [135...]: condition = (df['Product line'] == "Electronic accessories")
print(df[condition]['Quantity'].sum())
```

971

1.1.22 22. What is the average Sales for female customers?

```
In [136...]: condition = df['Gender'] == 'female'
df[condition]['Sales'].mean()
```

Out[136...]: nan

1.1.23 23. What is the most expensive Unit price among Customer type members only?

```
In [137...]: # df.columns
df.groupby('Customer type')['Unit price'].max()
```

```
Out[137...]: Customer type
Member      99.96
Normal      99.96
Name: Unit price, dtype: float64
```

1.1.24 24. How many orders with Rating >= 9 ?

```
In [ ]: condition = df['Rating'] >= 9
# df[condition].shape[0]
print(condition.sum())
```

166

1.1.25 25. What is the total Sales for Payment "Credit card" in Branch C?

```
In [139...]: s = df['Branch'].str.startswith('C');
cond = (df['Payment'] == "Credit card") & (s == True)
print(df[cond]['Sales'].sum())
```

37344.8565