

190031920

A Nikhil Reddy

DS Skill 2

In [1]:
import numpy as np
import pandas as pd

In [2]:
marketsalesdata = pd.read_csv('./supermarket_sales_-_Sheet1.csv')

In [3]:
marketsalesdata

Out[3]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	1/5/2019	13:08	Ewallet
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	3/8/2019	10:29	Credit card
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	3/3/2019	13:23	Credit card
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.0480	1/27/2019	20:33	Ewallet
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2/8/2019	10:37	Ewallet
...
995	233-67-5758	C	Naypyitaw	Normal	Male	Health and beauty	40.35	1	2.0175	42.3675	1/29/2019	13:46	Credit card
996	303-96-2227	B	Mandalay	Normal	Female	Home and lifestyle	97.38	10	48.6900	1022.4900	3/2/2019	17:16	Ewallet
997	727-02-1313	A	Yangon	Member	Male	Food and beverages	31.84	1	1.5920	33.4320	2/9/2019	13:22	Credit card
998	347-56-2442	A	Yangon	Normal	Male	Home and lifestyle	65.82	1	3.2910	69.1110	2/22/2019	15:33	Credit card
999	849-09-3807	A	Yangon	Member	Female	Fashion accessories	88.34	7	30.9190	649.2990	2/18/2019	13:28	Credit card

1000 rows × 17 columns

In [4]:
marketsalesdata.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 17 columns):
Column Non-Null Count Dtype
--- -
0 Invoice ID 1000 non-null object
1 Branch 1000 non-null object
2 City 1000 non-null object
3 Customer type 1000 non-null object
4 Gender 1000 non-null object
5 Product line 1000 non-null object
6 Unit price 1000 non-null float64
7 Quantity 1000 non-null int64
8 Tax 5% 1000 non-null float64
9 Total 1000 non-null float64
10 Date 1000 non-null object
11 Time 1000 non-null object
12 Payment 1000 non-null object
13 cogs 1000 non-null float64
14 gross margin percentage 1000 non-null float64
15 gross income 1000 non-null float64
16 Rating 1000 non-null float64
dtypes: float64(7), int64(1), object(9)
memory usage: 132.9+ KB

In [5]:
marketsalesdata.describe()

Out[5]:

	Unit price	Quantity	Tax 5%	Total	cogs	gross margin percentage	gross income	Rating
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	55.672130	5.510000	15.379369	322.966749	307.58738	4.761905	15.379369	6.97270
std	26.494628	2.923431	11.708825	245.885335	234.17651	0.000000	11.708825	1.71858
min	10.080000	1.000000	0.508500	10.678500	10.17000	4.761905	0.508500	4.00000
25%	32.875000	3.000000	5.924875	124.422375	118.49750	4.761905	5.924875	5.50000
50%	55.230000	5.000000	12.088000	253.848000	241.76000	4.761905	12.088000	7.00000
75%	77.935000	8.000000	22.445250	471.350250	448.90500	4.761905	22.445250	8.50000
max	99.960000	10.000000	49.650000	1042.650000	993.00000	4.761905	49.650000	10.00000

In [6]:
from scipy import stats
mode = stats.mode(marketsalesdata['Product line'])
mode

Out[6]:
ModeResult(mode=array(['Fashion accessories'], dtype=object), count=array([178]))

In [7]:
g1 = marketsalesdata.groupby('Product line').agg([np.mean])
g1

Out[7]:

	Unit price	Quantity	Tax 5%	Total	cogs	gross margin percentage	gross income	Rating
	mean	mean	mean	mean	mean	mean	mean	mean
Product line								
Electronic accessories	53.551588	5.711765	15.220597	319.632538	304.411941	4.761905	15.220597	6.924706
Fashion accessories	57.153652	5.067416	14.528062	305.089298	290.561236	4.761905	14.528062	7.029213
Food and beverages	56.008851	5.471264	15.365310	322.671517	307.306207	4.761905	15.365310	7.113218
Health and beauty	54.854474	5.618421	15.411572	323.643020	308.231447	4.761905	15.411572	7.003289
Home and lifestyle	55.316937	5.693750	16.030331	336.636956	320.606625	4.761905	16.030331	6.837500
Sports and travel	56.993253	5.542169	15.812630	332.065220	316.252590	4.761905	15.812630	6.916265

In [8]:
sd = marketsalesdata['Unit price'].std()
sd

Out[8]:
26.49462834791978

In [9]:
marketsalesdata['Gender'].value_counts()
marketsalesdata['Customer type'].value_counts()
marketsalesdata[marketsalesdata['Gender']=='Male'].value_counts()
marketsalesdata[marketsalesdata['Gender']=='Female'].value_counts()

Out[9]:

Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	cogs	gross margin percentage	gross income	Rating
101-81-4070	C	Naypyitaw	Member	Female	Health and beauty	62.82	2	6.2820	131.9220	125.64	4.761905	6.2820	4.9
632-90-0281	C	Naypyitaw	Normal	Female	Fashion accessories	37.55	10	18.7750	394.2750	375.50	4.761905	18.7750	9.3
650-98-6268	B	Mandalay	Member	Female	Food and beverages	20.87	3	3.1305	65.7405	62.61	4.761905	3.1305	8.0
649-11-3678	C	Naypyitaw	Normal	Female	Food and beverages	22.93	9	10.3185	216.6885	206.37	4.761905	10.3185	5.5
647-50-1224	A	Yangon	Normal	Female	Fashion accessories	29.42	10	14.7100	308.9100	294.20	4.761905	14.7100	8.9
346-84-3103	B	Mandalay	Member	Female	Electronic accessories	13.22	5	3.3050	69.4050	66.10	4.761905	3.3050	4.3
345-68-9016	C	Naypyitaw	Member	Female	Sports and travel	31.67	8	12.6680	266.0280	253.36	4.761905	12.6680	5.6
343-75-9322	B	Mandalay	Member	Female	Sports and travel	11.85	8	4.7400	99.5400	94.80	4.761905	4.7400	4.1
340-21-9136	A	Yangon	Member	Female	Sports and travel	40.05	4	8.0100	168.2100	160.20	4.761905	8.0100	9.7
895-03-6665	B	Mandalay	Normal	Female	Fashion accessories	36.51	9	16.4295	345.0195	328.59	4.761905	16.4295	4.2
Length: 501, dtype: int64													

In [10]:
marketsalesdata['Payment'].value_counts()

Out[10]:
Ewallet 345
Cash 344
Credit card 311
Name: Payment, dtype: int64

In [11]:
yangon = marketsalesdata.loc[marketsalesdata['City']=='Yangon']
yangon

Out[11]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	1/5/2019	13:08	Ewallet
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	3/3/2019	13:23	Credit card
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.0480	1/27/2019	20:33	Ewallet
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2/8/2019	10:37	Ewallet
6	355-53-5943	A	Yangon	Member	Female	Electronic accessories	68.84	6	20.6520	433.6920	2/25/2019	14:36	Ewallet
...
990	886-18-2897	A	Yangon	Normal	Female	Food and beverages	56.56	5	14.1400	296.9400	3/22/2019	19:06	Credit card
992	745-74-0715	A	Yangon	Normal	Male	Electronic accessories	58.03	2	5.8030	121.8630	3/10/2019	20:46	Ewallet
997	727-02-1313	A	Yangon	Member	Male	Food and beverages	31.84	1	1.5920	33.4320	2/9/2019	13:22	Credit card
998	347-56-2442	A	Yangon	Normal	Male	Home and lifestyle	65.82	1	3.2910	69.1110	2/22/2019	15:33	Credit card
999	849-09-3807	A	Yangon	Member	Female	Fashion accessories	88.34	7	30.9190	649.2990	2/18/2019	13:28	Credit card

340 rows × 17 columns

In [12]:
marketsalesdata['Gender'].unique()
female_fashion= pd.crosstab(marketsalesdata['Gender'],marketsalesdata['Product line']=='Fashion accessories')
female_fashion

Out[12]:

Product line	False	True
Gender		
Female	405	96
Male	417	82

In [14]:
marketsalesdata['Gender'].unique()
male_health= pd.crosstab(marketsalesdata['Gender'],marketsalesdata['Product line']=='Health and Beauty')
male_health

Out[14]:

Product line	False
Gender	
Female	501
Male	499