NAME: NERELLA VENKATA RADHAKRISHNA

ID: 190031187

SKILL-2

In [2]:
```
import pandas as pd
import numpy as np
from scipy import stats
```

In [3]:
```
df = pd.read_csv('supermarket_sales.csv')
df
```

Out[3]:

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

1000 rows × 17 columns

In [4]:
```
#1
#Information
df.describe()
```

Out[4]:

	Unit price	Quantity	Tax 5%	Total	cogs	gross margin percentage	gross income	Rating
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.00000	1000.000000	1000.000000	1000.00000
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 [5]:
```
#1
#Information
df.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 [15]:
```
#2. Mode of Product line
# a = df['Product line'] or
# df[start:end:step,start:end:step]
# a=df.iloc[:,5]
a.mode() # or stats.mode(a)
```

Out[15]:
```
0    Fashion accessories
dtype: object
```

In [7]:
```
#3
#Mean of product Line
mn = df.groupby('Product line').mean()
mn
```

Out[7]:

Product line	Unit price	Quantity	Tax 5%	Total	cogs	gross margin percentage	gross income	Rating
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]:
```
#4
#Standard deviation of Unit Price
b = df['Unit price']
b.std()
```

Out[8]:
```
26.49462834791978
```

In [9]:
```
#5
#Total Customers
tot_cust = df["Invoice ID"]
print("Total Customers:",tot_cust.shape[0])
# df["Invoice ID"].value_counts()
# df["Gender"].value_counts()

#Total Females
tot_fem = df[df["Gender"]=='Female']
print("Total Female Customers: ",tot_fem.shape[0])

#Total Males
tot_mal = df[df["Gender"]=='Male']
print("Total Male Customers: ",tot_mal.shape[0])
```

```
Total Customers: 1000
Total Female Customers:  501
Total Male Customers:  499
```

In [10]:
```
#6
#People paying through E-wallet
tot_wal = df[df["Payment"] == "Ewallet"]
print("Max customers paying through E-Wallet: ",tot_wal.shape[0])
```

```
Max customers paying through E-Wallet:  345
```

In [11]:
```
#7
#People coming from Yangon city
yangon = df[df["City"]=="Yangon"]
print("Max people coming from Yangon city are : ",yangon.shape[0])
```

```
Max people coming from Yangon city are :  340
```

In [12]:
```
#8
#Total average sales of Fashion accesories by females
fem_sales = df[(df["Gender"] == "Female") & (df["Product line"]=="Fashion accessories")]
print("Average Sales of Fashion Accesories by Female: ",fem_sales["Total"].mean())
```

```
Average Sales of Fashion Accesories by Female:  317.05625
```

In [13]:
```
#9
#Total average sales of Health & Beauty by males
male_sales = df[(df["Gender"] == "Male") & (df["Product line"]=="Health and beauty")]
print("Average Sales of Health and beauty by Male: ",male_sales["Total"].mean())
```

```
Average Sales of Health and beauty by Male:  348.0994602272727
```