Python Training Session

July 15, 2025

[ ]:
```
!pip install pandas
import pandas as pd
!pip install matplotlib
import matplotlib.pyplot as plt
!pip install seaborn
import seaborn as sns
!pip install numpy
import numpy as np
!pip install openpyxl
!pip install nbconvert
```

[ ]:
```
!pip install openpyxl
```

[55]:
```
import pandas as pd
df = pd.read_excel(r"C:\Users\Home\Documents\excel classes\Supermarket Sales␣
↪Cleaned.xlsx")
```

[56]:
```
df.to_csv("Supermarket Sales Cleaned.csv",index=False)
```

[57]:
```
df.head()
```

[57]:

	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	26.1415	548.9715	2019-01-05	0.547222	Ewallet	522.83	4.761905	26.1415
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	2019-03-08	0.436806	Cash	76.40	4.761905	3.8200
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	2019-03-03					
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.0480	2019-01-27					
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2019-02-08					

```
2  0.557639  Credit card  324.31                4.761905       16.2155
3  0.856250      Ewallet  465.76                4.761905       23.2880
4  0.442361      Ewallet  604.17                4.761905       30.2085

   Rating  Day  Month  Year
0     9.1    5      1  2019
1     9.6    8      3  2019
2     7.4    3      3  2019
3     8.4   27      1  2019
4     5.3    8      2  2019
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 20 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Invoice ID               500 non-null    object
 1   Branch                   500 non-null    object
 2   City                     500 non-null    object
 3   Customer type            500 non-null    object
 4   Gender                   500 non-null    object
 5   Product line             500 non-null    object
 6   Unit price               500 non-null    float64
 7   Quantity                 500 non-null    int64
 8   Tax 5%                   500 non-null    float64
 9   Total                    500 non-null    float64
 10  Date                     500 non-null    datetime64[ns]
 11  Time                     500 non-null    float64
 12  Payment                  500 non-null    object
 13  cogs                     500 non-null    float64
 14  gross margin percentage  500 non-null    float64
 15  gross income             500 non-null    float64
 16  Rating                   500 non-null    float64
 17  Day                      500 non-null    int64
 18  Month                    500 non-null    int64
 19  Year                     500 non-null    int64
dtypes: datetime64[ns](1), float64(8), int64(4), object(7)
memory usage: 78.3+ KB
```

[59]:
```
df.describe()
```

[59]:
```
       Unit price    Quantity      Tax 5%        Total  \
count  500.000000  500.000000  500.000000   500.000000
mean    54.850140    5.692000   15.714081   329.995701
min     10.590000    1.000000    0.627000    13.167000
25%     30.560000    3.000000    6.431750   135.066750
```

```
50%     52.425000    6.000000   12.892500   270.742500
75%     77.772500    8.000000   22.847875   479.805375
max     99.960000   10.000000   49.650000  1042.650000
std     26.848516    2.899301   11.709972   245.909409

                           Date        Time        cogs  \
count                       500  500.000000  500.000000
mean   2019-02-15 06:54:43.200000    0.646029  314.281620
min          2019-01-01 00:00:00    0.416667   12.540000
25%          2019-01-25 18:00:00    0.534375  128.635000
50%          2019-02-15 00:00:00    0.646875  257.850000
75%          2019-03-09 00:00:00    0.765451  456.957500
max          2019-03-30 00:00:00    0.874306  993.000000
std                          NaN    0.133405  234.199437

       gross margin percentage  gross income      Rating        Day  \
count               500.000000    500.000000  500.000000  500.00000
mean                  4.761905     15.714081    7.018600   14.70400
min                   4.761905      0.627000    4.000000    1.00000
25%                   4.761905      6.431750    5.600000    7.00000
50%                   4.761905     12.892500    7.000000   14.00000
75%                   4.761905     22.847875    8.500000   23.00000
max                   4.761905     49.650000   10.000000   31.00000
std                   0.000000     11.709972    1.719018    8.69918

            Month    Year
count  500.000000   500.0
mean     2.056000  2019.0
min      1.000000  2019.0
25%      1.000000  2019.0
50%      2.000000  2019.0
75%      3.000000  2019.0
max      3.000000  2019.0
std      0.842784     0.0
```

[60]:
```
df[df['Unit price'] > 50]
```

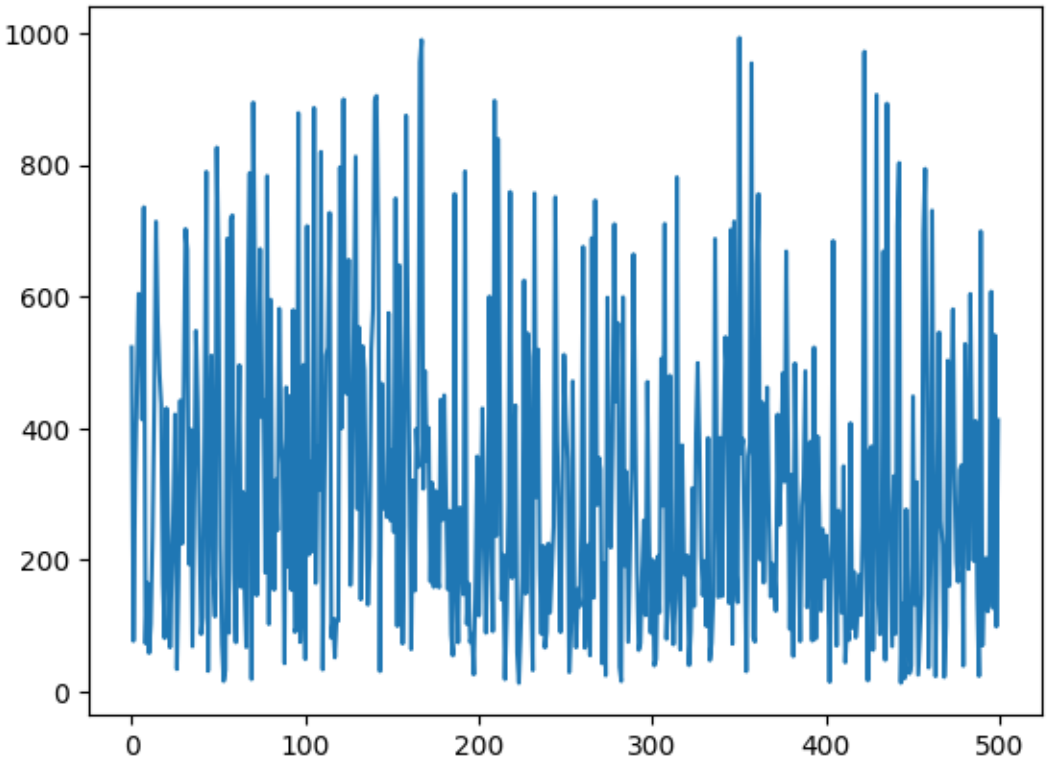
[60]:
```
      Invoice ID Branch       City Customer type  Gender  \
0    750-67-8428      A     Yangon        Member  Female
3    123-19-1176      A     Yangon        Member    Male
4    373-73-7910      A     Yangon        Normal    Male
5    699-14-3026      C  Naypyitaw        Normal    Male
6    355-53-5943      A     Yangon        Member  Female
..           …    …        …             …     …
495  632-32-4574      B   Mandalay        Normal    Male
496  556-97-7101      C  Naypyitaw        Normal  Female
497  862-59-8517      C  Naypyitaw        Normal  Female
```

```
498  401-18-8016      B   Mandalay        Member  Female
499  420-18-8989      A     Yangon        Member  Female

               Product line  Unit price  Quantity   Tax 5%     Total  \
0         Health and beauty       74.69         7  26.1415  548.9715
3         Health and beauty       58.22         8  23.2880  489.0480
4         Sports and travel       86.31         7  30.2085  634.3785
5    Electronic accessories       85.39         7  29.8865  627.6165
6    Electronic accessories       68.84         6  20.6520  433.6920
..                      …           …       …       …        …
495       Sports and travel       75.92         8  30.3680  637.7280
496  Electronic accessories       63.22         2   6.3220  132.7620
497      Food and beverages       90.24         6  27.0720  568.5120
498       Sports and travel       98.13         1   4.9065  103.0365
499       Sports and travel       51.52         8  20.6080  432.7680

           Date      Time  Payment    cogs  gross margin percentage  \
0    2019-01-05  0.547222  Ewallet  522.83                 4.761905
3    2019-01-27  0.856250  Ewallet  465.76                 4.761905
4    2019-02-08  0.442361  Ewallet  604.17                 4.761905
5    2019-03-25  0.770833  Ewallet  597.73                 4.761905
6    2019-02-25  0.608333  Ewallet  413.04                 4.761905
..          …       …       …     …                      …
495  2019-03-20  0.593056     Cash  607.36                 4.761905
496  2019-01-01  0.660417     Cash  126.44                 4.761905
497  2019-01-27  0.470139     Cash  541.44                 4.761905
498  2019-01-21  0.733333     Cash   98.13                 4.761905
499  2019-02-02  0.657639     Cash  412.16                 4.761905

     gross income  Rating  Day  Month  Year
0         26.1415     9.1    5      1  2019
3         23.2880     8.4   27      1  2019
4         30.2085     5.3    8      2  2019
5         29.8865     4.1   25      3  2019
6         20.6520     5.8   25      2  2019
..            …     …  …    …   …
495       30.3680     5.5   20      3  2019
496        6.3220     8.5    1      1  2019
497       27.0720     6.2   27      1  2019
498        4.9065     8.9   21      1  2019
499       20.6080     9.6    2      2  2019

[264 rows x 20 columns]
```

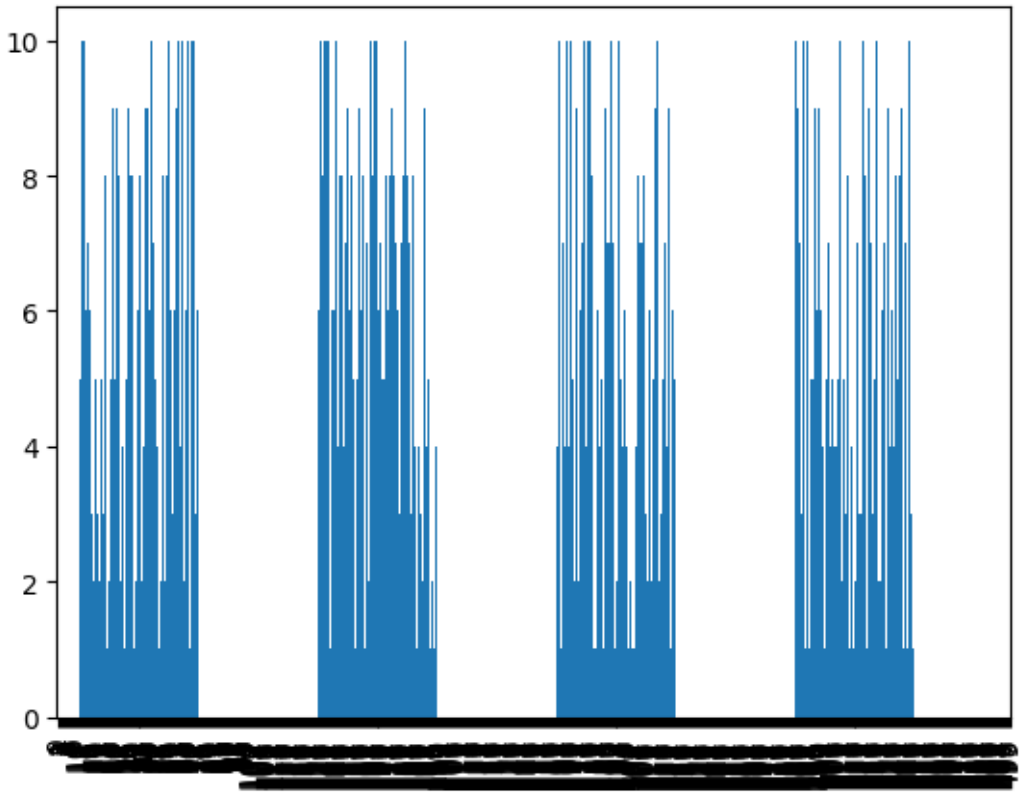
[61]:
```
print(df['Year'].dtype)
```

```
int64
```

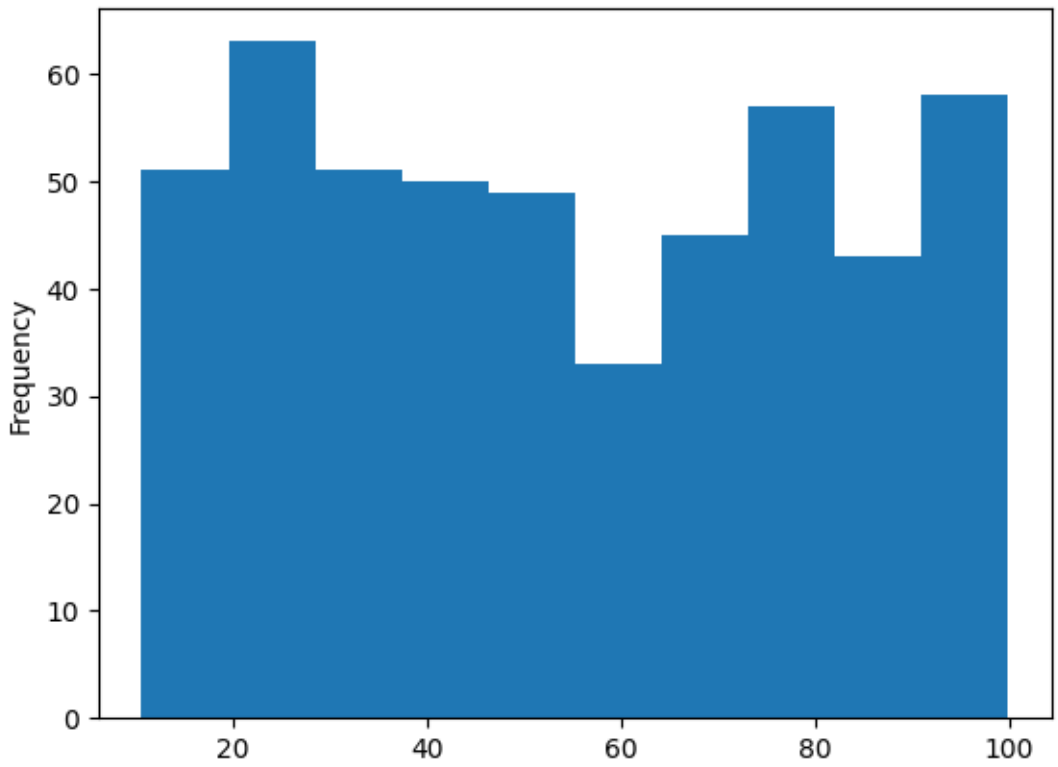
[63]:
```
Lineplot:df['cogs'].plot(kind ='line')
```



[64]:
```
Barchart:df['Quantity'].plot(kind = 'bar')
```



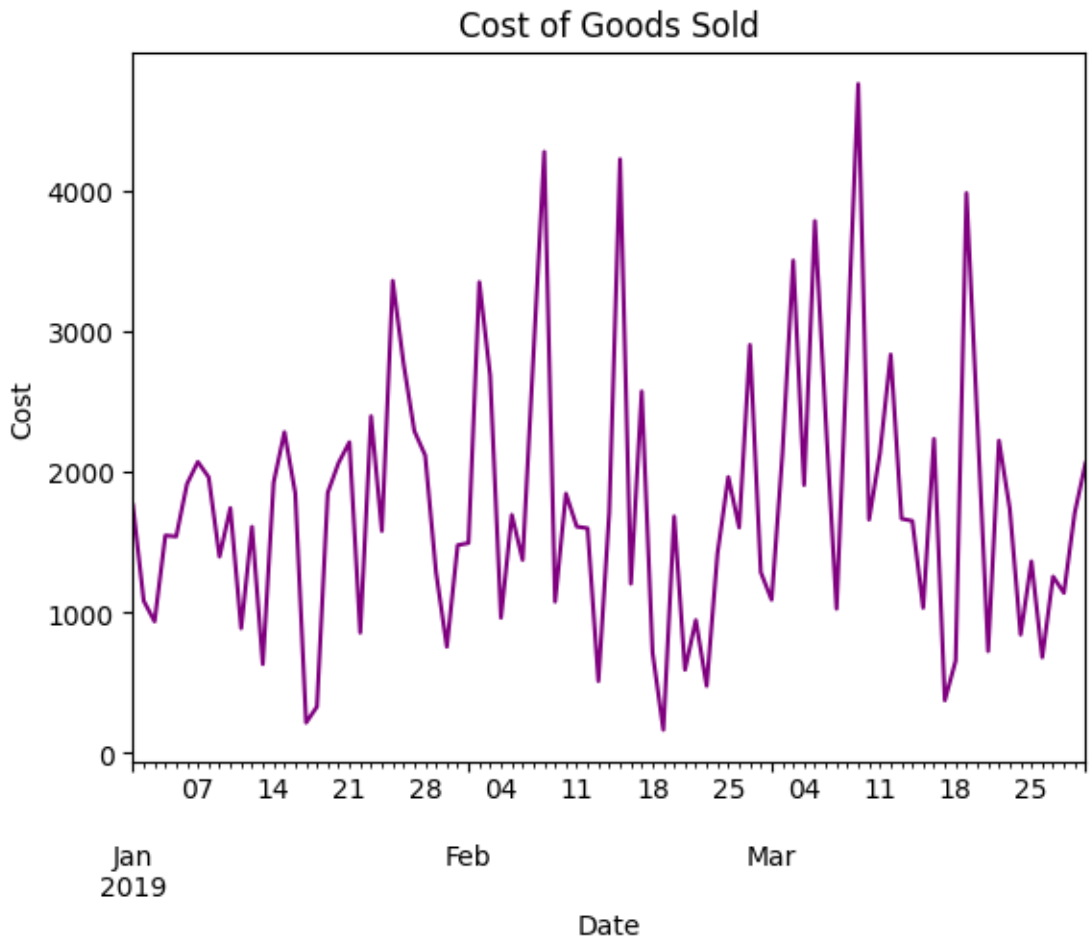
[65]:
```
Histogram:df['Unit price'].plot(kind='hist')
```



[66]:
```
import matplotlib.pyplot as plt
```

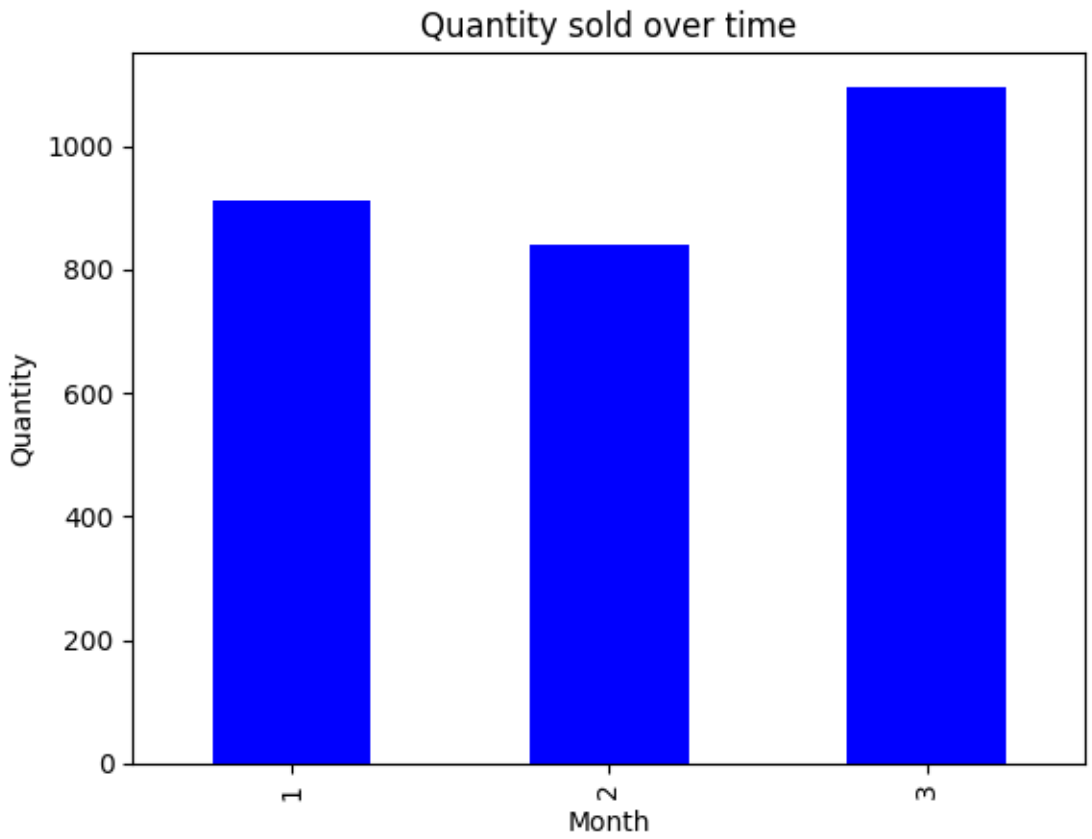
[67]:
```
df.groupby('Date')['cogs'].sum().plot(kind='line',color="purple")
plt.title("Cost of Goods Sold")
plt.xlabel("Date")
plt.ylabel("Cost")
```

[67]:
```
Text(0, 0.5, 'Cost')
```



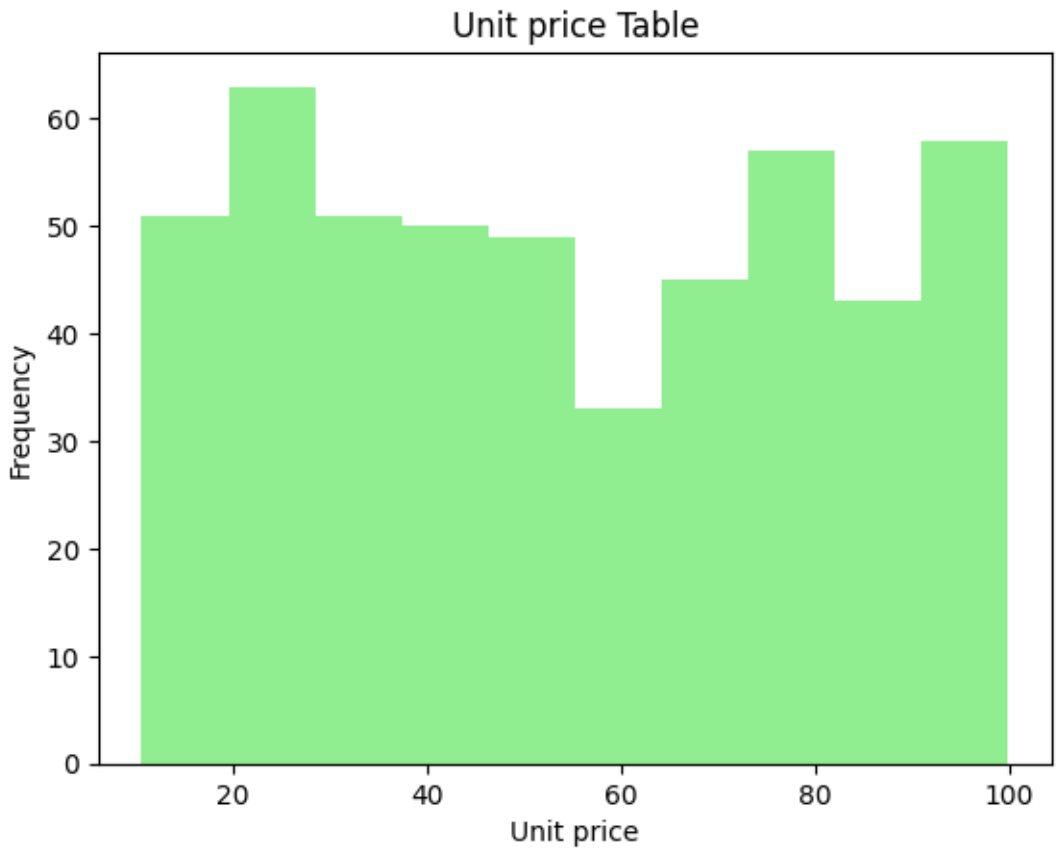
[68]:
```
df.groupby('Month')['Quantity'].sum().plot(kind ="bar",color='blue')
plt.title('Quantity sold over time')
plt.xlabel('Month')
plt.ylabel('Quantity')
```

[68]: Text(0, 0.5, 'Quantity')



[69]:
```
df['Unit price'].plot(kind='hist',color='Lightgreen')
plt.title('Unit price Table')
plt.xlabel('Unit price')
plt.ylabel('Frequency')
```

[69]: Text(0, 0.5, 'Frequency')



[ ]:
```
df.isnull().sum()
```

[71]:
```
df_no_duplicates=df.drop_duplicates('Product line')
```

[72]:
```
df.rename(columns={'cogs':'Cost_of_Goods_Sold'},inplace=True)
print(df.rename)
df.rename(columns={'Cost_of_Goods_Sold':'cogs'},inplace=True)
print(df.rename)
```

```
<bound method DataFrame.rename of       Invoice ID Branch       City Customer
type  Gender  \
0    750-67-8428      A     Yangon        Member  Female
1    226-31-3081      C  Naypyitaw        Normal  Female
2    631-41-3108      A     Yangon        Normal    Male
3    123-19-1176      A     Yangon        Member    Male
4    373-73-7910      A     Yangon        Normal    Male
..           …    …        …             …      …
495  632-32-4574      B   Mandalay        Normal    Male
496  556-97-7101      C  Naypyitaw        Normal  Female
497  862-59-8517      C  Naypyitaw        Normal  Female
```

```
498  401-18-8016      B   Mandalay        Member  Female
499  420-18-8989      A     Yangon        Member  Female

               Product line  Unit price  Quantity   Tax 5%     Total  \
0         Health and beauty       74.69         7  26.1415  548.9715
1    Electronic accessories       15.28         5   3.8200   80.2200
2        Home and lifestyle       46.33         7  16.2155  340.5255
3         Health and beauty       58.22         8  23.2880  489.0480
4         Sports and travel       86.31         7  30.2085  634.3785
..                      …          …       …      …       …
495       Sports and travel       75.92         8  30.3680  637.7280
496  Electronic accessories       63.22         2   6.3220  132.7620
497      Food and beverages       90.24         6  27.0720  568.5120
498       Sports and travel       98.13         1   4.9065  103.0365
499       Sports and travel       51.52         8  20.6080  432.7680

          Date      Time      Payment  Cost_of_Goods_Sold  \
0   2019-01-05  0.547222      Ewallet              522.83
1   2019-03-08  0.436806         Cash               76.40
2   2019-03-03  0.557639  Credit card              324.31
3   2019-01-27  0.856250      Ewallet              465.76
4   2019-02-08  0.442361      Ewallet              604.17
..         …       …           …                   …
495 2019-03-20  0.593056         Cash              607.36
496 2019-01-01  0.660417         Cash              126.44
497 2019-01-27  0.470139         Cash              541.44
498 2019-01-21  0.733333         Cash               98.13
499 2019-02-02  0.657639         Cash              412.16

     gross margin percentage  gross income  Rating  Day  Month  Year
0                   4.761905       26.1415     9.1    5      1  2019
1                   4.761905        3.8200     9.6    8      3  2019
2                   4.761905       16.2155     7.4    3      3  2019
3                   4.761905       23.2880     8.4   27      1  2019
4                   4.761905       30.2085     5.3    8      2  2019
..                       …             …     …  …    …   …
495                 4.761905       30.3680     5.5   20      3  2019
496                 4.761905        6.3220     8.5    1      1  2019
497                 4.761905       27.0720     6.2   27      1  2019
498                 4.761905        4.9065     8.9   21      1  2019
499                 4.761905       20.6080     9.6    2      2  2019

[500 rows x 20 columns]>
<bound method DataFrame.rename of       Invoice ID Branch       City Customer
type  Gender  \
0    750-67-8428      A     Yangon        Member  Female
1    226-31-3081      C  Naypyitaw        Normal  Female
2    631-41-3108      A     Yangon        Normal    Male
```

```
3    123-19-1176      A     Yangon        Member    Male
4    373-73-7910      A     Yangon        Normal    Male
..           …    …        …           …     …
495  632-32-4574      B   Mandalay        Normal    Male
496  556-97-7101      C  Naypyitaw        Normal  Female
497  862-59-8517      C  Naypyitaw        Normal  Female
498  401-18-8016      B   Mandalay        Member  Female
499  420-18-8989      A     Yangon        Member  Female

                Product line  Unit price  Quantity   Tax 5%     Total  \
0         Health and beauty       74.69         7  26.1415  548.9715
1    Electronic accessories       15.28         5   3.8200   80.2200
2        Home and lifestyle       46.33         7  16.2155  340.5255
3         Health and beauty       58.22         8  23.2880  489.0480
4         Sports and travel       86.31         7  30.2085  634.3785
..                      …          …       …      …       …
495       Sports and travel       75.92         8  30.3680  637.7280
496  Electronic accessories       63.22         2   6.3220  132.7620
497      Food and beverages       90.24         6  27.0720  568.5120
498       Sports and travel       98.13         1   4.9065  103.0365
499       Sports and travel       51.52         8  20.6080  432.7680

          Date      Time      Payment    cogs  gross margin percentage  \
0   2019-01-05  0.547222      Ewallet  522.83                 4.761905
1   2019-03-08  0.436806         Cash   76.40                 4.761905
2   2019-03-03  0.557639  Credit card  324.31                 4.761905
3   2019-01-27  0.856250      Ewallet  465.76                 4.761905
4   2019-02-08  0.442361      Ewallet  604.17                 4.761905
..         …       …           …     …                      …
495 2019-03-20  0.593056         Cash  607.36                 4.761905
496 2019-01-01  0.660417         Cash  126.44                 4.761905
497 2019-01-27  0.470139         Cash  541.44                 4.761905
498 2019-01-21  0.733333         Cash   98.13                 4.761905
499 2019-02-02  0.657639         Cash  412.16                 4.761905

     gross income  Rating  Day  Month  Year
0         26.1415     9.1    5      1  2019
1          3.8200     9.6    8      3  2019
2         16.2155     7.4    3      3  2019
3         23.2880     8.4   27      1  2019
4         30.2085     5.3    8      2  2019
..            …     …  …    …   …
495       30.3680     5.5   20      3  2019
496        6.3220     8.5    1      1  2019
497       27.0720     6.2   27      1  2019
498        4.9065     8.9   21      1  2019
499       20.6080     9.6    2      2  2019
```

```
[500 rows x 20 columns]>
```

[73]:
```
grouped_df=df.groupby('City')['Total'].mean()
print(grouped_df)
```

```
City
Mandalay      333.602544
Naypyitaw     336.058800
Yangon        320.229692
Name: Total, dtype: float64
```

[74]:
```
agg_df=df.groupby('City').agg({ 'Total':'sum'})
print(agg_df)

agg_df=df.groupby('City').agg({ 'Total':'max'})
print(agg_df)

agg_df=df.groupby('City').agg({ 'Total':'count'})
print(agg_df)
```

```
                Total
City
Mandalay    52709.2020
Naypyitaw   58810.2900
Yangon      53478.3585
               Total
City
Mandalay     944.622
Naypyitaw   1042.650
Yangon      1039.290
            Total
City
Mandalay      158
Naypyitaw     175
Yangon        167
```

[75]:
```
joined_df=df[['Customer type','Payment']]
print(joined_df.head())
merged_df=pd.merge(df[['City','Payment','Total']],df[['City','Year','Customer␣
 ↪type']], on='City',how='inner')
print(merged_df.head())
```

```
  Customer type      Payment
0        Member      Ewallet
1        Normal         Cash
2        Normal  Credit card
3        Member      Ewallet
4        Normal      Ewallet
     City  Payment     Total  Year Customer type
```

```
0  Yangon  Ewallet  548.9715  2019        Member
1  Yangon  Ewallet  548.9715  2019        Normal
2  Yangon  Ewallet  548.9715  2019        Member
3  Yangon  Ewallet  548.9715  2019        Normal
4  Yangon  Ewallet  548.9715  2019        Member
```

[ ]:
```
concatenated_df=pd.concat([df['City'],df['Total']],axis=1)
print(concatenated_df)
```

[ ]:
```
 df_sorted=df.sort_values(by='Unit price',ascending=False)
print(df_sorted)
```

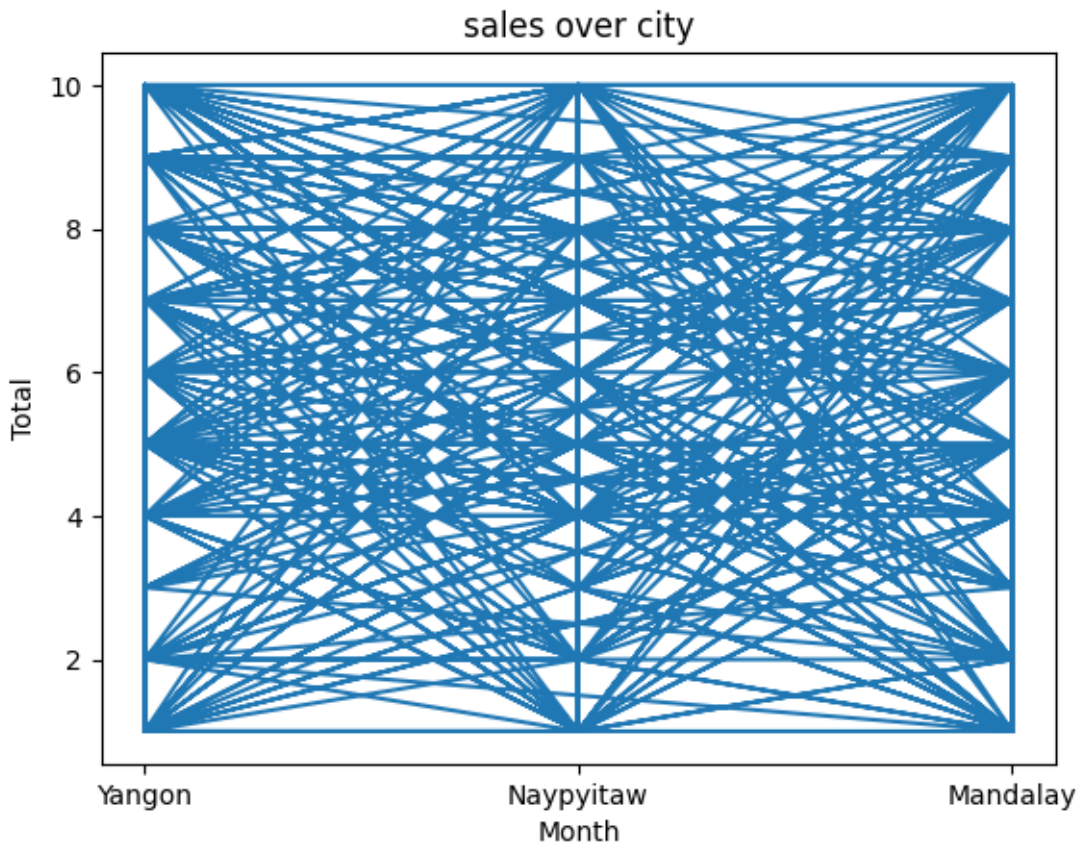
[ ]:
```
df_sorted=df.sort_values(by='Total',ascendin=True)
print(df_sorted)
```

[ ]:
```
filtered_df=df[df['Quantity']>5]
print(filtered_df)
```

[ ]:
```
filtered_df=df[df['cogs']>600]
print(filtered_df)
```

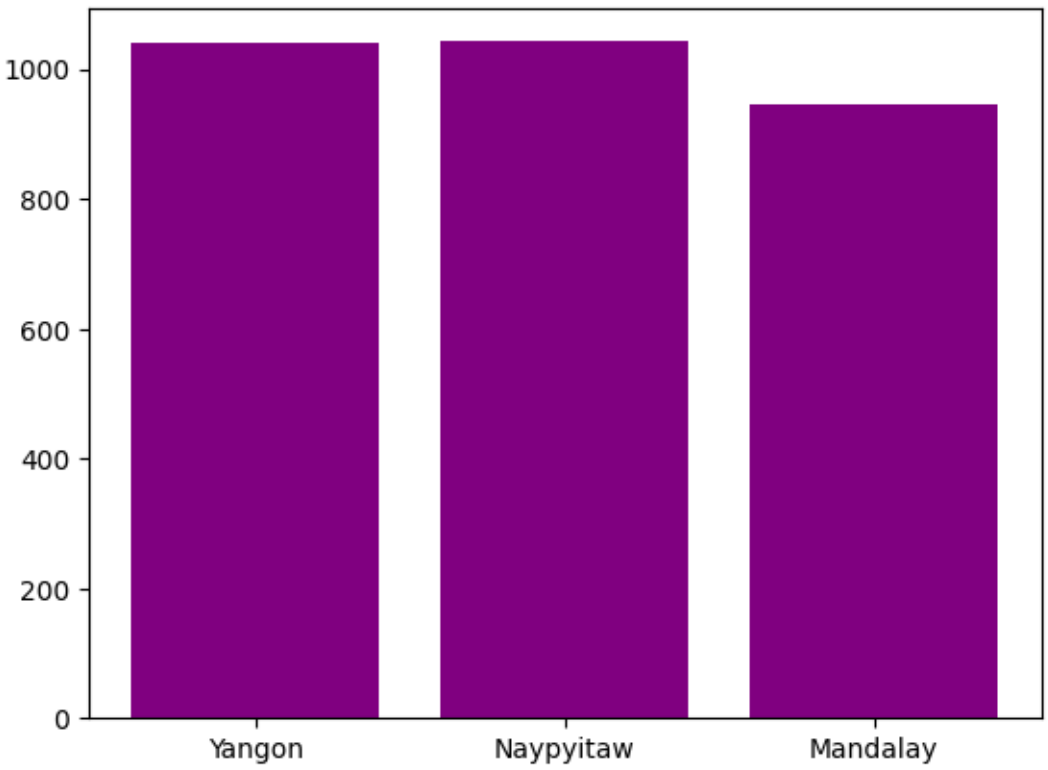
[ ]:
```
#Data visulaization with matplotlib and seaborn
```

[77]:
```
plt.plot(df['City'],df['Quantity'])
plt.title('sales over city')
plt.xlabel('Month')
plt.ylabel('Total')
plt.show()
```



[78]:
```
plt.bar(df['City'],df['Total'],color='purple')
plt.show
```

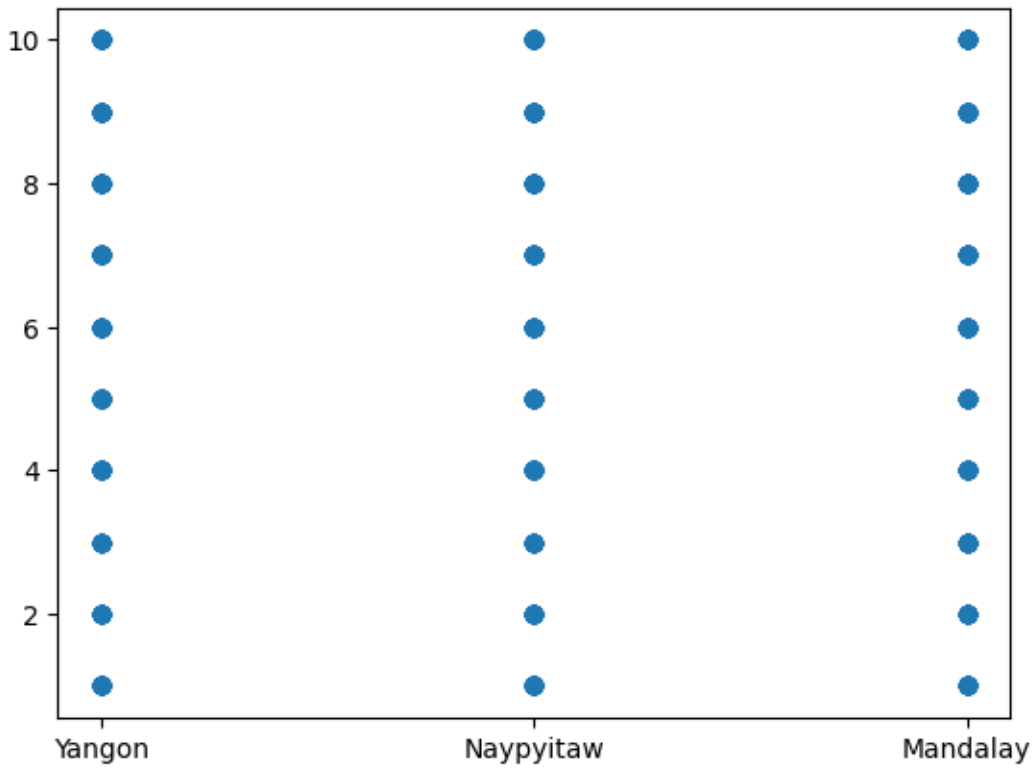
[78]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



[ ]:
```
import numpy as np
```

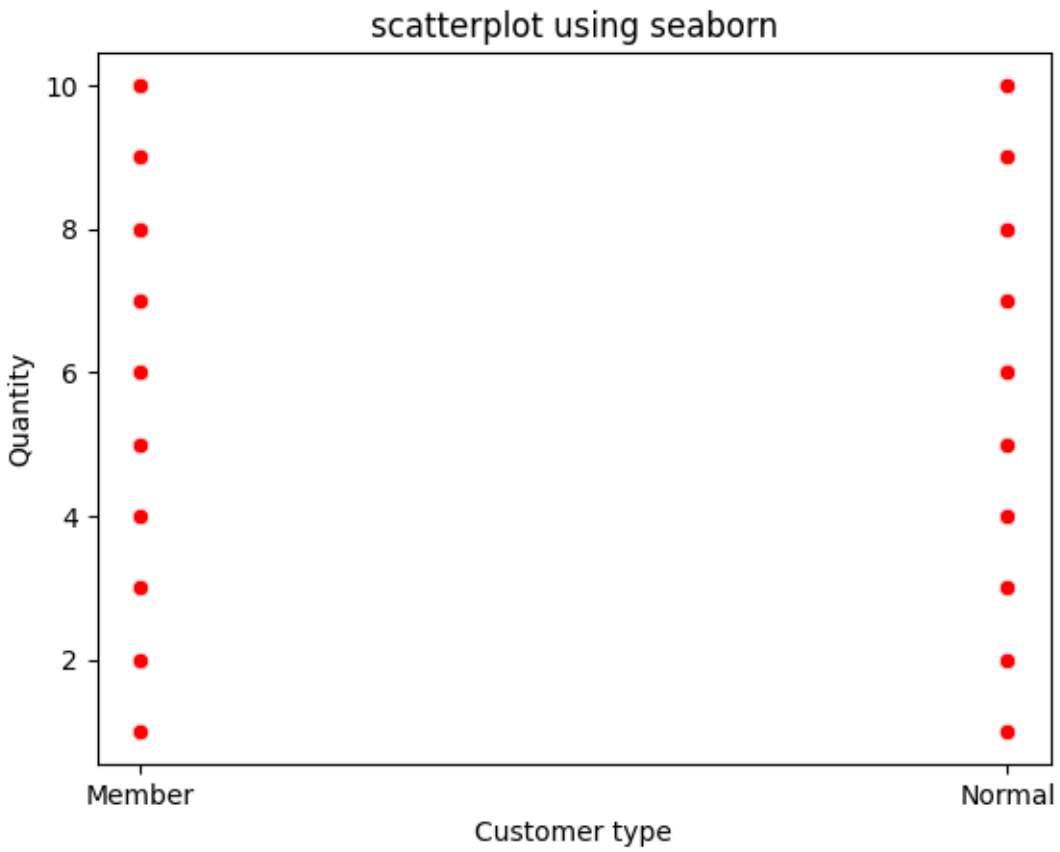
[ ]:
```
data=np.random.randn(1000)
plt.hist(data,bins=15,edgecolor='black')
plt.show()
```

[79]:
```
plt.scatter(df['City'],df['Quantity'])
plt.show()
```

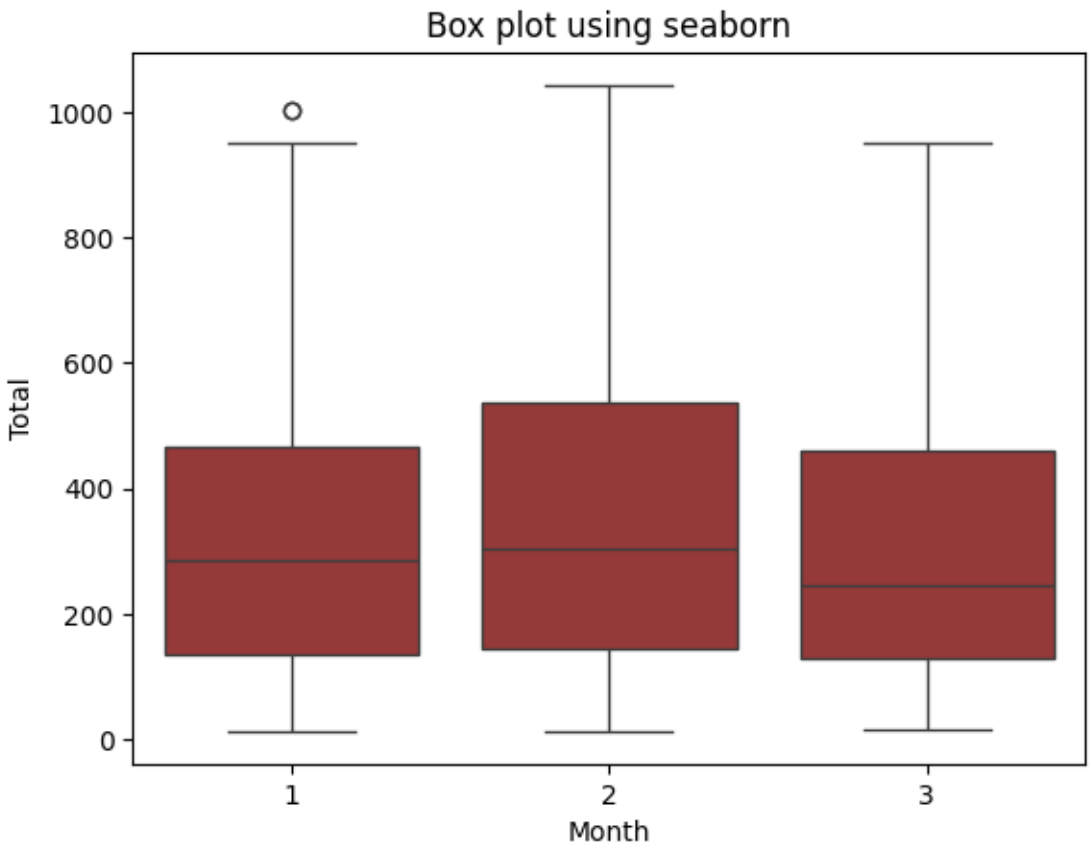


[ ]:
```
!pip install seaborn
import seaborn as sns
```

[80]:
```
sns.scatterplot(x='Customer type',y='Quantity',data=df,color='red')
plt.title('scatterplot using seaborn')
plt.show()
```

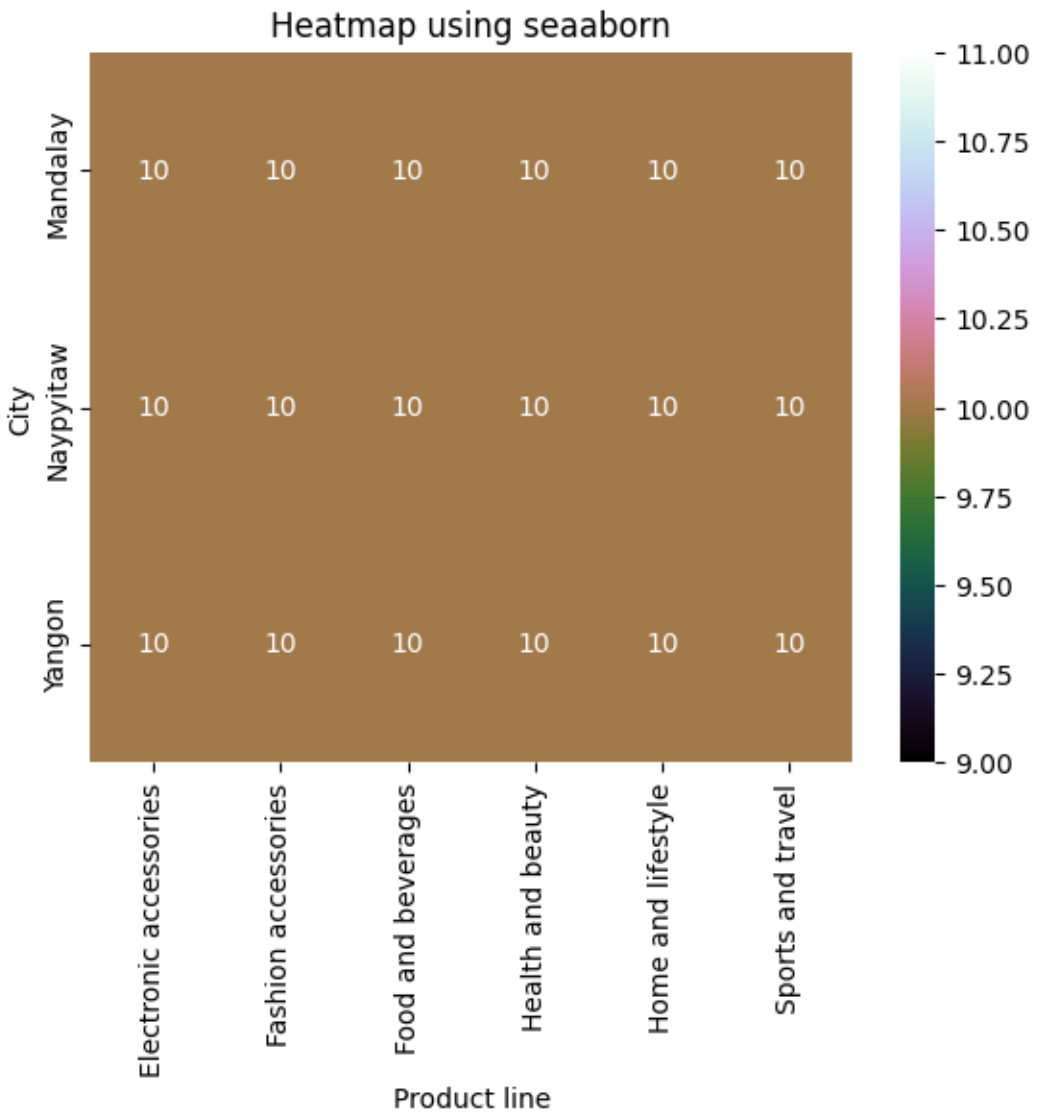


[81]:
```
sns.boxplot(x='Month',y='Total',data=df,color='brown')
plt.title('Box plot using seaborn')
plt.show()
```



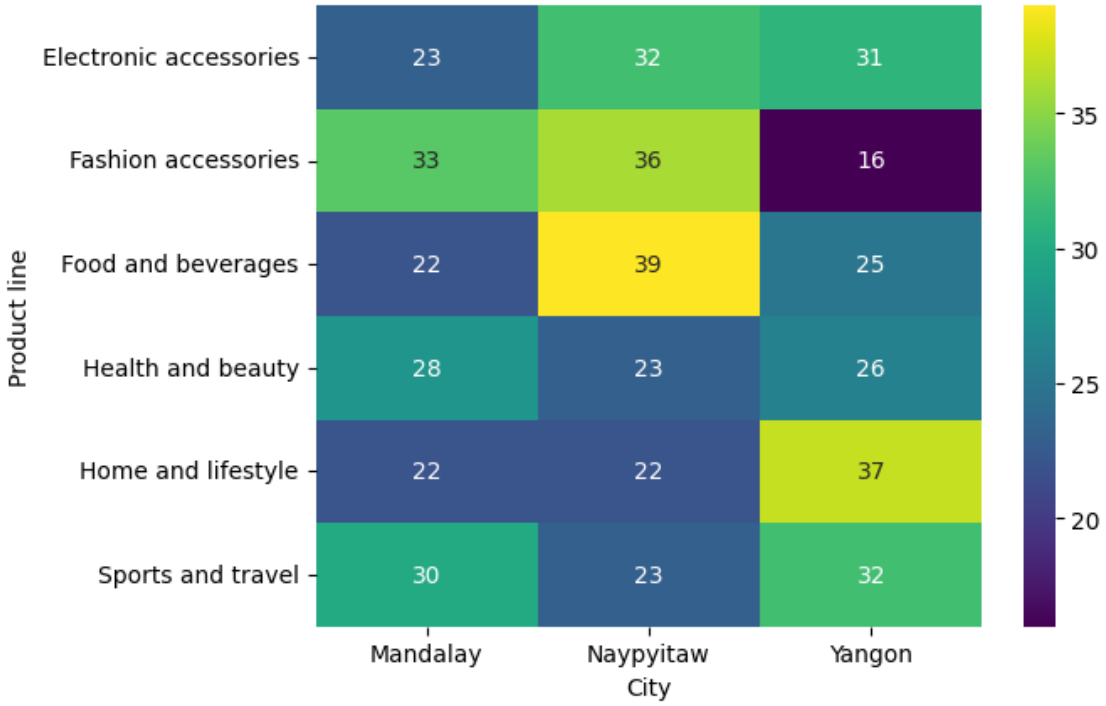
[ ]:
```
numeric_df=df[['Unit price','Total','Quantity','cogs','gross income']]
correlation_matrix = numeric_df.corr()
sns.heatmap(correlation_matrix,annot=True,cmap='coolwarm_r')
plt.title('Heatmap using seaaborn')
plt.show()
```

[82]:
```
pivot_table=df.pivot_table(index='City',columns='Product␣
↪line',values='Quantity',aggfunc='max')
sns.heatmap(pivot_table,annot=True,cmap='cubehelix')
plt.title('Heatmap using seaaborn')
plt.xlabel('Product line')
plt.ylabel('City')
plt.show()
```

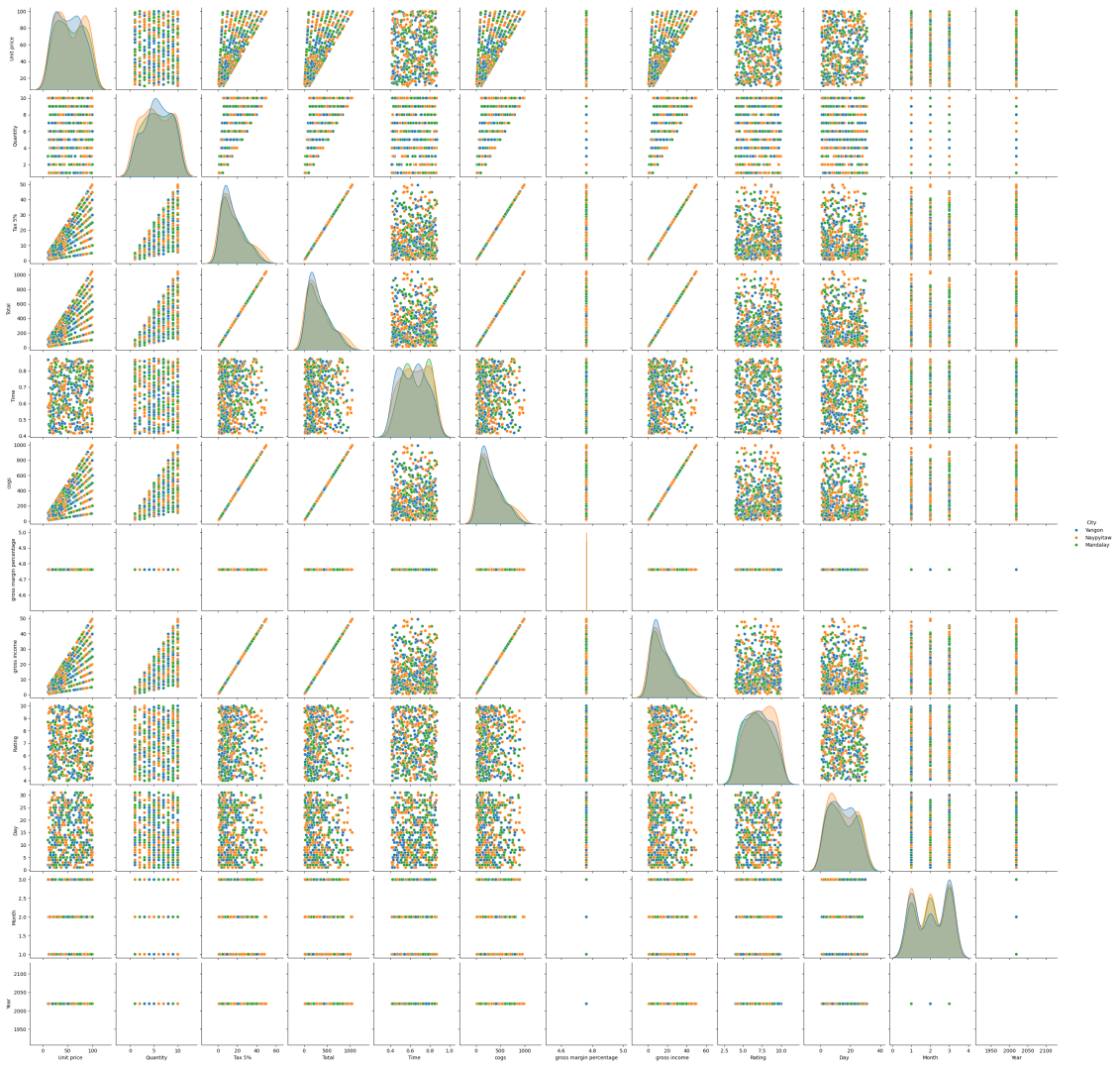


[83]:
```
pivot_table=df.pivot_table(index='Product line',columns='City',values='Quantity',aggfunc='count')
sns.heatmap(pivot_table,annot=True,cmap='viridis')
```

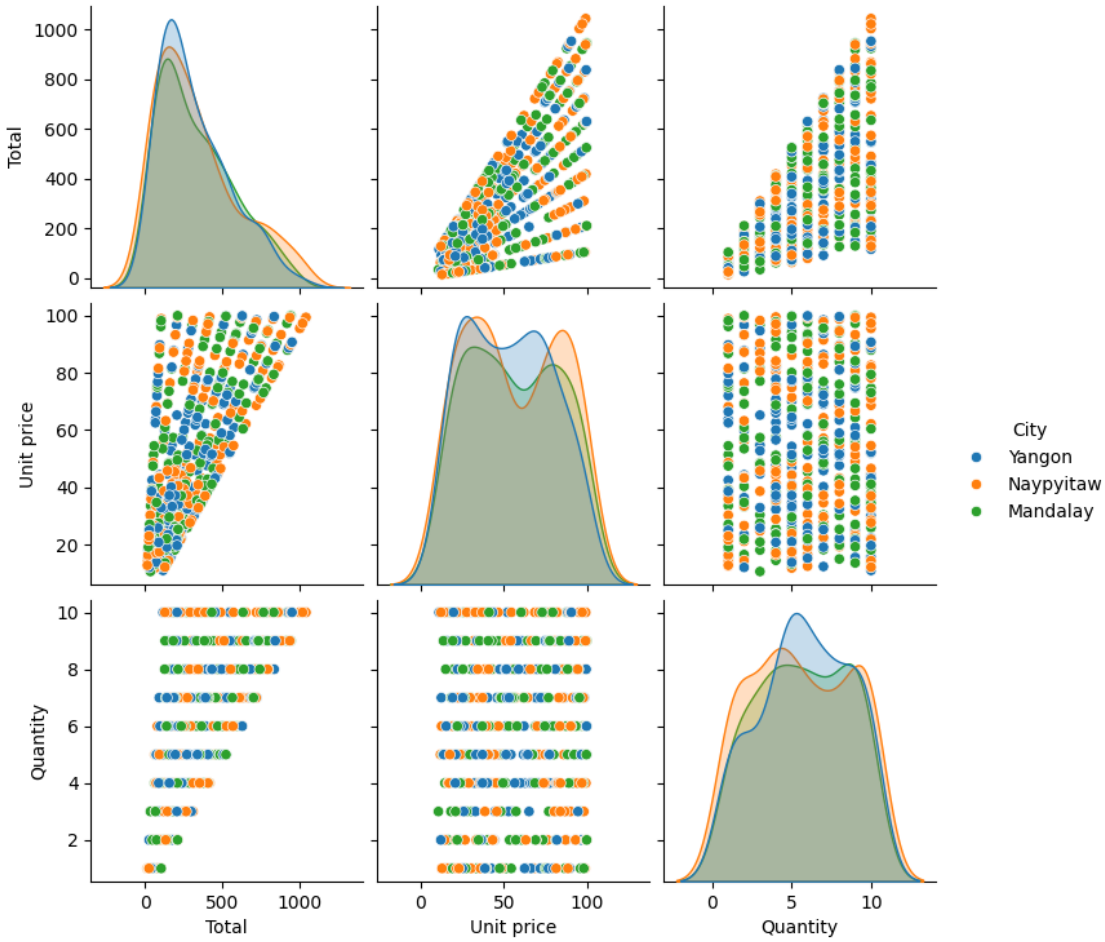
[83]: <Axes: xlabel='City', ylabel='Product line'>



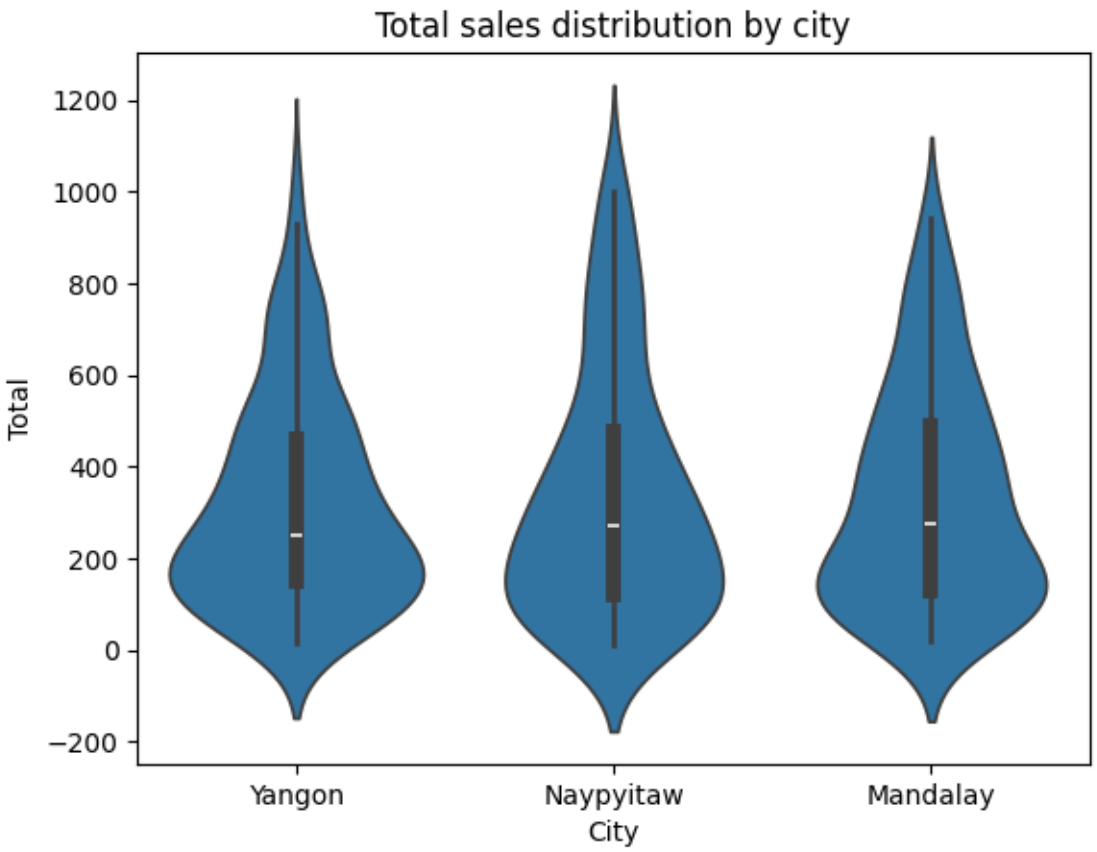
[84]:
```
sns.pairplot(df,hue='City')
plt.show()
```



[85]:
```
sns.pairplot(df[['City','Total','Unit price','Quantity']],hue='City')
plt.show()
```



[87]:
```
sns.violinplot(x='City',y='Total',data=df)
plt.title('Total sales distribution by city')
plt.show()
```



[ ]:
```
jupyter nbconvert --to pdf Python Training Session.ipynb
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: