In [121…

```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn_extra.cluster import KMedoids
```

In [122…

```
df=pd.read_csv("C:/Users/NITISH BOKKA/Downloads/archive (16)/sales_data_sample.c
```

In [123…

```
df.isnull().sum()
```

Out[123…

```
ORDERNUMBER            0
QUANTITYORDERED        0
PRICEEACH              0
ORDERLINENUMBER        0
SALES                  0
ORDERDATE              0
STATUS                 0
QTR_ID                 0
MONTH_ID               0
YEAR_ID                0
PRODUCTLINE            0
MSRP                   0
PRODUCTCODE            0
CUSTOMERNAME           0
PHONE                  0
ADDRESSLINE1           0
ADDRESSLINE2        2521
CITY                   0
STATE               1486
POSTALCODE            76
COUNTRY                0
TERRITORY           1074
CONTACTLASTNAME        0
CONTACTFIRSTNAME       0
DEALSIZE               0
dtype: int64
```

In [124…

```
df['ADDRESSLINE2']=df['ADDRESSLINE2'].bfill()
df['POSTALCODE']=df['POSTALCODE'].ffill()
df['STATE']=df['STATE'].ffill()
df['TERRITORY']=df['TERRITORY'].bfill()
```

In [125…

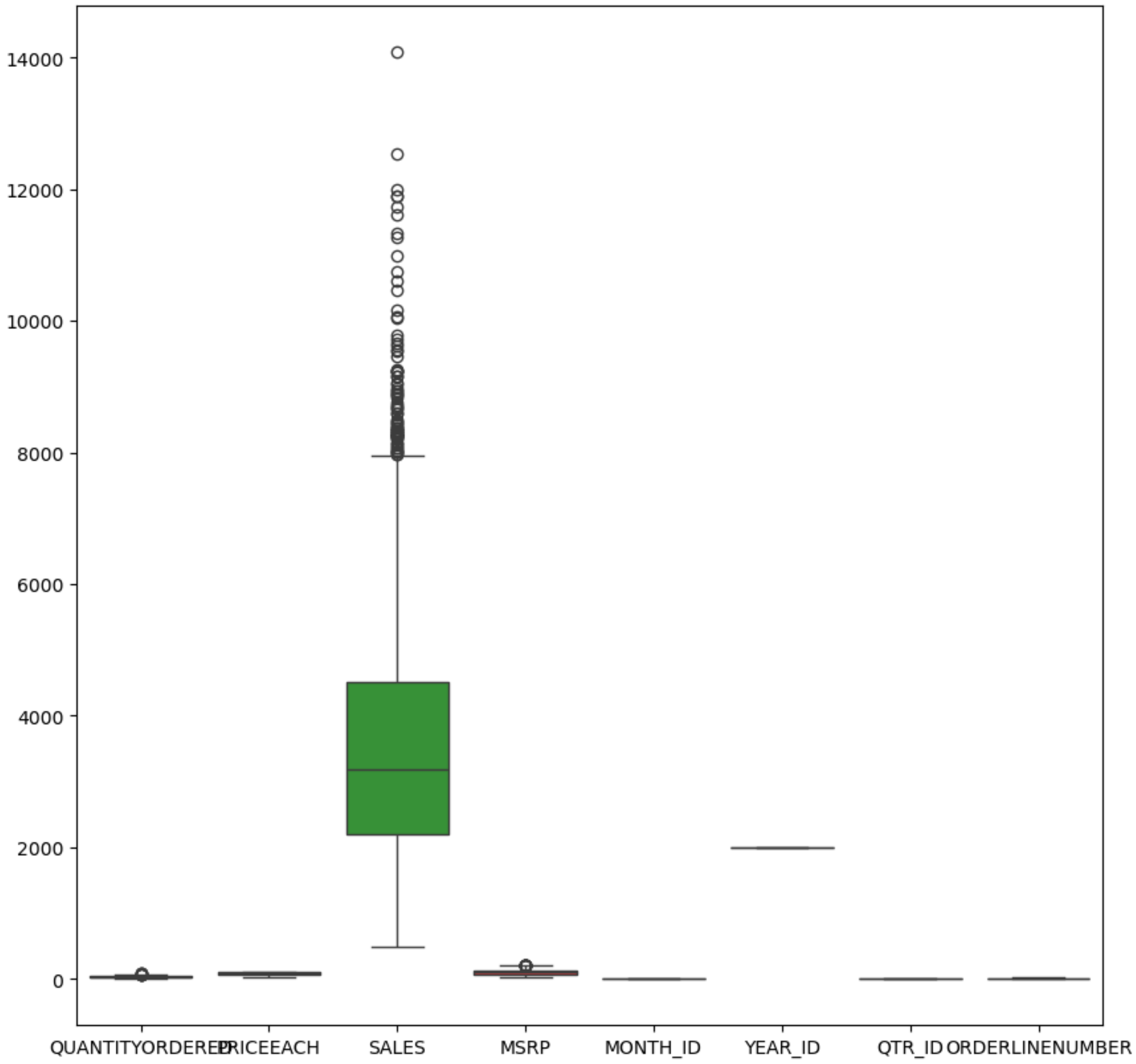
```
df.isnull().sum()
```

Out[125…

```
ORDERNUMBER         0
QUANTITYORDERED     0
PRICEEACH           0
ORDERLINENUMBER     0
SALES               0
ORDERDATE           0
STATUS              0
QTR_ID              0
MONTH_ID            0
YEAR_ID             0
PRODUCTLINE         0
MSRP                0
PRODUCTCODE         0
CUSTOMERNAME        0
PHONE               0
ADDRESSLINE1        0
ADDRESSLINE2        5
CITY                0
STATE               0
POSTALCODE          0
COUNTRY             0
TERRITORY           1
CONTACTLASTNAME     0
CONTACTFIRSTNAME    0
DEALSIZE            0
dtype: int64
```

In [126…

```python
num_cols=df[['QUANTITYORDERED','PRICEEACH','SALES','MSRP','MONTH_ID','YEAR_ID','
plt.figure(figsize=(10,10))
sns.boxplot(num_cols)
plt.show()
```



In [127…

```
Q1=df['QUANTITYORDERED'].quantile(0.25)
Q3=df['QUANTITYORDERED'].quantile(0.75)
IQR=Q3-Q1
lower_quartile=Q1-1.5*IQR
upper_quartile=Q3+1.5*IQR
df=df[(df['QUANTITYORDERED']>=lower_quartile) & (df['QUANTITYORDERED']<=upper_qu
```

In [128…

```
Q1=df['PRICEEACH'].quantile(0.25)
Q3=df['PRICEEACH'].quantile(0.75)
IQR=Q3-Q1
lower_quartile=Q1-1.5*IQR
upper_quartile=Q3+1.5*IQR
df=df[(df['PRICEEACH']>=lower_quartile) & (df['PRICEEACH']<=upper_quartile)]
```

In [129…

```
Q1=df['SALES'].quantile(0.25)
Q3=df['SALES'].quantile(0.75)
IQR=Q3-Q1
lower_quartile=Q1-1.5*IQR
upper_quartile=Q3+1.5*IQR
df=df[(df['SALES']>=lower_quartile) & (df['SALES']<=upper_quartile)]
```

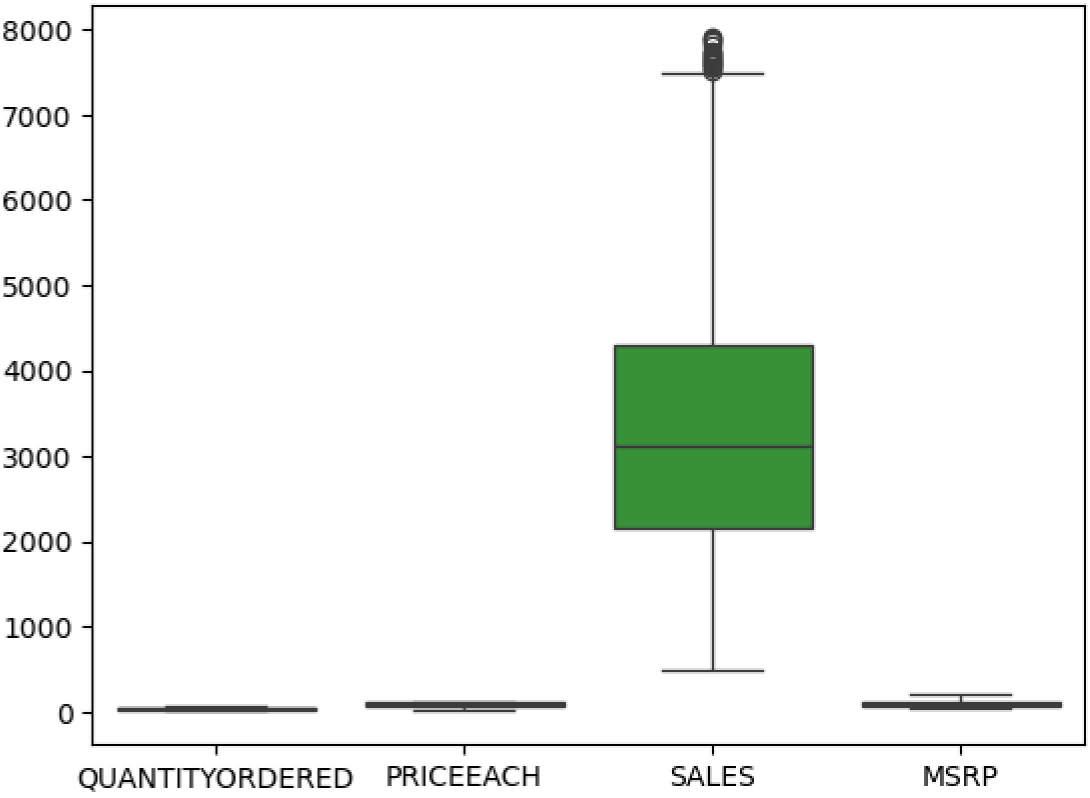
In [130…

```
Q1=df['MSRP'].quantile(0.25)
Q3=df['MSRP'].quantile(0.75)
IQR=Q3-Q1
lower_quartile=Q1-1.5*IQR
```

```
upper_quartile=Q3+1.5*IQR
df=df[(df['MSRP']>=lower_quartile) & (df['MSRP']<=upper_quartile)]
```

In [131…

```
num_cols=df[['QUANTITYORDERED','PRICEEACH','SALES','MSRP']]
sns.boxplot(num_cols)
plt.show()
```



In [132…

```
X=df[['QUANTITYORDERED','PRICEEACH','SALES','MSRP','MONTH_ID','YEAR_ID','QTR_ID'
scaler=StandardScaler()
X_scaled=scaler.fit_transform(X)
```

In [133…

```
k_range=range(1,21)
inertia_list=[]
for i in k_range:
    kmn=KMeans(n_clusters=i, random_state=42)
    kmn.fit(X_scaled)
    inertia_list.append(kmn.inertia_)
    print(i, kmn.inertia_)
```

```
1 21608.000000000007
2 16641.968372583986
3 13612.012474606314
4 12084.10539529429
5 10979.658325322398
6 10208.235629943432
7 9695.764847973925
8 9198.201876745454
9 8740.722878030774
10 8449.597922175664
11 8069.016953571581
12 7788.595622825555
13 7542.94850594196
14 7293.653760449979
15 7102.3075339065435
16 6963.875367712442
17 6886.105986466975
18 6667.932645102546
19 6504.34582857871
20 6407.093498463479
```

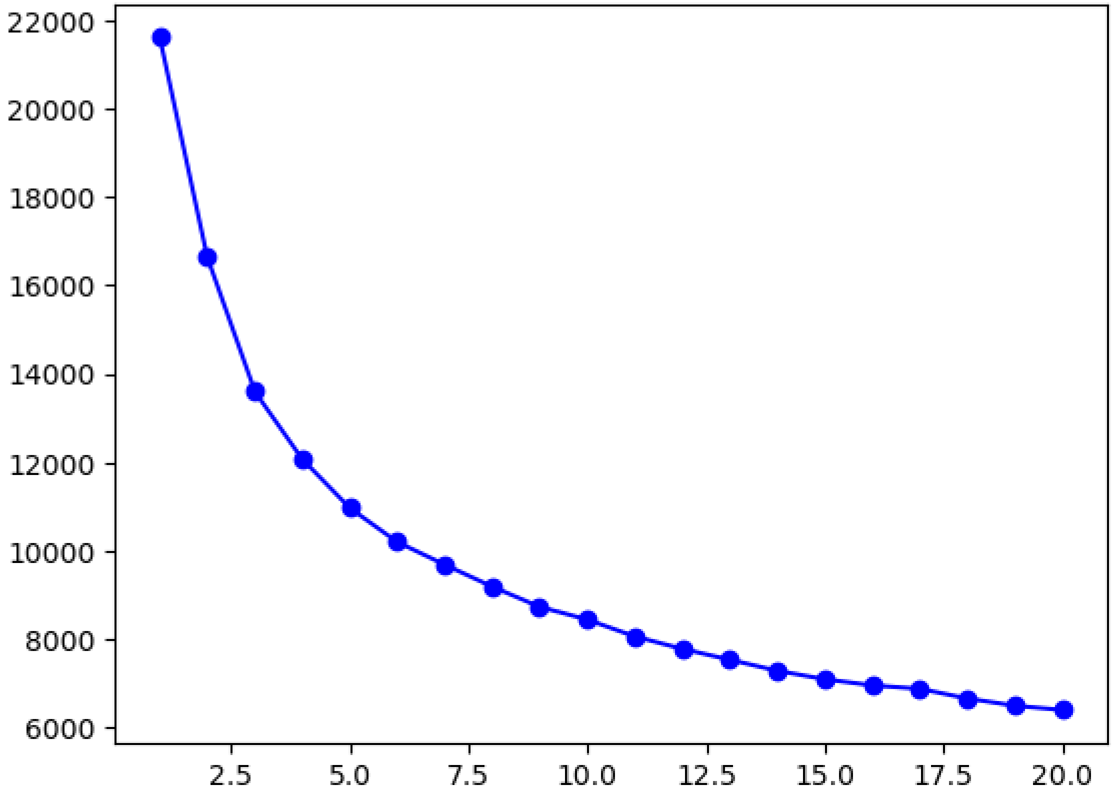
In [134…

```
print("KMeans using Euclidean")
kmn=KMeans(n_clusters=16)
kmn.fit(X_scaled)
print(kmn.inertia_)
```

```
KMeans using Euclidean
7030.311017948243
```

In [135…

```
plt.plot(k_range, inertia_list,'bo-')
plt.show()
```



In [136…

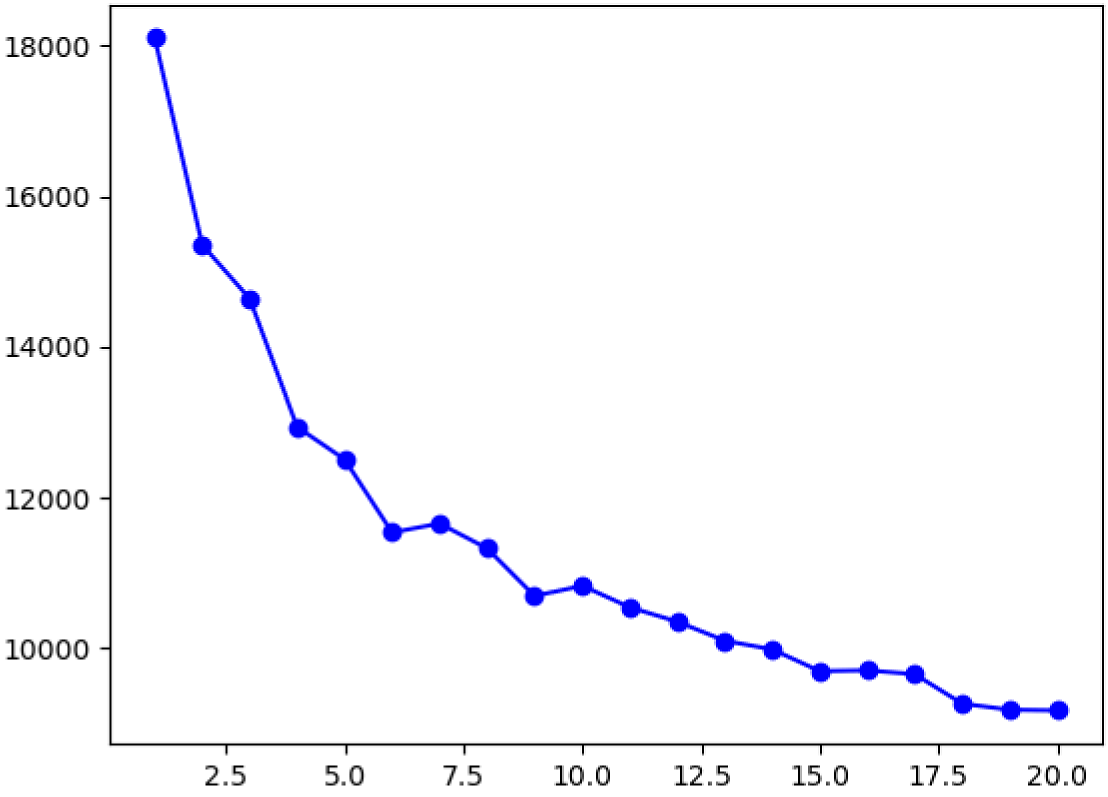
```
k_range=range(1,21)
k_manhattan=[]
inertia_list=[]
```

```
for i in k_range:
    kmn=KMedoids(n_clusters=i, metric='manhattan', random_state=42)
    kmn.fit(X_scaled)
    k_manhattan.append(kmn.inertia_)
    print(i, kmn.inertia_)
```

```
1 18105.313940642394
2 15351.000711074905
3 14633.84880579897
4 12932.104647630194
5 12500.296195025103
6 11527.150473404772
7 11646.616961969872
8 11316.27791752853
9 10682.229800159712
10 10817.487452167577
11 10529.845008100248
12 10339.20748714991
13 10085.709608251495
14 9969.896468579764
15 9680.933295636409
16 9691.009555800505
17 9640.34806520279
18 9248.003626357504
19 9166.397880454051
20 9157.289220324637
```

In [137…

```
plt.plot(k_range, k_manhattan,'bo-')
plt.show()
```



In [171…

```
print("KMeans using Manhattan")
kmn=KMedoids(n_clusters=18,metric='manhattan', random_state=42)
kmn.fit(X_scaled)
print(kmn.inertia_)
```

```
KMeans using Manhattan
9248.003626357504
```

In [ ]: