In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```python
df = pd.read_csv('sales_data_sample.csv', encoding='latin1')
df.head()
```

Out[2]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDA
0	10107	30	95.70	2	2871.00	2/24/20 0:
1	10121	34	81.35	5	2765.90	5/7/2003 0:
2	10134	41	94.74	2	3884.34	7/1/2003 0:
3	10145	45	83.26	6	3746.70	8/25/20 0:
4	10159	49	100.00	14	5205.27	10/10/20 0:

5 rows × 25 columns

In [3]:

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2823 entries, 0 to 2822
Data columns (total 25 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   ORDERNUMBER       2823 non-null   int64  
 1   QUANTITYORDERED   2823 non-null   int64  
 2   PRICEEACH         2823 non-null   float64
 3   ORDERLINENUMBER   2823 non-null   int64  
 4   SALES             2823 non-null   float64
 5   ORDERDATE         2823 non-null   object 
 6   STATUS            2823 non-null   object 
 7   QTR_ID            2823 non-null   int64  
 8   MONTH_ID          2823 non-null   int64  
 9   YEAR_ID           2823 non-null   int64  
 10  PRODUCTLINE       2823 non-null   object 
 11  MSRP              2823 non-null   int64  
 12  PRODUCTCODE       2823 non-null   object 
 13  CUSTOMERNAME      2823 non-null   object 
 14  PHONE             2823 non-null   object 
 15  ADDRESSLINE1      2823 non-null   object 
 16  ADDRESSLINE2      302 non-null    object 
 17  CITY              2823 non-null   object 
 18  STATE             1337 non-null   object 
 19  POSTALCODE        2747 non-null   object 
 20  COUNTRY           2823 non-null   object 
 21  TERRITORY         1749 non-null   object 
 22  CONTACTLASTNAME   2823 non-null   object 
 23  CONTACTFIRSTNAME  2823 non-null   object 
 24  DEALSIZE          2823 non-null   object 
dtypes: float64(2), int64(7), object(16)
memory usage: 551.5+ KB
```

In [4]:

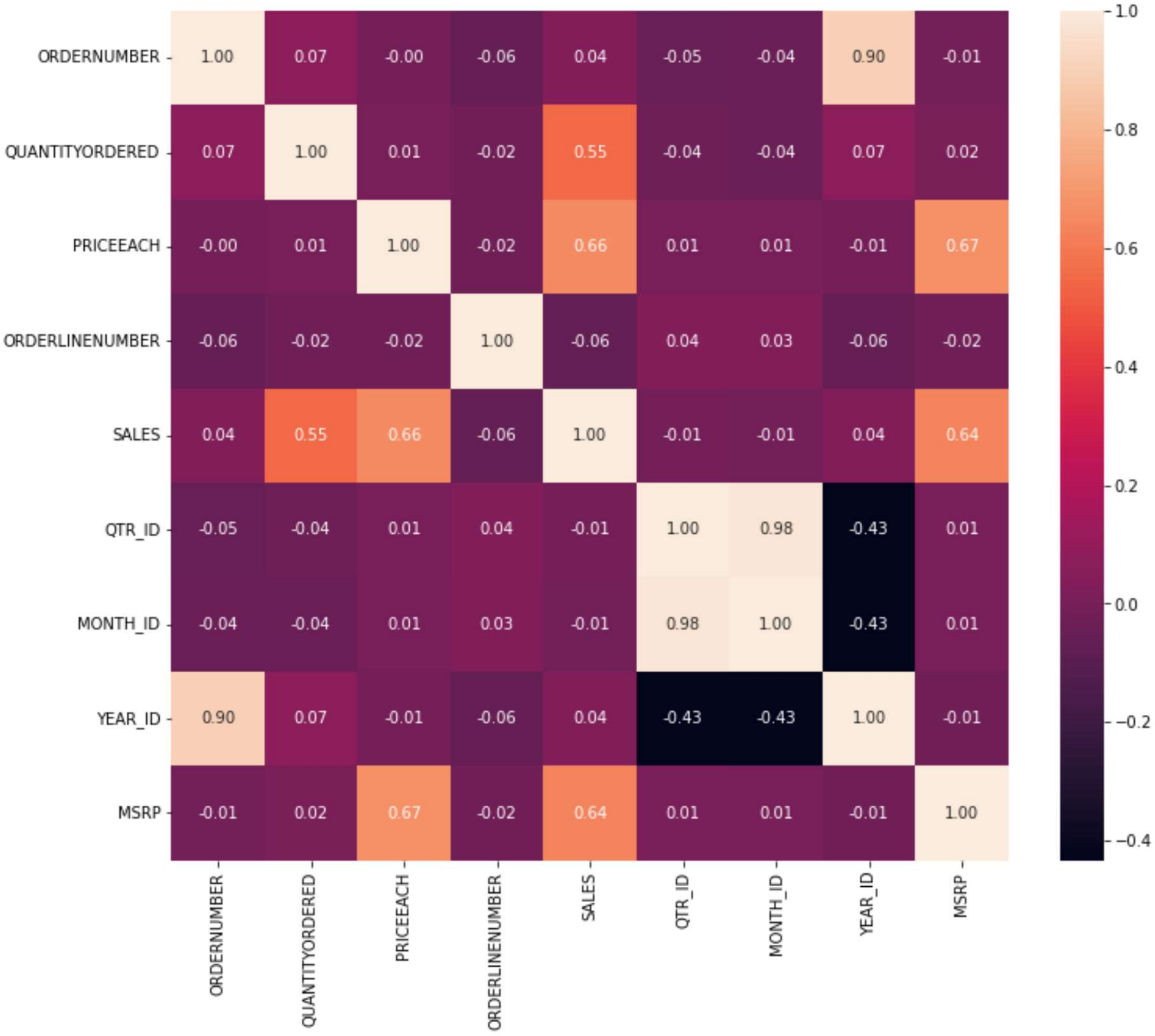
```python
df.describe()
```

Out[4]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES
count	2823.000000	2823.000000	2823.000000	2823.000000	2823.000000
mean	10258.725115	35.092809	83.658544	6.466171	3553.889072
std	92.085478	9.741443	20.174277	4.225841	1841.865106
min	10100.000000	6.000000	26.880000	1.000000	482.130000
25%	10180.000000	27.000000	68.860000	3.000000	2203.430000
50%	10262.000000	35.000000	95.700000	6.000000	3184.800000
75%	10333.500000	43.000000	100.000000	9.000000	4508.000000
max	10425.000000	97.000000	100.000000	18.000000	14082.800000

In [5]:

```
fig = plt.figure(figsize=(12,10))
sns.heatmap(df.corr(), annot=True, fmt='.2f')
plt.show()
```



In [6]:

```
df= df[['PRICEEACH', 'MSRP']]
```

In [7]:

```
df.head()
```

Out[7]:

	PRICEEACH	MSRP
0	95.70	95
1	81.35	95
2	94.74	95
3	83.26	95
4	100.00	95

In [8]:
```
df.isna().any()
```

Out[8]:
```
PRICEEACH    False
MSRP         False
dtype: bool
```

In [9]:
```
df.describe().T
```

Out[9]:

	count	mean	std	min	25%	50%	75%	max
PRICEEACH	2823.0	83.658544	20.174277	26.88	68.86	95.7	100.0	100.0
MSRP	2823.0	100.715551	40.187912	33.00	68.00	99.0	124.0	214.0

In [10]:
```
df.shape
```

Out[10]:
```
(2823, 2)
```

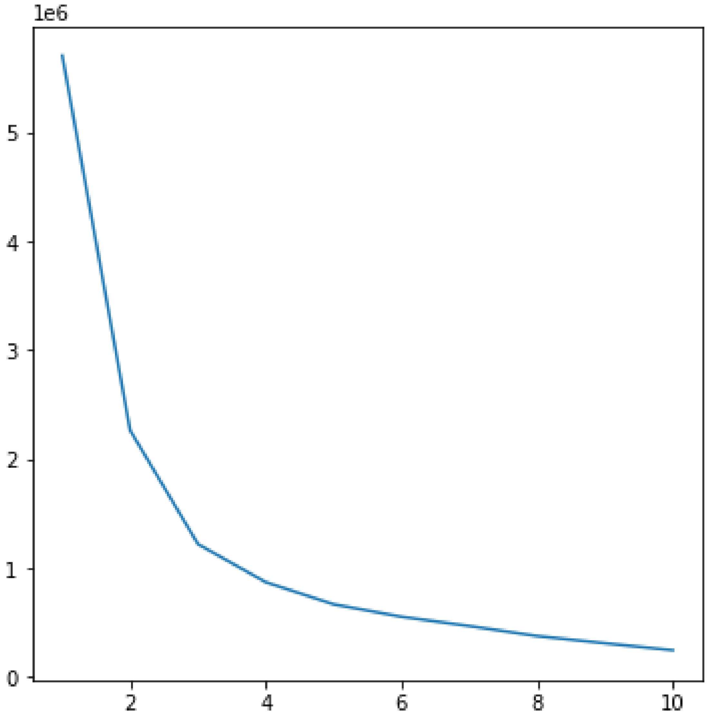
In [11]:
```
from sklearn.cluster import KMeans

inertia = []

for i in range(1, 11):
    clusters = KMeans(n_clusters=i, init='k-means++', random_state=42)
    clusters.fit(df)
    inertia.append(clusters.inertia_)

plt.figure(figsize=(6, 6))
sns.lineplot(x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], y = inertia)
```

Out[11]:
```
<AxesSubplot:>
```

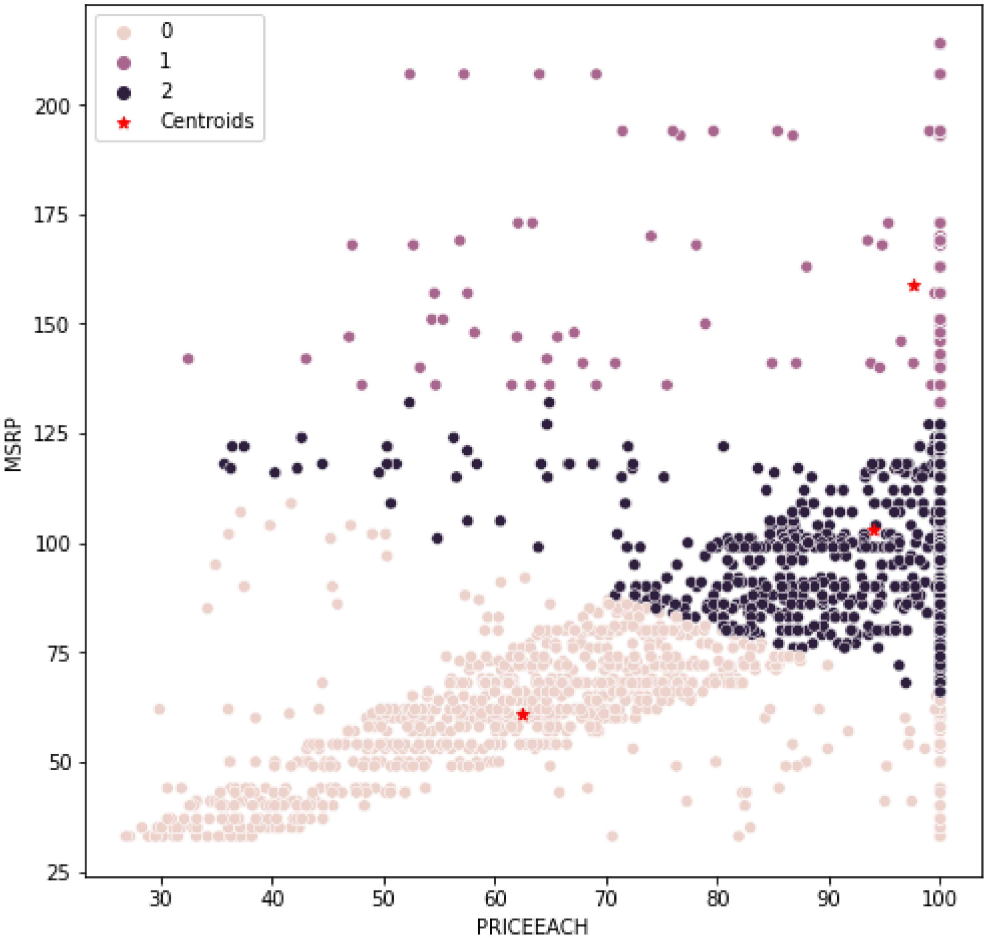


In [12]:
```
kmeans = KMeans(n_clusters = 3, random_state = 42)
y_kmeans = kmeans.fit_predict(df)
y_kmeans
```

Out[12]:
```
array([2, 2, 2, ..., 0, 0, 0], dtype=int32)
```

In [13]:
```
plt.figure(figsize=(8,8))
sns.scatterplot(x=df['PRICEEACH'], y=df['MSRP'], hue=y_kmeans)
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], c =
plt.legend()
```

Out[13]:
```
<matplotlib.legend.Legend at 0x7f9a64686b60>
```



In [14]:
```
kmeans.cluster_centers_
```

Out[14]:
```
array([[ 62.49548902,  60.71556886],
       [ 97.59890263, 158.7202473 ],
       [ 94.03841567, 102.88841567]])
```