In [30]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [31]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [32]:
```
data = pd.read_csv("sales_data_sample.csv", encoding='Latin-1')
data.head()
```

Out[32]:

	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 [33]:
```
data.shape
```

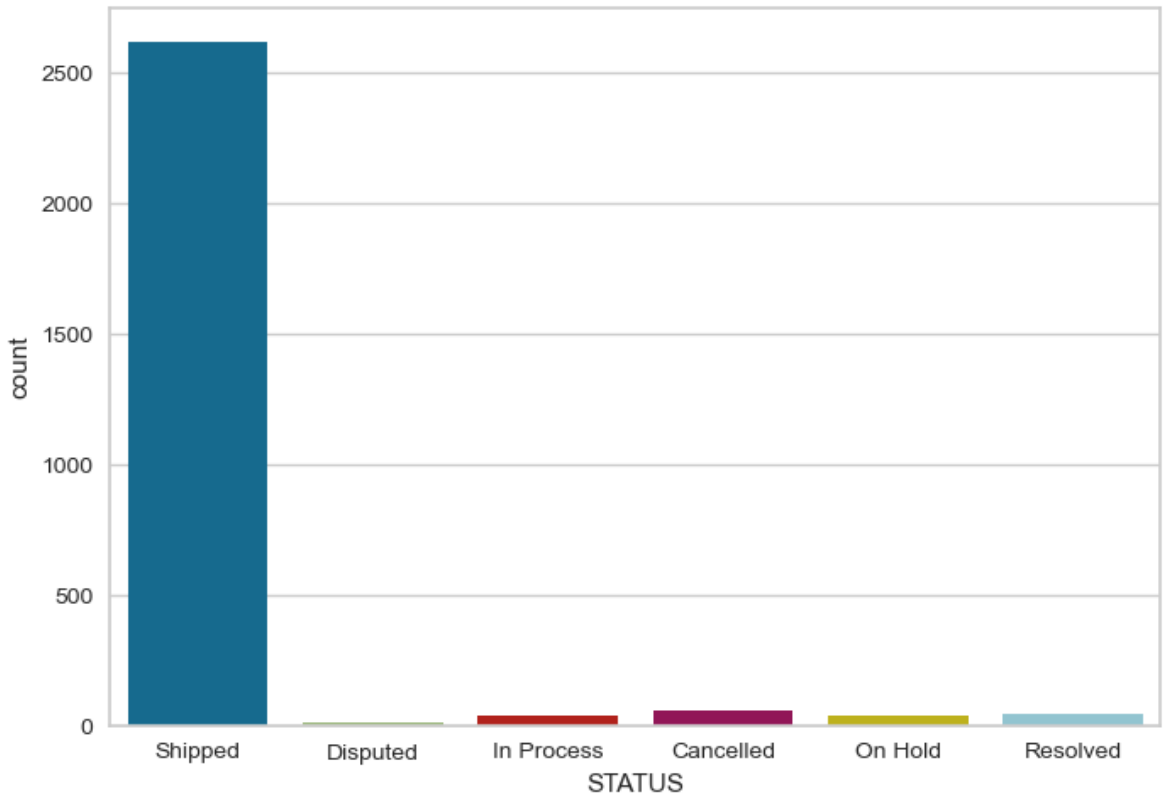
Out[33]:
```
(2823, 25)
```

In [34]:
```
data.isnull().sum()
```

Out[34]:
```
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 [35]:
```
sns.countplot(data = data , x = 'STATUS')
```

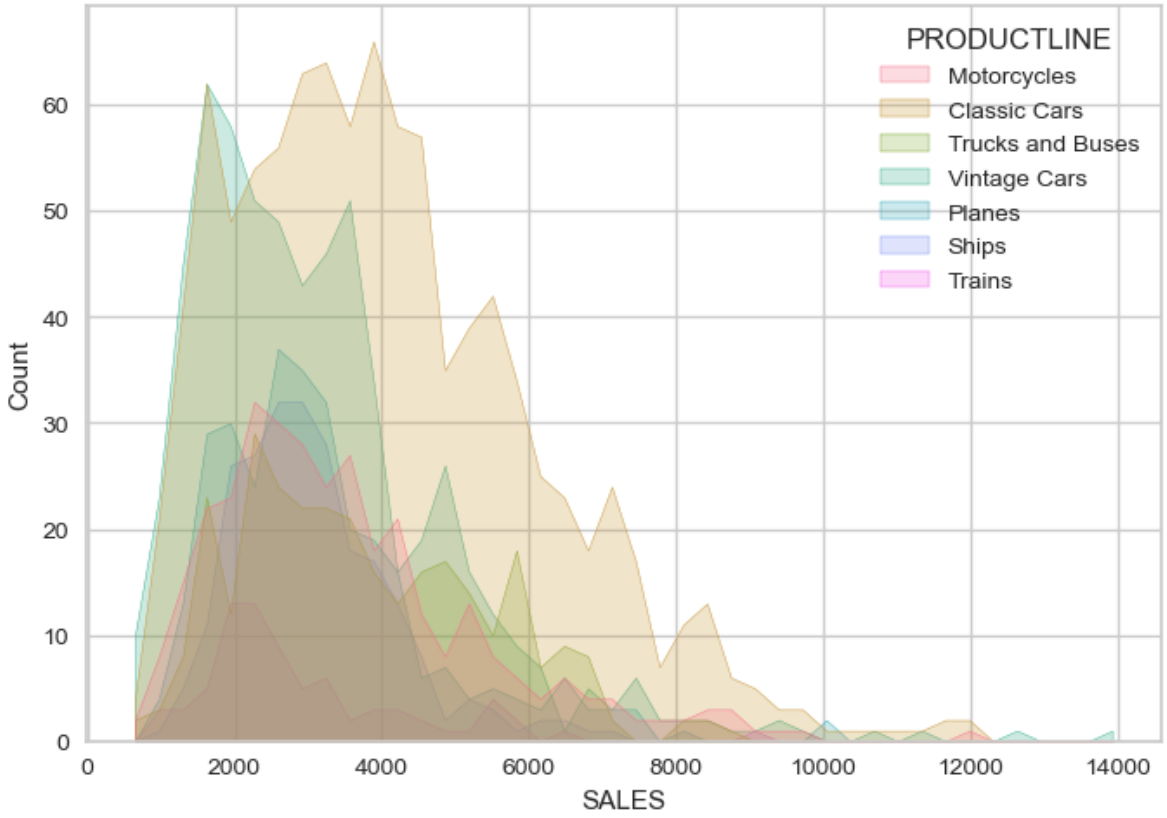
Out[35]:
```
<Axes: xlabel='STATUS', ylabel='count'>
```



In [36]:
```
import seaborn as sns
```

In [37]:
```
sns.histplot(x = 'SALES' , hue = 'PRODUCTLINE', data = data,
             element="poly")
```

Out[37]:
```
<Axes: xlabel='SALES', ylabel='Count'>
```



In [38]:
```
from sklearn.cluster import KMeans
```

In [39]:
```
data['PRODUCTLINE'].unique()
```

Out[39]:
```
array(['Motorcycles', 'Classic Cars', 'Trucks and Buses', 'Vintage Cars',
       'Planes', 'Ships', 'Trains'], dtype=object)
```

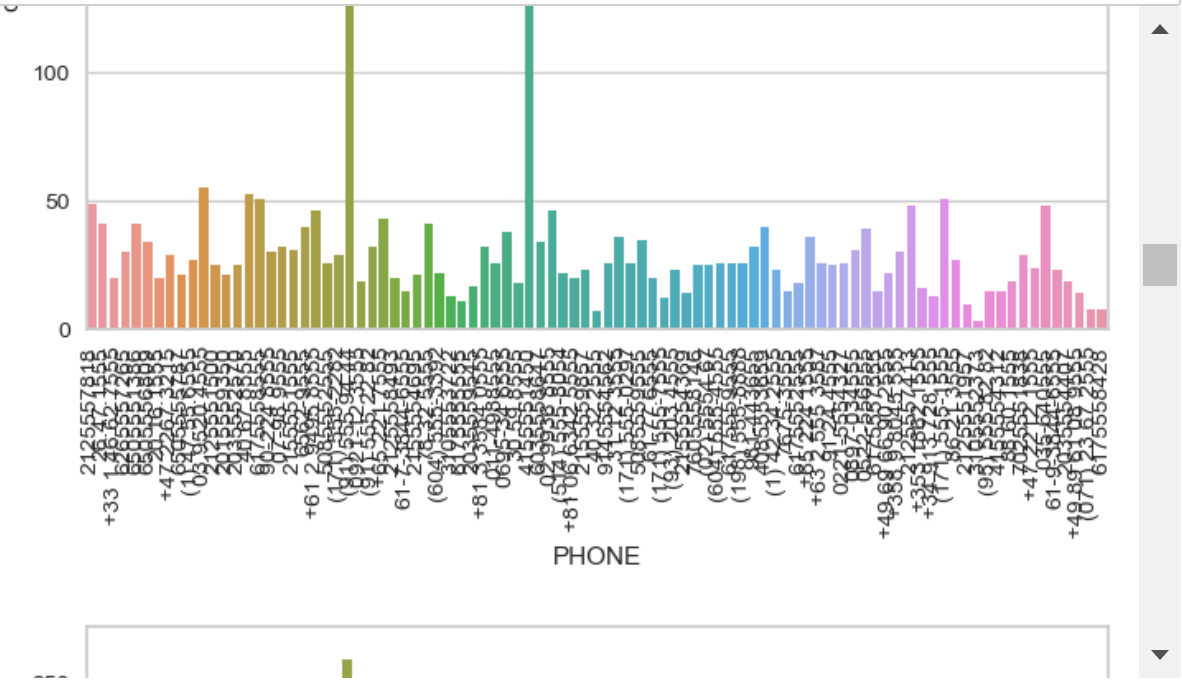
In [40]:
```
data.drop_duplicates(inplace=True)
```

In [41]:
```
list_cat = data.select_dtypes(include=['object']).columns.tolist()
```

In [42]:
```
list_cat
```

Out[42]:
```
['ORDERDATE',
 'STATUS',
 'PRODUCTLINE',
 'PRODUCTCODE',
 'CUSTOMERNAME',
 'PHONE',
 'ADDRESSLINE1',
 'ADDRESSLINE2',
 'CITY',
 'STATE',
 'POSTALCODE',
 'COUNTRY',
 'TERRITORY',
 'CONTACTLASTNAME',
 'CONTACTFIRSTNAME',
 'DEALSIZE']
```

In [43]:
```
for i in list_cat:
  sns.countplot(data = data ,x = i)
  plt.xticks(rotation = 90)
  plt.show()
```



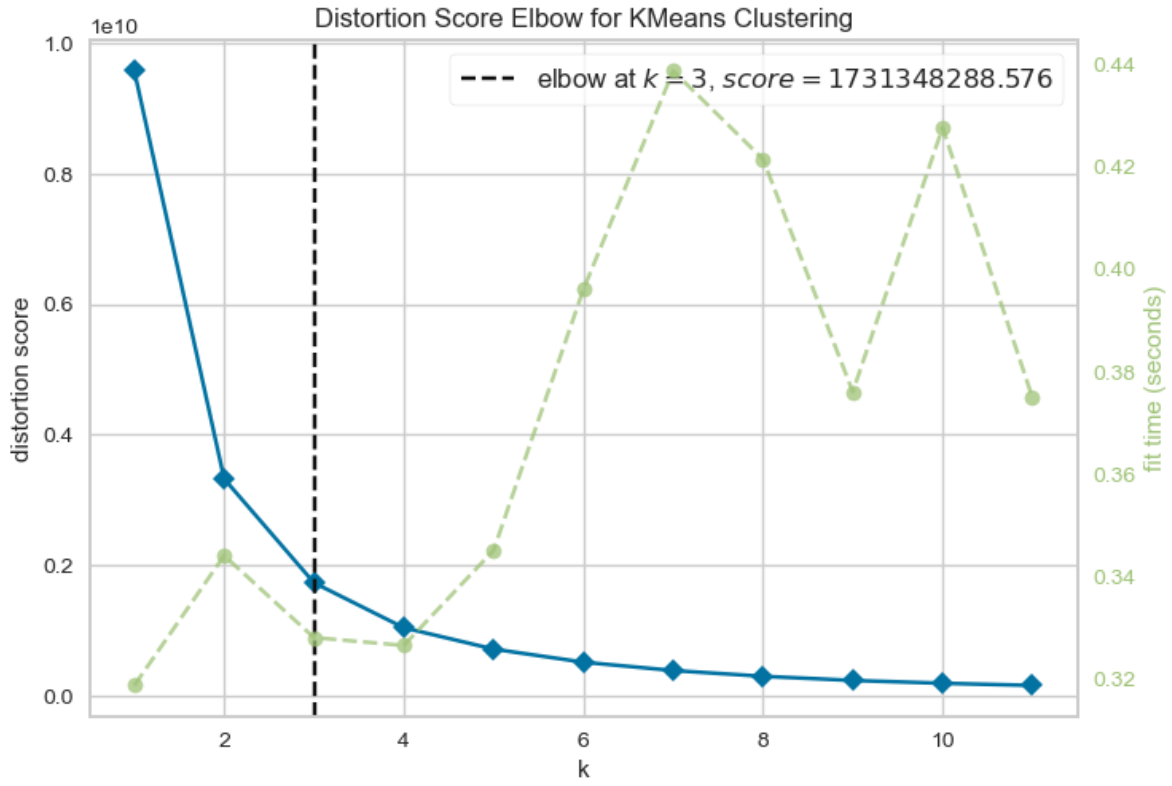
In [44]:
```
#dealing with the catagorical features
from sklearn import preprocessing
le = preprocessing.LabelEncoder()

# Encode labels in column 'species'.
for i in list_cat:
    data[i]= le.fit_transform(data[i])
```

In [45]:
```
data['SALES'] = data['SALES'].astype(int)
```

In [46]:
```
## taget feature are Sales and productline
X = data[['SALES','PRODUCTCODE']]
```

In [47]:
```
from yellowbrick.cluster import KElbowVisualizer
model = KMeans()
visualizer = KElbowVisualizer(model, k=(1,12)).fit(X)
visualizer.show()
```



Out[47]:
```
<Axes: title={'center': 'Distortion Score Elbow for KMeans Clustering'}, xlab
el='k', ylabel='distortion score'>
```

In [48]:
```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=4, init='k-means++', random_state=0).fit(X)
```

In [49]:
```
kmeans.labels_
```

Out[49]:
```
array([3, 3, 3, ..., 1, 0, 3])
```

In [50]:
```
kmeans.inertia_
```

Out[50]:
```
1042124306.2124939
```

In [51]:
```
kmeans.n_iter_
```

Out[51]:
```
4
```

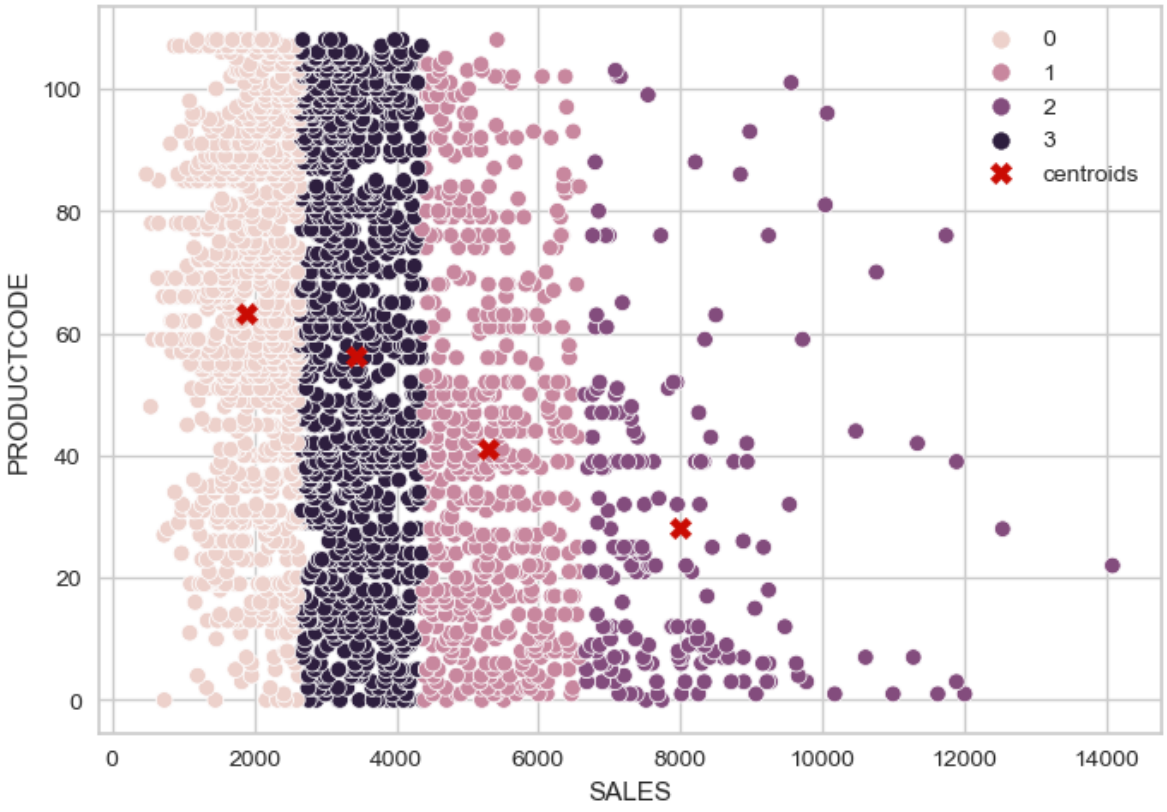
In [52]:
```
kmeans.cluster_centers_
```

Out[52]:
```
array([[1882.98554913,   63.28420039],
       [5295.90973451,   40.97522124],
       [7983.1758794 ,   28.05025126],
       [3424.0244858 ,   56.19980411]])
```

In [53]:
```
#getting the size of the clusters
from collections import Counter
Counter(kmeans.labels_)
```

Out[53]:
```
Counter({0: 1038, 3: 1023, 1: 563, 2: 199})
```

In [54]:
```
sns.scatterplot(data=X, x="SALES", y="PRODUCTCODE", hue=kmeans.labels_)
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1],
            marker="X", c="r", s=80, label="centroids")
plt.legend()
plt.show()
```



In [ ]:
```

```

In [ ]: