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

In [9]:
```
data = pd.read_csv("D:\ML\sales_data_sample.csv", encoding='latin1')
```

In [10]:
```
data.head()
```

Out[10]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	STATUS	QTR_ID	MONTH_ID	YEAR_ID	...
0	10107	30	95.70	2	2871.00	2/24/2003 0:00	Shipped	1	2	2003	...
1	10121	34	81.35	5	2765.90	5/7/2003 0:00	Shipped	2	5	2003	...
2	10134	41	94.74	2	3884.34	7/1/2003 0:00	Shipped	3	7	2003	...
3	10145	45	83.26	6	3746.70	8/25/2003 0:00	Shipped	3	8	2003	...
4	10159	49	100.00	14	5205.27	10/10/2003 0:00	Shipped	4	10	2003	...

5 rows × 25 columns

In [12]:
```
data.shape
```

Out[12]: (2823, 25)

In [13]:
```
data.isnull().sum
```

Out[13]:
```
<bound method NDFrame._add_numeric_operations.<locals>.sum of       ORDERNUMBER  QUANTITYORDERED  PRICEEACH  ORDERLI
NENUMBER  SALES  \
0           False            False      False            False  False
1           False            False      False            False  False
2           False            False      False            False  False
3           False            False      False            False  False
4           False            False      False            False  False
...           ...              ...        ...              ...    ...
2818        False            False      False            False  False
2819        False            False      False            False  False
2820        False            False      False            False  False
2821        False            False      False            False  False
2822        False            False      False            False  False

      ORDERDATE  STATUS  QTR_ID  MONTH_ID  YEAR_ID  ...  ADDRESSLINE1  \
0         False   False   False     False    False  ...         False
1         False   False   False     False    False  ...         False
2         False   False   False     False    False  ...         False
3         False   False   False     False    False  ...         False
4         False   False   False     False    False  ...         False
...         ...     ...     ...       ...      ...  ...           ...
2818      False   False   False     False    False  ...         False
2819      False   False   False     False    False  ...         False
2820      False   False   False     False    False  ...         False
2821      False   False   False     False    False  ...         False
2822      False   False   False     False    False  ...         False

      ADDRESSLINE2   CITY  STATE  POSTALCODE  COUNTRY  TERRITORY  \
0             True  False  False       False    False       True
1             True  False   True       False    False      False
2             True  False   True       False    False      False
3             True  False  False       False    False       True
4             True  False  False        True    False       True
...            ...    ...    ...         ...      ...        ...
2818          True  False   True       False    False      False
2819          True  False   True       False    False      False
2820          True  False   True       False    False      False
2821          True  False   True       False    False      False
2822          True  False  False       False    False       True

      CONTACTLASTNAME  CONTACTFIRSTNAME  DEALSIZE
0               False             False     False
1               False             False     False
2               False             False     False
3               False             False     False
4               False             False     False
...               ...               ...       ...
2818            False             False     False
2819            False             False     False
2820            False             False     False
2821            False             False     False
2822            False             False     False

[2823 rows x 25 columns]>
```

In [14]:
```
data.drop(["ORDERNUMBER", "PRICEEACH", "ORDERDATE", "PHONE", "ADDRESSLINE1", "ADDRESSLINE2"], axis=1, inplace=True)
```

In [15]:
```
data.head()
```

Out[15]:

	QUANTITYORDERED	ORDERLINENUMBER	SALES	STATUS	QTR_ID	MONTH_ID	YEAR_ID	PRODUCTLINE	MSRP	PRODUCTCODE	CUSTO
0	30	2	2871.00	Shipped	1	2	2003	Motorcycles	95	S10_1678	Land
1	34	5	2765.90	Shipped	2	5	2003	Motorcycles	95	S10_1678	Reims (
2	41	2	3884.34	Shipped	3	7	2003	Motorcycles	95	S10_1678	Lyon
3	45	6	3746.70	Shipped	3	8	2003	Motorcycles	95	S10_1678	Toys4Gro
4	49	14	5205.27	Shipped	4	10	2003	Motorcycles	95	S10_1678	Corporat

In [16]:
```
data.isnull().sum()
```

Out[16]:
```
QUANTITYORDERED        0
ORDERLINENUMBER        0
SALES                  0
STATUS                 0
QTR_ID                 0
MONTH_ID               0
YEAR_ID                0
PRODUCTLINE            0
MSRP                   0
PRODUCTCODE            0
CUSTOMERNAME           0
CITY                   0
STATE               1486
POSTALCODE            76
COUNTRY                0
TERRITORY           1074
CONTACTLASTNAME        0
CONTACTFIRSTNAME       0
DEALSIZE               0
dtype: int64
```

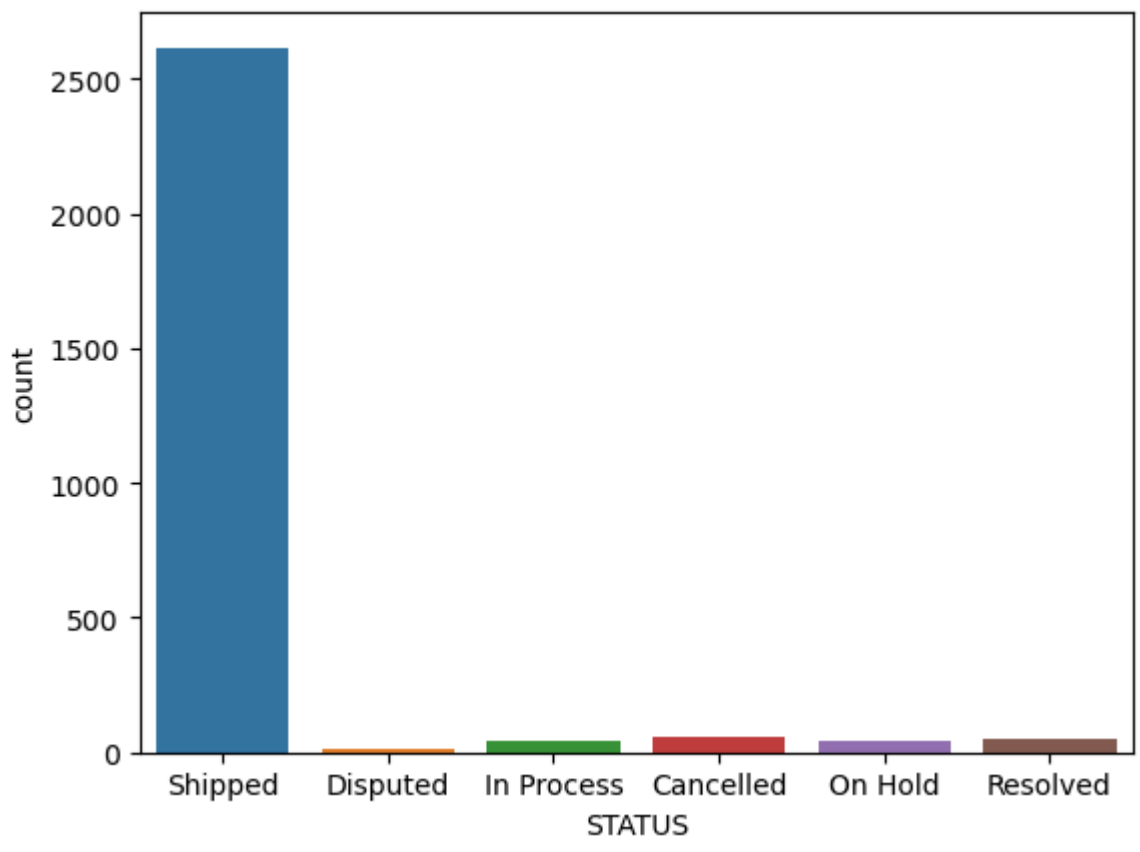
In [17]:
```
data.describe()
```

Out[17]:

	QUANTITYORDERED	ORDERLINENUMBER	SALES	QTR_ID	MONTH_ID	YEAR_ID	MSRP
count	2823.000000	2823.000000	2823.000000	2823.000000	2823.000000	2823.00000	2823.000000
mean	35.092809	6.466171	3553.889072	2.717676	7.092455	2003.81509	100.715551
std	9.741443	4.225841	1841.865106	1.203878	3.656633	0.69967	40.187912
min	6.000000	1.000000	482.130000	1.000000	1.000000	2003.00000	33.000000
25%	27.000000	3.000000	2203.430000	2.000000	4.000000	2003.00000	68.000000
50%	35.000000	6.000000	3184.800000	3.000000	8.000000	2004.00000	99.000000
75%	43.000000	9.000000	4508.000000	4.000000	11.000000	2004.00000	124.000000
max	97.000000	18.000000	14082.800000	4.000000	12.000000	2005.00000	214.000000

In [18]:
```
sns.countplot(data = data , x = 'STATUS')
```

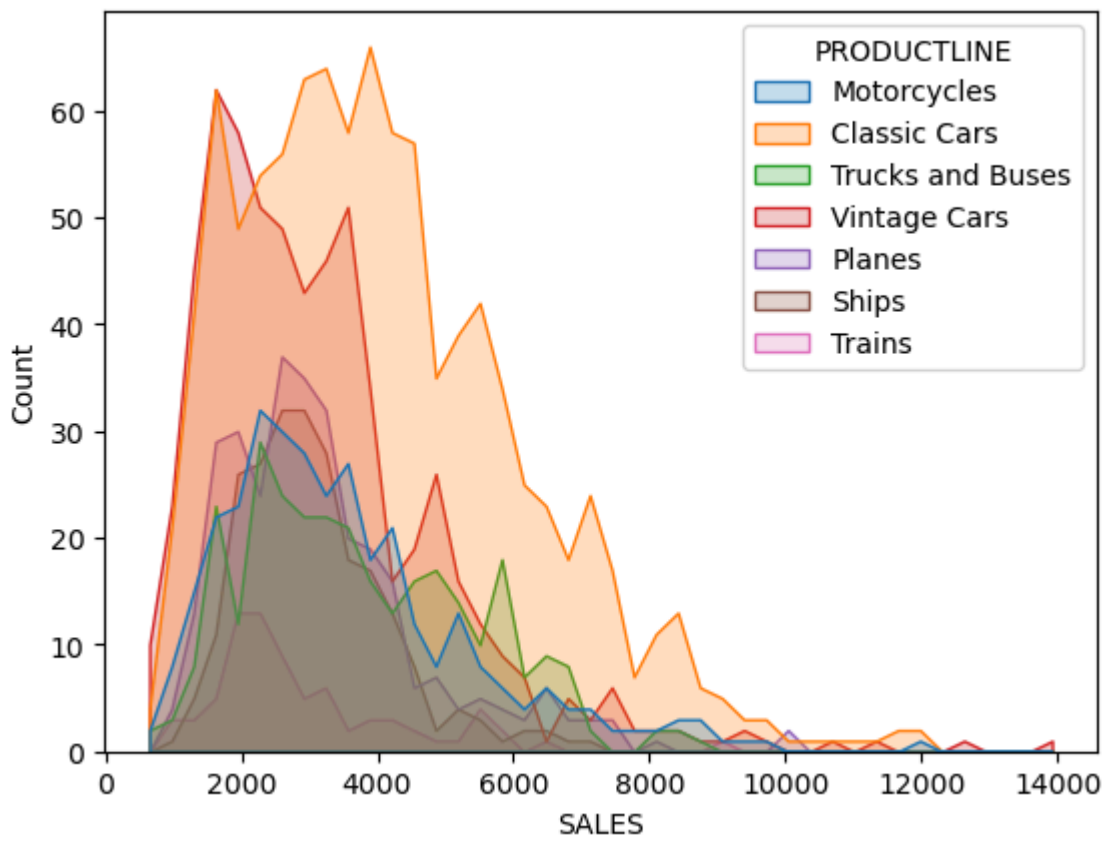
Out[18]:
```
<Axes: xlabel='STATUS', ylabel='count'>
```



In [19]:
```
import seaborn as sns
```

In [20]:
```
sns.histplot(x = 'SALES' , hue = 'PRODUCTLINE', data = data, element="poly")
```

Out[20]: <Axes: xlabel='SALES', ylabel='Count'>



In [21]:
```
data['PRODUCTLINE'].unique()
```

Out[21]:
```
array(['Motorcycles', 'Classic Cars', 'Trucks and Buses', 'Vintage Cars',
       'Planes', 'Ships', 'Trains'], dtype=object)
```

In [22]:
```
data.drop_duplicates(inplace=True)
```

In [23]:
```
data.info()
```

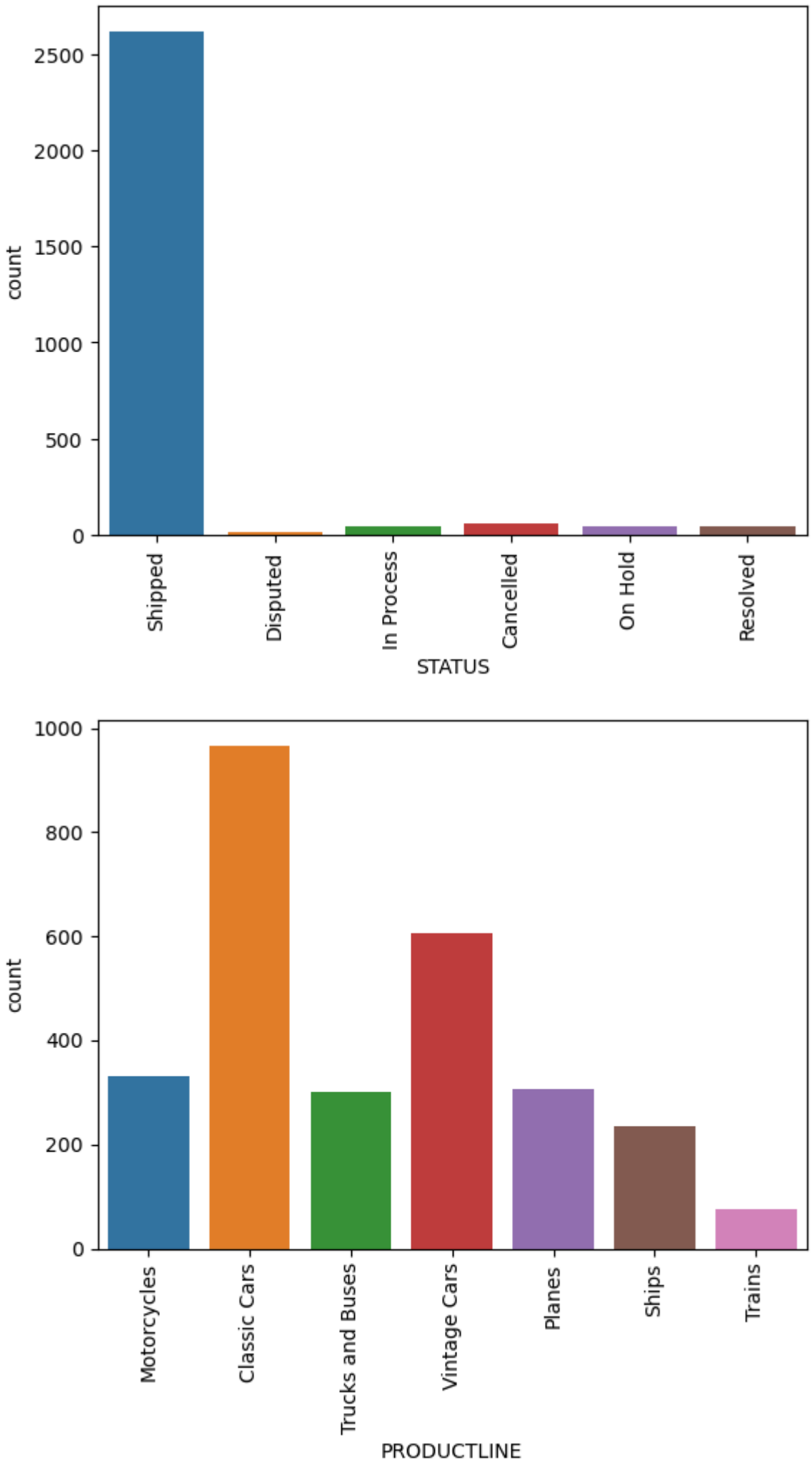
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2823 entries, 0 to 2822
Data columns (total 19 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   QUANTITYORDERED   2823 non-null   int64
 1   ORDERLINENUMBER   2823 non-null   int64
 2   SALES             2823 non-null   float64
 3   STATUS            2823 non-null   object
 4   QTR_ID            2823 non-null   int64
 5   MONTH_ID          2823 non-null   int64
 6   YEAR_ID           2823 non-null   int64
 7   PRODUCTLINE       2823 non-null   object
 8   MSRP              2823 non-null   int64
 9   PRODUCTCODE       2823 non-null   object
 10  CUSTOMERNAME      2823 non-null   object
 11  CITY              2823 non-null   object
 12  STATE             1337 non-null   object
 13  POSTALCODE        2747 non-null   object
 14  COUNTRY           2823 non-null   object
 15  TERRITORY         1749 non-null   object
 16  CONTACTLASTNAME   2823 non-null   object
 17  CONTACTFIRSTNAME  2823 non-null   object
 18  DEALSIZE          2823 non-null   object
dtypes: float64(1), int64(6), object(12)
memory usage: 419.2+ KB
```

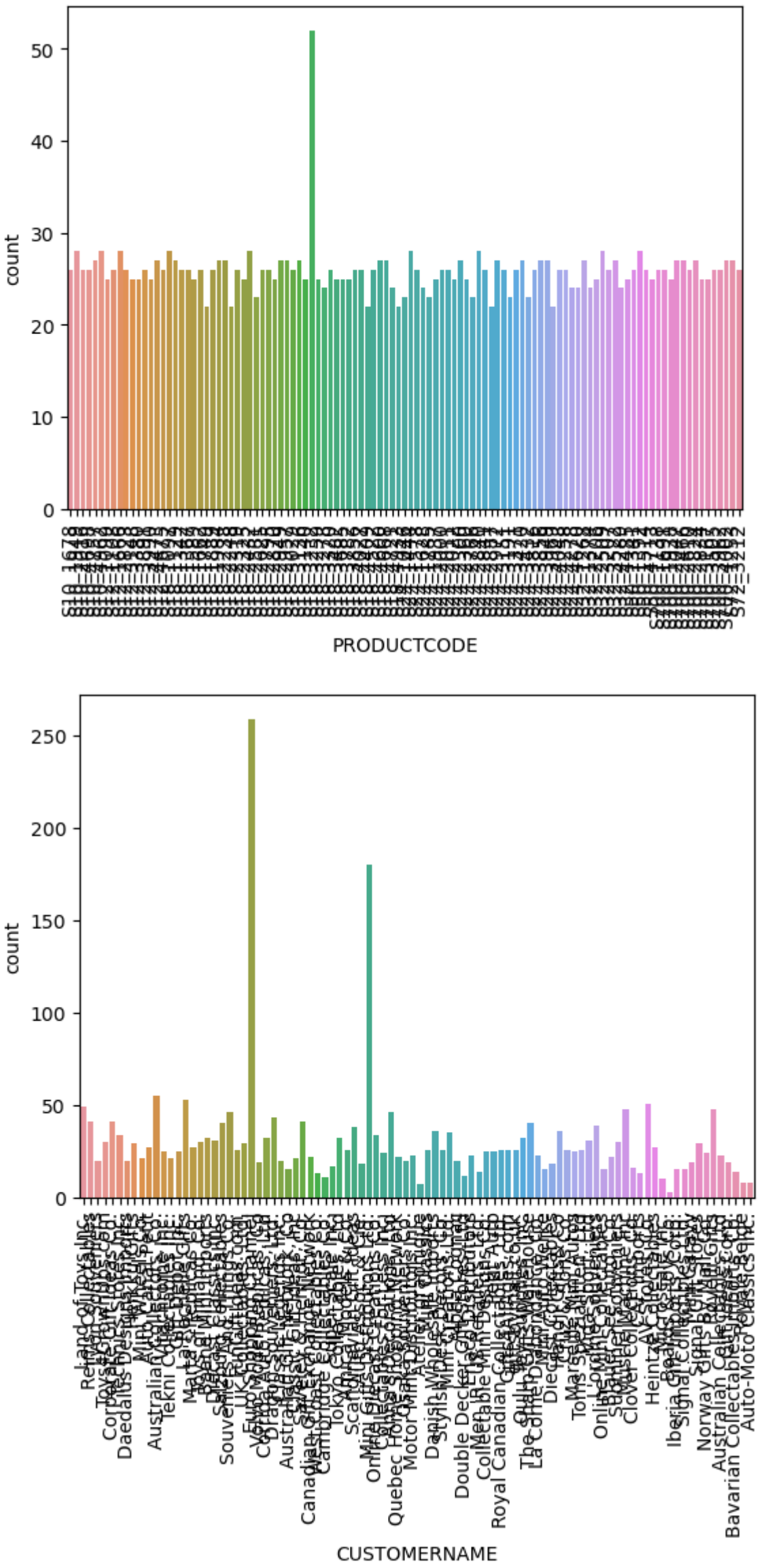
In [24]:
```
list_cat = data.select_dtypes(include=['object']).columns.tolist()
```

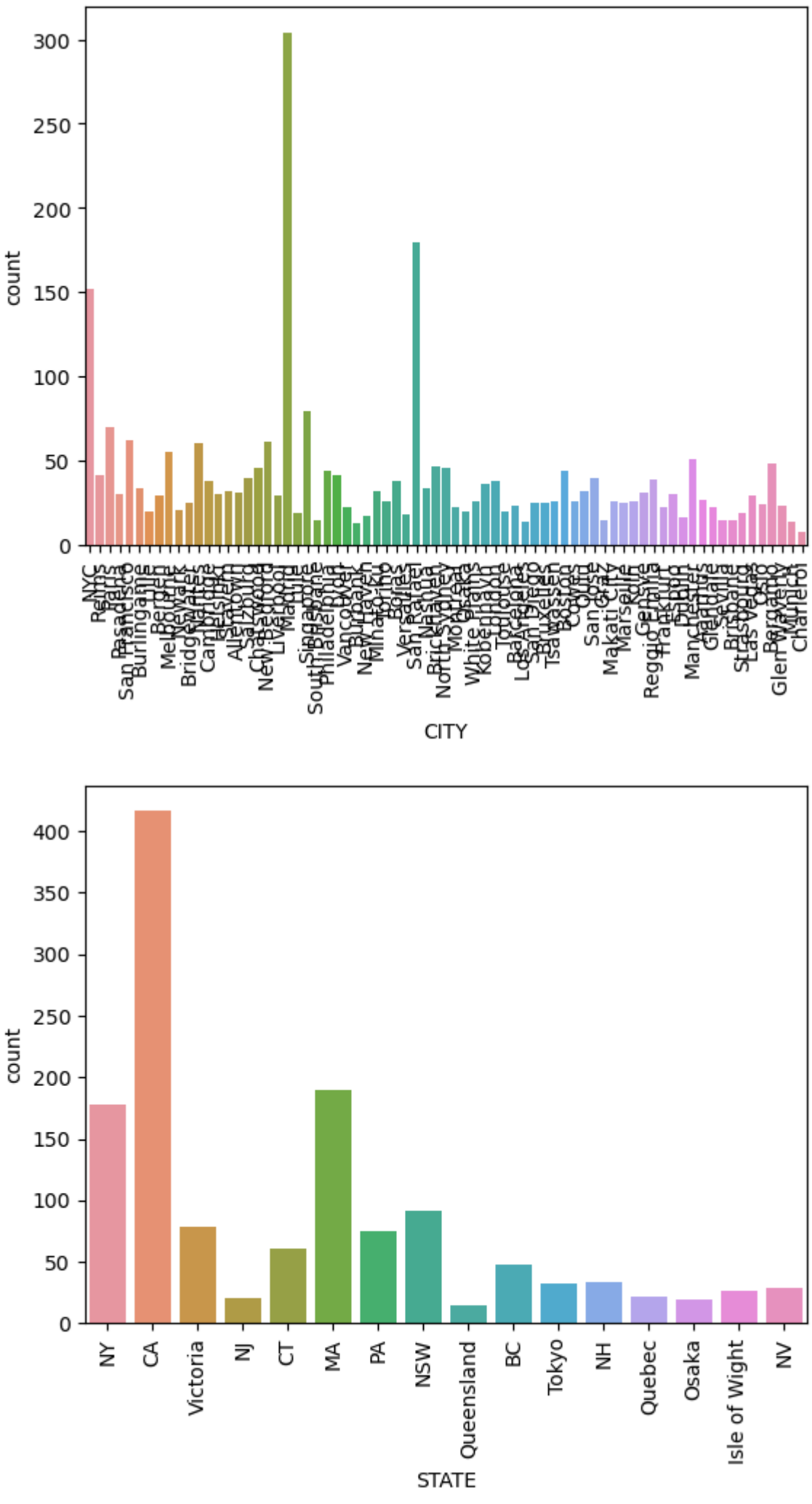
In [25]:
```
list_cat
```

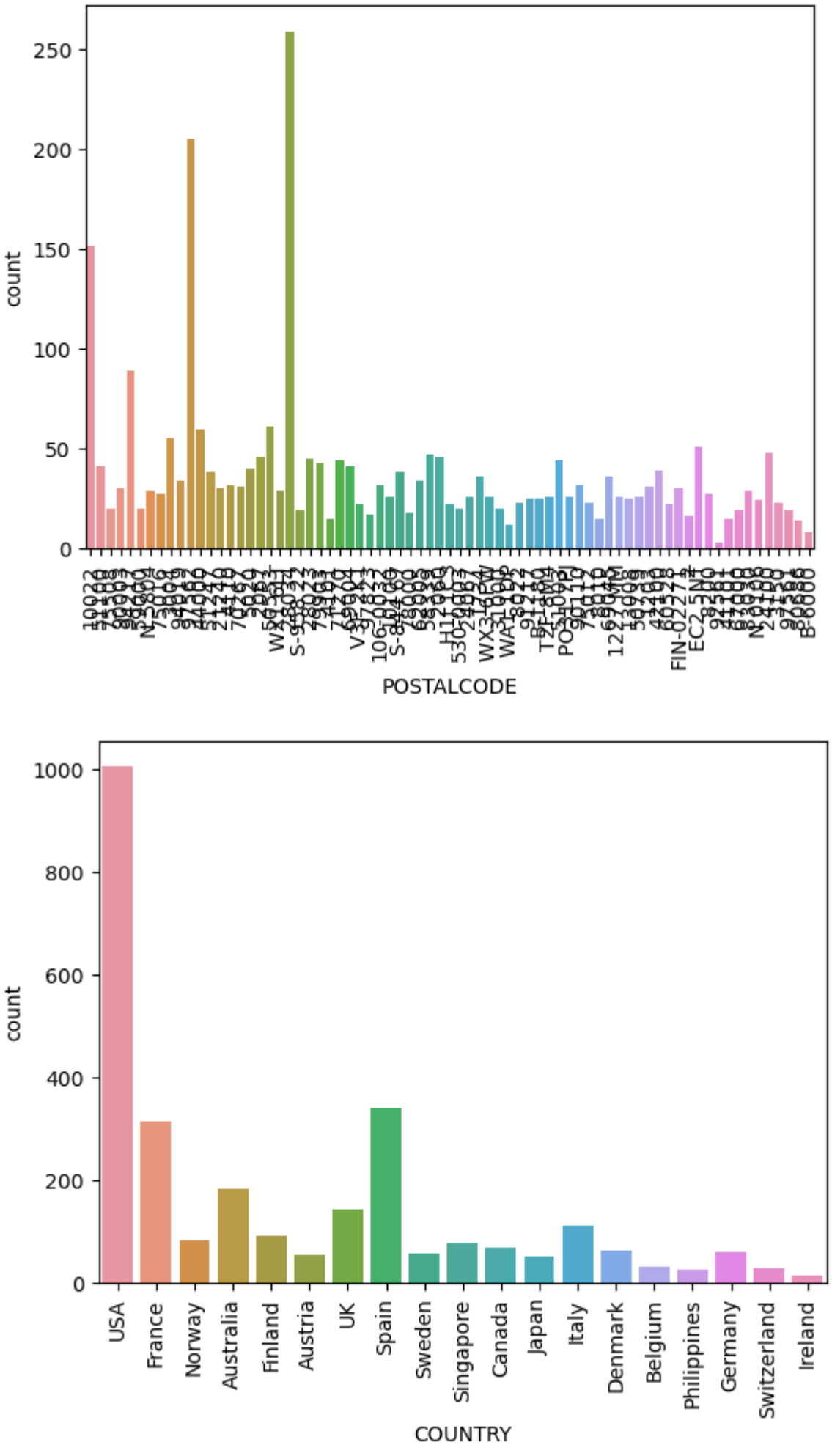
Out[25]:
```
['STATUS',
 'PRODUCTLINE',
 'PRODUCTCODE',
 'CUSTOMERNAME',
 'CITY',
 'STATE',
 'POSTALCODE',
 'COUNTRY',
 'TERRITORY',
 'CONTACTLASTNAME',
 'CONTACTFIRSTNAME',
 'DEALSIZE']
```

In [27]:
```
for i in list_cat:
    sns.countplot(data = data ,x = i)
    plt.xticks(rotation = 90)
    plt.show()
```

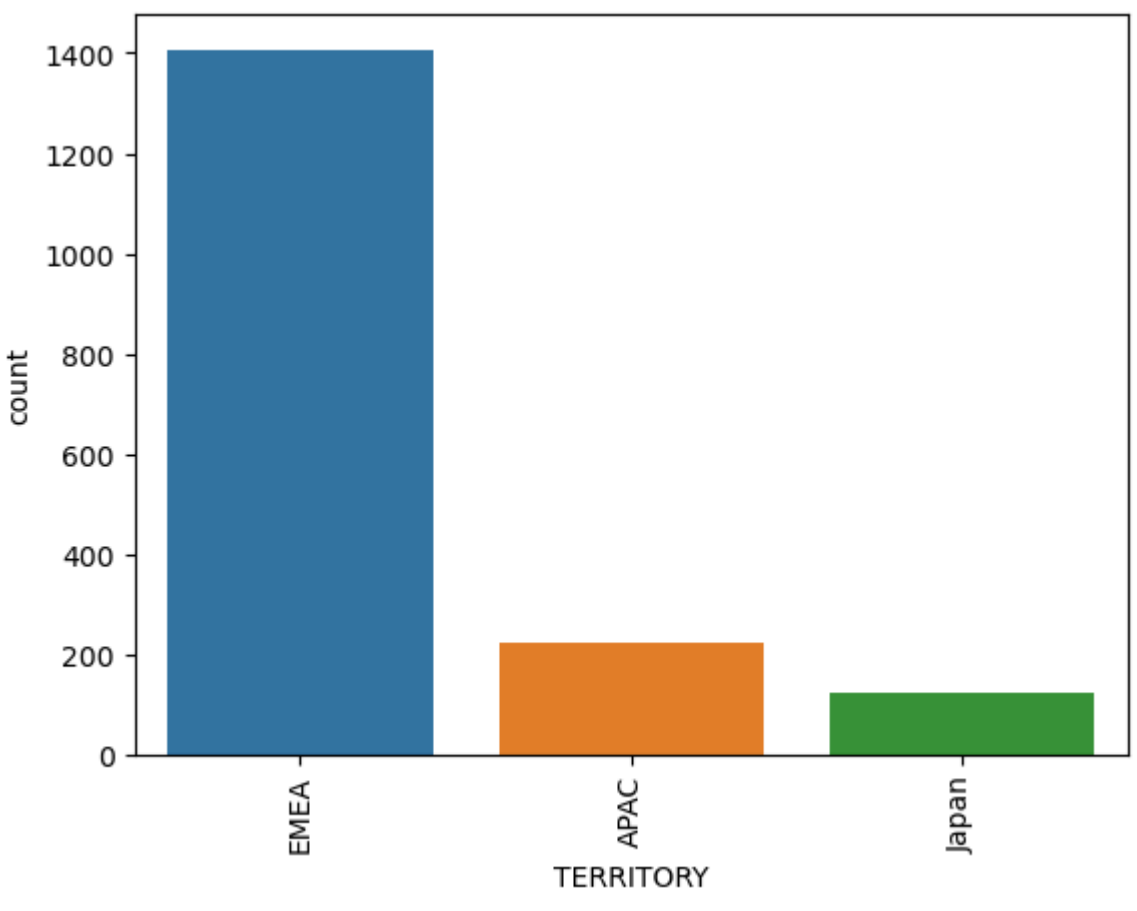


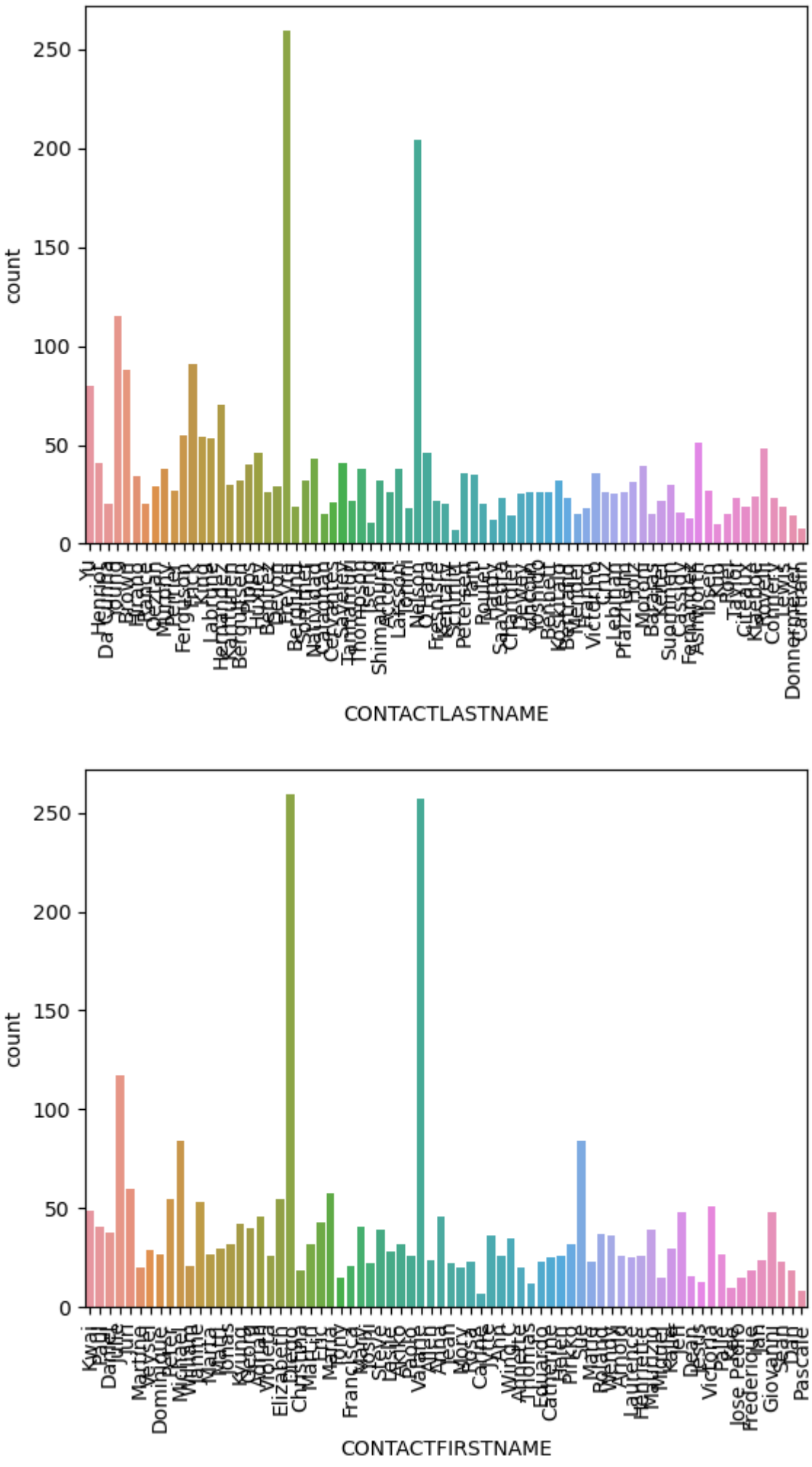

count
0
10
20
30
40
50
PRODUCTCODE
count
0
50
100
150
200
250
CUSTOMERNAME

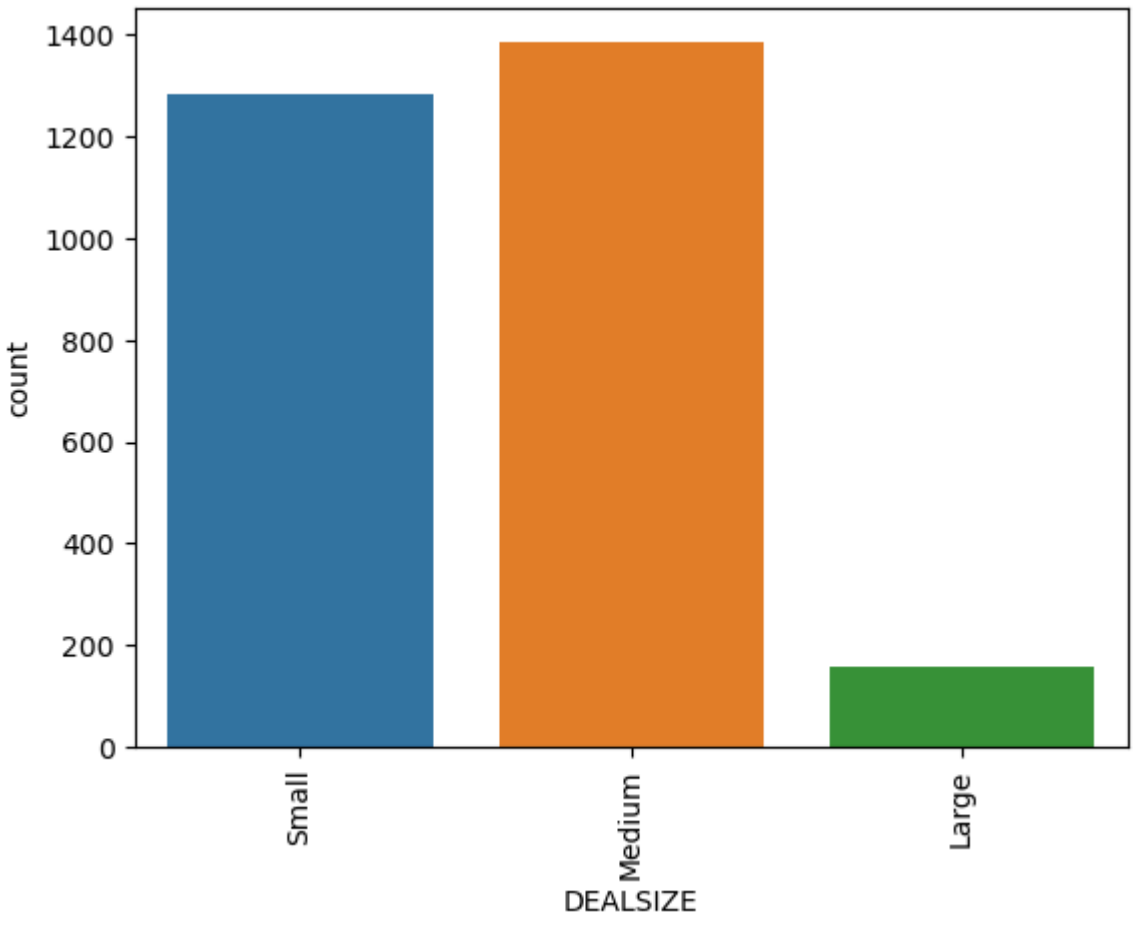

count
CITY
count
STATE


count
POSTALCODE

count
USA
France
Norway
Australia
Finland
Austria
UK
Spain
Sweden
Singapore
Canada
Japan
Italy
Denmark
Belgium
Philippines
Germany
Switzerland
Ireland
COUNTRY


count
EMEA
APAC
Japan
TERRITORY


count
CONTACTLASTNAME
count
CONTACTFIRSTNAME



In [29]:
```
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 [30]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2823 entries, 0 to 2822
Data columns (total 19 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   QUANTITYORDERED   2823 non-null   int64
 1   ORDERLINENUMBER   2823 non-null   int64
 2   SALES             2823 non-null   float64
 3   STATUS            2823 non-null   int32
 4   QTR_ID            2823 non-null   int64
 5   MONTH_ID          2823 non-null   int64
 6   YEAR_ID           2823 non-null   int64
 7   PRODUCTLINE       2823 non-null   int32
 8   MSRP              2823 non-null   int64
 9   PRODUCTCODE       2823 non-null   int32
 10  CUSTOMERNAME      2823 non-null   int32
 11  CITY              2823 non-null   int32
 12  STATE             2823 non-null   int32
 13  POSTALCODE        2823 non-null   int32
 14  COUNTRY           2823 non-null   int32
 15  TERRITORY         2823 non-null   int32
 16  CONTACTLASTNAME   2823 non-null   int32
 17  CONTACTFIRSTNAME  2823 non-null   int32
 18  DEALSIZE          2823 non-null   int32
dtypes: float64(1), int32(12), int64(6)
memory usage: 286.8 KB
```

In [31]:
```
data['SALES'] = data['SALES'].astype(int)
```

In [32]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2823 entries, 0 to 2822
Data columns (total 19 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   QUANTITYORDERED   2823 non-null   int64
 1   ORDERLINENUMBER   2823 non-null   int64
 2   SALES             2823 non-null   int32
 3   STATUS            2823 non-null   int32
 4   QTR_ID            2823 non-null   int64
 5   MONTH_ID          2823 non-null   int64
 6   YEAR_ID           2823 non-null   int64
 7   PRODUCTLINE       2823 non-null   int32
 8   MSRP              2823 non-null   int64
 9   PRODUCTCODE       2823 non-null   int32
 10  CUSTOMERNAME      2823 non-null   int32
 11  CITY              2823 non-null   int32
 12  STATE             2823 non-null   int32
 13  POSTALCODE        2823 non-null   int32
 14  COUNTRY           2823 non-null   int32
 15  TERRITORY         2823 non-null   int32
 16  CONTACTLASTNAME   2823 non-null   int32
 17  CONTACTFIRSTNAME  2823 non-null   int32
 18  DEALSIZE          2823 non-null   int32
dtypes: int32(13), int64(6)
memory usage: 275.8 KB
```

In [33]:
```
data.describe()
```

Out[33]:

	QUANTITYORDERED	ORDERLINENUMBER	SALES	STATUS	QTR_ID	MONTH_ID	YEAR_ID	PRODUCTLINE	MSRP
count	2823.000000	2823.000000	2823.000000	2823.000000	2823.000000	2823.000000	2823.00000	2823.000000	2823.000000
mean	35.092809	6.466171	3553.421537	4.782501	2.717676	7.092455	2003.81509	2.515055	100.715551
std	9.741443	4.225841	1841.865754	0.879416	1.203878	3.656633	0.69967	2.411665	40.187912
min	6.000000	1.000000	482.000000	0.000000	1.000000	1.000000	2003.00000	0.000000	33.000000
25%	27.000000	3.000000	2203.000000	5.000000	2.000000	4.000000	2003.00000	0.000000	68.000000
50%	35.000000	6.000000	3184.000000	5.000000	3.000000	8.000000	2004.00000	2.000000	99.000000
75%	43.000000	9.000000	4508.000000	5.000000	4.000000	11.000000	2004.00000	5.000000	124.000000
max	97.000000	18.000000	14082.000000	5.000000	4.000000	12.000000	2005.00000	6.000000	214.000000

In [34]:
```
X = data[['SALES','PRODUCTCODE']]
```

In [35]:
```
data.columns
```

Out[35]:
```
Index(['QUANTITYORDERED', 'ORDERLINENUMBER', 'SALES', 'STATUS', 'QTR_ID',
       'MONTH_ID', 'YEAR_ID', 'PRODUCTLINE', 'MSRP', 'PRODUCTCODE',
       'CUSTOMERNAME', 'CITY', 'STATE', 'POSTALCODE', 'COUNTRY', 'TERRITORY',
       'CONTACTLASTNAME', 'CONTACTFIRSTNAME', 'DEALSIZE'],
      dtype='object')
```

In [36]:
```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=4, init='k-means++', random_state=0).fit(X)
```

```
C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_in
it` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

In [37]:
```
kmeans.labels_
```

Out[37]:
```
array([3, 3, 3, ..., 1, 0, 3])
```

In [38]:
```
kmeans.inertia_
```

Out[38]:
```
1042124306.2124939
```

In [39]:
```
kmeans.n_iter_
```

Out[39]:
```
4
```

In [40]:
```
kmeans.cluster_centers_
```

Out[40]:
```
array([[1882.98554913,   63.28420039],
       [5295.90973451,   40.97522124],
       [7983.1758794 ,   28.05025126],
       [3424.0244858 ,   56.19980411]])
```

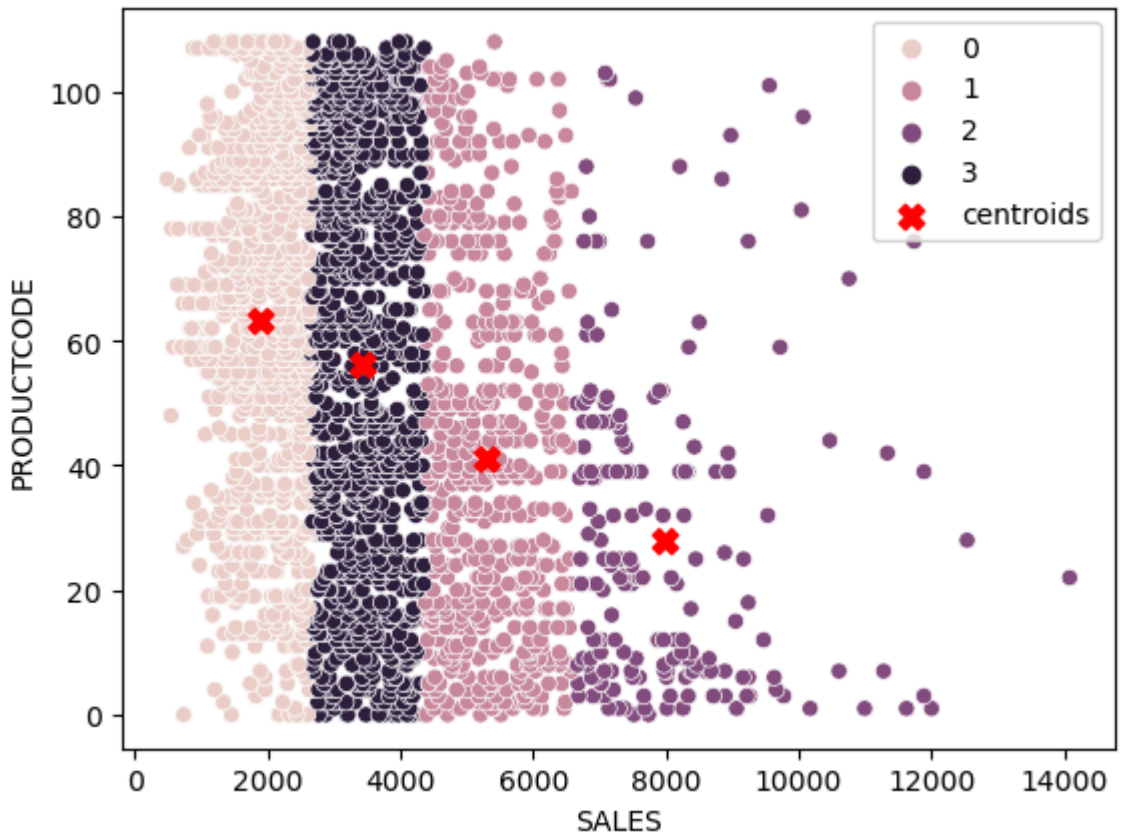
In [41]:
```
from collections import Counter
Counter(kmeans.labels_)
```

Out[41]:
```
Counter({0: 1038, 3: 1023, 1: 563, 2: 199})
```

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

```