In [1]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
```

In [2]:
```python
df=pd.read_csv('C:/Users/ambalkt/Downloads/Power BI Practice/Projects/Sales/Car_Sales.csv',sep=",", encoding='Latin-1')
df.head(5)
```

Out[2]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	STATUS	QTR_ID	MONTH_ID	YEAR_ID	...	ADDRESSLINE1	ADDRESSLINE2	CI
0	10107	30	95.70	2	2871.00	2/24/2003 0:00	Shipped	1	2	2003	...	897 Long Airport Avenue	NaN	N
1	10121	34	81.35	5	2765.90	5/7/2003 0:00	Shipped	2	5	2003	...	59 rue de l'Abbaye	NaN	Rei
2	10134	41	94.74	2	3884.34	7/1/2003 0:00	Shipped	3	7	2003	...	27 rue du Colonel Pierre Avia	NaN	Pa
3	10145	45	83.26	6	3746.70	8/25/2003 0:00	Shipped	3	8	2003	...	78934 Hillside Dr.	NaN	Pasade
4	10159	49	100.00	14	5205.27	10/10/2003 0:00	Shipped	4	10	2003	...	7734 Strong St.	NaN	S Francis

5 rows × 25 columns

In [3]:
```python
df.describe()
```

Out[3]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	QTR_ID	MONTH_ID	YEAR_ID	MSRP
count	2823.000000	2823.000000	2823.000000	2823.000000	2823.000000	2823.000000	2823.000000	2823.00000	2823.000000
mean	10258.725115	35.092809	83.658544	6.466171	3553.889072	2.717676	7.092455	2003.81509	100.715551
std	92.085478	9.741443	20.174277	4.225841	1841.865106	1.203878	3.656633	0.69967	40.187912
min	10100.000000	6.000000	26.880000	1.000000	482.130000	1.000000	1.000000	2003.00000	33.000000
25%	10180.000000	27.000000	68.860000	3.000000	2203.430000	2.000000	4.000000	2003.00000	68.000000
50%	10262.000000	35.000000	95.700000	6.000000	3184.800000	3.000000	8.000000	2004.00000	99.000000
75%	10333.500000	43.000000	100.000000	9.000000	4508.000000	4.000000	11.000000	2004.00000	124.000000
max	10425.000000	97.000000	100.000000	18.000000	14082.800000	4.000000	12.000000	2005.00000	214.000000

In [4]:
```python
df.shape
```

Out[4]:
```
(2823, 25)
```

In [5]:
```python
df.isnull().sum()
```

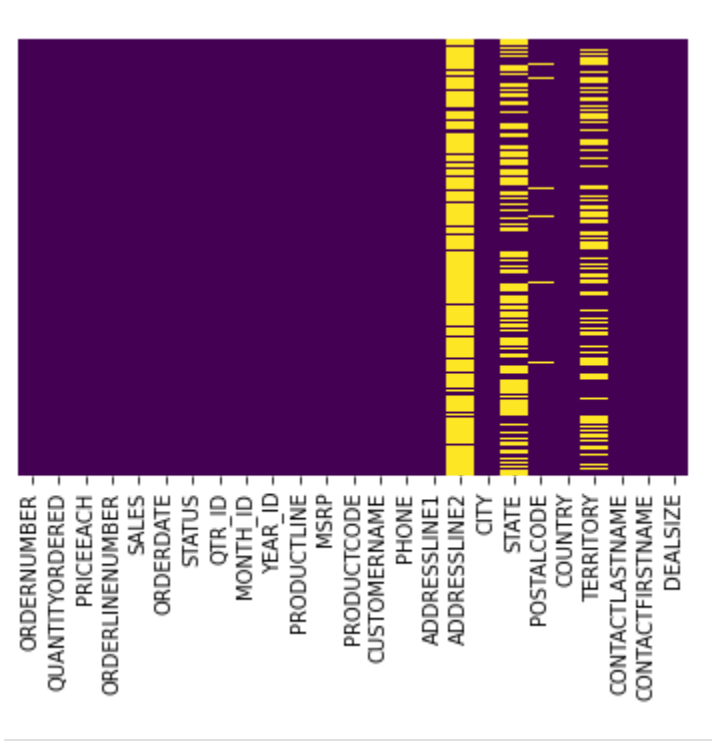
Out[5]:
```
ORDERNUMBER            0
QUANTITYORDERED        0
PRICEEACH              0
ORDERLINENUMBER        0
SALES                  0
ORDERDATE              0
STATUS                 0
QTR_ID                 0
MONTH_ID               0
YEAR_ID                0
PRODUCTLINE            0
MSRP                   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 [6]:
```python
for col in df.columns:
    print(f'Number of {col} unique values: {df[col].nunique()}')
```

```
Number of ORDERNUMBER unique values: 307
Number of QUANTITYORDERED unique values: 58
Number of PRICEEACH unique values: 1016
Number of ORDERLINENUMBER unique values: 18
Number of SALES unique values: 2763
Number of ORDERDATE unique values: 252
Number of STATUS unique values: 6
Number of QTR_ID unique values: 4
Number of MONTH_ID unique values: 12
Number of YEAR_ID unique values: 3
Number of PRODUCTLINE unique values: 7
Number of MSRP unique values: 80
Number of PRODUCTCODE unique values: 109
Number of CUSTOMERNAME unique values: 92
Number of PHONE unique values: 91
Number of ADDRESSLINE1 unique values: 92
Number of ADDRESSLINE2 unique values: 9
Number of CITY unique values: 73
Number of STATE unique values: 16
Number of POSTALCODE unique values: 73
Number of COUNTRY unique values: 19
Number of TERRITORY unique values: 3
Number of CONTACTLASTNAME unique values: 77
Number of CONTACTFIRSTNAME unique values: 72
Number of DEALSIZE unique values: 3
```

In [7]:
```python
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

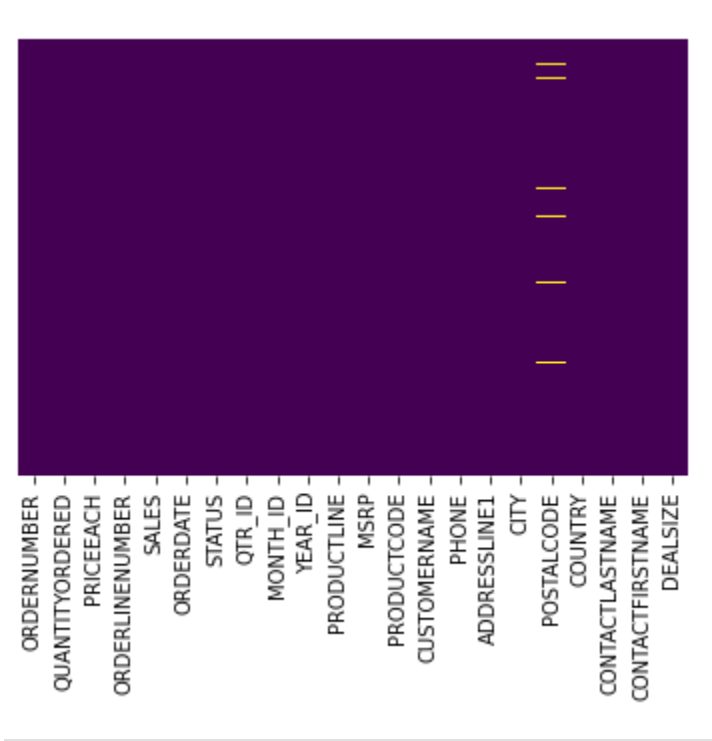
Out[7]: <AxesSubplot:>



In [8]:
```python
to_drop=['ADDRESSLINE2','TERRITORY','STATE']
df=df.drop(to_drop,axis=1)
```

In [9]:
```python
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[9]: <AxesSubplot:>



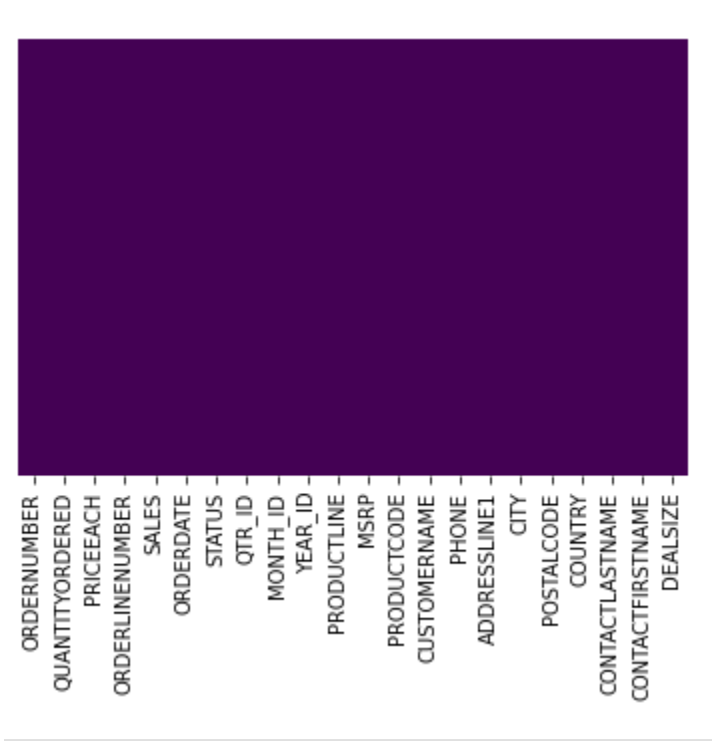
In [10]:
```python
df['POSTALCODE'][df.CITY=='San Francisco']=94016
```

```
C:\Users\ambalkt\AppData\Local\Temp\1/ipykernel_15340/814236309.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['POSTALCODE'][df.CITY=='San Francisco']=94016
```

In [11]:
```python
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[11]: <AxesSubplot:>



In [12]:
```python
df.dtypes
```

Out[12]:
```
ORDERNUMBER           int64
QUANTITYORDERED       int64
PRICEEACH           float64
ORDERLINENUMBER       int64
SALES               float64
ORDERDATE            object
STATUS               object
QTR_ID                int64
MONTH_ID              int64
YEAR_ID               int64
PRODUCTLINE          object
MSRP                  int64
PRODUCTCODE          object
CUSTOMERNAME         object
PHONE                object
ADDRESSLINE1         object
CITY                 object
POSTALCODE           object
COUNTRY              object
CONTACTLASTNAME      object
CONTACTFIRSTNAME     object
DEALSIZE             object
dtype: object
```

In [13]:
```python
min_thres=df['PRICEEACH'].quantile(0.05)
min_thres
```

Out[13]:
```
42.67
```

In [14]:
```python
df[df['PRICEEACH']<min_thres]
```

Out[14]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	STATUS	QTR_ID	MONTH_ID	YEAR_ID	...	PRODUCTCODE	CUSTOMERNAME
22	10375	21	34.91	12	733.11	2/3/2005 0:00	Shipped	1	2	2005	...	S10_1678	La Rochelle Gifts
390	10371	49	35.71	4	1749.79	1/23/2005 0:00	Shipped	1	1	2005	...	S12_4473	Mini Gifts Distributors Ltd.
440	10333	29	40.25	7	1167.25	11/18/2004 0:00	Shipped	4	11	2004	...	S18_1097	Mini Wheels Co.
495	10346	42	36.11	3	1516.62	11/29/2004 0:00	Shipped	4	11	2004	...	S18_1342	Signal Gift Stores
549	10378	34	42.64	5	1449.76	2/10/2005 0:00	Shipped	1	2	2005	...	S18_1589	Euro Shopping Channel
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2376	10401	56	35.35	7	1979.60	4/3/2005 0:00	On Hold	2	4	2005	...	S50_1341	Tekni Collectables Inc.
2772	10131	21	41.71	7	875.91	6/16/2003 0:00	Shipped	2	6	2003	...	S72_1253	Gift Ideas Corp.
2775	10167	40	41.71	4	1668.40	10/23/2003 0:00	Cancelled	4	10	2003	...	S72_1253	Scandinavian Gift Ideas
2778	10197	29	41.71	1	1209.59	11/26/2003 0:00	Shipped	4	11	2003	...	S72_1253	Enaco Distributors
2794	10398	34	40.22	1	1367.48	3/30/2005 0:00	Shipped	1	3	2005	...	S72_1253	Reims Collectables

139 rows × 22 columns

In [15]:
```python
max_thres=df["PRICEEACH"].quantile(0.95)
max_thres
```

Out[15]:
```
100.0
```

In [16]:
```python
df[df['PRICEEACH']>max_thres]
```

Out[16]:

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	STATUS	QTR_ID	MONTH_ID	YEAR_ID	...	PRODUCTCODE	CUSTOMERNAME	PHONI

0 rows × 22 columns

In [17]:
```python
min_thres_sales=df['SALES'].quantile(0.05)
min_thres_sales
```

Out[17]:
```
1268.7569999999998
```

In [18]:
```python
df[df['SALES']<min_thres_sales]
```

Out[18]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	STATUS	QTR_ID	MONTH_ID	YEAR_ID	...	PRODUCTCODE	CUSTOMERNAME
22	10375	21	34.91	12	733.11	2/3/2005 0:00	Shipped	1	2	2005	...	S10_1678	La Rochelle Gifts
125	10337	25	48.05	8	1201.25	11/21/2004 0:00	Shipped	4	11	2004	...	S10_4757	Classic Legends Inc.
315	10419	10	100.00	11	1092.20	5/17/2005 0:00	Shipped	2	5	2005	...	S12_3380	Salzburg Collectables
440	10333	29	40.25	7	1167.25	11/18/2004 0:00	Shipped	4	11	2004	...	S18_1097	Mini Wheels Co.
513	10244	20	58.22	6	1164.40	4/29/2004 0:00	Shipped	2	4	2004	...	S18_1367	Euro Shopping Channel
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2778	10197	29	41.71	1	1209.59	11/26/2003 0:00	Shipped	4	11	2003	...	S72_1253	Enaco Distributors
2782	10262	21	57.11	12	1199.31	6/24/2004 0:00	Cancelled	2	6	2004	...	S72_1253	Euro Shopping Channel
2785	10296	21	45.19	10	948.99	9/15/2004 0:00	Shipped	3	9	2004	...	S72_1253	Bavarian Collectables Imports, Co.
2795	10400	20	56.12	4	1122.40	4/1/2005 0:00	Shipped	2	4	2005	...	S72_1253	The Sharp Gifts Warehouse
2808	10232	24	49.69	3	1192.56	3/20/2004 0:00	Shipped	1	3	2004	...	S72_3212	giftsbymail.co.uk

142 rows × 22 columns

In [19]:
```python
max_thres_sales=df['SALES'].quantile(0.95)
max_thres_sales
```

Out[19]:
```
7108.120000000003
```

In [20]:
```python
df[df['SALES']>max_thres_sales]
```

Out[20]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	STATUS	QTR_ID	MONTH_ID	YEAR_ID	...	PRODUCTCODE	CUSTOMERNAME
20	10341	41	100.00	9	7737.93	11/24/2004 0:00	Shipped	4	11	2004	...	S10_1678	Salzburg Collectables
25	10417	66	100.00	2	7516.08	5/13/2005 0:00	Disputed	2	5	2005	...	S10_1678	Euro Shopping Channel
27	10112	29	100.00	1	7209.11	3/24/2003 0:00	Shipped	1	3	2003	...	S10_1949	Volvo Model Replicas, Co
28	10126	38	100.00	11	7329.06	5/28/2003 0:00	Shipped	2	5	2003	...	S10_1949	Corrida Auto Replicas, Ltd
29	10140	37	100.00	11	7374.10	7/24/2003 0:00	Shipped	3	7	2003	...	S10_1949	Technics Stores Inc.
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2505	10388	46	100.00	2	10066.60	3/3/2005 0:00	Shipped	1	3	2005	...	S700_1691	FunGiftIdeas.com
2586	10401	85	88.75	10	7543.75	4/3/2005 0:00	On Hold	2	4	2005	...	S700_2466	Tekni Collectables Inc.
2634	10336	46	100.00	2	9558.80	11/20/2004 0:00	Shipped	4	11	2004	...	S700_2824	La Corne D'abondance, Co.
2656	10318	50	100.00	8	7119.00	11/2/2004 0:00	Shipped	4	11	2004	...	S700_2834	Diecast Classics Inc.
2662	10388	50	100.00	3	7154.50	3/3/2005 0:00	Shipped	1	3	2005	...	S700_2834	FunGiftIdeas.com

142 rows × 22 columns

In [21]:
```python
min_thres_msrp=df['MSRP'].quantile(0.05)
min_thres_msrp
```

Out[21]:
```
43.0
```

In [22]:
```python
df[df['MSRP']<min_thres_msrp]
```

Out[22]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	STATUS	QTR_ID	MONTH_ID	YEAR_ID	...	PRODUCTCODE	CUSTOMERNAME
1539	10101	45	31.20	3	1404.00	1/9/2003 0:00	Shipped	1	1	2003	...	S24_1937	Blauer See Auto, Co.
1540	10110	20	35.51	3	710.20	3/18/2003 0:00	Shipped	1	3	2003	...	S24_1937	AV Stores, Co.
1541	10124	45	37.84	2	1702.80	5/21/2003 0:00	Shipped	2	5	2003	...	S24_1937	Signal Gift Stores
1542	10149	36	33.19	7	1194.84	9/12/2003 0:00	Shipped	3	9	2003	...	S24_1937	Signal Collectibles Ltd.
1543	10162	37	27.22	5	1007.14	10/18/2003 0:00	Shipped	4	10	2003	...	S24_1937	Corporate Gift Ideas Co.
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2217	10355	38	39.83	10	1513.54	12/7/2004 0:00	Shipped	4	12	2004	...	S32_2206	Euro Shopping Channel
2218	10364	48	48.28	1	2317.44	1/6/2005 0:00	Shipped	1	1	2005	...	S32_2206	Marseille Mini Autos
2219	10378	40	82.46	1	3298.40	2/10/2005 0:00	Shipped	1	2	2005	...	S32_2206	Euro Shopping Channel
2220	10390	41	44.56	11	1826.96	3/4/2005 0:00	Shipped	1	3	2005	...	S32_2206	Mini Gifts Distributors Ltd.
2221	10403	30	40.23	2	1206.90	4/8/2005 0:00	Shipped	2	4	2005	...	S32_2206	UK Collectables, Ltd.

127 rows × 22 columns

In [23]:
```python
max_thres_msrp=df['MSRP'].quantile(0.95)
max_thres_msrp
```

Out[23]:
```
170.0
```

In [24]:
```python
df[df['MSRP']>max_thres_msrp]
```

Out[24]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	STATUS	QTR_ID	MONTH_ID	YEAR_ID	...	PRODUCTCODE	CUSTOMERNAME
26	10103	26	100.00	11	5404.62	1/29/2003 0:00	Shipped	1	1	2003	...	S10_1949	Baane Mini Imports
27	10112	29	100.00	1	7209.11	3/24/2003 0:00	Shipped	1	3	2003	...	S10_1949	Volvo Model Replicas, Co
28	10126	38	100.00	11	7329.06	5/28/2003 0:00	Shipped	2	5	2003	...	S10_1949	Corrida Auto Replicas, Ltd
29	10140	37	100.00	11	7374.10	7/24/2003 0:00	Shipped	3	7	2003	...	S10_1949	Technics Stores Inc.
30	10150	45	100.00	8	10993.50	9/19/2003 0:00	Shipped	3	9	2003	...	S10_1949	Dragon Souveniers, Ltd.
...	...	...	...	...	...	...	...	...	...	...	...	...	...
337	10359	49	62.09	5	3042.41	12/15/2004 0:00	Shipped	4	12	2004	...	S12_3891	Reims Collectables
338	10372	34	100.00	1	5941.50	1/26/2005 0:00	Shipped	1	1	2005	...	S12_3891	Tokyo Collectables, Ltd
339	10382	34	95.35	12	3241.90	2/17/2005 0:00	Shipped	1	2	2005	...	S12_3891	Mini Gifts Distributors Ltd.
340	10396	33	100.00	3	6109.29	3/23/2005 0:00	Shipped	1	3	2005	...	S12_3891	Mini Gifts Distributors Ltd.
341	10413	22	100.00	1	3387.78	5/5/2005 0:00	Shipped	2	5	2005	...	S12_3891	Gift Depot Inc.

131 rows × 22 columns

In [25]:
```python
df2=df[(df['PRICEEACH']>min_thres)&(df['SALES']>min_thres_sales)&(df['SALES']<max_thres_sales)&(df['MSRP']>min_thres_msrp)&(df['MSRP']<max_thres_ms
```

In [26]:
```python
df2.shape
```

Out[26]:
```
(2336, 22)
```

path=('C:\Users\ambalkt\Downloads\Power BI Practice\Projects\Sales\Car_Sales2.csv') df2.to_csv(path,index=False)

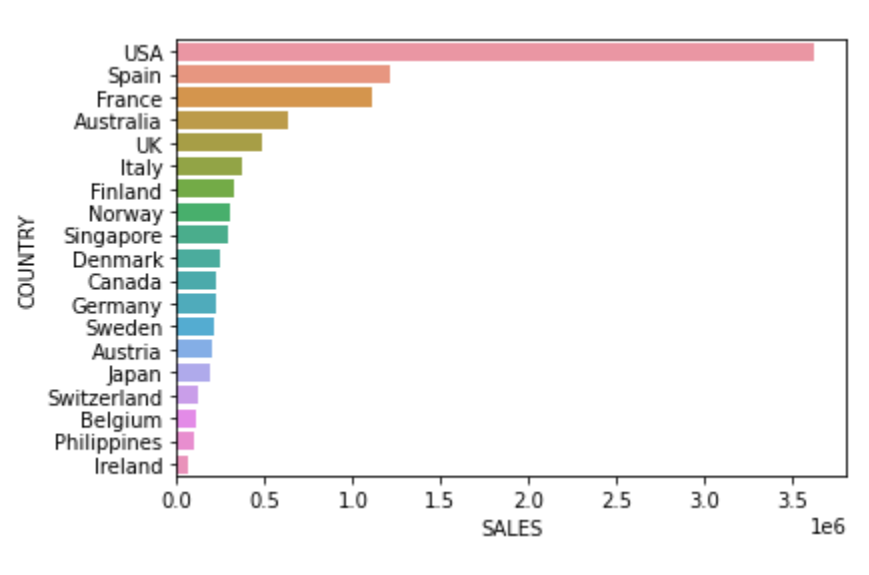
In [27]:
```python
top_country = df.groupby(['COUNTRY']).sum().sort_values('SALES', ascending = False)
top_country.reset_index(inplace = True)
top_country.head(5)
```

Out[27]:

	COUNTRY	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	QTR_ID	MONTH_ID	YEAR_ID	MSRP
0	USA	10297362	35659	84159.65	6369	3627982.83	2814	7380	2011812	101539
1	Spain	3513645	12429	28042.54	2186	1215686.92	859	2299	685341	33201
2	France	3223513	11090	25920.87	2028	1110916.52	776	1960	629233	31330
3	Australia	1898841	6246	15449.14	1181	630623.10	511	1345	370706	18493
4	UK	1476792	5013	11882.70	1027	478880.46	431	1166	288533	13876

In [28]:
```python
sns.barplot(x="SALES", y="COUNTRY", data=top_country, orient='h')
```

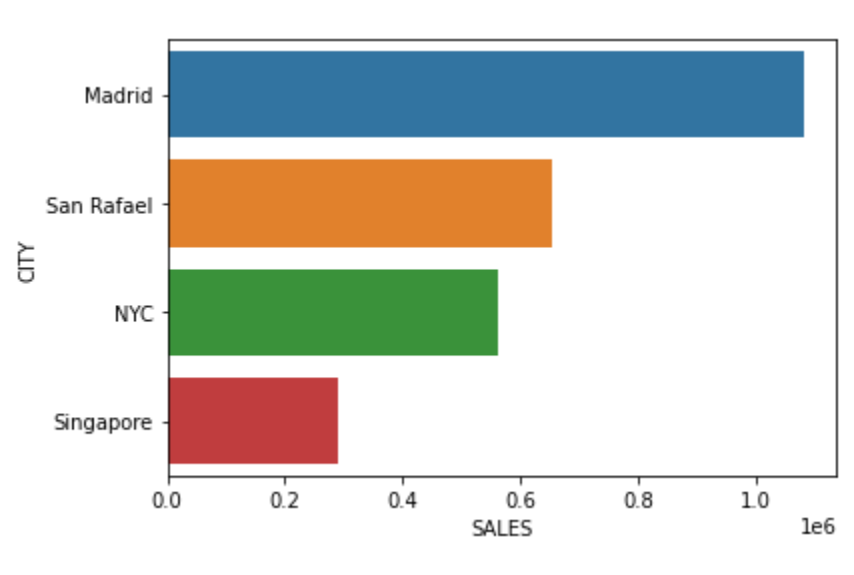
Out[28]: <AxesSubplot:xlabel='SALES', ylabel='COUNTRY'>



In [29]:
```python
top_city = df.groupby(['CITY']).sum().sort_values('SALES', ascending = False)
top_city.reset_index(inplace = True)
top_city=top_city.head(4)
```

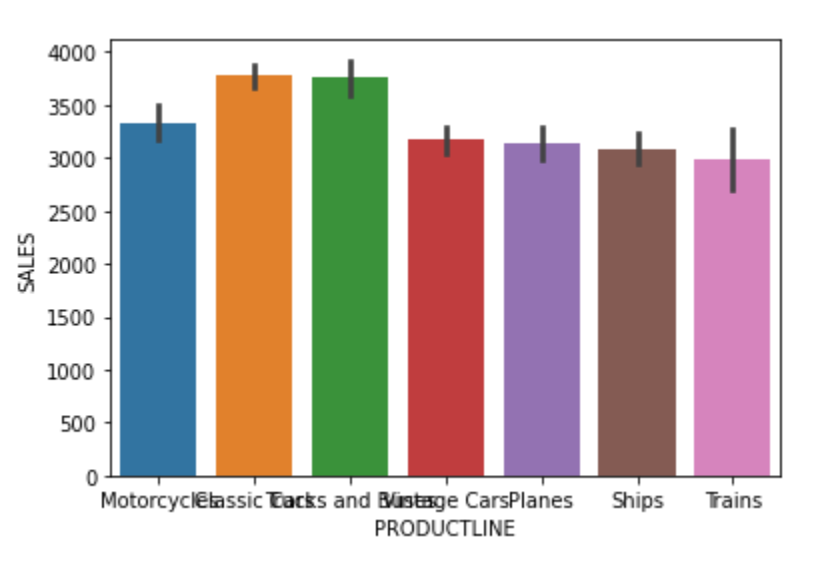
In [30]:
```python
sns.barplot(x="SALES", y="CITY", data=top_city, orient='h')
```

Out[30]: <AxesSubplot:xlabel='SALES', ylabel='CITY'>



In [31]:
```python
sns.barplot(x="PRODUCTLINE", y="SALES", data=df2)
```

Out[31]: <AxesSubplot:xlabel='PRODUCTLINE', ylabel='SALES'>



In [ ]: