In [1]:
```
import pandas as pd
```

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

In [3]:
```
df
```

Out[3]:

	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	...
...	...	...	...	...	...	...	...	...	...	...	...
2818	10350	20	100.00	15	2244.40	12/2/2004 0:00	Shipped	4	12	2004	...
2819	10373	29	100.00	1	3978.51	1/31/2005 0:00	Shipped	1	1	2005	...
2820	10386	43	100.00	4	5417.57	3/1/2005 0:00	Resolved	1	3	2005	...
2821	10397	34	62.24	1	2116.16	3/28/2005 0:00	Shipped	1	3	2005	...
2822	10414	47	65.52	9	3079.44	5/6/2005 0:00	On Hold	2	5	2005	...

2823 rows × 25 columns

In [4]:
```
df.describe()
```

Out[4]:

	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 [5]:
```
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]:
```
df.drop(columns=['ADDRESSLINE1','ADDRESSLINE2','STATUS','POSTALCODE','STATE','TERRITORY','CITY','ORDERNUMBER','CUSTOMER
```

In [7]:
```
df
```

Out[7]:
```
      QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  QTR_ID  MONTH_ID  YEAR_ID  PRODUCTLINE  MSRP  PRODUCTCODE  CO
   0               30      95.70                2  2871.00       1         2     2003  Motorcycles    95     S10_1678
   1               34      81.35                5  2765.90       2         5     2003  Motorcycles    95     S10_1678
   2               41      94.74                2  3884.34       3         7     2003  Motorcycles    95     S10_1678
   3               45      83.26                6  3746.70       3         8     2003  Motorcycles    95     S10_1678
   4               49     100.00               14  5205.27       4        10     2003  Motorcycles    95     S10_1678
 ...              ...        ...              ...      ...     ...       ...      ...          ...   ...          ...
2818               20     100.00               15  2244.40       4        12     2004        Ships    54     S72_3212
2819               29     100.00                1  3978.51       1         1     2005        Ships    54     S72_3212
2820               43     100.00                4  5417.57       1         3     2005        Ships    54     S72_3212
2821               34      62.24                1  2116.16       1         3     2005        Ships    54     S72_3212
2822               47      65.52                9  3079.44       2         5     2005        Ships    54     S72_3212

2823 rows × 12 columns
```

In [8]:
```
DEALSIZE = pd.get_dummies(df['DEALSIZE'])
```

In [9]:
```
DEALSIZE
```

Out[9]:
```
      Large  Medium  Small
   0  False   False   True
   1  False   False   True
   2  False    True  False
   3  False    True  False
   4  False    True  False
 ...    ...     ...    ...
2818  False   False   True
2819  False    True  False
2820  False    True  False
2821  False   False   True
2822  False    True  False

2823 rows × 3 columns
```

In [10]:
```
PRODUCTLINE = pd.get_dummies(df['PRODUCTLINE'])
COUNTRY = pd.get_dummies(df['COUNTRY'])
```

In [11]:
```
df = pd.concat([df,DEALSIZE,PRODUCTLINE,COUNTRY], axis=1)
```

In [12]:
```
df
```

Out[12]:
```
      QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  QTR_ID  MONTH_ID  YEAR_ID  PRODUCTLINE  MSRP  PRODUCTCODE  ...
   0               30      95.70                2  2871.00       1         2     2003  Motorcycles    95     S10_1678  ...
   1               34      81.35                5  2765.90       2         5     2003  Motorcycles    95     S10_1678  ...
   2               41      94.74                2  3884.34       3         7     2003  Motorcycles    95     S10_1678  ...
   3               45      83.26                6  3746.70       3         8     2003  Motorcycles    95     S10_1678  ...
   4               49     100.00               14  5205.27       4        10     2003  Motorcycles    95     S10_1678  ...
 ...              ...        ...              ...      ...     ...       ...      ...          ...   ...          ...  ...
2818               20     100.00               15  2244.40       4        12     2004        Ships    54     S72_3212  ...
2819               29     100.00                1  3978.51       1         1     2005        Ships    54     S72_3212  ...
2820               43     100.00                4  5417.57       1         3     2005        Ships    54     S72_3212  ...
2821               34      62.24                1  2116.16       1         3     2005        Ships    54     S72_3212  ...
2822               47      65.52                9  3079.44       2         5     2005        Ships    54     S72_3212  ...

2823 rows × 41 columns
```

```
In [13]: df.drop(columns=['DEALSIZE','PRODUCTLINE','COUNTRY'],axis=1, inplace=True)
```

```
In [14]: df
```

```
Out[14]:
      QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  QTR_ID  MONTH_ID  YEAR_ID  MSRP  PRODUCTCODE  Large  ...  Italy  Ja
   0               30      95.70                2  2871.00       1         2     2003    95     S10_1678  False  ...  False  F
   1               34      81.35                5  2765.90       2         5     2003    95     S10_1678  False  ...  False  F
   2               41      94.74                2  3884.34       3         7     2003    95     S10_1678  False  ...  False  F
   3               45      83.26                6  3746.70       3         8     2003    95     S10_1678  False  ...  False  F
   4               49     100.00               14  5205.27       4        10     2003    95     S10_1678  False  ...  False  F
 ...              ...        ...              ...      ...     ...       ...      ...   ...          ...    ...  ...    ...
2818               20     100.00               15  2244.40       4        12     2004    54     S72_3212  False  ...  False  F
2819               29     100.00                1  3978.51       1         1     2005    54     S72_3212  False  ...  False  F
2820               43     100.00                4  5417.57       1         3     2005    54     S72_3212  False  ...  False  F
2821               34      62.24                1  2116.16       1         3     2005    54     S72_3212  False  ...  False  F
2822               47      65.52                9  3079.44       2         5     2005    54     S72_3212  False  ...  False  F

2823 rows × 38 columns
```

```
In [15]: df['PRODUCTCODE'] = pd.Categorical(df['PRODUCTCODE']).codes
```

```
In [16]: df
```

```
Out[16]:
      QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  QTR_ID  MONTH_ID  YEAR_ID  MSRP  PRODUCTCODE  Large  ...  Italy  Ja
   0               30      95.70                2  2871.00       1         2     2003    95            0  False  ...  False  F
   1               34      81.35                5  2765.90       2         5     2003    95            0  False  ...  False  F
   2               41      94.74                2  3884.34       3         7     2003    95            0  False  ...  False  F
   3               45      83.26                6  3746.70       3         8     2003    95            0  False  ...  False  F
   4               49     100.00               14  5205.27       4        10     2003    95            0  False  ...  False  F
 ...              ...        ...              ...      ...     ...       ...      ...   ...          ...    ...  ...    ...
2818               20     100.00               15  2244.40       4        12     2004    54          108  False  ...  False  F
2819               29     100.00                1  3978.51       1         1     2005    54          108  False  ...  False  F
2820               43     100.00                4  5417.57       1         3     2005    54          108  False  ...  False  F
2821               34      62.24                1  2116.16       1         3     2005    54          108  False  ...  False  F
2822               47      65.52                9  3079.44       2         5     2005    54          108  False  ...  False  F

2823 rows × 38 columns
```

In [17]:
```
df.dtypes
```

Out[17]:
```
QUANTITYORDERED       int64
PRICEEACH           float64
ORDERLINENUMBER       int64
SALES               float64
QTR_ID                int64
MONTH_ID              int64
YEAR_ID               int64
MSRP                  int64
PRODUCTCODE            int8
Large                  bool
Medium                 bool
Small                  bool
Classic Cars           bool
Motorcycles            bool
Planes                 bool
Ships                  bool
Trains                 bool
Trucks and Buses       bool
Vintage Cars           bool
Australia              bool
Austria                bool
Belgium                bool
Canada                 bool
Denmark                bool
Finland                bool
France                 bool
Germany                bool
Ireland                bool
Italy                  bool
Japan                  bool
Norway                 bool
Philippines            bool
Singapore              bool
Spain                  bool
Sweden                 bool
Switzerland            bool
UK                     bool
USA                    bool
dtype: object
```

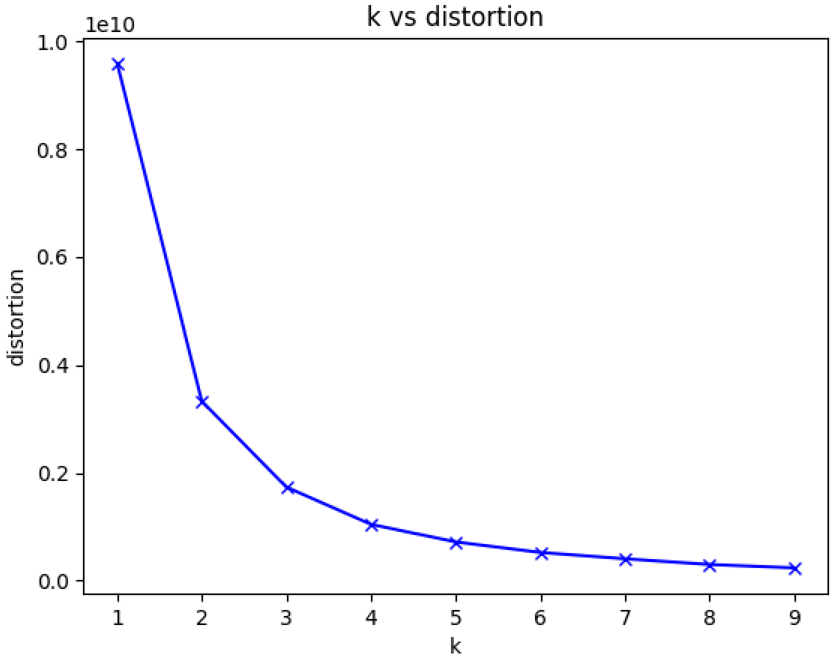
In [18]:
```
from sklearn.cluster import KMeans
```

In [19]:
```
distortion = []
k = range(1,10)
for n in k:
    km = KMeans(n_clusters=n)
    km.fit(df)
    distortion.append(km.inertia_)
```

In [20]:
```
import matplotlib.pyplot as plt
```

In [21]:

```
plt.plot(k,distortion,'bx-')
plt.xlabel('k')
plt.ylabel('distortion')
plt.title('k vs distortion')
plt.show()
```



In [22]:

```
x_train = df.values
model = KMeans(n_clusters=4, random_state=2)
model.fit(x_train)
```

Out[22]:

```
KMeans(n_clusters=4, random_state=2)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [23]:

```
pred = model.predict(x_train)
```

In [24]:

```
print(pred)
```

```
[3 3 3 ... 2 0 3]
```

In [25]:

```
import numpy as np
```

In [26]:

```
unique,count = np.unique(pred, return_counts=True)
```

In [27]:

```
print(unique)
```

```
[0 1 2 3]
```

In [28]:

```
print(count)
```

```
[1041  199  562 1021]
```

In [29]:

```
pred1 = pd.DataFrame(pred)
```

In [30]:
```
pred1
```

Out[30]:
```
         0
   0     3
   1     3
   2     3
   3     3
   4     2
 ...   ...
2818     0
2819     3
2820     2
2821     0
2822     3

2823 rows × 1 columns
```

In [31]:
```
df = pd.concat([df,pred1],axis=1)
df
```

Out[31]:
```
      QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  QTR_ID  MONTH_ID  YEAR_ID  MSRP  PRODUCTCODE  Large  ...  Japan  N
   0               30      95.70                2  2871.00       1         2     2003    95            0  False  ...  False
   1               34      81.35                5  2765.90       2         5     2003    95            0  False  ...  False
   2               41      94.74                2  3884.34       3         7     2003    95            0  False  ...  False
   3               45      83.26                6  3746.70       3         8     2003    95            0  False  ...  False
   4               49     100.00               14  5205.27       4        10     2003    95            0  False  ...  False
 ...              ...        ...              ...      ...     ...       ...      ...   ...          ...    ...  ...    ...
2818               20     100.00               15  2244.40       4        12     2004    54          108  False  ...  False
2819               29     100.00                1  3978.51       1         1     2005    54          108  False  ...  False
2820               43     100.00                4  5417.57       1         3     2005    54          108  False  ...  False
2821               34      62.24                1  2116.16       1         3     2005    54          108  False  ...  False
2822               47      65.52                9  3079.44       2         5     2005    54          108  False  ...  False

2823 rows × 39 columns
```

In [35]:
```
df_temp = df.drop(columns=[0],axis=1)
```

In [39]:
```
#cluster_centers is used to find the centroid values of the clusters
model.cluster_centers_
```

Out[39]:
```
array([[ 2.98893167e+01,  6.54380366e+01,  6.61886429e+00,
         1.88419459e+03,  2.72473532e+00,  7.13282002e+00,
         2.00381521e+03,  7.33378248e+01,  6.32569779e+01,
         2.08166817e-17,  5.55111512e-16,  1.00000000e+00,
         2.59865255e-01,  1.22232916e-01,  1.17420597e-01,
         8.95091434e-02,  4.42733397e-02,  8.75842156e-02,
         2.79114533e-01,  7.21847931e-02,  1.73243503e-02,
         1.44369586e-02,  2.88739172e-02,  1.82868142e-02,
         2.88739172e-02,  1.21270452e-01,  2.21366699e-02,
         6.73724735e-03,  4.13859480e-02,  1.63618864e-02,
         3.27237729e-02,  7.69971126e-03,  2.88739172e-02,
         1.18383061e-01,  1.82868142e-02,  4.81231954e-03,
         5.48604427e-02,  3.46487007e-01],
       [ 4.63718593e+01,  9.98418593e+01,  5.52763819e+00,
         7.98362548e+03,  2.65829146e+00,  6.89949749e+00,
         2.00391960e+03,  1.54291457e+02,  2.80502513e+01,
         7.88944724e-01,  2.11055276e-01,  3.33066907e-16,
         5.82914573e-01,  1.20603015e-01,  6.03015075e-02,
         1.00502513e-02,  5.02512563e-03,  7.53768844e-02,
         1.45728643e-01,  4.52261307e-02,  2.51256281e-02,
         5.02512563e-03,  1.00502513e-02,  3.51758794e-02,
         3.51758794e-02,  1.25628141e-01,  2.51256281e-02,
         1.00502513e-02,  3.51758794e-02,  2.01005025e-02,
         2.51256281e-02,  5.02512563e-03,  2.51256281e-02,
         1.15577889e-01,  2.51256281e-02,  5.02512563e-03,
         2.51256281e-02,  4.02010050e-01],
       [ 4.07491103e+01,  9.95422598e+01,  6.26690391e+00,
         5.30138568e+03,  2.73309609e+00,  7.12989324e+00,
         2.00380427e+03,  1.27149466e+02,  4.08665480e+01,
         2.08166817e-17,  1.00000000e+00, -6.66133815e-16,
         4.55516014e-01,  1.01423488e-01,  6.22775801e-02,
         3.91459075e-02,  1.95729537e-02,  1.61921708e-01,
         1.60142349e-01,  6.76156584e-02,  2.13523132e-02,
         1.24555160e-02,  1.95729537e-02,  2.13523132e-02,
         3.02491103e-02,  9.60854093e-02,  1.77935943e-02,
         5.33807829e-03,  3.02491103e-02,  1.77935943e-02,
         3.91459075e-02,  1.24555160e-02,  3.73665480e-02,
         1.26334520e-01,  2.13523132e-02,  1.77935943e-02,
         4.62633452e-02,  3.59430605e-01],
       [ 3.50762463e+01,  9.02900000e+01,  6.60312805e+00,
         3.42798675e+03,  2.71358749e+00,  7.06842620e+00,
         2.00380059e+03,  1.03577713e+02,  5.62355816e+01,
         2.08166817e-17,  7.62463343e-01,  2.37536657e-01,
         3.17693060e-01,  1.20234604e-01,  1.33919844e-01,
         1.14369501e-01,  1.85728250e-02,  1.01661779e-01,
         1.93548387e-01,  6.15835777e-02,  1.95503421e-02,
         9.77517107e-03,  2.63929619e-02,  2.44379277e-02,
         3.71456500e-02,  1.06549365e-01,  2.34604106e-02,
         3.91006843e-03,  4.49657869e-02,  2.05278592e-02,
         2.34604106e-02,  9.77517107e-03,  2.24828935e-02,
         1.22189638e-01,  2.05278592e-02,  1.46627566e-02,
         5.47409580e-02,  3.53861193e-01]])
```

In [40]:
```
cc = pd.DataFrame(data=model.cluster_centers_, columns=[df_temp.columns])
```

In [41]:
```
cc
```

Out[41]:

	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	QTR_ID	MONTH_ID	YEAR_ID	MSRP	PRODUCTCODE	Large
0	29.889317	65.438037	6.618864	1884.194591	2.724735	7.132820	2003.815207	73.337825	63.256978	2.081668e-17
1	46.371859	99.841859	5.527638	7983.625477	2.658291	6.899497	2003.919598	154.291457	28.050251	7.889447e-01
2	40.749110	99.542260	6.266904	5301.385676	2.733096	7.129893	2003.804270	127.149466	40.866548	2.081668e-17
3	35.076246	90.290000	6.603128	3427.986755	2.713587	7.068426	2003.800587	103.577713	56.235582	2.081668e-17

4 rows × 38 columns

In [ ]: