```
import pandas as pd

df = pd.read_csv('./sales_data_sample.csv',encoding='latin1')
df.head()

   ORDERNUMBER  QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER
SALES  \
0        10107               30      95.70                2  2871.00

1        10121               34      81.35                5  2765.90

2        10134               41      94.74                2  3884.34

3        10145               45      83.26                6  3746.70

4        10159               49     100.00               14  5205.27


          ORDERDATE   STATUS  QTR_ID  MONTH_ID  YEAR_ID  ...  \
0   2/24/2003 0:00  Shipped       1         2     2003  ...
1    5/7/2003 0:00  Shipped       2         5     2003  ...
2    7/1/2003 0:00  Shipped       3         7     2003  ...
3   8/25/2003 0:00  Shipped       3         8     2003  ...
4  10/10/2003 0:00  Shipped       4        10     2003  ...

                     ADDRESSLINE1  ADDRESSLINE2           CITY STATE  \
0         897 Long Airport Avenue           NaN            NYC    NY
1              59 rue de l'Abbaye           NaN          Reims   NaN
2  27 rue du Colonel Pierre Avia           NaN          Paris   NaN
3              78934 Hillside Dr.           NaN       Pasadena    CA
4                 7734 Strong St.           NaN  San Francisco    CA

  POSTALCODE COUNTRY TERRITORY CONTACTLASTNAME CONTACTFIRSTNAME
DEALSIZE
0      10022     USA       NaN              Yu              Kwai
Small
1      51100  France      EMEA         Henriot              Paul
Small
2      75508  France      EMEA        Da Cunha            Daniel
Medium
3      90003     USA       NaN           Young             Julie
Medium
4        NaN     USA       NaN           Brown             Julie
Medium

[5 rows x 25 columns]

df.describe().T

                  count          mean          std       min       25%
\
```

| | | | | | |
|---|---|---|---|---|---|
| ORDERNUMBER | 2823.0 | 10258.725115 | 92.085478 | 10100.00 | 10180.00 |
| QUANTITYORDERED | 2823.0 | 35.092809 | 9.741443 | 6.00 | 27.00 |
| PRICEEACH | 2823.0 | 83.658544 | 20.174277 | 26.88 | 68.86 |
| ORDERLINENUMBER | 2823.0 | 6.466171 | 4.225841 | 1.00 | 3.00 |
| SALES | 2823.0 | 3553.889072 | 1841.865106 | 482.13 | 2203.43 |
| QTR_ID | 2823.0 | 2.717676 | 1.203878 | 1.00 | 2.00 |
| MONTH_ID | 2823.0 | 7.092455 | 3.656633 | 1.00 | 4.00 |
| YEAR_ID | 2823.0 | 2003.815090 | 0.699670 | 2003.00 | 2003.00 |
| MSRP | 2823.0 | 100.715551 | 40.187912 | 33.00 | 68.00 |

| | 50% | 75% | max |
|---|---|---|---|
| ORDERNUMBER | 10262.0 | 10333.5 | 10425.0 |
| QUANTITYORDERED | 35.0 | 43.0 | 97.0 |
| PRICEEACH | 95.7 | 100.0 | 100.0 |
| ORDERLINENUMBER | 6.0 | 9.0 | 18.0 |
| SALES | 3184.8 | 4508.0 | 14082.8 |
| QTR_ID | 3.0 | 4.0 | 4.0 |
| MONTH_ID | 8.0 | 11.0 | 12.0 |
| YEAR_ID | 2004.0 | 2004.0 | 2005.0 |
| MSRP | 99.0 | 124.0 | 214.0 |

```
df.isnull().sum()
```

| | |
|---|---|
| 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
```

```
df.drop(columns =
['ADDRESSLINE1','ADDRESSLINE2','CITY','STATE','POSTALCODE','PHONE','TE
RRITORY','CONTACTLASTNAME','CONTACTFIRSTNAME','ORDERNUMBER','STATUS','
STATE','ORDERDATE'],axis =1, inplace=True)
```

```
df.head()
```

```
   QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  QTR_ID
MONTH_ID  \
0               30      95.70                2  2871.00       1
2
1               34      81.35                5  2765.90       2
5
2               41      94.74                2  3884.34       3
7
3               45      83.26                6  3746.70       3
8
4               49     100.00               14  5205.27       4
10

   YEAR_ID  PRODUCTLINE  MSRP PRODUCTCODE              CUSTOMERNAME
COUNTRY  \
0     2003  Motorcycles    95    S10_1678         Land of Toys Inc.
USA
1     2003  Motorcycles    95    S10_1678        Reims Collectables
France
2     2003  Motorcycles    95    S10_1678           Lyon Souveniers
France
3     2003  Motorcycles    95    S10_1678         Toys4GrownUps.com
USA
4     2003  Motorcycles    95    S10_1678  Corporate Gift Ideas Co.
USA

  DEALSIZE
0    Small
1    Small
2   Medium
3   Medium
4   Medium
```

```
df.shape
```

```
(2823, 13)
```

```
df.drop(columns = ['CUSTOMERNAME'],axis =1, inplace=True)

df.shape

(2823, 12)

df.dtypes

QUANTITYORDERED      int64
PRICEEACH          float64
ORDERLINENUMBER      int64
SALES              float64
QTR_ID               int64
MONTH_ID             int64
YEAR_ID              int64
PRODUCTLINE         object
MSRP                 int64
PRODUCTCODE         object
COUNTRY             object
DEALSIZE            object
dtype: object

dealSize = pd.get_dummies(df['DEALSIZE'])
dealSize

      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 x 3 columns]

COUNTRY = pd.get_dummies(df['COUNTRY'])
COUNTRY

      Australia  Austria  Belgium  Canada  Denmark  Finland  France
Germany  \
0         False    False    False   False    False    False   False
False
1         False    False    False   False    False    False    True
False
2         False    False    False   False    False    False    True
False
```

```
3         False    False    False   False    False    False   False
False
4         False    False    False   False    False    False   False
False
...         ...      ...      ...     ...      ...      ...     ...
...
2818      False    False    False   False    False    False   False
False
2819      False    False    False   False    False     True   False
False
2820      False    False    False   False    False    False   False
False
2821      False    False    False   False    False    False    True
False
2822      False    False    False   False    False    False   False
False

      Ireland  Italy  Japan  Norway  Philippines  Singapore  Spain
Sweden  \
0       False  False  False   False        False      False  False
False
1       False  False  False   False        False      False  False
False
2       False  False  False   False        False      False  False
False
3       False  False  False   False        False      False  False
False
4       False  False  False   False        False      False  False
False
...       ...    ...    ...     ...          ...        ...    ...
...
2818    False  False  False   False        False      False   True
False
2819    False  False  False   False        False      False  False
False
2820    False  False  False   False        False      False   True
False
2821    False  False  False   False        False      False  False
False
2822    False  False  False   False        False      False  False
False

      Switzerland     UK    USA
0           False  False   True
1           False  False  False
2           False  False  False
3           False  False   True
4           False  False   True
...           ...    ...    ...
```

```
2818         False  False  False
2819         False  False  False
2820         False  False  False
2821         False  False  False
2822         False  False   True

[2823 rows x 19 columns]
```

```
productLine = pd.get_dummies(df['PRODUCTLINE'])
productLine
```

```
      Classic Cars  Motorcycles  Planes  Ships  Trains  Trucks and
Buses  \
0             False         True   False  False   False
False
1             False         True   False  False   False
False
2             False         True   False  False   False
False
3             False         True   False  False   False
False
4             False         True   False  False   False
False
...             ...          ...     ...    ...     ...
..
2818          False        False   False   True   False
False
2819          False        False   False   True   False
False
2820          False        False   False   True   False
False
2821          False        False   False   True   False
False
2822          False        False   False   True   False
False

      Vintage Cars
0             False
1             False
2             False
3             False
4             False
...             ...
2818          False
2819          False
2820          False
2821          False
2822          False

[2823 rows x 7 columns]
```

```
df.shape
```

```
(2823, 12)
```

```
df = pd.concat([df,dealSize,COUNTRY,productLine],axis=1)
```

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

```
df.shape
```

```
(2823, 38)
```

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

```
df.describe().T
```

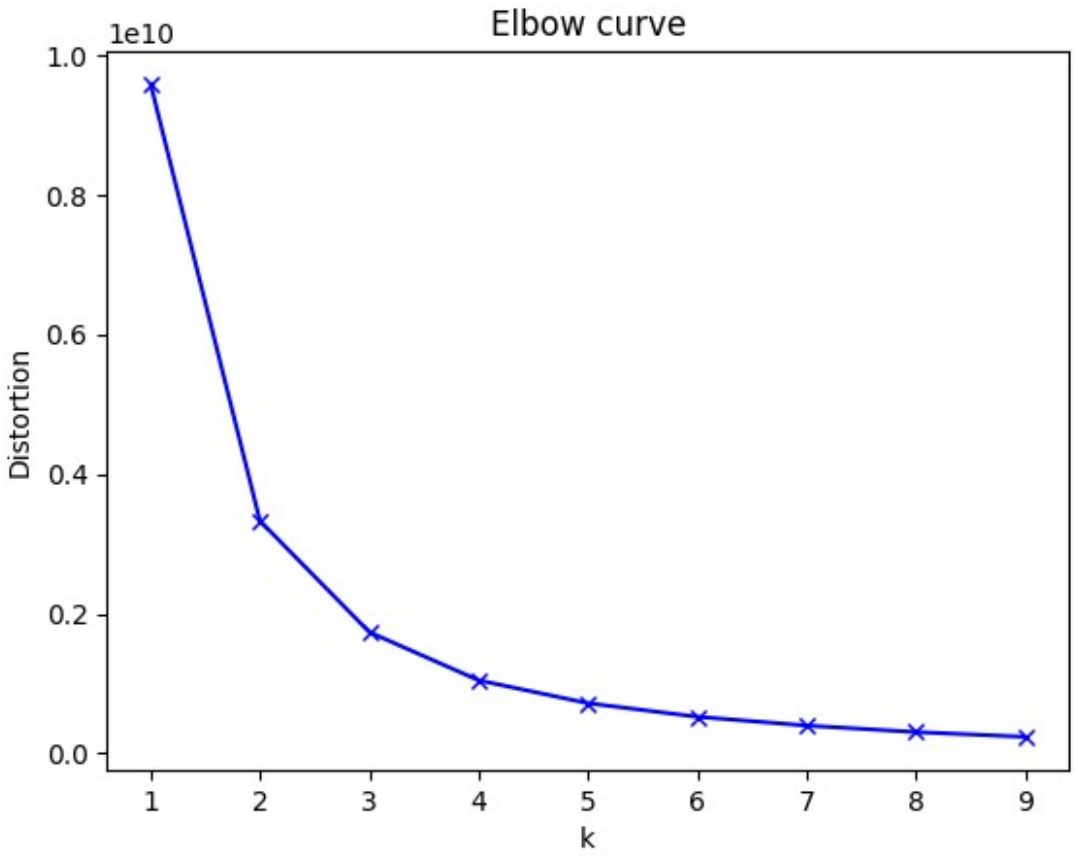
```
                    count         mean          std      min      25%
50%  \
QUANTITYORDERED  2823.0    35.092809     9.741443     6.00    27.00
35.0
PRICEEACH        2823.0    83.658544    20.174277    26.88    68.86
95.7
ORDERLINENUMBER  2823.0     6.466171     4.225841     1.00     3.00
6.0
SALES            2823.0  3553.889072  1841.865106   482.13  2203.43
3184.8
QTR_ID           2823.0     2.717676     1.203878     1.00     2.00
3.0
MONTH_ID         2823.0     7.092455     3.656633     1.00     4.00
8.0
YEAR_ID          2823.0  2003.815090     0.699670  2003.00  2003.00
2004.0
MSRP             2823.0   100.715551    40.187912    33.00    68.00
99.0
PRODUCTCODE      2823.0    53.773291    31.585298     0.00    27.00
53.0

                     75%      max
QUANTITYORDERED    43.0     97.0
PRICEEACH         100.0    100.0
ORDERLINENUMBER     9.0     18.0
SALES            4508.0  14082.8
QTR_ID              4.0      4.0
MONTH_ID           11.0     12.0
YEAR_ID          2004.0   2005.0
MSRP              124.0    214.0
PRODUCTCODE        81.0    108.0
```

```
df.head()
```

```
   QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  QTR_ID
MONTH_ID  \
0               30      95.70                2  2871.00       1
2
1               34      81.35                5  2765.90       2
5
2               41      94.74                2  3884.34       3
7
3               45      83.26                6  3746.70       3
8
4               49     100.00               14  5205.27       4
10

   YEAR_ID  MSRP  PRODUCTCODE  Large  ...  Switzerland     UK
USA  \
0     2003    95            0  False  ...        False  False   True

1     2003    95            0  False  ...        False  False  False

2     2003    95            0  False  ...        False  False  False

3     2003    95            0  False  ...        False  False   True

4     2003    95            0  False  ...        False  False   True


   Classic Cars  Motorcycles  Planes  Ships  Trains  Trucks and Buses
\
0         False         True   False  False   False              False

1         False         True   False  False   False              False

2         False         True   False  False   False              False

3         False         True   False  False   False              False

4         False         True   False  False   False              False


   Vintage Cars
0         False
1         False
2         False
3         False
4         False

[5 rows x 38 columns]
```

```
from sklearn.cluster import KMeans

import matplotlib.pyplot as plt
```

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


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




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

print(pred)
```

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

```
import numpy as np

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

print(unique)

[0 1 2 3]

print(count)

[1041  199  562 1021]

pred_df = pd.DataFrame(pred)

df = pd.concat([df,pred_df],axis=1)

df.head()
```

| | QUANTITYORDERED | PRICEEACH | ORDERLINENUMBER | SALES | QTR_ID | MONTH_ID | \ |
|---|---|---|---|---|---|---|---|
| 0 | 30 | 95.70 | 2 | 2871.00 | 1 | 2 | |
| 1 | 34 | 81.35 | 5 | 2765.90 | 2 | 5 | |
| 2 | 41 | 94.74 | 2 | 3884.34 | 3 | 7 | |
| 3 | 45 | 83.26 | 6 | 3746.70 | 3 | 8 | |
| 4 | 49 | 100.00 | 14 | 5205.27 | 4 | 10 | |

| | YEAR_ID | MSRP | PRODUCTCODE | Large | ... | UK | USA | Classic Cars | \ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 2003 | 95 | 0 | False | ... | False | True | False | |
| 1 | 2003 | 95 | 0 | False | ... | False | False | False | |
| 2 | 2003 | 95 | 0 | False | ... | False | False | False | |
| 3 | 2003 | 95 | 0 | False | ... | False | True | False | |
| 4 | 2003 | 95 | 0 | False | ... | False | True | False | |

| | Motorcycles | Planes | Ships | Trains | Trucks and Buses | Vintage Cars | 0 |
|---|---|---|---|---|---|---|---|
| 0 | True | False | False | False | False | False | 3 |
| 1 | True | False | False | False | False | False | 3 |
| 2 | True | False | False | False | False | False | 3 |

```
3          True   False  False   False              False          False
3
4          True   False  False   False              False          False
2

[5 rows x 39 columns]
```

```
df.shape
```

```
(2823, 39)
```

```
print(model.cluster_centers_)
```

```
[[ 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
   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  2.59865255e-01
   1.22232916e-01  1.17420597e-01  8.95091434e-02  4.42733397e-02
   8.75842156e-02  2.79114533e-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
   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  5.82914573e-01
   1.20603015e-01  6.03015075e-02  1.00502513e-02  5.02512563e-03
   7.53768844e-02  1.45728643e-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
   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  4.55516014e-01
   1.01423488e-01  6.22775801e-02  3.91459075e-02  1.95729537e-02
   1.61921708e-01  1.60142349e-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
   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  3.17693060e-01
   1.20234604e-01  1.33919844e-01  1.14369501e-01  1.85728250e-02
   1.01661779e-01  1.93548387e-01]]

df2 = df.drop(columns = [0],axis =1)

cc = pd.DataFrame(data = model.cluster_centers_, columns =
[df2.columns])

cc
```

```
  QUANTITYORDERED  PRICEEACH ORDERLINENUMBER        SALES    QTR_ID
MONTH_ID  \
0       29.889317  65.438037        6.618864  1884.194591  2.724735
7.132820
1       46.371859  99.841859        5.527638  7983.625477  2.658291
6.899497
2       40.749110  99.542260        6.266904  5301.385676  2.733096
7.129893
3       35.076246  90.290000        6.603128  3427.986755  2.713587
7.068426

       YEAR_ID        MSRP PRODUCTCODE         Large  ... Switzerland
\
0  2003.815207   73.337825   63.256978  2.081668e-17  ...    0.004812

1  2003.919598  154.291457   28.050251  7.889447e-01  ...    0.005025

2  2003.804270  127.149466   40.866548  2.081668e-17  ...    0.017794

3  2003.800587  103.577713   56.235582  2.081668e-17  ...    0.014663


         UK       USA Classic Cars Motorcycles    Planes     Ships
Trains  \
0  0.054860  0.346487     0.259865    0.122233  0.117421  0.089509
0.044273
1  0.025126  0.402010     0.582915    0.120603  0.060302  0.010050
0.005025
2  0.046263  0.359431     0.455516    0.101423  0.062278  0.039146
0.019573
3  0.054741  0.353861     0.317693    0.120235  0.133920  0.114370
0.018573

  Trucks and Buses Vintage Cars
0         0.087584     0.279115
1         0.075377     0.145729
2         0.161922     0.160142
3         0.101662     0.193548

[4 rows x 38 columns]
```