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

In [2]:
```
##### Doing this in Python was too slow. Used R instead

## Use when doing full dataset analysis
#df = pd.read_csv('Medicare_Provider_Util_Payment_PUF_CY2016.txt',sep
='\t',engine='python')
## Trim Dataset by dropping unnecessary columns and filter to business r
elevant data only
#df.drop(['NPI','NPPES_ENTITY_CODE','NPPES_PROVIDER_LAST_ORG_NAME','NPPE
S_PROVIDER_FIRST_NAME','NPPES_PROVIDER_MI','NPPES_PROVIDER_STREET1','NPP
ES_PROVIDER_STREET2','HCPCS_DESCRIPTION'],axis=1,inplace = True)
#df = df[(df['MEDICARE_PARTICIPATION_INDICATOR']=='Y') & (df['NPPES_PROV
IDER_COUNTRY']=='US') & (df['PROVIDER_TYPE']=='Diagnostic Radiology')]
#df.drop(['MEDICARE_PARTICIPATION_INDICATOR','NPPES_PROVIDER_COUNTRY','P
ROVIDER_TYPE'],axis=1,inplace = True)

df = pd.read_csv('MedicareData.csv')
df.drop(['Unnamed: 0'],inplace = True, axis = 1)
```

In [3]:
```
# Further subsegment the data to only include data in California
df = df[(df['NPPES_PROVIDER_STATE']=='CA') & (df['HCPCS_DRUG_INDICATOR']
=='N')]
```

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 100184 entries, 266 to 1259858
Data columns (total 16 columns):
NPPES_CREDENTIALS                98969 non-null object
NPPES_PROVIDER_GENDER            100184 non-null object
NPPES_PROVIDER_CITY              100184 non-null object
NPPES_PROVIDER_ZIP               100184 non-null int64
NPPES_PROVIDER_STATE             100184 non-null object
PROVIDER_TYPE                    100184 non-null object
PLACE_OF_SERVICE                 100184 non-null object
HCPCS_CODE                       100184 non-null object
HCPCS_DRUG_INDICATOR             100184 non-null object
LINE_SRVC_CNT                    100184 non-null float64
BENE_UNIQUE_CNT                  100184 non-null int64
BENE_DAY_SRVC_CNT                100184 non-null int64
AVERAGE_MEDICARE_ALLOWED_AMT     100184 non-null float64
AVERAGE_SUBMITTED_CHRG_AMT       100184 non-null float64
AVERAGE_MEDICARE_PAYMENT_AMT     100184 non-null float64
AVERAGE_MEDICARE_STANDARD_AMT    100184 non-null float64
dtypes: float64(5), int64(3), object(8)
memory usage: 13.0+ MB
```

In [5]:
```
df.head()
```

Out[5]:

| | NPPES_CREDENTIALS | NPPES_PROVIDER_GENDER | NPPES_PROVIDER_CITY | NPPES_PROVID |
|---|---|---|---|---|
| **266** | M.D | M | SAN DIEGO | |
| **267** | M.D | M | SAN DIEGO | |
| **268** | M.D | M | SAN DIEGO | |
| **269** | M.D | M | SAN DIEGO | |
| **270** | M.D | M | SAN DIEGO | |

In [6]:
```
df.describe()
```

Out[6]:

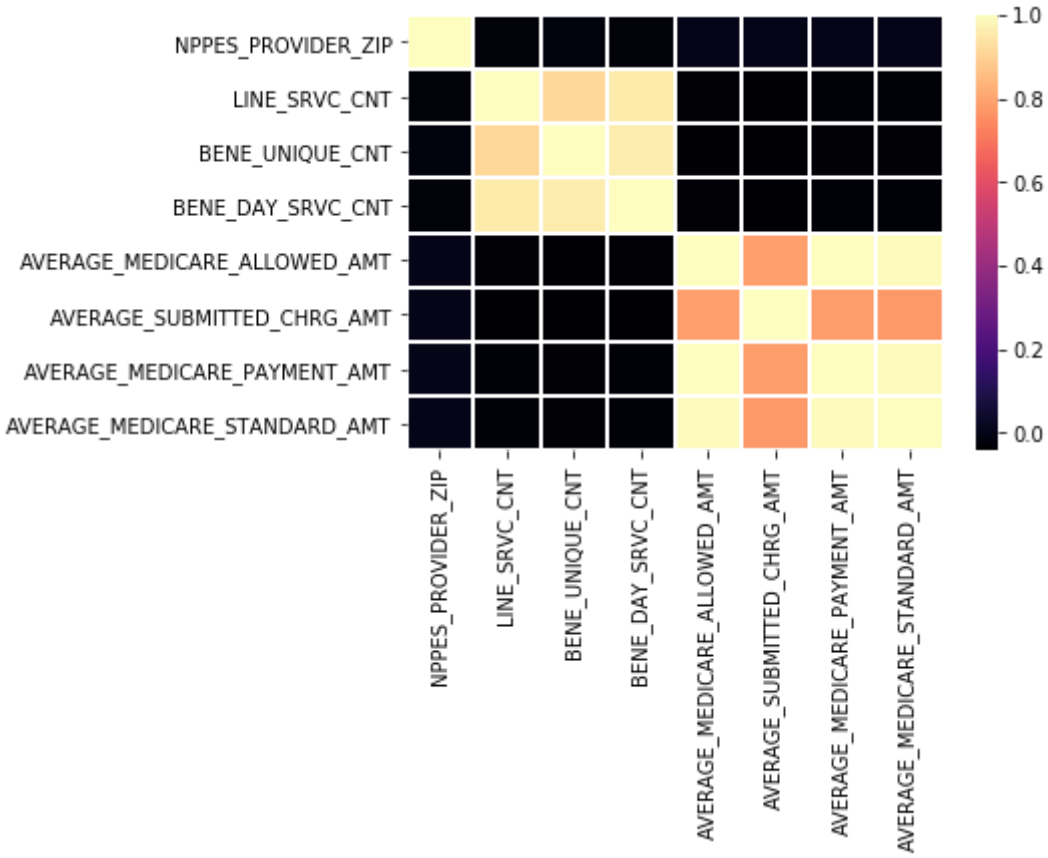
| | NPPES_PROVIDER_ZIP | LINE_SRVC_CNT | BENE_UNIQUE_CNT | BENE_DAY_SRVC_CNT | AVE |
|---|---|---|---|---|---|
| **count** | 1.001840e+05 | 100184.000000 | 100184.000000 | 100184.000000 | |
| **mean** | 8.438933e+08 | 73.624940 | 65.916893 | 71.381668 | |
| **std** | 2.717836e+08 | 197.305172 | 146.935958 | 183.850571 | |
| **min** | 6.267700e+04 | 11.000000 | 11.000000 | 11.000000 | |
| **25%** | 9.110827e+08 | 17.000000 | 16.000000 | 16.000000 | |
| **50%** | 9.269164e+08 | 28.000000 | 27.000000 | 27.000000 | |
| **75%** | 9.455347e+08 | 60.000000 | 56.000000 | 58.000000 | |
| **max** | 9.616148e+08 | 14925.000000 | 5626.000000 | 13187.000000 | |

In [8]:
```
sns.pairplot(df)
```

Out[8]: <seaborn.axisgrid.PairGrid at 0x1a2a7e43c8>

In [7]:
```
sns.heatmap(df.corr(),cmap = 'magma',linecolor = 'white',lw=1)
```

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1cc89828>



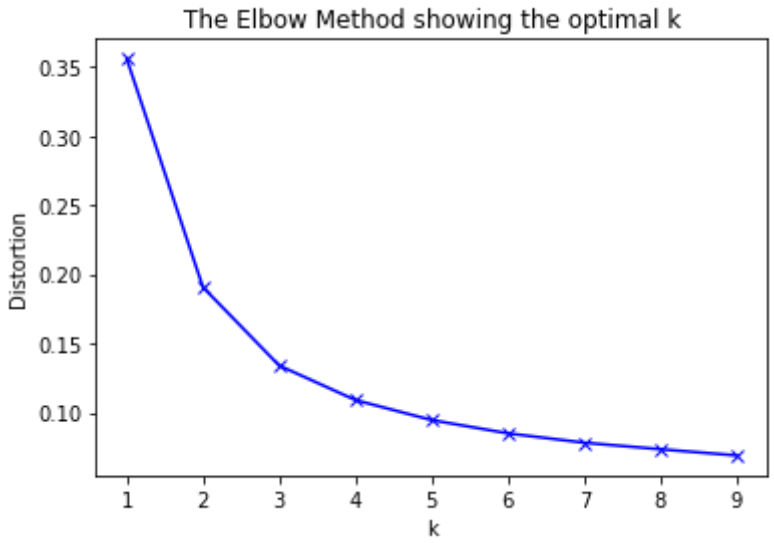

In [18]:
```
from sklearn.preprocessing import OneHotEncoder
df_analyze = pd.DataFrame()
Gender_OHE = OneHotEncoder(categories = 'auto')
X = Gender_OHE.fit_transform(df.NPPES_PROVIDER_GENDER.values.reshape(-1,
1)).toarray()
dfOneHot = pd.DataFrame(X,columns = ['Gender_'+str(int(i))for i in range
(X.shape[1])])
dfOneHot.reset_index(drop=True, inplace=True)
df.reset_index(drop=True, inplace=True)
df_analyze = pd.concat([df,dfOneHot],axis = 1)
```

In [19]:
```
POS_OHE = OneHotEncoder(categories = 'auto')
X = POS_OHE.fit_transform(df.PLACE_OF_SERVICE.values.reshape(-1,1)).toar
ray()
dfOneHot = pd.DataFrame(X,columns = ['PLACEOFSERVICE'+str(int(i))for i i
n range(X.shape[1])])
dfOneHot.reset_index(drop=True, inplace=True)
df_analyze = pd.concat([df_analyze,dfOneHot],axis = 1)
```

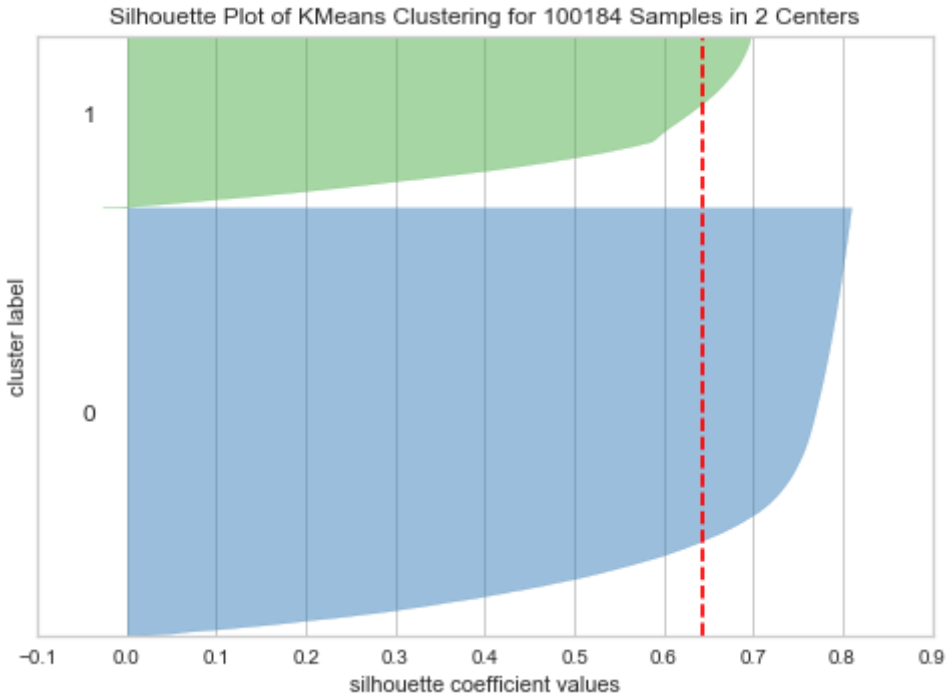
In [21]:
```
# drop unwanted columns from analysis
df_analyze.drop(['NPPES_CREDENTIALS','NPPES_PROVIDER_GENDER','NPPES_PROV
IDER_CITY','NPPES_PROVIDER_STATE','NPPES_PROVIDER_ZIP',
                 'PROVIDER_TYPE','HCPCS_CODE','HCPCS_DRUG_INDICATOR','PLA
CE_OF_SERVICE',
                 'BENE_UNIQUE_CNT','AVERAGE_MEDICARE_ALLOWED_AMT',
                  'AVERAGE_MEDICARE_PAYMENT_AMT'], axis=1,inplace = True)
```

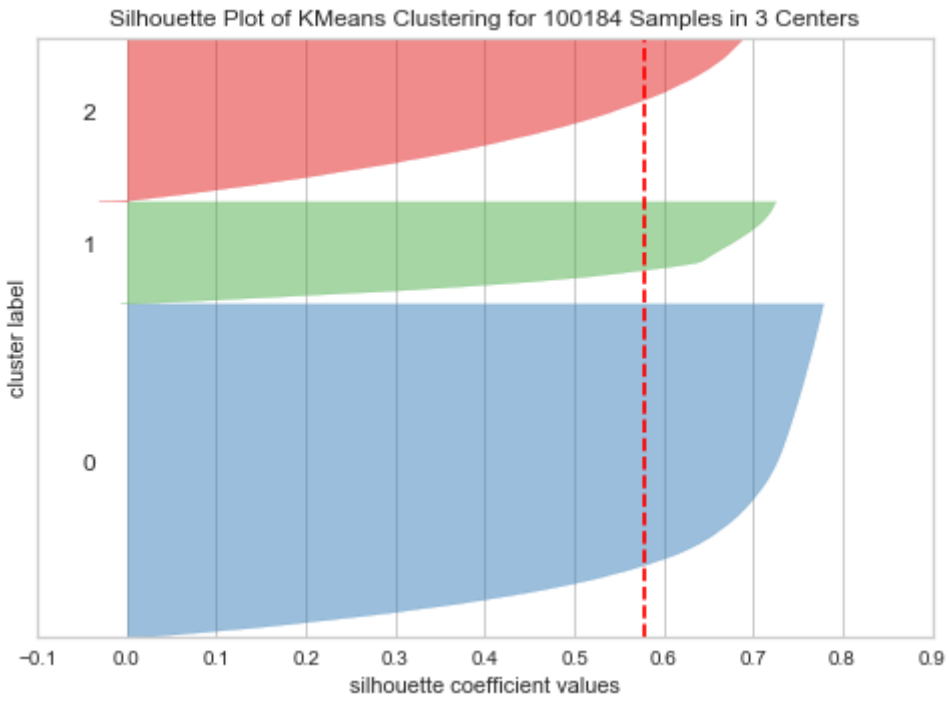
In [23]:
```
# Preprocess data using normalize function
from sklearn import preprocessing
df_normalized = preprocessing.normalize(df_analyze)
```

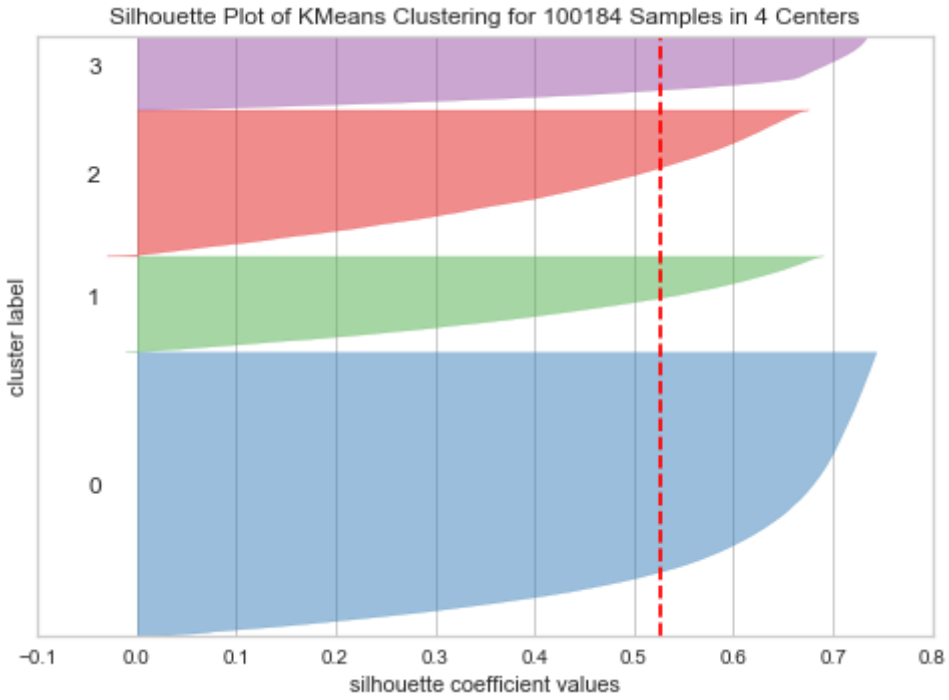
In [26]:
```
# Compute Scree plot for different number of clusters
from sklearn.cluster import KMeans
from sklearn import metrics
from scipy.spatial.distance import cdist

distortions = []
K = range(1,10)
for k in K:
    kmeanModel = KMeans(n_clusters=k).fit(df_normalized)
    kmeanModel.fit(df_normalized)
    distortions.append(sum(np.min(cdist(df_normalized, kmeanModel.cluste
r_centers_, 'euclidean'), axis=1)) / df_normalized.shape[0])

# Plot the elbow
plt.plot(K, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.show()
```

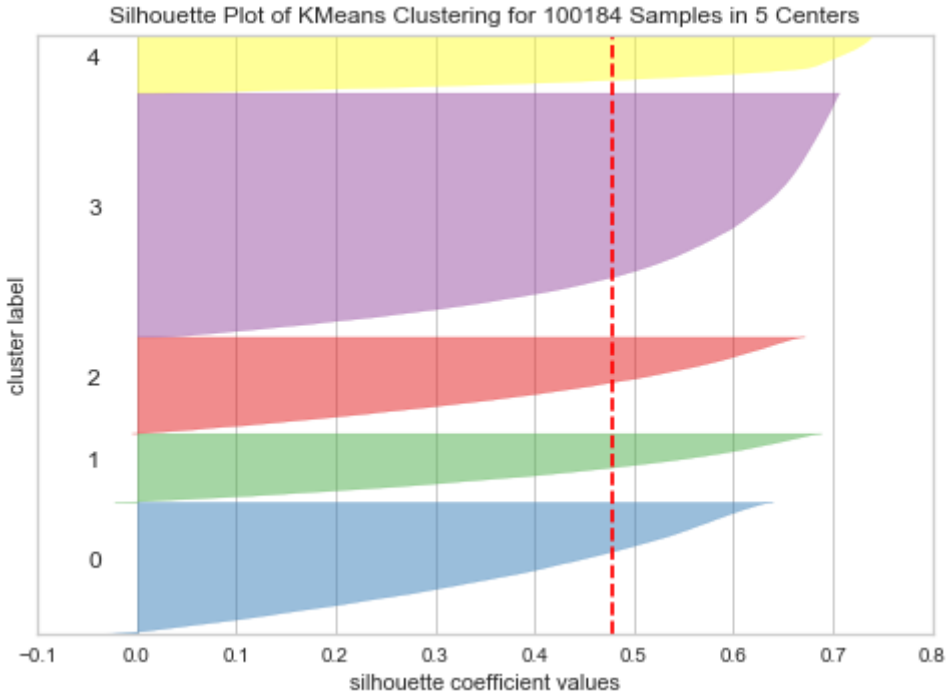


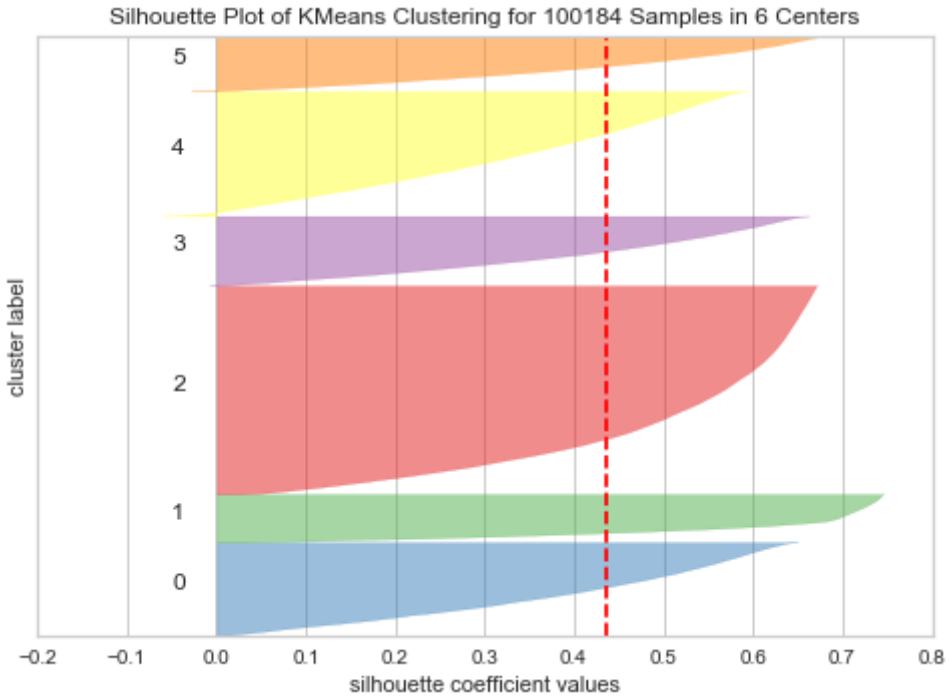
In [27]:
```
# Compute Silhoette Graph for different number of clusters to select
# optimal number of clusters
from sklearn.cluster import KMeans
from yellowbrick.cluster import SilhouetteVisualizer
for n_clusters in range(2, 9):
    model = SilhouetteVisualizer(KMeans(n_clusters))
    model.fit(df_normalized)
    model.poof()
```

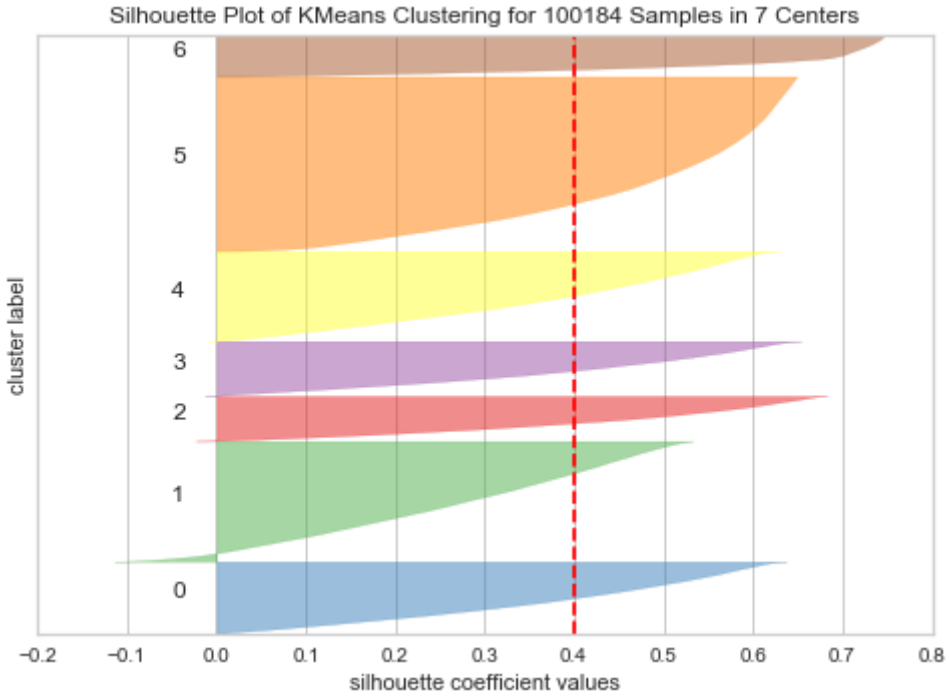

Silhouette Plot of KMeans Clustering for 100184 Samples in 2 Centers
cluster label
1
0
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
silhouette coefficient values

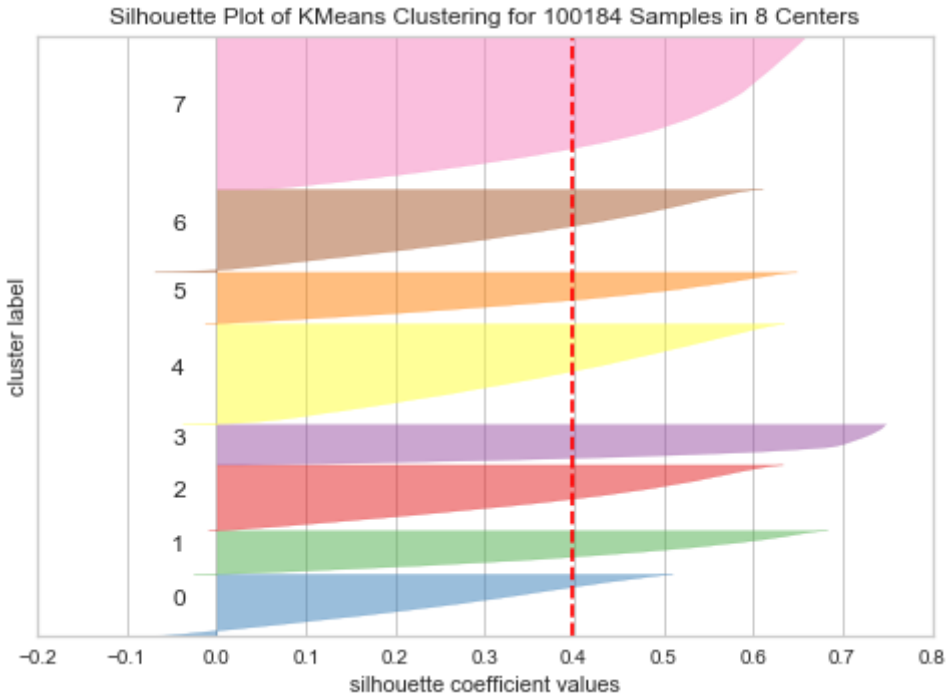

Silhouette Plot of KMeans Clustering for 100184 Samples in 3 Centers
cluster label
2
1
0
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
silhouette coefficient values


Silhouette Plot of KMeans Clustering for 100184 Samples in 4 Centers
cluster label
3
2
1
0
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
silhouette coefficient values

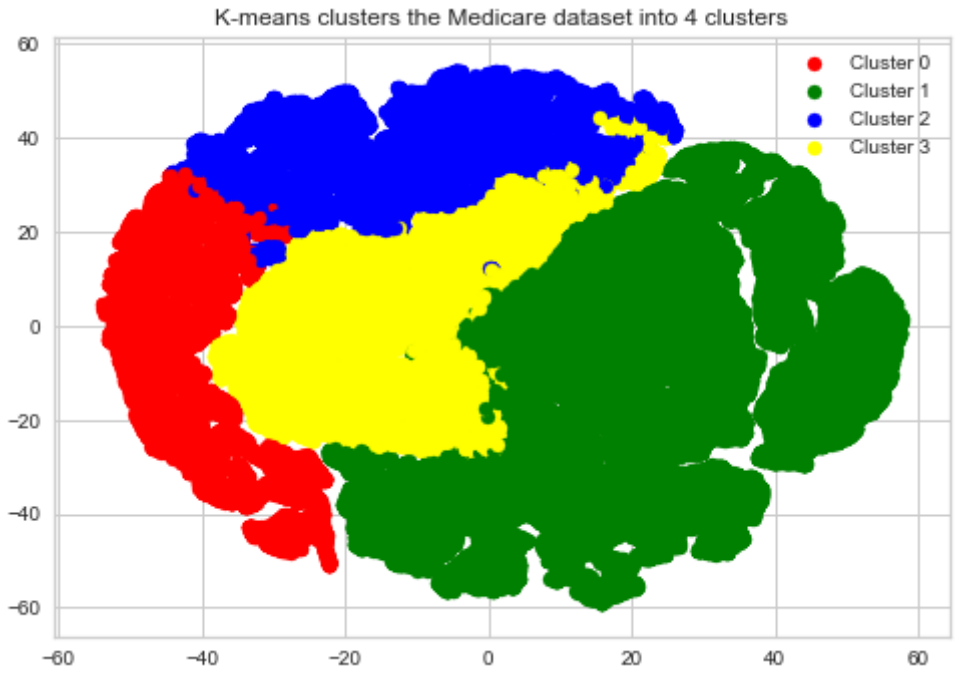

Silhouette Plot of KMeans Clustering for 100184 Samples in 5 Centers
cluster label
4
3
2
1
0
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
silhouette coefficient values


Silhouette Plot of KMeans Clustering for 100184 Samples in 6 Centers
cluster label
5
4
3
2
1
0
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
silhouette coefficient values


Silhouette Plot of KMeans Clustering for 100184 Samples in 7 Centers
cluster label
6
5
4
3
2
1
0
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
silhouette coefficient values


Silhouette Plot of KMeans Clustering for 100184 Samples in 8 Centers
cluster label
7
6
5
4
3
2
1
0
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
silhouette coefficient values

In [29]:
```
# Utlize TSNE to visualize data
from sklearn.cluster import KMeans
from sklearn.manifold import TSNE
import pylab as pl
num_of_clusters = 4
kmeans = KMeans(n_clusters = num_of_clusters)
kmeans.fit(df_normalized)

X = TSNE(n_components = 2).fit_transform(df_normalized)

for i in range(0, X.shape[0]):
    if kmeans.labels_[i] == 0:
        c1 = pl.scatter(X[i,0], X[i, 1], c='red')
    elif kmeans.labels_[i] == 1:
        c2 = pl.scatter(X[i,0], X[i, 1], c='green')
    elif kmeans.labels_[i] == 2:
        c3 = pl.scatter(X[i,0], X[i, 1], c='blue')
    elif kmeans.labels_[i] == 3:
        c4= pl.scatter(X[i,0], X[i, 1], c='yellow')

pl.legend([c1, c2, c3, c4], ['Cluster 0', 'Cluster 1', 'Cluster 2', 'Clu
ster 3'])
pl.title('K-means clusters the Medicare dataset into 4 clusters')
pl.show()
```



In [59]:
```
labels = pd.DataFrame(kmeans.labels_,columns = ['labels'])
df_describedata = pd.concat([df_analyze,labels],axis=1)

print(df_describedata[df_describedata['labels']==0.0].describe())
print(df_describedata[df_describedata['labels']==1.0].describe())
print(df_describedata[df_describedata['labels']==2.0].describe())
print(df_describedata[df_describedata['labels']==3.0].describe())
```

| | LINE_SRVC_CNT | BENE_DAY_SRVC_CNT | AVERAGE_SUBMITTED_CHRG_AMT | \ |
|---|---|---|---|---|
| count | 12285.000000 | 12285.000000 | 12285.000000 | |
| mean | 313.263655 | 302.425153 | 58.763626 | |
| std | 485.574402 | 446.391505 | 60.457091 | |
| min | 11.000000 | 11.000000 | 2.860000 | |
| 25% | 78.000000 | 73.000000 | 29.000000 | |
| 50% | 158.000000 | 151.000000 | 39.000000 | |
| 75% | 357.000000 | 349.000000 | 61.706683 | |
| max | 14925.000000 | 13187.000000 | 812.000000 | |

| | AVERAGE_MEDICARE_STANDARD_AMT | Gender_0 | Gender_1 | \ |
|---|---|---|---|---|
| count | 12285.000000 | 12285.000000 | 12285.000000 | |
| mean | 13.093735 | 0.198860 | 0.801140 | |
| std | 16.211376 | 0.399159 | 0.399159 | |
| min | 1.335938 | 0.000000 | 0.000000 | |
| 25% | 6.678151 | 0.000000 | 1.000000 | |
| 50% | 7.314721 | 0.000000 | 1.000000 | |
| 75% | 10.537655 | 0.000000 | 1.000000 | |
| max | 208.753372 | 1.000000 | 1.000000 | |

| | PLACEOFSERVICE0 | PLACEOFSERVICE1 | labels |
|---|---|---|---|
| count | 12285.000000 | 12285.000000 | 12285.0 |
| mean | 0.826943 | 0.173057 | 0.0 |
| std | 0.378312 | 0.378312 | 0.0 |
| min | 0.000000 | 0.000000 | 0.0 |
| 25% | 1.000000 | 0.000000 | 0.0 |
| 50% | 1.000000 | 0.000000 | 0.0 |
| 75% | 1.000000 | 0.000000 | 0.0 |
| max | 1.000000 | 1.000000 | 0.0 |

| | LINE_SRVC_CNT | BENE_DAY_SRVC_CNT | AVERAGE_SUBMITTED_CHRG_AMT | \ |
|---|---|---|---|---|
| count | 47410.000000 | 47410.000000 | 47410.000000 | |
| mean | 27.340688 | 26.933326 | 595.007566 | |
| std | 24.062081 | 23.532702 | 1216.891466 | |
| min | 11.000000 | 11.000000 | 54.000000 | |
| 25% | 14.000000 | 14.000000 | 185.000000 | |
| 50% | 20.000000 | 19.000000 | 315.000000 | |
| 75% | 31.000000 | 31.000000 | 603.565385 | |
| max | 590.000000 | 494.000000 | 80865.000000 | |

| | AVERAGE_MEDICARE_STANDARD_AMT | Gender_0 | Gender_1 | \ |
|---|---|---|---|---|
| count | 47410.000000 | 47410.000000 | 47410.000000 | |
| mean | 91.491511 | 0.183970 | 0.816030 | |
| std | 199.452863 | 0.387464 | 0.387464 | |
| min | 1.825000 | 0.000000 | 0.000000 | |
| 25% | 31.282619 | 0.000000 | 1.000000 | |
| 50% | 57.230483 | 0.000000 | 1.000000 | |
| 75% | 91.793803 | 0.000000 | 1.000000 | |
| max | 11906.130000 | 1.000000 | 1.000000 | |

| | PLACEOFSERVICE0 | PLACEOFSERVICE1 | labels |
|---|---|---|---|
| count | 47410.000000 | 47410.000000 | 47410.0 |
| mean | 0.587914 | 0.412086 | 1.0 |
| std | 0.492216 | 0.492216 | 0.0 |
| min | 0.000000 | 0.000000 | 1.0 |
| 25% | 0.000000 | 0.000000 | 1.0 |
| 50% | 1.000000 | 0.000000 | 1.0 |
| 75% | 1.000000 | 1.000000 | 1.0 |

```
max          1.000000         1.000000      1.0
       LINE_SRVC_CNT  BENE_DAY_SRVC_CNT  AVERAGE_SUBMITTED_CHRG_AMT  \
count   16039.000000       16039.000000                16039.000000
mean       69.562067          66.798741                   83.694082
std        77.375330          75.883517                   91.921778
min        11.000000          11.000000                    8.000000
25%        27.000000          25.000000                   33.000000
50%        41.000000          39.000000                   47.000000
75%        81.000000          78.000000                  102.000000
max      1331.000000        1317.000000                 1500.000000

       AVERAGE_MEDICARE_STANDARD_AMT      Gender_0      Gender_1  \
count                  16039.000000  16039.000000  16039.000000
mean                      17.288650      0.174076      0.825924
std                       19.722233      0.379186      0.379186
min                        1.344000      0.000000      0.000000
25%                        6.700909      0.000000      1.000000
50%                        8.215745      0.000000      1.000000
75%                       21.944087      0.000000      1.000000
max                      273.930688      1.000000      1.000000

       PLACEOFSERVICE0  PLACEOFSERVICE1   labels
count     16039.000000     16039.000000  16039.0
mean          0.823804         0.176196      2.0
std           0.380998         0.380998      0.0
min           0.000000         0.000000      2.0
25%           1.000000         0.000000      2.0
50%           1.000000         0.000000      2.0
75%           1.000000         0.000000      2.0
max           1.000000         1.000000      2.0
       LINE_SRVC_CNT  BENE_DAY_SRVC_CNT  AVERAGE_SUBMITTED_CHRG_AMT  \
count   24450.000000       24450.000000                24450.000000
mean       45.630634          44.487280                  138.322147
std        46.892540          46.309412                  140.951453
min        11.000000          11.000000                   20.000000
25%        17.000000          16.000000                   49.000000
50%        30.000000          29.000000                   95.000000
75%        57.000000          56.000000                  174.007143
max      1024.000000        1024.000000                 2458.487500

       AVERAGE_MEDICARE_STANDARD_AMT      Gender_0      Gender_1  \
count                  24450.000000  24450.000000  24450.000000
mean                      27.060357      0.176564      0.823436
std                       27.410830      0.381307      0.381307
min                        1.581818      0.000000      0.000000
25%                        8.163636      0.000000      1.000000
50%                       20.407664      0.000000      1.000000
75%                       33.352182      0.000000      1.000000
max                     1258.325723      1.000000      1.000000

       PLACEOFSERVICE0  PLACEOFSERVICE1   labels
count     24450.000000     24450.000000  24450.0
mean          0.767280         0.232720      3.0
std           0.422574         0.422574      0.0
min           0.000000         0.000000      3.0
25%           1.000000         0.000000      3.0
50%           1.000000         0.000000      3.0
```

| | | | |
|---|---|---|---|
| 75% | 1.000000 | 0.000000 | 3.0 |
| max | 1.000000 | 1.000000 | 3.0 |