In [2]:
```
# necessary libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [3]:
```
# dataset
dataset = pd.read_csv('gdp per capita.csv')

'''here we select only two years to verify for all countries in world'''
dataFrame = dataset[["2010","2011"]]


'''may some countires skipped bcoz they include null values'''

x = dataFrame.dropna()

# make array
X = x.values
```

In [5]:
```
'''count total number of values on each year
   mean values for both years for all countries
   min and max value is also described
   after which values is most in dataframe
   is all about the description'''
dataFrame.describe()
```

Out[5]:

	2010	2011
count	256.000000	258.000000
mean	14749.500789	16198.068908
std	22142.907842	24251.351938
min	234.235539	249.577979
25%	1655.486199	1866.674318
50%	5475.194363	5979.460608
75%	16722.986730	19868.319985
max	150737.892500	169016.196100

In [6]:
```
# normalizing the dataset

from sklearn import preprocessing

'''normalize the dataset with builtin library of python sklearn using preprocessing'''
normalized = preprocessing.normalize(X)
normalized
```

Out[6]:
```
array([[0.68533719, 0.72822588],
       [0.67826274, 0.73481947],
       [0.67665961, 0.73629598],
```

```
[0.6714834 , 0.7410196 ],
[0.61373543, 0.78951177],
[0.67814813, 0.73492524],
[0.68593919, 0.72765887],
[0.68697377, 0.72668222],
[0.65409859, 0.7564093 ],
[0.62863141, 0.77770338],
[0.67417374, 0.73857279],
[0.70688255, 0.70733094],
[0.71534502, 0.69877142],
[0.63977005, 0.76856638],
[0.67375786, 0.7389522 ],
[0.63071677, 0.77601311],
[0.68433938, 0.72916364],
[0.68178273, 0.73155472],
[0.67588431, 0.73700773],
[0.6530977 , 0.75727366],
[0.67160516, 0.74090925],
[0.65768798, 0.75329046],
[0.67721588, 0.73578438],
[0.71257155, 0.70159945],
[0.67314294, 0.7395124 ],
[0.67880249, 0.7343209 ],
[0.69457916, 0.71941628],
[0.7214797 , 0.69243559],
[0.64021892, 0.76819251],
[0.64856515, 0.76115915],
[0.6983114 , 0.7157941 ],
[0.59976786, 0.80017405],
[0.66104296, 0.75034805],
[0.64532232, 0.7639104 ],
[0.66282922, 0.74877061],
[0.67333909, 0.7393338 ],
[0.67196363, 0.74058414],
[0.64546495, 0.76378989],
[0.65851798, 0.752565  ],
[0.62965799, 0.77687246],
[0.69813385, 0.71596727],
[0.67824219, 0.73483844],
[0.65330958, 0.75709088],
[0.65375667, 0.75670484],
[0.65372579, 0.75673152],
[0.67272244, 0.73989494],
[0.67027763, 0.74211044],
[0.66539468, 0.74649174],
[0.68206994, 0.73128694],
[0.68231507, 0.73105823],
[0.71205029, 0.70212847],
[0.69163922, 0.72224317],
[0.67409717, 0.73864268],
[0.66486492, 0.74696361],
[0.67920359, 0.73394992],
[0.70214665, 0.71203236],
[0.68486597, 0.72866906],
[0.68469869, 0.72882625],
[0.63468321, 0.77277243],
[0.63595585, 0.77172544],
[0.67920311, 0.73395037],
[0.6556961 , 0.75502491],
[0.63131793, 0.77552413],
```

```
[0.67065012, 0.74177383],
[0.66523561, 0.7466335 ],
[0.68789471, 0.72581049],
[0.67901882, 0.73412086],
[0.61518351, 0.78838395],
[0.69396989, 0.72000403],
[0.64250559, 0.76628099],
[0.69385608, 0.7201137 ],
[0.67789006, 0.7351633 ],
[0.74731086, 0.66447459],
[0.67272249, 0.7398949 ],
[0.64118496, 0.76738638],
[0.68010884, 0.73311115],
[0.68172986, 0.73160399],
[0.69208469, 0.72181631],
[0.63345366, 0.77378063],
[0.68436809, 0.7291367 ],
[0.62657219, 0.77936339],
[0.64255299, 0.76624125],
[0.71838732, 0.69564334],
[0.74837876, 0.66327161],
[0.6214624 , 0.78344399],
[0.6241553 , 0.7813003 ],
[0.72360404, 0.69021532],
[0.7054732 , 0.7087366 ],
[0.68188029, 0.73146379],
[0.66217899, 0.7493457 ],
[0.70516872, 0.70903955],
[0.68226811, 0.73110207],
[0.68123466, 0.73206511],
[0.67952781, 0.73364975],
[0.67380167, 0.73891224],
[0.68026149, 0.73296951],
[0.68999114, 0.72381781],
[0.6792307 , 0.73392483],
[0.68045073, 0.73279383],
[0.6500844 , 0.75986201],
[0.65554445, 0.75515659],
[0.71394603, 0.70020073],
[0.66619283, 0.74577953],
[0.65076143, 0.75928226],
[0.75188861, 0.65929016],
[0.66977605, 0.74256316],
[0.68142352, 0.73188933],
[0.68199402, 0.73135775],
[0.64682107, 0.76264179],
[0.61027863, 0.79218684],
[0.67148296, 0.74102   ],
[0.67357106, 0.73912247],
[0.68194039, 0.73140776],
[0.6771734 , 0.73582348],
[0.69619478, 0.71785293],
[0.67794509, 0.73511255],
[0.61486182, 0.78863486],
[0.70540996, 0.70879954],
[0.61651549, 0.78734278],
[0.66495838, 0.74688042],
[0.65819054, 0.75285139],
[0.6840389 , 0.72944553],
[0.67703529, 0.73595055],
```

```
[0.62146851, 0.78343914],
[0.65463527, 0.75594488],
[0.63770347, 0.77028195],
[0.71105022, 0.70314123],
[0.65212283, 0.75811332],
[0.90836508, 0.41817805],
[0.68898327, 0.72477725],
[0.66454667, 0.74724676],
[0.66420868, 0.7475472 ],
[0.82672717, 0.56260304],
[0.66602791, 0.74592682],
[0.65835746, 0.75270543],
[0.66474463, 0.74707067],
[0.6528035 , 0.75752729],
[0.6563385 , 0.75446654],
[0.63680157, 0.77102773],
[0.6403962 , 0.76804473],
[0.68164755, 0.73168068],
[0.65038414, 0.75960547],
[0.61720868, 0.78679949],
[0.68181872, 0.73152118],
[0.6655994 , 0.74630921],
[0.63796522, 0.77006518],
[0.66414628, 0.74760265],
[0.69646163, 0.71759403],
[0.67822878, 0.73485082],
[0.67249648, 0.74010032],
[0.682685  , 0.7307128 ],
[0.64817846, 0.76148846],
[0.66810963, 0.74406284],
[0.64675422, 0.76269849],
[0.68545992, 0.72811036],
[0.57553121, 0.81777982],
[0.71652062, 0.69756591],
[0.67409613, 0.73864362],
[0.57535865, 0.81790123],
[0.73898663, 0.67372009],
[0.62166641, 0.78328212],
[0.65071873, 0.75931886],
[0.65631705, 0.7544852 ],
[0.66681674, 0.74522174],
[0.6560811 , 0.7546904 ],
[0.69453335, 0.71946051],
[0.68065542, 0.73260372],
[0.67757562, 0.73545311],
[0.68113202, 0.73216062],
[0.67574704, 0.7371336 ],
[0.67232729, 0.74025402],
[0.68507581, 0.72847178],
[0.65709576, 0.75380711],
[0.59530361, 0.80350085],
[0.5866419 , 0.80984645],
[0.65947563, 0.75172594],
[0.68268794, 0.73071005],
[0.66750987, 0.74460095],
[0.63910524, 0.7691193 ],
[0.64657487, 0.76285054],
[0.6536141 , 0.75682799],
[0.65460731, 0.75596909],
[0.67093659, 0.74151473],
```

```
       [0.68236952, 0.73100742],
       [0.62937365, 0.77710283],
       [0.67253348, 0.7400667 ],
       [0.66500845, 0.74683583],
       [0.69592192, 0.71811746],
       [0.69625579, 0.71779376],
       [0.63315257, 0.77402702],
       [0.66383665, 0.7478776 ],
       [0.65285374, 0.75748399],
       [0.68239573, 0.73098295],
       [0.63310709, 0.77406422],
       [0.67008087, 0.74228811],
       [0.59790748, 0.80156512],
       [0.67388364, 0.73883749],
       [0.67649401, 0.73644813],
       [0.62998212, 0.77660964],
       [0.69975816, 0.71437981],
       [0.68115136, 0.73214262],
       [0.65915903, 0.75200357],
       [0.63747484, 0.77047117],
       [0.66743172, 0.744671  ],
       [0.67441961, 0.73834828],
       [0.72466619, 0.68910007],
       [0.64422732, 0.76483407],
       [0.67545428, 0.73740187],
       [0.71156452, 0.70262076],
       [0.6754574 , 0.73739901],
       [0.6456737 , 0.76361343],
       [0.65595772, 0.75479764],
       [0.70679299, 0.70742043],
       [0.67458075, 0.73820106],
       [0.68355403, 0.72989992],
       [0.65644698, 0.75437216],
       [0.67984331, 0.73335739],
       [0.66333205, 0.74832519],
       [0.96827187, 0.24989914],
       [0.6936338 , 0.72032781],
       [0.67158211, 0.74093013],
       [0.63596377, 0.77171891],
       [0.63753898, 0.7704181 ],
       [0.67289337, 0.73973949],
       [0.67876289, 0.7343575 ],
       [0.66254655, 0.74902074],
       [0.61781692, 0.78632198],
       [0.66347468, 0.74819874],
       [0.65253617, 0.75775758],
       [0.71684763, 0.69722986],
       [0.66120612, 0.75020428],
       [0.67649401, 0.73644813],
       [0.6754574 , 0.73739901],
       [0.65914295, 0.75201767],
       [0.69618441, 0.71786299],
       [0.68516232, 0.72839041],
       [0.63879081, 0.76938046],
       [0.68925684, 0.72451709],
       [0.70280892, 0.71137868],
       [0.63909037, 0.76913166],
       [0.64127953, 0.76730735],
       [0.64423667, 0.7648262 ],
       [0.69685459, 0.71721244],
```

```
       [0.64740876, 0.76214296],
       [0.70993553, 0.70426667],
       [0.78375706, 0.62106752],
       [0.71519667, 0.69892326],
       [0.65383081, 0.75664078],
       [0.66675193, 0.74527972],
       [0.67439041, 0.73837495],
       [0.67182525, 0.74070968],
       [0.6476236 , 0.76196042],
       [0.69663487, 0.71742586],
       [0.67899036, 0.73414719],
       [0.6649697 , 0.74687034],
       [0.65512719, 0.75551861]])
```

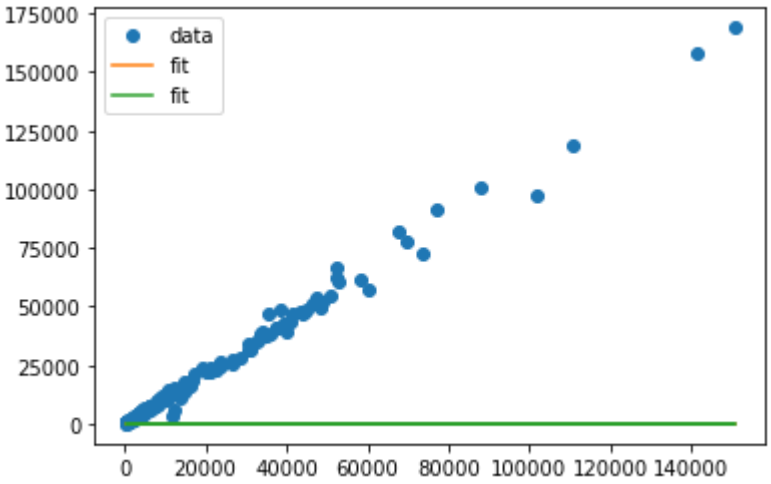
In [15]:

```
#curve_fit() method

#Import curve fitting package from scipy
from scipy.optimize import curve_fit


xdata = x["2010"]
ydata = x["2011"]

'''guassian function that use in curve fit this is chunk of function that use in future
    call this method to make curve fit that is builtin function of scipy'''
def guass(x, A, B):
    y = A*np.exp(-1*B*x**2)
    return y
param, cov = curve_fit(guass,xdata, ydata)

A = param[0]
B = param[1]

#     fit y
y = guass(x, A, B)

#     plotting
plt.plot(xdata, ydata, 'o', label='data')
plt.plot(xdata, y, '-', label='fit')
plt.legend()
```

```
E:\Files\lib\site-packages\scipy\optimize\minpack.py:833: OptimizeWarning: Covariance of
the parameters could not be estimated
  warnings.warn('Covariance of the parameters could not be estimated',
```

Out[15]:

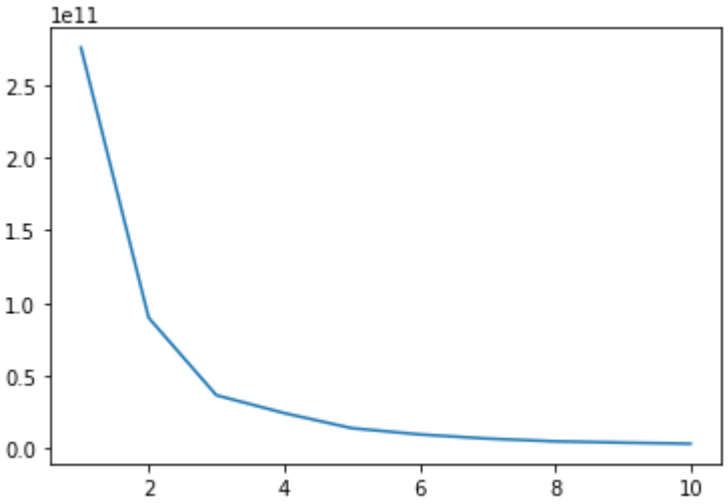
```
<matplotlib.legend.Legend at 0x2873bdeaca0>
```



In [18]:
```
# prediction for future values
from sklearn.cluster import KMeans

'''how many clusters make from this dataset'''
wcss = []
for i in range(1,11):
    kmeans = KMeans(n_clusters = i, init = "k-means++")
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
```

```
E:\Files\lib\site-packages\sklearn\cluster\_kmeans.py:881: UserWarning: KMeans is known
to have a memory leak on Windows with MKL, when there are less chunks than available thr
eads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

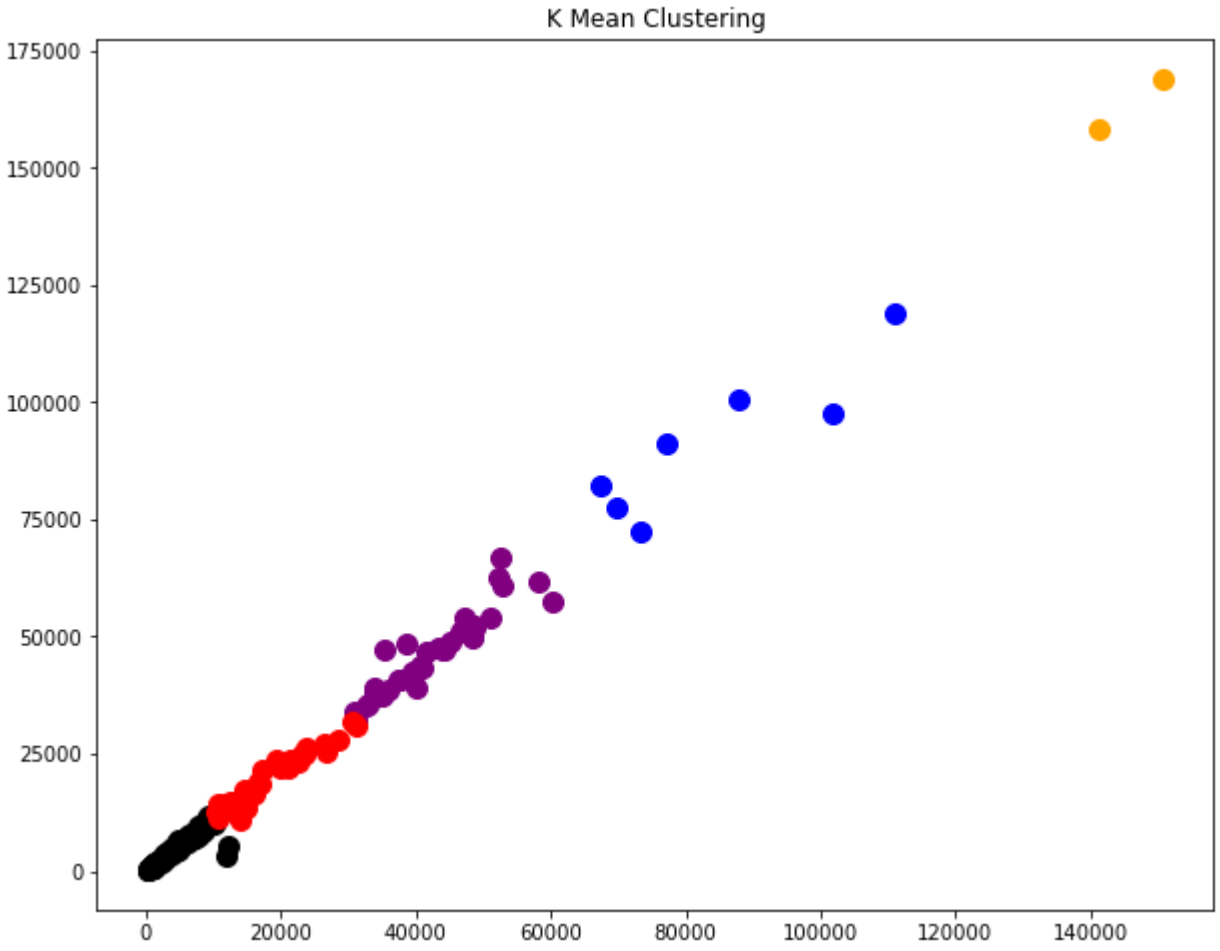
In [28]:
```
plt.plot(range(1,11),wcss)
plt.show()
```



In [21]:
```
'''from above elbow chart we easily make 5 different clusters'''

kmeans = KMeans(n_clusters = 5, init = "k-means++")
y_kmeans = kmeans.fit_predict(X)
y_kmeans
```

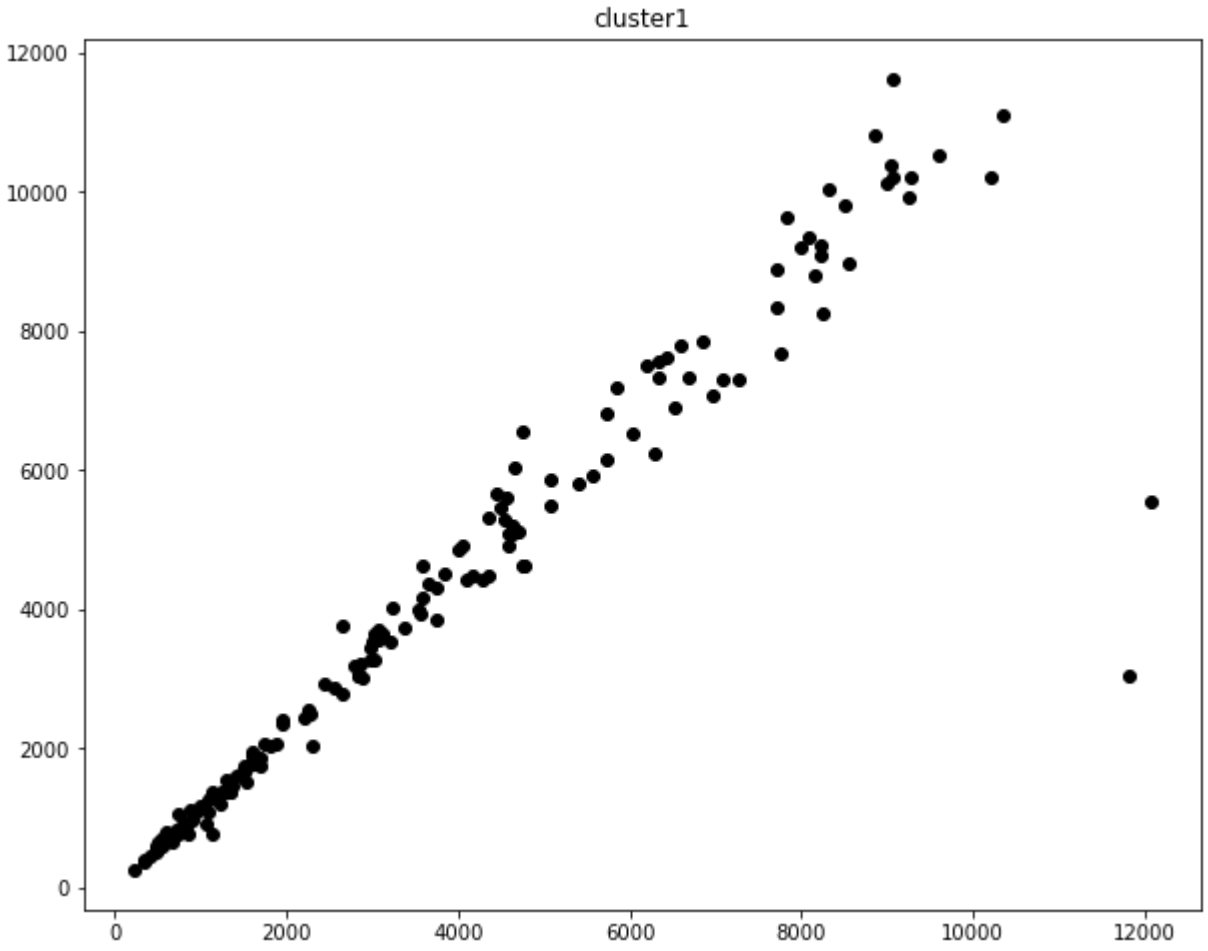
Out[21]:
```
array([3, 0, 0, 0, 0, 0, 1, 0, 1, 3, 0, 0, 3, 1, 1, 0, 0, 1, 0, 0, 0, 0,
       3, 3, 0, 0, 0, 4, 0, 3, 3, 1, 0, 0, 0, 1, 3, 4, 3, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 4, 1, 3, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 3, 0, 0,
       1, 0, 3, 3, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 3, 3, 0,
       1, 0, 3, 0, 1, 1, 0, 0, 3, 0, 3, 0, 0, 0, 0, 0, 0, 4, 0, 1, 0, 0,
       1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 3, 3, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 2, 0, 0, 0, 0, 0, 3, 4, 3, 1, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       3, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 4, 0, 0, 1,
       1, 3, 3, 0, 0, 0, 0, 0, 0, 3, 0, 3, 3, 0, 0, 0, 1, 4, 0, 3, 0, 0,
       3, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 3, 0, 0, 3, 3, 1, 0, 3, 0, 3,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 3, 0, 0, 0, 0, 0, 3,
       1, 0, 0, 3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

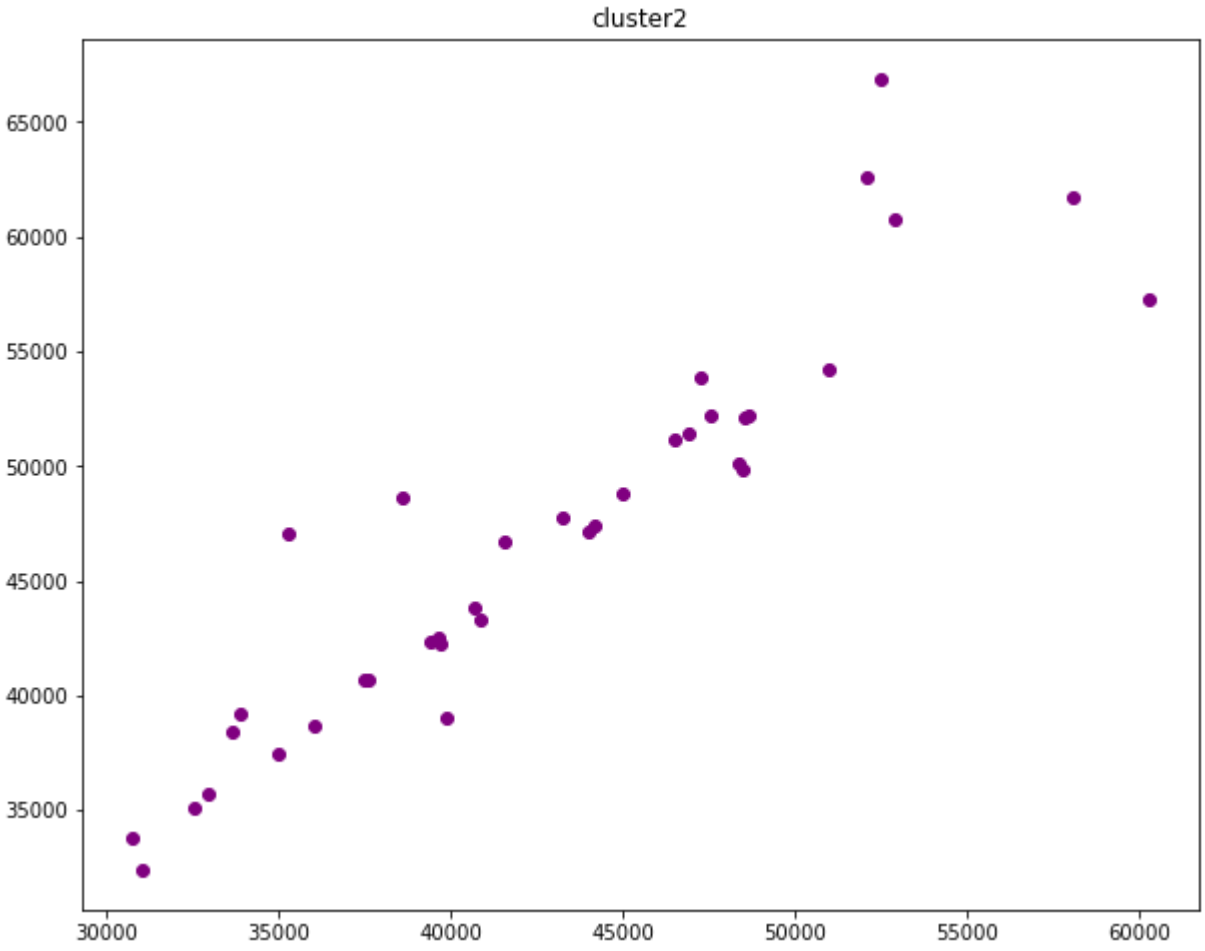
In [27]:
```
'''as the prediction above show which record belong to which cluster is mentioned in ab


plt.figure(figsize = (10,8))
plt.title("K Mean Clustering")
plt.scatter(X[y_kmeans==0,0], X[y_kmeans==0,1], s=100, c="black")
plt.scatter(X[y_kmeans==1,0], X[y_kmeans==1,1], s=100, c="purple")
plt.scatter(X[y_kmeans==2,0], X[y_kmeans==2,1], s=100, c="orange")
plt.scatter(X[y_kmeans==3,0], X[y_kmeans==3,1], s=100, c="red")
plt.scatter(X[y_kmeans==4,0], X[y_kmeans==4,1], s=100, c="blue")
plt.show()
```

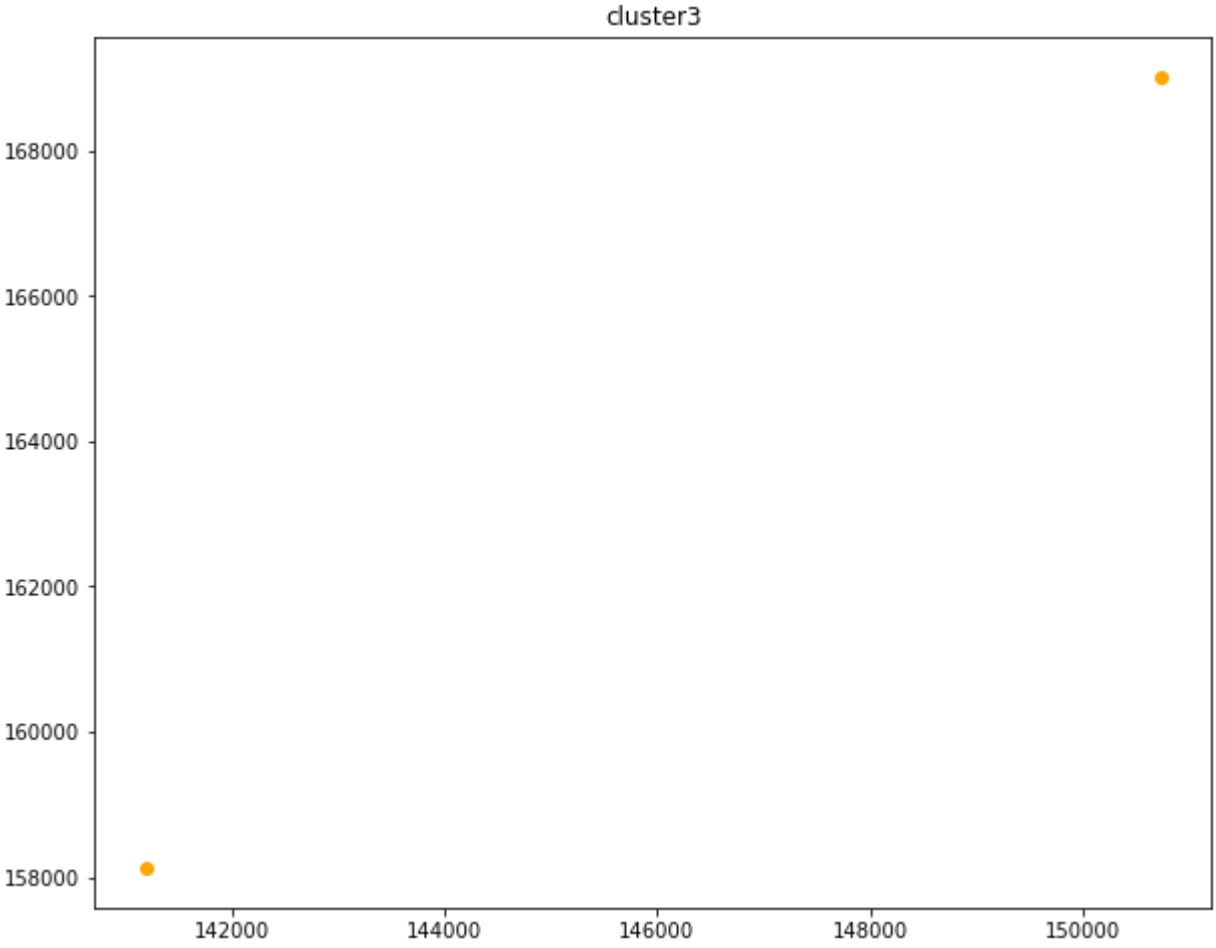


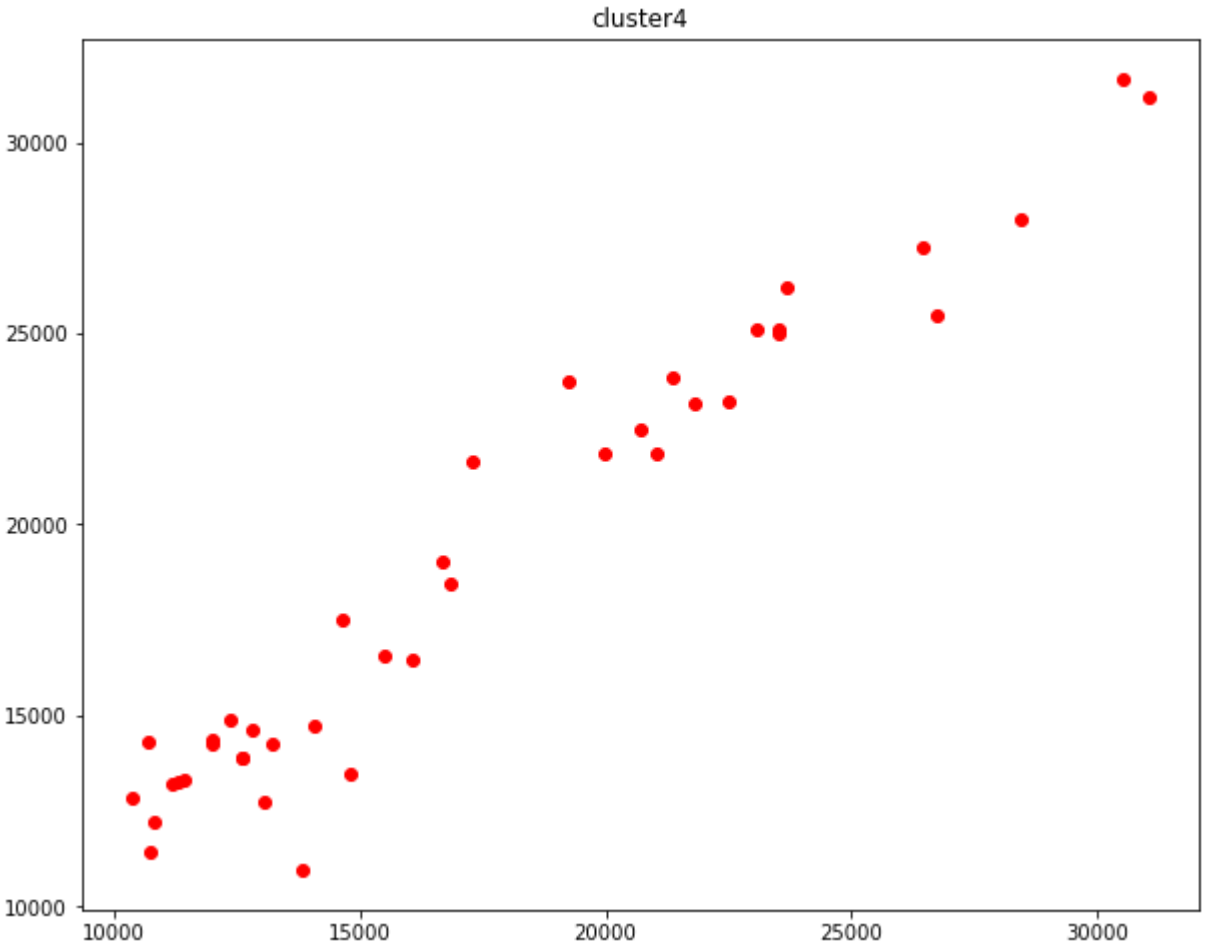
In [26]:
```
# Interpretation of the results.
'''black show most elements of dataset,
   red show after black in rich quantity
```

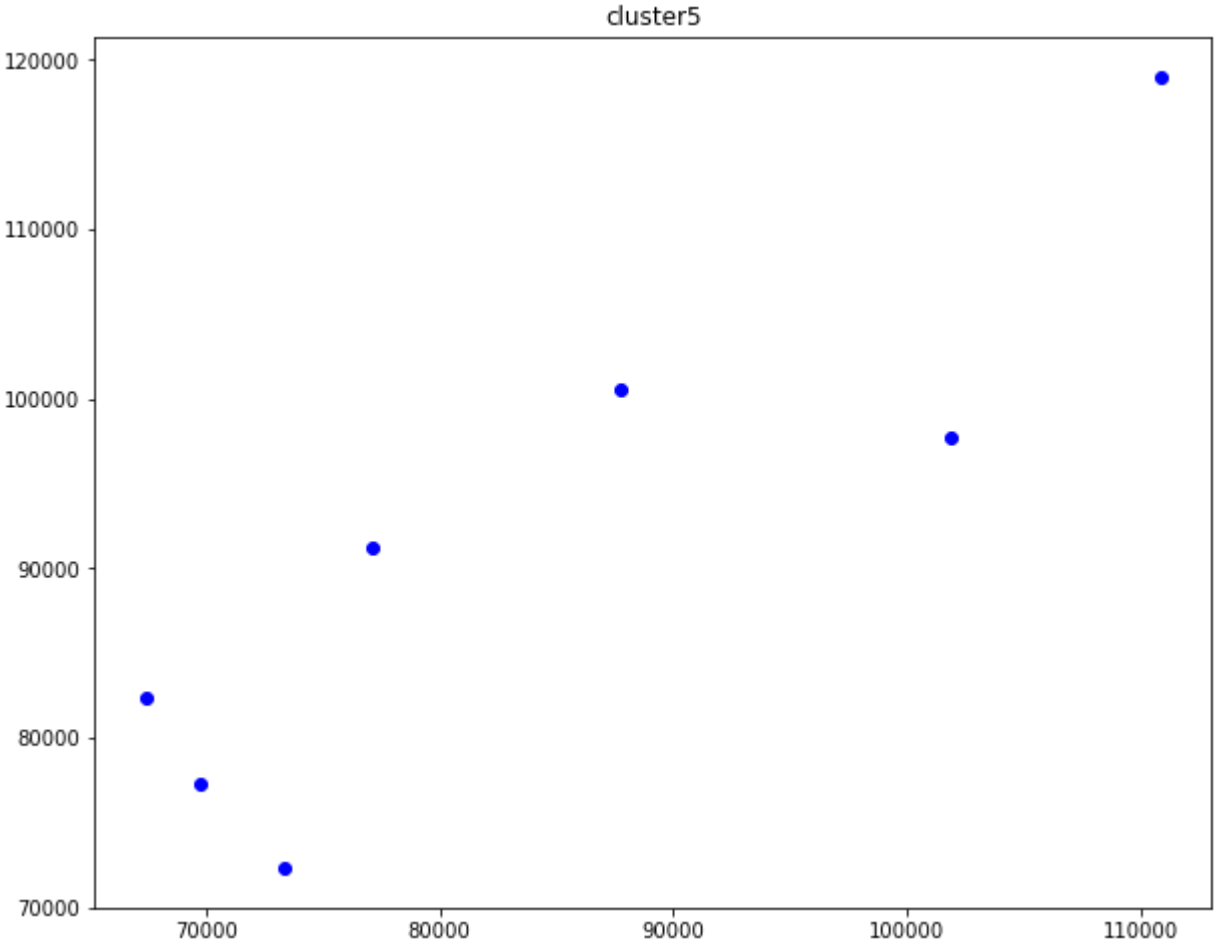
```
   purple placed after red
   then blue and orange'''
clor = ["black","purple", "orange","red","blue"]
for i in range(5):
    plt.figure(figsize=(10,8))
    plt.title(f"cluster{i+1}")
    plt.scatter(X[y_kmeans==i,0], X[y_kmeans==i,1], color=clor[i])
    plt.show()
```




cluster2
65000
60000
55000
50000
45000
40000
35000
30000
35000
40000
45000
50000
55000
60000


cluster3
168000
166000
164000
162000
160000
158000
142000
144000
146000
148000
150000


cluster4
30000
25000
20000
15000
10000
10000
15000
20000
25000
30000


cluster5
120000
110000
100000
90000
80000
70000
70000
80000
90000
100000
110000

In [ ]: