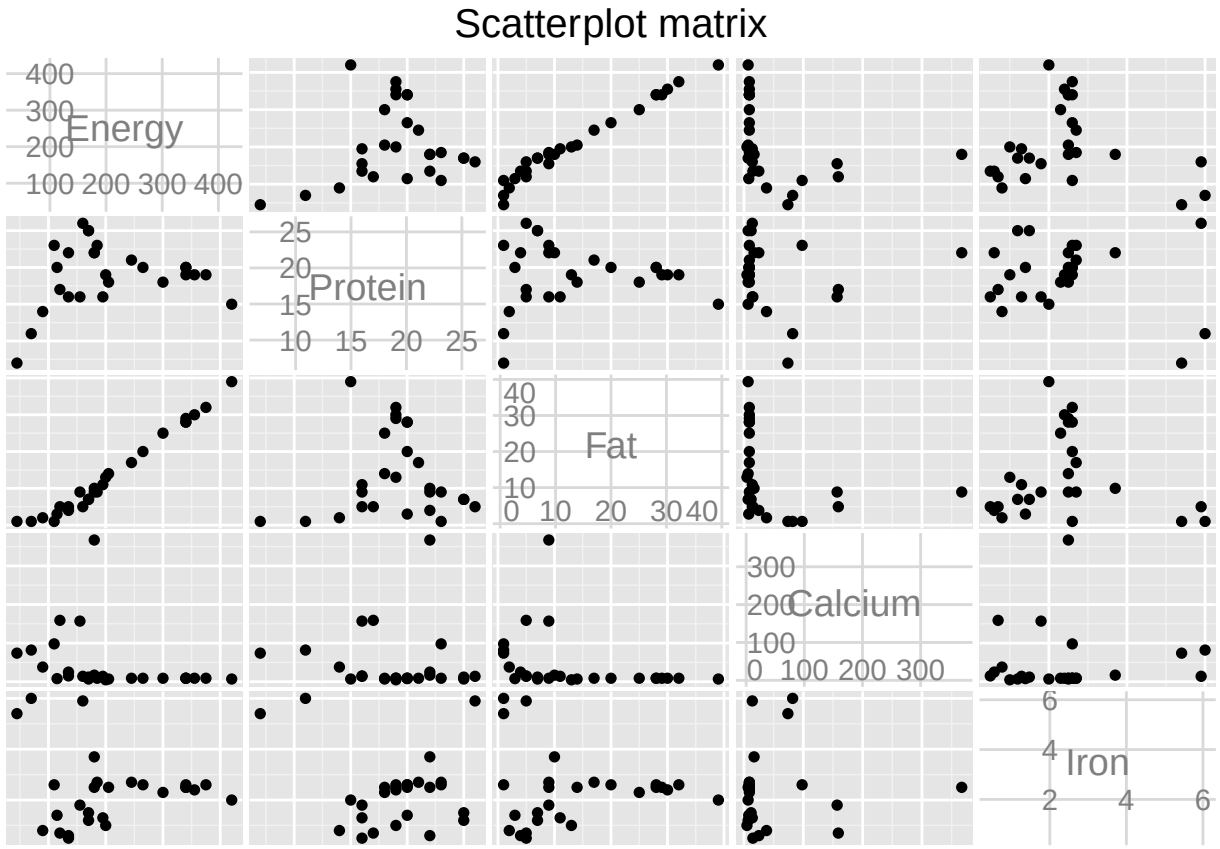
Data Mining - Lab 1 - Cluster Analysis

Gustav Sternelöv

February, 3, 2016

Simple K-means

I start with making a scatterplot matrix which includes all variables except *names*. The latter is excluded because it just gives the names of the respective products. An alternative could have been to create a categorical variable since the names of the products indicates quite well what kind of food it contains. However, K-means is not a good algorithm for categorical data so the variable would still not have been so interesting to include in this particular analysis.



The chosen attributes are *Energy*, *Protein*, *Fat* and *Iron*. This is motivated by the patterns visualised in the scatterplot matrix. My conclusion is that all of the variables but *calcium* seem to be interesting to include. For *calcium* most of the values are very close to each other apart from some outliers. It is therefore interpreted as not being the most interesting variable to include. The other variables have values that are more spread out in different groups and these groups might be possible to investigate closer with a cluster analysis.

Seed 10

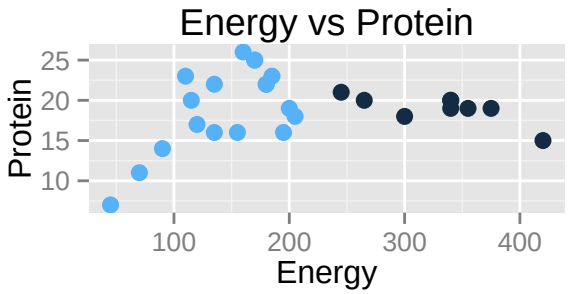
The initial clusters centroids are held fixed with the seed value *10* and two different cluster analysis are performed. In the first case are the data points divided into two clusters and in the second case into four different clusters.

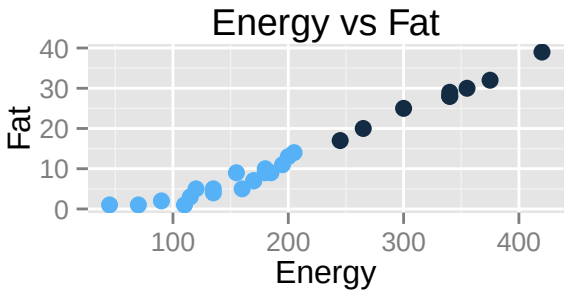
Two clusters

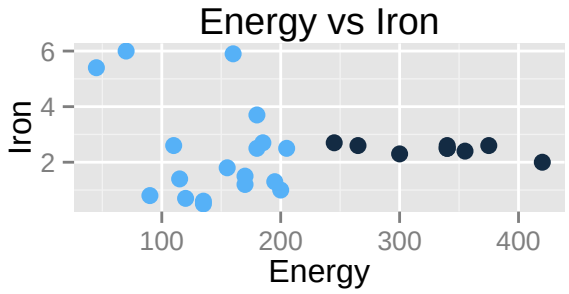
The number of iterations is two and the within cluster *SSE* is 3.99. The first cluster contains *1/3* of the data points and the second cluster the remainder of the data points. How the initial cluster centroids has changed can be seen in the output below.

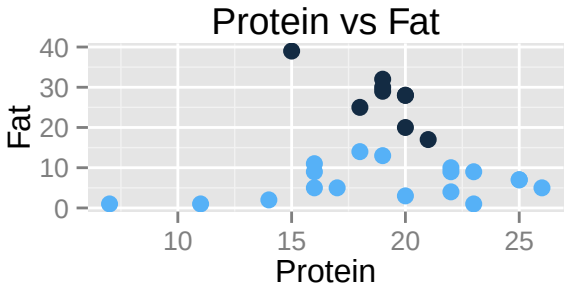
```
##
## kMeans
## ======
##
## Number of iterations: 2
## Within cluster sum of squared errors: 3.9886919330126585
##
## Initial starting points (random):
##
## Cluster 0: 340,20,28,2.6
## Cluster 1: 170,25,7,1.5
##
## Missing values globally replaced with mean/mode
##
## Final cluster centroids:
##                          Cluster#
## Attribute    Full Data          0          1
##                 (27.0)      (9.0)     (18.0)
## ========================================
## Energy        207.4074   331.1111   145.5556
## Protein             19         19         19
## Fat            13.4815    27.5556     6.4444
## Iron            2.3815     2.4667     2.3389
```

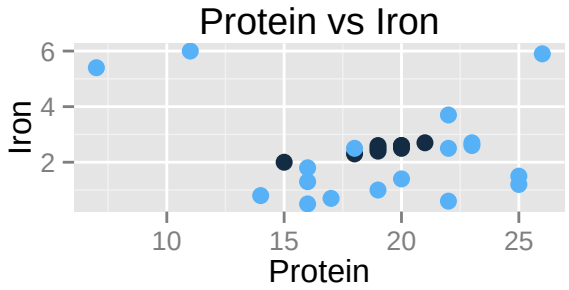
Another way to present the obtained clusters are through visualisation. How the data points for each variable are clustered into the respective clusters is shown by the graphs below.

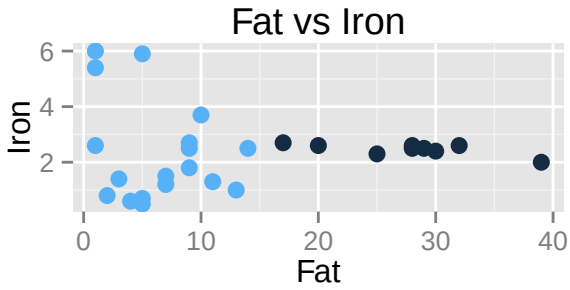












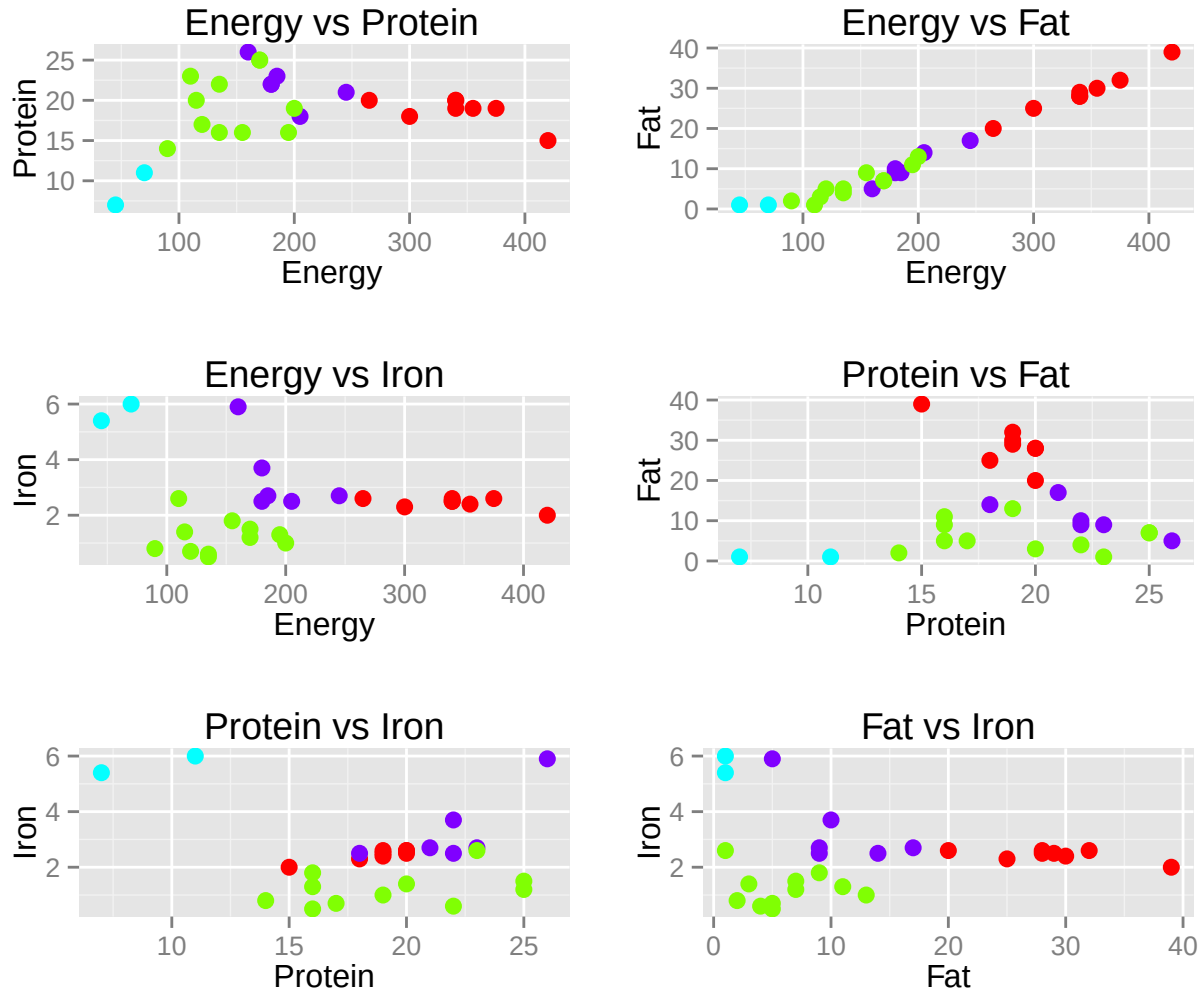
In general are the clusters very well separeted and dissimilar.The first cluster, light blue points, includes food with lower levels of energy and fat and the second cluster, dark blue points, includes food with high levels of energy and fat.
That the clusters only overlap in one of the plots, *"Protein vs Iron"*, shows that they are well separated and dissimilar. The problem with the clusters is instead that they are too general. Even though the two clusters are well separated the members inside the clusters are not always very similar. It could therefore be interesting to examine if better clusters are given if K is increased from two to four.

Four clusters

For the clustering with K=4 the number of iterations is six and the *SSE* is 1.56. Information about how many of the points that are clustered into each cluster and how the initial cluster centroids has changed is given in the output below.

```
##
## kMeans
## ======
##
## Number of iterations: 6
```

```
## Within cluster sum of squared errors: 1.563125484429238
##
## Initial starting points (random):
##
## Cluster 0: 340,20,28,2.6
## Cluster 1: 170,25,7,1.5
## Cluster 2: 90,14,2,0.8
## Cluster 3: 180,22,9,2.5
##
## Missing values globally replaced with mean/mode
##
## Final cluster centroids:
##                           Cluster#
## Attribute    Full Data          0          1          2          3
##                 (27.0)      (8.0)     (11.0)      (2.0)      (6.0)
## ===============================================================
## Energy        207.4074    341.875        145       57.5      192.5
## Protein             19      18.75    19.3636          9         22
## Fat            13.4815     28.875     6.0909          1    10.6667
## Iron            2.3815     2.4375     1.2182        5.7     3.3333
```



In general it seems like more natural clusters are obtained when data is divided into four clusters instead of two. The similarity is now higher within the clusters and the dissimilarity between the clusters is more evident. In some cases does the clusters overlap a bit, but as a whole is the similarity inside the clusters and the dissimilarity between the clusters relatively clear.

Seed 28

The seed is changed to the value *28*. This means that the starting centroids will have changed for the clusters presented below. The starting points are chosen "*randomly*" and each "*random*" choice of starting points is connected to a specific seed value. To try different starting points is a wise choice since the K-means algorithm often finds a local minima. Another local minima with a lower *SSE* might be found if the initial centroids are shifted.

Two clusters

For the new *SimpleKMeans* model with *K=2* the algorithm needed six iterations before it reached the local minima. The within cluster *SSE* is 3.99. The clusters found are the exact same clusters as those for the seed value *10*. The only difference is that for the new set of initial cluster centroids six iterations were needed instead of two before the local minima was found.

```
##
## kMeans
## ======
##
## Number of iterations: 6
## Within cluster sum of squared errors: 3.9886919330126585
##
## Initial starting points (random):
##
## Cluster 0: 155,16,9,1.8
## Cluster 1: 90,14,2,0.8
##
## Missing values globally replaced with mean/mode
##
## Final cluster centroids:
##                            Cluster#
## Attribute    Full Data          0          1
##                 (27.0)      (9.0)     (18.0)
## ==========================================
## Energy        207.4074   331.1111   145.5556
## Protein             19         19         19
## Fat            13.4815    27.5556     6.4444
## Iron            2.3815     2.4667     2.3389
```

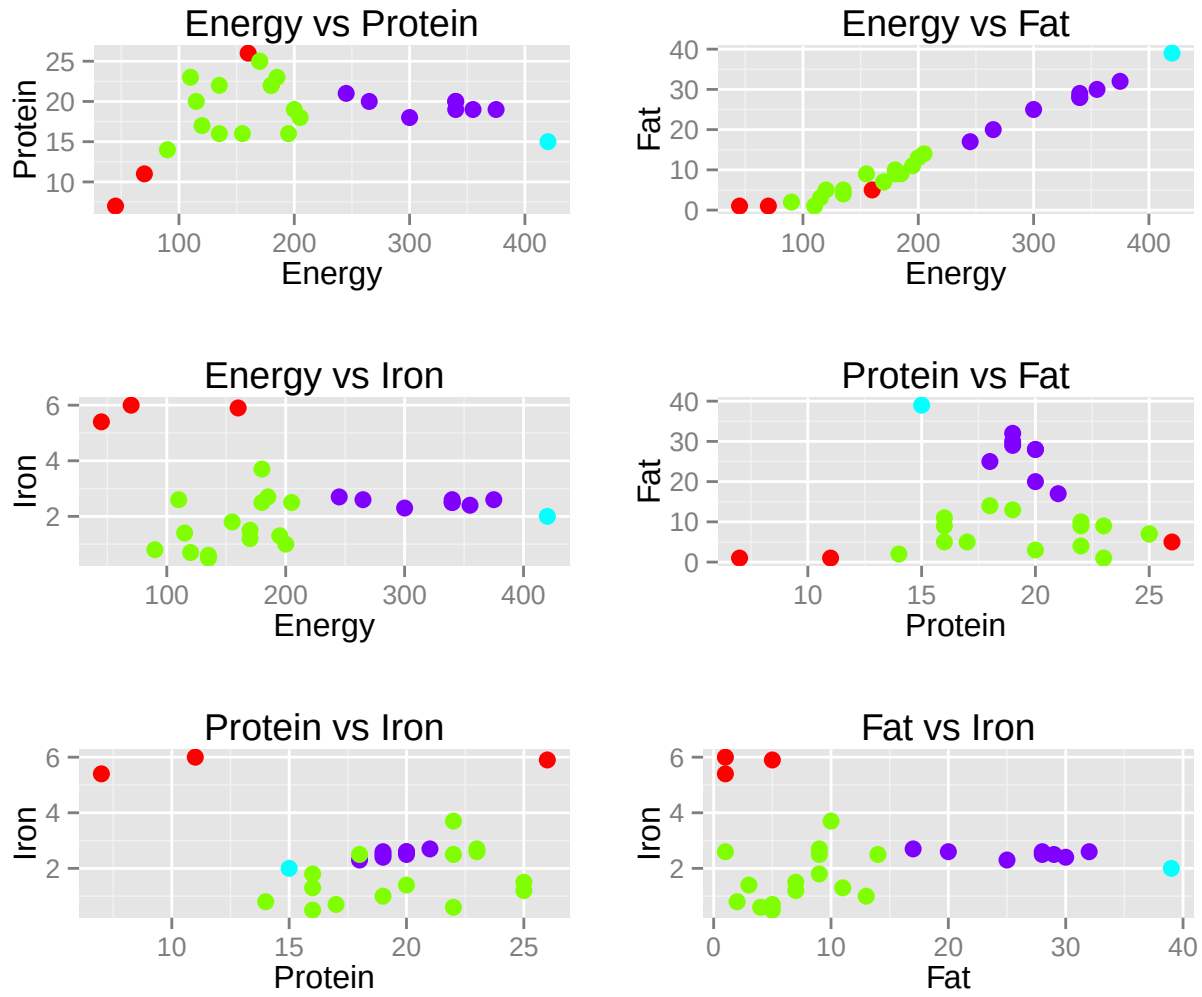
A visualisation of the clusters is not included as these clusters are identical to those presented for *K=2* and seed *10*.

Four clusters

A *SimpleKMeans* with *K=4* and the seed *28* has a higher number of iterations, eight versus six, and a higher *SSE* value, 2.06 versus 1.56, than the model with seed *10*. The new set of starting points led into another local minima which seem to give worse clusters than the previuos model with *K=4*. The results is thought to be worse because of the higher *SSE* for the latter model.

```
##
## kMeans
## ======
##
## Number of iterations: 8
## Within cluster sum of squared errors: 2.0634372954170264
##
## Initial starting points (random):
##
## Cluster 0: 155,16,9,1.8
## Cluster 1: 90,14,2,0.8
## Cluster 2: 420,15,39,2
## Cluster 3: 340,20,28,2.6
##
## Missing values globally replaced with mean/mode
##
## Final cluster centroids:
##                            Cluster#
## Attribute    Full Data          0          1          2          3
```

```
##                (27.0)      (3.0)     (15.0)      (1.0)      (8.0)
## ============================================================================
## Energy       207.4074    91.6667   156.3333        420        320
## Protein            19    14.6667    19.8667         15       19.5
## Fat           13.4815     2.3333     7.2667         39     26.125
## Iron           2.3815     5.7667     1.6533          2      2.525
```



The four clusters obtained with seed *28* are in general well separated and looks like a relatively natural partitioning. The two big clusters, the one with green dots and the one with purple dots, are similar within and clearly dissimilar to the other clusters. However, compared to the *SimpleKMeans* model with *K=4* and seed *10* these clusters appears to be a little bit worse. One of the points in the red cluster is not so similar to the two others for the variables *Energy* and *Protein* and should perhaps in an optimal case have been assigned into another cluster. Another difference with the clusters recieved with seed *28* is that they in practice just are three clusters since one of the clusters only consists of one point. With the new starting points this point is considered to be an outlier. To conclude, the result given with seed *10* were better than the result given with the new starting centroids.

Seed 10 and K=4

Of the three different clusters presented above I have chosen to examine the cluster with *K=4* and seed *10* more closely.
¤ The red cluster contains food with a high amount of energy and fat. It represents typical meat products like pork, beef and ham.
¤ The turquoise cluster contains food with a very low amount of protein, energy and fat and a high amount of iron. It seem to represent food containing clams since the two products in the cluster are *canned clams* and *raw clams*.
¤ The green cluster contains food with a low amount of energy, fat and iron. The foods that has these characteristics are different fish and seafood products and chicken, so this is a seafood and chicken cluster.
¤ The purple cluster is dissimilar to the others by having a moderately low amount of fat and energy and a moderately high amount of iron. It represents some more uncommon meat products like for example beef heart, beef tongue and veal cutlet.

MakeDensityBasedClusters

The objective here is to investigate the effect of different values for the setting *min standard deviation* when a *MakeDensityBasedClusters* clustering is performed. Two examples of *MakeDensityBasedClusters* clustering on the *SimpleKMeans* cluster with seed *10* and *K=4* are presented where the min standard deviation is set to five in the first example and to 100 in the second example.
Up to this part of the lab have I used **R** to make the computations and visualizations. However, the *MakeDensityBasedClusters* function is not available in the *Weka* **R** package so from here on have I used *Weka* instead of **R**.

Min standard deviation = 5

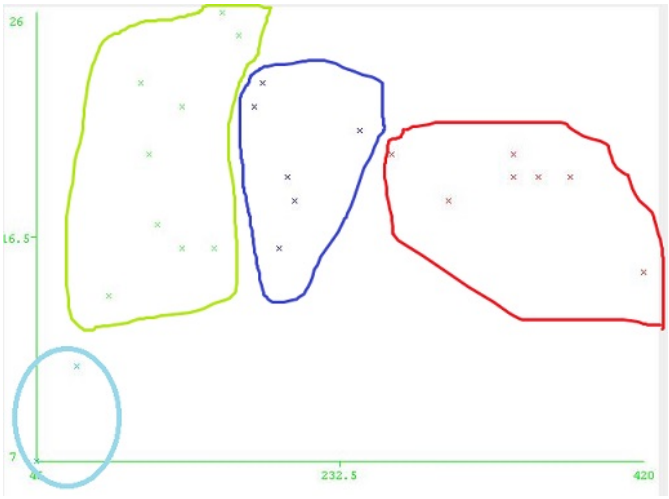
The number of clusters is unchanged and the number of objects in each cluster is similar, but the way the objects are assigned to the respective clusters have changed a little bit.

```
Clustered Instances

0       8 ( 30%)
1      10 ( 37%)
2       2 (  7%)
3       7 ( 26%)


Log likelihood: -14.41518
```

That some objects, as mentioned above, has changed clusters is visualised with a plot over *Energy* versus *Protein* for the new clusters.



Compared to the *SimpleKMeans* cluster with *K=4* and seed *10* there is no overlapping when looking at *Energy* versus *Protein*. In the earlier clustering there were some overlapping objects who were very similar to objects not belonging to its cluster.

Min standard deviation = 100

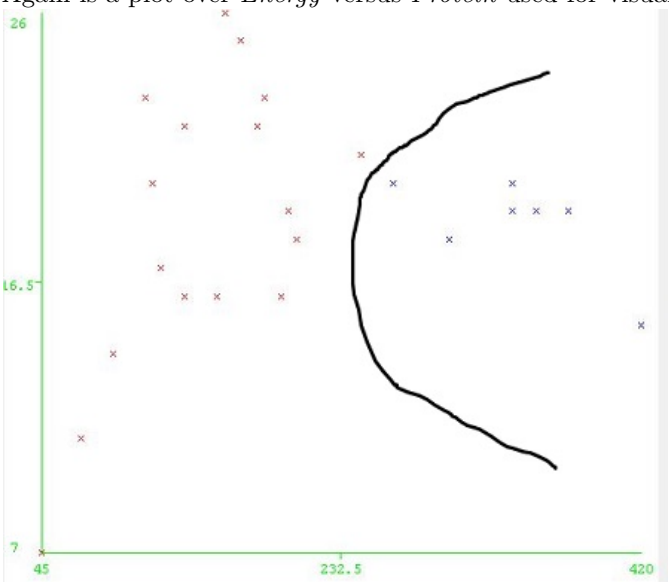
With this setting the cluster *0* is unchanged and the clusters *1*, *2* and *3* are merged together to one cluster. The result of this is a clustering that consists of two different clusters.

```
Clustered Instances

0       8 ( 30%)
1      19 ( 70%)


Log likelihood: -22.67427
```

Again is a plot over *Energy* versus *Protein* used for visualising the new clustering.



The result is a clustering that is rather similiar, but not exactly the same, to the *SimpleKMeans* with *K=2* that was presented earlier in the report.

As the results above has shown it is discoverad that a high value for the *min standard deviation* setting results in fewer clusters. That is a logical interpretation of this setting since a high standard deviation means that the objects within a cluster are allowed to be less similar. In the reverse case is it also logical that a low value for the min standard deviation gives more and smaller clusters where the objects inside each cluster are more similar.