Student’s name: Duong Nguyen

PID: 1646288

Professor: Dr. Christoph Eick

ASSIGNMENT 2

***QUESTION 0 – 3:*** The code is in the .R file as well as SourceCode.txt

***QUESTION 4:***

Learn Linear model that predicts the 9th attribute for zpima dataset.

Coefficients:

(Intercept) times\_pregnant glucose\_concentration

0.348958 0.069762 0.196761

blood\_pressure skin\_thickness serum\_insulin

-0.014012 0.001427 -0.007562

bmi diabetes\_function age

0.099092 0.042813 0.024695

By looking at the result using function, we can see that that is clear that there does not show a strong linear relationship between any of the attributes and the class variable that indicates the person is diabetes or not. A reason of it is because class variable is not continuous but nominal variable thus it would not be possible to show the "good" linear relationship. However, bmi together with glucose concentration attributes does show that they may have a positive linear relationship with class variable. On the other hand, the rest of variables do not show any significant linear relationship with class variable.

***QUESTION 5:***

*Results of overall purity:*

The purity of kmeans with 5 clusters: 0.6861979

Purity of each cluster for kmeans with k = 5 :

The purity of cluster 1 : 0.8928571

The purity of cluster 2 : 0.6428571

The purity of cluster 3 : 0.5675676

The purity of cluster 4 : 0.6127168

The purity of cluster 5 : 0.5048077

Majority class is class 1 with 280 elements.

*Centers:*

Cluster centers is tuple

(times\_pregnant ,glucose\_concentration ,blood\_pressure ,skin\_thickness ,serum\_insulin ,bmi ,diabetes\_function ,age )

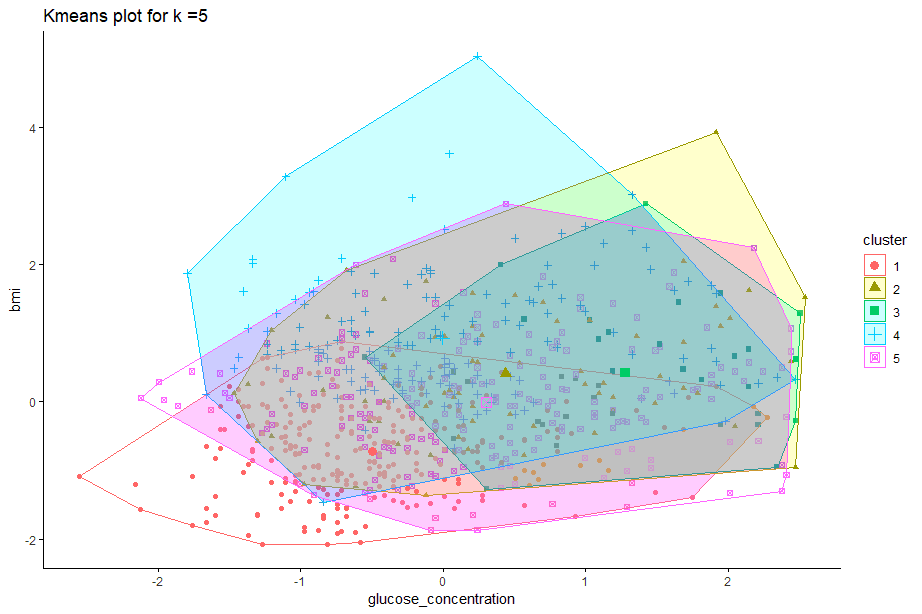
Cluster1 : (-0.4447511 ,-0.4981026 ,-0.5164601 ,-0.6712299 ,-0.375246 ,-0.7188994 ,-0.2495189 ,-0.6154057 )

Cluster2 : (-0.00913063 ,0.437351 ,0.1451757 ,0.3224563 ,0.09799615 ,0.4023395 ,2.1449 ,0.1447226 )

Cluster3 : (-0.05828725 ,1.279172 ,-0.2017728 ,0.3719967 ,3.308412 ,0.4171425 ,0.0626941 ,-0.04576294 )

Cluster4 : (-0.6127489 ,-0.007437726 ,0.2762424 ,0.7438473 ,-0.132568 ,0.926229 ,-0.2086642 ,-0.4171368 )

Cluster5 : (1.121787 ,0.3019785 ,0.4525106 ,0.1102066 ,-0.006095279 ,-0.01222983 ,-0.2235503 ,1.134812 )



*For kmeans with 8 clusters*

The purity of kmeans with 8 clusters: 0.6875

Purity of each cluster for kmeans with k = 8 :

The purity of cluster 1 : 0.6

The purity of cluster 2 : 0.6491228

The purity of cluster 3 : 0.6071429

The purity of cluster 4 : 0.5

The purity of cluster 5 : 0.5092593

The purity of cluster 6 : 0.9647059

The purity of cluster 7 : 0.5189873

The purity of cluster 8 : 0.7686567

Majority class is class 6 with 170 elements.

*Centers:*

Cluster centers is tuple

(times\_pregnant ,glucose\_concentration ,blood\_pressure ,skin\_thickness ,serum\_insulin ,bmi ,diabetes\_function ,age )

Cluster1 : (-0.07272486 ,1.351839 ,-0.1311702 ,0.4806227 ,3.738648 ,0.4134211 ,0.3570183 ,0.01919871 )

Cluster2 : (-0.5735945 ,0.8713944 ,0.8962282 ,1.306806 ,0.3599116 ,1.624151 ,0.5357511 ,-0.2293344 )

Cluster3 : (-0.1299597 ,0.4324226 ,0.32173 ,-0.2666054 ,-0.001810643 ,-0.3237517 ,-0.387028 ,-0.1632159 )

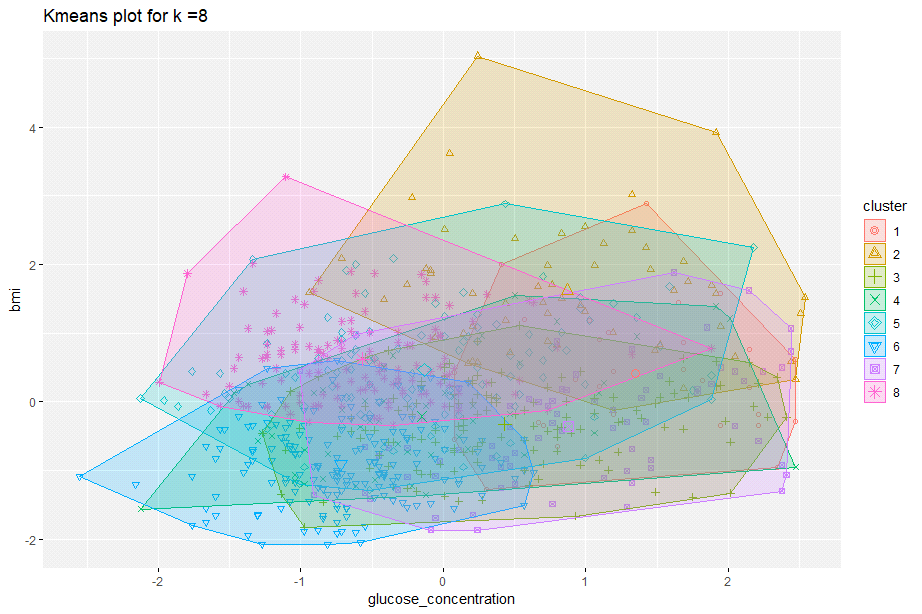
Cluster4 : (0.1409517 ,-0.1514262 ,-0.1047166 ,-0.2579929 ,-0.1111276 ,-0.2165574 ,2.16964 ,0.234614 )

Cluster5 : (1.587556 ,-0.1330634 ,0.326257 ,0.4542917 ,-0.09027887 ,0.4669132 ,-0.1954432 ,0.7243373 )

Cluster6 : (-0.5388331 ,-0.7189824 ,-0.7920245 ,-0.8908435 ,-0.4550825 ,-0.908405 ,-0.2354863 ,-0.7372557 )

Cluster7 : (0.654557 ,0.8769543 ,0.6869248 ,-0.1404418 ,0.07180766 ,-0.3581342 ,-0.0445551 ,1.927726 )

Cluster8 : (-0.6383655 ,-0.5662242 ,-0.3120518 ,0.5581537 ,-0.338979 ,0.6229007 ,-0.2304928 ,-0.6087292 )



**For PAM with 5 clusters**

Purity of PAM with k = 5 with weight (1,1,1,1,1,1,1,1): 0.7317708

Purity of each cluster for pam with k = 5 :

The purity of cluster 1 : 0.6814815

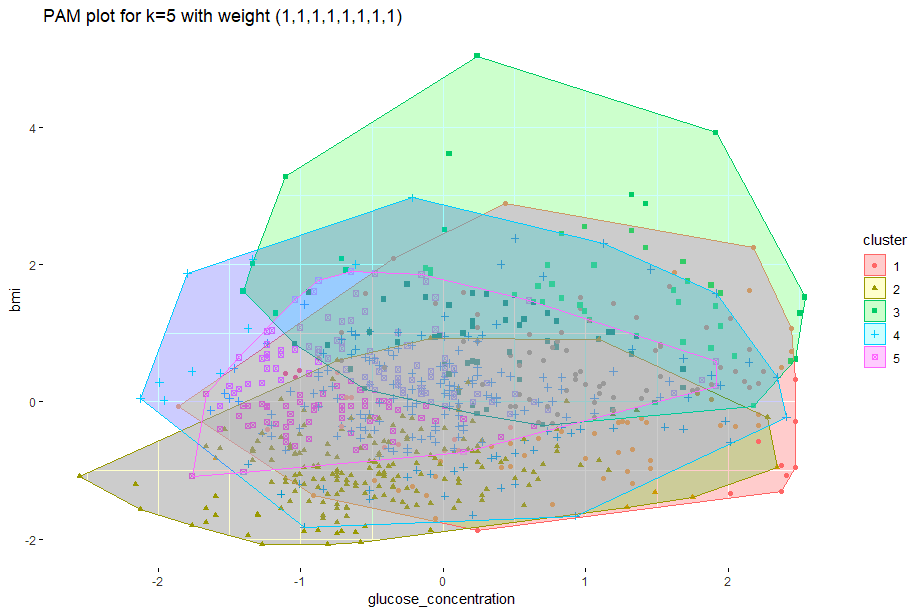
The purity of cluster 2 : 0.9234694

The purity of cluster 3 : 0.5684211

The purity of cluster 4 : 0.6030928

The purity of cluster 5 : 0.7972973

Majority class is class 2 with 196 elements.  
Centroids(indexes of centroids): 755 80 471 117 123



**For PAM with 5 clusters**

Purity of PAM with k = 5 with weight (0.2,1,0,0,0,1,0.2,0.2): 0.7630208

Purity of each cluster for pam with k = 5 :

The purity of cluster 1 : 0.741573

The purity of cluster 2 : 0.8971963

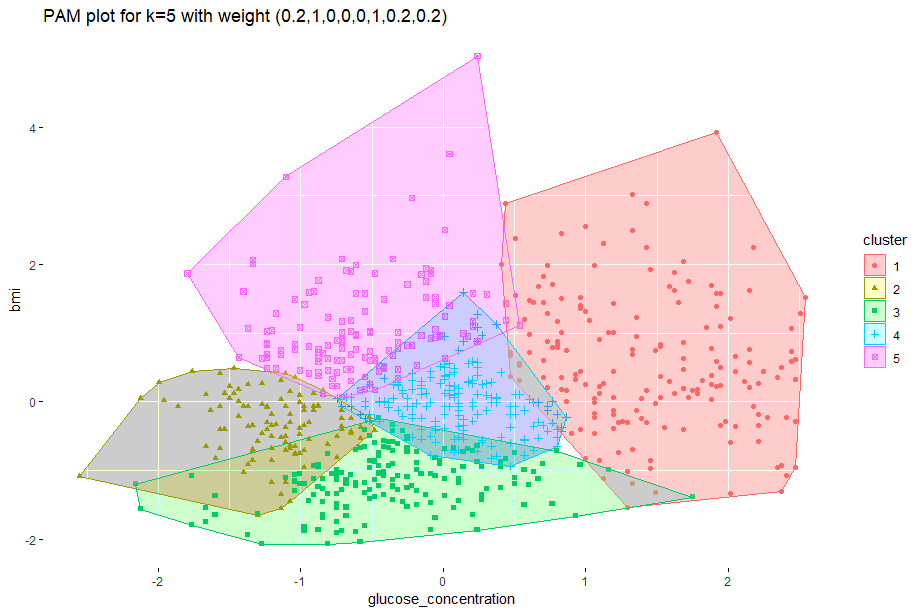
The purity of cluster 3 : 0.9314286

The purity of cluster 4 : 0.5517241

The purity of cluster 5 : 0.738806

Majority class is class 1 with 178 elements.

Centroids(indexes of centroids): 481 458 80 92 88



**For PAM with 5 clusters**

Purity of PAM with k = 5 with weight (0,1,0,1,0,1,0,0): 0.7473958

Purity of each cluster for pam with k = 5 :

The purity of cluster 1 : 0.7681159

The purity of cluster 2 : 0.8167539

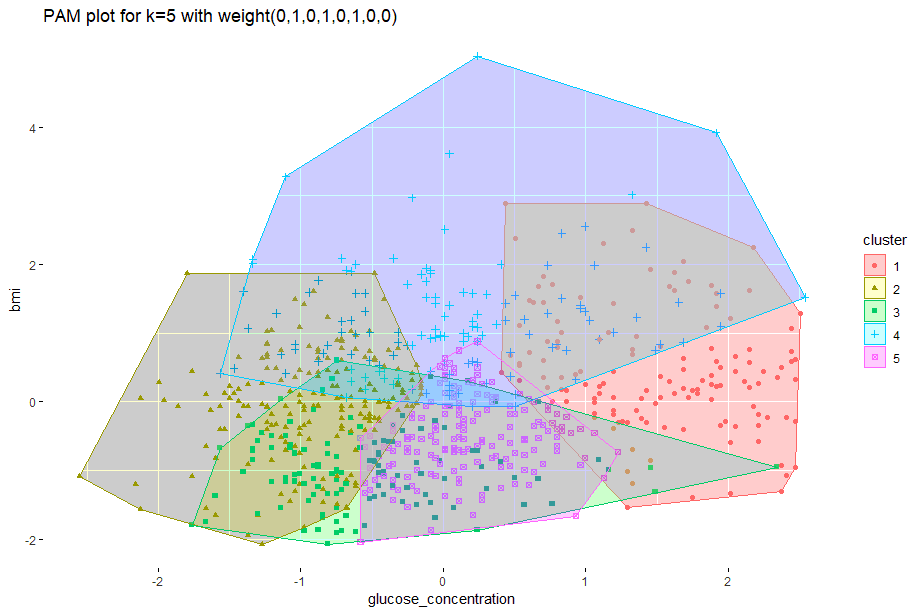
The purity of cluster 3 : 0.9307692

The purity of cluster 4 : 0.5808824

The purity of cluster 5 : 0.6473988

Majority class is class 2 with 191 elements.

Centroids(indexes of centroids): 703 401 489 503 106



***Interpretation of results:***

The overall purity of PAMs (purity of around 73%, 76%, 74%) are better than Kmean(purity of around 68%), and PAM with weight (0.2,1,0,0,0,1,0.2,0.2) achieved the best result, this also showed in the plot of it. The results from PAMs are better are probably because we have the weighted distance function. By having weighted distance function, the more important attributes are implied. And, because the results were better, we now know that glucose concentration and bmi play an important role in determining if a person would have diabetes. Furthermore, the result obtained from also weighing attributes: times pregnant, diabetes function and age with less importance are better than the result getting from only equally weighing glucose concentration, skin thickness, bmi and ignoring all other attributes. This is to hint that other attributes would have an impact as well while predicting the person would be likely to have diabetes. Nonetheless, there seems to be a difficulty in diagnosing diabetes since we do not have a clear idea of which attributes would contribute in the diagnosis.

***QUESTION 6:***

To answer this problem, instead of manually trying all possible parameters which is impossible, after trying a few parameters for min points and epsilon, I wrote some lines of code to get optimal result in the range that I think would give the desired result with has less than 20% outliers and the number of clusters is between 3 and 13. The code takes a little bit long because I set small increment between each epsilon. The smaller the epsilon, the more possible results we could achieve, and the better result might be revealed. However, since it would take too long for a very small increment of epsilon, I decided to set it as 0.001, and also I think that it would not make too much of a difference of smaller increment of epsilon.

The result I got is:

DBSCAN clustering for 768 objects.

Parameters: eps = 1.447, minPts = 2

The clustering contains 12 cluster(s) and 149 noise points.

0 1 2 3 4 5 6 7 8 9 10 11 12

149 591 4 2 2 2 3 3 2 2 3 3 2

Purity for DBSCAN result = (purity, outlier) = 0.7156704 0.1940104, the purity is around 71.567% and 19.4% outliers.

I tried to see how pure each cluster is and I got this:

Purity of each cluster for dbscan\_fast with k = 12 :

The purity of cluster 1 : 0.7072758

The purity of cluster 2 : 0.75

The purity of cluster 3 : 1

The purity of cluster 4 : 1

The purity of cluster 5 : 1

The purity of cluster 6 : 1

The purity of cluster 7 : 0.6666667

The purity of cluster 8 : 1

The purity of cluster 9 : 0.5

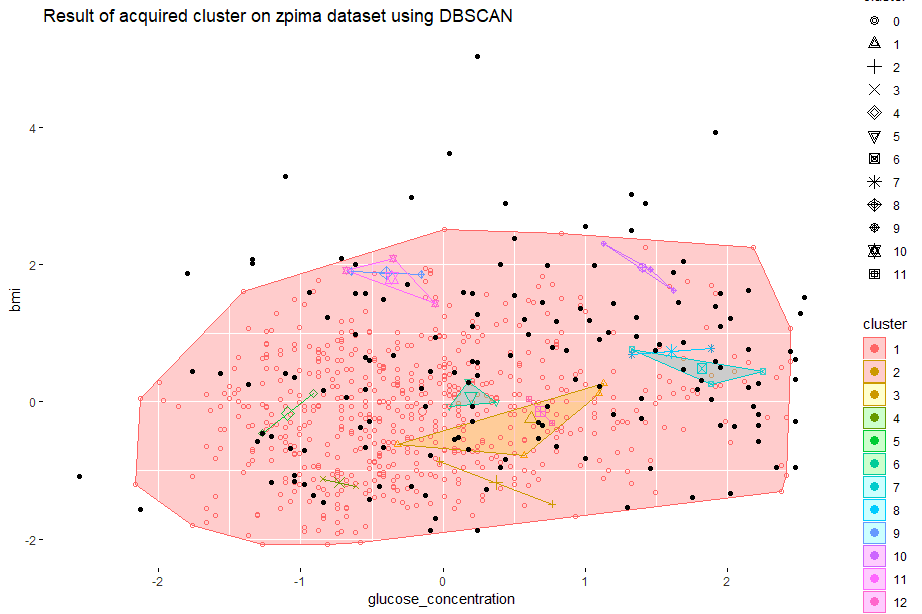
The purity of cluster 10 : 1

The purity of cluster 11 : 1

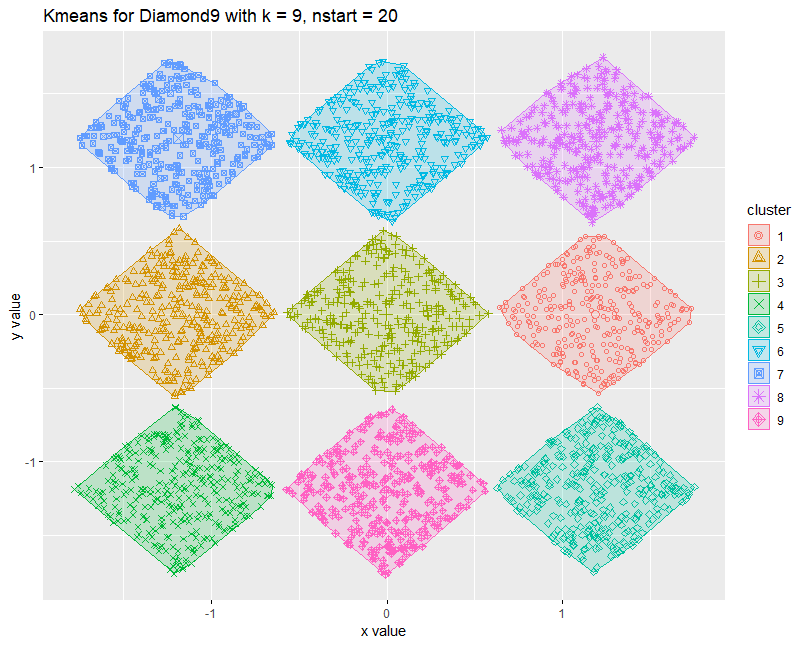
The purity of cluster 12 : 1

Majority class is class 2 with 591 elements. (Supposed to print class 1 but printed class 2 instead. This is because DBSCAN has class 0 also, and count from 0).

As we can see, the biggest cluster (cluster 1) has almost the same purity as the overall cluster, other clusters have "better" purity, some even are "perfectly pure". This is just because there are two few elements in those clusters. If I would have set the bigger min points, we would not have had less than 20% outliers. The overall purity using DBSCAN is also like the purity we achieved above using PAM.



***QUESTION 7:***



A screenshot of a cell phone

Description generated with high confidence

Purity of Kmeans for Diamond9 (k = 9, nstart = 20): 1

Purity of Kmeans for Diamond9 (k = 12, nstart = 20): 0.9943333

As we can see from the results, kmeans with k = 9 gave the perfect result with overall purity = 1. This is predictable since the data points are nicely spaced and we know the exact number of k, plus, k-means algorithm does not produce any outliers. Kmeans with k = 12 does give the almost perfect purity of 99.4% yet on the plot it shows that it should not be 12 clusters.

***QUESTION 8:***

To answer question 8, instead of manually trying all the possible parameters for min points and epsilon (which is impossible), I've tried to come up with some lines of code that would iterate through a range of values for eps and min points. Even though this might not be the "best" result, it does give the optimal result in the range that I've come up with (2<= minpoints <=5 and epsilon is in-between 1/6 of the range of the x or y values, whichever is smaller). The result may get better if we set even smaller increment for epsilon, but since it would take too long, I set it to 0.005.

The result I'd gotten:

DBSCAN clustering for 3000 objects.

Parameters: eps = 0.115, minPts = 3

The clustering contains 16 cluster(s) and 14 noise points.

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

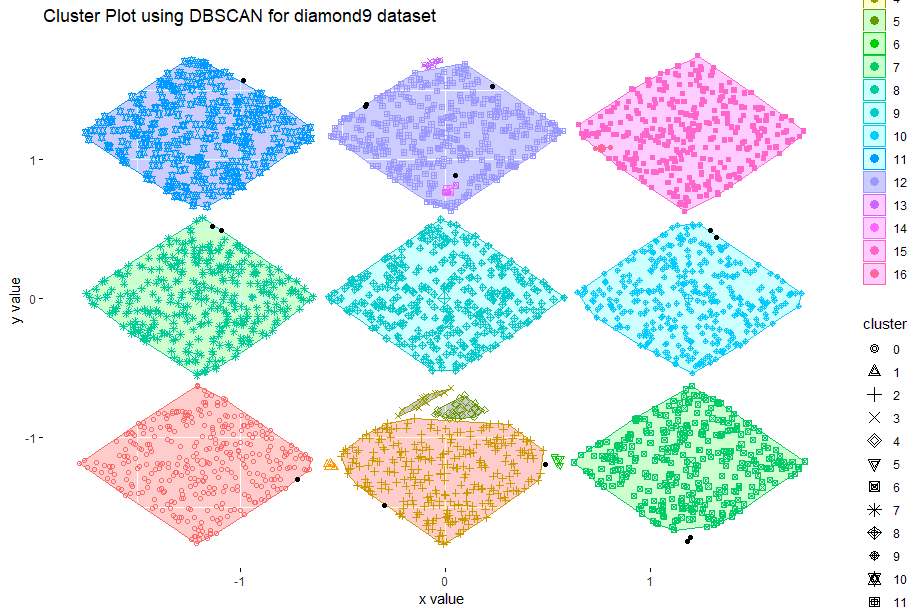
14 332 3 290 12 24 3 331 331 334 331 332 322 4 4 330 3

Purity for DBSCAN result = (purity, outlier) = 1 0.004666667

The "optimal result" in the range I set is the one that has eps = 0.115, minPts = 3

The purity is 1 and the number of outliers is very small, only 14 out of 3000, which is 0.004666667.

This result is not as good as the one obtaining from kmeans method even though the purity is still absolute. Instead of 9 desired 9 clusters, we got 16 clusters. This is not something unreasonable since kmeans does not allow outliers while DBSCAN does. However, the number of outliers is very minimal so DBSCAN in this case does, in fact, give the very good result.



***QUESTION 9:***

Kmeans is a well-known method for clustering tasks. It is intuitively easy to understand and has a fast running time. Furthermore, it does not allow outliers, therefore if the dataset is well separated and the k is known beforehand, the result acquired would be indeed good. To know the “best” k, usually an ability to visualize dataset would help. However, this will not be the case where our dataset’s dimensionality is high (>= 4) that makes it almost impossible to visualize on a 2D screen. Also, Kmeans will not work well if the data is not distributed in convex shape. Kmeans is also very sensitive to outliers and it is hard to tweak the distance and give it different weights between attributes using default Kmeans function in R.

DBSCAN on the other hand, works very well with convex as well as non-convex dataset. DBSCAN also help us to identify outliers which Kmeans would not be able to give us. For the cases of using Kmeans and DBSCAN for zpima dataset, DBSCAN got a better purity than kmeans, however, for diamond9 dataset, DBSCAN gave more clusters than we hope it would, it also gave the cluster of outliers which we would expect it not to.

After the experimental result, I’ve found it hard to tune the parameters to achieve better and better result since we do not know for sure which one of the attributes are main factors contributing in diagnosing diabetes and which ones we can safely ignore. For the given parameters from assignment, I’ve learnt that diabetes has a pretty strong correlation with BMI and glucose concentration, it also has correlation with other factors, such as age, diabetes functions, etc. Therefore, by weighing the distance matrix of attributes, we were able to discover better purity for PAM task. The “best” result was the one that had purity of around 76% with weight vector (0.2,1,0,0,0,1,0.2,0.2).