Experiment 10:

Implement K-Medoids clustering using R.

Solution:

Partition Around Medoids (PAM)

PAM stands for “Partition Around Medoids.” PAM converts each step of PAM from a deterministic computational to a statistical estimation problem and reduces the complexity of a sample size n to O(n log n). Medoids are data points chosen as cluster centers. K-Means clustering aims at minimizing the intra-cluster distance (often referred to as the total squared error). In contrast, K-Medoids minimizes dissimilarities between points in a cluster and points considered as centers of that cluster.

Algorithm

The fundamental concept of PAM includes:

1. Find a set of k Medoids (k refers to the number of clusters, and M is a collection of medoids) from the data points of size n (n being the number of records).
2. Using any distance metric (say d(.), could be euclidean, manhattan, etc.), try and locate Medoids that minimize the overall distance of data points to their closest Medoid.
3. Finally, swap Medoid and non-Medoid pairs that reduce the loss function L among all possible k(n-k) pairs. The loss function is defined as:

$$L\left(M\right) = \sum_{i=1}^{n} \min_{m \varepsilon M} d\left(m, x_i\right)$$

Update centroids: In the case of K-Means, we were computing the mean of all points present in the cluster. But for the PAM algorithm, the updation of the centroid is different. If there are m-point in a cluster, swap the previous centroid with all other (m-1) points and finalize the point as a new centroid with a minimum loss. Minimum loss is computed by the above cost function

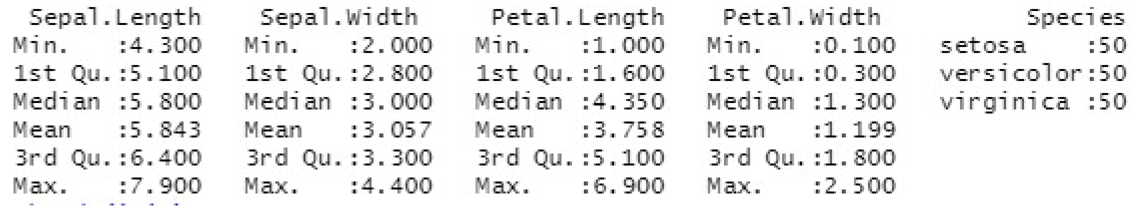
Algorithm implementation

1. Install the relevant packages and call their libraries

```
> library("ggplot2")
> library("cluster")
```

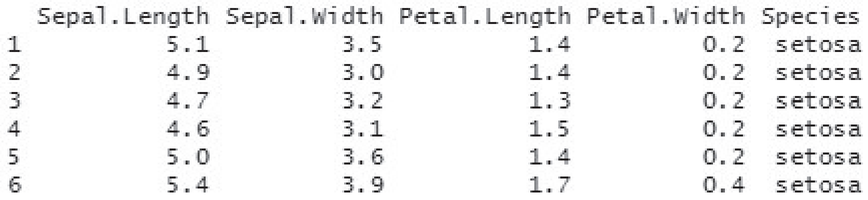
2. Loading and analyzing the dataset

```
> summary ("iris")
```



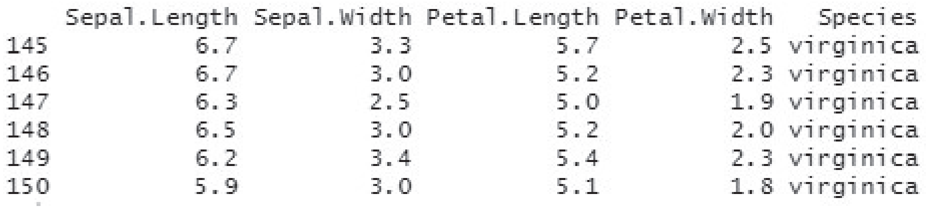
```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

> head ("iris")



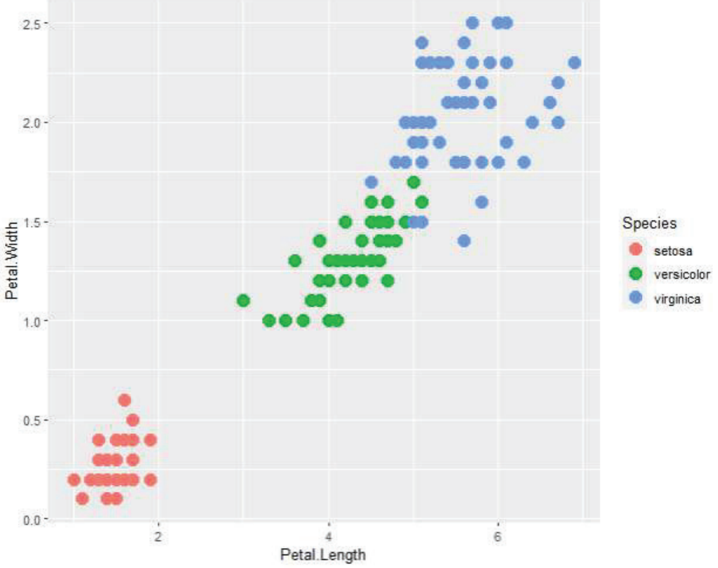
```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

> tail ("iris")



```
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
145          6.7         3.3          5.7         2.5 virginica
146          6.7         3.0          5.2         2.3 virginica
147          6.3         2.5          5.0         1.9 virginica
148          6.5         3.0          5.2         2.0 virginica
149          6.2         3.4          5.4         2.3 virginica
150          5.9         3.0          5.1         1.8 virginica
```

> ggplot(iris)+aes(Petal.Length,Petal.Width)+geom_point(aes(col=Species),size=4)



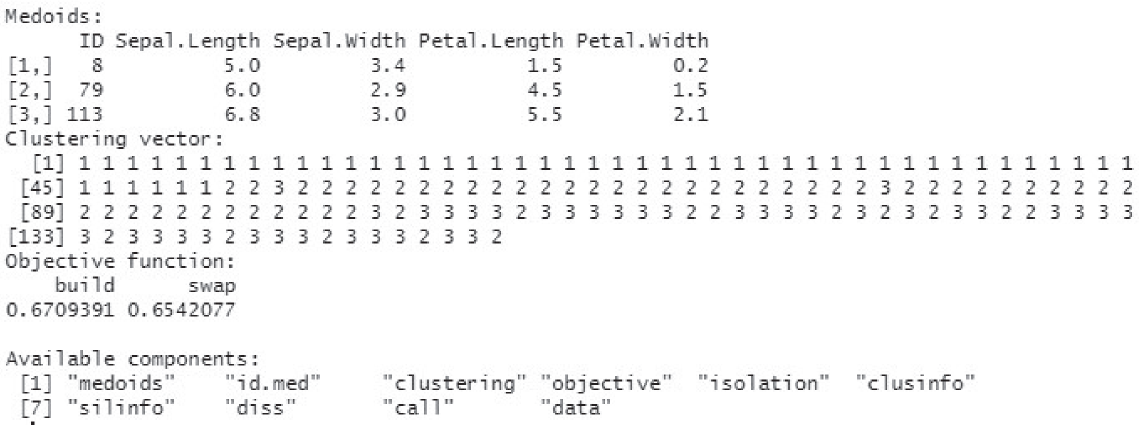
3. Eliminating the target variable

> data <- select (iris, c(1:4))

4. Apply k-medoids algorithm using PAM function

> kmediod <- pam(data, k=3, metric="euclidean")

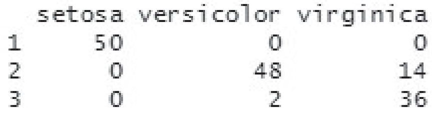
> kmediod



```
Medoids:
      ID Sepal.Length Sepal.Width Petal.Length Petal.Width
[1,]   8          5.0         3.4          1.5         0.2
[2,]  79          6.0         2.9          4.5         1.5
[3,] 113          6.8         3.0          5.5         2.1
Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [45] 1 1 1 1 1 1 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2
 [89] 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 3 2 3 3 3 3 3 3 2 2 3 3 3 3 2 3 2 3 2 3 3 2 2 3 3 3 3
[133] 3 2 3 3 3 3 2 3 3 3 2 3 3 3 2 3 3 2
Objective function:
    build      swap
0.6709391 0.6542077

Available components:
 [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"  "clusinfo"
 [7] "silinfo"    "diss"       "call"       "data"
```

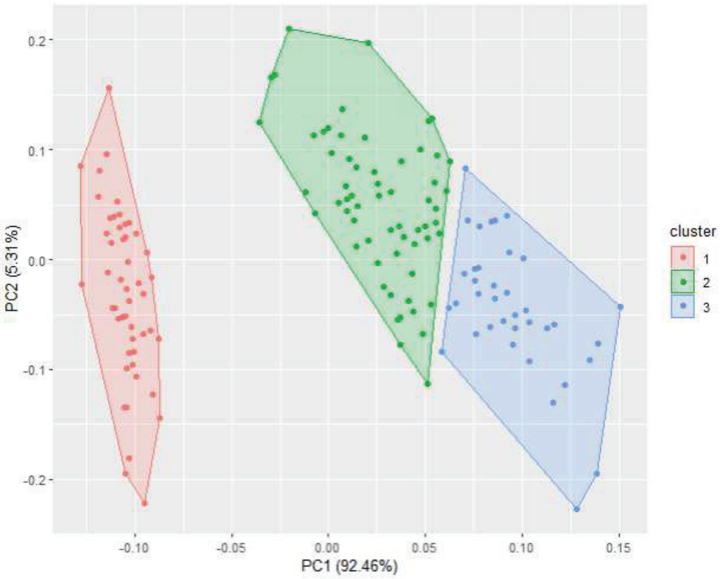
> table (kmediod$clustering, iris$Species)



```
   setosa versicolor virginica
 1     50          0         0
 2      0         48        14
 3      0          2        36
```

5. Plotting our data-points in clusters

> autoplot (kmediod, data, frame=TRUE)


PC2 (5.31%)
PC1 (92.46%)
cluster