# Data Mining Lab, Exercise 3

Team# 1: Poliakov Valerii, Holovnia Dmytro, Selvaraj Sinju

Dataset: f1.csv

## Task# 1

*(3pt) Perform the calculations using the k-means algorithm for data from file “f#.csv”.*

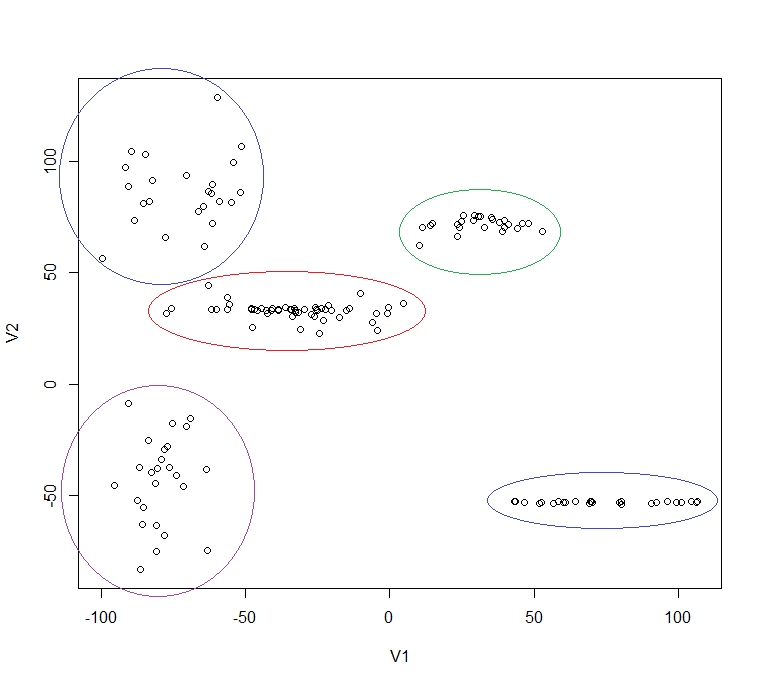
*a) Decide on the most likely number of clusters on the basis of the scatter plot of the data. For the chosen number of clusters, perform clustering using the k-means algorithm. Repeat the calculations 5 times. Write the values of SSB/TSS for each of the algorithm runs. Indicate the minimal and maximal value of SSB/TSS as well as its average value. Show the best solution (clusters) in the scatter plot. Indicate the centroids.*

Step 1. Read the data. Draw plot and guess number of clusters.

d<-read.csv(file="f1.csv", header = FALSE, sep=' ')

d

with(d, plot(V1, V2))



Dataset in the f1.csv file has no headers and whitespace as a separator. Plot shows five distinct groups. Most likely there are 5 clusters.

Step 2. Repeat 5 times k-means clustering for five clusters. Note the SBB/TSS values for each iteration. SBB (Sum of squares between clusters) and TSS (Total Sum of Squares) are used as measures to evaluate the performance of the algorithm.

Script command:

km<-kmeans(d, centers=5)

Iteration 1:

Within cluster sum of squares by cluster:

[1] 10670.589 18327.261 10260.915 3436.462 10864.708

(between\_SS / total\_SS = 94.1 %)

Iteration 2:

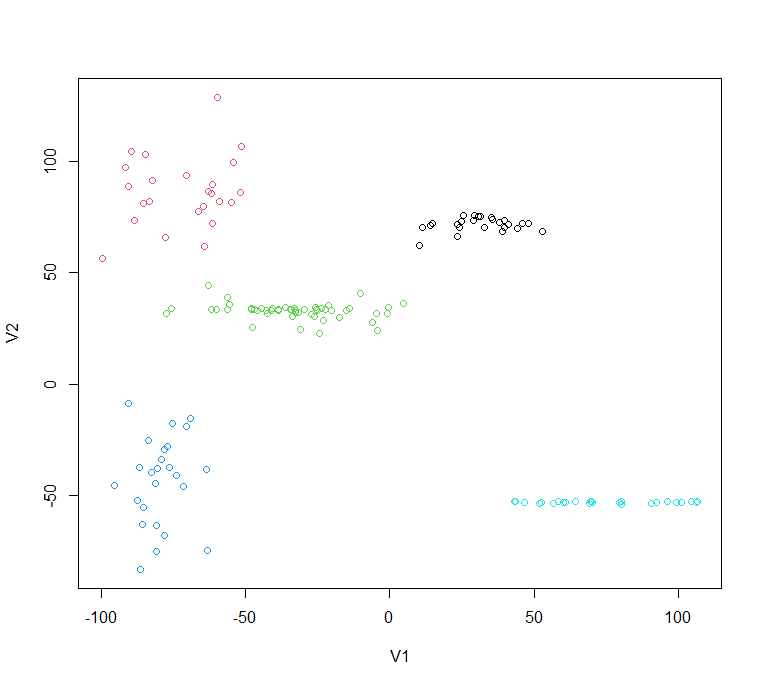
Within cluster sum of squares by cluster:

[1] 3436.462 10670.589 18327.261 10864.708 10260.915

(between\_SS / total\_SS = 94.1 %)

with(d, plot(V1, V2, col=km$cluster))

With the quality measure 94.1% diagram shows expected clusters



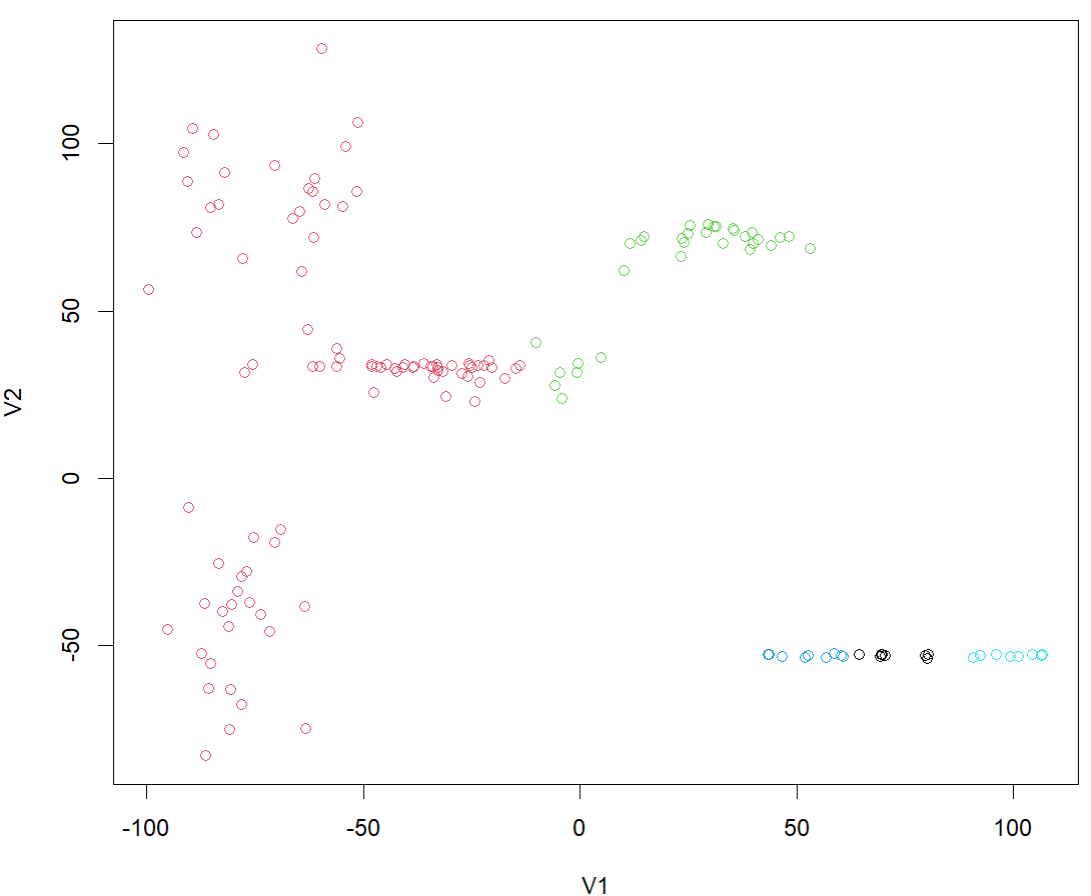
Iteration 3:

Within cluster sum of squares by cluster:

[1] 266.0465 274338.3389 18684.9885 385.3310 267.4781

(between\_SS / total\_SS = 67.9 %)

Quality measure is 67.9 which is worse than iteration 1 and 2. The diagram shows that algorithm identified 3 clusters in the right-bottom corner (where supposed to be only one) and remaining data split between remaining two clusters. For the greater number of clusters this might make sense, but we have only 5 so result is not very good.



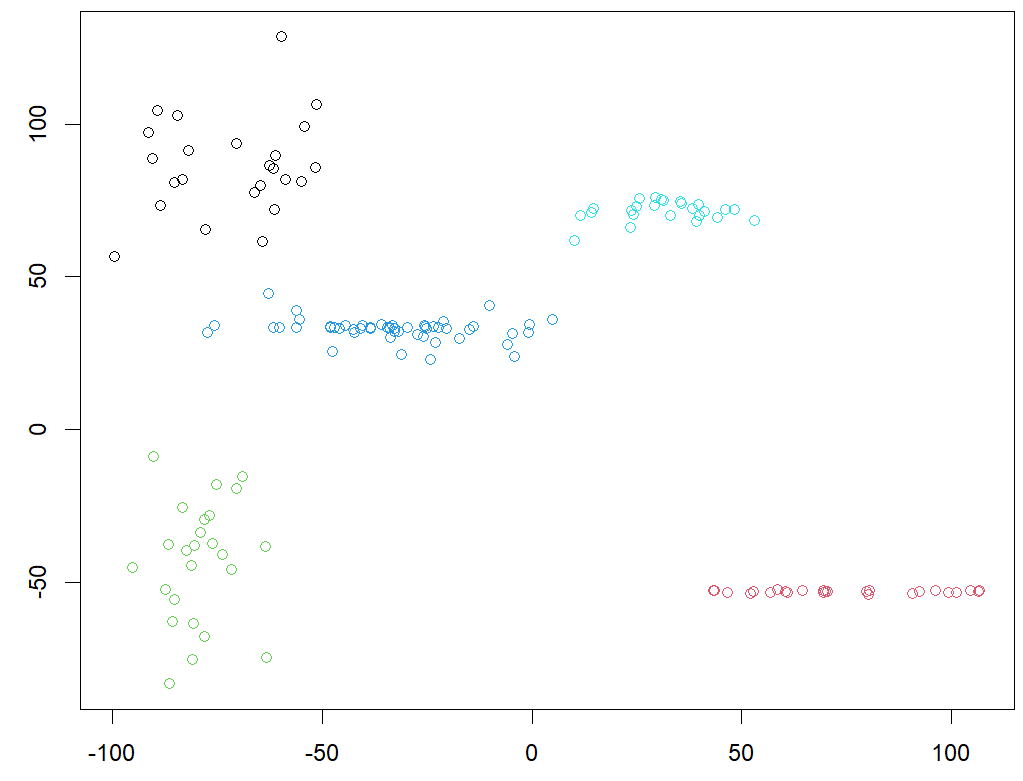
Iteration 4:

Within cluster sum of squares by cluster:

[1] 10670.589 10260.915 10864.708 18327.261 3436.462

(between\_SS / total\_SS = 94.1 %)

Again, very similar to Iteration 1 and 2



Iteration 5:

Within cluster sum of squares by cluster:

[1] 10670.589 10260.915 3436.462 10864.708 18327.261

(between\_SS / total\_SS = 94.1 %)

And last iteration also shows similar result.

Summary table for all 5 iterations

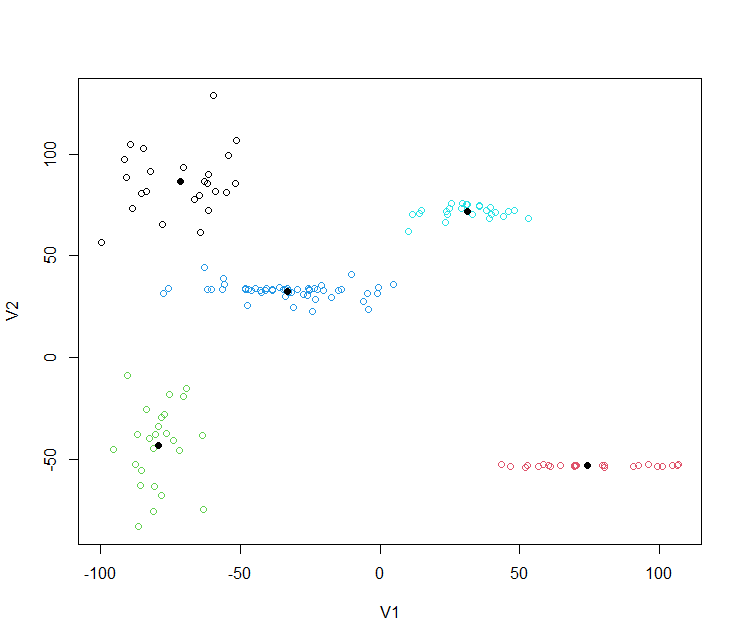
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Iteration 1 | Iteration 2 | Iteration 3 | Iteration 4 | Iteration 5 |
| SSB/TSS | 94.1% | 94.1% | 67.9% | 94.1% | 94.1% |

Minimal value: 67.9%

Maximal value: 94.1%

Average value: 88.86%

The solution with centroids shown on the diagram below.



*b) Identify the number of clusters using the Elbow Method. Draw the chart that shows how WSS depends on k. Compare the results with the results of Task 1a.*

Step 3. Use Elbow Method to identify number of clusters.

Script performs N iterations of k-means algorithm with number of clusters equal current iteration number from 1 to N. Algorithm runs K times in each iteration.

N <- 15 # Number of iterations

K <- 5 # Number of algorithm runs in each iteration

results <- list()

for (i in 1:N) {

km <- kmeans(d, centers = i, nstart = K)

results[[i]] <- km

}

wss <- sapply(results, function(x) x$tot.withinss)

wss

plot(wss, type = "b", xlab = "Iteration", ylab = "WSS")

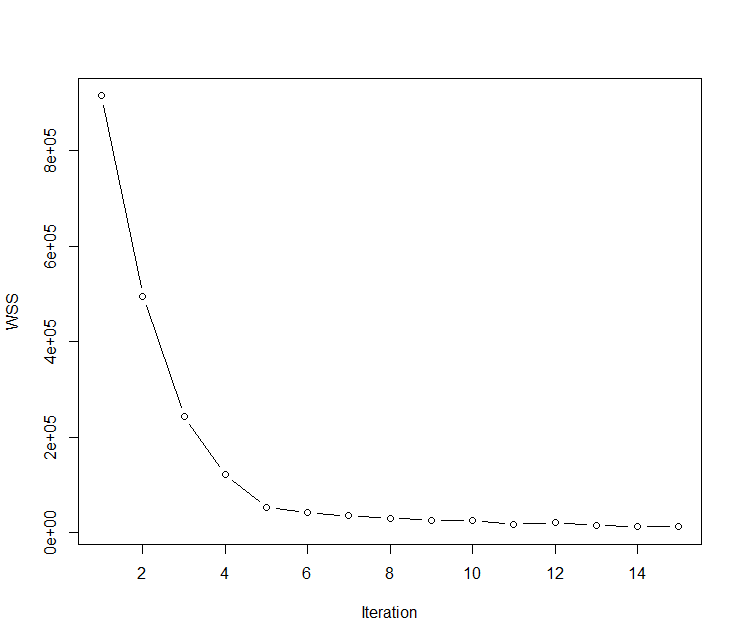
WSS in each iteration are:

[1] 914637.42 [2] 494055.40 [3] 244007.56 [4] 123037.13 [5] 53559.94

[6] 42632.39 [7] 35810.32 [8] 31048.06 [9] 26675.12 [10] 26372.47

[11] 18161.88 [12] 21246.19 [13] 15605.51 [14] 12704.59 [15] 13443.24

And the char is on the diagram below. On the diagram, we see that after 5 clusters, the WSS improves very slowly, so our initial assumption about 5 clusters seems correct.



Additionally, we can compare scatter plot for different number of clusters.

# plot 5th iteration result

with(d, plot(V1, V2, col=results[[5]]$cluster))

points(results[[5]]$centers, col=1,pch=16,cex=1)

|  |  |
| --- | --- |
| 5 clusters | 6 clusters |
| 7 clusters | 14 clusters |

On the scatter plot representation, we see that 7 clusters might make sense but not very different from 5 clusters. At the same time, 14 cluster definitely looks very redundant.