VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

ABHINAV VIJAYAKUMAR

19BCE1311

CSE3506 – ESSENTIALS OF DATA ANALYTICS
LAB-7

DR. LAKSHMI PATHI JAKKAMPUTI (L21 + L22)

Tasks for Week-7: Partitioning Based Clustering

Aim: To understand the following operations/functions on 'iris' data and perform similar operations on 'USArrests' dataset based on given instructions.

Algorithm:

1. Removing all the values from the global environment

2. Set the working directory to the dataset where we store by using setwd().

3. To see the dataset use view() function.

4. By using scale function. We scale the data and store it in another variable.

5. Using kmeans function we find the kmeans clustering with 2 center at first it can be of any centers and store the result in fit.

6. By using fit$cluster we can find the cluster values.

7. By using fit$size we can find the size of each cluster.

8. By using fit$withnss we can find with in cluster sum of squares for each cluster.

9. By using fit$tot.withnss we can find with in cluster sum of squares with respective to all clusters.

10. Create the no of iterations we need to find the perfect cluster and size of wcss and the nclust list.

11. To find the best no of center from 1 to 15 we create a for loop.

- **1.** find the kmeans cluster with each center value in for loop
- **2.** put to the total with in cluster sum of squares for each iteration in wcss
- **3.** put the size of cluster in nclust.

12. Plot the graph between the no of center and the wcss values for each center. the place where we find the bend that is our no of cluster should be taken.

13. In other way we can use factoextra library.

14. Using fviz_nbclust function we can find the graph.

15. Using fviz_cluster function we can find the clusters

16. Call cluster library

17. We use pam function to find the k medoid clusters and store the values in fitm.

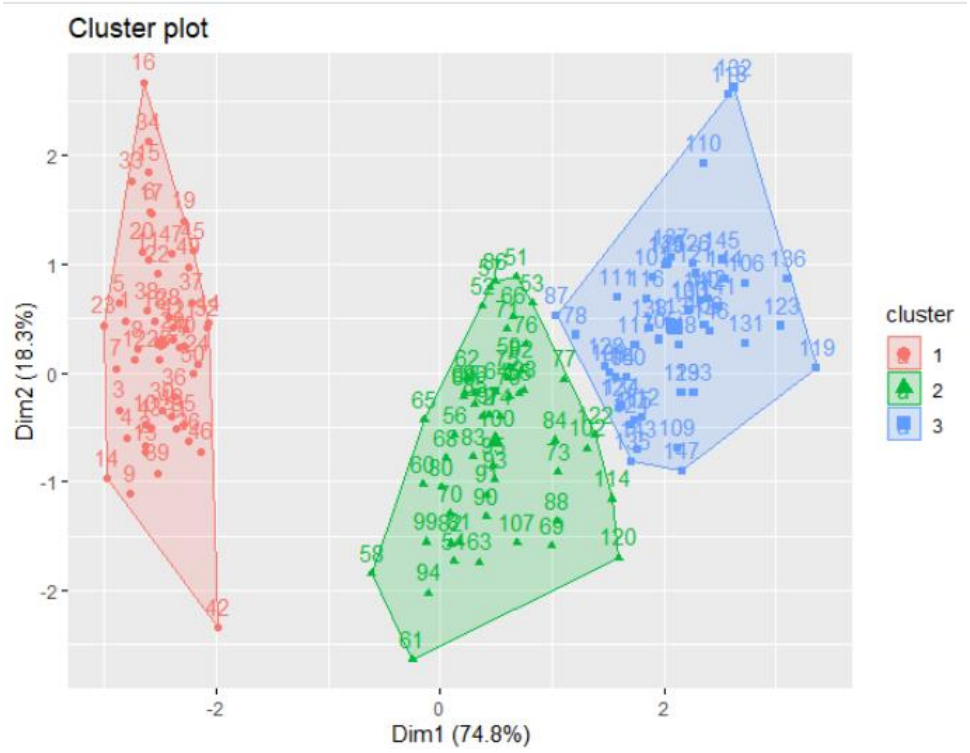
18. By using the fitm$medoid we can find no of medoid.

19. Using fviz_cluster function we can find the medoids.

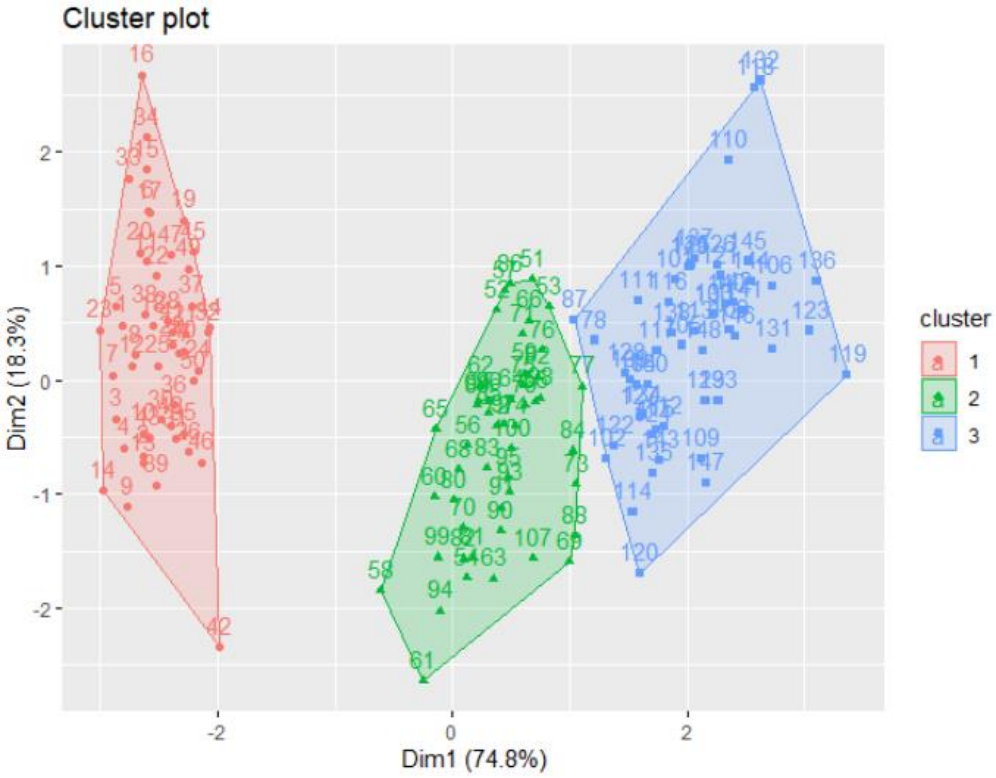
Result:

<u>For iris.csv</u>

K-means centers:

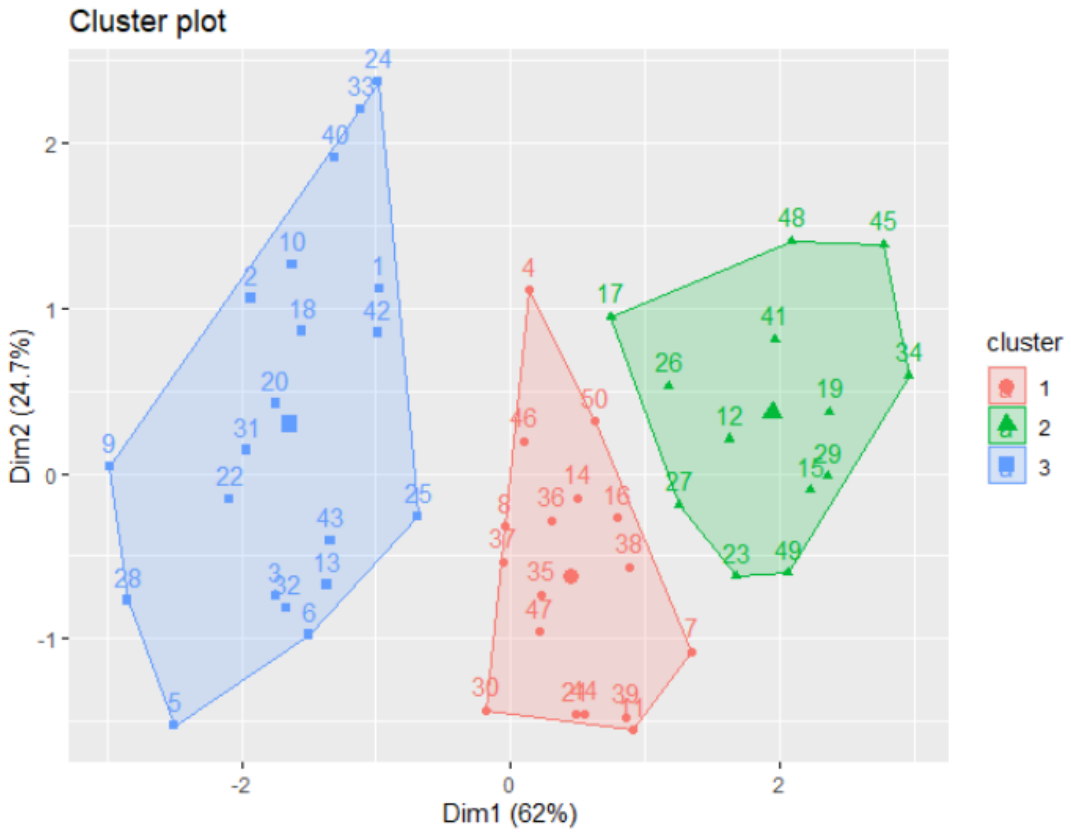


K-medoid centers:

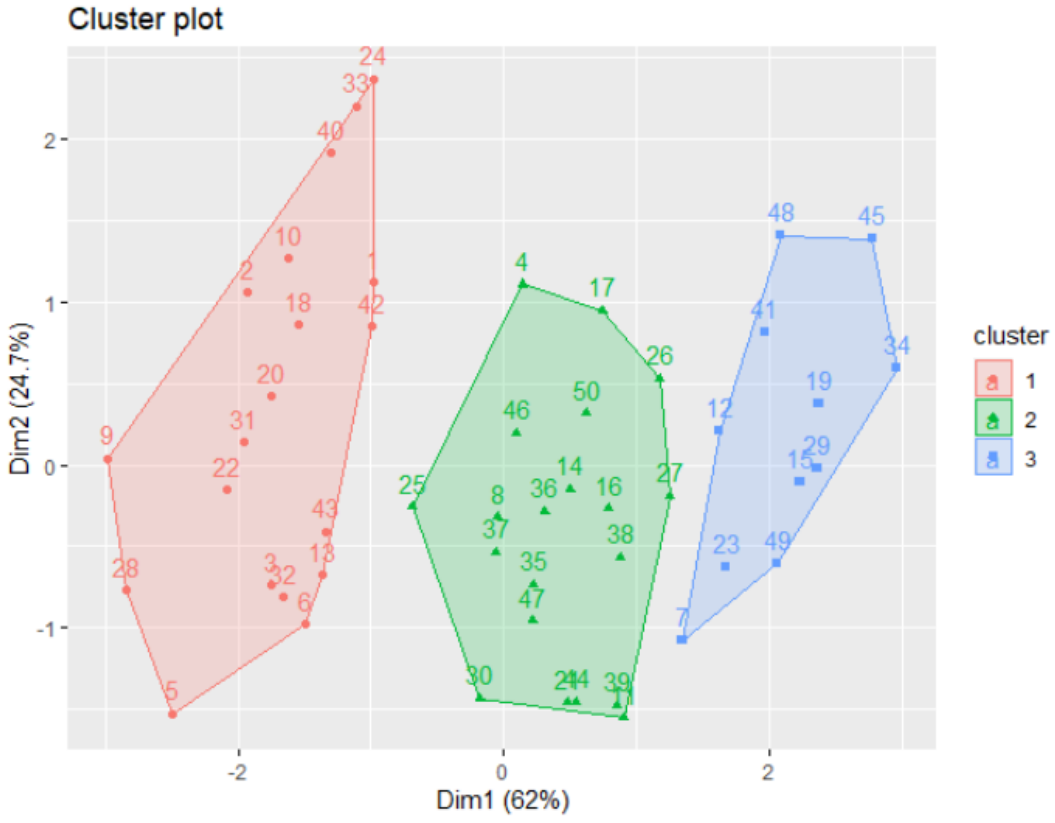


For USArrests.csv

K-means centers:



K-medoid centers:



Statistics:

For iris.csv

- **K-means centers**

X	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
-1.15087068	-1.01119138	0.85041372	-1.3006301	-1.2507035
0.07534946	0.03881135	-0.73324663	0.3059615	0.2137533
1.13936197	1.03196952	-0.07784286	1.0386287	1.0894947

- **K-medoid centers**

X	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
-1.07030973	-0.7769106	0.7861738	-1.3357516	-1.3110521
-0.08056095	0.3099591	-0.5903951	0.1370873	0.1320673
0.95522266	0.7930124	-0.1315388	0.9868021	0.7880307

For USArrests.csv

- **K-means centers**

Murder	Assault	UrbanPop	Rape
-0.4469795	-0.3465138	0.4788049	-0.2571398
-0.9615407	-1.1066010	-0.9301069	-0.9667633
1.0049340	1.0138274	0.1975853	0.8469650

- **K-medoid centers**

Murder	Assault	UrbanPop	Rape
0.8292944	1.3708088	0.3081225	1.1603196
-0.2727580	-0.2371077	0.1699510	-0.1315342
-1.2829727	-1.3770485	-0.5899924	-1.0603878

Program:

For iris.csv

```
rm(list=ls())
setwd("C:/Users/Abhinav Vijayakumar/Desktop/VIT Academics/Sem 6/EDA/LAB/LAB 7")
data1<-read.csv("iris.csv")
View(data1)
df<-scale(data1)
fit<-kmeans(df,centers=2)
fit$cluster
fit$size
fit$withinss
fit$tot.withinss
Kmax<-15
wcss<-rep(NA,Kmax)
nClust<- list()
for(i in 1:Kmax){
  fit<-kmeans(df,i)
  wcss[i]<-fit$tot.withinss
  nClust[[i]]<-fit$size
}
plot(1:Kmax,wcss,type="b",pch=19)
fit<-kmeans(df,centers=3)
fit$cluster
fit$size
fit$center
library(factoextra)
fviz_nbclust(df, kmeans, method = "wss")
fviz_cluster(fit, data1)
library(cluster)
fitm <- pam(df, 3, metric = "manhattan")
fitm
fitm$medoids
fviz_cluster(fitm, data1)
```

For USArrests.csv

```
rm(list=ls())
setwd("C:/Users/Abhinav Vijayakumar/Desktop/VIT Academics/Sem 6/EDA/LAB/LAB 7")
data2<-read.csv("USArrests.csv")
view(data2)
data2<-data2[,-1]
df1<-scale(data2)
fit1<-kmeans(df1,centers=2)
fit1$cluster
fit1$size
fit1$withinss
fit1$tot.withinss
Kmax1<-15
wcss1<-rep(NA,Kmax1)
nClust1<- list()
for(i in 1:Kmax1){
  fit1<-kmeans(df1,i)
  wcss1[i]<-fit1$tot.withinss
  nClust1[[i]]<-fit1$size
}
plot(1:Kmax1,wcss1,type="b",pch=19)
fit1<-kmeans(df1,centers=3)
fit1$cluster
fit1$size
fit1$center
library(factoextra)
fviz_nbclust(df1, kmeans, method = "wss")
fviz_cluster(fit1, data2)
library(cluster)
fitm1 <- pam(df1, 3, metric = "manhattan")
fitm1
fitm1$medoids
fviz_cluster(fitm1, data2)
```