DM ASSIGNMENT

AMMANAMANCHI SAI KARTHIK

B150310CS

10

K-means clustering with K=actual number of classes in your cleaned dataset

K Means Clustering is an unsupervised learning algorithm that tries to cluster data based on their similarity The algorithm randomly assigns each observation to a cluster, and finds the centroid of each cluster. Then, the algorithm iterates through two steps:

- Reassign data points to the cluster whose centroid is closest.
- Calculate new centroid of each cluster.

We are using the IRIS data set having 4 attributes and 1 target attribute.

Loading the data set

```
library(readxl)
mydata <- read_excel("C:/Users/karthik/Desktop/iris.xlsx")
```

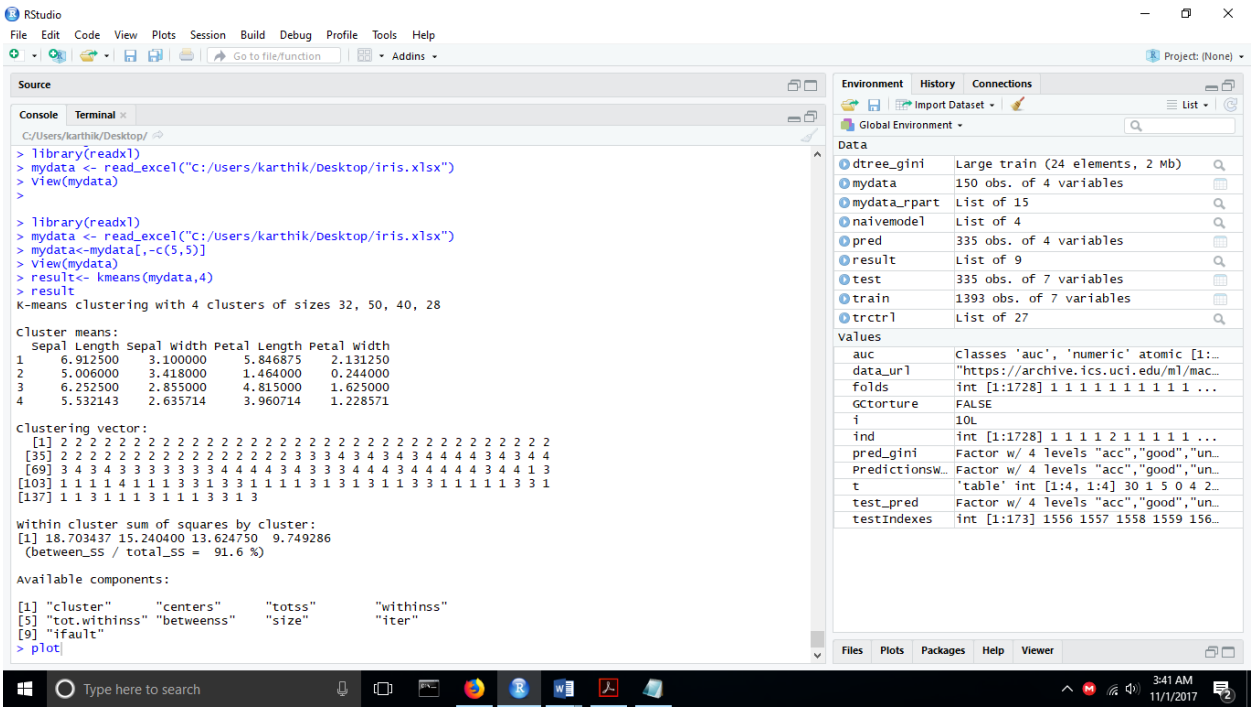
2.a With K=4

Removing the target attribute so that we can use the k means algorithm

```
mydata<-mydata[,-c(5,5)]
```

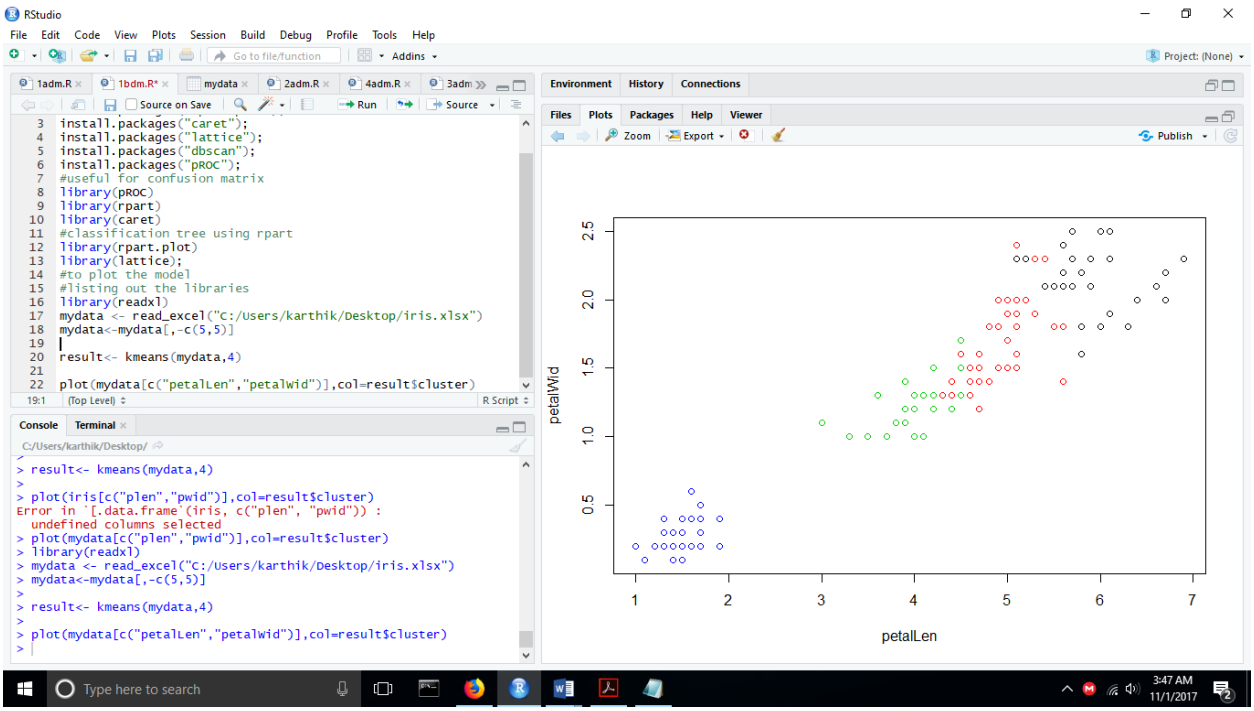
Using the K means algorithm

```
result<- kmeans(mydata,4)
```



Plotting the result

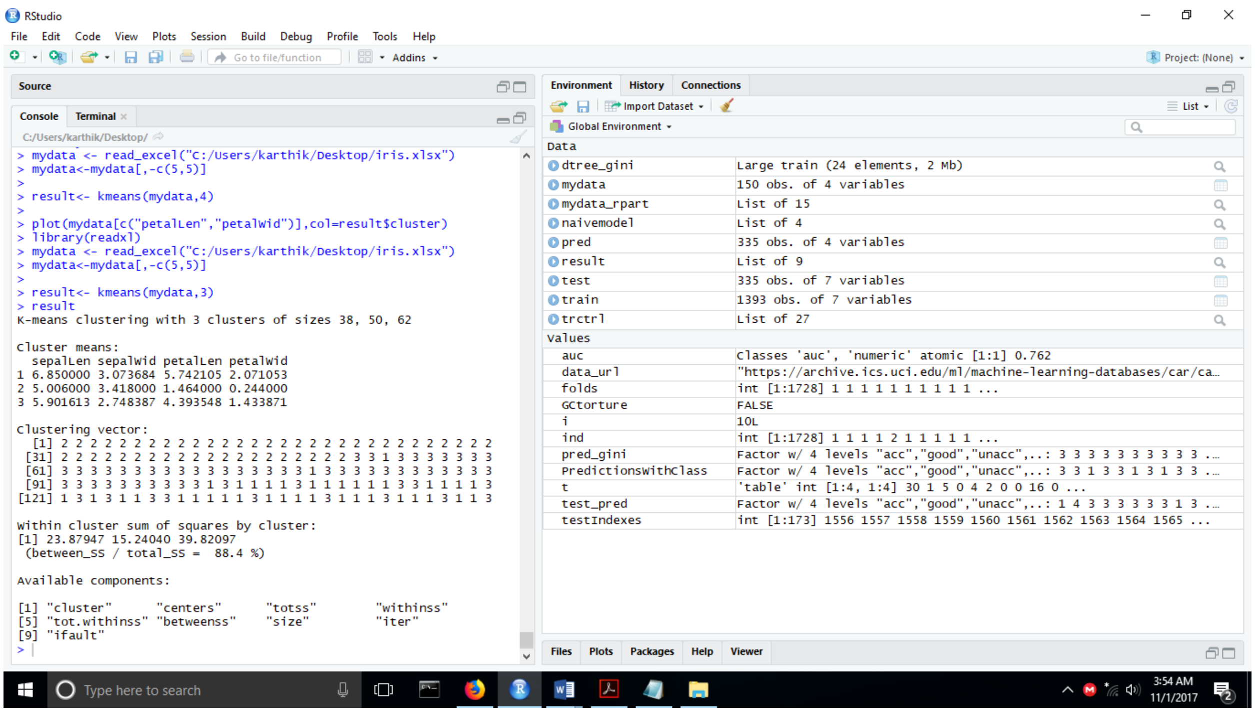
```
plot(mydata[c("petalLen","petalWid")],col=result$cluster)
```



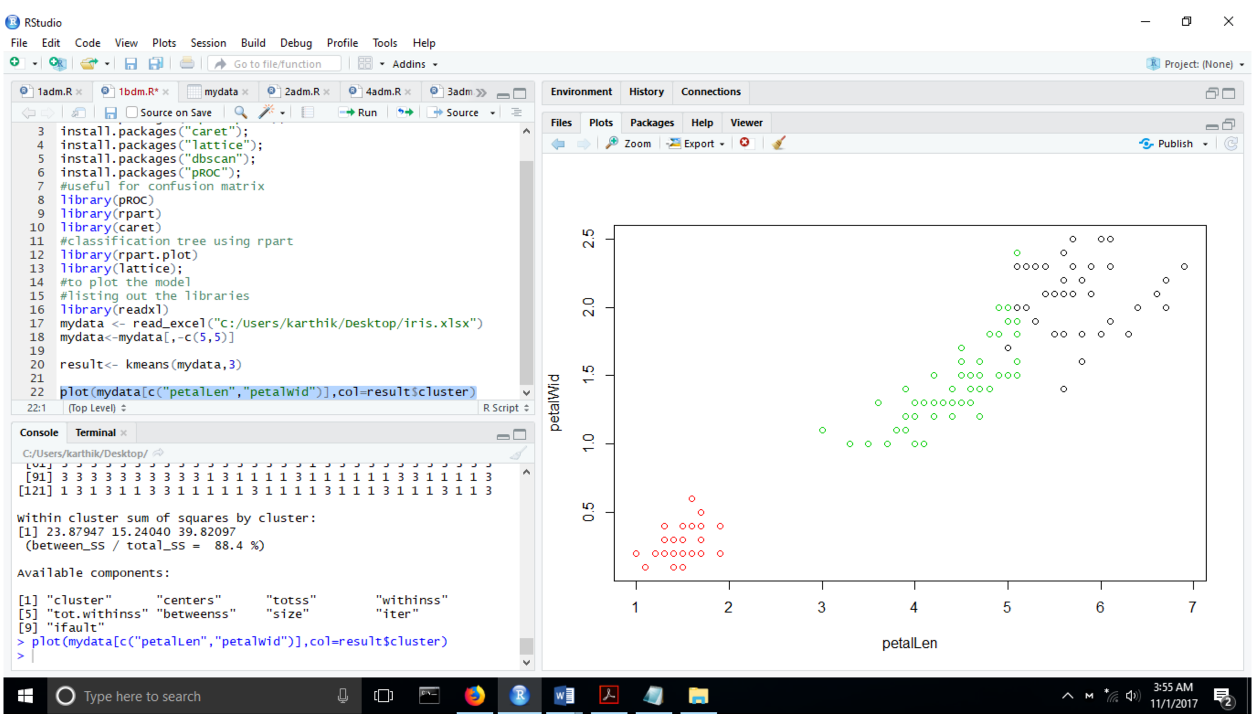
BSS/TSS Ratio =0.916

2.b With K=3

Repeat the above procedure with `result<- kmeans(mydata,3)`



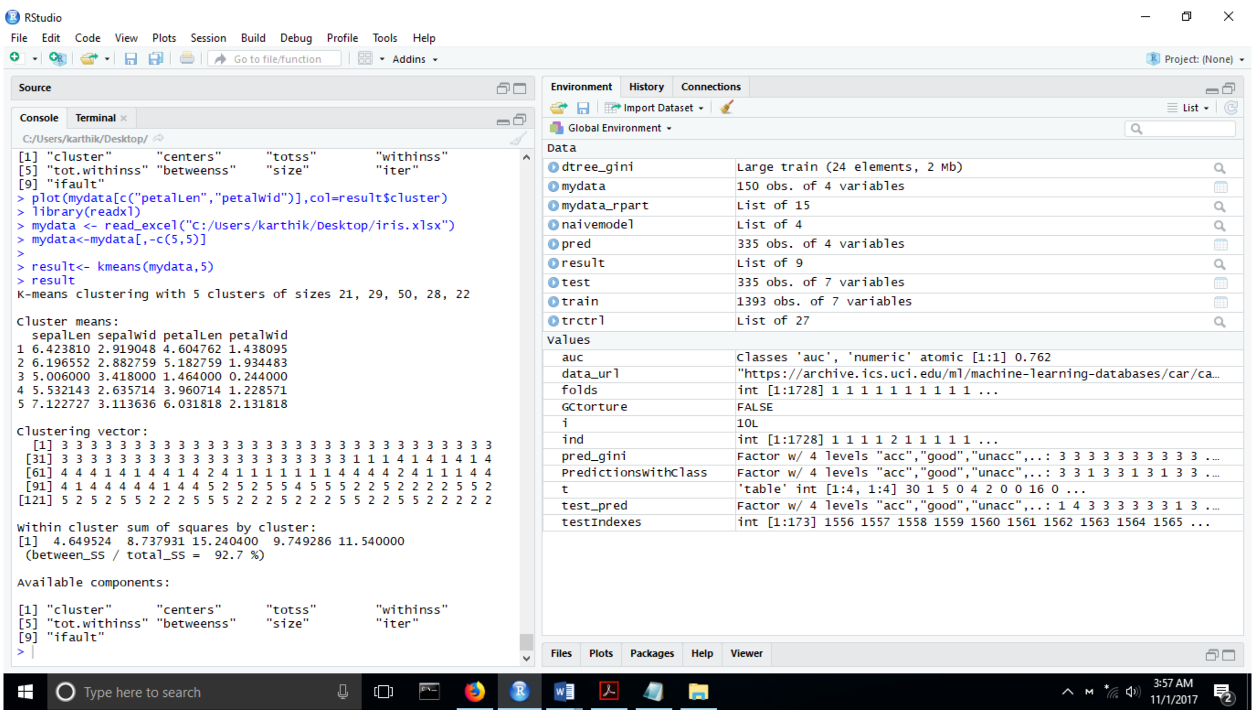
Plotting the kmeans result



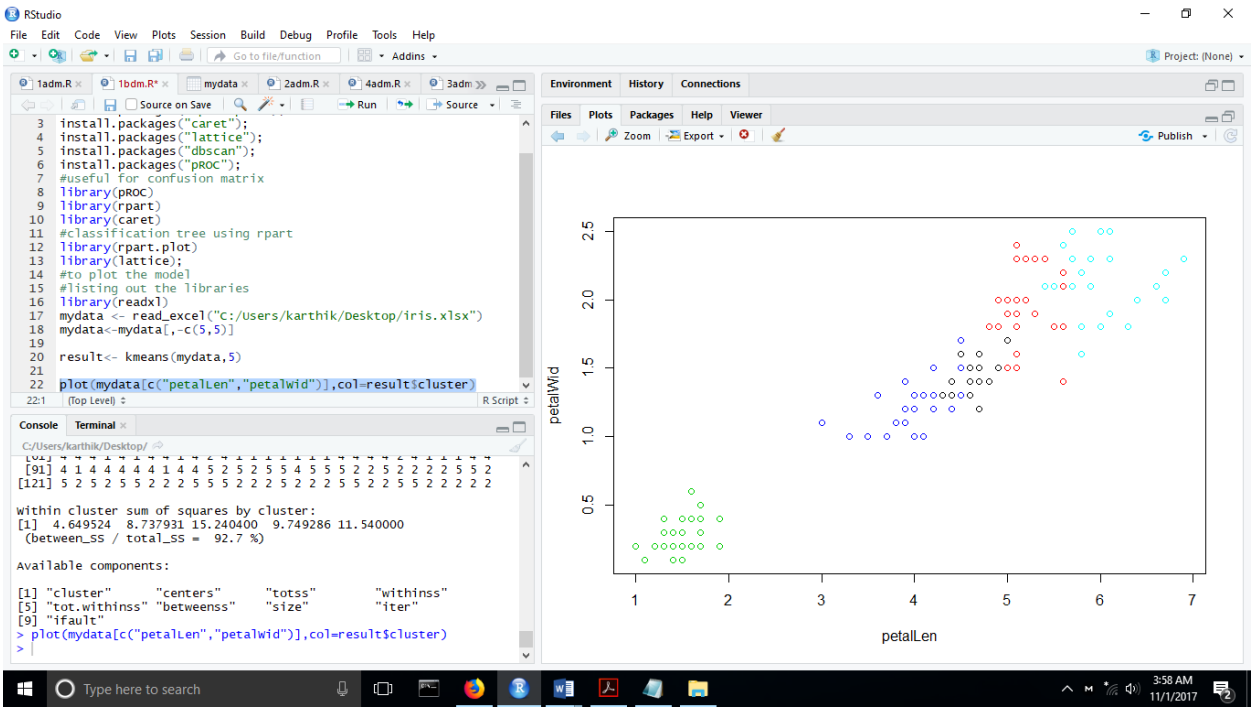
BSS/TSS Ratio =0.884

2.c With K=5

Repeat the above procedure with `result<- kmeans(mydata,5)`



Plotting the kmeans result



BSS/TSS Ratio =0.927

CASE	BSS/TSS
K=4	0.916
K=3	0.884
K=5	0.927

Hence the best cluster is given by k=5

The measure of the goodness of the classification k-means is based on the ratio between the between Sum of squares and the total Sum of Squares

Ideally you want a clustering that has the properties of internal cohesion and external separation, i.e. the BSS/TSS ratio should approach 1.

With-in-Sum-of-Squares (WSS): WSS is the total distance of data points from their respective cluster centroids

Total-Sum-of-Squares (TSS): TSS is the total distance of data points from global mean of data, for a given dataset this quantity is going to be constant

Between-Sum-of-Squares (BSS): BSS is the total weighted distance of various cluster centroids to the global mean of data

Plotting the silhouette

The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation).

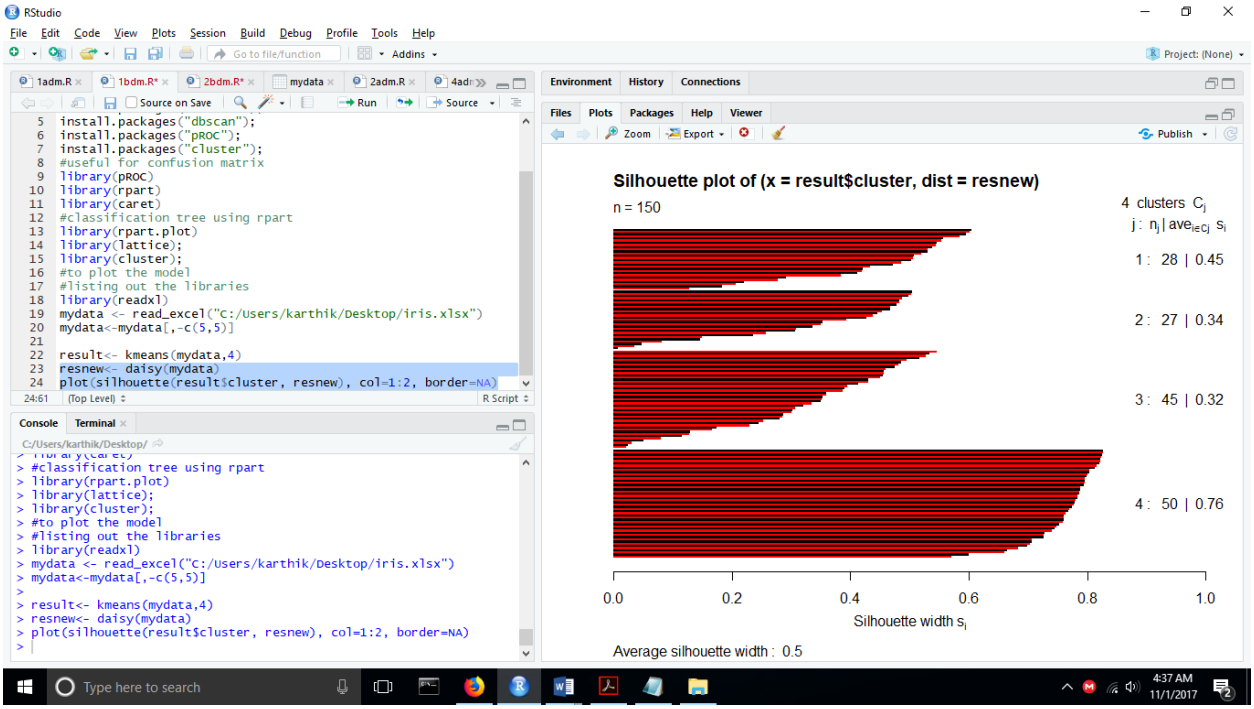
To use the daisy function we have to install the cluster package

We plot the silhouette by

resnew<- daisy(mydata)

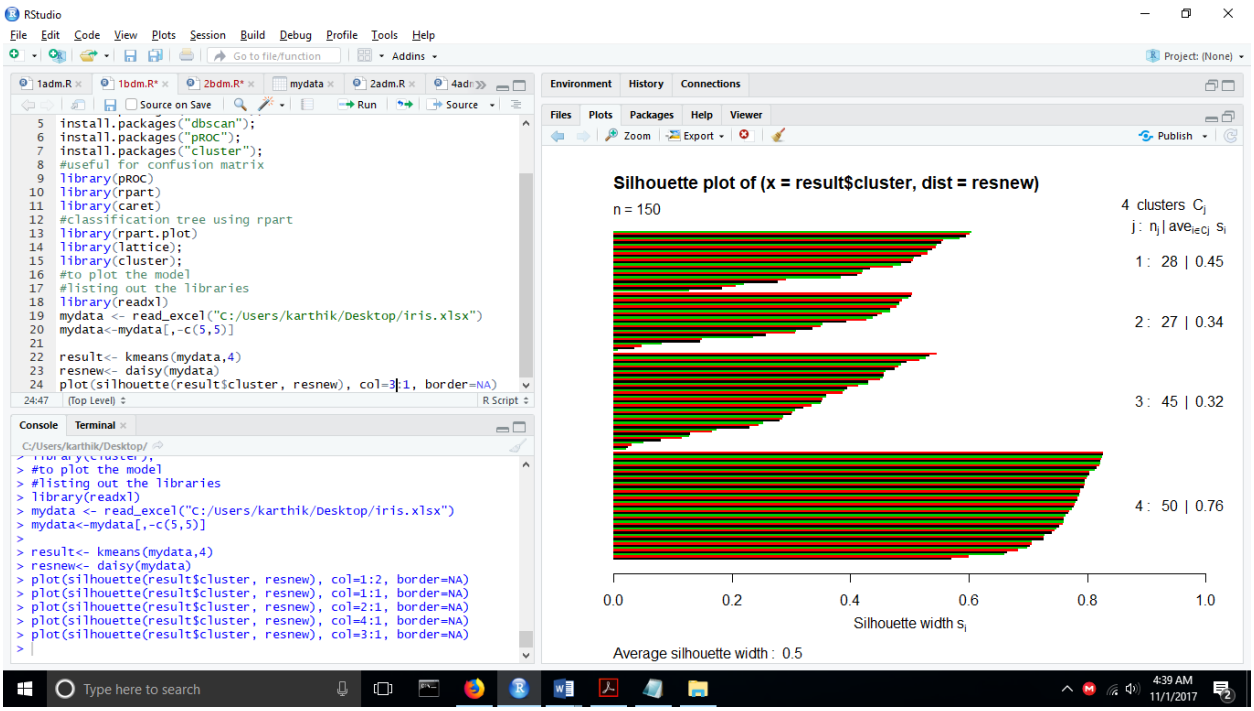
plot(silhouette(result$cluster, resnew), col=1:2, border=NA)

Dependency based on two of the four clusters



plot(silhouette(result$cluster, resnew), col=3:1, border=NA)

Dependency based on three of the four clusters

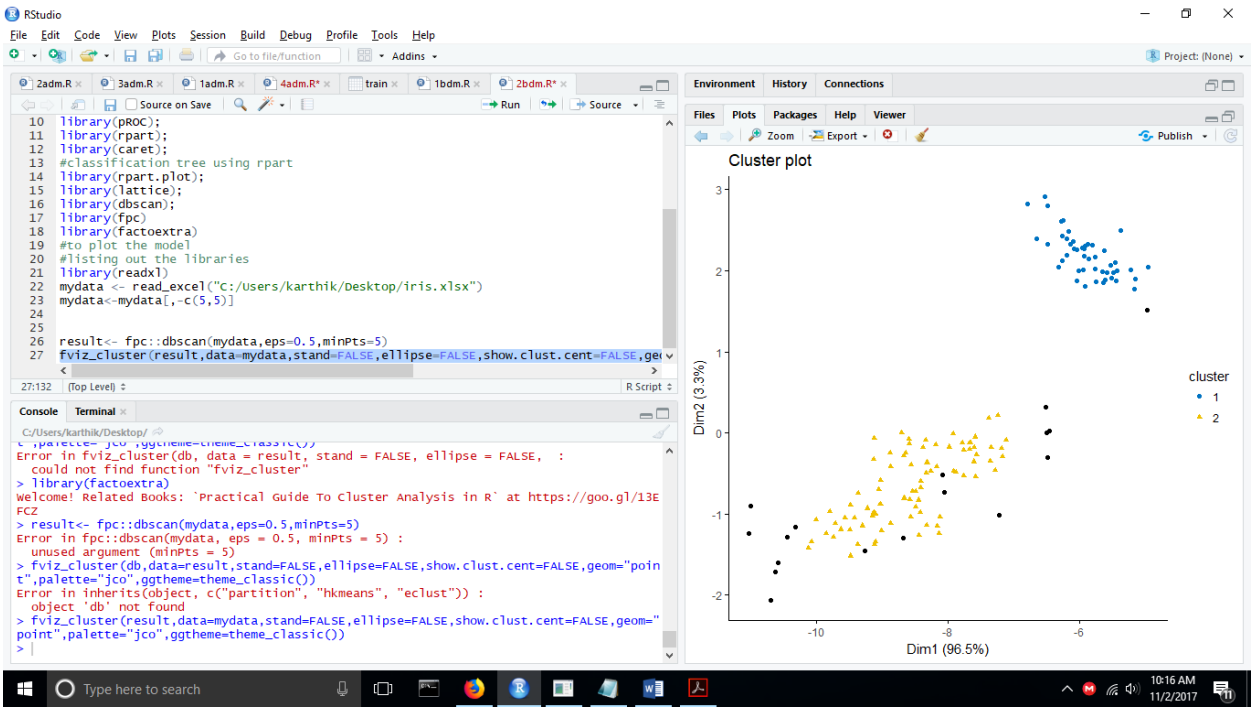


2.d Density based Clustering (DBSCAN)

Install the DBSCAN PACKAGE, fpc and factoextra packages

result<- fpc::dbscan(mydata,eps=0.5,minPts=5)

fviz_cluster(result,data=mydata,stand=FALSE,ellipse=FALSE,show.clust.cent=FALSE,geom="point",palette="jco",ggtheme=theme_classic())



The ouliers are presented by the black dots.

Review of case A and case D:

1.Unlike KMEANS, DBSCAN does not require user to mention the number of clusters to be generated.

2.DBSCAN can find any types of clusters. The clusters need not be circular.

3. DBSCAN can identify outliers.