Question 4.1

Describe a situation or problem from your job, everyday life, current events, etc., for which a clustering model would be appropriate. List some (up to 5) predictors that you might use.

**Answer:**

Supplier clustering:

Suppose we are running a plant, we need to divide our current suppliers for a single type of material into clusters in order to better integrate resources to optimize our put-away and manufacture process, as well as reduce cost.

**Table 1   Predictors in the Suppliers Clustering Model**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data Type** | **Reason** |
| **Delivery-time** | = Length of time from the date the order is made to the time of receiving material(often in days) | Ratio | The suppliers whose delivery time is relatively longer will have a negative effect on the production |
| **On Time Delivery Rate** | = (The actual material delivered on time by quantity / total material delivered confirmed by order) \*100% | Ratio | Suppliers with lower on-time delivery rate will increase uncertainty |
| **Qualification Rate** | = (Qualified incoming batch/total incoming batch)\*100% | Ratio | If the supplier has relatively low qualification rate, which means the materials are damaged and can’t be used, it will add extra costs |
| **Price Level** | Different suppliers can offer various price for the same type of material. | Ratio | Obviously suppliers who offers high price level material will endanger the actual profit rate |

Question 4.2

The *iris* data set iris.txt contains 150 data points, each with four predictor variables and one categorical response. The predictors are the width and length of the sepal and petal of flowers and the response is the type of flower. The data is available from the R library datasets and can be accessed with iris once the library is loaded. It is also available at the UCI Machine Learning Repository (<https://archive.ics.uci.edu/ml/datasets/Iris> ). *The response values are only given to see how well a specific method performed and should not be used to build the model.*

Use the R function kmeans to cluster the points as well as possible. Report the best combination of predictors, your suggested value of k, and how well your best clustering predicts flower type.

# Answer:

**First of all, we need to find the best k value.**

**Using 3 different ways:**

1. Using function: pamk

Fuction pamk calls the function pam or clara to perform a partitioning around medoids clustering with the number of clusters estimated by optimum average silhouette width.

"silwidth": Silhouette Coefficient to measure how similar an object is to its own cluster (cohesion) compared to other clusters (separation))

pamk.best1=pamk(data2[,c(1,2)])$nc

pamk.best2=pamk(data2[,c(1,3)])$nc

pamk.best3=pamk(data2[,c(1,4)])$nc

pamk.best4=pamk(data2[,c(2,3)])$nc

pamk.best5=pamk(data2[,c(2,4)])$nc

pamk.best6=pamk(data2[,c(3,4)])$nc

pamk.best7=pamk(data2[,c(1,2,3)])$nc

pamk.best8=pamk(data2[,c(1,2,4)])$nc

pamk.best9=pamk(data2[,c(1,3,4)])$nc

pamk.best10=pamk(data2[,c(2,3,4)])$nc

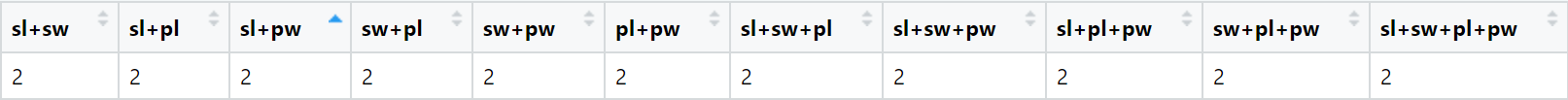
pamk.best11=pamk(data2[,c(1,2,3,4)])$nc

k\_result <- matrix(c(pamk.best1,pamk.best2,pamk.best3,pamk.best4,pamk.best5,pamk.best6,pamk.best7,pamk.best8,pamk.best9,pamk.best10,pamk.best11),nrow=1,ncol=11)

colnames(k\_result) <- c('sl+sw','sl+pl','sl+pw','sw+pl','sw+pw',

'pl+pw','sl+sw+pl','sl+sw+pw','sl+pl+pw',

'sw+pl+pw','sl+sw+pl+pw')



the result matrix shows that no matter which combination of features we choose, the best k value is 2.

2. Using fuction: NbCluster

library(NbClust)

nb\_clust <- NbClust(data2, distance='minkowski',min.nc=2,max.nc=10,method = 'kmeans',index='alllong', alphaBeale=0.1)

The result shows the best k value is 2.

3. Applying elbow method, calculating sum of squared error.

We use the enumeration approach to explore the best combination of predictors(Sepal.Length, Sepal.Width, Petal.Length and Petal.Width) and number of clusters. Specificlly, the range of k in our trails is 2 to 10, and all combinations of the four predictors are included in our trails. Then we draw the line plot, whose horizontal axis is k values and vertical axis is the total within-cluster sum of squares.

#elbow method, using sum of squared error

wssplot <- function(data,nc=10,seed=1234){

wss <- (nrow(data)-1)\*sum(apply(data,2,var))

for (i in 2:nc){

set.seed(seed)

wss[i] <- sum(kmeans(data,centers=i)$withinss)

}

plot(1:nc,wss,type='b',xlab='number of clusters',

ylab='Within groups sum of squares' )

}

wssplot(data2)

wssplot\_2 <- function(data,color\_type){

wss <- (nrow(data)-1)\*sum(apply(data,2,var))

for (i in 2:10){

set.seed(1234)

wss[i] <- sum(kmeans(data,centers=i)$withinss)

}

lines(1:10,wss,type='b',col=color\_type,xlab='number of clusters',

ylab='Within groups sum of squares' )

}

wssplot\_2(data2[,c(1,2)],2)

wssplot\_2(data2[,c(1,3)],3)

wssplot\_2(data2[,c(1,4)],4)

wssplot\_2(data2[,c(2,3)],5)

wssplot\_2(data2[,c(2,4)],6)

wssplot\_2(data2[,c(3,4)],7)

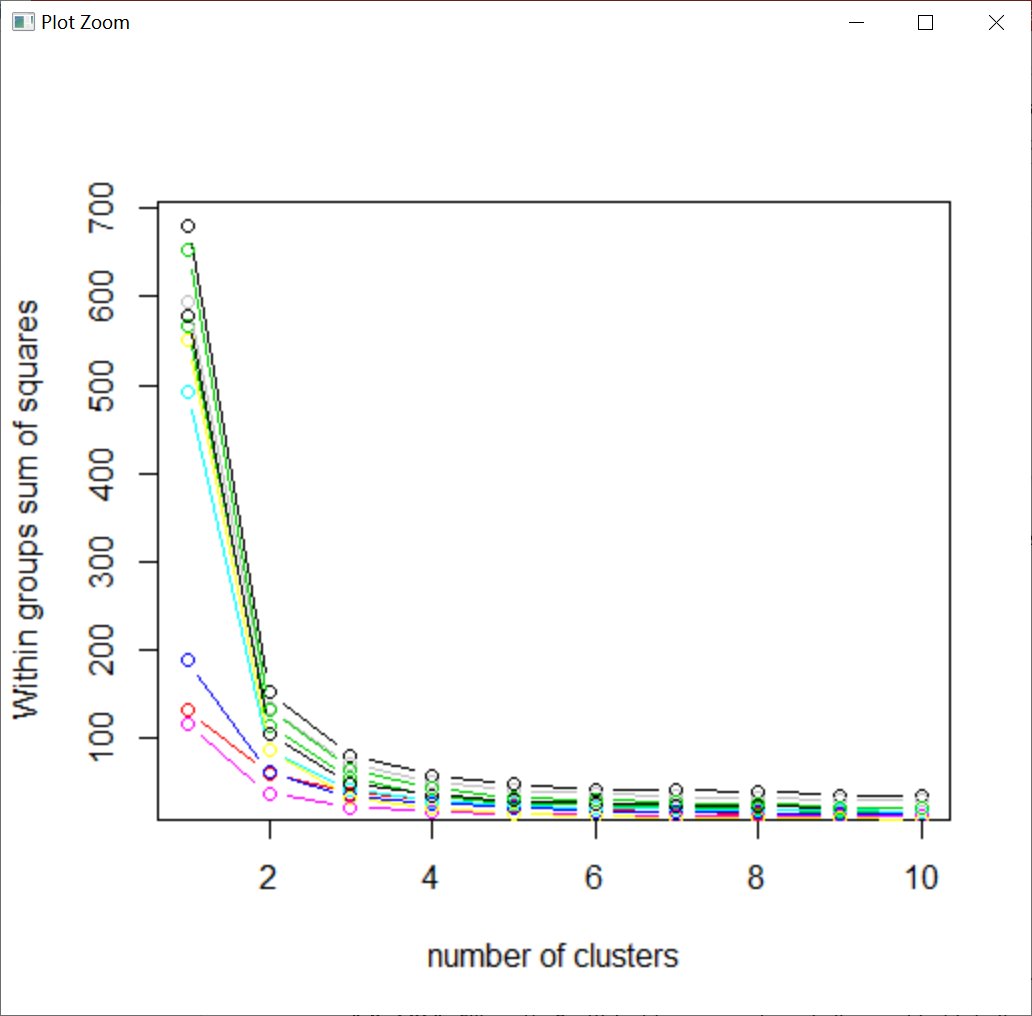
wssplot\_2(data2[,c(1,2,3)],8)

wssplot\_2(data2[,c(2,3,4)],9)

wssplot\_2(data2[,c(1,2,4)],10)

wssplot\_2(data2[,c(1,3,4)],11)

Figure 1 shows the change in SSE value when the number of clusters increases.



**Figure 1**

In this case k=2 or k=3 are both acceptable k values. And wssplot\_2(data2[,c(2,4)],6) has the least in-cluster total distance, which means that

the model with features petal.length

and petal.width has the least in-cluster total distance.

Thus we tried both k=2 and k=3 in order to find the best cluster method.

model <- kmeans(data2,k\_value)

pred <- model$cluster

result\_matrix <- matrix(c(data[,5],pred),nrow=150,ncol=2)

In order to find out how well our predictor is, we introduced following 3 parameters:

**"silwidth":** Silhouette Coefficient to measure how similar an object is to its own cluster (cohesion) compared to other clusters (separation))

**"average.within"**: the average distance between factors within the same cluster

**"average.between":** the average distance between different clusters

ave\_sil1=km\_stats1$avg.silwidth

ave\_with1=km\_stats1$average.within

ave\_bet1=km\_stats1$average.between

|  |  |  |  |
| --- | --- | --- | --- |
|  | **silwidth** | **average.within** | **average.between** |
| **k=2** | 0.6810462 | 1.2172417 | 4.022225 |
| **k=3** | 0.5528190 | 0.9187411 | 3.386163 |