Exercise 7

191820019 陈文杰

2020/12/6

R Markdown

#一、分别采用至少5种聚类算法对“鸢尾花”数据集进行分析 解：

##[策略]

（1）读取iris数据，进行数据概览，并作预处理（将分类变量Species因子化）

（2）因为用于聚类的数据不能存在double以外的数据类型，删去iris的Species列

（3）完成聚类模型构建，用到的包与函数如下：

```
①cluster包的kmeans函数  → K-means聚类
②cluster包的pam函数    → K-mediods聚类
③cluster包的hclust函数     →  层次聚类（Agnes）
④cluster包的diana函数    → 层次聚类（DIANA）
⑤fpc包的dbscan函数     →  密度聚类（DBSCAN）
⑥mclust包的Mclust函数  →   期望最大化聚类（EM）
```

（4）由于聚类结果难以像分类算法那样普遍利用Accuracy、Precision、Recall、F1-Scores等参数进行模型评估，故利用混淆矩阵（部分函数可用）、密度函数绘制、聚类结果可视化等方式来大体查看模型效果。

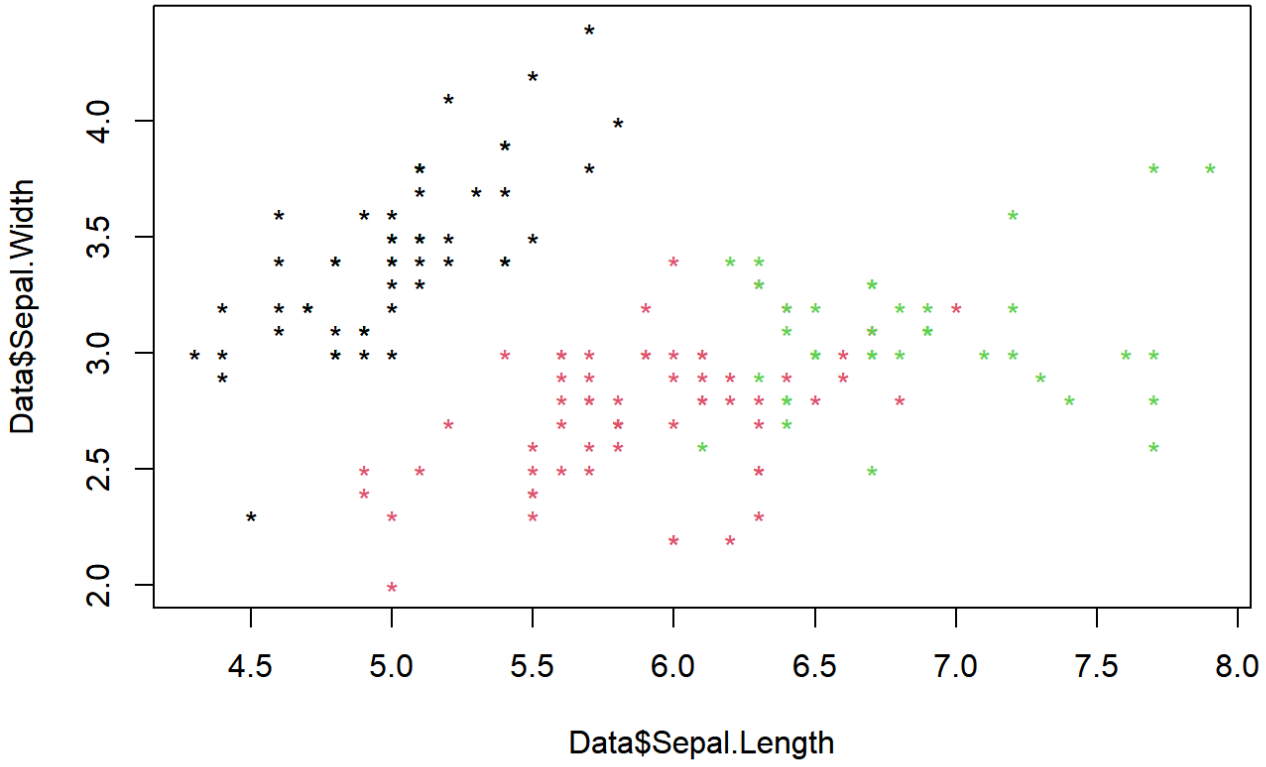
##[过程|结果]

1、K-means

```
library(cluster)

Data <- iris[1:4]

km <- kmeans(Data,center = 3)


#查看聚类模型
# print(km)

#混淆矩阵查看聚类结果
(table(actual=iris$Species,predictedclass=km$cluster))
```

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

```
#聚类结果可视化
par(mfrow = c(1, 1))
plot(Data$Sepal.Length,Data$Sepal.Width, col=km$cluster,pch="*")
```

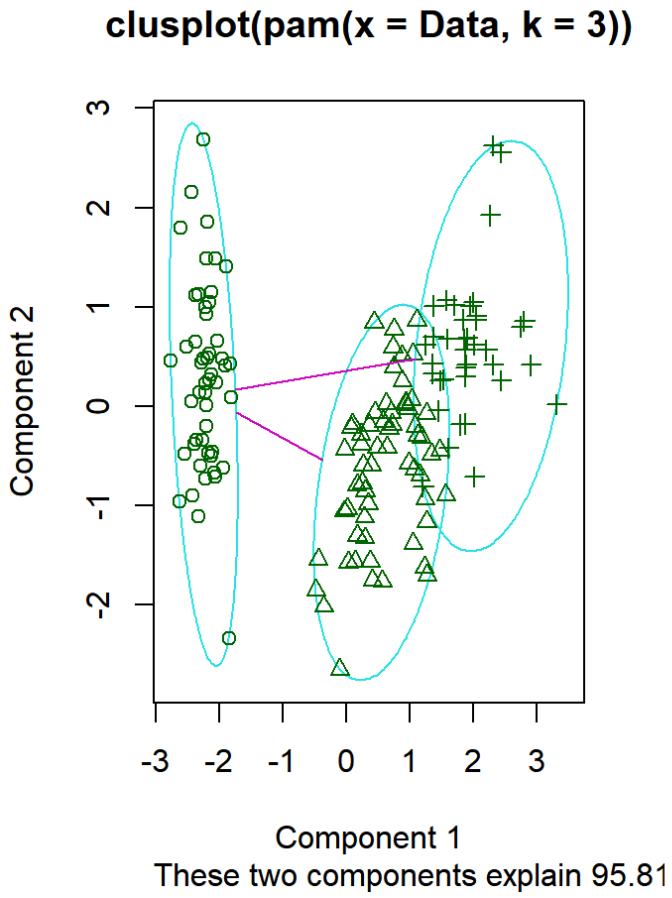


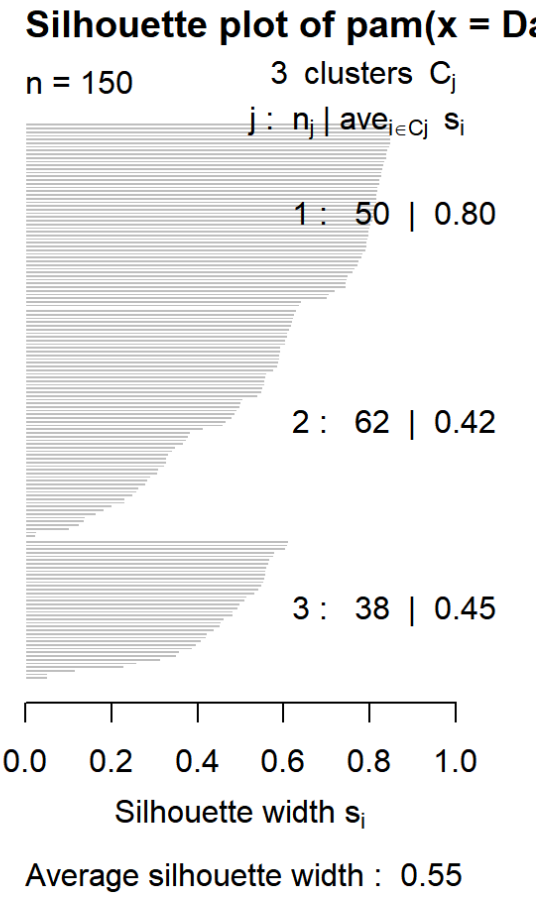
2、K-Mediods

```
Data <- iris[1:4]

pam <- pam(Data,3)


layout(matrix(c(1,2),1,2)) #页面布局调整

#混淆矩阵查看聚类结果
(table(actual=iris$Species,predictedclass=pam$cluster))
```

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

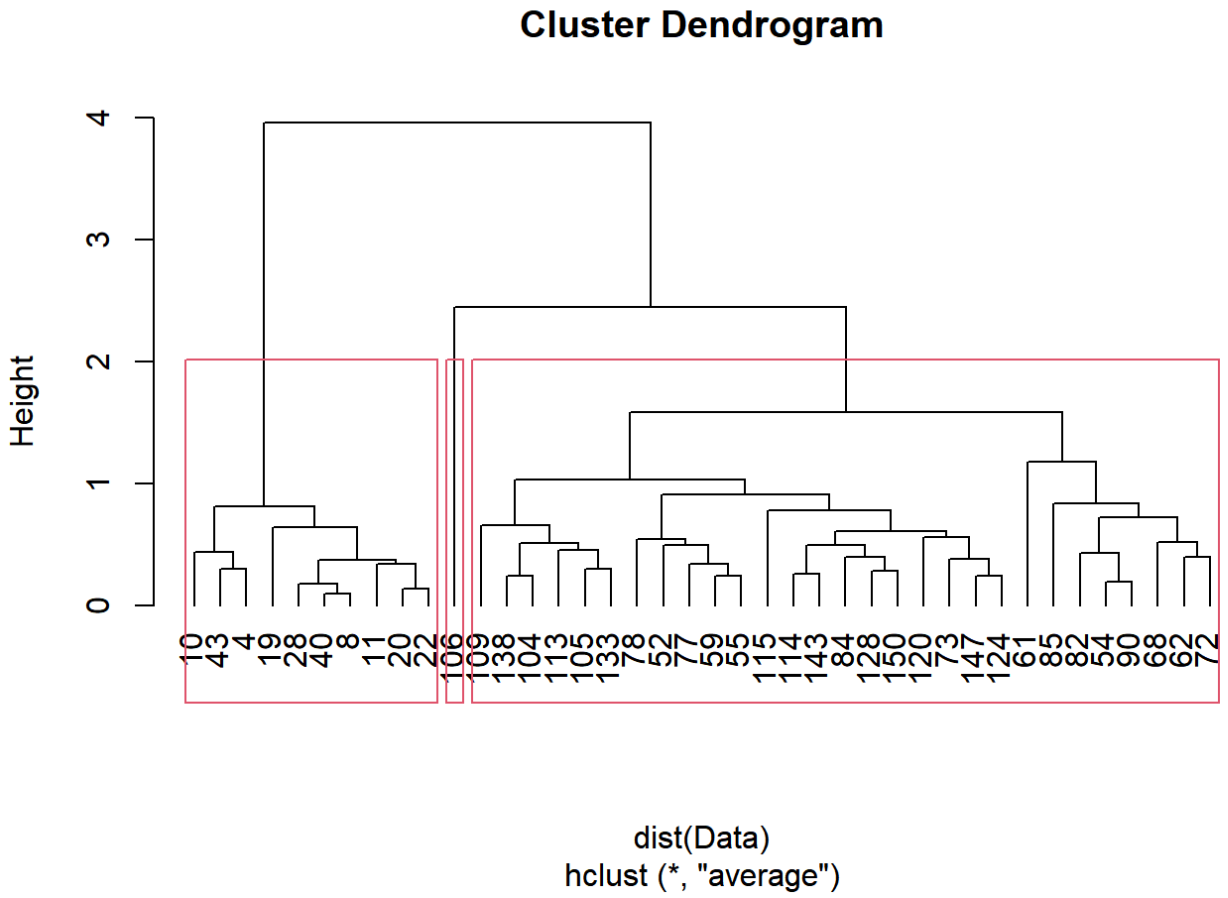
```
plot(pam)
```





3、层级聚类（AGNES）

```
#因为样本数量过多，最终图形显示可能不太友好，于是抽样示例

#dim(iris)#返回行列数

idx<-sample(1:dim(iris)[1],40)#sample的前一个参数代表从哪里取，第二个参数代表取多少个

Data<-iris[idx,-5]


hc<-hclust(dist(Data),method = "ave")  #注意hcluster里边传入的是dist返回值对象

layout(matrix(c(1,1)))

plot(hc,hang=-1)  #这里的hang=-1使得树的节点在下方对齐

#将树分为3块
rect.hclust(hc,k=3)
```



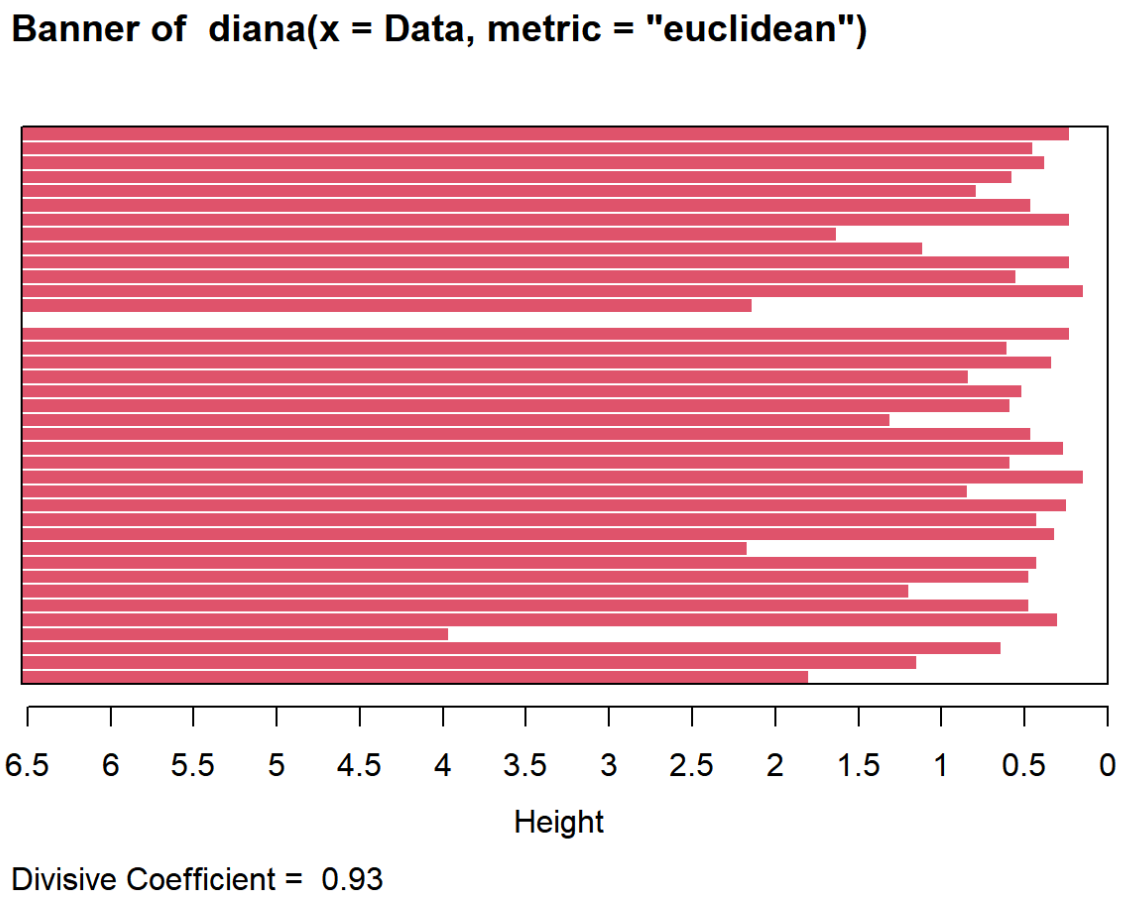

```
groups<-cutree(hc,k=3)
```

4、层次聚类（DIANA）

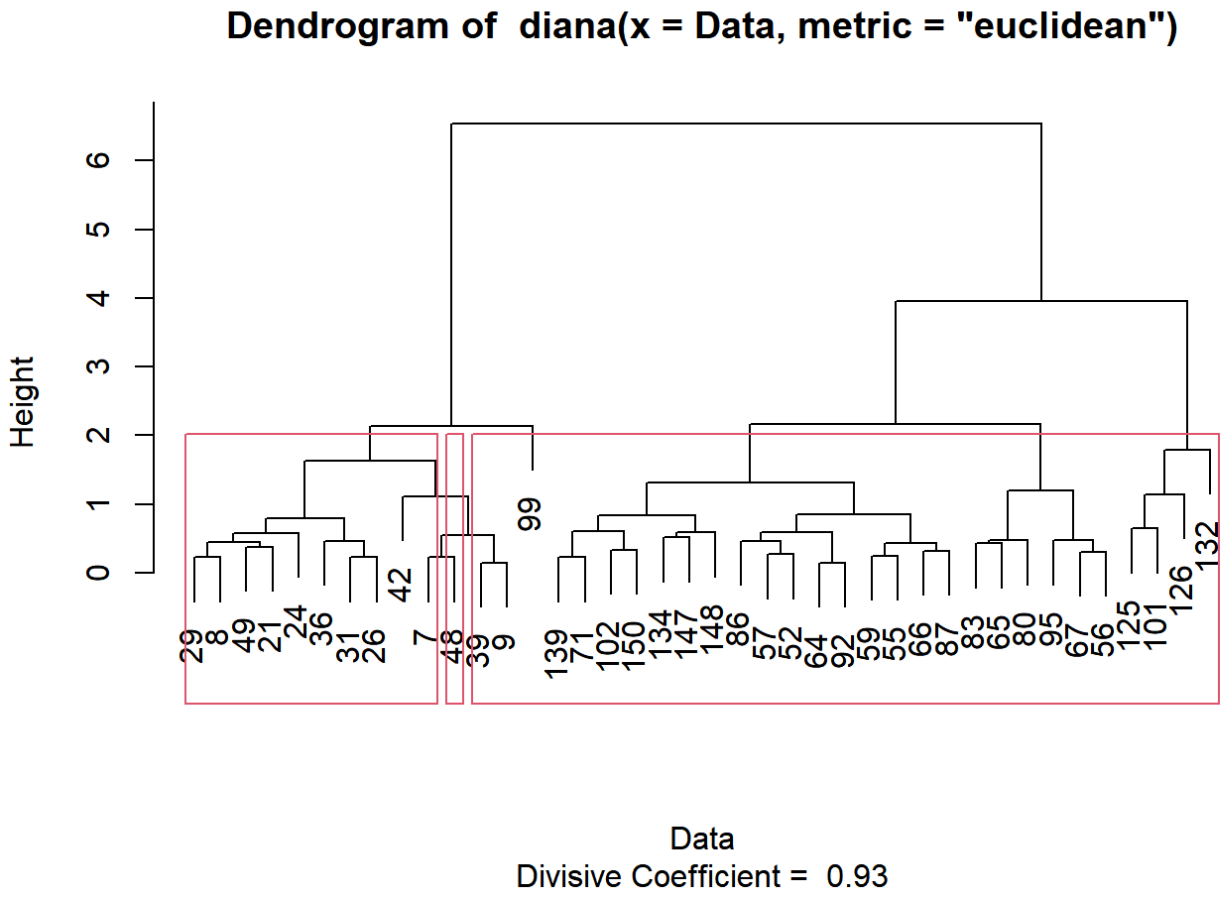
```
library(cluster)

#因为样本数量过多，最终图形显示可能不太友好，于是抽样示例

dim(iris)#返回行列数
```

```
## [1] 150   5
```

```
idx<-sample(1:dim(iris)[1],40)#sample的前一个参数代表从哪里取，第二个参数代表取多少个

Data<-iris[idx,-5]

dv = diana(Data,metric="euclidean")

plot(dv)
```




```
#将树分为3块
rect.hclust(hc,k=3)
```



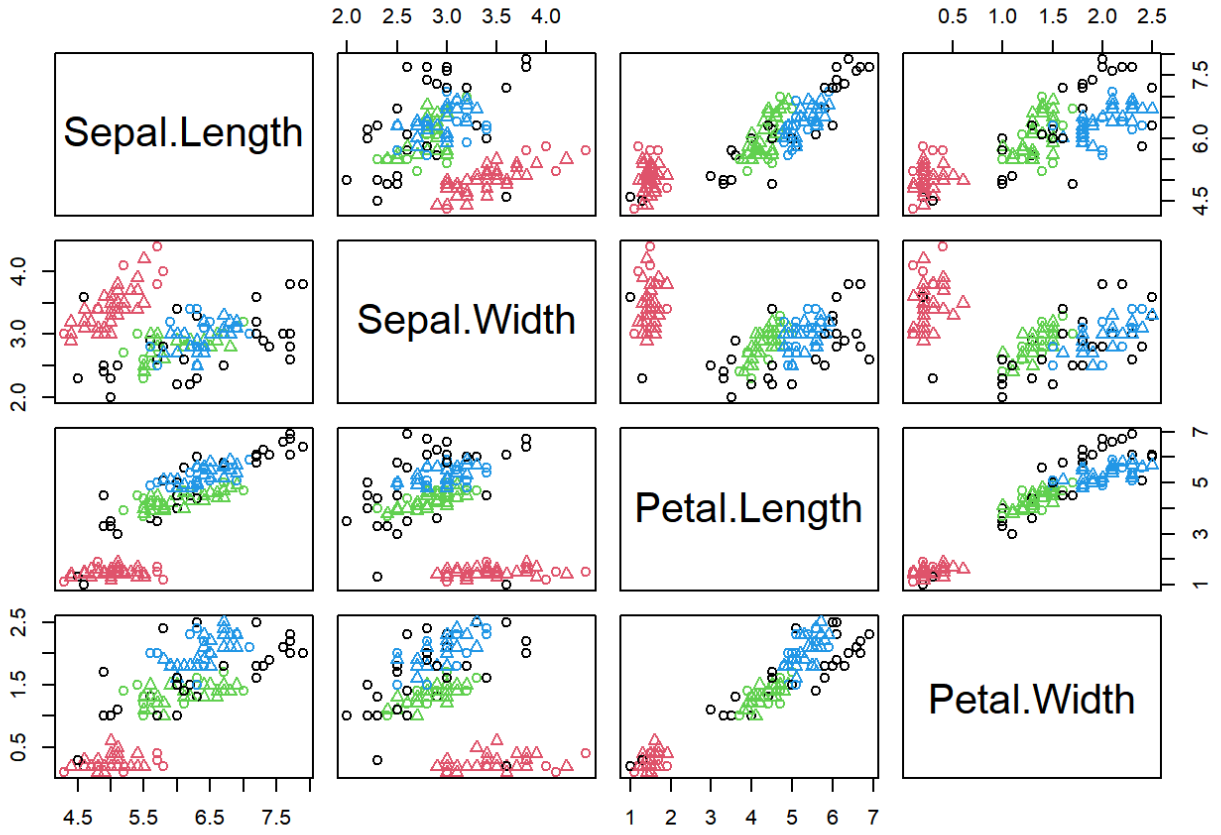
```
groups<-cutree(hc,k=3)
```

5、密度聚类（DBSCAN）

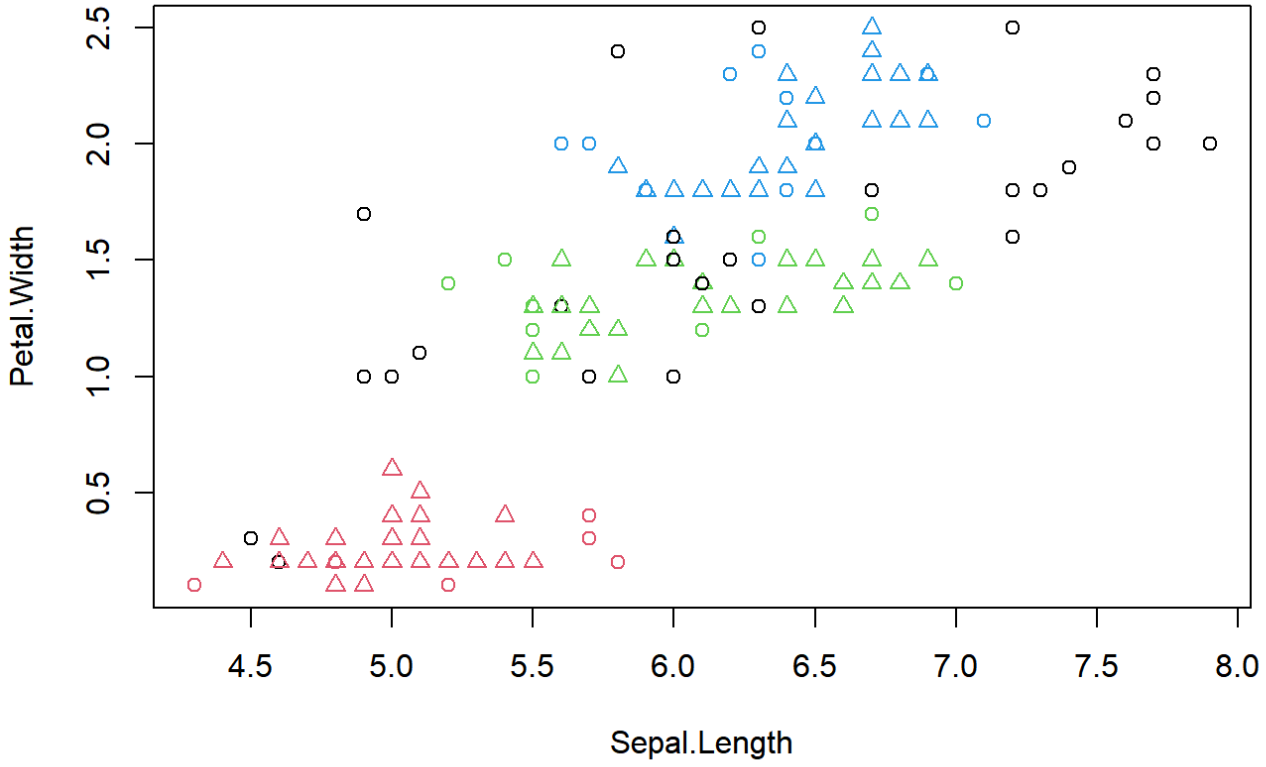
```
library(fpc)

Data<-iris[-5]

ds <- dbscan(Data,eps=0.42,MinPts=5)

table(actual=iris$Species,predictedclass=ds$cluster)
```

```
##             predictedclass
## actual         0  1  2  3
##   setosa       2 48  0  0
##   versicolor 10  0 37  3
##   virginica  17  0  0 33
```

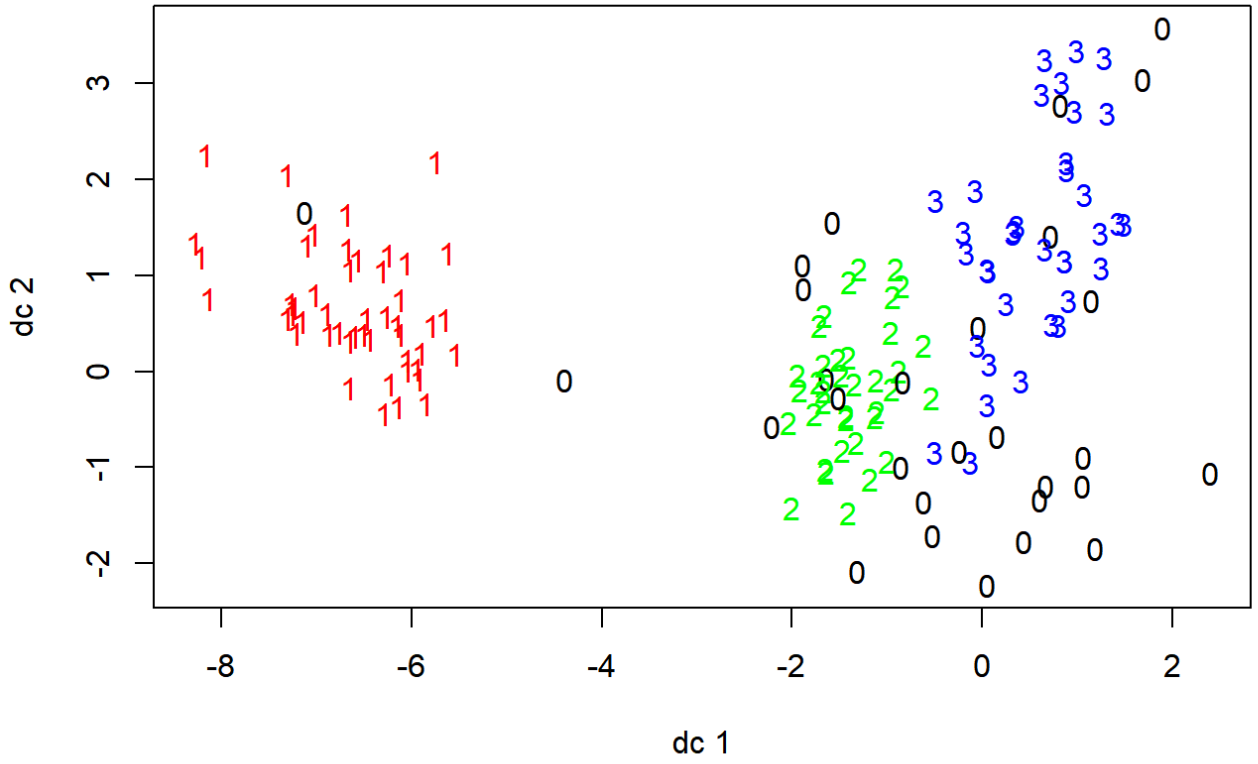
```
plot(ds,Data)
```




```
#打印出iris第一列和第四列为坐标轴的聚类结果
plot(ds,Data[,c(1,4)])
```




```
#另一个表示聚类结果的函数，plotcluster
plotcluster(Data,ds$cluster)
```



6、EM聚类算法/期望最大化聚类

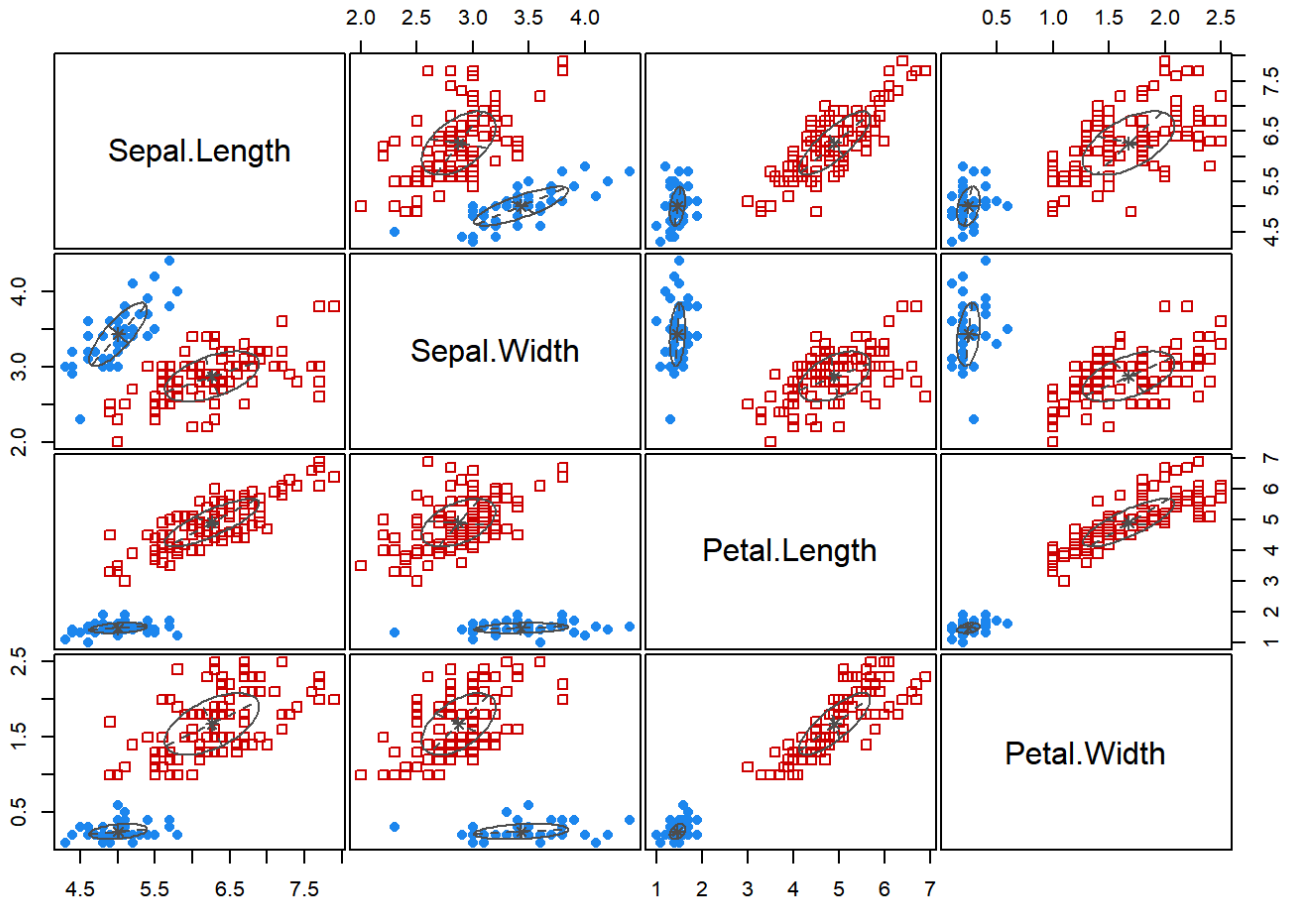
```
# 加载mclust包
library(mclust)
```

```
## Package 'mclust' version 5.4.7
## Type 'citation("mclust")' for citing this R package in publications.
```

```
set.seed(123)  #设置随机种子号

Data <- iris[-5]

EM <- Mclust(Data)


# 查看模型建模结果
summary(EM, parameter = TRUE)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEV (ellipsoidal, equal shape) model with 2 components:
##
##  log-likelihood   n df       BIC       ICL
##        -215.726 150 26 -561.7285 -561.7289
##
## Clustering table:
##   1   2
##  50 100
##
## Mixing probabilities:
##         1         2
## 0.3333319 0.6666681
##
## Means:
##                   [,1]     [,2]
## Sepal.Length 5.0060022 6.261996
## Sepal.Width  3.4280049 2.871999
## Petal.Length 1.4620007 4.905992
## Petal.Width  0.2459998 1.675997
##
## Variances:
## [,,1]
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length   0.15065114  0.13080115   0.02084463  0.01309107
## Sepal.Width    0.13080115  0.17604529   0.01603245  0.01221458
## Petal.Length   0.02084463  0.01603245   0.02808260  0.00601568
## Petal.Width    0.01309107  0.01221458   0.00601568  0.01042365
## [,,2]
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    0.4000438  0.10865444    0.3994018  0.14368256
## Sepal.Width     0.1086544  0.10928077    0.1238904  0.07284384
## Petal.Length    0.3994018  0.12389040    0.6109024  0.25738990
## Petal.Width     0.1436826  0.07284384    0.2573899  0.16808182
```

```
#绘制聚类结果概率分布图
plot(EM, what = "classification")
```




```
table(actual=iris$Species,predictedclass=EM$classification)
```

```
##             predictedclass
## actual        1  2
##   setosa     50  0
##   versicolor  0 50
##   virginica   0 50
```

#二、观察聚类结果中的误差，尝试分析误差产生的原因与改进措施。

1、K-means算法 （1）模型概述 k-means算法以k为参数，把n个对象分成k个簇，使簇内具有较高的相似度，而簇间的相似度较低。 （2）误差分析 在iris数据中集中，通过观察混淆矩阵发现，versicolor 、virginica的聚类区分效果不是很好，原因可能在于k-means的 固有缺陷：①初始聚类中心选择对聚类结果的影响；②非球状数据 难以聚类；③对孤立点和噪声比较敏感 （3）改进措施 采用k-mediods算法

2、K-mediods算法 （1）模型概述 选取有代表性的样本（而不是均值）来表示整个簇，即：选取最靠近中心点(medoid)的那个样本来代表整个簇。以降低聚类算法对离群点的敏感度。 （2）误差分析 versicolor 、virginica的聚类区分仍存在部分误差，但相对于k-means而言更精准，且更鲁棒。 （3）改进措施 采用k-mediods算法

3、层次聚类（Agnes、Diana）算法 （1）模型概述 自底向上方法（合并）：开始时，将每个样本作为单独的一个组；然后，依次合并相近的样本或组，直至所有样本或组被合并为一个组或者达到终止条件为止。 代表算法：AGNES算法 自顶向下方法（分裂）：开始时，将所有样本置于一个簇中；然后，执行迭代，在迭代的每一步中，一个簇被分裂为多个更小的簇，直至每个样本分别在一个单独的簇中或者达到终止条件为止。 代表算法：DIANA算法 （2）误差分析 根据聚簇图，观测到聚类效果较好 （3）改进措施 尝试不同的距离度量与相似性衡量method：min、max、ward.D2等

4、EM聚类算法 （1）模型概述 它将数据集看作一个含有隐性变量的概率模型,并以实现模型最优化,即获取与数据本身性质最契合的聚类方式为目的,通过”反复估计”模型参数找到最优解,同时给出相应的最优类别k.而”反复估计”的过程即是EM算法的精华所在,这一过程由E-step(Expectation)和M-step(Maximization)两个步骤交替进行来实

现。 。

（2）误差分析 难以控制划分的类别数，在iris数据集中，所有数据被划分成了2类。

5、密度聚类（DBSCAN）算法 （1）模型概述 DBSCAN 是一种简单、有效的基于密度的聚类算法.(Density-Based Spatial Clustering of Applications with Noise)。在基于中心的方法中，数据集中特定点的密度通过对该点Eps半径之内的点计数（包括点本身）来估计。

（2）误差分析 ①难以控制划分的类别数，在iris数据集中，所有数据被划分成了4类。②当簇的密度变化太大时，DBSCAN就会有麻烦。对于高维数据，它也有问题，因为对于这样的数据，密度定义更困难。

（3）改进措施 提升聚类精度 → 半径缩小，但会让分类完整性↓ 降低聚类精度 → 半径扩大，分类完整性↑