**聚类分析**

目录

[一．系统聚类 1](#_Toc431224554)

[二．快速聚类（k均值聚类） 7](#_Toc431224555)

# 一．系统聚类

R中，系统聚类的函数为hclust()，dist()函数用来计算距离矩阵，plot()函数可以画出系统聚类的谱系图，rect.hclust()函数用来给定类的个数或给定阈值来确定聚类的情况。

(1)dist()的使用方法：dist(x,method="euclidean",diag=F,upper=F,p=2)

其中，x为数据矩阵或数据框。 method为计算方法，包括：euclidean（欧氏距离）、maximum（切比雪夫距离）、manhattan（绝对值距离）、nberra（兰氏距离）、minkoeski（明氏距离）。diag为是否包含对角线元素。upper为是否需要上三角。p为明氏距离的幂次。

(2)hclust()的使用方法：hclust(d,method="ward.D",….)

其中，d为距离矩阵。method为系统聚类方法：single（最短距离法）、complete（最长距离法，缺省）、average（类平均法）、median（中间距离法）、centroid（重心法）、ward.D（ward法）。

(3)plot()的使用方法：plot(x, labels = NULL, hang = 0.1,

axes = TRUE, frame.plot = FALSE, ann = TRUE,

main = "Cluster Dendrogram",

sub = NULL, xlab = NULL, ylab = "Height", ...)

其中，x是由hclust()函数生成的对象。hang是表明谱系图中各类所在的位置，当hang取负值时，谱系图中的类从底部画起。其他参数见帮助文档。

(4)rect.hclust()的使用方法：rect.hclust(tree, k = NULL, which = NULL, x = NULL, h = NULL,

border = 2, cluster = NULL)

其中，tree是由hclust()生成的结构。k是类的个数。h是谱系图中的阈值，要求分成的分成的各类的距离大于h。border是数或向量，表明矩形框的颜色。

**例1**：对以下股票进行分类，数据集：d9.1

x1：主营业务利润率 x2：销售毛利率 x3：速动比率 x4：资产负债率 x5：主营业务收入增长率 x6：营业利润增长率

命令：

> X=read.table("clipboard",header=T) #读取数据

> Z=scale(X) #对数据做标准化处理

> D=dist(Z) #计算距离矩阵

> hc=hclust(D,"ward.D") #ward聚类法

> cbind(hc$merge,hc$height) #显示聚类过程，-表示原样品，+表示新类

[,1] [,2] [,3]

[1,] -7 -11 1.112209 #将原来第7个样品和第11个样品合并为一类，叫做1

[2,] -2 -6 1.150279#将原来第2个样品和第6个样品合并为一类，叫做2

[3,] -3 -14 1.156805

[4,] -4 -9 1.202639

[5,] -1 1 1.646166#将原来第1个样品和新类1合并为一类，叫做5

[6,] -10 -12 2.221257

[7,] 3 4 2.311630

[8,] -13 6 3.343961

[9,] -8 5 3.835969

[10,] -5 7 3.969620

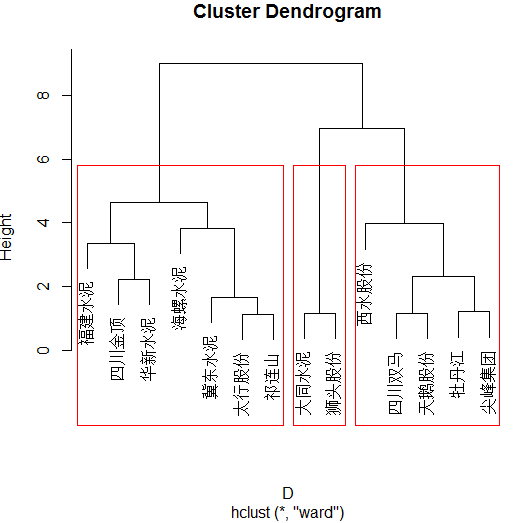
[11,] 8 9 4.630854

[12,] 2 10 6.978112

[13,] 11 12 9.014416

> plot(hc) #画聚类图

> rect.hclust(hc,k=3) #对聚类结果画框，k=3表示分3类



从分类图我们可以分三类：

第一类：福建水泥、四川金顶、华新水泥、海螺水泥、冀东水泥、太行股份、祁连山；

第二类：大同水泥、狮头股份；

第三类：西水股份、四川双马、天鹅股份、牡丹江、尖峰集团。

**例2：**为了研究我国31个省、市、自治区2001年城镇居民生活消费的分布规律，根据调查资料作区域消费类型划分。数据集：d7.2。

x1：人均食品支出 x2：人均衣着商品支出 x3：人均家庭设备用品及服务支出 x4：人均医疗保健支出 x5：人均交通和通信支出 x6：人均娱乐教育文化服务支出 x7：人均居住支出 x8：人均杂项商品和服务支出。

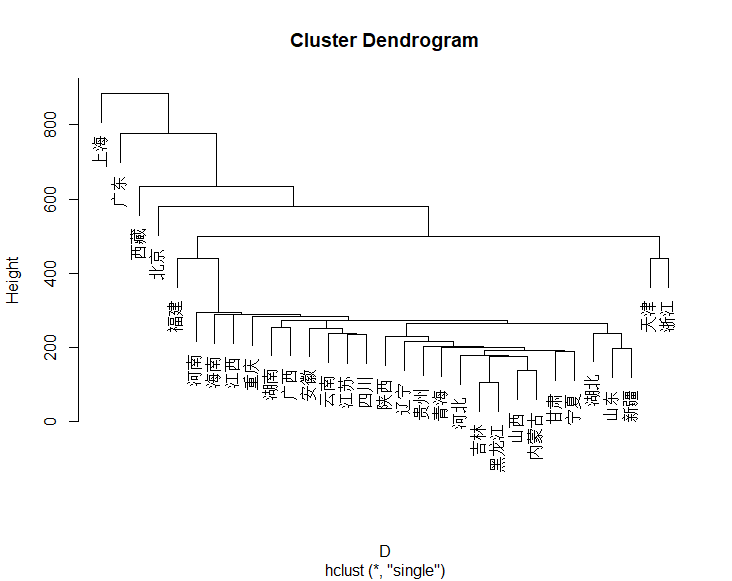
命令：

> X=read.table("clipboard",header=T) #读取数据

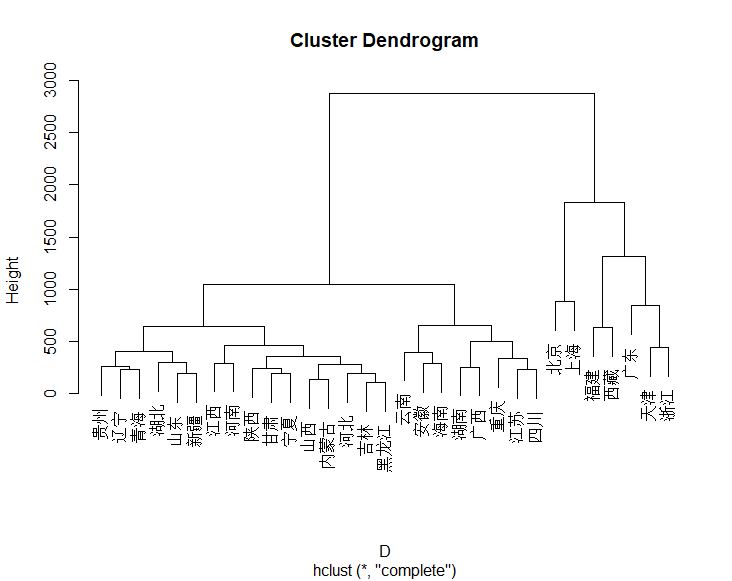
> D=dist(X)

> hc=hclust(D,"single")

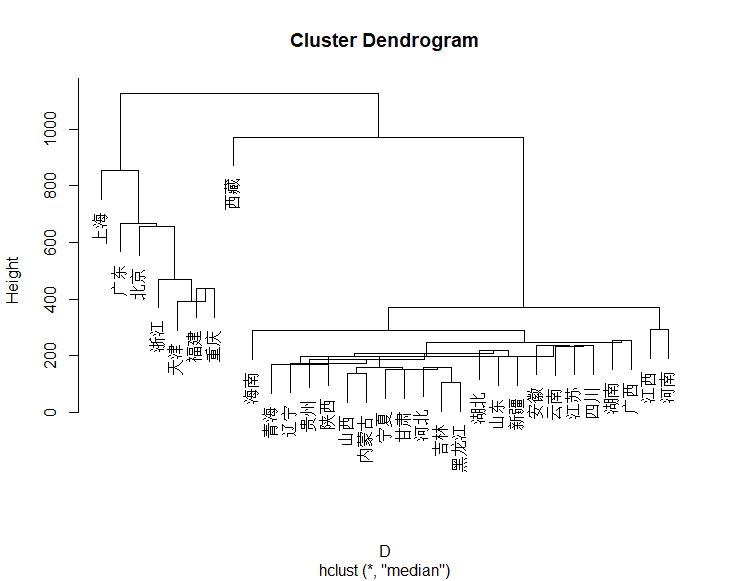
> plot(hc)



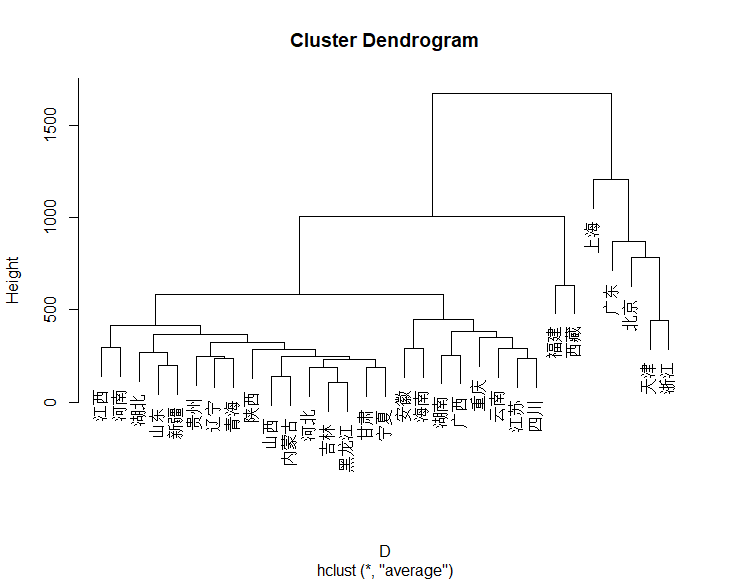
> plot(hclust(D,'complete')) #欧氏距离 最长距离法



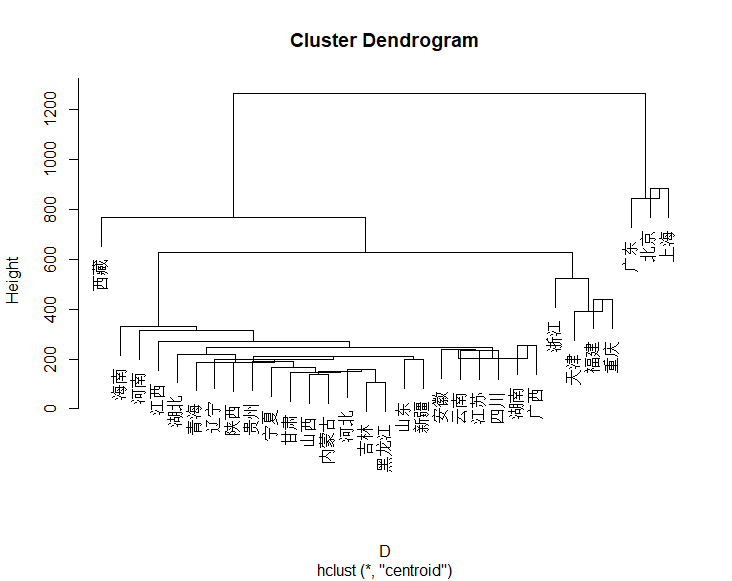
> plot(hclust(D,'median')) #欧氏距离 中间距离法



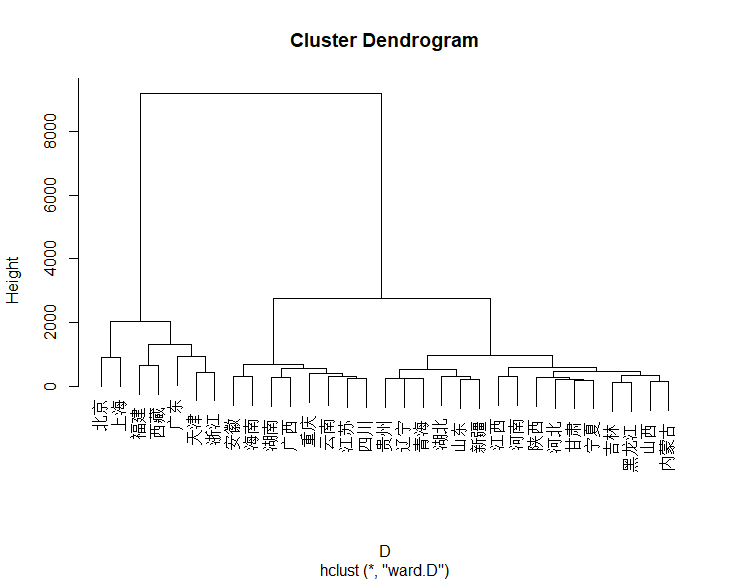
> plot(hclust(D,'average')) #欧氏距离 类平均法



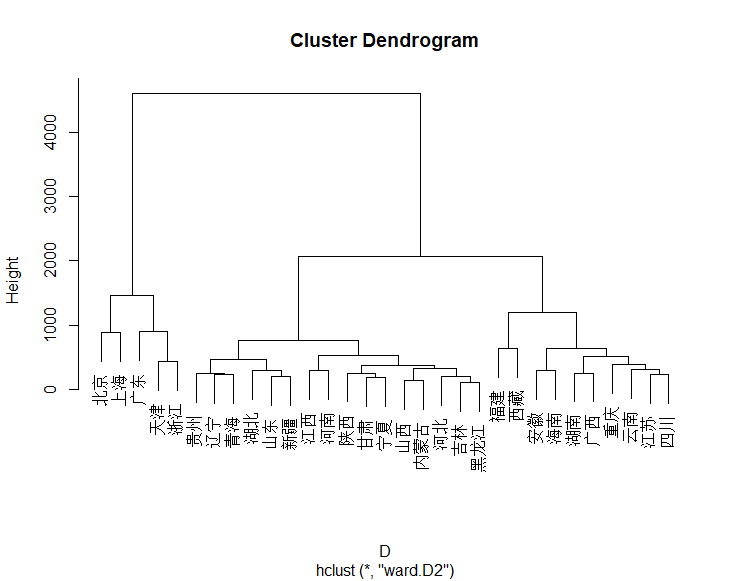
> plot(hclust(D,'centroid')) #欧氏距离 重心法



> plot(hclust(D,'ward.D')) #欧式距离 ward法



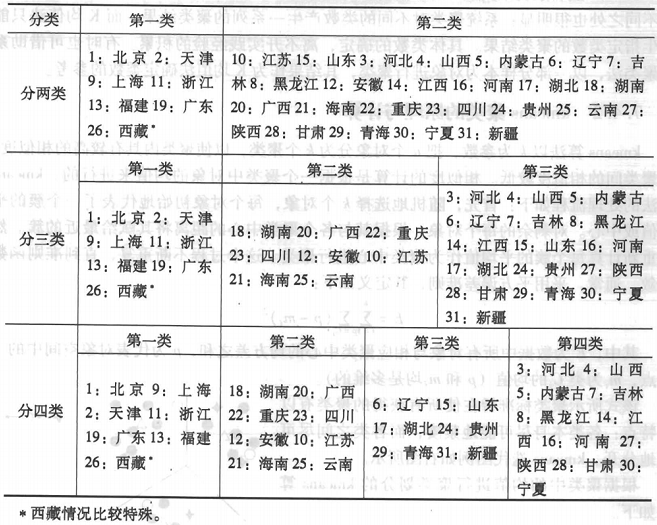
> plot(hclust(D,'ward.D2')) #欧式距离 ward法



将各种聚类方法进行对比，从中确定最好的聚类结果。

从直观上看，最短距离法分类效果最差，最长距离法和ward法分类效果较好。总的可分为三类：北京、上海、广东、浙江、天津为一类，为高消费地区；其余25个省份（不包括西藏，西藏情况比较特殊，自成一类）归为一大类，为中低消费区，可将该类进一步分类为中等消费区和低消费区。最长距离和类平均的分析结果基本上是相同的。

综合各种类型方法的分析结果，分为四类较为合适。



# 

# 二．快速聚类（k均值聚类）

R中可以进行快速聚类的函数为kmean()，其使用格式为kmean(x,centers,…)

其中，x为数据矩阵或数据框。centers为聚类的个数或初始聚类中心。

**例3：**本例模拟正态随机变量

首先用R模拟1000个均值为0，标准差为0.3的正态分布随机数，再把这些随机数转化为10个变量、100个对象的矩阵；其次，用同样的方法模拟1000个均值为1、标准差为0.3的正态分布数，再转化为10个变量、100个对象的矩阵；再次，把这两个矩阵合并成10个变量、200个样本的数据矩阵；最后用k均值法将其聚类成两类，观察其聚类效果如何。

命令：

> x1=matrix(rnorm(1000,mean=0,sd=0.3),ncol=10)

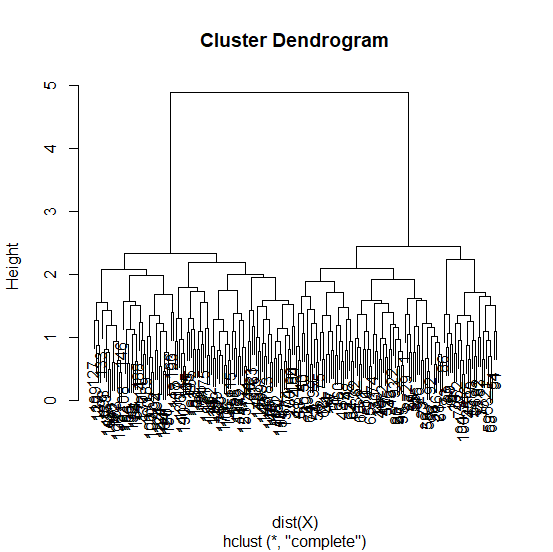
#均值为0，标准差为0.3的100\*10的正态随机矩阵

> x2=matrix(rnorm(1000,mean=1,sd=0.3),ncol=10)

#均值为1，标准差为0.3的100\*10的正态随机矩阵

> X=rbind(x1,x2) #按行合并为200\*10的矩阵

> plot(hclust(dist(X),"complete")) #可先用系统聚类法看聚类结果



> km<-kmeans(X,2) #用k均值法分类，存为cl，X为数据，2为分类个数

> km

K-means clustering with 2 clusters of sizes 100, 100

Cluster means: #每一类的均值

[,1] [,2] [,3] [,4] [,5] [,6]

1 0.98630275 0.962028415 0.99065213 0.95290059 1.01517499 1.00117240

2 -0.02757048 -0.009837257 0.02455744 0.01551709 -0.03148527 0.05606789

[,7] [,8] [,9] [,10]

1 1.03848761 0.99463023 1.023315871 1.003846408

2 0.04220529 0.05163364 0.003213441 0.003675369

Clustering vector: #分类结果

[1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

[37] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

[73] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1

[109] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

[145] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

[181] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Within cluster sum of squares by cluster:

[1] 88.06058 88.91206

(between\_SS / total\_SS = 73.3 %)#组间差距/总差距

Available components:

[1] "cluster" "centers" "totss" "withinss"

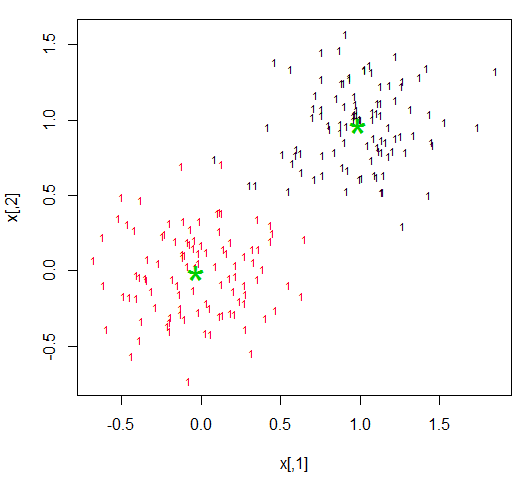
[5] "tot.withinss" "betweenss" "size"

> pch1=rep("1",100) #生成100个“1”

> pch2=rep("2",100) #生成100个“2”

> plot(X,col=km$clust,pch=c(pch1,pch2),cex=0.7) #画散点图，X表示数据，col表示颜色，col= cl$clust表示按分类的类别个数确定颜色，pch为点的形状，pch=c(pch1,pch2)表示按照上两句生成的pch1和pch2做点的形状，cex为文字大小；

> points(km$centers,col=3,pch="\*",cex=3) #在上述散点图中画中心点cl$centers，col=3为绿色，pch为中心点的形状，cex是中心点的大小；



从聚类结果来看，k均值法可以准确地把均值为0和均值为1的两类数据聚类开。图中“\*”分别是两类的聚类中心。

为了显示k均值法对大样本数据的优势，我们再模拟10000个均值为0，标准差为0.3的正态分布随机数，再把这些随机数转化为10个变量、1000个对象的矩阵；然后再用同样的方法模拟10000个均值为1、标准差为0.3的正态分布数，再转化为10个变量、1000个对象的矩阵；然后把这两个矩阵合并成10个变量、2000个样本的数据矩阵；最后用k均值法将其聚类成两类，观察其聚类效果如何。

命令：#解读同上

> x1=matrix(rnorm(10000,mean=0,sd=0.3),ncol=10)

> x2=matrix(rnorm(10000,mean=1,sd=0.3),ncol=10)

> x=rbind(x1,x2)

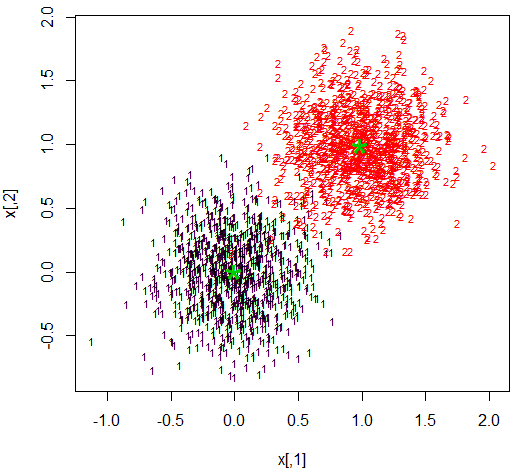
> cl<-kmeans(x,2)

> pch1=rep("1",1000)

> pch2=rep("2",1000)

> plot(x,col=cl$clust,pch=c(pch1,pch2),cex=0.7)

> points(cl$centers,col=3,pch="\*",cex=3)



从聚类结果来看，k均值法可以准确地把均值为0和均值为1的两类数据聚类开。图中“\*”分别是两类的聚类中心。**这里请不要使用系统聚类法，因为有可能电脑会死机。**

例：我们用例1对股票分类的例子用k均值法做分类。

> X=read.table("clipboard",header=T) #读取数据X

> Z=scale(X) #对数据X做标准化存入Z中

> (km<-kmeans(Z,5)) #对数据Z做K均值聚类，分5类

K-means clustering with 5 clusters of sizes 1, 2, 5, 5, 1#分类后每一类的个数

Cluster means: 1#分类后每一类的均值

x1 x2 x3 x4 x5 x6

1 1.19027530 1.19903390 -0.9307269 0.9356185 0.44655695 -1.99522518

2 0.07957372 0.05375786 2.1470906 -1.7021672 -0.84554988 -0.08103096

3 0.48522125 0.49550719 -0.2717295 0.5773688 -0.21933935 0.29367802

4 -1.05244268 -1.04636160 -0.1818662 -0.2158388 0.05636508 -0.27495132

5 1.48668442 1.44772242 -1.0954762 0.6610658 2.05941414 2.06365361

Clustering vector: #分类后每一个样品所属的类

冀东水泥 大同水泥 四川双马 牡丹江 西水股份 狮头股份 太行股份 海螺水泥

3 2 4 4 4 2 3 5

尖峰集团 四川金顶 祁连山 华新水泥 福建水泥 天鹅股份

4 3 3 3 1 4

Within cluster sum of squares by cluster:

[1] 0.0000000 0.6615705 10.0689851 10.6564465 0.0000000

(between\_SS / total\_SS = 72.6 %)

Available components:

[1] "cluster" "centers" "totss" "withinss" "tot.withinss" "betweenss"

[7] "size"

>plot(km$cluster) #对分类作图展示

> identify(km$cluster,labels=names(km$cluster),n=length(km$cluster),tolerance =0.25) #点击显示点的标签

