实验五：聚类分析

沈雨萱 3180104691

目录

1 实验概况

一．实验目的与要求：通过本试验项目，使学生理解并掌握如下内容 (1) 处理聚类分析的基本步骤；(2) 熟悉各类聚类方法；

二．实验内容本实验采用 "建筑数据"。这是一组 48 幢建筑的资料，有建筑面积，已经使用年份，结构，屋顶形式，电梯情况，空调个数，居住户数，07 年和 08 年用电量.

2 实验结果

一、数据来源和数据预测处理对数据进行正态性分析、相关性分析等

首先进行缺失值分析，由于有较多缺失值集中于 Aircondition 与 Families，因此用平均值进行填补.

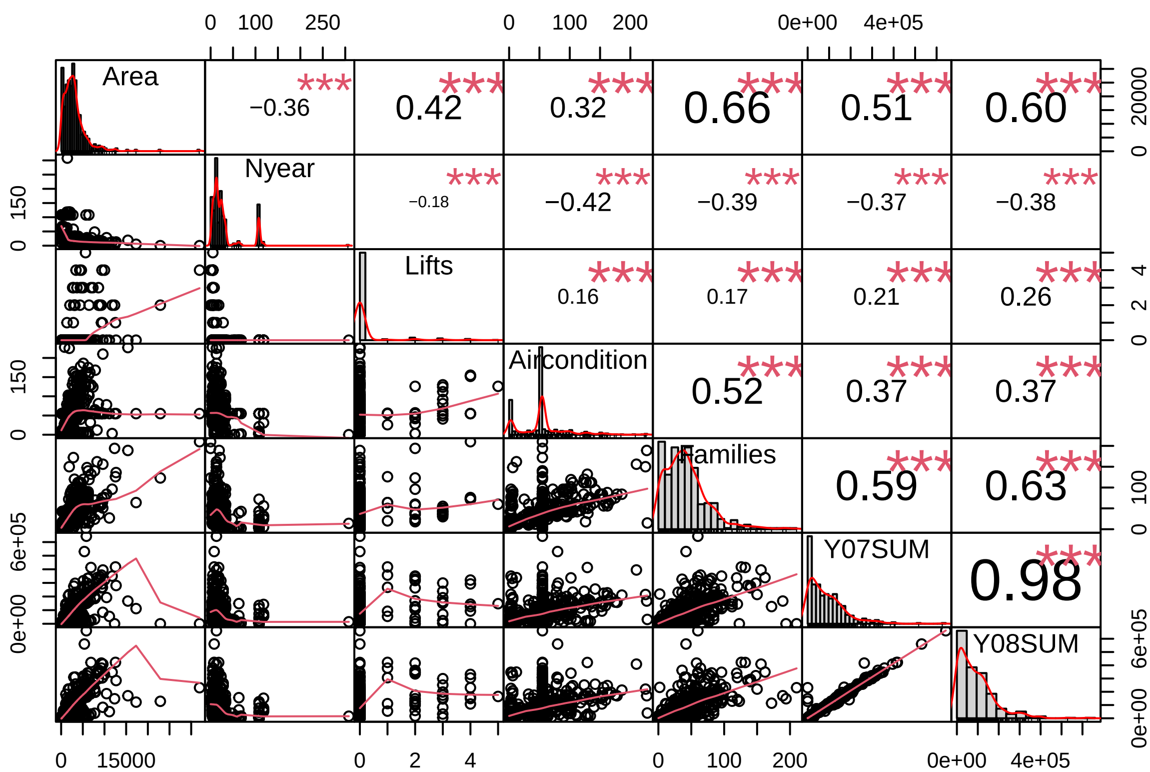
```
data_0 <- read.csv("data.csv",encoding = "UTF-8",na.strings=c(""," ","NA"),header=T)
sum(is.na(data_0))
```

```
## [1] 210
```

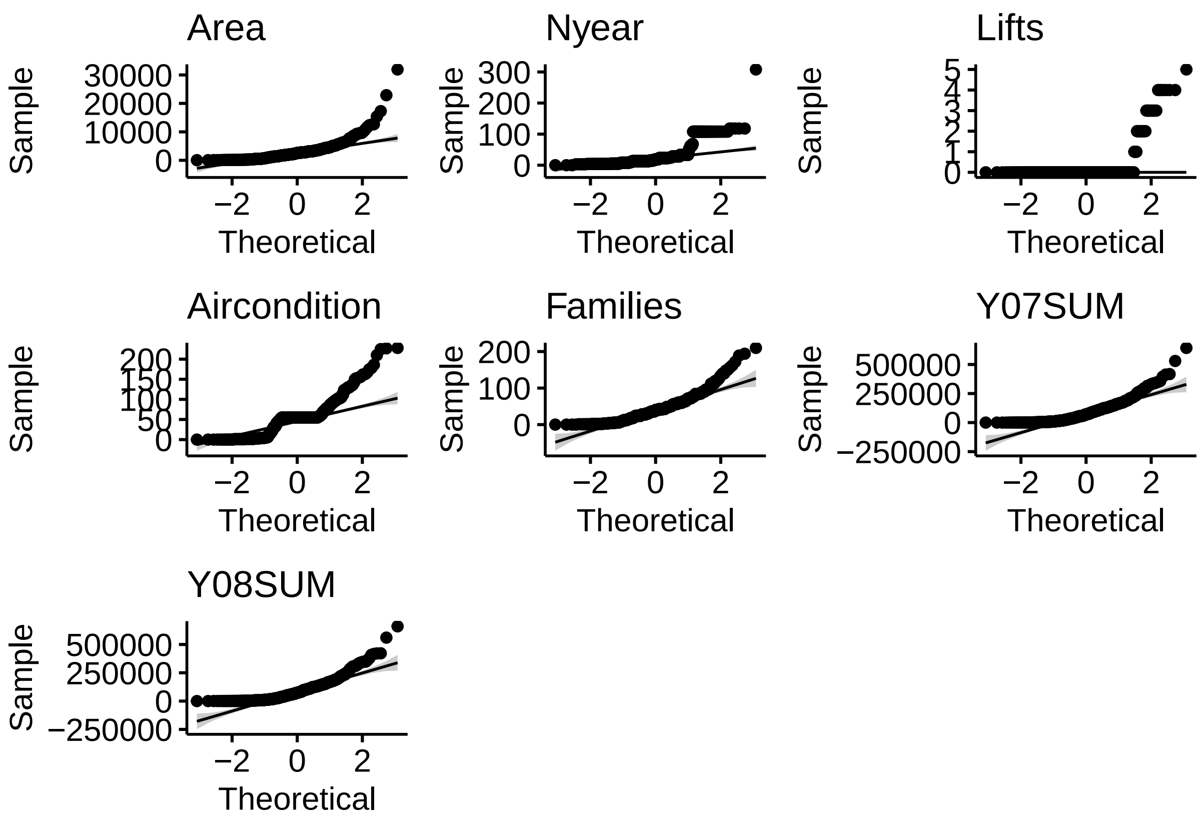
```
data <- data_0
data$Aircondition <- impute(data_0$Aircondition,mean)
data$Families <- impute(data_0$Families,mean)
```

进行正态性和相关性分析

```
X=data[,c('Area','Nyear','Lifts','Aircondition','Families','Y07SUM','Y08SUM')]
r <- rcorr(as.matrix(X))
chart.Correlation(X, histogram=TRUE, pch=19)
```




```
c1 <- ggqqplot(X$Area,main='Area')
c2 <- ggqqplot(X$Nyear,main='Nyear')
c3 <- ggqqplot(X$Lifts,main='Lifts')
c4 <- ggqqplot(X$Aircondition,main='Aircondition')
c5 <- ggqqplot(X$Families,main='Families')
c6 <- ggqqplot(X$Y07SUM,main='Y07SUM')
c7 <- ggqqplot(X$Y08SUM,main='Y08SUM')
c1+c2+c3+c4+c5+c6+c7
```



较

符合正态性的量有 Area, Families, Y07SUM, Y08SUM

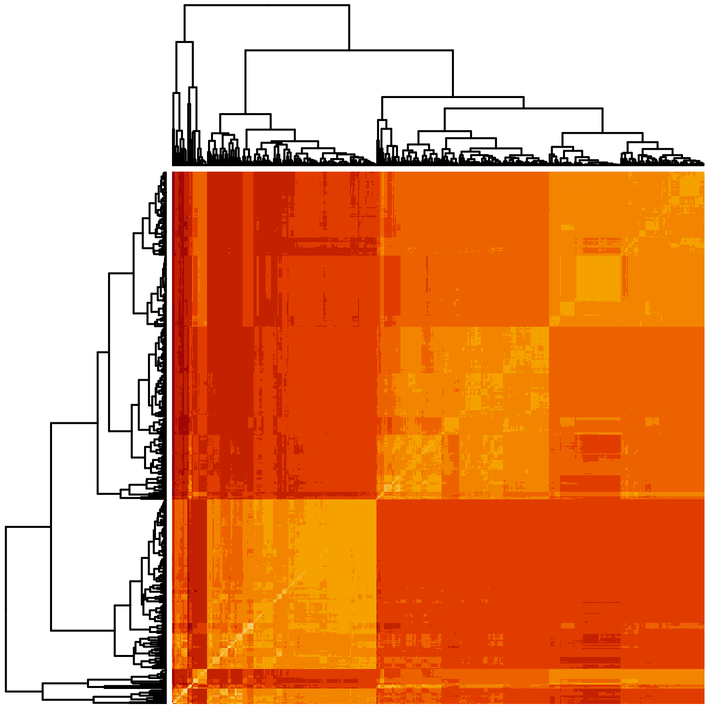
2.1 二、利用聚类方法对 482 幢建筑进行分类，并分析每类的特征

a. 数据中心化与标准化变换

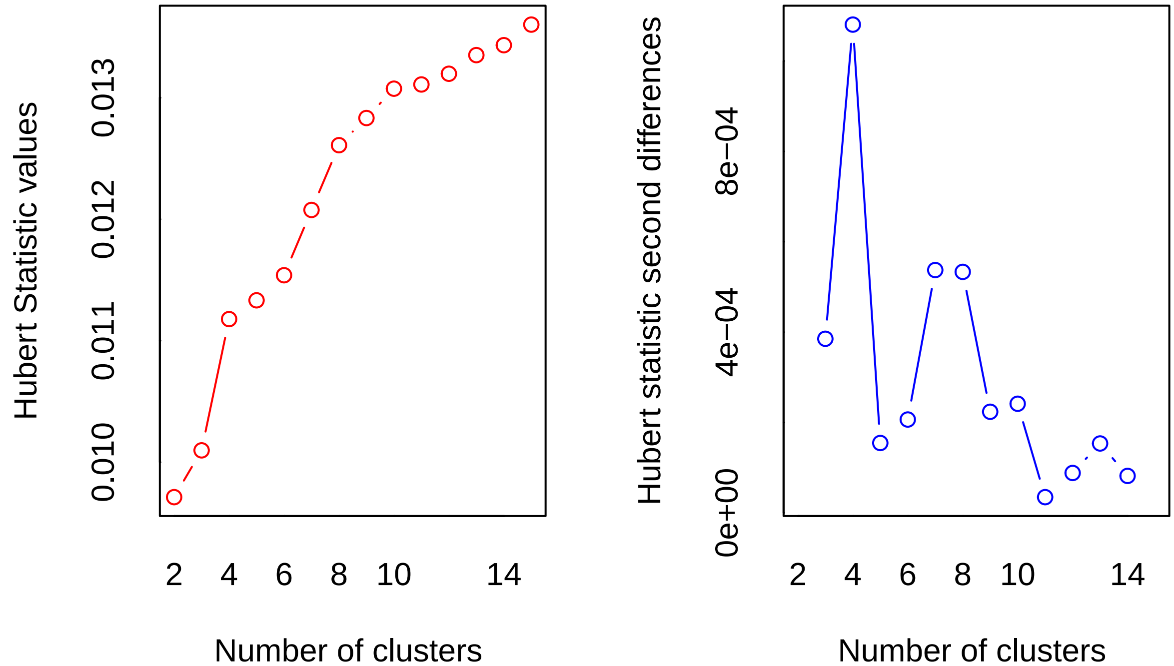
```
# 将结构和屋顶两项数值化
X$Constr=as.numeric(factor(data$Constr))
X$Form=as.numeric(factor(data$Form))
# 极差标准化
center<-sweep(X, 2, apply(X, 2, mean))# 按列中心化
R<-apply(X, 2, max)-apply(X, 2, min)# 计算列极差
X_star<-sweep(center, 2, R, "/")# 极差标准化，均值为 0，极差为 1
```

b. 系统聚类

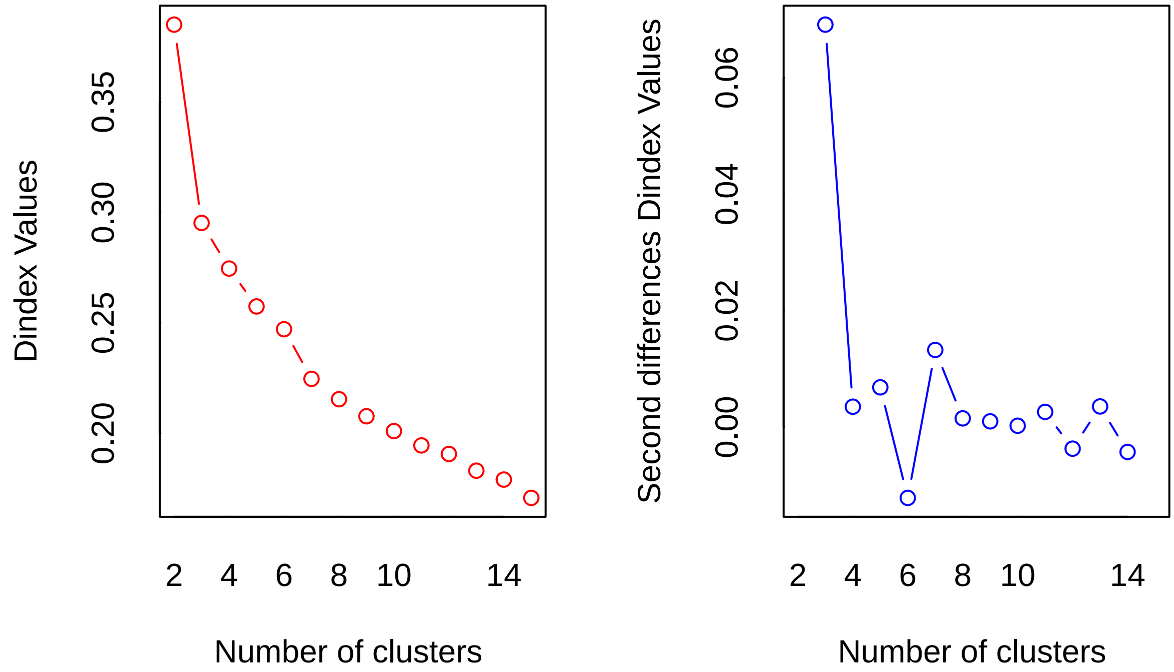
```
d<-dist(X_star,method = "euclidean")
heatmap(as.matrix(d),labRow = F, labCol = F)
```



```r
# 确定各类方法聚类个数，由于输出较多只展示最短距离法
cl_single <- NbClust(X_star, distance="euclidean",
              min.nc=2, max.nc=15, method="ward.D2")
```




```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hu
##                 index second differences plot.
##
```



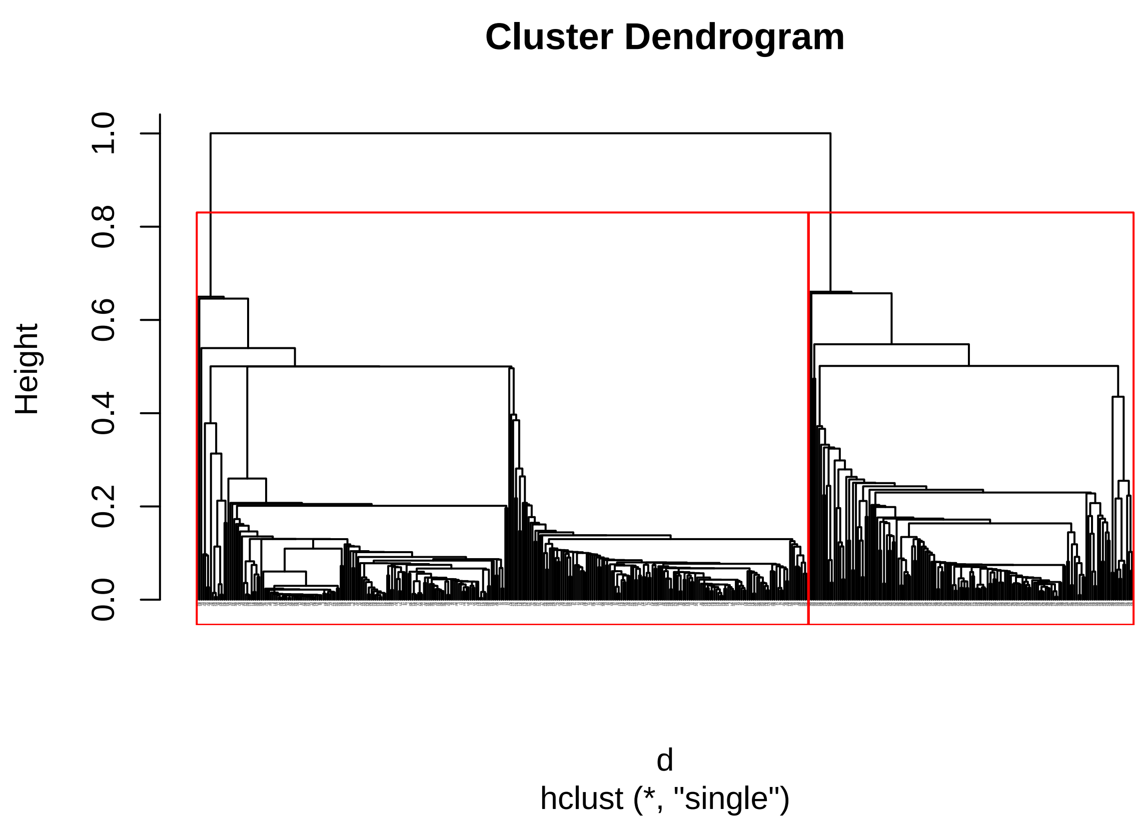

```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dinc
##                 second differences plot) that corresponds to a significant increase of the valu
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 8 proposed 2 as the best number of clusters
## * 6 proposed 3 as the best number of clusters
## * 1 proposed 4 as the best number of clusters
## * 3 proposed 7 as the best number of clusters
## * 2 proposed 11 as the best number of clusters
## * 3 proposed 15 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
```

```
##
## ********************************************************************
```

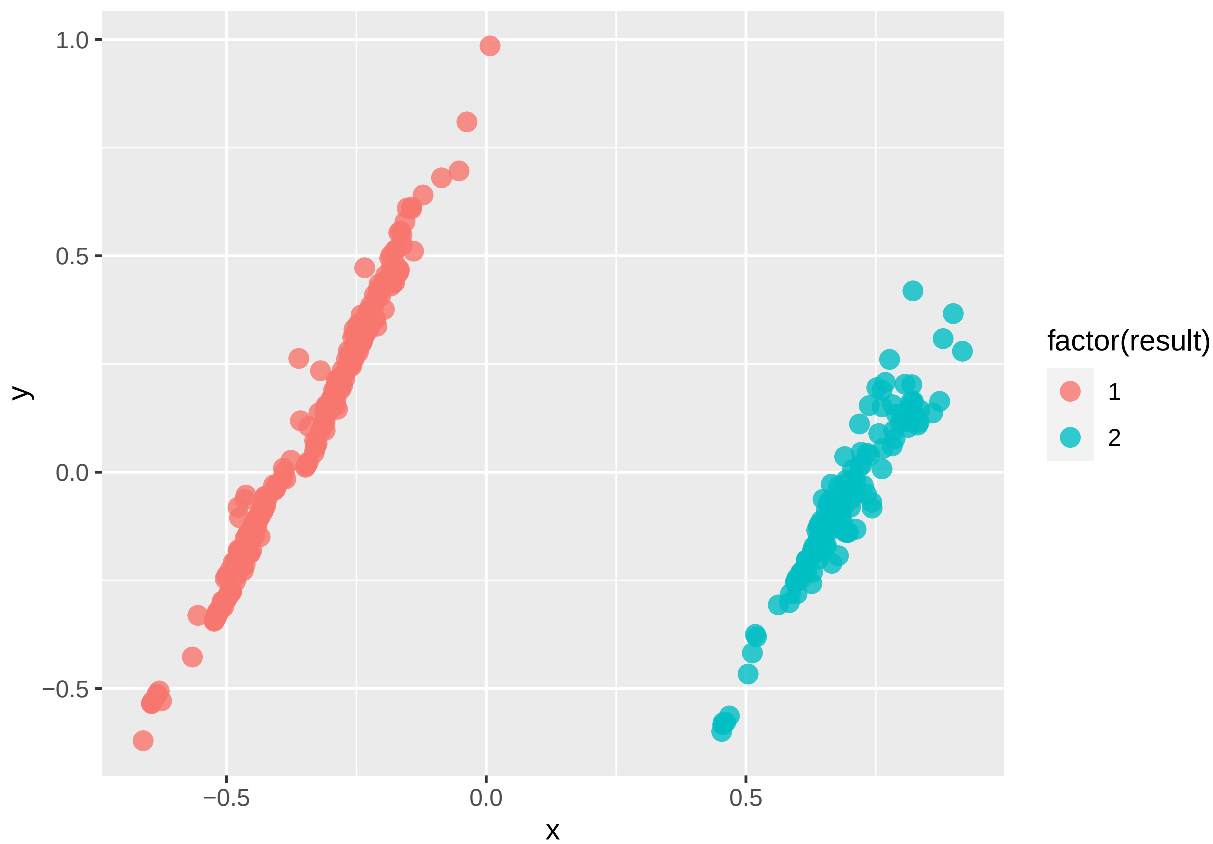
从热图来看，大致可以分成 2-4 类。由 NbCluster 分析结果得，最短距离法最佳聚为 2 类，最长距离法最佳聚为 3 类，中间距离法最佳聚为 2 类，类平均法最佳聚为 2 类，离差重心法最佳为 2 类

b.1 最短距离法聚类

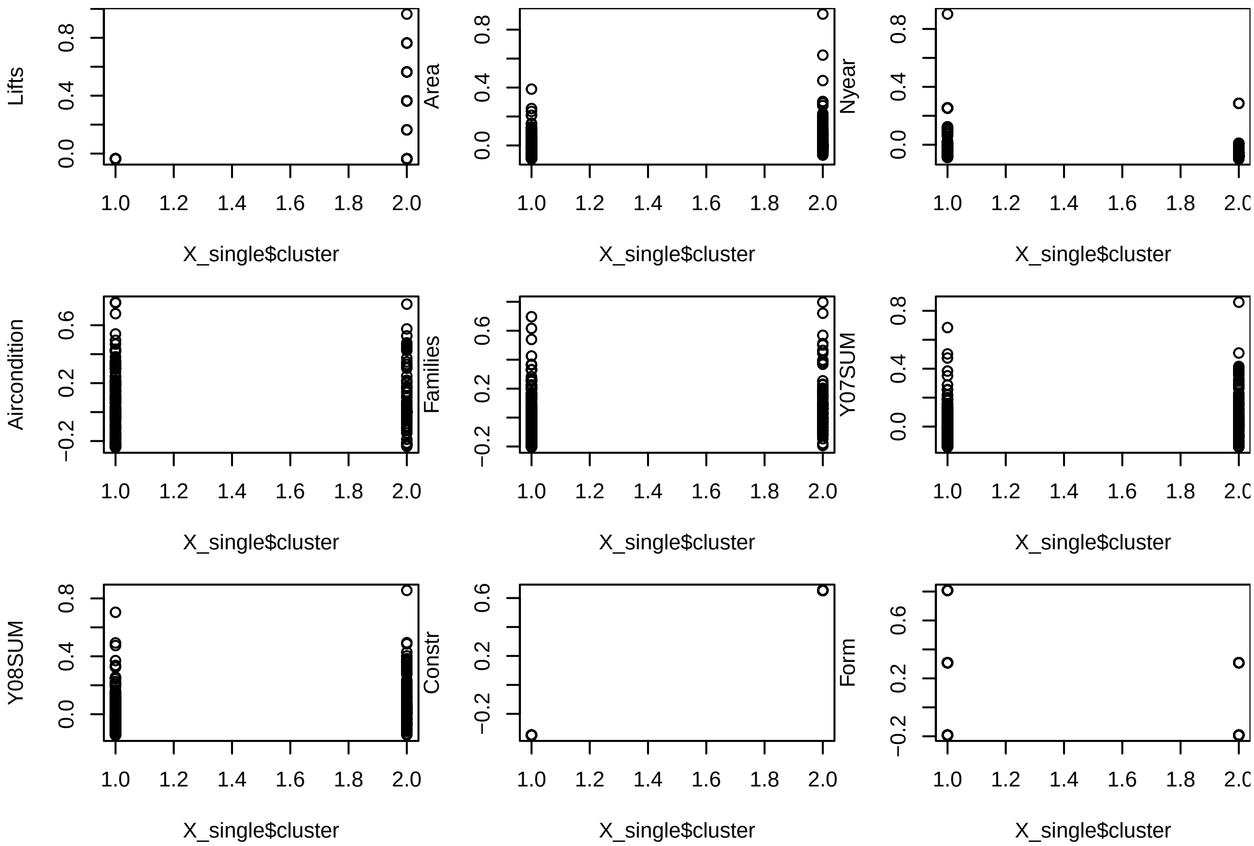
```
model1=hclust(d,method='single')
result=cutree(model1,k=2)
plot(model1,cex=0.1,hang=-1);re1<-rect.hclust(model1, k=2, border="red")
```




```
mds=cmdscale(d,k=2,eig=T)
x = mds$points[,1]
y = mds$points[,2]
p=ggplot(data.frame(x,y),aes(x,y))
p+geom_point(size=3,alpha=0.8,
             aes(colour=factor(result)))
```



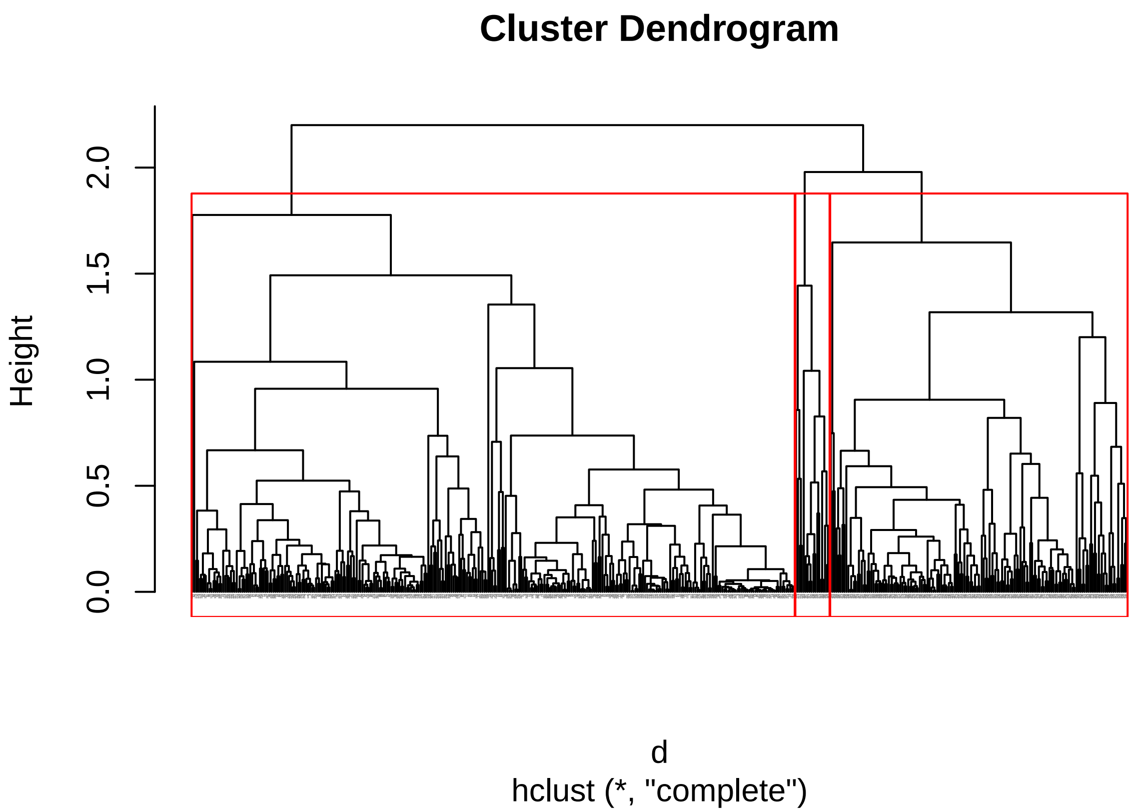

```
X_single <- X_star
X_single[,'cluster']=result
opar<-par(mfrow=c(3,3), mar=c(5.2,4,0,0))
plot(X_single$cluster,X_single$Lifts,ylab = "Lifts")
plot(X_single$cluster,X_single$Area,ylab="Area")
plot(X_single$cluster,X_single$Nyear,ylab="Nyear")
plot(X_single$cluster,X_single$Aircondition,ylab="Aircondition")
plot(X_single$cluster,X_single$Families,ylab="Families")
plot(X_single$cluster,X_single$Y07SUM,ylab="Y07SUM")
plot(X_single$cluster,X_single$Y08SUM,ylab="Y08SUM")
plot(X_single$cluster,X_single$Constr,ylab="Constr")
plot(X_single$cluster,X_single$Form,ylab="Form")
```

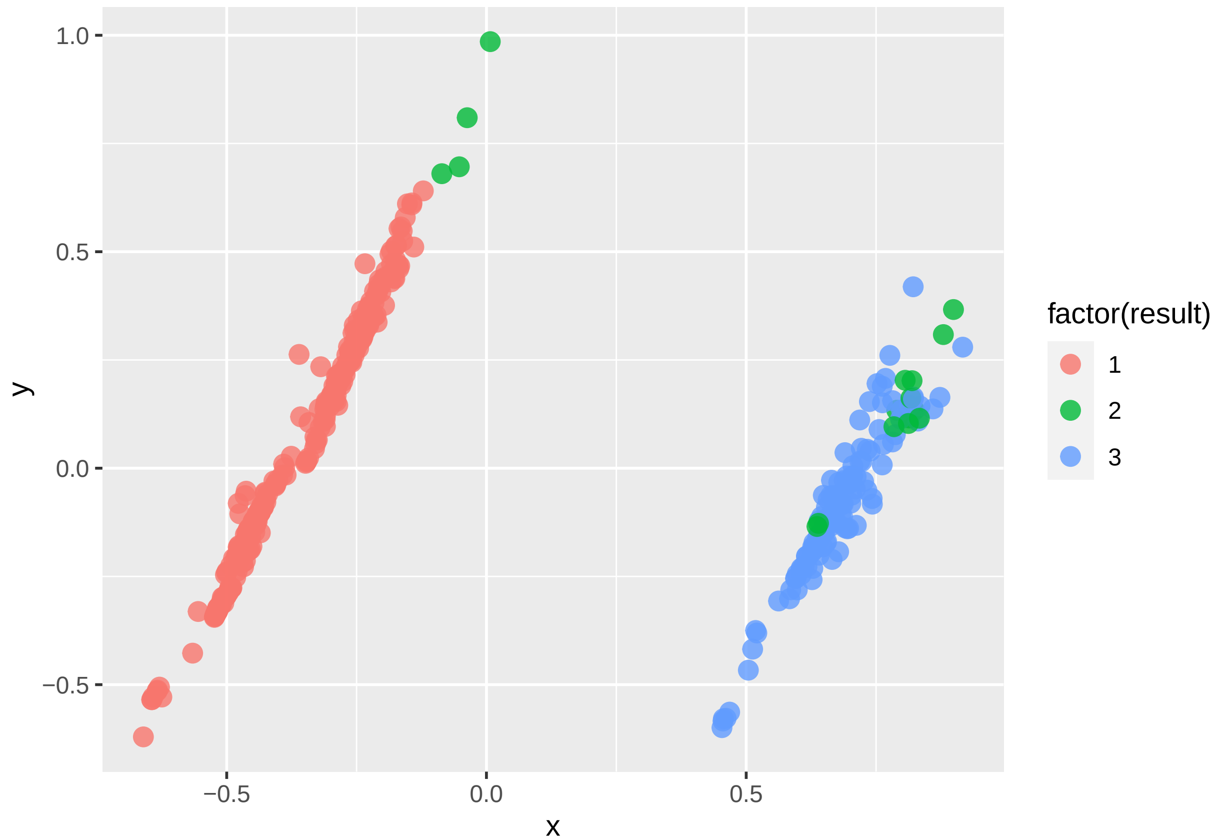


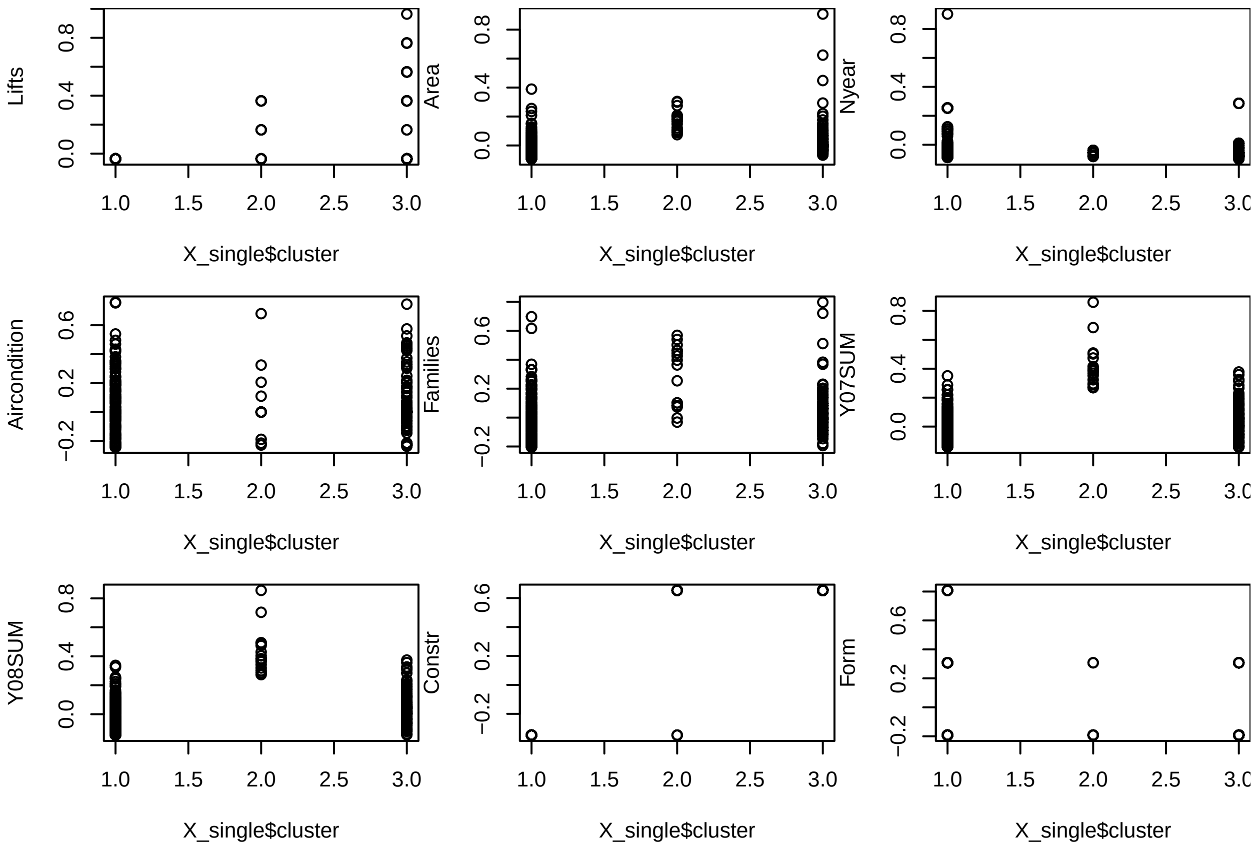

```
par(opar)
```

该方法将建筑分为两类，1 类有 316 个，2 类有 168 个。1 类的特点是电梯数较少，结构为混砖结构，面积相对较小；2 类的特点是电梯数量较分散，结构为框架结构。

b.2 最长距离法聚类

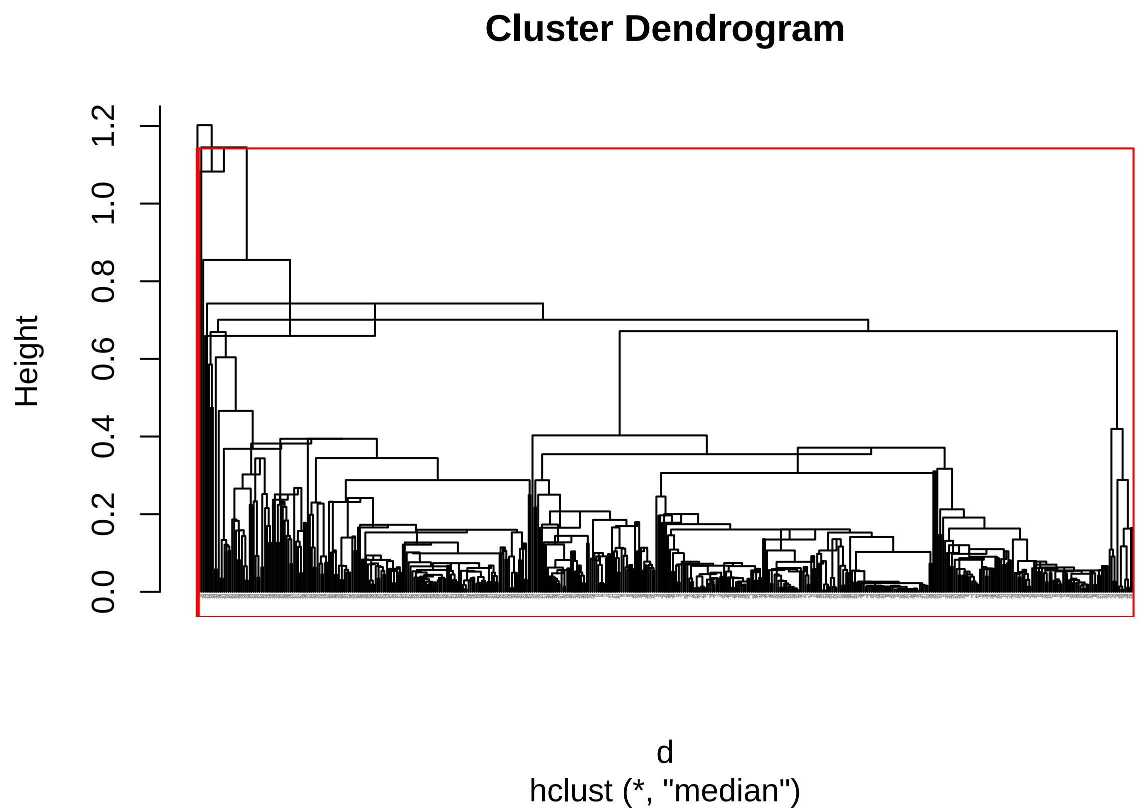

Cluster Dendrogram
Height
0.0
0.5
1.0
1.5
2.0
d
hclust (*, "complete")

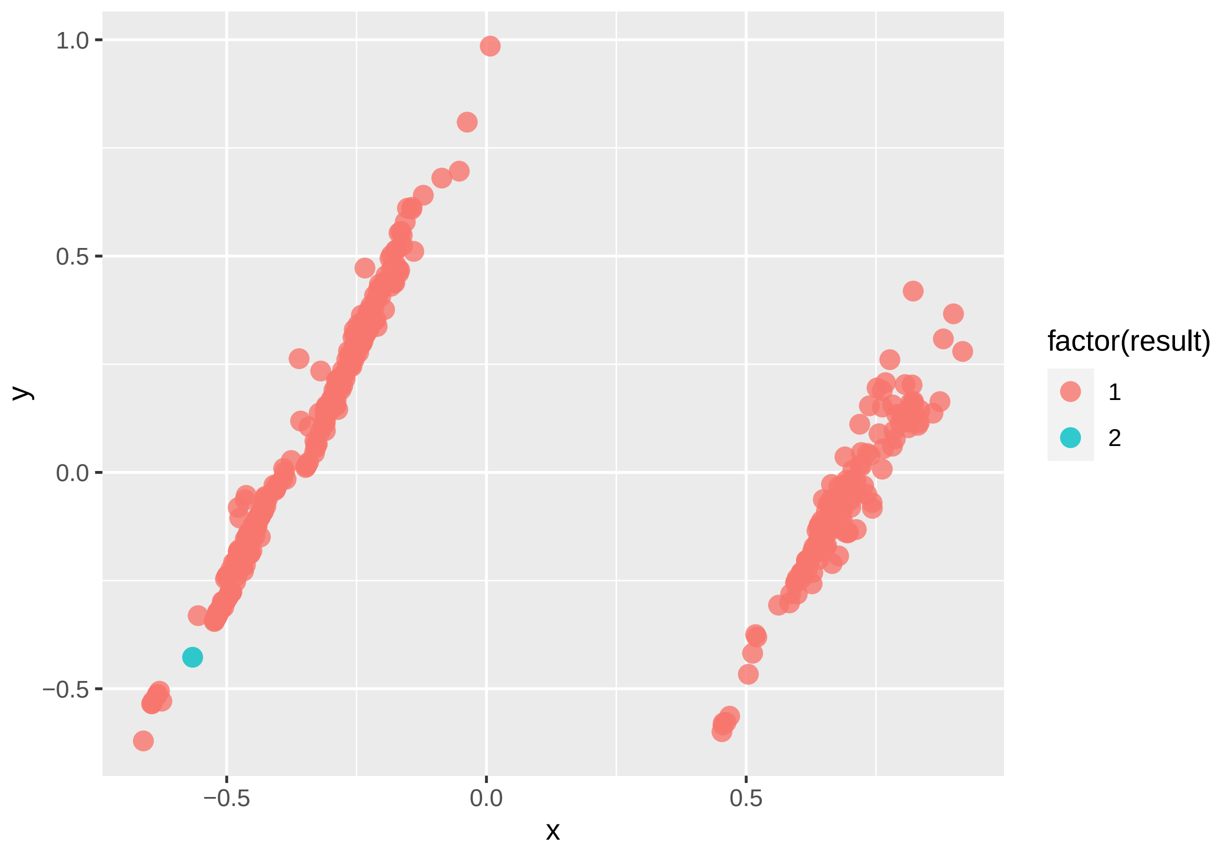

1.0
0.5
0.0
−0.5
y
−0.5
0.0
0.5
x
factor(result)
1
2
3

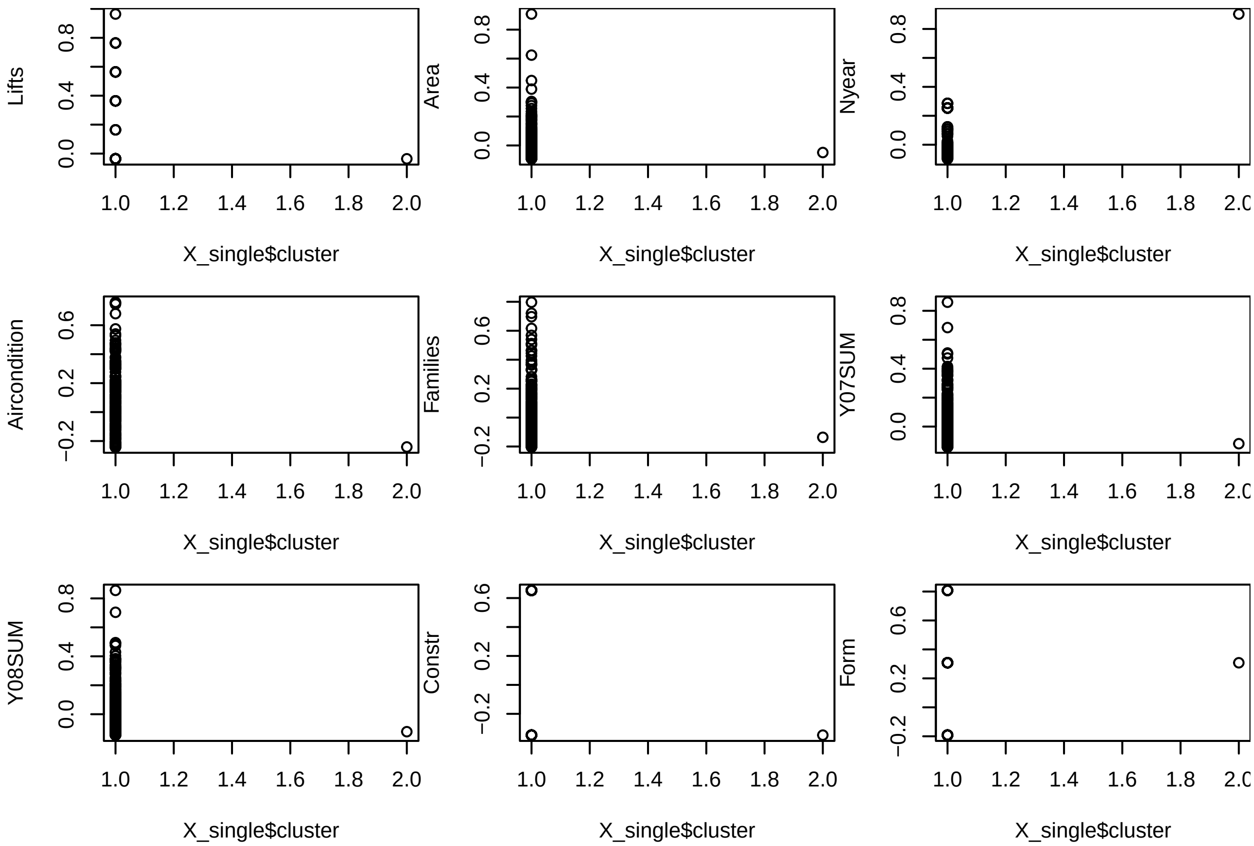


该方法将建筑分为 3 类。1 类有 312 个，电梯数量较少，年份相对较多，结构为混砖结构；2 类有 18 个，年份较少，07 年、08 年用电量较高；3 类电梯数量较分散，结构为框架结构。

b.3 中间距离法聚类

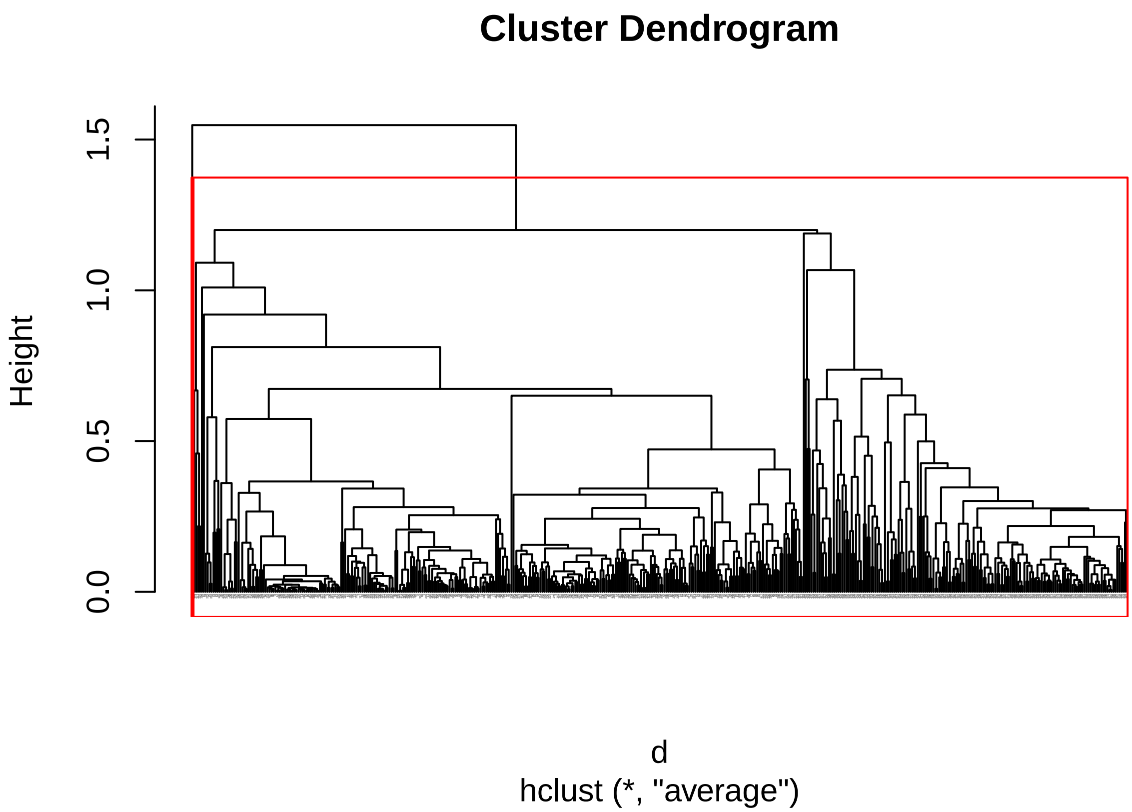

Cluster Dendrogram
Height
0.0
0.2
0.4
0.6
0.8
1.0
1.2
d
hclust (*, "median")

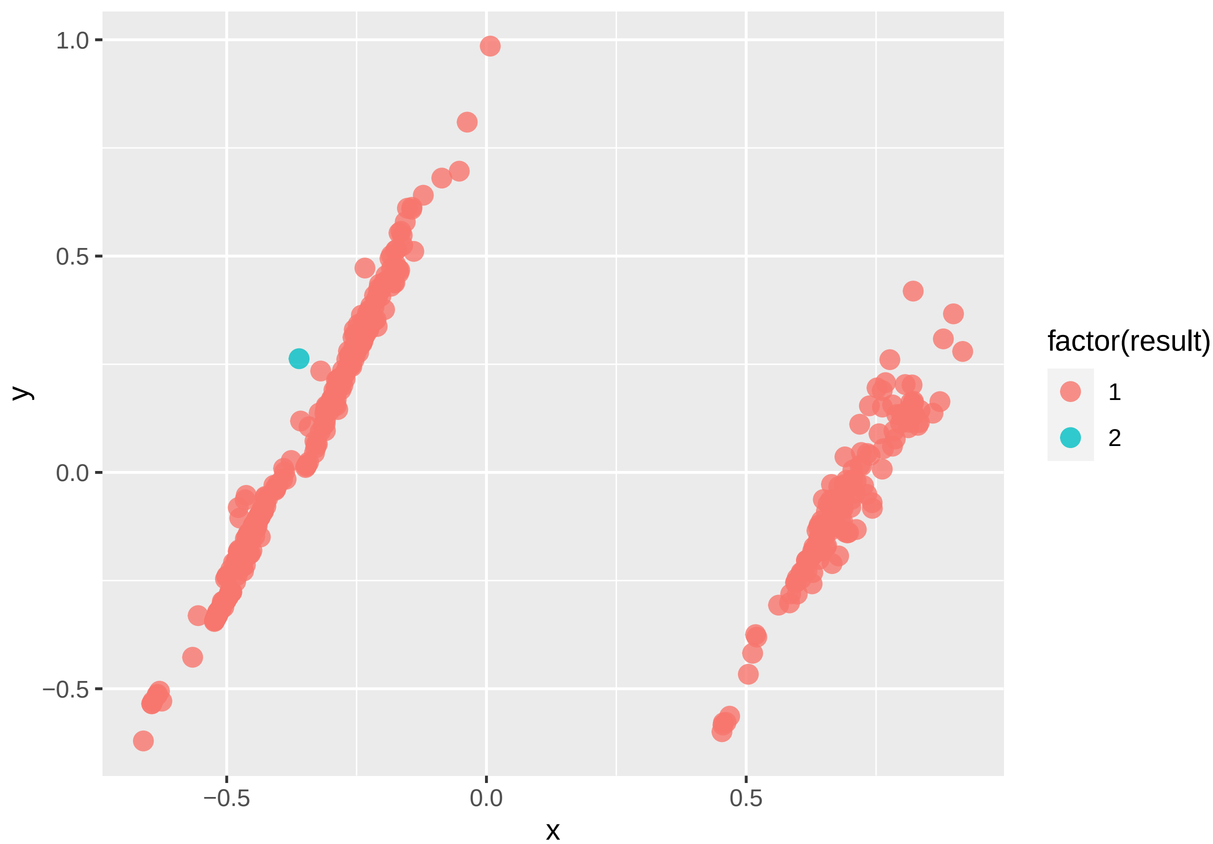

1.0
0.5
0.0
−0.5
y
−0.5
0.0
0.5
x
factor(result)
1
2

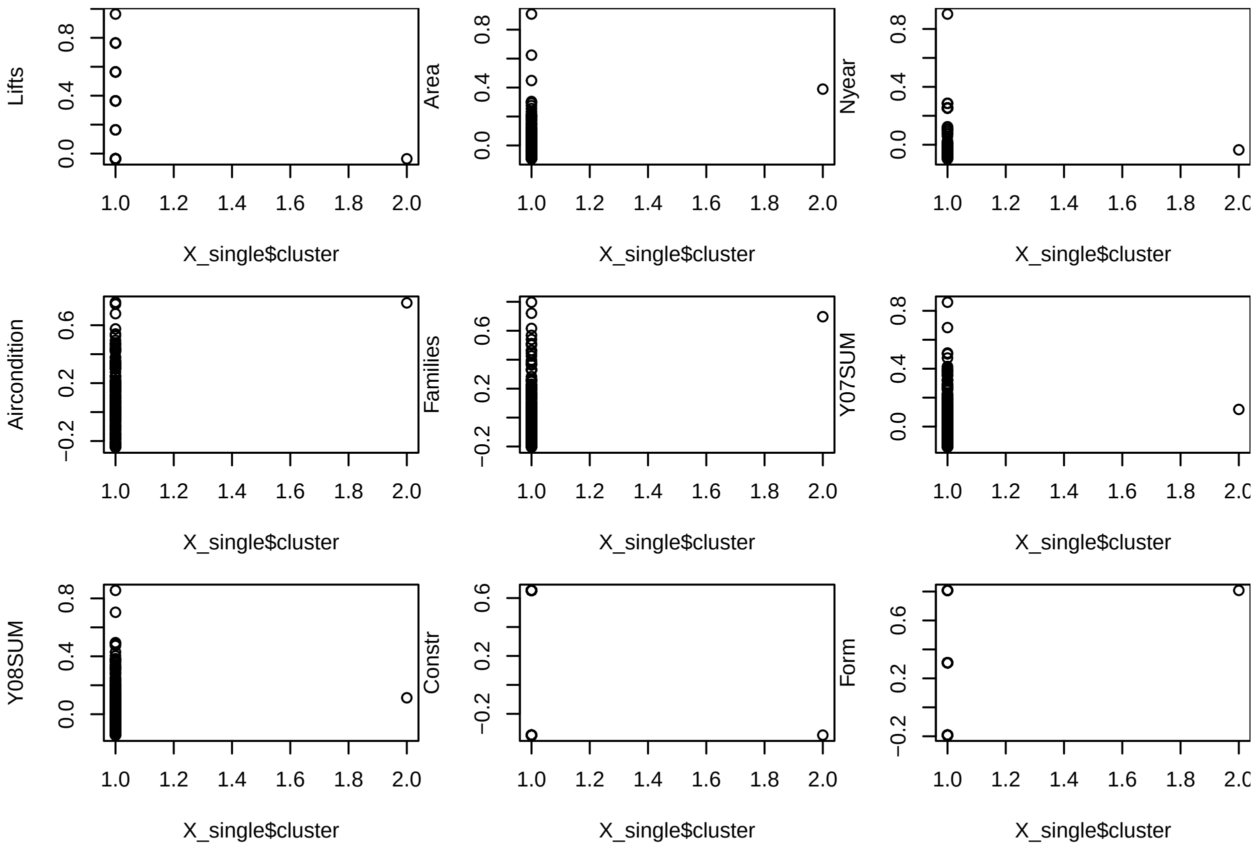


该方法将建筑分为两类，1 类只有一个。

b.4 类平均法

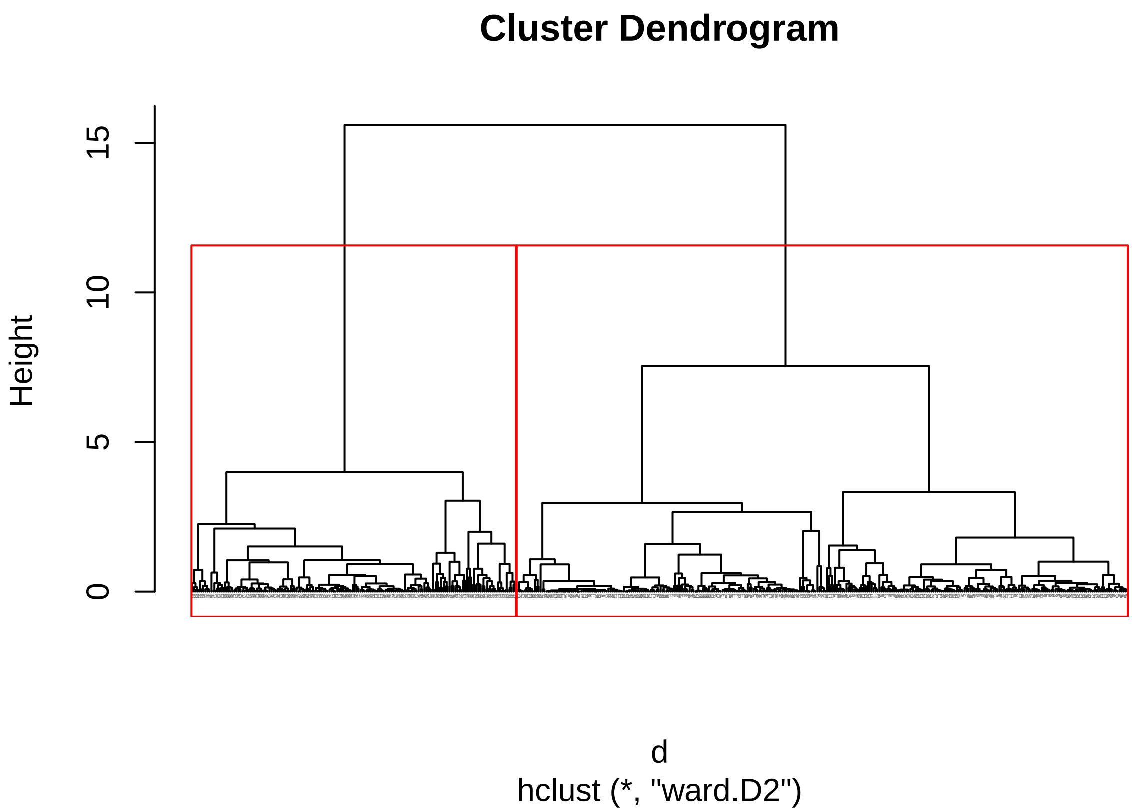

Cluster Dendrogram
Height
1.5
1.0
0.5
0.0
d
hclust (*, "average")

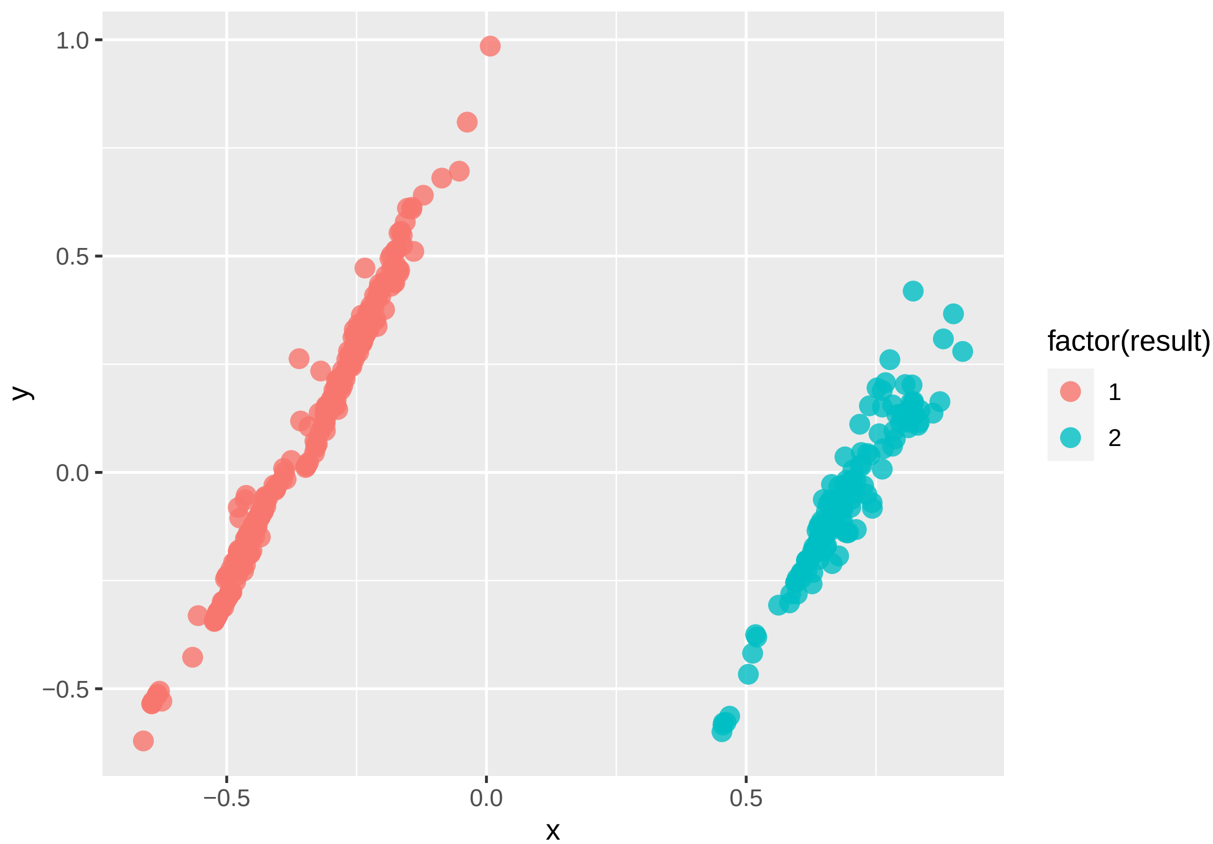

1.0
0.5
0.0
−0.5
y
−0.5
0.0
0.5
x
factor(result)
1
2

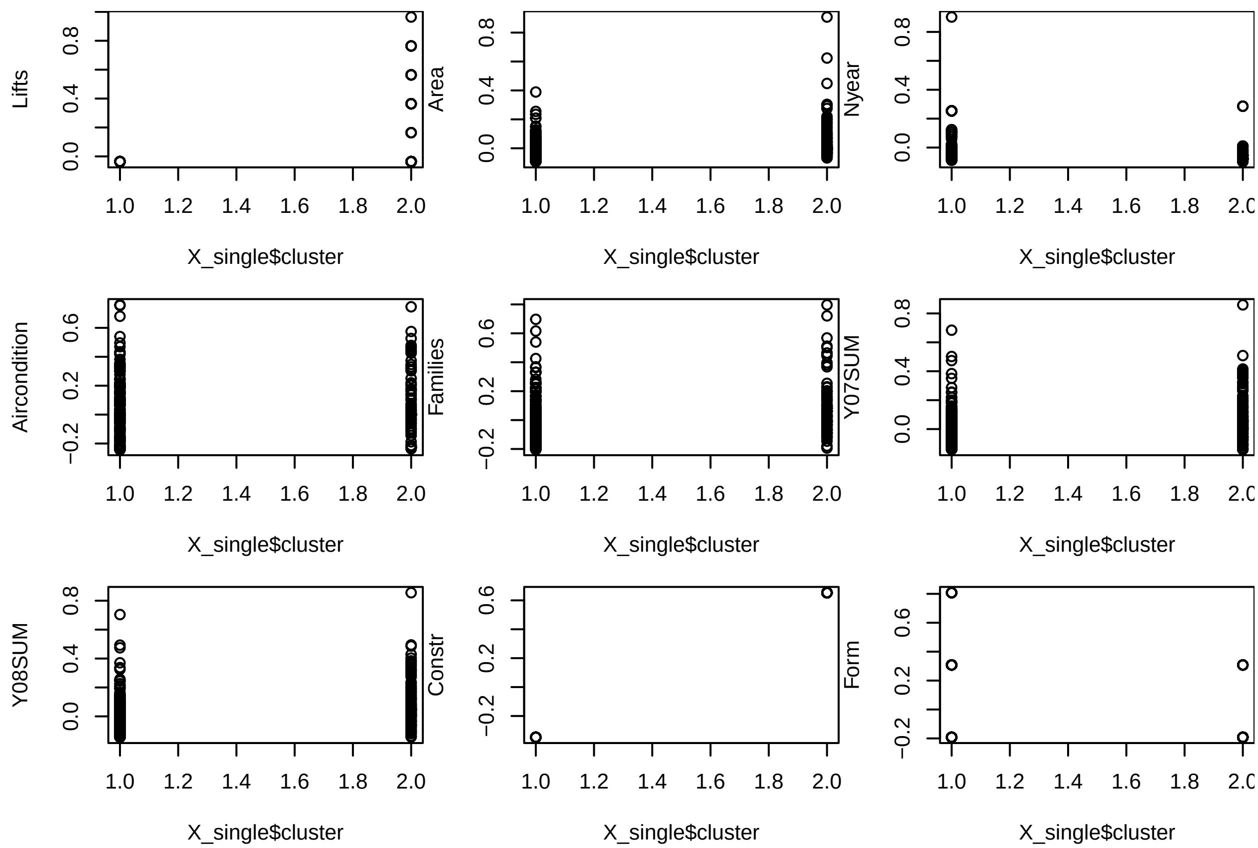


该方法将建筑分为两类，1 类只有一个。

b.5 离差重心法


Cluster Dendrogram
Height
15
10
5
0
d
hclust (*, "ward.D2")


1.0
0.5
0.0
−0.5
y
−0.5
0.0
0.5
x
factor(result)
1
2



该方法将建筑分为两类，1 类 168 个，电梯的数量分散，面积相对较大，结构为混砖结构；2 类有 316 个，电梯数量较少，面积相对较小，结构为框架结构。

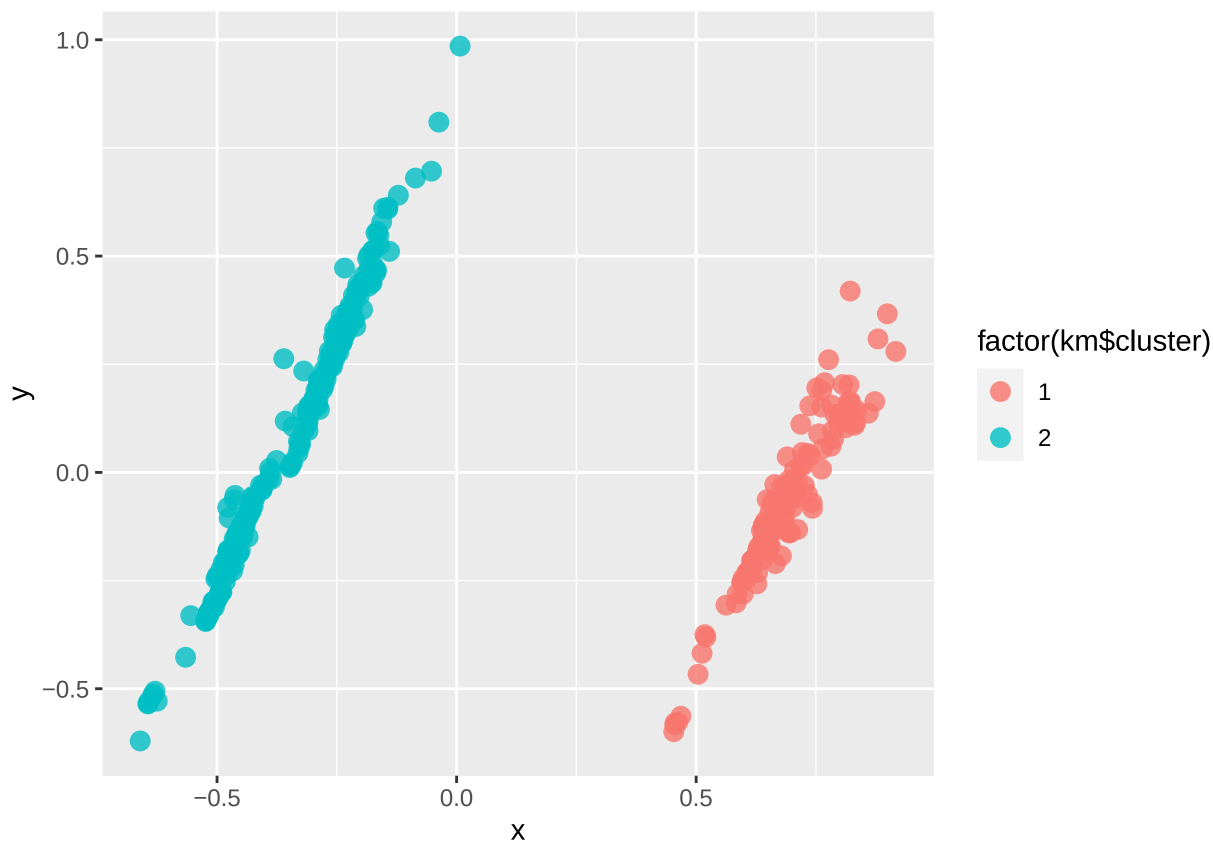
c. 动态聚类法

```
km <- kmeans(X_star, 2, algorithm="MacQueen")
km
```

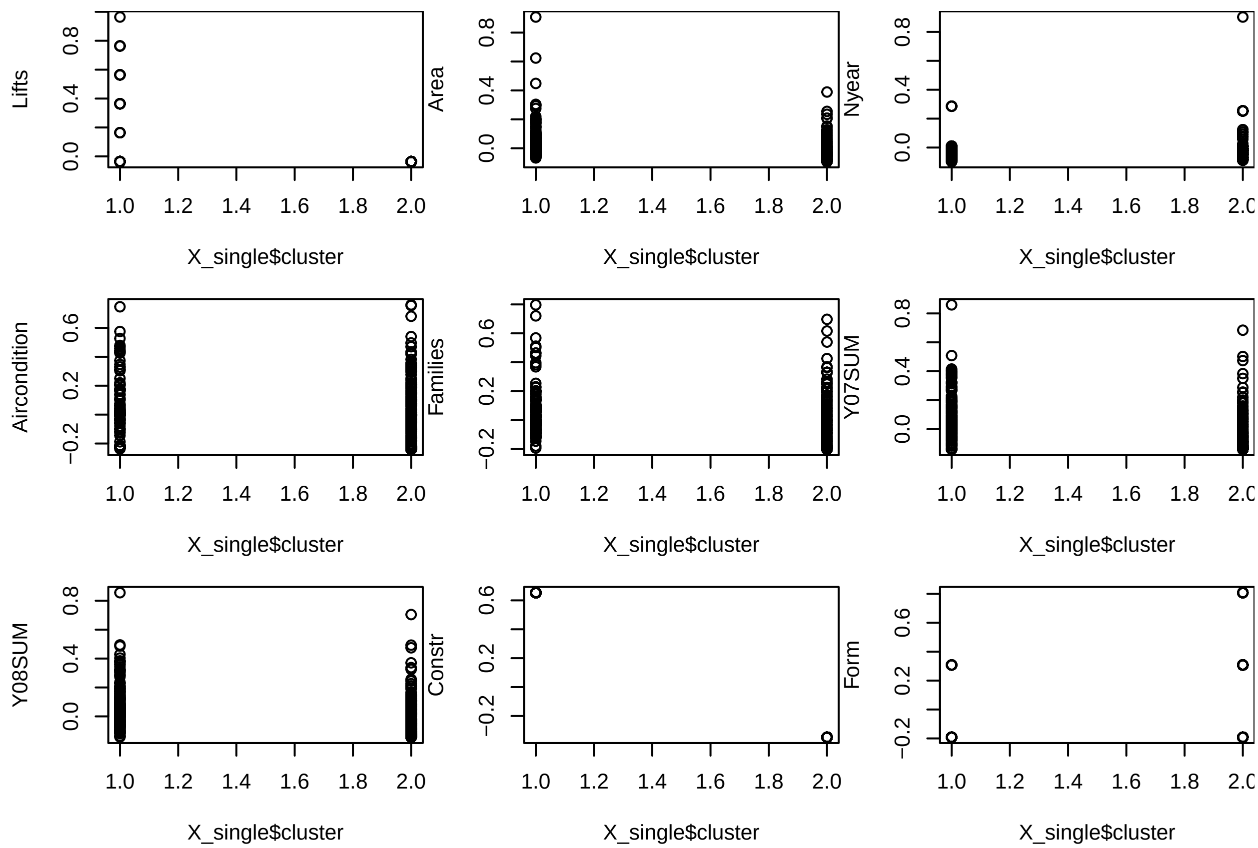
```
## K-means clustering with 2 clusters of sizes 168, 316
##
## Cluster means:
##          Area       Nyear       Lifts Aircondition    Families      Y07SUM
## 1  0.04297654 -0.04625425  0.06762102   0.02945000  0.05548131  0.06645469
## 2 -0.02284829  0.02459087 -0.03595041  -0.01565696 -0.02949639 -0.03533034
##        Y08SUM     Constr       Form
## 1  0.07221636  0.6528926 -0.1564345
## 2 -0.03839351 -0.3471074  0.0831677
##
## Clustering vector:
##   [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 2 2
```

```
##  [38] 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 2
##  [75] 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 2
## [112] 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 2
## [149] 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 2
## [186] 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 2
## [223] 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 2
## [260] 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 2
## [297] 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 1 1 1 1 1 1 1 1 1
## [334] 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 1
## [371] 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 1
## [408] 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 1
## [445] 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 1
## [482] 1 1 1
##
## Within cluster sum of squares by cluster:
## [1] 30.36277 57.73230
##  (between_SS / total_SS =  58.0 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
mds=cmdscale(d,k=2,eig=T)
x = mds$points[,1]
y = mds$points[,2]
p=ggplot(data.frame(x,y),aes(x,y))
p+geom_point(size=3,alpha=0.8,
             aes(colour=factor(km$cluster)))
```




```
X_single <- X_star
X_single[,'cluster']=km$cluster
opar<-par(mfrow=c(3,3), mar=c(5.2,4,0,0))
plot(X_single$cluster,X_single$Lifts,ylab = "Lifts")
plot(X_single$cluster,X_single$Area,ylab="Area")
plot(X_single$cluster,X_single$Nyear,ylab="Nyear")
plot(X_single$cluster,X_single$Aircondition,ylab="Aircondition")
plot(X_single$cluster,X_single$Families,ylab="Families")
plot(X_single$cluster,X_single$Y07SUM,ylab="Y07SUM")
plot(X_single$cluster,X_single$Y08SUM,ylab="Y08SUM")
plot(X_single$cluster,X_single$Constr,ylab="Constr")
plot(X_single$cluster,X_single$Form,ylab="Form")
```




```
par(opar)
```

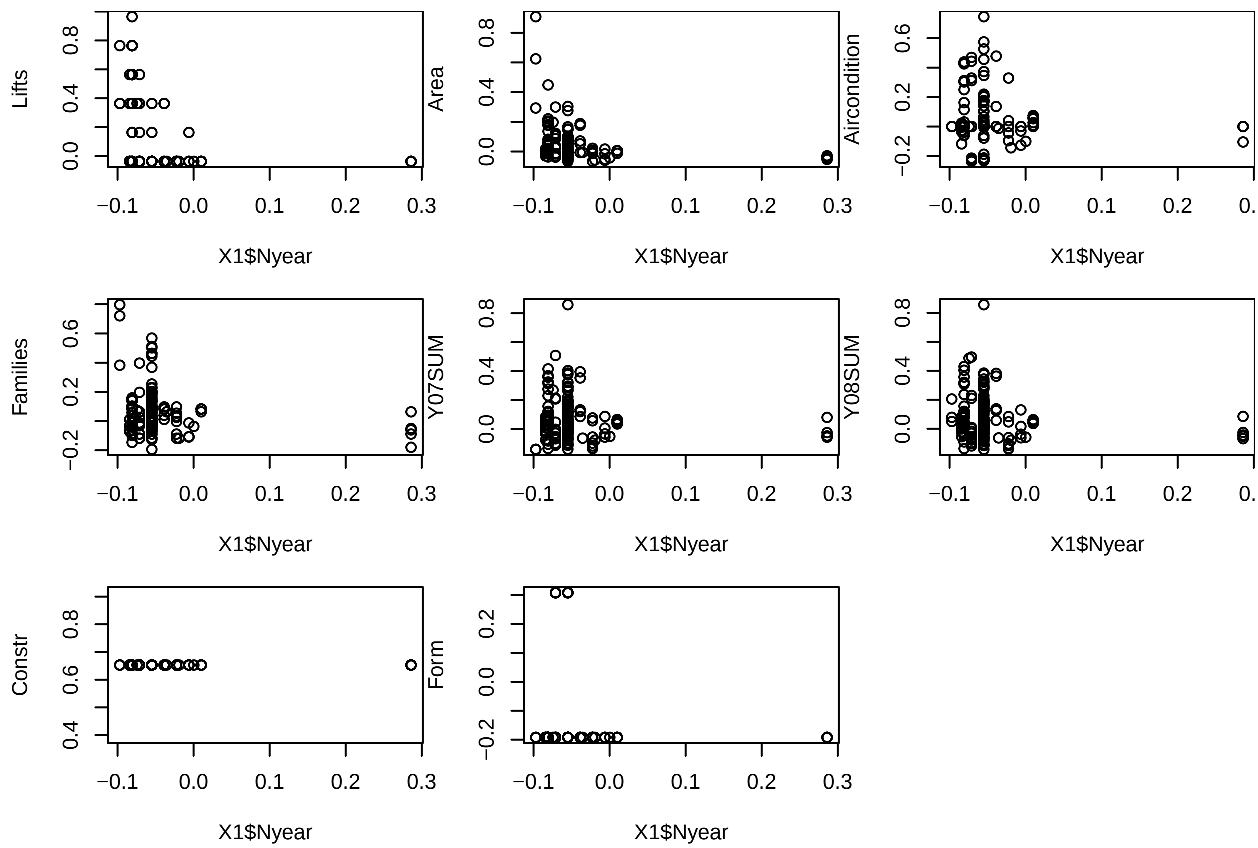
动态聚类法分两类时结果与离差重心法类似

2.2 三、比较类之间的差异，结合使用年份去分析各时期建筑的特点等。

按照上述动态聚类法及离心重力法的分类结果，第一类：

```
X_single <- X_star
X_single[,'cluster']=km$cluster
X1<-X_single %>% filter(cluster==1)
X2<-X_single %>% filter(cluster==2)
opar<-par(mfrow=c(3,3), mar=c(5.2,4,0,0))
plot(X1$Nyear,X1$Lifts,ylab = "Lifts")
plot(X1$Nyear,X1$Area,ylab="Area")
plot(X1$Nyear,X1$Aircondition,ylab="Aircondition")
plot(X1$Nyear,X1$Families,ylab="Families")
plot(X1$Nyear,X1$Y07SUM,ylab="Y07SUM")
```

```
plot(X1$Nyear,X1$Y08SUM,ylab="Y08SUM")
plot(X1$Nyear,X1$Constr,ylab="Constr")
plot(X1$Nyear,X1$Form,ylab="Form")
par(opar)
```

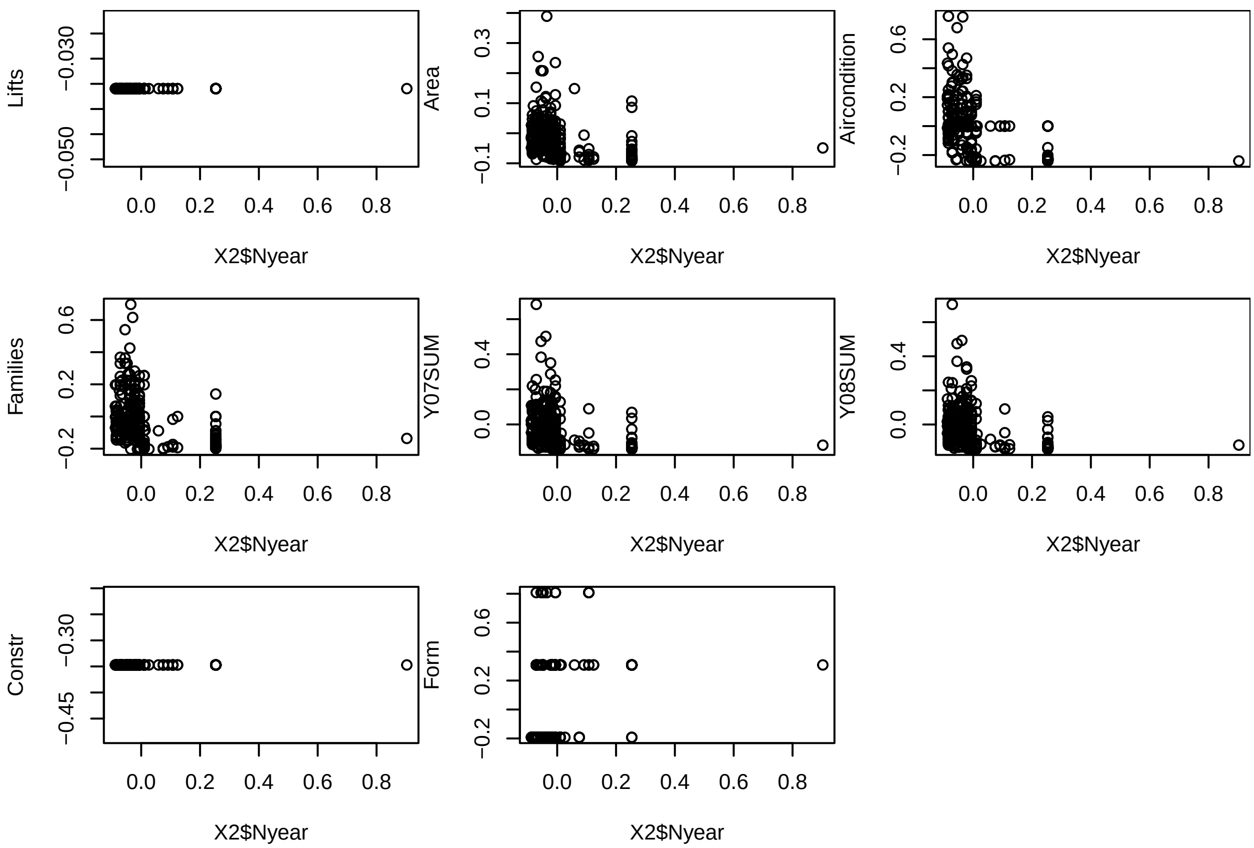


第一类中，随着使用年份增长，电梯数量减少，面积减小，空调数目先增加后减少，家庭数目先增加后减少，07、08 年用电量减少，结构均为框架结构

第二类：

```
X_single <- X_star
X_single[,'cluster']=km$cluster
X1<-X_single %>% filter(cluster==1)
X2<-X_single %>% filter(cluster==2)
opar<-par(mfrow=c(3,3), mar=c(5.2,4,0,0))
plot(X2$Nyear,X2$Lifts,ylab = "Lifts")
plot(X2$Nyear,X2$Area,ylab="Area")
plot(X2$Nyear,X2$Aircondition,ylab="Aircondition")
plot(X2$Nyear,X2$Families,ylab="Families")
plot(X2$Nyear,X2$Y07SUM,ylab="Y07SUM")
```

```
plot(X2$Nyear,X2$Y08SUM,ylab="Y08SUM")
plot(X2$Nyear,X2$Constr,ylab="Constr")
plot(X2$Nyear,X2$Form,ylab="Form")
par(opar)
```



第二类中，随着使用年份增长，电梯数量基本不变，面积先快速减小后上升，空调数目、07、08 年用电量减少和家庭数目以较快速度减少，结构均为混砖结构，屋顶三种类型都有。

2.3 四、按使用的年限进行有序分类，看看每个不同阶段建筑的特点。

```
ocluster = function(datasam, classnum) {
    # 有序样本聚类，输入 datasam 为样本数据阵，每一行为一个样本；
# 输入 classnum 为要分的类数
# 返回值 result1 为分类结果示意图
# 各类的起始点存在变量 breaks 中
# 输出三个矩阵 ra_dis: 距离矩阵 leastlost: 最小损失矩阵 classid: 分类标识矩阵
#author:banmudi 2010.11
```

```
# 样本数
    sam_n = dim(datasam)[1]
    # 子函数，计算 i-j 个样本组成的类的半径
    radi = function(i, j) {
        # 提取 i-j 个样本
        temp =as.matrix( datasam[i:j, ])
            mu = colMeans(matrix(temp,j-i+1))
            vec = apply(matrix(temp,j-i+1), 1, function(x) {
                x - mu
            })
            round(sum(apply(matrix(vec,j-i+1), 2, crossprod)),3)
    }



    # 计算距离矩阵
    ra_dis = matrix(0, sam_n, sam_n)
rownames(ra_dis) = 1:sam_n
    colnames(ra_dis) = 1:sam_n
    for (i in 1:(sam_n - 1)) {
        for (j in (i + 1):sam_n) {
            ra_dis[i, j] = radi(i, j)
            ra_dis[j, i] = radi(i, j)
        }
    }

    # 最小损失矩阵，行为样本数，列为分类数
#leastlost[i,j] 表示把 1:i 样本分成 j 类对应的最小损失
    leastlost = matrix(, sam_n - 1, sam_n - 1)
    rownames(leastlost) = 2:sam_n
    colnames(leastlost) = 2:sam_n
diag(leastlost) = 0
    #round(leastlost,3);

    # 记录下对应的分类结点
    classid = matrix(, sam_n - 1, sam_n - 1)
    rownames(classid) = 2:sam_n
    colnames(classid) = 2:sam_n
```

```
diag(classid) = 2:sam_n

# 分成两类时, 填写最小损失阵的第一列
leastlost[as.character(3:sam_n), "2"] = sapply(3:sam_n,
    function(xn) {
        min(ra_dis[1, 1:(xn - 1)] + ra_dis[2:xn, xn])
    })
classid[as.character(3:sam_n), "2"] = sapply(3:sam_n, function(xn) {
    which((ra_dis[1, 1:(xn - 1)] + ra_dis[2:xn, xn]) == (min(ra_dis[1,
        1:(xn - 1)] + ra_dis[2:xn, xn])))[1] + 1
})
# 分成 j 类时, 填写最小损失阵的 第二列到最后一列
for (j in as.character(3:(sam_n - 1))) {
    # 分成 j 类
    leastlost[as.character((as.integer(j) + 1):sam_n), j] = sapply((as.integer(j) +
        1):sam_n, function(xn) {
        min(leastlost[as.character(j:xn - 1), as.character(as.integer(j) -
            1)] + ra_dis[j:xn, xn])
    })

    classid[as.character((as.integer(j) + 1):sam_n), j] = sapply((as.integer(j) +
        1):sam_n, function(xn) {
        a = which((leastlost[as.character(j:xn - 1), as.character(as.integer(j) -
            1)] + ra_dis[j:xn, xn]) == min(leastlost[as.character(j:xn -
            1), as.character(as.integer(j) - 1)] + ra_dis[j:xn,
            xn]))[1] + as.integer(j) - 1
    })
}

diag(classid) = 2:sam_n

breaks = rep(0, 1, classnum)
breaks[1] = 1
breaks[classnum] = classid[as.character(sam_n), as.character(classnum)]
flag = classnum - 1
while (flag >= 2) {
    breaks[flag] = classid[as.character(breaks[flag + 1] -
        1), as.character(flag)]
```

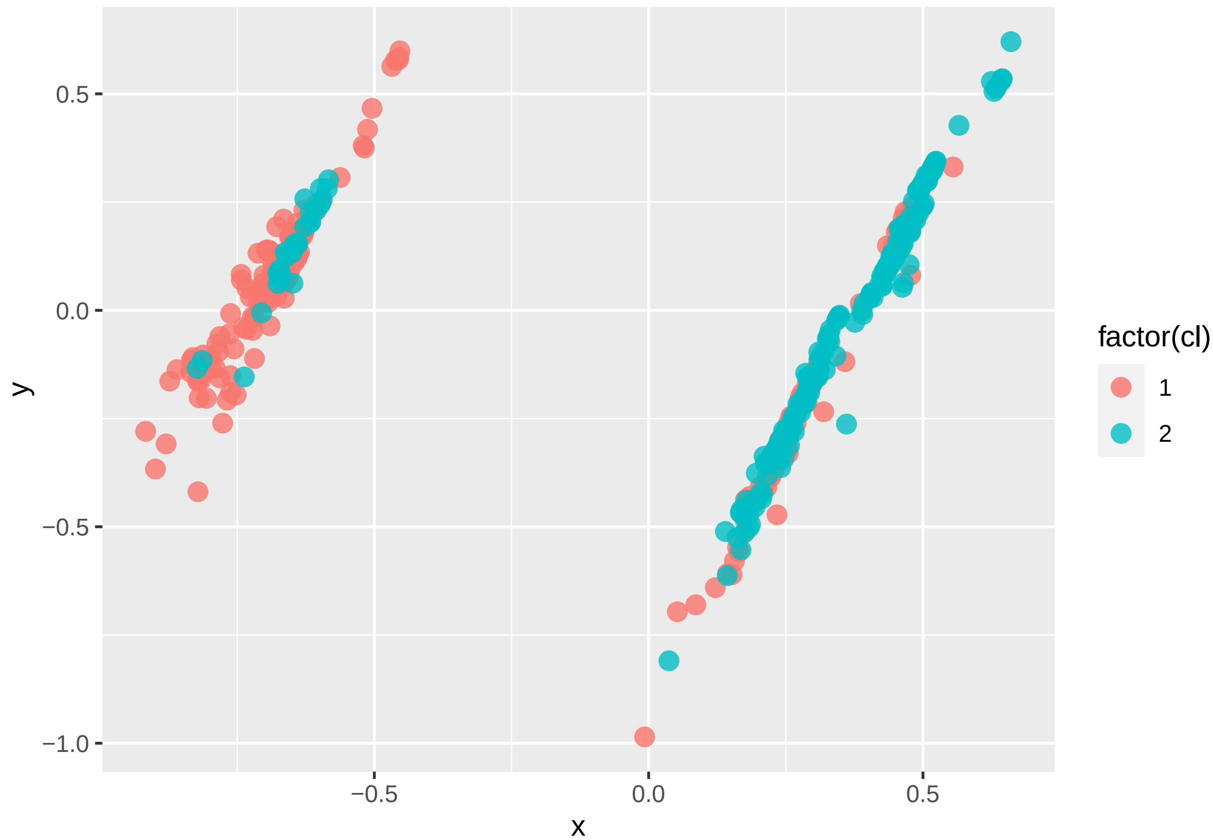
```
        flag = flag - 1
    }

#print("distance matrix:");#cat("\n")
#print(ra_dis[2:sam_n,1:(sam_n-1)], na.print = ""); # 输出距离矩阵
#    print("leastlost matrix:")
#print(leastlost[2:(sam_n-1),1:(sam_n-2)], na.print = ""); # 输出最小损失矩阵
#print("classid matrix:")
#print(classid[2:(sam_n-1),1:(sam_n-2)], na.print = ""); # 输出分类标识矩阵
#    cat("\n")
#print("result")
# 画一个简单的分类示意图
    result1=NULL
    for (p in 1:sam_n) {
        result1 <- cat(result1,p, " ")
        for (w in 1:length(breaks)) {
            if (p == breaks[w] - 1) {
                result1 <- cat(result1, "||")
            }
        }
        if (p == sam_n)
           result1= cat(result1, "\n")
    }
    return(breaks)
}

X_order=X_star[order(X_star$Nyear),]
re <- ocluster(X_order,2)
```

```
## 1   2   3   4   5   6   7   8   9   10   11   12   13   14   15   16   17   18   19   20   21
```

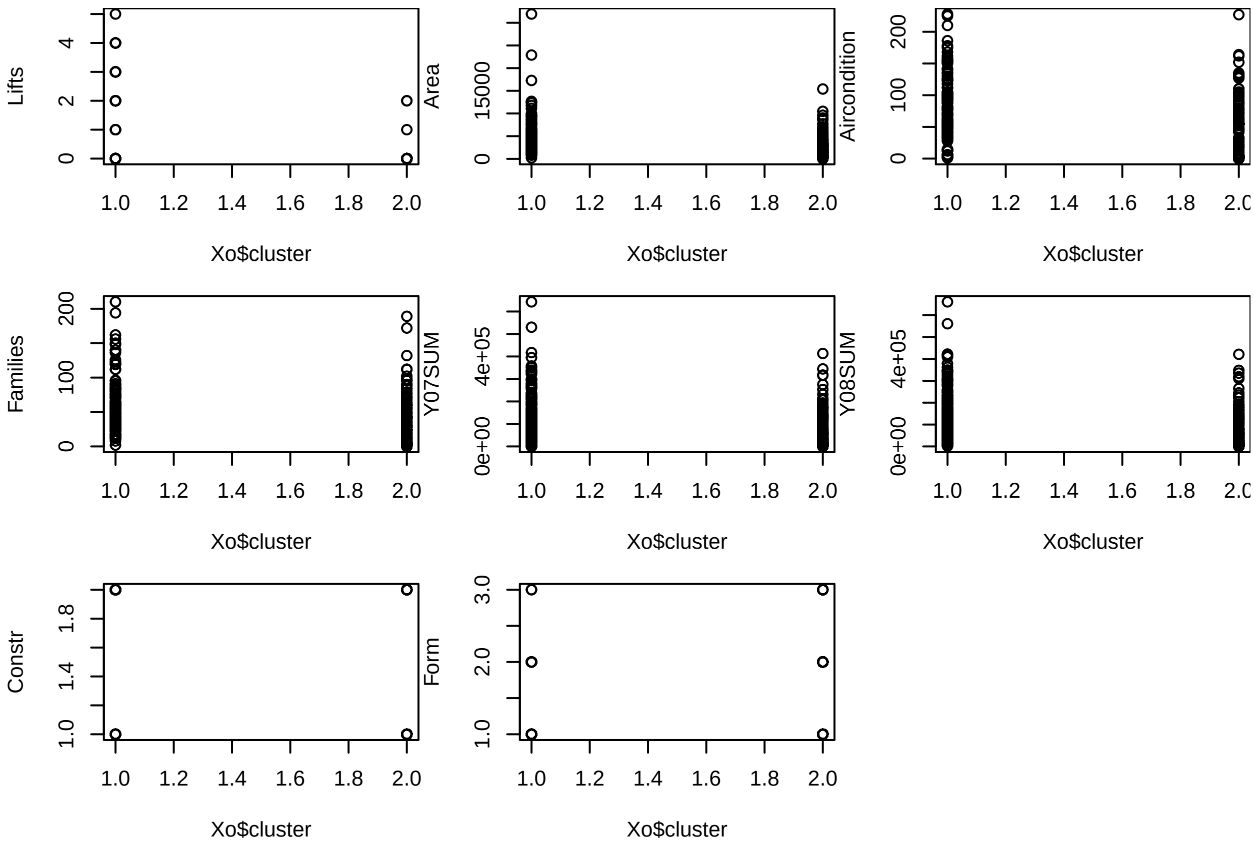
```
cl<-c(rep(1,re[2]-1),rep(2,nrow(X_order)-re[2]+1))
d<-dist(X_order,method = "euclidean")
mds=cmdscale(d,k=2,eig=T)
x = mds$points[,1]
y = mds$points[,2]
p=ggplot(data.frame(x,y),aes(x,y))
p+geom_point(size=3,alpha=0.8,
             aes(colour=factor(cl)))
```




```
X[order(X$Nyear),][re[2],'Nyear']
```

```
## [1] 15
```

```
Xo <- X[order(X$Nyear),]
Xo[,'cluster']=cl
opar<-par(mfrow=c(3,3), mar=c(5.2,4,0,0))
plot(Xo$cluster,Xo$Lifts,ylab = "Lifts")
plot(Xo$cluster,Xo$Area,ylab="Area")
plot(Xo$cluster,Xo$Aircondition,ylab="Aircondition")
plot(Xo$cluster,Xo$Families,ylab="Families")
plot(Xo$cluster,Xo$Y07SUM,ylab="Y07SUM")
plot(Xo$cluster,Xo$Y08SUM,ylab="Y08SUM")
plot(Xo$cluster,Xo$Constr,ylab="Constr")
plot(Xo$cluster,Xo$Form,ylab="Form")
par(opar)
```



利用有序聚类法将目标分为两类，第一类的使用年限小于 15 年，第二类的使用年限大于等于 15 年。使用年限小于 15 年的建筑相对面积更大，家庭更多，电梯数量分布更分散，07、08 年的用电量更多；使用年限大于 15 年的各项都相对更小一些。