班级	xxx	学号	xxx	姓名	xxx

结合课程上机 (R 语言), 完成以下上机作业问题:

注 1 基本要求:

1) 针对题目要求给出解答, 给出核心关键代码, 不必要粘贴所有源代码):

2) 简要概述你通过编程解决此问题所遇到的难点及收获的 R 语言或课程理论等方面的心得。

注 2 评价依据:

1) 解答的完整性、正确性: 是否缺少内容、是否计算无误;

2) 难点分析: 是否记录了解决问题过程中的难点和心得;

3) 核心关键代码是否有恰当的注释;

4) 雷同抄袭等: 超期作业评价不超过 60 分, 雷同抄袭一律 0 分。

1 习题 4.5

4.5 表4.9给出了1991年我国30个省、区、市城镇居民的月平均消费数据,所考察的八个指标如下(单位均为元/人)

X_1: 人均粮食支出; X_2: 人均副食支出;

X_3: 人均烟酒茶支出; X_4: 人均其他副食支出;

X_5: 人均衣着商品支出; X_6: 人均日用品支出;

X_7: 人均燃料支出; X_8: 人均非商品支出.

(1) 求样本相关系数矩阵 R;

(2) 从 R 出发做主成分分析,求各主成分的贡献率及前两个主成分的累计贡献率;

(3) 求出前两个主成分并解释其意义. 按第一主成分得分将30个省、区、市排序,结果如何?

导入数据:

[1]:
```
data <- read.table("ex_4_5.utf8.txt", head=TRUE); data
```

	X1 <dbl>	X2 <dbl>	X3 <dbl>	X4 <dbl>	X5 <dbl>	X6 <dbl>	X7 <dbl>	X8 <dbl>
山西	8.35	23.53	7.51	8.62	17.42	10.00	1.04	11.21
内蒙古	9.25	23.75	6.61	9.19	17.77	10.48	1.72	10.51
⋮								
上海	8.28	64.34	8.00	22.22	20.06	15.12	0.72	22.89
广东	12.47	76.39	5.52	11.24	14.52	22.00	5.46	25.50

1.1 (1)

在第一章中做过，用 cor 获取相关系数：

[2]:
```
R <- cor(data); R
```

	X1	X2	X3	X4	X5	X6	X7	X8
X1	1.000	0.333	-0.054	-0.061	-0.289	0.198	0.348	0.318
X2	0.333	1.000	-0.022	0.398	-0.156	0.711	0.413	0.834
X3	-0.054	-0.022	1.000	0.533	0.496	0.032	-0.139	-0.258
X4	-0.061	0.398	0.533	1.000	0.698	0.467	-0.171	0.312
X5	-0.289	-0.156	0.496	0.698	1.000	0.280	-0.208	-0.081
X6	0.198	0.711	0.032	0.467	0.280	1.000	0.416	0.701
X7	0.348	0.413	-0.139	-0.171	-0.208	0.416	1.000	0.398
X8	0.318	0.834	-0.258	0.312	-0.081	0.701	0.398	1.000

1.2 (2)

R 中用 princomp 来做 PCA，里面有个 cor 参数指定是否使用相关系数：

[3]:
```
pca <- princomp(data, cor=TRUE)
summary(pca)
```

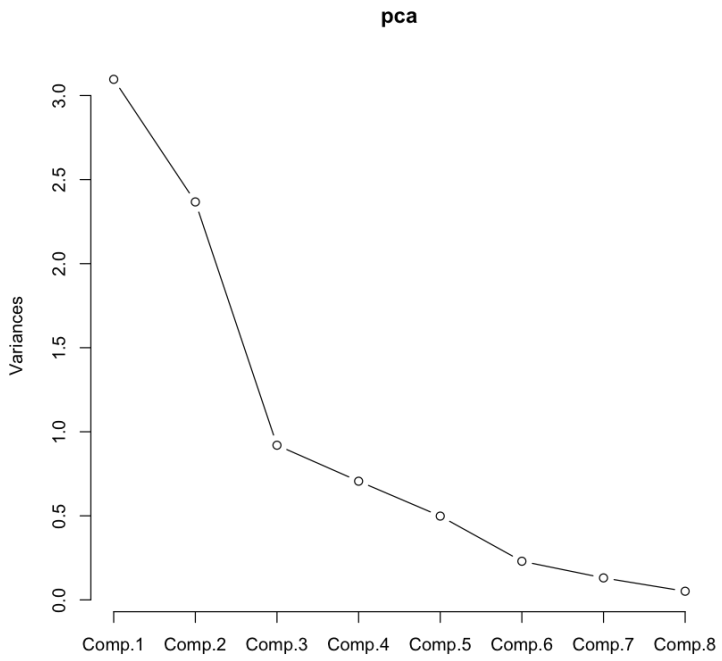
```
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4     Comp.5
Standard deviation     1.759627 1.5385783 0.9591597 0.84019364 0.70600448
Proportion of Variance 0.387036 0.2959029 0.1149984 0.08824067 0.06230529
Cumulative Proportion  0.387036 0.6829389 0.7979373 0.88617801 0.94848330
                           Comp.6     Comp.7      Comp.8
Standard deviation     0.47946669 0.36162932 0.226868982
Proportion of Variance 0.02873604 0.01634697 0.006433692
Cumulative Proportion  0.97721934 0.99356631 1.000000000
```

这里从 summary 输出中就有各主成分的贡献率 Proportion of Variance，以及累积贡献率 Cumulative Proportion。

其中前两个主成分的累积贡献率为 0.6829389。

这里可以作图来对 PCA 结果有更直观的感受：

[4]:
```
plot(pca, type="lines")
```



可以看到前两个主成分所占方差较大，所以只用前两个主成分就可以比较好的描述数据特征了。

1.3 (3)

从 pca$loading 可以获取到个主成分的构成：

[5]:
```
pca$loadings
```

```
Loadings:
   Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
X1  0.250  0.241  0.694  0.377  0.502
X2  0.519                0.225 -0.424         0.282  0.643
X3        -0.475  0.578        -0.510  0.173 -0.381
```

```
X4  0.254 -0.538         0.231        -0.399  0.472 -0.458
X5        -0.575        -0.285  0.516 -0.146 -0.159  0.521
X6  0.493 -0.135 -0.145 -0.224  0.177  0.755        -0.244
X7  0.317  0.261  0.286 -0.768        -0.355  0.131
X8  0.509        -0.271  0.177        -0.305 -0.708 -0.181



               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var  0.125  0.125  0.125  0.125  0.125  0.125  0.125  0.125
Cumulative Var  0.125  0.250  0.375  0.500  0.625  0.750  0.875  1.000
```

取出前两个主成分:

[6]:
```
pca$loadings[,1:2]
```

A matrix: 8 × 2 of type dbl

	Comp.1	Comp.2
X1	0.24960670	0.24123818
X2	0.51923425	0.03760741
X3	-0.01848008	-0.47543851
X4	0.25409160	-0.53808076
X5	0.02169482	-0.57544866
X6	0.49266309	-0.13467554
X7	0.31714669	0.26068239
X8	0.50933155	0.08708140

即得到:

$$Y_1 = 0.25x_1 + 0.519x_2 - 0.018x_3 + 0.254x_4 + 0.022x_5 + 0.493x_6 + 0.317x_7 + 0.509x_8$$

$$Y_2 = 0.241x_1 + 0.038x_2 - 0.475x_3 - 0.538x_4 - 0.575x_5 - 0.135x_6 + 0.261x_7 + 0.087x_8$$

- Y_1 反映了各省人均消费水平，除烟茶酒（X_3）外，其他支出越高，其人均总体消费水平越高，而烟茶酒对其消费水平评价成负相关。
- 在 Y_2 中人烟酒、其他副食、衣着、日用品系数为负；粮食、副食、燃料、非商品系数为正，说明 Y_2 的绝对值越大，各省人均消费的在生活必需品与高档品差异越大。

`pca$scores` 可以获取在个主成分下数据的得分:

[8]:
```
pca$scores
```

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8
山西	-1.713	-0.170	-0.165	0.235	0.569	0.298	-0.486	-0.091
内蒙古	-1.279	0.134	0.306	-0.230	1.012	0.071	-0.037	-0.121
吉林	-1.315	0.875	-0.111	-0.686	0.247	-1.084	0.572	0.059
⋮								
西藏	-0.135	-4.992	2.376	-0.462	-1.441	0.129	0.223	-0.147
上海	3.303	-2.604	-1.700	2.017	-0.031	-0.799	-0.293	-0.088
广东	7.013	2.317	0.827	-1.598	0.208	0.018	-0.246	-0.155

从中我们获取的一主成分的得分，并进行排序：

[9]:
```
Comp.1.scores <- pca$scores[,1]
Comp.1.scores.sorted <- Comp.1.scores[order(Comp.1.scores, decreasing = TRUE)]
```

将输出的数据整理得到下表：

排名	地区	第一主成分得分	排名	地区	第一主成分得分
1	广东	7.01379233	16	宁夏	-0.43775323
2	上海	3.30394931	17	湖南	-0.52687091
3	北京	1.82277984	18	陕西	-0.62321145
4	浙江	1.54097092	19	云南	-0.67809647
5	海南	1.42511451	20	新疆	-0.83249622
6	福建	1.17362616	21	青海	-1.13238706
7	广西	1.07457604	22	安徽	-1.13401837
8	天津	0.44287310	23	甘肃	-1.20243857
9	江苏	0.15591226	24	内蒙古	-1.27970296
10	辽宁	0.04597426	25	贵州	-1.28087147
11	西藏	-0.13551813	26	吉林	-1.31581695
12	四川	-0.13719329	27	黑龙江	-1.34833462
13	山东	-0.14352770	28	河南	-1.51135274
14	湖北	-0.17335839	29	山西	-1.71328041
15	河北	-0.39890334	30	江西	-1.99443644

2　习题 4.6

4.6　表 4.10 是 49 位女性在空腹情况下三个不同时刻的血糖含量（用 X_1, X_2, X_3 表示）和在摄入等量食糖一小时后的三个时刻的血糖含量（用 Y_1, Y_2, Y_3 表示）的观测值（单位：mg/100mL）. 分别从样本协方差矩阵 $\boldsymbol{S}$ 和样本相关系数矩阵 $\boldsymbol{R}$ 出发做主成分分析，求主成分的贡献率和各个主成分. 在两种情况下，你认为应保留几个主成分，其意义如何解释？就此题而言，你认为基于 $\boldsymbol{S}$ 和 $\boldsymbol{R}$ 的分析结果哪个更为合理？

读取数据：

[1]:
```
data <- read.csv("ex_4_6.csv")[-1]; data
```

A data.frame: 49 × 6

x1 <int>	x2 <int>	x3 <int>	y1 <int>	y2 <int>	y3 <int>
60	69	62	97	69	98
56	53	84	103	78	107
⋮					
65	60	70	119	94	89
52	70	76	92	94	100

1. 样本协方差矩阵 S：

[2]:
```
S <- cov(data); S
```

	x1	x2	x3	y1	y2	y3
x1	97.33333	17.809524	12.02976	58.720238	22.35119	61.529762
x2	17.80952	74.579932	14.21854	3.326105	61.62160	-3.855867
x3	12.02976	14.218537	76.96939	41.667517	31.21854	66.109269
y1	58.72024	3.326105	41.66752	779.153912	310.15944	192.423469
y2	22.35119	61.621599	31.21854	310.159439	510.07993	156.185799
y3	61.52976	-3.855867	66.10927	192.423469	156.18580	485.332483

从协方差做 PCA：

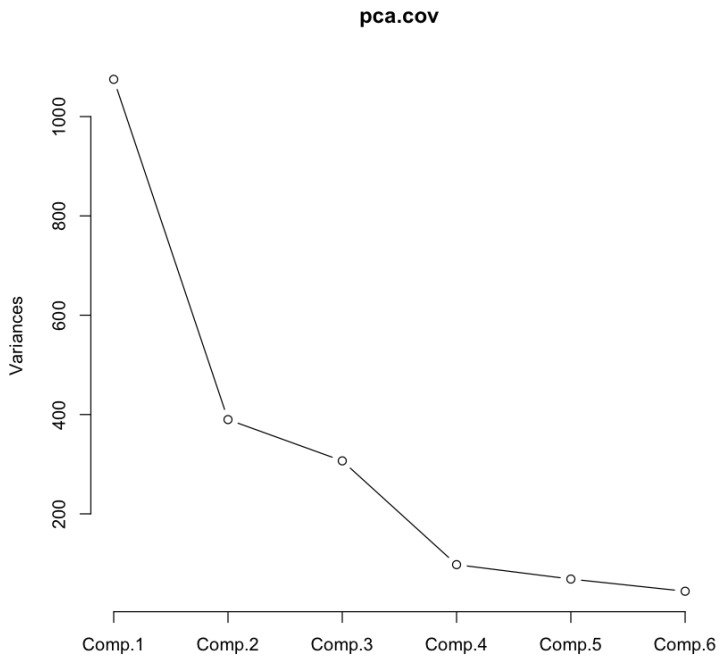
[3]:
```
pca.cov <- princomp(data, cor=FALSE)
summary(pca.cov)
```

```
Importance of components:
                          Comp.1     Comp.2     Comp.3     Comp.4     Comp.5
Standard deviation     32.7872275 19.7452189 17.5131159 9.88479688 8.28776305
Proportion of Variance  0.5423404  0.1966919  0.1547353 0.04929446 0.03465271
Cumulative Proportion   0.5423404  0.7390323  0.8937676 0.94306209 0.97771481
                           Comp.6
Standard deviation     6.64625362
Proportion of Variance 0.02228519
Cumulative Proportion  1.00000000
```

前三个主成分累积贡献率已占近 89.37%，第三主成分贡献率 15.47%，大于剩下三个成分的贡献率总和（0.10623236）。所以认为前三个主成分已包含原始数据的大量信息，所以保留前三个主成分即

可。

[5]:
```
plot(pca.cov, type="lines")
```



同时，前三个主成分所占方差较大，这也证明了上述结论。

2. 样本相关系数矩阵 R:

[6]:
```
R <- cor(data); R
```

	x1	x2	x3	y1	y2	y3
x1	1.0000000	0.20903091	0.1389848	0.21322856	0.1003115	0.28309667
x2	0.2090309	1.00000000	0.1876657	0.01379791	0.3159387	-0.02026709
x3	0.1389848	0.18766571	1.0000000	0.17014805	0.1575558	0.34204532
y1	0.2132286	0.01379791	0.1701481	1.00000000	0.4919877	0.31291525
y2	0.1003115	0.31593872	0.1575558	0.49198771	1.0000000	0.31390826
y3	0.2830967	-0.02026709	0.3420453	0.31291525	0.3139083	1.00000000

从相关系数做 PCA:

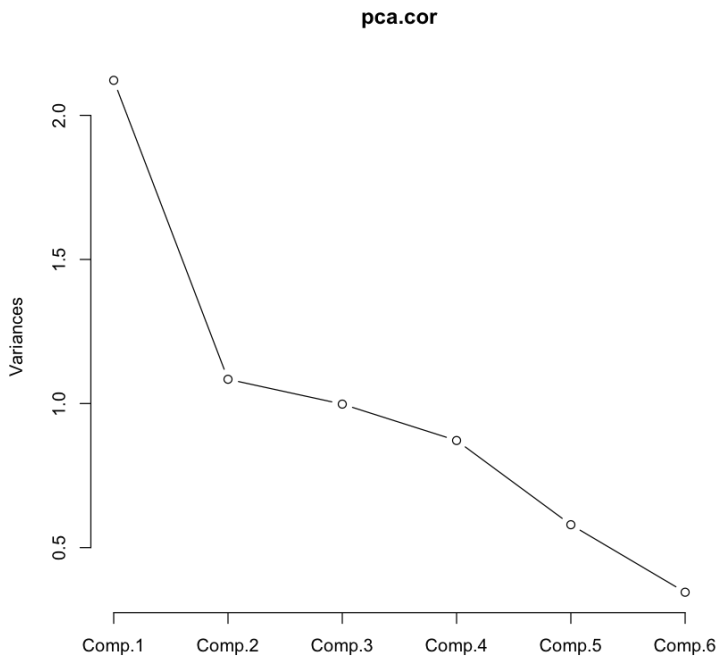
[7]:
```
pca.cor <- princomp(data, cor=TRUE)
summary(pca.cor)
```

```
Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4     Comp.5
Standard deviation     1.4565616 1.0412531 0.9989804 0.9336374 0.76120123
Proportion of Variance 0.3535953 0.1807013 0.1663270 0.1452798 0.09657122
Cumulative Proportion  0.3535953 0.5342966 0.7006236 0.8459034 0.94247458
                           Comp.6
Standard deviation     0.58749684
Proportion of Variance 0.05752542
Cumulative Proportion  1.00000000
```

从以结果可看出前四个主成分累积贡献率已占 84.59% 且第四个主成分的贡献率为 0.1452798，与剩下两个成分的总和（0.15409664）差不多，所以保留前四个主成分即可。

从方差来看，也可以佐证之。

[9]:
```
plot(pca.cor, type="lines")
```



相较之下，我认为使用基于样本协方差矩阵 S 的主成分分析结果更好。基于 S 的前 3 个主成分累积贡献就超过了 89.37%，而基于 R 前 4 个主成分累积贡献才不到 84.59%，显然使用基于 S 的分析更好。

同时考虑到具体的问题，空腹和摄入食糖的测量数据等量纲，无需进行标准化，这种数据基于协方

差矩阵 S 的分析结果更为合理。

3　习题 4.9

4.9　表4.11是25个家庭的成年长子的头长(X_1)、头宽(X_2)与成年次子的头长(Y_1)和头宽(Y_2)的观测数据. 试分别从样本协方差矩阵 $\boldsymbol{\Sigma}$ 和样本相关系数矩阵 $\boldsymbol{R}$ 出发做典型相关分析,求各典型变量对及典型相关系数,检验各典型变量对是否显著相关($\alpha = 0.05$). 两种情况下的结果有何异同?

[1]:
```
data <- read.table("ex_4_9.txt"); data
```

A data.frame: 25 × 4

	X1 <int>	X2 <int>	Y1 <int>	Y2 <int>
1	191	155	179	145
2	195	149	201	152
⋮		⋮		
24	197	167	200	158
25	190	163	187	150

[2]:
```
X <- data[,1:2]
Y <- data[,3:4]
```

1. 协方差阵 S:

[3]:
```
S <- cov(data); S
```

A matrix: 4 × 4 of type dbl

	X1	X2	Y1	Y2
X1	95.29333	52.86833	69.66167	46.11167
X2	52.86833	54.36000	51.31167	35.05333
Y1	69.66167	51.31167	100.80667	56.54000
Y2	46.11167	35.05333	56.54000	45.02333

基于 S 做典型相关分析:

[4]:
```
cca.s <- cancor(X, Y); cca.s
```

$cor 1. 0.788507916294635 2. 0.0537397044242775

$xcoef A matrix: 2 × 2 of type dbl

X1	0.01154653	-0.02857148
X2	0.01443910	0.03816093

$ycoef A matrix: 2 × 2 of type dbl	Y1	0.01025573	-0.03595605
	Y2	0.01637533	0.05349758

$xcenter	**X1** 185.72	**X2**	151.12
$ycenter	**Y1** 183.84	**Y2**	149.24

所以有:

第一对典型变量:

$$V_1 = 0.01154653X_1 + 0.01443910X_2$$
$$W_1 = 0.01025573Y_1 + 0.01637533Y_2$$

第一对典型相关系数 $\rho_1 = 0.788507916294635$.

第二对典型变量:

$$V_2 = -0.02857148X_1 + 0.03816093X_2$$
$$W_2 = -0.03595605Y_1 + 0.05349758Y_2$$

第二对典型相关系数 $\rho_2 = 0.0537397044242775$.

2. 相关系数矩阵 R:

[5]:
```
R <- cor(data); R
```

A matrix: 4 × 4 of type dbl

	X1	X2	Y1	Y2
X1	1.0000000	0.7345555	0.7107518	0.7039807
X2	0.7345555	1.0000000	0.6931573	0.7085504
Y1	0.7107518	0.6931573	1.0000000	0.8392519
Y2	0.7039807	0.7085504	0.8392519	1.0000000

[6]:
```
data.scale <- scale(data)
data.scale.X <- data.scale[,1:2]
data.scale.Y <- data.scale[,3:4]
data.scale
```

A matrix: 25 × 4 of type dbl

	X1	X2	Y1	Y2
1	0.54088217	0.52624987	-0.48205960	-0.63189808
2	0.95064139	-0.28753859	1.70912039	0.41132988
3	-0.48351588	-0.42317000	0.11553495	-0.03576782
⋮				
23	-0.99571490	-1.64385269	-0.78085687	-0.92996321
24	1.15552099	2.15382679	1.60952130	1.30552527
25	0.43844236	1.61130115	0.31473313	0.11326475

基于 R 做典型相关分析:

[7]:
```
cca.r <- cancor(data.scale.X, data.scale.Y); cca.r
```

$cor 1. 0.788507916294635 2. 0.0537397044242769

$xcoef A matrix: 2 × 2 of type dbl

X1	0.1127152	-0.2789099
X2	0.1064583	0.2813576

$ycoef A matrix: 2 × 2 of type dbl

Y1	0.1029701	-0.3610078
Y2	0.1098775	0.3589657

$xcenter **X1** 1.24344978758018e-16 **X2** -6.04932770542632e-16

$ycenter **Y1** -3.3806291099836e-16 **Y2** -1.35974564940966e-15

得到:

第一对典型变量:

$$V_1^* = 0.1127152X_1^* + 0.1064583X_2^*$$
$$W_1^* = 0.1029701Y_1^* + 0.1098775Y_2^*$$

第一对典型相关系数 $\rho_1 = 0.788507916294635$.

第二对典型变量:

$$V_2^* = -0.2789099X_1^* + 0.2813576X_2^*$$
$$W_2^* = -0.3610078Y_1^* + 0.3589657Y_2^*$$

第二对典型相关系数 $\rho_2 = 0.0537397044242769$.

由于样本数据同量纲，所以从协方差阵和相关系数矩阵进行典型相关分析，得到的结果是一样的。

3. 对典型变量进行显著性检验

这里需要自己编写函数实现 [1, 3, 4]:

[8]:
```r
corcoef.test <- function(cor, n, p, q) {
    # 相关系数检验
    # Args:
    #   r: 典型相关系数
    #   n: 样本个数 (n > p + q)
    #   p, q: 向量的维数
    # Returns:
    #   显著性检验表格
    ev <- cor^2
    ev2 <- 1 - ev
    l <- length(ev)
    m <- n - 1 - (p+q+1) / 2
    w <- cbind(NULL)  # 保存中间计算值
    for (i in 1:l) {
      w <- cbind(w,prod(ev2[i:l]))
    }
    Q <- c(NULL); d <- c(NULL)
    for (i in 1:l){
      Q <- cbind(Q, -(m-(i-1)) * log(w[i]))
      d <- cbind(d, (p-i+1) * (q-i+1))
    }
    pvalue <- pchisq(Q, d, lower.tail=FALSE)    # 计算卡方统计量对应的概率
    bat <- cbind(t(Q), t(d), t(pvalue))
    colnames(bat) <- c("Chi-Squared", "df", "pvalue")
    rownames(bat) <- 1:l
    bat    # ret
}
```

[9]:
```r
corcoef.test(cca.r$cor, n=25, p=2, q=2)
```

A matrix: 2 × 3 of type dbl

	Chi-Squared	df	pvalue
1	20.96417998	4	0.0003218897
2	0.05928875	1	0.8076236560

取显著水平为 $\alpha = 0.05$，其中第一对典型变量的检验 p 值为 $0.003 < 0.05$，认为第一对典型变量显

著相关，而第二对典型变量的检验 p 值为 $0.8031 > 0.05$，认为第二对典型变量不是显著相关。

4　习题 5.4

5.4　在有关地震预报的研究中，遇到砂基液化的问题. 选择了 7 个有关因素 $X_1 \sim X_7$，今从已液化和未液化的地层中得到容量分别为 12 与 23 的训练样本，第 1 组为液化，第 2 组为未液化，数据如表 5.9 所示. 假定各总体服从正态分布且协方差矩阵相等，分别就先验概率相等和按比例分配进行 Bayes 判别分析，写出线性判别函数，并给出误判率的回代估计与交叉确认估计.

导入数据:

[1]:
```
data <- read.csv("ex_5_4.csv")[-1]
attach(data)
data
```

A data.frame: 35 × 8

group <int>	x1 <dbl>	x2 <int>	x3 <dbl>	x4 <dbl>	x5 <int>	x6 <dbl>	x7 <int>
1	6.6	39	1.0	6.0	6	0.12	20
1	6.6	39	1.0	6.0	12	0.12	20
⋮							
2	7.8	172	1.0	3.5	6	0.21	45
2	7.8	233	1.0	4.5	6	0.18	45

R 中可以通过 MASS 包的 lda 进行判别分析:

[2]:
```
library(MASS)
```

4.1　先验概率按比例分配

[3]:
```
l = lda(group ~ ., data=data); l
```

```
Call:
lda(group ~ ., data = data)

Prior probabilities of groups:
        1         2
0.3428571 0.6571429

Group means:
```

```
        x1       x2       x3      x4        x5        x6       x7
1 7.358333 73.66667 1.458333 6.00000 15.250000 0.1716667 49.50000
2 7.686957 67.43478 2.043478 5.23913  6.347826 0.2156522 70.34783

Coefficients of linear discriminants:
             LD1
x1 -1.747293e-01
x2  1.416021e-05
x3  2.072826e-01
x4 -2.052228e-01
x5 -1.935168e-01
x6  8.395266e+00
x7  1.917129e-02
```

输出结果中 [5]:

- Call 表示调用方法;
- Prior probabilities of groups 表示先验概率;
- Group means 表示每一类样本的均值;
- Coefficients of linear discriminants 表示线性判别系数;

[4]:
```
# 输出线性判别函数
cat("W =", paste(round(l$scaling, 8), "*x_", 1:7, " +", sep=""))
```

```
W = -0.17472934*x_1 + 1.416e-05*x_2 + 0.20728259*x_3 + -0.20522277*x_4 +
-0.1935168*x_5 + 8.395266*x_6 + 0.01917129*x_7 +
```

对数据进行预测:

[5]:
```
pred <- predict(l)
pred.tab <- data.frame(pred, data.group=group); pred.tab
```

A data.frame: 35 × 5

	class <fct>	posterior.1 <dbl>	posterior.2 <dbl>	LD1 <dbl>	data.group <int>
1	1	0.6859662059	3.140338e-01	-0.9541788	1
2	1	0.9807303897	1.926961e-02	-2.1152796	1
⋮					
34	2	0.0324563440	9.675437e-01	0.5859424	2
35	2	0.1036058164	8.963942e-01	0.1297254	2

输出结果分别为分类结果和后验概率。在后面加上了数据的真实分类情况作为比较。

下面查看误判情况:

[6]:
```
table(group, pred$class)
```

```
group  1  2
    1 11  1
    2  1 22
```

有两个样本误判:

[7]:
```
bad <- pred.tab[pred$class != data$group,]; bad
```

A data.frame: 2 × 5

	class	posterior.1	posterior.2	LD1	data.group
	<fct>	<dbl>	<dbl>	<dbl>	<int>
9	2	0.2119048	0.7880952	-0.1816424	1
29	1	0.7578938	0.2421062	-1.0868825	2

- 9 号样本误判到 G_2;
- 29 号样本误判到 G_1.

[8]:
```
length(bad$class) / length(pred$class)
```

0.0571428571428571

误判率为: 5.71%

交叉确认误判率 [6]:

[9]:
```
ll <- lda(group ~ ., data=data, CV=TRUE)
```

[10]:
```
table(group, ll$class)
```

```
group  1  2
    1  8  4
    2  2 21
```

具体的误判样本:

[11]:
```
cv.bad <- pred.tab[ll$class != data$group,]; cv.bad
```

A data.frame: 6 × 5

	class	posterior.1	posterior.2	LD1	data.group
	<fct>	<dbl>	<dbl>	<dbl>	<int>
5	1	0.6533235	0.3466765	-0.8997285	1
9	2	0.2119048	0.7880952	-0.1816424	1
11	1	0.6906278	0.3093722	-0.9621920	1
12	1	0.6832010	0.3167990	-0.9494561	1
29	1	0.7578938	0.2421062	-1.0868825	2
35	2	0.1036058	0.8963942	0.1297254	2

计算误判率:

[12]:
```
length(cv.bad$class) / length(pred$class)
```

0.171428571428571

共有 6 个样本误判，误判率 17.14%.

4.2 先验概率相等

假设 $\Sigma_1 = \Sigma_2$，先验概率按相同的条件下，还是用 lda 函数来做，需要多传一个参数指定先验概率。

[13]:
```
le <- lda(group ~ ., data = data, prior = c(1, 1) / 2)
print(le)
```

```
Call:
lda(group ~ ., data = data, prior = c(1, 1)/2)

Prior probabilities of groups:
  1   2
0.5 0.5

Group means:
        x1       x2       x3      x4        x5        x6       x7
1 7.358333 73.66667 1.458333 6.00000 15.250000 0.1716667 49.50000
2 7.686957 67.43478 2.043478 5.23913  6.347826 0.2156522 70.34783

Coefficients of linear discriminants:
             LD1
x1 -1.747293e-01
x2  1.416021e-05
```

```
x3  2.072826e-01
x4 -2.052228e-01
x5 -1.935168e-01
x6  8.395266e+00
x7  1.917129e-02
```

[14]:
```
# 输出线性判别函数
cat("W =", paste(round(le$scaling, 8), "*x_", 1:7, " +", sep=""))
```

```
W = -0.17472934*x_1 + 1.416e-05*x_2 + 0.20728259*x_3 + -0.20522277*x_4 +
-0.1935168*x_5 + 8.395266*x_6 + 0.01917129*x_7 +
```

预测:

[15]:
```
pred_e <- predict(le)
pred_e.tab <- data.frame(pred_e, data.group=group); pred.tab
```

A data.frame: 35 × 5

	class <fct>	posterior.1 <dbl>	posterior.2 <dbl>	LD1 <dbl>	data.group <int>
1	1	0.6859662059	3.140338e-01	-0.9541788	1
2	1	0.9807303897	1.926961e-02	-2.1152796	1
⋮					
34	2	0.0324563440	9.675437e-01	0.5859424	2
35	2	0.1036058164	8.963942e-01	0.1297254	2

误判情况:

[16]:
```
table(group, pred_e$class)
```

```
group  1  2
    1 11  1
    2  1 22
```

[17]:
```
bad_e <- pred_e.tab[pred_e$class != data$group,]; bad_e
```

A data.frame: 2 × 5

	class <fct>	posterior.1 <dbl>	posterior.2 <dbl>	LD1 <dbl>	data.group <int>
9	2	0.3400897	0.6599103	0.2444662	1
29	1	0.8571422	0.1428578	-0.6607739	2

[18]:
```
# 误判率
length(bad_e$class) / length(pred_e$class)
```

0.0571428571428571

交叉确认:

[19]:
```
lel <- lda(group ~ ., data=data,  prior = c(1, 1) / 2, CV=TRUE)
table(group, lel$class)
```

```
group  1  2
    1 11  1
    2  3 20
```

[20]:
```
cv.bad_e <- pred_e.tab[lel$class != data$group,]; cv.bad_e
```

A data.frame: 4 × 5

	class <fct>	posterior.1 <dbl>	posterior.2 <dbl>	LD1 <dbl>	data.group <int>
9	2	0.3400897	0.6599103	0.24446615	1
28	2	0.4769042	0.5230958	0.03409383	2
29	1	0.8571422	0.1428578	-0.66077393	2
35	2	0.1813542	0.8186458	0.55583396	2

[21]:
```
# 误判率
length(cv.bad_e$class) / length(pred_e$class)
```

0.114285714285714

这里得到了误判率 11.43%.

5 习题 5.5

5.5 考察鸢尾属植物中三个不同品种的花的如下四个形状指标:

X_1:萼片长度; X_2:萼片宽度; X_3:花瓣长度; X_4:花瓣宽度.

从这三个品种(记为 1,2,3)各选取 50 株,测得上述指标的取值如表 5.10 所示.假定三个品种的这 4 个指标均服从 4 维正态分布,且先验概率相等,按下列要求进行 Bayes 判别分析:

(1) 只考虑指标 X_2 和 X_4，并假定各总体协方差矩阵不全相等，给出误判率的回代估计和交叉确认估计；

(2) 只考虑指标 X_2 和 X_4，并假定各总体协方差矩阵相等，写出线性判别函数，给出误判率的回代估计和交叉确认估计并与(1)中结果作比较；

(3) 假定有新样品 $x_0 = (x_2, x_4)^T = (35, 18)^T$，在(1)，(2)之下，该样品分别被判归哪个总体？

(4) 利用全部 4 个指标重复(1)和(2)的分析，结果如何？是否所用指标越多，分类效果越好？再尝试其他几种指标组合，情况又如何？

鸢尾花（iris）分类是常用数据集，在 R 有内置：

[1]:
```
data(iris)
attach(iris)
iris
```

A data.frame: 150 × 5

Sepal.Length <dbl>	Sepal.Width <dbl>	Petal.Length <dbl>	Petal.Width <dbl>	Species <fct>
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
⋮	⋮	⋮	⋮	⋮
6.2	3.4	5.4	2.3	virginica
5.9	3.0	5.1	1.8	virginica

这些数据形状与书上题目所给数据一直，只是 x_i 值相差常数倍（10x）。鸢尾花的三个品种在这里分别叫做 setosa、versicolor 和 virginica。

可以把数据放大 10 倍，得到书上的数据（这里并无必要，所以不这么做）：

[5]:
```
# iris[-5] <- iris[-5] * 10; iris
```

5.1 (2)

先做**总体协方差矩阵相同**的情况：

只考虑 X_2、X_4 即 `Sepal.Width` 和 `Petal.Width`:

[6]:
```
library(MASS)
```

[7]:
```
l1 = lda(Species ~ Sepal.Width + Petal.Width, data=iris); l1
```

```
Call:
lda(Species ~ Sepal.Width + Petal.Width, data = iris)
```

```
Prior probabilities of groups:
    setosa versicolor  virginica
 0.3333333  0.3333333  0.3333333

Group means:
           Sepal.Width Petal.Width
setosa           3.428       0.246
versicolor       2.770       1.326
virginica        2.974       2.026

Coefficients of linear discriminants:
                  LD1       LD2
Sepal.Width -1.986964 2.6800746
Petal.Width  5.477136 0.8169648

Proportion of trace:
   LD1    LD2
0.9884 0.0116
```

输出线性判别函数的系数:

[8]:
```
l1$scaling
```

A matrix: 2 × 2 of type dbl

	LD1	LD2
Sepal.Width	-1.986964	2.6800746
Petal.Width	5.477136	0.8169648

得到线性判别函数:

$$W_1 = -1.98696447x_1 + 5.47713646x_2$$
$$W_2 = 2.6800746x_1 + 0.81696482x_2$$

为方便进行误判确认，封装一个工具函数:

[10]:
```
prederr <- function(pred, true.group) {
    # lda 判别分析预测误判
    # Args:
    #    pred: lda 的预测, e.g. predict(ldm(Species ~ ., data = data))
```

```
    #    true.group: 实际的分类（data 的 label）
    # Returns:
    #    list(`tab`, `err`, `err.ratio`)
    #      tab: 实际与预判交叉表
    #      err: 错误分类数据列表
    #      err.ratio: 误判率
    tab <- table(Species, pred$class)
    err <- data.frame(pred, true.group)[pred$class != true.group,]
    err.ratio <- length(err$class) / length(pred$class)
    list(`tab`=tab, `err`=err, `err.ratio`=err.ratio)
}
```

[11]:
```
pred1 <- predict(l1)
prederr(pred1, Species)
```

```
$tab

Species      setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         49         1
  virginica       0          4        46

$err
         class posterior.setosa posterior.versicolor posterior.virginica
71   virginica     9.920568e-18            0.2331276          0.76687239
120 versicolor     1.919451e-19            0.7114697          0.28853027
130 versicolor     3.125859e-15            0.8792078          0.12079219
134 versicolor     5.868316e-15            0.9605066          0.03949344
135 versicolor     1.037139e-14            0.9878440          0.01215599
       x.LD1      x.LD2 true.group
71  3.006460  0.8730808 versicolor
120 3.350283 -2.0520832  virginica
130 2.308425  0.1736730  virginica
134 2.158105 -0.4440384  virginica
135 2.007784 -1.0617498  virginica

$err.ratio
```

```
[1] 0.03333333
```

得到误判率 3.33%.

交叉确认误判率:

[12]:
```
l1.cv <- lda(Species ~ Sepal.Width + Petal.Width, data=iris, CV=TRUE)
prederr(data.frame(class=l1.cv$class, l1.cv$posterior), Species)
```

```
$tab

Species      setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0          4        46

$err
         class       setosa versicolor   virginica true.group
71   virginica 5.100762e-18  0.2004581 0.799541885 versicolor
78   virginica 2.536707e-17  0.4845668 0.515433247 versicolor
120 versicolor 2.400102e-19  0.7871770 0.212822971  virginica
130 versicolor 1.021173e-15  0.8990896 0.100910442  virginica
134 versicolor 1.776898e-15  0.9704931 0.029506851  virginica
135 versicolor 3.229543e-15  0.9926251 0.007374932  virginica

$err.ratio
[1] 0.04
```

得到误判率 4%.

5.2 (1)

总体协方差矩阵不相同的情况, 用 qda 来做:

[13]:
```
qd <- qda(Species ~ Sepal.Width + Petal.Width, data=iris); qd
```

```
Call:
qda(Species ~ Sepal.Width + Petal.Width, data = iris)
```

```
Prior probabilities of groups:
    setosa versicolor  virginica
 0.3333333  0.3333333  0.3333333

Group means:
           Sepal.Width Petal.Width
setosa           3.428       0.246
versicolor       2.770       1.326
virginica        2.974       2.026
```

预测，并查看误判情况：

[14]:
```
predq <- predict(qd)
prederr(predq, Species)
```

```
$tab

Species      setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         46         4
  virginica       0          3        47

$err
         class posterior.setosa posterior.versicolor posterior.virginica
69   virginica     3.612396e-38            0.1288885          0.87111146
71   virginica     5.782918e-51            0.2345794          0.76542060
78   virginica     7.578016e-46            0.3801197          0.61988031
84   virginica     1.621740e-41            0.3894875          0.61051249
130 versicolor     4.565572e-40            0.7817783          0.21822173
134 versicolor     1.077297e-35            0.8674122          0.13258783
135 versicolor     1.359084e-31            0.9204160          0.07958403
    true.group
69  versicolor
71  versicolor
78  versicolor
84  versicolor
130  virginica
```

```
134  virginica
135  virginica

$err.ratio
[1] 0.04666667
```

得到误判率 4.67%.

交叉确认误判率:

[15]:
```
qd.cv <- qda(Species ~ Sepal.Width + Petal.Width, data=iris, CV=TRUE)
prederr(data.frame(class=qd.cv$class, qd.cv$posterior), Species)
```

```
$tab

Species      setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         45         5
  virginica       0          3        47

$err
         class       setosa versicolor  virginica true.group
69   virginica 4.037306e-38 0.02642376 0.97357624 versicolor
71   virginica 6.294572e-51 0.16685743 0.83314257 versicolor
73   virginica 7.169125e-37 0.48449460 0.51550540 versicolor
78   virginica 8.162732e-46 0.33228997 0.66771003 versicolor
84   virginica 1.761570e-41 0.33684744 0.66315256 versicolor
130 versicolor 4.722285e-40 0.80861274 0.19138726  virginica
134 versicolor 1.104200e-35 0.88907345 0.11092655  virginica
135 versicolor 1.390787e-31 0.94188622 0.05811378  virginica

$err.ratio
[1] 0.05333333
```

得到误判率 5.33%.

5.3 (3)

新样本:

[16]:
```
x0 = data.frame(Sepal.Width = c(35), Petal.Width=c(18)); x0
```

A data.frame: 1 × 2

Sepal.Width	Petal.Width
<dbl>	<dbl>
35	18

带入 lda 模型:

[17]:
```
predict(l1, newdata=x0)
```

$class virginica *Levels*: 1. 'setosa' 2. 'versicolor' 3. 'virginica'

$posterior A matrix: 1 × 3 of type dbl

	setosa	versicolor	virginica
1	6.24321e-146	4.105739e-87	1

$x A matrix: 1 × 2 of type dbl

	LD1	LD2
1	28.5506	99.33428

判为 `virginica` 类型，即品种 3。

带入 qda 模型:

[18]:
```
predict(qd, newdata=x0)
```

$class virginica *Levels*: 1. 'setosa' 2. 'versicolor' 3. 'virginica'

$posterior A matrix: 1 × 3 of type dbl

	setosa	versicolor	virginica
1	0	9.004664e-244	1

两种模型给出了相同的预测结果。

5.4 (4)

使用全部指标。

5.4.1 lda

[19]:
```
l2 = lda(Species ~ ., data=iris); l2
```

```
Call:
lda(Species ~ ., data = iris)
```

```
Prior probabilities of groups:
    setosa versicolor  virginica
 0.3333333  0.3333333  0.3333333

Group means:
           Sepal.Length Sepal.Width Petal.Length Petal.Width
setosa            5.006       3.428        1.462       0.246
versicolor        5.936       2.770        4.260       1.326
virginica         6.588       2.974        5.552       2.026

Coefficients of linear discriminants:
                    LD1         LD2
Sepal.Length  0.8293776  0.02410215
Sepal.Width   1.5344731  2.16452123
Petal.Length -2.2012117 -0.93192121
Petal.Width  -2.8104603  2.83918785

Proportion of trace:
   LD1    LD2
0.9912 0.0088
```

误判:

[20]:
```
pred2 <- predict(l2)
prederr(pred2, Species)
```

```
$tab

Species      setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0          1        49

$err
         class posterior.setosa posterior.versicolor posterior.virginica
71   virginica     7.408118e-28            0.2532282           0.7467718
84   virginica     4.241952e-32            0.1433919           0.8566081
134 versicolor     1.283891e-28            0.7293881           0.2706119
```

```
        x.LD1      x.LD2 true.group
71  -3.715896  1.0445144 versicolor
84  -4.498466 -0.8827499 versicolor
134 -3.815160 -0.9429859  virginica

$err.ratio
[1] 0.02
```

交叉确认误判率:

[21]:
```
l2.cv <- lda(Species ~ ., data=iris, CV=TRUE)
prederr(data.frame(class=l2.cv$class, l2.cv$posterior), Species)
```

```
$tab

Species      setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0          1        49

$err
         class       setosa versicolor virginica true.group
71   virginica 1.302246e-28 0.17727267 0.8227273 versicolor
84   virginica 1.125494e-33 0.09924153 0.9007585 versicolor
134 versicolor 5.464475e-29 0.78762376 0.2123762  virginica

$err.ratio
[1] 0.02
```

误判率比只用两个指标明显提升了。

5.4.2 qda

[22]:
```
qd2 = qda(Species ~ ., data=iris); qd2
```

```
Call:
qda(Species ~ ., data = iris)
```

```
Prior probabilities of groups:
    setosa versicolor  virginica
 0.3333333  0.3333333  0.3333333

Group means:
           Sepal.Length Sepal.Width Petal.Length Petal.Width
setosa            5.006       3.428        1.462       0.246
versicolor        5.936       2.770        4.260       1.326
virginica         6.588       2.974        5.552       2.026
```

误判:

[23]:
```
predq2 <- predict(qd2)
prederr(predq2, Species)
```

```
$tab

Species      setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0          1        49

$err
         class posterior.setosa posterior.versicolor posterior.virginica
71   virginica    1.052723e-103            0.3359442           0.6640558
84   virginica    4.102009e-114            0.1543483           0.8456517
134 versicolor    4.550670e-111            0.6049611           0.3950389
    true.group
71  versicolor
84  versicolor
134  virginica

$err.ratio
[1] 0.02
```

交叉确认误判率:

[24]:
```
qd2.cv <- qda(Species ~ ., data=iris, CV=TRUE)
prederr(data.frame(class=qd2.cv$class, qd2.cv$posterior), Species)
```

```
$tab

Species      setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         47         3
  virginica       0          1        49

$err
         class        setosa versicolor virginica true.group
69   virginica  1.376175e-89 0.31342177 0.6865782 versicolor
71   virginica 1.329043e-103 0.16164225 0.8383577 versicolor
84   virginica 4.504693e-114 0.07133282 0.9286672 versicolor
134 versicolor 4.988739e-111 0.66319758 0.3368024  virginica

$err.ratio
[1] 0.02666667
```

同样，比只使用两个特征做出来的效果更好，误判率更低。

6 习题 6.3

6.3 1976 年 74 个国家和地区的人口出生率 X_1 和死亡率 X_2 的观测数据如表 6.16 所示（国家与地区名从略），表中数据是每 10 万人的出生数和死亡数. 试对这 74 个国家与地区按人口出生率与死亡率进行快速聚类分析.

(1) 给出聚 3 类的结果，并画出 (X_1, X_2) 的散点图，该图是否反映了各类的集聚性？

(2) 聚为 4 类的结果又如何？

(3) 给出用绝对距离（L_1 距离）快速聚类的相应于(1)和(2)的结果.

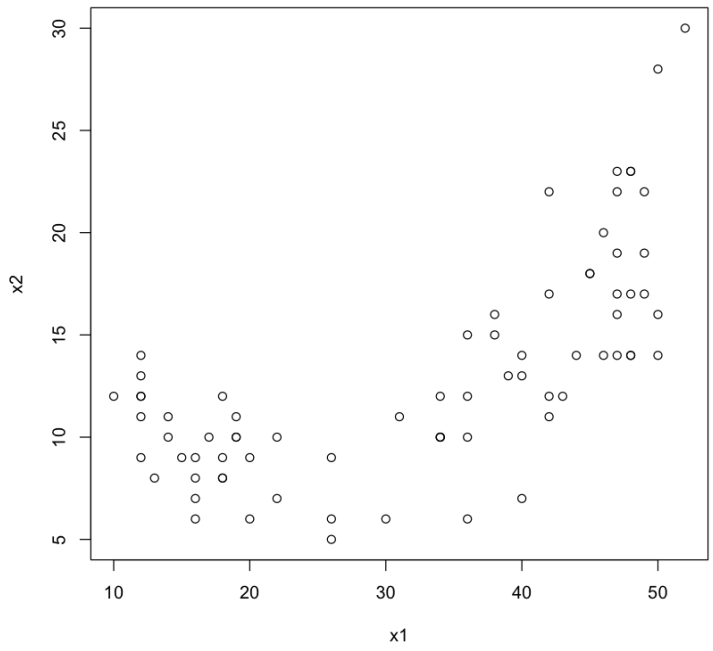
[1]:
```
data <- read.csv("ex_6_3.csv")
attach(data)
data
```

A data.frame: 74 × 3

Area	x1	x2
<int>	<int>	<int>
1	52	30
2	50	16
3	47	23
⋮	⋮	⋮
72	42	17
73	18	8
74	45	18

把数据画在散点图上:

[2]:
```
plot(x2 ~ x1, data=data)
```



6.1 (1) 聚为 3 类

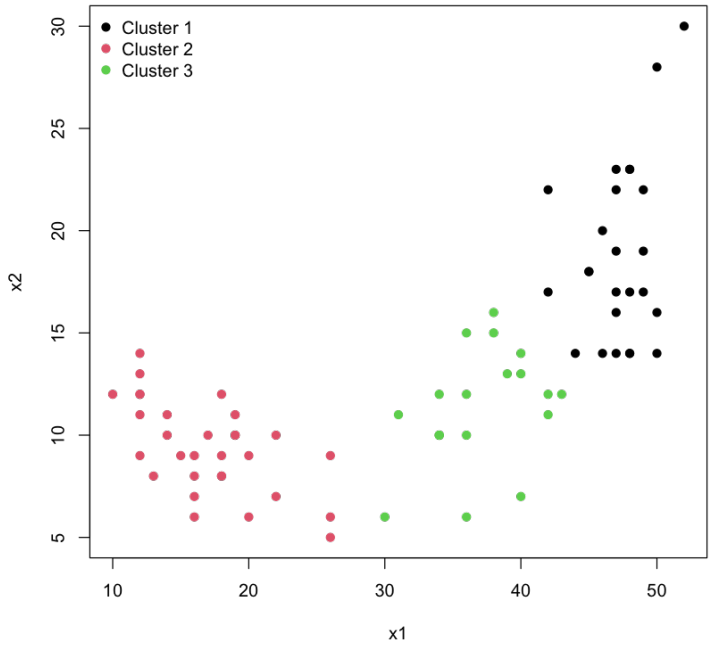
[3]:
```
km3 <- kmeans(data[-1], 3); km3
```

```
K-means clustering with 3 clusters of sizes 25, 30, 19


Cluster means:
     x1        x2
```

```
1 47.24 18.840000
2 17.00  9.366667
3 37.00 11.315789

Clustering vector:
 [1] 1 1 1 2 2 2 1 2 3 2 3 1 2 2 3 2 3 2 2 3 3 1 2 2 2 1 2 3 2 3 3 3 1 2 1 2 1 3
[39] 2 1 3 3 1 1 1 2 1 1 3 3 2 2 1 2 1 3 2 2 1 2 2 1 1 3 3 2 1 2 2 1 3 1 2 1

Within cluster sum of squares by cluster:
[1] 599.9200 712.9667 390.1053
 (between_SS / total_SS =  89.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

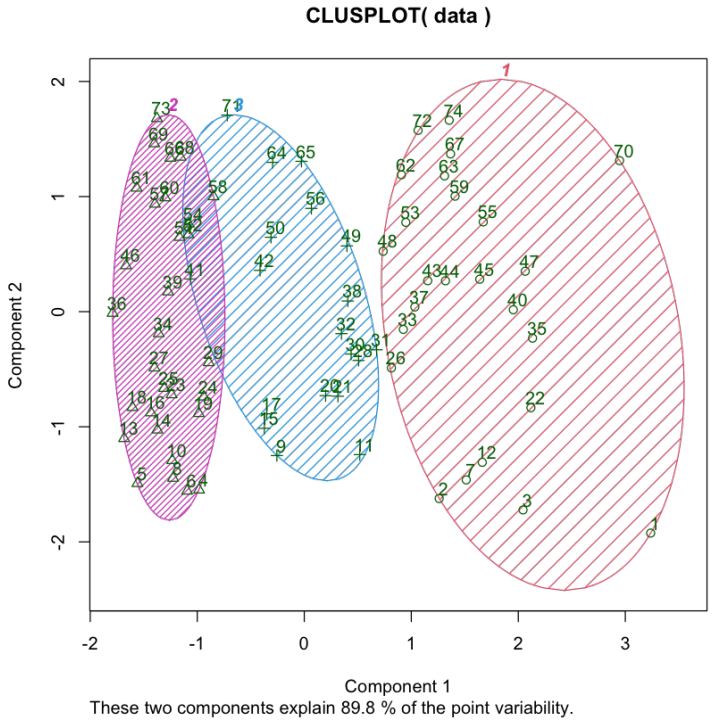
将结果画在散点图里 [7]:

[4]:
```
plot(x2 ~ x1,
     col=km3$cluster, pch = 19,
     data=data)
legend("topleft", legend = paste("Cluster", 1:3),
       col = 1:3, pch = 19, bty = "n")
```



用前两个主成分绘制聚类图 [8]:

[5]:
```
library(cluster)
clusplot(data, km3$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```



反应出了各类的聚集性。

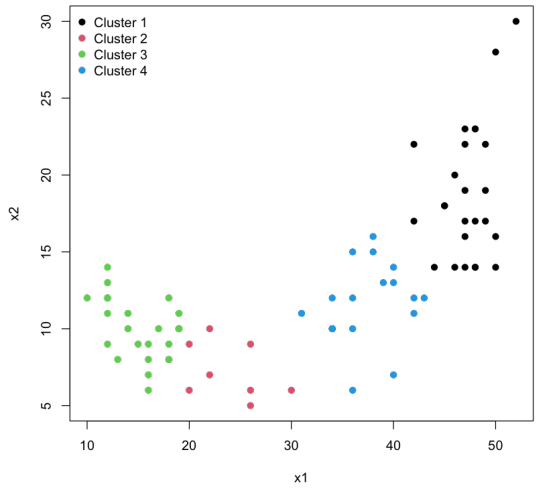
6.2 (2) 聚为 4 类

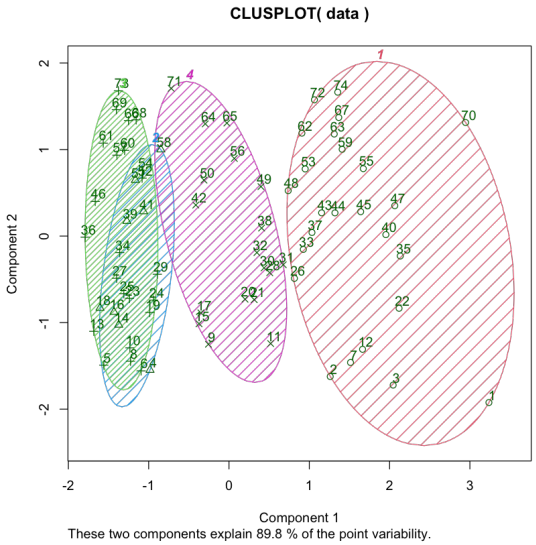
[6]:
```
km4 <- kmeans(data[-1], 4); km4
```

```
K-means clustering with 4 clusters of sizes 25, 8, 23, 18

Cluster means:
        x1        x2
1 47.24000 18.840000
2 24.00000  7.250000
3 15.13043  9.956522
4 37.38889 11.611111

Clustering vector:
 [1] 1 1 1 2 3 3 1 3 4 3 4 1 3 2 4 2 4 2 3 4 4 1 3 3 3 1 3 4 3 4 4 4 1 3 1 3 1 4
[39] 2 1 2 4 1 1 1 3 1 1 4 4 2 3 1 3 1 4 3 2 1 3 3 1 1 4 4 3 1 3 3 1 4 1 3 1

Within cluster sum of squares by cluster:
[1] 599.9200 111.5000 265.5652 308.5556
```

```
 (between_SS / total_SS =  92.0 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

聚类结果图:





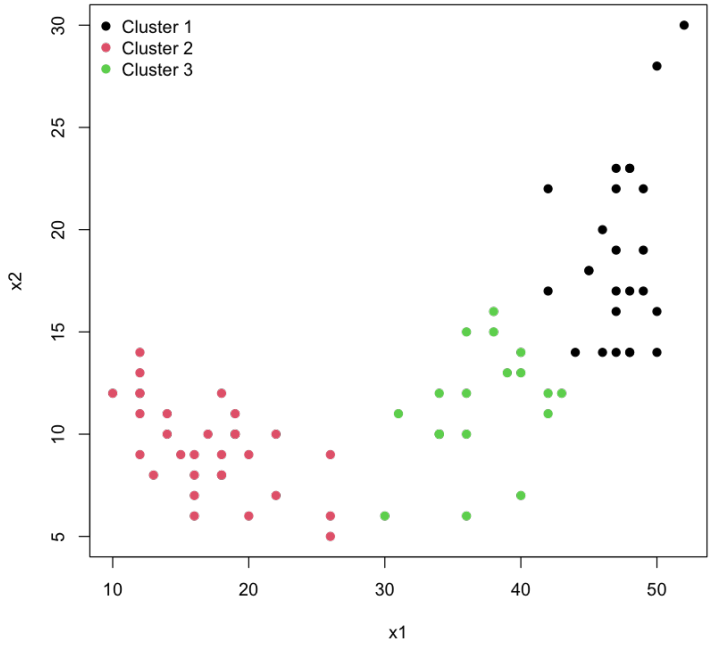
6.3 (3) 绝对距离

6.3.1 聚成 3 类

[10]:
```
pam3 <- pam(data[-1], 3, metric = "manhattan"); pam3
```

```
Medoids:
     ID x1 x2
[1,] 63 47 17
[2,] 27 16  9
[3,] 56 36 12
Clustering vector:
 [1] 1 1 1 2 2 2 1 2 3 2 3 1 2 2 3 2 3 2 2 3 3 1 2 2 2 1 2 3 2 3 3 3 1 2 1 2 1 3
[39] 2 1 3 3 1 1 1 2 1 1 3 3 2 2 1 2 1 3 2 2 1 2 2 1 1 3 3 2 1 2 2 1 3 1 2 1
Objective function:
  build    swap
5.22973 5.22973
```

```
Available components:
 [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
 [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```



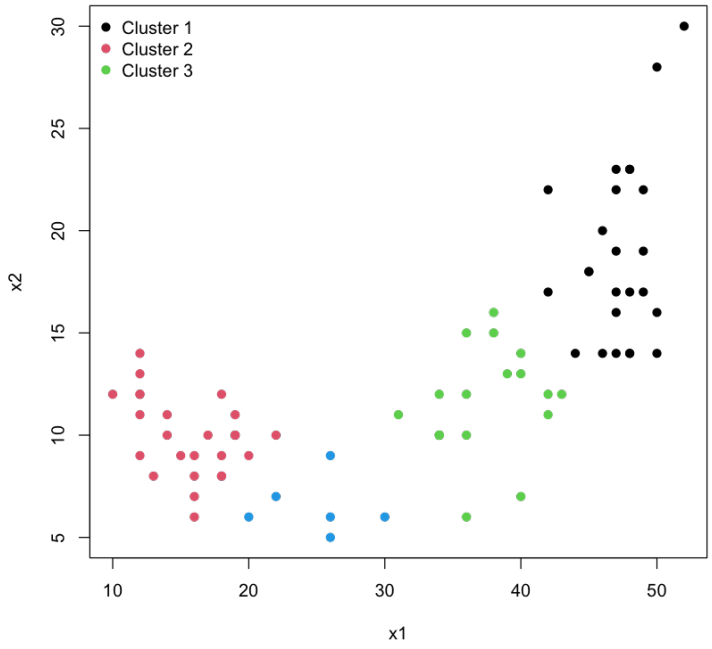
结果和 K-Means 聚类类似。

6.3.2 聚成 4 类

[12]:
```
pam4 <- pam(data[-1], 4, metric = "manhattan"); pam4
```

```
Medoids:
     ID x1 x2
[1,] 63 47 17
[2,] 27 16  9
[3,] 56 36 12
[4,] 39 26  6
Clustering vector:
 [1] 1 1 1 2 2 2 1 2 3 2 3 1 2 4 3 4 3 4 2 3 3 1 2 2 2 1 2 3 2 3 3 3 1 2 1 2 1 3
[39] 4 1 4 3 1 1 1 2 1 1 3 3 2 2 1 2 1 3 2 4 1 2 2 1 1 3 3 2 1 2 2 1 3 1 2 1
Objective function:
   build     swap
```

```
4.621622 4.621622

Available components:
 [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
 [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```



结果和 K-Means 聚类在边界上有部分区别。

7　习题 6.6

6.6 欧洲各国语言有许多相似之处，有的甚至十分相近. 以 E, N, Da, Du, G, Fr, S, I, P, H, Fi 分别表示英语、挪威语、丹麦语、荷兰语、德语、法语、西班牙语、意大利语、波兰语、匈牙利语和芬兰语这 11 种语言. 人们以两种语言对 1 ~ 10 数字拼写中第一个字母不相同的个数定义两种语言间的“距离”，这种“距离”是广义距离. 例如，英语和挪威语只有数字 1 和 8 的第一个字母不同，故这两种语言的距离定义为 2. 这样得到这 11 种语言的距离矩阵为：

$$
\boldsymbol{D} = \begin{bmatrix}
0 & & & & & & & & & & \\
2 & 0 & & & & & & & & & \\
2 & 1 & 0 & & & & & & & & \\
7 & 5 & 6 & 0 & & & & & & & \\
6 & 4 & 5 & 5 & 0 & & & & & & \\
6 & 6 & 6 & 9 & 7 & 0 & & & & & \\
6 & 6 & 5 & 9 & 7 & 2 & 0 & & & & \\
6 & 6 & 5 & 9 & 7 & 1 & 1 & 0 & & & \\
7 & 7 & 6 & 10 & 8 & 5 & 3 & 4 & 0 & & \\
9 & 8 & 8 & 8 & 9 & 10 & 10 & 10 & 10 & 0 & \\
9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 8 & 0
\end{bmatrix}
$$

用下列方法对这 11 种语言进行谱系聚类，写出聚类过程，并画出谱系图：

(1) 最短距离法；

(2) 最长距离法；

(3) 类平均距离法；

(4) 重心距离法.

[40]:
```
data <- read.csv("ex_6_6.csv", header=TRUE, row.names=1)
```

计算距离：

[43]:
```
d <- as.dist(data); d    # dist: 距离
```

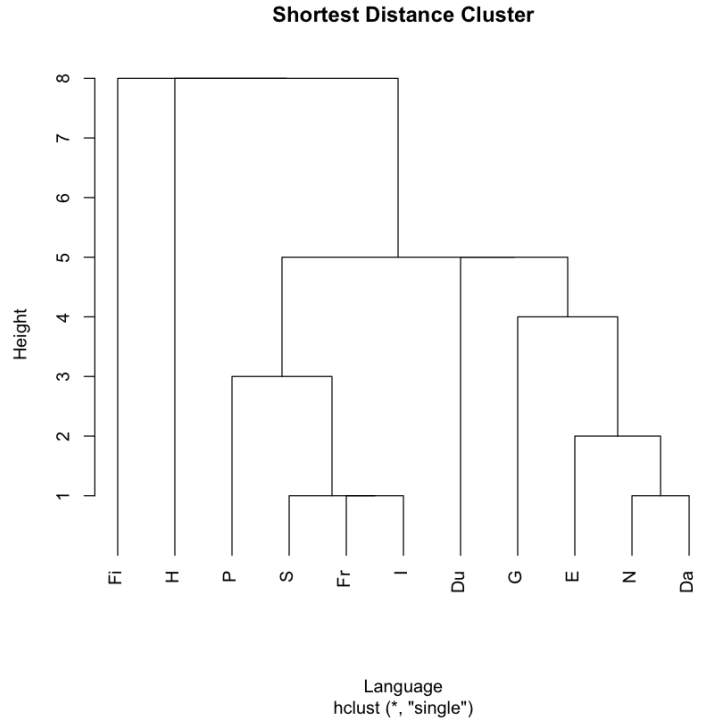
```
    E  N Da Du  G Fr  S  I  P  H
N   2
Da  2  1
Du  7  5  6
G   6  4  5  5
Fr  6  6  6  9  7
S   6  6  5  9  7  2
I   6  6  5  9  7  1  1
P   7  7  6 10  8  5  3  4
H   9  8  8  8  9 10 10 10 10
Fi  9  9  9  9  9  9  9  9  9  8
```

7.1 (1) 最短距离法

[52]:
```
h1 <- hclust(d, "single")
```

画出谱系图:

[67]:
```
plot(h1, hang=-1, main="Shortest Distance Cluster", xlab="Language")    #␣
↪hang=-1 表示线画到底
```

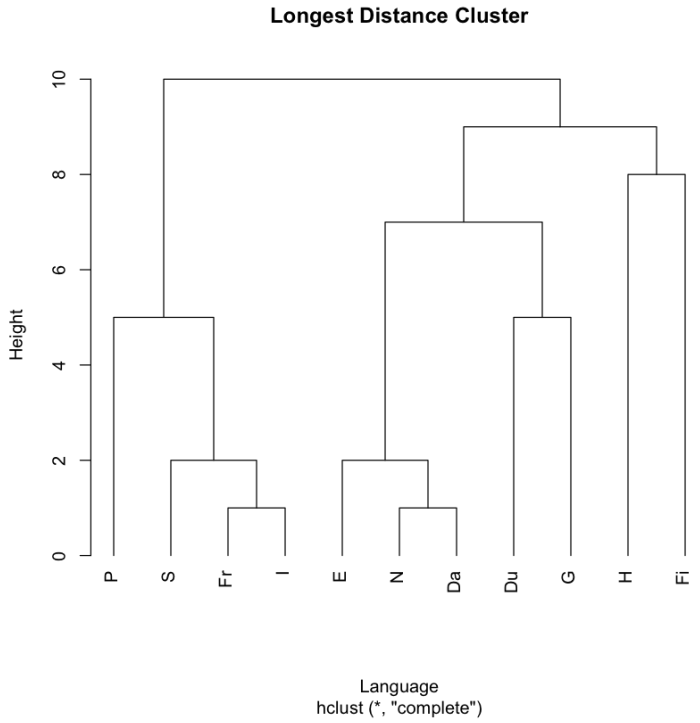


聚类过程:

首先在距离为 1 的时候，将挪威语和丹麦语聚为一类，得新类 CL10={丹麦语，挪威语}，其中包含 2 个样本，这是全部类被分为类；然后，将法语和意大利语聚为一类，CL9={法语，意大利语}；其中包含两个样本，这是全部样本被分为 9 类; 接着在最短距离为 2 的时候，波兰语被分到 CL9 当中，也即 CL8={CL9, 波兰语}，然后英语被分到 CL10 中，的新类 CL7={CL10, 英语}={丹麦语，挪威语，英语}，$\cdots$ 如此继续下去，直到距离为 8 的时候，所有语言合为一类，聚类结束。

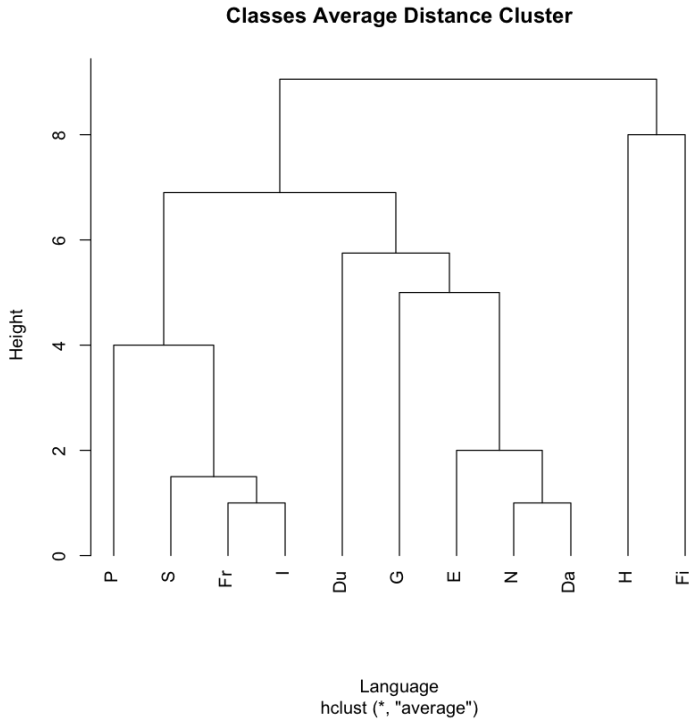
7.2 (2) 最长距离法

[68]:
```
h2 <- hclust(d, "complete")
plot(h2, hang=-1, main="Longest Distance Cluster", xlab="Language")
```



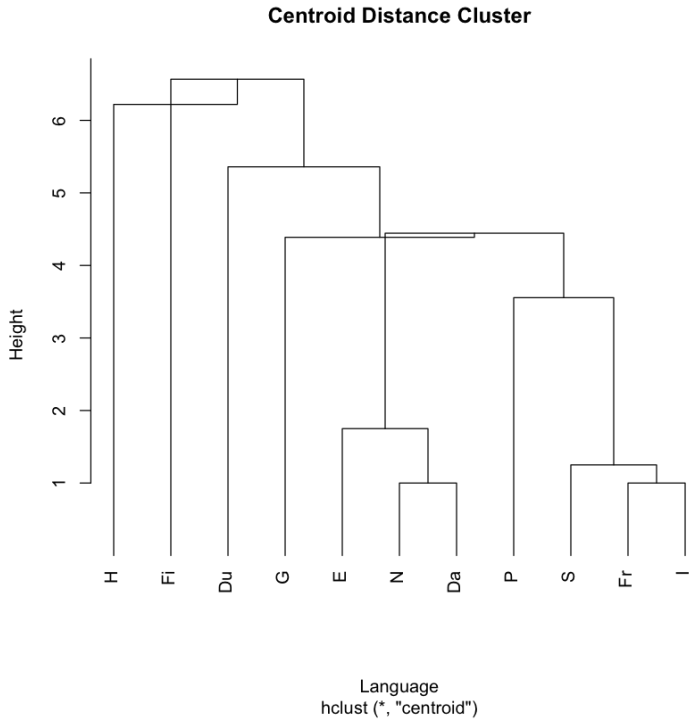
7.3 (3) 类平均距离法

[69]:
```
h3 <- hclust(d, "average")
plot(h3, hang=-1, main="Classes Average Distance Cluster", xlab="Language")
```



7.4 (4) 重心距离法

[70]:
```
h4 <- hclust(d, "centroid")
plot(h4, hang=-1, main="Centroid Distance Cluster", xlab="Language")
```



8 习题 6.7

6.7 考察 1985 年至 2000 年全国如下各价格指数：

X_1：商品零售价格指数；　　X_2：居民消费价格指数；

X_3：城市居民消费价格指数；　　X_4：农村居民消费价格指数；

X_5：农产品收购价格指数；　　X_6：农村工业品零售价格指数；

按年份进行如下谱系聚类分析，并画出谱系图：

(1) 最长距离法，给出聚为 3 类的结果；

(2) 类平均距离法，给出聚为 3 类的结果.

(3) 将数据标准化，再按上述方法聚类，情况又如何？

[1]:
```
data <- read.csv("ex_6_7.csv", row.names=1); data
```

A data.frame: 16 × 6

	x1 <dbl>	x2 <dbl>	x3 <dbl>	x4 <dbl>	x5 <dbl>	x6 <dbl>
1985	128.1	100.0	134.2	100.0	166.8	111.1
1986	135.8	106.5	143.6	106.1	177.5	114.7
⋮						
1999	359.8	329.7	472.8	314.3	424.3	280.5
2000	354.4	331.0	476.6	314.0	409.0	277.1

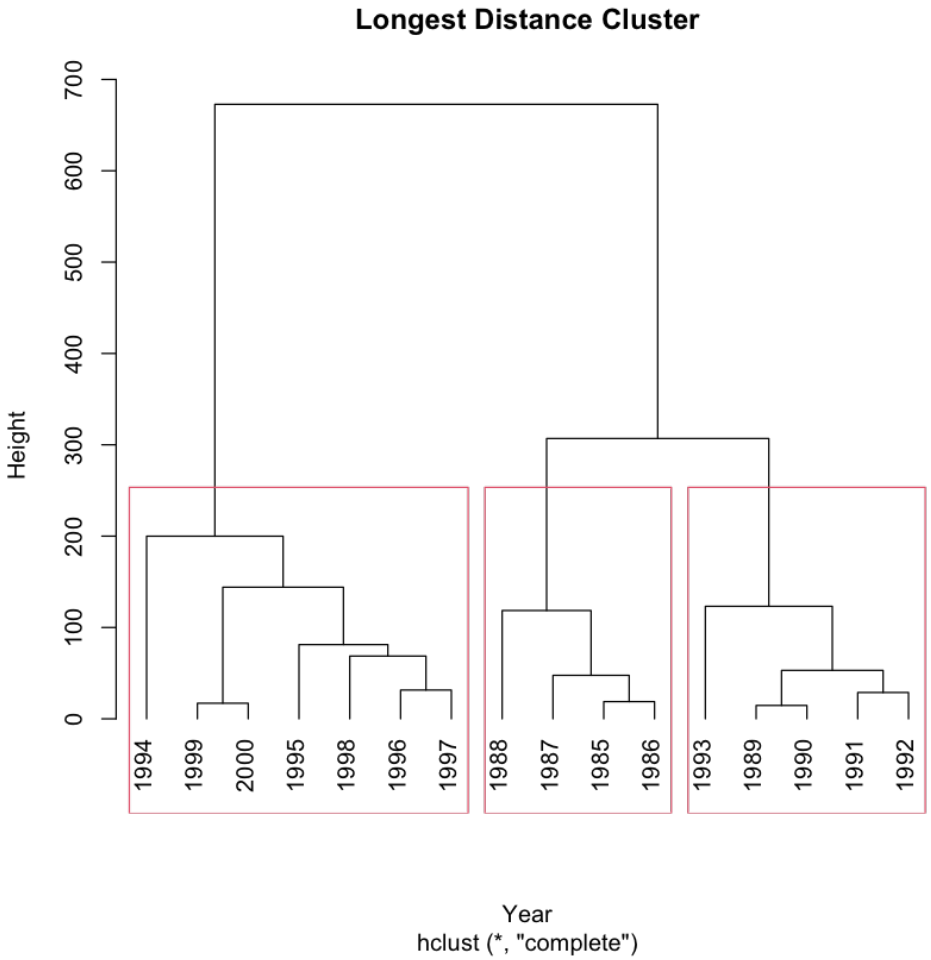
计算得到距离：

[2]:
```
d <- dist(data)
```

将距离代入 hclust 函数即可完成聚类。

8.1 (1) 最长距离法

[3]:
```
h1 <- hclust(d, method="complete")


plot(h1, hang=-1, main="Longest Distance Cluster", xlab="Year")     # hang=-1 表示线画到底
rect.hclust(h1, k=3)     # 画出分为 3 类的矩形框
```



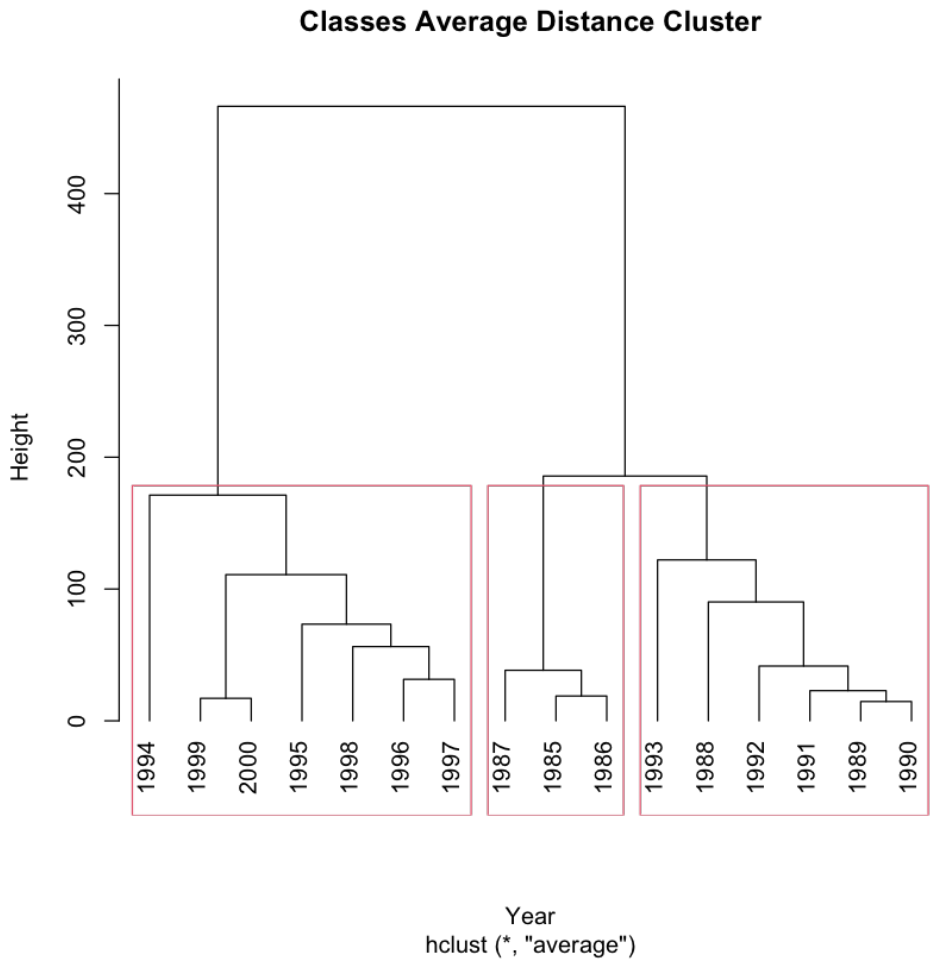
可以看出聚成三类的结果是：

- 第一类：1994，1999，2000，1995，1998，1996，1997

- 第二类：1988，1987，1985，1986
- 第三类：1993，1989，1990，1991，1992

8.2 (2) 类平均距离法

[5]:
```
h2 <- hclust(d, method="average")
plot(h2, hang=-1, main="Classes Average Distance Cluster", xlab="Year")
rect.hclust(h2, k=3)    # 画出分为 3 类的矩形框
```



聚成三类的结果是：

- 第一类：1994，1999，2000，1995，1998，1996，1997
- 第二类：1987，1985，1986
- 第三类：1993，1988，1992，1991，1989，1990

8.3 (3) 数据标准化

8.3.1 数据标准化

先用 scale 将数据标准化：

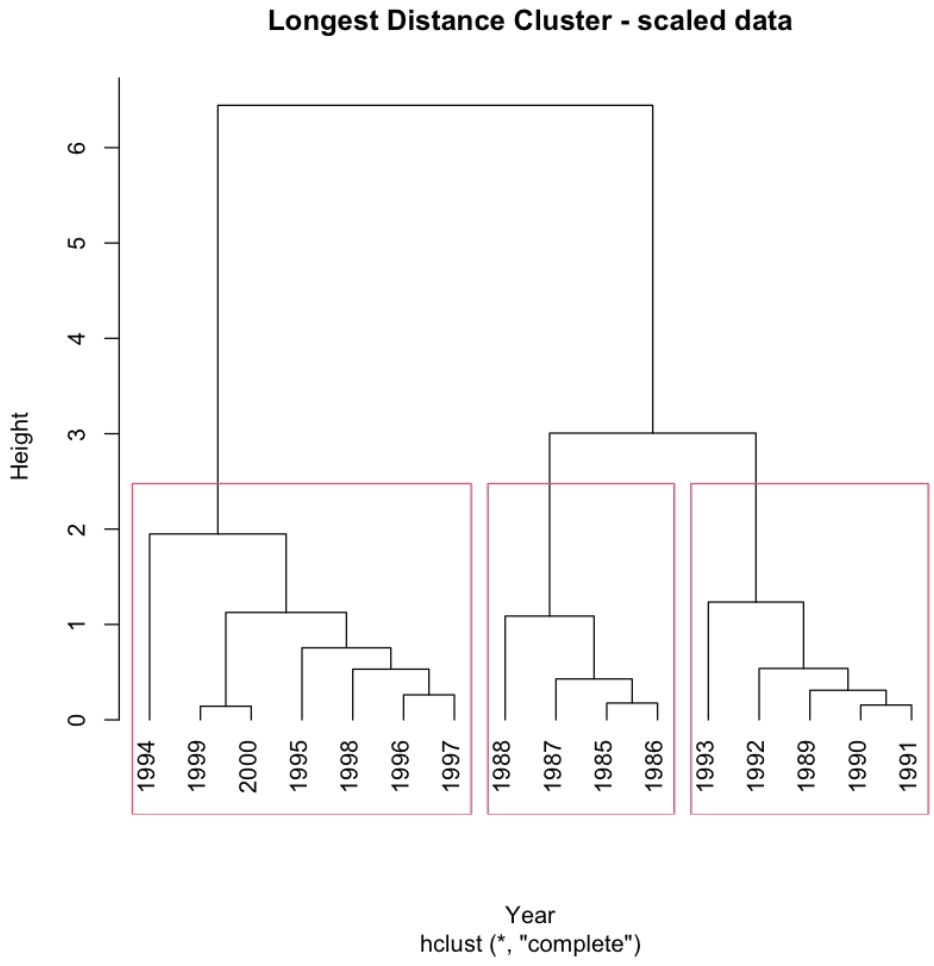
[7]:
```
data.scale <- scale(data); data.scale
```

重新计算标准化数据的距离：

[8]:
```
ds <- dist(data.scale)
```

8.3.2 最长距离法

[9]:
```
h3 <- hclust(ds, method="complete")
plot(h3, hang=-1, main="Longest Distance Cluster - scaled data", xlab="Year")
rect.hclust(h3, k=3)
```

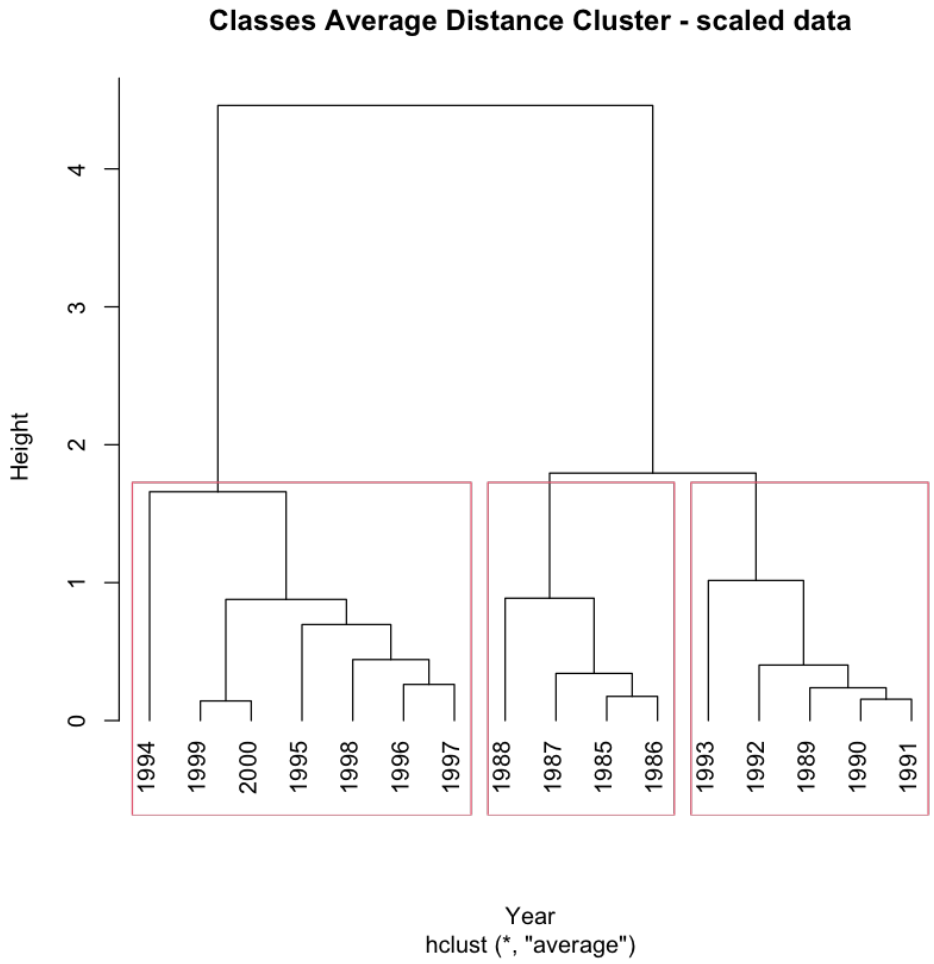


聚成三类的结果：

- 第一类：1994，1999，2000，1995，1998，1996，1997
- 第二类：1988，1987，1985，1986
- 第三类：1993，1992，1989，1990，1991

8.3.3 类平均距离法

[11]:
```
h4 <- hclust(ds, method="average")

plot(h4, hang=-1, main="Classes Average Distance Cluster - scaled data",␣
 ↪xlab="Year")
rect.hclust(h4, k=3)
```



聚成三类的结果：

- 第一类：1994，1999，2000，1995，1998，1996，1997
- 第二类：1988，1987，1985，1986

- 第三类：1993，1992，1989，1990，1991

可以看到，在数据标准化之前不同聚类方法得到的结果不仅尽相同，而且在标准化前后聚类结果也是不要一样的，但是在数据标准化之后，两种不同的聚类方法得到了相同的结果。[9]

参考文献

[1] 梅长林范金城. 数据分析方法 [M]. 高等教育出版社, 2006.

[2] R-project. An Introduction to R [M/OL]. https://cran.r-project.org/doc/manuals/r-release/R-intro.html

[3] Tiaaaaa. R 语言典型相关分析 [OL]. https://blog.csdn.net/Tiaaaaa/article/details/58137522

[4] 佚名. CCA 典型相关分析，计算向量之间的相关系数 [OL]. https://rstudio-pubs-static.s3.amazonaws.com/553282_c1046a4c0b1a40ac9319f51b6207a9d7.html

[5] DoubleHelix. R 语言数据分析与挖掘 (第八章): 判别分析 (3)——费歇尔 (Fisher) 判别分析 [OL]. https://cloud.tencent.com/developer/article/1553504

[6] Jason. Learn R | 多元统计之判别分析（下）[OL]. https://zhuanlan.zhihu.com/p/23965433

[7] Roger D. Peng. Exploratory Data Analysis with R: Chapter 10 Plotting and Color in R [M/OL]. https://bookdown.org/rdpeng/exdata/plotting-and-color-in-r.html

[8] 将子无怨. R 语言进阶之聚类分析 [OL]. https://zhuanlan.zhihu.com/p/140534259

[9] 毕业零距离. 聚类分析原理及 R 语言实现过程 [OL]. https://www.jianshu.com/p/50cb85285af0