数据分析实验
说明文档

基于 R 语言实现

https://github.com/j1o2h3n/Data-Analysis

目　录

1 数据集介绍 1

2 数据预分析处理 3

3 回归分析 5

3.1 问题描述与目标 5

3.2 实验步骤 5

3.3 结果与分析 6

4 主成分分析 10

4.1 问题描述与目标 10

4.2 实验步骤 10

4.3 结果与分析 11

5 因子分析 15

5.1 问题描述与目标 15

5.2 实验步骤 15

5.3 结果与分析 16

6 聚类分析 19

6.1 问题描述与目标 19

6.2 实验步骤 19

6.3 结果与分析 20

结语.....23

参考文献 23

附录.....24

1 数据集介绍

某些高浓度的有害藻类对河流生态环境的强大破坏是一个严重的问题，他们不仅仅破坏河流里的生物，也严重破坏河流水质。能够监测并在早期对海藻的繁殖进行预测对提高河流质量是非常有必要的。

针对此类问题的研究，有来自 ERUDIT 的研究数据集，并被用于 1999 年的 COIL 国际数据分析竞赛，可以从 UCI 机器学习数据库获得此数据集（http://archive.ics.uci.edu/ml/datasets/Coil+1999+Competition+Data）。该数据集是在大约一年时间里，在不同时间内对欧洲不同河流的站点进行水质样本取样。分析了这些样品中的各种化学物质，包括：硝酸盐，亚硝酸盐和氨形式的氮，磷酸盐，pH，氧气，氯化物。同时，收集藻类样品以确定藻类种群分布。对于每个水样，测定了它们的不同化学性质以及 7 种有害藻类的存在频率，同时测定记录了一些其它特性，例如收集水样的季节、河流大小和水流的速度。

本数据集总共采集了 340 个水样，其中分为两个数据集，第一个数据集有 200 个水样的完整数据，作为训练数据集，第二个数据集有 140 个水样的部分数据，作为测试数据集。该数据集的每一条水样记录是同一条河流在该年的同一个季节的三个月内收集的水样的平均值。

第一个数据集每条水样记录由 11 个变量构成。其中 3 个变量是名义变量，它们分别描述水样收集的季节、收集样品的河流大小和河水速度。余下的 8 个变量是所观测水样的不同化学参数，即

- 最大 pH 值
- 最小含氧量（O_2）
- 平均氯化物含量（Cl）
- 平均硝酸盐含量（NO_3^-）
- 平均氨含量（NH_4^+）
- 平均正磷酸盐含量（PO_4^{3-}）
- 平均磷酸盐含量（PO_4）
- 平均叶绿素含量

与这些参数相关的是 7 种不同有害藻类在相应水样中的频率数目。数据集并未提供所观察藻类的名称的有关信息。

第二个数据集由 140 个额外观测值构成。它们的基本结构和第一个数据集一样，但是它不包含 7 种藻类的频率数目。

数据集中值 0.0 表示频率非常低，数据集还包含一些空白字段，这些字段被

标记为与字符串 XXXXX。

这些文件的每一行代表个观测值。在数据集中，每一行的变量值之间由空格来分隔。数据集中值为 0.0 的表示频率非常低，数据集中还包含一些缺失值，缺失值由字符串“XXXXXXX”来表示。

获得的数据集部分内容如图 1-1 所示。

winter	small	medium	8.00000	9.80000	60.80000	6.23800	578.00000	105.00000	170.00000	50.00000	0.00000	0.00000	0.00000	0.00000	34.20000	8.30000	0.00000
spring	small	medium	8.35000	8.00000	57.75000	1.28800	370.00000	428.75000	558.75000	1.30000	1.40000	7.60000	4.80000	1.90000	6.70000	0.00000	2.10000
autumn	small	medium	8.10000	11.40000	40.02000	5.33000	346.66699	125.66700	187.05701	15.60000	3.30000	53.60000	1.90000	0.00000	0.00000	0.00000	9.70000
spring	small	medium	8.07000	4.80000	77.36400	2.30200	98.18200	61.18200	138.70000	1.40000	3.10000	41.00000	18.90000	0.00000	1.40000	0.00000	1.40000
autumn	small	medium	8.06000	9.00000	55.35000	10.41600	233.70000	58.22200	97.58000	10.50000	9.20000	2.90000	7.50000	0.00000	7.50000	4.10000	1.00000
winter	small	high	8.25000	13.10000	65.75000	9.24800	430.00000	18.25000	56.66700	28.40000	15.10000	14.60000	1.40000	0.00000	22.50000	12.60000	2.90000
summer	small	high	8.15000	10.30000	73.25000	1.59500	110.00000	61.25000	111.75000	3.20000	2.40000	1.20000	3.20000	3.90000	5.80000	6.80000	0.00000
autumn	small	high	8.05000	10.60000	59.06700	4.99000	205.66701	44.66700	77.43400	6.90000	18.20000	1.60000	0.00000	0.00000	5.50000	8.70000	0.00000
winter	small	medium	8.70000	3.40000	21.95000	0.88600	102.75000	36.30000	71.00000	5.54400	25.40000	5.40000	2.50000	0.00000	0.00000	0.00000	0.00000
winter	small	high	7.93000	9.90000	8.00000	1.39000	5.80000	27.25000	46.60000	0.80000	17.00000	0.00000	0.00000	2.90000	0.00000	0.00000	1.70000
spring	small	high	7.70000	10.20000	8.00000	1.52700	21.57100	12.75000	20.75000	0.80000	16.60000	0.00000	0.00000	0.00000	1.20000	0.00000	6.00000
summer	small	high	7.45000	11.70000	8.69000	1.58800	18.42900	10.66700	19.00000	0.60000	32.10000	0.00000	0.00000	0.00000	0.00000	0.00000	1.50000
winter	small	high	7.74000	9.60000	5.00000	1.22300	27.28600	12.00000	17.00000	41.00000	43.50000	0.00000	2.10000	0.00000	1.20000	0.00000	2.10000
summer	small	high	7.72000	11.80000	6.30000	1.47000	8.00000	16.00000	15.00000	0.50000	31.10000	1.00000	3.40000	0.00000	1.90000	0.00000	4.10000
winter	small	high	7.90000	9.60000	3.00000	1.44800	46.20000	13.00000	61.60000	0.30000	52.20000	5.00000	7.80000	0.00000	4.00000	0.00000	0.00000
autumn	small	high	7.55000	11.50000	4.70000	1.32000	14.75000	4.25000	98.25000	1.10000	69.90000	0.00000	1.70000	0.00000	0.00000	0.00000	0.00000
winter	small	high	7.78000	12.00000	7.00000	1.42000	34.33300	18.66700	50.00000	1.10000	46.20000	0.00000	0.00000	1.20000	0.00000	0.00000	0.00000
spring	small	high	7.61000	9.80000	7.00000	1.44300	31.33300	20.00000	57.83300	0.40000	31.80000	0.00000	3.10000	4.80000	7.70000	1.40000	7.20000
summer	small	high	7.35000	10.40000	7.00000	1.71800	49.00000	41.50000	61.50000	0.80000	50.60000	0.00000	9.90000	4.30000	3.60000	8.20000	2.20000
spring	small	medium	7.79000	3.20000	64.00000	2.82200	8777.59961	564.59998	771.59998	4.50000	0.00000	0.00000	0.00000	44.60000	0.00000	0.00000	1.40000
winter	small	medium	7.83000	10.70000	88.00000	4.82500	1729.00000	467.50000	586.00000	16.00000	0.00000	0.00000	0.00000	6.80000	6.10000	0.00000	0.00000
spring	small	high	7.20000	9.20000	0.80000	0.64200	81.00000	15.60000	18.00000	0.50000	15.50000	0.00000	0.00000	2.30000	0.00000	0.00000	0.00000
autumn	small	high	7.75000	10.30000	32.92000	2.94200	42.00000	16.00000	40.00000	7.60000	23.20000	0.00000	0.00000	0.00000	27.60000	11.10000	0.00000
winter	small	high	7.62000	8.50000	11.86700	1.71500	208.33299	3.00000	27.50000	1.70000	74.20000	0.00000	0.00000	3.70000	0.00000	0.00000	0.00000
spring	small	high	7.84000	9.40000	10.97500	1.51000	12.50000	3.00000	11.50000	1.50000	13.00000	8.60000	1.20000	3.50000	1.20000	1.60000	1.90000
summer	small	high	7.77000	10.70000	12.53600	3.97600	58.50000	9.00000	44.13600	3.00000	4.10000	0.00000	0.00000	0.00000	9.20000	10.10000	0.00000
winter	small	high	7.09000	8.40000	10.50000	1.57200	28.00000	4.00000	13.60000	0.50000	29.70000	0.00000	0.00000	4.90000	0.00000	0.00000	0.00000
autumn	small	high	6.80000	11.10000	9.00000	0.63000	20.00000	4.00000	XXXXXXX	2.70000	30.30000	1.90000	0.00000	0.00000	2.10000	1.40000	2.10000
winter	small	high	8.00000	9.80000	16.00000	0.73000	20.00000	26.00000	45.00000	0.80000	17.10000	0.00000	19.60000	0.00000	0.00000	0.00000	2.50000
spring	small	high	7.20000	11.30000	9.00000	0.23000	120.00000	12.00000	19.00000	0.50000	33.90000	1.00000	14.60000	0.00000	0.00000	0.00000	0.00000
autumn	small	high	7.40000	12.50000	13.00000	3.33000	60.00000	72.00000	142.00000	4.90000	3.40000	16.00000	1.20000	0.00000	15.30000	15.80000	0.00000
winter	small	high	8.10000	10.30000	26.00000	3.78000	60.00000	246.00000	304.00000	2.80000	6.90000	17.10000	20.20000	0.00000	4.00000	0.00000	2.90000
summer	small	high	7.80000	11.30000	20.08200	3.02000	49.50000	53.00000	130.75000	5.80000	0.00000	8.00000	1.90000	0.00000	11.20000	42.70000	1.20000
autumn	small	medium	8.40000	9.90000	34.50000	2.81800	3515.00000	20.00000	47.00000	2.30000	13.60000	9.10000	0.00000	0.00000	1.40000	0.00000	0.00000
winter	small	medium	8.27000	7.80000	29.20000	0.05000	6400.00000	7.40000	23.00000	0.90000	5.30000	40.70000	3.30000	0.00000	0.00000	0.00000	1.90000
summer	small	medium	8.66000	8.40000	30.52300	3.44400	1911.00000	58.87500	84.46000	3.60000	18.30000	12.40000	1.00000	0.00000	0.00000	0.00000	1.00000
winter	small	high	8.30000	10.90000	1.17000	0.73500	13.50000	1.62500	3.00000	0.20000	66.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
spring	small	high	8.00000	XXXXXXX	1.45000	0.81000	10.00000	2.50000	3.00000	0.30000	75.80000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
winter	small	medium	8.30000	8.90000	20.62500	3.41400	228.75000	196.62000	253.25000	12.32000	2.00000	38.50000	4.10000	2.20000	0.00000	0.00000	10.20000
spring	small	medium	8.10000	10.50000	22.28600	4.07100	178.57001	182.42000	255.28000	8.95700	2.20000	2.70000	1.00000	3.70000	2.70000	0.00000	0.00000
winter	small	medium	8.00000	5.50000	77.00000	6.09600	122.85000	143.71001	296.00000	3.70000	0.00000	5.90000	10.60000	1.70000	0.00000	0.00000	7.10000
summer	small	medium	8.15000	7.10000	54.19000	3.82900	647.57001	59.42900	175.04601	13.20000	0.00000	0.00000	0.00000	5.70000	11.30000	17.00000	1.60000
winter	small	high	8.30000	7.70000	50.00000	8.54300	76.00000	264.89999	344.60001	22.50000	0.00000	40.90000	7.50000	0.00000	2.40000	1.50000	0.00000
spring	small	high	8.30000	8.80000	54.14300	7.83000	51.42900	276.85001	326.85699	11.84000	4.10000	3.10000	0.00000	0.00000	19.70000	17.00000	0.00000
winter	small	high	8.40000	13.40000	69.75000	4.55500	37.50000	10.00000	40.66700	3.90000	51.80000	4.10000	0.00000	0.00000	3.10000	5.50000	0.00000
spring	small	high	8.30000	12.50000	87.00000	4.87000	22.50000	27.00000	43.50000	3.30000	29.50000	1.00000	2.70000	3.20000	2.90000	9.60000	0.00000
autumn	small	high	8.00000	12.10000	66.30000	4.53500	39.00000	16.00000	29.00000	0.80000	54.40000	3.40000	1.20000	0.00000	18.70000	2.00000	0.00000
winter	small	low	XXXXXXX	12.60000	9.00000	0.23000	10.00000	5.00000	6.00000	1.10000	35.50000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
spring	small	medium	7.60000	9.60000	15.00000	3.02000	40.00000	27.00000	121.00000	2.80000	89.80000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
autumn	small	medium	7.29000	11.21000	17.75000	3.07000	35.00000	13.00000	20.81200	12.10000	24.80000	7.40000	0.00000	2.50000	10.60000	17.10000	3.20000
winter	small	medium	7.60000	10.20000	32.30000	4.50800	192.50000	12.75000	49.33300	7.90000	0.00000	0.00000	0.00000	4.60000	1.20000	0.00000	3.90000
summer	small	medium	8.00000	7.90000	27.23300	1.65100	28.33300	7.30000	22.90000	4.50000	39.10000	0.00000	1.20000	2.20000	5.40000	1.50000	3.20000

图 1-1 数据集部分内容

数据集的数据结构如下图 1-2 所示。

A				K	a			g
$CC_{1,1}$				$CC_{1,11}$	$AG_{1,1}$			$AG_{1,7}$
$CC_{200,1}$				$CC_{200,11}$	$AG_{200,1}$			$AG_{200,7}$

图 1-2 数据集的数据结构

说明：

$CC_{i,j}$：河流特征或化学浓度　　$AG_{i,j}$：藻类频率

化学参数标记为 A，...，K　　藻类的列标记为 a，...，g

2 数据预分析处理

由于我们作业要求，所以只对数据进行分析即可，不需要具体量化数据分析模型的优劣，所以我们在数据集中只使用 200 个水质样本的训练样本集，不使用另外 140 个水质样本的测试数据集。

我们首先载入数据集，对应的代码为

```
algae <- read.table('C:\\数据分析与 R 软件\\Analysis.txt',
                    header=F,  #表示要读的文件的第一行不包括变量名
                    dec='.', #指出数值使用'.'字符分隔小数
                    col.names=c('season','size','speed','mxPH','mnO2','Cl',
                    'NO3','NH4','oPO4','PO4','Chla','a1','a2','a3','a4',
                    'a5','a6','a7'), #给正在读取的变量一个名称向量
                    na.strings=c('XXXXXXX')) #表示字符串被解释为未知值
```

再使用函数 head( )显示读取数据框的前 6 行，显示结果如图 2-1 所示。

```
> head(algae) #显示数据框前六行
  season  size  speed mxPH mnO2     Cl    NO3     NH4    oPO4     PO4 Chla   a1   a2   a3  a4   a5   a6  a7
1 winter small medium 8.00  9.8 60.800  6.238 578.000 105.000 170.000 50.0  0.0  0.0  0.0 0.0 34.2  8.3 0.0
2 spring small medium 8.35  8.0 57.750  1.288 370.000 428.750 558.750  1.3  1.4  7.6  4.8 1.9  6.7  0.0 2.1
3 autumn small medium 8.10 11.4 40.020  5.330 346.667 125.667 187.057 15.6  3.3 53.6  1.9 0.0  0.0  0.0 9.7
4 spring small medium 8.07  4.8 77.364  2.302  98.182  61.182 138.700  1.4  3.1 41.0 18.9 0.0  1.4  0.0 1.4
5 autumn small medium 8.06  9.0 55.350 10.416 233.700  58.222  97.580 10.5  9.2  2.9  7.5 0.0  7.5  4.1 1.0
6 winter small   high 8.25 13.1 65.750  9.248 430.000  18.250  56.667 28.4 15.1 14.6  1.4 0.0 22.5 12.6 2.9
```

图 2-1 head( )运行结果

我们采用 summary( )函数获取数据的描述性统计特性摘要，得到如图 2-2 所示结果。

```
> summary(algae)
    season       size       speed         mxPH            mnO2             Cl              NO3        
 autumn:40   large :45   high  :84   Min.   :5.600   Min.   : 1.500   Min.   :  0.222   Min.   : 0.050  
 spring:53   medium:84   low   :33   1st Qu.:7.700   1st Qu.: 7.725   1st Qu.: 10.981   1st Qu.: 1.296  
 summer:45   small :71   medium:83   Median :8.060   Median : 9.800   Median : 32.730   Median : 2.675  
 winter:62                           Mean   :8.012   Mean   : 9.118   Mean   : 43.636   Mean   : 3.282  
                                     3rd Qu.:8.400   3rd Qu.:10.800   3rd Qu.: 57.824   3rd Qu.: 4.446  
                                     Max.   :9.700   Max.   :13.400   Max.   :391.500   Max.   :45.650  
                                     NA's   :1       NA's   :2        NA's   :10        NA's   :2       
      NH4                oPO4              PO4              Chla              a1              a2        
 Min.   :    5.00   Min.   :  1.00   Min.   :  1.00   Min.   :  0.200   Min.   : 0.00   Min.   : 0.000  
 1st Qu.:   38.33   1st Qu.: 15.70   1st Qu.: 41.38   1st Qu.:  2.000   1st Qu.: 1.50   1st Qu.: 0.000  
 Median :  103.17   Median : 40.15   Median :103.29   Median :  5.475   Median : 6.95   Median : 3.000  
 Mean   :  501.30   Mean   : 73.59   Mean   :137.88   Mean   : 13.971   Mean   :16.92   Mean   : 7.458  
 3rd Qu.:  226.95   3rd Qu.: 99.33   3rd Qu.:213.75   3rd Qu.: 18.308   3rd Qu.:24.80   3rd Qu.:11.375  
 Max.   :24064.00   Max.   :564.60   Max.   :771.60   Max.   :110.456   Max.   :89.80   Max.   :72.600  
 NA's   :2          NA's   :2        NA's   :2        NA's   :12                                        
       a3              a4               a5               a6              a7        
 Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000  
 1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000  
 Median : 1.550   Median : 0.000   Median : 1.900   Median : 0.000   Median : 1.000  
 Mean   : 4.309   Mean   : 1.992   Mean   : 5.064   Mean   : 5.964   Mean   : 2.495  
 3rd Qu.: 4.925   3rd Qu.: 2.400   3rd Qu.: 7.500   3rd Qu.: 6.925   3rd Qu.: 2.400  
 Max.   :42.800   Max.   :44.600   Max.   :44.400   Max.   :77.600   Max.   :31.600  
```

图 2-2 summary( )运行结果

由结果我们可以得到数据中的统计特性概览，为我们给出均值、中位数、四

分位数以及极值等一系列统计信息，提供了变量值分布的初步信息。其中结果的 NA's 后面的数值表示缺失值的个数，我们由上面结果可以通过观察中位数和均值之间的差异以及四分位距，可以了解到数据大致的偏度和分散情况。

在数据集的原始数据中的存在部分的数据缺失，对此会导致我们后面数据的分析方法无法应用，为此我们需要对含有缺失值的数据进行处理。对此我们采用数据集个案的相似性方法来填补缺失值，我们尝试使用行（观察值）之间的相似性来填补缺失值。

同时我们可以注意到，在数据集中存在部分严重缺失数据的水样记录，为此我们使用 manyNAs( )进行检索缺失率大于 20%的样本，得到如下图 2-3 的结果。

```
> manyNAs(algae, 0.2)
[1]  62 199
```

图 2-3 manyNAs( )运行结果

我们可以得到第 62 条和第 199 条的水样记录，在数据集中观察到这两个水样记录在其中 11 个预测值中存在 6 个缺失值，对此类数据我们进行填补不具有太大意义，所以我们需要在处理前剔除这两个含有太多 NA 值的样本。

采用行之间的相似性来填补缺失值，我们所描述的方法假设是如果两个水样是相似的，其中一个水样在某些变量上有缺失值，那么该缺失值很可能与另一个水样的值是相似的。对此我们采用 knnImputation( )函数来完成，其效果是使用 KNN（K 最近邻算法）进行插补，其中需要输入参数 k 让函数用一个欧氏距离的变种来寻找距离个案最近的 k 个近邻样本进行插值。对此我们采用如下代码首先进行数据的剔除，再进行数据填补。

```
library(DMwR) #加载函数库
data(algae) #重新读取数据
algae <- algae[-manyNAs(algae), ] #数据缺失率大于 20%的样本剔除
clean.algae <- knnImputation(algae, k = 10) #进行数据填补
```

通过上述这些操作，数据集中不再含有缺失值 NA 了，为进行后续分析做好了充分的准备。

3 回归分析

3.1 问题描述与目标

回归分析是应用非常广泛的数据分析方法，主要是研究变量之间的相关关系，并寻求变量之间的近似函数关系。而多元线性回归分析，是寻求找到关于一个目标变量和一组解释变量关系的线性函数。

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_p X_{ip} + \varepsilon_i, \qquad i = 1, \ldots, n$$

多元线性回归模型如上式所示，其中 Y 为观测向量，X 为结构矩阵，β 是未知参数向量，ε 为未知向量。

简单来说，首先我们应该去除 11 个变量中的前三个变量，因为这三个是名义变量，是描述水样收集的季节、收集样品的河流大小和河水速度的变量，虽然 R 语言中有处理名义变量的方法，而将它们引入模型变量会复杂我们的模型，模型也会更加精准，但是任务也变得复杂。所以为了方便分析，我们采用的变量是河流水样的不同化学参数。即我们的任务就是需要使用每条水样记录后面 8 个变量，来线性拟合分析 7 种藻类中的某一种藻类的频率数目，建立出对应的多元线性回归模型，并进行分析模型。实际上我们需要对 7 种藻类建立不同的多元线性回归模型，对此我们仅对第一种藻类进行回归分析，其它的回归分析与此情况类似。

3.2 实验步骤

我们输入以下程序代码：

```
lm.a1 <- lm(a1 ~ .,data=clean.algae[,4:12]) #建立多元线性回归模型
formula(lm.a1) #提取模型公式
summary(lm.a1) #提取模型资料，显示模型拟合结果
anova(lm.a1) #计算方差分析表
plot(lm.a1) #制作模型诊断图
```

函数 lm( )建立一个线性回归模型。其中的第一个参数给出了模型的函数形式，第一个参数中的点子“.”代表数据框中的所有除 al 外的变量，这里我们选择 a1 作为因变量，其它作为自变量。参数 data 是用来设定建模所用的数据集，这里我们选定了数据框中的第 4 行至第 12 行的数据作为输入数据框。

函数 formula( )提取模型的公式，显著的体现回归模型中哪些变量是因变量以及自变量。

函数 summary( )用于提取模型资料，显示模型拟合结果，可以给出建立模型的一些诊断信息。可以给出模型的残差统计信息，回归系数及回归系数的显著性检验结果，以及给出残差标准差、决定系数及回归方程的显著性检验结果。

函数 anova( )用于得出计算方差分析表。

函数 plot( )用于绘制模型的诊断图，进行模型的回归诊断。

3.3 结果与分析

如下图 3-1 是 formula( )函数运行后的结果，它显示在我们的模型中，是将 a1 作为模型的因变量，其中 mxPH、mnO2、Cl、NO3、NH4、oPO4、PO4、Chla 八个变量为模型的自变量。

```
> lm.a1 <- lm(a1 ~ .,data=clean.algae[,4:12]) #建立多元线性回归模型
> formula(lm.a1) #提取模型公式
a1 ~ mxPH + mnO2 + Cl + NO3 + NH4 + oPO4 + PO4 + Chla
```

图 3-1 formula( )运行结果

```
> summary(lm.a1) #提取模型资料，显示模型拟合结果

Call:
lm(formula = a1 ~ ., data = clean.algae[, 4:12])

Residuals:
    Min      1Q  Median      3Q     Max
-32.098 -11.921  -2.442   7.214  67.291

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 72.7093211 20.6632776   3.519 0.000543 ***
mxPH        -6.0829108  2.4529483  -2.480 0.014019 *
mnO2         0.9009543  0.6473648   1.392 0.165641
Cl          -0.0493176  0.0328159  -1.503 0.134545
NO3         -1.6774676  0.5410732  -3.100 0.002229 **
NH4          0.0016898  0.0009968   1.695 0.091663 .
oPO4         0.0049463  0.0368351   0.134 0.893321
PO4         -0.0562290  0.0282042  -1.994 0.047629 *
Chla        -0.0752610  0.0753200  -0.999 0.318967
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 17.73 on 189 degrees of freedom
Multiple R-squared:  0.3431,    Adjusted R-squared:  0.3153
F-statistic: 12.34 on 8 and 189 DF,  p-value: 3.662e-14
```

图 3-2 summary( )运行结果

上面的图 3-2 显示的是 summary( )函数的运行结果，其结果分为三部分。

第一部分是残差的统计信息，如下图 3-3 所示。依次得到残差数据的最小值：-32.098、下四分位数：-11.921、中位数：-2.442、上四分位数：7.214 及最大值：67.291。我们可以看得出数据残差值值波动较大，我们线性回归拟合情况不是特别优秀。

```
Residuals:
    Min      1Q  Median      3Q     Max
-32.098 -11.921  -2.442   7.214  67.291
```

图 3-3 残差统计信息

第二部分是回归系数及回归系数的显著性检验，如下图 3-4 所示。各列依次代表的是各系数的估计值，标准误差，检验统计量 t 值和检验 p 值。其中最后一列标记着显著性检验的标记，其中在最后一行代表着它们对应的显著性标志与相应的显著性水平。我们可以看出回归系数里面部分系数趋近于 0，例如 NH4 和 oPO4 的系数，说明在回归模型中他们的贡献较小，对最后结果的预测影响范围有限。而在 p 值检验中我们可以看到输出结果中的截距的显著性水平非常显著，其次 mxPH、NO3 以及 PO4 的系数显著性检验结果是比较显著的。

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 72.7093211 20.6632776   3.519 0.000543 ***
mxPH        -6.0829108  2.4529483  -2.480 0.014019 *
mnO2         0.9009543  0.6473648   1.392 0.165641
Cl          -0.0493176  0.0328159  -1.503 0.134545
NO3         -1.6774676  0.5410732  -3.100 0.002229 **
NH4          0.0016898  0.0009968   1.695 0.091663 .
oPO4         0.0049463  0.0368351   0.134 0.893321
PO4         -0.0562290  0.0282042  -1.994 0.047629 *
Chla        -0.0752610  0.0753200  -0.999 0.318967
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

图 3-4 回归系数及显著性检验

第三部分是给出数据的残差标准差、决定系数及回归方程的显著性检验结果，如图 3-5 所示。我们可以知道残差的标准差结果在 189 个自由度中其值为 17.73，其中决定系数 R^2 的值为 0.3431，调整后的决定系数 R^2 的值为 0.3153，F 统计量为 12.34，P 值检验值为 3.662e-14，可以得到回归方程是显著的。

```
Residual standard error: 17.73 on 189 degrees of freedom
Multiple R-squared:  0.3431,    Adjusted R-squared:  0.3153
F-statistic: 12.34 on 8 and 189 DF,  p-value: 3.662e-14
```

图 3-5 summary( )其它结果

如下图 3-6 是 anova( )函数运行后的结果，它可以得出得到的回归模型的方差分析表，运行结果显示，其中每列依次是自由度、平方和、均方、F 值以及检验的 P 值，我们由图可以得到残差的自由度是 189 自由度，在显著性检验中 mxPH、mnO2、Cl、NO3 以及 oPO4 显示的检验结果为非常显著。

```
> anova(lm.a1) #计算方差分析表
Analysis of Variance Table

Response: a1
           Df Sum Sq Mean Sq F value    Pr(>F)
mxPH        1   6597  6596.7 20.9937 8.353e-06 ***
mnO2        1   5431  5430.5 17.2825 4.880e-05 ***
Cl          1   7293  7293.3 23.2107 2.971e-06 ***
NO3         1   4652  4651.6 14.8035 0.0001631 ***
NH4         1    632   631.9  2.0109 0.1578204
oPO4        1   4179  4179.0 13.2995 0.0003432 ***
PO4         1   1917  1917.0  6.1009 0.0143977 *
Chla        1    314   313.7  0.9984 0.3189673
Residuals 189  59388   314.2
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

图 3-6 anova( )运行结果

如下图 3-7 是 plot( )函数运行后的结果，其功能是绘制出模型诊断图，如图中所示，绘制出来共有(a)Residuals vs Fitted 图，(b)Normal Q-Q 图，(c)Scale-Location 图，(d)Residuals vs Leverage 图四种诊断图。

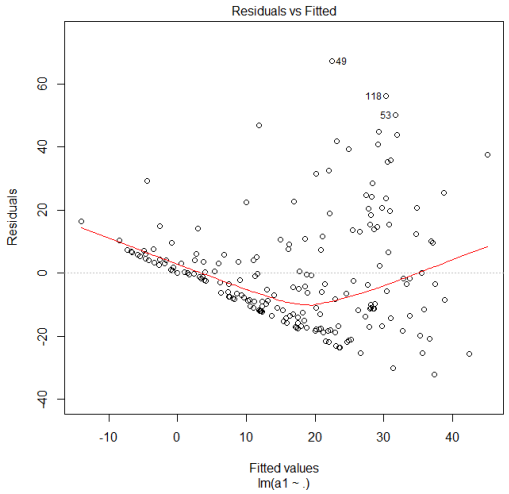
图 3-7(a)中所绘制的 Residuals vs Fitted 图是残差与真实值之间的关系图。在理想线性模型中有五大假设，其中之一便是残差应该是一个正态分布，与估计值无关。如图所示，图中的残差和估计值呈现一定关系，那么说明残差与估计值有关，说明我们得到的线性模型仍然还有许多值得改进修正的地方。

图 3-7(b)中所绘制的 Normal Q-Q 图是用来检测其残差是否是正态分布的，如果图上的点近似的在一条直线附近，可以认为样本数据来自正态分布总体，而我们得到的图中明显偏离线性拟合，故残差符合正态分布程度较低。

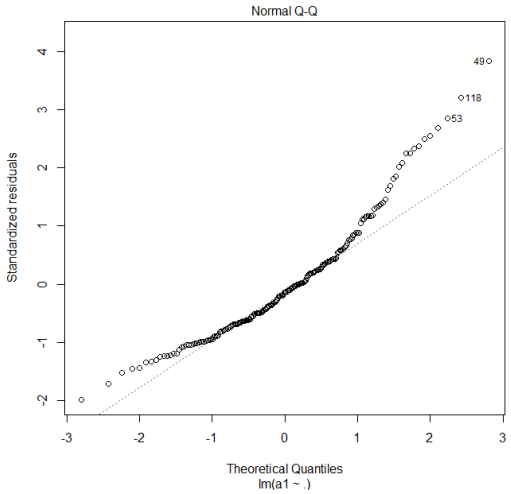
图 3-7(c)中所绘制的 Scale-Location 图是用来检查等方差假设的，一开始在假设预测的模型里的方差是一个定值，而我们需要检查方差假设是否接近定值。如图所示得到我们的方差是一个基本确定的情况，没有呈现太明显的趋势，所以我们的方差可以视作定值符合假设。

图 3-7(d)中所绘制的 Residuals vs Leverage 图是用来检查于检查数据分析项目中是否有特别极端的点，如图所示我们看到对数据而言，我们没有太多特别极

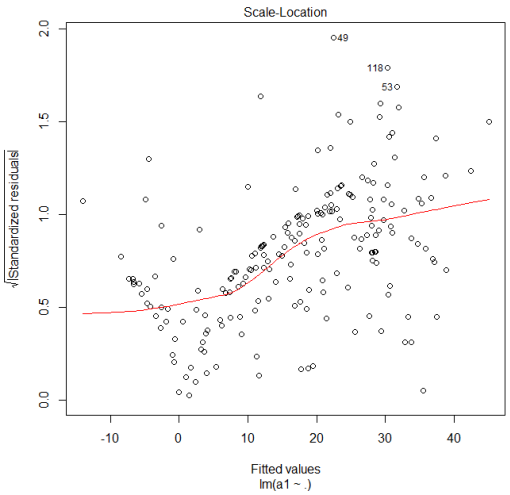
端的数据点，整体数据较为集中，基本符合数据分析的要求。



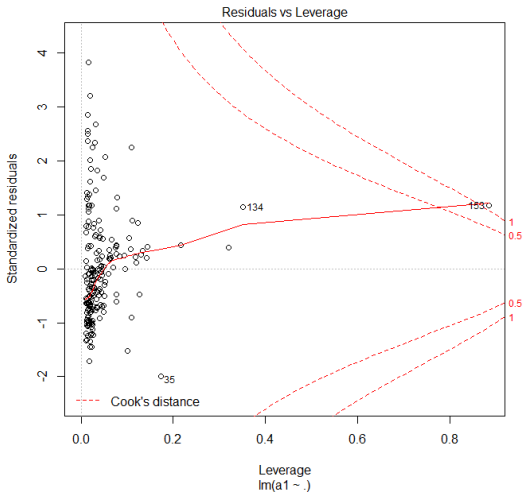
(a) Residuals vs Fitted 图



(b) Normal Q-Q 图



(c) Scale-Location 图



(d) Residuals vs Leverage 图

图 3-7 plot( )运行结果

4 主成分分析

4.1 问题描述与目标

在实际问题中，往往回设计众多变量，当变量太多不但会增加计算复杂性，也给合理地分析问题和解决问题带来困难。一般情况下，变量间会有一定的相关性，提供的信息在一定程度上有所重叠。所以我们希望使用数目较少的互不相关的变量反映原变量提供的绝大部分信息，主成分分析通过降维技术把多个变量化为少数几个主成分。主成分分析就是这种减少数据的维数同时保持数据中的对方差贡献最大的特征的一种方法。

$$\mathbf{Y} = \mathbf{W_L}^{\top}\mathbf{X} = \mathbf{\Sigma_L}\mathbf{V}^{\top}$$

主成分定义如上式所示，其中 Y 是主成分，X 是多维随机变量去均值的数据，X 通过 W_L 映射到只含有 L 个向量的低维空间中去，其中 Σ_L 是随机变量的协方差矩阵部分，V 是对应协方差矩阵的特征向量矩阵。

同上节回归分析的需求一样，我们只需要前面 11 个变量中的后面 8 个变量，即舍弃前三个名义变量，留下水样记录中的表示不同化学参数的八个变量。我们的任务即精简这八个变量，对这八个变量进行主成分分析，去除冗余的变量信息，提取出主成分。

4.2 实验步骤

我们输入以下程序代码：

```
std.x <- scale(clean.algae[4:11]) #数据标准化
prin1 <- princomp(std.x,cor=TRUE) #从相关阵 R 出发做主成分分析
summary(prin1) #列出主成分分析主要结果
loadings(prin1) #各主成分对应的系数，即相关矩阵 R 的单位正交化的特征向量
screeplot(prin1,type="lines") #画出主成分的碎石图
biplot(prin1) #画数据关于前两个主成分的散点图和原坐标在主成分下的方向


pre <- predict(prin1) #预测各个样本的主成分值
cor(std.x)
y <- eigen(cor(std.x)) #求 cor(std.x)的特征值和特征向量
```

```
y1 <- y$values[1] #第一个特征值赋给 y1
y2 <- y$values[2]
y3 <- y$values[3]
y4 <- y$values[4]
y5 <- y$values[5]
y6 <- y$values[6]
scores <- (y1*pre[,1]+y2*pre[,2]+y3*pre[,3]+y4*pre[,4]+y5*pre[,5]+y6*pre[,6])/
          (y1+y2+y3+y4+y5+y6)
#计算每个样本的综合得分
scores #显示得分情况
```

函数 scale( )是对数据进行中心化和标准化。

函数 princomp( )功能是从相关阵 R 出发对输入数据做主成分分析。

函数 summary( )功能是列出主成分分析的主要结果，如标准差、方差贡献率以及方差累积贡献率等。

函数 loadings( )主要是显示各主成分对应的系数，即相关矩阵 R 的单位正交化的特征向量，也即是载荷系数，其中空缺部分代表 0。

函数 screeplot( )功能是画出主成分的碎石图，进行分析，可视化的显示各主成分的占比，并选出其中前几的主成分。

函数 biplot( )是画数据关于前两个主成分的散点图和原坐标在主成分下的方向，便于分析数据。

函数 predict( )用来预测各个样本的主成分值。

函数 cor( )用来计算列与列间的相关系数，得到的是相关系数矩阵。

函数 eigen( )是求取输入矩阵的的特征值和特征向量。

4.3 结果与分析

如下图 4-1 是 summary( )函数运行后的结果，它我们主成分分析后的主要内容，其中第一行 Standard deviation 代表主成分的标准差，也就是特征值的开方，第二行的 Proportion of Variance 代表方差贡献率，第三行的 Cumulative Proportion 代表方差累积贡献率。由图我们科研看出前 6 个主成分的累积方差贡献率是

0.96494144，即前六个主成分方差的累积贡献率达到 90%以上。

```
> summary(prinl) #列出主成分分析主要结果
Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7    Comp.8
Standard deviation     1.6940724 1.3083338 1.0908996 0.86313124 0.80216038 0.74792663 0.46207972 0.258748477
Proportion of Variance 0.3587352 0.2139672 0.1487578 0.09312444 0.08043266 0.06992428 0.02668971 0.008368847
Cumulative Proportion  0.3587352 0.5727023 0.7214601 0.81458451 0.89501716 0.96494144 0.99163115 1.000000000
```

图 4-1 summary( )运行结果

下图 4-2 是 loadings 函数的运行结果图，显示的是个主成分的系数，即相关矩阵 R 的单位正交化的特征值，也叫做载荷矩阵。下面第二部分还显示了与主成分相关联的特征值和各主成分对整个变量的解释程度。SS loadings 行包含了与主成分相关联的特征值，下列数据表明每个相关联的特征值都是 1，Proportion Var 代表每个主成分对整个变量的解释程度，第每个主成分都解释了整个变量的 12.5%的方差，Cumulative Var 代表前面主成分对整个变量的解释程度累积和。

```
> loadings(prinl) #各主成分对应的系数，即相关矩阵R的单位正交化的特征向量

Loadings:
     Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
mxPH  0.195  0.367  0.562         0.147  0.690  0.100
mnO2 -0.352 -0.242  0.199  0.455  0.714        -0.218
Cl    0.376                0.805 -0.378        -0.245
NO3   0.210 -0.643  0.198  0.141                0.690
NH4   0.236 -0.606  0.114 -0.283         0.291 -0.619
oPO4  0.500        -0.301         0.458                -0.663
PO4   0.534        -0.212         0.322 -0.134         0.728
Chla  0.249  0.113  0.675 -0.181        -0.641 -0.119 -0.110

               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
```

图 4-2 loadings( )运行结果

由于前六个主成分方差的累积贡献率达到 90%，则由上图 4-2 的系数能够写出主成分，由于主成分方程过多，所以我们只列出了前两个主成分的方程：

$$\begin{cases} Y_1=0.195X_1-0.352X_2+0.376X_3+0.210X_4+0.236X_5+0.500X_6+0.534X_7+0.249X_8 \\ Y_2=0.367X_1-0.242X_2-0.643X_4-0.606X_5+0.113X_8 \\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet \end{cases}$$

下图 4-3 是 screeplot( )函数的运行结果，显示的是得到的碎石图，其中碎石图是以特征值作为纵轴，以特征值从大到小排序列出序号，对应序号为横坐标的散点图。碎石土科研帮助我们确定主成分合适的个数，它你能够做到很好的可视化显示特征值差异大小，即主成分的贡献排列情况。由图我们可以得到在第六个特征值作为转折点，此后突然特征值大小陡降，后面的特征值全部较小且彼此大小差不多，所以选取 6 作为我们主成分的个数，这里与前面通过主成分方差累积贡献率分析的结果是一致的。

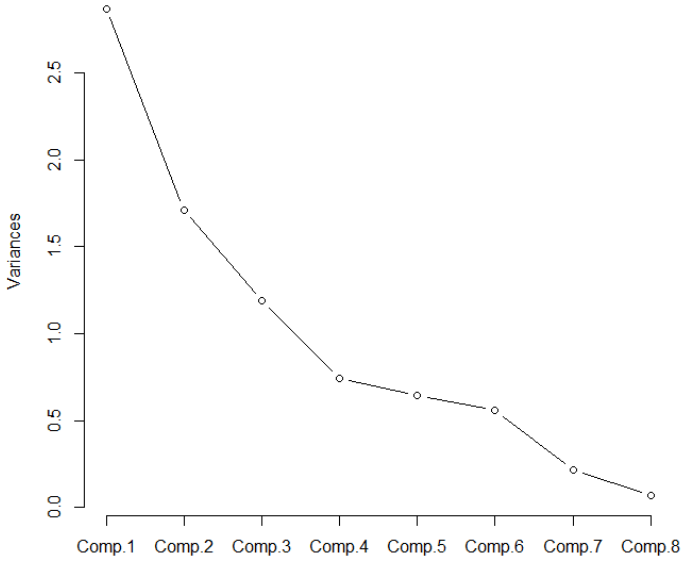


图 4-3 碎石图

图 4-4 是 biplott( )函数的运行结果，显示的是得到的成分图，以第一主成分为横轴，第二主成分为纵轴，做出的散点图，以及绘制出原坐标在主成分下的方向。图中的数据点坐标值代表观测者的对应横纵坐标上的主成分值，当都经过了一定量化，图中箭头红线代表载荷值，其端点的坐标意义与数据点一致，代表对应的载荷系数，我们可以知道图中所示的 NO3 和 NH4 箭头非常接近，说明两者对应的载荷系数值非常相似接近。由图可知整体数据点聚在一起，说明这些数据相似性高，特点相近。

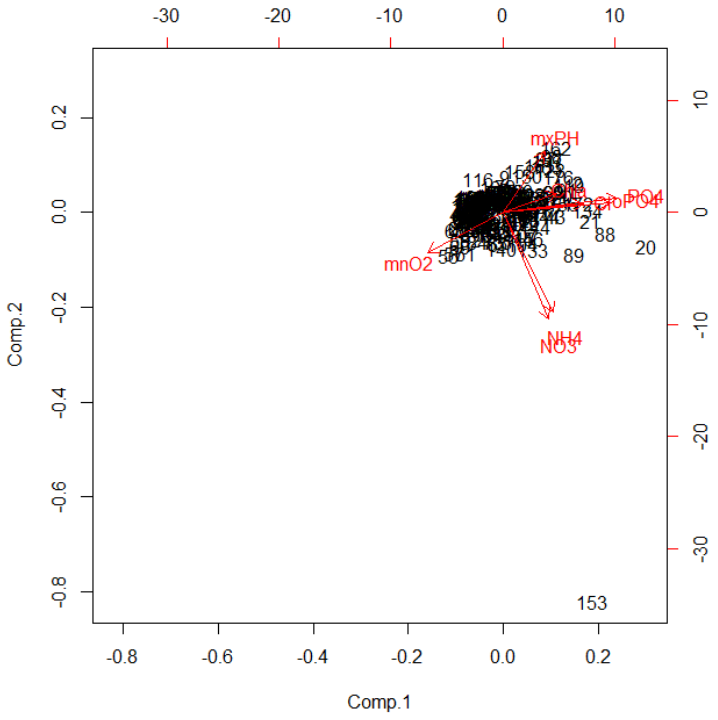


图 4-4 成分图

下面的图 4-5 是显示函数 cor( )的运行结果，显示出运算得到的样本的相关系数矩阵，其对应的值展示的是对应两元素的相关系数，是体现两个变量之间线性相关程度的量。例如图中所示 mxPH 与 Chla 的相关系数为 0.40308892，同时由理论推导也可以得出相关系数矩阵是对称矩阵。

```
> > > > > pre <- predict(prinl) #预测各个样本的主成分值
> cor(std.x)
              mxPH         mnO2           Cl          NO3          NH4         oPO4          PO4         Chla
mxPH  1.00000000 -0.17072468  0.19013569 -0.12912772 -0.09299463  0.1601855  0.1894722  0.40308892
mnO2 -0.17072468  1.00000000 -0.29558258  0.09739192 -0.08810753 -0.4174072 -0.4907571 -0.17068474
Cl    0.19013569 -0.29558258  1.00000000  0.22050804  0.07695853  0.4037214  0.4730340  0.17009405
NO3  -0.12912772  0.09739192  0.22050804  1.00000000  0.72144352  0.1445878  0.1714046  0.14227877
NH4  -0.09299463 -0.08810753  0.07695853  0.72144352  1.00000000  0.2272372  0.2088573  0.09445369
oPO4  0.16018553 -0.41740724  0.40372140  0.14458782  0.22723723  1.0000000  0.9146008  0.13227417
PO4   0.18947221 -0.49075706  0.47303400  0.17140456  0.20885732  0.9146008  1.0000000  0.27274570
Chla  0.40308892 -0.17068474  0.17009405  0.14227877  0.09445369  0.1322742  0.2727457  1.00000000
```

图 4-5 相关系数矩阵

下面的图 4-6 是各样本的综合得分情况。样本在某个主成分的得分是把对应样本的观测值带入主成分中，从而得到对应取值，而以各个主成分的方差贡献率为权的前六个主成分的加权平均得到每个样本的综合得分，这里我们就可以得出综合得分系数矩阵，例如图所示 1 号样本的综合得分是 0.42692013。

```
> y <- eigen(cor(std.x)) #求cor(std.x)的特征值和特征向量
> y1 <- y$values[1] #第一个特征值赋给y1
> y2 <- y$values[2]
> y3 <- y$values[3]
> y4 <- y$values[4]
> y5 <- y$values[5]
> y6 <- y$values[6]
> scores <- (y1*pre[,1]+y2*pre[,2]+y3*pre[,3]+y4*pre[,4]+y5*pre[,5]+y6*pre[,6])/
+          (y1+y2+y3+y4+y5+y6)
> scores #显示得分情况
         1          2          3          4          5          6          7          8          9         10         11         12         13         14
 0.42692013  1.54452932  0.16659050  0.16886138 -0.07448840 -0.03225483  0.03980627 -0.15690463  0.19292945 -0.51584710 -0.71339719 -0.88608855 -0.38271711 -0.75196367
        15         16         17         18         19         20         21         22         23         24         25         26         27         28
-0.55609103 -0.73223489 -0.65944534 -0.68673666 -0.78677200  1.82119548  1.51522754 -0.97753778 -0.50774172 -0.70664255 -0.63697393 -0.64018322 -1.01094114 -1.20799787
        29         30         31         32         33         34         35         36         37         38         39         40         41         42
-0.44561513 -0.99857215 -0.57954290  0.47397021 -0.35922912 -0.25023637 -0.41063786  0.08365523 -0.49338076 -0.63673987  0.48515962  0.29424734  0.51639187  0.18375907
        43         44         45         46         47         48         49         50         51         52         53         54         55         56
 0.96441960  0.87398265 -0.14057862 -0.06830568 -0.34739274 -0.74635906 -0.53491266 -0.83305531 -0.58894287 -0.40270406 -0.65927817 -0.60900382 -1.34017692 -1.90809913
        57         58         59         60         61         62         63         64         65         66         67         68         69         70         71
-1.79642992 -1.22773523 -1.35036058 -1.35564867 -1.37034919 -0.73462369 -0.84963379 -1.07939686 -0.91047572 -1.03800779 -1.19557285  0.46071986  0.27655705  0.23467147
        72         73         74         75         76         77         78         79         80         81         82         83         84         85
 0.16056659 -0.30752371  0.80664325  0.70509856  0.71161837 -0.23863693 -0.17854781 -0.10784187 -0.62933449 -0.74690891 -0.88811041 -0.52763039 -0.77349427 -0.26062387
        86         87         88         89         90         91         92         93         94         95         96         97         98         99
 0.12255353 -0.45530485  1.25330948  0.82091980  1.31988856  1.59743087  0.60189288  0.84254858  0.24949497  0.20006081  0.25569123  1.43253834  1.52116047  0.22065555
       100        101        102        103        104        105        106        107        108        109        110        111        112        113
 0.16146605  0.08795254 -0.04332543  0.07556298 -0.02002412  0.41334770  0.35115890  0.35470247 -0.01597418  0.34878669 -0.03077301 -0.19948272 -0.49170719 -0.34310615
       114        115        116        117        118        119        120        121        122        123        124        125        126        127
-0.19211296 -0.17190446  0.31727928 -0.28702504 -0.44810032  1.34699977  1.08204348  0.42211860  0.49907312  0.35020417 -0.63369140 -0.08439617 -0.02568702  1.22225189
       128        129        130        131        132        133        134        135        136        137        138        139        140        141
 1.54641529  0.43329298 -0.07973053 -0.15473512 -0.75566532  0.42398022  1.78050554  0.12219294  0.22870914 -0.25846652  0.06699750 -0.05570968 -0.18565116  0.29892148
       142        143        144        145        146        147        148        149        150        151        152        153        154        155
-0.61241164 -0.60472616  0.34104075 -0.08297703  0.09225552  0.19768117  0.03917012  0.13694808  0.63744416  0.31331919 -0.11433833 -1.11203858  0.09524364 -0.03490870
       156        157        158        159        160        161        162        163        164        165        166        167        168        169
 0.87819010  0.51219830  0.53767910 -0.09076431  0.02434147  1.28474592  1.84709581  1.48104590  0.62506577 -0.79298112 -0.48953012  1.38974575  0.41534352  0.76838581
       170        171        172        173        174        175        176        177        178        179        180        181        182        183
 0.35808285  0.97811883  1.31206150  0.35664810  0.37685888  1.16914775  1.56916550 -0.20293834 -0.54748381 -0.37765783 -0.29496461 -0.49364372 -0.33120477 -0.41265846
       184        185        186        187        188        189        190        191        192        193        194        195        196        197
-0.38943527 -0.41454727  0.74791715  0.10522897  0.04312348  0.08696292  0.20946056 -0.01991982 -0.22126469 -0.20500791  0.13433451 -0.11258918 -0.09552919 -0.23512144
       198        200
 0.71034254  0.50980324
```

图 4-6 样本的综合得分

5 因子分析

5.1 问题描述与目标

因子分析是主成分分析的推广，也是利用降维的思想，将复杂关系的变量综合为数量较少的几个公共因子，即根据变量相关性大小将变量分组，把相关性高的变量分到同一组，使得不同组的变量相关性较低，同一组的变量用一个不可观测的综合变量表示，这个综合变量就是公共因子，它可以反映原来变量的大部分公共信息，不同组的变量用不同的公共因子表示。

$$x_i - \mu_i = l_{i1}F_1 + \cdots + l_{ik}F_k + \varepsilon_i.$$

其因子分析模型如上式所示，其中 x 是随机变量，μ 是均值向量，l 是因子系数，也叫因子载荷，F 即是公共因子，ε 是特殊因子。简述而言因子分析就是通过研究众多变量之间的内部依赖关系，探求观测数据的基本结构，并用少数几个公共因子来表示原始数据。

对于本节的因子分析我们跟前面两节一样，我们只需要前面 11 个变量中的后面 8 个变量，即舍弃前三个名义变量，留下水样记录中的表示不同化学参数的八个变量。我们的任务即是对这些数据进行因子分析，对变量简化为较少的公因子来表示，同时求出累积贡献率大的公因子，写出数据对应的因子模型。

5.2 实验步骤

我们输入以下程序代码：

```
fact <- factanal(clean.algae[,4:11], 4, scores="Bartlett", rotation="varimax") #进行因子分析
fact #显示因子分析结果
fact$scores #显示所有样本的因子得分
colMeans(clean.algae[,4:11]) #计算样本的各化学元素对应的均值
```

函数 factanal( )是进行因子分析，它可以从样本数据、样本协方差矩阵或者相关矩阵出发对数据作因子分析，并且可以给出方差最大正交旋转后的因子载荷矩阵。

fact$scores 是在因子分析结果中提取 scores 的结果，即显示因子分析结果中所有样本的因子得分。

函数 colMeans( )即是对输入的数据进行按列求取每列的平均值，在这就是计

算样本的各化学元素对应的均值。

5.3 结果与分析

我们对数据集进行进行因子分析，我们设置的因子个数是 4 各公因子，其中采用的因子得分的算法是 Bartlett 因子得分方法，同时在因子分析中进行因子旋转，采用的是方差最大正交旋转方法。对此，我们得到的结果见图 5-1，其中显示结果分为三部分。

```
> fact <- factanal(clean.algae[,4:11], 4, scores="Bartlett", rotation="varimax")
> #进行因子分析
> fact #显示因子分析结果

Call:
factanal(x = clean.algae[, 4:11], factors = 4, scores = "Bartlett",     rotation = "varimax")

Uniquenesses:
 mxPH  mnO2    Cl   NO3   NH4  oPO4   PO4  Chla
0.005 0.708 0.730 0.005 0.457 0.005 0.014 0.686

Loadings:
     Factor1 Factor2 Factor3 Factor4
mxPH  0.120  -0.117   0.983
mnO2 -0.493   0.122          -0.159
Cl    0.422   0.199   0.162   0.163
NO3           0.991           0.107
NH4   0.116   0.727
oPO4  0.952   0.155          -0.245
PO4   0.968   0.140           0.143
Chla  0.179   0.111   0.394   0.339

               Factor1 Factor2 Factor3 Factor4
SS loadings      2.324   1.635   1.171   0.260
Proportion Var   0.291   0.204   0.146   0.032
Cumulative Var   0.291   0.495   0.641   0.674

Test of the hypothesis that 4 factors are sufficient.
The chi square statistic is 21.64 on 2 degrees of freedom.
The p-value is 2e-05
```

图 5-1 因子分析结果

第一部分如下图 5-2 所示，第一行显示的是因子分析的输入部分，第二行显示的是得到的各化学元素的特殊方差，即 ψ_{ii} 的估计值，例如 mxPH 的特殊方差值是 0.005。

```
Call:
factanal(x = clean.algae[, 4:11], factors = 4, scores = "Bartlett",     rotation = "varimax")

Uniquenesses:
 mxPH  mnO2    Cl   NO3   NH4  oPO4   PO4  Chla
0.005 0.708 0.730 0.005 0.457 0.005 0.014 0.686
```

图 5-2 特殊方差结果

第二部分如下图 5-3 所示，显示的旋转后的因子载荷矩阵，采用的是方差最大正交旋转方法进行因子载荷矩阵的旋转，所谓的因子旋转即是更换因子分析模型中的因子载荷矩阵和公因子向量，使得更换后的因子分析模型中参数更具有实际意义。例如图中我们得到的 mxPH 对应的第一个公因子的因子载荷值是 0.120，

第二个公因子的因子载荷值是-0.117.

```
Loadings:
     Factor1 Factor2 Factor3 Factor4
mxPH  0.120  -0.117   0.983
mnO2 -0.493   0.122          -0.159
Cl    0.422   0.199   0.162   0.163
NO3           0.991           0.107
NH4   0.116   0.727
oPO4  0.952   0.155          -0.245
PO4   0.968   0.140           0.143
Chla  0.179   0.111   0.394   0.339
```

图 5-3 旋转后的因子载荷矩阵

第三部分上面显示的是方差贡献、方差贡献率和累积方差贡献率，下面显示是四个公因子的充分假设检验结果。结果如图 5-4 所示，我们可以得到第 1 个公因子的方差贡献为 2.324，其方差贡献率是 29.1%，其中前四个公因子的累积方差贡献率是 67.4%。这四个公因子的假设检验结果，其中 χ^2 检验值是 21.64，检验 p 值为 2e-05，说明我们的结果是比较显著的。

```
                Factor1 Factor2 Factor3 Factor4
SS loadings       2.324   1.635   1.171   0.260
Proportion Var    0.291   0.204   0.146   0.032
Cumulative Var    0.291   0.495   0.641   0.674

Test of the hypothesis that 4 factors are sufficient.
The chi square statistic is 21.64 on 2 degrees of freedom.
The p-value is 2e-05
```

图 5-4 方差贡献以及假设检验

下图 5-5 显示的各个样本的因子得分，因为显示的数据量过于庞大，所以我们只截取了前 21 个样本的因子得分结果。因子得分的计算是通过 Bartlett 因子得分法计算得到的，例如图中所显示样本 1 的第一公因子得分是 0.187984663，第二公因子得分是 0.796844798，第三公因子得分是 0.04667412，第四公因子得分是-0.11304628。

```
> fact$scores #显示所有样本的因子得分
         Factor1      Factor2     Factor3     Factor4
1    0.187984663  0.796844798  0.04667412 -0.11304628
2    3.661120465 -0.396401523  0.11927667 -1.92295293
3    0.354134604  0.596842994  0.17938113 -0.52638189
4    0.009531927 -0.307608495  0.04079290  0.41036478
5   -0.588423349  1.937640590  0.37771510 -0.24411369
6   -0.950219847  1.615346525  0.70738169  0.03335471
7   -0.138672719 -0.435140484  0.19583868 -0.21489426
8   -0.521312134  0.504976087  0.18480029 -0.35453793
9   -0.494038765 -0.580808423  1.17258318 -0.27634341
10  -0.587257177 -0.438988371 -0.12340319 -0.52781064
11  -0.751232103 -0.398890292 -0.49387249 -0.59185758
12  -0.736472607 -0.393049443 -0.92710079 -0.56373672
13  -0.744908313 -0.484000812 -0.43163985 -0.52452363
14  -0.777781307 -0.387027017 -0.44962356 -0.82345781
15  -0.558943097 -0.498764551 -0.20159961  0.15179727
16  -0.339846176 -0.649558971 -0.87199237  1.09379539
17  -0.587696238 -0.464842199 -0.39194023 -0.24659333
18  -0.504264164 -0.485421560 -0.70156626 -0.07788949
19  -0.373162803 -0.369026729 -1.14080603 -0.57658679
20   5.360267841 -0.052276084 -1.02210413 -1.55974997
21   3.888247667  0.579688314 -0.67764753 -2.36804784
```

图 5-5 样本的因子得分

下面图 5-6 显示了各化学元素对应的所有值计算出来的均值，其对应的是因子分析模型中的均值向量，例如图中所示 mxPH 得到的均值就是 8.018826。

```
colMeans(clean.algae[,4:11]) #计算样本的各化学元素对应的均值
    mxPH       mnO2         Cl        NO3        NH4       oPO4        PO4       Chla
8.018826   9.132255  42.594535   3.282389 501.295828  73.590596 137.888956  13.470239
```

图 5-6 各化学元素的均值

由上面得到的结果，我们可以得出因子模型：

$$\begin{cases} x_1 - 8.018826 = \ \ 0.120 f_1 - 0.117 f_2 + 0.983 f_3 + \varepsilon_1 \\ x_2 - 9.132255 = -0.493 f_1 + 0.122 f_2 - 0.159 f_4 + \varepsilon_2 \\ \quad \bullet \bullet \bullet \bullet \bullet \bullet \end{cases}$$

由于因子模型的方程较多，故我们只列写出了前两个方程，由上式因子模型可见公因子 f_1 在 x_6，x_7 的载荷挺大，其中在 x_4 载荷为 0，该公因子它综合反映了 oPO4 和 PO4 的情况，这与化学元素性质上 oPO4 和 PO4 相似性相吻合，所以使用公因子 f_1 代表该种综合情况具有可行性，对此样本在此因子的因子得分越高说明在这两个化学元素上含量越高。其它因子也可按此类似分析得到其它结论。

6 聚类分析

6.1 问题描述与目标

聚类分析是研究如何将研究对象按照多个方面的特征进行综合分类的一种多元统计方法，它根据物以类聚的原理将相似的样品聚为一类。聚类是把相似的对象通过静态分类的方法分成不同的组别或者更多的子集，这样让在同一个子集中的成员对象都有相似的一些属性。

聚类按照方法有系统聚类法和快速聚类法，系统聚类法又叫谱系聚类法，快速聚类法又叫 K 均值法（K-means）。

$$\left.\begin{array}{ll} D_{HI}=min(d_{mn}) & ,m\in H \quad ,n\in I \\ D_{HJ}=min(d_{mn}) & ,m\in H \quad ,n\in J \end{array}\right\} \Rightarrow D_{HK}=min(D_{HI},D_{HJ})$$

系统聚类是将每个样品分成若干类的方法，其基本思想是：先将各个样品各看成一类，然后规定类与类之间的距离，选择距离最小的一对合并成新的一类，计算新类与其他类之间的距离，再将距离最近的两类合并，这样每次减少一类，直至所有的样品合为一类为止。其中最短距离法的递推公式如上式所示，假若 K 类是由 I 和 J 两类合并而成，D_{HK} 表示 H 类中的所有样本和 K 类中的所有样本之间的最小距离。

$$J=\sum_{j=1}^{k}\sum_{i=1}^{n}\left\|x_i^{(j)}-z_j\right\|^2$$

快速聚类法基本思想是：基于使聚类性能指标最小化，所用的聚类准则函数是聚类集中每一个样本点到该类中心的距离平方之和，并使其最小化。其算法的聚类准则函数为上式所示，x 表示样本点，z 表示聚类中心，我们的目的是要寻找聚类准则函数 J 的最小值作为目标。

对于本节的聚类分析我们跟前面三节一样，只需要前面 11 个变量中的后面 8 个变量，即舍弃前三个名义变量，留下水样记录中的表示不同化学参数的八个变量。我们的任务即是对这八个数据变量进行聚类分析，通过聚类把相似的变量分成同一组别，这样让在同一个类中的成员对象都有相似的一些属性。

6.2 实验步骤

本节的聚类分析的实验，我们分别按照系统聚类法和快速聚类法两个方法来

完成。

采用系统聚类法，我们输入以下程序代码：

```
#系统聚类法
std <- scale(t(clean.algae[,4:11]), center=TRUE, scale=TRUE) #数据标准化
d0 <- dist(std, method="minkowski", diag=TRUE, upper=FALSE, p=1) #计算距离矩阵
hcs <- hclust(d0, method="complete") #系统聚类
plot(hcs, hang=-1) #画树形图
rect.hclust(hcs, k=3, h=NULL, border=2) #分类
```

函数 scale( )是用来将数据进行中心化和标准化，为了消除量纲对数据结构的影响。

函数 dist( )是用来计算输入数据的距离矩阵的函数，其中 method 包括 6 种方法，表示不同的距离测度，这里我们选择的是闵可夫斯基（minkowski）度量，函数输出结果就是距离矩阵。

函数 hclust( )是系统聚类的函数，其中参数 method 可以进行方法选择，对系统聚类进行选择。这里我们选择的是 complete 参数，即对应着我们的系统聚类采用最长距离法进行聚类。

函数 plot( )即是画出系统聚类的树形图，其中 hang=-1 是使样品标号对齐，位于同一行。

采用快速聚类法，我们输入以下程序代码：

```
#快速聚类法
std <- scale(t(clean.algae[,4:11]), center=TRUE, scale=TRUE) #数据标准化
kmeans(std, 3, iter.max=20, algorithm="Hartigan-Wong") #快速聚类
```

函数 scale( )即是同上面一样，用来将数据进行中心化和标准化，为了消除量纲对数据结构的影响。

函数 kmeans( )是对数据进行进行快速聚类，这里我们聚的类别设置成三类，迭代次数最大为 20 次，采用迭代算法为 Hartigan-Wong 算法。

6.3 结果与分析

对于系统聚类法的选择，本节采用了闵可夫斯基距离作为样本之间的度量，其中采取最长距离法进行聚类，并且画出聚类树形图，采用红线框画出所聚的类

别，结果见图 6-1 所示。如图所示 oPO4、Cl、mxPH、mnO2、NO3 和 Chla 聚成一类，另外的 NH4 聚成一类，PO4 聚成一类，总共分成三类。并且由树形图能发现每类之间的距离差值都在 150 以上，并且三类之间的距离有明显的差距，对此表明我们得到的分类结果非常良好，符合要求。

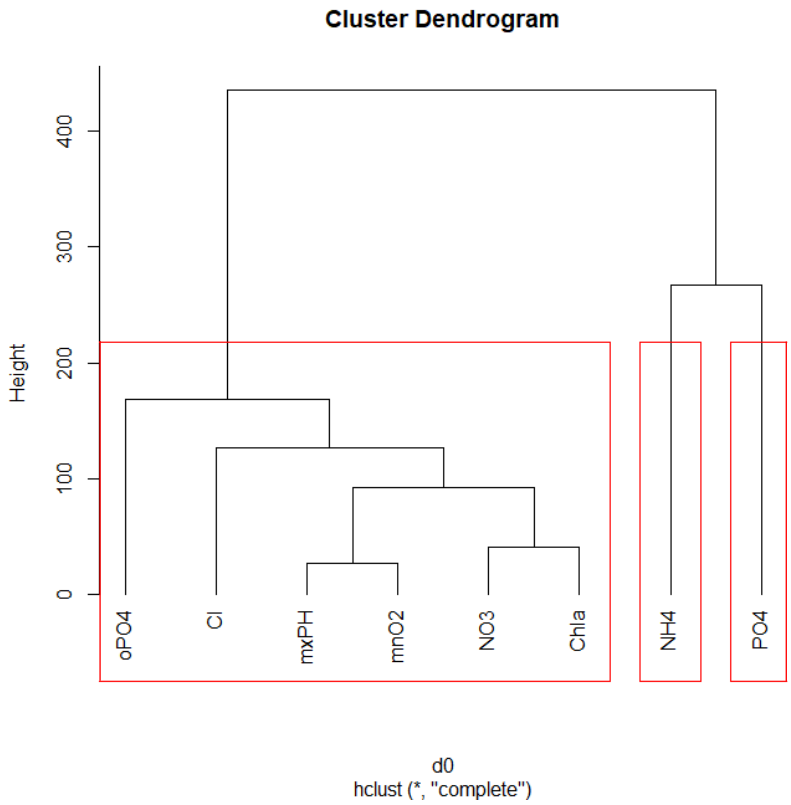


图 6-1 树形聚类图

对于快速聚类法的选择，得到四部分的结果，其中分别显示有聚得的各个类的样品个数，得到的各类的均值，各个样品的分类情况以及各个类的类内平方和这四个部分信息。

得到的第一部分显示结果如图 6-2 所示，显示的是各类样品的个数，我们得到聚类将八个变量分成的三类，结果显示三类分别分成 1 个变量个数，1 个变量个数，以及 6 个变量个数。

```
> std <- scale(t(clean.algae[,4:11]), center=TRUE, scale=TRUE) #数据标准化
> kmeans(std, 3, iter.max=20, algorithm="Hartigan-Wong") #快速聚类
K-means clustering with 3 clusters of sizes 1, 1, 6
```

图 6-2 各类样品个数

得到的第二部分显示结果如下图 6-3 所示，显示的是各类样品的均值。因为显示的结果数据过多，所以我们在这只截取部分结果显示。如图中所示我们可以得到的第一类的第 1 向量的均值为 2.3655583，第二类的第 1 向量的均值为

0.2421154，第三类的第 1 向量的均值为-0.4346123 等等，由此可以读出得到个类的各个样品向量上的均值。

```
Cluster means:
          1          2          3          4          5          6          7          8          9         10         11
1  2.3655583  0.8193610  2.0793034  0.9319940  2.2428302  2.4463853  1.3131183  2.2612870  1.9218215 -0.4877201  1.4381676
2  0.2421154  1.6302310  0.7736593  1.6998061  0.4816522 -0.1534937  1.3498441  0.3721611  1.0676338  2.1104821  1.3323550
3 -0.4346123 -0.4082653 -0.4754938 -0.4386334 -0.4540804 -0.3821486 -0.4438271 -0.4389080 -0.4982426 -0.2704603 -0.4617538
          12         13         14         15         16         17         18         19         20         21         22
1  1.276802  0.9258165 -0.0610640  1.2307765 -0.09686895  1.0283900  0.7418826  1.0983312  2.4628284  2.2839891  2.39536139
2  1.360954  0.1439572  1.1645934  1.9000783  2.44870383  1.9337754  2.1066196  1.6146292 -0.1650775  0.3718376  0.05142704
3 -0.439626 -0.1782956 -0.1839216 -0.5218091 -0.39197248 -0.4936942 -0.4747504 -0.4521601 -0.3829585 -0.4426378 -0.40779807
          23         24         25         26         27         28         29         30         31         32         33
1  1.3929599  2.45747507  1.1335937  1.919261  2.1386059  1.7130218  0.2853046  2.44532595  0.4188120 -0.1866870  0.3315427
2  1.2666773 -0.08840495  0.9165082  1.226552  0.4998492  1.0093151  1.9800158 -0.08528277  2.0852198  1.8268675  2.2096115
3 -0.4432729 -0.39484502 -0.3416837 -0.524302 -0.4397425 -0.4537228 -0.3775534 -0.39334053 -0.4173386 -0.2733634 -0.4235257
          34         35         36         37         38         39         40         41         42         43         44
1  2.4746712  2.4748485  2.4724422  1.6486695  1.2866203  1.2180565  0.9160815  0.3876309  2.3907054 -0.1657019 -0.3196838
2 -0.3299468 -0.3482134 -0.2688897 -0.3709927 -0.3707267  1.4355228  1.6573771  2.0658817  0.2451150  1.8742552  1.7854062
3 -0.3574541 -0.3544392 -0.3672588 -0.2129461 -0.1526489 -0.4422632 -0.4289098 -0.4089188 -0.4393034 -0.2847589 -0.2442871
          45         46         47         48         49         50         51         52         53         54         55
1  0.5933461 -0.12921634  0.6947056  0.8254926  0.2966658  2.0415071  2.4010869  1.3739146  1.7644545  2.2918895  1.2018789
2  0.7277759  0.62012246  0.6947056 -0.1086683  2.3422014  0.5911590  0.1523086  0.8714549  0.6404272  0.1947285  0.1354281
3 -0.2201870 -0.08181769 -0.2315685 -0.1194707 -0.4398112 -0.4387777 -0.4255659 -0.3742282 -0.4008136 -0.4144363 -0.2228845
          56         57         58         59         60         61         63         64         65         66
1  0.1499041  0.92380606  1.4990815  0.8823113  0.9438242  0.5534789  1.95034868  2.2742811  2.4692605  1.83990031
2 -0.8779657 -0.50672995 -0.3770015  1.1250042 -0.1183636  1.5854948 -0.01406552 -0.1060840 -0.3119025  0.09279092
3  0.1213436 -0.06951268 -0.1870133 -0.3345526 -0.1375768 -0.3564956 -0.32271386 -0.3613662 -0.3595597 -0.32211520
          67         68         69         70         71         72         73         74         75         76
1  1.6799787  1.8490710  2.3814029  2.44107820  2.45347346  1.9843582  2.0129178  1.3612756  1.6232165 -0.2656354
2  0.8564835 -0.5307752  0.2103294 -0.02414593 -0.07141782  1.1063720  0.6058253  1.4014467  1.2742021  2.3049620
3 -0.4227437 -0.2197160 -0.4319554 -0.40282204 -0.39700927 -0.5151217 -0.4364572 -0.4604537 -0.4829031 -0.3398878
          77         78         79         80         81         82         83         84         85         86         87
1  2.2422016  1.8273611  1.9770688  1.516375  2.4110561  2.2544767  1.7959066  1.8389345  2.3657892  2.3721983  2.2944141
2  0.6686711  1.2824112  1.0977175  1.673603  0.1011563  0.6774666  1.3846643  1.2426126  0.2932799  0.2623568  0.5362252
3 -0.4851455 -0.5182954 -0.5124644 -0.531663 -0.4187021 -0.4886572 -0.5300952 -0.5135912 -0.4431782 -0.4390925 -0.4717732
```

图 6-3 各类均值

第三部分如图 6-4 所示，显示的是个样品的分类，即是得到的八个变量的聚类情况。如图所示，我们得到 NH4 聚成第一类，PO4 聚成第二类，剩下的 mxPH、mnO2、Cl、NO3、oPO4 和 Chla 聚成第三类。这与我们上面采用的系统聚类方法得到的聚类结果一致，说明我们得到结果的正确性。

```
Clustering vector:
mxPH mnO2   Cl  NO3  NH4 oPO4  PO4 Chla
   3    3    3    3    1    3    2    3
```

图 6-4 样品分类情况

第四部分如图 6-5 所示，显示的结果是聚类的类内平方和。如图所示，显示的第一类类内平方和为 0，第二类的类内平方和也为 0，第三类的类内平方和为 320.9614，前两个的结果与我们前两类的类内样品数只有一个是吻合的。组间的距离平方和占了整体距离平方和的的 76.8%，也就是说各个聚类间的距离做到了最大，这也说明了我们得到的聚类结果的正确性。

```
Within cluster sum of squares by cluster:
[1]   0.0000   0.0000 320.9614
 (between_SS / total_SS =  76.8 %)

Available components:

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

图 6-5 类内平方和

结语

我们采用的海藻数据集进行数据分析，本质上最终应该是采用河流的一些变量来进行河流里面各类海藻的预测，并且评估模型。但是由于其它因素限制吧，没能完成预测的部分以及模型的评估，也是挺遗憾的。而且由于数据集变量中含有非数值类型变量，例如季节和河流大小以及流速等，虽然 R 也能处理这类数据，但是也会使我的工作难度增加，所以不得不抛弃了这类非数值变量的加入，只采用了河流里的八个数值类型的化学参数作为变量。在做数据分析之前我们先简单分析了下数据，并且进行了数据的预处理，完成缺失数据的填补和删除。后面第一个任务是做回归分析，通过变量预测河流内的藻类频率，我们只选择了 7 种藻类种的一种做回归分析（其它剩下 6 种分析方法一样），但是我们得到的的回归模型显示并不是很好，个人觉得可能因为我们抛弃了那三个非数值型的名义变量的原因吧，如果加上那三个可能分析出的模型会更好些。第二和第三个任务就是对八个变量进行主成分分析和因子分析，简单就是说将八个冗余变量变得更加精简，采用降维方法缩小描述性变量的维度，总体来说这两个得到的结果是还是非常令人满意的。第四个任务就是对八个变量进行聚类分析，将八个变量进行聚类，找到其中的相关性，我们最后得到的结果显示也是非常好的，得到的聚类效果也非常满意。总体来说，这次任务完成圆满。

参考文献

[1] 李素兰. 数据分析与 R 软件（第二版）[M]. 科学出版社, 2017.

[2] Torgo L. Data mining with R: learning with case studies[M]. Chapman and Hall/CRC, 2011.

附录

```
#########################################################################
### 第 2 章 数据预分析处理
#########################################################################
algae <- read.table('C:\\数据分析与 R 软件\\Analysis.txt',
            header=F,  #表示要读的文件的第一行不包括变量名
            dec='.', #指出数值使用'.'字符分隔小数
            col.names=c('season','size','speed','mxPH','mnO2','Cl',
            'NO3','NH4','oPO4','PO4','Chla','a1','a2','a3','a4',
            'a5','a6','a7'), #给正在读取的变量一个名称向量
            na.strings=c('XXXXXXX')) #表示字符串被解释为未知值
head(algae) #显示数据框前六行
summary(algae) #给出数据的统计特性概览
library(DMwR) #加载函数库
data(algae) #重新读取数据
algae <- algae[-manyNAs(algae), ] #数据缺失率大于 20%的样本剔除
clean.algae <- knnImputation(algae, k = 10) #进行数据填补

#########################################################################
### 第 3 章 回归分析
#########################################################################
data(algae)
algae <- algae[-manyNAs(algae), ]
clean.algae <- knnImputation(algae, k = 10) #预处理

lm.a1 <- lm(a1 ~ .,data=clean.algae[,4:12]) #建立多元线性回归模型
formula(lm.a1) #提取模型公式
summary(lm.a1) #提取模型资料，显示模型拟合结果
anova(lm.a1) #计算方差分析表
plot(lm.a1) #制作模型诊断图

#########################################################################
### 第 4 章 主成分分析
#########################################################################
data(algae)
algae <- algae[-manyNAs(algae), ]
clean.algae <- knnImputation(algae, k = 10) #预处理

std.x <- scale(clean.algae[4:11]) #数据标准化
prin1 <- princomp(std.x,cor=TRUE) #从相关阵 R 出发做主成分分析
summary(prin1) #列出主成分分析主要结果
loadings(prin1) #各主成分对应的系数，即相关矩阵 R 的单位正交化的特征向量
screeplot(prin1,type="lines") #画出主成分的碎石图
```

```
biplot(prin1) #画数据关于前两个主成分的散点图和原坐标在主成分下的方向
pre <- predict(prin1) #预测各个样本的主成分值
cor(std.x)
y <- eigen(cor(std.x)) #求 cor(std.x)的特征值和特征向量
y1 <- y$values[1] #第一个特征值赋给 y1
y2 <- y$values[2]
y3 <- y$values[3]
y4 <- y$values[4]
y5 <- y$values[5]
y6 <- y$values[6]
scores <- (y1*pre[,1]+y2*pre[,2]+y3*pre[,3]+y4*pre[,4]+y5*pre[,5]+y6*pre[,6])/
          (y1+y2+y3+y4+y5+y6) #计算每个样本的综合得分
scores #显示得分情况

#########################################################################
### 第 5 章 因子分析
#########################################################################
data(algae)
algae <- algae[-manyNAs(algae), ]
clean.algae <- knnImputation(algae, k = 10) #预处理

fact <- factanal(clean.algae[,4:11], 4, scores="Bartlett", rotation="varimax") #进行因子分析
fact #显示因子分析结果
fact$scores #显示所有样本的因子得分
colMeans(clean.algae[,4:11]) #计算样本的各化学元素对应的均值

#########################################################################
### 第 6 章 聚类分析
#########################################################################
data(algae)
algae <- algae[-manyNAs(algae), ]
clean.algae <- knnImputation(algae, k = 10) #预处理

#系统聚类法
std <- scale(t(clean.algae[,4:11]), center=TRUE, scale=TRUE) #数据标准化
d0 <- dist(std, method="minkowski", diag=TRUE, upper=FALSE, p=1) #计算距离矩阵
hcs <- hclust(d0, method="complete") #系统聚类
plot(hcs, hang=-1) #画树形图
rect.hclust(hcs, k=3, h=NULL, border=2) #分类
#快速聚类法
std <- scale(t(clean.algae[,4:11]), center=TRUE, scale=TRUE) #数据标准化
kmeans(std, 3, iter.max=20, algorithm="Hartigan-Wong") #快速聚类
```