（十）无指导学习

无指导学习

- 本课程主要内容都是有关回归和分类这样的**有指导学习(supervised learning)**方法
- 在有指导的学习中，通常会碰触到n个观测上p 个特征$X_1, X_2, \ldots, X_p$，对n个观测中的每一个都对应一个响应变量Y。有指导学习的目的是用$X_1, X_2, \ldots, X_p$去预测Y。
- 本节课的焦点转向**无指导学习(unsupervised learning)**。旨在研究仅包含由n个观测组成的p 个特征$X_1, X_2, \ldots, X_p$的情况。无指导学习的主要兴趣并非预测，因为数据中没有一个与之关联的响应变量Y。

学习目标

- 无指导学习旨在发现由$X_1, X_2, \ldots, X_p$构成的观测空间中一些有价值的模式：是否可以找到一种将数据中主要的信息集中显示出来的可视化方法？能否从变量或观测中找到一些子类？

- 我们讨论两类方法：
  - **主成分分析(principle components analysis)**，一种用于数据可视化以及在有指导学习方法之前对数据进行预处理的工具
  - **聚类分析(clustering)**，是一大类寻找数据中未知子类的方法

挑战

- 与有指导学习相比，无指导学习通常更具挑战性，因为不设定明确的目标作为分析的指向，比如预测中的响应变量
- 无指导学习技术在很多领域中变得越来越重要
  - 根据基因表达水平，从乳腺癌样本或基因中找到一些子类
  - 通过分析相似购物者浏览和购买商品的历史记录来定位某一类购物人群
  - 搜索引擎基于搜索历史记录将搜索模式相似的人归类

额外的优势

- 获取无标签数据(unlabeled data)是很容易的，比如通过计算机或实验设备。但获得有标签数据(labeled data)要困难的多，因为需要人工介入 / 标注
  - 例如，很难自动的评估一条电影评论的整体情感：是喜欢该电影还是不喜欢?

主成分分析

- 主成分分析(principle components analysis) — PCA
- PCA产生一个对数据集的低维表示。它可以找到具有最大方差且互不相关的变量的线性组合序列。
- 除了能够在有指导学习模型中充当派生变量之外，PCA还可以作为数据可视化（观测或变量的可视化）的工具。

主成分分析：细节

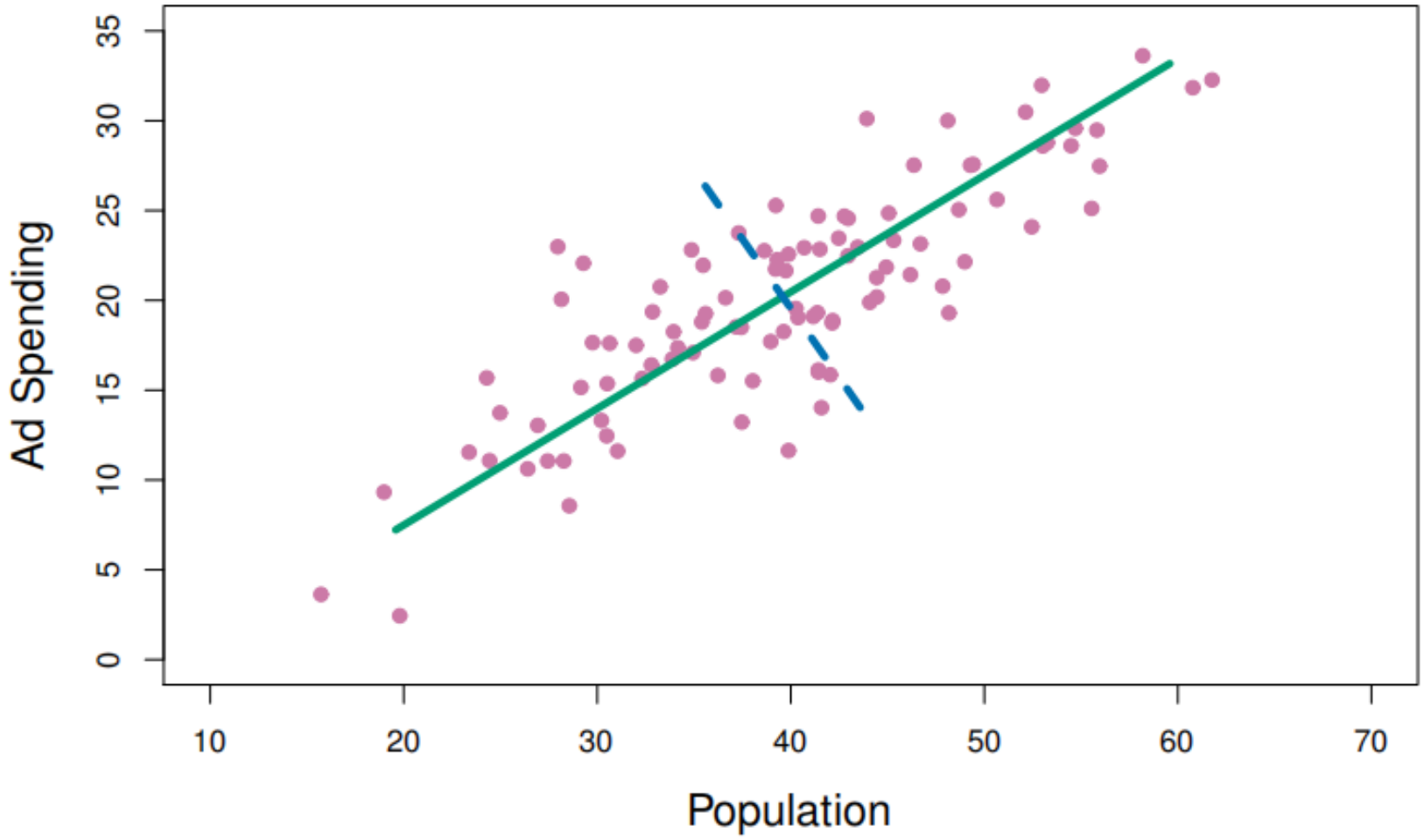
- 给定一组变量$X_1, X_2, \ldots, X_p$的**第一主成分(first principle component)**是变量标准化线性组合中方差最大的组合，如下：

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{p1}X_p$$

- **标准化(normalized)**的含义是$\sum_{j=1}^{p} \phi_{j1}^2 = 1$,其中，所述$\phi_{11}, \phi_{21}, \ldots, \phi_{p1}$指的是第一主成分的**载荷(loading)**。同时，这些载荷构成了主成分的载荷向量$\phi_1 = \left(\phi_{11}\ \phi_{21} \cdots \phi_{p1}\right)^T$。

- 为了防止载荷绝对值任意大而导致方差变得任意大，限定这些载荷的平方和为1。

主成分分析：示例



- 紫色圆圈表示100个不同城市的人口规模(pop)和广告支出(ad)。绿色实线表示第一主成分方向，蓝色虚线表示第二主成分方向。

- 假设有一个$n \times p$维数据集$\boldsymbol{X}$。因为主成分只对方差感兴趣，所以假设$\boldsymbol{X}$中的每个变量都经过中心化处理，其均值均为0（即矩阵$\boldsymbol{X}$ 在列方向上的均值均为0）。
- 寻找具有如下形式的样本特征值的线性组合：

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + \cdots + \phi_{p1}x_{ip} \quad (1)$$

- 该组合在限定条件$\sum_{j=1}^{p} \phi_{j1}^2 = 1$下，有最大的样本方差$(i = 1, \ldots, n)$。即第一主成分的载荷向量在解如下的最优化问题：

$$\text{maximize}_{\phi_{11},\ldots,\phi_{p1}} \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2, \ \sum_{j=1}^{p}\phi_{j1}^2 = 1 \quad (2)$$

- 将(1)代入(2)中，将需要最大化的目标函数写成$\frac{1}{\mathrm{n}}\sum_{i=1}^{n} z_{i1}^2$。

- 因为x_{ij}的均值为0（即$\frac{1}{\mathrm{n}}\sum_{i=1}^{n} x_{ij} = 0$），$z_{11}, \dots, z_{n1}$的均值也为0。
  - $\frac{1}{\mathrm{n}} \sum_{i=1}^{n} z_{i1} = \frac{1}{\mathrm{n}} \sum_{i=1}^{n} \sum_{j=1}^{p} \phi_{j1} x_{ij}$
  - $= \frac{1}{\mathrm{n}} \sum_{j=1}^{p} \sum_{i=1}^{n} \phi_{j1} x_{ij} = \frac{1}{\mathrm{n}} \sum_{j=1}^{p} (n\phi_{j1} \sum_{i=1}^{n} x_{ij}) = 0$
- 因此(2)中需要最大化的目标函数正是z_{i1}的n个值的样本方差。
- $z_{11}, \dots, z_{n1}$即为第一主成分的得分(score)。
- 可以通过线性代数中一个基本技术——特征分解——解问题(2)。

几何学解释

- 载荷向量$\phi_1 = (\phi_{11}\ \phi_{21}\ \cdots \phi_{p1})^T$定义了一个在向量空间上数据变异最大的方向。
- 如果将这n个数据点$x_1, \ldots, x_n$投影到这个方向上，这些投影值就是主成分的得分$z_{11}, \ldots, z_{n1}$。

- 第二主成分也是$X_1, X_2, \ldots, X_p$的线性组合，这个线性组合是与Z_1**不相关**的各种线性组合中方差最大的一个。
- 第二主成分得分$z_{12}, z_{22}, \ldots, z_{n2}$有以下形式：

$$z_{i2} = \phi_{12}x_{i1} + \phi_{22}x_{i2} + \cdots + \phi_{p2}x_{ip},$$

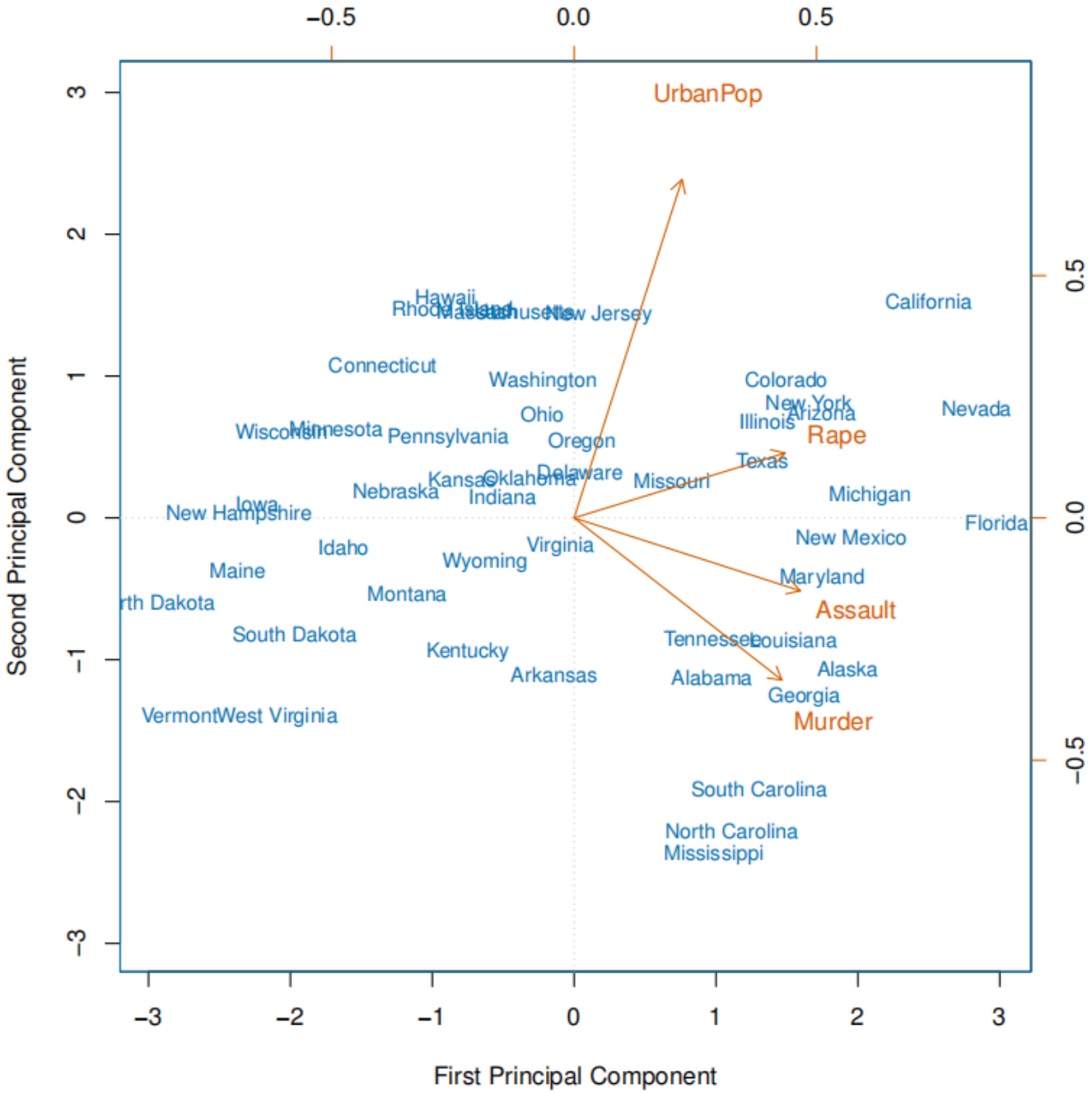
其中，ϕ_2表示的是第二主成分的载荷向量，其分量是$\phi_{12}, \phi_{22}, \ldots, \phi_{p2}$。

- 它表明要使得Z_2和Z_1不相关，相当于使ϕ_2的方向与ϕ_1的方向垂直(orthogonal)，以此类推。
- 主成分方向$\phi_1, \phi_2, \phi_3, \cdots$是矩阵$\mathbf{X}^T\mathbf{X}$的特征向量依次排序，主成分的方差是其特征根，数据最多可以有$\min(n-1, p)$个主成分。

USArrests数据集示例

- USArrests（犯罪统计）数据集包含了美国50个州中每100,000个居民中因犯Assault(强暴)、Murder(谋杀)和Rape(强奸)3种罪而被逮捕的人数，以及UrbanPop(各个州城镇居民的比例)。
- 主成分得分向量长度$n = 50$，主成分载荷向量的长度为$p = 4$。
- PCA之前每个变量都经过标准化，均值为0而且方差为1。

USArrests数据集：PCA图



PCA图详细信息

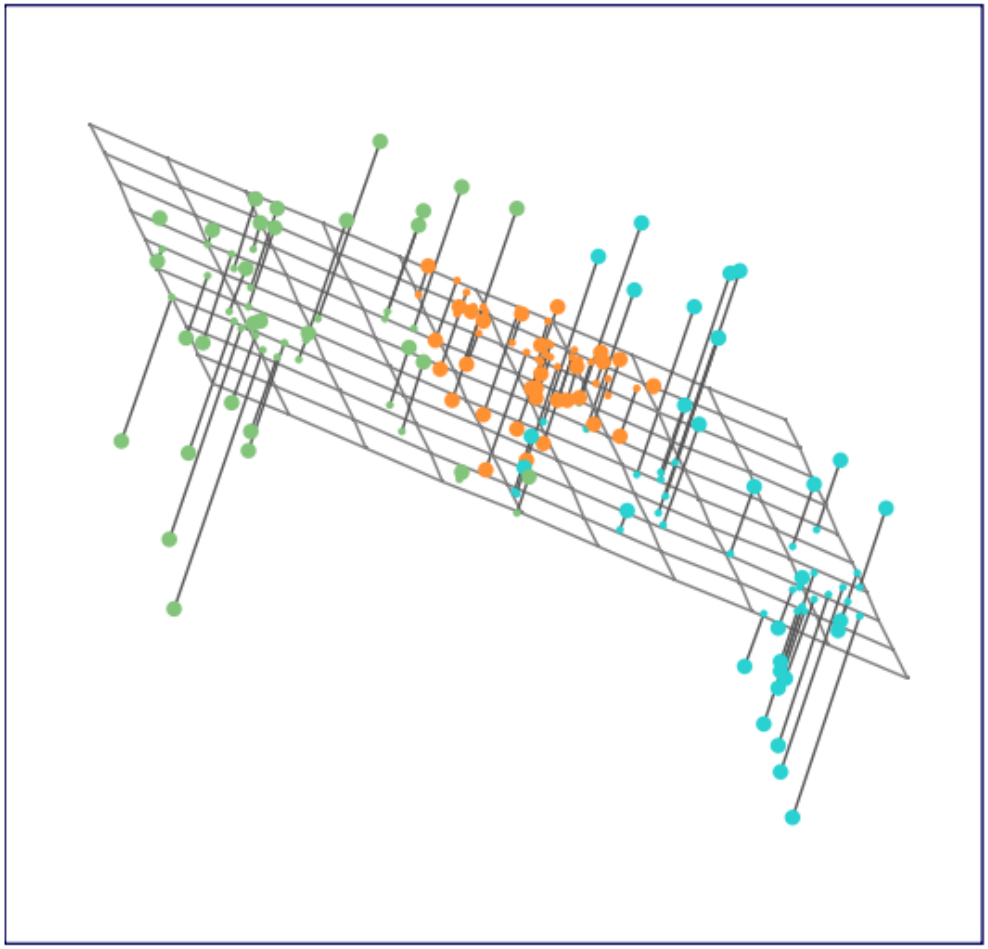
- 图中显示USArrests数据的前两个主成分
  - 蓝色的州名代表了前两个主成分的得分。
  - 橙色箭头表明了前两个主成分的载荷向量（数轴为图上方和右侧的轴)。例如，Rape数据的第一主成分的载荷是0.54，它的第二主成分的载荷是0.17（Rape这个词就是在点(0.54, 0.17)上）。
  - 这幅图就是**双标图(biplot)**，因为它同时显示了主成分得分和主成分载荷。

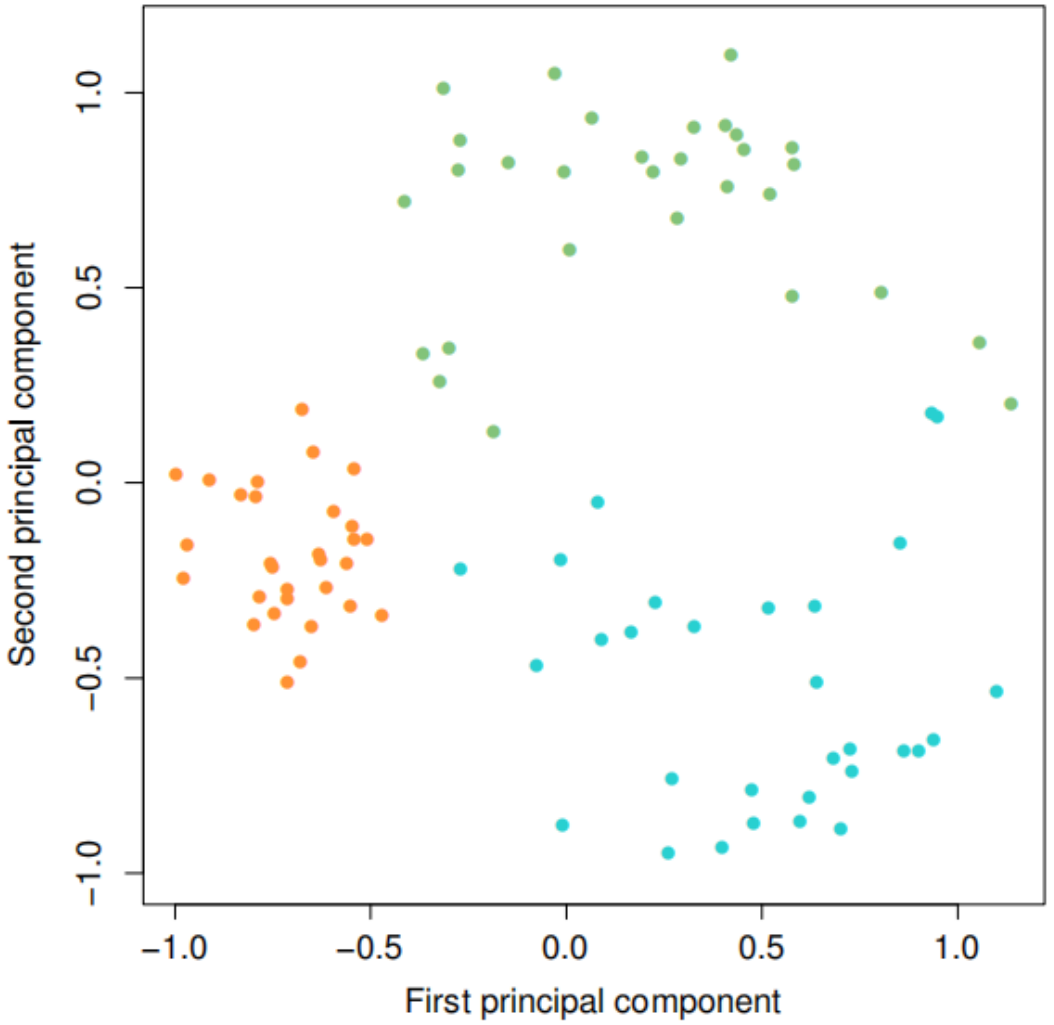
PCA载荷

- 下面是USArrests数据的主成分载荷向量ϕ_1和ϕ_2，与PCA图表示了相同的内容。

	PC1	PC2
Murder	0.5358995	-0.4181809
Assault	0.5831836	-0.1879856
UrbanPop	0.2781909	0.8728062
Rape	0.5434321	0.1673186

主成分的另一种解释



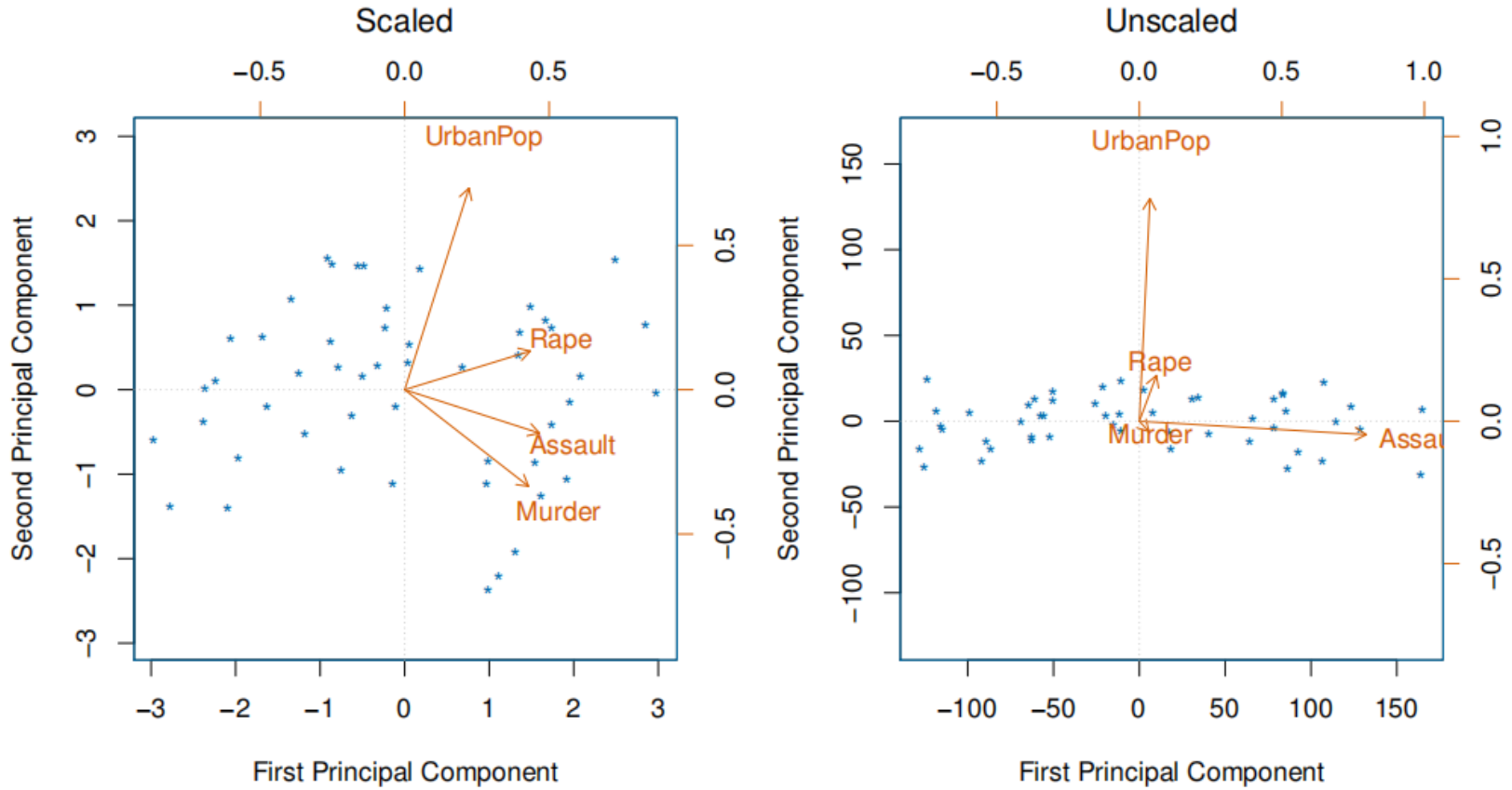

Second principal component
First principal component
−1.0
−0.5
0.0
0.5
1.0

主成分提供了一个与观测最为接近的低维线性空间

- 第一主成分载荷向量有一个特殊的性质：它是p维空间中一条**最接近**n个观测的线（用平均平方欧式距离度量接近的程度）。
- 与n个观测最接近的维度方向的主成分概念是对第一主成分内涵的延展。
- 比如，在平方欧式距离意义下，数据集的前两个主成分张成了与n个观测最接近的平面。

变量的标准化

- 主成分受变量度量单位的选择影响，会导致结果的任意性。通常在进行PCA之前，需要将每个变量都标准化，使得它们的方差都为1。
- 如果变量的度量单位相同，是否进行变量标准化均可。



方差的解释比例

- 对于PCA，我们感兴趣在一个给定的数据集中，将观测投影到前几个少数的主成分上损失了多少信息？即每个主成分的方差解释比率（proportion of variance explained, PVE）。
- 数据集（假设变量已中心化，其均值为0）的**总方差**定义如下：

$$\sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2 ,$$

- 第m个主成分的方差解释比率是：

$$\mathrm{Var}(Z_m) = \frac{1}{n} \sum_{i=1}^{n} z_{im}^2 ,$$

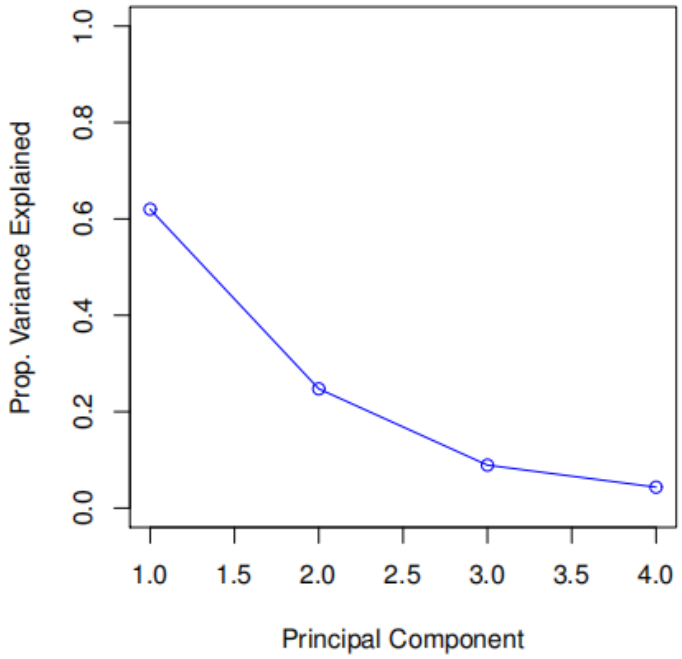
- 可以看出$\sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{m=1}^{M} \mathrm{Var}(z_m)$，其中$M = \min(n-1, p)$。

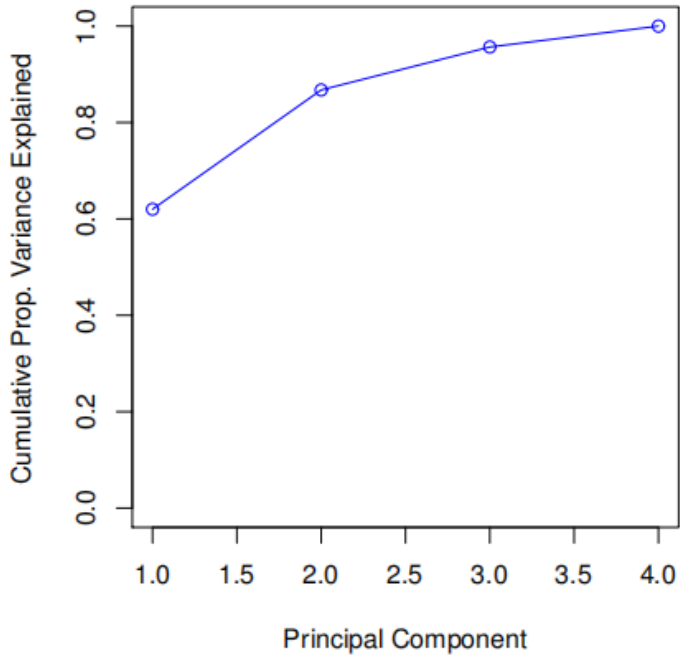
方差的解释比例

- 因此，第m个主成分的PVE可以由以下公式得到：

$$\frac{\sum_{i=1}^{n} z_{im}^2}{\sum_{j=1}^{p}\sum_{i=1}^{n} x_{ij}^2},$$

- 每个主成分的PVE都是正值，在0到1之间。一共有$\min(n-1, p)$个主成分，它们的PVE和是1。有时我们展示的是累计方差解释率。





决定主成分的数量

- 我们希望用最少量的主成分来形成对数据的一个很好的理解。那么到底需要多少个主成分呢?
- 这个问题没有唯一的（或者说简单的）答案，因为这里我们并不能使用交叉验证
  - **为什么?** （因为是无监督过程，没有响应变量Y）

决定主成分的数量

- 我们希望用最少量的主成分来形成对数据的一个很好的理解。

  那么到底需要多少个主成分呢?

- 这个问题没有唯一的（或者说简单的）答案，因为这里我们并不能使用交叉验证
  - **为什么?**
- 通常可以通过看碎石图(scree plot)来决定所需的主成分数量:

  寻找碎石图的肘(elbow)

主成分分析-R实验

对USArrests数据集进行PCA，数据集中的行包含50个州。

```
> data("USArrests")
> states=row.names(USArrests)
> states
 [1] "Alabama"        "Alaska"         "Arizona"        "Arkansas"
 [5] "California"     "Colorado"       "Connecticut"    "Delaware"
 [9] "Florida"        "Georgia"        "Hawaii"         "Idaho"
[13] "Illinois"       "Indiana"        "Iowa"           "Kansas"
[17] "Kentucky"       "Louisiana"      "Maine"          "Maryland"
[21] "Massachusetts"  "Michigan"       "Minnesota"      "Mississippi"
[25] "Missouri"       "Montana"        "Nebraska"       "Nevada"
[29] "New Hampshire"  "New Jersey"     "New Mexico"     "New York"
[33] "North Carolina" "North Dakota"   "Ohio"           "Oklahoma"
[37] "Oregon"         "Pennsylvania"   "Rhode Island"   "South Carolina"
[41] "South Dakota"   "Tennessee"      "Texas"          "Utah"
[45] "Vermont"        "Virginia"       "Washington"     "West Virginia"
[49] "Wisconsin"      "Wyoming"
```

主成分分析-R实验

粗略分析数据后发现变量的均值和方差之间存在较大差异。

```
> names(USArrests)
[1] "Murder"   "Assault"  "UrbanPop" "Rape"
> apply(USArrests, 2, mean)
  Murder  Assault UrbanPop     Rape
   7.788  170.760   65.540   21.232
> apply(USArrests, 2, var)
    Murder    Assault   UrbanPop       Rape
  18.97047 6945.16571  209.51878   87.72916
```

prcomp ( ) 函数可进行主成分分析，函数默认对变量进行中心化处理，选项scale=TURE可对变量进行标准化处理。

```
> pr.out=prcomp(USArrests, scale=TRUE)
```

prcomp函数的输出包括许多可用作继续分析的量。

```
> names(pr.out)
[1] "sdev"     "rotation" "center"   "scale"    "x"
```

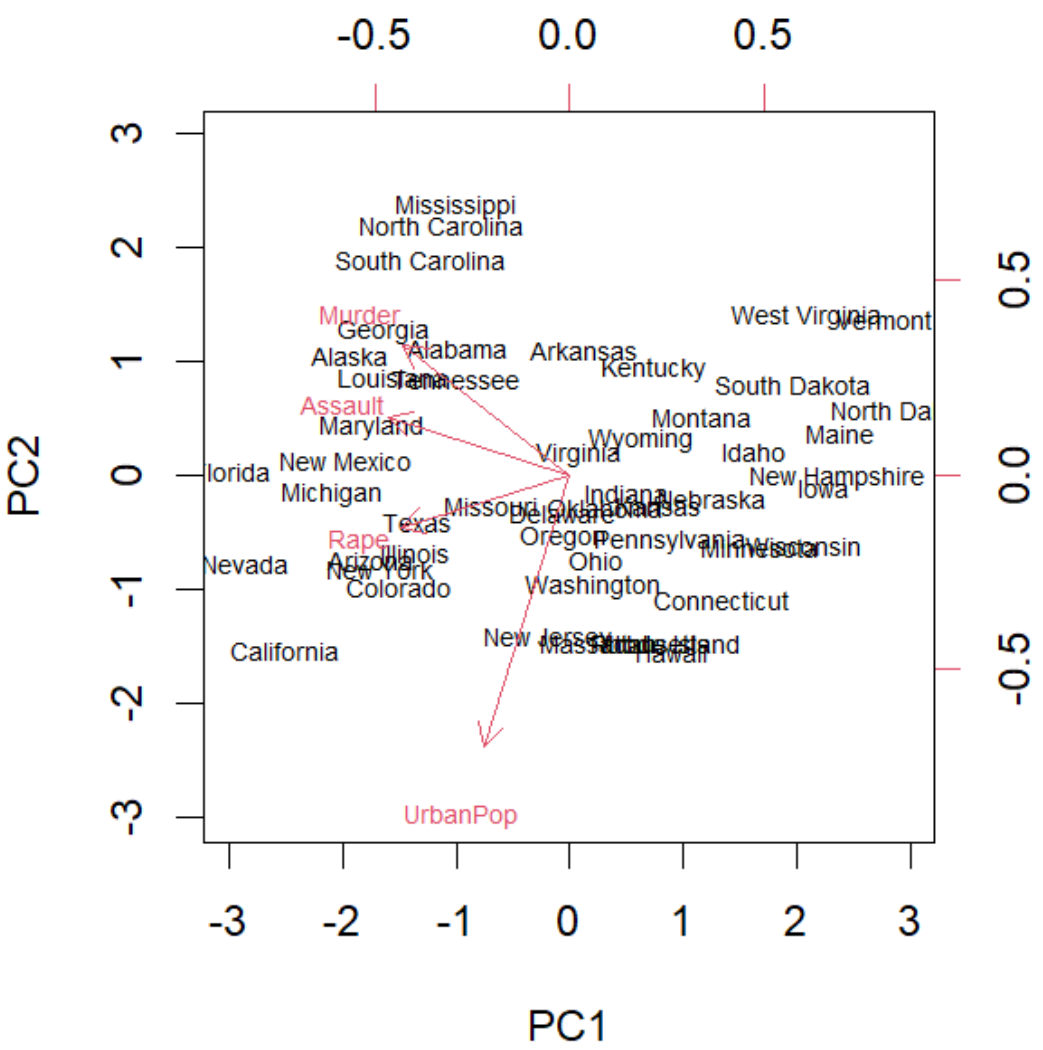
sdev可给出标准差，center和scale表示在实施PCA之前进行标准化以后变量的均值和标准差，rotation矩阵提供了主成分载荷信息

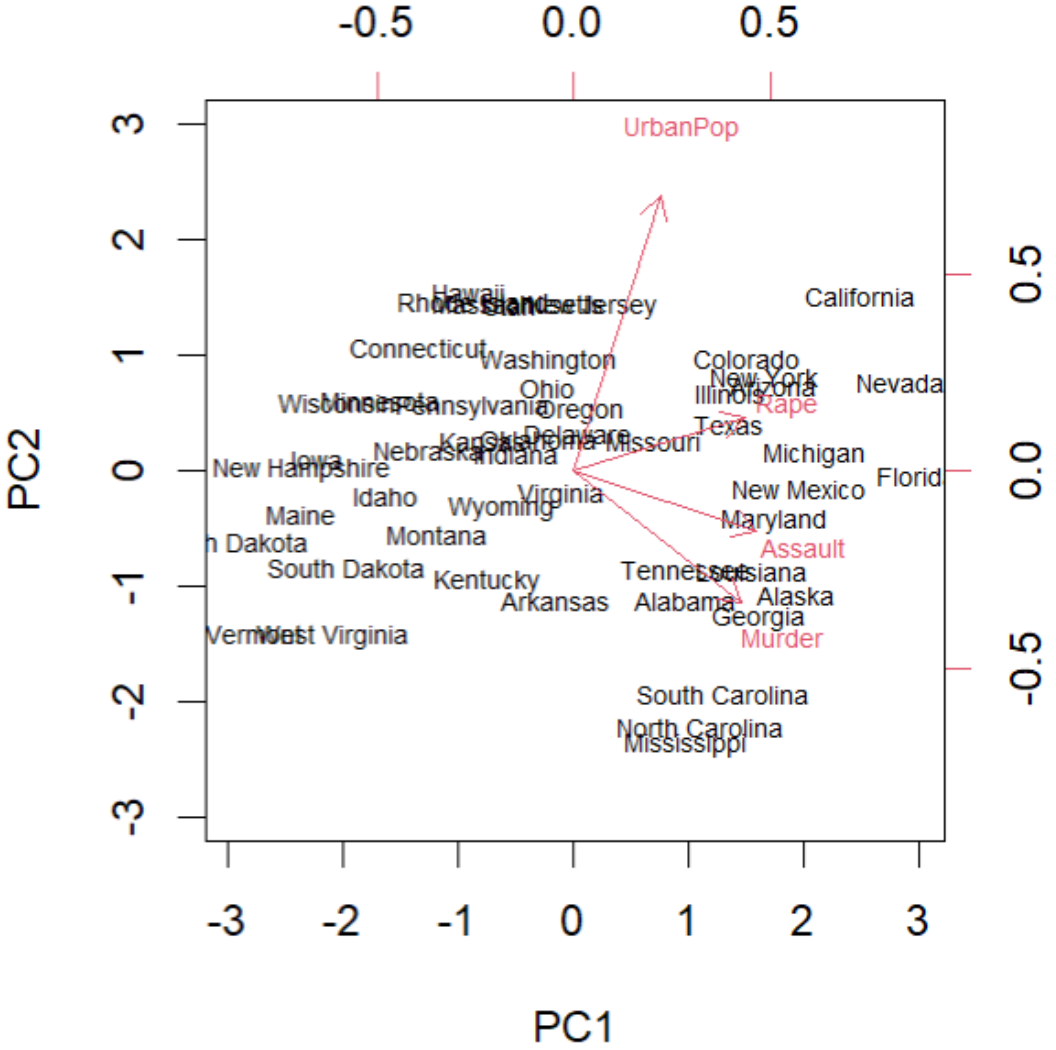
```
> pr.out$sdev
[1] 1.5748783 0.9948694 0.5971291 0.4164494

> pr.out$center
  Murder  Assault UrbanPop     Rape
   7.788  170.760   65.540   21.232
> pr.out$scale
    Murder   Assault  UrbanPop      Rape
 4.355510 83.337661 14.474763  9.366385
> pr.out$rotation
                PC1        PC2        PC3         PC4
Murder   -0.5358995  0.4181809 -0.3412327  0.64922780
Assault  -0.5831836  0.1879856 -0.2681484 -0.74340748
UrbanPop -0.2781909 -0.8728062 -0.3780158  0.13387773
Rape     -0.5434321 -0.1673186  0.8177779  0.08902432
```

这样的列为包含对应的主成分载荷向量

主成分分析-R实验


```
> biplot(pr.out, scale=0)
> pr.out$rotation=-pr.out$rotation
> pr.out$x=-pr.out$x
> biplot(pr.out, scale=0)
```





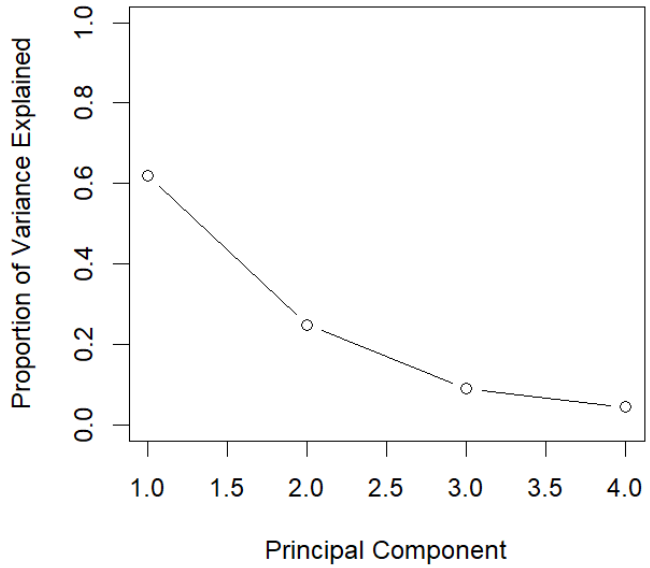
主成分分析-R实验

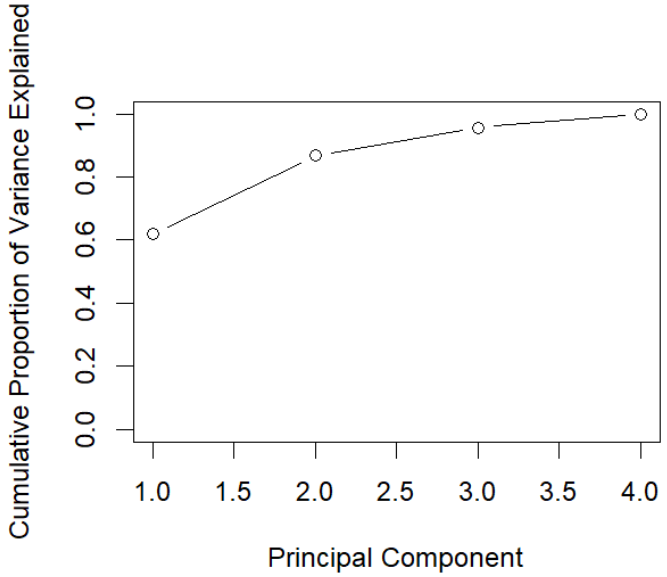
计算主成分解释的方差以及计算每个主成分的方差解释比例

```
> pr.var=pr.out$sdev^2
> pr.var
[1] 2.4802416 0.9897652 0.3565632 0.1734301
> pve=pr.var/sum(pr.var)
> pve
[1] 0.62006039 0.24744129 0.08914080 0.04335752
```

cumsum ( ) 函数可用于计算数值向量中的元素的累计和。

```
> plot(pve, xlab="Principal Component", ylab="Proportion of Variance Explained",
  ylim=c(0,1),type='b')
> plot(cumsum(pve), xlab="Principal Component", ylab="Cumulative Proportion of V
ariance Explained", ylim=c(0,1),type='b')
```





聚类分析方法

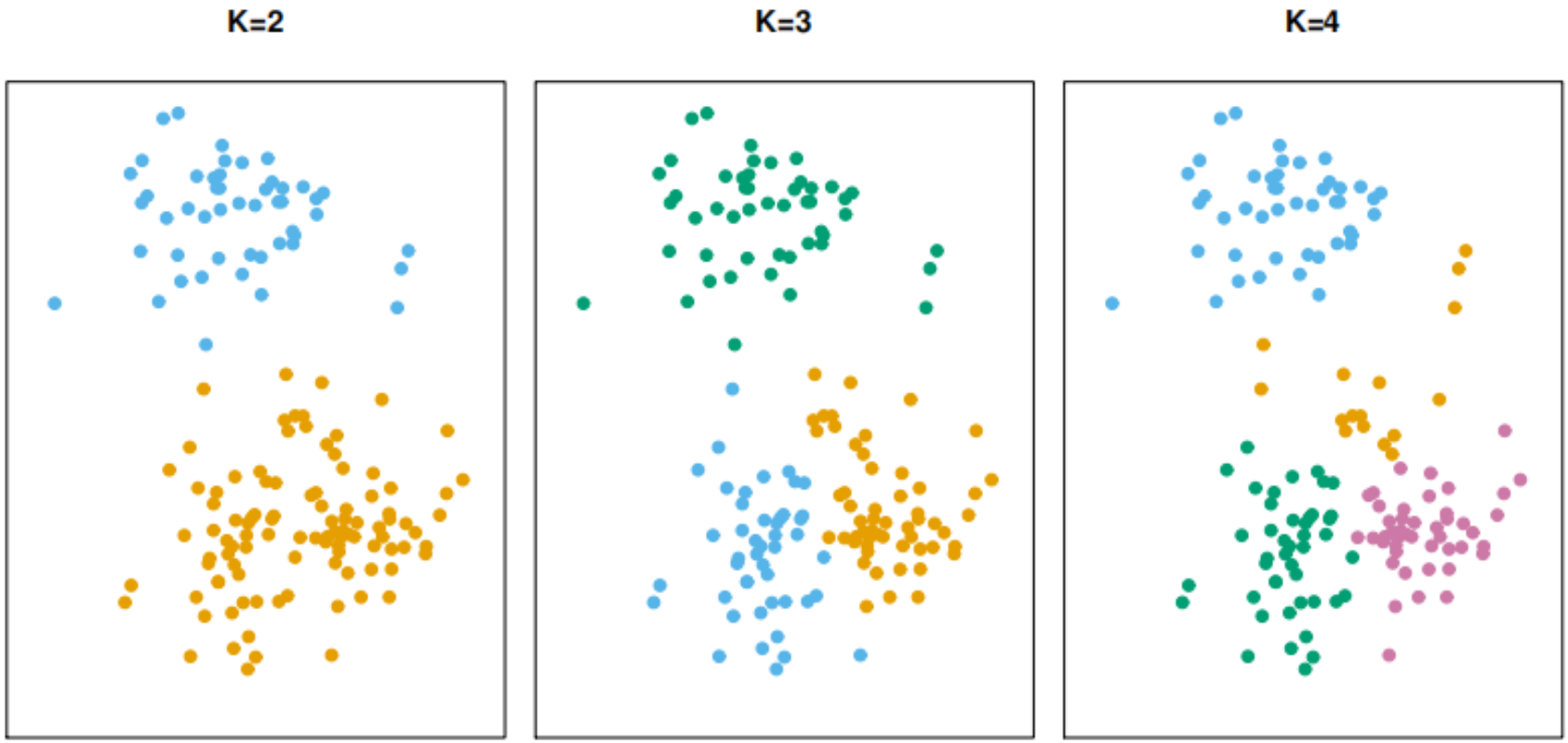
- 聚类分析(clustering)是在一个数据集中寻找**子群(subgroups)**或**类(clusters)**的技术，应用非常广泛。
- 希望将数据分割到不同的类中，使每个类内的观测彼此非常相似。
- 为了确切地表达相似的概念，必须对2个或更多观测的**相似(similar)**或者**相异(different)**进行定义。
- 这个问题则必须结合问题所来源的特殊背景经数据分析后方能获得答案。
- **PCA与聚类分析比较：**
  - PCA试图寻找观测的一个低维表示来解释大部分方差；
  - 聚类分析试图从观测中寻找同质子类。

聚类分析应用于市场细分

- 市场研究中常常可以获得有关人的大量观测数据（例如家庭收入中位数、职业、两个城市的最短距离等）。
- **市场细分(market segmentation)**的一个目的是通过识别更倾向于接受某特定形式的广告或者说更可能购买特定产品的人群进行市场分割。
- 这相当于用所得到的数据集对不同的人进行聚类。

- **K均值聚类(K-means clustering)**试图将观测划分到事先规定数量的类中。
- **系统聚类(hierarchical clustering)**并不需要事先规定所需的类数。我们最后会通过分析观测的树型表示，即**谱系图(dendrogram)**来确定类数。通过看谱系图还可以马上获得从1类到n类类数不等的分类情况。

两种聚类算法：K均值聚类



- 一个在二维空间中包含150个观测的模拟数据集。这些图展示了用不同的K值进行K均值聚类法的结果，其中K是指类数。每个观测的颜色表明它们在K均值聚类过程中被分到哪个类中。注意：类是没有顺序的，所以类的颜色是任意的。聚类分析并没有用到这些类标签，它们反而是这个聚类过程的输出结果。

K均值聚类细节

- 用$C_1, \ldots, C_k$表示在每个类中包含观测指标的集合。这些集合满足两个性质：
  - （1）$C_1 \cup C_2 \cup \cdots \cup C_K = \{1, \ldots, n\}$。即每个观测属于$K$个类中至少一个类。
  - （2） $C_k \cap C_{k'} = \emptyset$对每个$k \neq k'$都成立。即类与类之间是无重叠的：没有一个观测同时属于两个类或更多类。
- 例如，如果第i个观测在第k个类中，则$i \in C_k$

K均值聚类细节

- K均值聚类法的思想是，一个好的聚类法可以使**类内差异(within-cluster variation)**尽可能小。
- 第C_k类的类内差异是，对第C_k类中观测互不相同程度的度量$W(C_k)$。因此需要解决如下最小化问题：

$$\text{minimize}_{C_1,\ldots,C_k}\left\{\sum_{k=1}^{K} W(C_k)\right\}$$

- 这个公式的意思是把观测分割到K个类中，使得K个类总的类内差异尽可能小。

定义类内差异

- 通常使用平方欧式距离定义：

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2$$

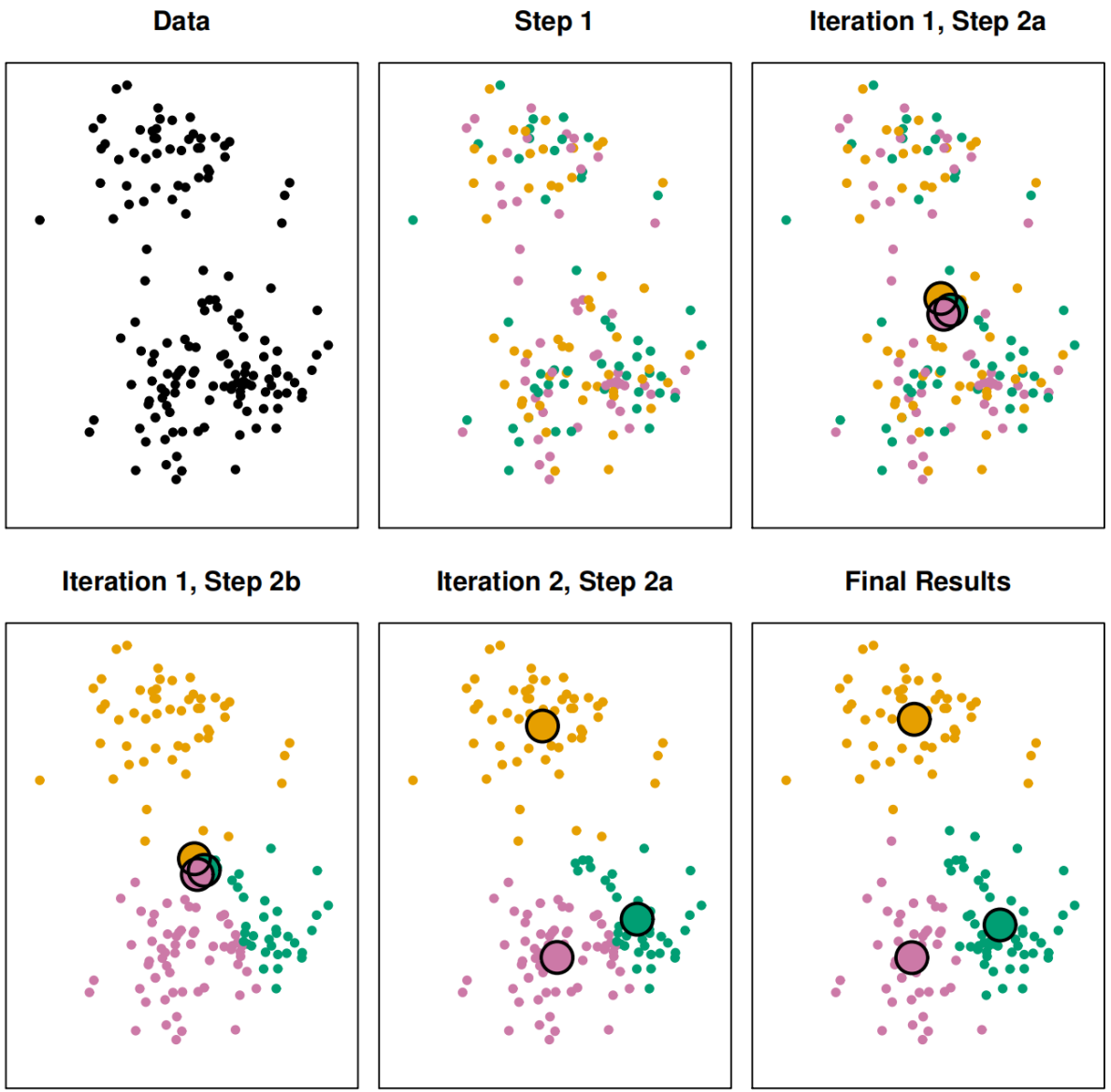
- 这里的$|C_k|$表示在第k个类中观测的数量。
- 结合前一页的最小化目标函数，可以得到定义K均值聚类法的最优化问题：

$$\text{minimize}_{C_1,\ldots,C_k} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2 \right\}$$

算法

- 1. 为每个观测随机分配一个从1到K的数字。这些数字可以看作对这些观测的初始类。
- 2. 重复下列操作，直到类的分配停止为止：
  - 2.1 分别计算K个类的类中心。第k个类**中心(centroid)**是第k个类中的p维观测向量的均值向量。
  - 2.2 将每个观测分配到距离其最近的类中心所在的类中（用欧式距离定义“最近(closest)”）。

聚类例子


Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results

图的解释

该图是K=3时的K均值聚类过程。

- **左上：**原始观测点。
- **上排中间：**算法的第1步，每个观测被随机分配到一个类中。
- **右上：**第2.1步中类中心的计算，这些类中心在图中用彩色大圆表示。可以看到，因为K均值聚类的初始类分配是随机的，所以初始的类中心几乎完全重叠。
- **左下：**在第2.2步中，每个观测被分配到了与之最近的类中。
- **下排中间：**第2.1步再次执行后得到了一个新的类中心。
- **右下：**10次迭代后得到的结果。

例子：用不同初始类进行聚类

图的解释

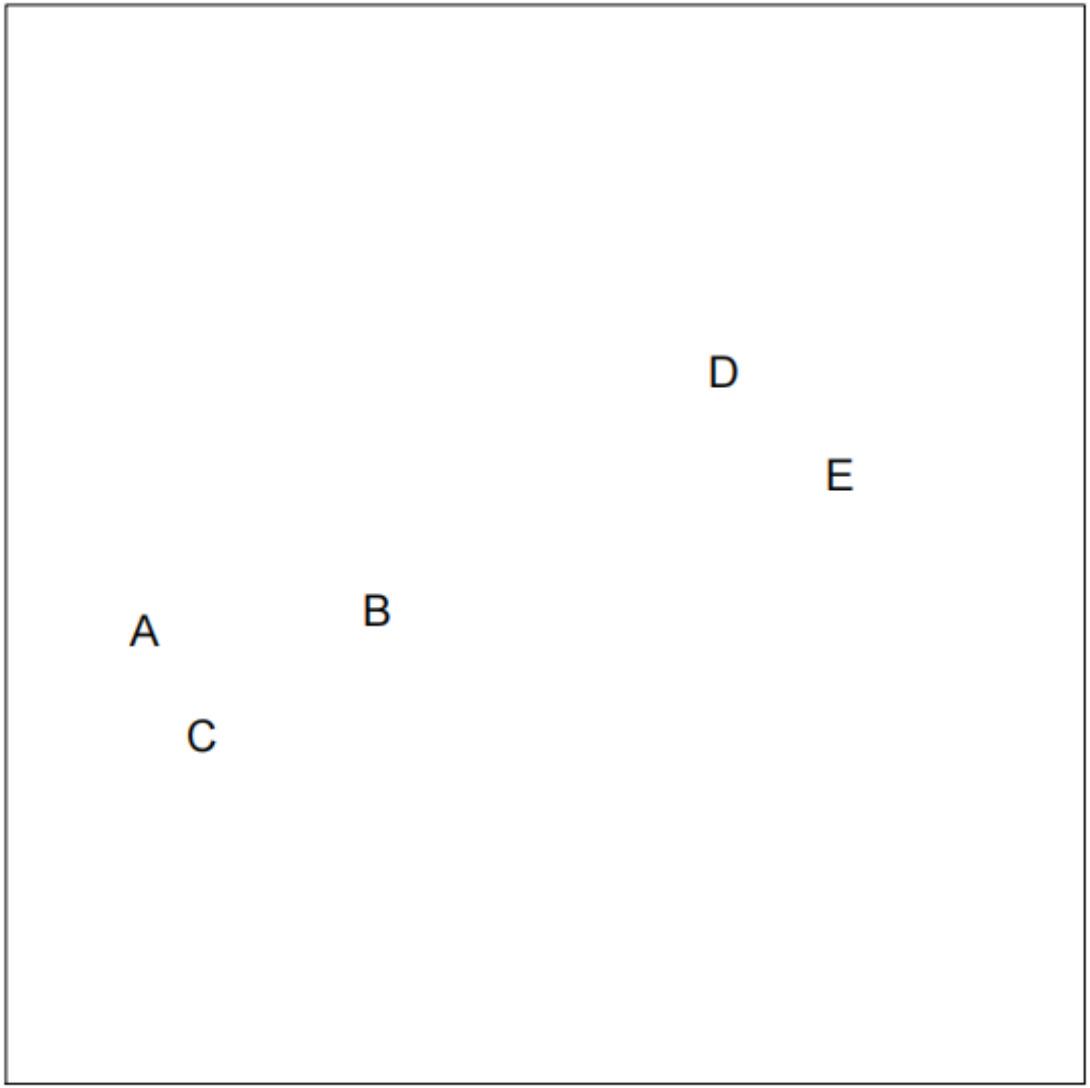
- 对上例中的数据运行6次K=3的K均值聚类法后得到的结果。在每次运行时，K均值聚类算法的第1步都会给每个观测随机分配一个不同的初始类。
- 每幅图上方的数字表示聚类后目标函数（即类内距离）的一个值。
- 我们得到了3种不同的局部最优解，其中一种局部最优解的目标值相对较小且提供了各类之间相对较好的划分。
- 图上方数字为红色的聚类都实现了一致的最优分配，它们共同的目标值都235.8

两种聚类算法：系统聚类法

- K均值聚类法的一个潜在的不足是它需要预设类数K。
- **系统聚类法(Hierarchical clustering)**是一种事先不需要规定类数K的聚类方法。
- 这里将介绍**自下而上的(bottom-up)**也称为**凝聚法(agglomerative)**的聚类方法。这是一种最为常见的系统聚类法，它的谱系图（通常被形容为一棵上下颠倒的树）从叶子开始将聚类集到树干上。

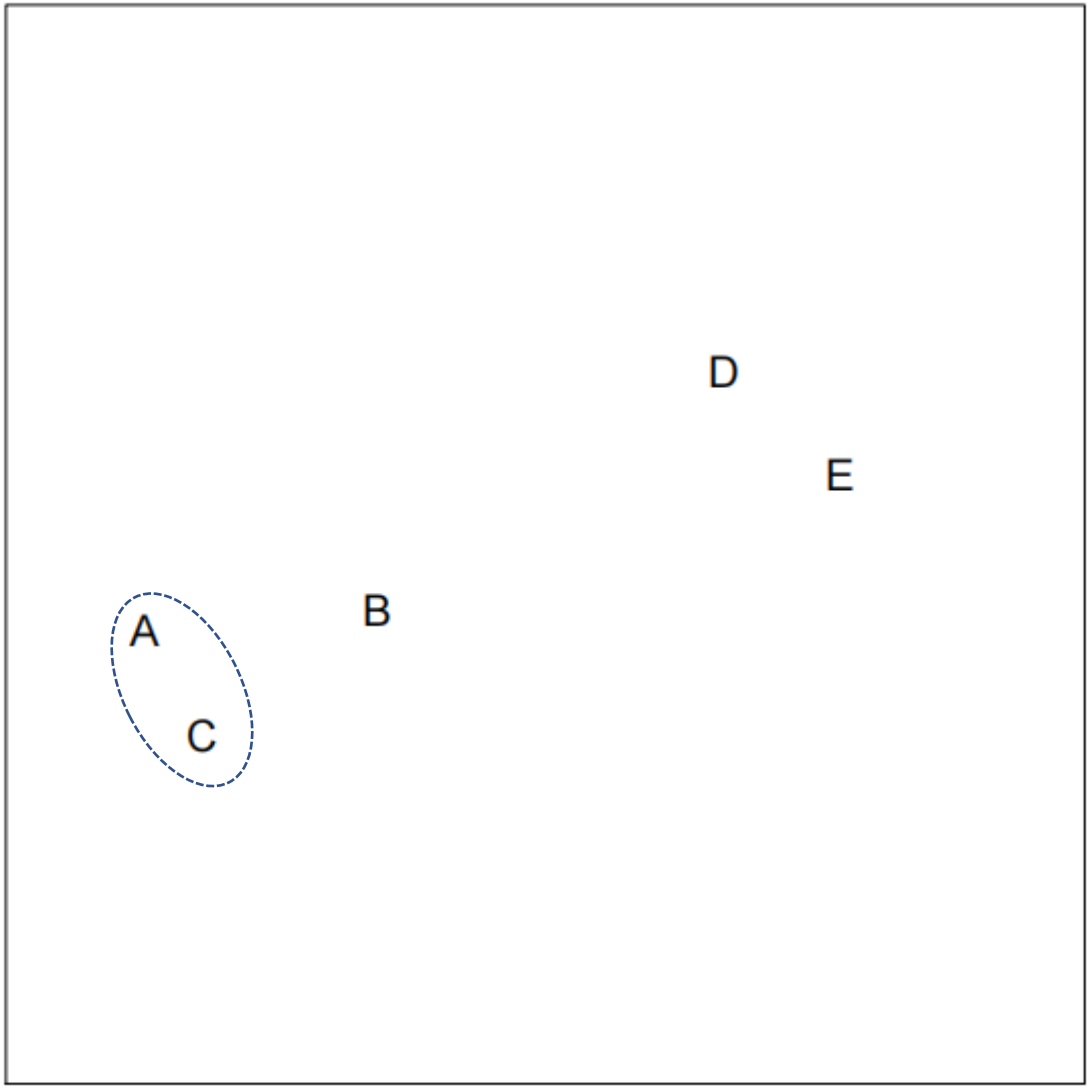
思想

- 用自下而上的方法进行系统聚类



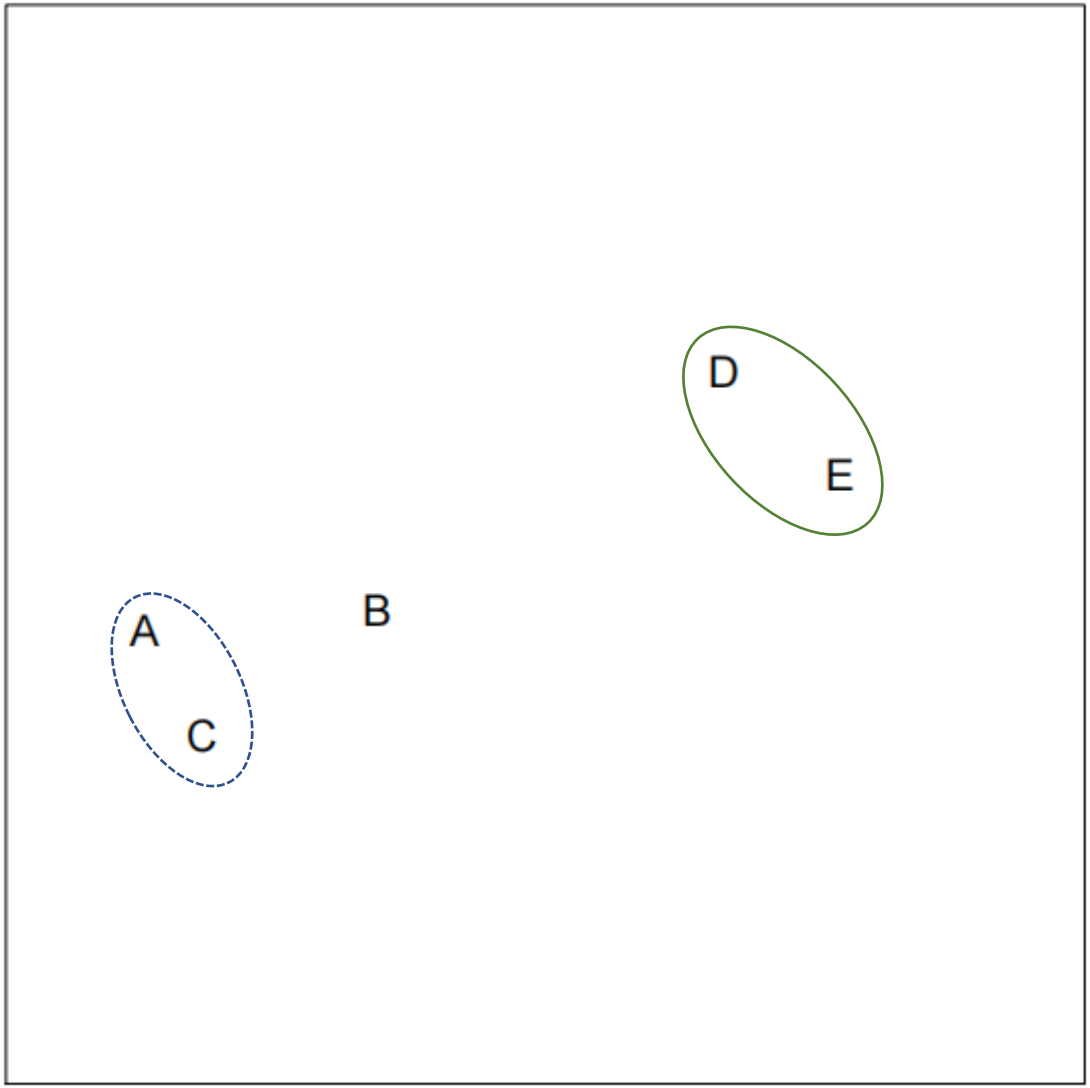
思想

- 用自下而上的方法进行系统聚类



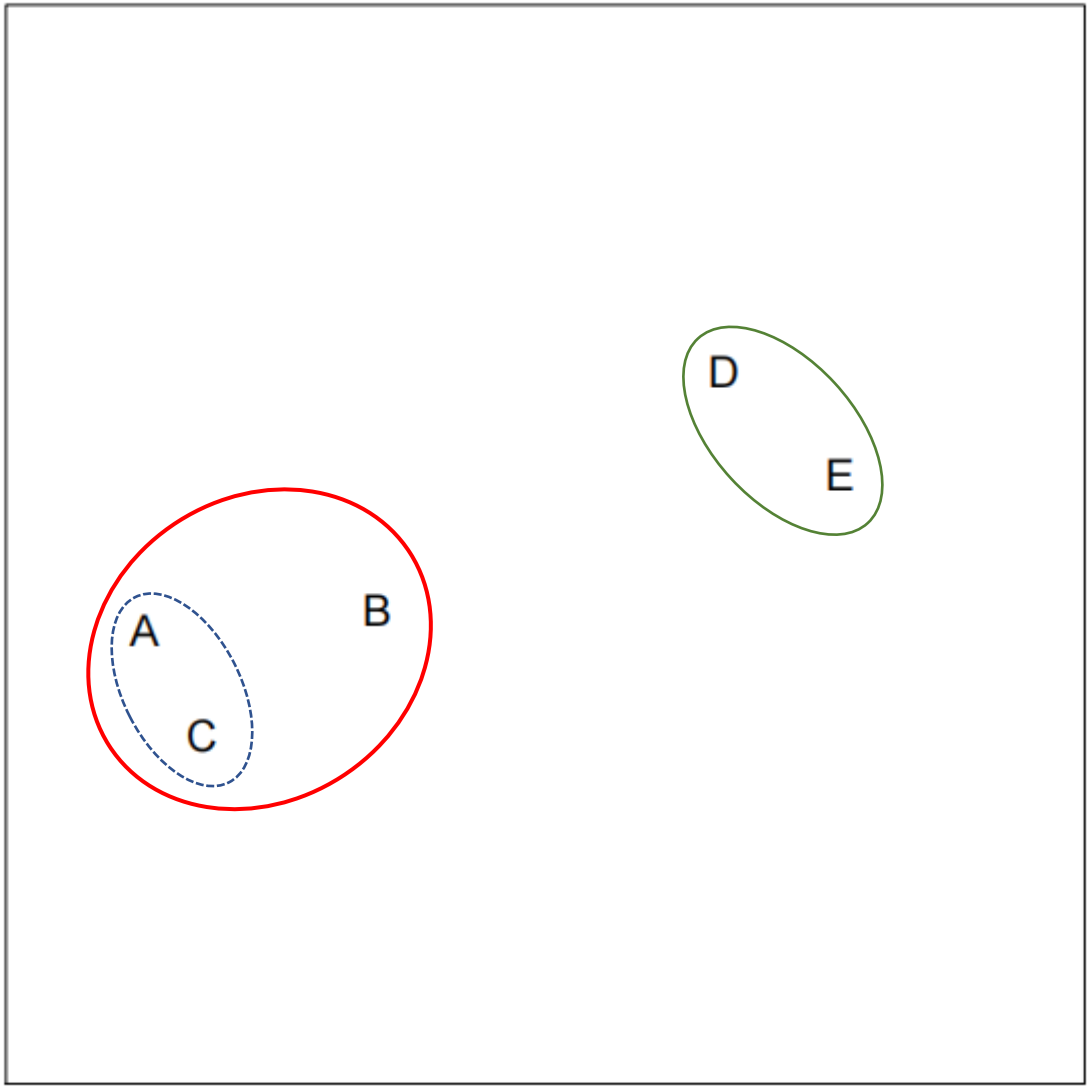
思想

- 用自下而上的方法进行系统聚类



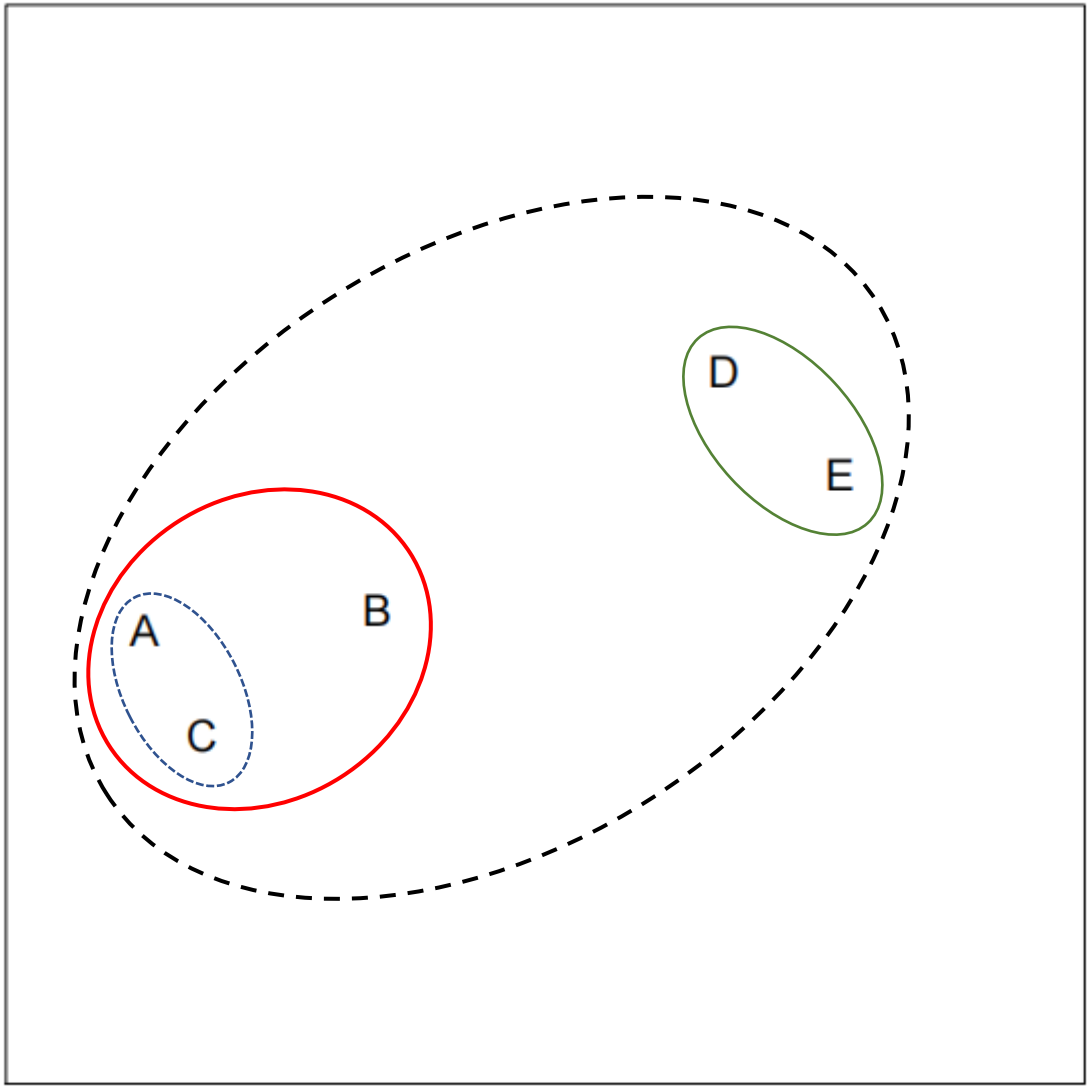
思想

- 用自下而上的方法进行系统聚类



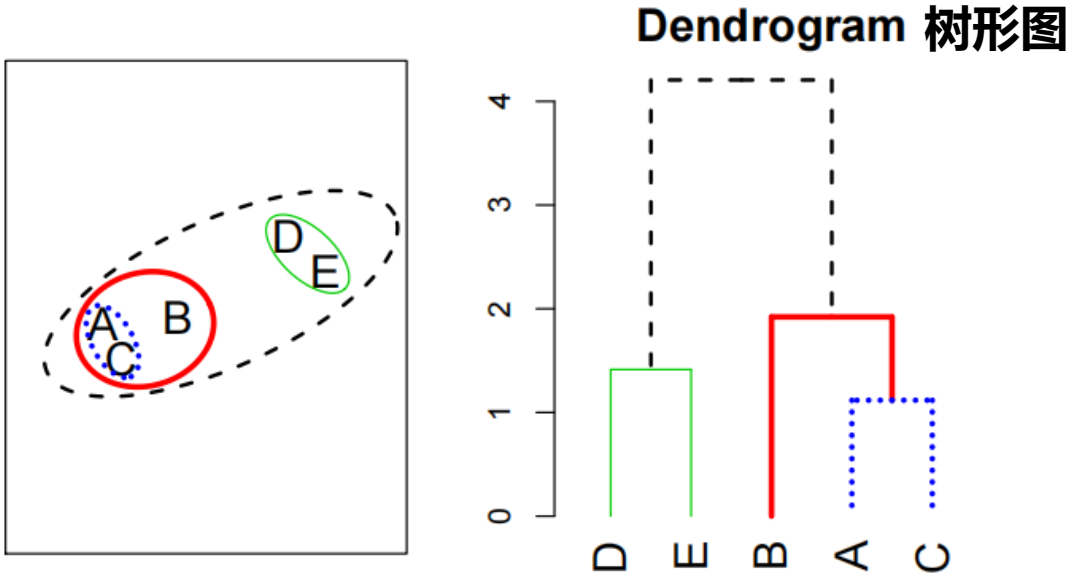
思想

- 用自下而上的方法进行系统聚类

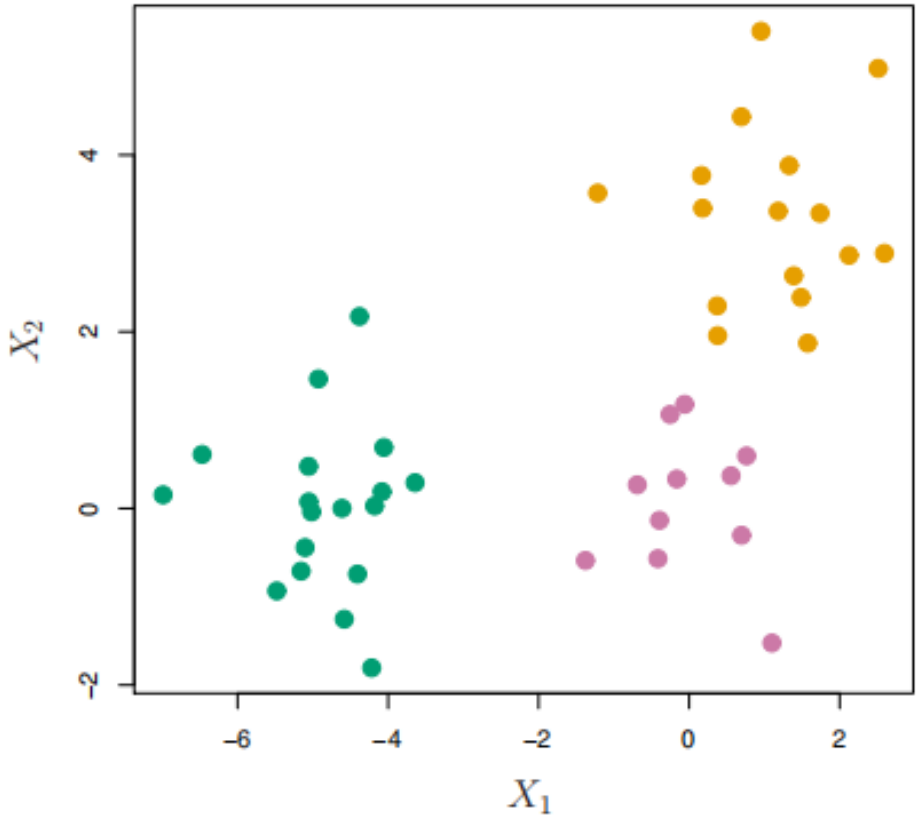


算法

- 从谱系图的底部开始，n个观测各自都被看作一类
- 再将两个**最为相似**的类汇合到一起，就得到了n-1个类；
- 然后再把两个**最为相似**的类汇合到一起，就得到了n-2个类；
- 如此进行下去，到所有观测都属于某一个类时停止

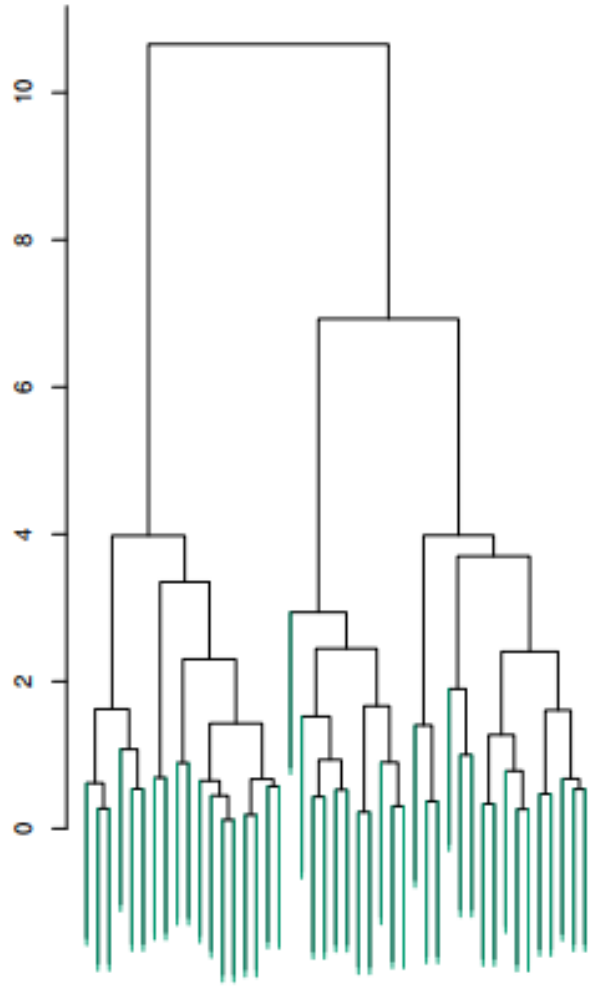


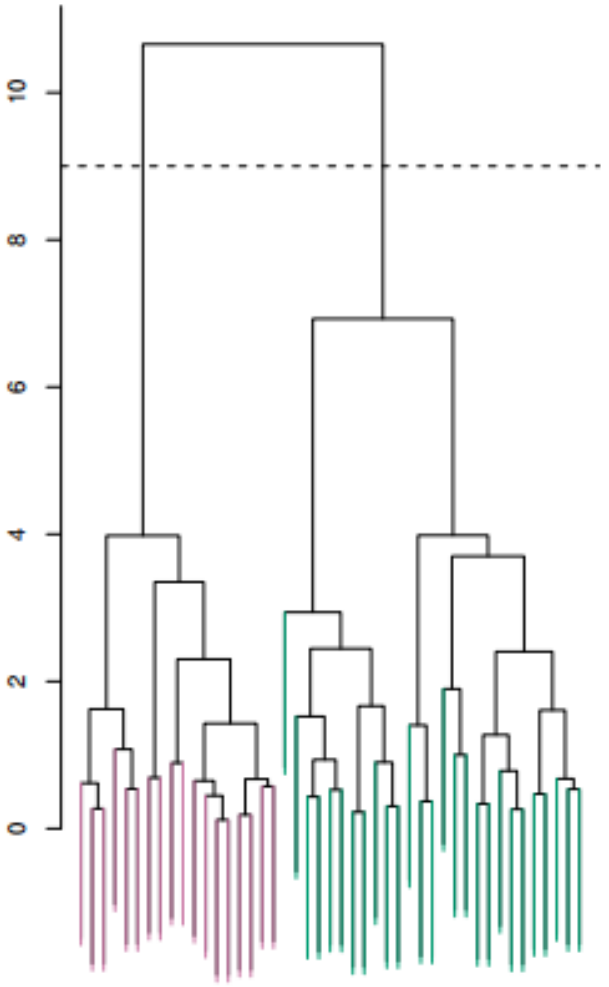
示例

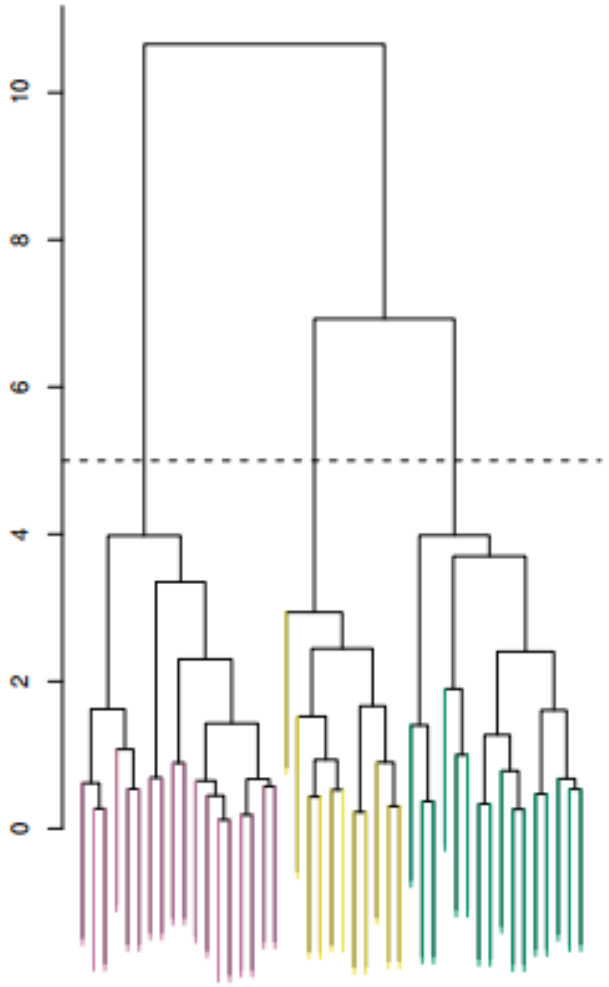


- 在二维空间中产生了45个观测，事实上，它们分属于3个不同的类，用3种不同的颜色表示，先将这些类别看作未知，然后尝试着对这些观测聚类寻找数据各自的归类。

应用







图的解释

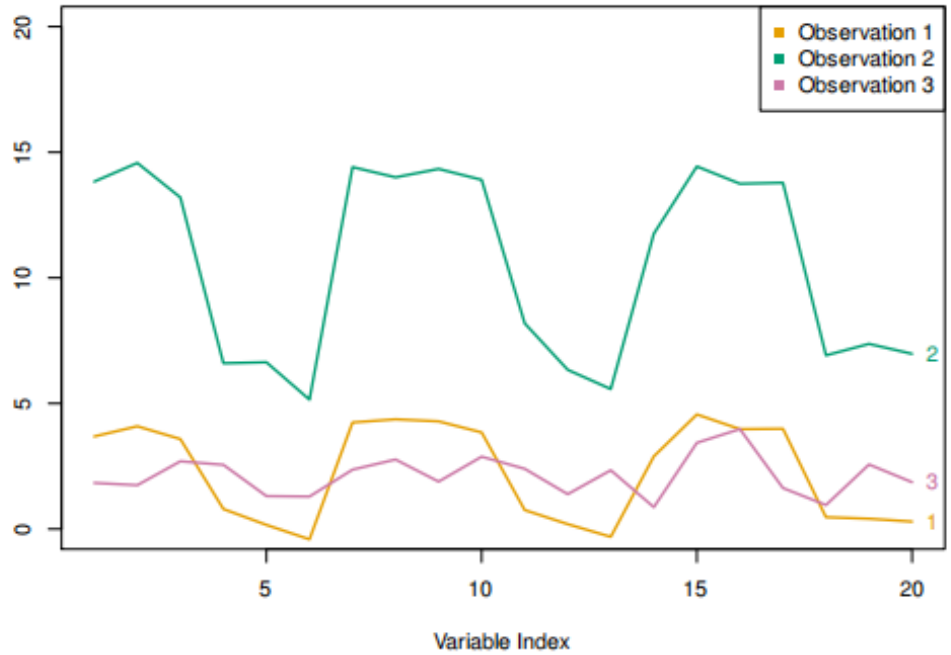
- **左：**对45个观测数据用最长距离法和欧式距离进行系统聚类法得到的谱系图。
- **中：**在左图中高度为9时切断（用虚线表示），切断后得到2个不同的类，用不同颜色表示。
- **右：**在左图中高度为5时切断，切断后得到3个不同的类，用不同的颜色表示。注：在聚类过程中并没有用到这些颜色，用不同颜色表示不同类只是为了在这幅图中更好地展示数据。

4种常用的距离形式

距离方式	描述
最长距离法（Complete）	最大类间相异度。计算A类和B类之间的所有观测的相异度，并记录**最大**的相异度。
最短距离法（Single）	最小类间相异度。计算A类和B类之间的所有观测之间的相异度，并记录**最小**的相异度。最短距离法会导致观测一个接一个地汇合延伸拖尾的类。
类平均法（Average）	平均类间相异度。计算A类和B类之间的所有观测的相异度，并记录这些相异度的**平均值**。
重心法（Centroid）	A类中心（长度为p的均值向量）和B类中心的相异度。重心法会导致一种不良的**倒置现象**的发生。

系统聚类法：相异度指标的选择

- 目前我们使用的相异度指标都是欧氏距离
- 另一种可选的方法是基于相关性的距离(correlation-based distance)，它用相关性的距离去度量两个观测的相似性。当两个观测之间高度相关时，可以考虑它。
- 这并非相关系数的常规用法，因为它用于变量间相关性的计算。但在这里，它计算的是每对观测的观测剖面之间的相关性。



聚类分析的实践问题

- 聚类分析之前，需要先回答如下问题。
- 观测或变量需要先经过某种**标准化处理**吗？比如，变量中心化为均值为0，或标准化为标准差为1;
- 系统聚类法中的问题：
  - 用什么指标度量相异度?
  - 选择怎样的距离计算距离?
  - 在谱系图的哪个位置切割出不同的类?
- 在K均值聚类法中，数据分成多少类比较合适？（并没有标准答案）
- 使用什么特性(features)用于聚类?

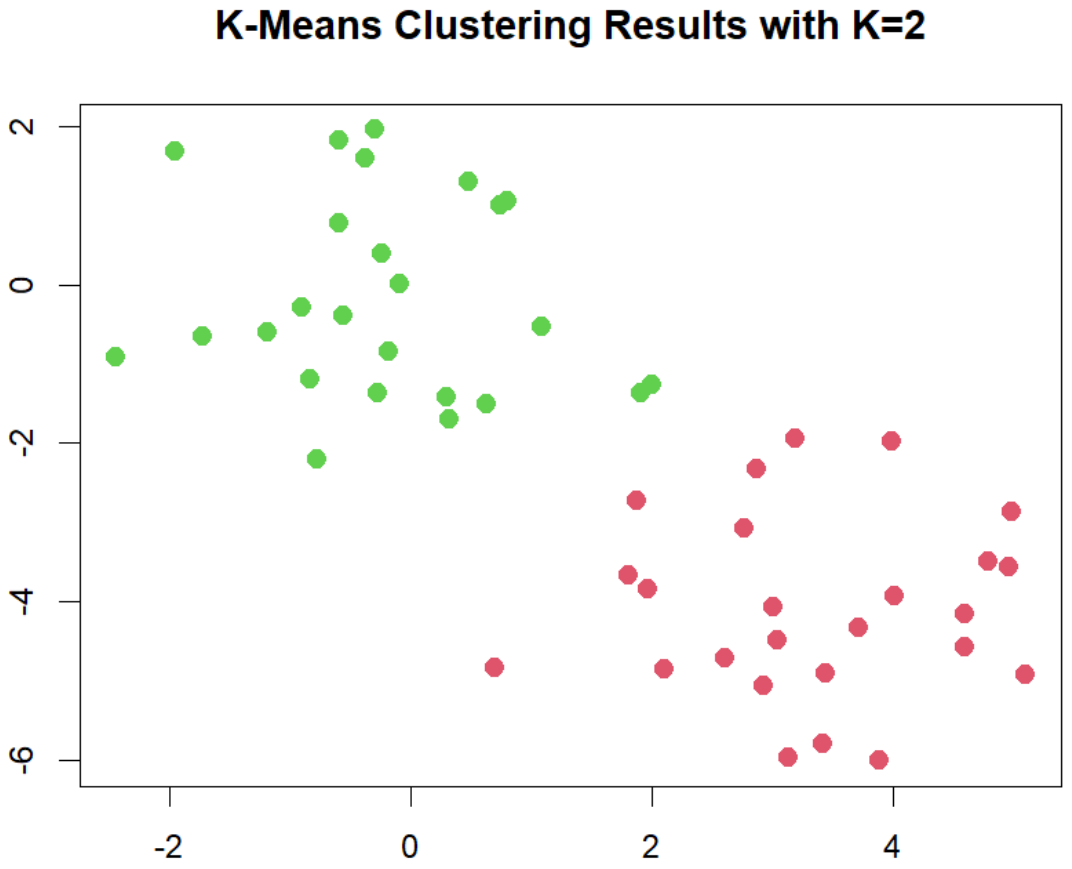
1、K均值聚类

在R中，kmeans( )函数用于执行k均值聚类，以下是一个模拟案例。

```
> set.seed(2)
> x=matrix(rnorm(50*2), ncol=2)
> x[1:25,1]=x[1:25,1]+3
> x[1:25,2]=x[1:25,2]-4
> km.out=kmeans(x,2,nstart=20)
> km.out$cluster
 [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
[26] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

1、K均值聚类

```
> plot(x, col=(km.out$cluster+1), main="K-Means Clustering
 Results with K=2", xlab="", ylab="", pch=20, cex=2)
```



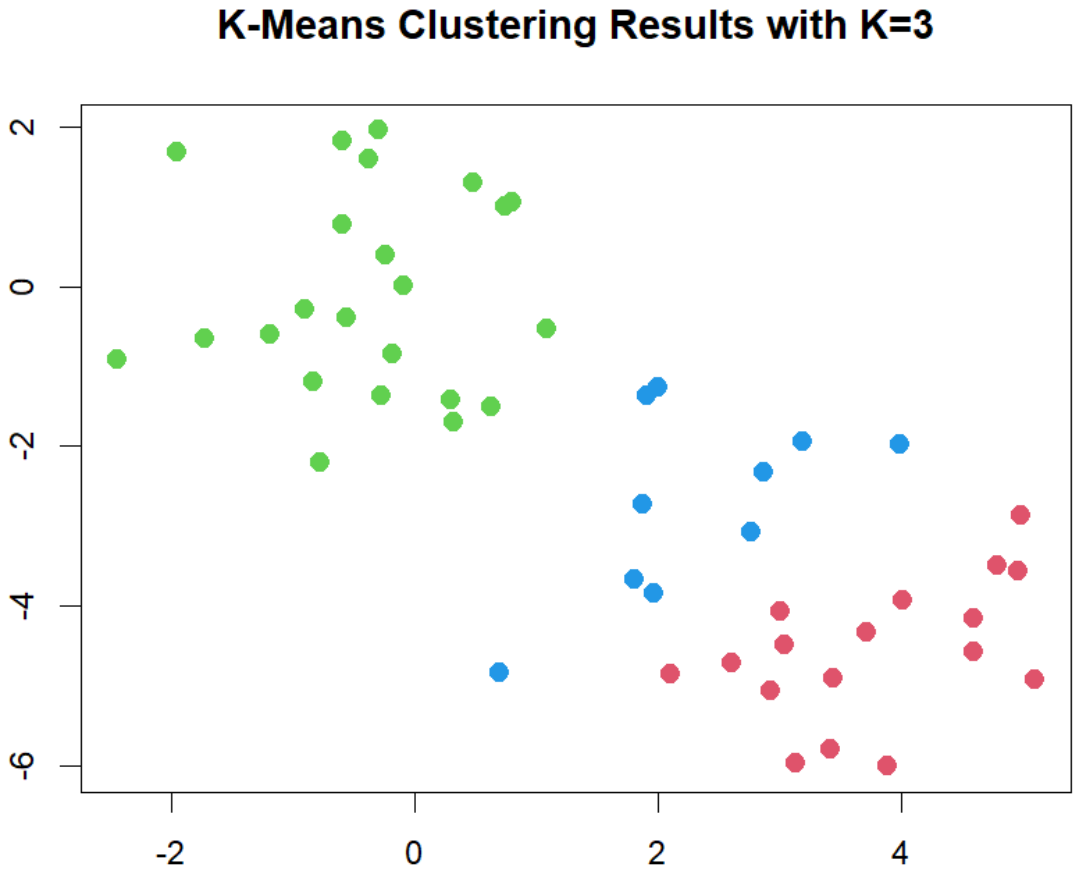
1、K均值聚类

在这个例子中对数据进行K=3的K均值聚类

```
> set.seed(4)
> km.out=kmeans(x,3,nstart=20)
> km.out
K-means clustering with 3 clusters of sizes 17, 23, 10

Cluster means:
        [,1]        [,2]
1  3.7789567 -4.56200798
2 -0.3820397 -0.08740753
3  2.3001545 -2.69622023

Clustering vector:
 [1] 1 3 1 3 1 1 1 3 1 3 1 3 1 3 1 3 1 1 1 1 1 3 1 1 1 2 2 2 2 2 2 2 2 2 2
[36] 2 2 2 2 2 2 2 2 3 2 3 2 2 2 2

Within cluster sum of squares by cluster:
[1] 25.74089 52.67700 19.56137
 (between_SS / total_SS =  79.3 %)

Available components:

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

1、K均值聚类

```
> plot(x, col=(km.out$cluster+1), main="K-Means Clustering Results with K=3",
  xlab="", ylab="", pch=20, cex=2)
```



1、K均值聚类

在kmeans( )函数执行中试验初始类的各种分配情况，可用nstart参数。若nstart的值比已用过的值大，那么R软件将会用书中算法10.1中多种随机分配情况运行K均值聚类，而kmeans( )函数将只记录最优的聚类结果。以下是nstart=1以及nstart=20时的情况。

```
> set.seed(3)
> km.out=kmeans(x,3,nstart=1)
> km.out$tot.withinss
[1] 97.97927
> km.out=kmeans(x,3,nstart=20)
> km.out$tot.withinss
[1] 97.97927
```

表示总的类内平方和，可通过K均值聚类将其最小化，其中每个类的组内平方和也包含在km.out$tot.withinss向量中。

2、系统聚类法

在R中，hclust( )函数用于执行系统聚类。首先用最长距离法作为距离方式对观测进行系统聚类。dist( )函数用于计算50×50观测间欧式距离的矩阵。

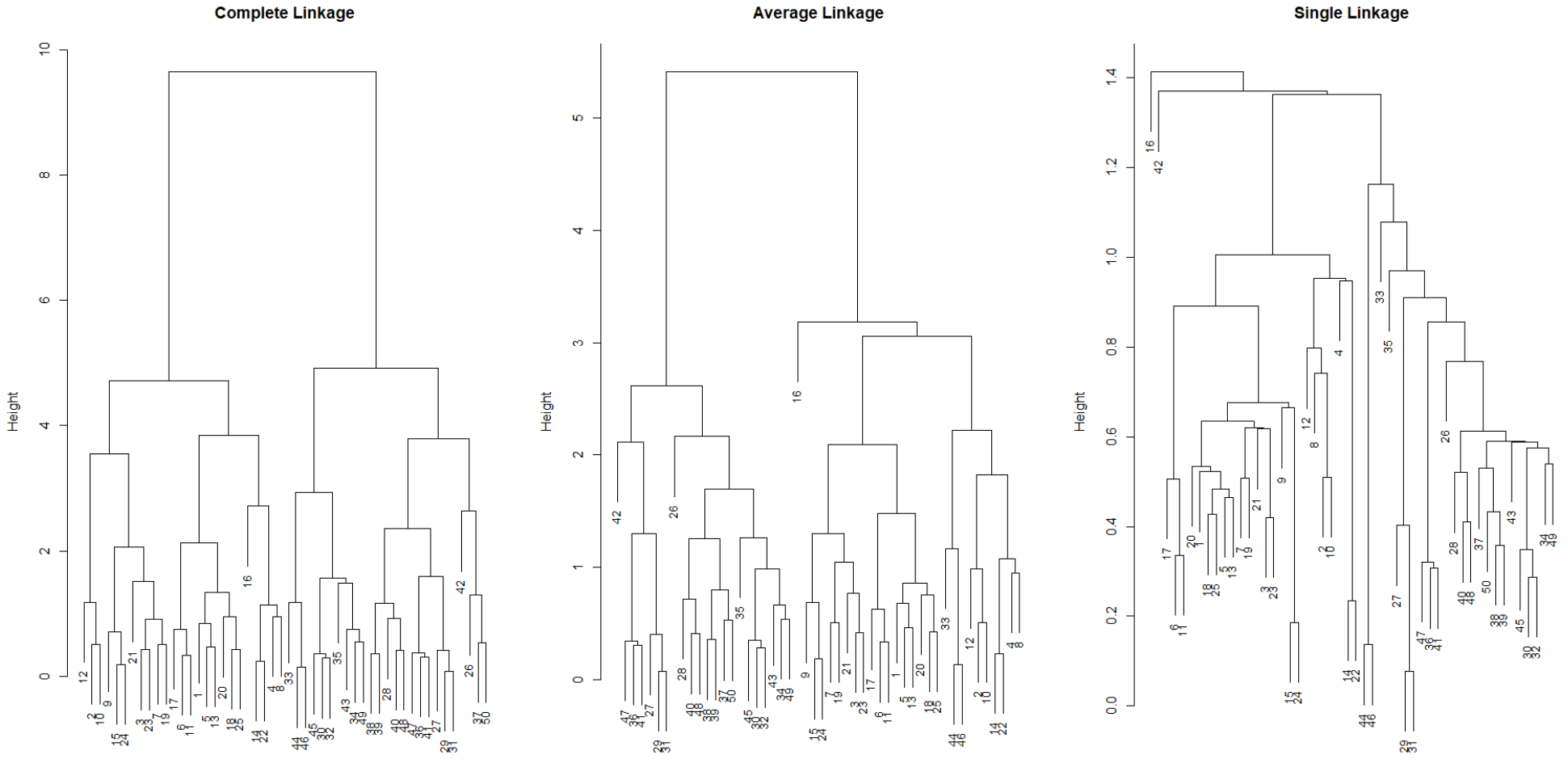
```
> hc.complete=hclust(dist(x), method="complete")
```

用类平均法和最短距离法代替最长距离法进行系统聚类。

```
> hc.average=hclust(dist(x), method="average")
> hc.single=hclust(dist(x), method="single")
```

2、系统聚类法

```
> par(mfrow=c(1,3))
> plot(hc.complete,main="Complete Linkage", xlab="", sub="", cex=.9)
> plot(hc.average, main="Average Linkage", xlab="", sub="", cex=.9)
> plot(hc.single, main="Single Linkage", xlab="", sub="", cex=.9)
```



2、系统聚类法

用cutree( )函数根据谱系图的切割获得各个观测的类标签。

```
> cutree(hc.complete, 2)
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
[36] 2 2 2 2 2 2 2 2 2 2 2 2 2 2
> cutree(hc.average, 2)
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 1 2 2
[36] 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2
> cutree(hc.single, 2)
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

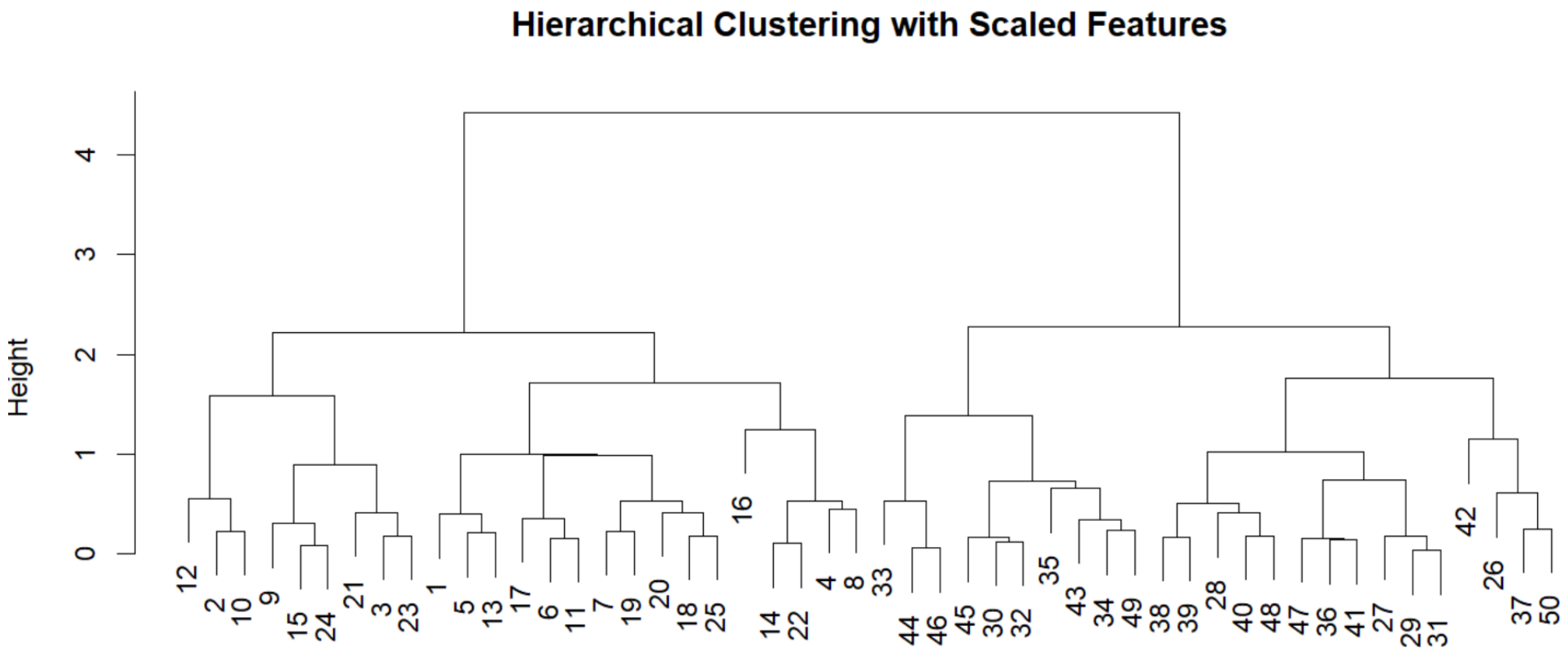
最长距离法和类平均法都准确地将观测分配到正确的类中。最短距离法进行系统聚类时，会有一个点自成一类。而用最短距离法将数据划分为4个类时，尽管仍然有2个观测自成一类，但得到的聚类比聚类数为2的结果更为合理。

```
> cutree(hc.single, 4)
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3
[36] 3 3 3 3 3 3 4 3 3 3 3 3 3 3
```

2、系统聚类法

在对观测进行系统聚类之前，可用scale()函数对变量进行标准化处理：

```
> xsc=scale(x)
> plot(hclust(dist(xsc), method="complete"), main="Hierarchical Clustering
 with Scaled Features")
```



总结

- **无指导学习**对于理解一组未标记数据的变化和分组结构很重要，并且可以成为在使用有指导学习前的，一种有用的预处理方式
- 它本质上比**有指导学习**更难，因为没有标准集（如响应变量）和目标（如测试集的准确性）。

本章作业
(11月8日第十周)

教材10.7习题1-4、10

上述内容下周二之前交（11月15日第十一周）