R 语言中的数据挖掘工具

聚类，分类与维度变换

Milton Deng 邓光宏

WISE
Xiamen University

April 25th, 2016

引言

- “大数据”、“数据挖掘”、“机器学习”等概念成为了近几年全球最火热的话题之一。相关统计模型和数据处理技术也在不断渗透到包括商业和金融分析在内的各个领域。
- 我们将简要讨论数据挖掘中的聚类、分类和维度变换三个重要方向。
- 对于每一个模块，我们都将简要介绍：
  1. 问题背景；
  2. 基本的数学模型；
  3. R 语言中对应的实现方法；
  4. 简单的案例介绍。
- 这里并不强调模型的数学过程，而是重点介绍模型直觉和模型的应用领域。

维度变换

主成分分析

- Principle Component Analysis，PCA，一种经典的降维方法。

典型相关分析

- Canonical Correlation Analysis, CCA，组变量之间的关联关系？

多维标度法

- Multidimensional Scaling，MDS，距离矩阵到原始维度的还原。

PCA:Why?

- 主成分分析的核心问题是，面对维数（个数）众多而又相互关联的变量，我们如何糅合并抽取出具有代表性的几个变量来简化我们的分析过程？
- 一个简单的情形：对于每家上市公司我们可能有数十个财务指标需要分析评价，然而实际上，这些财务指标并非互不关联，而是集中体现了公司几个方面的经营结果（如规模，成长性，盈利能力等）。
- 一个简单的方法是我们人为分析这些变量的含义，并给出变量分组和加权方法，形成一套系统的分类评价体系。但是这种方法虽然能尽可能符合我们的思维模式，但是分组和复权的方法无疑具有主观随意性。

PCA：数学表示

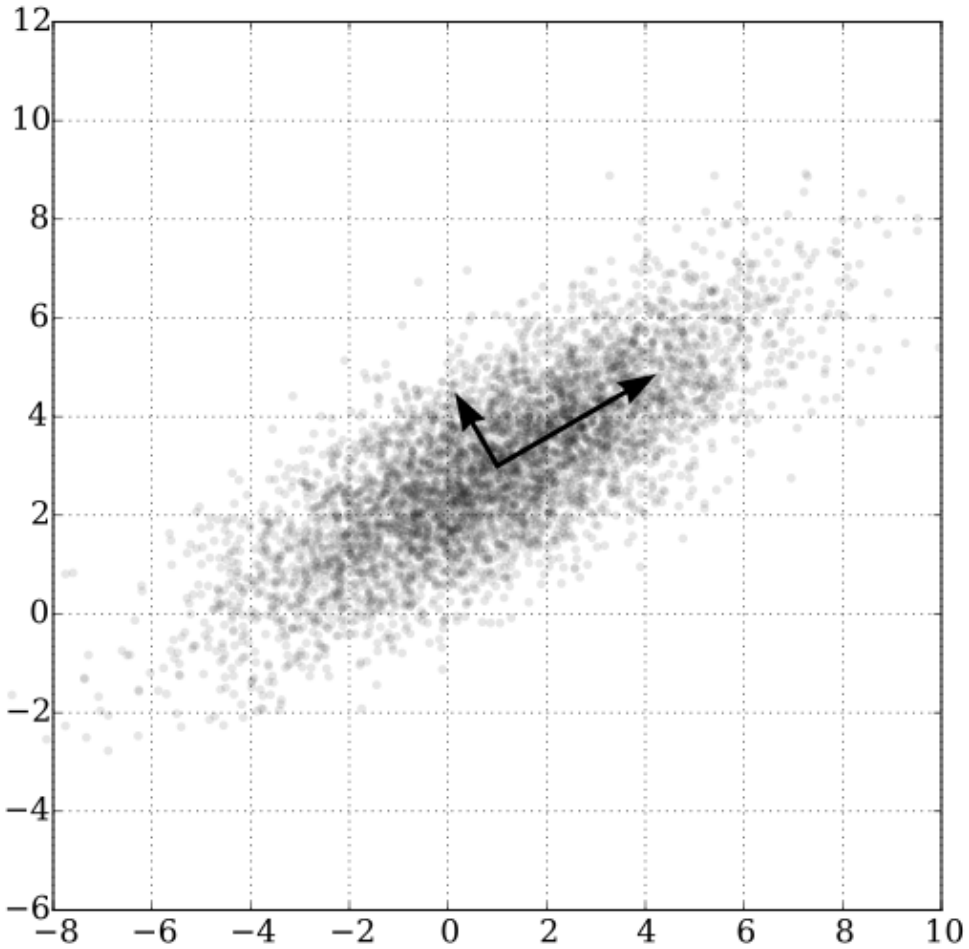
- 我们能否直接从数据出发，利用变量变换，得到一组新的变量，使得这组变量的维数明显减少，并且每个变量反映完全不同的方面，并且使得这组变量尽可能多地包含原有变量信息？
- 这在数学上表示什么？假设原本我们有一组 p 个变量

$$X_1, X_2, ..., X_p$$

- 变量变换：指线性变换，$Y_i = T_i' X$，其中 T_i 为一个系数向量。或者表示为 $Y = T'X$，这个 Y_i 我们即称为主成分。注意要求单位化条件 $T_i' T_i = 1$
- 新变量的独立性：$Cov(Y_i, Y_j) = 0$，这个性质我们成为正交性（Orthogonality）。
- 信息：数据的变异性，以方差衡量。“尽可能多地包含原有变量信息”或者“尽量减少信息损失”都意味着我们希望 Y 的方差尽可能大。

PCA：二维情形

- 二维情形下的变量旋转和主成分提取：



PCA：数学求解

- 现在假设我们只想用一个新变量 $Y_1 = T_1' X$ 代替原有的所有变量，关键即在于如何找到这个系数向量 T_1。由于第一个变量不需要考虑独立性的问题，所以我们可以很容易地构建一个优化模型：

$$max \quad D(Y_1) = T_1' \Sigma T_1$$

$$s.t. \quad T_1' T_1 = 1$$

- 其中 Σ 为 X 的协方差矩阵，$D(.)$ 表示求协方差。可以很容易由拉格朗日一阶条件求出 T_1。这个 Y_1 即我们的第一主成分，它是最能充分反映原有数据信息的一个新变量。

PCA：数学求解

- 而对于之后的 $Y_2, Y_3, \ldots$ 等等的思路是完全相同的。只是我们需要要求正交条件，对于 Y_2 来说，这意味着

$$s.t. T_1' T_2 = 0$$

- 除此之外，T_i 中第 j 元素的大小，实际上反映了 X_j 变量对主成分 Y_i 的贡献度，称为载荷 (Loading)

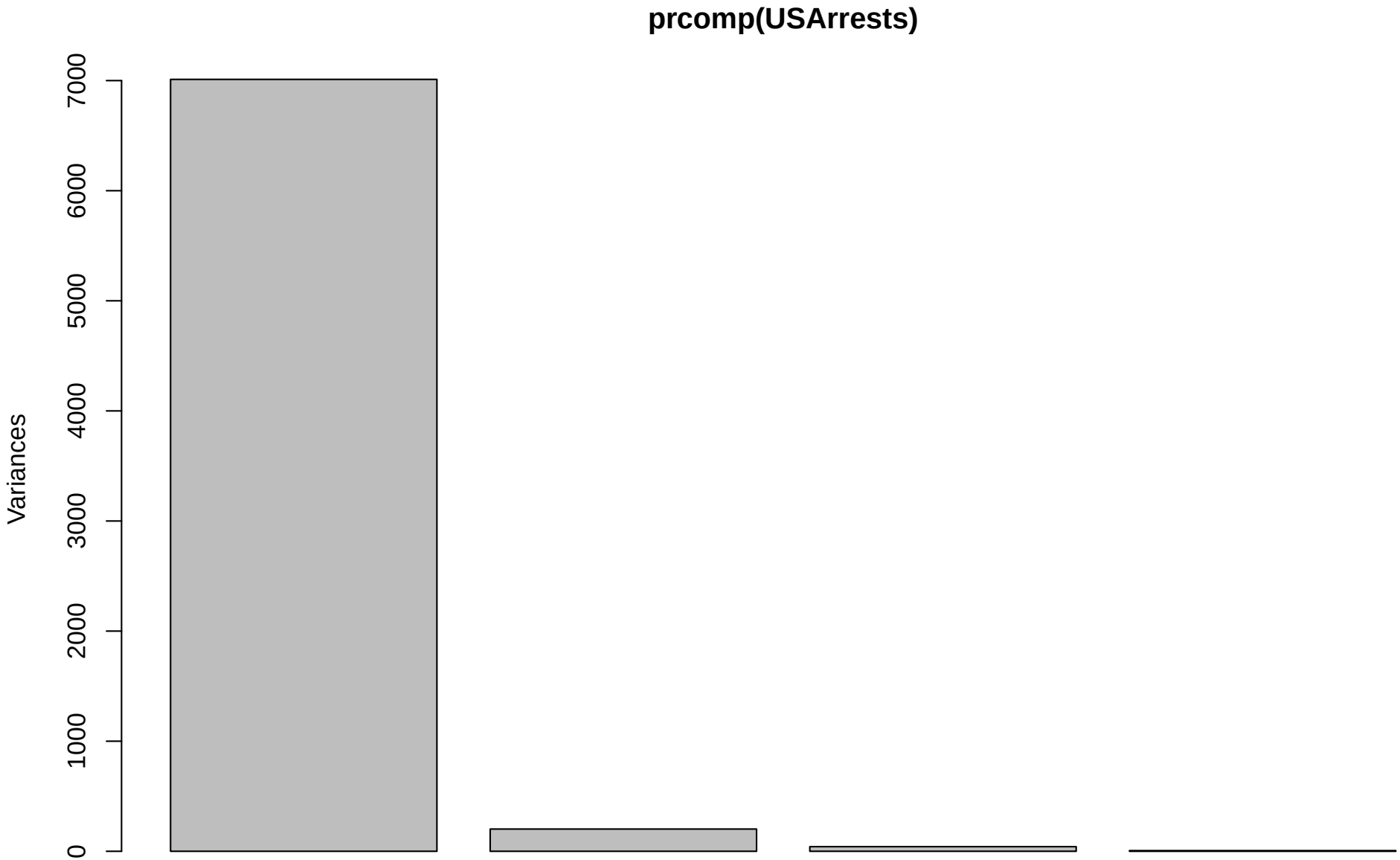
PCA：斯通的例子

- 这里我们不再纠结于数学的推导，而是提供三个分析性的例子。
- 宏观经济变量的分解：
- “美国的统计学家斯通在 1947 年所做的关于国民经济的研究中，曾利用美国 1929 年-1938 年各年的数据，得到 17 个反映国民收入与支出的变量要素，如雇主补贴、消费资料和生产资料、纯公共支出、净增库存、股息、利息外贸平衡等。
- “在他进行主成份分析后，发现能够以 97.4% 的精度，用 3 个新变量取代原来的 17 个变量。这 3 个新变量可以分别命名为总收入、总收入变化率和经济发展或衰退的趋势。” [参考资料 1]

PCA：R 语言实现的例子（1）

- R 语言中对应的函数为 prcomp()，以下为该函数自带的例子：

```
prcomp(USArrests, scale = TRUE) #scaling is needed
prcomp(~ Murder + Assault + Rape, data = USArrests, scale = TR
plot(prcomp(USArrests))
summary(prcomp(USArrests, scale = TRUE))
```



（这个例子并不容易解释，仅作示例）

PCA：R 语言实现的例子（2）

- 25 个运动员，8 项运动，如何缩减为两到三项综合运动能力？

```
# install.packages("HSAUR")

data("heptathlon", package = "HSAUR")

# score all the seven events in the same direction
heptathlon$hurdles <- max(heptathlon$hurdles) - heptathlon$hur
heptathlon$run200m <- max(heptathlon$run200m) - heptathlon$run
heptathlon$run800m <- max(heptathlon$run800m) -
heptathlon$run800m
```

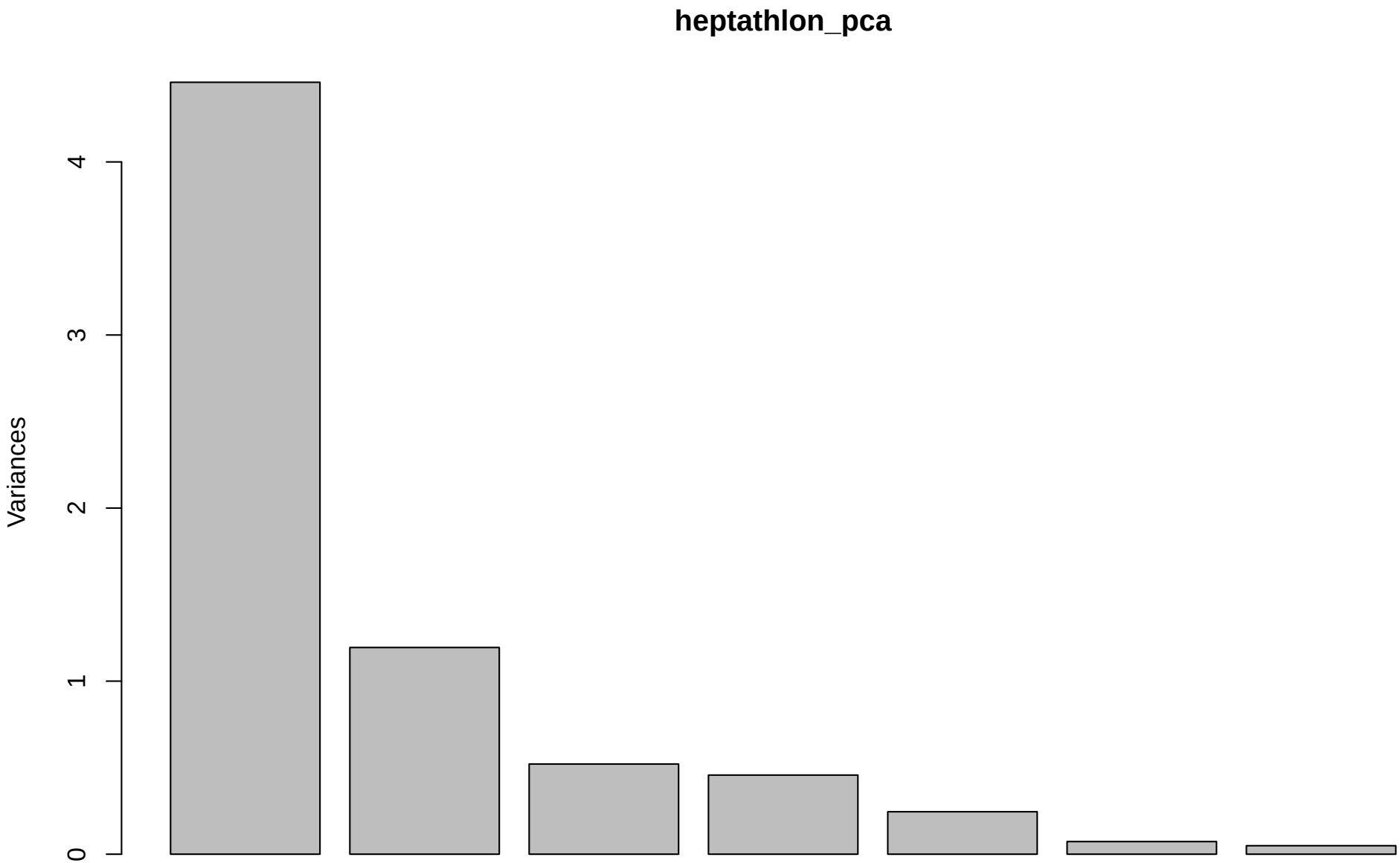
PCA：R 语言实现的例子（2）

- 25 个运动员，7 项运动，如何缩减为两到三项综合运动能力？

```
heptathlon_pca <- prcomp(heptathlon[, -8], scale = TRUE)
print(heptathlon_pca)
summary(heptathlon_pca)
plot(heptathlon_pca)
```

- 可以看到前两个主成分可以解释 81% 的方差变动。除 javelin 外，其他运动的载荷都是相对均衡的，javelin 反映了一个相对的独立的运动能力。

（来自参考资料 2）


heptathlon_pca
Variances
0
1
2
3
4

PCA：总结

- PCA 是多元统计中最为重要的技术之一。这种重要性不仅体现在它的降维能力和分析能力，更体现在许多其他分析方法都是以 PCA 为基础的。PCA 实际上集中体现了方差分解和投影（Projection）两个重要思想。
- 和两百年前孟德尔撒豌豆的年代相比，在这个数据爆炸的时代，我们缺少的往往不是样本数，也不是变量，而是在复杂而广泛的数据中抽取信息的能力。

CCA: Why?

- 当我们进行统计分析时，相关系数（Correlation Coefficient）甚至已经成为了我们潜意识的第一反应。但是相关系数是针对两两变量的，这使得我们不禁思考一个有意思的问题：
- 如果我们有两组变量，每组的变量个数都大于 1，如何刻画或分析组与组之间的相关性？我们是否能构建两个线性组合，使得它们的相关系数最大？

CCA: 数学模型

- 简单描述这个数学模型的思路：
- 两组变量：

$$X^{(1)} = (X_1^{(1)}, X_2^{(1)}, ..., X_p^{(1)})$$

$$X^{(2)} = (X_1^{(2)}, X_2^{(2)}, ..., X_q^{(2)})$$

- 新的线性组合：

$$U = a'X^{(1)}, \quad V = b'X^{(2)}$$

CCA: 数学模型

- U, V 之间的相关系数（优化的目标函数）:

$$Corr(U, V) = \frac{a'\Sigma_{12}b}{\sqrt{(a'\Sigma_{11}a)}\sqrt{(b'\Sigma_{22}b)}}$$

- 正则条件：

$$s.t. \quad D(U) = a'\Sigma_{11}a = 1, \quad D(V) = b'\Sigma 22b = 1$$

- U, V 的解可以表达为协方差矩阵 Σ 的一些乘积项，这里不再展开。

CCA: R 语言实现

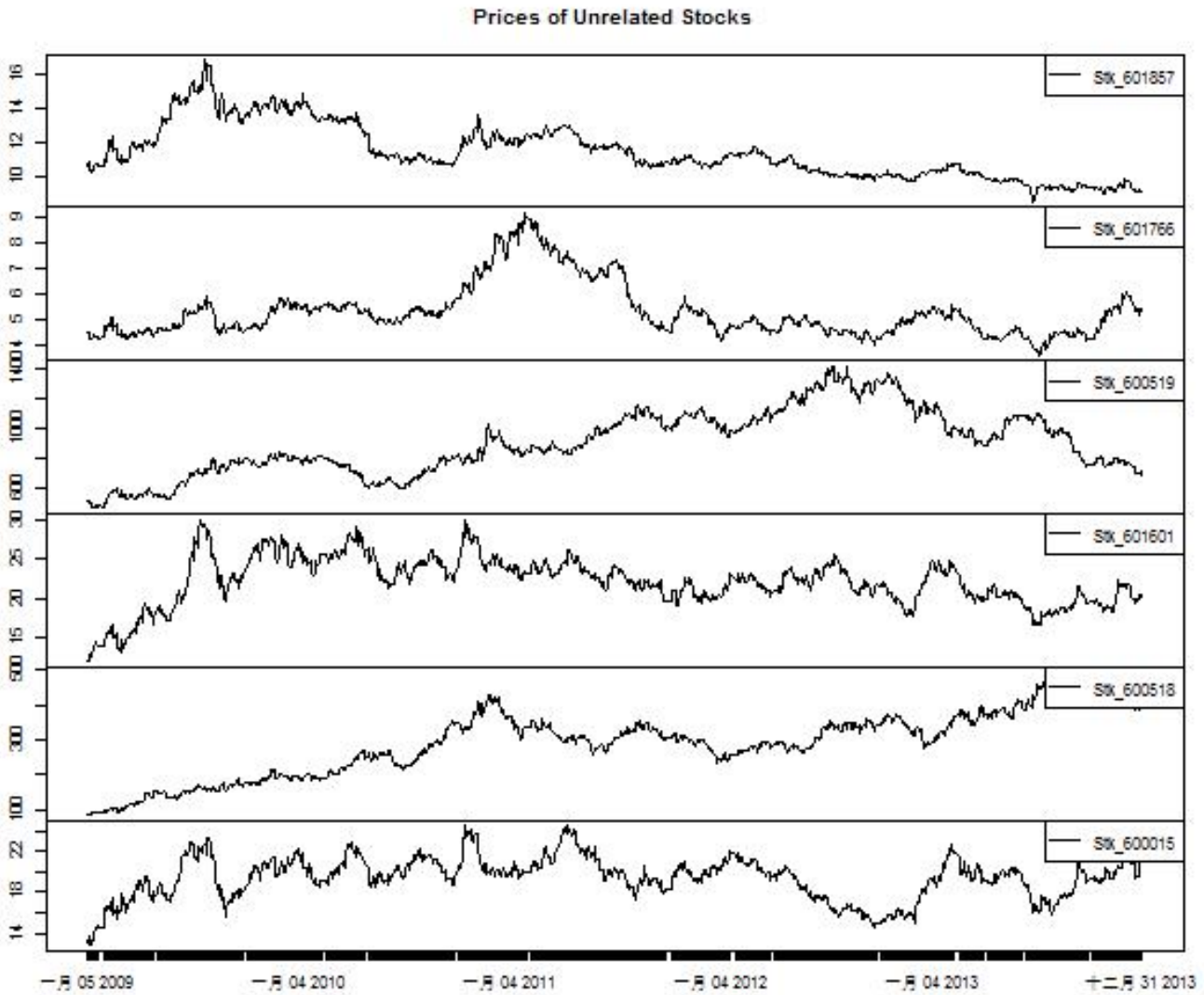
- R 语言中 `stats::cancor()` 实现了一个基础的 CCA 模型。另外有 CCA 包扩展了这一方法。

```
pop <- LifeCycleSavings[, 2:3]
oec <- LifeCycleSavings[, -(2:3)]
cancor(pop, oec)
```

CCA: 一个金融市场的例子

- 实际上 CCA 在解释上常常存在一些困难。我在我的本科毕业论文《基于协整理论与典型相关分析的统计套利模型研究》里做了一些应用的尝试。(尽管仍然缺乏应用的价值！)
- 这篇文章试图捏合两组看似不相关的股票组合，并构建套利机会。

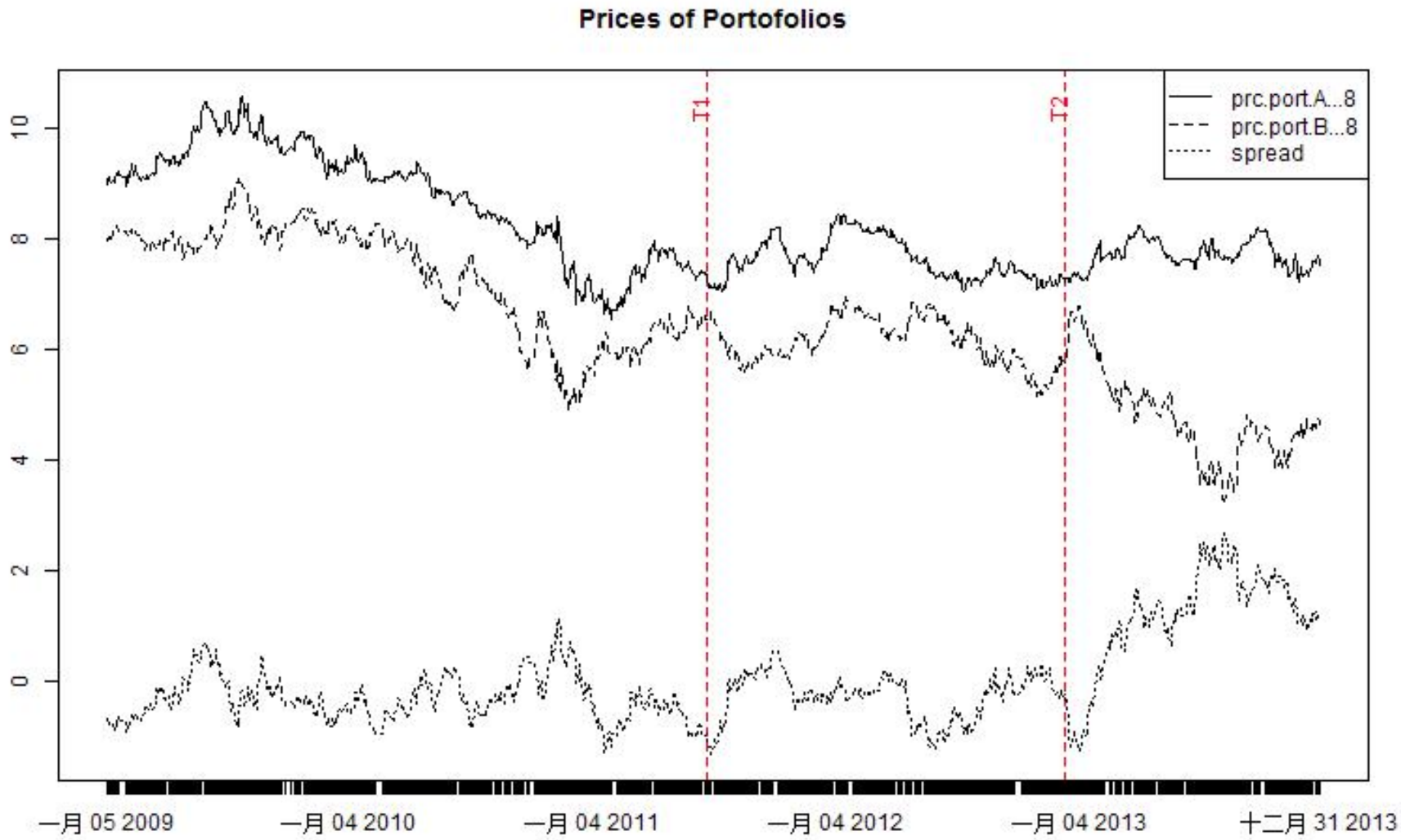
CCA: 一个金融市场的例子



CCA: 一个金融市场的例子

资产组合A			资产组合B		
股票代码	股票名称	系数	股票代码	股票名称	系数
601857	中国石油	-0.409	601601	中国太保	-0.147
601766	中国南车	0.535	600518	康美药业	0.013
600519	贵州茅台	0.002	600015	华夏银行	0.052

CCA: 一个金融市场的例子



MDS: Why?

- MDS 提出了另一个有意思的问题：设想有一些城市，我们知道它们彼此之间的距离是多少，但是不知道它们各自的坐标，有没有办法利用这些距离把它们的相对位置重新还原回地图上？
- 一个玄一点的例子：我们调查了很多消费者对 10 种饮料的相似度的评价，假设这 10 种饮料分别占据了消费者“心理空间”的一个角落，我们能否直接通过这个相似度度量还原出消费者的“心理空间”？
- MDS 更一般的表述为：我们知道 n 个对象两两之间的相似性（或者反过来说，距离），能否重新把这些个体映射到低维的空间上，使得这种映射的形变最小？（注意这时我们不知道这些个体分布在几维空间）

MDS：数学模型

- MDS 的数学模型相对比较复杂，这里只介绍大概的思路：
- 假设这些个体是 r 维空间的 n 个点 $X_1, X_2, ..., X_n$（均为 r 为向量），点与点的两两距离（欧式）为：

$$d_{ij}^2 = (X_i - X_j)'(X_i - X_j)$$

- 注意这时这些点仍是任意的，它们成为给定的距离阵 D 的拟合构图，而这些虚构的点的距离可以形成一个距离阵 D^*，我们希望 D^* 与 D 尽可能接近。

MDS：R 语言实现

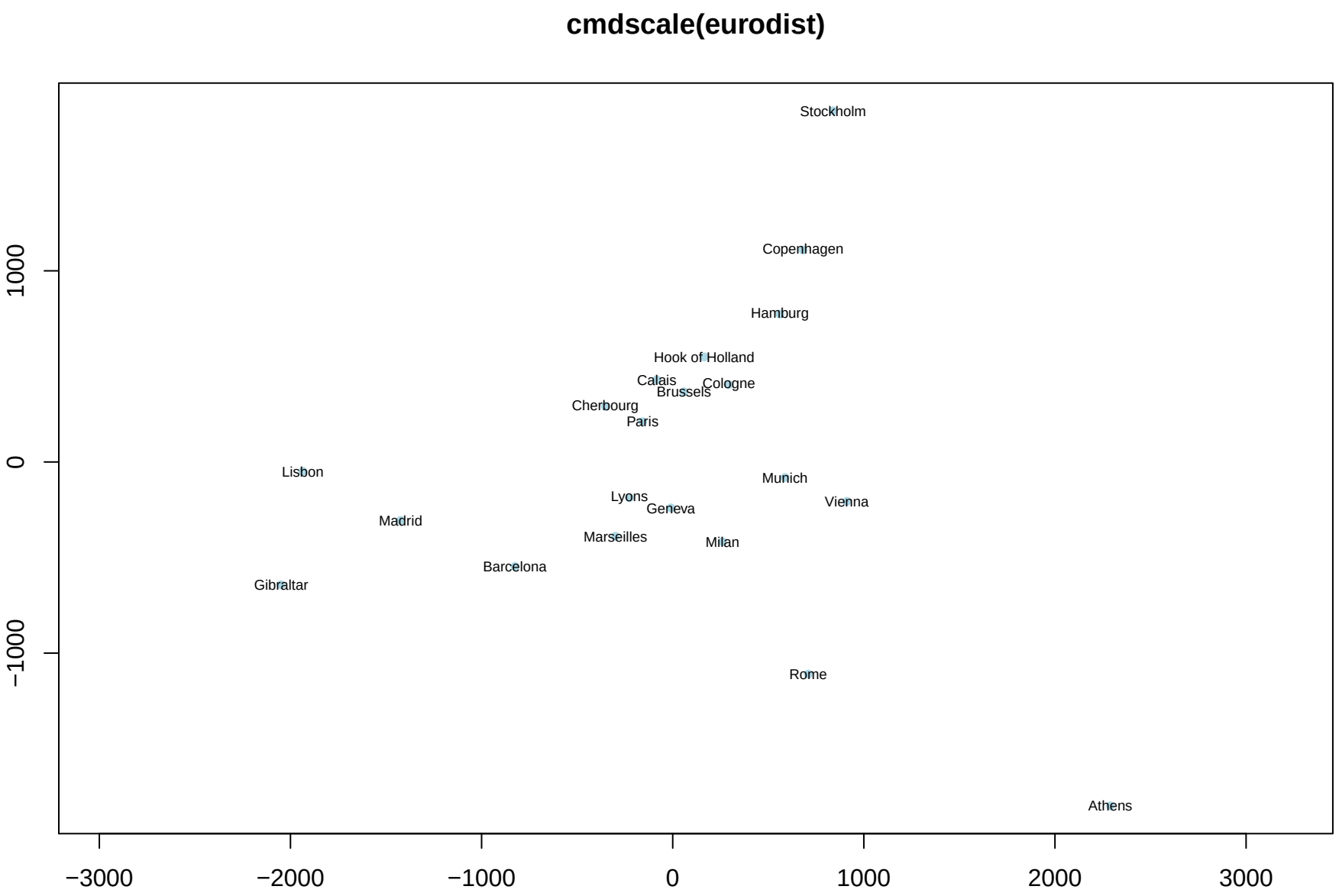
- R 语言中对应的函数为 cmdscale()。我们现在就分析引言中关于地理位置的例子。

```
require(graphics)

loc <- cmdscale(eurodist)
x <- loc[, 1]
y <- -loc[, 2] # reflect so North is at the top
plot(x, y, type = "p", pch = 20, xlab = "", ylab = "", asp = 1
     main = "cmdscale(eurodist)", col = "lightblue")
text(x, y, rownames(loc), cex = 0.6)
```

(来自函数自带示例)

MDS：R 语言实现


cmdscale(eurodist)
Stockholm
Copenhagen
Hamburg
Hook of Holland
Calais
Cologne
Brussels
Cherbourg
Paris
Lisbon
Munich
Lyons
Vienna
Geneva
Madrid
Marseilles
Milan
Barcelona
Gibraltar
Rome
Athens
1000
0
−1000
−3000
−2000
−1000
0
1000
2000
3000

分类和聚类：监督与无监督下的模式发现

- 分类（也可以称为判别）和聚类是统计分析中两个重点研究的问题。
- 分类：已知一些样本应该归属于哪类，并且我们也知道每个样本的一些特征（Feature），我们很想发现一些模式，能够使我们将特征和类别匹配起来。这样如果以后我们拿到一个新的样本，就能够利用这个模式，快速找到它应该归属的类别了。典型的情形如我们拿到了一个新的人类骨骼化石，我们能否通过这块骨骼的特征确定它的人种等信息？
- 聚类：我们有一堆样本，也知道它们的一些特征，但是它们是混杂在一起的，我们想知道它们内在是否具有某些同质性或者结构性？比如地上散落了一些红色的圆球，我们通过分析其实可以发现这些球其实一共就两种，一种叫“苹果”，一种叫“桃子”。这类问题在 CRM 的客户分析中非常常见。

分类和聚类：监督与无监督下的模式发现

- 第一类已知结果试图去发现模式的问题可以称为“有监督问题”（Supervised）。其实广义上，回归分析都是一种有监督的问题。
- 第二类不存在已知结果的问题称为“无监督问题”（Unsupervised）。
- 此外在目前大热的机器学习（Machine Learning）领域聚类和分类也是核心的研究对象。这里我也顺道给出一个关于机器学习的定义，并且体会 E、T、P 这些核心概念是如何体现在下面的模型的：
- Computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E. (Tom Mitchell, 1998)

分类 (Discriminant Analysis)

- 目前分类方法已经非常庞大，所有这些模型都会提供一个判别规则，也就是“分类器” (Classifier)。
- 我们这里只介绍一类广义线性模型 (Genearalized Linear Model, GLM)。其他经典的方法有 Fisher, Bayesian，决策树 (Decision Tree) 等方法。ANN (Artificial Neural Network, 人工神经网络) 等也可以构建出适当的分类器。

GLM：和 LM 什么关系？

- 回想我们的线性回归模型（Linear Regression Model, LM），我们基础的截面（Cross-Sectional）回归，也往时间轴上扩展过，做过时间序列分析（Time Series Analysis）和协整（Cointegration），我们甚至把两者结合起来形成面板数据（Panel Data），但是... 所有这些变量竟然都是连续的？
- GLM 的重要扩展是此时的因变量 Y 可以不是连续的了！它可以是二项分布，也可以说 Poisson 分布，甚至是多项分布（Multinomial Distribution）... 二项分布和多项分布就对应了典型的分类的问题。

GLM：构件

- GLM 一般表达为：

$$g(\mu) = \alpha + \beta_1 x_1 + ... + \beta_k x_k$$

- GLM 总是由三部分构成：

1. Random Component: 因变量 Y，它可以指定一个分布，并且这个分布不必是正态的；
2. Systematic Component: 一组自变量 X，用以解释 Y。对于多个自变量，我们总是去构建线性组合，并把这个线性组合称为 Linear Predictor；
3. Link Function: $\mu = E(Y)$，GLM 不直接对 μ 建模，而是先施加一个函数 $g(.)$，这个 $g(.)$ 即为 Link Function。

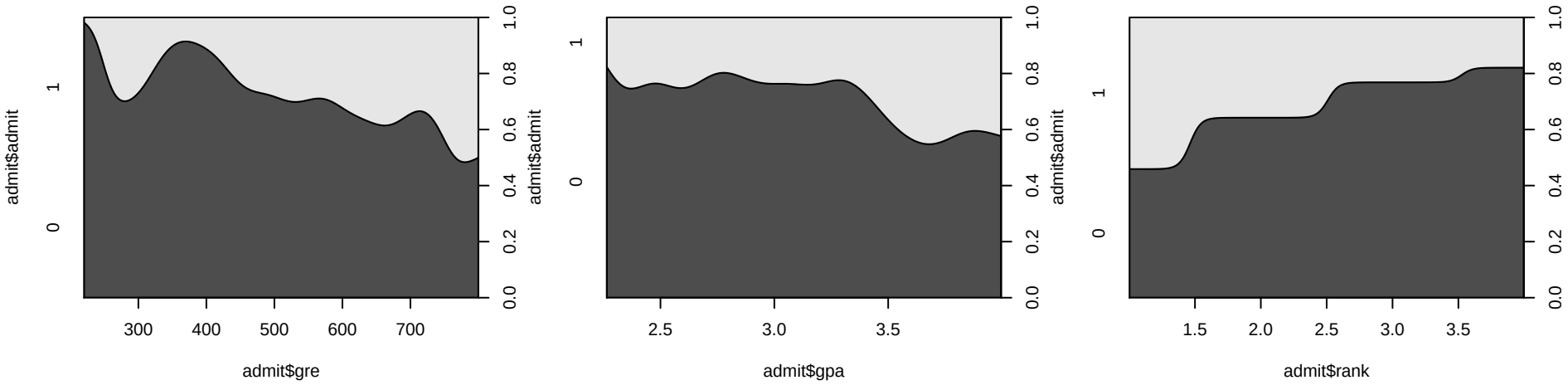
GLM：0-1 分类

- 最简单的分类即为 0-1 分类，比如能否成功，是否患病，都是典型的二分问题。一般我们假设此时 Y 是二项分布的，此时 μ 等于成功概率 π，并且 Link Function 有两个典型的形式：
- Logisitic(or Logit): $g(\pi(x)) = log(\frac{\pi(x)}{1-\pi(x)})$ (Log Odds Ratio，对数风险比)
- Probit: $g(\pi(x)) = \Phi^{-1}(\pi(x))$
- 两类 Link Function 均将概率映射到 (0, 1) 区间上的一条 S 型曲线上，以反映概率增长的有界性和非线性性。两个模型广泛应用于医学和生物等领域。而在经济学中，微观计量也大量应用这类模型来解释个体选择的影响因素。

GLM：R 语言实现

```
admit <- read.csv("http://www.ats.ucla.edu/stat/data/binary.cs
admit$admit <- as.factor(admit$admit)

par(mfrow = c(1, 3))
cdplot(admit$gre, admit$admit)
cdplot(admit$gpa, admit$admit)
cdplot(admit$rank, admit$admit) # prestige of the undergraduat
```




```
par(mfrow = c(1, 1))
```

GLM：R 语言实现

```
admit$rank <- factor(admit$rank)
logit <- glm(admit ~ gre + gpa + rank, data = admit, family =
probit <- glm(admit ~ gre + gpa + rank, data = admit, family =
```

- 这个模型结果反映了 GRE，GPA 和本科学校声誉如何影响被录取的概率。给定一个新的学生，只要我们知道它的三项数据，就可以得到其被录取的概率。
- 关于模型的评价，我们通常要涉及到 Sensitivity，Speticity，ROC 曲线和 AUC 等概念。
- `glm()` 函数只需要修改 `family` 参数就可以扩展到其他 GLM 模型。如 `family = poisson(link="log")` 就可以实现对数 Poisson 模型，该模型可以用来描述一些计数（事件发生频率）问题。

聚类 (Clustering Analysis)

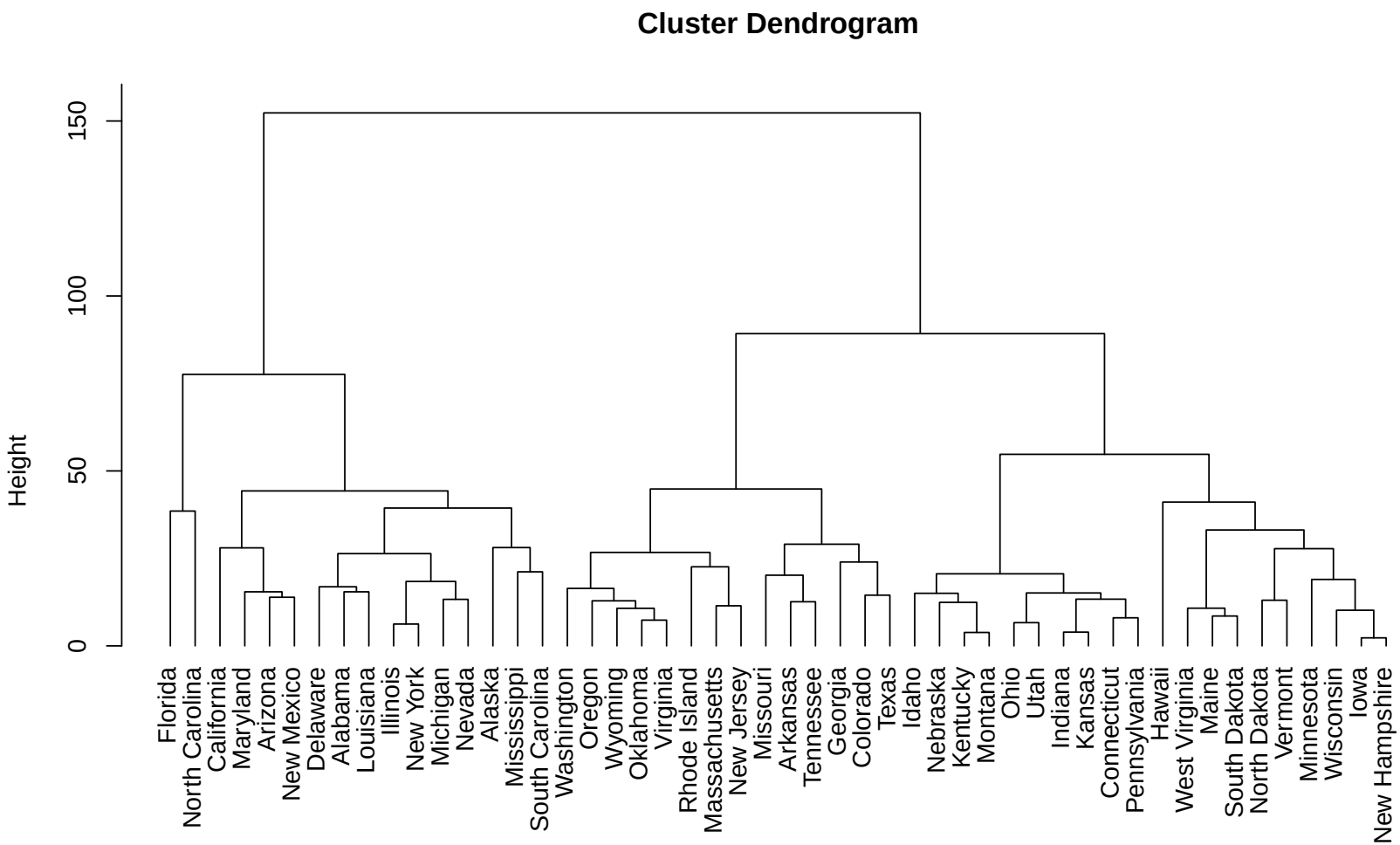
- 目前聚类方法已经发展地非常庞大。通常聚类方法可以分为两类：

1. 系统聚类 (Hierachichal Clustering Analysis，HCA)，这类方法先将 n 个个体各自设为一类，形成第一层，然后按照距离的远近，将距离最近的个体合成一类，重复这个过程直到所有的个体都聚集到最终的一个类当中。(Agglomerative，或者 Bottom-up 的方式)
2. 非系统聚类直接指定一个类的个数，并通过迭代一次形成类的划分。典型的方法如 K 均值聚类 (K Means Clustering)。

- 其他一些比较新的模型有支持向量机 (Supporting Vector Machine, SVM) 等。
- 经济和社会学研究中，一些基于调查问卷的人群分析也经常会用到聚类技术。

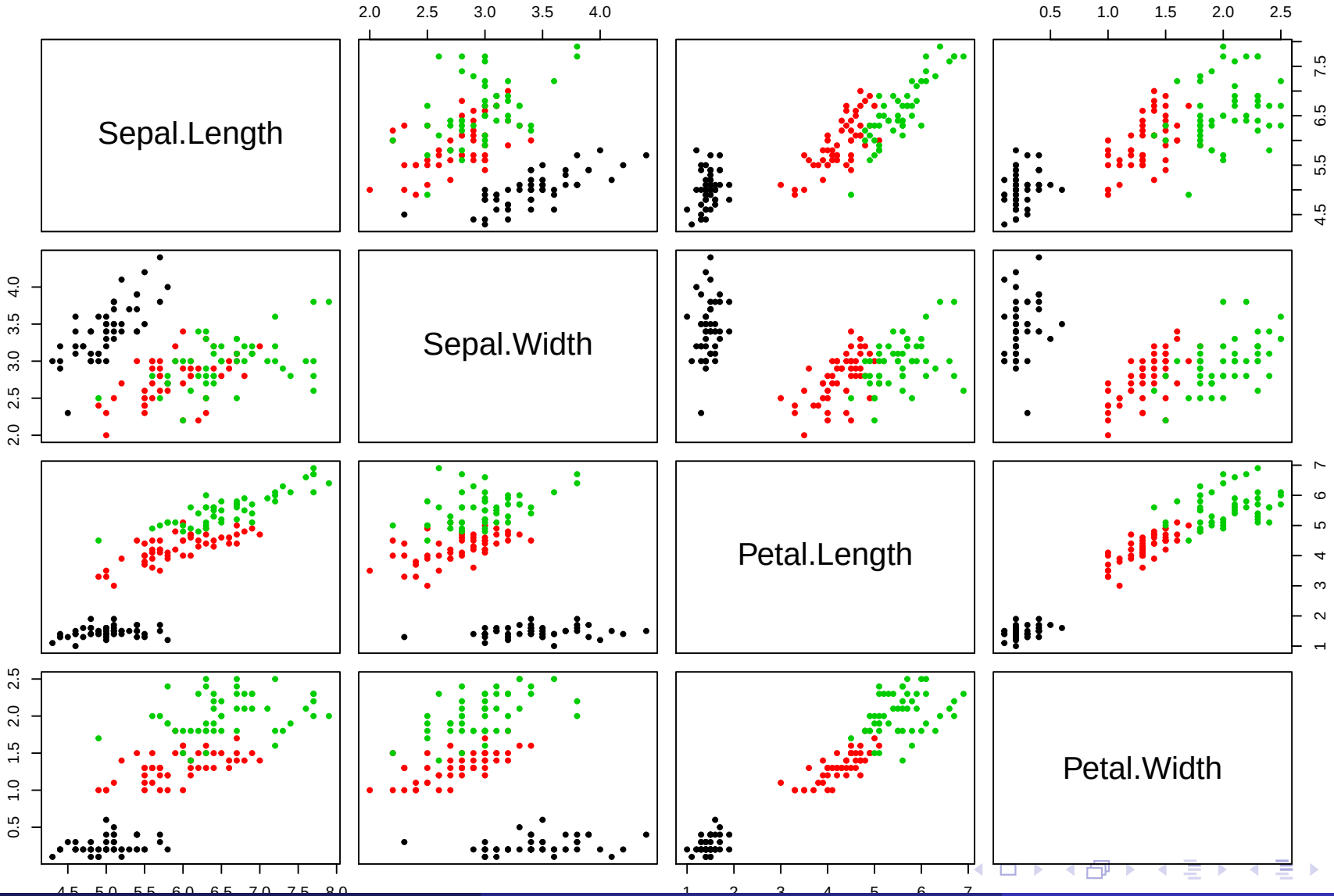
HCA：一个例子

```
hc <- hclust(dist(USArrests), "ave")
plot(hc, hang = -1)
```



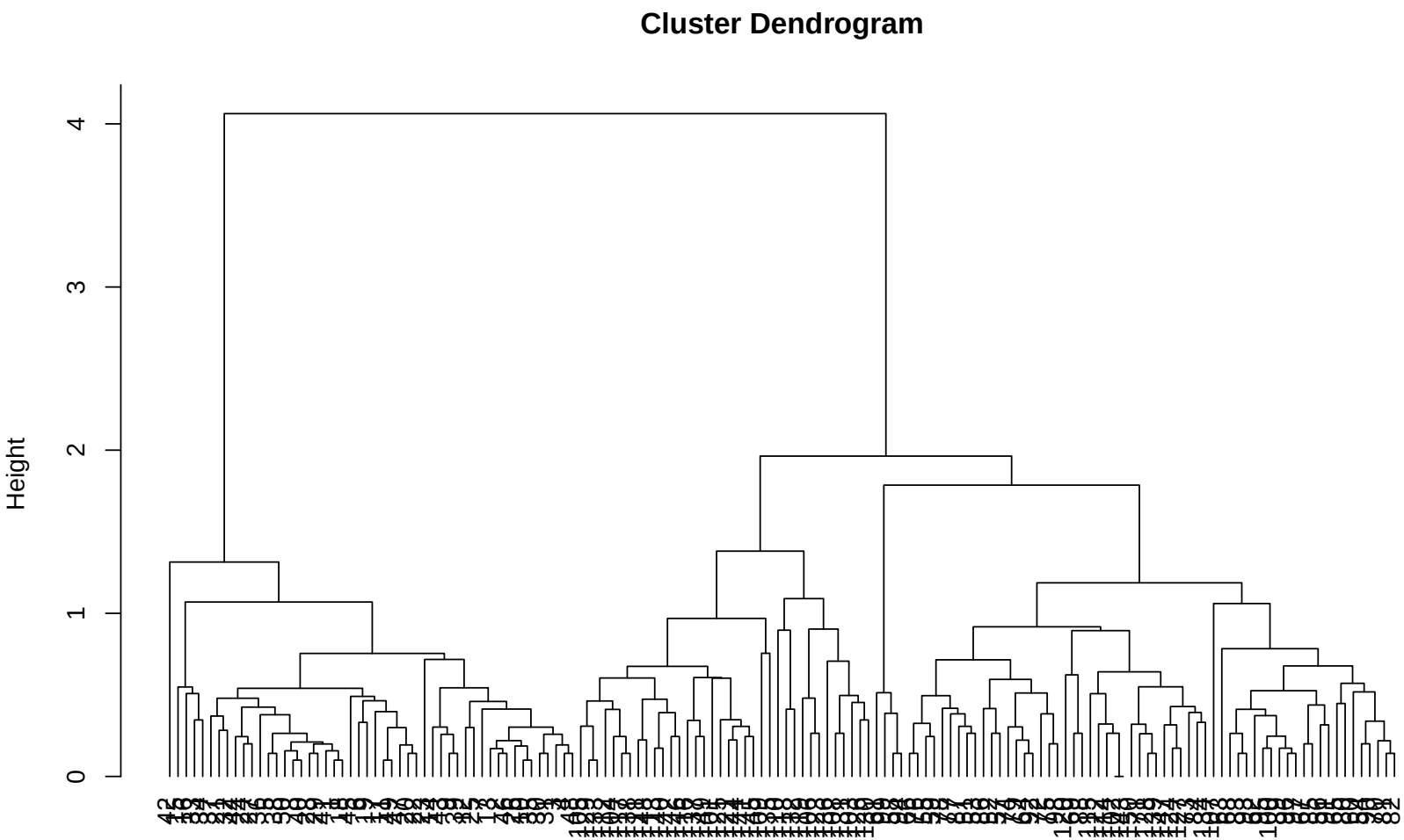
HCA：Iris 的例子

```
pairs(iris[, -5], col = iris[, 5], pch = 20)
```



HCA：Iris 的例子

```
hc <- hclust(dist(iris[, -5]), "ave")
plot(hc, hang = -1)
```



KNN：Iris 的例子

```
kc <- kmeans(dist(iris[, -5]), 3)
table(iris[, 5], kc$cluster)
```

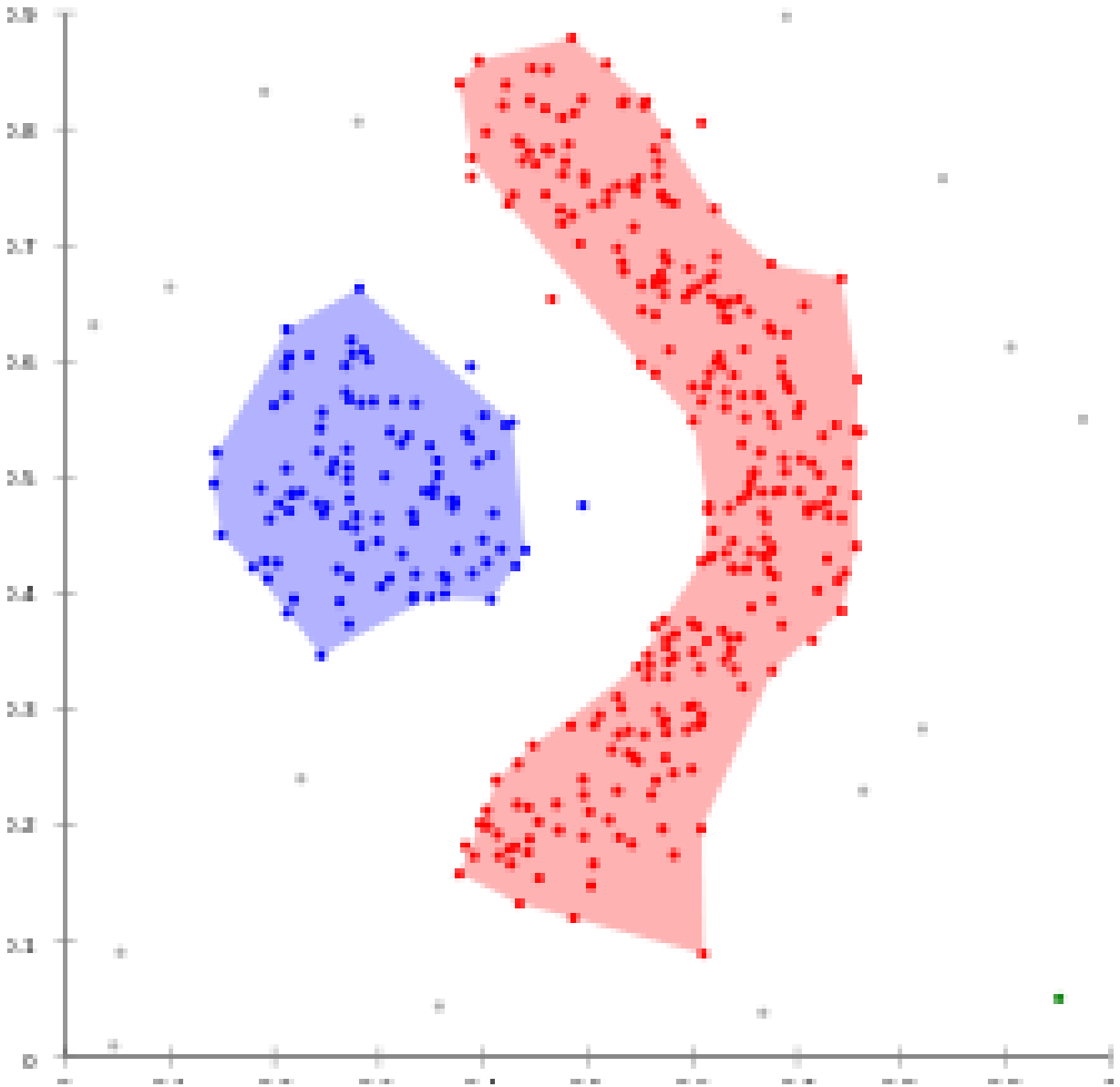
```
##
##                1  2  3
##    setosa      0  0 50
##    versicolor  1 49  0
##    virginica  37 13  0
```

DBSCAN：基于密度的聚类方法

- 除了上述经典方法外，我们这里再介绍一类基于密度的聚类方法（DBSCAN）。它的主要思想是，如果某些个体的点是归属于同一类的，那么我们就能够找到一片相连通（距离足够近）的点。
- 这类方法突破了传统方法的一些局限性，并且也成为了重要的聚类方法之一。

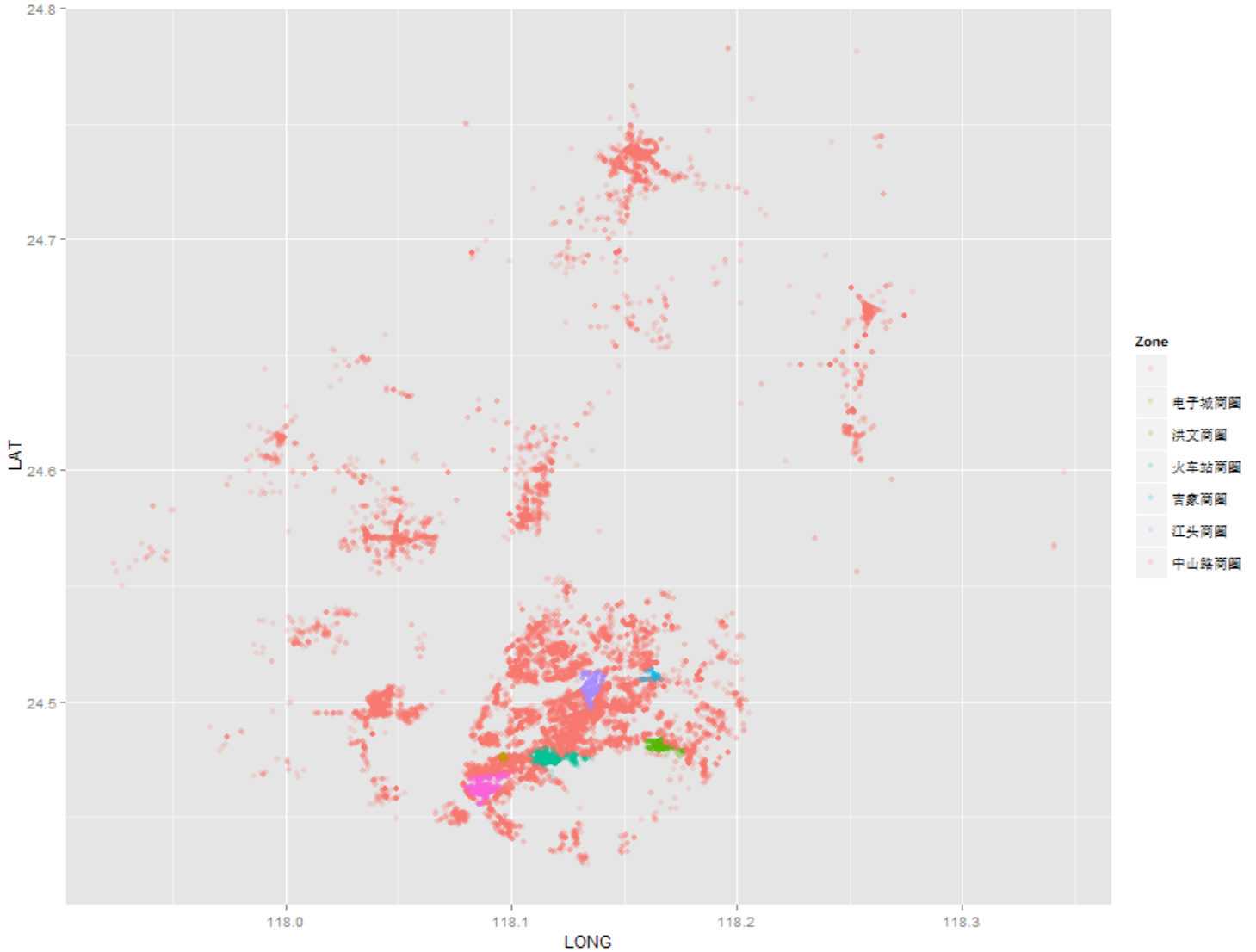
1. 对类在几何空间中的形状不再有球形限制；
2. 类的个数是内生的，让数据特征自己决定；
3. 剥离了噪声和异常点的影响，这更加符合实际，同时也提供了额外的分析能力。

DBSCAN：基于密度的聚类方法

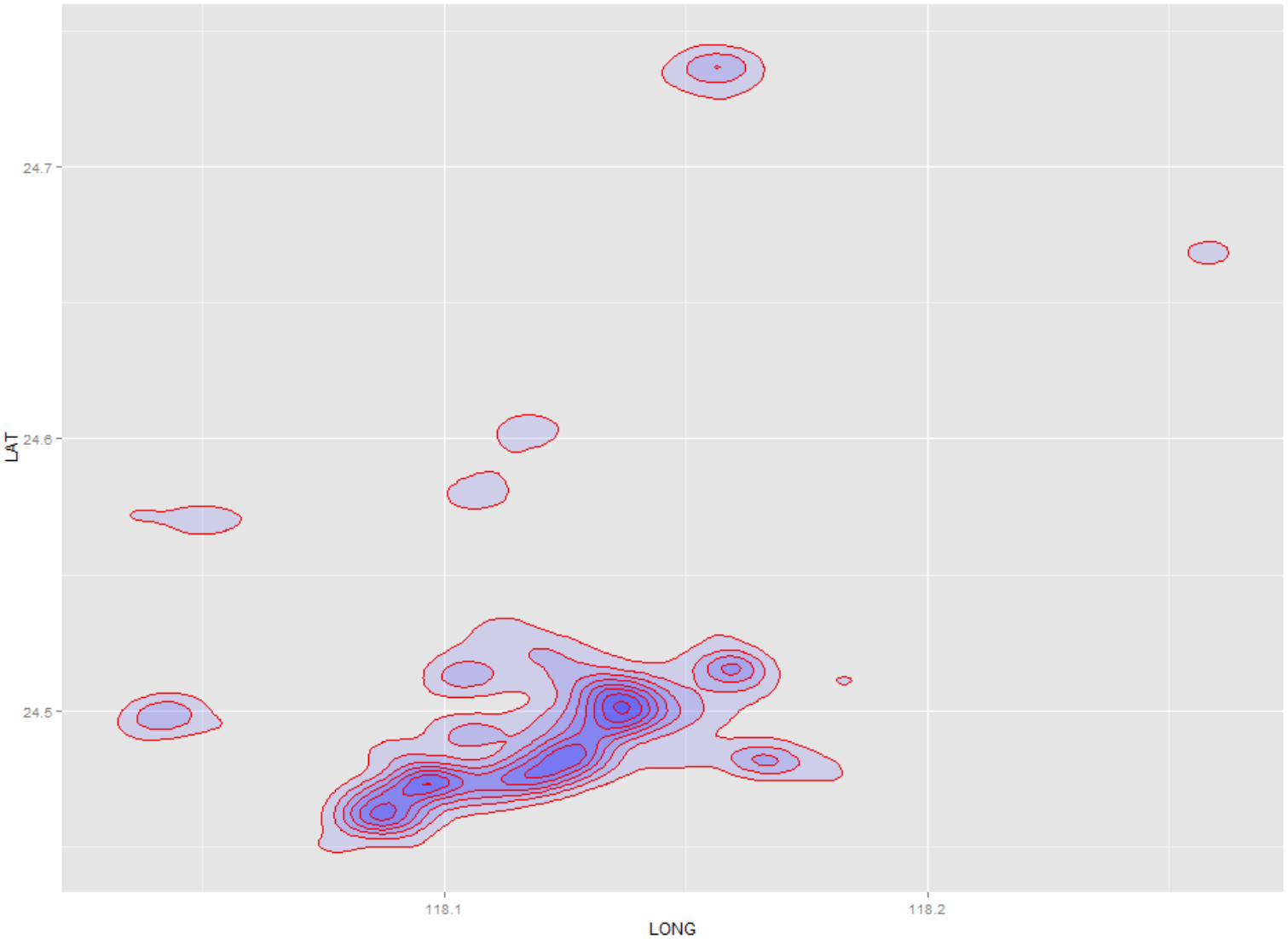


DBSCAN：基于密度的聚类方法

- 厦门 3 万多家商圈的地理位置



DBSCAN：基于密度的聚类方法



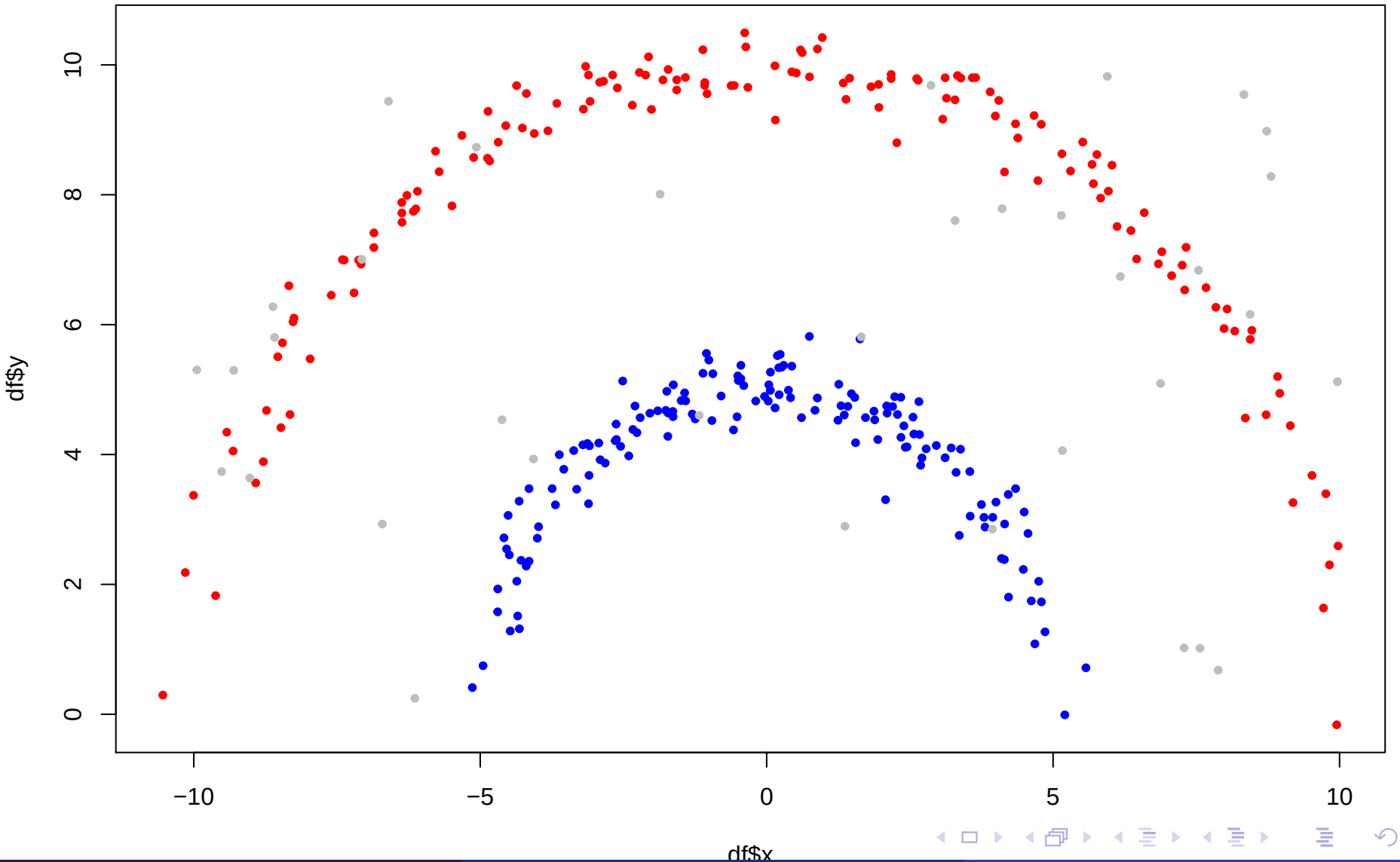
DBSCAN：密度聚类的一个例子

```
n <- 150
nn <- 35
s <- seq(-5, 5, length.out = n)
x1 <- s + rnorm(n, sd = 0.3)
y1 <- sqrt(5^2 - s^2) + rnorm(n, sd = 0.3)
s <- seq(-10, 10, length.out = n)
x2 <- s + rnorm(n, sd = 0.3)
y2 <- sqrt(10^2 - s^2) + rnorm(n, sd = 0.3)
x3 <- runif(nn, -10, 10)
y3 <- runif(nn, 0, 10)
g <- c(rep(1, n), rep(2, n), rep(0, nn))

df <- data.frame(x = c(x1, x2, x3), y = c(y1, y2, y3), g = g)
```

DBSCAN：密度聚类的一个例子

```
plot(df$x, df$y, col = c("grey", "blue", "red")[df$g + 1], pch
```



DBSCAN：密度聚类的一个例子

```
dbscan <- function(x, eps, min_points) {
  n <- nrow(x)
  c <- 0
  cluster <- rep(0, n) # 0 means noise
  visited <- rep(0, n) # 0 means not visited yet
  for (p in 1:n) {
    cat(p)
    if (visited[p] != 0) next;
    visited[p] <- 1
    neighbors <- regionQuery(p, x, eps)
    if (length(neighbors) < min_points) {
      cluster[p] <- 0 # redundant, just to make it clear
    } else {c <- c + 1
      #expandCluster
      cluster[p] <- c
      q_idx <- 1
```

DBSCAN：密度聚类的一个例子

```
    while(q_idx != length(neighbors) + 1) {
      q <- neighbors[q_idx]
      # print(paste(q_idx, length(neighbors), q, sep = " -
      if (visited[q] != 0) {
        q_idx <- q_idx + 1
        next
      }
      visited[q] <- 1
      q_neighbors <- regionQuery(q, x, eps)
      if (length(q_neighbors) >= min_points) {
        neighbors <- c(neighbors, q_neighbors[!(q_neighbors
      }
      if (cluster[q] == 0) cluster[q] <- c
      q_idx <- q_idx + 1
    }
  }
}
```

DBSCAN：密度聚类的一个例子

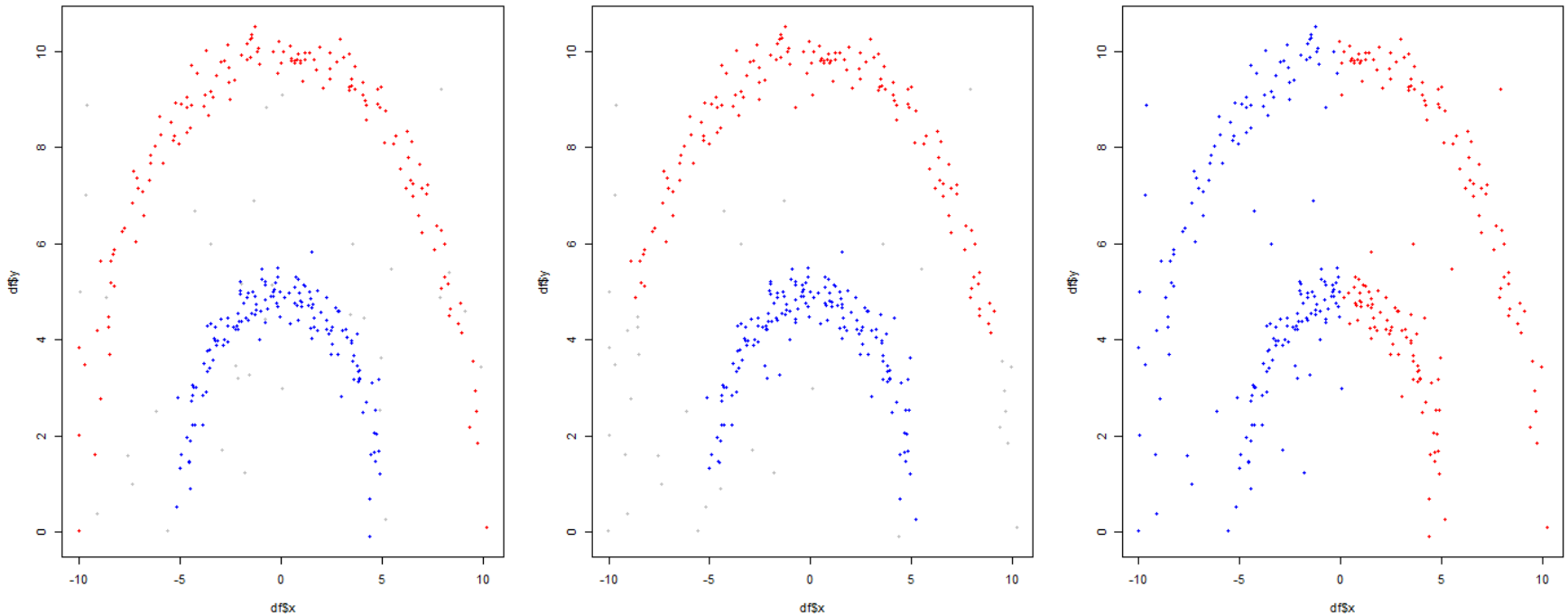
```
regionQuery <- function(p, x, eps) {
  d <- sapply(1:nrow(x), FUN = function(q) sum((x[p, ] - x[q,
  return(which(d <= eps))
}
```

DBSCAN：密度聚类的一个例子

```
cluster <- dbscan(df[, -3], eps = 1, min_points = 8)
k_cluster <- kmeans(df[, -3], 2)
# table(cluster, df$g)
par(mfrow = c(1, 3))
plot(df$x, df$y, col = c("grey", "blue", "red")[df$g + 1], pch
plot(df$x, df$y, col = c("grey", "blue", "red")[cluster + 1],
plot(df$x, df$y, col = c("grey", "blue", "red")[k_cluster$clus
par(mfrow = c(1, 1))
```

DBSCAN：密度聚类的一个例子

- 原始数据（左），DBSCAN（中），和 K 均值聚类（右）



参考资料

1. 多元统计分析，朱建平
2. A Handbook of Statistical Analyses Using R, Brian S. Everitt and Torsten Hothorn
3. Machine Learning Foundations, Andrew NG
4. Introduction to Categorical Data Analysis, Alan Agresti
5. Data Mining, Concepts and Techniques, Jiawei Han, Micheline Kambef
6. Wikipedia