统计建模与R软件

（上册）

薛　毅　陈立萍　编著

清华大学出版社

内 容 简 介

R 软件是一种统计软件，也是一种数学计算环境．它提供了有弹性的、互动的环境来分析、可视及展示数据；它提供了若干统计程序包，以及一些集成的统计工具和各种数学计算、统计计算的函数，用户只需根据统计模型，指定相应的数据库及相关的参数，便可灵活机动的进行数据分析等工作，甚至创造出符合需要的新的统计计算方法．使用 R 软件可以简化你的数据分析过程，从数据的存取，到计算结果的分享，R 软件提供了更加方便的计算工具，帮助你更好地分析和解决问题．通过 R 软件的许多内嵌统计函数，用户可以很容易学习和掌握 R 软件的语法，也可以编制自己的函数来扩展现有的 R 语言，完成你的科研工作．

本书既深入浅出、通俗易懂，又从数理统计的角度对 R 软件进行科学、准确和全面的介绍，不仅介绍其基本用法，而且简要介绍一些必须的专业知识背景，以便使读者能深刻理解该软件的精髓和灵活、高级的使用技巧．此外，我们还将介绍在工程技术、经济管理、社会生活等各方面的丰富的统计问题及其统计建模方法，通过该软件其问题进行求解，使读者获得从实际问题建模入手、到利用软件进行求解，以及对计算结果进行分析的全面训练．

本教材以统计理论为基础，按照数理统计教材的章节顺序，在讲明统计的基本概念的同时，以 R 软件为辅助计算手段，重点介绍统计计算的方法，从而有效地解决统计中的计算问题．

本书可作为理工、经济、管理、生物等专业学生数理统计课程的辅导教材或教学参考书，也作为统计计算课程的教材，和数学建模竞赛的辅导教材．

前　　言

本书力求将实用统计方法的介绍与在计算机上如何 R 软件实现这些方法紧密地联系起来，不仅介绍各种上数理统计方法的统计思想、实际背景、统计模型和计算方法，并且结合 R 软件，给出相应的解决问题的步骤和对计算结果进行分析.

关于数理统计的教材或教科书已非常多，这类教材主要是以数理统计的理论为基础，讲清其理论、方法与应用背景，但对于计算，讲的较少，基本是以手工计算为主，目的是为了帮助读者理解相应的统计方法，可操作性不强.

关于统计计算的书也有不少，目前，统计计算的教材一般是讲算法（这一点与数值分析或计算方法差不多），而没有相应的软件做支撑，有些内容是数值分析内容的重复，统计味不足.

结合软件讲统计的书，目前最多的是结合 SAS 软件、 SPSS 软件. 这类书籍基本上相当于软件使用说明书，虽然谈到一些统计概念，但讲的很少.

本书既不是单纯的一本关于数理统计或统计计算的教科书，也不只是一本关于 R 软件的使用手册，而是一本将两者相结合的教科书. 本书的特点是结合 R 软件来讲数理统计的基本概论与计算方法.

R 软件是一种统计软件，也是一种数学计算环境. 它提供了有弹性的、互动的环境来分析、可视及展示数据；它提供了若干统计程序包，以及一些集成的统计工具和各种数学计算、统计计算的函数，用户只需根据统计模型，指定相应的数据库及相关的参数，便可灵活机动的进行数据分析等工作，甚至创造出符合需要的新的统计计算方法. 使用 R 软件可以简化你的数据分析过程，从数据的存取，到计算结果的分享， R 软件提供了更加方便的计算工具，帮助你更好地分析和解决问题. 通过 R 软件的许多内嵌统计函数，用户可以很容易学习和掌握 R 软件的语法，也可以编制自己的函数来扩展现有的 R 语言，完成你的科研工作.

本教材的编写风格是： (1) 以目前常见的数理统计教材的内容为基准，首先对数理统计的基本概念、基本方法作一个简单、清晰的介绍，在注重基础的同时，侧重统计思想和统计方法的介绍. (2) 以 R 语言为主，编写相应的计算程序. 这部分内容的目的有两个，第一是学习 R 软件的编程方法，掌握 R 软件的基本技巧. 第二是通过编程加深对统计方法的了解与掌握，同时，还可以通过编

程，加深对 R 软件中相关函数的了解. (3) 介绍相关的计算函数. 针对许多统计方法， R 软件提供了大量的内嵌计算函数，使用者只需输入数据，就可得到相应的结果. 这一部分的写作重点是放在对计算结果的统计解释，如何通过结果来分析已有的数据，着重掌握相应的统计方法. 这些是本教材最主要的特色，也是不同于其他与软件有关的教材. 本书着重强调统计建模，以及如何使用 R 软件得到其计算结果和相应的结果解释.

本书的主要内容：第一章，概率统计的基本知识. 主要目的是复习统计的基本知识，便于对后面各章内容的理解. 第二章， R 软件的使用. 主要介绍 R 软件的基本使用方法. 第三章，数据描述性分析. 从数据描述开始分析数据，主要介绍数据的基本特征，如均值、方差，还有与数据有关的各种图形，如直方图、散点图等. 第四章，参数估计. 介绍参数估计的基本方法，如点估计和区间估计. 着重介绍 R 软件中与估计有关的函数. 第五章，假设检验. 介绍假设检验的基本方法，一类是参数检验；另一类是非参数检验. 非参数检验是该章的主要内容，重点介绍 R 软件中与非参数检验的各类函数和使用方法. 第六章，回归分析. 介绍回归分析的基本方法，着重介绍回归分析的过程与方法和如何使用 R 软件作回归分析. 除一般的回归方法外，还谈到逐步回归、非线性回归的等内容. 第七章，方差分析. 介绍单因素方差分析、双因素方差分析，以及正交试验设计与方差分析之间的关系. 第八章，应用多元分析（ I ）. 介绍判别分析和聚类分析，这些内容与判别和分类有关. 第九章，应用多元分析（ II ）. 介绍主成分分析、主因子分析和典型相关分析，它是应用多元分析中降维计算的内容. 第十章，计算机模拟. 介绍与计算机模拟的 Monte Carlo 方法，以及系统模拟方法，最后介绍模拟方法在排队论中的应用.

在学习本书的内容之后，你会发现，尽管有些统计内容其计算是相当复杂的，但在使用 R 软件之后，这些问题可以很轻松地得到解决.

本书所编写的 R 函数，以及所介绍的 R 函数均以 R-2.1.1 版为基础（目前的版本是 R-2.3.1, 而且大约每 3 至 4 个月版本会更新一次), 而且全部程序均运行通过，读者如果需要作者自编的 R 程序，可以发电子邮件向作者索取，邮件地址： xueyi@bjut.edu.cn.

本书是为理工、经济、管理、生物等专业学生或专业人员为解决统计计算问题而编写，可以作为上述专业学生数理统计课程的辅导教材或教学参考书，也作为统计计算课程的教材，和数学建模竞赛的辅导教材.

由于受编者水平所限，书中一定存在不足甚至错误之处，欢迎读者不吝指正，我们电子邮件地址是： xueyi@bjut.edu.cn (薛毅); chenliping@bjut.edu.cn (陈立萍).

编 者

2006 年 7 月

于北京工业大学

目 录

第一章　概率统计的基本知识

本书是一本统计建模与软件应用相结合的教科书，其讲述重点放在数理统计的基本方法和用 R 软件进行相应的计算．众所周知，数理统计是以概率论为基础、应用非常广泛的数学学科分支，是通过对试验或观察数据进行分析，来研究随机现象以达到对研究对象的客观规律性做出合理的估计和推断的目的，因此在介绍统计建模和 R 软件知识之前，有必要先回顾一下相关的概率与数理统计的基本概念，以及数理统计的各个应用分支．

本章用四节的内容简单回顾概率论的基础知识，用一节的内容简单介绍数理统计的基本概念．这样做的目的是使读者对已有概率论的知识有一个全面的了解与回顾，对数理统计的概念有一个基本的认识．

1.1　随机事件与概率

1.1.1　随机事件

1．随机事件

在一定条件下，所得的结果不能预先完全确定，而只能确定是多种可能结果中的一种，称这种现象为随机现象．例如，抛掷一枚硬币，其结果有可能是出现正面，也有可能是出现反面；电话交换台在 1 分钟内接到的呼叫次数，可能是 0 次、1 次、2 次、$\cdots$; 在同一工艺条件下生产出的灯泡，其使用寿命有长有短；测量同一物体的长度时，由于仪器及观察受到环境的影响，多次测量的结果往往有差异，等等．这些现象都是随机现象．

使随机现象得以实现和对它观察的全过程称为随机试验 (random experiment), 记为 E. 随机实验满足以下条件：

(1) 可以在相同条件下重复进行；

(2) 结果有多种可能性，并且所有可能结果事先已知；

(3) 作一次试验究竟哪个结果出现，事先不能确定．

称随机试验的所有可能结果组成的集合为样本空间 (sample space), 记为 Ω. 试验的每一个可能结果称为样本点 (sample point), 记为 ω.

称 Ω 中满足一定条件的子集为随机事件 (random event), 用大写字母 A, B, C, $\cdots$ 表示.

一个随机事件只含一个不可再分的试验结果称为一个基本事件 (即一个样本点所作成的集合 $\{\omega\}$).

在试验中, 称一个事件发生是指构成该事件的一个样本点出现. 由于样本空间 Ω 包含了所有的样本点, 所以在每次试验中, 它总是发生, 因此称 Ω 为必然事件 (certain event). 空集 $\emptyset$ 不包含任何样本点, 且在每次试验中总不发生, 所以称 $\emptyset$ 为不可能事件 (impossible event).

2. 随机事件之间的关系

若事件 A 的发生必然导致事件 B 的发生, 则称事件 A 包含于事件 B, 或事件 B 包含事件 A, 记为 $A \subset B$, 亦称为事件的包含 (contain) 关系.

若 $A \subset B$, 且 $B \subset A$, 则称事件 A 与事件 B 等价 (equivalent), 记为 $A = B$.

若事件 A 与事件 B 至少有一个发生, 则称为事件的和 (union), 记为 $A \cup B$. 若 n 个事件 A_1, A_2, $\cdots$, A_n 中至少有一个发生, 则称为 n 个事件的和, 记为 $A_1 \cup A_2 \cup \cdots \cup A_n$ 或 $\bigcup\limits_{i=1}^{n} A_i$.

同样, 可以定义可列无穷个事件的和 $A_1 \cup A_2 \cup \cdots \cup A_n \cup \cdots$ 或 $\bigcup\limits_{i=1}^{\infty} A_i$, 表示无穷个事件中至少有一个发生.

若事件 A 发生而事件 B 不发生, 则称为事件 A 与事件 B 的差, 记为 $A - B$.

若事件 A 与 B 同时发生, 则称事件 A 与事件 B 的积 (intersection), 记为 $A \cap B$ 或 AB. 若 n 个事件 A_1, A_2, $\cdots$, A_n 同时发生, 则称为 n 个事件的积, 记为 $A_1 \cap A_2 \cap \cdots \cap A_n$ 或 $\bigcap\limits_{i=1}^{n} A_i$.

同样, 可以定义可列无穷个事件的积 $A_1 \cap A_2 \cap \cdots \cap A_n \cap \cdots$ 或 $\bigcap\limits_{i=1}^{\infty} A_i$, 表示无穷个事件同时发生.

若事件 A 与 B 不能同时发生, 则称事件 A 与事件 B 为互斥事件 (mutually exclusive event) 或不相容事件 (incompatiable event), 记为 $AB = \emptyset$.

在一次试验中, 基本事件之间是两两互斥的.

若 A 为随机事件, 称“事件 A 不发生”的事件为事件 A 的对立事件 (opposite event) 或逆事件 (complementary event), 记为 $\overline{A}$. 事件与对其立事件有如下关

系：

$$A \cup \overline{A} = \Omega, \quad A\overline{A} = \emptyset.$$

由定义可知：对立事件一定是互斥事件，但互斥事件不一定是对立事件.

3. 随机事件的运算律

(1) 交换律

$$A \cup B = B \cup A, \quad AB = BA. \tag{1.1}$$

(2) 结合律

$$(A \cup B) \cup C = A \cup (B \cup C), \quad (A \cap B) \cap C = A \cap (B \cap C). \tag{1.2}$$

(3) 分配律

$$(A \cup B)C = (AC) \cup (BC), \quad A \cup (BC) = (A \cup B)(A \cup C). \tag{1.3}$$

(4) 德 · 摩根律

$$\overline{A_1 \cup A_2} = \overline{A}_1 \cap \overline{A}_2, \quad \overline{A_1 \cap A_2} = \overline{A}_1 \cup \overline{A}_2. \tag{1.4}$$

对于 n 个或可列无穷个事件有

$$\overline{\bigcup_{k=1}^{n} A_k} = \bigcap_{k=1}^{n} \overline{A}_k, \quad \overline{\bigcap_{k=1}^{n} A_k} = \bigcup_{k=1}^{n} \overline{A}_k, \quad \overline{\bigcup_{k=1}^{\infty} A_k} = \bigcap_{k=1}^{\infty} \overline{A}_k, \quad \overline{\bigcap_{k=1}^{\infty} A_k} = \bigcup_{k=1}^{\infty} \overline{A}_k. \tag{1.5}$$

(5) 减法满足

$$A - B = A\overline{B} \quad \text{或} \quad A - B = A \cap \overline{B}. \tag{1.6}$$

1.1.2 概率

1. 概率的公理化定义

在概率论中并非样本空间 Ω 的任何子集均可以看作事件，所定义的事件之间应满足一定的代数结构.

定义 1.1 设随机试验 E 的样本空间为 Ω, $\mathcal{F}$ 是 Ω 的子集组成的集族，满足

(1) $\Omega \in \mathcal{F}$;

(2) 若 $A \in \mathcal{F}$, 则 $\overline{A} \in \mathcal{F}$; (对逆运算封闭)

(3) 若 $A_i \in \mathcal{F}$, $i = 1, 2, \cdots$, 则 $\bigcup\limits_{i=1}^{\infty} A_i \in \mathcal{F}$. (对可列并运算封闭)

则称 $\mathcal{F}$ 为 Ω 的一个 σ- 代数 (事件体)， $\mathcal{F}$ 中的集合称为事件. 样本空间 Ω 和 σ 代数的二元体 $(\Omega, \mathcal{F})$ 称为可测空间.

定义 1.2 随机试验 E 的样本空间为 Ω, $(\Omega, \mathcal{F})$ 是可测空间, 对于每个事件 $A \in \mathcal{F}$, 定义一个实数 $P(A)$ 与之对应，若函数 $P(\cdot)$ 满足条件:

(1) 对每个事件 A, 均有 $0 \leq P(A) \leq 1$;

(2) $P(\Omega) = 1$;

(3) 若事件 A_1, A_2, $\cdots$ 两两互斥，即对于 $i, j = 1, 2, \cdots$, $i \neq j$, $A_i A_j = \emptyset$ 均有

$$P(A_1 \cup A_2 \cup \cdots) = P(A_1) + P(A_2) + \cdots,$$

则称 $P(A)$ 为事件 A 的概率 *(probability)*, 称 $(\Omega, \mathcal{F}, P)$ 为概率空间.

2. 概率的性质

性质 1: $P(\emptyset) = 0$, 即不可能事件的概率为零.

但性质反过来不成立，即 $P(A) = 0 \not\Rightarrow A = \emptyset$.

性质 2: 若事件 $A_1, A_2, \cdots, A_n$ 两两互斥，则有

$$P(A_1 \cup A_2 \cup \cdots \cup A_n) = P(A_1) + P(A_2) + \cdots + P(A_n), \tag{1.7}$$

即互斥事件和的概率等于它们各自概率的和.

性质 3: 对任一事件 A, 均有 $P(\overline{A}) = 1 - P(A)$.

性质 4: 对两个事件 A 和 B, 若 $A \subset B$, 则有

$$P(B - A) = P(B) - P(A), \quad P(B) \geq P(A). \tag{1.8}$$

性质 5: (加法公式) 对任意两个事件 A 和 B, 有

$$P(A \cup B) = P(A) + P(B) - P(AB). \tag{1.9}$$

性质 5 可以推广为：

$$\begin{aligned} P(A_1 \cup A_2 \cup A_3) &= P(A_1) + P(A_2) + P(A_3) - P(A_1A_2) - P(A_1A_3) \\ &\quad -P(A_2A_3) + P(A_1A_2A_3), \end{aligned} \tag{1.10}$$

$$P(A_1 \cup A_2 \cup \cdots \cup A_n) = S_1 - S_2 + S_3 - S_4 + \cdots + (-1)^{n-1} S_n, \tag{1.11}$$

其中 $S_1 = \sum\limits_{i=1}^{n} P(A_i)$, $S_2 = \sum\limits_{1 \le i < j \le n} P(A_iA_j)$, $S_3 = \sum\limits_{1 \le i < j < k \le n} P(A_iA_jA_k)$, $\cdots$, $S_n = P(A_1A_2 \cdots A_n)$.

1.1.3 古典概型

设随机事件 E 的样本空间中只有有限个样本点，即 $\Omega = \{\omega_1, \omega_2, \cdots, \omega_n\}$，其中 n 为样本点总数. 每个样本点 $\omega_i (i = 1, 2, \cdots, n)$ 出现是等可能的，并且每次试验有且仅有一个样本点发生，则称这类现象为古典概型 (classical probability model). 若事件 A 包含 m 个样本点，则事件 A 的概率定义为

$$P(A) = \frac{m}{n} = \frac{\text{事件 } A \text{ 包含的基本事件数}}{\text{基本事件总数}}. \tag{1.12}$$

例 1.1 设有 k 个不同的（可分辨）球，每个球都能以同样的概率 $1/l$ 落到 l 个格子 $(l \ge k)$ 的每一个中，且每个格子可容纳任意多个球，试分别求如下两事件 A 与 B 的概率.

A: 指定的 k 个格子中各有一个球;

B: 存在 k 个格子，其中各有一个球.

解：由于每个球可以落入 l 个格子中的任一个，并且每一个格子中可落入任意多个球，所以 k 个球落入 l 个格子中的分布情况相当于从 l 个格子中选取 k 个的可重复排列，故样本空间共有 l^k 种等可能的基本结果.

事件 A 所含基本结果数应是 k 个球在指定的 k 个格子中的全排列数，即 $k!$，所以

$$P(A) = \frac{k!}{l^k}.$$

为了算出事件 B 所含的基本事件数，可设想分两步进行：因为 k 个格子可以是任意选取的，故可先从 l 个格子中任意选出 k 个来，选法共有 C_l^k 种；对于

每种选定的 k 个格子，依上述各有一个球的推理，则有 $k!$ 个基本结果，故 B 含有 $C_l^k k!$ 个基本结果. 所以

$$P(B) = C_l^k \frac{k!}{l^k} = \frac{l!}{(l-k)!l^k}.$$

概率论的历史上有一个颇为著名的问题 — 生日问题：求 k 个同班同学没有两人生日相同的概率.

若把这 k 个同学看作例 1.1 中的 k 个球，而把一年 365 天看作格子，即 $l = 365$, 则上述的 $P(B)$ 就是所要求的概率. 例如, $k = 40$ 时, $P(B) = 0.109$. 或者换句话说，40 个同学中至少两个人同一天过生日的概率是： $P(\overline{B}) = 1 - 0.109 = 0.891$, 其概率大的出乎意料.

1.1.4 几何概型

当随机试验的样本空间是某一可度量的区域，并且任意一点落在度量 (长度、面积与体积) 相同的子区域内是等可能的，则事件 A 的概率定义为

$$P(A) = \frac{S_A}{S} = \frac{\text{构成事件 } A \text{ 的子区域的度量}}{\text{样本空间的度量}}. \tag{1.13}$$

这种概率模型称为几何概型 (geometric probability model).

例 1.2 (Buffon(蒲丰) 投针问题). *设平面上画有等距为 a 的一簇平行线. 取一枚长为 $l(l < a)$ 的针随意扔到平面上，求针与平行线相交的概率.*

解： 设 x 表示针的中心到最近一条平行线的距离， θ 表示针与此直线间的交角 (图 1.1(a)), 则 (θ, x) 完全决定针所落的位置. 针的所有可能的位置为

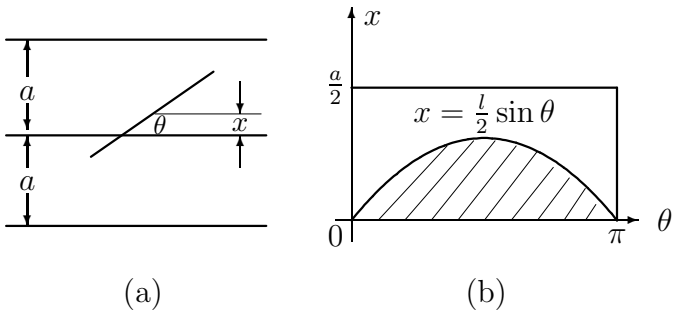
$$\Omega = \left\{(\theta, x) \ : \ 0 \le \theta \le \pi, \ 0 \le x \le \frac{a}{2}\right\}.$$

它可用 $\theta - x$ 平面上的一个矩形来表示 (图 1.1(b)). 针与平行线相交的充分必要条件是 $x \le \frac{l}{2}\sin\theta$, 即图 1.1(b) 中阴影部分，它的面积为

$$S_A = \int_0^\pi \frac{l}{2}\sin\theta d\theta = l.$$

因此，若把往平面上随意扔一枚针理解为 Ω 内的任一点为等可能，且记针与任一平行线相交的事件为 A, 则

$$P(A) = \frac{S_A}{S} = \frac{2l}{\pi a}. \tag{1.14}$$



(a) (b)

图 1.1: Buffon 投针的几何概率

由式 (1.14) 可以利用投针试验计算 π 值. 设随机投针 n 次，其中 k 次针线相交，当 n 充分大时，可用频率 $\frac{k}{n}$ 作为概率 p 的估计值，从而求得 π 的估计值为

$$\widehat{\pi} = \frac{2ln}{ak}. \tag{1.15}$$

根据公式 (1.15), 历史上曾有一些学者作了随机投针试验，并得到 π 的估计值.

1.1.5 条件概率

研究随机事件之间的关系时, 在已知某些事件发生的条件下考虑另一些事件发生的概率规律有无变化及如何变化，是十分重要的.

设 A 和 B 是两个事件，且 $P(B) > 0$, 称

$$P(A|B) = \frac{P(AB)}{P(B)} \tag{1.16}$$

为在事件 B 发生的条件下，事件 A 发生的条件概率 (conditional probability).

例如，某集体中有 N 个男人和 M 个女人，其中患色盲者男性 n 人，女性 m 人. 用 Ω 表示该集体， A 表示其中全体女性的集合， B 表示其中全体色盲者的集合. 如果从 Ω 中随意抽取一人，则这个人分别是女性、色盲者和同时既为女性又是色盲者的概率分别为

$$P(A) = \frac{M}{M+N}, \quad P(B) = \frac{m+n}{M+N}, \quad P(AB) = \frac{m}{M+N}.$$

如果限定只从女性中随机抽取一人 (即事件 A 已发生), 那么这个女人为色盲者的 (条件) 概率

$$P(B|A) = \frac{m}{M} = \frac{P(AB)}{P(A)}.$$

条件概率也是概率，它满足概率公理化定义中的三条，即

(1) 对每个事件 A, 均有 $0 \leq P(A|B) \leq 1$;

(2) $P(\Omega|B) = 1$;

(3) 若事件 $A_1, A_2, \cdots$, 两两互斥，即对于 $i, j = 1, 2, \cdots$, $i \neq j$, $A_iA_j = \emptyset$ 有

$$P\left((A_1 \cup A_2 \cup \cdots)|B\right) = P(A_1|B) + P(A_2|B) + \cdots,$$

并且对于在前面给出的概率性质和公式，也都适用于条件概率. 例如，对任意的事件 A_1, A_2, 有

$$P\left((A_1 \cup A_2)|B\right) = P(A_1|B) + P(A_2|B) - P(A_1A_2|B).$$

1.1.6 概率的乘法公式、全概率公式、 Bayes 公式

由条件概率公式，得

$$P(AB) = P(A|B)P(B) = P(B|A)P(A). \tag{1.17}$$

称式 (1.17) 为概率的乘法公式 (multiplication formula).

乘法公式的推广：对于任何正整数 $n \geq 2$, 当 $P(A_1A_2 \cdots A_{n-1}) > 0$ 时，有

$$P(A_1A_2 \cdots A_{n-1}A_n) = P(A_1)P(A_2|A_1)P(A_3|A_1A_2) \cdots P(A_n|A_1A_2 \cdots A_{n-1}). \tag{1.18}$$

定义 1.3 如果事件组 $B_1, B_2, \cdots$ 满足

(1) $B_1, B_2, \cdots$ 两两互斥, 即 $B_i \bigcap B_j = \emptyset$, $i \neq j$, $i, j = 1, 2, \cdots$, 且 $P(B_i) > 0$, $i = 1, 2, \cdots$.

(2) $B_1 \cup B_2 \cup \cdots = \Omega$,

则称事件组 $B_1, B_2, \cdots$ 是样本空间 Ω 的一个划分.

设 $B_1, B_2, \cdots$ 是样本空间 Ω 的一个划分， A 为任一事件，则

$$P(A) = \sum_{i=1}^{\infty} P(B_i)P(A|B_i). \tag{1.19}$$

称式 (1.19) 为全概率公式 (formula of total probability).

设 $B_1, B_2, \cdots$ 是样本空间 Ω 的一个划分，则对任一事件 A $(P(A) > 0)$, 有

$$P(B_i|A) = \frac{P(B_iA)}{P(A)} = \frac{P(B_i)P(A|B_i)}{\sum\limits_{j=1}^{\infty} P(B_j)P(A|B_j)}, \quad i = 1, 2, \cdots, \tag{1.20}$$

称式 (1.20) 为 Bayes (贝叶斯) 公式 (Bayes formula), 称式中的 $P(B_i)(i = 1, 2, \cdots)$ 为先验概率，称 $P(B_i|A)\,(i = 1, 2, \cdots)$ 为后验概率.

在实际中，常取对样本空间 Ω 的有限划分 $B_1, B_2, \cdots, B_n$（例如 B 与 $\overline{B}$ 就构成样本空间 Ω 的一个划分）. B_i 常被视为导致试验结果 A 发生的“原因”，而 $P(B_i)$ 表示各种“原因”发生的可能性大小，故称为先验概率； $P(B_i|A)$ 则反应当试验产生了结果 A 之后，再对各种“原因”概率的新认识，故称为后验概率.

例 1.3 假定用血清甲胎蛋白法诊断肝癌. 用 C 表示被检验者有肝癌这一事件，用 A 表示被检验者为阳性反应这一事件. 设 $P(A|C) = 0.95$, $P(\overline{A}|\overline{C}) = 0.90$. 若某人群中 $P(C) = 0.0004$, 现有一人呈阳性反应，求此人确为肝癌患者的概率 $P(C|A)$.

解：由 Bayes 公式，有

$$\begin{aligned} P(C|A) &= \frac{P(C)P(A|C)}{P(C)P(A|C) + P(\overline{C})P(A|\overline{C})} \\ &= \frac{0.0004 \times 0.95}{0.0004 \times 0.95 + 0.9996 \times 0.10} = 0.0038. \end{aligned}$$

1.1.7 独立事件

如果两事件 A, B 的积事件发生的概率等于这两个事件的概率的乘积，即

$$P(AB) = P(A)P(B),$$

则称事件 A 与事件 B 是相互独立的 (mutually independent).

性质：若事件 A 与事件 B 相互独立，则 A 与 $\overline{B}$, $\overline{A}$ 与 B, $\overline{A}$ 与 $\overline{B}$ 也相互独立.

推广：设 $A_1, A_2, \cdots, A_n$ 为 n 个事件，$n \geq 2$. 如果对于其中的任意 $k\,(k \geq 2)$ 个事件 $A_{i_1}, A_{i_2}, \cdots, A_{i_k}$, $1 \leq i_1 \leq i_2 \leq \cdots \leq i_k \leq n$, 等式

$$P(A_{i_1}A_{i_2}\cdots A_{i_k}) = P(A_{i_1})P(A_{i_2})\cdots P(A_{i_k})$$

均成立，则称 n 个事件 $A_1, A_2, \cdots, A_n$ 相互独立.

多个相互独立事件有如下性质：

(1) 若事件 $A_1, A_2, \cdots, A_n$ 相互独立，则 $A_1, A_2, \cdots, A_n$ 中任意 $k\,(k \geq 2)$ 个事件 $A_{i_1}, A_{i_2}, \cdots, A_{i_k}$, $1 \leq i_1 \leq i_2 \leq \cdots \leq i_k \leq n$, 也相互独立;

(2) 若事件 $A_1, A_2, \cdots, A_n$ 相互独立，则事件 $B_1, B_2, \cdots, B_n$ 也相互独立. 其中 B_i 或为 A_i 或为 $\overline{A}_i$, $i = 1, 2, \cdots, n$.

注意: 若 $A_1, A_2, \cdots, A_n$ 相互独立则有 $A_1, A_2, \cdots, A_n$ 两两相互独立, 反过来若 $A_1, A_2, \cdots, A_n$ 两两相互独立则不一定有 $A_1, A_2, \cdots, A_n$ 相互独立. 事实上，n 个事件相互独立，则要有 $C_n^2 + C_n^3 + \cdots + C_n^n = 2^n - n - 1$ 个等式成立，而两两独立只需有 $C_n^2 = \frac{n(n-1)}{2}$ 个等式成立.

例 1.4 设有 *4* 张卡片，其中 *3* 张上分别记有字母 *A* 和 *B, B* 和 *C, A* 和 *C,* 第 *4* 张是空白. 从中随机抽取一张，就用 *A, (B* 和 *C)* 分别记事件 *"*抽到的卡片上有字母 *A, (B* 和 *C)", *则显然有

$$\begin{aligned} P(A) &= P(B) = P(C) = \frac{1}{2}, \\ P(AB) &= P(AC) = P(BC) = \frac{1}{4}, \\ P(ABC) &= 0 \neq P(A)P(B)P(C). \end{aligned}$$

因此， *A, B, C* 三个事件中任意两个相互独立，但这三个事件并不相互独立.

1.1.8 n 重 Bernoulli 试验及其概率计算

如果一个随机试验只有两种可能的结果 A 和 $\overline{A}$, 并且

$$P(A) = p, \quad P(\overline{A}) = 1 - p = q,$$

其中 $0 < p < 1$, 则称此试验为 Bernoulli (伯努利) 试验 (Bernoulli trial). Bernoulli 试验独立重复进行 n 次，称为 n 重 Bernoulli 试验.

例如, 从一批产品中检验次品，在其中进行有放回抽样 n 次，抽到次品称为 "成功", 抽到正品称为 "失败", 这就是 n 重 Bernoulli 试验.

设

$$A_k = \{n \text{ 重 Bernoulli 试验中 } A \text{ 出现 } k \text{ 次}\},$$

则

$$P(A_k) = C_n^k p^k (1-p)^{n-k}, \quad k = 0, 1, 2, \cdots, n. \tag{1.21}$$

这就是著名的二项分布，常记作 $B(n, k)$.

1.2 随机变量及其分布

1.2.1 随机变量的定义

定义 1.4 设 E 是随机试验， Ω 是样本空间，如果对于每一个 $\omega \in \Omega$, 都有一个确定的实数 $X(\omega)$ 与之对应，若对于任意实数 $x \in R$, 有 $\{\omega : X(\omega) < x\} \in F$, 则称 Ω 上的单值实函数 $X(\omega)$ 为一个随机变量 *(random variable).*

从定义可知随机变量是定义在样本空间 Ω 上，取值在实数域上的函数. 由于它的自变量是随机试验的结果，而随机试验结果的出现具有随机性，因此，随机变量的取值也具有一定的随机性. 这是随机变量与普通函数的不同之处.

1.2.2 随机变量的分布函数

描述一个随机变量，不仅要说明它能够取那些值，而且还要关心它取这些值的概率. 因此，引入随机变量的分布函数的概念.

定义 1.5 设 X 是一个随机变量，对任意的实数 x, 令

$$F(x) = P\{X \le x\}, \quad x \in (-\infty, +\infty), \tag{1.22}$$

则称 $F(x)$ 为随机变量 X 的分布函数 *(distribution function),* 也称为概率累积函数 *(probability cumulative function).*

从直观上看, 分布函数 $F(x)$ 是一个定义在 $(-\infty, +\infty)$ 上的实值函数， $F(x)$ 在点 x 处取值为随机变量 X 落在区间 $(-\infty, x]$ 上的概率.

分布函数具有以下性质

(1) $0 \le F(x) \le 1$;

(2) $F(x)$ 是单调不减函数，即当 $x_1 < x_2$ 时， $F(x_1) \le F(x_2)$;

(3) $F(-\infty) = \lim\limits_{x \to -\infty} F(x) = 0$, $F(+\infty) = \lim\limits_{x \to +\infty} F(x) = 1$;

(4) $F(x)$ 是右连续的函数，即 $\lim\limits_{x \to x_0^+} F(x) = F(x_0)$, $\forall\, x_0 \in R$ 均成立;

(5) $P\{a < X \leq b\} = F(b) - F(a)$;

(6) $P\{X > a\} = 1 - P\{X \leq a\} = 1 - F(a)$.

在理论上已经证明：如果一个函数满足上述的前四条性质，则它一定是某个随机变量的分布函数.

1.2.3 离散型随机变量

1. 离散型随机变量

定义 1.6 如果随机变量 X 的全部可能取值只有有限多个或可列无穷多个，则称 X 为离散型随机变量.

定义 1.7 对于离散型随机变量 X 可能取值为 x_k 的概率为：

$$P\{X = x_k\} = p_k, \quad k = 1, 2, \cdots, \tag{1.23}$$

则称式 *(1.23)* 为离散型随机变量 X 的分布律.

离散型随机变量的分布律具 p_k 有以下性质：

(1) $p_k \geq 0$, $k = 1, 2, \cdots$;

(2) $\sum\limits_{k=1}^{\infty} p_k = 1$.

可用表 1.1 来表示其分布律.

表 1.1: 分布律

X	x_1	x_2	$\cdots$	x_k	$\cdots$
p_k	p_1	p_2	$\cdots$	p_k	$\cdots$

离散型随机变量的分布函数为

$$F(x) = P\{X \leq x\} = \sum_{x_k \leq x} P\{X = x_k\} = \sum_{x_k \leq x} p_k. \tag{1.24}$$

2. 常见的离散型分布

(1) 两点分布 (0−1 分布)

若随机变量 X 的分布律为：

$$P\{X = k\} = p^k (1-p)^{1-k}, \quad k = 0, 1, \ \ (0 < p < 1), \tag{1.25}$$

则称 X 服从参数为 p 的两点分布，记作 $X \sim B(1,p)$. 其分布函数为

$$F(x)=\begin{cases}0, & x<0,\\ 1-p, & 0\le x<1,\\ 1, & x\ge 1.\end{cases} \tag{1.26}$$

(2) Bernoulli 试验，二项分布

若随机变量 X 的分布律为

$$P\{X=k\}=C_n^k p^k(1-p)^{n-k}, \quad k=0,1,\cdots,n, \tag{1.27}$$

则称 X 服从参数为 n,p 的二项分布 (binomial distribution), 记为 $X\sim B(n,p)$, 其中 $C_n^k p^k(1-p)^{n-k}$ 是 n 重 Bernoulli 试验中事件 A 恰好发生 k 次的概率. 其分布函数为

$$F(x)=\sum_{k=0}^{\lfloor x\rfloor} C_n^k p^k(1-p)^{n-k}, \tag{1.28}$$

其中 $\lfloor x\rfloor$ 表示下取整，即不超过 x 的最大整数，下同.

(3) Poisson 分布

若随机变量 X 的分布律为

$$P\{X=k\}=\frac{\lambda^k \mathrm{e}^{-\lambda}}{k!}, \quad k=0,1,2,\cdots, \tag{1.29}$$

则称 X 服从参数为 λ 的 Poisson (泊松) 分布 (Poisson distribution), 记作 $X\sim P(\lambda)$ 或 $X\sim \pi(\lambda)$, 其中 $\lambda>0$ 为常数. 其分布函数为

$$F(x)=\sum_{k=0}^{\lfloor x\rfloor}\frac{\lambda^k\,\mathrm{e}^{-\lambda}}{k!}. \tag{1.30}$$

定理 1.1 (Poisson 定理)

在 Bernoulli 试验中，以 p_n 代表事件 A 在试验中出现的概率，它与试验总数 n 有关，如果 $np_n\to\lambda$, 则当 $n\to\infty$ 时，有

$$\lim_{n\to\infty} C_n^k p_n^k(1-p_n)^{n-k}=\frac{\lambda^k \mathrm{e}^{-\lambda}}{k!}. \tag{1.31}$$

当 n 很大且 p_n 很小时，二项分布可以用 Poisson 分布来近似代替，即

$$C_n^k p_n^k(1-p_n)^{n-k}\approx\frac{\lambda^k \mathrm{e}^{-\lambda}}{k!}, \tag{1.32}$$

其中 $\lambda=np_n$.

1.2.4 连续型随机变量

1. 连续型随机变量

定义 1.8 对于随机变量 X, 如果存在一个定义在 $(-\infty, +\infty)$ 上的非负函数 $f(x)$, 使得对于任意实数 x ，总有

$$F(x) = P\{X \le x\} = \int_{-\infty}^{x} f(t)\,\mathrm{d}t, \qquad -\infty < x < +\infty, \tag{1.33}$$

则称 X 为连续型随机变量， $f(x)$ 为 X 的概率密度函数 *(probability density function)*, 简称概率密度.

概率密度函数有如下性质：

(1) $\int_{-\infty}^{+\infty} f(x)dx = 1$;

(2) 对于任意的实数 $a, b(a < b)$, 都有 $P\{a < X \le b\} = \int_a^b f(x)\mathrm{d}x$;

(3) 若 $f(x)$ 在点 x 处连续, 则 $f(x) = F'(x)$;

(4) 对任意实数 a, 总有 $P\{X = a\} = 0$.

2. 常见的连续型分布

(1) 均匀分布

若随机变量 X 的概率密度函数为

$$f(x) = \begin{cases} \dfrac{1}{b-a}, & a \le x \le b, \\ 0, & \text{其它}, \end{cases} \tag{1.34}$$

则称 X 服从区间 $[a, b]$ 上的均匀分布 (uniform distribution), 记为 $X \sim U[a, b]$. 其分布函数为

$$F(x) = \begin{cases} 0, & x < a, \\ \dfrac{x-a}{b-a}, & a \le x < b, \\ 1, & x \ge b. \end{cases} \tag{1.35}$$

(2) 指数分布

若随机变量 X 的概率密度函数为

$$f(x) = \begin{cases} \lambda\,\mathrm{e}^{-\lambda x}, & x \ge 0, \\ 0, & x < 0, \end{cases} \tag{1.36}$$

其中 $\lambda > 0$ 为常数, 则称 X 服从参数为 λ 的指数分布 (exponential distribution). 其分布函数为

$$F(x) = \begin{cases} 1 - \mathrm{e}^{-\lambda x}, & x \geq 0, \\ 0, & x < 0. \end{cases} \tag{1.37}$$

(3) 正态分布

若随机变量 X 的概率密度函数为

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{(x-\mu)^2}{2\sigma^2} \right\}, \quad -\infty < x < +\infty, \tag{1.38}$$

其中 $\mu, \sigma(\sigma > 0)$ 是两个常数, 则称 X 服从参数为 μ, σ 的正态分布 (normal distribution), 也称为 Gauss 分布, 记作 $X \sim N(\mu, \sigma^2)$.

图 1.2 描绘的是参数为 $\mu = 0, \sigma = 1$, $\mu = 0, \sigma = 0.5$ 和 $\mu = 2, \sigma = 0.5$ 的正态分布的概率密度函数图.

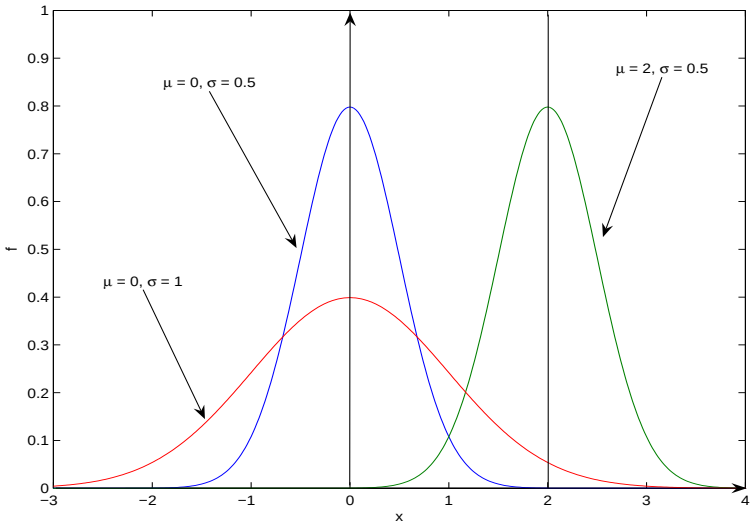


图 1.2: 正态分布的概率密度函数

如果改变 μ 值, 只会改变正态分布图形的位置, 而不会改变它的形状. 如果改变 σ 值, 则会改变正态分布的形状. 例如, 在图 1.2 中, 可以看到, 改变 μ 值, 实际上在改变正态分布的中心位置, μ 值变小, 图形向左移动, μ 值变大, 图形向右移动. 而改变 σ, 则改变图形的形状, σ 的值越小, 其图形越陡; 而 σ 越大, 则图形越平坦. 当我们讲过数学期望与方差的意义、正态随机变量的数学期望与方差后, 更容易理解这一点.

当 $\mu=0,\sigma=1$ 时，$X\sim N(0,1)$, 则称 X 服从标准正态分布. 其概率密度函数为

$$\phi(x)=\frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{x^2}{2}\right\},\qquad -\infty<x<+\infty. \tag{1.39}$$

其分布函数为

$$\Phi(x)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}\mathrm{e}^{-\frac{t^2}{2}}\,\mathrm{d}t, \tag{1.40}$$

且 $\Phi(-x)=1-\Phi(x)$.

图 1.3 给出标准正态分布的概率密度曲线，以及对应区间上积分 (相应的

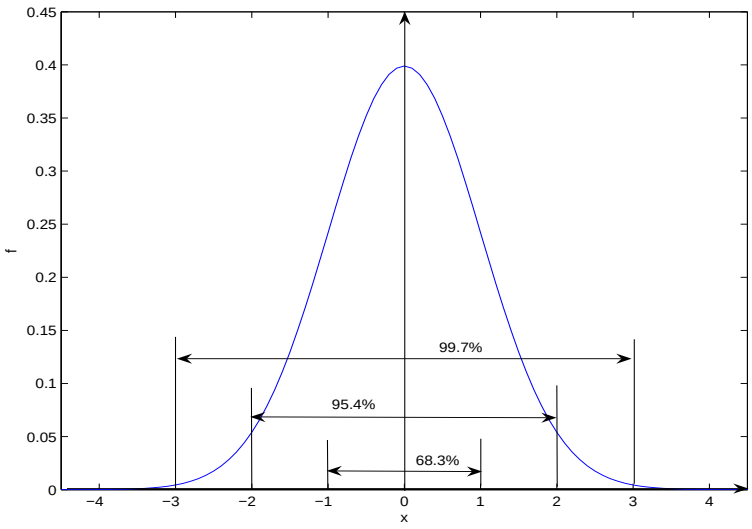


图 1.3: 标准正态分布和对应区间上积分 (面积) 的百分比

面积) 的百分比. 图 1.3 表明，当 $X\sim N(0,1)$ 时，$P\{-1\le X\le 1\}=0.683$, $P\{-2\le X\le 2\}=0.954$, $P\{-3\le X\le 3\}=0.997$, 这些数量指标在实际中是常用的，应该牢记.

这个概念可以推广到一般正态分布，也就是说，从 $\mu-3\sigma$ 到 $\mu+3\sigma$ 的区间上概率密度曲线之下的面积占总面积的 99.7%, 这就是著名的 3σ 原则.

若 $X\sim N(\mu,\sigma^2)$, 则

$$F(x)=\int_{-\infty}^{x}\frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\frac{(t-\mu)^2}{2\sigma^2}}\,\mathrm{d}t=\Phi\left(\frac{x-\mu}{\sigma}\right),\quad -\infty<x<+\infty. \tag{1.41}$$

图 1.4 给出了正态分布的概率密度函数与分布函数之间的关系，其中曲线为概

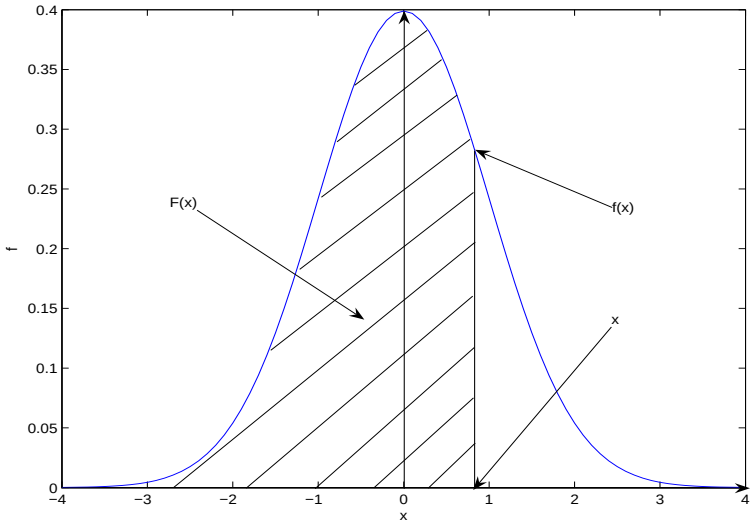


图 1.4: 概率密度函数与分布函数之间的关系

率密度函数 $f(x)$, 而阴影部分则是分布函数 $F(x)$. 由此容易得到

$$P\{x_1 < X \leq x_2\} = F(x_2) - F(x_1) = \Phi\left(\frac{x_2-\mu}{\sigma}\right) - \Phi\left(\frac{x_1-\mu}{\sigma}\right). \tag{1.42}$$

(图 1.4 中的概率密度函数是标准正态分布的概率密度函数).

设随机变量 $X \sim N(0,1)$, 对任给的 $0 < \alpha < 1$, 称满足条件

$$P\{X > Z_\alpha\} = \int_{Z_\alpha}^{+\infty} \phi(x)\,\mathrm{d}x = \alpha \tag{1.43}$$

的点 Z_α 为标准正态分布的上 α 分位点.

图 1.5 给出了标准正态分布的上 α 分位点 Z_α 的几何意义， 其中阴影部分面积的值为 α.

3. 随机变量的函数的分布

若随机变量 X 具有概率密度函数 $f_X(x)$, $-\infty < x < +\infty$, 又设 $g(x)$ 处处可导且 $g'(x)$ 不变号，则 $Y = g(X)$ 是连续型随机变量，其概率密度函数为：

$$f_Y(y) = \begin{cases} f_X\left(h(y)\right)|h'(y)|, & \alpha < y < \beta, \\ 0, & \text{其它}, \end{cases}$$

其中 $\alpha = \min\{g(-\infty), g(\infty)\}$, $\beta = \max\{g(-\infty), g(\infty)\}$, $x = h(y)$ 为 $y = g(x)$ 的反函数.

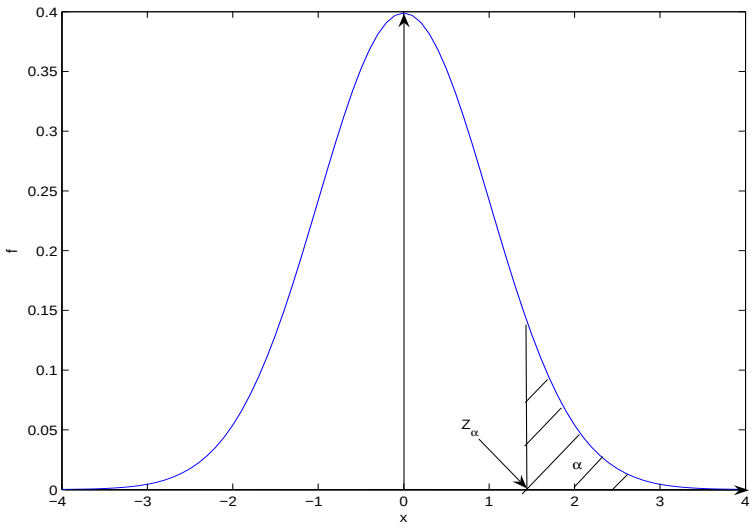


图 1.5: 标准正态分布的上 α 分位点

若 $g(x)$ 是非单调函数，设随机变量 X 的分布函数为 $F_X(x)$, 概率密度为 $f_X(x)$, $Y = g(X)$ 的分布函数为 $F_Y(y)$, 概率密度为 $f_Y(y)$, 则

$$F_Y(y) = \int_{g(x)\le y} dF_X(x).$$

由此再进一步求出 $f_Y(y)$, 不过需要具体问题具体分析.

1.2.5 随机向量

1. 随机向量的定义

定义 1.9 如果 X 和 Y 是定义在同一概率空间 $(\Omega, \mathcal{F}, P)$ 上的两个随机变量，称 (X, Y) 为二维随机向量 *(random vector)*, 并称 X 和 Y 是二维随机向量 (X, Y) 的两个分量.

二维随机向量 (X, Y) 是定义在样本空间 Ω 上, 取值于 R^2 上的函数. 类似, 可定义 n 维随机向量.

定义 1.10 设 Ω 为样本空间， $X_1 = X_1(\omega)$, $X_2 = X_2(\omega)$, $\cdots$, $X_n = X_n(\omega)$ 是 Ω 上的 n 个随机变量，则由它们构成的 n 维向量 $(X_1, X_2, \cdots, X_n)$ 称为 n 维随机向量 *($n-$dimensional random vector)*, 称 X_i 为 X 的第 i 个分量 *(component)*.

2. 随机向量的联合分布函数

定义 1.11 设 (X,Y) 是定义在 $(\Omega,\mathcal{F},P)$ 上的随机向量，对任意的 $(x,y)\in R^2$，二元函数

$$F(x,y)=P\{\omega:\ X(\omega)\le x,y(\omega)\le y\},\tag{1.44}$$

称为 (X,Y) 的联合分布函数 *(joint distribution function)*，其中 $\{X\le x,Y\le y\}$ 表示事件 $\{X\le x\}$ 与事件 $\{Y\le y\}$ 的积事件.

设 $X_1,X_2,\cdots,X_n$ 是一个 n 维随机向量，对任意的 $(x_1,x_2,\cdots,x_n)\in R^n$，$n$ 元函数

$$F(x_1,x_2,\cdots,x_n)=P\{\omega:\ X_1(\omega)\le x_1,X_2(\omega)\le x_2,\cdots,X_n(\omega)\le x_n,\},\tag{1.45}$$

称为 $(X_1,X_2,\cdots,X_n)$ 的联合分布函数.

3. 分布函数的性质

(1) 对于任意固定的 y，当 $x_2>x_1$ 时，$F(x_2,y)\ge F(x_1,y)$. 对于任意固定的 x，当 $y_2>y_1$ 时，$F(x,y_2)\ge F(x,y_1)$，即 $F(x,y)$ 对每个自变量是单调不减的.

(2) $0\le F(x,y)\le 1$，且对于任意固定的 y，$F(-\infty,y)=0$. 对于任意固定的 x，$F(x,-\infty)=0$，$F(-\infty,-\infty)=0$，$F(+\infty,+\infty)=1$.

(3) $F(x,y)=F(x+0,y)$，$F(x,y)=F(x,y+0)$，即 $F(x,y)$ 关于 x 右连续，也关于 y 右连续.

(4) 对于任意 (x_1,y_1)，(x_2,y_2)，$x_1<x_2$，$y_1<y_2$，下述不等式

$$F(x_2,y_2)-F(x_2,y_1)-F(x_1,y_2)+F(x_1,y_1)\ge 0$$

成立.

由以上性质可得以下结论.

随机点 (X,Y) 落在矩形域 $\{x_1<x\le x_2,y_1<y\le y_2\}$ 内的概率为

$$P\{x_1<X\le x_2,y_1<Y\le y_2\}=F(x_2,y_2)-F(x_2,y_1)-F(x_1,y_2)+F(x_1,y_1).\tag{1.46}$$

4. 离散型二维随机向量

定义 1.12 如果二维随机向量 $(X,\ Y)$ 的每个分量都是离散型随机变量，则称 $(X,\ Y)$ 是二维离散型随机向量.

定义 1.13 设二维离散型随机向量 (X,Y) 所有的可能取值为 (x_i,y_j), $i=1,2,\cdots$, $j=1,2,\cdots$ 的概率为：

$$P\{X=x_i,Y=y_j\}=p_{ij},\quad i,j=1,2,\cdots, \tag{1.47}$$

则称式 *(1.47)* 为离散型随机向量 (X,Y) 的分布律 *(*联合分布律*)*.

显然， p_{ij} $(i,j=1,2,\cdots)$ 满足以下两个条件：

(1) $p_{ij}\geq 0$, $i,j=1,2,\cdots$;

(2) $\sum\limits_i\sum\limits_j p_{ij}=1$.

离散型随机向量 (X,Y) 的分布函数为：

$$F(x,y)=\sum_{x_i\leq x,y_j\leq y} p_{ij},\quad \forall x,y\in R.$$

5. 连续型二维随机向量

定义 1.14 如果对于二维随机向量 (X,Y) 的分布函数 $F(x,y)$, 存在非负的函数 $f(x,y)$, 使对于任意的 x,y, 有

$$F(x,y)=\int_{-\infty}^{y}\int_{-\infty}^{x} f(u,v)\,\mathrm{d}u\,\mathrm{d}v, \tag{1.48}$$

则称 (X,Y) 是连续型的二维随机向量，函数 $f(x,y)$ 称为二维随机向量 (X,Y) 的概率密度函数.

概率密度函数有如下性质：

(1) $f(x,y)\geq 0$, $\forall\ x,y\in R$;

(2) $\displaystyle\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\,\mathrm{d}x\,\mathrm{d}y=F(+\infty,+\infty)=1$;

(3) 在 $f(x,y)$ 的连续点处有

$$\frac{\partial^2 F(x,y)}{\partial x\partial y}=f(x,y);$$

(4) 随机点 (X,Y) 落在平面区域 G 内的概率为

$$P\{(X,Y)\in G\}=\iint_G f(x,y)\,\mathrm{d}x\,\mathrm{d}y.$$

6. 边缘分布

X, Y 的边缘分布 (marginal distribution) 函数分别是

$$F_X(x) = P\{X \le x\} = P\{X \le x, Y < +\infty\} = F(x, +\infty), \tag{1.49}$$

$$F_Y(y) = P\{Y \le y\} = P\{X < +\infty, Y \le y\} = F(+\infty, y). \tag{1.50}$$

若 (X, Y) 为离散型随机向量, X 与 Y 的边缘分布律及边缘分布函数分别为

$$P\{X = x_i\} = \sum_{j=1}^{\infty} p_{ij} = p_{i\cdot}, \quad i = 1, 2, \cdots, \tag{1.51}$$

$$P\{Y = y_j\} = \sum_{i=1}^{\infty} p_{ij} = p_{\cdot j}, \quad j = 1, 2, \cdots, \tag{1.52}$$

$$F_X(x) = F(x, +\infty) = \sum_{x_i \le x} \sum_{j=1}^{\infty} p_{ij}, \tag{1.53}$$

$$F_Y(y) = F(+\infty, y) = \sum_{i=1}^{\infty} \sum_{y_j \le y} p_{ij}. \tag{1.54}$$

若 (X, Y) 为连续型随机向量, X 和 Y 的边缘概率密度分别为:

$$f_X(x) = \int_{-\infty}^{+\infty} f(x, y)dy, \tag{1.55}$$

$$f_Y(y) = \int_{-\infty}^{+\infty} f(x, y)dx, \tag{1.56}$$

其边缘分布函数分别为

$$F_X(x) = P\{X \le x\} = \int_{-\infty}^{x} \Big[\int_{-\infty}^{+\infty} f(x, y)dy\Big]dx = \int_{-\infty}^{x} f_X(x)dx, \tag{1.57}$$

$$F_y(y) = P\{Y \le y\} = \int_{-\infty}^{y} \Big[\int_{-\infty}^{+\infty} f(x, y)dx\Big]dy = \int_{-\infty}^{y} f_Y(y)dy. \tag{1.58}$$

7. 常见二维随机向量的分布

(1) 二维均匀分布

若 (X,Y) 具有如下概率密度函数

$$f(x,y)=\begin{cases}\dfrac{1}{A}, & (x,y)\in D,\\ 0, & \text{其它}.\end{cases} \tag{1.59}$$

其中 A 为平面区域 D 的面积值，则称此二维连续型随机向量 (X,Y) 在区域 D 内服从二维均匀分布.

(2) 二维正态分布 $N(\mu_1,\mu_2,\sigma_1^2,\sigma_2^2,\rho)$

如果 (X,Y) 具有如下概率密度函数

$$\begin{aligned} f(x,y) &= \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \\ &\cdot \exp\left\{-\frac{1}{2(1-\rho^2)}\left[\frac{(x-\mu_1)^2}{\sigma_1^2}-2\rho\frac{(x-\mu_1)(y-\mu_2)}{\sigma_1\sigma_2}+\frac{(y-\mu_2)^2}{\sigma_2^2}\right]\right\}, \\ & -\infty<x<+\infty,-\infty<y<+\infty. \end{aligned} \tag{1.60}$$

其中 μ_1, μ_2, $\sigma_1>0$, $\sigma_2>0$, $|\rho|<1$ 为实数，则称此二维连续型随机向量 (X,Y) 服从参数为 $\mu_1,\mu_2,\sigma_1,\sigma_2,\rho$ 的二维正态分布，记作 $(X,Y)\sim N(\mu_1,\mu_2,\sigma_1^2,\sigma_2^2,\rho)$，同时称 (X,Y) 为二维正态随机向量.

图 1.6 绘出了 ρ 取不同值的情况， 在图中 $\mu_1=0$, $\mu_2=0$, $\sigma_1=1$, $\sigma_2=2$. 当 $\rho=0$ 时，随机变量 X 与随机变量 Y 是独立的，当 $\rho\neq 0$ 时，随机变量 X 与随机变量 Y 相关 (不独立), 并且当 $|\rho|$ 越接近 1 时，相关程度越密切.

1.3 随机变量的数字特征

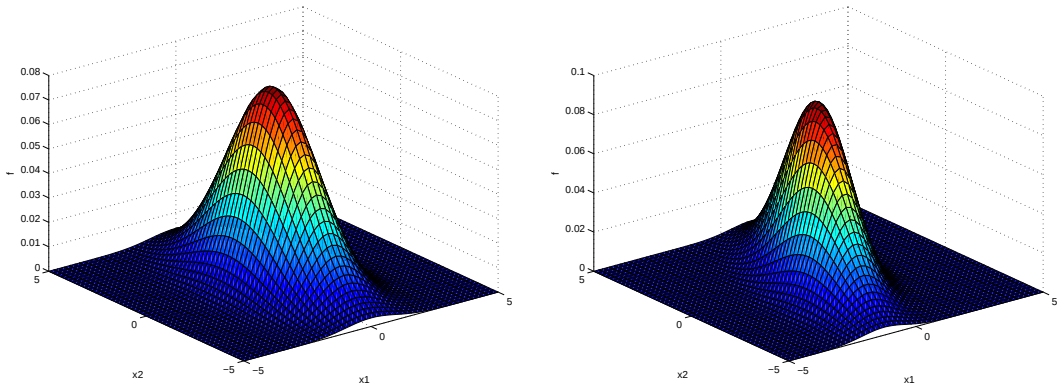
1.3.1 数学期望

定义 1.15 *设离散型随机变量 X 的分布律为 $P\{X=x_i\}=p_i$, $i=1,2,\cdots$, 若级数 $\sum\limits_i |x_i|p_i$ 收敛, 则称级数 $\sum\limits_i x_ip_i$ 的和为随机变量 X 的数学期望 (mathematical expectation), 记为 $E(X)$, 即*

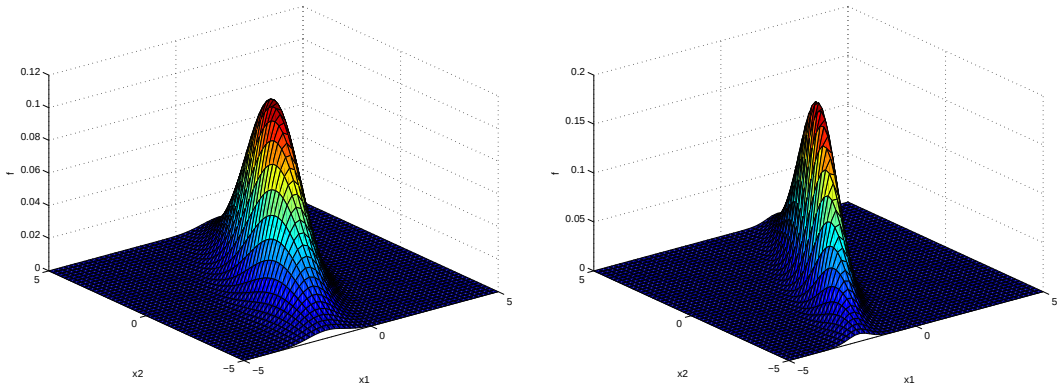
$$E(X)=\sum_i x_ip_i. \tag{1.61}$$

设连续型随机变量 X 的概率密度函数为 $f(x)$, 若积分 $\int_{-\infty}^{+\infty}|x|f(x)\mathrm{d}x$ 收敛，则称积分 $\int_{-\infty}^{+\infty}xf(x)\mathrm{d}x$ 的值为随机变量 X 的数学期望，记为 $E(X)$, 即

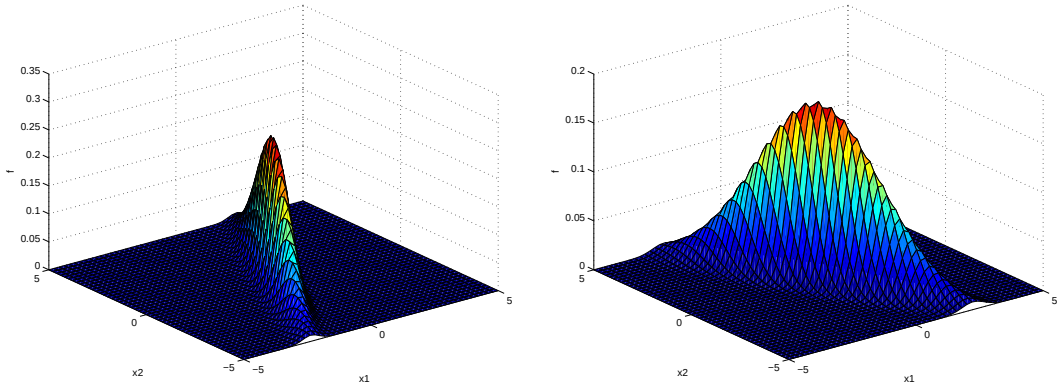
$$E(X)=\int_{-\infty}^{+\infty}xf(x)\mathrm{d}x. \tag{1.62}$$



$\rho = 0$ $\rho = 0.5$



$\rho = 0.7$ $\rho = 0.9$



$\rho = 0.95$ $\rho = -0.9$

图 1.6: 二维正态分布 ρ 取不同值的情况

$E(X)$ 又称为均值 *(mean).*

数学期望代表了随机变量取值的平均值, 是一个重要的数字特征. 数学期望具有如下性质:

(1) 若 c 是常数, 则 $E(c) = c$;

(2) $E(aX + bY) = aE(X) + bE(Y)$, 其中 a, b 为任意常数;

(3) 若 X, Y 相互独立, 则 $E(XY) = E(X)E(Y)$.

从数学期望的意义 (平均值), 很容易理解上述 3 条性质的意义.

如果 $X_1, X_2, \cdots, X_n$ 是 n 个随机变量, 反复运用性质 (2), 得到

$$E\left(\sum_{i=1}^{n} a_i X_i\right) = \sum_{i=1}^{n} a_i E(X_i), \tag{1.63}$$

其中 $a_i (i = 1, 2, \cdots, n)$ 是常数。

1.3.2 方差

定义 1.16 设 X 为随机变量, 如果 $E\{[X - E(X)]^2\}$ 存在, 则称 $E\{[X - E(X)]^2\}$ 为 X 的方差 *(variance),* 记为 $\mathrm{Var}(X)$, 即

$$\mathrm{Var}(X) = E\left\{[X - E(X)]^2\right\}, \tag{1.64}$$

并称 $\sqrt{\mathrm{Var}(X)}$ 为 X 的标准差 *(standard deviation)* 或均方差 *(root mean square).*

方差是用来描述随机变量取值相对于均值的离散程度的一个量, 也是非常重要的数字特征. 方差有如下性质:

(1) 若 c 是常数, 则 $\mathrm{Var}(c) = 0$;

(2) $\mathrm{Var}(aX + b) = a^2\mathrm{Var}(X)$, 其中 a, b 为任意常数;

(3) 如果 X, Y 相互独立, 则 $\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y)$.

从方差的意义 (离散程度), 很容易理解这 3 条性质的意义.

可以证明:

$$\mathrm{Var}(X) = E(X^2) - [E(X)]^2. \tag{1.65}$$

式 (1.65) 可作为方差的计算公式.

1.3.3 几种常用随机变量分布的期望与方差

(1) 若 X 服从参数为 p 的两点分布 $B(1,p)$, 其中 $0<p<1$, 则

$$E(X)=p,\quad \mathrm{Var}(X)=p(1-p). \tag{1.66}$$

(2) 若 X 服从参数为 n,p 的二项分布 $B(n,p)$, $0<p<1$, 则

$$E(X)=np,\quad \mathrm{Var}(X)=np(1-p). \tag{1.67}$$

(3) 若 X 服从参数为 λ 的 Poisson 分布 $P(\lambda)$, 则

$$E(X)=\lambda,\quad \mathrm{Var}(X)=\lambda. \tag{1.68}$$

(4) 若 X 服从参数为 a,b 的均匀分布 $U[a,b]$, 则

$$E(X)=\frac{a+b}{2},\quad \mathrm{Var}(X)=\frac{(b-a)^2}{12}. \tag{1.69}$$

(5) 若 X 服从参数为 λ 的指数分布，则

$$E(X)=\frac{1}{\lambda},\quad \mathrm{Var}(X)=\frac{1}{\lambda^2}. \tag{1.70}$$

(6) 若 X 服从参数为 μ,σ 的正态分布 $N(\mu,\sigma^2)$, 则

$$E(X)=\mu,\quad \mathrm{Var}(X)=\sigma^2. \tag{1.71}$$

由式 (1.71), 以及期望和方差的意义，可以进一步帮助我们理解图 1.2 的意义.

1.3.4 协方差与相关系数

1. 协方差

设 X,Y 为两个随机变量，称 $E\{[X-E(X)][Y-E(Y)]\}$ 为 X 和 Y 的协方差 (covariance), 记为 $\mathrm{Cov}(X,Y)$, 即

$$\mathrm{Cov}(X,Y)=E\{[X-E(X)][Y-E(Y)]\}. \tag{1.72}$$

协方差和下面介绍的相关系数都是描述随机变量 X 与随机变量 Y 之间的线性联系程度的数字量.

协方差具有如下基本性质：

(1) $\mathrm{Cov}(X,Y)=\mathrm{Cov}(Y,X)$;

(2) $\mathrm{Cov}(aX+b,cY+d)=ac\mathrm{Cov}(X,Y)$, 其中 a,b,c,d 为任意常数;

(3) $\mathrm{Cov}(X_1+X_2,Y)=\mathrm{Cov}(X_1,Y)+\mathrm{Cov}(X_2,Y)$;

(4) $\mathrm{Cov}(X,Y)=E(XY)-E(X)E(Y)$, 特别地，当 X 和 Y 相互独立时，有 $\mathrm{Cov}(X,Y)=0$;

(5) $|\mathrm{Cov}(X,Y)|\leq\sqrt{\mathrm{Var}(X)}\sqrt{\mathrm{Var}(Y)}$;

(6) $\mathrm{Cov}(X,X)=\mathrm{Var}(X)$.

如果 $X_1,X_2,\cdots,X_n$ 是 n 个随机变量，利用上述性质得到

$$\mathrm{Var}\left(\sum_{i=1}^{n}a_iX_i\right)=\sum_{i=1}^{n}\sum_{j=1}^{n}a_ia_j\mathrm{Cov}(X_i,X_j),\tag{1.73}$$

其中 $a_i(i=1,2,\cdots,n)$ 是常数。如果 $X_i(i=1,2,\cdots,n)$ 是 n 个相互独立的随机变量，则式 (1.73) 可改写为

$$\mathrm{Var}\left(\sum_{i=1}^{n}a_iX_i\right)=\sum_{i=1}^{n}a_i^2\mathrm{Var}(X_i).\tag{1.74}$$

2. 相关系数

当 $\mathrm{Var}(X)>0$, $\mathrm{Var}(Y)>0$ 时，称

$$\rho(X,Y)=\frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}}\tag{1.75}$$

为 X 与 Y 的相关系数 (coefficient of correlation), 它是无量纲的量. 其基本性质为：

(1) $|\rho(X,Y)|\leq 1$; $|\rho(X,Y)|=1$ 的充要条件为 X 与 Y 之间有线性关系，即存在常数 a,b $(a\neq 0)$ 使得

$$P\{Y=aX+b\}=1.$$

具体地 $a>0$ 时，对应 $\rho(X,Y)=1$; $a<0$ 时，对应 $\rho(X,Y)=-1$.

(2) 若 X 与 Y 相互独立且 $\mathrm{Var}(X)$, $\mathrm{Var}(Y)$ 存在，则 $\rho(X,Y)=0$; 特别地当 X 与 Y 均为正态分布时， X 与 Y 相互独立的充要条件为 $\rho(X,Y)=0$.

对于二维正态随机变量 X, Y, 其密度函数 (1.60) 中的 μ_1 表示 X 的均值, μ_2 表示 Y 的均值, σ_1^2 表示 X 的方差, σ_2^2 表示 Y 的方差, ρ 表示 X 与 Y 的相关系数. 这就是为什么在图 1.6 中, 当 $|\rho|$ 越接近于 1 时, 其图形越瘪.

1.3.5 矩与协方差矩阵

1. 矩

设随机变量 X 有分布函数 $F(x)$, 对任意给定的正整数 k, 若 $E(|X|^k)$ 存在, 则称

$$\alpha_k = E(X^k) = \int_{-\infty}^{\infty} x^k \mathrm{d}F(x) \tag{1.76}$$

为 X 的 k 阶原点矩 (moment about origin). 对于 $k > 1$, 若 $E(|X|^k)$ 存在, 则称

$$\mu_k = E\left([X - E(X)]^k\right) = \int_{-\infty}^{\infty} (x - E(X))^k \mathrm{d}F(x) \tag{1.77}$$

为 X 的 k 阶中心矩 (moment about centre).

矩是广泛应用的一类数字特征, 均值与方差分别就是一阶原点矩和二阶中心矩.

设分布函数 $F(x)$ 有中心矩 $\mu_2 = E(X - E(X))^2$, $\mu_3 = E(X - E(X))^3$, 则称

$$C_s = \mu_3 / \mu_2^{\frac{3}{2}} \tag{1.78}$$

为偏度系数 (coefficient of skewness).

偏度系数是一个无量纲的量, 它刻划分布函数的对称性. 当 $C_s > 0$ 时, $F(x)$ 所表示的概率分布偏向均值的右侧, 反之则偏向左侧.

设分布函数 $F(x)$ 有中心矩 $\mu_2 = E(X - E(X))^2$, $\mu_4 = E(X - E(X))^4$, 则称

$$C_k = \mu_4 / \mu_2^2 - 3 \tag{1.79}$$

为峰度系数 (kurtosis).

峰度系数是一个无量纲的量, 它刻划不同类型的分布的集中和分散程度.

设随机变量 X 有均值 μ 和方差 σ^2, 则称

$$X^* = (X - \mu)/\sigma \tag{1.80}$$

为标准化随机变量.

2. 协方差矩阵

设 $X=(X_1,X_2,\cdots,X_n)$, $Y=(Y_1,Y_2,\cdots,Y_m)$ 为两个随机变量，则称

$$\mathrm{Cov}(X,Y)=(\sigma_{ij})_{n\times m}$$

为 X 与 Y 的协方差阵 (covariance matrix), 其中 $\sigma_{ij}=\mathrm{Cov}(X_i,Y_j)$, $i=1,2,\cdots,n$, $j=1,2,\cdots,m$.

协方差阵具有如下性质：

(1) $\mathrm{Cov}(X,Y)=\mathrm{Cov}(Y,X)^T$.

(2) $\mathrm{Cov}(AX+b,Y)=A\mathrm{Cov}(X,Y)$, 其中 A 是矩阵, b 是向量.

(3) $\mathrm{Cov}(X+Y,Z)=\mathrm{Cov}(X,Z)+\mathrm{Cov}(Y,Z)$.

设 $X=(X_1,X_2,\cdots,X_n)$ 为随机变量，则称

$$\mathrm{Var}(X)=\mathrm{Cov}(X,X)=(\sigma_{ij})_{n\times n}$$

为 X 的方差阵 (variance matrix), 也称为方差 – 协方差矩阵 (variance-covariance matrix), 其中 $\sigma_{ij}=\mathrm{Cov}(X_i,X_j)$, $i,j=1,2,\cdots,n$.

方差矩阵具有如下性质：

(1) $\mathrm{Var}(X)$ 半正定，即 $\forall\ a\in R^n$, 有

$$a^T\mathrm{Var}(X)a\geq 0.$$

(2) $\forall\ a\in R^n$, 有

$$\mathrm{Var}(a^TX)=a^T\mathrm{Var}(X)a.$$

(3) $\forall\ A\in R^{k\times n}$, 有

$$\mathrm{Var}(AX)=A\mathrm{Var}(X)A^T.$$

(4) $\mathrm{Var}(X)=0$ 的充分必要条件是： $\exists\ a\in R^n$, $c\in R^1$, 使得

$$a^TX=c.$$

有了协方差矩阵的概念， n 维正态随机向量的概率密度函数的表示就变得容易了. n 维正态随机向量 $X=(X_1,X_2,\cdots,X_n)$ 的概率密度函数为

$$f(x)=\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}, \tag{1.81}$$

其中 $x=(x_1,x_2,\cdots,x_n)^T$, $\mu=(\mu_1,\mu_2,\cdots,\mu_n)^T=(E(X_1),E(X_2),\cdots,E(X_n))^T$, $\Sigma=\mathrm{Var}(X)$ 为 $n\times n$ 阶协方差矩阵且正定.

二维正态随机变量的密度函数 (1.60) 可以看成 n 维正态随机向量概率密度函数 (1.81) 的特例，其中协方差矩阵 Σ 为

$$\Sigma=\begin{bmatrix}\sigma_1^2 & \rho\sigma_1\sigma_2\\ \rho\sigma_1\sigma_2 & \sigma_2^2\end{bmatrix}.$$

对于 n 维正态随机向量 $(X_1,X_2,\cdots,X_n)$, 有如下的性质：

(1) $X_1,X_2,\cdots,X_n$ 相互独立与 $X_1,X_2,\cdots,X_n$ 两两互不相关等价；

(2) 设 $Y_1,Y_2,\cdots,Y_m$ 均是 $X_1,X_2,\cdots,X_n$ 的线性函数，则 $(Y_1,Y_2,\cdots,Y_m)$ 服从 m 维正态分布，该性质称为正态分布的线性变换不变性.

3. 相关矩阵

设 $X=(X_1,X_2,\cdots,X_n)$ 为随机变量，则称

$$\mathrm{Cor}(X)=(\rho_{ij})_{n\times n}$$

为 X 的相关矩阵 (correlation matrix), 其中 $\rho_{ij}=\mathrm{Cor}(X_i,X_j)$, $i,j=1,2,\cdots,n$.

相关矩阵具有如下性质：

(1) $\mathrm{Cor}(X)$ 为对角线元素均为 1 的半正定对称矩阵.

(2) 设 $\Sigma=(\sigma_{ij})_{n\times n}$ 为方差矩阵， $D=\mathrm{diag}\left(\sigma_{11}^{\frac{1}{2}},\sigma_{22}^{\frac{1}{2}},\cdots,\sigma_{nn}^{\frac{1}{2}}\right)$, 则

$$\mathrm{Cor}(X)=D^{-1}\Sigma D^{-1}.$$

1.4 极限定理

极限定理是概率论的基本定理之一，在概率论和数理统计的理论研究和实际应用中都具有重要的意义. 在极限定理中，最重要的是：大数定律和中心极限定理.

1.4.1 大数定律

大数定律是判断随机变量的算术平均值是否向常数收敛的定律, 是概率论和数理统计学的基本定律之一.

定义 1.17 设 $X_1, X_2, \cdots, X_k, \cdots$ 是随机变量序列且 $E(X_k)$ 存在 $(k = 1, 2, \cdots)$, 令 $Y_n = \frac{1}{n}\sum\limits_{k=1}^{n} X_k$, 若对于任意给定的 $\varepsilon > 0$, 有

$$\lim_{n\to\infty} P\{|Y_n - E(Y_n)| \geq \varepsilon\} = 0,$$

或

$$\lim_{n\to\infty} P\{|Y_n - E(Y_n)| < \varepsilon\} = 1,$$

则称随机变量序列 $\{X_k\}$ 服从大数定律.

关于大数定律有:

1. Bernoulli 大数定律

设 n_A 是 n 次独立重复试验中事件 A 发生的次数, p 是事件 A 在每次试验中发生的概率, 则对于任意的正数 $\varepsilon > 0$, 有

$$\lim_{n\to\infty} P\left\{\left|\frac{n_A}{n} - p\right| < \varepsilon\right\} = 1.$$

Bernoulli 大数定律揭示了 "频率稳定于概率" 说法的实质.

2. Chebyshev(切比雪夫) 大数定律

设随机变量 $X_1,\ X_2,\ \cdots,\ X_k,\ \cdots$ 相互独立, 且具有相同的期望与方差: $E(X_k) = \mu$, $\mathrm{Var}(X_k) = \sigma^2$ $(k = 1, 2, \cdots)$, 则对于任意的正数 $\varepsilon > 0$, 有

$$\lim_{n\to\infty} P\{|Y_n - \mu| < \varepsilon\} = 1.$$

3. Khintchin(辛钦) 大数定律

设随机变量 $X_1,\ X_2,\ \cdots,\ X_k,\ \cdots$ 相互独立, 服从相同的分布, 且其期望 $E(X_k) = \mu$ $(k = 1, 2, \cdots)$, 则对于任意的正数 $\varepsilon > 0$, 有

$$\lim_{n\to\infty} P\{|Y_n - \mu| < \varepsilon\} = 1.$$

若对随机变量序列 $X_1, X_2, \cdots, X_k, \cdots$, 存在常数 a, 使得对于任意的正数 $\varepsilon > 0$, 有

$$\lim_{n\to\infty} P\{|X_n - a| < \varepsilon\} = 1,$$

或

$$\lim_{n\to\infty} P\{|X_n - a| \geq \varepsilon\} = 0)$$

成立, 则称 X_n 依概率收敛于 a, 记作 $X_n \xrightarrow{P} a$. 故上面的 Chebyshev 大数定律与 Khintchin 大数定律有

$$Y_n = \frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{P} \mu.$$

对于大数定律, 有如下定理.

定理 1.2 设随机变量 X 具有期望 $E(X) = \mu$, 方差 $\mathrm{Var}(X) = \sigma^2$, 则对于任意 $\varepsilon > 0$, 有

$$P\{|X - \mu| \geq \varepsilon\} \leq \frac{\sigma^2}{\varepsilon^2}. \tag{1.82}$$

称定理 1.2 中的不等式 (1.82) 为 Chebyshev 不等式. 它是一个重要的理论工具, 应用很广. 例如, 在有关大数定律的证明中常用到它.

1.4.2 中心极限定理

中心极限定理是判断随机变量序列部分和的分布是否渐近于正态分布的一类定理. 在自然界及生产、科学实践中, 一些现象受到许多相互独立的随机因素的影响, 如果每个因素的影响都很小, 那么总的影响可以看作是服从正态分布. 中心极限定理正是从数学上论证了这一现象.

定义 1.18 凡是在一定条件下, 断定随机变量序列 $X_1, X_2, \cdots, X_k, \cdots$ 的部分和 $Y_n = \sum_{k=1}^{n} X_k$ 的极限分布为正态分布的定理, 均称为中心极限定理.

有两个最著名的中心极限定理.

1. 独立同分布的中心极限定理

设随机变量 $X_1, X_2, \cdots, X_k, \cdots$ 相互独立, 服从同一分布, 并且具有期望和方差: $E(X_k) = \mu, Var(X_k) = \sigma^2 > 0, k = 1, 2, \cdots$, 则随机变量

$$Y_n = \frac{\sum_{k=1}^{n} X_k - n\mu}{\sqrt{n}\sigma}$$

的分布函数 $F_n(x)$ 收敛到标准正态分布函数，即对于任意实数 x, 有

$$\lim_{n\to\infty} F_n(x) = \lim_{n\to\infty} P\{Y_n \le x\} = \Phi(x),$$

其中

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt.$$

从中心极限定理可知，当 n 足够大时， Y_n 近似服从标准正态分布 $N(0,1)$, 这在数理统计中有非常重要的应用.

2. De Moivre – Laplace (棣莫佛 – 拉普拉斯) 中心极限定理

设随机变量 $X_1, X_2, \cdots, X_k, \cdots$ 相互独立，并且服从参数为 p 的两点分布，则对于任意实数 x, 有

$$\lim_{n\to\infty} P\left\{ \frac{\sum_{i=1}^{n} X_i - np}{\sqrt{np(1-p)}} \le x \right\} = \Phi(x).$$

$\sum_{i=1}^{n} X_i$ 服从二项分布 $B(n,p)$. 从 De Moivre – Laplace 中心极限定理可知，当 n 足够大时， $B(n,p)$ 近似于正态分布. 它是独立同分布的中心极限定理的特殊情况.

1.5 数理统计的基本概念

前几节简单介绍了概率论的基本内容. 在概率论中，一般是在随机变量分布已知的情况下，着重讨论随机变量的性质. 但是对某个具体的随机变量来说，如何判断它服从某种分布？如果已知它服从某种类型的分布又该如何确定它的各个参数？对于这些问题概率论都没有涉及到，这些都是数理统计所要研究的内容，并且这些问题的研究都直接或间接建立在试验的基础上，数理统计学是利用概率论的理论对所要研究的随机现象进行多次的观察或试验，研究如何合理地获得数据，如何对所获得的数据进行整理、分析，如何对所关心的问题作出估计或判断的一门学科，其内容非常丰富.

下面给出数理统计的基本概念. 有关数理统计的各种方法和相应的 R 软件实现将在后续的各章中予以讨论.

1.5.1 总体、个体、简单随机样本

在数理统计中，称研究对象的全体为总体 (population), 通常用一个随机变量表示总体. 组成总体的每个基本单元叫个体 (individuals).

从总体 X 中随机抽取一部分个体 $X_1, X_2, \cdots, X_n$, 称 $X_1, X_2, \cdots, X_n$ 为取自 X 的容量为 n 的样本 (sample).

例如, 为了研究某厂生产的一批元件质量的好坏, 规定使用寿命低于 1 千小时的为次品, 则该批元件的全体就为总体, 每个元件就是个体. 实际上, 数理统计学中的总体是指与总体相联系的某个 (或某几个) 数量指标 X 取值的全体. 比如, 该批元件的使用寿命 X 的取值全体就是研究对象的总体. 显然 X 是随机变量, 这时, 就称 X 为总体.

为了判断该批元件的次品率, 最精确的办法是取出全部元件, 对作元件的寿命试验. 然而, 寿命试验具有破坏性, 即使某些试验是非破坏性的, 试验也要要花费人力、物力、时间, 因此只能从总体中抽取一部分, 比如说 n 个个体进行试验. 试验结果可得组数值集合 $\{x_1, x_2, \cdots, x_n\}$, 其中每个 x_i 是第 i 次抽样观察的结果. 由于要根据这些观察结果来对总体进行推断, 所以对每次抽样就需要有一定的要求, 要求每次抽取必须是随机的、独立的, 这样才能较好地反映总体情况. 所谓随机的是指每个个体被抽到的机会是均等的, 这样抽到的个体才具有代表性. 若 $X_1, X_2, \cdots, X_n$ 相互独立, 且每个 X_i 与 X 同分布, 则称 $X_1, X_2, \cdots, X_n$ 为简单随机样本 (simple random sample), 简称样本. 通常把 n 称为样本容量 (sample size).

值得注意的是，样本具有两重性，即当在一次具体地抽样后它是一组确定的数值. 但在一般叙述中样本也是一组随机变量，因为抽样是随机的. 今后，用 $X_1, X_2, \cdots, X_n$ 表示随机样本，它们取到的值记为 $x_1, x_2, \cdots, x_n$, 称为样本观测值 (sample value).

样本作为随机变量, 有一定的概率分布, 这个概率分布称为样本分布. 显然, 样本分布取决于总体的性质和样本的性质.

总体 X 具有分布函数 $F(x)$, 则 $(X_1, X_2, \cdots, X_n)$ 的联合概率分布函数为

$$F(X_1, X_2, \cdots, X_n) = \prod_{i=1}^{n} F(x_i).$$

若 X 具有概率密度函数 $f(x)$, 则 $(X_1, X_2, \cdots, X_n)$ 的联合概率密度为

$$f(X_1, X_2, \cdots, X_n) = \prod_{i=1}^{n} f(x_i).$$

例 1.5 要估计一物体的重量 a, 用天平将物体重复测量 n 次, 结果记为 X_1, X_2, $\cdots$, X_n, 求样本 $(X_1, X_2, \cdots, X_n)$ 的分布.

解: 假定各次测量是相互独立, 即 $X_1, X_2, \cdots, X_n$ 为一简单随机样本. 再假定测量的随机误差服从正态分布, 天平没有系统误差, 因此随机误差的均值为 0, 于是总体的概率分布可假定为 $N(a, \sigma^2)$, 其中 a 为物体之重量, σ^2 反映天平的精度. 故 $(X_1, X_2, \cdots, X_n)$ 的概率密度为

$$\begin{aligned} f(x_1, x_2, \cdots, x_n; a, \sigma^2) &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2\sigma^2}(x_i - a)^2\right\} \\ &= (\sqrt{2\pi}\sigma)^{-n} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - a)^2\right\} \end{aligned}$$

例 1.6 设某电子元件的寿命 X 从指数分布

$$f(x, \lambda) = \begin{cases} \lambda\, \mathrm{e}^{-\lambda x}, & x \geq 0, \\ 0, & x < 0, \end{cases}$$

今从一批产品中独立地抽取 n 件进行寿命试验, 测得寿命数据为 $X_1, X_2, \cdots, X_n$, 求样本 $(X_1, X_2, \cdots, X_n)$ 的概率分布.

解: 依题意有为 $X_1, X_2, \cdots, X_n$ 是独立同分布的, 且 $X_i \sim f(x, \lambda)$, 故所求概率密度为

$$\begin{aligned} f(x_1, x_2, \cdots, x_n; \lambda) &= \prod_{i=1}^{n} f(x_i, \lambda) \\ &= \begin{cases} \lambda^n \exp\left\{-\lambda \sum\limits_{i=1}^{n} x_i\right\}, & x_1, x_2, \cdots, x_n \geq 0, \\ 0, & \text{其它}, \end{cases} \end{aligned}$$

1.5.2 参数空间与分布族

在例 1.5 中总体分布为 $N(a, \sigma^2)$, 其中 a 与 σ^2 是确定分布的常数. 例 1.6 中总体分布为指数分布 $f(x, \lambda)$, λ 也是确定分布的常数. 在数理统计中, 称出现在

样本分布中的常数为参数 (parameter), 因此， a, σ^2 和 λ 都是参数. 这些参数是关于总体的重要的数量指标, 然而, 这些参数往往是未知的, 称为未知参数. 在例 1.5 中， a 是未知参数, 而 σ^2 是否为未知参数要看人们对天平精度的了解程度. 若对天平精度足够了解可以给出 σ^2 的值, 则 σ^2 就是已知参数; 若对天平的精度不够了解, 无法给出 σ^2 的值, 甚至于抽样的目的就是要估计推断这个精度, 那么， σ^2 就是未知参数, 这时, 称 (a, σ^2) 为参数向量. 参数所有可能的取值构成的集合称为参数空间. 如例 1.5 中 (a, σ^2) 都是参数, 则参数空间为 $\Theta = \{(a, \sigma^2) : \ a > 0, \sigma^2 > 0\}$. 例 1.6 的参数空间为 $\Theta = \{\lambda : \ \lambda > 0\}$.

当样本分布含有未知参数时, 不同的参数值对应于不同的分布. 因此, 可能的样本不止一个, 而是一族, 则称为样本分布族. 同样, 存在未知参数时, 总体分布也是一族, 构成总体分布族. 例 1.5 中, 若 a 和 σ^2 都是未知参数, 则总体分布族为 $\{N(a, \sigma^2) : \ a > 0, \sigma^2 > 0\}$, 样本分布族为 $\{f(x_1, x_2, \cdots, x_n; a, \sigma^2) : \ a > 0, \sigma^2 > 0\}$. 在例 1.6 中, 若 λ 是未知的, 则总体分布族为 $\{f(x, \lambda) : \lambda > 0\}$, 样本分布族为 $\{f(x_1, x_2, \cdots, x_n, \lambda) : \ \lambda > 0\}$.

1.5.3 统计量和抽样分布

数理统计的任务是采集和处理带有随机影响的数据, 或者说收集样本并对之进行加工, 以此对所研究的问题作出一定的结论, 这一过程称为为统计推断. 在统计推断中, 对样本进行加工整理, 实际上就是根据样本计算出一些量, 使得这些量能够将所研究问题的信息集中起来. 这种根据样本计算出的量就是下面将要定义的统计量, 因此, 统计量是样本的某种函数.

定义 1.19 设 $X_1, X_2, \cdots, X_n$ 是总体 X 的一个简单随机样本， $T(X_1, X_2, \cdots, X_n)$ 为一个 n 元连续函数, 且 T 中不含任何关于总体的未知参数, 则称 $T(X_1, X_2, \cdots, X_n)$ 为一个统计量 *(statistic)*. 称统计量的分布为抽样分布 *(sampling distribution)*.

1. 常用的统计量

(1) 样本均值

设 $X_1, X_2, \cdots, X_n$ 是总体 X 的一个简单随机样本, 称

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} X_i \tag{1.83}$$

为样本均值 (sample mean). 通常用样本均值来估计总体分布的均值和对有关总体分布均值的假设作检验.

(2) 样本方差

设 $X_1, X_2, \cdots, X_n$ 是总体 X 的一个简单随机样本, $\overline{X}$ 为样本均值, 称

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2 \tag{1.84}$$

为样本方差 (sample variance). 通常用样本方差来估计总体分布的方差和对有关总体分布均值或方差的假设作检验.

(3) k 阶样本原点矩

设 $X_1, X_2, \cdots, X_n$ 是总体 X 的一个简单随机样本, 称

$$A_k = \frac{1}{n}\sum_{i=1}^{n} X_i^k \tag{1.85}$$

样本的 k 阶原点矩, 通常用样本的 k 阶原点矩来估计总体分布的 k 阶原点矩.

(4) k 阶样本中心矩

设 $X_1, X_2, \cdots, X_n$ 是总体 X 的一个简单随机样本, $\overline{X}$ 为样本均值, 称

$$M_k = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^k \tag{1.86}$$

样本的 k 阶中心矩, 通常用样本的 k 阶中心矩来估计总体分布的 k 阶中心矩.

(5) 顺序统计量

设 $X_1, X_2, \cdots, X_n$ 是抽自总体 X 的样本, $x_1, x_2, \cdots, x_n$ 为样本观测值, 将 $x_1, x_2, \cdots, x_n$ 按照从小到大的顺序排列为

$$x_{(1)} \le x_{(2)} \le \cdots \le x_{(n)},$$

当样本 $X_1, X_2, \cdots, X_n$ 取值为 $x_1, x_2, \cdots, x_n$ 时, 定义 $X_{(k)}$ 取值为 $x_{(k)}$ $(k = 1, 2, \cdots, n)$, 称 $X_{(1)}, X_{(2)}, \cdots, X_{(n)}$ 为 $X_1, X_2, \cdots, X_n$ 的顺序统计量 (order statistic) .

显然, $X_{(1)} = \min\limits_{1\le i\le n}\{X_i\}$ 是样本观测中取值最小的一个, 称为最小顺序统计量 (smallest order statistic). $X_{(n)} = \max\limits_{1\le i\le n}\{X_i\}$ 是样本观测中取值最大的一个, 称为最大顺序统计量 (largest order statistic). 称 $X_{(r)}$ 为第 r 个顺序统计量.

(6) 经验分布函数

设 $X_1, X_2, \cdots, X_n$ 是抽自总体 X 的样本， $X \sim F(x)$, 则称

$$F_n(x) = \frac{1}{n}K(x), \qquad -\infty < x < \infty \tag{1.87}$$

为经验分布函数 (expirical distribution), 其中 $K(x)$ 表示 $X_1, X_2, \cdots, X_n$ 中不大于 x 的个数.

经验分布函数也可以表示成

$$F_n(x) = \begin{cases} 0, & x < X_{(1)}, \\ \frac{k}{n}, & X_{(k)} \le x < X_{(k+1)}, \\ 1, & x \ge X_{(n)} \end{cases} \tag{1.88}$$

$F_n(x)$ 是一个跳跃函数, 其跳跃点是样本观测值. 在每个跳跃点处跳跃度均为 $1/n$.

图 1.7 所示的是 $n = 10$, 抽自总体 $N(0,1)$ 的经验分布函数和 $N(0,1)$ 的总体分布函数图.

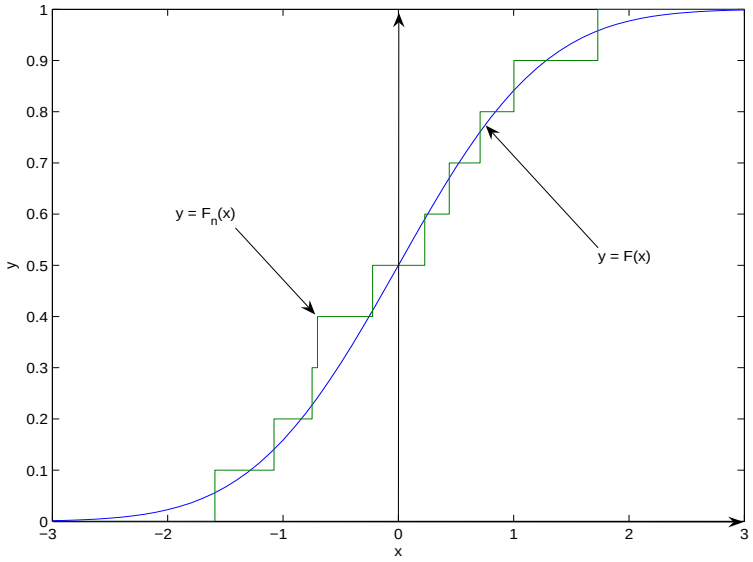


图 1.7: 经验分布和总体分布

对于经验分布函数有以下结果（ Glivenko (格里文科) 1933 年证明）

$$P\left\{\lim_{n\to\infty} \sup_{-\infty<x<\infty} |F_n(x) - F(x)| = 0\right\} = 1. \tag{1.89}$$

这个结果表明对任意的实数 x 当 n 充分大时，经验分布函数与总体分布函数的差异很小，因此 n 充分大时实际上可用 $F_n(x)$ 近似代替 $F(x)$.

2. 常用的分布和分位数

(1) χ^2 分布

设 $X_1, X_2, \cdots, X_n$ 是来自总体 $N(0,1)$ 的一个简单样本，则称统计量

$$Y = X_1^2 + X_2^2 + \cdots + X_n^2 \tag{1.90}$$

为服从自由度为 n 的 χ^2 分布 (chi-square distribution), 记为 $Y \sim \chi^2(n)$. 图 1.8

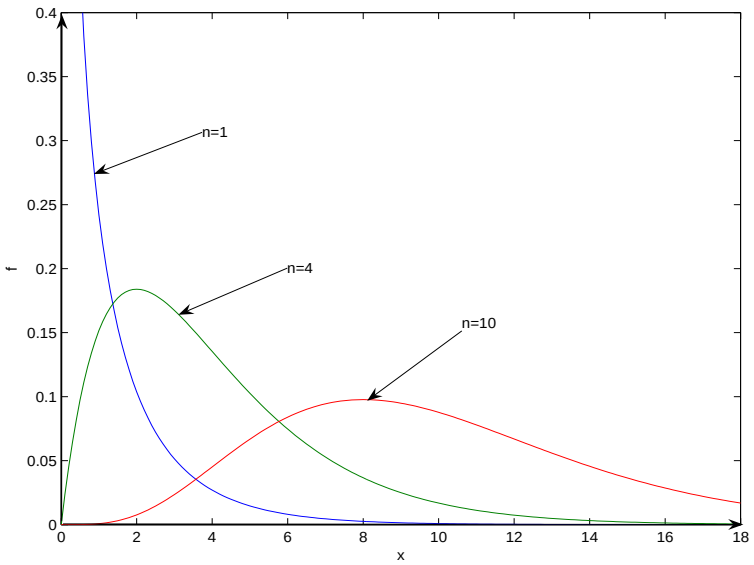


图 1.8: χ^2 分布密度函数曲线

给出了 $n = 1$ 、 $n = 4$ 和 $n = 10$ 的 χ^2 分布密度函数曲线.

从图 1.8 可以看出， χ^2 分布密度函数曲线的峰值偏左，其偏度系数 C_s 为正. 当 n 越小时，密度曲线越陡峭，其峰度系数 C_k 就越大；当 n 越大时，曲线越平坦，其峰度系数 C_k 就越小.

若对于给定的 α, $0 < \alpha < 1$, 存在 $\chi^2_\alpha(n)$ 使

$$P\{\chi^2 > \chi^2_\alpha(n)\} = \alpha,$$

则称点 $\chi^2_\alpha(n)$ 为 χ^2 分布的上 α 分位点. 图 1.9 所示的是 $n = 5$, $\alpha = 0.1$ 的 χ^2 分布的上 α 分位点 $\chi^2_\alpha(n)$.

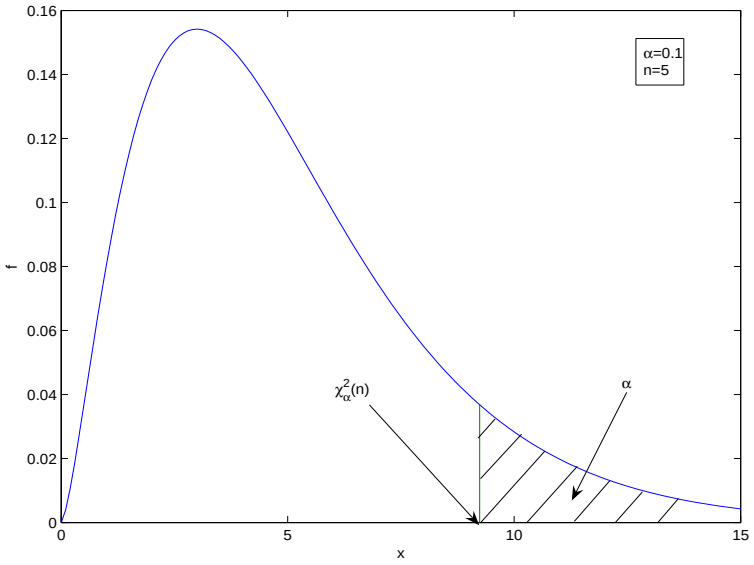


图 1.9: χ^2 分布的上 α 分位点

χ^2 分布具有如下性质.

(i) 可加性. 设 $Y_1 \sim \chi^2(m)$, $Y_2 \sim \chi^2(n)$, 且两者相互独立，则 $Y_1 + Y_2 \sim \chi^2(m+n)$.

(ii) 期望值与方差. 若 $Y \sim \chi^2(n)$, 则 $\mathrm{E}(Y) = n$, $\mathrm{Var}(Y) = 2n$.

(2) t 分布

设 $X \sim N(0,1)$, $Y \sim \chi^2(n)$, 且 X, Y 相互独立，则称随机变量

$$T = \frac{X}{\sqrt{Y/n}} \tag{1.91}$$

为服从自由度为 n 的 t 分布 (t-distribution), 记为 $T \sim t(n)$.

图 1.10 给出了 $n = 1$ 、 $n = 2$ 、 $n = 5$ 和 $n = \infty$ 的 t 分布密度函数曲线.

从图 1.10 可以看出， t 分布是对称分布，其偏度系数 C_s 为 0. n 越小，其峰度系数 C_k 越大， n 越大，其峰度系数 C_k 越小.

若对于给定的 α, $0 < \alpha < 1$, 称满足

$$P\{T > t_\alpha(n)\} = \alpha,$$

的点 $t_\alpha(n)$ 为 t 分布的上 α 分位点. 图 1.11 所示的是 $n = 5$, $\alpha = 0.1$ 的 t 分布的上 α 分位点 $t_\alpha(n)$.

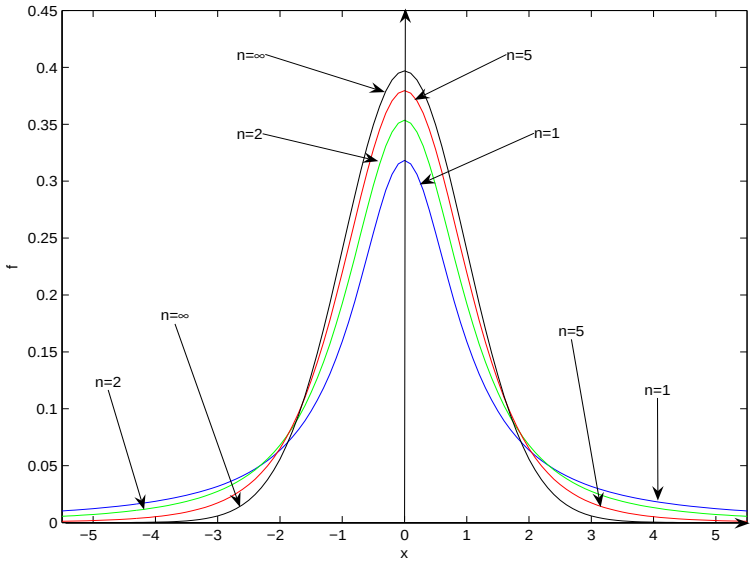


图 1.10: t 分布密度函数曲线

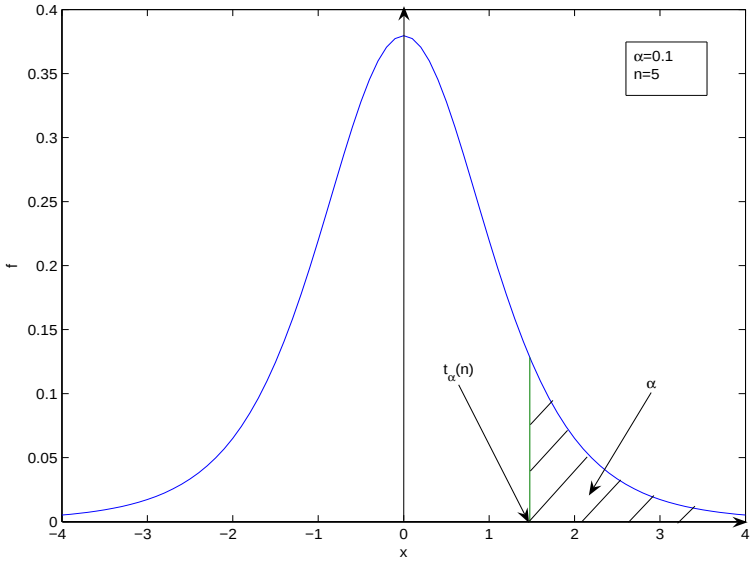


图 1.11: t 分布的上 α 分位点

由于 t 分布的概率密度函数 $f(t)$ 是偶函数，即 $f(t)=f(-t)$, 关于 $t=0$ 对称，因此对一切 n, 有 $E(T)=0$. 并且

$$\int_{-t_n(\alpha)}^{\infty} f(t)\mathrm{d}t = 1-\alpha,$$

所以 $t_{1-\alpha}(n) = -t_\alpha(n)$.

(3) F 分布

设 $X \sim \chi^2(n)$, $Y \sim \chi^2(m)$, 且 X 和 Y 相互独立，则称随机变量

$$F = \frac{X/n}{Y/m} \tag{1.92}$$

为服从自由度为 (n, m) 的 F 分布 (F-distribution), 称 n 为第一自由度， m 为第二自由度，记为 $F \sim F(n, m)$.

图 1.12 所示的是 $n = 5$, $m = 20$, $n = 7$, $m = 20$, $n = 20$, $m = 20$, $n = 20$, $m = 2$ 和 $n = 20$, $m = 7$ 的 F 分布密度函数曲线.

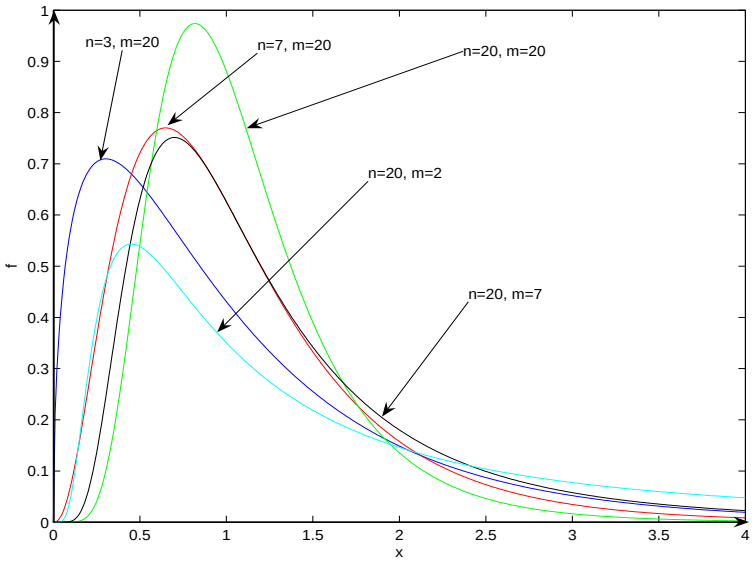


图 1.12: F 分布密度函数曲线

若对于给定的 α, $0 < \alpha < 1$, 称满足

$$P\{F > F_\alpha(n, m)\} = \alpha,$$

的点 $F_\alpha(n, m)$ 为 F 分布的上 α 分位点.

图 1.13 所示的是 $n = 5$, $m = 8$, $\alpha = 0.1F$ 分布的上 α 分位点 $F_\alpha(n, m)$.

F 分布具有如下性质：

(i) $X \sim F(n, m)$, 则 $1/X \sim F(m, n)$;

(ii) $F_{1-\alpha}(n, m) = \frac{1}{F_\alpha(m,n)}$.

(iii) 设 $X \sim t(n)$, 则 $X^2 \sim F(1, n)$.

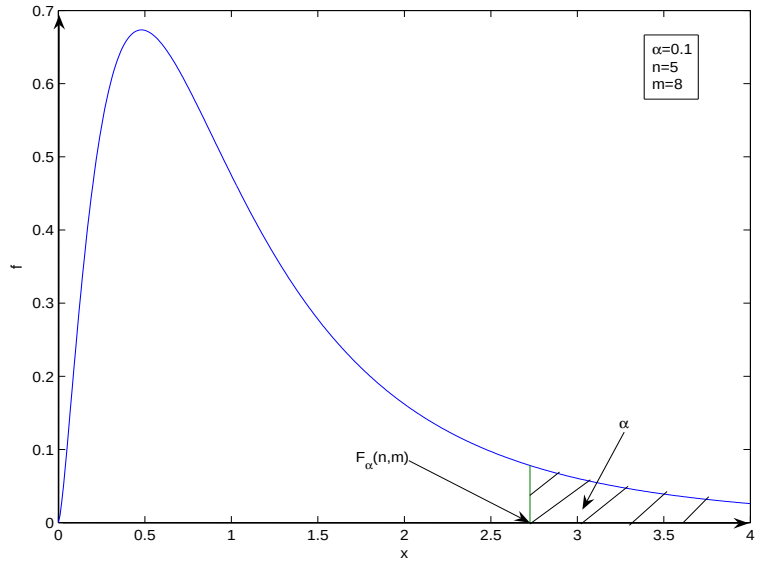


图 1.13: F 分布的上 α 分位点

1.5.4 正态总体样本均值与样本方差的分布

设 $X_1, X_2, \cdots, X_n$ 是来自于正态总体 $N(\mu, \sigma^2)$ 的样本，$\overline{X}$, S^2 分别为样本均值和样本方差，则有

$$\frac{\overline{X}-\mu}{\sigma/\sqrt{n}} \sim N(0,1), \tag{1.93}$$

$$\frac{(n-1)S^2}{\sigma^2} \sim \chi^2(n-1), \tag{1.94}$$

$$\frac{\overline{X}-\mu}{S/\sqrt{n}} \sim t(n-1), \tag{1.95}$$

且 $\overline{X}$ 与 S^2 相互独立.

设 $X_1, X_2, \cdots, X_{n_1}$ 与 $Y_1, Y_2, \cdots, Y_{n_2}$ 分别是来自于正态总体 $N(\mu_1, \sigma_1^2)$, $N(\mu_2, \sigma_2^2)$ 的样本，且这两样本相互独立，则有

$$\overline{X}-\overline{Y} \sim N\left(\mu_1-\mu_2, \frac{\sigma_1^2}{n_1}+\frac{\sigma_2^2}{n_2}\right), \tag{1.96}$$

或

$$\frac{\left(\overline{X}-\overline{Y}\right)-(\mu_1-\mu_2)}{\sqrt{\frac{\sigma_1^2}{n_1}+\frac{\sigma_2^2}{n_2}}} \sim N(0,1). \tag{1.97}$$

若 $\sigma_1^2 = \sigma_2^2 = \sigma^2$, 且 σ^2 未知，则

$$\frac{(\overline{X} - \overline{Y}) - (\mu_1 - \mu_2)}{S_w\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \sim t(n_1 + n_2 - 2), \tag{1.98}$$

其中

$$S_w^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}, \tag{1.99}$$

$\overline{X}, \overline{Y}$ 分别是两样本的均值， S_1^2, S_2^2 分别是两样本的方差.

习题一

1.1 设有 m 个人，每个人都以相同的概率 $\frac{1}{N}$ 被分入 N 个室 $(N \geq m)$ 中任意一个室中去住，且每室中人数不限，并允许有空室，求：

(1) 某指定的 m 个室中每室各分入 1 人的概率；

(2) 恰有 m 个室，其中每室各分入 1 人的概率；

(3) 若 $N = 10, m = 6$, 求恰有两人分入同一室 (即恰有一室，其中恰分入了两个人) 的概率.

1.2 甲、乙两轮驶向一个不能同时停泊两轮的码头，它们在一昼夜内到达的时刻是等可能的. 设甲轮的停泊时间是 1 小时，乙轮的停泊时间是 2 小时，求二轮都不需等待码头空出的概率.

1.3 一批产品共有 20 件，其中有 5 件次品，其余为正品. 现依次进行不放回抽取三次，求：

(1) 第三次才取到次品的概率；

(2) 在第一、第二次取到正品的条件下，第三次取到次品的概率；

(3) 第三次取到次品的概率.

1.4 有朋自远方来，他乘火车、轮船、汽车、飞机来的概率分别为 0.3, 0.2, 0.1, 0.4. 如果他乘火车、轮船、汽车、飞机来的话，迟到的概率分别为 1/4, 1/3, 1/12, 而乘飞机则不会迟到. 现朋友迟到了，问他是乘火车来的概率是多少？

1.5 设每人血清中含有肝炎病毒的概率为 0.004, 随机混合 100 人的血清. 求此血清中含有肝炎病毒的概率.

1.6 甲、乙、丙三门高射炮彼此独立地向同一架飞机射击，设甲、乙、丙炮射中飞机的概率分别为 $0.7, 0.8, 0.9$.

(1) 求飞机被射中的概率；

(2) 又设若只有一门炮射中飞机坠毁的概率为 0.7，若有两门炮射中飞机坠毁的概率为 0.9，若三门炮都射中，飞机必坠毁，求飞机坠毁的概率.

1.7 一个靶子是半径为 *2* 米的圆盘，设击中靶上任一同心圆盘上的点的概率与该圆盘的面积成正比，并设射击都能中靶，以 X 表示弹着点与圆心的距离，试求随机变量 X 的分布函数.

1.8 某单位招聘 *2500* 人，按考试成绩从高分到低分依次录用，共有 *10000* 人报名，假设报名者的成绩 $X \sim N(\mu, \sigma^2)$，已知 *90* 分以上有 *359* 人， *60* 分以下有 *1151* 人，问被录用者中最低分为多少？

1.9 现有 *90* 台同类型的设备，各台设备的工作是相互独立的，发生故障的概率是 0.01，且一台设备的故障能由一人处理，配备维修工人的方法有两种，一种是 *3* 人分开维护，每人负责 *30* 台，另一种是由 *3* 人共同维护 *90* 台，试比较两种方法在设备发生故障时不能及时维修的概率的大小.

1.10 设二维随机向量 (X, Y) 的分布函数为：

$$F(x,y) = \begin{cases} 1 - 2^{-x} - 2^{-y} + 2^{-x-y}, & x \geq 0, y \geq 0, \\ 0, & \text{其它}, \end{cases}$$

求 $P\{1 < X \leq 2, 3 < Y \leq 5\}$.

1.11 一个袋中装有 *5* 只球，其中 *4* 只红球， *1* 只白球. 每次从中随机地抽取一只，取后不放回，连续抽取两次，令

$$X = \begin{cases} 1, & \text{若第一次抽到红球}, \\ 0, & \text{若第一次抽到白球}, \end{cases}, Y = \begin{cases} 1, & \text{若第二次抽到红球}, \\ 0, & \text{若第二次抽到白球}, \end{cases}$$

试求：

(1) (X, Y) 的联合分布律；

(2)$P\{X \geq Y\}$.

1.12 设二维随机变量 (X,Y) 的联合概率密度函数为：

$$f(x,y)=\begin{cases} Ae^{-(2x+y)}, & x>0,y>0,\\ 0, & \text{其它}. \end{cases}$$

求：

(1) 常数 A;

(2) $P\{-1<X<1,-1<Y<1\}$;

(3) $P\{X+Y\le 1\}$;

(4) (X,Y) 的联合分布函数 $F(x,y)$.

1.13 飞机场送客汽车载有 *20* 位乘客，离开机场后共有 *10* 个车站可以下车，若某个车站无人下车该车站则不停车. 设乘客在每个车站下车的可能性相等且他们的行动相互独立，以 X 表示停车的次数，求 $E(X)$.

1.14 某保险公司制定赔偿方案：如果在一年内一个顾客的投保事件 A 发生，该公司就赔偿该顾客 a 元，若已知一年内事件 A 发生的概率为 p, 为使公司收益的期望值等于 a 的 5%, 该公司应该要求顾客交纳多少元的保险费？

1.15 设在总体 $N(\mu,\sigma^2)$ 中抽取一容量为 n 的样本，这里 μ,σ^2 均为未知. 当 $n=16$ 时，求 $P\{S^2/\sigma^2\le 2.04\}$.

1.16 设 $X_1,X_2,\cdots,X_n$ 和 $Y_1,Y_2,\cdots,Y_n$ 是分别来自于正态总体 $X\sim N(\mu_1,\sigma^2)$ 和 $Y\sim N(\mu_2,\sigma^2)$, 且相互独立，则以下统计量服从什么分布？

(1) $\dfrac{(n-1)(S_1^2+S_2^2)}{\sigma^2}$; *(2)* $\dfrac{n\left[(\overline{X}-\overline{Y})-(\mu_1-\mu_2)\right]^2}{S_1^2+S_2^2}$.

第二章　R 软件的使用

在第一章，介绍了概率统计的基本概念，从本章开始介绍如何用 R 软件求解统计问题．在介绍各种方法之前，先对 R 软件作一个基本的介绍．

2.1　R 软件简介

R 是一个开放的统计编程环境，是一种语言，是 S 语言的一种实现． S 语言是由 AT&T Bell 实验室的 Rick Becker, John Chambers 和 Allan Wilks 开发的一种用来进行数据探索、统计分析、作图的解释型语言．最初 S 语言的实现版本主要是 S-Plus. S-Plus 是一个商业软件，它基于 S 语言，并由 MathSoft 公司的统计科学部进一步完善． R 是一种软件，是一套完整的数据处理、计算和制图软件系统．其功能包括：数据存储和处理系统，数组运算工具，完整连贯的统计分析工具，优秀的统计制图功能．简便而强大的编程语言：可操纵数据的输入和输出，可实现分支、循环，用户可自定义功能．

Auckland (奥克兰) 大学的 Robert Gentleman 和 Ross Ihaka 及其他志愿人员开发了一个 R 系统，目前由 R 核心开发小组维护，他们完全自愿、工作努力负责，并将全球优秀的统计应用软件打包提供给我们．我们可以通过 R 软件的网站 (http://www.r-project.org) 了解有关 R 的最新信息和使用说明，得到最新版本的 R 软件和基于 R 的应用统计软件包．

R 是完全免费的，而 S-Plus 尽管是非常优秀的统计分析软件，但是需要付费的． R 可以在 UNIX 、 Windows 和 Macintosh 的操作系统上运行，它嵌入了一个非常实用的帮助系统，并具有很强的作图能力． R 的使用与 S-Plus 有很多类似之处，两个软件有一定的兼容性． S-Plus 的使用手册，只要经过不多的修改就能成为 R 的使用手册．

与其说 R 软件是一种统计软件，还不如说 R 是一种数学计算环境．因为 R 提供了有弹性的、互动的环境来分析、可视及展示数据；它提供了若干统计程序包，以及一些集成的统计工具和各种数学计算、统计计算的函数，用户只需根据统计模型，指定相应的数据库及相关的参数，便可灵活机动的进行数据分析等工作，甚至创造出符合需要的新的统计计算方法．使用 R 软件可以简化你的数据分析过程，从数据的存取，到计算结果的分享， R 软件提供了更加方便的计算

工具，帮助你更好地决策. 通过 R 软件的许多内嵌统计函数，用户可以很容易学习和掌握 R 软件的语法，也可以编制自己的函数来扩展现有的 R 语言，完成你的科研工作.

2.1.1 R 软件的下载与安装

R 软件是全免费的，在网站：

`http://cran.r-project.org/bin/windows/base/`

可下载到 R 软件的 Windows 版，当前的版本是 R-2.3.1 版 (2006 年 6 月 1 日发布), 大约是 27 兆，点击 `R-2.3.1-win32` 下载，或者选择距离你最近的镜像 (mirror near you) 下载. 注意，在 R-2.2.0 版本以前是点击 `rwXXXX.exe` 下载，其中 `XXXX` 是版的序号，如本书使用的版本是 R-2.1.1, 则点击 `rw2011.exe` 下载.

R 软件可以在 Windows 95, 98, ME, NT4, 2000, XP 和 2003 上运行，最好选择 Windows 98 以上的操作系统.

R 软件安装非常容易, 运行你刚才下载的程序, 如 `R-2.3.1-win32.exe` (R for Windows Setup), 按照 Windows 的提示安装即可. 当你开始安装后, 选择安装提示的语言 (中文或英文), 接受安装协议，选择安装目录 (缺省值 `C:\Program Files\R\R-2.3.1`), 并选择安装组件. 在安装组件中，最好将 `PDF Reference Manual` 项也选上，这样在 R 软件的帮助文件中有较为详细的 PDF 格式的软件说明.

注意, 在 R-2.2.0 以前的版本, 在安装组件中, 一定要选择东亚语言版 (Version for East Asian languages), 否则在中文 Windows 操作系统下的 R 窗口会出现乱码.

按照 Windows 的各种提示操作，你稍候片刻， R 软件就安装成功了.

安装完成后, 程序会创建 R 程序组并在桌面上创建 R 主程序的快捷方式 (也可以在安装过程中选择不要创建). 通过快捷方式运行 R, 便可调出 R 的主窗口，如图 2.1 所示.

R 软件的界面与 Windows 的其他编程软件相类似，是由一些菜单和快捷按钮组成. 快捷按钮下面的窗口便是命令输入窗口，它也是部分运算结果的输出窗口，有些运算结果 (如图形) 则会在新建的窗口中输出.

主窗口上方的一些文字 (如果是中文操作系统，则显示中文) 是刚运行 R 时出现的一些说明和指引. 文字下的 > 符号便是 R 的命令提示符 (矩形光标), 在其后可输出命令. R 一般采用交互式工作方式，在命令提示符后输入命令，回车

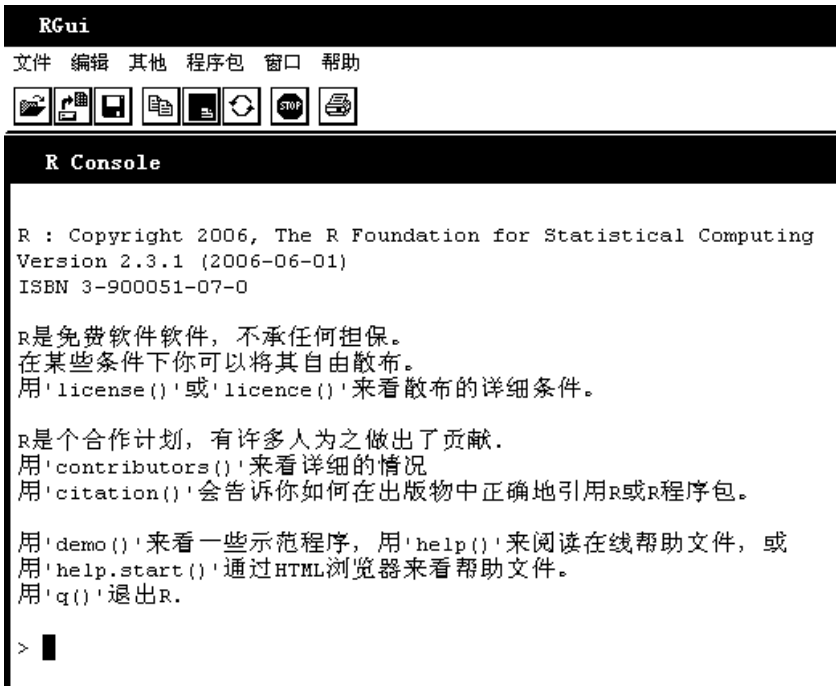


图 2.1: R 软件主窗口

后便会输出计算结果. 当然也可将所有的命令建立成一个文件, 运行这个文件的全部或部分来执行相应的命令, 从而得到相应的结果. 这种计算方式更加简便, 具体计算过程, 将在后面进行讨论.

2.1.2 初识 R

用三个简单的例子, 认识一下 R 软件.

例 2.1 *某学校在体检时测得 12 名女中学生体重 X_1(千克) 和胸围 X_2(厘米) 资料如表 2.1 所示. 试计算体重与胸围的均值与标准差.*

解: 直接在主窗口输入命令,

```
> #  输入体重数据
> X1 <- c(35, 40, 40, 42, 37, 45, 43, 37, 44, 42, 41, 39)
> mean(X1)    #  计算体重的均值
[1] 40.41667
> sd(X1)      #  计算体重的标准差
[1] 3.028901
```

表 2.1: 学生体检资料

学生编号	体重 X_1	胸围 X_2	学生编号	体重 X_1	胸围 X_2
1	35	60	7	43	78
2	40	74	8	37	66
3	40	64	9	44	70
4	42	71	10	42	65
5	37	72	11	41	73
6	45	68	12	39	75

```
> #  输入胸围数据
> X2 <- c(60, 74, 64, 71, 72, 68, 78, 66, 70, 65, 73, 75)
> mean(X2)   #  计算胸围的均值
[1] 69.66667
> sd(X2)     #  计算胸围的标准差
[1] 5.210712
```

从上述计算过程来看， R 软件计算这些统计量非常简单. 我们来逐句作一下解释.

“#” 号是说明语句字符， # 后面的语句是说明语句，大家学习运用说明语句，来说明程序要作的工作，增加程序的可读性.

<- 表示赋值， c( ) 表示数组， X1<-c( ) 即表示将一组数据赋给变量 X1.

mean( ) 是求均值函数， mean(X1) 表示计算数组 X1 的均值.

[1] 40.41667 是计算结果，这里的 [1] 表示第 1 行， 40.41667 是计算出的均值，即这 12 名女生的平均体重是 40.42 千克.

sd( ) 是求标准差函数， sd(X1) 表示计算数组 X1 的标准差.

上述过程中的 > 号，均是计算机提示符.

当你退出 R 系统时，计算机会询问你是否保存工作空间映象，你可选择保存 (是 (Y)) 或不保存 (否 (N)).

如果想将上述命令保存在文件中, 希望以后调用, 可以先将所有的命令放在一个文件中. 用鼠标点击 “文件” 窗口下的 “建立新的程序脚本”, 则屏幕会弹出一个 R 编辑 (R Editor) 窗口, 在窗口中输入相应的命令即可. 然后将文件保存起来, 如文件名: exam0201.R.

例 2.2 绘出例 *2.1* 中 *12* 名学生体重与胸围的散点图和体重的直方图.

解: 在主窗口下输入

```
> X1<-c(35, 40, 40, 42, 37, 45, 43, 37, 44, 42, 41, 39)
> X2 <- c(60, 74, 64, 71, 72, 68, 78, 66, 70, 65, 73, 75)
> plot(X1, X2)
```

则 R 软件会打开一个新的窗口, 新窗口绘出体重与胸围的散点图, 如图 2.2 所

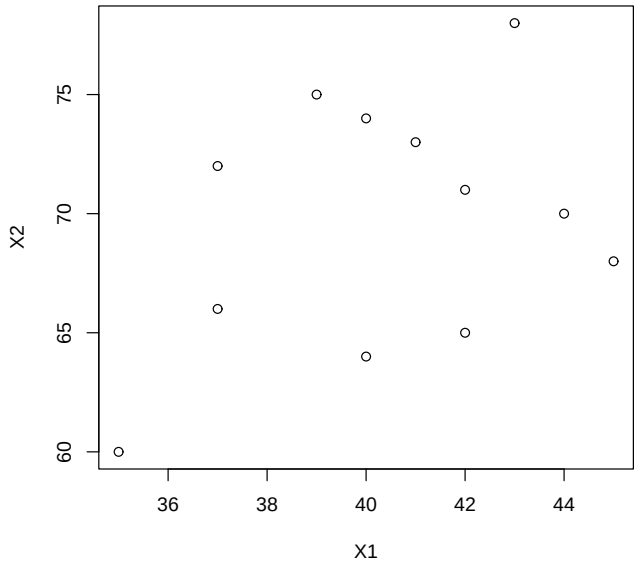


图 2.2: 12 名学生体重与胸围的散点图

示.

再键入

```
> hist(X1)
```

则屏幕会弹出另一个新窗口, 新窗口绘有体重的直方图, 如图 2.3 所示.

例 2.3 设有文本文件 exam0203.txt, 其内容与格式如下:

```
Name    Sex  Age Height  Weight
Alice     F   13   56.5    84.0
```

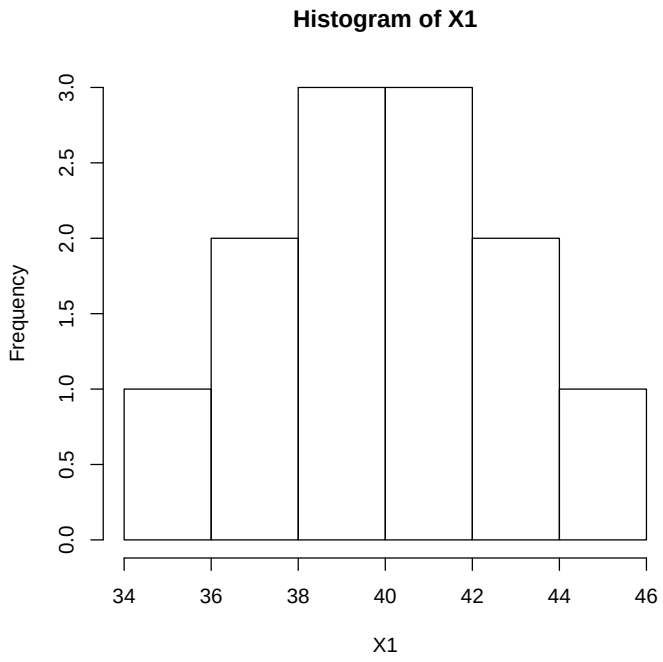


图 2.3: 12 名学生体重的直方图

Becka	F	13	65.3	98.0
Gail	F	14	64.3	90.0
Karen	F	12	56.3	77.0
Kathy	F	12	59.8	84.5
Mary	F	15	66.5	112.0
Sandy	F	11	51.3	50.5
Sharon	F	15	62.5	112.5
Tammy	F	14	62.8	102.5
Alfred	M	14	69.0	112.5
Duke	M	14	63.5	102.5
Guido	M	15	67.0	133.0
James	M	12	57.3	83.0
Jeffrey	M	13	62.5	84.0
John	M	12	59.0	99.5
Philip	M	16	72.0	150.0
Robert	M	12	64.8	128.0

```
Thomas    M   11   57.5    85.0
William   M   15   66.5   112.0
```

其中第一行相当于表头，是说明变量属性的，即说明各列的内容，如第一列是姓名，第二列是性别，第三列是年龄，第四列是身高 *(厘米)*，第五列是体重 *(磅)*. 从第二行至最后一行是变量的内容. 试从该文件中读出数据，并对身高和体重作回归分析.

解: (1) 建立 R 文件 (文件名： exam0203.R). 点击 "文件 | 建立新的程序脚本", R 窗口会弹出 R 编辑对话窗口 (R Editor), 在窗口中输入需要编辑的程序 (命令).

```
rt<-read.table("exam0203.txt", head=TRUE); rt
lm.sol<-lm(Weight~Height, data=rt)
summary(lm.sol)
```

下面解释一下每一个命令的意义. 文件的第一行是读文件 exam0203.txt, 并认为文本文件 exam0203.txt 中的第一行是文件的头 (head=TRUE); 否则 (FALSE) 文件中的第一行作为数据处理. 并将读出的内容放在变量 rt 中. 第二个 rt 是显示变量的内容 (如果一行执行多个命令，需用分号 (;) 隔开).

第二行是对数据 rt 中的重量 (Weight) 与高度 (Height) 作线性回归，其计算结果放置变量 lm.sol 中.

第三行是显示变量 lm.sol 中的详细内容，它将给出了回归的模型公式、残差的最小最大值等，和线性回归系数，以及估计与检验等. 有关具体含义将在后面作详细介绍.

(2) 执行文件 exam0203.R 的内容. 执行文件中的内容有几种方式, 第一种, 在 R 编辑窗口中用鼠标选中要执行的程序 (命令), 然后再单击 "执行行或选择项", 如图 2.4 所示. 第二种方法是单击 "编辑 | 执行一切". 第三种方法是采取复制、粘贴的方法将命令粘贴到主窗口，执行相应的命令.

执行后得到

```
> rt<-read.table("exam0203.txt", head=TRUE); rt
      Name Sex Age Height Weight
1    Alice   F  13   56.5   84.0
2    Becka   F  13   65.3   98.0
3     Gail   F  14   64.3   90.0
```

图 2.4: 执行 R 编辑窗口中的命令

```
4    Karen   F  12   56.3   77.0
5    Kathy   F  12   59.8   84.5
6     Mary   F  15   66.5  112.0
7    Sandy   F  11   51.3   50.5
8   Sharon   F  15   62.5  112.5
9    Tammy   F  14   62.8  102.5
10  Alfred   M  14   69.0  112.5
11    Duke   M  14   63.5  102.5
12   Guido   M  15   67.0  133.0
13   James   M  12   57.3   83.0
14 Jeffrey   M  13   62.5   84.0
15    John   M  12   59.0   99.5
16  Philip   M  16   72.0  150.0
17  Robert   M  12   64.8  128.0
18  Thomas   M  11   57.5   85.0
19 William   M  15   66.5  112.0
> lm.sol<-lm(Weight~Height, data=rt)
> summary(lm.sol)

Call:
lm(formula = Weight ~ Height, data = rt)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-17.6807  -6.0642   0.5115   9.2846  18.3698

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -143.0269    32.2746  -4.432 0.000366 ***
Height         3.8990     0.5161   7.555 7.89e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.23 on 17 degrees of freedom
Multiple R-Squared: 0.7705,     Adjusted R-squared: 0.757
F-statistic: 57.08 on 1 and 17 DF,  p-value: 7.887e-07
```

在执行中, 主窗口会重复显示编辑窗口的命令, 如主窗口显示的第一行与编辑窗口的第一行完全相同. 第二行以下的内容是显示变量 rt, 也就是文本文件 exam0203.txt 中的内容. 注意到, 显示内容比原内容增加了一列, 即标号列.

在 summary(lm.sol) 后面显示的是线性回归模型具体计算的结果.

从上面三个列子可以看出, 利用 R 软件计算各种统计量十分方便, 可以作图, 也可以从文件中读数据等. 掌握这些基本知识, 就可以用 R 软件来为我们服务.

为今后使用方便, 先介绍窗口中的菜单、快捷方式的意义.

2.1.3 R 主窗口命令与快捷方式

主窗口中的快捷方式如图 2.5 所示, 相关含义在主窗口命令中解释.

1. 文件

主窗口中的 "文件" 窗口如图 2.6 所示.

(1) 输入 R 代码...

执行要输入的程序. 单击 "输入 R 代码...", 打开 "选择要输入的程序文件" 窗口, 选择要输入的程序文件 (后缀为 .R), 如 MyFile.R. 选择好要输入的文件,

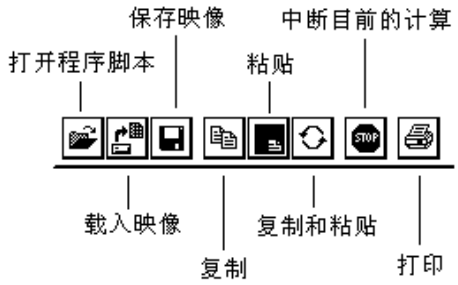


图 2.5: 主窗口中的快捷方式及意义

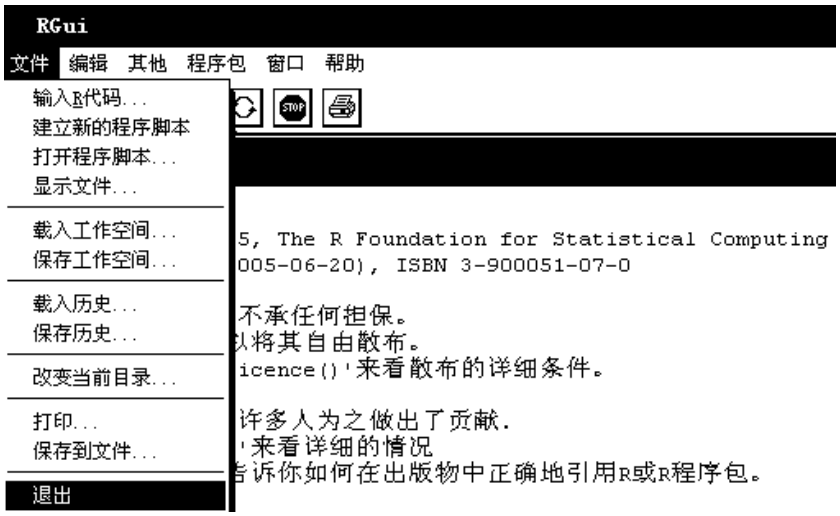


图 2.6: 主窗口中的文件菜单

按 “打开 (o)”. R 软件会执行该文件 (MyFile.R), 但在主窗口并不显示所执行的内容 (如有绘图命令, 则在另一窗口显示出所绘图形), 而只在主窗口显示

```
> source("MyFile.R")
```

当然, 在主窗口执行 source("MyFile.R") 命令, 具有同样的功能.

(2) 建立新的程序脚本

建立一个新程序脚本. 单击 “建立新的程序脚本”, 打开一个新的 R 程序编辑窗口, 输入你要编写的 R 程序. 输入完毕后, 选择保存, 并给一个文件名, 如 MyFile.R.

(3) 打开程序脚本...

打开已有的程序脚本. 单击 “打开程序脚本...”, 打开 “open script” 窗口, 选择一个 R 程序, 如 MyFile.R, 屏幕弹出 MyFile.R 编辑窗口, 可以利用这

个窗口对 R 程序 (MyFile.R) 进行编辑，或执行该程序中的部分或全部命令.

(4) 显示文件...

显示已有的文件. 单击 "显示文件...", 打开 "select files" 窗口，选择一个文件 (*.R 或 *.q), 如 MyFile.R. 屏幕弹出 MyFile.R 窗口，可利用该窗口执行该程序 (MyFile.R) 的部分或全部命令，但无法用该窗口对该程序进行编辑.

(5) 载入工作空间...

调入已保存的工作空间映像文件. 单击 "载入工作空间...", 打开 "选择要载入的映像" 窗口,在文件名窗口输入要载入的文件名,如 MyWorkSpace, 文件类型是 *.RData. 当调用成功后，保存在工作空间映像 MyWorkSpace.RData 中的全部命令就被调到内存中,这样在本次运算时,就不必重复工作空间 MyWorkSpace.RData 中已有的命令.

执行命令

```
> load("MyWorkSpace.RData")
```

具有同样的功能.

(6) 保存工作空间...

将当前的工作空间映像保存成一个文件. 单击 "保存工作空间...", 打开 "保存映像到" 窗口，在文件名窗口输入所需的文件名，如 MyWorkSpace, 文件类型为 *.RData, 按 "保存 (S)", 则当前的工作空间映像就保存到 MyWorkSpace.RData 文件中. 如果你保存的文件名与已有的文件名重名，则计算机会提示你是否替换已有文件，你可选择替换 (是 (Y)), 或不替换 (否 (N)).

保存工作空间映像的最大好处就是，在下次调用时，不必执行本次运算已执行的命令.

执行命令

```
> save.image("MyWorkSpace.RData")
```

具有同样的功能.

(7) 载入历史...

调入历史记录文件到内存中. 调入后，主窗口并不显示调入内容，只有在你按上下箭头，或 Ctrl+P 、 Ctrl+N, 才在命令行显示历史记录. 这样做可以减少你的键盘输入.

(8) 保存历史...

将在主窗口操作过的全部记录保存到一个文件中 (后缀为 .Rhistory), 如 MyWork.Rhistory. 该文件是纯文本文件, 用任何编辑器均能打开.

(9) 改变当前目录...

改变你当前的工作目录. 在缺省状态下, R 的工作目录是

C:\Program Files\R\rw2011

如图 2.7 所示. 在窗口输入所需的工作目录, 如 D:\XueYi\MyWorkSpace, 也可按 Browse, 选择所需要的工作目录, 按 OK 键确认.

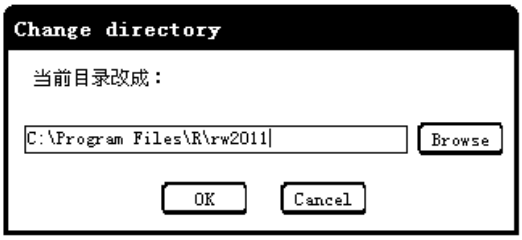


图 2.7: 改变当前目录窗口

(10) 打印...

打印文件.

(11) 保存到文件...

将主窗口的记录保存到文本文件中 (lastsave.txt).

(12) 退出

退出 R 系统. 如果退出前没有保存工作空间映像, 则系统会提示你保存工作空间映像, 你可选择保存 (是 (Y)), 或不保存 (否 (N)).

在主窗口执行 q() 命令, 具有同样的功能.

2. 编辑

主窗口中的 "编辑" 窗口如图 2.8 所示.

(1) 复制

将当前选中的文本复制到剪贴板中.

(2) 粘贴

将剪贴板中的内容粘贴到命令行.

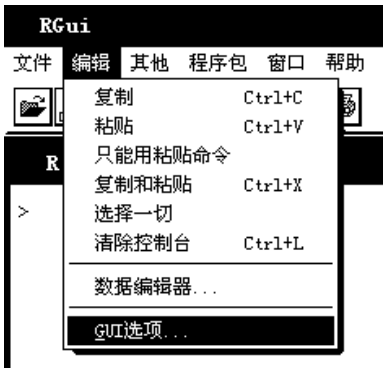


图 2.8: 主窗口中的编辑菜单

(3) 复制和粘贴

将当前选中的文本复制到剪贴板中，并将剪贴板中的内容粘贴到命令行.

(4) 选择一切

选定主窗口中的所有文本内容.

(5) 清除控制台

清除主窗口中的所有文本内容.

(6) 数据编辑器...

编辑已有的数据变量，并将新数据存入该变量. 例如，在例 2.3 中，将读出的数据放在变量 rt 中，现需要改动 rt 中的数据，单击 “数据编辑器”, 弹出 “Question” 窗口，输入变量 rt, 如图 2.9 所示. 按 OK, 弹出数据编辑窗口，如

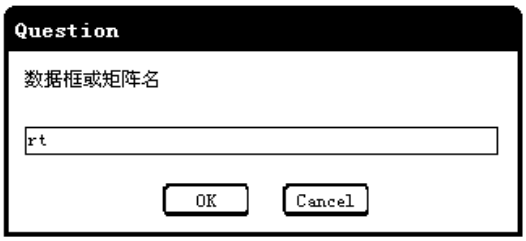


图 2.9: Question 窗口

图 2.10 所示. 你选择需要修改的数据进行修改，修改后关闭该窗口，此时变量 rt 中的数据已变成新数据.

在主窗口执行 fix(rt) 命令，可以达到同样的目的.

数据编辑器

	Name	Sex	Age	Height	Weight
1	Alice	F	13	56.5	84
2	Becka	F	13	65.3	98
3	Gail	F	14	64.3	90
4	Karen	F	12	56.3	77
5	Kathy	F	12	59.8	84.5
6	Mary	F	15	66.5	112
7	Sandy	F	11	51.3	50.5

图 2.10: 数据编辑器窗口

(7) GUI 选项. . .

改变 R 的图形用户界面. 单击 “GUI 选项. . . ”, 弹出 Rgui 配置编辑器. 你可根据需要更改配置编辑器中的内容. 建议初学者先不忙于更改配置, 使用缺省值.

3. 其他

主窗口中的 “其他” 窗口如图 2.11 所示.

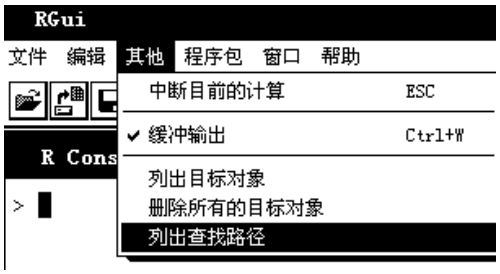


图 2.11: 主窗口中的其他菜单

(1) 中断目前的计算

单击 “中断目前的计算” 可停止当前正在执行的程序.

(2) 缓冲输出

单击 “缓冲输出” 会在 “缓冲输出” 前出现或取消 $\surd$, 即执行或取消缓冲输出.

(3) 列出目标对象

单击 “列出目标对象”, 列出全部变量名. 在主窗口执行 ls() 命令, 可以达到同样的目的.

(4) 删除所有目标对象

单击 “删除所有目标对象”, 将全部变量从内存中清除. 在主窗口执行

```
rm(list=ls(all=TRUE))
```

命令, 可以达到同样的目的.

(5) 列出查找路径

单击 “列出查找路径”, 列出查找文件 (或函数) 的路径或程序包, 以下基本的路径和程序包.

```
[1] ".GlobalEnv"        "package:methods"   "package:stats"
[4] "package:graphics"  "package:grDevices" "package:utils"
[7] "package:datasets"  "Autoloads"         "package:base"
```

在主窗口执行 search() 命令, 可以达到同样的目的.

4. 程序包

主窗口中的 “程序包” 窗口如图 2.12 所示.

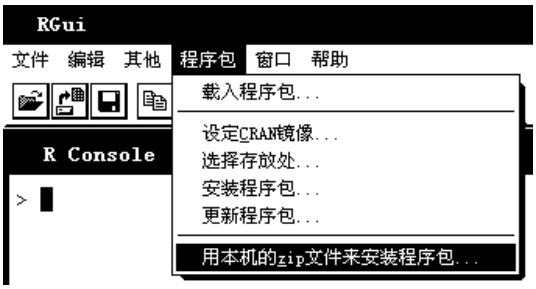


图 2.12: 主窗口中的 “程序包” 菜单

(1) 载入程序包...

R 软件除上述基本程序包外, 还有许多程序包, 只是在使用前需要调入. 如需要读 SPSS 软件的数据文件, 需要用函数 read.spss, 但在使用前需要调入 foreign 程序包.

单击 “载入程序包...”, 弹出选择程序窗口, 如图 2.13 所示. 选择 foreign, 按确定. 这样就可以使用 read.spss 函数.

(2) 选择 CRAN 镜像...

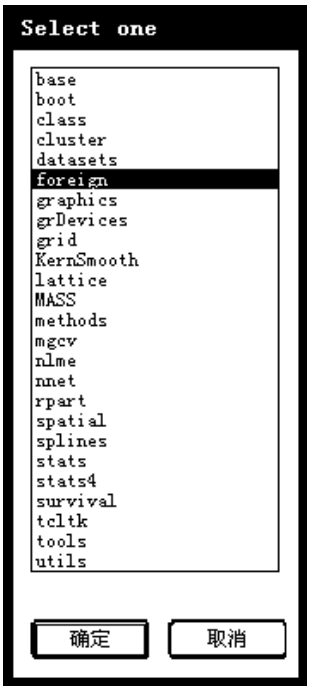


图 2.13: 选择程序包窗口

单击 “选择 CRAN 镜像”, 弹出 CRAN 镜像窗口, 选择一个镜像点, 按 “确定”, 联接到指定的镜像点.

(3) 选择存放处...

选择程序包库. 打开库窗口, 选择一个库, 按 “确定”. 计算机将自动联接到所选的库.

(4) 安装程序包...

安装新的程序包. 单击 “安装程序包”, 弹出 CRAN 镜像窗口, 选择合适的镜像点, 按 “确定”. 此时, 计算机将自动联接到指定的镜像点, 下载程序包, 并自动安装.

(5) 更新程序包...

更新已有的程序包. 单击 “更新程序包”, 弹出 CRAN 镜像窗口, 选择合适的镜像点, 按 “确定”. 此时, 计算机将自动联接到指定的镜像点, 下载程序包, 并自动更新.

(6) 用本机的 zip 文件来安装程序包...

打开 “Select files”, 选择需要安装的 zip 文件.

5. 窗口

主窗口中的 “窗口” 窗口如图 2.14 所示.

图 2.14: 主窗口中的 “窗口” 菜单

(1) 层叠

将所有窗口层叠.

(2) 平铺

将所有窗口平铺.

(3) 安排按钮

6. 帮助

主窗口中的 “帮助” 窗口如图 2.15 所示.

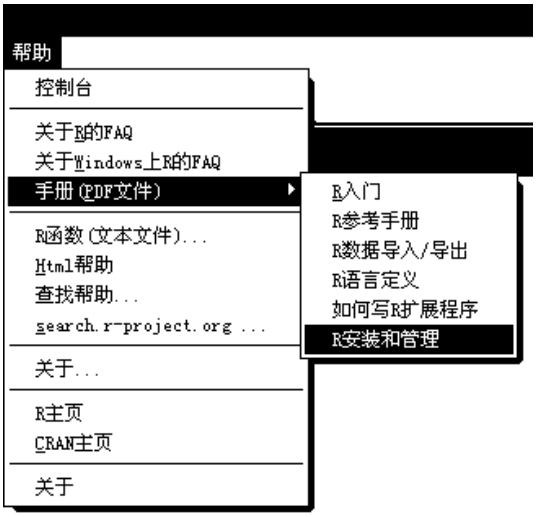


图 2.15: 主窗口中的 “帮助” 菜单

(1) 控制台

说明控制命令. 单击 “控制台”, 弹出说明控制命令窗口, 如图 2.16 所示. 在窗口中说明全部的控制命令.

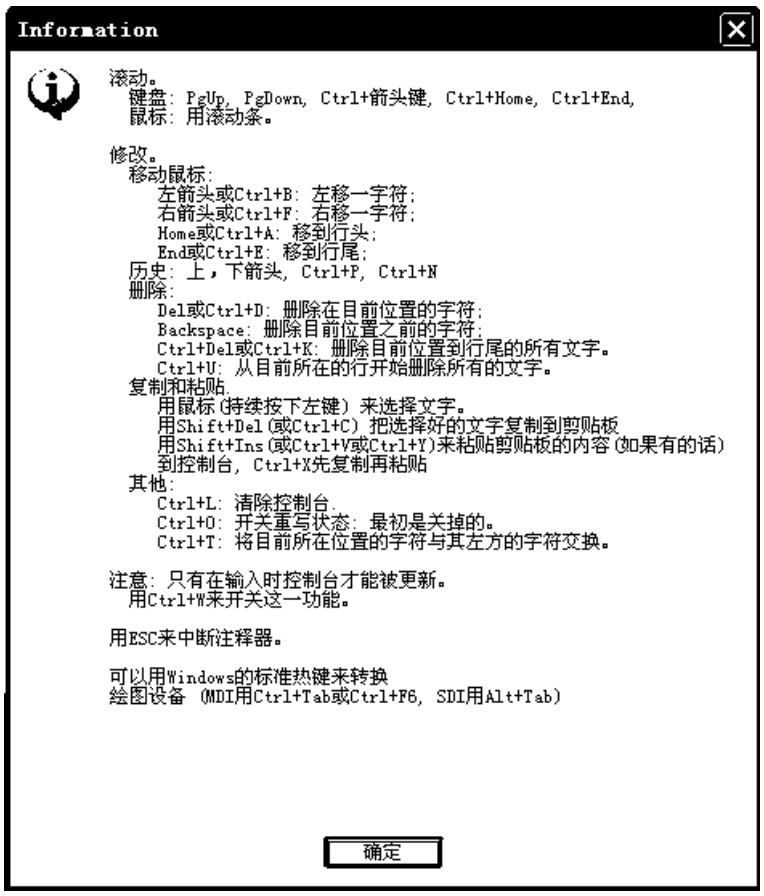


图 2.16: 控制命令窗口

(2) 关于 R 的 FAQ

R 常见问答. FAQ 是 frequently asked questions 的简写. 单击 “关于 R 的 FAQ”, 弹出 R FAQ 网页式窗口, 解释 R 的基本问题, R 的介绍、 R 基本知识、 R 语言与 S 语言, 以及 R 程序等.

(3) 关于 Windows 上 R 的 FAQ

关于 R 软件的进一步的常见问答. 单击 “关于 R 的 FAQ”, 弹出 R for Windows FAQ 网页式窗口, 其内容有安装与用户、程序包、 Windows 的特点、工作空间和控制台与字体等. 该窗口的问题更加深入.

(4) 手册 (PDF 文件)

给出 R 软件的使用手册. 有《 R 入门》、《 R 参考手册》、《 R 数据导入 / 导出》、《 R 语言的定义》、《写 R 扩展程序》和《 R 安装与管理》. 所有

手册均是 PDF 格式的文件 [1]. 这些手册为学习 R 软件提供了有利的帮助.

以上三条文本帮助文件是逐步深入的，用它们可以帮助使用者快速掌握 R 软件的使用.

(5) R 函数 (文本文件)...

帮助命令. 相当于 `help("Fun_Name")`. 单击 "R 函数 (文本文件)...", 出现帮助对话窗口，在窗口中输入需要帮助的函数名，如 `lm`(线性模型) 函数，按 `OK`, 则屏幕上会出现新的对话框，解释 `lm` 的意义与使用方法.

当帮助不成功时, 计算机会建议你使用 `help.search("read.spss")`(查找帮助).

(6) Html 帮助

网页形式的帮助窗口. 单击 "Html 帮助", 弹出网页形式的窗口菜单，使用者可以选择需要帮助的内容，双击，打开需要的内容.

(7) 查找帮助...

查找帮助. 相当于 `help.search("Fun_Name")`. 单击 "查找帮助...", 出现查找帮助对话窗口，在窗口中输入需要帮助的函数名，如 `lm`(线性模型) 函数，按 `OK` 键，则屏幕上会出现新的对话框，上面列出与 `lm` (线性模型) 有关的全部函数名 (包括广义线性模型函数名).

(8) search.r-project.org

在网站上查找. 单击 "search.r-project.org", 屏幕出现 "搜索邮件列表档案和文档" 对话框，输入查找内容，则计算机将自动联接网站 (http://search.r-project.org), 查找你需要的内容.

(9) 关于...

列出相关的函数与变量. 相当于 `apropos("Fun_Name")`. 单击 "关于...", 出现关于对话窗口，在窗口中输入需要查找的函数名或变量名，如 `lm`, 按 `OK`, 则屏幕上会出现新的对话框，上面列出含有字符串 `lm` 的全部函数名与变量名.

注意： "R 函数 (文本文件)..." 和 "关于..." 是在当前已有的程序包中查找，而 "查找帮助..." 是在整个程序包中查找. 例如， "帮助" 和 "关于" 对话框中输入 "`read.spss`" (读 SPSS 数据文件函数), 则主窗口出现 "`character(0)`",

[1]需要在你的计算机中安装 PDF 阅读软件 Adobe Acrobat Reader 才能阅读使用手册.

即无法查到. 而利用 “查找帮助” 对话框, 则屏幕上会出现新的窗口, 告诉你 read.spss 属于 foreign 程序包.

(10) R 主页

联接到 R 主页, 即 http://www.r-project.org/.

(11) CRAN 主页

联接到 CRAN 主页, 即 http://cran.r-project.org/.

(12) 关于

介绍 R 的版本信息.

2.2 数字、字符与向量

本节介绍 R 软件最简单的运算, 数字与向量的运算.

2.2.1 向量

1. 向量的赋值

R 软件中最简单的运算向量赋值. 如果打算建立一个名为 x 的向量, 相应的分量是 10.4, 5.6, 3.1, 6.4 和 21.7, 用 R 命令是

```
> x <- c(10.4, 5.6, 3.1, 6.4, 21.7)
```

其中 x 是变量名, <- 为赋值符, c( ) 为向量建立函数. 上述命令就是将函数 c( ) 中数据赋给变量 x.

另一个赋值函数是 assign( ), 其命令形式为

```
> assign("x", c(10.4, 5.6, 3.1, 6.4, 21.7))
```

第三种赋值形式为

```
> c(10.4, 5.6, 3.1, 6.4, 21.7) -> x
```

进一步有

```
> y <- c(x, 0, x)
```

定义变量 y 有 11 个分量, 其中两边是变量 x , 中间是零。

2. 向量的运算

对于向量可以作加 (+)、减 (−)、乘 (∗)、除 (/) 和乘方 (∧) 运算，其含意是对向量的每一个元素进行运算，其中加、减和数乘运算与我们通常的向量运算基本相同，如

```
> x <- c(-1, 0, 2);  y <- c(3, 8, 2)
> v <- 2*x + y + 1; v
[1] 2 9 7
```

第一行，输入向量 x 和 y. 第二行，将向量的计算结果赋给变量 v, 其中 2*x+y 是作通常的向量运算， +1 表示向量的每个分量均加 1. 分号后的 v 是为显示计算内容，因为 R 软件完成计算后进行赋值，并不显示相应的计算内容.

对于向量的乘法、除法、乘方运算，其意义是：对应向量的每个分量作乘法、除法和乘方运算，如

```
> x * y
[1] -3  0  4
> x / y
[1] -0.3333333  0.0000000  1.0000000
> x^2
[1] 1 0 4
> y^x
[1] 0.3333333 1.0000000 4.0000000
```

由于没有作赋值运算，所以， R 在运算后会直接显示计算结果.

另外， % / % 表示整数除法 (例如 5 % / % 3 为 1), % % 表示求余数 (例如 5 % % 3 为 2).

还可以作函数运算，如基本初等函数，如 log, exp, cos, tan, sqrt 等. 当自变量为向量时，函数的返回值也是向量，即每个分量取相应的函数值. 如

```
> exp(x)
[1] 0.3678794 1.0000000 7.3890561
> sqrt(y)
[1] 1.732051 2.828427 1.414214
```

但 sqrt(-2) 会给出 NAN 和相应的警告信息，因为负数不能开方. 但如果需要作复数运算，则输入形式应改为 sqrt(-2+0i).

3. 与向量运算有关的函数

介绍一些与向量运算有关的函数.

(1) 求向量的最小值、最大值和范围的函数.

$\min(x)$ 、 $\max(x)$ 、 $\text{range}(x)$ 分别表示求向量 x 的最小分量、最大分量和向量 x 的范围, 即 $[\min(x), \max(x)]$. 如

```
> x <- c(10, 6, 4, 7, 8)
> min(x)
[1] 4
> max(x)
[1] 10
> range(x)
[1]  4 10
```

与 `min()`(`max()`) 有关的函数是 `which.min()`(`which.max()`), 表示在第几个分量求到最小 (最大) 值, 如

```
> which.min(x)
[1] 3
> which.max(x)
[1] 1
```

(2) 求和函数、求乘积函数.

$\text{sum}(x)$ 表示求向量 x 分量之和, 即 $\sum_{i=1}^{n} x_i$. $\text{prod}(x)$ 表示求向量 x 分量联乘积, 即 $\prod_{i=1}^{n} x_i$. 还有 $\text{length}(x)$ 表示求向量 x 分量的个数, 即 n.

(3) 中位数、均值、方差、标准差和顺序统计量.

$\text{median}(x)$ 表示求向量 x 的中位数. $\text{mean}(x)$ 表示求向量 x 的均值, 即 $\text{sum}(x)/\text{length}(x)$. $\text{var}(x)$ 表示求向量 x 的方差, 即

$$\text{var}(x) = \text{sum}\left((x - \text{mean}(x))^2\right)/(\text{length}(x) - 1).$$

$\text{sd}(x)$ 表示求向量 x 的标准差, 即 $\texttt{sd}(x) = \sqrt{\texttt{var}(x)}$.

$\text{sort}(x)$ 表示求与向量 x 大小相同, 按递增顺序排列的向量, 即顺序统计量. 相应的下标由 $\text{order}(x)$ 或 $\text{sort.list}(x)$ 列出. 例如, 当 `x<-c(10, 6, 4, 7, 8)` 时, $\text{sum}(x)$ 、 $\text{prod}(x)$ 、 $\text{length}(x)$ 、 $\text{median}(x)$ 、 $\text{mean}(x)$ 、 $\text{var}(x)$ 和 $\text{sort}(x)$ 的计算结果分别是 35 、 13440 、 5 、 7 、 7 、 5 和 4 6 7 8 10.

有关均值、方差等统计量的性质和函数的使用方法, 在第三章还会介绍.

2.2.2 产生有规律的序列

1. 等差数列

`a:b` 表示从 `a` 开始，逐项加 1(或减 1), 直到 `b` 为止. 如 `x <- 1:30` 表示向量 $x = (1, 2, \cdots, 30)$, `x <- 30:1` 表示向量 $x = (30, 29, \cdots, 1)$. 当 `a` 为实数， `b` 为整数时，向量 `a:b` 是实数，其间隔差 1. 而当 `a` 为整数， `b` 为实数时， `a:b` 表示其间隔差 1 的整数向量. 如

```
> 2.312:6
[1] 2.312 3.312 4.312 5.312
> 4:7.6
[1] 4 5 6 7
```

注意： `x <- 2*1:15` 并不是表示 2 到 15, 而是表示向量 $x = (2, 4, \cdots, 30)$, 即 `x <- 2 * (1:15)`, 也就是等差运算优于乘法运算. 同理， `1:n-1` 并不是表示 1 到 n-1, 而是表示向量 1:n 减去 1. 若需要表示 1 到 n-1, 则需要对 n-1 加括号. 比较下面两种表示的差别.

```
> n<-5
> 1:n-1
[1] 0 1 2 3 4
> 1:(n-1)
[1] 1 2 3 4
```

注意：这一点对于初学者非常容易引起混淆.

2. 等间隔函数

`seq()` 函数是更一般的函数，它产生等距间隔的数列，其基本形式为

seq(from=value1, to= value2, by=value3)

即从 value1 开始，到 value2 结束，中间的间隔为 value3. 如

```
> seq(-5, 5, by=.2) -> s1
```

表示向量 $s1 = (-5.0, -4.8, -4.6, \cdots, 4.6, 4.8, 5.0)$. 从上述定义来看， `seq(2,10)` 等价于 2:10, 在不作特别声明的情况下，其间隔为 1.

对于 `seq` 函数还有另一种使用方式,

seq(length=value2, from=value1, by=value3)

即从 value1 开始，间隔为 value3, 其向量的长度为 value2. 如

```
> s2 <- seq(length=51, from=-5, by=.2)
```

产生的 s2 与向量 s1 相同。

3. 重复函数

rep() 是重复函数，它可以将某一向量重复若干次再放入新的变量中，如

```
> s <- rep(x, times=3)
```

即将变量 x 重复 3 倍，放在变量 s 中. 如

```
> x <- c(1, 4, 6.25); x
[1] 1.00 4.00 6.25
> s <- rep(x, times=3); s
[1] 1.00 4.00 6.25 1.00 4.00 6.25 1.00 4.00 6.25
```

2.2.3 逻辑向量

与其它语言一样， R 软件允许使用逻辑操作. 当逻辑运算为真时，返回值为 TRUE, 当逻辑运算为假时，返回值为 FALSE. 例如

```
> x <- 1:7
> l <- x > 3
```

其结果为

```
> l
[1] FALSE  FALSE  FALSE  TRUE  TRUE  TRUE  TRUE
```

逻辑运算符有 <，<=，>，>=，== (表示等于) 和 !=(表示不等于). 如果 c1 和 c2 是两个逻辑表达式，则 c1 & c2 表示 c1 “与” c2, c1 | c2 表示 c1 “或” c2, !c1 表示 “非 c1 ”.

逻辑变量也可以赋值，如

```
> z <- c(TRUE, FALSE, F, T)
```

其中 T 是 TRUE 的简写， F 是 FALSE 简写.

判断一个逻辑向量是否都为真值的函数是 all, 如

```
> all(c(1, 2, 3, 4, 5, 6, 7) > 3)
[1] FALSE
```

判断是否其中有真值的函数是 any, 如

```
> any(c(1, 2, 3, 4, 5, 6, 7) > 3)
[1] TURE
```

2.2.4 缺失数据

用 NA 表示某处的数据缺省或缺失. 如

```
> z <- c(1:3, NA); z
[1]  1  2  3  NA
```

函数 is.na() 是检测缺失数据的函数, 如果返回值为真 (TRUE), 则说明此数据是缺失数据. 如果返回值为假 (FALSE), 则此数据不是缺失数据. 如

```
> ind <- is.na(z); ind
[1] FALSE FALSE FALSE  TRUE
```

如果需要将缺失数据改为 0, 则用如下命令

```
> z[is.na(z)] <- 0; z
[1] 1 2 3 0
```

类似的函数还有 is.nan()(检测数据是否不确定, TRUE 为不确定, FALSE 为确定), is.finite()(检测数据是否有限, TRUE 为有限, FALSE 为无穷), is.infinite()(检测数据是否为无穷, TRUE 为无穷, FALSE 为有限). 例如,

```
> x<-c(0/1, 0/0, 1/0, NA); x
[1]   0 NaN Inf  NA
> is.nan(x)
[1] FALSE  TRUE FALSE FALSE
> is.finite(x)
[1]  TRUE FALSE FALSE FALSE
> is.infinite(x)
[1] FALSE FALSE  TRUE FALSE
> is.na(x)
[1] FALSE  TRUE FALSE  TRUE
```

在 x 的四个分量中, $0/1$ 为 0, 只有在 is.finite 的检测下是真, 其余均为假. $0/0$ 为不确定, 但对函数 is.nan 和 is.na 的检测下均为真, 这是因为不确定数据也认为是缺失数据. $1/0$ 为无穷, 因此只在 is.infinite 检测下为真. NA

为缺失数据，只有在 is.na 检测下为真，因为缺失数据并不是不确定数据，所以在 is.nan 检测下仍为假.

如果对不确定数据、缺失数据赋值，可以采用对缺失数据赋值的方法为它们赋值.

2.2.5 字符型向量

向量元素可以取字符串值. 例如,

```
> y <-c ("er", "sdf", "eir", "jk", "dim")
```

或

```
> c("er", "sdf", "eir", "jk", "dim") -> y
```

则得到

```
> y
[1] "er"  "sdf" "eir" "jk"  "dim"
```

可用 paste 函数用来把它的自变量连成一个字符串，中间用空格分开，例如,

```
> paste("My","Job")
[1] "My Job"
```

连接的自变量可以是向量，这时各对应元素连接起来，长度不相同时较短的向量被重复使用. 自变量可以是数值向量，连接时自动转换成适当的字符串表示，例如,

```
> labs<-paste("X", 1:6, sep = ""); labs
[1] "X1" "X2" "X3" "X4" "X5" "X6"
```

分隔用的字符可以用 sep 参数指定，例如下例产生若干个文件名:

```
> paste("result.", 1:4, sep="")
[1] "result.1" "result.2" "result.3" "result.4"
```

关于 paste 函数，还有以下几种用法.

```
> paste(1:10) # same as as.character(1:10)
[1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10"
> paste("Today is", date())
[1] "Today is Tue Sep 13 16:16:29 2005"
> paste(c('a', 'b'), collapse='.')
[1] "a.b"
```

2.2.6 复数向量

R 支持复数运算. 复数常量只要用通常的格式, 如 `3.5+2.1i`. `complex` 模式的向量为复数元素的向量, 可以用 `complex()` 函数生成复数向量. 如

```
> x <- seq(-pi, pi, by=pi/10)
> y <- sin(x)
> z <- complex(re=x, im=y)
> plot(z)
> lines(z)
```

其中第一行是给出向量 `x` 的值, 第二行是计算向量 `y` 的值, 第三行是构造复数

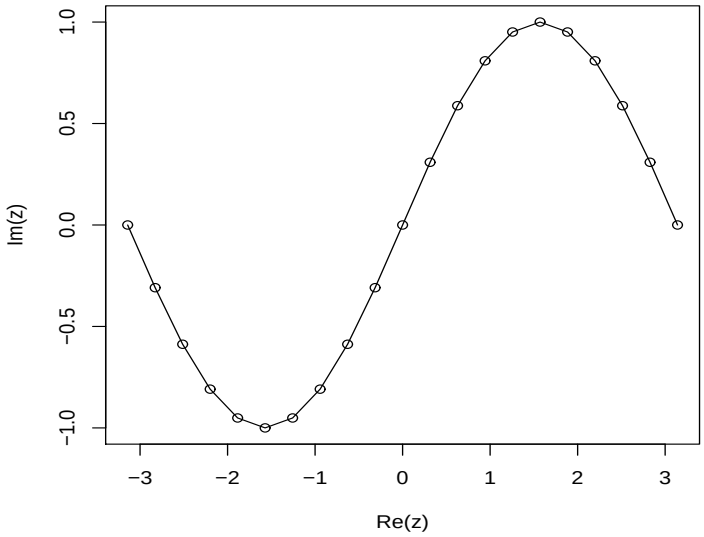


图 2.17: 复数 $z = x + i\sin(x)$ 的散点图和折线图

向量, 其中 `x` 为实部, `y` 为虚部. 第四行是绘出复数向量 `z` 的散点图, 第五行是用实线连接这些散点. 图 2.17 给出了相应的图形.

对于复数运算, `Re()` 是计算复数的实部, `Im()` 是计算计算复数的虚部, `Mod()` 是计算复数的模, `Arg()` 是计算复数的幅角.

2.2.7 向量下标运算

R 软件提供了十分灵活的访问向量元素和向量子集的功能. 某一个元素只要用 `x[i]` 的格式访问, 其中 `x` 是一个向量名, 或一个取向量值的表达式, 如

```
> x <- c(1,4,7)
> x[2]
[1] 4
> (c(1, 3, 5) + 5)[2]
[1] 8
```

可以单独改变一个元素的值，如

```
> x[2] <- 125
> x
[1]   1  125   7
> x[c(1,3)] <- c(144, 169)
> x
[1] 144 125 169
```

1. 逻辑向量

v 为和 x 等长的逻辑向量， x[v] 表示取出所有 v 为真值的元素，如

```
> x <- c(1,4,7)
> x < 5
[1]  TRUE  TRUE  FALSE
> x[x<5]
[1] 1 4
```

可以将向量中缺失数据赋为 0, 如

```
> z <- c(-1, 1:3, NA)
> z[is.na(z)] <- 0
> z
[1] -1  1  2  3  0
```

也可以将向量中非缺失数据赋给另一个向量，如

```
> z <- c(-1, 1:3, NA)
> y <- z[!is.na(z)]
> y
[1] -1  1  2  3
```

或作相应的运算,

```
> (z+1)[(!is.na(z)) & z>0] -> x
> x
[1] 2 3 4
```

改变部分元素值的技术与逻辑值下标方法结合可以定义向量的分段函数, 例如, 要定义

$$y=\begin{cases}1-x, & x<0\\ 1+x, & x\geq 0\end{cases},$$

可以用

```
> y <- numeric(length(x))
> y[x<0] <- 1 - x[x<0]
> y[x>=0] <- 1 + x[x>=0]
```

来表示, 其中 numeric 函数是产生数值型向量.

2. 下标的正整数运算

v 为一个向量, 下标取值在 1 到 length(v) 之间, 取值允许重复, 例如,

```
> v <- 10:20
> v[c(1,3,5,9)]
[1] 10 12 14 18
> v[1:5]
[1] 10 11 12 13 14
> v[c(1,2,3,2,1)]
[1] 10 11 12 11 10
> c("a","b","c")[rep(c(2,1,3), times=3)]
[1] "b" "a" "c" "b" "a" "c" "b" "a" "c"
```

3. 下标的负整数运算

v 为一个向量, 下标取值在 $-$lenght(x) 到 -1 之间, 如

```
> v[-(1:5)]
[1] 15 16 17 18 19 20
```

表示扣除相应的元素.

4. 取字符型值的下标向量

在定义向量时可以给元素加上名字，如

```
> ages <- c(Li=33, Zhang=29, Liu=18)
> ages
   Li Zhang   Liu
   33    29    18
```

这样定义的向量可以用通常的办法访问，另外还可以用元素名字来访问元素或元素子集，例如：

```
> ages["Zhang"]
Zhang
   29
```

向量元素名可以后加，如

```
> fruit <- c(5, 10, 1, 20)
> names(fruit) <- c("orange", "banana", "apple", "peach")
> fruit
orange banana  apple  peach
     5     10      1     20
```

2.3 对象和它的模式与属性

R 是一种基于对象的语言. R 的对象包含了若干个元素作为其数据，另外还可以有一些特殊数据称为属性 (attribute), 并规定了一些特定操作 (如打印、绘图). 比如，一个向量是一个对象，一个图形也是一个对象. R 对象分为单纯 (atomic) 对象和复合 (recursive) 对象两种，单纯对象的所有元素都是同一种基本类型 (如数值、字符串), 元素不再是对象；复合对象的元素可以是不同类型的对象，每一个元素是一个对象.

2.3.1 固有属性： mode 和 length

R 对象都有两个基本的属性： `mode`(类型) 属性和 `length`(长度) 属性. 比如向量的类型为 `logical`(逻辑型) 、 `numeric`(数值型) 、 `complex`(复数型) 、 `character`(字符型), 比如

```
> mode(c(1,3,5)>5)
```

```
[1] "logical"
```

R 对象有一种特别的 null(空值型) 型，只有一个特殊的 NULL 值为这种类型，表示没有值 (不同于 NA, NA 是一种特殊值，而 NULL 根本没有对象值).

要判断某对象是否某类型，有许多个类似于 is.numeric() 的函数可以完成. is.numeric(x) 用来检验对象 x 是否为数值型，它返回一个逻辑型结果. is.character() 可以检验对象是否为字符型，等等. 如

```
> z <- 0:9
> is.numeric(z)
[1] TRUE
> is.character(z)
[1] FALSE
```

长度属性表示 R 对象元素的个数，比如

```
> length(2:4)
[1] 3
> length(z)
[1] 9
```

注意向量允许长度为 0, 如数值型向量长度为零表示为 numeric() 或 numeric(0), 字符型向量长度为零表示为 character() 或 character(0).

R 可以强制进行类型转换，例如

```
> digits <- as.character(z); digits
[1] "0" "1" "2" "3" "4" "5" "6" "7" "8" "9"
> d <- as.numeric(digits); d
[1] 0 1 2 3 4 5 6 7 8 9
```

第一个赋值把数值型的 z 转换为字符型的 digits. 第二个赋值把 digits 又转换为了数值型的 d, 这时 d 和 z 是一样的了. R 还有许多这样的以 as. 开头的类型转换函数.

2.3.2 修改对象的长度

对象可以取 0 长度或正整数为长度. R 允许对超出对象长度的下标赋值，这时对象长度自动伸长以包括此下标，未赋值的元素取缺失值 (NA), 例如

```
> x <- numeric()
> x[3] <- 17
> x
[1] NA NA 17
```

要增加对象的长度只需作赋值运算就可以了，如

```
> x <- 1:3
> x <- 1:4
[1] 1 2 3 4
```

要缩短对象的长度又怎么办呢？只要给它赋一个长度短的子集就可以了．如

```
> x <- x[1:2]
> x
[1] 1 2
> alpha <- 1:10
> alpha <- alpha[2 * 1:5]
> alpha
[1]  2  4  6  8  10
```

或给对象的长度赋值，如

```
> length(alpha) <- 3
> alpha
[1]  2  4  6
```

2.3.3 attributes() 和 attr() 函数

attributes(object) 返回对象 object 的各特殊属性组成的列表，不包括固有属性 mode 和 length. 例如,

```
> x <- c(apple=2.5,orange=2.1); x
 apple orange
   2.5    2.1
> attributes(x)
$names
[1] "apple"  "orange"
```

可以用 attr(object, name) 的形式存取对象 object 的名为 name 的属性. 例如,

```
> attr(x,"names")
[1] "apple"  "orange"
```

也可以把 attr() 函数写作赋值的左边以改变属性值或定义新的属性, 例如,

```
> attr(x,"names") <- c("apple","grapes"); x
 apple grapes
   2.5    2.1
> attr(x,"type") <- "fruit"; x
 apple grapes
   2.5    2.1
attr(,"type")
[1] "fruit"
> attributes(x)
$names
[1] "apple"  "grapes"

$type
[1] "fruit"
```

2.3.4 对象的 class 属性

在 R 中可以用特殊的 class 属性来支持面向对象的编程风格, 对象的 class 属性用来区分对象的类, 可以写出通用函数根据对象类的不同进行不同的操作, 比如, print() 函数对于向量和矩阵的显示方法就不同, plot() 函数对不同类的自变量作不同的图形.

为了暂时去掉一个有类的对象的 class 属性, 可以使用 unclass(object) 函数.

2.4 因子

统计中的变量有几种重要类别: 区间变量、名义变量和有序变量. 区间变量取连续的数值, 可以进行求和、平均值等运算. 名义变量和有序变量取离散值,

可以用数值代表，也可以是字符型值，其具体数值没有加减乘除的意义，不能用来计算，而只能用来分类或计数. 名义变量如性别、省份、职业，有序变量如班级、名次.

2.4.1 factor() 函数

因为离散变量有各种不同表示方法，在 R 软件中，为了统一起见，使用因子 (factor) 来表示这种类型的变量. 例如，知道 5 位学生的性别，用因子变量表示

```
> sex <- c("M","F","M","M", "F")
> sexf <- factor(sex); sexf
[1] M F M M F
Levels:  F M
```

函数 factor() 用来把一个向量编码成为一个因子. 其一般形式为:

```
factor(x, levels = sort(unique(x), na.last = TRUE),
       labels, exclude = NA, ordered = FALSE)
```

其中 x 是向量， levels 是水平，可以自行指定各离散取值，不指定时由 x 的不同值来求得. labels 可以用来指定各水平的标签，不指定时用各离散取值的对应字符串. exclude 参数用来指定要转换为缺失值 (NA) 的元素值集合. 如果指定了 levels, 则因子的第 i 个元素当它等于水平中第 j 个时元素值取 "j", 如果它的值没有出现在 levels 中，则对应因子元素值取 NA. ordered 取值为真 (TRUE) 时，表示因子水平是有次序的 (按编码次序); 否则 (缺省值) 是无次序的.

可以用 is.factor() 检验对象是否因子，用 as.factor() 把一个向量转换成一个因子.

用函数 levels() 可以得到因子的水平，如

```
> sex.level <- levels(sexf); sex.level
[1] "F" "M"
```

对于因子向量，可用函数 table() 来统计各类数据的频数. 例如,

```
> sex.tab <- table(sexf); sex.tab
sexf
F  M
2  3
```

表示男性 3 人，女性 2 人. table() 的结果是一个带元素名的向量，元素名为因子水平，元素值为该水平的出现频数. 关于 table 的使用方法，在后面还会讲到.

2.4.2 tapply() 函数

我们除了知道 5 位学生的性别，还知道 5 位学生的身高，分组求身高的平均值.

```
> height <- c(174, 165, 180, 171, 160)
> tapply(height, sex, mean)
    F     M
162.5 175.0
```

函数 tapply() 的一般使作格式为:

```
tapply(X, INDEX, FUN = NULL, ..., simplify = TRUE)
```

其中 X 是一对象，通常是一向量， INDEX 是与 X 有同样长度的因子， FUN 是需要计算的函数， simplify 是逻辑变量，取为 TRUE(缺省) 和 FALSE.

2.4.3 gl() 函数

gl() 函数可以方便地产生因子，其一般用法是

```
gl(n, k, length = n*k, labels = 1:n, ordered = FALSE)
```

其中 n 为水平数， k 为重复的次数， length 为结果的长度， labels 是一个 n 维向量，表示因子水平， ordered 是逻辑变量，表示是否为有序因子，缺省值为 FALSE. 如

```
> gl(3,5)
 [1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
Levels:  1 2 3
> gl(3,1,15)
 [1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
Levels:  1 2 3
```

2.5 多维数组和矩阵

2.5.1 生成数组或矩阵

数组 (array) 可以看成是带多个下标的类型相同的元素的集合，常用的是数值型的数组如矩阵，也可以有其它类型 (如字符型、逻辑型、复数型). R 可以很容易地生成和处理数组，特别是矩阵 (二维数组).

数组有一个特征属性叫做维数向量 (dim 属性), 维数向量是一个元素取正整数值的向量，其长度是数组的维数，比如维数向量有两个元素时数组为二维数组 (矩阵). 维数向量的每一个元素指定了该下标的上界，下标的下界总为 1.

1. 将向量定义成数组

向量只有定义了维数向量 (dim 属性) 后才能被看作是数组. 比如：

```
> z<-1:12
> dim(z)<-c(3,4)
> z
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

注意：矩阵的元素是按列存放的. 也可以把向量定义为一维数组，例如：

```
> dim(z)<-12
> z
 [1]  1  2  3  4  5  6  7  8  9 10 11 12
```

2. 用 `array()` 函数构造多维数组

R 软件可以用 `array()` 函数直接构造数组，其构造形式为

```
array(data = NA, dim = length(data), dimnames = NULL)
```

其中 `data` 是一个向量数据， `dim` 是数组各维的长度，缺省时为原向量的长度. `dimnames` 是数组维的名字，缺省时为空. 如

```
> X <- array(1:20,dim=c(4,5))
```

产生一个 4×5 的二维数组 (矩阵), 即

```
> X
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
```

另一种方式为

```
> Z <- array(0,dim=c(3, 4, 2))
```

它定义了一个 $3\times4\times2$ 的三维数组, 其元素均为 0. 这种方法常用来对数组作初始化.

3. 用 matrix() 函数构造矩阵

函数 matrix() 是构造矩阵 (二维数组) 的函数, 其构造形式为

```
matrix(data=NA, nrow=1, ncol=1, byrow=FALSE, dimnames=NULL)
```

其中 data 是一个向量数据, nrow 是矩阵的行数, ncol 是矩阵的列数. 当 byrow=TRUE 时, 生成矩阵的数据按行放置, 缺省时相当于 byrow=FALSE, 数据按列放置. dimnames 是数组维的名字, 缺省时为空.

如构造一个 3×5 阶的矩阵

```
> A<-matrix(1:15, nrow=3,ncol=5,byrow=TRUE)
> A
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    3    4    5
[2,]    6    7    8    9   10
[3,]   11   12   13   14   15
```

注意, 下面两种格式与前面的格式是等价的.

```
> A<-matrix(1:15, nrow=3,byrow=TRUE)
> A<-matrix(1:15, ncol=5,byrow=TRUE)
```

如果将语句中的 byrow=TRUE 去掉, 则数据按列放置.

2.5.2 数组下标

数组与向量一样, 可以对数组中的某些元素进行访问, 或进行运算.

1. 数组下标

要访问数组的某个元素，只要写出数组名和方括号内的用逗号分开的下标即可，如 a[2，1，2]. 如

```
> a <- 1:24
> dim(a) <- c(2,3,4)
> a[2, 1, 2]
[1] 8
```

更进一步还可以在每一个下标位置写一个下标向量，表示这一维取出所有指定下标的元素，如 a[1，2:3，2:3] 取出所有第一下标为 1，第二下标为 2 或 3，第三下标为 2 或 3 的元素. 如

```
> a[1, 2:3, 2:3]
     [,1] [,2]
[1,]    9   15
[2,]   11   17
```

注意，因为第一维只有一个下标，所以退化了，得到的是一个维数向量为 2×2 的数组.

另外，如果略写某一维的下标，则表示该维全选. 例如，

```
> a[1, , ]
     [,1] [,2] [,3] [,4]
[1,]    1    7   13   19
[2,]    3    9   15   21
[3,]    5   11   17   23
```

取出所有第一下标为 1 的元素，得到一个形状为 3×4 的数组.

```
> a[ , 2, ]
     [,1] [,2] [,3] [,4]
[1,]    3    9   15   21
[2,]    4   10   16   22
```

取出所有第二下标为 2 的元素得到一个 2×4 的数组.

```
> a[1,1, ]
[1]  1  7  13 19
```

则只能得到一个长度为 4 的向量，不再是数组. a[, ,] 或 a[] 都表示整个数组. 比如

```
> a []<-0
```

可以在不改变数组维数的条件下把元素都赋成 0.

还有一种特殊下标办法是对于数组只用一个下标向量 (是向量，不是数组), 比如

```
> a[3:10]
[1]  3  4  5  6  7  8  9 10
```

这时忽略数组的维数信息把表达式看作是对数组的数据向量取子集.

2. 不规则的数组下标

在 R 语言中, 甚至可以把数组中的任意位置的元素作为数组访问, 其方法是用一个二维数组作为数组的下标, 二维数组的每一行是一个元素的下标, 列数为数组的维数. 例如, 要把上面的形状为 $2\times3\times4$ 的数组 a 的第 [1,1,1], [2,2,3], [1,3,4], [2,1,4] 号共四个元素作为一个整体访问，先定义一个包含这些下标作为行的二维数组：

```
> b <- matrix(c(1,1,1,2,2,3,1,3,4,2,1,4), ncol=3, byrow=T)
> b
     [,1] [,2] [,3]
[1,]    1    1    1
[2,]    2    2    3
[3,]    1    3    4
[4,]    2    1    4
> a[b]
[1]  1 16 23 20
```

注意取出的是一个向量. 我们还可以对这几个元素赋值，如：

```
> a[b] <- c(101,102,103,104)
```

或

```
> a[b] <- 0
```

2.5.3 数组的四则运算

可以对数组之间进行四则运算 (+、 −、 ∗、 /), 这时进行的是数组对应元素的四则运算, 参加运算的数组一般应该是相同形状的 (dim 属性完全相同). 例如,

```
> A <- matrix(1:6, nrow=2, byrow=T); A
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
> B <- matrix(1:6, nrow=2); B
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> C <- matrix(c(1,2,2,3,3,4), nrow=2); C
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    2    3    4
> D <- 2*C+A/B; D
     [,1]     [,2] [,3]
[1,]    3 4.666667  6.6
[2,]    6 7.250000  9.0
```

从这个例子可以看到, 数组的加、减法运算和数乘运算满足原矩阵运算的性质, 但数组的乘、除法运算实际上是数组中对应位置的元素作运算.

形状不一致的向量 (或数组) 也可以进行四则运算, 一般的规则是将向量 (或数组) 中的数据与对应向量 (或数组) 中的数据进行运算, 把短向量 (或数组) 的数据循环使用, 从而可以与长向量 (或数组) 数据进行匹配, 并尽可能保留共同的数组属性. 例如,

```
> x1 <- c(100,200)
> x2 <- 1:6
> x1+x2
[1] 101 202 103 204 105 206
> x3 <- matrix(1:6, nrow=3)
```

```
> x1+x3
     [,1] [,2]
[1,]  101  204
[2,]  202  105
[3,]  103  206
```

可以看到，当向量与数组共同运算时，向量按列匹配. 当两个数组不匹配时， R 会提出警告. 如

```
> x2 <- 1:5
> x1+x2
[1] 101 202 103 204 105
```

警告信息:

长的目标对象长度不是短的目标对象长度的整倍数 `in: x1 + x2`

2.5.4 矩阵的运算

这里简单地介绍 R 软件中矩阵的基本运算.

1. 转置运算

对于矩阵 A, 函数 `t(A)` 表示矩阵 A 的转置，即 A^T. 如

```
> A<-matrix(1:6,nrow=2); A
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> t(A)
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
```

2. 求方阵的行列式

函数 `det()` 是求方阵行列式的值. 如

```
>  det(matrix(1:4, ncol=2))
[1] -2
```

3. 向量的内积

对于 n 维向量 x, 可以看成 $n \times 1$ 阶矩阵或 $1 \times n$ 阶矩阵. 若 x 与 y 是相同维数的向量, 则 `x %*% y`表示x与y作内积. 例如,

```
> x <- 1:5; y <- 2*1:5
> x %*% y
     [,1]
[1,]  110
```

函数 `crossprod()` 是内积运算函数 (表示交叉乘积), `crossprod(x,y)` 计算向量 x 与 y 的内积, 即 `'t(x) %*% y'`. `crossprod(x)`表示x与x的内积, 即 $\|x\|_2^2$.

类似地, `tcrossprod(x,y)` 表示 `'x %*% t(y)'`, 即x与y的外积, 也称为叉积. `tcrossprod(x)` 表示 x 与 x 作外积.

4. 向量的外积 (叉积)

设 x, y 是 n 维向量, 则 `x %o% y`表示x与y作外积. 例如,

```
> x <- 1:5; y <- 2*1:5
> x %o% y
     [,1] [,2] [,3] [,4] [,5]
[1,]    2    4    6    8   10
[2,]    4    8   12   16   20
[3,]    6   12   18   24   30
[4,]    8   16   24   32   40
[5,]   10   20   30   40   50
```

函数 `outer()` 是外积运算函数, `outer(x,y)` 计算向量 x 与 y 的外积, 它等价于 `x %o% y`.

函数 `outer()` 的一般调用格式为

```
outer(X, Y, fun = "*", ...)
```

其中 `X`, `Y` 矩阵 (或向量), `fun` 是作外积运算函数, 缺省值为乘法运算. 函数 `outer()` 在绘制三维曲面时非常有用, 它可生成一个 `X` 和 `Y` 的网格. 关于它在绘制三维曲面的用法将在第三章 3.3.1 节中讲到.

5. 矩阵的乘法

如果矩阵 A 和 B 具有相同的维数，则 A * B 表示矩阵中对应的元素的乘积， A % * % B 表示通常意义下的两个矩阵的乘积 (当然要求矩阵 A 的列数等于矩阵 B 的行数). 如

```
> A <- array(1:9,dim=(c(3,3)))
> B <- array(9:1,dim=(c(3,3)))
> C <- A * B; C
     [,1] [,2] [,3]
[1,]    9   24   21
[2,]   16   25   16
[3,]   21   24    9
> D <- A %*% B; D
     [,1] [,2] [,3]
[1,]   90   54   18
[2,]  114   69   24
[3,]  138   84   30
```

由乘法的运算规则可以看出， x % * % A % * % x表示的是二次型.

函数 crossprod(A,B) 表示的是 t(A) % * % B, 函数 tcrossprod(A,B) 表示的是 A % * % t(B).

6. 生成对角阵和矩阵取对角运算

函数 diag() 依赖于它的变量，当 v 是一个向量时， diag(v) 表示以 v 的元素为对角线元素的对角阵. 当 M 是一个矩阵时，则 diag(M) 表示的是取 M 对角线上的元素的向量. 如

```
> v<-c(1,4,5)
> diag(v)
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    4    0
[3,]    0    0    5
> M<-array(1:9,dim=c(3,3))
> diag(M)
[1] 1 5 9
```

7. 解线性方程组和求矩阵的逆矩阵

若求解线性方程组 $Ax = b$, 其命令形式为 `solve(A,b)`, 求矩阵 A 的逆, 其命令形式为 `solve(A)`. 设矩阵

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

则解方程组 $Ax = b$ 的解 x 和求矩阵 A 的逆矩阵 B 的命令如下

```
> A <- t(array(c(1:8, 10),dim=c(3,3)))
> b <- c(1,1,1)
> x <- solve(A,b); x
[1] -1.000000e+00  1.000000e+00 -4.728549e-16
> B <- solve(A); B
           [,1]      [,2] [,3]
[1,] -0.6666667 -1.333333    1
[2,] -0.6666667  3.666667   -2
[3,]  1.0000000 -2.000000    1
```

8. 求矩阵的特征值与特征向量

函数 `eigen(Sm)` 是求对称矩阵 Sm 的特征值与特征向量, 其命令形式为

```
> ev <- eigen(Sm)
```

则 `ev` 存放着对称矩阵 `Sm` 特征值和特征向量, 是由列表形式给出的 (有关列表的概念见 2.6 节), 其中 `ev$values` 是 `Sm` 的特征值构成的向量, `ev$vectors` 是 `Sm` 的特征向量构成的矩阵. 如

```
> Sm<-crossprod(A,A)
> ev<-eigen(Sm); ev
$values
[1] 303.19533618   0.76590739   0.03875643
$vectors
           [,1]          [,2]       [,3]
[1,] -0.4646675  0.833286355  0.2995295
```

```
[2,] -0.5537546 -0.009499485 -0.8326258
[3,] -0.6909703 -0.552759994  0.4658502
```

9. 矩阵的奇异值分解

函数 svd(A) 是对矩阵 A 作奇异值分解，即 $A = UDV^T$, 其中 U, V 是正交阵， D 为对角阵，也就是矩阵 A 的奇异值． svd(A) 的返回值也是列表，svd(A)$d 表示矩阵 A 的奇异值，即矩阵 D 的对角线上的元素． svd(A)$u 对应的是正交阵 U, svd(A)$v 对应的是正交阵 V. 例如，

```
> svdA<-svd(A); svdA
$d
[1] 17.4125052  0.8751614  0.1968665
$u
           [,1]        [,2]       [,3]
[1,] -0.2093373  0.96438514  0.1616762
[2,] -0.5038485  0.03532145 -0.8630696
[3,] -0.8380421 -0.26213299  0.4785099
$v
           [,1]         [,2]       [,3]
[1,] -0.4646675 -0.833286355  0.2995295
[2,] -0.5537546  0.009499485 -0.8326258
[3,] -0.6909703  0.552759994  0.4658502
> attach(svdA)
> u %*% diag(d) %*% t(v)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8   10
```

在上面的语句中， attach(svdA) 是说明下面的变量 u，v，d 是附属于 svdA 的，关于 attach() 函数的使用方法将在 2.6.2 节作详细介绍.

10. 求矩阵的行列式的值

函数 det(A) 是求矩阵 A 的行列式值。如

```
> det(A)
[1] -3
```

11. 最小拟合与 QR 分解

函数 lsfit() 的返回值是最小二乘拟合的结果，命令

```
> lsfit.sol <- lsfit(X, y)
```

给出最小二乘拟合结果，其中 y 是观测向量， X 是设计矩阵. 例如

x	0.0	0.2	0.4	0.6	0.8
y	0.9	1.9	2.8	3.3	4.2

作线性最小二乘拟合，其命令如下：

```
> x<-c(0.0, 0.2, 0.4, 0.6, 0.8)
> y<-c(0.9, 1.9, 2.8, 3.3, 4.2)
> lsfit.sol <- lsfit(x, y)
```

得到的计算结果是列表形式 (关于列表的概念将在 2.6 节讨论)

```
> lsfit.sol
$coefficients
Intercept         X
     1.02      4.00
$residuals
[1] -0.12  0.08  0.18 -0.12 -0.02
$intercept
[1] TRUE
$qr
$qt
[1] -5.85849810  2.52982213  0.23749843 -0.02946714  0.10356728
$qr
      Intercept          X
[1,] -2.2360680 -0.8944272
[2,]  0.4472136  0.6324555
[3,]  0.4472136 -0.1954395
```

```
[4,]  0.4472136 -0.5116673
[5,]  0.4472136 -0.8278950
$qraux
[1] 1.447214 1.120788
$rank
[1] 2
$pivot
[1] 1 2
$tol
[1] 1e-07
attr(,"class")
[1] "qr"
```

其中 $coefficients 是拟合系数, $residuals 是拟合残差, 其他参数我们先不作解释, 大家可看在线帮助.

与 lsfit() 函数有密切关系的函数是 ls.diag(), 它给出拟合的进一步的统计信息.

另一个最小二乘拟全有密切关系的函数是 QR 分解函数 qr(), 和它的同类函数, 有如下函数 qr(), qr.coef(), qr.fitted() 和 qr.resid(). 为了进一步理解这些命令, 还看上面的例子

```
> X<-matrix(c(rep(1,5), x), ncol=2)
> Xplus <- qr(X); Xplus
$qr
           [,1]       [,2]
[1,] -2.2360680 -0.8944272
[2,]  0.4472136  0.6324555
[3,]  0.4472136 -0.1954395
[4,]  0.4472136 -0.5116673
[5,]  0.4472136 -0.8278950
$rank
[1] 2
$qraux
```

```
[1] 1.447214 1.120788
$pivot
[1] 1 2
attr(,"class")
[1] "qr"
```

QR 分解函数 qr() 输入的设计矩阵需要加以 1 为元素的列，其返回值是列表，其中 $qr 矩阵的上三角阵是 QR 分解中得到的 R 矩阵，下三角阵是 QR 分解得到的正交阵 Q 的部分信息， $qraux 是 Q 的附加信息. 注意，这两个参数的结果与函数 lsfit() 得到的结果是相同的.

可用 QR 分解得到的结果计算最小二乘的系数

```
> b <- qr.coef(Xplus, y); b
[1] 1.02 4.00
```

得到的系数与函数 lsfit() 也是相同的，但为什么用这种方法计算呢？这是因为用 QR 分解在计算最小二乘拟合时，其计算误差比一般方法要小.

类似地，可以通过 QR 分解得到最小二乘的拟合值和残差值.

```
> fit <- qr.fitted(Xplus, y); fit
[1] 1.02 1.82 2.62 3.42 4.22
> res <- qr.resid(Xplus, y); res
[1] -0.12  0.08  0.18 -0.12 -0.02
```

2.5.5 与矩阵 (数组) 运算有关的函数

1. 取矩阵的维数

函数 dim(A) 得到矩阵 A 的维数，函数 nrow(A) 得到矩阵 A 的行数，函数 ncol(A) 得到矩阵 A 的列数. 如

```
> A<-matrix(1:6,nrow=2); A
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> dim(A)
[1] 2 3
```

```
> nrow(A)
[1] 2
> ncol(A)
[1] 3
```

2. 矩阵的合并

函数 cbind() 把其自变量横向拼成一个大矩阵， rbind() 把其自变量纵向拼成一个大矩阵. cbind() 的自变量是矩阵或看作列向量的向量时，自变量的高度应该相等. rbind() 的自变量是矩阵或看作行向量的向量时，自变量的宽度应该相等. 如果参与合并的自变量比其变量短，则循环补足后合并. 如

```
> x1 <- rbind(c(1,2), c(3,4)); x1
     [,1] [,2]
[1,]    1    2
[2,]    3    4
> x2 <- 10+x1
> x3 <- cbind(x1, x2); x3
     [,1] [,2] [,3] [,4]
[1,]    1    2   11   12
[2,]    3    4   13   14
> x4 <- rbind(x1, x2); x4
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]   11   12
[4,]   13   14
> cbind(1, x1)
     [,1] [,2] [,3]
[1,]    1    1    2
[2,]    1    3    4
```

3. 矩阵的拉直

设 A 是一个矩阵，则函数 as.vector(A) 就可以将矩阵转化为向量. 如

```
> A<-matrix(1:6,nrow=2); A
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> as.vector(A)
[1] 1 2 3 4 5 6
```

4. 数组的维名字

数组可以有一个属性 dimnames 保存各维的各个下标的名字, 缺省时为 NULL. 如

```
> X <- matrix(1:6, ncol=2,
  dimnames=list(c("one","two","three"), c("First","Second")),
  byrow=T); X
      First Second
one       1      2
two       3      4
three     5      6
```

也可以先定义矩阵 X 然后再为 dimnames(X) 赋值. 例如,

```
> X<-matrix(1:6, ncol=2, byrow=T)
> dimnames(X) <- list(
        c("one", "two", "three"), c("First", "Second"))
```

对于矩阵, 还可以使用属性 rownames 和 colnames 来访问行名与列名. 例如,

```
> X<-matrix(1:6, ncol=2, byrow=T)
> colnames(X) <- c("First", "Second")
> rownames(X) <- c("one", "two", "three")
```

5. 数组的广义转置

可以用 aperm(A, perm) 函数把数组 A 的各维按 perm 中指定的新次序重新排列. 例如,

```
> A<-array(1:24, dim = c(2,3,4))
> B<-aperm(A, c(2,3,1))
```

结果 B 把 A 的第 2 维移到了第 1 维， A 的第 3 维移到了第 2 维， A 的第 1 维移到了第三维. 这时有 B[i,j,k]=A[j,k,i].

对于矩阵 A, aperm(A, c(2,1)) 恰好是矩阵转置，即 t(A).

6. apply 函数

对于向量，可以用 sum 、 mean 等函数对其进行计算. 对于数组 (矩阵)，如果想对其一维 (或若干维) 进行某种计算，可用 apply 函数，其一般形式为

```
apply(A, MARGIN, FUN, ...)
```

其中 A 为一个数组， MARGIN 是固定哪些维不变， FUN 是用来计算的函数. 如

```
> A<-matrix(1:6,nrow=2); A
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> apply(A,1,sum)
[1]  9 12
> apply(A,2,mean)
[1] 1.5 3.5 5.5
```

2.6 列表与数据框

2.6.1 列表 (list)

1. 列表的构造

列表是一种特别的对象集合，它的元素也由序号 (下标) 区分，但是各元素的类型可以是任意对象，不同元素不必是同一类型. 元素本身允许是其它复杂数据类型，比如，列表的一个元素也允许是列表. 下面是如何构造列表的例子.

```
> Lst <- list(name="Fred", wife="Mary", no.children=3,
              child.ages=c(4,7,9))
> Lst
$name
[1] "Fred"
```

```
$wife
[1] "Mary"
$no.children
[1] 3
$child.ages
[1] 4 7 9
```

列表元素总可以用 “列表名 [[下标]]” 的格式引用. 例如,

```
> Lst[[2]]
[1] "Mary"
> Lst[[4]][2]
[1] 7
```

但是, 列表不同于向量, 我们每次只能引用一个元素, 如 `Lst[[1:2]]` 的用法是不允许的.

注意: “列表名 [下标]” 或 “列表名 [下标范围]” 的用法也是合法的, 但其意义与用两重括号的记法完全不同, 两重记号取出列表的一个元素, 结果与该元素类型相同, 如果使用一重括号, 则结果是列表的一个子列表 (结果类型仍为列表).

在定义列表时如果指定了元素的名字 (如 `Lst` 中的 `name`, `wife`, `no.children`, `child.ages`), 则引用列表元素还可以用它的名字作为下标, 格式为 “列表名 [[" 元素名 "]]”, 如

```
> Lst[["name"]]
[1] "Fred"
> Lst[["child.age"]]
[1] 4 7 9
```

另一种格式是 “列表名 $ 元素名”, 如

```
> Lst$name
[1] "Fred"
> Lst$wife
[1] "Mary"
> Lst$child.ages
[1] 4 7 9
```

构造列表的一般格式为

```
Lst <- list(name_1=object_1, ..., name_m=object_m)
```

其中 name 是列表元素的名称， object 是列表元素的对象.

2. 列表的修改

列表的元素可以修改，只要把元素引用赋值即可，如将 Fred 改成 John.

```
> Lst$name <- "John"
```

如果需要增加一项家庭收入，夫妻的收入分别是 1980 和 1600, 则输入

```
> Lst$income <- c(1980, 1600)
```

如果要删除列表的某一项，则将该项赋空值 (NULL).

几个列表可以用连接函数 c() 连接起来, 结果仍为一个列表, 其元素为各自变量的列表元素. 如

```
>list.ABC <- c(list.A, list.B, list.C)
```

3. 返回值为列表的函数

在 R 中, 有许多函数的返回值是列表, 如求特征值特征向量的函数 eigen(), 奇异值分解函数 svd() 和最小二乘函数数 lsfit() 等，这里不再一一讨论，在用到时再讨论相关函数的意义.

2.6.2 数据框 (data.frame)

数据框是 R 的一种数据结构. 它通常是矩阵形式的数据，但矩阵各列可以是不同类型的. 数据框每列是一个变量，每行是一个观测.

但是, 数据框有更一般的定义. 它是一种特殊的列表对象, 有一个值为 "data .frame" 的 class 属性, 各列表成员必须是向量 (数值型、字符型、逻辑型) 、因子、数值型矩、列表，或其它数据框. 向量、因子成员为数据框提供一个变量，如果向量非数值型会被强制转换为因子，而矩阵、列表、数据框这样的成员为新数据框提供了和其列数、成员数、变量数相同个数的变量. 作为数据框变量的向量、因子或矩阵必须具有相同的长度 (行数).

尽管如此，一般还是可以把数据框看作是一种推广了的矩阵，它可以用矩阵形式显示，可以用对矩阵的下标引用方法来引用其元素或子集.

1. 数据框的生成

数据框可以用 data.frame() 函数生成，其用法与 list() 函数相同，各自变量变成数据框的成分，自变量可以命名，成为变量名．例如

```
> df<-data.frame(
    Name=c("Alice", "Becka", "James", "Jeffrey", "John"),
    Sex=c("F", "F", "M", "M", "M"),
    Age=c(13, 13, 12, 13, 12),
    Height=c(56.5, 65.3, 57.3, 62.5, 59.0),
    Weight=c(84.0, 98.0, 83.0, 84.0, 99.5)
  ); df
     Name Sex Age Height Weight
1   Alice   F  13   56.5   84.0
2   Becka   F  13   65.3   98.0
3   James   M  12   57.3   83.0
4 Jeffrey   M  13   62.5   84.0
5    John   M  12   59.0   99.5
```

如果一个列表的各个成分满足数据框成分的要求，它可以用 as.data.frame() 函数强制转换为数据框．比如，

```
> Lst<-list(
    Name=c("Alice", "Becka", "James", "Jeffrey", "John"),
    Sex=c("F", "F", "M", "M", "M"),
    Age=c(13, 13, 12, 13, 12),
    Height=c(56.5, 65.3, 57.3, 62.5, 59.0),
    Weight=c(84.0, 98.0, 83.0, 84.0, 99.5)
  ); Lst
$Name
[1] "Alice"   "Becka"   "James"   "Jeffrey" "John"
$Sex
[1] "F" "F" "M" "M" "M"
$Age
[1] 13 13 12 13 12
$Height
```

```
[1] 56.5 65.3 57.3 62.5 59.0
$Weight
[1] 84.0 98.0 83.0 84.0 99.5
```

则 as.data.frame(Lst) 是与 df 相同的数据框.

一个矩阵可以用 data.frame() 转换为一个数据框, 如果它原来有列名则其列名被作为数据框的变量名; 否则系统自动为矩阵的各列起一个变量名. 如

```
> X <- array(1:6, c(2,3))
> data.frame(X)
  X1 X2 X3
1  1  3  5
2  2  4  6
```

2. 数据框的引用

引用数据框元素的方法与引用矩阵元素的方法相同, 可以使用下标或下标向量, 也可以使用名字或名字向量. 如

```
> df[1:2, 3:5]
  Age Height Weight
1  13   56.5     84
2  13   65.3     98
```

数据框的各变量也可以用按列表引用 (即用双括号 [[ ]] 或 $ 符号引用). 如

```
> df[["Height"]]
[1] 56.5 65.3 57.3 62.5 59.0
> df$Weight
[1] 84.0 98.0 83.0 84.0 99.5
```

数据框的变量名由属性 names 定义, 此属性一定是非空的. 数据框的各行也可以定义名字, 可以用 rownames 属性定义. 如

```
> names(df)
[1] "Name"   "Sex"    "Age"    "Height" "Weight"
> rownames(df)<-c("one", "two", "three", "four", "five")
> df
          Name Sex Age Height Weight
```

```
one     Alice   F  13   56.5   84.0
two     Becka   F  13   65.3   98.0
three   James   M  12   57.3   83.0
four  Jeffrey   M  13   62.5   84.0
five     John   M  12   59.0   99.5
```

3. attach() 函数

数据框的主要用途是保存统计建模的数据. R 的统计建模功能都需要以数据框为输入数据. 我们也可以把数据框当成一种矩阵来处理. 在使用数据框的变量时可以用 "数据框名 $ 变量名" 的记法. 但是, 这样使用较麻烦, R 提供了 attach() 函数可以把数据框中的变量 "连接" 到内存中, 这样便于数据框数据的调用. 例如,

```
> attach(df)
> r <- Height/Weight; r
[1] 0.6726190 0.6663265 0.6903614 0.7440476 0.5929648
```

后一语句将在当前工作空间建立一个新变量 r, 它不会自动进入数据框 df 中, 要把新变量赋值到数据框中, 可以用

```
> df$r <- Height/Weight
```

这样的格式.

为了取消连接, 只要调用 detach()(无参数即可).

注意: R 中名字空间的管理是比较独特的. 它在运行时保持一个变量搜索路径表, 在读取某个变量时到这个变量搜索路径表中由前向后查找, 找到最前的一个; 在赋值时总是在位置 1 赋值 (除非特别指定在其它位置赋值). attach() 的缺省位置是在变量搜索路径表的位置 2, detach() 缺省也是去掉位置 2. 所以, R 编程的一个常见问题是当你误用了一个自己并没有赋值的变量时有可能不出错, 因为这个变量已在搜索路径中某个位置有定义, 这样不利于程序的调试, 需要留心这样的问题.

attach() 除了可以连接数据框, 也可以连接列表.

2.6.3 列表与数据框的编辑

如果需要对列表或数据框中的数据进行编辑, 也可调用函数 edit() 进行编辑、修改, 其命令格式为

```
> xnew <- edit(xold)
```

其中 xold 是原列表或数据框图， xnew 是修改后的列表或数据框. 注意：原数据 xold 并没有改动，改动的数据存放在 xnew 中.

函数 edit() 也可以对向量，数组或矩阵类型的数据进行修改或编辑.

2.7 读、写数据文件

在应用统计学中，数据量一般是比较大的，变量也很多. 如果用上述方法来建立数据集，是不可取的. 上述方法适用于少量数据、少量变量的分析. 对于大量数据和变量，一般应在其他软件中输入 (或数据来源是其他软件的输出结果)，再读到 R 中处理. R 软件有多种读数据文件的方法.

另外，所有的计算结果也不应只在屏幕上输出，应当保存在文件中，以备使用.

这里介绍一些 R 软件读、写数据文件的方法.

2.7.1 读纯文本文件

读纯文本文件有两个函数，一个是 read.table() 函数，另一个是 scan() 函数.

1. read.table( ) 函数

read.table() 函数是读表格形式的文件. 若 "住宅" 数据已经输入一个纯文本文件 "houses.data" 中，其格式如下：

```
    Price   Floor  Area  Rooms   Age   Cent.heat
01  52.00  111.0    830    5     6.2      no
02  54.75  128.0    710    5     7.5      no
03  57.50  101.0   1000    5     4.2      no
04  57.50  131.0    690    6     8.8      no
05  59.75   93.0    900    5     1.9     yes
```

其中第一行为变量名，第一列为记录序号.

利用 read.table() 函数可读入数据，如

```
> rt <- read.table("houses.data")
```

此时变量 rt 是一个数据框，其形式与纯文本文件 "houses.data" 格式相同. 我们对它进行测试，得到

```
> is.data.frame(rt)
[1] TRUE
```

如果数据文件中没有第一列记录序号，如

Price	Floor	Area	Rooms	Age	Cent.heat
52.00	111.0	830	5	6.2	no
54.75	128.0	710	5	7.5	no
57.50	101.0	1000	5	4.2	no
57.50	131.0	690	6	8.8	no
59.75	93.0	900	5	1.9	yes

则相应的命令改为

```
> rt <- read.table("houses.data", header=TRUE)
```

在 rt 会自动加上记录序号.

read.table() 的使用格式为

```
read.table(file, header = FALSE, sep = "", quote = "\"'",
           dec = ".", row.names, col.names, as.is = FALSE,
           na.strings = "NA", colClasses = NA, nrows = -1,
           skip = 0, check.names = TRUE,
           fill = !blank.lines.skip, strip.white = FALSE,
           blank.lines.skip = TRUE, comment.char = "#")
```

其中 file 是读入数据的文件名. header=TRUE 表示所读数据的第一行为变量名；否则 (缺省值) 第一行作为数据. sep 是数据分隔的字符，通常用空格作为分隔符. skip 表示读数据时跳过的行数. 其他参数的用法请见帮助.

2. scan( ) 函数

scan() 函数可以直接读纯文本文件数据. 例如，有 15 名学生的体重数据已经输入一个纯文本文件 "weight.data" 中，其格式如下：

```
75.0  64.0  47.4  66.9  62.2  62.2  58.7  63.5
66.6  64.0  57.0  69.0  56.9  50.0  72.0
```

则

```
w <- scan("weight.data")
```

将文件中的 15 个数据读入，并赋给向量 w.

假设数据中有不同的属性，如下面

```
172.4  75.0  169.3  54.8  169.3  64.0  171.4  64.8  166.5  47.4
171.4  62.2  168.2  66.9  165.1  52.0  168.8  62.2  167.8  65.0
165.8  62.2  167.8  65.0  164.4  58.7  169.9  57.5  164.9  63.5
 ...    ...    ...   ...    ...   ...    ...   ...    ...   ...
```

是 100 名学生的身高和体重的数据，放在纯文本数据文件 "h_w.data", 其中第 1 、 3 、 5 、 7 、 9 列是身高 (cm), 第 2 、 4 、 6 、 8 、 10 列是体重 (kg), 则

```
> inp <- scan("h_w.data", list(height=0, weight=0))
```

将数据读入，并以列表的方式赋给变量 inp.

```
> is.list(inp)
[1] TRUE
```

可以将由 scan() 读入的数据存放成矩阵形式. 如果将 “weight.data” 中的体重数据放在一个 3 行 5 列的矩阵中，而且数据按行放置. 其命令格式为

```
> X <- matrix(scan("weight.data", 0),
              nrow=3, ncol=5, byrow=TRUE)
Read 15 items
> X
     [,1] [,2] [,3] [,4] [,5]
[1,] 75.0 64.0 47.4 66.9 62.2
[2,] 62.2 58.7 63.5 66.6 64.0
[3,] 57.0 69.0 56.9 50.0 72.0
```

由前面讲到的函数 matrix() 的用法，下面两种写法是等价的。

```
> X <- matrix(scan("input.dat", 0), ncol=5, byrow=TRUE)
> X <- matrix(scan("input.dat", 0), nrow=3, byrow=TRUE)
```

也可以用 scan() 函数直接从屏幕上输数据。如

```
> x<-scan()
  1: 1 3 5 7 9
  6:
```

```
  Read 5 items
> x
[1] 1 3 5 7 9
```

scan() 读文件的一般格式为

```
scan(file = "", what = double(0), nmax = -1,
     n = -1, sep = "",
     quote = if(identical(sep, "\n")) "" else "'\"",
     dec = ".", skip = 0, nlines = 0, na.strings = "NA",
     flush = FALSE, fill = FALSE, strip.white = FALSE,
     quiet = FALSE, blank.lines.skip = TRUE,
     multi.line = TRUE, comment.char = "",
     allowEscapes = TRUE)
```

其中 file 为文件名. what 为指定一个列表, 则列表每项的类型为需要读取的类型. skip 控制可以跳过文件的开始不读行数. sep 控制可以指定数据间的分隔符. 其它参数见帮助文件.

2.7.2 读其它格式的数据文件

R 软件除了可以读纯文本文件外, 还可以读其他统计软件格式的数据, 如 Minitab 、 S-Plus 、 SAS 、 SPSS 等. 要读入其他格式数据库, 必须先调入 "foreign" 模块. 它不属于 R 的内在模块, 需要在使用前调入. 调入的方法很简便, 只需键入命令:

```
> library(foreign)
```

或用 2.1.3 节介绍的载入程序包调入.

1. 读 SPSS 、 SAS 、 S-PLUS 、 Stata 数据文件

已知数据由表 2.2 所示. 分别存成 SPSS 数据文件 ("educ_scores.sav") 、SAS 数据文件 ("educ_scores .xpt") 、 S-Plus 数据文件 ("educ_scores") 和 Stata 数据文件 ("educ_scores .dta").

读 SPSS 文件的格式是:

```
> rs <- read.spss("educ_scores.sav")
```

其变量 rs 是一个列表, 如果打算形成数据框, 则命令格式为

表 2.2: 某学院学生数据

Student	Language Aptiude (x_1)	Analogical Reasoning (x_2)	Geometric Reasoning (x_3)	Sex of student (Male = 1) (x_4)
A	2	3	15	1
B	6	8	9	1
C	5	2	7	0
D	9	4	3	1
E	11	10	2	0
F	12	15	1	0
G	1	4	12	1
H	7	3	4	0

```
> rs<-read.spss("educ_scores.sav", to.data.frame=TRUE)
```

读 SAS 文件的格式是：

```
> rx <- read.xport("educ_scores.xpt")
```

其变量 `rx` 是一个数据框.

读 S-PLUS 文件的格式是：

```
> rs <- read.S("educ_scores")
```

其变量 `rs` 是一个数据框.

读 Stata 文件的格式是：

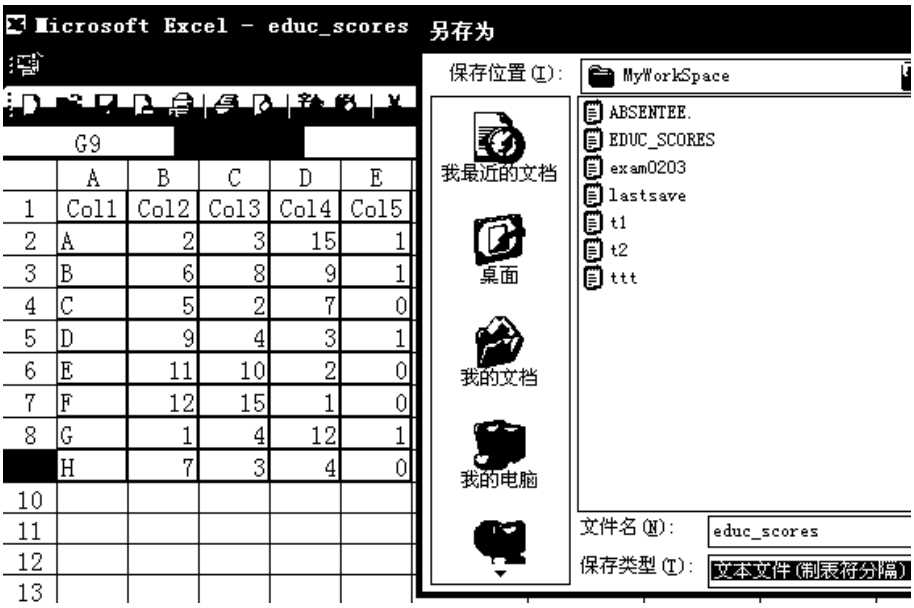
```
> rd <- read.dta("educ_scores.dta")
```

其变量 `rd` 是一个数据框.

2. 读 Excel 数据文件

将上述数据存为 Excel 表 (`"educ_scores.xls"`), 但 R 软件无法直接读 Excel 表，需要将 Excel 表进入转化成其他格式，然后才能被 R 软件读出.

第一种转化格式是将 Excel 表转化成 “文本文件 (制表符分隔)” 文件，如图 2.18 所示.



	A	B	C	D	E
1	Col1	Col2	Col3	Col4	Col5
2	A	2	3	15	1
3	B	6	8	9	1
4	C	5	2	7	0
5	D	9	4	3	1
6	E	11	10	2	0
7	F	12	15	1	0
8	G	1	4	12	1
	H	7	3	4	0
10					
11					
12					
13					

图 2.18: 将 Excel 表存为文本文件

用函数 `read.delim()` 读该文本文件，即

```
> rd <- read.delim("educ_scores.txt")
```

得到的变量 `rd` 是一个数据框.

第二种转化格式是将 Excel 表转化成 “CSV(逗号分隔)” 文件，如图 2.19 所示.

用函数 `read.csv()` 读该文本文件，即

```
> rc <- read.csv("educ_scores.csv")
```

得到的变量 `rc` 是一个数据框.

2.7.3 链接嵌入的数据库

R 软件中提供了 50 多个数据库和其他可利用的软件包，可以用 `data()` 函数调用这些数据库与软件包. 用

```
> data()
```

命令，列出在基本软件包 (base) 所有可利用的数据集. 如果装载某一个数据集，只需在括号中加入相应的名字. 如

```
> data(infert)
```

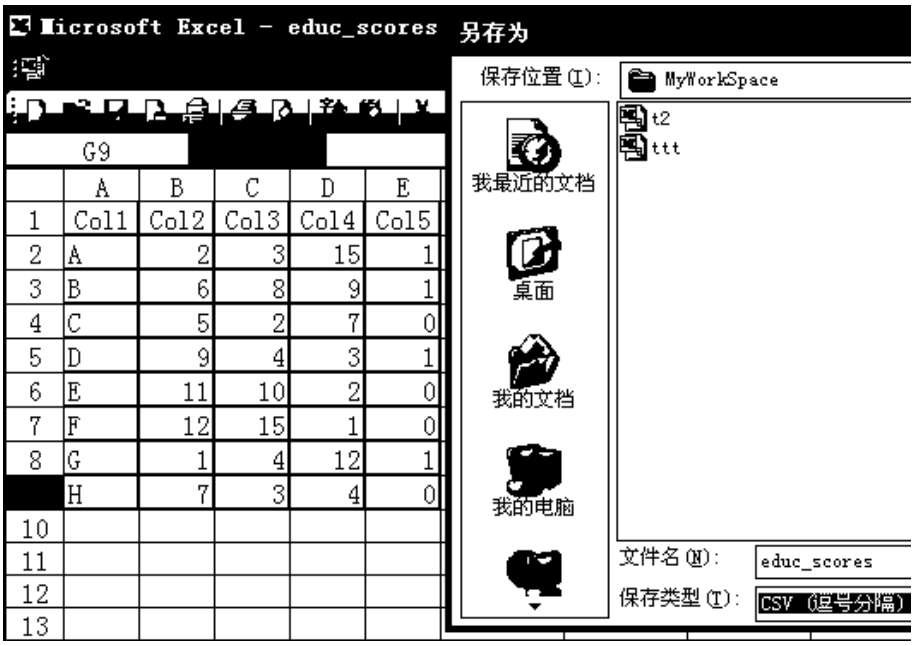


图 2.19: 将 Excel 表存为 CSV 文件

如果需要从其他的软件包链接数据，可以使用参数 package, 例如,

```
> data(package="nls")
> data(Puromycin,package="nls")
```

如果一个软件包已被 library 附加在库中，则这个数据库将自动地被包含在其中，如

```
> library(nls)
> data()
> data(Puromycin)
```

在 data() 中，除包含基本软件包 (base) 还包含 nls 软件包.

2.7.4 写数据文件

1. write( ) 函数

write() 函数写数据文件的格式是

```
write(x, file = "data",
      ncolumns = if(is.character(x)) 1 else 5,
```

```
            append = FALSE)
```

其中 x 是数据，通常是矩阵，也可以是向量. file 是文件名 (缺省时文件名为 "data"). append=TRUE 时，在原文件上填加数据；否则 (FALSE, 缺省值) 写一个新文件. 其它参数见帮助文件.

2. write.table( ) 函数和 write.csv( ) 函数

对于列表数据或数据框数据，可以用 write.table() 函数或 write.csv() 函数写纯文本格式的数据文件，或 CSV 格式的 Excel 数据文件，例如,

```
> df <- data.frame(
    Name=c("Alice", "Becka", "James", "Jeffrey", "John"),
    Sex=c("F", "F", "M", "M", "M"),
    Age=c(13, 13, 12, 13, 12),
    Height=c(56.5, 65.3, 57.3, 62.5, 59.0),
    Weight=c(84.0, 98.0, 83.0, 84.0, 99.5)
  )
> write.table(df, file="foo.txt")
> write.csv(df, file="foo.csv")
```

write.table() 函数和 write.csv() 函数的使用格式为

```
write.table(x, file = "", append = FALSE, quote = TRUE,
            sep = " ", eol = "\n", na = "NA", dec = ".",
            row.names = TRUE, col.names = TRUE,
            qmethod = c("escape", "double"))

write.csv(..., col.names = NA, sep = ",",
          qmethod = "double")
```

其中 x 是对象. file 是文件名. append=TRUE 时，在原文件上填加数据；否则 (FALSE, 缺省值) 写一个新文件. sep 是数据间隔字符. 其它参数见帮助文件.

2.8 控制流

R 是一个表达式语言，其任何一个语句都可以看成是一个表达式. 表达式之间以分号分隔或用换行分隔. 表达式可以续行，只要前一行不是完整表达式 (比

如末尾是加减乘除等运算符，或有未配对的括号) 则下一行为上一行的继续.

若干个表达式可以放在一起组成一个复合表达式，作为一个表达式使用. 组合用花括号 “ { }” 表示.

R 语言也提供了其它高级程序语言共有的分支、循环等程序控制结构.

2.8.1 分支语句

分支语句有 if / else 语句、 switch 语句.

1. if / else 语句

if / else 语句是分支语句中主要的语句， if / else 语句的格式为

```
if(cond) statement_1
if(cond) statement_1  else  statement_2
```

第一句的意义是：如果条件 cond 成立，则执行表达式 statement_1; 否则跳过.
第二句的意义是：如果条件 cond 成立，则执行表达式 statement_1; 否则执行表达式 statement_2.

例如,

```
if( any(x <= 0) ) y <- log(1+x) else y <- log(x)
```

注意：此命令与下面的命令

```
y <- if( any(x <= 0) ) log(1+x) else log(x)
```

等价.

对于 if / else 语句，还有下面的用法

```
if ( cond_1 )
    statement_1
else if ( cond_2 )
    statement_2
else if ( cond_3 )
    statement_3
else
    statement_4
```

2. switch 语句

switch 语句是多分支语句，其使用方法是：

```
switch (statement, list)
```

其中 statement 是表达式， list 是列表，可以用有名定义. 如果表达式的返回值在 1 到 length(list), 则返回列表相应位置的值；否则返回 “NULL” 值. 例如,

```
> x <- 3
> switch(x, 2+2, mean(1:10), rnorm(4))
[1]  0.8927328 -0.7827752  1.0772888  1.0632371
> switch(2, 2+2, mean(1:10), rnorm(4))
[1] 5.5
> switch(6, 2+2, mean(1:10), rnorm(4))
NULL
```

当 list 是有名定义时， statement 等于变量名时，返回变量名对应的值；否则返回 “NULL” 值. 例如,

```
> y <- "fruit"
> switch(y,fruit="banana",vegetable="broccoli",meat="beef")
[1] "banana"
```

2.8.2 中止语句与空语句

中止语句是 break 语句， break 语句的作用是中止循环，使程序跳到循环以外. 空语句是 next 语句， next 语句是继续执行，而不执行某个实质性的内容. 关于 break 语句和 next 语句的例子，将结合循环语句来说明.

2.8.3 循环语句

循环语句有 for 循环、 while 循环和 repeat 循环语句.

1. for 循环语句

for 循环的格式为

```
> for (name in expr_1) expr_2
```

其中 name 是循环变量， expr_1 是一个向量表达式 (通常是个序列，如 1:20), expr_2 通常是一组表达式.

如构造一个 4 阶的 Hilbert 矩阵,

```
> n<-4; x<-array(0, dim=c(n,n))
> for (i in 1:n){
     for (j in 1:n){
        x[i,j]<-1/(i+j-1)
     }
  }
> x
          [,1]      [,2]      [,3]      [,4]
[1,] 1.0000000 0.5000000 0.3333333 0.2500000
[2,] 0.5000000 0.3333333 0.2500000 0.2000000
[3,] 0.3333333 0.2500000 0.2000000 0.1666667
[4,] 0.2500000 0.2000000 0.1666667 0.1428571
```

2. while 循环语句

while 循环语句 while 语句的格式为

```
> while (condition) expr
```

当条件 condition 成立, 则执行表达式 expr. 例如, 编写一个计算 1000 以内的 Fibonacci 数.

```
> f<-1; f[2]<-1; i<-1
> while (f[i]+f[i+1]<1000) {
     f[i+2]<-f[i]+f[i+1]
     i<-i+1;
  }
> f
[1]   1   1   2   3   5   8  13  21  34  55  89 144
[13] 233 377 610 987
```

3. repeat 循环语句

repeat 语句的格式为

```
> repeat expr
```

repeat 循环依赖 break 语句跳出循环. 例如, 用 repeat 循环编写一个计算 1000 以内的 Fibonacci 数的程序.

```
> f<-1; f[2]<-1; i<-1
> repeat {
     f[i+2]<-f[i]+f[i+1]
     i<-i+1
     if (f[i]+f[i+1]>=1000) break
  }
```

或将条件语句改为 `if (f[i]+f[i+1]<1000) next else break`, 也有同样的计算结果.

2.9 编写自己的函数

R 软件允许用户自己创建模型的目标函数. 有许多 R 函数存贮为特殊的内部形式, 并可以被进一步的调用. 这样在使用时可以使语言更有力、更方便, 而且程序也更美观. 学习写自己的程序是你学习使用 R 语言的主要方法之一.

事实上, R 系统提供的绝大多数函数, 如 `mean()`, `var()`, `postscript()` 等, 是系统编写人员写在 R 语言中的函数, 与你自己写的函数本质上没有多大差别.

函数定义的格式如下,

```
> name <- function(arg_1, arg_2, ...) expression
```

expression 是 R 中的表达式 (通常是一组表达式), `arg_1, arg_2, ...` 表示函数的参数. 表达式中, 放在程序最后的信息是函数的返回值, 返回值可以是向量、数组 (矩阵) 、列表或数据框.

调用函数的格式为 `name(expr_1, expr_2, ...)`, 并且在任何时调用都是合法的.

在调用自己编写的函数 (程序) 时, 需要将已写好的函数调到内存中, 即使用 2.1.3 节介绍的 "输入 R 代码..." 命令, 执行 `source()` 函数. 关于函数的调用, 后面的各章还会有介绍.

2.9.1 简单的例子

与其他程序一样, R 可以很容易地编写自己需要的函数.

例 2.4 编写一个用二分法求非线性方程根的函数，并求方程

$$x^3 - x - 1 = 0$$

在区间 $[1, 2]$ 内的根，精度要求 $\varepsilon = 10^{-6}$.

解：取初始区间 $[a, b]$, 当 $f(a)$ 与 $f(b)$ 异号，作二分法计算；否则停止计算（输出计算失败信息）.

二分法计算过程如下：取中点 $x = \frac{a+b}{2}$, 若 $f(a)$ 与 $f(x)$ 异号，则置 $b = x$; 否则 $a = x$. 当区间长度小于指定要求时，停止计算.

编写二分法程序，程序名： `bisect.R`.

```
fzero <- function(f, a, b, eps=1e-5){
   if (f(a)*f(b)>0)
      list(fail="finding root is fail!")
   else{
      repeat {
         if (abs(b-a)<eps) break
         x <- (a+b)/2
         if (f(a)*f(x)<0) b<-x  else  a<-x
      }
      list(root=(a+b)/2, fun=f(x))
   }
}
```

在二分法求根的函数 (程序) 中，输入值 `f` 是求根的函数， `a`，`b` 是二分法的左右端点． `eps=1e-5` 是精度要求，是有名参数 (后面将介绍). 函数 (程序) 的返回值是列表，当初始区间不满足要求时，返回值为 "finding root is fail!"(求根失败); 当满足终止条件时，返回值为方程根的近似值和在近似点处的函数值.

建立求根的非线性函数

```
f<-function(x) x^3-x-1
```

求它在区间 $[1, 2]$ 内的根.

```
> fzero(f, 1, 2, 1e-6)
$root
[1] 1.324718
```

```
$fun
[1] -1.857576e-06
```

事实上，大家不用编写求根函数， R 软件已提供了求一元方程根的函数 uniroot(), 其使用格式为

```
uniroot(f, interval,
   lower = min(interval), upper = max(interval),
   tol = .Machine$double.eps^0.25, maxiter = 1000, ...)
```

例如，要求例 2.4 的根，只需输入命令

```
> uniroot(f, c(1,2))
```

就可得到

```
$root
[1] 1.324718
$f.root
[1] -5.634261e-07
$iter
[1] 7
$estim.prec
[1] 6.103516e-05
```

其计算结果与我们编写的程序的计算结果是相同的.

下面编写一个与统计有关的函数 — 计算两样本的 T 统计量.

例 2.5 已知两样本

```
A:  79.98 80.04 80.02 80.04 80.03 80.03 80.04 79.97
    80.05 80.03 80.02 80.00 80.02
B:  80.02 79.94 79.98 79.97 79.97 80.03 79.95 79.97
```

计算两样本的 T 统计量.

解: 当两样本的方差相同，且未知，则 T 统计量的计算公式为

$$T = \frac{(\overline{X} - \overline{Y})}{S\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}, \tag{2.1}$$

其中

$$S^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}, \tag{2.2}$$

$\overline{X}, \overline{Y}$ 分别是两组数据的样本均值， S_1^2, S_2^2 分别是两组数据的样本方差， n_1, n_2 分别为两组数据的个数.

按照式 (2.1) 和 (2.2) 编写相应的程序 (程序名： twosam.R)

```
twosam <- function(y1, y2) {
   n1 <- length(y1); n2 <- length(y2)
   yb1 <- mean(y1); yb2 <- mean(y2)
   s1 <- var(y1); s2 <- var(y2)
   s <- ((n1-1)*s1 + (n2-1)*s2)/(n1+n2-2)
   (yb1 - yb2)/sqrt(s*(1/n1 + 1/n2))
}
```

在函数 (程序) 中，输入值 y1，y2 是需要计算 T 统计量的两组数据. 函数 (程序) 的返回值是数值型变量，给出相应的 T 统计量.

输入数据 A, B, 并计算 T 统计量.

```
> A <- c(79.98, 80.04, 80.02, 80.04, 80.03, 80.03,
    80.04, 79.97, 80.05, 80.03, 80.02, 80.00, 80.02)
> B <- c(80.02, 79.94, 79.98, 79.97, 79.97, 80.03,
    79.95, 79.97)
> twosam(A,B)
[1] 3.472245
```

在后面我们还会讲到，用 T 统计量来估计两样本均值是否相同.

2.9.2 定义新的二元运算

R 软件可以定义的二元运算，其形式为 %anything%. 设 x, y 是两个向量，定义 x 与 y 的内积

$$\langle x, y\rangle = \exp(-\|x-y\|^2/2),$$

其运算符号用 %!% 表示，则二元运算的定义如下

```
"%!%" <- function(x, y) {exp(-0.5*(x-y) %*% (x-y))}
```

2.9.3 有名参数与省缺

如果用这种形式 “ name=object ” 给出被调用函数中的参数，则这些参数可以按照任何顺序给出. 如定义如下函数

```
> fun1 <- function(data, data.frame, graph, limit) {
    [function body omitted]
  }
```

则下面的三种调用方法

```
> ans <- fun1(d, df, TRUE, 20)
> ans <- fun1(d, df, graph=TRUE, limit=20)
> ans <- fun1(data=d, limit=20, graph=TRUE, data.frame=df)
```

都是等价的.

如果在例 2.4 中，其精度要求取 `1e-5`(10^{-5}), 则不必输入精度要求，直接输入区间端点即可.

```
> fzero(1,2)
$root
[1] 1.324718
$fun
[1] -1.405875e-05
```

下面利用有名参数的方法编写一个求非线性方程组根的 Newton 法的程序.

例 2.6 编写求非线性方程组解的 *Newton* 法的程序，并用此程序求解非线性方程组

$$\begin{cases} x_1^2 + x_2^2 - 5 = 0 \\ (x_1 + 1)x_2 - (3x_1 + 1) = 0 \end{cases}$$

的解，取初始点 $x^{(0)} = (0, 1)^T$, 精度要求 $\varepsilon = 10^{-5}$.

解: 求解非线性方程组

$$f(x) = 0, \quad f : R^n \to R^n \in C^1$$

的 Newton 法的迭代格式为

$$x^{(k+1)} = x^{(k)} - [J(x^{(k)})]^{-1} f(x^{(k)}), \quad k = 0, 1, \cdots,$$

其中 $J(x)$ 为函数 $f(x)$ 的 Jacobi 矩阵，即

$$J(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}.$$

因此，相应的程序 (程序名： Newtons.R) 为

```
Newtons<-function (fun, x, ep=1e-5, it_max=100){
    index<-0; k<-1
    while (k<=it_max){
        x1 <- x; obj <- fun(x);
        x  <- x - solve(obj$J, obj$f);
        norm <- sqrt((x-x1) %*% (x-x1))
        if (norm<ep){
            index<-1; break
        }
        k<-k+1
    }
    obj <- fun(x);
    list(root=x, it=k, index=index, FunVal= obj$f)
}
```

在此函数 (程序) 中，输入变量有： fun 是由方程构成的函数，具体形式在下面介绍． x 是初始变量， ep 是精度要求，缺省时为 10^{-5}. it_max 是最大迭代次数，缺省时为 100.

函数 (程序) 以列表的形式作为输出变量，有： root 是方程解的近似值. it 是迭代次数． index 是指标， index=1 表明计算成功； index=0 表明计算失败． FunVal 是方程在 root 处的函数值.

编写求方程的函数 (程序名： funs.R)

```
funs<-function(x){
   f<-c(x[1]^2+x[2]^2-5, (x[1]+1)*x[2]-(3*x[1]+1))
```

```
    J<-matrix(c(2*x[1], 2*x[2], x[2]-3, x[1]+1),
              nrow=2, byrow=T)
    list(f=f, J=J)
}
```

函数 (程序) 的输入变量是 x. 在函数 (程序) 中, f 是所求方程的函数, J 是相应的 Jacobi 矩阵. 函数的输出以列表形式给出, 输出函数值和相应的 Jacobi 矩阵.

下面求解该方程

```
> Newtons(funs, c(0,1))
$root
[1] 1 2
$it
[1] 6
$index
[1] 1
$FunVal
[1] 1.598721e-14 6.217249e-15
```

即方程的解 $x^* = (1, 2)^T$, 总共迭代了 6 次.

2.9.4 递归函数

R 函数是可以递归的, 可以在函数自身内定义函数本身. 下面的例子是用递归函数计算数值积分.

例 2.7 用递归函数计算数值积分 $\int_1^5 \frac{dx}{x}$, 精度要求 $\varepsilon = 10^{-6}$.

解: 采用自动选择步长的复化梯形公式, 其方法是: 每次将区间二等分, 在子区间上采用梯形求积公式, 如果计算满足精度要求或达到最大迭代次数, 则停止计算; 否则继续将区间对分. 编写相应的计算程序 (程序名: area.R)

```
area <- function(f, a, b, eps = 1.0e-06, lim = 10) {
   fun1 <- function(f, a, b, fa, fb, a0, eps, lim, fun) {
      d <- (a + b)/2; h <- (b - a)/4; fd <- f(d)
      a1 <- h * (fa + fd); a2 <- h * (fd + fb)
      if(abs(a0 - a1 - a2) < eps || lim == 0)
```

```
            return(a1 + a2)
        else {
            return(fun(f, a, d, fa, fd, a1, eps, lim - 1, fun)
                + fun(f, d, b, fd, fb, a2, eps, lim - 1, fun))
        }
    }
    fa <- f(a); fb <- f(b); a0 <- ((fa + fb) * (b - a))/2
    fun1(f, a, b, fa, fb, a0, eps, lim, fun1)
}
```

程序的输入变量， f 是被积函数， a,b 是积分的端点， eps 是积分精度要求，缺省值为 10^{-6}. lim 是对分区间的上限，缺省值为 10, 即被积区间最多被等分为 2^{10} 个子区间. 输出变量为积分值.

area 函数相当于主程序，首先用梯形公式计算出积分的近似值，然后调用函数 fun1.

fun1 函数相当于子程序，该函数是采用递归的定义方式编写的函数，其意义是：将区间对分，采用复化求积公式，若本次的计算值与上一次的计算值相差小于精度要求 eps 或 lim = 0 时，则停止计算；否则分别调用自身函数.

下面计算各分. 先定义函数

```
> f <- function(x) 1/x
```

再计算其积分值

```
> quad<-area(f,1,5); quad
[1] 1.609452
```

该积分的精确值为 $\ln 5 = 1.609438$.

习题二

2.1 建立一个 R 文件，在文件中输入变量 $x = (1, 2, 3)^T$, $y = (4, 5, 6)^T$, 并作以下运算.

(1) 计算 $z = 2x + y + e$, 其中 $e = (1, 1, 1)^T$;

(2) 计算 x 与 y 的内积;

(3) 计算 x 与 y 的外积.

2.2 将 $1, 2, \cdots, 20$ 构成两个 4×5 阶的矩阵，其中矩阵 A 是按列输入，矩阵 B 是按行输入，并作如下运算.

(1) $C = A + B$;

(2) $D = AB$;

(3) $E = (e_{ij})_{n\times n}$, 其中 $e_{ij} = a_{ij} \cdot b_{ij}$;

(4) F 是由 A 的前 *3* 行和前 *3* 列构成的矩阵;

(5) G 是由矩阵 B 的各列构成的矩阵，但不含 B 的第 *3* 列.

2.3 构造一个向量 x, 向量是由 *5* 个 *1*, *3* 个 *2*, *4* 个 *3* 和 *2* 个 *4* 构成，注意用到 `rep()` 函数.

2.4 生成一个 *5* 阶的 *Hilbert* 矩阵,

$$H = (h_{ij})_{n\times n}, \quad h_{ij} = \frac{1}{i+j-1}, \quad i, j = 1, 2, \cdots, n.$$

(1) 计算 *Hilbert* 矩阵 H 的行列式,

(2) 求 H 的逆矩阵;

(3) 求 H 的特征值和特征向量.

2.5 已知有 *5* 名学生的数据，如表 *2.3* 所示. 用数据框的形式读入数据.

表 2.3: 学生数据

序号	姓名	性别	年龄	身高 (cm)	体重 (kg)
1	张三	女	14	156	42.0
2	李四	男	15	165	49.0
3	王五	女	16	157	41.5
4	赵六	男	14	162	52.0
5	丁一	女	15	159	45.5

2.6 将例 *2.5* 中的数据表 *2.3* 的数据写成一个纯文本文件, 用函数 `read.table()` 读该文件，然后再用函数 `write.csv()` 写成一个能用 *Excel* 表能打开的文件, 并用 *Excel* 表打开.

2.7 编写一个 R 程序 *(*函数*)*. 输入一个整数 n, 如果 $n \leq 0$, 则中止运算，并输出一句话：“要求输入一个正整数”; 否则，如果 n 是偶数，则将 n 除 2, 并赋给 n; 否则，将 $3n+1$ 赋给 n. 不断循环，只到 $n=1$, 才停止计算，并输出一句话：“运算成功”. 这个例子是为了检验数论中的一个简单的定理.

第三章　数据描述性分析

统计分析分为统计描述和统计推断两个部分. 统计描述是通过绘制统计图、编制统计表、计算统计量等方法来表述数据的分布特征. 它是数据分析的基本步骤, 也是进行统计推断的基础. 本章介绍统计描述, 也就是数据的描述性分析, 关于统计推断的内容, 将在后面各章陆续介绍.

用计算机软件作数据的描述性分析, 可以更加方便、直观, 有利于对统计描述的理解. 本章除介绍描述统计的基本概念外, 重点介绍如何运用 R 软件中的函数对数据进行描述性分析.

3.1　描述统计量

已知一组试验 (或观测) 数据为

$$x_1, x_2 \cdots, x_n.$$

它们可以是从所要研究的对象的全体 — 总体 X 中取出的, 这 n 个观测值就构成一个样本. 在某些简单的实际问题中, 这 n 个观测值就是所要研究问题的全体. 数据分析的任务就是要对这全部 n 个数据进行分析, 提取数据中包含的有用信息.

数据作为信息的载体, 当然要分析数据中包含的主要信息, 即要分析数据的主要特征. 也就是说, 要研究数据的数字特征. 对于数据的数字特征, 要分析数据的集中位置、分散程度和数据分布等.

3.1.1　位置的度量

所谓位置的度量就是那些用来描述定量资料的集中趋势的统计量. 常用的有均值、众数、中位数、百分位数等.

1. 均值

均值 (mean) 是数据的平均数, 均值 (记为 $\overline{x}$) 定义为

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{3.1}$$

它描述数据取值的平均位置.

在 R 软件中，可用 mean() 函数计算样本的均值，其的使用方法是

```
mean(x, trim = 0, na.rm = FALSE)
```

其中 x 是对象 (如向量、矩阵、数组或数据框), trim 是计算均值前去掉与均值差较大数据的比例，缺省值为 0, 即包括全部数据. 当 na.rm = TRUE 时，允许数据中有缺失数据. 函数的返回值是对象的均值.

有关它的使用，将用例子来作进一步的介绍.

例 3.1 *已知 15 位学生的体重 (单位：千克)*

75.0	64.0	47.4	66.9	62.2	62.2	58.7	63.5
66.6	64.0	57.0	69.0	56.9	50.0	72.0	

求学生体重的平均值.

解：利用 mean() 函数求解. 建立 R 文件 (文件名： exam0301.R)

```
w <- c(75.0, 64.0, 47.4, 66.9, 62.2, 62.2, 58.7, 63.5,
       66.6, 64.0, 57.0, 69.0, 56.9, 50.0, 72.0)
w.mean <- mean(w); w.mean
```

执行 exam0301.R 的的全部程序得到：学生体重的均值为 62.36.

注意，当 x 是矩阵 (或数组) 时，函数 mean() 的返回值，并不是向量，而是一个数，即矩阵中全部数据的平均值. 例如,

```
> x <- 1:12; dim(x)<-c(3,4)
> mean(x)
[1] 6.5
```

与 mean(1:12) 的返回值相同，而这里 x 是一个 3×4 的矩阵.

如果你需要得到矩阵各行或各列的均值，需要调用 apply() 函数 (见第二章 2.5.5 节) 计算. 如计算矩阵各行的均值,

```
> apply(x,1,mean)
[1] 5.5 6.5 7.5
```

计算矩阵各列的均值,

```
> apply(x,2,mean)
[1]  2  5  8 11
```

如果 x 是数据框，则 mean() 的返回值就是向量，如

```
> mean(as.data.frame(x))
V1 V2 V3 V4
 2  5  8 11
```

可以看出它是按列求平均值的，其中命令 as.data.frame(x)(见第二章 2.6.2 节)是将矩阵 x 强制转化成数据框.

因此，今后在作多元数据分析时，多元数据的输入最好采用数据框的形式，这样便于后面的数据处理.

求和函数 sum() 是与求均值有关的函数，其使用格式为

```
sum(..., na.rm = FALSE)
```

参数 na.rm 的意义与均值函数 mean() 中的参数意义相同.

如果 x 是向量，函数 length(x) 的返回值是向量 x 的长度 (维数). 因此，由公式 (3.1), 例 3.1 的均值可由下面的计算得到，即

```
> mean <- sum(w)/length(w); mean
[1] 62.36
```

可以看出，两者的计算是相同的.

但如果在数据中，某些数据是异常值，再用公式 (3.1) 就不合理了. 也就是说，不能简单地用 mean(w) 计算样本均值. 例如，如果第一个学生的体重少输入一个点，变为 750 千克，此时按照式 (3.1) 计算出的值会出现不合理的现象，看一下计算结果

```
> w[1] <- 750
> w.mean <- mean(w); w.mean
[1] 107.36
```

学生的平均体重为 107.36 千克，这显然是不合理的.

如果采用下述方法，可以减少由于输入误差对计算的影响.

```
> w.mean <- mean(w, trim=0.1); w.mean
[1] 62.53846
```

其中 trim 的取值在 0 至 0.5 之间，表示在计算均值前需要去掉异常值的比例. 利用这个参数可以有效的改善异常值的对计算的影响.

na.rm 是控制缺失数据的参数. 例如，如果共有 16 位学生，但第 16 位学生的体重缺失，如果按照通常的计算方法，将得不到结果.

```
> w.na <- c(75.0, 64.0, 47.4, 66.9, 62.2, 62.2, 58.7, 63.5,
            66.6, 64.0, 57.0, 69.0, 56.9, 50.0, 72.0, NA)
> w.mean <- mean(w.na); w.mean
[1] NA
```

选用参数 na.rm = TRUE 可以很好地处理这个问题, 看一下计算结果.

```
> w.mean <- mean(w.na, na.rm = TRUE); w.mean
[1] 62.36
```

对于 sum() 函数, 此参数的意义是相同的, 即 na.rm = TRUE 表示可以求带有缺失数据的和.

与均值函数 mean() 相关的函数还有 weighted.mean(), 即计算数据的加权平均值, 具体的使用格式为

```
weighted.mean(x, w, na.rm = FALSE)
```

其中 x 是数值向量, w 是数据 x 是权, 与 x 的维数相同. 参数 na.rm 的意义与 mean() 函数相同. 该函数可以对矩阵和数组计算加权平均值, 但对数据框不适用 (对于数据框, weighted.mean() 函数的计算结果与矩阵的计算结果是相同的, 而 mean() 函数两者的计算结果是不同的).

2. 顺序统计量

设 n 个数据 (观测值) 按从小到大的顺序排列为

$$x_{(1)} \le x_{(2)} \le \cdots \le x_{(n)},$$

称为顺序统计量 (order statistic), 显然, 最小顺序统计量为 $x_{(1)}$, 最大顺序统计量为 $x_{(n)}$.

在 R 软件中, sort() 给观测量的顺序统计量. 如

```
> x <- c(75, 64, 47.4, 66.9, 62.2, 62.2, 58.7, 63.5)
> sort(x)
[1] 47.4 58.7 62.2 62.2 63.5 64.0 66.9 75.0
```

实际上, 函数 sort() 不只是给出了样本的顺序统计量, 还有更广泛的功能, 其使用格式为

```
sort(x, partial = NULL, na.last = NA, decreasing = FALSE,
     method = c("shell", "quick"), index.return = FALSE)
```

其中 x 是数值、或字符、或逻辑型向量. partial 是部分排序的指标向量. na.last 是控制缺失数据的参数, 当 na.last = NA(缺省值) 时, 不处理缺失数据; 当 na.last = TRUE 时, 缺失数据排在最后; 当 na.last = FALSE 时, 缺失数据排在最前面. decreasing 是逻辑变量, 控制数据排列的顺序, 当 decreasing = FALSE (缺省值), 给出的返回值, 是由小到大排序的; 如果 decreasing = TRUE, 则函数的返回值由大到小排列. method 是排序的方法, 如果 method = "shell" (缺省值), 则选择 Shell 排序法排序, 其运算量为 $O(n^{4/3})$; 如果 method = "quick", 则采用快速排序法排序, 对于数值型向量, 快速排序法的运算量一般要低于 Shell 排序法. index.return 是逻辑变量, 是控制排序下标的返回值, 当 index.return = TRUE 时 (缺省值为 FALSE), 函数的返回值是一列表, 列表的第一个变量 $x 是排序的顺序, 第二个变量是 $ix 是排序顺序的下标对应的值.

下面用数值例子看一下函数 sort() 中各种参数的使用方法. 如需要将数据由大到小排, 则用参数 decreasing = TRUE. 如

```
> sort(x, decreasing = TRUE)
[1] 75.0 66.9 64.0 63.5 62.2 62.2 58.7 47.4
```

当数据中有缺失数据时, 并不希望处理缺失数据, 则不必调整任何参数. 如

```
> x.na <- c(75.0,64.0,47.4,NA,66.9,62.2,62.2,58.7,63.5)
> sort(x.na)
[1] 47.4 58.7 62.2 62.2 63.5 64.0 66.9 75.0
```

如果希望在排序后的数据中保留缺失数据, 并将缺失数据排在最后, 则用 na.last = TRUE. 如果将缺失数据排在最前, 则用 na.last = FALSE. 如

```
> sort(x.na, na.last = TRUE)
[1] 47.4 58.7 62.2 62.2 63.5 64.0 66.9 75.0   NA
> sort(x.na, na.last = FALSE)
[1]   NA 47.4 58.7 62.2 62.2 63.5 64.0 66.9 75.0
```

与 sort() 函数相关的函数有: order() 给出排序后的下标; rank() 给出样本的秩统计量, 关于 rank() 函数在第五章还会介绍.

3. 中位数

中位数 (median, 记为 m_e) 定义为数据排序位于中间位置的值，即

$$m_e = \begin{cases} x_{(\frac{n+1}{2})}, & \text{当 } n \text{ 为奇数时,} \\ \dfrac{1}{2}\left(x_{(\frac{n}{2})} + x_{(\frac{n}{2}+1)}\right), & \text{当 } n \text{ 为偶数时.} \end{cases} \tag{3.2}$$

中位数描述数据中心位置的数字特征. 大体上比中位数大或小的数据个数为整个数据的一半. 对于对称分布的数据，均值与中位数比较接近；对于偏态分布的数据，均值与中位数不同. 中位数的又一显著特点是不受异常值的影响，具有稳健性，因此它是数据分析中相当重要的统计量.

在 R 软件中，函数 `median()` 给观测量的中位数. 如

```
> x <- c(75, 64, 47.4, 66.9, 62.2, 62.2, 58.7, 63.5)
> median(x)
[1] 62.85
```

`median()` 函数的使用格式为

```
median(x, na.rm = FALSE)
```

其中 `x` 是数值型向量， `na.rm` 是逻辑变量，当 `na.rm = TRUE` 时，函数可以处理带有缺失数据的向量；否则 (`na.rm = FALSE`, 缺省值) 不能处理带有缺失数据的向量. 如

```
> x.na <- c(75.0,64.0,47.4,NA,66.9,62.2,62.2,58.7,63.5)
> median(x.na)
[1] NA
> median(x.na, na.rm = TRUE)
[1] 62.85
```

4. 百分位数

百分位数 (percentile) 是中位数的推广. 将数据按从小到大的排列后，对于 $0 \le p < 1$, 它的 p 分位点定义为

$$m_p = \begin{cases} x_{([np]+1)}, & \text{当 } np \text{ 不是整数时,} \\ \dfrac{1}{2}\left(x_{(np)} + x_{(np+1)}\right), & \text{当 } np \text{ 是整数时,} \end{cases} \tag{3.3}$$

其中 $[np]$ 表示 np 的整数部分.

p 分位数又称为第 $100p$ 百分位数. 大体上整个样本的 $100p$ 的观测值不超过 p 分位数. 如 0.5 分位数 $m_{0.5}$(第 50 百分位数) 就是中位数 m_e. 在实际计算中，

0.75 分位数与 0.25 分位数 (第 75 百分位数与第 25 百分位数) 比较重要，它们分别称为上、下四分位数，并分别记为 $Q_3 = m_{0.75}$, $Q_1 = m_{0.25}$.

在 R 软件中， quantile() 函数计算观测量的百分位数. 如

```
> w <- c(75.0, 64.0, 47.4, 66.9, 62.2, 62.2, 58.7, 63.5,
         66.6, 64.0, 57.0, 69.0, 56.9, 50.0, 72.0)
> quantile(w)
   0%   25%   50%   75%  100%
47.40 57.85 63.50 66.75 75.00
```

quantile() 函数的一般使用格式为

```
quantile(x, probs = seq(0, 1, 0.25), na.rm = FALSE,
      names = TRUE, type = 7, ...)
```

其中 x 是由数值构成的向量. probs 是给出相应的百分位数，缺省时是 0 、$\frac{1}{4}$ 、$\frac{1}{2}$ 、$\frac{3}{4}$ 、 1. na.rm 是逻辑变量，当 na.rm = TRUE 时，可处理缺失数据. 其余见帮助.

如果打算给出 0%, 20%, 40%, 60%, 80% 和 100% 的百分位数，则选择

```
> quantile(w, probs = seq(0, 1, 0.2))
   0%   20%   40%   60%   80%  100%
47.40 56.98 62.20 64.00 67.32 75.00
```

3.1.2 分散程度的度量

表示数据分散 (或变异) 程度的特征量有方差、标准差、极差、四分位极差、变异系数和标准误等.

1. 方差、标准差与变异系数

方差 (variance) 是描述数据取值分散性的一个度量. 样本方差 (sample variance) 是样本相对于均值的偏差平方和的平均，记为 s^2, 即

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2 . \tag{3.4}$$

其中 $\overline{x}$ 是样本的均值.

样本方差的开方称为样本标准差 (standard deviation), 记为 s, 即

$$s = \sqrt{s^2} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}. \tag{3.5}$$

变异系数是刻划数据相对分散性的一种度量，记为 CV,

$$\mathrm{CV} = 100 \times \frac{s}{\overline{x}}(\%), \tag{3.6}$$

它是一个无量纲的量，用百分数表示.

与分散程度有关的统计量还有下列数字特征：

样本校正平方和

$$\mathrm{CSS} = \sum_{i=1}^{n}(x_i - \overline{x})^2. \tag{3.7}$$

样本未校正平方和

$$\mathrm{USS} = \sum_{i=1}^{n} x_i^2. \tag{3.8}$$

在 R 软件中，若 x 是由样本构成的向量，则 `var(x)` 计算样本方差， `sd(x)` 计算样本标准差，即 `sd(x)` $= \sqrt{\texttt{var(x)}}$. 例如，对于 15 名学生的体重数据，有

```
> var(w)
[1] 56.47257
> sd(w)
[1] 7.514823
```

方差函数 `var()` 和标准差函数 `sd()` 的使用格式为

```
var(x, y = NULL, na.rm = FALSE, use)
sd(x, na.rm = FALSE)
```

其中 x 是数值向量、矩阵或数据框. `na.rm` 是逻辑变量，当 `na.rm = TRUE` 时，可处理缺失数据. 其余见帮助.

与方差函数 `var()` 相关的函数还有： `cov()`— 求协方差矩阵； `cor()` — 求相关矩阵. 这两个函数将在后面介绍.

对于变异系数、校正平方和、未校正平方和等指标，需要编写简单的程序. 例如，对于 15 名学生的体重数据

```
> cv <- 100*sd(w)/mean(w); cv
[1] 12.05071
> css <- sum((w-mean(w))^2); css
[1] 790.616
> uss <- sum(w^2); uss
[1] 59122.16
```

2. 极差与标准误

样本极差 (记为 R) 的计算公式为

$$R = x_{(n)} - x_{(1)} = \max(x) - \min(x), \tag{3.9}$$

其中 x 是由样本构成的向量. 样本极差是描述样本分散性的数字特征. 当数据越分散, 其极差越大.

样本上、下四分位数之差称为四分位差 (或半极差), 记为 R_1, 即

$$R_1 = Q_3 - Q_1, \tag{3.10}$$

它也是度量样本分散性的重要数字特征, 特别对于具有异常值的数据, 它作为分散性具有稳健性, 因此它在稳健性数据分析中具有重要作用.

样本标准误 (记为 s_m) 定义为

$$s_m = \sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{n}(x_i - \overline{x})^2} = \frac{s}{\sqrt{n}}. \tag{3.11}$$

对于样本极差与样本标准误, 可以简单编程方法计算.

3.1.3 分布形状的度量

在第一章的 1.3.5 节介绍过总体的偏度 (skewness) 系数和峰度 (kurtosis) 系数, 这里介绍样本的偏度系数和峰度系数.

1. 偏度系数

样本的偏度系数 (记为 g_1) 的计算公式为

$$g_1 = \frac{n}{(n-1)(n-2)s^3}\sum_{i=1}^{n}(x_i - \overline{x})^3 = \frac{n^2\mu_3}{(n-1)(n-2)s^3}, \tag{3.12}$$

其中 s 是标准差， μ_3 是样本 3 阶中心矩，即 $\mu_3 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^3$.

偏度系数是刻划数据的对称性指标. 关于均值对称的数据其偏度系数为 0, 右侧更分散的数据偏度系数为正，左侧更分散的数据偏度系数为负.

2. 峰度系数

样本的峰度系数 (记为 g_2) 的计算公式为

$$\begin{aligned} g_2 &= \frac{n(n+1)}{(n-1)(n-2)(n-3)s^4}\sum_{i=1}^{n}(x_i-\overline{x})^4 - 3\frac{(n-1)^2}{(n-2)(n-3)} \\ &= \frac{n^2(n+1)\mu_4}{(n-1)(n-2)(n-3)s^4} - 3\frac{(n-1)^2}{(n-2)(n-3)}, \end{aligned} \tag{3.13}$$

其中 s 是标准差， μ_4 是样本 4 阶中心矩，即 $\mu_4 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^4$.

当数据的总体分布为正态分布时，峰度系数近似为 0; 当分布较正态分布的尾部更分散时，峰度系数为正；否则为负. 当峰度系数为正时，两侧极端数据较多；当峰度系数为负时，两侧极端数据较少.

最后编写一个统计的函数 (程序名： `data_outline.R`), 计算样本的各种描述性统计量.

```
data_outline <- function(x){
   n <- length(x)
   m <- mean(x)
   v <- var(x)
   s <- sd(x)
   me <- median(x)
   cv <- 100*s/m
   css <- sum((x-m)^2)
   uss <- sum(x^2)
   R <-  max(x)-min(x)
   R1 <- quantile(x,3/4)-quantile(x,1/4)
   sm <- s/sqrt(n)
   g1 <- n/((n-1)*(n-2))*sum((x-m)^3)/s^3
   g2 <- ((n*(n+1))/((n-1)*(n-2)*(n-3))*sum((x-m)^4)/s^4
```

```
            - (3*(n-1)^2)/((n-2)*(n-3)))
    data.frame(N=n, Mean=m, Var=v, std_dev=s,
       Median=me, std_mean=sm, CV=cv, CSS=css, USS=uss,
       R=R, R1=R1, Skewness=g1, Kurtosis=g2, row.names=1)
}
```

函数的输入变量 x 是数值型向量, 由样本构成. 函数的返回值是数据框, 包含以下指标: N 样本的个数; Mean 样本均值; Var 样本方差; std_dev 样本标准差; Median 样本中位数; std_mean 样本的标准误; CV 样本的变异系数; CSS 样本校正平方和; USS 样本未校正平方和; R 样本极差; R1 样本半极差; Skewness 样本峰度系数; Kurtosis 样本偏度系数.

例 3.2 计算例 *3.1* 中 *15* 位学生的体重的各种统计量.

解: 将编好的程序调入内存 (见第二章中输入 R 代码), 输入数据并计算得到相应的结果.

```
> source("data_outline.R")
> w <- c(75.0, 64.0, 47.4, 66.9, 62.2, 62.2, 58.7, 63.5,
         66.6, 64.0, 57.0, 69.0, 56.9, 50.0, 72.0)
> data_outline(w)
   N  Mean      Var  std_dev Median std_mean       CV
1 15 62.36 56.47257 7.514823   63.5 1.940319 12.05071
      CSS      USS    R  R1   Skewness   Kurtosis
1 790.616 59122.16 27.6 8.9 -0.4299561 0.09653947
```

3.2 数据的分布

数据的数字特征刻划了数据的主要特征, 而要对数据的总体情况作全面的描述, 就要研究数据的分布. 对数据分布的主要描述方法有直方图、茎叶图和数据的理论分布即总体分布. 数据分析的一个重要问题是要研究数据是否来自正态总体, 这是分布的正态性检验的问题.

3.2.1 分布函数

在第一章给出了分布函数 $F(x)$ 的定义 (定义 1.5) 、分布律 (定义 1.7), 即

$$P\{X = x_k\} = p_k, \quad k = 1, 2, \cdots,$$

和概率密度函数 $f(x)$ 的定义 (定义 1.8), 以及概率密度函数 $f(x)$ 与分布函数 $F(x)$ 的关系

$$F(x) = P\{X \le x\} = \int_{-\infty}^{x} f(t)dt, \quad -\infty < x < \infty.$$

并给出了一些典型的分布, 如正态分布、 Poisson 分布等.

在 R 软件中, 提供了计算这些典型分布的分布函数、分布律或概率密度函数, 以及分布函数的反函数的各种函数.

例如, 考虑正态分布, 设 μ 是均值, σ^2 是方差, 对于任意的变量 x, 其分布函数为

$$F(x) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{x} \exp\left\{-\frac{(t-\mu)^2}{2\sigma^2}\right\} \mathrm{d}t = \texttt{pnorm(x, mu, sigma)},$$

其中函数 pnorm 是 R 软件中计算分布函数 (正态分布) 的函数, mu 是均值 μ, sigma 是标准差 σ. 相应的概率密度函数为

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} = \texttt{dnorm(x, mu, sigma)},$$

其中函数 dnorm 是 R 软件中计算概率密度函数 (正态分布) 的函数.

计算标准正态分布的上 $\alpha/2$ ($\alpha = 0.05$) 分位点, 其计算公式为

$$z_{\alpha/2} = \texttt{qnorm(1-0.025, 0,1)} = 1.959964.$$

其中函数 qnorm 是 R 软件中计算下分位点的函数.

产生 100 个标准正态态分布的随机数

```
r <- rnorm(100, 0, 1)
```

其中函数 rnorm 是 R 软件中生成 (正态分布) 随机数的函数, 参数 0, 1 可以缺省.

关于正态分布函数 dnorm() 、 pnorm() 、 qnorm() 和 rnorm() 的使用方法是

```
dnorm(x, mean=0, sd=1, log = FALSE)
pnorm(q, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean=0, sd=1)
```

其中 x,q 是由数值型变量构成的向量. p 是由概率构成的向量. n 是产生随机数的个数. mean 是要计算的正态分布的均值, 缺省值为 0. sd 是要计算的正态分布的标准差, 缺省值为 1. 函数 dnorm() 的返回值是正态分布的概率密度函数. 函数 pnorm() 的返回值是正态分布的分布函数. 函数 qnorm() 的返回值是给定概率 p 后的下分位点. 函数 rnorm() 的返回值是由 n 个正态分布随机数构成的向量.

log, log.p 是逻辑变量, 当它为真 (TRUE) 时, 函数的返回值不再是正态分布, 而是对数正态分布. lower.tail 是逻辑变量, 当它为真 (TRUE, 缺省值) 时, 分布函数的计算公式为

$$F(x) = P\{X \le x\},$$

当 lower.tail = FALSE 时, 分布函数的计算公式为

$$F(x) = P\{X > x\}.$$

再看一个离散随机变量计算函数的例子, 如 Poisson 分布. Poisson 分布的使用格式为

```
dpois(x, lambda, log = FALSE)
ppois(q, lambda, lower.tail = TRUE, log.p = FALSE)
qpois(p, lambda, lower.tail = TRUE, log.p = FALSE)
rpois(n, lambda)
```

其中 lambda 是 Poisson 分布的参数 λ. 其余参数的意义与上面介绍的函数 (正态分布) 中参数的意义相同.

注意, 由于 Poisson 分布是离散分布, 当 x 是整数 k 时, 其意义为

$$P\{X = k\} = \frac{\lambda \mathrm{e}^{-\lambda}}{k!} = \texttt{dpois(k, lambda)},$$

当 x 不是整数时, dpois(x, lambda)=0. 对于函数 ppois(), 无论 x 是否为整数, 其意义为

$$F(x) = \sum_{k=0}^{\lfloor x \rfloor} \frac{\lambda \mathrm{e}^{-\lambda}}{k!} = \texttt{ppois(x, lambda)}.$$

给定概率 p, `qpois(p, lambda)` 的返回值是 $P\{X = k\} \geq p$ 的最小的整数 k.

其他的分布函数也有类似的结果. 表 3.1 列出了各种常用的分布函数, 概率密度函数或分布律, 以及 R 中的名称和调用函数用到的参数.

表 3.1: 分布函数或分布律

分布	R 中的名称	附加参数
beta	beta	shape1, shape2, ncp
binomial	binom	size, prob
Cauchy	cauchy	location, scale
chi-squared	chisq	df, ncp
exponential	exp	rate
F	f	df1, df2, ncp
gamma	gamma	shape, scale
geometric	geom	prob
hypergeometric	hyper	m, n, k
log-normal	lnorm	meanlog, sdlog
logistic	logis	location, scale
negative binomial	nbinom	size, prob
normal	norm	mean, sd
Poisson	pois	lambda
Student's t	t	df, ncp
uniform	unif	min, max
Weibull	weibull	shape, scale
Wilcoxon	wilcox	m, n

在表 3.1 所列的分布中, 加上不同的前缀表示不同的意义,

- `d` — 概率密度函数 $f(x)$, 或分布律 p_k;
- `p` — 分布函数 $F(x)$;
- `q` — 分布函数的反函数 $F^{-1}(p)$, 即给定概率 p 后, 求其下分位点;

- r — 仿真 (产生相同分布的随机数).

3.2.2 直方图、经验分布图与 QQ 图

1. 直方图

对于数据分布，常用直方图 (histogram) 进行描述. 将数据取值的范围分成若干区间 (一般是等间隔的), 在等间隔的情况下，每个区间长度称为组距. 考察数据落入每一区间的频数与频率，在每个区间上画一个矩形，它的宽度是组距，它的高度可以是频数、频率或频率 / 组距，在高度是频率 / 组距的情况下，每一矩形的面积恰是数据落入区间的频率，这种直方图可以估计总体的概率密度. 组距对直方图的形态有很大的影响，组距太小，每组的频数较少，由于随机性的影响，邻近区间上的频数可能很大；组距太大，直方图所反映的形态就不灵敏.

在 R 软件中，用函数 hist() 画出样本的直方图，其格式为

```
hist(x)
```

或

```
hist(x, breaks = "Sturges", freq = NULL, probability = !freq,
     include.lowest = TRUE, right = TRUE,
     density = NULL, angle = 45, col = NULL, border = NULL,
     main = paste("Histogram of" , xname),
     xlim = range(breaks), ylim = NULL,
     xlab = xname, ylab,
     axes = TRUE, plot = TRUE, labels = FALSE,
     nclass = NULL, ...)
```

其中 x 是由样本构成的向量. breaks 规定直方图的组距，由以下几种形式给出：

- 向量，给出直方图的起点、终点与组距.
- 数，定义直方图的组距.
- 字符串，（见缺省状态）.
- 函数，计算组距的宽度.

freq 是逻辑变量：

- TRUE 绘出频率直方图;

- counts 绘出频率直方图;
- FALSE 绘出密度直方图

probability 是逻辑变量与 freq 相反，是与 S-PLUS 相兼容的参数,

- TRUE 绘出密度直方图;
- FALSE 绘出频率直方图

col 表示直方图中填充的颜色. plot 是逻辑变量:

- TRUE 表示给出直方图;
- FALSE 表示列出绘出直方图的各种结果（并不绘图）.

其它参数见帮助文件.

2. 核密度估计函数

与直方图相配套的是核密度估计 (kernal density estimate) 函数 density(), 其目的是用已知样本，估计其密度. 它的使用方法是:

```
density(x, bw = "nrd0", adjust = 1,
    kernel = c("gaussian", "epanechnikov", "rectangular",
               "triangular", "biweight", "cosine", "optcosine"),
    window = kernel, width,
    give.Rkern = FALSE,
    n = 512, from, to, cut = 3, na.rm = FALSE)
```

其中 x 是由样本构成的向量. bw 是带宽，可选择. 当 bw 为省略值时, R 软件会画出光滑的曲线. 其它参数见帮助文件.

例 3.3 *绘出例 3.1 中 15 位学生的体重的直方图和核密度估计图，并与正态分布的概率密度函数作对照.*

解: 写出 R 程序（程序名: exam0303.R ）

```
w <- c(75.0, 64.0, 47.4, 66.9, 62.2, 62.2, 58.7, 63.5,
       66.6, 64.0, 57.0, 69.0, 56.9, 50.0, 72.0)
hist(w, freq = FALSE)
lines(density(w), col = "blue")
x <- 44:76
lines(x, dnorm(x, mean(w), sd(w)), col = "red")
```

执行后绘出直方图和密度估计曲线和正态分布的概率密度曲线，如图 3.1 所示.

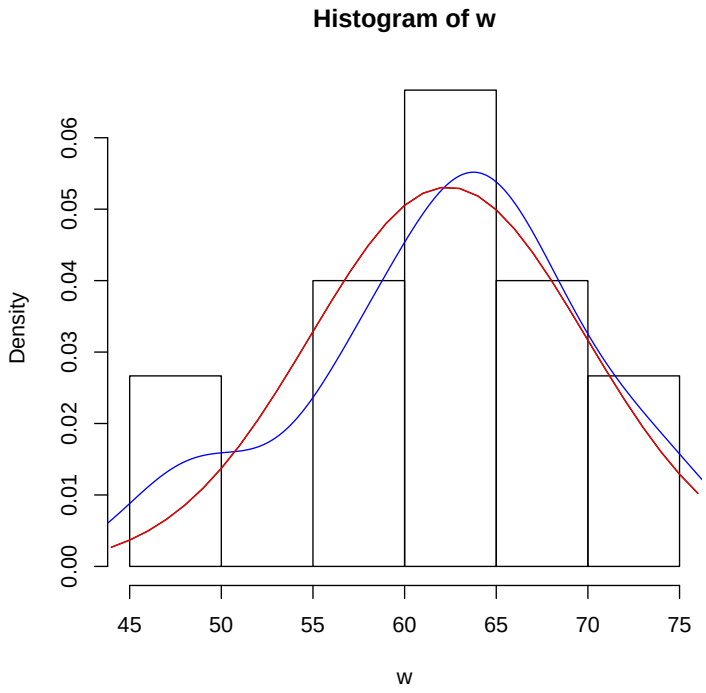


图 3.1: 学生体重的直方图、密度估计曲线与正态分布密度曲线

注意到，密度估计曲线与正态分布的概率密度曲线还是有一定的差别的.

结合直方图和密度估计曲线来进一步分析例 3.2 中统计量的意义. 如偏度小于 0, 直方图偏右等.

3. 经验分布

直方图的制作适合于总体为连续型分布的场合. 对于一般的总体分布，若要估计它的总体分布函数 $F(x)$, 可用经验分布函数 (empirical distribution function) 作估计. 在第一章的 1.5.3 节给出了经验分布的定义 (见式 (1.87)), 在 R 中，用函数 `ecdf()` 绘出样本的经验分布函数，其用法是:

```
ecdf(x)
plot(x, ..., ylab="Fn(x)", verticals = FALSE,
     col.01line = "gray70")
```

其中, 在函数 `ecdf()` 中的 `x` 是由观察值得到的数值型向量, 而在函数 `plot()` 中的 `x` 是由函数 `ecdf()` 生成的向量. `verticals` 是逻辑变量, 当 `verticals = TRUE` 表示画竖线；否则 (`FALSE`, 缺省值) 不画竖线.

例 3.4 绘出例 *3.1* 中 *15* 位学生的体重的经验分布图和相应的正态分布图.

解: 写出 R 程序 (程序名： exam0304.R)

```
plot(ecdf(w),verticals = TRUE, do.p = FALSE)
x <- 44:78
lines(x, pnorm(x, mean(w), sd(w)))
```

其中 do.p 是逻辑变量, 当 do.p = FALSE 表示不画点处的记号; 否则 (TRUE, 缺省值) 画记号.

执行后绘出经验分布图和正态分布曲线, 如图 3.2 所示.

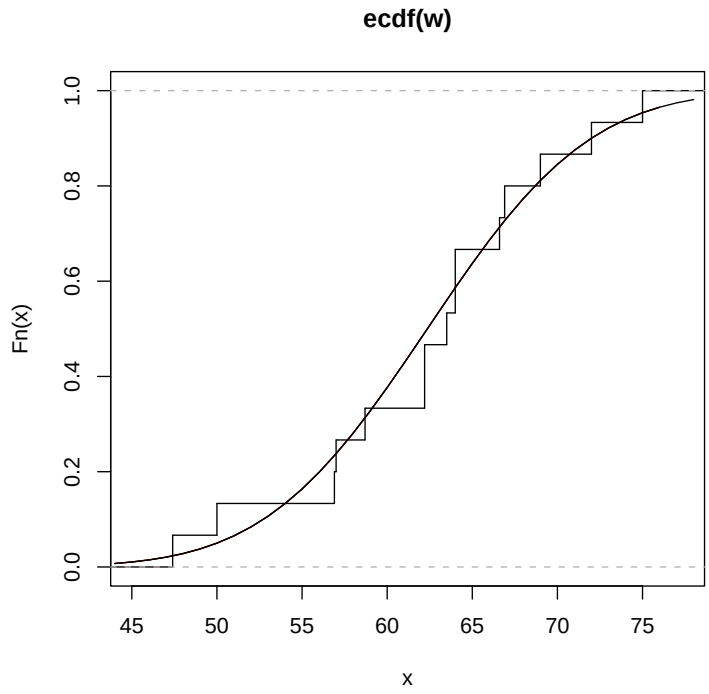


图 3.2: 学生体重的经验分布图和正态分布曲线

4. QQ 图

不论是直方图还经验分布图, 要从比较上鉴别样本是否近似于某种类型的分布是困难的, QQ 图可以帮助我们鉴别样本的分布是否近似于某种类型的分布.

现假定总体为正态分布 $N(\mu,\sigma^2)$, 对于样本 $x_1,x_2,\cdots,x_n$, 其顺序统计量是 $x_{(1)},x_{(2)},\cdots,x_{(n)}$. 设 $\Phi(x)$ 是标准正态分布 $N(0,1)$ 的分布函数, $\Phi^{-1}(x)$ 是反函

数，对应正态分布的 QQ 图是由以下的点

$$\left(\Phi^{-1}\left(\frac{i-0.375}{n+0.25}\right), x_{(i)}\right), \quad i=1,2,\cdots,n \tag{3.14}$$

构成的散点图. 若样本数据近似于正态分布，在 QQ 图上这些点近似地在直线

$$y=\sigma x+\mu$$

附近. 此直线的斜率是标准差 σ, 截距是均值 μ. 所以利用正态 QQ 图可以作直观的正态性检验. 若正态 QQ 图上的点近似地在一条直线附近，可以认为样本数据来自正态分布总体.

在 R 软件中，函数 qqnorm() 和 qqline() 提供了画正态 QQ 图和相应直线的方法. 其使用方法是:

```
qqnorm(y, ...)
qqnorm(y, ylim, main = "Normal Q-Q Plot",
       xlab = "Theoretical Quantiles",
       ylab = "Sample Quantiles", plot.it = TRUE,
       datax = FALSE, ...)
qqline(y, datax = FALSE, ...)
qqplot(x, y, plot.it = TRUE, xlab = deparse(substitute(x)),
       ylab = deparse(substitute(y)), ...)
```

其中 x 是第一列样本. y 是第二列样本或只有此列样本. xlab，ylab，main 是图标. 其它参数见帮助文件.

例 3.5 *绘出例 3.1 中 15 位学生的体重的正态 QQ 图，并从直观上鉴别样本数据是否来自正态分布总体.*

解: 写出 R 程序（程序名: exam0305.R ）

```
w <- c(75.0, 64.0, 47.4, 66.9, 62.2, 62.2, 58.7, 63.5,
       66.6, 64.0, 57.0, 69.0, 56.9, 50.0, 72.0)
qqnorm(w); qqline(w)
```

执行后绘出正态 QQ 图，如图 3.3 所示.

从正态 QQ 图（图 3.3 ）来看，样本的数据基本上可以看成来自正态总体.

对于对数正态、指数等分布也可以作相应的 QQ 图，用以鉴别样本数据是否来自某一类型的总体分布.

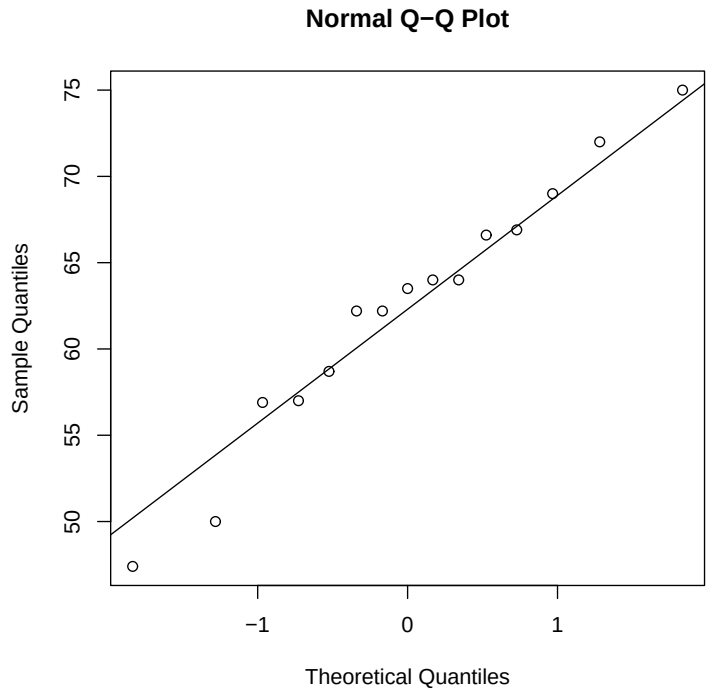


图 3.3: 学生体重的正态 QQ 图

3.2.3 茎叶图、箱线图及五数总括

1. 茎叶图

与直方图比较, 茎叶图更能细致地看出数据分布的结构. 下面用具体的例子来说明茎叶图的意义.

例 3.6 某班有 *31* 名学生, 某门课的考试成绩如下:

```
25  45  50  54  55  61  64  68  72  75  75
78  79  81  83  84  84  84  85  86  86  86
87  89  89  89  90  91  91  92  100
```

做出其茎叶图.

解: 在 R 软件中, 用 `stem()` 函数作茎叶图, 其命令 (程序名: `exam0306.R`) 如下

```
> x<-c(25, 45, 50, 54, 55, 61, 64, 68, 72, 75, 75,
       78, 79, 81, 83, 84, 84, 84, 85, 86, 86, 86,
       87, 89, 89, 89, 90, 91, 91, 92, 100)
```

```
> stem(x)
  The decimal point is 1 digit(s) to the right of the |
   2 | 5
   3 |
   4 | 5
   5 | 045
   6 | 148
   7 | 25589
   8 | 1344456667999
   9 | 0112
  10 | 0
```

下面对茎叶图给出相应的解释.

第一个数 25 的十位为 2, 个位为 5. 以个位为单位，将 25 用 | 号分开：

$$25 \quad \rightarrow \quad 2 \mid 5$$

每一个数都可以这样处理. 因此，茎叶图将十位数 2,3,4,5,6,7,8,9,10 按纵列从上到下排列，在纵列右侧从上到下画一竖线，再在竖线右侧写上原始数据的相应的个们数. 例如，在十位数 5 的竖线右侧依次应是 0,4,5, 即

$$5 \mid 045$$

它们分别对应着 50, 54, 55 这三个数据. 又如在十位数 3 的竖线的右侧，因为从原始数据看，没有对应的数据可填，可以空着.

在茎叶图中，纵轴为测定数据，横轴为数据频数. 数据的十位数部分表示“茎”, 作为纵轴的刻度；个位数部分作为 “叶”, 显示频数的个数，作用与直方图的直方类似.

`stem()` 函数的使用方法是：

```
stem(x, scale = 1, width = 80, atom = 1e-08)
```

其中 `x` 是数据向量. `scale` 控制绘出茎叶图的长度. `width` 绘图的宽度. `atom` 是容差.

如果选择 `scale = 2`, 即将 10 个个位数分成两段， $0 \sim 4$ 为一段， $5 \sim 9$ 为另一段，看下面的计算结果

```
> stem(x, scale = 2)
```

```
  The decimal point is 1 digit(s) to the right of the |
   2 | 5
   3 |
   3 |
   4 |
   4 | 5
   5 | 04
   5 | 5
   6 | 14
   6 | 8
   7 | 2
   7 | 5589
   8 | 13444
   8 | 56667999
   9 | 0112
   9 |
  10 | 0
```

如果选择 scale = 1/2, 即将 10 个个位数分成 1/2 段, 即 20 个数为一段, 如

```
> stem(x, scale = .5)
  The decimal point is 1 digit(s) to the right of the |
   2 | 5
   4 | 5045
   6 | 14825589
   8 | 13444566679990112
  10 | 0
```

例 3.7 绘出例 *3.1* 中 *15* 位学生的体重的茎叶图.

解:

```
> stem(w)
  The decimal point is 1 digit(s) to the right of the |
  4 | 7
  5 | 0779
```

```
6 | 22444779
7 | 25
```

注意到：为了使数据分析简化，将原始数据小数点后数值四舍五入.

2. 箱线图

茎叶图是探索性数据分析所采用的重要方法. 而箱线图确能直观简洁地展现数据分布的主要特征. 在 R 软件中，用 `boxplot()` 函数作箱线图.

例 3.8 *绘出例 3.6 学生考试成绩的箱线图.*

解: 输入命令

```
> boxplot(x)
```

得到箱线图，如图 3.4 所示.

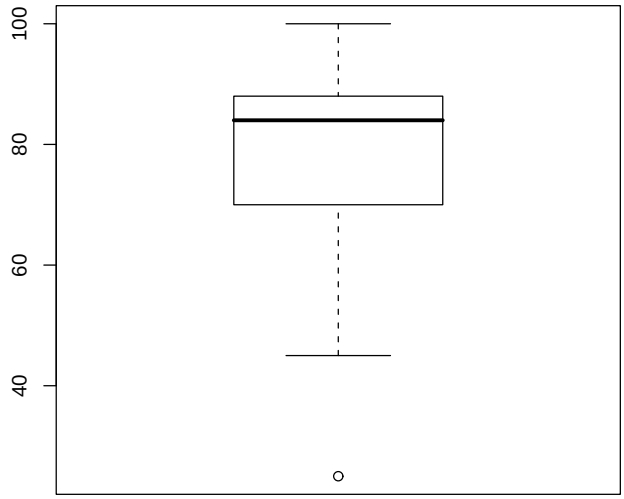


图 3.4: 学生成绩的箱线图

在箱线图中，上 (Q_3) 下 (Q_1) 四分位数分别确定出中间箱体的顶部主底部. 箱体中间的粗线是中位数 (m_e) 所在的位置. 由箱体向上下伸出的垂直部分称为“触须”, 表示数据的散布范围，最远点为 1.5 倍四分位数间距. 超出此范围的点称为异常值点，异常值点用 “○” 号表示.

`boxplot()` 函数的使用方法有三种形式，第一种格式为

```
boxplot(x, ...)
```

其中 x 是由数据构成的数值型向量，或者是列表，或者是数据框. 上面例子的使用方法就是这种形式. 第二种形式为

```
boxplot(formula, data = NULL, ..., subset, na.action = NULL)
```

其中 formula 是公式，如 y ~ grp, 这里 y 是由数据构成的数值型向量， grp 是数据的分组，通常是因子. data 是数据结构. 第三种形式为

```
boxplot(x, ..., range = 1.5, width = NULL, varwidth = FALSE,
        notch = FALSE, outline = TRUE, names, plot = TRUE,
        border = par("fg"), col = NULL, log = "",
        pars = list(boxwex = 0.8, staplewex = 0.5, outwex = 0.5),
        horizontal = FALSE, add = FALSE, at = NULL)
```

其中 x 的意义与第一种情况相同. range 是 "触须" 的范围 (缺省值为 1.5). notch 是逻辑变量，当 notch = TRUE(缺省值为 FALSE) 时，画出的箱线图带有切口. outline 是逻辑变量，当 outline = FALSE(缺省值为 TRUE) 时，不标明异常值点. col 是颜色变量，赋给不同的值，将绘出不同颜色的箱线图. horizontal 是逻辑变量，当 horizontal = TRUE(缺省值为 FALSE) 时，将把箱线图绘成水平状. add 是逻辑变量，当 add = TRUE 时，在原图上画图；否则 (FALSE, 缺省值) 替换上一张图. 其余参数的意义在线帮助文件.

可以用 boxplot() 函数作两样本的均值检验，考查两样本的均值是否相同.

例 3.9 已知由两种方法得到如下数据：

```
Method A: 79.98 80.04 80.02 80.04 80.03 80.03 80.04 79.97
          80.05 80.03 80.02 80.00 80.02
Method B: 80.02 79.94 79.98 79.97 79.97 80.03 79.95 79.97
```

问两组数据的均值是否相同？

解：输入数据，调用 boxplot() 函数 (程序名： exam0309.R) 画出两组数据的箱线图，

```
A <- c(79.98, 80.04, 80.02, 80.04, 80.03, 80.03, 80.04,
       79.97, 80.05, 80.03, 80.02, 80.00, 80.02)
B <- c(80.02, 79.94, 79.98, 79.97, 79.97, 80.03, 79.95,
       79.97)
boxplot(A, B, notch=T, names=c('A', 'B'), col=c(2,3))
```

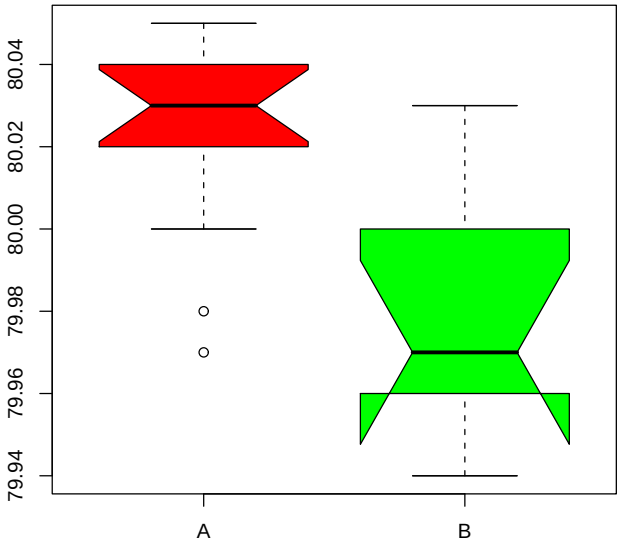


图 3.5: 两组数据的箱线图

得到箱线图，如图 3.5 所示.

从图形可以看出，两组数据的均值是不相同的，第一组值高于第二组. 我们将第五章将给出两样本均值检验的统计方法.

注意到: 由于使用了参数 `notch = T`, 画出的箱线图带有切口. `col = c(2,3)`, 所以关于 A 的箱线图是红色 (2 表示红色), 关于 B 的箱线图是绿色 (3 表示红绿), 也可以将参数写成 `col = c('red', 'green')`.

在 R 软件中， `InsectSprays` 是 R 提供的数据框，它是由两列数据构成，一列叫 count, 由数据构成，另一叫 spray, 由因子构成，共有 A, B, C, D, E, F 六个水平. 现画出数据 count 在这六个水平下的箱线图，其命令 (程序名: `figure0306.R`) 如下:

```
boxplot(count ~ spray, data = InsectSprays,
        col = "lightgray")
boxplot(count ~ spray, data = InsectSprays,
        notch = TRUE, col = 2:7, add = TRUE)
```

第一个命令是画出矩形的箱线图, 而且图中的颜色是青灰色 (`col="lightgray"`). 第二个命令表示画出的箱线图带有切口 (`notch = TRUE`), 而且每一个箱线图用一种颜色 (`col = 2:7`) 画出，并将这次画的图叠加到上一张图上 (`add = TRUE`), 其图形如图 3.6 所示.

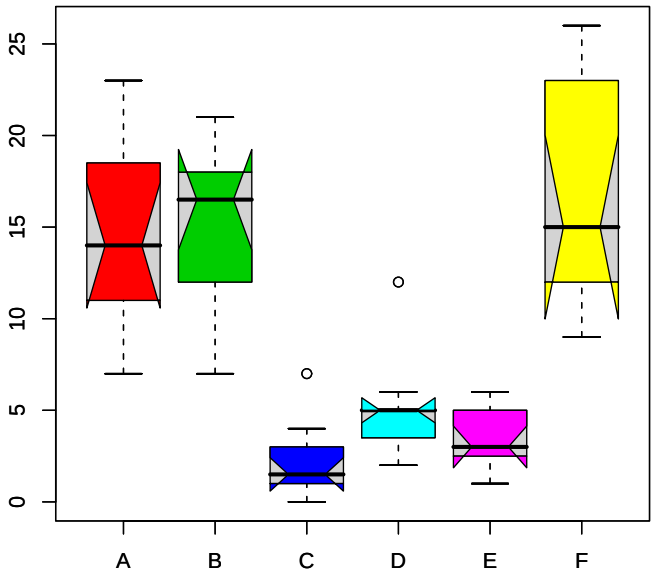


图 3.6: 不同参数下箱线图的叠加

由上述例子可以看出，各种画箱线图的绘图方法可以混合使用.

3. 五数总括

在探索性数据分析中，认为最有代表性的、能反映数据重要特征的五个数：中位数 m_e, 下四分位数 Q_1, 上四分位数 Q_3, 最小值 min 和最大值 max. 这五个数称为样本数据的五数总括.

在 R 软件中，函数 `fivenum()` 计算样本的五数总括. 使用格式为

```
fivenum(x, na.rm = TRUE)
```

其中 `x` 是样本数据， `na.rm` 是逻辑变量，当 `na.rm = TRUE`(缺省值) 时，在计算五数总括之前，所有的 NA 和 NAN 数据将被去掉.

例 3.10 *求例 3.6 学生考试成绩的五数总括.*

解: (程序名： `exam0310.R`)

```
> x<-c(25, 45, 50, 54, 55, 61, 64, 68, 72, 75, 75,
       78, 79, 81, 83, 84, 84, 84, 85, 86, 86, 86,
       87, 89, 89, 89, 90, 91, 91, 92, 100)
> fivenum(x)
[1]  25  70  84  88 100
```

3.2.4 正态性检验与分布拟合检验

上面介绍的茎叶图、箱线图等对随机性、确定性的数据都有用，其特点是图像生动直观. 在直方图、经验分布函数介绍中，曾提到在总体存在某种类型的分布时，配一条合适的总体概率密度曲线或总体分布函数曲线. 然而，所配曲线是否合适，是需要进行统计检验的. 有关的统计检验方法将在第五章中介绍，这里只简单介绍两种检验方法，一种方法是关于正态分布的检验，另一种方法是关于分布函数的拟合检验.

1. 正态性 W 检验方法

利用 Shapiro–Wilk (夏皮罗 – 威尔克) W 统计量作正态性检验，因此称这种检验方法为正态 W 检验方法.

在 R 软件中，函数 `shapiro.test()` 提供 W 统计量和相应的 p 值，当 p 值小于某个显著性水平 α(比如 0.05), 则认为样本为不是来自正态分布的总体；否则承认样本来自正态分布的总体.

函数 `shapiro.test()` 的使用格式为

```
shapiro.test(x)
```

其中 `x` 是由数据构成的向量，并且向量的长度在 3 至 5000 之间.

对于例 3.1 中 15 位学生的体重数据,

```
>  w <- c(75.0, 64.0, 47.4, 66.9, 62.2, 62.2, 58.7, 63.5,
          66.6, 64.0, 57.0, 69.0, 56.9, 50.0, 72.0)
> shapiro.test(w)
        Shapiro-Wilk normality test
data:  w
W = 0.9686, p-value = 0.8371
```

p 值为 $0.8371 > 0.05$, 因此，认为来自正态分布的总体，与 QQ 图得到的结论相同. 又如

```
> shapiro.test(runif(100, min = 2, max = 4))
        Shapiro-Wilk normality test
data:  runif(100, min = 2, max = 4)
W = 0.9493, p-value = 0.0007515
```

p 值为 $0.0007515 < 0.05$, 认为样本不是来自正态分布的总体. 当然, 这是来自均匀分布的随机数.

2. 经验分布的 Kolmogorov-Smirnov 检验方法

经验分布函数 $F_n(x)$ 是总体分布函数 $F(x)$ 的估计. 经验分布拟合检验的方法是检验经验分布 $F_n(x)$ 与假设的总体分布函数 $F_0(x)$ 之间的差异. Kolmogorov-Smirnov (科尔莫戈罗夫 – 斯米尔诺夫) 统计量是计算 $F_n(x)$ 与 $F_0(x)$ 的距离 D, 即

$$D = \sup_{-\infty<x<\infty} |F_n(x) - F_0(x)| . \tag{3.15}$$

在 R 软件中, 函数 `ks.test()` 给出了 Kolmogorov-Smirnov 检验方法, 其使用方法是:

```
ks.test(x, y, ...,
        alternative = c("two.sided", "less", "greater"),
        exact = NULL)
```

其中 `x` 是待检测的样本构成的向量, `y` 是原假设的数据向量或是描述原假设的字符串.

例如,

```
> x<-rt(100,5)
> ks.test(x, "pf",2,5)
        One-sample Kolmogorov-Smirnov test
data:  x
D = 0.5596, p-value < 2.2e-16
alternative hypothesis: two.sided
```

因为 `x` 是来自 t_5 的随机数, 对 `x` 作 $F_{2,5}$ 检验 (即认为是来自总体是自由度为 $(2, 5)$ 的 F 分布), 其结果是拒绝的, 即不认为 `x` 服从 $F_{2,5}$ 的分布.

有关数据分布的检验, 将在第五章有详细的介绍.

3.3 R 软件中的绘图命令

在前面介绍的数据描述性分析中, 数据作图是数据分析的重要方法之一, 因此, 利用绘图的方法研究已知数据, 是一种直观、有效的方法. 这里将介绍 R 软

件中，一些数据作图的基本方法.

在作图函数中，有二类作图函数，一类是高水平作图函数，另一类是低水平作图函数. 所谓高水平作图函数，是与低水平的作图函数相对应的，即所有的绘图函数均可产生图形，可以有坐标轴，以及图和坐标轴的说明文字等. 所谓低水平作图函数是自身无法生成图形，只能在高水平作图函数产生的图形的基础上，增加新的图形.

3.3.1 高水平绘图函数

高水平作图函数有： plot() 、 pairs() 、 coplot() 、 qqnorm() 、 qqline() 、 hist() 和 contour() 等.

1. plot() 函数

函数 plot() 可绘出数据的散点图、曲线图等. plot() 函数有以下四种使用方法.

(1) plot(x, y)

其中 x 和 y 是向量，生成 y 关于 x 的散点图. 例如，第二章中的例 2.2就是这种使用方法.

(2) plot(x)

其中 x 是一时间序列，生成时间序列图形. 如果 x 是向量，则产生 x 关于下标的散点图. 如果 x 是复向量，则绘出复数的实部与虚部的散点图. 第二章的 2.2.6 节介绍了复数绘图的情况.

(3) plot(f)
plot(f, y)

其中 f 是因子， y 是数值向量. 第一种格式生成 f 的直方图；第二种格式生成 y 关于 f 水平的箱线图.

例 3.11 利用四种不同配方的材料 A_1 、 A_2 、 A_3 、 A_4 生产出来的元件，测得其使用寿命如表 *3.2* 所示. 绘出四种不同配方材料寿命的箱线图，并四种不同配方下元件的使用寿命有无显著的差异？

解：使用因子格式输入数据，并绘出相应的箱线图 (程序名： exam0311.R).

```
y<-c(1600, 1610, 1650, 1680, 1700, 1700, 1780, 1500, 1640,
     1400, 1700, 1750, 1640, 1550, 1600, 1620, 1640, 1600,
```

表 3.2: 元件寿命数据

材料	使 用 寿 命							
A_1	1600	1610	1650	1680	1700	1700	1780	
A_2	1500	1640	1400	1700	1750			
A_3	1640	1550	1600	1620	1640	1600	1740	1800
A_4	1510	1520	1530	1570	1640	1600		

```
    1740, 1800, 1510, 1520, 1530, 1570, 1640, 1600)
f<-factor(c(rep(1,7),rep(2,5), rep(3,8), rep(4,6)))
plot(f,y)
```

运行后得到相应寿命的箱线图, 如图 3.7 所示. 从图中可以看出四种不同配方

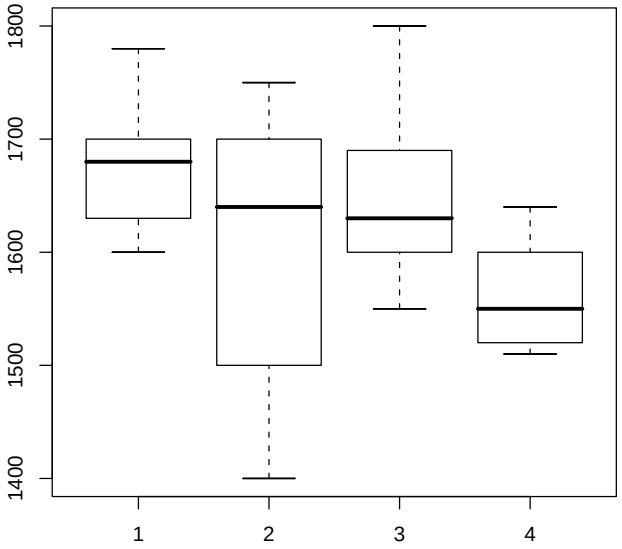


图 3.7: 四种不同配方材料寿命的箱线图

材料寿命没有明显变化.

(4) `plot(df)`

`plot(~ expr)`

`plot(y ~ expr)`

其中 `df` 是数据框, `y` 是任意一个对象, `expr` 是对象名称的表达式如 `(a+b+c)`.

例如输入学生的年龄、身高和体重构成数据框 (文件名: `student_data.R`)

```
df<-data.frame(
   Age=c(13, 13, 14, 12, 12, 15, 11, 15, 14, 14, 14,
         15, 12, 13, 12, 16, 12, 11, 15 ),
   Height=c(56.5, 65.3, 64.3, 56.3, 59.8, 66.5, 51.3,
            62.5, 62.8, 69.0, 63.5, 67.0, 57.3, 62.5,
            59.0, 72.0, 64.8, 57.5, 66.5),
   Weight=c( 84.0,  98.0,  90.0,  77.0,  84.5, 112.0,
             50.5, 112.5, 102.5, 112.5, 102.5, 133.0,
             83.0,  84.0,  99.5, 150.0, 128.0,  85.0,
            112.0))
plot(df)
attach(df)
plot(~Age+Height)
plot(Weight~Age+Height)
```

plot(df) 绘出的图形如图 3.8 所示.

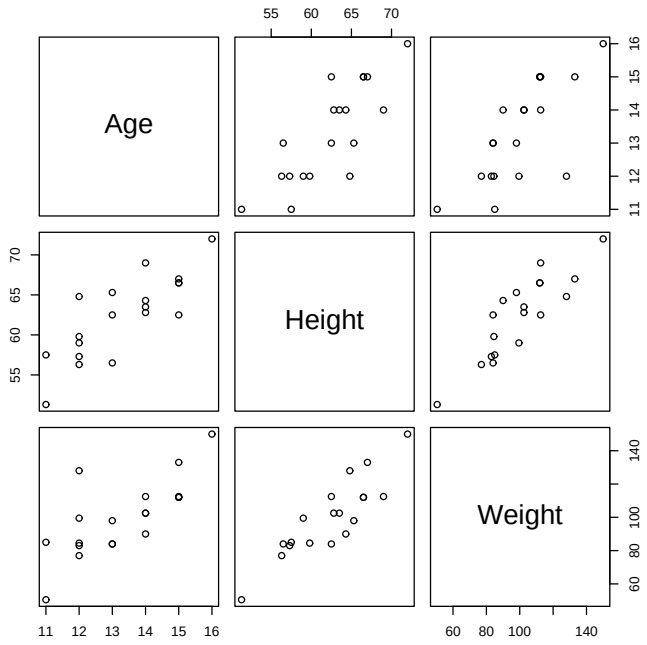


图 3.8: 年龄、身高和体重三项指标构成的散布图

plot(~Age+Height) 绘出身高与年龄的散点图. plot(Weight~Age+Height) 绘出两张散点图，第一张是体重与年龄，第二张是体重与身高.

plot 还可以作回归诊断图等，有些较深入的知识，将随着后面统计知识的深入再介绍.

2. 显示多变量数据

R 软件为显示多变量数据提供了两个非常有用的函数. 一个是 pairs() 函数，当 X 是矩阵或数据框时

```
> pairs(X)
```

绘出关于矩阵各列的散布图. 例如，以学生的数据框为例,

```
> pairs(df)
```

绘出的图形与前面的 plot(df) 相同.

另一方面个函数是 coplot(). 当有三、四个变量时， coplot() 可以将散点图画的更细. 假设 a 和 b 是数值向量，并且 c 是向量或因子 (所有变量具有相同的长度)，则

```
> coplot(a ~ b | c)
```

绘出在给定 c 值下， a 关于 b 的散点图. 仍然以学生的年龄、身高和体重的数据为例,

```
> coplot(Weight ~ Height | Age)
```

绘出了按年龄段给出的体重与身高的散点图，如图 3.9 所示.

对于四个变量 a，b，c，d, 还可以有如下命令;

```
> coplot(a ~ b | c + d)
```

即按 c 、 d 划分下， a 关于 b 的散点图.

3. 显示图形

其他的高水平绘图函数有 qqnorm(), hist(), dotchart(), contour() 等.

(1)
```
qqnorm(x)
qqline(x)
qqplot(x, y)
```

其中 x，y 数值型向量，绘出数据的 QQ 散点图 (已在 3.2.2 节介绍过).

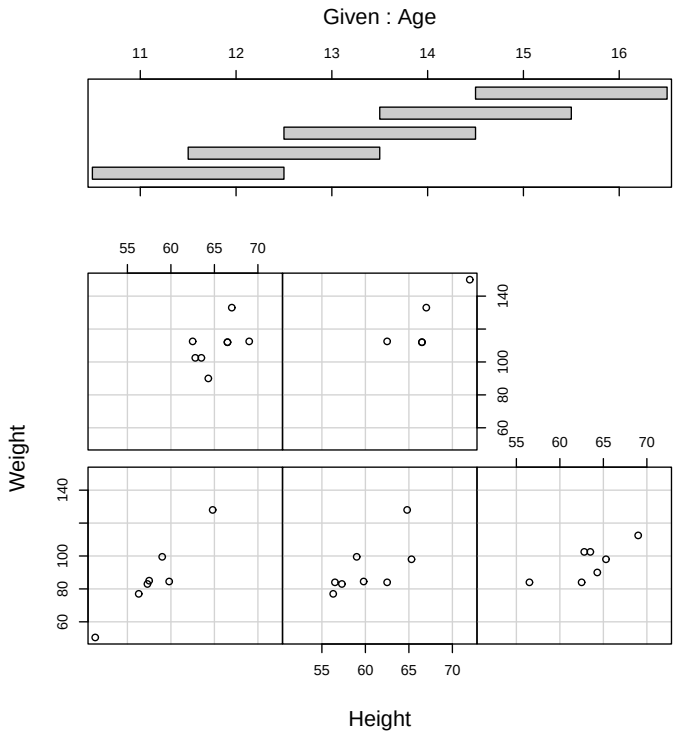


图 3.9: 按年龄划分的体重与身高的散点图

(2) `hist(x)`

`hist(x, nclass=n)`

`hist(x, breaks=b, ...)`

其中 x 数值型向量，绘出数据的直方图 (已在 3.2.2 节介绍过).

(3) `dotchart(x, ...)`

构造数据 x 的点图. 在点图中， y 轴是数据 x 标记， x 轴是数据 x 的数值.

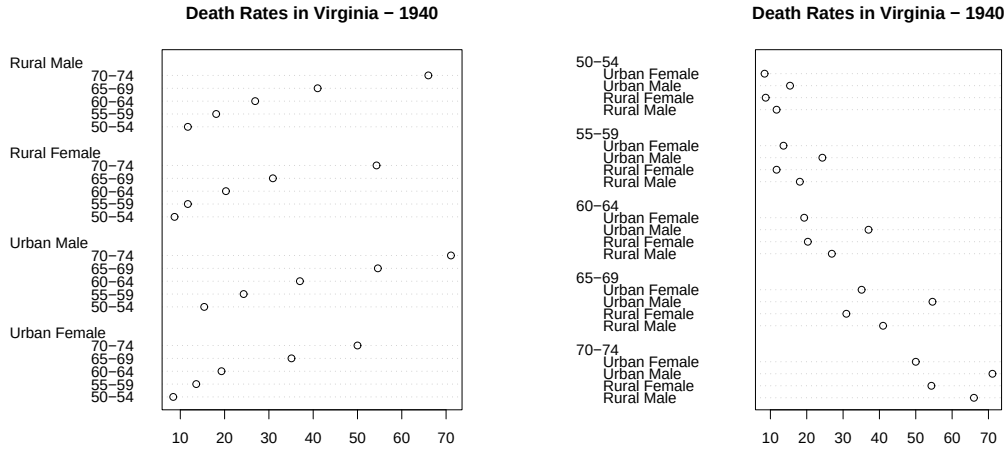
例如， R 软件中，数据 VADeaths 给出了 Virginia (弗吉尼亚) 州在 1940 年的人口死亡率,

```
      Rural Male Rural Female Urban Male Urban Female
50-54       11.7          8.7       15.4          8.4
55-59       18.1         11.7       24.3         13.6
60-64       26.9         20.3       37.0         19.3
65-69       41.0         30.9       54.6         35.1
70-74       66.0         54.3       71.1         50.0
```

我们画出该数据的点图,

```
> dotchart(VADeaths, main = "Death Rates in Virginia - 1940")
> dotchart(t(VADeaths), main = "Death Rates in Virginia - 1940")
```

如图 3.10 所示, 其中 (a) 是第一个命令, (b) 是第二个命令.



(a) (b)

图 3.10: Virginia 州在 1940 年的人口死亡率的点图

(4)
```
image(x, y, z, ...)
contour(x, y, z, ...)
persp(x, y, z, ...)
```

其中 x, y 是数值型向量, z 与 x 和 y 对应的矩阵 (z 的行数是 x 的维数, z 的列数是 y 的维数). image() 绘出三维图形的映象, contour() 绘出三维图形的等值线, persp() 绘出三维图形的表面曲线.

例 3.12 (山区地貌图) 在某山区 (平面区域 $(0, 2800) \times (0, 2400)$ 内, 单位: 米) 测得一些地点的高度 (单位: 米) 如表 *3.3* 所示. 试作出该山区的地貌图和等值线图。

解: 输入数据, 调用 contour() 函数画等值, 调用 persp() 函数画三维图形 (程序名: exam0312.R).

```
x<-seq(0,2800, 400); y<-seq(0,2400,400)
z<-scan()
```

表 3.3: 某山区地形高度数据

2400	1430	1450	1470	1320	1280	1200	1080	940
2000	1450	1480	1500	1550	1510	1430	1300	1200
1600	1460	1500	1550	1600	1550	1600	1600	1600
1200	1370	1500	1200	1100	1550	1600	1550	1380
800	1270	1500	1200	1100	1350	1450	1200	1150
400	1230	1390	1500	1500	1400	900	1100	1060
0	1180	1320	1450	1420	1400	1300	700	900
y/x	0	400	800	1200	1600	2000	2400	2800

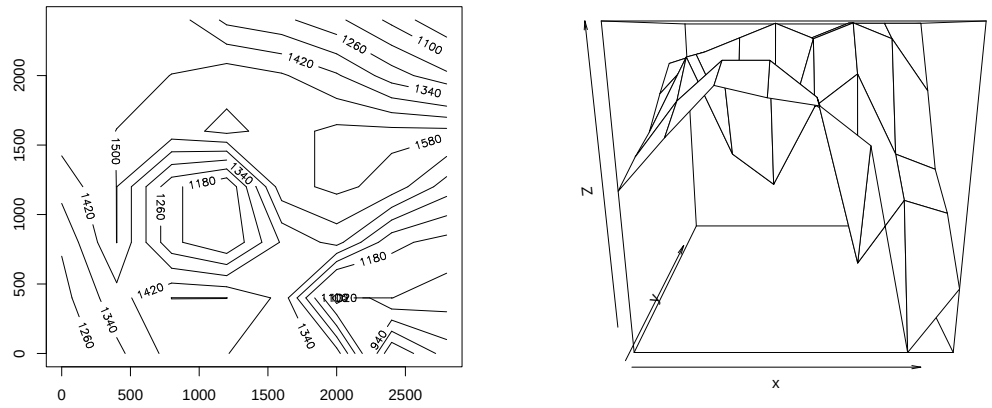
```
1180 1320 1450 1420 1400 1300  700  900
1230 1390 1500 1500 1400  900 1100 1060
1270 1500 1200 1100 1350 1450 1200 1150
1370 1500 1200 1100 1550 1600 1550 1380
1460 1500 1550 1600 1550 1600 1600 1600
1450 1480 1500 1550 1510 1430 1300 1200
1430 1450 1470 1320 1280 1200 1080  940

Z<-matrix(z, nrow=8)
contour(x, y, Z, levels = seq(min(z), max(z), by = 80))
persp(x, y, Z)
```

将绘出两幅图形，一幅是等值线图，如图 3.11(a) 所示，另一幅是三维曲面，如图 3.11(b) 所示.

我们可以看到，图 3.11 有两个缺点，一是过于粗糙，其原因是由于数据量过少造成的，如果数据量稍大一些，图形质量将会有很大的改善；二是三维图的观察角度不理想，这是由于只用到函数中各种参数的缺省值状态，如果改变某些参数的值，图形的观察角度也会随之改变. 例如，将命令改成

```
> persp(x, y, Z,  theta = 30, phi = 45, expand = 0.7)
```

其观察角度将好的多.



(a) 等值线图 (b) 三维曲面图

图 3.11: 三维数据的等值线与网格曲面

例 3.13 *在 $[-2\pi, 2pi] \times [-2\pi, 2\pi]$ 的正方形区域内绘函数 $z = \sin(x)\sin(y)$ 的等值线图和三维曲面图.*

解: 写出相应的 R 程序 (程序名: `exam0313.R`)

```
x<-y<-seq(-2*pi, 2*pi, pi/15)
f<-function(x,y) sin(x)*sin(y)
z<-outer(x, y, f)
contour(x,y,z,col="blue")
persp(x,y,z,theta=30, phi=30, expand=0.7,col="lightblue")
```

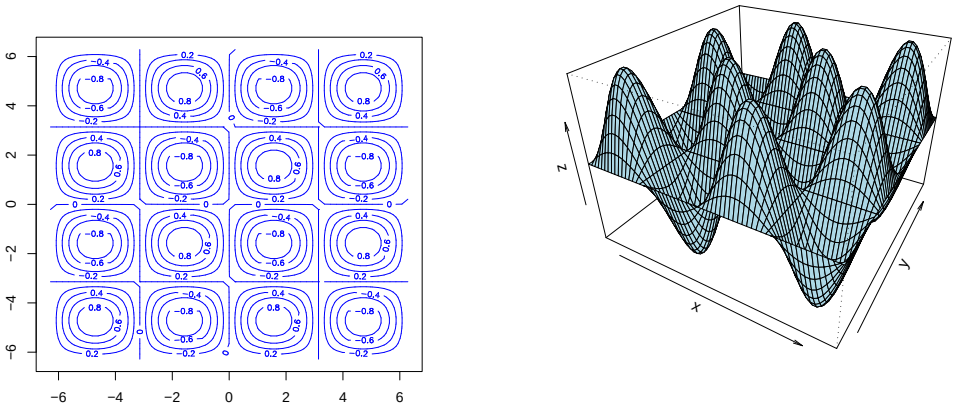
注意: 在绘三维图形时, z 并不是简单地关于 x 与 y 的某些运算, 而是需要在函数 f 关系下作外积运算 (`outer(x, y, f)`), 形成网格, 这样才能绘出三维图形, 请初学者特别注意这一点. 所绘出的图形如图 3.12 所示. 在绘图命令中增加了图形的颜色和观察图形的角度.

3.3.2 高水平绘图中的命令

在高水平绘函数中, 可以加一些命令, 不断完善图的内容, 或增加一些有用的说明.

1. 图中的逻辑命令

`add = TRUE` 表示所绘图在原图上加图, 缺省值为 `add = FALSE`, 即新的图替换原图.



(a) 等值线图　　(b) 三维曲面图

图 3.12: 函数 $z = \sin(x)\sin(y)$ 的等值线与网格曲面

`axes = FALSE` 表示所绘图形没有坐标轴，缺省值为 `axes = TRUE`.

2. 数据取对数

`log = "x"` 表示 x 轴的数据取对数， `log = "y"` 表示 y 轴的数据取对数，`log = "xy"` 表示 x 轴与 y 轴的数据同时取取对数.

3. type 命令

- `type="p"` 绘散点图（缺省值）;
- `type="l"` 绘实线;
- `type="b"` 所有点被实线连接;
- `type="o"` 实线通过所有的点;
- `type="h"` 绘出点到 x 轴的竖线;
- `type="s" or "S"` 绘出阶梯形曲线;
- `type="n"` 不绘任何点或曲线.

4. 图中的字符串

`xlab=` 字符串，其字符串的内容是 x 轴的说明， `ylab=` 字符串，其字符串的内容是 y 轴的说明. `main=` 字符串，其字符串的内容是图的说明，和 sub= 字符串，其字符串的内容是子图的说明.

3.3.3 低水平作图函数

有时高水平的作图函数并不能完全达到作图的指标, 需要低水平的作图函数对图形予以补充. 所有的低水平作图函数所作的图形必须是在高水平作图函数所绘图形的基础之上，增加新的图形.

低水平作图函数有 points()、lines()、text()、abline()、polygon()、legend()、 title() 和 axis() 等.

1. 加点与线的函数

加点函数是 points(), 其作用是在已有图上加点，命令 points(x, y) 其功能相当于 plot(x,y).

加线函数 lines(), 其作用是在已有图上加线，命令 lines(x, y) 其功能相当于 plot(x, y, type="l").

2. 在点处加标记

函数 text() 的作用是在图上加标记，命令格式为:

```
text(x, y, labels, ...)
```

其中 x,y 是数据向量， labels 可以是整数，也可以是字符串. 在缺省状态下, labels=1:length(x). 例如，需要绘出 (x, y) 的散点图，并将所有点用数字标记，其命令为

```
> plot(x, y, type = "n"); text(x, y)
```

3. 在图上加直线

函数 abline() 可以在图上加直线，其使用方法有四种格式.

(1) `abline(a, b)`

表示画一条 $y = a + bx$ 的直线.

(2) `abline(h=y)`

表示画出一条过所有点的水平直线.

(3) `abline(v=x)`

表示画出一条过所有点的竖直直线.

(4) `abline(lm.obj)`

表示绘出线性模型得到的线性方程. 以第二章的例 2.3为例, 说明该命令的用法. 输入命令 (程序名： add_line.R)

```
rt<-read.table("exam0203.txt", head=TRUE);
lm.sol<-lm(Weight~Height, data=rt)
attach(rt)
plot(Weight~Height); abline(lm.sol)
```

得到学生体重与高度的散点图和线性回归直线，如图 3.13 所示.

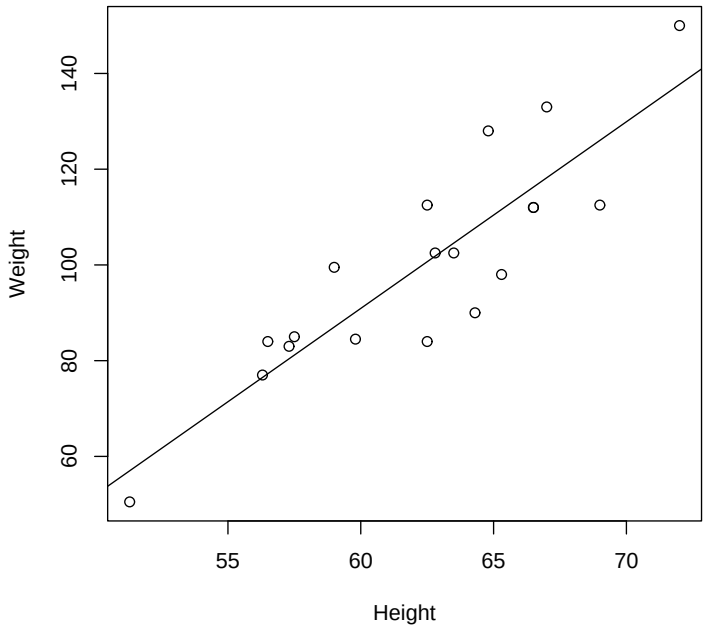


图 3.13: 学生体重与高度的散点图和线性回归直线图

函数 polygon() 可以在图上加多边形，其使用方法为

```
polygon(x, y, ...)
```

以数据的 (x, y) 为坐标，依次连接所有的点，绘出一多边形.

4. 在图上加标记、说明或其他内容

在图上加说明文字、标记或其他内容有两个函数. 一个是加图的题目，用法是

```
title(main="Main Title", sub = "sub title",)
```

其中主题目加在图的项部，子题目加在图的底部.

另一个是在坐标轴上加标记、说明或其他内容，用法是

```
axis(side, ...)
```

其中 side 是边， side=1 表示所加内容放在图的底部， side=2 表示所加内容放在图的左侧， side=3 表示所加内容放在图的顶部， side=4 表示所加内容放在图的右侧.

在 R 软件中，还有其他一些作图函数或作图命令，需要大家在绘图实践中逐步掌握. 在后面的各章中，结合相应的统计知识，还会介绍更加深入的绘图方法.

3.4 多元数据的数据特征与相关分析

在上述各节的分析中，其样本数据基本上是来自一元总体 X, 而在实际情况中，许多数据来自多元数据的总体，即来自总体 $(X_1, X_2, \cdots, X_p)^T$. 对于来自多元总体的数据，除了分析各个分量的取值特点外，更重要的是分析各个分量之间的相关关系，这就是多元数据的相关分析.

3.4.1 二元数据的数字特征及相关系数

设 $(X, Y)^T$ 是二元总体, 从中取得观测样本 $(x_1, y_1)^T$, $(x_2, y_2)^T$, $\cdots$, $(x_n, y_n)^T$. 其样本观测矩阵为

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \end{bmatrix},$$

记

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i,$$

则称 $(\overline{x}, \overline{y})^T$ 为二元观测样本的均值向量. 记

$$\begin{aligned} s_{xx} &= \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2, \\ s_{yy} &= \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \overline{y})^2, \\ s_{xy} &= \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y}), \end{aligned}$$

则称 s_{xx} 为变量 X 的观测样本的方差，称 s_{yy} 为变量 Y 的观测样本的方差，称 s_{xy} 为变量 X, Y 的观测样本的协方差. 称

$$S = \begin{bmatrix} s_{xx} & s_{xy} \\ s_{xy} & s_{yy} \end{bmatrix}$$

为观测样本的协方差矩阵. 称

$$r = \frac{s_{xy}}{\sqrt{s_{xx}}\sqrt{s_{yy}}}$$

为观测样本的相关系数.

在 R 软件中，计算二元样本的均值方差的命令基本上与一元变量的命令相同，有些地方略有一些改动. 计算多元数据的均值与方差采用数据框的结构输入数据，在计算中较为方便，看下面的例子.

例 3.14 *某种矿石有两种有用成分 A, B, 取 10 个样本，每个样本中成分 A 的含量百分数 $x(\%)$ 及 B 的含量百分数 $y(\%)$ 的数据如表 3.4 所示. 计算样本的均值、方差、协方差和相关系数.*

表 3.4: 矿石中有用成分含量的百分数

$x(\%)$	67	54	72	64	39	22	58	43	46	34
$y(\%)$	24	15	23	19	16	11	20	16	17	13

解: 采用数据框方式输入数据, 用 mean() 函数计算均值, 用 cov() 函数计算协方差阵, 用 cor() 函数计算相关矩阵 (相关系数). (程序名： exam0314.R)

```
ore<-data.frame(
     x=c(67, 54, 72, 64, 39, 22, 58, 43, 46, 34),
     y=c(24, 15, 23, 19, 16, 11, 20, 16, 17, 13)
)
ore.m<-mean(ore); ore.s<-cov(ore); ore.r<-cor(ore)
```

显示结果为

```
> ore.m
   x    y
49.9 17.4
```

```
> ore.s
          x        y
x 252.7667 60.60000
y  60.6000 17.15556
> ore.r
           x         y
x 1.0000000 0.9202595
y 0.9202595 1.0000000
```

在上述计算中， var(ore) 得到的计算结果与 cov(ore) 得到的结果相同.

函数 cov() 和 cor() 的使用格式为

```
cov(x, y = NULL, use = "all.obs",
    method = c("pearson", "kendall", "spearman"))
cor(x, y = NULL, use = "all.obs",
    method = c("pearson", "kendall", "spearman"))
```

其中 x 是数值型向量、矩阵或数据框． y 是空值 (NULL, 缺省值) 、向量、矩阵或数据框，但需要与 x 的维数相一致． cov() 的返回值是协方差或协方差矩阵. cor() 的返回值是相关系数或相关矩阵.

与 cov 和 cor 有关的函数还有： cov.wt — 计算加权协方差 (加权协方差矩阵); cor.test — 计算相关性检验.

3.4.2 二元数据的相关性检验

对于一般的检验问题我们将在第五章讨论，这里主要论述二元数据相关性的检验问题.

对于二元数据

$$(x_1, y_1)^T,\ (x_2, y_2)^T,\ \cdots,\ (x_n, y_n)^T,$$

可以计算出样本的相关系数 r_{xy}. 假设样本来自总体 (X, Y), 由第一章的知识可知，总体的相关系数为

$$\rho(X, Y) = \frac{\mathrm{Cov}(X, Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}}.$$

那么样本的相关系数与总体的相关系数有什么关系呢？

可以证明，当样本个数 n 充分大， r_{xy} 可以作为 $\rho(X,Y)$ 的估计，也就是说，当样本个数较大时，样本相关，总体也相关. 但当样本个数较小时，就无法得到相应的结论. 现在的问题是：当样本个数 n 至少取到多少时，样本相关才能保证总体也相关？

Ruben (鲁宾) 给出了总体相关系数的区间估计（一般区间估计的知识将在第四章作详细的介绍）的近似逼近公式. 设 n 是样本个数， r 是样本相关系数， u 是标准正态分布的上 $\alpha/2$ 分位点，即 $u = z_{\alpha/2}$. 则计算

$$r^* = \frac{r}{\sqrt{1-r^2}}, \tag{3.16}$$

$$a = 2n-3-u^2, \tag{3.17}$$

$$b = r^*\sqrt{(2n-3)(2n-5)}, \tag{3.18}$$

$$c = (2n-5-u^2)r^{*2} - 2u^2. \tag{3.19}$$

求方程 $ay^2 - 2by + c = 0$ 的根

$$y_1 = \frac{b-\sqrt{b^2-ac}}{a}, \quad y_2 = \frac{b+\sqrt{b^2-ac}}{a}, \tag{3.20}$$

则 $1-\alpha$ 的双侧置信区间为

$$L = \frac{y_1}{\sqrt{1+y_1^2}}, \quad U = \frac{y_2}{\sqrt{1+y_2^2}}. \tag{3.21}$$

按照公式 (3.16)-(3.21) 编写出 R 程序 (程序名： `ruben.R`)

```
ruben.test <- function(n, r, alpha=0.05){
   u <- qnorm(1-alpha/2)
   r_star <- r/sqrt(1-r^2)
   a <- 2*n-3-u^2
   b <- r_star*sqrt((2*n-3)*(2*n-5))
   c <- (2*n-5-u^2)*r_star^2-2*u^2
   y1 <- (b-sqrt(b^2-a*c))/a
   y2 <- (b+sqrt(b^2-a*c))/a
   data.frame(n = n, r = r, conf = 1-alpha,
      L = y1/sqrt(1+y1^2), U = y2/sqrt(1+y2^2))
}
```

当 $n=6, r=0.8$ 时，调入已编好的函数 ruben.test(), 并计算得到

```
> source("ruben.test.R")
> ruben.test(6, 0.8)
  n   r conf           L         U
1 6 0.8 0.95 -0.09503772 0.9727884
```

置信区间为 $(-0.095, 0.97)$, 其置信下界是负数，即使 $r=0.8$, 也不能说明总体是相关的.

考虑 $n=25, r=0.7$, 计算得到

```
> ruben.test(25, 0.7)
   n   r conf         L         U
1 25 0.7 0.95 0.4108176 0.8535657
```

置信区间为 $(0.41, 0.85)$, 此时，基本上能说总体是相关的.

关于置信区间的近似逼近方法还有 David (大卫， 1954) 提出的图表方法, Kendall (肯德尔) 和 Stuart (斯图亚特， 1961) 提出的 Fisher 逼近方法等.

确认总体是否相关最有效的方法是作总体 $(X,Y)^T$ 的相关性检验.

可以证明，当 $(X,Y)^T$ 是二元正态总体，且 $\rho(X,Y)=0$, 则统计量

$$t=\frac{r_{xy}\sqrt{n-2}}{\sqrt{1-r_{xy}^2}} \tag{3.22}$$

服从自由度为 $n-2$ 的 t 分布.

利用统计量 t 服从自由度为 $n-2$ 的 t 分布的性质，可以对数据 X 和 Y 的相关性进行检验. 由于相关系数 r_{xy} 被称为 Pearson (皮尔森) 相关系数，因此，此检验方法也称为 Pearson 相关性检验.

对于相关性检验，还有 Spearman 秩检验和 Kendall 秩检验，这里只介绍用 R 软件进行检验的方法，有关检验原理请读者参看有关的数理统计教材.

在 R 软件中， cor.test() 提供了上述三种检验方法. 其使用方法是:

```
cor.test(x, y,
  alternative = c("two.sided", "less", "greater"),
  method = c("pearson", "kendall", "spearman"),
  exact = NULL, conf.level = 0.95, ...)
```

其中 x，y 是数据长度相同的向量， alternative 是备择假设 (有关概念将在第五章中详细介绍)，缺省值为 "two.sided", method 是选择的检验方法，缺省值为 Pearson 检验． conf.level 是置信区间水平，缺省值为 0.95.

cor.test() 函数还有另一种使用格式

```
cor.test(formula, data, subset, na.action, ...)
```

其中 formula 是公式，形如 '~u+v', 'u', 'v' 必须是具有相同长度的数值向量． data 是数据框． subset 是可选择向量，表示观察值的子集.

例 3.15 *对例 3.14 的两组数据进行相关性检验.*

解:

```
> attach(ore)
> cor.test(x,y)
        Pearson's product-moment correlation
data:  x and y
t = 6.6518, df = 8, p-value = 0.0001605
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6910290 0.9813009
sample estimates:
      cor
0.9202595
```

其 p 值为 $0.0001605 < 0.05$, 拒绝原假设，认为变量 X 与 Y 相关.

实际上， cor.test() 也提供了相关系数的区间估计，这里计算的区间是 $(0.69, 0.98)$, 因此从这一点也可看出变量 X 与 Y 是相关的.

另外可用

```
cor.test(x,y, method="spearman")
cor.test(x,y, method="kendall")
```

命令作另外两种检验.

3.4.3 多元数据的数字特征及相关矩阵

对于 p 元总体 $(X_1, X_2, \cdots, X_n)$, 其样本为

$$(x_{11}, x_{12}, \cdots, x_{1p})^T, (x_{21}, x_{22}, \cdots, x_{2p})^T, \cdots, (x_{n1}, x_{n2}, \cdots, x_{np})^T,$$

其中第 i 本样本为

$$(x_{i1}, x_{i2}, \cdots, x_{ip})^T, \quad i = 1, 2, \cdots, n.$$

样本的第 j 个分量的均值定义为

$$\overline{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}, \quad j = 1, 2, \cdots, p. \tag{3.23}$$

样本的第 j 个分量的方差定义为

$$s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2, \quad j = 1, 2, \cdots, p. \tag{3.24}$$

样本的第 j 个分量与第 k 个分量的协方差定义为

$$s_{jk} = \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)(x_{ik} - \overline{x}_k), \quad j, k = 1, 2, \cdots, p. \tag{3.25}$$

称 $\overline{x} = (\overline{x}_1, \overline{x}_2, \cdots, \overline{x}_p)^T$ 为 p 元样本的均值，称

$$S = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1p} \\ s_{21} & s_{22} & \cdots & s_{2p} \\ \vdots & \vdots & & \vdots \\ s_{p1} & s_{p2} & \cdots & s_{pp} \end{bmatrix} \tag{3.26}$$

为样本的协方差矩阵.

样本的第 j 个分量与第 k 个分量的相关系数定义为

$$r_{jk} = \frac{s_{jk}}{\sqrt{s_{jj}}\sqrt{s_{kk}}}, \quad j, k = 1, 2, \cdots, p. \tag{3.27}$$

称

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1p} \\ r_{21} & r_{22} & \cdots & r_{2p} \\ \vdots & \vdots & & \vdots \\ r_{p1} & r_{p2} & \cdots & r_{pp} \end{bmatrix} \tag{3.28}$$

为样本的相关矩阵（ Pearson 相关矩阵）.

对于多元数据，与二元数据相同，采用数据框的输入方式，可以用 `mean()` 函数、 `cov()` 函数和 `cor()` 函数计算样本的均值、协方差阵和相关矩阵.

关于相关性检验，R 软件没有为多元数据提供更多的函数，仍是 `cor.test()` 作两两分量的相关性检验.

例 3.16 为了解某种橡胶的性能，今抽取 *10* 个样品，每个测量三项指标：硬度、变形和弹性，其数据如表 *3.5* 所示． 试计算样本均值、样本协方差阵和样本相关矩阵．并用 *Pearson* 相关性检验确认变量 X_1, X_2, X_3 是否相关？

表 3.5: 橡胶的三项指标

序号	硬度 (X_1)	变形 (X_2)	弹性 (X_3)
1	65	45	27.6
2	70	45	30.7
3	70	48	31.8
4	69	46	32.6
5	66	50	31.0
6	67	46	31.3
7	68	47	37.0
8	72	43	33.6
9	66	47	33.1
10	68	48	34.2

解：建立数据文件 (文件名： `rubber.data`), 其格式为

```
   X1  X2   X3
1  65  45 27.6
2  70  45 30.7
3  70  48 31.8
4  69  46 32.6
5  66  50 31.0
6  67  46 31.3
7  68  47 37.0
8  72  43 33.6
9  66  47 33.1
10 68  48 34.2
```

读数据，并计算均值、协方差阵和相关矩阵

```
> rubber<-read.table("rubber.data")
```

```
> mean(rubber)
   X1    X2    X3
68.10 46.50 32.29
> cov(rubber)
          X1         X2        X3
X1  4.766667 -1.9444444 1.9344444
X2 -1.944444  3.8333333 0.6166667
X3  1.934444  0.6166667 6.1898889
> cor(rubber)
           X1         X2        X3
X1  1.0000000 -0.4548832 0.3561291
X2 -0.4548832  1.0000000 0.1265962
X3  0.3561291  0.1265962 1.0000000
```

再作相关性检验

```
> cor.test(~X1+X2, data=rubber)
        Pearson's product-moment correlation
data:  X1 and X2
t = -1.4447, df = 8, p-value = 0.1865
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.8430535  0.2448777
sample estimates:
       cor
-0.4548832

> cor.test(~X1+X3, data=rubber)
        Pearson's product-moment correlation
data:  X1 and X3
t = 1.078, df = 8, p-value = 0.3125
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
```

```
 -0.3525486  0.8052056
sample estimates:
      cor
0.3561291

> cor.test(~X2+X3, data=rubber)
        Pearson's product-moment correlation
data:  X2 and X3
t = 0.361, df = 8, p-value = 0.7275
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.5465985  0.7003952
sample estimates:
      cor
0.1265962
```

从上述计算结果可以看出，只能认为 X_1, X_2, X_3 两两均是不相关的.

3.4.4 基于相关系数的变量分类

本小节以一个例子说明相关系数的应用 — 基于相关系数的变量分类.

例 3.17 现有 *48* 位应聘者应聘某公司的某职位，公司为这些应聘者的 *15* 项指标打分，这 *15* 项指标分别是：求职信的形式 *(FL)* 、外貌 *(APP)* 、专业能力 *(AA)* 、讨人喜欢 *(LA)* 、自信心 *(SC)* 、洞察力 *(LC)* 、诚实 *(HON)* 、推销能力 *(SMS)* 、经验 *(EXP)* 、驾驶水平 *(DRV)* 、事业心 *(AMB)* 、理解能力 *(GSP)* 、潜在能力 *(POT)* 、交际能力 *(KJ)* 和适应性 *(SUIT)*. 每项分数是从 *0* 分到 *10* 分， *0* 分最低， *10* 分最高. 每位求职者的 *15* 项指标列在表 *3.6* 中. 公司计划录用 *6* 名最优秀的申请者，问公司将如何挑选这些应聘者？

解：通常的作法是：作 15 项指标的平均值

$$\mathrm{AVG} = (\mathrm{FL} + \mathrm{APP} + \cdots + \mathrm{SUIT})/15,$$

录用分数最高的 6 名应聘者.

录入数据（文件名： `applicant.data` ）

表 3.6: 48 名应聘者的得分情况

ID	FL	APP	AA	LA	SC	LC	HON	SMS	EXP	DRV	AMB	GSP	POT	KJ	SUIT
1	6	7	2	5	8	7	8	8	3	8	9	7	5	7	10
2	9	10	5	8	10	9	9	10	5	9	9	8	8	8	10
3	7	8	3	6	9	8	9	7	4	9	9	8	6	8	10
4	5	6	8	5	6	5	9	2	8	4	5	8	7	6	5
5	6	8	8	8	4	4	9	5	8	5	5	8	8	7	7
6	7	7	7	6	8	7	10	5	9	6	5	8	6	6	6
7	9	9	8	8	8	8	8	8	10	8	10	8	9	8	10
8	9	9	9	8	9	9	8	8	10	9	10	9	9	9	10
9	9	9	7	8	8	8	8	5	9	8	9	8	8	8	10
10	4	7	10	2	10	10	7	10	3	10	10	10	9	3	10
11	4	7	10	0	10	8	3	9	5	9	10	8	10	2	5
12	4	7	10	4	10	10	7	8	2	8	8	10	10	3	7
13	6	9	8	10	5	4	9	4	4	4	5	4	7	6	8
14	8	9	8	9	6	3	8	2	5	2	6	6	7	5	6
15	4	8	8	7	5	4	10	2	7	5	3	6	6	4	6
16	6	9	6	7	8	9	8	9	8	8	7	6	8	6	10
17	8	7	7	7	9	5	8	6	6	7	8	6	6	7	8
18	6	8	8	4	8	8	6	4	3	3	6	7	2	6	4
19	6	7	8	4	7	8	5	4	4	2	6	8	3	5	4
20	4	8	7	8	8	9	10	5	2	6	7	9	8	8	9
21	3	8	6	8	8	8	10	5	3	6	7	8	8	5	8
22	9	8	7	8	9	10	10	10	3	10	8	10	8	10	8
23	7	10	7	9	9	9	10	10	3	9	9	10	9	10	8
24	9	8	7	10	8	10	10	10	2	9	7	9	9	10	8

表 3.6 续： 48 名应聘者的得分情况

ID	FL	APP	AA	LA	SC	LC	HON	SMS	EXP	DRV	AMB	GSP	POT	KJ	SUIT
25	6	9	7	7	4	5	9	3	2	4	4	4	4	5	4
26	7	8	7	8	5	4	8	2	3	4	5	6	5	5	6
27	2	10	7	9	8	9	10	5	3	5	6	7	6	4	5
28	6	3	5	3	5	3	5	0	0	3	3	0	0	5	0
29	4	3	4	3	3	0	0	0	0	4	4	0	0	5	0
30	4	6	5	6	9	4	10	3	1	3	3	2	2	7	3
31	5	5	4	7	8	4	10	3	2	5	5	3	4	8	3
32	3	3	5	7	7	9	10	3	2	5	3	7	5	5	2
33	2	3	5	7	7	9	10	3	2	2	3	6	4	5	2
34	3	4	6	4	3	3	8	1	1	3	3	3	2	5	2
35	6	7	4	3	3	0	9	0	1	0	2	3	1	5	3
36	9	8	5	5	6	6	8	2	2	2	4	5	6	6	3
37	4	9	6	4	10	8	8	9	1	3	9	7	5	3	2
38	4	9	6	6	9	9	7	9	1	2	10	8	5	5	2
39	10	6	9	10	9	10	10	10	10	10	8	10	10	10	10
40	10	6	9	10	9	10	10	10	10	10	10	10	10	10	10
41	10	7	8	0	2	1	2	0	10	2	0	3	0	0	10
42	10	3	8	0	1	1	0	0	10	0	0	0	0	0	10
43	3	4	9	8	2	4	5	3	6	2	1	3	3	3	8
44	7	7	7	6	9	8	8	6	8	8	10	8	8	6	5
45	9	6	10	9	7	7	10	2	1	5	5	7	8	4	5
46	9	8	10	10	7	9	10	3	1	5	7	9	9	4	4
47	0	7	10	3	5	0	10	0	0	2	2	0	0	0	0
48	0	6	10	1	5	0	10	0	0	2	2	0	0	0	0

```
   FL APP AA LA SC LC HON SMS EXP DRV AMB GSP POT KJ SUIT
1   6   7  2  5  8  7   8   8   3   8   9   7   5  7   10
2   9  10  5  8 10  9   9  10   5   9   9   8   8  8   10
3   7   8  3  6  9  8   9   7   4   9   9   8   6  8   10
.   .   .  .  .  .  .   .   .   .   .   .   .   .  .    .
.   .   .  .  .  .  .   .   .   .   .   .   .   .  .    .
```

读数据，计算各应聘者的平均得分，再将平均得分排序 (由大到小), 得到

```
> rt <- read.table("applicant.data")
> AVG <- apply(rt, 1, mean)
> sort(AVG, decreasing = TRUE)
      40       39        8        7       23       22        2
9.600000 9.466667 9.000000 8.600000 8.600000 8.533333 8.466667
      24        9       10       16        3       44       12
8.400000 8.133333 7.666667 7.666667 7.400000 7.400000 7.200000
       .        .        .        .        .        .        .
       .        .        .        .        .        .        .
```

这样得到前 6 名应聘者是： 40 、 39 、 8 、 7 、 23 和 22 号.

将上述语句中的 mean 改为 sum, 即求应聘者的总得分，其选择结果是相同的.

显然，上述作法认为每项指标的权重是相同的. 当然，也可以按加权平均值

$$\text{WTD\_AVG} = w_1\text{FL} + w_2\text{APP} + \cdots + w_{15}\text{SUIT},$$

其中 $w_1, w_2, \cdots, w_{15}$ 是权值，满足 $w_1 + w_2 + \cdots + w_{15} = 1$. w_i $(i = 1, 2, \cdots, 15)$ 表示第 i 项指标的重要性. 这里需要确定每项指标的权重.

上述两种方法有它的缺点，因为有些指标是相关的，而有些指标不相关，只作简单的平均计算，实际上，相关类多的项占的权重大，而相关类少的项占的权重小. 因此，在作评分前，应先作相关性分析.

作数据的相关性计算，计算相关矩阵

```
> cor(rt)
```

	FL	APP	AA	LA	SC
FL	1.00000000	0.2388057	0.044040889	0.306313037	0.092144656
APP	0.23880573	1.0000000	0.123419296	0.379614151	0.430769427
AA	0.04404089	0.1234193	1.000000000	0.001589766	0.001106763
LA	0.30631304	0.3796142	0.001589766	1.000000000	0.302439887
SC	0.09214466	0.4307694	0.001106763	0.302439887	1.000000000
LC	0.22843205	0.3712589	0.076824494	0.482774928	0.807545017
HON	-0.10674947	0.3536910	-0.030269601	0.645408595	0.410090809
SMS	0.27069919	0.4895490	0.054727421	0.361643880	0.799630538
EXP	0.54837963	0.1409249	0.265585352	0.140723415	0.015125832
DRV	0.34557633	0.3405493	0.093522030	0.393164148	0.704340067
AMB	0.28464484	0.5496359	0.044065981	0.346555034	0.842122228
GSP	0.33820196	0.5062987	0.197504552	0.502809305	0.721108973
POT	0.36745292	0.5073769	0.290032151	0.605507554	0.671821239
KJ	0.46720619	0.2840928	-0.323319352	0.685155768	0.482455962
SUIT	0.58591822	0.3842084	0.140017368	0.326957419	0.250283416

	LC	HON	SMS	EXP	DRV
FL	0.2284320	-0.106749472	0.27069919	0.54837963	0.34557633
APP	0.3712589	0.353690969	0.48954902	0.14092491	0.34054927
AA	0.0768245	-0.030269601	0.05472742	0.26558535	0.09352203
LA	0.4827749	0.645408595	0.36164388	0.14072342	0.39316415
SC	0.8075450	0.410090809	0.79963054	0.01512583	0.70434007
LC	1.0000000	0.355844464	0.81802080	0.14720197	0.69751518
HON	0.3558445	1.000000000	0.23990754	-0.15593849	0.28018499
SMS	0.8180208	0.239907539	1.00000000	0.25541758	0.81473421
EXP	0.1472020	-0.155938495	0.25541758	1.00000000	0.33722821

DRV	0.6975152	0.280184989	0.81473421	0.33722821	1.00000000
AMB	0.7575421	0.214606359	0.85952656	0.19548192	0.78032317
GSP	0.8828486	0.385821758	0.78212322	0.29926823	0.71407319
POT	0.7773162	0.415657447	0.75360983	0.34833878	0.78840024
KJ	0.5268356	0.448245522	0.56328419	0.21495316	0.61280767
SUIT	0.4161447	0.002755617	0.55803585	0.69263617	0.62255406

	AMB	GSP	POT	KJ	SUIT
FL	0.28464484	0.3382020	0.3674529	0.4672062	0.585918216
APP	0.54963595	0.5062987	0.5073769	0.2840928	0.384208365
AA	0.04406598	0.1975046	0.2900322	-0.3233194	0.140017368
LA	0.34655503	0.5028093	0.6055076	0.6851558	0.326957419
SC	0.84212223	0.7211090	0.6718212	0.4824560	0.250283416
LC	0.75754208	0.8828486	0.7773162	0.5268356	0.416144671
HON	0.21460636	0.3858218	0.4156574	0.4482455	0.002755617
SMS	0.85952656	0.7821232	0.7536098	0.5632842	0.558035847
EXP	0.19548192	0.2992682	0.3483388	0.2149532	0.692636173
DRV	0.78032317	0.7140732	0.7884002	0.6128077	0.622554062
AMB	1.00000000	0.7838707	0.7688695	0.5471256	0.434768242
GSP	0.78387073	1.0000000	0.8758309	0.5494076	0.527816315
POT	0.76886954	0.8758309	1.0000000	0.5393968	0.573873154
KJ	0.54712558	0.5494076	0.5393968	1.0000000	0.395798842
SUIT	0.43476824	0.5278163	0.5738732	0.3957988	1.000000000

为了便于选择哪些变量是相关的,将上述相关矩阵中相关系数的绝对值 ≥ 0.5 的值画上下划线.

下面将变量分组，分组的原则是：同一组中变量之间的相关系数尽可能的高，而不同组间的相关系数尽可能的低. 从相关系数最大的变量开始， LC(洞察力) 与 GSP(理解能力) 的相关系数是 0.882, GSP 与 POT(潜在能力) 的相关系数

是 0.876, 而 LC 与 POT 之间的相关系数是 0.777, 因此，这三个变量可以看成一组. SMS(推销能力) 也应该包含在这组中，因为它与 LC 、 GSP 和 POT 的相关系数分别是： 0.818 、 0.782 和 0.754. AMB(事业心) 也应在此组中，其相关系数分别是： 0.758 、 0.860 、 0.784 和 0.769. 进一步研究，发现变量 DRV(驾驶水平) 和 SC(自信心) 也就在此组中. 此组中各个变量的相关系数至少在 0.672 以上.

在选择第二组的变量，按照同样的原理选择 FL(求职信的形式) 、 EXP(经验) 和 SUIT(适应性), 其相关系数分别是： 0.548 、 0.586 和 0.693.

第三组先选择 KJ(交际能力) 、 LA(讨人喜欢), 相关系数是 0.685, 现选择 HON(诚实), 它与 LA 的相关系数是 0.645, 但它与 KJ 的相关系数只有 0.448. 由于全部数据均来自 “人” 的打分， HON 变量分在此组也可以认为是合理的.

再看 AA(专业能力) 、 APP(外貌) 两个变量. AA 变量与其他变量的相关系数没有超过 0.5, 而 APP 变量与其他变量的相关系数虽然刚刚超过 0.5 的，但低其他组内的相关系数.

最后得到五个组：

组 1： SC, LC, SMS, DRV, AMB, GSP 和 POT

组 2： FL, EXP 和 SUIT

组 3： LA, HON 和 KJ

组 4： AA

组 5： APP

由于每一组的指标基本上代表了同一组能力，因此，我们先得到各组的得分，即

$$\begin{aligned} G_1 &= (\mathtt{SC}+\mathtt{LC}+\mathtt{SMS}+\mathtt{DRV}+\mathtt{AMB}+\mathtt{GSP}+\mathtt{POT})/7 \\ G_2 &= (\mathtt{FL}+\mathtt{EXP}+\mathtt{SUIT})/3 \\ G_3 &= (\mathtt{LA}+\mathtt{HON}+\mathtt{KJ})/3 \\ G_4 &= \mathtt{AA} \\ G_5 &= \mathtt{APP} \end{aligned}$$

最后，每位申请者的得分是：

$$\mathrm{AVG} = (G_1+G_2+G_3+G_4+G_5)/5.$$

编写相应的 R 程序 (程序名： group.R), 计算得到

```
> attach(rt)
> rt$G1<-(SC+LC+SMS+DRV+AMB+GSP+POT)/7
> rt$G2<-(FL+EXP+SUIT)/3
> rt$G3<-(LA+HON+KJ)/3
> rt$G4<-AA
> rt$G5<-APP
> AVG<-apply(rt[,16:20], 1, mean)
> sort(AVG, decreasing = TRUE)
       8       40       39        7       23        9        2
9.000000 8.971429 8.914286 8.619048 8.390476 8.209524 8.066667
      22       24       16       46        5       10       20
8.057143 8.038095 7.571429 7.533333 7.314286 7.304762 7.219048
       .        .        .        .        .        .        .
       .        .        .        .        .        .        .
       .        .        .        .        .        .        .
```

在分组情况下，前 6 名应聘者是： 8 、 40 、 39 、 7 、 23 和 9 号.

或计算分组情况下的加权平均分

$$\text{WTD\_AVG} = w_1G_1 + w_2G_2 + \cdots + w_5G_5,$$

其中 $w_1 + w_2 + \cdots + w_5 = 1$.

3.5 多元数据的图表示方法

在前面介绍了许多数据的图形表示方法，但大多数是针对一、二元数据的，三维图形虽然能画出来，但并不方便. 对于三维以上数据如何来描述呢？这是本节要讨论的问题. 许多统计学家给出了多种多元数据的图示方法，但这方面的研究还处于不成熟的状态，目前尚未有公认的方法. 这里结合 R 软件的特点，介绍几种多元数据的图示方法.

设变量是 p 维数据，有 n 个观测数据，其中第 k 次的观测值为

$$X_k = (x_{k1}, x_{k2}, \cdots, x_{kp}), \quad k = 1, 2, \cdots, n,$$

n 次观测数据组成矩阵 $X = (x_{ij})_{n\times p}$.

3.5.1 轮廓图

轮廓图由以下作图步骤完成：

(1) 作直角坐标系，横坐标取 p 个点，以表示 p 个变量；

(2) 对给定的一次观测值，在 p 个点上的纵坐标（即高度）与对应的变量取值成正比；

(3) 连结此 p 个点得一折线，即为该次观测值的一格轮廓线；

(4) 对于 n 次观测值，每次都重复上述步骤，可画出 n 条折线，构成 n 次观测值的轮廓图.

编写画轮廓画函数（函数名： outline.R ）

```
outline <- function(x, txt = TRUE){
   if (is.data.frame(x) == TRUE)
      x <- as.matrix(x)
   m <- nrow(x); n <- ncol(x)
   plot(c(1,n), c(min(x),max(x)), type = "n",
        main = "The outline graph of Data",
        xlab = "Number", ylab = "Value")
   for(i in 1:m){
      lines(x[i,], col=i)
      if (txt == TRUE){
         k <- dimnames(x)[[1]][i]
         text(1+(i-1)%%n, x[i,1+(i-1)%%n], k)
      }
   }
}
```

其中 x 是矩阵或数据框. txt 是逻辑变量，当 txt = TRUE(缺省值) 时，绘图时给出观测值的标号；否则 (FALSE) 不给出标号. 函数的运行结果是绘出 n 次观测值的轮廓图.

例 3.18 *为考查学生的学习情况，学校随机的抽取 12 名学生的 5 门课期末考试的成绩，如表 3.7 所示. 画出 12 名学生学习成绩的轮廓图.*

表 3.7: 12 名学生 5 门课程的考试成绩

序号	政治 (X_1)	语文 (X_2)	外语 (X_3)	数学 (X_4)	物理 (X_5)
1	99	94	93	100	100
2	99	88	96	99	97
3	100	98	81	96	100
4	93	88	88	99	96
5	100	91	72	96	78
6	90	78	82	75	97
7	75	73	88	97	89
8	93	84	83	68	88
9	87	73	60	76	84
10	95	82	90	62	39
11	76	72	43	67	78
12	85	75	50	34	37

解: 将数据输入到数据文件中 (文件名: `course.data`), 其格式为

```
    X1  X2  X3  X4  X5
1   99  94  93 100 100
2   99  88  96  99  97
3  100  98  81  96 100
.   ..  ..  ..  ..  ..
```

读数据, 利用编写的 `outline()` 函数

```
> X<-read.table("course.data")
> source("outline.R")
> outline(X)
```

绘出数据的轮廓图, 如图 3.14 所示.

由轮廓图 (图 3.14) 可以直观的看出, 哪个学生成绩相似、哪些属于优秀、哪些中等、哪些较差; 对各门课程而言, 也可直观地看出各课程成绩的好坏和分散情况等等. 这种图形在聚类分析中颇有帮助.

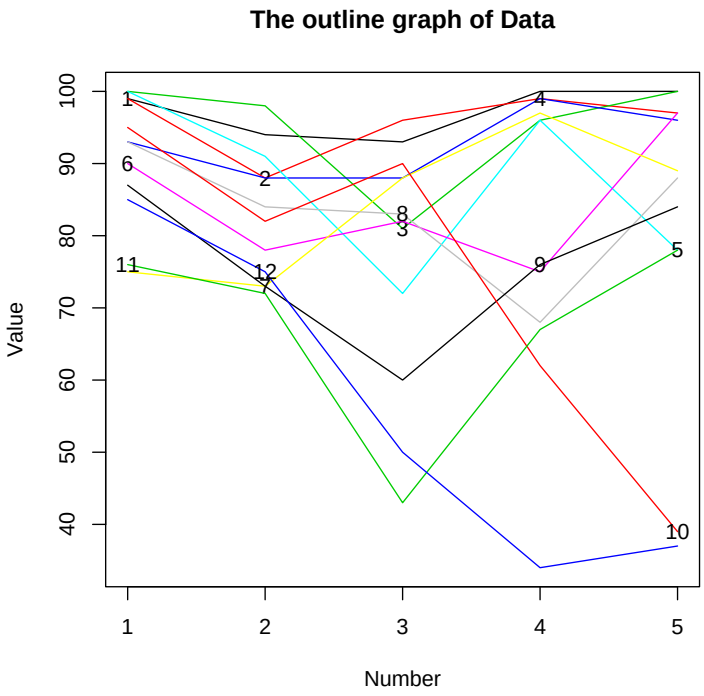


图 3.14: 12 名学生 5 门课程的考试成绩的轮廓图

3.5.2 星图

星图的作图步骤是:

(1) 作一圆，并将圆周 p 等分;

(2) 连结圆心和各分点，把这 p 条半径依次定义为变量的坐标轴，并标以适当的刻度;

(3) 对给定的一次观测值，把 p 个变量值分别取在相应的坐标轴上，然后将它们连结成一个 p 边形;

(4) n 次观测值可画出 n 个 p 边形.

R 软件包给出作星图的函数 `stars()`, 例如，画出例 3.18 中 12 名学生学习成绩的星图，只需

```
> stars(X)
```

就可画出星图，如图 3.15 所示.

星图中水平轴是变量 X_1, 沿逆时针方向，依次是 $X_2, X_3, \cdots$. 由于星图既像雷达屏幕上看到的图像，也像一个蜘蛛网，因此，星图也称为雷达图或蜘蛛图.

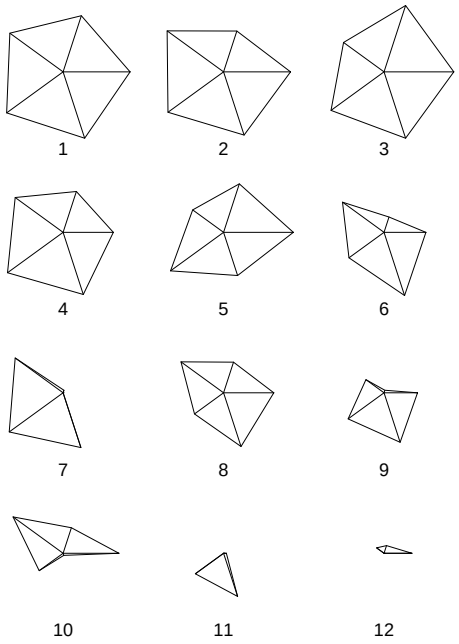


图 3.15: 12 名学生 5 门课程的考试成绩的星图

从图 3.15 中可以看出 1 、 2 号学生学习成绩优秀， 11 、 12 号学生学习成绩较差，而 7 、 10 号学生偏科.

函数 stars() 可以加各种参数，画各种不同的星图，其使用方法如下：

```
stars(x, full = TRUE, scale = TRUE, radius = TRUE,
   labels = dimnames(x)[[1]], locations = NULL,
   nrow = NULL, ncol = NULL, len = 1,
   key.loc = NULL, key.labels = dimnames(x)[[2]], key.xpd = TRUE,
   xlim = NULL, ylim = NULL, flip.labels = NULL,
   draw.segments = FALSE, col.segments = 1:n.seg, col.stars = NA,
   axes = FALSE, frame.plot = axes,
   main = NULL, sub = NULL, xlab = "", ylab = "",
   cex = 0.8, lwd = 0.25, lty = par("lty"), xpd = FALSE,
   mar = pmin(par("mar"),
              1.1+ c(2*axes+ (xlab != ""),
              2*axes+ (ylab != ""), 1,0)),
```

```
   add = FALSE, plot = TRUE, ...)
```

其中 x 是矩阵或数据框. full 是逻辑变量, 如果 full = TRUE (缺省值), 则星图画成圆的; 否则 (FALSE) 画成上半圆图形. scale 是逻辑变量, 当 scale = TRUE (缺省值), 数据矩阵的每一列是独立的, 并且每列的最大值为 1, 最小值为 0; 否则 (FALSE) 所有星图会叠在一起. radius 是逻辑变量, 当 radius = TRUE (缺省值), 绘出星图的半径构成的连线; 否则 (FALSE) 绘出的星图无半径构成的连线. len 是半径尺度因子 (缺省值为 1), 表明星图的比例. key.loc 是一个由 x 与 y 坐标构成的向量 (缺省值为 NULL), 它表明标准星的的位置. draw.segments 是逻辑变量, 当 draw.segments = TRUE (缺省值是 FALSE), 绘出的星图是一段一段的弧. 其他参数的使用方法请参见在线帮助.

调整函数 stars() 中的参数, 可将例 3.18 中 12 名学生学习成绩的星图画成另一种形式

```
> stars(X, full=FALSE, draw.segments = TRUE,
        key.loc = c(5,0.5),  mar = c(2,0,0,0))
```

画出星图如图 3.16 所示.

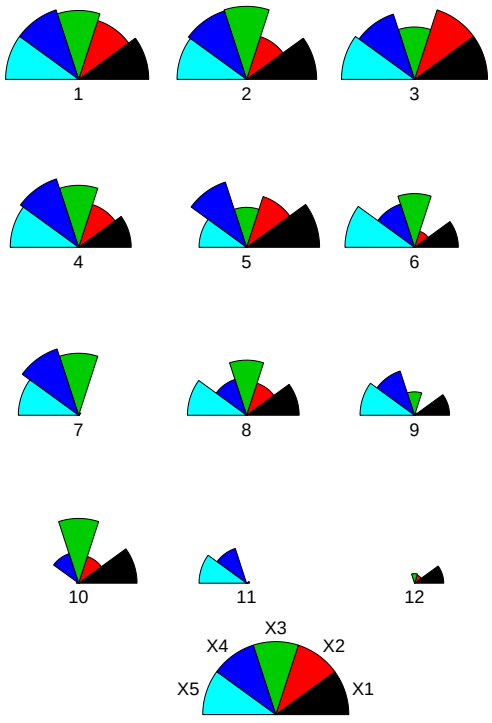


图 3.16: 12 名学生 5 门课程的考试成绩的星图 (带参数)

3.5.3 调和曲线图

调和曲线图是 Andrews (安德鲁斯) 在 1972 年提出来的三角表示法，其思想是将多维空间中的一个点对应于二维平面的一条曲线，对于 p 维数据，假设 X_r 是第 r 观测值，即

$$X_r^T = (x_{r1}, x_{r2}, \cdots, x_{rp}),$$

则对应的调和曲线是

$$\begin{aligned} f_r(t) \;=\; & \frac{x_{r1}}{\sqrt{2}} + x_{r2}\cdot\sin(t) + x_{r3}\cdot\cos(t) + x_{r4}\cdot\sin(2t) + x_{r5}\cdot\cos(2t) + \\ & +\cdots+, \qquad -\pi \le t \le \pi. \end{aligned} \tag{3.29}$$

n 次观测数据对应 n 条曲线，现在同一张平面上就是一张调和曲线图. 当各变量数据的数值相差太悬殊，最好先标准化再作图.

按照式 (3.29) 编写画调和曲线图函数（函数名： `unison.R` ）

```
unison <- function(x){
   if (is.data.frame(x) == TRUE)
      x <- as.matrix(x)
   t <- seq(-pi, pi, pi/30)
   m <- nrow(x); n<-ncol(x)
   f <- array(0, c(m,length(t)))
   for(i in 1:m){
      f[i,] <- x[i,1]/sqrt(2)
      for( j in 2:n){
          if (j%%2 == 0)
             f[i,] <- f[i,]+x[i,j]*sin(j/2*t)
          else
             f[i,] <- f[i,]+x[i,j]*cos(j%/%2*t)
      }
   }
   plot(c(-pi,pi), c(min(f), max(f)), type = "n",
        main = "The Unison graph of Data",
        xlab = "t", ylab = "f(t)")
```

```
        for(i in 1:m) lines(t, f[i,] , col = i)
    }
```

其中 x 是矩阵或数据框. 函数的输出结果是调和曲线.

例 3.19 画出例 *3.18* 中 *12* 名学生学习成绩的调和曲线图.

解: 用编好的函数 unison() 作图,

```
> source("unison.R")
> unison(X)
```

绘出调和曲线图, 如图 3.17 所示.

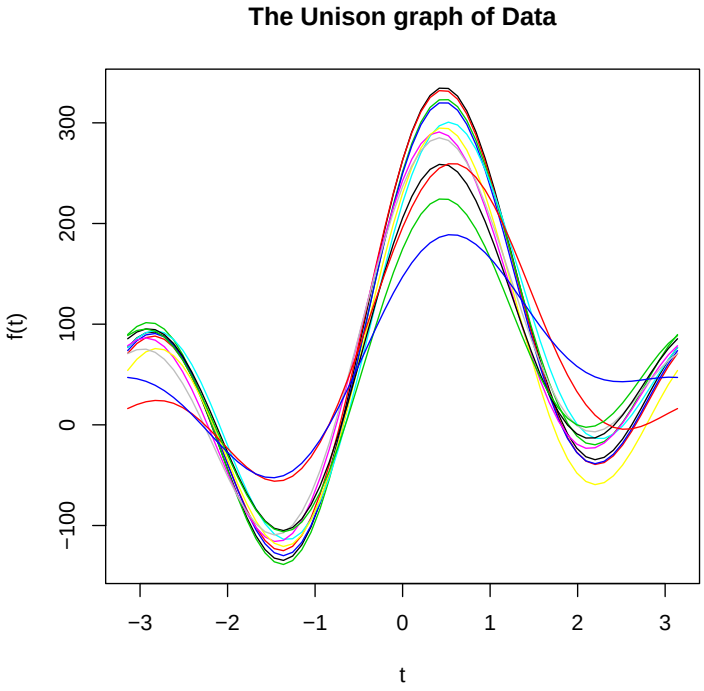


图 3.17: 12 名学生 5 门课程的调和曲线图

Andrews 证明了三角式多项式图有许多很好的性质, 这种图对聚类分析帮助很大. 如果选择聚类统计量为距离, 则同类的曲线拧在一起, 不同类的曲线拧成不同的束, 非常直观.

习题三

3.1 某单位对 *100* 名女生测定血清总蛋白含量 (g/L), 数据如下:

```
74.3 78.8 68.8 78.0 70.4 80.5 80.5 69.7 71.2 73.5
79.5 75.6 75.0 78.8 72.0 72.0 72.0 74.3 71.2 72.0
75.0 73.5 78.8 74.3 75.8 65.0 74.3 71.2 69.7 68.0
73.5 75.0 72.0 64.3 75.8 80.3 69.7 74.3 73.5 73.5
75.8 75.8 68.8 76.5 70.4 71.2 81.2 75.0 70.4 68.0
70.4 72.0 76.5 74.3 76.5 77.6 67.3 72.0 75.0 74.3
73.5 79.5 73.5 74.7 65.0 76.5 81.6 75.4 72.7 72.7
67.2 76.5 72.7 70.4 77.2 68.8 67.3 67.3 67.3 72.7
75.8 73.5 75.0 73.5 73.5 73.5 72.7 81.6 70.3 74.3
73.5 79.5 70.4 76.5 72.7 77.2 84.3 75.0 76.5 70.4
```

计算均值、方差标准差、极差、标准误、变异系数、偏度、峰度.

3.2 绘出习题 *3.1* 的直方图、密度估计曲线、经验分布图和 *QQ* 图，并将密度估计曲线与正态密度曲线相比较，将经验分布曲线与正态分布曲线相比较（其中正态曲线的均值和标准差取习题 *3.1* 计算出的值）.

3.3 绘出习题 *3.1* 的茎叶图、箱线图，并计算五数总括.

3.4 分别用 *W* 检验方法和 *Kolmogorov-Smirnov* 检验方法检验习题 *3.1* 的数据是否服从正态分布.

3.5 小白鼠在接种了 *3* 种不同菌型的伤寒杆菌后的存活天数如表 *3.8* 所示，试

表 3.8: 白鼠试验数据

菌型	存 活 日 数											
1	2	4	3	2	4	7	7	2	2	5	4	
2	5	6	8	5	10	7	12	12	6	6		
3	7	11	6	6	7	9	5	5	10	6	3	10

绘出数据的箱线图（采用两种方法，一种是 plot 语句，另一种是 boxplot 语句）来判断小白鼠被注射三种菌型后的平均存活天数有无显著差异？

3.6 绘出例 *3.16* 关于三项指标的离散图，从图中分析例 *3.16* 的结论的合理性.

3.7 某校测得 *19* 名学生的四项指标, 性别、年龄、身高 *(cm)* 和体重 *(*磅*),* 具体数据由表 *3.9* 所示. *(1)* 试绘出体重对于身高的散点图; *(2)* 绘出不同性别情

表 3.9: 学生身高、体重的数据

学号	姓名	性别	年龄	身高	体重
01	Alice	F	13	56.5	84.0
02	Becka	F	13	65.3	98.0
03	Gail	F	14	64.3	90.0
04	Karen	F	12	56.3	77.0
05	Kathy	F	12	59.8	84.5
06	Mary	F	15	66.5	112.0
07	Sandy	F	11	51.3	50.5
08	Sharon	F	15	62.5	112.5
09	Tammy	F	14	62.8	102.5
10	Alfred	M	14	69.0	112.5
11	Duke	M	14	63.5	102.5
12	Guido	M	15	67.0	133.0
13	James	M	12	57.3	83.0
14	Jeffrey	M	13	62.5	84.0
15	John	M	12	59.0	99.5
16	Philip	M	16	72.0	150.0
17	Robert	M	12	64.8	128.0
18	Thomas	M	11	57.5	85.0
19	William	M	15	66.5	112.0

况下, 体重与身高的散点图; *(3)* 绘出不同年龄段的体重与身高的散点图; *(4)* 分不同性别和不同年龄段的体重与身高的散点图.

3.8 画出函数 $z = x^4 - 2x^2y + x^2 - 2xy + 2y^2 + \frac{9}{2}x - 4y + 4$ 在区域 $-2 \le x \le 3$, $-1 \le y \le 7$ 上的三维网格曲面和二维等值线, 其中 x 与 y 各点之间的间隔为

0.05, 等值线的值分别为 *0, 1, 2, 3, 4, 5, 10, 15, 20, 30, 40, 50, 60, 80, 100,* 共 *15* 条. *(*注：在三维图形中选择合适的角度.*)*

3.9 用 *pearson* 相关检验法检验习题 *3.7* 中的身高与体重是否相关.

3.10 绘出例 *3.17* 中 *48* 名求职者数据的星图. *(1)* 以 *15* 项自变量 FL，APP, $\cdots$, SUIT 为星图的轴; *(2)* 以 $G_1, G_2, \cdots, G_5$ 为星图的轴. 通过这些星图，你能否说明应选哪 *6* 名应聘者. 为使星图能够充分反映应聘者的情况，在作图中可适当调整各种参数.

3.11 绘出例 *3.17* 中 *48* 名求职者数据的调和曲线，以 $G_1, G_2, \cdots, G_5$ 为自变量.

第四章　参数估计

总体是由总体分布来刻画的. 在实际问题中我们根据问题本身的专业知识或以往的经验或用适当的统计方法, 有时可以判断总体分布的类型, 但是总体分布的参数还是未知的, 需要通过样本来估计. 例如, 为了研究人们的市场消费行为, 要先搞清楚人们的收入状况. 若假设某城市人均年收入服从正态分布 $N(\mu, \sigma^2)$, 但参数 μ 和 σ^2 的具体取值并不知道, 需要通过样本来估计. 又如, 假定某城市在单位时间 (譬如一个月) 内交通事故发生次数服从 Poisson 分布 $P(\lambda)$, 其中的参数 λ 也是未知的, 同样需要用样本来估计. 根据样本来估计总体分布所包含的未知参数, 叫作参数估计 (parametric estimation). 它是统计推断的一种重要形式.

如何根据样本的取值来寻找这些参数的估计呢？通常有两种形式：一种称为点估计 (point estimation), 另一种称为区间估计 (interval estimation). 点估计就是用一个统计量来估计一个未知参数. 点估计的优点是：能够明确地告诉人们 "未知参数大致是多少". 其缺点是：不能反映出估计的可信程度. 区间估计是用两个统计量所构成的区间来估计一个未知的参数, 并同时指明此区间可以覆盖住这个参数的可靠程度 (置信度). 它的缺点是：不能直接地告诉人们 "未知参数具体是多少" 这一明确的概念.

4.1　点估计

设总体 X 分布由有限个未知参数 $\theta = (\theta_1, \theta_2, \cdots, \theta_m)^T$ 所决定, 记为 F_θ, 称 θ 可能取值的范围为参数空间 (parameter space), 记作 Θ.

记 $f(x;\theta)$ 为总体 X 的概率密度函数或分布律, 若总体 X 分布为连续型的, 则 $f(x;\theta)$ 是概率密度函数. 若总体 X 分布为离散型的, 则 $f(x;\theta)$ 是分布律. 例如, 对于 Poisson 分布 $P(\lambda)$, $\theta = \lambda$ 就是 1 维未知参数. 对于正态分布 $N(\mu, \sigma^2)$, $\theta = (\mu, \sigma^2)$ 就是 2 维未知参数.

为了估计总体 X 的参数 θ, 就要从总体 X 中抽出一个样本 $X_1, X_2, \cdots, X_n$ (即 $X_1, X_2, \cdots, X_n$ 是独立同分布), 它们的共同分布就是总体分布 $f(x;\theta)$. 为了估计 θ, 需要构造适当的统计量 $\hat{\theta}(X_1, X_2, \cdots, X_n)$, 它只依赖于样本, 不依赖于未知参数. 也就是说, 一旦有了样本 $X_1, X_2, \cdots, X_n$, 就可以计算出 $\hat{\theta}(X_1, X_2, \cdots, X_n)$

的值，作为 θ 的估计值. 称统计量 $\hat{\theta}(X_1, X_2, \cdots, X_n)$ 为 θ 的估计，简记为 $\hat{\theta}$. 因为未知参数 θ 和估计 $\hat{\theta}$ 都是空间上的点，因此称这样的估计为点估计. 寻找点估计的常用方法有：矩法、极大似然法和最小二乘法等.

4.1.1 矩法

矩法 (method of moments) 是由英国统计学家 K · Pearson 在 20 世纪初提出来的，它的中心思想就是用样本矩去估计总体矩.

设总体 X 的分布中的未知参数为 $\theta = (\theta_1, \theta_2, \cdots, \theta_m)^T$, 假定总体 X 的 k 阶原点矩

$$\mathrm{E}(X^k) = \alpha_k(\theta_1, \theta_2, \cdots, \theta_m), \quad k = 1, 2, \cdots, m$$

存在，我们令总体的 k 阶原点矩等于它样本的 k 阶原点矩

$$A_k = \frac{1}{n}\sum_{i=1}^{n} X_i^k, \quad k = 1, 2, \cdots, m,$$

即

$$\alpha_k(\theta_1, \theta_2, \cdots, \theta_m) = \mathrm{E}(X^k) = \frac{1}{n}\sum_{i=1}^{n} X_i^k = A_k, \quad k = 1, 2, \cdots, m. \tag{4.1}$$

由方程 (4.1) 可以得到关于未知量 θ 的解

$$\hat{\theta}_i = \hat{\theta}_i(X_1, X_2, \cdots, X_n), \quad i = 1, 2, \cdots, m. \tag{4.2}$$

取 $\hat{\theta} = (\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_m)^T$ 作为 $\theta = (\theta_1, \theta_2, \cdots, \theta_m)^T$ 的估计，则称 $\hat{\theta}$ 为 θ 的矩估计 (estimation by moments), 用矩估计参数的方法称为矩法.

例 4.1 *设总体 X 的均值为 μ, 方差为 σ^2, $X_1, X_2, \cdots, X_n$ 是来自总体 X 的一个样本，试用矩方法估计均值 μ, 和方差 σ^2.*

解：计算总体 X 的一阶、二阶原点矩

$$\begin{aligned} \alpha_1 &= E(X) = \mu, \\ \alpha_2 &= E(X^2) = \mathrm{Var}(X) + [E(X)]^2 = \sigma^2 + \mu^2. \end{aligned}$$

和样本的一阶、二阶原点矩

$$A_1 = \frac{1}{n}\sum_{i=1}^{n} X_i = \overline{X}, \quad A_2 = \frac{1}{n}\sum_{i=1}^{n} X_i^2.$$

由式 (4.1) 得到方程组

$$\begin{cases} \mu = \overline{X}, \\ \sigma^2 + \mu^2 = \dfrac{1}{n}\sum\limits_{i=1}^{n} X_i^2. \end{cases}$$

解上述方程组得到均值 μ 和方差 σ^2 的矩估计

$$\hat{\mu} = \overline{X}, \tag{4.3}$$

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} X_i^2 - \overline{X}^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})^2. \tag{4.4}$$

需要特别注意的是：方差的矩估计并不等于样本方差 S^2, 而是有如下关系式

$$\hat{\sigma}^2 = \frac{n-1}{n}S^2. \tag{4.5}$$

对于正态分布 $N(\mu, \sigma^2)$, 因为 μ 和 σ^2 分别为总体的均值和方差，由式 (4.3) 和式 (4.4) 得到参数 μ 和 σ^2 的矩估计

$$\hat{\mu} = \overline{X}, \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})^2.$$

从上述过程，可以看到，利用矩法估计均值和方差，就等价于用样本的一阶原点矩估计均值，用样本的二阶中心矩估计方差.

例 4.2 设总体 X 服从指数分布，密度函数是

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0, \\ 0 & x < 0, \end{cases}$$

其中 λ 是未知参数. 若 $X_1, X_2, \cdots, X_n$ 来自总体 X 的一个样本，试用矩估法估计参数 λ.

解：指数分布的一阶矩（均值）是 $1/\lambda$, 因此，它的估计是

$$\widehat{\lambda} = n \Big/ \sum_{i=1}^{n} X_i.$$

例 4.3 设总体 X 是区间 $[0, \theta]$ 上的均匀分布，其中 θ 是未知参数， $X_1, X_2, \cdots, X_n$ 是总体 X 的一个样本，试用矩法估计参数 θ.

解: 均匀分布的一阶矩 (均值) 是 $\theta/2$, 因此, 它的估计是

$$\theta = 2\overline{X} = \frac{2}{n}\sum_{i=1}^{n} X_i.$$

例 4.4 *设总体 X 是区间 $[a,\ b]$ 上的均匀分布, 其中 $a,\ b$ 是未知参数, $X_1, X_2, \cdots, X_n$ 是总体 X 的一个样本, 试用矩估法估计参数 a 和 b.*

解: 由例 4.1 的计算过程 (式 (4.3)-(4.4)) 可知, 用一、二阶原点矩作估计, 本质上相当用一阶原点估计均值, 二阶中心矩估计方差, 即

$$\mathrm{E}(X) = A_1 = \frac{1}{n}\sum_{i=1}^{n} X_i, \quad \mathrm{Var}(X) = M_2 = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2.$$

均匀分布的均值是 $(b-a)/2$, 方差是 $(b-a)^2/12$, 所以令

$$\frac{b+a}{2} = \overline{X}, \quad \frac{(b-a)^2}{12} = M_2,$$

解上述方程组得到 a 和 b 的估计分别为

$$\hat{a} = \overline{X} - \sqrt{3M_2}, \quad \hat{b} = \overline{X} + \sqrt{3M_2}. \tag{4.6}$$

如果不能得到方程 (4.1) 解的解析表达式, 则可以通过数值的方法求解方程 (4.1), 得到相应的矩估计.

例 4.5 *设总体 X 服从二项分布 $B(k,p)$, 其中 k,p 为未知参数, $X_1, X_2, \cdots, X_n$ 是总体 X 的一个样本, 求参数 k,p 的矩估计 $\hat{k}, \hat{p}$.*

解: 尽管本例可以得到方程 (4.1) 解的解析表达式, 但为了演示数值计算的过程和比较数值计算的精确程度, 这里还是采用数值计算的方法进行矩估计.

二项分布的均值 (总体一阶原点矩) 是 kp, 方差 (总体二阶中心矩) 是 $kp(1-p)$. 建立方程组

$$kp - \overline{X} = 0, \quad kp(1-p) - M_2 = 0. \tag{4.7}$$

编写相应的 R 函数 (程序名: `moment_fun.R`)

```
moment_fun<-function(p){
   f<-c(p[1]*p[2]-A1, p[1]*p[2]-p[1]*p[2]^2-M2)
   J<-matrix(c(p[2], p[1],
```

```
                p[2]-p[2]^2, p[1]-2*p[1]*p[2]),
              nrow=2, byrow=T)
    list(f=f, J=J)
}
```

其中 p[1] 表示参数 k, p[2] 表示参数 p, f 是由方程 (4.7) 左端构造的函数， J 为函数 f 的 Jacobi 矩阵.

考虑用 Newton 法 (见第二章 2.9.3 节) 求解非线性方程组 (4.7), 其中样本取值由随机数产生. 建立矩估计的 R 函数 (程序名： moment_estimate.R)

```
x<-rbinom(100, 20, 0.7); n<-length(x)
A1<-mean(x); M2<-(n-1)/n*var(x)
source("moment_fun.R"); source("Newtons.R")
p<-c(10,0.5); Newtons(moment_fun, p)
```

在程序中, 第一句是产生 100 个 $k = 20$, $p = 0.7$ 的二项分布的随机数; 第二句是计算样本均值 (样本一阶原点矩) 和样本二阶中心矩. 第三句是调入已编好的程序 moment_fun.R 和 Newtons.R, 其中 source() 语句是已编好的程序调入内存, 其使用格式是:

```
source("FileName")
```

文件名 ("FileName") 中可以包含文件的路径.

最后一句是给出初值, 调用 Newton 法计算方程的根. 其计算结果如下

```
$root
[1] 19.4957061  0.7237491
$it
[1] 11
$index
[1] 1
$FunVal
[1]  0.000000e+00 -2.220446e-15
```

经过 11 次迭代, 得到计算结果.

下面给出方程 (4.7) 解析解的计算结果

$$\hat{k} = \frac{\overline{X}^2}{\overline{X} - M_2} = 19.49571, \quad \hat{p} = \frac{\overline{X} - M_2}{\overline{X}} = 0.7237491.$$

两者比较，误差是很小的.

此例表明，在无法得到方程 (4.1) 解析解的情况下，利用数值计算，得到数值解也不失一种较好的方法.

通过上述的例子可以看出，矩法的优点是：在其能用的情况下，计算往往很简单. 但矩法相对其他估计方法，如极大似然法，其效率往往较低.

4.1.2 极大似然法

极大似然法是 Fisher(费希尔) 在 1912 年提出的一种应用非常广泛的参数估计方法，其思想始于 Gauss 的误差理论，它具有很多优良的性质. 它充分利用总体分布函数的信息，克服了矩法的某些不足.

设 Θ 是参数空间，参数 θ 可取 Θ 的所有值，在给定样本的观察值 $(x_1, x_2, \cdots, x_n)$ 后，不同的 θ 对应于 $(X_1, X_2, \cdots, X_n)$ 落入 $(x_1, x_2, \cdots, x_n)$ 的邻域内的概率大小不同，既然在一次试验中就观察到了 $(X_1, X_2, \cdots, X_n)$ 的取值为 $(x_1, x_2, \cdots, x_n)$, 因此，可以认为 θ 是最有可能来源于使 $(X_1, X_2, \cdots, X_n)$ 落入 $(x_1, x_2, \cdots, x_n)$ 邻域内的概率达到最大者 $\hat{\theta}$, 即

$$\prod_{i=1}^{n} f(x_i; \hat{\theta}) = \sup_{\theta \in \Theta} \prod_{i=1}^{n} f(x_i; \theta). \tag{4.8}$$

取 $\hat{\theta}$ 作为 θ 的估计，这就是极大似然原理.

注意到，当 X 为连续型随机变量时，式 (4.8) 中的 $f(x_i; \theta)$ 是参数的取值为 θ 时， X 的概率密度函数在 x_i 处的取值，当 X 为离散型随机变量时， $f(x_i; \theta)$ 为参数 θ 时， X 取 x_i 的概率 (分布律).

定义 4.1 设总体 X 的概率密度函数或分布律为 $f(x; \theta)$, $\theta \in \Theta$ 是未知参数，$X_1, X_2, \cdots, X_n$ 来自总体 X 的样本，称

$$L(\theta; x) = L(\theta; x_1, x_2, \cdots, x_n) = \prod_{i=1}^{n} f(x_i; \theta)$$

为 θ 的似然函数 *(likelihood function).*

显然，若样本取值 x 固定时， $L(\theta; x)$ 是 θ 的函数. 若参数 θ 固定，当 X 为连续型随机变量时，它就是样本 $(X_1, X_2, \cdots, X_n)$ 的联合概率密度函数；当 X 为离散型随机变量时，它就是样本 $(X_1, X_2, \cdots, X_n)$ 的联合分布律.

定义 4.2 设总体 X 的概率密度函数或分布律为 $f(x;\theta)$, $\theta \in \Theta$ 是未知参数, $X_1, X_2, \cdots, X_n$ 来自总体 X 的样本, $L(\theta; x)$ 为 θ 的似然函数, 若 $\hat{\theta} = \hat{\theta}(X) = \hat{\theta}(X_1, X_2, \cdots, X_n)$ 是一个统计量且满足

$$L(\hat{\theta}(X); X) = \sup_{\theta \in \Theta} L(\theta; X),$$

则称 $\hat{\theta}(X)$ 为 θ 的极大似然估计 *(maximum likelihood estimation)*, 简记为 *MLE*. 用极大似然估计来估计参数的方法为称极大似然法.

下面分不同情况介绍极大似然法的求解过程.

(1) 似然函数 $L(\theta; X)$ 为 θ 的连续函数, 且关于 θ 的各分量的偏导数存在.

设 θ 是 m 维变量, 且 $\Theta \subset R^m$ 为开区域, 则由极值的一阶必要条件, 得到

$$\frac{\partial L(\theta; X)}{\partial \theta_i} = 0, \quad i = 1, 2, \cdots, m. \tag{4.9}$$

通常称式 (4.9) 为似然方程. 由于独立同分布的样本的似然函数 $L(\theta; X)$ 具有连乘积的形式, 故对 $L(\theta; X)$ 取对数后再求偏导数是方便的, 因此实用上常采用与 (4.9) 等价的形式

$$\frac{\partial \ln L(\theta; X)}{\partial \theta_i} = 0, \quad i = 1, 2, \cdots, m. \tag{4.10}$$

称式 (4.10) 为对数似然方程 (loglikelihood equation).

值得注意的是: 由极值的必要条件知, 极大似然估计一定是似然方程或对数似然方程的解, 但似然方程或对数似然方程的解未必都是极大似然估计. 严格地讲, 似然函数 $L(\theta; X)$ 或对数似然函数 $\ln L(\theta; X)$ 对于参数 θ 的二阶 Hesse 矩阵 $\nabla_\theta^2 L(\theta; X)$ 或 $\nabla_\theta^2 \ln L(\theta; X)$ 负定 (若 θ 是一元变量, $\frac{\partial^2 L(\theta;X)}{\partial \theta^2} < 0$ 或 $\frac{\partial^2 \ln L(\theta;X)}{\partial \theta^2} < 0$), 则似然方程或对数似然方程的解才是极大似然估计.

例 4.6 设总体 X 服从正态分布 $N(\mu, \sigma^2)$, 其中 μ, σ^2 为未知参数, $X_1, X_2, \cdots, X_n$ 是来自总体 X 的一个样本, 试用极大似然法估计参数 (μ, σ^2).

解: 正态分布的似然函数为

$$L(\mu, \sigma^2; x) = \prod_{i=1}^{n} f(x_i; \mu, \sigma^2) = (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2\right],$$

相应的对数似然函数为

$$\ln L(\mu, \sigma^2; x) = -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2.$$

令

$$\begin{cases} \dfrac{\partial \ln L(\mu,\sigma^2;x)}{\partial \mu} = \dfrac{1}{\sigma^2}\sum\limits_{i=1}^{n}(x_i-\mu) = 0, \\ \dfrac{\partial \ln L(\mu,\sigma^2;x)}{\partial \sigma^2} = -\dfrac{n}{2\sigma^2} + \dfrac{1}{2\sigma^4}\sum\limits_{i=1}^{n}(x_i-\mu)^2 = 0, \end{cases}$$

解此似然方程组得到:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i = \overline{x}, \qquad \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i-\overline{x})^2.$$

进一步验证, 对于对数似然函数 $\ln L(\mu,\sigma^2;x)$ 的二阶 Hesse 矩阵

$$\begin{bmatrix} -\frac{n}{\sigma^2} & -\frac{1}{\sigma^4}\sum\limits_{i=1}^{n}(x_i-\mu) \\ -\frac{1}{\sigma^4}\sum\limits_{i=1}^{n}(x_i-\mu) & \frac{n}{2\sigma^4} - \frac{1}{\sigma^6}\sum\limits_{i=1}^{n}(x_i-\mu)^2 \end{bmatrix} = \begin{bmatrix} -\frac{n}{\sigma^2} & 0 \\ 0 & -\frac{n}{2\sigma^4} \end{bmatrix}$$

是负定矩阵, 所以 $\left(\overline{x},\ \dfrac{1}{n}\sum\limits_{i=1}^{n}(x_i-\overline{x})^2\right)$ 是 $L(\mu,\sigma^2;x)$ 的极大值点. 故 (μ,σ^2) 的极大似然估计是

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i = \overline{X}, \qquad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(X_i-\overline{X}\right)^2.$$

与例 4.1 相比较, 两者的计算结果是相同的.

例 4.7 设总体 X 的服从指数分布, 密度函数是

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0, \\ 0 & x < 0, \end{cases}$$

其中 λ 是未知参数. 若 $X_1, X_2, \cdots, X_n$ 来自总体 X 的一个样本, 试用极大似然估计求参数 λ.

解: 只考虑 $x_i \geq 0$ 部分, 指数分布的似然函数为

$$L(\lambda;x) = \prod_{i=1}^{n} f(x_i;\lambda) = \lambda^n \exp\left[-\lambda\sum_{i=1}^{n} x_i\right],$$

相应的对数似然函数为

$$\ln L(\lambda;x) = n\ln\lambda - \lambda\sum_{i=1}^{n} x_i.$$

令

$$\frac{\partial \ln L(\lambda; x)}{\partial \lambda} = \frac{n}{\lambda} - \sum_{i=1}^{n} x_i = 0,$$

解此似然方程组得到:

$$\lambda = n \Big/ \sum_{i=1}^{n} x_i.$$

由于 $\dfrac{\partial^2 \ln L(\lambda; x)}{\partial \lambda^2} = -\dfrac{n}{\lambda^2} < 0$, 因此, $n \Big/ \sum\limits_{i=1}^{n} x_i$ 是 $L(\lambda; x)$ 的极大值点. 故 λ 的极大似然估计是 $n \Big/ \sum\limits_{i=1}^{n} X_i$.

与例 4.2 相比较, 两者的计算结果也是相同的.

(2) 似然函数 $L(\theta; x)$ 关于 θ 有间断点.

当 Θ 为 R^m 中的开区域, 此时求似然方程组解的方法不适用, 要具体问题具体分析.

例 4.8 设总体 X 是区间 $[a,\ b]$ 上的均匀分布, 其中 $a,\ b$ 是未知参数, $X_1, X_2, \cdots, X_n$ 是总体 X 的一个样本, 试用极大似然法估计参数 a 和 b.

解: 对于样本 $X_1, X_2, \cdots, X_n$, 其似然函数为

$$L(a, b; x) = \begin{cases} \dfrac{1}{(b-a)^n}, & \text{若} a \le x_i \le b, \quad i = 1, 2, \cdots, n, \\ 0, & \text{其它}. \end{cases}$$

很显然, $L(a, b; x)$ 不是 (a, b) 的连续函数, 因此不能用似然方程组 (4.10) 求解, 而必需从极大似然估计的定义出发来求 $L(a, b; x)$ 的最大值. 为了使 $L(a, b; x)$ 达到最大, 则 $b - a$ 应该尽可能的小, 但 b 不能小于 $\max\{x_1, x_2, \cdots, x_n\}$; 否则 $L(a, b; x) = 0$. 类似地, a 不能大于 $\min\{x_1, x_2, \cdots, x_n\}$. 因此, a 和 b 的极大似然估计为

$$\hat{a} = \min\{X_1, X_2, \cdots, X_n\} = X_{(1)}, \qquad \hat{b} = \max\{X_1, X_2, \cdots, X_n\} = X_{(n)}.$$

同样的理由, 若用极大似然法估计例 4.5 中的 θ, 得到的结果是

$$\widehat{\theta} = X_{(n)}.$$

对于这两个例子, 极大似然法与矩法估计出的值是不相同的.

(3) Θ 为离散参数空间.

在此情况下，为求极大似然估计，经常考虑参数取相邻的值时，似然函数的比值.

例 4.9 *在鱼池中随机地捕捞 500 条鱼，做上记号后再放入池中，待充分混合后，再捕捞 1000 条，结果发现其中有 72 条鱼带有记号. 试问鱼池中可能有多少条鱼？*

解：先将问题一般化. 设池中有 N 条鱼，其中 r 条带有记号，随机地捕捞到 s 条，发现 x 条带有记号，用上述信息来估计 N.

用 X 记捕捞到的 s 条鱼中带有记号的鱼数，则有

$$P\{X=x\}=\frac{C_{N-r}^{s-x}C_r^x}{C_N^s}.$$

因此，似然函数为

$$L(N;x)=P\{X=x\},$$

考虑似然函数的比

$$g(N)=\frac{L(N;x)}{L(N-1;x)}=\frac{(N-s)(N-r)}{N(N-r-s+x)}=\frac{N^2-(r+s)N+rs}{N^2-(r+s)N+xN},$$

当 $rs>xN$ 时，有 $g(N)>1$, 当 $rs<xN$ 时，有 $g(N)<1$. 即

$$\begin{cases} L(N;x)>L(N-1;x), & \text{当 } N<\dfrac{rs}{x}, \\ L(N;x)<L(N-1;x), & \text{当 } N>\dfrac{rs}{x}. \end{cases}$$

因此，似然函数 $L(N;x)$ 在 $N=\dfrac{rs}{x}$ 附近达到极大，注意到 N 只取正整数，易得 N 的极大似然估计为：

$$\hat{N}=\left\lceil\frac{rs}{x}\right\rceil,$$

其中 $\lceil\ \cdot\ \rceil$ 表示下取整，即小于该值的最大整数.

将题目中的数字代入，得到 $\hat{N}=\left\lceil\dfrac{500\times1000}{72}\right\rceil=6944$. 即鱼池中鱼的总数估计为 6944 条.

(4) 如果在解 (对数) 似然方程时无法得到解析表达式，只能采用数值方法.

例 4.10 设总体 X 服从 *Cauchy* 分布，其概率密度函数为

$$f(x;\theta)=\frac{1}{\pi[1+(x-\theta)^2]},\qquad -\infty<x<\infty,$$

其中 θ 为未知参数. $X_1, X_2, \cdots, X_n$ 来自总体 X 的样本，求 θ 的极大似然估计.

解：Cauchy 分布的似然函数为

$$L(\theta;x)=\prod_{i=1}^{n}f(x_i;\theta)=\frac{1}{\pi^n}\prod_{i=1}^{n}\frac{1}{1+(x_i-\theta)^2},$$

相应的对数似然函数为

$$\ln L(\theta;x)=-n\ln(\pi)-\sum_{i=1}^{n}\ln\left(1+(x_i-\theta)^2\right), \tag{4.11}$$

得到对数似然方程

$$\sum_{i=1}^{n}\frac{x_i-\theta}{1+(x_i-\theta)^2}=0. \tag{4.12}$$

可以看到，得到对数似然方程 (4.12) 的解析解是困难的，下面考虑用 R 软件求数值解.

在第二章 (2.9.1 节) 介绍了方程求根函数 `uniroot()`, 这里用它求似然方程 (4.12) 的根. 关于样本 X 的取值用随机数产生.

```
> x <- rcauchy(1000,1)
> f <- function(p)  sum((x-p)/(1+(x-p)^2))
> out <- uniroot(f, c(0, 5))
```

在程序中，第一句是产生 1000 个参数 $\theta=1$ 的随机数；第二句写出似然方程 (4.12) 对应的函数. 第三句是用求根函数 `uniroot()` 求似然方程在区间 $(0,5)$ 内的根. 其计算结果为

```
> out
$root
[1] 1.049538
$f.root
[1] -0.006061751
```

```
$iter
[1] 5
$estim.prec
[1] 6.103516e-05
```

在计算结果中，$root 是方程根的近似解, 即估计值为 $\widehat{\theta} = 1.049538$. $f.root 是函数 f 在近似值处的函数值. $iter 的迭代次数, 即用了 5 次迭代. $estim.prec 是近似解与精确解的误差估计，即近似解与精确解误差的绝对值不超过 6.104×10^{-5}.

函数 uniroot() 的一般使用格式为

```
uniroot(f, interval,
        lower = min(interval), upper = max(interval),
        tol = .Machine$double.eps^0.25, maxiter = 1000, ...)
```

其中 f 是所求方程的函数. interval 是包含有方程根的初始区间. lower 是初始区间的左端点， upper 是初始区间的右端点. tol 是计算精度， maxiter 是最大迭代次数 (缺省值为 1000).

前面讨论的是如何用 R 软件中的函数求 (对数) 似然方程的根. 事实上，也可以直接用 R 软件中的函数求 (对数) 似然函数的极值.

R 软件中函数 optimize()(或 optimise()) 可直接求一维变量函数的极小点，这里用它求对数似然函数 (4.11) 的极值点，其程序如下

```
> loglike <- function(p) sum(log(1+(x-p)^2))
> out <- optimize(loglike, c(0, 5))
```

在程序中，第一句是对数似然函数 (4.11) (略去常数项，由于求极小，加一个负号). 第二句是用函数 optimize() 求函数 loglike 在区间 $(0, 5)$ 上的极小点. 其计算结果为

```
> out
$minimum
[1] 1.049513
$objective
[1] 1303.192
```

在计算结果中， $minimum 是极小点的近似解，即估计值为 $\widehat{\theta} = 1.049513$. $objective 是目标函数在近似解处的函数值.

与求似然方程根的方法比较，两者的计算结果相差不大. 事实上，求似然方程根的方法可能更准确一些，但此方法需要先求导数，这对于较为复杂的函数，可能会带来一定的困难.

函数 optimize()(和 optimise()) 的一般用法是：

```
optimize(f = , interval = , lower = min(interval),
         upper = max(interval), maximum = FALSE,
         tol = .Machine$double.eps^0.25, ...)
optimise(f = , interval = , lower = min(interval),
         upper = max(interval), maximum = FALSE,
         tol = .Machine$double.eps^0.25, ...)
```

其中 f 是求极小的目标函数. interval 是包含有极小的初始区间. lower 是初始区间的左端点， upper 是初始区间的右端点. maximum 是逻辑变量，如果 maximum = FALSE(缺省值) 表示求函数极小值点；否则 (maximum = TRUE) 表示求函数的极大值点. tol 是计算精度.

当未知参数 θ 是多元变量时，极大似然法求解的数值方法要适用于多变量函数. 例如，可以用 Newton 法 (见第二章的 2.9.3 节) 求解对数似然方程 (4.10). 也可以用 R 软件中的 nlm() 函数直接求解无约束问题

$$\min_{\theta} \ L(\theta; x) \quad \text{或} \quad \min_{\theta} \ \ln L(\theta; x),$$

这里 x 是随机变量 X 的取值.

为了了解 nlm 函数求多元函数极小的方法，这里简单介绍如何用函数 nlm() 求多变量函数 $f(x)$ 的极小值点. 有关 nlm() 函数在统计中的使用，将会在第六章的 6.7.2 节中有关非线性回归的计算中讲到.

用 nlm() 函数求无约束优化问题

$$\min \quad f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \tag{4.13}$$

的极小点，取初始点 $x^{(0)} = (-1.2, 1)^T$. 称函数 (4.13) 为 Rosenbrock 函数，或橡胶函数.

写出目标函数 (程序名： Rosenbrock.R),

```
obj<-function(x){
```

```
    f<-c(10*(x[2]-x[1]^2), 1-x[1])
    sum(f^2)
}
```

将函数调入内存，再调用 nlm() 函数求解

```
> source("Rosenbrock.R")
> x0<-c(-1.2,1); nlm(obj,x0)
```

其中 x0 是初始值，得到

```
$minimum
[1] 3.973766e-12
$estimate
[1] 0.999998 0.999996
$gradient
[1] -6.539275e-07  3.335996e-07
$code
[1] 1
$iterations
[1] 23
```

其中 $minimum 是函数的最优目标值，即 $f^* = 3.973766 \times 10^{-12}$. $estimate 是最优点的估计值，即 $x^* = (0.999998, 0.999996)^T$. $gradient 是在最优点处 (估计值) 目标函数梯度值，即 $\nabla f^* = (-6.539275 \times 10^{-7}, 3.335996 \times 10^{-7})^T$. $code 是指标，这里是 1, 表示迭代成功. $iterations 是迭代次数，这里是 23, 表示进行了 23 次迭代.

实际上， Rosenbrock 函数的最优点是 $x^* = (1,1)^T$, 最优目标函数值为 $f(x^*) = 0$.

通过上述分析和相应的例子，可以得到：矩法的优点是简单，只需知道总体的矩，总体的分布形式不必知道. 而极大似然法则必须知道总体分布形式，并且在一般情况下，似然方程组的求解较为复杂，往往需要在计算机上通过迭代运算才能计算出其近似解.

在上述例子中，分别用矩法和极大似然法对正态分布和均匀分布的参数进行估计，在所得到的估计中，对于正态分布，两种方法得到的参数估计值是一致的，而对均匀分布，两种方法得到的参数估计值不一样. 对某种参数进行估计，

究竟哪种好呢？下面给出估计量的优良性的判别准则.

4.2 估计量的优良性准则

从前面两节的讨论中可以看到，对总体中同一参数 θ, 采用不同的估计方法得到的估计量 $\widehat{\theta}$ 可能是一样的，但对于大多数情况是不一样的. 例如，对于均匀分布 $U[a,b]$, 参数估计的矩法与极大似然法估计的结果是不同的. 究竟如何选择 "较好" 的估计量呢？即如何评价估计量的优劣？这里简单介绍评价估计量优劣的准则 — 估计量的无偏性、有效性和相合性 (一致性).

4.2.1 无偏估计

估计量是随机变量，对于不同的样本值就会得到不同的估计值. 这样，要确定一个估计量的好坏，就不能仅仅依据某次抽样的结果来衡量，而必须由多次抽样的结果来衡量. 对此，一个自然而基本的衡量标准是要求估计量无系统偏差，也就是说，尽管在一次抽样申得到的估计值不一定恰好等于待估参数的真值，但在大量重复抽样 (样本容量相同) 时，所得到的估计值平均起来应与待估参数的真值相同，换句话说，希望估计量的数学期望应等于未知参数的真值，这就是所谓无偏性的要求. 这一直观要求用概率语言描述就是以下定义.

定义 4.3 设 X 是总体，$\theta \in \Theta$ 是包含在总体 X 的分布中的待估参数，$X_1, X_2, \cdots, X_n$ 是来自总体 X 的一个样本. 若估计量 $\widehat{\theta} = \widehat{\theta}(X_1, X_2, \cdots, X_n)$ 的数学期望 $E(\widehat{\theta})$ 存在，且对于任意 $\theta \in \Theta$ 有

$$E(\hat{\theta}) = \theta, \tag{4.14}$$

则称 $\hat{\theta}$ 是 θ 的无偏估计量或无偏估计 *(unbiased estimate)*.

称 $E(\hat{\theta}) - \theta$ 为以 $\hat{\theta}$ 作为 θ 的估计的系统误差或偏差. 无偏估计的实际意义就是无系统误差.

若 $E(\hat{\theta}) - \theta \neq 0$, 但当样本容量 $n \to \infty$ 时，有

$$\lim_{n\to\infty} \Big[E(\hat{\theta}) - \theta\Big] = 0, \tag{4.15}$$

则称 $\hat{\theta}$ 为 θ 的渐近无偏估计.

一个估计量如果不是无偏的，则称它是有偏估计量.

例 4.11 设总体 X 的 k 阶原点矩 $\alpha_k = E(X^k)(k \geq 1)$ 存在，$X_1, X_2, \cdots, X_n$ 是 X 的一个样本，$A_k = \dfrac{1}{n}\sum_{i=1}^{n} X_i^k$ 为样本的 k 阶原点矩，证明：无论总体 X 服从什么分布，则 k 阶样本原点矩 A_k 是 k 阶总体原点矩 α_k 的无偏估计.

证明：设 $X_1, X_2, \cdots, X_n$ 与 X 同分布且相互独立，故有

$$E(X_i^k) = E(X^k) = \alpha_k, \quad i = 1, 2, \cdots, n,$$

即有

$$E(A_k) = E\left(\frac{1}{n}\sum_{i=1}^{n} X_i^k\right) = \frac{1}{n}\sum_{i=1}^{n} E(X_i^k) = \alpha_k.$$

特别地，不论总体 X 服从什么分布，只要数学期望 μ 存在，必有 $E(\overline{X}) = \mu$，即 $\overline{X}$ 是 μ 的无偏估计.

例 4.12 设总体 X 的均值 μ 、方差 σ^2 存在，μ, σ^2 为未知参数，则 σ^2 的估计量

$$\widehat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2$$

是有偏估计量.

证明：由于

$$\begin{aligned}
\widehat{\sigma}^2 &= \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2 = \frac{1}{n}\sum_{i=1}^{n} X_i^2 - \overline{X}^2, \\
E(\widehat{\sigma}^2) &= E\left(\frac{1}{n}\sum_{i=1}^{n} X_i^2\right) - E(\overline{X}^2) = \frac{1}{n}\sum_{i=1}^{n} E\left(X_i^2\right) - E(\overline{X}^2),
\end{aligned}$$

和

$$\begin{aligned}
E\left(X_i^2\right) &= \mathrm{Var}(X_i) + [E(X_i)]^2 = \sigma^2 + \mu^2, \\
E(\overline{X}^2) &= \mathrm{Var}(\overline{X}) + \left[E(\overline{X})\right]^2 = \frac{\sigma^2}{n} + \mu^2,
\end{aligned}$$

则得到

$$E\left(\widehat{\sigma}^2\right) = \sigma^2 + \mu^2 - \left(\frac{\sigma^2}{n} + \mu^2\right) = \frac{n-1}{n}\sigma^2 \neq \sigma^2.$$

所以 $\widehat{\sigma}^2$ 是有偏的，若用 $\widehat{\sigma}^2$ 去估计 σ^2, 则估计值平均偏小，但它是 σ^2 的渐近无偏估计.

对于样本方差，有

$$\begin{aligned} S^2 &= \frac{1}{n-1}\sum_{i=1}^{n}(X_i-\overline{X})^2 = \frac{n}{n-1}\widehat{\sigma}^2, \\ E(S^2) &= \frac{1}{n-1}E(\widehat{\sigma}^2) = \frac{n}{n-1}\cdot\frac{n-1}{n}\sigma^2 = \sigma^2. \end{aligned}$$

这就是说，样本方差 S^2 是总体方差 σ^2 的无偏估计. 故一般都采用 S^2 作为总体方差 σ^2 的估计量.

4.2.2 有效性

在许多情况下，总体参数 θ 的无偏估计量不是惟一的. 那么，如何衡量一个参数的两个无偏估计量何者更好呢？一个重要标准就是观察它们谁的取值更集中于待估计参数的真值附近，即哪一个估计量的方差更小. 这就是下面的有效性概念.

定义 4.4 *设 $\hat{\theta}_1 = \hat{\theta}_1(X_1, X_2, \cdots, X_n)$ 与 $\hat{\theta}_2 = \hat{\theta}_2(X_1, X_2, \cdots, X_n)$ 都是 θ 的无偏估计，若*

$$\mathrm{Var}(\hat{\theta}_1) \le \mathrm{Var}(\hat{\theta}_2),$$

则称 $\hat{\theta}_1$ 比 $\hat{\theta}_2$ 有效.

考察 θ 的所有无偏估计量，如果其中存在一个估计量 $\hat{\theta}_0$ 的方差最小，则此估计量应当最好，并称此估计量 $\hat{\theta}_0$ 为 θ 的最小方差无偏估计 (minimum variance unbiased estimate).

可以证明，对于正态总体 $N(\mu, \sigma^2)$, $(\overline{X}, S^2)$ 是 (μ, σ^2) 的最小方差无偏估计.

有效性的意义是：用 $\hat{\theta}$ 估计 θ 时，除无系统偏差外，还要求估计精度更高.

例 4.13 *设总体 X 的均值 μ 和方差 σ^2 存在，$X_1, X_2, \cdots, X_n$ 是来自总体 X 的一个样本，证明估计 μ 时，$\hat{\mu}_1 = \overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ 比 $\hat{\mu}_2 = \sum_{i=1}^{n} c_i X_i$ 有效，其中 $\sum_{i=1}^{n} c_i = 1$, $c_i > 0$, $i = 1, 2, \cdots, n$.*

解：容易验证，$E(\hat{\mu}_1) = E(\hat{\mu}_2) = \mu$, 都是 μ 的无偏估计. 计算方差得到

$$\begin{aligned} \mathrm{Var}(\hat{\mu}_1) &= \mathrm{Var}(\overline{X}) = \frac{\sigma^2}{n}, \\ \mathrm{Var}(\hat{\mu}_2) &= \mathrm{Var}\left(\sum_{i=1}^{n} c_i X_i\right) = \sum_{i=1}^{n}\left(c_i^2 \mathrm{Var}(X_i)\right) = \sigma^2 \sum_{i=1}^{n} c_i^2. \end{aligned}$$

由不等式 $\left(\sum_{i=1}^{n} c_i\right)^2 \leq n\sum_{i=1}^{n} c_i^2$, 得到

$$\mathrm{Var}(\hat{\mu}_1) = \frac{\sigma^2}{n} = \frac{\sigma^2}{n}\left(\sum_{i=1}^{n} c_i\right)^2 \leq \sigma^2 \sum_{i=1}^{n} c_i^2 = \mathrm{Var}(\hat{\mu}_2),$$

故 $\hat{\mu}_1$ 比 $\hat{\mu}_2$ 有效.

4.2.3 相合性 (一致性)

估计量 $\hat{\theta}$ 的无偏性和有效性都是在样本容量 n 固定的情况下讨论的. 然而, 由于估计量 $\hat{\theta}(X_1, X_2, \cdots, X_n)$ 依赖于样本容量 n, 自然会想到, 一个好的估计量 $\hat{\theta}$, 当样本容量 n 越大时, 由于关于总体的信息也随之增加, 该估计理应越精确越可靠, 特别是当 $n \to \infty$ 时, 估计值将与参数真值几乎完全一致, 这就是估计量的相合性 (或称为一致性). 相合性的严格定义如下:

定义 4.5 设 $\hat{\theta}(X_1, X_2, \cdots, X_n)$ 为未知参数 θ 的估计量, 若对于任意 $\theta \in \Theta$, 当 $n \to \infty$ 时, $\hat{\theta}(X_1, X_2, \cdots, X_n)$ 依概率收敛于 θ, 即对任意 $\varepsilon > 0$, 有

$$\lim_{n\to\infty} P\{|\hat{\theta} - \theta| < \varepsilon\} = 1,$$

则称 $\hat{\theta}$ 为 θ 的相合估计 *(consistent estimate)* 量或一致估计量, 并记为 $\hat{\theta} \xrightarrow{P} \theta(n \to \infty)$.

若当 $n \to \infty$ 时, $\hat{\theta}$ 均方收敛于 θ, 即

$$\lim_{n\to\infty} E(\hat{\theta} - \theta)^2 = 0,$$

则称 $\hat{\theta}$ 为 θ 的均方相合估计量 *(*或一致估计量*)*, 并记为 $\hat{\theta} \xrightarrow{L^2} \theta(n \to \infty)$.

4.3 区间估计

前面介绍的点估计方法是针对总体的某一未知参数 θ, 构造 θ 的一个估计量 $\hat{\theta}(X_1, X_2, \cdots, X_n)$, 对于某次抽样的结果, 即一个样本观察值 $(x_1, x_2, \cdots, x_n)$, 可用估计 $\hat{\theta}(x_1, x_2, \cdots, x_n)$ 作为 θ 的一个近似值, 即认为 $\hat{\theta}(x_1, x_2, \cdots, x_n) \approx \theta$. 但是, 人们要问这种估计的精确性如何? 可信程度如何? 点估计无法回答这些问题. 为了解决这些问题, 需要讨论参数的区间估计.

定义 4.6 设总体 X 的分布函数 $F(x;\theta)$ 含未知参数 θ, 对于给定值 α $(0<\alpha<1)$, 若由样本 $X_1, X_2, \cdots, X_n$ 确定的两个统计量 $\hat{\theta}_1(X_1, X_2, \cdots, X_n)$ 和 $\hat{\theta}_2(X_1, X_2, \cdots, X_n)$ 满足

$$P\left\{\hat{\theta}_1(X_1, X_2, \cdots, X_n) < \theta < \hat{\theta}_2(X_1, X_2, \cdots, X_n)\right\} = 1-\alpha, \tag{4.16}$$

则称随机区间 $(\hat{\theta}_1, \hat{\theta}_2)$ 是参数 θ 的置信度为 $1-\alpha$ 的置信区间 *(confidence interval)*, $\hat{\theta}_1$ 和 $\hat{\theta}_2$ 分别称为置信度为 $1-\alpha$ 的双侧置信区间的置信下限与置信上限, 称 $1-\alpha$ 为置信度或置信系数.

置信区间 $(\hat{\theta}_1, \hat{\theta}_2)$ 是一个随机区间, 对每次的抽样来说, 往往有所不同, 并有时包含了参数 θ, 有时不包含 θ. 但是, 此区间包含 θ 的可能性 (置信度) 是 $1-\alpha$. 显然, 在置信度一定的前提下置信区间的长度越短, 其精度越高, 估计也就越好. 在实用中, 通常给定一定的置信度, 求尽可能短的置信区间.

4.3.1 一个正态总体的情况

假设正态总体 $X \sim N(\mu, \sigma^2)$, $X_1, X_2, \cdots, X_n$ 为来自总体 X 的一个样本, $1-\alpha$ 为置信度, $\overline{X}$ 为样本均值, S^2 为样本方差.

1. 均值 μ 的区间估计

分别讨论总体 X 的方差 σ^2 已知和方差 σ^2 未知两种情形.

当 σ^2 已知时, 由于

$$\frac{\overline{X}-\mu}{\sigma/\sqrt{n}} \sim N(0,1), \tag{4.17}$$

因此有

$$P\left\{\left|\frac{\overline{X}-\mu}{\sigma/\sqrt{n}}\right| \le Z_{\alpha/2}\right\} = 1-\alpha, \tag{4.18}$$

其中 Z_α 为标准正态分布 $N(0,1)$ 上的 α 分位点, 即 $\Phi(Z_\alpha) = 1-\alpha$. 由式 (4.18) 得到关于均值 μ, 置信度为 $1-\alpha$ 的双侧置信区间

$$\left[\overline{X} - \frac{\sigma}{\sqrt{n}}Z_{\alpha/2},\ \ \overline{X} + \frac{\sigma}{\sqrt{n}}Z_{\alpha/2}\right]. \tag{4.19}$$

当 σ^2 未知时, 由于

$$T = \frac{\frac{\overline{X}-\mu}{\sigma/\sqrt{n}}}{\sqrt{\frac{(n-1)S^2}{\sigma^2}/(n-1)}} = \frac{\left(\overline{X}-\mu\right)}{S/\sqrt{n}} \sim t(n-1), \tag{4.20}$$

有

$$P\left\{\left|\frac{(\overline{X}-\mu)}{S/\sqrt{n}}\right| \le t_{\alpha/2}\right\} = 1-\alpha, \tag{4.21}$$

其中 $t_\alpha(n-1)$ 表示自由度为 $n-1$ 的为 $t-$ 分布上 α 分位点. 由式 (4.21) 得到关于均值 μ, 置信度为 $1-\alpha$ 的双侧置信区间

$$\left[\overline{X} - \frac{S}{\sqrt{n}}t_{\alpha/2}(n-1),\ \ \overline{X} + \frac{S}{\sqrt{n}}t_{\alpha/2}(n-1)\right]. \tag{4.22}$$

根据公式 (4.19) 和公式 (4.22) 写出总体方差已知和方差未知两种情况均值 μ 区间估计的 R 程序 (程序名： `interval_estimate1.R`).

```
interval_estimate1<-function(x, sigma=-1, alpha=0.05){
   n<-length(x); xb<-mean(x)
   if (sigma>=0){
      tmp<-sigma/sqrt(n)*qnorm(1-alpha/2); df<-n
   }
   else{
      tmp<-sd(x)/sqrt(n)*qt(1-alpha/2,n-1); df<-n-1
   }
   data.frame(mean=xb, df=df, a=xb-tmp, b=xb+tmp)
}
```

在程序中， `x` 是来自总体的数据 (样本) 构成的向量. `sigma` 是总体的标准差, 当标准差已知时, 输入相应的值, 程序采用正态分布计算区间端点, 当标准差未知时, 输入项可缺省, 程序采用 $t-$ 分布计算区间端点. `alpha` 是显著性水平, 缺省值为 0.05. 函数以数据框的形式输出, 输出的内容有：样本均值 `mean`, 自由度 `df` 和均值区间估计的上下限 `a`, `b`.

注意：在 R 软件中, 所有的分位点均是按下分位点计算的, 而本书中的数学表达式所使用的分位点均是上分位点, 因此数学表达式与 R 软件中的函数有如下关系

$$Z_\alpha = \texttt{qnorm(1-alpha)}, \quad t_\alpha(n-1) = \texttt{qt(1-alpha, n-1)}.$$

其他分布函数也相同. 请注意两者的差别, 在编程中不要混淆.

在得到观测数据后, 可以用此函数对参数 μ 作区间估计.

例 4.14 某工厂生产的零件长度 X 被认为服从 $N(\mu, 0.04)$, 现从该产品中随机抽取 *6* 个，其长度的测量值如下（单位：毫米）

14.6, 15.1, 14.9, 14.8, 15.2, 15.1,

试求该零件长度的置信系数为 *0.95* 的区间估计.

解: 输入数据，调用函数 interval_estimate1()(程序名： exam0414.R)

```
X<-c(14.6, 15.1,14.9, 14.8, 15.2, 15.1)
source("interval_estimate.R")
interval_estimate(X, sigma=0.2)
```

得到

```
   mean df        a        b
1 14.95  6 14.78997 15.11003
```

因此，该零件长度的置信系数为 0.95 的置信区间为 $[14.79, 15.11]$.

例 4.15 为估计一件物体的重量 μ, 将其称了 *10* 次，得到的重量（单位：千克）为

10.1, 10, 9.8, 10.5, 9.7, 10.1, 9.9, 10.2, 10.3, 9.9,

假设所称出的物体重量服从 $N(\mu, \sigma^2)$, 求该物体 μ 置信系数为 *0.95* 的置信区间.

解: 输入数据，调用函数 interval_estimate1()(程序名： exam0415.R)

```
X<-c(10.1, 10, 9.8, 10.5, 9.7, 10.1, 9.9, 10.2, 10.3, 9.9)
source("interval_estimate.R")
interval_estimate(X)
```

得到

```
   mean df        a        b
1 10.05  9 9.877225 10.22278
```

因此，该物体 μ 置信系数为 0.95 置信区间为 $[9.87, 10.22]$.

R 软件中的 t.test 检验函数可以完成相应的区间估计工作，例如

```
> t.test(X)

        One Sample t-test

data:  X
t = 131.5854, df = 9, p-value = 4.296e-16
```

```
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
  9.877225 10.222775
sample estimates:
mean of x
    10.05
```

得到相应的区间估计 $[9.88, 10.22]$ 和其它的一些信息. 注意到：由 `t.test()` 函数得到的区间估计与我们编写函数得到的区间估计是相同的, 从这里可以帮助大家了解 `t.test()` 的计算过程. 关于 `t.test()` 函数进一步的使用方法将在下一章介绍.

2. 方差 σ^2 的区间估计

分别讨论总体 X 均值 μ 已知和均值 μ 未知两种情形.

当 μ 是已知时, 用 σ^2 的极大似然估计

$$\widehat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)^2 \tag{4.23}$$

来导出 σ^2 的置信区间. 由 χ^2 分布的定义容易推出

$$\frac{n\widehat{\sigma}^2}{\sigma^2} = \sum_{i=1}^{n}(X_i - \mu)^2 \Big/ \sigma^2 \quad \sim \quad \chi^2(n). \tag{4.24}$$

因此有

$$P\left\{\chi^2_{1-\alpha/2}(n) \le \frac{n\widehat{\sigma}^2}{\sigma^2} \le \chi^2_{\alpha/2}(n)\right\} = 1 - \alpha, \tag{4.25}$$

其中 $\chi^2_{1-\alpha/2}(n)$ 和 $\chi^2_{\alpha/2}(n)$ 分别表示自由度为 n 的为 χ^2- 分布上 $1-\alpha/2$ 和 $\alpha/2$ 分位点. 由此得到 σ^2 的置信度为 $1-\alpha$ 的双侧置信区间

$$\left[\frac{n\widehat{\sigma}^2}{\chi^2_{\alpha/2}(n)}, \quad \frac{n\widehat{\sigma}2}{\chi^2_{1-\alpha/2}(n)}\right]. \tag{4.26}$$

当 μ 是未知时, σ^2 的极大似然估计

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2$$

且满足

$$\frac{(n-1)S^2}{\sigma^2} \quad \sim \quad \chi^2(n-1), \tag{4.27}$$

因此，有

$$P\left\{\chi^2_{1-\alpha/2}(n-1) \le \frac{(n-1)S^2}{\sigma^2} \le \chi^2_{\alpha/2}(n-1)\right\} = 1-\alpha,$$

其中 $\chi^2_{1-\alpha/2}(n-1)$ 和 $\chi^2_{\alpha/2}(n-1)$ 分别表示自由度为 $n-1$ 的为 χ^2- 分布上 $1-\alpha/2$ 和 $\alpha/2$ 分位点. 由此得到 σ^2 的置信度为 $1-\alpha$ 的双侧置信区间

$$\left[\frac{(n-1)S^2}{\chi^2_{\alpha/2}(n-1)}, \quad \frac{(n-1)S^2}{\chi^2_{1-\alpha/2}(n-1)}\right]. \tag{4.28}$$

根据公式 (4.26) 和公式 (4.28) 写出总体均值已知和均值未知两种情况方差 σ^2 区间估计的 R 程序（程序名： `interval_var1.R` ）

```
interval_var1<-function(x, mu=Inf, alpha=0.05){
   n<-length(x)
   if (mu<Inf){
      S2 <- sum((x-mu)^2)/n; df <- n
   }
   else{
      S2 <- var(x); df <- n-1
   }
   a<-df*S2/qchisq(1-alpha/2,df)
   b<-df*S2/qchisq(alpha/2,df)
   data.frame(var=S2, df=df, a=a, b=b)
}
```

在程序中， `x` 是由来自总体的数据 (样本) 构成的向量. `mu` 是总体均值，当均值已知时，输入相应的值，程序采用自由度为 n 的 χ^2- 分布计算区间端点. 当均值未知时，输入项可缺省，程序采用自由度为 $n-1$ 的 χ^2- 分布计算区间端点. 数据输出采用数据框的形式，输出值是样本方差 `var`，自由度 `df` 和方差的区间估计 `a`，`b`.

例 4.16 用区间估计方法估计例 *4.15* 的测量误差（即方差 σ^2 ），分别对均值 μ 已知 ($\mu = 10$) 和均值 μ 未知两种情况进行讨论.

解: 用上面编好的函数计算.

```
#### 输入数据，调用编好的程序
> X<-c(10.1,10,9.8,10.5,9.7,10.1,9.9,10.2,10.3,9.9)
> source("interval_var1.R")
#### 作方差的区间估计，认为均值已知
> interval_var1(X, mu=10)
    var df          a         b
1 0.055 10 0.02685130 0.1693885
#### 作方差的区间估计，认为均值未知
> interval_var1(X)
         var df          a         b
1 0.05833333  9 0.02759851 0.1944164
```

当均值已知 ($\mu = 10$) 时，其方差 σ^2 的区间估计为 $[0.0268, 0.169]$, 当均值未知时，其方差 σ^2 的区间估计为 $[0.0276, 0.194]$. 从计算结果来看，在均值已知的情况下，计算结果更好一些.

4.3.2 两个正态总体的情况

假设有两个正态总体 $X \sim N(\mu_1, \sigma_1^2)$ 和 $Y \sim N(\mu_2, \sigma_2^2)$, $X_1, X_2, \cdots, X_{n_1}$ 为来自总体 X 的一个样本， $Y_1, Y_2, \cdots, Y_{n_2}$ 为来自总体 Y 的一个样本， $1-\alpha$ 为置信度， $\overline{X}, \overline{Y}$ 分别为第一、第二样本均值， S_1^2, S_2^2 分别为第一、第二样本方差.

1. 均值差 $\mu_1 - \mu_2$ 的区间估计

分三种情况讨论.

(1) 当两总体的方差 σ_1^2, σ_2^2 已知时，由正态分布的性质有

$$\overline{X} - \overline{Y} \sim N\left(\mu_1 - \mu_2, \frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}\right), \tag{4.29}$$

类似于单个总体区间估计的推导，得到 $\mu_1 - \mu_2$ 的置信度为 $1-\alpha$ 的双侧置信区间:

$$\left[\overline{X} - \overline{Y} - Z_{\alpha/2}\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}},\ \ \overline{X} - \overline{Y} + Z_{\alpha/2}\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}\right]. \tag{4.30}$$

(2) 当两总体的方差相同，即 $\sigma_1^2=\sigma_2^2=\sigma^2$, 且未知时，可以得到

$$T=\frac{\overline{X}-\overline{Y}-(\mu_1-\mu_2)}{S_w\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}}\ \sim\ t(n_1+n_2-2), \tag{4.31}$$

其中

$$S_w=\sqrt{\frac{(n_1-1)S_1^2+(n_2-1)S_2^2}{n_1+n_2-2}}. \tag{4.32}$$

仿照式 (4.22) 的推导，得到 $\mu_1-\mu_2$ 的置信度为 $1-\alpha$ 的双侧置信区间：

$$\begin{aligned}\Big[\ &\overline{X}-\overline{Y}-t_{\alpha/2}(n_1+n_2-2)S_w\sqrt{\tfrac{1}{n_1}+\tfrac{1}{n_2}},\\ &\overline{X}-\overline{Y}+t_{\alpha/2}(n_1+n_2-2)S_w\sqrt{\tfrac{1}{n_1}+\tfrac{1}{n_2}}\ \Big].\end{aligned} \tag{4.33}$$

(3) 当两总体的方差 σ_1^2 和 σ_2^2 未知，且 $\sigma_1^2\neq\sigma_2^2$ 时，可以证明

$$T=\frac{\overline{X}-\overline{Y}-(\mu_1-\mu_2)}{\sqrt{\frac{S_1^2}{n_1}+\frac{S_2^2}{n_2}}}\ \sim\ t(\nu) \tag{4.34}$$

近似成立，其中

$$\nu=\left(\frac{\sigma_1^2}{n_1}+\frac{\sigma_2^2}{n_2}\right)^2\Big/\left(\frac{(\sigma_1^2)^2}{n_1^2(n_1-1)}+\frac{(\sigma_2^2)^2}{n_2^2(n_2-1)}\right). \tag{4.35}$$

但由于 σ_1^2,σ_2^2 未知，用样本方差 S_1^2,S_2^2 似来近似，因此，

$$\widehat{\nu}=\left(\frac{S_1^2}{n_1}+\frac{S_2^2}{n_2}\right)^2\Big/\left(\frac{(S_1^2)^2}{n_1^2(n_1-1)}+\frac{(S_2^2)^2}{n_2^2(n_2-1)}\right). \tag{4.36}$$

可以近似地认为

$$T\ \ \sim\ \ t(\widehat{\nu}).$$

由此得到 $\mu_1-\mu_2$ 的置信度为 $1-\alpha$ 的双侧置信区间：

$$\left[\overline{X}-\overline{Y}-t_{\alpha/2}(\widehat{\nu})\sqrt{\frac{S_1^2}{n_1}+\frac{S_2^2}{n_2}},\ \ \overline{Y}-\overline{X}+t_{\alpha/2}(\widehat{\nu})\sqrt{\frac{S_1^2}{n_1}+\frac{S_2^2}{n_2}}\right]. \tag{4.37}$$

根据公式 (4.30) 、公式 (4.33) 和公式 (4.37) 写出三种情况下均值差 $\mu_1-\mu_2$ 区间估计的 R 程序 (程序名： `interval_estimate2.R`).

```
interval_estimate2<-function(x, y,
   sigma=c(-1,-1), var.equal=FALSE, alpha=0.05){
   n1<-length(x); n2<-length(y)
   xb<-mean(x); yb<-mean(y)
   if (all(sigma>=0)){
      tmp<-qnorm(1-alpha/2)*sqrt(sigma[1]^2/n1+sigma[2]^2/n2)
      df<-n1+n2
   }
   else{
      if (var.equal ==  TRUE){
         Sw<-((n1-1)*var(x)+(n2-1)*var(y))/(n1+n2-2)
         tmp<-sqrt(Sw*(1/n1+1/n2))*qt(1-alpha/2,n1+n2-2)
         df<-n1+n2-2
      }
      else{
         S1<-var(x); S2<-var(y)
         nu<-(S1/n1+S2/n2)^2/(S1^2/n1^2/(n1-1)+S2^2/n2^2/(n2-1))
         tmp<-qt(1-alpha/2, nu)*sqrt(S1/n1+S2/n2)
         df<-nu
      }
   }
   data.frame(mean=xb-yb, df=df, a=xb-yb-tmp, b=xb-yb+tmp)
}
```

在程序中， x，y 分别是来自两总体的数据 (样本) 构成的向量. sigma 是由两总体标准差构成的向量，当标准差已知时，输入相应的值，程序采用正态分布计算区间的端点. 当标准差未知时，输入项可缺省，此时需要考虑两总体的方差是否相同：若认为两总体方差相同，输入 var.equal = TRUE, 程序采用自由度为 n_1+n_2-2 的 $t-$ 分布计算区间端点；若认为两总体方差不同，输入 var.equal = FALSE (或缺省), 程序采用自由度为 ν 的 $t-$ 分布计算区间端点. 当 ν 不是整数时，程序在计算 $t-$ 分布时，其值采用插值方法得到.

程序输出采用数据框的形式，输出两样本均值差 mean, 自由度 df, 和均值差

的区间估计的端点 a，b.

例 4.17 欲比较甲、乙两种棉花品种的优劣. 现假设用它们纺出的棉纱强度分别服从 $N(\mu_1, 2.18^2)$ 和 $N(\mu_2, 1.76^2)$, 试验者从这两种棉纱中分别抽取样本 $X_1, X_2, \cdots, X_{100}$ 和 $Y_1, Y_2, \cdots, Y_{100}$ (其数据用计算机随机产生，其随机数的均值分别为 $\mu_1 = 5.32$, $\mu_2 = 5.76$). 试给出 $\mu_1 - \mu_2$ 的置信系数为 *0.95* 的区间估计.

解: 首先用 R 软件产生 200 个随机数，再调用函数 interval_estimate2() 进行计算 (程序名： exam_0417.R).

```
x<-rnorm(100, 5.32, 2.18)
y<-rnorm(100, 5.76, 1.76)
source("interval_estimate2.R")
interval_estimate2(x,y, sigma=c(2.18, 1.76))
```

得到计算结果

```
        mean  df        a         b
1 -0.2549302 200 -0.80407 0.2942096
```

因此， $\mu_1 - \mu_2$ 的置信系数为 0.95 的区间估计为 $[-0.804, 0.294]$.

注意：由于数据是由计算机随机产生的，因此，每一次的计算结果是不相同的，但总的趋势是相同的.

例 4.18 某公司利用两条自动化流水线灌装矿泉水. 现从生产线上随机抽取样本 $X_1, X_2, \cdots, X_{12}$ 和 $Y_1, Y_2, \cdots, Y_{17}$ (数据由计算机模拟产生)，它们是每瓶矿泉水的体积 (毫升). 假设这两条流水线所装的矿泉水的体积都服从正态分布，分别为 $N(\mu_1, \sigma^2)$ 和 $N(\mu_2, \sigma^2)$. 给定置信系数 *0.95* ，试求 $\mu_1 - \mu_2$ 的区间估计. 讨论两种情况， *(1)* 两总体方差相同； *(2)* 两总体方差不同. (注：计算机产生随机数的均值 $\mu_1 = 501.1$ 和 $\mu_2 = 499.7$, 标准差 $\sigma_1 = 2.4$, $\sigma_2 = 4.7$.)

解: 首先用 R 软件产生相应的随机数，再调用函数 interval_estimate2() 进行计算 (程序名： exam_0418.R).

```
x<-rnorm(12, 501.1, 2.4)
y<-rnorm(17, 499.7, 4.7)
source("interval_estimate2.R")
interval_estimate2(x, y, var.equal=TRUE)
interval_estimate2(x, y)
```

认为方差相同的计算结果是

```
> interval_estimate2(x, y, var.equal=TRUE)
        mean df         a        b
1 -0.7120126 27 -3.667566 2.243541
```

因此，在认为方差相同的情况下， $\mu_1 - \mu_2$ 的置信系数为 0.95 的区间估计为 $[-3.67, 2.24]$.

认为方差不同的计算结果是

```
> interval_estimate2(x, y)
        mean       df         a        b
1 -0.7120126 23.09151 -3.344401 1.920376
```

因此，在认为方差不同的情况下， $\mu_1 - \mu_2$ 的置信系数为 0.95 的区间估计为 $[-3.34, 1.92]$.

两计算结果作比较，可认为在两总体方差不同的假设下，计算结果更精确一些.

在这两个例子中， $\mu_1 - \mu_2$ 的区间估计包含了零，也就是说， μ_1 可能大于 μ_2, 也可能小于 μ_2, 这时我们就认为 μ_1 与 μ_2 并没有显著差异.

R 软件中的 t.test() 函数可以给出双样本差的区间估计，如

```
> t.test(x, y)
        Welch Two Sample t-test
data:  x and y
t = -0.5594, df = 23.092, p-value = 0.5813
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.344401  1.920376
sample estimates:
mean of x mean of y
 500.0234  500.7354
```

由于没有声明，在计算时总认为两样本方差是不同的. 如果认为方差相同，需要声明，即在变量中给出 var.equal=TRUE, 如

```
> t.test(x, y, var.equal=TRUE)
        Two Sample t-test
data:  x and y
```

```
t = -0.4943, df = 27, p-value = 0.6251
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.667566  2.243541
sample estimates:
mean of x mean of y
 500.0234  500.7354
```

比较两种程序的计算结果，发现由 t.test() 函数得到的计算结果与我们编写函数的计算结果是完全相同的，结合前面的例子，帮助我们理解 t.test() 的函数的计算过程. 有关 t.test() 函数的其他用法，后面还会讨论.

2. 配对数据的区间估计

因为配对数据的每一对都可计算其差值 d, 所以，虽然配对数据是两组数据间的比较，但求出每一对差值后，就变成了单个样本了，其置信区间可按单个总体均值 μ 的区间估计的方法求出. 这里也可以分成方差 σ_d^2 已知和方差 σ_d^2 未知的情况来讨论. 由于前面对单个总体样本均值估计讨论的比较仔细，这里只给出其应用方法.

例 4.19 *为了调查应用克矽平治疗矽肺的效果，今抽查应用克矽平治疗矽肺的患者 10 名，记录下治疗前后血红蛋白的含量数据，如表 4.1 所示. 试求治疗前后变化的区间估计 ($\alpha = 0.05$).*

表 4.1: 治疗前后血红蛋白的含量数据

病人编号	1	2	3	4	5	6	7	8	9	10
治疗前 (X)	11.3	15.0	15.0	13.5	12.8	10.0	11.0	12.0	13.0	12.3
治疗后 (Y)	14.0	13.8	14.0	13.5	13.5	12.0	14.7	11.4	13.8	12.0

解: 输入数据，调入 t.test() 函数.

```
> X<-c(11.3, 15.0, 15.0, 13.5, 12.8, 10.0, 11.0, 12.0, 13.0, 12.3)
> Y<-c(14.0, 13.8, 14.0, 13.5, 13.5, 12.0, 14.7, 11.4, 13.8, 12.0)
> t.test(X-Y)

        One Sample t-test
```

```
data:  X - Y
t = -1.3066, df = 9, p-value = 0.2237
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -1.8572881  0.4972881
sample estimates:
mean of x
    -0.68
```

所以得到，治疗前后变化的区间估计为 $[-1.86, 0.497]$.

由于 0 包含在区间估计的区间内，因此可以认为：治疗前后病人血红蛋白的含量无显著差异. 关于假设检验部分我们在下章再介绍.

3. 方差比 σ_1^2/σ_2^2 的区间估计

仍分总体均值 μ_1 、 μ_2 已知和总体均值 μ_1 、 μ_2 未知两种情况讨论.

(1) μ_1 与 μ_2 已知. 此时

$$\widehat{\sigma}_1^2 = \frac{1}{n_1}\sum_{i=1}^{n_1}(X_i-\mu_1)^2, \quad \widehat{\sigma}_2^2 = \frac{1}{n_2}\sum_{i=1}^{n_2}(Y_i-\mu_2)^2 \tag{4.38}$$

分别为 σ_1^2 和 σ_2^2 的最小无偏估计，由于

$$F = \frac{\widehat{\sigma}_1^2/\sigma_1^2}{\widehat{\sigma}_2^2/\sigma_2^2} \quad \sim \quad F(n_1, n_2), \tag{4.39}$$

因此

$$P\left\{F_{1-\alpha/2}(n_1,n_2) \le \frac{\widehat{\sigma}_1^2/\sigma_1^2}{\widehat{\sigma}_2^2/\sigma_2^2} \le F_{\alpha/2}(n_1,n_2)\right\} = 1-\alpha, \tag{4.40}$$

其中 $F_{1-\alpha/2}(n_1,n_2)$ 和 $F_{\alpha/2}(n_1,n_2)$ 分别表示自由度为 (n_1,n_2) 的为 $F-$ 分布上 $1-\alpha/2$ 和 $\alpha/2$ 分位点. 因此， σ_1^2/σ_2^2 的置信水平 $1-\alpha$ 的置信区间为

$$\left[\frac{\widehat{\sigma}_1^2/\widehat{\sigma}_2^2}{F_{\alpha/2}(n_1,n_2)}, \frac{\widehat{\sigma}_1^2/\widehat{\sigma}_2^2}{F_{1-\alpha/2}(n_1,n_2)}\right]. \tag{4.41}$$

(2) μ_1 与 μ_2 未知. 此时 S_1^2 和 S_2^2 分别为 σ_1^2 和 σ_2^2 的最小无偏估计，由于

$$F = \frac{S_1^2/\sigma_1^2}{S_2^2/\sigma_2^2} \quad \sim \quad F(n_1-1, n_2-1), \tag{4.42}$$

因此

$$P\left\{F_{1-\alpha/2}(n_1-1,n_2-1)\le\frac{S_1^2/\sigma_1^2}{S_2^2/\sigma_2^2}\le F_{\alpha/2}(n_1-1,n_2-1)\right\}=1-\alpha,\quad(4.43)$$

则 σ_1^2/σ_2^2 的置信水平 $1-\alpha$ 的置信区间为

$$\left[\frac{S_1^2/S_2^2}{F_{\alpha/2}(n_1-1,n_2-1)},\frac{S_1^2/S_2^2}{F_{1-\alpha/2}(n_1-1,n_2-2)}\right].\quad(4.44)$$

根据公式 (4.41) 和公式 (4.44) 写出上述两种情况下方差比 σ_1^2/σ_2^2 区间估计的 R 程序 (程序名： interval_var2.R).

```
interval_var2<-function(x,y,
   mu=c(Inf, Inf), alpha=0.05){
   n1<-length(x); n2<-length(y)
   if (all(mu<Inf)){
      Sx2<-1/n1*sum((x-mu[1])^2); Sy2<-1/n2*sum((y-mu[2])^2)
      df1<-n1; df2<-n2
   }
   else{
      Sx2<-var(x); Sy2<-var(y); df1<-n1-1; df2<-n2-1
   }
   r<-Sx2/Sy2
   a<-r/qf(1-alpha/2,df1,df2)
   b<-r/qf(alpha/2,df1,df2)
   data.frame(rate=r, df1=df1, df2=df2, a=a, b=b)
}
```

在程序中， x，y 分别是来自两总体的数据 (样本) 构成的向量. mu 是由两总体均值构成的向量, 当均值已知时, 输入相应的值, 程序采用自由度为 (n_1,n_2) 的 F– 分布计算区间估计的两个端点; 否则 (输入值缺省), 程序采用自由度为 (n_1-1,n_2-1) 的 F– 分布计算区间估计的两个端点. alpha 是显著性水平, 缺省值为 0.05. 输出采用数据框形式, 输出的变量有：样本方差比 rate, 第一自由度 df1, 第二自由度 df2, 和方差比的区间估计的端点 a，b.

例 4.20 已知两组数据

A: 79.98 80.04 80.02 80.04 80.03 80.03 80.04 79.97
80.05 80.03 80.02 80.00 80.02
B: 80.02 79.94 79.98 79.97 79.97 80.03 79.95 79.97

试用两种方法作方差比的区间估计. *(1)* 均值已知 $\mu_1 = \mu_2 = 80$*; (2)* 均值未知.

解: 输入数据, 调用函数 interval_var2() 进行计算 (程序名：exam0419.R).

```
#### 用 scan() 函数输入数据
> A<-scan()
1: 79.98 80.04 80.02 80.04 80.03 80.03 80.04 79.97
9: 80.05 80.03 80.02 80.00 80.02
14:
Read 13 items
> B<-scan()
1: 80.02 79.94 79.98 79.97 79.97 80.03 79.95 79.97
9:
Read 8 items
#### 调用编好的程序
> source("interval_var2.R")
#### 方差比的区间估计, 认为均值已知
> interval_var2(A, B, mu=c(80,80))
       rate df1 df2         a        b
1 0.7326007  13   8 0.1760141 2.482042
#### 方差比的区间估计, 认为均值未知
> interval_var2(A, B)
       rate df1 df2         a        b
1 0.5837405  12   7 0.1251097 2.105269
```

两种计算结果稍有差异.

从计算结果可以看到, 1 包含在区间估计的区间中, 也就是说, 有理由认为两总体的方差比为 1, 即可认为两总体的的方差是相同的.

在 R 软件中, var.test() 函数能够提供双样本方差比的区间估计, 如

```
> var.test(A,B)

        F test to compare two variances

data:  A and B
F = 0.5837, num df = 12, denom df = 7, p-value = 0.3938
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.1251097 2.1052687
sample estimates:
ratio of variances
         0.5837405
```

与我们所编写函数的计算结果相同 (均值未知), 从这里也可以帮助我们理解函数 var.test() 的计算过程. 有关 var.test() 函数的其他用法, 后面的内容中还会进行讨论.

4.3.3 非正态总体的区间估计

当数据不服从正态分布时, 估计均值的一种有效的方法就是所谓的大样本方法, 即要求样本的量比较大, 利用中心极限定理进行分析.

设总体 X 均值为 μ, 方差为 σ^2, $X_1, X_2, \cdots, X_n$ 为抽自总体 X 的一个样本. 因为这些样本是独立同分布的, 根据中心极限定理, 对于充分大的 n, 有

$$\frac{\sum_{i=1}^{n} X_i - n\mu}{\sqrt{n}\sigma} \sim N(0,1)$$

近似成立, 这样就导出 μ 的置信度为 $1-\alpha$ 的双侧近似置信区间

$$\left[\overline{X} - \frac{\sigma}{\sqrt{n}} Z_{\alpha/2},\ \ \overline{X} + \frac{\sigma}{\sqrt{n}} Z_{\alpha/2}\right]. \tag{4.45}$$

在形式上, 该式与式 (4.19) 完全相同, 所不同的是这里的置信系数是近似的.

如果方差 σ^2 是未知的, 可以用它的估计 S^2 来代替 σ^2, 由此得到相应的近似置信区间

$$\left[\overline{X} - \frac{S}{\sqrt{n}} Z_{\alpha/2},\ \ \overline{X} + \frac{S}{\sqrt{n}} Z_{\alpha/2}\right]. \tag{4.46}$$

根据公式 (4.45) 和公式 (4.46) 写出非正态总体区间估计的 R 程序 (程序名: interval_estimate3.R).

```
interval_estimate3<-function(x,sigma=-1,alpha=0.05){
   n<-length(x); xb<-mean(x)
   if (sigma>=0)
      tmp<-sigma/sqrt(n)*qnorm(1-alpha/2)
   else
      tmp<-sd(x)/sqrt(n)*qnorm(1-alpha/2)
   data.frame(mean=xb, a=xb-tmp, b=xb+tmp)
}
```

在程序中， x 是来自非正态分布总体的数据 (样本) 向量， sigma 是总体标准差, 当标准差已知时, 输入相应的标准差; 当标准差未知时, 输入项缺省, 程序用样本的标准差代替总体的标准差. 输出采用数据框形式, 输出样本均值 mean, 均值的区间估计的两个端点 a,b.

例 4.21 某公司欲估计自己生产的电池寿命. 现从其产品中随机抽取 50 只电池做寿命试验 (数据由计算机随机产生, 服从均值 $1/\lambda = 2.266$ (单位： 100 小时) 的指数分布). 求该公司生产的电池平均寿命的置信系数为 95% 的置信区间.

解: 首先用 R 软件产生相应的随机数, 再调用函数 interval_estimate3() 进行计算.

```
> x<-rexp(50, 1/2.266)
> source("interval_estimate3.R")
> interval_estimate3(x)
      mean        a        b
1 2.293804 1.612363 2.975244
```

因此, 该公司电池的平均寿命的置信系数约为 95% 的置信区间为 $[1.612, 2.975]$.

4.3.4 单侧置信区间估计

对于某些问题, 人们只关心 θ 在某一方向上的界限. 例如, 对于设备、元件的寿命来说, 我们常常关心的是平均寿命 θ 的 “下限”. 而当我们考虑产品的废品率 p 时, 关心的是参数 p 的 “上界”. 称这类区间估计问题为单侧区间估计.

定义 4.7 设 $X_1, X_2, \cdots, X_n$ 是来自总体 X 的一个样本, θ 是包含在总体分布中的未知参数, 对于给定的 $\alpha(0 < \alpha < 1)$, 若统计量 $\underline{\theta} = \underline{\theta}(X_1, X_2, \cdots, X_n)$ 满足

$$P\{\underline{\theta}(X_1, X_2, \cdots, X_n) \leq \theta\} = 1 - \alpha,$$

则称随机区间 $[\underline{\theta},\ +\infty)$ 是 θ 的置信度为 $1-\alpha$ 的单侧置信区间，称 $\underline{\theta}$ 为 θ 的置信度为 $1-\alpha$ 的单侧置信下限. 若统计量 $\overline{\theta}=\overline{\theta}(X_1,X_2,\cdots,X_n)$ 满足

$$P\left\{\theta\leq\overline{\theta}(X_1,X_2,\cdots,X_n)\right\}=1-\alpha,$$

则称随机区间 $(-\infty,\ \overline{\theta}]$ 是 θ 的置信度为 $1-\alpha$ 的单侧置信区间，称 $\overline{\theta}$ 为 θ 的置信度为 $1-\alpha$ 的单侧置信上限.

类似于双侧置信区间估计的研究，对于给定的置信度 $1-\alpha$, 选择置信下限 $\underline{\theta}$ 时，应是 $E(\underline{\theta})$ 越大越好，而选择置信上限 $\overline{\theta}$ 时，应是 $E(\overline{\theta})$ 越小越好.

1．一个总体求均值

假设正态总体 $X\ \sim\ N(\mu,\sigma^2)$, $X_1,X_2,\cdots,X_n$ 为来自总体 X 的一个样本，$1-\alpha$ 为置信度，$\overline{X}$ 为样本均值，S^2 为样本方差.

分别讨论总体均值 σ^2 已知和未知情况下，均值 μ 的单侧置信区间估计.

若 σ^2 已知，由式 (4.17), 得到

$$P\left\{\frac{\overline{X}-\mu}{\sigma/\sqrt{n}}\leq Z_\alpha\right\}=1-\alpha,\quad P\left\{-Z_\alpha\leq\frac{\overline{X}-\mu}{\sigma/\sqrt{n}}\right\}=1-\alpha.$$

于是得到 μ 的置信度为 $1-\alpha$ 的单侧置信区间

$$\left[\overline{X}-\frac{\sigma}{\sqrt{n}}Z_\alpha,\ +\infty\right),\quad\left(-\infty,\ \overline{X}+\frac{\sigma}{\sqrt{n}}Z_\alpha\right].\tag{4.47}$$

因此，μ 的置信度为 $1-\alpha$ 的单侧置信下限、上限分别为

$$\underline{\mu}=\overline{X}-\frac{\sigma}{\sqrt{n}}Z_\alpha,\quad\overline{\mu}=\overline{X}+\frac{\sigma}{\sqrt{n}}Z_\alpha.\tag{4.48}$$

若 σ^2 未知，由式 (4.20), 得到

$$P\left\{\frac{\overline{X}-\mu}{S/\sqrt{n}}\leq t_\alpha(n-1)\right\}=1-\alpha,\quad P\left\{-t_\alpha(n-1)\leq\frac{\overline{X}-\mu}{S/\sqrt{n}}\right\}=1-\alpha,$$

于是得到 μ 的置信度为 $1-\alpha$ 的单侧置信区间

$$\left[\overline{X}-\frac{S}{\sqrt{n}}t_\alpha(n-1),\ +\infty\right),\quad\left(-\infty,\ \overline{X}+\frac{S}{\sqrt{n}}t_\alpha(n-1)\right].\tag{4.49}$$

因此， μ 的置信度为 $1-\alpha$ 的单侧置信下限、上限分别为

$$\underline{\mu} = \overline{X} - \frac{S}{\sqrt{n}} t_\alpha(n-1), \quad \overline{\mu} = \overline{X} + \frac{S}{\sqrt{n}} t_\alpha(n-1). \tag{4.50}$$

根据公式 (4.47) 和公式 (4.49), 以及双侧置信区间的公式写出下面的 R 程序（程序名： `interval_estimate4.R` ），并可控制求上、下置信区间或双侧置信区间.

```
interval_estimate4<-function(x, sigma=-1, side=0, alpha=0.05){
   n<-length(x); xb<-mean(x)
   if (sigma>=0){
      if (side<0){
         tmp<-sigma/sqrt(n)*qnorm(1-alpha)
         a <- -Inf; b <- xb+tmp
      }
      else if (side>0){
         tmp<-sigma/sqrt(n)*qnorm(1-alpha)
         a <- xb-tmp; b <- Inf
      }
      else{
         tmp <- sigma/sqrt(n)*qnorm(1-alpha/2)
         a <- xb-tmp; b <- xb+tmp
      }
      df<-n
   }
   else{
      if (side<0){
         tmp <- sd(x)/sqrt(n)*qt(1-alpha,n-1)
         a <- -Inf; b <- xb+tmp
      }
      else if (side>0){
         tmp <- sd(x)/sqrt(n)*qt(1-alpha,n-1)
         a <- xb-tmp; b <- Inf
```

```
        }
        else{
           tmp <- sd(x)/sqrt(n)*qt(1-alpha/2,n-1)
           a <- xb-tmp; b <- xb+tmp
        }
        df<-n-1
     }
     data.frame(mean=xb, df=df, a=a, b=b)
}
```

在程序中, x 是由来自总体的数据 (样本) 构成的向量. sigma 是总体的标准差, 当标准差已知时, 输入相应的值, 程序采用正态分布估计区间端点; 否则 (输入项缺省), 程序采用 $t-$ 分布估计区间端点. side 是控制求置信区间上下限, 若求置信区间上限, 输入 side=-1; 若求置信区间下限, 输入 side=1; 若求双侧置信区间, 输入 side=0 或缺省. 输出采用数据框形式, 输出样本均值 mean, 自由度 df, 和均值的区间估计的两个端点 a,b.

上述程序实际上包含了求双侧置信区间的情况, 也就是说, 函数 interval_estimate4 包含了函数 interval_estimate1 的功能.

例 4.22 从一批灯泡中随机地取 *5* 只作寿命试验, 测得寿命 (以小时计) 为

1050, 1100, 1120, 1250, 1280.

设灯泡寿命服从正态分布, 求灯泡寿命平均值的置信度为 *0.95* 的单侧置信下限.

解: 输入数据, 调用函数 interval_estimate4()

```
> X<-c(1050, 1100, 1120, 1250, 1280)
> source("interval_estimate4.R")
> interval_estimate4(X, side=1)
  mean df        a   b
1 1160  4 1064.900 Inf
```

也就是说有 95% 的灯泡寿命在 1064.9 小时以上.

R 软件中的 t.test() 函数也可以完成单侧区间估计, 如

```
> t.test(X, alternative = "greater")
        One Sample t-test
data:  X
```

```
t = 26.0035, df = 4, p-value = 6.497e-06
alternative hypothesis: true mean is greater than 0
95 percent confidence interval:
 1064.900      Inf
sample estimates:
mean of x
     1160
```

相应的区间估计为 $[1064.900, \infty]$, 与我们编写函数具有相同的计算结果.

在程序中, `alternative` 是指备择假设, 这个概念将在下一章假设检验中作详细介绍.

2. 一个总体求方差

假设与前面相同, $\widehat{\sigma}^2$ 是由式 (4.23) 定义, 分别讨论总体均值 μ 已知、未知的情况, 方差 σ^2 的单侧置信区间估计.

当 μ 是已知时, 由式 (4.24), 有

$$P\left\{\frac{n\widehat{\sigma}^2}{\sigma^2} \leq \chi^2_{\alpha}(n)\right\} = 1-\alpha, \quad P\left\{\chi^2_{1-\alpha}(n) \leq \frac{n\widehat{\sigma}^2}{\sigma^2}\right\} = 1-\alpha,$$

于是得到 σ^2 的置信度为 $1-\alpha$ 的单侧置信区间

$$\left[\frac{n\widehat{\sigma}^2}{\chi^2_{\alpha}(n)},\ +\infty,\right), \quad \left[0,\ \frac{n\widehat{\sigma}^2}{\chi^2_{1-\alpha}(n)}\right]. \tag{4.51}$$

σ^2 的置信度为 $1-\alpha$ 的单侧置信下、上限为

$$\underline{\sigma^2} = \frac{n\widehat{\sigma}^2}{\chi^2_{\alpha}(n)}, \quad \overline{\sigma^2} = \frac{n\widehat{\sigma}^2}{\chi^2_{1-\alpha}(n)}. \tag{4.52}$$

当 μ 是未知时, 由式 (4.27), 有

$$P\left\{\frac{(n-1)S^2}{\sigma^2} \leq \chi^2_{\alpha}(n-1)\right\} = 1-\alpha,\ P\left\{\chi^2_{1-\alpha}(n-1) \leq \frac{(n-1)S^2}{\sigma^2}\right\} = 1-\alpha,$$

于是得到 σ^2 的置信度为 $1-\alpha$ 的单侧置信区间

$$\left[\frac{(n-1)S^2}{\chi^2_{\alpha}(n-1)},\ +\infty\right), \quad \left[0,\ \frac{(n-1)S^2}{\chi^2_{1-\alpha}(n-1)}\right]. \tag{4.53}$$

σ^2 的置信度为 $1-\alpha$ 的单侧置信下、上限为

$$\underline{\sigma^2}=\frac{(n-1)S^2}{\chi^2_{\alpha}(n-1)},\qquad \overline{\sigma^2}=\frac{(n-1)S^2}{\chi^2_{1-\alpha}(n-1)}. \tag{4.54}$$

根据公式 (4.51) 和公式 (4.54), 以及双侧置信区间的公式写出下面的 R 程序（程序名： interval_var3.R), 并可控制求上置信区间、双侧置信区间.

```
interval_var3<-function(x,mu=Inf,side=0,alpha=0.05){
   n<-length(x)
   if (mu<Inf){
      S2<-sum((x-mu)^2)/n; df<-n
   }
   else{
      S2<-var(x); df<-n-1
   }
   if (side<0){
      a <- 0
      b <- df*S2/qchisq(alpha,df)
   }
   else if (side>0){
      a <- df*S2/qchisq(1-alpha,df)
      b <- Inf
   }
   else{
      a<-df*S2/qchisq(1-alpha/2,df)
      b<-df*S2/qchisq(alpha/2,df)
   }
   data.frame(var=S2, df=df, a=a, b=b)
}
```

在程序中， x 是来自总体的数据 (样本) 构成的向量. mu 是总体均值，当均值已知时，输入相应的值，程序采用自由度为 n 的 χ^2- 分布计算区间端点；当均值未知时，输入项可缺省，程序采用自由度为 $n-1$ 的 χ^2- 分布计算区间端点. side 是控制求置信区间上下限，若求置信区间上限，输入 side=-1; 若求置

信区间下限，输入 side=1; 若求双侧置信区间，输入 side=0 或缺省. 数据输出采用数据框的形式，输出值是样本方差 var, 自由度 df 和方差的区间估计 a，b.

事实上，此函数已包含了前面讲过的方差的区间估计函数 interval_var1 的功能.

例 4.23 *求例 4.21 中 10 个数据的方差置信区间上限* $(\alpha = 0.05)$.

解: 输入数据，调用函数 interval_var3()

```
> X<-c(10.1,10,9.8,10.5,9.7,10.1,9.9,10.2,10.3,9.9)
> source("interval_var3.R")
> interval_var3(X, side=-1)
          var df a         b
1 0.05833333  9 0 0.1578894
```

σ^2 的置信上限为 0.1579.

关于单侧置信区间估计本质上与双侧置信区间估计是相同的, 不同的只是考虑区间的一侧，因此，前面介绍双侧估计的方法，基本上可以平行的移到单侧区间估计中，有关的 R 软件编程，原则上也是相同的.

3. 两个总体求均值差

假设有两个正态总体 $X \sim N(\mu_1, \sigma_1^2)$ 和 $Y \sim N(\mu_2, \sigma_2^2)$, $X_1, X_2, \cdots, X_{n_1}$ 为来自总体 X 的一个样本， $Y_1, Y_2, \cdots, Y_{n_2}$ 为来自总体 Y 的一个样本， $1-\alpha$ 为置信度， $\overline{X}, \overline{Y}$ 分别为第一、第二样本均值， S_1^2, S_2^2 分别为第一、第二样本方差.

分别讨论两总体的方差 σ_1^2, σ_2^2 已知、未知和是否相同情况下, 均值差 $\mu_1 - \mu_2$ 的单侧置信区间估计.

当 σ_1^2, σ_2^2 已知时，由式 (4.29) 和类似于双侧置信区间的估计的推导，得到 $\mu_1 - \mu_2$ 的置信度为 $1-\alpha$ 的单侧置信区间:

$$\left[\overline{X} - \overline{Y} - Z_\alpha\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}},\ +\infty\right), \quad \left(-\infty,\ \overline{X} - \overline{Y} + Z_\alpha\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}\right]. \quad (4.55)$$

当 $\sigma_1^2 = \sigma_2^2 = \sigma^2$, 且未知时，由式 (4.31) 和类似于双侧置信区间的估计的推导，得到 $\mu_1 - \mu_2$ 的置信度为 $1-\alpha$ 的单侧置信区间:

$$\left[\overline{X} - \overline{Y} - t_\alpha(n_1 + n_2 - 2)S_w\sqrt{\frac{1}{n_1} + \frac{1}{n_2}},\ \ +\infty\right), \quad (4.56)$$

和

$$\left(-\infty,\ \ \overline{X}-\overline{Y}+t_\alpha(n_1+n_2-2)S_w\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}\right]. \tag{4.57}$$

当 σ_1^2 和 σ_2^2 未知, 且 $\sigma_1^2\neq\sigma_2^2$ 时, $\mu_1-\mu_2$ 的置信度为 $1-\alpha$ 的单侧置信区间:

$$\left[\overline{X}-\overline{Y}-t_\alpha(\widehat{\nu})\sqrt{\frac{S_1^2}{n_1}+\frac{S_2^2}{n_2}},\ \ +\infty\right),\quad \left(-\infty,\ \overline{Y}-\overline{X}+t_\alpha(\widehat{\nu})\sqrt{\frac{S_1^2}{n_1}+\frac{S_2^2}{n_2}}\right], \tag{4.58}$$

其中 $\widehat{\nu}$ 由式 (4.36) 得到.

根据公式 (4.55)– 公式 (4.58), 以及双侧置信区间的公式写出下面的 R 程序 (程序名: `interval_estimate5.R`), 并可控制求上、下置信区间或双侧置信区间.

```
interval_estimate5<-function(x, y,
   sigma=c(-1,-1), var.equal=FALSE, side=0, alpha=0.05){
   n1<-length(x); n2<-length(y)
   xb<-mean(x); yb<-mean(y); zb<-xb-yb
   if (all(sigma>=0)){
      if (side<0){
         tmp<-qnorm(1-alpha)*sqrt(sigma[1]^2/n1+sigma[2]^2/n2)
         a <- -Inf; b <- zb+tmp
      }
      else if (side>0){
         tmp<-qnorm(1-alpha)*sqrt(sigma[1]^2/n1+sigma[2]^2/n2)
         a <- zb-tmp; b <- Inf
      }
      else{
         tmp<-qnorm(1-alpha/2)*sqrt(sigma[1]^2/n1+sigma[2]^2/n2)
         a <- zb-tmp; b <- zb+tmp
      }
      df<-n1+n2
   }
```

```
   else{
      if (var.equal ==  TRUE){
         Sw<-((n1-1)*var(x)+(n2-1)*var(y))/(n1+n2-2)
         if (side<0){
            tmp<-sqrt(Sw*(1/n1+1/n2))*qt(1-alpha,n1+n2-2)
            a <- -Inf; b <- zb+tmp
         }
         else if (side>0){
            tmp<-sqrt(Sw*(1/n1+1/n2))*qt(1-alpha,n1+n2-2)
            a <- zb-tmp; b <- Inf
         }
         else{
            tmp<-sqrt(Sw*(1/n1+1/n2))*qt(1-alpha/2,n1+n2-2)
            a <- zb-tmp; b <- zb+tmp
         }
         df<-n1+n2-2
      }
      else{
         S1<-var(x); S2<-var(y)
         nu<-(S1/n1+S2/n2)^2/(S1^2/n1^2/(n1-1)+S2^2/n2^2/(n2-1))
         if (side<0){
            tmp<-qt(1-alpha, nu)*sqrt(S1/n1+S2/n2)
            a <- -Inf; b <- zb+tmp
         }
         else if (side>0){
            tmp<-qt(1-alpha, nu)*sqrt(S1/n1+S2/n2)
            a <- zb-tmp; b <- Inf
         }
         else{
            tmp<-qt(1-alpha/2, nu)*sqrt(S1/n1+S2/n2)
            a <- zb-tmp; b <- zb+tmp
         }
```

```
        df<-nu
      }
   }
   data.frame(mean=zb, df=df, a=a, b=b)
}
```

在程序中，x，y 分别是来自两总体的数据 (样本) 构成的向量. sigma 是由两总体标准差构成的向量，当标准差已知时，输入相应的值，程序采用正态分布计算区间的端点. 当方差未知时，输入项缺省，此时需要考虑两总体是否相同：若认为两总体方差相同，输入 var.equal=TRUE, 程序采用自由度为 n_1+n_2-2 的 $t-$ 分布计算区间端点；若认为两总体方差不同，输入 var.equal=FALSE 或缺省，程序采用自由度为 ν 的 $t-$ 分布计算区间端点. 当 ν 不是整数时，程序在计算 $t-$ 分布时，其值采用插值方法得到. side 是控制求置信区间上下限，若求置信区间上限，输入 side=-1; 若求置信区间下限，输入 side=1; 若求双侧置信区间，输入 side=0 或缺省. 输出采用数据框形式，输出样本均值差 mean, 自由度 df, 和均值差的区间估计的两个端点 a,b.

上述程序实际上包含了求双侧置信区间的情况，也就是说，函数 interval_estimate5 包含了函数 interval_estimate2 的功能.

4. 求两个总体方差的情况

假设与前面相同， $\widehat{\sigma}_1^2$ 和 $\widehat{\sigma}_2^2$ 是由式 (4.38) 定义的，分别讨论两总体均值 μ_1 与 μ_2 已知和 μ_1 与 μ_2 未知情况下，方差比 σ_1^2/σ_2^2 的单侧区间估计.

当 μ_1 与 μ_2 已知时，由式 (4.39), 有

$$P\left\{\frac{\widehat{\sigma}_1^2/\sigma_1^2}{\widehat{\sigma}_2^2/\sigma_2^2}\le F_\alpha(n_1,n_2)\right\}=1-\alpha,\quad P\left\{F_{1-\alpha}(n_1,n_2)\le\frac{\widehat{\sigma}_1^2/\sigma_1^2}{\widehat{\sigma}_2^2/\sigma_2^2}\right\}=1-\alpha,$$

因此， σ_1^2/σ_2^2 的置信水平 $1-\alpha$ 的单侧置信区间为

$$\left[\frac{\widehat{\sigma}_1^2/\widehat{\sigma}_2^2}{F_\alpha(n_1,n_2)},\ +\infty\right),\qquad\left[0,\ \frac{\widehat{\sigma}_1^2/\widehat{\sigma}_2^2}{F_{1-\alpha}(n_1,n_2)}\right]. \tag{4.59}$$

当 μ_1 与 μ_2 未知时，由式 (4.42) 和 (4.43), 得到

$$\begin{aligned}P\left\{\frac{S_1^2/\sigma_1^2}{S_2^2/\sigma_2^2}\le F_\alpha(n_1-1,n_2-1)\right\}&=1-\alpha,\\ P\left\{F_{1-\alpha}(n_1-1,n_2-1)\le\frac{S_1^2/\sigma_1^2}{S_2^2/\sigma_2^2}\right\}&=1-\alpha,\end{aligned}$$

则 σ_1^2/σ_2^2 的置信水平 $1-\alpha$ 的单侧置信区间为

$$\left[\frac{S_1^2/S_2^2}{F_\alpha(n_1-1,n_2-1)},\ +\infty\right),\qquad \left[0,\ \frac{S_1^2/S_2^2}{F_{1-\alpha}(n_1-1,n_2-2)}\right]. \tag{4.60}$$

根据公式 (4.59) 和公式 (4.60), 以及双侧置信区间的公式写出下面的 R 程序 (程序名: interval_var4.R), 并可控制求上置信区间、双侧置信区间.

```
interval_var4<-function(x,y,
   mu=c(Inf, Inf), side=0, alpha=0.05){
   n1<-length(x); n2<-length(y)
   if (all(mu<Inf)) {
      Sx2<-1/n1*sum((x-mu[1])^2); df1<-n1
      Sy2<-1/n2*sum((y-mu[2])^2); df2<-n2
   }
   else{
      Sx2<-var(x); Sy2<-var(y); df1<-n1-1; df2<-n2-1
   }
   r<-Sx2/Sy2
   if (side<0) {
      a <- 0
      b <- r/qf(alpha,df1,df2)
   }
   else if (side>0) {
      a <- r/qf(1-alpha,df1,df2)
      b <- Inf
   }
   else{
      a<-r/qf(1-alpha/2,df1,df2)
      b<-r/qf(alpha/2,df1,df2)
   }
   data.frame(rate=r, df1=df1, df2=df2, a=a, b=b)
}
```

在程序中, x, y 分别是来自两总体的数据 (样本) 构成的向量. mu 是

由两总体均值构成的向量，当均值已知时，输入相应的值，程序采用自由度为 (n_1, n_2) 的 F– 分布计算区间估计的两个端点；否则 (输入缺省), 程序采用自由度为 (n_1-1, n_2-1) 的 F– 分布计算区间估计的两个端点. side 是控制求置信区间上下限，若求置信区间上限，输入 side=-1; 若求置信区间下限，输入 side=1; 若求双侧置信区间，输入 side=0 或缺省. alpha 是显著性水平，缺省值为 0.05. 输出采用数据框形式，输出的变量有：样本方差比 rate, 第一自由度 df1, 第二自由度 df2, 和方差比的区间估计的端点 a，b.

习题四

4.1 设总体的分布密度为

$$f(x;\alpha)=\begin{cases}(\alpha+1)x^{\alpha}, & 0<x<1,\\ 0, & \text{其他,}\end{cases}$$

$X_1, X_2, \cdots, X_n$ 为其样本，求参数 α 的矩估计量 $\widehat{\alpha}_1$ 和极大似然估计量 $\widehat{\alpha}_2$. 现测得样本观测值为

0.1　0.2　0.9　0.8　0.7　0.7

求参数 α 的估计值.

4.2 设元件无故障工作时间 X 具有指数分布，取 *1000* 个元件工作时间的记录数据，经分组后得到它的频数分布为

组中值 x_i	*5*	*15*	*25*	*35*	*45*	*55*	*65*
频数 v_i	*365*	*245*	*150*	*100*	*70*	*45*	*25*

如果各组中数据都取为组中值，试用极大似然估计求 λ 的点估计.

4.3 为检验某自来水消毒设备的效果，现从消毒后的水中随机抽取 *50* 升，化验每升水中大肠杆菌的个数 *(*假设一升水中大肠杆菌个数服从 *Poisson* 分布*)*, 其化验结果如下：

大肠杆菌数 / 升	*0*	*1*	*2*	*3*	*4*	*5*	*6*
升数	*17*	*20*	*10*	*2*	*1*	*0*	*0*

试问平均每升水中大肠杆菌个数为多少时，才能使上述情况的概率为最大？

4.4 利用 R 软件中的 `nlm()` 函数求解无约束优化问题

$$\begin{aligned}\min f(x) &= (-13+x_1+((5-x_2)x_2-2)x_2)^2 \\ &+ (-29+x_1+((x_2+1)x_2-14)x_2)^2,\end{aligned}$$

取初始点 $x^{(0)}=(0.5,-2)^T$.

4.5 正常人的脉搏平均每分钟 *72* 次，某医生测得 *10* 例四乙基铅中毒患各的脉搏数 *(*次 / 分*)* 如下：

54 67 68 78 70 66 67 70 65 69

已知人的脉搏次数服从正态分布，试计算这 *10* 名患者平均脉搏次数的点估计和 *95%* 的区间估计. 并作单侧区间估计，试分析这 *10* 名患者的平均脉搏次数是否低于正常人的平均脉搏次数.

4.6 甲、乙两种稻种分别播种在 *10* 块试验田中，每块试验田甲、乙稻种各种一半. 假设两稻种产量 X, Y 均服从正态分布，且方差相等. 收获后 *10* 块试验田的产量如下 所示 *(*单位：千克*)*.

甲种	*140*	*137*	*136*	*140*	*145*	*148*	*140*	*135*	*144*	*141*
乙种	*135*	*118*	*115*	*140*	*128*	*131*	*130*	*115*	*131*	*125*

求出两稻种产量的期望差 $\mu_1-\mu_2$ 的置信区间 $(\alpha=0.05)$.

4.7 甲、乙两组生产同种导线，现从甲组生产的导线中随机抽取 *4* 根，从乙组生产的导线中随机抽取 *5* 根，它们的电阻值 *(*单位： Ω*)* 分别为

甲组	*0.143*	*0.142*	*0.143*	*0.137*	
乙组	*0.140*	*0.142*	*0.136*	*0.138*	*0.140*

假设两组电阻值分别服从正态分布 $N(\mu_1,\sigma^2)$ 和 $N(\mu_2,\sigma^2)$, σ^2 未知. 试求 $\mu_1-\mu_2$ 的置信系数为 *0.95* 的区间估计.

4.8 对习题 *4.6* 中甲乙两种稻种的数据作方差比的区间估计，并用其估计值来判定两数据是否等方差. 若两数据方差不相等，试重新计算两稻种产量的期望差 $\mu_1-\mu_2$ 的置信区间 $(\alpha=0.05)$.

4.9 设电话总机在某段时间内接到的呼唤的次数服从参数未知的 *Poisson* 分布 $P(\lambda)$, 现收集了 *42* 个数据

接到呼唤次数	0	1	2	3	4	5	6
出现的频数	7	10	12	8	3	2	0

试求出平均呼唤次数 λ 的估计值和它的置信系数为 0.95 的置信区间.

4.10 已知某种灯泡寿命服从正态分布, 在某星期所生产的该灯泡中随机抽取 10 只, 测得其寿命 (单位：小时) 为

1067 919 1196 785 1126 936 918 1156 920 948

求灯泡寿命平均值的置信度为 0.95 的单侧置信下限.

第五章　假设检验

假设检验 (test of hypothesis) 是统计推断中的一个重要内容，它是利用搜索到的数据对某个事先作出的统计假设按照某种设计好的方法进行检验，判断此假设是否正确.

5.1　假设检验的基本概念

5.1.1　基本概念

在数理统计分析中，只能由估计量估计总体的参数. 尽管能获得总体参数的无偏估计，总体的参数始终是不可知的. 只能通过统计检验，由统计量推断总体的参数. 在统计推断过程中，需要对参数提出一定的假设，然后对提出的假设进行假设检验. 用一个例子说明假设检验的基本概念.

例 5.1 设某工厂生产的一批产品，其次品率 p 是未知的. 按规定，若 $p \le 0.01$, 则这批产品为可接受的；否则为不可接受的. 这里 "$p \le 0.01$" 便是一个需要的假设，记为 H. 假定从这批数据很大的产品中随机地抽取 *100* 件样品，发现其中有三件次品，这一抽样结果便成为判断假设 H 是否成立的依据. 显然，样品中次品个数愈多对假设 H 愈不利；反之则对 H 有利. 记样品中次品个数为 X, 问题是： X 大到什么程序就应该拒绝 H?

分析：由于否定了 H 就等于否定了一大批产品，这个问题应该慎重处理. 统计学上常用的作法是：先假定 H 成立，来计算 $X \ge 3$ 的概率有多大？由于 X 分布为 $B(n,p)$, 其中 $n = 100$, 容易计算出 $P_{p=0.01}\{X \ge 3\} \approx 0.08$. 显然，对 $p < 0.01$, 这概率值还要小，也就是说，当假设 $H(p \le 0.01)$ 成立时， 100 个样品中有 3 个或 3 个以上次品的概率不超过 0.08. 这可以看作是一个 "小概率" 事件. 而在一次试验中就发生了一个小概率事件是不大可能的. 因此，事先作出的假设 "$p \le 0.01$" 是非常可疑的. 在需要作出最终判决时，就应该否定这个假设，而认定这批产品不可接受 (即认为 $p > 0.01$).

上述例子中包含了假设检验的一些重要的基本概念. 一般，设 θ 为用以确定总体分布的一个未知参数，其一切可能值的集合记为 Θ. 则关于 θ 的任一假设可用 "$\theta \in \Theta'$" 来表示，其中 Θ' 为 Θ 的一个真子集. 在统计假设检验中，首先要有一个作为检验的对象的假设，常称不原假设或零假设 (null hypothesis). 与

之相应，为使问题表述得更明确，还常提出一个与之对应的假设，称为备择假设 (alternative hypothesis). 原假设和备择假设常表示为

$$H_0: \quad \theta \in \Theta_0, \qquad H_1: \quad \theta \in \Theta_1,$$

其中 Θ_0 和 Θ_1 为 Θ 的两个不相交的真子集， H_0 表示原假设， H_1 表示备择假设.

关于一维实参数的假设常有以下三种形式 (其中 θ_0 为给定值):

(1) 单边检验

$$H_0: \quad \theta \leq \theta_0, \qquad H_1: \quad \theta > \theta_0.$$

(2) 单边检验

$$H_0: \quad \theta \geq \theta_0, \qquad H_1: \quad \theta < \theta_0.$$

(3) 双边检验

$$H_0: \quad \theta = \theta_0, \qquad H_1: \quad \theta \neq \theta_0.$$

通常也称双边检验为二尾检验，称单边检验为一尾检验.

假设检验的依据是样本. 样本的某些取值可能对原假设 H_0 有利，而另一些取值可能对 H_0 不利，因此可以根据某种公认的合理准则将样本空间分成两部分. 一部分你为拒绝域 (critical region), 当样本落入拒绝域时，便拒绝 H_0; 另一部分可称为接受域 (acceptance region), 当样本落入它时不拒绝 H_0.

构造拒绝域的常用方法是寻找一个统计量 g(如例 5.1 中的样品中次品的件数 X), g 的大小可以反映对原假设 H_0 有利或不利. 因此，确定拒绝域 W 的问题转化为确定 g 的一个取值域 C 的问题.

定义 5.1 对假设检验问题，设 $X_1, X_2, \cdots, X_n$ 为样本， W 为样本空间中的一个子集，对于给定的 $\alpha \in (0,1)$, 若 W 满足

$$P_\theta\{(X_1, X_2, \cdots, X_n) \in W\} \leq \alpha, \qquad \forall\, \theta \in \Theta_0, \tag{5.1}$$

则称由 W 构成拒绝域的检验方法为显著性水平 (evidence level) α 的检验.

显著性水平 α 常用的取值为 0.1, 0.05 和 0.01 等. 对一个显著性水平 α 的检验，假定原假设 H_0 成立，而样本落入拒绝域 W 中，就意味着一个小概率发生了. 而在一次试验中发生一个小概率事件是可疑的，结果就导致了对原假设 H_0

的否定. 在例 5.1 中, 如果事先给定 $\alpha = 0.1$, 而 $P_{p=0.01}\{X \geq 3\} = 0.08$, 因此当 $p < 0.01$ 时, 这个概率还要小. 根据定义 5.1, $W = \{X \geq 3\}$ 便给出了假设检验 $H_0 : p \leq p_0 = 0.01$ 的显著性水平 $\alpha = 0.1$ 的拒绝域, 由 $X = 3$ 便可拒绝 H_0. 但如果事先给定的显著性水平 $\alpha = 0.05$, 这时, 相应的显著性水平 α 的检验的拒绝域 $W = \{X \geq 4\}$, 这时 $X = 3$ 就不能拒绝 H_0. 由此可见, 显著性水平 α 愈小, 则拒绝原假设愈困难. 换言之, 显著性水平 α 愈小, 则当样本落入拒绝域因而拒绝 H_0 就愈加可信.

通常, 作假设者对原假设 H_0 往往事先有一定的信任度, 或者一旦否定了 H_0 就意味着作出一个重大的决策, 需谨慎从事, 因此把检验的显著性水平 α 取得比较小其中体现了一种 "保护原假设" 的思想.

5.1.2 假设检验的基本思想与步骤

假设检验的基本思想:

(1) 用了反证法的思想. 为了检验一个 "假设" 是否成立, 就先假定这个 "假设" 是成立的, 而看由此会产生的后果. 如果导致一个不合理的现象的出现, 那么就表明原先的假定不正确, 也就是说, "假设" 不成立. 因此, 我们就拒绝这个 "假设". 如果由此没有导出不合理的现象发生, 则不能拒绝原来这个 "假设", 称原假设是相容的.

(2) 它又区别于纯数学中的反证法. 因为这里所谓的 "不合理", 并不是形式逻辑中的绝对矛盾, 而是基于人们实践中广泛采用的一个原则: 小概率事件在一次观察中可以认为基本上不会发生.

假设检验的一般步骤为:

(1) 对待检验的未知参数 θ 根据问题的需要作出一个单边或双边的假设. 选择原假设的原则是: 事先有一定信任度或出于某种考虑是否要加以 "保护".

(2) 选定一个显著性水平 α, 最常用的是 $\alpha = 0.05$, 放松一点可取 $\alpha = 0.075$ 或 0.1, 严格一些可取 $\alpha = 0.025$ 或 0.01.

(3) 构造一个统计量 g, g 的大小反映对 H_0 有利或不利, 拒绝域有形式 $W = \{g \in C\}$.

(4) 根据定义 5.1 来确定 W.

5.1.3 假设检验的两类错误

在根据假设检验作出统计决断时，可能犯两类错误. 第一类错误是否定了真实的原假设. 犯一型错误的概率定义为显著性水平 α, 即

$$\alpha = P\{\text{否定}H_0 \mid H_0\text{是真实的}\},$$

可以通过控制显著性水平 α 来控制犯第一类错误的概率.

第二类错误是接受了错误的原假设. 犯第二类错误的概率常用 β 表示，即

$$\beta = P\{\text{接受}H_0 \mid H_0\text{是错误的}\}.$$

通常来讲，在给定样本容量的情况下，如果减少犯第一类错误的概率，就会增加犯第二类错误的概率. 而减少犯第二类错误的概率，也会增加犯第一类错误的概率. 如果希望同时减少犯第一类和第二类错误的概率，就需要增加样本容量，但样本容量的增加，是需要增加抽样成本，这有时是不可行的.

在统计检验中，评价一个假设检验好坏的标准是统计检验功效，所谓功效就是正确地否定了错误的原假设的概率，常用 π 表示，即

$$\pi = 1 - \beta = P\{\text{否定}H_0 \mid H_0\text{是错误的}\}.$$

如果统计检验接受了原假设 $H_0 : \theta = \theta_0$, 则可以通过计算置信区间，推断总体参数 θ 的取值范围. 置信区间是根据一定置信程度而估计的区间，它给出了未知的总体参数的上下限.

5.2 重要的参数检验

由于实际问题中大多数随机变量服从或近似服从正态分布，因此，这里重点介绍正态参数的假设检验. 按总体的个数，又可分为单个正态总体和两个正态总体的参数检验.

5.2.1 正态总体均值的假设检验

1. 单个总体的情况

设总体 $X \sim N(\mu, \sigma^2)$, $X_1, X_2, \cdots, X_n$ 是来自总体 X 的一个样本，均值 μ 的检验分为：双边检验和单边检验. 在讨论中，又分为总体方差 σ^2 已知和总体方差 σ^2 未知两种情况.

(1) 双边检验，即

$$H_0: \ \mu = \mu_0, \qquad H_1: \ \mu \neq \mu_0.$$

当方差 σ^2 已知时，由第一章 1.5.4 节的统计知识 (式 (1.93)) 可知，当 H_0 为真时，

$$Z = \frac{\overline{X} - \mu_0}{\sigma/\sqrt{n}} \sim N(0,1), \tag{5.2}$$

因此用 Z 来确定拒绝域，即当

$$|Z| \geq Z_{\alpha/2},$$

则认为 H_0 不成立，其中 α 为显著性水平. 这种方法称为正态检验法.

当方差 σ^2 未知时，由统计知识 (1.5.4 节的式 (1.95)) 可知，当 H_0 为真时，

$$T = \frac{\overline{X} - \mu_0}{S/\sqrt{n}} \ \sim \ t(n-1), \tag{5.3}$$

因此当

$$|T| \geq t_{\alpha/2}(n-1),$$

则认为 H_0 不成立. 这种方法称为 $t-$ 检验法.

在实际问题中，正态总体的方差通常是未知的，所以常用 t 检验法来检验关于正态总体均值的检验问题.

(2) 单检验，即

$$H_0: \ \mu \leq \mu_0, \quad H_1: \ \mu > \mu_0 \ \ (\text{或 } H_0: \ \mu \geq \mu_0, \quad H_1: \ \mu < \mu_0),$$

当方差 σ^2 已知时，其拒绝域为

$$Z \geq Z_\alpha \ \ (\text{或 } Z \leq -Z_\alpha).$$

当方差 σ^2 未知时，其拒绝域为

$$T \geq t_\alpha(n-1) \ \ (\text{或 } T \leq -t_\alpha(n-1)).$$

在传统的方法中，通常采用查表的方法来确定临界值，而在计算机软件的计算中，通常是计算 $P-$ 值，当 $P-$ 值小于指定的显著性水平 α, 则拒绝原假设.

所谓 $P-$ 值就是随机变量 X 大于 (或小于) 某个指定值的概率.

对于单边检验比较简单，以正态分布为例，在给定 z 值后，只需考虑 $X \geq z$ 的概率，即

$$\begin{aligned} P-\text{值} &= P\{X \geq z\} = \int_z^{\infty} \phi(x)dx = 1-\Phi(z) \\ &= 1-\texttt{pnorm(z, 0, 1)}, \end{aligned} \tag{5.4}$$

或者考虑 $X \leq z$ 的概率，即

$$P-\text{值} = P\{X \leq z\} = \int_{-\infty}^{z} \phi(x)dx = \texttt{pnorm(z, 0, 1)}. \tag{5.5}$$

对于双边检验，还是以正态分布为例，在给定 z 值后，需要考虑 $X \geq |z|$ 和 $X \leq -|z|$ 的概率，或者考虑 $X \geq |z|$ 概率的两倍. 因此， $P-$ 值的计算公式为

$$\begin{aligned} P-\text{值} &= \begin{cases} 2P\{X \leq z\}, & \text{如果 } P\{X \leq z\} < P\{X \geq z\} \\ 2P\{X \geq z\}, & \text{否则} \end{cases} \\ &= \begin{cases} 2\displaystyle\int_{-\infty}^{z} \phi(x)dx, & \text{如果 } \displaystyle\int_{-\infty}^{z} \phi(x)dx < \int_{z}^{\infty} \phi(x)dx \\ 2\displaystyle\int_{z}^{\infty} \phi(x)dx, & \text{否则} \end{cases} \\ &= \begin{cases} 2\Phi(z), & \text{如果 } \Phi(z) < (1-\Phi(z)) \\ 2(1-\Phi(z)) & \text{否则} \end{cases} \\ &= \begin{cases} 2\,\text{pnorm(z)}, & \text{如果 pnorm(z)} < \dfrac{1}{2} \\ 2(1-\text{pnorm(z)}) & \text{否则} \end{cases} \end{aligned} \tag{5.6}$$

将式 (5.4)–(5.6) 编写成求 P– 值的 R 程序 (程序名： `P_value.R`)

```
P_value<-function(cdf, x, paramet=numeric(0), side=0){
   n<-length(paramet)
   P<-switch(n+1,
      cdf(x),
      cdf(x, paramet),
```

```
        cdf(x, paramet[1], paramet[2]),
        cdf(x, paramet[1], paramet[2], paramet[3])
    )
    if (side<0)         P
    else if (side>0)  1-P
    else
        if (P<1/2)     2*P
        else           2*(1-P)
}
```

其中输入值 cdf 是分布函数, 如正态分布就是 pnorm. x 是计算 P– 值的给定值. paramet 是对应分布的参数, 如正态分布的参数为 paramet=c(mu, sigma). side 是计算单侧 P– 值或双侧 P– 值的指标参数, 输入 side=-1, 计算左侧的 P– 值; 输入 side=1, 计算右侧的 P– 值; 输入 side=0 或缺省, 计算双侧 P– 值. 函数的输出值是相应的 P– 值.

在得到 $P-$ 值后, 其检验标准改为: 当 P- 值小于指定的显著性水平 α 时, 则拒绝原假设; 否则不拒绝原假设.

将上面进述介绍的正态检验方法 (式 (5.2)) 和 t 检验方法 (式 5.3) 与求 P– 值的 R 程序相结合, 编写求一个正态总体均值检验的 R 程序 (程序名: mean.test1.R)

```
mean.test1<-function(x, mu=0, sigma=-1, side=0){
    source("P_value.R")
    n<-length(x); xb<-mean(x)
    if (sigma>0){
        z<-(xb-mu)/(sigma/sqrt(n))
        P<-P_value(pnorm, z, side=side)
        data.frame(mean=xb, df=n, Z=z, P_value=P)
    }
    else{
        t<-(xb-mu)/(sd(x)/sqrt(n))
        P<-P_value(pt, t, paramet=n-1, side=side)
        data.frame(mean=xb, df=n-1, T=t, P_value=P)
```

```
    }
}
```

在上述程序中, 输入值 x 是数据 (样本) 构成的向量. mu 是原假设 μ_0. sigma 是标准差 σ, 当 σ 已知时, 输入相应的值, 程序采用正态检验法; 当 σ 未知时 (缺省), 程序采用 $t-$ 检验法. side 是指双边检验还是单边检验. 输入 side = 0 (或缺省), 程序作双边检验, 其备择假设为: $\mu \neq \mu_0$; 输入 side = -1 (或 < 0 的值), 程序作单边检验, 其备择假设为: $\mu < \mu_0$; 输入 side = 1 (或 > 0 的值), 程序作单边检验, 其备择假设为: $\mu > \mu_0$.

程序以数据框形式输出, 输出的内容有: 均值 (mean), 自由度 (df), 统计量 (T 值或 z 值), 和 P- 值.

例 5.2 某种元件的寿命 X (以小时计) 服从正态分布 $N(\mu, \sigma^2)$, 其中 μ, σ^2 均未知. 现测得 16 只元件的寿命如下:

159 280 101 212 224 379 179 264
222 362 168 250 149 260 485 170

问是否有理由认为元件的平均寿命大于 225 (小时)?

解: 按题意 (注意前面提到的假设检验运用了反证法的思想), 需检验

$$H_0: \ \mu \leq \mu_0 = 225, \qquad H_1: \ \mu > \mu_0 = 225.$$

此问题是单边检验问题.

输入数据, 调用函数 mean.test1(), 得到

```
> X<-c(159, 280, 101, 212, 224, 379, 179, 264,
       222, 362, 168, 250, 149, 260, 485, 170)
> source("mean.test1.R")
> mean.test1(X, mu=225, side=1)
   mean df         T   P_value
1 241.5 15 0.6685177 0.2569801
```

计算出 $P-$ 值是 $0.2569801 (> 0.05)$, 不能拒绝原假设, 接受 H_0, 即认为平均寿命不大于 225 小时.

实际上, 参数的区间估计也作假设检验, 换句话说, 区间估计与假设检验本质上是相同的. 对例 5.2 中的数据作单侧区间估计 (估计下限),

```
> source("interval_estimate4.R")
> interval_estimate4(X, side=1)
   mean df        a   b
1 241.5 15 198.2321 Inf
```

置信下限为 $198.23 < 225$, 因此只能接受原假设, 认为平均寿命不大于 225 小时.

在 R 软件中, 函数 t.test() 提供了 T 检验和相应的区间估计的功能, t.test() 的使用格式如下:

```
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95, ...)
```

其中 x,y 是由数据构成向量 (如果只提供 x, 则作单个正态总体的均值检验; 否则作两个总体的均值检验), alternative 表示备择假设, two.sided(缺省) 表示双边检验 ($H_1 : \mu \neq \mu_0$), less 表示单边检验 ($H_1 : \mu < \mu_0$), greater 表示单边检验 ($H_1 : \mu > \mu_0$). mu 表示原假设 μ_0. conf.level 是置信水平, 即 $1-\alpha$, 通常是 0.95.

再用 t.test() 函数计算例 5.2.

```
> t.test(X, alternative = "greater", mu = 225)
        One Sample t-test
data:  X
t = 0.6685, df = 15, p-value = 0.257
alternative hypothesis: true mean is greater than 225
95 percent confidence interval:
 198.2321      Inf
sample estimates:
mean of x
    241.5
```

可以看到, 所计算的 T 值、 $P-$ 值、和均值, 以及区间估计值与我们所编程序的计算值完全相同, 因此, 可以利用函数 t.test() 对单个总体正态数据作均值检验和区间估计. 由这个例子和自编的程序的计算结果, 可以使我们加深对 R 软件中的 t.test() 函数的认识. 当然, t.test() 函数还有更强大的功能,

这些功能我们将在后面予以介绍.

2. 两个总体的情况

假设 $X_1, X_2, \cdots, X_{n_1}$ 是来自总体 $X \sim N(\mu_1, \sigma_1^2)$ 的样本, $Y_1, Y_2, \cdots, Y_{n_2}$ 是来自总体 $Y \sim N(\mu_2, \sigma_2^2)$ 的样本, 且两样本独立. 其检验问题有

$$\begin{aligned}
&\text{双边检验:} && H_0: \ \mu_1 = \mu_2, && H_1: \ \mu_1 \neq \mu_2,\\
&\text{单边检验 I:} && H_0: \ \mu_1 \leq \mu_2, && H_1: \ \mu_1 > \mu_2,\\
&\text{单边检验 II:} && H_0: \ \mu_1 \geq \mu_2, && H_1: \ \mu_1 < \mu_2.
\end{aligned}$$

分几种情况讨论.

(1) 方差 σ_1^2 和 σ_2^2 已知. 由统计知识 (1.5.4 节的式 (1.97)) 可知, 当 H_0 为真时,

$$Z = \frac{\overline{X} - \overline{Y}}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \sim N(0,1). \tag{5.7}$$

因此, 当 Z 满足 (称为拒绝域)

$$\begin{aligned}
&\text{双边检验:} && |Z| \geq Z_{\alpha/2},\\
&\text{单边检验 I:} && Z \geq Z_{\alpha},\\
&\text{单边检验 II:} && Z \leq -Z_{\alpha}.
\end{aligned}$$

则认为 H_0 不成立. 此方法仍称为正态检验法.

(2) 方差 $\sigma_1^2 = \sigma_2^2 = \sigma^2$ 未知. S_1^2 和 S_2^2 分别是 X 和 Y 的样本方差. 由统计知识 (1.5.4 节的式 (1.98)) 可知, 当 H_0 为真时,

$$T = \frac{\overline{X} - \overline{Y}}{S_w\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \sim t(n_1 + n_2 - 2), \tag{5.8}$$

其中

$$S_w = \sqrt{\frac{(n_1-1)S_1^2 + (n_2-1)S_2^2}{n_1 + n_2 - 2}}. \tag{5.9}$$

因此, 当 T 满足 (称为拒绝域)

$$\begin{aligned}
&\text{双边检验:} && |T| \geq t_{\alpha/2}(n_1 + n_2 - 2),\\
&\text{单边检验 I:} && T \geq t_{\alpha}(n_1 + n_2 - 2),\\
&\text{单边检验 II:} && T \leq -t_{\alpha}(n_1 + n_2 - 2).
\end{aligned}$$

则认为 H_0 不成立. 此方法仍称为 $t-$ 检验法.

(3) 方差 $\sigma_1^2 \neq \sigma_2^2$ 未知. S_1^2 和 S_2^2 分别是 X 和 Y 的样本方差. 可以证明

$$T = \frac{\overline{X} - \overline{Y}}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}} \sim t(\widehat{\nu}) \tag{5.10}$$

近似成立, 其中

$$\widehat{\nu} = \left(\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}\right)^2 \Big/ \left(\frac{(S_1^2)^2}{n_1^2(n_1-1)} + \frac{(S_2^2)^2}{n_2^2(n_2-1)}\right). \tag{5.11}$$

因此, 当 T 满足 (称为拒绝域)

$$\begin{aligned} &\text{双边检验:} && |T| \geq t_{\alpha/2}(\widehat{\nu}), \\ &\text{单边检验 I:} && T \geq t_{\alpha}(\widehat{\nu}), \\ &\text{单边检验 II:} && T \leq -t_{\alpha}(\widehat{\nu}). \end{aligned}$$

则认为 H_0 不成立.

根据公式 (5.7) 、公式 (5.8) 和公式 (5.10) 写出三种情况下两总体均值检验的 R 程序 (程序名: `mean.test2.R`).

```
mean.test2<-function(x, y,
   sigma=c(-1, -1), var.equal=FALSE, side=0){
   source("P_value.R")
   n1<-length(x); n2<-length(y)
   xb<-mean(x); yb<-mean(y)
   if (all(sigma>0)){
      z<-(xb-yb)/sqrt(sigma[1]^2/n1+sigma[2]^2/n2)
      P<-P_value(pnorm, z, side=side)
      data.frame(mean=xb-yb, df=n1+n2, Z=z, P_value=P)
   }
   else{
      if (var.equal ==  TRUE){
         Sw<-sqrt(((n1-1)*var(x)+(n2-1)*var(y))/(n1+n2-2))
         t<-(xb-yb)/(Sw*sqrt(1/n1+1/n2))
```

```
        nu<-n1+n2-2
      }
      else{
        S1<-var(x); S2<-var(y)
        nu<-(S1/n1+S2/n2)^2/(S1^2/n1^2/(n1-1)+S2^2/n2^2/(n2-1))
        t<-(xb-yb)/sqrt(S1/n1+S2/n2)
      }
      P<-P_value(pt, t, paramet=nu, side=side)
      data.frame(mean=xb-yb, df=nu, T=t, P_value=P)
   }
}
```

在上述程序中，输入值 x,y 是来自两个总体数据构成的向量. sigma 是由两总体标准差构成的向量，当标准差已知时，输入相应的值，程序采用正态检验法；当标准差未知时 (缺省)，程序采用 $t-$ 检验法. var.equal 是逻辑变量，输入 var.equal=TRUE，表示认为两总体的方差相同；输入 var.equal=FALSE (或缺省)，表示认为两总体的方差不同. side 是指双边检验还是单边检验. 输入 side = 0 (或缺省)，程序作双边检验，其备择假设为： $\mu_1 \neq \mu_2$；输入 side = -1 (或 < 0 的值)，程序作单边检验，其备择假设为： $\mu_1 < \mu_2$；输入 side = 1 (或 > 0 的值)，程序作单边检验，其备择假设为： $\mu_1 > \mu_2$.

程序以数据框形式输出，输出的内容有：均值的差 (mean)，自由度 (df)，统计量 (T 值或 z 值)，和 P- 值.

例 5.3 在平炉上进行一项试验以确定改变操作方法的建议是否会增加钢的得率，试验是在同一个平炉上进行的. 每炼一炉钢时除操作方法外，其他条件都尽可能做到相同. 先用标准方法炼一炉，然后用新方法炼一炉，以后交替进行，各炼了 *10* 炉，其得率分别为

标准方法	78.1	72.4	76.2	74.3	77.4	78.4	76.0	75.5	76.7	77.3
新方法	79.1	81.0	77.3	79.1	80.0	79.1	79.1	77.3	80.2	82.1

设这两样本相互独立，且分别来自正态总体 $N(\mu_1, \sigma^2)$ 和 $N(\mu_2, \sigma^2)$，其中 μ_1, μ_2 和 σ^2 未知. 问新的操作能否提高得率？（取 $\alpha = 0.05$ ）

解: 根据题意，需要假设

$$H_0:\ \mu_1 \geq \mu_2, \quad H_1:\ \mu_1 < \mu_2,$$

这里假定 $\sigma_1^2 = \sigma_2^2 = \sigma^2$, 因此选择 $t-$ 检验法，方差相同的情况（程序名：exam_0503.R).

```
X<-c(78.1,72.4,76.2,74.3,77.4,78.4,76.0,75.5,76.7,77.3)
Y<-c(79.1,81.0,77.3,79.1,80.0,79.1,79.1,77.3,80.2,82.1)
source("mean.test2.R")
mean.test2(X, Y, var.equal=TRUE, side=-1)
```

得到

```
  mean df          T      P_value
1 -3.2 18 -4.295743 0.0002175927
```

计算出 P– 值是 $0.0002176 \ll 0.05$, 故拒绝原假设. 即认为新的操作方能够提高得率.

如果认为两总体方差不同，则

```
> hypothesis.test2(X, Y, side=-1)
  mean       df          T      P_value
1 -3.2 17.31943 -4.295743 0.0002354815
```

仍然是拒绝原假设.

实际上，利用区间估计也可以作假设检验，例如，利用两个总体均值差的区间估计作假设检验，

```
#### 调用两个总体均值差的区间估计函数
> source("interval_estimate5.R")
#### 作单侧区间估计，并认为两总体方差相同
> interval_estimate5(X, Y, var.equal=TRUE, side=-1)
  mean df    a         b
1 -3.2 18 -Inf -1.908255
#### 作单侧区间估计，并认为两总体方差不同
> interval_estimate5(X,Y, side=-1)
  mean       df    a         b
1 -3.2 17.31943 -Inf -1.905500
```

无论是认为两样本方差相同，还是认为两样本方差不同，其均值差的上限估计均 < 0, 也就是说 $\mu_1 - \mu_2 < 0$, 即 $\mu_1 < \mu_2$.

在 R 软件中，函数 t.test() 也可以作双样本检验，其使用格式为

```
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95, ...)
```

其中 x，y 是来自两总体数据构成的向量，alternative 是备择假设，two.sided (缺省) 表示双边检验 $(H_1 : \mu_1 \neq \mu_2)$, less 表示单边检验 $(H_1 : \mu_1 < \mu_2)$, greater 表示单边检验 $(H_1 : \mu_1 > \mu_2)$. var.equal 是逻辑变量， var.equal=TRUE 表示认为两样本方差相同； var.equal=FALSE(缺省) 表示认为两样本方差不同.

用 t.test() 函数对上例进行计算.

```
> t.test(X, Y, var.equal=TRUE, alternative = "less")
        Two Sample t-test
data:  X and Y
t = -4.2957, df = 18, p-value = 0.0002176
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
      -Inf -1.908255
sample estimates:
mean of x mean of y
    76.23     79.43
```

从计算结果可以看到， t.test() 不但可以作两个总体的均值检验，还可以作两个总体均值差的区间估计，其计算结果与我们编程的计算结果相同. 这一点可以很好地帮助我们理解 t.test() 函数的功能与计算过程.

结合单个总体的均值检验，发现 t.test() 函数，可以作单、双总体的均值检验，还提供了均值的区间估计. 完成均值检验与估计的全部工作.

事实上, 均值的区间估计与均值的假设检验本质上是对一个问题从两个不同角度的讨论，有着内在的联系，这也就是为什么 t.test() 将区间估计与假设检验放在一起的原因，可以使我们从多角度对问题进行判断，提高判断的准确性.

3. 成对数据的 $t-$ 检验

如果数据是成对出现的, 即 $(X_i, Y_i),\ (i = 1, 2, \cdots, n)$, 则认为用成对 $t-$ 检验要优于双样本均值检验. 所谓成对 $t-$ 检验就是令 $Z_i = X_i - Y_i,\ (i = 1, 2, \cdots, n)$, 对 Z 作单样本均值检验. 例如, 对于例 5.3 中的数据就应作成对 $t-$ 检验.

```
> X<-c(78.1,72.4,76.2,74.3,77.4,78.4,76.0,75.5,76.7,77.3)
> Y<-c(79.1,81.0,77.3,79.1,80.0,79.1,79.1,77.3,80.2,82.1)
> t.test(X-Y, alternative = "less")
        One Sample t-test
data:  X - Y
t = -4.2018, df = 9, p-value = 0.001150
alternative hypothesis: true mean is less than 0
95 percent confidence interval:
      -Inf -1.803943
sample estimates:
mean of x
     -3.2
```

同样说明, 新方法优于标准方法, 但它计算的 $P-$ 值更小, 说明判断更可靠.

5.2.2 正态总体方差的假设检验

1. 单个总体的情况

设 $X_1, X_2, \cdots, X_n$ 是来自总体 $X \sim N(\mu, \sigma^2)$ 的样本, 其检验问题为

$$\begin{aligned}
&\text{双边检验:} && H_0: \ \sigma^2 = \sigma_0^2, && H_1: \ \sigma^2 \neq \sigma_0^2,\\
&\text{单边检验 I:} && H_0: \ \sigma^2 \leq \sigma_0^2, && H_1: \ \sigma^2 > \sigma_0^2,\\
&\text{单边检验 II:} && H_0: \ \sigma^2 \geq \sigma_0^2, && H_1: \ \sigma^2 < \sigma_0^2.
\end{aligned}$$

分均值 μ 已知和均值 μ 未知两种情形讨论.

当均值 μ 是已知时, 当 H_0 为真时, 令 $\widehat{\sigma}^2 = \dfrac{1}{n}\sum_{i=1}^{n}(X_i - \mu)^2$, 则有

$$\chi^2 = \frac{n\widehat{\sigma}^2}{\sigma_0^2} \quad \sim \quad \chi^2(n), \tag{5.12}$$

因此用 χ^2 来确定拒绝域，即当

$$\begin{aligned}
&\text{双边检验:} \quad \chi^2 \geq \chi^2_{\alpha/2}(n) \text{ 或 } \chi^2 \leq \chi^2_{1-\alpha/2}(n),\\
&\text{单边检验 I:} \quad \chi^2 \geq \chi^2_{\alpha}(n),\\
&\text{单边检验 II:} \quad \chi^2 \leq \chi^2_{1-\alpha}(n).
\end{aligned}$$

则认为 H_0 不成立.

当均值 μ 是未知时，当 H_0 为真时，有

$$\chi^2 = \frac{(n-1)S^2}{\sigma_0^2} \sim \chi^2(n-1), \tag{5.13}$$

因此用 χ^2 来确定拒绝域，即当

$$\begin{aligned}
&\text{双边检验:} \quad \chi^2 \geq \chi^2_{\alpha/2}(n-1) \text{ 或 } \chi^2 \leq \chi^2_{1-\alpha/2}(n-1),\\
&\text{单边检验 I:} \quad \chi^2 \geq \chi^2_{\alpha}(n-1),\\
&\text{单边检验 II:} \quad \chi^2 \leq \chi^2_{1-\alpha}(n-1).
\end{aligned}$$

则认为 H_0 不成立.

与均值检验相同，在计算中仍用 $P-$ 值的大小来判断是否拒绝 H_0. 当 P- 值小于 α, 则拒绝 H_0; 否则不拒绝 H_0. 关于 $P-$ 值的计算方法与均值检验的方法相同.

根据公式 (5.12) 和公式 (5.13) 写出总体均值已知和均值未知两种情况方差检验的 R 程序 (程序名： `var.test1.R`), 在程序中调用 P– 值计算程序.

```
var.test1<-function(x, sigma2=1, mu=Inf, side=0){
   source("P_value.R")
   n<-length(x)
   if (mu<Inf){
      S2<-sum((x-mu)^2)/n; df=n
   }
   else{
      S2<-var(x); df=n-1
   }
   chi2<-df*S2/sigma2;
```

```
    P<-P_value(pchisq, chi2, paramet=df, side=side)
    data.frame(var=S2, df=df, chisq2=chi2, P_value=P)
}
```

在上述程序中, 输入值 x 是数据构成的向量. sigma2 是原假设 σ_0^2. mu 是均值, 当 μ 已知时, 输入相应的值, 程序采用自由度为 n 的 χ^2 检验; 否则 (缺省), 程序采用自由度为 $n-1$ 的 χ^2 检验. side 是指双边检验还是单边检验. 输入 side = 0 (或缺省), 程序作双边检验, 其备择假设为: $\sigma^2 \neq \sigma_0^2$; 输入 side = -1 (或 < 0 的值), 程序作单边检验, 其备择假设为: $\sigma^2 < \sigma_0^2$; 输入 side = 1 (或 > 0 的值), 程序作单边检验, 其备择假设为: $\sigma^2 > \sigma_0^2$.

程序以数据框形式输出, 输出的内容有: 方差 (var), 自由度 (df), 统计量 (chisq2), 和 P- 值.

例 5.4 从小学五年级男学生中抽取 *20* 名, 测量其身高 (单位: 厘米), 其数据如下:

136	144	143	157	137	159	135	158	147	165
158	142	159	150	156	152	140	149	148	155

以 $\alpha = 0.05$ 作假设检验:

$$(1) \qquad H_0: \ \mu = 149, \qquad H_1: \ \mu \neq 149;$$
$$(2) \qquad H_0: \ \sigma^2 = 75, \qquad H_1: \ \sigma^2 \neq 75.$$

解: 输入数据, 用上面编写的程序, 就方差已知和方差未知情况作均值检验, 就均值已知和均值未知的情况作方差检验.

```
#### 用 scan() 函数读数据
> X<-scan()
1: 136  144  143  157  137  159  135  158  147  165
11: 158  142  159  150  156  152  140  149  148  155
21:
Read 20 items
#### 调用均值检验函数 mean.test1
> source("mean.test1.R")
```

```
#### 认为方差已知, 作均值检验
> mean.test1(X, mu=149, sigma=sqrt(75))
   mean df         Z   P_value
1 149.5 20 0.2581989 0.7962534
#### 认为方差未知, 作均值检验
> mean.test1(X, mu=149)
   mean df         T   P_value
1 149.5 19 0.2536130 0.8025186
#### 调用均值检验函数 var.test1
> source("var.test1.R")
#### 认为均值已知, 作方差检验
> var.test1(X, sigma2=75, mu=149)
   var df chisq2   P_value
1 74.1 20  19.76 0.9460601
#### 认为均值未知, 作方差检验
> var.test1(X, sigma2=75)
       var df   chisq2   P_value
1 77.73684 19 19.69333 0.8264785
```

无论是哪种方法, 其 $P-$ 值均大于 0.79, 因此接受原假设.

2. 两个总体的情况

设 $X_1, X_2, \cdots, X_{n_1}$ 是来自总体 $X \sim N(\mu_1, \sigma_1^2)$ 的样本, $Y_1, Y_2, \cdots, Y_{n_2}$ 是来自总体 $Y \sim N(\mu_2, \sigma_2^2)$ 的样本, 且两样本独立. 其检验问题为

$$\begin{aligned}
&\text{双边检验:} && H_0: \ \sigma_1^2 = \sigma_2^2, && H_1: \ \sigma_1^2 \neq \sigma_2^2,\\
&\text{单边检验 I:} && H_0: \ \sigma_1^2 \leq \sigma_2^2, && H_1: \ \sigma_1^2 > \sigma_2^2,\\
&\text{单边检验 II:} && H_0: \ \sigma_1^2 \geq \sigma_2^2, && H_1: \ \sigma_1^2 < \sigma_2^2.
\end{aligned}$$

分均值 μ_1, μ_2 已知和未知两种情况讨论.

当 μ_1 与 μ_2 已知时, 令 $\widehat{\sigma}_1^2 = \dfrac{1}{n_1}\sum\limits_{i=1}^{n_1}(X_i - \mu_1)^2$, $\widehat{\sigma}_2^2 = \dfrac{1}{n_2}\sum\limits_{i=1}^{n_2}(Y_i - \mu_2)^2$, 当

H_0 为真时,

$$F = \frac{\widehat{\sigma}_1^2}{\widehat{\sigma}_2^2} \quad \sim \quad F(n_1, n_2), \tag{5.14}$$

因此用 F 来确定拒绝域, 即当

双边检验: $F \geq F_{\alpha/2}(n_1, n_2)$ 或 $F \leq F_{1-\alpha/2}(n_1, n_2)$,

单边检验 I: $F \geq F_{\alpha}(n_1, n_2)$,

单边检验 II: $F \leq F_{1-\alpha}(n_1, n_2)$.

则认为 H_0 不成立.

当 μ_1 与 μ_2 未知时, 当 H_0 为真, 有

$$F = \frac{S_1^2}{S_2^2} \ \sim \ F(n_1 - 1, n_2 - 1). \tag{5.15}$$

因此用 F 来确定拒绝域, 即当

双边检验: $F \geq F_{\alpha/2}(n_1 - 1, n_2 - 1)$ 或 $F \leq F_{1-\alpha/2}(n_1 - 1, n_2 - 1)$,

单边检验 I: $F \geq F_{\alpha}(n_1 - 1, n_2 - 1)$,

单边检验 II: $F \leq F_{1-\alpha}(n_1 - 1, n_2 - 1)$.

则认为 H_0 不成立.

根据公式 (5.14) 和公式 (5.15) 写出均值已知和均值未知两种情况方差比检验的 R 程序 (程序名: `var.test2.R`).

```
var.test2<-function(x, y, mu=c(Inf, Inf), side=0){
   source("P_value.R")
   n1<-length(x); n2<-length(y)
   if (all(mu<Inf)){
      Sx2<-sum((x-mu[1])^2)/n1; Sy2<-sum((y-mu[2])^2)/n2
      df1=n1; df2=n2
   }
   else{
      Sx2<-var(x); Sy2<-var(y); df1=n1-1; df2=n2-1
   }
```

```
    r<-Sx2/Sy2
    P<-P_value(pf, r, paramet=c(df1, df2), side=side)
    data.frame(rate=r, df1=df1, df2=df2, F=r, P_value=P)
}
```

在程序中，x，y 是来自两总体的数据向量. mu 是均值，当均值已知时，采用自由度为 (n_1, n_2) 的 $F-$ 分布计算 F 值；否则，采用自由度为 (n_1-1, n_2-1) 的 $F-$ 分布计算 F 值. side 是指双边检验还是单边检验. 当 side = 0 作双边检验，其备择假设为： $\sigma_1^2 \neq \sigma_2^2$; 当 side < 0 作单边检验，其备择假设为：$\sigma_1^2 < \sigma_2^2$; 当 side > 0 作单边检验，其备择假设为： $\sigma_1^2 > \sigma_2^2$.

输出采用数据框形式，输出的变量有：方差比 rate, 第一自由度 df1, 第二自由度 df2, F 值和 P- 值.

例 5.5 试对例 *5.3* 中的数据假设检验

$$H_0: \quad \sigma_1^2 = \sigma_2^2, \qquad H_1: \quad \sigma_1^2 \neq \sigma_2^2.$$

解：输出数据，调用 var.test2() 函数

```
> X<-c(78.1,72.4,76.2,74.3,77.4,78.4,76.0,75.5,76.7,77.3)
> Y<-c(79.1,81.0,77.3,79.1,80.0,79.1,79.1,77.3,80.2,82.1)
> source("var.test2.R")
> var.test2(X,Y)
      rate df1 df2        F   P_value
1 1.494481   9   9 1.494481 0.5590224
```

$P-$ 值为 $0.559 \gg 0.05$, 因此，无法拒绝原假设，认为两总体的方差是相同的. 这也说明在例 5.3 中，假设两总体方差相同是合理的.

用两总体方差比的区间估计也能作样本的方差检验.

```
#### 调用方差的区间估计函数 interval_var4
> source("interval_var4.R")
#### 作方差比的区间估计，考虑均值未知的情况
> interval_var4(X, Y)
      rate df1 df2         a        b
1 1.494481   9   9 0.3712079 6.016771
```

由于方差比 1 在所估计的区间内，因此认为方差是相同的.

在 R 软件中， var.test() 函数提供作方差比的检验和相应的区间估计. 该函数的使用格式是

```
var.test(x, y, ratio = 1,
         alternative = c("two.sided", "less", "greater"),
         conf.level = 0.95, ...)
```

其中 x,y 是来自两样本数据构成的向量， ratio 是方差比的原假设，缺省值为 1. alternative 是备择假设， two.sided 表示双边检验 ($H_1 : \sigma_1^2/\sigma_2^2 \neq$ ratio), less 表示单边检验 ($H_1 : \sigma_1^2/\sigma_2^2 <$ ratio), greater 表示单边检验 ($H_1 : \sigma_1^2/\sigma_2^2 >$ ratio).

下面用 var.test() 函数计算例 5.5.

```
> var.test(X,Y)

        F test to compare two variances

data:  X and Y
F = 1.4945, num df = 9, denom df = 9, p-value = 0.559
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.3712079 6.0167710
sample estimates:
ratio of variances
          1.494481
```

与我们前面的计算结果是完全相同的. 后面还将介绍直接用 var.test() 作双总体方差比的检验或方差比的区间估计. 这个例子也使我们可以更清楚的了解函数 var.test() 的计算过程.

5.2.3 二项分布总体的假设检验

前面介绍的是正态总体的假设检验问题, 这里介绍非正态总体的检验问题. 关于非正态总体的检验有很多，这里只介绍二项分布的假设检验问题.

类似于正态分布，我们也可以推导出二项分布的统计量和所服从的分布，导出相应的估计值 (点估计和区间估计), 以及相应的假设检验方法. 这里我们仅给出 R 软件中关于二项分布检验和估计的函数 binom.test().

binom.test() 函数的使用方法是：

```
binom.test(x, n, p = 0.5,
           alternative = c("two.sided", "less", "greater"),
           conf.level = 0.95)
```

其中 x 是成功的次数；或是一个由成功数和失败数构成的二维向量. n 是试验总数，当 x 是二维向量时，此值无效. p 是原假设的概率.

例 5.6 有一批蔬菜种子的平均发芽率 $p_0 = 0.85$, 现随机抽取 500 粒，用种衣剂进行浸种处理，结果有 445 粒发芽. 试检验种衣剂对种子发芽率有无效果.

解：根据题意，所检验的问题为：

$$H_0 : p = p_0 = 0.85, \quad H_1 : p \neq p_0.$$

调用 binom.test() 函数,

```
> binom.test(445,500,p=0.85)
        Exact binomial test
data:  445 and 500
number of successes = 445, number of trials = 500, p-value = 0.01207
alternative hypothesis: true probability of success is not equal to 0.85
95 percent confidence interval:
 0.8592342 0.9160509
sample estimates:
probability of success
                  0.89
```

$P-$ 值 $= 0.01207 < 0.05$, 拒绝原假设，认为种衣剂对种子发芽率有显著效果，从区间估计值来看，种衣剂可以提高种子的发芽率.

我们可作单侧检验来证实这一结论. 下面举一个单侧检验的例子.

例 5.7 据以往经验，新生儿染色体异常率一般为 1%, 某医院观察了当地 400 名新生儿，只有 1 例染色体异常，问该地区新生儿染色体异常是否低于一般水平？

解：根据题意，所检验的问题为：

$$H_0 : p \geq 0.01, \quad H_1 : p < 0.01.$$

调用 binom.test() 函数,

```
> binom.test(1, 400, p = 0.01, alternative = "less")

        Exact binomial test

data:  1 and 400
number of successes = 1, number of trials = 400, p-value = 0.09048
alternative hypothesis: true probability of success is less than 0.01
95 percent confidence interval:
 0.00000000 0.01180430
sample estimates:
probability of success
                0.0025
```

$P-$ 值 $= 0.09048 > 0.05 = \alpha$, 并不能认为该地区新生儿染色体异常率低于一般水平. 另外, 从区间估计值也能说明这一点, 区间估计的上界为 $0.0118 > 0.01$.

另一种输入方法

```
> binom.test(c(1, 399), p = 0.01, alternative = "less")
```

具有同样的结果.

5.3 若干重要的非参数检验

在统计推断问题中, 若给定或假定了总体分布的具体形式 (如正态分布), 只是其中含有若干未知参数, 要基于来自总体分布对参数做出估计或者进行某种形式的假设检验, 这类推断方法称为参数方法.

但在许多实际问题中, 人们往往对总体的分布知之甚少, 很难对总体的分布形式作出正确的假定, 最多只能对总体的分布做出诸如连续型分布、关于某点对称分布等一般性的假定. 这种不假定总体分布的具体形式, 尽量从数据 (或样本) 本身来获得所需要的信息的统计方法称为非参数方法.

对于非参数方法的检验问题称为非参数检验法, 它涉及的范围很广, 这里只能介绍几种与 R 软件有关的、在应用上较为重要的检验法.

5.3.1 Pearson 拟合优度 χ^2 检验

前面几节介绍的假设检验问题称为参数检验问题, 即事先认为样本分布具有某种指定的形式, 而其中的一些参数未知, 检验的目标是关于某个参数落在特定

的范围内的假设. 这里要介绍的是另一类假设, 其目标不是针对具体的参数, 而是针对分布的类型. 例如, 通常假定总体分布具有正态性, 则 "总体分布为正态" 这一断言本身在一定场合下就是可疑的, 有待于检验.

在第三章, 我们通过直方图、 QQ 图和经验分布图大概描述观测数据是否服从某种分布, 这里介绍如何用统计方法检验观测数据是否服从某种分布. 在第三章介绍的 W 正态性检验和 Kolmogorov-Smirnov 检验都属于拟合优度检验.

1. 理论分布完全已知的情况

假设根据某理论、学说甚至假定, 某随机变量应当有分布 F, 现对 X 进行 n 次观察, 得到一个样本 $X_1, X_2, \cdots, X_n$, 要据以检验

$$H_0: \ X \text{ 具有分布 } F.$$

这里虽然没有明确指出对立假设, 但可以说, 对立假设是

$$H_1: \ X \text{ 不具有分布 } F.$$

本问题的真实含义是估量实测数据与该理论或学说符合得怎么样, 而不在于当认为不符合时, X 可能备择的分布如何. 故问题中不明确标出对立假设, 反而使人感到提法更为贴近现实.

上述问题的检验方法是, 将数轴 $(-\infty, \infty)$ 分成 m 个区间:

$$I_1 = (-\infty, a_1), I_2 = [a_1, a_2), \cdots, I_m = [a_{m-1}, \infty).$$

记这些区间的理论概率分别为

$$p_1, p_2, \cdots, p_m, \quad p_i = P\{X \in I_i\}, \ i = 1, 2, \cdots, m.$$

记 n_i 为 $X_1, X_2, \cdots, X_n$ 中落在区间 I_i 内的个数, 则在原假设成立下, n_i 的期望值为 np_i, n_i 与 np_i 的差距 $(i = 1, 2, \cdots, m)$ 可视为理论与观察之间偏离的衡量, 将它结合起来形成一个综合指标: $\sum_{i=1}^{m} c_i(n_i - np_i)^2$, 其中 $c_i > 0$ 为适当的常数, 通常取 $c_i = 1/np_i$, 因此得到统计量

$$K = \sum_{i=1}^{m} \frac{(n_i - np_i)^2}{np_i}, \tag{5.16}$$

称 K 为 Pearson χ^2 统计量. Pearson 证明了, 在原假设成立的条件下, 当 $n \to \infty$ 时, K 依分布收敛于自由度为 $m-1$ 的 χ^2 分布. 在这个基础上, 引进一个大

样本检验：给定显著性水平 α, 当

$$K > \chi^2_\alpha(m-1) \tag{5.17}$$

则拒绝原假设. 这就是 Neyman-Pearson 拟合优度 χ^2 检验.

这个问题还可以讨论得更细一些, 按式 (5.17), 只要 $K > \chi^2_\alpha(m-1)$, 就否定原假设, 但是一个远远大于 $\chi^2_\alpha(m-1)$ 的 K 与一个只略大于 $\chi^2_\alpha(m-1)$ 的 K, 意义有所不同, 前者否定的理由更强一些. 为反映这一点, 在计算出 K 值后, 可计算出 $P-$ 值,

$$P - \text{值} = P\{\chi^2(m-1) > K\}. \tag{5.18}$$

可将 $P-$ 值称为所得数据与原假设的似合优度. $P-$ 值越大, 支持原假设的证据就越强. 给定一个显著性水平 α, 当 $P-$ 值 $< \alpha$, 就拒绝原假设.

例 5.8 某消费者协会为了确定市场上消费者对 *5* 种品牌啤酒的喜好情况, 随机抽取了 *1000* 名啤酒爱好者作为样本进行如下试验：每个人得到 *5* 种品牌的啤酒各一瓶, 但未标明牌子. 这 *5* 种啤酒按分别写着 A 、 B 、 C 、 D 、 E 字母的 *5* 张纸片随机的顺序送给每一个人. 表 *5.1* 是根据样本资料整理得到的各种品牌啤酒爱好者的频数分布. 试根据这些数据判断消费者对这 *5* 种品牌啤酒的爱好有无明显差异？

表 5.1: 5 种品牌啤酒爱好者的频数

最喜欢的牌子	A	B	C	D	E
人数 X	210	312	170	85	223

解: 如果消费者对 5 种品牌啤酒喜好无显著差异, 那么, 就可以认为喜好这 5 种品牌啤酒的人呈均匀分布, 即 5 种品牌啤酒爱好者人数各占 20%. 据此假设：

$$H_0 : \text{喜好 5 种啤酒的人数分布均匀.}$$

按式 (5.16) 和式 (5.17) 编写计算公式, 用 R 软件计算.

```
> X<-c(210, 312, 170, 85, 223)
> n<-sum(X); m<-length(X)
> p<-rep(1/m, m)
```

```
> K<-sum((X-n*p)^2/(n*p));K
[1] 136.49
> Pr<-1-pchisq(K, m-1);Pr
[1] 0
```

$P-$ 值为 0, 因此，拒绝原假设，认为消费者对 5 种品牌啤酒的喜好是有明显差异.

我们可以将上述过程编写成一个程序进行计算，实际上， R 软件已完成了此项工作，所提供的 chisq.test() 函数可以方便地完成此项工作. 我们只需输入

```
> chisq.test(X)
```

就可以得到

```
        Chi-squared test for given probabilities
data:  X
X-squared = 136.49, df = 4, p-value < 2.2e-16
```

chisq.test() 函数的使用格式为

```
chisq.test(x, y = NULL, correct = TRUE,
    p = rep(1/length(x), length(x)), rescale.p = FALSE,
    simulate.p.value = FALSE, B = 2000)
```

其中 x 是由观测数据构成的向量或矩阵， y 是数据向量 (当 x 为矩阵时， y 无效). correct 是逻辑变量，表明是否用于连续修正， TRUE(缺省值) 表示修正，FALSE 表示不修正. p 是原假设落在小区间的理论概率, 缺省值表示均匀分布. rescale.p 是逻辑变量，选择 FALSE(缺省值) 时，要求输入的 p 满足 $\sum_{i=1}^{m} p_i = 1$; 选择 TRUE 时，并不要求这一点，程序将重新计算 p 值. simulate.p.value 是逻辑变量 (缺省值为 FALSE), 当为 TRUE, 将用仿真的方法计算 $P-$ 值，此时， B 表示仿真的次数.

例 5.9 用 *Pearson* 拟合优度 χ^2 检验方法检验例 *3.6*中学生成绩是否服从正态分布.

解: 我们分几步进行，然后将这些步骤编写成 R 程序进行计算.

第一步：先输入数据，这里用 scan() 函数.

第二步: 对 31 名学生成绩进行分组, 计算各组的频数, 其中 $A_1 = \{X < 70\}$,

$A_2 = \{70 \le X < 80\}$, $A_3 = \{80 \le X < 90\}$, $A_4 = \{90 \le X \le 100\}$. 这里调用 cut() 函数和 table() 函数进行分组和记数.

第三步：计算原假设 (正态分布) 在各小区间的理论概率值. 先计算学生成绩的均值 (mean) 、标准差 (sd), 再用 pnorm() 计算理论概率.

第四步：作 Pearson χ^2 检验. 调用 chisq.test() 函数.

下面写出相应的 R 程序 (程序名： exam0509.R)

```
#### 第一步, 输入数据
X<-scan()
25  45  50  54  55  61  64  68  72  75  75
78  79  81  83  84  84  84  85  86  86  86
87  89  89  89  90  91  91  92  100

#### 第二步, 分组和记数
A<-table(cut(X, br=c(0,69,79,89,100)))
#### 第三步, 构造理论分布
p<-pnorm(c(70,80,90,100), mean(X), sd(X))
p<-c(p[1], p[2]-p[1], p[3]-p[2], 1-p[3])
#### 第四步, 作检验
chisq.test(A,p=p)
```

计算结果如下：

```
        Chi-squared test for given probabilities
data:  A
X-squared = 8.334, df = 3, p-value = 0.03959
```

$P-$ 值 $= 0.03959 < 0.05$, 因此认为该门课程的成绩不服从正态分布.

在这个例子中用到了两个函数, 一个是 cut() 函数, 另一个是 table() 函数, 下面简单介绍这两个函数的用法.

cut() 函数是将变量的区域分成若干个区间, 其使用方法是：

```
cut(x, breaks, labels = NULL,
    include.lowest = FALSE, right = TRUE, dig.lab = 3, ...)
```

其中 x 是由数据构成的向量， breaks(简写为 br) 是所分区间的端点构成的向量.

table() 函数是计算因子合并后的个数，其使用方法是：

```
table(..., exclude = c(NA, NaN), dnn = list.names(...),
      deparse.level = 1)
```

这里用这两个函数计算随机变量落在某个区间的频数.

例 5.10 大麦的杂交后代关于芒性的比例应是无芒：长芒：短芒 $= 9:3:4$. 实际观测值为 $335:125:160$. 试检验观测值是否符合理论假设？

解：根据题意,

$$H_0: \quad p_1 = \frac{9}{16}, \quad p_2 = \frac{3}{16}, \quad p_3 = \frac{4}{16}.$$

调用 chisq.test() 函数

```
> chisq.test(c(335, 125, 160), p=c(9,3,4)/16)

        Chi-squared test for given probabilities

data:  c(335, 125, 160)
X-squared = 1.362, df = 2, p-value = 0.5061
```

$P-$ 值 $= 0.5061 > 0.05$, 接受原假设，即大麦芒性的分离符合 $9:3:4$ 的比例.

例 5.11 为研究电话总机在某段时间内接到的呼叫次数是否服从 *Poisson* 分布，现收集了 *42* 个数据，如表 *5.2* 所示. 通过对数据的分析，问能否确认在某段时间内接到的呼叫次数服从 *Poisson* 分布 $(\alpha = 0.1)$*?*

表 5.2: 电话总机在某段时间内接到的呼叫次数的频数

接到呼唤次数	0	1	2	3	4	5	6
出现的频数	7	10	12	8	3	2	0

解：编写相应的计算程序 (程序名： exam0511.R)

```
#### 输入数据
X<-0:6; Y<-c(7, 10, 12, 8, 3, 2, 0)
#### 计算理论分布，其中 mean(rep(X,Y)) 为样本均值
```

```
q<-ppois(X, mean(rep(X,Y))); n<-length(Y)
p[1]<-q[1]; p[n]<-1-q[n-1]
for (i in 2:(n-1))
   p[i]<-q[i]-q[i-1]
#### 作检验
chisq.test(Y, p=p)
```

但计算结果会出现警告.

```
        Chi-squared test for given probabilities
data:  Y
X-squared = 1.5057, df = 6, p-value = 0.9591
Warning message:
Chi-squared 近似算法有可能不准 in: chisq.test(Y, p = p)
```

为什么会出现这种情况呢？这是因为 Pearson χ^2 检验要求在分组后，每组中的频数至少要大于等于 5, 而后三组中出现的频数分别为 3, 2, 0, 均小于 5. 解决问题的方法是将后三组合成一组，此时的频数为 5, 满足要求. 下面给出相应的 R 程序.

```
#### 重新分组
Z<-c(7, 10, 12, 8, 5)
#### 重新计算理论分布
n<-length(Z); p<-p[1:n-1]; p[n]<-1-q[n-1]
#### 作检验
chisq.test(Z, p=p)
```

计算得到

```
        Chi-squared test for given probabilities
data:  Z
X-squared = 0.5389, df = 4, p-value = 0.9696
```

P– 值 $\gg 0.1$, 因此，能确认在某段时间内接到的呼叫次数服从 Poisson 分布.

从例 5.11 的结果可以看出，在习题 4.9 中，将在某段时间内接到的呼叫次认为数服从 Poisson 分布是合理的.

2. 理论分布依赖于若干个未知参数的情况

如果分布族 F 依赖于 r 个参数 $\theta_1, \theta_2, \cdots, \theta_r$, 要根据样本 $X_1, X_2, \cdots, X_n$ 去检验假设

$$H: X \text{ 的分布属于 } \{F(x, \theta_1, \theta_2, \cdots, \theta_r)\}.$$

解决这个问题的步骤是，先通过样本作出 $(\theta_1, \theta_2, \cdots, \theta_r)$ 的极大似然估计 $(\widehat{\theta}_1, \widehat{\theta}_2, \cdots, \widehat{\theta}_r)$ 再检验假设

$$H: X \text{ 有分布 } F(x, \widehat{\theta}_1, \widehat{\theta}_2, \cdots, \widehat{\theta}_r).$$

然后再按理论分布已知的情况进行处理，所不同的是由式 (5.16) 得到的统计量 K 服从自由度为 $m-1-r$ 的 χ^2 分布，即自由度减少了 r.

5.3.2 Kolmogorov-Smirnov 检验

在第三章描述性统计中，介绍了 Kolmogorov-Smirnov 检验，该检验实际上是属于拟合优度检验. 这里再进出一步介绍它的使用方法.

Kolmogorov-Smirnov 检验有单样本检验和双样本检验，在第三章中我们介绍的就是单样本检验的使用方法.

1. 单样本检验

通过第三章的介绍，我们知道 Kolmogorov-Smirnov 检验是通过经验分布与假设分布的上确界来构造统计量的，因此理论上可以检验任何分布，即原假设为

$$H_0: \ X \text{ 具有分布 } F.$$

R 软件提供了 Kolmogorov-Smirnov 检验的函数 `ks.test()`, 我们用例子进一步说明它的使用方法.

例 5.12 对一台设备进行寿命检验，纪录 *10* 次无故障工作时间，并按从小到大的次序排列如下： *(*单位：小时*)*

420 500 920 1380 1510 1650 1760 2100 2300 2350

试用 *Kolmogorov-Smirnov* 检验方法检验此设备无故障工作时间的分布是否服从 $\lambda = 1/1500$ 的指数分布？

解: 输入数据，调用 `ks.test()` 函数.

```
> X<-c(420, 500, 920, 1380, 1510, 1650, 1760, 2100, 2300, 2350)
> ks.test(X, "pexp", 1/1500)
```

```
        One-sample Kolmogorov-Smirnov test
data:  X
D = 0.3015, p-value = 0.3234
alternative hypothesis: two.sided
```

其 $P-$ 值大于 0.05, 无法拒绝原假设, 因此认为此设备无故障工作时间的分布服从 $\lambda = 1/1500$ 的指数分布.

2. 双样本检验

假设 $X_1, X_2, \cdots, X_{n_1}$ 为来自分布为 $F(x)$ 总体的样本, 且 $F(x)$ 未知, $Y_1, Y_2, \cdots, Y_{n_2}$ 为来自分布为 $G(x)$ 总体的样本, 且 $G(x)$ 未知. 假定 $F(x)$ 和 $G(x)$ 均为连续分布函数, 检验这两分布是否相同, 即原假设为

$$H_0: \ F(x) = G(x).$$

例 5.13 假定从分布函数为未知的 $F(x)$ 和 $G(x)$ 的总体中分别抽出 *25* 个和 *20* 个观察值的随机样本, 其数据由表 *5.3* 所示. 现检验 $F(x)$ 和 $G(x)$ 是否相同.

表 5.3: 抽自不同分布的数据

$F(x)$	0.61	0.29	0.06	0.59	−1.73	−0.74	0.51	−0.56	0.39
	1.64	0.05	−0.06	0.64	−0.82	0.37	1.77	1.09	−1.28
	2.36	1.31	1.05	−0.32	−0.40	1.06	−2.47		
$G(x)$	2.20	1.66	1.38	0.20	0.36	0.00	0.96	1.56	0.44
	1.50	−0.30	0.66	2.31	3.29	−0.27	−0.37	0.38	0.70
	0.52	−0.71							

解: 编写相应的计算程序 (程序名: exam0513.R).

```
#### 输入数据
X<-scan()
0.61  0.29  0.06  0.59  -1.73 -0.74  0.51 -0.56  0.39
1.64  0.05 -0.06  0.64  -0.82  0.37  1.77  1.09 -1.28
2.36  1.31  1.05 -0.32  -0.40  1.06 -2.47
```

```
Y<-scan()
2.20  1.66  1.38  0.20  0.36  0.00  0.96  1.56  0.44
1.50 -0.30  0.66  2.31  3.29 -0.27 -0.37  0.38  0.70
0.52 -0.71

#### 作 K-S 检验
ks.test(X, Y)
```

运行后得到

```
        Two-sample Kolmogorov-Smirnov test
data:  X and Y
D = 0.23, p-value = 0.5286
alternative hypothesis: two.sided
```

$P-$ 值大于 0.05, 故接受原假设 H_0, 即认为 $F(x)$ 和 $G(x)$ 两个分布函数相同.

Kolmogorov-Smirnov 检验与 Pearson χ^2 检验相比， Kolmogorov 检验不须将样本分组, 少了一个任意性, 这是其优点. 其缺点是只有用在理论分布为一维连续分布且分布完全已知的情形, 适用面比 Pearson 检验小. 研究也显示：在 Kolmogorov 检验可用的场合下, 其功效一般来说略优于 Pearson 检验.

5.3.3 列联表数据的独立性检验

设两个随要变量 X, Y 均为离散型的， X 取值于 $\{a_1, a_2, \cdots, a_I\}$, Y 的取值于 $\{b_1, b_2, \cdots, b_J\}$. 设 $(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n)$ 为简单样本, 记 n_{ij} 为 $(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n)$ 中等于 (a_i, b_j) 的个数, 要据此检验假设

$$H_0 : X \text{ 与 } Y \text{ 独立.}$$

1. Pearson χ^2 检验

在求解问题时, 常把数据列为表 5.4 的形式, 称为列联表 (contingency table).

记

$$\begin{aligned} p_{ij} &= P\{X_i = a_i, Y_j = b_j\}, \\ p_{i\cdot} &= P\{X_i = a_i\} = \sum_{j=1}^{J} p_{ij}, \quad p_{\cdot j} = P\{Y_j = b_j\} = \sum_{i=1}^{I} p_{ij}, \end{aligned}$$

表 5.4: 列联表

	b_1	b_2	$\cdots$	b_J	$\sum$
a_1	n_{11}	n_{12}	$\cdots$	n_{1J}	$n_{1\cdot}$
a_2	n_{21}	n_{22}	$\cdots$	n_{2J}	$n_{2\cdot}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
a_I	n_{I1}	n_{I2}	$\cdots$	n_{IJ}	$n_{I\cdot}$
$\sum$	$n_{\cdot 1}$	$n_{\cdot 2}$	$\cdots$	$n_{\cdot J}$	

则假设 H 可表示为

$$H: \quad p_{ij} = p_{i\cdot} \cdot p_{\cdot j}, \qquad i = 1, 2, \cdots, I, \quad j = 1, 2, \cdots, J. \tag{5.19}$$

这里只知道 $p_{i\cdot}, p_{\cdot j} \geq 0$, $\sum\limits_{i=1}^{I} p_{i\cdot} = 1$, $\sum\limits_{j=1}^{J} p_{\cdot j} = 1$, 而其它情况未知，所以这是一个带参数 $p_{i\cdot}, (i = 1, 2, \cdots, I)$, $p_{\cdot j}, (j = 1, 2, \cdots, J)$ 的拟合优度检验问题. 因此，需要先用极大似然估计来估计 $p_{i\cdot}$, $p_{\cdot j}$, 得到

$$\begin{aligned} \hat{p}_{i\cdot} &= \frac{n_{i\cdot}}{n}, \qquad i = 1, 2, \cdots, I, \\ \hat{p}_{\cdot j} &= \frac{n_{\cdot j}}{n}, \qquad j = 1, 2, \cdots, J, \end{aligned}$$

其中 $n_{i\cdot} = \sum\limits_{j=1}^{J} n_{ij}$, $n_{\cdot j} = \sum\limits_{i=1}^{I} n_{ij}$. 这样就可以计算 Pearson χ^2 统计量

$$K = \sum_{i=1}^{I}\sum_{j=1}^{J} \frac{\left[n_{ij} - n\left(\frac{n_{i\cdot}}{n}\right)\left(\frac{n_{\cdot j}}{n}\right)\right]^2}{n\left(\frac{n_{i\cdot}}{n}\right)\left(\frac{n_{\cdot j}}{n}\right)} = \sum_{i=1}^{I}\sum_{j=1}^{J} \frac{\left[n \cdot n_{ij} - n_{i\cdot} \cdot n_{\cdot j}\right]^2}{n \cdot n_{i\cdot} \cdot n_{\cdot j}}. \tag{5.20}$$

然后再计算自由度. (X, Y) 的值域一共划分成 IJ 个集合，但估计了一些未知参数. 由于 $\sum\limits_{i=1}^{I} p_{i\cdot} = 1$, $p_{i\cdot}(i = 1, 2, \cdots, I)$ 中未知参数只有 $I - 1$ 个，同理，$p_{\cdot j}(j = 1, 2, \cdots, J)$ 中未知参数只有 $J - 1$ 个，故共有 $I + J - 2$ 个未知参数，而 K 的自由度就为

$$IJ - 1 - (I + J - 2) = (I - 1)(J - 1).$$

这样在计算出 K 值后，其拒绝域为

$$K > \chi^2_\alpha((I - 1)(J - 1)).$$

或计算其 $P-$ 值

$$P-\text{值}=P\{\chi^2((I-1)(J-1))>K\}.$$

当 $I=J=2$ 时，列联表中只有 4 个格子，称为 “四格表”，这时式 (5.20) 简单化为

$$K=\frac{n(n_{11}n_{22}-n_{12}n_{21})^2}{n_{1\cdot}n_{2\cdot}n_{\cdot 1}n_{\cdot 2}},$$

自由度为 1.

chisq.test() 函数也可以作独立性检验，只需将列联表数据写成矩阵形式即可.

例 5.14 为了研究吸烟是否与患肺癌有关，对 *63* 位肺癌患者及 *43* 名非肺癌患者 *(*对照组*)* 调查了其中的吸烟人数，得到 2×2 列联表，如表 *5.5* 所示.

表 5.5: 列联表数据

	患肺癌	未患肺癌	合计
吸烟	60	32	92
不吸烟	3	11	14
合计	63	43	106

解: 输入数据，用 chisq.test() 作检验.

```
> x<-c(60, 3, 32, 11)
> dim(x)<-c(2,2)
> chisq.test(x,correct = FALSE)
        Pearson's Chi-squared test
data:  x
X-squared = 9.6636, df = 1, p-value = 0.001880
```

或带连续校正.

```
> chisq.test(x)
        Pearson's Chi-squared test with
           Yates' continuity correction
data:  x
X-squared = 7.9327, df = 1, p-value = 0.004855
```

无论是哪种方法，其 $P-$ 值均小于 0.05, 因此拒绝原假设，也就是说吸烟与患肺癌有关.

例 5.15 *在一次社会调查中，以问卷方式调查了总共 901 人的年收入及对工作的满意程度，其中年收入 A 分为小于 6000 元、 6000 元至 15000 元、 15000 元至 25000 元及超过 25000 元四档. 对工作的满意程度 B 分为很不满意、较不满意、基本满意和很满意四档. 调查结果用 4×4 列联表表示，如表 5.6 所示.*

表 5.6: 工作满意程度与年收入列联表

	很不满意	较不满意	基本满意	很满意	合计
<6000	20	24	80	82	206
$6000\sim 15000$	22	38	104	125	289
$15000\sim 25000$	13	28	81	113	235
>25000	7	18	54	92	171
合计	62	108	319	412	901

解: 输入数据，用 `chisq.test()` 作检验.

```
x<-scan()
20 24  80  82  22 38 104 125
13 28  81 113   7 18  54  92

dim(x)<-c(4,4)
chisq.test(x)
        Pearson's Chi-squared test
data:  x
X-squared = 11.9886, df = 9, p-value = 0.2140
```

其 $P-$ 值均大于 0.05, 接受原假设，即工作的满意程度与年收入无关.

在用 `chisq.test()` 函数作计算时，要注意单元的期望频数. 如果没有空单元 (所有单元频数都不为零), 并且所有单元的期望频数大于等于 5, 那么 Pearson χ^2 检验是合理的；否则计算机会显示警告信息.

如果数据不满足 χ^2 检验的条件时，应使用 Fisher 精确检验.

2. Fisher 精确的独立检验

在样本较小时 (单元的期望频数小于 4), 需要用 Fisher 精确检验来作独立性检验.

Fisher 精确检验最初是针对 2×2 这种特殊的列联表提出的. 当 χ^2 检验的条件不满足时, 这个精确检验是非常有用的. Fisher 检验是建立在超几何分布的基础上, 对于单元频数小的表来说, 特别适合.

这里不再推导相关的统计量, 而是直接绘出 R 软件关于 Fisher 精确检验的方法.

例 5.16 *某医师为研究乙肝免疫球蛋白预防胎儿宫内感染 HBV 的效果, 将 33 例 HBsAg 阳性孕妇随机分为预防注射组和对照组, 结果由表 5.7 所示. 问两组新生儿的 HBV 总体感染率有无差别?*

表 5.7: 两组新生儿 HBV 感染率的比较

组别	阳性	阴性	合计	感染率 (%)
预防注射组	4	18	22	18.18
对照组	5	6	11	45.45
合计	9	24	33	27.27

解: 有一个单元频数小于 5, 应该作 Fisher 精确概率检验.

在 R 软件中, 函数 `fisher.test()` 作精确概率检验. 其使用方法是

```
fisher.test(x, y = NULL, workspace = 200000, hybrid = FALSE,
    control = list(), or = 1, alternative = "two.sided",
    conf.int = TRUE, conf.level = 0.95)
```

其中 `x` 是具有二维列联表形式的矩阵或是由因子构成的对象. `y` 是由因子构成的对象, 当 `x` 是矩阵时, 此值无效. `workspace` 的输入值是一整数, 其整数表示用于网络算法工作空间的大小. `hybrid` 为逻辑变量, `FALSE`(缺省值) 表示精确计算概率, `TRUE` 表示用混合算法计算概率. `alternative` 为备择, 有 `"two.sided"`(缺省值) 双边, `"less"` 单边小于, `"greater"` 单边大于. `conf.int` 逻辑变量, 当 `conf.int=TRUE`(缺省值), 给出区间估计. `conf.level` 为置信水平, 缺省值为 0.95. 其余参数见在线说明.

对于 2×2 列联表，原假设 “两变量无关” 等价于赔率比 (odds rate) 等于 1.

输入数据，并计算 Fisher 检验

```
> x<-c(4,5,18,6); dim(x)<-c(2,2)
> fisher.test(x)
        Fisher's Exact Test for Count Data
data:  x
p-value = 0.1210
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.03974151 1.76726409
sample estimates:
odds ratio
 0.2791061
```

因为 $P-$ 值 $=0.1210>0.05$, 且区间估计得到的区间包含有 1, 因此说明两变量是独立的，即认为两组新生儿的 HBV 总体感染率无差别.

如果用 Pearson χ^2 检验 (`chisq.test()` 函数) 对这组数据作检验时，你会发现计算机在得到结果的同时，给出警告，认为其计算值可能有误.

用 Fisher 精确检验 (`fisher.test()` 函数), 对例 5.14 的数据作检验，得到

```
> x<-c(60, 3, 32, 11); dim(x)<-c(2,2)
> fisher.test(x)
        Fisher's Exact Test for Count Data
data:  x
p-value = 0.002820
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
  1.626301 40.358904
sample estimates:
odds ratio
   6.74691
```

其 $P-$ 值小于 0.05, 因此拒绝原假设，即认为吸烟与患肺癌有关. 由于赔率比大于 1, 因此还是正相关，也就是说，吸烟越多，患肺癌的可能性也就越大.

3. McNemar 检验

McNemar 检验虽然不是独立性检验，但它是关于列联表数据的检验，所以放在这里来处理.

McNemar 检验是在相同个体上的两次检验，检验两无数据的两个相关分布的频数比变化的显著性.

如果作为样本的一批个体分别在某一时间间隔或不同条件下作两次研究，比如是关于二元特征的强度，那么确定研究的不再是独立的样本，而是相关样本. 每个试验单元可提供一对数据. 从第一次到第二次研究中，两种选择的频数比率有或多或少的改变. McNemar 检验是检验这个变化强度，它能较精确地得知在第一次和第二次研究之间有多少个体从这一类变成另一类. 我们可以得出具有第一次研究划分出的两类和第二次研究划分出的两类的列联表，如图 5.8 所示.

表 5.8: 不同方法的研究结果

研究 I	研究 II		合计
	$+$	$-$	
$+$	a	b	$a+b$
$-$	c	d	$c+d$
合计	$a+c$	$b+d$	$a+b+c+d$

问题的原假设为

$$H_0: \text{在这个总体中两次研究的频数没有区别.}$$

原假设表示频数 b 和 c 只表示在这个样本中的随机变差.

在 R 软件中， mcnemar.test() 函数给出了 McNemar 检验，其具体的使用方法是

```
mcnemar.test(x, y = NULL, correct = TRUE)
```

其中 x 是具有二维列联表形式的矩阵或是由因子构成的对象. y 是由因子构成的对象，当 x 是矩阵时，此值无效. correct 是逻辑变量， TRUE (缺省值) 表示在计算检验统计量时用连续修正， FALSE 是不用修正.

例 5.17 某胸科医院同时用甲、乙两种方法测定 *202* 份痰标本中的抗酸杆菌，结果如表 *5.9* 所示. 问甲、乙两法的检出率有无差别？

表 5.9: 甲、乙两法检测痰标本中的抗酸杆菌结果

甲法	乙 法		合计
	+	−	
+	49	25	74
−	21	107	128
合计	70	132	202

解: 输入数据, 调用 mcnemar.test() 函数作 McNemar 检验.

```
> X<-c(49, 21, 25, 107); dim(X)<-c(2,2)
> mcnemar.test(X,correct=FALSE)
        McNemar's Chi-squared test
data:  X
McNemar's chi-squared = 0.3478, df = 1, p-value = 0.5553
```

其统计量为 0.3478, $P-$ 值为 $0.5553 > 0.05$, 因此, 不能认定两种检测方法的检出率有差异.

5.3.4 符号检验

1. 检验一个样本是否来自某个总体

假设某个总体的中位数为 M_0, 如果样本中位数 $M = M_0$, 我们就接受样本来自某个总体的假设. 其具体的检验方法是这样的. 首先从每个样本观察值中减去总体中位数 M_0, 得出的正、负差额用正 $(+)$ 、负 $(-)$ 号加以表示. 如果总体中位数等于样本中位数, 即 $M = M_0$, 那么, 样本观察值在中位数上、下的数目应各占一半, 因现时出现正号或负号的概率应各占 $1/2$. 设样本容量为 n, 就可以用二项分布 $B(n, 1/2)$ 来计算出现负号 (或正号) 个数的概率, 从而根据一定的显著性水平 α, 作出是否接受原假设 $H_0 : M = M_0$ 的判定.

例 5.18 *联合国人员在世界上 66 个大城市的生活花费指数 (以纽约市 1996 年 12 月为 100) 按自小至大的次序排列如下 (这里北京的指数为 99)*:

66 75 78 80 81 81 82 83 83 83 83
84 85 85 86 86 86 86 87 87 88 88

```
 88   88   88   89   89   89   89   90   90   91   91
 91   91   92   93   93   96   96   96   97   99  100
101  102  103  103  104  104  104  105  106  109  109
110  110  110  111  113  115  116  117  118  155  192
```

假设这个样本是从世界许多大城市中随机抽样得到的. 试用符号检验分析, 北京是在中位数之上, 还是在中位数之下.

解: 样本的中位数 (M) 作为城市生活水平的中间值, 因此需要检验:

$$H_0 : M \geq 99, \quad H_1 : M < 99.$$

输入数据, 作二项检验.

```
> X<-scan()
1:  66   75   78   80   81   81   82   83   83   83   83
12:  84   85   85   86   86   86   86   87   87   88   88
23:  88   88   88   89   89   89   89   90   90   91   91
34:  91   91   92   93   93   96   96   96   97   99  100
45: 101  102  103  103  104  104  104  105  106  109  109
56: 110  110  110  111  113  115  116  117  118  155  192
67:
Read 66 items
> binom.test(sum(X>99), length(X), al="l")
        Exact binomial test
data:  sum(X > 99) and length(X)
number of successes = 23, number of trials = 66, p-value = 0.009329
alternative hypothesis: true probability of success is less than 0.5
95 percent confidence interval:
 0.0000000 0.4563087
sample estimates:
probability of success
             0.3484848
```

在程序中, `sum(X>99)` 表示样本中大于 99 的个数. `al` 是 `alternative` 的缩写, `"l"` 是 `"less"` 的缩写. 计算出的 $P-$ 值小于 0.05, 拒绝原假设, 也就是说,

北京的生活水平高于世界中间水平. 注意, 单侧区间估计的上界为 0.4563, 低于 0.5, 所得的结论还是拒绝原假设.

2. 用成对样本来检验两个总体间是否存在显著差异

符号检验法也可用于以成对随机样本观察值来检验两个总体之间是否存在显著差异. 如果两个总体无显者差异, 则两个成对随机样本观察值正、负差额的个数应大体相等. 假定 $x_i - y_i > 0$ 用正号表示, $x_i - y_i < 0$ 用负号表示, 则如果两个总体无显显著差异, 那么出现正号和负号的概率各占 $1/2$. 和上面检验样本是否来自某个总体一样, 可用二项分布 $B(n, 1/2)$, 根据一定的显著性水平和正号 (或负号) 的个数, 作出接受或拒绝两个总体无显著差异的判断.

例 5.19 用两种不同的饲料养猪, 其增重情况如表 *5.10* 所示. 试分析两种饲料养猪有无显著差异.

表 5.10: 不同饲料养猪的增重情况

对编号	1	2	3	4	5	6	7	8	9	10	11	12	13	14
饲料 X	25	30	28	23	27	35	30	28	32	29	30	30	31	16
饲料 Y	19	32	21	19	25	31	31	26	30	25	28	31	25	25

解: 采用成对符号检验. 输入数据, 调用 binom.test() 作检验.

```
> x<-scan()
1: 25 30 28 23 27 35 30 28 32 29 30 30 31 16
15:
Read 14 items
> y<-scan()
1: 19 32 21 19 25 31 31 26 30 25 28 31 25 25
15:
Read 14 items
> binom.test(sum(x<y), length(x))
        Exact binomial test
data:  sum(x < y) and length(x)
number of successes = 4, number of trials = 14, p-value = 0.1796
```

```
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.08388932 0.58103526
sample estimates:
probability of success
             0.2857143
```

sum(x < y) 表示样本 X 小于样本 Y 的个数. 计算出的 $P-$ 值大于 0.05, 无法拒绝原假设, 可以认为两种饲料养猪无显著差异. 计算出的区间估计包含 0.5, 也就是说, 可以认为 $X < Y$ 和 $X \geq Y$ 的概率各占 1/2, 得到的结论也不无法拒绝原假设, 两种饲料养猪无显著差异.

在人们的日常生活中, 常常遇到很难用数值确切表示的问题, 而符号检验法也可用于这类问题的研究, 例如我们要了解消费者是喜欢咖啡, 还是喜欢奶茶就属于这一类的问题. 消费者很难用 5 表示对咖啡的爱好, 或者用 8 表示对奶茶的爱好, 一般只能表示某消费者对咖啡的爱好超过奶茶, 或者对奶茶的爱好超过咖啡, 或者两者同样爱好. 因而我们可以用符号检验法来研究这一类的现象. 现举例说明这个检验方法的具体应用.

例 5.20 某饮料店为了解顾客对饮料的爱好情况, 进一步改进他们的工作, 对顾客喜欢咖啡还是喜欢奶茶, 或者两者同样爱好进行了调查. 该店在某日随机地抽取了 *13* 名顾客进行了调查, 顾客喜欢咖啡超过奶茶用正号表示, 喜欢奶茶超过咖啡用负号表示, 两者同样爱好用 *0* 表示. 现将调查的结果列在表 *5.11* 中. 试

表 5.11: 不同顾客的爱好情况

顾客编号	1	2	3	4	5	6	7	8	9	10	11	12	13
喜欢咖啡	1		1	1	1	0	1		1	1	1		1
喜欢奶茶		1						1				1	

分析顾客是喜欢咖啡还是喜欢奶茶.

解: 根据题意可检验如下假设:

H_0 : 顾客喜欢咖啡等于喜欢奶茶; H_1 : 顾客喜欢咖啡超过奶茶.

以上资料中有 1 人 (即 6 号顾客) 表示对咖啡和奶茶有同样爱好, 用 0 表示, 因而在样本容量中不加计算, 所以实际上 $n = 12$. 如果 H_0 假设为真, 即

顾客对咖啡和奶茶同样爱好，那么会出现 $x-y<0$, 即负号的概率为 $1/2$, 所以出现负号的个数服从二项分布， $B(12,1/2)$. 负号个数愈少，说明顾客喜欢咖啡超过奶茶的人数愈多，负号个数少到一定程度就要推翻 H_0 假设，而接受 H_1 假设，即顾客喜欢咖啡超过喜欢奶茶. 所以本例属于单边备择假设检验.

用 R 软件进行计算，显著性水平取 $\alpha=0.10$,

```
> binom.test(3,12,p=1/2, al="l", conf.level = 0.90)

        Exact binomial test

data:  3 and 12
number of successes = 3, number of trials = 12, p-value = 0.073
alternative hypothesis: true probability of success is less than 0.5
90 percent confidence interval:
 0.0000000 0.4752663
sample estimates:
probability of success
                  0.25
```

$P-$ 值 $=0.073<0.10$, 间侧区间估计为 $[0,0.475]$, 因此拒绝原假设，认为喜欢咖啡的人超过喜欢奶茶的人.

如果显著性水平定在 $\alpha=0.05$ 时，则不能拒绝原假设，只能认为喜欢咖啡和奶茶的人一样多.

一般来说，符号检验比参数统计 t 检验法的效能低，特别是正、负符号所代表的差额的绝对值比较大时，表现的更为明显.

在符号检验法中，只计算符号的个数，而不考虑每个符号差中所包含的绝对值的大小. 为了弥补这一缺点，所以在非参数统计中还要使用其他的检验方法.

5.3.5 秩统计量

前面介绍了符号检验，下面介绍另一中检验方法 — 秩检验. 在介绍秩检验之前，先介绍与秩检验有关的概念 — 秩统计量 (rank statistics).

秩统计量是在非参数检验中有广泛应用的统计量, 它的一个重要的特性是分布无关性 (distribution-freeness).

定义 5.2 *设 $X_1,X_2,\cdots,X_n$ 为一组样本 (不必取自同一总体), 将 $X_1,X_2,\cdots,X_n$ 从小到大排成一列，用 R_i 记为 X_i 在上述排列中的位置号， $i=1,2,\cdots,n$. 称*

$R_1, R_2, \cdots, R_n$ 为样本 $X_1, X_2, \cdots, X_n$ 产生的秩统计量 *(rank statistics).*

例 5.21 有下列一组样本

x_1	x_2	x_3	x_4	x_5
1.2	0.8	-3.1	2.0	1.2

解: 由此产生的秩统计量 R 为

R_1	R_2	R_3	R_4	R_5
3	2	1	5	4

注意：在上述数据中 $x_1 = x_5$, 这时就按自然顺序将 x_1 排在 x_5 前面.

在 R 软件中，函数 `rank()` 可以计算秩统计量. 如上面的例子,

```
> x<-c(1.2, 0.8, -3.1, 2.0, 1.2)
> rank(x)
[1] 3.5 2.0 1.0 5.0 3.5
```

这里并不象人为排序那样，第一次出现的排在前面，而是同等处理，其顺序均为 3.5. 这种情况在计算统计量时，有时程序会给出警告. 如果希望得到人为规定的排列次序，将第二次出现的值 (x_5) 增加一个很小的值. 如

```
> x<-c(1.2, 0.8, -3.1, 2.0, 1.2+1e-5)
> rank(x)
[1] 3 2 1 5 4
```

这与人工计算的结果相同.

显然, 若样本 $X_1, X_2, \cdots, X_n$ 是取自连续分布总体的独立同分布样本, 则统计量 $R_1, R_2, \cdots, R_n$ 的分布是对称等概率的, 即对 $1, 2, \cdots, n$ 的任一排列 $i_1, i_2, \cdots, i_n$ 有

$$P\{R_1 = i_1, R_2 = i_2, \cdots, R_n = i_n\} = \frac{1}{n!}, \tag{5.21}$$

这时, $R_1, R_2, \cdots, R_n$ 的分布与总体分布无关.

5.3.6 秩相关检验

秩相关检验是秩检验的一个重要应用. 在第三章，我们介绍了 Pearson 相关检验，它实际应用在正态分布总体的数据，这里介绍的秩相关检验并不要求所检验的数据来自正态分布的总体.

1. Spearman 秩相关检验

设 $(X_1,Y_1),(X_2,Y_2),\cdots,(X_n,Y_n)$ 为取自某个二元总体的独立样本，要检验变量 X 与变量 Y 是否相关. 通常以 "X 与 Y 相互独立 (不相关)" 为原假设，"X 与 Y 相关" 为备择假设.

设 $r_1,r_2,\cdots,r_n$ 为由 $X_1,X_2,\cdots,X_n$ 产生的秩统计量， $R_1,R_2,\cdots,R_n$ 为由 $Y_1,Y_2,\cdots,Y_n$ 产生的秩统计量，则有

$$\begin{aligned}\overline{r}=\frac{1}{n}\sum_{i=1}^{n}r_i &= \frac{n+1}{2} = \overline{R}=\frac{1}{n}\sum_{i=1}^{n}R_i,\\ \frac{1}{n}\sum_{i=1}^{n}(r_i-\overline{r})^2 &= \frac{n^2-1}{12} = \frac{1}{n}\sum_{i=1}^{n}(R_i-\overline{R})^2.\end{aligned}$$

定义 5.3 称

$$r_s=\left[\frac{1}{n}\sum_{i=1}^{n}r_iR_i-\left(\frac{n+1}{2}\right)^2\right]\Big/\left(\frac{n^2-1}{12}\right)$$

为 *Spearman*（斯皮尔曼）秩相关系数.

当 X 与 Y 相互独立时， $(r_1,r_2,\cdots,r_n)$ 与 $(R_1,R_2,\cdots,R_n)$ 相互独立时，$E(r_s)=0$. 当 X 与 Y 正相关时， r_s 倾向于取正值；当 X 与 Y 负相关时， r_s 倾向于取负值. 这样就可以得用 r_s 的分布来检验 X 与 Y 是否独立.

可以证明：当 n 较大时， $\sqrt{n-1}\,r_s$ 的近似分布为 $N(0,1)$. 由此可以构造拒绝域和计算相应的 $P-$ 值，当 $P-$ 值小于某一显著性水平 α 时，则拒绝原假设. 我们可以根据问题构造单边检验或双边检验.

R 软件中的检验函数 `cor.test()` 可以进行 Spearman 秩相关检验，其使用方法为

```
cor.test(x, y,
         alternative = c("two.sided", "less", "greater"),
         method = "spearman", conf.level = 0.95, ...)
```

例 5.22 一项有六个人参加表演的竞赛，有两人进行评定，评定结果用表 *5.12* 所示， 试用 *Spearman* 秩相关检验方法检验这两个评定员对等级评定有无相关关系.

解：输入数据，作检验

表 5.12: 两位评判者的评定成绩

参加者编号	1	2	3	4	5	6
甲的打分 (X)	1	2	3	4	5	6
乙的打分 (Y)	6	5	4	3	2	1

```
> x<-c(1,2,3,4,5,6); y<-c(6,5,4,3,2,1)
> cor.test(x, y, method = "spearman")
        Spearman's rank correlation rho
data:  x and y
S = 70, p-value = 0.002778
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
 -1
```

由于计算出的 $P-$ 值小于 0.05, 因此拒绝原假设，认为变量 X 与 Y 相关. 事实上，由于计算出的 $r_s = -1$, 表示这两个量是完全负相关，即两人的结论有关系，但完全相反.

2. Kendall 相关检验

这里从另一个观点来看相关问题. 同样考虑原假设 H_0 : 变量 X 与 Y 不相关，和三个备择假设

$$H_1 : \text{正或负相关} \quad (\text{或者}) \quad \text{正相关} \quad (\text{或者}) \quad \text{负相关}$$

引进协同的概念. 如果乘积 $(X_j - X_i)(Y_j - Y_i) > 0$, 则称对子 (X_i, Y_i) 及 (X_j, Y_j) 是协同的 (concordant) 或者说，它们有同样的倾向. 反之，如果乘积 $(X_j - X_i)(Y_j - Y_i) < 0$, 则称该对子是不协同的 (disconcordant) . 令

$$\Psi(X_i, X_j, Y_i, Y_j) = \begin{cases} 1, & \text{如果 } (X_j - X_i)(Y_j - Y_i) > 0, \\ 0, & \text{如果 } (X_j - X_i)(Y_j - Y_i) = 0, \\ -1, & \text{如果 } (X_j - X_i)(Y_j - Y_i) < 0. \end{cases} \tag{5.22}$$

定义 Kendall (肯达尔) τ 相关系数

$$\widehat{\tau} = \sum_{1\le i<j\le n} \Psi(X_i, X_j, Y_i, Y_j) = \frac{K}{C_n^2} = \frac{n_d - n_c}{C_n^2}, \tag{5.23}$$

其中 n_c 是协同对子的数目， n_d 是不协同对子的数目. 显然,

$$K \equiv \sum \Psi = n_c - n_d = 2n_c - C_n^2. \tag{5.24}$$

上面定义的 $\widehat{\tau}$ 为概率差

$$\tau = P\{(X_j - X_i)(Y_j - Y_i) > 0\} - P\{(X_j - X_i)(Y_j - Y_i) < 0\}$$

的一个估计. 容易看出， $-1 \le \widehat{\tau} \le 1$. 事实上，当所有对子都是协同的，则 $K = C_n^2$, 此时， $\widehat{\tau} = 1$. 当所有对子都是不协同的，则 $K = -C_n^2$, 此时， $\widehat{\tau} = -1$.

设 $r_1, r_2, \cdots, r_n$ 为由 $X_1, X_2, \cdots, X_n$ 产生的秩统计量， $R_1, R_2, \cdots, R_n$ 为由 $Y_1, Y_2, \cdots, Y_n$ 产生的秩统计量，可以证明

$$K = \sum_{1\le i<j\le n} \operatorname{sign}(r_i - r_j) \cdot \operatorname{sign}(R_i - R_j). \tag{5.25}$$

结合式 (5.25) 和式 (5.23), 可以计算出估计值 $\widehat{\tau}$, 这样就可以利用 $\widehat{\tau}$ 值作检验. 当 $\widehat{\tau}$ 接近于 0 时，表示两变量独立；当 $\widehat{\tau}$ 大于某一值时，表示两变量相关 (正数表示正相关，负数表示负相关).

在 R 软件中， Kendall 相关检验仍有函数 `cor.test()` 计算，其计算方法与 Spearman 秩相关检验相同，只需将参数 `method` 改成 `method = "kendall"`.

例 5.23 *某幼儿园对 9 对双胞胎的智力进行检验，并按百分制打分. 现将资料如表 5.13 所示. 试用 Kendall 相关检验方法检验双胞胎的智力是否相关.*

表 5.13: 9 对双胞胎的得分情况

双胞胎对的编号	1	2	3	4	5	6	7	8	9
先出生的儿童 (X)	86	77	68	91	70	71	85	87	63
后出生的儿童 (Y)	88	76	64	96	65	80	81	72	60

解: 输入数据，作检验

```
> X<-c(86, 77, 68, 91, 70, 71, 85, 87, 63)
> Y<-c(88, 76, 64, 96, 65, 80, 81, 72, 60)
> cor.test(X, Y, method = "kendall")
        Kendall's rank correlation tau
data:  X and Y
T = 31, p-value = 0.005886
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
0.7222222
```

$P-$ 值小于 0.05, 拒绝原假设，认为双胞胎的智力是相关的，而且是正相关的.

5.3.7 Wilcoxon 秩检验

1. 对来自一个总体样本的检验

符号检验利用了观测值和原假设的中心位置之差的符号来进行检验，但是它并没有利用这些差的大小 (体现于差的绝对值的大小) 所包含的信息，不同的符号代表了中心位置的哪一边，而差的绝对值的秩的大小代表距中心位置的远近. 如果将两者结合起来，自然比仅仅利用符号更有效. 这也是下面要介绍的 Wilcoxon(威尔科克逊) 符号秩检验 (Wilcoxon signed-rank test) 的宗旨.

为了弥补符号检验法之不足, 在这里将介绍一种在一定程度上考虑到样本观察值与总体中位数之间的差额，即 $|x_i - M_0|$(其中 $i = 1, 2, \cdots, n$) 的大小的检验方法. 在这里假定： (1) 总体分布是连续的； (2) 总体对其中位数是对称的. 这样，将以上 $|x_i - M_0|$ 得到的差额，按递增次序排列，并报据差额的次序给出相应的秩次 R_i, 如差额绝对值最小者给以秩次 1, 次小者给以秩次 $2, \cdots, \cdots$, 最大值给以秩次 n. 再按 $x_i - M_0 > 0$ 为正秩次， $x_i - M_0 < 0$ 为负秩次. 然后按照正秩次和进行检验，这就是秩序和检验. 这种方法首先由 Wilcoxon 提出的，所以称为 Wilcoxon 符号秩检验.

Wilcoxon 检验不仅考虑到每个观察值比总体中位数 M_0 大还是小，而且在一定程度上也考虑了大多少，小多少. 在进行检验时，如果观察值与总体中位数的差额的绝对值相等时, 就要用平均秩次来代替. 例如， $|x_i - M_0| = |x_j - M_0| =$

$|x_k - M_0|$, 首先，给以相应的秩次为 4 、 5 、 6, 其平均值为 5 (R 软件以平均值定义相同值的秩次，三个数据的秩次均是 5). 此外，如果 $x_i - M_0 = 0$, 就将 x_i 从观察数据中去掉.

如果原观察值的数目为 n', 减去差额为 0 的观察数据后，其样本数为 n. 用 $R_i^{(+)}$ 表示正秩次， W 表示正秩次的和，则 Wilcoxon 统计量为

$$W = \sum_{i=1}^{n} R_i^{(+)}. \tag{5.26}$$

因为 n 个整数 $1, 2, \cdots, n$ 的总和用 $\dfrac{n(n+1)}{2}$ 计算，而正秩次总和可以在区间 $\left(0, \dfrac{n(n+1)}{2}\right)$ 内变动，如果观察值来自中位数为 M_0 的某个总体的假设为真，那么 Wilcoxon 检验统计量的取值将是秩次和的平均数，即 $\mu_W = \dfrac{n(n+1)}{4}$ 的左右变动. 如果该假设不成立，则 W 的取值将向秩次和的两头的数值靠近. 这样，在一定的显著性水平，便可进行检验了.

R 软件中的 `wilcox.tets()` 函数可以作 Wilcoxon 符号秩检验，其基本格式为：

```
wilcox.test(x, y = NULL,
    alternative = c("two.sided", "less", "greater"),
    mu = 0, paired = FALSE, exact = NULL, correct = TRUE,
    conf.int = FALSE, conf.level = 0.95, ...)
```

其中 `x,y` 是观察数据构成的数据向量. `alternative` 是备择假设，有单侧检验和双侧检验. `mu` 待检参数，如中位数 M_0. `paired` 是逻辑变量，说明变量 `x`，`y` 是否为成对数据. `exact` 是逻辑变量，说明是否精确计算 $P-$ 值，当样本量较小时，此参数起作用，当样本量较大时，软件采用正态分布近似计算 $P-$ 值. `correct` 是逻辑变量，说明是否对 $P-$ 值的计算采用连续性修正. `conf.int` 是逻辑变量，说明是否给出相应的置信区间.

例 5.24 假定某电池厂宣称该厂生产的某种型号电池寿命的中位数为 *140* 安培小时. 为了检验该厂生产的电池是否符合其规定的标准，现从新近生产的一批电池中抽取 *20* 个随机样本，并对这 *20* 个电池的寿命进行了测试，其结果如下 *(*单位：安培小时*)*：

```
137.0 140.0 138.3 139.0 144.3 139.1 141.7 137.3 133.5 138.2
```

141.1 139.2 136.5 136.5 135.6 138.0 140.9 140.6 136.3 134.1

试用 *Wilcoxon* 符号秩检验分析该厂生产的电池是否符合其标准.

解: 根据题意作如下假设:

$$H_0: \text{电池中位数 } M \geq 140 \text{ 安培小时;}$$
$$H_1: \text{电池中位数 } M < 140 \text{ 安培小时.}$$

输入数据, 调用 `wilcox.test()` 函数,

```
> X<-scan()
1: 137.0 140.0 138.3 139.0 144.3 139.1 141.7 137.3 133.5 138.2
11: 141.1 139.2 136.5 136.5 135.6 138.0 140.9 140.6 136.3 134.1
21:
Read 20 items
> wilcox.test(X, mu=140, alternative="less",
            exact=FALSE, correct=FALSE, conf.int=TRUE)
        Wilcoxon signed rank test
data:  X
V = 34, p-value = 0.007034
alternative hypothesis: true mu is less than 140
95 percent confidence interval:
     -Inf 139.2000
sample estimates:
(pseudo)median
      138.2000
```

这里 `V = 34` 是 Wilcoxon 统计量, $P-$ 值 $0.007034 < 0.05$, 拒绝原假设, 即中位达不到 140 安培小时. 从相应的区间估计也能得到相应的结论.

上面介绍了用 Wilcoxon 符号秩检验方法检验一个样本是否来自某个总体的内容. 同样, 这个方法也可用于成对样本的检验, 从而说明两个总体是否存在显著差异.

例 5.25 为了检验一种新的复合肥和原来使用的肥料相比是否显著地提高了小麦的产量, 在一个农场中选择了 *10* 块田地, 每块等分为两部分, 其中任指定一部分使用新的复合肥料, 另一部分使用原肥料. 小麦成熟后称得各部分小麦产量如表 *5.14* 所示. 试用 *Wilcoxon* 符号检验法检验新复合肥是否会显著提高小麦的

表 5.14: 使用不同肥料情况下小麦的产量 (单位：千克)

田　块	1	2	3	4	5	6	7	8	9	10
新复合肥	459	367	303	392	310	342	421	446	430	412
原肥料	414	306	321	443	281	301	353	391	405	390

产量，并与符号检验作比较 $(\alpha = 0.05)$.

解: 根据题意作如下假设：

H_0 : 新复合肥的产量与原肥料的产量相同,

H_1 : 新复合肥的产量高于原肥料的产量.

输入数据，调用 wilcox.test() 函数,

```
> x<-c(459, 367, 303, 392, 310, 342, 421, 446, 430, 412)
> y<-c(414, 306, 321, 443, 281, 301, 353, 391, 405, 390)
> wilcox.test(x, y, alternative = "greater", paired = TRUE)
        Wilcoxon signed rank test
data:  x and y
V = 47, p-value = 0.02441
alternative hypothesis: true mu is greater than 0
```

$P-$ 值 $0.02441 < 0.05$, 拒绝原假设，即新复合肥能够显著提高小麦的产量.

用下述命令

```
> wilcox.test(x-y, alternative = "greater")
```

具有相同的效果.

如符号检验计算

```
> binom.test(sum(x>y), length(x), alternative = "greater")
        Exact binomial test
data:  sum(x > y) and length(x)
number of successes = 8, number of trials = 10, p-value = 0.05469
alternative hypothesis: true probability of success is greater than 0.5
95 percent confidence interval:
 0.4930987 1.0000000
sample estimates:
```

```
probability of success
                  0.8
```

$P-$ 值 $0.05469 > 0.05$, 无法拒绝原假设. 此结果表明, 在 $\alpha = 0.05$ 的水平下, 就所给数据而言, 符号检验还不足以区分两种肥料对提高小麦的产量产生差异.

比较两个计算结果, 可以发现, Wilcoxon 符号检验比符号检验在探测差异性方面更有效.

2. 非成对样本的秩次和检验

假定两个非成对样本的观察值为 $X_1, X_2, \cdots, X_{n_1}$ 和 $Y_1, Y_2, \cdots, Y_{n_2}$, 其样本容量分别为 n_1 和 n_2. 现要检验两个随机样本来自两个总体的中位数是否相等 (如果中位数相等, 则认为两个总体无差异).

将样本的观察值排在一起, $X_1, X_2, \cdots, X_{n_1}, Y_1, Y_2, \cdots, Y_{n_2}$, 仍设 $r_1, r_2, \cdots, r_{n_1}$ 为由 $X_1, X_2, \cdots, X_{n_1}$ 产生的秩统计量, $R_1, R_2, \cdots, R_{n_2}$ 为由 $Y_1, Y_2, \cdots, Y_{n_2}$ 产生的秩统计量, 则 Wilcoxon-Mann-Whitney 统计量定义为

$$U = n_1 n_2 + \frac{n_2(n_2+1)}{2} - \sum_{i=1}^{n_2} R_i. \tag{5.27}$$

类似单一总体的 Wilcoxon 符号检验一样, 可以通过统计量 U 进行检验, 该检验称为 Wilcoxon 秩和检验.

R 软件中, 仍然是用 `wilcox.test()` 完成 Wilcoxon 秩和检验.

例 5.26 *今测得 10 名非铅作业工人和 7 名铅作业工人的血铅值, 如表 5.15 所示. 试用 Wilcoxon 秩和检验分析两组工人血铅值有无差异.*

表 5.15: 两组工人的血铅值 (单位: 10^{-6}mmol/L)

非铅作业组	24	26	29	34	43	58	63	72	87	101
铅作业组	82	87	97	121	164	208	213			

解: 根据题意作如下假设:

H_0 : 两组工人血铅无差异, H_1 : 铅作业组血铅高于非铅作业组.

输入数据, 调用 `wilcox.test()` 函数,

```
> x<-c(24, 26, 29, 34, 43, 58, 63, 72, 87, 101)
> y<-c(82, 87, 97, 121, 164, 208, 213)
```

```
#### 不采用连续修正
> wilcox.test(x,y,alternative="less",exact=FALSE,correct=FALSE)
        Wilcoxon rank sum test
data:  x and y
W = 4.5, p-value = 0.001449
alternative hypothesis: true mu is less than 0
#### 采用连续修正
> wilcox.test(x, y, alternative="less", exact=FALSE)
        Wilcoxon rank sum test with continuity correction
data:  x and y
W = 4.5, p-value = 0.001698
alternative hypothesis: true mu is less than 0
```

W = 4.5 是 Wilcoxon-Mann-Whitney 统计量. 在上述计算中, 无论采用连续修正, 要还是不采用连续修正, 其 $P-$ 值均小于 0.05, 因此拒绝原假设, 即铅作业组工人血铅值高于非铅作业组的工人.

例 5.27 为了了解新的数学教学方法的效果是否比原来方法的效果有所提高, 从水平相当的 *10* 名学生中随机地各选 *5* 名接受新方法和原方法的教学试验. 充分长一段时间后, 由专家通过各种方式 *(*如考试提问等*)* 对 *10* 名学生的数学能力予以综合评估 *(*为公证起见, 假定专家对各个学生属于哪一组并不知道*)*, 并按其数学能力由弱到强排序, 结果如表 *5.16* 所示. 对 $\alpha = 0.05$, 检验新方法是否比

表 5.16: 学生数学能力排序结果 (1)

新方法			3		5		7		9	10
原方法	1	2		4		6		8		

原方法显著地提高了教学效果. 若排序结果如表 *5.17* 所示, 情况又如何?

表 5.17: 学生数学能力排序结果 (2)

新方法				4		6	7		9	10
原方法	1	2	3		5			8		

解: 因为 Wilcoxon 秩和检验本质只需排出样本的秩次，而且题目中的数据本身就是一个排序，因此可直接使用.

```
> x<-c(3, 5, 7, 9, 10); y<-c(1, 2, 4, 6, 8)
> wilcox.test(x, y, alternative="greater")
        Wilcoxon rank sum test
data:  x and y
W = 19, p-value = 0.1111
alternative hypothesis: true mu is greater than 0
```

$P-$ 值 $= 0.1111 > 0.05$, 无法拒绝原假设，即认为新的教学效果并不显著优于原方法.

对于第二种情况,

```
> X<-c(4, 6, 7, 9, 10); Y<-c(1, 2, 3, 5, 8)
> wilcox.test(X, Y, alternative="greater")
        Wilcoxon rank sum test
data:  X and Y
W = 21, p-value = 0.04762
alternative hypothesis: true mu is greater than 0
```

$P-$ 值 $= 0.04762 < 0.05$, 拒绝原假设，即认为新的教学效果显著优于原方法.

例 5.28 某医院用某种药物治疗两型慢性支气管炎患者共 *216* 例, 疗效由表 *5.18* 所示. 试分析该药物对两型慢性支气管炎的治疗是否相同.

表 5.18: 某种药物治疗两型慢性支气管炎疗效结果

疗效	控制	显效	进步	无效
单纯型	62	41	14	11
喘息型	20	37	16	15

解: 我们想象各病人的疗效用 4 个不同的值表示 (1 表示最好, 4 表示最差), 这样就可以为这 216 名病人排序，因此，可用 Wilcoxon 秩和检验来分析问题.

```
> x<-rep(1:4, c(62, 41, 14,11)); y<-rep(1:4, c(20, 37, 16, 15))
> wilcox.test(x, y, exact=FALSE)
```

```
        Wilcoxon rank sum test with continuity correction
data:  x and y
W = 3994, p-value = 0.0001242
alternative hypothesis: true mu is not equal to 0
```

$P-$ 值 $= 0.0001242 < 0.05$, 拒绝原假设，即认为该药物对两型慢性支气管炎的治疗是不相同的. 因为数据有结点存在，故无法精确计算 $P-$ 值，其参数为 `exact=FALSE`.

本节介绍了一些重要的非参数检验方法， R 软件还提供了另外一些非参数检验方法，这里就不一一列举了. 因为掌握了已有的方法，再学习其他方法就不困难了，使用时可通过在线帮助了解其基本的使用方法.

习题五

5.1 正常男子血小板计数均值为 $225 \times 10^9/L$, 今测得 *20* 名男性油漆作业工人的血小板计数值 *(*单位： $10^9/L$*)*

220 188 162 230 145 160 238 188 247 113
126 245 164 231 256 183 190 158 224 175

问油漆工人的血小板计数与正常成年男子有无差异？

5.2 已知某种灯泡寿命服从正态分布，在某星期所生产的该灯泡中随机抽取 *10* 只，测得其寿命 *(*单位：小时*)* 为

1067 919 1196 785 1126 936 918 1156 920 948

求这个星期生产出的灯泡能使用 *1000* 小时以上的概率.

5.3 为研究某铁剂治疗和饮食治疗营养性缺铁性贫血的效果，将 *16* 名患者按年龄、体重、病程和病情相近的原则配成 *8* 对，分别使用饮食疗法和补充铁剂治疗的方法， *3* 个月后测得两种患者血红蛋白如表 *5.19* 所示， 问两种方法治疗后的

表 5.19: 铁剂和饮食两种方法治疗后患者血红蛋白值 (g/L)

铁剂治疗组	113	120	138	120	100	118	138	123
饮食治疗组	138	116	125	136	110	132	130	110

患者血红蛋白有无差异？

5.4 为研究国产四类新药阿卡波糖胶囊效果，某医院用 *40* 名 *II* 型糖尿病病人进行同期随机对照实验. 试验者将这些病人随机等分到试验组 *(*阿卡波糖胶囊组*)* 和对照组 *(*拜唐苹胶囊组*),* 分别测得试验开始前和 *8* 周后空腹血糖，算得空腹血糖下降值，如表 *5.20* 所示. 能否认为国产四类新药阿卡波糖胶囊与拜唐苹胶囊

表 5.20: 试验组与对照组空腹腔血糖下降值 (mmol/L)

试验组	-0.70	-5.60	2.00	2.80	0.70	3.50	4.00	5.80	7.10	-0.50
$(n_1 = 20)$	2.50	-1.60	1.70	3.00	0.40	4.50	4.60	2.50	6.00	-1.40
对照组	3.70	6.50	5.00	5.20	0.80	0.20	0.60	3.40	6.60	-1.10
$(n_2 = 20)$	6.00	3.80	2.00	1.60	2.00	2.20	1.20	3.10	1.70	-2.00

对空腹血糖的降糖效果不同？

(1) 检验试验组和对照组的的数据是否来自正态分布，采用正态性 W 检验方法 *(*见第三章*)* 、 *Kolmogorov-Smirnov* 检验方法和 *Pearson* 拟合优度 χ^2 检验;

(2) 用 $t-$ 检验两组数据均值是否有差异，分别用方差相同模型、方差不同模型和成对 $t-$ 检验模型;

(3) 检验试验组与对照组的方差是否相同.

5.5 为研究某种新药对抗凝血酶活力的影响，随机安排新药组病人 *12* 例，对照组病人 *10* 例，分别测定其抗凝血酶活力 *(*单位： mm^3*),* 其结果如下：

新药组： *126 125 136 128 123 138 142 116 110 108 115 140*

对照组： *162 172 177 170 175 152 157 159 160 162*

试分析新药组和对照组病人的抗凝血酶活力有无差别 $(\alpha = 0.05)$.

(1) 检验两组数据是否服从正态分布;

(2) 检验两组样本方差是否相同;

(3) 选择最合适的检验方法检验新药组和对照组病人的抗凝血酶活力有无差别.

5.6 一项调查显示某城市老年人口比重为 *14.7%.* 该市老年研究协会为了检验

该项调查是否可靠，随机抽选了 *400* 名居民，发现其中有 *57* 人是老年人. 问调查结果是否支持该市老年人口比重为 *14.7%* 的看法 ($\alpha = 0.05$).

5.7 作性别控制试验，经某种处理后，共是雏鸡 *328* 只，其中公雏 *150* 只，母雏 *178* 只，试问这种处理能否增加母雏的比例？ *(*性别比应为 $1:1$*).*

5.8 *Mendel* 用豌豆的两对相对性状进行杂交实验，黄色圆滑种子与绿色皱缩种的豌豆杂交后，第二代根据自由组合规律，理论分离比为

$$\text{黄圆}:\text{黄皱}:\text{绿圆}:\text{绿皱} = \frac{9}{16}:\frac{3}{16}:\frac{3}{16}:\frac{1}{16}$$

实际实验值为：黄圆 *15* 粒，黄皱 *101* 粒，绿圆 *108* 粒，绿皱 *32* 粒，共 *556* 粒，问此结果是否符合自由组合规律？

5.9 观察每分钟进入某商店的人数 X, 任取 *200* 分钟，所得数据如下

顾客人数	*0*	*1*	*2*	*3*	*4*	*5*
频数	*92*	*68*	*28*	*11*	*1*	*0*

试分析，能否认为每分钟顾客数 X 服从 *Poisson* 分布 ($\alpha = 0.1$).

5.10 观察得两样本值如下

I	*2.36*	*3.14*	*7.52*	*3.48*	*2.76*	*5.43*	*6.54*	*7.41*
II	*4.38*	*4.25*	*6.53*	*3.28*	*7.21*	*6.55*		

试分析，两样本是否来自同一总体 ($\alpha = 0.05$).

5.11 为研究分娩过程中使用胎儿电子监测仪对剖腹产率有无影响，对 *5824* 例分娩的经产妇进行回顾性调查，结果如表 *5.21* 所示，试进行分析.

表 5.21: 5824 例经产妇回顾性调查结果

剖腹产	胎儿电子监测仪		合计
	使用	未使用	
是	358	229	587
否	2492	2745	5237
合计	2850	2974	5824

5.12 在高中一年级男生中抽取 *300* 名考察其两个属性： B 是 *1500* 米长跑，C 是每天平均锻炼时间，得到 4×3 列联表，如表 *5.22* 所示. 试对 $\alpha = 0.05$,

表 5.22: 300 名高中学生体育锻炼的考察结果

1500 米长跑记录	锻炼时间			合计
	2 小时以上	1 ～ 2 小时	1 小时以下	
$5''01' \sim 5''30'$	45	12	10	67
$5''31' \sim 6''00'$	46	20	28	94
$6''01' \sim 6''30'$	28	23	30	81
$6''31' \sim 7''00'$	11	12	35	58
合计	130	67	103	300

检验 B 与 C 是否独立.

5.13 为比较两种工艺对产品的质量是否有影响，对其产品进行抽样检查，其结果如表 *5.23* 所示. 试进行分析.

表 5.23: 两种工艺下产品质量的抽查结果

	合格	不合格	合计
工艺一	3	4	7
工艺二	6	4	10
合计	9	8	17

5.14 应用核素法和对比法检测 *147* 例冠心病患者心脏收缩运动的符合情况，其结果如表 *5.24* 所示. 试分析这两种方法测定结果是否相同.

5.15 在某养鱼塘中，根据过去经验，鱼的长度的中位数为 *14.6cm,* 现对鱼塘中鱼的长度进行一次估测，随机地从鱼塘中取出 *10* 条鱼长度如下：

```
13.32 13.06 14.02 11.86 13.58 13.77 13.51 14.42 14.44 15.43
```

将它们作为一个样本进行检验. 试分析，该鱼塘中鱼的长度是在中位数之上，还是在中位数之下.

(1) 用符号检验分析;

表 5.24: 两法检查室壁收缩运动的符合情况

对比法	核素法			合计
	正常	减弱	异常	
正常	58	2	3	63
减弱	1	42	7	50
异常	8	9	17	34
合计	67	53	27	147

(2) 用 *Wilcoxon* 符号秩检验.

5.16 用两种不同的测定方法，测定同一种中草药的有效成分，共重复 *20* 次，得到实验结果如表 *5.25* 所示.

表 5.25: 两种不同的测定方法得到的结果

方法	48.0	33.0	37.5	48.0	42.5	40.0	42.0	36.0	11.3	22.0
A	36.0	27.3	14.2	32.1	52.0	38.0	17.3	20.0	21.0	46.1
方法	37.0	41.0	23.4	17.0	31.5	40.0	31.0	36.0	5.7	11.5
B	21.0	6.1	26.5	21.3	44.5	28.0	22.6	20.0	11.0	22.3

(1) 试用符号检验法检验两测定有无显著差异;

(2) 试用 *Wilcoxon* 符号秩检验法检验两测定有无显著差异;

(3) 试用 *Wilcoxon* 秩和检验法检验两测定有无显著差异;

(4) 对数据作正态性和方差齐性检验，该数据是否作 $t-$ 检验，如果能，请作 $t-$ 检验;

(5) 分析各种的检验方法，试说明哪种检验法效果最好.

5.17 调查某大学学生每周学习时间与得分的平均等级之间的关系，现抽查 *10* 个学生的资料如表下:

学习时间	*24*	*17*	*20*	*41*	*52*	*23*	*46*	*18*	*15*	*29*
学习等级	*8*	*1*	*4*	*7*	*9*	*5*	*10*	*3*	*2*	*6*

其中等级 *10* 表示最好，*1* 表示最差. 试用秩相关检验 *(Spearman* 检验和 *Kendall* 检验*)* 分析学习等级与学习成绩有无关系.

5.18 为比较一种新疗法对某种疾病的治疗效果, 将 *40* 名患者随机地分为两组，每组 *20* 人，一组采用新疗法，另一组用原标准疗法. 经过一段时间的治疗后，对每个患者的疗效作仔细的评估，并划分为差、较差、一般、较好和好五个等级. 两组中处于不同等级的患者人数如表 *5.26* 所示. 试分析，由此结果能否认为新

表 5.26: 不同方法治疗后的结果

等级	差	较差	一般	较好	好
新疗法组	0	1	9	7	3
原疗法组	2	2	11	4	1

方法的疗效显著地优于原疗法 ($\alpha = 0.05$).