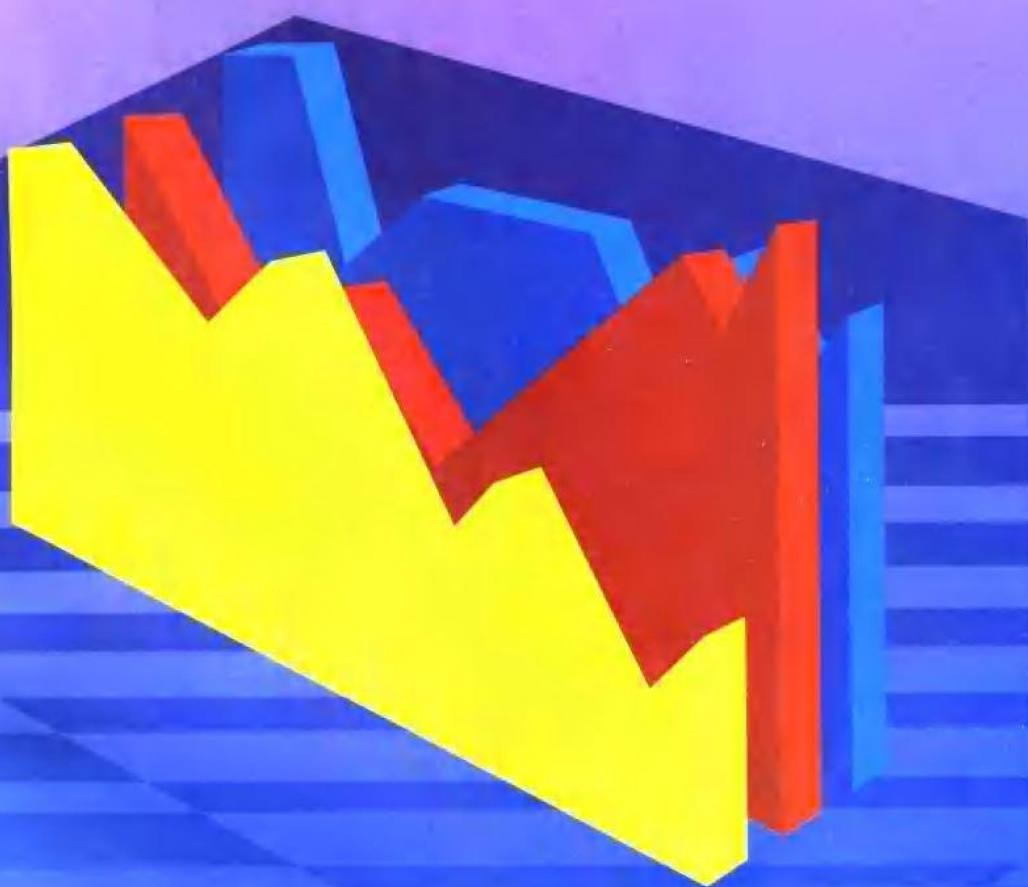


多因素分析与SAS应用

田俊 编著



福建科学技术出版社

内容提要

本书介绍多因素分析及应用统计分析系统软件 SAS 对医学科研工作实践中获取的数据资料进行多因素分析处理的方法与技巧。除了介绍如何应用 SAS 进行多元回归分析、判别分析、聚类分析、主成分分析及因子分析这些常用统计方法之外,本书还介绍了新近发展起来的多元协方差分析、趋势面分析、典型相关分析、对数线性模型、生存分析及成组资料的 LOGISTIC 回归分析、曲线拟合等多因素分析方法的 SAS 应用。全书共分 14 章,每章先以简洁易懂的方式概述统计方法,然后介绍相应的基本 SAS 语句、编程方法和输出结果。每章内容均配以丰富的实例,详细讲解 SAS 程序,并对输出结果加以解释和分析。本书适用于从事医学科研的实际工作者、从事医学统计研究与教学的人员及计算机应用系统开发人员等,也适用于农、林、经济管理等领域相关专业的科研人员。

(闽)新登字 03 号

多因素分析与 SAS 应用

田俊 编著

*

福建科学技术出版社出版、发行

(福州市东水路 76 号)

福建省新华书店经销

福建省科发电脑排版服务公司排版

福州市东南印刷厂印刷

开本 787×1092 毫米 1/16 17 印张 2 插页 426 千字

1997 年 8 月第 1 版

1997 年 8 月第 1 次印刷

印数: 1—1 200

ISBN 7-5335-1178-6/TP·56

定价: 20.00 元

书中如有印装质量问题,可直接向承印厂调换

前 言

在客观世界中，任何事物的形成、变化和发展都受到多种因素的影响和制约，所以多因素分析对于人们正确地认识世界和改造世界有重要的应用意义。医学现象尤其复杂多变。例如疾病的发生，病情的变化就往往受到多种因素的支配，各种病因之间也常存在着千丝万缕的内在联系和相互制约。多因素分析在研究这些问题时有广阔的用武之地，它为研究者提供了科学的方法和现代化的研究手段。

多因素分析的计算过程都比较复杂，且计算量较大，用普通的计算工具很难胜任。即使在一些情况下可以计算，也只能对小规模的实验数据进行分析。而多因素分析方法又恰恰是在对较大规模的研究数据进行分析时才显示出它的科学性、有效性。计算机技术的飞速发展，使多因素分析方法在各学科领域的科研中有了更广阔的应用前景，而优秀的统计分析软件包的问世又推动了科研数据处理向更高水平迈进。美国 SAS 研究所于 20 世纪 60 年代开始研制 SAS 软件包。直到 1985 年才植入微机。它是计算机技术与统计技术相结合的产物，是一个优秀的统计系统。随着对外交流的扩大，SAS 已经引入国内，但由于专业限制及语言文字的障碍，许多医学科研工作者感到 SAS 系统太高深，它的输出结果太复杂，想用而不会用。目前科研工作者最关心和最感困惑的问题有三点：(1) 用何种统计方法处理自己的科研数据；(2) 选定统计方法后，如何使用 SAS 来进行统计分析；(3) 如何理解 SAS 所输出的统计分析结果。作者针对上述问题编著了本书。它将多因素分析与 SAS 软件两个主题紧密联系在一起，从医学科研资料的多因素统计处理方法入手，介绍 SAS 使用方法，使读者能够懂得选用何种统计方法处理自己的数据，并且在选定统计方法时就知道调用 SAS 中哪一个过程来为自己服务。同时，对语句、命令说明所涉及的术语均在方法概述部分提到，使读者能够快速理解程序说明，从而可将程序应用于自己的数据中。在结果输出部分配以详细解释，使读者能够从大量的输出内容中快速找到自己所要的结果。同时对 SAS 输出的结果怎样分析作详细介绍，使读者能够明白由这些结果输出可以得到哪些与研究内容有关的专业结论，从而举一反三，融会贯通。

由于时间仓促和作者水平有限，书中不足与错漏之处在所难免，恳切希望有关专家、实际工作者和有兴趣的读者不吝批评指正。

编著者

目 录

第一章 SAS 系统基本知识	(1)
§ 1.1 SAS 程序	(1)
§ 1.1.1 SAS 程序结构	(1)
§ 1.1.2 SAS 编程基础	(1)
§ 1.1.3 SAS 程序的屏幕编辑	(4)
§ 1.2 SAS 程序的存贮、读取及打印	(7)
§ 1.2.1 SAS 程序的存贮	(7)
§ 1.2.2 SAS 程序的读取	(8)
§ 1.2.3 SAS 程序的打印	(8)
§ 1.3 SAS 数据步	(8)
§ 1.3.1 数据步中所用的语句	(9)
§ 1.3.2 建立 SAS 数据集	(10)
§ 1.3.3 存贮 SAS 数据集	(12)
§ 1.3.4 SAS 数据集读入内存	(12)
§ 1.3.5 数据文件的修改	(13)
§ 1.3.6 数据文件的增加	(15)
§ 1.3.7 对数据文件作删除	(17)
§ 1.3.8 SAS 数据集的连接	(18)
§ 1.3.9 SAS 数据集的合并	(19)
§ 1.3.10 数据排序	(20)
§ 1.4 SAS 数据集与 DBASE 数据库的转换	(21)
§ 1.4.1 DBASE 数据文件转换成 SAS 数据集	(21)
§ 1.4.2 SAS 数据集转换成 DBASE 数据文件	(22)
§ 1.4.3 补充说明	(22)
§ 1.5 外部数据文件转成 SAS 数据集	(22)
第二章 两样本平均值向量之间的比较	(24)
§ 2.1 方法概述	(24)
§ 2.2 基本 SAS 语句	(25)
§ 2.3 实例	(25)
§ 2.4 补充说明	(33)
第三章 多个样本均值向量之间的比较	(34)
§ 3.1 方法概述	(34)
§ 3.2 基本 SAS 语句	(35)
§ 3.3 实例	(36)

§ 3.4 多个样本间的两两比较·····	(39)
第四章 多元协方差分析 ·····	(44)
§ 4.1 多元协方差分析的计算步骤·····	(44)
§ 4.2 SAS 语句编写的程序及运行结果解释 ·····	(47)
第五章 曲线拟合 ·····	(58)
§ 5.1 曲线拟合的用途·····	(58)
§ 5.2 常用的曲线模型·····	(59)
§ 5.3 曲线拟合的一般步骤·····	(63)
§ 5.4 基本 SAS 语句 ·····	(64)
§ 5.5 实例·····	(65)
§ 5.5.1 多项式曲线的拟合·····	(65)
§ 5.5.2 指数曲线的拟合·····	(71)
§ 5.5.3 双曲线的拟合·····	(78)
§ 5.5.4 Logistic 曲线的拟合 ·····	(82)
§ 5.5.5 催化模型的拟合·····	(84)
§ 5.6 补充说明·····	(89)
第六章 典型相关分析 ·····	(92)
§ 6.1 典型相关分析方法概述·····	(92)
§ 6.2 典型相关分析计算步骤·····	(93)
§ 6.3 SAS 语句 ·····	(96)
§ 6.4 实例·····	(97)
第七章 趋势面分析 ·····	(108)
§ 7.1 方法概述 ·····	(108)
§ 7.2 基本 SAS 语句 ·····	(110)
§ 7.3 实例 ·····	(111)
§ 7.4 趋势图的绘制 ·····	(116)
第八章 多元回归 ·····	(122)
§ 8.1 多元回归的概念及分析实例 ·····	(122)
§ 8.1.1 多元回归的一般解法 ·····	(122)
§ 8.1.2 假设检验 ·····	(123)
§ 8.1.3 区间估计 ·····	(124)
§ 8.1.4 回归分析中的共线性诊断与残差分析 ·····	(124)
§ 8.1.5 实例 ·····	(126)
§ 8.2 逐步回归 ·····	(129)
§ 8.2.1 逐步回归的概念 ·····	(129)

§ 8.2.2 逐步回归的计算步骤	(130)
§ 8.2.3 实例	(131)
§ 8.3 线性模型应用于排除混杂因素的影响	(134)
§ 8.3.1 方法	(134)
§ 8.3.2 实例	(135)
§ 8.4 岭回归分析	(136)
§ 8.4.1 方法	(137)
§ 8.4.2 应用 SAS 估计岭回归系数	(137)
§ 8.4.3 实例	(139)
第九章 判别分析	(144)
§ 9.1 Bayes 多类判别	(144)
§ 9.2 逐步判别分析	(147)
§ 9.3 基本 SAS 语句	(148)
§ 9.4 实例	(150)
第十章 聚类分析	(162)
§ 10.1 样品聚类法	(162)
§ 10.1.1 聚类分析的原理	(162)
§ 10.1.2 SAS 基本语句	(166)
§ 10.1.3 聚类图	(168)
§ 10.1.4 实例	(168)
§ 10.2 指标聚类法	(173)
§ 10.2.1 指标聚类常用统计量	(174)
§ 10.2.2 指标聚类过程	(174)
§ 10.2.3 SAS 基本语句	(175)
§ 10.2.4 实例	(176)
第十一章 因子分析	(180)
§ 11.1 主成分分析	(180)
§ 11.1.1 主成分分析的思想	(180)
§ 11.1.2 主成分的性质	(182)
§ 11.1.3 主成分的选取	(183)
§ 11.1.4 主成分分析的应用	(183)
§ 11.1.5 基本 SAS 语句	(186)
§ 11.1.6 实例	(187)
§ 11.2 因子分析	(190)
§ 11.2.1 因子分析的模型	(190)
§ 11.2.2 因子分析的步骤	(190)
§ 11.2.3 因子分析应注意的问题	(192)

§ 11.2.4	因子分析的应用	(193)
§ 11.2.5	基本 SAS 语句	(194)
§ 11.2.6	实例	(195)
第十二章	Logistic 回归	(199)
§ 12.1	Logistic 回归模型的基本原理	(199)
§ 12.2	多因素 Logistic 回归模型的参数估计	(200)
§ 12.3	SAS 基本语句	(202)
§ 12.4	运行 SAS 程序后 RR 的计算方法	(203)
§ 12.5	实例	(205)
§ 12.6	多分类的 Logistic 模型	(225)
§ 12.6.1	模型	(225)
§ 12.6.2	判别系数的估计	(226)
§ 12.6.3	分类规则	(226)
§ 12.6.4	实例	(226)
§ 12.6.5	补充说明	(233)
第十三章	生存分析	(234)
§ 13.1	生存分析的概念	(234)
§ 13.1.1	生存时间	(234)
§ 13.1.2	终检数据	(234)
§ 13.1.3	生存时间 T 的三种主要分布	(235)
§ 13.1.4	参数回归模型	(236)
§ 13.2	全参数回归模型	(237)
§ 13.2.1	指数分布	(237)
§ 13.2.2	威布尔 (Weibull) 分布	(238)
§ 13.3	SAS 语句	(239)
§ 13.4	实例	(240)
§ 13.5	补充说明	(249)
第十四章	对数线性模型	(250)
§ 14.1	对数线性模型的原理	(250)
§ 14.1.1	二维列联表模型	(250)
§ 14.1.2	三维列联表模型	(251)
§ 14.1.3	参数估计	(253)
§ 14.2	模型的选择	(253)
§ 14.3	基本 SAS 语句	(255)
§ 14.4	实例	(256)
§ 14.5	补充说明	(266)

第一章 SAS 系统基本知识

SAS 的英文全名是 Statistical Analysis System，译成中文就是统计分析系统。SAS 系统具有信息存储简单，语言编程能力强，可对数据作连续处理，统计分析方法丰富且使用简单等特点。近年来在医学、农学、社会科学等领域正越来越广泛地使用 SAS 来处理科研数据。科研人员学习和使用 SAS，将会提高科研信息的处理水平。

§ 1.1 SAS 程序

§ 1.1.1 SAS 程序结构

在 SAS 系统中，对数据的处理大体上分成两步，它们分别为数据步和过程步。数据步将数据读入 SAS 系统来建立 SAS 数据集。数据步以 DATA 语句开始，以 RUN 语句结束。过程步调用各种已经编好的过程来处理、分析数据集中的数据。过程步以 PROC 语句开始，以 RUN 语句结束。每个语句的后面都要以“;”结束。例如，用 DATA 步建立一个 SAS 数据集，存放 4 个观察对象的身高、体重值，并用 SAS 中的 MEANS 过程计算 4 个观察对象的身高、体重均数。可以写以下 SAS 程序：

```
DATA Exam1;                                158.2  53.6
INPUT height weight;                        ;
CARDS;                                     RUN;
156.3  47.1                                PROC MEANS;
172.4  61.5                                RUN;
169.2  64.5
```

上述 SAS 程序中，从第一句 DATA 开始到第一个 RUN 语句为数据步，最后两句为过程步。

当有多个数据步或多个过程步或数据步与过程步混合时，由于后一个 DATA 或 PROC 语句起到前一步的 RUN 语句的作用，故两步中间的 RUN 语句常可省略。但整个程序必须以 RUN 结尾，否则最后一步（数据步或是过程步）将不能运行。

§ 1.1.2 SAS 编程基础

一、SAS 变量及命名

给定特性的数据值的集合组成了变量。在 SAS 数据集中，每一行数据是由各个变量的具

体值组成。因此，在数据集中每一列数据是一个变量。

SAS 变量名最多含 8 个字符，它可以是数字或字母，但变量名称的开头一个字符必须是字母。空格不能出现在 SAS 变量名中，特殊字符（如 \$，@，#）也不允许在 SAS 变量名中使用。此外，SAS 系统保留了一定的名称作为特殊的变量名，这些名称以下划线开始和结尾。例如 _N_ 和 _ERROR_ 等。

二、SAS 变量特性

SAS 变量有两种类型：数值型和字符型。字符型变量在名后用一个 “\$” 来表示。例如：

```
DATA Exam2;                                M 169.3 64.5
INPUT sex $ height weight;                  F 158.2 53.6
CARDS;                                     ;
F 156.3 47.1                                RUN;
M 172.4 61.5
```

在上述 DATA 步中，sex 是字符型变量，它表示每个观察对象的性别，取值为 ‘F’ 或 ‘M’，而 height 和 weight 是数值型变量。

三、SAS 变量的简化表示

在 SAS 程序中有些变量的形式可以简化表示。例如：

```
DATA Exam3;                                ZhangLi 0 0 19 1
INPUT name $ x1 x1 x3 x4;                    ;
CARDS;                                     RUN;
Wanghai 1 0 23 2
```

上述程序中的 INPUT 语句可以写成：

```
INPUT name $ x1--x4;
```

各种不同的缩写清单如表 1.1 所示。应注意有序号的变量（例如 x1，x2，…，xn）不必全部列出，但这些变量必须是类型相同的。

表 1.1 SAS 变量简写清单

变量名的形式	缩写	代表
形如 x1, x2, …, xn 的带有序号的名称	x1-xn	从 x1 到 xn 的所有变量
形如 height 的名称范围	h-numeric-t	从 h 到 t 的所有数值变量
形如 name 的字符型变量名称范围	n-character-e	从 n 到 e 的所有字符变量
特殊 SAS 名称	_numeric _	所有数值变量
	_character _	所有字符变量
	_all _	所有变量

四、缺项值的表示

当一个 SAS 变量在某些观察对象上未能测到或变量的表达式中出现非法（如分母为零、负数取对数等）时，该变量作为缺项值处理。在 SAS 中用 “.” 表示缺项值。

五、SAS 函数

SAS 函数是一个程序，它对一个或多个参数进行计算后返回一个值。每一个 SAS 函数有一个关键字名。调用函数时，写出函数名和参数：函数名（参数，参数）。当参数多于一个时，

参数之间用逗号分隔。例如：

SUM (x1, x2, x3, x4)

或 SUM (OF x1—x4)

上述两种写法都是正确的。SAS 中的函数分为算术函数、数学函数、三角函数和双曲函数、概率函数、样本统计函数、随机函数、分位数函数、字符函数和日期时间函数。下面介绍常用的几种函数表示方式和意义。

1. 算术函数

ABS (x)	取 x 的绝对数
DIM (A)	定义数组 A 的维数
MAX (x, y, ...)	求 x, y, ... 中的最大值
MIN (x, y, ...)	求 x, y, ... 中的最小值
MOD (x, y)	求 x/y 的余项
SIGN (x)	取 x 的符号, 当 x=0 时 SIGN (x) = 0
SQRT (x)	计算 x 的平方根

2. 数学函数

EXP (x)	求自然对数的底 e 的幂 e^x
LOG (x)	求 x 的自然对数 $\ln x$
LOG2 (x)	求底为 2 的 x 的对数 $\log_2 x$
LOG10 (x)	求底为 10 的 x 的对数 $\lg x$

3. 三角函数、双曲函数

COS (x)	计算 x 的余弦 $\cos x$
SIN (x)	计算 x 的正弦 $\sin x$
TAN (x)	计算 x 的正切 $\tan x$
ARCOS (x)	计算 x 的反余弦 $\cos^{-1}x$
ARSIN (x)	计算 x 的反正弦 $\sin^{-1}x$
ARTAN (x)	计算 x 的反正切 $\text{tg}^{-1}x$

4. 概率函数

PROBBNML (p, n, r)	二项式分布的概率函数
PROBCHI (x, df)	卡方分布的概率函数
PROBF (x, df1, df2)	F 分布概率函数
PROBNORM (x)	标准正态分布的概率函数

5. 样本统计函数

MEAN (x, y, ...)	计算 x, y, ... 的均数
STD (x, y, ...)	计算 x, y, ... 的标准差
SUM (x, y, ...)	计算 x, y, ... 的和

六、SAS 操作符

SAS 操作符是表示作算术计算、比较或者逻辑操作的记号。

1. 算术操作符

算术操作符指出要执行的算术计算。算术操作符是：

*	*	乘方	例如， 5^3 写为： $5 * * 3$
*		乘	例如， 6×6 写为 $6 * 6$
/		除	
+		加	
-		减	

2. 比较操作符

比较操作符用于确定两个数据量间是否存在操作符所定义的关系。如果不存在这种关系，则结果值为 0，否则为 1。

比较操作符是：

= 或 EQ	等于	< 或 LT	小于
^ = 或 NE	不等于	>= 或 GE	大于等于
> 或 GT	大于	<= 或 LE	小于等于

3. 逻辑操作符

逻辑操作符通常用在连接一系列比较表达式中。

& 或 AND “且” 运算符。如果 AND 两端都为真，则 AND 操作的结果是 1。

1 或 OR “或” 运算符。如果 OR 两端至少有一个为真，则 OR 操作的结果为 1。

比较操作符、逻辑操作符经常与 IF 指令配合使用。例如：

```
DATA Exam3;
INPUT x y z;
IF x>y AND x>z THEN c=1;
IF x<y OR x<z THEN c=2;
CARDS;
7 8 5;
RUN;
```

§ 1.1.3 SAS 程序的屏幕编辑

一、SAS 程序的输入、修改和运行

1. 程序输入

例 1.1 某医生将血糖在 6.16—6.72mmol/L 的患者随机分成二组。A 组 8 人，采用饮食

疗法治疗；B组6人，采用运动疗法治疗。治疗一段时间后测定二组患者血糖下降值。结果如下：

A组 0.8568 0.8176 0.6608 0.8288 0.8904 0.7168 0.9464 0.8680

B组 0.7056 0.7168 0.6552 0.7224 0.6328 0.6664

现用t检验来比较两组的血糖下降值是否有差别。

将光标移至程序编辑窗口（PROGRAM EDITOR），在行号区的右边打入如图1.1的程序：

```
PROGRAM EDITOR
Command⇒
00001 data exam4;
00002 do group=1 to 2;
00003 input n;
00004 do i=1 to n;
00005 input x@@; output;
00006 end; end;
00007 drop i;
00008 cards;
00009 8
00010 0.8568 0.8176 0.6608 0.8288 0.8904 0.7168 0.9464 0.8680
00011 6
00012 0.7056 0.7168 0.6552 0.7224 0.6328 0.6664
00013 ;
00014 proc ttest;
00015 class group;
00016 var x;
00017 run;
00018
00019
00020
00021
```

ZOOM

图 1.1 程序

输入程序时，可以使用大写或小写字母混合输入SAS语句，语句中每个单词和数据之间用空格间隔，每个变量之间也用空格间隔，空格数可由用户任意选定。每行输完后按回车键（RETURN），则光标移到下一行的开始处。在程序输入过程中，可以上下左右移动光标去作必要的修改。

SAS语句书写格式自由，可在各行的任何位置开始写程序。一个语句可以连续写在几个行中，也可以一行写几个语句。每个语句的后面要用“;”号结束。

2. 改变窗口大小

在SAS中，可以用ZOOM命令或F7功能键扩大当前窗口，使这个窗口充满整个屏幕。为了消除ZOOM命令的作用，再一次按F7键，屏幕就退回到原来的样子。

3. 运行程序

当程序输入完毕，在命令行（COMMAND⇒处）上键入SUBMIT↵，或在PC机上按功能键F10，则当前窗口中的SAS程序开始运行。与此同时，窗口的所有语句均被清除了，在窗口的右下方出现一个“R”字，告诉用户这个程序步正在运行。

4. 运行信息

程序执行情况及某些运行结果会在 LOG 窗口记录下来，应注意观察 LOG 窗口中的信息，以确信程序是否被正确执行。在 PC 机中，若程序有语法错误，运行时在 LOG 窗口中显示红色的出错信息。当出现红色出错信息时，应当找出错误原因，调回程序进行修改后重新运行。应当注意的是，如果程序中有语句，后面忘记使用分号“;”，则 SAS 给出的出错信息往往不是指出丢失了分号，所以当看到 LOG 窗口中有红色出错信息时，应先检查程序中的语句是否没有用分号结尾。

5. 调回已发送的程序

当程序运行时，程序编辑窗口中的程序会被清除，若用户需要再看到程序并对其进行修改，可在程序编辑窗口用 RECALL 命令或用功能键 F9 将程序调回程序编辑窗口。

6. 程序的输出

当 SAS 系统运行用户的程序后，在 OUTPUT 窗口输出程序运行结果，如图 1.2 所示：

TTEST PROCEDURE						
Variable: X						
GROUP	N	Mean	Std Dev	Std Error	Minimum	Maximum
1	8	0.82320000	0.09303419	0.03289255	0.66080000	0.94640000
2	6	0.68320000	0.03680696	0.01502638	0.63280000	0.72240000
Variances	T	DF	Prob> T			
Unequal	3.8714	9.6	0.0033			
Equal	3.4600	12.0	0.0047			
For HO: Variances are equal, $F' = 6.39$ $DF = (7, 5)$ $Prob>F' = 0.0579$						

图 1.2 输出结果

用户可用窗口调用的功能键将光标移至 OUTPUT 窗口看到自己的程序运行结果。也可在 OUTPUT 窗口的命令行上用下面的命令将 OUTPUT 窗口的内容从打印机打印出来：

FILE 'PRN:' ↵

7. 清除窗口程序

可以采用 CLEAN 命令来清除 PROGRAM EDITOR、LOG、OUTPUT 三个窗口的内容。在程序编辑窗口时，若要清除旧程序以便输入新的程序，可以用 Home 键将光标移到命令行上，键入 CLEAN↵就可清除当前窗口中内容。

二、全屏幕编辑命令及窗口调用

程序编辑窗口是 SAS 的文字编辑器，在此窗口可以编辑任何文本文件和程序。可以使用行命令去移动、删除、拷贝和插入程序行。全屏幕编辑命令定义如下：

A, B

当用 C, M, CC, MM 去移动或拷贝一行或多行时，用 A, B 来标记源行的目标位置。在一行的行号上指明 A (After)，表示要把源行放在该行之后，若指明 B (Before)，表示要把源行放在该行之前。

C [n]

拷贝一行或多行到由目标行命令 (A, B) 指定的位置。在要拷贝的行的行号上打入 C, 然后移动光标至目标行的行号上, 打入 A 表示拷到该行之后, 打入 B 表示拷到该行之前。可以指定要拷贝行的行数 n, 只要在 C 之后加上行数 n 的数值和一个空格即可。

CC

将一行块拷贝到另外的位置时, CC 要放在所要拷贝的行块的第一行和最后一行的行号上。

D [n]

删除一行或多行。把 D 放到所要删除行的行号上后按回车键, 则删除该行。为了一次性删除多行, 可在 D 的后面加要删除的行数 n 和一个空格并回车即可。

DD

删除一行块。将 DD 放在所要删除的块的第一行及最后一行的行号上。

I [n]

插入一个或多个新行。将 I 放到要插入的行号之前的行的行号上并回车, 即可在执行 I 命令的行的后面插入一行。也可将 I 命令与 A, B 命令结合使用。例如, 要在第 3 行和第 4 行之间插入一个新行, 可以在第 3 行的行号上打入 I↵ 或 IA↵, 也可以在第 4 行的行号上打入 IB↵。要一次性插入多行时, 可在 I 后面加上要插入的新行数 n 和一个空格。

M [n]

移动一行或多行到文件中的另外位置。M 要放在所要移动的行的行号上。要一次移动多行时, 可在 M 后面加上要移动的行数 n 和一个空格, 然后在目标行的行号上指出 A 或 B, 以表明放在该行之后或之前。

MM

移动一个行块到文件中的另外地方。MM 要放在被移动的行块的首行及末行的行号上。

SAS 系统中有许多窗口, 在每一窗口都有如下所示的命令:

COMMAND⇒

在命令行上, 可以打入窗口调用命令:

PGM (F6)	程序编辑窗口
LOG (F3)	日记窗口
OUT (F4)	输出窗口
KEYS (F2)	功能键定义窗口
HELP (F1)	帮助信息窗口

在除 PGM、LOG 和 OUT 之外的任何窗口, 要想返回调用本窗口之前的窗口状态, 可在命令行上使用 END 命令。

§ 1.2 SAS 程序的存贮、读取及打印

§ 1.2.1 SAS 程序的存贮

可以把窗口中的程序贮存到软盘或硬盘上, 以便以后调用。使用 FILE 命令存贮当前窗口

中的程序时，所有输入的行都将被保存，而不仅仅是当前显示在窗口中的那些行。在命令行上打入：

```
FILE '文件名' ↵
```

则 SAS 程序就存贮在硬盘的当前目录中了。例如，要将程序存贮到 A 盘上，则在命令行上打入：

```
FILE 'a: 文件名' ↵。
```

若要将程序存到硬盘上的非当前目录中，则在 FILE 命令中的文件名前指出目录名并加上斜线“\”。例如，将程序以 ABC 为文件名存贮到硬盘上的 TJ 子目录下，命令格式为：

```
FILE 'C: \TJ\ABC' ↵
```

FILE 命令也可用于存贮 OUT（输出）窗口中的程序运行结果。将光标用功能键 F4 跳到 OUT 窗口的命令行上，打入 FILE 命令，格式如前所述。

§ 1.2.2 SAS 程序的读取

欲将一个存贮在硬盘或软盘上的 SAS 程序读入程序编辑窗口进行编辑，可使用 INCLUDE 命令。在命令行上打入：

```
INCLUDE '盘号：\子目录名\SAS 文件名' ↵
```

例如，将 A 盘上 TJ 子目录下的名为 PGM.SAS 的 SAS 文件读入程序编辑窗口，可在命令行上打入：

```
INCLUDE 'A: \TJ\PGM.SAS' ↵
```

若当前程序编辑窗口已经有内容，则使用 INCLUDE 命令从硬盘或软盘读取出来的文件被加到窗口中现有内容的后面。

§ 1.2.3 SAS 程序的打印

FILE 命令也可用于打印 PROGRAM EDITOR 窗口、LOG 窗口、OUTPUT 窗口的内容。用户若想将自己的 SAS 程序从打印机上打印出来，可将光标移到命令行上（用 Home 键移动光标），然后打入：

```
'FILE 'PRN:'' ↵
```

则当前窗口中的内容就从打印机中打印出来了。

§ 1.3 SAS 数据步

数据步是建立一个 SAS 数据集的。它有三个组成部分：

- (1) 定义数据集名；
- (2) 定义数据集中的变量个数及变量名；
- (3) 每个变量的所有观测值。这些观测值可以是数值型的，也可以是字符。

SAS 数据集的命令与变量命名的规则相同，以字母开头，长度不超过 8 个字符，数据集

名可以是字母数字混合。下面是一个数据步的例子：

```
DATA Exam5;                                JANE F 96
INPUT name $ sex $ IQ;                     FORD M 110
CARDS;                                     ;
JOHN M 101                                RUN;
MARY F 104
```

在这个例子中，DATA 语句后定义数据集名为 Exam5，用 INPUT 语句定义数据集中的变量个数、变量类型、变量名。本例共有三个变量，它们分别名为：name、sex、IQ，其中 name 变量和 sex 变量是字符型的，用“\$”表示这两个变量的类型。CARDS 语句表明数据行的开始，在它之后是每个变量的观测值。整个数据步以 RUN 语句结束。

应当注意，当数据步之后是接着过程步时，数据步中最后一个语句 RUN 可以省略，但最后一行数据的下一行上的“;”不能省略，该分号是数据行结束标志。

下面各章将数据写为 DATA 步。

§ 1.3.1 数据步中所用的语句

表 1.2～表 1.5 是数据步中所用的语句及作用说明。

表 1.2 文件操作语句

语句	作 用
DATA	表明数据步的开始，并指明所建立的数据集名称
INPUT	表明数据的变量名、顺序、类型
CARDS	表明数据行的开始
CARDS4	当数据行含有分号时，用于指示数据行的开始，用“;”表示数据结束
INFILE	当数据来自外部文件时，指明文件名
SET	从另一数据集中读入数据或将两个数据集中的数据纵向连接
MERGE	将两个数据集中的各观测值横向合并
UPDATE	利用一个数据集中的数据项更新另一个数据集中相应的项
FILE	指出文件名

表 1.3 作用语句

语句	作 用
赋值	产生变量或改变变量值
求和	计算累加和
IF	仅处理满足 IF 语句规定条件的观测值
DELETE	删除观测值
OUTPUT	将输出内容写入数据集
MISSING	规定特殊的缺项值
STOP	中断数据步

表 1.4 控制语句

语句	作 用
IF-THEN/ELSE	条件判断
GOTO	转向语句
DO-END	将 DO 与 END 之间的语句作为一个单元执行
LINK-RETURN	转向标记指出的程序处，遇到 RETURN 返回 LINK 标记的下一句

表 1.5 信息语句

语句	作用
ATTRIB	指明变量特性
FORMAT	指定变量输出格式
INFORMAT	指定变量输入格式
LABEL	给出变量的详细说明
LENGTH	说明变量的长度特性
RETAIN	设变量初始值
RENAME	改变变量名
ARRAY	定义数组
BY	定义分组处理变量
DROP	指定不写到数据集中的变量
KEEP	指定要写到数据集中的变量

§ 1.3.2 建立 SAS 数据集

SAS 数据集可分为临时的数据集与永久的数据集。临时的数据集仅在当前使用时存在，关机或退出 SAS 系统后就不存在了。永久的数据集在建立时就在用户指定的盘上存贮下来，以便以后调用。两种 SAS 数据集在建立时采用的语句略有不同。下面介绍 SAS 数据步建立临时数据集的方法。

一、DATA 语句

语句格式：DATA [数据集名] [选择项];

DATA 语句的作用是表明数据步的开始，并给出所建数据集的名称。

在 DATA 语句中，如果不给出数据集名，则 SAS 自动以 DATA1, DATA2 等依次命名所建立的数据集。

DATA 语句中的选择项一般可以不写，当数据集是特殊结构的，如相关矩阵、协方差矩阵时，这个选择项就不能省略，要用选择项：(TYPE=数据集类型)，来说明 DATA 步所建立的数据集的类型。例如，所建立的数据集名为 DATA1，DATA1 中的数据是相关矩阵中的元素，则 DATA 语句写成如下格式：

```
DATA DATA1 (TYPE=CORR);
```

二、INPUT 语句

语句格式：INPUT 变量表;

INPUT 语句的作用是指明要建立的数据集中所有变量的变量名、长度及数据输入格式。该语句的变量表中包括变量名、变量类型、变量长度。变量名之间用空格隔开。

INPUT 语句中的字符型变量要在变量名后跟一“\$”符号。如果在 INPUT 语句中说明的数据类型与输入的数据不相符，则该变量的值作为缺项值处理。

SAS 数据集的数据输入方法有三种：列表输入、列输入、格式化输入。三种输入方法的 INPUT 语句不同。