

# S A S 应用讲义

(中高级教材)

Statistical Analysis System 简称为 S A S , 可用来分析数据和编写报告。它是美国 S A S 研究所的产品, 在国际上被誉为标准软件, 在我国深受医学、农林、财经、社会科学、行政管理等众多领域的专业工作者的好评。

有关 S A S 的最新信息, 可以查看 <http://www.sas.com.cn>。

S A S 采用积木式模块结构, 其中的 S A S /S T A T 模块是目前功能最强的多元统计分析程序集, 可以做回归分析、聚类分析、判别分析、主成分分析、因子分析、典型相关分析以及各种试验设计的方差分析和协方差分析。

本讲义围绕 S A S 的应用, 讲述以下八部分内容:

- |                  |                 |
|------------------|-----------------|
| (1) S A S 应用基础;  | (2) S A S 常用语句; |
| (3) S A S 服务过程;  | (4) 描述性统计程式;    |
| (5) 方差分析程式;      | (6) 回归分析程式;     |
| (7) 聚类分析及判别分析程式; | (8) 互依性分析程式。    |

## 第一讲 S A S 应用基础

### 1. 1 S A S 的显示管理系统

启动计算机, 点击 S A S 图标后, 即可进入 S A S 的显示管理系统 D M S 。

D M S 是 Display Manager System 的缩写。在 D M S 中有四个主要的窗口:

- (1) 编辑窗口 (PROGRAM EDITOR) ——编辑程式和数据文件;
- (2) 日志窗口 (LOG) ——记录运行情况, 显示 E R R O R 信息;
- (3) 输出窗口 (OUTPUT) ——输出运行的结果;
- (4) 图形窗口 (GRAPH) ——输出图形。

点击 Globals 菜单中的 Program editor、Log、Output、Graph 命令可以进入编辑、日志、输出及图形窗口。

按功能键 F5、F6、F7 也可以进入编辑、日志及输出窗口。

退出 D M S 有两种方法:

- (1) 点击 File 菜单中的 Exit 命令;
- (2) 点击窗口右上角的 ×。

### 1. 2 S A S 的功能键

用功能键可以代替对菜单的点击, 有时比较方便。

最常用的功能键有 F1 : 显示帮助信息 (H E L P);

F4 : 显示已经运行的程式 (R E C A L L);

F5 : 进入编辑窗口 ( P G M );  
F6 : 进入日志窗口 ( L O G );  
F7 : 进入输出窗口 ( O U T P U T );  
F8 : 程式提交运行 ( S U B M I T );  
F9 : 显示功能键 ( K E Y S );

以上功能键的用法熟悉之后, 还可以定义或修改功能键。例如, 可以定义 Ctrl+E 为功能键, 用来清屏。

### 1.3 SAS 的程式结构

在 S A S 中, 对数据的处理可划分为两大步骤:

- (1) 将数据读入 S A S 建立的 S A S 数据集, 称为数据步;
- (2) 调用 S A S 的模块处理和分析数据集中的数据, 称为过程步。

每一数据步都是以 D A T A 语句开始, 以 R U N 语句结束。而每一过程步则都是以 P R O C 语句开始, 以 R U N 语句结束。当有多个数据步或过程步时, 由于后一个 D A T A 或 P R O C 语句可以起到前一步的 R U N 语句的作用, 两步中间的 R U N 语句也就可以省略。但是最后一步的后面必须有 R U N 语句, 否则不能运行。

S A S 还规定, 每个语句的后面都要用符号 “;” 作为这个语句结束的标志。

在编辑 S A S 程式时, 一个语句可以写成多行, 多个语句也可以写成一, 可以从一行的开头写起, 也可以从一行的任一位置写起。每一行输入完成后, 用 E N T E R 键可以使光标移到下一行的开头处。

### 1.4 SAS 程式的输入及运行

S A S 程式的输入及运行步骤如下:

- (1) 进入 S A S 的显示管理系统;
- (2) 进入并扩大编辑窗口;
- (3) 调出、编辑或修改 S A S 程式或数据文件;
- (4) 将编辑窗口的 S A S 程式或数据文件存盘;
- (5) 按功能键 F8 或点击 “跑步” 键运行 S A S 程式并注意观察日志窗口中的信息, 如有 E R R O R 出现, 应将光标移到日志窗口, 用 P g U p 和 P g D n 两键翻页, 找到错误的所在;
- (6) 将光标移到编辑窗口, 按功能键 F4 或点击 Locals 菜单中的 Recall text 命令调出已经运行的 S A S 程式, 改正错误后转入步骤 (4), 直到日志窗口中的信息没有 E R R O R 出现为止;
- (7) 将光标移到输出窗口, 用 P g U p 和 P g D n 两键翻页阅读输出的结果。

【上机练习 1.1】进入 S A S 的显示管理系统, 在编辑窗口输入程式:

```
data ex; a=1; b=2; c=3;  
x=2*a+b; y=a-b/2; z=b+c**3;
```

```
proc; print; run;
```

在程式中，`data` 后面的 `ex` 是给数据集所取的名字，在这个数据集内将储存变量 `a`、`b`、`c`、`x`、`y`、`z` 及它们所取的值，`proc` 后面的 `print` 是要打印 6 个变量所取的值。

将程式提交运行后，注意日志窗口中的信息。

NOTE: The data set WORK.EX has 1 observations and 6 variables.

NOTE: The DATA statement used 0.71 seconds.

```
3      proc;
```

ERROR 10-205: Expecting the name of the procedure to be executed.

```
3      print;
```

ERROR 180-322: Statement is not valid or it is used out of proper order.

```
3      run;
```

ERROR 180-322:Statement is not valid or it is used out of proper order.

在日志中记录了一个错误，那就是在 `proc` 与 `print` 之间不应该有 “;” 这个符号。将它去掉以后，请再注意日志窗口中的信息并阅读输出窗口中输出的结果。

【上机练习 1. 2】进入 S A S 的显示管理系统，在编辑窗口输入程式：

```
data ex; input no$ sex$ age h w;
cards;
10 f 47 156.3 47.1
24 m 38 172.4 61.5
53 m 41 169.2 64.5
46 f 52 158.2 53.6
38 f 39 160.1 48
;
proc sort; by sex;
proc means; by sex; var h w; run;
```

在程式中，`input` 定义与数据相对应的变量名、顺序及类型，其中 `no$` 与 `sex $` 是非数字型的串变量，`cards` 数据行开始的标志，数据行下面的 “;” 独占一行是数据行结束的标志，`sort` 是将 S A S 数据集中的观测值按一个或多个变量进行排序，后面的 `by sex` 是要根据 `sex` 的值对上述观测值进行分组处理，而 `means` 则是要在分组处理后按 `var` 的要求给出变量 `h` 和 `w` 的简单的描述性统计分析结果，如下图所示。

```
-----SEA=f-----
      N Obs Variable N Minimum Maximum Mean Std Dev
          3 H          3  156.3    160.1  158.2   1.90
          W          3   47.1     53.6   49.6   3.52
-----SEA=m-----
      N Obs Variable N Minimum Maximum Mean Std Dev
```

2 H	2	169.2	172.4	170.8	2.26
W	2	61.5	64.5	63.0	2.12

在程式中，cards；与“；”之间是数据，第一列是变量 no\$ 的数据，表示学生的号码；第二列是变量 sex\$ 的数据，表示性别；第三列是变量 age 的数据，表示年龄；第四列是变量 h 的数据，表示身高；第五列是变量 w 的数据，表示体重。

而输出的结果为：女生组有 3 人，身高在 156.3 至 160.1 之间，均值为 158.2，标准差为 1.90；体重在 47.1 至 53.6 之间，均值为 49.6，标准差为 3.52；男生组有 2 人，身高在 169.2 至 172.4 之间，均值为 170.8，标准差为 2.26；体重在 61.5 至 64.5 之间，均值为 63，标准差为 2.12。

## 第二讲 SAS 常用语句

在 SAS 中，数据集的建立、变换以及对数据集的统计分析都是通过 SAS 语句来实现的。常用的 SAS 语句有：

### 2.1 DATA 语句

DATA 语句的作用是表明数据步的开始并给出数据集的名称。

DATA 语句的格式为：

DATA 数据集的名称；

数据集的名称必须以英文字母开始，最长不超过 8 个字符。

数据集的名称都是由符号“.”分隔的两个部分所组成。第一部分标记数据集所存储的位置，称作一级名或库逻辑名。第二部分标记一个特定的数据集，称作二级名。如果不想把数据集永久储存在指定的磁盘上，可以只使用二级名。例如建立一个名为 EX 的数据集的语句为“DATA EX；”，这时 SAS 系统会自动地把 EX 作为数据集的二级名，并添加 WORK 作为数据集的一级名，因此在日志窗口显示的信息中记该数据集为 WORK.EX。但是这个数据集是临时的，它仅仅在程式运行期间有效，过后它就被复盖，不能重新调用。

### 2.2 CARDS 语句

CARDS 语句的作用是与“；”呼应，标志数据行的开始与结束。

CARDS 语句的格式为：

CARDS；

数据行

；

如果使用 CARDS 语句，在 CARDS 的后面必须紧跟数据行，并且在一个数据步中最多只能有一个 CARDS 语句。

### 2.3 INPUT语句

INPUT语句的作用是描述输入记录中的数据，并把输入值赋给相应的变量。

INPUT语句的格式为：

INPUT 数据的变量名、顺序及类型；

用INPUT语句是为了读外部文件的数据或跟在CARDS语句后面的数据。除非在INPUT语句中的变量名后有串符号或用字符的输入格式表示、或该变量事先已被定义为字符型，否则SAS认定用INPUT语句读入的是数值型变量的值。

用INPUT语句时，外部文件中的数据和CARDS语句后面的数据都采取列表输入的方法，各个变量的值由它们之间的空格来分隔。为从一行读入多个观测值，应使用行保持符@@限制读数指针，使其保持在这一行上读数，直到数据读完为止。

例如：INPUT X Y @@；

CARDS；

3.16 2.9 5.8 3 4.9 4.17

；

### 2.4 INFILE语句

INFILE语句的作用是指明外部数据文件的名称，并从这个外部数据文件中读取数据。

INFILE语句的格式为：

INFILE ‘外部数据文件名’；

例如，在A驱动器上有一个名称为YU.DAT的数据文件，其中数据排列为：

A 3.16 B 2.9 B 5.8

B 3 A 4.9 A 4.17

用INFILE语句读取数据的程式为：

DATA EX；

INFILE ‘A: YU.DAT’；

INPUT Z\$ X @@；

用INFILE语句读取数据时，要求外部数据文件必须是在操作系统中用TYPE命令打印出全部内容的ASCII码文本文件。

### 2.5 SET语句

SET语句的作用是按照指定的条件从指定的数据集中读取数据建立新的数据集或将两个数据集中的观测值纵向连接建立新的数据集。

SET语句的格式为：

SET 数据集名表；

例如，从数据集EX中读取数据建立新的数据集EX1，可编辑语句：

```
D A T A   E X 1 ;
```

```
S E T   E X ;
```

将两个数据集 D 1 和 D 2 中的观测值纵向连接建立新的数据集 D 3 , 可编辑语句:

```
D A T A   D 3 ;
```

```
S E T   D 1   D 2 ;
```

【上机练习 2. 1】进入 S A S 的显示管理系统, 在编辑窗口输入程式:

```
data d1; input g$ a b @@;
```

```
cards;
```

```
1 5 7 1 7 9 1 3 4
```

```
;
```

```
data d2; input g$ a c @@;
```

```
cards;
```

```
2 4 8 2 6 6 2 8 10
```

```
;
```

```
data d3; set d1 d2;
```

```
proc print; run;
```

将程式提交运行后, 注意日志窗口中的信息, 如无 E R R O R , 则转到输出窗口。

本程式运行的结果如下:

OBS	G	A	B	C
1	1	5	7	.
2	1	7	9	.
3	1	3	4	.
4	2	4	.	8
5	2	6	.	6
6	2	8	.	10

## 2. 6 M E R G E 语句

M E R G E 语句的作用是将两个数据集中的各个观测值横向合并建立新的数据集。

M E R G E 语句的格式为:

M E R G E 数据集名表;

例如, 将数据集 D 1 与 D 2 中的观测值横向合并建立新的数据集 D 3 , 可编辑语句:

```
D A T A   D 3 ;
```

```
M E R G E   D 1   D 2 ;
```

【上机练习 2. 2】进入 S A S 的显示管理系统, 在编辑窗口输入程式:

```
data d1; input a b c @@;
```

```
cards;
```

```

1 5 7 1 7 9 1 3 4 1 2 6
;
data d2;input x y z @@;
cards;
2 4 8 2 6 6
;
data d3; merge d1 d2;
proc print; run;

```

本程式运行的结果如下：

OBS	A	B	C	X	Y	Z
1	1	5	7	2	4	8
2	1	7	9	2	6	6
3	1	3	4	.	.	.
4	1	2	6	.	.	.

## 2. 7 D R O P 语句

D R O P 语句的作用是指定不写到数据集中的变量。

D R O P 语句的格式为：

D R O P 变量名表；

如果在 D A T A 步中有 D R O P 语句, 那么 D R O P 语句中已出现的变量的观测值被删除, 未出现的变量的观测值仍保留在新的数据集中。

## 2. 8 K E E P 语句

K E E P 语句的作用是指定要写到数据集中的变量。

K E E P 语句的格式为：

K E E P 变量名表；

如果在 D A T A 步中有 K E E P 语句, 那么 K E E P 语句中未出现的变量的观测值被删除, 已出现的变量的观测值被保留在新的数据集中。

## 2. 9 I F 语句

I F 语句的作用是使 S A S 继续处理符合 I F 条件规定的观测值, 因而所得到的数据集是原数据集的子集。

I F 语句的格式为：

I F 条件表达式；

如果表达式的值是真, S A S 处理当前的观测值, 执行 D A T A 步中的语句。如果表达式的值是假, S A S 马上返回到 D A T A 步的开始, 不处理当前的观测值, 不执行

D A T A 步中的语句。

【上机练习 2. 3】进入 S A S 的显示管理系统，在编辑窗口输入程式：

```
data ex; input group$ age @@;
cards;
x 23 y 10 x 7 y 22 x 8 y 4
x 21 y 12 x 9 y 22 x 6 y 2
;
data ex1; set ex; if group='x';
proc print;
data ex2; set ex; if age<20;
proc print; run;
```

本程式运行的结果如下：

OBS	GROUP	AGE
1	x	23
2	x	7
3	x	8
4	x	21
5	x	9
6	x	6

OBS	GROUP	AGE
1	y	10
2	x	7
3	x	8
4	y	4
5	y	12
6	x	9
7	x	6
8	y	2

## 2. 1 0 P R O C 语句

P R O C 语句的作用是指定需要调用的过程以及该过程的若干选择项。

P R O C 语句的格式为：

P R O C S A S 的过程名；

例如，调用 P R I N T 过程，打印数据集 E X 的内容：

P R O C P R I N T D A T A = E X；

这里的 D A T A = 数据集名，用来指定本过程所要处理的数据集名，如缺省则处理



最新建立的数据集。

在 P R O C 步中, 还必须确认一些最基本的信息, 包括:

- (1) 处理的数据集名, 格式为 D A T A = 数据集名;
- (2) 所涉及的变量名, 格式为 V A R 变量名;
- (3) 分组处理的标志, 格式为 B Y 组变量名。

### 2. 1 1 C L A S S 语句

C L A S S 语句的作用是在分析的过程中定义分类变量, 再按分类变量的值分组进行分析。

C L A S S 语句的格式为:

C L A S S 变量名;

C L A S S 语句对数据分析所起的作用类似于 B Y 语句, 但是它们打印输出的格式不相同, 并且 B Y 语句要求先对变量排序后才能使用。

### 2. 1 2 F R E Q 语句

F R E Q 语句的作用是规定一个变量, 以它的值作为数据集中观测值出现的频数。

F R E Q 语句的格式为:

F R E Q 变量;

假定某观测值的 F R E Q 变量的值为 n, 在分析时, 该数据集中这个观测值出现的频数便是 n。若  $n < 1$ , 则观测值不参加分析。若 n 不是整数, 则自动取整。观测值的总数等于 F R E Q 变量的和。

【上机练习 2. 4】进入 S A S 的显示管理系统, 在编辑窗口输入程式:

```
data ex; input group$ sex$ age number @@;
cards;
x f 23 10 y f 22 8
x m 21 12 y m 22 6
;
proc means maxdec=3; class group; var age;
proc means maxdec=3; class group; freq number; var age; run;
```

程式中的 maxdec=3 用来确定输出的数值只保留三位小数。

本程式运行的结果如下:

Analysis Variable : AGE						
GROUP	N	Obs	N	Minimum	Maximum	Mean Std Dev
x	2	2	21.000	23.000	22.000	1.414
y	2	2	22.000	22.000	22.000	0.000

Analysis Variable : AGE						
GROUP	N	Obs	N	Minimum	Maximum	Mean Std Dev
-----						
x	22	22		21.000	23.000	21.909 1.019
y	14	14		22.000	22.000	22.000 0.000
-----						

### 第三讲 SAS 服务过程

SAS 服务过程不能直接给出统计值，专门用来准备或整理统计资料。使频次最多的 SAS 语服务过程有：

#### 3.1 PRINT 过程

PRINT 过程可以打印一个 SAS 数据集中的全体或部分观测值，还可以打印数值变量的总和或部分和。

在 PRINT 过程中经常使用的语句有：

PROC PRINT [选择项];

VAR 变量表;

BY 变量表;

SUM 变量表;

在 PROC PRINT 语句中可能出现的选择项有 DATA = 数据集名，如果省略这一选择，则打印最新建立的数据集中的数据。

SUM 语句被用来求变量表中诸变量的总和。

#### 3.2 SORT 过程

SORT 过程将 SAS 数据集中的观测值按一个或多个变量的值进行排序，以便其他的 SAS 过程利用 BY 语句对这些观测值进行分组处理，排序后的观测值存放在一个新的 SAS 数据集中或者代替原来的数据集。

在 SORT 过程中经常使用的语句有：PROC SORT [选择项];

BY 变量表;

在 PROC SORT 语句中可能出现的选择项有 DATA = 数据集名，用来说明要排序的数据集名，如果省略这一选择，则指定最新建立的数据集。还有 OUT = 数据集名，用来说明要输出的数据集名，如果省略这一选择，则原来的数据集被输出的数据集替换。

【上机练习 3.1】进入 SAS 的显示管理系统，在编辑窗口输入程式：

```

data ex; length name$ 9;
input name sex$ group$ t1-t3 @@;
s=sum(of t1-t3);
cards;
wangdong 1 1 90 70 60 xueping 2 2 85 95 88
zhouhua 1 1 77 84 69 heyang 1 2 95 78 88
sunhong 2 3 89 97 87 zhaobin 2 3 66 98 86
huangshan 2 3 67 75 76 hufang 1 2 78 77 69
lili 2 2 70 82 63 zhaoli 2 3 95 90 91
zhangqun 1 1 93 91 89 iping 1 3 59 88 93
liyun 1 1 89 96 76 liyu 2 1 79 89 87
huajin 2 2 89 97 80
;
proc sort; by group s;
proc print; by group; run;

```

这里是按 S 的值升序排列，若要按 S 的值降序排列，则应在 B Y 语句中 S 的前面增加选择项 D E S C E N D I N G。

程式中的 `length name$ 9` 用来定义字符串的长度，当全部字符串的长度为 8 时，可以省略这个语句。

本程式运行的结果如下：

```

----- GROUP=1 -----
   OBS   NAME      SEX    T1    T2    T3    S
     1  wangdong    1     90    70    60   220
     2  zhouhua     1     77    84    69   230
     3  liyu        2     79    89    87   255
     4  liyun       1     89    96    76   261
     5  zhangqun    1     93    91    89   273
----- GROUP=2 -----
   OBS   NAME      SEX    T1    T2    T3    S
     6  lili        2     70    82    63   215
     7  hufang      1     78    77    69   224
     8  heyang      1     95    78    88   261
     9  huajin     2     89    97    80   266
    10  xueping     2     85    95    88   268
----- GROUP=3 -----
   OBS   NAME      SEX    T1    T2    T3    S
    11  huangshan   2     67    75    76   218
    12  iping       1     59    88    93   240

```

13	zhaobin	2	66	98	86	250
14	sunhong	2	89	97	87	273
15	zhaoli	2	95	90	91	276

### 3.3 FORMAT过程

FORMAT过程可以对字符或数值变量定义一个所需要的输出格式,使DATA步以及含有FORMAT语句的过程步中的变量都受到这一输出格式的限制。

通过PROC FORMAT过程所定义的格式包括:

- (1) 将输出值转变成不同的值、数值可以变成字符、字符可以变成另外的字符。
- (2) 使输出值保留一定的小数位,否则输出值将有8个小数位。

FORMAT过程由下列语句控制:

PROC FORMAT;

VALUE 格式名 格式的定义;

为便于识别,格式名通常是原变量名加FMT,如果原变量是字符串,则应当在格式名前加串符号。后面的过程中,如果涉及到某变量的输出格式,必须有语句:

FORMAT 变量名 格式名.;

【上机练习3.2】进入SAS的显示管理系统,在编辑窗口输入程式:

```
data ex; length name$ 10;
  input name sex$ group$ t1-t3 @@;
cards;
(数据同上机练习3.1)
;
proc format;
  value $sexfmt '1'='m' '2'='f';
  value $groupfmt '1'='a' '2'='b' '3'='c';
  value t1fmt 0-<60='C' 60-<80='B' 80-100='A';
proc print;
  format sex $sexfmt. ;
  format group $groupfmt. ;
  format t1 t1fmt. ;
  format t2 t3 4.1; run;
```

这里的sex、group是串变量,格式名为\$sexfmt、\$groupfmt, t1为数值变量格式名为t1fmt, t1的格式定义是0-<60变换为C、60-<80变换为B、80-100变换为A。本程式运行的结果如下:

OBS	NAME	SEX	GROUP	T1	T2	T3
1	wangdong	m	a	A	70.0	60.0
2	xueping	f	b	A	95.0	88.0

3	zhouhua	m	a	B	84.0	69.0
4	heyan	m	b	A	78.0	88.0
5	sunhong	f	c	A	97.0	87.0
6	zhaobin	f	c	B	98.0	86.0
7	huangshan	f	c	B	75.0	76.0
8	hufang	m	b	B	77.0	69.0
9	lili	f	b	B	82.0	63.0
10	zhaoli	f	c	A	90.0	91.0
11	zhangqun	m	a	A	91.0	89.0
12	iping	m	c	C	88.0	93.0
13	liyun	m	a	A	96.0	76.0
14	liyu	f	a	B	89.0	87.0
15	huajin	f	b	A	97.0	80.0

### 3.4 STANDARD过程

STANDARD过程按给定的平均值和标准差对SAS数据集中的部分或全体变量进行“标准化”变换，并形成新的数据集。

在STANDARD过程中经常使用的语句有：

PROC STANDARD [选择项];

VAR 变量表;

FREQ 变量表;

BY 变量表;

在PROC STANDARD语句中可能出现的选择项有DATA=数据集名，用来说明要“标准化”变换的数据集名，如果省略这一选择，则指定最新建立的数据集。还有OUT=数据集名，用来说明含有“标准化”变量的新数据集名，如果省略这一选择，SAS将按内部程式给出新的数据集名。

此过程将打印输出每个“标准化”变量的输入频数、平均值及标准差。

【上机练习3.3】进入SAS的显示管理系统，在编辑窗口输入程式：

```
data ex; length name$ 10;
    input name sex$ group$ t1-t3 @@;
cards;
    (数据同上机练习3.1)
;
data st; set ex; st1=t1; st2=t2; st3=t3;
proc standard mean=80 std=5 out=new;
var st1-st3; format st1-st3 5.2;
proc print data=new;
proc means data=new(drop=name group) maxdec=1; run;
```

程式中的 `format st1-st3 5.2` 用来定义新变量的格式、即小数点前两位、小数点后两位、包括小数点共计五位，可以省略这个语句。

本程式运行的结果如下：

	NAME	MEAN	STD	N
	ST1	81.40	11.58	15
	ST2	87.13	9.04	15
	ST3	80.13	10.68	15

  

OBS	NAME	SEX	GROUP	T1	T2	T3	ST1	ST2
ST3								
1	wangdong	1	1	90	70	60	83.71	70.52
2	xueping	2	2	85	95	88	81.55	84.35
3	zhouhua	1	1	77	84	69	78.10	78.27
4	heyang	1	2	95	78	88	85.87	74.95
5	sunhong	2	3	89	97	87	83.28	85.46
6	zhaobin	2	3	66	98	86	73.35	86.01
7	huangshan	2	3	67	75	76	73.78	73.29
8	hufang	1	2	78	77	69	78.53	74.39
9	lili	2	2	70	82	63	75.08	77.16
10	zhaoli	2	3	95	90	91	85.87	81.59
11	zhangqun	1	1	93	91	89	85.01	82.14
12	iping	1	3	59	88	93	70.33	80.48
13	liyun	1	1	89	96	76	83.28	84.90
14	liyu	2	1	79	89	87	78.96	81.03
15	huajin	2	2	89	97	80	83.28	85.46

  

N Obs	Variable	N	Minimum	Maximum	Mean	Std Dev
15	T1	15	59.0	95.0	81.4	11.6
	T2	15	70.0	98.0	87.1	9.0
	T3	15	60.0	93.0	80.1	10.7
	ST1	15	70.3	85.9	80.0	5.0
	ST2	15	70.5	86.0	80.0	5.0
	ST3	15	70.6	86.0	80.0	5.0

### 3.5 TRANSPOSE过程

TRANSPOSE过程将一个数据集进行转置，使行变为列而列变为行，也就是使原数据集中样品的观测值变换成新数据集中变量的观测值，而原数据集中变量的观测

值则变换成新数据集中样品的观测值。

TRANSPOSE 过程的格式为：

PROC TRANSPOSE [选择项];

VAR 变量表;

BY 变量表;

在PROC TRANSPOSE语句中可能出现的选择项有：

(1) DATA=数据集名, 用来说明要转置的数据集名, 如果省略这一选择, 则指定最新建立的数据集。

(2) OUT=数据集名, 用来说明转置所建立的新数据集名, 如果省略这一选择, SAS将按内部程式给出新的数据集名。

(3) PREFIX=字母(指定一个前缀, 用来构成新变量名)。

(4) NAME=字母(为含有输入数据集中被转置的变量名的输出数据集中的变量命名, 如果省略这一选择, SAS将按内部程式指定)。

在VAR语句中应列出要被转置的变量名, 否则原数据集中未在其他语句中列出的所有数值型变量都将被转置, 字符型变量必须在VAR语句中列出才能被转置, 未被转置的变量不进入新的数据集, 除非它们已列入COPY或BY语句。

BY语句使输入数据集分组转置, 分组变量被包括在输出数据集中。

【上机练习 3.4】进入SAS的显示管理系统, 在编辑窗口输入程式：

```
data ex; input a b c @@;  
cards;  
50 147 183  
49 85 136  
77 195 244  
;  
proc transpose prefix=y;  
proc print; run;
```

本程式运行的结果如下：

OBS	_NAME_	Y1	Y2	Y3
1	X1	50	49	77
2	X2	147	85	195
3	X3	183	136	244