第十六章　数据管理和综合应用

§16.1　数据管理及其计算机软件

为了更有效使用统计软件包，有一个计算机数据管理的映象是至关重要的。如方差分析手工的做法是先把数据分组，但在软件包处理时，是放在线性模型的框架之中的，需要用一个分组变量。反之，若在SAS 的数据步中使用简单的命令则可给分组数据加上分组标志。

数据管理即对数据的操作，内容包括数据库创建、数据录入、数据编辑、查找、索引、合并、追加和汇总、存档等；也即从表格录入、逻辑检错，到汇总分析和产出的全过程。

数据库是公用的为特定目标服务的数据集合，用于满足多种类型终端用户的需要，数据库管理系统(DBMS) 是数据库与用户间的接口。数据库模型有三种，即层次型、网状型和关系型，以关系型数据库最为重要。它用表格来表示实体与实体间的关系，用行或横栏表示记录或不同的观察对象，用列表示具有相同特征的数据。如人口普查中的数据项目有年龄、性别、职业、文化程度，住址等，横栏是不同的人，各项目表示了不同人的特征。关系型数据库的一个重要特征就是一个表的记录可以与另一个表中的记录关联。存贮信息的不同表形成一个数据字典，它记录了数据环境的逻辑和物理方面的安排。它可以是活动的或被动的，后者对于数据定义进行维护，但对数据库的访问无控制。一个好的数据字典应能够：ａ、对数据格式和类型提供标准定义；ｂ、维护对应用程序提供的交叉参照数据列表；ｃ、对与系统有关的项目，包括用户数、缓冲区数目和大小、那一个终端与系统连接、那一个程序与用户受系统特定变动的影响，等等。与数据库不同，数据字典记录的是数据库自身的结构，数据字典的内容可视做关于数据与系统的综合资料，有时称做公用数据字典(common data dictionary, CDD)。SPSS/PC+的数据录入工具DE是一个范例。SAS使用Windows下的动态数据交换可直接读写EXCEL的数据。

数据库是计算机领域中最重要的技术之一，已成为计算机软件独立的分支。数据库技术产生于六十年代初期。现已广泛应用于工农业生产、交通运输、商业、行政管理、科学研究、医疗卫生事业和国防建设等。

各种统计软件包有其自身的数据格式，称为系统文件，其中包括数据的创建时间、变量的格式、标号等。其优点在于操作方便，处理速度快，但其内容多在终端上不能正确显示，也难以用其它软件进行编辑，必须经中介的文件格式进行软件包的数据交换。软件包在数据管理上有其自身的特点，如StatGraphics 软件进行数据分析时，能够进行数据文件之间的跨越调用。多数软件包能够处理缺失值，在运算时，或者当缺失发生在分析的变量时删除该变量，或者当一个记录出现缺失时，删除该记录，进行统计分析时应注意适当选用。处理的原始数据，既可以是坐标类型原始数据，也可以是前期运行后的对称矩阵。另一种软件间常用的数据交换格式是ASCII 类型文件，PC 机上比较标准的格式如dBASE 数据库文件、电子报表如LOTUS 1-2-3 格式等。

dBASE 是一个优秀的微机关系型数据库管理软件，dBASE Ⅱ、dBASE Ⅲ、dBASE ⅢPLUS及dBASE Ⅳ均是Ashton-Tate 公司推出的dBASE 产品。利用dBASE 进行数据管理，首先要根据调查项目建立数据库结构，包括各个字段的名称、类型、长度、小数位数等。以后便可进行数据的录入、编辑、插入等操作，还可以增加新的项目或字段。这些操作即是数据库的维护，各操作对应不同的数据库管理功能。这些命令的组合，形成了数据库管理程序。在VAX/VMS系统和UNIX 系统下也有dBASEⅣ相应的软件。FoxBASE+ 是美国FOX 软件公

司于1987 年2 月推出的关系数据库系统，它与Ashton-Tate 公司的dBASE lllPLUS 完全兼容，而且有多方面扩展。其速度更快，适用机种广泛，支持的操作系统多，最近推出的FoxPro 2.0 则保持了同FoxBASE+ 的兼容并对FoxBASE+ 的功能进行了拓展。

其它流行的数据库软件如ORACLE 最早用于IBM 大型机,开发于1979 年，使用结构查询语言SQL，现已用于MS-DOS、Unix、VM/SP、MVS/SP、MVX/XA 及VMS 系统，本章结合SAS用例加以介绍。SQL 与宿主语言结合的另一种方式是嵌入方式，目前SQL 语言标准允许嵌入的宿主语言有Fortran、COBOL、Pascal、PL/I 四种。SAS和SPSS均对它有相应的支持。

综合应用是即充分利用现有计算机各种系统资源，进行计算机与软件包之间数据与程序的交换，以扬长避短。本章介绍几个PC机与VAX/VMS系统操作的例子。

§16.2 原始数据的录入和管理

计算机数据处理和统计分析，首先要进行原始数据的录入。如在流行病学研究中，常常要进行一般项目的调查，如姓名、年龄、性别等，调查表可能是以下的形式：

识别码(ID) □□□

一、姓名(NAME) _____ ----------

二、年龄(AGE) _____ 岁 □□□

三、性别(SEX) 1.男(M) 2.女(F) □

四、身高(HEIGHT) _____ 厘米 □□□.□□

五、体重(WEIGHT) _____ 公斤 □□□.□□

. . .

第二列专为计算机录入使用，使用中英文数据库管理软件，数据库变量名可为中文或英文，(英文名放在括号内)，各变量的信息为：

名称	(英文名)	类型	宽度	小数位数	起止位置
识别码	(ID)	数字型	3	0	1-3
姓名	(NAME)	字符型	10		4-13
年龄	(AGE)	数字型	3	0	14-16
性别	(SEX)	字符型	1		17
身高	(HEIGHT)	数字型	5	2	18-23
体重	(WEIGHT)	数字型	5	2	24-29

对调查、实验获得的原始表，首先要进行标识项目的检查，从表格上应该看出它是来自那个单位，其隶属关系如何。同时，标明调查日期。为了方便计算机分类处理，不同的表格可以编上一些标识码；其次是数据项之间合理性的判断。对于大量的数据，只要有可能，对数据进行编码；使用最详细的记录，如记录年龄时记录生日；经常对数据进行备份；对每个量用一个代码表示缺失值。数据处理和分析的流程是：

建库→原始数据录入→逻辑检查→数据分析→报告。

本例使用前面介绍的Epi Info 软件可直接录入并处理。现用流行的dBASE 软件建库，使用CREATE、APPEND 命令即将数据录入，生成.DBF 文件，使用带有SDF 选项的COPY 命令，可将数据拷成标准格式的ASCII 码文件，由所用的软件读取。

dBASE数据库命令的语法可在进入数据库系统后，借助软件的帮助命令(HELP) 获得。dBASE用前四个英文字母作为关键字。其命令子句也是其可选项，如：

COPY <范围> TO <文件名> FIELDS <字段名表> [FOR/WHILE< 条件表达式>] [SDF/DELIMITED]。

其中大写字母表示关键字，方括号表示可选项，当关键字被指定时，其后的尖括号为必选项。<范围>有三种：ALL 表示所有记录；RECORD <表达式> 是表达式所指定的单一记录；NEXT <表达式> 表示从当前记录开始后指定数目的记录，表达式指定了记录的数目。字段名表是一系列用逗号分开的字段名，只有被指定的相关的字段才被操作。在dBASE 的语句中，使用FOR/WHILE 限定作为数据库操作的逻辑表达式，最后的SDF/DELIMITED 子句指定系统生成标准格式或带有分界符的ASCII 文件。考虑到一些微机统计软件尚不能直接使用dBASE 的数据库，SDF /DELIMITED 子句很有用。

下面列出的是其最常用的指令，各指令的用法可参有关资料。

USE	打开和关闭数据库。
APPEND	向数据库追加记录。
EDIT	编辑记录。
DELETE	删除记录。
INSERT	插入记录。
JOIN	数据库文件间的连接。
CHAGNE	改变字段和记录内容。
CREATE	创建数据库文件、报表格式文件和标签文件。
UPDATE	改变记录内容。
DISPLAY	分屏显示数据库结构和记录内容。
LIST	连续显示数据库结构和记录内容。
MODIFY	修改数据库结构、命令文件、报表格式文件和标签文件。
SORT	按关键字段对数据库进行排序。
TOTAL	按关键字段对数据库的数据求和。
INDEX	建立数据库索引文件。
REPORT	生成报表格式文件。
LABEL	生成标签文件。

dBASE 常用文件类型，可以通过文件的扩展名来区分，这样，可以对有关文件进行适当的维护，如：

.DBF	数据文件，用于保存数据和暂存文件。
.PRG	命令文件，用于完成数据管理的某种功能。
.FOX	编译程序，功能与.PRG 文件相同。
.FMT	格式文件，进行数据录入时的屏幕格式和产出时的格式定义。
.BAK	备份文件，在数据库文件修改时可以生成。
.NDX	索引文件，用于查找、排序等功能。
.FRM	报表文件，用于产生报表。
.LBL	标签文件，用于打印数据库项目的标签。
.MEM	内存变量文件，可存放dBASE 运行时的内存变量。

需要管理多个数据库时，可启用不同的区。数据库管理程序是运行采用DO < .PRG 命令文件> 执行。

与软件包的系统文件相比，dBASE 文件没有专门的缺失值定义，使用SAS 和SPSS/PC+ 等软件在dBASE 数据转贮时进行了特殊的处理。另外，dBASE 不能使用变量标号，因而数据文件不很直观；对于每个记录，则可使用注释字段的方式来解决。单个dBASE 文件最多处理128 个变量。

§16.3 软件包数据管理

§16.3.1 SAS

具有强大的数据管理功能。大部分操作是通过SAS/BASE 来完成，其强大的语言特色也主要表现在其数据步上。SAS也用一系列专用过程如CATALOG、DATASETS、COMPARE、APPEND、COPY、TRANS用于目录和数据集的管理，CONTENTS 用于浏览不同存贮类型文件的内容。SAS/FSP 提供了全屏幕操作，包括文件的创建、编辑等，有专门的屏幕控制语言(SCL) 。另外，SAS/ACCESSS、SAS/CONNECT 、SAS/SHARE 等工具用于各种计算机系统和软件包间的数据交换。在PC SAS上，使用PROC DBF 和DIF 进行dBASE 等文件的转换。使用DBF 生成dBASE 文件时，由于后者没有缺失值和标号，用填满9 的16 位宽的字段表示缺失值，转出时，为了保证数据格式的正确，应使用FORMAT 语句对变量格式行说明。假设现有ASCII 文件，以外部文件的格式读入，则应在DATA 步中使用INFILE 语句，若是dBASE 的COPY 命令生成的标准数据集(SDF)，可以采用带有格式的INPUT 语句在DATA 步读入，如上节的例子，INPUT ID 1-3 NAME $ 4-13 AGE 14-16 SEX 17 HEIGHT 18-22 WEIGHT 23-27；使用dBASE 的COPY 命令生成的有分界符的文本文件，则可使用DELI WITH BLAN 子句，一个记录的数据项之间用特殊分界符分开，可在INFILE 语句中指示dlm='分界符'。当文本文件宽度超常时，应指定LRECL= <记录宽>，其它的选项可参考有关说明书。

【例16.1】以下程序读入一个对称阵，当一行读不满时，使用MISSOVER 避免了到下一行继续读入数据。

* 杨维权等:《多元统计分析》;

```
data p341(type=corr);
infile cards missover;
input _name_$ x1-x3 _type_$;
cards;
x1  corr 1.0000000
x2  corr -.3333333  1.0000000
x3  corr 0.6666667  0.0000000  1.0000000
.   n    5          5          5
;
```

数据作为统计过程的选项，使用DATA= 引用被分析数据，OUTSTAT= 生成包含统计量的数据集，OUT= 指示生成的带有原始数据的文件。象PROC REG 一类过程可以使用专门的OUTPUT OUT= 语句保留计算结果。

SAS 在系统文件管理上，采用两水平的文件名，首先把计算机物理路径赋给一个逻辑的库名，在以后的操作中使用“库名.数据集名”的格式调用。SAS 把这样生成的数据集称为永久性数据集。

【例16.2】下面的程序利用DATASETS 过程进行永久性数据集间的操作。

```
LIBNAME MY 'C:\MYDIR1';                /* 定义库名*/
LIBNAME YOUR 'C:\YOUR\MYDIR2';
PROC DATASETS LIBRARY=MY COPY OUT=YOUR; /* 调用DATASETS 过程*/
SELECT CLASS1 EXAM1;
```

```
CHANGE EXAM1=EX1 EXAM2=EX2;
DELTE CLASS2;
MODIFY CLASS1; /* 修改数据集CLASS1 */
RENAME NAME1=NAME GOUK1=TOTAL;
LABEL EIGO='TEST-1' SUGAKU='TEST-2';
RUN;
```

【例16.3】下面程序用CONTENTS 把D:\SAS\SASINST 目录中所有文件列出来。

```
LIBNAME INST 'D:\SAS\SASINST';
PROC CONTENTS DATA=INST._ALL_; RUN;
PROC PRINT DATA=INST.CLASS;    RUN;
```

CONTENTS 关于数据集的产出包括数据集名、记录数、变量数、数据库及变量标号、记录长度及类型和格式。如第四章用例数据集CLASS.SSD 的内容如下：

```
Data Set Name:  SS.CLASS                Type:
Observations:   19                      Record Len: 37
Variables:      5
Label:          Student information
          -----Alphabetic List of Variables and Attributes-----
#  Variable  Type  Len  Pos  Label
3  AGE       Num     8   13  Age in years
4  HEIGHT    Num     8   21  Height in inches
1  NAME      Char    8    4  First name
2  SEX       Char    1   12  Gender
5  WEIGHT    Num     8   29  Weight in pounds
```

数据列表：姓名(NAME) 与性别(SEX) 是字符型数据，OBS 栏指示记录号。

OBS	AGE	NAME	SEX	HEIGHT	WEIGHT
1	15	JANET	F	62.5	112.5
2	11	JOYCE	F	51.3	50.5
3	14	JUDY	F	64.3	90.0
4	14	CAROL	F	62.8	102.5
5	12	JANE	F	59.8	84.5
6	12	LOUISE	F	56.3	77.0
7	13	BARBARA	F	65.3	98.0
8	15	MARY	F	66.5	112.0
9	13	ALICE	F	56.5	84.0
10	12	JOHN	M	59.0	99.5
11	12	JAMES	M	57.3	83.0
12	14	ALFRED	M	69.0	112.5
13	15	WILLIAM	M	66.5	112.0

```
14     13    JEFFREY     M      62.5       84.0
15     15    RONALD      M      67.0      133.0
16     11    THOMAS      M      57.5       85.0
17     16    PHILIP      M      72.0      150.0
18     12    ROBERT      M      64.8      128.0
19     14    HENRY       M      63.5      102.5
```

【例16.4】下面是一个论文报告的例子，采自SAS 6.07 PROC SQL 示范程序。是数据查询管理的好范例，程序paper.sas 创建一个表，用于以后的查询。变量为作者名、专题类别、标题、开始时间和持续时间。

```
data paper;
 input author$1-8 section$9-16 title$17-43 @45 time time5.
       duration;
 format time time5.; label title='Paper Title';
 cards;
 Tom     Testing Automated Product Testing    9:00 35
 Jerry   Testing Involving Users              9:50 30
 Nick    Testing Plan to test, test to plan  10:30 20
 Peter   Info SysArtificial Intelligence      9:30 45
 Paul    Info SysQuery Languages             10:30 40
 Lewis   Info SysQuery Optimisers            15:30 25
 Jonas   Users   Starting a Local User Group 14:30 35
 Jim     Users   Keeping power users happy   15:15 20
 Janet   Users   Keeping everyone informed   15:45 30
 Marti   GraphicsMulti-dimensional graphics  16:30 35
 Marge   GraphicsMake your own point!        15:10 35
 Mike    GraphicsMaking do without color     15:50 15
 Jane    GraphicsPrimary colors, use em!     16:15 25
 ;
```

下面程序便用PROC SQL 过程对上述数据进行查询操作：

1、选择

```
%include paper;
proc sql;
  * 以下语句列出表的所有信息;
  select * from paper;
  * 多长时间结束?;
  select author, title, time, duration,
              time + duration*60 as endtime
     from paper;
  * 现在加上一些标号和格式以更明了;
```

```
  select author, title, time,
                duration label='How Long it Takes',
                time + duration*60 as endtime format=time5.
     from paper;
  * 哪些论文是上午报告? 使用where 子句控制;
  select author, title, time,
                duration label='How Long it Takes',
                time + duration*60 as endtime format=time5.
     from paper where time < '12:00't;
```

2、创建

```
%include paper;
proc sql;
  /* SQL 用三种方式创建SAS 数据集*/
  /* 1) 作为其它表的空拷贝, 2) 作为任何有效SQL select 表达式的结果，3) 从传统的SQL 数据操纵语言(DML) 生成。下面的表P2 是PAPER的拷贝，P3 包括了所有12:00 以后交的论文*/
  create table p2 like paper;
  create table p3 as select * from paper where time > '12:00't;
  * 下面是一个新表;
  create table counts(section char(20),papers    num);
```

3、删除

```
%include paper;
proc sql;
  /* 创建一个表，用于演示数据的增删*/
  create table counts( section char(20), papers num );
  insert into counts values('Graphics', 4)
                          values('Info Sys', 3)
                          values('Testing',   2)
                          values('Users',      3)
                          values(",            1);
  * 删除以前的情况;
  select * from counts;
  * 用where 删除那些对于section 变量无意义的论文;
  delete from counts where section is null;
  select * from counts;
  * 宣读的论文仍然很多，含论文数目最小的专题将被取消;
  delete from counts
     where papers = ( select min(papers) from counts );
  select * from counts;
```

4、插入

```
%include paper;
proc sql;
```

```
/* 有两种方法把数据插入SAS 数据集*/
* 1) 插入常数值, 2) 使用SQL select 选择的数据;
* Jost 提交一份外语问题的新论文;
insert into paper(author, title, time)
      values('Jost', 'Foreign Language Issues', '11:15't);
* 插入以后的结果;
select * from paper;
* 创建一个新表Counts, 包括专题号及其论文数目;
create table counts( section char(20), papers num );
insert into counts
select section, count(*) from paper group by section;
select * from counts;
```

5、连接

先构造一个专题和园桌讨论会的数据，内容为专题、所在房间号、专题召集人、园桌会议主持人和讨论题。

```
data section;
 input section$1-8 room$ convenor$;
 cards;
Graphics  Sable  Denise
Info Sys  Kudu   Peter
Testing   Sable  Jenny
Users     Kudu   Sally
data roundt;
 input leader$1-8 subject$9-30;
 label subject='Roundtable Subject';
 cards;
Mary    External DBMS's
Nick    Testing Networks
Jerry   User Specifications
Peter   Selling Solutions
Jim     Distasteful Jokes
Marge   Designing Fonts
```

```
proc sql;
  * 哪些作者是园桌会议的主持人?;
  select author, title, subject
     from paper, roundt where author = leader;
  * 将讨论哪些论文和园桌论题?;
  select author, title, subject
     from paper full join roundt on author = leader;
  * 讨论的内容是什么，谁负责？;
  select coalesce(author, leader) as person, title, subject
```

```
    from paper full join roundt on author = leader;
  * 下一位报告人是谁?;
  select p.author label='Just Heard',
         n.author label='Then Try', n.title, n.section, n.time
    from paper p, paper n
  where n.time between p.time + p.duration*60
                   and p.time + (30+p.duration)*60
  order by p.author;
  * 谁非常忙，交论文、主持园桌会议和专题?;
  select distinct author from paper, section, roundt
  where author=convenor and author=leader;
  * 有必要在门厅处张贴一个简报，写明论文、园桌论题内容及负责人;
  select coalesce(author,convenor,leader) as person,
         title label='Gives Paper:',
         section.section label='Convenes Section:',
         subject label='Leads Roundtable on:'
    from paper full join section on author=convenor
                full join roundt   on   coalesce(author, convenor)
                =leader
  order by 1;
6、更新
%include paper;
  /* update 语句就地更新SAS 数据集*/
  * 使用的值有：1) 常量, 2) SQL 查询的结果, 3) 关于原始值的表达式;
  * 为了更新，先插入新到的Jost 的情况;
proc sql;
  insert into paper(author, title, time)
    values('Jost', 'Foreign Language Issues', '11:15't);
  * Jost 无专题，给它命名为"Users";
  update paper set section='Users' where author='Jost';
  select * from paper;
  /* Jost 认为他的论文会和别人的一样长，因此使用SQL update 语句设其持续时间
为所有论文的平均值。UNDO_POLICY 的默认选项可使PROC SQL 唯一地访问被插
入的数据集(sql.newprice)，对同一个表第二次引用时会失败，使用UNDO_POLICY=OPTIONAL
使查询得以继续*/
  reset undo_policy=optional;
  update paper
      set duration = ( select avg(duration) from paper )
    where author = 'Jost';
  reset undo_policy=required;
  select * from paper where author='Jost';
  * 因为大家都想多睡一会儿，因此每篇论文宣读也较原来的推迟30 分钟;
```

```
  update paper set time = time + '0:30't;
  select * from paper;
```

7、视图

```
%include paper;
```

创建视图，给原始数据增加一列，根据是starttime 和duration 得到的结束时间。

```
proc sql;
        create view pt as
        select author, title, time, duration label='Duration',
                time + duration*60 as endtime format=time5.
          from paper;
  * PROC SQL 中视图若同真正数据集一样;
  select * from pt where endtime > '14:00't;
  * 做一个论文宣读的清单和时间表;
  * SAS 过程可以访问SQL 视图，就象真数据库一样;
proc print data=pt;run;
proc timeplot data=pt;
  plot time='<' endtime='>' / overlay ref='12:00't hiloc;
  class author;
  run;
  * 论文宣读的顺序是什么？;
proc sql;
  create view pt_time as select * from pt order by time;
proc print data=pt_time; run;
  * 这个新的次序影响时间表吗？;
proc timeplot data=pt_time;
  plot time='<' endtime='>' / overlay ref='12:00't hiloc;
  class author;
  run;
```

【例16.5】使用SAS for Windows 与MS Excel进行数据交换。假设在MSExcel 有一个活动的工作表文件名为sheet1，现用SAS 产生三个随机数写入该活动文件，然后再把它读取打印，程序如下：

```
filename random dde 'excel |sheet1 ! r1c1:r100c3';
data random;
  file random;
  do i=1 to 100;
       x=ranuni(i);y=10+x;z=x-10;put x y z;
  end;
run;
filename monthly dde 'excel | sheet1 ! r1c1:r10c3';
```

```
data monthly;
   infile monthly;
   input var1 var2 var3;
run;
proc print;run;
```

§16.3.2 SPSS/PC+

命令GET 和SAVE 用于读入(GET) 和存贮(SAVE) SPSS/PC+ 系统文件，其格式为：

GET /FILE='文件名'.

SAVE /OUTFILE='文件名'.

SPSS/PC+ 的WRITE 命令可用于转贮ASCII 文件，FLIP 用于进行数据的转置。

在SPSS/PC+ 的显示管理方式下，运行命令DE. 即启用它的全屏幕数据管理工具DATA ENTRY。DATA ENTRY 工具除进行系统文件的录入、读写外，也可以读入dBASE/Lotus 等格式的文件。

DM 主菜单(Main Menu) 内容如下，用↑F1-↑F10(Shift 与功能键的组合)来调用。DE 的功能分成七类，在各部分中，功能键有不同的定义，使用F1 进入菜单，然后打需要帮助的功能键，使用空格键退出帮助窗口，使用ESC 键退出帮助状态。

↑F1 Help	帮助信息
↑F2 Files	文件操作
↑F3 Forms	屏幕表式
↑F4 Dictionary	数据字典
↑F5 Data	数据管理
↑F6 Cleaning	数据清理
↑F7 Skip&Fill	跳转定义
↑F8 Options	运行选项
↑F9	(未用)
↑F0 Exit	退出

在文件操作中，可以定义一个新文件，读取或存贮SPSS/PC+ 系统文件，或者电子报表、数据库文件、交换文件，或ASCII 码文件；把现有的数据字典拷贝到新文件，列目录；编辑、读取或存贮ASCII 数据模板。

使用逻辑定义可以定义和修改数据的范围、规则(RULES, 变量间的逻辑表达式) 。范围和规则定义之后，可以对数据进行扫描并且打印符合条件的记录，可利用数据部分来检查或改正这些记录。

数据字典可以为新文件定义变量，或者修改现有文件中变量的定义。新文件只有定义了变量以后才能录入数据，从现有文件增加或删除变量，必须存贮这个文件，然后读入开始录入。

数据管理用于数据录入，或者修改数据。这可以由电子报表或表样方式录入。前者以记录号为行，以变量为列，每屏20行，后者按表部分的定义显示。

跳转定义允许定义和修改跳转逻辑规则，每个规则与一个变量关联。若规则在数据部分作了修改，跳转规则可用于填充其它变量的取值以及决定下一个编辑的变量。

运行选项为DE 程序运行设定或修改环境。

```
————————Create/Edit Dictionary————————
                    Help F1    aF1 Help
          Define Variable F2   aF2
            Edit Variable F3   aF3
            Copy Variable F4   aF4
       Edit Value Labels F5    aF5 Edit Field Help
       Copy Value Labels F6    aF6 Copy Field Help
                          F7   aF7
Set Display Mode (All) F8      aF8
          Delete Variable F9   aF9
                          F0   aF0
————————Press Function Key to Select————————
```

退出DE 程序将返回DOS 或SPSS/PC+。

在DE运行的任何时刻打F1，系统给出相应的提示。控制菜单(Ctrl Menu) 使用Ctrl 键与功能键的组合，如^F5显示缓冲区，^F9提供变量信息，^F10表示完成。

用例，现拟建一个名为NEW.SYS 的系统文件。选择↑F2 读入文件部分，使用F4 定义新文件，退出。然后使用↑F4定义数据字典，其屏幕格式如下：

打F2定义变量。现设变量为年龄(age)，标号为"age in years'，宽度为3，缺失值为-999，有关信息如下：

```
Variable Name     age
Variable Label    age in years
Type of Variable  Numeric
Variable Length   3
Decimal Places    0
Display Mode      Edit
Missing           999
```

以^F10 完成。

使用↑F6，选择F2(Define Range/Rule)定义年龄的取值范围，现规定为1-150，即1 thru 150，以^F10退出。

使用↑F7，F2 (Define Skip Rule)定义变量间的跳转定义，^ F10退出，有关的跳转表达式操作符如下。

————————————Available Operators————————————

-	mod	and	&	lt	<	″	not	if	exit
+	in	or	*mid*	le	<=	**	implies	else	->
⋆	thru	eq	=	ge	>=	,	(...)	display	;
/	by	ne	″=	gt	>	<>	{...}	nextcase	

本例只定义年龄一个变量故不使用该选项。

为了今后数据录入方便，定义数据录入的屏幕格式，使用↑F3，F2(Generate default form)，使用F10(draw box)可以画方框，结束时用^F10退出。

—File Type to Save—

SPSS/PC+
SPSS Portable
123 Rel 1A
123 Rel 2
Symphony Rel 1.0
Symphony Rel 1.1-2.0
dBase II
dBase III
dBase IV
Multiplan Symbolic

——————————————Variables ——————————————————
ALL TO $CASENUM $DATE $WEIGHT AGE
—————————————— AGE ——————————————————

Age in years
Type: Numeric Missing value: 999 Width: 3 Decimals: 0
No value labels *
————Type Esc or punctuation character to remove menu——

录入样本数据：退回主菜单后使用↑F5，或者按照定义的屏幕格式录入，或者按照电子表格那样的形式录入，该项功能用F10切换。数据录入过程中，如出现非法数据或条件，屏幕自动跳出给予相应的警告信息。

全屏幕数据录入工具运行环境可用↑F8来改动，F2为读取现有设置。

录入结束后，仍以↑F2，选择文件存贮，系统提示：

择第一项，存贮为SPSS/PC+格式，压缩存贮。

现用↑F10退出DE 至SPSS/PC+ 系统看建立的效果。

```
SPSS/PC:get /file 'new.sys'.
 The SPSS/PC+ system file is read from
     file new.sys
 The file was created on 12/02/93 at 15:47:59
 and is titled SPSS/PC+ System File Written by Data Entry II
 The SPSS/PC+ system file contains
       7 cases, each consisting of
       4 variables (including system variables).
       4 variables will be used in this session.
```

打F1 键，选择变量列表(Var List)，其各变量内容如下：

若活动文件已经存在，将给予<active file>的提示，下面是BASETEST. INC文件生成的数据提示。

————————————<active file>————————————

Title: SPSS SYSTEM FILE. IBM PC DOS, SPSS/PC+ V3.0
Date of Creation : 12/2/93
Time of Creation : 15:46:07
Number of Variables: 18 (23)
Number of cases : 100
System File Version 2, Compressed

————————Press space to continue-————————

SPSS/PC+ 也可读入矩阵类型的数据，下面示例是上面SAS 因子分析用例在SPSS/PC+ 中实现的程序。

```
DATA LIST MATRIX FREE/ X1 TO X3.
N 88.
BEGIN DATA.
1.0000000
-.3333333  1.0000000
0.6666667  0.0000000  1.0000000
END DATA.
FACTOR READ=COR TRIANGLE /VARIABLES=X1 to X3/CRITERIA FACTORS (3)
      /EXTRACTION ML /PRINT=CORRELATION EXTRACTION ROTATION FSCORE.
```

JOIN 命令进行文件的合并(MATCH) 或追加(ADD)，其格式为：

JOIN MATCH FILE='文件名' /KEEP=变量名/DROP=变量名/RENAME (旧名=新名) /MAP (/BY 变量名)

JOIN ADD 的用法与之类似。

生成总计(AGGREGATE) 文件，其格式为：

AGGREGATE OUTFILE='文件名' /PRESORTED /BREAK=变量表(A|D) /MISSING=columnwise /AGGVAR '标号'... =函数(变量表,参数)

转入其它格式的数据文件:

TRANSLATE FROM FILE='文件名' /TYPE=WKS ... /DROP=变量表/KEEP=变量表/FIELDNAMES RANGE 范围/MAP.

转贮其它类型的文件使用TRANSLATE TO 命令，其格式与TRANSLATE FROM 相类似。可进行交换的数据格式有：LOTUS(WKS、WK1、WK3) 和SYMPHONEY 数据(WRK、WR1、SLK)，以及DBASE 数据(DB2、DB3、DB4)。

SPSS-X 格式文件转入：

IMPORT FILE='文件名' /KEEP=变量表/DROP=变量表/RENAME=旧名=新名

SPSS-X 格式文件转贮：

EXPORT OUTFILE='文件名' /KEEP=变量表/DROP=变量表/RENAME=旧名=新名/MAP /DIGITS=小数位数

§16.3.3 BMDP

这里介绍DM模块。DM模块各段落的用法可经HELP段来查看，如：HELP READ. / 给出READ段的用法。除END和FINISH外，几乎DM的所有段落中均具有FILE=c，以下除非专门指明，它都表示内部工作文件。

1. READ 段：读取系统文件。SFILE = c. 输入文件名。
   REWIND. 读取数据前重绕(rewind) 输入文件。
   FORMAT= 'c'. 输入记录格式。
   VNAME = list. 变量名。
   VARIAB= #. 变量数。
   RLABEL= v1,v2. 记录标号。
   BLANK = ZERO|MISS. 使用Fortran 类型格式时，空格的处理方法。
   MCHAR = c. 数据中的缺失值。
   CODE = c. BMDP 文件码(文件中的记录类型)。
   CONTENT= c. BMDP 文件内容。
   LABEL = 'c'. BMDP 文件标号。
   KEEP = list. 保持变量(文件中的一种记录类型)。
   DELETE= list. 删除变量(文件中的一种记录类型)。
   NEWNAME= list. 从BMDP 文件读取变量的新名.
   RECT = list. 字母数字记录类型标识。
   RECN = list. 数值类型标识。
   RECID = #1,#2. 记录标识为输入记录#1 到#2 间的字符。
   LEVEL = list. 记录类型的层次水平。
   注：使用FORMAT(t)=, VNAME(t)=, 读取多种类型的记录。

2. SORT 段：对数据文件的记录排序。
   KEY = list. 排序关键字。
   ORDER = list. 用A, D, 或C, 指示记录按排序关键字进行升序、降序、或字符类型排序。
   KEEP = list. 生成排序文件中保持的变量名。
   DELETE = list. 删除变量。
   NEWNAME= list. 排序文件变量名。
   NEWFILE= c. 输出工作文件，未指定时使用输入文件名。

3. EXTRACT 段：抽取记录和变量。
   RECT = list. 抽取的记录类型。
   KEEP(t)= list. 从类型t 中抽取的变量名。
   DEL(t) = list. 从类型t 中删除的变量名。
   NEWN(t)= list. 类型为t 的保持变量的新名。
   NEWFILE= c. 输出工作文件名。

4. CHECK 段：标记缺失值和超出指定围的值，也用于经'hot-deck' 过程填充缺失值。
   MISS = list. 每个变量的缺失值。
   MIN = list. 每个变量的最小值。
   MAX = list. 每个变量的最大值。
   HOTD = list. hot-deck 过程工作的变量。
   注：使用MISS(t)=, MIN(t)=, 等处理多种记录类型。

5. GROUP 段：指示变量分组，每种记录类型分别使用。
   RECT = list. 记录类型。
   CODE(v)= list. 变量代码。
   CUTP(v)= list. 变量分隔点。
   NAME(v)= list. 变量的分类名。
   FROMF = c. 指定含有分组信息的文件名。
   FROMV = list. 含有分组信息的FROMF 文件变量。
   VARIAB = list. 接受分组信息的变量名。

6. TRANSFORM 段：每次对一个记录有效。
   RECT = c. 转换记录类型。
   KEEP = list. 保持变量。
   DELETE = list. 删除变量。
   NEWNAME= list. 保持变量新名。
   NEWFILE= c. 输出工作文件名。
   RETAIN. 记录之间保持新变量的值，
   影响转换的语句具有形式：
   v = 表达式。
   IF (逻辑条件) THEN (操作语句.) ELSE (语句.) UNDEFINED (语句.).
   FOR v = list DO (语句.).
   WHILE (l逻辑变量) DO (语句.).
   SHOW (变量值、变量、或表达式).
   TEXT (显示文本).

7. MERGE 段：两个或多个文件的联接。
   FILES = list. 合并的工作文件名。
   KEY(f) = list. 文件f 中引导合并的变量。
   ORDER = list. A, D, 或C, 的列表，指示关键字是升序、降序或字符顺序。
   KEEP(f)= list. 文件f 保持的变量。
   DEL(f) = list. 文件f 删除的变量。
   NEWNAME= list. 合并生成文件的变量名。
   NEWFILE= c. 输出工作文件名，不指明时为第一个输入文件。

STOP = list.	若相同的关键变量值在多于一个文件内找到，用这些指示
FIRST = list.	指明保持重复的第一个、最后一个、所有的或不保持。
LAST = list.	
ALL = list.	List: 是字符串，每个的长度= 文件数。
NONE = list.	如：FIRST='.B.D','..CB'. 表明若记录在第二个和第四个
UPDATE= list.	找到，则保持第二个，若在第三个和第四个找到，则保持
PRINT = list.	第三个。

8. JOIN 段：两个或多个文件记录的合并。
   FILES = list. 工作文件名。
   KEY(f)= list. 引导合并的变量名。
   ORDER = list. A, D, 或C 的列表，指示合并顺序。
   KEEP(f)= list. 保持变量名。
   DEL(f) = list. 删除变量名。
   NEWNAME= list. 合并文件变量名。
   NEWFILE= c. 输出工作文件名，不指明为第一个工作文件。

PAD = list.	若指定在文件中不出现，HOTREC 和HOTKEY 使用前一个记
DROP = list.	录和符合条件的记录来代替缺失值。
HOTREC= list.	
HOTKEY= list.	List: 字符串；每个长度与文件数相同。
STOP = list.	如: STOP='.BC'. 表明第一个文件出现缺失时停止。
PRINT = list.	

9. AGGREGATE 段：把记录归并到新文件。
   WITHIN = list. 按WITHIN 变量分组归并(输入文件应排序)，或：
   RECT = c. 类型为c 的记录被归并。
   MAXCOPY= #. 每个归并集合的最多记录数(默认20)。
   KEEP = list. 保持变量名。
   DELETE = list. 删除变量名。
   NEWNAME= list. 保持变量的新名。
   NEWFILE= c. 输出工作文件名。
   RETAIN. 保持新变量值。
   APPEND. 追加结果到每个集合。
   影响归并的语句为：
   v = 表达式。
   IF (逻辑条件) THEN (执行语句.) ELSE (语句.) UNDEFINED (语句.).
   FOR v = list DO (语句.).
   WHILE (逻辑变量) DO (语句.).
   SHOW (值、变量或表达式).
   TEXT (要显示的文本).

10. PACK 段：从记录的集合中选出一个变量。
    WITHIN = list. 按WITHIN 的分组压缩(输入文件应排序)，或：

RECT = c. 类型为c的记录被压缩。
VARIAB = list. 待压缩的变量。
KEYS = list. 引导压缩的变量。
LEVEL = list. 关键变量的水平数。
CODE(v)= list. 关键变量的编码。
CUTP(v)= list. 关键变量的间隔。
PICK = list. 对CODES 使用EXACT, CLOSEST, BELOW, 或ABOVE。
对CUTPOINTS 使用FIRST, LAST, SMALL, 或LARGE。
MAXPAD = #. 压缩的最大记录数。
KEEP = list. 除了压缩变量外保持的变量。
NEWNAME= list. 压缩文件变量新名。
NEWFILE= c. 输出工作文件名，默认为第一个文件名。

11. UNPACK 段：把记录分解到几个记录。
VARIAB = list. 操作变量名。
SEQU = list. 产生的case sequencing 变量名。
LEVEL = list. case sequencing 水平数，或
CODE(v)= list. case sequencing 变量v 的值。
KEEP = list. 除unpacked和sequencing变量外拷贝到每个未压缩记录的变量。
NEWNAME=list. 在未压缩文件中的变量名。
NEWFILE= c. 输出工作文件名。未声明时为输入文件名。

12. PRINT 段：打印数据、文件名和内容、软件信息，亦用于控制行、页的大小和输出多少。
NAMES. 显示工作文件名、记录数、记录类型、变量名。
NEWS. 打印程序的限制及错误。
PAGE = #. 每页行数。
LINEs = #. 每行字符数。
LEVEL = c. 详细程度: MINIMAL, BRIEF, NORMAL, or VERBOSE.
POINTERS. 显示文件名、记录类型、变量名、纠错中的文件指针。
以下8 个参数用于打印数据：
FILE = c. 工作文件名。
VAR = list. 打印的变量名或指标，可以使用VAR=ALL。
FIELD = list. 打印变量的域宽。
FORMAT= 'c'. 变量的打印格式。
NUMBER= #. 每页打印的记录数。
CASES = #. 打印记录数。
HEAD = 'c'. 每页首打印的标题。
RECT = c. 记录类型名(非方形数据文件)

13. MAP 文件：打印映象，即文件中记录的结构。

WITHIN= list. 同一组记录在一条线上画出(输入文件应排序)。
VARIAB= v. 使用1,2,...,9,A,B,...,Z 编码的变量。

TIME = v. 水平轴画的变量。
DELTA = #. 水平轴的字符增量。
RANGE =#1,#2. 水平轴范围。
CUTP = list. 定义编　码的间隔值。
CODE = list. 定义编码。
SYMB = list. 代换1,2,...,9,A,B,...,Z 的记号。
RECT = c. 显示的记录类型。

14. STATISTICS 段：报告变量的统计量，可以针对部分记录。
WITHIN= list. WITHIN 变量分组(输入文件应选排序)。
VARIAB= list. 需要显示统计量的变量。
CELLWISE. 记录的所有统计量同时报告，NO CELLW 则是每变量一组。
RECT = c. 显示统计量的记录类型，参数仅用于含有多种记录类型的文件，对这种类型的每个连续记录报告统计量。

15. HISTOGRAM 段：打印变量直方图，可以针对部分记录。
WITHIN= list. WITHIN 变量分组直方图(输入文件应先排序)，或：
RECT = c. 类型c 的记录集合组成一个直方图。
VARIAB= list. 产生直方图的变量。
MIN = list. 每个变量的最小尺度。
MAX = list. 每个变量的最大尺度。

16. SAVE 段：存贮方形文件到BMDP 文件、FORMATTED 或二进制文件。

FILE = c. 存贮文件名。
SFILE = c. 操作系统认可的文件名，默认值为FILE 的文件名。
NEW. 从SFILE 中删除BMDP 文件。
CODE = c. BMDP 文件码，未指定时为FILE 的名字。
LABEL = 'c'. BMDP 文件标号，至多40 字符。
KEEP = list. 存贮的变量。
DELETE= list. 删除的变量。
FORMAT= 'c'. 输出格式或BINARY 字，默认为BMDP 文件。
RECT = c. 记录存贮类型。

17. DELETE 段：删除一个或多个工作文件。
FILE = list. 删除文件名，该选项不删除操作系统下的外部文件。

18. END 段：删除所有工作文件，但不终止程序。

19. FINISH 段：终止程序并且把控制返回到系统。

20. CONTROL 段：控制程序执行环境并且为程序错误提供诊断信息。
INTERACT. 设定执行状态为交互式，NO INTERA 则相反。
FILE = c. 程序指令从名为c 的文件读取。文件结束后返回。
MACRO = c. 宏文件名。
ERROR = c. 程序停止的水准NONE, INTERACT, NORMAL, 或STRINGENT。
DUMP. 批处理方式下，打印完整的BMDP 存贮显示，在交互方式下，显示选择的数组。
LENGTH= #. 所使用存贮区的长度，减少存贮区的大小可节省DEBUG=TEST或INFO下CPU 的时间。
DEBUG = NORM. 不做特殊纠错。
TRACE. 报告子程序进入/退出信息。
TEST. 用期性检查内存。
INFO. 检查内存，空间分配、GETME 调用以及子程序跟踪情况。

把第 6 章例6.4的程序增加SAVE段，指定存于文件2L，编码为2L，程序为：

```
save code='2L'. content=data. new. file is '2L'. code='2L'./
```

现转入DM模块，用READ段读取，指定工作文件为UU：

```
read sfile='2L.'. file='uu'. code='2L'./
```

使用PRINT段浏览其内容：

```
print head='The Data of Example 5.3'. names. pages=60. lines=72. var=all. /
```

利用SORT段对数据排序，并使用MAP段：

```
sort key=group, survival. /
map variables=survival. within=group. /
```

利用STATISTICS段求取描述统计量：

```
statistics variables=survival. within=group. /
```

利用HISTOGRAM段绘直方图：

```
histogram variables=survival. within=group. /
```

这里也举一个读取矩阵数据的例子[3]，程序如下：

```
problem   title is 'BMDP4M'./
input     type is correlation.
          shape is square.
          variables are 12.
          format is '(12F5.0)'./
variables names are le,aso,in,..../
plot      initital is 0.
          final is 0.
          fscore is 0.
```

```
factor    method is pf.
          constant is 0.
          iterate is 25.
          commun is smcs.
rotate    method is vmax.
          normal./
end/
```

...数据矩阵...

程序调用4M模块进行因子分析，同一般程序一样，分为两部分。第一部分是数据的录入，内容有变量数、变量名、数据的形式、数据的格式。第二部分是进行因子分析，包括因子的抽取方法、因子数目、禄始公因子方差的选择、旋转方法。

数据的类型有DATA,CORR,COVB,LOAD,FSCF几种， 形状有方SQUARE,LOWER两种，格式可以是固定的或自由(FREE)的。因子分析方法有PC,PF,ML,LJ几种，公因子方差除了SMCS以外有UN,SM,MAX几种，旋转方法有VMAX,NONE,QRMAX, EQM, DQ, DOBL ,ORTHOG,ORHTOB几种。

§16.3.4 SYSTAT

源于SYSTAT 模块化的特点，SYSTAT 专用DATA 模块进行数据管理并为后续的分析作准备，SYSTAT 的语言特色也在该模块体现得最好，如一系列统计函数和类似BASIC 语言的语句。运行后在系统指示下打入EDIT，即进入全屏幕编辑方式。SYSTAT 4.0 提供了专门的EDIT 模块进行全屏幕数据管理，在DATA/EDIT 模块经SWITCH TO 命令转入其它分析模块。实际上，不同模块间可经此命令相互切换。下面是一个全屏幕编辑运行的示意图，在DATA 块中打入命令USE IRIS 和EDIT，系统进入编辑状态。第一列是例号，后面几列是对应各变量的观察值。

SYSTAT Editor IRIS.SYS

Case	SPECIES	SEPALLEN	SEPALWID	PETALLEN	PETALWID
142	3.000	6.900	3.100	5.100	2.300
143	3.000	5.800	2.700	5.100	1.900
144	3.000	6.800	3.200	5.900	2.300
145	3.000	6.700	3.300	5.700	2.500
146	3.000	6.700	3.000	5.200	2.300
147	3.000	6.300	2.500	5.000	1.900
148	3.000	6.500	3.000	5.200	2.000
149	3.000	6.200	3.400	5.400	2.300
150	3.000	5.900	3.000	5.100	1.800
151					

编辑时可用ESC 键进行命令行与数据录入间的切换。输入的变量为字串类型时，变量名前应导以单引号(')。在命令行打入HELP有下面的菜单提示。

EDIT 产生和编辑SYSTAT 文件。

光标命令以及等效功能键(alternative keys) 有:

←(Cntl-S)	→(Cntl-D)	↓(Cntl-X)
Ins (page left, Cntl-A)	Del (page right, Cntl-F)	PgDn (Cntl-C)
PgUp (Cntl-R)	Home (Cntl-W)	End (Cntl-Z)

编辑(EDIT) 命令有:

Esc (Cntl-Q)	进行数据窗与命令行间的切换。
USE <file name>	把数据区用SYSTAT 文件填充。
SAVE <file name>	把工作区存入一个SYSTAT 文件。
FEDIT <filename> |*| > |#|	启用SYSTAT 的文本编辑器。
FIND <expression>	把光标移至被选定的观察号。
FORMAT <#>	设定显示的小数位数。
FPATH <path>/GET|OUTPUT|SAVE| SUBMIT|USE|FEDIT|TRANSFER	给指定的文件设定路径前缀。
LET <var>=<expression>	转换或生成变量。
IF <expression> THEN LET <var>=<expression>	条件转换。
REPEAT <#>	把工作区填充至指定数目<#> 的观察。
TYPE <type of matrix>	指示CORR, COVARIANCE 等类型。
HELP <command>	提示信息。
NEW	清除工作区以供新数据集使用。
DOS 'DOS command'	执行一个MS-DOS 或PC-DOS 命令。
SWITCHTO 'module'[<file>/ECHO]	切换至另一个SYSTAT 模块。
END or QUIT	返回DOS

系统文件的左右合并是通过USE 命令来完成的，要进行这种合并，只需同时指示几个文件名及其相应的变量。SYSTAT 以PUT/GET 命令存/取一个ASCII 文件。GET 在读入一个文件时，要求首先要运行SAVE 命令指示要存贮的系统文件名。当读入ASCII 数据宽度超常时，应用LRECL= 命令指示记录的宽度。SYSTAT 活动数据的转置也用TRANSPOSE 命令。

SYSTAT 使用IMPORT 命令把一个外部文件转换成SYSTAT 文件，其句法是：

```
IMPORT <file> [(<var1>,<...>)] /,
 TYPE= LOTUS | LOTUS2 | SYMPHONY | SYMPHONY11 | DBASE2 | DBASE3 | DIF |
 MAP | PORTABLE [RANGE=<range>] [ROWS=<#>-<#>]
```

如：IMPORT 'MYFILE.WK1' / TYPE=LOTUS2 ROWS=1-50　意为把Lotus 1-2-3 第二版的文件MYFILE.WK1 转换成SYSTAT 文件，仅仅使用1-50 行的内容。

EXPORT 命令把一个SYSTAT文件转成其它格式的文件，其句法是：

```
EXPORT <file> [(<var1>,<...>)] /,
 TYPE= LOTUS | LOTUS2 | SYMPHONY | SYMPHONY11 | DBASE2 | DBASE3 | DIF |
 MAP | PORTABLE [ROWS=<#>-<#>]
```

如：EXPORT LOTUSFIL / TYPE=LOTUS2 ROWS= 1- 50　由内存的文件生成一个Lotus1-2-3 文件。

微机SYSTAT 的一个特点是提供了PC 与Macintosh 机间SYSTAT 文件的转换功能。

§16.3.5 Stata

Stata 的系统文件以.DTA 作为扩展名，这类文件存贮了数据的格式、标签等。

命令use 从磁盘调一个Stata 格式的数据到内存，其格式为：

use 文件名[, clear nolabel]

clear 允许所有情况下调入内存，不论内存改动的数据是否已存盘。

nolabel 不允许存贮的数据中的标号被调入。

使用describe using 文件名可以浏览文件的内容。

存贮Stata 格式数据的命令是：

save filename [, replace nolabel]

replace 允许覆盖已存在的数据集。

nolabel 省略数据集中的村标号。

文件扩展名不指定时，用.dta 或者.xp，此时文件含有一个交叉乘积矩阵。

如：save myfile

File myfile.dta already exists

r(602) ;

系统报错，增加选项replace，命令为：save myfile, replace

存贮的数据将以压缩二进制格式存放。

Stata 的.DCT 文件包含一个数据字典，它描述了文件所含的变量及其格式、标号，以及数据存放方式等，数据可以在这些描述的下面或者放在其它文件，但它仍然是一个ASCII 文件。Stata 读取该格式的数据时需要指示dictionary 选项。其它软件可以以固定格式或自由格式读取。Stata 专门用.RAW 扩展名指示ASCII 格式的数据文件，infile/outfile 命令读取/存贮ASCII格式的数据文件。infile 的格式是：

infile [变量表[_skip[(#)] [变量表[_skip[(#)] ..]]]] using 文件名[in 范围] [if 表达式] [, automatic byvariable(#)]

文件默认具有.RAW 扩展名。指定变量列表时，文件中所含数据是自由格或用逗号分开的。若不指示变量列表，则文件中含有一个数据字典。若文件扩展名未指示，则隐含使用.DCT。

执行infile 命令时，内存中不应该有数据，这可以预先执行drop _all 命令。也可参照help maxvar 给出的说明。

现有数据文件为myfile.raw，内容为

1 2 3 1, 2 3

4 5 6 或4,5 或1 2 3 4,5 6

6

三个变量读入时赋为A、B、C，则可用命令: infile a b c using myfile

变量列表中使用_skip可以跳过一些量，如：

infile a _skip c using myfile

infile _skip(2) c or infile _skip _skip c

第一句只读变量a 和c，而第二句则只读变量C。

infile str20 name age sex using myfile

infile str20(name age) sex using myfile

infile str20 name age int sex using myfile

第一句中读入量name 为长度为20 的字串，age 和sex 为浮点数。第二句中name 和sex 为字串而sex 为浮点数，第三句中name 为字串，age 为点数，sex 为整数。

infile 也可以读入非数值类型的变量并产生数值类型标号，可用automatic 来做到，如对数据：

”James Smith” 38 ”male”

Branton 32 female

”Bill Ross” 27,’male’

用命令：infile str20 name age int sex:sexlbl using myfile, automatic

整型量sex 对于male 将取值为0，对于female 则取值为1。

与infile 相仿，outfile 则是把数据以ASCII 码形式写入磁盘，语法为：

outfile [变量表] using 文件名[if 表达式] [in 范围] [, comma dictionary nolabel replace]

最后的选项指定Stata 生成用逗号分隔或字典格式文件，nolabel以数值记录标号变量的值。

Stata 管理数据有以下约定：

. 字串总是用双引号括起。

. 除comma 以外的所有格式，数据均以表的形式存贮。

. outfile 的行不超于80 个字符，因而一个观察可以点数据文件的几行上。

. 在comma 格式中，数值缺失值记作”,,”，否则以圆点”.”存贮。

. 所有格式的缺失字串均以双引号(””) 记录。

在Stata内部录入数据之前，消除已调入的数据，使用命令drop _all 和label drop _all 消除内存数据和标号。录入数据只消使用命令input 变量列表，如：

```
. input id mpg weight price
           id        mpg     weight      price
  1. 1 22 2930 4099
  2. end
```

每次录入以end 结束，一旦内存数据生成，继续录入使用input 即可。

```
. input
          id        mpg     weight      price

2. 2 17 3350 4749
3. 3 22 2640 3799
4. 4 20 3250 4816
5. 5 15 4080 7827
6. end
```

录入字符量时应施以前缀str#，#的取值为2 至80。因为未声明时，默认为float，如：

```
. drop _all
. input str14 make mpg weight price
```

大批量数据这样的录入仍然很繁琐，使用infile 读入ASCII 数据较方便。

对DOS 用户来说，Stata 可以读入Lotus, Symphony, dBase, Gauss, SPSS，或SYSTAT 格式的数据，这个工具称为Stat/Transfer，它是一个菜单驱动的模块。

§16.3.6 DBMS/COPY

可以转换的数据类型有：Lotus, Quatro, Clipper, Database, Smart, ASCII, ACT!, Datalex, ABstat, Bass, BMDP, CSS, 4CasT/2, Forecastpro,Microstat -II, NCSS, Probe, RATF, SigmaPlot, StatGraphics, SyGraph, Excel, Autobox, Gauss, GLIM, Minitab, SAS, SCA, Soritex, SPSS, Stata, Statpac, SYSTAT，S-Plus等也就是说，本书涉及的多数软件可经它转换。该软件使用方便，在ASCII 转至其它有格式文件时也生成一个数据字典。

以SPSS/PC+为例，它可以调用DBMS/COPY进行与其它软件包间的数据交换。

```
DBMSCOPY FROM ' ' TO ' '.
PLOT /plot y with x.
NPPlot/ variables x.
QED
SET BOX='-|-++++....'.
```

公司的地址：P.O.Box 56627, Houston, TX 77256, 电话(800) Stat-wow即(800)7827-969。

§16.4 数据交换用例

不同计算机系统和软件间数据与程序的交换交换受许多因素的制约，考虑经过网络传输时，要对计算机网络有所了解，如DOS与VMS系统间的交换，常用的途径有：

```
. Pathworks (PCSA), DEC
. PC-NFS, SUN Microsystem
. DECNET, DEC
. Kermit, Columbia University (treeware)
```

对于DOS和UNIX之间的交换，常用PC-NFS、TCP/IP和Kermit。

§16.4.1 程序交换用例

【例16.6】VAX/VMS SAS 样本程序的使用

VAX/VMS SAS 6.07 提供了许多.SAS 样本文件，通过网络传输至PC 机时，由于其仅仅使用换行符，没有硬回车，致使大多数PC 机的编辑软件不能调用，也不能在PC SAS 下调入，这时可以使用DOS 5.0 中的编辑EDIT，也可以采用下面的BASIC 程序进行转换：

```
INPUT "请输入文件名";INP$
INPUT "请输出文件名";OUT$
OPEN INP$ FOR INPUT AS #1
OPEN INP$ FOR OUTPUT AS #2
WHILE NOT EOF(1)
   A$=INPUT$(1,1)
   IF A$=CHR$(10) THEN PRINT #2, ELSE PRINT #2,A$;
WEND
CLOSE #1,#2
END
```

运行时指定VAX/VMS SAS 样本程序为源文件，指定PC 机文件名为目标文件。程序的处理办法是把换行符换成DOS 下的回车换行符，这样一处理，就可以在微机上正常编辑使用了。第三章介绍的DOS2UNIX/UNIX2DOS功能与之类似。

【例16.7】VAX/VMS 下SAS 对BMDP 的调用

由于BMDP 模块化的特性，使得在SAS 内启用BMDP 很方便。下面是SAS 用户手册上的用例，建立数据集，启用BMDP 进行分析，产成结果用CONVERT 过程转入SAS。

```
DATA TEMP;
   INPUT A B C@@; CARDS;
   1 2 3 4 5 6 7 8 9
PROC CONTENTS;
   TITLE 'CONTENTS OF SAS DATA SET TO BE RUN THROUGH BMDP1D';
PROC BMDP PROG=BMDP1D DATA=TEMP;
   PARMCARDS; /* 指示 BMDP 语句引用开始 */
   /PROB TITLE='SHOW SAS/BMDP INTERFACE'.
   /INPUT UNIT=3. CODE='TEMP'.
   /SAVE CODE='BOUT'. NEW. UNIT=4.
   /END
   /FINISH
;
PROC CONVERT BMDP=FT04F001 OUT=FROMBMDP;
PROC CONTEMTS;
   TITLE 'SAS DATA SET CONVERTED FROM BMDP SAVE FILE';
PROC PRINT;
```

§16.4.2 数据交换用例

【例16.8】用SAS/RTERM 进行PC 与VAX/VMS SAS 数据交换

卫生部进行1991 年全国医院行业“纠风”的调查时，对门诊病人、住院病人或出院病人进行询问，以了解不同级别医院、不同科别的病人就医时对医生、护士的满意情况，同时看病人在医疗福利类型的影响情况，共调查 2 万余例。在dBASE lll数据文件约7 兆，生成SAS数据文件约10 兆，在微机上制表太费时间，此时拟转至VAX 机上完成。

利用PROC DOWNLOAD 过程. 在AUTOEXEC.SAS 中已用语句filename rlink 'd:\sas\saslink\logvms.scr'; 进入SAS 系统，在PGM 窗口命令行上打入: SIGNON，据提示输入帐户名和口令进行登录。然后远程提交(rsubmit) 程序。

```
filename pc 'd:\dBASE3\bank.dbf';
proc dbf db3=pc out=bank;
run;
DM 'rsubmit';
libname user '[]';
proc upload data=bank out=user.file;
run;
```

则把数据库文件传至VAX 机，数据库的格式亦完整地由PC 过录到VAX 机。由于计算机网络的发达，使用SAS的传输格式更为方便：

```
libname us xport 'us.tds';
libname counties xport 'counties.tds';
proc copy in=maps out=us mtype=data;
  select us;
run;
proc copy in=maps out=counties mtype=data;
  select counties;
run;
```

将SAS/GRAPH中maps的图形数据集转为传输格式，在主机上使用类似的语句转为SAS数据集。

【例16.9】SPSS/PC+ 到VAX/VMS SAS 数据的转用

北京阜外医院心外科拟进行心脏瓣膜移植的研究，该医院拥有SPSS/PC+ 软件。由于被分析的变量和生成的变量数目很大，超过了128 个，故不用dBASE Ⅲ的格式存放而改用SPSS/PC+ 格式存放，为了能在VAX/VMS SAS 上使用，首先把SPSS/PC+ 格式换成小型机上SPSS-X格式，使用EXPORT 命令，文件经DECnet 直接拷贝到VAX 机。然后于VAX 机重新登录，运行SAS 软件和启用转换的数据集。

由上例的做法得到启示，1992 年医院满意度调查分析，直接经SPSS/PC+ 进行转换，速度可以改善。程序如下：

```
SET /MORE OFF /LISTING='D:\FOX\BANK.LOG'.
trans from '\fox\bank.dbf' /TYPE DB3.
EXPORT /OUTFILE '\fox\bank.sys' /MAP.
EXIT.
```

把程序运行情况存于BANK.LOG，第二句把BANK.DBF 转成SPSS/PC+ 文件，第三句进行转换，结果生成bank.sys，转换时用MAP 选项列出转换信息。仍经DECnet 网把bank.sys 传至VAX。VAX 上的文件转换程序为：

```
filename user 'bank.sys';
proc convert SPSS=user out=data1;
run;
libname user '[]';
data user.file;
set work.data1;
run;
```

启用专用过程PROC CONVERT，程序运行结果是在VAX 机当前目录下生成一个名为FILE的SAS 系统文件。

§16.4.3 综合用例

【例16.10】1993 年卫生部行业纠风调查的数据处理是比较典型的，现加以介绍。调查分

为几个步骤，正如第二章介绍的那样，首先确定调查时间、对象、内容，调查一览表与病人问卷表等。

医院抽法：拟按部级、省级和地市级(计划单列市)几个水平。因为分层后可能不足，最终进行个别调整。

抽样框架的确定：利用卫生单位代码库作为原始抽样框(FRAME) ，它是一个dBASE格式的数据库文件，含有各单位的行政区划代码、相应的医院情况信息等，这样就能利用dBASE或FoxBASE+中的SET FILT TO 命令计算出各种条件下医院的总数，其信息可用SET ALTE TO 命令和? 命令到文本文件中，设计数行用*** 作标记，则用BASIC 程序读取之，结合随机函数将随机号与流水号以及医院名称等有关信息同量连续列出，设列表文件为SAMPLED.TXT，BASIC程序如下：

```
OPEN "I", #1, "sampled.txt"
OPEN "O", #2, "result"
RANDOMIZE TIMER
DO WHILE NOT EOF(1)
    LINE INPUT #1, line$
    IF INSTR(line$, "***") <> 0 THEN
       num = VAL(LEFT$(line$, 10))
       print #2, line$
    ENDIF
    i = 1
    WHILE i <= num
       LINE INPUT #1, line$
       index$ = STR$(i)
       sel$ = STR$(INT(RND * num) + 1)
       line$ = index$ + SPACE$(5 - LEN(index$)) + sel$ +
                SPACE$(5 - LEN(sel$)) + line$
       PRINT #2, line$
       i = i + 1
    WEND
    LOOP
END
```

随机数的种子是当前时间(TIMER)，随机数范围随医院的数目而定。

抽样框的形式为：

⋆⋆⋆⋆ 医院数目

随机号	流水号	行政区划码	医院名称	床位数	.....
xxxxxx	xxxxxx	xxxxxxxxxx	xxxxxxxx	xxxxxx	xxxxxx

从任意一个随机号开始，读取几个随机号，其内容到对应的流水号中查取即得到抽中的医院号，这样做法也避免了每年编程抽取的麻烦。

编程与发盘。91 年、92 年采用dBASEIII/FoxBASE+ 程序，当时EPI INFO 还没有汉化，而且部分省市没有286 以上计算机，其CGA或MDA显示器只能显示10 行汉字。进行此选择是比较合适的。93 年度由于已在几个调查中使用，所有省市卫生厅已配备286计算机或具备使

用EGA/VGA 显示器、25 行汉字的能力，采用汉化EPI INFO 是必要的，也考虑今后对该软件的进一步推广应用。利用原始调查表，制做.QES 文件，为后继SAS 软件的处理，变量名大多使用英文，放在大括号内，设其文件名为SURV.QES，其内容见【例15.3】。

以上程序下发的同时，还提供了往年数据操作处理的样本程序，以及相应的EPI INFO 分析模块，这样在地方水平上也能够方便产出。

使用ENTER 录入，生成数据文件SURV. REC，则可用于生成dBASE 或FoxBASE+ 文件，或SAS、SPSS 文件了。由于全国单位较多，可将转换过程编入DOS 批处理文件CONV.BAT，其内容如下：

```
REM 本程序用于将EPI INFO 文件转成DBASEIII。
REM DOS 文件名至多有八个字符，故有一些省市名不完全。

convert ANHUI    ANHUI    8 Y
convert BEIJING  BEIJING  8 Y
convert FUJIAN   FUJIAN   8 Y
convert GANSU    GANSU    8 Y
convert GUANGDON GUANGDON 8 Y
convert GUANGXI  GUANGXI  8 Y
convert GUIZHOU  GUIZHOU  8 Y
convert HAINAN   HAINAN   8 Y
convert HEBEI    HEBEI    8 Y
convert HEILONGJ HEILONGJ 8 Y
convert HENAN    HENAN    8 Y
convert HUBEI    HUBEI    8 Y
convert HUNAN    HUNAN    8 Y
convert JIANGSU  JIANGSU  8 Y
convert JIANGXI  JIANGXI  8 Y
convert JILIN    JILIN    8 Y
convert NEIMENG  NEIMENG  8 Y
convert NINGXIA  NINGXIA  8 Y
convert QINGHAI  QINGHAI  8 Y
convert SHANDONG SHANDONG 8 Y
convert SHANGHAI SHANGHAI 8 Y
convert SHAANXI  SHAANXI  8 Y
convert SHANXI   SHANXI   8 Y
convert SICHUAN  SICHUAN  8 Y
convert TIANJIN  TIANJIN  8 Y
convert XINJIANG XINJIANG 8 Y
convert YUNNAN   YUNNAN   8 Y
convert ZHEJIANG ZHEJIANG 8 Y
convert LIAONING LIAONING 8 Y

REM 完成!!!
```

由于各省的数据库中没有省名信息，拟最终合并时加入，考虑到EPI INFO 虽然录入方

便，但它对数据细处的操作不如dBASEIII或FoxBASE+，因此仍用后者编程解决。为了处理的方便，将各省报盘相同的文件名SURV.REC (或其它改动后的文件名) 拷入子目录时即改名为相应的省名，如：

COPY A:SURV.REC BEIJING.REC

这样前面的格式转换程序和下面的数据合并程序都可以自动完成。

合并程序由两部分完成，即对各省调用替换、追加和具体追加和替换，对应的文件是ASSE.PRG和REPL.PRG，其内容为：

⋆ ASSE.PRG

⋆⋆⋆ 把现有数据合并起来

```
set safe off
set echo off
set talk off
sele 1
use
sele 2
use
use bank
zap
set safe on
do repl with 'ANHUI'
do repl with 'BEIJING'
do repl with 'FUJIAN'
do repl with 'GANSU'
do repl with 'GUANGDON'
do repl with 'GUANGXI'
do repl with 'GUIZHOU'
do repl with 'HAINAN'
do repl with 'HEBEI'
do repl with 'HEILONGJ'
do repl with 'HENAN'
do repl with 'HUBEI'
do repl with 'HUNAN'
do repl with 'JIANGSU'
do repl with 'JIANGXI'
do repl with 'JILIN'
do repl with 'NEIMENG'
do repl with 'NINGXIA'
do repl with 'QINGHAI'
do repl with 'SHANDONG'
do repl with 'SHANGHAI'
do repl with 'SHAANXI'
```

```
do repl with 'SHANXI'
do repl with 'SICHUAN'
do repl with 'TIANJIN'
do repl with 'XINJIANG'
do repl with 'YUNNAN'
do repl with 'ZHEJIANG'
do repl with 'LIAONING'
set talk on
retu
```

程序首先把数据库BANK.DBF的内容清空，用SET SAFE OFF 指定操作为自动。注意数据库BANK 可由空的SURV.REC 文件经CONVERT 而来，但必须事先在dBASE 或FoxBASE+下运行了MODI STRU 命令以增加字段PROV 用于存贮省名。

```
* REPL.PRG
** 用于进行全国数据汇总
parameter name
if .not.file("&name..dbf")
   retu
endi
sele 1
use &name
num=recc()
use
sele 2
use bank
appe from &name
skip -num+1
repl next num prov with '&name'
retu
```

REPL.PRG 的操作有两部分，即在第一个区内对各省数据库&NAME 记录计数，第二部分在第二区对已追加到BANK.DBF 参加计数的记录进行PROV变量赋值。

设PC SAS 上的AUTOEXEC.SAS 文件内容为：

```
filename rlink 'd:\sas\saslink\decnet.scr';
options remote=cterma;
run;
```

SAS/BASE 中的SASZRLNK.EXE 应进行替换。

在DOS 下设定环境变量CTERMA，如设VAX 主机节点名为VAX1，则设法很简单，使用DOS 命令：SET CTERMA=VAX1 即可，设定后可单独打SET 命令确实一下，此步可参照CTERM 中的READ.ME 进行。

运行STARTNET.BAT 上DECnet 网，进入SAS，并在命令行打入命令：

SIGNON

系统提示账户和口令，系统自动进行远程调用进入VAX/VMS SAS，可以使用远程提交命令RSUBMIT 了，为了简便，上述过程可放在SAS 中的显示管理命令DM 中，转贮数据集的程序为：

```
libname local '.';
filename bank 'd:\surv\bank.dbf';
proc dbf db3=bank out=local.bank;
run;
dm 'rsubmit';
libname remote '[]';
proc upload data=local.file out=remote.file;
run;
```

BANK.DBF 是全国数据库，用PC SAS 转成SAS数据集放在LOCAL库名即SAS 子目录下，经VAX主机SAS PROC UPLOAD 完成数据转贮。

制表分析程序为：

```
/*********************************************
标题：医院满意度调查分地区、分省份描述分析
作者：卫生部卫生统计信息中心
日期：1994 年1 月

产品：SAS/BASE
过程：DBF、FORMAT、DATASETS、TABULATE
*********************************************/


title1 '1993 年度纠正医院不正之风调查汇总表';
proc printto print='d:result' new;
run;
dm 'rsubmit';
libname remote '[]';
options ls=130 ps=300 nodate formchar=' ----------';
options missing=' ' nocenter mprint;
proc format;
     value yesno 1='是'      2='否';
     value ssfmt 1='门诊'    2='住院'     3=' 出院';
     value  i    1='省级'    2='地市级'   3=' 区县级';
     value ii    1='内科'    2='外科'     3=' 妇产科'
                 4='儿科'    5='中医科'   6=' 眼科'
                 7='耳鼻喉科'8='皮肤科'   9=' 口腔科' 10='其它';
     value iii   1='工人'    2='农民'     3=' 军人'
                 4='教师等'  5='干部'
```

```
                6='个体'    7='商业服务'
                8='离退休' 9='无职业者' 10='学生'   11='其他';
value iv        1='公费'    2='劳保'      3='半自费'
                4='自费'    5='商业性医疗保险';
value sat       1,2='满意较满意'    3='一般'
                4,5='不满意很不满意'6='没接触';
value vi        1,2='好较好'        3='一般'
                4,5='不好很不好'    6='说不好';
value viii      1,2='好较好'        3='一般'
                4,5='不好很不好'    6='没接触';
value ix        1,2='信任较信任'    3,4=' 不信任很不信任'
                5='说不好';
value value 1='50 元以下'       2='50-'3='100-'
                4='200-'            5='500 以上';
value for       1='出于感激'        2=' 想得到方便和照顾'
                3='担心不认真看病' 4='受人影响' 5='暗示的'
                6='直接索要';
value act       1,2='拒绝收和事后退还' 3='照价付了钱'
                4='付部分钱'          5,6=' 推辞过和没有拒绝';
value $prov '安徽省'
                'Abeijing'  ,'BEIJING'='北京市'
                'Mfujian'   ,'FUJIAN'='福建省'
                'Zgansu'    ,'GANSU'='甘肃省'
                'Sguangdong','GUANGDON'='广东省'
                'Tguangxi'  ,'GUANGXI'='广西'
                'Wguizhou'  ,'GUIZHOU'='贵州省'
                'Uhainan'   ,'HAINAN'='海南省'
                'Chebei'    ,'HEBEI'='河北省'
                'Hheilongj' ,'HEILONGJ'='黑龙江省'
                'Phenan'    ,'HENAN'='河南省'
                'Qhubei'    ,'HUBEI'='湖北省'
                'Rhunan'    ,'HUNAN'='湖南省'
                'Jjiangsu'  ,'JIANGSU'='江苏省'
                'Njiangxi'  ,'JIANGXI'='江西省'
                'Gjilin'    ,'JILIN'='吉林省'
                'Eneimeng'  ,'NEIMENG'='内蒙古'
                'bningxia'  ,'NINGXIA'='宁夏'
                'aqinghai'  ,'QINGHAI'='青海省'
                'Oshandong' ,'SHANDONG'='山东省'
                'Ishanghai' ,'SHANGHAI'='上海市'
                'Yshaanxi'  ,'SHAANXI'='陕西省'
                'Dshanxi'   ,'SHANXI'='山西省'
```

```
                'Vsichuan'  ,'SICHUAN'='四川省'
                'Btianjin'  ,'TIANJIN'='天津市'
                'cxinjiang' ,'XINJIANG'='新疆'
                'Xyunnan'   ,'YUNNAN'='云南省'
                'Lzhejiang' ,'ZHEJIANG'='浙江省'
                'Fliaoning' ,'LIAONING'='辽宁省';
run;
%macro format;
     class prov rprov ss _numeric_;
     keylabel n='计数' all='合计';
     format ss  ssfmt.   n1  i.     n2  ii.    n3  iii.
            n4  iv.      n5  sat.   n6  vi.    n7
            n8  viii.    n9  ix.    NA NB NC ND sat.
            NE0 yesno. NE1 yesno. NE2 yesno. NE3 value.
            NF0 yesno. NF1 value. NF2 for.   NF3 act.
            NG0 yesno. NG1 value. NG2 for.   NG3 act.
            prov rprov $prov.
%mend;
%macro table(a,b,box);
       table &a all,all &b*(n pctn<&a all>='列 %'*f=5.2
             pctn<&b all>='行 %'*f=5.2)/rts=16 box=&box;
%mend;
data remote.tran;
   set remote.file;
   length rprov $20.;
   if prov='ANHUI'     then rprov='Kanhui';
   if prov='BEIJING'   then rprov='Abeijing';
   if prov='FUJIAN'    then rprov='Mfujian';
   if prov='GANSU'     then rprov='Zgansu';
   if prov='GUANGDON'  then rprov='Sguangdong';
   if prov='GUANGXI'   then rprov='Tguangxi';
   if prov='GUIZHOU'   then rprov='Wguizhou';
   if prov='HAINAN'    then rprov='Uhainan';
   if prov='HEBEI'     then rprov='Chebei';
   if prov='HEILONGJ'  then rprov='Hheilongj';
   if prov='HENAN'     then rprov='Phenan';
   if prov='HUBEI'     then rprov='Qhubei';
   if prov='HUNAN'     then rprov='Rhunan';
   if prov='JIANGSU'   then rprov='Jjiangsu';
   if prov='JIANGXI'   then rprov='Njiangxi';
   if prov='JILIN'     then rprov='Gjilin';
   if prov='NEIMENG'   then rprov='Eneimeng';
```

```
   if prov='NINGXIA'   then rprov='bningxia';
   if prov='QINGHAI'   then rprov='aqinghai';
   if prov='SHANDONG'  then rprov='Oshandong';
   if prov='SHANGHAI'  then rprov='Ishanghai';
   if prov='SHAANXI'   then rprov='Yshaanxi';
   if prov='SHANXI'    then rprov='Dshanxi';
   if prov='SICHUAN'   then rprov='Vsichuan';
   if prov='TIANJIN'   then rprov='Btianjin';
   if prov='XINJIANG'  then rprov='cxinjiang';
   if prov='YUNNAN'    then rprov='Xyunnan';
   if prov='ZHEJIANG'  then rprov='Lzhejiang';
   if prov='LIAONING'  then rprov='Fliaoning';
run;
proc datasets library=remote;
     modify tran;
     label prov ='省市名'     rprov='省市名'
      name ='医院名称'        id1='医院编号'
      id2  ='病人编号'        ss ='病人类别'
      n1 ='在哪级医院'        NE0='是否托关系'
      n2 ='在哪科就医'        NE1='本院职工'
      n3 ='主要职业'          NE2='中间收礼'
      n4 ='费用支付方式'      NE3='托人价值'
      n5 ='医生服务态度'      NF0='送钱物'
      n6 ='医生的技术'        NF1='送礼价值'
      n7 ='护士服务态度'      NF2='送礼原因'
      n8 ='护士的技术'        NF3='送礼态度'
      n9 ='医疗质量是否信任'  NG0='宴请'
      na ='挂号处态度'        NG1='宴请价值'
      nb ='药房态度'          NG2='宴请原因'
      nc ='医院膳食'          NG3='宴请态度'
      nd ='环境卫生'
;
     format ss _numeric_ 20.;
run;
proc tabulate data=remote.tran f=6. noseps;
     %format;
     %table(rprov,n1,' ');
     %table(rprov,n2,' ');
     %table(rprov,n3,' ');
     %table(rprov,n4,' ');
     %table(rprov,n5,' ');
     %table(rprov,n6,' ');
```

```
     %table(rprov,n7,' ');
     %table(rprov,n8,' ');
     %table(rprov,n9,' ');
     %table(rprov,na,' ');
     %table(rprov,nb,' ');
     %table(rprov,nc,' ');
     %table(rprov,nd,' ');
     %table(rprov,ne0,' ');
     %table(rprov,nf0,' ');
     %table(rprov,ng0,' ');
run;
proc tabulate data=remote.tran f=6. noseps;
     where ne0=1;
     %format;
     %table(rprov,ne1,'托关系者分类');
     %table(rprov,ne2,'托关系者分类');
     %table(rprov,ne3,'托关系者分类');
proc tabulate data=remote.tran f=6. noseps;
     where nf0=1;
     %format;
     %table(rprov,nf1,'送钱物者分类');
     %table(rprov,nf2,'送钱物者分类');
     %table(rprov,nf3,'送钱物者分类');
proc tabulate data=remote.tran f=6. noseps;
     where ng0=1;
     %format;
     %table(rprov,ng1,'宴请者分类');
     %table(rprov,ng2,'宴请者分类');
     %table(rprov,ng3,'宴请者分类');
run;
```

上述程序中，LIBNAME REMOTE '[]'; 语句在VAX 机生成一个逻辑库名。程序首先进行格式和宏定义，并使用汉字，调用时很简便，为了看其运行的具体程序，则在OPTIONS 语句中指定MPRINT 打印。注意省名在具体使用时，在各省的汉语拼音前加了一个A-Z 等的序号，这样产出时是按照全国大区分的，但格式化过程的说明可以同时指定，其它如“好”、“较好”一类的合并也是如此。TABULATE 的选项中特别指定了参数FORMCHAR='————-' 这样产出的表是一般统计学书上所习惯使用的格式，过程也用WHERE 语句限定处理符合条件的数据子集。

最后得到完整的汇总产出表。

运行RSUBMIT后，仍用SIGNOFF对系统复位。

其它方案也是可行的，如合并成大的dBASE数据库文件后，可利用EPI INFO的CONVERT功能直接转到SAS 的数据步程序，仍然用VAX/VMS SAS。但可能由于系统间的差异，转成的

程序在运行时会报一些错误；也可以经SPSSX 的格式传输，进而利用SAS PROC CONVERT。

分析报告的撰写，实际是以上工作的小结。