05. 数据步创建数据集的过程

数据步创建数据集的过程分为两个阶段：

1. 编译阶段：扫描语法错误，生成数据集的“描述信息”；

2. 执行阶段：逐条记录地读入并处理输入数据（循环执行若干次数据步）。

1. **编译阶段**

将数据从外部文件读入“程序数据向量”（Program Data Vector）。

一、**在输入缓冲区（内存）创建“程序数据向量”**

注意：是读入外部数据时创建，而不是读入SAS数据集时创建。

**例1** 下面的数据步代码：

**data** invents;

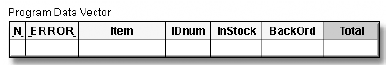
infile 'D:\我的文档\My SAS Files\9.3\invent.dat';

input Item $ **1**-**13** IDnum $ **15**-**19** InStock **21**-**22** BackOrd **24**-**25**;

Total=instock+backord;

**run**;

将创建如下的“程序数据向量”：包括



\_N\_：记录数据步执行的次数，读入一条记录则+1；

\_ERROR\_：用来指示错误，默认是0，遇到错误则变为1；

注意：\_N\_和\_ERROR\_是该处理过程自动生成的，将来也不会出现在数据集的观测值中。

Item、IDnum、InStock、BackOrd：数据集自身的变量，其变量属性（长度、类型等）在第一次读到数据时确定；

Total：数据步中赋值语句生成的变量。

**二、检查语法错误**

（1）关键词缺少或拼写错误；

（2）无效的变量名；

（3）标点符号缺失或拼写错误；

（4）无效的可选参数。

三、**生成数据集的“描述信息”**

遇到数据步的run;语句时生成，包括：

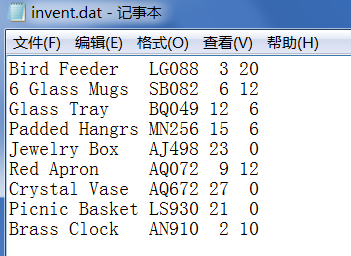
1. 数据集的名称；
2. 数据集包含变量的个数；
3. 数据集各变量的变量名和属性。

注意：此时，“程序数据向量”中还没有内容，数据集中也还没有观测值，观测值将在执行阶段从“程序数据向量”中逐条读取。

1. **执行阶段**

逐条记录地从“程序数据向量”读入并处理输入数据（循环执行若干次数据步）。

例如，**例1**中的invent.dat文件共有9条记录：



数据步读入它将循环执行9次。执行阶段具体步骤：

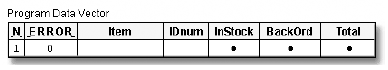
第1次循环：

**一、初始化“程序数据向量”**

\_N\_ = 1

\_ERROR\_ = 0

其它变量 = 缺失值（数值型是 . 字符型是空格）



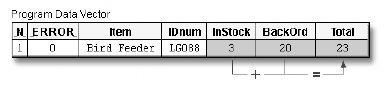
**二、执行input语句**

前例中，代码

input Item $ **1**-**13** IDnum $ **15**-**19** InStock **21**-**22** BackOrd **24**-**25**;

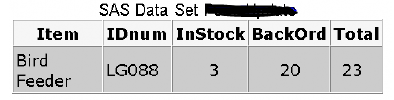
Total=instock+backord;

依次将外部数据文件的第一行的1-13列读入数据存入Item变量，……，赋值语句将instock和backord值相加赋给Total.



**三、数据步收尾工作**

1. 将当前“程序数据向量”中的条目，作为第1条观测值写入输出数据集。



2. 回到数据步起始位置（循环），\_N\_+1=2, \_ERROR=0（若必要）

进入第2次循环……

……直到读取到外部数据文件的“文件尾标识”，完成数据步。

注意：

1. 若不想读入全部数据，可以在infile语句中加上可选参数，控制读入观测值数目：例如

infile 'C:\invent.dat' obs=**10**;

也可以在input语句中加上if语句做选择读入，例如

**data** finance.newcalc;

infile newloans;

input LoanID $ **1**-**4** Rate **5**-**8** Amount **9**-**19**;

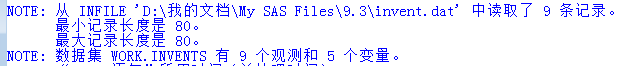
if rate>**0** then

Interest=amount\*(rate/**12**);

else put 'DATA ERROR ' rate= \_n\_=;

**run**;

1. 数据步中若另有指示语句，将按指示语句执行；
2. SAS日志文件信息，将确保外部数据正确地读入数据集：

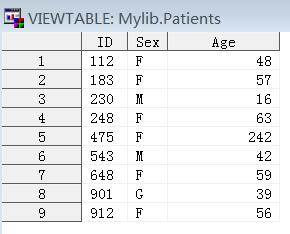


1. **修改错误数据**

数据集创建完成后，需要检查数据的正确性（检查日志文件、输出数据报表、输出数据的均值或频率等）。

发现某个数据值错误，可以用“if语句”对错误数值进行修改，当然也可以在可视化窗口操作修改数据。

**例2** 路径“D:\我的文档\My SAS Files\9.3”下，有数据集patients. sas7bdat:



发现：性别G应修改为F；年龄242明显是错误（删除该条观测值）。

代码：

libname mylib 'D:\我的文档\My SAS Files\9.3';

**data** newpatients;

set mylib.patients;

Sex=upcase(Sex);

if Sex='G' then Sex='F';

if Age>**110** then delete;

**run**;

**proc** **print** data = newpatients;

**run**;

运行结果：



1. **RETAIN和SUM语句**

从前面（二）中知道，迭代（循环）执行从“程序数据向量”读入数据（观测值）时，SAS会先将所有变量值设为缺失值，再通过input或赋值语句改变它。

RETAIN和SUM语句可以改变这种方式。

1. RETAIN语句

retain语句可以让SAS保存前一次变量的值。它可以出现在数据步的任何位置，基本形式为：

retain 变量1 变量2 …;

也可以指定一个初始值，而不是用缺失值或前一次的值代替初始值：

retain 变量 初始值;

1. SUM语句

SUM语句用于你想将一个表达式的值累加到一个变量上去时，基本形式为：

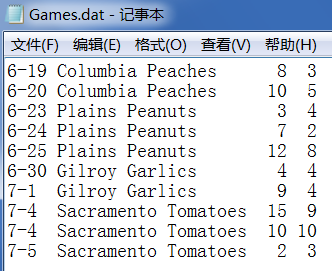
变量 + 表达式;

该语句将表达式的值赋给变量，同时将变量的值保留到下一次迭代。这个变量必须是数值型，且初始值为0。因此，语句等价于如下形式：

retain 变量 0;

变量 = sum(变量,表达式);

**例3** 某赛季棒球比赛的数据（C:\MyRawData\Games.dat），包含比赛日期、参赛队伍、hit数据、runs数据：



要求增加两个变量，一个反应本赛季的总runs数，一个反应一场比赛中最大的runs数。

代码：

**data** gamestats;

infile 'c:\MyRawData\Games.dat';

input Month **1** Day **3**-**4** Team $ **6**-**25** Hits **27**-**28** Runs **30**-**31**;

retain MaxRuns;

MaxRuns = MAX(MaxRuns, Runs);

RunsToDate + Runs;

**run**;

**proc** **print** data = gamestats;

title "Season's Record to Date";

**run**;

运行结果：



程序说明：变量maxruns取前面迭代的maxruns和runs中最大值；变量runstodate将每一场比赛的runs都加到自己身上。