**09. 用SET语句创建,修改数据集**

（一）**用SET语句创建/修改数据集**

用SET语句，可以从已有数据集创建新数据集、增加新变量、修改数据集。

注意：SET语句是一次一个观测值地，对已有数据集循环执行数据步创建新数据集。

**一、基本语法**

data 新数据集名 < (可选项) >;

set 已有数据集名 < (可选项) >;

示例（从数据集sales创建子数据集Friday，增加变量Total）：

**data** friday;

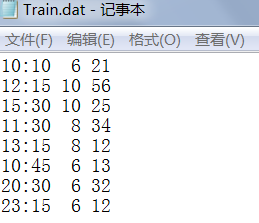
set sales;

if Day = 'F';

Total = Popcorn + Peanuts;

**run**;

**例1** 关于火车运汽车的数据（C:\MyRawData\Train.dat），包括发车时间、火车上的汽车数、火车中的人数：



数据被读入一个永久数据集trains.sas7bdat中，储存在“D:\我的文档\My SAS Files\9.3”目录下。

代码：

**data** 'c:\MySASLib\trains';

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

input Time TIME5. Cars People;

**run**;

由于每辆车的最大乘客数为6人，现在想知道一列火车上，平均每辆汽车的乘客数是多少，在一个新数据集中插入一列计算。

代码：

**data** averagetrain;

set 'D:\我的文档\My SAS Files\9.3\trains';

PeoplePerCar = People / Cars;

**run**;

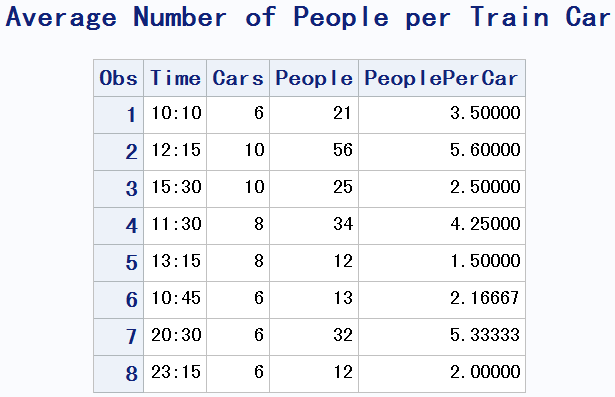
**proc** **print** data = averagetrain;

title 'Average Number of People per Train Car';

format Time TIME5.;

**run**;

运行结果：



**二、用“drop（舍弃）和keep（保留）选项”选择变量（列）**

在data步中该选项以及SET语句，可以控制从原数集中读出的变量是否被写入要创建的数据集。

基本形式：

data 新数据集名 (keep = 变量1…) (drop = 变量1…);

set 旧数据集;

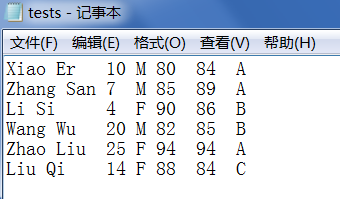
注：keep和drop选项也可用于输出数据集时的可选参数；

另外，rename选项可以对列变量重命名，示例：

**data** animals (RENAME = (Class = Type Habitat = Home));

set animals;

**例2** 学生成绩数据（c:\MyRawData\tests.txt），包含变量Name, Subject, Gender, Exam1, Exam2, Homework:



读入数据，创建新数据集，舍弃变量Subject和Homework，重命名变量Name为Student Name.

代码：

**data** Test;

infile 'c:\MyRawData\tests.txt';

input Name $**1**-**9** Subject **11**-**12** Gender $ **14** Exam1 **16**-**18** Exam2 **20**-**22** Homework $ **24**;

**run**;

**proc** **print** data=Test;

title 'Test';

**run**;

**data** Test1 (DROP = Subject Homework RENAME =(Name = Student Name));

/\* 用 (KEEP = Name Gender Exam1 Exam2) 也一样效果 \*/

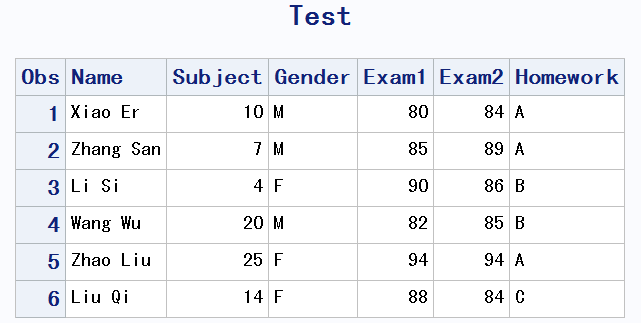
set Test;

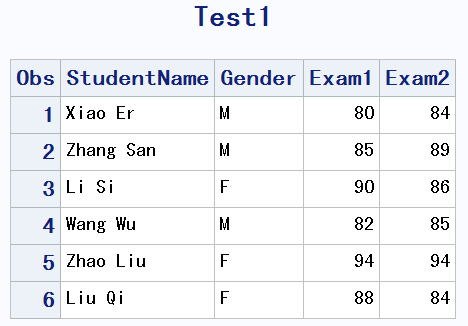
**proc** **print** data = Test1;

title 'Test1';

**run**;

运行结果：





1. **SET语句中使用可选项 FIRSTOBS=n 和 OBS=m 选择观测值（行）**

表示只选择旧数据集中的第 n 条到第 m 条观测值。示例：

**data** animals

set animals (FIRSTOBS = **101** OBS = **120**);

另外，选择观测值（行），也可用IF-THEN（条件）语句选出满足条件的观测值（行）。【见前文】

**（二）用 UPDATE 语句更新数据集的数据**

经常会遇到需要不断更新的数据集，比如银行账户每笔交易完需要更新有变动的数据。用 UPDATE 语句可以实现这一功能，需要主数据集和新数据集有共同变量（来匹配）。

基本形式：

data *主数据集*;

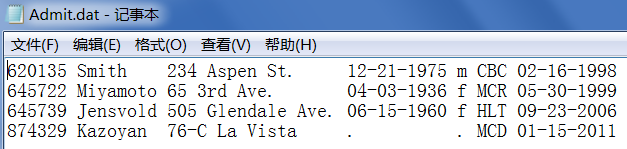
update *主数据集 新数据集*;

by 共同（匹配）变量;

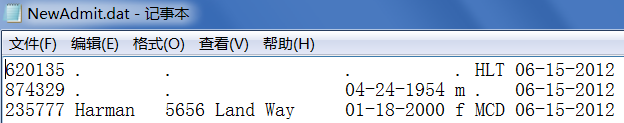
注：（1）主数据集和新数据集都需要事先按共同变量排好序，且共同变量必须具有唯一性；

（2）新数据集中的缺失值不会改下主数据集的相应数值。

**例3** 医院有一份关于病人的主数据（C:\MyRawData\Admit.dat）变量包括病人账户号、姓名、地址、出生日期、性别、保险代码、信息最后更新的时间：



当有新病人或其他病人再进医院时，信息会被更新，比如，第一个病人的保险代码被更换了、最后一个病人的缺失数据被填补上、有新病人加入。更新数据文件（C:\MyRawData\NewAdmit.dat）如下：



将主数据存入一个名为 patientmaster 的永久数据集，读取新数据并排序，使用 update 语句将新数据更新到主数据中。

代码：

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

**data** perm.patientmaster;

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

input Account LastName $ **8**-**16** Address $ **17**-**34**

BirthDate MMDDYY10. Sex $ InsCode $ **48**-**50** @**52** LastUpdate MMDDYY10.;

**run**;

**data** perm.transactions;

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

input Account LastName $ **8**-**16** Address $ **17**-**34** BirthDate MMDDYY10.

Sex $ InsCode $ **48**-**50** @**52** LastUpdate MMDDYY10.;

**run**;

**proc** **sort** data = transactions;

by Account;

**run**;

\* Update patient data with transactions;

**data** perm.patientmaster;

update perm.patientmaster transactions;

by Account;

**run**;

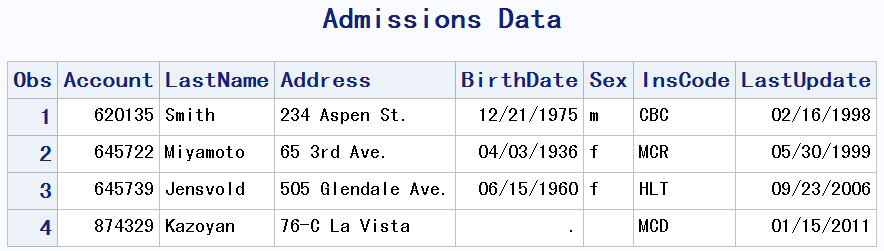
**proc** **print** data = perm.patientmaster;

format BirthDate LastUpdate MMDDYY10.;

title 'Admissions Data';

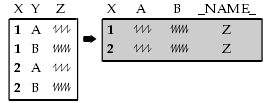
**run**;

运行结果：



另外，也可以用 MODIFY 语句修改数据集，比用 SET 语句修改更有效率，因为 MODIY 修改数据集，只是在原数据集上做修改，而不会创建新数据集。

**（三）用 TRANSPOSE 过程步对数据集转置**



有时候需要将数据集（表）的观测值（行）和变量（列）交换，即把观测值变为变量，把变量变为观测值，用 TRANSPOSE 过程实现。

基本形式：

proc transpose data=原数据集 out=转置数据集 <可选项> ;

by 原数据集中不参与转置的列变量;

id 给转置后的列赋予变量名;

var 原数据中需要转置的列变量;

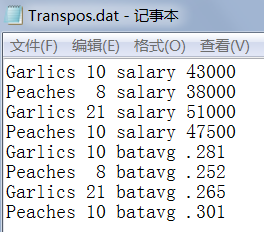
说明：（1）by 语句指定原数据集中不参与转置的变量（转置之前需要先按该变量排好序），一般“用来分组的变量”不参与转置，例如上图中的 X 列；

（2）id语句给转置后的列赋予变量名，默认是：COL1 COL2……，可以使用原数据集的某列（若该列是数值，则加下划线作为前缀，以符号变量的命名规范）；

（3）可选项“NAME = 变量名”，为原数据集各列变量名（在转置数据集作为第一列）起个变量名，默认是用\_NAME\_;

（4）可选项“PREFIX = 前缀”，用“前缀”替换（2）中默认列变量名的“COL”。

**例4** 关于儿童棒球联盟选手的数据（C:\MyRawData\Transpos.dat），包括包括队名、选手编号、类型（击球数/击球率）、以及击球数/击球率：



观察数据，第 3 列的 salary(击球数)、batavg（击球率）应该是列变量名，其取值是第 4 列。读取数据，对第 3 列进行转置。

代码：

**data** baseball;

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

input Team $ Player Type $ Entry;

**proc** **sort** data = baseball;

by Team Player;

**proc** **print** data = baseball;

title 'Baseball Data After Sorting and Before Transposing';

**run**;

\* Transpose data so salary and batavg are variables;

**proc** **transpose** data = baseball OUT = flipped;

by Team Player;

id Type;

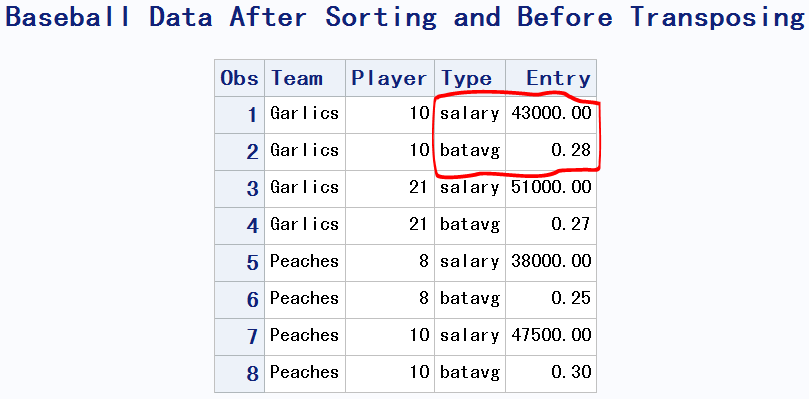
var Entry;

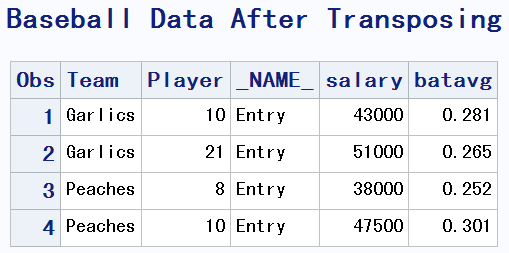
**proc** **print** data = flipped;

title 'Baseball Data After Transposing';

**run**;

运行结果：（相当于将红圈的四个值转置，同时两行并一行）





程序说明：

（1）“by Team Player”指定原 Team 列和 Player 列，不参与转置，仍作为转置数据集的列；

（2）“id Type”指定 Type 列的值：salary 和 batavg 作为转置数据的变量名；

（3）“var Entry”指定对 Entry 列进行转置，同时列名 “Entry”将作为“值”成为转置数据集的一列，默认列名是“\_NAME\_”，可用【可选项“NAME = …”】修改它。