**12. SAS数组**

使用SAS数组（ARRAY语句），主要是对多个变量做相同操作时，可以通过数组存储这些变量，借用数组下标执行循环结构来实现，从而大大简化和缩短程序代码。

SAS数组是存储一组同类型（数值型或字符型）的变量，这些变量可以是已存在的，也可以是新创建的。

**一、基本语法**

ARRAY *数组名[n]* <$> *变量列表*;

说明：（1）n是数组的长度（即变量个数）；也可以用“[\*]”不指定数组长度，而是让SAS根据变量列表数目自己判断；也可以指定数组的下标范围，例如，

array Year[**2005**:**2010**] YR2005 - YR2010;

（2）若是字符型变量需要加“$”，也可以指定字符的长度（“$1”表示数组元素是1个字节的字符）；

（3）若变量列表各变量是“相同字符+连续数字”可以简写（下面两句代码功能相同）：

array Cat8 - Cat12;

array Cat8 Cat9 Cat10 Catll Cat12;

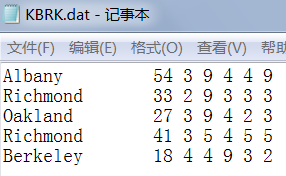
示例：array store[**4**] Macys Penneys Sears Target;

定义数组store，含有4个数值型变量：Macys，Penneys，Sears，Target

使用数组变量Sears用“store[3]”即可。

注意：数组本身不储存在数据集中，只在数据步中定义和使用，即不会创建变量“store[1]，store[2]……”；

**例1** 广播电台KBRK做了一份歌曲的听众调查，对5首歌进行打分，分值在1-5，如果没听过则填9. 数据文件（C:\MyRawData\KBPK.dat）包括了被访者姓名、年龄、以及5首歌的打分：



读取数据，将打分为9的改为缺失值。

代码：

**data** songs;

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

input City $ **1**-**15** Age wj kt tr filp ttr;

array song[**5**] wj kt tr filp ttr;

do i = **1** to **5**;

if song[i] = **9** THEN song[i] =**.**;

end;

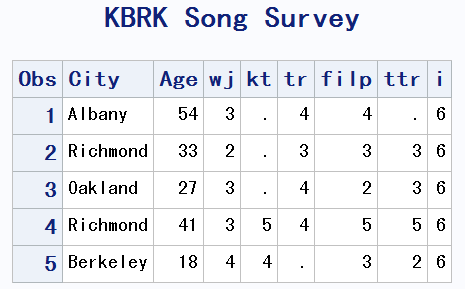
**run**;

**proc** **print** data = songs;

title 'KBRK Song Survey';

**run**;

运行结果：



注意：循环变量i会自动作为一列新变量写入数据集，要想避免它，需要加上一句“drop i;”。

**二、临时数组**

有时候需要单纯用数组保存一组数值或字符值，用于数据集各观测做比较的“标准”。此时，适合用临时数组。

临时数组不创建任何变量。

**例2** 用临时数组保存5科考试的及格分数，读取学生成绩，然后与这些及格分数进行比较，统计学生的及格科目数。

代码：

**data** Passing;

array Pass[**5**] \_TEMPORARY\_ (**65** **70** **65** **80** **75**);

array Score[**5**];

input ID $ Score[\*];

Pass\_Num = **0**;

do i=**1** to **5**;

if Score[i] >= Pass[i] then Pass\_Num + **1**;

end;

drop i;

datalines;

001 64 69 68 82 74

002 80 80 80 60 80

;

**proc** **print** data = Passing;

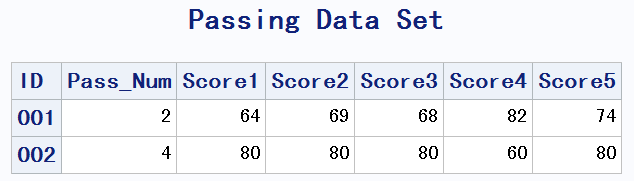
title "Passing Data Set";

id ID;

var PASS\_NUM SCORE1-SCORE5;

**run**;

运行结果：



**例3** 用临时数组对10道选择题的测验计分，第一行数据是正确答案，其它行包括学生ID和学生对10道题的作答。不同于 **例2**中给临时数组赋值，而是读取数据的第一行作为临时数组。

代码：

**data** Score;

array Key[**10**] $**1** \_TEMPORARY\_;

array Ans[**10**] $**1**;

array Score[**10**] \_TEMPORARY\_;

\* 读入标准答案存入临时数组Key;

if \_N\_ = **1** then

do i=**1** to **10**;

input Key[i] @;

end;

\* 读入学生 ID 和学生作答的答案;

input ID $ @**5** (Ans1-Ans10) ($1.);

RawScore = **0**;

do i=**1** to **10**;

Score[i] = (Ans[i] = Key[i]);

RawScore + Score[i]; \* 累加各题的得分;

end;

Percent = **100** \* RawScore / **10**; \* 将得分转化为百分制;

drop i;

datalines;

A B C D E E D C B A

001 ABCDEABCDE

002 AAAAABBBBB

;

**proc** **print** data = Score;

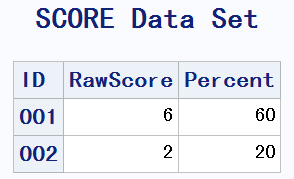
title "SCORE Data Set";

id ID;

var RawScore Percent;

**run**;

运行结果：



程序说明：

（1）“$1”告诉SAS数值元素是1个字节的字符；

（2）IF语句，在第1次读入时（\_N\_=1），选择把第一行数据读入 Key 数组，作为标准答案；

（3）“Score[i] = (Ans[i] = Key[i]);”表示若第 i 道题的作答，与该题答案相同，则把逻辑值1赋给第 i 题的得分Score[i].