**07. IF和WHERE语句**

1. IF-THEN语句

**一、基础语法**

做选择时需要用到条件语句：若满足……条件，则执行……

语法：

if 条件 then 执行语句;

若有多个执行语句，则用

if 条件 then DO;

执行语句1;

执行语句2;

END;

多选择分支，则用

if 条件 then 执行语句;

else if 条件 then 执行语句;

else 执行语句;

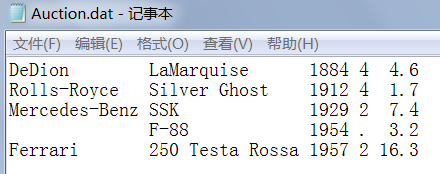
**SAS中常用的比较、逻辑算符（也见系列02）：**

|  |  |  |
| --- | --- | --- |
| **符号** | **含义** | **示例** |
| = 或eq | 等于 | name = 'Jones, C.'; |
| ^= 或ne | 不等于 | temp ^= 212; |
| > 或gt | 大于 | income > 20000; |
| < 或lt | 小于 | partno < "BG05"; |
| >= 或ge | 大于等于 | id >= '1543'; |
| <= 或le | 小于等于 | pulse <= 85 |
| contains或 ？ | 包含 | firstname contains 'Jon'; |
| in | 属于 | actlevel in ('LOW','MOD');  fee in (124.80,178.20); |
| and | 且 | age<=55 and pulse>75; |
| or | 或 | area='A' or region='S'; |

**二、数据步中使用IF-THEN语句赋值**

数据步中经常需要有选择地赋值新变量或修改原数值。

**例1** 稀有古董汽车拍卖的数据文件（C:\MyRawData\ Auction.dat）包含了品牌、型号、制造年份、座位数、价格（百万美元）：



使用IF-THEN语句填满缺失值，并创建一个新变量Veteran.

代码：

**data** oldcars;

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

input Make $ **1**-**13** Model $ **15**-**29** YearMade Seats MillionsPaid;

if YearMade < **1890** then Veteran = 'Yes';

if Model = 'F-88' then DO;

Make = 'Oldsmobile';

Seats = **2**;

end;

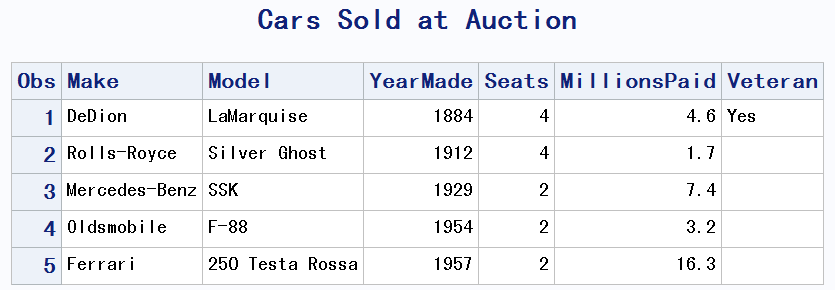
**run**;

**proc** **print** data = oldcars;

title 'Cars Sold at Auction';

**run**;

运行结果：



1. **数据步中使用IF-THEN语句创建子数据集（也见系列03）**
2. 数据步读入数据时，用IF-THEN语句选择满足条件的数据读入。语法：

if 条件;

如果“条件”为真，则数据步将继续执行。还可以使用DELETE语句，来删除哪些不要的数据： 下面这两句话是等价的：

if Sex='f';

if Sex='m' then DELETE;

**例2** 莎士比亚歌剧的清单（C:\MyRawData\Shakespeare.dat），包含歌剧名、首次表演年份、类型：



读取数据，并且用IF语句构造一个只包含喜剧（comedies）的子集：

代码：

**data** comedy;

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

input Title $ **1**-**26** Year Type $;

if Type = 'comedy';

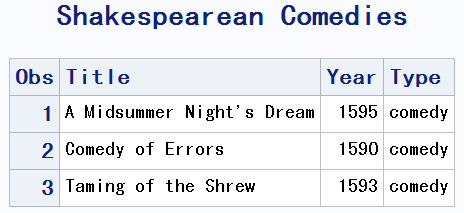
**run**;

**proc** **print** data = comedy;

title 'Shakespearean Comedies';

**run**;

运行结果：



1. 数据步中用IF-THEN语句创建多个数据集

若想在一个数据步中创建多个数据集，在DATA语句后面多接几个数据集名即可。用IF-THEN语句可以分别将数据写入不同的数据集。

**例3** 动物园给动物喂食的数据（C:\MyRawData\zoo.dat），变量为动物类型、生物学分类、居住区域、喂食时间（上午/下午/两者）：



读入数据并输出两个列表，一个是早上喂食，一个是下午喂食。

代码：

**data** morning afternoon;

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

input Animal $ **1**-**9** Class $ **11**-**18** Enclosure $ FeedTime $;

if FeedTime = 'am' then output morning;

else if FeedTime = 'pm' then output afternoon;

else if FeedTime = 'both' then output;

/\* 每个数据集都输出，省略数据集名 \*/

**run**;

**proc** **print** data = morning;

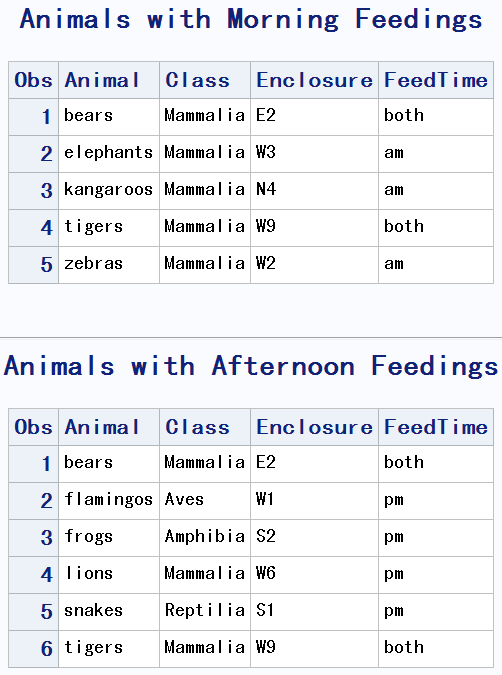
title 'Animals with Morning Feedings';

**proc** **print** data = afternoon;

title 'Animals with Afternoon Feedings';

**run**;

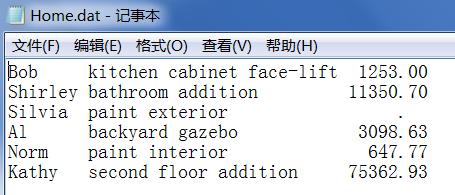
运行结果：



**三、用IF-THEN语句将观测值分组**

对观测值分组，实际上只是为每个观测值分配一个分组标签，用新的列变量存储分组标签，这样后续就可以借助分组关键词（class），对数据做分组处理。

**例4** 住房改善的数据（C:\MyRawData\home.dat），包括了姓名、改善工作、改善成本：



读取数据，并新建了一个CostGroup的变量。根据Cost的值将数据分成high、medium、low和missing三类。

代码：

**data** homeimprovements;

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

input Owner $ **1**-**7** Description $ **9**-**33** Cost;

if Cost =**.** then CostGroup = 'missing';

else if Cost < **2000** then CostGroup = 'low';

else if Cost < **10000** then CostGroup = 'medium';

else CostGroup = 'high';

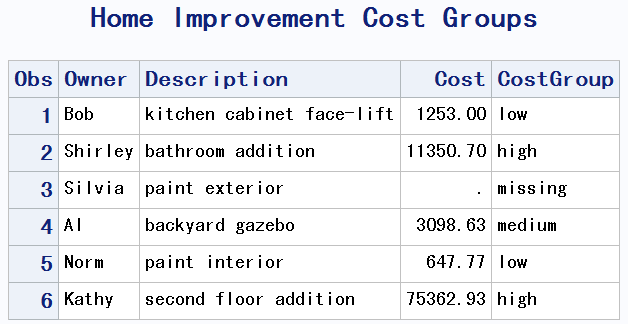
**run**;

**proc** **print** data = homeimprovements;

title 'Home Improvement Cost Groups';

**run**;

运行结果：



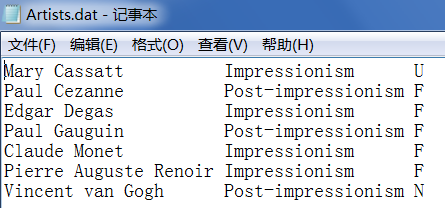
**程序说明**：缺失值要单独分组，否则将归类到low类（缺失值默认是最小值，甚至比负数都小）。

1. **WHERE语句**
2. **基本语法**

WHERE语句和IF语句作用基本相同，不同在于IF语句只用于数据步，WHERE语句还可用于过程步（不创建新数据集），作用是让“只有满足条件的观测值被proc过程处理”。语法：

where 条件;

**例5** 画家信息的数据文件（C:\MyRawData\ Artists.dat），包括画家姓名、流派、国别：



读入数据，只输出流派为Impressionism的画家。

代码：

**data** painters;

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

input Name $ **1**-**21** Genre $ **23**-**40** Origin $ **42**;

**run**;

**proc** **print** data = painters;

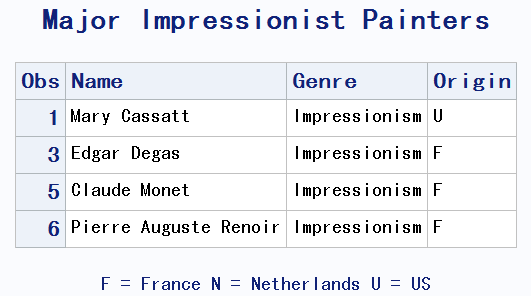
where Genre = 'Impressionism';

title 'Major Impressionist Painters';

footnote 'F = France N = Netherlands U = US';

**run**;

运行结果：



**二、更灵活的应用：“（WHERE=（条件））”作为选项**

同样的功能，WHERE语句作为选项，用起来更加灵活。语法：

( WHERE = (条件) )

示例：

**data** gone;

set animals (WHERE = (Status = 'Extinct'));

**data** uncommon (WHERE = (Status IN ('Endangered', 'Threatened')));

set animals;

**proc** **import** datafile = 'c:\MyRawData\Wildlife.csv'

out = animals (WHERE = (Class = 'Mammalia')) REPLACE;

**proc** **print** data = animals (WHERE = (Habitat='Riparian'));

**proc** **export** data = animals (WHERE = (Status='Threatened'))

outfile = 'c:\MyRawData\Wildlife.xls';

注意：set关键词表示从数据集animals中创建数据。

**例6** 山脉数据（C:\MyRawData\ Mountains.dat）包括名称、位置、高度：



读入数据，分别输出高度>6000, 以及位于美洲的山脉。

代码：

**data** tallpeaks (WHERE = (Height > **6000**))

american (WHERE = (Continent CONTAINS ('America')));

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

input Name $**1**-**14** Continent $**15**-**28** Height;

**run**;

**proc** **print** data = tallpeaks;

title 'Members of the Seven Summits above 6,000 Meters';

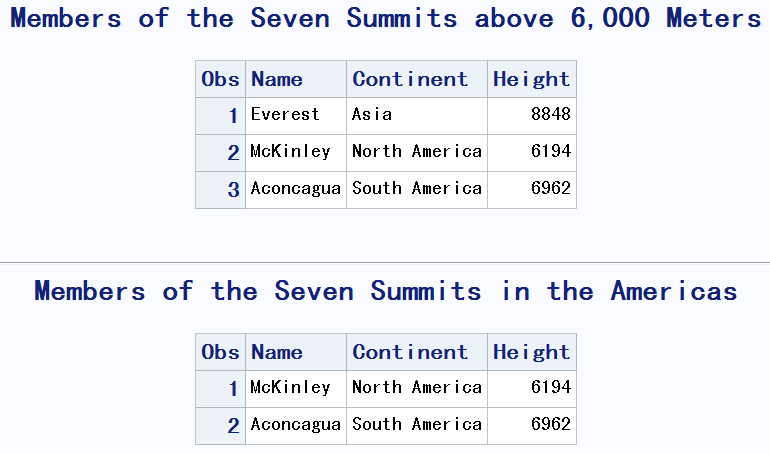
**run**;

**proc** **print** data = american;

title 'Members of the Seven Summits in the Americas';

**run**;

运行结果：



1. **IF与WHERE的区别**

将数据集SASHELP.workers第10到15条观测中满足条件"ELECTRIC>260"的观测提取出来，生成新的数据集tmp。

代码1：

**data** tmp;

set SASHELP.workers(firstobs=**10** obs=**15**);

if ELECTRIC > **260**;

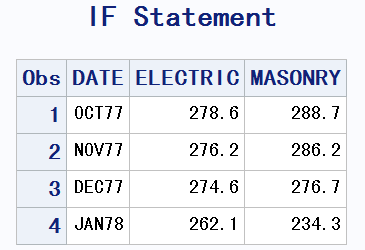
**run**;

**proc** **print** data = tmp;

title 'IF Statement';

**run**;

运行结果（4条记录）：



代码2：

**data** tmp;

set SASHELP.workers(firstobs=**10** obs=**15**);

where ELECTRIC > **260**;

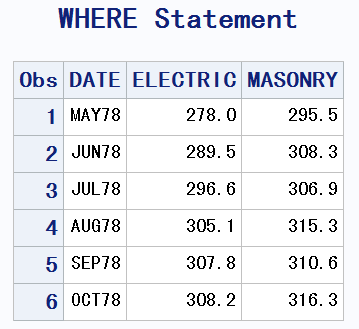
**run**;

**proc** **print** data = tmp;

title 'WHERE Statement';

**run**;

运行结果（6条记录）：



上述两种方法为什么输出结果不一样？请注意IF语句和WHERE语句的区别：

（1）IF语句是面向“程序数据向量”（Program Data Vector）的，对当前PDV中的数据进行判断，满足条件时将其写入到外部数据集；WHERE语句也是面向PDV的，它使用于从外部数据源读数据到PDV之前进行判断，当满足条件时才被写入到PDV。显然一个在写入PDV之前，一个在写入PDV之后，两者是有差异的。

（2） 当没有数据集选项firstobs=10 obs=15时，IF语句和WHERE语句用法和结果相同，但有这两个选项时效果就不同了。有这两个选项，IF语句是从原数据集（或数据源）的观测记录进行计算个数，即从原数据集的第10个观测开始读入到PDV中，然后再判断是否满足IF条件，若满足则输出到外部数据集，直到原数据集的第15个观测结束（满足条件的只有4个）。而WHERE语句是在读入到PDV之前就进行判断的，所以这里的firstobs的意思是从使得满足WHERE条件的第10个观测开始，而不是原数据集的第10个观测开始，直到满足WHERE条件的第15个观测结束（共6个）。