1. 输出数据报表

利用SAS数据集，可以用PROC PRINT过程步根据需要输出各种满足一定条件的报表。

1. 直接输出

语法：

proc print data = 数据集;

run;

注：此时（不加任何参数），默认

1. 输出数据集中的所有观测值和变量；
2. 报表最左侧增加一列观测值计数列“[Obs]列”；
3. 报表中变量出现的顺序与数据集中位置相同。

（4）若要双倍行距输出报表，可以在数据集后面加上可选参数：“double”.

1. 输出路径'D:\我的文档\My SAS Files\9.3'下的SAS数据集therapy.

代码：

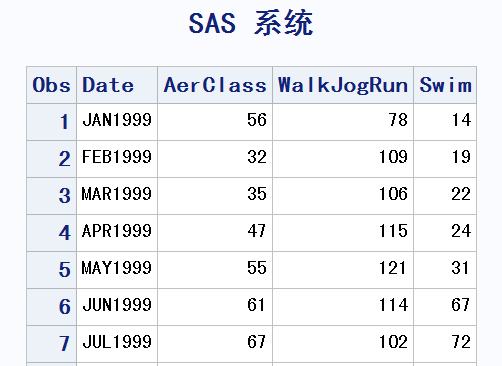
libname patients 'D:\我的文档\My SAS Files\9.3';

**proc** **print** data=patients.therapy;

/\* 注意数据集前加上 data = 否则报错 \*/

**run**;

运行结果（部分）：



1. 选择变量和输出顺序

语法：

proc print data = 数据集;

var 变量1 变量2 …;

run;

注：（1）若不输出观测值计数列[Obs]，需加上参数noobs；

（2）若要指定某列或某几列代替“[Obs]列”，可用

id 变量1 变量2 …

注意：若一个变量既是var变量又是id变量，将输出两次。

1. (1) 输出原始数据集sasuser.admit

(2)只输出变量age height weight fee

代码：

**proc** **print** data=sasuser.admit noobs;

**run**;

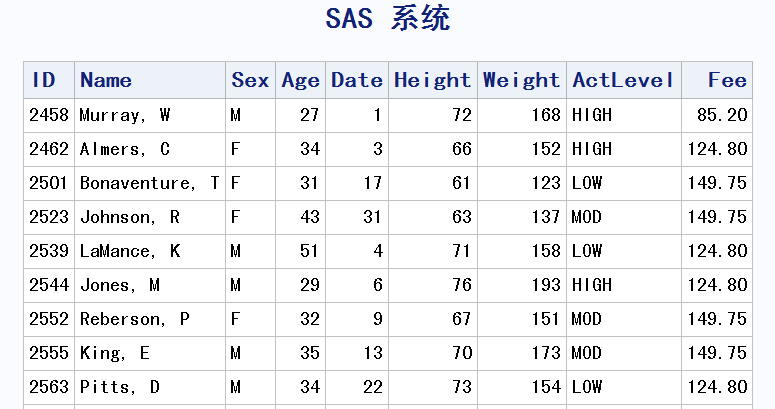
**proc** **print** data=sasuser.admit;

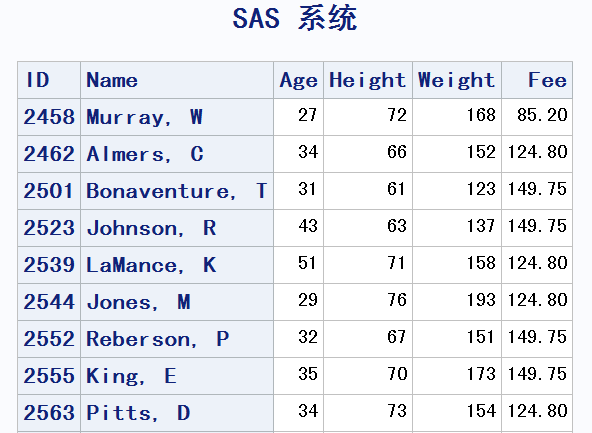
var Age Height Weight Fee;

id ID Name;

**run**;

运行结果（部分）：





1. 选择部分观测值

语法：

proc print data = 数据集;

where 条件语句

run;

表示选择满足某条件的观测值。注意，where语句可以指定数据集中任何变量，而不受var语句的限制。

条件语句可以由各种SAS算符和括号组合而成，为此下面介绍一点SAS中的比较、逻辑算符：

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

1. 输出数据集Sasuser.admit中满足条件Age>30并且Height>65的观测值，只输出变量Age Height Weight Fee.

代码：

**proc** **print** data=sasuser.admit;

var Age Height Weight Fee;

where Age>**30** and Height>**65**;

**run**;

运行结果（部分）：



1. 对数据进行排序

语法：

proc sort data=数据集 out=新数据集;

by < descending > 变量1 变量2 …;

run;

注：（1）省略“out=新数据集”，原数据集将被排好序的数据集替换；

（2）“< descending >”为可选参数（递减排序），只对紧随其后变量起作用；默认是递增排序；

（3）先按变量1排序，变量1相同，再按变量2排序…

（4）缺失值，当成最小的值。

1. 对数据集Sasuser.admit按照Weight递减，Age递增排序，将排序好的数据存为临时数据集work.wgtadmit. 并输出Age>30的观测值，可选变量为Weight Age Height Fee.

代码：

**proc** **sort** data=Sasuser.admit out=work.wgtadmit;

by descending Weight Age;

**run**;

**proc** **print** data=work.wgtadmit;

var Weight Age Height Fee;

where Age>**30**;

**run**;

运行结果（部分）：



1. 对数值变量生成按列汇总
2. 语法：

proc print data = 数据集;

sum 变量1 变量2 …;

run;

注：列的汇总，出现在最后一行，格式与变量列相同。

1. 输出数据集Sasuser.insure中满足条件Pctinsured < 80的观测值，对Balancedue列作汇总，只输出变量Name Policy Balancedue.

代码：

**proc** **print** data=Sasuser.insure;

var Name Policy Balancedue;

where Pctinsured < **80**;

sum Balancedue;

**run**;

运行结果：



2. 若要分组汇总，可以用by和sum组合来实现。语法：

sum 变量1 变量2 …;

by <descending> 变量1 <notsorted> …;

注：（1）需要对数据集进行“分组”处理时，可以用by语句，但要求数据集事先必须已经按by变量排好序了；（另一个“分组”处理语句是class，不要求数据集事先按class变量排序，按class变量的不同值进行分类计算和分析）

（2）notsorted选项的作用就是告诉SAS：数据不是按照字母或者数字大小排序的，而是按照by变量的值按组排列的；

（3）输出结果根据by变量不同的值，分成多个报表。

1. 对数据集Sasuser.admit按Actlevel分组分别对各组的Fee进行汇总。

代码：

**proc** **sort** data=Sasuser.admit out=work.activity;

by Actlevel;

**run**;

**proc** **print** data=work.activity;

var Age Height Weight Fee;

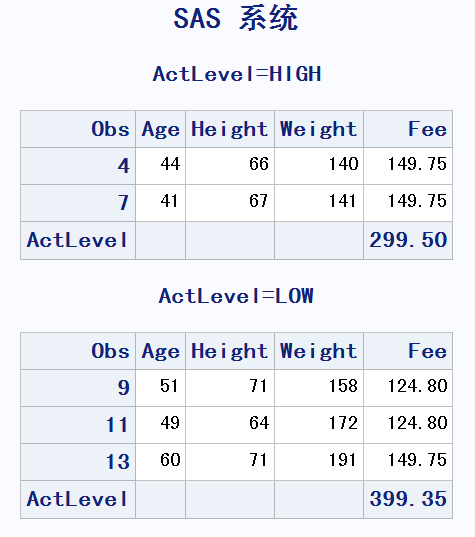
where Age>**40**;

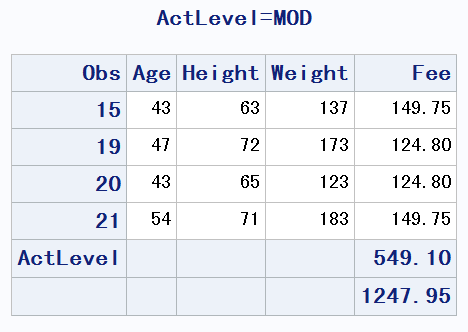
sum Fee;

by Actlevel;

**run**;

运行结果：





若要以更紧凑直观的报表输出（用分组变量列代替[Obs]列），可以再加一个id语句，改写例6的代码如下：

代码：

**proc** **sort** data=Sasuser.admit out=work.activity;

by Actlevel;

**run**;

**proc** **print** data=work.activity;

var Age Height Weight Fee;

where Age>**40**;

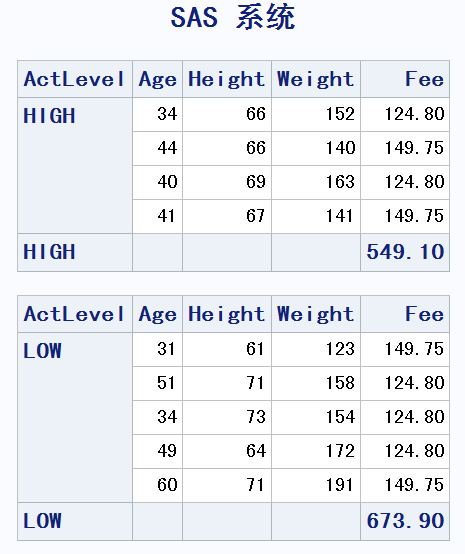
sum Fee;

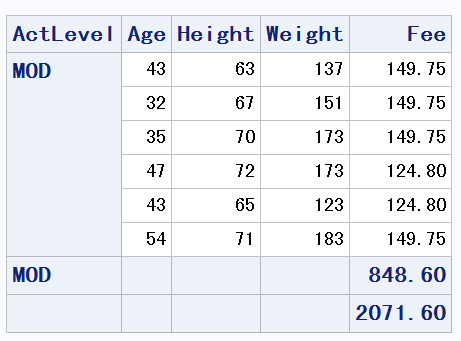
by Actlevel;

id Actlevel;

**run**;

运行结果：





若在每页输出一个报表，可以加上“pageby actlevel;”。

1. 设置报表的标题和脚注

语法：

title<n> ‘标题名’;

footnote<n> '脚注名';

注：（1）默认的标题是“SAS 系统”（居中），默认无脚注；

（2）可选参数n，用来指定标题/脚注所在的行（title=title1）；

（3）title和footnote是全局命令，除非重新定义新的title和footnote或者结束当前SAS会话，否则将对后面的输出始终有效（取消前面的title，可以只用tilte关键词后面不加标题名）。

1. 设置标题和脚注的例（无title2，会空出第2行）.

代码：

title1 'Heart Rates for Patients with';

title3 'Increased Stress Tolerance Levels';

footnote1 'Data from Treadmill Tests';

footnote3 '1st Quarter Admissions';

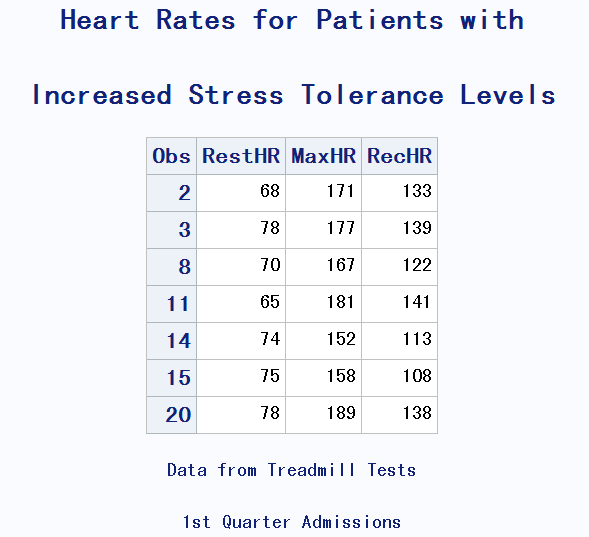
**proc** **print** data=Sasuser.stress;

var resthr maxhr rechr;

where tolerance='I';

**run**;

运行结果：



1. 为变量指定描述性标签

数据集的变量名可能是简写或略写，表义并不明确，可以用label命令重新指定变量标签。语法：

proc print data=数据集 label;

label 变量1 = ‘变量标签1’ 变量2=‘变量标签2’ …;

run;

注：数据集后面不要忘了加上label.

1. 给变量设置描述性标签的例. 将变量名height设为Height in Inches, 将变量名weight设为Weight in Pounds.

代码：

**proc** **print** data=Sasuser.admit label;

var actlevel height weight;

label actlevel='Activity Level'

height='Height in Inches'

/\* 也可以每个标签，都用label命令引出 \*/

weight='Weight in Pounds';

**run**;

运行结果（部分）：



1. 为输出变量指定格式
2. 日期、数值等都有不同格式，可以根据需要指定。语法：

proc print data=数据集;

format 变量1 格式1 变量2 格式2…;

run;

常用变量格式列表：

|  |  |  |
| --- | --- | --- |
| **格式语句** | **含义** | **示例** |
| w. | 整数值共占w位，无小数位 | 5678 |
| w.d | 数值共占w位，d位小数 | 5678.90 |
| $w. | 字符值共占w位，无小数位 | char |
| DATE7. | 日期值共占7位，无小数位 | 16OCT99 |
| DATE9. | 日期值共占9位，无小数位 | 16OCT1999 |
| MMDDYY 8. | 月日年共占8位，无小数位 | 06/05/03 |
| MMDDYY10. | 月日年共占10位，无小数位 | 09/12/1997 |
| COMMAw.d | 逗号表示数值，共占w位，d位小数 | 5,678.90 |
| DOLLAR w.d | “美元号+数值”共占w位，d位小数 | $38,245.40 |

注：（1）若指定的总位数w不够，SAS系统将自动按“最优选择”输出，例如，同样是“38245.3975美元”，

DOLLAR12.2 输出 $38,245.40;

DOLLAR9.2 输出 $38245.40

DOLLAR8.2 输出 38245.40

（2）SAS中的日期变量是按数值存储的，从01/01/1960开始计数，即0=01/01/1960, 1=01/02/1960…….

（3）在proc步中指定的label和format只是临时指定，只对当前程序步起作用。若要永久的指定label和format，可以在data步中使用label和forma命令；

（4）也可以自己定义格式来使用，例如：

**proc** **format**;

value $repfmt /\* 数值型不用加 $ \*/

'TFB'='Bynum'

'MDC'='Crowley'

'WKK'='King';

**run**;

**proc** **print** data=vcrsales;

var salesrep type unitsold;

format salesrep $repfmt.;

**run**;