**Chapter IV – Creating List Report**

1. Using **PROC PRINT** step to create a list report

General form, basic **PROC PRINT** step (List all the variables that the file included):

Basic code:

**PROC PRINT DATA =** *SAS-data-set;*

**RUN;**

*SAS-data-set* is the name of SAS data set to be printed

* **SELECTING VARIABLES**
* General form **, VAR (**variable**)** statement (List only the variables that has been specified under VAR statement):

Select variables and control the order in which they appear by using a VAR statement in your PROC PRINT step.

Basic code:

**VAR** *variable(s)***;**

*variable(s)* is one or more variable names, separated by blanks.

Eg:

**PROC PRINT** data=clinic.admit;

**VAR** age height weight fee;

**RUN**;

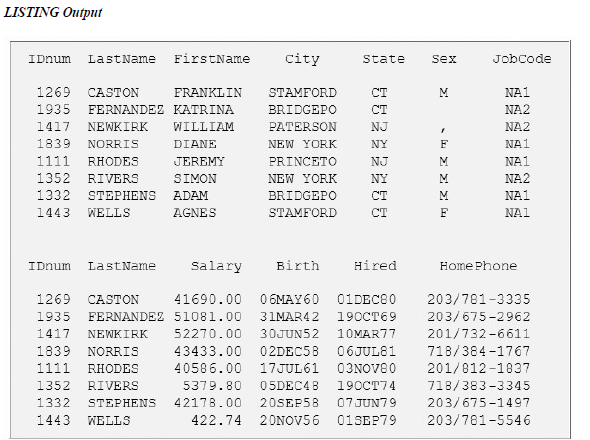
* Remove the **OBS** column:

To remove the Obs column, specify the **NOOBS** option

Eg:

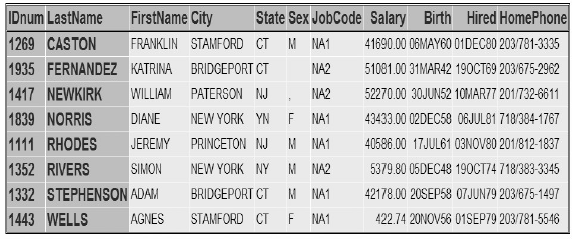
**PROC PRINT** data=work.example **NOOBS**;

**VAR** age height weight fee;

**RUN**;

* **ID** Statement:

Useful when observations are too long to print on one line. 当observations太长（或者说variable过多）无法再同一行中显示，SAS会自动将第一行显示不下的variable在另一行中显示，并且在每行的开头显示相同的variable。

* Basic code:

**ID** *variable(s)***;**

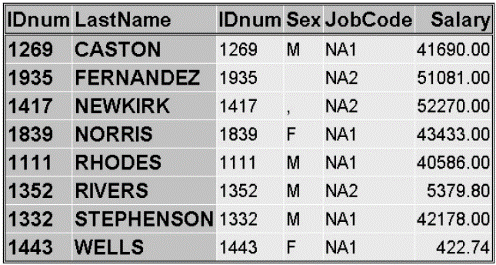
*variable(s)* specifies one or more variables to print instead of the observation number at the beginning of each row of the report.

Eg: **PROC PRINT** data=sales.reps;

**ID** idnum lastname;

**RUN**;

* In **LISTING** output, the IDnum and LastName columns are repeated for each observation that is printed on more than one line.
* 如果在ID statement的基础上，要限定显示的variable：

 Basic code:

**PROC PRINT** data=sales.reps;

**ID** idnum lastname;

**VAR** idnum sex jobcode salary;

**RUN**;

如果已有**ID** statement listed，在**VAR**可以选择不列举**ID** statement中已经list的variables

* Selecting Observation by using **WHERE** statement

Only one **WHERE** statement should be written in each line, if there are more than one **WHERE** statement, one the last will be processed

**WHERE** statement can be both character and numeric variables

* Basic code:

**WHERE** *where-expression***;**

*where-expression* specifies a condition for selecting observations.

Eg:

**PROC PRINT** data=clinic.admit;

**VAR** age height weight fee;

**WHERE** age>30;

**RUN**;

**Comparison Operator**

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Meaning** | **Example** |
| = or eq | equal to | **where name='Jones, C.';** |
| ^= or ne | not equal to | **where temp ne 212;** |
| > or gt | greater than | **where income>20000;** |
| < or lt | less than | **where partno lt "BG05";** |
| >= or ge | greater than or equal to | **where id>='1543';** |
| <= or le | less than or equal to | **where pulse le 85;** |
| in | equal to one of a list | **if region in ('NE','NW','SW') then Rate=fee-25;** |

* Using the **CONTAINS** Operator

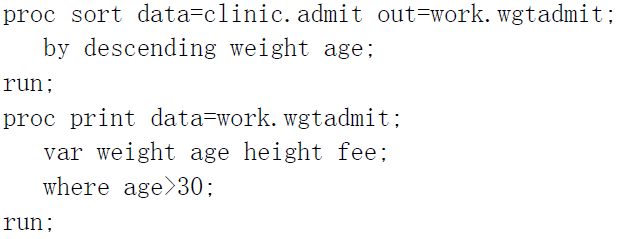
The **CONTAINS** operator selects observations that include the specified substring. The symbol for the **CONTAINS** operator is **?**

Eg:

**WHERE** firstname **CONTAINS** 'Jon'; 或者 **where** firstname ? 'Jon';

* **SORTING DATA**

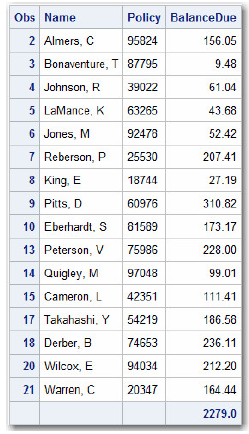
Must use **PROC SORT** before using the **PRINT**

Basic code:

**PROC SORT DATA=***SAS-data-set*<OUT=*SAS-data-set*>;

**BY** <DESCENDING> *BY-variable(s)*;

**RUN;**

* + - the **DATA**= option specifies the data set to be read
    - the **OUT**= option creates an output data set that contains the data in sorted order
    - ***BY-variable(s)***in the required **BY** statement specifies one or more variables whose values are used to sort the data
    - The **DESCENDING** option in the **BY** statement sorts observations in descending order. If you have more that one variable in the BY statement, **DESCENDING** applies only to the variable that immediately follows it. (50题，Q12)
    - **Caution**: If you don't use the **OUT**= option, **PROC SORT** overwrites the data set specified in the **DATA**= option.
*  **GENERATING COLUMN TOTALS**

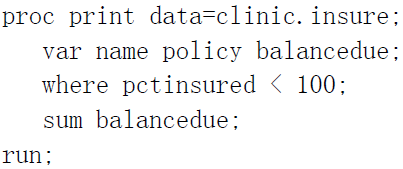
Produce column totals for numerical variables by using **SUM** statement

Basic code:

**SUM** *variable(s)***;**

*variable(s)* is one or more numeric variable names, separated by blanks.

Eg:



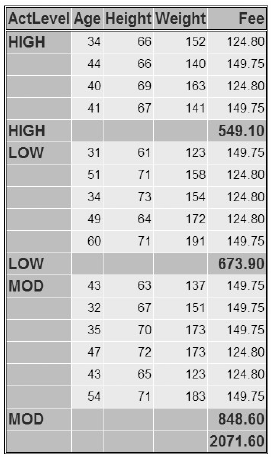
* **REQUESTING SUBTOTALS**
* Basic code:

**BY** <DESCENDING> *BY-variable-1*

<...<DESCENDING><*BY-variable-n*>>

<NOTSORTED>**;**

* ***BY-variable***specifies a variable that the procedure uses to form **BY** groups. You can specify more than one variable, separated by blanks.
* The **DESCENDING** option specifies that the data set is sorted in descending order by the variable that immediately follows.
* The **NOTSORTED** option specifies that observations are not necessarily sorted in alphabetic or numeric order. If observations that have the same values for the **BY** variables are not contiguous, the procedure treats each contiguous set as a separate BY group.
* **Caution**: If you do not use the NOTSORTED option in the BY statement, the observations in the data set must either be sorted by all the variables that you specify, or they must be indexed appropriately.
* 注意和**PROC** **SORT**中的**BY**区分开来。

 Eg:

**PROC SORT** data=clinic.admit out=work.activity;

**BY** actlevel;

**RUN**;

**PROC PRINT** data=work.activity;

**VAR** age height weight fee;

**WHERE** age>30;

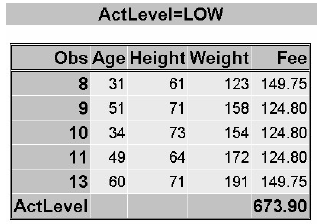
**SUM** fee;

**BY** actlevel;

**ID** actlevel;

**RUN**;

如果没有**ID** statement，表格会被**BY** actlevel statement按照不同的actlevel被分成若干相互独立的小表格。而增加**ID** statement可以把不同的表格合成为一个整体避免一个变量在不同表格中重复出现，并且因为**ID** statement会重复显示已经specified的variable（例子中为actlevel）一下为去掉**ID** statement之后的output，为三个分别得表格：



* Requesting subtotal on separate pages using **PAGEBY** statement
* 注意: The variable specified in the **PAGEBY** statement must also be specified in the BY statement in the **PROC PRINT** step.
* The PAGEBY statement in the program below prints BY groups for the variable ActLevel separately. The BY groups appear separated by horizontal lines in the HTML output
* Eg:

**PROC SORT** data=clinic.admit out=work.activity;

**BY** actlevel;

**RUN**;

**PROC PRINT** data=work.activity;

**VAR** age height weight fee;

**WHERE** age>30;

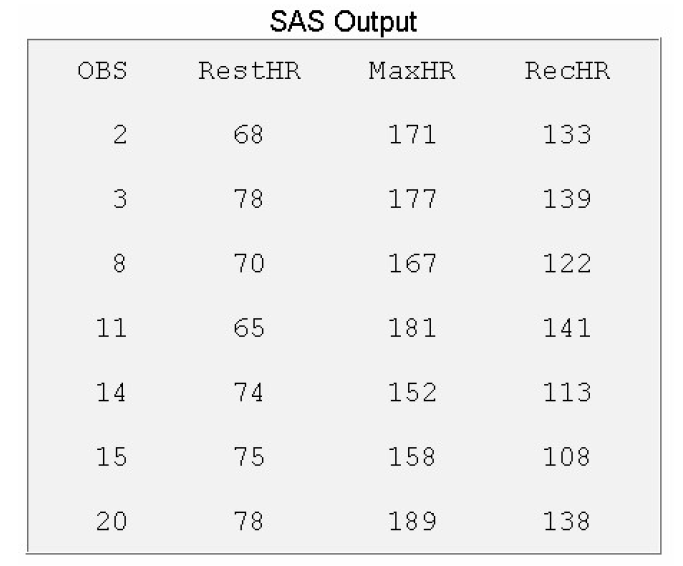
**SUM** fee;

**BY** actlevel;

**ID** actlevel;

**PAGEBY** actlevel;

**RUN**;

*  **DOUBLE SPACING LISTING OUTPUT**

Double space does not apply to HTML output

* Using **DOUBLE** option

Eg:

**PROC PRINT** data=clinic.stress **DOUBLE**;

**VAR** resthr maxhr rechr;

**WHERE** tolerance='I';

**RUN**;

* Using SAS window

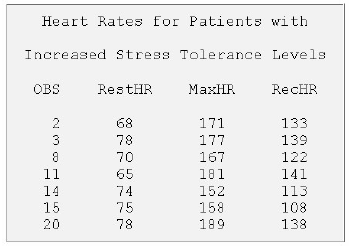
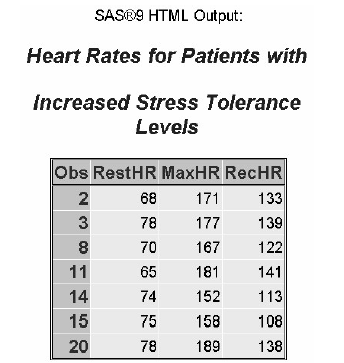
**Tool** – **Option** – **Preferences** – **Results** tab – **Create listing** option

* **TITLE & FOOTNOTE statement** 
  + Up to 10 **TITLE** and 10 **FOOTNOTE**.
  + **TITLE & FOOTNOTE** are global, they can be assigned within or before **PRINT** procedure
  + If TITLE statements are not specified, the automatic title is “The SAS System”, FOOTNOTE will not be automatically printed
* Basic Code (**TITLE & FOOTNOTE** coding 很相近，以**TITLE**为例):

**TITLE**<*n*> *'text'***;**

**FOOTNOTE**<*n*> *'text'***;**

*n* is a number from 1 to 10 that specifies the title or footnote line, and *'text'* is the actual title or footnote to be displayed. The maximum title or footnote length depends on your operating environment and on the value of the LINESIZE= option.

 Eg:

**TITLE1** 'Heart Rates for Patients with';

**TITLE3** 'Increased Stress Tolerance Levels';

**PROC PRINT** data=clinic.stress;

**VAR** resthr maxhr rechr;

**WHERE** tolerance='I';

**RUN**;

* Redefining a title or footnote line cancels any higher-numbered title or footnote lines, respectively. In the example below, defining a title for line 2 in the second report automatically cancels title line 3.

Eg:

**TITLE3** 'Participation in Exercise Therapy';

**PROC PRINT** data=clinic.therapy;

**VAR** swim walkjogrun aerclass;

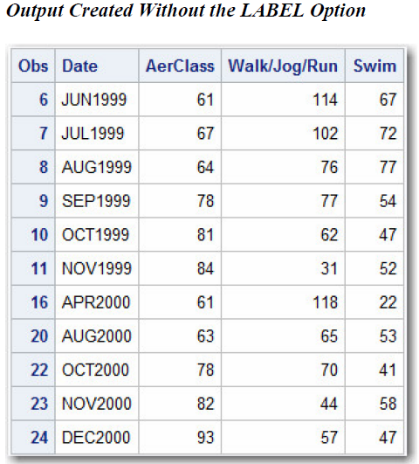
**RUN**;

**TITLE2** 'Report for March';

**PROC** **PRINT** data=clinic.therapy;

**RUN**;

To cancel all previous titles or footnotes, specify a null TITLE or FOOTNOTE statement or a TITLE1 or FOOTNOTE1 statement with no text. This will also cancel the default title The SAS System.

* **TEMPORIARILY ASSIGN LABELS AND FORMATS**
* Assign labels

**LABEL** statement可以用来对variable做一个进一步的解释，在**PROC PRINT**中specify **LABEL** statement to display the label。

Eg:

**PROC PRINT** data=clinic.therapy label;

**LABEL** walkjogrun=Walk/Jog/Run;

**RUN**;

* Formatting data value

**FORMAT** statement does not effect the value that has been stored in the SAS data set, it only effect how the data value is displayed

It applied only to the **PROC** step

* Basic code:

**FORMAT** *variable(s) format-name***;**

*variable(s)* is the name of one or more variables whose values are to be written according to a particular pattern

*format-name* specifies a SAS format or a user-defined format that is used to write out the values.

Eg:

**PROC PRINT** data=clinic.admit;

**VAR** actlevel fee;

**WHERE** actlevel='HIGH';

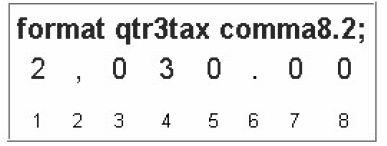
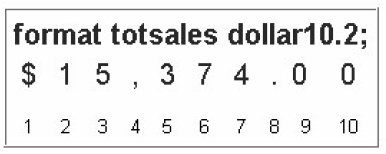
**FORMAT** fee dollar4.;

**RUN**;

|  |  |  |
| --- | --- | --- |
| Format | Specifies values | Example |
| **COMMAw.d** | That contains commas and decimal places | COMMA8.2 |
| **DOLLARw.d** | That contain dollar signs, commas, and decimal places | dollar6.2 |
| **MMDDYYw.** | as date values of the form 09/12/97 (MMDDYY8.) or 09/12/1997 (MMDDYY10.) | mmddyy10 |
| **w.** | rounded to the nearest integer in **w** spaces | 7. |
| **w.d** | rounded to d decimal places in **w** spaces | 8.2 |
| **$w.** | as character values in **w** spaces | $12. |
| **DATEw.** | as date values of the form 16OCT99 (DATE7.) or 16OCT1999 (DATE9.) | date9. |

* Field widths and Decimal place

When you use a SAS format, be sure to specify a field width (w) that is wide enough for the largest possible value. Otherwise, values might not be displayed properly. You must count the comma, because it occupies a position in the output.

Note for DATE (因为SAS DATE纪年从1960年1月1日开始，此处须和YEARCUTOFF= Option的1582年区分开来):

|  |  |  |
| --- | --- | --- |
| Stored value | Format | Displayed Value |
| 0 | MMDDYY8 | 01/01/60 |
| 0 | MMDDYY10 | 01/01/1960 |
| 0 | MMDDYY7 | 01JAN60 |
| 0 | MMDDYY9 | 01JAN1960 |

* **PERMANENTLY ASSIGN LABELS AND FORMATS**

当我们在**PROC** step中assign **LABEL**或者**FORMAT**，**LABEL**和**FORMAT** will be saved temporarily。当我们在**DATA** step中assign它们时，它们将会被saved permanently

Eg:

**DATA** sasuser.paris;

**SET** sasuser.laguardia;

**WHERE** dest="PAR" and (boarded=155 or boarded=146);

**LABEL** date=’Departure Date’;

**FORMAT** date date9.;

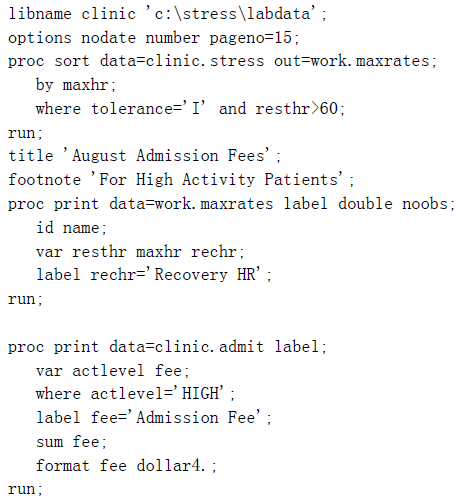
**RUN**;

**PROC** **PRINT** data=sasuser.paris;

**VAR** date dest boarded;

**RUN**;

1. Simple program example：



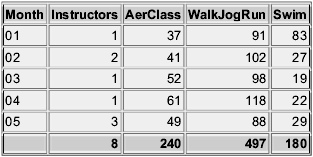
1. Points to Remember

* **VAR**, **WHERE**, **SUM**, **FORMAT** and **LABEL** statements remain in effect only for the **PROC** step in which they appear.
* If you don't use the **OUT**= option, **PROC** **SORT** permanently sorts the data set specified in the **DATA**= option.
* **TITLE** and **FOOTNOTE** statements remain in effect until you modify them, cancel them, or end your SAS session.
* Be sure to match the quotation marks that enclose the text in **TITLE**, **FOOTNOTE**, and **LABEL** statements.
* To display labels in **PRINT** procedure output, remember to add the **LABEL** option to the **PROC** **PRINT** statement.
* To permanently assign labels or formats to data set variables, place the **LABEL** or **FORMAT** statement inside the **DATA** step.

**练习**

1. Which of the following statements selects from a data set only those observations for which the value of the variable Style is RANCH, SPLIT, or TWOSTORY?

In the WHERE statement, the IN operator enables you to select observations based on several values. You specify values in parentheses and separate them by spaces or commas. Character values must be enclosed in quotation marks and must be in the same case as in the data set.

1. where style='RANCH' or 'SPLIT' or 'TWOSTORY';
2. where style in 'RANCH' or 'SPLIT' or 'TWOSTORY';
3. where style in (RANCH, SPLIT, TWOSTORY);
4. where style in ('RANCH','SPLIT','TWOSTORY');
5. Which of the following statements can you use in a **PROC** **PRINT** step to create this output?
6. **VAR** month instructors;

**SUM** instructors aerclass walkjogrun swim;

1. **VAR** month;

**SUM** instructors aerclass walkjogrun swim;

1. **VAR** month instructors aerclass;

**SUM** instructors aerclass walkjogrun swim;

You do not need to name the variables in a **VAR** statement if you specify them in the **SUM** statement, but you can. If you choose not to name the variables in the **VAR** statement as well, then the **SUM** statement determines the order of the variables in the output.

1. all of the above
2. What happens if you submit the following program?

**PROC** **SORT** data=clinic.diabetes;

**RUN**;

The BY statement is required in PROC SORT. Without it, the PROC SORT step fails. However, the PROC PRINT step prints the original data set as requested.

**PROC** **PRINT** data=clinic.diabetes;

**VAR** age height weight pulse;

**WHERE** sex='F';

**RUN**;

1. The **PROC** **PRINT** step runs successfully, printing observations in their sorted order.
2. The **PROC** **SORT** step permanently sorts the input data set.
3. The **PROC** **SORT** step generates errors and stops processing, but the **PROC** **PRINT** step runs successfully, printing observations in their original (unsorted) order.
4. The **PROC** **SORT** step runs successfully, but the **PROC** **PRINT** step generates errors and stops processing.
5. If you submit the following program, which output does it create?

**PROC** **SORT** data=finance.loans **OUT**=work.loans;

**BY** months amount;

Column totals appear at the end of the report in the same format as the values of the variables, so b is incorrect. Work.Loans is sorted by Month and Amount, so c is incorrect. The program sums both Amount and Payment, so d is incorrect.

**RUN**;

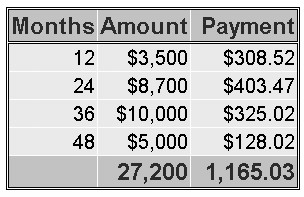
**PROC** **PRINT** data=work.loans noobs;

**VAR** months;

**SUM** amount payment;

**WHERE** months<360;

**RUN**;

1. 
2. 
3. 
4. 
5. What does **PROC PRINT** display by default?
6. **PROC PRINT** does not create a default report; you must specify the rows and columns to be displayed.
7. **PROC** **PRINT** displays all observations and variables in the data set. If you want an additional column for observation numbers, you can request it.
8. **PROC** **PRINT** displays columns in the following order: a column for observation numbers, all character variables, and all numeric variables.
9. **PROC** **PRINT** displays all observations and variables in the data set, a column for observation numbers on the far left, and variables in the order in which they occur in the data set.