Chapter 8 CREATING ENHANCED LIST AND

SUMMARY REPORTS

A more sophisticated way to produce a listing of the report is to use **PROC REPORT**. The syntax is

proc report data=*dataname* <options>;

<column statement>;

<where statement>;

<define statement>;

run;

Options include

* **WINDOWS** (or **WD**) – the report appears in the REPORT window (this is a default in SAS).
* **NOWINDOWS (**or **NOWD**) – the report appears in the OUTPUT window, and no REPORT window is opened.

Example. In the exercise example, type

proc report nowd;

run;

The output displayed in the OUTPUT window is

id age actlevel gender payment

2810 61 MOD F 34

2804 38 HIGH F 29

2807 42 LOW M 27.5

2816 26 HIGH M 20.25

2833 32 MOD F 15.5

2823 29 HIGH M 23

* The **COLUMN statement** may be used to select and order the variables to be displayed in the report.

Example. In the exercise example, type

proc report nowd;

column id actlevel payment gender age;

run;

The output displayed in the OUTPUT window is

id actlevel payment gender age

2810 MOD 34 F 61

2804 HIGH 29 F 38

2807 LOW 27.5 M 42

2816 HIGH 20.25 M 26

2833 MOD 15.5 F 32

2823 HIGH 23 M 29

* The **WHERE statement** may be used to select certain observations that appear in the report.

Example. In the exercise example, type

proc report nowd;

where gender='F';

run;

The output displayed in the OUTPUT window is

id age actlevel gender payment

2810 61 MOD F 34

2804 38 HIGH F 29

2833 32 MOD F 15.5

* The **DEFINE statement** may be used to describe how to display variables in the report. The syntax is

define variablename / <usage> <attributes> <options>

<justification><heading>;

where

* usage specifies how to use the variable. Valid options are DISPLAY, ANALYSIS, ORDER, GROUP, ACROSS, or COMPUTED.
* attributes specifies attributes for the variable, including FORMAT, WIDTH and SPACING.
* options specifies formatting options, including DESCENDING and NOPRINT.
* justification specifies column justification (CENTER, LEFT, or RIGHT).
* heading specifies a label for the column heading.
* Note that usage, attributes, options, justification, and label may be listed in any order.

DEFINING VARIABLE USAGE

PROC REPORT uses each variable in one of six ways (DISPLAY, ANALYSIS, ORDER, GROUP, ACROSS, or COMPUTED).

* By default, SAS uses character variables as **display variables**, and numeric variables as **analysis variables**, which are used to calculate the SUM statistic.
* Rows in a report are ordered according to an **order variable**.

Example. In the exercise example, type

proc report nowd;

define actlevel/order;

run;

The output is

id age actlevel gender payment

2804 38 HIGH F 29

2816 26 M 20.25

2823 29 M 23

2807 42 LOW M 27.5

2810 61 MOD F 34

2833 32 F 15.5

By default, the order is ascending. It may be made descending by typing

proc report nowd;

define actlevel/order descending;

run;

The output is

id age actlevel gender payment

2810 61 MOD F 34

2833 32 F 15.5

2807 42 LOW M 27.5

2804 38 HIGH F 29

2816 26 M 20.25

2823 29 M 23

* Totals are computed for all numeric variables for different values of **group variables.**

Example. In the exercise example, type

proc report nowd;

column actlevel gender payment;

define actlevel/group;

define gender/group;

run;

Total payments are computed for all levels of actlevel and gender under each level of actlevel. The output is

actlevel gender payment

HIGH F 29

M 43.25

LOW M 27.5

MOD F 49.5

* Frequencies for each value of **across variables** are computed, and sums for all numeric variables are given.

Example. In the exercise example, type

proc report nowd;

column actlevel gender payment;

define actlevel/across;

define gender/across;

run;

The output is

acltevel gender

HIGH LOW MOD F M payment

3 1 2 3 3 149.25

* A **computed variable** is not added to the data set, but is calculated and displayed in the report. In the program, include the name of a computed variable in the COLUMN statement to the right of variables used in calculation; define the variable as COMPUTED in the DEFINE statement; and compute the value of the variable in the COMPUTE block. The COMPUTE block is closed with an ENDCOMP statement. The variables used for calculations should be named with **compound names** that include the original variable name, period delimiter, and the statistics (sum, mean, max, min, stdev, or others).

Example. In the exercise example, type

data exercise;

input id $ age actlevel $ gender $ payment months;

cards;

2810 61 MOD F 34.00 1

2804 38 HIGH F 29.00 3

2807 42 LOW M 27.50 2

2816 26 HIGH M 20.25 4

2833 32 MOD F 15.50 2

2823 29 HIGH M 23.00 1

;

title;

options nonumber nodate;

proc report nowd;

column id payment months totalpayment;

define id/group;

define totalpayment/computed;

compute totalpayment;

totalpayment=payment.sum\*months.sum;

endcomp;

run;

The output is

totalpaym

id payment months ent

2804 29 3 87

2807 27.5 2 55

2810 34 1 34

2816 20.25 4 81

2823 23 1 23

2833 15.5 2 31

DEFINING COLUMN ATTRIBUTES

* To assign a format to a specific report column, use the **FORMAT=** in the DEFINE statement for that column.

Example. In the exercise example, type

proc report nowd;

column id payment;

define payment/ format=dollar6.2;

run;

The output is

paymen

ID t

2810 $34.00

2804 $29.00

2807 $27.50

2816 $20.25

2833 $15.50

2823 $23.00

* To specify width for columns in the report, use the **WIDTH=** in the DEFINE statement. It is typically used for the headings to appear on one line.

Example. In the exercise example, type

proc report nowd;

column id payment;

define payment/ format=dollar6.2 width=7;

run;

The output is

id payment

2810 $34.00

2804 $29.00

2807 $27.50

2816 $20.25

2833 $15.50

2823 $23.00

* To specify the number of blank characters between the selected column and the column immediately to its left, use the **SPACING=** in the DEFINE statement. The default spacing is 2 characters.

Example. In the exercise example, if no spacing is specified, the report looks like this:

id age actlevel gender payment

2810 61 MOD F 34

2804 38 HIGH F 29

2807 42 LOW M 27.5

2816 26 HIGH M 20.25

2833 32 MOD F 15.5

2823 29 HIGH M 23

If the following code is used

proc report nowd;

column id age actlevel gender payment;

define actlevel/spacing=10;

define gender/spacing=10;

run;

the columns are more spread out

id age actlevel gender payment

2810 61 MOD F 34

2804 38 HIGH F 29

2807 42 LOW M 27.5

2816 26 HIGH M 20.25

2833 32 MOD F 15.5

2823 29 HIGH M 23

DEFINING COLUMN OPTIONS

* The **DESCENDING option** is used in the DEFINE statement for an order variable, if it is desired to group the data in accordance with values of the variable sorted in descending order (see above).
* The **NOPRINT option** is used in the DEFINE statement for a variable that will not be displayed in the report.

Example. In the exercise example, if the following DEFINE statement is used

proc report nowd;

column id age actlevel gender payment;

define id/noprint;

run;

the column ID is not printed in the report

age actlevel gender payment

61 MOD F 34

38 HIGH F 29

42 LOW M 27.5

26 HIGH M 20.25

32 MOD F 15.5

29 HIGH M 23

DEFINING COLUMN JUSTIFICATION

Characters within a column may be positioned using the **CENTER**, **LEFT**, or **RIGHT** justification options in the DEFINE statement. By default, character variables are left-positioned, while numeric variables are right-positioned.

Example. In the exercise example, if no justification options are given, the output looks like this:

id age actlevel gender payment

2810 61 MOD F 34

2804 38 HIGH F 29

2807 42 LOW M 27.5

2816 26 HIGH M 20.25

2833 32 MOD F 15.5

2823 29 HIGH M 23

Note that id, actlevel, and gender are left-positioned, and age and payment are right-positioned. Suppose it is desired to center-position all columns. It is possible to change positioning for numeric variables only if they are formatted. The following SAS program achieves the desired centering.

proc report nowd;

column id age actlevel gender payment;

define id/center;

define actlevel/center;

define gender/center;

define age/format=2.0 width=3 center;

define payment/format=dollar6.2 width=7 center;

run;

The output is

id age actlevel gender payment

2810 61 MOD F $34.00

2804 38 HIGH F $29.00

2807 42 LOW M $27.50

2816 26 HIGH M $20.25

2833 32 MOD F $15.50

2823 29 HIGH M $23.00

DEFINING COLUMN HEADINGS

To define a column heading, enclose the heading text in single quotation marks in the DEFINE statement.

* To display headings on one line, specify the column width properly.

Example. In the exercise example, label the ActLevel column as Activity Level and make the column 15 characters wide; and label the Payment column as Payment Amount Due and make the column 20 characters wide.

proc report nowd;

column id age actlevel gender payment;

define actlevel/ width=15 'Activity Level';

define payment/width=20 'Payment Amount Due';

run;

The output is

id age Activity Level gender Payment Amount Due

2810 61 MOD F 34

2804 38 HIGH F 29

2807 42 LOW M 27.5

2816 26 HIGH M 20.25

2833 32 MOD F 15.5

2823 29 HIGH M 23

* To **split** long column headings across two lines, use a **split character** in the column label. A default split character is forward slash (/). A custom split character may be used instead of the slash, but prior to use it has to be defined in the PROC REPORT statement with **SPLIT=** option.

Example. In the exercise example, type

proc report nowd;

column id age actlevel gender payment;

define actlevel/ width=15 **'Activity/ Level'**;

define payment/width=20 **'Payment/ Amount Due'**;

run;

or

proc report nowd **split='&'**;

column id age actlevel gender payment;

define actlevel/ width=15 **'Activity & Level'**;

define payment/width=20 **'Payment & Amount Due'**;

run;

The output is

Activity Payment

id age Level gender Amount Due

2810 61 MOD F 34

2804 38 HIGH F 29

2807 42 LOW M 27.5

2816 26 HIGH M 20.25

2833 32 MOD F 15.5

2823 29 HIGH M 23

* To enhance headings further, two options in the PROC REPORT statement may be used:
* **HEADLINE**, which underlines all column headings and the spaces between them.
* **HEADSKIP**, which writes a blank line under all column headings or after the underline, if the HEADLINE option is used.

Example. In the exercise example, type

proc report nowd headline headskip;

column id age actlevel gender payment;

define actlevel/ width=15 'Activity Level';

define payment/width=20 'Payment Amount Due';

run;

The output is

id age Activity Level gender Payment Amount Due ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ

2810 61 MOD F 34

2804 38 HIGH F 29

2807 42 LOW M 27.5

2816 26 HIGH M 20.25

2833 32 MOD F 15.5

2823 29 HIGH M 23