

# **Chapter 4**

## **Creating List Reports**

# Topics

- Specify SAS data sets to print.
- Select variables and observations to print.
- Sort data by the values of one or more variables.
- Specify column totals for numeric variables.
- Double spacing SAS listing output.
- Add titles and footnotes to output.
- Assign descriptive labels to variables.
- Apply formats to the values of variables.

# Creating List Reports

- One of the easiest ways to list the contents in a SAS data set is to create a report with PROC PRINT.
- The defaults of the PRINT procedure are simple, but additional data management can be performed, e.g.,
  - ✓ Sort the observations
  - ✓ Label variables
  - ✓ Select a subset of variables and/or observations

# Creating List Reports

```
options pagesize=500
linesize=200;
PROC PRINT data =
sashelp.cars;
run;
```

By default,

- All observations and variables are printed
- They appear in the same order as in the data set
- Observation numbers appear in the far left

The SAS System

| Obs | Make  | Model                          | Type   | Origin | Drive Train | MSRP     | Invo   |
|-----|-------|--------------------------------|--------|--------|-------------|----------|--------|
| 1   | Acura | MDX                            | SUV    | Asia   | All         | \$36,945 | \$33,3 |
| 2   | Acura | RSX Type S 2dr                 | Sedan  | Asia   | Front       | \$23,820 | \$21,7 |
| 3   | Acura | TSX 4dr                        | Sedan  | Asia   | Front       | \$26,990 | \$24,6 |
| 4   | Acura | TL 4dr                         | Sedan  | Asia   | Front       | \$33,195 | \$30,2 |
| 5   | Acura | 3.5 RL 4dr                     | Sedan  | Asia   | Front       | \$43,755 | \$39,0 |
| 6   | Acura | 3.5 RL w/Navigation 4dr        | Sedan  | Asia   | Front       | \$46,100 | \$41,1 |
| 7   | Acura | NSX coupe 2dr manual S         | Sports | Asia   | Rear        | \$89,765 | \$79,9 |
| 8   | Audi  | A4 1.8T 4dr                    | Sedan  | Europe | Front       | \$25,940 | \$23,5 |
| 9   | Audi  | A4 1.8T convertible 2dr        | Sedan  | Europe | Front       | \$35,940 | \$32,5 |
| 10  | Audi  | A4 3.0 4dr                     | Sedan  | Europe | Front       | \$31,840 | \$28,8 |
| 11  | Audi  | A4 3.0 Quattro 4dr manual      | Sedan  | Europe | All         | \$33,430 | \$30,3 |
| 12  | Audi  | A4 3.0 Quattro 4dr auto        | Sedan  | Europe | All         | \$34,480 | \$31,3 |
| 13  | Audi  | A6 3.0 4dr                     | Sedan  | Europe | Front       | \$36,640 | \$33,1 |
| 14  | Audi  | A6 3.0 Quattro 4dr             | Sedan  | Europe | All         | \$39,640 | \$35,9 |
| 15  | Audi  | A4 3.0 convertible 2dr         | Sedan  | Europe | Front       | \$42,490 | \$38,3 |
| 16  | Audi  | A4 3.0 Quattro convertible 2dr | Sedan  | Europe | All         | \$44,240 | \$40,0 |
| 17  | Audi  | A6 2.7 Turbo Quattro 4dr       | Sedan  | Europe | All         | \$42,840 | \$38,8 |
| 18  | Audi  | A6 4.2 Quattro 4dr             | Sedan  | Europe | All         | \$49,690 | \$44,9 |
| 19  | Audi  | A8 L Quattro 4dr               | Sedan  | Europe | All         | \$69,190 | \$64,7 |

# Creating List Reports: the **VAR** statement

You can select the variables of interest and control the order in which they appear by using a VAR statement.

```
PROC PRINT data = sashelp.cars;
    var make model weight mpg_city;
run;
```

The SAS System  
11:57 Sunday, September 15, 201

| Obs | Make  | Model                          | Weight | MPG_City |
|-----|-------|--------------------------------|--------|----------|
| 1   | Acura | MDX                            | 4451   | 17       |
| 2   | Acura | RSX Type S 2dr                 | 2778   | 24       |
| 3   | Acura | TSX 4dr                        | 3230   | 22       |
| 4   | Acura | TL 4dr                         | 3575   | 20       |
| 5   | Acura | 3.5 RL 4dr                     | 3880   | 18       |
| 6   | Acura | 3.5 RL w/Navigation 4dr        | 3893   | 18       |
| 7   | Acura | NSX coupe 2dr manual S         | 3153   | 17       |
| 8   | Audi  | A4 1.8T 4dr                    | 3252   | 22       |
| 9   | Audi  | A4 1.8T convertible 2dr        | 3638   | 23       |
| 10  | Audi  | A4 3.0 4dr                     | 3462   | 20       |
| 11  | Audi  | A4 3.0 Quattro 4dr manual      | 3583   | 17       |
| 12  | Audi  | A4 3.0 Quattro 4dr auto        | 3627   | 18       |
| 13  | Audi  | A6 3.0 4dr                     | 3561   | 20       |
| 14  | Audi  | A6 3.0 Quattro 4dr             | 3880   | 18       |
| 15  | Audi  | A4 3.0 convertible 2dr         | 3814   | 20       |
| 16  | Audi  | A4 3.0 Quattro convertible 2dr | 4013   | 18       |
| 17  | Audi  | A6 2.7 Turbo Quattro 4dr       | 3836   | 18       |
| 18  | Audi  | A6 4.2 Quattro 4dr             | 4024   | 17       |
| 19  | Audi  | A8 L Quattro 4dr               | 4399   | 17       |
| 20  | Audi  | S4 Quattro 4dr                 | 3825   | 14       |
| 21  | Audi  | RS 6 4dr                       | 4024   | 15       |
| 22  | Audi  | TT 1.8 convertible 2dr (coupe) | 3131   | 20       |

# Creating List Reports: the **WHERE** statement

You can select observations by using a WHERE statement.

```
Options linesize=80;
PROC PRINT data = sashelp.cars;
    var make model weight mpg_city;
    where weight>5000;
run;
```

The SAS System

11:57 Sunday, September 15, 20

| Obs | Make          | Model                 | Weight | MPG_City |
|-----|---------------|-----------------------|--------|----------|
| 56  | Cadillac      | Escalade              | 5367   | 14       |
| 63  | Cadillac      | Escalade EXT          | 5879   | 13       |
| 65  | Chevrolet     | Tahoe LT              | 5050   | 14       |
| 85  | Chevrolet     | Avalanche 1500        | 5678   | 14       |
| 119 | Ford          | Excursion 6.8 XLT     | 7190   | 10       |
| 138 | Ford          | F-150 Supercab Lariat | 5464   | 14       |
| 143 | GMC           | Yukon 1500 SLE        | 5042   | 16       |
| 144 | GMC           | Yukon XL 2500 SLT     | 6133   | 13       |
| 148 | GMC           | Sierra HD 2500        | 5440   | 13       |
| 167 | Hummer        | H2                    | 6400   | 10       |
| 216 | Land Rover    | Range Rover HSE       | 5379   | 12       |
| 220 | Lexus         | LX 470                | 5590   | 13       |
| 230 | Lincoln       | Navigator Luxury      | 5969   | 13       |
| 252 | Mercedes-Benz | G500                  | 5423   | 13       |
| 300 | Nissan        | Pathfinder Armada SE  | 5013   | 13       |
| 315 | Nissan        | Titan King Cab XE     | 5287   | 14       |
| 375 | Toyota        | Sequoia SR5           | 5270   | 14       |
| 378 | Toyota        | Land Cruiser          | 5390   | 13       |
| 402 | Volkswagen    | Touareg V6            | 5086   | 15       |
| 412 | Volkswagen    | Phaeton 4dr           | 5194   | 16       |
| 413 | Volkswagen    | Phaeton W12 4dr       | 5399   | 12       |

# Creating List Reports: the WHERE statement

A list of **comparison operators** that can be used in WHERE statements.

| Symbol                    | Meaning                  | Example                    |
|---------------------------|--------------------------|----------------------------|
| <b>=</b> or <b>eq</b>     | equal to                 | where<br>name='Jones, C.'; |
| <b>^=</b> or <b>ne</b>    | not equal to             | where temp ne<br>212;      |
| <b>&gt;</b> or <b>gt</b>  | greater than             | where<br>income>20000;     |
| <b>&lt;</b> or <b>lt</b>  | less than                | where partno lt<br>"BG05"; |
| <b>&gt;=</b> or <b>ge</b> | greater than or equal to | where id>='1543';          |
| <b>&lt;=</b> or <b>le</b> | less than or equal to    | where pulse le 85;         |

# Creating List Reports: the WHERE statement

You can select observations based on multiple conditions by using **logical operators** in a **compound WHERE expression**.

| Operator |   | Meaning   |
|----------|---|---|
| AND      | & | and, both. If both expressions are true, then the compound expression is true.  |
| OR       |   | or, either. If either expression is true, then the compound expression is true. |



# Creating List Reports: the WHERE statement

Example 1: selecting observations based on multiple conditions by using **logical operators** in a **compound WHERE expression**.

```
PROC PRINT data = sashelp.cars;
    var make model weight mpg_city;
    where weight>3500 AND make ="Audi";
run;
```

| Obs | Make | Model                          | Weight | MPG_City |
|-----|------|--------------------------------|--------|----------|
| 9   | Audi | A4 1.8T convertible 2dr        | 3638   | 23       |
| 11  | Audi | A4 3.0 Quattro 4dr manual      | 3583   | 17       |
| 12  | Audi | A4 3.0 Quattro 4dr auto        | 3627   | 18       |
| 13  | Audi | A6 3.0 4dr                     | 3561   | 20       |
| 14  | Audi | A6 3.0 Quattro 4dr             | 3880   | 18       |
| 15  | Audi | A4 3.0 convertible 2dr         | 3814   | 20       |
| 16  | Audi | A4 3.0 Quattro convertible 2dr | 4013   | 18       |
| 17  | Audi | A6 2.7 Turbo Quattro 4dr       | 3836   | 18       |
| 18  | Audi | A6 4.2 Quattro 4dr             | 4024   | 17       |
| 19  | Audi | A8 L Quattro 4dr               | 4399   | 17       |
| 20  | Audi | S4 Quattro 4dr                 | 3825   | 14       |
| 21  | Audi | RS 6 4dr                       | 4024   | 15       |
| 25  | Audi | A6 3.0 Avant Quattro           | 4035   | 18       |
| 26  | Audi | S4 Avant Quattro               | 3936   | 15       |

# Creating List Reports: the WHERE statement

Example 2: selecting observations based on multiple conditions by using **logical operators** in a **compound WHERE expression** – a more complicated situation

```
PROC PRINT data = sashelp.cars;
  var make model weight mpg_city;
  where weight>3500 AND (make ="Audi" OR make="BMW");
run;
```

| Obs | Make | Model                          | Weight | MPG_City |
|-----|------|--------------------------------|--------|----------|
| 9   | Audi | A41.8T convertible 2dr         | 3638   | 23       |
| 11  | Audi | A4 3.0 Quattro 4dr manual      | 3583   | 17       |
| 12  | Audi | A4 3.0 Quattro 4dr auto        | 3627   | 18       |
| 13  | Audi | A6 3.0 4dr                     | 3561   | 20       |
| 14  | Audi | A6 3.0 Quattro 4dr             | 3880   | 18       |
| 15  | Audi | A4 3.0 convertible 2dr         | 3814   | 20       |
| 16  | Audi | A4 3.0 Quattro convertible 2dr | 4013   | 18       |
| 17  | Audi | A6 2.7 Turbo Quattro 4dr       | 3836   | 18       |
| 18  | Audi | A6 4.2 Quattro 4dr             | 4024   | 17       |
| 19  | Audi | A8 L Quattro 4dr               | 4399   | 17       |
| 20  | Audi | S4 Quattro 4dr                 | 3825   | 14       |
| 21  | Audi | RS 6 4dr                       | 4024   | 15       |
| 25  | Audi | A6 3.0 Avant Quattro           | 4035   | 18       |
| 26  | Audi | S4 Avant Quattro               | 3936   | 15       |
| 27  | BMW  | X3 3.0i                        | 4023   | 16       |
| 28  | BMW  | X5 4.4i                        | 4824   | 16       |
| 31  | BMW  | 325Ci convertible 2dr          | 3560   | 19       |
| 37  | BMW  | 330Ci convertible 2dr          | 3616   | 19       |
| 39  | BMW  | 545iA 4dr                      | 3814   | 18       |
| 40  | BMW  | 745i 4dr                       | 4376   | 18       |
| 41  | BMW  | 745Li 4dr                      | 4464   | 18       |
| 43  | BMW  | M3 convertible 2dr             | 3781   | 16       |
| 46  | BMW  | 325xi Sport                    | 3594   | 19       |

# Creating List Reports: the WHERE statement

Example 3: selecting observations based on multiple conditions by using **logical operators** in a **compound WHERE expression** – a more complicated situation using the **IN operator**.

```
PROC PRINT data = sashelp.cars;
    var make model weight mpg_city;
    where weight>3500 AND make IN ("Audi", "BMW");
run;
```

| Obs | Make | Model                          | Weight | MPG_City |
|-----|------|--------------------------------|--------|----------|
| 9   | Audi | A4 1.8T convertible 2dr        | 3638   | 23       |
| 11  | Audi | A4 3.0 Quattro 4dr manual      | 3583   | 17       |
| 12  | Audi | A4 3.0 Quattro 4dr auto        | 3627   | 18       |
| 13  | Audi | A6 3.0 4dr                     | 3561   | 20       |
| 14  | Audi | A6 3.0 Quattro 4dr             | 3880   | 18       |
| 15  | Audi | A4 3.0 convertible 2dr         | 3814   | 20       |
| 16  | Audi | A4 3.0 Quattro convertible 2dr | 4013   | 18       |
| 17  | Audi | A6 2.7 Turbo Quattro 4dr       | 3836   | 18       |
| 18  | Audi | A6 4.2 Quattro 4dr             | 4024   | 17       |
| 19  | Audi | A8 L Quattro 4dr               | 4399   | 17       |
| 20  | Audi | S4 Quattro 4dr                 | 3825   | 14       |
| 21  | Audi | RS 6 4dr                       | 4024   | 15       |
| 25  | Audi | A6 3.0 Avant Quattro           | 4035   | 18       |
| 26  | Audi | S4 Avant Quattro               | 3936   | 15       |
| 27  | BMW  | X3 3.0i                        | 4023   | 16       |
| 28  | BMW  | X5 4.4i                        | 4824   | 16       |
| 31  | BMW  | 325Ci convertible 2dr          | 3560   | 19       |
| 37  | BMW  | 330Ci convertible 2dr          | 3616   | 19       |
| 39  | BMW  | 545iA 4dr                      | 3814   | 18       |
| 40  | BMW  | 745i 4dr                       | 4376   | 18       |
| 41  | BMW  | 745Li 4dr                      | 4464   | 18       |
| 43  | BMW  | M3 convertible 2dr             | 3781   | 16       |
| 46  | BMW  | 325xi Sport                    | 3594   | 19       |

# Creating List Reports: the WHERE statement

Using the **CONTAINS (?)** operator to select observations.

```
PROC PRINT data = sashelp.cars;
  var make model weight mpg_city;
  where weight>3500 AND (make ? "Au" OR make CONTAINS
    "BM");
run;
```

| Obs | Make | Model                          | Weight | MPG_City |
|-----|------|--------------------------------|--------|----------|
| 9   | Audi | A4 1.8T convertible 2dr        | 3638   | 23       |
| 11  | Audi | A4 3.0 Quattro 4dr manual      | 3583   | 17       |
| 12  | Audi | A4 3.0 Quattro 4dr auto        | 3627   | 18       |
| 13  | Audi | A6 3.0 4dr                     | 3561   | 20       |
| 14  | Audi | A6 3.0 Quattro 4dr             | 3880   | 18       |
| 15  | Audi | A4 3.0 convertible 2dr         | 3814   | 20       |
| 16  | Audi | A4 3.0 Quattro convertible 2dr | 4013   | 18       |
| 17  | Audi | A6 2.7 Turbo Quattro 4dr       | 3836   | 18       |
| 18  | Audi | A6 4.2 Quattro 4dr             | 4024   | 17       |
| 19  | Audi | A8 L Quattro 4dr               | 4399   | 17       |
| 20  | Audi | S4 Quattro 4dr                 | 3825   | 14       |
| 21  | Audi | RS 6 4dr                       | 4024   | 15       |
| 25  | Audi | A6 3.0 Avant Quattro           | 4035   | 18       |
| 26  | Audi | S4 Avant Quattro               | 3936   | 15       |
| 27  | BMW  | X3 3.0i                        | 4023   | 16       |
| 28  | BMW  | X5 4.4i                        | 4824   | 16       |
| 31  | BMW  | 325Ci convertible 2dr          | 3560   | 19       |
| 37  | BMW  | 330Ci convertible 2dr          | 3616   | 19       |
| 39  | BMW  | 545iA 4dr                      | 3814   | 18       |
| 40  | BMW  | 745i 4dr                       | 4376   | 18       |
| 41  | BMW  | 745Li 4dr                      | 4464   | 18       |
| 43  | BMW  | M3 convertible 2dr             | 3781   | 16       |
| 46  | BMW  | 325xi Sport                    | 3594   | 19       |

# Creating List Reports: the NOOBS option

If you do not want to display the observation numbers, specify the **NOOBS** option in the PROC PRINT statement.

```
Options linesize=80;
PROC PRINT data = sashelp.cars noobs;
    var make model weight mpg_city;
    where weight>5000;
run;
```

The SAS System

11:57 Sunday, September 15, 20

| Make          | Model                 | Weight | MPG_City |
|---------------|-----------------------|--------|----------|
| Cadillac      | Escalade              | 5367   | 14       |
| Cadillac      | Escalade EXT          | 5879   | 13       |
| Chevrolet     | Tahoe LT              | 5050   | 14       |
| Chevrolet     | Avalanche 1500        | 5678   | 14       |
| Ford          | Excursion 6.8 XLT     | 7190   | 10       |
| Ford          | F-150 Supercab Lariat | 5464   | 14       |
| GMC           | Yukon 1500 SLE        | 5042   | 16       |
| GMC           | Yukon XL 2500 SLT     | 6133   | 13       |
| GMC           | Sierra HD 2500        | 5440   | 13       |
| Hummer        | H2                    | 6400   | 10       |
| Land Rover    | Range Rover HSE       | 5379   | 12       |
| Lexus         | LX 470                | 5590   | 13       |
| Lincoln       | Navigator Luxury      | 5969   | 13       |
| Mercedes-Benz | G500                  | 5423   | 13       |
| Nissan        | Pathfinder Armada SE  | 5013   | 13       |
| Nissan        | Titan King Cab XE     | 5287   | 14       |
| Toyota        | Sequoia SR5           | 5270   | 14       |
| Toyota        | Land Cruiser          | 5390   | 13       |
| Volkswagen    | Touareg V6            | 5086   | 15       |
| Volkswagen    | Phaeton 4dr           | 5194   | 16       |
| Volkswagen    | Phaeton W12 4dr       | 5399   | 12       |

# Creating List Reports: the **ID** statement

You can also use one or more variables to replace the Obs column in the output with the **ID** statement.

```
Options linesize=80;
PROC PRINT data = sashelp.cars;
    id horsepower;
    var make model weight mpg_city;
    where weight>5000;
run;
```

The SAS System

11:57 Sunday, September 15, 2014

| Horsepower | Make          | Model                 | Weight | MPG_City |
|------------|---------------|-----------------------|--------|----------|
| 295        | Cadillac      | Escalade              | 5367   | 14       |
| 345        | Cadillac      | Escalade EXT          | 5879   | 13       |
| 295        | Chevrolet     | Tahoe LT              | 5050   | 14       |
| 295        | Chevrolet     | Avalanche 1500        | 5678   | 14       |
| 310        | Ford          | Excursion 6.8 XLT     | 7190   | 10       |
| 300        | Ford          | F-150 Supercab Lariat | 5464   | 14       |
| 285        | GMC           | Yukon 1500 SLE        | 5042   | 16       |
| 325        | GMC           | Yukon XL 2500 SLT     | 6133   | 13       |
| 300        | GMC           | Sierra HD 2500        | 5440   | 13       |
| 316        | Hummer        | H2                    | 6400   | 10       |
| 282        | Land Rover    | Range Rover HSE       | 5379   | 12       |
| 235        | Lexus         | LX 470                | 5590   | 13       |
| 300        | Lincoln       | Navigator Luxury      | 5969   | 13       |
| 292        | Mercedes-Benz | G500                  | 5423   | 13       |
| 305        | Nissan        | Pathfinder Armada SE  | 5013   | 13       |
| 305        | Nissan        | Titan King Cab XE     | 5287   | 14       |
| 240        | Toyota        | Sequoia SR5           | 5270   | 14       |
| 325        | Toyota        | Land Cruiser          | 5390   | 13       |
| 220        | Volkswagen    | Touareg V6            | 5086   | 15       |
| 335        | Volkswagen    | Phaeton 4dr           | 5194   | 16       |
| 420        | Volkswagen    | Phaeton W12 4dr       | 5399   | 12       |

# Creating List Reports: the ID statement

More example: using one or more variables to replace the Obs column in the output.

```
Options linesize=80;
PROC PRINT data = sasuser.customer;
    id customer_id customer_name;
    where country='CA';
run;
```

| The SAS System |                  |         |        | 11:57 Sunday, September |                    |                   |
|----------------|------------------|---------|--------|-------------------------|--------------------|-------------------|
| Customer_ID    | Customer_Name    | Country | Gender | Personal_ID             | Customer_FirstName | Customer_LastName |
| 11171          | Bill Cuddy       | CA      | M      |                         | Bill               | Cuddy             |
| 17023          | Susan Krasowski  | CA      | F      |                         | Susan              | Krasowski         |
| 26148          | Andreas Rennie   | CA      | M      |                         | Andreas            | Rennie            |
| 46966          | Lauren Krasowski | CA      | F      |                         | Lauren             | Krasowski         |
| 54655          | Lauren Marx      | CA      | F      |                         | Lauren             | Marx              |
| 70046          | Tommy Mcdonald   | CA      | M      |                         | Tommy              | Mcdonald          |
| 70059          | Colin Byarley    | CA      | M      |                         | Colin              | Byarley           |
| 70079          | Lera Knott       | CA      | F      |                         | Lera               | Knott             |
| 70100          | Wilma Yeargan    | CA      | F      |                         | Wilma              | Yeargan           |
| 70108          | Patrick Leach    | CA      | M      |                         | Patrick            | Leach             |
| 70165          | Portia Reynoso   | CA      | F      |                         | Portia             | Reynoso           |
| 70187          | Soberina Berent  | CA      | F      |                         | Soberina           | Berent            |
| 70201          | Angel Borwick    | CA      | F      |                         | Angel              | Borwick           |
| 70210          | Alex Santinello  | CA      | M      |                         | Alex               | Santinello        |
| 70221          | Kenan Talarr     | CA      | M      |                         | Kenan              | Talarr            |

# Sorting Data

You can use **PROC SORT** to sort your list of observation based on the value(s) of a BY variable(s).

The SORT procedure

- rearranges the observations in a SAS data set
- creates a new SAS data set that contains the rearranged observations
- replaces the original SAS data set by default
- can sort on multiple variables
- can sort in ascending or descending order
- does not generate printed output
- treats missing values as the smallest possible values



# Sorting Data

PROC SORT syntax:

```
PROC SORT DATA=SAS-data-set<OUT=SAS-data-set>;  
  BY <DESCENDING> BY-variable(s);  
RUN;
```

Note:

- The OUT= option creates an output data set that contains the data in sorted order so that the original data set is not affected.
- If you don't use the OUT= option, by default PROC SORT overwrites the data set specified in the DATA= option.
- BY-variable(s) specifies one or more variables whose values are used to sort the data.
- The DESCENDING option sorts observations in descending order. If you have more than one variable in the BY statement, **DESCENDING** applies only to the variable that immediately follows it.

# Sorting Data

## PROC SORT example 1:

```
proc sort data=clinic.admit out=work.wgtadmit;  
  by weight age;  
run;  
proc print data=work.wgtadmit;  
  var weight age height fee;  
  where age>30;  
run;
```

| Obs | Weight | Age | Height | Fee    |
|-----|--------|-----|--------|--------|
| 2   | 123    | 31  | 61     | 149.75 |
| 3   | 123    | 43  | 65     | 124.80 |
| 4   | 137    | 43  | 63     | 149.75 |
| 6   | 140    | 44  | 66     | 149.75 |
| 7   | 141    | 41  | 67     | 149.75 |
| 9   | 151    | 32  | 67     | 149.75 |
| 10  | 152    | 34  | 66     | 124.80 |
| 11  | 154    | 34  | 73     | 124.80 |
| 12  | 158    | 51  | 71     | 124.80 |
| 13  | 163    | 40  | 69     | 124.80 |
| 15  | 172    | 49  | 64     | 124.80 |
| 16  | 173    | 35  | 70     | 149.75 |
| 17  | 173    | 47  | 72     | 124.80 |
| 18  | 183    | 54  | 71     | 149.75 |
| 20  | 191    | 60  | 71     | 149.75 |

# Sorting Data

PROC SORT example 2: use the DESCENDING option

```
* The data set must be sorted first;
proc sort data=clinic.admit out=work.wgtadmit;
  by descending weight age;
run;
proc print data=work.wgtadmit;
  var weight age height fee;
  where age>30;
run;
```

| Obs | Weight | Age | Height | Fee    |
|-----|--------|-----|--------|--------|
| 2   | 191    | 60  | 71     | 149.75 |
| 4   | 183    | 54  | 71     | 149.75 |
| 5   | 173    | 35  | 70     | 149.75 |
| 6   | 173    | 47  | 72     | 124.80 |
| 7   | 172    | 49  | 64     | 124.80 |
| 9   | 163    | 40  | 69     | 124.80 |
| 10  | 158    | 51  | 71     | 124.80 |
| 11  | 154    | 34  | 73     | 124.80 |
| 12  | 152    | 34  | 66     | 124.80 |
| 13  | 151    | 32  | 67     | 149.75 |
| 15  | 141    | 41  | 67     | 149.75 |
| 16  | 140    | 44  | 66     | 149.75 |
| 18  | 137    | 43  | 63     | 149.75 |
| 19  | 123    | 31  | 61     | 149.75 |
| 20  | 123    | 43  | 65     | 124.80 |

# Generating Column Totals

You can use the **SUM statement** to produce column totals of numeric variables in a PROC PRINT step, for example:

```
proc print data=clinic.insure;
  var name policy balancedue;
  where pctinsured < 100;
  sum balancedue;
run;
```

| Obs | Name           | Policy | Balance Due |
|-----|----------------|--------|-------------|
| 2   | Almers, C      | 95824  | 156.05      |
| 3   | Bonaventure, T | 87795  | 9.48        |
| 4   | Johnson, R     | 39022  | 61.04       |
| 5   | LaMance, K     | 63265  | 43.68       |
| 6   | Jones, M       | 92478  | 52.42       |
| 7   | Reberson, P    | 25530  | 207.41      |
| 8   | King, E        | 18744  | 27.19       |
| 9   | Pitts, D       | 60976  | 310.82      |
| 10  | Eberhardt, S   | 81589  | 173.17      |
| 13  | Peterson, V    | 75986  | 228.00      |
| 14  | Quigley, M     | 97048  | 99.01       |
| 15  | Cameron, L     | 42351  | 111.41      |
| 17  | Takahashi, Y   | 54219  | 186.58      |
| 18  | Derber, B      | 74653  | 236.11      |
| 20  | Wilcox, E      | 94034  | 212.20      |
| 21  | Warren, C      | 20347  | 164.44      |
|     |                |        | =====       |
|     |                |        | 2279.0      |

# Generating Subtotals

You can use a **SUM statement** and a **BY statement** to produce subtotals and column totals of numeric variables in a PROC PRINT step, e.g.:

```
* The data set must be sorted first;  
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;  
run;
```

| ----- ActLevel=HIGH ----- |     |        |        |        |
|---------------------------|-----|--------|--------|--------|
| Obs                       | Age | Height | Weight | Fee    |
| 2                         | 34  | 66     | 152    | 124.80 |
| 4                         | 44  | 66     | 140    | 149.75 |
| 5                         | 40  | 69     | 163    | 124.80 |
| 7                         | 41  | 67     | 141    | 149.75 |
| -----                     |     |        |        | -----  |
| ActLevel                  |     |        |        | 549.10 |

| ----- ActLevel=LOW ----- |     |        |        |        |
|--------------------------|-----|--------|--------|--------|
| Obs                      | Age | Height | Weight | Fee    |
| 8                        | 31  | 61     | 123    | 149.75 |
| 9                        | 51  | 71     | 158    | 124.80 |
| 10                       | 34  | 73     | 154    | 124.80 |
| 11                       | 49  | 64     | 172    | 124.80 |
| 13                       | 60  | 71     | 191    | 149.75 |
| -----                    |     |        |        | -----  |
| ActLevel                 |     |        |        | 673.90 |

| ----- ActLevel=MOD ----- |     |        |        |         |
|--------------------------|-----|--------|--------|---------|
| Obs                      | Age | Height | Weight | Fee     |
| 15                       | 43  | 63     | 137    | 149.75  |
| 16                       | 32  | 67     | 151    | 149.75  |
| 17                       | 35  | 70     | 173    | 149.75  |
| 19                       | 47  | 72     | 173    | 124.80  |
| 20                       | 43  | 65     | 123    | 124.80  |
| 21                       | 54  | 71     | 183    | 149.75  |
| -----                    |     |        |        | -----   |
| ActLevel                 |     |        |        | 848.60  |
|                          |     |        |        | =====   |
|                          |     |        |        | 2071.60 |

# Generating Subtotals: Creating a Customized Layout with the ID Statement

You can add an ID statement to the SUM statement and a BY statement to produce a customized layout

```
* The data set must be sorted first;
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;
```

| Act Level | Age | Height | Weight | Fee     |
|-----------|-----|--------|--------|---------|
| HIGH      | 34  | 66     | 152    | 124.80  |
|           | 44  | 66     | 140    | 149.75  |
|           | 40  | 69     | 163    | 124.80  |
|           | 41  | 67     | 141    | 149.75  |
| -----     |     |        |        | -----   |
| HIGH      |     |        |        | 549.10  |
| LOW       | 31  | 61     | 123    | 149.75  |
|           | 51  | 71     | 158    | 124.80  |
|           | 34  | 73     | 154    | 124.80  |
|           | 49  | 64     | 172    | 124.80  |
|           | 60  | 71     | 191    | 149.75  |
| -----     |     |        |        | -----   |
| LOW       |     |        |        | 673.90  |
| MOD       | 43  | 63     | 137    | 149.75  |
|           | 32  | 67     | 151    | 149.75  |
|           | 35  | 70     | 173    | 149.75  |
|           | 47  | 72     | 173    | 124.80  |
|           | 43  | 65     | 123    | 124.80  |
|           | 54  | 71     | 183    | 149.75  |
| -----     |     |        |        | -----   |
| MOD       |     |        |        | 848.60  |
|           |     |        |        | =====   |
|           |     |        |        | 2071.60 |

# Generating Subtotals: Creating a Customized Layout with the ID Statement and the **PAGEBY** statement

You can add a PAGEBY statement to the previous code to produce a subtotals on separate pages.

```
* The data set must be sorted first;  
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;
```

# Generating Subtotals: Creating a Customized Layout with the ID Statement and the PAGEBY statement

The output:

The following tables represent the data shown in the three screenshots, with subtotals generated using the ID statement and PAGEBY statement.

| Act Level | Age | Height | Weight | Fee    |
|-----------|-----|--------|--------|--------|
| HIGH      | 34  | 66     | 152    | 124.80 |
|           | 44  | 66     | 140    | 149.75 |
|           | 40  | 69     | 163    | 124.80 |
|           | 41  | 67     | 141    | 149.75 |
| HIGH      |     |        |        | 549.10 |

  

| Act Level | Age | Height | Weight | Fee    |
|-----------|-----|--------|--------|--------|
| LOW       | 31  | 61     | 123    | 149.75 |
|           | 51  | 71     | 158    | 124.80 |
|           | 34  | 73     | 154    | 124.80 |
|           | 49  | 64     | 172    | 124.80 |
|           | 60  | 71     | 191    | 149.75 |
| LOW       |     |        |        | 673.90 |

  

| Act Level | Age | Height | Weight | Fee     |
|-----------|-----|--------|--------|---------|
| MOD       | 43  | 63     | 137    | 149.75  |
|           | 32  | 67     | 151    | 149.75  |
|           | 35  | 70     | 173    | 149.75  |
|           | 47  | 72     | 173    | 124.80  |
|           | 43  | 65     | 123    | 124.80  |
|           | 54  | 71     | 183    | 149.75  |
| MOD       |     |        |        | 848.60  |
|           |     |        |        | 2071.60 |



# Double Spacing Listing Output

You can specify the **DOUBLE option** in the PROC PRINT statement to double spacing the listing output.

```
proc print data=clinic.stress2 double;  
  var resthr maxhr rechhr;  
  where tolerance='I';  
run;
```

| Obs | Rest<br>HR | Max<br>HR | Rec<br>HR |
|-----|------------|-----------|-----------|
| 2   | 68         | 171       | 133       |
| 3   | 78         | 177       | 139       |
| 8   | 70         | 167       | 122       |
| 11  | 65         | 181       | 141       |
| 14  | 74         | 152       | 113       |
| 15  | 75         | 158       | 108       |
| 20  | 78         | 189       | 138       |

# Specify **Titles** and **Footnotes**

- You can assign up to 10 titles by using **TITLE** statements and specify up to 10 footnotes by using **FOOTNOTE** statements.
- **TITLE** and **FOOTNOTE** statements are global and should be placed anywhere within or before the PRINT procedure. They are assigned as soon as **TITLE** or **FOOTNOTE** statements are run; they apply to all subsequent outputs.

**Syntax:****TITLE**<n> 'text';**FOOTNOTE**<n> 'text';

- n is a number from 1 to 10 that specifies the title or footnote line.
- '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.
- The keyword **title** is equivalent to **title1**, and **footnote** is equivalent to **footnote1**. If you don't specify a title, the default title is The SAS System. No footnote is printed unless you specify one.

# Specify Titles and Footnotes

Example: `options pagesize=20 nodate;`  
`title1 'Heart Rates for Patients with';`  
`title3 'Increased Stress Tolerance Levels';`  
`footnote1 'Data from Treadmill Tests';`  
`footnote3 '1st Quarter Admissions';`  
`proc print data=clinic.stress2;`  
`var resthr maxhr rechr;`  
`where tolerance='I';`  
`run;`

| Heart Rates for Patients with<br>Increased Stress Tolerance Levels |            |           |           |
|--|------------|-----------|-----------|
| Obs  | Rest<br>HR | Max<br>HR | Rec<br>HR |
| 2  | 68         | 171       | 133       |
| 3  | 78         | 177       | 139       |
| 8  | 70         | 167       | 122       |
| 11   | 65         | 181       | 141       |
| 14   | 74         | 152       | 113       |
| 15   | 75         | 158       | 108       |
| 20   | 78         | 189       | 138       |
| Data from Treadmill Tests<br>1st Quarter Admissions                |            |           |           |

# Modify Titles and Footnotes

Redefining a title or footnote line replaces/cancels any same/higher-numbered title or footnote lines, respectively.

```
*Code continuous from the previous slide;  
title3 'Participation in Exercise Therapy';  
proc print data=clinic.therapy;  
  var swim walkjogrun aerclass;  
run;
```

```
*Code continuous from the left example;  
title3 'Participation in Exercise Therapy';  
title2 'Report for March';  
proc print data=clinic.therapy;  
  var swim walkjogrun aerclass;  
run;
```

| Heart Rates for Patients with Participation in Exercise Therapy |      |             |           |
|---|------|-------------|-----------|
| Obs   | Swim | Walk JogRun | Aer Class |
| 1   | 14   | 78          | 56        |
| 2   | 19   | 109         | 32        |
| 3   | 22   | 106         | 35        |
| 4   | 24   | 115         | 47        |
| 5   | 31   | 121         | 55        |
| 6   | 67   | 114         | 61        |
| 7   | 72   | 102         | 67        |
| 8   | 77   | 76          | 64        |
| 9   | 54   | 77          | 78        |
| Data from Treadmill Tests                                       |      |             |           |
| 1st Quarter Admissions  |      |             |           |

| Heart Rates for Patients with Report for March |      |             |           |
|--|------|-------------|-----------|
| Obs  | Swim | Walk JogRun | Aer Class |
| 1  | 14   | 78          | 56        |
| 2  | 19   | 109         | 32        |
| 3  | 22   | 106         | 35        |
| 4  | 24   | 115         | 47        |
| 5  | 31   | 121         | 55        |
| 6  | 67   | 114         | 61        |
| 7  | 72   | 102         | 67        |
| 8  | 77   | 76          | 64        |
| 9  | 54   | 77          | 78        |
| 10   | 47   | 62          | 81        |
| Data from Treadmill Tests                      |      |             |           |
| 1st Quarter Admissions                         |      |             |           |

# Cancel Titles and Footnotes

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

```

title1;
footnote1 'Data from Treadmill
Tests'; footnote3 '1st Quarter
Admissions';
proc print data=clinic.stress2;
  var resthr maxhr rechr;
  where tolerance='I';
run;
footnote;
proc tabulate data=clinic.stress2;
  var timemin timesec;
  table max*(timemin timesec);
run;
  
```

| Obs | Rest<br>HR | Max<br>HR | Rec<br>HR |
|-----|------------|-----------|-----------|
| 2   | 68         | 171       | 133       |
| 3   | 78         | 177       | 139       |
| 8   | 70         | 167       | 122       |
| 11  | 65         | 181       | 141       |
| 14  | 74         | 152       | 113       |
| 15  | 75         | 158       | 108       |
| 20  | 78         | 189       | 138       |

Data from Treadmill Tests  
1st Quarter Admissions

| Max     |         |
|---------|---------|
| TimeMin | TimeSec |
| 17.00   | 57.00   |



# Assign Descriptive Labels

You can use the **LABEL statement** and a **LABEL option** in the PROC PRINT statement to label columns with more descriptive text (up to 256 characters). It only applies to the current step.

```
proc print data=clinic.admit label;  
  var age height;  
  label age='Age of Patient';  
  label height='Height in Inches';  
run;
```

Or

```
proc print data=clinic.admit label;  
  var age height;  
  label age='Age of Patient'  
        height='Height in Inches';  
run;
```

| Obs | Age of<br>Patient | Height<br>in<br>Inches |
|-----|-------------------|------------------------|
| 14  | 40                | 69                     |
| 15  | 47                | 72                     |
| 16  | 60                | 71                     |
| 17  | 43                | 65                     |
| 18  | 25                | 75                     |
| 19  | 22                | 63                     |
| 20  | 41                | 67                     |
| 21  | 54                | 71                     |

# Format Data Values

To make data values more understandable when they are displayed in your output, you can use the **FORMAT statement**, which associates formats with variables. Formats affect only how the data values appear in output, not the actual data values as they are stored in the SAS data set.

| Format     | Specifies values ...   | Example   |
|------------|--|-----------|
| COMMAw.d   | that contain commas and decimal places                                   | comma8.2  |
| DOLLAR w.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.    |

Note: w is the total field width; it counts all the positions.

# Format Data Values

You can use a separate FORMAT statement for each variable, or you can format several variables (using either the same format or different formats) in a single FORMAT statement.

**Syntax:** `FORMAT variable(s) format-name;`

| This FORMAT statement ...                            | Associates ...  | To display values as ... |
|--|---|--------------------------|
| <code>format date mmddyy8.;</code>                   | the format MMDDYY8. with the variable Date  | 06/05/03                 |
| <code>format net comma5.0<br/>gross comma8.2;</code> | the format COMMA5.0 with the variable Net and the format COMMA8.2 with the variable Gross | 1,234<br>5,678.90        |
| <code>format net gross<br/>dollar9.2;</code>         | the format DOLLAR9.2 with both variables, Net and Gross                                   | \$1,234.00<br>\$5,678.90 |



# Format Data Values

```
proc print data=clinic.admit;  
  var actlevel fee;  
  where actlevel='HIGH';  
  format fee dollar4.;  
run;
```

| Obs | Act<br>Level | Fee   |
|-----|--------------|-------|
| 1   | HIGH         | \$85  |
| 2   | HIGH         | \$125 |
| 6   | HIGH         | \$125 |
| 11  | HIGH         | \$150 |
| 14  | HIGH         | \$125 |
| 18  | HIGH         | \$85  |
| 20  | HIGH         | \$150 |

# Examples of SAS Formats, Stored Values and Displayed Values

| Stored Value | Format     | Displayed Value |
|--------------|------------|-----------------|
| 38245.3975   | COMMA12.2  | 38,245.40       |
| 38245.3975   | 12.2       | 38245.40        |
| 38245.3975   | DOLLAR12.2 | \$38,245.40     |
| 38245.3975   | DOLLAR9.2  | \$38245.40      |
| 38245.3975   | DOLLAR8.2  | 38245.40        |
| 0            | MMDDYY8.   | 01/01/60        |
| 0            | MMDDYY10.  | 01/01/1960      |
| 0            | DATE7.     | 01JAN60         |
| 0            | DATE9.     | 01JAN1960       |

# Permanently Assigned Labels and Formats

Permanent labels and formats can be assigned in the DATA step. These labels and formats are saved with the data set, and they can later be used by procedures that reference the data set.

```
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 label;  
  var date dest boarded;  
run;
```

| Obs | Departure<br>Date | Dest | Boarded |
|-----|-------------------|------|---------|
| 1   | 04MAR1999         | PAR  | 146     |
| 2   | 07MAR1999         | PAR  | 155     |
| 3   | 04MAR1999         | PAR  | 146     |
| 4   | 07MAR1999         | PAR  | 155     |