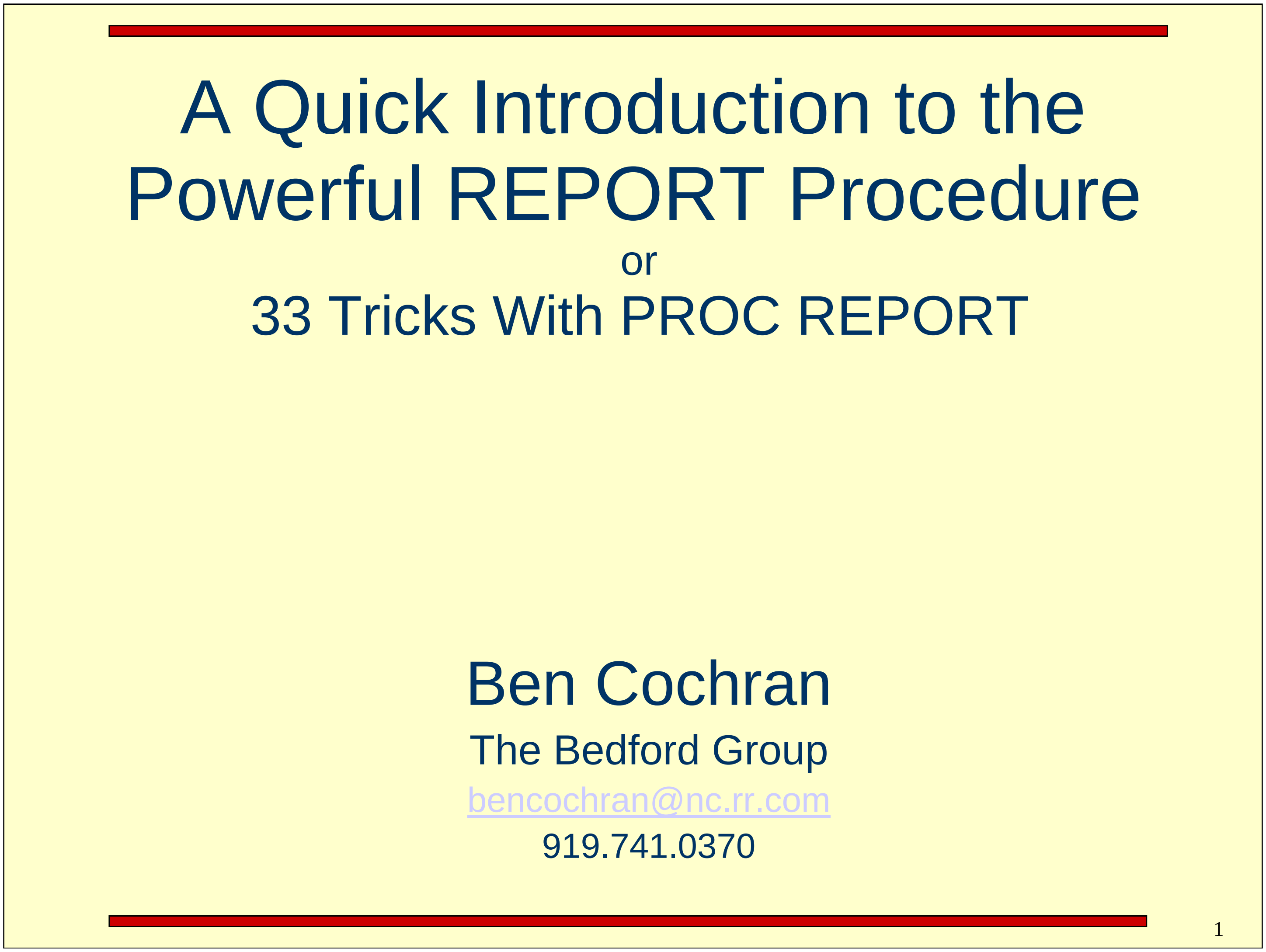


A Quick Introduction to the Powerful REPORT Procedure

or

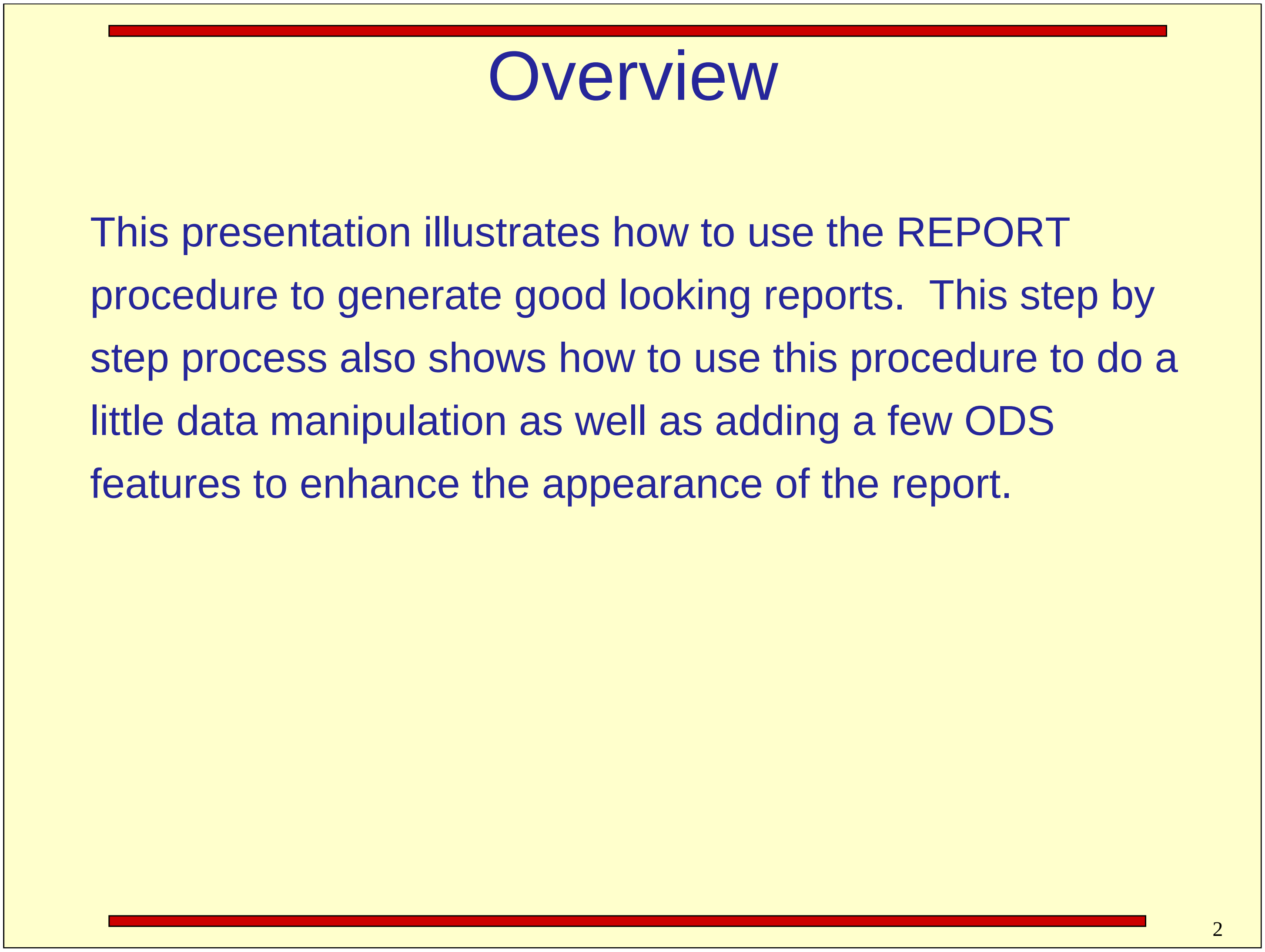
33 Tricks With PROC REPORT

Ben Cochran

The Bedford Group

bencochran@nc.rr.com

919.741.0370



Overview

This presentation illustrates how to use the REPORT procedure to generate good looking reports. This step by step process also shows how to use this procedure to do a little data manipulation as well as adding a few ODS features to enhance the appearance of the report.

Acknowledgements

Tech Support at SAS Institute

- Bari Lawhorn
- Alison McMahill Booth
- Chevelle Parker
- The whole Division

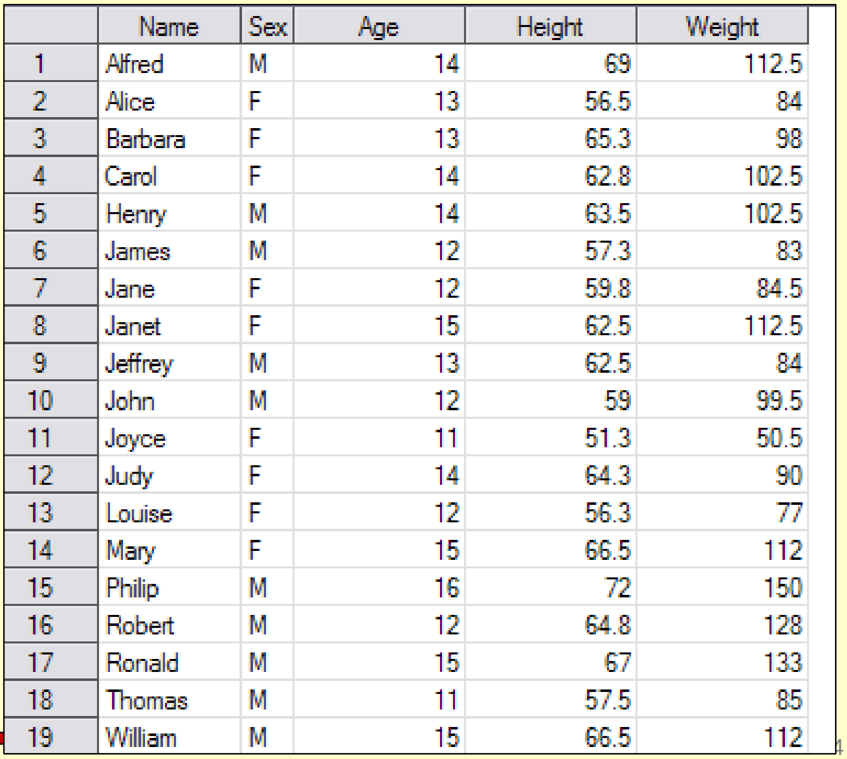
PROC REPORT

The REPORT procedure is widely used in generating reports which include:

- data listing,
- summary statistics,
- and ‘tabular’ reports.

The REPORT procedure has powerful report writing capabilities not found in other SAS procedures.

The SASHELP.CLASS → data set is used in the following examples:



	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5
2	Alice	F	13	56.5	84
3	Barbara	F	13	65.3	98
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84
10	John	M	12	59	99.5
11	Joyce	F	11	51.3	50.5
12	Judy	F	14	64.3	90
13	Louise	F	12	56.3	77
14	Mary	F	15	66.5	112
15	Philip	M	16	72	150
16	Robert	M	12	64.8	128
17	Ronald	M	15	67	133
18	Thomas	M	11	57.5	85
19	William	M	15	66.5	112

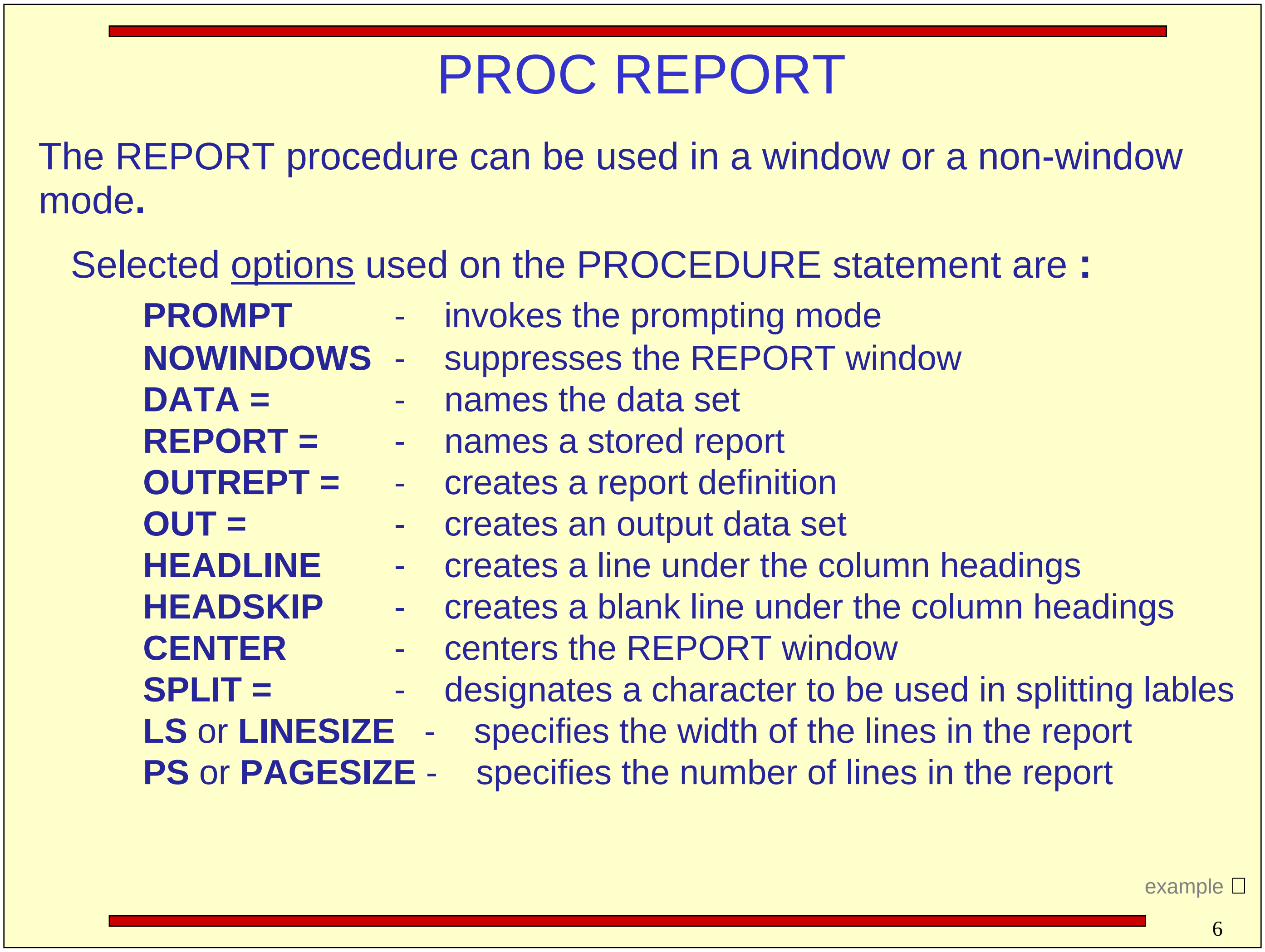
In the SASUSER.PM data set, each row represents the INCOME and OVERHEAD per YEAR, per TYPE, per COUNTRY, per HUB.

PROC REPORT

The typical form of the REPORT procedure**:**

```
PROC REPORT  data= SAS-data-set  options ;
          COLUMNS variable_1 …. variable_n;
          DEFINE variable_1;
          DEFINE variable_2;
          . . .
          DEFINE variable_n;

          COMPUTE  blocks
          BREAK … ;
          RBREAK … ;
RUN;
```

- **COLUMNS** statement defines the columns and their order,
- **DEFINE** statements declare how each variable is to be used,
- **COMPUTE** blocks allow calculations in the report,
- **BREAK/RBREAK** allow physical breaks (ie. blank lines) in the report.



PROC REPORT

The REPORT procedure can be used in a window or a non-window mode**.**

Selected options used on the PROCEDURE statement are **:**

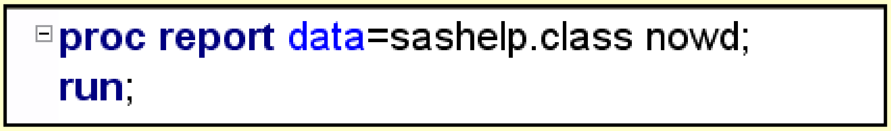
PROMPT	-	invokes the prompting mode
NOWINDOWS	-	suppresses the REPORT window
DATA =	-	names the data set
REPORT =	-	names a stored report
OUTREPT =	-	creates a report definition
OUT =	-	creates an output data set
HEADLINE	-	creates a line under the column headings
HEADSKIP	-	creates a blank line under the column headings
CENTER	-	centers the REPORT window
SPLIT =	-	designates a character to be used in splitting lables
LS or **LINESIZE**	-	specifies the width of the lines in the report
PS or **PAGESIZE**	-	specifies the number of lines in the report

example □

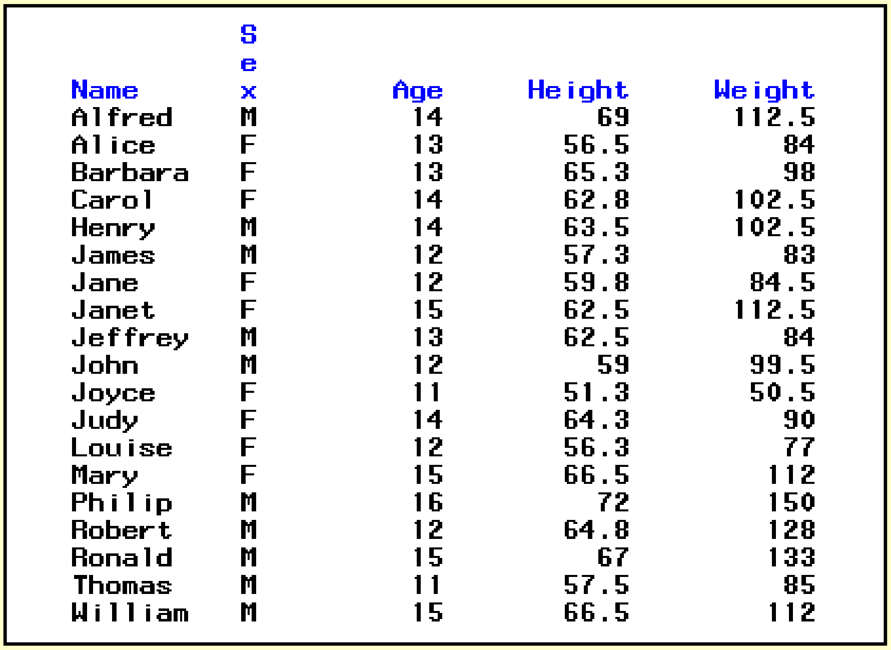
There are many STATEMENTS and OPTIONS that allow you to have much flexibility in creating and Customizing your reports.

PROC REPORT

Trick 1: Generate a **default** report using the REPORT procedure**:**



```
proc report data=sashelp.class nowd;
run;
```



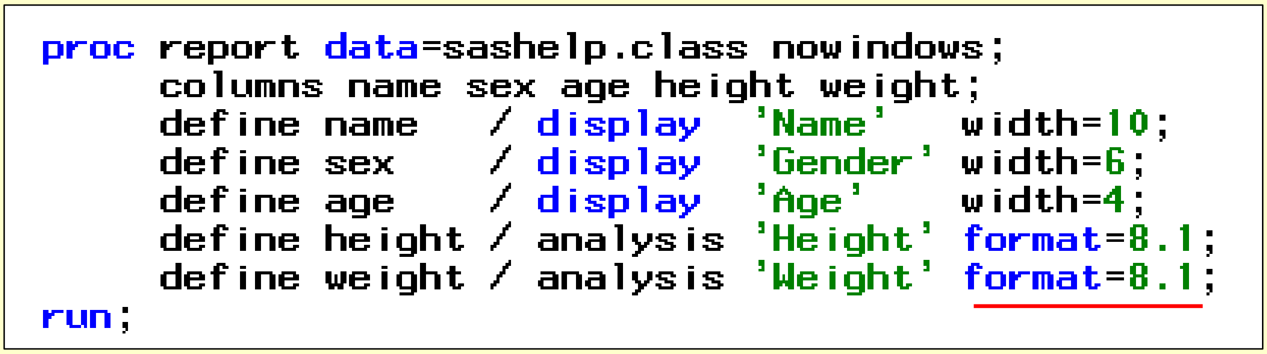
```
                S
                e
Name            x          Age     Height     Weight
Alfred          M           14         69      112.5
Alice           F           13       56.5         84
Barbara         F           13       65.3         98
Carol           F           14       62.8      102.5
Henry           M           14       63.5      102.5
James           M           12       57.3         83
Jane            F           12       59.8       84.5
Janet           F           15       62.5      112.5
Jeffrey         M           13       62.5         84
John            M           12         59       99.5
Joyce           F           11       51.3       50.5
Judy            F           14       64.3         90
Louise          F           12       56.3         77
Mary            F           15       66.5        112
Philip          M           16         72        150
Robert          M           12       64.8        128
Ronald          M           15         67        133
Thomas          M           11       57.5         85
William         M           15       66.5        112
```

Notice the defaults…

output □

PROC REPORT

Trick 2: Generate a **basic** report using the REPORT procedure**:**



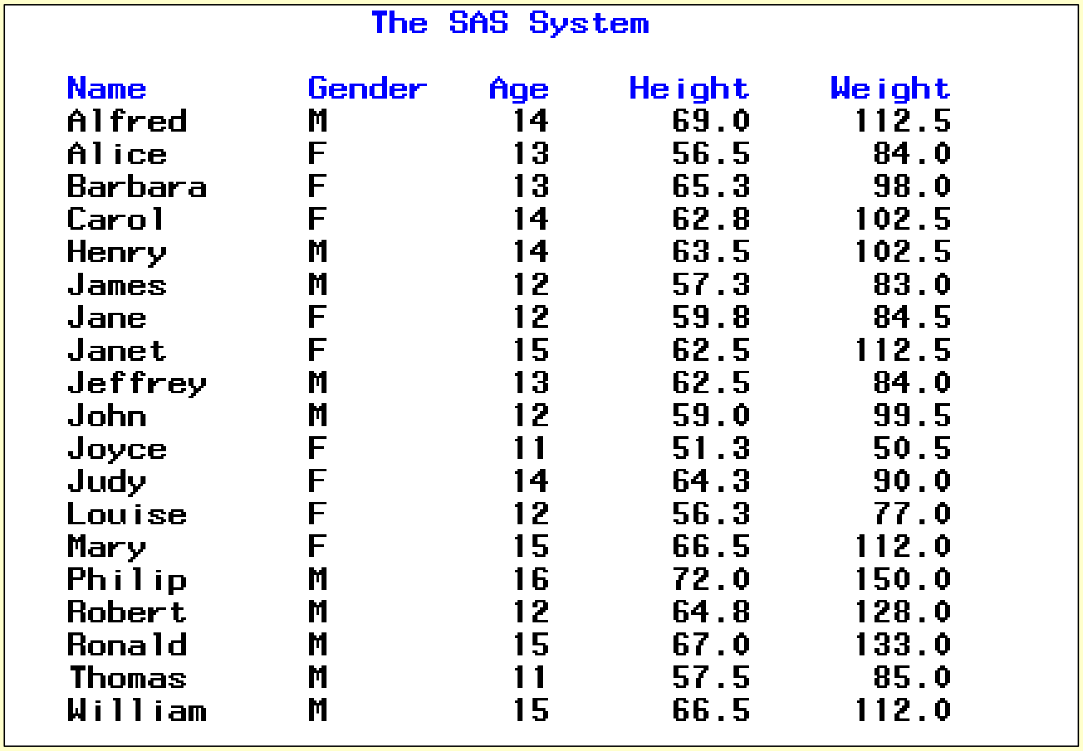
```
proc report data=sashelp.class nowindows;
     columns name sex age height weight;
     define name   / display  'Name'   width=10;
     define sex    / display  'Gender' width=6;
     define age    / display  'Age'    width=4;
     define height / analysis 'Height' format=8.1;
     define weight / analysis 'Weight' format=8.1;
run;
```

Columns can be defined as**:**

- **GROUP** observations into categories,
- **DISPLAY** values for each observation,
- **ANALYSIS** contribute values to a statistic,
- **ORDER** defines the order of the report rows,
- **ACROSS** creates columns for each of its values,
- **COMPUTED** values are created in a compute block.

output □

PROC REPORT



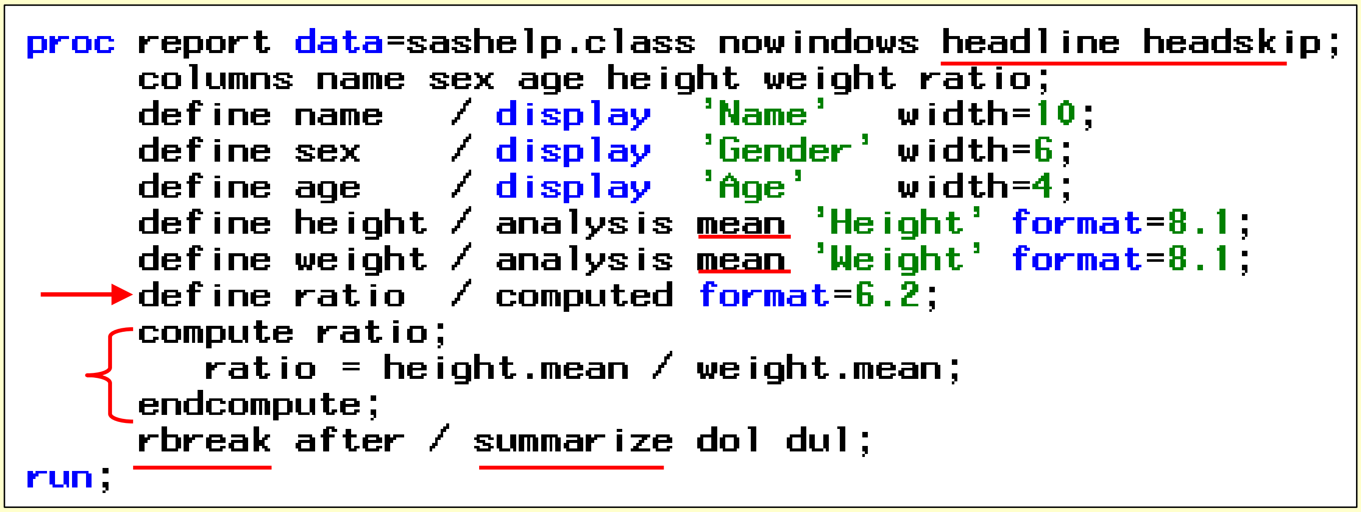
```
                      The SAS System

Name           Gender     Age     Height     Weight
Alfred         M           14       69.0      112.5
Alice          F           13       56.5       84.0
Barbara        F           13       65.3       98.0
Carol          F           14       62.8      102.5
Henry          M           14       63.5      102.5
James          M           12       57.3       83.0
Jane           F           12       59.8       84.5
Janet          F           15       62.5      112.5
Jeffrey        M           13       62.5       84.0
John           M           12       59.0       99.5
Joyce          F           11       51.3       50.5
Judy           F           14       64.3       90.0
Louise         F           12       56.3       77.0
Mary           F           15       66.5      112.0
Philip         M           16       72.0      150.0
Robert         M           12       64.8      128.0
Ronald         M           15       67.0      133.0
Thomas         M           11       57.5       85.0
William        M           15       66.5      112.0
```

Any enhancements?

more □

PROC REPORT

Trick 3: Enhance the report by adding a blank line after the column names and **calculating** values for a new column… **RATIO.**



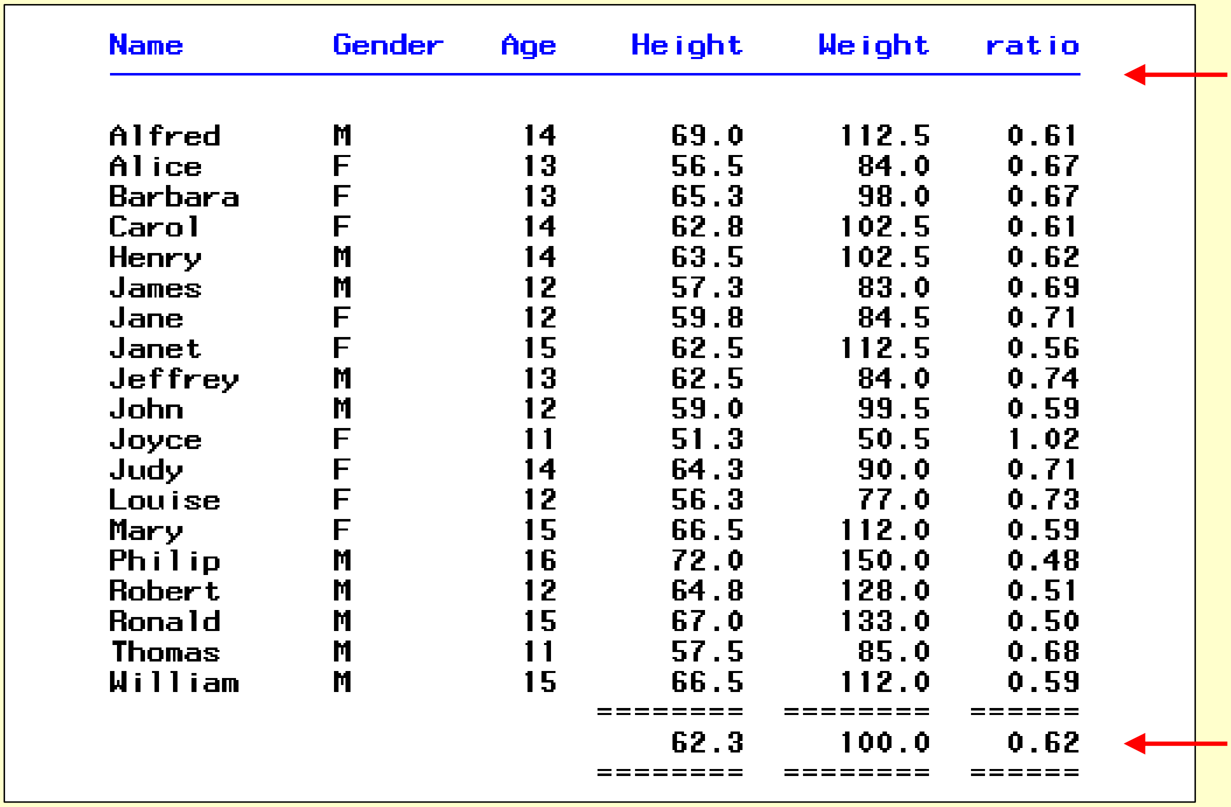
```
proc report data=sashelp.class nowindows headline headskip;
     columns name sex age height weight ratio;
     define name   / display  'Name'   width=10;
     define sex    / display  'Gender' width=6;
     define age    / display  'Age'    width=4;
     define height / analysis mean 'Height' format=8.1;
     define weight / analysis mean 'Weight' format=8.1;
     define ratio  / computed format=6.2;
     compute ratio;
        ratio = height.mean / weight.mean;
     endcompute;
     rbreak after / summarize dol dul;
run;
```

Notice the following:

- **HEADLINE and HEADSKIP** options,
- the **COMPUTE** block,
- the **RBREAK** statement

output □

PROC REPORT



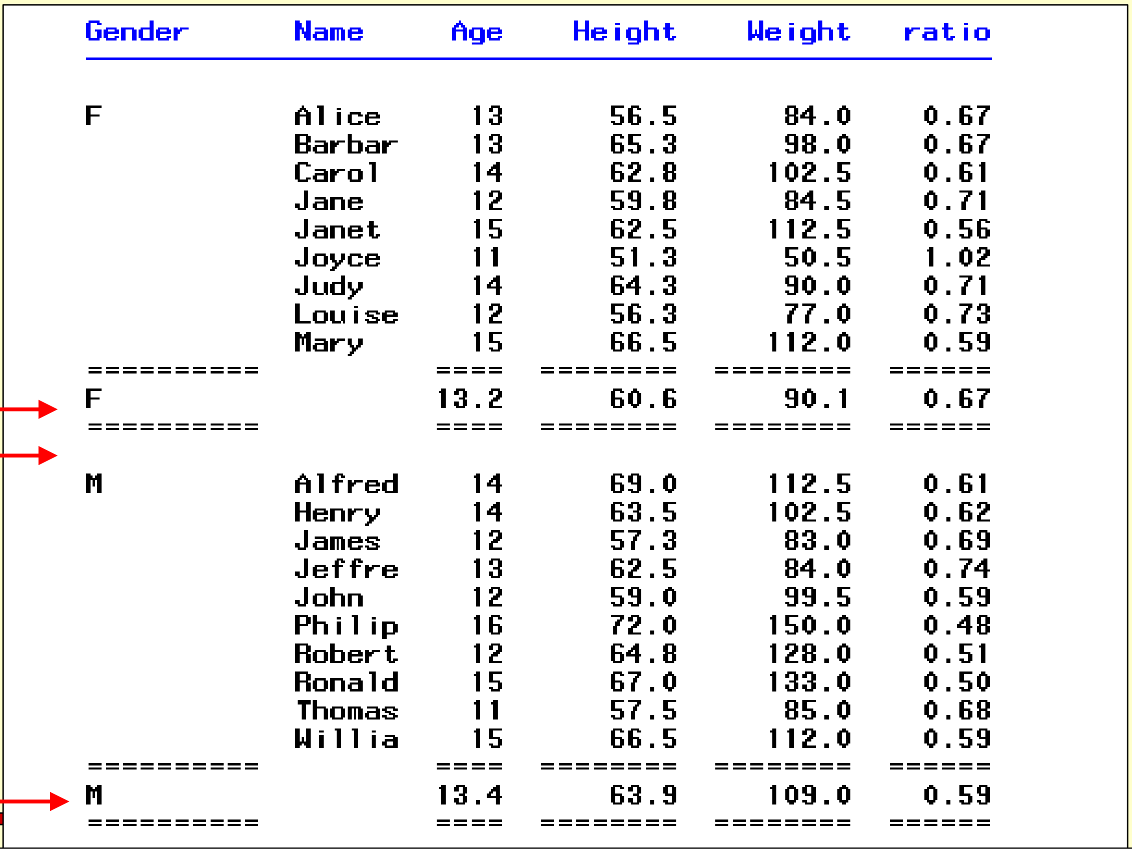
```
Name         Gender    Age    Height     Weight   ratio
-------------------------------------------------------

Alfred       M          14      69.0      112.5    0.61
Alice        F          13      56.5       84.0    0.67
Barbara      F          13      65.3       98.0    0.67
Carol        F          14      62.8      102.5    0.61
Henry        M          14      63.5      102.5    0.62
James        M          12      57.3       83.0    0.69
Jane         F          12      59.8       84.5    0.71
Janet        F          15      62.5      112.5    0.56
Jeffrey      M          13      62.5       84.0    0.74
John         M          12      59.0       99.5    0.59
Joyce        F          11      51.3       50.5    1.02
Judy         F          14      64.3       90.0    0.71
Louise       F          12      56.3       77.0    0.73
Mary         F          15      66.5      112.0    0.59
Philip       M          16      72.0      150.0    0.48
Robert       M          12      64.8      128.0    0.51
Ronald       M          15      67.0      133.0    0.50
Thomas       M          11      57.5       85.0    0.68
William      M          15      66.5      112.0    0.59
                          ========   ========  ======
                              62.3      100.0    0.62
                          ========   ========  ======
```

… more enhancements?

more □

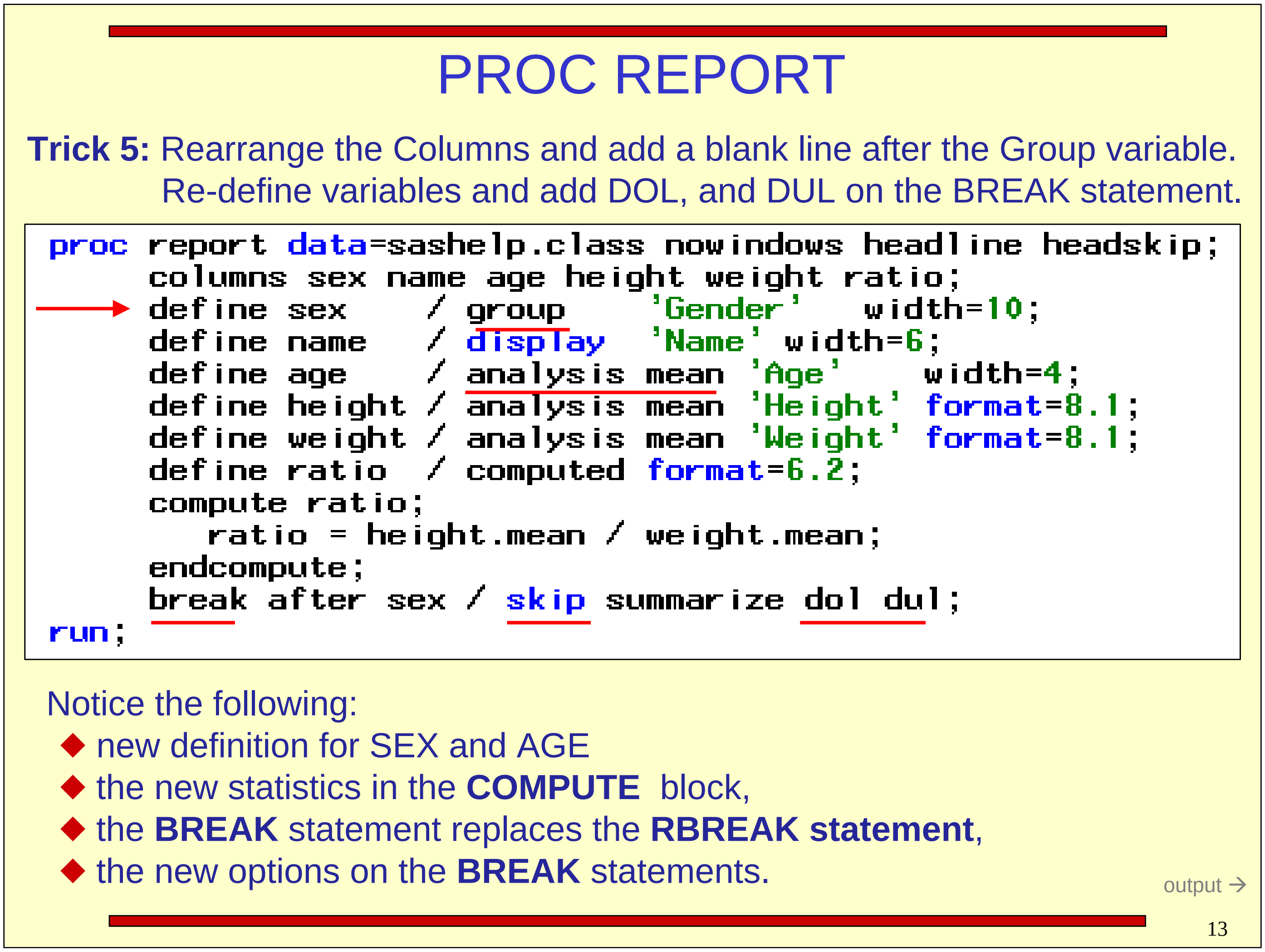
PROC REPORT

Trick 4: Find the Mean AGE, HEIGHT, WEIGHT, & RATIO for each gender.



```
Gender      Name      Age   Height    Weight   ratio
----------------------------------------------------

F           Alice      13     56.5      84.0    0.67
            Barbar     13     65.3      98.0    0.67
            Carol      14     62.8     102.5    0.61
            Jane       12     59.8      84.5    0.71
            Janet      15     62.5     112.5    0.56
            Joyce      11     51.3      50.5    1.02
            Judy       14     64.3      90.0    0.71
            Louise     12     56.3      77.0    0.73
            Mary       15     66.5     112.0    0.59
==========           ==== ======== ======== ======
F                    13.2     60.6      90.1    0.67
==========           ==== ======== ======== ======

M           Alfred     14     69.0     112.5    0.61
            Henry      14     63.5     102.5    0.62
            James      12     57.3      83.0    0.69
            Jeffre     13     62.5      84.0    0.74
            John       12     59.0      99.5    0.59
            Philip     16     72.0     150.0    0.48
            Robert     12     64.8     128.0    0.51
            Ronald     15     67.0     133.0    0.50
            Thomas     11     57.5      85.0    0.68
            Willia     15     66.5     112.0    0.59
==========           ==== ======== ======== ======
M                    13.4     63.9     109.0    0.59
==========           ==== ======== ======== ======
```

more



PROC REPORT

Trick 5: Rearrange the Columns and add a blank line after the Group variable. Re-define variables and add DOL, and DUL on the BREAK statement.

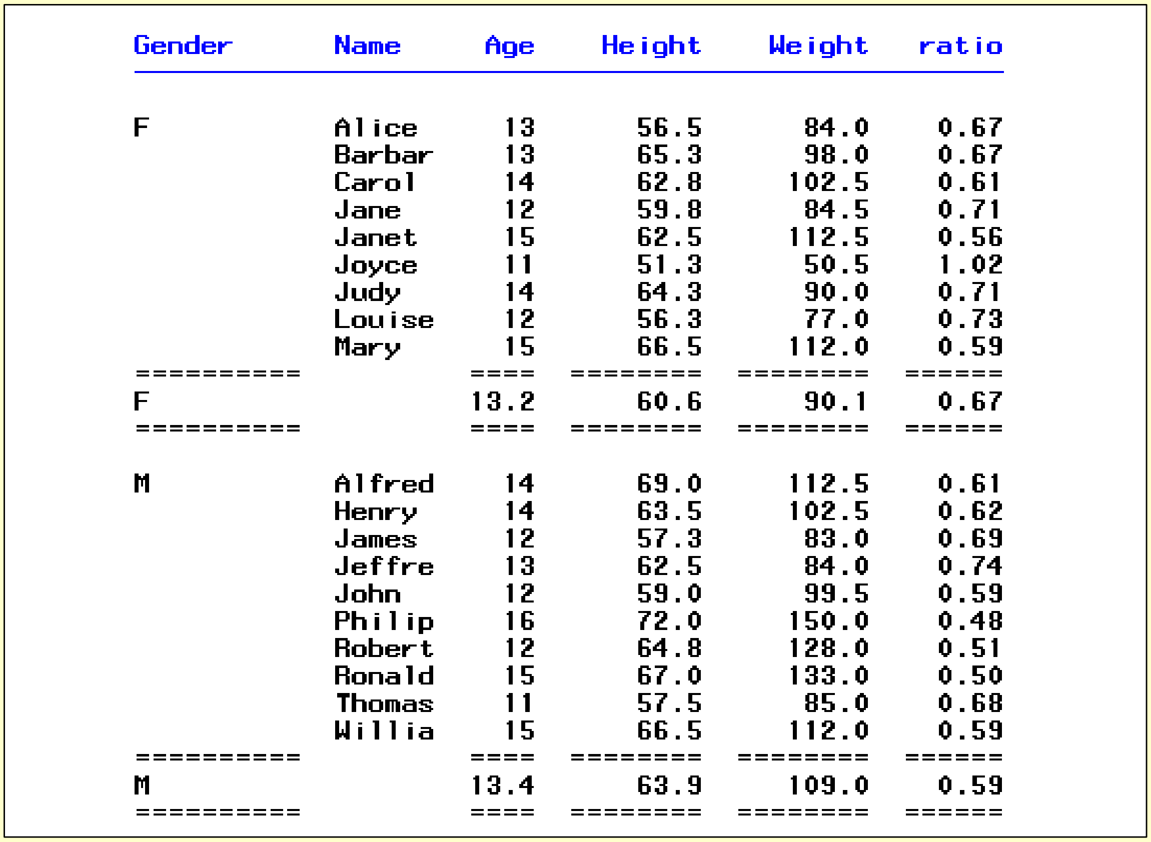
```
proc report data=sashelp.class nowindows headline headskip;
     columns sex name age height weight ratio;
     define sex    / group     'Gender'   width=10;
     define name   / display  'Name' width=6;
     define age    / analysis mean 'Age'    width=4;
     define height / analysis mean 'Height' format=8.1;
     define weight / analysis mean 'Weight' format=8.1;
     define ratio  / computed format=6.2;
     compute ratio;
        ratio = height.mean / weight.mean;
     endcompute;
     break after sex / skip summarize dol dul;
run;
```

Notice the following:

- new definition for SEX and AGE
- the new statistics in the **COMPUTE** block,
- the **BREAK** statement replaces the **RBREAK statement**,
- the new options on the **BREAK** statements.

output →

PROC REPORT

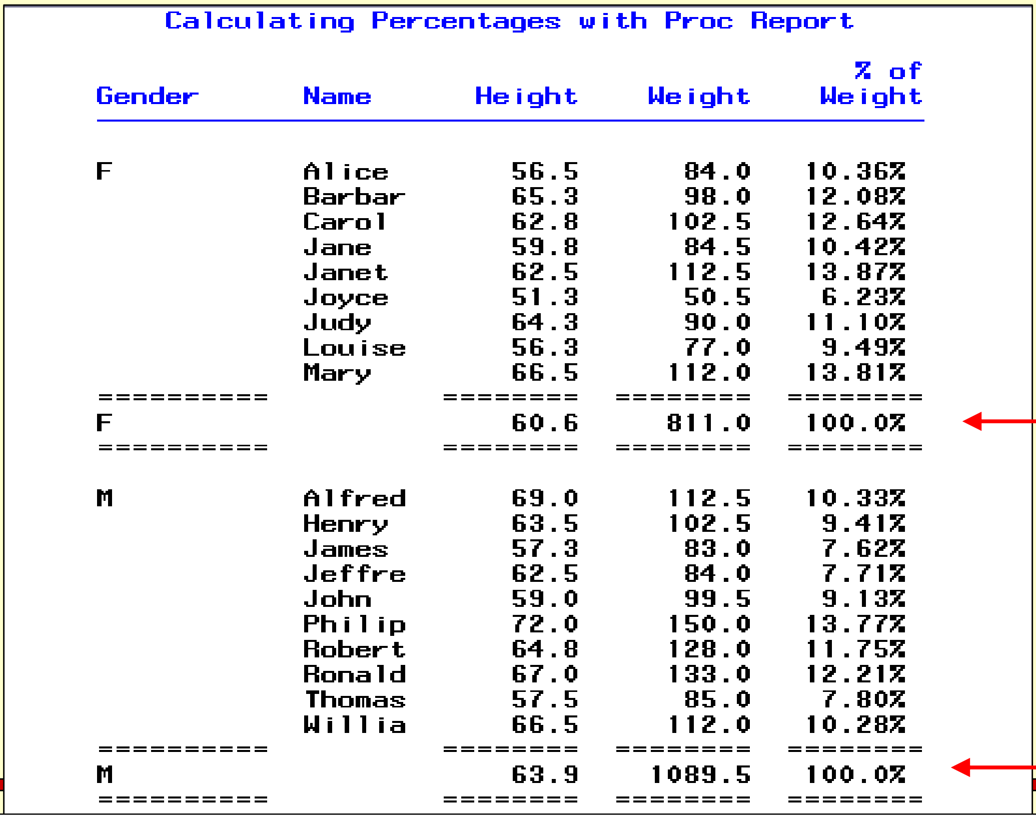


```
Gender          Name       Age     Height     Weight    ratio
_____________________________________________________________


F               Alice       13       56.5       84.0     0.67
                Barbar      13       65.3       98.0     0.67
                Carol       14       62.8      102.5     0.61
                Jane        12       59.8       84.5     0.71
                Janet       15       62.5      112.5     0.56
                Joyce       11       51.3       50.5     1.02
                Judy        14       64.3       90.0     0.71
                Louise      12       56.3       77.0     0.73
                Mary        15       66.5      112.0     0.59
==========                ====   ========   ========   ======
F                         13.2       60.6       90.1     0.67
==========                ====   ========   ========   ======

M               Alfred      14       69.0      112.5     0.61
                Henry       14       63.5      102.5     0.62
                James       12       57.3       83.0     0.69
                Jeffre      13       62.5       84.0     0.74
                John        12       59.0       99.5     0.59
                Philip      16       72.0      150.0     0.48
                Robert      12       64.8      128.0     0.51
                Ronald      15       67.0      133.0     0.50
                Thomas      11       57.5       85.0     0.68
                Willia      15       66.5      112.0     0.59
==========                ====   ========   ========   ======
M                         13.4       63.9      109.0     0.59
==========                ====   ========   ========   ======
```

Here we see the AVERAGE   AGE, HEIGHT, WEIGHT and RATIO

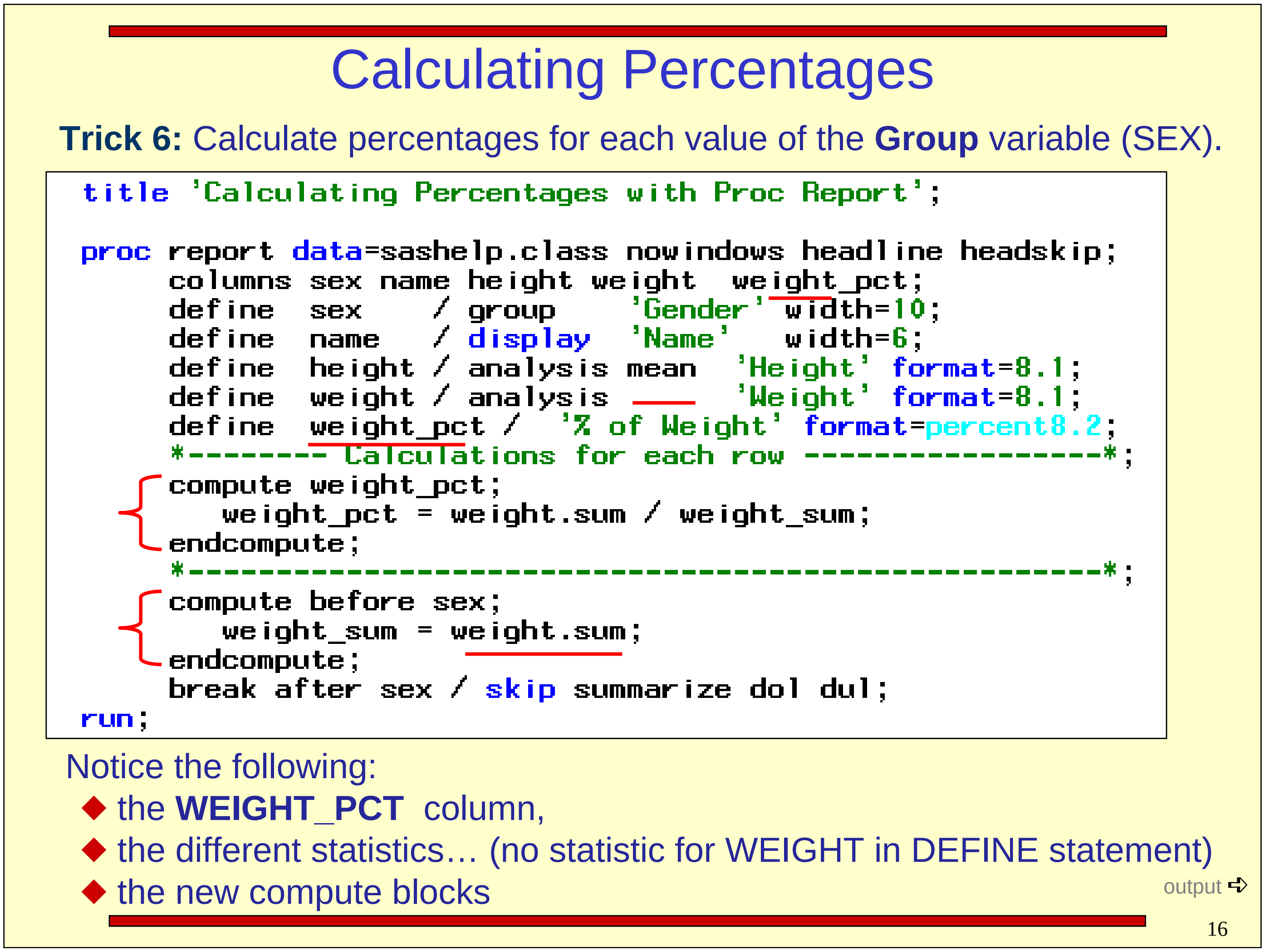
Calculating Percentages

Trick 6: Enhance the report by calculating percentages so that they add up to 100 for each value of the **Group** variable (SEX).



```
              Calculating Percentages with Proc Report

                                                             % of
    Gender          Name           Height      Weight      Weight
    ---------------------------------------------------------------

    F               Alice            56.5        84.0      10.36%
                    Barbar           65.3        98.0      12.08%
                    Carol            62.8       102.5      12.64%
                    Jane             59.8        84.5      10.42%
                    Janet            62.5       112.5      13.87%
                    Joyce            51.3        50.5       6.23%
                    Judy             64.3        90.0      11.10%
                    Louise           56.3        77.0       9.49%
                    Mary             66.5       112.0      13.81%
    ==========                   ========    ========    ========
    F                                60.6       811.0      100.0%
    ==========                   ========    ========    ========

    M               Alfred           69.0       112.5      10.33%
                    Henry            63.5       102.5       9.41%
                    James            57.3        83.0       7.62%
                    Jeffre           62.5        84.0       7.71%
                    John             59.0        99.5       9.13%
                    Philip           72.0       150.0      13.77%
                    Robert           64.8       128.0      11.75%
                    Ronald           67.0       133.0      12.21%
                    Thomas           57.5        85.0       7.80%
                    Willia           66.5       112.0      10.28%
    ==========                   ========    ========    ========
    M                                63.9      1089.5      100.0%
    ==========                   ========    ========    ========
```

pgm ➪



Calculating Percentages

Trick 6: Calculate percentages for each value of the **Group** variable (SEX).

```
 title 'Calculating Percentages with Proc Report';

proc report data=sashelp.class nowindows headline headskip;
     columns sex name height weight  weight_pct;
     define  sex    / group     'Gender' width=10;
     define  name   / display   'Name'   width=6;
     define  height / analysis mean   'Height' format=8.1;
     define  weight / analysis        'Weight' format=8.1;
     define  weight_pct /  '% of Weight' format=percent8.2;
     *-------- Calculations for each row -----------------*;
     compute weight_pct;
        weight_pct = weight.sum / weight_sum;
     endcompute;
     *---------------------------------------------------*;
     compute before sex;
        weight_sum = weight.sum;
     endcompute;
     break after sex / skip summarize dol dul;
run;
```

Notice the following:

- the **WEIGHT_PCT** column,
- the different statistics… (no statistic for WEIGHT in DEFINE statement)
- the new compute blocks

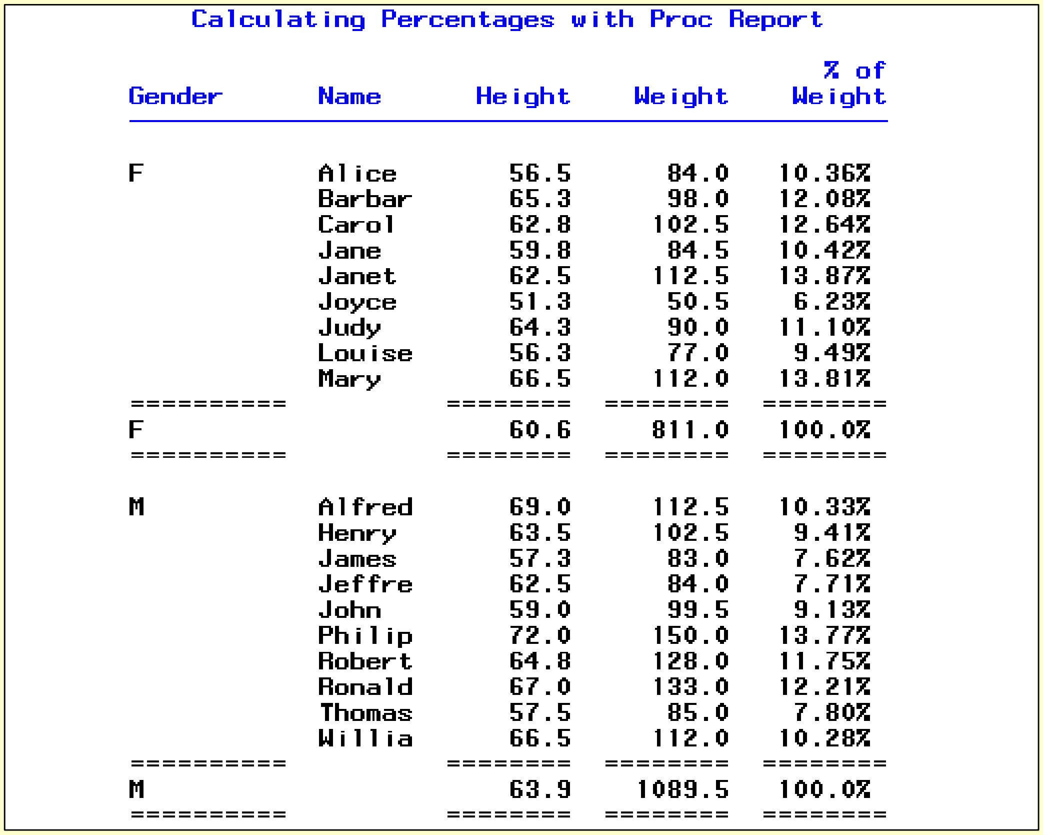
output ➪

16

The MEAN statistic is removed from the DEFINE statement for WEIGHT, so that the SUM could be Calculated in the new compute block.

The SUMMARIZE option on the BREAK statement causes the SUM to be calculated at the BREAK.

Calculating Percentages

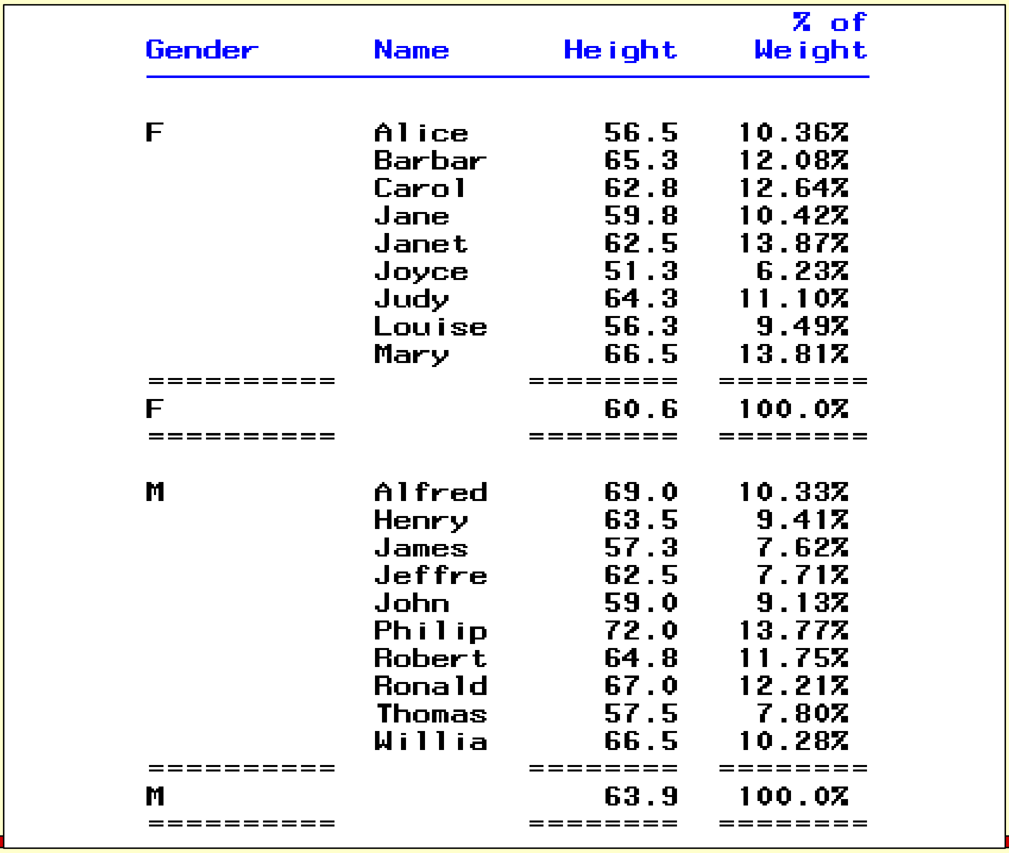


```
              Calculating Percentages with Proc Report

                                                            % of
Gender       Name        Height    Weight     Weight
--------------------------------------------------


F            Alice         56.5      84.0    10.36%
             Barbar        65.3      98.0    12.08%
             Carol         62.8     102.5    12.64%
             Jane          59.8      84.5    10.42%
             Janet         62.5     112.5    13.87%
             Joyce         51.3      50.5     6.23%
             Judy          64.3      90.0    11.10%
             Louise        56.3      77.0     9.49%
             Mary          66.5     112.0    13.81%
==========             ========  ========  ========
F                          60.6     811.0    100.0%
==========             ========  ========  ========

M            Alfred        69.0     112.5    10.33%
             Henry         63.5     102.5     9.41%
             James         57.3      83.0     7.62%
             Jeffre        62.5      84.0     7.71%
             John          59.0      99.5     9.13%
             Philip        72.0     150.0    13.77%
             Robert        64.8     128.0    11.75%
             Ronald        67.0     133.0    12.21%
             Thomas        57.5      85.0     7.80%
             Willia        66.5     112.0    10.28%
==========             ========  ========  ========
M                          63.9    1089.5    100.0%
==========             ========  ========  ========
```

Notice the **Weight** column. Does it ‘make sense’ to SUM weight?

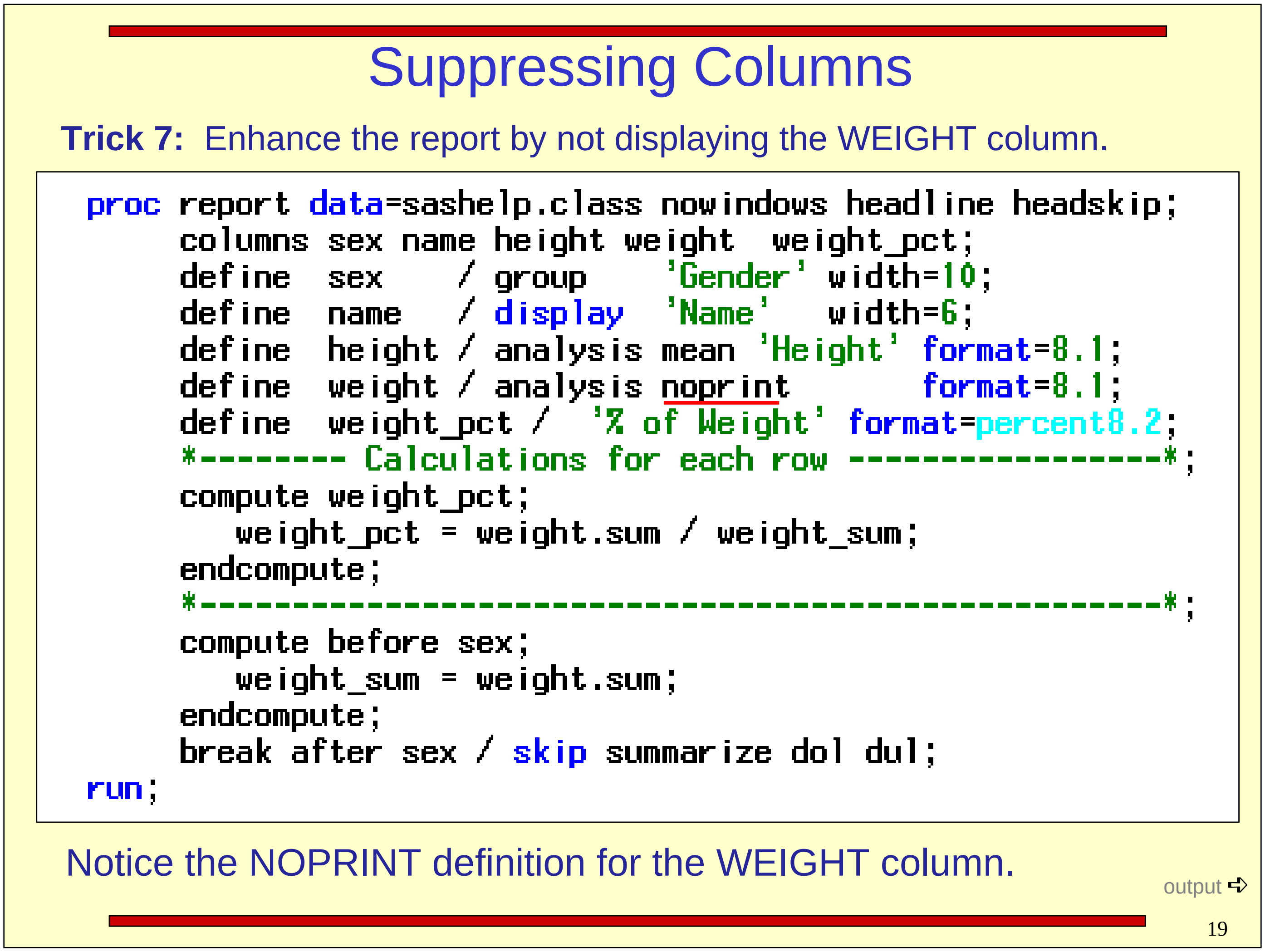
Suppressing Columns

Trick 7: Enhance the report by **not** displaying the WEIGHT column.



```
                                            % of
Gender       Name          Height         Weight
------------------------------------------------


F            Alice           56.5         10.36%
             Barbar          65.3         12.08%
             Carol           62.8         12.64%
             Jane            59.8         10.42%
             Janet           62.5         13.87%
             Joyce           51.3          6.23%
             Judy            64.3         11.10%
             Louise          56.3          9.49%
             Mary            66.5         13.81%
==========                 ========     ========
F                            60.6         100.0%
==========                 ========     ========

M            Alfred          69.0         10.33%
             Henry           63.5          9.41%
             James           57.3          7.62%
             Jeffre          62.5          7.71%
             John            59.0          9.13%
             Philip          72.0         13.77%
             Robert          64.8         11.75%
             Ronald          67.0         12.21%
             Thomas          57.5          7.80%
             Willia          66.5         10.28%
==========                 ========     ========
M                            63.9         100.0%
==========                 ========     ========
```

pgm ➪



Suppressing Columns

Trick 7: Enhance the report by not displaying the WEIGHT column.

```
proc report data=sashelp.class nowindows headline headskip;
     columns sex name height weight  weight_pct;
     define  sex    / group     'Gender' width=10;
     define  name   / display  'Name'   width=6;
     define  height / analysis mean 'Height' format=8.1;
     define  weight / analysis noprint       format=8.1;
     define  weight_pct /  '% of Weight' format=percent8.2;
     *-------- Calculations for each row ----------------*;
     compute weight_pct;
        weight_pct = weight.sum / weight_sum;
     endcompute;
     *--------------------------------------------------*;
     compute before sex;
        weight_sum = weight.sum;
     endcompute;
     break after sex / skip summarize dol dul;
run;
```

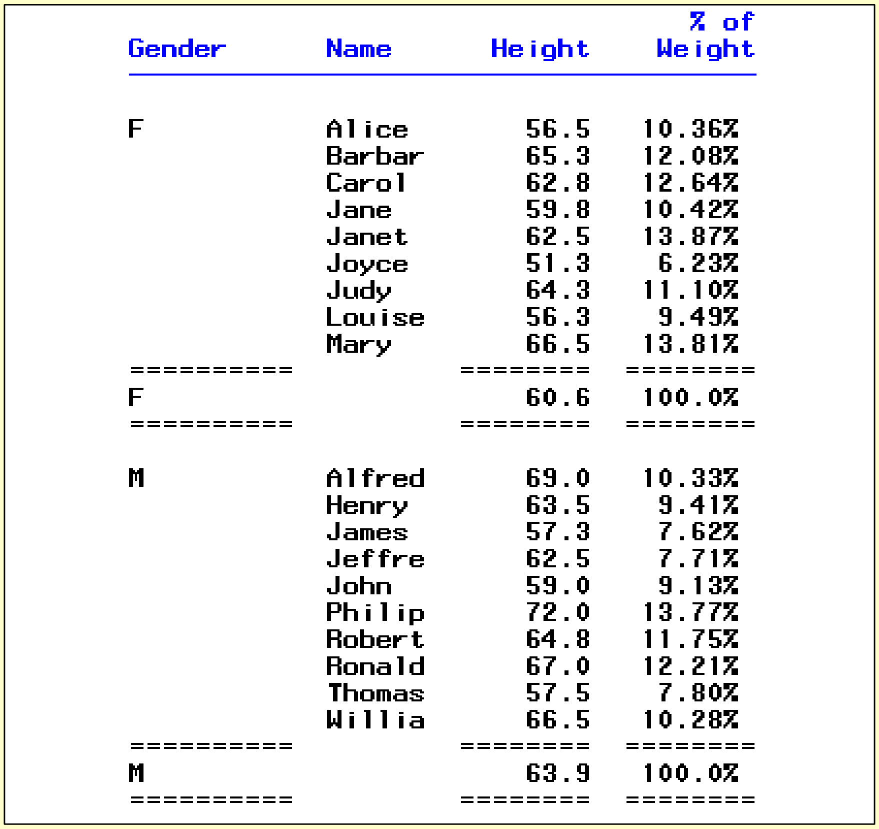
Notice the NOPRINT definition for the WEIGHT column.

output ➪

19

The NOPRINT option is used in the DEFINE statement to suppress the printing of the WEIGHT column. However, the WEIGHT variable still needs to be in the procedure to calculate the proper statistics in the COMPUTE blocks.

Suppressing Columns

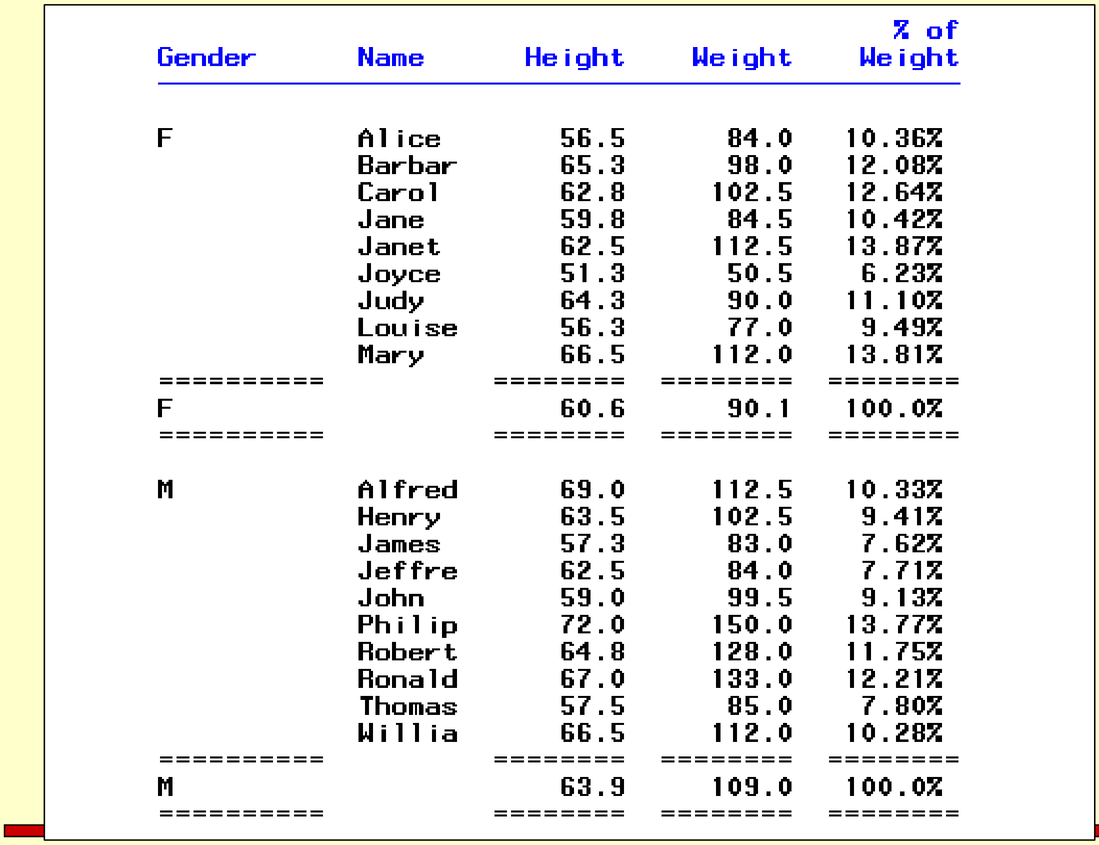


```
                                  % of
Gender      Name          Height  Weight


F           Alice           56.5   10.36%
            Barbar          65.3   12.08%
            Carol           62.8   12.64%
            Jane            59.8   10.42%
            Janet           62.5   13.87%
            Joyce           51.3    6.23%
            Judy            64.3   11.10%
            Louise          56.3    9.49%
            Mary            66.5   13.81%
==========          ========  ========
F                           60.6   100.0%
==========          ========  ========

M           Alfred          69.0   10.33%
            Henry           63.5    9.41%
            James           57.3    7.62%
            Jeffre          62.5    7.71%
            John            59.0    9.13%
            Philip          72.0   13.77%
            Robert          64.8   11.75%
            Ronald          67.0   12.21%
            Thomas          57.5    7.80%
            Willia          66.5   10.28%
==========          ========  ========
M                           63.9   100.0%
==========          ========  ========
```

Notice the absence of the WEIGHT column.

Calculating Percentages

Trick 8: Add **WEIGHT** to the report, calculate its’ **AVERAGE** for each group.



```
                                                     % of
Gender         Name        Height     Weight       Weight
-------------------------------------------------------------


F              Alice         56.5       84.0     10.36%
               Barbar        65.3       98.0     12.08%
               Carol         62.8      102.5     12.64%
               Jane          59.8       84.5     10.42%
               Janet         62.5      112.5     13.87%
               Joyce         51.3       50.5      6.23%
               Judy          64.3       90.0     11.10%
               Louise        56.3       77.0      9.49%
               Mary          66.5      112.0     13.81%
==========               ========   ========   ========
F                            60.6       90.1     100.0%
==========               ========   ========   ========

M              Alfred        69.0      112.5     10.33%
               Henry         63.5      102.5      9.41%
               James         57.3       83.0      7.62%
               Jeffre        62.5       84.0      7.71%
               John          59.0       99.5      9.13%
               Philip        72.0      150.0     13.77%
               Robert        64.8      128.0     11.75%
               Ronald        67.0      133.0     12.21%
               Thomas        57.5       85.0      7.80%
               Willia        66.5      112.0     10.28%
==========               ========   ========   ========
M                            63.9      109.0     100.0%
==========               ========   ========   ========
```

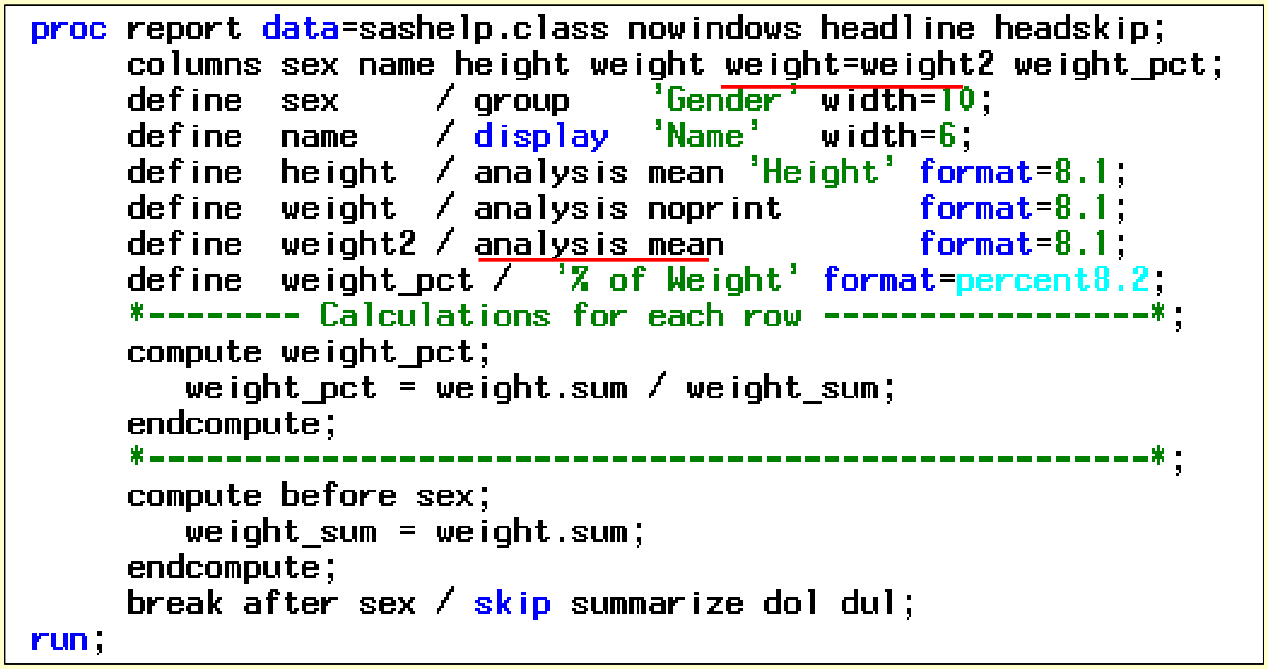
pgm ➪

21

This is going to be tricky because we will need to calculate 2 statistics for WEIGHT…. SUM and MEAN.   We want to display the MEAN, but we need to do the SUM to calculate statistics.

Calculating Percentages

Task 8: Add WEIGHT to the report, calculate its’ AVERAGE for each group.



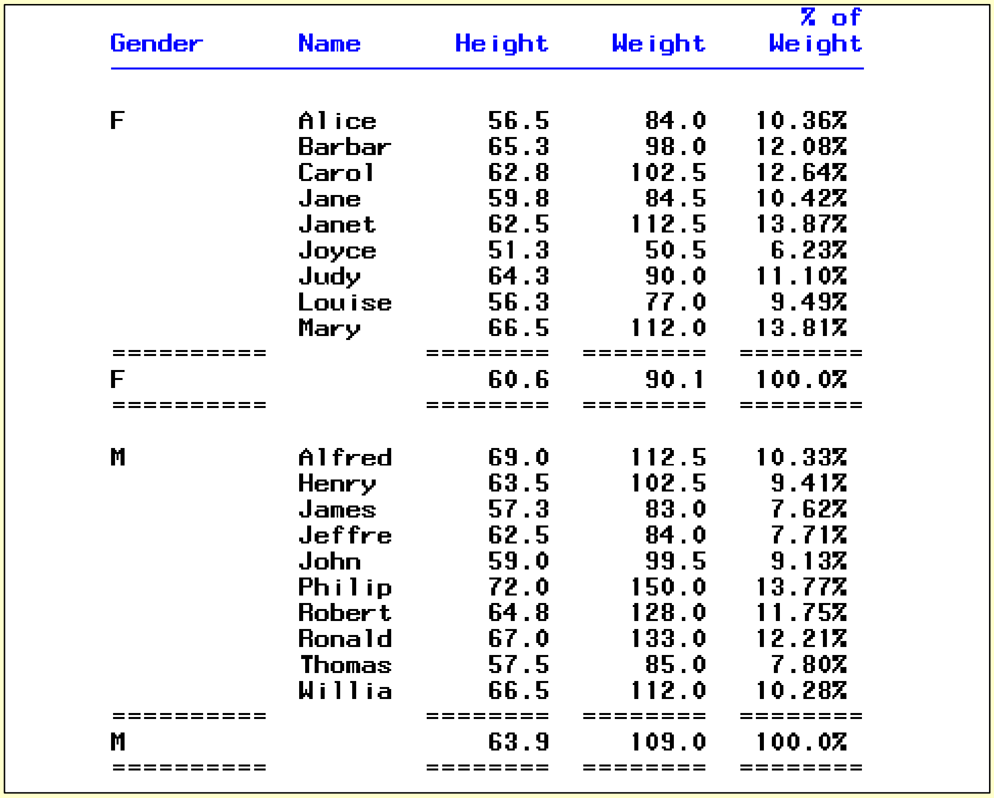
```
proc report data=sashelp.class nowindows headline headskip;
     columns sex name height weight weight=weight2 weight_pct;
     define  sex     / group    'Gender' width=10;
     define  name    / display  'Name'   width=6;
     define  height  / analysis mean 'Height' format=8.1;
     define  weight  / analysis noprint       format=8.1;
     define  weight2 / analysis mean          format=8.1;
     define  weight_pct /  '% of Weight' format=percent8.2;
     *-------- Calculations for each row -----------------*;
     compute weight_pct;
        weight_pct = weight.sum / weight_sum;
     endcompute;
     *----------------------------------------------------*;
     compute before sex;
        weight_sum = weight.sum;
     endcompute;
     break after sex / skip summarize dol dul;
run;
```

Notice the following**:**

- the **WEIGHT** alias (**WEIGHT2**),
- the definitions of the 2 WEIGHT columns (2 stats for WEIGHT).

output ➪

Calculating Percentages

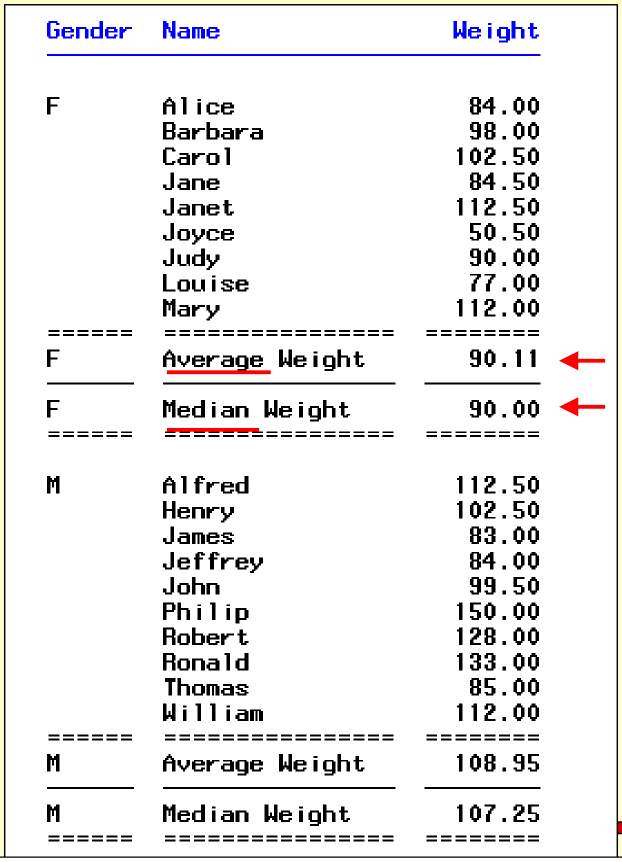


```
                                                      % of
Gender       Name        Height     Weight     Weight
-------------------------------------------------------


F            Alice         56.5       84.0     10.36%
             Barbar        65.3       98.0     12.08%
             Carol         62.8      102.5     12.64%
             Jane          59.8       84.5     10.42%
             Janet         62.5      112.5     13.87%
             Joyce         51.3       50.5      6.23%
             Judy          64.3       90.0     11.10%
             Louise        56.3       77.0      9.49%
             Mary          66.5      112.0     13.81%
==========             ========   ========   ========
F                          60.6       90.1     100.0%
==========             ========   ========   ========

M            Alfred        69.0      112.5     10.33%
             Henry         63.5      102.5      9.41%
             James         57.3       83.0      7.62%
             Jeffre        62.5       84.0      7.71%
             John          59.0       99.5      9.13%
             Philip        72.0      150.0     13.77%
             Robert        64.8      128.0     11.75%
             Ronald        67.0      133.0     12.21%
             Thomas        57.5       85.0      7.80%
             Willia        66.5      112.0     10.28%
==========             ========   ========   ========
M                          63.9      109.0     100.0%
==========             ========   ========   ========
```

Notice the WEIGHT column (WEIGHT2) now displays **averages** for each value of Sex. The percent column has no change from the last report.

Calculating Multiple Statistics in a Column

Task 9. Calculate **two different** statistics for the same column… WEIGHT.



```
Gender   Name                   Weight
------------------------------------------

F        Alice                   84.00
         Barbara                 98.00
         Carol                  102.50
         Jane                    84.50
         Janet                  112.50
         Joyce                   50.50
         Judy                    90.00
         Louise                  77.00
         Mary                   112.00
======   ================     ========
F        Average Weight          90.11
------   ----------------     --------
F        Median Weight           90.00
======   ================     ========

M        Alfred                 112.50
         Henry                  102.50
         James                   83.00
         Jeffrey                 84.00
         John                    99.50
         Philip                 150.00
         Robert                 128.00
         Ronald                 133.00
         Thomas                  85.00
         William                112.00
======   ================     ========
M        Average Weight         108.95
------   ----------------     --------
M        Median Weight          107.25
======   ================     ========
```

The ‘trick’ that makes this work is to have a different **‘by variable’** for each **statistic.** In this case, we need 2 different variables for GENDER: one for MEDIAN, and one for MEAN.

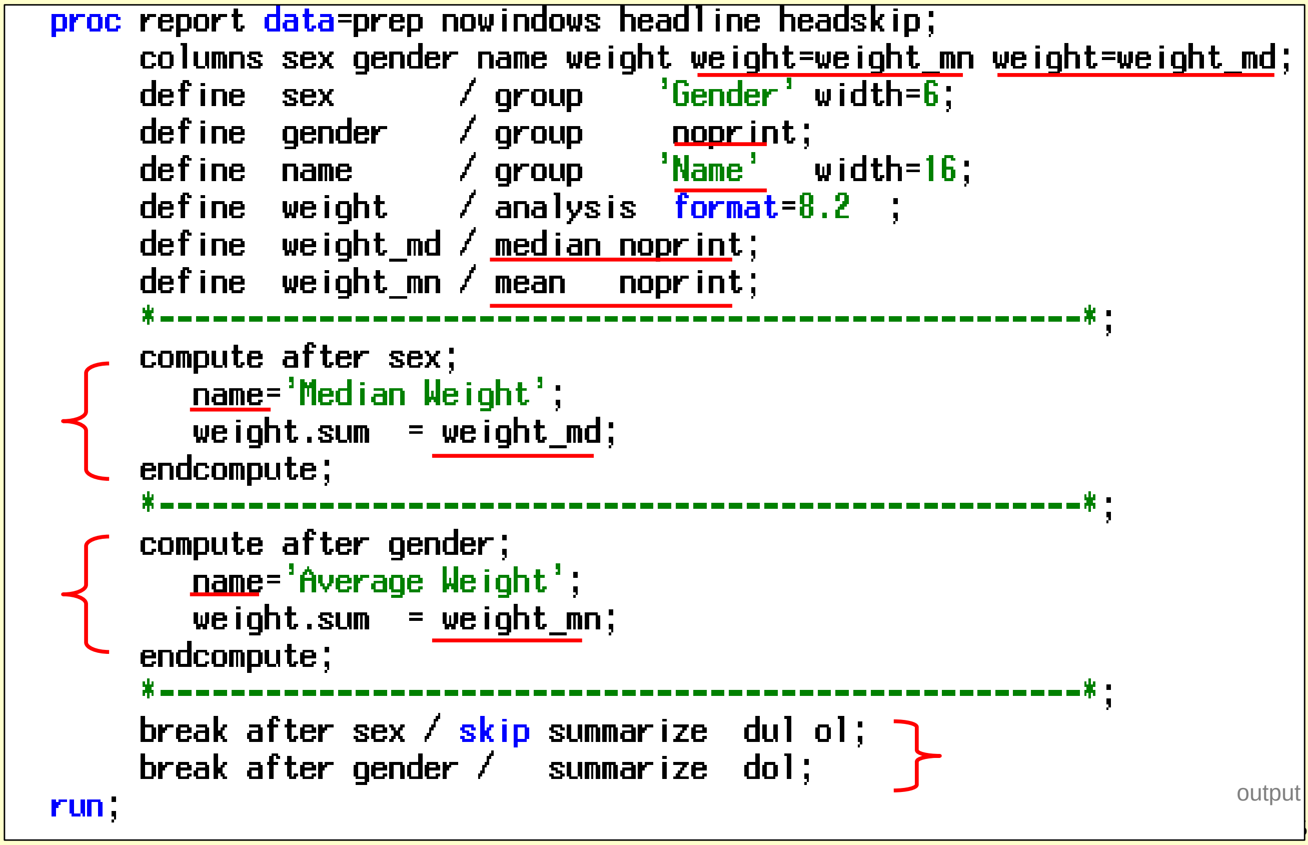
The DATA Step is used to prep the data.

```
data prep;
   length NAME $ 16;
   set SASHELP.CLASS;
   gender = sex;
run;
```

ppm ➪

Calculating Multiple Statistics in a Column

Trick 9. Calculate **two** statistics for WEIGHT.



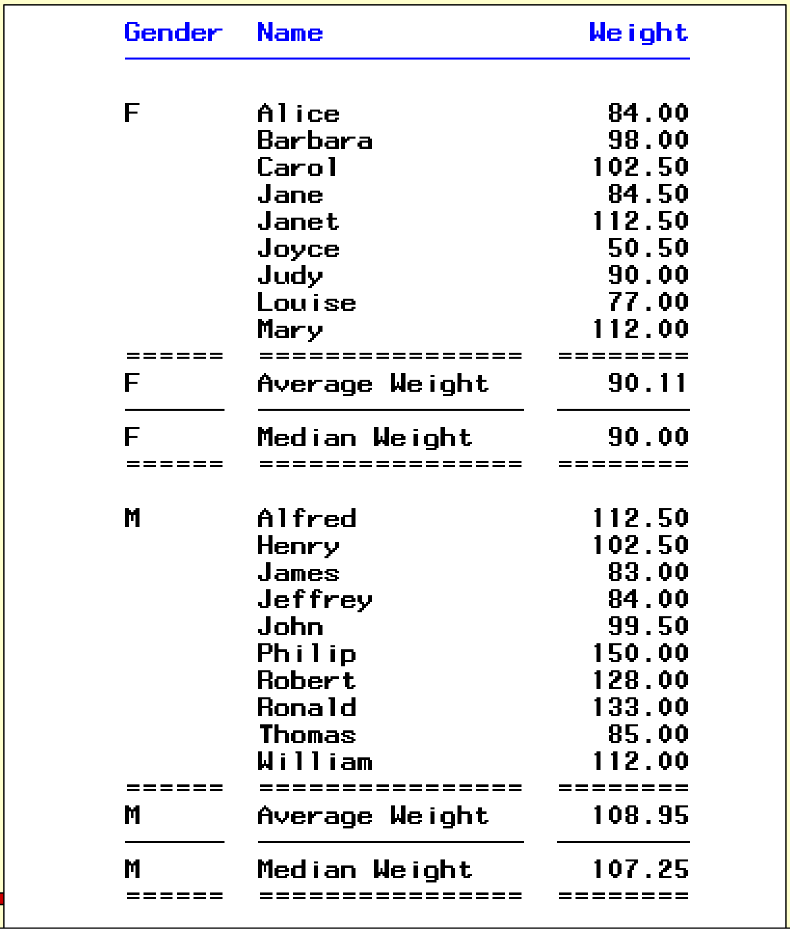
```
   proc report data=prep nowindows headline headskip;
        columns sex gender name weight weight=weight_mn weight=weight_md;
        define  sex       / group    'Gender' width=6;
        define  gender    / group     noprint;
        define  name      / group    'Name'   width=16;
        define  weight    / analysis  format=8.2  ;
        define  weight_md / median noprint;
        define  weight_mn / mean   noprint;
        *----------------------------------------------------*;
        compute after sex;
           name='Median Weight';
           weight.sum  = weight_md;
        endcompute;
        *----------------------------------------------------*;
        compute after gender;
           name='Average Weight';
           weight.sum  = weight_mn;
        endcompute;
        *----------------------------------------------------*;
        break after sex / skip summarize  dul ol;
        break after gender /   summarize  dol;
   run;
```

output ⇨

Calculating Multiple Statistics in a Column

In the program, notice the:

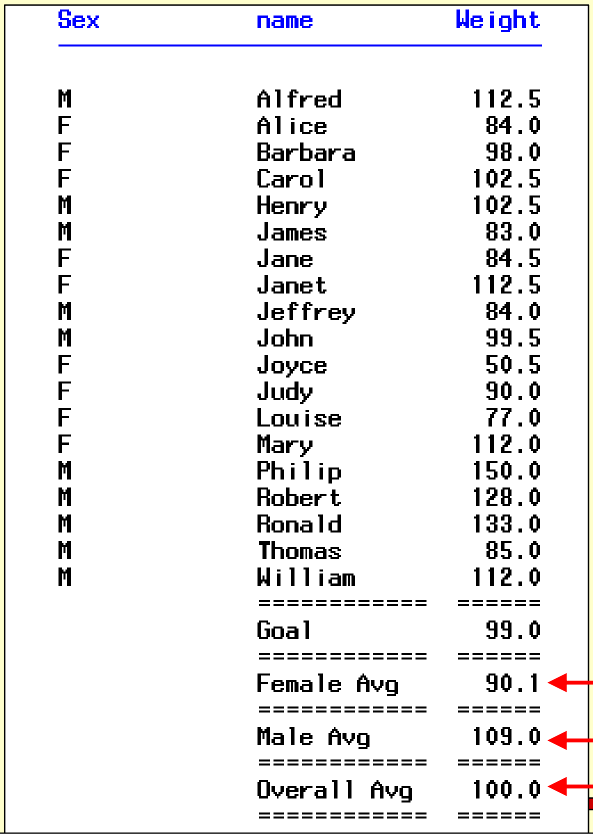
- DATA Step,
- alias' for WEIGHT,
- COMPUTE blocks,
- 3 NOPRINT variables,



```
Gender  Name                  Weight
------------------------------------

F       Alice                  84.00
        Barbara                98.00
        Carol                 102.50
        Jane                   84.50
        Janet                 112.50
        Joyce                  50.50
        Judy                   90.00
        Louise                 77.00
        Mary                  112.00
======  ================    ========
F       Average Weight         90.11
------  ----------------    --------
F       Median Weight          90.00
======  ================    ========

M       Alfred                112.50
        Henry                 102.50
        James                  83.00
        Jeffrey                84.00
        John                   99.50
        Philip                150.00
        Robert                128.00
        Ronald                133.00
        Thomas                 85.00
        William               112.00
======  ================    ========
M       Average Weight        108.95
------  ----------------    --------
M       Median Weight         107.25
======  ================    ========
```

Calculating Statistics on Different Values

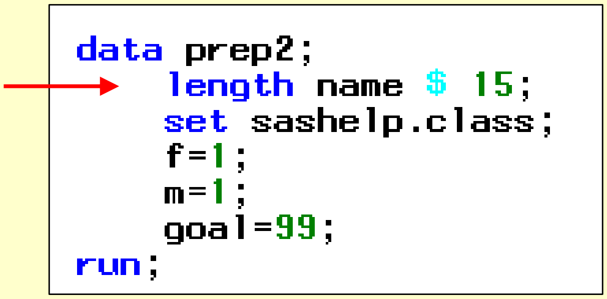
Trick 10. Calculate **Average** WEIGHT for Females, Males and the Overall Average, and place these in the same column at the **end** of the report.



```
Sex              name          Weight
_____________________________________


M                Alfred         112.5
F                Alice           84.0
F                Barbara         98.0
F                Carol          102.5
M                Henry          102.5
M                James           83.0
F                Jane            84.5
F                Janet          112.5
M                Jeffrey         84.0
M                John            99.5
F                Joyce           50.5
F                Judy            90.0
F                Louise          77.0
F                Mary           112.0
M                Philip         150.0
M                Robert         128.0
M                Ronald         133.0
M                Thomas          85.0
M                William        112.0
                 ============  ======
                 Goal            99.0
                 ============  ======
                 Female Avg      90.1
                 ============  ======
                 Male Avg       109.0
                 ============  ======
                 Overall Avg    100.0
                 ============  ======
```

The 'trick' that makes this work is to have a different **GROUP variable** for each **GENDER,** plus a group variable for all genders**.**

Again, the DATA Step is used to prep the data.



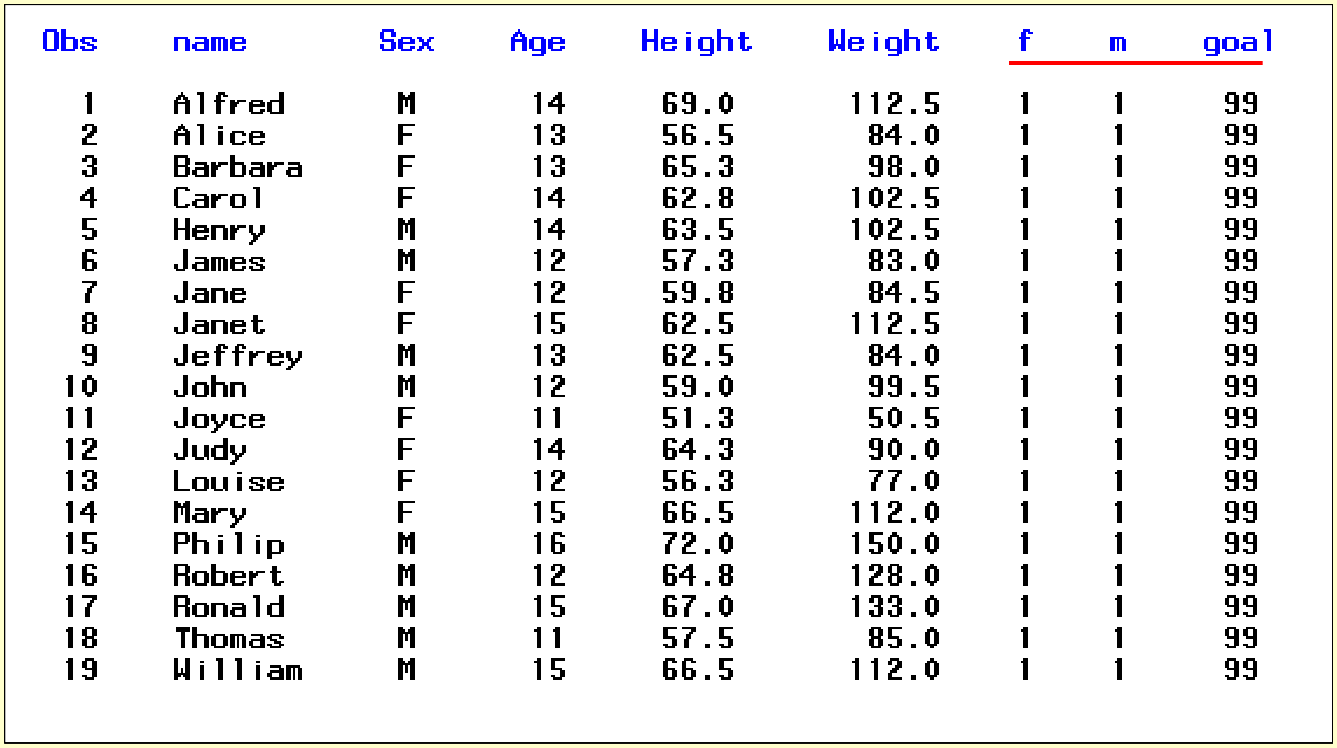
```
data prep2;
     length name $ 15;
     set sashelp.class;
     f=1;
     m=1;
     goal=99;
run;
```

data ➪

27

Calculating Statistics on Different Values

The WORK.PREP2 data set.

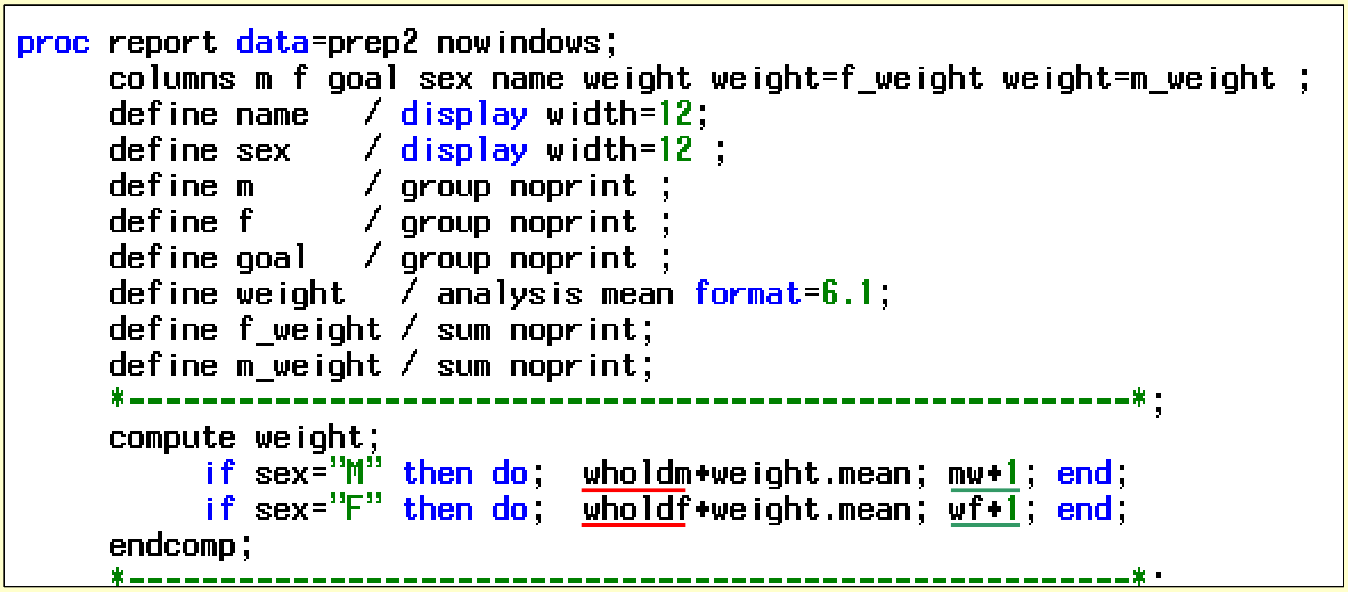


```
 Obs    name       Sex    Age    Height    Weight    f    m    goal

   1    Alfred      M      14     69.0     112.5     1    1     99
   2    Alice       F      13     56.5      84.0     1    1     99
   3    Barbara     F      13     65.3      98.0     1    1     99
   4    Carol       F      14     62.8     102.5     1    1     99
   5    Henry       M      14     63.5     102.5     1    1     99
   6    James       M      12     57.3      83.0     1    1     99
   7    Jane        F      12     59.8      84.5     1    1     99
   8    Janet       F      15     62.5     112.5     1    1     99
   9    Jeffrey     M      13     62.5      84.0     1    1     99
  10    John        M      12     59.0      99.5     1    1     99
  11    Joyce       F      11     51.3      50.5     1    1     99
  12    Judy        F      14     64.3      90.0     1    1     99
  13    Louise      F      12     56.3      77.0     1    1     99
  14    Mary        F      15     66.5     112.0     1    1     99
  15    Philip      M      16     72.0     150.0     1    1     99
  16    Robert      M      12     64.8     128.0     1    1     99
  17    Ronald      M      15     67.0     133.0     1    1     99
  18    Thomas      M      11     57.5      85.0     1    1     99
  19    William     M      15     66.5     112.0     1    1     99
```

pgm ➪

Calculating Statistics on Different Values

Task 10. Calculate Average WEIGHT for Females, Males and the Overall Average.



```
proc report data=prep2 nowindows;
     columns m f goal sex name weight weight=f_weight weight=m_weight ;
     define name    / display width=12;
     define sex     / display width=12 ;
     define m       / group noprint ;
     define f       / group noprint ;
     define goal    / group noprint ;
     define weight    / analysis mean format=6.1;
     define f_weight / sum noprint;
     define m_weight / sum noprint;
     *-------------------------------------------------------*;
     compute weight;
          if sex="M" then do;  wholdm+weight.mean; mw+1; end;
          if sex="F" then do;  wholdf+weight.mean; wf+1; end;
     endcomp;
     *-------------------------------------------------------*;
```

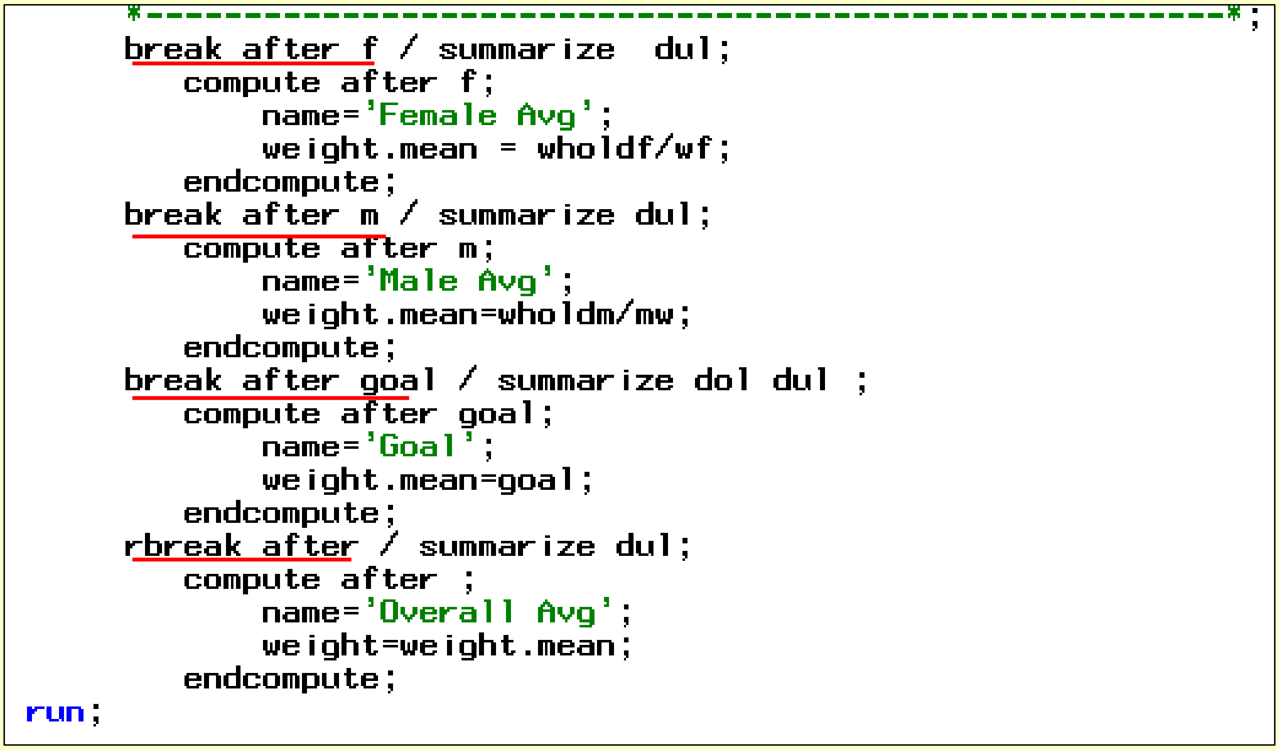
Notice the ‘Holding’ variables in this partial PROC REPORT step.

Notice the ‘Counter’ variables in this partial PROC REPORT step.

more ➪

Calculating Statistics on Different Values

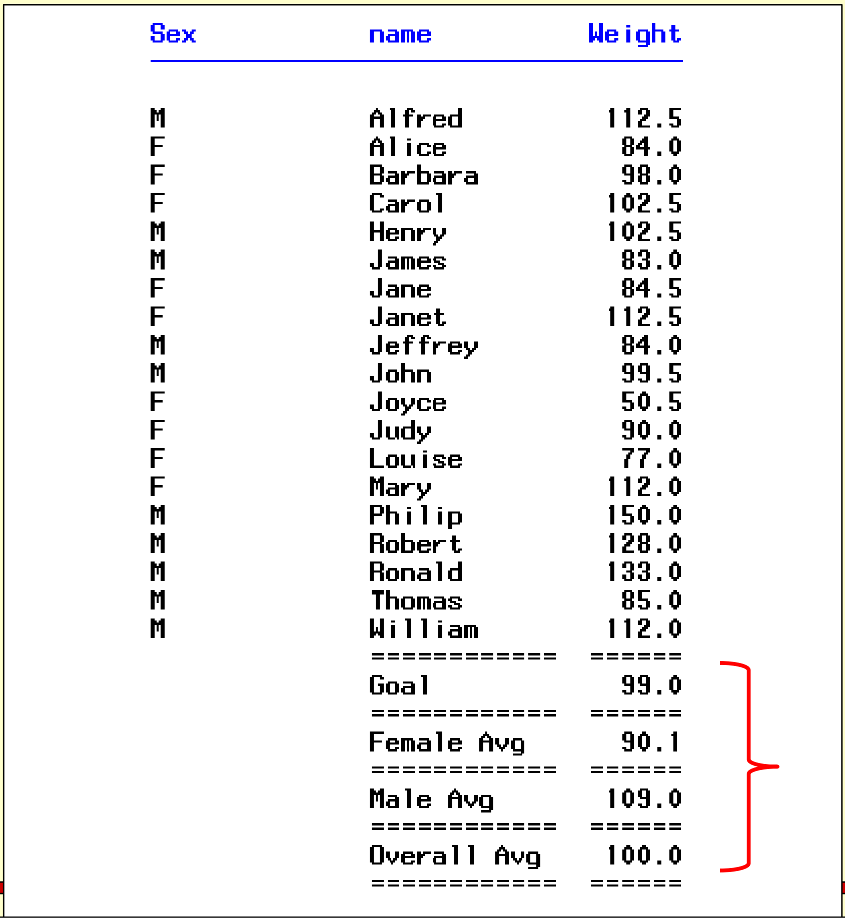
Notice the BREAK and RBREAK statements at the end of the PROC REPORT step.



```
     *--------------------------------------------------------*;
     break after f / summarize  dul;
        compute after f;
            name='Female Avg';
            weight.mean = wholdf/wf;
        endcompute;
     break after m / summarize dul;
        compute after m;
            name='Male Avg';
            weight.mean=wholdm/mw;
        endcompute;
     break after goal / summarize dol dul ;
        compute after goal;
            name='Goal';
            weight.mean=goal;
        endcompute;
     rbreak after / summarize dul;
        compute after ;
            name='Overall Avg';
            weight=weight.mean;
        endcompute;
run;
```

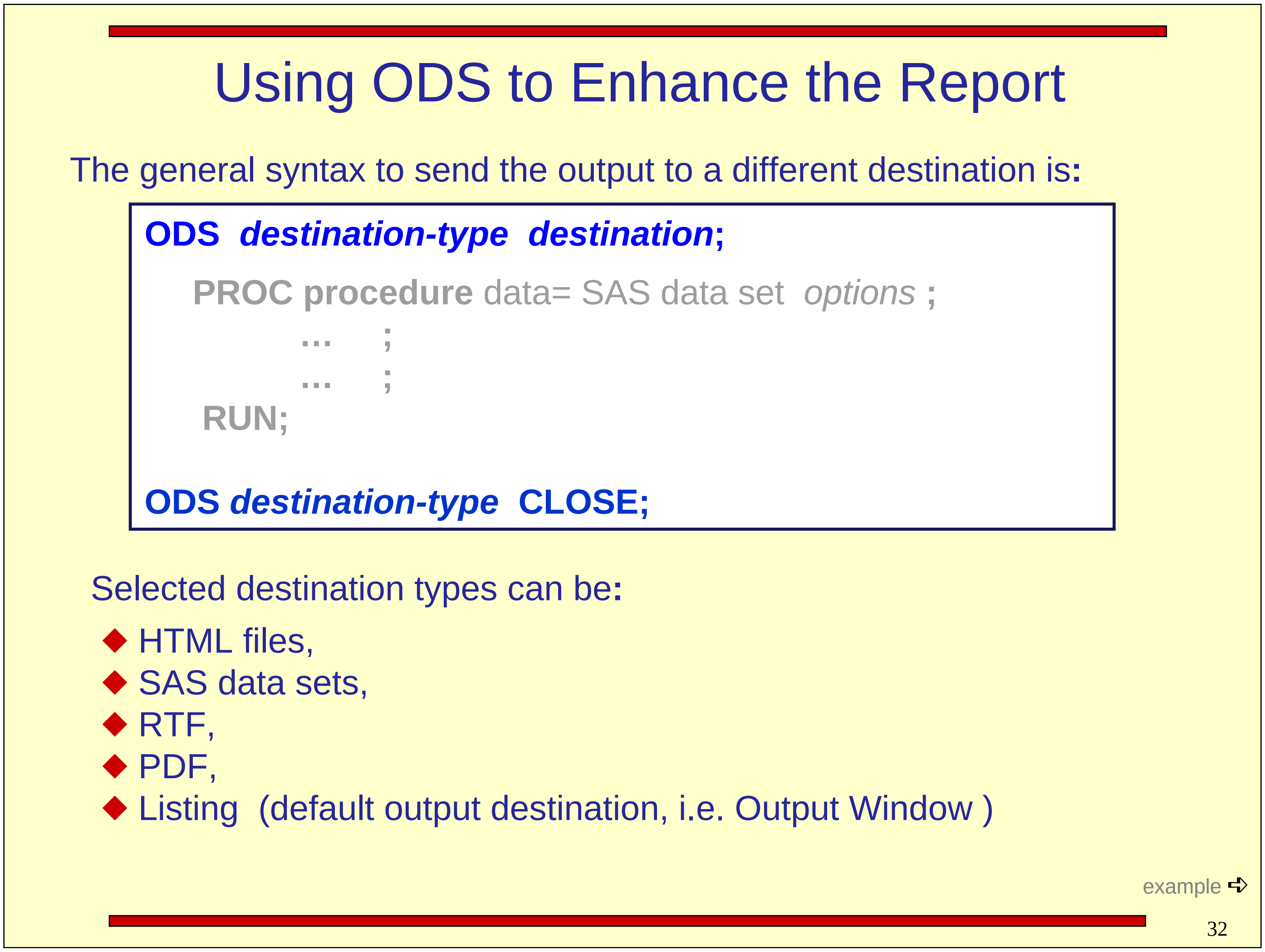
Notice the reassigning of the NAME variable in each of the COMPUTE BLOCKS.

Calculating Statistics on Different Values



```
Sex           name           Weight
-----------------------------------

M             Alfred          112.5
F             Alice            84.0
F             Barbara          98.0
F             Carol           102.5
M             Henry           102.5
M             James            83.0
F             Jane             84.5
F             Janet           112.5
M             Jeffrey          84.0
M             John             99.5
F             Joyce            50.5
F             Judy             90.0
F             Louise           77.0
F             Mary            112.0
M             Philip          150.0
M             Robert          128.0
M             Ronald          133.0
M             Thomas           85.0
M             William         112.0
              ============   ======
              Goal             99.0
              ============   ======
              Female Avg       90.1
              ============   ======
              Male Avg        109.0
              ============   ======
              Overall Avg     100.0
              ============   ======
```

ods ➪



Using ODS to Enhance the Report

The general syntax to send the output to a different destination is**:**

```
ODS  destination-type  destination;

    PROC procedure data= SAS data set  options ;
              ...    ;
              ...    ;
     RUN;

ODS destination-type  CLOSE;
```

Selected destination types can be**:**

- HTML files,
- SAS data sets,
- RTF,
- PDF,
- Listing (default output destination, i.e. Output Window)

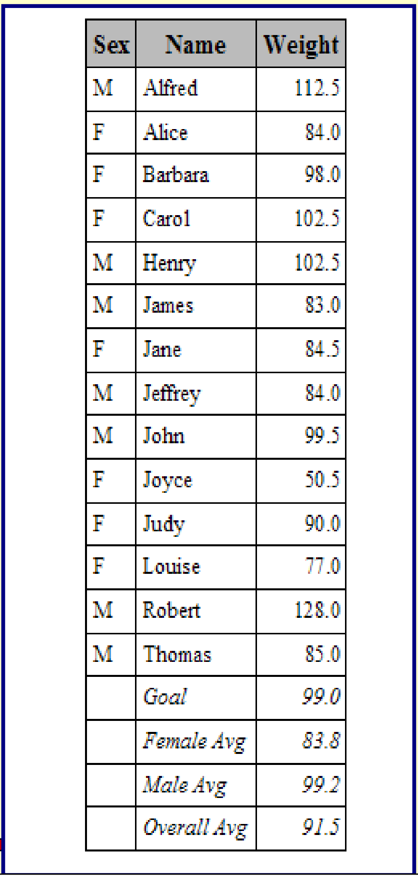
example ➪

32

Using ODS to Enhance the Report

Trick 9. ‘Sandwich’ the previous PROC REPORT step in between basic ODS statements.

```
ods rtf  file = ‘c:\sgf.rtf’ ;

    previous PROC REPORT step . . .  ;
              …     ;
              …     ;
     RUN;

ods  rtf  CLOSE;
```



Sex	Name	Weight
M	Alfred	112.5
F	Alice	84.0
F	Barbara	98.0
F	Carol	102.5
M	Henry	102.5
M	James	83.0
F	Jane	84.5
M	Jeffrey	84.0
M	John	99.5
F	Joyce	50.5
F	Judy	90.0
F	Louise	77.0
M	Robert	128.0
M	Thomas	85.0
	Goal	*99.0*
	Female Avg	*83.8*
	Male Avg	*99.2*
	Overall Avg	*91.5*

This is the **default** appearance when using ODS to write to an RTF file. The report can be enhanced by using some new ODS syntax…

more ➪

Using ODS STYLES to Enhance the Report

The **STYLE =** option can be used to control just about every aspect of the Report’s appearance**.**

The typical form of the STYLE = option is**:**

```
STYLE = { attribute – 1 = value – 1 …
              attribute – n = value – n  } ;
```

where ‘attribute’ is a report feature such as **:**

- background
- foreground
- font

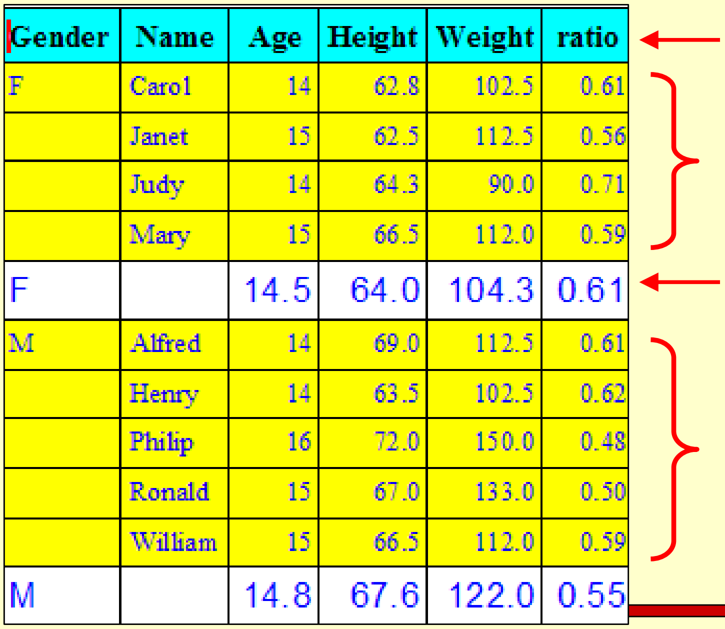
The STYLE = option can be abbreviated as S= **.**

➪

Using ODS STYLES to Enhance the Report

The STYLE = (COMPONENT) = {attribute = value } syntax can also be used to control the appearance of the report.

The following ‘COMPONENTS’ can be controlled by the STYLE = option:



Gender	Name	Age	Height	Weight	ratio
F	Carol	14	62.8	102.5	0.61
	Janet	15	62.5	112.5	0.56
	Judy	14	64.3	90.0	0.71
	Mary	15	66.5	112.0	0.59
F		14.5	64.0	104.3	0.61
M	Alfred	14	69.0	112.5	0.61
	Henry	14	63.5	102.5	0.62
	Philip	16	72.0	150.0	0.48
	Ronald	15	67.0	133.0	0.50
	William	15	66.5	112.0	0.59
M		14.8	67.6	122.0	0.55

Header = {background=cyan}

Report = {background=yellow}

Summary = {font=‘Arial’ … }

Column= {foreground=blue}

Using ODS STYLES to Enhance the Report

Trick 10. Use ODS STYLES to enhance the report. Modify Trick 3's example.

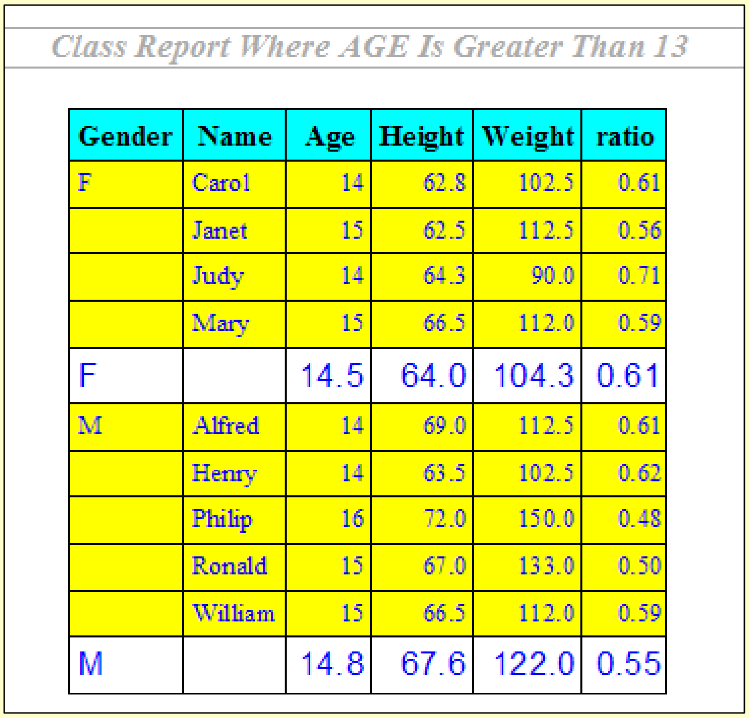
```
ods rtf file='c:\sugi30.rtf';

    title 'Class Report Where AGE Is Greater Than 13';
    proc report data=sashelp.class(where=(age ge 14)) nowd
         style(report) = {background=yellow}
         style(header) = {background=cyan}
         style(summary)= {font_size=13pt background=white font=('Arial')}
         style(column) = {foreground=blue} ;
         columns sex name age height weight ratio;
         define sex    / group     'Gender'   width=10;
         define name   / display  'Name' width=6;
         define age    / analysis mean 'Age'    width=4;
         define height / analysis mean 'Height' format=8.1;
         define weight / analysis mean 'Weight' format=8.1;
         define ratio  / computed format=6.2;
         compute ratio;
            ratio = height.mean / weight.mean;
         endcompute;
         break after sex / skip summarize dol dul;
     run;

 ods rtf close;
```

Notice the STYLE options and their placement on the PROC statement.

output ➪

Using ODS STYLES to Enhance the Report

Trick 10. Output.



Class Report Where AGE Is Greater Than 13

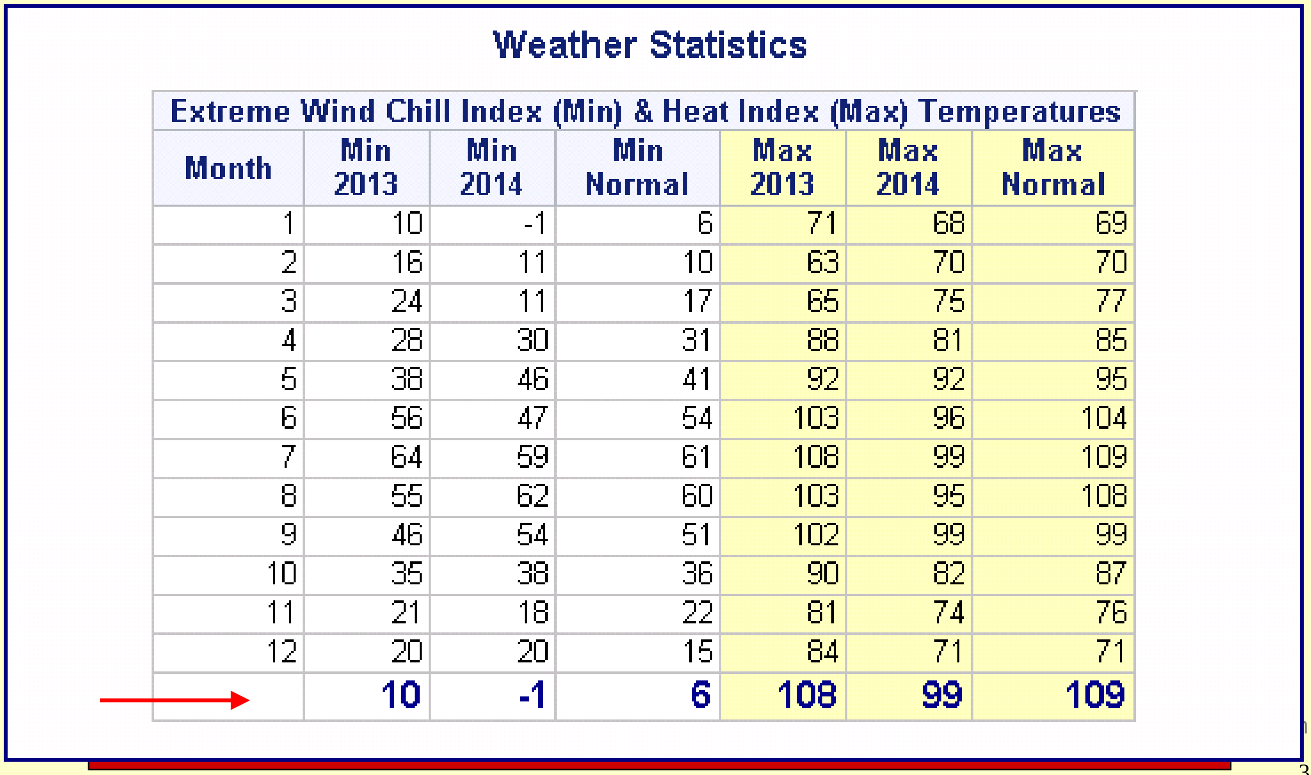
Gender	Name	Age	Height	Weight	ratio
F	Carol	14	62.8	102.5	0.61
	Janet	15	62.5	112.5	0.56
	Judy	14	64.3	90.0	0.71
	Mary	15	66.5	112.0	0.59
F		14.5	64.0	104.3	0.61
M	Alfred	14	69.0	112.5	0.61
	Henry	14	63.5	102.5	0.62
	Philip	16	72.0	150.0	0.48
	Ronald	15	67.0	133.0	0.50
	William	15	66.5	112.0	0.59
M		14.8	67.6	122.0	0.55

Notice the font sizes.

output ➪

Using ODS STYLES to Enhance the Report

Trick 11. Use a single row to Summarize different statistics. The same row shows Minimum and Maximum values for different columns.



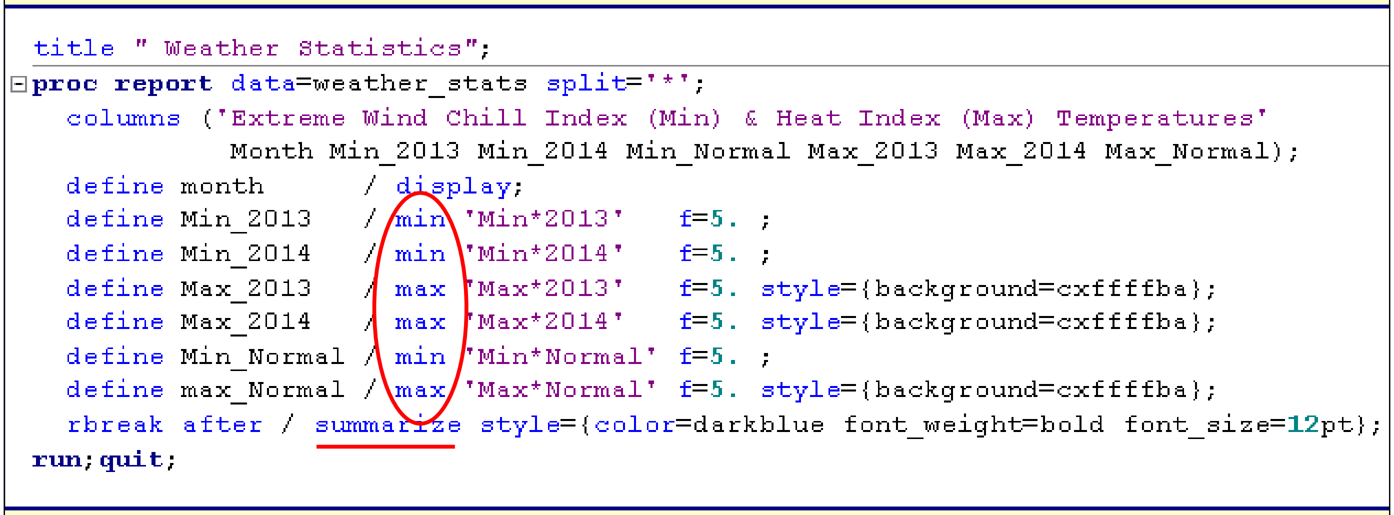
Weather Statistics

Extreme Wind Chill Index (Min) & Heat Index (Max) Temperatures

Month	Min 2013	Min 2014	Min Normal	Max 2013	Max 2014	Max Normal
1	10	-1	6	71	68	69
2	16	11	10	63	70	70
3	24	11	17	65	75	77
4	28	30	31	88	81	85
5	38	46	41	92	92	95
6	56	47	54	103	96	104
7	64	59	61	108	99	109
8	55	62	60	103	95	108
9	46	54	51	102	99	99
10	35	38	36	90	82	87
11	21	18	22	81	74	76
12	20	20	15	84	71	71
	10	**-1**	**6**	**108**	**99**	**109**

Using ODS STYLES to Enhance the Report

Notice the different statistics in the DEFINE statements. When the RBREAK statement summarizes, it uses the statistics in the DEFINE statements.

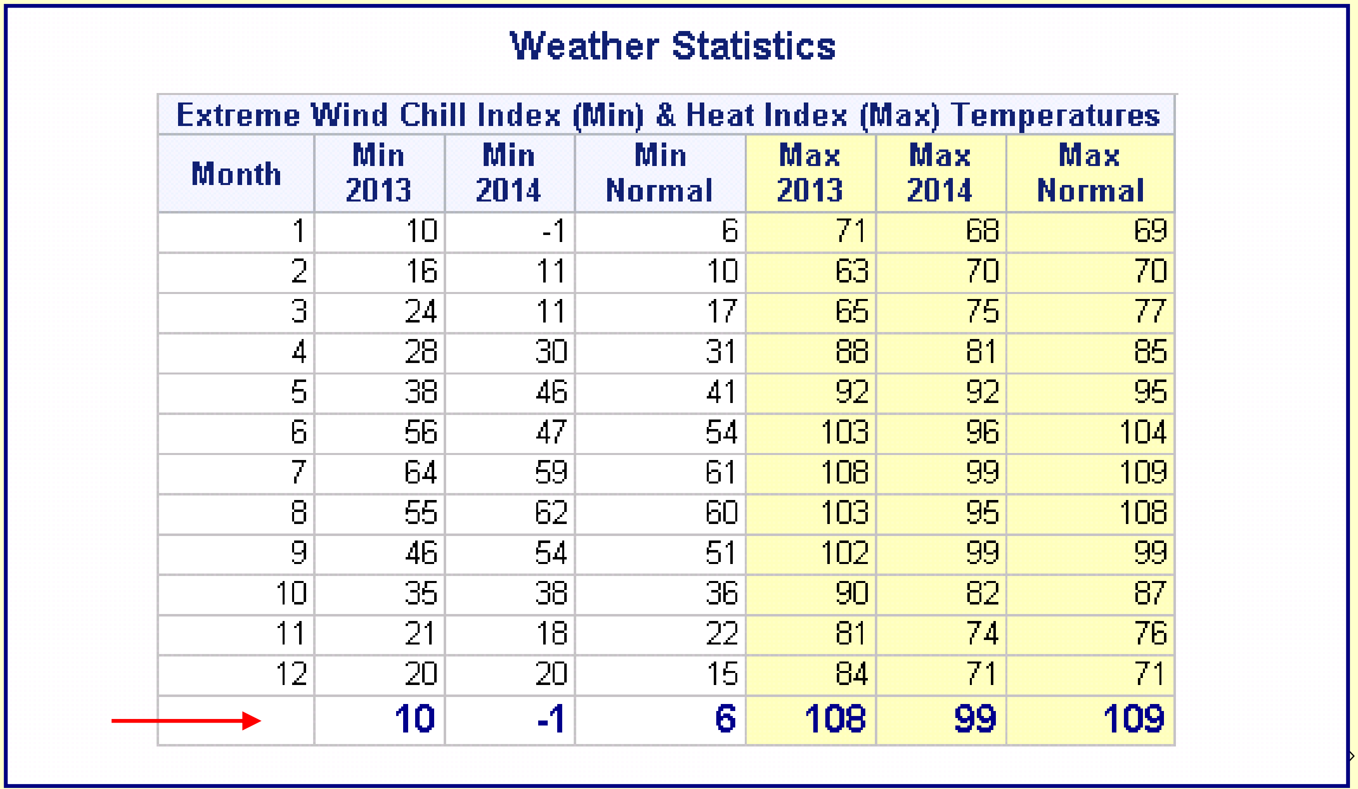


```
 title " Weather Statistics";
⊟proc report data=weather_stats split='*';
   columns ('Extreme Wind Chill Index (Min) & Heat Index (Max) Temperatures'
             Month Min_2013 Min_2014 Min_Normal Max_2013 Max_2014 Max_Normal);
   define month      / display;
   define Min_2013   / min 'Min*2013'   f=5. ;
   define Min_2014   / min 'Min*2014'   f=5. ;
   define Max_2013   / max 'Max*2013'   f=5. style={background=cxffffba};
   define Max_2014   / max 'Max*2014'   f=5. style={background=cxffffba};
   define Min_Normal / min 'Min*Normal' f=5. ;
   define max_Normal / max 'Max*Normal' f=5. style={background=cxffffba};
   rbreak after / summarize style={color=darkblue font_weight=bold font_size=12pt};
 run;quit;
```

pgm ➪

Using ODS STYLES to Enhance the Report

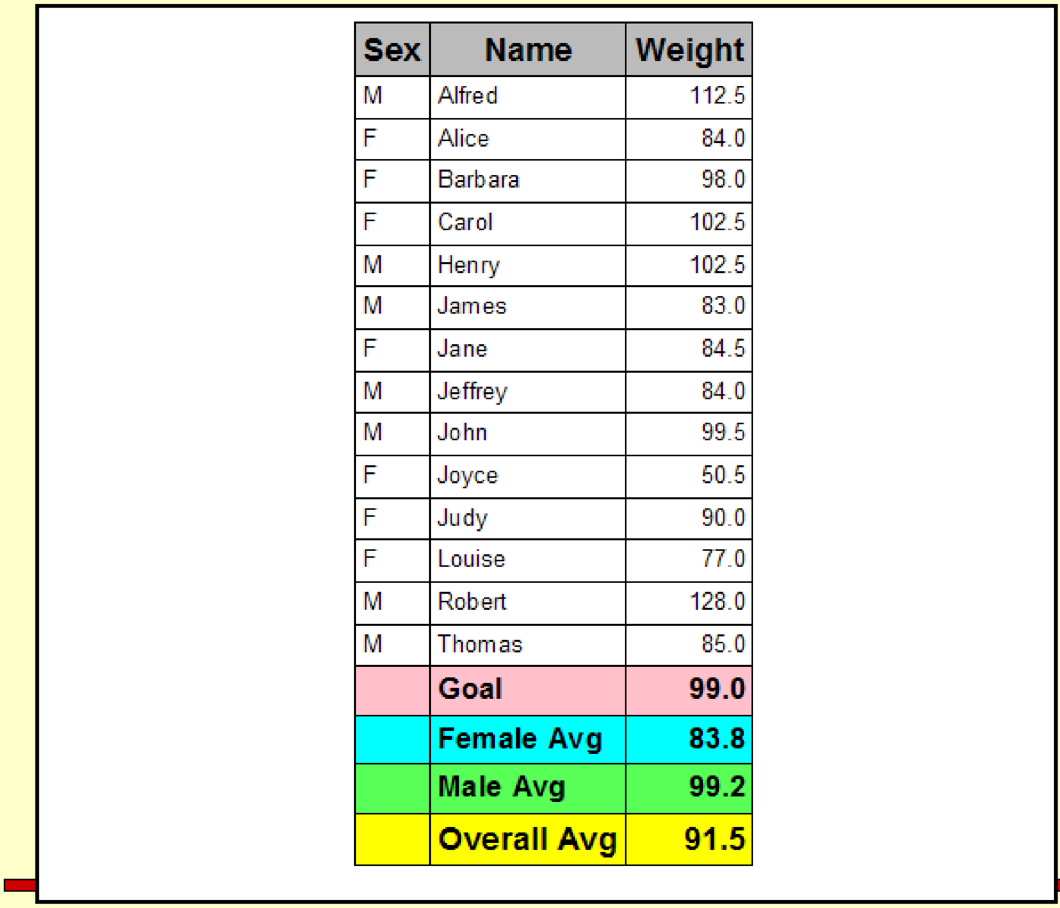
Trick 11 output.



Weather Statistics

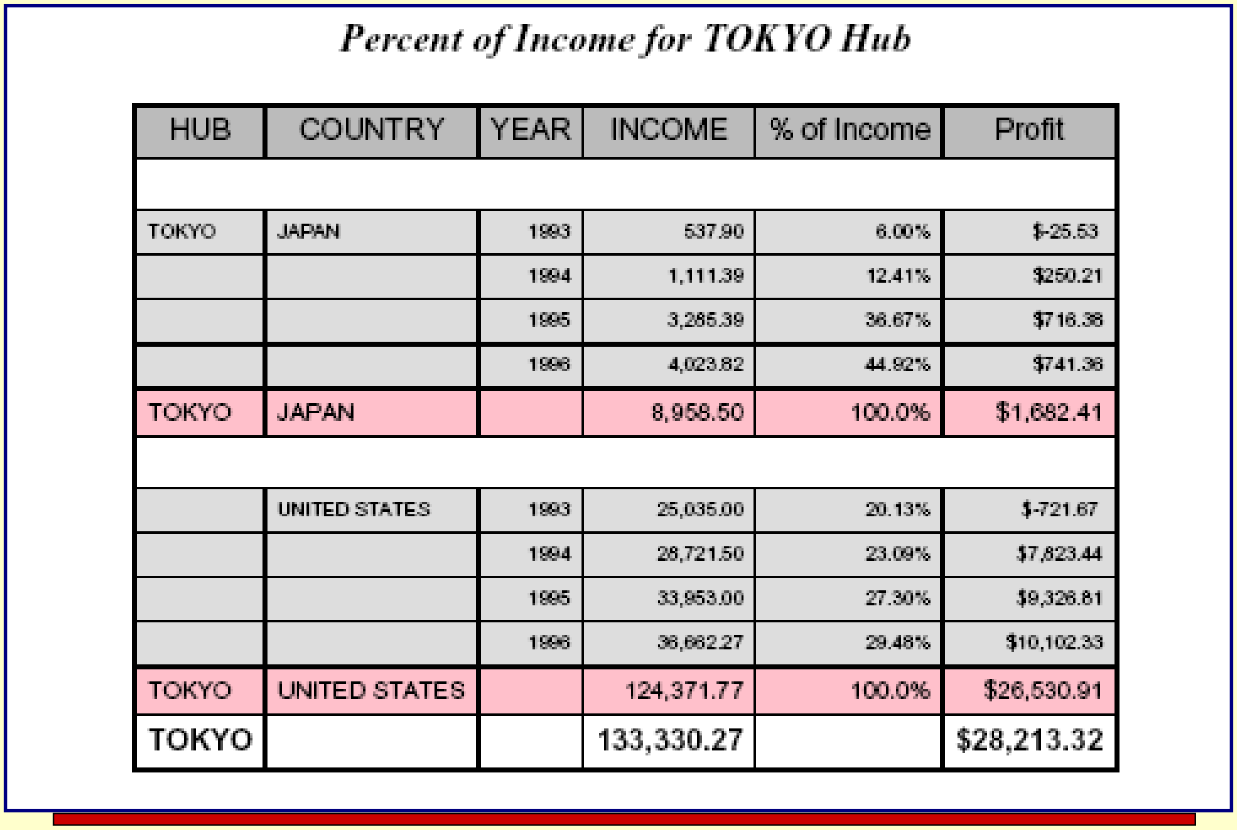
Extreme Wind Chill Index (Min) & Heat Index (Max) Temperatures						
Month	Min 2013	Min 2014	Min Normal	Max 2013	Max 2014	Max Normal
1	10	-1	6	71	68	69
2	16	11	10	63	70	70
3	24	11	17	65	75	77
4	28	30	31	88	81	85
5	38	46	41	92	92	95
6	56	47	54	103	96	104
7	64	59	61	108	99	109
8	55	62	60	103	95	108
9	46	54	51	102	99	99
10	35	38	36	90	82	87
11	21	18	22	81	74	76
12	20	20	15	84	71	71
	10	**-1**	**6**	**108**	**99**	**109**

Other ODS Examples



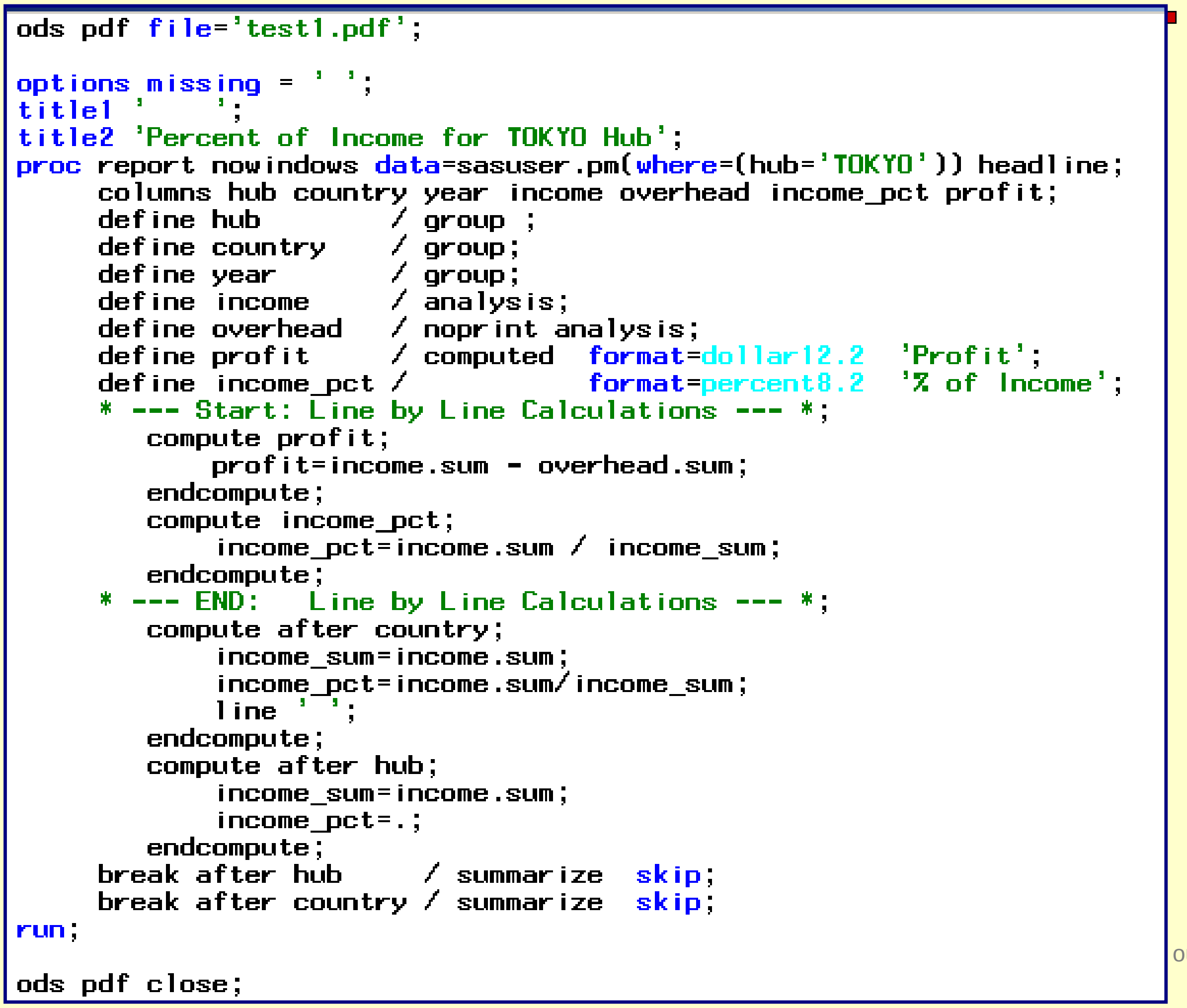
Sex	Name	Weight
M	Alfred	112.5
F	Alice	84.0
F	Barbara	98.0
F	Carol	102.5
M	Henry	102.5
M	James	83.0
F	Jane	84.5
M	Jeffrey	84.0
M	John	99.5
F	Joyce	50.5
F	Judy	90.0
F	Louise	77.0
M	Robert	128.0
M	Thomas	85.0
	Goal	**99.0**
	Female Avg	**83.8**
	Male Avg	**99.2**
	Overall Avg	**91.5**

Other ODS Examples



Percent of Income for TOKYO Hub

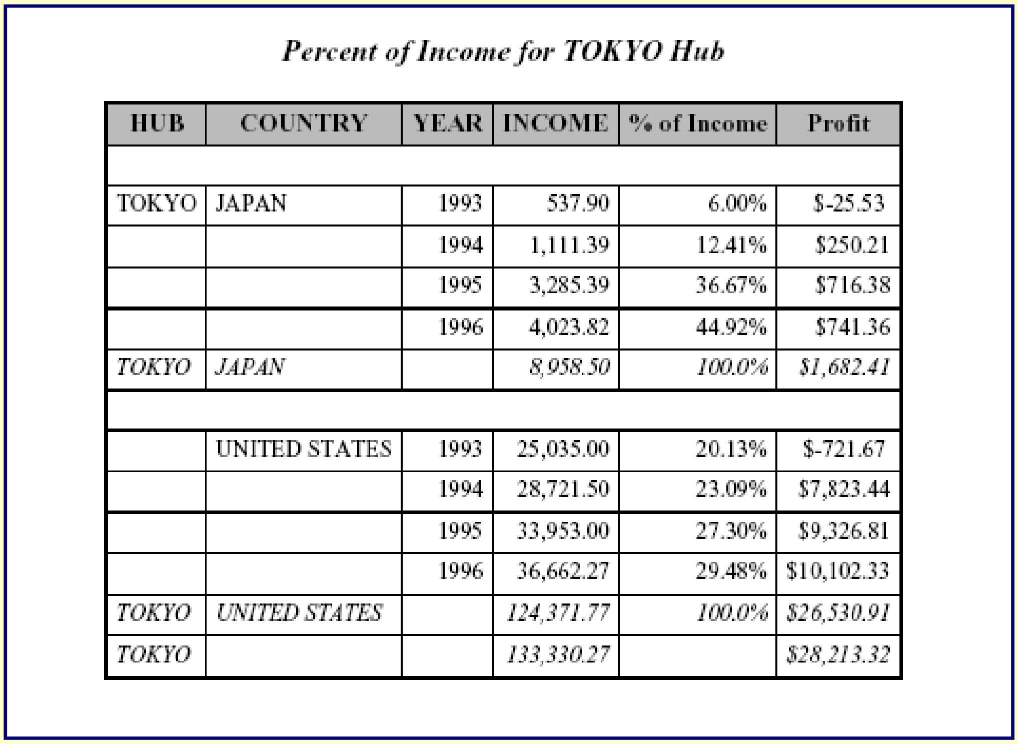
HUB	COUNTRY	YEAR	INCOME	% of Income	Profit
TOKYO	JAPAN	1993	537.90	6.00%	$-25.53
		1994	1,111.39	12.41%	$250.21
		1995	3,285.39	36.67%	$716.38
		1996	4,023.82	44.92%	$741.36
TOKYO	JAPAN		8,958.50	100.0%	$1,682.41
	UNITED STATES	1993	25,035.00	20.13%	$-721.67
		1994	28,721.50	23.09%	$7,823.44
		1995	33,953.00	27.30%	$9,326.81
		1996	36,662.27	29.48%	$10,102.33
TOKYO	UNITED STATES		124,371.77	100.0%	$26,530.91
TOKYO			**133,330.27**		**$28,213.32**



```
ods pdf file='test1.pdf';

options missing = ' ';
title1 '     ';
title2 'Percent of Income for TOKYO Hub';
proc report nowindows data=sasuser.pm(where=(hub='TOKYO')) headline;
     columns hub country year income overhead income_pct profit;
     define hub        / group ;
     define country    / group;
     define year       / group;
     define income     / analysis;
     define overhead   / noprint analysis;
     define profit     / computed   format=dollar12.2  'Profit';
     define income_pct /            format=percent8.2  '% of Income';
     * --- Start: Line by Line Calculations --- *;
        compute profit;
            profit=income.sum - overhead.sum;
        endcompute;
        compute income_pct;
            income_pct=income.sum / income_sum;
        endcompute;
     * --- END:   Line by Line Calculations --- *;
        compute after country;
            income_sum=income.sum;
            income_pct=income.sum/income_sum;
            line ' ';
        endcompute;
        compute after hub;
            income_sum=income.sum;
            income_pct=.;
        endcompute;
     break after hub     / summarize  skip;
     break after country / summarize  skip;
run;

ods pdf close;
```

output ➪

Using ODS to Enhance the Report



Percent of Income for TOKYO Hub

HUB	COUNTRY	YEAR	INCOME	% of Income	Profit
TOKYO	JAPAN	1993	537.90	6.00%	$-25.53
		1994	1,111.39	12.41%	$250.21
		1995	3,285.39	36.67%	$716.38
		1996	4,023.82	44.92%	$741.36
TOKYO	*JAPAN*		*8,958.50*	*100.0%*	*$1,682.41*
	UNITED STATES	1993	25,035.00	20.13%	$-721.67
		1994	28,721.50	23.09%	$7,823.44
		1995	33,953.00	27.30%	$9,326.81
		1996	36,662.27	29.48%	$10,102.33
TOKYO	*UNITED STATES*		*124,371.77*	*100.0%*	*$26,530.91*
TOKYO			*133,330.27*		*$28,213.32*

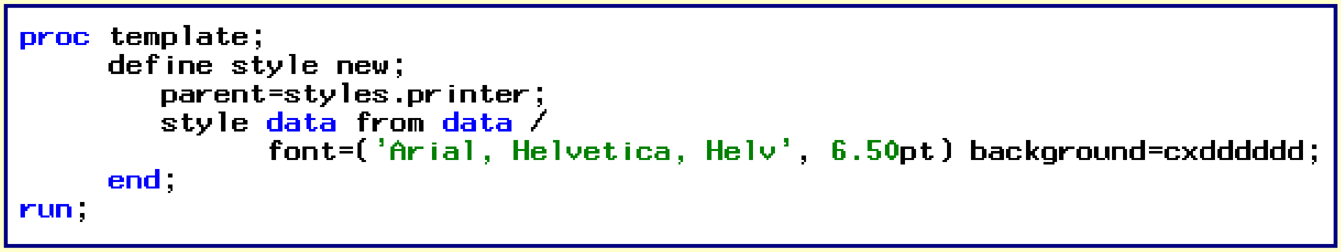
This is the default appearance when using ODS to write to a PDF file.

This report can be enhanced by using some new ODS syntax…

Using ODS to Enhance the Report

The TEMPLATE procedure allows you to control the appearance of almost every aspect of the report ... the font style, font weight, font face, and color.

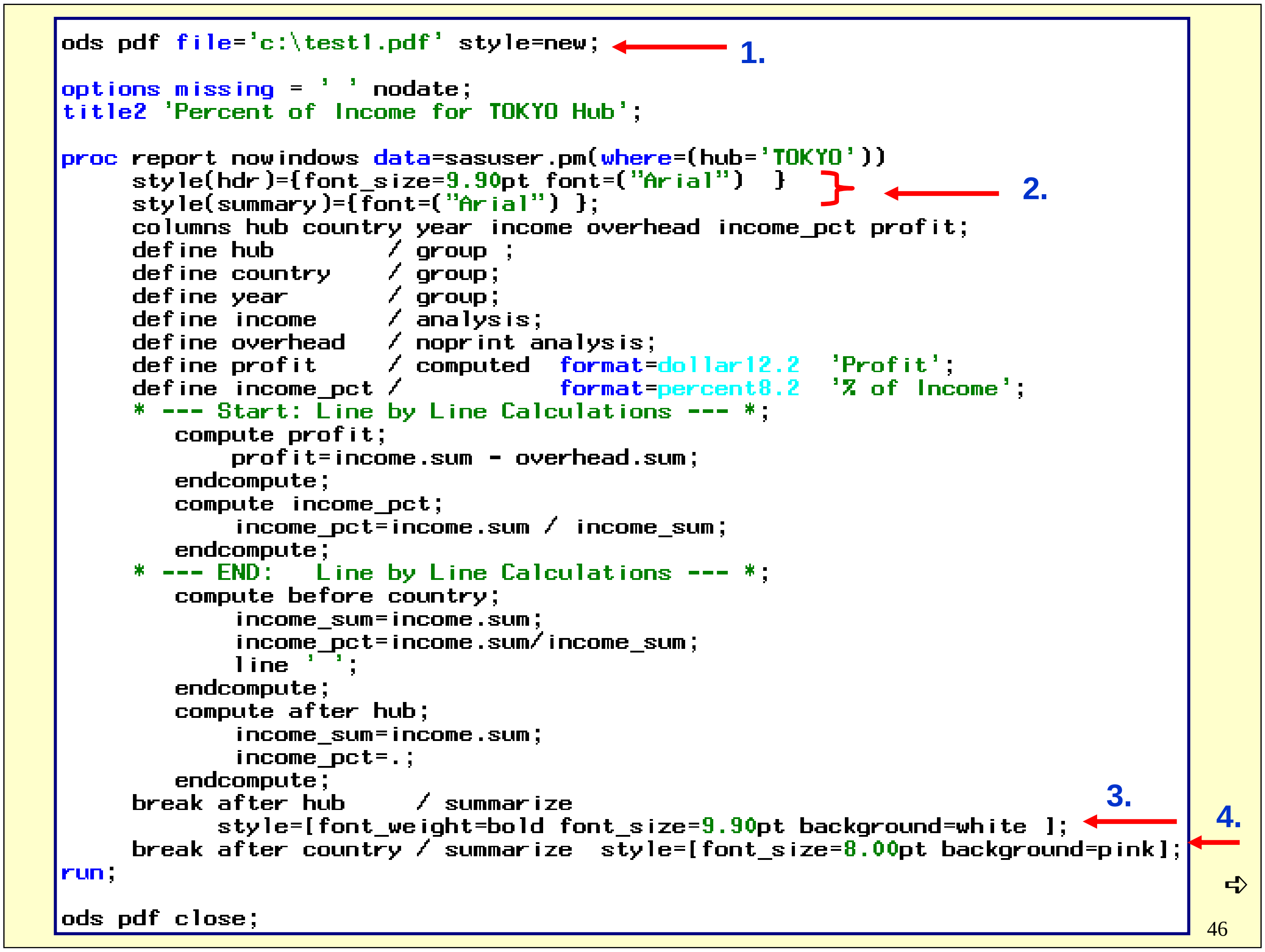
Use the TEMPLATE procedure to define a style (**NEW**) that controls the background color, font face, and font size of the data at the most detail level.



```
proc template;
     define style new;
        parent=styles.printer;
        style data from data /
              font=('Arial, Helvetica, Helv', 6.50pt) background=cxdddddd;
     end;
run;
```

The background color: cx dddddd is a medium gray.

On the next page, ODS is invoked along with the **NEW** style…



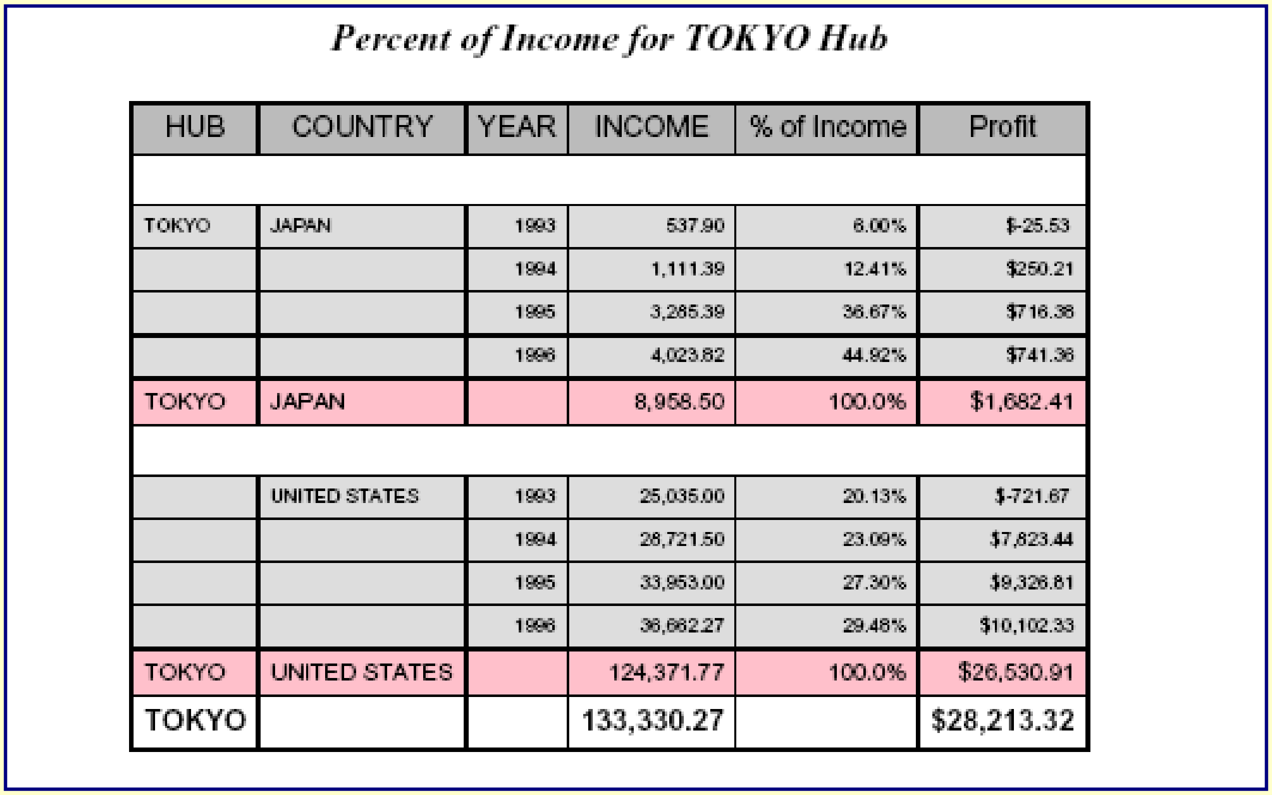
```
ods pdf file='c:\test1.pdf' style=new;

options missing = ' ' nodate;
title2 'Percent of Income for TOKYO Hub';

proc report nowindows data=sasuser.pm(where=(hub='TOKYO'))
     style(hdr)={font_size=9.90pt font=("Arial")  }
     style(summary)={font=("Arial") };
     columns hub country year income overhead income_pct profit;
     define hub         / group ;
     define country     / group;
     define year        / group;
     define income      / analysis;
     define overhead    / noprint analysis;
     define profit      / computed   format=dollar12.2  'Profit';
     define income_pct /             format=percent8.2  '% of Income';
     * --- Start: Line by Line Calculations --- *;
        compute profit;
            profit=income.sum - overhead.sum;
        endcompute;
        compute income_pct;
            income_pct=income.sum / income_sum;
        endcompute;
     * --- END:   Line by Line Calculations --- *;
        compute before country;
            income_sum=income.sum;
            income_pct=income.sum/income_sum;
            line ' ';
        endcompute;
        compute after hub;
            income_sum=income.sum;
            income_pct=.;
        endcompute;
     break after hub      / summarize
           style=[font_weight=bold font_size=9.90pt background=white ];
     break after country / summarize  style=[font_size=8.00pt background=pink];
run;

ods pdf close;
```

1. ODS is invoked with the NEW style (which was created in PROC TEMPLATE).

2. The STYLE (HDR) controls the appearance of the column headings,
   The STYLE (SUMMARY) controls the font for the summary rows.

3. This STYLE = option controls the appearance of the totals at the HUB level.
4. This STYLE = option controls the appearance of the totals at the COUNTRY level.

Using ODS to Enhance the Report



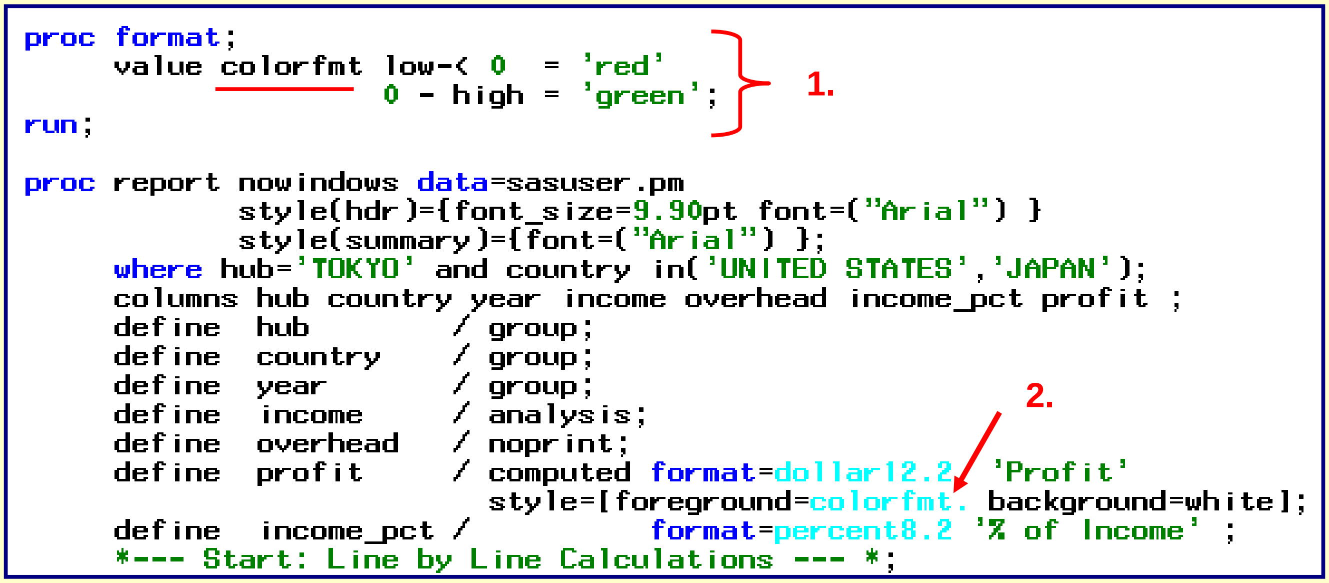
Percent of Income for TOKYO Hub

HUB	COUNTRY	YEAR	INCOME	% of Income	Profit
TOKYO	JAPAN	1993	537.90	6.00%	$-25.53
		1994	1,111.39	12.41%	$250.21
		1995	3,285.39	36.67%	$716.38
		1996	4,023.82	44.92%	$741.36
TOKYO	JAPAN		8,958.50	100.0%	$1,682.41
	UNITED STATES	1993	25,035.00	20.13%	$-721.67
		1994	28,721.50	23.09%	$7,823.44
		1995	33,953.00	27.30%	$9,326.81
		1996	36,662.27	29.48%	$10,102.33
TOKYO	UNITED STATES		124,371.77	100.0%	$26,530.91
TOKYO			133,330.27		$28,213.32

Notice the colors as well as the font size throughout the report. Alter the report so that 'traffic lighting' is applied to the PROFIT column.

Using ODS to Enhance the Report

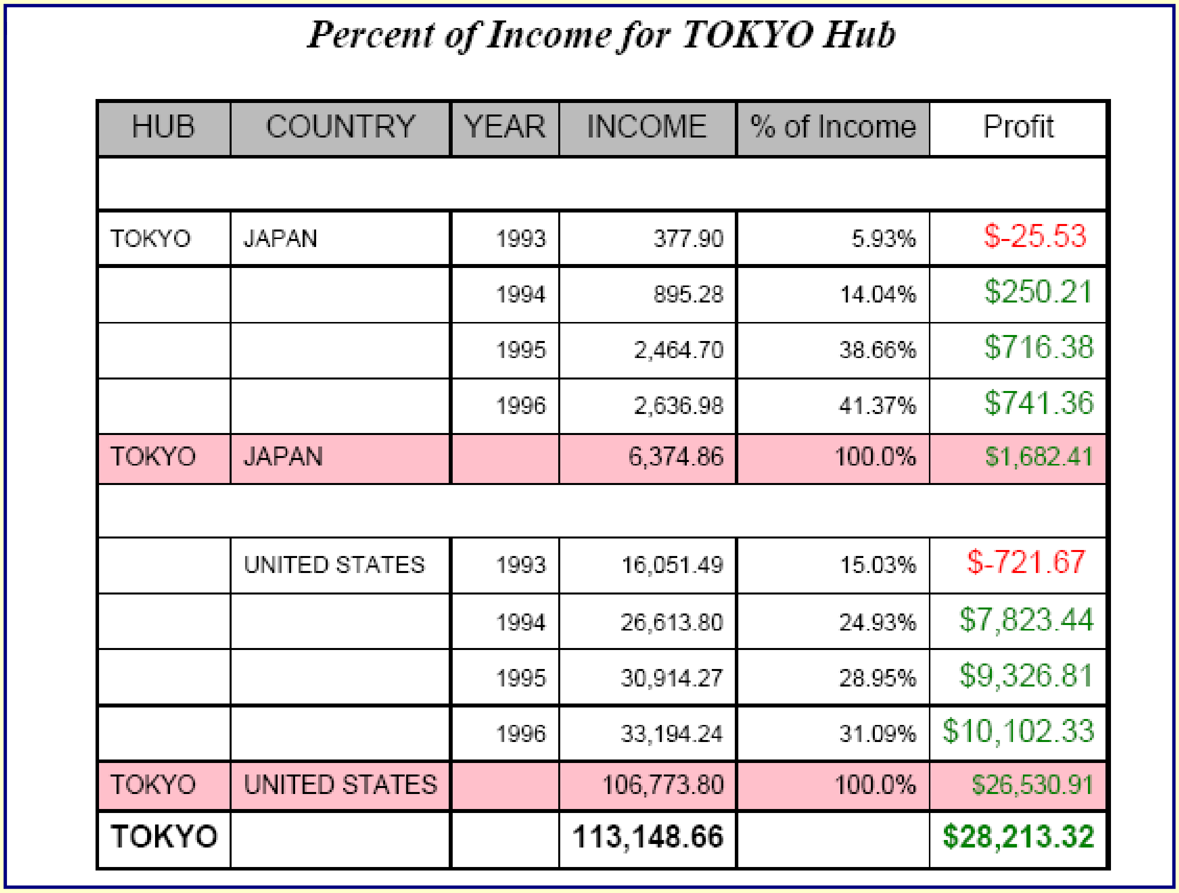
Use the FORMAT procedure to create the traffic lighting format. Demonstrate the WHERE statement to subset the data.



```
proc format;
     value colorfmt low-< 0  = 'red'
                    0 - high = 'green';
run;

proc report nowindows data=sasuser.pm
            style(hdr)={font_size=9.90pt font=("Arial") }
            style(summary)={font=("Arial") };
     where hub='TOKYO' and country in('UNITED STATES','JAPAN');
     columns hub country year income overhead income_pct profit ;
     define  hub        / group;
     define  country    / group;
     define  year       / group;
     define  income     / analysis;
     define  overhead   / noprint;
     define  profit     / computed format=dollar12.2 'Profit'
                          style=[foreground=colorfmt. background=white];
     define  income_pct /          format=percent8.2 '% of Income' ;
     *--- Start: Line by Line Calculations --- *;
```

Note: partial program

Step 1. Create the COLORFMT format.
Step 2. Associate the format with the foreground attribute of PROFIT.



Percent of Income for TOKYO Hub

HUB	COUNTRY	YEAR	INCOME	% of Income	Profit
TOKYO	JAPAN	1993	377.90	5.93%	$-25.53
		1994	895.28	14.04%	$250.21
		1995	2,464.70	38.66%	$716.38
		1996	2,636.98	41.37%	$741.36
TOKYO	JAPAN		6,374.86	100.0%	$1,682.41
	UNITED STATES	1993	16,051.49	15.03%	$-721.67
		1994	26,613.80	24.93%	$7,823.44
		1995	30,914.27	28.95%	$9,326.81
		1996	33,194.24	31.09%	$10,102.33
TOKYO	UNITED STATES		106,773.80	100.0%	$26,530.91
TOKYO			**113,148.66**		**$28,213.32**

Notice the colors as well as the font size of the PROFIT column. Next, the CEO wants to see a similar report where there is a separate column for each YEAR.

Rotating the Report

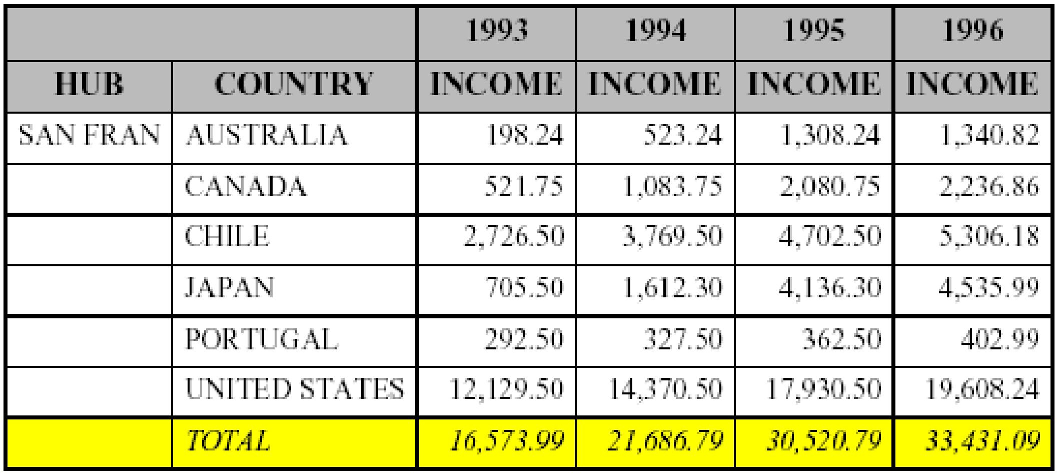
Modify the report to only show INCOME from the San Francisco HUB. Create a column for each year.

```
ods pdf file='c:\ben1.pdf';

proc report nowindows data=sasuser.pm(where=(hub='SAN FRAN'));
     columns (hub country year, (income )) ;
     define hub     / group;
     define country / group;
     define year    / across ' ';
     define income  / analysis sum ;

     rbreak after / dol dul summarize style=[background=yellow] ;

     compute after;
        country = 'TOTAL';
     endcompute;
run;

ods pdf close;
```

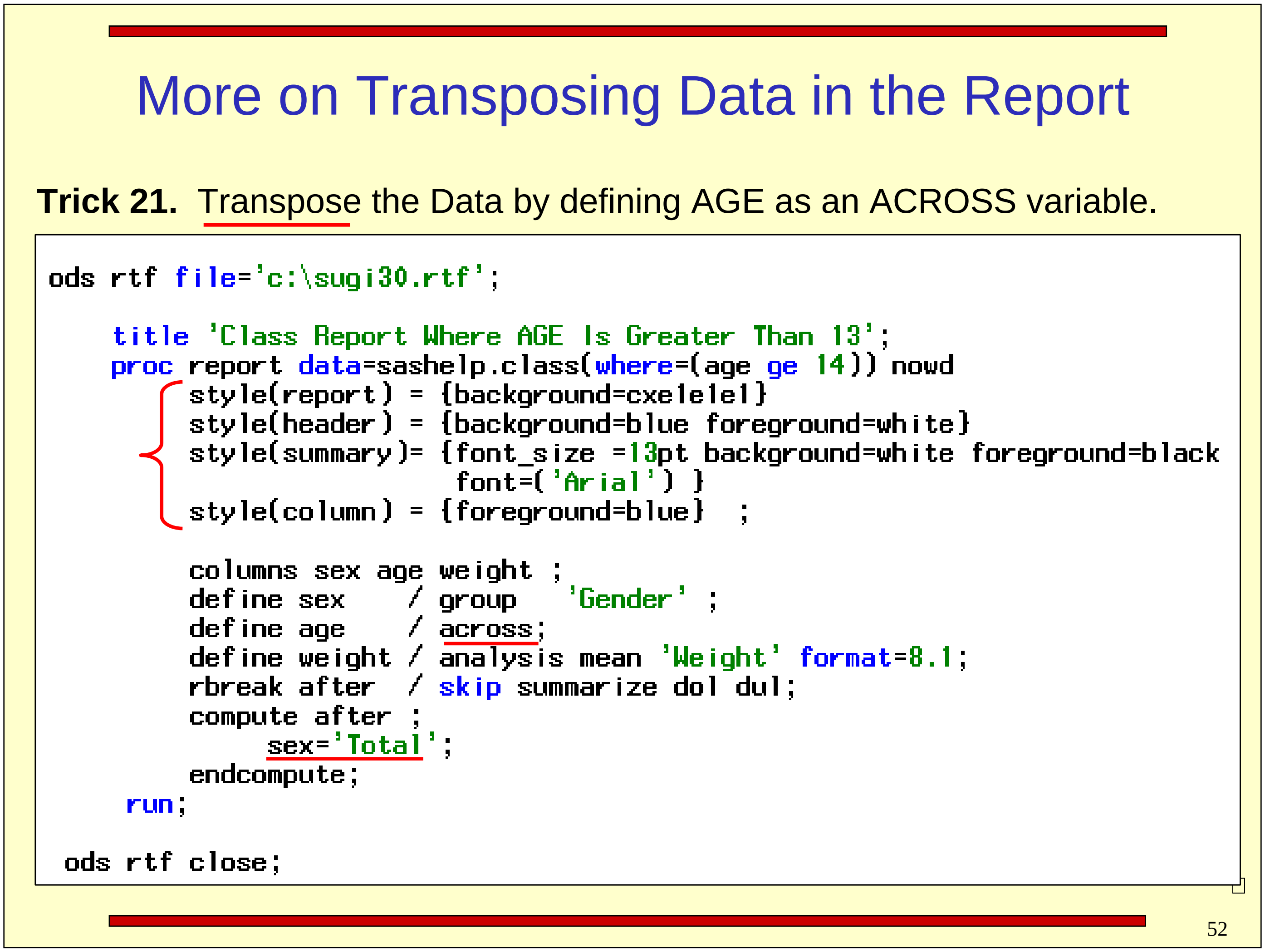
output ➪

Rotating the Report

The final report.



		1993	1994	1995	1996
HUB	**COUNTRY**	**INCOME**	**INCOME**	**INCOME**	**INCOME**
SAN FRAN	AUSTRALIA	198.24	523.24	1,308.24	1,340.82
	CANADA	521.75	1,083.75	2,080.75	2,236.86
	CHILE	2,726.50	3,769.50	4,702.50	5,306.18
	JAPAN	705.50	1,612.30	4,136.30	4,535.99
	PORTUGAL	292.50	327.50	362.50	402.99
	UNITED STATES	12,129.50	14,370.50	17,930.50	19,608.24
	TOTAL	*16,573.99*	*21,686.79*	*30,520.79*	*33,431.09*

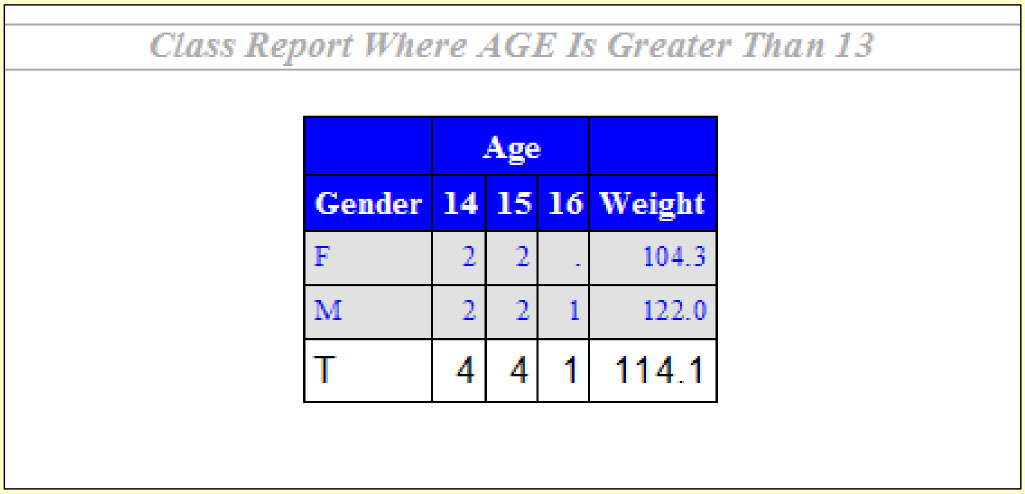


More on Transposing Data in the Report

Trick 21. Transpose the Data by defining AGE as an ACROSS variable.

```
ods rtf file='c:\sugi30.rtf';

    title 'Class Report Where AGE Is Greater Than 13';
    proc report data=sashelp.class(where=(age ge 14)) nowd
         style(report) = {background=cxe1e1e1}
         style(header) = {background=blue foreground=white}
         style(summary)= {font_size =13pt background=white foreground=black
                          font=('Arial') }
         style(column) = {foreground=blue}  ;

         columns sex age weight ;
         define sex    / group   'Gender' ;
         define age    / across;
         define weight / analysis mean 'Weight' format=8.1;
         rbreak after  / skip summarize dol dul;
         compute after ;
              sex='Total';
         endcompute;
     run;

 ods rtf close;
```

Using ODS to Enhance the Report

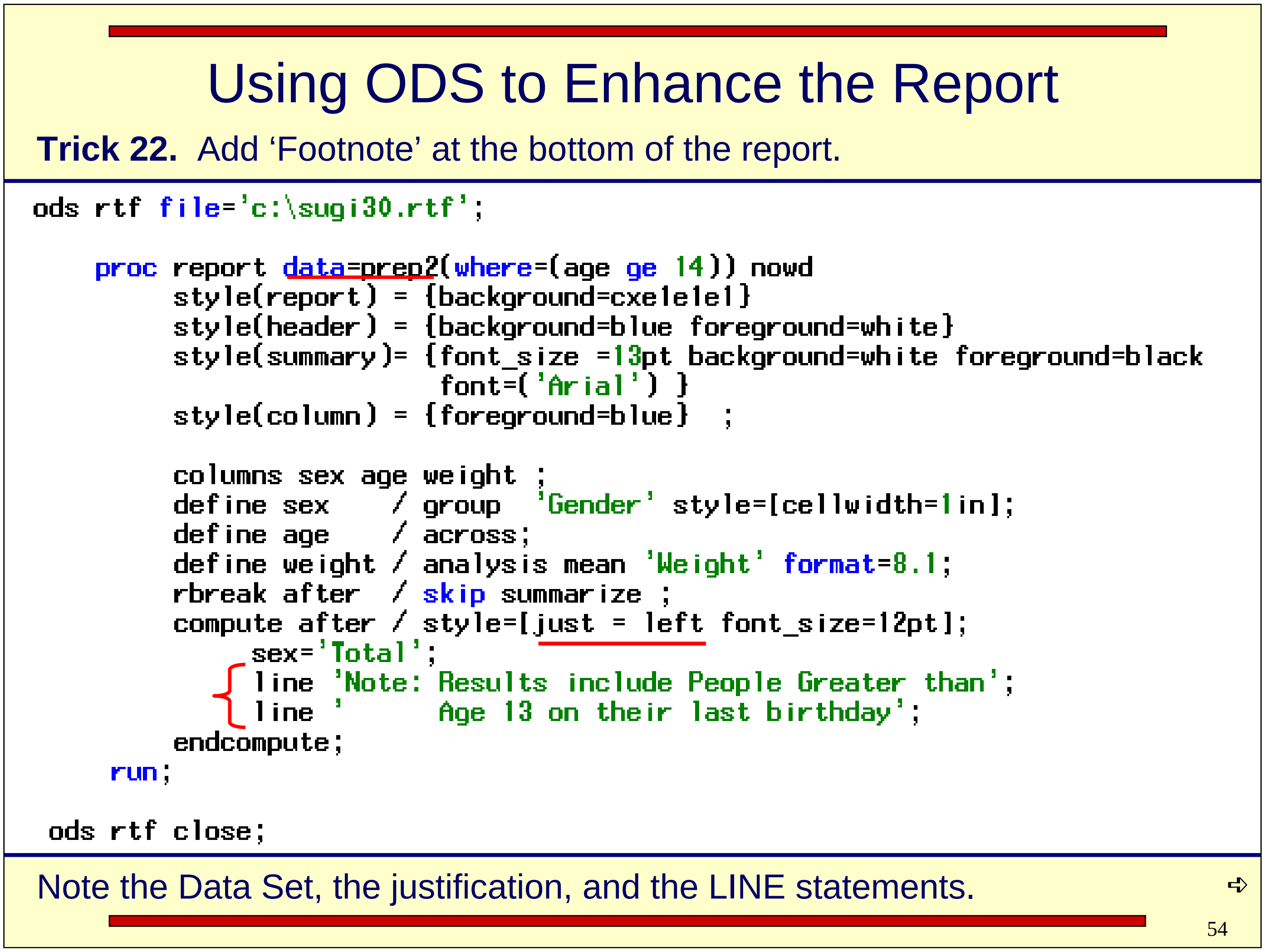


Class Report Where AGE Is Greater Than 13

	Age			
Gender	14	15	16	Weight
F	2	2	.	104.3
M	2	2	1	122.0
T	4	4	1	114.1

Why does a **‘**T’ appear in stead of the word ‘Total’ in the last row of the GENDER column?

Next, Let’s fix this as well as enhance this report.



Using ODS to Enhance the Report

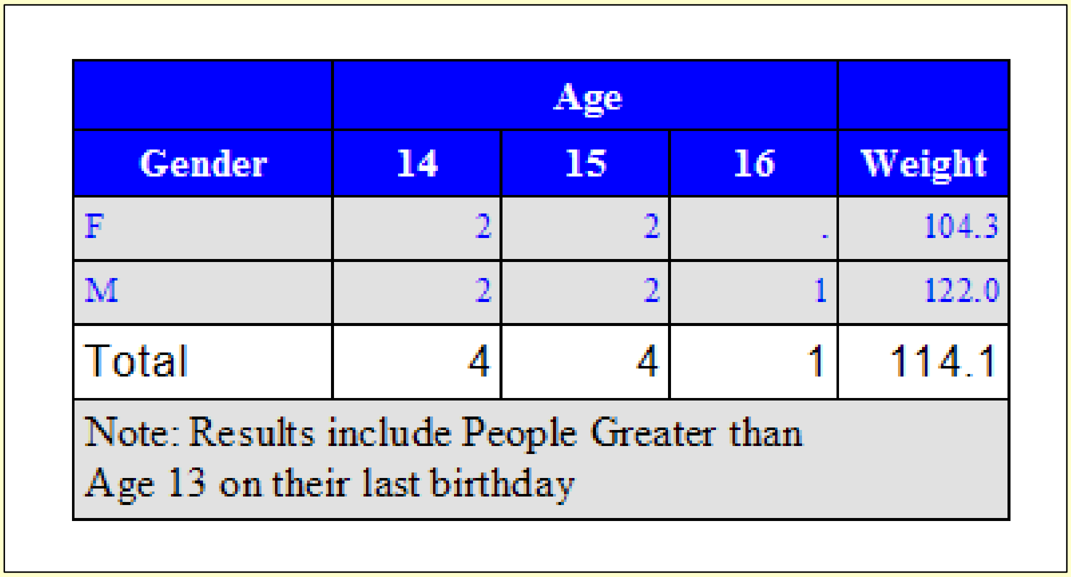
Trick 22. Add ‘Footnote’ at the bottom of the report.

```
ods rtf file='c:\sugi30.rtf';

    proc report data=prep2(where=(age ge 14)) nowd
         style(report) = {background=cxe1e1e1}
         style(header) = {background=blue foreground=white}
         style(summary)= {font_size =13pt background=white foreground=black
                          font=('Arial') }
         style(column) = {foreground=blue}  ;

         columns sex age weight ;
         define sex    / group  'Gender' style=[cellwidth=1in];
         define age    / across;
         define weight / analysis mean 'Weight' format=8.1;
         rbreak after  / skip summarize ;
         compute after / style=[just = left font_size=12pt];
              sex='Total';
              line 'Note: Results include People Greater than';
              line '      Age 13 on their last birthday';
         endcompute;
     run;

 ods rtf close;
```

Note the Data Set, the justification, and the LINE statements.

54

Using ODS to Enhance the Report



	Age			Weight
Gender	14	15	16	Weight
F	2	2	.	104.3
M	2	2	1	122.0
Total	4	4	1	114.1
Note: Results include People Greater than Age 13 on their last birthday				

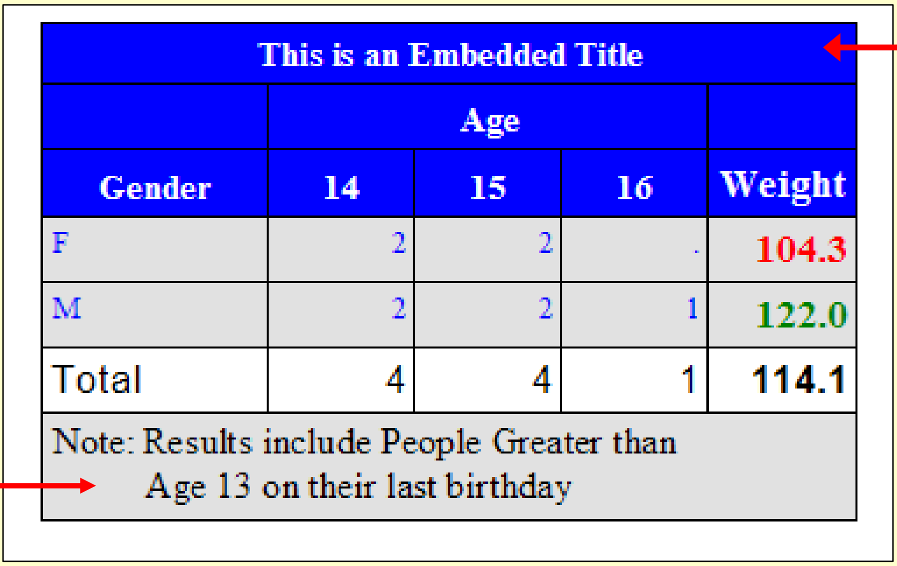
Not quite what was wanted. Notice the second line of the ‘footnote’ does not indent.

Next, let’s fix this as well as enhance this report.

□

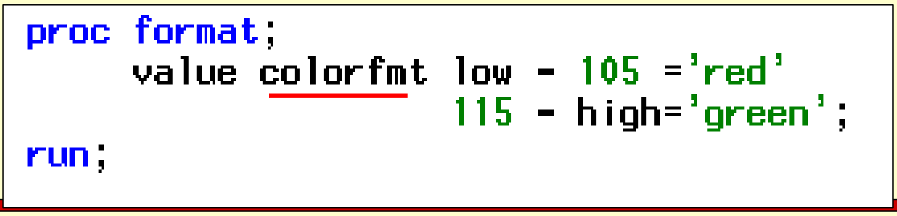
Using ODS to Enhance the Report

Trick 23. Add a format, indent the ‘footnote’, and ‘embed’ a title.

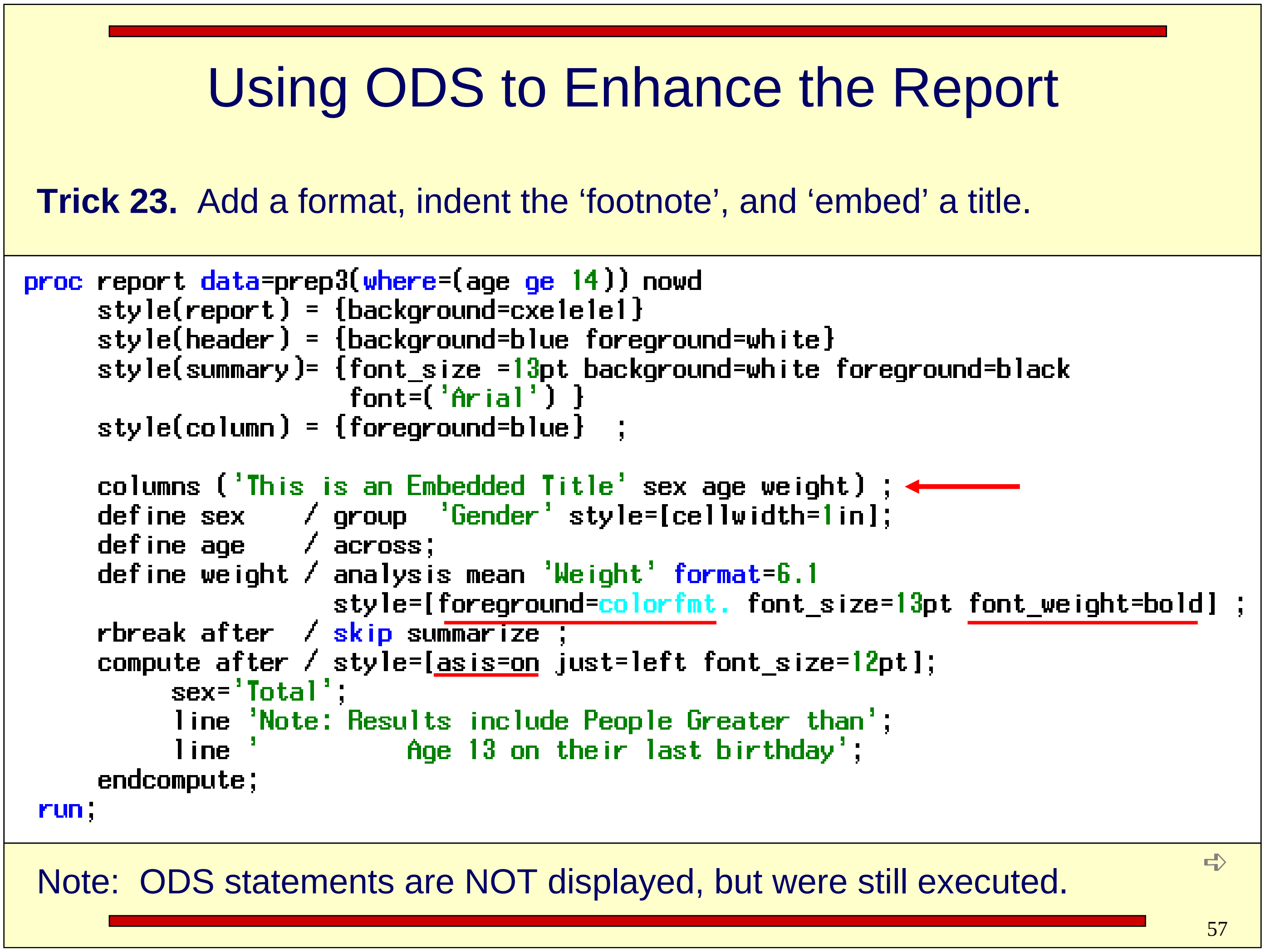


This is an Embedded Title				
	Age			
Gender	14	15	16	Weight
F	2	2	.	104.3
M	2	2	1	122.0
Total	4	4	1	114.1

Note: Results include People Greater than
      Age 13 on their last birthday



```
proc format;
     value colorfmt low - 105 ='red'
                    115 - high='green';
run;
```



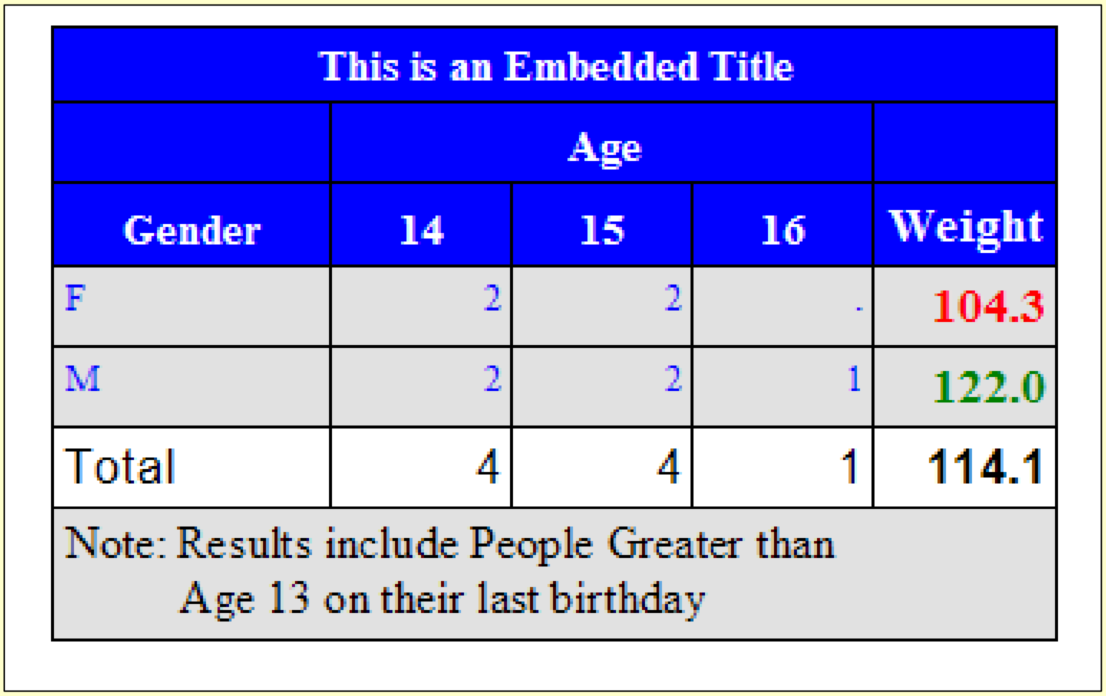
Using ODS to Enhance the Report

Trick 23. Add a format, indent the ‘footnote’, and ‘embed’ a title.

```
proc report data=prep3(where=(age ge 14)) nowd
     style(report) = {background=cxe1e1e1}
     style(header) = {background=blue foreground=white}
     style(summary)= {font_size =13pt background=white foreground=black
                      font=('Arial') }
     style(column) = {foreground=blue}  ;

     columns ('This is an Embedded Title' sex age weight) ;
     define sex    / group  'Gender' style=[cellwidth=1in];
     define age    / across;
     define weight / analysis mean 'Weight' format=6.1
                     style=[foreground=colorfmt. font_size=13pt font_weight=bold] ;
     rbreak after  / skip summarize ;
     compute after / style=[asis=on just=left font_size=12pt];
          sex='Total';
          line 'Note: Results include People Greater than';
          line '          Age 13 on their last birthday';
     endcompute;
 run;
```

Note: ODS statements are NOT displayed, but were still executed.

Using ODS to Enhance the Report



This is an Embedded Title

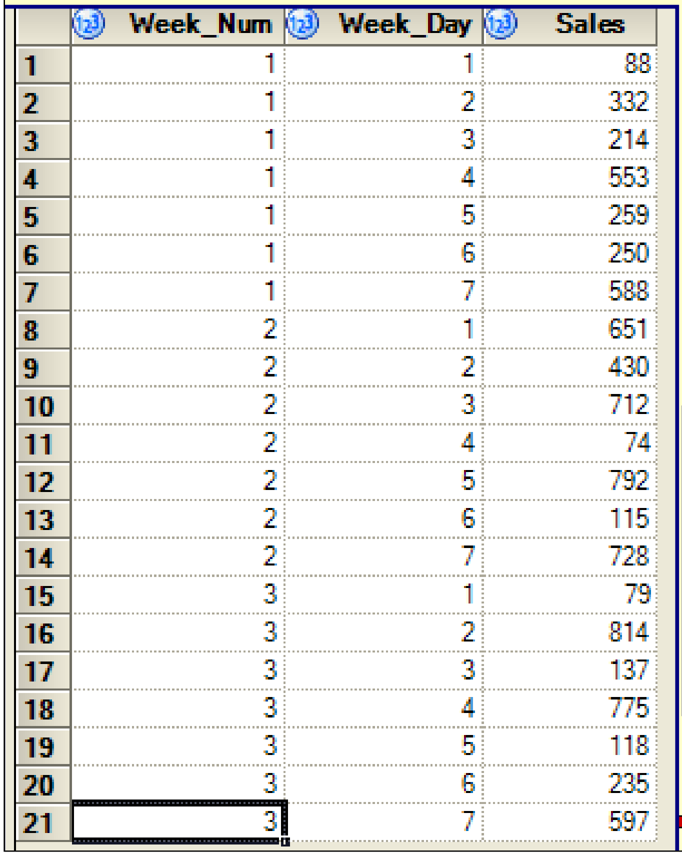
	Age			
Gender	14	15	16	Weight
F	2	2	.	104.3
M	2	2	1	122.0
Total	4	4	1	114.1

Note: Results include People Greater than
Age 13 on their last birthday

More on Using an ACROSS Column

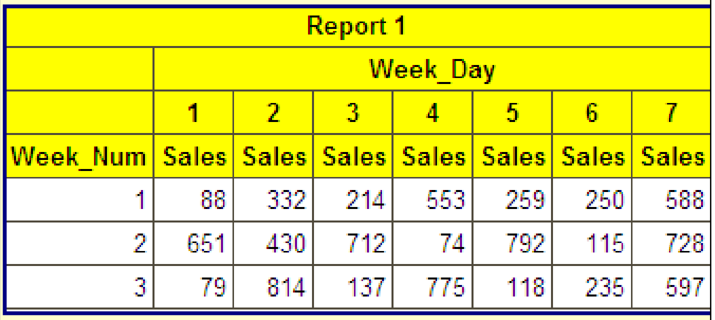
Trick 24: Transpose the data.

The following tasks will use the ‘Sales’ dataset shown below.



	Week_Num	Week_Day	Sales
1	1	1	88
2	1	2	332
3	1	3	214
4	1	4	553
5	1	5	259
6	1	6	250
7	1	7	588
8	2	1	651
9	2	2	430
10	2	3	712
11	2	4	74
12	2	5	792
13	2	6	115
14	2	7	728
15	3	1	79
16	3	2	814
17	3	3	137
18	3	4	775
19	3	5	118
20	3	6	235
21	3	7	597

This data set has 1 row per week day for 3 weeks. Management wants a report with a **column for each day of the week.** There needs to be an eighth column on the right that displays the **Total.**

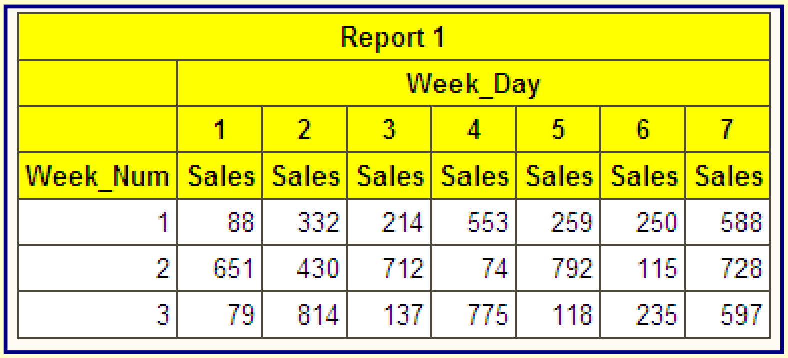


Report 1

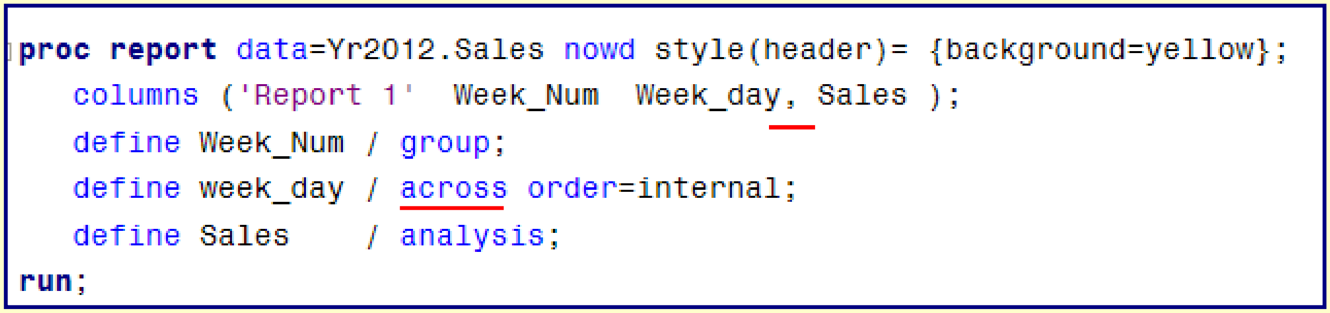
	Week_Day						
	1	2	3	4	5	6	7
Week_Num	Sales	Sales	Sales	Sales	Sales	Sales	Sales
1	88	332	214	553	259	250	588
2	651	430	712	74	792	115	728
3	79	814	137	775	118	235	597

Using an ACROSS Column

Analyze the report, then the program that was used to create it.

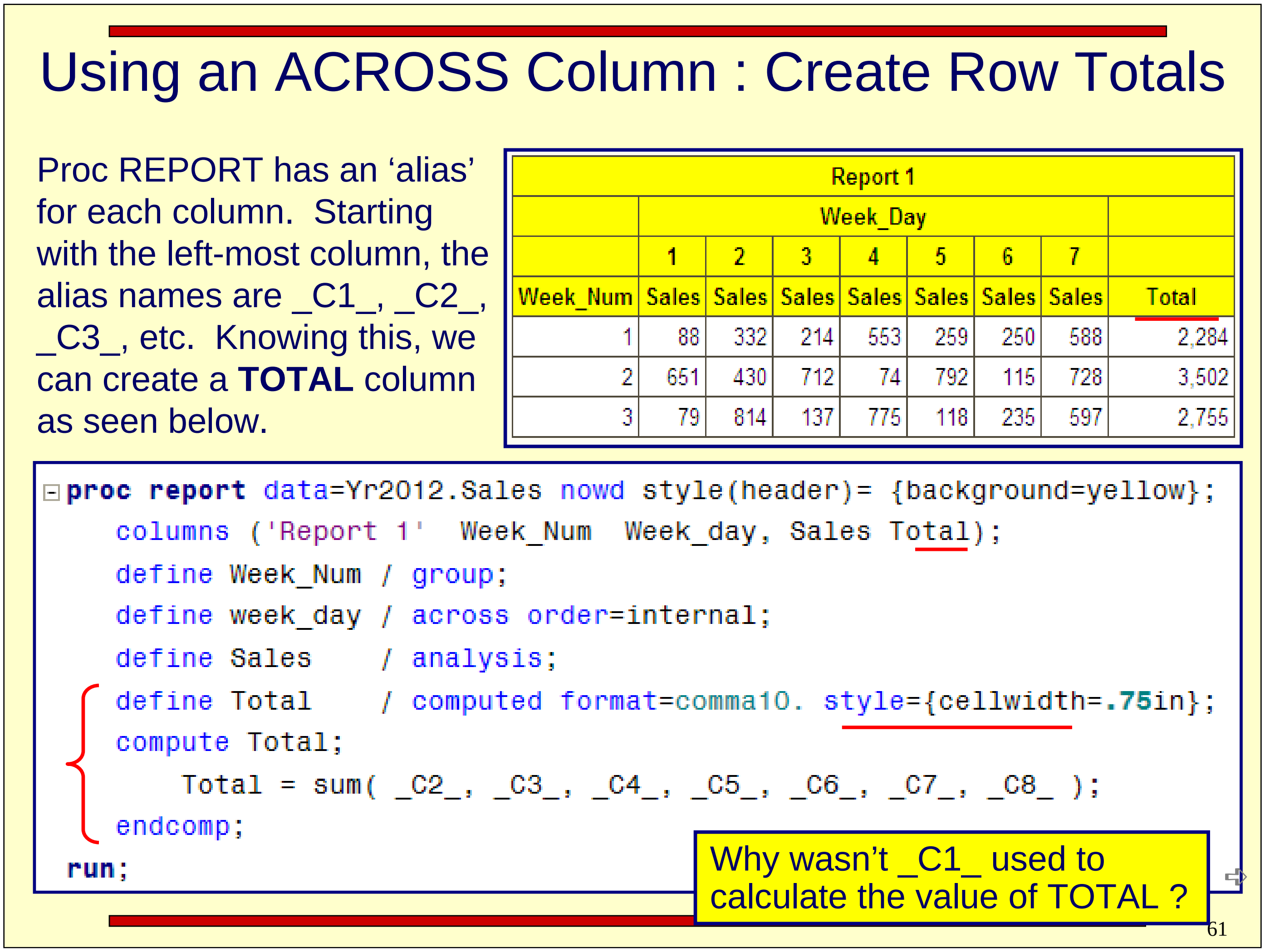


Report 1							
	Week_Day						
	1	2	3	4	5	6	7
Week_Num	Sales	Sales	Sales	Sales	Sales	Sales	Sales
1	88	332	214	553	259	250	588
2	651	430	712	74	792	115	728
3	79	814	137	775	118	235	597



```
proc report data=Yr2012.Sales nowd style(header)= {background=yellow};
   columns ('Report 1'  Week_Num  Week_day, Sales );
   define Week_Num / group;
   define week_day / across order=internal;
   define Sales    / analysis;
run;
```

Notice the **Columns** statement… especially the use of parentheses to create the ‘embedded’ title. Also notice the comma after **Week_Day.** Notice the **Across** variable. What is needed next is the Total column.



Using an ACROSS Column : Create Row Totals

Proc REPORT has an ‘alias’ for each column. Starting with the left-most column, the alias names are _C1_, _C2_, _C3_, etc. Knowing this, we can create a **TOTAL** column as seen below.

Report 1								
	Week_Day							
	1	2	3	4	5	6	7	
Week_Num	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Total
1	88	332	214	553	259	250	588	2,284
2	651	430	712	74	792	115	728	3,502
3	79	814	137	775	118	235	597	2,755

```
proc report data=Yr2012.Sales nowd style(header)= {background=yellow};
    columns ('Report 1'  Week_Num  Week_day, Sales Total);
    define Week_Num / group;
    define week_day / across order=internal;
    define Sales    / analysis;
    define Total    / computed format=comma10. style={cellwidth=.75in};
    compute Total;
        Total = sum( _C2_, _C3_, _C4_, _C5_, _C6_, _C7_, _C8_ );
    endcomp;
run;
```

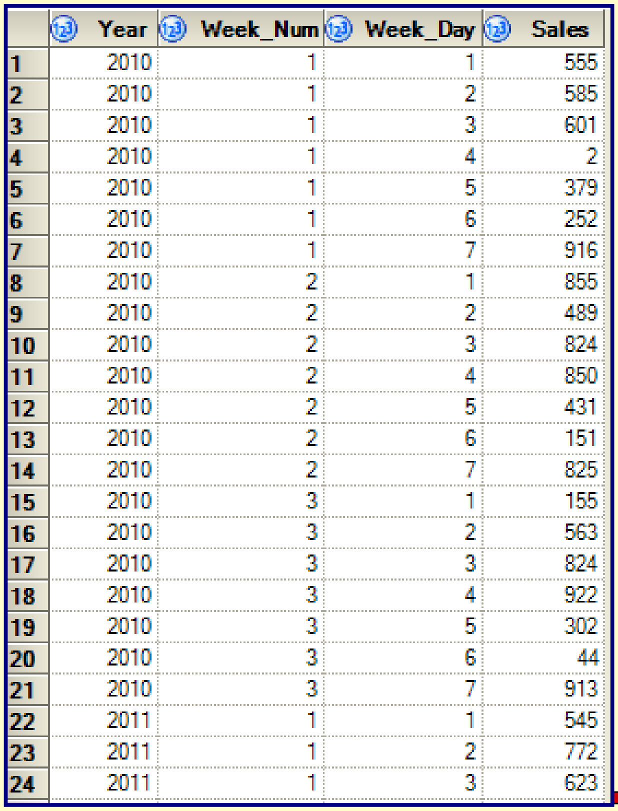
Why wasn’t _C1_ used to calculate the value of TOTAL ?

61

Because Week_Day is defined as as ACROSS variable, we have to use _C2_ to refer to first Week_Day column, _C3_ to refer to the second Week_Day column, etc. _C1_ is the alias for the Week_Num column.

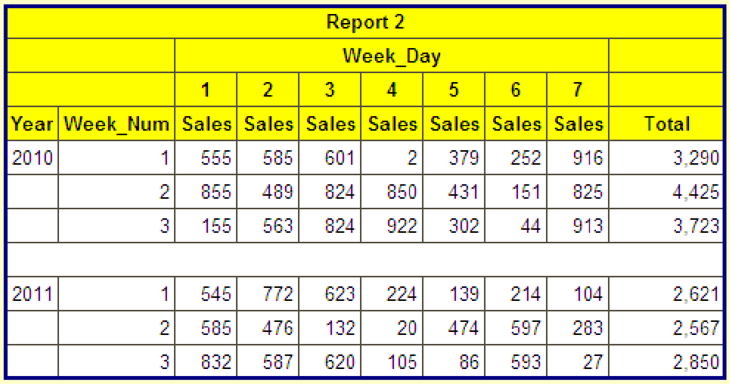
Using an ACROSS Column : Create Row Totals

The ‘Sales’ dataset has been modified to include a column for **Year.**

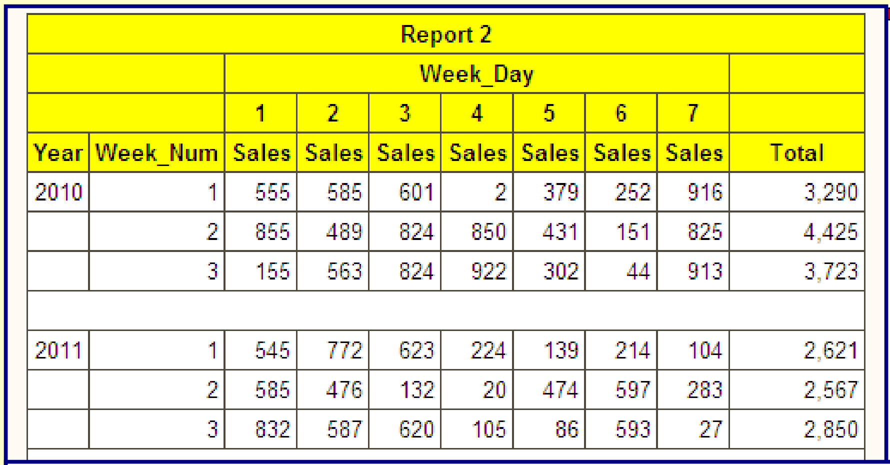


	Year	Week_Num	Week_Day	Sales
1	2010	1	1	555
2	2010	1	2	585
3	2010	1	3	601
4	2010	1	4	2
5	2010	1	5	379
6	2010	1	6	252
7	2010	1	7	916
8	2010	2	1	855
9	2010	2	2	489
10	2010	2	3	824
11	2010	2	4	850
12	2010	2	5	431
13	2010	2	6	151
14	2010	2	7	825
15	2010	3	1	155
16	2010	3	2	563
17	2010	3	3	824
18	2010	3	4	922
19	2010	3	5	302
20	2010	3	6	44
21	2010	3	7	913
22	2011	1	1	545
23	2011	1	2	772
24	2011	1	3	623

This data will be used for the next several examples. Notice the YEAR column in the report.



Report 2									
		Week_Day							
		1	2	3	4	5	6	7	
Year	Week_Num	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Total
2010	1	555	585	601	2	379	252	916	3,290
	2	855	489	824	850	431	151	825	4,425
	3	155	563	824	922	302	44	913	3,723
2011	1	545	772	623	224	139	214	104	2,621
	2	585	476	132	20	474	597	283	2,567
	3	832	587	620	105	86	593	27	2,850

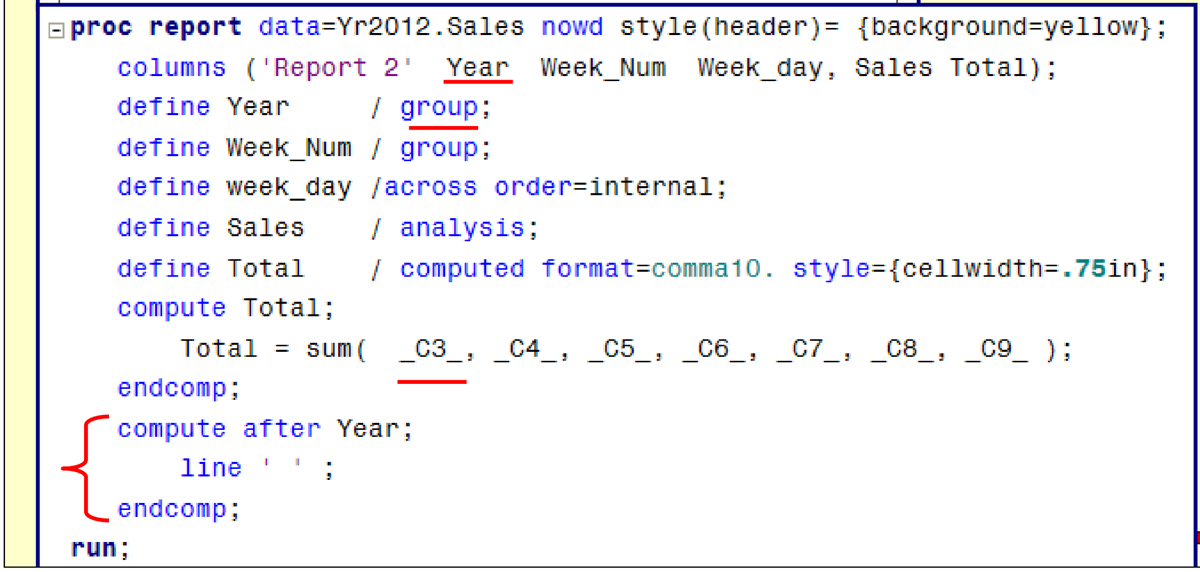


Report 2									
		Week_Day							
		1	2	3	4	5	6	7	
Year	Week_Num	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Total
2010	1	555	585	601	2	379	252	916	3,290
	2	855	489	824	850	431	151	825	4,425
	3	155	563	824	922	302	44	913	3,723
2011	1	545	772	623	224	139	214	104	2,621
	2	585	476	132	20	474	597	283	2,567
	3	832	587	620	105	86	593	27	2,850

Add **YEAR** and Define it as a GROUP variable.

Notice the computation for Total.   Why does it start with _C3_ ?

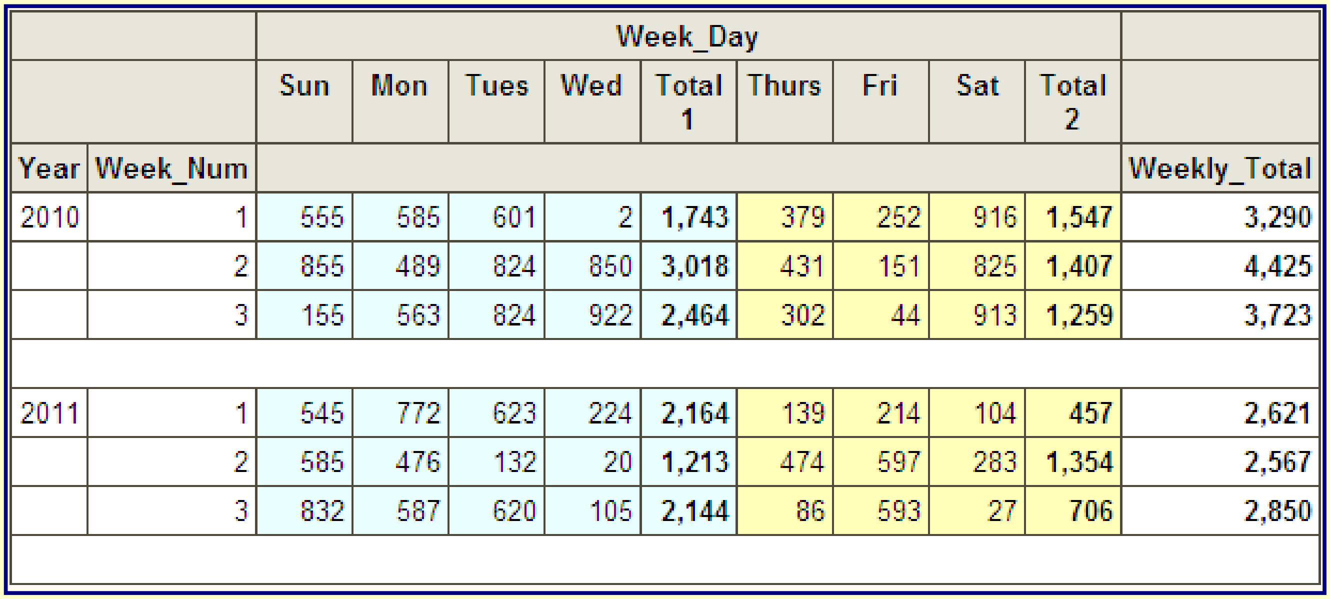
The **Compute After** block generates the blank line after each Year.



```
proc report data=Yr2012.Sales nowd style(header)= {background=yellow};
   columns ('Report 2'  Year  Week_Num  Week_day, Sales Total);
   define Year      / group;
   define Week_Num / group;
   define week_day /across order=internal;
   define Sales     / analysis;
   define Total     / computed format=comma10. style={cellwidth=.75in};
   compute Total;
       Total = sum(  _C3_, _C4_, _C5_, _C6_, _C7_, _C8_, _C9_ );
   endcomp;
   compute after Year;
       line ' ' ;
   endcomp;
run;
```

Using an ACROSS Column : Create Row Totals

Next, management has decided that they want **TWO sub** totals for the Week, a new one after Wednesday to total Sunday through Wednesday, as well as one after Saturday to total Thursday through Saturday. They still want to to total all seven days.

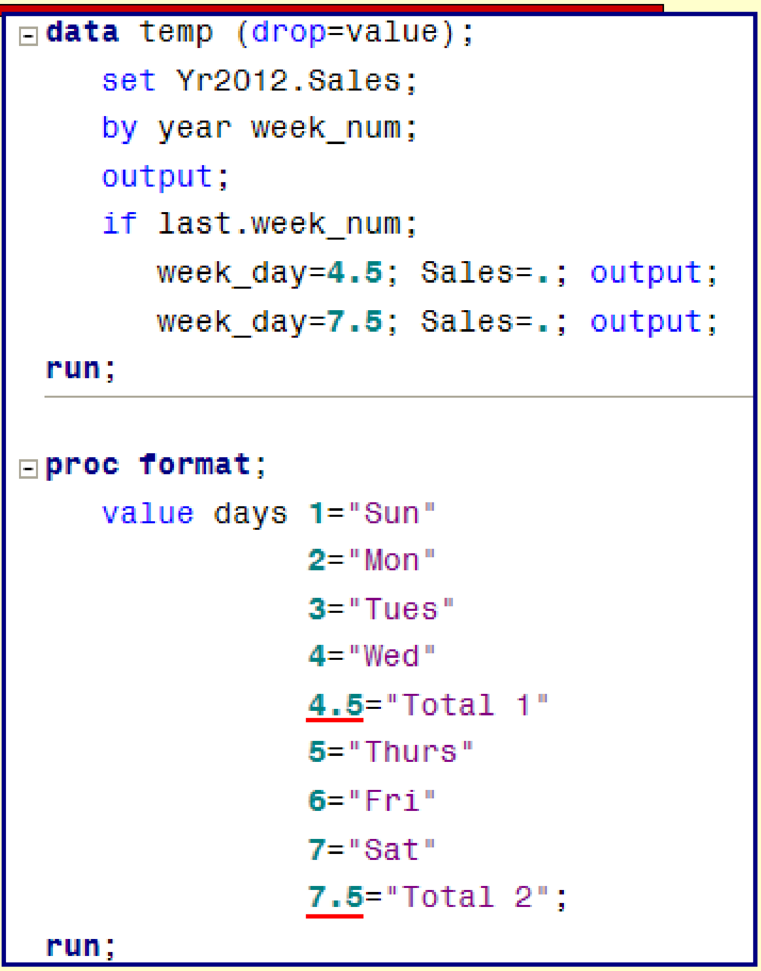


		Week_Day									
		Sun	Mon	Tues	Wed	Total 1	Thurs	Fri	Sat	Total 2	
Year	Week_Num										Weekly_Total
2010	1	555	585	601	2	**1,743**	379	252	916	**1,547**	**3,290**
	2	855	489	824	850	**3,018**	431	151	825	**1,407**	**4,425**
	3	155	563	824	922	**2,464**	302	44	913	**1,259**	**3,723**
2011	1	545	772	623	224	**2,164**	139	214	104	**457**	**2,621**
	2	585	476	132	20	**1,213**	474	597	283	**1,354**	**2,567**
	3	832	587	620	105	**2,144**	86	593	27	**706**	**2,850**

To do this, the data has to be manipulated to insert 2 new columns (‘Week days’).

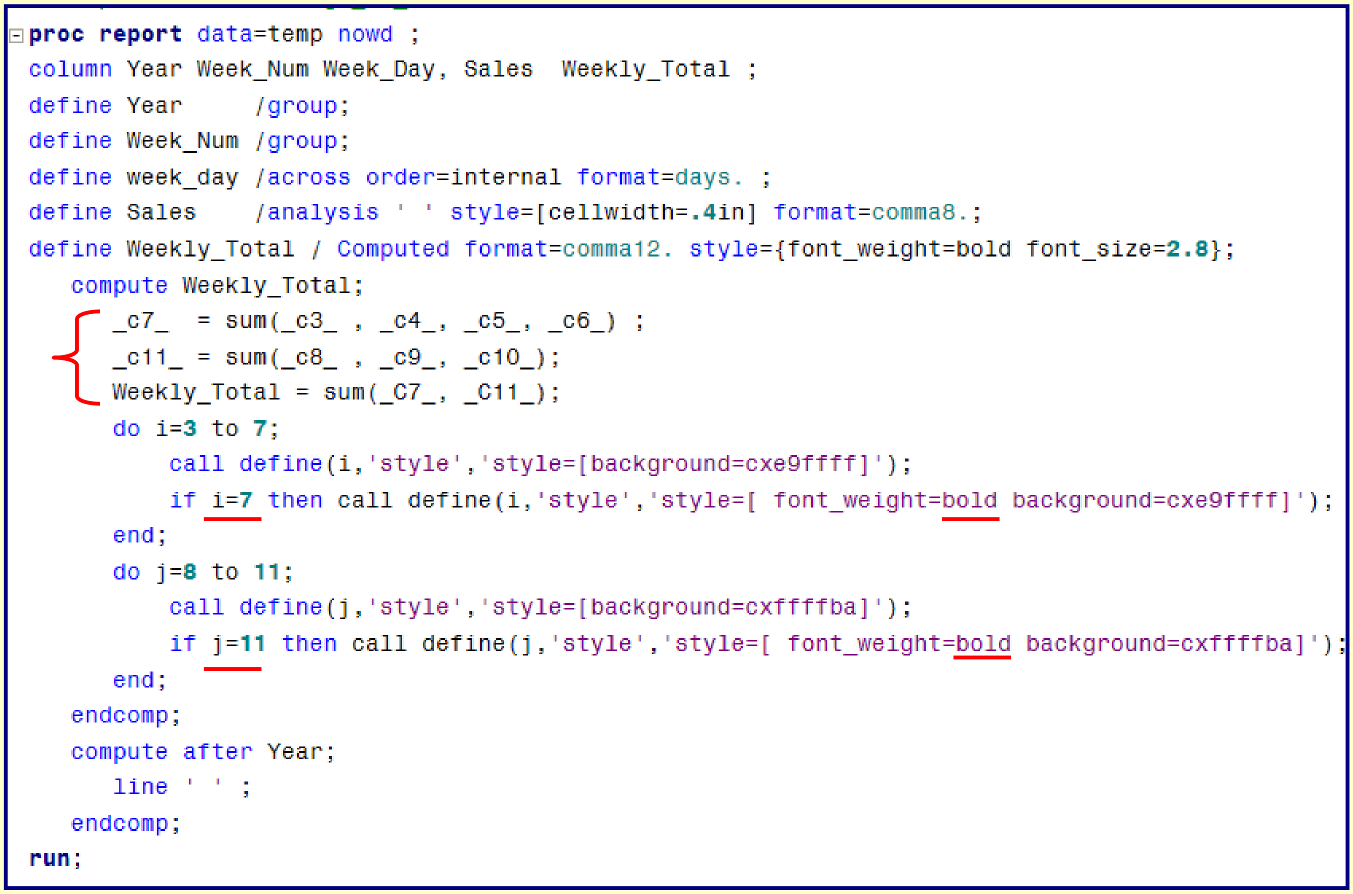
First, write a DATA step that will ‘add’ 2 values to Week_Day.

Then write a Proc FORMAT step to create a format for the week days.



```
data temp (drop=value);
   set Yr2012.Sales;
   by year week_num;
   output;
   if last.week_num;
      week_day=4.5; Sales=.; output;
      week_day=7.5; Sales=.; output;
run;

proc format;
   value days 1="Sun"
              2="Mon"
              3="Tues"
              4="Wed"
              4.5="Total 1"
              5="Thurs"
              6="Fri"
              7="Sat"
              7.5="Total 2";
run;
```

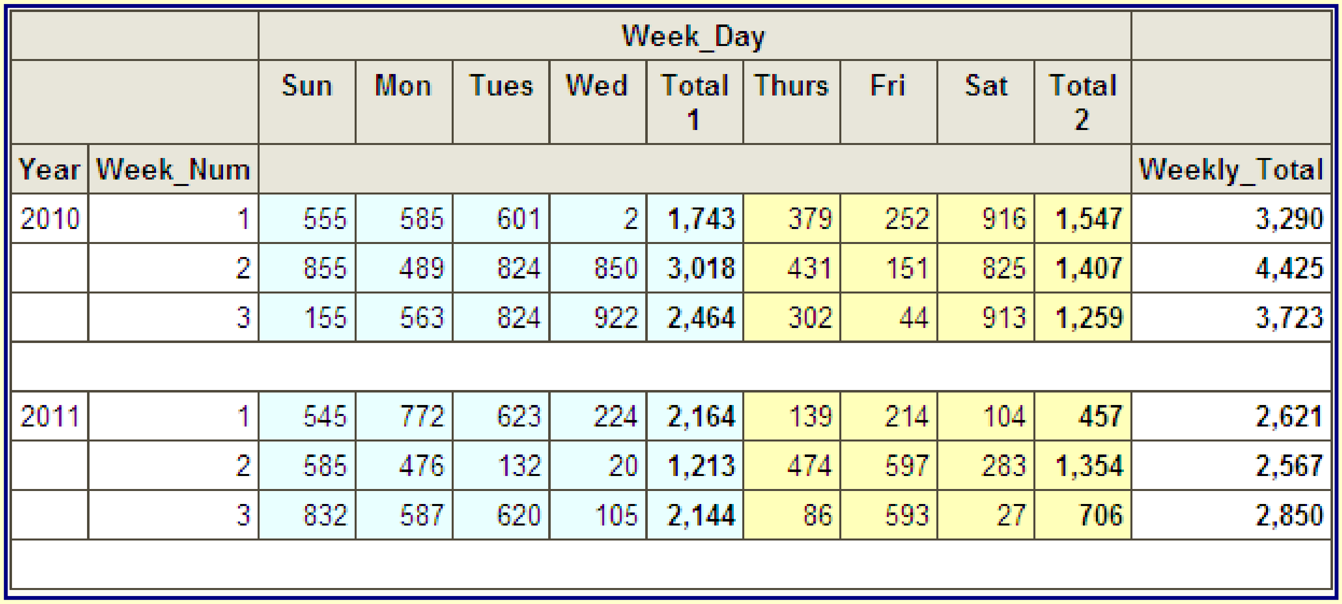
Next, write a Proc REPORT step to generate the report. Create the weekly Sub Totals and put them in Columns _**C7**_ and _**C11**_ **.**



```
proc report data=temp nowd ;
column Year Week_Num Week_Day, Sales  Weekly_Total ;
define Year     /group;
define Week_Num /group;
define week_day /across order=internal format=days. ;
define Sales    /analysis ' ' style=[cellwidth=.4in] format=comma8.;
define Weekly_Total / Computed format=comma12. style={font_weight=bold font_size=2.8};
   compute Weekly_Total;
      _c7_  = sum(_c3_ , _c4_, _c5_, _c6_) ;
      _c11_ = sum(_c8_ , _c9_, _c10_);
      Weekly_Total = sum(_C7_, _C11_);
      do i=3 to 7;
          call define(i,'style','style=[background=cxe9ffff]');
          if i=7 then call define(i,'style','style=[ font_weight=bold background=cxe9ffff]');
      end;
      do j=8 to 11;
          call define(j,'style','style=[background=cxffffba]');
          if j=11 then call define(j,'style','style=[ font_weight=bold background=cxffffba]');
      end;
   endcomp;
   compute after Year;
      line ' ' ;
   endcomp;
run;
```

Using an ACROSS Column : Create **Row** Totals

The final report looks like this…

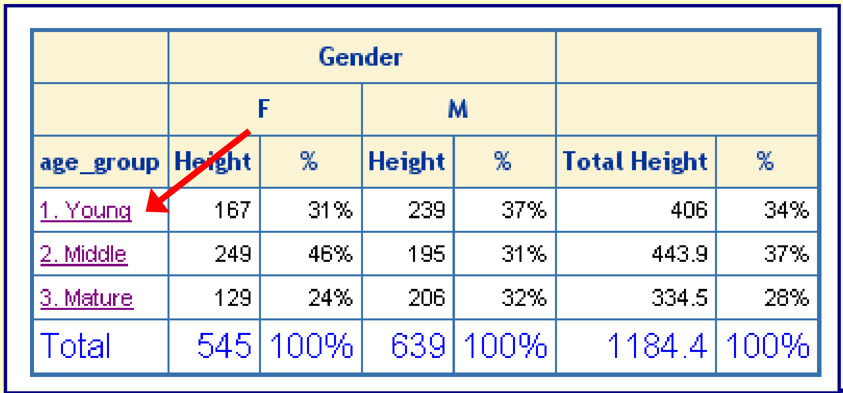


		Week_Day								
		Sun	Mon	Tues	Wed	Total 1	Thurs	Fri	Sat	Total 2	
Year	Week_Num										Weekly_Total
2010	1	555	585	601	2	**1,743**	379	252	916	**1,547**	**3,290**
	2	855	489	824	850	**3,018**	431	151	825	**1,407**	**4,425**
	3	155	563	824	922	**2,464**	302	44	913	**1,259**	**3,723**
2011	1	545	772	623	224	**2,164**	139	214	104	**457**	**2,621**
	2	585	476	132	20	**1,213**	474	597	283	**1,354**	**2,567**
	3	832	587	620	105	**2,144**	86	593	27	**706**	**2,850**

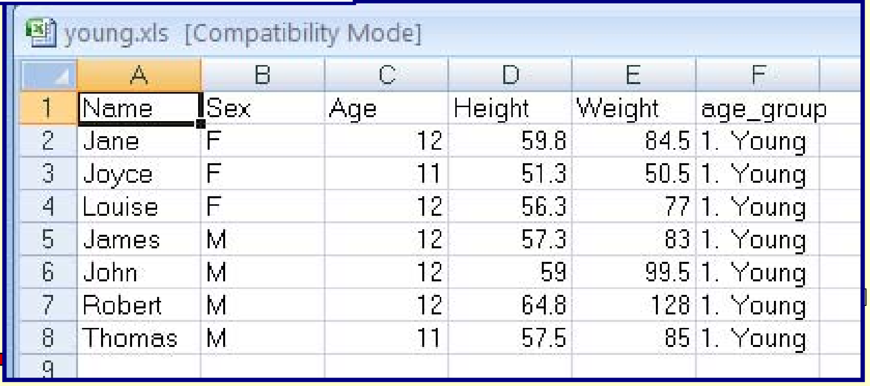
link ➪

Linking

Create a report from the CLASS dataset that allows you to **link** to another file. Here, PROC REPORT creates the report on the left, and when **‘1. Young’** is selected from the **AGE_GROUP** column, the spreadsheet opens showing the detail data.



	Gender					
	F		M			
age_group	Height	%	Height	%	Total Height	%
1. Young	167	31%	239	37%	406	34%
2. Middle	249	46%	195	31%	443.9	37%
3. Mature	129	24%	206	32%	334.5	28%
Total	545	100%	639	100%	1184.4	100%



	A	B	C	D	E	F
1	Name	Sex	Age	Height	Weight	age_group
2	Jane	F	12	59.8	84.5	1. Young
3	Joyce	F	11	51.3	50.5	1. Young
4	Louise	F	12	56.3	77	1. Young
5	James	M	12	57.3	83	1. Young
6	John	M	12	59	99.5	1. Young
7	Robert	M	12	64.8	128	1. Young
8	Thomas	M	11	57.5	85	1. Young
9						

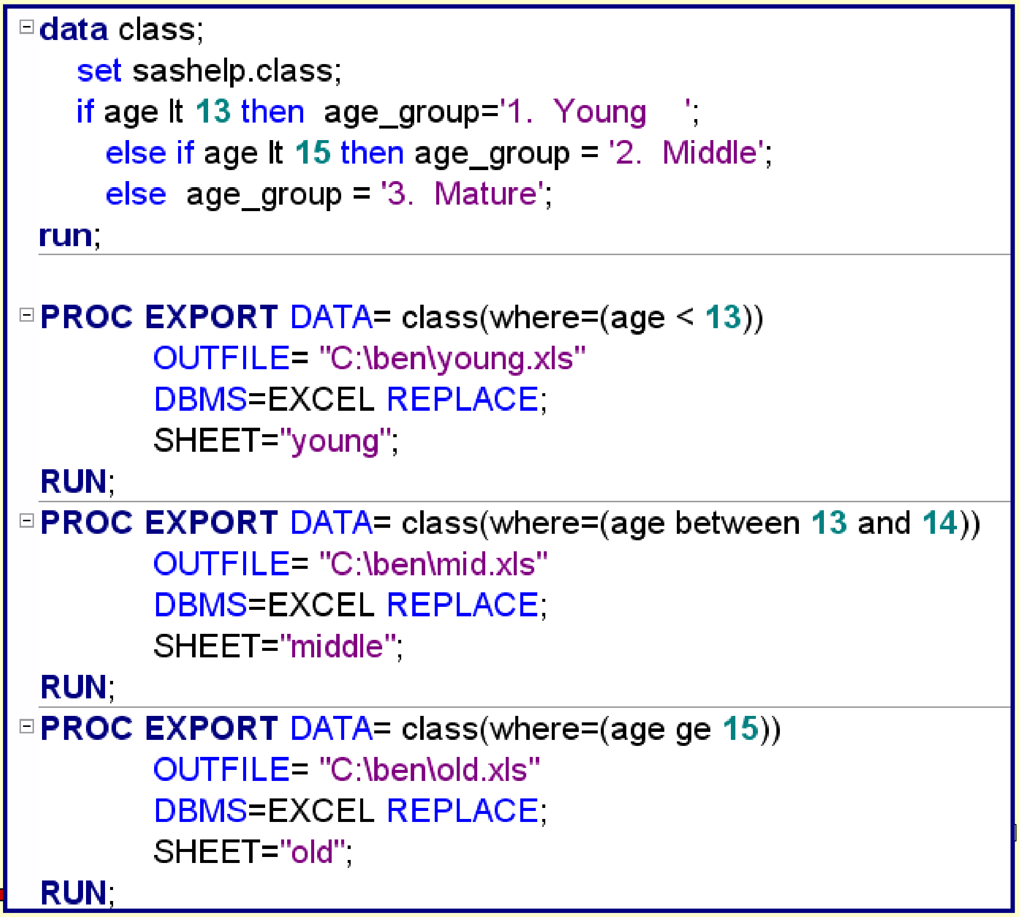
Linking

The first step is to create three age groups based on the value of age.

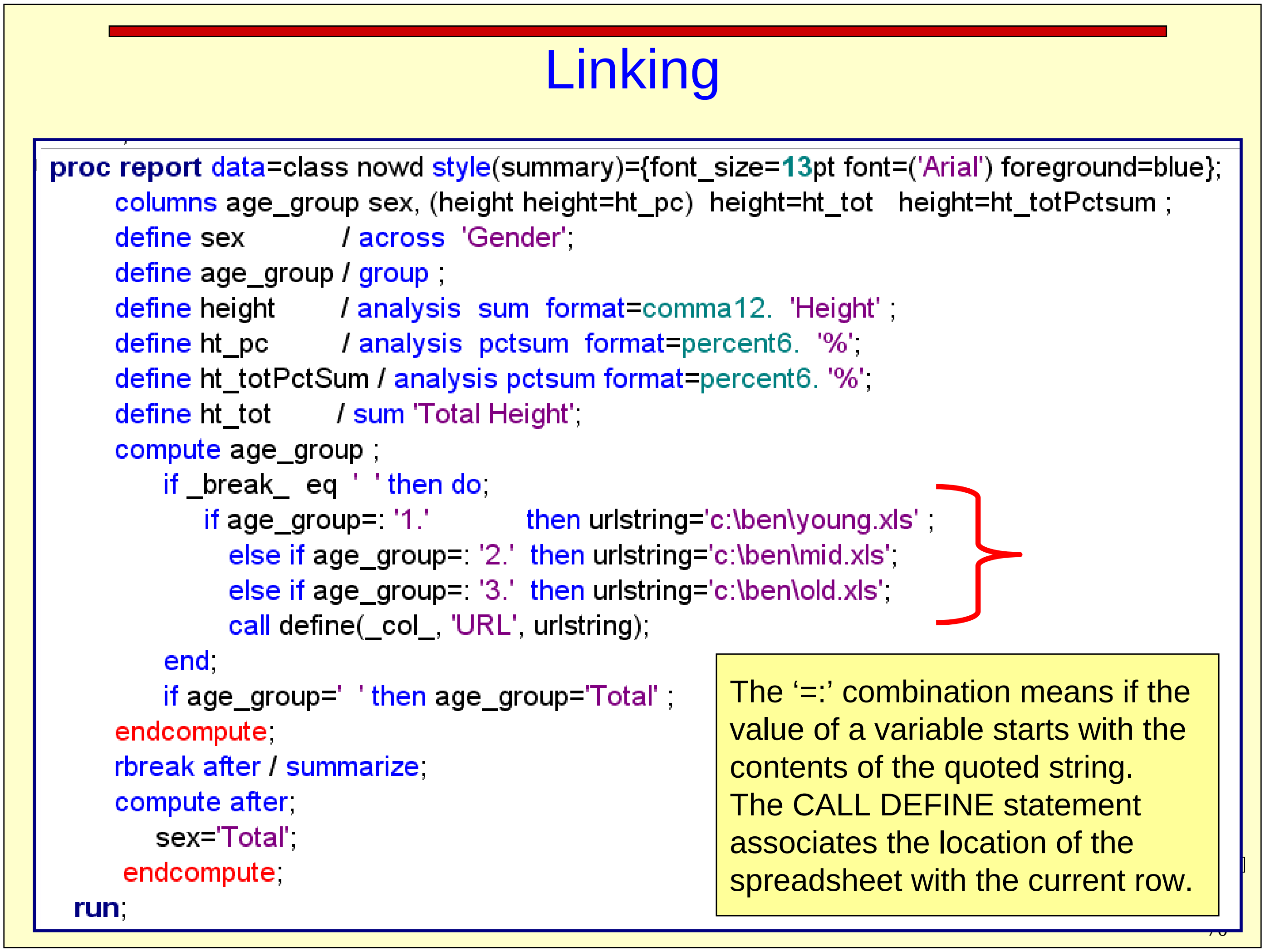
Next, create the spreadsheets that contain detail data by writing a series of PROC EXPORT steps.

Notice the use of the **WHERE=** option.

Notice the locations of the spreadsheets.



```
data class;
   set sashelp.class;
   if age lt 13 then  age_group='1.  Young     ';
     else if age lt 15 then age_group = '2.  Middle';
     else  age_group = '3.  Mature';
run;

PROC EXPORT DATA= class(where=(age < 13))
            OUTFILE= "C:\ben\young.xls"
            DBMS=EXCEL REPLACE;
            SHEET="young";
RUN;
PROC EXPORT DATA= class(where=(age between 13 and 14))
            OUTFILE= "C:\ben\mid.xls"
            DBMS=EXCEL REPLACE;
            SHEET="middle";
RUN;
PROC EXPORT DATA= class(where=(age ge 15))
            OUTFILE= "C:\ben\old.xls"
            DBMS=EXCEL REPLACE;
            SHEET="old";
RUN;
```

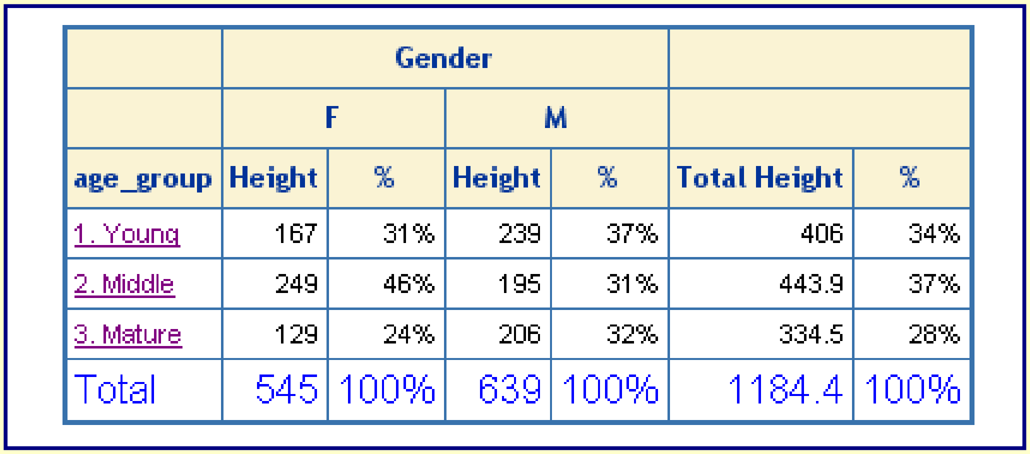


Linking

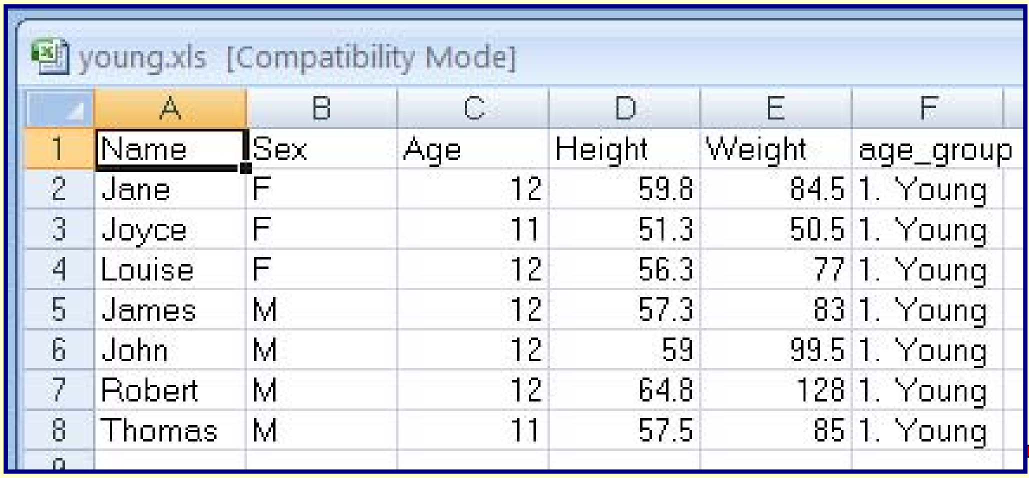
```
proc report data=class nowd style(summary)={font_size=13pt font=('Arial') foreground=blue};
      columns age_group sex, (height height=ht_pc)  height=ht_tot   height=ht_totPctsum ;
      define sex           / across  'Gender';
      define age_group / group ;
      define height        / analysis  sum  format=comma12.  'Height' ;
      define ht_pc         / analysis  pctsum  format=percent6.  '%';
      define ht_totPctSum / analysis pctsum format=percent6. '%';
      define ht_tot        / sum 'Total Height';
      compute age_group ;
           if _break_  eq  '  ' then do;
               if age_group=: '1.'          then urlstring='c:\ben\young.xls' ;
                 else if age_group=: '2.'  then urlstring='c:\ben\mid.xls';
                 else if age_group=: '3.'  then urlstring='c:\ben\old.xls';
                 call define(_col_, 'URL', urlstring);
           end;
           if age_group='  ' then age_group='Total' ;
      endcompute;
      rbreak after / summarize;
      compute after;
           sex='Total';
       endcompute;
  run;
```

The ‘=:’ combination means if the value of a variable starts with the contents of the quoted string. The CALL DEFINE statement associates the location of the spreadsheet with the current row.

Linking



	Gender					
	F		M			
age_group	Height	%	Height	%	Total Height	%
1. Young	167	31%	239	37%	406	34%
2. Middle	249	46%	195	31%	443.9	37%
3. Mature	129	24%	206	32%	334.5	28%
Total	545	100%	639	100%	1184.4	100%



	A	B	C	D	E	F
1	Name	Sex	Age	Height	Weight	age_group
2	Jane	F	12	59.8	84.5	1. Young
3	Joyce	F	11	51.3	50.5	1. Young
4	Louise	F	12	56.3	77	1. Young
5	James	M	12	57.3	83	1. Young
6	John	M	12	59	99.5	1. Young
7	Robert	M	12	64.8	128	1. Young
8	Thomas	M	11	57.5	85	1. Young

The End