
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

1

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.

2

Acknowledgements

Tech Support at SAS Institute

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

3

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

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.

□

5

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 labels
- LS or LINESIZE** - specifies the width of the lines in the report
- PS or PAGESIZE** - specifies the number of lines in the report

example □

6

PROC REPORT

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

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

Name	Sex	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

7

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

8

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

9

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

10

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 □

11

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 □

12

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 →

13

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** % →

14

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

Gender	Name	Height	Weight	% of 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%
=====		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%
=====		63.9	1089.5	100.0%
=====				

pgm →

15

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 mean '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

Calculating Percentages

Calculating Percentages with Proc Report

Gender	Name	Height	Weight	% of 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? →

17

Suppressing Columns

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

Gender	Name	Height	% of 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 →

18

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

Suppressing Columns

Gender	Name	Height	% of 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.

→

20

Calculating Percentages

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

Gender	Name	Height	Weight	% of 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%
=====		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%
=====		63.9	109.0	100.0%
=====				

pgm →

21

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 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 →

22

Calculating Percentages

Gender	Name	Height	Weight	% of 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. →

23

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	<u>Average Weight</u>	90.11 ←
=====		=====
F	<u>Median Weight</u>	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	<u>Average Weight</u>	108.95
=====		=====
M	<u>Median Weight</u>	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 →

24

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
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 →

28

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 →

29

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. →

30

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	Janes	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 →

31

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;
```

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...

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

more →

33

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=** .

→

34

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}

→

35

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 →

36

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 →

37

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

→

38

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	36,682.27	29.48%	\$10,102.33
TOKYO	UNITED STATES		124,371.77	100.0%	\$26,530.91
TOKYO			133,330.27		\$28,213.32

→

39

```
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 →

40

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...



41

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...



42

```

ods pdf file='c:\test1.pdf' style=new; ← 1.
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')} ← 2.
  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 ]; ← 3.
  break after country / summarize style=[font_size=8.00pt background=pink]; ← 4.
run;
ods pdf close;

```

43

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	3,265.39	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%	\$9,328.81
		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.

44

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.

→

45

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.

→

46

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.pn(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 →

47

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

→

48

More on Transposing Data in the Report

Task 11. 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=cx1e1e1}
  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;
```

49

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.

⇒

50

Using ODS to Enhance the Report

Task 12. 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=cx1e1e1}
  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. →

51

Using ODS to Enhance the Report

	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				

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.

□

52

Using ODS to Enhance the Report

Task 13. 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;
```

⇒

53

Using ODS to Enhance the Report

Task 13. Add a format, indent the 'footnote', and 'embed' a title.

```
proc report data=prep3(where=(age ge 14)) nowd
  style(report) = {background=cx4e4e4e}
  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.

54

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				

⇒

55

More on Using an ACROSS Column

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

⇒

56

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. ⇒

57

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** ? ⇒

58

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

⇒

59

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;
    
```

⇒

60

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 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

⇒

61

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;

```

62

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

The End



About the Speaker

Speaker Ben Cochran

Office Location The Bedford Group
3224 Bedford Ave.
Raleigh, NC 27607

Telephone (919) 831.1191
Fax (919) 831.1191

E-Mail bencochran@nc.rr.com

65

Using an ACROSS Column : Create Column Totals

The final report should look like this...

Gender	Age in Years								
	11			12			13		
	Total Weight	Total Kids	Avg Weight	Total Weight	Total Kids	Avg Weight	Total Weight	Total Kids	Avg Weight
F	50.5	1	50.5	161.5	2	80.8	84.0	1	84.0
M	85.0	1	85.0	310.5	3	103.5	84.0	1	84.0
Totals	135.5	2	67.8	472.0	5	94.4	168.0	2	84.0

Needs Reworking.
Drop Total Weight.

Notice that the **Total Weight** and **Total Kids** columns are **summed** while the **Avg Weight** columned is **Averaged**.

⇒

66

Using an ACROSS Column : Create Column Totals

VIEWTABLE: Sashelp.Class

	NAME	SEX	AGE	HEIGHT	WEIGHT
1	Alice	F	13	56.5	84
2	Becka	F	44	65.3	98
3	Gail	F	14	64.3	90
4	Karen	F	12	56.3	77
5	Kathy	F	12	59.8	84.5
6	Mary	F	15	66.5	112
7	Sandy	F	11	51.3	50.5
8	Sharon	F	15	62.5	112.5
9	Tammy	F	14	62.8	102.5
10	Alfred	M	14	69	112.5
11	Duke	M	33	63.5	102.5
12	Guido	M	15	67	133
13	James	M	12	57.3	83
14	Jeffrey	M	13	62.5	84
15	John	M	12	59	99.5
16	Philip	M	16	72	150
17	Robert	M	12	64.8	128
18	Thomas	M	11	57.5	85
19	William	M	15	66.5	112

The next series of reports use the CLASS dataset.

AGE is used as the **ACROSS** variable and

SEX is used as the **GROUP** variable.

⇒

67

Using an ACROSS Column : Create Column Totals

```
data class; length sex $9;
set sashelp.class;
a=1; n=1;
run;
```

Step 1. Create a temporary data set that has two new variables (**n** and **a**) and make the variable **SEX** have a length of 9.

```
proc report data=class(where=(age lt 14)) nowd split='*';
style(summary) = {background=pink font_size=12pt font=('Arial')};
columns a sex age, (weight n weight=weight_avg);
define a /group noprint;
define sex /group 'Gender' format=$9.;
define age /across 'Age in Years';
define n /analysis sum 'Total* Kids';
define weight /analysis sum 'Total*Weight' format=5.1;
define weight_avg /analysis mean 'Avg * Weight' format=5.1;
break after a / summarize;
compute after a ;
sex = 'Totals';
endcomp;
run;
```

Step 2. Write the PROC REPORT step and filter the dataset where age is less than 14. Notice the ANALYSIS variables and the BREAK statement and COMPUTE block.

68

Using an ACROSS Column : Create Column Totals

The report has a **Totals** Row that contains Column totals.

Gender	Age in Years								
	11			12			13		
	Total Weight	Total Kids	Avg Weight	Total Weight	Total Kids	Avg Weight	Total Weight	Total Kids	Avg Weight
F	50.5	1	50.5	161.5	2	80.8	84.0	1	84.0
M	85.0	1	85.0	310.5	3	103.5	84.0	1	84.0
Totals	135.5	2	67.8	472.0	5	94.4	168.0	2	84.0

Next, a new row will be added that contains column Averages.

Lets say that we want to have only **ONE** column for **Weight**, but have a row for the **SUM** and a row for the **AVERAGE**.

69

Using an ACROSS Column : Create Column Totals

Modify the program as follows:

```
proc report data=class(where=(age lt 14)) nowd split="*"
  style(summary) = {background=pink font_size=12pt font=('Arial')};
  columns a sex age, (weight n weight=weight_avg);
  define a /group noprint;
  define sex /group 'Gender' format=$9.;
  define age /across 'Age in Years';
  define n /analysis sum 'Total* Kids' noprint;
  define weight /analysis sum 'Weight' format=5.1;
  define weight_avg /analysis mean 'Avg *Weight' format=5.1 noprint;
  break after a /summarize;
  compute after a;
  sex = 'Totals';
  line '';
endcomp;
rbreak after / skip summarize style=[background=gold];
compute after;
sex = 'Averages';
_C3_ = _C5_;
_C6_ = _C8_;
_C9_ = _C11_;
endcomp;
run;
```

Notice the **NOPRINT**s and the modified label.

Add the **RBREAK** statement and the **compute block** to generate the **Averages** row.

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

	Age in Years		
	11	12	13
Gender	Weight	Weight	Weight
F	50.5	161.5	84.0
M	85.0	310.5	84.0
Totals	135.5	472.0	168.0
Averages	67.8	94.4	84.0

The End

71

The End

72

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

Conclusion: You can generate very colorful and attractive reports using PROC REPORT and ODS. PROC TEMPATE was not used at all.

```
ods pdf file='c:\test1.pdf' style=new; ← 1.
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')} ← 2.
  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]; ← 3.
  break after country / summarize style=[font_size=8.00pt background=pink]; ← 4.
run;
ods pdf close;
```

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.

75

9.8 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 SASUSER.PM data set is used in the following examples:

	HUB	COUNTRY	TYPE	YEAR	INCOME	OVERHEAD
1	LONDON	AUSTRALIA	MD11	1993	288.24	230.59
2	LONDON	AUSTRALIA	MD11	1994	523.24	418.59
3	LONDON	AUSTRALIA	MD11	1995	1,500.24	1,200.19
4	LONDON	AUSTRALIA	MD11	1996	1,660.57	1,328.45
5	LONDON	AUSTRALIA	DC10	1993	499.24	394.40
6	LONDON	AUSTRALIA	DC10	1994	523.24	413.36
7	LONDON	AUSTRALIA	DC10	1995	804.24	635.35
8	LONDON	AUSTRALIA	DC10	1996	874.62	690.95
9	SAN FRAN	AUSTRALIA	A300	1993	198.24	152.64
10	SAN FRAN	AUSTRALIA	A300	1994	523.24	402.89
11	SAN FRAN	AUSTRALIA	A300	1995	1,308.24	1,007.34
12	SAN FRAN	AUSTRALIA	A300	1996	1,340.82	1,032.43
13	NEW YORK	AUSTRALIA	MD11	1993	529.00	322.69
14	NEW YORK	AUSTRALIA	MD11	1994	1,170.00	713.70

76

PROC REPORT

The typical form of the REPORT procedure:

```
PROC REPORT data= SAS-data-set options ;  
  COLUMN variable_1 .... variable_n;  
  DEFINE variable_1;  
  DEFINE variable_2;  
  ...  
  DEFINE variable_n;  
  
  COMPUTE blocks  
  BREAK ... ;  
  RBREAK ... ;  
  
RUN;
```

- ◆ **COLUMN** 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 (blank lines) in the report.

example □

77

PROC REPORT

You can use the REPORT procedure in windowing or a non-windowing 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 titles
- LS or LINESIZE - specifies the width of the lines in the report
- PS or PAGESIZE - specifies the number of lines in the report

example □

78

PROC REPORT

Generate a basic report using the REPORT procedure:

```

title 'SASUSER.PM data set';
proc report nowindows data=sasuser.pm;
  columns hub country type year income overhead;
  define hub      / group ;
  define country  / group;
  define type     / group;
  define year     / group;
  define income   / analysis;
  define overhead / analysis;

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

SASUSER.PM data set					
HUB	COUNTRY	TYPE	YEAR	INCOME	OVERHEAD
FRANKFRT	AUSTRALIA	A300	1993	640.71	512.57
			1994	251.29	201.03
			1995	1,106.29	885.00
	CANADA	747	1996	1,318.61	1,054.49
			1993	2,630.20	2,261.97
			1994	3,595.20	3,091.87
CHILE	747	1995	4,584.20	3,942.41	
		1996	4,703.86	4,045.32	
		1993	234.00	145.08	
JAPAN	DC10	DC10	1994	456.00	282.72
			1995	560.00	347.20
			1996	600.00	372.00
	A300	A300	1993	264.00	163.68
			1994	390.00	241.80
			1995	516.00	319.92
MD11	A300	A300	1996	560.34	347.41
			1993	1,798.03	1,401.12
			1994	2,535.79	1,961.85
	DC10	DC10	1995	5,944.79	4,461.30
			1996	6,819.89	5,119.98
			1993	636.60	483.82
MD11	MD11	1994	269.40	204.74	
		1995	999.40	759.54	
		1996	1,073.97	816.21	
A300	A300	1993	1,315.15	942.79	
		1994	1,436.85	993.08	
		1995	3,056.85	2,108.59	
DC10	DC10	1996	3,418.48	2,358.04	
		1993	346.90	218.55	
		1994	290.10	182.76	
A300	A300	1995	1,051.10	662.19	
		1996	1,131.69	712.96	

Any enhancements?

more □

PROC REPORT

Enhance the report by adding a blank line after the column names and calculating profit for each row.

```

title 'Profit Report';
proc report nowindows data=sasuser.pm headline headskip;
  columns hub country type year income overhead profit;
  define hub / group;
  define country / group;
  define type / group;
  define year / group;
  define income / analysis;
  define overhead / analysis;
  define profit / computed format=dollar12.2;
  compute profit;
    profit=income.sum - overhead.sum;
  endcompute;
run;

```

Notice the following:

- ◆ **HEADLINE** and **HEADSKIP** options,
- ◆ the **COMPUTE** block,
- ◆ the new variable (**PROFIT**),
- ◆ the syntax within the compute block : the association of a statistic with the variable name (for line by line calculations).

output

81

PROC REPORT

Profit Report						
HUB	COUNTRY	TYPE	YEAR	INCOME	OVERHEAD	profit
FRANKFRT	AUSTRALIA	A300	1993	640.71	512.57	\$128.14
			1994	251.29	201.03	\$50.26
			1995	1,106.29	885.03	\$221.26
	CANADA	747	1996	1,318.61	1,054.89	\$263.72
			1993	2,630.20	2,261.97	\$368.23
			1994	3,595.20	3,091.87	\$503.33
	CHILE	747	1995	4,584.20	3,942.41	\$641.79
			1996	4,703.86	4,045.32	\$658.54
			1993	234.00	145.08	\$88.92
	JAPAN	747	1994	456.00	282.72	\$173.28
			1995	560.00	347.20	\$212.80
			1996	600.00	372.00	\$228.00
	PORTUGAL	DC10	1993	264.00	163.68	\$100.32
			1994	390.00	241.80	\$148.20
			1995	516.00	319.92	\$196.08
	PORTUGAL	A300	1996	560.34	347.41	\$212.93
			1993	1,738.03	1,401.12	\$336.91
			1994	2,535.79	1,961.85	\$573.94
	PORTUGAL	MD11	1995	5,944.79	4,461.30	\$1,483.49
			1996	6,819.89	5,119.98	\$1,699.91
			1993	636.60	483.82	\$152.78
	PORTUGAL	DC10	1994	269.40	204.74	\$64.66
			1995	999.40	759.54	\$239.86
			1996	1,079.97	816.21	\$257.75
	PORTUGAL	DC10	1993	1,315.15	942.79	\$372.36
			1994	1,436.85	993.08	\$443.77
			1995	3,056.85	2,108.59	\$948.26
	PORTUGAL	DC10	1996	3,418.48	2,358.04	\$1,060.43
			1993	346.90	218.55	\$128.35
			1994	290.10	192.76	\$107.34
	PORTUGAL	DC10	1995	1,051.10	662.19	\$388.91
			1996	1,131.69	712.96	\$418.72
			1993	1,565.00	970.30	\$594.70
	PORTUGAL	DC10	1994	2,894.00	1,794.28	\$1,099.72
			1995	4,223.00	2,618.26	\$1,604.74

Any enhancements?

more

82

PROC REPORT

Enhance the report by adding a blank line after each country and calculating profit for each country.

```

title1 ' ';
title2 'Profit Report by Country';
proc report nowindows data=sasuser.pm headline headskip;
  columns hub country type year income overhead profit;
  define hub / group;
  define country / group;
  define type / group width=6 spacing=2;
  define year / group width=6 spacing=2;
  define income / analysis;
  define overhead / analysis;
  define profit / computed format=dollar12.2 'Profit';
  compute profit;
    profit=income.sum - overhead.sum;
  endcompute;
  compute after country;
    line ' ';
  endcompute;
  break after country / summarize dul dol;
run;

```

Notice the following:

- ◆ **BREAK** statement,
- ◆ the new **COMPUTE** block,
- ◆ the new options on the **DEFINE** statements.

output

83

PROC REPORT

Profit Report by Country							
HUB	COUNTRY	TYPE	YEAR	INCOME	OVERHEAD	Profit	
FRANKFRT	AUSTRALIA	A300	1993	640.71	512.57	\$128.14	
			1994	251.29	201.03	\$50.26	
			1995	1,106.29	885.03	\$221.26	
			1996	1,318.61	1,054.89	\$263.72	
				3,316.90	2,653.52	\$663.38	
FRANKFRT	CANADA	747	1993	2,630.20	2,261.97	\$368.23	
			1994	3,595.20	3,091.87	\$503.33	
			1995	4,584.20	3,942.41	\$641.79	
			1996	4,703.86	4,045.32	\$658.54	
				15,513.46	13,341.57	\$2,171.88	
FRANKFRT	CHILE	747	1993	234.00	145.08	\$88.92	
			1994	456.00	282.72	\$173.28	
			1995	560.00	347.20	\$212.80	
			1996	600.00	372.00	\$228.00	
	DC10			1993	264.00	163.68	\$100.32
				1994	390.00	241.80	\$148.20
				1995	516.00	319.92	\$196.08
				1996	560.34	347.41	\$212.93
					3,580.34	2,219.81	\$1,360.53

Any enhancements?

more

84

PROC REPORT

Enhance the report by having less detail, drop TYPE and YEAR, then change the variable on the BREAK statement from COUNTRY to HUB.

```

title1 ' ';
title2 'Profit Report by Hub';
proc report nowindows data=sasuser.pm headline headskip;
  columns hub country income overhead profit;
  define hub / group;
  define country / group;
  define income / analysis;
  define overhead / analysis;
  define profit / computed format=dollar12.2 'Profit';
  compute profit;
    profit=income.sum - overhead.sum;
  endcompute;
  compute after hub;
  line ' ';
  endcompute;
  break after hub / summarize dul dol;
run;

```

Notice the following:

- ◆ **TYPE** and **YEAR** have been dropped from the report,
- ◆ the variable on the **COMPUTE** block,

output □

85

PROC REPORT

Profit Report by Hub				
HUB	COUNTRY	INCOME	OVERHEAD	Profit
FRANKFRT	AUSTRALIA	3,316.90	2,653.52	\$663.38
	CANADA	15,513.46	13,341.57	\$2,171.88
	CHILE	3,580.34	2,219.81	\$1,360.53
	JAPAN	32,124.98	23,387.53	\$8,737.45
	PORTUGAL	13,196.58	8,181.88	\$5,014.70
	UNITED STATES	31,395.76	22,693.76	\$8,702.00
FRANKFRT		99,128.02	72,478.07	\$26,649.95
LONDON	AUSTRALIA	6,673.60	5,311.87	\$1,361.73
	CANADA	51,480.80	43,391.81	\$8,089.00
	CHILE	26,809.50	17,597.67	\$9,211.83
	JAPAN	4,108.91	3,130.90	\$978.01
	PORTUGAL	56,871.95	39,241.64	\$17,630.30
	UNITED STATES	15,616.08	12,132.70	\$3,483.37
LONDON		161,560.84	120,806.60	\$40,754.24
NEW YORK	AUSTRALIA	7,954.26	5,436.45	\$2,517.80
	CANADA	4,215.20	3,582.92	\$632.28
	CHILE	1,840.18	1,370.77	\$469.41
	JAPAN	12,571.85	10,707.78	\$1,864.07
	PORTUGAL	822.33	592.48	\$229.85
	UNITED STATES	45,761.09	34,694.10	\$11,066.99
NEW YORK		73,164.91	56,384.50	\$16,780.41

Notice the level of summarization. Compare this report to the previous one. □

86

Calculating Percentages

Enhance the report by calculating the **Percent of Income by Country**.

```

title1 ' ';
title2 'Percent of Income by Country';
proc report nowindows data=sasuser.pm headline headskip;
  columns hub country income overhead profit income_pct;
  define hub / group;
  define country / group;
  define income / analysis;
  define overhead / 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 hub;
    income_sum=income.sum;
    income_pct=income.sum/income_sum;
  endcompute;
  break after hub / summarize dul dol skip;
run;

```

Notice the following:

- ◆ the **SKIP** option on the **BREAK** statement,
- ◆ the definition of **INCOME_PCT**
- ◆ the **BEFORE** option on the **COMPUTE** block at the **HUB** level,

output □

87

Calculating Percentages

Percent of Income by Country					
HUB	COUNTRY	INCOME	OVERHEAD	Profit	% of Income
FRANKFRT	AUSTRALIA	3,316.90	2,653.52	\$663.38	3.35%
	CANADA	15,513.46	13,341.57	\$2,171.88	15.65%
	CHILE	3,580.34	2,219.81	\$1,360.53	3.61%
	JAPAN	32,124.98	23,387.53	\$8,737.45	32.41%
	PORTUGAL	13,196.58	8,181.88	\$5,014.70	13.31%
	UNITED STATES	31,395.76	22,693.76	\$8,702.00	31.67%
FRANKFRT		99,128.02	72,478.07	\$26,649.95	100.0%
LONDON	AUSTRALIA	6,673.60	5,311.87	\$1,361.73	4.13%
	CANADA	51,480.80	43,391.81	\$8,089.00	31.86%
	CHILE	26,809.50	17,597.67	\$9,211.83	16.53%
	JAPAN	4,108.91	3,130.90	\$978.01	2.54%
	PORTUGAL	56,871.95	39,241.64	\$17,630.30	35.20%
	UNITED STATES	15,616.08	12,132.70	\$3,483.37	9.67%
LONDON		161,560.84	120,806.60	\$40,754.24	100.0%
NEW YORK	AUSTRALIA	7,954.26	5,436.45	\$2,517.80	10.87%
	CANADA	4,215.20	3,582.92	\$632.28	5.76%
	CHILE	1,840.18	1,370.77	\$469.41	2.52%
	JAPAN	12,571.85	10,707.78	\$1,864.07	17.18%
	PORTUGAL	822.33	592.48	\$229.85	1.12%
	UNITED STATES	45,761.09	34,694.10	\$11,066.99	62.55%
NEW YORK		73,164.91	56,384.50	\$16,780.41	100.0%

Notice the level of summarization. Compare this report to the previous one. □

88

Calculating Percentages

Enhance the report by summarizing at the HUB and COUNTRY level.

```

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 / computed 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 hub;
    income_sum=income.sum;
    income_pct=income.sum/income_sum;
  endcompute;
  break after hub / summarize dul dol skip;
  break after country / summarize dul dol skip;
run;

```

Notice the following:

- ◆ the **WHERE** option in the PROC statement,
- ◆ the two **BREAK** statements and their options,

output □

89

Calculating Percentages

Percent of Income for TOKYO Hub					
HUB	COUNTRY	YEAR	INCOME	% of Income	Profit
TOKYO	JAPAN	1993	537.90	0.40%	\$-25.53
		1994	1,111.39	0.83%	\$250.21
		1995	3,285.39	2.46%	\$716.38
		1996	4,023.82	3.02%	\$741.36
=====			8,958.50	6.72%	\$1,682.41
TOKYO	UNITED STATES	1993	25,035.00	18.78%	\$-721.67
		1994	28,721.50	21.54%	\$7,823.44
		1995	33,953.00	25.47%	\$9,326.81
		1996	36,662.27	27.50%	\$10,102.33
=====			124,371.77	93.28%	\$26,530.91
=====			133,330.27	100.0%	\$28,213.32
=====					

Notice that percent adds up to 100 at the HUB level. What if we wanted it to add up to 100 at the COUNTRY level? □

90

Modify the report so that **Income %** adds up to 100 at the country level.

```

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 before country;
income_sum=income.sum;
income_pct=income.sum/income_sum;
endcompute;
compute after hub;
income_sum=income.sum;
income_pct=.;
endcompute;
break after hub / summarize dul dol skip;
break after country / summarize dul dol skip;
run;

```

Notice the :
 ◆ the COMPUTE AFTER HUB block,
 ◆ the OPTIONS statement.

output □

Calculating Percentages

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
=====			8,958.50	100.0%	\$1,682.41
=====					
TOKYO	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
=====			124,371.77	100.0%	\$26,530.91
=====					
TOKYO			133,330.27		\$28,213.32
=====					

Notice that percent adds up to 100 at the COUNTRY level. Why is it blank at the HUB level?

The double lines over and under the % INCOME column need to be enhanced. How can we make this look better ?

□

9.9 Using ODS to Enhance the Report

What is ODS?

Beginning with Version 7, procedure output became much more flexible than earlier versions of the SAS System. The Output Delivery System (ODS) has been designed to overcome the limitations of traditional SAS output and to make it easy to make new formatting options available to users.

ODS is a method of delivering output in a variety of formats and of making the formatted output easy to access.

ODS combines data with one or more table definitions to produce one or more output objects. These objects can be sent to any or all ODS destinations.

The currently available ODS destinations can produce:

- ◆ an output data set
- ◆ traditional monospace output
- ◆ output that is formatted for a high-resolution printer, and
- ◆ output that is formatted in HTML.

output □

93

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;

Destination types can be:

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

table □

94

Creating Percentages

Create a report that shows what percentage each person's weight is of the whole class.

Sex	Name	Height	Weight	% Weight
F	Alice	56.50	84.00	4.42%
F	Barbara	65.30	98.00	5.16%
F	Carol	62.80	102.50	5.39%
F	Jane	59.80	84.50	4.45%
F	Janet	62.50	112.50	5.92%
F	Joyce	51.30	50.50	2.66%
F	Judy	64.30	90.00	4.74%
F	Louise	56.30	77.00	4.05%
F	Mary	66.50	112.00	5.89%
M	Alfred	69.00	112.50	5.92%
M	Henry	63.50	102.50	5.39%
M	James	57.30	83.00	4.37%
M	Jeffrey	62.50	84.00	4.42%
M	John	59.00	99.50	5.24%
M	Philip	72.00	150.00	7.89%
M	Robert	64.80	128.00	6.74%
M	Ronald	67.00	133.00	7.00%
M	Thomas	57.50	85.00	4.47%
M	William	66.50	112.00	5.89%
		=====	=====	=====
		62.34	100.03	100.0%
		=====	=====	=====

95

Creating Percentages

Notice the COLUMNS statement.

```
proc report data=sashelp.class nowd headline headskip;
  columns sex name height weight weight=wt_pc;
  define sex / format=$3.;
  define height / analysis mean format=8.2;
  define wt_pc / analysis pctsum format=percent8.2 '% Weight' ;
  define weight / analysis mean format=8.2;
  rbreak after / summarize dol dul;
run;
```

Notice the following about the **WT_PC** variable:

- ◆ it is created from the **WEIGHT** column.
- ◆ the **PCTSUM** statistic used in the definition,
- ◆ the **PERCENT.** format used in the definition.

96

Creating Percentages

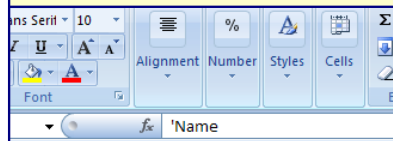
Sex	Name	Height	Weight	% Weight
F	Alice	56.50	84.00	4.42%
F	Barbara	65.30	98.00	5.16%
F	Carol	62.80	102.50	5.39%
F	Jane	59.80	84.50	4.45%
F	Janet	62.50	112.50	5.92%
F	Joyce	51.30	50.50	2.66%
F	Judy	64.30	90.00	4.74%
F	Louise	56.30	77.00	4.05%
F	Mary	66.50	112.00	5.89%
M	Alfred	69.00	112.50	5.92%
M	Henry	63.50	102.50	5.39%
M	James	57.30	83.00	4.37%
M	Jeffrey	62.50	84.00	4.42%
M	John	59.00	99.50	5.24%
M	Philip	72.00	150.00	7.89%
M	Robert	64.80	128.00	6.74%
M	Ronald	67.00	133.00	7.00%
M	Thomas	57.50	85.00	4.47%
M	William	66.50	112.00	5.89%
		=====	=====	=====
		62.34	100.03	100.0%
		=====	=====	=====

97

Linking

Create a report from the SASHELP.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 to show 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						

98

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

HTML - Code1

age_group	Gender					
	F		M			
	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%

□