Chapter 8

Producing Descriptive Statistics

Overview

- If the data values that you want to describe are continuous numeric values, then you can use the **MEANS** procedure or the **SUMMARY** procedure to calculate statistics such as the mean, sum, minimum, and maximum.

Variable	N	Mean	Std Dev	Minimum	Maximum
Age	20	47	13	15	63
Height	20	67	4	61	75
Weight	20	175	36	102	240
Pulse	70	75	8	65	100
FastGluc	20	299	126	152	568
PostGluc	20	355	126	206	625

- If the data values that you want to describe are discrete, then you can use the **FREQ** procedure to show the distribution of these values, such as percentages and counts.

Eye Color	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Brown	92	58.60	92	58.60
Blue				

Topics

This chapter will show you how to use the MEANS, SUMMARY, and FREQ procedures to describe your data. The topics are

- determine the *n*-count, mean, standard deviation, minimum, and maximum of numeric variables using the MEANS procedure
- control the number of decimal places used in PROC MEANS output
- specify the variables for which to produce statistics
- use the PROC SUMMARY procedure to produce the same results as the PROC MEANS procedure
- describe the difference between the SUMMARY and MEANS procedures
- create one-way frequency tables for categorical data using the FREQ procedure
- create two-way and n-way crossed frequency tables
- control the layout and complexity of crossed frequency tables.

Computing Statistics Using PROC MEANS

PROC MEANS DATA=*SAS-data-set*
<statistic-keyword(s)> <option(s)>;
RUN;

where

- *SAS-data-set* is the name of the data set to be used
- *statistic-keyword(s)* specifies the statistics to compute
- *option(s)* controls the content, analysis, and appearance of output.

The Simplest From of the PROC MEANS Procedure

```
proc means data=perm.survey;
run;
```

It produces some default statistics: *n*-count, mean, standard deviation, minimum, and maximum.

The MEANS Procedure					
Variable	N	Mean	Std Dev	Minimum	Maximum
Item1	4	3.7500000	1.2533057	2.0000000	5.0000000
Item2	4	3.0000000	1.6329932	1.0000000	5.0000000
Item3	4	4.2500000	0.5000000	4.0000000	5.0000000
ttem4	4	3.5000000	1.2909944	2.0000000	5.0000000
Item5	4	3.0000000	1.6329932	1.0000000	5.0000000
Item6	4	3.7500000	1.2533057	2.0000000	5.0000000
ttem7	4	3.0000000	1.8257419	1.0000000	5.0000000
Item8	4	2.7500000	1.5000000	1.0000000	4.0000000
Item9	4	3.0000000	1.4142136	2.0000000	5.0000000
Item10	4	3.2500000	1.2533057	2.0000000	5.0000000

PROC MEANS Procedure: Selecting Statistics

The procedure provides many keywords to compute statistics.

Descriptive Statistics

Keyword	Description
CLM	Two-sided confidence limit for the mean
CSS	Corrected sum of squares
CV	Coefficient of variation
KURTOSIS	Kurtosis
LCLM	One-sided confidence limit below the mean
MAX	Maximum value
MEAN	Average
MODE	Value that occurs most frequently (new in SAS 9.2)
MIN	Minimum value
N	Number of observations with nonmissing values
NMISS	Number of observations with missing values
RANGE	Range
SKEWNESS	Skewness
STDDEV / STD	Standard deviation
STDERR	Standard error of the mean
SUM	Sum
SUMWGT	Sum of the Weight variable values
UCLM	One-sided confidence limit above the mean
USS	Uncorrected sum of squares
VAR	Variance

Quantile Statistics

Keyword	Description
MEDIAN / P50	Median or 50th percentile
P1	1st percentile
P5	5th percentile
P10	10th percentile
Q1 / P25	Lower quartile or 25th percentile
Q3 / P75	Upper quartile or 75th percentile
P90	90th percentile
P95	95th percentile
P99	99th percentile
QRANGE	Difference between upper and lower quartiles: Q3-Q1

Hypothesis Testing

Keyword	Description
PROBT	Probability of a greater absolute value for the t value
T	Student's t for testing the hypothesis that the population mean is 0

PROC MEANS Procedure: Selecting Statistics Example

To see the **median** and **range** of Perm.Survey numeric values, add the MEDIAN and RANGE keywords.

```
proc means data=perm.survey median range;
run;
```

MEANS Procedure Output Displaying Median and Range		
The MEANS Procedure		
Variable	Median	Range
Item1	4.0000000	3.0000000
Item2	3.0000000	4.0000000
Item3	4.0000000	1.0000000
Item4	3.5000000	3.0000000
Item5	3.0000000	4.0000000
Item6	4.0000000	3.0000000
Item7	3.0000000	4.0000000
Item8	3.0000000	3.0000000
Item9	2.5000000	3.0000000
Item10	3.0000000	3.0000000

Note:
When you specify a statistic keyword in the PROC MEANS statement, the default statistics are not produced.

PROC MEANS Procedure: Limiting Decimal Places

By default, PROC MEANS output automatically uses the *BESTw.* format to display numeric values in the report. When there is no format specification, SAS chooses the format that provides the most information about the value according to the available field width. This can result in unnecessary decimal places. You can use the **MAXDEC=n** (n: the maximum number of decimal places) option to limit the decimal places.

```
proc means data=clinic.diabetes
      min max;
run;
```

The MEANS Procedure		
Variable	Minimum	Maximum
Age	15.0000000	63.0000000
Height	61.0000000	75.0000000
Weight	102.0000000	240.0000000
Pulse	65.0000000	100.0000000
FastGluc	152.0000000	568.0000000
PostGluc	206.0000000	625.0000000

```
proc means data=clinic.diabetes
      min max maxdec=0;
run;
```

The MEANS Procedure		
Variable	Minimum	Maximum
Age	15	63
Height	61	75
Weight	102	240
Pulse	65	100
FastGluc	152	568
PostGluic	206	625

Specifying Variables in PROC MEANS

By default, the MEANS procedure generates statistics for every numeric variable in a data set. To specify the variables, add a **VAR** statement and list the variable names.

```
proc means data=clinic.diabetes min max maxdec=0;
        var age height weight;
run;
```

The MEANS Procedure		
Variable	Minimum	Maximum
Age	15	63
Height	61	75
Weight	102	240

Specifying Variables in PROC MEANS

You can use a numbered range of variables.

```
proc means data=perm.survey mean stderr maxdec=2;
        var item1-item5;
run;
```

The MEANS Procedure		
Variable	Mean	Std Error
Item1	3.75	0.63
Item2	3.00	0.82
Item3	4.25	0.25
Item4	3.50	0.65
Item5	3.00	0.82

PROC MEANS: Group Processing Using the CLASS Statement

To produce separate statistics of grouped observations, add a CLASS statement to the MEANS procedure.

```
proc means data=clinic.heart
        maxdec=1;
        var arterial heart cardiac
        urinary;
        class survive sex;
run;
```

The MEANS Procedure

Survive	Sex	N Obs	Variable	N	Mean	Std Dev	Minimum	Maximum
DIED	1	4	Arterial	4	92.5	10.5	83.0	103.0
			Heart	4	111.0	53.4	54.0	183.0
			Cardiac	4	176.8	75.2	95.0	260.0
			Urinary	4	98.0	186.1	0.0	377.0
	2	6	Arterial	6	94.2	27.3	72.0	145.0
			Heart	6	103.7	16.7	81.0	130.0
			Cardiac	6	318.3	102.6	156.0	424.0
			Urinary	6	100.3	155.7	0.0	405.0
SURV	1	5	Arterial	5	77 2	12 2	61 0	88 0
			Heart	5	109.0	32.0	77.0	149.0
			Cardiac	5	298.0	139.8	66.0	410.0
			Urinary	5	100.8	60.2	44.0	200.0
	2	5	Arterial	5	78.8	6.8	72.0	87.0
			Heart	5	100.0	13.4	84.0	111.0
			Cardiac	5	330.2	87.0	256.0	471.0
			Urinary	5	111.2	152.4	12.0	377.0

PROC MEANS: Group Processing Using the BY Statement

To produce separate statistics of grouped observations, you can **also** use a BY statement in the MEANS procedure. However, BY and CLASS differ in the following ways:

- BY processing requires that your data already be sorted or indexed in the order of the BY variables. Unless data set observations are already sorted, you will need to run the SORT procedure before using PROC MEANS with any BY group.
- BY group results have a layout that is different from the layout of CLASS group results. The BY statement creates small separate tables; a CLASS statement would produce a single large table.
- Because it doesn't require a sorting step, the CLASS statement is easier to use than the BY statement. However, BY group processing can be more efficient when your categories might contain many levels.

PROC MEANS: Group Processing Using the BY Statement (cont'd)

```
proc sort data=clinic.heart out=work.heartsort;
    by survive sex;
run;
proc means data=work.heartsort maxdec=1;
    var arterial heart cardiac urinary;
    by survive sex;
run;
```

The MEANS Procedure					
Survive=DIED Sex=1					
Variable	N	Mean	Std Dev	Minimum	Maximum
Arterial	4	92.5	10.5	83.0	103.0
Heart	4	111.0	53.4	54.0	183.0
Cardiac	4	176.8	75.2	95.0	260.0
Urinary	4	98.0	186.1	0.0	377.0
Survive=DIED Sex=2					
Variable	N	Mean	Std Dev	Minimum	Maximum
Arterial	6	94.2	27.3	72.0	145.0
Heart	6	103.7	16.7	81 0	130.0
Cardiac	6	318.3	102.6	156.0	424.0
Urinary	6	100.3	155.7	0.0	405.0
Survive=SURV Sex=1					
Variable	N	Mean	Std Dev	Minimum	Maximum
Arterial	5	77.2	12.2	61.0	88.0
Heart	5	109.0	32.0	77.0	149.0
Cardiac	5	298.0	139.8	66.0	410.0
Urinary	5	100.8	60.2	44.0	200.0
Survive=SURV Sex=2					
Variable	N	Mean	Std Dev	Minimum	Maximum
Arterial	5	78.8	6.8	72.0	87.0
Heart	5	100.0	13.4	84.0	111.0
Cardiac	5	330.2	87.0	256.0	471.0
Urinary	5	111.2	152.4	12.0	377.0

Creating a Summarized Data Set Using PROC MEANS

You can create an output SAS data set by using the OUTPUT statement in PROC MEANS. The Syntax is

OUTPUT OUT=*SAS-data-set statistic=variable(s)*;

where

- **OUT=** specifies the name of the output data set
- *statistic*= specifies the summary statistic written out
- *variable(s)* specifies the names of the variables to create. These variables represent the statistics for the analysis variables that are listed in the VAR statement.

Creating a Summarized Data Set Using PROC MEANS (cont'd)

```
proc means data=clinic.diabetes; /*add the noprint option to suppress the report */
    var age height weight;
    class sex;
    output out=work.sum_gender mean=AvgAge AvgHeight AvgWeight min=MinAge
        MinHeight MinWeight;
run;
```

- When you use the OUTPUT statement, the summary statistics N, MEAN, STD, MIN, and MAX are produced in your report for all of the numeric variables or for *all* of the variables that are listed in a VAR statement *by default*.
- **The variables must be listed in the same order as in the VAR statement.**

The MEANS Procedure

Sex	N Obs	Variable	N	Mean	Std Dev	Minimum	Maximum
F	11	Age	11	48.9090909	13.307550S	16.0000000	63.0000000
		Height	11	63.9090909	2.1191765	61.0000000	68.0000000
		Weight	11	150.4545455	18.4464828	102.0000000	168.0000000
M	9	Age	9	44.0000000	12.3895117	15.0000000	54.0000000
		Height	9	70.6665667	2.6457513	66.0000000	75.0000000
		Weight	9	204.2222222	30.2893454	140.0000000	240.0000000

Creating a Summarized Data Set Using PROC MEANS (cont'd)

To see the contents of the output data set, submit the following PROC PRINT step.

```
proc print data=work.sum_gender;
run;
```

Obs	Sex	_TYPE_	_FREQ_	AvgAge	AvgHeight	AvgWeight	MinAge	MinHeight	MinWeight
1		0	20	46.7000	66.9500	174.650	15	61	102
2	F	1	11	48.9091	63.9091	150.455	16	61	102
3	M	1	9	44.0000	70.6667	204.222	15	66	140

Note: the procedure adds the _TYPE_ and _FREQ_ variables to the output data set.

Creating a Summarized Data Set Using PROC SUMMARY

In **PROC SUMMARY**, you use very similar code to produce the output data set that you would use with PROC MEANS. The difference is that PROC MEANS produces a report by default (but you can use the NOPRINT option to suppress it). By contrast, you must include a PRINT option in the PROC SUMMARY statement to produce a report .

```
proc summary data=clinic.diabetes;
     var age height weight;
     class sex;
     output out=work.sum_gender
               mean=AvgAge AvgHeight AvgWeight;
run;
```

This code creates an output data set but does not create a report.

Creating a Summarized Data Set Using PROC SUMMARY

Use the PRINT option in the PROC SUMMARY statement to produce a report, which is the same report by using PROC MEANS .

```
proc summary data=clinic.diabetes print;
     var age height weight;
     class sex;
     output out=work.sum_gender
               mean=AvgAge AvgHeight AvgWeight;
run;
```

The SUMMARY Procedure							
Sex	N Obs	Variable	N	Mean	Std Dev	Minimum	Maximum
F	11	Age	11	48.9090909	13.3075508	16.0000000	63.0000000
		Height	11	63.9090909	2.1191765	61.0000000	68.0000000
		Weight	11	150.4545455	18.4464828	102.0000000	168.0000000
M	9	Age	9	44.0000000	12.3895117	15.0000000	54.0000000
		Height	9	70.6666667	2.6457513	66.0000000	75.0000000
		Weight	9	204.2222222	30.2893454	140.0000000	240.0000000

Producing Frequency Tables Using PROC FREQ

You can use the FREQ procedure to

- produce **one-way** and ***n*-way** frequency tables
- report the distribution of variable values
- create crosstabulation tables that summarize data for two or more categorical variables by showing the number of observations for each combination of variable values.

By default, PROC FREQ creates a one-way table with the frequency, percent, cumulative frequency, and cumulative percent of every value of all variables in a data set. This can produce excessive or inappropriate output. It is recommended that you always use a TABLES statement with PROC FREQ.

Syntax:

```
PROC FREQ  DATA=SAS-data-set ;
<TABLES variable(s);>
RUN;
```

Specifying Variables in PROC FREQ

The FREQ procedure works best with categorical variables, whose values are best summarized by counts rather than by averages. To specify the variables , use the TABLES statement.

```
proc freq data=finance.loans;
   tables rate months;
run;
```

The variables Rate and Months are best described as categorical variables.

Rate	Frequency	Percent	Cumulative Frequency	Cumulative Percent
9.50%	2	22.22	2	22.22
9.75%	1	11.11	3	33.33
10.00%	2	22.22	5	55.56
10.50%	4	44.44	9	100.00

Months	Frequency	Percent	Cumulative Frequency	Cumulative Percent
12	1	11.11	1	11.11
24	1	11.11	2	22.22
36	1	11.11	3	33.33
48	1	11.11	4	44.44
60	2	22.22	6	66.67
360	3	33.33	9	100.00

Specifying Variables in PROC FREQ

You can use a numbered range of variables:

```
proc freq data=perm.survey;
  tables item1-item3;
run;
```

The FREQ Procedure				
Item1	Frequency	Percent	Cumulative Frequency	Cumulative Percent
2	1	25.00	1	25.00
4	2	50.00	3	75.00
5	1	25.00	4	100.00
Item2	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	1	25.00	1	25.00
3	2	50.00	3	75.00
5	1	25.00	4	100.00
Item3	Frequency	Percent	Cumulative Frequency	Cumulative Percent
4	3	75.00	3	75.00
5	1	25.00	4	100.00

Specifying Variables in PROC FREQ

You can use the **NOCUM** option to suppresses the display of cumulative frequencies and cumulative percentages in one-way frequency tables :

```
proc freq data=perm.survey;
  tables item1-item3 /nocum;
run;
```

The FREQ Procedure		
Item1	Frequency	Percent
2	1	25.00
4	2	50.00
5	1	25.00
Item2	Frequency	Percent
1	1	25.00
3	2	50.00
5	1	25.00
Item3	Frequency	Percent
4	3	75.00
5	1	25.00

Creating Two-Way Tables in PROC FREQ

It is often helpful to crosstabulate frequencies of two or more variables. The simplest crosstabulation is a two-way table. To create a two-way table, join two variables with an asterisk (*) in the TABLES statement .

TABLES *variable-1 * variable-2 <…variable-n>;*

↑ Row ↑ Column

```
proc format;
value wtfmt low-139='< 140' 140-180='140-
                180' 181-high='> 180';
value htfmt low-64='< 5''5"' 65-70='5''5-10"'
                71-high='> 5''10"';
run;

proc freq data=clinic.diabetes;
        tables weight*height;
        format weight wtfmt. height htfmt.;
run;
```

Frequency
Percent
Row Pct
Col Pct

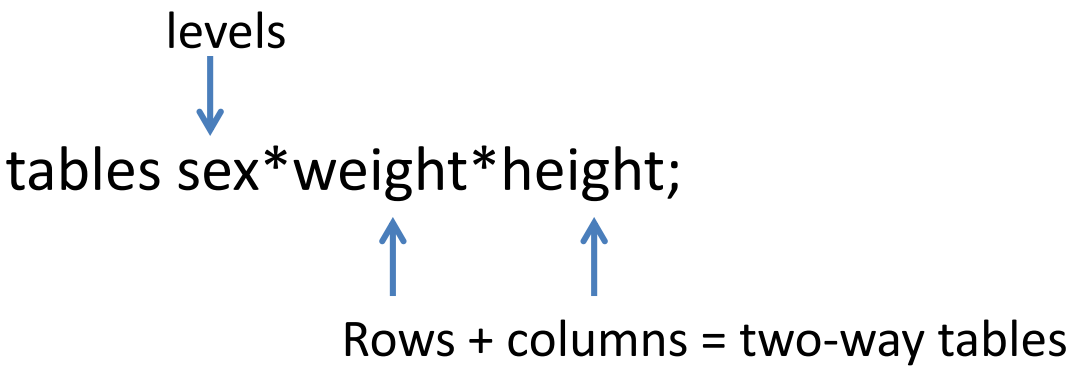
Table of Weight by Height

Weight	Height			
	< 5'5"	5'5-10"	> 5'10"	Total
< 140	2	0	0	2
	10.00	0.00	0.00	10.00
	100.00	0.00	0.00	
	28.57	0.00	0.00	
140-180	5	5	0	10
	25.00	25.00	0.00	50.00
	50.00	50.00	0.00	
	71.43	62.50	0.00	
> 180	0	3	5	8
	0.00	15.00	25.00	40.00
	0.00	37.50	62.50	
	0.00	37.50	100.00	
Total	7	8	5	20
	35.00	40.00	25.00	100.00

Creating *N*-Way Tables in PROC FREQ

- For a frequency analysis of more than two variables, use PROC FREQ to create *n*-way crosstabulations. A series of two-way tables is produced, with a table for each level of the other variables.
- **The order of the variables is important**. The last two variables of the TABLES statement become the two-way rows and columns. Variables that precede the last two variables stratify the crosstabulation tables.

```
PROC FREQ data=clinic.diabetes;
   tables sex*weight*height;
   format weight wtfmt. height htfmt.;
Run;
```



Creating *N*-Way Tables in PROC FREQ

```
proc format;
   value wtfmt
         low-139='< 140'
         140-180='140-180'
         181-high='> 180';
   value htfmt
         low-64='< 5''5"'
         65-70='5''5-10"'
         71-high='> 5''10"';
run;

proc freq data=clinic.diabetes;
   tables sex*weight*height;
   format weight wtfmt. height
   htfmt.;
run;
```

"Sex" is the stratification variable, "weight" is rows, and "height" is columns.

The FREQ Procedure

Table 1 of Weight by Height
Controlling for Sex=F

Frequency Percent Row Pct Col Pct	Height			
Weight	< 5'5"	5'5-10"	> 5'10"	Total
< 140	2	0	0	2
	18.18	0.00	0.00	18.18
	100.00	0.00	0.00	
	28.57	0.00	.	
140-180	5	4	0	9
	45.45	36.36	0.00	81.82
	55.56	44.44	0.00	
	71.43	100.00	.	
> 180	0	0	0	0
	0.00	0.00	0.00	0.00
	.	.	.	
	0.00	0.00	.	
Total	7	4	0	11
	63.64	36.36	0.00	100.00

Table 2 of Weight by Height
Controlling for Sex=M

Frequency Percent Row Pct Col Pct	Height			
Weight	< 5'5"	5'5-10"	> 5'10"	Total
< 140	0	0	0	0
	0.00	0.00	0.00	0.00
	.	.	.	
	.	0.00	0.00	
140-180	0	1	0	1
	0.00	11.11	0.00	11.11
	0.00	100.00	0.00	
	.	25.00	0.00	
> 180	0	3	5	8
	0.00	33.33	55.56	88.89
	0.00	37.50	62.50	
	.	75.00	100.00	
Total	0	4	5	9
	0.00	44.44	55.56	100.00

Creating Tables in List Format in PROC FREQ

Often complex crosstabulations are easier to read as a continuous list. To generate the list output, add a slash (**/**) and the **LIST** option to the TABLES statement.

```
proc format;
   value wtfmt
         low-139='< 140'
         140-180='140-180'
         181-high='> 180';
   value htfmt
         low-64='< 5''5"'
         65-70='5''5-10"'
         71-high='> 5''10"';
run;
proc freq data=clinic.diabetes;
   tables sex*weight*height /list;
   format weight wtfmt. height
   htfmt.;
run;
```

Sex	Weight	Height	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	< 140	< 5'5"	2	10.00	2	10.00
F	140-180	< 5'5"	5	25.00	7	35.00
F	140-180	5'5-10"	4	20.00	11	55.00
M	140-180	5'5-10"	1	5.00	12	60.00
M	> 180	5'5-10"	3	15.00	15	75.00
M	> 180	> 5'10"	5	25.00	20	100.00

Note: When you use the LIST option, statistical options cannot be used.

Changing the Table Format in PROC FREQ

Beginning in SAS®9, you can use the **CROSSLIST** option to display crosstabulation tables in ODS (Output Delivery System) column format, which can be customized by the TEMPLATE procedure later.

```
proc format;
   value wtfmt
         low-139='< 140'
         140-180='140-180'
         181-high='> 180';
   value htfmt
         low-64='< 5''5"'
         65-70='5''5-10"'
         71-high='> 5''10"';
run;
proc freq data=clinic.diabetes;
   tables sex*weight*height /crosslist;
   format weight wtfmt. height htfmt.;
run;
```

Changing the Table Format in PROC FREQ

The Result

Table of Weight by Height					
Controlling for Sex=F					
Weight	Height	Frequency	Percent	Row Percent	Column Percent
< 140	< 5'5"	2	18.18	100.00	28.57
	5'5-10"	0	0.00	0.00	0.00
	> 5'10"	0	0.00	0.00	.
	Total	2	18.18	100.00	
140-180	< 5'5"	5	45.45	55.56	71.43
	5'5-10"	4	36.36	44.44	100.00
	> 5'10"	0	0.00	0.00	.
	Total	9	81.82	100.00	
> 180	< 5'5"	0	0.00	.	0.00
	5'5-10"	0	0.00	.	0.00
	> 5'10"	0	0.00	.	.
	Total	0	0.00	.	
Total	< 5'5"	7	63.64		100.00
	5'5-10"	4	36.36		100.00
	> 5'10"	0	0.00		.
	Total	11	100.00		

Table of Weight by Height					
Controlling for Sex=M					
Weight	Height	Frequency	Percent	Row Percent	Column Percent
< 140	< 5'5"	0	0.00	.	.
	5'5-10"	0	0.00	.	0.00
	> 5'10"	0	0.00	.	0.00
	Total	0	0.00	.	
140-180	< 5'5"	0	0.00	0.00	.
	5'5-10"	1	11.11	100.00	25.00
	> 5'10"	0	0.00	0.00	0.00
	Total	1	11.11	100.00	
> 180	< 5'5"	0	0.00	0.00	.
	5'5-10"	3	33.33	37.50	75.00
	> 5'10"	5	55.56	62.50	100.00
	Total	8	88.89	100.00	
Total	< 5'5"	0	0.00		.
	5'5-10"	4	44.44		100.00
	> 5'10"	5	55.56		100.00
	Total	9	100.00		

Suppressing Table Information in PROC FREQ

You can use some options in the TABLES statement to suppress some statistics and to control the depth of crosstabulation results. These options are

- NOFREQ suppresses cell frequencies
- NOPERCENT suppresses cell percentages
- NOROW suppresses row percentages
- NOCOL suppresses column percentages

Suppressing Table Information in PROC FREQ

```
proc format;
      value wtfmt
            low-139='< 140'
            140-180='140-180'
            181-high='> 180';
run;
      proc freq data=clinic.diabetes;
             tables sex*weight /
                  nofreq norow nocol;
             format weight wtfmt.;
run;
```

The FREQ Procedure

Percent

Table of Sex by Weight

Sex	Weight			
	< 140	140-180	> 180	Total
F	10.00	45.00	0.00	55.00
M	0.00	5.00	40.00	45.00
Total	2 10.00	10 50.00	8 40.00	20 100.00

You have only cell percentages left.