ABSTRACT
Calculating the totals correctly is not the end of a PROC REPORT project, it is the beginning of the end. A key part of completing a table is to decide how to present the numbers. This paper discusses two parts of a table: cells and footnotes. Displaying cells with only numbers, with only symbols, and with both numbers and symbols is discussed separately. The use of standard SAS formats and user defined formats to display standard numbers and numbers truncated, rounded, and expressed to two significant digits is demonstrated. Compute statements are used to illustrate how to display table results to two significant digits and to add warning flags. When numbers are not enough, cell notes are needed. Cell notes are numbers or symbols that replace or supplement numbers in a cell. Examples of cell notes are symbols to represent missing data (NA), and values suppressed to prevent disclosure (D). Examples are given of the use of the COMPUTE statement to create conditional cell notes to either replace a number with a cell note or to add an accompanying symbol to the number. A brief discussion of footnotes along and an example of PROC REPORT code showing how to display footnotes is given.

INTRODUCTION
While there are guidelines and styles for tables, organization do them differently. They are different because of the needs of the user, specialized content of the data they display, preferences of individuals developing them, and ultimately their purpose. Tables are usually of three types: Official documents, working tables, and presentation tables. Tables as official documents record results of surveys, and experiments. Often they have an associated obligation to preserve privacy, warn users of potential problems, or explain when may be missing. Tables as reports of new information have different objectives of wanting the reader to compare data or to see a pattern. How the numbers are presented influence this. Often to make the patterns more recognizable, numbers are rounded, truncated, or expressed to two or three significant digits.

PARTS OF A TABLE
A table is compose of five basic parts: (1) Title, (2) Boxhead (columns), (3) Stub (rows), (3) Field (cells), and (5) Footnote. Each has a separate and distinct purpose.

The heading consists of the table number, and headnote. The table number is an identifying method; the title is a brief statement of the nature, classification, and time reference of the information in the table and serves as a reference guide.

The boxhead describes the data contained in the individual column heads or captions. It may also contain banner and spanner heads that classify or qualify a caption spreading across two or more individual columns or lower level spanners. The individual column head gives a descriptive title to the individual column to which it refers. A banner head is a spanner which extends across the entire boxhead of the table.

The stub is almost always the first column on the left and contains the line captions and associated center heads and stubheads. The stub describes table data and makes clear the relationship among rows or lines. The line caption describes data on a particular line or row of cells. The stubhead describes the stub listing as a whole in terms of classifications presented. A block or center head may also be included in the stub.

The field is the depository of statistical information in the table and the area below the column headings and to the right of the stub. The field is made up of columns, rows, and cells. The cell is the basic element in presentation in the table. Cells can contain numbers, words, symbols, or a combination of the three. A row or line is a group of cells in a horizontal line; a column is a group of cells in a vertical line.

The footnote is information put at the bottom of the table. It can be either a specific footnote, source note, or general note. A specific footnote explains a certain cell entry, column heading, line caption, or word. A source note explains where the information in the table comes from. A general note qualifies, describes, or explains the table as a whole.

This paper concerns the cells that make up the table's field and with the footnotes that explain notations in cells to users. There are a number of excellent references on the preparation of statistical tables (Census Bureau (1949), EIA (1970)).

THE PROBLEM AND A SOLUTION
The problem is that just tabulating the data is not enough. Missing values need to be dealt with, numbers need to formatted or rounded, strange values need to be explained, and special rules need to be applied. All these are non mathematical; but, all are necessary. The example below illustrates both the problem and a solution.

Problem
The problem below is missing values and explanation why there is not ratio for the South (S) region when data exists for both revenue and expense. Some think it is a cluttered and not easily read.

<table>
<thead>
<tr>
<th>Region</th>
<th>firms</th>
<th>revenue</th>
<th>expense</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>NE</td>
<td>200</td>
<td>1752</td>
<td>1293</td>
<td>1.35</td>
</tr>
<tr>
<td>MW</td>
<td>23</td>
<td>375</td>
<td>4291</td>
<td>0.09</td>
</tr>
<tr>
<td>S</td>
<td>150</td>
<td></td>
<td>2950</td>
<td>.</td>
</tr>
<tr>
<td>W</td>
<td>2</td>
<td>3156</td>
<td>0</td>
<td>.</td>
</tr>
<tr>
<td>Total</td>
<td>375</td>
<td>5283</td>
<td>8534</td>
<td>0.62</td>
</tr>
</tbody>
</table>
Solution
The solution is to change how the data is displayed and add comments. Revenue and Expense have been rounded to thousands, missing values identified by a symbol, value rounded to zero identified, and data associated with 2 or fewer firms suppressed. The meaning of the cell notes are explained in a specific footnote at the bottom of the table.

<table>
<thead>
<tr>
<th>Region</th>
<th>firms</th>
<th>REVENUE (000s)</th>
<th>EXPENSE (000s)</th>
<th>RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>NE</td>
<td>200</td>
<td>2</td>
<td>1</td>
<td>1.35</td>
</tr>
<tr>
<td>MW</td>
<td>23</td>
<td>(Z)</td>
<td>4</td>
<td>0.09</td>
</tr>
<tr>
<td>S</td>
<td>150</td>
<td>(M)</td>
<td>3</td>
<td>(M)</td>
</tr>
<tr>
<td>W</td>
<td>2</td>
<td>(S)</td>
<td>(S)</td>
<td>(NM)</td>
</tr>
</tbody>
</table>

(23/200) = 0.09

The final table many times requires a table statement. This is illustrated in the table below. The solution is use a computation statement. The table produced by the PROC REPORT code above is given below.

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2</td>
<td>Standard numeric</td>
<td>-12334.57</td>
</tr>
<tr>
<td>COMMA12.2</td>
<td>Write numeric values with</td>
<td>-1,234.57</td>
</tr>
<tr>
<td></td>
<td>commas and decimal points</td>
<td></td>
</tr>
<tr>
<td>COMMAX12.2</td>
<td>Write numeric values with</td>
<td>-1,234.57</td>
</tr>
<tr>
<td></td>
<td>commas and periods.</td>
<td></td>
</tr>
<tr>
<td>DOLLARr12.2</td>
<td>Write numeric values with</td>
<td>$-1,234.57</td>
</tr>
<tr>
<td></td>
<td>dollar signs, commas, and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>decimal points</td>
<td></td>
</tr>
<tr>
<td>NEGPAREN12.2</td>
<td>Write negative numeric</td>
<td>(1,234.57)</td>
</tr>
<tr>
<td></td>
<td>values in parentheses</td>
<td></td>
</tr>
<tr>
<td>E12.</td>
<td>Scientific notation</td>
<td>-1.23457E+03</td>
</tr>
</tbody>
</table>

Formats allow the programmer to customize a table to different audiences and make numbers more readable (dollar signs, comma separators and parentheses for negative numbers).

Truncating and Rounding
Rounding is usually expressed in terms such as “rounding to two decimal places” or “rounding to thousands”. Truncation refers to either dropping left most digits of a number or replacing the digits with zeros. In general formats can not be used for rounding; however, they can be used to round the decimal parts of a number. Formats don’t work for the integer part. If you need to round results in PROC REPORT table, the solution is use a compute statement.

```sas
proc report data=example nowd;
    columns row original truncate format;
    define row / order;
    define original / display = 'Original';
    define truncate / display format='comma18.0';
    define format / display = 'Rounded';
    compute truncate = int (truncate / 100) *100;
    endcomp;
    compute format = int (( format + 50)/100) *100;
    endcomp;
    run;
```

The table produced by the PROC REPORT code above is given below.

<table>
<thead>
<tr>
<th>row</th>
<th>Original</th>
<th>Truncated</th>
<th>Rounded</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>99123</td>
<td>99,100</td>
<td>99100</td>
</tr>
<tr>
<td>2</td>
<td>99133</td>
<td>99,100</td>
<td>99100</td>
</tr>
<tr>
<td>3</td>
<td>99143</td>
<td>99,100</td>
<td>99100</td>
</tr>
<tr>
<td>4</td>
<td>99153</td>
<td>99,100</td>
<td>99200</td>
</tr>
<tr>
<td>5</td>
<td>99163</td>
<td>99,100</td>
<td>99200</td>
</tr>
<tr>
<td>6</td>
<td>99173</td>
<td>99,100</td>
<td>99200</td>
</tr>
<tr>
<td>7</td>
<td>99183</td>
<td>99,100</td>
<td>99200</td>
</tr>
<tr>
<td>8</td>
<td>99193</td>
<td>99,100</td>
<td>99200</td>
</tr>
<tr>
<td>9</td>
<td>99203</td>
<td>99,200</td>
<td>99200</td>
</tr>
</tbody>
</table>

CELL NOTES
The cell is the basic object of most tables. A cell can contain:

- number only
- number and unit-indicator
- number and symbol
- symbol only

While the purpose of a statistical table is to display numeric data in table cells, special situations require comments be added to the data or in place of the data. These comments are called cell notes because data is missing, has been changed in some nonstandard way, are not meaningful, or the sampling error imputation rate is too high.

CELLS WITH ONLY NUMBERS
Most cells in statistical tables contain only numbers. Changes to calculated values occur because you want to present the data more meaningfully by simplifying the numbers thru truncation, rounding, or significant digits. Truncation refers to either dropping some of the right most digits (123897 becomes 123) or replacing the dropped numbers with zeros; rounding refers to changing the right most digit displayed depending on whether the first drop digit was greater than or equal to 5 or less than 5 (123897 rounded to thousands becomes 124000); and significant digits refers to keeping only the first two or three digits regardless of the position in the number and changing the others to zero (123897 to two significant digits is 120000).

SAS Formats
SAS supplied formats provide an easy and simple way to increase the readability and understandability of data presented in a PROC REPORT table. Use of SAS defined formats allow the presentation of understandable standard numbers. Common SAS supplied defined formats used in tables are the COMMA and DOLLAR formats. European versions for these also exist (DOLLARxw.d and COMMANw.d). SAS comes with a wide variety of formats. They are described in excruciating detail in the SAS Language Reference Manual. Some common standard SAS formats are given below and show the same number displayed in different formats.
Displaying data with a lot of digits they argue suggests spurious patterns in the data. To do this, some table makers argue precisely. The purpose of other reports is to show relationships. Some reports are for the record and data must be expressed accurately and makes meaningful comparisons difficult. The significant digits refer to the digits of a number from left to right starting with the first nonzero digit. Some examples are given below.

<table>
<thead>
<tr>
<th>Original Number</th>
<th>Truncated to thousands using trunc12. format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.23456789</td>
<td>Z</td>
</tr>
<tr>
<td>12.3456789</td>
<td>Z</td>
</tr>
<tr>
<td>123.456789</td>
<td>Z</td>
</tr>
<tr>
<td>1234.56789</td>
<td>1</td>
</tr>
<tr>
<td>12345.7869</td>
<td>12</td>
</tr>
<tr>
<td>123456.789</td>
<td>123</td>
</tr>
<tr>
<td>1234567.89</td>
<td>1,234</td>
</tr>
</tbody>
</table>

**Examples of One, Two, and Three Significant Digits**

<table>
<thead>
<tr>
<th>Original Number</th>
<th>One Digit Significant</th>
<th>Two Digits Significant</th>
<th>Three Digits Significant</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234.578</td>
<td>1234</td>
<td>12345</td>
<td>123456</td>
</tr>
</tbody>
</table>

Thus this SAS data set can be create use the following code. This file can be used to convert any number from zero to 10^13 to a number with two significant digits. A new file would be needed if you wanted to expressed number to three significant digits or for bigger numbers.

```sas
data fmt_brute;
  do i = 0 to 9;
    j = i;
    start = i;
    end = i;
    y = put ( y, comma10.);
    label = x;
    fmtname = 'brute';
    output;
    end;
  end;
data example;
  precision for a variable through out the table.
  When you use formats, you are locked in to the same level of significant digits and the grand total to three significant digits. The compute statement is that you can express basic rows to two significant digits in a compute statement. One advantage of the actually creates the table. It demonstrates calculating the COMPUTE STATEMENT – The PROC REPORT code below showing the three different approaches is given below.

**PROC FORMAT with Picture STATEMENT – This is the most logical approach; but, it is suffers from the problem that you need to substitute the capital letter "O" for the number "0" to display trailing "zeros". This is needed because of a limitation of PROC FORMAT. To use this you need to know the maximum value you will encounter. A user defined format to do this is given below.**

```sas
PROC FORMAT;
picture trunc
  i='0'
  1-999='Z'
1000-999999='000'
100000-999999999='000,000,000'
run;
```

```sas
proc format cntlin=fmt_brute;
  label   = x;
x       = put ( y, comma10.);
start   = i*10**(j-1) ;
end     = i*10**(j-1) + (10**(j -1)-1 );
x       = put ( y, comma10.);
label   = x;
fmtname = 'brute';
output;
end;
end;
run;
```

```sas
proc format cntlin=fmt_brute;
run;
```

```sas
proc format cntlin=fmt_brute;
run;
```

**COMPUTE STATEMENT – The PROC REPORT code below actually creates the table. It demonstrates calculating the significant digits in a compute statement. One advantage of the compute statement is that you can express basic rows to two significant digits and the grand total to three significant digits. When you use formats, you are locked in to the same level of precision for a variable through out the table.**

```sas
DATA example;
  DO I=1 TO 14;
    X= (UNIFORM(0)) *10**6;
    Y=int ( X );
    Z=int ( X );
    W=int(X);
    OUTPUT;
  END;
```

<table>
<thead>
<tr>
<th>GROUP</th>
<th>ORIGINAL</th>
<th>PICTURE</th>
<th>BRUTE</th>
<th>FORCE</th>
<th>COMPUTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,757,239</td>
<td>1,700,000</td>
<td>1,700,000</td>
<td>1,700,000</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>374,824</td>
<td>370,000</td>
<td>370,000</td>
<td>370,000</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>186,945</td>
<td>180,000</td>
<td>180,000</td>
<td>180,000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1,921,500</td>
<td>1,900,000</td>
<td>1,900,000</td>
<td>1,900,000</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>394,672</td>
<td>390,000</td>
<td>390,000</td>
<td>390,000</td>
<td></td>
</tr>
<tr>
<td>13,502,022</td>
<td>13,000,000</td>
<td>13,000,000</td>
<td>13,500,000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
proc report data=example nowd;  
column row ('.Distance_' distance xx);  
define row / display;  
define distance / analysis format=comma12.  
  (feet)';  
define xx / computed '"(Recoded)';  
compute xx / char length=16;  
if distance.sum >0 then xx =0;  
if distance.sum >0 and distance.sum <3  
then xx = (right(put(int((distance.sum/1)  
  *10)/10, comma6.1) ||' feet'));  
if distance.sum >=3 and distance.sum <16.5  
then xx = (right(put( int((distance.sum/1)  
  *10)/10, comma6.1) || ' yards'));  
if distance.sum >=16.5 and distance.sum <66  
then xx = (right(put( int((distance.sum/16.5)  
  *10)/10, comma6.1) || ' rods'));  
if distance.sum >=66 and distance.sum <5280  
then xx = (right(put( int((distance.sum/5280)  
  *10)/10, comma6.1) || ' miles'));  
endcomp;  
rbreak after / ol;  
run;

The table produced by this SAS code is given below. The cells  
are expressed with different units of measurement.

<table>
<thead>
<tr>
<th>Distance</th>
<th>(feet)</th>
<th>(Recoded)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.5</td>
<td>0.5 feet</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1.0 feet</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>6.0 yards</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>12.0 yards</td>
</tr>
<tr>
<td>5</td>
<td>62</td>
<td>3.7 rods</td>
</tr>
<tr>
<td>6</td>
<td>123</td>
<td>1.8 chains</td>
</tr>
<tr>
<td>7</td>
<td>617</td>
<td>9.3 chains</td>
</tr>
<tr>
<td>8</td>
<td>6,173</td>
<td>1.1 miles</td>
</tr>
<tr>
<td>9</td>
<td>12,345</td>
<td>2.3 miles</td>
</tr>
</tbody>
</table>

More complex compute statements can produce cells with  
combinations of units of measure such as 7 yards 2.1 feet.

Warning Flag
A common use of symbols in statistical tables is to alert readers  
to special situations or to warn users of the reliability problems  
in a cell. Some organizations require a warning flag when the  
sampling error or imputation rate is high. The cell symbol is  
conditional on the value of one or more other cells in the row. For  
imputation rate it could be the value of the number of records in  
the cell and the number of those records imputed. With  
conditional cell flags you almost always need to use a COMPUTE  
statement. An example of this type of PROC REPORT table is  
given below. In the table below, a flag is displayed if the  
imputation rate was greater than 50 percent.

proc report data=sugi128.virginia_pop nowd ;  
column county pct_imputed population pop ;  
define county / group;  
define population / analysis  
  format=comma15. "Population";  
define pct_imputed/ format=percent9.3;  
define pop / computed "Population";  
compute pop / char length=23;  
if percent_imputed.sum > .5 then  
  pop = (right(put(population.sum,  
    comma15.,) || '"!!' ));  
else pop = (right(put(population.sum,  
    comma15.,) || '"' ));  
endcomp;  
rbreak after / summarize ol;  
run;
This code produces the following table.

<table>
<thead>
<tr>
<th>COUNTY</th>
<th>Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9,160</td>
</tr>
<tr>
<td>2</td>
<td>5,752</td>
</tr>
<tr>
<td>3</td>
<td>1,902</td>
</tr>
<tr>
<td>4</td>
<td>1,484 !!</td>
</tr>
<tr>
<td>5</td>
<td>7,601</td>
</tr>
</tbody>
</table>

!! Percent Imputed greater than 50 percent

Warning flags displayed either before or after the number are used to alert the table reader to reliability problems with respect to complete or item nonresponse error, coverage error, or elevated sampling error. These conditional cell notes are similar to conditional notes used when a number is suppressed and only a sampling error. These conditional cell notes are similar to complete or item nonresponse error, coverage error, or elevated rounding to zero are examples of this. Examples of unconditional cell notes depends only on the value of cell. Missing values and unconditional cell notes. Unconditional means that the symbol displayed depends on the value of cell. Missing values and rounding to zero are examples of this. Examples of unconditional symbols are symbols based on the value of the cell such as a missing value, an extremely small value, and units of measurement based on the cell value. When the value of the cell is determined by the value of other associated cells is called a conditional cell note. The dividing line may get a little murky. Examples of unconditional cell notes are notes related to cell quality or reliability, whether they are meaningful, and other qualities.

### Commonly Used Cell Notes

<table>
<thead>
<tr>
<th>CellNote</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>NM</td>
<td>Not Meaningful</td>
</tr>
<tr>
<td>B</td>
<td>Base less than ____________</td>
</tr>
<tr>
<td>D</td>
<td>Data withheld to avoid Disclosure</td>
</tr>
<tr>
<td>NS</td>
<td>Not Significant</td>
</tr>
<tr>
<td>X</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>S</td>
<td>Data does not meet publication standards</td>
</tr>
<tr>
<td>-</td>
<td>Zero or rounds to zero</td>
</tr>
<tr>
<td>NA</td>
<td>Not Available</td>
</tr>
<tr>
<td>R</td>
<td>Revised</td>
</tr>
<tr>
<td>M</td>
<td>Missing</td>
</tr>
<tr>
<td>t</td>
<td>Less than one-half unit of measurement (trace)</td>
</tr>
</tbody>
</table>

A variety of symbols commonly appear in tables. Statistical groups that use many of these symbols in their tables include US Department of Agriculture, Census Bureau, Bureau of Economic Analysis, National Institute of Transportation Statistics, and the Environmental Protection Association.

### Missing Values

A common situation is for cells to have missing values. The general solution is often to use a simple user defined format. The format below replaces all missing values with the symbol (M) and most other cells with data having a "dollar12." format. A simplistic but often effective solution to missing values is to redefine the missing value symbol using the options statement. The default missing value symbol of a number is ".". The statement below redefines the missing value symbol to an "M" and any table cell with a missing value displays the symbol "M".

```proc format;
  value Revenu
    . = '('M')';
    0 = '0';
  .5 - .<5 = '('t')';
  0 = '0';
run;
```

The missing value symbol can also be incorporate by creating a simple user defined format. The code below inserts an "(M)" when the value is missing.

```proc format;
  value Revenu
    . = '('M')';
  END;
run;
```

### True Zero versus Round Zero

An important distinction to many users is whether a cell is missing, reported zero, or rounded to zero. A cell note is commonly used to make the distinction. The cell note "(t)" is used to represent values rounded to zero.

### Generic Format

A generic format that unconditionally creates a cell note for missing values, true zero, values rounded to zero, and all other values is given below.

```proc format;
  value Revenu
    . = '('M')';
    0 = '0';
  .5 = '<5';
  . = '('M')';
  END;
run;
```

### Conditional Cell Notes

Conditional cell notes depend on the values of other cells on the same row. The table below uses conditional cell notes to identify cell with firms considered reliable and a ratio that is not meaningful because the denominator is zero.

```
COUNTY  Populations  Revenue Expenses Ratio
1        27,534,999  19,982,298  1.38
2        33,464,667  (Z)       (NM)
3        32,705,477  17,814,355  1.84
4        33,248,724  15,279,846  2.18
5        32,534,999  19,982,298  1.38
```

(Z) -- Zero
(S) -- Suppressed because firms less than 15
(NM) -- Not Meaningful

The PROC REPORT code that produces the table is given below.

```proc report data = example nowd;
  header/ headline = "SUGI 28 Data Presentation"
  column/CTY CTY CTY CTY Firms Revenue Expenses R E RATIO ;
define CTY / group "County" ;
define Firms / ANALYSIS "Firms" ;
define Revenue / ANALYSIS "Revenue" ;
define Expenses / ANALYSIS "Expenses" ;
define R / COMPUTED "Revenue" ;
define E / COMPUTED "Expenses" ;
define ratio / COMPUTED "Revenue/Expenses" ;
compute R / CHAR LENGTH=13; R = LEFT( PUT(REVENUE.SUM , COMMA13.0));
compute E / CHAR LENGTH=13; E = LEFT( PUT(EXPENSES.SUM , COMMA13.0));
compute RATIO / CHAR LENGTH=13; RATIO=LEFT(PUT((REVENUE.SUM/EXPENSES.SUM) , COMMA7.2));
  if Firms.SUM < 15 then E = "(S)" ;
  if Firms.SUM < 15 then R = "(S)" ;
  if Firms.SUM < 15 then RATIO= "(NM)" ;
  if Firms.SUM < 15 then RATIO= "(S)" ;
  if Firms.SUM < 15 then RATIO= "(Z)" ;
  if Firms.SUM < 15 then RATIO= "(Z)" ;
  if Firms.SUM < 15 then RATIO= "(Z)" ;
run;
```

The PROC REPORT code that produces the table is given below.

```
proc report data = example nowd;
  header/ headline = "SUGI 28 Data Presentation"
  column/CTY CTY CTY CTY Firms Revenue Expenses R E RATIO ;
define CTY / group "County" ;
define Firms / ANALYSIS "Firms" ;
define Revenue / ANALYSIS "Revenue" ;
define Expenses / ANALYSIS "Expenses" ;
define R / COMPUTED "Revenue" ;
define E / COMPUTED "Expenses" ;
define ratio / COMPUTED "Revenue/Expenses" ;
compute R / CHAR LENGTH=13; R = LEFT( PUT(REVENUE.SUM , COMMA13.0));
compute E / CHAR LENGTH=13; E = LEFT( PUT(EXPENSES.SUM , COMMA13.0));
compute RATIO / CHAR LENGTH=13; RATIO=LEFT(PUT((REVENUE.SUM/EXPENSES.SUM) , COMMA7.2));
  if Firms.SUM < 15 then E = "(S)" ;
  if Firms.SUM < 15 then R = "(S)" ;
  if Firms.SUM < 15 then RATIO= "(NM)" ;
  if Firms.SUM < 15 then RATIO= "(S)" ;
  if Firms.SUM < 15 then RATIO= "(Z)" ;
  if Firms.SUM < 15 then RATIO= "(Z)" ;
  if Firms.SUM < 15 then RATIO= "(Z)" ;
run;
```
Conditional cell notes allows great flexibility in putting cell notes in tables.

FOOTNOTES
A statement or note inserted at the bottom (foot) of a table is a footnote or source note. Specific footnotes define symbols (S) or numbers (2) appearing usually in one or more cells. For tables that are entirely on one page, specific footnotes go at the bottom of the table. For tables that span multiple pages, a common practice is to put a standard message such as “See footnotes at the end of the table.” at the bottom of each page except the last and a complete listing of all footnotes at the bottom of the table on the last page. An example table is given below.

<table>
<thead>
<tr>
<th>dept</th>
<th>firms</th>
<th>revenue</th>
<th>expenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>2351</td>
<td>2823</td>
<td>3016</td>
</tr>
<tr>
<td>10</td>
<td>3086</td>
<td>1424</td>
<td>1823</td>
</tr>
<tr>
<td>11</td>
<td>2082</td>
<td>2269</td>
<td>2945</td>
</tr>
<tr>
<td>12</td>
<td>2060</td>
<td>2879</td>
<td>2300</td>
</tr>
</tbody>
</table>

See Notes at end of table
-------------------------------

The code below shows the use of the COMPUTE and LINE statement to add a specific footnote to a PROC REPORT table. The key is the use of the “after” and “_page_” options to the COMPUTE statement. The “after” option executes at the end of the table; The “after _page_” executes at the end of each page of the table.

```
proc report data=example nowd;
   column dept firms revenue expenses;
   define dept / group;
   define firms / analysis sum;
   define revenue / analysis sum format=bdat7.;
   define expenses / analysis sum;
   rbreak after / summarize;
   compute before;
      hold=' ';
   endcomp;
   compute after ;
      LINE @2 "(Z) Zero ";
      LINE @2 "(S) Suppressed";
      LINE @2 " ";
      Hold='x';
   endcomp;
   compute after _page_ / left ;
      line "See Notes at end of table ";
   endcomp;
run;
```

CONCLUSION
Formats and the PROC REPORT compute statements are excellent tools for add the necessary detail to make tables more readable, more usable, and more understandable.

REFERENCES
Guide to Tabular Presentation, Example of Table Structure, Statistical Abstract of the United State 2000, US Census Bureau, Page ix-x

DISCLAIMER
This paper reports the results of research and analysis undertaken by Census Bureau staff. It has undergone a more limited review by the Census Bureau than its official publications. This report is release to inform interested parties and to encourage discussion.

ACKNOWLEDGMENTS
I would like to acknowledge the help of the NESUG 15 - 2002 Code Clinic team (John Cohen, Elizabeth Axilrod, and Bob Virgil) in coming up with the idea of a one-to-one format to convert number to two significant digits.

CONTACT INFORMATION
Your comments and questions are encouraged. Contact the author at:
David D. Chapman, Chief
Frame Development Staff
Economic Statistical Methods and Programing Division
U.S. Census Bureau
Washington, DC 20233-6400
Work Phone: 301-457-4904
Fax: 301-457-1382
Email: david.d.chapman@census.gov