ADVANCED FEATURES OF PROC REPORT WORKSHOP

Kim L. Kolbe Ritzow
Systems Seminar Consultants, Kalamazoo, MI

Abstract
This workshop is designed for users who have an existing knowledge of PROC REPORT's syntax. The paper will focus on uses of the FLOW option, the creation of multi-column reports, customized report breaks, and creating reports containing counts and percentages.

Background
One of the main advantages of using PROC REPORT versus PROC PRINT for generating reports is that PROC REPORT provides much more flexibility and control over the appearance of the report. Prior to PROC REPORT most of the tasks discussed in this paper required a substantial amount of work in the Data Step prior to reporting, which now can all be done in a single SAS procedure.

The FLOW Option
The FLOW option on the DEFINE statement can be used to improve the appearance of reports that have data which require more than one row to display its values.

Without the FLOW option, truncation will result for any value that exceeds the width defined by the WIDTH= option:

```
PROC REPORT DATA=STOREDAT NOWINDOWS;
  COLUMNS STORE DEPT SALEAMT COMMENT;

DEFINE STORE/DISPLAY ORDER=FORMATTED
   'STORE';
DEFINE DEPT /DISPLAY ORDER=FORMATTED
   'DEPT';
DEFINE SALEAMT/ANAYLSIS SUM
   FORMAT=DOLLAR10.2 WIDTH=10
   'AMOUNT OF/SALE';
DEFINE COMMENT/DISPLAY WIDTH=10
   ORDER=FORMATTED 'COMMENT';
RUN;
```

(See Output #1 for results)

Short of providing a width large enough that will accommodate the value, but will also increase the amount of white space on the report, the FLOW option can be used which allows the column to flow on to another line each time it exceed the specified width without truncating the value:

```
PROC REPORT DATA=STOREDAT NOWINDOWS;
  COLUMNS STORE DEPT SALEAMT COMMENT;

DEFINE STORE/DISPLAY ORDER=FORMATTED
   'STORE';
DEFINE DEPT /DISPLAY ORDER=FORMATTED
   'DEPT';
DEFINE SALEAMT/ANAYLSIS SUM
   FORMAT=DOLLAR10.2 WIDTH=10
   'AMOUNT OF/SALE';
DEFINE COMMENT/DISPLAY WIDTH=10 FLOW
   ORDER=FORMATTED 'COMMENT';
RUN;
```

(See Output #2 for results)

The FLOW option may split a word within the column when flowing the text. Unfortunately, there is no way to prevent this from happening.

It is also possible to dynamically create a comment on a report by creating them as a COMPUTED variable with accompanying logic:

```
PROC REPORT DATA=STOREDAT NOWINDOWS;
  COLUMNS STORE DEPT SALEAMT SALECOMM;

DEFINE STORE/GROUP ORDER=FORMATTED
   'STORE';
DEFINE DEPT /GROUP ORDER=FORMATTED
   'DEPT';
DEFINE SALEAMT/ANALYSIS SUM
   FORMAT=DOLLAR10.2 WIDTH=10
   'AMOUNT OF/SALE';
DEFINE SALECOMM/COMPUTED FLOW
   WIDTH=10 ORDER=FORMATTED
   'SALES/COMMENT';

COMPUTE SALECOMM/CHAR LENGTH=30;
 IF SALEAMT.SUM > 1000 AND
    DEPT IN (100, 110,150) THEN
    SALECOMM='EXCEEDED DAILY SALES
    GOAL';
 ELSE IF SALEAMT.SUM < 1000 AND
    DEPT IN (100, 110,150) THEN
    SALECOMM='AT OR BELOW DAILY SALES
    GOAL';
 ELSE SALECOMM='NOT MONITORING AT THIS
    TIME';
ENDCOMP;
```

(See Output #3 for results)

The PANELS Option

Creating multi-column reports is very easy within PROC REPORT by using the PANELS= option on the PROC statement. This is especially useful if you want a list of something like store numbers, zip codes or phone numbers. A lot of paper would be wasted to print a small amount of data down the left-hand side of each page:

```
PROC REPORT DATA=STOREINF NOWINDOWS;
  COLUMNS STORE;

DEFINE STORE/DISPLAY ORDER=FORMATTED
  'STORE/NUMBERS';
RUN;
```

(See Output #4 for results)

The PANELS= option will create a multi-column report based on the number of panels specified. This is sometimes referred to as phonebook style report:

```
PROC REPORT DATA=STOREDAT NOWINDOWS
     PANELS=2;
  COLUMNS STORE;

DEFINE STORE/ORDER ORDER=FORMATTED
   'Store/Numbers';
```

(See Output #5 for results)

By specifying PANELS=99, PROC REPORT will print as many columns as will fit across the page before going on to a new page:

```
PROC REPORT DATA=STOREDAT NOWINDOWS
   PANELS=99;
  COLUMNS STORE;

DEFINE STORE/ORDER ORDER=FORMATTED
   'Store/Numbers';
```

(See Output #6 for results)

Compute Blocks

Compute Blocks are essential in creating more sophisticated reports within PROC REPORT. They support a number of SAS language features:

- all the DATA step functions
- comments
- A number of DATA step statements:

assignment statements	LENGTH
CALL routines	LINK
DO (all varieties)	RETURN
END	SELECT
IF THEN/ELSE	GO TO
sum statement	

- DM statement
- %INCLUDE statement
- macro variables
- null statements

The Various Uses of the COMPUTE Block

COMPUTE Blocks are used most often to dynamically create new variables or to create customized report breaks.

The following is an example of how a COMPUTE Block can be used to create customized report breaks:

```
PROC FORMAT;
  VALUE $STORNAM
   '101010'='NORFOLK STORE'
   '101110'='SOUTHDALE STORE';
RUN;
```

(the entire format is not shown here)

```
PROC REPORT DATA=STOREDAT
   NOWINDOWS HEADLINE HEADSKIP;

COLUMNS  STORE DEPT SALEAMT TAX NET;

DEFINE STORE/GROUP ORDER=FORMATTED
   'STORE';
DEFINE DEPT/GROUP 'DEPT';
DEFINE SALEAMT/ANALYSIS SUM
   FORMAT=DOLLAR10.2 WIDTH=10
   'AMOUNT/OF SALE';
DEFINE TAX/COMPUTED FORMAT=DOLLAR10.2
   WIDTH=10 'SALES/TAX';
DEFINE NET/COMPUTED FORMAT=DOLLAR10.2
   WIDTH=10 'NET/SALE';

COMPUTE TAX;
  TAX=SALEAMT.SUM * .05;
ENDCOMP;

COMPUTE NET;
  NET=SALEAMT.SUM - TAX;
```

```
ENDCOMP;

COMPUTE AFTER STORE;
  LINE ' ';
  LINE 52*'*';
  LINE ' ';
  LINE '* TOTAL SALES FOR ' STORE $STORNAM.
      ' WERE: ' SALEAMT.SUM DOLLAR10.2 ' *';
  LINE ' ';
  LINE 52*'*';
  LINE ' ';
ENDCOMP;
RUN;
```

(see Output #7 for results)

Since we are creating customized break text, the BREAK statement is not needed. The COMPUTE Block will control the entire breaking and text writing process.

Just like when PUTting literal text (anything within quotes) we are in control of the entire process, so if blanks within the literal strings are forgotten, text will run together (notice how blanks were left in the literal strings before the value of variables were written out). Similar to the feature available on PUT statement, multipliers (52* '*') can be specified which will print the character '*' fifty-two times.

If a location is not specified with a pointer reference (@), SAS will center the text in the COMPUTE block if the CENTER system option is in effect.

If you happen to have a BREAK statement in effect with a SUPPRESS option specified in addition to the COMPUTE block (you shouldn't really need BREAK statement), the values for STORE will not appear on the report. If for some reason a BREAK statement is needed in addition to a COMPUTE block DO NOT USE THE SUPPRESS OPTION ON THE BREAK STATEMENT.

Notice on the output generated that the format used when writing out values on the LINE statement determines the amount of space used to write its values. Character variables will fill the space with trailing blanks, whereas numeric variables fill the space with leading blanks.

Controlling Space and Grand Totals

The trailing blanks on character variables and leading blanks on numeric variables can be controlled by using the $VARYING. format on the LINE statement. In addition to using the $VARYING. format, assignment statements are needed in the COMPUTE blocks to convert values and to check lengths:

```
PROC FORMAT;
  VALUE $STORNAM
  '101010'='NORFOLK STORE'
  '101110'='SOUTHDALE STORE';
RUN;
```

(the entire format is not shown here)

```
PROC REPORT DATA=STOREDAT NOWINDOWS
  HEADLINE HEADSKIP;

COLUMNS STORE DEPT SALEAMT TAX NET;

DEFINE STORE/GROUP ORDER=FORMATTED
   'STORE';
DEFINE DEPT/GROUP 'DEPT';
DEFINE SALEAMT/ANALYSIS SUM
   FORMAT=DOLLAR10.2
   WIDTH=10 'AMOUNT/OF SALE';
DEFINE TAX/COMPUTED FORMAT=DOLLAR10.2
   WIDTH=10 'SALES/TAX';
DEFINE NET/COMPUTED FORMAT=DOLLAR10.2
   WIDTH=10 'NET/SALE';

COMPUTE TAX;
  TAX=SALEAMT.SUM * .05;
ENDCOMP;

COMPUTE NET;
  NET=SALEAMT.SUM - TAX;
ENDCOMP;

COMPUTE AFTER STORE;
  STOREC=PUT(STORE,$STORNAM.);
  VARLEN1=LENGTH(STOREC);

SALESC=LEFT(PUT(SALEAMT.SUM,DOLLAR10.2));
  VARLEN2=LENGTH(SALESC);
  LINE ' ';
  LINE @7 52*'*';
  LINE ' ';
  LINE @7 '* TOTAL SALES FOR '
      STOREC $VARYING. VARLEN1 ' WERE: '
      SALESC $VARYING. VARLEN2 @57 ' *';
  LINE ' ';
  LINE @7 52*'*';
  LINE ' ';
ENDCOMP;
RUN;
```

(see Output #8 for results)

Since the character variable REGION is actually a

formatted value, in addition to checking its length with the LENGTH function, the format's value needs to be associated with the variable. This is done by using the PUT function. It could have been done all in one step:

```
VARLEN1=LENGTH(PUT(STORE,$STORNAM.));
```

If the character variable was a normal unformatted character variable, the PUT would not be necessary, only the LENGTH function would be required. Since the $VARYING. format can only be used on character variables, to control the spacing on a numeric variable it must first be converted to a character using the LEFT and PUT functions and then find its length with the LENGTH function. ALL NUMERIC VARIABLES MUST BE CONVERTED TO CHARACTER, THE LENGTH STATEMENT ALONE IS NOT ENOUGH. Similarly this could have all been done in one step:

```
VARLEN2=LENGTH(LEFT(PUT(SALEAMT.SUM,DOLLAR10.2)));
```

Since the text will now vary in length (it will not be padded out with spaces after characters or before numerics), the length of the text string will vary. Therefore, the location of the '*' literal with a pointer references will have to be fixed to a location (like @7) when writing out the text line. Once a pointer reference (@) is used anywhere within the COMPUTE block the centering option, if in effect, is turned off for the entire COMPUTE block, that is why the location of the '*' string (@7 52* '*') was also fixed. If left as it was in the last example (52* '*'), the string of asterisks would be left aligned because the centering is automatically turned off.

Adding Counts in Customized Breaks

Counts can be added in customized breaks by using the N statistic defined on the COLUMNS statement (if this style of referencing does not look familiar to you, reference my paper on Advanced Features of PROC REPORT presented at SUGI '94, and MWSUG '95 conferences):

```
PROC FORMAT;
  VALUE $STORNAM
  '101010'='NORFOLK STORE'
  '101110'='SOUTHDALE STORE'
RUN;
PROC REPORT DATA=STOREDAT NOWINDOWS
    HEADLINE HEADSKIP;
COLUMNS STORE DEPT SALEAMT TAX NET N;
DEFINE STORE/GROUP ORDER=FORMATTED
   'STORE';
DEFINE DEPT/GROUP 'DEPT';
DEFINE SALEAMT/ANALYSIS SUM
   FORMAT=DOLLAR10.2 WIDTH=10
   'AMOUNT OF/SALE';
DEFINE TAX/COMPUTED FORMAT=DOLLAR10.2
   WIDTH=10 'SALES/TAX';
DEFINE NET/COMPUTED FORMAT=DOLLAR10.2
   WIDTH=10 'NET/SALE';
DEFINE N/NOPRINT ;

COMPUTE TAX;
  TAX=SALEAMT.SUM * .05;
ENDCOMP;

COMPUTE NET;
  NET=SALEAMT.SUM - TAX;
ENDCOMP;

COMPUTE AFTER STORE;
  STOREC=PUT(STORE, $STORNAM.);
  VARLEN1=LENGTH(STOREC);
SALESC=LEFT(PUT(SALEAMT.SUM,DOLLAR10.2));
  VARLEN2=LENGTH(SALESC);
  LINE ' ';
  LINE @7 52*'*';
  LINE ' ';
  LINE @7 '* TOTAL SALES FOR ' STOREC
      $VARYING. VARLEN1
      ' WERE: ' SALESC $VARYING. VARLEN2
      @57 ' *';
  LINE @7 '* TOTAL NUMBER OF SALES WERE: '
      N COMMA10. @57 ' *';
  LINE ' ';
  LINE @7 52*'*';
  LINE ' ';
ENDCOMP;
RUN;
```

(see output #9 for results)

While it may look odd that the total number of sales are 43 and 35 when only 4 and 3 lines of data are printing, respectively. This IS correct. Keep in mind that grouping is in effect on this report. Grouping causes only one summarized line to appear for each group value, but the count reflects how many total observations the grouping represents.

Calculating Percentages

To calculate a percentage you need to first know the total so you can divide each value by the total in order to arrive at the percentage. This is accomplished in PROC REPORT by using a COMPUTE Block to accumulate the total before each group:

```
PROC REPORT DATA=STOREDAT NOWINDOWS;
COLUMNS STORE DEPT SALEAMT PCTSALES;
```

```
DEFINE STORE/GROUP ORDER=FORMATTED
  'STORE';
DEFINE DEPT/GROUP ORDER=FORMATTED
  'DEPT';
DEFINE SALEAMT/ANALYSIS SUM
  FORMAT=DOLLAR10.2 WIDTH=10
  'AMOUNT OF/SALE';
DEFINE PCTSALES/COMPUTED
  FORMAT=PERCENT7.2 'PERCENT/OF/SALES';

COMPUTE BEFORE STORE;
 STORETOT=SALEAMT.SUM;
ENDCOMP;

COMPUTE PCTSALES;
 PCTSALES=SALEAMT.SUM/STORETOT;
ENDCOMP;

BREAK AFTER STORE/SKIP SUMMARIZE;
RUN;
```

(See Output #10 for results)

Note, if the PERCENT format is used to display the value, you do not need to divide by 100.

Rather than each group adding up to 100 percent, you can also calculate a percentage based on what percent the group represents of the whole by using a COMPUTE BEFORE without a specified variable:

```
PROC REPORT DATA=STOREDAT NOWINDOWS;
COLUMNS STORE DEPT SALEAMT PCTSALES;

DEFINE STORE/GROUP ORDER=FORMATTED
  'STORE';

DEFINE DEPT/GROUP ORDER=FORMATTED
  'DEPT';
DEFINE SALEAMT/ANALYSIS SUM
  FORMAT=DOLLAR10.2 WIDTH=10
  'AMOUNT OF/SALE';
DEFINE PCTSALES/COMPUTED
  FORMAT=PERCENT7.2 'PERCENT/OF/SALES';

COMPUTE BEFORE;
 STORETOT=SALEAMT.SUM;
ENDCOMP;

COMPUTE PCTSALES;
 PCTSALES=SALEAMT.SUM/STORETOT;
ENDCOMP;

BREAK AFTER STORE/SKIP SUMMARIZE ;
RBREAK AFTER/SKIP SUMMARIZE;
RUN;
```

(See Output #11 for results)

Cumulative percentages are easy to calculate on PROC REPORT, all you need to do is add some additional logic within the COMPUTE Block to accumulate information:

```
PROC REPORT DATA=STOREDAT NOWINDOWS;
COLUMNS STORE DEPT SALEAMT PCTSALES
  CUMPCT;

DEFINE STORE/GROUP ORDER=FORMATTED
  'STORE';
DEFINE DEPT/GROUP ORDER=FORMATTED
  'DEPT';
DEFINE SALEAMT/ANALYSIS SUM
  FORMAT=DOLLAR10.2 WIDTH=10
  'AMOUNT OF/SALE';
DEFINE PCTSALES/COMPUTED
  FORMAT=PERCENT7.2 'PERCENT/OF/SALES';
DEFINE CUMPCT/COMPUTED
  FORMAT=PERCENT7.2 WIDTH=10
  'CUMULATIVE/PERCENT/OF/SALES';

COMPUTE BEFORE STORE;
 STORETOT=SALEAMT.SUM;
 CUMTOT=0;
ENDCOMP;


COMPUTE PCTSALES;
 PCTSALES=SALEAMT.SUM/STORETOT;
 IF DEPT NE ' ' THEN CUMTOT+PCTSALES;
ENDCOMP;

COMPUTE CUMPCT;
 CUMPCT=CUMTOT;
ENDCOMP;

BREAK AFTER STORE/SKIP SUMMARIZE ;
RBREAK AFTER/SKIP SUMMARIZE;
RUN;
```

(See Output #12 for results)

The Percent of Sales appearing at the bottom of the report appears to be incorrect, but it IS correct, given what you are asking PROC REPORT to do. At the RBREAK level PCTSALES is calculated as: SALEAMT.SUM/STORETOT. For this row, the value of SALEAMT.SUM is $9,920 (sales for everyone). STORETOT is a data step variable whose value only changes when it is explicitly changed. At the RBREAK level, the value of STORETOT is $4,400 (from the

previous store) so: 225=9920/4400 * 100 (multiplied by 100 because the SAS format PERCENT is being used, which automatically displays the result multiplied by 100).

Solving the Previous Problem

We can suppress the display of this column by changing some code and using a user-defined format for displaying the percentage value:

```
PROC FORMAT;
  PICTURE PCT LOW-100='009.9%'
            101-HIGH=' ';
RUN;

PROC REPORT DATA=STOREDAT NOWINDOWS;
COLUMNS STORE DEPT SALEAMT PCTSALES
   CUMPCT;

DEFINE STORE/GROUP ORDER=FORMATTED
   'STORE';
DEFINE DEPT/GROUP 'DEPT';
DEFINE SALEAMT/ANALYSIS SUM
   FORMAT=DOLLAR10.2 WIDTH=10
   'AMOUNT OF/SALE';
DEFINE PCTSALES/COMPUTED FORMAT=PCT7.2
   WIDTH=7 'PERCENT/OF/SALES';


DEFINE CUMPCT/COMPUTED
   FORMAT=PERCENT7.2 WIDTH=10
   'CUMULATIVE/PERCENT/OF/SALES';

COMPUTE BEFORE STORE;
 STORETOT=SALEAMT.SUM;
 CUMTOT=0;
ENDCOMP;

COMPUTE PCTSALES;
  PCTSALES=SALEAMT.SUM/STORETOT * 100;
  PCTSALE=SALEAMT.SUM/STORETOT;
  IF DEPT NE ' ' THEN CUMTOT+PCTSALE;
ENDCOMP;

COMPUTE CUMPCT;
  CUMPCT=CUMTOT;
ENDCOMP;

BREAK AFTER STORE/SKIP SUMMARIZE;
RBREAK AFTER/SKIP SUMMARIZE;
RUN;
```

(see output #13 for results)

The user-defined format was used to suppress the display of the bogus percent value. Because this format is used rather than the SAS-defined PERCENT format, the COMPUTE block that calculates PCTSALES has to be changed as well in order to multiply the results by 100 and then to create an additional variable that does multiply by 100 so that it can be used for accumulation.

In Summary

PROC REPORT is a powerful new Base SAS Procedure which provides us with more control over our report's appearance than any other existing Base SAS Proc.

We have seen in the course of this paper how PROC REPORT can create customized report breaks, multi-column reports containing counts and percents and some uses of the FLOW option.

While not discussed in this presentation some new features will be available in Version 6.11 like the BOX and ID options and new statistics like PCTN and PCTSUM.

Trademark Notice

Useful Publications

SAS Institute Inc. (1995), SAS Software Changes and Enhancements Release 6.11, Cary, NC: SAS Institute

Kolbe Ritzow, Kim (1995), *"More Advanced Features of PROC REPORT"*, Proceedings of the 8th Annual NorthEast SAS Users Group Conference

Yindra, Chris (1995), *"Advanced Features of PROC REPORT in Batch Mode"*, Proceedings of the 8th Annual NorthEast SAS Users Group Conference

Kolbe Ritzow, Kim (1995), *"An Introduction to PROC REPORT"*, Proceedings of the 6th Annual MidWest SAS Users Group Conference

Kolbe Ritzow, Kim (1995), *"Advanced Features of PROC REPORT"*, Proceedings of the 20th Annual SUGI Conference and Proceedings of the 6th Annual MidWest SAS Users Group Conference.

SAS Institute Inc. (1993), SAS® Technical Report P-258, Using the REPORT Procedure in a Nonwindowing Environment, Release 6.07, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990), SAS® Guide to the Report Procedure, Usage and Reference, Version 6, First Edition, Cary, NC: SAS Institute Inc.

Any questions or comments regarding the paper may be directed to the author:

Kim L. Kolbe Ritzow
Systems Seminar Consultants
Kalamazoo Office
927 Lakeway Drive
Kalamazoo, MI 49001
Phone: (616) 345-6636
Fax: (616) 345-5793
E-mail: KRITZOW@AOL.COM
(or) KRITZOW@MCIMAIL.COM

Output #1 (Partial Output):

STORE	DEPT	AMOUNT OF SALE	COMMENT
101010	100	$50.00	COUPONS #8
101010	100	$60.00	COUPONS #1
101010	100	$60.00	COUPONS #3
101010	100	$60.00	COUPONS #8
101010	100	$70.00	NO COUPONS
.....	ETC	.....	

Output #2 (Partial Output):

STORE	DEPT	AMOUNT OF SALE	COMMENT
101010	100	$50.00	COUPONS #8,11,20,21 USED
101010	100	$60.00	COUPONS #10,20 & 21 USED
101010	100	$60.00	COUPONS #3,5,15,14 USED
101010	100	$60.00	COUPONS #8,11,20,21 USED
.....	ETC	.....	

Output #3 (Partial Output):

STORE	DEPT	AMOUNT OF SALE	SALES COMMENT
101010	100	$1,020.00	EXCEEDED DAILY SALES GOAL
	110	$1,290.00	EXCEEDED DAILY SALES GOAL
	120	$1,550.00	NOT
.....	ETC	.....	

Output #4 (Partial Output):

STORE NUMBERS
101010
101030
101050
101070
101090
... ETC ...

Output #5 (Partial Output):

```
      STORE         STORE
    NUMBERS       NUMBERS
     101010        102170
     101030        102190
     101050        102210
     101070        102230
     101090        102250
    ........ ETC .......
```

Output #6 (Partial Output):

```
      STORE         STORE         STORE         STORE
    NUMBERS       NUMBERS       NUMBERS       NUMBERS
     101010        102170        103330        104490
     101030        102190        103350        104510
     101050        102210        103370        104530
     101070        102230        103390        104550
     101090        102250        103410        104570
     ..................... ETC .....................
```

Output #7 (Partial Output):

```
                      Amount of       Sales         Net
Store     Dept             Sale         Tax        Sale
-------------------------------------------------------

101010    100        $1,020.00      $51.00     $969.00
          110        $1,290.00      $64.50   $1,225.50
          120        $1,550.00      $77.50   $1,472.50
          130        $1,660.00      $83.00   $1,577.00

 ************************************************

 * total sales for Norfolk Store      were:  $5,520.00 *

 ************************************************
 ....................... etc ........................
```

Output #8 (Partial Output):

```
                        Amount       Sales         Net
 Store        Dept     of Sale         Tax        Sale
 ---------------------------------------------------

101010        100   $1,020.00      $51.00     $969.00
              110   $1,290.00      $64.50   $1,225.50
              120   $1,550.00      $77.50   $1,472.50
              130   $1,660.00      $83.00   $1,577.00
              140   $1,800.00      $90.00   $1,710.00

 ***************************************************

 * total sales for Norfolk Store were: $18,020.00   *

 ***************************************************
..................... etc .......................
```

Output #9 (Partial Output):

```
                      Amount of       Sales         Net
 Store     Dept            Sale         Tax        Sale
 ---------------------------------------------------

 101010    100        $1,020.00      $51.00     $969.00
           110        $1,290.00      $64.50   $1,225.50
           120        $1,550.00      $77.50   $1,472.50
           130        $1,660.00      $83.00   $1,577.00

  ***************************************************

  * total sales for Norfolk Store were: $5,520.00    *
  * total number of sales were:         43           *

  ***************************************************
..................... etc .......................
```

Output #10 (Partial Output):

```
                                      Percent
                          Amount of       of
     Store     Dept            Sale    Sales
     101010    100        $1,020.00    18.5%
               110        $1,290.00    23.4%
               120        $1,550.00    28.1%
               130        $1,660.00    30.1%
     101010               $5,520.00     100%
     ............... etc ...................
```

Output #11:

Store	Dept	Amount of Sale	Percent of Sales
101010	100	$1,020.00	10.3%
	110	$1,290.00	13.0%
	120	$1,550.00	15.6%
	130	$1,660.00	16.7%
101010		$5,520.00	55.6%
101110	100	$1,380.00	13.9%
	110	$1,440.00	14.5%
	120	$1,580.00	15.9%
101110		$4,400.00	44.4%
		$9,920.00	100%

Output #12:

Store	Dept	Amount of Sale	Percent of Sales	Cumulative Percent of Sales
101010	100	$1,020.00	18.5%	18.5%
	110	$1,290.00	23.4%	41.8%
	120	$1,550.00	28.1%	69.9%
	130	$1,660.00	30.1%	100%
101010		$5,520.00	100%	100%
101110	100	$1,380.00	31.4%	31.4%
	110	$1,440.00	32.7%	64.1%
	120	$1,580.00	35.9%	100%
101110		$4,400.00	100%	100%
		$9,920.00	225%	100%

Output #13:

Store	Dept	Amount of Sale	Percent of Sales	Cumulative Percent of Sales
101010	100	$1,020.00	18.4%	18.5%
	110	$1,290.00	23.3%	41.8%
	120	$1,550.00	28.0%	69.9%
	130	$1,660.00	30.0%	100%
101010		$5,520.00	100.0%	100%
101110	100	$1,380.00	31.3%	31.4%
	110	$1,440.00	32.7%	64.1%
	120	$1,580.00	35.9%	100%
101110		$4,400.00	100.0%	100%
		$9,920.00		100%