

Making Sense of Proc Tabulate

Jonas V. Bilenas
JP Morgan Chase
Wilmington, DE



Introduction

- The TABULATE procedure is a powerful procedure to produce customized tabular reports.
 - Reports generate summary statistics that are available in PROC FREQ, PROC MEANS, and additional summaries that are unique to PROC TABULATE.
- However, the TABULATE procedure can be very complex looking for the beginning TABULATE programmer.
- One reason for the complexity of the syntax is that it was based on a table generation code developed at U.S. Department of Labor.
- This tutorial will demystify PROC TABULATE and provide a framework to learn PROC TABULATE

Examples Used in Presentation

- **This presentation will look at simulated data that maybe typical of data used in a credit card Decision Science Division.**
 - **Generate a Performance Report for a Direct Mail Balance Transfer Offer mailed to existing credit card accounts.**
 - **Generate Approval/Decline Reports for Take-One Application processing.**
 - **If there is time we will look at an additional example:**
 - **Get tabulate to calculate LOGS and Column Differences**

DATA for Balance Transfer Mailing

```
proc format;
  value offer low    - 0.45 = 'A'
              0.45< - high = 'B';
  value $grroff 'A' = 'aoff'  'B' = 'boff';
  value aoff  low-0.10 = '1' other = '0';
  value boff  low-0.05 = '1' other = '0';
  value $mline 'A' = '6500'  'B' = '5400';
run;

data test;
  do campaign = '2004/3', '2004/4'; /* campaign quarters */
    do i = 1 to 1e6;
      mailed=1;
      offer=put(ranuni(12),offer.);
      fmtuse=put(offer,$grroff.);
      respond=input(putn(ranuni(14),fmtuse),best12.);
      if respond then
        baltran=rannor(15)*500+input(put(offer,$mline.),best12.);
      else baltran=.;
      output;
    end;
  end;
run;
```

Nine Steps to Tabulate Bliss

1. Don't panic.
2. Design the report on paper.
3. Generate the initial test code.
4. Test, retest and verify using a small sample.
5. Clean up the appearance of the report.
6. Run code with OBS=MAX if testing on a sample data set.
7. Add some ODS functionality.
8. Need to generate multi-label formats?
9. Sit back, smile and be proud of your report.

Step 1: Don't panic!

- Relax
- Meditate
- Listen to some music
- Have a snack

Step 2: Design the Report on Paper

Campaign	Offer	Mail Base	%	Respond	Response Rate	Average BT per responder	Average BT per mail base
2004/3	A						
	B						
	TOTAL						
2004/4	A						
	B						
	TOTAL						

Step 3 - Generate the initial test code

Some Syntax Rules:

- CLASS: categories (numeric or character variables)
- VAR: analysis variables (numeric variables)
- TABLE: Specifies table design.
 - Table dimensions (page, row, column) are separated by commas.
 - * : Use to specify any of the following:
 - Another CLASS variable Split
 - A VAR variable
 - A statistic
 - A format
 - <> : Specify the denominator CLASS dimensions for PCTN or to specify denominator VAR variable or CLASS dimensions when using the PCTSUM statistic.
 - () : To group CLASS, VAR, and/or statistics.
 - = : For format and/or label specification

Build the Code in Stages

Test code with small number of records and few initial elements. Let's try to add mail base and responder frequency counts:

```
proc tabulate data=test;
  class offer campaign;
  var respond;
  table campaign*offer
        ,
        n='Mailed'*f=comma9
        respond*f=comma9.
;
run;
```

DEFAULT Statistic for a
VAR variable is SUM

		respond	
		Mailed	Sum
campaign	offer		
2004/3	A	450,137	45,288
	B	549,863	27,748
2004/4	A	449,678	45,082
	B	550,322	27,484

Step 4 - Test, Retest, and Verify using Small Sample

Let's add some totals. This is done with the ALL qualifier in the TABLE statement:

```
proc tabulate data=test;
  class offer campaign;
  var respond;
  table campaign*(offer all)
        ,
        n='Mailed'*f=comma9.
        respond*f=comma9.
;
run;
```

		respond	
		Mailed	Sum
campaign	offer		
2004/3	A	450,137	45,288
	B	549,863	27,748
	All	1,000,000	73,036
2004/4	offer		
	A	449,678	45,082
	B	550,322	27,484
	All	1,000,000	72,566

TEST:

- (campaign all)*offer
- (campaign all)*(offer all)
- campaign *offer all

ALL Party

table campaign*(offer all)

		Mailed	respond
			Sum
campaign	offer		
2004/3	A	450,137	45,288
	B	549,863	27,748
	All	1,000,000	73,036
2004/4	offer		
	A	449,678	45,082
	B	550,322	27,484
	All	1,000,000	72,566

table (campaign all)*offer

		Mailed	respond
			Sum
campaign	offer		
2004/3	A	450,137	45,288
	B	549,863	27,748
2004/4	A	449,678	45,082
	B	550,322	27,484
All	A	899,815	90,370
	B	1,100,185	55,232

Additional tests of ALL

table (campaign all)*(offer all)

		Mailed	respond
			Sum
campaign	offer		
2004/3	A	450,137	45,288
	B	549,863	27,748
	All	1,000,000	73,036
2004/4	offer		
	A	449,678	45,082
	B	550,322	27,484
	All	1,000,000	72,566
All	offer		
	A	899,815	90,370
	B	1,100,185	55,232
	All	2,000,000	145,602

table all campaign*offer

		Mailed	respond
			Sum
campaign	offer		
All		2,000,000	145,602
2004/3	A	450,137	45,288
	B	549,863	27,748
	All		
2004/4	A	449,678	45,082
	B	550,322	27,484

Add Percentage Statistics with PCTN

Percentage of mail base can be added using PCTN statistic. It requires a denominator definition inside “<>”. The definition specifies CLASS variables used for calculating percentages:

```
proc tabulate data=test;
  class offer campaign;
  var respond;
  table campaign*(offer all)
        ,
        n='Mailed'*f=comma9.
        pctn<offer all>
        respond*f=comma9.
        ;
run;
```

				respond
		Mailed	PctN	Sum
campaign	offer			
2004/3	A	450,137	45.01	45,288
	B	549,863	54.99	27,748
	All	1,000,000	100.00	73,036
2004/4	offer			
	A	449,678	44.97	45,082
	B	550,322	55.03	27,484
	All	1,000,000	100.00	72,566

Don't be Afraid of PCTN

Test and verify results with different PCTN configurations:

Change statement **pctn<offer all>**

PCTN

		Mailed	PctN	respond
				Sum
campaign	offer			
2004/3	A	450,137	22.51	45,288
	B	549,863	27.49	27,748
	All	1,000,000	50.00	73,036
2004/4	offer			
	A	449,678	22.48	45,082
	B	550,322	27.52	27,484
	All	1,000,000	50.00	72,566

PCTN<campaign>

		Mailed	PctN	respond
				Sum
campaign	offer			
2004/3	A	450,137	50.03	45,288
	B	549,863	49.98	27,748
	All	1,000,000	50.00	73,036
2004/4	offer			
	A	449,678	49.97	45,082
	B	550,322	50.02	27,484
	All	1,000,000	50.00	72,566

PCTN<campaign all> and

PCTN<campaign offer> give same results as above

Continue Adding Components of Table

Let us add some MEAN statistics for Response Rate and Average Balance Transfer:

```
proc tabulate data=test;
  class offer campaign;
  var respond baltran;
  table campaign*(offer all)
          ,
          n='Mailed'*f=comma9.
          pctn<offer all>
          respond*f=comma9.
          respond*mean*f=percent9.2
          baltran*mean*f=dollar9.
          ;
run;
```

				respond	respond	baltran
		Mailed	PctN	Sum	Mean	Mean
campaign	offer					
2004/3	A	450,137	45.01	45,288	10.06%	\$6,498
	B	549,863	54.99	27,748	5.05%	\$5,402
	All	1,000,000	100.00	73,036	7.30%	\$6,082
2004/4	offer					
	A	449,678	44.97	45,082	10.03%	\$6,504
	B	550,322	55.03	27,484	4.99%	\$5,400
	All	1,000,000	100.00	72,566	7.26%	\$6,086

Add Percentage Statistics with PCTSUM

```
proc tabulate data=test;
  class offer campaign;
  var respond baltran mailed;
  table campaign*(offer all)
    ,
    n='Mailed'*f=comma9.
    pctn<offer all>
    respond*f=comma9.
    respond*mean*f=percent9.2
    baltran*mean*f=dollar9.
    baltran*pctsum<mailed>*f=btm.
  ;
run;
```

				respond	respond	baltran	baltran
		Mailed	PctN	Sum	Mean	Mean	PctSum
campaign	offer						
2004/3	A	450,137	45.01	45,288	10.06%	\$6,498	\$654
	B	549,863	54.99	27,748	5.05%	\$5,402	\$273
	All	1,000,000	100.00	73,036	7.30%	\$6,082	\$444
2004/4	offer						
	A	449,678	44.97	45,082	10.03%	\$6,504	\$652
	B	550,322	55.03	27,484	4.99%	\$5,400	\$270
	All	1,000,000	100.00	72,566	7.26%	\$6,086	\$442

Step 5 - Clean up Appearance of the Report

- Eliminate some of the statistic labels and/or replace with descriptive labels.
- Add percentage sign to PTCN and/or PCTSUM statistics if required.
- Remove some lines.
- Add a missing label.

Step 5 (continued)

```
proc format;
  picture btm (round) low - high = '0,000,009' (prefix='$' mult=.01)
  ;
  picture pct (round) low - < 0 = '0009.99%' (prefix='- ' )
        0 - high = '0009.99%'
  ;
proc tabulate data=test noseps;
  class offer campaign;
  var respond baltran mailed;
  keylabel n=' ' sum=' ' mean=' ' pctn=' ' pctsum=' ' ;
  table campaign*(offer all=' TOTAL ' )
        ,
        n='Mailed' *f=comma9.
        pctn<offer all>='%' *f=pct.
        respond='Responders' *f=comma10.
        respond='Response Rate' *mean*f=percent9.2
        baltran='Balance Transfer per respond' *mean*f=dollar9.0
        baltran=' Balance Transfer per mailed' *pctsum<mailed>*f=btm.
        /rts=19 row=float misstext=' ' box=' PNWSUG ' ;
run;
```

Step 5 (Output)

PNWSUG		Mailed	%	Responders	Response Rate	Balance Transfer per respond	Balance Transfer per mailed
campaign offer							
2004/3	A	450,137	45.01%	45,288	10.06%	\$6,498	\$654
	B	549,863	54.99%	27,748	5.05%	\$5,402	\$273
	TOTAL	1,000,000	100.00%	73,036	7.30%	\$6,082	\$444
2004/4	offer						
	A	449,678	44.97%	45,082	10.03%	\$6,504	\$652
	B	550,322	55.03%	27,484	4.99%	\$5,400	\$270
	TOTAL	1,000,000	100.00%	72,566	7.26%	\$6,086	\$442

Try a PAGE Dimension

```
proc tabulate data=test nosepts;
  class offer campaign;
  var respond baltran mailed;
  keylabel n=' ' sum=' ' mean=' ' pctn=' ' pctsum=' ' ;
  table campaign

      ,
  (offer all='TOTAL' )
      ,
  n='Mailed' *f=comma9.
  pctn<offer all>='%' *f=pct.
  respond='Responders' *f=comma10.
  respond='Response Rate' *mean*f=percent9.2
  baltran='Balance Transfer      per respond'
          *mean*f=dollar9.0
  baltran=' Balance Transfer      per mailed'
          *pctsum<mailed>*f=btm.
  /rts=19 row=float misstext=' ' box=_page_ condense;
run;
```

Try a PAGE Dimension

campaign 2004/3	Mailed	%	Responders	Response Rate	Balance Transfer per respond	Balance Transfer per mailed
offer A	450,137	45.01%	45,288	10.06%	\$6,498	\$654
B	549,863	54.99%	27,748	5.05%	\$5,402	\$273
TOTAL	1,000,000	100.00%	73,036	7.30%	\$6,082	\$444

campaign 2004/4	Mailed	%	Responders	Response Rate	Balance Transfer per respond	Balance Transfer per mailed
offer A	449,678	44.97%	45,082	10.03%	\$6,504	\$652
B	550,322	55.03%	27,484	4.99%	\$5,400	\$270
TOTAL	1,000,000	100.00%	72,566	7.26%	\$6,086	\$442

Be Careful with PICTURE FORMATS

picture btm (round) low - high = '0,000,009.00' (prefix='\$' mult=.01);

NESUG		Mailed	%	Responders	Response Rate	Balance Transfer per respond	Balance Transfer per mailed
campaign offer							
2004/3	A	450,137	45.01%	45,288	10.06%	\$6,498	\$6.54
	B	549,863	54.99%	27,748	5.05%	\$5,402	\$2.73
	TOTAL	1,000,000	100.00%	73,036	7.30%	\$6,082	\$4.44
2004/4 offer							
	A	449,678	44.97%	45,082	10.03%	\$6,504	\$6.52
	B	550,322	55.03%	27,484	4.99%	\$5,400	\$2.70
	TOTAL	1,000,000	100.00%	72,566	7.26%	\$6,086	\$4.42

Be Careful with PICTURE FORMATS

picture btm (round) low - high = '0,000,009.00' (prefix='\$' mult=1);

NESUG		Mailed	%	Responders	Response Rate	Balance Transfer per respond	Balance Transfer per mailed
campaign offer							
2004/3	A	450,137	45.01%	45,288	10.06%	\$6,498	\$653.77
	B	549,863	54.99%	27,748	5.05%	\$5,402	\$272.59
	TOTAL	1,000,000	100.00%	73,036	7.30%	\$6,082	\$444.17
2004/4 offer							
	A	449,678	44.97%	45,082	10.03%	\$6,504	\$652.03
	B	550,322	55.03%	27,484	4.99%	\$5,400	\$269.68
	TOTAL	1,000,000	100.00%	72,566	7.26%	\$6,086	\$441.61

Step 5 (Where can you put the f=?)

- You can add formats with the f= after statistic specification, variables, (), here, there and everywhere.

```
proc tabulate data=test noseps missing;
  class campaign offer;
  var baltran;
  table campaign*(offer all)all
    ,
    baltran='Balance Transfer Amounts'*f=dollar7.
    *
    (
    (n nmiss)*f=comma9.
    min p25 mean median p75 max
    )
    /rts=19 row=float misstext=' ';
run;
```

```
WARNING: Dimension crossing has multiple format modifiers.
WARNING: Dimension crossing has multiple format modifiers.
```

Step 5 (Where can you put the f=?)

OUTPUT:

		Balance Transfer Amounts							
		N	NMiss	Min	P25	Mean	Median	P75	Max
campaign offer									
2004/3	A	45,288	404,849	\$4,474	\$6,164	\$6,498	\$6,498	\$6,834	\$8,559
	B	27,748	522,115	\$3,388	\$5,061	\$5,402	\$5,402	\$5,741	\$7,540
	All	73,036	926,964	\$3,388	\$5,543	\$6,082	\$6,141	\$6,625	\$8,559
2004/4	offer								
	A	45,082	404,596	\$4,444	\$6,169	\$6,504	\$6,505	\$6,841	\$8,513
	B	27,484	522,838	\$3,467	\$5,066	\$5,400	\$5,399	\$5,737	\$7,448
	All	72,566	927,434	\$3,467	\$5,546	\$6,086	\$6,147	\$6,631	\$8,513
All		145,602	1,854,398	\$3,388	\$5,545	\$6,084	\$6,144	\$6,629	\$8,559

↑
“N” here are for the number of records with non-missing BALTRAN

Step 5 (Where can you put the f=?)

- This will not get you the same results:

```
proc tabulate data=test nosepts missing;
  class campaign offer;
  var baltran;
  format baltran dollar7.;
  table campaign*(offer all)all
      ,
      baltran='Balance Transfer Amounts'
      *
      (
      (n nmiss)*f=comma9.
      min p25 mean median p75 max
      )
      /rts=19 row=float misstext=' ';
run;
```

Step 5 (Where can you put the f=?)

- This will get you desired results and not print out warning messages in the LOG:

```
proc tabulate data=test noseps missing;
  class campaign offer;
  var baltran;
  table (campaign*(offer all)all)*f=dollar7.
      ,
      baltran='Balance Transfer Amounts'
      *
      (
      (n nmiss)*f=comma9.
      min p25 mean median p75 max
      )
      /rts=19 row=float misstext=' ';
run;
```

Step 6 - Run Code with OBS=MAX



Step 7 - Add some ODS Functionality

ODS, Output Delivery System, can output results to WEB, PDF, RTF and Spreadsheet files.

```
ODS Listing CLOSE;
ODS HTML file='c:\NESUG2005\TAB1.xls';
proc tabulate data=test noseps;
  class offer campaign;
  var respond baltran mailed;
  keylabel n=' ' sum=' ' mean=' ' pctn=' ' pctsum=' ' ;
  table campaign*(offer all=' TOTAL ' )
    ,
    n='Mailed' *f=comma9.
    pctn<offer all>=' % ' *f=pct.
    respond='Responders' *f=comma10.
    respond='Response Rate' *mean*f=percent9.2
    baltran='Balance Transfer      per respond' *mean*f=dollar9.0
    baltran=' Balance Transfer      per mailed' *pctsum<mailed>*f=btm.
  /rts=19 row=float misstext=' ' box=' NESUG ' ;
run;
ODS HTML close;
ODS Listing;
```


Step 7 (continued)

The SAS System

NESUG		Mailed	%	Responders	Response Rate	Balance Transfer per respond	Balance Transfer per mailed
campaign	offer	450,137	45.01%	45,288	10.06%	\$6,498	\$654
2004/3	A						
	B	549,863	54.99%	27,748	5.05%	\$5,402	\$273
	TOTAL	1,000,000	100.00%	73,036	7.30%	\$6,082	\$444
2004/4	offer	449,678	44.97%	45,082	10.03%	\$6,504	\$652
	A						
	B	550,322	55.03%	27,484	4.99%	\$5,400	\$270
	TOTAL	1,000,000	100.00%	72,566	7.26%	\$6,086	\$442

On some versions of SAS, code will produce shading of cells. To minimize Shading:

```
ODS HTML file='c:\NESUG2005\TAB1.xls' style=minimal;
```

Want to eliminate all lines? Look at SAS support document to create a noborder style:

<http://support.sas.com/sassamples/quicktips/04feb/ods-excel.html>


Step 7 - Add some ODS Functionality

tagsets.MSOffice2K in SAS9

```
ODS Listing CLOSE;
ODS tagsets.MSOffice2K file='c:\presentations\philasug\TAB1.xls'
    style=statistical;
proc tabulate data=test noseps;
  class offer campaign;
  var respond baltran mailed;
  keylabel n=' ' sum=' ' mean=' ' pctn=' ' pctsum=' ' ;
  table campaign*(offer all=' TOTAL ' )
    ,
    n='Mailed' *f=comma9.
    pctn<offer all>=' % ' *f=pct.
    respond='Responders' *f=comma10.
    respond='Response Rate' *mean*f=percent9.2
    baltran='Balance Transfer      per respond' *mean*f=dollar9.0
    baltran=' Balance Transfer      per mailed' *pctsum<mailed>*f=btm.
  /rts=24 row=float misstext=' ' box=[Label='PhilaSUG'
    Style=[preimage='C:\Presentations\Philasug\psa_logo.gif']];
run;
ODS tagsets.MSOffice2K close;
ODS Listing;
```

Step 7 - Add some ODS Functionality

The SAS System

 PhilaSUG		Mailed	%	Responders	Response Rate	Balance Transfer per respond	Balance Transfer per mailed
campaign	offer						
2004/3	A	450,137	45.01%	45,288	10.06%	\$6,498	\$654
	B	549,863	54.99%	27,748	5.05%	\$5,402	\$273
	TOTAL	1,000,000	100.00%	73,036	7.30%	\$6,082	\$444
2004/4	offer						
	A	449,678	44.97%	45,082	10.03%	\$6,504	\$652
	B	550,322	55.03%	27,484	4.99%	\$5,400	\$270
	TOTAL	1,000,000	100.00%	72,566	7.26%	\$6,086	\$442

Step 7 (continued)

Your Boss wants changes:

- Put Offer in Rows and Campaign in columns
- For each Offer show statistics in Rows
- Put labels for offers

Step 7 (continued)

```
proc format;
value $offer 'A' = 'A: Really good Offer'
             'B' = 'B: Good Offer'
;

ODS Listing CLOSE;
ODS HTML file='c:\NESUG2005\TAB2.xls' style=minimal;
proc tabulate data=test noseps;
  class offer campaign;
  format offer $offer.;
  var respond baltran mailed;
  keylabel n=' ' sum=' ' mean=' ' pctn=' ' pctsum=' ' ;
  table (offer all='TOTAL' )
        *
        (n='Mailed' *f=comma9.
         pctn<offer all>=' % ' *f=pct.
         respond='Responders' *f=comma10.
         respond='Response Rate' *mean*f=percent9.2
         baltran='Balance Transfer per respond' *mean*f=dollar9.0
         baltran='Balance Transfer per mailed' *pctsum<mailed>*f=btm.
        )
        ,
        campaign
  /rts=19 row=float misstext=' ' box='NESUG' ;
run;
ODS HTML close;
ODS Listing;
```

Step 7 (continued)

SUGI		campaign	
		2004/3	2004/4
offer		450,137	449,678
A: Really good Offer	Mailed		
	%	45.01%	44.97%
	Responders	45,288	45,082
	Response Rate	10.06%	10.03%
	Balance Transfer per respond	\$6,498	\$6,504
	Balance Transfer per mailed	\$654	\$652
B: Good Offer	Mailed	549,863	550,322
	%	54.99%	55.03%
	Responders	27,748	27,484
	Response Rate	5.05%	4.99%
	Balance Transfer per respond	\$5,402	\$5,400
	Balance Transfer per mailed	\$273	\$270
TOTAL	Mailed	1,000,000	1,000,000
	%	100.00%	100.00%
	Responders	73,036	72,566
	Response Rate	7.30%	7.26%
	Balance Transfer per respond	\$6,082	\$6,086
	Balance Transfer per mailed	\$444	\$442

Step 7 (continued)

Results without ROW=FLOAT option

The SAS System

NESUG		campaign	
		2004/3	2004/4
offer		450,137	449,678
A: Really good Offer	Mailed		
	%	45.01%	44.97%
	Responders	45,288	45,082
	Response Rate	10.06%	10.03%
	Balance Transfer per respond	\$6,498	\$6,504
	Balance Transfer per mailed	\$654	\$652
B: Good Offer	Mailed	549,863	550,322
	%	54.99%	55.03%
	Responders	27,748	27,484
	Response Rate	5.05%	4.99%
	Balance Transfer per respond	\$5,402	\$5,400
	Balance Transfer per mailed	\$273	\$270
TOTAL	Mailed	1,000,000	1,000,000
	%	100.00%	100.00%
	Responders	73,036	72,566
	Response Rate	7.30%	7.26%
	Balance Transfer per respond	\$6,082	\$6,086
	Balance Transfer per mailed	\$444	\$442

Step 7 (continued)

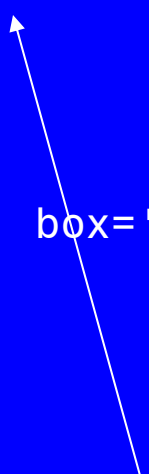
Your Boss wants MORE changes:

- Total dollars transferred for each offer
- Total percent of dollars per offer in each campaign

Step 7 (continued)

```
ODS Listing CLOSE;
ODS HTML file='TAB2.xls' style=minimal;
proc tabulate data=test noseps;
  class offer campaign;
  format offer $offer.;
  var respond baltran mailed;
  keylabel n=' ' sum=' ' mean=' ' pctn=' ' pctsum=' ' ;
  table ...
    baltran='Total Transfer'*f=dollar13.
    baltran='% Total Transfer'*colpctsum=' '*f=pct.
  )
  ,
    campaign
  /rts=19 row=float misstext=' ' box='NESUG' ;
run;
ODS HTML close;
ODS Listing;
```

```
baltran='% Total Transfer'*pctsum<offer all>*f=pct.
```



Step 7 (continued)

The SAS System

NESUG		campaign	
		2004/3	2004/4
offer		450,137	449,678
A: Really good Offer	Mailed		
	%	45.01%	44.97%
	Responders	45,288	45,082
	Response Rate	10.06%	10.03%
	Balance Transfer per respond	\$6,498	\$6,504
	Balance Transfer per mailed	\$654	\$652
	Total Transfer	\$294,285,379	\$293,204,204
	% Total Transfer	66.26%	66.39%
B: Good Offer	Mailed	549,863	550,322
	%	54.99%	55.03%
	Responders	27,748	27,484
	Response Rate	5.05%	4.99%
	Balance Transfer per respond	\$5,402	\$5,400
	Balance Transfer per mailed	\$273	\$270
	Total Transfer	\$149,885,279	\$148,408,109
	% Total Transfer	33.74%	33.61%
TOTAL	Mailed	1,000,000	1,000,000
	%	100.00%	100.00%
	Responders	73,036	72,566
	Response Rate	7.30%	7.26%
	Balance Transfer per respond	\$6,082	\$6,086
	Balance Transfer per mailed	\$444	\$442
	Total Transfer	\$444,170,659	\$441,612,313
	% Total Transfer	100.00%	100.00%

Step 8 - Need to Generate Multi-Label Formats?

- SAS8 and higher allow for multi-label user formats. It allows for a 1-to-many or many-to-many table look-up generation.
- Example: Credit Card Approval Report: There are 3 approval codes and 2 decline codes. Desire report with totals for each code and totals for all approvals and all declines.

Step 8 (continued)

```
proc format;
  picture p8r (round)low - < 0 = '0009.99%' (prefix='-')
                    0 - high = '0009.99%'
;
value $decode (multilabel notsorted)
  'a0' - 'a9' = 'APPROVE TOTALS'
  'a1' = ' a1: Approval'
  'a2' = ' a2: Weak Approval'
  'a4' = ' a4: Approved Alternate Product'
  'd0' - 'd9' = 'DECLINE TOTALS'
  'd1' = ' d1: Decline for Credit'
  'd6' = ' d6: Decline Other'
;
proc tabulate data=decision noseps formchar=' ' ;
  class decision/mlf preloadfmt order=data;
  format decision $decode.;
  table (decision all)
    ,n*f=comma5.
    pctn='%'*f=p8r.
    /rts=33 row=float misstext=' ';
run;
```

Step 8 (continued)

	N	%
decision		
APPROVE TOTALS	314	31.40%
a1: Approval	163	16.30%
a2: Weak Approval	45	4.50%
a4: Approved Alternate Product	106	10.60%
DECLINE TOTALS	686	68.60%
d1: Decline for Credit	453	45.30%
d6: Decline Other	233	23.30%
All	1,000	100.00%

One More ODS Trick:
Using ODS OUTPUT to get TABULATE to calculate
Column LOG(ODDS) and Column Differences

```
options nocenter mprint symbolgen errorabend errors=2;
libname fff 'C:\SAS Credit Industry\DATA';
proc rank data=fff.data_gb
          out=ranks
          ties=low
          groups=10;
  var ntrades;
  ranks r_ntrades;
run;

data ranks_sums;
  set ranks;
  good=(bad=0);
run;

proc format;
  picture pct (round) low - < 0 = '0009.99%' (prefix='-')
                    0 - high = '009.99%'
;
run;
```

One More ODS Trick:
Using ODS OUTPUT to get TABULATE to calculate
Column LOG(ODDS) and Column Differences

```
ods output table=woe;
proc tabulate data=ranks_sums;
  class r_ntrades;
  var ntrades good bad;
  table (r_ntrades='Rank ntrades' )
        ,
        n*f=comma5.
        pctn='% '*f=pct.
        ntrades*(min mean max)
        bad*(sum='N '*f=comma5. pctsum<r_ntrades>='% '*f=pct.)
        good*(sum='N '*f=comma5. pctsum<r_ntrades>='% '*f=pct.)
        /rts=10 row=float misstext=' ';
run;
ods output close;
```

One More ODS Trick:
Using ODS OUTPUT to get TABULATE to calculate
Column LOG(ODDS) and Column Differences

```
data _null_;
  set woe end=eof;
  good_tot+good_sum;
  bad_tot+bad_sum;
  if eof then do;
    ln_odds=log(good_tot/bad_tot);
    call symputx('ln_odds',ln_odds);
  end;
run;
```

```
data woe_calc;
  set woe end=eof;
  log_odds = log(good_sum/bad_sum);
  woe = log_odds-&ln_odds;
  ivalue = woe*(good_PctSum_0-bad_PctSum_0)/100;
  ivalue_tot+ivalue;
  nrecs=n;
  if eof then call symputx('ivalue',put(ivalue_tot,6.2));
run;
```


One More ODS Trick: Using ODS OUTPUT to get TABULATE to calculate Column LOG(ODDS) and Column Differences

```
ods tagsets.MSOffice2k file='woe2.xls' style=statistical;
proc tabulate data=woe_calc;
  class r_ntrades;
  var nrecs PctN_0 ntrades_min ntrades_mean ntrades_max good_sum bad_sum
      good_pctsum_0 bad_pctsum_0 log_odds woe ivalue;
  keylabel sum=' ';
  table (r_ntrades='Rank ntrades' )
      ,
      nrecs*f=comma5.
      PctN_0='% '*f=pct.
      ntrades_min ntrades_mean ntrades_max
      bad_sum='bad'
      bad_pctsum_0='bad% '*f=pct.
      good_sum='good'
      good_pctsum_0='good% '*f=pct.
      log_odds*f=6.2
      woe*f=6.2
      ivalue='Info_Value '*f=6.2
      /rts=10 row=float misstext=' ' box="INFO_V=&ivalue";
run;
ods tagsets.MSOffice2k close;
```

One More ODS Trick:
Using ODS OUTPUT to get TABULATE to calculate Column
LOG(ODDS) and Column Differences

INFO_V=0.18	nrecs	%	ntrades_ Min	ntrades_ Mean	ntrades_ Max	bad	bad%	good	good%	log_o dds	woe	Info_V alue
Rank ntrades												
0	134	13.40%	0	0.96	2	19	9.05%	115	14.56%	1.8	0.48	0.03
1	95	9.50%	3	3.46	4	22	10.48%	73	9.24%	1.2	-0.13	0
2	88	8.80%	5	5.52	6	14	6.67%	74	9.37%	1.67	0.34	0.01
3	83	8.30%	7	7.57	8	14	6.67%	69	8.73%	1.6	0.27	0.01
4	101	10.10%	9	10.02	11	22	10.48%	79	10.00%	1.28	-0.05	0
5	113	11.30%	12	12.95	14	17	8.10%	96	12.15%	1.73	0.41	0.02
6	93	9.30%	15	16.05	17	18	8.57%	75	9.49%	1.43	0.1	0
7	103	10.30%	18	19.46	21	18	8.57%	85	10.76%	1.55	0.23	0
8	105	10.50%	22	24.28	27	32	15.24%	73	9.24%	0.82	-0.5	0.03
9	85	8.50%	28	35.92	58	34	16.19%	51	6.46%	0.41	-0.92	0.09

The Fish Problem

Problem:

- A tabulate report generates measurements of fish lengths, in formatted ranges. However, if a range is not present in the sample, we also wish to include missing ranges in the report.

Solution:

- Use **CLASSDATA** option in **TABULATE**

The Fish Problem

Simulated DATA:

```
80
81  data test;
82    do i = 1 to 1000;
83      length = int(100*ranuni(23));
84      if length < 40 or length > 50 then output;
85    end;
86  run;
```

NOTE: The data set WORK.TEST has 897 observations and 2 variables.

NOTE: DATA statement used (Total process time):

real time	0.02 seconds
cpu time	0.02 seconds

The Fish Problem

FORMAT:

```
proc format;
  value length low - < 20 = '0 - LT 20'
              20 - < 40 = '20 - LT 40'
              40 - < 50 = '40 - LT 50'
              50 - < 75 = '50 - LT 75'
              75 - high = '75+'
;
  picture pct (round) low - high = '0009.99%'
;
run;
```

Include Missing Groups

TABULATE CODE:

```
proc tabulate data=test noseps formchar=' ' missing;
  class length;
  format length length.;
  table length all
      ,
      n*f=comma6.
  pctn<length all>='% '*f=pct.
  /rts=20 row=float misstext=' ';
run;
```

OUTPUT

	N	%
length		
0 - LT 20	201	22.41%
20 - LT 40	208	23.19%
50 - LT 75	211	23.52%
75+	277	30.88%
All	897	100.00%

The "40 - LT 50" range is missing

Include Missing Groups

CREATE A CLASSDATA:

```
data classes;
  do length = 0,20,40,50,75;
    output;
  end;
run;
proc tabulate data=test noseps formchar=' ' missing
  classdata=classes;
  . . .
```

OUTPUT

	N	%
length		
0 - LT 20	201	22.41%
20 - LT 40	208	23.19%
40 - LT 50		0.00%
50 - LT 75	211	23.52%
75+	277	30.88%
All	897	100.00%

Include Missing Groups

Replace the missing N for the 40-LT 50 range:

Use MISSTEXT='0' or, if on a Windows PC, use a PICTURE FORMAT:

```
proc format;
  value zmiss . = '  0'
             other = [comma6.]
;
run;
proc tabulate . . .
  n*f=zmiss.
  . . .
```

The spaces are alt-255

OUTPUT

	N	%
length		
0 - LT 20	201	22.41%
20 - LT 40	208	23.19%
40 - LT 50	0	0.00%
50 - LT 75	211	23.52%
75+	277	30.88%
All	897	100.00%

**Step 9 - Sit Back, Smile and Be Proud that you are a
TABULATE Master!**

Why use TABULATE over PIVOT Tables in Excel?

- TABULATE provides code that can replicate results for current and future table reports. Pivot Tables are hard to reproduce since no code is generated. TABULATE: Documented reliable reports.

SAS® is a registered trademark of SAS Institute.

The contents of this paper are the work of the author and do not necessarily represent the opinions, recommendations, or practices of JP MORGAN CHASE.

Contact Information:

Email:

jonas.bilenas@chase.com

jonas@jonasbilenas.com



The Power of PROC FORMAT

By Jonas V. Bilenas

Order #: 59498