

## DON'T COPY AND PASTE—USE BY STATEMENT PROCESSING WITH ODS TO MAKE YOUR SUMMARY TABLES

Jeffrey M. Gossett and Mallikarjuna Rettiganti; University of Arkansas for Medical Sciences

### ABSTRACT

A majority of the manuscripts in medical journals contain summary tables that combine simple summaries and between-group comparisons. These tables typically combine summary statistics for categorical and continuous variables into one table. The FREQ procedure is generally used to summarize categorical variables and compare percentages between groups using a Chi-square or a Fisher's exact test. For continuous variables, the MEANS procedure is used to summarize data as either means and standard deviation or medians and quartiles. These are then compared between groups by using the GLM (or TTEST) or NPAR1WAY procedure, depending on whether one is interested in a parametric test or a non-parametric test. The outputs from these different procedures are then combined and presented in a concise format ready for publication. Currently there is no straightforward way in SAS® to build these tables in a presentable format that can then be customized to individual tastes. In this paper, we focus on presenting summary statistics and results from comparing categorical variables between two or more independent groups. The macro takes the dataset, the number of treatment groups, and the type of test (either chi-square or Fisher's exact) as input and presents the results in a publication-ready table. This macro also automates summarizing data to a certain extent and minimizes risky typographical errors when copying results or typing them into a table.

### INTRODUCTION

Table 1 is an example of a summary table that can be obtained by using the macro we propose. The data is from a stress echocardiography study (Garfinkel 1999). In that study, the authors compared 469 patients with no echocardiograph events ("Controls") versus 89 who had at least one event ("Treatments"). The list of patient characteristics is given in the first column. The second and third columns have summary measures for the control and treatment subgroups, respectively. The fourth column gives the p-value for a statistical test comparing the control group to the treatment group. We summarized the continuous variables as median (Q1, Q3). In the original paper, the continuous variables were summarized as mean and standard deviation. We used Pearson's chi-square tests to compare distributions of categorical variables by the group variable. Wilcoxon tests were used to compare the distributions of continuous variables by group. Categorical variables were summarized with counts and column percents. Some of the categorical variables are binary and were summarized in one row (e.g. "history of angioplasty". The "Yes" and "No" responses sum to 100%, the "No" line was omitted. Similarly, for the variable Gender, either the "Gender: male" line or the "Gender: female" line could have been omitted in this table. Note that the percentages, medians and interquartile ranges have also been rounded to whole numbers. In this paper, we discuss how this table was generated. We make repeated use of the TRANSPOSE procedure to convert between wide and long data formats and take advantage of SAS's powerful by group processing.

**Table 1: Example summary table**

	Group		P-value
	Control	Treatment	
N	469 (100%)	89 (100%)	
Gender : female	292 (62%)	46 (52%)	0.061
Gender : male	177 (38%)	43 (48%)	
Experienced chest pain	139 (30%)	33 (37%)	0.16
History of angioplasty	32 (7%)	9 (10%)	0.28
History of smoking : heavy	98 (21%)	24 (27%)	0.35
History of smoking : moderate	115 (25%)	23 (26%)	
History of smoking : non-smoker	256 (55%)	42 (47%)	
Basal blood pressure	132 (120 to 150)	136 (120 to 150)	0.63
Basal heart rate	73 (64 to 85)	74 (67 to 83)	0.75
Pct max predicted HR	78 (69 to 89)	76 (69 to 88)	0.28
Age (years)	68 (59 to 76)	70 (64 to 75)	0.17
Basal double product	9720 (8400 to 11696)	10050 (8466 to 11480)	0.56
Baseline ejection fraction	58 (55 to 63)	53 (42 to 60)	<0.001
Systolic blood pressure	142 (122 to 172)	140 (110 to 166)	0.18

Categorical variables were summarized as n (%), whereas continuous were summarized as median and interquartile range. P-values are based on Pearson's chi-square test for categorical variables and Wilcoxon tests for continuous.

## WHAT IS THE NAME OF THAT OUTPUT DELIVERY SYSTEM (ODS) TABLE?

Tip: In order to find ODS table names enclose procedure code with:

```
ODS trace on / listing;

PROC ...;

run;

ODS trace off;
```

## PLANNING THE TABLE

In approaching this problem there are several issues to consider: 1) The number of levels of the grouping variable 2) The type of variables to be summarized 3) the type of summary statistics 4) whether or not the categorical variables are formatted; are they character valued? or are they numeric with formats? How are missing values coded? 5) the statistical tests to use for categorical and for continuous variables? 6) the number of decimal places to keep.

To simplify the coding we made some assumptions. Although the data start out in wide format (one record per subject), it is convenient to transpose the data into long format for analysis. The advantage is the ability to do by group processing by variable name and having nicely formatted output data sets. The disadvantage is that if numeric variables with formats are transposed, there is extra work involved in reformatting. For simplicity, we assume that categorical variables are coded as character variables unless the variable is binary in which case they are coded 0/1 with the '1' being the category of interest. It

is important to note that we will drop rows coded as “0”. If the data are numeric with formats, please see, for example, (Wang 2011) or (King 2010).

As statisticians working with relatively small data sets, we were willing to sacrifice a little computing efficiency for simplicity. Our objective is to use the code to generate the estimates needed in a basic format. We typically do some final modification within a Microsoft Excel® spreadsheet (e.g. rearranging rows, modifying labels, tweaking the rounding of really big or really small numbers, etc. King (2010) provides elegant code that applies individual numeric formats to each variable.

## STRESS ECHO DATA

The stress echocardiographic data had 558 observations on 32 variables. See acknowledgement for more information about data source. The *row\_id* variable is an identifier for each subject and is numbered 1 to 558. All variables in the data set are labeled. Binary categorical variables (other than gender) are coded 0=No/1=Yes. All other categorical variables are character valued.

Macro variables are used to identify the data set name (dsname), subject identifier (id), the group variable (xgroup), the continuous variables (xconts), and the categorical variables (xcats). The rounding units for continuous summaries (roundcont) and percentages (roundpct) are set with macro variables. If all of the variables have labels, it is convenient to use labels rather than variable names in tables (vlabel=vlabel). If variable names are preferred, or if some variables are missing labels, use vlabel=vnames. The code for the Stress Echo data follows:

```
*****;  
* Declaring variables to be used;  
*****;  
%let dsname = stress;  
%let id = row_id;  
%let xgroup = group; * real group name is: any_event; * a single grouping  
                    variable;  
%let xconts = bhr basebp basedp pkhr sbp dp maxhr mbp age baseef dobef  
             pctMphr;  
%let xcats = gender chestpain hxofht hxofdm hxofcig hxofmi hxofptca  
            hxofcabg;  
%let roundcont = 1;  
%let roundpct = 1;  
%let vlabel = vlabel;  
*** Note: binary variables coded 0/1. Redundant 0 level will be dropped  
        From summaries;
```

## SUMMARIZING CATEGORICAL VARIABLES

Using a data step, we create a new data set that selects the ID, categorical, and the grouping variables. In order to get overall counts per group, we create a variable N which is set equal to 1:

```
data wide_cat1;  
  set &dsname(keep= &id &xcats &xgroup);  
  N=1; label N="N";  
run;
```

We use the TRANSPOSE procedure to convert the data from wide to long format. Since the categorical variables in this dataset are a mixture of numeric and character variables, the variables to be transposed must be explicitly named. The transposed data will have a column named *&vlabel* identifying the original categorical variable, a variable named value that y1 containing the values (note prefix=y option), a group variable *&xgroup*, and the ID variable *&id*.

```

proc transpose data=wide_cat1 out=long_cat1 label=vlabel prefix=y ;
  by &id &xgroup;
  var &xcats;
run;
data long_cat1;
  set long_cat1(rename=( _name_ = vname));
run;

```

Here's where we sacrifice efficiency for simplicity. We run the FREQ procedure by variable (*&ylabel*) and by the group variable (*&xgroup*), to obtain the counts and column percentages. The format of the OneWayFreqs output delivery system (ODS) table is much simpler than the CrossTabFreqs table (Long et al 2008):

```

proc sort data=long_cat1;
  by &vlabel &xgroup;
run;
ods output OneWayFreqs=long_cat2;
proc freq data=long_cat1 ;
  by &vlabel &xgroup;
  table y1 / noCum;
run;

```

We use a data step to combine the *n* and percent, *n (%)*, into one summary variable named *n\_pct*. The SAS catenation functions CATS and CATX are used to control the spacing. The CATS function removes all leading and trailing blanks. The CATX function also removes the blanks, but allows for a separation character between the members being concatenated.

```

* Combine "N (percent)" into one variable;
data long_cat3;
  set long_cat2;
  n_pct = catx(' ',Frequency, cats('(' , round(Percent, &roundpct), '%)'));
run;

```

The select variables from the first 8 observations of data set LONG\_CAT3 are shown in Table 1. Note that *Any\_event=1* is our treatment group, and *Any\_event=0* is the control group. The *y1* variable is the categorical level of the variable.

**Table 1: First 8 observations of long\_cat3 (pre-transpose)**

vlabel	Group	y1	Frequency	Percent	n_pct
N	Control	1	469	100	469 (100%)
N	Treatment	1	89	100	89 (100%)
experienced chest pain	Control	0	330	70.36	330 (70%)
experienced chest pain	Treatment	0	56	62.92	56 (63%)
experienced chest pain	Control	1	139	29.64	139 (30%)
experienced chest pain	Treatment	1	33	37.08	33 (37%)
gender	Control	female	292	62.26	292 (62%)
gender	Treatment	female	46	51.69	46 (52%)

We use PROC TRANSPOSE to convert the summaries from long to wide format:

```
proc sort data=long_cat3;
  by &vlabel y1 &xgroup;
run;
proc transpose data=long_cat3 out=long_cat4;
  by &vlabel y1 ;
  id &xgroup;
  var n_pct;
run;
```

A partial print output from the data set LONG\_CAT4 is found in Table 2.

**Table 2: Selection from data set LONG\_CAT4**

vlabel	y1	_NAME_	Control	Treatment
N	1	n_pct	469 (100%)	89 (100%)
experienced chest pain	0	n_pct	330 (70%)	56 (63%)
experienced chest pain	1	n_pct	139 (30%)	33 (37%)
gender	female	n_pct	292 (62%)	46 (52%)

Recall that categorical variables were assumed to be character valued, but we coded binary variables as 0=No/1=Yes. After the initial PROC TRANSPOSE, the variables were converted to a common character format. We delete rows where cats(y1)="0". Again, the CATS function removes all leading and trailing blanks:

```
data long_cat5;
  set long_cat4;
  where cats(y1) NOT = "0";
  *cats function removes spaces from y1. Assumes the only 0's are the
  binary variables. Use with care;
  ** Add a row identifier. May be useful for ordering or merging;
  order = _n_;
run;
```

Tip: If PROC TRANSPOSE CREATES extra Y variables (i.e. Y2, Y3, ...) it means there is more than one record for at least one ID. We create an index within each variable "row". This may be used in merging p-values with summaries:

```
proc sort data= long_cat5;
  by &vlabel y1;
run;
data long_cat6;
  set long_cat5;
  by &vlabel y1;
  retain row 0;
  if first.&vlabel then row=0;
  row + 1;
run;
proc print data=long_cat6 noobs;
run;
```

Table 3 shows our current table in data set long\_cat6. Note that the summary variables are named Control and Treatment in this data set based on the coding of the grouping variable GROUP.

**Table 3: PROC PRINT output for long\_cat6.**

vlabel	y1	Control	Treatment	order	Row
N	1	469 (100%)	89 (100%)	1	1
experienced chest pain	1	139 (30%)	33 (37%)	2	1
Gender	female	292 (62%)	46 (52%)	3	1
Gender	male	177 (38%)	43 (48%)	4	2
history of angioplasty	1	32 (7%)	9 (10%)	5	1
history of bypass surgery	1	68 (14%)	20 (22%)	6	1
history of diabetes	1	162 (35%)	44 (49%)	7	1
history of heart attack	1	113 (24%)	41 (46%)	8	1
history of hypertension	1	320 (68%)	73 (82%)	9	1
history of smoking	heavy	98 (21%)	24 (27%)	10	1
history of smoking	moderate	115 (25%)	23 (26%)	11	2
history of smoking	non-smoker	256 (55%)	42 (47%)	12	3

## GENERATE PEARSON CHI-SQUARE P-VALUES FOR CATEGORICAL VARIABLES

We test each categorical for association with the &xgroup variable using Pearson's chi-square test.

```
ods output ChiSq=cat_p1;
proc freq data=long_cat1 ;
  by &vlabel ;
  table y1*&xgroup / chisq;
run;
```

Let's print the subset of the SAS data table named cat\_p1:

```
proc print data=cat_p1;
  where vlabel="gender";
run;
```

Output is contained in Table 4. Note that there are 7 rows, so we need to filter out the row corresponding to the test of interest.

**Table 4: ChiSq ODS table for the variable "gender"**

Obs	vlabel	Table	Statistic	DF	Value	Prob
8	gender	Table y1 * any_event	Chi-Square	1	3.5026	0.0613
9	gender	Table y1 * any_event	Likelihood Ratio Chi-Square	1	3.4462	0.0634
10	gender	Table y1 * any_event	Continuity Adj. Chi-Square	1	3.0739	0.0796
11	gender	Table y1 * any_event	Mantel-Haenszel Chi-Square	1	3.4964	0.0615
12	gender	Table y1 * any_event	Phi Coefficient	—	0.0792	—
13	gender	Table y1 * any_event	Contingency Coefficient	—	0.079	—
14	gender	Table y1 * any_event	Cramer's V	—	0.0792	—

We select “Chi-square” test with a where clause in a data step:

```
data cat_p2;
  set cat_p1(keep = &vlabel Statistic Prob);
  where statistic="Chi-Square";
  row = 1;
  drop statistic;
run;
```

Now that there is one record per variable, we can merge the p-values with the summaries by the variable &vlabel:

```
proc sort data=cat_p2;
  by &vlabel row;
run;
proc sort data=long_cat6;
  by &vlabel row;
run;
data cat_table1;
  merge long_cat6 cat_p2;
  by &vlabel /*row*/;
run;
```

Now print the table:

```
proc print data=cat_table1 noobs;
run;
```

The results are in Table 5. We have all of the categorical variables summarized by the &xgroup variable. The column labeled \_0 corresponds to “no events”, and \_1 corresponds to “had events”; normally one combines the columns labeled “vlabel” and “y1” into one column. Note that within a column, the percentages for gender:female and gender:male sum to 100%, so one of the rows is redundant and could be dropped. Due to rounding, the percentages for “history of smoking” add up to 101% in the \_0 column.

Table 5: PROC PRINT of Cat\_table1

Vlabel	y1	_NAME_	Control	Treatment	order	row	Prob
N	1	n_pct	469 (100%)	89 (100%)	1	1	
experienced chest pain	1	n_pct	139 (30%)	33 (37%)	2	1	0.1634
Gender	female	n_pct	292 (62%)	46 (52%)	3	1	0.0613
Gender	male	n_pct	177 (38%)	43 (48%)	4	2	0.0613
history of angioplasty	1	n_pct	32 (7%)	9 (10%)	5	1	0.2756
history of bypass surgery	1	n_pct	68 (14%)	20 (22%)	6	1	0.0585
history of diabetes	1	n_pct	162 (35%)	44 (49%)	7	1	0.0076
history of heart attack	1	n_pct	113 (24%)	41 (46%)	8	1	<.0001
history of hypertension	1	n_pct	320 (68%)	73 (82%)	9	1	0.009
history of smoking	heavy	n_pct	98 (21%)	24 (27%)	10	1	0.3536
history of smoking	moderate	n_pct	115 (25%)	23 (26%)	11	2	0.3536
history of smoking	non-smoker	n_pct	256 (55%)	42 (47%)	12	3	0.3536

## ANOTHER OPTION: FISHER'S EXACT P-VALUES FOR CATEGORICAL VARIABLES

It is fairly straightforward to modify the code to compare the P values from Fisher's exact test with the Pearson's chi-square test. We add the EXACT option on the PROC FREQ table statement. The ODS table name is FishersExact:

```
ods output FishersExact=P_fish1 ChiSq=cat_p1;
proc freq data=long_cat1 ;
  by &vlabel ;
  table y1*&xgroup / chisq exact; run;
```

We want a 2 sided test, so we filter out the rows where name1="XP2\_FISH" in a data step:

```
data P_fish2;
  set p_fish1(keep = &vlabel label1 Name1 nValue1);
  where name1="XP2_FISH";
  row = 1;
  rename nValue1=P_Fisher;
  drop label1 Name1;
  run;
```

We repeat the code to filter out the Pearson's chi-square test:

```
data cat_p2;
  set cat_p1(keep = &vlabel Statistic Prob);
  where statistic="Chi-Square";
  row = 1;
  drop statistic;
  run;
```

We can then merge the 2 data tables with a data step:

```
data cat_p3;
  merge p_fish2 cat_p2(rename=(prob=p_PearsonChi2));
  by &vlabel;
  format p_fisher p_PearsonChi2 pvalue8.3;
  run;
proc sort data=cat_p3;
  by &vlabel row;
  run;
proc sort data=long_cat6;
  by &vlabel row;
  run;
data cat_table1r;
  merge long_cat6 cat_p3;
  by &vlabel /*row*/;
  run;
proc print data=cat_table1r noobs;
  run;
```

The results follow



Table 6.

Table 6: Cat\_table1r - with Fisher's and Pearson's p-values

vlabel	y1	Control	Treatment	p_PearsonChi2	P_Fisher
N	1	469 (100%)	89 (100%)		
experienced chest pain	1	139 (30%)	33 (37%)	0.163	0.17
Gender	female	292 (62%)	46 (52%)	0.061	0.076
Gender	male	177 (38%)	43 (48%)	0.061	0.076
history of angioplasty	1	32 (7%)	9 (10%)	0.276	0.271
history of bypass surgery	1	68 (14%)	20 (22%)	0.058	0.079
history of diabetes	1	162 (35%)	44 (49%)	0.008	0.009
history of heart attack	1	113 (24%)	41 (46%)	<.001	<.001
history of hypertension	1	320 (68%)	73 (82%)	0.009	0.008
history of smoking	heavy	98 (21%)	24 (27%)	0.354	0.342
history of smoking	moderate	115 (25%)	23 (26%)	0.354	0.342
history of smoking	non-smoker	256 (55%)	42 (47%)	0.354	0.342

## SUMMARIES OF CONTINUOUS VARIABLES

Using a dataset, we create a wide data set selecting the columns for ID's (&id), continuous variables to summarize (&xconts), and the grouping variable (&xgroup):

```
data wide_cont1;
  set stress2(keep= &id &xconts &xgroup);
run;
```

We use PROC TRANSPOSE to convert the data from wide to long format:

```
proc transpose data=wide_cont1 out=long_cont1 label=vlabel prefix=y;
  by &id &xgroup;
  var &xconts;
run;
```

A data step is used to rename \_name\_ to vname. The first 8 observations of the resulting table are shown in Table 7:

```
data long_cont1;
  set long_cont1;
  rename _name_ = vname;
run;
proc print data=long_cont1(obs=8) noobs ;
run;
```

Note that the first 8 rows (Table 7) correspond to the first subject.

Table 7: First 8 observations of transposed continuous variables in data set long\_cont1

row_id	Group	vname	vlabel	y1
1	Control	Bhr	Basal heart rate	92
1	Control	basebp	Basal blood pressure	103
1	Control	basedp	basal double product	9476

row_id	Group	vname	vlabel	y1
1	Control	Pkhr	peak heart rate	114
1	Control	Sbp	systolic blood pressure	86
1	Control	Dp	double product (= pkhr x sbp)	9804
1	Control	Maxhr	max heart rate	100
1	Control	Mbp	max blood pressure	121

The continuous variables were summarized by group variable (&xgroup) using the MEANS procedure. Note that by running the summaries by &VLABEL and &XGROUP, the output data set is in a nice format with one row per variable and grouping variable:

```
proc means data=long_cont1 n mean std min max median q1 q3 maxdec=1 nway;
  var y1;
  class &vLABEL &xgroup; * could use variable names rather than labels *;
  output out=sum_cont1 n=n mean=mean std=std min=min max=max median=median
  q1=q1 q3=q3;
run;
```

We use a data step with the concatenation functions CATX and CATS to create composite variables with mean and standard deviation combined and median and interquartile range. The numbers are rounded based on the rounding parameter &roundcont:

```
data sum_cont2;
  set sum_cont1;
  mean_sd = catx(' ', round(mean, &roundcont),
                 cats('(', round(std, &roundcont), ')'));
  if missing(mean)=0 then mean_sd = catx(' ',
    round(mean, &roundcont), cats("(", round(std, &roundcont), ")"));
  if missing(median)=0 then median_q1_q3 = catx(' ', round(median,
    &roundcont), cats("(", round(q1, &roundcont), " to ",
    cats(round(q3, &roundcont), ")"));
  row=_n_;
run;
```

A data step is used to select variables of interest, and to sort the resulting data prior to transposing. We print the first 6 rows of the data set sum\_cont3 and display it in Table 8

```
data sum_cont3;
  set sum_cont2(keep=row &xgroup &vlabel n mean_sd median_q1_q3 min_max);
run;
proc sort data=sum_cont3;
  by &vlabel &xgroup;
run;
proc print data=sum_cont3(obs=6) noobs;
  var &xgroup &vlabel mean_sd median_q1_q3;
run;
```

Table 8: first 6 rows of select variables for data set sum\_cont3

Group	Vlabel	mean_sd	median_q1_q3
Control	Basal blood pressure	135 (20.2)	132 (120 to 150)
Treatment	Basal blood pressure	136.9 (23.6)	136 (120 to 150)

Group	Vlabel	mean_sd	median_q1_q3
Control	Basal heart rate	75.2 (15.7)	73 (64 to 85)
Treatment	Basal heart rate	75.7 (14.1)	74 (67 to 83)
Control	Pct max predicted HR	78.9 (15.3)	78 (69 to 89)
Treatment	Pct max predicted HR	76.9 (14)	76 (69 to 88)

At this point we are ready to transpose the data to put the estimates for each level of the grouping variable &xgroup in a separate variable:

```
proc transpose data=sum_cont3 out=sum_cont4 ;
  by &vlabel;
  var median_q1_q3;
  id &xgroup;
run;
proc print data=sum_cont4;
run;
```

**Note:** that we only transpose the median with quartiles. The first 6 rows of data set SUM\_CONT4 are printed in

**Table 9.**

**Table 9: First 6 rows of dataset sum\_cont4**

vlabel	_NAME_	Control	Treatment
Basal blood pressure	median_q1_q3	132 (120 to 150)	136 (120 to 150)
Basal heart rate	median_q1_q3	73 (64 to 85)	74 (67 to 83)
Pct max predicted HR	median_q1_q3	78 (69 to 89)	76 (69 to 88)
age (years)	median_q1_q3	68 (59 to 76)	70 (64 to 75)
basal double product	median_q1_q3	9720 (8400 to 11696)	10050 (8466 to 11480)
baseline ejection fraction	median_q1_q3	58 (55 to 63)	53 (42 to 60)

## GENERATE WILCOXON P-VALUES FOR CONTINUOUS VARIABLES WITH NPAR1WAY PROCEDURE

We sort our long data by &vlabel and run PROC NPAR1WAY by &vlabel. The output data set for the Wilcoxon p-values is WILCOXONTEST. The test with Name1="P2\_WIL", is our two sided test of interest:

```
proc sort data=long_cont1;
  by &vlabel;
run;
ods output WilcoxonTest=wtestp;
proc npar1way data=long_cont1 WILCOXON;
  by &vlabel;
  class &xgroup;
  var y1;
run;
data wtestp2;
  set wtestp;
  where Name1 = "P2_WIL"; *Normal approximation, 2 sided test. Use P2T_WIL
    for t-distribution approx 2 sided test.;
  keep &vlabel cvalue1 nvalue1;
```

```
run;
```

## ALTERNATIVE T-TESTS FOR CONTINUOUS VARIABLES WITH TTEST PROCEDURE

Although we are using Wilcoxon tests for our summary table, we also show how we could use t-tests with the TTEST procedure. There are 2 t-tests to consider depending on whether you want to assume equal variances or unequal variances. The ODS output table name is TTESTS. Since there are 2 p-values per variable, we use PROC TRANSPOSE to convert the two p-value rows to two columns:

```
ods output ttests=ttests1;
proc ttest data=long_cont1;
  by &vlabel;
  var y1;
  class &xgroup;
run;
proc sort data=ttests1;
  by &vlabel variances;
run;
proc transpose data=ttests1 out=ttests2 prefix=p_;
  by &vlabel;
  id variances;
  var probt;
run;
```

## COMBINE SUMMARIES AND P-VALUES FOR CONTINUOUS VARIABLES

For our table, we summarized our continuous variables as medians with interquartile ranges. We used non-parametric Wilcoxon tests for p-values. We use a data step to merge summaries and p-values:

```
data cont_table1;
  merge sum_cont4 wtestp2(keep= &vlabel nvalue1 cvalue1
    rename=(nvalue1=pw cvalue1=pw_c));;
  by &vlabel;
run;
```

At this point we have 2 separate tables for categorical and continuous variables. We could copy and paste into a Microsoft Excel® spreadsheet to finalize the formatting our tables. To do so in SAS, we have to standardize variable names. Currently, we have an extra column in the categorical table. We use a SAS data step to combine the information in the *&vlabel* and *y1* columns.

```
data cat_table2;
  set cat_table1;
  format rowlabel $40.;
  * Combine information from &vlabel and y1 into one label;
  * If not binary (i.e. y1 Not = "1", then combine the 2;
  if cats(y1) = "1" then rowlabel= &vlabel;
  else rowlabel = catx(" ", &vlabel, ":", y1);
run;
proc print data=cat_table2;
  var rowlabel Control Treatment Prob;
run;
```

The data set CAT\_TABLE2 is printed in

Table 10. The PROC PRINT code assumes that the group variables will have levels CONTROL and TREATMENT.

**Table 10: Final format of categorical variables**

<b>rowlabel</b>	<b>Control</b>	<b>Treatment</b>	<b>Prob</b>
N	469 (100%)	89 (100%)	.
experienced chest pain	139 (30%)	33 (37%)	0.1634
gender : female	292 (62%)	46 (52%)	0.0613
gender : male	177 (38%)	43 (48%)	0.0613
history of angioplasty	32 (7%)	9 (10%)	0.2756
history of bypass surgery	68 (14%)	20 (22%)	0.0585
history of diabetes	162 (35%)	44 (49%)	0.0076
history of heart attack	113 (24%)	41 (46%)	<.0001
history of hypertension	320 (68%)	73 (82%)	0.009
history of smoking : heavy	98 (21%)	24 (27%)	0.3536
history of smoking : moderate	115 (25%)	23 (26%)	0.3536
history of smoking : non-smoker	256 (55%)	42 (47%)	0.3536

We standardize the variable names, merge the categorical and continuous tables and keep the variables in the final table using data steps and merges:

```
data cat_table2;
  set cat_table1;
  format rowlabel $40.;
  * Combine information from &vlabel and y1 into one label;
  * If not binary (i.e. y1 Not = "1", then combine the 2;
  if cats(y1) = "1" then rowlabel= &vlabel;
  else rowlabel = catx(" ", &vlabel, ":", y1);
run;
data cont_table2;
  set cont_table1(keep= &vlabel _0 _1 pw
  rename=(pw=prob &vlabel=rowlabel));
run;
data final;
  set cat_table2(keep= rowlabel _0 _1 Prob)
  cont_table2(keep=rowlabel _0 _1 Prob);
  label _0 = "No Events";
  label _1 = "Events";
run;
proc print data=final;
  var rowlabel _0 _1 Prob;
run;
```

The data set FINAL is essentially Table 3.

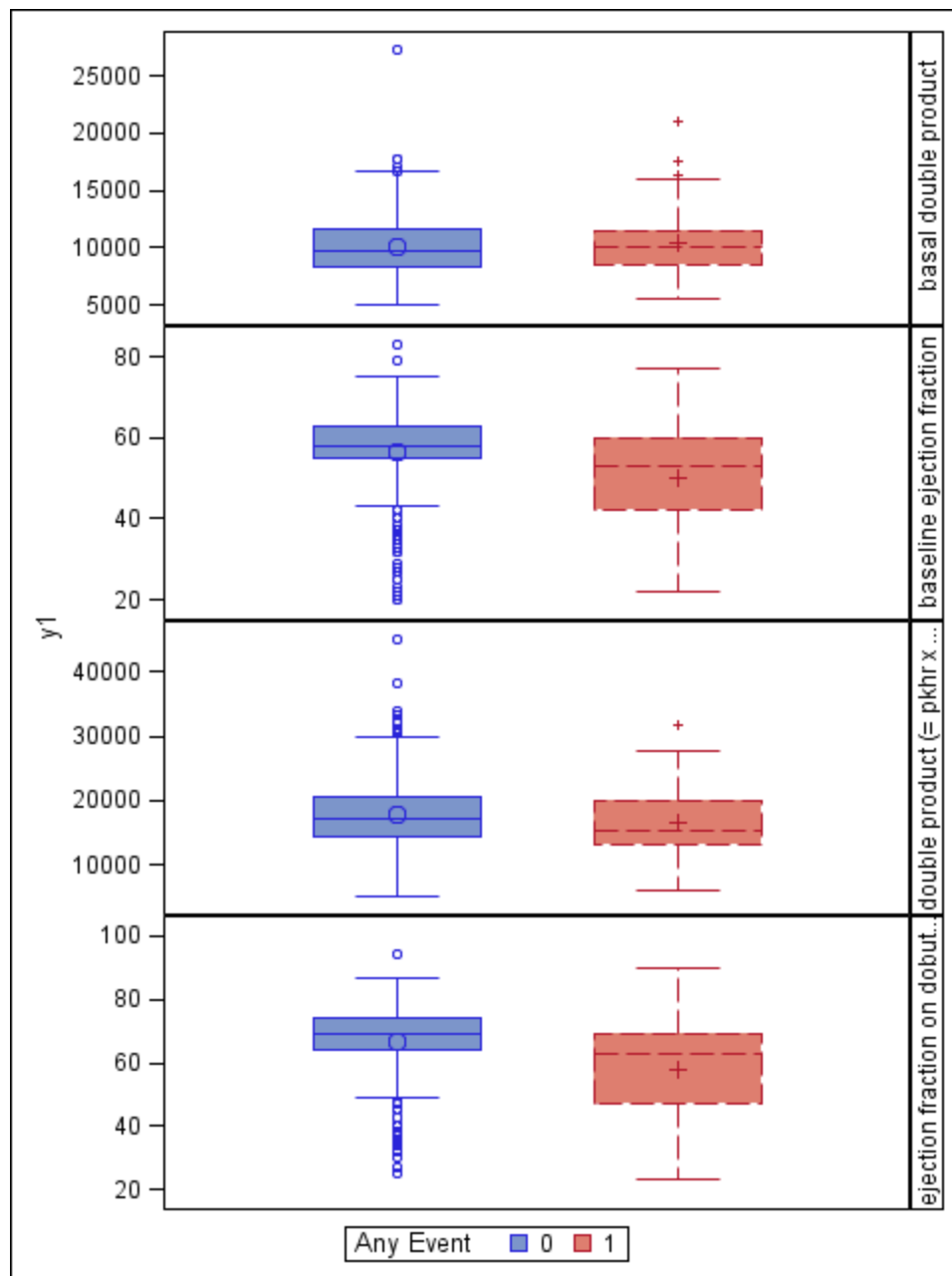
## LONG FORMAT CONVENIENT FOR MAKING GRAPHS TOO

It turns out that the long data format is very convenient for making graphs with the SG PANEL procedure. The following code produces box plots for each continuous variable by the grouping variable &xgroup:

```
* With long/transposed data, make boxplots for each continuous variable ;
proc sgpanel data=long_cont1;
  panelby &vlabel / onepanel rows=4 layout=rowlattice novarname
  uniscale=column;
  vbox y1 / group=&xgroup;
run;
```

See Schwartz (2009) for additional details in using PROC SG PANEL. The results of the SG PANEL procedure call is Figure 1





**Figure 1: Box plots with unequal y-scales produced with SGPPANEL procedure**

In conclusion, we present a macro to summarize and compare categorical data between two groups but the methods in this paper can be easily extended to more than two groups. Transposing data from wide to long format allows us to take advantage of the strong by processing and the output delivery system in SAS. We allow the user to choose between a Chi-square test or a Fisher's exact test for comparing categorical variables between groups. The p values are set up for two groups at least for the continuous variables but can be extended to multiple groups fairly easily.

## REFERENCES

Garfinkel, Alan, et. al. "Prognostic Value of Dobutamine Stress Echocardiography in Predicting Cardiac Events in Patients With Known or Suspected Coronary Artery Disease." Journal of the American College of Cardiology 33.3 (1999) 708-16.

How to Keep Multiple Formats in One Variable after Transpose. Proceedings of the NESUG. Paper available at: <http://www.nesug.org/Proceedings/nesug11/cc/cc36.pdf>. M. Wang. (2011)

King, John H. (2010), The Ubiquitous Clinical Trials Data Summary Table - "Summary Statistics in Rows", Proceedings of PharmaSUG 2010 - Paper TT05. <http://www.lexjansen.com/pharmasug/2010/tt/tt05.pdf>

Long, Stuart and Abolafia, Jeff (2008). Adventures in ODS: Producing Customized Reports Using Output from Multiple SAS® Procedures. Available at: <http://www2.sas.com/proceedings/forum2008/030-2008.pdf>

Susan Schwartz (2009). Clinical Trial Reporting Using SAS/GRAPH® SG Procedures. Proceedings of SAS Global Forum 2009. Available at: <https://support.sas.com/resources/papers/proceedings09/174-2009.pdf>

## ACKNOWLEDGMENTS

This paper uses data from the UCLA Stress Echocardiography data set (downloaded from: <http://biostat.mc.vanderbilt.edu/wiki/Main/DataSets>) . A description of the data is found at: <http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/stressEcho.html>. (Garfinkel 1999).

## CONTACT INFORMATION <HEADING 1>

Your comments and questions are valued and encouraged. Contact the author at:

Jeff Gossett  
University of Arkansas for Medical Sciences  
Department of Pediatrics  
Little Rock, Arkansas  
email: [gossettjeffrey@uams.edu](mailto:gossettjeffrey@uams.edu)

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.