

# Creating Clinical Trial Summary Tables Containing P-Values: A Practical Approach Using Standard SAS® Macros

Jun Zuo, Ph.D., STATPROBE, Inc., Ann Arbor, MI

Carl R. Haske, Ph.D., STATPROBE, Inc., Ann Arbor, MI

## ABSTRACT

P-value is a key criterion for evaluating the effectiveness and safety of new drugs in clinical trials, particularly in comparative studies. However, p-values are generally not presented in data summary tables generated with SAS software, because of the complexity of incorporating p-values into a formatted table that contains summary statistics, such as mean, proportion, or standard deviation.

The purpose of this paper is to present a practical approach to create summary tables containing p-values which are calculated using customized SAS macros developed at STATPROBE, Inc. Three major types of SAS macros are presented in this paper to calculate p-value, according to whether variables are categorical (Fisher's exact test, Cochran-Mantel-Haenzel procedure, or logistic regression), quantitative (t-test, Wilcoxon rank-sum test, or general linear models), or survival (parametric or nonparametric estimates). SAS statistical procedures, such as PROC FREQ, PROC NPAR1WAY, PROC CATMOD, PROC GLM, and PROC LIFETEST, provide the function part of the macros. The output locations of p-values from SAS statistical procedures are also considered for the purpose of effective creation of the macros and generation of the summary tables. The programming techniques to prepare the input data set and to merge data sets containing p-values and other summary statistics are discussed with examples.

## INTRODUCTION

P-value is a probability value commonly used in comparative clinical trials to test whether there exists a significant difference for a specific summary measure (e.g., mean, proportion, response, or survival time) among different treatment groups. It is a key criterion used in clinical trials to evaluate the effectiveness and safety of a new drug, a new medical product, or a new medical intervention. However, in general, most summary tables generated with SAS software contain only such common summary statistics as proportion, mean, median, or standard deviation. P-values are not presented in these summary tables because of the theoretical and technical complexity of generating and incorporating them into the tables, and also because of the separation of responsibilities between statisticians and programmers. Although a word processor can be used to combine the common summary statistics, computed mostly by programmers using Base SAS, and the p-values, calculated mainly by statisticians using SAS/STAT or SAS statistical macros (e.g., Ward and Haske 1996), it is still desirable for both programmers and statisticians to pursue a practical approach to generate summary tables containing p-values.

This paper presents a practical approach to create integrated summary tables containing p-values. Specifically, it discusses various standard and customized SAS macros developed to calculate different types of p-values, as well as some strategies and techniques for incorporating p-values into summary tables. The paper focuses on generating summary tables for the clinical trials with two independent treatment groups. However, the methods discussed here can be extended and applied to clinical trials with multiple dosing regimen or paired treatment groups by extending the corresponding SAS macros. Table 1 illustrates the specification of the summary tables that are the focus of this paper.

**Table 1.** Specification of Summary Table with P-Values

Report X (Page x of y)			
Title of Clinical Trial			
Title of Table			
Investigator: abc			
Variable	Treatment A (N = xx)	Treatment B (N= xx)	P-Value
Var1, N (%)	n <sub>a</sub> (%)	n <sub>b</sub> (%)	x.xxx
Var2			
MEAN	mean <sub>a</sub>	mean <sub>b</sub>	x.xxx
STD	std <sub>a</sub>	std <sub>b</sub>	
MEDIAN	med <sub>a</sub>	med <sub>b</sub>	
MIN-MAX	min <sub>a</sub> -max <sub>a</sub>	min <sub>b</sub> -max <sub>b</sub>	

The authors first discuss the classification of various p-values in terms of methodological considerations and output locations of p-values from SAS statistical procedures. The methodological considerations include:

- Type (categorical, quantitative, or survival) of key variable to be analyzed.
- Number of other variables (class variables, independent variables, or controlling variables) to be included.
- Concern about statistical distributions (parametric or non-parametric).

Output locations include three possible sources of the p-values from SAS statistical procedures:

- SAS data set created by procedures.
- Log file printed after using procedures.
- Calculation based on estimated parameters and covariance.

The classification provides an adequate framework for effective development of various standard and customized SAS macros for generating p-values.

Three groups of SAS macros are developed in the next section: function macros, procedure macros, and utility macros. A function macro corresponds to a specific statistical test to generate a specified p-value; it contains a SAS statistical procedure that may be integrated in a procedure macro and several data manipulation steps that may be captured by the utility macros. The third section discusses some programming techniques and examples for efficient and accurate use of those function macros, such as preparing input data sets, manipulating output data sets, and utilizing options in the macros. The concluding section provides some directions for building a complete standardized system to generate most summary tables for any clinical trial.

## CLASSIFICATION OF STATISTICAL TESTS (P-Values)

In this section, methodological aspects and output locations related to corresponding statistical tests and SAS statistical procedures are discussed and used to classify statistical tests, with a view to establish an adequate framework for effective development of various standard and customized SAS macros for generating p-values. The methodological aspects focus on differences in the statistical tests used to generate p-values in terms of differences in the variables analyzed and distributions assumed;

the output locations refer to the differences in output of p-values from various SAS statistical procedures.

Only those statistical tests dealing with comparative clinical trial studies for two independent treatment groups are considered in this paper. These statistical tests, or corresponding p-values, can be classified by three factors: types of key variables to be tested, number of other variables to be included, and the statistical distribution to be assumed. Table 2 displays the classification of statistical tests for generating p-values and the corresponding SAS procedures.

**Table 2.** Classification of Statistical Tests (or P-Values) for Clinical Trials with Two Independent Treatment Groups

Type of Key Variable	Other Variables Included	Statistical Tests (SAS Procedures)	
		Parametric	Nonparametric
Category	Treatment	Chi-square (Pearson or continuity adj.) (PROC FREQ)	Fisher's exact test (PROC FREQ)
	Treatment + Others	CMH with table score (PROC FREQ) Logistic Regression (PROC LOGISTIC or PROC CATMOD)	CMH with rank score (PROC FREQ)
Quantitative	Treatment	Unpaired t test (PROC TTEST)	Wilcoxon rank-sum test (PROC NPAR1WAY)
	Treatment + Others	GLM (PROC GLM)	GLM with rank-transformed data (PROC RANK + PROC GLM)
Survival	Treatment or Treatment + Others	Chi-square (PROC LIFEREG)	Log-rank test (PROC LIFETEST)
		Cox proportional hazards model (Semiparametric) (PROC PHREG)	

Three types of variables are regularly treated in statistical analysis for clinical trials: categorical, quantitative, and survival.

Categorical variables are characterized as having a limited number of discrete values that can be nominal, ordinal, or interval. Typical examples of categorical variables in clinical trials are measures of treatment outcomes (e.g., failure or success, NIH stroke scale score, Glasgow outcome score), investigator number, treatment group, race, sex, age group, and variables capturing whether a patient took a specific medication or had a specific adverse event. Chi-square statistic, Fisher's exact test, Cochran-Mantel-Haenzel (CMH) procedure, and logistic regression are commonly used to deal with such categorical variables.

Quantitative, or continuous, variables are defined as those that can be put into an infinite number of continuous values. In clinical trials, some treatment outcomes (e.g., rate of new episodes over a specific period) and most laboratory test parameters (e.g., RBC, WBC, SGOT, SGPT, and pH) are quantitative variables. Typical statistical tests used for quantitative variables include

unpaired t-test, Wilcoxon rank-sum test, and general linear models (GLM).

Survival variables are used to measure the duration of time for the occurrence of a specific event. Some common examples of survival variables are time to failure, time to a new episode, or time to withdraw from treatment. Survival data can be analyzed by Chi-square statistic, log-rank test, or the Cox proportional hazards model.

One major task of statistical analysis in a comparative clinical trial is to compare the difference of a specific measure (e.g., mean or proportion) between two treatment groups. The treatment variable alone may enter statistical models as an independent variable or a class variable. In this case, simple statistical tests and corresponding SAS statistical procedures can be applied to generate a single p-value. Unpaired t-test, Fisher's exact test, or Wilcoxon rank-sum test is usually used to deal with this case.

Other variables (either categorical or quantitative) often need to be included into the statistical models as additional independent variables, stratified variables, or controlling variables. The Cochran-Mantel-Haenzel (CMH) procedure, logistic regression, and general linear models (GLM) are the major tools applied, and multiple p-values associated with treatment and other variables may be produced in these cases.

Distributional assumption is the third factor for classifying and choosing statistical tests. Parametric tests depend on a specific distributional assumption, such as normal distribution, whereas nonparametric approaches perform tests without any strict distributional assumptions. Table 2 contains a detailed profile of parametric versus nonparametric tests that are commonly used in clinical trials with two independent treatment groups. In general, nonparametric methods are often as powerful as parametric methods, even if the data are distributed normally. In addition to parametric and nonparametric tests, some methods, defined as semiparametric tests, are partly based on a specific distribution assumption. One example is the Cox proportional hazards model, which is used in survival analysis and incorporated in this paper.

In the classification of statistical tests for effective development of SAS macros to obtain and incorporate various p-values, the output locations of p-values from SAS procedures must also be considered. Three possible ways to obtain p-values from SAS statistical procedures are shown in Table 3.

Some SAS statistical procedures output p-values into a SAS working data set by options such as OUT = <data set name> or OUTTEST = <data set name>. PROC FREQ and PROC GLM are two examples. By applying these SAS procedures, the p-values can be directly obtained and merged with other SAS data sets that contain summary statistics, such as mean, standard deviation, or proportion. However, not every SAS statistical procedure generates a SAS working data set that contains either p-values or other estimated parameters. For example, PROC TTEST, PROC NPAR1WAY, and PROC LIFEREG with class variables do not generate a SAS working data set at all. These procedures provide statistical results only in a so-called log file, which can be captured by an external hardware device such as screen, printer, or hard disk. In this case, a set of suitable SAS code can be created to output log information into a temporary ASCII file and to extract either p-values or parameters from this temporary file into a SAS data set.

Some SAS statistical procedures, instead of providing p-values directly, provide estimated parameters and covariance, from which p-values can be calculated. This paper demonstrates this situation for some SAS procedures, including PROC LOGISTIC, PROC CATMOD, and PROC LIFETEST.

**Table 3.** Classification of P-Values Based on Source

Source of P-Value	SAS Procedure	Statistical Tests
In SAS data set from procedure	PROC FREQ	Chi-square, Fisher's exact test, CMH procedure
	PROC GLM	With or without rank-transformed data
In log file from procedure	PROC TTEST	Unpaired t-test
	PROC NPAR1WAY	Wilcoxon rank-sum test
	PROC LIFEREG	Chi-square
Calculation	PROC LOGISTIC	Logistic regression
	PROC CATMOD	Logistic regression
	PROC LIFETEST	Log-rank test
	PROC PHREG	Cox proportional hazards model

## SAS MACROS

The classification presented in the previous section provides an adequate framework to effectively develop various standard and customized SAS macros for generating p-values. This section discusses three groups of SAS macros developed at STAT-PROBE:

- Function macros.
- Procedure macros.
- Utility macros.

The classification of p-values in the previous section is the basis for the creation of function and procedure macros.

### Function Macros

A function macro corresponds to a specific statistical test to generate a specified p-value. It contains an SAS statistical procedure, which may be integrated in a procedure macro, and several data manipulation steps, which may be captured by several utility macros. The p-value function macros developed at STAT-PROBE are shown in Table 4. Among them, the function macros for Fisher's exact test (%fisher), Wilcoxon rank-sum test (%wilcoxon), and log-rank test (%log\_rank) are exhibited in detail as follows.

#### 1. Fisher's exact test.

```
%Macro fisher(inds=, outds=, var=, class=, byvars=,
pvalue=);
%freq(test=exact, inds=&inds, outds=&outds, var=&var,
class=&class, byvars=&byvars, pvalue=&pvalue,
type=2)
%Mend wilcoxon;
```

#### 2. Wilcoxon rank-sum test.

```
%Macro wilcoxon(ind=, outds=, var=, class=, byvars=,
pvalue=);
%t_wilcox(proc=npar1way, ind=&inds, outds=&outds,
var=&var, class=&class, byvars=&byvars,
pvalue=&pvalue,)
%Mend wilcoxon;
```

**Table 4.** P-Value Function Macros

Function macro	Procedure macro or procedure	Statistical test (p-value)
Categorical data:		
%chisq_ca	%freq	Pearson chi-square
%fisher	%freq	Fisher's exact test
%cmh_ga	%freq	CMH: general association statistic
%cmh_cor	%freq	CMHJ: correlation statistic
%cmh_rms	%freq	CMH: row mean scores
%logit_ca	%log_cat	Logistic regression: categorical independent variables
%logit_qu		Logistic regression: quantitative independent variables
Quantitative data:		
%t_test	%t_wilcox	Unpaired t-test
%wilcoxon	%t_wilcox	Wilcoxon rank-sum test
%glm-raw	%glm	GLM: raw data
%glm_rank	%glm	GLM: rank-transformed data
Survival data:		
%chisq-su	PROC LIFEREG	Chi-square
%log_rank	PROC LIFETEST	Log-rank test
%cox_su	PROC PHREG	Cox proportional hazards model

#### 3. Log-rank test.

```
%Macro log_rank(ind=, outds=, timevar=, testvar=,
cenvar=, cenlst=, stravar=, byvars=, pvalue=);
%If &byvars^= %then %do;
  %sort(data=&ind, sortvars=&byvars);
%End;
Proc lifetest data=&ind outtest=&outds noprint;
  %If &cenvar^= %then %do;
    Time &timevar*&cenvar.&cenlst;
  %End;
  %Else %do;
    Time &timevar;
  %End;
  Test &testvar;
  %If &byvars^= %then %do;
    By &byvars;
  %End;
  %If &stravar^= %then %do;
    Strata &stravar;
  %End;
Run;
Data &outds;
  Set &outds;
  If _type_= 'LOG RANK' and
    _name_= %upcase("&timevar");
    &pvalue=1-probchi(&timevar,1);
  Keep &byvars &pvalue
Run;
%Mend lifetest;
```

## Procedure Macros

A procedure macro is based on a single SAS statistical procedure with multiple options to generate different p-values or a similar specification that may be used for multiple SAS statistical procedures. Three procedure macros are formulated here to cover most statistical tests discussed in the previous section, except statistical tests for survival data.

1. *Same procedure generating different p-values (PROC FREQ or PROC GLM).* The procedure macro based on PROC FREQ is displayed below. The option test=exact can be replaced with test=cmh. The option type=2 can be replaced with type=ga, cor, or rms.

```
%Macro freq(test=exact, inds=, outds=, var=, class=,
           byvars=, pvalue=pvalue, type=2);
%If &byvars^= %then %do;
  %sort(data=&inds, sortvars=&byvars);
  %Let bylist=%clist(&byvars);
  %Makeview(&inds,%quote(&bylist),byvar)
%End;
Proc freq data=&inds noprint;
  Tables &var*&class / &test;
  %If &byvars^= %then %do;
    By &byvars;
  %End;
  Output out=tempds &test;
Run;
%Dsexist(tempds);
Data &outds;
  %If %upcase(&exist)=TRUE %then %do;
    Set tempds;
    &pvalue=p_&test&type;
  %End;
  %Else %do;
    %If &byvars^= %then %do;
      Set byvars;
    %End;
    %End;
    Keep &pvalue &byvars;
  Run;
  %cleanup(tempds);
%Mend freq;
```

2. *Identical specification and process for different SAS procedures* (for PROC TTEST and PROC NPAR1WAY). This procedure macro is shown below. The option proc=ttest can be replaced with proc=npar1way.

```
%Macro t_wilcox(proc=ttest, inds=, outds=, var=, class=,
                byvars=, pvalue=pvalue);
%If &byvars^= %then %do;
  %sort(data=&inds, sortvars=&byvars);
  %sort(data=&inds nodupkey out=byvars,
        sortvars=&byvars);
  Data byvars;
    Set byvars;
    byid=_n_;
    Keep &byvars byid;
  Run;
  Data tempset;
    Merge &inds byvars;
    By &byvars;
  Run;
%End;
%Else %do;
```

```
Data tempset;
  Set &inds;
  Run;
%End;
Filename tempfile 'pvalue.txt';
Proc printto print=tempfile new;
Proc &proc data=&inds
  %If %upcase(&proc)=NPAR1WAY %then %do;
    wilcoxon
  %End;;
  Class &class;
  Var &var;
  %If &byvars^= %then %do;
    By byid;
  %End;;
Run;
Proc printto Run;
Data &outds;
  Infile tempfile print length=linelen;
  Input linein $varying180. linelen;
  Length pvalue 8. readline $180;
  %If &byvars^= %then %do;
    If index(linein,'BYID=')>0 then do;
      readline=substr(linein,index(linein,
        'BYID=')+5);
      byid=scan(readline,1,' ');
    End;
  %End;
  %If %upcase(&proc)=TTEST %then %do;
    If index(linein,'Equal ')>0 then do;
      pvalue=scan(linein,11,' ');
      output;
    End;
  %End;
  %If %upcase(&proc)=NPAR1WAY %then %do;
    If index(linein,'|Z| =')>0 then do;
      readline=substr(linein,index(linein,
        '|Z| =')+6);
      pvalue=scan(readline,1,' ');
      output;
    End;
  %End;
Run;
Data &outds;
  %If &byvars^= %then %do;
    Merge &outds (in=a) byvars;
    By byid;
    If a;
  %End;
  %Else %do;
    Set &outds;
  %End;
  &pvalue=pvalue;
  Keep &pvalue &byvars;
Run;
%Mend t_wilcox;
```

3. *Almost identical specification for different SAS procedures* (for PROC LOGISTIC and PROC CATMOD). The following is the code of this procedure macro, where the option proc=logistic can be replaced with proc=catmod. This procedure macro is an example of calculating p-value based on estimated parameters and covariance. The p-values are represented in the output data set with the same variable names as in the list of effect-variables (effvars=). Note that

the micro variables &num\_vars and &&varnm&l are generated by utility macro %varlist.

```
%Macro log_cat(proc=logistic, inds=, outds=pvalue, var=,
effvars=, byvars=);
%If &byvars^= %then %do;
  %sort(data=&inds, sortvars=&byvars);
%End;

Proc &proc data=&inds
  %If %upcase(&proc)=LOGISTIC %then %do;
    outest=&outds covout
  %End;;
  Model &var=&effvars;
  %If %upcase(&proc)=CATMOD %then %do;
    Response logit / outest=&outds;
  %End;
  %If &byvars^= %then %do;
    By &byvars;
  %End;
Run;
%varlist(&effvars);

Data params;
  Set &outds;
  If _TYPE_='PARMS';
  %If %upcase(&proc)=CATMOD %then %do;
    %Do i=1 %to &num_vars;
      %Let j=%eval(&i+1);
      Rename b&j=&&varnm&i;
    %End;
  %End;
  Drop _type_ _name_;
Run;
Data covars;
  Set &outds;
  If _type_='COV';
  If _name_ in ('B1','INTERCEP') then delete;
Run;
%sort(data=covars,sortvars=&byvars _type_ _name_);
Data covars;
  Set covars;
  By &byvars _type_ _name_;
  Retain cov1-cov&num_vars;
  %Do i=1 %to &num_vars;
    %If %upcase(&proc)=LOGISTIC %then %do;
      If _name_=%upcase(&&varnm&i)
        then cov&i=&&varnm&i;
    %End;
    %If %upcase(&proc)=CATMOD %then %do;
      %Let j=%eval(&i+1);
      If _name_="B&j" then cov&i=b&j;
    %End;
  %End;
  If last._type_ then output;
  Keep &byvars cov1-cov&num_vars;
Run;
%If &byvars^= %then %do;
  %sort(data=params, sortvars=&byvars);
  %sort(data=covars,sortvars=&byvars);
%End;

Data &outds;
  Merge params covars;
  %If &byvars^= %then %do;
    By &byvars;
  %End;
  %Do i=1 %to &num_vars;
```

```
If cov&i>0 then &&varnm&i=
  1-probchi(&&varnm&i**2/(cov&i),1);
Else &&varnm&i=.;
If &&varnm&i>1 then &&varnm&i=.;
%End;
Keep &byvars
%Do i=1 %to &num_vars;
  &&varnm&l
%End;;
Run;
%cleanup(params covars);
%Mend log_cat;
```

## Utility Macros

The processes for generating different p-values may include some of the same data manipulation steps. Utility macros are created to efficiently capture those commonly used data manipulation steps that contain data steps, data procedures, or both. At STATPROBE, these utility macros are either in a general macro library or a specific project macro library. Table 5 displays standard macros often used in generating p-values.

**Table 5.** Major Utility Macros Used in This Paper

Macro Name	Purpose
%cleanup	To clean up SAS data sets in a working directory
%clist	To insert commas into a blank delimited text
%dsexist	To check whether a SAS data set exists or not
%makeview	To create a table view containing distinct values of selected variables
%sort	To sort a SAS data set
%varlist	To obtain the number of words and content of each word in a blank delimited text string

## MACRO USES AND EXAMPLES

### Input Data Sets

Each function macro has a option to indicate an input data set that contains data to be analyzed. The numbers of observations and numbers of key variables to be analyzed allow for two ways of arranging the data set:

1. The number of observations equals the number of patients included in analysis, and all key variables exist in the data set.
2. The number of observations equals the product of the number of patients and the number of key variables, and only one variable will be tested.

In the first case, the function macro is applied multiple times for each of the key variables, given that there is more than one key variable to be analyzed. However, sometimes it is inconvenient to arrange a data set in this way, as with event data sets for medications or adverse events, which contain an uncertain number of events. In such cases, the variable(s) representing the events can be included in by-variables, and the corresponding function macro needs to be run only once to a variable to indicate whether a patient has an event or not. In dealing with event data, we should note that the original event data set is not complete, because a patient will not have an event record if the patient did not experience any events. Some function macros, e.g., the macro of Fisher's exact test, cannot be used correctly unless those patients who did not have a record for an event are added into the data set with a value, usually zero, representing that the patient did not experience any events.

## Output Data Sets

Output data sets contain the calculated p-values associated with by-variables. When an output data set is merged with a data set containing other summary statistics, such as mean, standard deviation, or proportion, the treatment variable should not be included in by-variables, so that the p-value can be merged in both records of the treatment group.

## Examples

Only one example, summary table of patient characteristics, is presented in this paper due to space limit. The summary table is based on patient characteristic data set which contains the following variables: prot (protocol number), inv (investigator number), pt (patient number), race, sex, age, height, weight, r1-r3 (count for Caucasian, black, or other, respectively), r4-r5 (count for male or female), r6-r8 (count for age <1, 1-2, or >2, respectively), r9 (=age), r10 (=height), and r11 (=weight).

The main SAS code (integrated into STATPROBE's standard programming system) is shown below.

```
%Let ds1=patient;
%readdata(&ds1);
%rxgrp(&ds1);
%strandata(&ds1);
%calcstat(inds=&ds1, outds=stats, vars=r1-r11,
byvars=prot inv rxgrp, overall=YES);
%percent(stats);
%fisher(inds=&ds1, outds=pval1, var=race, class=rxgrp,
byvars=prot inv, pvalue=pvalue1);
%fisher(inds=&ds1, outds=pval4, var=sex, class=rxgrp,
byvars=prot inv, pvalue=pvalue4);
%fisher(inds=&ds1, outds=pval6, var=age, class=rxgrp,
byvars=prot inv, pvalue=pvalue6);
%Macro dowilcox;
%Do i=9 %to 11;
%wilcoxon(inds=&ds1, outds=pval&i, var=r&i,
class=rxgrp, byvars=prot inv,
pvalue=pvalue&i);
%End;
%Mend dowilcox;
%Dowilcox;
Data stats;
Merge stats pval1 pval4 pval6 pval9 pval10 pval11;
By prot inv;
Run;
%report(inds=stats, sortvars=prot inv rxgrp, <other
options>);
%cleanup(&ds1 stats pval1 pval4 pval6 pval9 pval10
pval11);
```

The summary table generated by use of the above code appears in Appendix.

## CONCLUSION

This paper presents a practical approach to creating clinical trial summary tables containing p-values. The standard and customized SAS macros are developed and integrated into the existing system at STATPROBE, Inc. Both statisticians and programmers can use this system to generate any designated summary tables.

Further enhancement of both SAS macros and system, which are used to generate listing and summary tables, is under consideration in order to obtain a more user-friendly interface. SAS/AF is an ideal tool for reaching our next goal: to establish a comprehensive and user-friendly clinical trial table-generating system.

## REFERENCES

Agresti, Alan (1990), *Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Marubini, Ettore, and Valsecchi, Maria G. (1995), *Analysing Survival Data from Clinical Trials and Observational Studies*, New York: John Wiley & Sons, Inc.

SAS Institute, Inc. (1990), *SAS/STAT User's Guide, Version 6, Fourth Edition*, Cary, NC: SAS Institute Inc.

Ward, Vivienne, and Haske, Carl R. (1996), "A Statistical Analysis Macro Library in SAS Software," *Proceedings of the Twenty-First Annual SAS Users Group International Conference*, 1156-1160.

## ACKNOWLEDGEMENT

SAS, SAS/STAT, and SAS/AF are registered trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

## AUTHORS' ADDRESSES

Jun Zuo, Ph.D.  
STATPROBE, Inc.  
3885 Research Park Drive  
Ann Arbor, MI 48108  
Phone: (313) 769-5000 x139  
E-Mail: jzuo@statprobe.com or jzuo@ix.netcom.com

Carl R. Haske, Ph.D.  
STATPROBE, Inc.  
3885 Research Park Drive  
Ann Arbor, MI 48108  
Phone: (313) 769-5000 x115  
E-Mail: chaske@statprobe.com

## APPENDIX Sample Summary Table

SUMMARY OF PATIENT CHARACTERISTICS			
INVESTIGATOR: OVERALL			
CHARACTERISTIC	DRUG 1 (N = 128)	DRUG 2 (N= 122)	P-VALUE*
RACE, N (%)			
CAUCASIAN	110 (85.9)	108 (88.5)	0.232
BLACK	13 (10.2)	6 (4.9)	
OTHER	5 (3.9)	8 (6.6)	
SEX, N (%)			
MALE	74 (57.8)	68 (55.7)	0.799
FEMALE	54 (42.2)	54 (44.3)	
AGE (YEARS), N (%)			
<1	65 (50.8)	47 (38.5)	0.057
1-2	45 (35.2)	45 (36.9)	
>2	18 (14.1)	30 (24.6)	
MEAN	1.2	1.4	0.037
SD	0.7	0.7	
MEDIAN	1.0	1.2	
MIN-MAX	0.5-3.7	0.5-3.2	
HEIGHT, CM			
MEAN	76.2	78.3	0.043
SD	9.0	9.6	
MEDIAN	74.0	76.0	
MIN-MAX	48-99	32-97	
WEIGHT, KG			
MEAN	10.2	10.7	0.079
SD	2.2	2.2	
MEDIAN	9.9	10.5	
MIN-MAX	6.1-17.4	6.4-16.3	

REFERENCE: LISTING X

\* P-VALUE COMPARING THE TREATMENT GROUPS USING FISHER'S EXACT TEST FOR CATEGORICAL FACTORS AND WILCOXON RANK SUM TEST FOR CONTINUOUS VARIABLES.