Report Macro Documentation

November 2018, Updated April 2024

Contents

1 Introduction

This document provides documentation on how to use a set of macros that produce customized report tables. These macros are used extensively by statisticians at the Division of Biostatistics at the University of Minnesota to generate statistical reports for DSMB and other purposes. These macros have been presented at several SAS conferences over the years (first presented in 1994: "A Set of SAS Macros for Producing Customized Reports" by Greg Grandits and Ken Svendsen, published in the 19th SAS User's Group International). These macros originally produced text based reports but have been enhanced to generate html tables which can be inserted into Word, posted as Web pages, or converted to PDF files using external HTML to PDF converters.

Briefly, producing customized reports is a high priority in both research and business settings. SAS has several procedures that generate reports, but only the PUT statement within a data step gives complete flexibility to placement of text and data values. The advent of ODS, while helpful, still does not provide the flexibility needed, leaving the data step as the alternative. However, data step processing and writing PUT statements are cumbersome to program. The described macros, working behind the scenes do all the tedious work, allow the user to produce customized reports that are easy to program and give complete flexibility to placement of text and data values. Information that can be placed on the report page include text, summary statistics (counts, means, etc.) and more complex statistics (regression coefficients, hazard ratios, etc.) and significance levels (P-values)

The user first defines columns and column widths across the report page. Text or data values (usually summary statistics) are then moved to these columns and specified lines using macros MOVE or NMOVE. Items can be moved to single columns or to combined columns with the items automatically centered. Lines to where items are moved can be listed or a starting line can be given along with the number of lines to be printed with optional skip patterns. Summary statistics are accessible using one of several available macros that run SAS procedures and output information to data sets. Statistics are placed in array type names (e.g., M1-M20) which can be moved to the report page after a SET statement. An example of a call is %nmove(m1-m20, col=2 3, line=12L10).

The enhancements to make html tables allow formatting such as font type, size, and shading of cells. In addition, multiple statistics can be placed in one column such as mean ± SD giving a journal finished appearance.

2 Steps in Writing Report

Report programs are made up of the following statements:

%wreport statement that indicates a new report is starting.

%colset statement that defines columns and column widths across the report page. This macro needs to be called before any move statements and is called once for each report.

%move statements which move text to the report. Formatting features include centering, underlining, bolding, and repeating text. Font size and type can also be specified.

set statement(s) that read in statistical information from a SAS data set. These one observation data sets will have been generated from one of the statistical generating macros described later.

%nmove statements which place the statistical information to the report page.

3 Report Macros

%wreport is used to indicate the start of a new report.

Example: %wreport (htmlfile = 'out.htm') will write the table to the indicated file.
Example: %wreport (htmlfile = 'out.htm' mod) mod option appends to existing file.

%colset is used as follows:

%colset(columns1 size columns2 size ...)

Example:%colset(27 9 9 2x 9 9)

This statement sets up 5 columns. The first column is three times longer than each of the other four columns. The 2x will separate colums 2-3 from columns 4-5. The size of the columns are only relative to the total size.

%move is used as follows:

%move ('string1':'string2', line=, col=, center=, under=, pat=, mfirst=);

'string1':'string2':'stringN' character strings enclosed by single or double quotes and separated by colons. **Example: 'Men':'Women':'Total'** (3 text strings)

line = line numbers to which strings are moved. This can be given in two ways.
line = 12 21 33 moves text strings to lines 12, 21, and 33
line = 12L3 moves text strings to lines 12, 13, and 14

col = column numbers to which text strings are moved. This can be given in 3 ways.
col = 3 4 8 moves three strings to defined columns 3,4, and 8
col = 2-3 4-5 6-7 moves strings to columns formed by combining columns 2-3, 4-5, and 6-7

col = 2.4 moves strings to columns 2 through 4 (shortcut for col=2 3 4)

center = set to Y or 1 to center text, N or 2 to left justify, 3 to right justify. Default is Y

under = set to Y or 1 to underline text, N if not. Default is N. Ignored for html reports.

pat = Used to skip lines; used with second method of specifying lines
line=10L15=
pat=5th/1 skips one line after every 5th line

mfirst= determines whether lines or columns move fastest when placing text; set to L for lines

and C for Columns. Default is L.

Note: HTML parameters are described in section on HTML reports.

Example: %move('Total Cholesterol':'LDL-C':'HDL-C', col=1, line=10L3 20L3)

This statement would produce the following text in line 10-12 and 20-22. Note that the text strings are repeated until the total lines are exhausted.

Total Cholesterol
LDL-C
HDL-C

Total Cholesterol
LDL-C
HDL-C

%nmove is used as follows:

%nmove (varlist, line=, col=, pat=, mfirst=, fmt=, scaler=, specfmt=);

The line, col, pat, and mfirst parameters are identical to those in %move

varlist= list of variables to move to report; two formats are allowed
var1 var2 var3 a list separated by spaces
var1-varN a starting and ending variable with same root name; n variables to be moved

fmt= used to format the numeric values
fmt=6.2 values are displayed with 2 decimals

scaler= multiplies values by this number before printing. Useful for converting fractions to percentages and vice vera.
scaler=100 multiplies values by 100 before printing

specfmt= used to apply a SAS format to the value
specfmt=pvalue5.3 uses the SAS internal p-value format to display the values

4 Procedure Macros

Here is a summary of each macro and statistics available to place on a report

Macro Name	SAS Procedure	Description
breakdn	summary	Stores descriptive statistics for a list of analysis variables by levels of 1 or more class variables
pchisqp	freq	Stores Pearson Chi-square tests and p-values for the association between a list of dependent categorical variables and an independent categorical variable.
chisqp	freq	Stores Cochran-Mantel-Haenszel Chi-square tests and p-values for the association between a list of dependent categorical variables and an independent categorical variable, adjusted (optionally) for a set of other categorical covariates.
cmhp	freq	An updated version of chisqp. Call is the same.
cmhpx	freq	A version of cmhp that has a parameter that requires a minimum number of events for the analyis
freqdis	freq	Stores counts, percents, cumulative counts, and cumulative percentages for a class variable for each level of a second class variable
fisherp	freq	Stores p-values from Fisher's exact test for a list of dependent variables and one independent variable.
barnardp	freq	Stores p-values from Barnard's exact test for a list of dependent variables and one independent variable. Computer time can be high.
glmp	glm	Stores statistics from GLM for 1 or more independent variables for a list of dependent variables. Statistics include p-values from factors in the model, least square means, contrasts and p-values from the contrasts.
logistp	logist	Stores statistics from LOGIST for 1 or more independent variables for a list of dependent variables. Statistics include estimates, SE estimates, odds ratios, and confidence intervals.
mixedp	mixed	Stores statistics from MIXED for 1 or more independent variables for a list of dependent variables. Statistics include p-values from factors in the model, contrasts and p-values from the contrasts.
phregp	phreg	Stores estimates from PHREG for 1 or more independent variables for a list of dependent variables. Statistics include estimates, SE of estimates, hazard ratios, and confidence intervals.
regp	reg	Stores regresion coefficients, standard errors, t-statistics, and p-values for 1 or more independent variables for a list of dependent variables.
lifetab		Macro LIFETAB reads the data set output in the OUTSURV option of PROC LIFETEST and outputs the survival values for specified time values for each strata.
genmodp	genmod	Stores statistics from PROC GENMOD.

4.1 BREAKDN

BREAKDN uses PROC SUMMARY to compute descriptive statistics for a list of analysis variables for each combination of a list of class variables and outputs them to a one observation data set. The statistics are contained in array like names as described below.

Statistics Available	**Names on Output Data Set**
N	n1-n?
MEAN	m1-m?
MEDIAN	med1-med?
STD	s1-s1?
STDERR	se1-se?
MIN	min1-min?
MAX	max1-max?
SUM	sum1-sum?
25th Percentile	Lq1-Lq?
75th Percentile	uq1-uq?

where ? is the product of the number of analysis variables and number of combinations defined by the class variables

Example Call

%breakdn (data=rand, class=gender 2T group 6, var = age sbp dbp, out=out1)

Note: Class factors must be sequential and start with the number 1.

Macro with default and optional parameters listed

%breakdn (data=_last_, class=, var =, sfirst=var, where=, print=no, out=out, nzero=)

Description of parameters

data	SAS data set read
class	List of class variables. Statistics for the variables listed in VAR are calculated for each combination of class levels and optionally for subtotals and grand totals. Each class variable is listed by the number of levels (starting with one through the number specified). A "T" appended to the number indicates that totals are desired. Example: class=gender 2T group 6

There are two class variables, gender and group with 2 and 6 levels, respectively. Statistics for each level of gender (and total) by each level of group will be computed and stored in a data set (18=3*6 combinations in the order gender=1, group=1,2,3,4,5,6 followed by gender=2, group=1,2,3,4,5,6 followed by combined genders, group=1,2,3,4,5,6. Note that the class variable to the right moves the fastest.

var= List of variables that statistics are to be computed

Example: var=age dbp sbp

sfirst= Specifies the order in which the array elements of the statistics are stored. The default is SFIRST=VAR which means that for a fixed combination of the class variables, statistics for each of the variables listed in VAR are stored consecutively. By setting SFIRST=CLASS, for a fixed variable listed in VAR, statistics for all combinations of the class variables will be stored consecutively. The following examples illustrate the effect of SFIRST on the order the elements are stored.

Examples:

%breakdn(data=rand, class=gender 2T, var=age dbp dbp, out=t1, sfirst=var)

n's for age, dbp, and sbp for gender=1 are stored in n1-n3
n's for age, dbp, and sbp for gender=2 are stored in n4-n6
n's for age, dbp, and sbp for gender=total are stored in n7-n9

%breakdn(data=rand, class=gender 2T, var=age dbp dbp, out=t1, sfirst=class)

n's for age for gender=1,2, and total are stored in n1-n3
n's for dbp for gender=1,2, and total are stored in n4-n6
n's for sbp for gender=1,2, and total are stored in n7-n9

The statistics are stored similarly for the other statistics.

where= optional parameter which allows the user to add a WHERE clause to the call.
Example: where = where ondrug = 1

out= SAS data set to which statistics are written

print= if set to yes, the output from PROC SUMMARY is displayed. Used to verify results.

nzero = if set to y then missing n's get the value 0 rather than period (.)

4.2 PCHISQP

PCHISQP uses PROC FREQ to compute standard Perason Chi-square tests and p-values for the association between a list of dependent categorical variables and an independent variable. These statistics are output to a one observation data set. The statistics are contained in array like names as described below.

Statistics Available	**Names on Output Data Set**
P-value	p1-p?
Chi-aquare statistic	chisq1-chisq?
DF	df1-df?

where ? is the number of dependent variables.

Example Call

%pchisqp (data=rand, dlist=fever nausea headache, group=trt, pdata=p)

This would generate the following PROC FREQ

```
PROC FREQ DATA=rand;
TABLES site*trt*(fever nausea headache)/chisq;
```

Macro with default and optional parameters listed

%pchisq (data=_last_, dlist=, group=, where=, pdata=)

Description of parameters

data=	SAS data set read
dlist=	List of categorical dependent variables
group=	Categorical variable to be tested for association with each dependent variable
where	optional parameter which allows the user to add a WHERE clause to the call.
	Example: where = where ondrug = 1
pdata =	SAS data set to which statistics are written

4.3 CHISQP/CMHP/CMHPX

CHISQP uses PROC FREQ to compute Cochran-Mantel-Haenszel Chi-square tests and p-values for the association between a list of dependent categorical variables and an independent variable, adjusted (optionally) for a set of categorical covariates. These statistics are output to a one observation data set. The statistics are contained in array like names as described below. The CMHP macro is a newer version of CHISQP which call is the same. The CMHPX macro has an added parameter called MINEVENTS, which is the minimum number of events required to perform the analysis. If the number of events is less than this number then the analysis is not done and the statistics are set to missing.

Statistics Available	**Names on Output Data Set**
P-value	p1-p?
Chi-square statistic	chisq1-chisq?

where ? is the number of dependent variables.

Example Call

%chisqp (data=rand, dlist=fever nausea headache, group=trt, clist=site, pdata=p)

This would generate the following PROC FREQ

```
PROC FREQ DATA=rand;
TABLES site*trt*(fever nausea headache)/cmh noprint;
```

Macro with default and optional parameters listed

%chisqp (data=_last_, dlist=, group=, clist=, test=G, where=, pdata=)
%cmhp (data=_last_, dlist=, group=, clist=, test=G, where=, pdata=)
%cmhpx (data=_last_, dlist=, group=, clist=, test=G, where=, pdata=, minevents=1)

Description of parameters

data=	SAS data set read
dlist=	List of categorical dependent variables
group=	Categorical variable to be tested for association with each dependent variable
clist=	List of categorical variables to be used as covariates
test=	Type of Chi-square test. The default is test=G, which does the General Association test. Set test=C for Correlation (1 df) and test=R for Row Means Differ

where optional parameter which allows the user to add a WHERE clause to the call.

Example: where = where ondrug = 1

pdata = SAS data set to which statistics are written

4.4 FREQDIS

FREQDIS uses PROC SUMMARY to obtain counts, percents, cululative counts, and cumulative percents for a class variable for each level of another class variable (if given) and outputs them to a one observation data set. This macro is useful for displaying the distribution of a variable for each level of another variable.

Statistics Available	**Names on Output Data Set**
Counts	n1-n?
Cumulative counts	cn1-cn?
Percents	p1-pn?
Cumulative percents	cp1-cpn?

where ? is the product of the number of levels of the class variables.

Example Call

%freqdis (data=rand, class=gender 2T agecat 5T, out=out1)

Macro with default and optional parameters listed

%freqdis (data=_last_, class=, where=, out=&class)

Description of parameters

data= SAS data set read

class= List of up to two class variables for which counts, percents, cumulative counts, and cumulative percents are computed. Each class variable is followed by the number of levels (needs to start with one and continue through the number specified). A "T" appended to the number signifies that totals are also included. When two class variables are specified, the statistics are calculated for each level of the first class variable.

Example: class = gender 2T agecat 5T

n1-n6 contain the counts for each of the 5 age categories and the total for gender=1.
n7-n12 contain the counts for each of the 5 age categories and the total for gender=2.
n13-n18 contain the counts for each of the 5 age categories and the total for both genders combined.

p1-p18 contain the percents as above.
cn1-cn18 contain the cumulative counts as above.
cp1-cp18 contain the cumulative perents as above.

where optional parameter which allows the user to add a WHERE clause to the call.

Example: where = where ondrug = 1

out = SAS data set to which statistics are written

4.5 FISHERP

FISHERP runs PROC FREQ and stores p-values from Fisher's exact test for a list of outcome variables and a group variable.

Statistics Available	**Names on Output Data Set**
P-Values	p1-p?

? is number of dependent variables.

Example Call

%fisherp(data=prevail, dlist=sae malaria, group=vaccine, out=out1);

Macro with default and optional parameters listed

%fisherp(data=, dlist=, group=, out=, where=, minevents=1);

Description of parameters

data=	SAS data set read
dlist=	list of dependent categorical variables with two levels
group=	independent group variable.
out=	output dataset containing p-values
where=	complete where clause including WHERE.
minevents=	minimum number of events to run test. If events are less than this number the pvalue is set to missing.

4.6 BARNARDP

BARNARDP runs PROC FREQ and stores p-values from Barnard's exact test for a list of outcome variables and a group variable. This call is computer intensive.

Statistics Available	**Names on Output Data Set**
P-Values	p1-p?

? is number of dependent variables.

Example Call

%barnardp(data=prevail, dlist=sae malaria, group=vaccine, out=out1);

Macro with default and optional parameters listed

%barnardp(data=, dlist=, group=, out=, where=);

Description of parameters

data=	SAS data set read
dlist=	list of dependent categorical variables with two levels
group=	independent group variable.
out=	output dataset containing p-values
where=	complete where clause including WHERE.

4.7 GLMP

GLMP runs PROC GLM and stores statistics generated from the analyses into one or more SAS data sets. Statistics include p-values from factors in the model (class variables or non-class variables), least square means and contrasts and p-values from the contrasts.

Statistics Available	**Names on Output Data Set**
For independent variables	
P-Value from ANOVA	p1-p?
F-statistic from ANOVA	f1-f?
If lsmean statement used	
Adjusted (LSM) Means	lsm1-lsm?
LSM Standard Errors	lsmse1-lsmse?
If estimate statement used	
Estimate	e1-e?
Standard errors of estimate	se1-se?
T-value for estimate	t1-t?
DF from contrast	df1-df?
P-value from contrast	p1-p?
If contrast statement used	
F-value for contrast	f1-f?
DF from contrast	df1-df?
P-value from contrast	p1-p?

See below for value of ? for each of the types of statistics and how they are stored

Example Calls

%let listd = chold hdld ldld trigd;
%let listb = cholbl hdlbl ldlbl trigbl;
%glmp (data=rand, class=group, dlist=&listd, ilist=group age, blist=&listb, factor=group, pdata=p, lsmean=yes, lsdata=lsmeans);

%glmp(data=rand, class=trt, dlist=&listd, factor=trt, pdata=p, estimate = '1 v 3' trt 1 0 -1/'2 v 3' trt 0 1 -1, constrast=%str('2 df test' trt 1 0 -1, trt 0 1 -1), estdata=e, condata=c);

Macro with default and optional parameters listed

%glmp (data=_last_, class=, dlist=, ilist=, blist=, factor=, where=, pdata=&factor, lsmean=no, lsclass=&class, lsdata=&lsclass, estimate=, estdata=, contrast=, condata=);

Description of parameters

data= SAS data set read

class= class variable(s) for GLM. List all variables that are to be treated as categorical variables.

dlist= dependent variable list. An analysis is run for each variable

ilist= independent variable list. This list is used for each dependent variable given in dlist.

blist= optional list of variables where the i'th variable in the list is added as a covariate to the i'th analysis. It is used mainly to adjust for the baseline value of the dependent variable.

Note: if blist is not given then only one GLM is run with multiple dependent variables in the model statement. If blist is given then a separate GLM is run for each dependent variable

factor independent variable(s) for which statistics from the ANOVA table are obtained.

pdata SAS dataset(s) in which statistics are stored for the variables listed in factor. There is one dataset per factor with one observation in each dataset.

lsmean= set to YES if least square means are desired

lsclass = the class variable for which LS means are desired

lsdata= SAS dataset containing the LS means. This dataset contains one observation with the following variables:

lsm1-lsm? = the LS means
lsmse1-lsmse? = the standard errors of the LS means

? = the number of levels of the class variable times the number of dependent variables. For example, if the class variable has 5 levels and there are 2 dependent variables then ?=10; LS1-LS5 contain the LS means for the first dependent variable and LS6-LS10 contain the LS means for the second dependent variable.

Note: If you have 2 or more class variables and request LS means for one of them then PROC GLM does not give the "correct" results (the adjusted means it computes assume a balanced design). In this case define indicator variables for the levels of the additional class variables and add them as independent variables in the model. The means of the adjusted means will then be equal to the overall unadjusted mean.

estimate If given, an estimate statement(s) is generated to perform 1 df contrasts. Multiple contrasts are separated by a slash (/).

Example: estimate = '1 v 3' trt 1 0 -1/'2 v 3' trt 0 1 -1

This would perform two contrasts, the first comparing the first and third levels of trt; the second comparing the second and third levels of trt.

estdata The dataset the estimates are stored. The variables are:

e1-e? contrast (estimate) values
se1-se? standard errors of the estimates
t1-t? t-statistics
p1-p? p-values

? is the total estimate statements times the number of dependent variables. They are stored in the order generated. If there are 2 dependent variables and 2 contrasts then:

e1 = estimate for analysis 1, contrast 1
e2 = estimate for analysis 1, contrast 2
e3 = estimate for analysis 2, contrast 1
e4 = estimate for analysis 2, contrast 2

contrast= This generates the contrast statement in GLM to perform >1 df contrasts. Contrasts are separated by a slash (/).

Example: Contrast = %str('2 df test' trt 1 0 -1, trt 1 -1 0)

The %str function is required because a comma is used in the parameter.

condata= The dataset the contrasts are stored. The variables are:

f1-f? F-statistics for contrasts
p1-p? P-values for contrasts
df1-df? Degrees of freedom of the contrast

? is the total number of contrast statements times the number of dependent variables. They are stored in the same order as the estimate statements (see above).

where= Optional parameter which allows the user to add a WHERE clause to the call

Example: where = where ondrug eq 1

4.8 LOGISTP

LOGISTP runs PROC LOGIST to perform logistic regression and saves results for factors of interest into SAS datasets. Results for each factor is placed in a separate dataset.

Statistics Available	**Names on Output Data Set**
For class variables	
Wald Chi-Square statistic	chisq1-chisq?
P-value from Chi-square	p1-p?
For non-class variables	
Estimate (beta coefficient)	e1-e?
Standard error of estimate	se1-se?
Z-statistic (beta/SE)	z1-z?
P-value for estimate	p1-p?
Odds ratio	o1-o?
Upper bound of 95% CI	u1-u?
Lower bound of 95% CI	L1-L?
If estimate statement used	
Estimate	e1-e?
Standard errors of estimate	se1-se?
Z-value for estimate	z1-z?
P-value from contrast	p1-p?
Odds ratio of contrast	o1-o?
Upper bound of 95% CI	u1-u?
Lower bound of 95% CI	L1-L?
If contrast statement used	
Chi-Square for contrast	chisq1-chisq?
P-value from contrast	p1-p?

For all analyses the overall likelihood ratio chi-square statistic (lrchi1-lrchi?) and associated p-value (plr1-plr?) are placed on the dataset. Also on all datasets is the number of events (ne1-ne?), number of non-events (nc1-nc?) and total observations in analysis (nt1-nt?).

? is the number of analyses (dependent variables)

Example Call

%logistp (data=rand, dlist=chd cvd allcause, ilist=age dbp chol , factor=age, out=mi, order=d, units=10)

Macro with default and optional parameters listed

%logistp (data=_last_, dlist=, ilist=, class=, factor=, units=, where=, out=&factor, order=D);

Description of parameters

data= SAS data set read

class= class variable(s) for LOGIST. List all variables that are to treated as categorical variables.

dlist= dependent variable list. An analysis is run for each variable. Each variable should have 2 values, one value for the event and one value for a non-event (usually 1 for event and 0 for non-event)

ilist= independent variable list. This list is used for each dependent variable given in dlist.

factor= independent variable(s) in model for which statistics are obtained

units= used to obtain odds ratios for more than a unit difference. Before the odds ratio is calculated, the coefficients are multiplied by the value of this parameter. The default is 1. This is given as a list over the factors. If there are 3 factors then there should be 3 units given in the list.

where= Optional parameter which allows the user to add a WHERE clause to the call

Example: where = where ondrug eq 1

out SAS datsset(s) in which statistics are stored. There is one dataset for each factor.

order Set to D to estimate the probability that the dependent variable is equal to 1 rather than the default which models the probability equal to 0.

estimate If given, an estimate statement(s) is generated to perform 1 df contrasts. Multiple contrasts are separated by a slash (/).

Example: estimate = '1 v 3' trt 1 0 -1/'2 v 3' trt 0 1 -1
This would perform two contrasts, the first comparing the first and third levels of trt; the second comparing the second and third levels of trt.

estdata The dataset the estimates are stored. The variable are:

e1-e? contrast (estimate) values
se1-se? standard errors of the estimates
z1-z? z-statistics
p1-p? p-values
o1-o? odds ratio
l1-l? lower 95%CI
u1-u? upper 95%CI

? is the total estimate statements times the number of dependent variables. They are stored in the order generated. If there are 2 dependent variables and 2 contrasts then:

e1 = estimate for analysis 1, contrast 1
e2 = estimate for analysis 1, contrast 2
e3 = estimate for analysis 2, contrast 1
e4 = estimate for analysis 2, contrast 2

contrast= This generates the contrast statement in LOGIST to perform ¿ 1 df contrasts. Contrasts are separated by a slash (/).

Example: Contrast = %str('2 df test' trt 1 0 -1, trt 1 -1 0)

The %str function is required because a comma is used in the parameter.

condata= The dataset the contrasts are stored. The variables are:

chisq1-chisq? Chi-Square statistics for contrasts
p1-p? P-values for contrasts
df1-df? Degrees of freedom of the contrast

? is the total number of contrast statements times the number of dependent variables. They are stored in the same order as the estimate statements (see above).

4.9 MIXEDP

MIXEDP runs PROC MIXED and stores estimates, standard errors, t-statistics, and p-values for factors of interest into SAS datasets. Each factor is placed in a separate dataset. Results from estimate and contrast statements can also be put in SAS datasets.

Statistics Available	**Names on Output Data Set**
For class variables	
F-statistic	f1-f?
P-value from F-statistic	p1-p?
For non-class variables	
Estimate (beta coefficient)	e1-e?
Standard error of estimate	se1-se?
T-statistic (beta/SE)	t1-t?
P-value for estimate	p1-p?
If estimate statement used	
Estimate	e1-e?
Standard errors of estimate	se1-se?
T-value for estimate	t1-t?
P-value from contrast	p1-p?
If contrast statement used	
F-value from contrast	f1-f?
DF from constrast	df1-df?
P-value from contrast	p1-p?

See below for value of ? for each statistic and how they are stored.

Example Call

%mixedp (data=rand, class= ptid active stratum, dlist= sbp dbp,
ilist=onmeds stratum active stratum*active,
blist = sbpbl dbpbl,
random = int/type=un sub=ptid
factor = stratum*active,
out = interaction,
estimate= 'Stratum' stratum 1 -1 / 'Active' active 1 -1,
estdata=con);

Macro with default and optional parameters listed

%mixedp (data=_last_, class=, dlist=, ilist=, blist=, factor=, random=, repeated=, out=&factor,
estimate=, estdata=, contrast=, condata=, lsmeans=, where=, option=noclprint);

Description of parameters

data=	SAS data set read
dlist=	Dependent variable list. A separate MIXED is run for each variable in dlist.
ilist=	Independent variable list. The same list will be used for each dependent variable.
blist=	Optional list of variables where the i'th variable in the list is added as a covariate to the i'th analysis. Used mainly to adjust for the baseline value of the dependent variable.
option=	Any legal option on the PROC MIXED statement. Default is NOCLPRINT.
class =	List of class variables including grouping unit (usually the ID).
random =	The entire RANDOM statement used in MIXED. Example: random= visit/sub=ptid;
repeated =	The entire REPEATED statement used in MIXED. Example: repeated= visit/sub=ptid;
lsmeans=	include factors desired. No output dataset is created.
factor=	Independent variable(s) for which statistics are to be obtained
where=	Optional paramter which allows the user to add a WHERE clause.
out=	SAS dataset(s) in which statistics are stored for the variables listed in factor. There is one dataset per factor with one observation in each dataset.
estimate=	If given, an estimate statement(s) is generated to perform 1 df contrasts. Multiple contrasts are separated by a slash (/). Example: estimate = '1 v 3' trt 1 0 -1/'2 v 3' trt 0 1 -1 This would perform two contrasts, the first comparing the first and third levels of trt; the second comparing the second and third levels of trt.
estdata=	The dataset the estimates are stored. The variable are: e1-e? contrast (estimate) values se1-se? standard errors of the estimates t1-t? t-statistics

p1-p? p-values

? is the total estimate statements times the number of dependent variables. They are stored in the order generated. If there are 2 dependent variables and 2 contrasts then:

e1 = estimate for analysis 1, contrast 1
e2 = estimate for analysis 1, contrast 2
e3 = estimate for analysis 2, contrast 1
e4 = estimate for analysis 2, contrast 2

contrast= This generates the contrast statement in LOGIST to perform ¿ 1 df contrasts. Contrasts are separated by a slash (/).

Example: Contrast = %str('2 df test' trt 1 0 -1, trt 1 -1 0)

The %str function is required because a comma is used in the parameter.

condata= The dataset the contrasts are stored. The variables are:

f1-f? F-statistics for contrasts
p1-p? P-values for contrasts
df1-df? Degrees of freedom of the contrast

? is the total number of contrast statements times the number of dependent variables. They are stored in the same order as the estimate statements (see above).

4.10 PHREGP

PHREGP runs PROC PHREG to perform Cox regression and saves results for factors of interest into SAS datasets. Results for each factor is placed in a separate dataset. Note: PHREGPX is a version of PHREGP that allows the user to specify the minimum number of events to run the phreg. For events in the dependent variable list that are not run the values of all variables are set to missing.

Statistics Available	**Names on Output Data Set**
For class variables	
Wald Chi-Square statistic	chisq1-chisq?
P-value from Chi-square	p1-p?
For non-class variables	
Estimate (beta coefficient)	e1-e?
Standard error of estimate	se1-se?
Z-statistic (beta/SE)	z1-z?
P-value for estimate	p1-p?
Hazard Ratio	r1-r?
HR lower 95% CI	L1-L?
HR upper 95% CI	u1-u?
If estimate statement used	
Estimate	e1-e?
Standard errors of estimate	se1-se?
Z-value for estimate	z1-z?
P-value from contrast	p1-p?
Hazard Ratio	r1-r?
HR lower 95% CI	L1-L?
HR upper 95% CI	u1-u?
If contrast statement used	
Chi-Square for contrast	chisq1-chisq?
P-value from contrast	p1-p?

Both class and non-class datasets contain the number of events (NE1-NE?), the number of non-events (NC1-NC?) and total observations in analysis (NT1-NT?).

? is the number of analyses (dependent variables)

Example Call

%phregp (data=rand, dlist=chd cvd allcause, ilist=age dbp chol , tlist= t t t, strata=clinic, factor=age, out=out1)

Macro with default and optional parameters listed

%phregp (data=_last_, dlist=, ilist=, tlist=, units=, strata=, class=, factor=, out=&factor, where=, skipout=, ties=breslow, estimate=, estdata=, contrast=, condata=);

Description of parameters

data=	SAS data set read
dlist=	dependent variable list. An analysis is run for each variable in dlist. Each variable should be coded 1 for event and 0 if censored.
class=	list of class variables. The GLM parameterization is used.
ilist=	independent variable list. This list will be used for each dependent variable given in dlist.
strata=	strata variable(s).
factor=	independent variables(s) for which statistics are to be obtained.
units=	used to obtain RR (HR) for more than a unit difference. Before the risk ratios are calculated, the coefficients are multiplied by the value of this parameter. The default is 1. This is given as a list over the factors. If there are 3 factors then there should be 3 units given in the list.
ties=	method for handling ties. Default is Breslow.
skipout=	if set to Y then no datasets are created. Macro used only to generate PHREG code.
where=	optional parameter which allows the user to add a WHERE clause or any other option coding statement. Example: where = where ondrug=1
out	SAS daatset(s) in which statistics are stored. There is one dataset for each factor.
estimate	If given, an estimate statement(s) is generated to perform 1 df contrasts. Multiple contrasts are separated by a slash (/). Example: estimate = '1 v 3' trt 1 0 -1/'2 v 3' trt 0 1 -1 This would perform two contrasts, the first comparing the first and third levels of trt; the second comparing the second and third levels of trt.
estdata	The dataset the estimates are stored. The variable are:

? is the total estimate statements times the number of dependent variables. They are stored in the order generated. If there are 2 dependent variables and 2 contrasts then:

e1 = estimate for analysis 1, contrast 1
e2 = estimate for analysis 1, contrast 2
e3 = estimate for analysis 2, contrast 1
e4 = estimate for analysis 2, contrast 2

contrast= This generates the contrast statement in LOGIST to perform ¿ 1 df contrasts. Contrasts are separated by a slash (/).

Example: Contrast = %str('2 df test' trt 1 0 -1, trt 1 -1 0)

The %str function is required because a comma is used in the parameter.

condata= The dataset the contrasts are stored.

? is the total number of contrast statements times the number of dependent variables. They are stored in the same order as the estimate statements (see above).

4.11 REGP

REGP runs PROC REG and stores regression coefficients, standard errors, t-statistics, and p-values of covariates from the regression into one or more SAS datasets.

Statistics Available	**Names on Output Data Set**
Regression coefficients (betas)	b1-b?
Standard errors of betas	seb1-seb?
T-statistics (beta/SE)	t1-t?
P-value for T-statistic	p1-p?
Lower 95% CI for beta	L1-L?
Upper 95% CI for beta	u1-u?

Where ? is the number of dependent variables.

Example Call

```
%let listd = chold hdld ldld trigd;
%let listb = cholbl hdlbl ldlbl trigbl;
%let listi = group2 clinica clinicb clinicc stratum2;
%regp (data=rand, dlist=&listd, ilist=&listi, blist=&listb, factor=group2, cohort=N, out=out1);
```

Macro with default and optional parameters listed

```
%regp (data=_last_, dlist=, ilist=, blist=, factor=, cohort=Y, where=, out=&factor);
```

Description of parameters

data=	SAS data set read
dlist=	dependent variable list. An analysis is run for each variable
ilist=	independent variable list. This list is used for each dependent variable given in dlist.
blist=	optional list of variables where the i'th variable in the list is added as a covariate to the i'th analysis. It is used mainly to adjust for the baseline value of the dependent variable.
factor=	independent variable(s) for which statistics are obtained.
cohort=	if set to Y (default), a single PROC REG is run which excludes cases with any dependent variable having missing having missing data from all regressions. If cohort=N, then PROC REG is run separately for each dependent variable.
out =	dataset(s) to which statistics are stored. There is one dataset for each variable in factor.

where= Optional parameter which allows the user to add a WHERE clause to the call

Example: where = where ondrug eq 1

4.12 LIFETAB

LIFETAB reads the dataset created in the OUTSURV option of PROC LIFETEST and outputs the survival estimates for specified time values for each strats. This macro is useful for displaying life table estimates from selected points on the graph. Unlike the other statistical macros, the user needs to run the LIFETEST procedure before calling the macro.

Statistics Available	**Names on Output Data Set**
Survival estimates	s1-s?
1-survival estimates	f1-f?
Upper bound of 95% CI for S	u1-u?
Lower bound of 95% CI for S	L1-L?
Times	t1-t?
Strata value	g1-g?

Where ? is the number of time points specified times the number of strata levels. If there were 5 time points and 2 levels for strata then ? would be 10 (5x2).

Example Call

%lifetab (data=surv, t=tmort, times= 1 2 3 4 5 6, out=surv, strata=trt);

Macro with default and optional parameters listed

%lifetab (data=, t=, times=, inter=, ninter=, out=&data, print=Y, zero=)

Description of parameters

data= SAS dataset previously output from the OUTSURV option of PROC LIFETEST.

t= name of variable containing event time;

times= list of times to output survival curve values for each strata.

ninter= number of intervals (used with option inter) of time to output.

inter= output survival values at ninter intervals of this length (used with ninter).

Use TIMES or INTER with NINTER to indicate the times you want the survival estimates to be output. The units should be consistent with the units for T.

Example: TIMES = 0.5 1 2 3 4 5 (output at 6 months and then yearly)
Example: NINTER = 5, INTER= 365 (output at 365, 365*2, 365*3, 365*4, 365*5)

zero= if set to Y the the survival value at t=0 is output.

out= SAS dataset to which statistics are strored.

print= Y prints out values of survival curve at all uncensored points.

4.13 GENMODP

GENMODP runs PROC GENMOD and stores statistics from the model into one or more datasets. Default parameters are set for Poisson regression.

Statistics Available	**Names on Output Data Set**
For class variables	
Chi-Square statistic	k1-k?
P-value from Chi-square	p1-p?
For non-class variables	
Estimate (beta coefficient)	e1-e?
Standard error of estimate	se1-se?
Z-statistic (beta/SE)	z1-z?
P-value for estimate	p1-p?
Relative Rates	r1-r?
Lower 95% CI for RR	L1-L?
Upper 95% CI for RR	u1-u?
If estimate statement used	
Estimate	e1-e?
Standard errors of estimate	se1-se?
Z-value for estimate	z1-z?
P-value from contrast	p1-p?
Odds ratio of contrast	o1-o?
Lower 95% CI for RR	L1-L?
Upper 95% CI for RR	u1-u?

See below for value of ? for each of the statistics and how they are stored.

Example Call

%genmodp(data= temp, class= ptid yearcat, dlist= nhosp naids, ilist= yearcat cd4, offset=lpr,
repeated = subject=pin/type=cs corrw,
factor = yearcat, out=out1,
estimate = '2005 v 2006' yearcat 1 -1 0 — '2006 v 2007' yearcat 0 1 -1, estdata=est)

Macro with default and optional parameters listed

%macro genmodp(data=_last_, class=, dlist=, ilist=, blist=, link=log, dist=poisson, offset=, repeated=, factor=, out= &factor, estimate=, estdata=, units=, where=, options=);

Description of parameters

data=	SAS dataset read
dlist=	dependent variable list. A separate GENMOD is run for each variable is dlist.
ilist=	independent variable list. This same list will be used for each dependent variable in the analysis.
blist =	optional list of variables where the i'th variable in the list is added as a covariate to the i'th analysis. It is used mainly to adjust for the baseline value of the dependent variable.
link=	link function (default is log)
offset =	offset variable (used for Poisson regression)
option =	any legal option of the GENMOD statement.
class =	list of class variables including grouping unit
repeated =	the complete REPEATED statement used in GENMOD. If left blank then no repeated statement is generated. Example: repeated = subject=pin/type=cs
factor=	independent variables(s) for which statistics from GENMOD are to be obtained.
out=	dataset(s) in which statistics are stored for the variable listed in factor. There is one dataset per factor with one observation in each dataset.
estimate	If given, an estimate statement(s) is generated to perform 1 df contrasts. Multiple contrasts are separated by a slash (/). Example: estimate = '1 v 3' trt 1 0 -1/'2 v 3' trt 0 1 -1 This would perform two contrasts, the first comparing the first and third levels of trt; the second comparing the second and third levels of trt.
estdata	The dataset the estimates are stored. The variable are: e1-e? contrast (estimate) values se1-se? standard errors of the estimates z1-z? z-statistics p1-p? p-values l1-l? lower 95%CI u1-u? upper 95%CI

? is the total estimate statements times the number of dependent variables. They are stored in the order generated. If there are 2 dependent variables and 2 contrasts then:

e1 = estimate for analysis 1, contrast 1
e2 = estimate for analysis 1, contrast 2
e3 = estimate for analysis 2, contrast 1
e4 = estimate for analysis 2, contrast 2

where= Optional parameter which allows the user to add a WHERE clause to the call

Example: where = where ondrug eq 1

4.14 MEDIANP

MEDIANP runs PROC NPAR1WAY and stores statistics from the analysis into a dataset.

Statistics Available	**Names on Output Data Set**
Chi-Square statistic	chisq1-chisq?
P-value from Chi-square	p1-p?
Degrees of freedom	df1-df?

Example Call

%medianp(data=weight, class = sex, var=chol, test=w, print=y);

Macro with default and optional parameters listed

%macro medianp(data=_last_, class=, var=, out=, test=W, print=noprint, where=);

Description of parameters

data=	SAS dataset read
var=	dependent variable list. A separate NPAR1WAY is run for each variable in var.
class=	class variable to be compared for each variable in var.
test=	type of test to perform: M (Median); S(Savage); V(VW); W(Wilcoxon).
print=	default is NOPRINT of results; set to Y to print results
where=	Optional parameter which allows the user to add a WHERE clause to the call Example: where = where ondrug eq 1

5 HTML Reports

The macros can also be used to create an HTML table. These are usually what the user generates. This HTML file can them be viewed using a web browser, imported into Word, or converted into a PDF document using a converter program. Most of the formatting options that are available in html such as font type, fontsize, font weight, and indenting and underlining text have been incorporated as well as more general html features such as background and text color. In addition, multiple pieces of data can be combined and placed in a single column with various formatting, such as mean ± SD, N(%), or HR (95%CI) giving the report table a journal finished look.

The most simple HTML table can be made by making the following changes to the usual report:

%wreport(htmlfile='report.html'); and

Add the following 2 lines at the end of your report

stop; %makehtml;

The stop statement is reqired; without it the html file will be incorrect. The macro call to makehtml generates the html file.

There are several enhancements that can be made by using options on the %move and %nmove statements. These are given in the following table.

Macro Parameter	**Description**	**Default**	**Other Values**
Indent	Used to indent text (%move only)	0=none	1=some indent; 2=more indent
Fontfamily	any usual font name recognized by html	Times New Roman	Arial; Courier New
Fontsize	size of font including unit of measurement	10pt	12pt
Fontweight	used to bold text	Normal	Bold; bolder
Fontstyle	used to italicize text	Normal	Italic
Lineheight	size of cell in proportion to text size	125%	200%
Textdecoration	used for underlining	None	underline
Background	used for shading or changing background color	White (#ffffff)	Use #d9d9d9 or #b3b3b3 for shading

Note: Default values are controlled through macro makehtml.

%macro makehtml(parameters)

This macro creates the html file. It is usually not needed to change the default values.

Macro Parameter	Description	Default	Other Values
border	Size of width in pixels of the frame around the table	2	Any number
frame	Specifies which sides of the frame surrounding the table will be visible	Box - all 4 sides	Above; below; hsides; vsides; lhs; rhs (see html manual)
rules	Controls the rules (lines) separating cells	Groups	None; all
cellspacing	Amount of white space within cell in pixels	1	Any value
bgcolor	Background color	#ffffff (white)	Any valid color
textcolor	Text color	000000	Any valid color
width	Width of table	100% (uses all of browser width)	Other %s or actual widths (e.g. 700px)

In addition, you can change the default values for fontfamily, fontsize, fontweight, fontstyle, and lineheight by setting those parameters in %makehtml .

Combining data into single cells using %nmove

Use the combine and fchar parameters to combine and format multiple variables into a single cell.

combine= y to combine data in same cell (see examples)

fchar= specifies how to format combined data.

= 1	uses ± to combine two numbers
= 1C	uses a comma between 2 numbers
= 2	puts second number in parenthesis
= 2P	as in 2 but adds percent sign at end of number
= 3	second and third number are placed in parenthesis and a dash placed between second and third number. Used for HR and CI.
= 3B	second and third number are placed in square brackets and a dash is placed between second and third number.
= 3S	a slash betwwen first and second number with third number in ( )
= 4	second and third number are placed in parenthesis and a comma is placed between second and third number.
= 4B	second and third number are placed in square brackets and a comma is placed between second and third number.

Example using fchar option:

= 1	45.2 ± 6.7
= 1C	1.18,1.96
= 2	231 (17.1)

= 2P	231 (17.1%)
= 3	1.02 (0.88 - 1.18)
= 3B	1.02 [0.88 - 1.18]
= 3S	10/100 (10%)
= 4	1.02 (0.88,1.18)
= 4B	1.02 [0.88,1.18]

Notes: To use the combined feature the variables to be combined must be specified in array type notation. For example, the following call puts the three variables r1, L1, and u1 together in one cell, the variables r2, L2, and u2 together in a second cell, etc. The combine option must be set to Y, otherwise the 21 variables are treated as individual variables. Also, separate formats can be specified for the pair or triplet of numbers as can separate scaler options.

%nmove (r1-r7 L1-L7 u1-u7, col=4, line=14L7, combine=y, fmt=5.2 5.2 5.2, fchar=3);

%nmove (m1-m7 se1-se7, col=4, line=14L7, combine=y, fmt=5.1 5.1, fchar=1);

%nmove (sum1-sum7 m1-m7, col=4, line=14L7, combine=y, fmt=5.0 5.1, fchar=2P, scaler=1 100);

Some notes on using wreport for making html files

- The line numbers given in move and nmove statements are used only in a relative sense, i.e, the lowest line number is the first row of the table, the next line number is the second row of the table, etc. There are no skipped rows. The user can use this to their advantage. For example, a footnote text can be moved to a high number so the footnote text is always at the end of the report.
- The column numbers in %colset are used in a relative sense. Thus, a value of 20 will take twice the space as a value of 10.
- The report displays different grid lines depending on whether or not consecutive line numbers are filled. For example, moving 3 character strings to consecutive lines (e.g. line=22L3) will yield a slightly different looking report then if the strings were moved to lines 22,24,and 26 (line=22 24 26). Moving to consecutive lines will give a block looking section which may be nicer to view then having grid lines between each row. However, this distiction is not honored by Word but is honored by most browsers.
- Underlines (u=1) on %move statements are ignored.
- You may use html tags within text strings to do formatting not controlled by the macro formatting parameters. Use the
 tag in the text to start a new line within the cell. Other useful HTML tags are to superscript the text between tags and to subscript the text between words. If you only wish to bold or italicize a portion of the text in a %move statement, and may be used. If you really want to emphasize a word, <u>important word </u>will give you ***important word***.

5.1 HTML Report Example 1

Each table is followed by the code to generate the table.

Number and Percent of Selected CVD Events by Treatment Group and Hazard Ratio From Cox Regression Analysis

	Number of Patients With Event		Cox Regression Analysis Treatment vs Control	
Endpoint	**Treatment**	**Control**	**HR (95% CI)**	**P-value**
Primary CVD	364 (4.5%)	365 (4.4%)	1.02 (0.88,1.18)	0.771
MI (Fatal/NF)	133 (1.6%)	166 (2.0%)	0.82 (0.65,1.03)	0.089
Stroke (Fatal/NF)	133 (1.6%)	118 (1.4%)	115 (0.90,1.48)	0.265
CVD Death	152 (1.9%)	143 (1.7%)	1.09 (0.87,1.37)	0.471
Any CVD Hospitalization[1]	793 (9.7%)	775 (9.3%)	1.05 (0.95,1.16)	0.307
Revascularization	163 (2.0%)	166 (2.0%)	1.01 (0.82,1.26)	0.913
TIA	89 (1.1%)	105 (1.3%)	0.87 (0.66,1.15)	0.330
Angina	202 (2.5%)	190 (2.3%)	1.09 (0.89,1.33)	0.389
Heart Failure	126 (1.5%)	100 (1.2%)	1.30 (1.00,1.69)	0.051
Hypertension	22 (0.3%)	18 (0.2%)	1.26 (0.67,2.34)	0.474
Renal Failure	27 (0.3%)	34 (0.4%)	0.81 (0.49,1.35)	0.426

[1]Includes events below plus primary events

```
 libname convince '/data/convince/';
data temp;
 set convince.convince ;

 if abgroup = 'A' then group = 1;
 if abgroup = 'B' then group = 0;
 if abgroup = 'A' then cgroup = 1;
 if abgroup = 'B' then cgroup = 2;

 %let evlist = primary mi str cvddth allcvd corevas tia angina chf acchyp renalf;
 %let tmlist = tprimary tmi tstr tcvddth tallcvd tcorevas ttia tangina tchf tacchyp trenalf;
 keep patno group cgroup &evlist &tmlist soc cstrata;
%breakdn(data=temp, class= cgroup 2, var=&evlist, out=out1);
%phregp(data=temp, dlist = &evlist, strata=soc, tlist=&tmlist, ilist=group, factor=group, out=out2);

%wreport(htmlfile='example1.htm');
%colset(25 14 14 21 7);

%move('Number and Percent of Selected CVD Events by Treatment Group and <br >' 'Hazard Ratio
from Cox Regression Analyses', col=1-0, line=3, fontweight=bold, fontsize=12pt);
%move('Number of Patients <br>With Event', col=2-3, line=9, fontweight=bold);
%move('Cox Regression Analyses <br>Treatment/Control' col=4-0, line=9, fontweight=bold);
%move('Treatment':'Control':'HR (95%CI)':'P-Value', col=2.0, line=11, fontweight=bold);
```

```
%move('Endpoint', col=1, center=2, line=11, fontstyle=italic, fontweight=bold);
%move('Primary CVD':'Any CVD Hospitalization <sup>1', col=1, center=2, line=14 19, fontweight=bold);
%move('MI (Fatal/NF)':
 'Stroke (Fatal/NF)':
 'CVD Death':
 'Revascularization':
 'TIA':
 'Angina':
 'Heart Failure':
 'Hypertention':
 'Renal Failure',
 col=1, center=2, line=15L3 20L6, indent=1);
%move ('<sup>1</sup>Includes events below plus primary events', col=1-0, line=30, center=2);

set out1;
%nmove(sum1-sum22 m1-m22, col=2 3, line=14L4 19L7, scaler=1 100, combine=y, fmt=5.0 5.1, fchar=2P);

set out2;
%nmove(r1-r11, L1-L11, u1-u11, col=4, combine=y, fmt=5.2 5.2 5.2, fchar=4);
%nmove(p1-p11, col=5, fmt=6.3);

stop;
run;
%makehtml;
run;
```

5.2 HTML Report Example 2

Mean Levels of Measures Contributing to the Rasmussen Score and Other Determinations by Gender for Rasmussen Population

Variables Contributing to Score	Men	Women	Total	P(M/W)
Large Artery Elasticity	18.5 ± 5.8	14.5 ± 5.1	16.9 ± 5.8	<.001
Small Artery Elasticity	7.4 ± 3.2	5.2 ± 2.8	6.5 ± 3.3	<.001
Resting SBP (mmHg)	133.2 ± 16.8	130.0 ± 20.8	131.9 ± 18.6	0.009
Resting DBP (mmHg)	85.3 ± 9.8	80.8 ± 9.7	83.5 ± 10.0	<.001
Exercise SBP (mmHg)	156.3 ± 21.6	153.4 ± 25.1	155.1 ± 23.1	0.062
Exercise DBP (mmHg)	75.1 ± 11.9	70.7 ± 11.1	73.4 ± 11.8	<.001
Change in SBP (mmHg)	23.1 ± 15.4	23.6 ± 15.7	23.3 ± 15.5	0.662
Change in DBP (mmHg)	-10.2 ± 9.0	-10.3 ± 8.7	-10.2 ± 8.9	0.995
LVMI (g/m2)	94.7 ± 25.6	79.0 ± 22.0	88.3 ± 25.4	<.001
Microalbumin (mg/mmoL)	0.8 ± 1.5	1.3 ± 3.2	1.0 ± 2.3	0.005
BNP (pg/dL)	18.4 ± 22.9	30.0 ± 28.7	23.1 ± 26.0	<.001
Other Risk Factors				
Total Cholesterol (mg/dL)	199.2 ± 35.0	210.9 ± 38.5	203.9 ± 36.9	<.001
LDL Cholesterol (mg/dL)	126.6 ± 31.2	127.2 ± 34.4	126.9 ± 32.5	0.778
HDL Cholesterol (mg/dL)	45.0 ± 11.4	59.2 ± 17.6	50.7 ± 15.8	<.001
Triglycerides (mg/dL)	146.0 ± 115.6	125.0 ± 116.2	137.6 ± 116.2	0.007
Glucose (mg/dL)	95.5 ± 16.2	91.3 ± 17.0	93.8 ± 16.6	<.001
CRP (mg/dL)	0.3 ± 0.4	0.4 ± 0.5	0.3 ± 0.4	<.001
Homocysteine (umol/L)	8.9 ± 2.4	7.7 ± 2.2	8.4 ± 2.4	<.001
Age (y)	51.0 ± 11.4	52.6 ± 11.1	51.7 ± 11.3	0.033
10-Year CHD Risk (%))	7.1 ± 5.6	1.9 ± 2.0	5.0 ± 5.2	<.001
BMI (kh/m2))	27.8 ± 4.4	26.9 ± 5.5	27.4 ± 4.9	0.008
Number of Participants	**565**	**379**	**944**	

```
 %let dlist = c1 c2 rest_sbp rest_dbp ex_sbp ex_dbp _sbp _dbp lvmindx malb bnp chol ldl hdl trig
gluc hscrp homocys age chd10 bmi;

%breakdn(data=temp, class=gender 2T, var=&dlist, out=out1);
%breakdn(data=temp, class=gender 2T, var=age, out=out2);
%regp(data=temp, dlist = &dlist, ilist=gender, factor=gender, cohort=N, out=out3);

%wreport(htmlfile = 'rasmussen.html');
%colset(35 25 25 25 10);

%move('Mean Levels of Measures Contributing to the Rasmussen Score and Other<br>'
'Determinations by Gender for Rasmussen Population',
col=1-0, line=3, fontweight=bold);
%move('Men':'Women':'Total':'P (M/W)', line=9, col=2.5, fontweight=bold);
%move('Variables Contributing to Score':'Other Risk Factors', col=1, center=2, line=9 24,
```

```
fontweight=bold, fontstyle=italic);
%move('Number of Patients', col=1, line=37, center=2);
%move('Large Artery Elasticity':
 'Small Artery Elasticity':
 'Resting SBP (mmHg)':
 'Resting DBP (mmHg)':
 'Exercise SBP (mmHg)':
 'Exercise DBP (mmHg)':
 'Change in SBP (mmHg)':
 'Change in DBP (mmHg)':
 'LVMI (g/m2)':
 'Microalbumin (mg/mmol)':
 'BNP (pg/dL)':
 'Total Cholesterol (mg/dL)':
 'LDL Cholesterol (mg/dL)':
 'HDL Cholesterol (mg/dL)':
 'Triglycerides (mg/dL)':
 'Glucose (mg/dL)':
 'CRP (mg/L)':
 'Homocysteine (umol/L)':
 'Age (y)':
 '10-Year CHD Risk (%)':
 'BMI (kg/m2)',
 col=1, center=2, line=12L11 25L10, indent=2);

set out1;
%nmove(m1-m63 s1-s63, col=2.4, combine=y, fchar=1);
set out2;
%nmove(n1-n3, col=2.4, line=37, fmt=4.0);
set out3;
%nmove(p1-p21, col=5, line=12L11 25L10, fmt=6.0, specfmt=pvalue6.3);

stop;
run;
%makehtml;
```

5.3 HTML Report Example 3

Age, BP, and Additional Risk Factors by Treatment for Patients Randomized in CONVINCE

Baseline Variable	Treatment	Control	Total
Age(y)	65.6 ± 7.4	65.6 ± 7.4	65.6 ± 7.4
Systolic BP (mmHg)	150.1 ± 15.8	150.1 ± 16.0	150.1 ± 15.9
Diastolic BP (mmHg)	86.8 ± 9.8	86.8 ± 9.8	86.8 ± 9.8
Additional Risk Factor (%)			
MI	607 (7.5)	652 (7.9)	1259 (7.7)
Stroke	370 (4.5)	393 (4.8)	763 (4.7)
Cigarette Use	1912 (23.5)	1883 (22.8)	3795 (23.2)
Diabetes	1616 (19.9)	1623 (19.7)	3239 (19.8)
LVH	1000 (12.3)	1019 (12.4)	2019 (12.3)
Cholesterol	2540 (31.2)	2575 (31.2)	5115 (31.2)
TIA	184 (2.3)	162 (2.0)	346 (2.1)
Overweight	4150 (51.0)	4096 (49.6)	8246 (50.3)
Vascular Disease	1362 (16.7)	1387 (16.8)	2749 (16.8)
Vascular Bruit	403 (5.0)	409 (5.0)	812 (5.0)
More than 1 condition	4060 (49.9)	4038 (48.9)	8098 (49.4)
Number of Patients	**8179**	**8297**	**16476**

```
 %let clist = agerand sbpbl dbpbl;
%let blist = rsk001 rsk002 rsk003 rsk004 rsk005 rsk006 rsk007 rsk008 rsk009 rsk010 rskgt1;

%breakdn(data=temp, class=group 2T, var=&clist, out=out1);
%breakdn(data=temp, class=group 2T, var=&blist, out=out2);

%wreport(htmlfile = 'basechar.html');
%colset(28 13 13 13);

%move('Age, BP, and Additional Risk Factors by Treatment for Patients<br>'
   'Randomized in CONVINCE', col=1-0, line=3, fontweight=bold);
%move('Treatment':'Control':'Total', col=2.0, line=9, fontweight=bold);
%move('Baseline Variable':'Additional Risk Factor (%)',
   col=1, center=2, line=9 16, fontweight=bold, fontstyle=italic);
%move('Age (y)':'Systolic BP (mmHg)':'Diastolic BP (mmHg)',
   col=1, center=2 line = 12L3);
%move('MI':
   'Stroke':
   'Cigarette Use':
   'Diabetes':
   'LVH':
   'Cholesterol':
   'TIA':
```

```
   'Overweight':
   'Vascular Disease':
   'Vascular Bruit':
   'More than 1 condition':
   col=1, center=2, line=19L11, indent=1);
%move('Number of Patients', col=1, center=2, line=36, fontweight=bold, fontstyle=italic);

set out1;
%nmove(m1-m9 s1-s9, col=2.4, line=12L3, combine=y, fchar=1);
%nmove(n1 n4 n7, col=2.4, line=36, fmt=5.0, fontweight=bold);

set out2;
%nmove(sum1-sum33 m1-m33, line=19L11, col=2.4, combine=y, fchar=2, fmt=5.0 5.1,
   scaler=0.01 1);
stop;
%makehtml;
```

5.4 Adding Special Characters

The following table gives the code to display various mathematical and other special characters. Each starts with a &#, followed by a number, followed by a semicolon. Other codes can be found by searching the web for special characters in html.

Placing Special Characters in Tables		
	Code for Character	Character Produced
Mathematical/scientific		
Plus or minus	±	±
Less than or equal to	≤	≤
Greater than or equal to	≥	≥
Not equal to	≠	≠
Micro	µ	
Greek Letters		
Alpha	α	α
Beta	β	β
Delta	δ	δ
Epsilon	ε	ϵ
Mu	μ	μ
Sigma	σ	σ
Tau	τ	τ
Footnote symbols		
Dagger	†	†
Double dagger	‡	‡
Other characters		
Colon	:	:
Degree	°	°

Examples of Text Moves Using Special Characters and HTML Tags		
Type of text desired	**Example of %move statement**	**Result**
Greek letter beta, plus or minus sign	%move('β ± SE');	β ± SE
Two lines within a cell	%move('Mean ± SD or N(%)');	Mean ± SD or N(%)
Superscript	%move('P-value¹');	P-value[1]
Subscript, bold a portion of text only	%move('Viral Load(log₁₀')	**Viral Load** ($\log_{10}$)
Footnotes	%move('<u>Notes: </u>¹Adjusted for clinical center');	**Notes:** [1]Adjusted for clinical center

6 Macro Collate

Macro COLLATE takes two or more datasets with common array-type variable names and concatenates them across into one dataset, with the array lengths extended to include the variables from all datasets. This macro can be useful when several report procedure macros have been run and you wish to simplify the %nmove statements in the report section. The description of the parameters are given below.

Parameter	**Description**
data	list of datasets to concatenate
var	list of common prefix names of variables you want to concatenate
suf	a list of the number of array elements in each dataset.
out	output dataset.
delete	if set to Y then input datasets are deleted after out is created.

Macro with default and optional parameters listed

%collate (data=, var=, suf=, out=out, delete=);

Example 1:

Dataset named data1 contains variables m1-m10 and n1-n10 and dataset named data2 contains variables m1-m5 and n1-n5.

%collate (data=data1 data2, var=m n, suf=10 5, out=alldata);

Dataset alldata will contain variables m1-m15 and n1-n15 where m1-m10 and n1-n10 will be same as in data1 and m11-m15 and n11-n15 are the values of m1-m5 and n1-n5 from data2.

Example 2: Used with %breakdn.

%breakdn(data=temp, class= agecat 4, var=death, out=out1);
%breakdn(data=temp, class= sex 2, var=death, out=out2);
%breakdn(data=temp, class= race 4, var=death, out=out3);

%collate(data=out1 out2 out3, var=m sum, suf=4 2 4, out=out);

Data OUT will contain variables sum1-sum10 and m1-m10 containing the number and percent of deaths across the categories of age, sex, and race.

Features

If you have several datasets to collate you can use the dash notation for the dataset list, i.e. data=data1-date5 rather than data=data1 data2 data3 data4 data5. The suf= option can be given as, for example, 5*4 rather than 4 4 4 4 4.