

SUDAAN Language Manual Addendum

Release 9.0.3

September 28, 2007

An electronic copy of this addendum to the SUDAAN Language Manual is available at our website at www.rti.org/SUDAAN.

Copyright 2007 by RTI International
P.O. Box 12194
Research Triangle Park, NC 27709

All rights reserved. No part of this publication may be reproduced or transmitted by any means without permission from the publisher.

Table of Contents

	<u>Page</u>
1. Overview.....	3
2. SUDAAN's New Virtual Memory Management	4
3. SUDWORK Parameter	5
4. PRINT Statement: New Options for RTF Files.....	5
5. Revised CLASS Statement	7
6. Revised PERCENTILE and HISTPCT Statements in DESCRIPT	7
7. Revised SORTBY Statement in RECORDS	9
8. CROSSTAB Procedure Enhancements	9
8.1 Overview of CROSSTAB Enhancements	9
8.2 Syntax Changes Specific to CROSSTAB.....	10
8.2.1 SCORES Statement	11
8.2.2 TEST Statement	12
8.2.3 GOFIT Statement.....	15
8.2.4 Revisions to PROC CROSSTAB Statement.....	16
8.2.5 New PRINT and OUTPUT Groups in CROSSTAB: STEST, ATEST, GOF, and TABLECELL.....	17
8.2.6 CROSSTAB Syntax Examples	19
9. References.....	20

1. Overview

This document describes some of the enhancements that will be introduced in *SUDAAN Release 10.0*, scheduled for shipment in the latter part of 2008. In order to provide annual license holders and new SUDAAN users with the most up-to-date SUDAAN software, the enhancements described in this addendum have been introduced in beta-form with the current maintenance release, *SUDAAN Release 9.0.3*. Any questions or comments about these new enhancements are most welcomed and should be directed via email to SUDAAN@rti.org. *SUDAAN Release 9.0.3* also incorporates several bug fixes, as documented on the SUDAAN website (see the Technical Assistance page on www.rti.org/SUDAAN.)

In summary, the new enhancements that are being introduced in beta-form in *Release 9.0.3* include:

- One of the most exciting enhancements introduced in *Release 9.0.3* is SUDAAN's use of a new, internal **virtual memory manager**. This is available in most procedures and will allow users to process very large datasets in SUDAAN. This is discussed in **Section 2** of this Addendum.
- In conjunction with the new virtual memory manager, a new SUDWORK parameter option has been introduced for all PROC statements. This option will allow users to redirect their SUDAAN work space to an alternate location (e.g. another hard drive, an external drive or another location on a network drive). This is discussed in **Section 3** of this Addendum.
- A second equally exciting enhancement introduced in *Release 9.0.3* is that SUDAAN can now provide output in both **ASCII** and **RTF** format. This is discussed in **Section 4** of this Addendum.
- A few new enhancements have been introduced with the CLASS statement (see **Section 5** of this Addendum).
- Enhancements to the PERCENTILES statement and a change in the default number of bins used to compute percentiles have been introduced in the DESCRIP procedure. See **Section 6** of this Addendum.
- The SORTBY statement has been revised in the RECORDS procedure (see **Section 7** of this Addendum)
- Four major analytical enhancements have been incorporated in the CROSSTAB procedure:
 1. Additional hypothesis tests for single 2-way tables and stratified 2-way tables,
 2. A goodness-of-fit test against known proportions,
 3. Additional test statistics for all hypotheses (analogous to regression procedures), and
 4. A new method for computing confidence intervals for extreme proportions

These CROSSTAB enhancements are discussed in **Section 8** of this Addendum.

The following addendum is intended to provide SUDAAN users with a brief overview of these enhancements as well as provide some instruction on how to implement these enhancements. Examples of these enhancements are in the *SUDAAN Example Manual Addendum, Release 9.0.3* which is posted on the Technical Assistance page on the SUDAAN website, located at www.rti.org/SUDAAN.

2. SUDAAN's New Virtual Memory Management

For efficiency purposes, many of the computations in previous releases of SUDAAN were conducted by holding all the necessary data in core (i.e. in the computer's RAM) or when the need arose, allowing the computer operating system to build files on external disks in order to accommodate the memory needs of certain SUDAAN jobs that required a large amount of memory. In recent years, as the data processing requirements of SUDAAN's users grew, it became clear that this was an inefficient approach for SUDAAN. In some instances, for example, SUDAAN users were attempting to fit relatively simple models with large datasets and the amount of computing memory required for the analytic computations exceeded the bounds of the computer operating system, which at times would result in an "out-of-memory" error in SUDAAN.

In response to the growing analytic needs of SUDAAN users, the entire virtual memory manager in SUDAAN has been retooled beginning in *Release 9.0.3*. When the need arises, by default SUDAAN will now use external disk space to hold computations – thereby by-passing the memory manager limitations of the computer operating system running SUDAAN. This means *SUDAAN Release 9.0.3* can process much larger datasets than ever before. Furthermore, extensive testing has revealed that the new virtual memory manager will often decrease the computation time needed for jobs to run.

The new memory manager has certainly expanded the analytic capabilities of SUDAAN, but it should be noted that some limitations with this software still exists. For example, users will likely not be able to fit models with thousands of explanatory variables or build descriptive tables by crossing a large number of variables. However, for most applications, SUDAAN will be able to adequately process large data files and large analytic requests provided SUDAAN has access to enough external disk space.

In *Release 9.0.3*, SUDAAN's new virtual memory manager is available in the following procedures:

- REGRESS
- LOGISTIC (RLOGIST in SAS-Callable versions)
- LOGLINK
- MULTILog
- KAPMEIER
- RECORDS (useful for sorting)
- DESCRIPT (relevant to percentile estimation only)

New Syntax: USEVMEM Parameter on PROC Statement

To provide some control with the new virtual memory manager, SUDAAN 9.0.3 has a new parameter on the PROC statement (USEVMEM) in all of the procedures listed above. The syntax is:

USEVMEM = *number of megabytes*

Number of megabytes must be an integer between 0 and 2048. By default, *Number of megabytes* is set equal to the 80% of the available physical RAM on the computer running the job. Specifically,

When <i>number of megabytes</i> =	Result
0	When <i>number of megabytes</i> is set equal to zero, it will force SUDAAN to never use any external location to store work files. If there is not enough RAM available for the job, SUDAAN will provide an error message and stop.
1	When <i>number of megabytes</i> is set equal to one, it will force SUDAAN to always use an external location to store work files, regardless of how big the job is. This may be advantageous if a user is running other applications on the computer while running SUDAAN.
2-2048	When <i>number of megabytes</i> is set equal to some number between 2 and 2048, this represents the maximum amount of RAM (in megabytes) that SUDAAN will use for a job before resorting to using an external location to store work files.

3. SUDWORK Parameter

To specify the disk location for all temporary files created during a SUDAAN job, use the new SUDWORK parameter. This includes those work files created by SUDAAN's new virtual memory manager. This new parameter is available in all procedures. The syntax is:

SUDWORK = *path*

By default, Windows versions of SUDAAN will create temporary files in the user's TEMP directory. Solaris and LINUX versions of SUDAAN will create temporary files in the directory associated with the SUDWORK environment parameter. Make sure the location you specify (i.e. the *path*) satisfies the following:

- You must have write access to the location specified
- There must be sufficient free space in the location specified to save the work files

Note that SUDAAN automatically removes all work files at the end of the analysis. However, if SUDAAN is interrupted abnormally for any reason, it is unable to delete these temporary files. In this case you should remove the temporary files to free up disk space.

Example

The following call to MULTILog tells SUDAAN to store computations on an external drive located in the directory c:\sudaan\work. The USEVMEM=1 parameter tells SUDAAN to store all computations on the external drive and therefore limits the amount of RAM SUDAAN will use for the job.

```
PROC MULTILog DATA=Main DESIGN=WR USEVMEM=1 SUDWORK="c:\sudaan\work";
```

4. PRINT Statement: New Options for RTF Files

In addition to the new virtual memory manager (see **Section 2**), a second significant and very exciting enhancement being introduced in *Release 9.0.3* is the ability for SUDAAN to create output in RTF format. This can be accomplished by using the option FILETYPE=RTF that is available on the PRINT statements for all SUDAAN procedures. Users can also specify the font size, font name, the RTF file name, and margin sizes.

The following new options are available on the PRINT statement (after the slash) in all procedures:

FILETYPE=TEXT | RTF

The default file type is TEXT.

FILENAME=*filename*

This is not a new option on the PRINT statement but is relevant to RTF-style output. When you specify FILETYPE=RTF, you must also specify FILENAME=*filename*. The *filename* is the name of the external file that will hold the output.

Note: As in previous releases of SUDAAN, the FILENAME option on the PRINT statement is invalid in SAS-callable SUDAAN when FILETYPE=TEXT. With SAS-callable SUDAAN, the FILENAME option should only be present when FILETYPE=RTF.

REPLACE

This is not a new option on the PRINT statement but is relevant to RTF-style output. When you specify REPLACE, if the file specified by *filename* exists then it will be overwritten with new output. Otherwise, if *filename* exists, SUDAAN will stop and provide an error message indicating the *filename* already exists.

FONTNAME="font"

The value *font* will be the name of the font that should be used in the RTF file. For example, *font* may be "Arial".

- "font" must be the name of a font that is already available on the operating system. Font names need to be surrounded by single or double quotes since there can be spaces in the font names.
- The default *font* is "Times New Roman"

FONTSIZE=*fsize*

This option will apply only when FILETYPE=RTF is specified. The value *fsize* must be an integer that indicates the point size of the font to use in the printed output. By default, the FONTSIZE is 12.

TOPINCH=*number*

Any integer or fractional value that indicates the top margin (in inches) for each page in the printed output. The default is 1 inch.

LEFTINCH=*number*

Any integer or fractional value that indicates the left margin (in inches) for each page in the printed output. The default is 1 inch.

RIGHTINCH=*number*

Any integer or fractional value that indicates the right margin (in inches) for each page in the printed output. The default is 1 inch.

BOTTOMINCH=*number*

Any integer or fractional value that indicates the bottom margin (in inches) for each page in the printed output. The default is 1 inch.

Whenever SUDAAN creates an output file with FILETYPE=RTF, a page break will appear at the end of the printout. Consequently, when the table is printed, a blank page will appear at the end of the file. This page break has been included to facilitate inclusion of the RTF file inside an existing report. This extra page break can be manually omitted if desired.

5. Revised CLASS Statement

Beginning in *Release 9.0.3*, SUDAAN has three minor changes in the CLASS statement in all procedures:

- When SUDAAN displays CLASS frequencies, it uses the formatted values of the CLASS variable if these are available. In previous releases of SUDAAN the software would display unformatted frequencies.
- SUDAAN will now process SUBPOPN statements *before* determining the levels of CLASS variables. Thus, only CLASS variable values which occur on records in the subpopulation will be considered valid (this can essentially reduce the number of valid levels of a CLASS variable that is used in descriptive and regression procedures).
- NOPRINT option on the PROC statement automatically implies NOFREQ on any CLASS statement.

6. Revised PERCENTILE and HISTPCT Statements in DESCRIPT

In certain situations, the DESCRIPT procedure in previous releases of SUDAAN would produce point estimates for percentiles that were outside the range of the confidence intervals for these percentiles. These anomalies were a result of the computational algorithm SUDAAN used to compute the percentiles and associated confidence intervals. To address this anomaly, SUDAAN 9.0.3 modifies the current algorithm used to estimate the lower and upper bound of the confidence interval for the quantiles and adds an option to eliminate ties in the data.

The new algorithm introduced in SUDAAN 9.0.3 for computing the confidence interval will become the default method, and the new tie-breaking method will only be invoked by a new option on the PERCENTILE statement. Specifically, the new tie-breaking algorithm is implemented by specifying the NOISE option after the slash on the PERCENTILE statement. Note the current default method for calculating the point estimates will not change in SUDAAN 9.0.3, only the default method used to calculate the bounds of the confidence interval.

The new method for calculating the bounds for the confidence intervals will provide different results than the current method when there are numerous observations with the same values (ties) for the variable on the VAR statement. The changes occur in the value that SUDAAN uses for $\hat{F}(x_j)$, the CDF (cumulative distribution function) for the variable in the VAR statement. When there are ties in the data, SUDAAN has a choice of using the lower end point or the upper end point of the CDF for the tied observations. When calculating the lower confidence bound, the lower end point is used for $\hat{F}(x_j)$, and when calculating the upper confidence bound the upper end point is used.

The second new feature is an option that can be invoked by the user. The solution adopted by the new tie-breaking algorithm for correcting the percentile ‘point estimate’ problem is to simply add random noise to all of the observations that have a legitimate, non-missing value of the variable(s) in the VAR statement. Noise is added from a uniform distribution. All eligible observations for a particular variable on the VAR statement will have a value from a random variable added to it. The random variable comes from a uniform distribution with parameters $-a$ and a , $U[-a, a]$. For each variable on the VAR statement, SUDAAN creates a new internal variable defined as follows:

$$new_var = VAR_variable + U[-a, a]$$

where *VAR_variable* is the analysis variable on the VAR statement. Observations will be drawn from U[-a,a] using a pseudo random number generator. *New_var* is only used for computing percentiles, not for any other calculations in DESCRIPT. By default, SUDAAN sets $a=10^{-6}$. SUDAAN users can stipulate an alternate value using the BOUND parameter on the PERCENTILE statement.

To ensure that results are reproducible, users can also specify a seed for the pseudo random number generator. The default seed is 56237485. Specify a different seed using the SEED parameter on the PERCENTILE statement. The value of SEED must be between 0 and 67108864.

Note: Invoking the NOISE option has the potential to significantly change the point estimates, standard errors and the confidence intervals for the quantiles when compared to the default method. The degree of the change depends on the nature of the variable on the VAR statement. As noted earlier, this will have not affect on any other statistics produced by DESCRIPT.

A third feature that is new to *Release 9.0.3* is a change in the default number of bins used to calculate percentiles. In previous releases of SUDAAN, the default number of bins was 20. Beginning in *Release 9.0.3*, the default number of bins is 100. The number of bins can be controlled by using the HISTPCT / NPCT=*integer* option in DESCRIPT.

Syntax Changes

In *Release 9.0.3*, SUDAAN has one new option and two new parameters on the PERCENTILE statement in the DESCRIPT procedure. These invoke and control the new percentile method.

The new PERCENTILE statement syntax is as follows:

```
PERCENTILE [percentiles] / [MEDIAN] [QUARTILES] [DECILES] [GROUPED]
           [MINCELL=number] [NOISE] [BOUND=number]
           [SEED=integer];
```

When the NOISE option is present on the PERCENTILE statement, a uniform random variate is added to all eligible observations using the method described above. Note that the NOISE option is not available with the GROUPED option on the PERCENTILE statement.

When the NOISE option is present, two new parameters are available:

BOUND = *number*

Permits specification of the bound to be used in defining the uniform distribution used to generate noise values. The default value is 10^{-6} .

SEED = *integer*

Permits specification of the seed to be used by the random number generator. The default value is 56237485, and the value must be between 0 and 67108864.

If the NOISE option is not included on the PERCENTILE statement, the current quantile calculation method described in *Section 8.9.6* of the *SUDAAN 9 Language Manual* will be used.

7. Revised SORTBY Statement in RECORDS

The syntax of the SORTBY statement in the RECORDS procedure has changed to permit association of distinct sort directions with each sort variable. The new syntax is:

SORTBY [DESCENDING] variable < [DESCENDING] variable>;

So beginning with SUDAAN 9.0.3, all of the following are valid SORTBY statements.

Statement	Description
SORTBY X;	Data is sorted by ascending values of the variable X
SORTBY DESCENDING Y;	Data is sorted by descending values of the variable Y
SORTBY A DESCENDING B C;	Data is sorted first by ascending values of A, then by descending values of B within each level of A, and then by ascending values of C within each level of B.

8. CROSSTAB Procedure Enhancements

Four major enhancements have been introduced to the CROSSTAB procedure in SUDAAN 9.0.3, including:

- Additional hypothesis tests for single and stratified 2-way tables (via the TEST statement),
- A goodness-of-fit test for categorical count data (see the new GOFIT statement),
- Additional test statistics for all hypothesis tests on the TEST and GOFIT statements, and
- A new method for computing confidence intervals for extreme proportions (available via options on the PROC and PRINT/OUTPUT statements).

Section 8.1 provides an overview of these enhancements and **Section 8.2** provides some detail on the syntax changes in CROSSTAB associated with these enhancements. As noted in the introduction of this Addendum, additional examples utilizing these enhancements will be available on the Technical Assistance page of the SUDAAN website located at www.rti.org/SUDAAN.

8.1 Overview of CROSSTAB Enhancements

Additional Hypothesis Tests (TEST and GOFIT Statements)

Prior to *Release 9.0.3*, SUDAAN produced three hypothesis tests: two tests of independence for 2-way tables (LLCHISQ and CHISQ) and the Cochran-Mantel-Haenszel (CMH) test for general association (Mantel and Haenszel, 1959) between two *nominal* categorical variables, row variable *Y* and column variable *X*, while controlling for one or more analytic stratification variables.

SUDAAN 9.0.3 extends the capabilities of CROSSTAB to produce three new hypothesis tests on the TEST and GOFIT statements.

TEST Statement:

- *Generalized CMH Test for Trend* (Mantel, 1963)

This trend CMH test (TCMH) assumes that both Y and X lie on an *ordinal* scale, and is sensitive to a linear association between Y and X of common sign in each stratum. If either Y or X does not lie on an ordinal scale, the trend statistic is not particularly meaningful.

- *Generalized CMH ANOVA-Type Test* (Landis et al., 1978; Agresti, 2002)
This ANOVA-type CMH test (ACMH) assumes that the row variable Y lies on a *nominal* scale and the column variable X lies on an *ordinal* scale. It is sensitive to differences in mean scores among the rows in each stratum. If the column variable X does not lie on an ordinal scale, the ANOVA-type statistic is not particularly meaningful. Its worth noting that when the row variable has only 2 levels, the Trend and ANOVA hypotheses are equivalent.

GOFIT Statement:

- *Goodness-of-Fit Test for Categorical Count Data* (Rao and Scott, 1981)
This is a goodness-of-fit (GOF) test of categorical count data against a set of known proportions.

Additional Test Statistics for all Hypothesis Tests

Thomas and Rao (1987) have shown that standard test statistics (namely, the asymptotic Wald chi-square) for categorical count data can have inflated Type I error rates in finite samples as the degrees of freedom for estimating the variance of parameter estimates decreases and the number of cells in a table increases. Therefore, through SUDAAN 9.0.1 and 9.0.2, CROSSTAB would transform the Wald chi-square test statistic to Shah's Wald F -statistic with the appropriate degrees of freedom, and the p -value for the hypothesis test would be computed from the F -statistic. CROSSTAB would print the untransformed Wald chi-square test statistic value (asymptotically correct, but uncorrected for inflated Type I error rates in finite samples), along with the p -value based on the F -statistic.

SUDAAN 9.0.3 extends the capabilities of CROSSTAB to include the full suite of corrected and uncorrected test statistics already available in the regression procedures: the basic Wald chi-square (asymptotically correct), the adjusted Wald F (Fellegi, 1980), Shah's Wald F (this remains the default test), Rao and Scott's (1981) Satterthwaite-adjusted chi-square, and Shah's Satterthwaite-adjusted F . These test statistics are available for all hypothesis tests on the TEST and GOFIT statements.

Confidence Intervals for Extreme Percentages

Extreme percentages can arise in a number of ways. Even when the overall sample size is large, there may be few positive counts because the percentage is very small (rare events). In other cases, the overall sample size is relatively small, as well as the cells of interest. Calculating confidence intervals for very small (or very large) percentages requires special consideration. The standard methods of generating confidence intervals for extreme percentages can generate intervals with poor statistical coverage probabilities. SUDAAN 9.0.3 extends the capabilities of CROSSTAB to provide an alternative method of generating confidence intervals when the row, column, or total percentages are very small or very large (Korn and Graubard, 1998, 1999). The user specifies what constitutes a 'small' or 'large' percentage.

8.2 Syntax Changes Specific to CROSSTAB

Important Note on Compatibility with Previous Releases

Please note that all CROSSTAB syntax from *SUDAAN Release 9.0.1* and *Release 9.0.2* is still available in *Release 9.0.3*. Furthermore, results in 9.0.3 will be equivalent to 9.0.1 and 9.0.2, with the following exception: by default, *Release 9.0.3* computes the Wald F -test statistic and associated p -value for each of the hypotheses. All releases previous to 9.0.3 print the Wald chi-square test statistic, but the p -value associated with the Wald F

statistic. So the default p-values will be the same between *Release 9.0.3* and *Release 9.0.1/9.0.2* (both based on Wald-F), but the default test statistics will differ (Wald F vs. Wald chi-square).

8.2.1 SCORES Statement

A new SCORES statement has been introduced that allows for specifying the desired row and column variable scores to be used in the TCMH and ACMH hypotheses. This syntax for this statement is as follows:

```
SCORES variable1=(vector) variable2=(vector) ....;
```

Requirements

- The length of the *vector* of scores values must be equal to the number of distinct levels of the named variable in the data, *i.e.*, variable1, variable2, etc. noted in the SCORES syntax above.
- SCORES variable(s) must also be listed on the CLASS or SUBGROUP/LEVELS statements.
- SCORES statement cannot be specified more than once.
- SCORES statement must also be accompanied by a TEST statement.
- If a SCORES variable is *not* included as a row or column variable on any TABLES requests, or if there are no hypothesis tests on the TEST statement that use scores (*i.e.*, no ACMH or TCMH), then the scores for that variable will be ignored.

If there are no SCORES variables associated with the row or column variable on a TABLES request, CROSSTAB will create default scores based on the values implicitly defined by specifying the row and/or column variables on the SUBGROUP and LEVELS statements. If the row and/or column variables are instead specified in the CLASS statement, CROSSTAB will create default scores defined by the actual values of the variable.

For example, if AGE is included on the SUBGROUP and LEVELS statements as follows:

```
SUBGROUP AGE ;  
LEVELS    3 ;
```

And furthermore, if AGE is not listed on the SCORES statement, CROSSTAB will use the values (1,2,3) as the scores associated with the AGE variable.

Suppose DOSE is a four-level variable coded 0, 50, 250, and 1000 and is included on the CLASS statement:

```
CLASS DOSE ;
```

If DOSE is not listed on the SCORES statement, CROSSTAB will use the values (0, 50, 250, 1000) as the scores associated with the DOSE variable.

8.2.2 TEST Statement

New hypothesis tests, test statistics, and related options are contained on the TEST statement. The associated statistics are included in the new STEST and ATEST groups on the PRINT and OUTPUT statements, which will be discussed later in this section. This syntax for this statement is as follows:

```
TEST [CHISQ LLCHISQ CMH TCMH ACMH] / ALL DISPLAY  
< WALDCHI WALDF ADJWALDF  
SATADJCHI SATADJF >;
```

The five *hypothesis test parameters* specified before the slash on the TEST statement are as follows:

- **CHISQ** is the stratum-specific test for independence (based on observed minus expected values) in two-way tables
- **LLCHISQ** is the stratum-specific test for independence (based on a log-linear model) in two-way tables
- **CMH** is the stratum-adjusted Cochran-Mantel-Haenszel test of independence in stratified two-way tables
- **TCMH** is the stratum-adjusted Cochran-Mantel-Haenszel test for trend in stratified two-way tables.
- **ACMH** is the stratum-adjusted Cochran-Mantel-Haenszel ANOVA-type test in stratified two-way tables.

There are no default hypothesis tests. In order to obtain *any* tests of hypothesis, the user must include the TEST statement and specify a subset of the hypothesis test parameters. The user can request any hypothesis test for both stratified and non-stratified 2-way tables.

The five *test statistic parameters* specified after the slash on the TEST statement are as follows:

- **WALDCHI** is the Wald chi-square test (asymptotically correct, but uncorrected for inflated Type I error rates in finite samples).
- **ADJWALDF** is the Adjusted Wald F-test, based on transforming the Wald Chi-square (Fellegi, 1980).
- **WALDF** (default) is the Wald F-test, based on transforming the Wald Chi-square (Shah).
- **SATADJCHI** is the Satterthwaite-adjusted Wald chi-square, with Satterthwaite-corrected degrees of freedom (Rao and Scott, 1981)
- **SATADJF** is the Satterthwaite-adjusted F-test, based on transforming the Satterthwaite Wald chi-square, with Satterthwaite-adjusted numerator degrees of freedom (Shah).

Only those test statistics specified will be computed. If no test parameters are included after the slash, the WALDF test will be the *default* used to test all specified hypotheses.

Also note that:

- **ALL** is shorthand for requesting *all* available test statistics for each specified hypothesis. **ALL** cannot be specified in combination with individual test statistic names. The user may either specify 1) the test statistic(s) of interest, 2) the ALL option to get all available tests, or 3) neither of these, which defaults to the Wald F-test.
- **DISPLAY** can be specified along with the TCMH and/or ACMH hypotheses to request a table of scores (obtained from the SCORES, CLASS, or SUBGROUP statements) assigned to each level of the row and column variables on the TABLES statement request(s). The DISPLAY option is not permitted in the absence of the TCMH and/or ACMH options.

The following table summarizes the 5 test statistics that are available for each of the hypothesis tests in CROSSTAB. The first 3 test statistics (WALDCHI, ADJWALDF, and WALDF) are based on the Wald chi-square Q statistic that uses a design-consistent variance estimator, and the last 2 test statistics (Satterthwaite-adjusted tests SATADJCHI and SATADJF) are based on adjusting the Wald chi-square Q^* statistic that uses a simple random sample variance estimator.

Test Statistic	Formula	Parameter Name on TEST and GOFIT Statements
Wald chi-square	$Q \sim \chi^2_{DFH}$	WALDCHI
Adjusted Wald F (Fellegi, 1980)	$\frac{DF_V - DF_H + 1}{(DF_V)(DF_H)} Q \sim F_{DFH, DFV - DFH + 1}$	ADJWALDF
Shah's Wald F (default)	$\frac{Q}{DF_H} \sim F_{DFH, DFV}$	WALDF
Satterthwaite-Adjusted Chi-square (Rao and Scott, 1981)	$\frac{Q^*}{\bar{\lambda}(1+a^2)} \sim \chi^2_{DFH^*}$	SATADJCHI
Satterthwaite-Adjusted F (Shah)	$\frac{Q^*}{\bar{\lambda}(1+a^2)DF_H^*} \sim F_{DFH^*, DFV}$	SATADJF

Where:

DF_H = Degrees of Freedom associated with the independence or goodness-of-fit hypothesis

DF_V = Degrees of Freedom for estimating the variance-covariance matrix of the cell estimates. For Taylor series and delete-1 Jackknife variance estimation methods, DF_V is the number of PSUs minus the number of survey strata. For BRR and replicate weight Jackknife variance methods, DF_V is the number of replicates.

$\bar{\lambda}$ = Average eigenvalue for Rao and Scott's (1979, 1981) *generalized design effect matrix*. For more information, see Skinner, Holt, and Smith (1989).

a = coefficient of variation of λ_i

DF_H^* = $DF_H / (1+a^2)$.

Important Exceptions for Test Statistics

Taylor series designs: SATADJCHI and SATADJF tests can *only* be requested on the TEST statement when the DEFT1 option is specified on the PROC statement (default design effect is DEFT4).

Jackknife and BRR designs: SATADJCHI and SATADJF tests are not available on the TEST statement.

Multiply imputed data: SATADJCHI and SATADJF tests are not available on the TEST statement.

SCORES and TEST Statements: Example

Suppose Y is the outcome variable, coded 1=present and 2=absent, exposure variable X is a categorical variable coded 1, 2, 3, 4 in the data, representing dosages of 0, 50, 250, and 1000 mg/kg/day of exposure to a compound,

and *Z* is an analytic stratification variable with levels 1, 2, 3. The CROSSTAB computation statements would look as follows:

```
SUBGROUP X Y Z;  
LEVELS 4 2 3;  
SCORES X=(0 50 250 1000);  
TABLES Z*Y*X;  
TEST TCMH / display;
```

In the above statements, *Y* is the row variable, *X* is the column variable, and *Z* is the stratification variable (determined by the request on the TABLES statement). Since the variable *X* appears on the SCORES statement, the levels of the variable *X* as 0, 50, 250, and 1000 (corresponding to levels 1, 2, 3, and 4, respectively) would be used in the calculation of the CMH trend test statistic (TCMH). All other variable levels would be those implied by the SUBGROUP and LEVELS statements. Note that SAS formats are *never* used as scores in the calculation of the ACMH and TCMH hypothesis tests. The DISPLAY option would produce a table of scores that are used for *Y* and *X* in the trend hypothesis test.

As a further point in this example, for $2 \times C$ or $I \times 2 \times C$ tables (row variable has 2 levels, column variable is ordinal), the trend and ANOVA-type hypothesis tests are equivalent. Therefore, including the ACMH option on the TEST statement above would produce the same test results as TCMH. Both are 1 degree-of-freedom tests in this situation.

Note on ACMH Test

For the ANOVA-type CMH test (ACMH), CROSSTAB will treat only the *column* variable as ordinal. For example: the following code would treat the column variable PR (pain relief) as ordinal, the row variable AGE as nominal (since the ACMH test is requested), and GENDER is considered the stratification variable.

```
TEST ACMH;  
CLASS gender age pr;  
TABLES gender*age*pr;
```

For the ACMH test, the only variable treated as ordinal is the column variable (it is always the last variable listed in the table request). Since row variables are treated as nominal for the ACMH test, CROSSTAB will disregard row variable scores if they appear on the SCORES statement.

It is possible for the user to request the TCMH, ACMH, and CMH hypothesis tests for the same table. In this case, CROSSTAB will use only the column variable scores in the computing the ACMH test. It will use both row and column scores in computing the TCMH test, and it will use none of the scores in computing the CMH test (both row and column variables treated as nominal).

Additional Considerations and Syntax Conflicts:

- INCLUDE=MISSING option on SUBGROUP and CLASS statements is not allowed when the TEST statement options ACMH and/or TCMH are specified and/or when there is a SCORES statement present. This is intended so that missing values are not allowed to form valid levels when scores are used.

- If a variable appears on both the CLASS and SCORES statements, the list of user-supplied scores applies to the sort order specified on the CLASS statement (SORT and DIR options determine the sort order).

8.2.3 GOFIT Statement

Specify the GOFIT statement with the appropriate variable names and proportions to compute a goodness-of-fit test against known proportions. The associated statistics are included in the new GOF group on the PRINT and OUTPUT statements, which will be discussed later in this section.

GOFIT variable <*variable...*variable> = (*proportions*) / ALL <WALDCHI WALDF ADJWALDF
SATADJCHI SATADJF>;

Where *variable* is the name of a variable (or a cross between variables) for which the user wants to compare the observed distribution against a set of user-specified *proportions*.

Exactly *k* numeric values should appear in the *proportions* list, where *k* is the product of the number of levels of the GOFIT variables. The order of the *proportions* corresponds to the order of the levels of the *variables*. When a product of variables is specified, the order of the levels is such that the last listed variable varies most rapidly. List these values in the order in which the corresponding variable (or product of variables) levels appear in PRINT tables. It is advised to first print the specified table using keywords from the TABLECELL group (*e.g.*, NSUM) to ensure that you have ordered the known proportions to correctly match the order of the levels according to the way CROSSTAB interprets the SUBGROUP and CLASS statement options. Finally, CROSSTAB checks to see if the proportions sum to one. If not, the proportions are normalized by dividing by their sum.

The user can optionally specify the names of the test statistics of interest (or the ALL option) after the slash. If nothing is specified after the slash, WALDF is the default test. The 5 test statistic parameters appearing *after* the slash are the same as those for the TEST statement (see **Section 8.2.2**). For both the TEST and GOFIT statements, Satterthwaite tests are not available for multiply imputed data.

Requirements

- GOFIT variables must be specified on the CLASS or SUBGROUP/LEVELS statements. However, they do *not* need to be specified on the TABLES statement (in fact, GOFIT statement hypotheses are completely independent of the TABLES statement).
- Multiple GOFIT statements are allowed. One goodness-of-fit hypothesis may be specified on each statement.

Additional Considerations and Syntax Conflicts

- INCLUDE=MISSING option on SUBGROUP and CLASS statements is not allowed in the presence of a GOFIT statement. This is intended so that missing values are not allowed to form valid levels for GOFIT variables.
- If a variable appears on both the CLASS and GOFIT statements, the values supplied on the GOFIT statement apply to the sort order specified on the CLASS statement (SORT and DIR options determine the sort order).

8.2.4 Revisions to PROC CROSSTAB Statement

In SUDAAN 9.0.3, CROSSTAB implements the extreme percentage confidence interval method of Korn and Graubard (1998, 1999) when small or large percentages (as specified by the user) are detected in the cross-classification tables. The estimates that are potentially affected by the alternative confidence interval method are the row, column, and total percentages in each table.

The default confidence interval based on a logit transformation is used in all situations unless otherwise specified by the user. When the user-specified criteria are met, the small percentage confidence interval (SPCI) of Korn and Graubard is then computed in place of the standard logit confidence interval. In addition, a table column or output variable indicating that the confidence interval is calculated using the small percentage method will be included in any PRINT tables and OUTPUT datasets.

To implement the SPCI, two new options are available on the PROC statement in CROSSTAB:

SMCONF = *value*

Use SMCONF to specify the size of estimated percentages that should receive the SPCI formula. The small percentage confidence interval will be calculated for all percentages that are \leq *value* or $\geq 100 - \textit{value}$. *value* must be some number between 0 to 50. The default is SMCONF=0.00.

SMCOUNT = *integer*

Use SMCOUNT to specify the unweighted cell count that determines when to apply the SPCI formula. The variable *integer* must be ≥ 0 and must be an integer. The small percentage confidence interval will be calculated for all cells that have an unweighted count \leq *integer*. Note that this references the actual cell counts, not the weighted cell counts. The default is SMCOUNT=0.

If at least one of these terms is specified, the PRINT and OUTPUT confidence limit keywords LOWROW, UPROW, LOWCOL, UPCOL, LOWTOT and UPTOT will contain the row, column, and total small percentage confidence intervals, respectively, when the user-specified criteria are met. In those cases, the SPCI replaces the default logit confidence interval in the PRINT tables and OUTPUT data sets.

For percentages in which the user-specified criteria are not met, the default logit confidence intervals are still computed. Thus, when the SMCONF or SMCOUNT options are used, the results will likely be a mix of logit and small percentage confidence intervals. Three new PRINT and OUTPUT statement keywords in the TABLECELL group (namely, ROWSPCI, COLSPCI, and TOTSPCI) provide indicators of the type of confidence interval computed for each percentage. CROSSTAB supplies a warning if no values meet the user-specified criterion for the SMCOUNT or SMCONF options. All confidence intervals for row, column, and total percentages are then calculated using the default logit confidence interval.

8.2.5 New PRINT and OUTPUT Groups in CROSSTAB: STEST, ATEST, GOF, and TABLECELL

- When the TEST statement specifies CHISQ or LLCHISQ hypotheses (stratum-specific), the STEST group becomes available on PRINT and OUTPUT statements.
- When the TEST statement specifies CMH, TCMH, or ACMH hypotheses (stratum-adjusted), the ATEST group becomes available.
- Both STEST and ATEST groups are available when stratum-specific and stratum-adjusted hypotheses are specified on the TEST statement.
- When the GOFIT statement is specified, the GOF group becomes available.
- When the SMCONF or SMCOUNT options are specified on the PROC statement, the existing TABLECELL group contains 3 new default keywords: ROWSPCI, COLSPCI, and TOTSPCI. These keywords also generate indicator variable(s) in the OUTPUT data sets and extra column(s) in the appropriate PRINT tables. For OUTPUT data sets, the value of each of these indicator variables is 1 if the default logit confidence interval was used, 2 if the small percentage confidence interval was used. For tables generated by the PRINT statement, the value of each indicator variable is blank if the default confidence interval was used, or the symbol ^ if the small percentage confidence interval was used.

The table below contains the keywords in the three new PRINT and OUTPUT groups, as well as the 3 new keywords in the TABLECELL group:

Keyword	Description	Default Indicator (with TEST or GOFIT Statement)	Default Format
STEEST Group: CHISQ and LLCHISQ hypotheses			
STEESTVAL	Stratum-Specific Test Statistic Value	Default	F8.4
SDF	Hypothesis Degrees of Freedom	Default	F8.0
SPVAL	P-Value for CHISQ and LLCHISQ	Default	F8.4
SADJDF	Satterthwaite-Adjusted Degrees of Freedom	Requires SATADJCHI and/or SATADJF test options)	F8.2
SDDF	Denominator Degrees of Freedom (for multiply imputed data only)	Requires MI data	F8.2

Keyword	Description	Default Indicator (with TEST or GOFIT Statement)	Default Format
ATEST Group: CMH, TCMH, ACMH hypotheses			
AESTVAL	Stratum-Adjusted Test Statistic Value	Default	F8.4
ADF	Hypothesis Degrees of Freedom	Default	F8.0
APVAL	P-Value for CMH, TCMH, ACMH	Default	F8.4
AADJDF	Satterthwaite-Adjusted Degrees of Freedom	Requires SATADJCHI and/or SATADJF test options)	F8.2
ADDF	Denominator Degrees of Freedom (for multiply imputed data only)	Requires MI data	F8.2
GOF Group: Goodness of Fit Hypotheses			
GOFVAL	GOF test statistic value	Default	F8.4
GOFDF	Hypothesis Degrees of Freedom	Default	F8.0
GOFPVAL	P-Value for GOF	Default	F8.4
GOFADJDF	Satterthwaite-Adjusted Degrees of Freedom	Requires SATADJCHI and/or SATADJF test options)	F8.2
GOFDDF	Denominator Degrees of Freedom (for multiply imputed data only)	Requires MI data	F8.2
Additions to TABLECELL Group			
ROWSPCI	SPCI Row Percentage Indicator PRINT tables: contain the symbol ^ if the small percentage confidence interval is calculated for the row percentage within a cell; otherwise it prints a space. OUTPUT datasets: ROWSPCI=2 if the small percentage confidence interval is used for the row percentage; ROWSPCI=1 if the logit confidence interval is used.	Requires SMCONF or SMCOUNT options on PROC statement.	F1.0
COLSPCI	SPCI Column Percentage Indicator Same as above, except for column percentages (COLSPCI).	Requires SMCONF or SMCOUNT options on PROC statement.	F1.0
TOTSPCI	SPCI Total Percentage Indicator Same as above, except for total percentage (TOTSPCI)	Requires SMCONF or SMCOUNT options on PROC statement.	F1.0

Reference: For a complete set of output keywords and groups in CROSSTAB, see **Section 6.7.2** in the *SUDAAN 9 Language Manual*.

8.2.6 CROSSTAB Syntax Examples

The following are some common uses of the TEST, GOFIT, and PRINT statements in CROSSTAB.

In the following example, SUDAAN will print the CHISQ hypothesis test results for a 2-way table using the default Wald F-test (Wald F is default when no test statistics are specified on the TEST statement). All test results are in the STEST group.

```
TABLES A*B;  
TEST chisq;  
PRINT / STEST=default;
```

In the following example, SUDAAN will print the CHISQ hypothesis test results for a 2-way table using the Wald chi-square test, the Adjusted Wald F-test, and the Satterthwaite-adjusted chi-square test.

```
TABLES A*B;  
TEST chisq / waldchi adjwaldf satadjchi;  
PRINT / STEST=default;
```

In the following example, SUDAAN will print the CHISQ and CMH hypothesis test results for a 3-way (stratified) table using the Wald chi-square test and the Adjusted Wald-F test. The CHISQ hypothesis test results are stratum-specific ($B*C$, within each level of A), while the CMH hypothesis test results are stratum-adjusted ($B*C$, adjusted for A). CHISQ results are in the STEST group, CMH results are in the ATEST group.

```
TABLES A*B*C;  
TEST chisq cmh / waldchi adjwaldf;  
PRINT / STEST=default ATEST=default;
```

In the following example, SUDAAN will print the CMH hypothesis test results for a 3-way stratified table ($B*A$, adjusted for C) using the Wald chi-square test, and the GOFIT hypothesis that A has proportions of .2, .3, .3, and .2, respectively, using the default Wald F-test.

```
SUBGROUP A B C;  
LEVELS 4 2 3;  
GOFIT A = (.2 .3 .3 .2);  
TABLES C*B*A;  
TEST cmh / waldchi;  
PRINT / ATEST=default GOF=default;
```

In the following example, SUDAAN will print column percentages (colper) and related statistics from the unstratified 2*4 table (B*A), as well as the CHISQ (general association) and TCMH (trend) hypothesis tests for the table using the default Wald *F*-test. For a single 2-way table, the CMH (not requested here) and CHISQ hypotheses are equivalent.

```
SUBGROUP B A;
LEVELS 2 4;
TABLES B*A;
TEST chisq tcmh;
PRINT nsum wsum colper secol / STEST=default ATEST=default;
```

Below are some examples of the SPCI options with the PRINT statement in CROSSTAB.

The following code would produce small percentage confidence intervals for all cells that have a count less than or equal to 3 (SMCOUNT=3). The ROWSPCI keyword on the PRINT statement will produce a column of ^ or blanks that indicate if the small percentage confidence interval was used. In this example, only row percentages (ROWPER) and related statistics are requested.

```
PROC CROSSTAB DATA=WICDAT FILETYPE=SAS DESIGN=WR SMCOUNT=3;
NEST STRATUM SITE;
WEIGHT ANALWGT1;
CLASS EDUC RACEMOM MOMSMK BFEED;
TABLES EDUC*BFEED*RACEMOM*MOMSMK;
PRINT NSUM WSUM ROWPER SEROW UPROW LOWROW ROWSPCI;
```

This code specifies the SMCONF option. All row and column percentages (ROWPER and COLPER) $\leq 5\%$ will use the SPCI to compute their confidence limits.

```
PROC CROSSTAB DATA=WICDAT FILETYPE=SAS DESIGN=WR SMCONF=5.0;
NEST STRATUM SITE;
WEIGHT ANALWGT1;
CLASS EDUC RACEMOM MOMSMK BFEED;
TABLES EDUC*BFEED*RACEMOM*MOMSMK;
PRINT NSUM WSUM
      ROWPER SEROW UPROW LOWROW ROWSPCI
      COLPER SECOL UPCOL LOWCOL COLSPCI;
```

9. References

This section includes references not included in the SUDAAN 9 *Language Manual*:

Korn, E. L. and B. I. Graubard (1998). Confidence intervals for proportions with small expected number of positive counts estimated from survey data. *Survey Methodology* 24, 193-201.

Korn, E. L. and B. I. Graubard (1999). Analysis of Health Surveys. NY: Wiley.

Landis, J. Richard, Heyman, Eugene R., Koch, Gary G. (1978). "Average Partial Association in Three-way Contingency Tables: a Review and Discussion of Alternative Tests". *International Statistical Review* **46**, 237-254.

Mantel, N. and Haenszel, W. (1959). Statistical aspects of the analysis of data from retrospective studies of disease. *Journal of the National Cancer Institute* **22**, 719-748.

Mantel, N. (1963). Chi-Square Tests with One Degree of Freedom; Extensions of the Mantel-Haenszel Procedure. *Journal of the American Statistical Association* **58**, 690-700.

Rao, J. N. K. and A. J. Scott (1981). "The analysis of categorical data from complex surveys: Chi-squared tests for goodness of fit and independence in two-way tables." *Journal of the American Statistical Association*, 76, (374), 221-230.

Research Triangle Institute (2004). *SUDAAN Example Manual, Release 9.0*. Research Triangle Park, NC: Research Triangle Institute.

Research Triangle Institute (2004). *SUDAAN Language Manual, Release 9.0*. Research Triangle Park, NC: Research Triangle Institute.