Paper 93-28

A Macro Using SAS® ODS to Summarize Client Information from Multiple Procedures

Stuart Long, Westat, Durham, NC
Rebecca Darden, Westat, Durham, NC

Abstract

If the client requests the programmer to run multiple procedures, repetitively, massive amounts of output may be generated. However, the client may only be interested in reviewing a fraction of each set of statistics which are output by the procedures executed. This macro demonstrates how to use ODS OUTPUT to summarize large quantities of output from multiple procedures and combine them into one summary dataset. In our paper, we combine output from PROC LOGISTIC and PROC FREQ. Requested exposures are modeled sequentially to generate crude and adjusted statistics. Desired statistics from each procedure are kept as one or more observations in a dataset per exposure. These datasets are then merged by exposure to generate a dataset which contains both crude and adjusted statistics for the exposure. These merged summary datasets are finally concatenated sequentially, as the macro executes, to produce one summary dataset for all exposures for which the macro was executed. This summary dataset may then be used by the programmer to generate output for the client to review, by using PROC REPORT or any other SAS reporting utility. Although the macro code is advanced, using the macro is simple for those with additional statistical reporting needs.

Introduction

This paper will present an example of how to combine multiple datasets created with SAS ODS Output from multiple procedures. An example is provided for calling the macro with the types of exposure variables that the macro can model in the procedures. This is followed by a detailed analysis of how the statistics are generated for the modeled exposure variables and how, subsequently, these statistics are combined into one summary dataset. It is important to note that the focus of this paper is the summary dataset (**__sumary**), that is returned from the macro. Reporting of the statistics in the summary dataset can be done with whatever method the programmer desires.

Example

In an example presented in Appendix 2, we show how the programmer would include the macro code in a program for use. After including the code for the macro, the programmer may call the macro (**%summary**). There are 5 parameters in the macro call: the working dataset (**sasdset**), the outcome variable (**WHEEZE**), the exposure variable (**farmsize** or **eyecolor**), the value of the referent level for the exposure variable (**0** or **2**), and the list of base model covariates (**&log_vars**). If the exposure variable is continuous, insert a "#" before it in the macro call to avoid its inclusion in the CLASS statement. If the exposure variable is nominal, insert a "@" before it in the macro call to prevent crude and adjusted trend statistics from being generated. If any of the base model exposures are to be included in the CLASS statement, insert a "@" before the variable name, as per the example in Appendix 2.

In our example, we present 5 base model covariates. To improve readability of the macro call, we suggest long lists of covariates are passed as the value of a macro variable (**log_vars**). The macro call returns a dataset of crude and adjusted statistics for all of the macro calls performed (**__sumary**). The dataset will contain 14 observations, one observation for the trend test of the ordinal exposure variable **farmsize**, five observations for each level of the ordinal variable **farmsize**, two observations for the dichotomous exposure of **MilkCows**(only one will be printed in the summary table), one observation for the continuous exposure variable **yrsonfarm**, and 5 observations for each of the levels in the nominal exposure variable **eyecolor**.

Appendix 3 contains a copy of the PROC CONTENTS for the summary dataset(**__sumary**). All variables used with this program must be labeled for proper execution. All non-continuous variables used with this program must be formatted for proper execution. Due to the limitations of PROC LOGISTIC in SAS, formatted data is limited to 16 characters.

Appendix 4 contains an example of a summary table printed from the dataset, **__sumary.**

Coding Methods

This macro code offers the programmer a simple method for summarizing the crude and adjusted statistics for the client to review and report. The programmer is required to supply a minimum number of parameters needed for the modeling, and then is returned a dataset which contains all the desired statistics which can then be reported in whatever method the client wishes. The parameters supplied by the programmer include the original working dataset, the dichotomous outcome variable, the exposure variable, the referent level of the exposure variable, and the list of additional base model covariates included in the model. All variables must be numeric variables with labels and formats. In the following description, macro variables and datasets within parentheses are for reference to the macro code in Appendix 1.

The macro begins by separating the list of additional base model exposures into separate macro variables (**__bv&__k**). Base model ordinal variables are then identified for inclusion in the CLASS statement of the PROC LOGISTIC(by a preceding @). The modeled exposure variable (**__expo**) is identified as ordinal(by a preceding #) or nominal(by a preceding @) for appropriate analysis by the macro. An analysis dataset is created by keeping only the observations where the outcome, as well as all the exposure variables, are recorded (**__logis**). The purpose for this is to ensure that the crude statistics are generated for the same number of observations as the adjusted statistics.

Macro variables are then generated which contain the number of levels in the outcome(**__oc_kt**) and modeled exposure(**__exp_kt**) variables, which were passed in the macro call(**__outcom,__expo**). The values of each level in the outcome(**__oc&i**) and modeled exposure(**__exp&il**) variables as well as the formats (**__fmt&i**) are also placed in macro variables. It is important to set a macro variable to the value of the format of the referent level of the exposure variable (**__reffmt**) since the CLASS statement in the PROC LOGISTIC will require the formatted value to identify the referent level in a multi-level exposure variable.

Next, the crude association between the outcome and exposure variable is determined (**__crudep**) for the continuous, dichotomous, and trend of the ordinal exposures The crude association is then generated for each class of the ordinal as well as nominal exposures. This is followed by the assignment of column frequencies and percents between the exposure and

outcome variable where the modeled exposure variable is dichotomous, ordinal, or nominal. The frequencies and percents are stored in one dataset for each level of the outcome variable and are merged together with the crude associations generated in the PROC LOGISTIC ODS Output datasets(__**fstats**).

If the modeled exposure variable contains more than 2 levels and is not nominal, then the adjusted statistics are generated for a trend test of the modeled exposure variable. These adjusted statistics are output using ODS Output datasets(__**p,** __**o**) and subsequently merged to create a one-observation dataset (__**trend**). An additional PROC LOGISTIC is executed to generate the adjusted statistics for the dichotomous exposure variable, or CLASS variables in the event that the exposure variable contains more than 2 levels and is either ordinal or nominal. The adjusted statistics are retained in datasets (__**param,** __**odds**) as one observation for each comparison level of the exposure variable (r-1 observations).

These datasets are then merged together with the dataset that contains the crude statistics (__**fstats**) to form a dataset that contains an observation for each comparison level of the exposure variable as well as an observation for the referent level (__**cstats**). If the exposure variable contains r>2 levels, and is not nominal, then the dataset containing the adjusted statistics from the trend test of the exposure variable (__**trend**) are added as an additional observation into the dataset containing the crude and adjusted CLASS statistics (__**cstats**). If the variable was identified by the programmer as continuous, then the dataset containing the continuous statistics(__**trend**) will be set to __**cstats**.

Finally, the macro concatenates the statistics generated and stored in the final dataset (__**cstats**) with a summary dataset that contains statistics from each previous call of the macro for other exposure variables (__**sumary**).

Summary

The accompanying macro code, using ODS, alleviates the tedious programming and manual extraction associated with the creation of individual datasets for PROC FREQ, the running of necessary crude and adjusted analyses using PROC FREQ and PROC LOGISTIC, and the extraction of pertinent cell counts and statistics when we have a large case-control study with numerous exposure variables. It provides the client with a dataset of pertinent summary information without overwhelming him or her with pages and pages of output. The programmer can then determine what method of outputting the data will be used for reporting the summary statistics to the researcher.

DISCLAIMER: The contents of this paper are the work of the author(s) and do not necessarily represent the opinions, recommendations, or practices of Westat.

Appendix 1. Macro Code

```
%MACRO summary(__dset, __outcom, __expo,
               __ref, __basevs);
/* create macro variables for each of the     */
/* base variables specified                   */
/* as well as a counter for how many base     */
/* variables are passed into the macro.       */
  %LET __varkt=0;
  %LET __k=1;
  %LET __w=%SCAN(&__basevs,&__k);
  %DO %WHILE(&__w NE);
      %LET __bv&__k=&__w;
      %LET __varkt=&__k;
      %LET __k=%EVAL(&__k+1);
      %LET __w=%SCAN(&__basevs,&__k);
  %END;
  /* determine which base model variables will*/
  /* be in the class statement.               */
  %LET __cvkt=1;
  %DO i = 1 %TO &__varkt;
      %LET __amp=%SUBSTR(&&__bv&i,1,1);
      %IF &__amp=@ %THEN %DO;
          %LET __bv&i=%SUBSTR(&&__bv&i,2);
          %LET __cv&__cvkt=&&__bv&i;
          %LET __cvkt=%EVAL(&__cvkt+1);
  %END;%END;
  %LET __cvkt=%EVAL(&__cvkt-1);
  /* check to see if the user requested the   */
  /* exposure variable to be continuous.  If  */
  /* it is continuous, then set the __ctn     */
  /* macro flag to '1' and reset the exposure */
  /* variable to be equal to __expo minus the */
  /* included pound sign (#).                 */
  %LET __pound=%SUBSTR(&__expo,1,1);
  %IF &__pound=# %THEN %DO;
      %LET __ctn=1;
      %LET __expo=%SUBSTR(&__expo,2);
  %END;%ELSE %LET __ctn=0;
  /* check to see if the user requested the   */
  /* exposure variable to be categorical. If  */
  /* it is categorical, then set the __cat    */
  /* macro flag to '1' and reset the exposure */
  /* variable to be equal to __expo minus the */
  /* included ampersand sign (@).             */
  %IF &__pound=@ %THEN %DO;
      %LET __cat=1;
      %LET __expo=%SUBSTR(&__expo,2);
  %END;%ELSE %LET __cat=0;
/* create an analysis dataset that contains   */
/* only the observations that will be in the  */
/* model.  Also use this data step to assign  */
/* the label and formats for the exposure     */
/* variable to macro variables.               */
DATA __logis;
    SET &__dset
     (KEEP = &__outcom &__expo
     %DO i = 1 %TO &__varkt; &&__bv&i %END;);
    IF _N_=1 THEN DO;
       CALL SYMPUT('__explab',VLABEL(&__expo));
       CALL SYMPUT('__oclab',VLABEL(&__outcom));
       CALL SYMPUT
("__vfmt",TRIM(PUT(VFORMATN(&__expo),$CHAR8.)));
       CALL SYMPUT
("__ofmt",TRIM(PUT(VFORMATN(&__outcom),$CHAR8.))
); END;
    IF &__outcom > .z;
    IF &__expo    > .z;
    /* repeat the exclusions for each exposure*/
    /* variable in the base model             */
    %DO i = 1 %TO &__varkt;
        IF &&__bv&i   > .z;
    %END; RUN;
/* create a PROC FREQ output dataset that     */
/* will be used to attain the total number of */
/* levels in the outcome variable             */
 PROC FREQ DATA=__logis;
   TABLES &__outcom / NOPRINT OUT=__oc; RUN;
/* Create two macro variables that contain    */
/* the levels of the outcome variable. Set    */
/* macro variables for the n's of the two     */
/* levels of the outcome variable put the     */
/* total number of outcome levels into a macro*/
/* variable                                   */
DATA _NULL_;
  SET __oc NOBS=obs_tot;
  IF _N_=1 THEN CALL SYMPUT
     ('__oc_kt',LEFT(PUT(obs_tot ,4.)));
  %DO i= 1 %TO 2;
      IF &i=_N_ THEN DO;
         %GLOBAL __oc&i.;
```

```
CALL SYMPUT("__oc&i.",LEFT(PUT(&__outcom,4.)));
CALL SYMPUT("__oc_kt&i.",LEFT(PUT(COUNT,5.)));
      END;
  %END;RUN;
/* create a PROC FREQ output dataset that    */
/* will be used to attain the total number of*/
/* levels in the exposure variable           */
  PROC FREQ DATA=__logis;
    TABLES &__expo / NOPRINT OUT=__levels;RUN;
/* put the total number of levels of exposure*/
/* variable  into a macro variable           */
  DATA _NULL_;
    IF 0 THEN SET __levels NOBS=obs_tot;
    IF _N_=1 THEN DO;
       CALL SYMPUT
       ('__exp_kt' , LEFT(PUT(obs_tot ,4.)));
       STOP;END;RUN;
/* Create two sets of macro variables. The   */
/* first set contains the values of the levels*/
/* in the exposure variable. The second set  */
/* contains the values of the formats for each*/
/* level in the exposure variable            */
%IF &__ctn NE 1 %THEN %DO;
  %DO i= 1 %TO &__exp_kt;
      %GLOBAL __exp&i. __fmt&i.;
      DATA _NULL_;
        LENGTH zzz $ 16;
        SET __levels NOBS=obs_tot;
        IF &i=_N_ THEN DO;
           CALL SYMPUT
           ("__exp&i.",LEFT(PUT(&__expo,4.)));
     zzz=SUBSTR(LEFT(PUT(&__expo,&__vfmt..)),1);
           CALL SYMPUT
      ("__fmt&i" ,
      SUBSTR(LEFT(PUT(zzz,$CHAR16.)),1));;
           IF &__expo=&__ref THEN
              CALL SYMPUT
      ("__reffmt" ,
      SUBSTR(LEFT(PUT(zzz,$CHAR16.)),1));;
        END; RUN;
%END;%END;
/* determine the crude p-value for the       */
/* associations of ordinal, dichotomous, and */
/* continuous exposures                      */
  QUIT;
  %IF &__cat NE 1 %THEN %DO;
  ODS OUTPUT  ParameterEstimates=__pe;
  PROC LOGISTIC DATA=__logis;
     MODEL &__outcom = &__expo;RUN;
  ODS OUTPUT CLOSE;
  DATA _NULL_;
    SET  __pe;
    IF UPCASE(VARIABLE)=UPCASE("&__expo")
    THEN CALL SYMPUT

('__crudep',LEFT(PUT(PROBCHISQ,5.4)));RUN;
  %END;
/* determine the crude association for each  */
/* comparison level of the exposure variable */
/* with the referent level                   */
%IF __exp_kt>2 AND &__ctn NE 1 %THEN %DO;
  ODS OUTPUT parameterestimates=__par; RUN;
  PROC LOGISTIC DATA=__logis DESCENDING;
    CLASS &__expo (PARAM=REF REF="&__reffmt"
          ORDER=INTERNAL);
    MODEL &__outcom = &__expo;RUN;
  ODS OUTPUT CLOSE;
  DATA __par;
    SET __par;
    IF UPCASE(VARIABLE)=UPCASE("&__expo");
    RENAME CLASSVAL0 = exposure_label;
    RENAME ProbChiSq = crude_p_value;
    %DO i = 1 %TO &__exp_kt;
        IF CLASSVAL0="&&__fmt&i" THEN
           exposure_value=&&__exp&i.;
    %END;
    LABEL CLASSVAL0       =
"Exposure Label(Class/Dichotomous Level Format)"
          exposure_value =
"Dichotomous/Class: Value of Exposure Level";
    DROP VARIABLE DF ESTIMATE STDERR WALDCHISQ;
  RUN;
%END;
/* If the exposure is not continuous, then   */
/* run PROC FREQ to generate counts and      */
/* percents for each level in the exposure   */
/* variable crossed with the outcome variable*/
%IF &__ctn NE 1 %THEN %DO;
  ODS OUTPUT CrossTabFreqs = __ctab;
  PROC FREQ DATA=__logis;
    TABLES &__expo * &__outcom;RUN;
  ODS OUTPUT CLOSE;
  DATA __ctab1 __ctab2;
    SET __ctab (KEEP = colpercent frequency
                percent &__expo &__outcom);
    exposure_value=&__expo;
    exposure="&__expo";
    outcome="&__outcom";
    outcome_label="&__oclab";
    IF &__outcom= &__oc1 AND
       &__expo>. THEN DO;
         n_controls=FREQUENCY;
         pct_controls=COLPERCENT;
         outcome_value1=&__outcom;
        OUTPUT __ctab1 ;
    END;
    ELSE IF &__outcom= &__oc2 AND
       &__expo>. THEN DO;
         n_cases=FREQUENCY;
         pct_cases=COLPERCENT;
         outcome_value2=&__outcom;
         OUTPUT __ctab2;
    END;
    DROP &__expo.;
%END;
%IF &__ctn = 0 %THEN %DO;
PROC SORT DATA=__ctab1;
  BY exposure_value; RUN;
PROC SORT DATA=__ctab2;
  BY exposure_value; RUN;
PROC SORT DATA=__par;
  BY exposure_value; RUN;
DATA __fstats;
 MERGE __ctab1 (KEEP=exposure_value n_controls
               pct_controls outcome_value1
               exposure outcome outcome_label)
       __ctab2 (KEEP=exposure_value n_cases
               pct_cases outcome_value2
               exposure outcome outcome_label)
       __par;
   BY exposure_value;
  %DO i = 1 %TO &__exp_kt;
     IF exposure_value=&&__exp&i THEN
       exposure_label="&&__fmt&i";
  %END;
RUN;
%END;
/* If the exposure variable passed in the    */
/* macro call contains more than 2 levels and */
/* is not nominal, then run the adjusted model*/
/* with the trend or continuous variable.    */
%IF &__exp_kt>2 AND &__cat=0 %THEN %DO;
  QUIT;
  ODS OUTPUT parameterestimates=__p
             oddsratios=__o;  RUN;
  /* include the base model exposure variables*/
  /* as well as the additional exposure       */
  /* variable for which statisitics are being */
  /* generated in the macro                   */
  PROC LOGISTIC DATA=__logis DESCENDING;
    %IF (&__cvkt NE) %THEN %DO;
```

```
   CLASS
      %DO i = 1 %TO &__cvkt;
          &&__cv&i
      %END;;
    %END;
    MODEL &__outcom =
     %DO i = 1 %TO &__varkt;
         &&__bv&i
     %END; &__expo; RUN;
  ODS OUTPUT CLOSE;
  DATA __p;
    LENGTH exposure $ 32;
    SET __p;
    IF UPCASE(VARIABLE)=UPCASE("&__expo");
    LABEL VARIABLE =
"Name of Exposure Variable";
    exposure=VARIABLE;
    DROP VARIABLE; RUN;
  DATA __o;
    LENGTH exposure $ 32;
    SET __o;
    IF UPCASE(EFFECT)=UPCASE("&__expo");
    LABEL EFFECT =
"Name of Exposure Variable";
    exposure=EFFECT;
    DROP EFFECT; RUN;
  DATA __trend;
    LENGTH exposure $ 32;
    LENGTH exposure_type $ 11;
    LENGTH exposure_label $ 42;
    LENGTH outcome_label $ 42;
    MERGE __p __o;
      BY exposure;
    outcome_value1 = &__oc1;
    outcome_value2 = &__oc2;
    Exposure_label="&__explab";
    n_controls=&__oc_kt1;
    n_cases=&__oc_kt2;
    pct_controls=100;
    pct_cases=100;
    %IF &__ctn = 1 %THEN
        exposure_type="Continuous";
    %ELSE exposure_type="Trend";;
    crude_p_value=&__crudep;
    outcome="&__outcom";
    outcome_label="&__oclab";
    FORMAT crude_p_value pvalue.
           outcome_value1 outcome_value2
           &__ofmt..;
    LABEL oddsratioest   =
'PROC LOGISTIC Adjusted Odds Ratio Estimate'
          outcome_value1 =
'Value of first level for outcome variable'
          outcome_value2 =
'Value of second level for outcome variable'
          probchisq      =
'PEOC LOGISTIC Adjusted p-value'
          exposure_type  =
'Dichotomous/Trend/Class/Continuous'
          crude_p_value  =
'PROC FREQ CMH p-value'
          n_cases        =
'PROC FREQ Frequency of Cases(Column 2)'
          n_controls     =
'PROC FREQ Frequency of Controls(Column 1)'
          pct_cases      =
'PROC FREQ Percent of Cases'
          pct_controls   =
'PROC FREQ Percent of Controls'
          exposure_label =
"Exposure Label(Class/Dichotomous Level Format)"
          outcome        =
"Name of Dependent Outcome Variable"
          outcome_label  =
"Label of Outcome Variable";
   DROP DF ESTIMATE STDERR WALDCHISQ CLASSVAL0;
RUN;
%END;
/* Run the PROC LOGISTIC with the CLASS option*/
/* for both the dichotomous variables and     */
/* variables containing more than 2 levels.   */
/* Statistics will be generated and kept for  */
/* each level determined to be added by the   */
/* CLASS statement.                           */
   QUIT;
%IF &__ctn NE 1 %THEN %DO;
  ODS OUTPUT parameterestimates=__param
             oddsratios=__odds; RUN;
  /* request class variables for the modeling */
  /* of the additional exposure variable in   */
  /* the model.                               */
  PROC LOGISTIC DATA=__logis DESCENDING;
    CLASS
      %DO i = 1 %TO &__cvkt;
          &&__cv&i
      %END;
      &__expo (PARAM=REF REF="&__reffmt"
               ORDER=INTERNAL);
    MODEL &__outcom =
     %DO i = 1 %TO &__varkt;
         &&__bv&i
     %END; &__expo;RUN;
  ODS OUTPUT CLOSE;RUN;
  DATA __param;
    SET __param;
    IF UPCASE(VARIABLE)=UPCASE("&__expo");
    %DO i = 1 %TO &__exp_kt;
        IF CLASSVAL0="&&__fmt&i" THEN
           exposure_value=&&__exp&i.;
    %END;
    DROP VARIABLE DF ESTIMATE STDERR WALDCHISQ
         CLASSVAL0; RUN;
  DATA __odds;
    SET __odds;
    found_var=INDEXW
    (UPCASE(EFFECT),UPCASE(TRIM("&__expo")));
    IF found_var>=1;
    %DO i = 1 %TO &__exp_kt;
  /* continue if the class value is not the   */
  /* referent level                           */
        IF &&__exp&i ^= &__ref THEN DO;
           found_label=
           INDEXW(EFFECT,TRIM("&&__fmt&i"));
           IF found_label>=1 THEN
              exposure_value=&&__exp&i.;
        END;
    %END; /* %DO loop */
    LABEL exposure_value =
"Dichotomous/Class: Value of Exposure Level";
    DROP EFFECT found_var found_label;RUN;
  PROC SORT DATA=__param;
    BY exposure_value;RUN;
  PROC SORT DATA=__fstats;
    BY exposure_value;RUN;
  PROC SORT DATA=__odds;
    BY exposure_value;RUN;
  DATA __cstats;
    LENGTH exposure_type $ 11;
    MERGE __fstats __param __odds;
      BY exposure_value;
    IF &__exp_kt>2 THEN exposure_type='Class';
    ELSE IF &__exp_kt=2 THEN
         Exposure_type='Dichotomous';
    LABEL oddsratioest =
'PROC LOGISTIC Adjusted Odds Ratio Estimate'
          probchisq     =
'PROC LOGISTIC Adjusted p-value'
          exposure_type      =
'Dichotomous/Trend/Class/Continuous';RUN;
%END;RUN;
/* If there are more than 2 levels to the     */
/* exposure variable passed in the macro call,*/
```

```
/* then the trend model was performed.  Add   */
/* the statistics from the trend model to the */
/* dataset containing the statistics from the */
/* model with the 'CLASS' statement.          */
%IF &__exp_kt>2 AND &__ctn = 0 AND &__cat NE 1
%THEN %DO;
   DATA __cstats;
     SET __trend __cstats;RUN;
%END;
%ELSE %IF &__ctn = 1 %THEN %DO;
     DATA __cstats;
       SET __trend;RUN;
%END;
%IF %SYSFUNC(EXIST(__sumary)) %THEN %DO;
    DATA __sumary;
      SET __sumary __cstats;RUN;
%END; %ELSE %DO;
    DATA __sumary;
      SET __cstats;
%END;RUN;
%MEND summary;
```

Appendix 2. Example SAS program

```
%LET log_vars= age @state @smoking_status
               asthma atopy);
/* Include the file that contains the code for*/
/* the macro.                                 */
%INCLUDE 'd:\studies\macros\summary.sas';
/* The dataset '__sumary' will need to be     */
/* removed from the working library if it     */
/* has been created by previous runs of the   */
/* MACRO summary.                             */
PROC DATASETS;
  DELETE __sumary;RUN;QUIT;

/*dataset|outcome |exposure|referent|BaseModel*/
/*       |variable|variable|level   |variables*/
/* ----------------------------------------*/
%summary
(sasdset, WHEEZE,  farmsize,   0    ,&log_vars)
%summary
(sasdset, WHEEZE,  MilkCows,   0    ,&log_vars)
%summary
(sasdset, WHEEZE, #yrsonfarm,  1    ,&log_vars)
%summary
(sasdset, WHEEZE, @eyecolor,   2    ,&log_vars)
```

Appendix 3. PROC CONTENTS of __sumary dataset created by the macro

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Format	Label
	exposure	Char		Name of Exposure Variable
	exposure_label	Char		Exposure Label(Class/Dichotomous:Format)
	exposure_type	Char		Dichotomous/Trend/Class/Continuous
	exposure_value	Num		Dichotomous/Class: Value of Exposure Level
	outcome	Char		Name of Dependent Outcome Variable
	outcome_label	Char		Label of Outcome Variable
	outcome_value1	Num	_NY.	Value of first level for outcome variable
	outcome_value2	Num	_NY.	Value of second level for outcome variable
	n_cases	Num		PROC FREQ Frequency of Cases(Column 2)
	pct_cases	Num		PROC FREQ Percent of Cases
	n_controls	Num		PROC FREQ Frequency of Controls(Column 1)
	pct_controls	Num		PROC FREQ Percent of Controls
	crude_p_value	Num	PVALUE.	PROC LOGISTIC unadjusted p-value
	ProbChiSq	Num	PVALUE6.4	PROC LOGISTIC Adjusted p-value
	OddsRatioEst	Num	ODDSR8.3	PROC LOGISTIC Adjusted Odds Ratio Estimate
	LowerCL	Num	ODDSR8.3	Lower 95% Confidence Limit for Odds Ratio
	UpperCL	Num	ODDSR8.3	Upper 95% Confidence Limit for Odds Ratio

Appendix 4. Example of summary table created using DATA _NULL_

Variable Label/Format	Test Type	WHEEZE Yes N	WHEEZE Yes %	WHEEZE No N	WHEEZE No %	P-Value Crude	P-Value Adjusted	OR	95% CL	
Size of Farm	Trend	3831	100	16672	100	0.0001	0.0008	1.071	1.029	1.115
1) No Farm	Class	235	6.1	1055	6.3	.	.			
2) 1-99 acres	Class	914	23.9	4215	25.3	0.7396	0.4578	1.067	0.900	1.265
3) 100-499 acres	Class	1545	40.3	7134	42.8	0.7162	0.2008	1.114	0.944	1.314
4) 500-999 acres	Class	998	26.1	3736	22.4	0.0239	0.0037	1.286	1.085	1.524
5) 1000+ acres	Class	139	3.6	532	3.2	0.1818	0.3988	1.114	0.867	1.431
Milk Cows(Yes)	Dichotomous	239	6.3	908	5.5	0.0543	0.0011	1.297	1.109	1.517
Years lived on farm	Continuous	3438	100	14850	100	<.0001	<.0001	1.066	1.057	1.076
Eyes=Blue	Class	2392	63.2	11555	70.1	<.0001	<.0001	0.780	0.707	0.860
Eyes=Brown	Class	831	22.0	2963	18.0	.	.			
Eyes=Green	Class	338	8.9	1209	7.3	0.9653	0.2818	0.918	0.786	1.073
Eyes=Hazel	Class	146	3.9	478	2.9	0.4046	0.2928	0.889	0.714	1.107
Eyes=Black	Class	75	2.0	277	1.7	0.7957	0.0941	0.782	0.586	1.043

Acknowledgments

The authors would like to thank David Shore for his assistance and review of the methods, coding, and manuscript.

Contact Information

Stuart Long, Rebecca Darden
Westat
1009 Slater Road, Suite 110
Durham, NC 27703

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