

## Williams' Test: The Macro

Derek B. Janszen, Chemical Industry Institute of Toxicology, Research Triangle Park, NC;  
and William P. Hahn, North Carolina State University, Raleigh, NC

### ABSTRACT

Williams' test (Williams, 1971; 1972) is used in pharmacology and toxicology for determining the lowest dose group that is significantly different from the control group. It assumes a strictly increasing or decreasing response with increasing dose and is more powerful (can declare smaller differences to be significant) than Dunnett's test. In SAS/STAT<sup>®</sup> Version 6.08 (SAS Institute Inc., 1996), "WILLIAMS" was introduced as an option for the `probsmc()` function. Given the maximum likelihood estimates (MLE) of the monotonic responses, `probsmc()` computes the p-values. Frequently, the actual responses are not monotonic. This article describes a macro that computes the MLE of the monotonic responses for a set of data (regardless of the actual response pattern) and the corresponding p-values for Williams' test. It requires only the base and SAS/STAT modules of the SAS<sup>®</sup> System. The macro is well documented. Among the key features are the ability to accept data in either a columnar or stacked format, to perform several different transformations on the data, and to perform Shirley's test (Shirley, 1977), a nonparametric analog of Williams' test.

### INTRODUCTION

In the fields of pharmacology and toxicology, one is frequently interested in the question of whether there is an overall increasing or decreasing response with an increase in dose. Various tests have been devised to determine if the trend is significant. Another question of interest is, "What is the lowest dose that is significantly different from the control group?" Dunnett's test is frequently used to determine those treatment groups that are significantly different from the control group. When the responses are strictly increasing or decreasing (monotonic), Dunnett's may be sufficient. Frequently, however, the higher doses may have a decreased response, whereas the lower doses have a minimal or an increased response. Thus, Dunnett's test may determine that one of the lower doses is different from control but that a higher dose is not different from control. This makes interpretation of the results difficult. However, if the response is expected to be monotonic but is not perhaps due to sampling error, then Williams' test may be applied. Williams determined how to compute the p-values based on the assumption of a monotonic response. Furthermore, he showed how to compute the MLE of the responses under this assumption. In Williams' test, if a dose group is found that is significantly different from the control group, then all higher dose groups are also significantly different.

Williams' test first appeared as a part of the SAS System in PROC TABLES, a procedure written by Stan Young and George Fraction (1986) for the SUGI Supplemental Library Version 5. The procedure was originally written to produce publication-ready tables. The p-values for Williams' test were determined either by a table-lookup method or by approximating the p-value through the use of Chebyshev polynomials (S.S. Young, personal communication). In SAS/STAT Version 6.07, the `probsmc()` function was introduced; in Version 6.08, "WILLIAMS" was included as a test for which p-values could be computed. The function requires the number of treatment groups besides the control group, degrees of freedom (df) for the mean squared error (MSE), the root mean squared error, and the monotonic MLE of the treatment means. The macro to be described computes the MLE, the p-value for the comparison of the treatment mean against the control, and summary statistics for the treatment groups.

### DESCRIPTION OF MACRO

A copy of the macro is included as an Appendix to this article. A brief description of the rationale and an explanation of various parts of the macro follows, which may aid the user who wants to modify the macro to suit his or her needs. The macro is well documented to facilitate modification.

Data sets are usually created in either a stacked format (all observations in one column, with a grouping variable containing the names of the treatment groups in another column) or columnar format (the observations for each treatment group are in separate columns). The macro utilizes the "parmbuff" option to allow for both types of formats. The macro can also perform various common transformations prior to analysis. If one has percentage-type data that are strictly between 0 and 100% (for example, labeling indices), the "PCT" data\_type is specified, and the data are arcsine-transformed. If one has skewed data or percentages that can be greater than 100% (for example, body weight percentage gain), the "RATIO" data\_type does a ln-transformation. For data that are not approximately normally distributed, even with a transformation, the "NONP" data\_type performs Williams' test on the ranks of the data using Shirley's test. For data that do not require a transformation, the "NORM" data\_type is specified.

For a stacked data format, the macro is invoked with the following arguments:

```
%williams(stack, SAS_dataset_name, variable_name,
           grouping_variable_name, data_type,)
```

The data must be stacked in the order CONTROL, LOW\_DOSE → HIGH\_DOSE. For columnar type data, the arguments are:

```
%williams(col, SAS_dataset_name, variable_name,
           grouping_variable_name, data_type,
           comma_separated_list_of_columns)
```

The list of columns must be in the order CONTROL, LOW\_DOSE → HIGH\_DOSE.

The macro proceeds by scanning the argument, assigning values to various macro variables (data, var, group, type), manipulating the data into a columnar format if the "STACK" format was selected, and printing the original data set. It then stacks the data, performs any necessary transformations, and uses PROC GLM to obtain the MSE and least-squares means of the dose groups. PROC REG is used to determine if the trend is increasing or decreasing, and PROC MEANS computes the mean, standard error of the mean, and number of observations for each dose group.

The monotonic MLE are computed in the DATA "mhat" step, which utilizes arrays. Arrays are filled by row order. The following code finds the minimum value for each row by listing the variables for a particular row. If there is an upward trend (dec=0), we want the minimum of rats&x-rats&y, where rats&x is the same as rats(j,u,1) and rats&y is the same as rats(j,u,k). But if there is a decreasing trend (dec=1), we want the maximum of rats&x-rats&y. The same array (named "mins") is used to hold these values. "k" is the number of dose groups besides CONTROL; it was determined and assigned when the data were first manipulated.

```
%do j = 1 %to &k;
  %do u = 1 %to &j;
    %let x =
      %eval(1 + (&u-1)*&k + (&j-1)*&k*&k);
    %let y =
      %eval(&k + (&u-1)*&k + (&j-1)*&k*&k);
    if (&dec) then
      mins(&j,&u) = max(of rats&x-rats&y);
    else mins(&j,&u) = min(of rats&x-rats&y);
  %end;
%end;
```

The next step is to find the largest minimum value for an increasing trend or the smallest maximum value for a decreasing trend. The same array, "mhats", is used for both.

```

%do j = 1 %to &k;
  %let x = %eval(1 + (&j-1)*&k);
  %let y = %eval(&k + (&j-1)*&k);
  if (&dec) then
    mhats(&j) = min(of mins&x-mins&y);
  else mhats(&j) = max(of mins&x-mins&y);
%end;

```

The Williams' t-statistic is then computed, and the corresponding p-value is determined using the probmc() function, with the appropriate arguments. An output table is created and printed. System options are reset, and the temporary data sets created by the macro are deleted.

#### EXAMPLE

The following data are from a study in which the tumor-promoting potential of PCB-118 was evaluated in a two-stage initiation-promotion bioassay in female Sprague-Dawley rats (Haag-Grönlund et al., 1997). The data represent the number of foci per cubic centimeter positive for glutathione S-transferase P after initiation by intra-peritoneal administration of *N*-nitrosodiethylamine followed by subcutaneous administration of PCB-118 at 6 increasing dose levels for 20 weeks.

| 0    | 10    | 40   | 160   | 640   | 2500  | 10000 |
|------|-------|------|-------|-------|-------|-------|
| 3387 | 9295  | 1506 | 4741  | 7479  | 7536  | 12521 |
| 2638 | 5829  | 5502 | 7081  | 7225  | 10405 | 7134  |
| 6787 | 6710  | 5771 | 4520  | 7391  | 12334 | 9276  |
| 5208 | 4701  | 3028 | 7295  | 12242 | 11859 | 2469  |
| 6071 | 5312  | 5846 | 5502  | 6589  | 10483 | 11855 |
| 6724 | 3240  | 4823 | 4462  | 7902  | 7558  | 19006 |
| 6076 | 5568  | 4168 | 6981  | 7148  | 7868  | 4561  |
| 8177 | 7128  | 7295 | 10588 | 5960  | 9584  | 7375  |
| 6403 | 11035 | 8129 | 4550  | 4439  | 7317  | 14145 |
| 3617 | 5819  | 4717 |       | 8921  | 7948  | 8973  |
| 4424 |       |      |       |       |       |       |
| 4814 |       |      |       |       |       |       |
| 7348 |       |      |       |       |       |       |
| 5716 |       |      |       |       |       |       |

The original analysis used Duncan's multiple range test on ln-transformed data and found that only the two highest dose groups were different from the control group. The same results were obtained with Dunnett's test. The following partial output from the Williams' test macro shows that the 640- $\mu$ g/kg group is also significantly different from control. Note that the two lowest dose groups have the same p-value. This result occurs because the mean of the 40- $\mu$ g/kg group was lower than the mean of the 10- $\mu$ g/kg group, and the assumption of a monotonically increasing response meant that the same MLE for the mean was used for both groups. The same reasoning applies to the two highest dose groups.

| DOSE  | N  | TRANS-<br>FORMED<br>MEAN | TRANS-<br>FORMED<br>SEMEAN | WILLIAMS<br>TEST<br>p-VALUE |
|-------|----|--------------------------|----------------------------|-----------------------------|
| 0     | 14 | 3.96751                  | 0.08664                    | -                           |
| 10    | 10 | 4.11587                  | 0.10861                    | 0.6261                      |
| 40    | 10 | 3.84019                  | 0.15312                    | 0.6261                      |
| 160   | 9  | 4.08348                  | 0.09983                    | 0.3257                      |
| 640   | 10 | 4.29038                  | 0.08262                    | 0.0263                      |
| 2500  | 10 | 4.51316                  | 0.06327                    | 0.0009                      |
| 10000 | 10 | 4.44259                  | 0.18694                    | 0.0009                      |

Bartlett's test of homogeneity for the ln-transformed data was rejected ( $p = 0.0289$ ). However, Shirley's test also found the 640- $\mu$ g/kg group to be different from control ( $p = 0.0098$ ).

#### CONCLUSION

Williams' test is a powerful method for evaluating dose-response data. The ability to determine p-values by this method was included in SAS/STAT Version 6.08. A macro that requires only the base and

SAS/STAT modules has been written to calculate the ordered MLE and to determine the corresponding p-values. Features of the macro include the ability to use data in a stacked or columnar format, to transform data prior to analysis, and to perform Shirley's test. The macro is well documented, facilitating the inclusion of additional features such as additional transformations of data, tests for homogeneity of variance, and graphical plots.

#### REFERENCES

Haag-Grönlund, M., Wärmgård, L., Flodström, S., Scheu, G., Kronevi, T., Ahlborg, U.G., and Fransson-Steen, R. (1997). "Promotion of altered hepatic foci by 2,3',4,4',5-pentachlorobiphenyl in Sprague-Dawley female rats." *Fundam. Appl. Toxicol.* 35: 120-130.

SAS Institute Inc. (1996). *SAS/STAT® Software: Changes and Enhancements through Release 6.11*, Cary, NC: SAS Institute Inc.

Shirley, E. (1977). "A non-parametric equivalent of Williams' test for contrasting increasing dose levels of a treatment." *Biometrics* 33: 386-389.

Williams, D.A. (1971). "A test for differences between treatment means when several dose levels are compared with a zero dose control." *Biometrics* 27: 103-117.

Williams, D.A. (1972). "The comparison of several dose levels with a zero dose control." *Biometrics* 28: 519-531.

Young, S.S. and Fraction, G.F. (1986). "The TABLES Procedure." In *SUGI Supplemental Library User's Guide, Version 5 Edition* (SAS Institute Inc.). Cary, NC: SAS Institute Inc., pp. 579-605.

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

Derek B. Janszen, Ph.D.  
Chemical Industry Institute of Toxicology  
6 Davis Drive  
Research Triangle Park, NC 27709  
(919) 558-1390  
janszen@ciit.org

William P. Hahn, Ph.D.  
Dept. of Mathematics  
Baylor University  
Dallas, TX

#### APPENDIX

[The macro is available via e-mail. Send request to janszen@ciit.org]

```

%MACRO williams(x)/parmbuff;
/* some type of argument is required. */

/* get values of macro variables */
%let arrange = %scan(&syspbuff,1,',');
%let data = %scan(&syspbuff,2,',');
%let var = %scan(&syspbuff,3,',');
%let group = %scan(&syspbuff,4,',');
%let type = %scan(&syspbuff,5,',');

%let nonp = %eval("%upcase(&type)"="NONP");
%let pct = %eval("%upcase(&type)"="PCT");
%let ratio = %eval("%upcase(&type)"="RATIO");

```

```

/* The following sets up the original data to
be printed as part of the output. It
determines how many dose groups there are in
addition to the control group, and assigns
that number to the macro variable "k". If the
keyword COL or STACK is not the first word of
the macro argument, exit the macro. */

```

```

%if %upcase(&arrange)=STACK %then
%do;
  data original(keep=&group &var);
  set &data(rename=(&group=zzzzzzz));
  format &group $20.;
  &group = left(zzzzzzz);
run;
proc transpose data=original out=out1
  (drop=_name_ _label_);
  by &group notsorted;
run;
data out1;
  length dosez $7.;
  set out1;
  by &group notsorted;
  if (_n_ = 1) then dosez = 'CONTROL';
  else if (_n_ < 11) then
    dosez = 'DOSE_'||compress(_n_-1);
  else dosez = 'DOSE_'||compress(_n_-1);
  if last.&group then
    call symput('k', compress(_n_-1));
run;
proc transpose data=out1
  out=original(drop=_name_);
  id &group;
run;
proc transpose data=out1
  out=out2(drop=_name_);
  id dosez;
run;
/* make column of grouping variable names */
data grpnames(keep=&group); set out1; run;
%end;
%else %if %upcase(&arrange)=COL %then
%do;
  %let stop = %length(&syspbuff);
  %let ctl = %scan(&syspbuff,6,',');
  data original(keep=&ctl);
  set &data;
run;
data out2(rename=(&ctl=control));
  set original;
run;
%do i=1 %to &stop;
  %let column =
    %scan(&syspbuff,%eval(&i+6),',');
  %if &column ne %then
  %do;
    data out1(keep=&column);
    set &data;
run;
    data original;
    merge original out1;
run;
    %if %eval(&i < 10) %then
    %do;
      data out2(rename=
        (&column=dose_&i));
      merge out2 out1;
run;
    %end;
  %else
  %do;
    data out2(rename=
      (&column=dose&i));
    merge out2 out1;
run;
  %end;
%end;

```

```

%let k = &i;
%end;
%else %let i = &stop;
%end;
/* make column of grouping variable names */
proc transpose data=original out=grpnames
  (rename=(name=&group) keep=_name_);
run;
%end;
%else
%do;
  %put The first argument to macro WILLIAMS
  must be the word "STACK" or "COL";
  %put Exiting macro WILLIAMS...;
  %goto exit_mac;
%end;

```

```

options missing=' ';
title "Original Data for Variable
%upcase(&var) by Grouping Variable
%upcase(&group)";
title2 "Dataset %upcase(&data)";
title4 "SAS macro written by William P. Hahn
and Derek B. Janszen";
title5 "CIIT, 6 Davis Drive, RTP, NC 27709";
proc print data=original; run;
options formdlm=' ';

```

/\* The following code stacks the data into one variable (value), and adds three other variables to the dataset, one for dose (taking values CONTROL, DOSE1-DOSE&k), one for dos\_valu (determination of increasing/decreasing trend uses proc reg, which needs a numerical independent variable), and one for compound (taking the value 1 for every observation). The compound variable is included because when a BY statement is used with SET, MERGE, or UPDATE in a data statement, FIRST.byvariable and LAST.byvariable are created. Since the by-variable compound takes only one value, FIRST.compound acts as an indicator of the first observation and LAST.compound acts as an indicator of the last observation in the dataset. When this dataset is sorted by group, the original order is maintained. \*/

```

data stak (keep=value dosez origdata
  dos_valu compound);
  set out2;
  value = control;
  dosez = "CONTROL";
  origdata = value;
  dos_valu = 0;
  compound = 1;
  if value ne .;
run;
%do i=1 %to &k;
  data data&i (keep=value dosez origdata
    dos_valu compound);
    set out2;
    %if %eval(&i<10) %then
    %do;

```

```

        value = dose&i;
        dosez = "DOSE&i";
    %end;
%else
    %do;
        value = dose&i;
        dosez = "DOSE&i";
    %end;
    origdata = value;
    dos_valu = &i;
    compound = 1;
    if value ne .;
run;
data stak; set stak data&i; run;
%end;

/* If necessary, perform the
   appropriate transformation */
%if &nonp %then
    %do;
        proc rank data=stak out=rankdata;
            var value;
            ranks rankvar;
        run;
        data stak;
            set rankdata;
            value = rankvar;
        run;
    %end;
%else
    %do;
        data stak;
            set stak;
            if &pct then
                value = arsin(sqrt(value/100));
            else if &ratio then
                value = log(value);
        run;
    %end;
proc sort data=stak; by compound dosez; run;

/* do some cleanup */
proc datasets library=work nolist;
    delete out1 out2 original;
    %if &nonp %then delete rankdata;
    %do i=1 %to &k;
        delete data&i;
    %end;
run;

/* Do ANOVA of value (the response variable)
   on dose to calculate mse and to obtain least-
   squares means of the doses. */
proc glm data=stak outstat=parms noprint;
    by compound;
    class dosez;
    model value=dosez/nouni;
    lsmeans dosez/out=lsmsns;
run;
proc sort data=lsmsns; by compound dosez; run;

/* The following dataset uses the output from
   proc glm to get an estimate of mse and its
   degrees of freedom (df_e). */
data mse(keep=mse df_e compound);
    set parms; by compound;
    if (_type_ eq "ERROR");
    mse = ss/df;

```

```

        rename df = df_e;
run;
proc sort data=mse; by compound; run;

/* determine if trend is pos or neg and assign
   value to macro var "dec" */
proc reg data=stak outest=reg_beta noprint;
    model value = dos_valu;
run;
data _null_;
    set reg_beta;
    x = (dos_valu < 0);
    call symput('dec', compress(put(x,12.)));
run;

/* Compute the mean, standard error, and count
   for each dose, putting them into the dataset
   stats. */
proc means data=stak noprint;
    by compound dosez;
    var origdata value;
    output out=stats mean=mn_orig mn_tran
        std=sd_orig sd_tran
        stderr=sem_orig sem_tran n=n;
run;

/* The following dataset puts the means of the
   doses into variables mean0-mean&k, the
   standard errors into sem0-sem&k, and the
   counts into n0-n&k. In SAS, an array is just
   an ordered list of variable names, i.e. the
   elements of the array act as pseudonyms for
   the variables in the list. */
data mns (keep=mean0-mean&k sem0-sem&k
    n0-n&k compound);
    set stats; by compound;
    array ns      (*) n0-n&k;
    array mns     (*) mean0-mean&k;
    array sems    (*) sem0-sem&k;
    retain mean0-mean&k sem0-sem&k n0-n&k;

    if first.compound then i=0;
    i+1;
    mns(i) = mn_tran;
    sems(i) = sem_tran;
    ns(i) = n;
    if last.compound then output;
run;

/* The following creates the dataset lsmns by
   merging datasets lsmns and stats, and contains
   the counts (in n0-n&k) and least-squares means
   (in lsmean0-lsmean&k) for dose groups 0
   through k (i.e. including the control group).
   Note: this dataset contains 1 observation and
   multiple variables. */
data lsmns(drop=i _name_ lsmean stderr n
    dosez);
    merge lsmns stats(keep=n compound dosez);
    by compound dosez;
    array ns      (*) n0-n&k;
    array lsmns   (*) lsmean0-lsmean&k;
    array doses   (*) $8. dose0-dose&k;
    retain lsmean0-lsmean&k n0-n&k dose0-dose&k;

    if first.compound then i=0;
    i+1;
    lsmns(i)=lsmean;

```

## statistics

```

ns(i)=n;
doses(i)=dosez;
if last.compound then output;
run;

/* The following dataset calculates Williams
mhat1 through mhatk. */
data mhat (keep=mhat1-mhat&k compound);
set lsmns; by compound;
%let kkk = %eval(&k*&k*&k);
%let kk = %eval(&k*&k);
array ns (*) n1-n&k;
array lsmns (*) lsmean1-lsmean&k;
array nums {&k,&k} nums1-nums&kk;
array dens {&k,&k} dens1-dens&kk;
array rats {&k,&k,&k} rats1-rats&kkk;
array mins {&k,&k} mins1-mins&kk;
array mhats (*) mhat1-mhat&k;

do u = 1 to &k;
  nums(u,u) = ns(u)*lsmns(u);
  dens(u,u) = ns(u);
end;
do u = 1 to &k-1;
  do j = 1 to &k-u;
    nums(u,u+j) = nums(u,u+j-1) +
      ns(u+j)*lsmns(u+j);
    dens(u,u+j) = dens(u,u+j-1) + ns(u+j);
  end;
end;
do j = 1 to &k;
  do u = 1 to j;
    do v = j to &k;
      rats(j,u,v) = nums(u,v)/dens(u,v);
    end;
  end;
end;

/* since SAS fills an array by row order, the
following allows us to find the minimum value
for each row by listing the variables for a
particular row */
%do j = 1 %to &k;
  %do u = 1 %to &j;
    %let x = %eval(1 + (&u-1)*&k +
      (&j-1)*&k*&k);
    %let y = %eval(&k + (&u-1)*&k +
      (&j-1)*&k*&k);
  /* If there is an upward trend (dec=0), we
  want the minimum of rats&x-rats&y, where
  rats&x is the same as rats(j,u,1) and rats&y
  is the same as rats(j,u,&k). But if there is a
  decreasing trend (dec=1), we want the maximum
  of rats&x-rats&y. The same array (named
  "mins") is used to hold these values. */
  if (&dec) then
    mins(&j,&u) = max(of rats&x-rats&y);
  else mins(&j,&u) = min(of rats&x-rats&y);
  %end;
%end;

/* same thing except we're finding the maximum
of the minimums (for increasing trend; we need
the minimum of the maximums for a decreasing
trend; the same array, mhats, is used for
both) */
%do j = 1 %to &k;
  %let x = %eval(1 + (&j-1)*&k);

```

```

%let y = %eval(&k + (&j-1)*&k);
if (&dec) then
  mhats(&j) = min(of mins&x-mins&y);
else mhats(&j) = max(of mins&x-mins&y);
%end;
run;

/* The following dataset calculates the t
needed for Williams Test and finds the
accompanying p-values. In the merge statement,
the dataset mns contains (compound mean0-
mean&k sem0-sem&k n0-n&k); lsmns contains
(compound lsmean0-lsmean&k dose0-dose&k); mhat
contains (compound mhat1-mhat&k). */
data t(keep=name_t t0-t&k)
  prob(keep=name_p prob0-prob&k);
merge mns lsmns mhat mse; by compound;
array ns {0:&k} n0-n&k;
array ts {0:&k} t0-t&k;
array probs {0:&k} prob0-prob&k;
array mhats {1:&k} mhat1-mhat&k;

do j = 1 to &k;
  if (&dec) then
    ts(j) = (lsmean0-mhats(j))/
      sqrt((mse/ns(j))+(mse/n0));
  else ts(j) = (mhats(j)-lsmean0)/
    sqrt((mse/ns(j))+(mse/n0));
  probs(j) =
    1-probmc("williams",ts(j),,df_e,&k);
end;
name_t = 'wms_t'; name_p = 'p_values';
run;

/* The following transposes the variables t0-
t&k into 'Wms t', and the variables prob0-
prob&k into 'P values'. */
proc transpose data=t out=t;
  id name_t;
  var t0-t&k;
run;
proc transpose data=prob out=prob;
  id name_p;
  var prob0-prob&k;
run;
data stats (drop=name _type _freq);
  merge stats t prob;
run;
proc sort data=stats; by dosez; run;

title; /* clear all previous titles */
options nodate nonumber;

/* print results */
%if %eval(&pct | &ratio) %then
  %do; /*inverse transform to original scale*/
    %if &pct %then
      %do;
        %let transfrm = arcsine(sqrt(data));
        where 0 <= data <= 1;
        data stats(drop=compound);
        merge grpnames stats(drop=dosez);
        mn_un = (sin(mn_tran)**2)*100;
        sem_un =
          ((sin(mn_tran)**2) - (sin(mn_tran -
            tinv(.8413,n-1)*sem_tran)**2))*100;
        /* create 95% confidence intervals for mean */

```

```

lo_ci = (sin(mn_tran -
  tinv(.975,n-1)*sem_tran)**2)*100;
hi_ci = (sin(mn_tran +
  tinv(.975,n-1)*sem_tran)**2)*100;
run;
%end;
%else
%do;
%let transform = natural_log(data);
data stats(drop=compound);
merge grpnames stats(drop=dosez);
mn_un = exp(mn_tran);
sem_un =
  exp(mn_tran) - exp(mn_tran -
    tinv(.8413,n-1)*sem_tran);
/* create 95% confidence intervals for mean */
lo_ci = (exp(mn_tran -
  tinv(.975,n-1)*sem_tran);
hi_ci = (exp(mn_tran +
  tinv(.975,n-1)*sem_tran);
run;
%end;
proc print data=stats label noobs;
title 'Descriptive Statistics';
title2 'THESE VALUES ARE NOT VALID FOR
**ANY** STATISTICAL TESTS';
var &group n mn_orig sd_orig sem_orig;
label &group='%qupcase(&group)'
  mn_orig='MEAN' sd_orig='STDEV'
  sem_orig='SEMEAN';
run;
proc print data=stats label
  split='*' noobs;
title2 'for Transformed and
Untransformed Data';
title3 'and Results for Williams''
Test';
title5 '***NOTE: Untransformed SEMEAN
values are not valid for use in other
testing';
format wms_t p_values 9.4;
var &group n mn_tran sem_tran mn_un
  sem_un lo_ci hi_ci wms_t p_values;
label mn_tran='TRANS-*FORMED*MEAN'
  sem_tran='TRANS-*FORMED*SEMEAN'
  mn_un='UNTRANS-*FORMED*MEAN'
  sem_un='UNTRANS-*FORMED*SEMEAN'
  lo_ci='LOWER*95% CI*FOR MEAN'
  hi_ci='UPPER*95% CI*FOR MEAN'
  wms_t='WILLIAMS*TEST t-*STATISTIC'
  p_values='WILLIAMS*TEST*p-VALUE'
  &group='%qupcase(&group)';
run;
%end;
%else
%do;
data stats(drop=compound);
merge grpnames stats(drop=dosez);
lo_ci = mn_orig -
  tinv(.975,n-1)*sem_orig;
hi_ci = mn_orig +
  tinv(.975,n-1)*sem_orig;
run;
proc print data=stats
  label split='*' noobs;
title 'Descriptive Statistics';
title2 'and Results for Williams'' Test';
%if &nomp %then

```

```

%str(title4 'NOTE: Williams'' test
performed on ranks of data (Shirley''s
test)'););
format wms_t p_values 9.4;
var &group n mn_orig sd_orig sem_orig
  lo_ci hi_ci wms_t p_values;
label mn_orig='MEAN' sd_orig='STDEV'
  sem_orig='SEMEAN'
  lo_ci='LOWER*95% CI*FOR MEAN'
  hi_ci='UPPER*95% CI*FOR MEAN'
  wms_t='WILLIAMS*TEST t-*STATISTIC'
  p_values='WILLIAMS*TEST*p-VALUE'
  &group='%qupcase(&group)';
run;
%end;

/* clear titles and append references to end
of output page */
title;
data _null_;
file print;
%if &nomp %then
%do;
put 'REFERENCE:';
put 'Shirley, E. 1977. A non-parametric
equivalent of Williams'' test for';
put '      contrasting increasing
dose levels of a treatment. Biometrics 33:
386-389.';
%end;
%else
%do;
put 'REFERENCES:';
put 'Williams, D.A. 1971. A test for
differences between treatment means when
several dose levels';
put '      are compared with a zero
dose control. Biometrics 27: 103-117.';
put 'Williams, D.A. 1972. The comparison
of several dose levels with a zero dose
control.';
put '      Biometrics 28: 519-531.';
%end;
run;

/* reset system options and delete all
datasets used in macro */
options formdlm=' ' date number missing='.';
proc datasets library=work nolist;
delete stak stats mns parms lsms mlms
  reg_beta mhat mse t prob grpnames;
run;
%exit_mac: %MEND williams;

```