

A Macro for Calculating Kendall's Tau-b Correlation on Left Censored Environmental Data

Dennis J. Beal, Leidos, Oak Ridge, Tennessee

ABSTRACT

Kendall's tau-b correlation is a nonparametric method for calculating monotonic correlation between two variables with multiple censoring levels. While SAS® does calculate Kendall's tau-b as an option in the PROC CORR procedure within SAS/BASE, its calculation assumes all values are known and uncensored. It does not incorporate censoring that is common with chemical or environmental data. Environmental data often are reported from the analytical laboratory as left censored, meaning the actual concentration for a given contaminant was not detected above the method detection limit. Therefore, the true concentration is known only to be between 0 and the reporting limit. Kendall's tau-b can be used on left censored data with multiple reporting limits with minimal assumptions. This paper presents a SAS macro that calculates Kendall's tau-b by incorporating the additional ties that occur when comparing detected concentrations with non-detects. This paper is for intermediate SAS users of SAS/BASE.

Key words: Kendall's tau-b, nonparametric, left censoring, environmental data, correlation

INTRODUCTION

Statisticians, environmental scientists, and risk assessors collect environmental data for many purposes, including characterization, quantification of risks and hazards, waste disposal, and compliance to applicable environmental regulations. Environmental analytical data that are left-censored at the reporting limit are called "non-detects" and often qualified with a laboratory or validation qualifier of "<" or "U." Unlike detected data that are reported as measured concentrations and are uncensored, for chemicals the estimated concentration for non-detects is known only to be within the interval from 0 to the reporting limit provided by the laboratory. For example, the laboratory may report a detected (and hence uncensored) arsenic concentration of 10 mg/kg in soil from sample 1 and <5 (or 5 U) mg/kg from sample 2. The censored concentration X in sample 2 is known only to be in the interval $0 \leq X < 5$ mg/kg. However, for radionuclides that are reported as background corrected, negative and zero concentrations can be reported with "U" qualifiers, so the lower bound of the censoring interval may extend below 0 (Beal 2015 and Beal 2010).

This paper will focus on applying the nonparametric Kendall's tau-b method to chemicals for calculating monotonic correlation between two laboratory chemicals. This method is not appropriate for radionuclides because the actual reported concentration (whether positive, negative, zero, non-detected or detected) can be used as reported from the laboratory. For radionuclides the Pearson product-moment correlation coefficient is more appropriate because a numerical value is reported for all concentrations and the non-detects are not left censored. SAS calculates the Pearson product-moment correlation coefficient that measures linear correlation in the PROC CORR procedure.

Spearman's rho is a rank-transform measure of the monotonic association between two variables. However, Spearman's rho assumes only one censoring level where all non-detects are reported at the same detection limit. Any detected concentrations reported below the detection limit must be recensored at the highest detection limit in order to compute Spearman's rho (Helsel 2005). The SAS code presented in this paper uses the SAS System for personal computers version 9.4 running on a Windows® 7 Professional platform.

KENDALL'S TAU-B METHOD

The Kendall's tau-b correlation can be applied to multiply censored data. It is computed by comparing all pairs of observations, counting all positive differences between both the X and Y variables as positive and all negative differences as negative. Kendall's tau-b does not assume a normal distribution. Helsel (2012) also describes Kendall's tau-b in more detail.

Let $x_1, x_2, \dots, x_n$ represent the n concentrations (either detected concentrations or non-detects) for the first chemical obtained from environmental samples sorted in order from smallest to largest. Let $y_1, y_2, \dots, y_n$ represent the n concentrations (either detected concentrations or non-detects) for the second chemical for the same station, date collected and laboratory sample number. Pairs of results (x_1, y_1) and (x_2, y_2) are concordant if $x_2 > x_1$ and $y_2 > y_1$ or $x_2 < x_1$ and $y_2 < y_1$. That is, the pairs are concordant if both the X and Y variables increase or both variables decrease

together. Pairs are discordant if either $x_2 > x_1$ and $y_2 < y_1$ or $x_2 < x_1$ and $y_2 > y_1$. That is, pairs are discordant if X increases and Y decreases or X decreases and Y increases.

A tie is when $x_1 = x_2$ or $y_1 = y_2$ or both. When multiple censoring levels are present, all non-detects are considered tied because it is unknown which of the two non-detects is greater or less than the other. For example, if the non-detects <5 and <10 are compared, the <5 could be greater than <10 if the true unknown concentration for <5 is 4 and the true unknown concentration for <10 is 2. Or the <10 could be greater than the <5 if the true unknown concentration for <10 is 9 and the true unknown concentration for <5 is 3. Or both concentrations could be tied at 1. Because we do not know the true concentrations for a non-detect, all non-detects are considered tied with each other.

Non-detects can also be considered tied with detected concentrations. If the non-detect concentration is greater than a detected concentration, it is unknown which concentration is greater or less than the other. For example, if <10 is compared with a detected 5, then the <10 could have a true unknown concentration of 8, which is greater than 5. Or the <10 could have a true unknown concentration of 2, which is less than 5. Because it is unknown which is greater in this situation, we consider non-detects tied with detects when the non-detect detection limit is greater than the detected concentration.

However, a detected concentration reported with the same numeric value as a non-detect is not a tie. For example, a comparison of <5 with a detected 5 shows the detected 5 is greater than the <5. Similarly, a detected 5 is obviously greater than a <3 and <1.

All possible $n(n-1)/2$ pairs are compared. The number of concordant pairs is summed (N_c) along with the number of discordant pairs (N_d). The number of ties for the X variable is summed (T_x) and the number of ties for the Y variable is summed (T_y).

The Kendall's tau-b (τ_b) is then calculated using Equation 1.

$$\tau_b = \frac{N_c - N_d}{\sqrt{\left(\frac{n(n-1)}{2} - T_x\right)\left(\frac{n(n-1)}{2} - T_y\right)}} \quad (1)$$

Kendall's tau-b ranges from -1 (perfect negative monotonic correlation) to 1 (perfect positive monotonic correlation). We define S is the number of concordant pairs (N_c) minus the number of discordant pairs (N_d) as shown in Equation 2.

$$S = N_c - N_d \quad (2)$$

In order to construct a test statistic, we must calculate the variance of S. Kendall and Gibbons (1990) describes the variance of S in the presence of ties as shown in Equations 3 through 9.

We define u as the number of tied detected concentrations at each concentration level in the X variable (chemical 1). For example, if the X variable consists of the detected concentrations 1, 3, 5, 5, 7, 7, 7, 10, 10, 10, and 10, then $u = 2$ tied concentrations at 5, $u = 3$ tied concentrations at 7 and $u = 4$ tied concentrations at 10. When there are no tied concentrations, $u = 1$ at each concentration level. So $u = 1$ at 1 and $u = 1$ at 3 in this example. However, all non-detects are considered tied at the same concentration level even if they have different reporting limits. For example, if the X variable consists of the non-detects <1, <3, <5, <5, <5, <10, and <10 then $u = 7$ tied non-detects. Equation 3 then sums the u number of ties at each distinct detected concentration and all non-detects as one censoring level

$$U_1 = \sum^k u(u-1)(2u+5) \quad (3)$$

for k distinct detected concentrations in X with ties and the one censoring level for all non-detects combined (if any).

Similarly, we define v as the number of tied detected concentrations at each concentration level in the Y variable (chemical 2) as well as the one combined censoring level for non-detects as shown in Equation 4

$$V_1 = \sum^m v(v-1)(2v+5) \quad (4)$$

for m distinct detected concentrations in Y with ties and the one censoring level for all non-detects combined (if any).

$$U_2 = \sum_{u=1}^k u(u-1)(u-2) \quad (5)$$

$$V_2 = \sum_{v=1}^m v(v-1)(v-2) \quad (6)$$

$$U_3 = \sum_{u=1}^k u(u-1) \quad (7)$$

$$V_3 = \sum_{v=1}^m v(v-1) \quad (8)$$

Using Equations 3 through 8 the variance of S is shown in Equation 9.

$$\text{Var}(S) = \frac{n(n-1)(2n+5) - U_1 - V_1}{18} + \frac{U_2 V_2}{9n(n-1)(n-2)} + \frac{U_3 V_3}{2n(n-1)} \quad (9)$$

If the X variable has no tied detected concentrations and no non-detects, then $U_1 = U_2 = U_3 = 0$. Similarly, if the Y variable has no tied detected concentrations and no non-detects, then $V_1 = V_2 = V_3 = 0$.

The test statistic Z has a standard normal distribution and is shown in Equation 10

$$Z = \frac{S - \text{sgn}(S)}{\sqrt{\text{Var}(S)}} \quad (10)$$

where $\text{sgn}(S)$ is the sign function that equals 1 if $S > 0$ and -1 if $S < 0$. The sign function is a continuity correction factor that is necessary for the standard normal distribution to better approximate the discrete distribution of S. The two-sided p -value $P(|Z| \geq z)$ can then be calculated using the standard normal table found in any statistics textbook or statistical software package.

KENDALL'S TAU-B CALCULATION EXAMPLE

Suppose the project has collected $n = 12$ discrete environmental surface soil samples from a site suspected to be contaminated with lead and arsenic. The project wants to see if concentrations of lead and arsenic are correlated. These 12 samples were collected from a statistical sampling design and are representative of the site for characterization. The 12 arsenic concentrations (mg/kg) are first sorted from smallest to largest and are shown in Table 1. The corresponding lead concentrations (mg/kg) associated with these samples are also shown where "<" denotes a non-detect or left censored concentration. This data set is typical for environmental data as it has several distinct censoring levels or reporting limits and detected concentrations tied with non-detects.

In addition to the original concentrations and the sampling locations, Table 1 also shows the number of concordant pairs (N_c), the number of discordant pairs (N_d), the number of tied arsenic concentrations and the number of tied lead concentrations.

The calculation of N_c , N_d and ties in arsenic and lead goes as follows. The pair of concentrations with the lowest arsenic concentration is <1 mg/kg along with its lead concentration of 1 mg/kg. The <1 arsenic concentration is lower than the next higher arsenic concentration of 2 mg/kg, but the detected 1 mg/kg lead concentration is tied with the <10 concentration in SS02. Therefore, this first comparison of pairs of concentrations is neither a concordant (both concentrations increase or decrease) or discordant (one variable increases while the other decreases). This first pair from SS01 is then compared to each subsequent pair from SS03 through SS12. Because either the arsenic or lead concentration is tied, there are no concordant pairs and no discordant pairs. Thus, $N_c = 0$ and $N_d = 0$. For the SS01 paired concentrations there are 6 ties in arsenic: <1 with <3 twice and <1 with <10 four times. There are 7 ties in lead: 1 with <10, 1 with <5, 1 with <8, 1 with <3, 1 with <7, 1 with <10 and 1 with <15. All comparisons with the first pair of concentrations in SS01 are now complete. SS02 is then compared to every other location SS03 through SS12 in a similar manner. Note there are no concordant pairs, while there are 3 discordant pairs: (<10, 20) in SS09 with (20, <15) in SS12, (<10, 20) in SS10 with (20, <15) in SS12 and (<10, 30) in SS11 with (20, <15) in SS12.

Table 1 shows that the sum of the N_c column is $N_c = 0$, $N_d = 3$, ties in the arsenic concentrations $T_x = 40$ and ties in the lead concentrations $T_y = 30$.

Table 1. Calculations Used for Kendall's Tau-B Example

Sampling Location	Arsenic (mg/kg)	Lead (mg/kg)	N_c	N_d	Ties in Arsenic	Ties in Lead
SS01	<1	1				
SS02	2	<10	0	0	6	7
SS03	<3	<5	0	0	6	6
SS04	<3	<8	0	0	5	5
SS05	3	<3	0	0	4	4
SS06	5	<7	0	0	4	3
SS07	5	<10	0	0	5	2
SS08	<10	10	0	0	4	1
SS09	<10	20	0	0	3	1
SS10	<10	20	0	1	2	1
SS11	<10	30	0	1	1	0
SS12	20	<15	0	1	0	0
Totals			0	3	40	30

Table 2 shows the calculated values of u as the number of tied concentrations in the X variable (arsenic) at each detected concentration or all nondetects (NDs). All NDs are considered tied regardless of their censoring level. The quantities $u(u-1)(2u+5)$, $u(u-1)(u-2)$ and $u(u-1)$ are then calculated. These quantities are then summed as shown in Equations 3, 5, and 7 to determine U_1 , U_2 and U_3 , respectively. Table 2 shows $U_1 = 816$, $U_2 = 210$ and $U_3 = 44$.

Table 2. Calculations of U_1 , U_2 and U_3 for the X Variable Arsenic

Arsenic (mg/kg)	u	$u(u-1)(2u+5)$	$u(u-1)(u-2)$	$u(u-1)$
2	1	0	0	0
3	1	0	0	0
5	2	18	0	2
20	1	0	0	0
NDs	7	798	210	42
Totals		816	210	44

Similarly, Table 3 shows the calculated values of v as the number of tied concentrations in the Y variable (lead) at each detected concentration or all nondetects (NDs). The quantities $v(v-1)(2v+5)$, $v(v-1)(v-2)$ and $v(v-1)$ are then calculated. These quantities are then summed as shown in Equations 4, 6, and 8 to determine V_1 , V_2 and V_3 , respectively. Table 3 shows $V_1 = 816$, $V_2 = 210$ and $V_3 = 44$. The U_1 and V_1 , U_2 and V_2 , and U_3 and V_3 variables happen to be equal in this example. However, they do not have to be.

Table 3. Calculations of V_1 , V_2 and V_3 for the Y Variable Lead

Lead (mg/kg)	v	$v(v-1)(2v+5)$	$v(v-1)(v-2)$	$v(v-1)$
1	1	0	0	0
10	1	0	0	0
20	2	18	0	2
30	1	0	0	0
NDs	7	798	210	42
Totals		816	210	44

Kendall's tau-b is then calculated in Equation 11 using Equation 1 and the information calculated in Tables 1, 2, and 3.

$$\tau_b = \frac{N_c - N_d}{\sqrt{\left(\frac{n(n-1)}{2} - T_x\right)\left(\frac{n(n-1)}{2} - T_y\right)}} = \frac{0 - 3}{\sqrt{\left(\frac{12(12-1)}{2} - 40\right)\left(\frac{12(12-1)}{2} - 30\right)}} = -0.0981 \quad (11)$$

S is calculated using the information in Table 1 with Equation 2.

$$S = N_c - N_d = 0 - 3 = -3 \quad (12)$$

The variance of S is calculated using Tables 1, 2, and 3 and Equation 9.

$$\text{Var}(S) = \frac{12(12-1)(2(12)+5)-816-816}{18} + \frac{210(210)}{9(12)(12-1)(12-2)} + \frac{44(44)}{2(12)(12-1)} = 133.045 \quad (13)$$

The test statistic Z is calculated using Equation 10.

$$Z = \frac{S - \text{sgn}(S)}{\sqrt{\text{Var}(S)}} = \frac{-3 - (-1)}{\sqrt{133.045}} \approx -0.1734 \quad (14)$$

The two-sided p -value $P(|Z| < -0.1734) = 0.8623$ from the standard normal distribution. Because this p -value $0.8623 >$ the selected significance level 0.05 , we fail to reject the null hypothesis of zero correlation between arsenic and lead. There is not enough evidence to support the alternative hypothesis of nonzero correlation between arsenic and lead at the 0.05 significance level.

CALCULATION OF KENDALL'S TAU-B IN SAS

If we use the KENDALL option within PROC CORR to calculate Kendall's tau-b within SAS ignoring whether the data are left censored or not, the correlation would be 0.6279 with a two-sided p -value of 0.0081 . The SAS code using PROC CORR is below.

```
data a;
  input x y;
  datalines;
1 1
2 10
3 5
3 8
3 3
5 7
5 10
10 10
10 20
10 20
10 30
20 15
;

proc corr data=a kendall; var x y; run;
```

Therefore, by ignoring the censoring status of each result, the correlation of 0.6279 would be significantly different from zero. This contradicts the Kendall's tau-b calculation in Equation 11 of -0.0981 , which is not significantly different from zero. Note even the signs are different.

If we use the KENDALL option within PROC CORR within SAS by replacing all data (detects and non-detects) at or below the highest detection limit with the highest detection limit, the correlation would be -0.168 with a two-sided p -value of 0.5673 . The SAS code using PROC CORR is below.

```

data b;
  input x y;
  datalines;
10 15
10 15
10 15
10 15
10 15
10 15
10 15
10 15
10 15
10 20
10 20
10 30
20 15
;

proc corr data=b kendall; var x y; run;

```

Therefore, treating all data (detects and non-detects) as tied at or below the highest detection limit discards important information that should be used in calculating the correlation and resulting p -value. This method comes closer to the calculated correlation of -0.0981 compared to the 0.6279 correlation where censoring is ignored.

A third way to input the data into SAS is to rank the data assuming only distinct censoring levels are tied. For example, <3 and <3 would be given tied ranks below <10, <10, <10 and <10. Detected concentrations would be given distinct ranks between censoring levels. Using this approach the Kendall's tau-b correlation using PROC CORR is 0.5745 with a two-sided p -value of 0.0134. The SAS code follows.

```

data c;
  input x y;
  datalines;
1 1
2 6.5
3.5 3
3.5 5
5 2
6.5 4
6.5 6.5
9.5 9
9.5 10.5
9.5 10.5
9.5 12
12 8
;

proc corr data=c kendall; var x y; run;

```

These three approaches of attempting to incorporate the chemical data into a single number or rank for PROC CORR fail to correctly calculate the Kendall's tau-b correlation when non-detects are present and detected concentrations are reported between multiple censoring levels. Therefore, SAS code was developed to properly account for non-detects and detected concentrations.

SAS CODE TO CALCULATE KENDALL'S TAU-B FOR LEFT CENSORED DATA

The following SAS code will calculate the Kendall's tau-b correlation for left censored chemical data.

```

%macro obsnvars(ds);
  /* This macro returns the number of variables and observations from a data set;
  %global dset nvars nobs;
  %let dset=&ds;
  %let dsid = %sysfunc(open(&dset));
  %if &dsid %then %do;
    %let nobs =%sysfunc(attrn(&dsid,NOBS));
    %let nvars=%sysfunc(attrn(&dsid,NVARS));
    %let rc = %sysfunc(close(&dsid));
  %end;
%end;

```

```

    %end;
    %else
        %put Open for data set &dset failed - %sysfunc(sysmsg());
    %mend obsnvars;

data getdata;
    input XVARHIT XVAR YVARHIT YVAR;
    XVARCHEM = 'Lead';
    YVARCHEM = 'Arsenic';
    call symput('XVAR', trim(left(xvarchem)));
    call symput('YVAR', trim(left(yvarchem)));
    attrib d_collected format=date9. informat=date9.;
    datalines;
0      1      1      1
1      2      0     10
0      3      0      8
0      3      0      5
1      3      0      3
1      5      0     10
1      5      0      7
0     10      1     10
0     10      1     30
0     10      1     20
0     10      1     20
1     20      0     15
;

%macro corrs;

data trendit;
    set getdata;
    call symput('XVAR', trim(left(XVAR)));
    call symput('YVAR', trim(left(YVAR)));
    run;

proc summary data=trendit; var xvarhit yvarhit; output out=nout(drop=_type_ _freq_)
    n=n sum=HITSUMX HITSUMY; run;
run;

data nout2;
    set nout;
    NDX = n - hitsumx;  ** Number of non-detects in X;
    NDY = n - hitsumy;  ** Number of non-detects in Y;
    call symput('NDX', trim(left(put(ndx, 11.0))));
    call symput('NDY', trim(left(put(ndy, 11.0))));
    call symput('HITSUMX', trim(left(put(hitsumx, 11.0))));
    call symput('HITSUMY', trim(left(put(hitsumy, 11.0))));
    run;

    %obsnvars(trendit); %let MAXN = &nobs;

    proc sort data=trendit; by xvar xvarhit yvar yvarhit; run;
    ** VERY important to run here!;

proc transpose data=trendit out=xt    prefix=XRES; var xvar;    run;
proc transpose data=trendit out=xhitt prefix=XHIT; var xvarhit; run;

proc transpose data=trendit out=yt    prefix=YRES; var yvar;    run;
proc transpose data=trendit out=yhitt prefix=YHIT; var yvarhit; run;

    ** Find number of ties among DETECTS only at each concentration level for X
    variable;

%if &HITSUMX > 0 %then %do;
    proc sort data=trendit; by xvar; run;    ** VERY important to run here!;
proc summary data=trendit;

```

```

    where xvarhit=1;
    var xvarhit;
    by xvar;
    output out=outxvardet1(drop=_type_ _freq_) n=u;
run;
%end;

%if &NDX > 0 %then %do;
    ** Find number of distinct NON-DETECTS only!;
proc summary data=trendit;
    where xvarhit=0;
    var xvarhit;
    output out=outndx1(drop=_type_ _freq_) n=u;
run;
%end;

data calc_sum_u;    ** Calculate sum of  $u(u-1)(2u+5)$ ,  $u(u-1)(u-2)$  and  $u(u-1)$  for
    Var(S) term;
    %if &HITSUMX > 0 and &NDX > 0 %then %do;
        set outndx1 outxvardet1;
    %end;
    %else %if &HITSUMX = 0 and &NDX > 0 %then %do;
        set outndx1;
    %end;
    %else %if &HITSUMX > 0 and &NDX = 0 %then %do;
        set outxvardet1;
    %end;
    TIECALC3U = u*(u - 1)*(2*u + 5);
    TIECALC3Ub = u*(u - 1)*(u - 2);
    TIECALC2U = u*(u - 1);
run;

proc summary data=calc_sum_u;
    var tiecalc3u tiecalc3ub tiecalc2u;
    output out=tiecalcsumx(drop=_type_ _freq_) sum=TIECALC3U TIECALC3Ub TIECALC2U;
run;

    ** Find number of ties among DETECTS only at each concentration level for Y
    variable;
%if &HITSUMY > 0 %then %do;
proc sort data=trendit; by yvar;
proc summary data=trendit;
    where yvarhit=1;
    var yvarhit;
    by yvar;
    output out=outyvardet1(drop=_type_ _freq_) n=v;
run;
%end;

%if &NDY > 0 %then %do;
    ** Find number of distinct NON-DETECTS only!;
proc summary data=trendit;
    where yvarhit=0;
    var yvarhit;
    output out=outndy1(drop=_type_ _freq_) n=v;
run;
%end;

data calc_sum_v;    ** Calculate sum of  $v(v-1)(2v+5)$ ,  $v(v-1)(v-2)$  and  $v(v-1)$  for
    Var(S) term;
    %if &HITSUMY > 0 and &NDY > 0 %then %do;
        set outndy1 outyvardet1;
    %end;
    %else %if &HITSUMY = 0 and &NDY > 0 %then %do;
        set outndy1;
    %end;
    %else %if &HITSUMY > 0 and &NDY = 0 %then %do;

```

```

    set outyvardet1;
    %end;
    TIECALC3V = v*(v - 1)*(2*v + 5);
    TIECALC3Vb = v*(v - 1)*(v - 2);
    TIECALC2V = v*(v - 1);
run;

proc summary data=calc_sum_v;
    var tiecalc3v tiecalc3vb tiecalc2v;
    output out=tiecalcsumy(drop=_type_ _freq_) sum=TIECALC3V TIECALC3Vb TIECALC2V;
run;

data precalckendalltb;
    set yt;
    if _N_=1 then set yhitt;
    if _N_=1 then set xt;
    if _N_=1 then set xhitt;
    if _N_=1 then set nout;
    if _N_=1 then set tiecalcsumx;
    if _N_=1 then set tiecalcsumy;
    XVAR = "&XVAR";
    YVAR = "&YVAR";
run;

data calckendalltb;
    set precalckendalltb;
    array xres(&MAXN.) xres1-xres&MAXN.;
    array xhit(&MAXN.) xhit1-xhit&MAXN.;
    array yres(&MAXN.) yres1-yres&MAXN.;
    array yhit(&MAXN.) yhit1-yhit&MAXN.;
    sum_s = 0;
    num_tiesx = 0;
    num_tiesy = 0;
    do i = 1 to &MAXN - 1;
        do j = i + 1 to &MAXN;
            ** Concordant pair is when both X and Y increase or
            both decrease. A discordant pair is when X and Y go in opposite directions;
            ** i and j are detected in both X and Y;
            if xhit(i) = 1 and xhit(j) = 1 and xres(i) ^=. and xres(j) ^=. and yhit(i) =
            1 and yhit(j) = 1 and yres(i) ^=. and yres(j) ^=. then do;
                ** Both detects
                for X and Y variables;
                if round(xres(j), 0.000001) < round(xres(i), 0.000001) and
                round(yres(j), 0.000001) < round(yres(i), 0.000001) then s = 1;
                ** Both X and
                Y decrease;
                else if round(xres(j), 0.000001) > round(xres(i), 0.000001) and
                round(yres(j), 0.000001) > round(yres(i), 0.000001) then s = 1;
                ** Both X and
                Y increase;
                else if round(xres(j), 0.000001) > round(xres(i), 0.000001) and
                round(yres(j), 0.000001) < round(yres(i), 0.000001) then s = -1;
                ** X
                increases and Y decreases;
                else if round(xres(j), 0.000001) < round(xres(i), 0.000001) and
                round(yres(j), 0.000001) > round(yres(i), 0.000001) then s = -1;
                ** X
                decreases and Y increases;
                else if round(xres(i), 0.000001) = round(xres(j), 0.000001) and
                round(yres(i), 0.000001) ^= round(yres(j), 0.000001) then do; s = 0; num_tiesx
                + 1; end;
                ** X is tied, but Y is not;
                else if round(xres(i), 0.000001) ^= round(xres(j), 0.000001) and
                round(yres(i), 0.000001) = round(yres(j), 0.000001) then do; s = 0; num_tiesy
                + 1; end;
                ** X is not tied, but Y is tied;
                else if round(xres(i), 0.000001) = round(xres(j), 0.000001) and
                round(yres(i), 0.000001) = round(yres(j), 0.000001) then do; s = 0; num_tiesx
                + 1; num_tiesy + 1; end;
                ** X and Y are both tied;
            end;
            ** X i and j are detected, Y i detected, Y j not detected;
            else if xhit(i) = 1 and xhit(j) = 1 and xres(i) ^=. and xres(j) ^=. and
            yhit(i) = 1 and yhit(j) = 0 and yres(i) ^=. and yres(j) ^=. then do;

```

```

        if round(xres(j), 0.000001) < round(xres(i), 0.000001) and
round(yres(j), 0.000001) =< round(yres(i), 0.000001) then s = 1;    ** both X
and Y decrease;
        else if round(xres(j), 0.000001) > round(xres(i), 0.000001) and
round(yres(j), 0.000001) =< round(yres(i), 0.000001) then s = -1;    ** X
increases and Y decreases;
        else if round(xres(j), 0.000001) = round(xres(i), 0.000001) and
round(yres(j), 0.000001) > round(yres(i), 0.000001) then do; s = 0; num_tiesx +
1; num_tiesy + 1; end;    ** both X and Y are tied;
        else if round(xres(j), 0.000001) ^= round(xres(i), 0.000001) and
round(yres(j), 0.000001) > round(yres(i), 0.000001) then do; s = 0; num_tiesy +
1; end;    ** Y is tied only;
        else if round(xres(j), 0.000001) = round(xres(i), 0.000001) and
round(yres(j), 0.000001) <= round(yres(i), 0.000001) then do; s = 0; num_tiesx
+ 1; end;    ** X only is tied;
        end;
    ** X i and j are detected, Y i not detected, Y j detected;
    else if xhit(i) = 1 and xhit(j) = 1 and xres(i) ^=. and xres(j) ^=. and
yhit(i) = 0 and yhit(j) = 1 and yres(i) ^=. and yres(j) ^=. then do;
        if round(xres(j), 0.000001) > round(xres(i), 0.000001) and
round(yres(j), 0.000001) >= round(yres(i), 0.000001) then s = 1;    ** Both X
and Y increase;
        else if round(xres(j), 0.000001) < round(xres(i), 0.000001) and
round(yres(j), 0.000001) >= round(yres(i), 0.000001) then s = -1;    ** X
decreases and Y increases;
        else if round(xres(j), 0.000001) = round(xres(i), 0.000001) and
round(yres(j), 0.000001) < round(yres(i), 0.000001) then do; s = 0; num_tiesx +
1; num_tiesy + 1; end;    ** Both X and Y are tied;
        else if round(xres(j), 0.000001) ^= round(xres(i), 0.000001) and
round(yres(j), 0.000001) < round(yres(i), 0.000001) then do; s = 0; num_tiesy +
1; end;    ** Y is tied only;
        else if round(xres(j), 0.000001) = round(xres(i), 0.000001) and
round(yres(j), 0.000001) >= round(yres(i), 0.000001) then do; s = 0; num_tiesx
+ 1; end;    ** X only is tied;
        end;
    ** X i not detected, X j is detected, Y i and Y j detected;
    else if xhit(i) = 0 and xhit(j) = 1 and xres(i) ^=. and xres(j) ^=. and
yhit(i) = 1 and yhit(j) = 1 and yres(i) ^=. and yres(j) ^=. then do;
        if round(xres(j), 0.000001) >= round(xres(i), 0.000001) and
round(yres(j), 0.000001) > round(yres(i), 0.000001) then s = 1;    ** Both X and
Y increase;
        else if round(xres(j), 0.000001) >= round(xres(i), 0.000001) and
round(yres(j), 0.000001) < round(yres(i), 0.000001) then s = -1;    ** X
increases and Y decreases;
        else if round(xres(j), 0.000001) < round(xres(i), 0.000001) and
round(yres(j), 0.000001) = round(yres(i), 0.000001) then do; s = 0; num_tiesx +
1; num_tiesy + 1; end;    ** Both X and Y are tied;
        else if round(xres(j), 0.000001) >= round(xres(i), 0.000001) and
round(yres(j), 0.000001) = round(yres(i), 0.000001) then do; s = 0; num_tiesy +
1; end;    ** Y is tied only;
        else if round(xres(j), 0.000001) < round(xres(i), 0.000001) and
round(yres(j), 0.000001) ^= round(yres(i), 0.000001) then do; s = 0; num_tiesx
+ 1; end;    ** X only is tied;
        end;
    ** X i detected, X j is not detected, Y i and Y j detected;
    else if xhit(i) = 1 and xhit(j) = 0 and xres(i) ^=. and xres(j) ^=. and
yhit(i) = 1 and yhit(j) = 1 and yres(i) ^=. and yres(j) ^=. then do;
        if round(xres(j), 0.000001) <= round(xres(i), 0.000001) and
round(yres(j), 0.000001) < round(yres(i), 0.000001) then s = 1;    ** Both X and
Y decrease;
        else if round(xres(j), 0.000001) <= round(xres(i), 0.000001) and
round(yres(j), 0.000001) > round(yres(i), 0.000001) then s = -1;    ** X
decreases and Y increases;
        else if round(xres(j), 0.000001) > round(xres(i), 0.000001) and
round(yres(j), 0.000001) = round(yres(i), 0.000001) then do; s = 0; num_tiesx +
1; num_tiesy + 1; end;    ** Both X and Y are tied;

```

```

        else if round(xres(j), 0.000001) <= round(xres(i), 0.000001) and
round(yres(j), 0.000001) = round(yres(i), 0.000001) then do; s = 0; num_tiesy +
1; end;    ** Y is tied only;
        else if round(xres(j), 0.000001) > round(xres(i), 0.000001) and
round(yres(j), 0.000001) ^= round(yres(i), 0.000001) then do; s = 0; num_tiesx
+ 1; end;    ** X only is tied;
        end;
        ** X i and X j are not detected, Y i and Y j detected;
        else if xhit(i) = 0 and xhit(j) = 0 and xres(i) ^=. and xres(j) ^=. and
yhit(i) = 1 and yhit(j) = 1 and yres(i) ^=. and yres(j) ^=. then do;
        if round(yres(j), 0.000001) = round(yres(i), 0.000001) then do; s
= 0; num_tiesx + 1; num_tiesy + 1; end;    ** both X and Y are tied;
        else if round(yres(j), 0.000001) ^= round(yres(i), 0.000001) then do; s
= 0; num_tiesx + 1; end;    ** X is tied only;
        end;
        ** X i and X j are detected, Y i and Y j are not detected;
        else if xhit(i) = 1 and xhit(j) = 1 and xres(i) ^=. and xres(j) ^=. and
yhit(i) = 0 and yhit(j) = 0 and yres(i) ^=. and yres(j) ^=. then do;
        if round(xres(j), 0.000001) = round(xres(i), 0.000001) then do; s
= 0; num_tiesx + 1; num_tiesy + 1; end;    ** both X and Y are tied;
        else if round(xres(j), 0.000001) ^= round(xres(i), 0.000001) then do; s
= 0; num_tiesy + 1; end;    ** Y is tied only;
        end;
        ** X i and Y i are not detected, X j and Y j are detected;
        else if xhit(i) = 0 and xhit(j) = 1 and xres(i) ^=. and xres(j) ^=. and
yhit(i) = 0 and yhit(j) = 1 and yres(i) ^=. and yres(j) ^=. then do;
        if round(xres(j), 0.000001) >= round(xres(i), 0.000001) and
round(yres(j), 0.000001) >= round(yres(i), 0.000001) then s = 1;    ** Both X
and Y increase;
        else if round(xres(j), 0.000001) < round(xres(i), 0.000001) and
round(yres(j), 0.000001) < round(yres(i), 0.000001) then do; s = 0; num_tiesx +
1; num_tiesy + 1; end;    ** Both X and Y are tied;
        else if round(xres(j), 0.000001) >= round(xres(i), 0.000001) and
round(yres(j), 0.000001) < round(yres(i), 0.000001) then do; s = 0; num_tiesy +
1; end;    ** Y is tied only;
        else if round(xres(j), 0.000001) < round(xres(i), 0.000001) and
round(yres(j), 0.000001) >= round(yres(i), 0.000001) then do; s = 0; num_tiesx
+ 1; end;    ** X only is tied;
        end;
        ** X i and Y i are detected, X j and Y j are not detected;
        else if xhit(i) = 1 and xhit(j) = 0 and xres(i) ^=. and xres(j) ^=. and
yhit(i) = 1 and yhit(j) = 0 and yres(i) ^=. and yres(j) ^=. then do;
        if round(xres(j), 0.000001) <= round(xres(i), 0.000001) and
round(yres(j), 0.000001) <= round(yres(i), 0.000001) then s = 1;    ** Both X
and Y decrease;
        else if round(xres(j), 0.000001) > round(xres(i), 0.000001) and
round(yres(j), 0.000001) > round(yres(i), 0.000001) then do; s = 0; num_tiesx +
1; num_tiesy + 1; end;    ** Both X and Y are tied;
        else if round(xres(j), 0.000001) <= round(xres(i), 0.000001) and
round(yres(j), 0.000001) > round(yres(i), 0.000001) then do; s = 0; num_tiesy +
1; end;    ** Y is tied only;
        else if round(xres(j), 0.000001) > round(xres(i), 0.000001) and
round(yres(j), 0.000001) <= round(yres(i), 0.000001) then do; s = 0; num_tiesx
+ 1; end;    ** X only is tied;
        end;
        ** X i and Y j are detected, X j and Y i are not detected;
        else if xhit(i) = 1 and xhit(j) = 0 and xres(i) ^=. and xres(j) ^=. and
yhit(i) = 0 and yhit(j) = 1 and yres(i) ^=. and yres(j) ^=. then do;
        if round(xres(j), 0.000001) <= round(xres(i), 0.000001) and
round(yres(j), 0.000001) >= round(yres(i), 0.000001) then s = -1;    ** X
decreases and Y increases;
        else if round(xres(j), 0.000001) > round(xres(i), 0.000001) and
round(yres(j), 0.000001) < round(yres(i), 0.000001) then do; s = 0; num_tiesx +
1; num_tiesy + 1; end;    ** Both X and Y are tied;
        else if round(xres(j), 0.000001) <= round(xres(i), 0.000001) and
round(yres(j), 0.000001) < round(yres(i), 0.000001) then do; s = 0; num_tiesy +
1; end;    ** Y is tied only;

```

```

        else if round(xres(j), 0.000001) > round(xres(i), 0.000001) and
round(yres(j), 0.000001) >= round(yres(i), 0.000001) then do; s = 0; num_tiesx
+ 1; end;    ** X only is tied;
        end;
        ** X i and Y j are not detected, X j and Y i are detected;
        else if xhit(i) = 0 and xhit(j) = 1 and xres(i) ^=. and xres(j) ^=. and
yhit(i) = 1 and yhit(j) = 0 and yres(i) ^=. and yres(j) ^=. then do;
        if round(xres(j), 0.000001) >= round(xres(i), 0.000001) and
round(yres(j), 0.000001) <= round(yres(i), 0.000001) then s = -1;    ** X
increases and Y decreases;
        else if round(xres(j), 0.000001) < round(xres(i), 0.000001) and
round(yres(j), 0.000001) > round(yres(i), 0.000001) then do; s = 0; num_tiesx +
1; num_tiesy + 1; end;    ** Both X and Y are tied;
        else if round(xres(j), 0.000001) >= round(xres(i), 0.000001) and
round(yres(j), 0.000001) > round(yres(i), 0.000001) then do; s = 0; num_tiesy +
1; end;    ** Y is tied only;
        else if round(xres(j), 0.000001) < round(xres(i), 0.000001) and
round(yres(j), 0.000001) <= round(yres(i), 0.000001) then do; s = 0; num_tiesx
+ 1; end;    ** X only is tied;
        end;
        ** X i is detected, X j, Y i and Y j are not detected;
        else if xhit(i) = 1 and xhit(j) = 0 and xres(i) ^=. and xres(j) ^=. and
yhit(i) = 0 and yhit(j) = 0 and yres(i) ^=. and yres(j) ^=. then do;
        if round(xres(j), 0.000001) > round(xres(i), 0.000001) then do; s
= 0; num_tiesx + 1; num_tiesy + 1; end;    ** both X and Y are tied;
        else if round(xres(j), 0.000001) <= round(xres(i), 0.000001) then do; s
= 0; num_tiesy + 1; end;    ** Y is tied only;
        end;
        ** Y i is detected, Y j, X i and X j are not detected;
        else if xhit(i) = 0 and xhit(j) = 0 and xres(i) ^=. and xres(j) ^=. and
yhit(i) = 1 and yhit(j) = 0 and yres(i) ^=. and yres(j) ^=. then do;
        if round(yres(j), 0.000001) > round(yres(i), 0.000001) then do; s
= 0; num_tiesx + 1; num_tiesy + 1; end;    ** both X and Y are tied;
        else if round(yres(j), 0.000001) <= round(yres(i), 0.000001) then do; s
= 0; num_tiesx + 1; end;    ** X is tied only;
        end;
        ** X j is detected, X i, Y i and Y j are not detected;
        else if xhit(i) = 0 and xhit(j) = 1 and xres(i) ^=. and xres(j) ^=. and
yhit(i) = 0 and yhit(j) = 0 and yres(i) ^=. and yres(j) ^=. then do;
        if round(xres(j), 0.000001) < round(xres(i), 0.000001) then do; s
= 0; num_tiesx + 1; num_tiesy + 1; end;    ** both X and Y are tied;
        else if round(xres(j), 0.000001) >= round(xres(i), 0.000001) then do; s
= 0; num_tiesy + 1; end;    ** Y is tied only;
        end;
        ** Y j is detected, X i, Y i and X j are not detected;
        else if xhit(i) = 0 and xhit(j) = 0 and xres(i) ^=. and xres(j) ^=. and
yhit(i) = 0 and yhit(j) = 1 and yres(i) ^=. and yres(j) ^=. then do;
        if round(yres(j), 0.000001) < round(yres(i), 0.000001) then do; s
= 0; num_tiesx + 1; num_tiesy + 1; end;    ** both X and Y are tied;
        else if round(yres(j), 0.000001) >= round(yres(i), 0.000001) then do; s
= 0; num_tiesx + 1; end;    ** X is tied only;
        end;
        ** X i, X j, Y i, Y j are not detected ;
        else if xhit(i) = 0 and xhit(j) = 0 and xres(i) ^=. and xres(j) ^=. and
yhit(i) = 0 and yhit(j) = 0 and yres(i) ^=. and yres(j) ^=. then do;
        s = 0; num_tiesx + 1; num_tiesy + 1;    ** Both X and Y are tied;
        end;
        else if xres(i) = . or xres(j) = . or yres(i) = . or yres(j) = . then s = 0;
        sum_s + s;
        end;
    end;
if n*(n-1)/2 > num_tiesx and n*(n-1)/2 > num_tiesy then KENDALL_TAUB = sum_s /
sqrt((n*(n-1)/2 - num_tiesx)*(n*(n-1)/2 - num_tiesy));    ** Eqn. 11.5 in
Helsel p. 190 "Nondetects and Data Analysis" book;
else KENDALL_TAUB = 0;

```

```

VAR_S = (n*(n-1)*(2*n+5) - tiecalc3u - tiecalc3v)/18 + tiecalc3ub * tiecalc3vb /
(9*n*(n-1)*(n-2)) + tiecalc2u * tiecalc2v / (2*n*(n-1));    ** see p. 66 Eqn.
4.4 of "Rank Correlation Methods" book;
if      sum_s > 0 and var_s > 0 then Z = (sum_s - 1) / sqrt(var_s);
else if sum_s < 0 and var_s > 0 then Z = (sum_s + 1) / sqrt(var_s);
else Z = 0;
if z <= 0 then TAU_PVALUE = 2*probnorm(z);
else if z > 0 then tau_pvalue = 2*(1 - probnorm(z));
if tau_pvalue <= 0.05 then KENDALL_TAUB_SIG = 'Yes';
else if tau_pvalue > 0.05 then KENDALL_TAUB_SIG = 'No';
drop _name_ i j s;
%mend corrs;

```

CONCLUSION

Kendall's tau-b correlation is a nonparametric method for calculating monotonic correlation between two variables with multiple censoring levels. While SAS does calculate Kendall's tau-b as an option in the PROC CORR procedure within SAS/BASE, its calculation assumes all values are known and uncensored. It does not incorporate censoring that is common with chemical or environmental data. SAS code was written to properly calculate the Kendall's tau-b correlation with multiply censored data. These results were compared with the output from PROC CORR, which cannot adequately account for censoring in its calculation of Kendall's tau-b. The SAS code provided in this paper makes minimal underlying assumptions about the data, can be used with multiple reporting limits, and efficiently uses the censoring information from the data. Ties are easily incorporated among detected concentrations and non-detects of differing censoring levels.

REFERENCES

- Beal, Dennis J. 2010. "A Macro for Calculating Summary Statistics on Left Censored Environmental Data using the Kaplan-Meier Method," *Proceedings of the 18th Annual Conference of the SouthEast SAS Users Group*.
- Beal, Dennis J. 2015. "A Macro for Calculating Percentiles on Left Censored Environmental Data using the Kaplan-Meier Method," *Proceedings of the 23rd Annual Conference of the SouthEast SAS Users Group*.
- Helsel, Dennis R. 2005. *Nondetects and Data Analysis: Statistics for Censored Environmental Data*. New Jersey: John Wiley & Sons, Inc.
- Helsel, Dennis R. 2012. *Statistics for Censored Environmental Data Using Minitab and R*. New Jersey: John Wiley & Sons, Inc. (Second Edition).
- Kendall, Maurice and Jean Gibbons. 1990. *Rank Correlation Methods*. New York: Oxford University Press (Fifth Edition).

CONTACT INFORMATION

The author welcomes and encourages any questions, corrections, feedback, and remarks. Contact the author at:

Dennis J. Beal, Ph.D.
 Senior Statistician/Risk Scientist
 Leidos
 301 Laboratory Road
 P.O. Box 2502
 Oak Ridge, Tennessee 37830

phone: 865-481-8736
 e-mail: beald@leidos.com

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. Other brand and product names are registered trademarks or trademarks of their respective companies.