1996-01

Computing min and max scorings for two-sample ordinal data

Whitaker, Lyn R.; Whitaker, Michael D.

Monterey, California. Naval Postgraduate School

NPS-OR-96-002

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Computing Min and Max Scorings for Two-Sample Ordinal Data

by

Lyn R. Whitaker
Michael D. Whitaker

January 1996

Approved for public release; distribution is unlimited.

Prepared for: Institute of Joint Warfare Analysis (IJWA)
Monterey, CA 93943

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5000

Rear Admiral M. J. Evans
Superintendent

Richard Elster
Provost

This report was prepared for and funded by the Institute of Joint Warfare Analysis, Monterey, CA.

This report was prepared by:

LYN R. WHITAKER
Associate Professor of
Operations Research

MICHAEL D. WHITAKER
Scientific Programmer
McGraw-Hill, Monterey, CA

Reviewed by:

FRANK PETHO
Acting Chairman
Department of Operations Research

Released by:

GORDON E. SCHACHER
Dean of Research (Acting)

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1996	Technical

4. TITLE AND SUBTITLE
Computing Min and Max Scorings for Two-Sample Ordinal Data

5. FUNDING NUMBERS

6. AUTHOR(S)
Lyn R. Whitaker and Michael D. Whitaker

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER
NPS-OR-96-002

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Institute of Joint Warfare Analysis (IJWA)
Monterey, CA 93943

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Ordinal response variables often occur in practice. For example, in clinical trials a subject's response to a drug regime might be categorized as negative, none, fair, or good. There are several common approaches to analyzing two-sample ordinal response data. These procedures applied to the same data can lead to contradictory conclusions. In an attempt to reconcile contradictory results and provide guidance to the practitioner, Kimledorf, Sampson and Whitaker (1992) propose an alternative approach. They find the scores which when assigned to the levels of the ordinal response variable maximize a two-sample test statistic and the scores that minimize that same statistic. Since many of the two-sample statistics are related by monotonic transformations, these extreme scores are in fact extreme scores for several test statistics. Both minimized and maximized test statistics falling into the rejection region clearly indicate a difference between the two populations or treatments. On the other hand if neither of the two extreme statistics fall in the rejection region then no matter what scores are used there will be no significant difference in the two populations. In this paper we review the KSW procedure and its implementation in SAS® software.

14. SUBJECT TERMS	15. NUMBER OF PAGES
SAS software, Ordinal Data, Two-sample	18
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Computing Min and Max Scorings for Two-Sample Ordinal Data

Lyn R. Whitaker, Naval Postgraduate School, Monterey, CA
Michael D. Whitaker, CTB/McGraw-Hill, Monterey, CA

ABSTRACT

Ordinal response variables often occur in practice. For example, in clinical trials a subject's response to a drug regime might be categorized as negative, none, fair, or good. There are several common approaches to analyzing two-sample ordinal response data. These procedures applied to the same data can lead to contradictory conclusions. In an attempt to reconcile contradictory results and provide guidance to the practitioner, Kimledorf, Sampson and Whitaker (1992) propose an alternative approach. They find the scores which when assigned to the levels of the ordinal response variable maximize a two-sample test statistic and the scores that minimize that same statistic. Since many of the two-sample statistics are related by monotonic transformations, these extreme scores are in fact extreme scores for several test statistics. Both minimized and maximized test statistics falling into the rejection region clearly indicate a difference between the two populations or treatments. On the other hand if neither of the two extreme statistics fall in the rejection region then no matter what scores are used there will be no significant difference in the two populations. In this paper we review the KSW procedure and its implementation in SAS® software.

1. INTRODUCTION

Ordinal response variables often occur in practice. For example, in clinical trials a subject's response to a drug regime might be categorized as negative, none, fair, or good. There are several common approaches to analyzing two-sample ordinal response data. Among them are assigning arbitrary scores to the levels of the ordinal variable and then using a t-test, nonparametric approaches such as Wilcoxen-Mann-Whitney test and the

Cochran-Mantel-Haensel test (Mantel (1963)) and the generalized linear model approach with ordinal response variables (McCullagh and Nelder (1983)). It is common for practitioners to try several of these tests and then, when results are contradictory, wonder which to use. Kimledorf, Sampson and Whitaker (1992) propose an alternative approach. They find the scores which when assigned to the levels of the ordinal response variable maximize a two-sample test statistic and the scores that minimize that same statistic. Since many of the two-sample statistics are related by monotonic transformations, these extreme scores can in fact be used to find extreme test statistics for several different two-sample tests.

Let $x_1 \leq x_2 \leq \ldots \leq x_k$ $(x_1 \neq x_k)$ be the nondecreasing scores assigned to the levels of an ordinal response variable. The KSW procedure encompasses several of the common methods. The Wilcoxen-Mann-Whitney statistic is a special case of the two-sample t-statistic with marginal midrank scores assigned to the $x_1, \ldots, x_k$ (e.g., Conover and Iman (1981)). The Cochran-Mantel-Haensel (CMH) statistic is usually calculated using uniform or equal spacing scores for the $x_1, \ldots, x_k$, marginal mid-rank scores (ridits), or modified ridit scores. The FREQ procedure allows the choice of these scores for calculating the CMH statistic as well as arbitrary user-provided scores. In addition, both the signed CMH statistic and the two-sample t-statistic are increasing functions of Pearson's correlation coefficient $\rho(x_1, \ldots, x_k)$ between the scores assigned to the ordinal variable and the binary variable indicating whether the response is from Treatment 1 or not.

Thus, by finding the scores $s_1, \ldots, s_k$ which maximize $\rho(x_1, \ldots, x_k)$ and the scores $t_1, \ldots, t_k$ which minimize $\rho(x_1, \ldots, x_k)$ among $x_1 \leq x_2 \leq \ldots \leq x_k$ where $x_1 \neq x_k$, we have also found the maximum and minimum of the two-sample t-statistic and the CMH statistic. If both of the extreme values of the statistic lie in the rejection region then it is clear that no matter how the levels of the ordinal response are scored, the test statistic will be significant. When both of the extreme values of the test statistic fail to lie in the rejection region then the result is also clear, no matter what scores are assigned to the ordinal response variable, the test statistics will always fail to reject the null hypotheses. In the third case, when the scores straddle a critical value, the conclusion becomes more difficult because some non-decreasing scores assigned to the data will result in rejecting the null hypothesis and yet another assignment of scores will result in acceptance of the null hypothesis.

In the next chapter we outline the KSW procedure for finding the minimum and maximum scores and present a SAS macro used to implement this procedure. In Chapter 3 we give a numerical example and in Chapter 4 we provide a conclusion.

2. THE KSW PROCEDURE AND ITS IMPLEMENTATION

The SAS code is a single macro. This macro needs only base SAS software to run and is implemented within a DATA step. The macro uses data in contingency table form, and does all the computations needed to report the minimum and maximum scores and their corresponding t-statistics, CMH statistics, and Pearson's correlations. The complete code is available from the authors.

The two-sample data with scores $x_1 \le x_2 \le \ldots \le x_k$ where $x_1 \ne x_k$ assigned to the levels of the ordinal response variable can be represented as:

TREATMENT	x_1	x_2	...	x_k	TOTAL
0	m_1	m_2	...	m_k	m
1	n_1	n_2	...	n_k	n
TOTAL	$m_1 + n_1$	$m_2 + n_2$	...	$m_k + n_k$	N

Because correlation is scale and location invariant we can, without loss of generality and for ease of use, optimize $\rho(x_1, \ldots, x_k)$ over scores $x_1 = 0 \le x_2 \le \ldots \le x_k = 1$. The notion of stochastic ordering plays an important role in the computations. The empirical distribution of Treatment 1 is said to be stochastically greater than that of Treatment 0 if

$$(n_j + \ldots + n_k)/n \ge (m_j + \ldots + m_k)/m \qquad (2.1)$$

for $j = 2, \ldots, k$. If the inequality (2.1) is reversed then Treatment 0 is said to be stochastically greater than Treatment 1. If neither hold, then the empirical distributions from the two treatments are stochastically incomparable. For simplicity, we compute the scores $s_1, \ldots, s_k$ that maximize and the scores $t_1, \ldots, t_k$ that minimize in three different cases:

Case 1, Treatment 0 data is stochastically greater than Treatment 1 data,
Case 2, Treatment 1 data is stochastically greater than Treatment 0 data,
Case 3, Treatment 0 and Treatment 1 data are stochastically incomparable.

Thus, the first step in computation is to decide in which of the three cases the data fall.

If the data fall into case 1, we find the maximum scores, $s_1, \ldots, s_k$, by first finding the isotonic regression $y_1, \ldots, y_k$ of $n_i/(m_i + n_i)$ with weights $(m_i + n_i)$. There are several algorithms for computing the isotonic regression. In the SAS macro, we use the Pool Adjacent Violators Algorithm (PAVA) (see Robertson, Dykstra and Wright (1988)). The PAVA code is given in the Appendix. The scores $s_1, \ldots, s_k$ are computed by re-scaling the isotonic regression as $s_i = (y_i - y_1)/(y_k - y_1)$. The minimum scores $t_1, \ldots, t_k$ are found by computing $\rho(x_1, \ldots, x_k)$ for the k-2 scores of the form $0 = x_1 = \ldots = x_j$ and

$1 = x_{(j+1)} = \ldots = x_k$ for $j = 2, \ldots, k-1$ and finding the one that gives the smallest $\rho(x_1, \ldots, x_k)$.

If the data fall into case 2 then finding the maximum scores $s_1, \ldots, s_k$ is similar to finding the minimum scores in case 1, i.e. the scores that maximize $\rho(x_1, \ldots, x_k)$ among scores of the form $0 = x_1 = \ldots = x_j$ and $1 = x_{(j+1)} = \ldots = x_k$ for $j = 2, \ldots, k-1$. The minimum score $t_1, \ldots, t_k$ are found as are the maximum score in case 1. Compute the isotonic regression $y_1, \ldots, y_k$ of $m_i/(m_i + n_i)$ with weights $(m_i + n_i)$ and then re-scale to get $t_i = (y_i - y_1)/(y_k - y_1)$ for $i = 1, \ldots, k$.

For case 3, the scores $s_1, \ldots, s_k$ are computed as in case1 and the scores $t_1, \ldots, t_k$ are computed as in case 2. The macro KSW, implementing this procedure is:

```
%*********************************************;
%* Macro :KSW                                  ;
%* Author:Michael Whitaker                     ;
%* Input: n_lev = The number of ordinal levels;
%*        treat0=The freq dist for treatment 0;
%*        treat1=The freq dist for treatment 1;
%* Output:Minscore= Scores that give min r     ;
%*        Maxscore= Scores that give max r     ;
%*        Min_t= Min t-statistic               ;
%*        Max_t= Max t-statistic               ;
%*        Min_r= Min Pearson corr              ;
%*        Max_r= Max Pearson corr              ;
%*        Min_CMH= Min CMH Statistic           ;
%*        Max_CMH= Max CMH Statistic           ;
%* Required Macros : PAV, Stoc_ord, Cov        ;
%* Required Procs  : None                      ;
%* Comments :                                  ;
%*    variables with scores, t, r and CMH      ;
%*  statistics will always be returned         ;
%*  by the macro.                              ;
%*                                             ;
%*********************************************;
%macro ksw(n_lev=,treat0=,treat1=,minscore=_min_scr_,
          maxscore=_max_scr_,min_r=_min_r_,
          max_r=_max_r_,min_t=_min_t_,
          max_t=_max_t_,min_cmh=_min_c_,
          max_cmh=_max_c_);
%*;
%* Define the work arrays;
%*;
  array _w_  {&n_lev} _TEMPORARY_;
  array _t0_ {&n_lev} _TEMPORARY_
    (0 %do j = 1 %to %eval(&n_lev-1); ,0 %end;);
  array _t1_ {&n_lev} _TEMPORARY_
    (0 %do j = 1 %to %eval(&n_lev-1); ,0 %end;);
  array _y0_ {&n_lev} _TEMPORARY_;
  array _y1_ {&n_lev} _TEMPORARY_;
  array _z0_ {%eval(&n_lev-1),&n_lev} _TEMPORARY_;
  array _z1_ {%eval(&n_lev-1),&n_lev} _TEMPORARY_;
  array _r0_ {%eval(&n_lev-1)} _temporary_;
  array _r1_ {%eval(&n_lev-1)} _temporary_;
  array _cmh0_ {%eval(&n_lev-1)} _temporary_;
  array _cmh1_ {%eval(&n_lev-1)} _temporary_;
  array _stt0_ {%eval(&n_lev-1)} _temporary_;
  array _stt1_ {%eval(&n_lev-1)} _temporary_;
```

```
%*;
%* Check for Stochastic Ordering of the Empirical;
%*    Distributions. The result is placed in
%*    the variable _case_;
%*;
  %stoc_ord(pop0=&treat0, pop1=&treat1, case=_case_);
%*;
%*   Case1: For max, use Isotonic Regression;
%*          For min, search over scores of 0s & 1s;
%*   Case2: For max. search over scores of 0s & 1s;
%*          For min, use Isotonic Regression;
%*   Case3: For both max and min, use Isotonic;
%*          Regression;
%*;
  select(_case_);
    when(1)
      do;
%*;
%* create the yis from the empirical distribution;
%*;

      do _ksw_j_ = 1 to dim(&treat0);
        _w_(_ksw_j_)  = (&treat0(_ksw_j_)+
                                  &treat1(_ksw_j_));
        _y0_(_ksw_j_) = &treat0(_ksw_j_)/
                        (&treat0(_ksw_j_)+&treat1(_ksw_j_));
        end;
%*;
%* find the isotonic regression;
%*;
      %pav(max_els=&n_lev,array=_y0_,weights=_w_);
%*;
%* Re-scale to include 0 and 1;
%*;
      do _ksw_j_ = 1 to dim(_y0_);
        _t0_(_ksw_j_) = (_y0_(_ksw_j_) - _y0_(1))/
                (_y0_(&n_lev) - _y0_(1));
        end;
%*;
%* Compute the correlation, the t, and
%* CMH for those scores;
%*;
       %cor(pop0=&treat0, pop1=&treat1,
            score=_t0_,r=&min_r,
            t=&min_t,cmh=&min_cmh);
%*;
%* copy these values into the output variables;
%*;
       do _ksw_k_ = 1 to dim(&minscore);
          &minscore(_ksw_k_) = _t0_(_ksw_k_);
          end;
%*;
%* This finishes the minimum score. ;
%* Now, construct scores of the form ;
%*   0=x(1),…,x(j) and 1=x(j+1),…,x(k) for ;
%*     j=2,…,k-1 ;
%*   then pick the one that gives the minimum ;
%*      correlation ;
%*;
%*   construct a score;
%*;
      do _ksw_j_ = dim(&treat0) to 2 by -1;
        do _ksw_k_ = _ksw_j_ to dim(&treat1);
          _t1_(_ksw_k_) = 1;
          end;
```

```
%*;
%* compute the correlation, t and CMH;
%*;
        %cor(pop0=&treat0, pop1=&treat1,
             score=_t1_,r=_r_,t=_stud_t_,
             cmh=_cmh_);
%*;
%* copy the score and statistics into an ;
%* array for later interrogation;
%*;
        _r1_(_ksw_j_-1) = _r_;
        _stt1_(_ksw_j_-1) = _stud_t_;
        _cmh1_(_ksw_j_-1) = _cmh_;
        do _ksw_k_ = 1 to dim(_t0_);
          _z1_(_ksw_j_-1,_ksw_k_) = _t1_(_ksw_k_);
          end;
        end;
%*;
%* find the score that gives the max correlation;
%*;
        _max_r_ = -1;
        do _ksw_k_ = 1 to dim(_r1_);
          if (_max_r_ <= _r1_(_ksw_k_)) then
             do;
             _max_r_ = _r1_(_ksw_k_);
             _in_max_ = _ksw_k_;
             end;
          end;
%*;
%* copy these values to the output variables;
%*;
       do _ksw_k_ = 1 to dim(&maxscore);
          &maxscore(_ksw_k_) = _z1_(_in_max_,_ksw_k_);
          end;
       &max_r = _r1_(_in_max_);
       &max_t =   _stt0_(_in_max_);
       &max_cmh = _cmh0_(_in_max_);
       end;
    when(2)
      do;
%*;
%* the following is the same as above with the ;
%* roles of the distributions reversed;
%*;
      do _ksw_j_ = 1 to dim(&treat1);
        _w_(_ksw_j_)  = (&treat0(_ksw_j_)+
                         &treat1(_ksw_j_));
        _y1_(_ksw_j_) = &treat1(_ksw_j_)/
                        (&treat0(_ksw_j_)+
                         &treat1(_ksw_j_));
        end;

      %pav(max_els=&n_lev,array=_y1_,weights=_w_);

      do _ksw_j_ = 1 to dim(_y1_);
        _t1_(_ksw_j_) = (_y1_(_ksw_j_) -
            _y1_(1))/(_y1_(&n_lev) - _y1_(1));
        end;

      %cor(pop0=&treat0, pop1=&treat1,
           score=_t1_, r=&max_r,
           t=&max_t,cmh=&max_cmh);

       do _ksw_k_ = 1 to dim(&maxscore);
          &maxscore(_ksw_k_) = _t1_(_ksw_k_);
```

```
          end;

        do _ksw_j_ = dim(&treat0) to 2 by -1;
          do _ksw_k_ = _ksw_j_ to dim(&treat0);
            _t0_(_ksw_k_) = 1;
            end;

          %cor(pop0=&treat0,pop1=&treat1,
               score=_t0_,r=_r_,
               t=_stud_t_,cmh=_cmh_);

          _r0_(_ksw_j_-1) = _r_;
          _stt0_(_ksw_j_-1) = _stud_t_;
          _cmh0_(_ksw_j_-1) = _cmh_;
          do _ksw_k_ = 1 to dim(_t0_);
            _z0_(_ksw_j_-1,_ksw_k_) = _t0_(_ksw_k_);
            end;
          end;

          _min_r_ = 1;
          do _ksw_k_ = 1 to dim(_r0_);
            if (_min_r_ >= _r0_(_ksw_k_)) then
               do;
               _min_r_ = _r0_(_ksw_k_);
               _in_min_ = _ksw_k_;
               end;
            end;

         do _ksw_k_ = 1 to dim(&minscore);
            &minscore(_ksw_k_) = _z0_(_in_min_,_ksw_k_);
            end;
         &min_r =   _r0_(_in_min_);
         &min_t =   _stt0_(_in_min_);
         &min_cmh = _cmh0_(_in_min_);
        end;
      when(3)
        do;
%*;
%* Create the y sub i from the empirical distributions;
%*;
        do _ksw_j_ = 1 to dim(&treat0);
          _w_(_ksw_j_) = (&treat0(_ksw_j_)+&treat1(_ksw_j_));
          _y0_(_ksw_j_) =
           &treat0(_ksw_j_)/
           (&treat0(_ksw_j_)+&treat1(_ksw_j_));
          _y1_(_ksw_j_) = &treat1(_ksw_j_)/
           (&treat0(_ksw_j_)+&treat1(_ksw_j_));
          end;
%*;
%* Find the isotonic regression;
%*;
        %pav(max_els=&n_lev,array=_y0_,weights=_w_);
        %pav(max_els=&n_lev,array=_y1_,weights=_w_);
%*;
%* Re-scale to include 0 and 1;
%*;
        do _ksw_j_ = 1 to dim(_y0_);
         _t0_(_ksw_j_) = (_y0_(_ksw_j_) -
             _y0_(1))/(_y0_(&n_lev) - _y0_(1));
         _t1_(_ksw_j_) = (_y1_(_ksw_j_) -
             _y1_(1))/(_y1_(&n_lev) - _y1_(1));
          end;
%*;
%* compute the correlation, t and CMH for those scores;
%*;
```

```
      %cor(pop0=&treat0, pop1=&treat1,
           score=_t0_, r=&min_r,
           t=&min_t,cmh=&min_cmh);
      %cor(pop0=&treat0, pop1=&treat1,
           score=_t1_, r=&max_r,
           t=&max_t,cmh=&max_cmh);
%*;
%* copy these values into the output variables;
%*;
      do _ksw_j_ = 1 to dim(&maxscore);
         &minscore(_ksw_j_) = _t0_(_ksw_j_);
         &maxscore(_ksw_j_) = _t1_(_ksw_j_);
         end;
      end;
  end;
  Drop _case_ _r_ _stud_t_
       _cmh_ _ksw_k_
       _ksw_j_ _in_max_
       _in_min_;
  %mend ksw;
```

3. EXAMPLE

We illustrate this procedure with an example using data from Agresti (1984), where two treatments are used to try to heal ulcer craters.

Treatment	Larger	< 2/3 Healed	≥ 2/3 Healed	Healed
A	12	10	4	6
B	5	8	8	11

The DATA step implementing the KSW procedure for this data is:

```
options sasautos='c:\sugi';
data agresti;
  infile cards;
  array treat0 {*} a1 - a4;
  array treat1 {*} b1 - b4;
  array minscr {4};
  array maxscr {4};
  input a1 - a4;
  input b1 - b4;
  %ksw(treat0=treat0,treat1=treat1,
       n_lev=4,min_t=min_t,max_t=max_t,
       maxscore=maxscr,minscore=minscr);
  put minscr(*)= Min_t=;
  put maxscr(*)= max_t=;
  cards;
   12 10 4 6
   5  8  8 11
;
run;
```

The log for this example is:

NOTE: Copyright (c) 1989-1993 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) Proprietary Software Release 6.10 TS019

```
Licensed to CTB/MCGRAW-HILL, Site 0009289001.

NOTE: The SAS System for Microsoft Windows, Release 6.10 Limited Production
1   options sasautos='c:\sugi';
2   data agresti;
3    infile cards;
4    array treat0 {*} a1 - a4;
5    array treat1 {*} b1 - b4;
6    array minscr {4};
7    array maxscr {4};
8    input a1 - a4;
9    input b1 - b4;
10    %ksw(treat0=treat0,treat1=treat1,n_lev=4,
11       min_t=min_t,max_t=max_t,
12       maxscore=maxscr,minscore=minscr);
13    put minscr(*)= Min_t=;
14    put maxscr(*)= max_t=;
15    cards;

MINSCR1=0 MINSCR2=0 MINSCR3=0 MINSCR4=1 MIN_T=1.4151268421
MAXSCR1=0 MAXSCR2=0.4163545568 MAXSCR3=1 MAXSCR4=1
MAX_T=2.508647573
NOTE: The data set WORK.AGRESTI has 1 observations and 22 variables.
NOTE: The DATA statement used 6.7 seconds.
18  ;
19  run;
```

Note that there are 22 variables in this example. Eight are for the frequencies, 8 are the extreme scores, 2 are t-statistics, 2 are CMH statistics, and 2 are Pearson's correlations.

The empirical distribution of ulcer crater size for Treatment A is stochastically less than that for Treatment B. Thus, the minimum scores are found by searching through the scores of 0's and 1's and the maximum scores are found using the PAVA. The resulting output gives the minimum score $t_1 = t_2 = t_3 = 0$ and $t_4 = 1$ with minimum t of 1.42 and the maximum score of $s_1 = 0$, $s_2 = .4164$, $s_3 = s_4 = 1$ with a corresponding maximum t of 2.508. There are no scores which will accept the alternative that Treatment A is better than Treatment B. It is clear that there are some scores which lead to rejection of the null hypothesis that the two treatments are the same and that there are some scores that fail to reject the null hypothesis in favor of a difference in the two treatments (or that Treatment B is better than Treatment A). This straddling situation requires the practitioner to re-evaluate what differences in the treatments are of practical significance. Upon closer

inspection of the minimum and maximum scores, we see that if the practitioner is interested in drugs that show any type of improvement regardless of the degree of improvement then the two treatments are very similar. On the other hand, if the practitioner is really interested in completely or almost completely healing ulcer craters then this data presents evidence that Treatment B is better than Treatment A.

4. CONCLUSION

The KSW procedure gives an approach for analyzing two-sample ordinal data. Most methods either explicitly or tacitly assign scores to the levels of the ordinal variable. For true ordinal variables there is no one underlying score that adequately describes the levels. Thus, practitioners often try different scores or different methods, often with conflicting results. KSW helps reconcile these differences by finding the scores which maximize and the scores which minimize both the CMH and the t-statistic. In this paper, we implement the KSW procedure. To enhance the portability of the KSW macro, the code is written using only base SAS software.

The KSW statistics should not be thought of as test statistics. They are extreme values over a set of test statistics generated from all possible ordinal scorings (including scorings that pool levels of the ordinal variable). Thus, we have purposely left p-values out of the KSW macro. As was seen in the ulcer crater example, even though there is no distribution theory for the KSW procedure, both the extreme t-statistics and the corresponding scores provide a deeper insight into the data than any one of the usual methods used alone.

The more general problem of finding extreme scores for ordinal response variables in an ANOVA setting is treated in Gautam (1991). Streitberg and Roehmel (1988) give a method for computing bounds for p-values for a class of permutation tests in the two-sample setting. They do not give extreme scores and their algorithm is implemented in TESTIMATE.

5. REFERENCES

Agresti, A. (1984). *Analysis of Ordinal Categorical Data*, John Wiley and Sons, New York.

Conover, W. J. and Iman, R. L. (1981). "Rank Transformations as a Bridge between Parametric and Nonparametric Statistics". *The American Statistician*, 35, 124-129.

Gautam, S. (1991). "Applications of the t, T^2 and F Statistics to Ordinal Categorical Data". Unpublished PhD dissertation. The University of Texas at Dallas, Program in Mathematical Statistics.

Kimledorf, G., Sampson, A. R. and Whitaker L. R. (1992). "Min and Max Scoring for Two-Sample Ordinal Data". *Journal of the American Statistical Association*, 87, 241-247.

Mantel, N. (1963). "Chi-Squared Tests with One Degree of Freedom: Extensions of the Mantel-Haenlszel Procedure". *Journal of the American Statistical Association*, 58, 690-700.

McCullagh, P. and Nelder, J. A. (1983). *Generalized Linear Models*, London: Chapman & Hall.

Robertson, T., Wright, F. T. and Dykstra, R. L. (1988). *Order Restricted Statistical Inference*, John Wiley and Sons, New York.

Streitberg, B. and Roehmel, J. (1988). "Simultane Anwendung aller mØglichen Tests zu einem vorgelegten Testproblem: P-Wert Schranken fòr monotone Permutationstests im 2-Stichprobenproblem, in: Muliple Hypothesenpròfung". edited by P. Bauer, G. Hommel, E. Sonnemann, Symposium 6. Und 7. Nov. 1987, Springer-Verlag, Berlin.

TESTIMATE 5.1 (1992). idv Datenanalyse und Versuchsplanung Wessobrunner Straße 6, 8033 Gauting, Germany.

6. APPENDIX

```
%*****************************************;
%* Macro :PAV                            ;
%* Author:Michael Whitaker               ;
%* Input: max_els = The max numb of elements  ;
%*        array = The array of data      ;
%*        weights= Weights used in the   ;
%*                 regression            ;
%* Output:array = the same array as above     ;
%* Required Macros : None                ;
%* Required Procs  : None                ;
%* Comments :                            ;
%*    This will perform an Isotonic regression;
%*    in one dimension. The array will   ;
%*    hold the processed data            ;
%*                                       ;
%*****************************************;
%macro Pav(max_els=,Array=,Weights=);
%global index;
%if %quote(&index)=
 %then
  %let index = 1;
 %else
  %let index = %eval(&index+1);
%let pooled = _pool&index._;
%let parray = _parr&index._;
```

```
%let pwghts = _pwgt&index._;
array &pooled (&max_els) _temporary_;
array &parray (&max_els) _temporary_;
array &pwghts (&max_els) _temporary_;
If dim(&array) = 1 then Go to epav&index;
do _pav_j_ = 1 to dim(&array);
   &pooled(_pav_j_) = 0;
   &parray(_pav_j_) = 0;
   &pwghts(_pav_j_) = 0;
   end;
&parray(1) = &array(1);
&pwghts(1) = &weights(1);
_pav_j_ = 1;
Do _pav_i_ = 2 to dim(&array);
  If (&parray(_pav_j_) > &array(_pav_i_))
    then
      do;
      _plwght_ = &pwghts(_pav_j_) + &weights(_pav_i_);
      _plval_ = ((&parray(_pav_j_)*&pwghts(_pav_j_)) +
                 (&array(_pav_i_)*&weights(_pav_i_)))/
                 _plwght_;
      &pooled(_pav_i_) = 1;
      if _pav_j_ > 1 then
         do;
         _pav_j_ = _pav_j_ - 1;
         _pav_jj_ = _pav_i_;
         do while((&parray(_pav_j_) > _plval_) &
                  (_pav_i_ >= 1));
           _tplval_ = _plval_;
           _tplwgt_ = _plwght_;
           do until(&pooled(_pav_jj_));
              _pav_jj_ = _pav_jj_ - 1;
              end; /* do until */
           _plwght_ = &pwghts(_pav_j_) + _tplwgt_;
           _plval_ = ((&parray(_pav_j_)*&pwghts(_pav_j_)) +
                      (_tplval_*_tplwgt_))/_plwght_;
           &pooled(_pav_jj_) = 1;
           _pav_j_ = _pav_j_ - 1;
           end; /* do while */
         _pav_j_ = _pav_j_ + 1;
         end; /* If _pav_j_ > 1 */
       &parray(_pav_j_) = _plval_;
       &pwghts(_pav_j_) = _plwght_;
       end; /*  (&parray(_pav_i_) > &array(_pav_i_)) then */
   else
     do;
     _pav_j_ = _pav_j_ + 1;
     &parray(_pav_j_) = &array(_pav_i_);
     &pwghts(_pav_j_) = &weights(_pav_i_);
     end;
   end; /* _pav_i_ = 2 to dim(&array); */
&array(1) = &parray(1);
_pav_j_ = 1;
_pav_jj_ = 1;
do _pav_j_ = 2 to dim(&array);
  if ^&pooled(_pav_j_) then _pav_jj_ = _pav_jj_ + 1;
  &array(_pav_j_) = &parray(_pav_jj_);
  end;
Epav&index:
drop _pav_j_ _pav_i_ _pav_jj_ _plval_ _plwght_ _tplval_ _tplwgt_;
%mend;

%*****************************************;
%* Macro :Stoc_ord                       ;
%* Author:Michael Whitaker               ;
```

```
%* Input: pop0 = The first freq dist          ;
%*        pop1 = The second freq dist         ;
%* Output:case = the case (1, 2, or 3)        ;
%* Required Macros : None                     ;
%* Required Procs  : None                     ;
%* Comments :                                 ;
%*    This will check two empirical           ;
%*     probability distributions for          ;
%*     stochastic dominance. Case=1 is pop1   ;
%*     is dominate, case=2 is pop1 is         ;
%*     dominate, and case = 3 is neither      ;
%*     are dominate;                          ;
%*                                            ;
%********************************************;
%macro Stoc_ord(pop0=, pop1=, case=);
&case=.;
_sum_M_ = 0;
_sum_N_ = 0;
do _stoc_j_ = 1 to dim(&pop0);
   _sum_M_ = _sum_M_ + &pop0(_stoc_j_);
   _sum_N_ = _sum_N_ + &pop1(_stoc_j_);
   end;
_case_1_ = 1;
_case_2_ = 1;
Do _stoc_j_ = 2 to dim(&pop0);
   _psum_M_ = 0;
   _psum_n_ = 0;
   do _stoc_k_ = _stoc_j_ to dim(&pop0);
      _psum_M_ = _psum_m_ + &pop0(_stoc_k_);
      _psum_n_ = _psum_n_ + &pop1(_stoc_k_);
      end;
   _Case_1_ = (_Case_1_ & ((_Psum_M_/_sum_m_) >= (_psum_n_ / _sum_n_)));
   _Case_2_ = (_Case_2_ & ((_Psum_M_/_sum_m_) <= (_psum_n_ / _sum_n_)));
   end;
if _case_1_ then &case=1;
else if _case_2_ then &case=2;
else &case=3;
drop _psum_m_ _psum_n_ _sum_m_ _sum_n_ _case_1_ _case_2_ _stoc_k_
_stoc_j_;
%mend stoc_ord;


%********************************************;
%* Macro :cor                                 ;
%* Author:Michael Whitaker                    ;
%* Input: pop0 = The first freq dist          ;
%*        pop1 = The second freq dist         ;
%*        score = the score to use            :
%* Output: r= the correlation statistic       ;
%*         t= the Student t                   ;
%*         cmh= the cmh statistic             ;
%* Required Macros : None                     ;
%* Required Procs  : None                     ;
%* Comments :                                 ;
%*    This will copute the r, t and cmh for   ;
%*      pop0, pop1 and score                  ;
%*                                            ;
%********************************************;
%macro cor(pop0=,pop1=,score=,r=,cmh=,t=);
%put &cmh &r &t;
_tmx_ = 0;
_tnx_ = 0;
_tmnx2_ = 0;
_tm_ = 0;
```

```
_tn_ = 0;
do _cor_k_ = 1 to dim(&pop0);
   _tmx_ = _tmx_ + &pop0(_cor_k_)
           * &score(_cor_k_);
   _tnx_ = _tnx_ + &pop1(_cor_k_)
           * &score(_cor_k_);
   _tmnx2_ = _tmnx2_ +
             ((&pop0(_cor_k_) + &pop1(_cor_k_)) *
             &score(_cor_k_) ** 2);
   _tm_ = _tm_ + &pop0(_cor_k_);
   _tn_ = _tn_ + &pop1(_cor_k_);
   end;
_tt_ = _tm_ + _tn_;
&r = sqrt((_tt_-1)/_tt_) /
     sqrt(_tt_-1) * sqrt(_tm_*_tn_) *
     ((_tnx_/_tn_) - (_tmx_/_tm_)) /
      sqrt(_tmnx2_ - ((_tmx_ + _tnx_) ** 2 /_tt_));
&cmh = (_tt_-1) * &r ** 2;
&t   = sqrt(_tt_-2) * &r / sqrt(1 - &r ** 2);
drop _cor_k_ _tm_ _tn_ _tt_
     _tmx_ _tnx_ _tmnx2_;
%mend;
```

INITIAL DISTRIBUTION LIST

1. Research Office (Code 08) 1
   Naval Postgraduate School
   Monterey, CA 93943-5000

2. Dudley Knox Library (Code 52) 2
   Naval Postgraduate School
   Monterey, CA 93943-5002

3. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

4. Department of Operations Research 1
   Editorial Assistant (Code OR/Bi)
   Naval Postgraduate School
   Monterey, CA 93943-5000

5. Prof. Lyn R. Whitaker (Code OR/Wh) 15
   Naval Postgraduate School
   Monterey, CA 93943-5000

6. Michael D. Whitaker 5
   CTB/McGraw-Hill
   20 Ryan Ranch Rd
   Monterey, CA 93940