A SAS macro (*ordalpha*) to compute ordinal Coefficient Alpha

Karen L. Spritzer and Ron D. Hays (December 14, 2015)

Polychoric ordinal alpha can be used to assess the reliability of polytomous ordinal items (Bonanomi et al., 2015). Gadermann, Guhn and Zumbo (2012) published an article that included R code to estimate ordinal alpha. Similarly, Kapitula (2014) provided SAS code to compute ordinal alpha but the code was specific to one application.

We revised the Kapitula (2014) SAS code and created a macro (*ordalpha*) for estimation of ordinal alpha. The macro (*ordalpha.mac*) is provided in Appendix A. A SAS file to call up ordalpha.mac with a specific example is provided in Appendix B. The example is an analysis of the reliability of 6 physical functioning items (Hays et al., in press). This study reported a coefficient alpha (unstandardized) of 0.92. Ordinal alpha is 0.98 (see below). Note that ordinal alpha is based on (polychoric) correlations among items and corresponds to standardized coefficient alpha rather than the more commonly reported unstandardized alpha. Consistent with Kapitula's (2014) program, *ordalpha.mac* also yields auxiliary output such as a scree plot of eigenvalues and a figure with factor loadings based on polychoric correlations (not shown below).

```
                                  itemcor

                1  0.9564111  0.9060553  0.8784835  0.8717454  0.9283596
        0.9564111          1  0.9308338  0.9028814  0.8744958   0.943748
        0.9060553  0.9308338          1  0.8815739  0.8431046  0.9402031
        0.8784835  0.9028814  0.8815739          1  0.9142043  0.9267331
        0.8717454  0.8744958  0.8431046  0.9142043          1  0.8969335
        0.9283596   0.943748  0.9402031  0.9267331  0.8969335          1

                                          numbitem

                            Number of items         6

                                   alpha

                                 0.9830772


                                   theta

                                 0.9831203
```

References

Bonanomi, A., Cantaluppi, G., Ruscone, M. N., & Osmetti, S. A. (2015). A new estimator of Zumbo's ordinal alpha: A copula approach. Qual Quant, 49, 941-953.

Gadermann, A. M., Guhn, M., & Zumbo, B. D. (2012). Estimating ordinal reliability for Likert-typeand ordinal item response data: A conceptual, empirical, and practical guide. Practical Assessment, Research & Evaluation, 17 (3). Available online: http://pareonline.net/getvn.asp?v=17&n=3

Hays, R.D., Mallett, J. S., Gaillot, S., & Elliott, M. N. (in press). Performance of the Consumer Assessment of Healthcare Providers and Systems (CAHPS®) Physical Functioning Items. Medical Care.

Kapitula, L. R. (2014). Estimating ordinal reliability using SAS®. Paper 2042-2014.
http://support.sas.com/resources/papers/proceedings14/2042-2014.pdf
http://www.sascommunity.org/wiki/Estimating_Ordinal_Reliability_Using_SAS%C2%AE

Acknowledgements

This work was supported by a grant from National Cancer Institute (1U2-CCA186878-01).

Appendix A: ordalpha.mac

```
%macro ordalph(ds=,items=);
*************************************************************;
data itemdata;
set &ds(keep=&items);
%let qitems=q1-q&nitems;
if nmiss(of &qitems) =0; /* use listwise deletion*/
totscore= sum(of &qitems);
label totscore = "Overall Score";
keep totscore &qitems;
run;

* get sampsize for use later on;
proc sql noprint;
        select count(*) into :sampsize
        from itemdata;
quit;
*------------------------------------------------------------;
*------------------------------------------------------------;
TITLE "&ds: Pearson correlation matrix"; run;
proc corr data=itemdata outp=pearsonc alpha noprob nosimple;
        var &qitems;
run;
proc print data=pearsonc(where=(_type_="CORR")) noobs label ;
var _name_ &qitems;
format &qitems 5.3;
label _name_ ="item";
run;

ods exclude all;
proc freq data=itemdata ;
tables (&qitems)*(&qitems) /plcorr noprint ;
ods output measures=mycorr (where=(statistic="Tetrachoric Correlation"
                                or statistic="Polychoric Correlation")
                            keep = statistic table value);
run;
ods exclude none;
*------------------------------------------------------------;
```

```
*------------------------------------------------------------;
data mycorr;
        set mycorr;
x = scan(table, 2, " *");
y = scan(table, 3, " *");
group=substr(y,2)+0;
run;

proc sort data=mycorr;
        by group x;

proc transpose data = mycorr out=polycorr (drop = group _name_)     ;
id x;
by group;
   var value ;
run;

data polycorr(type=corr);
   _TYPE_='CORR';
     set polycorr end=endc;
     _name_=compress("q"||put(_n_,2.));
     output;
     array q _numeric_;
     if endc then do;
       _TYPE_='N';
         do over q;
           q=&sampsize;
           end;
       output;
       end;
run;

TITLE "&ds: Polychoric correlation matrix"; run;
proc print data = polycorr(where=(_type_="CORR")) noobs label;
        var _name_ &qitems;
label _name_ ="item";
format &qitems 5.3;
run;
```

```
*------------------------------------------------------------;
*------------------------------------------------------------;
TITLE "&ds: Factor Analysis on Polychoric Matrix"; run;
/* do Factor Analysis on the Correlation Matrix */
proc factor data=polycorr method=ml nfactors=1 plots=all;
run;
*------------------------------------------------------------;
/* calculate alpha and omega and store it in a data set */
%macro varlist;
%do i = 1 %to &nitems;
        q&i
%end;
%mend;
%macro calcalpha(qs=q, cordat=polycorr, out=alpha);
/* qs= the subsection of the instrument */
/* cordat is the correlation matrix data */

TITLE "&ds: IML input=&cordat"; run;
proc iml;
     use &cordat(where=(_type_="CORR")) var {%varlist};
     read all var {%varlist } into itemcor;
     section="&qs";
     datused="&cordat";
     numbitem=trace(itemcor);
     one=j(numbitem,1);
     print 'The Input Correlation';
     print itemcor;
     print 'Number of items' numbitem;
     sumcor=(one`)*itemcor*one; /*sum all correlations*/
     alpha=(numbitem/(numbitem-1))*(1-(numbitem / sumcor));
     print alpha;
     L = eigval(itemcor);
     theta=(numbitem/(numbitem-1))*(1-1/L[1]);
     print 'Theta';
     print theta;
     /* to save results in a data set */
     create &out var {section datused alpha theta};
                            /** create data set **/
     append;   /** write data **/
     close &out;        /** close the data set **/
quit;
run;
%mend;
********************************************************;
*********************************;
%calcalpha(qs=q, cordat=pearsonc, out=alphap);
%calcalpha(qs=q, cordat=polycorr);
************************************************;
*close support files;
ods pdf close;
quit;
%mend ordalph;
```

Appendix B: ordalpha_setup.sas

```
ods graphics on;
******************************************************************************************
*  Program Name      : ordalpha.sas
*
*  Purpose of Code   : ordinal alpha reliability
*
*  Author/Date       : Modified by RH and KS, Dec 11 2015.
*  Based on Laura Ring Kapitula, Estimating ordinal reliability using SAS,
*  Paper 2042-2014.
*  http://www.sascommunity.org/wiki/Estimating_Ordinal_Reliability_Using_SAS%C2%AE
*
*  Required edits include: dsname (and data), workdir, and progname.
*  Within datastep that sets up processing, be sure to name your items from q1 up,
*  and set the nitems value.
*****************************************************************************************;
%let progname=ordalph_setup;
%let workdir=C:\Ordinal_Alpha;
libname mydata "&workdir";

/* always output supporting file */
ods pdf file="&workdir\&progname..pdf";
footnote2 "Program Name: &progname..sas on %sysfunc(today(),mmddyy10.)
%sysfunc(time(),time.)";
*************************************************************;
** rename items to q1-q.. and set nitems and ds(name);
*************************************************************;
DATA pfdata;
 SET mydata.pfdata;
rename
BATHING1  = q1
DRESSING1 = q2
EATING1   = q3
CHAIRS1   = q4
WALKING1  = q5
TOILET1   = q6;
```

```
%let nitems=6;  * number of items;

%let ds=pfdata; * datasetname;

RUN;

**************************************************************;

** no changes beyond this point;

%include "&workdir\ordalpha.mac" /nosource;

*options macrogen mprint mlogic symbolgen;

%ordalph(ds=&ds,items=q1-q&nitems);
```