Paper 264-27

Using SAS® to Perform Robust *I*-Sample Analysis of Means Type Randomization Tests for Variances for Unbalanced Designs

Peter Wludyka, University of North Florida, Jacksonville, FL
Ping Sa, University of North Florida, Jacksonville, FL

ABSTRACT

A SAS macro for performing an Analysis of Means (ANOM) type randomization test for testing the equality of *I* variances when the samples are of different sizes (an unbalanced design) is presented. Robust tests are required when sampling from non-normal populations. Randomization techniques for testing statistical hypotheses can be used when parametric tests that require normality are inappropriate. Suppose that *I* independent samples have been collected. Randomization tests are based on shuffles or rearrangements of the (combined) sample. Putting each of the *I* samples "in a bowl" forms the combined sample. Drawing samples "from the bowl" forms a shuffle. Shuffles can be made with replacement (bootstrap shuffling) or without replacement (permutation shuffling). The tests that are presented here offer two advantages. They are robust to non-normality and they allow the user to graphically present the results via a decision chart similar to a Shewhart control chart. The decision chart facilitates easy assessment of both statistical and practical significance.

INTRODUCTION

Often it is useful to test whether each of *I* populations have the same variance. This homogeneity of variance (HOV) hypothesis may be written

$$H_0 : \sigma_1^2 = \ldots = \sigma_I^2 \qquad (1)$$

where σ_i^2 is the variance of the ith population. The alternative hypothesis is $H_A :$ not H_0. This paper is concerned with the case where $I > 2$ and the *I* sample sizes are different; that is, where three or more populations are being compared based on samples of size n_i for $i = 1,\ldots, I$.

RANDOMIZATION TESTS

There are two types of randomization tests: exact randomization tests and approximate randomization tests. Randomization tests are based on shuffles (resamplings or rearrangements) of the (combined) sample. Suppose that *I* samples of size *n* have been selected. The combined sample is formed by putting each of the *I* samples "in a bowl." Let *X* denote the combined sample. "Drawing from the bowl" forms shuffles. The shuffles can be made with replacement (called a bootstrap shuffle) or without replacement (called a permutation shuffle).

ANALYSIS OF MEANS TYPE RANDOMIZATION TESTS FOR VARIANCES

Four ANOM type randomization tests for variances when the samples are of equal size were proposed by Bernard and Wludyka (2001). These randomization tests are randomization versions of the Analysis of Means for Variances (ANOMV), the normal based test proposed by Wludyka and Nelson (1997). These tests can be used when normality cannot be safely assumed. A SAS® macro for performing one of these randomization tests for variances (the general purpose test deemed best) is in Wludyka (2000).

Two versions of a randomization test for variances useful when the sample sizes are not all the same was proposed in Sa and Wludyka (to appear). A SAS® macro to perform this test is presented in this paper. This test can be performed by plotting the *I* sample variances (or standard deviations) on a decision chart or equivalently by evaluating a set of empirical p-values. Using the decision chart, the HOV hypothesis is rejected if any sample variance plots outside the decision lines.

Either permutation shuffles or bootstrapping shuffles can be used but the test presented here is a permutation test. Previous work by the authors has shown that permutation tests are more powerful than the bootstrap versions.

See Bernard and Wludyka (2001) and Sa and Wludyka (to appear) for Monte Carlo results which show that randomization tests are as powerful and robust as typical robust alternatives, such as Levene's test, and that in certain circumstances the randomization tests are much more powerful than Levene's test.

UBRANDANOMV-R

RANDANOMV-R, which is based on the ratio of each sample variance to the sum of the sample variances, is the balanced randomization version of the Analysis of Means for Variances (ANOMV) proposed by Wludyka and Nelson (1997). ANOMV is a test that is suitable when the *I* populations can safely be assumed to be normal. For a SAS® macro to perform this test see Wludyka (1999). A SAS® macro for the randomization test RANDANOMV-R was presented in Wludyka (2000).

UBRANDANOMV-R will be used to denote the randomization test used in the unbalanced case.

The basic idea behind the procedure is as follows. (Note that there are two versions of the test: one based on the sum of the variances and the other based on MSE. The description will be in terms of the sum of the variances. For details see Sa and Wludyka (to appear).) Using the initial sample, for each sample determine the ratio of the sample variance to the sum of the *I* variances. Determine the number of distinct sample sizes and denote that by *K*. For each sample size (values of k = 1, …,K), and based on the initial samples, determine the largest ratio and the smallest ratio. Permutation shuffle the data *NS* times, where *NS* is typically 1000 or more. The variables *ngmx(k)* and *ngmn(k)* are used to count the number of times that the shuffled values (the maximum and minimum variance ratios) are more extreme than those from the initial sample (the unshuffled data). Small values of *ngmx(k)* and *ngmn(k)* are evidence against the HOV hypothesis (1). In particular, whenever

$$\frac{ngmx(k)+1}{NS+1} < \frac{\alpha}{2K}$$

or

$$\frac{ngmn(k)+1}{NS+1} < \frac{\alpha}{2K}$$

the equal variances hypothesis is rejected. This method is the p-value method. Equivalently, the quantiles of the empirical distribution of the variance rations can be used to determine decision lines for a decision chart. There is one set of upper and lower decision lines for each sample size (k = 1, …,K). If a

sample variance plots outside the corresponding decision lines, the HOV hypothesis is rejected.

THE %UBRANOMV MACRO

The %UBRANOMV macro can be used to perform the UBRADANOMV-R test for the equality of *I* variances. The variable *NS* is used to identify the user specified number of shuffles --- typically 1000.

Data Preparation

The input data file must contain three variables: a classification variable that identifies each of the *I* populations, a measurement variable, and an observation number (within the classification). The basic idea is that *NS* permutation shuffles of the data will be made. This is straightforwardly achieved by replicating the data set *NS* times and assigning a shuffle variable (1,…,NS) and a uniform (pseudo) random variable (in the data step in the program this variable is denoted shufno) to each data point. The data set is then sorted by the random number (shufno) within shuffle, so that there are then NS random permutation samples. These samples are used to gauge how unusual the original sample is with regard to the ratio of smallest variance to the sum of the variances and the ratio of the largest variance to the sum of the variances (for each distinct sample size). See the INPUT DATA comment in the source code.

%UBRANOMV Variables

See the comments with the definitions of the macro. The key is to indicate the number of distinct samples sizes (K). Put any value for *n*. For small alpha use more than 1,000 shuffles.

%UBRANOMV Output

In the example corresponding to the output in Tables 1-3 and Figure 1 there are five populations being compared (see source code for details). The output is for the MSE method. For the first two populations samples of size five were selected, for population three a sample of size 10 was selected, and for populations four and five samples of size twenty were selected. Hence there are K=3 distinct sample sizes. The p-value tables can be used to decide the hypothesis for any test-wide alpha the users wishes. Using table 1, one sees that there is a p-value-high (PVALHI = .004) lower than the critical p-value (CRITPVAL = alpha/2K = 0.05/2(3) = .0083). That would be sufficient to reject the HOV hypothesis for alpha = 0.05 and conclude that at least one variance from a population from which the sample size equals 20 is higher than the other variances. From table 2, one can see that there is a *p*-value-low less than the critical *p*-value so that at least one variance for a population from which a sample of size twenty was taken is lower than the others.

Table 1: P-value high output

```
    emprical p-values hi with critical p-value using method 2
            method 1 is the sum of variances method
                    method 2 is the mse method    07:25 Sunday, September

  OBS    SSIZE    _TYPE_    _FREQ_    NGMX    PVALHI    CRITPVAL

   1        5        0        1000      988   0.98801    .0083333
   2       10        0        1000      340   0.34066    .0083333
   3       20        0        1000        3   0.00400    .0083333
```

Table 2: P-value low output

```
    empirical p-values low with ctitical p-values using method 2
            method 1 is the sum of variances method
                    method 2 is the mse method    07:25 Sunday, September

  OBS    SSIZE    _TYPE_    _FREQ_    NGMN    PVALLO    CRITPVAL

   1        5        0        1000      135   0.13586    .0083333
   2       10        0        1000      660   0.66034    .0083333
   3       20        0        1000        0   0.00100    .0083333
```

The decision table (Table 3) can be used to decide the hypothesis. The (test-wide) level of significance is 0.05 for this example. If any of the variances are larger than the corresponding Upper Decision Limit (UDL) or less than the Lower Decision Limit (LDL) then the HOV hypothesis is rejected. Observe that the sample variance for population 4 (TREAT = 4) plots below the LDL = 6.13, which implies that the variance for population 4 is smaller than the others. Since the variance for population 5 is higher than the UDL = 79.77 one can conclude that the variance for population 5 is higher than the others

Table 3: Decision Table

```
                        decision limits using method 2
                    method 1 is the sum of variances method
                        method 2 is the mse method    07:25 Sunday, September 9,

  OBS    TREAT    SSIZE    RANKHI    RANKLOW    VARBYTO    LDL        CL        UDL

   1       1        5       993        8        2.5298   0.15136   37.0038   200.022
   2       2        5       993        8        1.2711   0.15136   37.0038   200.022
   3       3       10       993        8       41.0978   1.66452   37.0038   117.268
   4       4       20       993        8        0.4920   6.12953   37.0038    79.774
   5       5       20       993        8       86.3565   6.12953   37.0038    79.774
```
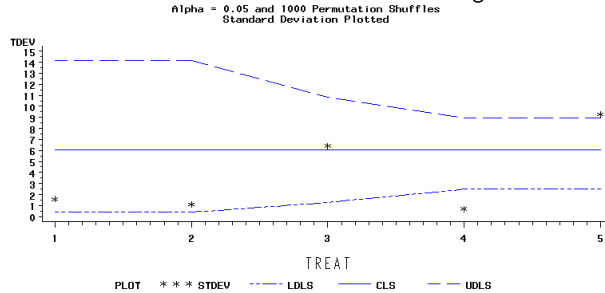

Figure 1: Decision Chart

The decision chart is quite easy to interpret. The decision lines in Figure 1 are scaled so that the standard deviations are plotted. The standard deviation for population 4 plots below the lower decision line. The standard deviation for population 5 plots above the UDL. Hence, the HOV hypothesis is rejected at alpha = 0.05 and one concludes that variability is significantly lower for population four and higher for population five. Practical significance can be assessed once statistical significance is established.

%UBRANOMV SOURCE CODE

```
/*******************
     UBRANDANOMV-R TEST PERFORMED USING
     UBRANOMV
     A TEST FOR THE EQUALITY
     OF I VARIANCES BASED ON I
INDEPENDENT
     SAMPLES OF SIZE N(i).
     Sum of variances and MSE OPTIONS
     users provides data
**************/


/*****************************
     DATA INPUT
     data in ascending
     sample size order
*****************************/
```

```
data data1a ;
    input
treat      bcount      obnum;

if treat = 1 then bcount = bcount;
if treat = 2 then bcount = bcount;
if treat = 3 then bcount=bcount*5;
if treat = 4 then bcount = bcount;
if treat = 5 then bcount = bcount*10;
cards;

1      -0.705729235     1
1      -2.685592335      2
1      0.722670848       3
1      1.053513188       4
1      0.93268909      5
2      -1.186581358      1
2      -1.226901531      2
2      0.959655608       3
2      -0.679501682      4
2      1.01833848      5
3      -0.011049224      1
3      -0.198134558      2
3      -0.085267402      3
3      -1.781118044      4
3      1.967819117     5
3      -0.994184575      6
3      -0.761863248      7
3      2.409469744     8
3      0.708728294     9
3      0.186736315     10
4      -0.232964794      1
4      -1.062487627      2
4      0.015598971     3
4      0.937827735     4
4      0.591667231     5
4      -0.170523435      6
4      0.914187694     7
4      -0.266122697      8
4      -0.787879344      9
4      -0.016746071     10
4      -1.253631581     11
4      -0.530573061     12
4      -0.081325879     13
4      -0.185347062     14
4      0.403908871     15
4      -0.725128757     16
4      -1.0078179     17
4      1.075268301     18
4      -0.834087359      19
4      -0.874413217      20
5      0.264196842     1
5      1.401967893     2
5      2.300876076     3
5      -1.112236987      4
5      0.204508979     5
5      -0.175798505      6
5      -0.08929419     7
5      -0.470470241      8
5      -0.825048119      9
5      -0.533109414     10
5      0.993361482     11
5      -0.731470209     12
5      -1.299210908     13
5      -0.776744855     14
5      0.29379521     15
5      -0.108632321      16
5      1.434614205     17
5      0.76429842     18
5      0.16658305     19
5      -0.301947694      20
;
/* proc print ;*/
title ' summary statics for data set';
proc means;by treat;  var bcount; run;

/****************************
     DATA PREPARATION
       user provides
       tops =  (see Macro)
       pops =
       array dimension
       sample sizes
       tops also in oshufrep
****************************/
data data1b;
tops = 2;
pops = 5;
       array samp(5);
       samp(1) = 5;
samp(2)=5;samp(3)=10;
samp(4)=20;samp(5)=20;
       ns =1000;
do treat=1 to pops;
            do obnum = 1 to
samp(treat);
                ssize = samp(treat);
                do shuf = 1 to ns;
                       shufno =
uniform(-1);
                       output ;
       end; end; end;

       /* proc print; */
data data1;
       merge data1b data1a;
       by treat obnum;
data data1c;
       set data1;
       proc sort; by shuf treat;
       run;

proc means data=data1c noprint;
     by shuf treat ; var bcount;
     output out = origvars var =
     varbyt0 n=ssize ;
data origvars;
       set origvars;
     ss0 = (ssize-1)*varbyt0; df =
     (ssize-1);
proc means data = origvars noprint;
     by shuf ; var varbyt0; output
     out= osumvar sum=sumvar0;
proc means data = origvars noprint;
     by shuf ; var ss0; output out=
     osumss sum=sumss0;
proc means data = origvars noprint;
```

```
      by shuf ; var df; output out=
      osumdf sum=sumdf0;
data origsums;
      merge osumvar osumss osumdf; by
      shuf;
data oshufrep;
        merge origvars origsums; by
shuf;
        tops = 2;
            if tops = 1 then
      ratbyt0=varbyt0/sumvar0; else
      ratbyt0 = ss0/sumss0;
        mse0 = sumss0/sumdf0;
        title 'oshufrep: original
ratios shuffle replicated';
        /* proc print;*/
        run;
data admax0;
        set oshufrep;
        proc means noprint;
      by shuf ssize; var ratbyt0;
      output out = admax0 max =
      maxratbn min = minratbn;
        title 'admax0: max ratios by
ssize';
        /* proc print data = admax0;*/
        /* proc print; */

/******************************
        DEFINE MACRO
******************************/

%macro ubranomv(
  k=,      /* the number of populations
being compared */
  n=,      /* the sample size
*/
  alpha=, /* level of significance
*/
  ds=,     /* the data set containing
the observations */
  varname=,          /* the variable
name for the observations    */
  classvar=,    /* the variable name
for the populations    */
  ns =,   /* the number of shufles */
  tops=,          /*  1 = randanomv-R
*/
  kn =);  /* the number of distinct
sample sizes*/




/*************************
Print Data Set
*************************/

data basedat;
        set data1;
        if  shuf > 1 then delete;
        title 'basedat: original data
set';
```

```
        proc print;  var &classvar
&varname;
        run;

/*********************************
   DETERMINE RANOMV CRITICAL VALUES
*********************************/
title 'initial data set';
        /* proc print data = &ds;*/
        run;



data shufdat1;
      set &ds;
      proc sort; by shuf shufno;
      title 'shuffled data';
      /* proc print;*/
        run;




data shufdat2;
        set shufdat1;
        drop &classvar;
      title 'shuffled data without
      clasvar';
        /*proc print;*/
        run;




data origdat0;
        set data1;
        proc sort; by shuf;
        title 'original data with
variable';
        /*proc print;*/



data origdat;
        set data1; drop &varname;
        proc sort; by shuf;
        title 'original data without
variable';
        /* proc print; */



data shufdat3;
        merge shufdat2 origdat;
        title 'merged data';
      /*   proc print; */
        run;

proc means data = shufdat3 noprint;
        by shuf &classvar; var &varname
; output out = stats1 var = varx
n=ssize;
```

```
        title 'variance of each
shuffled sample';
        /*proc print data = stats1;*/
        run;
data stats1; set stats1;
        ss=(ssize-1)*varx;
        title 'ss calculated';
       /* proc print;*/
        run;


proc means data=stats1 noprint;
        by shuf; var varx; output out =
stats2a sum = sumvarx;
        /*proc print data = stats2;*/
        run;
proc means data=stats1 noprint;
        by shuf; var ss; output out =
stats2b sum = sumss;
        run;
data stats2;
        merge stats2a stats2b; by shuf;
        title 'stats2: stats for
sumstats and mse';
      /*  proc print data = stats2; */
        run;
data vardat;
        merge stats1 stats2; by shuf;
        if &tops = 1 then varrat =
varx/sumvarx; else varrat = ss/sumss;
run;
        title 'vardat: variances of
shuffled samples along with sums and
ratios';
        /*  proc print data = vardat; */
        run;
data vardat12;
        set vardat;
proc sort data = vardat12;by ssize;
        title 'Summarization of
Variance ratios by Sample Size';
proc means data= vardat12; by ssize;
        var varrat;
        run;


/* START P-VALUE CALCULATIONS AND
OUTPUT BY SAMPLE SIZE  */



data adwork;
        set vardat;
        proc means noprint;
        by shuf ssize; var varrat;
output out = admax max = maxrat min =
minrat;
        /* proc sort data = admax ; by
maxrat;*/
        title 'admax: admax
distribution';
        /* proc print data = admax; */
run;
```

```
data pvaldat;
        merge admax0 admax; by shuf
ssize;
        if maxrat > maxratbn then gx =
1; else gx=0;
        if minrat < minratbn then lx =
1; else lx = 0;
        title ' pvaldat: stuff for p-
values';
        /* proc print data = pvaldat;
*/
        run;
data pvaldats;
        set pvaldat;
        proc sort; by ssize;
        run;
        proc means data=pvaldats
noprint;
        by ssize; var gx; output out =
ngmxdat sum = ngmx;
data pvouthi;
        set ngmxdat;
        pvalhi = (ngmx+1)/(&ns+1);
critpval = &alpha/(2*&kn);
        title "emprical p-values hi
with critical p-value using method
&tops";
        title2 'method 1 is the sum of
variances method';
        title3 'method 2 is the mse
method';
        proc print data = pvouthi;
        proc means data=pvaldats
noprint;
        by ssize; var lx; output out =
ngmndat sum = ngmn;

data pvoutlo;
        set ngmndat;
        pvallo = (ngmn+1)/(&ns+1);
        critpval = &alpha/(2*&kn);
        title "empirical p-values low
with ctitical p-values using method
&tops";
        title2 'method 1 is the sum of
variances method';
        title3 'method 2 is the mse
method';
        proc print data=pvoutlo;
        run;
 /***************************
        END P-VALUE CALCULATIONS AND
OUTPUT BY SAMPLE SIZE
***************************/

/***************************
        DETERMINE QUANTILES OF ADMIN
AND ADMAX DISTRIBUTIONS
***************************/


proc rank data=pvaldats    out=rkadmin;
by ssize;
        var minrat;
```

```
        ranks rminrat;  title
'rkadmin:';
      /* proc print data = rkadmin;*/
run;




data critlow;
        set rkadmin;
        ranklow = &ns-floor((&ns+1)*(1-
&alpha/(2*&kn))-1)-1;
        if rminrat > ranklow then
delete; if rminrat < ranklow then
delete;
        lowcrit = minrat;
        title 'crit low';
      /* proc print data = critlow; */
run;




proc rank data=pvaldats    out=rkadmax;
by ssize;
        var maxrat ;
        ranks rmaxrat;
        title 'rkadmax:  ranked admax';
        /*proc print data = rkadmax; */
run;




data crithi;
        set rkadmax;
        rankhi = &ns-
floor((&ns+1)*(&alpha/(2*&kn))-1);
        if rmaxrat > rankhi then
delete; if rmaxrat < rankhi then
delete;
        hicrit = maxrat;
        title 'crithi: crit high';
      /*  proc print; */
run;




data critdum1;
        merge crithi critlow;
        codex =1;
        title 'critdum1: critcial
values';
      /*  proc print; */
run;
data c2;set critdum1;codex=2;
data c3;set critdum1;codex=3;
data c4;set critdum1;codex=4;
data c5;set critdum1;codex=5;
data c6;set critdum1;codex=6;
data c7;set critdum1;codex=7;
data c8;set critdum1;codex=8;
data c9;set critdum1;codex=9;
data c10;set critdum1;codex=10;
data c11;set critdum1;codex=11;
data c12;set critdum1;codex=12;
data c13;set critdum1;codex=13;
data c14;set critdum1;codex=14;
data c15;set critdum1;codex=15;
data c16;set critdum1;codex=16;
data c17;set critdum1;codex=17;
data c18;set critdum1;codex=18;
data c19;set critdum1;codex=19;
data c20;set critdum1;codex=20;


data critvals;
        set critdum1 c2 c3 c4 c5 c6 c7
c8 c9 c10 c11 c12 c13 c14 c15 c16 c17
c18 c19 c20;
        if codex > &k then delete;
        title 'critvals: critical
values';
     /*   proc print data=critvals;
var  rankhi hicrit ranklow lowcrit ;
*/


/*********************************
    DETERMINE DECISION LINES
*********************************/

data owork1;
        set oshufrep;
        if shuf > 1 then delete;
        title 'owork1: original data
with sums';
        /* proc print data=owork1; */
data owork2;
        merge owork1 critdum1;
        by ssize;
        if &tops = 1 then do;
                UDL = sumvar0*hicrit;
                LDL = sumvar0*lowcrit;
                CL =   sumvar0/&k;
                stdev = sqrt(varbyt0);
                UDLs = sqrt(UDL);
                LDLs = sqrt(LDL);
                CLs = sqrt(CL);
        end;
        else do;
                UDL = (sumss0/(ssize-
1))*hicrit;
                LDL = (sumss0/(ssize-
1))*lowcrit;
                CL =   mse0;
                stdev = sqrt(varbyt0);
                UDLs = sqrt(UDL);
                LDLs = sqrt(LDL);
                CLs = sqrt(CL);
        end;
        title "decision limits using
method &tops";
        title2 'method 1 is the sum of
variances method';
        title3 'method 2 is the mse
method';
      proc print; var &classvar ssize
rankhi ranklow varbyt0 ldl cl udl;
```

```
        run;



/******************************
    OUTPUT ANOMV DECISION CHART
******************************/
 proc gplot data=owork2 ;

        plot stdev*&classvar=4
                ldls*&classvar=1
                cls*&classvar=2
                udls*&classvar=3
                /overlay
                haxis=axis2
                /* annotate=bars */
                legend;
        symbol1 c=BLUE,i=join, l=14,
v=none;
        symbol2 c=BLUE, i=join, l=1,
v=none;
        symbol3 c=BLUE, i=join, l=2
v=none;
        symbol4 c=BLACK, i=none,
v=star;
        axis2 order=(1 to &k by 1)
offset=(2) label=(h=1.5);
        title1 "RANDANOMV Decision
Chart for &varname using Method &tops";
        title2 "Alpha = &alpha and &ns
Permutation Shuffles";
        title3 "Standard Deviation
Plotted";
run;

%mend ranomv;
/*********************
MACRO
********************/
%ubranomv(k=5,n=10,alpha=0.05,ds=data1,
varname=bcount,classvar=treat,ns=1000,t
ops=2,kn=3);
```

DOWNLOADING THE SAS® PROGRAM

The source code can be downloaded from the University of North Florida Center for Research and Consulting in Statistics web page (www.unf.edu/coas/math-stat/CRCS) as technical report #011502.

REFERENCES

A. J. Bernard and P. S. Wludyka, "Robust *I*-Sample Analysis of Means Type Tests for Variances", *Journal of Statistical Computation and Simulation*, 2001, Vol. 69, pp 57-88.

P. Sa and P. S. Wludyka, "A Robust *I*-Sample Analysis of Means Type Test for Variances for Unbalanced Designs", to appear. See technical report #101501 at (www.unf.edu/coas/math-stat/CRCS) for preliminary Monte Carlo results.

P. S. Wludyka, "Using SAS® to Perform Robust I-Sample Analysis of Means Type Tests for Variances", *Conference Proceedings Southeast SAS Users Group*, Charlotte, North Carolina, 2000, pp 537-542.

P. S. Wludyka, "Using SAS® to Perform the Analysis of Means for Variances Test", *Conference Proceedings Southeast SAS Users Group*, Mobile, Alabama, 1999, pp 420-425.

P. S. Wludyka and P. R. Nelson, "An Analysis of Means Type Test for Variances from Normal Populations," *Technometrics*, Vol. 39, No. 3, 1997, pp 274-285..

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Peter Wludyka
Associate Professor of Statistics
Director of the Center for Research and Consulting in Statistics
University of North Florida
Jacksonville, Florida
Work Phone: 904-620-1048
Fax: 904-620-2818
Email: pwludyka@unf.edu
Web: www.unf.edu/coas/math-stat/~pwludyka

Ping Sa
Associate Professor of Statistics
University of North Florida
Jacksonville, Florida
Work Phone: 904-620-1047
Fax: 904-620-2818
Email: psa@unf.edu