Paper ST11

Using SAS® to Estimate Variance by the Jackknife Method

Thomas Mule, U.S. Census Bureau, Washington, D.C.

ABSTRACT

Calculating variances for estimates from survey sample data usually can not be done by using traditional SAS® statistical procedures like MEANS® or REG® . These traditional procedures assume that the data is drawn from a simple random sample. Most survey samples involve clustering, stratification and unequal probabilities of selection. There are two methods which can be used to account for these when estimating variance: Taylor Expansion or Repeated Replication. SAS® introduced the SURVEYMEANS® and SURVEYREG® procedures starting in version 8 which uses the Taylor Expansion methodology. For analysts who want to use repeated replication methods like the jackknife, there is currently no SAS® procedure to do this. I present a SAS® macro which uses the jackknife methodology to estimate the variance for totals and ratios.

INTRODUCTION

In order to analyze sample survey data, estimates of sampling variance are needed. You need them to determine the precision of the estimates you are investigating. Replication methods like the jackknife provide a way to estimate the variance of your estimator when you have complex sample designs and/or nonlinear estimators. They provide flexibility in being able to estimate the sampling variance of totals, ratios and other nonlinear estimators**.**

The complex sample designs include using clustering, unequal probabilities of selection and stratification. For many populations, we can not afford visits to *n* units drawn randomly from the entire area. Clustering reduces the cost of data collection but can increase variance based on the degree of homogeneity of the elements with the cluster. The sample design can visit all or a subsample of the units in the cluster or primary sampling unit (PSU). Subsampling minimizes the amount of increase in variance due to homogeneity but requires more clusters to reach a fixed sample size. Also by clustering, the elements in your final sample can not be assumed to be indep\endent and identically distributed.

Unequal probabilities of selection allow us to sample certain sampling units at different rates. One reason to do this is that by sampling more of certain units, we can increase the sample size for subdomains of interest. One example is sampling blocks in Hawaii at a higher rate if we were interested in estimating a characteristic for the Hawaiian and Native Islander population. The goal is to have a lower variance estimate because of the increase in sample size. However, the unequal probabilities of selection lead to weight variation which leads to an increase in variance. The estimate of variance should be lower if the benefit of the increase in sample size is greater than the loss because of weight variation.

Stratification involves grouping the sampling units on your frame. You would like these groups to be homogenous with the measure you are trying to estimate. This measure is usually unknown at the start of the survey so you can make the groups homogenous with a measure you already have and you believe that measure is associated with the measure you are trying to estimate. An example is trying to measure income for a survey in 2003. You can use income measures from the 2000 Long Form to group your sampling units on your frame.

Background

The jackknife is a subsampling replication technique. The jackknife derives estimates of the parameter of interest from each of several subsamples of the full sample and then estimates the variance of the full sample estimator from the variability between the subsample estimates.

The estimator of variance using the jackknife methodology is:

$$\left(\frac{NPSU-1}{NPSU}\right)\sum_{k}^{NPSU}\left(\hat{\theta}_{(k)}-\hat{\theta}\right)^2 \quad (1)$$

where NPSU is the number of PSUs selected,

$\hat{\theta}_{(k)}$ is the estimate of the parameter from the kth subsample and

$\hat{\theta}$ is the estimate of the parameter from the full sample.

The jackknife creates subsamples by deleting one PSU at a time and multiplying the sampling weights of the other PSUs by NPSU / (NPSU-1).

For linear estmates, the estimate of the parameter from the kth subsample when the kth PSU is deleted is:

$$\hat{\theta}_{(k)}=\left(\frac{NPSU}{NPSU-1}\right)\left(\hat{\theta}-\hat{\theta}_k\right) \quad (2)$$

Wolter (1985) shows that applying the jackknifing techniques to linear estimators simply reproduces the textbook variance estimators in most cases. The benefit lies in the variance estimation for nonlinear statistics.

For nonlinear estimates like the ratio, Y / X, the same formula in equation (1) can be used. For estimating the ratio for the subsample, equation (2) can be applied to Y and X individually and the ratio of the two subsample estimates can be calculated. This methodology can be used for other nonlinear estimates besides the ratio without having to know the Taylor Expansion of the estimate.

The estimates from the subsamples and full samples use the sampling weights which are the inverse of the probability of selection. This accounts if elements have unequal probabilites of selection. Wolter (1985) shows how the above estimator can be extended to handle stratification and when the primary sampling unit is a cluster. For multistage cluster samples, the jackknife technique is usually applied at the PSU level.

USING SAS® TO ESTIMATE JACKKNIFE VARIANCE

My SAS® programs estimate the unstratified jackknife variance. This accounts for the clustering by applying the methods at the

PSU level and using the the sampling weights to account for the unequal probabilities of selection. It does not account for any stratification in the sample design but can easily be extended to account for it.

I wrote two SAS® programs to do this. The first program LAUNCH_JACK.SAS allows the user to specify:

- the input data set,
- specify the PSU or cluster variable,
- estimate variance of specified variables,
- estimate variance of linear and nonlinear estimates,
- estimate variance overall or by specified subdomains,
- create a permanent output data set for results.

When the first program is submitted, it calls the second program. The second program, JACK.SAS, uses the information specified by the user and macros to generate the SAS® code to estimate the variances requested and create the output data set of the results.

The code for both programs is listed at the end of this paper. A TEST data set is in the LAUNCH_JACK program. The LAUNCH_JACK program macro variables have been specified to estimate the variance of the proportion of persons employed by Minority/Non-minority.

EXAMPLE

Table 1 shows how the methodology estimates the variance of a ratio. This example estimates the ratio Y/X based on a sample of two clusters.

CONCLUSION

My paper shows how SAS® can be used to calculate estimates of variance using the jackknife methodology. By using SAS Macros, I've been able to write a general program that estimates variance based on variables, linear and nonlinear estimates specified by the user**.**

REFERENCES

Wolter, K. (1985), *Introduction to Variance Estimation*, Springer-Verlag, New York.

CONTACT INFORMATION

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

Thomas Mule
Census Bureau
Building 2 Room 2505
Suitland, MD
301 457 8322
vincent.t.mule.jr@census.gov

Table 1: Example of How To Estimate Variance of a Ratio

PSU or Cluster	Y	X	Y/X (Full Sample)	Y(k) (kth Subsample Estimate)	X(k) (kth Subsample Estimate)	Y/X(k) kth Subsample Estimate	$(Y/X(k) - Y/X)^2$
1	80	100		2× (140-80)=120	2× (200-100)=200	120/200=.6	$(.6-.7)^2$ =.01
2	60	100		2× (140-60)=160	2× (200-100)=200	160/200=.8	$(.8-.7)^2$ =.01
Total	140	200	.70				
Var(Y/X)		½(.01 + .01) = .01			Standard Error (Y/X)	sqrt(.01) = .1	

```
/*  Launch_jack.sas Program*/

option mprint;
/*** File with SAS macros for estimating
Jackknife variance based on data, variables
and estimators specified by the user ***/

filename jack 'jack.sas';

/* TEST DATA */

data temp;

input clust person minority employed total;
cards;
1 1 1 40 60
1 2 2 20 40
1 3 1 50 70
1 4 2 25 45
1 5 1 20 25
2 1 2 30 50
2 2 1 30 50
2 3 2 60 65
2 4 1 20 35
2 5 2 50 70
;

/*  Assign Libname for Final Data Set */

libname here 'c:\SAS'; /* OR SPECIFY ANOTHER
DIRECTORY */

/*  Input DataSet */

%let dset = temp;

/* Permanent Output Data Set for Results*/

%let fdata = test;

/*  Delete a Primary Sampling Unit (Cluster
ID Variable) for Jackknife method*/
%let cluster = clust;

/*  Number of Variables */
%let vnum = 2;

/*  Variables */
%let var1 = employed;
%let nvar1 = Persons Employed;
```

```
%let var2 = total;
%let nvar2 = Total Persons;

/*  Are you estimating any combinations of
       these variables
est = 1 (yes), est = 0 (no) */

%let est = 1;

/* If Yes, How Many? */

%let nest = 1;

/* For each, list a name identifier
(ESTNAMEn), the estimator (ESTn) based on the
variables listed above and the replicate
estimator (RESTn): placing an "r" infront
of each &VARn in the ESTn equation.  */

%let estname1 = Employment Rate;

%let est1 = &var1 / &var2;
%let rest1 = r&var1 / r&var2 ;

/*  Subdomains
1 = YES, Estimates by subdomains
0 = No, Overall Estimates  */

%let subdom = 1;

/* If yes, how many subdomains */
%let nsubdom = 1;

/* If Yes, What is the Subdomain Variable */
%let subvar1 = minority;

%inc jack;

/*** END OF LAUNCH_JACK.SAS PROGRAM***/

/* JACK.SAS PROGRAM */
/* Program called by the "%inc jack"
statement when the LAUNCH_JACK.SAS program is
run */

/*  Macro A prints the variables which need
to be summed in the Proc Summary statement in
the Main Macro. The user specifies var1-varn
in the Launch program */

%macro a;
       %do i = 1 %to &vnum;
              &&var&i
       %end;
%mend;

/*  Macro C prints the output variables for
the Proc Summary in the main macro for the
cluster-level summation of the variables.
There is one variable  for each variable
specified by the user  */

%macro c;
       %do i = 1 %to &vnum;
              c&&var&i
       %end;
%mend;

/*  Macro SEST is called to generate  full
sample estimate for the Estimators specified
by the user on the SUMS2 data set. These full
sample estimates will  be merged onto the J3
data set.   */

%macro sest;
       %if &est = 1 %then %do;
              %do i = 1 %to &nest;
/*  Calculate Full Estimate for estimate i */
                     est&i =  &&est&i;
              %end;
       %end;
%mend;

/*  Macro RSUM calculates replicate estimates
for each variable and the Estimators
specified by the User during the SUMC4 data
step */

%macro rsum;
       %do i = 1 %to &vnum;

       r&&var&i = (&&var&i - c&&var&i ) *
              (&nclust / (&nclust - 1));
       %end;
       %if &est = 1 %then %do;
              %do i = 1 %to &nest;
/*  Calculate Full Estimate for estimate i */
                     est&i =  &&est&i;
/*Calculate Replicate Estimate for estimate i
*/
                     rest&i = &&rest&i ;
              %end;
       %end;
%mend;

/*  Macro DIFF calculates the difference
between the replicate estimate and the full
sample estimate for all of the variables and
estimators specified by the user during the
SUMC4 data step */

%macro diff;
       %do i = 1 %to &vnum;
              ds&&var&i =
              (r&&var&i - &&var&i ) ;
       %end;
       %if &est = 1 %then %do;
              %do i = 1 %to &nest;
                     dest&i =
                     (rest&i - est&i) ;
              %end;
       %end;
%mend;

/*  Macro SDIFF squares the difference
calculated in the DIFF macro.  This is done
for all variables and estimators specified by
the User during the SUMC4 data step.  */

%macro sdiff;
       %do i = 1 %to &vnum;
              ds2&&var&i = ds&&var&i ** 2;
       %end;
       %if &est = 1 %then %do;
              %do j = 1 %to &nest;
                     dest2_&j = dest&j **2;
              %end;
       %end;
%mend;

/*  Macro J1VAR lists the squared components
as variables for the Proc Summary step.  This
Proc Summary sums squared differences for the
JK variance calculation.  These squared
components were just calculated in the SUMC4
data step */

%macro j1var;
       %do i = 1 %to &vnum;
              ds2&&var&i
       %end;
       %if &est = 1 %then %do;
              %do j = 1 %to &nest;
                     dest2_&j
```

```
            %end;
        %end;
%mend;

/*  Macro J1SUM lists the summation variables
on the output data step from the Proc Summary
step.  This Proc Summary step sums the
squared difference which is needed to
estimate JK variance */

%macro j1sum;
        %do i = 1 %to &vnum;
            sds2&&var&i
        %end;
        %if &est = 1 %then %do;
            %do j = 1 %to &nest;
                sdest2_&j
            %end;
        %end;
%mend;

/*  Macro SE estimates the Standard Error of
the variables and estimators specified by the
user.  The standard error equation for JK
variance uses the number of clusters and the
sum of squared differences */

%macro se;
        %do i = 1 %to &vnum;
         se_s&&var&i  =
         sqrt (((&nclust - 1)/(&nclust)) *
                 sds2&&var&i );
        %end;
        %if &est = 1 %then %do;
             %do j = 1 %to &nest;

                se_dest&j =
           sqrt(((&nclust-1)/(&nclust)) *
              sdest2_&j );
              %end;
        %end;
%mend;

/*  Macro CV estimates the coefficient of
variation for each variable and estimattor
specified by the user.  The CV is the
standard error divided by the estimate. */

%macro cv;
        %do i = 1 %to &vnum;

            cv_s&&var&i =
            se_s&&var&i / &&var&i;
        %end;
        %if &est = 1 %then %do;
             %do j = 1 %to &nest;

          cv_dest&j = se_dest&j / est&j ;

             %end;
        %end;
%mend;

/*  Macro OUT1 is called during creation of
the final permanent output data set.  The
input data set has one record for each
subdomain (or 1 record if no subdomains).
There are multiple variables for each
variable and estimate  specified by the user.
This macro creates a record on the permanent
data set that has one record for each
subdomain/estimate.

Each record then has 5 variables:
name ("English explanation given by user")
varname (variables used for this estimate)
est (Estimate)
se_est (Standard Error of the Estimate)
cv_est (CV of the Estimate) */

%macro out1;
        length name $150. varname $70.;
        %do i = 1 %to &vnum;

            id = id + 1;

            name = "&&nvar&i" ;
            varname = "&&var&i";
            est = &&var&i;
            se_est = se_s&&var&i;
            cv_est = cv_s&&var&i;

            %if &subdom = 1 %then %do;
                %do j = 1 %to &nsubdom;

                subdom&j = &&subvar&j ;

                %end;
            %end;
            output here.&fdata;
        %end;
        %if &est = 1 %then %do;
            %do k = 1 %to &nest;

                id = id + 1;

                name = "&&estname&k";
                varname = "&&est&k";
                est = est&k;
                se_est = se_dest&k;
                cv_est = cv_dest&k;

                output here.&fdata;
            %end;
        %end;
%mend;

/*  Main Macro */
/*  This is the main macro of the JK variance
program */

%macro main;

/*  Determine number of primary sampling
units and assign it to the macro variable
NCLUST */

proc freq data=&dset noprint;
        table &cluster /list missing
out=nclust (keep = &cluster ) ;
run;

/*  The following assigns the number of
records in the NCLUST data set to the macro
variable NCLUST without reading the whole
dataset. */

%let dsid = %sysfunc(open(nclust));
%let nclust = %sysfunc(attrn(&dsid,nobs));
%let return = %sysfunc(close(&dsid));

/*  Proc Summary to sum the variables
specified by the user to the cluster-level.
The macro variable SUBDOM which was
specified by the User is used to determine if
Subdomains have been requested.  If yes, then
the summations will be at the
cluster/subdomain level. Macro A is called to
complete the VAR statement by listing  the
variables specified by the User.
Macro C is called to complete the OUTPUT
statement by listing the variable names for
the cluster-level (or cluster/subdomain
level) totals. */
```

```
proc summary data=&dset nway;

%if &subdom = 0 %then %do;
        class &cluster;
%end;
%else %if &subdom = 1 %then %do;
        class
        %do i = 1 %to &nsubdom ;
                &&subvar&i
        %end;
        &cluster;
%end;
        var %a ;
        output out=sumc (drop = _freq_ _type_)
        sum = %c ;
run;

/*  SUMC2 datastep adds a dummy variable
equal to 1 for each record.  This is for
merging purposes later in the program */

data sumc2;
        set sumc;
        dummy = 1;
run;

/*  Proc Summary to sum the variables
specified by the user to the overall level.
The macro variable SUBDOM which was
specified by the User is used to determine if
Subdomains have been requested.  If yes, then
the summations will be at the
overall/subdomain level.  Macro A is called
to complete the VAR statement by listing  the
variables specified by the User.  Macro S is
called to complete the OUTPUT statement by
listing the variable names for the overall
(or overall/subdomain level) totals. */

proc summary data=&dset nway;
%if &subdom =1 %then %do;
        class
        %do i = 1 %to &nsubdom ;
                &&subvar&i
        %end;
        ;
%end;
        var %a ;
        output out=sums (drop = _freq_ _type_)
        sum = %a ;
run;

/*  SUMS2 data step 1)assigns the dummy
variable of 1 to each observation, 2)
calculates the full sample estimate for the
variables and estimators specified by the
User.  Macro SEST is used to calculate the
full sample estimate */

data sums2;
        set sums;
        dummy = 1;
        %sest;
run;

/*  If Subdomains have been requested then we
need to sort   the overall (SUMS2) data set
by the Subdomain variables specified  by the
User.  Macro variable NSUBDOM (number of
subdomain variables) is used in the macro do
loop to the list the subdomain variables
specified by the user */

%if &subdom = 1 %then %do;
        proc sort data=sums2;
                by
        %do i = 1 %to &nsubdom;
                &&subvar&i
        %end;
        dummy;
        run;
%end;

/*  If Subdomains have been requested then we
need to sort   the cluster (SUMC2) data set
by the Subdomain variables specified by the
User.  Macro variable NSUBDOM (number of
subdomain variables) is used in the macro do
loop to the list the subdomain variables
specified by the user */

%if &subdom = 1 %then %do;
        proc sort data=sumc2;
                by
        %do i = 1 %to &nsubdom;
                &&subvar&i
        %end;
        dummy;
        run;
%end;

/*  SUMC3 data step merges together the
cluster (SUMC2) and overall data sets (SUMS2)
If Subdomains have not been specified then we
can merge by the dummy variables.
If Subdomains have been specified then we use
the NSUBDOM macro variable in a macro do loop
to list subdomain variables in the merge. The
dummy variable is placed at the end.  Note:
The Dummy variable is probably not needed
especially for the No Subdomain case but I
like to control my merges with a "by"
statement.  */

data sumc3;
        merge sumc2 (in=a) sums2 (in=b);
%if &subdom = 0 %then %do;
        by dummy;
%end;
%else %if &subdom = 1 %then %do;
        by
        %do i = 1 %to &nsubdom;
                &&subvar&i
        %end;
        dummy;
%end;
        if a & b then do;
                output sumc3;
        end;
run;

/*  SUMC4 data step performs the calculations
for the JK variance.  1)  I calculate a
replicate estimate of each variable and
estimator specified by the User in the RSUM
macro,  2)  I calculate the difference of the
replicate estimate and the full sample
estimate for each variable and estimator
specified by the user in the DIFF macro,
3)  I calculate the squared difference of the
replicate estimate and the full sample
estimate for each in the SDIFF macro. */

data sumc4;
        set sumc3;
        /*  Replicate Estimates */
        %rsum;
        /*  Difference Summations*/
        %diff;
        /*  Squared Difference */
        %sdiff;
run;
```

```
/*  Proc Summary to sum the squared
differences calculated for each variable in
the SUMC4 data step.  The SUBDOM and NSUBDOM
macro variables are used to list any
subdomain variables in the class statement
if subdomains have been requested.  I used
the Macro J1VAR to list the variables that
represent the squared differences for each
variable and estimator specified by the User.
I use the Macro J1SUM to list the output
variables for the summation on the J1 output
data set */

proc summary data=sumc4 nway;

%if &subdom = 1 %then %do;
       class
       %do i =1 %to &nsubdom;
              &&subvar&i
       %end;
       ;
%end;
       var %j1var ;
       output out=j1 (drop = _type_ _freq_)
       sum = %j1sum ;
run;

/*  J2 data step adds the dummy variable
       to each record */
data j2;
       set j1;
       dummy = 1;
run;

/*  If Subdomains have been requested then we
need to sort the variance summation (J2) data
set by the Subdomain variables specified by
the User.  Macro variable NSUBDOM (number of
subdomain variables) is used in the macro do
loop to the list the subdomain variables
specified by the user */

%if &subdom = 1 %then %do;
       proc sort data=j2;
              by
       %do i = 1 %to &nsubdom;

              &&subvar&i
       %end;
       dummy;
       run;
%end;

/* J3 data step merges the variance
summations in the J2 data set and the full
sample estimates from the SUMS2 data set
The SUBDOM and NSUBDOM macro variables are
used to list any subdomain variables in the
by statement if subdomains have been
requested. J3 data set estimates the standard
error and coefficient of variation for each
variable and estimate specified by the user.
Macro SE estimates the standard error
Macro CV estimates the coefficient of
variation */

data j3;
       merge j2 (in=a) sums2 (in=b);
%if &subdom = 0 %then %do;
       by dummy;
%end;
%else %if &subdom = 1 %then %do;
       by
       %do i = 1 %to &nsubdom;
              &&subvar&i
       %end;
       dummy;
%end;
       if a & b then do;
       /*  Calculate Standard Error */
              %se;
       /*  Calculate CV */
              %cv;
              output j3;
       end;
run;

/*  HERE.&fdata data step creates a permanent
data set of the results.  The user specifies
the FDATA macro variable.  The variables on
the output data set are
       Name:  Label Name specified by User
       Varname:  Variables or combination of
       variables for this estimate
       EST: Estimate
       SE_EST:  Standard error of estimate
       CV_EST:  CV of estimate
       ID:  Each estimate will have the same
       ID for each subdomain value(s).

If Subdomains are specified the subdomain
variables are also listed on the output data
set.  I use the Macro OUT1 to do create the
individual records */

data here.&fdata
%if &subdom = 0 %then %do;
       (keep = name varname est se_est cv_est
id) ;
%end;
%else %if &subdom = 1 %then %do;
       (keep =
       %do i = 1 %to &nsubdom;
              &&subvar&i
       %end;
       name varname est se_est cv_est id) ;

%end;
       set j3;
       %if &subdom = 1 %then %do;
              by
              %do i = 1 %to &nsubdom;
                     &&subvar&i
              %end;
       ;
       if first.&&subvar&nsubdom then id = 0;
       %end;

       retain id(0);
       %out1;
run;

/* Print the results from the Permanent data
set.  If subdomains have been requested then
the SUBDOM and NSUBDOM macro variables are
used to print the results by subdomains */

proc print data=here.&fdata;
       title1 'Estimates, Standard Errors and
CVs      in Permanent Data Set';
       format est 15.6 se_est 15.8;
%if &subdom = 1 %then %do;
       by
       %do i = 1 %to &nsubdom;
              &&subvar&i
       %end;
       ;
       title2 'by subdomains';
%end;
run;
%mend;

%main;
/***END of JACK.SAS PROGRAM***/
```