Abstract

Prior to approval of a benchmark survey for Federal Reserve System consumer credit components, questions arise as to how much error can be expected in the estimates. The amount of error depends on the sample size, stratification, and method of estimation. SAS Macros were written to use Neyman allocation over a number of possible sample sizes and produce standard errors using ratio estimation.

Background

The Federal Reserve System collects consumer credit data from a monthly sample of about 300 insured commercial banks. Reporting detail includes 'auto', 'revolving', 'mobile homes' and 'other' credit categories. Estimates of the corresponding aggregates for the 14,500 bank universe have been produced by separate ratio estimators, using Report of Condition benchmark data which is collected quarterly from the universe. However, changes in the 1984 Report of Condition eliminated autos, mobile homes and other, so that banks just report revolving and total credit. The importance of the missing items led Board staff to consideration of a benchmarking sample to produce estimates.

Objective

Our objective is to estimate the amount of error expected in the benchmark estimates of consumer credit components over sample sizes ranging from 300 to, say, 5000 banks. The 300 level would imply using the present monthly sample for benchmarking purposes. A larger sample will be requested if it lowers the error significantly and is within cost constraints. We assume that the level of error is dependent on:

a) the sample size
b) the stratification

c) method of estimation (here we use ratio estimation).

After specifying a stratification variable and strata limits, we use the macros to allocate sample sizes to strata and compute standard errors with plots. The design that appears best is judgmentally taken as the stratification over which to estimate.

Method of Estimation

To estimate the universe 'automobile', 'mobile homes' and 'other' credit components from the benchmark sample we assume that ratio estimation will be used over the stratification selected. As covariate X, we take the difference between total consumer credit and revolving credit, as all banks report these items. Then, each stratum h is estimated as:

\[ Y_h(k) = (y_h(k)/x_h) x_h \]

where

\[ y_h(k) = \text{sample total item } k \]
\[ x_h = \text{sample covariate} \]
\[ X_h = \text{universe covariate} \]

The variance of the estimator for item k, stratum h is:

\[ \text{Var}_h(k) = N_h(N_h - n_h) \sum_{i=1}^{N_h} \left( y_{hi}(k) - \frac{1}{N_h} \sum_{i=1}^{N_h} y_{hi}(k) \right)^2 \]

where

\[ N_h = \text{number of universe banks in stratum } h \]
\[ n_h = \text{number of sample banks in stratum } h \]

The sample size \( (n_h) \) will depend upon the total sample size \( n = \sum n_h \). For each hypothetical sample size \( n \) we will use a best possible allocation to strata (Neyman allocation).
More Than 100 Percent Allocation

The allocation of a sample size to strata h is given by the equation:

\[ n_h = \frac{n N_h S_h}{\sum_{h=1}^{L} n_h S_h} \quad h=1, \ldots, L. \]

We choose \( S_h = \left( \frac{1}{N_h} \sum_{i=1}^{N_h} x_{hi} \right)^2 / (N_h - 1) \)

where

\( Y_{hi} = \text{automobile credit for } i\text{th bank stratum } h \)

\( x_{hi} = \text{covariate for } i\text{th bank stratum } h \)

\( N_h = \frac{1}{N} \sum_{i=1}^{N} x_{hi} \)

We stratify the universe based on the size of a financial variable. As an example of the type of universe, the table below shows the current monthly estimation stratification based on total assets for December 1983:

<table>
<thead>
<tr>
<th>Assets ($ millions)</th>
<th>Number of Banks</th>
<th>Total Assets ($ billions)</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0,15)</td>
<td>3186</td>
<td>30.4</td>
<td>1.5</td>
</tr>
<tr>
<td>[15,30)</td>
<td>3658</td>
<td>80.1</td>
<td>4.0</td>
</tr>
<tr>
<td>[30,45)</td>
<td>2172</td>
<td>80.0</td>
<td>4.0</td>
</tr>
<tr>
<td>[45,100)</td>
<td>3162</td>
<td>206.5</td>
<td>10.2</td>
</tr>
<tr>
<td>[100,160)</td>
<td>976</td>
<td>121.7</td>
<td>6.0</td>
</tr>
<tr>
<td>[160,350)</td>
<td>704</td>
<td>161.7</td>
<td>8.0</td>
</tr>
<tr>
<td>[350,1000)</td>
<td>340</td>
<td>193.9</td>
<td>9.6</td>
</tr>
<tr>
<td>[1000,7000)</td>
<td>223</td>
<td>524.8</td>
<td>26.0</td>
</tr>
<tr>
<td>7000 &amp; over</td>
<td>31</td>
<td>621.0</td>
<td>30.7</td>
</tr>
</tbody>
</table>

The universe is very skewed with a large number of small banks and few large banks. The 254 largest banks hold 56.7 percent of total assets. When allocating a fixed sample size to this type of universe, theory will request a more than 100 percent allocation for one or more strata. So, in general, we consider the case where \( n_j > N_j \) for some \( j \in \{1, \ldots, L\} \). The following routine is taken from Cochran [1].

The optimum revised allocation is:

\[ n_j^* = \frac{n_j}{N_j} \left( N_j - n_j \right) \left( \frac{1}{N_j} \sum_{h \in \{1, \ldots, L\} \setminus j} n_h S_h \right) \]

If it happens that \( n_j > N_j \), change the allocation to

\[ n_j = n_j \quad n_h = (n - n_j - n_j) \frac{N_h S_h}{\sum_{h \in \{1, \ldots, L\} \setminus j} N_h S_h} \]

This process is continued until every \( n_h < N_h \). The allocations over strata are passed to the standard error macro and then a table and plot are generated.

We illustrate the process with the following code:

```
GOPTIONS DEVICE=CAL5200 UNIT=2 ROTATE;
%MACRO STRATIFY(DATE=, STRAT_VR=, SIZEGRPS=);
*--------------------------------------------
ACCESS HISTORICAL DATA BASE
VARIABLE DEFINITIONS
AUTOS = CONSUMER CREDIT FOR AUTOMOBILES
REVOLV = CONSUMER CREDIT FOR REVOLVING LNS
MH = CONSUMER CREDIT FOR MOBILE HOMES
OTHER = CONSUMER CREDIT FOR OTHER LOANS
TOTCLOAN = TOTAL CONSUMER CREDIT
ASSETS = TOTAL ASSETS
PROC ARCHIVE DD=RCR1&DATE OUT=UNIV;
ITEMS AUTOS REVOLV MH OTHER TOTCLOAN ASSETS
BANKNAME CITY STATE CHARTER DISTRICT;
DATA UNIV; SET UNIV;
LENGTH STRATSZE $ 8;
* COVARIANCE TERM;
COVAR = TOTCLOAN-REVOLV;
%LET SIZE = %SCAN(&SIZEGRPS, 1);
IF ASTRAT VR LT &SIZE * 1000
THEN STRATSZE = "&SIZE";
%LET SIZE = %SCAN(&SIZEGRPS, 2);
%LET COUNT = 2;
%DO %WHILE(&SIZE NE );
ELSE IF ASTRAT VR LT &SIZE * 1000
THEN STRATSZE = "&SIZE";
%LET COUNT = %EVAL(&COUNT + 1);
%LET SIZE = %SCAN(&SIZEGRPS, &COUNT);
%END;
ELSE STRATSZE = "INFINITY";
RUN;
*****************************************************************************;
* COMPUTE S(h) TERM FOR NEYMAN ALLOCATION;
PROC SORT DATA=UNIV; BY STRATSZE ;
```
data residualjmerge univ rterms;by stratsze;
  resid2=(autos-r*covar)**2;
run;

proc means noprint; by stratsze; var resid2;
output out = ss sum = ss n = n;
data ss;set ss; by stratsze;
  sh = sqrt(ss/(n-1));
proc print;
  *****************************************************;
run;
%end stratify;

%macro build(smplszs=);
* parse sample sizes;
%let count = 1;
%let sampsz = %scan(&smplszs, &count);
%do %while(&sampsz ne);
  * call neyman allocation;
  %neyalloc(ntot=);*
  * build dataset over all;
  * possible samples:
  run;
data neyalloc;set neyalloc;
  (keep=stratsze sampsz nh);
  proc append base=bigsamp data=neyalloc;
  run;
  proc print data=bigsamp;run;
  %let count = %eval(&count+1);
  %let sampsz = %scan(&smplszs, &count);
%end;
proc sort data = bigsamp; by stratsze;
%end build;

%macro neyalloc(ntot=);
  *****************************************************;
  * this routine calculates a neyman allocation
  * on a stratified sampling design. correcting
  * whenever a more than 100 percent allocation
  * is called for
  *****************************************************;
  %let lagntot = 0; * flag;
  data neyalloc;set ss; * bring in strata stats;
  ns = %sh;
  * weighting factor:
  nh = 0; * final sample size;
  ntot=0; * total sample size;
  sigmans=0; * sum(ns)*
  ntot2=0; * remaining sampling units;
  sigmans2=0; * current value of sum(ns);
  sampsize=ntot;
  *compute sum(ns);
  proc means noprint sum; var ns;
  output out=sums sum=sigmans;
  data_null;set sums;
  call symput('sigmans',sigmans);
  xdo until(&ntot eq &lagntot);
  %let lagntot = &ntot;
  data neyalloc;
  set neyalloc;
  (drop=sigmans ntot sigmans2 ntot2 end=eof);
  retain sigmans ntot sigmans2 ntot2 sampsize;
  if _n_ eq 1
    then do;
      ntot = &ntot;
      * initialize;
      ntot2= ntot;
      sigmans = &sigmans;
      sigmans2=sigmans;
      end;
    * neyman allocation;
      smalln = round(ntot * ns / sigmans);
      *equate to 0 if certainty stratum;
      if n eq 0 then smalln = 0;
      if smalln gt n
    then do;
      smalln = n;
      ntot2= ntot2- smalln;
      sigmans2= sigmans2- ns;
      n = 0;
    end;
    *noncertainty stratum;
    else do;
      ntot2= ntot2- smalln;
      n = n - smalln;
    end;
    nh = nh + smalln; * build sample size;
  output;
  *select last observation of ssuhr to get;
  *ntot and sigmans;
  if eof then do;
    call symput('ntot',ntot2);
    call symput('sigmans',sigmans2);
  end;
  run;
%end neyalloc;
%end build;

%macro stderror;
*get universe sums to calc std errors;
proc means data=univ noprint;
  by stratsze;
  var covar autos nh other;
  output out=sums
    sum = xcovar yautos ynh yother
    n = n;
run;
752
DATA VARS;
MERGE UNIV SUMS; BY STRATSZE;
RETAIN SAUTOS SMH SOTHER 0;
KEEP SAUTOS SMH SOTHER YAUTOS YMH YOTHER STRATSZE N;
ARRAY SUMS(I) SAUTOS SMH SOTHER;
ARRAY Y(I) AUTOS MH OTHER;
ARRAY YT(I) YAUTOS YMH YOTHER;
DO OVER SUMS; * VARIANCES;
SUMS = SUMS + (Y-(YT/XCOVAR)*COVAR)**2;
END;
IF LAST.STRATSZE THEN DO;
OUTPUT;
DO OVER SUMS;
SUMS = 0;
END;
END;
DATA CNCR; MERGE BIGSAMP VARS; BY STRATSZE;
RUN;
PROC SORT; BY SAMPSIZE STRATSZE; RUN;
DATA STDERR;
SET CNCR; BY SAMPSIZE STRATSZE;
RETAIN STDAUTOS STDMH STDOOTHER 0;
ARRAY Y(I) SAUTOS SMH SOTHER;
ARRAY STD(I) STDAUTOS STDMH STDOTHER;
ARRAY PER(I) PERAUTOS PERMH PEROTHER;
ARRAY YT(I) YAUTOS YMH YOTHER;
DO OVER Y;
STD = STD + (N*(N-NH)/(NH*(N-1)) * Y;
END;
IF LAST.SAMPSIZE THEN DO;
OUTPUT;
DO OVER STD;
STD = SQRT(STD);
PER = STD / YT;
END;
OUTPUT;
DO OVER STD;
STD = 0;
END;
END;
RUN;
%MEND STDERR;
%MACRO TABLES ( DATE = );
PROC PRINT DATA = STDERR LABEL SPLIT=*;
ID SAMPSIZE;
VAR STDAUTOS PERAUTOS STDMH STDOOTHER PEROTHER;
LABEL SAMPSIZE = 'SAMPLE SIZE' STDAUTOS = 'STD AUTOS' PERAUTOS = '1% OF AGGREGATE' STDMH = 'STD MH HOMES' PERMH = '1% OF AGGREGATE' STDOOTHER = 'STD OTHER' PEROTHER = '1% OF AGGREGATE';
TITLE1 "STANDARD ERRORS FOR CONSUMER CREDIT ITEMS";
TITLE2 "(AMOUNTS IN THOUSANDS)";
TITLE3 "OVER DIFFERENT SAMPLE SIZES";
TITLE4 "&DATE R/C DATA";
RUN;
%MEND TABLES;
%MACRO GRAPH;
DATA TEMP; SET STDERR;
ARRAY STD(I) STDAUTOS STDMH STDOOTHER;
STDERR = STDAUTOS;
ITEM = 'AUTOS';
OUTPUT;
STDERR = STDMH;
ITEM = 'MH';
OUTPUT;
STDERR = STDOOTHER;
ITEM = 'OTHER';
OUTPUT;
RUN;
SYMBOL1 I=JOIN V=+ L=1 C=BLACK;
SYMBOL2 I=JOIN V=+ L=2 C=BLACK;
SYMBOL3 I=JOIN V=+ L=4 C=BLUE;
PROC GAXPLOT DATA = TEMP;
PLOT STDERR*SAMPSIZE = ITEM;
RUN;
%MEND GRAPH;
%LET DATE 8312;
%STRATIFY ( DATE = &DATE,
STRAT VR = ASSETS,
SIZEGRPS ~ 15 30 45 100 160 350 1000 7000);
%BUILD ( SMPLSZS = 300 500 650 750 850 1000 1500 2500 5000 );
%STDERROR;
%TABLES ( DATE = &DATE);
%GRAPH;
### Table of Standard Errors

**STANDARD ERRORS FOR CONSUMER CREDIT ITEMS**  
(Amounts in thousands)  
Over different sample sizes  
8312 R/C Data

<table>
<thead>
<tr>
<th>SAMPLE SIZE</th>
<th>STDERR AUTOS</th>
<th>% OF AGGREGATE</th>
<th>STDERR MD HOMES</th>
<th>% OF AGGREGATE</th>
<th>STDERR OTHER</th>
<th>% OF AGGREGATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>1926944</td>
<td>0.137496</td>
<td>775922</td>
<td>0.459458</td>
<td>1488096</td>
<td>0.120713</td>
</tr>
<tr>
<td>500</td>
<td>1352264</td>
<td>0.096535</td>
<td>561267</td>
<td>0.329679</td>
<td>1071560</td>
<td>0.086923</td>
</tr>
<tr>
<td>650</td>
<td>1105851</td>
<td>0.078946</td>
<td>472164</td>
<td>0.277361</td>
<td>897751</td>
<td>0.072864</td>
</tr>
<tr>
<td>750</td>
<td>978562</td>
<td>0.068928</td>
<td>429138</td>
<td>0.249143</td>
<td>803594</td>
<td>0.063526</td>
</tr>
<tr>
<td>850</td>
<td>888226</td>
<td>0.062837</td>
<td>381417</td>
<td>0.224037</td>
<td>725740</td>
<td>0.058871</td>
</tr>
<tr>
<td>1000</td>
<td>772527</td>
<td>0.055149</td>
<td>335333</td>
<td>0.194669</td>
<td>639831</td>
<td>0.051983</td>
</tr>
<tr>
<td>1500</td>
<td>590928</td>
<td>0.039329</td>
<td>240419</td>
<td>0.131335</td>
<td>446213</td>
<td>0.037656</td>
</tr>
<tr>
<td>2500</td>
<td>348493</td>
<td>0.024878</td>
<td>152612</td>
<td>0.089729</td>
<td>297365</td>
<td>0.021805</td>
</tr>
<tr>
<td>5000</td>
<td>166019</td>
<td>0.011852</td>
<td>725172</td>
<td>0.042592</td>
<td>145617</td>
<td>0.013650</td>
</tr>
</tbody>
</table>

### Plot of Standard Errors

**STANDARD ERRORS FOR CONSUMER CREDIT ITEMS**  
(Amounts in thousands)  
Over different sample sizes  
8312 R/C Data
Producing the Sample

The macros presented here serve a dual purpose. They have helped in evaluating the analytical question of how much error can be expected. Then, after a sample size is chosen, the information they generate,
a) specification of a sampling universe
b) optimum number of banks to select from each stratum
is passed to a sampling routine and the benchmark sample is selected.

* SAS is the registered trademark of SAS Institute, Inc.

REFERENCES
