Paper 217-2013

A SAS® Macro for Generating a Set of All Possible Samples with Unequal Probabilities without Replacement

Alan Ricardo da Silva, Universidade de Brasília, Dep. de Estatística, Brazil
Igor Ferreira do Nascimento, Universidade de Brasília, Dep. de Estatística, Brazil

ABSTRACT

This paper considers listing all possible samples of size *n* with unequal probabilities without replacement in order to find the sample distribution. The main application of that is to estimate the Horvitz-Thompson (HT) estimator and possibly to know the shape of its sample distribution to construct confidence intervals. The algorithm computes all possible samples of the population, in contrast with PROC SURVEYSELECT which generates any samples of size *n*, but not all possible samples, and at the end it is possible to plot the sample distribution of the estimator. The equations are encoded in a SAS/IML Macro and the graphics are made using PROC GPLOT.

1. INTRODUCTION

The Horvitz-Thompson (HT) Estimator is the best-known general estimate of the population total for unequal probability sampling without replacement (Cochran, 1977). It is used mainly in finite populations and with small samples. For that, an auxiliary variable as a measure of size is required to permit an efficient estimation. If that measure of size is the same for all units then the Horvitz-Thompson estimator generates the same results of the simple or stratified random sampling.

The main purpose of this paper is to present a SAS® Macro to compute all possible samples of size *n* of a population of size *N* with unequal probabilities without replacement, allowing showing properties as expectation and variance of the estimators and its sample distribution. It is possible to use the PROC SURVEYSELECT to select units without replacement and with the probability proportional to size (PPS Method) and then use the PROC SURVEYMEANS to estimate the population total, but this method is a little bit different of the proposed by Horvitz and Thompson (1952). Besides, this work can help the sampling teaching because there are many calculations to compute HT estimates and using the algorithm presented here, this job become easier.

The paper is organized as follows. In Section 2, the theory about the Horvitz-Thompson Estimator and its relation with simple random sampling is given. In Section 3, a SAS® macro is presented and in Section 4 we introduce an illustration.

2 THE HORVITZ-THOMPSON ESTIMATOR

The Hansen-Hurwitz (HH) estimator (Hansen and Hurwitz, 1943) is used in finite populations for sampling *n* units of *N* by Simple Random Sampling with replacement (SRS). In this case, each drawn is independent and the probability of the *i*th unit is in the sample is p_i . The population total estimator of $\sum_{i=1}^{N} y_i$ is given by:

$$\hat{t}_{pwr} = \frac{1}{n}\sum_{i=1}^{n}\frac{y_i}{p_i} \tag{1}$$

and its variance is given by:

$$V(\hat{t}_{pwr}) = \frac{1}{n}\sum_{i=1}^{N}\left(\frac{y_i}{p_i} - \hat{t}_{pwr}\right)^2 p_i \tag{2}$$

An unbiased sample estimate, following Sarndal, Swensson, and Wretman (1992) is:

$$\hat{V}(\hat{t}_{pwr}) = \frac{1}{n(n-1)}\sum_{i=1}^{n}\left(\frac{y_i}{p_i} - \hat{t}_{pwr}\right)^2 \tag{3}$$

The Horvitz-Thompson estimator (Horvitz and Thompson, 1952) is used in sampling without replacement and it is a generalization of Hansen-Hurwitz estimator. The unbiased HT estimator for the population total is:

$$\hat{t}_{pwr} = \sum_{i=1}^{n} \frac{y_i}{\pi_i} \quad (4)$$

where π_i is the probability of the *i*th unit is in the sample.

To find π_i is necessary to compute the conditional probability. Let ψ_i = P(select unit *i* on the first draw). In general (Lohr, 1999), P(unit *i* chosen first, unit *j* chosen second) = *P*(*i*)*P*(*j* | *i*)

$$P(i)P(j \mid i) = \psi_i \frac{\psi_j}{1-\psi_i} \quad (5)$$

Similarly,

$$P(j)P(i \mid j) = \psi_j \frac{\psi_i}{1-\psi_j} \quad (6)$$

Note that in sampling without replacement,

$$\psi_i \frac{\psi_j}{1-\psi_i} \neq \psi_j \frac{\psi_i}{1-\psi_j}$$

and for *n* = 2, π_{ij} is given by

$$\pi_{ij} = \psi_i \frac{\psi_j}{1-\psi_i} + \psi_j \frac{\psi_i}{1-\psi_j} \quad (7)$$

So, π_i is computed by (Lohr, 1999)

$$\pi_i = \sum_{j=1,\, j\neq i}^{N} \pi_{ij} \big/ (n-1) \quad (8)$$

Also, note that $\sum_{i=1}^{N} \pi_i = n$.

The variance of HT estimator (Horvitz and Thompson, 1952) is given by:

$$V(\hat{t}_\pi) = \sum_{i=1}^{N} \frac{1-\pi_i}{\pi_i} y_i^2 + \sum_{i=1}^{N} \sum_{j\neq i}^{N} \frac{(\pi_{ij} - \pi_i \pi_j)}{\pi_i \pi_j} y_i y_j \quad (9)$$

where π_{ij} is the probability of *i*th and *j*th units are both in the sample.

The second part of (9) is about the covariance between the populations units $[Cov(y_i, y_j)]$. Depending of population size *N* much computational resource is required because are necessary C_2^N combinations. The sample estimate of variance is given by (Horvitz and Thompson, 1952):

$$\hat{V}_1(\hat{t}_\pi) = \sum_{i=1}^{N} \frac{1-\pi_i}{\pi_i^2} y_i^2 + \sum_{i=1}^{N} \sum_{j\neq i}^{N} \frac{(\pi_{ij} - \pi_i \pi_j)}{\pi_{ij} \pi_i \pi_j} y_i y_j \quad (10)$$

However, this estimator can result in negative estimates. Another sample estimator of variance is given by Yates and Grundy (1953) as:

$$\hat{V}_2(\hat{t}_\pi) = \sum_{i=1}^{N} \sum_{j>i}^{N} \frac{(\pi_i \pi_j - \pi_{ij})}{\pi_{ij}} \left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2 \quad (11)$$

The above formulas for the Horvitz-Thompson estimator look very similar to the formulas that are applying under equal probability sampling. However, the variance of the HT estimator for the population total looks a little bit different and it is not calculated by PROC SURVEYMEANS.

3 SAS® MACRO

The SAS® Macro basically uses the IML (Interactive Matrix Language) procedure and the parameters of the algorithms are:

```
%HT_General(tab=,*the name of database*
            var=,*the name of the variable to be analyzed*
            aux=,*the name of the variable to be used in the probability selection*
            n=*the sample size*,
            str=*the name of the variable that constitute a stratum in a stratified
                     design*)
```

If it will not be specified any value for the parameter "str" then the Simple Random Sampling will be done. The parameter "n" is used to draw a sample of size *n* in all strata or the sample size in a Simple Random Sampling design.

The macro begins with the generation of all permutations of *n* elements using the program developed by Wicklin (2010). This step is the only outside of IML environment, because it was not possible to use recursive functions to generate the permutations inside IML. These samples are the key of the HT estimator, because as it was seen before, the sample order is very important.

```
data perm&n (drop=i);
  array a{&n};
  do i = 1 to &n; a[i]=i; end; /** initialize **/
  do i = 1 to fact(&n);
    call allperm(i, of a[*]);
    output;
  end;
run;
```

The dataset "perm&n", where &n is the sample size and it is resolved to perm1, perm2 and so forth is created with the positions (indices) of the samples. The real samples are created in the dataset "estimates" together with the HT estimator, SRS estimator, the Probability of selected sample and the variances.

3.1 Macro and Function Call

The results presented in the Section 4 were created using the call bellow.

```
/***** GENERAL SAS *******/
data database;
input id sale size srs;
cards;
1 11 100 1
2 20 200 1
3 24 300 1
4 245 1000 1
;
/* Using HT Estimator*/
%HT_general(tab=database,var=sale,aux=size,n=2,str=);
/* Using SRS Estimator*/
%HT_general(tab=database,var=sale,aux=srs,n=2,str=);
```

4 ILLUSTRATION

To illustrate the potential of the algorithm, we consider the example presented by Lohr (1999) about supermarkets. A town has four supermarkets, ranging in size from 100 square meters (m^2) to 1000 m^2. We want to estimate the total amount of sales in the four stores for last month by sampling two stores. The data are in Table 1.

Store	Size (m^2)	Sales (in Thousands)
A	100	11
B	200	20
C	300	24
D	1000	245
Total	1600	300

Table 1- Sales by Supermarkets Size.

We can see that the Total amount of sales is 300 and considering a sample of size *n* = 2, the joint probability, π_{ij}, and the stores probability π_i are in Table 2. The probability of Stores A and B is in the sample is computed by $\frac{100}{1600}\frac{200}{1500}+\frac{200}{1600}\frac{100}{1500}=0.0172619$. Also, note that $\sum_{i=1}^{4}\pi_i=2$ and all possible samples are shown in Table 3. The expected value, which is equal to 300 (an unbiased estimate) is computed by summing the product of the variables "HT" and "Prob". It is interesting to note that the total number of samples 12 is computed by $C_2^N n! = C_2^4 2!$ and that the probability of Sample 1 is given by $\frac{100}{1600}\frac{200}{1500}=0.0083$. The variance of HT estimator using Equation (9) and for *n* = 2 is equal to 4383.5622. In the same way, the expectation of VarHT1 and VarHT2 (Table 3) are both equal to 4383.5622. Again, unbiased estimates. To evaluate the efficiency of HT estimator in relation to SRS we calculate the design effect, *deff* = 8.51%, knowing that the SRS variance also for *n* = 2 is equal to 51496, showing the great efficiency of HT estimator.

	A	B	C	D	π_i
A	0	0.0172619	0.0269231	0.1458333	0.1900183
B	0.0172619	0	0.0556319	0.297619	0.3705128
C	0.0269231	0.0556319	0	0.4567308	0.5392857
D	0.1458333	0.297619	0.4567308	0	0.9001832
π_i	0.1900183	0.3705128	0.5392857	0.9001832	2

Table 2: Joint Probability Selection.

Sample	Selection1	Selection2	HT	Prob	VarHT2	VarHT1
1	11	20	111.868	0.00833	47.06	-14691.48
2	20	11	111.868	0.00893	47.06	-14691.48
3	11	24	102.392	0.01250	502.81	-10832.07
4	24	11	102.392	0.01442	502.81	-10832.07
5	11	245	330.056	0.04167	7939.75	4659.30
6	245	11	330.056	0.10417	7939.75	4659.30
7	20	24	98.483	0.02679	232.72	-9705.15
8	24	20	98.483	0.02885	232.72	-9705.15
9	20	245	326.146	0.08929	5744.06	5682.80
10	245	20	326.146	0.20833	5744.06	5682.80
11	24	245	316.670	0.14423	3259.78	6782.82
12	245	24	316.670	0.31250	3259.78	6782.82

Table 3: All Possible Samples

If one desire to compute SRS estimates from HT estimator, it is enough to let the Size variable all equal to 1. The results are in Tables 4 and 5. Now, the probability of Stores A and B (or any store) is in the sample of size *n* = 2 is computed by $\frac{1}{4}\frac{1}{3}+\frac{1}{4}\frac{1}{3}=0.1666667$, and the probability of any store is in the sample is $\frac{n}{N}=0.5$. As the probability of the *i*th element drawn is the same, then the probability of any sample is $\frac{1}{4}\frac{1}{3}=0.0833$. The variance of HT estimator using Equation (9), for *n* = 2, and the expected values for VarHT1 and VarHT2 are all equal to 51496, the same variance of SRS estimator shown above.

	A	B	C	D	π_i
A	0	0.1666667	0.1666667	0.1666667	0.5
B	0.1666667	0	0.1666667	0.1666667	0.5
C	0.1666667	0.1666667	0	0.1666667	0.5
D	0.1666667	0.1666667	0.1666667	0	0.5
π_i	0.5	0.5	0.5	0.5	2

Table 4: Joint Probability Selection

Sample	Selection1	Selection2	HT	Prob	VarHT2	VarHT1
1	11	20	62	0.083333	162	162
2	20	11	62	0.083333	162	162
3	11	24	70	0.083333	338	338
4	24	11	70	0.083333	338	338
5	11	245	512	0.083333	109512	109512
6	245	11	512	0.083333	109512	109512
7	20	24	88	0.083333	32	32
8	24	20	88	0.083333	32	32
9	20	245	530	0.083333	101250	101250
10	245	20	530	0.083333	101250	101250
11	24	245	538	0.083333	97682	97682
12	245	24	538	0.083333	97682	97682

Table 5: All Possible Samples

4.1 Sample distribution

The sample distribution is important to calculate the confidence interval for the parameter. For the case of HT estimator, simulations show that sample distribution can be symmetric or asymmetric (left or right). So, the normal distribution can not be the best choice to construct the confidence interval for the HT estimator, but as was shown before, when the variable Size is equal to 1 for all observation then the HT estimator computes the same SRS estimate. Figure 1 shows the exact 95% confidence interval for the HT estimator, exact 95% confidence interval for the SRS estimator, and normal and t approximation 95% confidence interval for the SRS estimator for the data presented in Table 1. We can see that interval for the HT estimator is the smallest and it has the true parameter, in this case equal to 300. Also, note the asymmetry of the distribution of the HT estimator and the symmetry of the distribution of the SRS estimator.

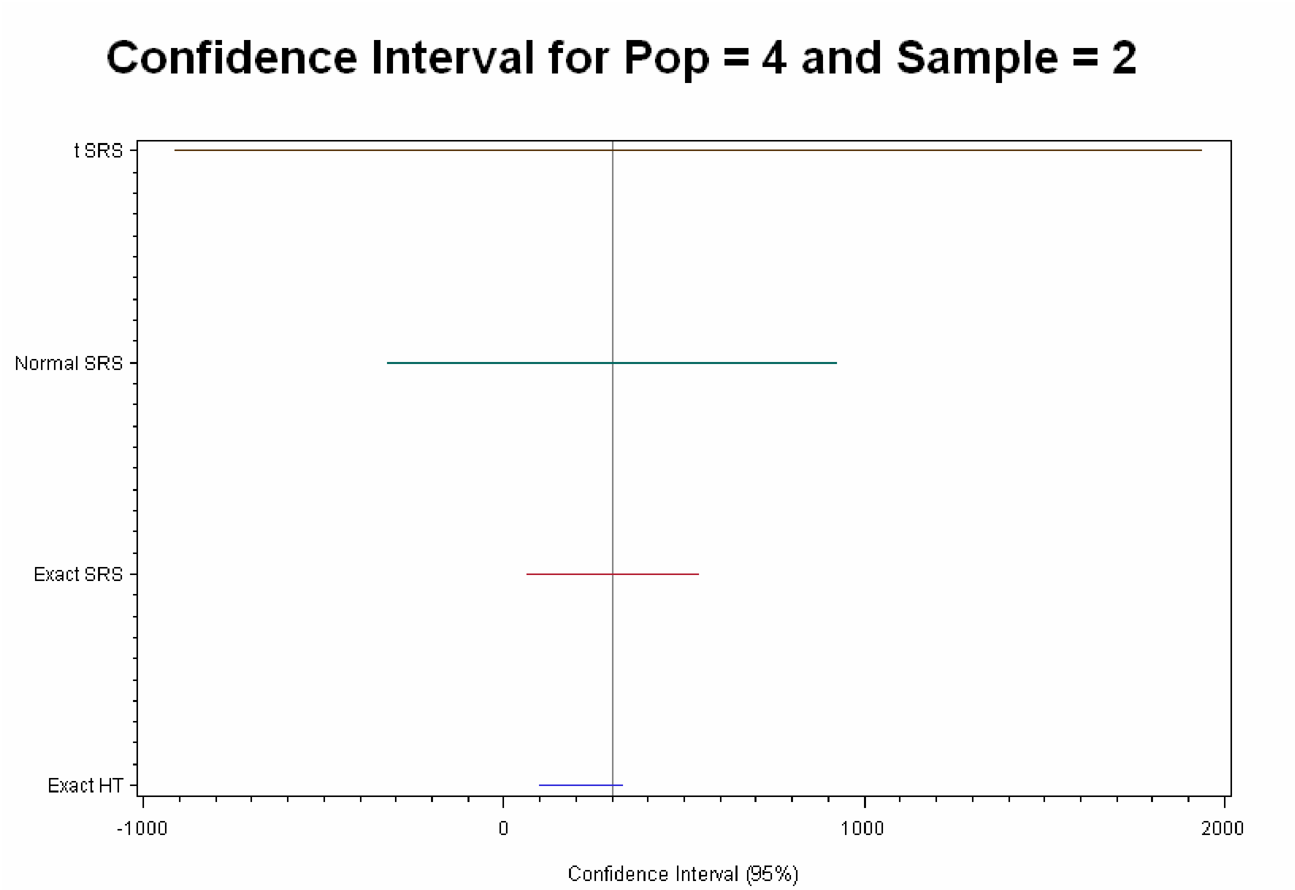

Figure 1. Confidence Interval for the HT and SRS estimators.

CONCLUSIONS

The HT estimator reduces considerably the variance of the total estimate in relation to Simple Random Sampling or Stratified Random Sampling. However, to compute its variance is necessary many calculations because the $[Cov(y_i, y_j)]$ and because of the sampling without replacement, which considers the order of the drawn. The algorithm presented here is important besides to generate all possible samples to verify, computationally, properties as expectation and population variance $V(\hat{t}_\pi)$ instead $\hat{V}(\hat{t}_\pi)$, become easier the sampling teaching, in contrast with the PROC SURVEYSELECT which generates any samples of size *n*. Additionally, it was possible to compute the exact confidence interval for the HT estimator and we have as a first approximation for the probability distribution for the HT estimator, the skew normal.

It is important to note that the HT Estimator is the general case and only it needs to be computationally implemented. However, the macro presented here has a limitation to handle a population only of size less than 170 because it is not possible (lack of memory) to compute the factorial of numbers greater than 170.

REFERENCES

Cochran, W.G. (1977). *Sampling Techniques*. John Wiley & Sons, 3rd edition.

Durbin, J. (1953). *Some Results in Sampling when the units are selected with unequal probabilities*. Journal of the Royal Statistical Society, B, 15, 262-269.

Hansen, M.H. and Hurwitz, W.N. (1943). *On the theory of sampling from finite population*. The Annals of Mathematical Statistics, pp. 333-362.

Horvitz, D.G. and Thompson, D.J. (1952). *A Generalization of Sampling Without Replacement From a Finite Universe*. Journal of the American Statistical Association, pp. 663-685.

Lohr, S.L. (1999). Sampling: Design and Analysis. Duxbury Press.

Sarndal, C.E., Swensson, B. and Wretman, J. (1992). *Model Assisted Survey Sampling.* Springer Series in Statistics, 2th edition.

Wicklin, R. (2010). *Generating All Permutations*. http://blogs.sas.com/iml/index.php?/archives/26-Generating-All-Permutations.html.

Yates, F., Grundy, P.M. (1953). *Selection without replacement from within strata with probability proportional to size*. Journal of the Royal Statistical Society, pp. 253-261.

CONTACT INFORMATION

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

Name: Alan Ricardo da Silva
Enterprise: Universidade de Brasília
Address: Campus Universitário Darcy Ribeiro, Departamento de Estatística, Prédio CIC/EST sala A1 35/28
City, State ZIP: Brasília, DF, Brazil, 70910-900
Work Phone: +5561 3107 3672
Fax: +5561 3107 6768
E-mail: alansilva@unb.br
Web: www.est.unb.br

Name: Igor Ferreira do Nascimento
Enterprise: Universidade de Brasília
Address: Campus Universitário Darcy Ribeiro, Departamento de Estatística, Prédio CIC/EST sala A1 35/28
City, State ZIP: Brasília, DF, Brazil, 70910-900
Work Phone: +5561 91084901 and +5561 33291807
E-mail: igor.ferreira.n@gmail.com

APPENDIX I – SAS® MACRO

```
/************************************
*****/
/********** GENERAL
*********************/
/************************************
*****/

%macro
HT_general(tab=,var=,aux=,n=,str=);
/** generate all permutations of n
elements **/
data perm&n (drop=i);
array a{&n};
do i = 1 to &n; a[i]=i; end; /**
initialize **/
do i = 1 to fact(&n);
call allperm(i, of a[*]);
output;
end;
run;

PROC IML;
******************************;
*COMBINATION FUNCTION - BEGIN;
******************************;
start combination(pop,tamamostra);
N=pop;
m=tamamostra;
vetor=1:N;
e = 0;
h = m;
indice = 1:H;
vec=vetor[indice];
count =fact(n)/(fact(m)*fact(n-m));
matriz=vec;
do i=1 to count-1 ;
       matriz = matriz || vec;
end;
if ncol(matriz) ^= count then
  matriz = matriz || vec;
i = 2;
criterio = n - m + 1;
DO k=1 to count;
       aa=indice[1];
    IF (aa^=criterio) THEN DO;
       IF (e < n - h) THEN DO;
              h = 1;
              e = indice[m];
              G = 1;
       END;
       ELSE DO;
              e = indice[m - h];
              h = h + 1;
              G = 1:H;
       END;
   indice[m - h + G] = e +G;
   vec = vetor[indice];
   matriz[, i] = vec;
   i = i + 1;
   END;
END;
return(matriz);
finish combination;
*****************************;
*COMBINATION FUNCTION - END;
*****************************;

/***** inserting data and using
comb/perm ****/
str=1;
%if &str ^= %then %do;
use &tab var{&str};
read all;
tstr=nrow(&str);
str=unique(&str);
varStHT=0;
%end;
do nstr=1 to ncol(str);
st=str[nstr];
%if &str ^= %then %do;print st;%end;
use &tab var{&var};
%if &str ^= %then %do;use &tab
var{&var &str} where(&str=:st);%end;
read all into y;
y=y[,1];
use &tab var{&aux &str};
%if &str ^= %then %do;use &tab
var{&aux &str} where(&str=:st);%end;
read all into trab;
trab=trab[,1];
use perm&n;
read all into perm;
y=y`;
trab=trab`;

pop=ncol(y);
n=&n;
comb=combination(POP,n);
comb2=combination(n,2);

/**************** end
*********************/

/******** joint probability matrix
**********/
P=trab/trab[+];
pi=J(1,pop,0);
PP=J(pop,pop,0);
_Total_=y[+];

l1=comb(pop,n);
l2=fact(n);
l3=comb(n,2);

free ppp hts varht2 varht1 pij;
do i=1 to l1;
vec1=comb[,i]`;
do j=1 to l2;
vec2=perm[j,];

ind=vec1[vec2];
pijk=1;
soma=0;
```

```
do k=1 to n;
pk=p[1,ind[k]]/(1-soma);
soma=soma+p[1,ind[k]];
pijk=pijk*pk;
end;
ppp=ppp||pijk;/*probability of
selected sample*/

do k=1 to l3;
vec3=comb2[,k]`;
ind=vec1[vec2[vec3]];
pp[ind[1],ind[2]]=pp[ind[1],ind[2]]+
pijk;
pp[ind[2],ind[1]]=pp[ind[2],ind[1]]+
pijk;
*print pp;
end;
end;
end;

do i=1 to pop;
pi[i]=sum(pp[,i])/(n-1);
end;

/************** end
***********************/


*********************************;
*********** HT Estimator ********;
*********************************;
um=j(n,1,1/n);
free varsrsa srs amostra_pop;

do i=1 to l1;
vec1=comb[,i];
do j=1 to l2;
vec2=perm[j,];
ind=vec1[vec2];
hts=hts || sum(y[ind]/pi[ind]);
srs=srs || (pop/n)*sum(y[ind]);
amostra_pop=amostra_pop||y[ind];
amos=y[ind];
media=amos`*um;
varm=(amos`-media)*(amos`-
media)`/(n-1);
varsrsa=varsrsa||(varm/n)*pop**2*(1-
n/pop);

vp=0;
do ii=1 to (n-1);
       do jj=(ii+1) to n;
vp=vp+((pi[ind][ii]*pi[ind][jj]-
pp[ind,ind][ii,jj])/pp[ind,ind][ii,j
j])*((y[ind][ii]/pi[ind][ii])-
(y[ind][jj]/pi[ind][jj]))**2;
end;
end;
varht2=varht2||vp;

var1=0;
do ii=1 to n;
       var1=var1+((1-
pi[ind][ii])/(pi[ind][ii]**2))*y[ind
][ii]**2;
end;

var2=0;
do ii=1 to (n-1);
       do jj=(ii+1) to n;

       var2=var2+((pp[ind,ind][ii,jj]
-
(pi[ind][ii]*pi[ind][jj]))/(pp[ind,i
nd][ii,jj]*pi[ind][ii]*pi[ind][jj]))
*(y[ind][ii]*y[ind][jj]);
       end;
end;
varht1=varht1||var1+2*var2;
end;
end;

amostra_pop2=amostra_pop`;

do i=1 to (pop-1);
       do j=(i+1) to pop;
              pij= pij || PP[i,j];
       end;
end;

*#Var HT;
var1=0;
do i=1 to pop;
var1=var1+((1-
pi[i])/pi[i])*(y[i]**2);
end;

var2=0;
cont=0;
do i=1 to (pop-1);
       do j=(i+1) to pop;
              cont=cont+1;

       var2=var2+2*((pij[cont]-
(pi[i]*pi[j]))/(pi[i]*pi[j]))*(y[i]*
y[j]);
       end;
end;

varHT=var1+var2;
%if &str ^= %then %do;
varStHT=varStHT+varHT;
%end;

/*average and variance of SRS */;
um=j(pop,1,1/pop);
media=y*um;
var=(y-media)*(y-media)`/(pop-1);
varSRS=(var/n)*pop**2*(1-n/pop);

/*average and variance of HT*/
E_hts=hts*ppp`;
E_vht2=varht2*ppp`;
E_vht1=varht1*ppp`;
```

```
print PP;
sumpi=pi[+];
print Pi sumpi;
print 'Number of Samples: ' (l1*l2);
if l1*l2<100 then do;
             print
hts[format=comma10.4];
             print
varht2[format=comma10.4];
             print
varht1[format=comma10.4];
      end;
print _Total_;

print E_hts E_vht2 E_vht1;
print varHT varSRS;
deff=varHT/varSRS;
print deff[format=percent10.2];

strata=j(ncol(hts),1,st)`;
if nstr=1 then do;
Estimates=amostra_pop2||hts`||srs`||
ppp`||varht2`||varht1`||varsrsa`||st
rata`;
end;
else do;
Estimates=Estimates//(amostra_pop2||
hts`||srs`||ppp`||varht2`||varht1`||
varsrsa`||strata`);
end;
end;
%if &str ^= %then %do;

print "************************
stratified estimates
*************************";
hts=Estimates[,ncol(amostra_pop2)+1]
;
ppp=Estimates[,ncol(amostra_pop2)+3]
;
varht2=Estimates[,ncol(amostra_pop2)
+4];
varht1=Estimates[,ncol(amostra_pop2)
+5];
E_hts=hts`*ppp;
E_vht2=varht2`*ppp;
E_vht1=varht1`*ppp;
print E_hts E_vht2 E_vht1;

varStRS=0;
use &tab var{&var};
read all into y;
popt=nrow(y);
read all into y;
close &tab;

do i=1 to ncol(str);
st=str[i];
use &tab var{&var &str}
where(&str=:st);
read all into y;
y=y[,1];
pop=nrow(y);
um=j(pop,1,1/pop);
media=y`*um;
var=(y-media)`*(y-media)/(pop-1);
*Stratified Sample;
varStRS=varStRS+popt**2*(var/n)*(pop
/popt)**2*(1-n/pop);
end;
print varStHT varStRS;
deff1=varStHT/varStRS;
print
deff1[colname={"Deff(StHT/StRS)"}
format=percent10.2];
use &tab var{&var};
read all into y;
pop=nrow(y);
um=j(pop,1,1/pop);
media=y`*um;
n=ncol(str)*n;
var=(y-media)`*(y-media)/(pop-1);
*SRS;
varSRS=(var/n)*pop**2*(1-n/pop);
print varSRS;
deff2=varStHT/varSRS;
print
deff2[colname={"Deff(StHT/SRS)"}
format=percent10.2];
%end;
colnames="selection1":"selection&n"|
|{"hts" "srs" "prob" "varht2"
"varht1" "varsrs" "strata"};
create Estimates from
Estimates[colname=colnames];
append from Estimates;
quit;
%mend HT_general;
```