

Poisson Regressions for Complex Surveys

Overview

Researchers often use sample survey methodology to obtain information about a large population by selecting and measuring a sample from that population. Researchers apply probability-based scientific designs to select the sample in order to reduce the risk of a distorted view of the population and to enable statistically valid inferences to be made from the sample.

The SURVEYMEANS, SURVEYFREQ, SURVEYREG, SURVEYLOGISTIC, and SURVEYPHREG procedures in SAS/STAT software properly analyze complex survey data by taking into account the sample design. You can use these procedures for multistage or single-stage designs, with or without stratification, and with or without unequal weighting. The survey analysis procedures provide a choice of variance estimation methods, which include Taylor series linearization, balanced repeated replication (BRR), and the jackknife.

When you use most other SAS/STAT procedures, statistical inference is based on the assumption that the sample is drawn from an infinite population by simple random sampling. If the sample is in fact selected from a finite population by using a complex survey design, these procedures usually do not calculate the estimates and their variances according to the design that is actually used. Using analyses that are not appropriate for your sample design might lead to incorrect statistical inferences.

However, there might be times when you want to analyze data that are sampled from a finite population by using a complex survey design, but the only SAS/STAT procedure capable of fitting the type of model that you need is not designed to account for sampling based on a complex survey design. In such cases, you can sometimes use a non-survey procedure to obtain valid point estimates of the model parameters, and use the SURVEYMEANS procedure and a little programming to obtain valid standard errors for the model parameter estimates. Specifically, this example demonstrates how to combine the generalized linear modeling capabilities of the GENMOD procedure and the delete-1 jackknife (resampling) method of the SURVEYMEANS procedure to fit a Poisson model to count data that are sampled from a finite population by using a complex survey design.

Performing the delete-1 jackknife estimation of the standard errors of the model parameter estimates requires fitting a model to each of the jackknife replicates. As is typical in programming, there is more than one way to perform most tasks. This example demonstrates two different ways to accomplish the same task. “Step 3a: Fit a Model to Each Replicate Sample by Using BY-Group Processing” on page 7 uses the GENMOD procedure’s BY-group processing capabilities to fit a model to each replicate; this is the most efficient method. “Step 3b: Fit a Model to Each Replicate Sample by Looping Through the Replicates” on page 8 demonstrates how to perform the same task by using a SAS macro to loop through the replicates. Looping is less efficient than by-group processing but requires less computer memory, which might become an issue if you have a very large sample.

The SAS source code for this example is available as an attachment in a text file. In Adobe Acrobat, right-click the icon and select **Save Embedded File to Disk**. You can also double-click the icon to open the file.

[source code](#)

Analysis

Obtaining Point Estimates of Model Parameters

Consider a finite population whose members are indexed by $U = \{1, 2, \dots, N\}$ and where \mathcal{F}_N is the set of values for the population. Suppose you specify a population density function $f(y, \theta)$, where the parameter θ is of interest. If the entire population is observed, then this likelihood can be used to estimate θ . Let θ_N be the desired estimator. θ_N is obtained by maximizing the log likelihood

$$L(\theta) = \sum_{i \in U} \log [f(y_i, \theta)]$$

with respect to θ .

Assume that probability sample A is selected from the finite population U and π_i is the selection probability for unit i . An estimator of the finite population log likelihood is

$$L_\pi(\theta) = \sum_{i \in A} \pi_i^{-1} \log [f(y_i, \theta)]$$

A sample-based estimator $\hat{\theta}$ for the finite population quantity θ_N can be obtained by maximizing the pseudo-log-likelihood $l_\pi(\theta)$ with respect to θ . The design-based variance for $\hat{\theta}$ is obtained by assuming the set of finite population values \mathcal{F}_N to be fixed. For more information about maximum pseudo-likelihood estimators and other inferential approaches for survey data, see Kish and Frankel (1974); Godambe and Thompson (1986); Pfeiffermann (1993); Korn and Graubard (1999, chapter 3); Chambers and Skinner (2003, chapter 2); and Fuller (2009, section 6.5).

The practical implication of the preceding analysis is that if a SAS/STAT procedure performs weighted maximum likelihood estimation and the weights are applied such that the weights can be factored out of the log likelihood, then that procedure can generate valid point estimates of model parameters when the data are sampled according to a complex survey design.

The WEIGHT statement in the GENMOD procedure identifies a variable in the input data set to be used as the exponential family dispersion parameter weight for each observation. The exponential family dispersion parameter is divided by the WEIGHT variable value for each observation. This is true regardless of whether the parameter is estimated by the procedure or specified in the MODEL statement by using the SCALE= option. It is also true for distributions such as the Poisson and binomial that are not usually defined to have a dispersion parameter. For these distributions, a WEIGHT variable weights the overdispersion parameter, which has the default value of 1.

Consider a Poisson regression model of the observed number of counts, y_i , on a set of covariates, \mathbf{x}_i , for units $i \in A$. Assume that $y_i \sim \text{Poisson}(\theta_i)$ and the mean θ_i of the response in the i th observation is related

to a linear predictor through the link function $\log(\theta_i) = \mathbf{x}'_i \boldsymbol{\beta}$, where $\boldsymbol{\beta}$ is a vector of unknown parameters. The log likelihood can be written as

$$ll_i = w_i [y_i \log(\theta_i) - \theta_i - \log(y_i!)]$$

Because the weight, w_i , can be factored out of the log likelihood, you can use PROC GENMOD with a WEIGHT statement to obtain valid point estimates of the model parameters.

Caution

However, the log likelihood for the negative binomial model is

$$ll_i = y_i \log\left(\frac{k\theta}{w_i}\right) - (y_i + w_i/k) \log\left(1 + \frac{k\theta}{w_i}\right) + \log\left(\frac{\Gamma(y_i + w_i/k)}{\Gamma(y_i + 1)\Gamma(w_i/k)}\right)$$

The weight, w_i , cannot be factored out of the log likelihood, so you cannot use PROC GENMOD with a WEIGHT statement to obtain point estimates of the model parameters that account for the unequal weights.

Whereas the weighted maximum likelihood point estimates that PROC GENMOD generates appropriately account for the unequal weights for distributions such as the Poisson, the weighted maximum likelihood variances and standard errors that PROC GENMOD computes do not account for the complex survey design. You must compute the variances and standard errors by using a different method. One such method is the delete-1 jackknife (resampling) method.

Obtaining Variance Estimates by Using the Delete-1 Jackknife Method

The jackknife method of variance estimation deletes one primary sampling unit (PSU) at a time from the full sample to create replicates. This method is also known as the delete-1 jackknife method, because it deletes exactly one PSU in every replicate. The total number of replicates R is the same as the total number of PSUs. In each replicate, the sampling weights of the remaining PSUs are modified by the *jackknife coefficient* α_r . The modified weights are called replicate weights.

Let PSU i in stratum h_r be omitted from the r th replicate; then the jackknife coefficient and replicate weights are computed as

$$\alpha_r = \begin{cases} \frac{n_{h_r} - 1}{n_{h_r}} & \text{for a stratified design} \\ \frac{R - 1}{R} & \text{for designs without stratification} \end{cases}$$

and

$$w_{hij}^{(r)} = \begin{cases} w_{hij} & \text{if observation unit } j \text{ is not in donor stratum } h_r \\ 0 & \text{if observation unit } j \text{ is in PSU } i \text{ of donor stratum } h_r \\ w_{hij}/\alpha_r & \text{if observation unit } j \text{ is not in PSU } i \text{ but in donor stratum } h_r \end{cases}$$

You can use the `VARMETHOD=JACKKNIFE(OUTJKCOEFS=)` *method-option* with any of the survey estimation procedures to store the jackknife coefficients in a SAS data set and use the `VARMETHOD=JACKKNIFE(OUTWEIGHTS=)` *method-option* to store the replicate weights in a SAS data set.

Let $\hat{\theta}$ be the estimated parameters from the full sample, and let $\hat{\theta}_r$ be the estimated parameters for the r th replicate. You can estimate the covariance matrix of $\hat{\theta}$ by

$$\widehat{V}(\hat{\theta}) = \sum_{r=1}^R \alpha_r (\hat{\theta}_r - \hat{\theta}) (\hat{\theta}_r - \hat{\theta})'$$

It is common to assume that the distribution of $\widehat{V}(\hat{\theta})$ can be approximated by using a χ^2 distribution with $R - H$ degrees of freedom, where R is the number of replicates and H is the number of strata, or $R - 1$ degrees of freedom when there is no stratification.

If one or more components of $\hat{\theta}_r$ cannot be calculated for some replicates, then you use only the replicates for which the parameters can be estimated. Estimability and nonconvergence are two common reasons why $\hat{\theta}_r$ might not be available for a replicate sample even if $\hat{\theta}$ is defined for the full sample. Let R_a be the number of replicates where $\hat{\theta}_r$ are available, and let $R - R_a$ be the number of replicates where $\hat{\theta}_r$ are not available. Without loss of generality, assume that $\hat{\theta}_r$ is available only for the first R_a replicates; then the jackknife variance estimator is

$$\widehat{V}(\hat{\theta}) = \sum_{r=1}^{R_a} \alpha_r (\hat{\theta}_r - \hat{\theta}) (\hat{\theta}_r - \hat{\theta})'$$

with $R_a - H$ degrees of freedom, where H is the number of strata.

Example

Consider a hypothetical regional survey that seeks to describe the number of visits to health professionals that are made annually by members of a population. The survey is conducted by using a stratified clustered sampling design. The following statements create the SAS data set `Counts`. The variable `Visits` is a count variable that records the number of visits to a health professional; `Sex` is a binary variable that records the gender of the respondent; `Race` is a categorical variable that records each respondent's race; `Marital` is a categorical variable that records each respondent's marital status; `Private` is a categorical variable that records whether a respondent has private health insurance, and if so, what type; `Education` is a categorical variable that records each respondent's highest attained level of education; `Person` is a respondent's unique identifier; `Strata` identifies the stratum from which each observation is sampled; `PSU` identifies the primary sampling units; and `SamplingWeight` records the sampling weights.

```

data counts;
  input visits sex race marital private education
        person strata psu SamplingWeight @@;
  datalines;
5 1 1 2 1 5 71511 1 1 1002.59 1 2 1 4 2 3 307568 1 1 1002.59
2 1 1 4 4 3 457473 1 1 1002.59 9 1 1 3 1 5 849963 1 1 1002.59
3 2 1 3 2 5 892466 1 1 1002.59 0 2 1 2 3 3 249075 1 2 1002.59
3 1 1 2 4 1 835408 1 2 1002.59 1 2 1 4 2 4 159262 1 3 1002.59

... more lines ...

2 2 1 1 4 2 244599 5 40 998.26 1 2 1 3 4 4 738928 5 40 998.26
2 2 1 3 2 2 830211 5 40 998.26 3 1 1 3 2 3 920025 5 40 998.26
;
run;

```

Step 1: Generate the Jackknife Coefficients and Replicate Weights

In the first step in the process, you generate the jackknife coefficients and replicate weights by using the SURVEYMEANS procedure and save the number of replicates and the number of strata in macro variables.

The following statements analyze the variable Visits and save the jackknife coefficients and replicate weights in the data sets JKcoefs and JKweights, respectively. It does not matter which variable you choose to analyze; the jackknife coefficients and replicate weights are the same regardless of the variable that you choose. If the replicate weights are available to you, then you can skip the PROC SURVEYMEANS step. However, you still need to create the macro variables &Replicates and &H, which are generated to contain the number of replicates and the number of strata, respectively.

```

ods select none;
ods output VarianceEstimation=VE Summary=Summary;
proc surveymeans data=counts plots=none
  varmethod=jackknife(outweights=jkweights outjkcoefs=jkcoefs);
  cluster psu;
  strata strata;
  weight SamplingWeight;
  var visits;
run;

```

The first statement suppresses all ODS output. You can omit this statement if you want to see the output from each step. The ODS OUTPUT statement saves variance estimation table in the data set VE and saves the sampling design summary information in the data set Summary. VE contains the number of jackknife replicates that are created, and Summary contains the number of strata. Both the number of jackknife replicates and the number of strata are later retrieved and saved in macro variables. The VARMETHOD=JACKKNIFE option in the PROC SURVEYMEANS statement specifies the delete-one jackknife variance estimation method. The OUTWEIGHTS= suboption saves the jackknife replicate weights in the data set JKweights. The OUTCOEFS= suboption saves the jackknife coefficients in the data set JKcoefs. The CLUSTER, STRATA, and WEIGHT statements specify the sampling design. The VAR statement names the variable to be analyzed.

The following statements retrieve the number of replicates from the VE data set and the number of strata from the Summary data set. These values are stored in the macro variables &Replicates and &H, respectively.

```

data _null_;
  set VE(where=(Label1="Number of Replicates"));
  call symput('replicates',cValue1);
run;

data _null_;
  set Summary;
  if Label1="Number of Strata" then do;
    call symput('H',cValue1);
  end;
run;

```

Step 2: Fit the Model by Using the Full Sample and the Original Sampling Weights

In the second step you fit a model by using the full sample and the original sampling weights. You then compute the number of parameters that are estimated by using the full sample and save that value in a macro variable.

The following statements use the GENMOD procedure to fit a Poisson model by using the full sample and the original sampling weights:

```

ods output ParameterEstimates=FullSample(where=(Parameter ne "Scale")
                                           keep=Parameter Estimate Level1
                                           rename=(Estimate=Estimate0)
                                           ParameterEstimates=parms(keep=df));
proc genmod data=jkweights;
  class sex race marital private education;
  weight SamplingWeight;
  model visits = sex race marital private education / dist=poisson;
run;

```

The ODS OUTPUT statement saves the parameter estimates from the Poisson model to the data set FullSample; the scale parameter is excluded and the variable Estimate, which contains the parameter estimates, is renamed Estimate0. The same statement also saves the variable DF, which contains the number of regression parameters that are estimated by using the full sample, in the data set Parm. The DATA= option in the PROC GENMOD statement specifies that the data set JKweights, which contains the original data as well as the replicate weights, be used. The CLASS statement names the classification variables to be used as explanatory variables in the analysis. The WEIGHT statement specifies that the variable SamplingWeight be used as the exponential family dispersion parameter weight for each observation. The MODEL statement specifies the response variable and the explanatory variables, and the DIST= option specifies the Poisson distribution.

The following statements compute the number of parameters that are estimated by using the full sample and saves that value in the macro variable &P. This step is needed because the full model might not be defined in some replicate samples and you need to exclude replicate models that do not have the same number of parameters as the full model.


```

        ConvergenceStatus=converge;
proc genmod data=jkweights;
  class sex race marital private education;
  weight repweight;
  model visits = sex race marital private education / dist=poisson;
  by replicate;
run;

```

The ODS OUTPUT statement saves the parameter estimates for all R models in the data set JKparms, the degrees of freedom for all the models in the data set JKDF, and the convergence status for all the models in the data set Converge. The WEIGHT statement specifies that the variable RepWeight be used as the exponential family dispersion parameter weight for each observation. The BY statement requests separate analyses of observations in groups that are indexed by the variable Replicate.

Step 3b: Fit a Model to Each Replicate Sample by Looping Through the Replicates

Rather than fitting a model to each replicate sample by using the GENMOD procedure's by-group processing capabilities, you can write and execute the macro %JKLOOP. This method is less efficient but requires less computer memory, which might become an issue if you have a very large sample. The macro %JKLOOP has one required argument, REPLICATES=, which specifies the number of jackknife replicates. The macro loops through the R replicates and fits a Poisson model by using the appropriate replicate sample and jackknife replicate weights. The parameter estimates from each model are saved in the temporary data set Temp, the degrees of freedom for each model is saved in the temporary data set Temp2, and the convergence status of the model is saved in the temporary data set Temp3. A series of DATA steps then add the variable Replicate to Temp, Temp2, and Temp3. The data sets Temp, Temp2, and Temp3 are then appended to the data sets JKparms, JKDF, and Converge, respectively. Finally, the variable Estimate in the data set FullSample is renamed Estimate0.

The following statements create the macro %JKLOOP:

```

%macro jkloop(replicates=);
  %local _nopt; %let _nopt = %sysfunc(getoption(notes));
  options nonotes;
  ods select none;
  %do i=1 %to &replicates;
    ods output ParameterEstimates=temp(where=(Parameter ne "Scale")
      keep=Parameter Estimate Level1)
      ParameterEstimates=temp2(where=(Parameter ne "Scale")
      keep=Parameter Level1 df)
      ConvergenceStatus=temp3;
    proc genmod data=jkweights;
      class sex race marital private education;
      weight RepWt_&i;
      model visits = sex race marital private education / dist=poisson;
    run;

    data temp;
      set temp;
      Replicate=&i;
  %end;
%endmacro;

```

```

run;

data temp2;
  set temp2;
  Replicate=&i;
run;

data temp3;
  set temp3;
  Replicate=&i;
run;

proc append base=jkparms data=temp;
run;

proc append base=jkdf data=temp2;
run;

proc append base=converge data=temp3;
run;

%end;

data FullSample;
  set FullSample;
  rename estimate=estimate0;
run;

ods select all;
options &_nopt;
%mend jkloop;

%jkloop(replicates=&replicates)

```

Step 4: Compute the Jackknife Variances and Print the Results

In the fourth step, you merge the full-sample parameter estimates, the parameter estimates from the R replicates, and the jackknife coefficients into a single data set; compute the jackknife variances of the parameter estimates; and print the results.

Because generalized linear models are not guaranteed to converge and because the full model might not be defined in some replicate samples, the following statements check to see how many of the replicate models both converged and have the same number of parameters as the full-sample model. This number is retrieved and saved in the macro variable `&R`. The number is used later to compute confidence intervals for the parameter estimates.

```

ods output Statistics=statistics(keep=replicate sum);
proc surveymeans data=jkdf sum;
  var df;
  by replicate;
run;

```

```

data statistics;
  set statistics;
  full=ifn(sum=&p,0,1);
run;

data converge;
  merge converge statistics;
  by replicate;
run;

data converged;
  set converge(where=(Status=0 & full=0));
run;

data nobs;
  dsid=open("converged");
  converged_replicates=attrn(dsid, "nobs");
  call symput('R',converged_replicates);
run;

```

The following statements create the data set JK by sorting and merging the data sets JKparms, Converge, FullSample, and JKcoefs, which contain the parameter estimates from the replicate models, the convergence status of the replicate models, the parameter estimates that were obtained by using the full sample, and the jackknife coefficients, respectively:

```

proc sort data=jkparms;
  by parameter level1;
run;

proc sort data=FullSample;
  by parameter level1;
run;

data jk;
  merge jkparms FullSample;
  by parameter level1;
run;

proc sort data=jk;
  by replicate parameter level1;
run;

data jk;
  merge jk jkcoefs converge;
  by replicate;
run;

```

The next statements create the data set JKconverged by subsetting the data set JK so that JKconverged contains only parameter estimates from the replicate models that converged and that have the same number of parameters as the full-sample model. The variable SqrDev is created by computing the weighted squared deviations of the parameter estimates; the jackknife coefficients are used as the weights. JKconverged is then sorted by the variables Parameter and Level1.

```

data jkconverged;
  set jk(where=(Status=0 & full=0));
  sqrdev=JKCoefficient*(estimate-estimate0)**2;
run;

data vce;
  set jkconverged(keep= replicate parameter
                  level1 estimate estimate0);
  diff=estimate0-estimate;
run;

proc sort data=jkconverged;
  by parameter Level1;
run;

```

The following statements compute the sum of squared deviations of the parameter estimated by using PROC SURVEYMEANS. The computed sums are in fact the jackknife variances of the parameter estimates. The ODS OUTPUT statement saves the computed variances in the data set JKvariance.

```

ods output Statistics=jkVariance;
proc surveymeans data=jkconverged sum plots=none;
  var sqrdev;
  by parameter Level1;
run;

```

The following DATA step merges the data set JKvariance, which contains the jackknife variances, with the data set FullSample, which contains the full-sample parameter estimates. The variable StdErr is created by computing the square roots of the variances; the full covariance matrix of the parameter estimates is computed later. The variables UL and LL are also created to contain the 95% confidence limits of the parameter estimates.

```

data jkVariance(drop=stddev varname);
  merge jkVariance fullsample(rename=(estimate0=Estimate));
  by parameter Level1;
  StdErr=sqrt(Sum);
  rename Sum=Variance;
  DF=&R - &H;
  t=quantile('T', .975, &R-&H);
  ul=estimate+t*stderr;
  ll=estimate-t*stderr;
  label ul="Upper 95% CL";
  label ll="Lower 95% CL";
run;

```

The following statements print the parameter estimates, the standard errors, the degrees of freedom, and the 95% confidence limits:

```

ods select all;
title "Survey Poisson Regression";
title2 "with Delete-1 Jackknife Variance Estimation";
proc print data=jkVariance noobs label;
  var Parameter Level1 Estimate StdErr DF ll ul;
run;

```

```
title;title2;
```

Output 1 displays the parameter estimates, the jackknife standard errors, the degrees of freedom, and the 95% confidence limits. The table displays how the numbers of visits made by different groups are different. For example, the average number of visits made by a female is $\exp(0.08)$ times higher than the average number of visits made by males, after adjusting for race, education, marital status, and private insurance coverage in the study population. However, because the 95% confidence interval contains 0, the difference is not statistically significant at the 0.05 level.

Output 1 Parameter Estimates and Jackknife Confidence Intervals

**Survey Poisson Regression
with Delete-1 Jackknife Variance Estimation**

Parameter	Level1	Estimate	StdErr	DF	Lower	Upper
					95% CL	95% CL
Intercept		0.2854	0.10767	195	0.07304	0.49772
education	1	0.0412	0.09177	195	-0.13976	0.22221
education	2	0.1228	0.06841	195	-0.01214	0.25770
education	3	0.0355	0.06498	195	-0.09266	0.16365
education	4	-0.0217	0.06656	195	-0.15294	0.10960
education	5	0.0000	0.00000	195	0.00000	0.00000
marital	1	0.0366	0.06832	195	-0.09816	0.17133
marital	2	0.0489	0.06579	195	-0.08083	0.17868
marital	3	0.2383	0.05623	195	0.12739	0.34917
marital	4	0.0000	0.00000	195	0.00000	0.00000
private	1	1.3705	0.06588	195	1.24061	1.50047
private	2	-0.0805	0.06181	195	-0.20245	0.04137
private	3	-0.1291	0.09129	195	-0.30919	0.05090
private	4	0.0000	0.00000	195	0.00000	0.00000
race	1	0.1219	0.08017	195	-0.03617	0.28006
race	2	0.2789	0.09114	195	0.09912	0.45863
race	3	0.0000	0.00000	195	0.00000	0.00000
sex	1	0.0848	0.04523	195	-0.00444	0.17398
sex	2	0.0000	0.00000	195	0.00000	0.00000

Step 5: Compute the Full Jackknife Covariance Matrix

In the fifth and final step, you use statements such as the following to generate the covariance matrix of the parameter estimates, which you need if you want to perform hypothesis tests that involve more than one parameter:

```
proc sort data=jkdf;
  by replicate parameter level1;
run;

data temp;
  merge vce jkdf;
  by replicate parameter level1;
```

```

run;

proc transpose data=temp (where=(df=1)) out=temp2(drop=_name_) prefix=parm;
  by replicate;
  var diff;
run;

data temp3(drop=donorstratum);
  merge temp2 jkcoefs;
  by replicate;
  do i=1 to &p;
    row=i;
    output;
  end;
run;

data temp3(drop=parm: jkcoefficient i j);
  set temp3;
  array col[&p];
  array parm[*] parm;
  do i=1 to &p;
    if row=i then do;
      do j = 1 to &p;
        col[j]=jkcoefficient*parm[i]*parm[j];
      end;
    end;
  end;
run;

proc sort data=temp3;
  by row;
run;

ods select none;
ods output Statistics=statistics(drop=StdDev);
proc surveymeans data=temp3 sum plots=none;
  var coll-coll14;
  by row;
run;
ods select all;

proc transpose data=statistics out=CovB(drop=_name_ row) prefix=parm;
  var sum;
  by row;
run;

proc print data=covb noobs;
run;

```

Output 2 displays the covariance matrix.

Output 2 Parameter Estimates Covariance Matrix

parm1	parm2	parm3	parm4	parm5	parm6	parm7	parm8
0.011592	-0.002057	-0.002977	-0.002101	-0.002247	-0.001635	-0.001023	-0.001872
-0.002057	0.008421	0.002328	0.001955	0.001429	-0.000579	-0.000520	-0.000238
-0.002977	0.002328	0.004680	0.002394	0.002534	0.000059913	-0.000012181	0.000237
-0.002101	0.001955	0.002394	0.004223	0.002354	0.000010428	0.000209	0.000257
-0.002247	0.001429	0.002534	0.002354	0.004430	0.000297	0.000194	0.000095299
-0.001635	-0.000579	0.000059913	0.000010428	0.000297	0.004668	0.001877	0.001714
-0.001023	-0.000520	-0.000012181	0.000209	0.000194	0.001877	0.004328	0.001706
-0.001872	-0.000238	0.000237	0.000257	0.000095299	0.001714	0.001706	0.003161
-0.002992	0.000445	-0.000311	-0.000263	-0.000622	0.000139	-0.000478	0.000479
-0.002539	0.000262	-0.000336	-0.000492	-0.000470	0.000295	-0.000080900	0.000203
-0.002066	-0.000020050	-0.000905	-0.000900	-0.000869	0.001283	0.000821	0.000504
-0.005925	0.000185	0.000299	0.000306	0.000372	-0.000067801	-0.000177	-0.000323
-0.006169	0.000172	0.000561	0.000008650	0.000142	0.000103	-0.000350	-0.000230
-0.001456	0.000033987	0.000466	-0.000108	-0.000077882	-0.000332	-0.000416	0.000121

parm9	parm10	parm11	parm12	parm13	parm14
-0.002992	-0.002539	-0.002066	-0.005925	-0.006169	-0.001456
0.000445	0.000262	-0.000020050	0.000185	0.000172	0.000033987
-0.000311	-0.000336	-0.000905	0.000299	0.000561	0.000466
-0.000263	-0.000492	-0.000900	0.000306	0.000008650	-0.000108
-0.000622	-0.000470	-0.000869	0.000372	0.000142	-0.000077882
0.000139	0.000295	0.001283	-0.000067801	0.000103	-0.000332
-0.000478	-0.000080900	0.000821	-0.000177	-0.000350	-0.000416
0.000479	0.000203	0.000504	-0.000323	-0.000230	0.000121
0.004340	0.002663	0.002782	0.000424	0.000725	0.000012122
0.002663	0.003821	0.003159	0.000039949	0.000069672	-0.000154
0.002782	0.003159	0.008334	-0.000706	-0.000586	0.000038905
0.000424	0.000039949	-0.000706	0.006427	0.005783	0.000355
0.000725	0.000069672	-0.000586	0.005783	0.008307	0.000708
0.000012122	-0.000154	0.000038905	0.000355	0.000708	0.002046

References

- Chambers, R. L., and Skinner, C. J. (2003). *Analysis of Survey Data*. Chichester, UK: John Wiley & Sons.
- Fuller, W. A. (2009). *Sampling Statistics*. Hoboken, NJ: John Wiley & Sons.
- Godambe, V. P., and Thompson, M. E. (1986). "Parameters of Superpopulation and Survey Population: Their Relationships and Estimation." *International Statistical Review* 54:127–138.
- Kish, L., and Frankel, M. R. (1974). "Inference from Complex Samples." *Journal of the Royal Statistical Society, Series B* 36:1–37.
- Korn, E. L., and Graubard, B. I. (1999). *Analysis of Health Surveys*. New York: John Wiley & Sons.
- Pfeffermann, D. (1993). "The Role of Sampling Weights When Modeling Survey Data." *International Statistical Review* 61:317–337.