A picture containing drawing

Description automatically generated

May 18, 2020

We are currently assembling a database of observed survey parameters to help WHO staff and consultants when planning vaccination coverage cluster surveys. This document is meant to guide Stata users to extract the parameters of interest. There is a companion document to help R users accomplish the same task.

We are also providing a set of user-written Stata programs that you can install on your computer that will automate the tasks described here, but it is also possible to do each of the commands interactively and we want the reasoning behind the syntax to be clear. We welcome your suggestions and comments and questions.

We are also happy to work with you to do the data management to prepare your datasets or extract these quantities from your dataset. If you want some help, just let us know. Contact details are at the end of this document.

**Preparing your dataset**

This section describes the variables you need to assemble or create and relays the names that we use for these variables in our work. In many cases I think you will already have a variable that meets the requirements of this work and you will not need to edit your dataset or change your variable names. In other cases, you may find it helpful to make a new dataset that serves as an input for this work, and change the way you have coded some of your variables or add one or two variables to your dataset, to make your dataset compatible with our programs. If so, you may also change your variable names to be compatible with ours, but it is not required.

Example 1. Consider a post-campaign coverage survey (PCCS) conducted in a country with four sub-national strata (call these “states” or *admin1* strata). Each stratum had 40 clusters and each cluster had from 0 to 10 respondents in the age range of interest (9m-14y). Suppose there is only one outcome of interest here: whether the child received the campaign dose.

The dataset should have one row per child in the eligible group and should include variables that capture the following concepts:

1. Unique stratum identifier (This is a numeric variable – an integer – and we use the variable name **stratumid** – all lower case with no underscores, but you may call it anything you like.) When we say ‘stratum’ we mean the survey sampling strata – or the domains in which the survey was designed to provide representative results. In a MICS these might be sub-national or “admin1” strata, like states of Nigeria; in a DHS these might be the urban or rural subsets of admin1 strata. If it’s not clear how the sampling was done, hopefully the survey report can clarify this question. This variable must not be missing. In this example, the stratumid might take on the values 1, 2, 3, and 4. (Or any four integers.)  
     
   Note: If your survey was only conducted in a single stratum and the dataset does not already contain a stratumid, then please generate a variable named stratumid and set all its values equal to 1 and either give it a value label with the name of the stratum or generate an accompanying stratumname string variable and set it equal to the name of the stratum.
2. Cluster identifier – an integer variable that is unique within each stratum. We usually name this variable **clusterid**. Of course the word *cluster* here is a synonym for *primary sampling unit*. In this example, clusterid could take values from 1 up to 160, meaning that every cluster in the country has a unique ID, or it could take value of 1-40 in each stratum. This variable must not be missing.
3. Outcome variable – binary integer outcome, coded as 1 if the child is in the eligible group and received the outcome of interest (in this case the campaign dose) and 0 if the child is in the eligible group and did not receive the outcome, and coded as missing (.) if the child is not in the eligible group. For PCCS work we often name this **got\_sia\_dose** but you can name it anything. Note that this variable must be coded in this manner: 1 means they have the outcome of interest and 0 means they do not and missing means they are not in the eligible group of interest (maybe too old or too young).
4. Weight variable – numeric variable indicating the relative number of kids in the population who are represented by this respondent. If your survey was *self-weighted* or *unweighted*, your survey may not have a weight variable. In that case for purposes of this analysis you could generate a weight variable and set it equal to 1 for every eligible respondent or you can specify the “selfweighted” option in our data extraction program – see the Stata and R guidance documents for details. We usually call our weight variable **psweight**. It does not need to be an integer.

Additional variables:

1. If you have it, it will be interesting to also see an indicator of whether the cluster was urban or rural. In our work we usually name this **urban\_cluster** and we code it as 1 if the cluster is urban and 0 if the cluster is rural. We use an integer variable and we apply a value label.
2. Other stratification variables.

We are interested in generating results at the national level, the national urban and rural levels, and the level of admin1 states or provinces, so if the survey strata were not the admin1 strata, you will need an integer variable to identify those. In our work we call this **level3id** and we usually have an accompanying string variable named **level3name**.

For example, if you were working with a DHS survey, the stratumid might contain a unique value for the urban portion of a state and another unique value for the rural portion of that same state, but level3id would contain the same value for the urban and rural portions of the state.

Perhaps your admin1 strata are sometimes grouped together. In Nigeria the admin1 strata are states, but reports often group states into six zones. We code those with an integer variable named **level2id** and we provide an accompanying string variable named **level2name.**

Your survey may have collected data on another cluster-level characteristic that would be interesting to report. For example, in Nepal, the country is divided into provinces, but also divided into ecological belts. Nepal sits on the side of the Himalayan mountains, so the belts correspond to areas of low- and medium- and high-elevation. They are named the plain (Terai), mountainside without snow (Hill), and the mountains portion that contains snow (Mountain or Himal). The dataset might include an integer variable to categorize whether the PSU is from the Terai, Hill, or Himal. This could be an integer with a value label.

Additional outcomes:

Some surveys will have a single outcome variable, and some will have several. See below that we have included several outcomes in the example RI dataset.

**Example datasets**

In the Demo.zip file accompanying this document we made three example datasets available:

sia\_example\_dataset\_Harmonia

* **clusterid** (integer)
* **stratumid** (integer – takes the values 1-10)
* **psweight** (double precision real number)
* **urban\_cluster** (integer – with value label – takes values 0 or 1)
* **got\_sia\_dose** (integer, coded 0 or 1 or . )
* **level3id** (integer holding the district level outcomes)
* **level3name** (string holding the district names)
* **level2id** (integer holding the province level outcomes)
* **level2name** (string holding the province names)
* **level1id** (integer holding the single country ID)
* **level1name** (string holding the country name)

tt\_example\_dataset\_Harmonia

Same variables as above, except the outcome variable is:

* **protected\_at\_birth\_to\_analyze** (integer, coded 0 or 1 or . )

ri\_example\_dataset\_Harmonia

Same variables as above, but with five outcome variables:

* **had\_card** (integer coded 1/0/. – respondent showed a card to the survey interviewer)
* **got\_crude\_bcg\_to\_analyze** (integer coded 1/0/. – Received BCG)
* **got\_crude\_penta1\_to\_analyze** (integer coded 1/0/. – Received Penta1)
* **got\_crude\_penta3\_to\_analyze** (integer coded 1/0/. – Received Penta3)
* **got\_crude\_mcv1\_to\_analyze** (integer coded 1/0/. – Received Measles)

**Extracting parameters from your dataset**

Once the dataset is prepared, you can run our program named iccloop to extract all the parameters of interest. You can type ‘help iccloop’ to read about the inputs. We describe them briefly below.

First, example syntax. To extract national level ICC estimates from the dataset, we do the following Stata commands:

\* cd “<the full pathname of the folder where you have saved ri\_example\_dataset\_Harmonia.dta>”

cd “C:/ICC examples”

iccloop id(1) row(1) data(sia\_example\_dataset\_Harmonia) outcome(got\_sia\_dose) output(sia\_icc\_dataset) stratvar(level1id) ///  
 stratnamevar(level1name) agegroup(9m-14y) comment(2016 Measles SIA) replace

Then to add parameters at the district level, run the command again, but without the “replace” option.

iccloop id(1) row(1) data(sia\_example\_dataset\_Harmonia) outcome(got\_sia\_dose) output(sia\_icc\_dataset) stratvar(level3id) ///  
 stratnamevar(level3name) agegroup(9m-14y) comment(2016 Measles SIA)

Then to add parameters at the urban and rural levels, run the command a third time, without the replace and stratnamevar options.

iccloop id(1) row(1) data(sia\_example\_dataset\_Harmonia) outcome(got\_sia\_dose) output(sia\_icc\_dataset) stratvar(urban\_cluster) ///  
 agegroup(9m-14y) comment(2016 Measles SIA) replace

What is iccloop and what do those options mean?

The Stata command iccloop was written at Biostat Global Consulting for this project. It takes a set of inputs and extracts the parameters of interest from a vaccination coverage cluster survey dataset. The details for each parameter are described later in the document. The command may be run interactively from the Stata command line, or as part of a saved .do file. We recommend running it as part of a .do file and we provide an example file that runs with the Harmonia example datasets.

Required inputs:

* id(integer) – this is a unique ID number assigned to your survey for the purpose of this project – it is for the purposes of tracking all the different surveys from all our different collaborators – we will tell you the study ID numbers for your surveys when you write to describe the survey to us; you can use the value 1 if we haven’t given you a specific value yet.
* data(dataset name) – this is the name of the Stata dataset that holds the variables described above. Note that the dataset should only include respondents in the age group of interest. If you interviewed children 12-23m and 24-35m about vaccination coverage, we are mostly interested in the parameters for children age 12-23m, so you should make a new dataset that holds only those children. If you decide to also report ICCs, etc., for the older children, please make a separate dataset that holds the data for them and run iccloop on that dataset separately.
* outcome(variable list) – list of outcome variable names. For an SIA survey, this might hold only one name; for an RI survey, it might hold four or five names. The variable names should be separated by a space, but no punctuation:
  + Correct: outcome(got\_sia\_dose)
  + Incorrect: outcome(had\_card, got\_crude\_bcg\_to\_analyze, got\_crude\_penta1\_to\_analyze, got\_crude\_penta3\_to\_analyze)
  + Correct: outcome(had\_card got\_crude\_bcg\_to\_analyze got\_crude\_penta1\_to\_analyze got\_crude\_penta3\_to\_analyze)
* stratvar(variable name) – name of the variable that defines the groups for this analysis; typically level1id, level2id, level3id, or urban\_cluster; the iccloop program will generate one row of data per level of the stratvar
* output(dataset name) – name of the Stata dataset where the results should be appended

Optional inputs:

* row(integer) – this is another ID number assigned to your survey – we will tell you what value to use; use 1 as a default
* stratlevel(integer) – We use 1 for national results, 2 for sub-national results, and 3 for sub-sub-national results; not required
* replace – means that if the output dataset exists, you want to replace it with the new results; if you do not say ‘replace’ and the output dataset exists, then iccloop will append new results to that dataset
* stratnamevar(variable name) – name of the variable that holds the names for the stratvar values; typically level1name, level2name, or level3name
* agegroup(string) – please enter a short string that describes the age of respondents eligible for your survey; this will be passed thru to the output dataset
* comment(string) – this string will pass through to the output dataset; sometimes we write the name of the SIA dose here, e.g., Measles-Rubella; Japanese Encephalitis, etc.
* svyset(string) – this string holds the svyset syntax to use with your dataset. If you do not provide it, the program assumes that the correct syntax is:   
  **svyset clusterid, strata(stratumid) weight(psweight) singleunit(scaled)**  
  If your variables are not named clusterid and stratumid, then you need to be sure to specify this option. Contact us if you need help with this.
* weightvar(variable name) if your weight variable is not named psweight, provide the name of the weight variable here
* selfweighted – if your survey was self-weighted or un-weighted, you can specify the option selfweighted and iccloop will do the right thing. Your other alternative is to create a weight variable and set the value equal to 1 for every row in the dataset

We will be happy to work with you to identify the correct set of options for iccloop, depending on your situation.

Resulting dataset

The command iccloop makes a small dataset with one row per level of stratvar that records the following parameters.

| Parameter | Label | Description |
| --- | --- | --- |
| survey\_id | Dataset ID Number | this is an ID number for our ICC paper study that we assign to your survey; we will tell you what number to use. |
| unique\_survey\_row\_id | Unique Row ID Within Survey | Same; this is a number for our dataset purposes; we will assign it here. |
| path\_to\_dataset | Pathname to Dataset | The program records where the survey dataset is located on your computer |
| dataset\_name | Dataset File Name |  |
| agegroup | Age Group for this Outcome | This is passed through if you specify a string using iccloop’s agegroup option |
| outcome\_name | Outcome Name | Name of the outcome variable |
| stratum\_level | Geographic Level | Passed through if you use the stratlevel option; not required |
| stratvar | Stratification Variable | Required; name of variable specifying different stratum levels |
| stratum\_id | Stratum ID Number | iccloop returns parameters from every level of stratvar |
| stratum\_name | Stratum Name | If you specify stratnamevar, iccloop takes the name from that; otherwise it takes the name from the value label of stratvar |
| path\_to\_opplot | Pathname to Organ Pipe Plot | Where on your computer did iccloop save the organ pipe plot |
| opplot\_name | Organ Pipe Plot Name | File name of the organ pipe plot for this outcome and stratum |
| n\_clusters | Number of Clusters | How many clusters in this stratum |
| n\_strata | Number of distinct survey sample strata in this calculation | Number of survey sampling strata in this stratum |
| n | Number of Respondents | Number of respondents with a 0 or 1 outcome in this stratum |
| m\_avg | Average Number of Respondents Per Cluster |  |
| m\_avg\_wtd | Weighted average cluster size | This is a weighted parameter often listed in the ICC literature |
| m\_sd | Standard Deviation of No. Respondents per Cluster | To document variability in cluster sample size |
| m\_min | Min No. Respondents per Cluster |  |
| m\_max | Max No. Respondents per Cluster |  |
| ncp100 | Number of Clusters with 100% Coverage | The ncp100 and ncp0 parameters help us identify reasons for very high ICCs |
| ncp0 | Number of Clusters with 0% Coverage | The ncp100 and ncp0 parameters help us identify reasons for very high ICCs |
| p | Coverage Estimate | Weighted estimate of outcome coverage (0-100%) |
| p\_se | Coverage Estimate Standard Error | Standard error (on the 0-1 scale…not 0-100% scale) |
| p\_lb | Coverage Estimate CI Lower Bound | Two-sided 95% CI bound (see ci\_method for the type of confidence interval that was calculated) |
| p\_ub | Coverage Estimate CI Upper Bound | Two-sided 95% CI bound (see ci\_method for the type of confidence interval that was calculated) |
| ci\_method | Confidence Interval Method | iccloop calculates a Wilson interval if possible by default; if you change your copy, I want to know what kind of interval you calculated, so this string holds the name of the method |
| deff | Design Effect | This is the observed deff, accounting for weights and stratification and clustering |
| neff | Effective Sample Size | Equals n / deff |
| deff\_nostrat\_or\_wt | Design Effect w/o Stratification or Weights | This deff ignores stratification and weights and should be due only to the clustering effect |
| cvw | Coefficient of Variation on Weights |  |
| icc | Intracluster Correlation Coefficient | Calculated using the mean squares and average cluster size from Stata’s loneway command, but we have modified the command slightly to return a negative ICC when observed, rather than clip the ICC at 0. Our modified program is named loneway\_plus.ado. |
| icc\_lb | Intracluster Correlation Coefficient CI Lower Bound | Calculated using bootstrap (preferred) or loneway |
| icc\_ub | Intracluster Correlation Coefficient CI Upper Bound | Calculated using bootstrap (preferred) or loneway |
| icc\_ci\_method | ICC CI method | String to tell us how the CI was calculated |
| ms\_between | One-way ANOVA mean squares between clusters | We use this to calculate ICC using Ridout’s ANOVA formula (and allowing negative values) |
| ms\_within | One-way ANOVA mean squares within clusters | We use this to calculate ICC using Ridout’s ANOVA formula (and allowing negative values) |
| comment | Comments | Passed through from call to iccloop; you might put the name of the SIA dose in the comment or some other important info |
| p\_lb\_logit | Coverage Estimate CI Lower Bound (Logit) |  |
| p\_ub\_logit | Coverage Estimate CI Upper Bound (Logit) |  |
| p\_lb\_agresti | Coverage Estimate CI Lower Bound (Agresti) |  |
| p\_ub\_agresti | Coverage Estimate CI Upper Bound (Agresti) |  |
| p\_lb\_clopper | Coverage Estimate CI Lower Bound (Clopper) |  |
| p\_ub\_clopper | Coverage Estimate CI Upper Bound (Clopper) |  |
| p\_lb\_fleiss | Coverage Estimate CI Lower Bound (Fleiss) |  |
| p\_ub\_fleiss | Coverage Estimate CI Upper Bound (Fleiss) |  |
| p\_lb\_jeffreys | Coverage Estimate CI Lower Bound (Jeffreys) |  |
| p\_ub\_jeffreys | Coverage Estimate CI Upper Bound (Jeffreys) |  |
| p\_lb\_wilson | Coverage Estimate CI Lower Bound (Wilson) |  |
| p\_ub\_wilson | Coverage Estimate CI Upper Bound (Wilson) |  |
| p\_lb\_wald | Coverage Estimate CI Lower Bound (Wald) |  |
| p\_ub\_wald | Coverage Estimate CI Upper Bound (Wald) |  |

If all goes well, you call iccloop and it produces a dataset with one row per stratum, listing 48 different parameters. If you specify the ‘replace’ option, then it over-writes earlier results. If you do not specify the ‘replace’ option, then it appends the new output to any earlier output, so our usual practice is to:

1. Call iccloop to generate one row of national parameter estimates and use the ‘replace’ option.
2. Call iccloop to generate one row per admin1 stratum, and do not use the ‘replace’ option.
3. Call iccloop one more time to generate one row for urban and one row for rural parameters, and do not use the ‘replace’ option.

The result will be a dataset with 3 rows (national, national urban, national rural) plus one row per admin1 stratum.

In some cases there may be a natural grouping of admin1 strata: for instance in Nigeria the states are often aggregated into zones. So you might call iccloop another time to calculate parameters at that aggregated level.

1. stratvar(level1id) stratnamevar(level1name) // national results for all clusters; we specify the stratnamevar option because level1id does not have a value label so we need to get the stratum name from the string variable level1name
2. stratvar(level3id) stratnamevar(level3name) // one row per admin1 survey stratum
3. stratvar(urban\_cluster) // one row for urban and one for rural; no stratnamevar() option needed because urban\_cluster has a variable label
4. stratvar(level2name) stratnamevar(level2name) // one row per aggregated ‘zone’ stratum

**Organ pipe plots**

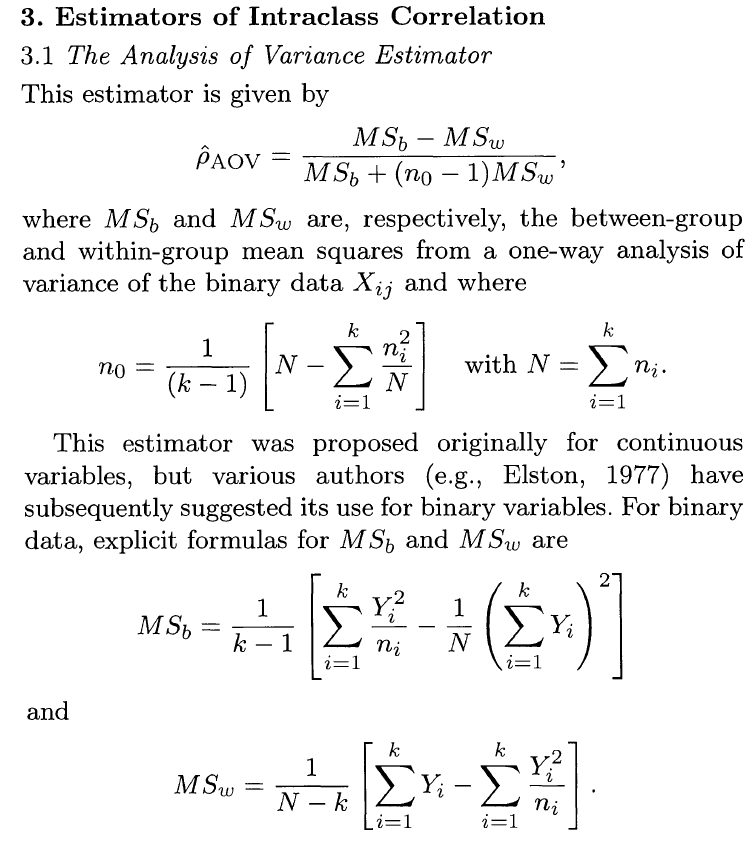
The iccloop program will make an organ pipe plot for each stratum. It will save the plot on your computer. We would like you to send the organ pipe plots (.gph files and .png files) to us along with the dataset of parameters. You can read about organ pipe plots at (World Health Organization 2018; Rhoda and Prier 2019).

**Stata syntax for key parameters**

Some of you will be interested in the Stata syntax that we use to estimate the key parameters for this study. The zip file that accompanies this document holds a .do file named “Demonstrate parameter extraction.do”. You may run that program line-by-line to see the syntax we use to generate each of the 48 parameters for a single analysis. The zip file also holds a .do file named “Demonstrate iccloop on Harmonia data.do” which runs iccloop four times for TT data, four times for SIA data and four times for RI data and then makes a simple plot.

ICC

We calculate the ANOVA estimator described in Ridout et al., 1999, implemented in the calcicc.ado program.



The syntax is: loneway binary\_outcome\_variable clusterid

ICC 95% confidence interval

Our calcicc.ado command bootstrapped 1000 times where the bootstrap sample consists of subsets of clusters selected with replacement in each stratum. If the bootstrap will not run because of odd sample strata (with only a single PSU) then we use the CI from the loneway command run once with the full sample.

Vaccination coverage

We estimate coverage using Stata’s svy: proportion command. Importantly, we use the subpop syntax to be sure that degrees of freedom account properly for clusters in the sample that did not yield respondents in the age group of interest.

Vaccination coverage confidence intervals

We estimate the 95% CIs using Biostat Global Consulting’s svypd.ado command, which is part of VCQI.

Coverage design effect

Design effect is calculated using the svypd.ado command. It may also be calculated using the command “estat effects” after using the svy: proportion command.

Coverage design effect without weights or stratification

Change the svyset command to include only the clusterid, but not the strata or weights, then re-estimate coverage and re-calculate the design effect. The second estimate of the design effect is what we will use to estimate the parameter that Kalton et al., 2005 call *roh*, the *rate of homogeneity*. We will calculate roh in a post-processing step after you send your data to us to be appended to the project database.

Coefficient of variation of weights

Use Stata’s summarize command to calculate the mean and standard deviation of weights for eligible respondents with either a 0 or 1 outcome. CV = standard deviation / mean

Organ pipe plots

We generate organ pipe plots using Biostat Global Consulting’s opplot.ado command, which is part of VCQI.

**Instructions**

1. Navigate to the project Dropbox
2. Copy the folder named ‘Resources for working in Stata’ to your computer
3. Copy the Stata .ado files and their accompanying .sthlp files to a folder that is in your Stata adopath. If you type the command “adopath” at the command prompt , Stata will tell you the name of your so-called PERSONAL folder. I suggest you save the .ado programs from this project in your PERSONAL folder (at least temporarily).
4. Restart Stata.
5. Type the command “which iccloop” at the Stata prompt.
6. Stata should find the iccloop.ado file in your PERSONAL folder. If so, you are ready to try the demo.
7. Navigate to where you saved the ‘Demo’ folder on your computer.
8. Edit the file named “Demonstrate iccloop on Harmonia data.do”.
9. Change the cd command at the top of the program to point to the location of the ‘Demo’ folder on your computer.
10. Run the .do file.
11. If it does not run successfully, contact Dale for some assistance.
12. Now prepare your own survey dataset(s).
13. Make a copy of “Demonstrate iccloop on Harmonia data.do” and edit it to run on your dataset(s).
14. Contact Dale for assistance, if needed.

**Questions**

Contact Dale Rhoda ([Dale.Rhoda@biostatglobal.com](mailto:Dale.Rhoda@biostatglobal.com))

Whatsapp at +1 614 499-2351

Skype: dalerhoda

**References**

Kalton, Graham, J Michael Brick, and Thanh Lê. 2005. “Chapter VI Estimating Components of Design Effects for Use in Sample Design.” In Household Sample Surveys in Developing and Transition Countries, 95–122. United Nations Statistics Division.

Rhoda, Dale A., and Mary L. Prier. 2019. *OPPLOT: Stata Module to Generate a Vertical Bar Chart to Summarize a Binary Outcome in Cluster Survey Data* (version 1.13). Stata. Stata. VCQI. Biostat Global Consulting. https://ideas.repec.org/c/boc/bocode/s458627.html.

World Health Organization. 2018. “Vaccination Coverage Cluster Surveys: Reference Manual.” World Health Organization. http://www.who.int/immunization/documents/who\_ivb\_18.09/en/.

Ridout, Martin S., Clarice GB Demétrio, and David Firth. 1999. “Estimating Intraclass Correlation for Binary Data.” Biometrics 55 (1): 137–148.