

Multiple imputation details and purpose

In the 2017 NSCH, missing values were imputed for several demographic variables used in the construction of survey weights. Child sex, race, and Hispanic origin were imputed using hot-deck imputation while Adult 1 education and household size were imputed using sequential regression imputation methods. Total family income was also imputed using sequential regression as an input to the family poverty ratio (FPL). Imputation is useful because it uses observed data to estimate a plausible response for a missing value. It is preferable to preserve sample size and avoid bias in only using observed or known values in a “complete-case” analysis, which assumes that data are missing completely at random. In particular, 16.03% of the sample (18.21% of the weighted sample) was missing one or more components of FPL, which varied by other known demographic characteristics and would have severely limited sample size and biased estimates to only use the known or reported data.

Using the same sequential regression imputation methods, FPL was also multiply imputed and contains six versions or implicates. Multiple imputation creates several plausible responses for a missing value, using other variables in the dataset to adjust the missing response (Allison, 2001; Rubin, 1996; Schaefer and Graham, 2006). These multiple imputations offer a means of accurately incorporating the uncertainty of the imputed values for missing items. More specifically, combining or averaging estimates across all six imputed values will appropriately increase the standard error to account for this uncertainty while only slightly altering the point estimates. Using only a single imputation, particularly with a large amount of missing data as in the case of FPL, incorrectly assumes certainty in the imputation as if there were no missing data at all—and will produce standard errors that are too low and tests of significance that are too high (increased Type 1 error).

In contrast to the 2016 NSCH, in which the imputed file was released separately and required merging, the 2017 public use file includes all six imputed values for FPL [FPL_I1-FPL_I6]. This document includes example code to show how to analyze multiply imputed FPL data using SAS, SAS-callable SUDAAN, and Stata. These procedures or commands will appropriately combine or average the point estimates across implicates and increase standard errors so that significance levels are not overstated. The term implicate will be used in this documentation, although other sources may use imputation (StataCorp LP, 2013).

Analyzing data in a multiple imputation framework

The NSCH public use file contains the imputed values stored in different variables, one for each of the imputed responses. These variables contain both fully reported and imputed values. Table 1 shows an example dataset, a wide file, with FPL_I1 -- FPL_I6. For the case ID 1, the FPL_I1 -- FPL_I6 are not identical because there was missing data on either income or household count and these values are imputed. For the case ID 2, the poverty ratio variables are identical because there was no missing data. SAS-callable SUDAAN and Stata can accommodate the wide dataset form.

Table 1. Example of a wide dataset with an imputed observation

ID	SC_AGE	FPL_I1	FPL_I2	FPL_I3	FPL_I4	FPL_I5	FPL_I6
1	10	125	135	100	90	130	115
2	16	250	250	250	250	250	250

Table 2 shows how the dataset needs to be re-organized to do analyses using the multiple imputation variables in SAS with 6 stacked rows of complete data for each observation, one for each implicate. In this long dataset, the variable 'Implicate' reflects the implicate number 1 through 6. In SAS, the actual variable will be called '_Imputation_'. SAS-callable SUDAAN and Stata can use the long dataset form but it is a less efficient form of storage that requires more computational resources.

Table 2. Example of a long dataset with an imputed observation

ID	SC_AGE	FPL	Implicate
1	10	125	1
1	10	135	2
1	10	100	3
1	10	90	4
1	10	130	5
1	10	115	6
2	16	250	1
2	16	250	2
2	16	250	3
2	16	250	4
2	16	250	5
2	16	250	6

Example

This documentation includes example code for analyzing multiply imputed data in SAS, SAS-callable SUDAAN, and Stata. The example code estimates the proportion of children in four poverty categories by children with special health care needs status (SC_CSHCN). We first create a variable named 'povcat_i' that reflects family income as a percentage of the federal threshold by family composition (1='<100% FPL', 2='100%-199% FPL', 3='200%-399% FPL', 4='400%+ FPL').

How to obtain estimates in SAS:

In SAS, you will need to reshape data from a wide to long format. This data step is included in the example code. In this step we copy the non-imputed variables (e.g. age) in the dataset along with a single FPL variable and FWC variable, until each respondent has six observations in the dataset, one for each implicate (see Table 2).

Once the data have been reshaped, we can use proc surveymeans to get the mean of the variable poor for each imputed dataset. The proc mianalyze procedure will then combine the estimates by averaging the mean across the implicates and calculate the standard error according to Rubin's formula (Rubin, 1996; SAS Institute, 2009).

```

libname file "<<Replace with file directory>>";

/*****
In order to use proc mianalyze, we will need to create a long, or stacked,
dataset.
*****/
data stacked;
  set file.nsch_2017_topical;
  array fpli{6} fpl_i1-fpl_i6;
  do _Imputation_=1 to 6;
    fpl_i=fpli[_Imputation_];

/*****
Creating a four category poverty variable
*****/
    if fpl_i < 100 then povcat_i = 1;
    if 100<=fpl_i<200 then povcat_i = 2;
    if 200<=fpl_i<400 then povcat_i = 3;
    if fpl_i>=400 then povcat_i = 4;
  output;
end;
run;
/*****
Estimate parameter of interest for each implicate after sorting by imputation
*****/
proc sort data=stacked;
by _Imputation_;
run;
proc surveyfreq data=stacked;
strata stratum fipsst; * design statements;
cluster hhid;
weight fwc;
by _Imputation_; * identify the imputation;
tables sc_cshcn*povcat_i / row cl; * request crosstab with row % and CIs;
ods output crosstabs = mi_table ; * estimates stored in new dataset mi_table;
run;

/*****
Combine the implicates using proc mianalyze after sorting by variables of
interest. This applies Rubin's rules (Rubin, 1996) to properly inflate
standard errors for the imputed cases.
*****/
proc sort data=mi_table;
by sc_cshcn povcat_i;
run;
proc mianalyze data=mi_table;
by sc_cshcn povcat_i; * requests data for each combination of cshcn and
poor;
modeleffects rowpercent ; * combined percentage over all imputations;
stderr rowstderr; * combined standard error over all imputations;
run;

```

How to obtain estimates in SAS-callable SUDAAN:

Using SUDAAN, you can leave the data in wide form without re-shaping. A data step is needed to convert the design variables to numeric per SUDAAN requirements and to create the poverty variable. The sorted file can then be analyzed in any procedure using the `mi_var` statement to identify the implicates. The confidence intervals for SUDAAN crosstab rely on the logit transformation and will be slightly different from the normal or symmetric intervals produced in SAS and Stata.

```

/*****
SUDAAN can analyze implicate data in two forms (one wide dataset
or separate datasets for each implicate). This example will show
the easier or more efficient option of a single wide dataset.
*****/
libname file "<<Replace with file directory>>";
data example;
  set file.nsch_2017_topical;
/*****
Converting design variables to numeric per SUDAAN requirements
*****/
hhidnum = input(hhid,8.);
fipsstnum = input(fipsst,8.);
if stratum='2A' then stratum='2';
stratumnum = input(stratum,8.);
/*****
Creating a four category poverty variable
*****/
array fpl_i{6} fpl_i1-fpl_i6;
array povcat_i{6} povcat_i1-povcat_i6;
do i=1 to 6;
  if fpl_i{i}<100 then povcat_i{i} = 1;
  if 100<=fpl_i{i}<200 then povcat_i{i} = 2;
  if 200<=fpl_i{i}<400 then povcat_i{i} = 3;
  if fpl_i{i}>=400 then povcat_i{i} = 4;
end;
drop i;
run;
/*****
Data must be sorted prior to analysis
*****/
proc sort data=example;
by fipsstnum stratumnum hhidnum;
run;
/*****
Analyzing multiple implicate data
*****/
proc crosstab data=example design=wr ;
nest fipsstnum stratumnum hhidnum / psulev=3; * design statements;
weight fwc;
mi_var povcat_i1-povcat_i6; * identifies implicates, called by first
variable listed in remainder of code;
class sc_cshcn povcat_i1;
table sc_cshcn*povcat_i1; * requests crosstab;
print nsum wsum powper serow lowrow uprow /style=nchs nsumfmt=f10.0
wsumfmt=f10.0; * requests row percentages;
run;

```

How to obtain estimates in Stata:

In Stata, just as you declare the data to be svyset, you declare it to be an MI (multiple imputation) dataset. In order for Stata to recognize that a variable has been imputed, you need to use mi import and register the imputed variables. Here, you will declare the FPL variables to be imputed.

Stata makes a missing flag when it imputes variables based on the '.' responses. These missing values are not available in the Public Use File. The work-around we advise is generating the variable FPL_I0 and then setting all values to '.'. Rubin's (1996) formula will calculate the correct variance across imputates regardless of whether the values were imputed or reported.

Once the data have been imported, and mi set, they are ready for analysis. Simply using the 'mi est: svy:' prefix will combine the estimates by averaging across the imputates and calculate the standard error according to Rubin's formula (Rubin, 1996).

```
local file = "<<Replace with file directory>>"
use "`file'\nsch_2017_topical", clear

egen statacross=group(fipsst stratum) /* create single cluster variable for svy */
gen fpl_i0=. /* create missing variable for original fpl, m=0 */

save "`file'\nsch_2017_topical", replace /* must be saved prior to declaring imputation */

mi import wide, imputed(fpl_i0=fpl_i1-fpl_i6) drop /* declare imputed data */

mi passive: generate povcat_i=0 /* generate new variable based on imputed fpl */
mi passive: replace povcat_i=2 if fpl_i0>=100&fpl_i0<200
mi passive: replace povcat_i=3 if fpl_i0>=200&fpl_i0<400
mi passive: replace povcat_i=4 if fpl_i0>=400

mi svyset hhid [pweight=fwc], strata(statacross) /* declare survey data */

mi est: svy: proportion povcat_i, over(sc_cshcn) /* request crosstab of povcat_i by sc_cshcn */
```

References

- Allison, P. D. 2001. Missing Data. Thousand Oaks: Sage Publications.
- Rubin, D.B. 1996. Multiple Imputation After 18+ Years. Journal of the American Statistical Association 91: 473-489.
- Schaefer, J.L. and J.W. Graham. 2002. Missing Data: Our View of State of the Art. Psychological Methods 7(2): 147-177.
- SAS Institute Inc. 2009. SAS/STAT 9.2 User's Guide, Second Edition. Cary, NC: SAS Institute Inc.
- StataCorp, LP. 2013. Stata Multiple-Imputation Reference Manual: Release 13. College Station, TX: Stata Press.