Multiple Imputation in Practice

Trivellore Raghunathan

Part 2: Application

(Slides adapted from course slides jointly taught with Rod Little)

Research Questions

- Surveys rely on self-report of medical conditions
- How accurate are these reporting?
- NHANES allows examining this aspect
- NHANES asks self-report of diabetes
  - 1: No, 2: Prediabetes, 3: Moderate, 4: Definite
  - If (2,3,or 4) medication use (insulin or pills)
- NHANES measures Fasting Glucose and Hemoglobin A1C
- Questions:
  - What is the population proportion undiagnosed diabetics?
  - What is the relationship between SES and being undiagnosed?
- Covariates of interest: Education, marital status and poverty status
- Confounders: Age and gender

NHANES Data 2017-2020

Summary	Variable	DESCRIPTION
NHANES Data codebook for Workshop data	SEQN	ID VARIABLE
Year: 2017-2020	SDMVSTRA	STRATUM
Population: Age 20+	SDMVPSU	PSU
N=9,232	WTINTPRP	SAMPLING WEIGHT
	AGE	AGE IN YEARS
	FEMALE	MALE=0 FEMLAE=1
	EDUC	1: LESS THAN HS; 2: HS; 3: SOME COLLEGE; 4 COMPLETED COLLEGE
	MARSTATUS	1: MARRIED; 2: WIDOWED/DIVORCED/SEPARATED; 3: NEVER MARRIED
	POVERTY	1: <1.30; 2: [1.30,1.85); 3: > 1.85
	DIAB_SR	1: NO 2: PREDIABETES 3: MODERATE 4: DIABETIC
	MEDYES	1: TAKING PILLS OR INSULIN 0: NO (ASKED ONLY IF DIAB_SR > 1)
	LBXGLU	FASTING GLUCOSE
	LBXGH	GLYCOHEMOGLOBIN (A1C)

IVEware

- A collection routines to implement multiple imputation analysis
  - Uses Sequential Regression Multivariate Imputation
  - Standalone or in conjunction with SAS, R, STATA, and SPSS
  - An XML file is used to type in your commands and then submit them
    - There are other ways of running the program within SAS, R, STATA and SPSS
  - Can be used to perform complex survey analysis (account for stratification, clustering and weighting)
  - Unix, Windows and Mac

Routines

- IMPUTE : Create multiple imputations
- DESCRIBE: Descriptive analysis
- REGRESS: Regression analysis
- BBDESIGN: Nonparametric synthetic populations
- SYNTHESIZE: Parametric synthetic populations
- COMBINE: Combining data sets to align common variables
- SASMOD: Use a variety of procs in SAS
- Using the toggle, for example, <R>, R Commands, </R>, you can execute any R programming statement

Workshop Files

- Workshop.sas7bdat: SAS Datafile
- Workshop.txt: Text data file
- Workshop_SAS.xml: Example Program with SAS
- Workshop_R.xml: Example Program file with R
- Workshop_standalone.xml: Example Program for standalone
- Workshopimputed.sas7bdat: Multiply imputed data set
- Workshop_bbdesign.xml: Synthetic population approach

```
<sas name="SASoutput">
/* The command above indicates beginning of commands
to be executed in SAS */
/* Read data */
libname workshop 'e:\workshop';
data one;
set workshop.workshop;
/* Imputation Commands */
<impute name="SASimpout">
datain one;       /* Input data */
dataout oneimp all; /* Imputed data; "all" will stack the data */
default categorical; /* Unless indicated everything is categorical */
continuous lbxglu lbxgh age wtintprp; /* Continuous variables */
transfer seqn sdmvpsu a1c_sr testfora1c; /* Variables to be transferred and not to be used as predictors */
restrict medyes (diab_sr>1); /* Variable Medyes is only to be imputed for dian_sr >1 */
bounds lbxgh (>=2.8, <=16.2) lbxglu(>=47, <=451); /* Bounds for the imputed values to be in the observed range */
iterations 10; /* Number of Iterations */
multiples 10;  /* Number of Imputations */
diagnose lbxglu lbxgh; /* Create diagnostic Plots for these variables */
seed 23456; /* specify seed for replicability */
run; /* Run the commands. End Impute portion */
</impute>
/* Post processing of the stacked imputed data */
data imp_anal;
set oneimp;
undiagnosed=0;
if diab_sr=1 and (lbxglu> 126 or lbxgh > 6.5) then undiagnosed=1;
proc means mean;
var undiagnosed;
```

Toggle SAS (will create SASoutput.sas, SASoutput.log SASoutput.lst)

Read in SAS data

Toggle Impute
Impute commands
End Impute
(will create SASimpout.set SASimpout.sas SASimpout,log SASimpout.lst)

Process imputed data

```
weight wtintprp;
/* Store the imputed data for future use */
data workshop.workshopimputed;
set imp_anal;
/* Macro to split the stacked data into individual data sets */
%macro split;
%do i =1 %to 10;
data anal&i;
set imp_anal;
where _mult_=&i;
%end;
%mend;
%split;
/* Start of the Survey Regression Analysis */
<regress name="SASregout">
datain anal1 anal2 anal3 anal4 anal5 anal6 anal7 anal8 anal9 anal10; /* Ten imputed data sets */
categorical educ marstatus poverty undiagnosed; /* Categorical Variables in the model */
dependent undiagnosed(0);  /* Dependent variable and the reference/denominator value */
predictor age female educ marstatus poverty; /* Predictor Variable */
link logistic;    /* Link function */
stratum sdmvstra;  /* Design Variables */
cluster sdmvpsu;
weight wtintprp;
run;    /* Run the Program and end Regress module */
</regress>
/* Close out SAS */
</sas>
```

Store the imputed data for future use

Macro to create 10 imputed data sets

Toggle regress
Regress commands

End Regress
(will create SASregout.set SASregout.sas SASregout.log SASregout.lst)

End SAS

```
IVEware Iterative Imputation Procedure, Fri Jul 08 11:05:04 2022
1

Imputation 1

Variable                Observed                Imputed    Double counted
WTINTPRP                    9232                      0                 0
SDMVSTRA                    9232                      0                 0
LBXGH                       8081                   1151                 0
LBXGLU                      4004                   5228                 0
age                         9232                      0                 0
female                      9232                      0                 0
educ                        9217                     15                 0
marstatus                   9222                     10                 0
poverty                     7828                   1404                 0
diab_sr                     9229                      3                 0
medyes                      1525                   7707                 0
```

This column is useful to diagnose problems with restrictions, if it is not zero

Variable LBXGH

	Observed	Imputed	Combined
Number	8081	1151	9232
Minimum	2.8	3.20141	2.8
Maximum	16.2	10.2164	16.2
Mean	5.86238	5.89034	5.86587
Std Dev	1.12154	1.08409	1.11692

Variable LBXGLU

	Observed	Imputed	Combined
Number	4004	5228	9232
Minimum	47	47.1618	47
Maximum	451	381.322	451
Mean	113.572	113.837	113.722
Std Dev	38.0963	35.9717	36.9064

Variable poverty

	Observed		Imputed		Combined	
Code	Freq	Per	Freq	Per	Freq	Per
1	2218	28.33	464	33.05	2682	29.05
2	1143	14.60	189	13.46	1332	14.43
3	4467	57.06	751	53.49	5218	56.52
Total	7828	100.00	1404	100.00	9232	100.00

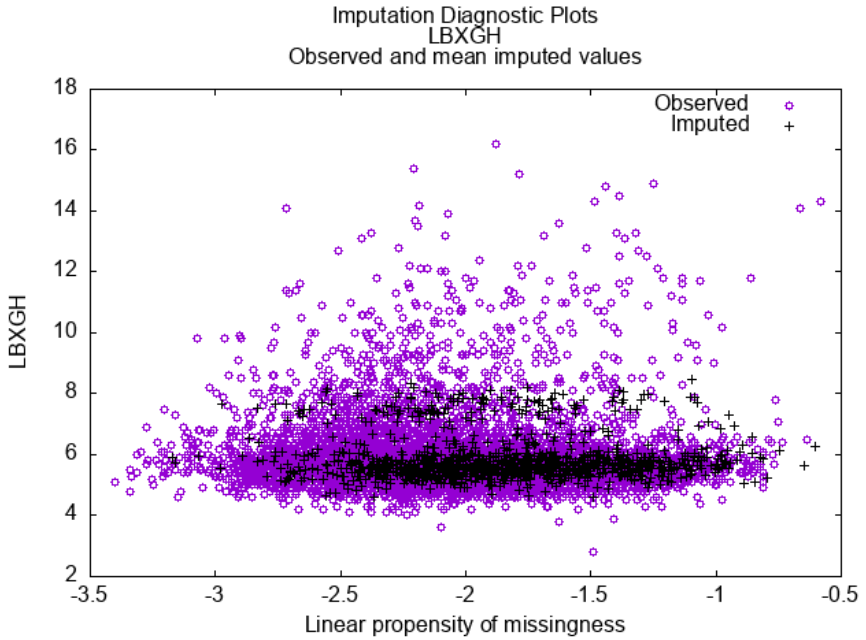
Variable medyes

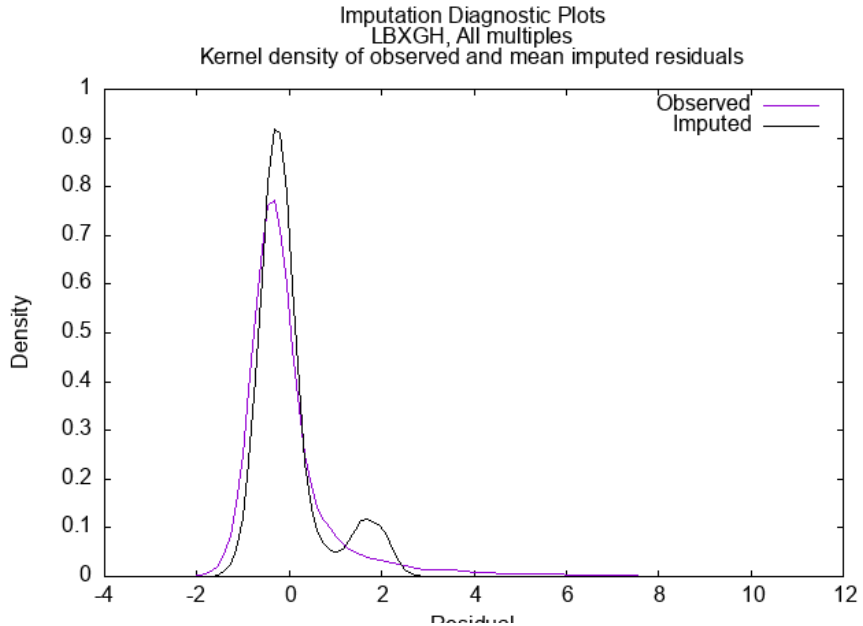
	Observed		Imputed		Combined	
Code	Freq	Per	Freq	Per	Freq	Per
0	223	14.62	0	0.00	223	2.42
1	1302	85.38	1072	13.91	2374	25.71
2	0	0.00	6635	86.09	6635	71.87
Total	1525	100.00	7707	100.00	9232	100.00

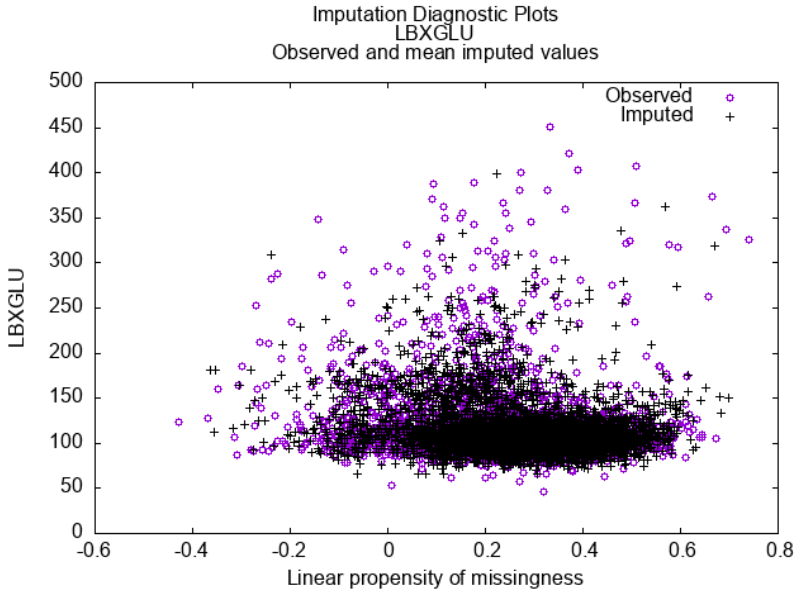
A new category is created for the “not applicable” group

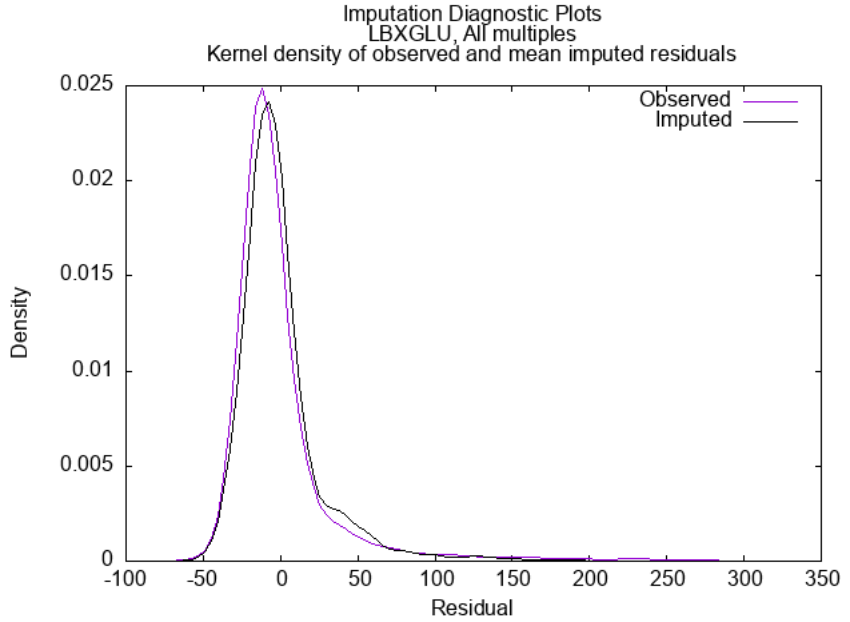
All imputed to be MEDYES=1

Frequency Percent Row Pct Col Pct	Table of diab_sr by medyes			
	medyes			
diab_sr	.	0	1	Total
.	3 0.03 100.00 0.04	0 0.00 0.00 0.00	0 0.00 0.00 0.00	3 0.03
1	6633 71.85 100.00 86.06	0 0.00 0.00 0.00	0 0.00 0.00 0.00	6633 71.85
2	862 9.34 94.31 11.18	0 0.00 0.00 0.00	52 0.56 5.69 3.99	914 9.90
3	206 2.23 78.63 2.67	0 0.00 0.00 0.00	56 0.61 21.37 4.30	262 2.84
4	3 0.03 0.21 0.04	223 2.42 15.70 100.00	1194 12.93 84.08 91.71	1420 15.38
Total	7707 83.48	223 2.42	1302 14.10	9232 100.00


Imputation Diagnostic Plots
LBXGH
Observed and mean imputed values
Observed
Imputed
LBXGH
Linear propensity of missingness


Imputation Diagnostic Plots
LBXGH, All multiples
Kernel density of observed and mean imputed residuals
Observed
Imputed
Density


Imputation Diagnostic Plots
LBXGLU
Observed and mean imputed values
Observed
Imputed
LBXGLU
Linear propensity of missingness


Imputation Diagnostic Plots
LBXGLU, All multiples
Kernel density of observed and mean imputed residuals
Observed
Imputed
Density
Residual

```
Regression type:          Logistic
Dependent variable:       undiagnosed
Predictors:               age
                          female
                          educ
                          marstatus
                          poverty
Cat. var. ref. codes:     educ  4
                          marstatus  3
                          poverty  3
                          undiagnosed  0
Stratum variable:         SDMVSTRA  Masked variance pseudo-stratum
Cluster variable:         SDMVPSU  Masked variance pseudo-PSU
Weight variable:          WTINTPRP  Full sample interview weight
```

Variable	Estimate	Std Error	T Test	Prob > |T|
Intercept	-2.6257750	0.2628579	-9.98933	0.00000
age	0.0058773	0.0032340	1.81738	0.09632
female	-0.3066201	0.1944321	-1.57700	0.14294
educ.1	0.3260654	0.2445508	1.33332	0.20923
educ.2	0.1010917	0.2363320	0.42775	0.67704
educ.3	0.0486406	0.2073642	0.23457	0.81883
marstatus.1	-0.0548861	0.2012290	-0.27275	0.79007
marstatus.2	-0.0281356	0.2515655	-0.11184	0.91295
poverty.1	0.0590971	0.1959279	0.30163	0.76853
poverty.2	0.0962625	0.1599039	0.60200	0.55931

Variable	Odds Ratio	95% Confidence Interval Lower	Upper
Intercept			
age	1.0058946	0.9987652	1.0130750
female	0.7359302	0.4798585	1.1286519
educ.1	1.3855059	0.8091166	2.3724970
educ.2	1.1063780	0.6578955	1.8605878
educ.3	1.0498429	0.6653467	1.6565350
marstatus.1	0.9465929	0.6080613	1.4735985
marstatus.2	0.9722566	0.5590917	1.6907472
poverty.1	1.0608783	0.6894667	1.6323671
poverty.2	1.1010481	0.7745768	1.5651216

Variable	Design Effect	SRS Estimate	% Diff SRS v Est
Intercept	1.46912	-2.4315062	-7.39853
age	1.26718	0.0014450	-75.41317
female	2.88350	-0.3006148	-1.95853
educ.1	1.97808	0.3067634	-5.91968
educ.2	2.32906	0.2162526	113.91731
educ.3	2.08176	0.0300750	-38.16892
marstatus.1	1.87319	0.0468392	-185.33891
marstatus.2	2.45196	0.0705756	-350.84125
poverty.1	2.18845	0.0255388	-56.78510
poverty.2	1.72655	0.0623030	-35.27806

```
/* This SAS program uses previously imputed data to perform
analysis using the PROC MIANALYZE */
options ls=80 ps=72 nodate;
libname workshop "e:\workshop";
/* Read data */
data imp_anal;
set workshop.workshopimputed;
_imputation_=_mult_; /* MIANALYZE uses _imputation_ but IVEWARE uses _MULT_ */
/* SAS uses 1 -1 coding for variables in the class statement.
Dummy variables may be better */
educ1=0; if educ=1 then educ1=1;
educ2=0; if educ=2 then educ2=1;
educ3=0; if educ=3 then educ3=1;
marstatus1=0; if marstatus=1 then marstatus1=1;
marstatus2=0; if marstatus=2 then marstatus2=1;
poverty1=0; if poverty=1 then poverty1=1;
poverty2=0; if poverty=2 then poverty2=1;
/* Need to sort by imputation */
proc sort;
by _imputation_;
/* Fit Complex survey logistic regress model on each data set and store the output */
proc surveylogistic data=imp_anal;
cluster sdmvpsu;
strata sdmvstra;
weight wtintprp;
model undiagnosed(desc)=age female educ1 educ2 educ3 marstatus1 marstatus2 poverty1 poverty2/link=logit;
by _imputation_;
ods output parameterestimates=parms;
run;
/* Sort the data by imputation */
proc sort data=parms;
by _imputation_;
run;
/* Combine the results using PROC MIANALYZE */
proc mianalyze parms=parms;
modeleffects intercept age female educ1 educ2 educ3 marstatus1 marstatus2 poverty1 poverty2;
run;
```

Same Analysis Performed using PROC SURVEYLOGISTIC and MIANALYZE

Parameter Estimates										
Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
intercept	-2.625768	0.264916	-3.15535	-2.09618	61.84	-2.915270	-2.455040	0	-9.91	<.0001
age	0.005877	0.003244	-0.00052	0.01227	204.01	0.003217	0.007763	0	1.81	0.0715
female	-0.306620	0.194171	-0.70872	0.09548	22.57	-0.627969	-0.137741	0	-1.58	0.1282
educ1	0.326095	0.244676	-0.16505	0.81724	51.284	0.144497	0.582169	0	1.33	0.1885
educ2	0.101091	0.236400	-0.37593	0.57811	42.16	-0.100864	0.420524	0	0.43	0.6711
educ3	0.048645	0.206863	-0.36430	0.46159	66.624	-0.128370	0.231163	0	0.24	0.8148
marstatus1	-0.054897	0.202971	-0.46247	0.35267	50.552	-0.262926	0.122420	0	-0.27	0.7879
marstatus2	-0.028147	0.251271	-0.53086	0.47457	59.433	-0.257669	0.236295	0	-0.11	0.9112
poverty1	0.059111	0.195228	-0.34150	0.45972	26.952	-0.160678	0.233157	0	0.30	0.7644
poverty2	0.096267	0.160766	-0.22248	0.41501	105.76	-0.043460	0.223800	0	0.60	0.5506

The MIANALYZE Procedure

Variance Information

Parameter	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
intercept	0.024339	0.043407	0.070180	61.84	0.616797	0.400571	0.961486
age	0.000002009	0.000008313	0.000010524	204.01	0.265881	0.217668	0.978697
female	0.021643	0.013895	0.037702	22.57	1.713463	0.660292	0.938061
educ1	0.022799	0.034787	0.059867	51.284	0.720930	0.440328	0.957824
educ2	0.023473	0.030064	0.055885	42.16	0.858833	0.485853	0.953666
educ3	0.014298	0.027064	0.042792	66.624	0.581134	0.385710	0.962861
marstatus1	0.015803	0.023815	0.041197	50.552	0.729926	0.443529	0.957531
marstatus2	0.022336	0.038568	0.063137	59.433	0.637039	0.408709	0.960734
poverty1	0.020022	0.016089	0.038114	26.952	1.368895	0.606050	0.942858
poverty2	0.006854	0.018306	0.025846	105.76	0.411850	0.304734	0.970428

```
<R name="Routput">
# The above command is the beginning of R
# set working dir
setwd ("E:/WORKSHOP")

# load packages necessary for this analysis
library(haven)
library(mitools)
library(graphics)
library(survey)

# Read in the SAS data set using the haven package (use 2 slashes not 1 backslash)
one=read_sas("e:\\WORKSHOP\\workshop.sas7bdat")

# Obtain summary stats and save the data in a file
summary(one)
save(one, file="one.rda")
# Start of Impute Commands
<impute name="Rimpout">
 datain one;          /* Input data */
dataout oneimp all; /* Imputed data; "all" will stack the data */
default categorical; /* Unless indicated everything is categorical */
continuous lbxglu lbxgh age wtintprp; /* Continuous variables */
transfer seqn sdmvpsu a1c_sr testfora1c; /* Variables to be transferred and not to be used as predictors */
restrict medyes (diab_sr>1); /* Variable Medyes is only to be imputed for dian_sr >1 */
bounds lbxgh (>=2.8, <=16.2) lbxglu(>=47, <=451); /* Bounds for the imputed values to be in the observed range */
iterations 10; /* Number of Iterations */
multiples 10;  /* Number of Imputations */
diagnose lbxglu lbxgh; /* Create diagnostic Plots for these variables */
seed 23456; /* specify seed for replicability */
run; /* Run the commands. End Impute portion */
</impute>

# use imputed data in r to post process
# must load rda and extract needed parts of data
# "imp" object stores the imputed data
load("oneimp.rda")
ls()
names(imp)
```

Toggle R (will create Routput.R, Routput.out)

Read in SAS data

Toggle Impute

Impute commands (same as before)

End Impute
(will create Rimpout.set, Rimpout,log Rimpout.lst)

Process imputed data

```
summary(imp)
# Make a copy for analysis
imp1 <- imp
summary(imp1)
# create undiagnosed, note case of variables
imp1$undiagnosed  <- ifelse(imp1$diab_sr == 1 & (imp1$LBXGLU > 126 | imp1$LBXGH > 6.5), 1,0)
# change the reference level in the categorical predictors
imp1$educ=as.factor(imp1$educ)
imp1$educ=relevel(imp1$educ,ref=4)
imp1$marstatus=as.factor(imp1$marstatus)
imp1$marstatus=relevel(imp1$marstatus,ref=3)
imp1$poverty=as.factor(imp1$poverty)
imp1$poverty=relevel(imp1$poverty,ref=3)
names(imp1)
summary(imp1)

# set survey design
nhc <- svydesign(id=~SDMVPSU, strata=~SDMVSTRA, weights=~WTINTPRP, data=imp1,
nest=TRUE)

# split the stacked data by imputation indicatior (note case of MULT_)
milist <- imputationList(split(imp1,imp1$MULT_))

# To deal with any strata with one psu. Not needed here
# options(survey.lonely.psu="certainty")

# Fit the model on each data set
miresults <- with(milist,svyglm(undiagnosed~ age+ female+ educ+
marstatus+poverty,family="quasibinomial",design=nhc))

# Combine the results
results <- MIcombine(miresults)

# summarize results
summary(results)
# Close out R
</R>
```

Make a copy and define dependent variable and set the reference levels for creating dummy variables

Define Survey Design

Split the stacked data into individual imputed data sets.

Analyze each imputed data

Combine the results and summarize

Close R

```
Multiple imputation results:
      with(milist, svyglm(undiagnosed ~ age + female + educ + marstatus +
    poverty, family = "quasibinomial", design = nhc))
      MIcombine.default(miresults)
                 results         se        (lower      upper) missInfo
(Intercept) -2.733858094 0.14913946 -3.026166059 -2.44155013      0 %
age          0.006380158 0.00217258  0.002121979  0.01063834      0 %
female      -0.165427930 0.07710502 -0.316551001 -0.01430486      0 %
educ1        0.323340618 0.10937868  0.108962344  0.53771889      0 %
educ2        0.229915378 0.11300219  0.008435156  0.45139560      0 %
educ3        0.121122942 0.09698434 -0.068962867  0.31120875      0 %
marstatus1  -0.036956663 0.13022894 -0.292200696  0.21828737      0 %
marstatus2  -0.029878669 0.14983543 -0.323550706  0.26379337      0 %
poverty1     0.016430521 0.12556505 -0.229672446  0.26253349      0 %
poverty2    -0.009720488 0.11775863 -0.240523157  0.22108218      0 %
```

IVEware Regress and SAS Surveylogistic/Mianalyze point estimates are the same but estimates from R differ.

IVEware uses Jackknife whereas SAS and R use linearization

```
/* Read a delimited text file with the variable names in the first row */
<getdata name="one">
table "workshop.txt";
run;
</getdata>
/* Ready to perform Imputation */
<impute name="Aimpout">
datain one;      /* Input data */
dataout oneimp1;    /* Imputed data; */
default categorical; /* Unless indicated everything is categorical */
continuous lbxglu lbxgh age wtintprp; /* Continuous variables */
transfer seqn sdmvpsu a1c_sr testfora1c; /* Variables to be transferred and not to be used as predictors */
restrict medyes (diab_sr>1); /* Variable Medyes is only to be imputed for dian_sr >1 */
bounds lbxgh (>=2.8, <=16.2) lbxglu(>=47, <=451); /* Bounds for the imputed values to be in the observed range */
iterations 10; /* Number of Iterations */
multiples 10;  /* Number of Imputations */
diagnose lbxglu lbxgh; /* Create diagnostic Plots for these variables */
seed 23456; /* specify seed for replicability */
run; /* Run the commands. End Impute portion */
</impute>
/* Extract the 10 data sets */
<putdata name="Aimpout",mult=2,dataout="oneimp2"/>
<putdata name="Aimpout",mult=3,dataout="oneimp3"/>
<putdata name="Aimpout",mult=4,dataout="oneimp4"/>
<putdata name="Aimpout",mult=5,dataout="oneimp5"/>
<putdata name="Aimpout",mult=6,dataout="oneimp6"/>
<putdata name="Aimpout",mult=7,dataout="oneimp7"/>
<putdata name="Aimpout",mult=8,dataout="oneimp8"/>
<putdata name="Aimpout",mult=9,dataout="oneimp9"/>
<putdata name="Aimpout",mult=10,dataout="oneimp10"/>
```

Standalone version is used impute and fit a multinomial logit model

Read in the data set from a text file

Imputation (same as before)

Split the imputed data into separate data sets

```
/* Multinomial Logit Model with self report diabetes as the dependent variable */
<regress name="Aregout">
datain oneimp1 oneimp2 oneimp3 oneimp4 oneimp5 oneimp6 oneimp7 oneimp8 oneimp9 oneimp10;
categorical diab_sr educ marstatus poverty;
cluster sdmvpsu;
stratum sdmvstra;
weight wtintprp;
dependent diab_sr;
predictor age female educ marstatus poverty;
link logistic;
run;
</regress>
</srcware>
```

Fit a multinomial logit model

```
<sas name="bboutput">
optione ls=80 ps=72 nodate;
libname workshop "e:\workshop";
data one;
set workshop.workshop;
<bbdesign name="synthetic">
datain one;
dataout onepop;
stratum sdmvstra;
cluster sdmvpsu;
weight wtintprp;
csamples 25;
wsamples 5;
seed 23456;
run;
</bbdesign>
<impute name="synimp">
datain onepop;        /* Input data */
dataout workshop.onepopimp all; /* Imputed data; "all" will stack the data */
default categorical; /* Unless indicated everything is categorical */
continuous lbxglu lbxgh age wtintprp; /* Continuous variables */
transfer seqn sdmvpsu a1c_sr testfora1c _impl_ _obs_; /* Variables to be transferred and not to be used as predictors */
restrict medyes (diab_sr>1); /* Variable Medyes is only to be imputed for dian_sr >1 */
bounds lbxgh (>=2.8, <=16.2) lbxglu(>=47, <=451); /* Bounds for the imputed values to be in the observed range */
iterations 5; /* Number of Iterations */
multiples 5;  /* Number of Imputations */
by _impl_; /* Impute each synthetic population
seed 23456; /* specify seed for replicability */
run; /* Run the commands. End Impute portion */
</impute>
data synthpops;
set workshop.onepopimp;
undiagnosed=0;
if diab_sr=1 and (lbxglu> 126 or lbxgh > 6.5) then undiagnosed=1;
/* SAS uses 1 -1 coding for variables in the class statement.
Dummy variables may be better */
educ1=0; if educ=1 then educ1=1;
educ2=0; if educ=2 then educ2=1;
educ3=0; if educ=3 then educ3=1;
```

Generate synthetic populations:

- 25 Bayesian Bootstrap of samples of clusters
- 5 within cluster Finite Population Bayesian Bootstrap of nonsampled units with *f=0.1*
- A total 125 synthetic populations (stacked up)

Multiply impute each synthetic population. L=5

Process all 625 multiply imputed synthetic populations

```
marstatus1=0; if marstatus=1 then marstatus1=1;
marstatus2=0; if marstatus=2 then marstatus2=1;
poverty1=0; if poverty=1 then poverty1=1;
poverty2=0; if poverty=2 then poverty2=1;
indexL=_mult_;
indexS=floor((_impl_-1)/5)+1;
indexB=_impl_-(indexS-1)*5;
proc sort data=synthpops;
by indexS indexB indexL;
proc logisitic descending data=synthpops;
by indexS indexB indexL;
model undiagnosed=age female educ1 educ2 educ3 marstatus1 marstatus2 poverty1 poverty2;
ods output ParameterEstimates=outparms;
run;
proc sort data=outparms;
by variable indexS;
proc means data=outparms mean noprint;
var estimate;
by variable indexS;
output out=step1of2 mean=thetabar;
proc sort;
by variable;
proc means data=step1of2 mean var noprint;
var thetabar;
by variable;
output out=step2of2 mean=theta_MI var=TMI;
data final;
set step2of2;
SE_MI=sqrt((1+1/25)*TMI);
tvalue=quantile('T',0.975,24);
lowerlimit=theta_MI-tvalue*SE_MI;
upperlimit=theta_MI+tvalue*SE_MI;
proc print data=final;
var variable theta_MI SE_MI lowerlimit upperlimit;
run;
</sas>
```

Create the three indices S, B and L as described on slides 66-69, Workshop-Part 1

Fit the logistic regression model on each data set

Compute the mean, variance, standard error and 95% confidence interval

- Average over B and L (Step 1 of 2)
- Computation over S (Step 2 of 2)

This will take several hours/days to run