IVEware.2--Cover PageV2092311 9/23/2011

IVEware:

Imputation and Variance Estimation

Version 0.2 Users Guide (Supplement)

Survey Methodology Program
Survey Research Center, Institute for Social Research
University of Michigan

September 2011

(Partially funded by the NIH Grant 5-RC1-RR-028366-02)

IVEware, is the imputation and variance estimation software developed at the University of Michigan's Survey Research Center. This document provides a visual guide in the use of Version 0.2 of *IVEware*, which employees a user interface to develop the script and run the *IVEware* commands in SAS. This is a supplemental document to the older version of the Users Guide available at http://www.isr.umich.edu/src/smp/ive. The original *IVEware* Users Guide provides a fuller explanation of *IVEware* procedures and programming features. Version 0.2 *IVEware* allows users to input programming instructions through graphical interfaces—windows and dialogue boxes. In addition to the original modules IMPUTE, DESCRIBE, REGRESS and SASMOD, the new version includes three new modules SYNTHESIZE , COMBINE and DATAPREP. SYNTHESIZE creates synthetic data sets by synthesizing one or more variables in the data set for statistical disclosure limitation. Variables are synthesized using the sequential regression approach (the same methodology used in IMPUTE). COMBINE allows users to concatenate (stack) two or more datasets. DATAPREP provides the user with the ability to recode and transform variables within the *IVEware* environment.

Table of Contents

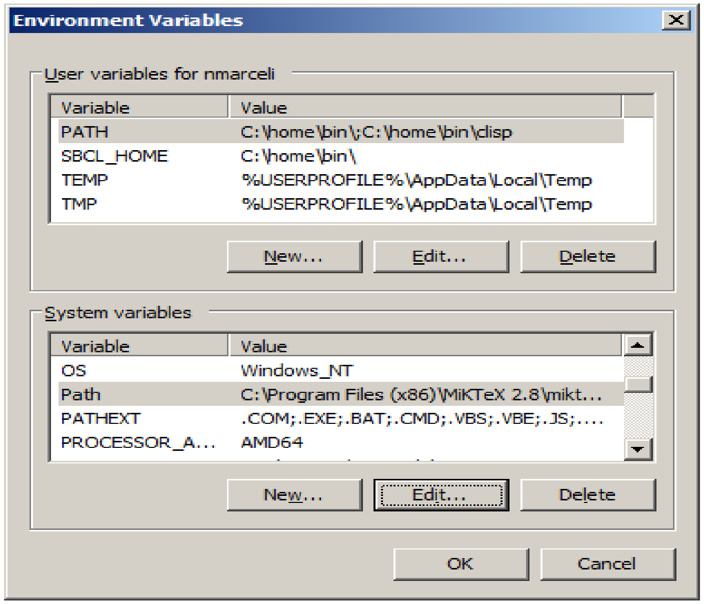
1. The Basics

Install Software

First make sure that the directory containing the SAS executable binary file (SAS.EXE) is on the System Path. A quick way to check this is simply to select the Start Menu, Run… prompt and then type in cmd, to open a Windows Command Prompt. Change into the C:\iveware directory; then type sas at the command prompt. If SAS executes then it is already set on the System Path. Alternatively, select the Start Menu, Settings, Control Panel, and then System. In the System Properties window click on Advanced Tab then on Environment Variables and then in the System variables section select Path and click on Edit. In the Edit System Variable Window check to see if there is some variable value with SASFoundation present. If there is, then SAS is probably already setup on the Path; if SAS is not on the path then you have to manually add it.

First find the directory where SAS.EXE is located. Usually, it will be "C:\Program Files\SAS\SAS 9.x". Sometimes, it could be a network location, if you are using the network version of SAS. You can use windows explorer to search for SAS.EXE file. Typically, you can find the correct value through: Select the Start Menu, Programs, SAS, SAS 9.2 (Additional Languages), and highlight SAS 9.2 (English). This will display the proper value for the SAS path in a popup window; in this case: C:\Program Files\SAS\SASFoundation\9.2\ (If There is another version of SAS being used the steps performed are similar.)

Once you have determine the location of the SAS.EXE file then append ; "C:\Program Files\SAS\SASFoundation\9.2" to the end of the Variable Value: Click Ok three times to effect changes and to exit out of the system window—The control Panel window can be closed.
For Example:

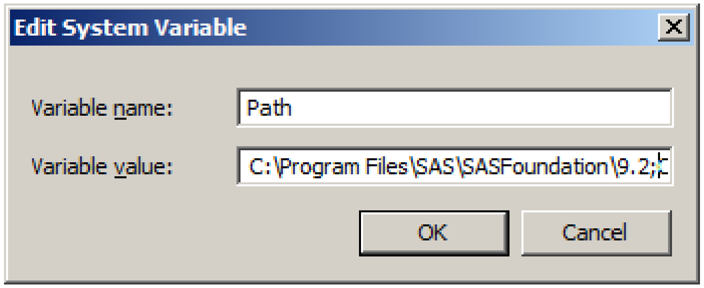

Environment Variables

User variables for nmarceli

Variable	Value
PATH	C:\home\bin\;C:\home\bin\clisp
SBCL_HOME	C:\home\bin\
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

System variables

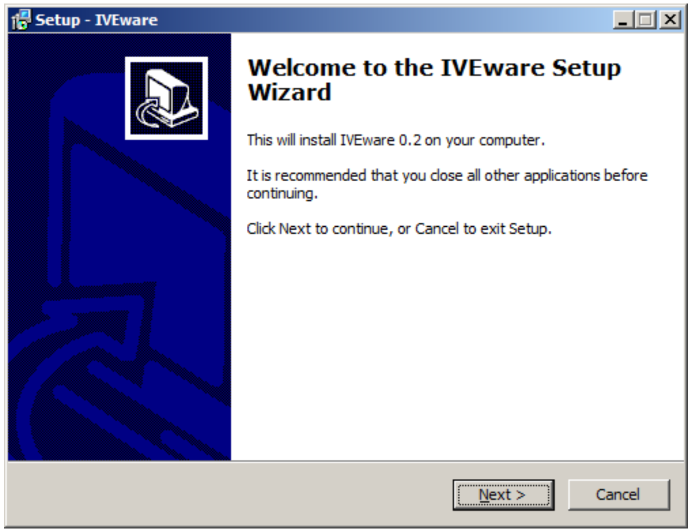
Variable	Value
OS	Windows_NT
Path	C:\Program Files (x86)\MiKTeX 2.8\mikt...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	AMD64


Edit System Variable

Variable name: Path

Variable value: C:\Program Files\SAS\SASFoundation\9.2;

Download and launch ivewaresetup.exe:



Welcome to the IVEware Setup Wizard

This will install IVEware 0.2 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Click on Next, accept the default values and click on Finish.

The default main directory is C:\iveware where the IVEware executable and supporting modules are located. For user analysis, the subdirectory C:\iveware\data\datasets\datain contains all input data sets—where datasets can be placed to be analyzed. The directory C:\iveware\data\datasets\dataout contains all data sets created by *IVEware*. The directory C:\iveware\data\output stores the results of submitted and processed scripts and contains the results of analyses such as log, output, and script files. The output directory performs the useful role of archiving all user analyses—including generated datasets. In this way the user can compare analyses across different program execution runs and navigate through multiple analyses facilitated by having output subdirectories generated and tagged with a timestamp.

You can now launch *IVEware* by clicking on the desktop icon (if installed) or running it from the start programs menu.

Uninstall Software

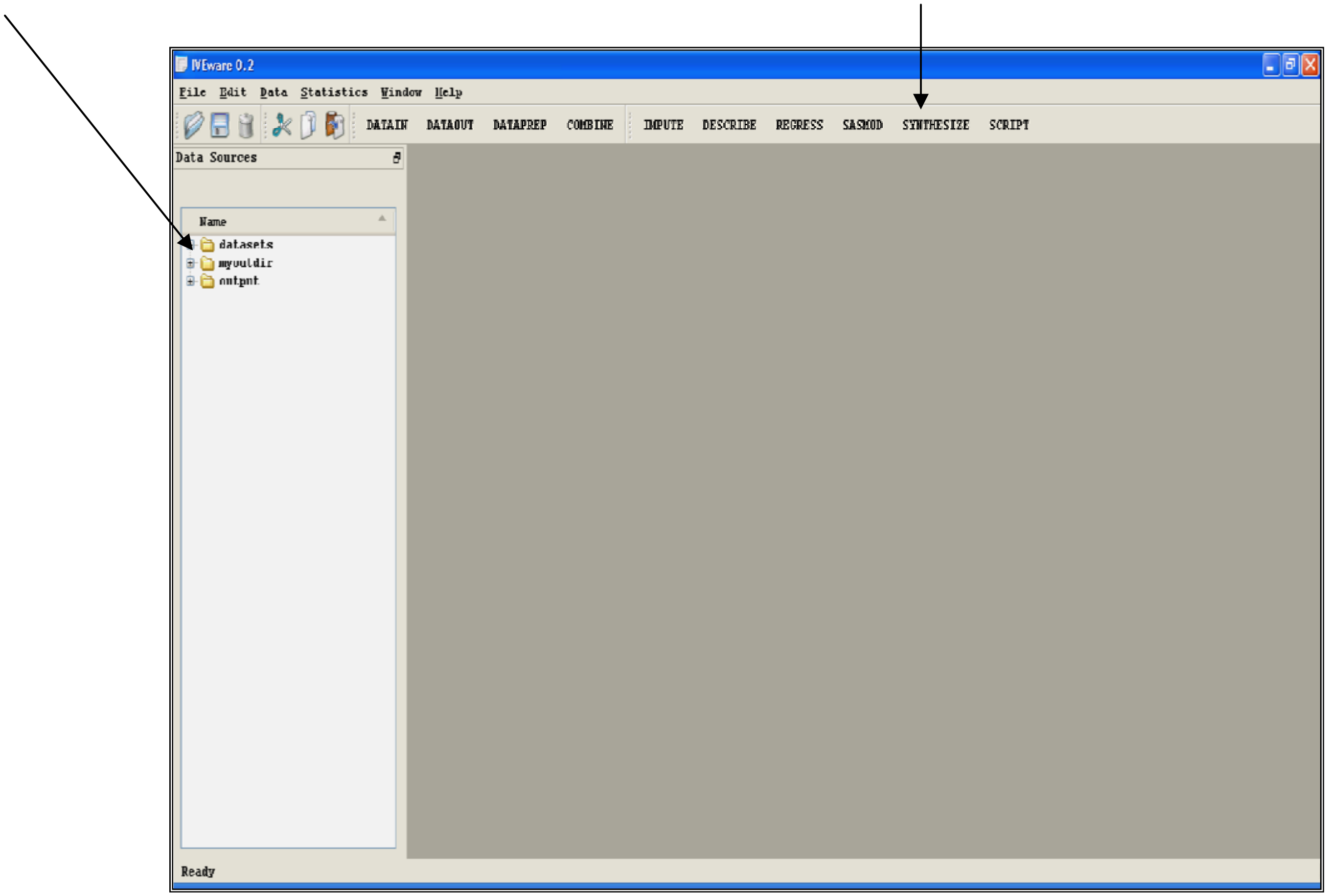
To uninstall the software for some reason, please follow these steps:

1. Select Start menu, Programs, IVEware, Uninstall   or
2. Using Windows Explorer delete the directory C:\iveware

Dataset Folders and Data Management and Procedure Buttons

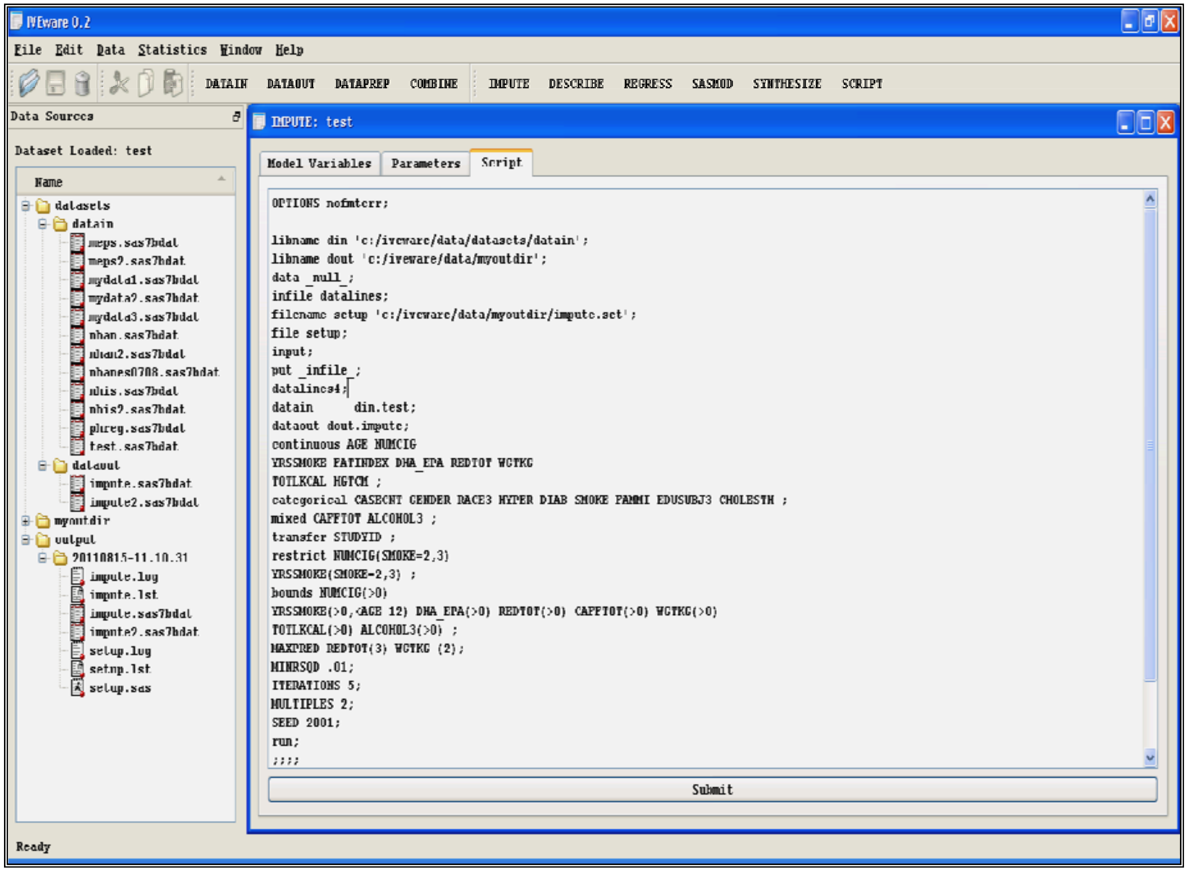
When you first start *IVEware*, the below window appears. This guide makes frequent reference to “Dataset Folders,” and data management and procedure “Buttons.”

Dataset Folders can be found in the white box at the left and the Buttons are arrayed in a row at the top of the window.


IVEware 0.2
File Edit Data Statistics Window Help
DATAIN DATAOUT DATAPREP COMBINE IMPUTE DESCRIBE REGRESS SASMOD SYNTHESIZE SCRIPT
Data Sources
Name
datasets
myoutdir
output
Ready

***IVEware* File Folder Structure**

*IVE*ware file folders are located in the directory C:\iveware\data. The directory contains three main folders—DATASETS, MYOUTDIR, and OUTPUT.



```
OPTIONS nofmterr;

libname din 'c:/iveware/data/datasets/datain';
libname dout 'c:/iveware/data/myoutdir';
data _null_;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/impute.set';
file setup;
input;
put _infile_;
datalines4;
datain      din.test;
dataout dout.impute;
continuous AGE NUMCIG
YRSSMOKE FATINDEX DHA_EPA REDTOT WGTKG
TOTLKCAL HGTCM ;
categorical CASECNT GENDER RACE3 HYPER DIAB SMOKE FAMMI EDUSUBJ3 CHOLESTH ;
mixed CAFFTOT ALCOHOL3 ;
transfer STUDYID ;
restrict NUMCIG(SMOKE=2,3)
YRSSMOKE(SMOKE=2,3) ;
bounds NUMCIG(>0)
YRSSMOKE(>0,<AGE 12) DHA_EPA(>0) REDTOT(>0) CAFFTOT(>0) WGTKG(>0)
TOTLKCAL(>0) ALCOHOL3(>0) ;
MAXPRED REDTOT(3) WGTKG (2);
MINRSQD .01;
ITERATIONS 5;
MULTIPLES 2;
SEED 2001;
run;
;;;;
```

The DATASETS folder has two sub-folders—DATAIN and DATAOUT. DATAIN contains the SAS datasets that you plan to use with an *IVEware* procedure. You can place datasets in DATAIN by coping or dragging them to C:\iveware\data\datasets\datain. In the example, there are 12 SAS datasets available for analysis.

The DATAOUT sub-folder is where datasets outputted by *IVEware* are automatically placed. In the example, the *IVE*ware impute procedure outputted two imputed datasets—impute.sas7bdat and impute2.sas7bdat. These datasets will be replaced if a subsequent imputation calls for output files with the same names.

The MYOUTDIR is a work folder where outputted files are temporarily placed during the submission of an *IVE*ware procedure. At the completion of the submission the files are removed from the folder.

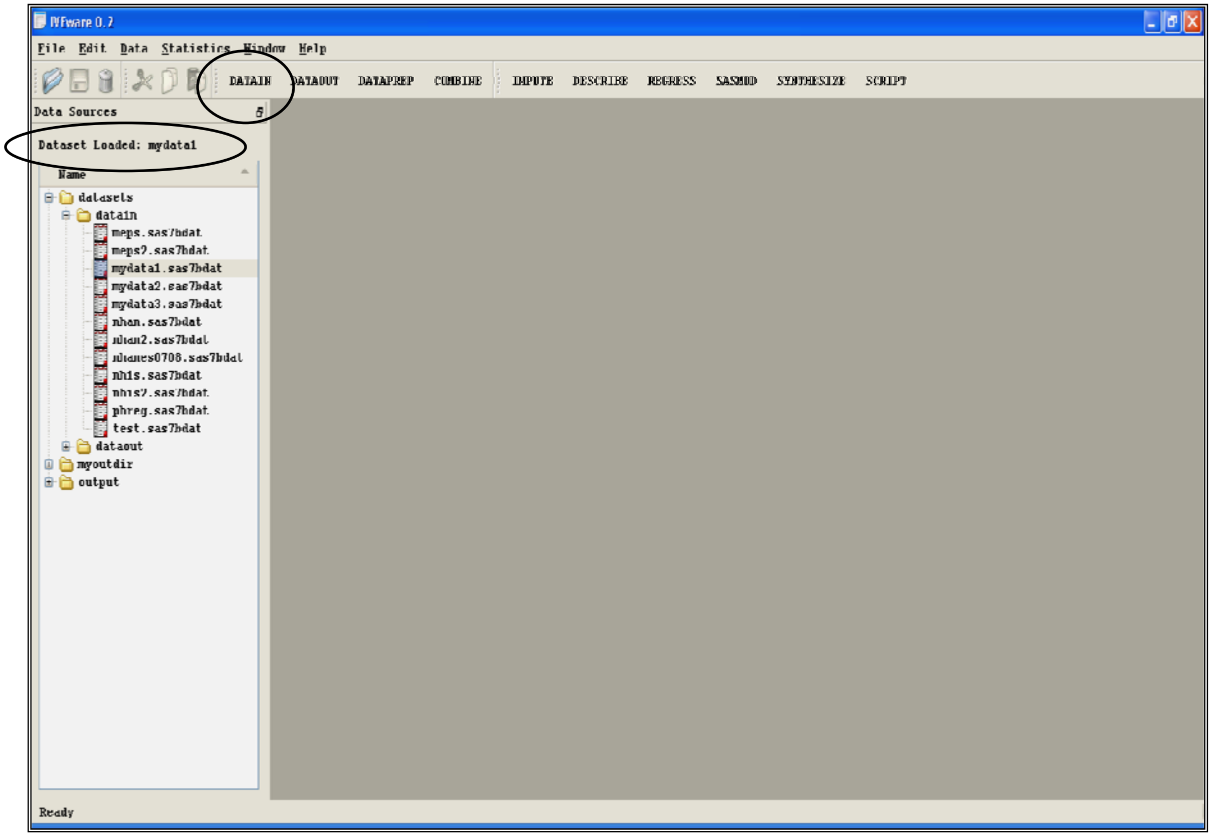
The OUTPUT folder contains separate sub-folders for each *IVEware* submission. The sub-folder name reflects the date and time of the submission and contains the results of the submission. The .log file contains a history of the submission including any errors in the submitted script. The .lst file provides the results of the submission (i.e., regression estimates, imputation results). If the submission calls for outputted datasets they are also included in the sub-folder. The outputted datasets will be identical to those outputted in the DATAOUT sub-folder (i.e., impute.sas7bdat and impute2.sas7bdat). Outputted datasets in the OUTPUT folder will be retained.

Loading Datasets

All datasets in the directory C:\iveware\data\datasets\datain will be available for your use. To see available datasets, click on the DATAIN folder and in the white box. If you want to add more data sets, you can drag and drop them into C:\iveware\data\datasets\datain using Windows Explorer.

Choose the dataset you want to work with by highlighting its name and clicking the DATAIN button.

The DATASET LOADED message with the name of the data set you selected will appear under the DATA SOURCES heading. In the example, the dataset MYDATA1 was loaded.

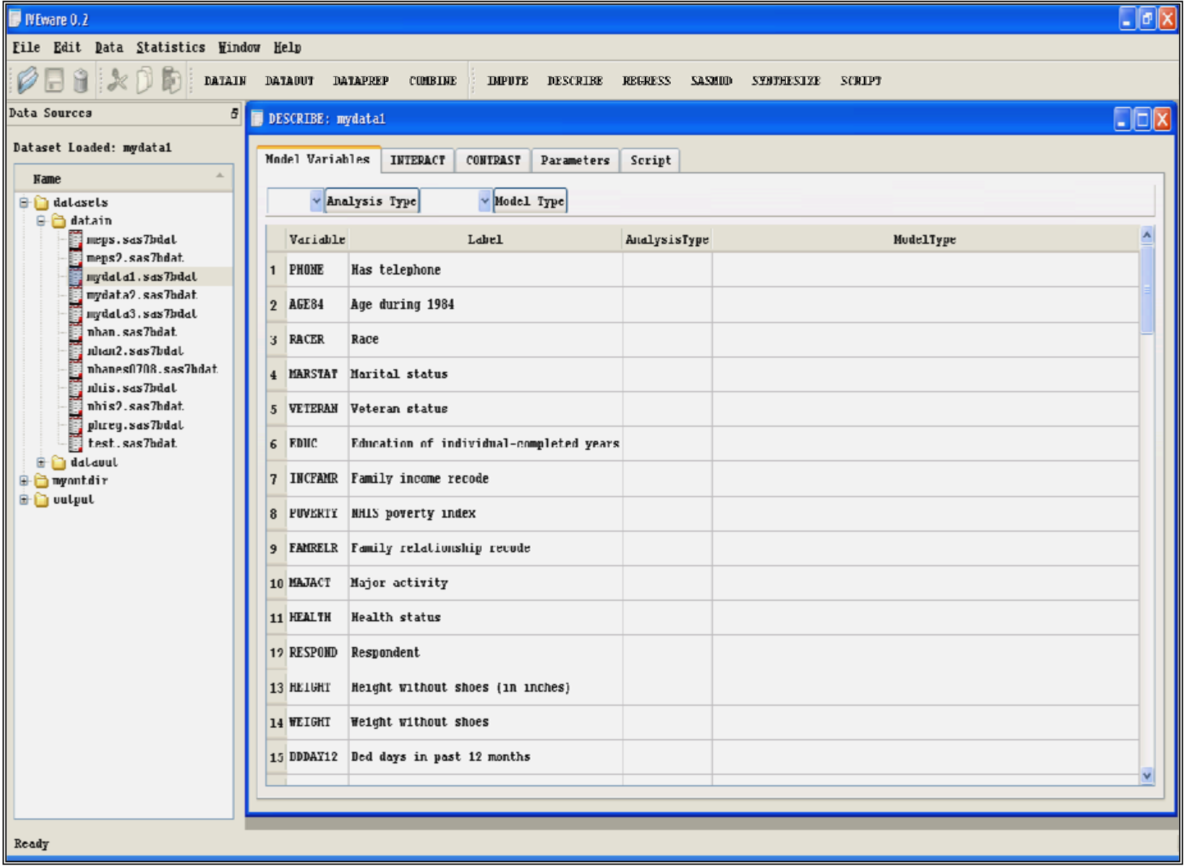


Dataset Loaded: mydata1

Selecting an *IVEware* Procedure

To select an *IVEware* procedure, click on a button labeled with the procedure name—IMPUTE, DESCRIBE, REGRESS, SASMOD or SYNTHESIZE.

After clicking on the procedure button, the Model Variables window will appear. In the example, the DESCRIBE procedure was selected.

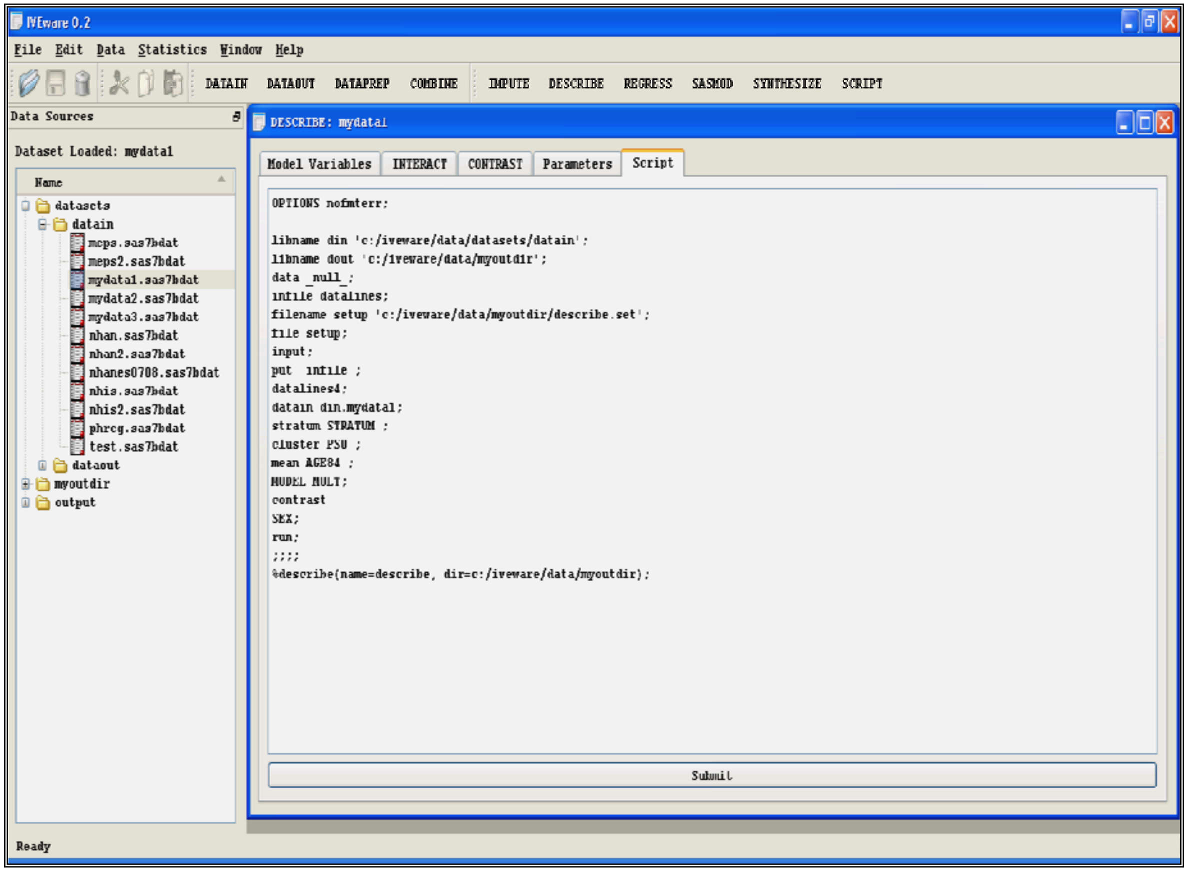


Dataset Loaded: mydata1

	Variable	Label	AnalysisType	ModelType
1	PHONE	Has telephone		
2	AGE84	Age during 1984		
3	RACER	Race		
4	MARSTAT	Marital status		
5	VETERAN	Veteran status		
6	EDUC	Education of individual-completed years		
7	INCFAMR	Family income recode		
8	POVERTY	NHIS poverty index		
9	FAMRELR	Family relationship recode		
10	MAJACT	Major activity		
11	HEALTH	Health status		
12	RESPOND	Respondent		
13	HEIGHT	Height without shoes (in inches)		
14	WEIGHT	Weight without shoes		
15	DDDAY12	Bed days in past 12 months		

Running Programs The Way You Use To

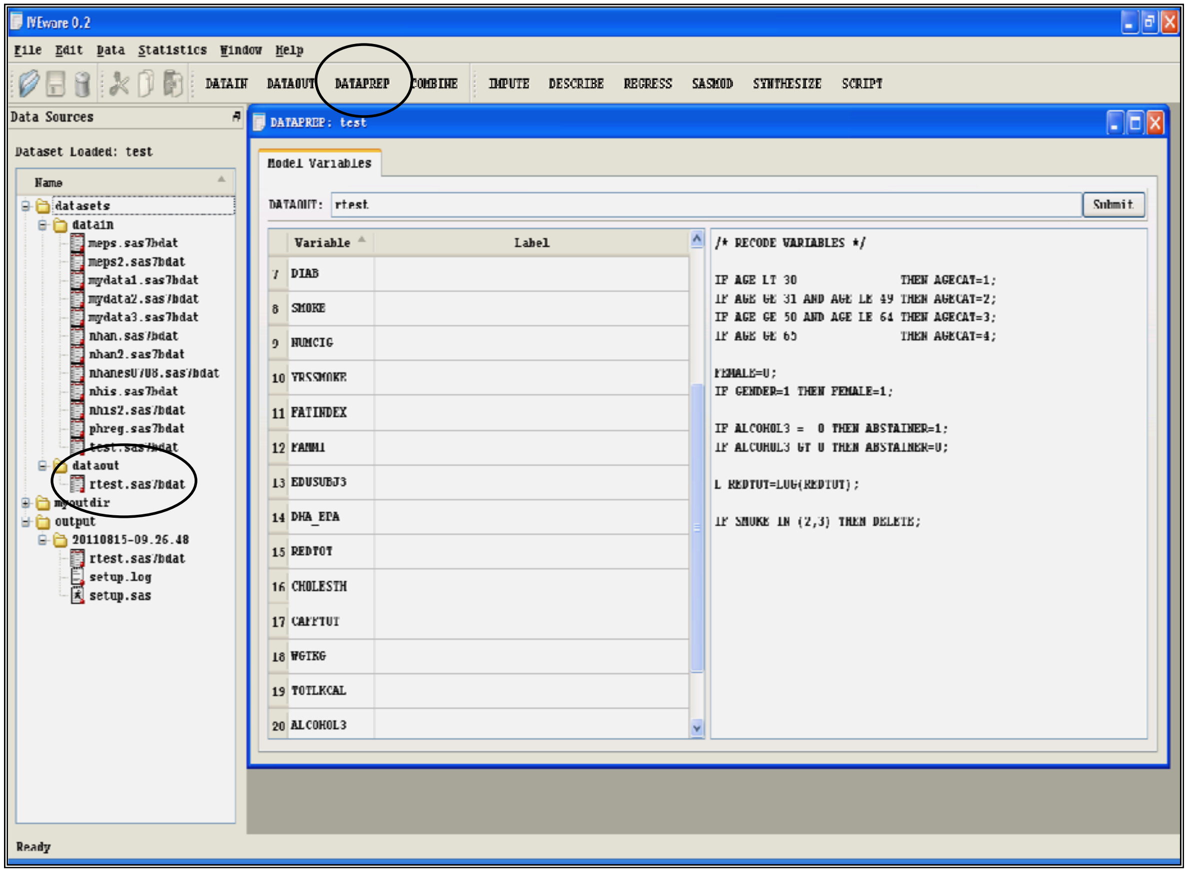
We recognize that some people may prefer to run the software the way they have used it before. Many may have setup files that they may want to rerun after installing the new version. To accommodate such situations, we created an integrated editor where you can open your setup files and submit the job directly. In order to do so, click the SCRIPT button. This will open a blank editor window. You load the previously run setup files or directly type in the *IVEware* commands, as you used to do in the SAS Program window. After you are done, you can submit the script by clicking the SUBMIT button at the bottom of the script window. The output is stored in the subdirectory “C:\iveware\data\output”.



```
OPTIONS nofmterr;

libname din 'c:/iveware/data/datasets/datain';
libname dout 'c:/iveware/data/myoutdir';
data _null_;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/describe.set';
file setup;
input;
put _infile_ ;
datalines4;
datain din.mydata1;
stratum STRATUM ;
cluster PSU ;
mean AGE84 ;
MODEL MULT;
contrast
SEX;
run;
;;;;
%describe(name=describe, dir=c:/iveware/data/myoutdir);
```

Variable Recodes

In the course of your analysis, it may be desirable to create recodes of variables such as transforming the variables in the imputation or regression analysis, collapsing and combining a variable to create new categories. You can make such transformation by clicking the DATAPREP button. A blank editor window will be created where you can type in any of the SAS commands used in the SAS Data Step. When you submit these commands, a new data set with the prefix R will be created in the DATAOUT directory. For example, suppose that TEST is loaded when you submitted the commands, a new data set RTEST will be created in the DATAOUT directory with the original variables and the recode variables.



DATAOUT: rtest

	Variable	Label
7	DIAB	
8	SMOKE	
9	NUMCIG	
10	YRSSMOKE	
11	FATINDEX	
12	FAMH1	
13	EDUSUBJ3	
14	DHA_EPA	
15	REDTOT	
16	CHOLESTH	
17	CAFFTOT	
18	WGTKG	
19	TOTLKCAL	
20	ALCOHOL3	

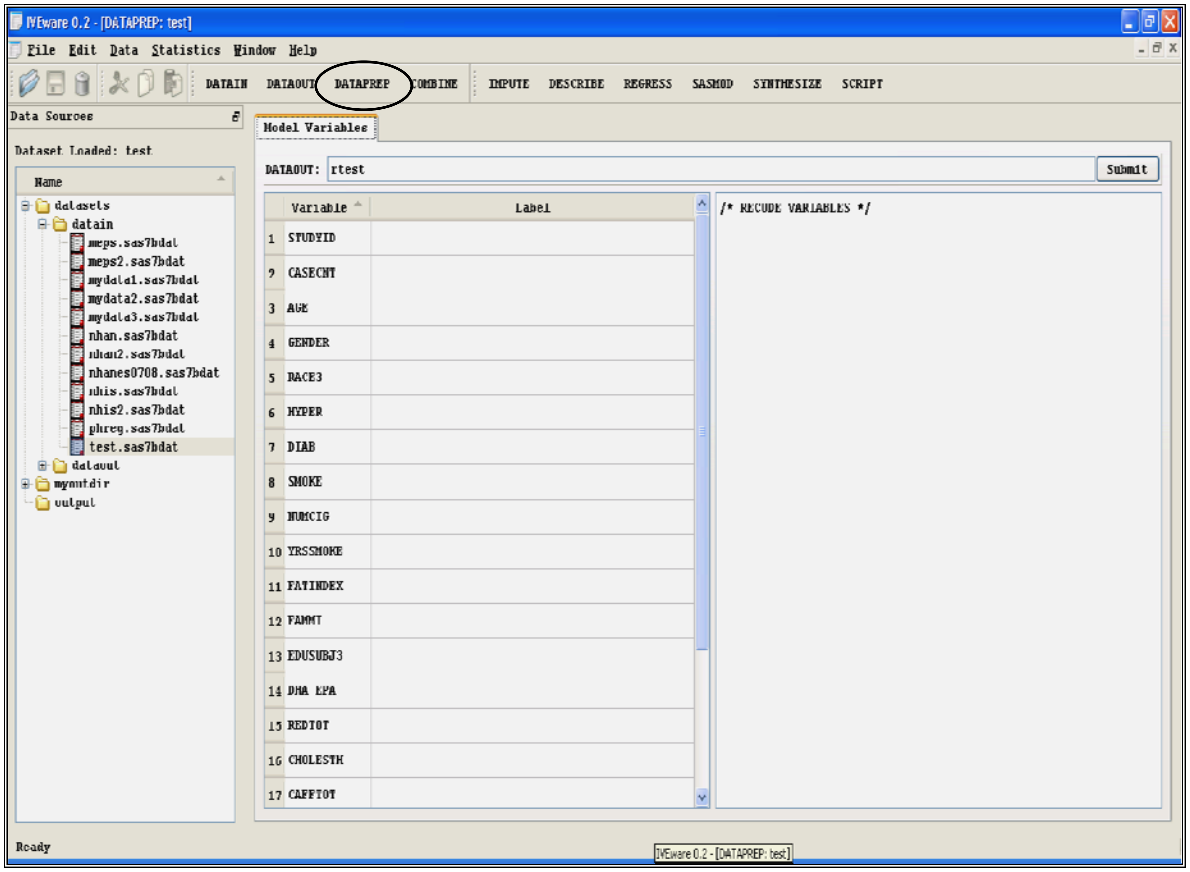
```
/* RECODE VARIABLES */

IF AGE LT 30                  THEN AGECAT=1;
IF AGE GE 31 AND AGE LE 49 THEN AGECAT=2;
IF AGE GE 50 AND AGE LE 64 THEN AGECAT=3;
IF AGE GE 65                  THEN AGECAT=4;

FEMALE=0;
IF GENDER=1 THEN FEMALE=1;

IF ALCOHOL3 =  0 THEN ABSTAINER=1;
IF ALCOHOL3 GT 0 THEN ABSTAINER=0;

L REDTOT=LOG(REDTOT);

IF SMOKE IN (2,3) THEN DELETE;
```

2. Data Prep

Selecting Data Prep

After loading a dataset, click on the DATAPREP button. This will open the Model Variables window, listing all variables in the loaded dataset.

In the example, TEST was previously loaded. The dataset must be loaded before selecting an *IVEware* procedure (see page 7).


IVEware 0.2 - [DATAPREP: test]
File Edit Data Statistics Window Help
DATAIN DATAOUT DATAPREP COMBINE IMPUTE DESCRIBE REGRESS SASMOD SYNTHESIZE SCRIPT
Data Sources
Model Variables
Submit
Ready
IVEware 0.2 - [DATAPREP: test]

Dataset Loaded: test

- datasets
  - datain
    - meps.sas7bdat
    - meps2.sas7bdat
    - mydata1.sas7bdat
    - mydata2.sas7bdat
    - mydata3.sas7bdat
    - nhan.sas7bdat
    - nhan2.sas7bdat
    - nhanes0708.sas7bdat
    - nhis.sas7bdat
    - nhis2.sas7bdat
    - phreg.sas7bdat
    - test.sas7bdat
  - dataout
- myoutdir
- output

DATAOUT: rtest

	Variable	Label
1	STUDYID	
2	CASECNT	
3	AGE	
4	GENDER	
5	RACE3	
6	HYPER	
7	DIAB	
8	SMOKE	
9	NUMCIG	
10	YRSSMOKE	
11	FATINDEX	
12	FAMMT	
13	EDUSUBJ3	
14	DHA EPA	
15	REDTOT	
16	CHOLESTH	
17	CAFFTOT	

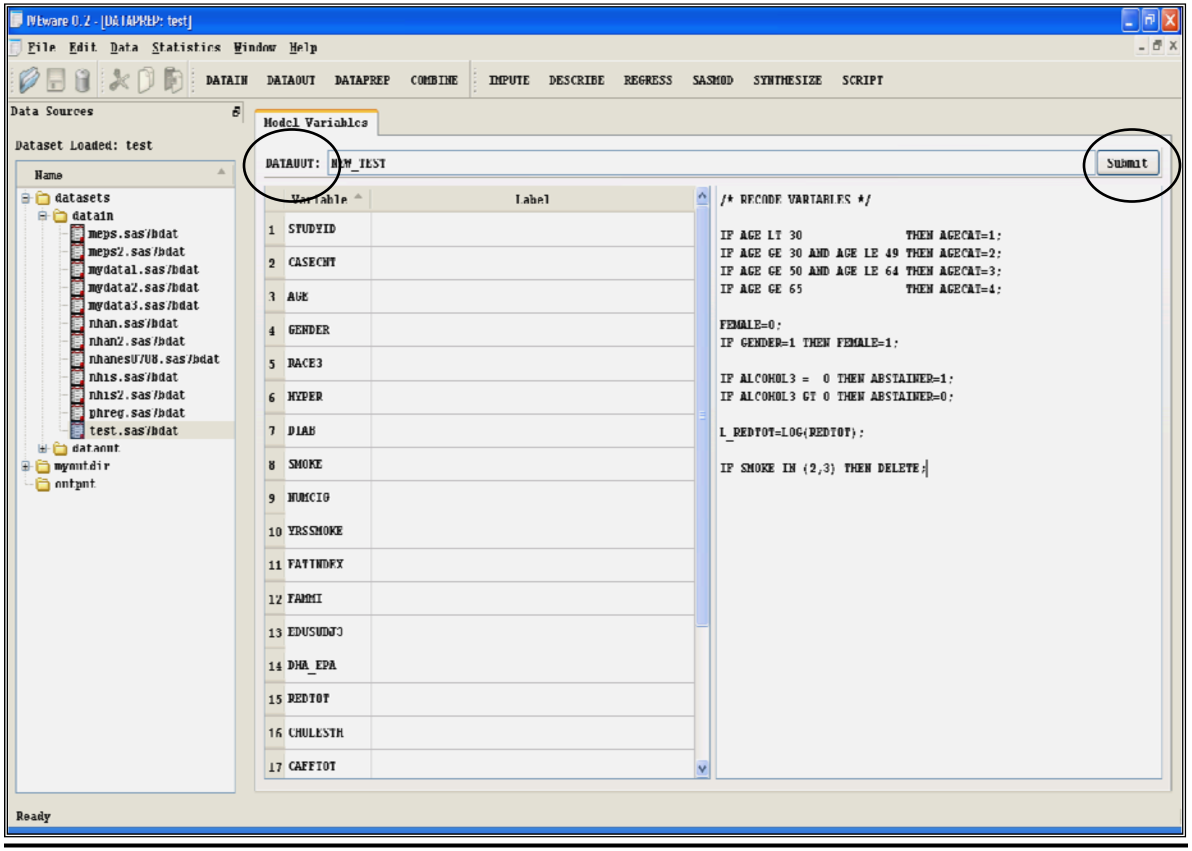
```
/* RECODE VARIABLES */
```

Data and File Management Script Window

The script window on the right is where SAS recode and data management instructions can be typed in directly. In the example, the continuous variable AGE is recoded into four categories, GENDER is recoded into the dummy variable FEMALE, a log transformation of REDTOT is undertaken, and ABSTAINER is constructed from the continuous alcohol consumption variable ALCOHOL3. TEST is subset so that only non-smokers are included in the new transformed dataset.

The user may provide a name for the newly transformed dataset in the DATAOUT box. In the example, the transformed dataset will be called NEW_TEST. The default is an R added to the front of the original dataset name (i.e.; RTEST).

Click on the SUBMIT button on the right side of the window to generate the transformed dataset.



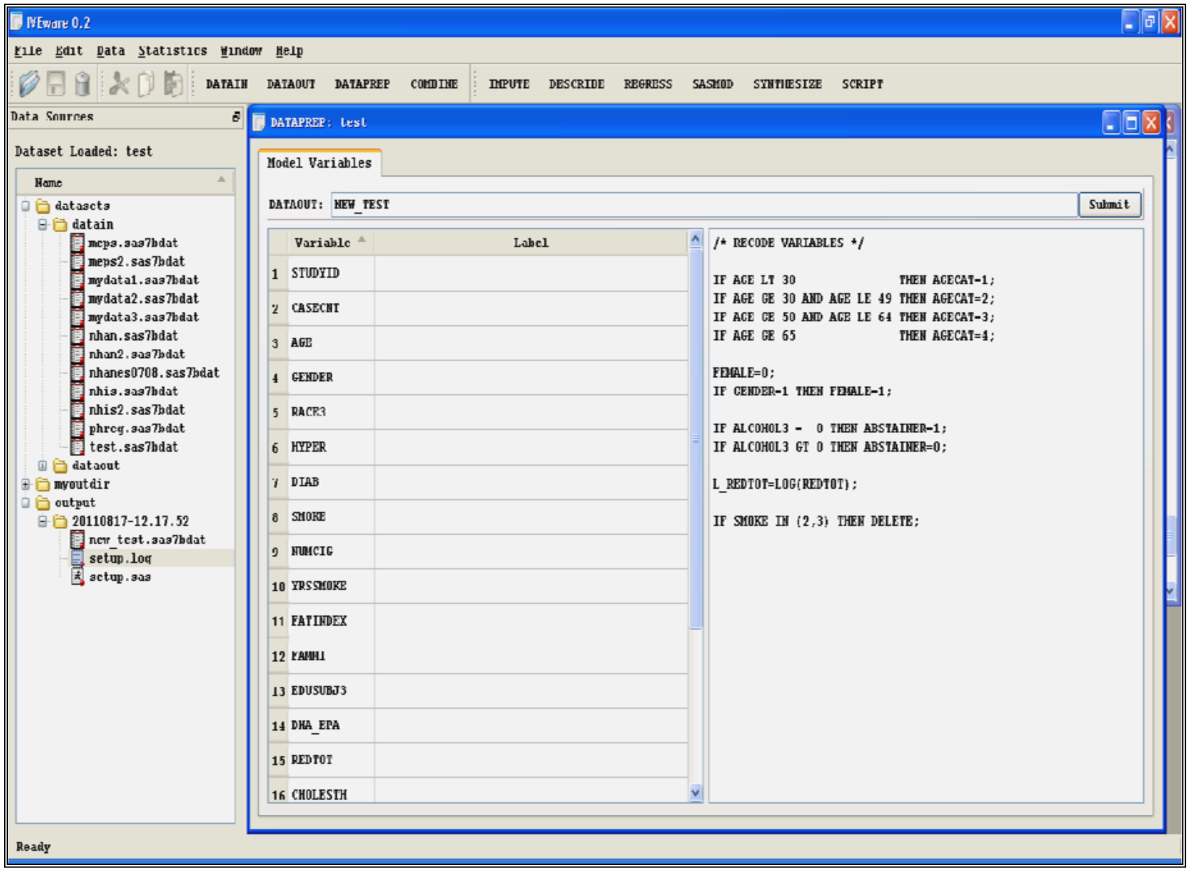
DATAOUT: NEW_TEST

	Variable	Label
1	STUDYID	
2	CASECNT	
3	AGE	
4	GENDER	
5	RACE3	
6	HYPER	
7	DIAB	
8	SMOKE	
9	NUMCIG	
10	YRSSMOKE	
11	FATINDEX	
12	FAMMI	
13	EDUSUDJ3	
14	DHA_EPA	
15	REDTOT	
16	CHOLESTR	
17	CAFFTOT	

```
/* RECODE VARIABLES */

IF AGE LT 30                THEN AGECAT=1;
IF AGE GE 30 AND AGE LE 49 THEN AGECAT=2;
IF AGE GE 50 AND AGE LE 64 THEN AGECAT=3;
IF AGE GE 65                THEN AGECAT=4;

FEMALE=0;
IF GENDER=1 THEN FEMALE=1;

IF ALCOHOL3 =  0 THEN ABSTAINER=1;
IF ALCOHOL3 GT 0 THEN ABSTAINER=0;

L_REDTOT=LOG(REDTOT);

IF SMOKE IN (2,3) THEN DELETE;
```

Output Files

After the script submission is completed, the OUTPUT folder will contain a sub-folder labeled with the date and time of the script submission. Here you will find the results of the Dataprep procedure. Errors in the submitted script will be reported in the setup.log file.


IVEware 0.2
File Edit Data Statistics Window Help
DATAIN DATAOUT DATAPREP COMBINE IMPUTE DESCRIBE REGRESS SASMOD SYNTHESIZE SCRIPT
Data Sources
DATAPREP: test
Model Variables
Submit
Ready

Dataset Loaded: test

```
Name
datasets
  datain
    meps.sas7bdat
    meps2.sas7bdat
    mydata1.sas7bdat
    mydata2.sas7bdat
    mydata3.sas7bdat
    nhan.sas7bdat
    nhan2.sas7bdat
    nhanes0708.sas7bdat
    nhis.sas7bdat
    nhis2.sas7bdat
    phrcg.sas7bdat
    test.sas7bdat
  dataout
myoutdir
output
  20110817-12.17.52
    new_test.sas7bdat
    setup.log
    setup.sas
```

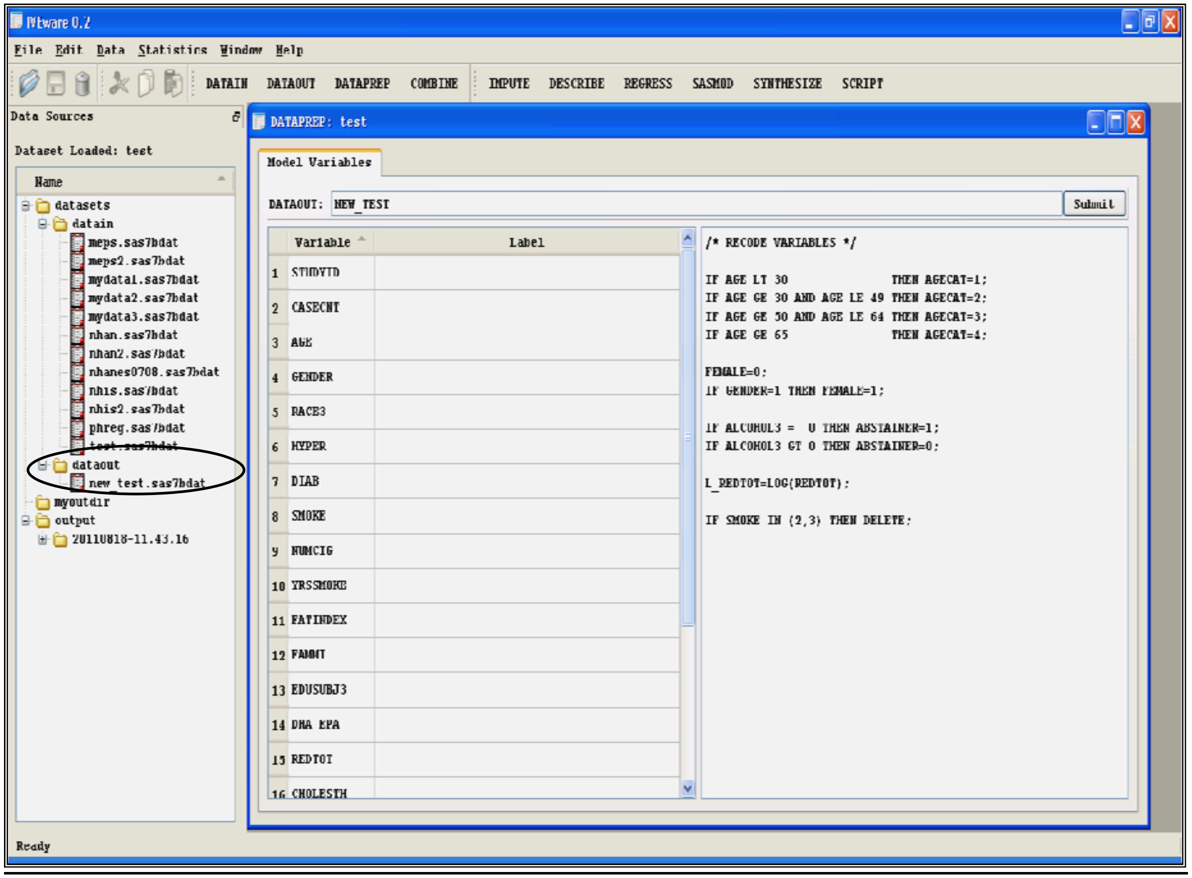
DATAOUT: NEW_TEST

	Variable	Label
1	STUDYID	
2	CASECNT	
3	AGE	
4	GENDER	
5	RACE3	
6	HYPER	
7	DIAB	
8	SMOKE	
9	NUMCIG	
10	YRSSMOKE	
11	FATINDEX	
12	FAMHI	
13	EDUSUBJ3	
14	DHA_EPA	
15	REDTOT	
16	CHOLESTH	

```
/* RECODE VARIABLES */

IF AGE LT 30                 THEN AGECAT=1;
IF AGE GE 30 AND AGE LE 49 THEN AGECAT=2;
IF AGE GE 50 AND AGE LE 64 THEN AGECAT=3;
IF AGE GE 65                 THEN AGECAT=4;

FEMALE=0;
IF GENDER=1 THEN FEMALE=1;

IF ALCOHOL3 =  0 THEN ABSTAINER=1;
IF ALCOHOL3 GT 0 THEN ABSTAINER=0;

L_REDTOT=LOG(REDTOT);

IF SMOKE IN (2,3) THEN DELETE;
```

Transformed Dataset

After a successful submission of the script the DATAOUT folder will display the new transformed dataset. NEW_TEST now appears in the DATAOUT folder.



DATAOUT: NEW_TEST

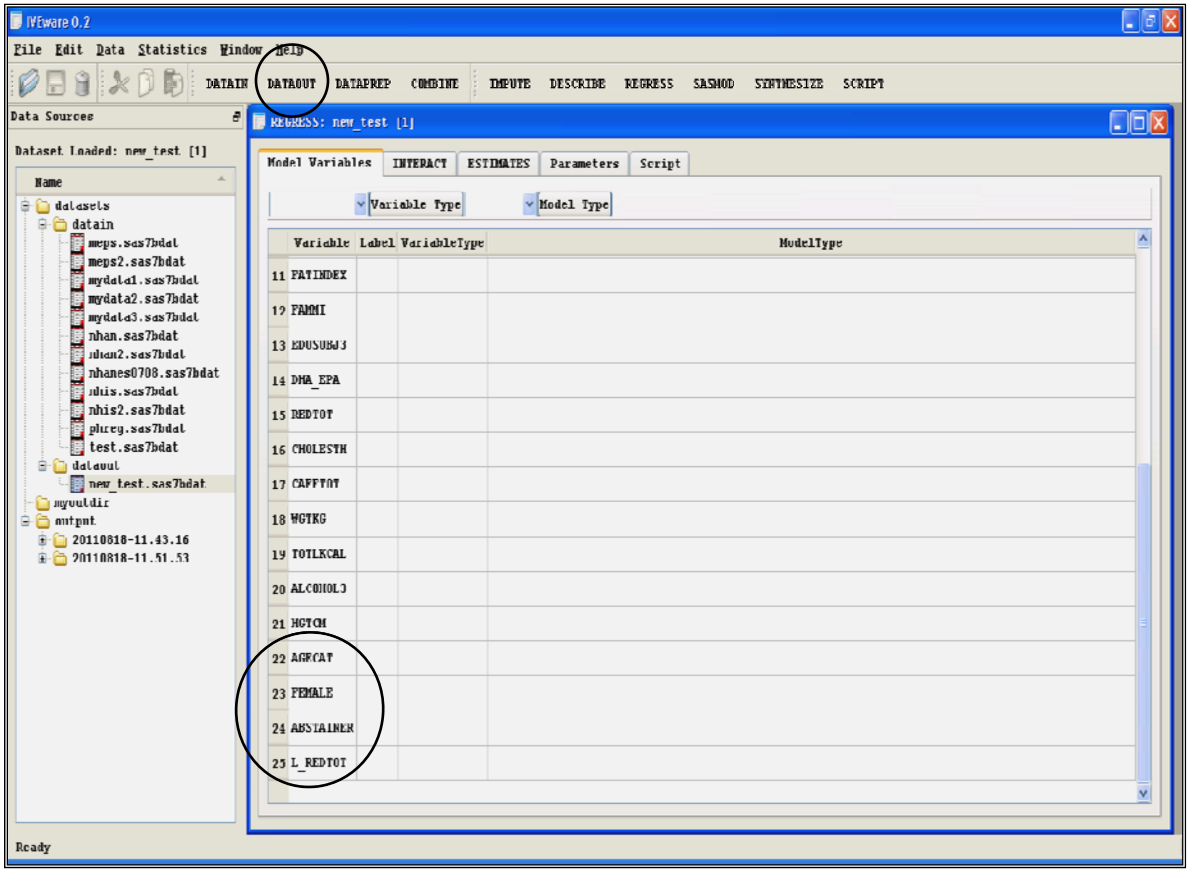
	Variable	Label
1	STUDYID	
2	CASECNT	
3	AGE	
4	GENDER	
5	RACE3	
6	HYPER	
7	DIAB	
8	SMOKE	
9	NUMCIG	
10	YRSSMOKE	
11	FATINDEX	
12	FAMHT	
13	EDUSUBJ3	
14	DHA EPA	
15	REDTOT	
16	CHOLESTH	

```
/* RECODE VARIABLES */

IF AGE LT 30                 THEN AGECAT=1;
IF AGE GE 30 AND AGE LE 49 THEN AGECAT=2;
IF AGE GE 50 AND AGE LE 64 THEN AGECAT=3;
IF AGE GE 65                 THEN AGECAT=4;

FEMALE=0;
IF GENDER=1 THEN FEMALE=1;

IF ALCOHOL3 =  0 THEN ABSTAINER=1;
IF ALCOHOL3 GT 0 THEN ABSTAINER=0;

L_REDTOT=LOG(REDTOT);

IF SMOKE IN (2,3) THEN DELETE;
```

Analyzing the Transformed Dataset

By highlighting the transformed dataset and clicking the DATAOUT button, the dataset is loaded and can be used in an *IVEware* analysis procedure.

In this example, NEW_TEST was loaded and the REGRESS procedure opened. The new variables—AGECAT, FEMALE, ABSTAINER, and L_REDTOT now appear at the end of the variable list.



	Variable	Label	VariableType	ModelType
11	FATINDEX			
12	FAMMI			
13	EDUSUBJ3			
14	DHA_EPA			
15	REDTOT			
16	CHOLESTH			
17	CAFFTOT			
18	WGTKG			
19	TOTLKCAL			
20	ALCOHOL3			
21	HGTCM			
22	AGECAT			
23	FEMALE			
24	ABSTAINER			
25	L_REDTOT			

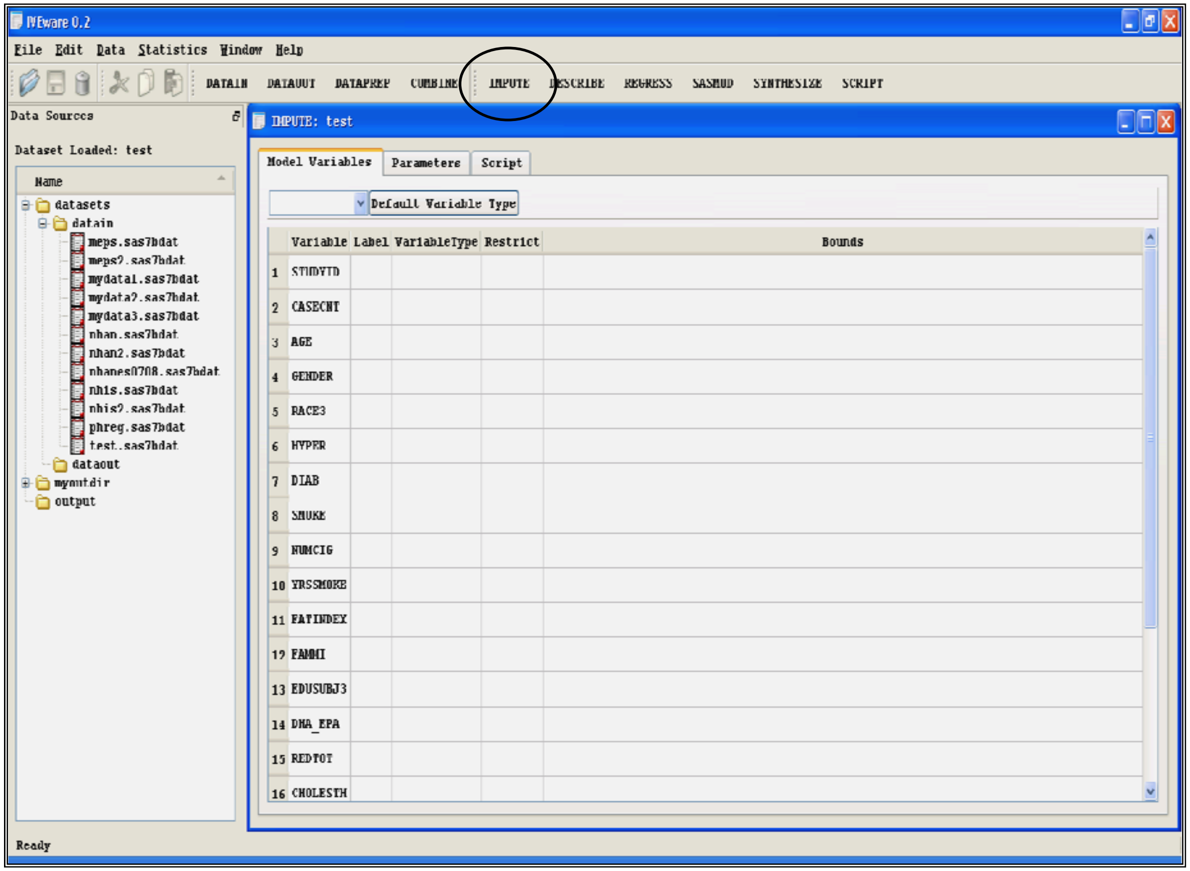
.

3. The Impute Procedure

Selecting the Impute Procedure

After loading a dataset, click on the IMPUTE button. This will open the Model Variables window, listing all variables in the loaded data set.

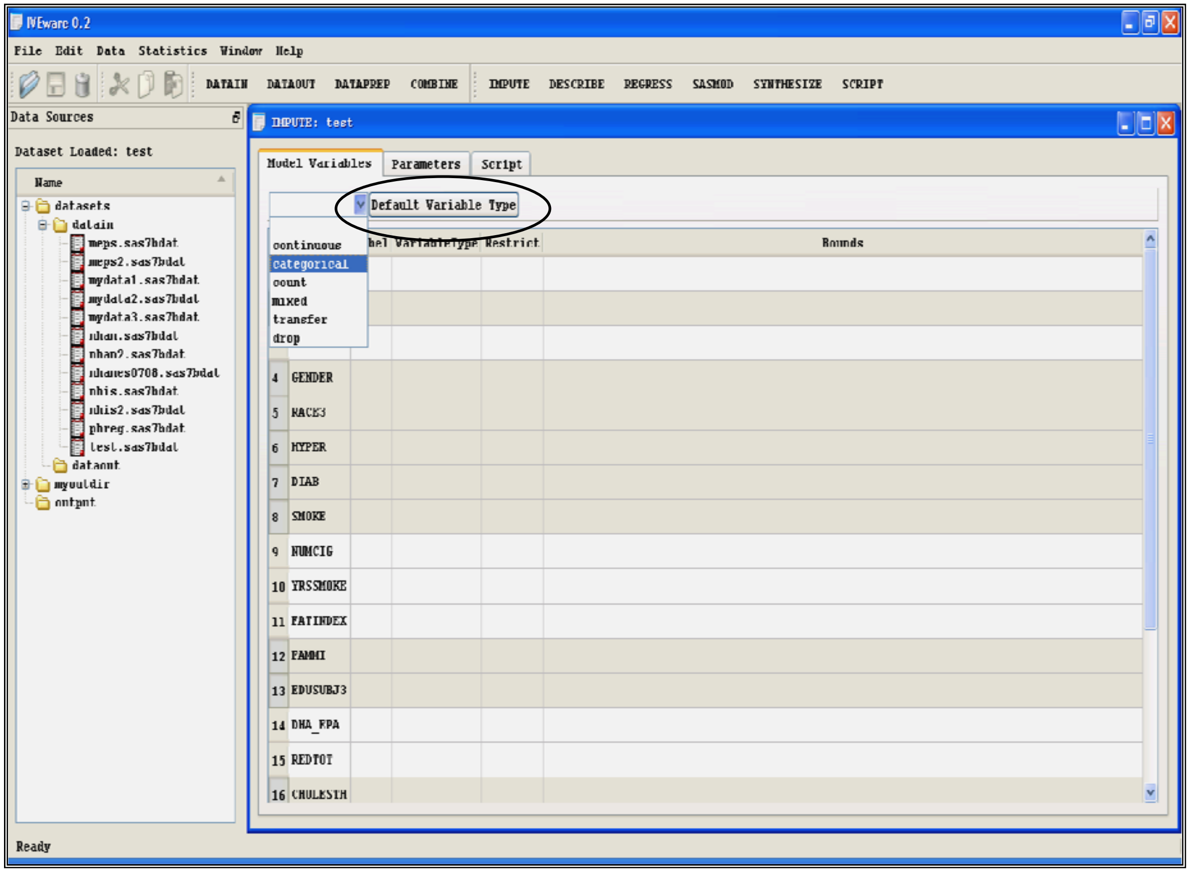
In the example, TEST was previously loaded. The dataset must be loaded before selecting an *IVEware* procedure (see page 7).



	Variable	Label	VariableType	Restrict	Bounds
1	STUDYID				
2	CASECNT				
3	AGE				
4	GENDER				
5	RACE3				
6	HYPER				
7	DIAB				
8	SMOKE				
9	NUMCIG				
10	YRSSMOKE				
11	FATINDEX				
12	FAMMI				
13	EDUSUBJ3				
14	DHA_EPA				
15	REDTOT				
16	CHOLESTH				

Declaring Variable Type

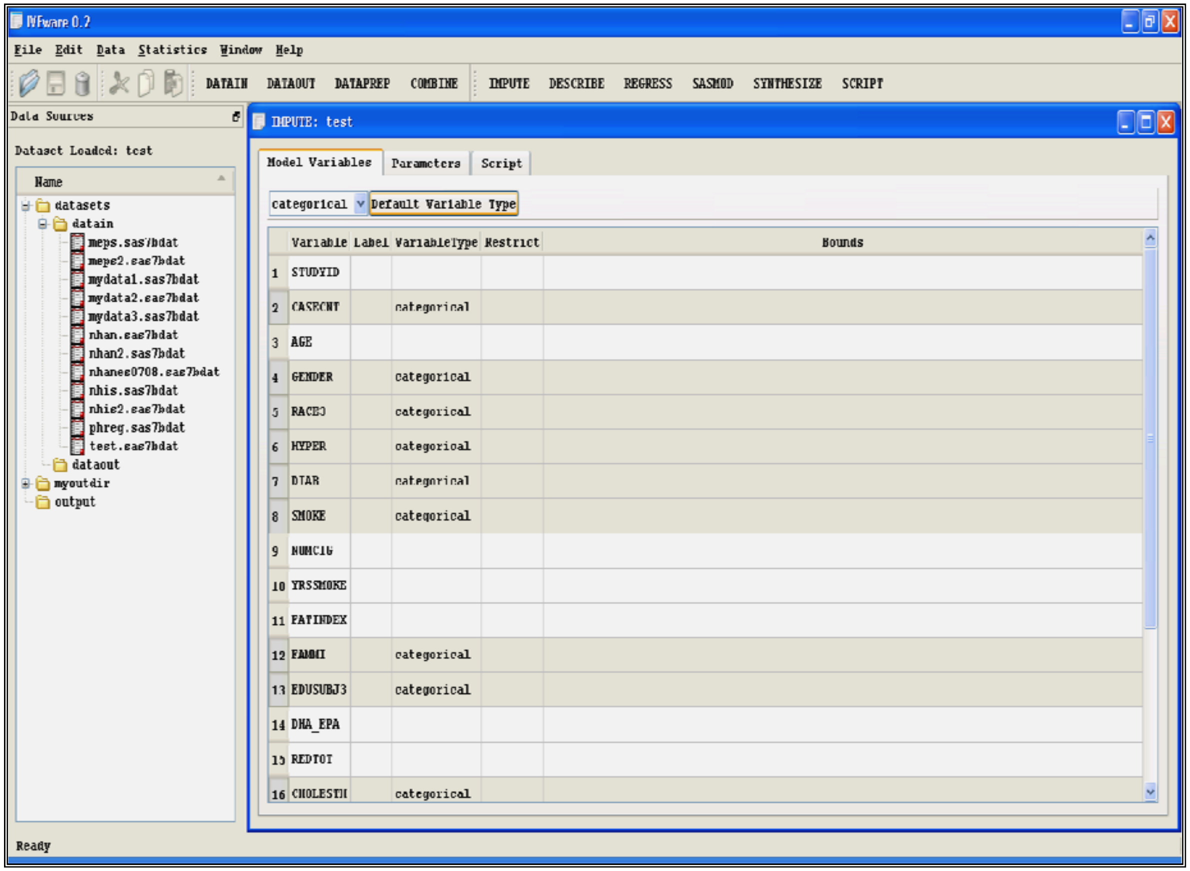
To declare a variable type, highlight the variable (s) in the Model Variables window. Click the arrow on the dialog box to the left of the DEFAULT VARIABLE TYPE button and select the appropriate type. Then click on the DEFAULT VARIABLE TYPE button.



4	GENDER
5	RACE3
6	HYPER
7	DIAB
8	SMOKE
9	NUMCIG
10	YRSSMOKE
11	FATINDEX
12	FAMMI
13	EDUSUBJ3
14	DHA_EPA
15	REDTOT
16	CHOLESTH

Declaring Variable Type

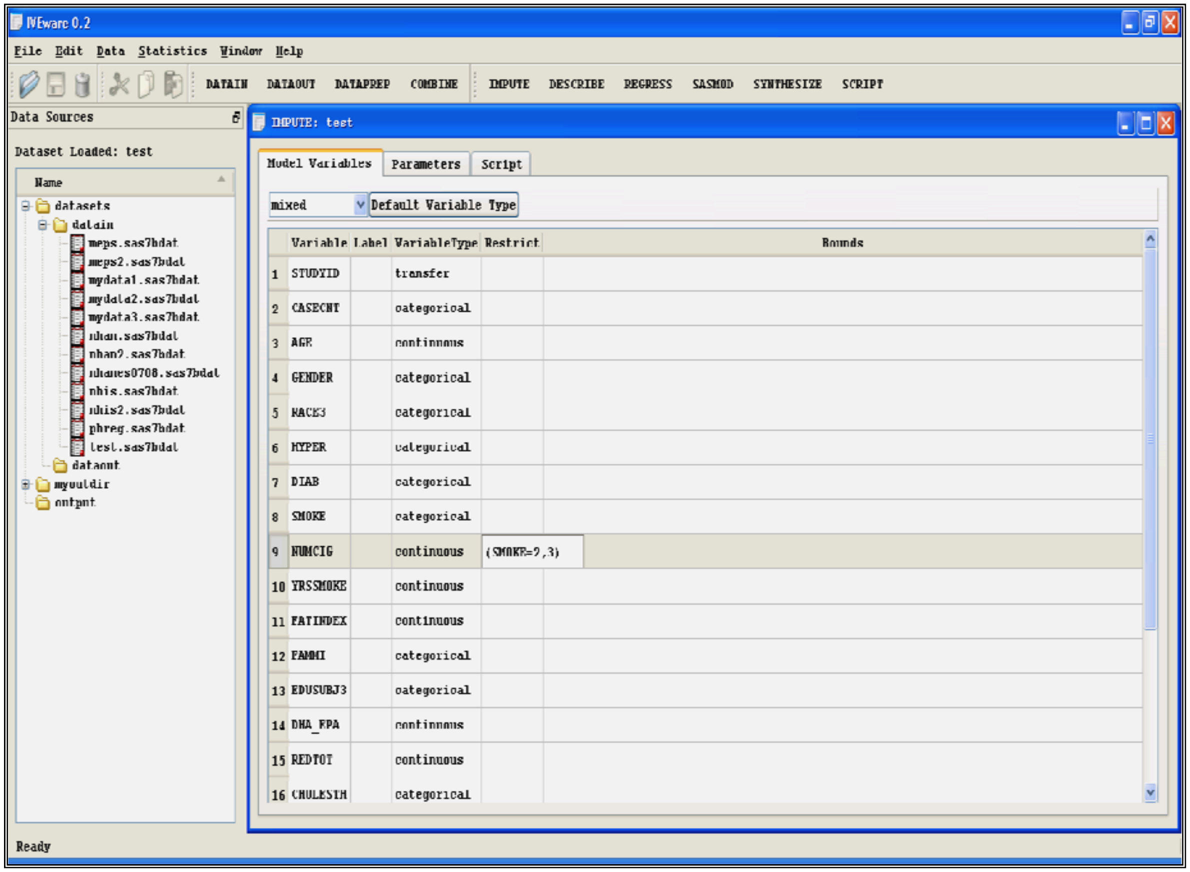
After the DEFAULT VARIABLE TYPE button is clicked the selected type appears in the column headed “Variable Type.” In the example, nine variables were declared categorical.



	Variable	Label	VariableType	Restrict	Bounds
1	STUDYID				
2	CASECNT		categorical		
3	AGE				
4	GENDER		categorical		
5	RACE3		categorical		
6	HYPER		categorical		
7	DIAB		categorical		
8	SMOKE		categorical		
9	NUMCIG				
10	YRSSMOKE				
11	FATINDEX				
12	FAMMI		categorical		
13	EDUSUBJ3		categorical		
14	DHA_EPA				
15	REDTOT				
16	CHOLESTH		categorical		

Restricting Imputation Variables

To restrict a variable, highlight a variable by clicking on its name and then click the box in the "Restrict" column. In the box, type in the restriction. In the example, the imputation of the variable NUMCIG (Number of Cigarettes Smoked) is restricted to SMOKE codes 2 and 3 (Smokers and Former Smokers).



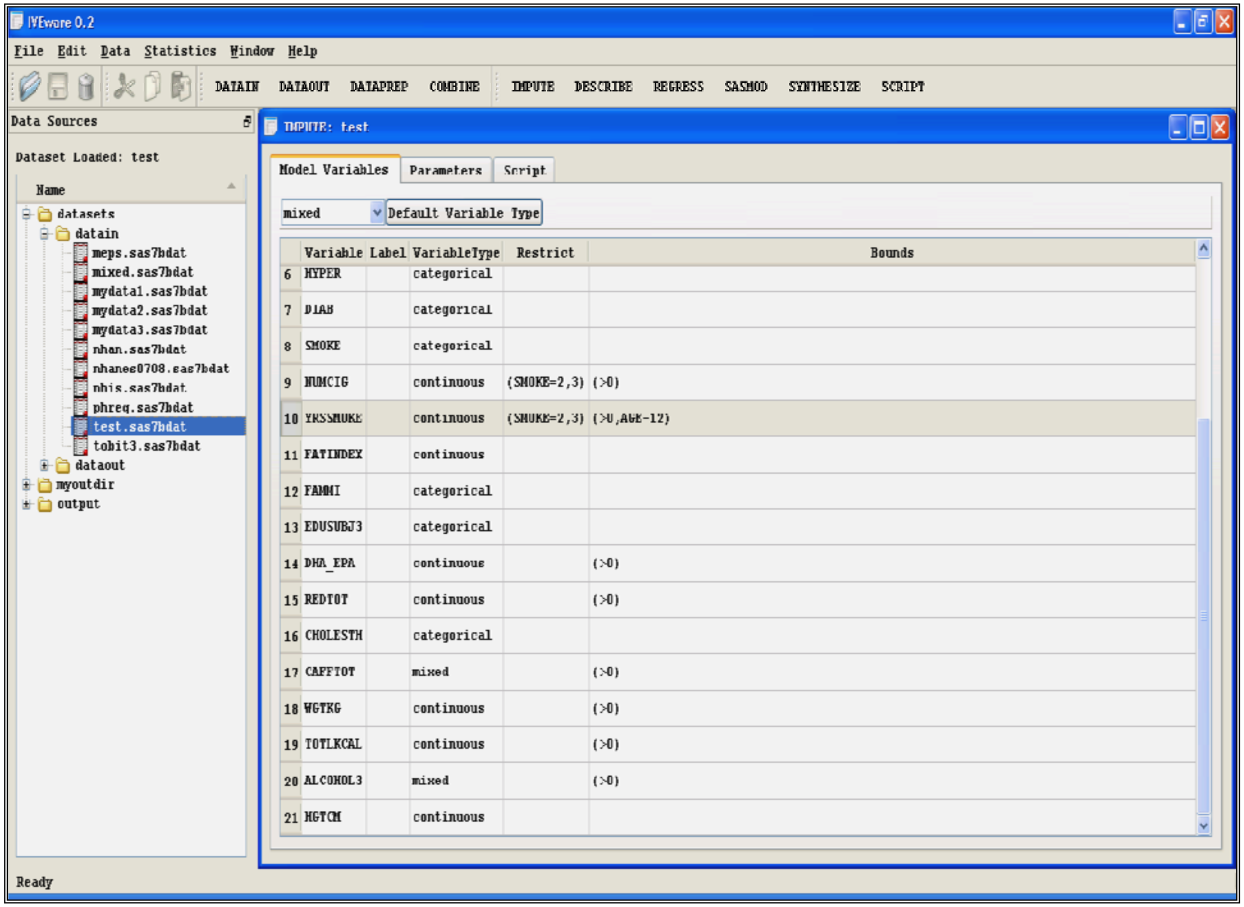
Dataset Loaded: test

mixed

	Variable	Label	VariableType	Restrict	Bounds
1	STUDYID		transfer		
2	CASECNT		categorical		
3	AGE		continuous		
4	GENDER		categorical		
5	RACE3		categorical		
6	HYPER		categorical		
7	DIAB		categorical		
8	SMOKE		categorical		
9	NUMCIG		continuous	(SMOKE=2,3)	
10	YRSSMOKE		continuous		
11	FATINDEX		continuous		
12	FAMMI		categorical		
13	EDUSUBJ3		categorical		
14	DHA_EPA		continuous		
15	REDTOT		continuous		
16	CHOLESTH		categorical		

Applying Bounds to Imputed Variables

To place bounds on an imputed value, highlight a variable by clicking on its name. Then click the box in the "Bound" column and type in the bound instruction. In the example, the imputed value of YRSSMOKE (Number of Years Smoked) must be greater than 0 and less than the respondent's current age minus 12. Smoking is assumed not to begin before the age of 12.



Dataset Loaded: test

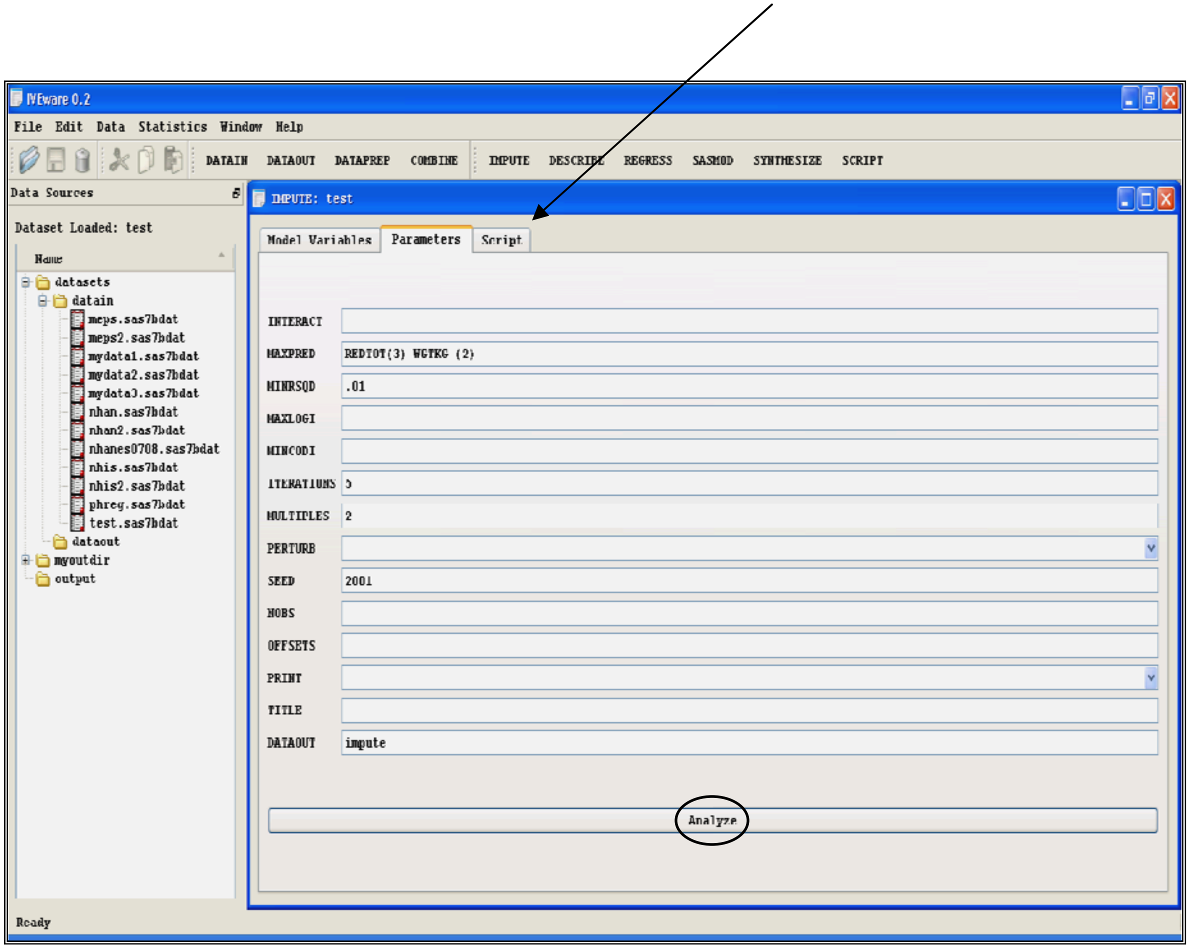
mixed

	Variable	Label	VariableType	Restrict	Bounds
6	HYPER		categorical		
7	DIAB		categorical		
8	SMOKE		categorical		
9	NUMCIG		continuous	(SMOKE=2,3)	(>0)
10	YRSSMOKE		continuous	(SMOKE=2,3)	(>0,AGE-12)
11	FATINDEX		continuous		
12	FAMMI		categorical		
13	EDUSUBJ3		categorical		
14	DHA_EPA		continuoue		(>0)
15	REDTOT		continuous		(>0)
16	CHOLESTH		categorical		
17	CAFFTOT		mixed		(>0)
18	WGTKG		continuous		(>0)
19	TOTLKCAL		continuous		(>0)
20	ALCOHOL3		mixed		(>0)
21	HGTCM		continuous		

Imputation Parameters

Clicking on the Parameters tab opens the Imputation Parameter window. Enter the appropriate parameter value in the space to the right of the parameter name. In the last parameter, DATAOUT, you can specify a name for the outputted imputed datasets. The default name is Impute.

After you complete entering all parameters, click on the ANALYZE bar at the bottom of the window. This will generate the imputation script. The script can be viewed by clicking on the SCRIPT tab.



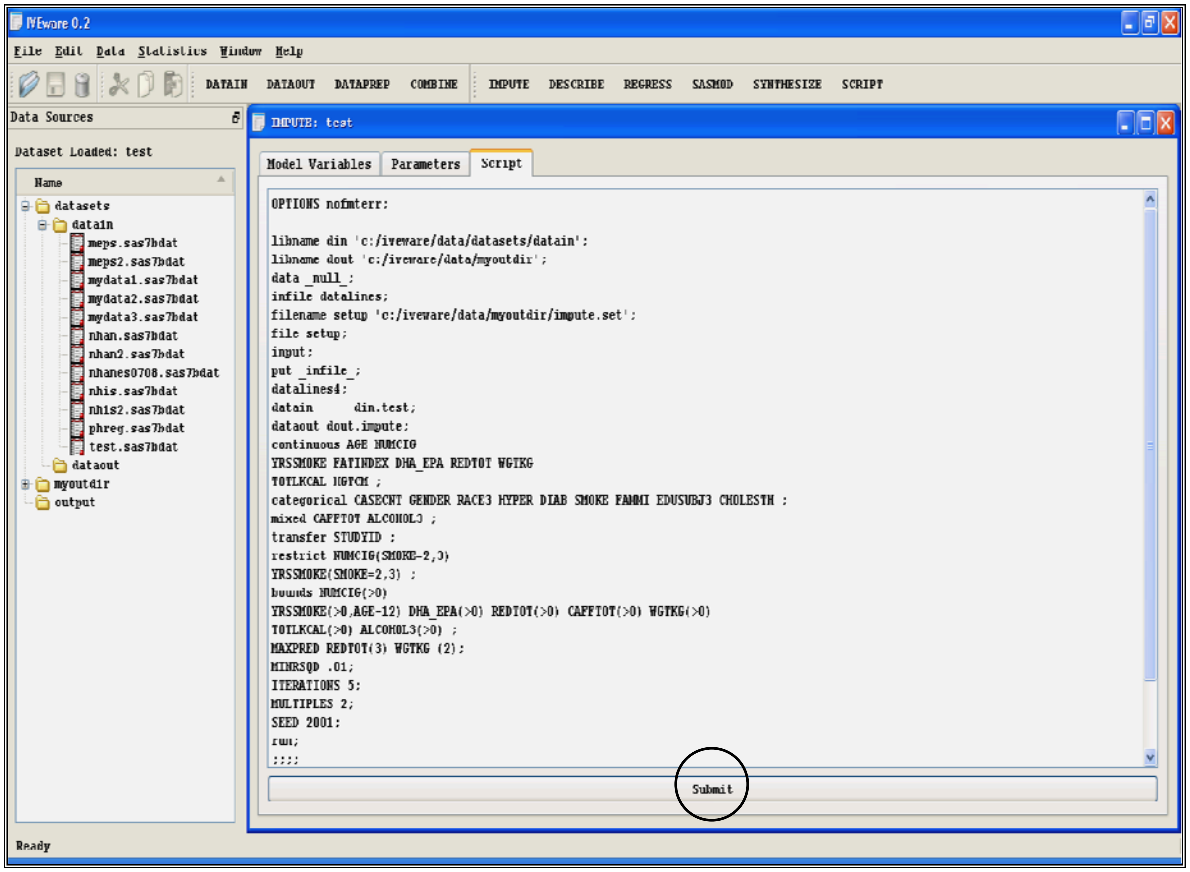
Dataset Loaded: test

INTERACT	
MAXPRED	REDTOT(3) WGTKG (2)
MINRSQD	.01
MAXLOGI	
MINCODI	
ITERATIONS	5
MULTIPLES	2
PERTURB	
SEED	2001
NOBS	
OFFSETS	
PRINT	
TITLE	
DATAOUT	impute

The Impute Script Window

The imputation script contains instructions entered in the Model Variables and Parameter windows. The script can be modified by returning to the Model Variables or Parameter window and entering changes. After entering changes you must once again click on the ANALYZE bar in Parameter window to update the script. You can also type script changes directly in the script window.

Submit your script for processing by clicking on the SUBMIT bar at the bottom of the Script window.

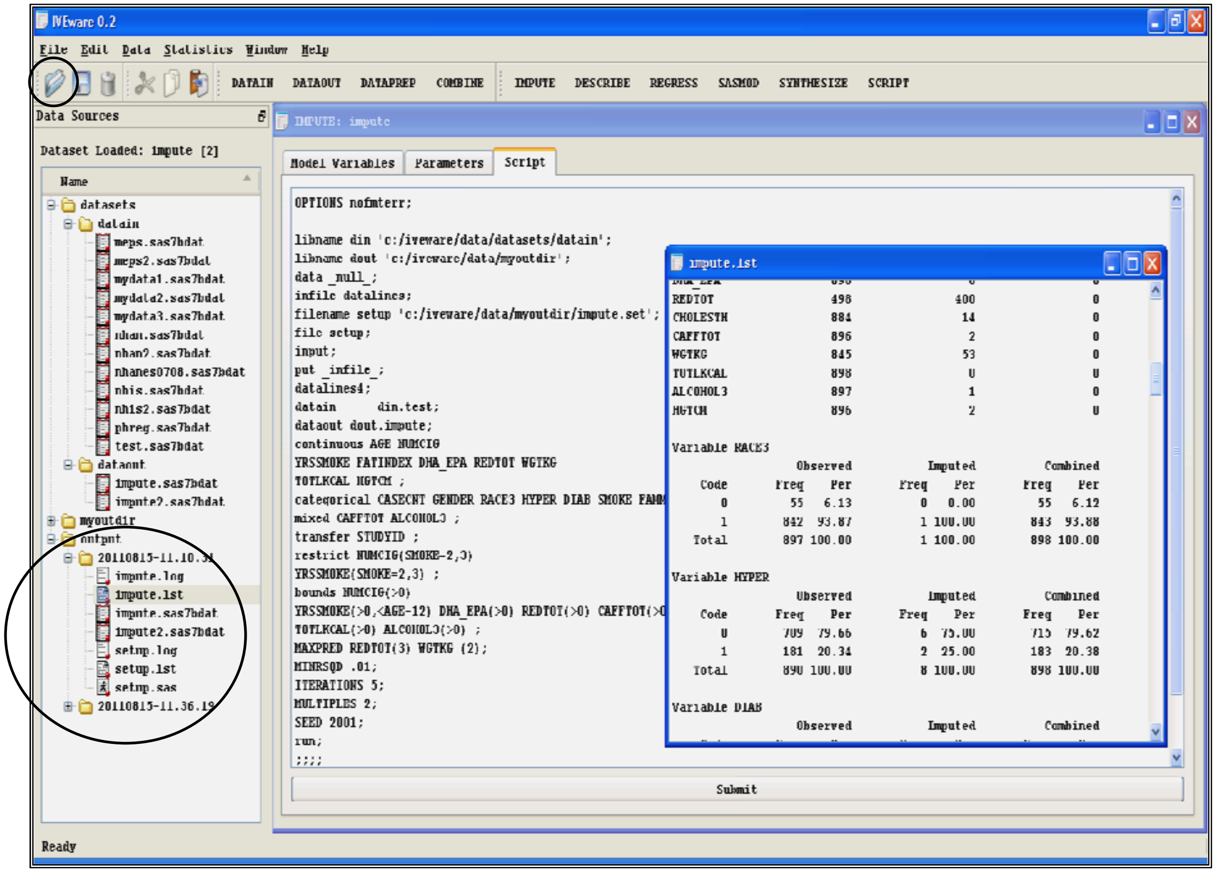


```
OPTIONS nofmterr:

libname din 'c:/iveware/data/datasets/datain':
libname dout 'c:/iveware/data/myoutdir';
data _null_;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/impute.set';
file setup;
input:
put _infile_;
datalines4:
datain      din.test;
dataout dout.impute:
continuous AGE NUMCIG
YRSSMOKE FATINDEX DHA_EPA REDTOT WGTKG
TOTLKCAL HGTCM ;
categorical CASECNT GENDER RACE3 HYPER DIAB SMOKE FAMMI EDUSUBJ3 CHOLESTH :
mixed CAFFTOT ALCOHOL3 ;
transfer STUDYID :
restrict NUMCIG(SMOKE=2,3)
YRSSMOKE(SMOKE=2,3) ;
bounds NUMCIG(>0)
YRSSMOKE(>0,AGE-12) DHA_EPA(>0) REDTOT(>0) CAFFTOT(>0) WGTKG(>0)
TOTLKCAL(>0) ALCOHOL3(>0) ;
MAXPRED REDTOT(3) WGTKG (2);
MINRSQD .01;
ITERATIONS 5:
MULTIPLES 2;
SEED 2001:
run;
;;;;
```

Output Files

After the script submission is completed, the OUTPUT folder will contain a sub-folder labeled with the date and time of the script submission. Here you will find the results of the Imputation procedure. Errors in the submitted script will be reported in the impute.log file. If the script was successful the imputation results can be found in impute.lst.

To open an output file, highlight the file and click on the File icon at the top left of the window. In the example, impute.lst has been opened.

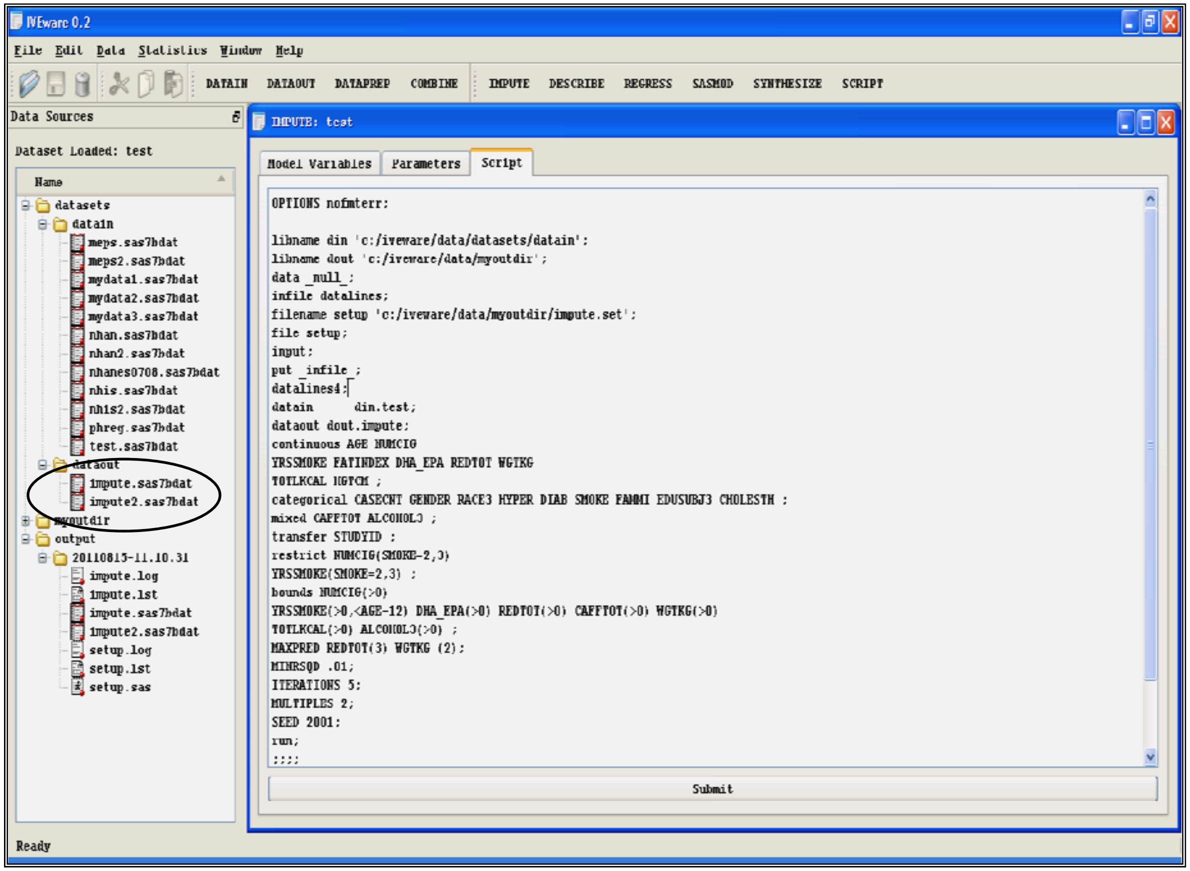


```
OPTIONS nofmterr;

libname din 'c:/iveware/data/datasets/datain';
libname dout 'c:/iveware/data/myoutdir';
data _null_;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/impute.set';
file setup;
input;
put _infile_;
datalines4;
datain      din.test;
dataout dout.impute;
continuous AGE NUMCIG
YRSSMOKE FATINDEX DHA_EPA REDTOT WGTKG
TOTLKCAL HGTCM ;
categorical CASECNT GENDER RACE3 HYPER DIAB SMOKE FAMM
mixed CAFFTOT ALCOHOL3 ;
transfer STUDYID ;
restrict NUMCIG(SMOKE=2,3)
YRSSMOKE(SMOKE=2,3) ;
bounds NUMCIG(>0)
YRSSMOKE(>0,<AGE-12) DHA_EPA(>0) REDTOT(>0) CAFFTOT(>0
TOTLKCAL(>0) ALCOHOL3(>0) ;
MAXPRED REDTOT(3) WGTKG (2);
MINRSQD .01;
ITERATIONS 5;
MULTIPLES 2;
SEED 2001;
run;
;;;;
```

```
REDTOT              498           400             0
CHOLESTH            884            14             0
CAFFTOT             896             2             0
WGTKG               845            53             0
TOTLKCAL            898             0             0
ALCOHOL3            897             1             0
HGTCM               896             2             0

Variable RACE3
                  Observed          Imputed          Combined
     Code       Freq    Per       Freq    Per       Freq    Per
        0         55   6.13          0   0.00         55   6.12
        1        842  93.87          1 100.00        843  93.88
    Total        897 100.00          1 100.00        898 100.00

Variable HYPER
                  Observed          Imputed          Combined
     Code       Freq    Per       Freq    Per       Freq    Per
        0        709  79.66          6  75.00        715  79.62
        1        181  20.34          2  25.00        183  20.38
    Total        890 100.00          8 100.00        898 100.00

Variable DIAB
                  Observed          Imputed          Combined
```

Imputed Datasets

After a successful submission of the imputation script, the imputed datasets can be found in the DATAOUT folder. The example script, calling for two imputations, outputted the datasets Impute and Impute2.

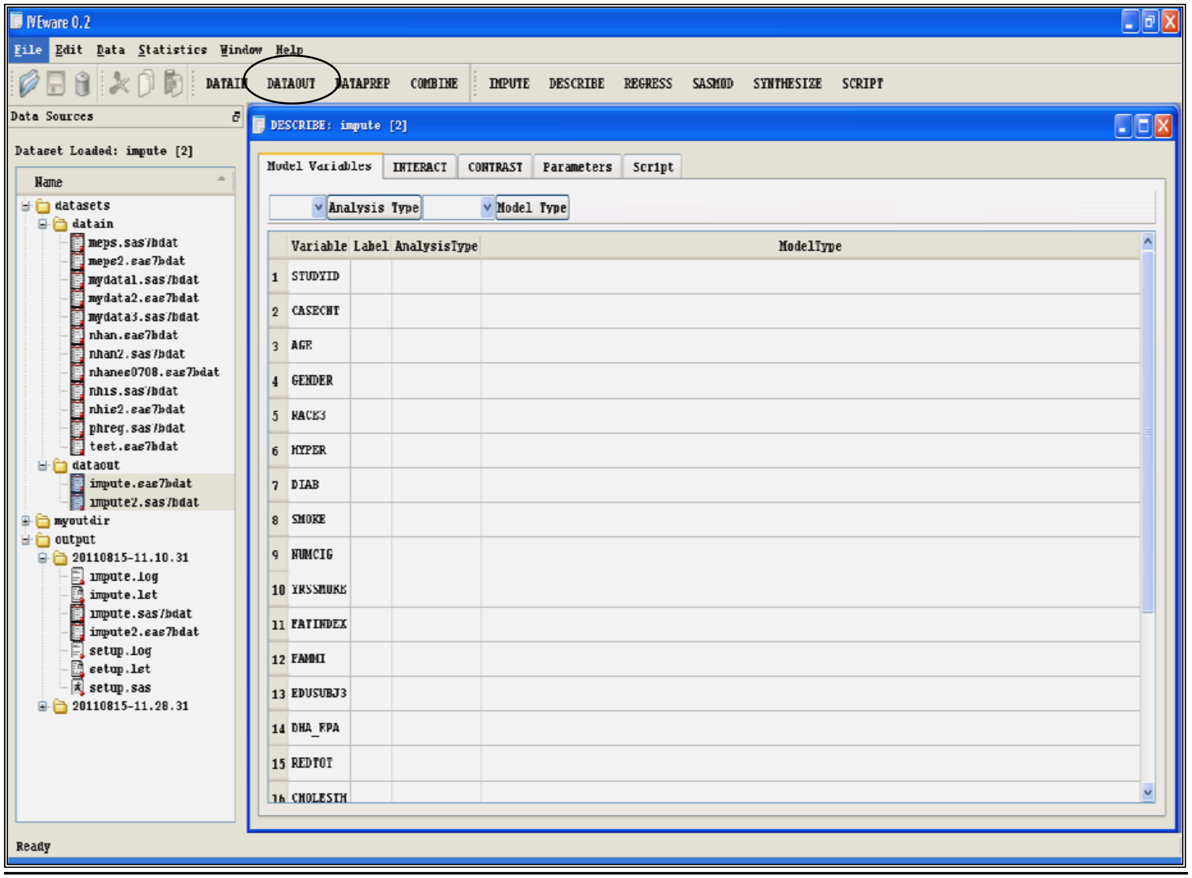


```
OPTIONS nofmterr:

libname din 'c:/iveware/data/datasets/datain':
libname dout 'c:/iveware/data/myoutdir';
data _null_;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/impute.set';
file setup;
input:
put _infile_;
datalines4;
datain      din.test;
dataout dout.impute;
continuous AGE NUMCIG
YRSSMOKE FATINDEX DHA_EPA REDTOT WGTKG
TOTLKCAL HGTCM ;
categorical CASECNT GENDER RACE3 HYPER DIAB SMOKE FAMMI EDUSUBJ3 CHOLESTH :
mixed CAFFTOT ALCOHOL3 ;
transfer STUDYID :
restrict NUMCIG(SMOKE-2,3)
YRSSMOKE(SMOKE=2,3) ;
bounds NUMCIG(>0)
YRSSMOKE(>0,<AGE-12) DHA_EPA(>0) REDTOT(>0) CAFFTOT(>0) WGTKG(>0)
TOTLKCAL(>0) ALCOHOL3(>0) ;
MAXPRED REDTOT(3) WGTKG (2);
MINRSQD .01;
ITERATIONS 5:
MULTIPLES 2;
SEED 2001:
run;
;;;;
```

Analyzing Imputed Datasets

By highlighting the imputed datasets and clicking the DATAOUT button, the imputed datasets are loaded and can be used in an *IVEware* analysis procedure.

In the example, Impute and Impute2 were loaded and the DESCRIBE procedure opened. The datasets to be analyzed are listed at the top of the DESCRIBE window. (The loaded message lists only the name of the first imputed data set loaded.)


IVEware 0.2
File Edit Data Statistics Window Help
DATAIN DATAOUT DATAPREP COMBINE IMPUTE DESCRIBE REGRESS SASMOD SYNTHESIZE SCRIPT
Data Sources
DESCRIBE: impute [2]
Model Variables INTERACT CONTRAST Parameters Script
Analysis Type
Model Type
Ready

Dataset Loaded: impute [2]

```
Name
datasets
  datain
    meps.sas7bdat
    meps2.sas7bdat
    mydata1.sas7bdat
    mydata2.sas7bdat
    mydata3.sas7bdat
    nhan.sas7bdat
    nhan2.sas7bdat
    nhanes0708.sas7bdat
    nhis.sas7bdat
    nhis2.sas7bdat
    phreg.sas7bdat
    test.sas7bdat
  dataout
    impute.sas7bdat
    impute2.sas7bdat
myoutdir
output
  20110815-11.10.31
    impute.log
    impute.lst
    impute.sas7bdat
    impute2.sas7bdat
    setup.log
    setup.lst
    setup.sas
  20110815-11.28.31
```

	Variable	Label	AnalysisType	ModelType
1	STUDYID			
2	CASECNT			
3	AGE			
4	GENDER			
5	RACE3			
6	HYPER			
7	DIAB			
8	SMOKE			
9	NUMCIG			
10	YRSSMOKE			
11	FATINDEX			
12	FAMMI			
13	EDUSUBJ3			
14	DHA_EPA			
15	REDTOT			
16	CHOLESTH			

Glossary of Impute Commands

DECLARING VARIABLE TYPES

IMPUTE requires that the SAS data set variables be defined by type. Six types of variables are recognized by the IMPUTE module: continuous, categorical, count, mixed, transfer and drop. If no variable types are specified, all variables will be assumed to be continuous. Variable types should be declared before any BOUNDS, INTERACT, or RESTRICT statements (see below).

CONTINUOUS variable list;

Variables declared as CONTINUOUS may take on any value on a continuum. Income is an example of a continuous variable. A normal linear regression model is used to impute the missing values in these variables. You may want to transform the variable to achieve normality and then impute on the transformed scale. After imputation you may re-transform the variable back to its original form.

CATEGORICAL variable list;

CATEGORICAL variables have values that represent discrete values. Gender is a categorical variable. A logistic or generalized logistic model is used to impute missing categorical values.

MIXED variable list;

Variables declared as MIXED are both categorical and continuous. In a mixed variable a value of zero is treated as a discrete category, while values greater than zero are considered continuous. Alcohol consumption is an example of a mixed variable. A two stage model is use to impute the missing values. First, a logistic regression model is used to impute zero vs. non-zero status. Conditional on imputing a non-zero status, a normal linear regression model is used to impute non-zero values.

COUNT variable list;

COUNT variables have non-negative integer values. A Poisson regression model is used to impute the missing values. The number of annual doctor visits is an example of a COUNT variable

DROP variable list;

Variables listed after the DROP keyword will be excluded from the imputation procedure and will not appear in the imputed data set.

TRANSFER variable list;

Variables listed after the TRANSFER keyword are carried over to the imputed data set, but are not imputed nor used as predictors in the imputation model. Transfer variables, however, can be used in the RESTRICT and BOUNDS statements (see below). ID is an example of a variable that you might want to treat as a transfer variable.

DEFAULT variable type;

Variable type can be Continuous, Categorical, Count, Mixed, Transfer or Drop. This keyword declares that by default all the variables in the data set should be treated as the **variable type**. The most efficient use of the DEFAULT statement is to declare the most numerous variable type in your data set as the default type, eliminating the need to type a long list of variables.

Optional Statements

RESTRICT variable(logical expression);

This command is used to restrict the imputation of a variable to those observations that satisfy the logical expression. For instance, suppose that the variable **Yrssmoke** indicates the number of years an individual smoked, and the variable **Smoke** takes the value 1 for a current smoker, 2 for a former smoker or 0 for someone who never smoked.

Then the declaration,

```
RESTRICT Yrssmoke(Smoke=1,2);
```

will impute **Yrssmoke** values only for current and former smokers. It will automatically set **Yrssmoke** equal to 0 for never smokers.

Restrictions on more than one variable may be combined as follows:

```
RESTRICT Yrssmoke(Smoke=1,2) Births(Gender=2) Income(Employed=1);
```

When the restriction is not met, the value of the restricted variable will be set to zero for continuous variables and one higher than the highest observed code for categorical variables.

BOUNDS variable (logical expression);

This keyword is useful for restricting the range of values to be imputed for a variable.

For example,

```
BOUNDS Yrssmoke (> 0,<= Age-12);
```

will ensure that the imputed values for **Yrssmoke** are between 0 and the individual’s Age minus 12. Smoking is assumed not to begin before the age of 12.

Again, as in the RESTRICT statement more than one variable can be included in the BOUNDS statement.

For example,

```
BOUNDS Yrssmoke (>0,<= Age-12) Numcig(>0);
```

Model-Building Statements

The following commands are useful in the specification of the imputation model.

INTERACT variable*variable;

This keyword enables the users to specify interaction terms to be include in the imputation regression model.

```
INTERACT Income*Income, Age*Race;
```

In this example, the imputation model for all the variables will include a square term for Income and an interaction term of Age and Race.

STEPWISE REGRESSION

MAXPRED number; OR **MAXPRED varlist2 (number);**

Specifies the maximum number of predictor variables to be included as predictors in the regression model. A step-wise regression procedure is used to select the best predictors subject to the maximum number. Setting MAXPRED to a small number of predictors will greatly reduce the computational time especially for a very large data sets but the imputations will not be fully conditional.

For example,

```
MAXPRED 5;
```

will include the five best predictor variables, the five making the largest contribution to the r-square.

You can also restrict the number of predictors for selected variables.

```
MAXPRED Income (7) Educ (3);
```

will limit the number of predictors of Income to the seven largest contributors to the r-square, while the number of predictors of Educ are limited to the three largest contributors. For other variables, all variables will be used as predictors.

MINRSQD decimal;

Specifies the minimum marginal r-squared for a stepwise regression. (Minimum initial marginal r-squared for a logistic regression, and minimum initial r-squares for any code being predicted for a polytomous regression.) This can reduce computation time. A small decimal number like 0.005 would build very large regression models whereas 0.25 will include a smaller number of predictors in the regression models. If neither MAXPRED nor MINRSQD is set then no stepwise regression will be preformed.

```
MINRSQD 0.01;
```

In this example, only variables with minimum additional r-square of 0.01 or higher will be included as predictors.

MAXLOGI number;

Specifies the maximum number of iterative algorithms to be performed in a logistic or multilogit regression model. The default is 50. This is useful if the Newton-Raphson algorithm used in producing maximum likelihood estimates does not converge after 50 iterations. This applies to the convergence criterion for the logistic, polytomous and Poisson regression models. You can check whether you have such a non-convergence problem by inspecting the log file (e.g., mysetup.log).

MINCODI decimal;

Specifies the minimum proportional change in any regression coefficient to continue the logistic regression iteration process. This applies to the convergence criterion for the logistic, polytomous and Poisson regression models.

ITERATIONS number;

Specifies the number of cycles you would like the imputation program to carry out. You can specify any number greater than or equal to 2. Current investigations show that about 10 cycles are sufficient for most imputations. You may want to experiment with several values and check the differences in the resulting analysis.

MULTIPLES number;

Indicates the number of imputations to be performed. By default only a single imputation is generated. Multiples and iterations determine *p* (see page 11). If multiples were specified as 5 and iterations as 10 then a total of 50 cycles will be performed. After every 10th cycle an imputed data set will be created.

(See section 3.4 for more information about multiple imputations.)

PERTURB instruction;

The keyword PERTURB followed by an instruction (COEF/SIR) allows the user to control perturbations of imputed values. By default the IMPUTE module will perturb model coefficients using a multivariate normal approximation of the posterior distribution and the predicted values using the appropriate regression model conditional on the perturbed coefficients. This is equivalent to using the COEF instruction. SIR uses the Sampling-Importance-Resampling algorithm to generate coefficients from the actual posterior distribution of parameters in the logistic, polytomous and Poisson regression models (See Rubin 1987a, Raghunathan and Rubin 1988, Raghunathan 1994, Gelman, et. al 1995). This is appropriate in situations where normal approximation to the posterior distribution is not appropriate.

```
PERTURB Sir;
```

SEED number;

Specifies a seed for the random draws from the posterior predictive distribution. **Number** should be greater than zero. A zero seed will result in no perturbations of the predicted values or the regression coefficients. If the SEED keyword is missing from the setup file then the seed will be determined by your computer's internal clock.

NOBS number;

NOBS indicates the number of observations to be used in the analysis. By default all observations in the data set will be used. You might use NOBS to subset a large data set while testing your setup file.

OFFSETS count variables (offset variable) ;

This statement is used to specify an offsets variable when fitting a Poisson regression model.

For example,

```
OFFSETS  Injuries(Years);
```

will fit a model predicting the number for injuries occurring per year.

PRINT instruction;

Indicates the printout desired. The options are STANDARD, DETAILS, COEF, and ALL.

For the IMPUTE procedure, the STANDARD and DETAILS keywords instruct *IVEware* to print the number and distribution of observed values, imputed values, and combined observed and imputed values for each variable.

The keyword COEF instructs IMPUTE to also print the unperturbed and perturbed coefficients for each iteration of each multiple imputation. When the ALL keyword is used, in addition to the above, the coefficient covariance matrix for each iteration of each multiple imputation is also printed.

IMPUTE also printouts a list of the variables used in the imputation model with columns indicating the number of observed cases and the number of imputed cases for each of the variables.

The third column of the variable list, labeled "double counted," is to be used for diagnostic purposes. This entry should be zero. A non-zero entry indicates that the imputed value of a restricting variable has caused the observed value of a restricted variable to be set to the restricted value (zero for continuous variables, one higher than the highest observed code for categorical variables; see RESTRICT above). This usually indicates a mis-specification of the restriction or an inconsistency in the observed data. In either case, you need to run a data step before the imputation to check the appropriateness of the restriction or correct the data inconsistency.

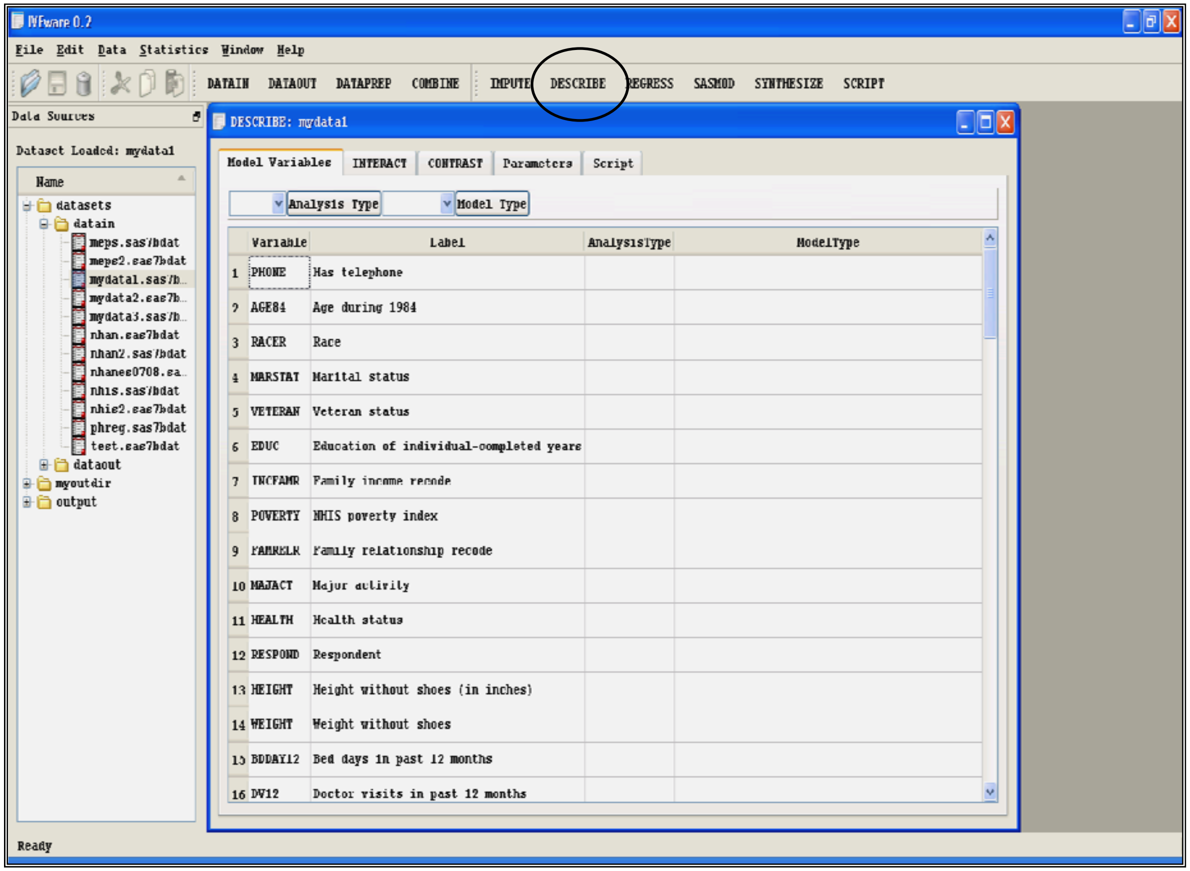
For example, if the variable SMOKE, indicating whether or not a respondent smokes, is missing and the variable YRSMK, indicating the number of years the respondent has smoked, is observed, then logically the respondent should be classified as a smoker. If SMOKE is not given a value indicating the respondent is a smoker in a SAS data step prior to imputation, the missing value could possibly be imputed to a nonsmoker value, causing IMPUTE to change the observed value for YRSMK to zero.

4. The Describe Procedure

Selecting the Describe Procedure

After loading a dataset, click on the DESCRIBE button. This will open the Model Variables window, listing all variables in the loaded dataset.

In this example, MYDATA1 was previously loaded. The dataset must be loaded before selecting an *IVEware* procedure (see page 7).

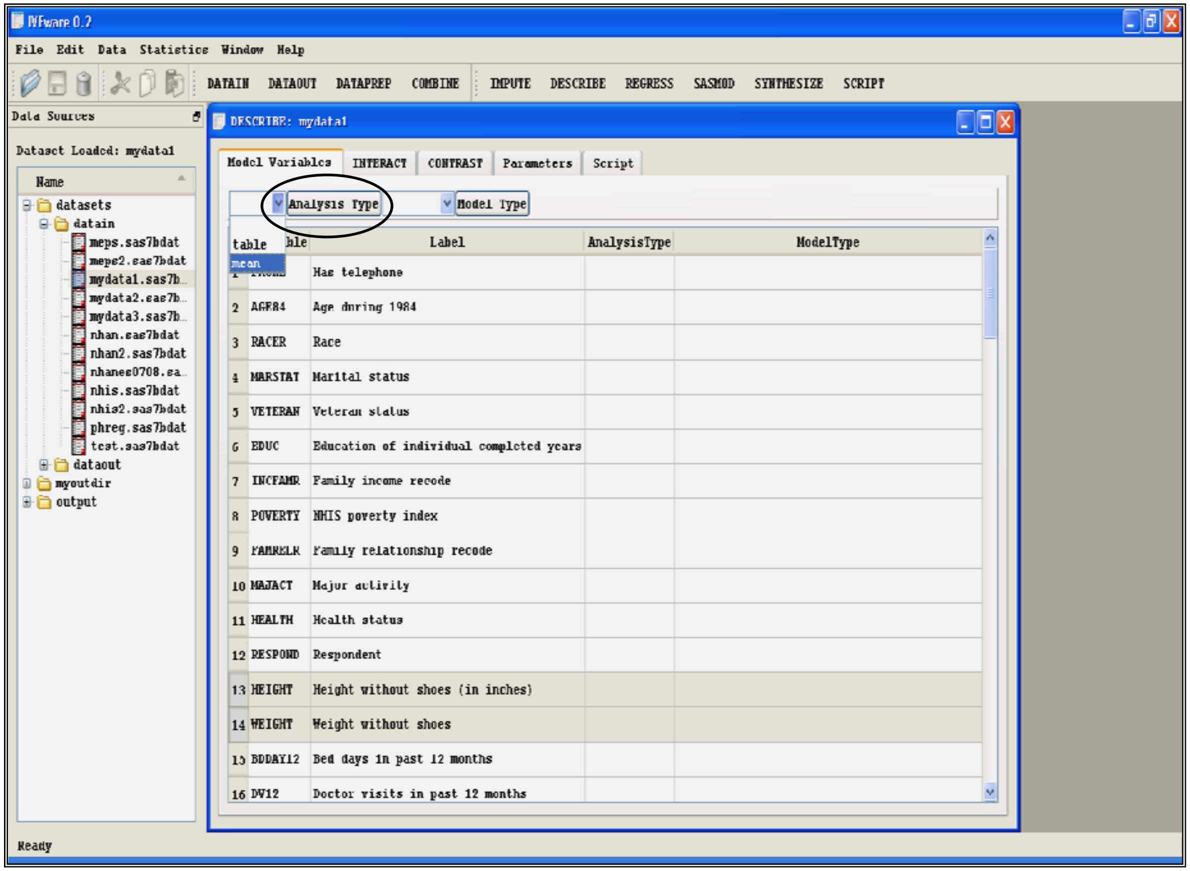


Dataset Loaded: mydata1

	Variable	Label	AnalysisType	ModelType
1	PHONE	Has telephone		
2	AGE84	Age during 1984		
3	RACER	Race		
4	MARSTAT	Marital status		
5	VETERAN	Veteran status		
6	EDUC	Education of individual-completed years		
7	INCFAMR	Family income recode		
8	POVERTY	NHIS poverty index		
9	FAMRELR	Family relationship recode		
10	MAJACT	Major activity		
11	HEALTH	Health status		
12	RESPOND	Respondent		
13	HEIGHT	Height without shoes (in inches)		
14	WEIGHT	Weight without shoes		
15	BDDAY12	Bed days in past 12 months		
16	DV12	Doctor visits in past 12 months		

Variable Selection and Analysis Type

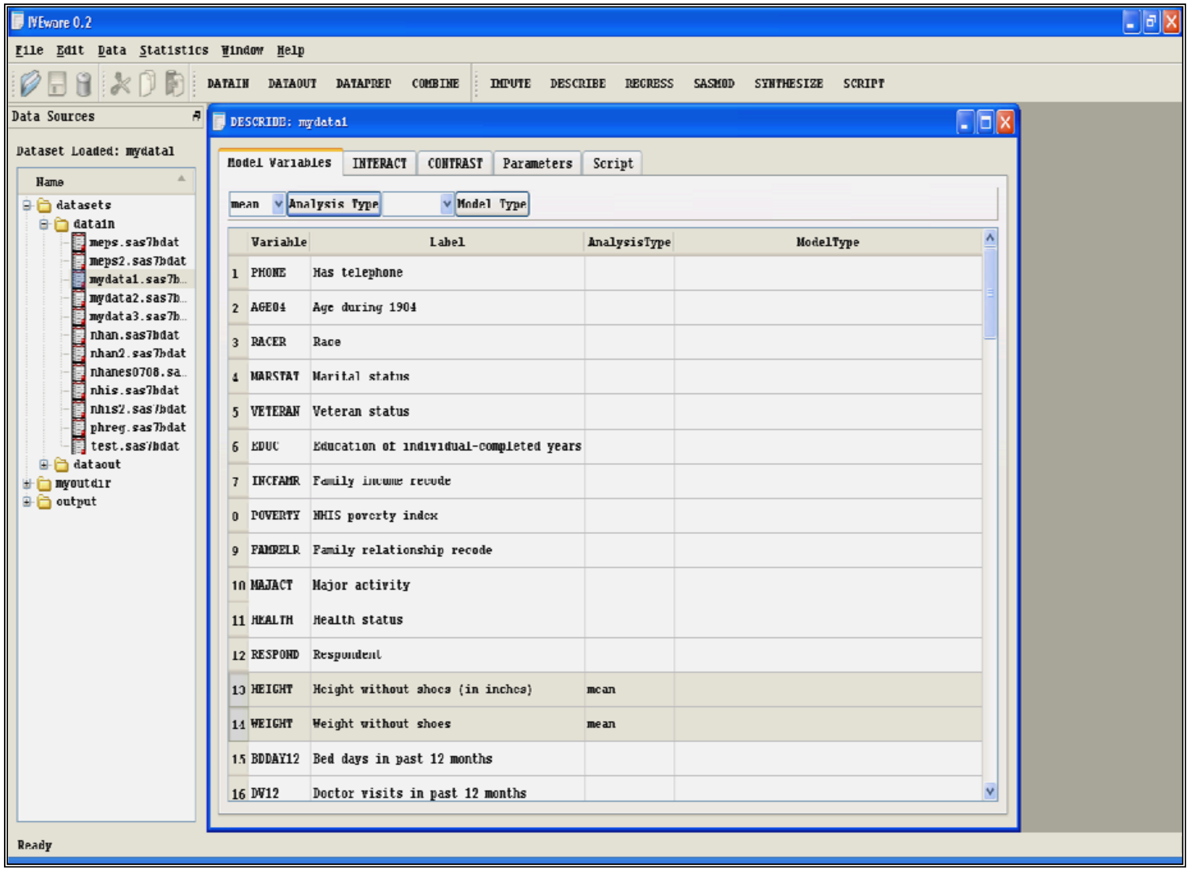
To perform a descriptive analysis of a variable (s), highlight the variable in the Model Variables window. Click the arrow on the dialog box to the left of the ANALYSIS TYPE button and select Table or Mean as your analysis type. Then click on the ANALYSIS TYPE button.



	ble	Label	AnalysisType	ModelType
1	[illegible]	Has telephone		
2	AGE84	Age during 1984		
3	RACER	Race		
4	MARSTAT	Marital status		
5	VETERAN	Veteran status		
6	EDUC	Education of individual completed years		
7	INCFAMR	Family income recode		
8	POVERTY	NHIS poverty index		
9	FAMRELR	Family relationship recode		
10	MAJACT	Major activity		
11	HEALTH	Health status		
12	RESPOND	Respondent		
13	HEIGHT	Height without shoes (in inches)		
14	WEIGHT	Weight without shoes		
15	BDDAY12	Bed days in past 12 months		
16	DV12	Doctor visits in past 12 months		

Variable Selection and Analysis Type

After ANALYSIS TYPE is clicked, the selected type appears in the column headed “Analysis Type.” In the example, the means and standard errors will be calculated for the variables HEIGHT and WEIGHT.



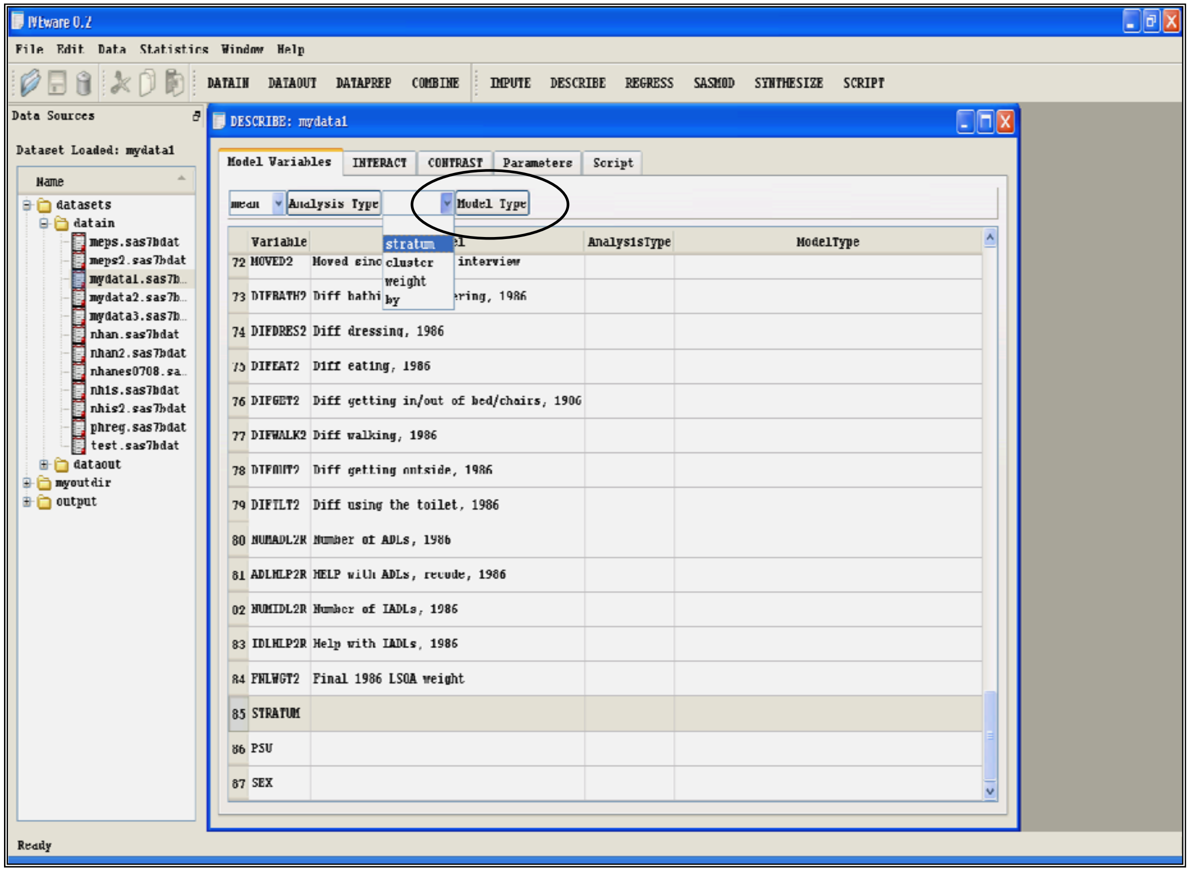
Dataset Loaded: mydata1

mean

	Variable	Label	AnalysisType	ModelType
1	PHONE	Has telephone		
2	AGE04	Age during 1904		
3	RACER	Race		
4	MARSTAT	Marital status		
5	VETERAN	Veteran status		
6	EDUC	Education of individual-completed years		
7	INCFAMR	Family income recode		
0	POVERTY	NHIS poverty index		
9	FAMRELR	Family relationship recode		
10	MAJACT	Major activity		
11	HEALTH	Health status		
12	RESPOND	Respondent		
13	HEIGHT	Height without shoes (in inches)	mean	
14	WEIGHT	Weight without shoes	mean	
15	BDDAY12	Bed days in past 12 months		
16	DV12	Doctor visits in past 12 months		

Defining Model Type

To incorporate survey design variables in your analysis, highlight the design variable in the Model Variables window. Click the arrow on the dialog box to the left of MODEL TYPE button and select the appropriate design feature—stratum, cluster or weight. Then click on the MODEL TYPE button to the right of the dialogue box.



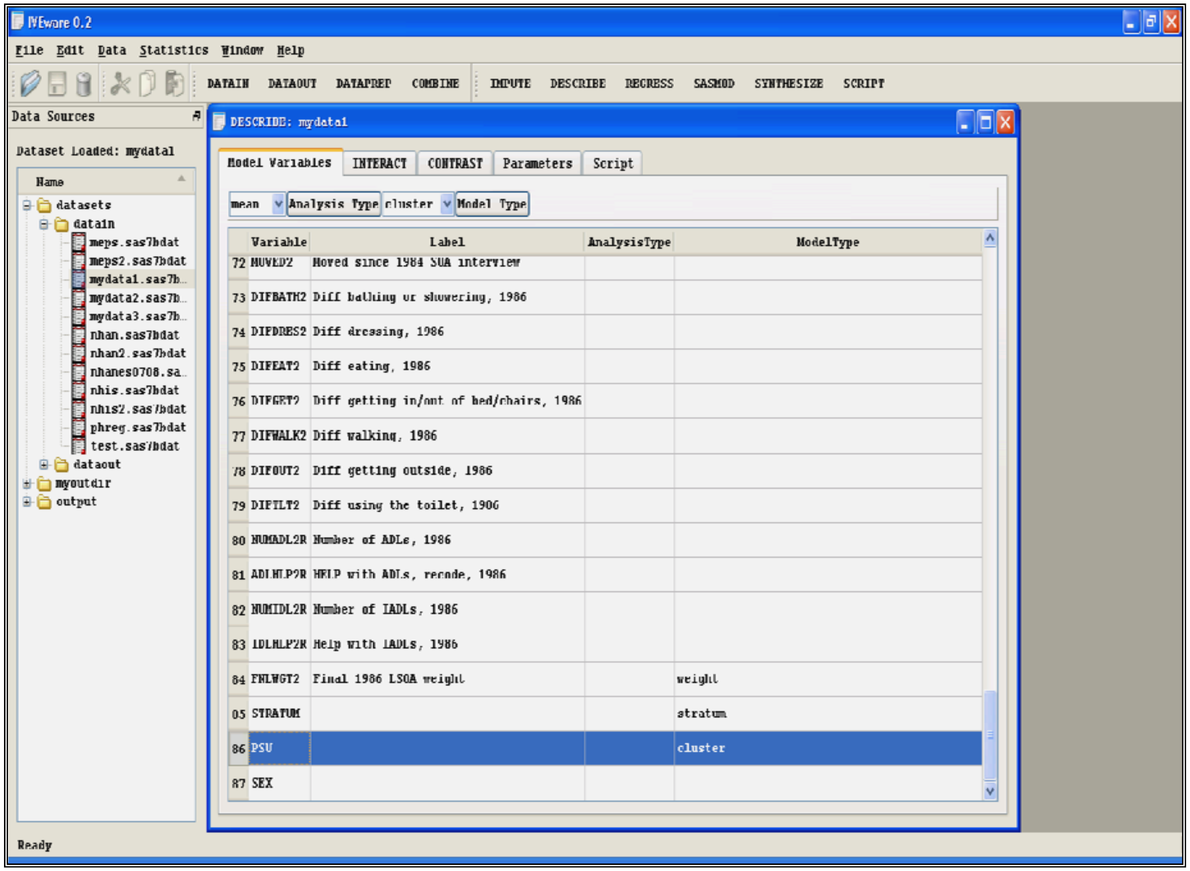
Dataset Loaded: mydata1

Name
- datasets
  - datain
    - meps.sas7bdat
    - meps2.sas7bdat
    - mydata1.sas7b...
    - mydata2.sas7b...
    - mydata3.sas7b...
    - nhan.sas7bdat
    - nhan2.sas7bdat
    - nhanes0708.sa...
    - nhis.sas7bdat
    - nhis2.sas7bdat
    - phreg.sas7bdat
    - test.sas7bdat
  - dataout
- myoutdir
- output

	Variable	el	AnalysisType	ModelType
72	MOVED2	Moved sinc interview		
73	DIFBATH2	Diff bathi ering, 1986		
74	DIFDRES2	Diff dressing, 1986		
75	DIFEAT2	Diff eating, 1986		
76	DIFGET2	Diff getting in/out of bed/chairs, 1986		
77	DIFWALK2	Diff walking, 1986		
78	DIFOUT2	Diff getting outside, 1986		
79	DIFTLT2	Diff using the toilet, 1986		
80	NUMADL2R	Number of ADLs, 1986		
81	ADLHLP2R	HELP with ADLs, recode, 1986		
82	NUMIDL2R	Number of IADLs, 1986		
83	IDLHLP2R	Help with IADLs, 1986		
84	FNLWGT2	Final 1986 LSOA weight		
85	STRATUM			
86	PSU			
87	SEX			

Defining Model Type

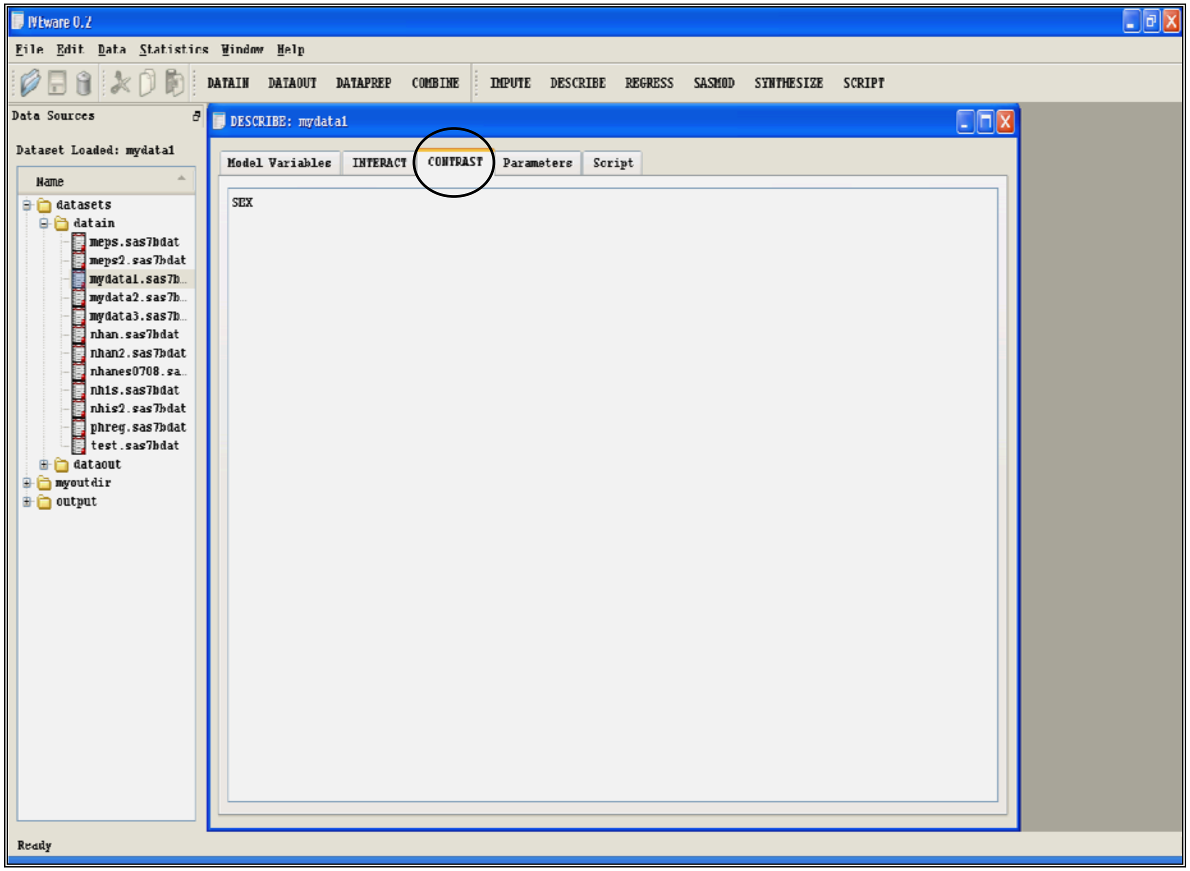
After MODEL TYPE is clicked, the selected design feature appears in the column headed “Model Type.” In the example, FNLWGT2 was assigned as the weight variable, STRATUM as the stratum variable, and PSU as the cluster variable.



	Variable	Label	AnalysisType	ModelType
72	MOVED2	Moved since 1984 SOA interview		
73	DIFBATH2	Diff bathing or showering, 1986		
74	DIFDRES2	Diff dressing, 1986		
75	DIFEAT2	Diff eating, 1986		
76	DIFGET2	Diff getting in/out of bed/chairs, 1986		
77	DIFWALK2	Diff walking, 1986		
78	DIFOUT2	Diff getting outside, 1986		
79	DIFTLT2	Diff using the toilet, 1906		
80	NUMADL2R	Number of ADLs, 1986		
81	ADLHLP2R	HELP with ADLs, recode, 1986		
82	NUMIDL2R	Number of IADLs, 1986		
83	IDLHLP2R	Help with IADLs, 1986		
84	FNLWGT2	Final 1986 LSOA weight		weight
05	STRATUM			stratum
86	PSU			cluster
87	SEX			

The Describe Contrast Window

A Describe analysis can be further defined with a contrast statement. Click on the Contrast tab and type in the contrast variable. In the example, analysis results will be contrasted by respondent sex. Multiple contrasts can also be specified (i.e.; SEX RACER) as well as complex contrasts (i.e.; SEX*RACER).

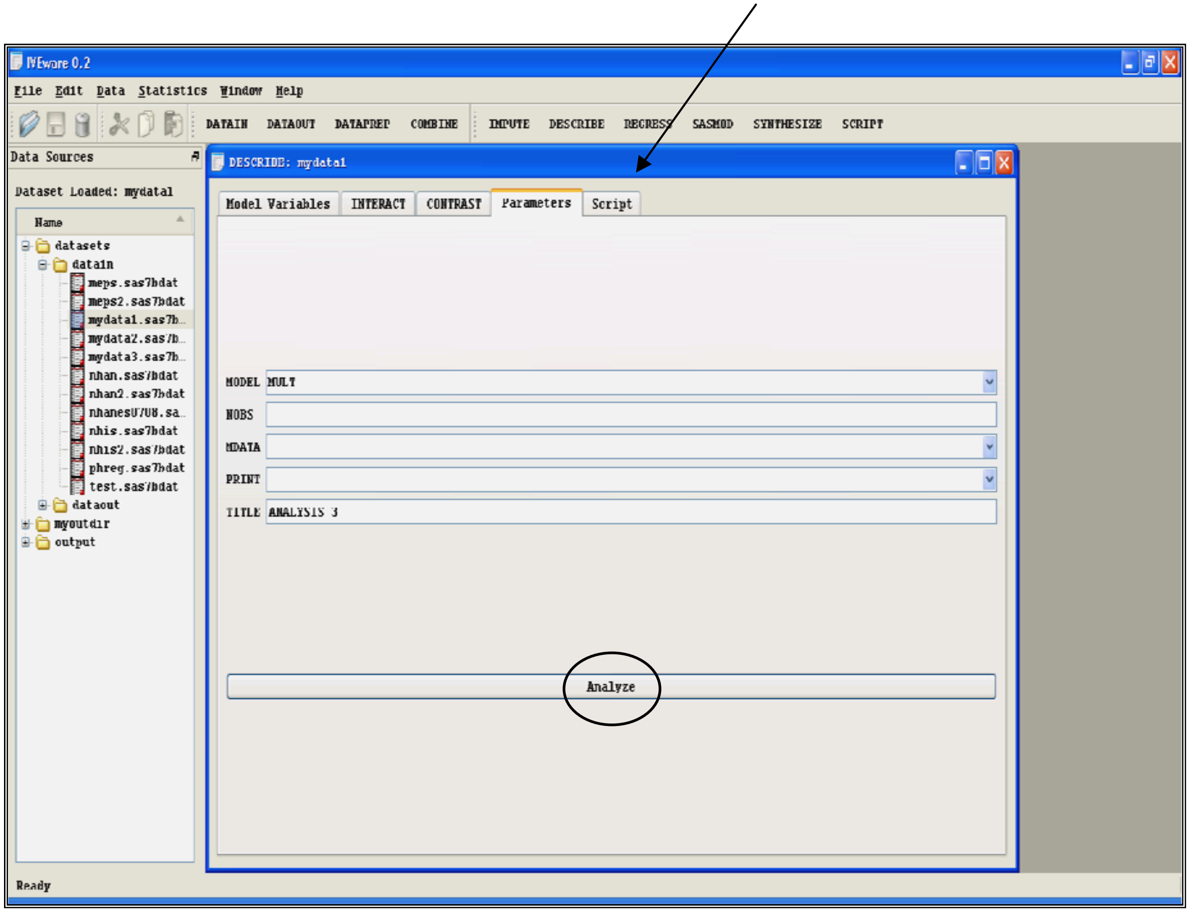


SEX

Describe Parameters

Clicking on the Parameters tab opens the Describe parameter window. Enter the appropriate parameter value in the space to the right of the parameter name. In the example, the default method of variance estimation, MULT, has been retained and the analysis title “ANALYSIS 3” has been added.

After you complete entering all parameters, click on the ANALYZE bar at the bottom of the window. This will generate the Describe script. The script can be viewed by clicking on the SCRIPT tab.



MODEL MULT

NOBS

MDATA

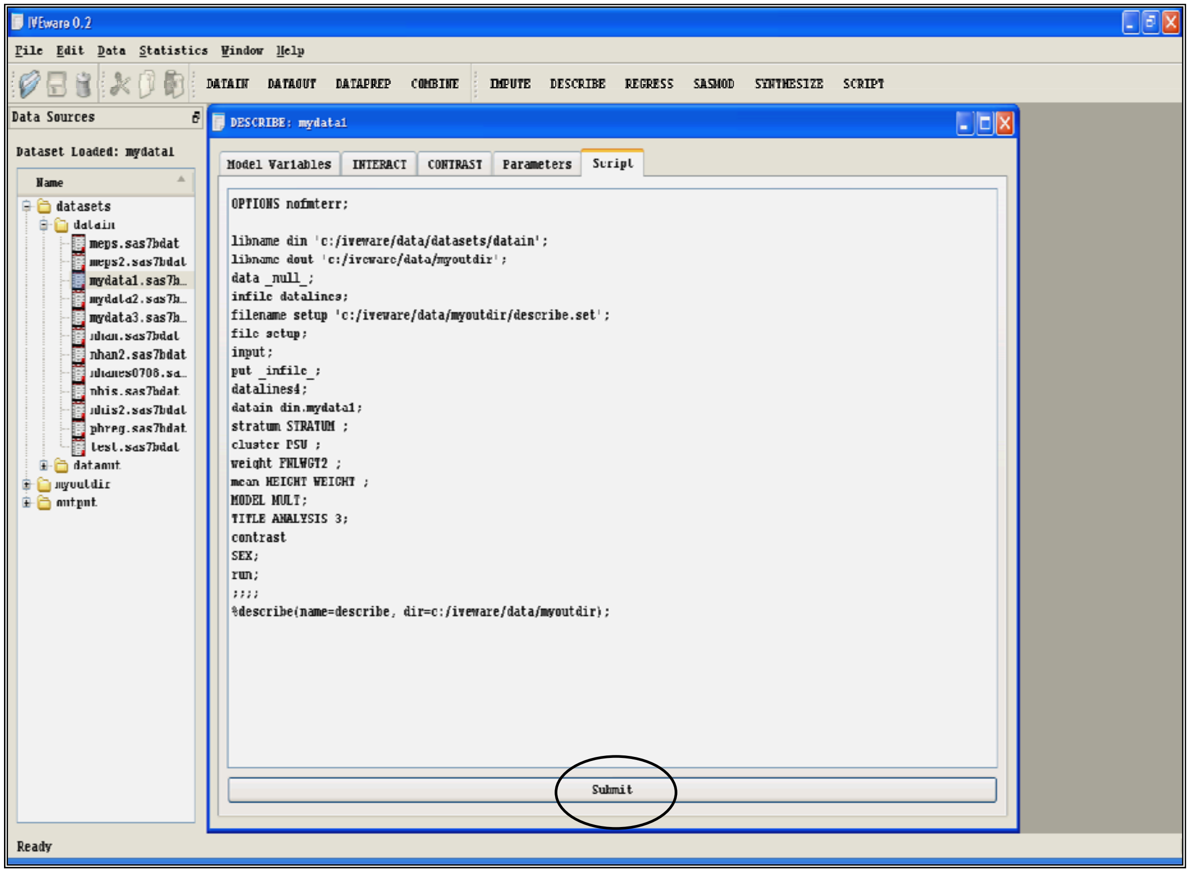
PRINT

TITLE ANALYSIS 3

The Describe Script Window

The Describe script contains instructions entered in the Model Variables, Contrast, and Parameter windows. The script can be modified by returning to those windows and entering changes. After entering changes you must once again click on the ANALYZE bar in Parameter window to update the script. You can also type script changes directly in the script window.

Submit your script for processing by clicking on the SUBMIT bar at the bottom of the Script window.



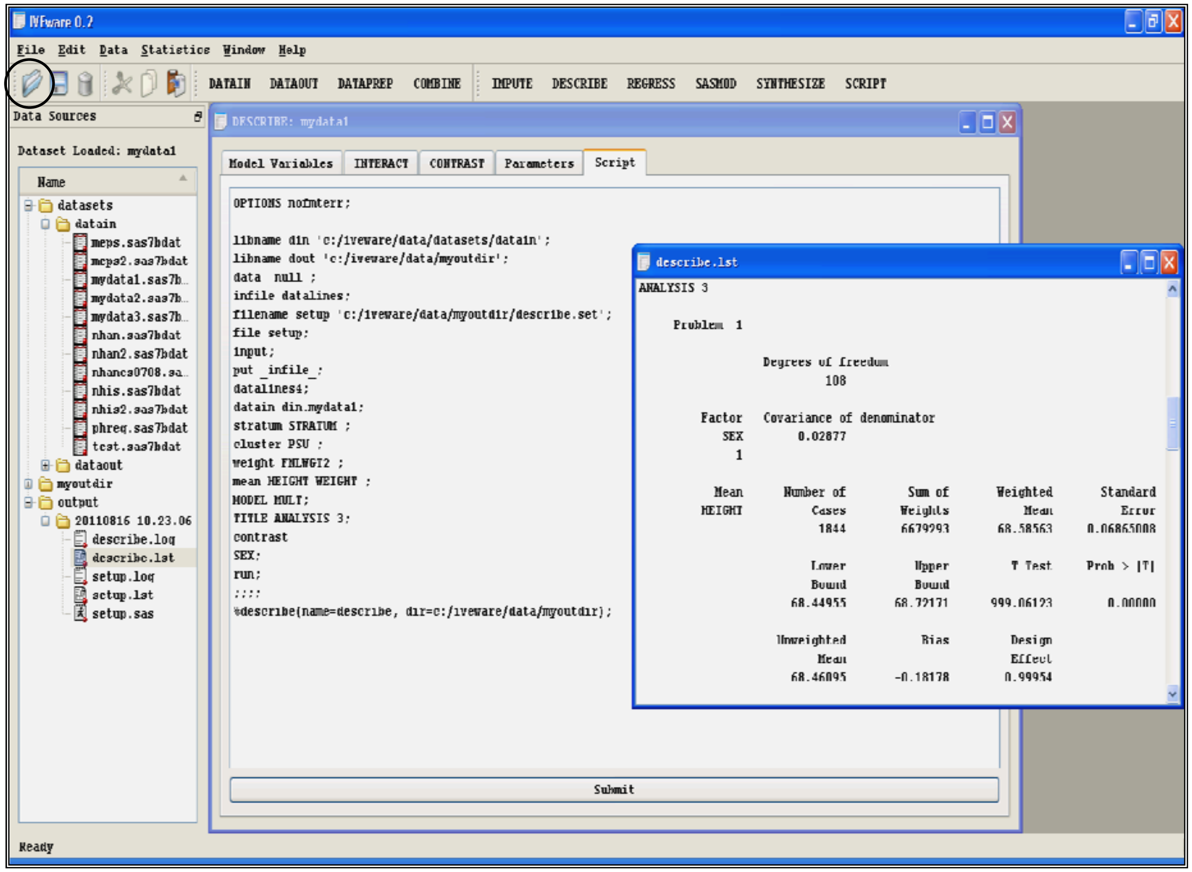
```
OPTIONS nofmterr;

libname din 'c:/iveware/data/datasets/datain';
libname dout 'c:/iveware/data/myoutdir';
data _null_;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/describe.set';
file setup;
input;
put _infile_;
datalines4;
datain din.mydata1;
stratum STRATUM ;
cluster PSU ;
weight FNLWGT2 ;
mean HEIGHT WEIGHT ;
MODEL MULT;
TITLE ANALYSIS 3;
contrast
SEX;
run;
;;;;
%describe(name=describe, dir=c:/iveware/data/myoutdir);
```

..

Output Files

After the script submission is completed, the OUTPUT folder will contain a sub-folder labeled with the date and time of the script submission. Here you will find the results of the Describe procedure. Errors in the submitted script will be reported in the describe.log file. If the script was successful the Describe results can be found in describe.lst.

To open an output file, highlight the file and click on the File icon at the top left of the window. In the example, describe.lst has been opened.



```
OPTIONS nofmterr;

libname din 'c:/iveware/data/datasets/datain';
libname dout 'c:/iveware/data/myoutdir';
data  null ;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/describe.set';
file setup;
input;
put _infile_;
datalines4;
datain din.mydata1;
stratum STRATUM ;
cluster PSU ;
weight FMLWGT2 ;
mean HEIGHT WEIGHT ;
MODEL MULT;
TITLE ANALYSIS 3;
contrast
SEX;
run;
;;;;
%describe(name=describe, dir=c:/iveware/data/myoutdir);
```

```
ANALYSIS 3


     Problem  1


                   Degrees of freedom
                             108


          Factor   Covariance of denominator
             SEX         0.02877
               1


            Mean     Number of       Sum of     Weighted     Standard
          HEIGHT         Cases      Weights         Mean        Error
                          1844      6679293     68.58563   0.06865008


                         Lower        Upper       T Test   Prob > |T|
                         Bound        Bound
                      68.44955     68.72171    999.06123      0.00000


                    Unweighted         Bias       Design
                          Mean                    Effect
                      68.46095     -0.18178      0.99954
```

Glossary of Describe Commands

STRATUM variable name;

variable name is the name of the stratum variable No missing values are allowed for the stratum variable**.**

CLUSTER variable name;

variable name is the Primary Sampling Unit (PSU) or Sampling Error Computing Unit (SECU) variable. No missing values are allowed for the cluster variable.

WEIGHT variable name;

variable name is the sampling weight variable. Sampling weights are usually the product of selection, nonresponse adjustment and poststratification weights. No missing values are allowed for the weight variable.

MODEL method;

MODEL indicates the variance estimation method to be used. **Mult** (Default) is useful when there are multiple PSUs within a stratum, **Pair** employs the paired selection method, and **Diff** employs the successive differences method. You can specify different methods for each stratum.

For example,

```
MODEL Pair(15,16,17) Diff(20,21,27);
```

will use paired differences for strata 15, 16, 17, the successive differences for strata 20, 21,27, and **Mult** for the rest.

TABLE variable list;

This command will produce the weighted proportions and their standard errors for all levels of a variable(s).

```
TABLE Race;
```

Crosstabulations may be indicated with an asterisk, for example:

```
TABLE Race*Gender;
```

MEAN variable list;

Means, standard errors, and design effects are calculated for the list of variables following the MEAN keyword.

For example,

```
MEAN BMI Age;
```

will compute the means of BMI and Age.

On the other hand,

```
MEAN  Var1-Var20;
```

will compute the means of all the variables between the "locations" of Var1 and Var20 in the SAS data set.

BY list;

The BY keyword is used in conjunction with the TABLE or MEAN keyword. The analyses will be performed for each level of the variable(s) specified in the BY statement.

For instance,

```
TABLE Race;
 BY Gender;
```

will produce the weighted proportion of each Race category for each of the two levels of Gender.

If variable Agecat is age in 3 categories then

```
TABLE Race;
 BY Gender Agecat;
```

will produce weighted proportion of each Race category for each of the six combinations of Gender and Agecat.

CONTRAST specifications;

CONTRAST is used in conjunction with the MEAN keyword to compare or estimate linear combinations of cell means or proportions.

For example,

```
MEAN Income;
CONTRAST Race;
```

will produce all the pairwise comparisons of mean Income defined by Race. If Race has three categories then three pairwise comparisons will be produced**.**

```
MEAN Income;
CONTRAST Race*Gender;
```

will produce comparisons of Income means for all combinations of Race and Gender.

```
MEAN Income;
CONTRAST Race (.33 .33 .33);
```

will produce the average across the three categories of Race. (If Race has more than three levels then the above statement will produce an error message).

You can also specify complicated statements such as

```
MEAN Income;
CONTRAST Race(-1 0 1)*Gender(-1 1);
```

This can be useful in testing the significance of some preplanned contrasts in an ANOVA setting.

NOBS number;

NOBS indicates the number of observations to be used in the analysis. By default all observations in the data set will be used. You might use NOBS to subset a large data set while testing your setup file.

MDATA instruction;

The keyword MDATA followed by an instruction (STOP/IMPUTE/SKIP) indicates how missing data should be treated by the DESCRIBE module. If MDATA is not included in your setup, cases with missing data will be excluded from your analysis. This is equivalent to using the SKIP instruction.

```
MDATA STOP;
```

will cause the DESCRIBE module to stop if missing data are encountered in any analysis variables.

```
MDATA IMPUTE;
```

will impute missing data when used in conjunction with IMPUTE keywords.

PRINT instruction;

Indicates the printout desired. The options are STANDARD (default) and DETAILS.

When a DESCRIBE procedure includes the IMPUTE missing-data option (see MDATA above) the DETAILS keyword instructs *IVEware* to print the number and distribution of observed values, imputed values, and combined observed and imputed values for each variable.

When a DESCRIBE procedure includes multiple imputations the DETAILS keyword instructs *IVEware* to print estimates and statistics for each imputed data set as well as combined estimates and statistics across the imputed data sets.

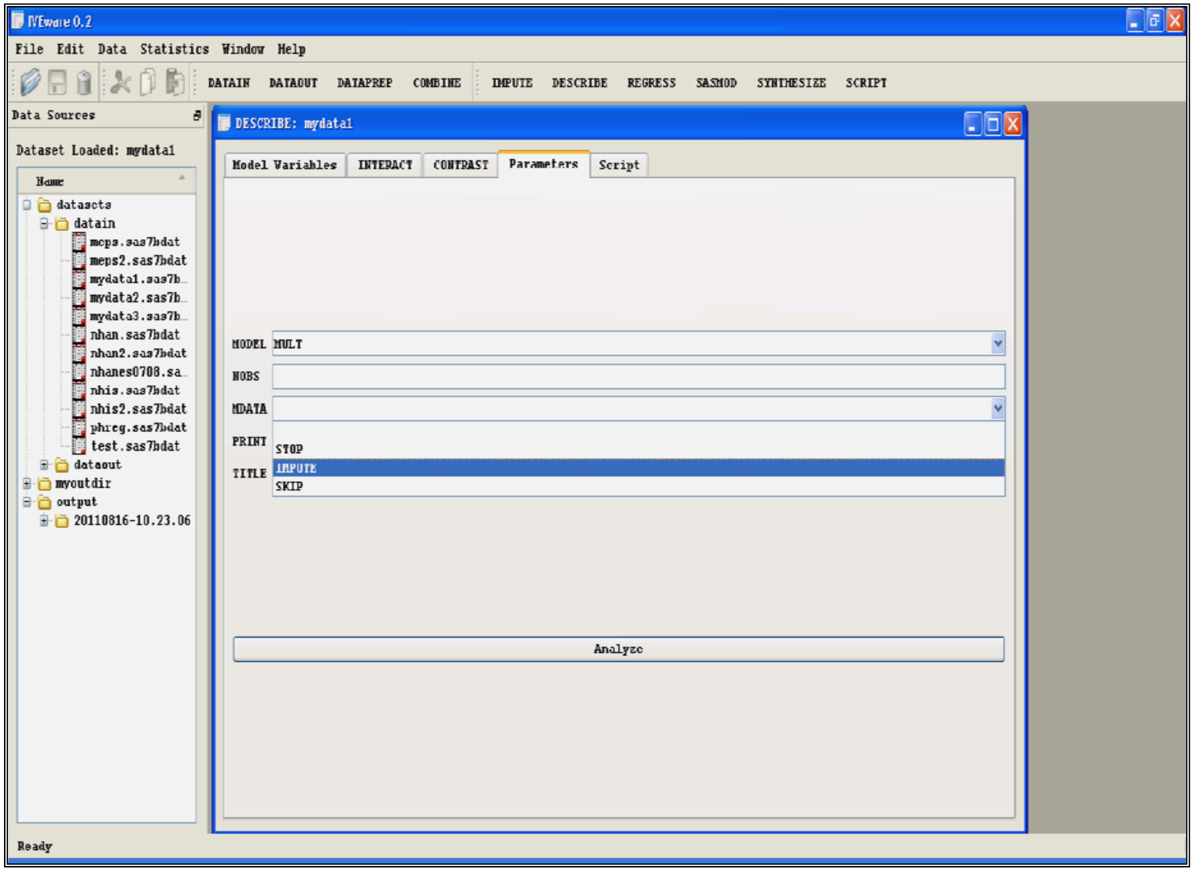
The STANDARD DESCRIBE printout does not include imputation results.

5. Describe and Impute Combined

Adding Impute to the Describe Procedure

If the data to be analyzed by DESCRIBE contain missing values you can, if you wish, incorporate IMPUTE into the DESCRIBE procedure. The data will be imputed prior to performing the DESCRIBE analysis. No imputed datasets will be outputted.

To incorporate IMPUTE, click on the MDATA parameter value box in the Describe Parameter window and select IMPUTE from the pull down menu.



MODEL MULT

NOBS

MDATA

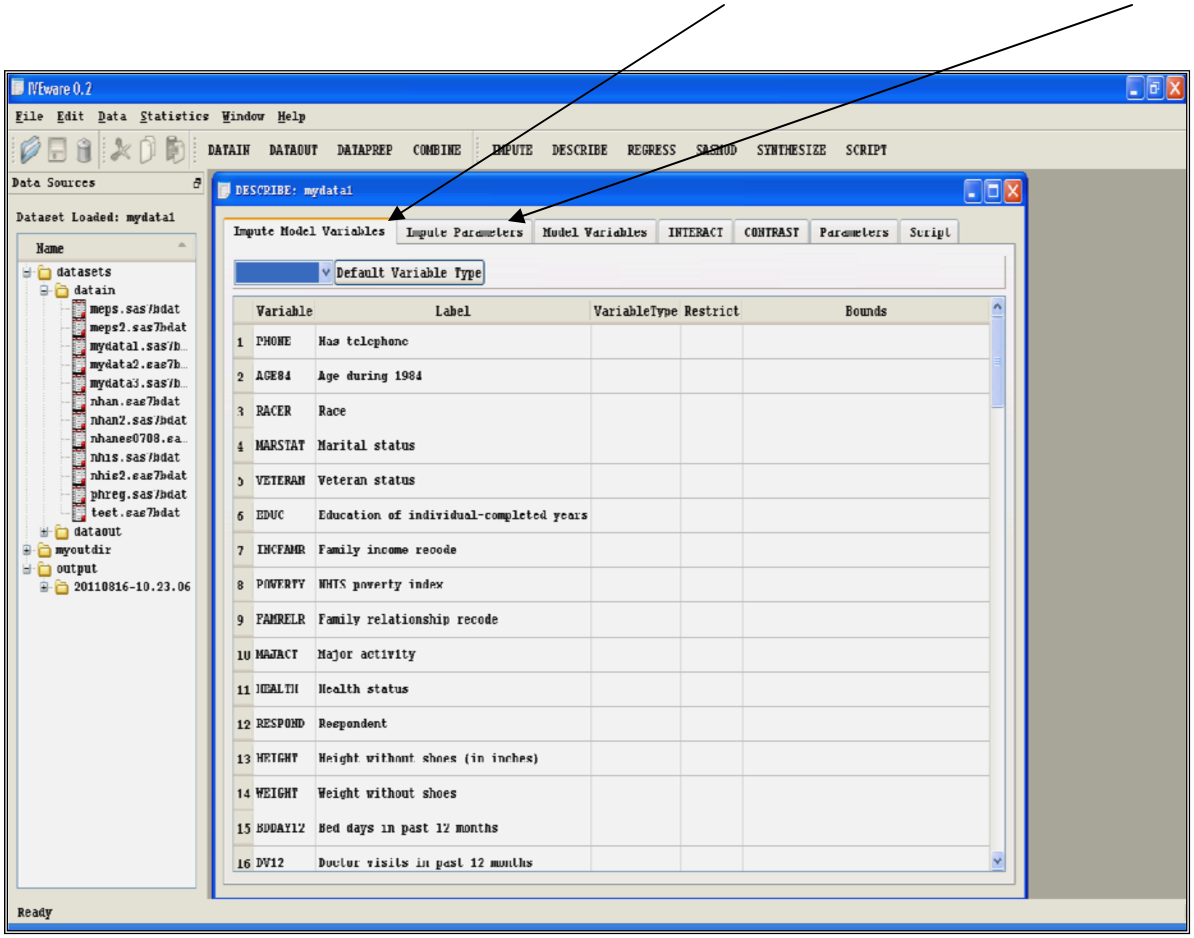
PRINT

TITLE

STOP
IMPUTE
SKIP

Adding Impute to the Describe Procedure

The DESCRIBE window will now show two new tabs—Impute Model Variables and Impute Parameters.

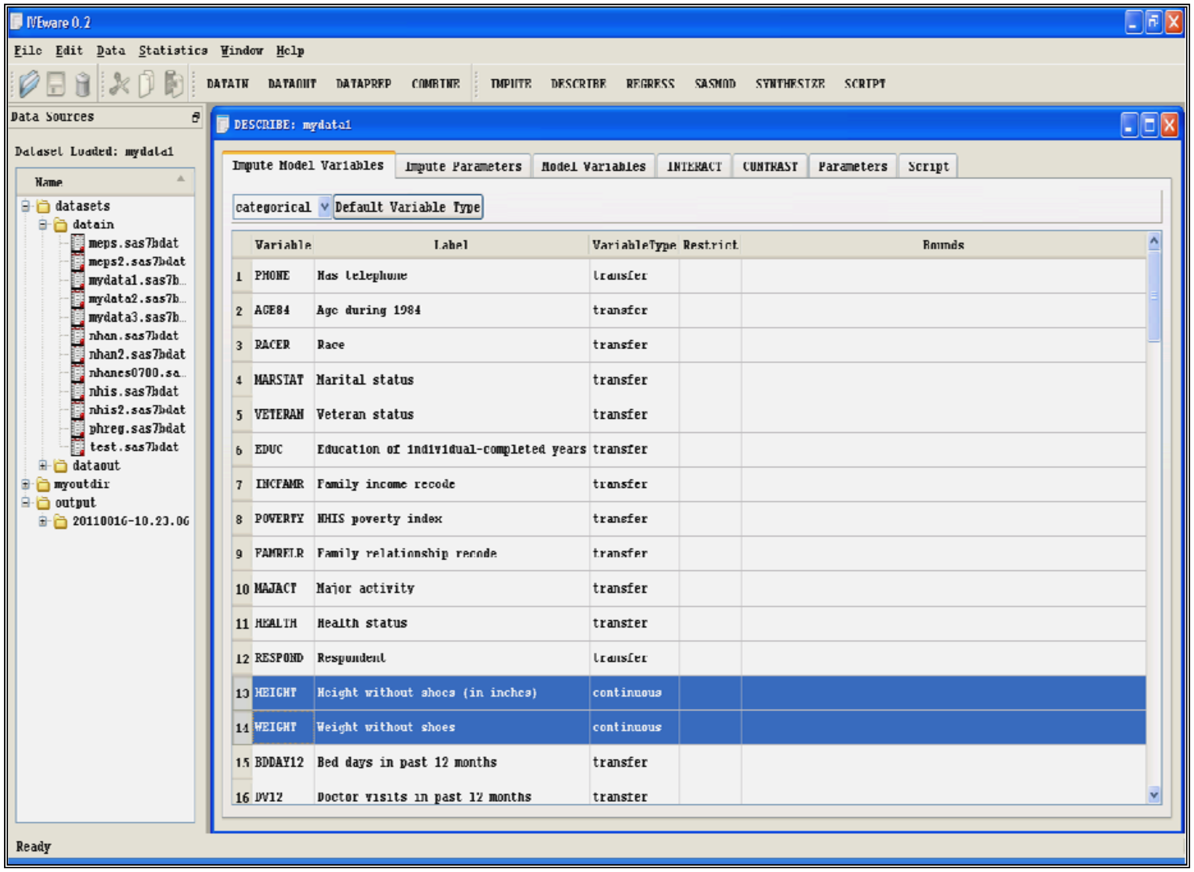


Dataset Loaded: mydata1

	Variable	Label	VariableType	Restrict	Bounds
1	PHONE	Has telephone			
2	AGE84	Age during 1984			
3	RACER	Race			
4	MARSTAT	Marital status			
5	VETERAN	Veteran status			
6	EDUC	Education of individual-completed years			
7	INCFAMR	Family income recode			
8	POVERTY	NHIS poverty index			
9	FAMRELR	Family relationship recode			
10	MAJACT	Major activity			
11	HEALTH	Health status			
12	RESPOND	Respondent			
13	HEIGHT	Height without shoes (in inches)			
14	WEIGHT	Weight without shoes			
15	BDDAY12	Bed days in past 12 months			
16	DV12	Doctor visits in past 12 months			

Declaration and Treatment of Imputation Variables

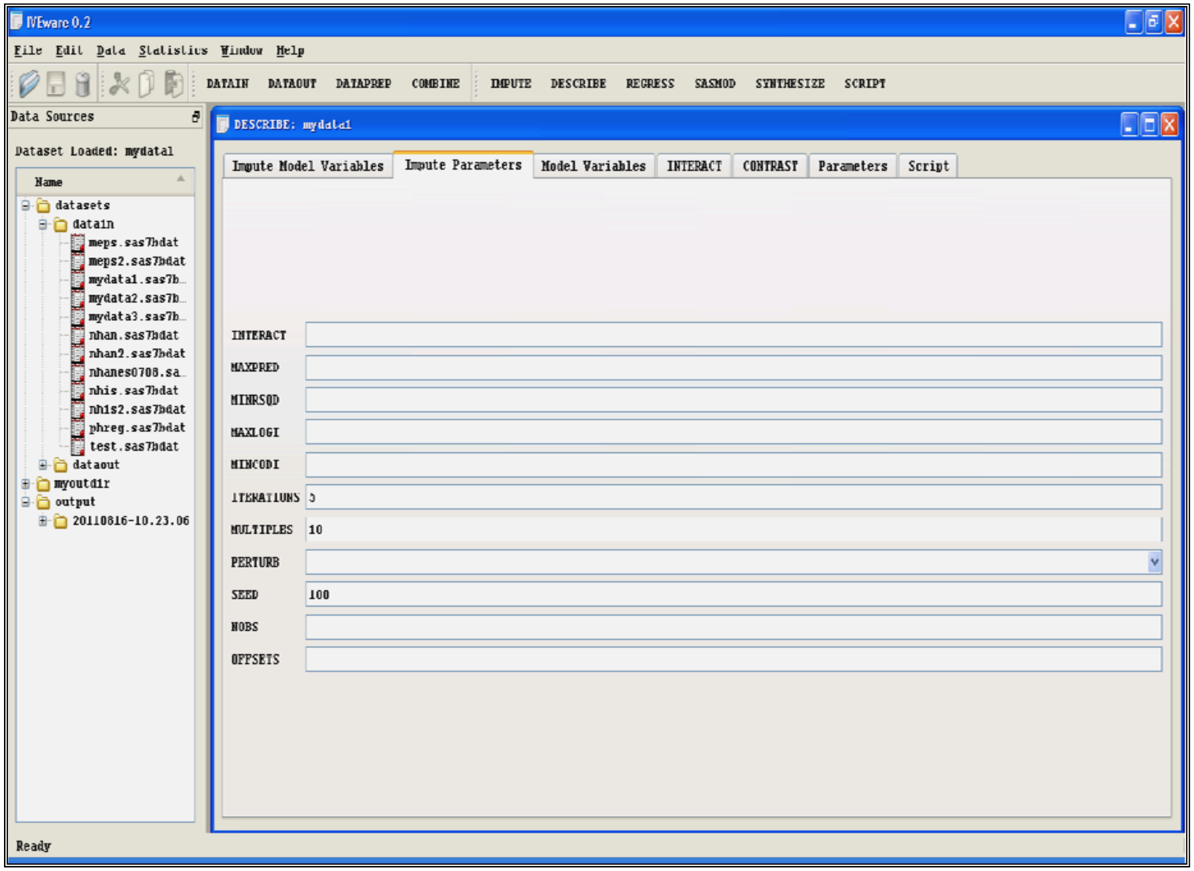
In the Impute Model Variables window you can declare variable type, restrict imputation variables, and apply value bounds to imputed variables. See The Impute Procedure (pages 11-27) for more details.



	Variable	Label	VariableType	Restrict	Bounds
1	PHONE	Has telephone	transfer		
2	AGE84	Age during 1984	transfer		
3	RACER	Race	transfer		
4	MARSTAT	Marital status	transfer		
5	VETERAN	Veteran status	transfer		
6	EDUC	Education of individual-completed years	transfer		
7	INCFAMR	Family income recode	transfer		
8	POVERTY	NHIS poverty index	transfer		
9	FAMRELR	Family relationship recode	transfer		
10	MAJACT	Major activity	transfer		
11	HEALTH	Health status	transfer		
12	RESPOND	Respondent	transfer		
13	HEIGHT	Height without shoes (in inches)	continuous		
14	WEIGHT	Weight without shoes	continuous		
15	BDDAY12	Bed days in past 12 months	transfer		
16	DV12	Doctor visits in past 12 months	transfer		

Imputation Parameters

Imputation parameters are entered in the Impute Parameter Window. In the example, 10 multiple imputations with 5 iterations are requested. The starting seed is 100. See The Impute Procedure (pages 11-27) for more details.



INTERACT

MAXPRED

MINRSQD

MAXLOGI

MINCODI

ITERATIONS 5

MULTIPLES 10

PERTURB

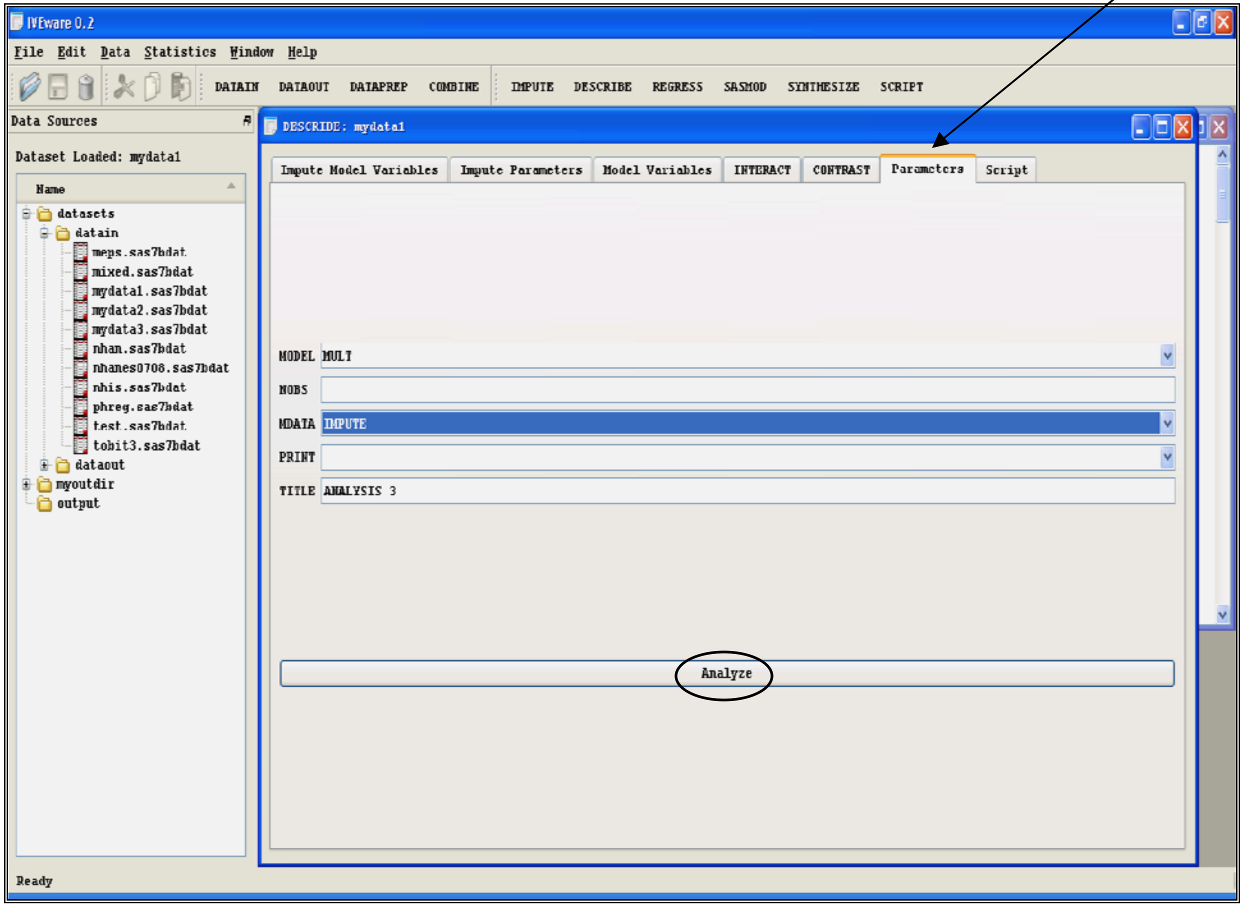
SEED 100

NOBS

OFFSETS

Describe Parameters

After you complete entering the imputation parameters, return to the original DESCRIBE Parameters window and click on the ANALYZE bar at the bottom to update the Script with the new imputation instructions.



Dataset Loaded: mydata1

MODEL MULT

NOBS

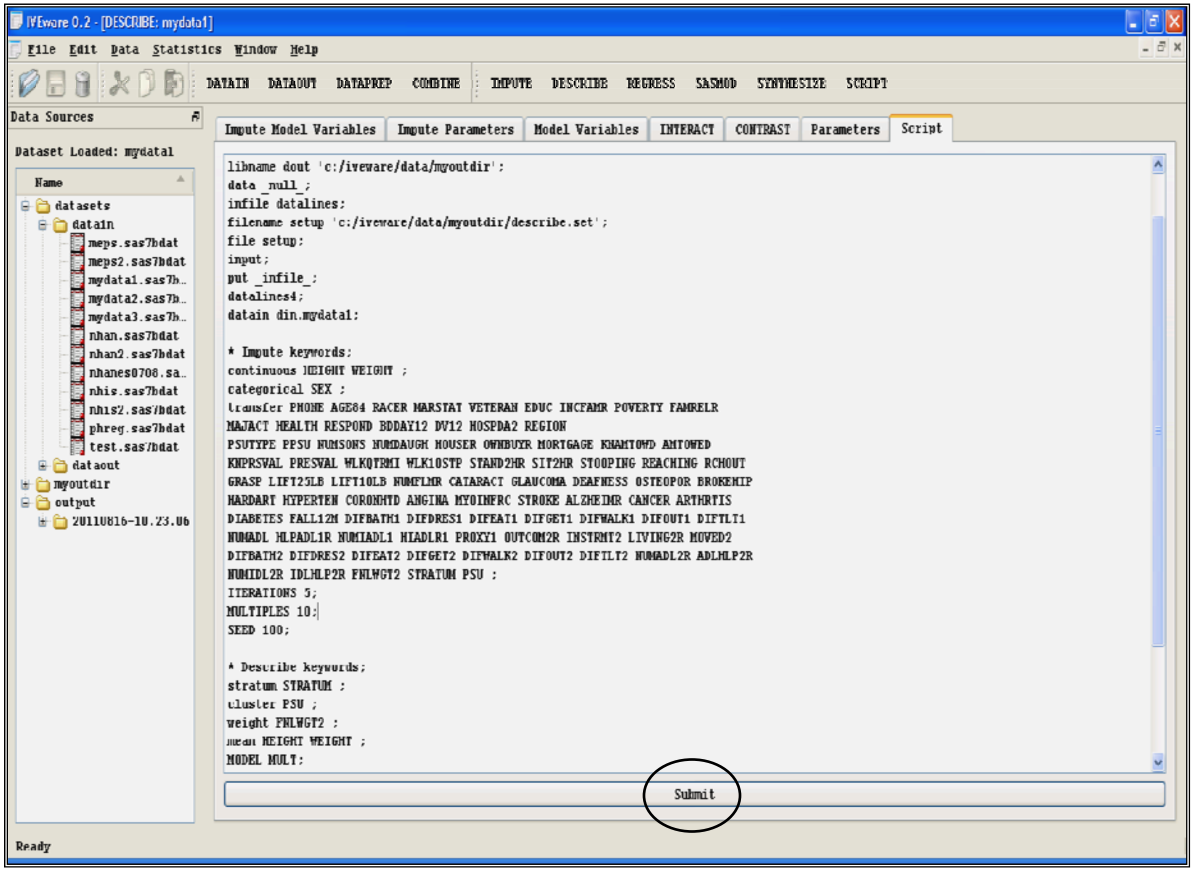
MDATA IMPUTE

PRINT

TITLE ANALYSIS 3

The Describe Script with Imputation Instructions

After clicking on the Script tab, the script will now include the imputation instructions. Click the SUBMIT bar at the bottom of the window and the combined DESCRIBE and IMPUTE script will be submitted for processing.

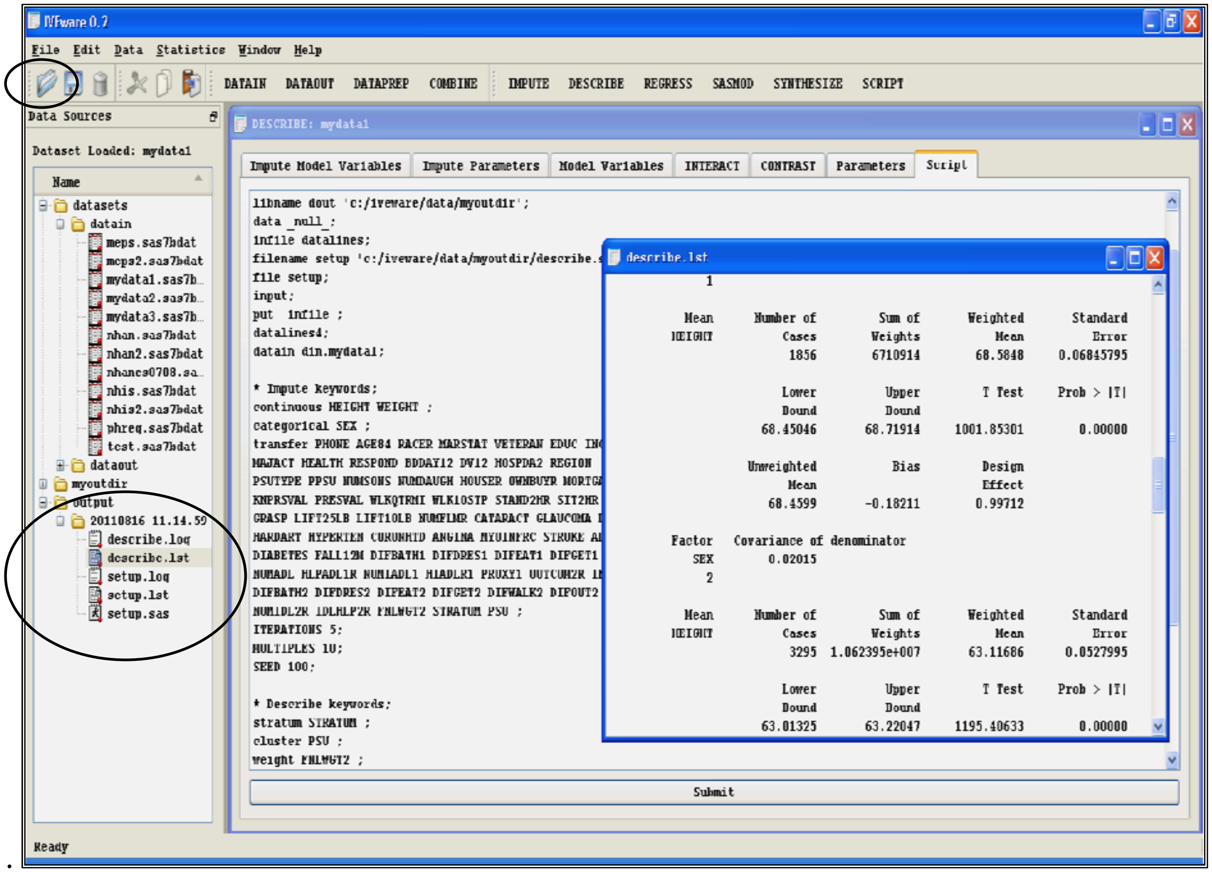


```
libname dout 'c:/iveware/data/myoutdir';
data _null_;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/describe.set';
file setup;
input;
put _infile_;
datalines4;
datain din.mydata1;

* Impute keywords;
continuous HEIGHT WEIGHT ;
categorical SEX ;
transfer PHONE AGE84 RACER MARSTAT VETERAN EDUC INCFAMR POVERTY FAMRELR
MAJACT HEALTH RESPOND BDDAY12 DV12 HOSPDA2 REGION
PSUTYPE PPSU NUMSONS NUMDAUGH HOUSER OWNBUYR MORTGAGE KHAMTOWD AMTOWED
KNPRSVAL PRESVAL WLKQTRMI WLK10STP STAND2HR SIT2HR STOOPING REACHING RCHOUT
GRASP LIFT25LB LIFT10LB NUMFLMR CATARACT GLAUCOMA DEAFNESS OSTEOPOR BROKEHIP
HARDART HYPERTEN CORONHTD ANGINA MYOINFRC STROKE ALZHEIMR CANCER ARTHRTIS
DIABETES FALL12M DIFBATH1 DIFDRES1 DIFEAT1 DIFGET1 DIFWALK1 DIFOUT1 DIFTLT1
NUMADL HLPADL1R NUMIADL1 HIADLR1 PROXY1 OUTCOM2R INSTRMT2 LIVING2R MOVED2
DIFBATH2 DIFDRES2 DIFEAT2 DIFGET2 DIFWALK2 DIFOUT2 DIFTLT2 NUMADL2R ADLHLP2R
NUMIDL2R IDLHLP2R FNLWGT2 STRATUM PSU ;
ITERATIONS 5;
MULTIPLES 10;
SEED 100;

* Describe keywords;
stratum STRATUM ;
cluster PSU ;
weight FNLWGT2 ;
mean HEIGHT WEIGHT ;
MODEL MULT;
```

Output Files

After the script submission is completed, the OUTPUT folder will contain a sub-folder labeled with the date and time of the script submission. Here you will find the results of the Describe procedure. Errors in the submitted script will be reported in the describe.log file. If the script was successful the Describe results can be found in describe.lst. No imputed datasets will be outputted.

To open an output file, highlight the file and click on the File icon at the top left of the window. In the example, describe.lst has been opened.



```
libname dout 'c:/iveware/data/myoutdir';
data _null_;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/describe.s
file setup;
input;
put  infile ;
datalines4;
datain din.mydata1;

* Impute keywords;
continuous HEIGHT WEIGHT ;
categorical SEX ;
transfer PHONE AGE84 RACER MARSTAT VETERAN EDUC INC
MAJACT HEALTH RESPOND BDDAY12 DV12 HOSPDA2 REGION
PSUTYPE PPSU NUMSONS NUMDAUGH HOUSER OWNBUYR MORTGA
KNPRSVAL PRESVAL WLKQTRMI WLK10STP STAND2HR SIT2HR
GRASP LIFT25LB LIFT10LB NUMFLMR CATARACT GLAUCOMA I
HARDART HYPERTEN CORONHTD ANGINA MYOINFRC STROKE AL
DIABETES FALL12M DIFBATH1 DIFDRES1 DIFEAT1 DIFGET1
NUMADL HLPADL1R NUMIADL1 HIADLR1 PROXY1 OUTCOM2R IN
DIFBATH2 DIFDRES2 DIFEAT2 DIFGET2 DIFWALK2 DIFOUT2
NUMIDL2R IDLHLP2R FNLWGT2 STRATUM PSU ;
ITERATIONS 5;
MULTIPLES 10;
SEED 100;

* Describe keywords;
stratum STRATUM ;
cluster PSU ;
weight FNLWGT2 ;
```

```
           1

        Mean     Number of        Sum of      Weighted      Standard
      HEIGHT         Cases       Weights          Mean         Error
                      1856       6710914       68.5848    0.06845795

                     Lower         Upper        T Test    Prob > |T|
                     Bound         Bound
                  68.45046      68.71914    1001.85301       0.00000

                Unweighted          Bias        Design
                      Mean                      Effect
                   68.4599      -0.18211       0.99712

      Factor   Covariance of denominator
         SEX       0.02015
           2

        Mean     Number of        Sum of      Weighted      Standard
      HEIGHT         Cases       Weights          Mean         Error
                      3295  1.062395e+007     63.11686     0.0527995

                     Lower         Upper        T Test    Prob > |T|
                     Bound         Bound
                  63.01325      63.22047    1195.40633       0.00000
```

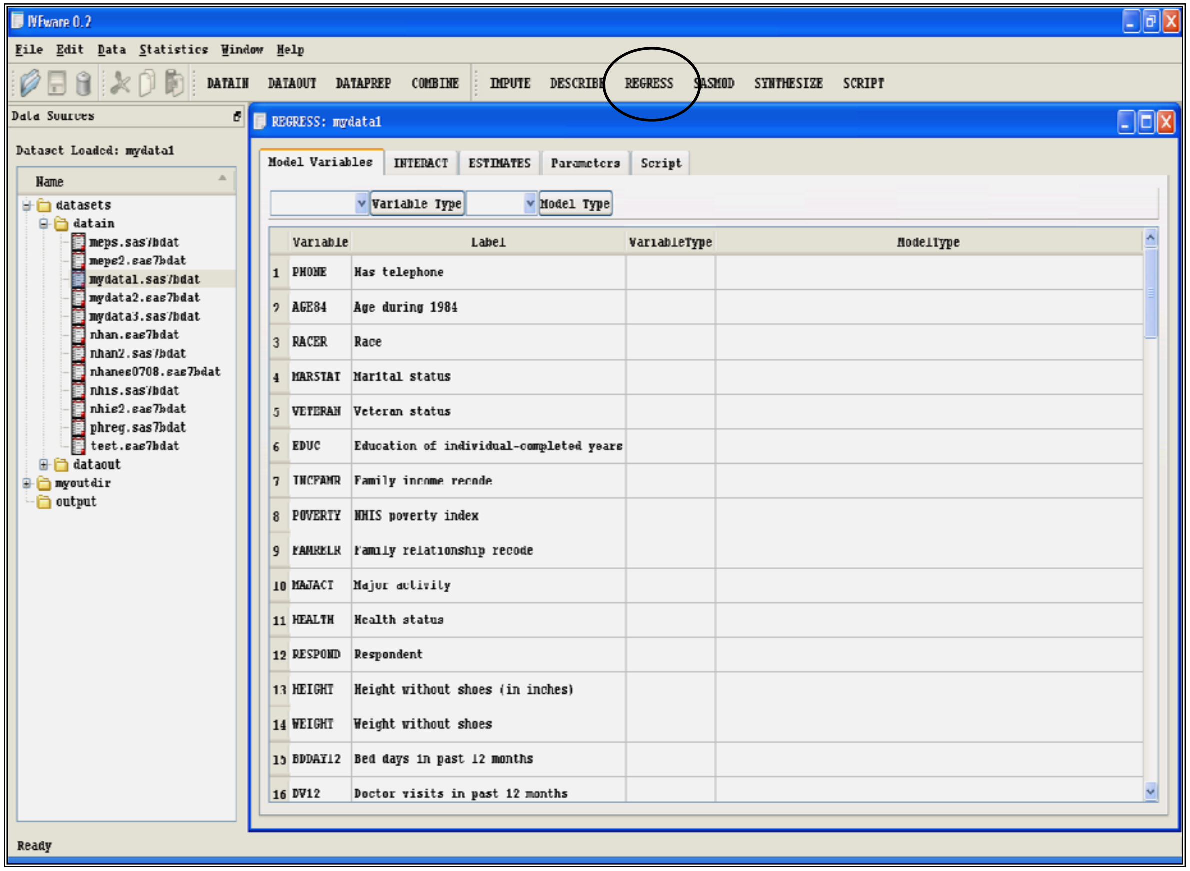
.

6. The Regress Procedure

Selecting the Regress Procedure

After loading a dataset, click on the REGRESS button. This will open the Model Variables window, listing all variables in the loaded dataset.

In this example, MYDATA1 was previously loaded. The dataset must be loaded before selecting an *IVEware* procedure (see page 7).

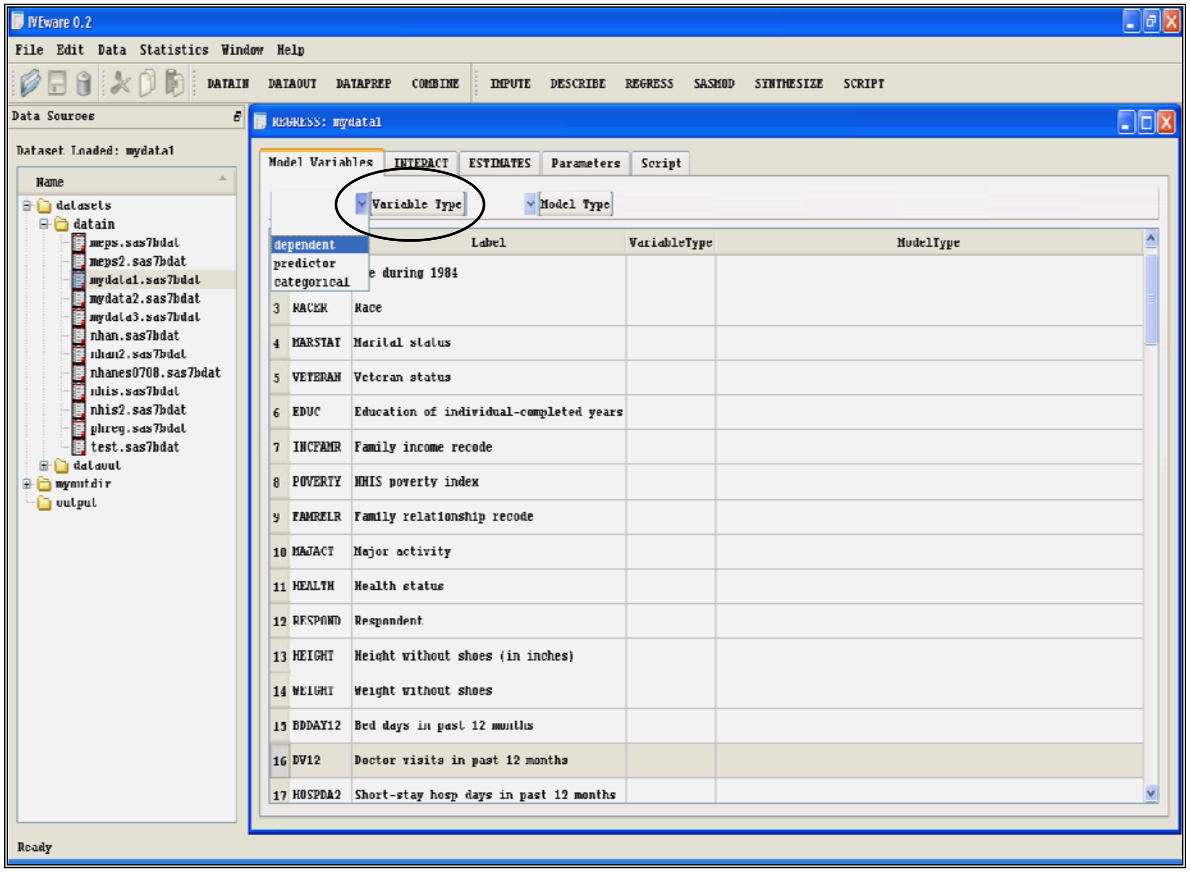


Dataset Loaded: mydata1

	Variable	Label	VariableType	ModelType
1	PHONE	Has telephone		
2	AGE84	Age during 1984		
3	RACER	Race		
4	MARSTAT	Marital status		
5	VETERAN	Veteran status		
6	EDUC	Education of individual-completed years		
7	INCFAMR	Family income recode		
8	POVERTY	NHIS poverty index		
9	FAMRELR	Family relationship recode		
10	MAJACT	Major activity		
11	HEALTH	Health status		
12	RESPOND	Respondent		
13	HEIGHT	Height without shoes (in inches)		
14	WEIGHT	Weight without shoes		
15	BDDAY12	Bed days in past 12 months		
16	DV12	Doctor visits in past 12 months		

Variable Selection and Model Type

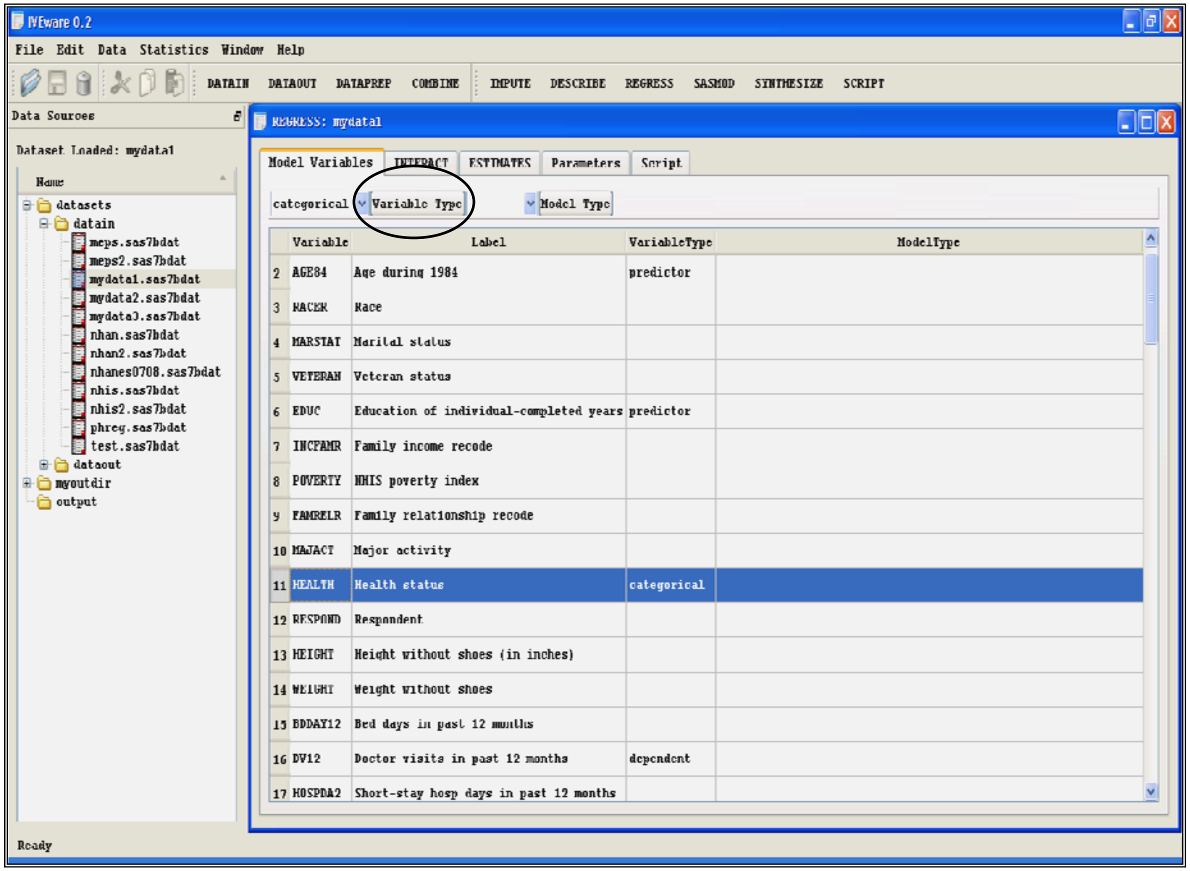
To select a variable (s) to be included in your regression model, highlight the variable name, click the arrow on the dialogue box to the left of the VARIABLE TYPE button and choose the variable type—Dependent, Predictor or Categorical. Then click on the VARIABLE TYPE button. In the example, DV12 is selected as the model's dependent variable.



		Label	VariableType	ModelType
		e during 1984		
3	RACER	Race		
4	MARSTAT	Marital status		
5	VETERAN	Veteran status		
6	EDUC	Education of individual-completed years		
7	INCFAMR	Family income recode		
8	POVERTY	NHIS poverty index		
9	FAMRELR	Family relationship recode		
10	MAJACT	Major activity		
11	HEALTH	Health status		
12	RESPOND	Respondent		
13	HEIGHT	Height without shoes (in inches)		
14	WEIGHT	Weight without shoes		
15	BDDAY12	Bed days in past 12 months		
16	DV12	Doctor visits in past 12 months		
17	HOSPDA2	Short-stay hosp days in past 12 months		

Categorical Predictors

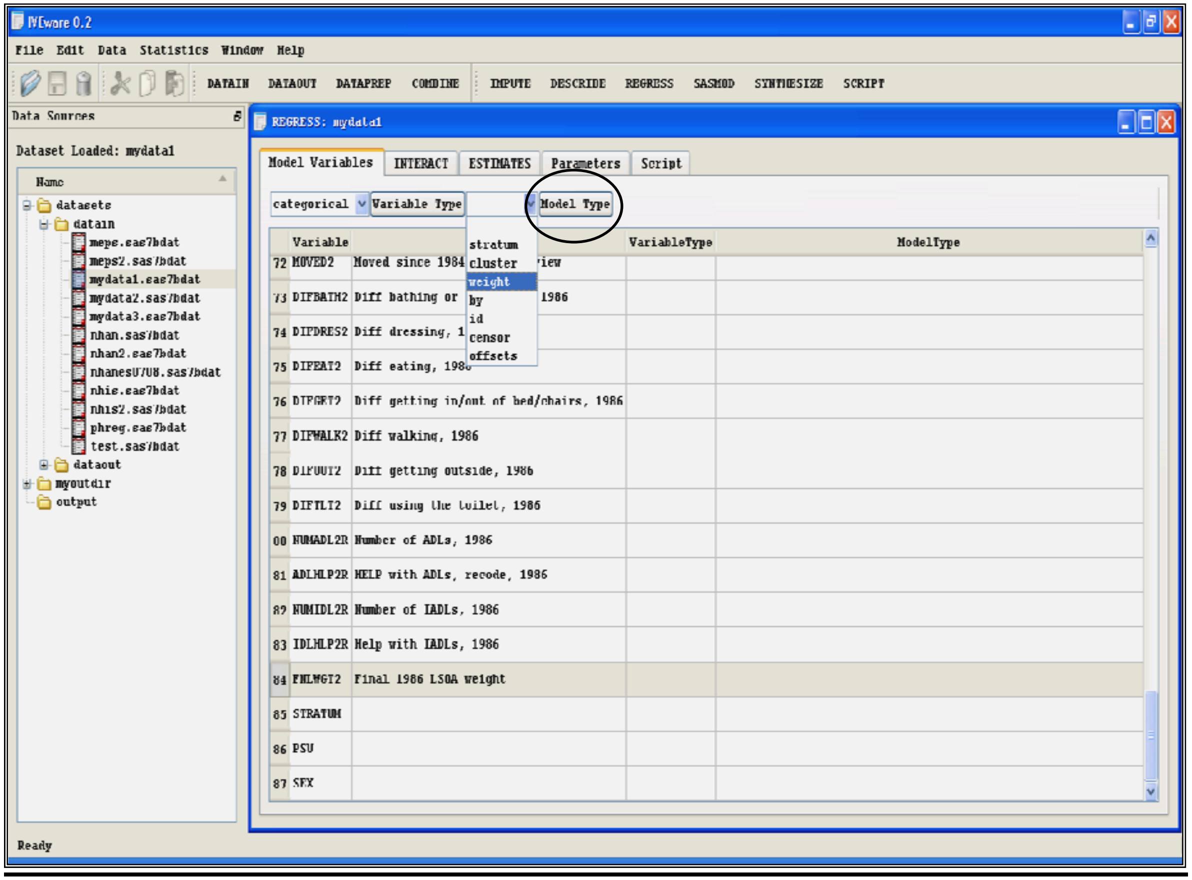
Categorical predictors need to be explicitly defined so that *IVEware* can create dummy variables. Highlight the variable name, click on CATEGORICAL in the dialogue box, and then click on the VARIABLE TYPE button. In the example, the variable HEALTH was defined as a categorical predictor.



	Variable	Label	VariableType	ModelType
2	AGE84	Age during 1984	predictor	
3	RACER	Race		
4	MARSTAT	Marital status		
5	VETERAN	Veteran status		
6	EDUC	Education of individual-completed years	predictor	
7	INCFAMR	Family income recode		
8	POVERTY	NHIS poverty index		
9	FAMRELR	Family relationship recode		
10	MAJACT	Major activity		
11	HEALTH	Health status	categorical	
12	RESPOND	Respondent		
13	HEIGHT	Height without shoes (in inches)		
14	WEIGHT	Weight without shoes		
15	BDDAY12	Bed days in past 12 months		
16	DV12	Doctor visits in past 12 months	dependent	
17	HOSPDA2	Short-stay hosp days in past 12 months		

Defining Model Type

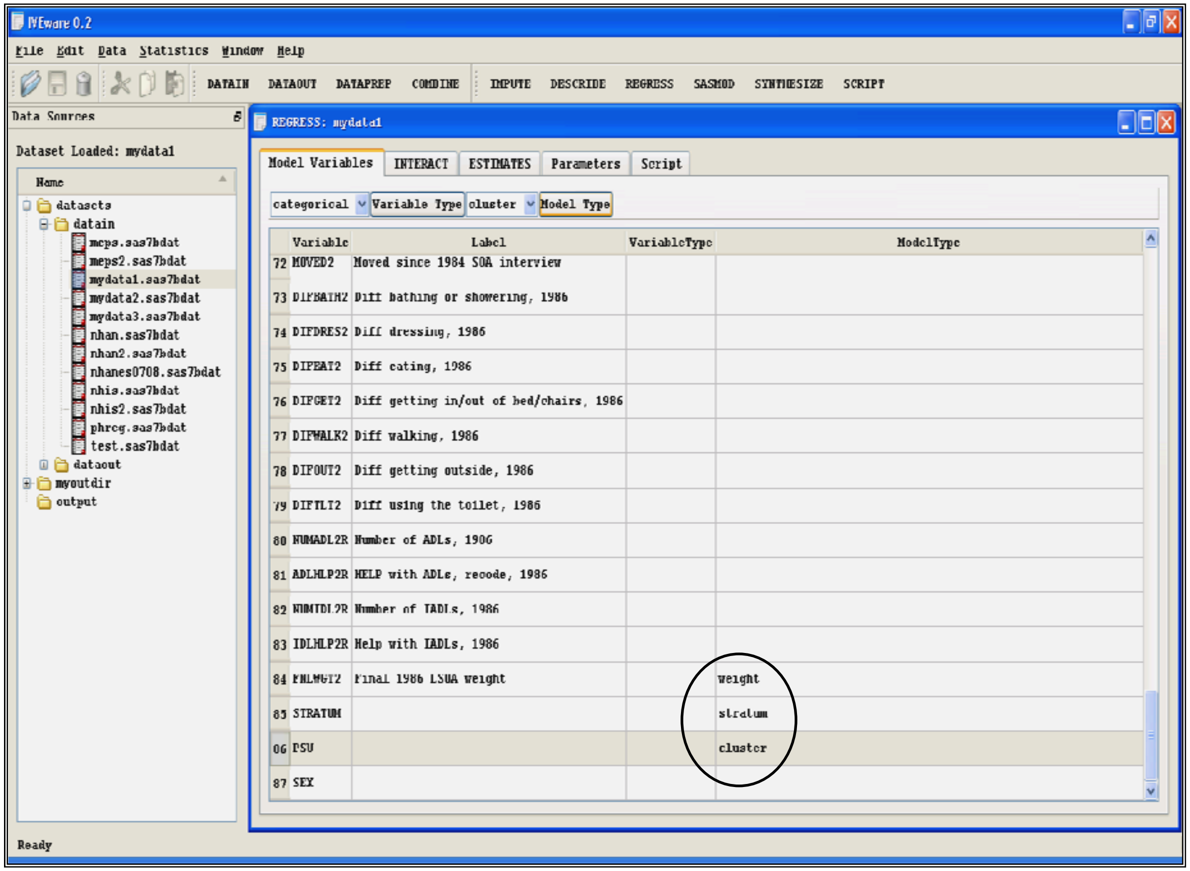
To incorporate survey design variables in your analysis, highlight the design variable in the Model Variables window. Click the arrow on the dialog box to the left of the MODEL TYPE button and select the appropriate design feature—stratum, cluster or weight. Then click on the MODEL TYPE button.



	Variable		VariableType	ModelType
72	MOVED2	Moved since 1984 iew		
73	DIFBATH2	Diff bathing or 1986		
74	DIFDRES2	Diff dressing, 1		
75	DIFEAT2	Diff eating, 198		
76	DIFGET2	Diff getting in/out of bed/chairs, 1986		
77	DIFWALK2	Diff walking, 1986		
78	DIFOUT2	Diff getting outside, 1986		
79	DIFTLT2	Diff using the toilet, 1986		
80	NUMADL2R	Number of ADLs, 1986		
81	ADLHLP2R	HELP with ADLs, recode, 1986		
82	NUMIDL2R	Number of IADLs, 1986		
83	IDLHLP2R	Help with IADLs, 1986		
84	FNLWGT2	Final 1986 LSOA weight		
85	STRATUM			
86	PSU			
87	SEX			

Defining Model Type

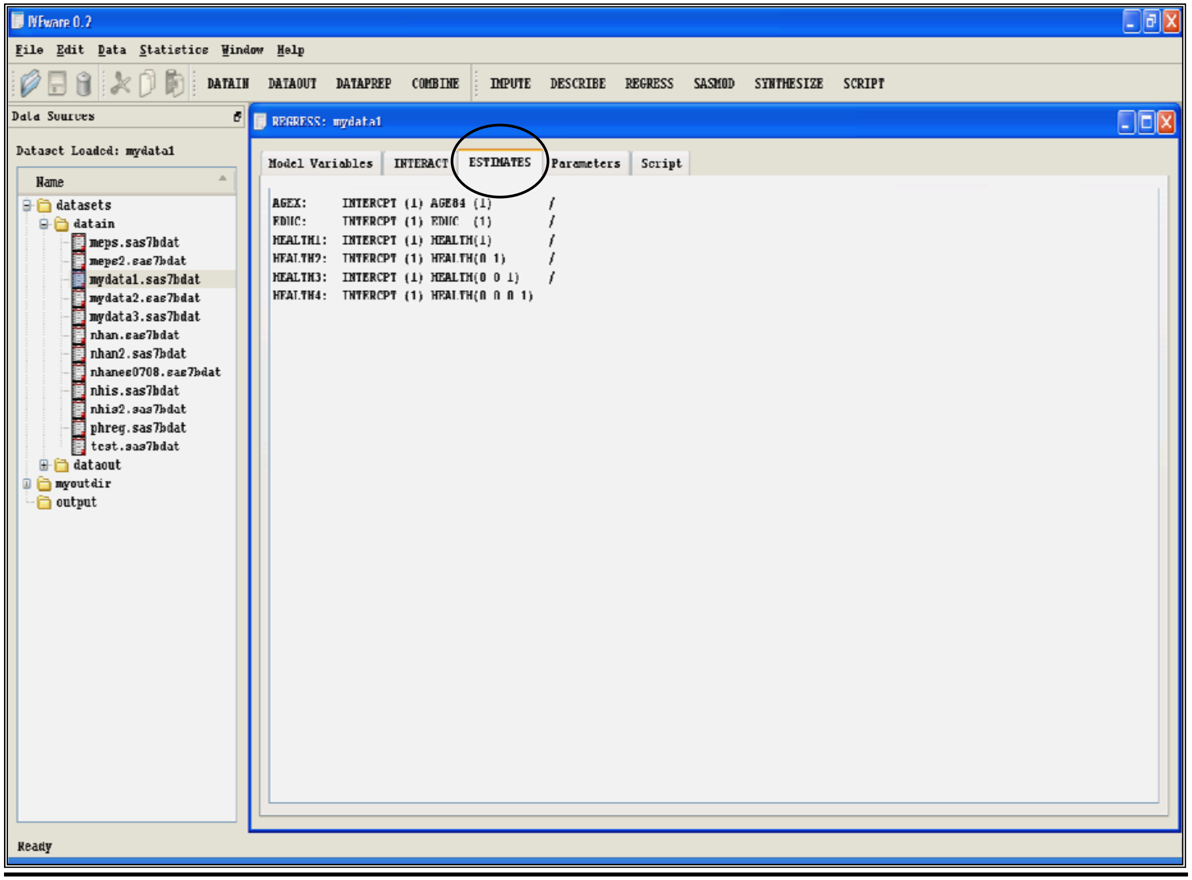
After MODEL TYPE is clicked, the selected design feature appears in the column headed “Model Type.” In the example, FNLWGT2 was assigned as the weight variable, STRATUM as the stratum variable, and PSU as the cluster variable.



	Variable	Label	VariableType	ModelType
72	MOVED2	Moved since 1984 SOA interview		
73	DIFBATH2	Diff bathing or showering, 1986		
74	DIFDRES2	Diff dressing, 1986		
75	DIFEAT2	Diff eating, 1986		
76	DIFGET2	Diff getting in/out of bed/chairs, 1986		
77	DIFWALK2	Diff walking, 1986		
78	DIFOUT2	Diff getting outside, 1986		
79	DIFTLT2	Diff using the toilet, 1986		
80	NUMADL2R	Number of ADLs, 1986		
81	ADLHLP2R	HELP with ADLs, recode, 1986		
82	NUMIDL2R	Number of IADLs, 1986		
83	IDLHLP2R	Help with IADLs, 1986		
84	FNLWGT2	Final 1986 LSOA weight		weight
85	STRATUM			stratum
86	PSU			cluster
87	SEX			

The Regress Estimate Window

To estimate values of the dependent variable for specific predictors, click the Estimates tab and directly type the estimate statements into the opened window.



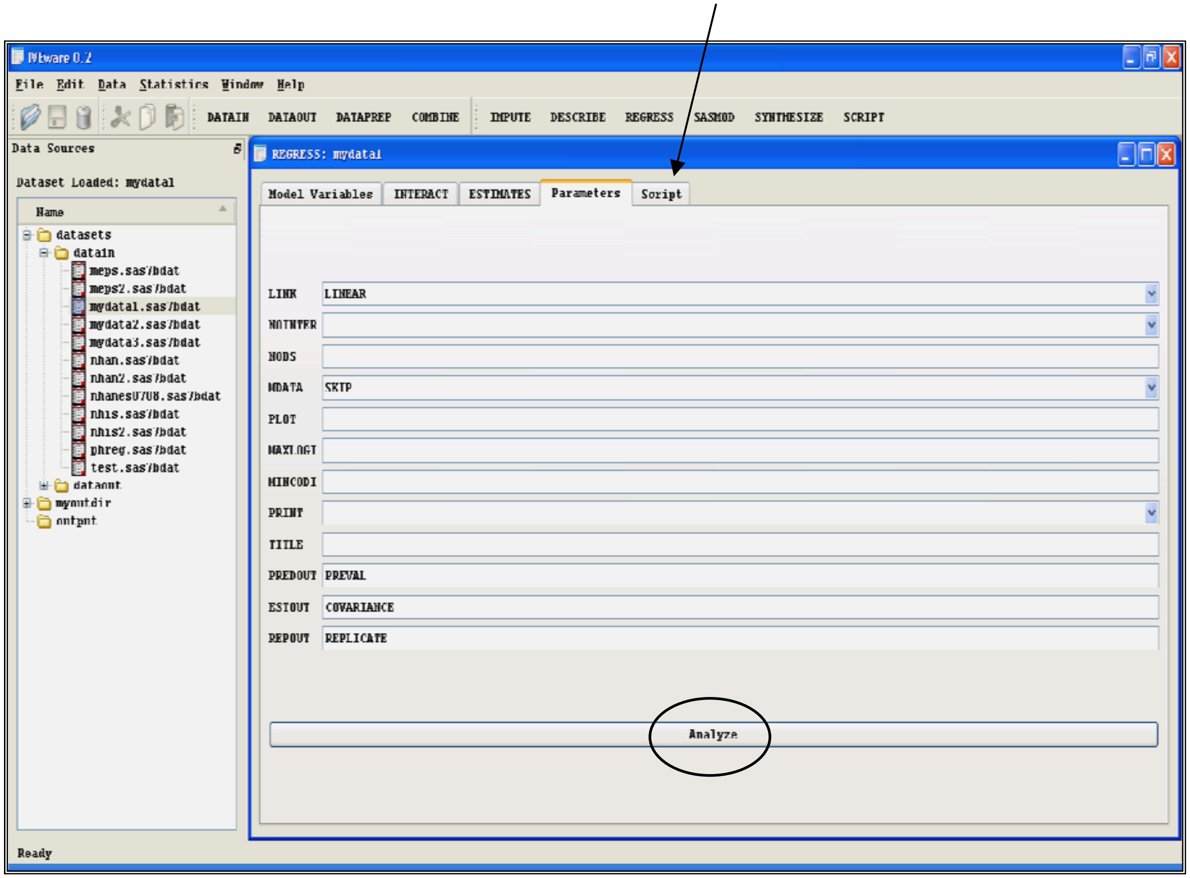
```
AGEX:     INTERCPT (1) AGE84 (1)        /
EDUC:     INTERCPT (1) EDUC  (1)        /
HEALTH1:  INTERCPT (1) HEALTH(1)        /
HEALTH2:  INTERCPT (1) HEALTH(0 1)      /
HEALTH3:  INTERCPT (1) HEALTH(0 0 1)    /
HEALTH4:  INTERCPT (1) HEALTH(0 0 0 1)
```

Regress Parameters

Clicking on the Parameters tab opens the Regress parameter window. Enter the appropriate parameter value in the space to the right of the parameter name.

In the example, the defaults for LINK (LINEAR) and MDATA (SKIP) have been retained. The values for PREDOUT, ESTOUT, and REPOUT are the output file names that will include predicted values, estimates and covariances, and estimates for sampling replicates respectively.

After you complete entering all parameters, click on the ANALYZE bar at the bottom of the window. This will generate the Regress script. The script can be viewed by clicking on the SCRIPT tab.



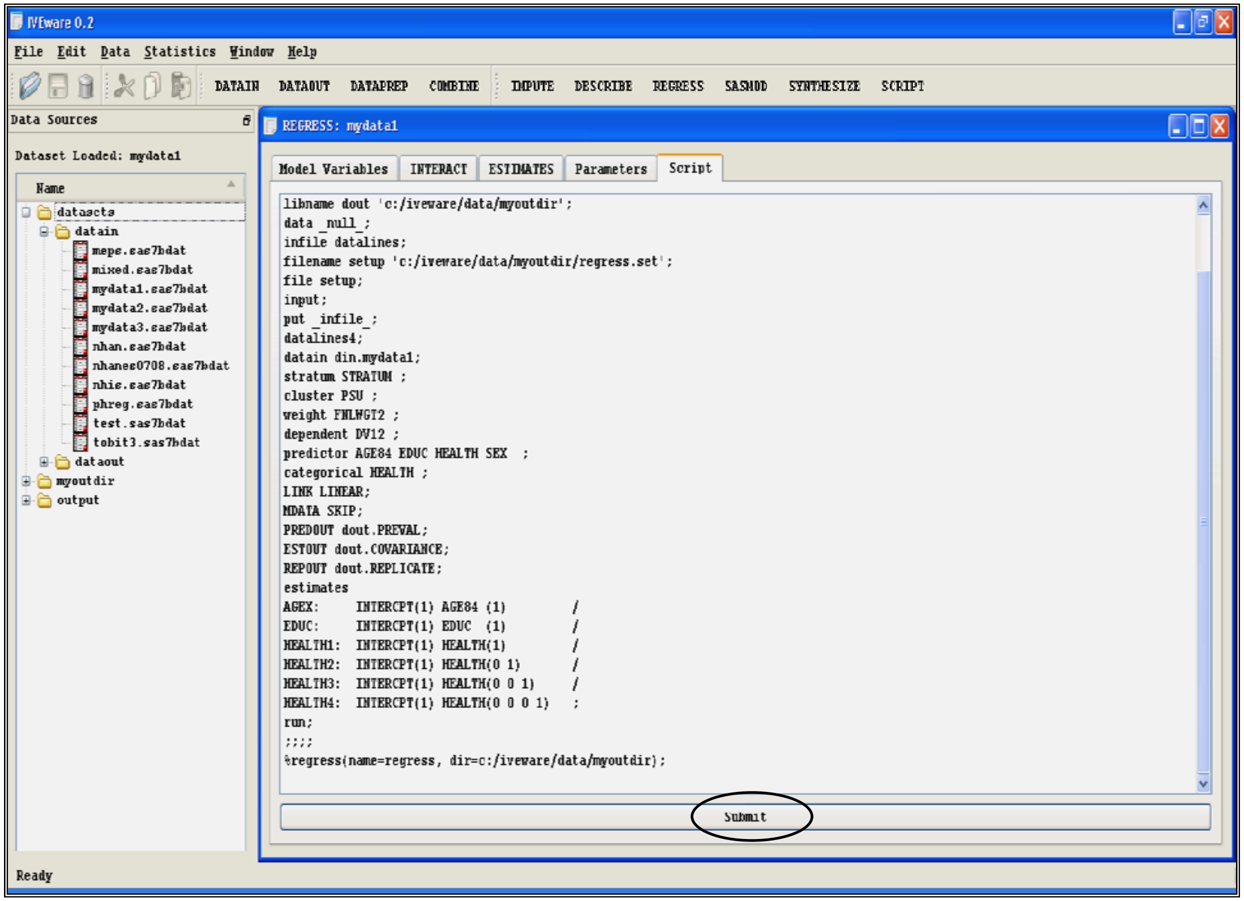
LINK LINEAR
NOINTER
NOBS
MDATA SKIP
PLOT
MAXLOGT
MINCODI
PRINT
TITLE
PREDOUT PREVAL
ESTOUT COVARIANCE
REPOUT REPLICATE

.

The Regress Script Window

The Regress script contains instructions entered in the Model Variables, Estimate, and Parameter windows. The script can be modified by returning to those windows and entering changes. After entering changes you must once again click on the ANALYZE bar in Parameter window to update the script. You can also type script changes directly in the script window.

Submit your script for processing by clicking on the SUBMIT bar at the bottom of the Script window.



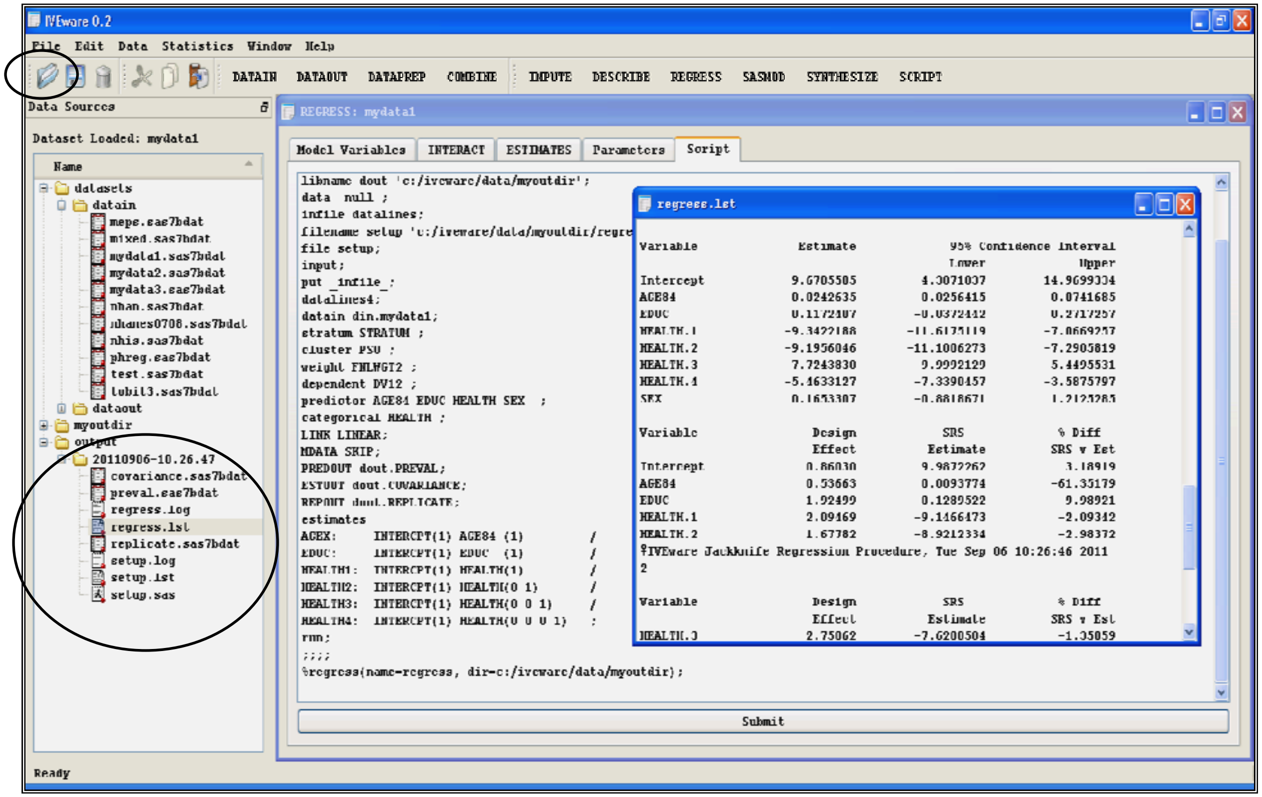
```
libname dout 'c:/iveware/data/myoutdir';
data _null_;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/regress.set';
file setup;
input;
put _infile_;
datalines4;
datain din.mydata1;
stratum STRATUM ;
cluster PSU ;
weight FNLWGT2 ;
dependent DV12 ;
predictor AGE84 EDUC HEALTH SEX  ;
categorical HEALTH ;
LINK LINEAR;
MDATA SKIP;
PREDOUT dout.PREVAL;
ESTOUT dout.COVARIANCE;
REPOUT dout.REPLICATE;
estimates
AGEX:     INTERCPT(1) AGE84 (1)        /
EDUC:     INTERCPT(1) EDUC  (1)        /
HEALTH1:  INTERCPT(1) HEALTH(1)        /
HEALTH2:  INTERCPT(1) HEALTH(0 1)      /
HEALTH3:  INTERCPT(1) HEALTH(0 0 1)    /
HEALTH4:  INTERCPT(1) HEALTH(0 0 0 1)  ;
run;
;;;;
%regress(name=regress, dir=c:/iveware/data/myoutdir);
```

Output Files

After the script submission is completed, the OUTPUT folder will contain a sub-folder labeled with the date and time of the script submission. Here you will find the results of the Regress procedure. Errors in the submitted script will be reported in the regress.log file. If the script was successful the Regress results can be found in regress.lst.

To open an output file, highlight the file and click on the File icon at the top left of the window. In the example, regress.lst has been opened.

The diagnostic files are also included in the OUTPUT folder. The example script, calling for three diagnostic files, outputted PREDVAL, COVARIANCE and REPLICATE. The contents of these files can be view in SAS outside the *IVEware* environment.



```
libname dout 'c:/iveware/data/myoutdir';
data  null ;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/regre
file setup;
input;
put _infile_;
datalines4;
datain din.mydata1;
stratum STRATUM ;
cluster PSU ;
weight FNLWGT2 ;
dependent DV12 ;
predictor AGE84 EDUC HEALTH SEX   ;
categorical HEALTH ;
LINK LINEAR;
MDATA SKIP;
PREDOUT dout.PREVAL;
ESTOUT dout.COVARIANCE;
REPOUT dout.REPLICATE;
estimates
AGEX:      INTERCPT(1) AGE84 (1)          /
EDUC:      INTERCPT(1) EDUC  (1)          /
HEALTH1:   INTERCPT(1) HEALTH(1)          /
HEALTH2:   INTERCPT(1) HEALTH(0 1)        /
HEALTH3:   INTERCPT(1) HEALTH(0 0 1)      /
HEALTH4:   INTERCPT(1) HEALTH(0 0 0 1)    ;
run;
;;;;
%regress(name=regress, dir=c:/iveware/data/myoutdir);
```

```
Variable              Estimate          95% Confidence Interval
                                           Lower          Upper
Intercept            9.6705505        4.3071037     14.9699334
AGE84                0.0242635        0.0256415      0.0741685
EDUC                 0.1172107       -0.0372442      0.2717257
HEALTH.1            -9.3422188      -11.6175119     -7.0669257
HEALTH.2            -9.1956046      -11.1006273     -7.2905819
HEALTH.3             7.7243830        9.9992129      5.4495531
HEALTH.4            -5.4633127       -7.3390457     -3.5875797
SEX                  0.1653307       -0.8818671      1.2125285

Variable                Design           SRS          % Diff
                        Effect        Estimate      SRS v Est
Intercept              0.86030       9.9872262        3.18919
AGE84                  0.53663       0.0093774      -61.35179
EDUC                   1.92499       0.1289522        9.98921
HEALTH.1               2.09469      -9.1466473       -2.09342
HEALTH.2               1.67782      -8.9212334       -2.98372
♀IVEware Jackknife Regression Procedure, Tue Sep 06 10:26:46 2011
2

Variable                Design           SRS          % Diff
                        Effect        Estimate      SRS v Est
HEALTH.3               2.75062      -7.6200504       -1.35059
```

Glossary of Regress Commands

Required or Standard Statements

DEPENDENT variable name;

This specifies the name of the dependent variable in the regression model. Dependent variables are assumed to be continuous unless the CATEGORICAL keyword is included in the setup file (see below).

PREDICTOR variable list;

This specifies the right hand side of the regression model. Predictor variables are assumed to be continuous unless they are defined as CATEGORICAL (see below). Interaction terms can be specified by using the "*" notation.

For example,

```
PREDICTOR Income Age Income*Age;
```

LINK model;

LINK defines the type of regression model to be fit. Specify **Linear** for fitting a multiple linear regression model, **Logistic** for fitting a logistic (binary) or generalized logistic (polytomous) regression model, **Log** for fitting a Poisson regression model for a count variable, **Tobit** for fitting a tobit model or **Phreg** for fitting Proportional Hazards model (Cox model).

RUN;

This should be the last statement in the setup file.

Optional Statements

STRATUM variable name;

variable name is the name of the stratum variable. No missing values are allowed for the stratum variable**.**

CLUSTER variable name;

variable name is the Primary Sampling Unit (PSU) or Sampling Error Computing Unit (SECU) variable. No missing values are allowed for the cluster variable.

WEIGHT variable name;

variable name is the survey weight variable. Survey weights are usually the product of selection, nonresponse adjustment and poststratification weights. No missing values are allowed for the weight variable.

NOTES:

1. If the STRATUM, CLUSTER and WEIGHT variable are not specified then a simple random sample analysis will be performed.
2. If a design based analysis involves only a WEIGHT variable and no STRATUM or CLUSTER variable then a pseudo stratification variable and a pseudo cluster variable should be used. When using pseudo variables, all observations in the data set should have the same value for the pseudo STRATUM variable (e.g., 1), while each observation should have a unique value on the pseudo CLUSTER variable (e.g., observation ID number or SAS system variable _N_). The pseudo variables should be created in a SAS data step prior to performing the analysis. See the Appendix for an example data step creating pseudo stratification and pseudo cluster variables. The inclusion of pseudo variables will increase the time REGRESS needs for analysis.

CENSOR variable name (number);

variable name is a censoring variable, and number is the code indicating censoring. If the number is omitted then by default 1 will be considered as the code indicating censoring. The Censor statement is required if the **LINK** is specified as **Phreg.**

For example,

```
LINK          Phreg;
DEPENDENT     Survivaltime;
CENSOR        Died (0);
```

In this example, respondent survival time is predicted censoring on whether or not the respondent died.

CATEGORICAL variable list;

declares that the listed variables are to be treated as categorical. If a variable with *k* categories is listed on the CATEGORICAL and PREDICTOR statement then *k*-1 predictors (dummies) will be included in the regression model. The category with the highest code value will be the reference category.

OFFSETS count variable(offset variable);

This statement is used to specify an offsets variable when fitting a Poisson regression model.

For example,

```
OFFSETS  Injuries(Years);
```

will fit a model predicting the number for injuries occurring per year.

ID variable name;

Specifies the variable to be used as the unique subject identifier. This allows for linking the PREDOUT file (see below) created by the REGRESS module to other files.

NOINTER;

This keyword will fit regression models without the intercept term.

ESTIMATES label: specification;

This is useful for estimating values of the dependent variable for a specific set of covariates or testing hypotheses involving the estimated regression coefficients.

For example, suppose that the following regression model is fit:

$$Y = \beta_o + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

If we are interested in predicting Y when $x_1 = 1, x_2 = 2$ and $x_3 = 0$ then we can obtain the predicted value and the 95% confidence interval by using the following statement:

```
ESTIMATES   Mylabel : Intercept (1) X1(1)  X2(2);
```

Several estimates can be requested by separating them with “/ ” .

```
ESTIMATES   Mylabel1: Intercept (1)  X1(1) X2(2) /
            Mylabel2: Intercept (1)  X1(1) X3(1) /
            Mylabel3: Intercept (1)  X2(1) X3(1) ;
```

BY variable list;

The regression analysis will be performed for each level of the variable(s) specified in the BY statement.

For instance,

```
BY Gender;
```

will produce regressions for each of the two levels of Gender.

If the variable Agecat is age in 3 categories then

```
BY Gender Agecat;
```

will produce regressions for each of the six combinations of Gender and Agecat.

NOBS number;

NOBS indicates the number of observations to be used in the analysis. By default all observations in the data set will be used. You might use NOBS to subset a large data set while testing your setup file.

MDATA instruction;

The keyword MDATA followed by an instruction (STOP/IMPUTE/SKIP) indicates how missing data should be treated by the REGRESS module. If MDATA is not included in your setup cases with missing data for model variables will be excluded from the regression. This is equivalent to using the SKIP instruction.

```
MDATA Stop;
```

will cause the REGRESS module to stop if missing data are encountered for variables in the regression model.

```
MDATA Impute;
```

will impute missing data when used in conjunction with IMPUTE keywords. See section 5.4 for more on combining IMPUTE and REGRESS functions.

PLOT libname.filename;

This keyword creates a series of diagnostic plots including residual, leverage, influence and normal probability plots. The plots will be stored in the file name specified after the PLOT keyword.

MAXLOGI number;

Specifies the maximum number of iterative algorithms to be performed in a logistic or multilogit regression model. The default is 50. This is useful if the Newton-Raphson algorithm used in producing maximum likelihood estimates does not converge after 50 iterations. This applies to the convergence criterion for the logistic, polytomous and Poisson regression models. You can check whether you have a non-convergence problem by inspecting the log file (e.g.,mysetup.log).

MINCODI decimal;

Specifies the minimum proportional change in any regression coefficient to continue the logistic regression iteration process. This applies to the convergence criterion for the logistic, polytomous and Poisson regression models.

PRINT instruction;

Indicates the printout desired. The options are STANDARD (default) and DETAILS.

When a REGRESS procedure includes the IMPUTE missing-data option (see MDATA above) the DETAILS keyword instructs *IVEware* to print the number and distribution of observed values, imputed values, and combined observed and imputed values for each variable.

When a REGRESS procedure includes multiple imputations the DETAILS keyword instructs *IVEware* to print the estimates and statistics for each multiple as well as for the combined multiples.

The STANDARD REGRESS printout does not include imputation results.

```
line;
```

OUTPUT FILES

REGRESS can output SAS data sets for diagnostic purposes and for further analysis.

PREDOUT

outputs a file containing predicted values, residuals and leverage information.

For example,

```
PREDOUT Mylib.Predval;
```

creates an output file called **Predval** containing the predicted values, their standard errors and 95% confidence intervals. If an ID statement is included in the setup an ID variable is also included in the data set.

ESTOUT

Outputs a file containing estimates and their variances-covariances. (ESTOUT produces a SAS estimate type data set. See the SAS user manual for more details).

For example,

```
ESTOUT Mylib.Est;
```

creates an output file called **Est** containing the estimates and variances-covariances.

REPOUT

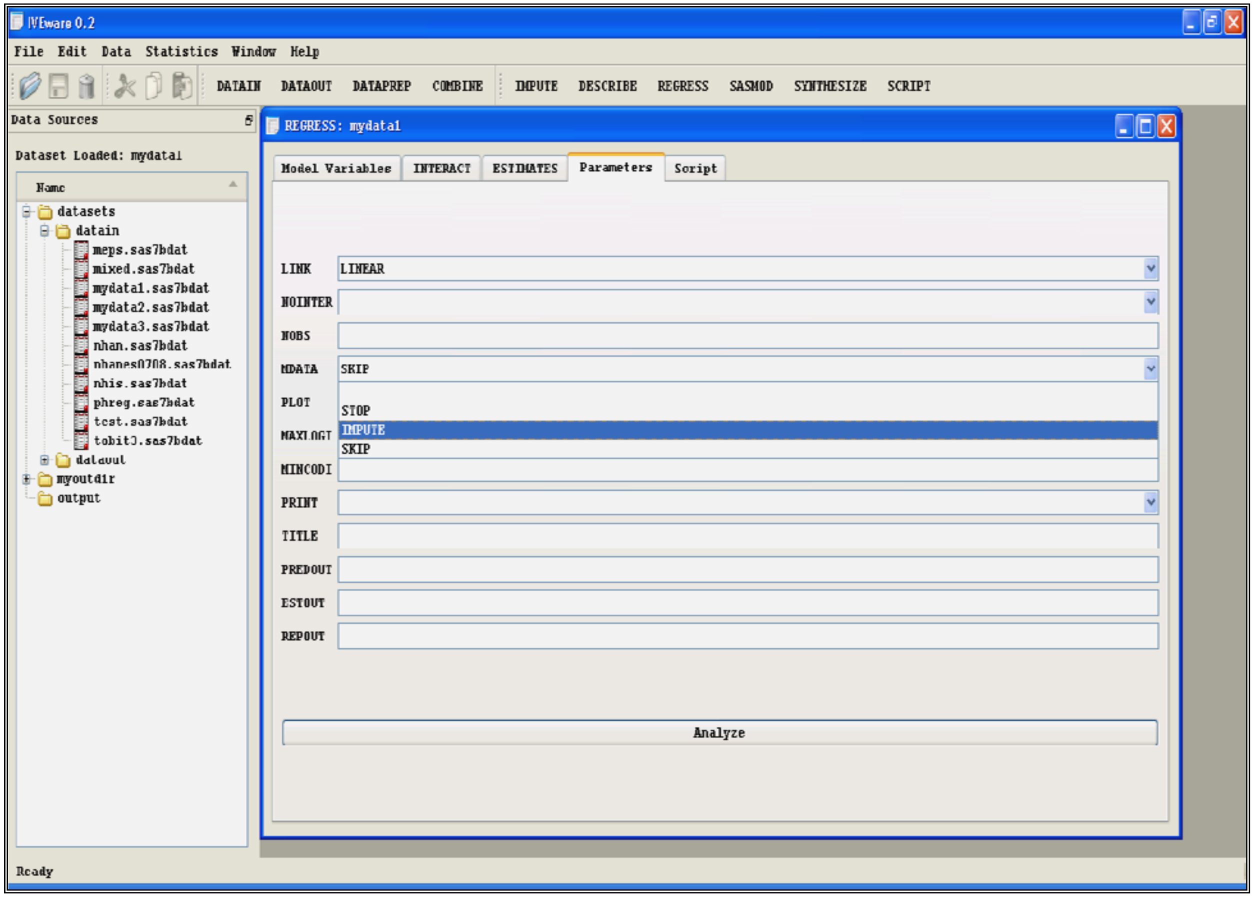
Outputs a file containing estimates for each replicate. Estimated regression coefficients are provided for each combination of STRATUM, CLUSTER and BY variable.

7. Regress and Impute Combined

Adding Impute to the Regress Procedure

If the data to be analyzed by REGRESS contain missing values you can, if you wish, incorporate IMPUTE into the REGRESS procedure. The data will be imputed prior to performing the REGRESS analysis. No imputed datasets will be outputted.

To incorporate IMPUTE, click on the MDATA parameter value box in the Regress Parameter window and select IMPUTE from the pull down menu.



Dataset Loaded: mydata1

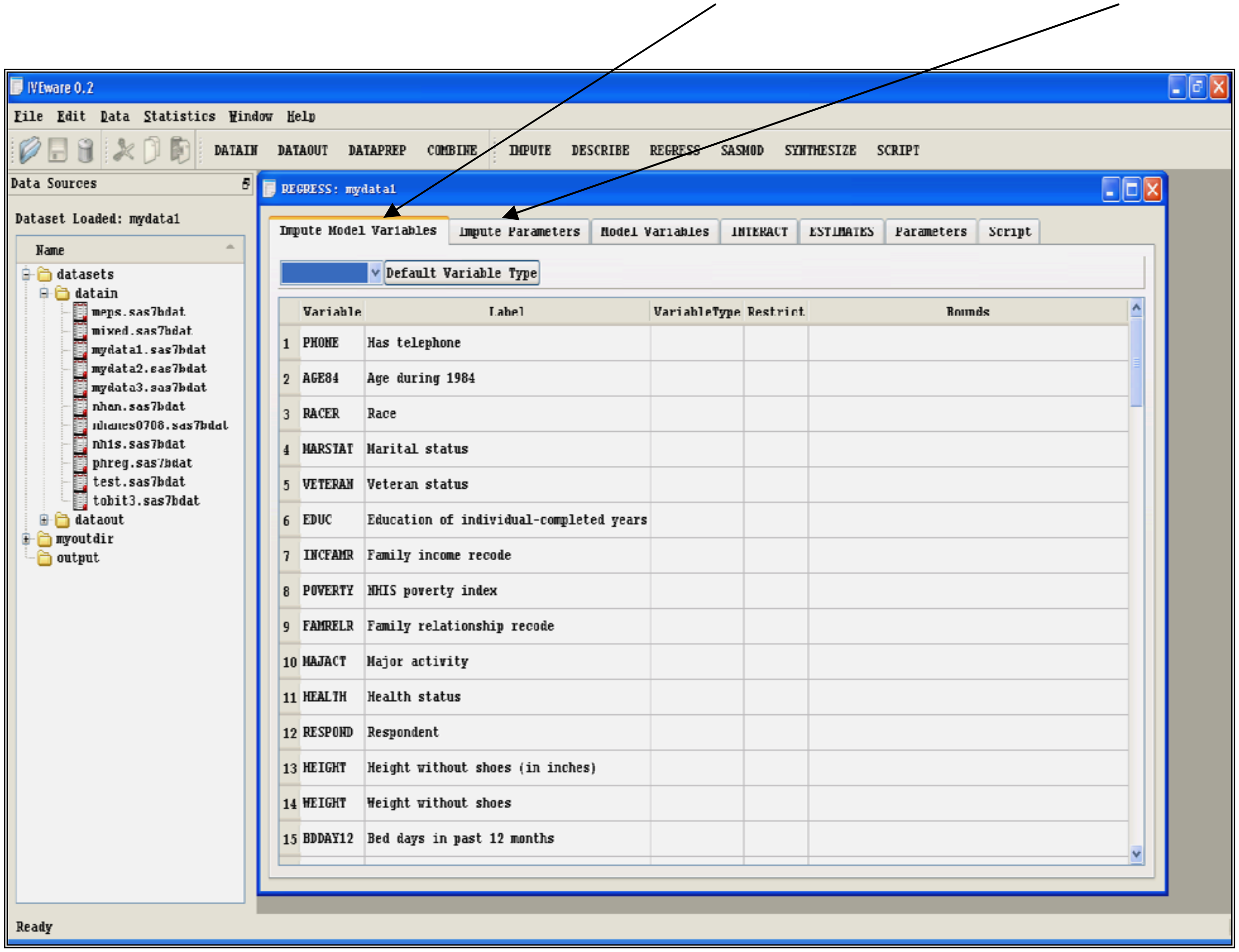
REGRESS: mydata1

LINK LINEAR

MDATA SKIP

Adding Impute to the Regress Procedure

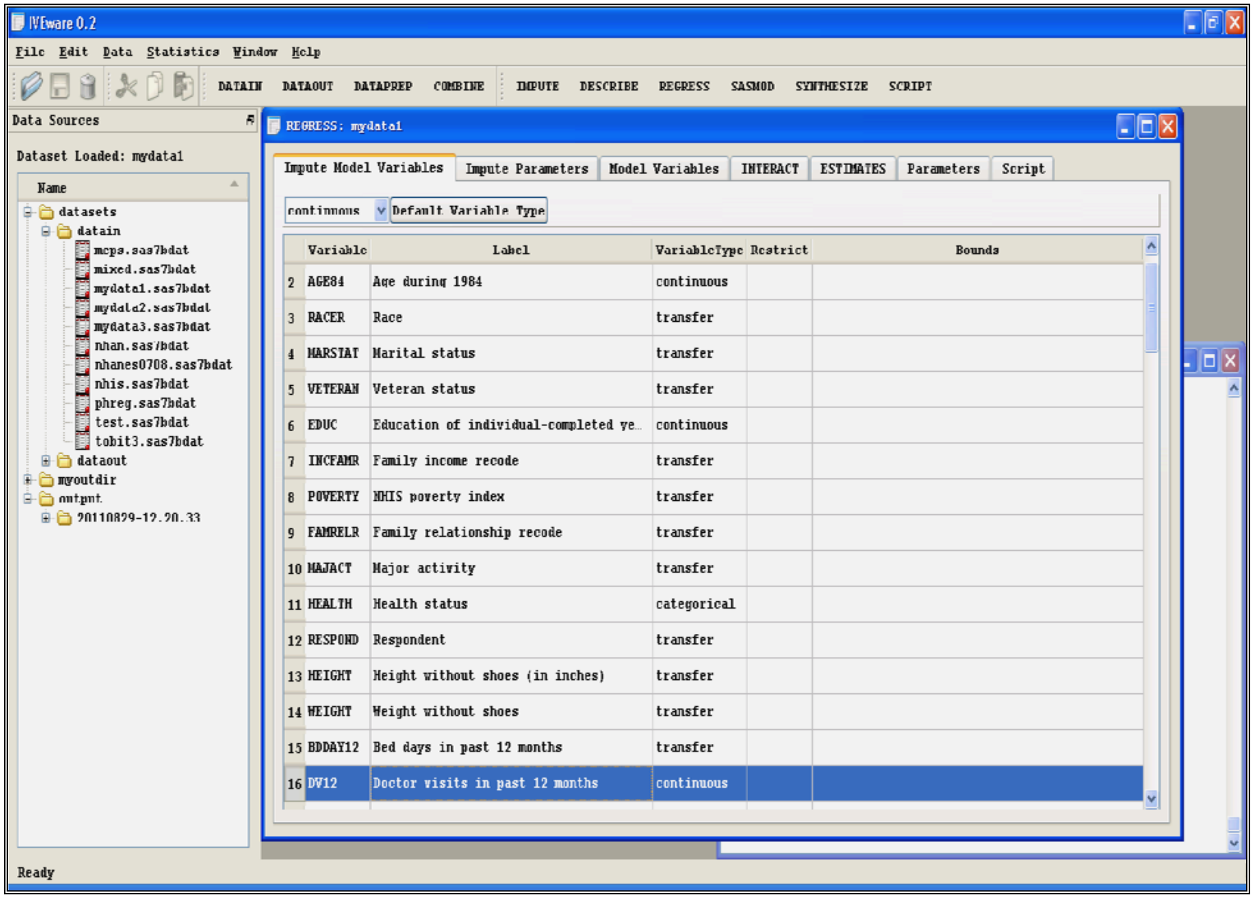
The IMPUTE window will now show two new tabs—Impute Model Variables and Impute Parameters.



	Variable	Label	VariableType	Restrict	Bounds
1	PHONE	Has telephone			
2	AGE84	Age during 1984			
3	RACER	Race			
4	MARSTAT	Marital status			
5	VETERAN	Veteran status			
6	EDUC	Education of individual-completed years			
7	INCFAMR	Family income recode			
8	POVERTY	NHIS poverty index			
9	FAMRELR	Family relationship recode			
10	MAJACT	Major activity			
11	HEALTH	Health status			
12	RESPOND	Respondent			
13	HEIGHT	Height without shoes (in inches)			
14	WEIGHT	Weight without shoes			
15	BDDAY12	Bed days in past 12 months			

Declaration and Treatment of Imputation Variables

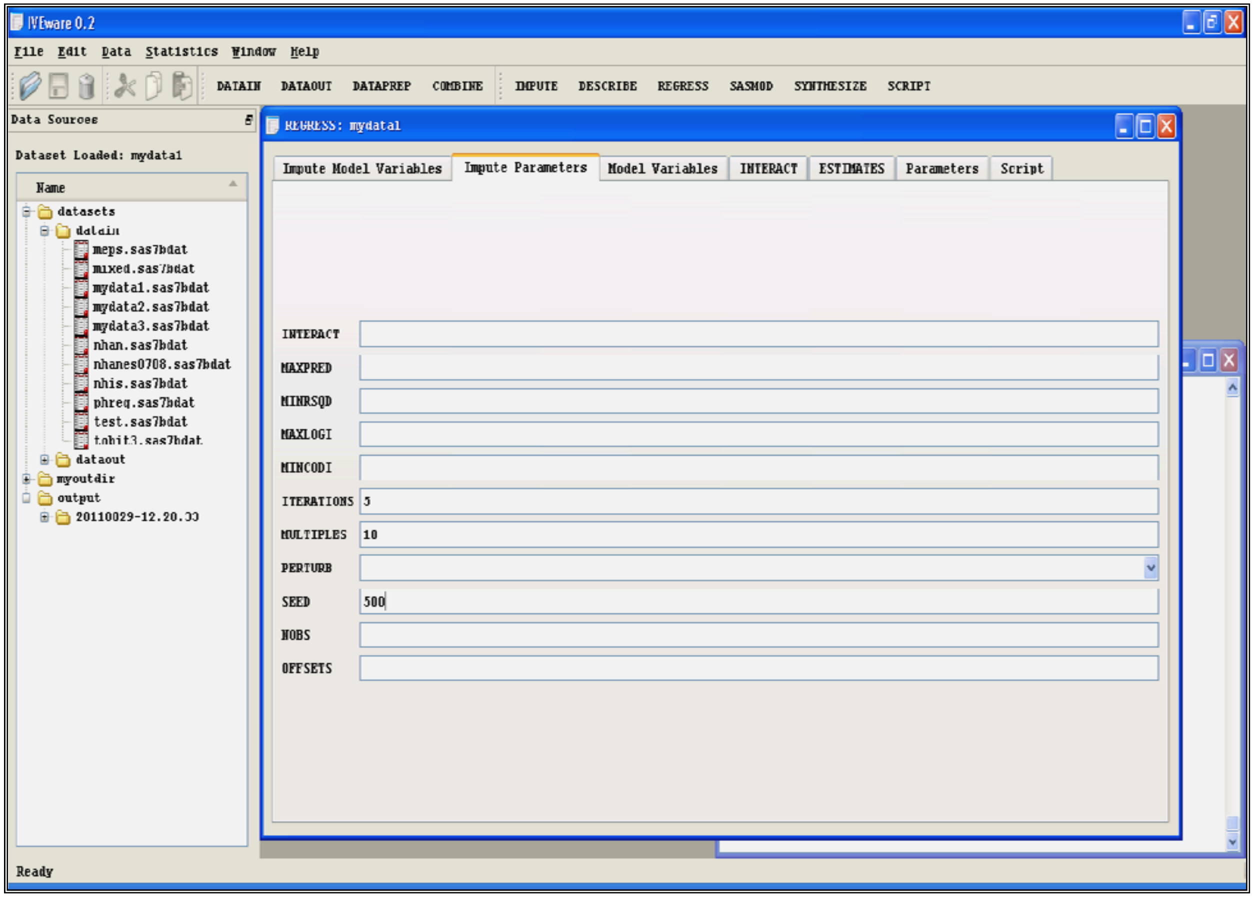
In the Impute Model Variables window you can declare variable type, restrict imputation variables, and apply value bounds to imputed variables. See The Impute Procedure (pages 11-27) for more details.



	Variable	Label	VariableType	Restrict	Bounds
2	AGE84	Age during 1984	continuous		
3	RACER	Race	transfer		
4	MARSTAT	Marital status	transfer		
5	VETERAN	Veteran status	transfer		
6	EDUC	Education of individual-completed ye…	continuous		
7	INCFAMR	Family income recode	transfer		
8	POVERTY	NHIS poverty index	transfer		
9	FAMRELR	Family relationship recode	transfer		
10	MAJACT	Major activity	transfer		
11	HEALTH	Health status	categorical		
12	RESPOND	Respondent	transfer		
13	HEIGHT	Height without shoes (in inches)	transfer		
14	WEIGHT	Weight without shoes	transfer		
15	BDDAY12	Bed days in past 12 months	transfer		
16	DV12	Doctor visits in past 12 months	continuous		

Imputation Parameters

Imputation parameters are entered in the Impute Parameter Window. In the example, 10 multiple imputations with 5 iterations are requested. The starting seed is 500. See The Impute Procedure (pages 11-27) for more details.



INTERACT

MAXPRED

MINRSQD

MAXLOGI

MINCODI

ITERATIONS 5

MULTIPLES 10

PERTURB

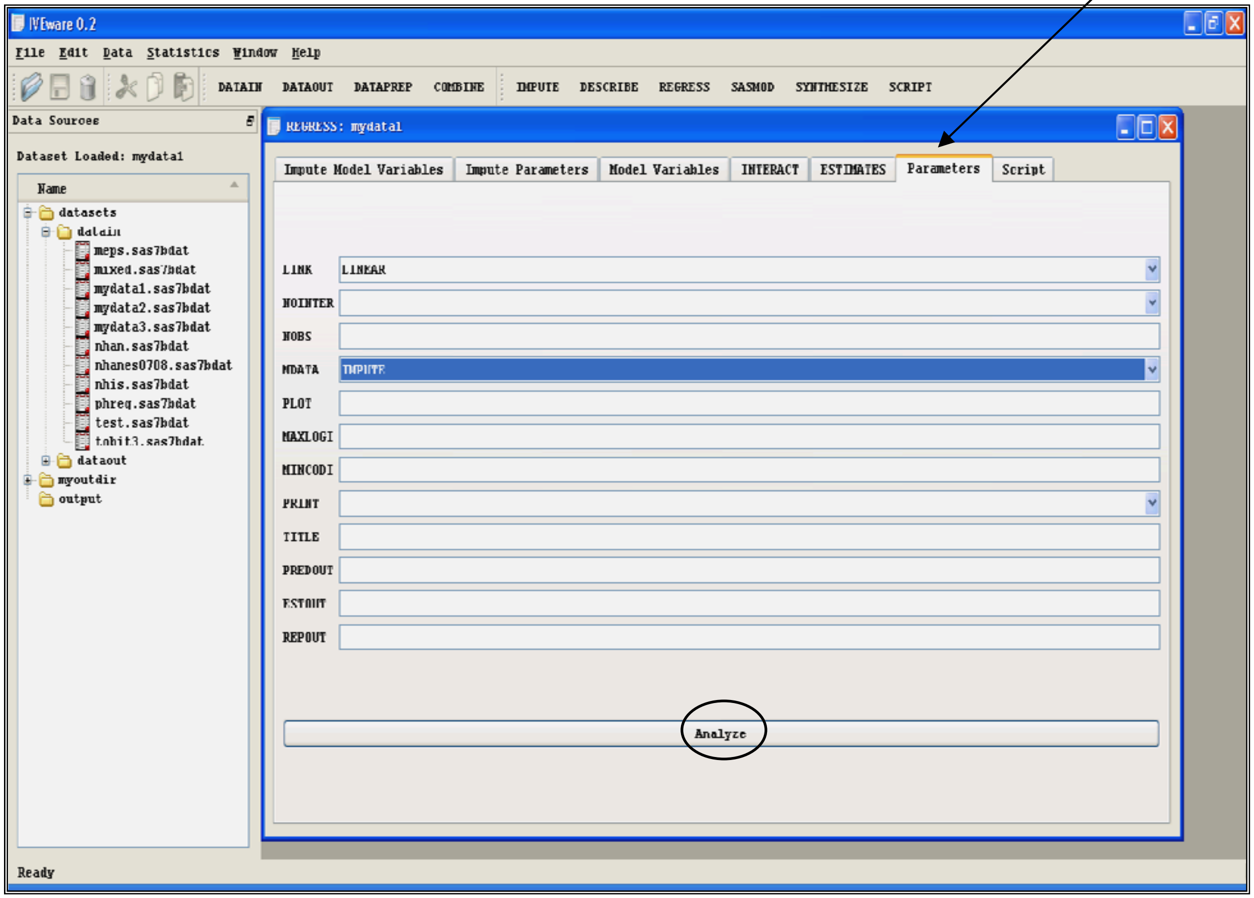
SEED 500

NOBS

OFFSETS

Regress Parameters

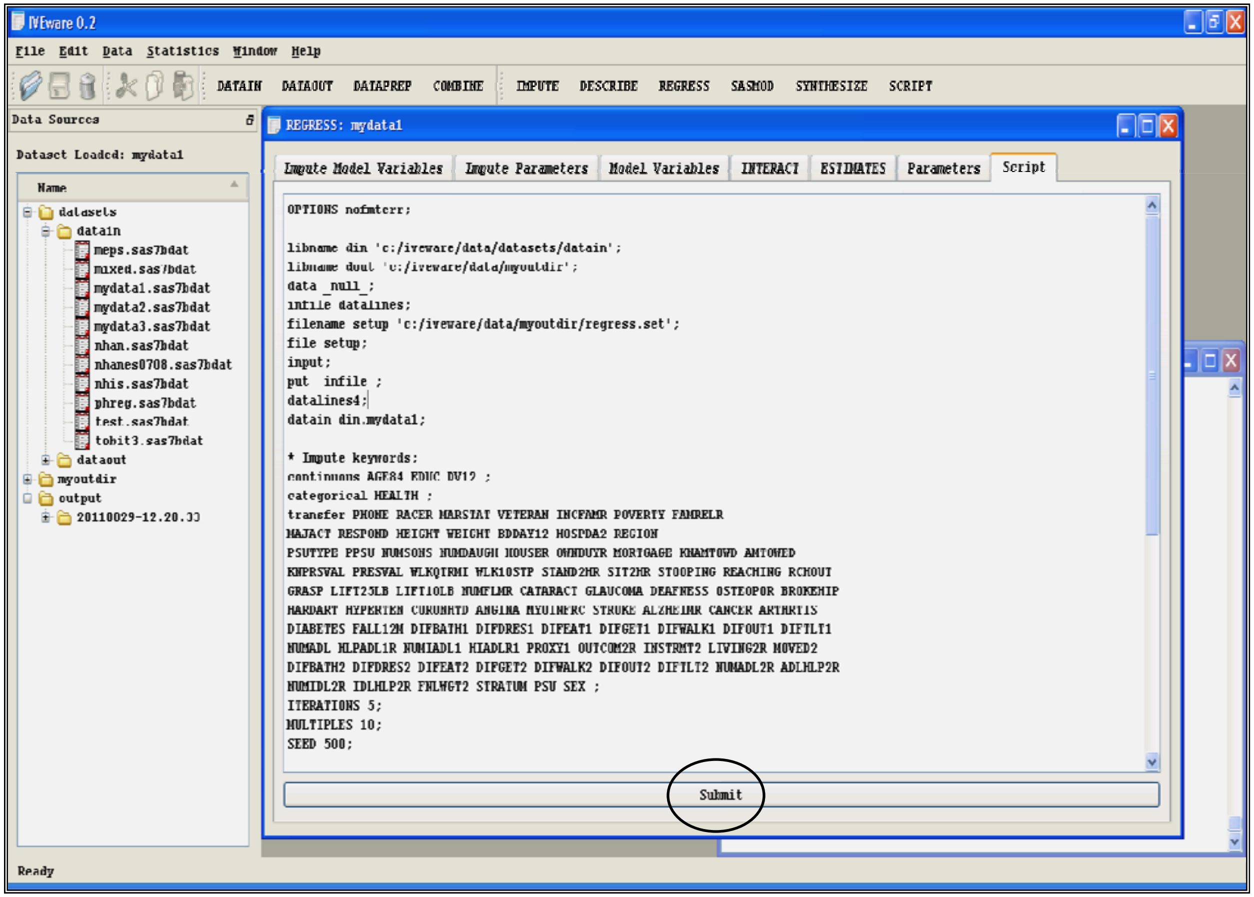
After you complete entering the imputation parameters, return to the original REGRESS Parameters window and click on the ANALYZE bar at the bottom to update the script with the new imputation instructions.



LINK: LINEAR
NOINTER:
NOBS:
MDATA: IMPUTE
PLOT:
MAXLOGI:
MINCODI:
PRINT:
TITLE:
PREDOUT:
ESTOUT:
REPOUT:

The Regress Script with Imputation Instructions

After clicking on the Script tab, the script will now include the imputation instructions. Click the SUBMIT bar at the bottom of the window and your combined REGRESS and IMPUTE script will be submitted for processing.

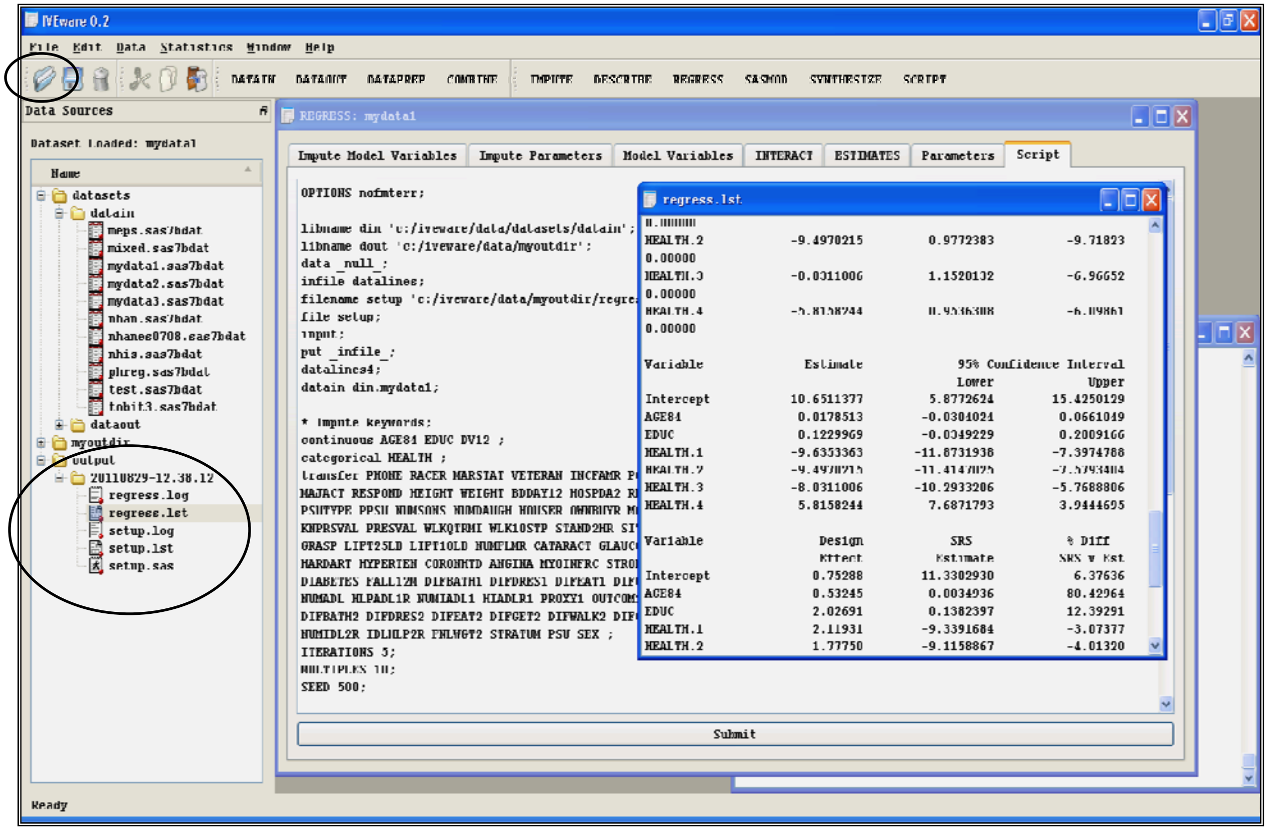


```
OPTIONS nofmterr;

libname din 'c:/iveware/data/datasets/datain';
libname dout 'c:/iveware/data/myoutdir';
data _null_;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/regress.set';
file setup;
input;
put  infile ;
datalines4;
datain din.mydata1;

* Impute keywords:
continuous AGE84 EDUC DV12 ;
categorical HEALTH ;
transfer PHONE RACER MARSTAT VETERAN INCFAMR POVERTY FAMRELR
MAJACT RESPOND HEIGHT WEIGHT BDDAY12 HOSPDA2 REGION
PSUTYPE PPSU NUMSONS NUMDAUGH HOUSER OWNDUYR MORTGAGE KHAMTOWD AMTOWED
KNPRSVAL PRESVAL WLKQIRMI WLK10STP STAND2HR SIT2HR STOOPING REACHING RCHOUT
GRASP LIFT25LB LIFT10LB NUMFLMR CATARACT GLAUCOMA DEAFNESS OSTEOPOR BROKEHIP
HARDART HYPERTEN CORONHTD ANGINA MYOINFRC STROKE ALZHEIMR CANCER ARTHRTIS
DIABETES FALL12M DIFBATH1 DIFDRES1 DIFEAT1 DIFGET1 DIFWALK1 DIFOUT1 DIFTLT1
NUMADL HLPADL1R NUMIADL1 HIADLR1 PROXY1 OUTCOM2R INSTRMT2 LIVING2R MOVED2
DIFBATH2 DIFDRES2 DIFEAT2 DIFGET2 DIFWALK2 DIFOUT2 DIFTLT2 NUMADL2R ADLHLP2R
NUMIDL2R IDLHLP2R FNLWGT2 STRATUM PSU SEX ;
ITERATIONS 5;
MULTIPLES 10;
SEED 500;
```

Output Files

After the script submission is completed, the OUTPUT folder will contain a sub-folder labeled with the date and time of the script submission. Here you will find the results of the Regress procedure. Errors in the submitted script will be reported in the regress.log file. If the script was successful the Regress results can be found in regress.lst. No imputed datasets will be outputted.

To open an output file, highlight the file and click on the File icon at the top left of the window. In the example, regress.lst has been opened.



```
OPTIONS nofmterr;

libname din 'c:/iveware/data/datasets/datain';
libname dout 'c:/iveware/data/myoutdir';
data _null_;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/regre
file setup;
input;
put _infile_;
datalines4;
datain din.mydata1;

* impute keywords;
continuous AGE84 EDUC DV12 ;
categorical HEALTH ;
transfer PHONE RACER MARSTAT VETERAN INCFAMR P
MAJACT RESPOND HEIGHT WEIGHT BDDAY12 HOSPDA2 R
PSUTYPE PPSU NUMSONS NUMDAUGH HOUSER OWNBUYR M
KNPRSVAL PRESVAL WLKQTRMI WLK10STP STAND2HR SI
GRASP LIFT25LD LIFT10LD NUMFLMR CATARACT GLAUC
HARDART HYPERTEN CORONHTD ANGINA MYOINFRC STRO
DIABETES FALL12M DIFBATH1 DIFDRES1 DIFEAT1 DIF
NUMADL HLPADL1R NUMIADL1 HIADLR1 PROXY1 OUTCOM
DIFBATH2 DIFDRES2 DIFEAT2 DIFGET2 DIFWALK2 DIF
NUMIDL2R IDLHLP2R FNLWGT2 STRATUM PSU SEX ;
ITERATIONS 5;
MULTIPLES 10;
SEED 500;
```

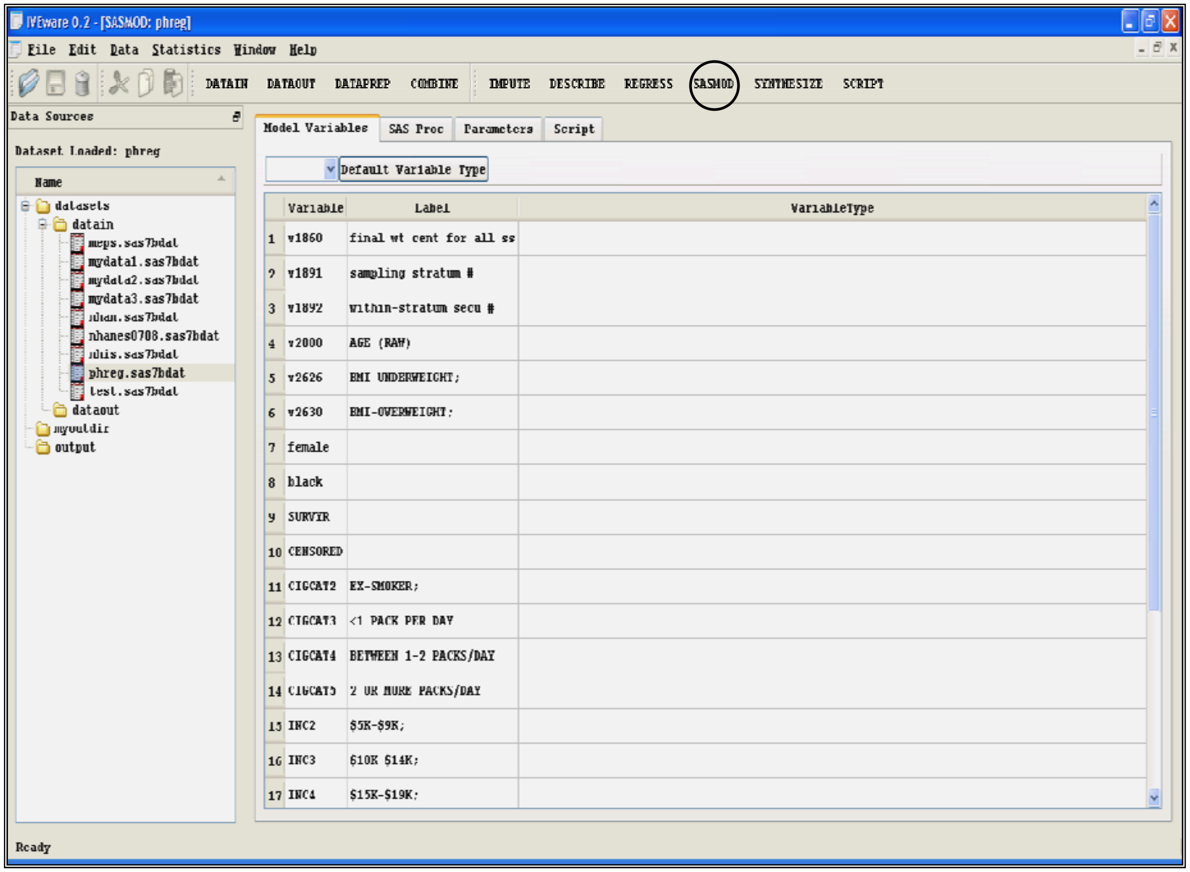
```
0.000000
HEALTH.2             -9.4970215        0.9772383        -9.71823
0.00000
HEALTH.3             -8.0311006        1.1520132        -6.96652
0.00000
HEALTH.4             -5.8158244        0.9536308        -6.09861
0.00000

Variable               Estimate        95% Confidence Interval
                                           Lower           Upper
Intercept            10.6511377        5.8772624      15.4250129
AGE84                 0.0178513       -0.0304024       0.0661049
EDUC                  0.1229969       -0.0349229       0.2009166
HEALTH.1             -9.6353363      -11.8731938      -7.3974788
HEALTH.2             -9.4970215      -11.4147025      -7.5793404
HEALTH.3             -8.0311006      -10.2933206      -5.7688806
HEALTH.4              5.8158244        7.6871793       3.9444695

Variable                 Design              SRS          % Diff
                         Effect         Estimate       SRS v Est
Intercept               0.75288       11.3302930         6.37636
AGE84                   0.53245        0.0034936        80.42964
EDUC                    2.02691        0.1382397        12.39291
HEALTH.1                2.11931       -9.3391684        -3.07377
HEALTH.2                1.77750       -9.1158867        -4.01320
```

8. The SASMOD Procedure

Selecting the SASMOD Procedure

After loading a dataset, click on the SASMOD button. This will open the Model Variables window, listing all variables in the loaded dataset.

In the example, PHREG was previously loaded. The dataset must be loaded before selecting an *IVEware* procedure (see page 7).

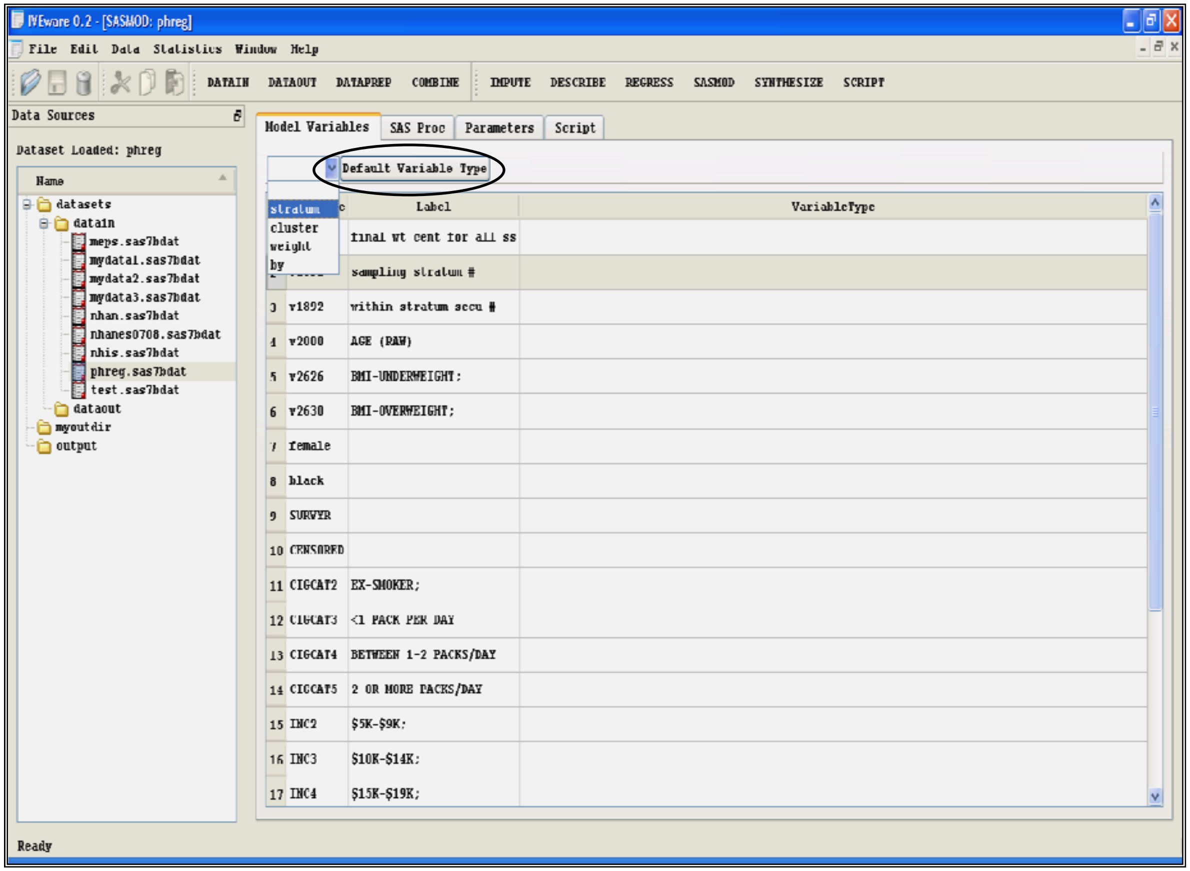


Dataset Loaded: phreg

	Variable	Label	VariableType
1	v1860	final wt cent for all ss	
2	v1891	sampling stratum #	
3	v1892	within-stratum secu #	
4	v2000	AGE (RAW)	
5	v2626	BMI UNDERWEIGHT;	
6	v2630	BMI-OVERWEIGHT;	
7	female		
8	black		
9	SURVIR		
10	CENSORED		
11	CIGCAT2	EX-SMOKER;	
12	CIGCAT3	<1 PACK PER DAY	
13	CIGCAT4	BETWEEN 1-2 PACKS/DAY	
14	CIGCAT5	2 OR MORE PACKS/DAY	
15	INC2	$5K-$9K;	
16	INC3	$10K $14K;	
17	INC4	$15K-$19K;	

Survey Design Variable Selection

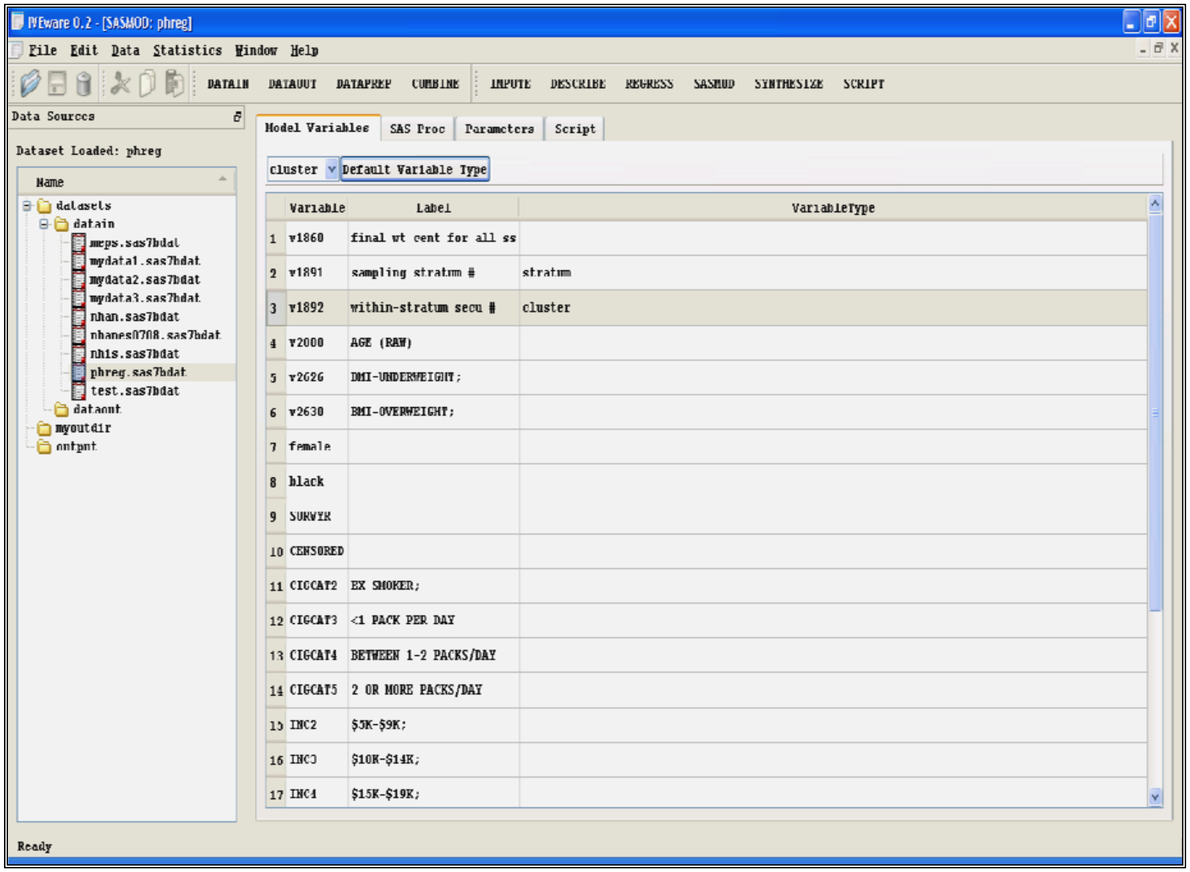
To incorporate survey design variables in your analysis, highlight the design variable in the Model Variables window. Then click the arrow on the dialog box to the left of the <u>DEFAULT VARIABLE TYPE</u> button and select the appropriate design feature—stratum, cluster or weight. Then click on the DEFAULT VARIABLE TYPE button.



	c	Label	VariableType
		final wt cent for all ss	
		sampling stratum #	
3	v1892	within stratum secu #	
4	v2000	AGE (RAW)	
5	v2626	BMI-UNDERWEIGHT;	
6	v2630	BMI-OVERWEIGHT;	
7	female		
8	black		
9	SURVYR		
10	CENSORED		
11	CIGCAT2	EX-SMOKER;	
12	CIGCAT3	<1 PACK PER DAY	
13	CIGCAT4	BETWEEN 1-2 PACKS/DAY	
14	CIGCAT5	2 OR MORE PACKS/DAY	
15	INC2	$5K-$9K;	
16	INC3	$10K-$14K;	
17	INC4	$15K-$19K;	

Survey Design Variable Selection

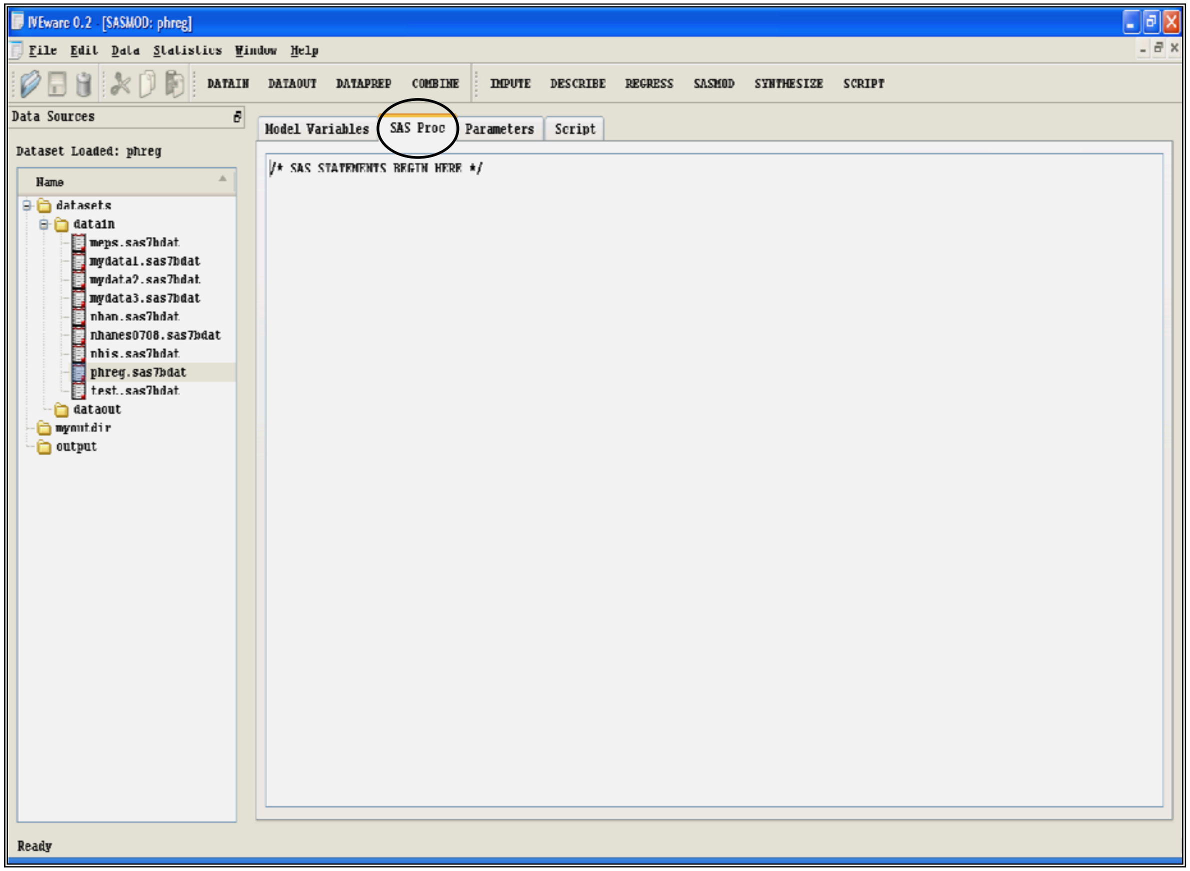
In the example, V1891 was selected as the stratum variable and V1982 as the cluster variable. No weight variable was assigned for this analysis.



	Variable	Label	VariableType
1	v1860	final wt cent for all ss	
2	v1891	sampling stratum #	stratum
3	v1892	within-stratum secu #	cluster
4	v2000	AGE (RAW)	
5	v2626	BMI-UNDERWEIGHT;	
6	v2630	BMI-OVERWEIGHT;	
7	female		
8	black		
9	SURVYR		
10	CENSORED		
11	CIGCAT2	EX SMOKER;	
12	CIGCAT3	<1 PACK PER DAY	
13	CIGCAT4	BETWEEN 1-2 PACKS/DAY	
14	CIGCAT5	2 OR MORE PACKS/DAY	
15	INC2	$5K-$9K;	
16	INC3	$10K-$14K;	
17	INC4	$15K-$19K;	

SAS Proc Window

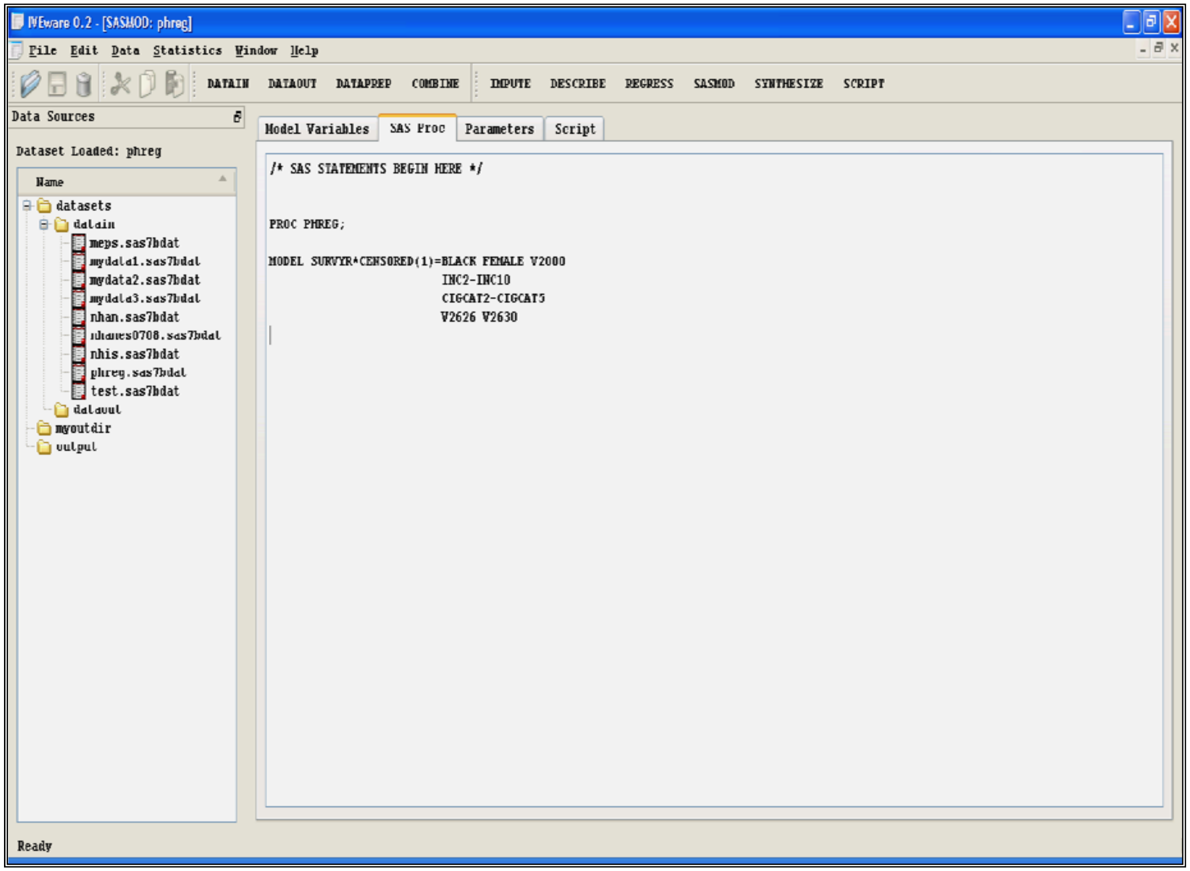
Clicking on the SAS Proc tab opens a window in which you can directly type SAS statements associated with the SAS Proc you wish to use. Currently the following SAS Procs are available: CALIS, CATMOD, GENMOD, LIFEREG, MIXED, NLIN, PHREG, and PROBIT.



```
/* SAS STATEMENTS BEGIN HERE */
```

SAS Proc Window

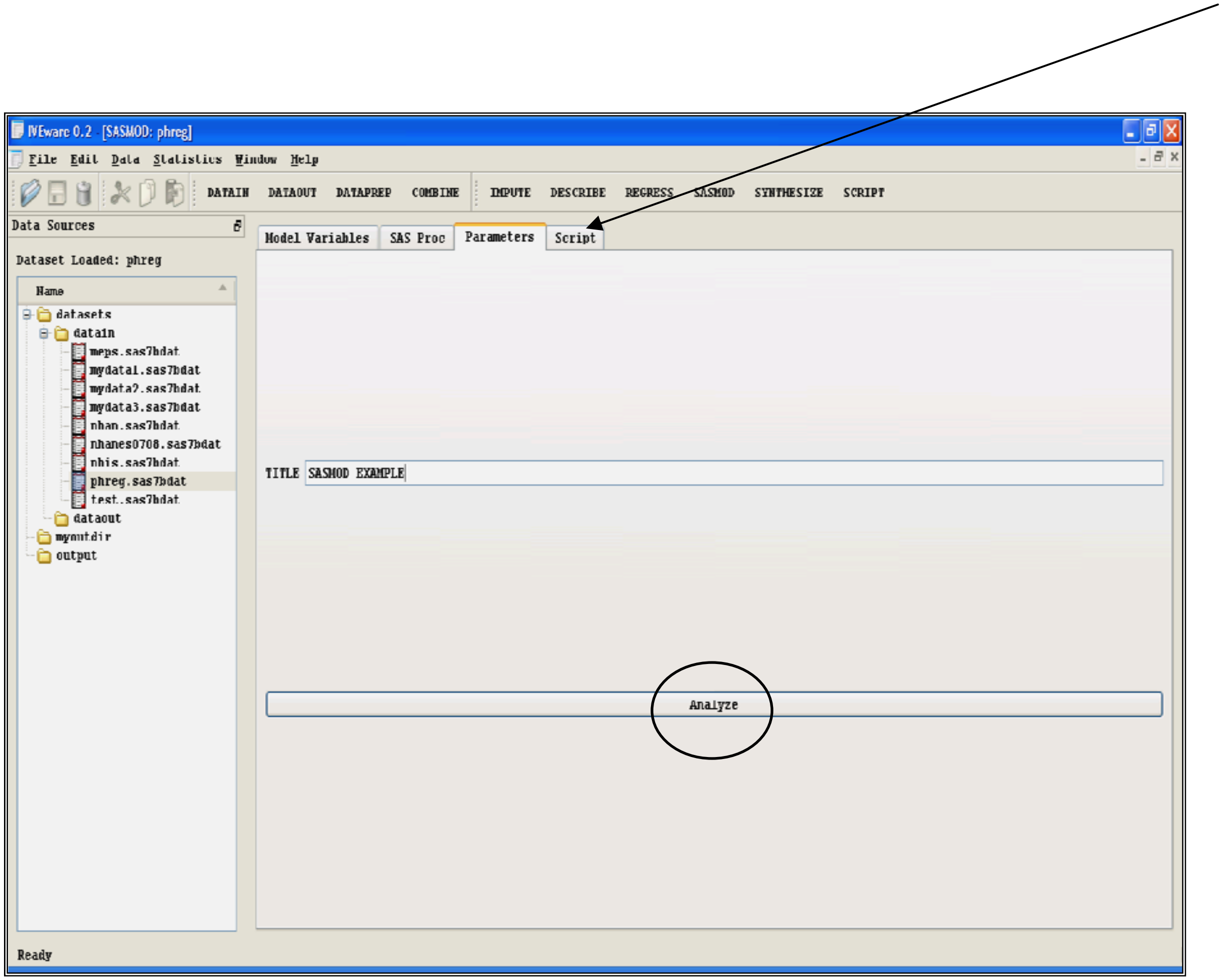
In the example, the SAS proportional hazard modeling procedure PHREG will be used. Users should consult SAS documentation regarding the appropriate SAS statements associated with a selected procedure.



```
/* SAS STATEMENTS BEGIN HERE */


PROC PHREG;

MODEL SURVYR*CENSORED(1)=BLACK FEMALE V2000
                         INC2-INC10
                         CIGCAT2-CIGCAT5
                         V2626 V2630
```

SASMOD Parameters

The parameter window has only one optional parameter--TITLE. A title can be useful in organizing and keeping track of your *IVEware* submissions. Before exiting this window you must click on the ANALYZE bar in at the bottom portion of Parameter window to produce the SASMOD script. The script can be viewed by clicking on the SCRIPT tab.



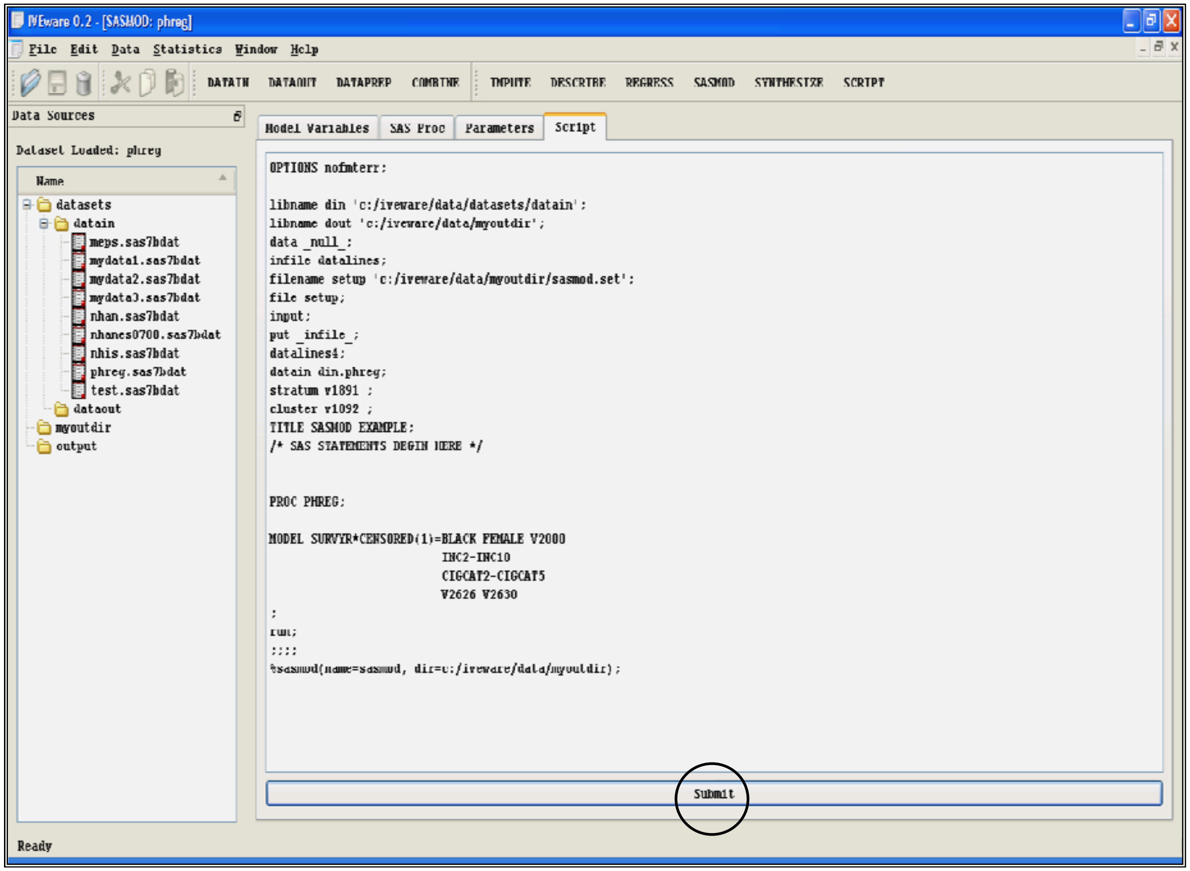
Dataset Loaded: phreg

TITLE SASMOD EXAMPLE

The SASMOD Script Window

The SASMOD script contains instructions entered in the Model Variables, SAS Proc, and Parameter windows. The script can be modified by returning to those windows and entering changes. After entering changes you must once again click on the ANALYZE bar in Parameter window to update the script. You can also type changes directly in the script window.

Submit your script for processing by clicking on the SUBMIT bar at the bottom of the Script window.

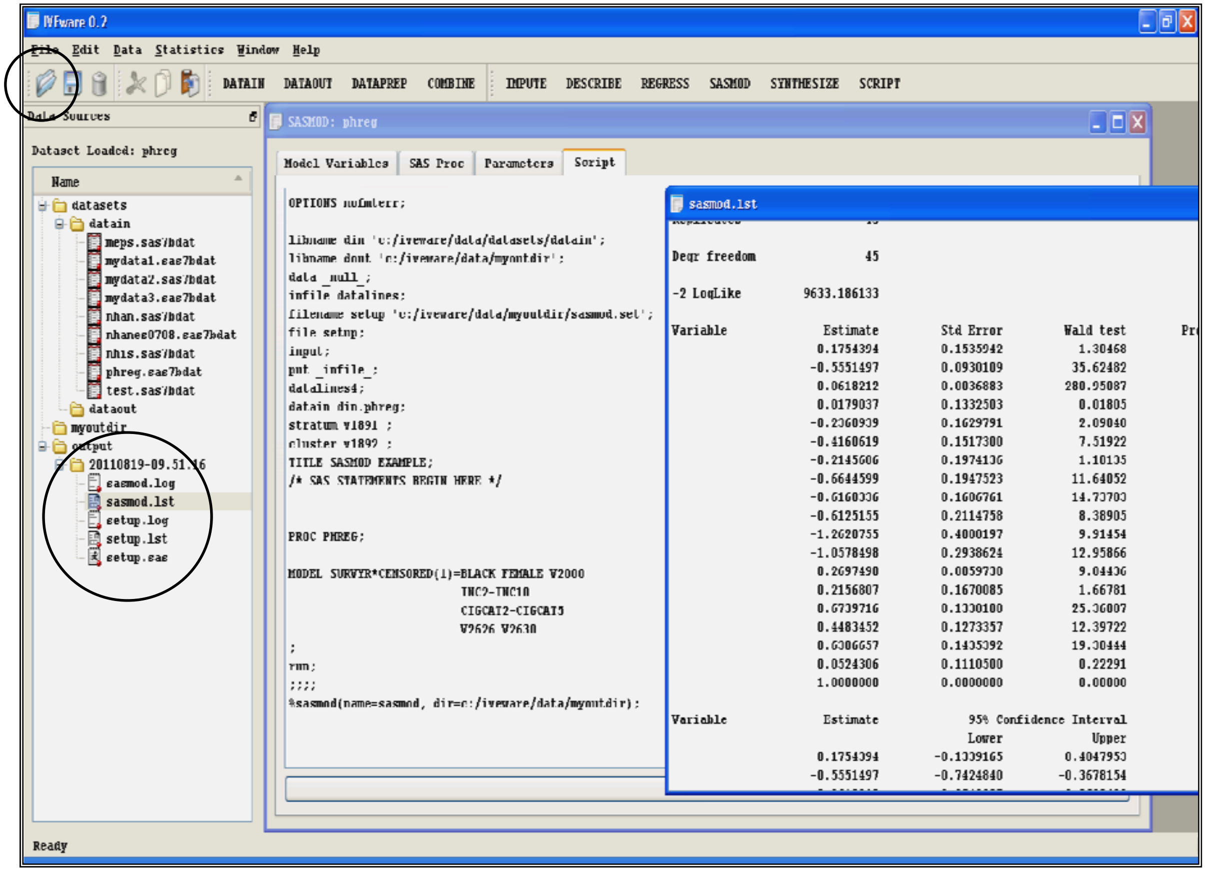


```
OPTIONS nofmterr;

libname din 'c:/iveware/data/datasets/datain';
libname dout 'c:/iveware/data/myoutdir';
data _null_;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/sasmod.set';
file setup;
input;
put _infile_;
datalines4;
datain din.phreg;
stratum v1891 ;
cluster v1092 ;
TITLE SASMOD EXAMPLE;
/* SAS STATEMENTS BEGIN HERE */


PROC PHREG;

MODEL SURVYR*CENSORED(1)=BLACK FEMALE V2000
                         INC2-INC10
                         CIGCAT2-CIGCAT5
                         V2626 V2630
;
run;
;;;;
%sasmod(name=sasmod, dir=c:/iveware/data/myoutdir);
```

Output Files

After the script submission is completed, the OUTPUT folder will contain a sub-folder labeled with the date and time of the script submission. Here you will find the results of the SASMOD procedure. Errors in the submitted script will be reported in the sasmod.log file. If the script was successful the SASMOD results can be found in sasmod.lst.

To open an output file, highlight the file and click on the File icon at the top left of the window. In the example, sasmod.lst has been opened.



```
OPTIONS nofmterr;

libname din 'c:/iveware/data/datasets/datain';
libname dout 'c:/iveware/data/myoutdir';
data _null_;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/sasmod.set';
file setup;
input;
put _infile_;
datalines4;
datain din.phreg;
stratum v1891 ;
cluster v1892 ;
TITLE SASMOD EXAMPLE;
/* SAS STATEMENTS BEGIN HERE */


PROC PHREG;

MODEL SURVYR*CENSORED(1)=BLACK FEMALE V2000
                         INC2-INC10
                         CIGCAT2-CIGCAT5
                         V2626 V2630
;
run;
;;;;
%sasmod(name=sasmod, dir=c:/iveware/data/myoutdir);
```

```
Degr freedom           45

-2 LogLike       9633.186133

Variable            Estimate       Std Error       Wald test      Pr
                   0.1754394       0.1535942         1.30468
                  -0.5551497       0.0930109        35.62482
                   0.0618212       0.0036883       280.95087
                   0.0179037       0.1332503         0.01805
                  -0.2360939       0.1629791         2.09040
                  -0.4160619       0.1517300         7.51922
                  -0.2145606       0.1974136         1.10135
                  -0.6644599       0.1947523        11.64052
                  -0.6160336       0.1606761        14.73703
                  -0.6125155       0.2114758         8.38905
                  -1.2620755       0.4000197         9.91454
                  -1.0578498       0.2938624        12.95866
                   0.2697490       0.0059730         9.04436
                   0.2156807       0.1670085         1.66781
                   0.6739716       0.1330100        25.36007
                   0.4483452       0.1273357        12.39722
                   0.6306657       0.1435392        19.30444
                   0.0524306       0.1110500         0.22291
                   1.0000000       0.0000000         0.00000

Variable            Estimate       95% Confidence Interval
                                       Lower           Upper
                   0.1754394      -0.1339165       0.4047953
                  -0.5551497      -0.7424840      -0.3678154
```

Glossary of SASMOD Commands

STRATUM variable name;

variable name is the name of the stratum variable. No missing values are allowed for the stratum variable**.**

CLUSTER variable name;

variable name is the Primary Sampling Unit (PSU) or Sampling Error Computing Unit (SECU) variable. No missing values are allowed for the cluster variable.

WEIGHT variable name;

variable name is the survey weight variable. Survey weights are usually the product of selection, nonresponse adjustment and poststratification weights. No missing values are allowed for the weight variable.

NOTES:

1. If the STRATUM, CLUSTER and WEIGHT variable are not specified then a simple random sample analysis will be performed.
2. If a design based analysis involves only a WEIGHT variable and no STRATUM or CLUSTER variable then a pseudo stratification variable and a pseudo cluster variable should be used. When using pseudo variables, all observations in the data set should have the same value for the pseudo STRATUM variable (e.g., 1), while each observation should have a unique value on the pseudo CLUSTER variable (e.g., observation ID number or SAS system variable _N_). The pseudo variables should be created in a SAS data step prior to performing the analysis. See the Appendix for an example data step creating pseudo stratification and pseudo cluster variables. The inclusion of pseudo variables will increase the time SASMOD needs for analysis.

BY variable list;

The regression analysis will be performed for each level of the variable(s) specified in the BY statement.

For instance,

```
BY Gender;
```

will produce regressions for each of the two levels of Gender.

If the variable Agecat is age in 3 categories then

```
BY Gender Agecat;
```

will produce regressions for each of the six combinations of Gender and Agecat.

SAS statements

PROC procedure name;

procedure name is the name of the SAS procedure. The currently implemented procedures are CALIS, CATMOD, GENMOD, LIFEREG, MIXED, NLIN, PHREG, and PROBIT. This statement must follow the SASMOD statements described above, except for RUN.

Other SAS statements

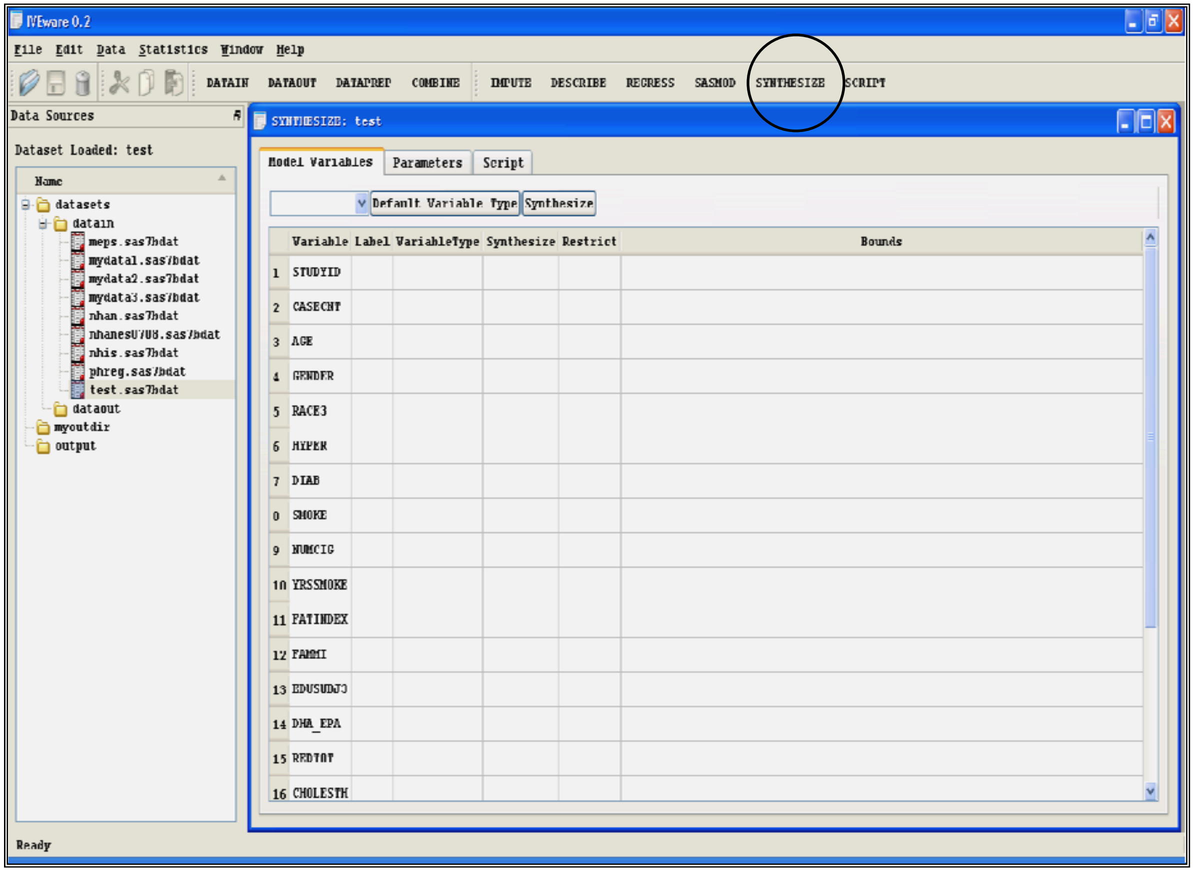
can be used as appropriate for the procedure. However, do not use statements that might lead to more than one model or different models in different replicates or multiples. For example, more than one model statement or specifying a stepwise model is not permitted.

9. The Synthesize Procedure

Selecting the Synthesize Procedure

After loading a dataset, click on the SYNTHESIZE button. This will open the Model Variables window, listing all variables in the loaded dataset.

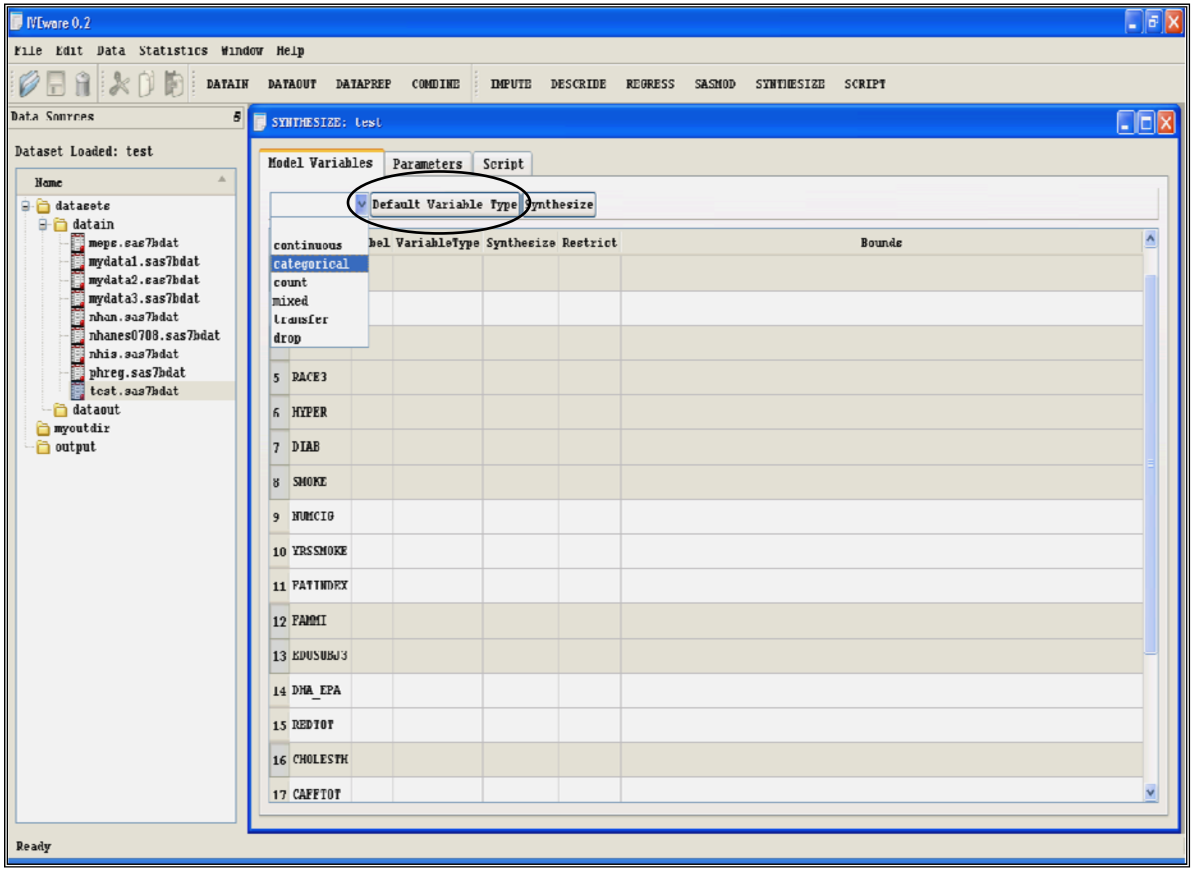
In this example, TEST was previously loaded. The dataset must be loaded before selecting an *IVEware* procedure (see page 7).



	Variable	Label	VariableType	Synthesize	Restrict	Bounds
1	STUDYID					
2	CASECNT					
3	AGE					
4	GENDER					
5	RACE3					
6	HYPER					
7	DIAB					
8	SMOKE					
9	NUMCIG					
10	YRSSMOKE					
11	FATINDEX					
12	FAMMI					
13	EDUSUDJ3					
14	DHA_EPA					
15	REDTOT					
16	CHOLESTH					

Declaring Variable Type

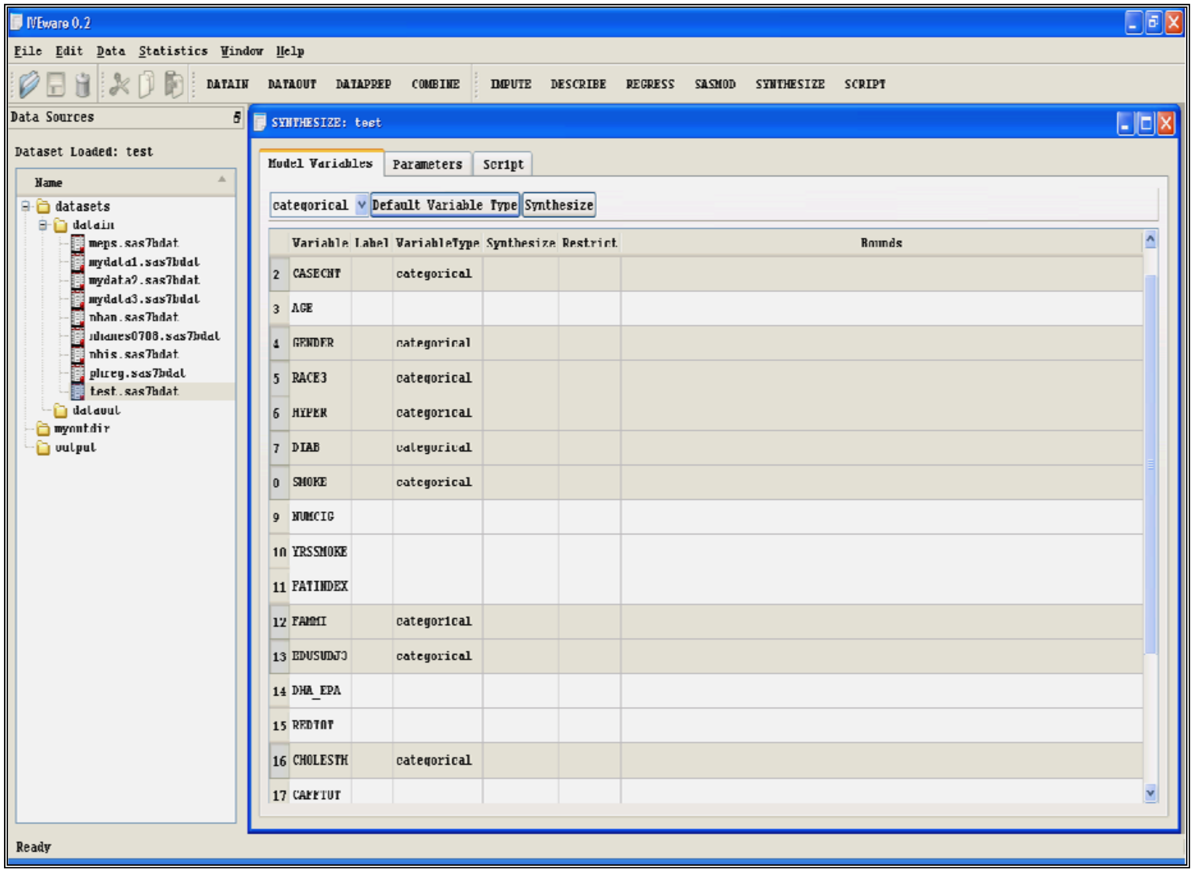
To declare a variable type, highlight the variable (s) in the Model Variables window. Click the arrow on the dialog box to the left of the DEFAULT VARIABLE button and select the appropriate type. Then click on the DEFAULT VARIABLE TYPE button.



		bel	VariableType	Synthesize	Restrict	Bounds
5	RACE3					
6	HYPER					
7	DIAB					
8	SMOKE					
9	NUMCIG					
10	YRSSMOKE					
11	FATINDEX					
12	FAMMI					
13	EDUSUBJ3					
14	DHA_EPA					
15	REDTOT					
16	CHOLESTH					
17	CAFFTOT					

Declaring Variable Type

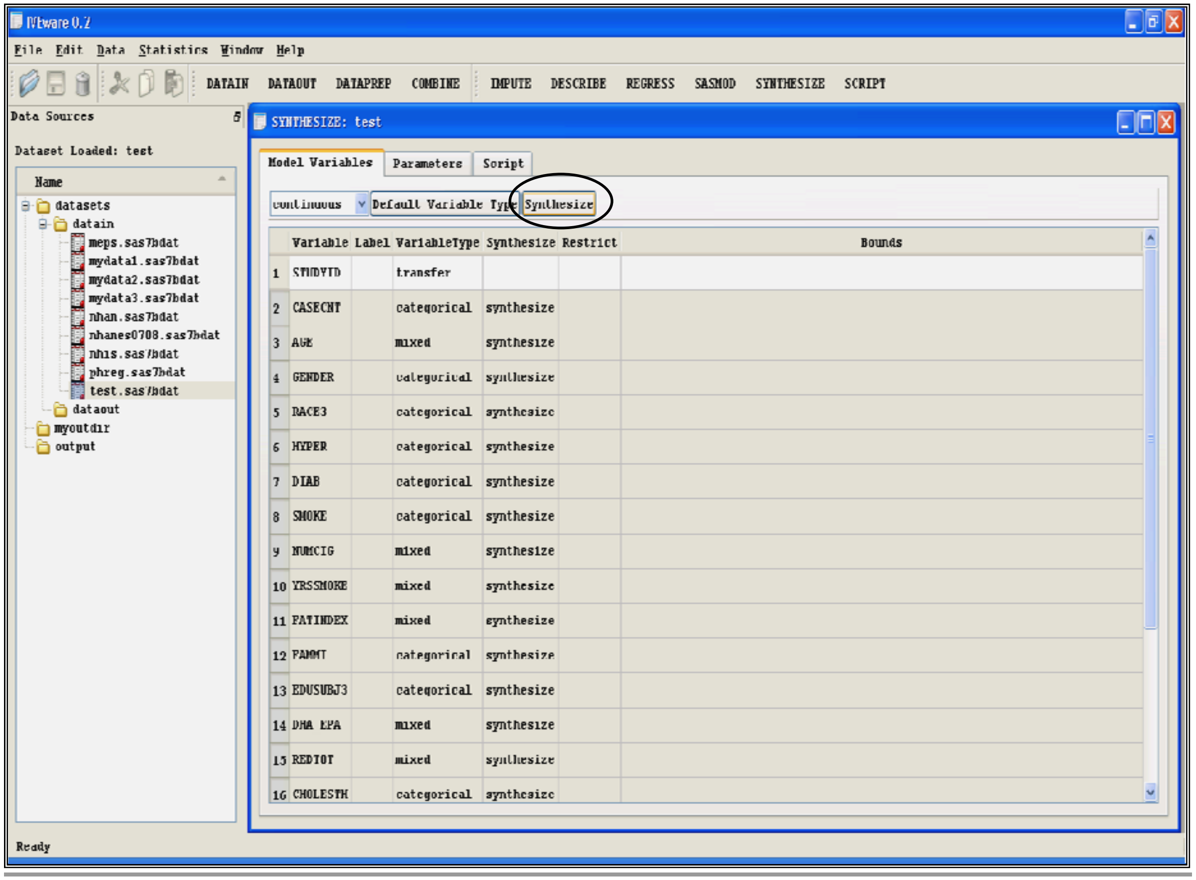
After the DEFAULT VARIABLE TYPE button is clicked the selected type appears in the column headed "Variable Type." In the example, nine variables were declared categorical.



	Variable	Label	VariableType	Synthesize	Restrict	Bounds
2	CASECNT		categorical			
3	AGE					
4	GENDER		categorical			
5	RACE3		categorical			
6	HYPER		categorical			
7	DIAB		categorical			
0	SMOKE		categorical			
9	NUMCIG					
10	YRSSMOKE					
11	FATINDEX					
12	FAMMI		categorical			
13	EDUSUDJ3		categorical			
14	DHA_EPA					
15	REDTOT					
16	CHOLESTH		categorical			
17	CAFFTOT					

Selecting Variables to be Synthesized

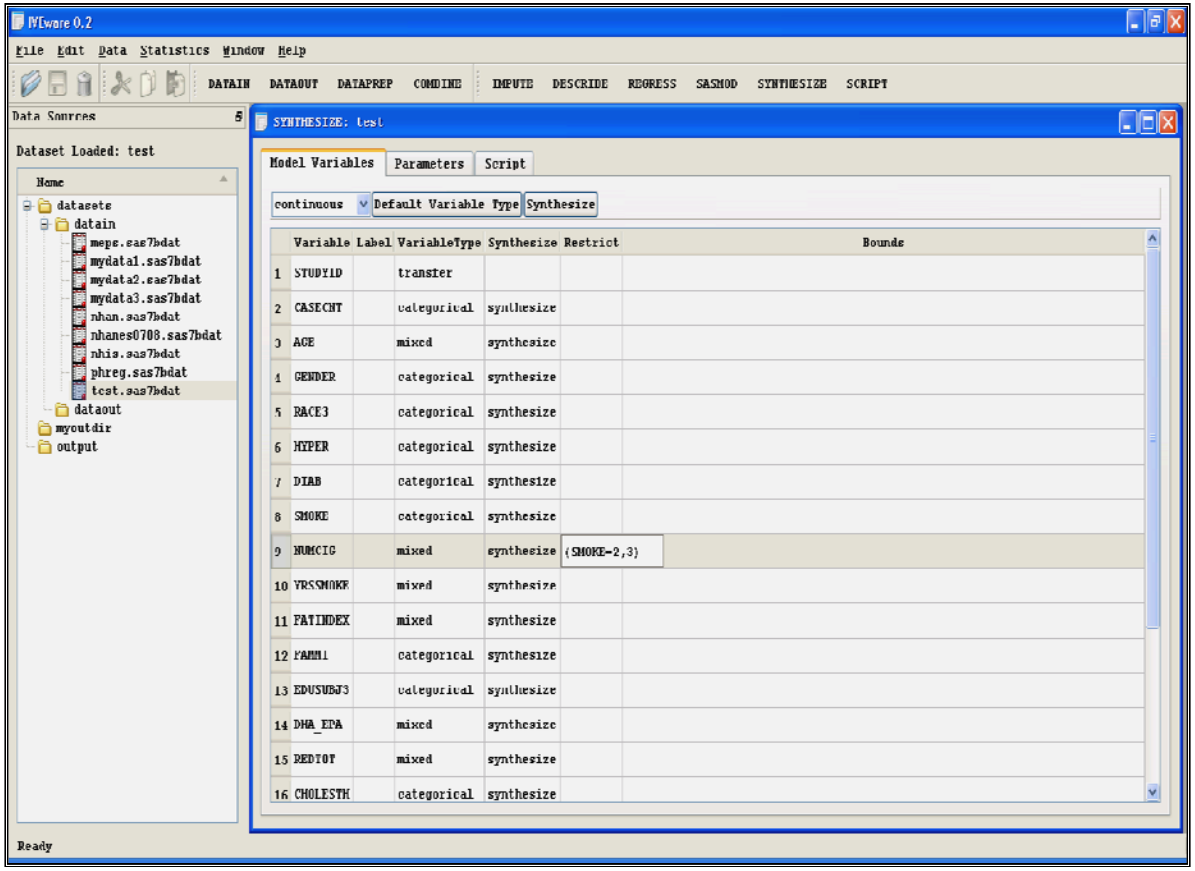
To select variables to be synthesized, highlight the variables and click on the SYNTHESIZE button.



	Variable	Label	VariableType	Synthesize	Restrict	Bounds
1	STUDYID		transfer			
2	CASECNT		categorical	synthesize		
3	AGE		mixed	synthesize		
4	GENDER		categorical	synthesize		
5	RACE3		categorical	synthesize		
6	HYPER		categorical	synthesize		
7	DIAB		categorical	synthesize		
8	SMOKE		categorical	synthesize		
9	NUMCIG		mixed	synthesize		
10	YRSSMOKE		mixed	synthesize		
11	FATINDEX		mixed	synthesize		
12	FAMMT		categorical	synthesize		
13	EDUSUBJ3		categorical	synthesize		
14	DHA EPA		mixed	synthesize		
15	REDTOT		mixed	synthesize		
16	CHOLESTH		categorical	synthesize		

Restricting Synthesized Variables

To restrict a variable, highlight a variable by clicking on its name and then click the box in the “Restrict” column. In the box type the restriction. In the example, the variable NUMCIG (Number of Cigarettes Smoked) is restricted to SMOKE codes 2 and 3 (Smokers and Former Smokers).



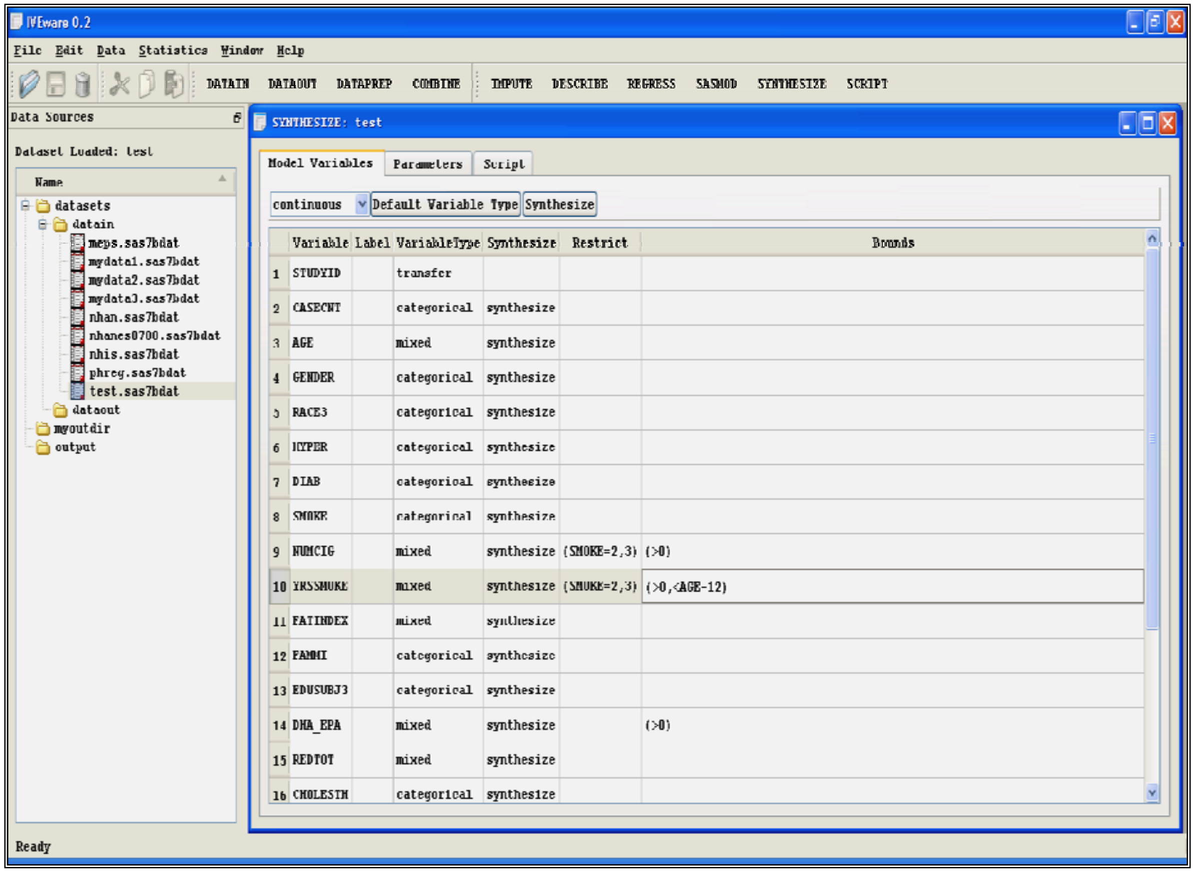
Dataset Loaded: test

	Variable	Label	VariableType	Synthesize	Restrict	Bounds
1	STUDYID		transfer			
2	CASECNT		categorical	synthesize		
3	AGE		mixed	synthesize		
4	GENDER		categorical	synthesize		
5	RACE3		categorical	synthesize		
6	HYPER		categorical	synthesize		
7	DIAB		categorical	synthesize		
8	SMOKE		categorical	synthesize		
9	NUMCIG		mixed	synthesize	{SMOKE=2,3}	
10	YRSSMOKE		mixed	synthesize		
11	FATINDEX		mixed	synthesize		
12	FAMHI		categorical	synthesize		
13	EDUSUBJ3		categorical	synthesize		
14	DHA_EPA		mixed	synthesize		
15	REDTOT		mixed	synthesize		
16	CHOLESTH		categorical	synthesize		

.

Applying Bounds to Synthesized Variables

To place bounds on a synthesized value, highlight a variable by clicking on its name. Then click the box in the “Bound” column and type in the bound instruction. In the example, the synthesized value of YRSSMOKE (Number of Years Smoked) must be greater than 0 and less than the respondent’s current age minus 12. Smoking is assumed not to begin before the age of 12.



Dataset Loaded: test

continuous

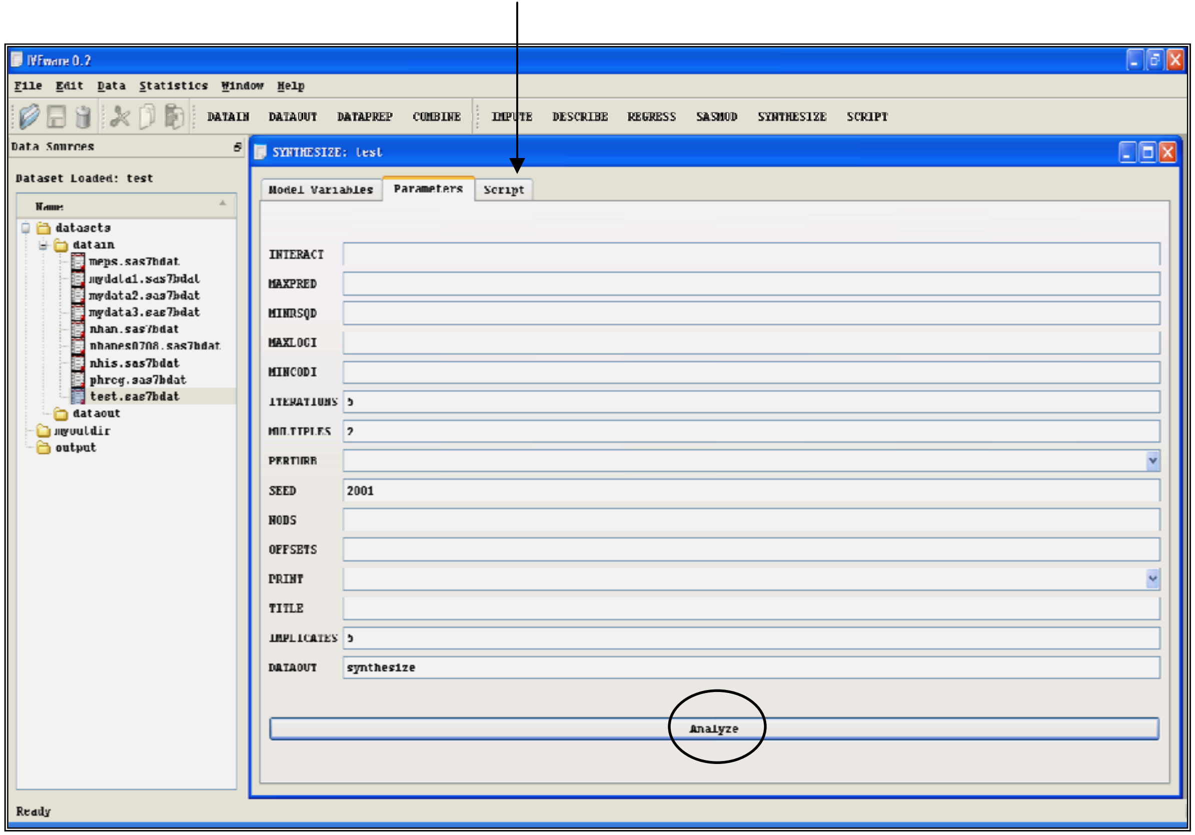
	Variable	Label	VariableType	Synthesize	Restrict	Bounds
1	STUDYID		transfer			
2	CASECNT		categorical	synthesize		
3	AGE		mixed	synthesize		
4	GENDER		categorical	synthesize		
5	RACE3		categorical	synthesize		
6	HYPER		categorical	synthesize		
7	DIAB		categorical	synthesize		
8	SMOKE		categorical	synthesize		
9	NUMCIG		mixed	synthesize	(SMOKE=2,3)	(>0)
10	YRSSMOKE		mixed	synthesize	(SMOKE=2,3)	(>0,<AGE-12)
11	FATINDEX		mixed	synthesize		
12	FAMMI		categorical	synthesize		
13	EDUSUBJ3		categorical	synthesize		
14	DHA_EPA		mixed	synthesize		(>0)
15	REDTOT		mixed	synthesize		
16	CHOLESTH		categorical	synthesize		

Synthesize Parameters

Clicking on the Parameters tab opens the Synthesize Parameter window. Enter the appropriate parameter value in the space to the right of the parameter name. In the example, IMPLICATES is set to 5 and MULTIPLES is set to 2. Ten synthetic datasets will be outputted, five for each multiple imputation.

In the last parameter, DATAOUT, you can specify a name for the outputted files. The default name is Synthesize.

After entering all parameters, click on the ANALYZE bar at the bottom of the window. This will generate the synthesize script. The script can be viewed by clicking on the SCRIPT tab.



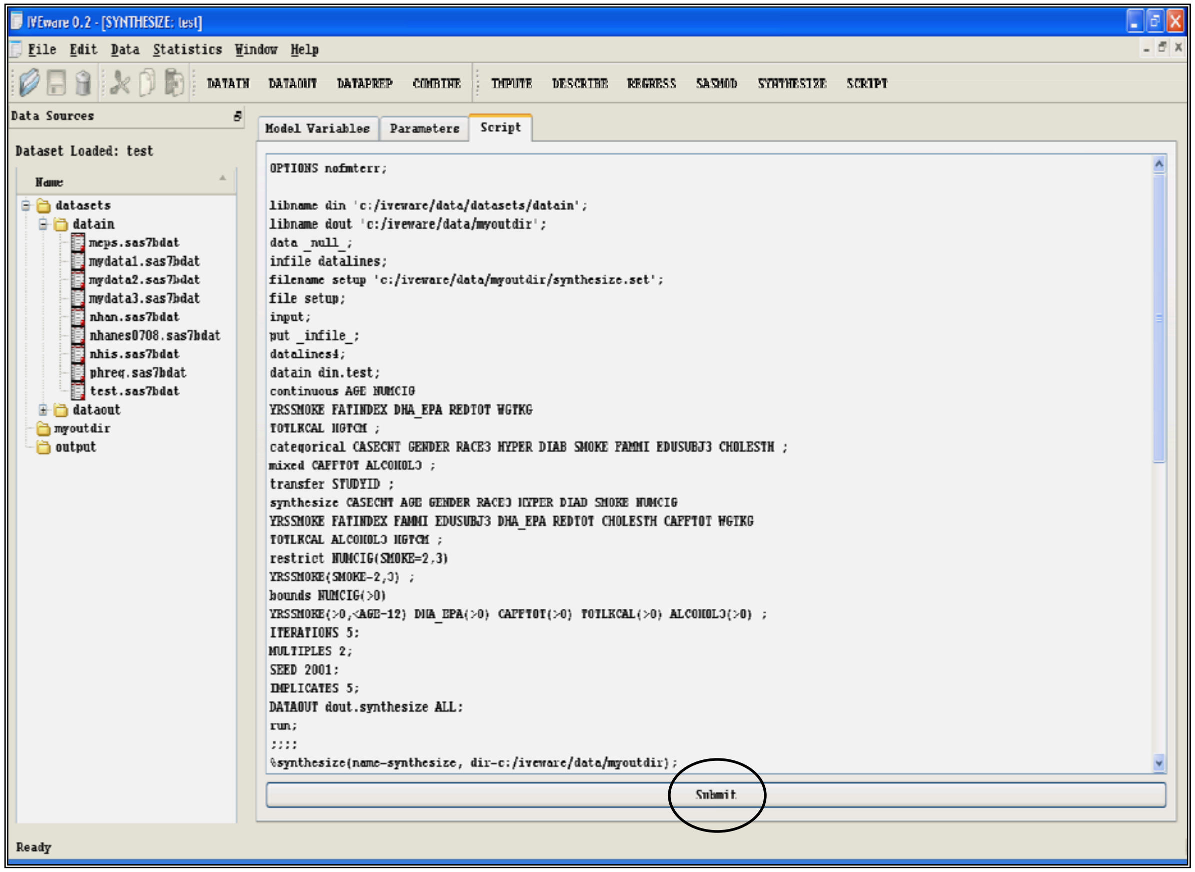
INTERACT
MAXPRED
MINRSQD
MAXLOGI
MINCODI
ITERATIONS 5
MULTIPLES 2
PERTURB
SEED 2001
NOBS
OFFSETS
PRINT
TITLE
IMPLICATES 5
DATAOUT synthesize

.

The Synthesize Script Window

The synthesize script contains instructions entered in the Model Variables and Parameter windows. The script can be modified by returning to the Model Variables or Parameter window and entering changes. After entering changes you must once again click on the ANALYZE bar in the Parameter window to update the script. You can also type changes directly in the script window.

Submit your script for processing by clicking on the SUBMIT bar at the bottom of the Script window.

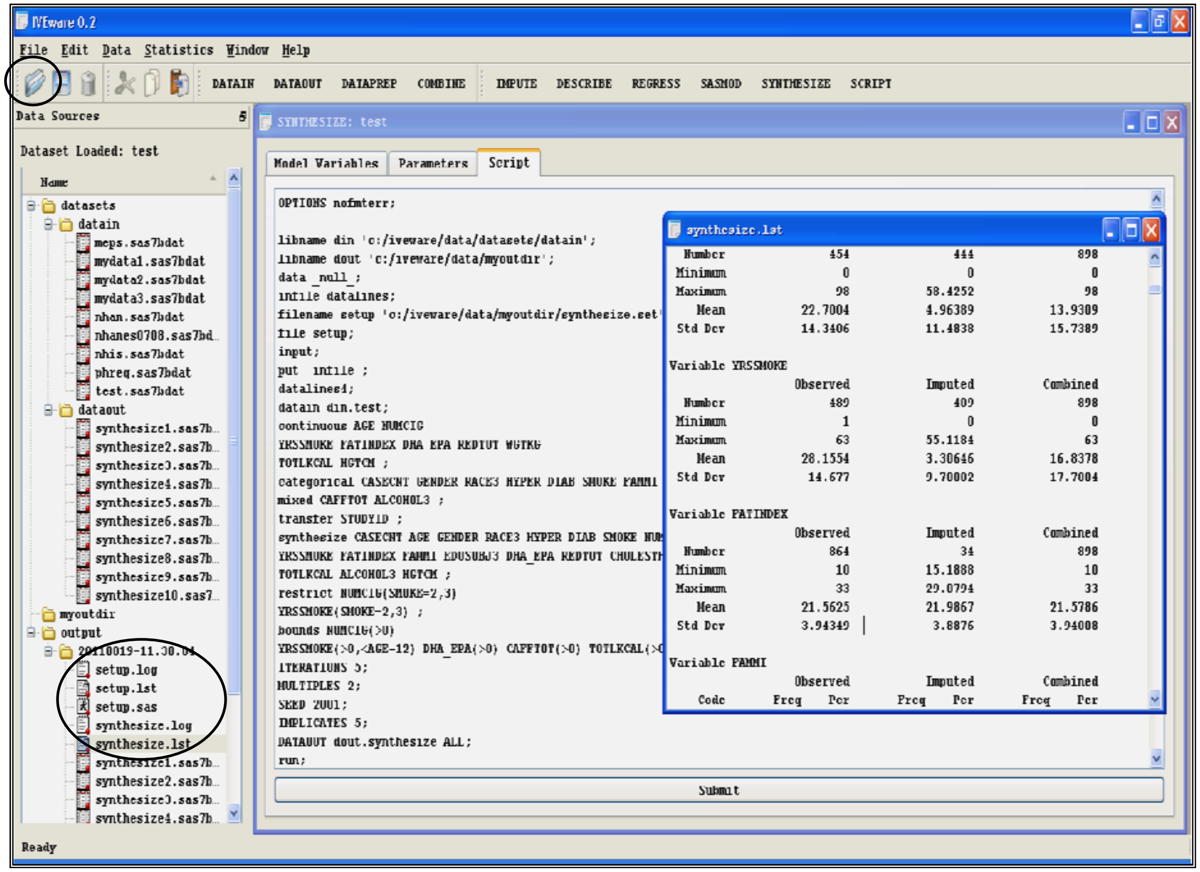


```
OPTIONS nofmterr;

libname din 'c:/iveware/data/datasets/datain';
libname dout 'c:/iveware/data/myoutdir';
data _null_;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/synthesize.set';
file setup;
input;
put _infile_;
datalines4;
datain din.test;
continuous AGE NUMCIG
YRSSMOKE FATINDEX DHA_EPA REDTOT WGTKG
TOTLKCAL HGTCM ;
categorical CASECNT GENDER RACE3 HYPER DIAB SMOKE FAMMI EDUSUBJ3 CHOLESTH ;
mixed CAFFTOT ALCOHOL3 ;
transfer STUDYID ;
synthesize CASECNT AGE GENDER RACE3 HYPER DIAB SMOKE NUMCIG
YRSSMOKE FATINDEX FAMMI EDUSUBJ3 DHA_EPA REDTOT CHOLESTH CAFFTOT WGTKG
TOTLKCAL ALCOHOL3 HGTCM ;
restrict NUMCIG(SMOKE=2,3)
YRSSMOKE(SMOKE=2,3) ;
bounds NUMCIG(>0)
YRSSMOKE(>0,<AGE-12) DHA_EPA(>0) CAFFTOT(>0) TOTLKCAL(>0) ALCOHOL3(>0) ;
ITERATIONS 5:
MULTIPLES 2;
SEED 2001:
IMPLICATES 5;
DATAOUT dout.synthesize ALL:
run;
;;;;
%synthesize(name=synthesize, dir=c:/iveware/data/myoutdir);
```

Output Files

After the script submission is completed, the OUTPUT folder will contain a sub-folder with the date and time of the script submission. Here you will find the results of the Synthesize procedure. Errors in the submitted script will be reported in the synthesize.log file. If the script was successful the imputation results can be found in synthesize.lst.

To open an output file, highlight the file and click on the File icon at the top left of the window. In the example, synthesize.lst has been opened.

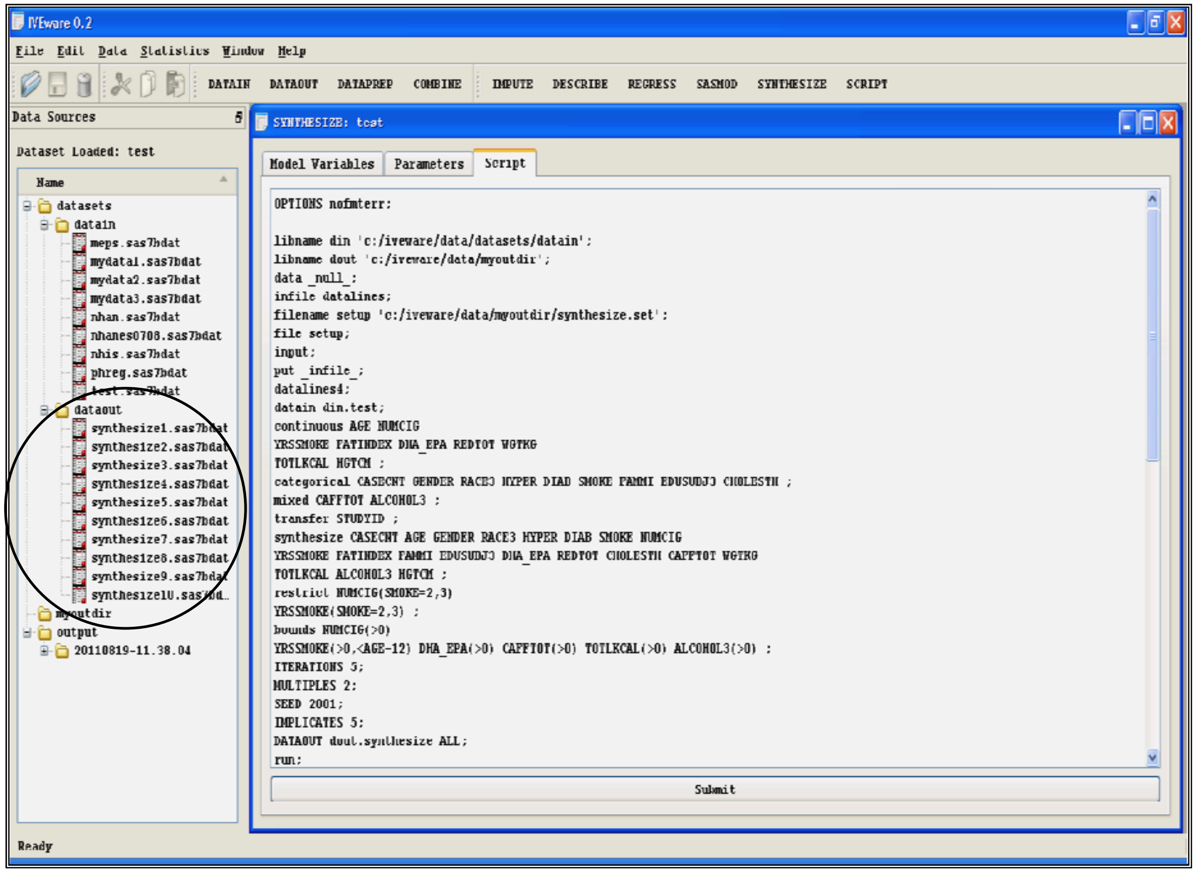


```
OPTIONS nofmterr;

libname din 'c:/iveware/data/datasets/datain';
libname dout 'c:/iveware/data/myoutdir';
data _null_;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/synthesize.set'
file setup;
input;
put  infile ;
datalines4;
datain din.test;
continuous AGE NUMCIG
YRSSMOKE FATINDEX DHA EPA REDTOT WGTKG
TOTLKCAL HGTCM ;
categorical CASECNT GENDER RACE3 HYPER DIAB SMOKE FAMMI
mixed CAFFTOT ALCOHOL3 ;
transfer STUDYID ;
synthesize CASECNT AGE GENDER RACE3 HYPER DIAB SMOKE NUM
YRSSMOKE FATINDEX FAMMI EDUSUBJ3 DHA_EPA REDTOT CHOLESTH
TOTLKCAL ALCOHOL3 HGTCM ;
restrict NUMCIG(SMOKE=2,3)
YRSSMOKE(SMOKE=2,3) ;
bounds NUMCIG(>0)
YRSSMOKE(>0,<AGE-12) DHA_EPA(>0) CAFFTOT(>0) TOTLKCAL(>C
ITERATIONS 5;
MULTIPLES 2;
SEED 2001;
IMPLICATES 5;
DATAOUT dout.synthesize ALL;
run;
```

```
 Number           454          444          898
Minimum             0            0            0
Maximum            98      58.4252           98
   Mean       22.7004      4.96389      13.9309
Std Dev       14.3406      11.4838      15.7389

Variable YRSSMOKE
             Observed      Imputed     Combined
 Number           489          409          898
Minimum             1            0            0
Maximum            63      55.1184           63
   Mean       28.1554      3.30646      16.8378
Std Dev        14.677      9.70002      17.7004

Variable FATINDEX
             Observed      Imputed     Combined
 Number           864           34          898
Minimum            10      15.1888           10
Maximum            33      29.0794           33
   Mean       21.5625      21.9867      21.5786
Std Dev       3.94349       3.8876      3.94008

Variable FAMMI
             Observed      Imputed     Combined
   Code     Freq  Per    Freq  Per    Freq  Per
```

Synthesized Datasets

After a successful submission of the synthesize script, the synthesized datasets can be found in the DATAOUT folder. The example script, calling for two imputations and five implicates, outputted 10 synthesized datasets.

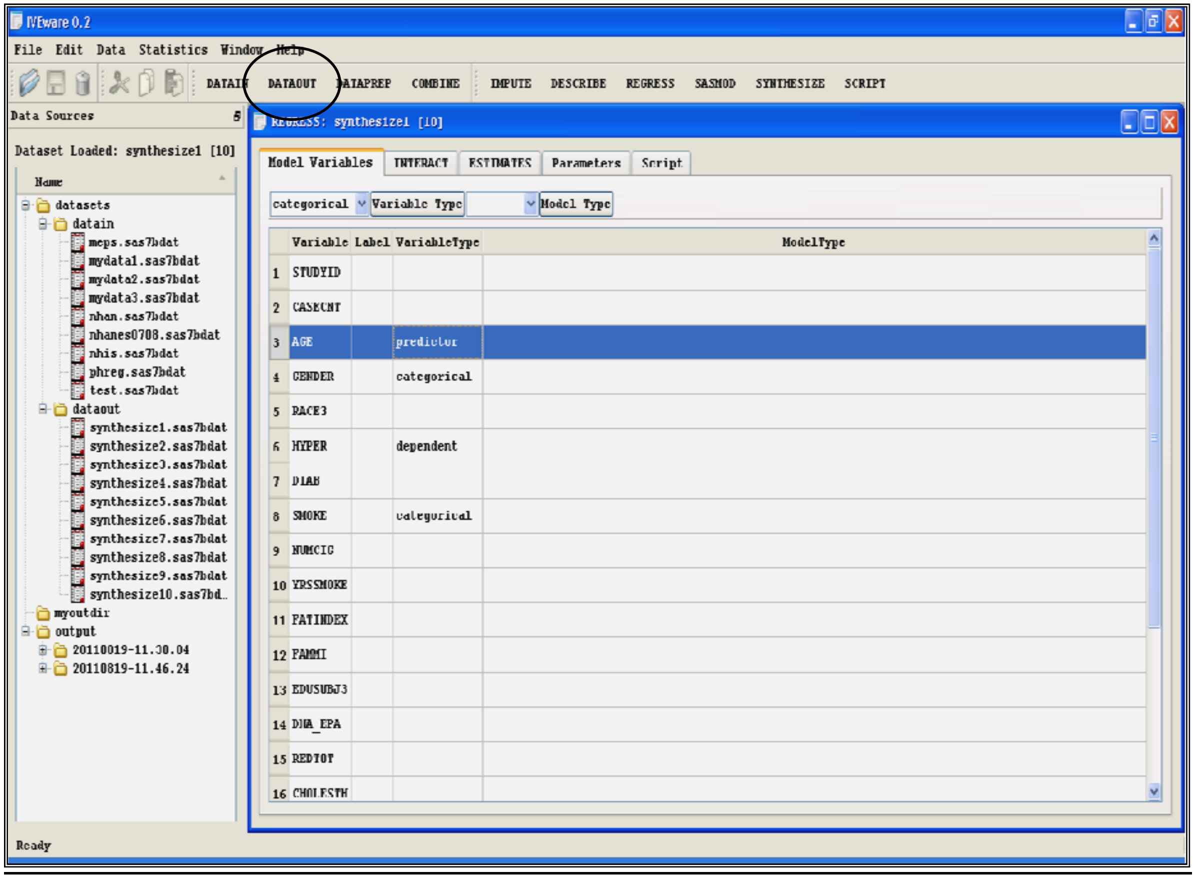


```
OPTIONS nofmterr;

libname din 'c:/iveware/data/datasets/datain';
libname dout 'c:/iveware/data/myoutdir';
data _null_;
infile datalines;
filename setup 'c:/iveware/data/myoutdir/synthesize.set';
file setup;
input;
put _infile_;
datalines4;
datain din.test;
continuous AGE NUMCIG
YRSSMOKE FATINDEX DHA_EPA REDTOT WGTKG
TOTLKCAL HGTCM ;
categorical CASECNT GENDER RACE3 HYPER DIAB SMOKE FAMMI EDUSUDJ3 CHOLESTH ;
mixed CAFFTOT ALCOHOL3 ;
transfer STUDYID ;
synthesize CASECNT AGE GENDER RACE3 HYPER DIAB SMOKE NUMCIG
YRSSMOKE FATINDEX FAMMI EDUSUDJ3 DHA_EPA REDTOT CHOLESTH CAFFTOT WGTKG
TOTLKCAL ALCOHOL3 HGTCM ;
restrict NUMCIG(SMOKE=2,3)
YRSSMOKE(SMOKE=2,3) ;
bounds NUMCIG(>0)
YRSSMOKE(>0,<AGE-12) DHA_EPA(>0) CAFFTOT(>0) TOTLKCAL(>0) ALCOHOL3(>0) ;
ITERATIONS 5;
MULTIPLES 2;
SEED 2001;
IMPLICATES 5;
DATAOUT dout.synthesize ALL;
run;
```

Analyzing Synthesized Datasets

By highlighting the synthesized datasets and clicking the DATAOUT button, the synthesized datasets are loaded and can be used in an *IVEware* analysis procedure.

In the example, synthesize1 through synthesize10 were loaded and the REGRESS procedure opened. The datasets to be analyzed are listed at the top of the regress window. (The loaded message lists only the name of the first synthesized dataset loaded.)



Dataset Loaded: synthesize1 [10]

	Variable	Label	VariableType	ModelType
1	STUDYID			
2	CASECNT			
3	AGE		predictor	
4	GENDER		categorical	
5	RACE3			
6	HYPER		dependent	
7	DIAB			
8	SMOKE		categorical	
9	NUMCIG			
10	YRSSMOKE			
11	FATINDEX			
12	FAMMI			
13	EDUSUBJ3			
14	DHA_EPA			
15	REDTOT			
16	CHOLESTH			

Glossary of Synthesize Commands

Except for the command IMPLICATES which specifies the number synthesized data sets are to be generated, SYNTHESIZE commands are the same as those for Impute.

DECLARING VARIABLE TYPES

SYNTHESIZE requires that the SAS data set variables be defined by type. Six types of variables are recognized by the IMPUTE module: continuous, categorical, count, mixed, transfer and drop. If no variable types are specified, all variables will be assumed to be continuous. Variable types should be declared before any BOUNDS, INTERACT, or RESTRICT statements (see below).

CONTINUOUS variable list;

Variables declared as CONTINUOUS may take on any value on a continuum. Income is an example of a continuous variable. A normal linear regression model is used to synthesize the missing values in these variables. You may want to transform the variable to achieve normality and then SYNTHESIZE on the transformed scale. After imputation you may re-transform the variable back to its original form.

CATEGORICAL variable list;

CATEGORICAL variables have values that represent discrete values. Gender is a categorical variable. A logistic or generalized logistic model is used to impute missing categorical values.

MIXED variable list;

Variables declared as MIXED are both categorical and continuous. In a mixed variable a value of zero is treated as a discrete category, while values greater than zero are considered continuous. Alcohol consumption is an example of a mixed variable. A two stage model is use to impute the missing values. First, a logistic regression model is used to impute zero vs. non-zero status. Conditional on imputing a non-zero status, a normal linear regression model is used to impute non-zero values.

COUNT variable list;

COUNT variables have non-negative integer values. A Poisson regression model is used to impute the missing values. The number of annual doctor visits is an example of a COUNT variable

DROP variable list;

Variables listed after the DROP keyword will be excluded from the imputation procedure and will not appear in the imputed data set.

TRANSFER variable list;

Variables listed after the TRANSFER keyword are carried over to the imputed data set, but are not imputed nor used as predictors in the imputation model. Transfer variables, however, can be used in the RESTRICT and BOUNDS statements (see below). ID is an example of a variable that you might want to treat as a transfer variable.

DEFAULT variable type;

Variable type can be Continuous, Categorical, Count, Mixed, Transfer or Drop. This keyword declares that by default all the variables in the data set should be treated as the **variable type**. The most efficient use of the DEFAULT statement is to declare the most numerous variable type in your data set as the default type, eliminating the need to type a long list of variables.

Optional Statements

RESTRICT variable(logical expression);

This command is used to restrict the imputation of a variable to those observations that satisfy the logical expression. For instance, suppose that the variable **Yrssmoke** indicates the number of years an individual smoked, and the variable **Smoke** takes the value 1 for a current smoker, 2 for a former smoker or 0 for someone who never smoked.

Then the declaration,

```
RESTRICT Yrssmoke(Smoke=1,2);
```

will impute **Yrssmoke** values only for current and former smokers. It will automatically set **Yrssmoke** equal to 0 for never smokers.

Restrictions on more than one variable may be combined as follows:

```
RESTRICT Yrssmoke(Smoke=1,2) Births(Gender=2) Income(Employed=1);
```

When the restriction is not met, the value of the restricted variable will be set to zero for continuous variables and one higher than the highest observed code for categorical variables.

BOUNDS variable (logical expression);

This keyword is useful for restricting the range of values to be imputed for a variable.

For example,

```
BOUNDS Yrssmoke (> 0,<= Age-12);
```

will ensure that the imputed values for **Yrssmoke** are between 0 and the individual's Age minus 12. Smoking is assumed not to begin before the age of 12.

Again, as in the RESTRICT statement more than one variable can be included in the BOUNDS statement.

For example,

```
BOUNDS Yrssmoke (>0,<= Age-12) Numcig(>0);
```

Model-Building Statements

The following commands are useful in the specification of the imputation model.

INTERACT variable*variable;

This keyword enables the users to specify interaction terms to be include in the imputation regression model.

```
INTERACT Income*Income, Age*Race;
```

In this example, the imputation model for all the variables will include a square term for Income and an interaction term of Age and Race.

STEPWISE REGRESSION

MAXPRED number; OR **MAXPRED varlist2 (number);**

Specifies the maximum number of predictor variables to be included as predictors in the regression model. A step-wise regression procedure is used to select the best predictors subject to the maximum number. Setting MAXPRED to a small number of predictors will greatly reduce the computational time especially for a very large data sets but the imputations will not be fully conditional.

For example,

```
MAXPRED 5;
```

will include the five best predictor variables, the five making the largest contribution to the r-square.

You can also restrict the number of predictors for selected variables.

```
MAXPRED Income (7) Educ (3);
```

will limit the number of predictors of Income to the seven largest contributors to the r-square, while the number of predictors of Educ are limited to the three largest contributors. For other variables, all variables will be used as predictors.

MINRSQD decimal;

Specifies the minimum marginal r-squared for a stepwise regression. (Minimum initial marginal r-squared for a logistic regression, and minimum initial r-squares for any code being predicted for a polytomous regression.) This can reduce computation time. A small decimal number like 0.005 would build very large regression models whereas 0.25 will include a smaller number of predictors in the regression models. If neither MAXPRED nor MINRSQD is set then no stepwise regression will be preformed.

```
MINRSQD 0.01;
```

In this example, only variables with minimum additional r-square of 0.01 or higher will be included as predictors.

MAXLOGI number;

Specifies the maximum number of iterative algorithms to be performed in a logistic or multilogit regression model. The default is 50. This is useful if the Newton-Raphson algorithm used in producing maximum likelihood estimates does not converge after 50 iterations. This applies to the convergence criterion for the logistic, polytomous and Poisson regression models. You can check whether you have such a non-convergence problem by inspecting the log file (e.g., mysetup.log).

MINCODI decimal;

Specifies the minimum proportional change in any regression coefficient to continue the logistic regression iteration process. This applies to the convergence criterion for the logistic, polytomous and Poisson regression models.

ITERATIONS number;

Specifies the number of cycles you would like the imputation program to carry out. You can specify any number greater than or equal to 2. Current investigations show that about 10 cycles are sufficient for most imputations. You may want to experiment with several values and check the differences in the resulting analysis.

IMPLICATES number;

Indicates the number of synthesized data sets to be created. By default only a single synthesized data set is generated.

MULTIPLES number;

You can perform an imputation within the SYNTHESIZE procedure. Multiple indicates the number of imputations to be performed. By default only a single imputation is generated. IMPUTE is processed prior to SYNTHESIZE. For each multiple, a set of synthesized data set are created base on the number of implicates specified. For example, if 2 multiples and 5 implicates are specified then 10 synthesized data sets will be created. Five for multiple 1 and 5 for multiple 2;

PERTURB instruction;

The keyword PERTURB followed by an instruction (COEF/SIR) allows the user to control perturbations of imputed values. By default the IMPUTE module will perturb model coefficients using a multivariate normal approximation of the posterior distribution and the predicted values using the appropriate regression model conditional on the perturbed coefficients. This is equivalent to using the COEF instruction. SIR uses the Sampling-Importance-Resampling algorithm to generate coefficients from the actual posterior distribution of parameters in the logistic, polytomous and Poisson regression models (See Rubin 1987a, Raghunathan and Rubin 1988, Raghunathan 1994, Gelman, et. al 1995). This is appropriate in situations where normal approximation to the posterior distribution is not appropriate.

```
PERTURB Sir;
```

SEED number;

Specifies a seed for the random draws from the posterior predictive distribution. **Number** should be greater than zero. A zero seed will result in no perturbations of the predicted values or the regression coefficients. If the SEED keyword is missing from the setup file then the seed will be determined by your computer's internal clock.

NOBS number;

NOBS indicates the number of observations to be used in the analysis. By default all observations in the data set will be used. You might use NOBS to subset a large data set while testing your setup file.

OFFSETS count variables (offset variable) ;

This statement is used to specify an offsets variable when fitting a Poisson regression model.

For example,

```
OFFSETS  Injuries(Years);
```

will fit a model predicting the number for injuries occurring per year.

PRINT instruction;

Indicates the printout desired. The options are STANDARD, DETAILS, COEF, and ALL.

For the IMPUTE procedure, the STANDARD and DETAILS keywords instruct *IVEware* to print the number and distribution of observed values, imputed values, and combined observed and imputed values for each variable.

The keyword COEF instructs IMPUTE to also print the unperturbed and perturbed coefficients for each iteration of each multiple imputation. When the ALL keyword is used, in addition to the above, the coefficient covariance matrix for each iteration of each multiple imputation is also printed.

IMPUTE also printouts a list of the variables used in the imputation model with columns indicating the number of observed cases and the number of imputed cases for each of the variables.

The third column of the variable list, labeled “double counted,” is to be used for diagnostic purposes. This entry should be zero. A non-zero entry indicates that the imputed value of a restricting variable has caused the observed value of a restricted variable to be set to the restricted value (zero for continuous variables, one higher than the highest observed code for categorical variables; see RESTRICT above). This usually indicates a mis-specification of the restriction or an inconsistency in the observed data. In either case, you need to run a data step before the imputation to check the appropriateness of the restriction or correct the data inconsistency.

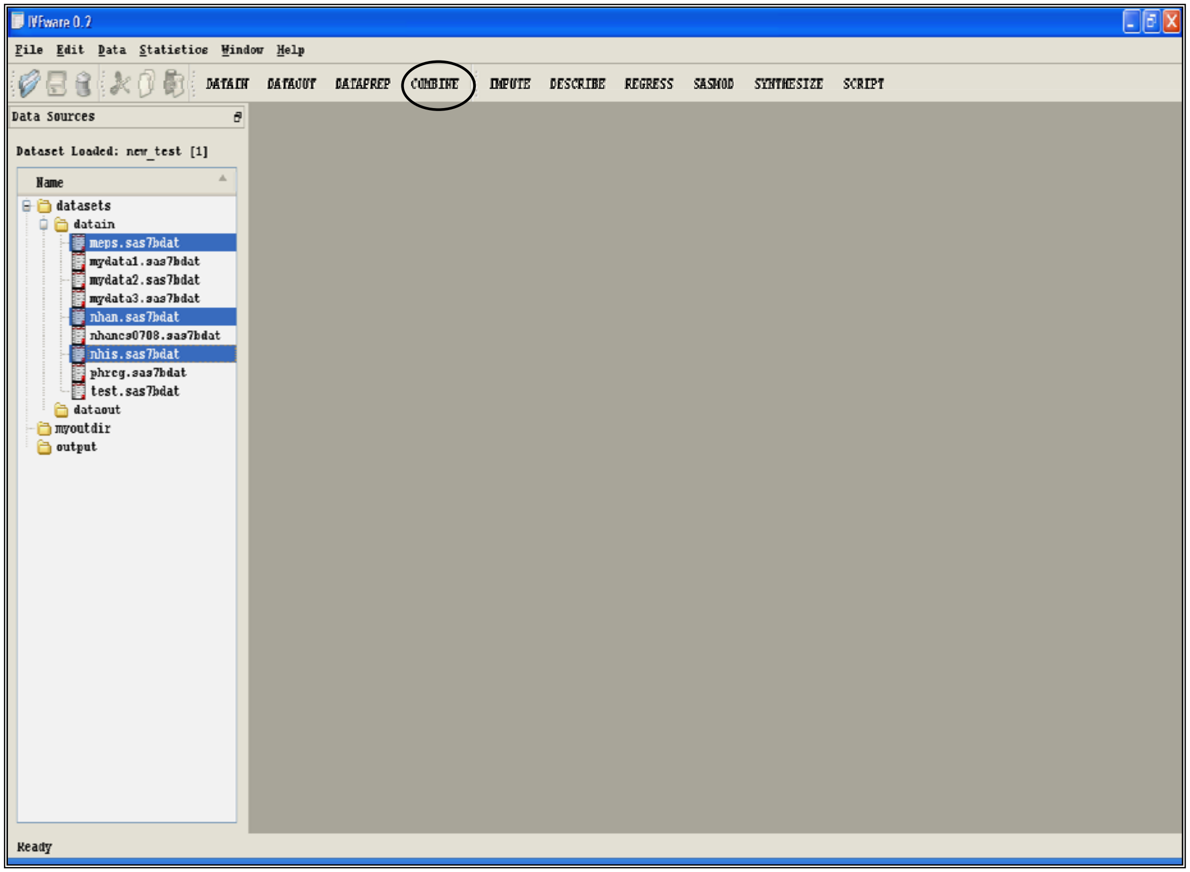
For example, if the variable SMOKE, indicating whether or not a respondent smokes, is missing and the variable YRSMK, indicating the number of years the respondent has smoked, is observed, then logically the respondent should be classified as a smoker. If SMOKE is not given a value indicating the respondent is a smoker in a SAS data step prior to imputation, the missing value could possibly be imputed to a nonsmoker value, causing IMPUTE to change the observed value for YRSMK to zero.

10. The Combine Procedure

Selecting Combine

The Combine data procedure allows the user to concatenate (stack) multiple datasets. The datasets need not contain the same variables. Variables with a shared name will be treated as the same variable in the combined dataset. It's important that they have the same value structure. If a variable does not appear in one of the datasets it is treated as missing cases in the combined dataset. The user may want to impute the missing values prior to analyzing the combine dataset.

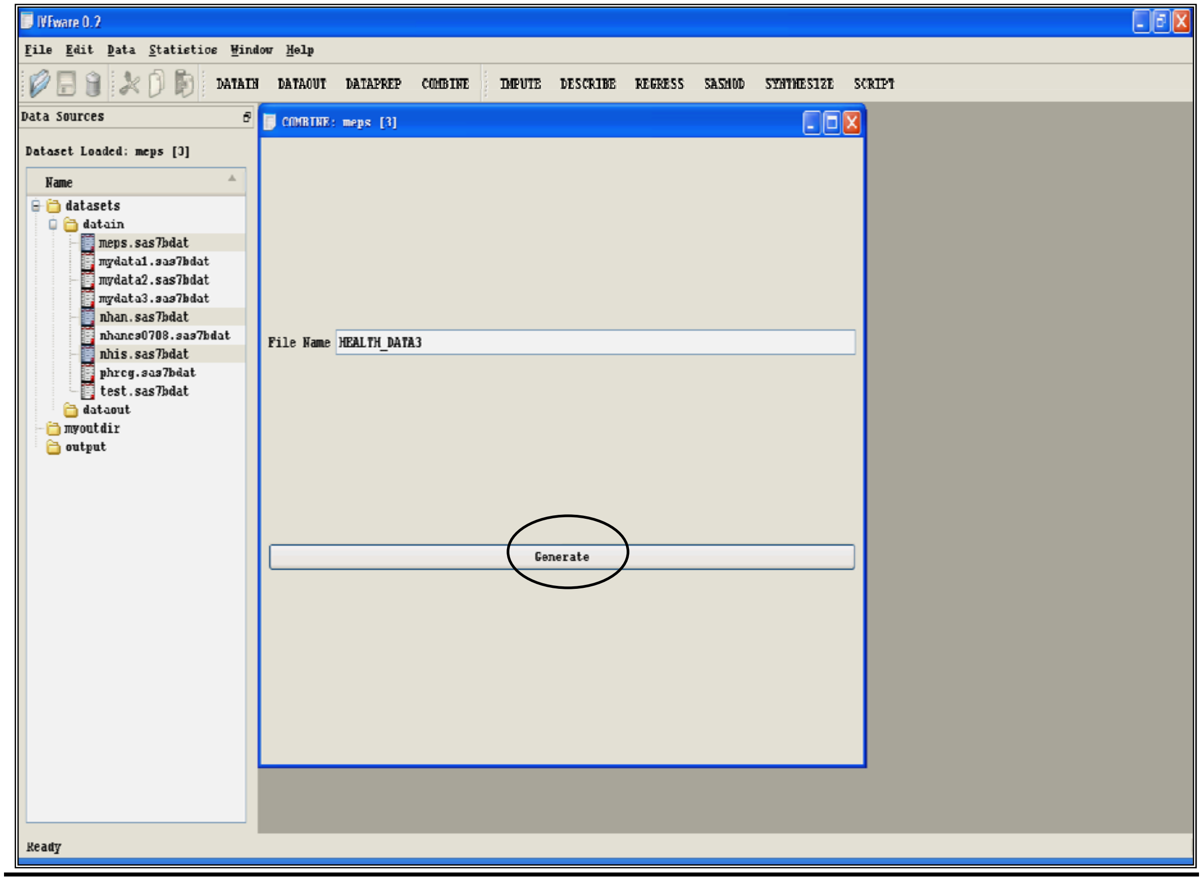
To combine multiple datasets, highlight the datasets in the DATAIN folder and then click on the COMBINE button.



Combine File Name

After clicking the COMBINE button the user can enter the name of the combined file. In the example, the new file, a combination of MEPS, NHAN, and NHIS, will be called HEALTH_DATA3. The default file name is COMBINE with the date and time appended.

Click on the GENERATE bar in the bottom portion of the Combine window to create the new combined dataset.

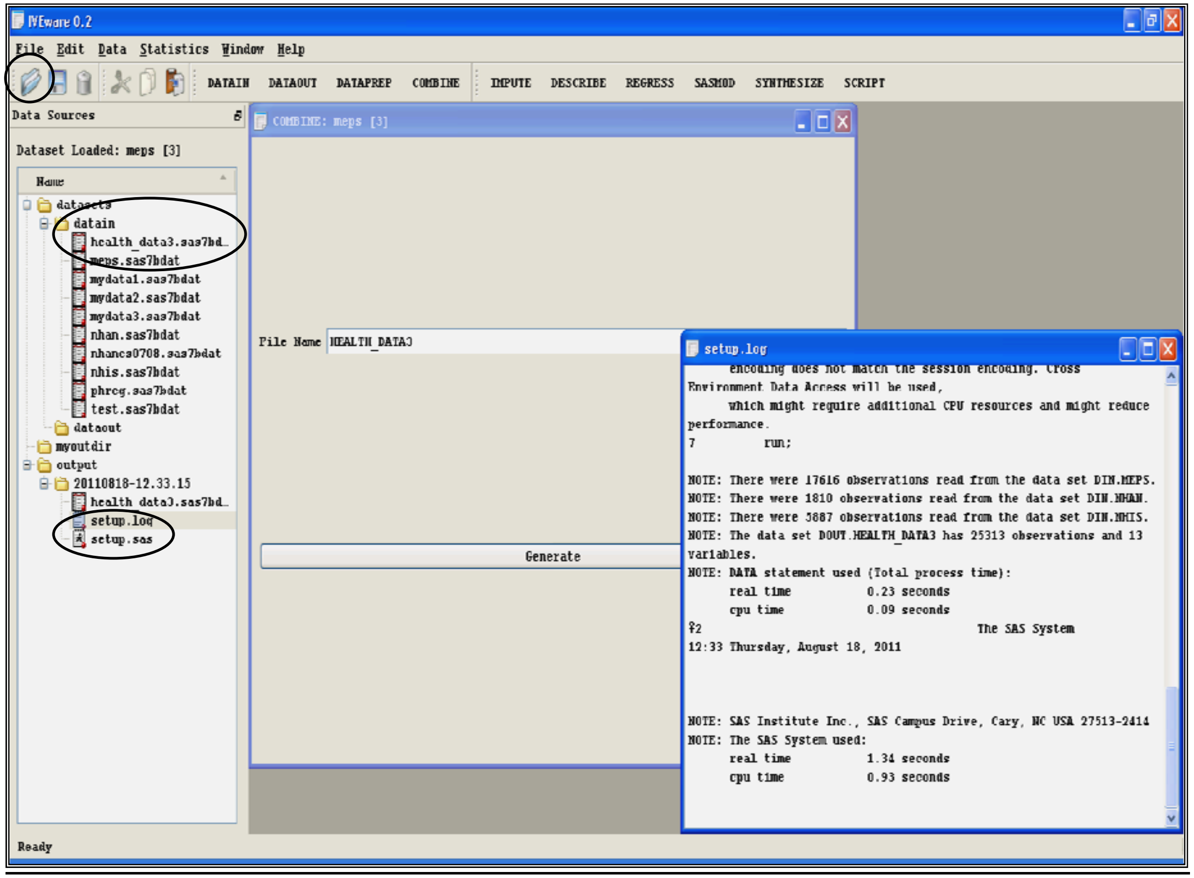


File Name HEALTH_DATA3

Output Files

After the script submission is completed, the OUTPUT folder will contain a sub-folder labeled with the date and time of the script submission. Here you will find the results of the Combine procedure. Errors in the submitted script will be reported in the setup.log file. The output files can be opened by highlighting the file name and clicking on the File icon at the top left of the window. In the example, setup.log was opened in a separate window.

The new combined file, HEALTH_DATA3, is located in the DATAIN folder.



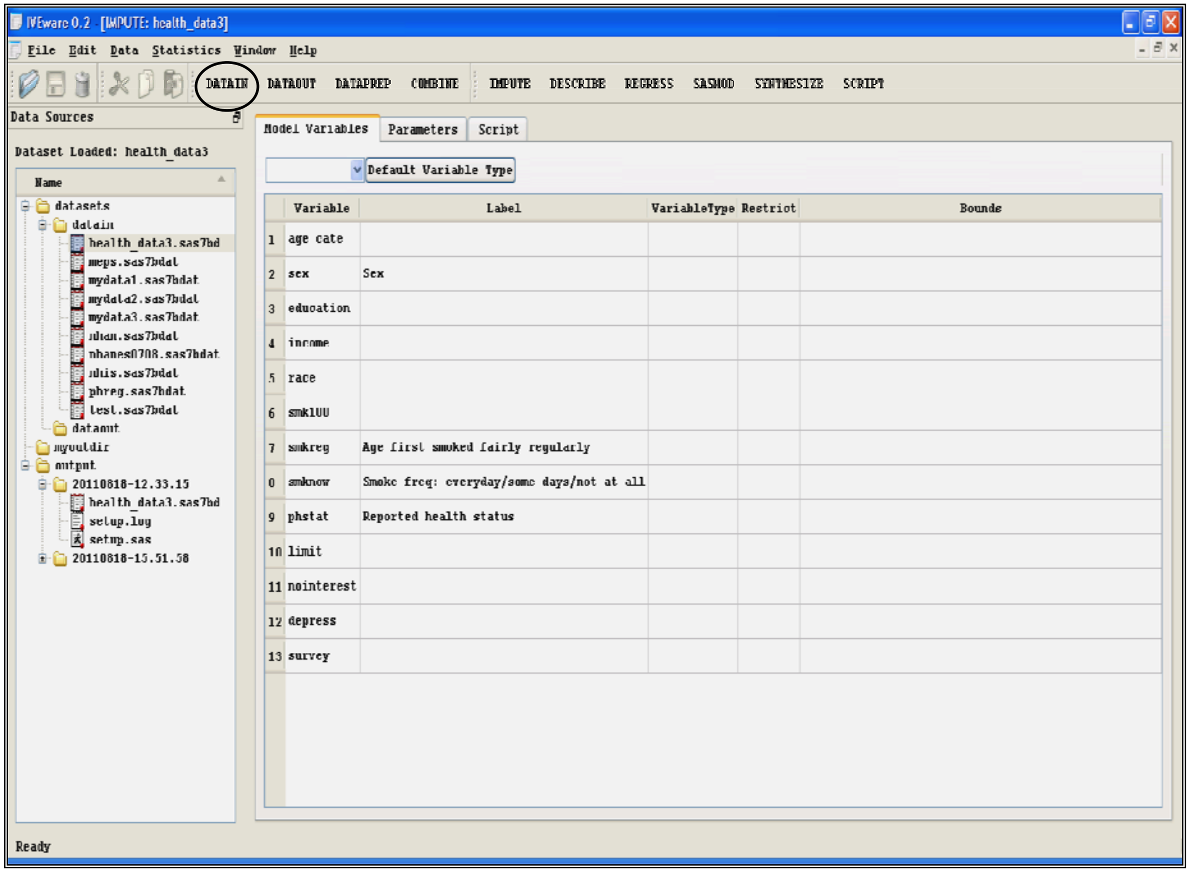
File Name HEALTH_DATA3

```
      encoding does not match the session encoding. Cross
Environment Data Access will be used,
      which might require additional CPU resources and might reduce
performance.
7          run;

NOTE: There were 17616 observations read from the data set DIN.MEPS.
NOTE: There were 1810 observations read from the data set DIN.NHAN.
NOTE: There were 5887 observations read from the data set DIN.NHIS.
NOTE: The data set DOUT.HEALTH_DATA3 has 25313 observations and 13
variables.
NOTE: DATA statement used (Total process time):
      real time           0.23 seconds
      cpu time            0.09 seconds
♀2                                          The SAS System
12:33 Thursday, August 18, 2011



NOTE: SAS Institute Inc., SAS Campus Drive, Cary, NC USA 27513-2414
NOTE: The SAS System used:
      real time           1.34 seconds
      cpu time            0.93 seconds
```

Analyzing the Combined Dataset

By highlighting the new combined dataset and clicking the DATAIN button, the combined dataset is loaded and can be used in an *IVEware* analysis procedure.

In this example, HEALTH_DATA3 was loaded and the IMPUTE procedure opened.


IVEware 0.2 - [IMPUTE: health_data3]
File Edit Data Statistics Window Help
DATAIN DATAOUT DATAPREP COMBINE IMPUTE DESCRIBE REGRESS SASMOD SYNTHESIZE SCRIPT
Data Sources
Model Variables Parameters Script
Default Variable Type
Name

Dataset Loaded: health_data3

- datasets
  - datain
    - health_data3.sas7bd
    - meps.sas7bdat
    - mydata1.sas7bdat
    - mydata2.sas7bdat
    - mydata3.sas7bdat
    - idian.sas7bdat
    - nhanes0708.sas7bdat
    - idis.sas7bdat
    - phreg.sas7bdat
    - test.sas7bdat
  - dataout
- myoutdir
- output
  - 20110818-12.33.15
    - health_data3.sas7bd
    - setup.log
    - setup.sas
  - 20110818-15.51.58

	Variable	Label	VariableType	Restrict	Bounds
1	age cate				
2	sex	Sex			
3	education				
4	income				
5	race				
6	smk100				
7	smkreg	Age first smoked fairly regularly			
8	smknow	Smoke freq: everyday/some days/not at all			
9	phstat	Reported health status			
10	limit				
11	nointerest				
12	depress				
13	survey				

Ready

.