

mi export nhanes1 — Export mi data to NHANES format

Syntax Menu Description Options
 Remarks and examples Also see

Syntax

```
mi export nhanes1 filenamestub [ , options odd_options ]
```

<i>options</i>	Description
<code>replace</code>	okay to replace existing files
<code>uppercase</code>	uppercase prefix and suffix
<code>passiveok</code>	include passive variables

<i>odd_options</i>	Description
<code>nacode(#)</code>	not applicable code; default is 0
<code>obscode(#)</code>	observed code; default is 1
<code>impcode(#)</code>	imputed code; default is 2
<code>impprefix("string" "string")</code>	variable prefix; default is "" ""
<code>impsuffix("string" "string")</code>	variable suffix; default is "if" "mi"

Note: The *odd_options* are not specified unless you want to create results that are nhanes1-like but not really nhanes1 format.

Menu

Statistics > Multiple imputation

Description

`mi export nhanes1` writes the `mi` data in memory to disk files in `nhanes1` format. The files will be named `filenamestub.dta`, `filenamestub1.dta`, `filenamestub2.dta`, and so on. In addition to the variables in the original `mi` data, new variable `seqn` will be added to record the sequence number. After using `mi export nhanes1`, you can use `outfile` (see [\[D\] outfile](#)) or `export delimited` (see [\[D\] import delimited](#)) or a transfer program such as Stat/Transfer to convert the resulting `.dta` files into a format suitable for sending to a non-Stata user. Also see [\[U\] 21 Entering and importing data](#).

`mi export nhanes1` leaves the data in memory unchanged.

Options

`replace` indicates that it is okay to overwrite existing files.

`uppercase` specifies that the new sequence variable `SEQN` and the variable suffixes `IF` and `MI` be in uppercase. The default is lowercase. (More correctly, when generalizing beyond `nhanes1` format, the `uppercase` option specifies that `SEQN` be created in uppercase along with all prefixes and suffixes.)

`passiveok` specifies that passive variables are to be written as if they were imputed variables. The default is to issue an error if passive variables exist in the original data.

`nacode(#)`, `obscode(#)`, and `impcode(#)` are optional and are never specified when reading true `nhanes1` data. The default `nacode(0)` `obscode(1)` `impcode(2)` corresponds to the `nhanes1` definition. These options allow changing the codes for not applicable, observed, and imputed.

`impprefix("string" "string")` and `impsuffix("string" "string")` are optional and are never specified when reading true `nhanes1` data. The default `impprefix("" "")` `impsuffix("if" "mi")` corresponds to the `nhanes1` definition. These options allow setting different prefixes and suffixes.

Remarks and examples

[stata.com](http://www.stata.com)

`mi export nhanes1` is the inverse of `mi import nhanes1`; see [\[MI\] mi import nhanes1](#) for a description of the `nhanes1` format.

Below we use `mi export nhanes1` to convert `miproto.dta` to `nhanes1` format. `miproto.dta` happens to be in wide form, but that is irrelevant.

```
. use http://www.stata-press.com/data/r13/miproto
(mi prototype)
. mi describe
Style:  wide
       last mi update 01feb2013 11:09:26, 2 days ago
Obs.:   complete      1
       incomplete     1 (M = 2 imputations)
       -----
       total          2
Vars.:  imputed:  1; b(1)
       passive:  1; c(1)
       regular:  1; a
       system:   1; _mi_miss
       (there are no unregistered variables)
. list
```

	a	b	c	_1_b	_2_b	_1_c	_2_c	_mi_miss
1.	1	2	3	2	2	3	3	0
2.	4	.	.	4.5	5.5	8.5	9.5	1

```
. mi export nhanes1 mynh, passiveok replace
files mynh.dta mynh1.dta mynh2.dta created
```

```
. use mynh
(mi prototype)
. list
```

	seqn	a	bif	cif
1.	1	1	1	1
2.	2	4	2	2

```
. use mynh1
(mi prototype)
. list
```

	seqn	a	bmi	cmi
1.	1	1	2	3
2.	2	4	4.5	8.5

```
. use mynh2
(mi prototype)
. list
```

	seqn	a	bmi	cmi
1.	1	1	2	3
2.	2	4	5.5	9.5

Also see

[MI] [intro](#) — Introduction to mi

[MI] [mi export](#) — Export mi data

[MI] [mi import nhanes1](#) — Import NHANES-format data into mi