Title

stata.com

cmsample — Display reasons for sample exclusion

Description

`cmsample` displays a table with the reasons why observations in a choice model were dropped from the estimation sample. It also flags choice-model data errors, such as errors in the alternatives variable, dependent variable, or case-specific variables. With the use of its `generate()` option, observations that were dropped or led to an error message can be identified.

Quick start

Tabulate reasons `cmclogit` excluded observations from the estimation sample when fitting a model of `y` on alternative-specific variables `x1`, `x2`, and `x3` and case-specific variables `cv1` and `cv2`

```
cmclogit y x1 x2 x3, casevars(cv1 cv2)
cmsample x1 x2 x3, choice(y) casevars(cv1 cv2)
```

Create a variable named `problem` identifying categories of problems in the choice variable `y`, and list the problem observations

```
cmsample, choice(y) generate(problem)
list y if problem != 0
```

Replace `problem` with categories of problems in case-specific variables `cv1` and `cv2`, and list the problem observations

```
cmsample, casevars(cv1 cv2) generate(problem, replace)
list cv1 cv2 if problem != 0
```

Use alternativewise rather than casewise deletion of observations with missing values

```
cmsample x1 x2 x3, choice(y) casevars(cv1 cv2) altwise
```

Allow `y` to be ranks instead of a 0/1 variable (the default)

```
cmsample x1 x2 x3, choice(y) casevars(cv1 cv2) ranks
```

Menu

Statistics > Choice models > Setup and utilities > Display reasons for sample exclusion

Syntax

```
cmsample [varlist] [if] [in] [weight] [, options]
```

varlist is a list of alternative-specific numeric variables.

options	Description
Main	
`choice(`*choicevar*`)`	0/1 variable indicating the chosen alternatives
`casevars(`*varlist*$_c$`)`	case-specific variables
`altwise`	use alternativewise deletion instead of casewise deletion
`ranks`	allow *choicevar* to be ranks
`generate(`*newvar*`[ , replace ])`	create new variable containing reasons for omitting observations and for error messages; optionally replace existing variable

You must `cmset` your data before using `cmsample`; see [CM] **cmset**.

varlist and *varlist*$_c$ may contain factor variables; see [U] **11.4.3 Factor variables**.

`collect` is allowed; see [U] **11.1.10 Prefix commands**.

`fweights`, `iweights`, and `pweights` are allowed; see [U] **11.1.6 weight**. Weights are checked for missing values and other errors but are not used for the tabulation.

Options

Main

`choice(`*choicevar*`)` specifies a 0/1 variable indicating the chosen alternatives. Typically, it is a dependent variable in the choice model.

`casevars(`*varlist*$_c$`)` specifies case-specific numeric variables. These are variables that are constant for each case.

`altwise` specifies that alternativewise deletion be used when omitting observations due to missing values in your variables. The default is to use casewise deletion; that is, the entire group of observations making up a case is omitted if any missing values are encountered. This option does not apply to observations that are excluded by the `if` or `in` qualifier; these observations are always handled alternativewise regardless of whether `altwise` is specified.

`ranks` allows *choicevar* to be ranks. Any numeric value in *choicevar* is allowed.

`generate(`*newvar*`[ , replace ])` creates a new variable containing categories of the reasons for omitting observations and for error messages. The variable *newvar* is numeric and potentially valued 0, 1, 2, . . . , 16. The value 0 indicates the observation is to be included in the estimation sample. The values 1–16 indicate cases and observations that were either marked out or would generate error messages. See the table of reasons for omitting observations and error messages for the list of values and their meaning. Specifying `replace` allows any existing variable named *newvar* to be replaced.

Remarks and examples

stata.com

`cm` commands have special data requirements. `cmsample` lets you see whether the requirements are met and, if not, to identify the problems. `cmsample` can also flag cases with missing values. It can be used to check your data before running `cm` estimation commands, or it can be used after estimation commands to diagnose why cases and observations were excluded from the estimation sample.

Data for all cm commands must be arranged in long form with each case consisting of multiple Stata observations, one Stata observation for each alternative. There must be a case ID variable identifying cases for cross-sectional data. For panel data, there must be a panel ID variable and a time variable that together uniquely identify cases.

Most choice models are formulated with specified alternatives, and for these models, there must be a variable identifying the alternatives available to each case. For discrete choice models, the dependent variable must be a 0/1 variable representing a single chosen alternative within a case.

cm commands check whether the specified variables meet their data requirements and, when a check fails, respond in different ways depending on the problem. Sometimes an error message is issued. For example,

```
at least one choice set has more than one instance of the same alternative
r(459);
```

Sometimes cases with problems are simply dropped from the estimation sample, and a message is displayed. For example,

```
note: 2 cases (6 obs) dropped due to varlist having no positive
      outcome per case.
```

A researcher may want to look at the problematic observations to see whether the problem is caused by bad values in the data that can be fixed or simply to see and understand the problem. cmsample is a tool for doing this.

All cm commands, including cmsample, require your data to be cmset, and you may get an error message when cmsetting your data. You can identify the problematic observations by using the force option with cmset and then typing cmsample. See example 1.

If your data have missing values and you run a cm estimator, you might be surprised by how many cases were omitted from the estimation sample. By default, a missing value in any observation of a case will cause the entire case to be "marked out"—Stata talk for dropping observations from the estimation sample. This is called casewise deletion of observations. Dropping the whole case makes sense because a case is a single statistical observation. Stata found a problem with the data in this statistical observation, and Stata dropped it. The statistical observation consisted of multiple Stata observations, so multiple Stata observations were dropped, even though there may have been only one missing value in one Stata observation.

cmsample and all the other cm commands have the option altwise, which deletes only missing Stata observations from the sample, rather than deleting entire cases. It is called altwise, short for alternativewise. altwise deletion causes only the alternative containing the missing value to be dropped. The case remains in the sample; only it now lacks one or more of its alternatives and any data associated with that alternative. There may be consequences from this. For example, data with balanced choice sets may become unbalanced. With the default casewise deletion, balanced data always stays balanced. For an example of altwise deletion, see example 2.

By default, cmsample displays a table, but its real utility is its generate() option, which creates a variable with categories of the reasons observations were marked out or gave an error message. This variable can then be used to produce a listing of the problematic observations.

The table below gives cmsample's categories of reasons for deleting observations and error message possibilities.

Value	Description	Error
0	observations included	
1	if or in exclusion	
2	case ID variable missing	
3	time variable missing	
4	alternatives variable missing	
5	*varlist* missing	
6	weight missing	
7	casevars(*varlist*$_c$) missing	
8	choice variable missing	
9	choice sets of size one	
10	choice variable not 0/1	error (unless ranks specified)
11	choice variable all 0	
12	choice variable all 1	
13	choice variable multiple 1s	
14	weight not constant within case	error
15	repeated alternatives within case	error
16	casevars(*varlist*$_c$) not constant within case	error

Possibilities labeled as "error" in the table will not cause cmsample to end with an error message, but any other cm command will.

The possibilities are ordered in roughly the order that the cm estimators use when omitting observations. For example, if the choice variable is not 0/1 for the same case in which the alternatives variable has repeated values, it will be categorized as the former, not the latter—and other cm commands will complain about the former, not the latter, as well. Possibilities can be isolated simply by specifying fewer variables to cmsample.

▷ Example 1: Casewise (default) deletion of missing values

We have created a dataset with lots of missing values and several errors. Let's load it into memory and try to cmset it.

```
. use https://www.stata-press.com/data/r17/carchoice_missing
(Car choice data with missing values)

. cmset consumerid car
note: 1 case ignored because it has only one alternative.
note: 11 cases have missing values of car.
at least one choice set has more than one instance of the same alternative
r(459);
```

cmset gives an error message. But as we mentioned earlier, the thing to do now is to use the force option with cmset.

```
. cmset consumerid car, force
note: 1 case ignored because it has only one alternative.
note: 11 cases have missing values of car.
note: at least one choice set has more than one instance of the same
      alternative.

     Case ID variable: consumerid
Alternatives variable: car
```

The `force` option lets you `cmset` your data, but it only puts off the point at which you get an error message. If you try running a `cm` estimator after using `force` with `cmset`, the `cm` estimator will issue an error message (unless you use an `if` restriction to avoid the problematic observations).

`cmsample` will show you reasons for any error messages plus reasons why cases were dropped. Note that the syntax of `cmsample` is almost the same as that of a `cm` estimator. You put the case-specific variables in `casevars()`. Weights and `if` or `in` restrictions follow standard Stata syntax. The only difference is that you put the dependent variable in the `choice()` option.

```
. cmsample dealers [fw=weight] if income > 20.5,
> choice(purchase) casevars(i.gender income)
```

Reason for exclusion	Freq.	Percent	Cum.
observations included	2,812	89.04	89.04
if or in exclusion	79	2.50	91.55
caseid variable missing	7	0.22	91.77
alternatives variable missing	39	1.23	93.00
varlist missing	12	0.38	93.38
weight missing	25	0.79	94.17
casevars missing	112	3.55	97.72
choice variable missing	38	1.20	98.92
choice sets of size one	1	0.03	98.96
choice variable not 0/1*	4	0.13	99.08
choice variable all 0	6	0.19	99.27
choice variable all 1	4	0.13	99.40
choice variable multiple 1s	4	0.13	99.53
weight not constant within case*	4	0.13	99.65
repeated alternatives within case*	4	0.13	99.78
casevars not constant within case*	7	0.22	100.00
Total	3,158	100.00	

```
* indicates an error
```

As we said, we created a dataset with lots of problems! However, it is easy to track them down. The table indicates that there are several problems with the choice variable, the variable `purchase` in the `choice()` option. The variable `purchase` is supposed to be a 0/1 variable, intended to be our dependent variable in our choice model. To focus on the problems with this variable, we run `cmsample` again with just `choice(purchase)` and add the option `generate(flag)` to create a variable `flag` that identifies the errors.

```
. cmsample, choice(purchase) gen(flag)
```

Reason for exclusion	Freq.	Percent	Cum.
observations included	3,045	96.42	96.42
caseid variable missing	7	0.22	96.64
alternatives variable missing	39	1.23	97.88
choice variable missing	42	1.33	99.21
choice sets of size one	1	0.03	99.24
choice variable not 0/1*	4	0.13	99.37
choice variable all 0	8	0.25	99.62
choice variable all 1	4	0.13	99.75
choice variable multiple 1s	4	0.13	99.87
repeated alternatives within case*	4	0.13	100.00
Total	3,158	100.00	

* indicates an error

`cmsample` will always show you problems with the case ID variable and the alternatives variable. Suppose we do not care about them for now. Nor do we care about the missing values in the choice variable `purchase`. But we care about improper values in `purchase`. These are `cmsample`'s categories 9–13. So we list the observations for which `flag` has these values. We also include the case ID variable `consumerid` and the alternatives variable `car` in the listing.

```
. sort flag consumerid car
. list consumerid car purchase flag if inrange(flag, 9, 13),
> sepby(consumerid) abbr(10)
```

	consumerid	car	purchase	flag
3134.	101	American	0	choice sets of size one
3135.	80	American	0	choice variable not 0/1*
3136.	80	Japanese	2	choice variable not 0/1*
3137.	80	European	0	choice variable not 0/1*
3138.	80	Korean	0	choice variable not 0/1*
3139.	250	American	0	choice variable all 0
3140.	250	Japanese	0	choice variable all 0
3141.	250	European	0	choice variable all 0
3142.	250	Korean	0	choice variable all 0
3143.	540	Japanese	0	choice variable all 0
3144.	540	European	0	choice variable all 0
3145.	639	American	0	choice variable all 0
3146.	639	European	0	choice variable all 0
3147.	201	American	1	choice variable all 1
3148.	201	Japanese	1	choice variable all 1
3149.	201	European	1	choice variable all 1
3150.	201	Korean	1	choice variable all 1
3151.	202	American	0	choice variable multiple 1s
3152.	202	Japanese	1	choice variable multiple 1s
3153.	202	European	1	choice variable multiple 1s
3154.	202	Korean	1	choice variable multiple 1s

These are the cases we may wish to examine for possible data errors. Only the value of `purchase` $= 2$ in observation number 3136 must be fixed (or omitted with an `if` restriction). A value that is not

0, 1, or missing for a choice variable gives an error message. The other problematic observations are automatically dropped from the estimation sample, so they can be left in the dataset, and the estimation command will run.

◁

▷ Example 2: altwise handling of missing values

Let's illustrate the difference between casewise deletion (the default) and alternativewise deletion, which is done when you specify the `altwise` option.

We load the choice model dataset used in example 1 of [CM] **cmclogit**. The alternative-specific variable `dealers` has no missing values. Let's randomly change 5% of its values to missing.

```
. use https://www.stata-press.com/data/r17/carchoice, clear
(Car choice data)
. cmset consumerid car
note: alternatives are unbalanced across choice sets; choice sets of
      different sizes found.
     Case ID variable: consumerid
Alternatives variable: car
. set seed 1
. replace dealers = . if runiform() < 0.05
(153 real changes made, 153 to missing)
```

We first run `cmsample` with the default casewise deletion:

```
. cmsample dealers, choice(purchase)
```

Reason for exclusion	Freq.	Percent	Cum.
observations included	2,638	83.48	83.48
varlist missing	522	16.52	100.00
Total	3,160	100.00	

Now with `altwise` deletion:

```
. cmsample dealers, choice(purchase) altwise gen(flag)
```

Reason for exclusion	Freq.	Percent	Cum.
observations included	2,898	91.71	91.71
varlist missing	153	4.84	96.55
choice sets of size one	3	0.09	96.65
choice variable all 0	106	3.35	100.00
Total	3,160	100.00	

Note that we randomly created 153 missing values; that's $153/3160 = 4.8\%$ of the total observations. The casewise deletion excluded 16.5% of the dataset's observations (522 out of 3160), whereas the `altwise` deletion excluded only 8.3% ($153 + 3 + 106 = 262$ out of 3160).

The `altwise` deletion introduced problems into the data. Now there are choice sets of size 1 and cases in which the choice variable is all 0. The casewise marked-out sample had only two distinct choice sets: $(1, 2, 3)$ and $(1, 2, 3, 4)$. After `altwise` deletion, there are nine choice sets. We can see the choice sets by typing `cmchoiceset` with the restriction `if flag == 0`, which includes only observations that would remain in an estimation sample.

```
. cmchoiceset if flag == 0
Tabulation of choice-set possibilities
```

Choice set	Freq.	Percent	Cum.
1 2	17	2.02	2.02
1 2 3	349	41.55	43.57
1 2 3 4	412	49.05	92.62
1 2 4	19	2.26	94.88
1 3	10	1.19	96.07
1 3 4	9	1.07	97.14
2 3	6	0.71	97.86
2 3 4	17	2.02	99.88
3 4	1	0.12	100.00
Total	840	100.00	

```
Note: Total is number of cases.
```

Be aware of the consequences of `altwise` deletion of missing values. ◁

Stored results

`cmsample` stores the following in `r()`:

Scalars

`r(N)`	number of observations both included and excluded
`r(N_included)`	number of observations included
`r(nc_included)`	number of cases included
`r(N_ex_if_in)`	number of observations excluded by `if` or `in`
`r(N_ex_caseid)`	number of observations excluded: case ID variable missing
`r(N_ex_size_1)`	number of observations excluded because of choice sets of size 1
`r(nc_ex_size_1)`	number of cases excluded because of choice sets of size 1
`r(N_ex_altvar)`	number of observations excluded because alternatives variable missing
`r(nc_ex_altvar)`	number of cases excluded because alternatives variable missing
`r(N_err_altvar)`	number of observations with repeated alternatives within case (error)
`r(nc_err_altvar)`	number of cases with repeated alternatives within case (error)
`r(N_ex_varlist)`	number of observations excluded because *varlist* missing
`r(nc_ex_varlist)`	number of cases excluded because *varlist* missing
`r(N_ex_wt)`	number of observations excluded because weight missing
`r(nc_ex_wt)`	number of cases excluded because weight missing
`r(N_err_wt_nc)`	number of observations with weight not constant within case (error)
`r(nc_err_wt_nc)`	number of cases with weight not constant within case (error)
`r(N_ex_choice)`	number of observations excluded because *choicevar* missing
`r(nc_ex_choice)`	number of cases excluded because *choicevar* missing
`r(N_ex_choice_0)`	number of observations excluded because *choicevar* all 0 for case
`r(nc_ex_choice_0)`	number of cases excluded because *choicevar* all 0 for case
`r(N_ex_choice_1)`	number of observations excluded because *choicevar* all 1 for case
`r(nc_ex_choice_1)`	number of cases excluded because *choicevar* all 1 for case
`r(N_ex_choice_011)`	number of observations excluded because *choicevar* has multiple 1s for case
`r(nc_ex_choice_011)`	number of cases excluded because *choicevar* has multiple 1s for case
`r(N_err_choice)`	number of observations with *choicevar* not 0/1 (error)
`r(nc_err_choice)`	number of cases with *choicevar* not 0/1 (error)
`r(N_ex_casevar)`	number of observations excluded because *casevars* missing
`r(nc_ex_casevar)`	number of cases excluded because *casevars* missing
`r(N_err_casevar_nc)`	number of observations with *casevars* not constant within case (error)
`r(nc_err_casevar_nc)`	number of cases with *casevars* not constant within case (error)
`r(N_ex_time)`	number of observations excluded because *timevar* missing

Macros

`r(caseid)`	name of case ID variable
`r(altvar)`	name of alternatives variable (if set)
`r(timevar)`	name of time variable (if panel data)
`r(marktype)`	`casewise` or `altwise`, type of markout

Also see

[CM] **cmchoiceset** — Tabulate choice sets

[CM] **cmset** — Declare data to be choice model data

[CM] **cmsummarize** — Summarize variables by chosen alternatives

[CM] **cmtab** — Tabulate chosen alternatives