cmchoiceset — Tabulate choice sets

Description Quick start Menu Syntax
Options Remarks and examples Stored results Also see

Description

`cmchoiceset` tabulates choice sets for choice data. It is useful when choice sets are unbalanced, that is, when alternatives are not the same for every case.

Quick start

One-way tabulation of choice sets for `cmset` data

```
cmchoiceset
```

Same as above, but omit missing values of the alternatives variable observation by observation rather than casewise (the default)

```
cmchoiceset, altwise
```

One-way tabulation of the size of the choice sets

```
cmchoiceset, size
```

Two-way tabulation of choice sets versus the case-specific variable `x`

```
cmchoiceset x
```

Same as above, but `x` is not a case-specific variable, and tabulation will be by observations, not cases

```
cmchoiceset x, observations
```

Generate a variable with categories for the choice-set patterns

```
cmchoiceset, generate(cvar)
```

For panel choice data, display a two-way tabulation of choice sets versus the time variable

```
cmchoiceset, time
```

Menu

Statistics > Choice models > Setup and utilities > Tabulate choice sets

Syntax

```
cmchoiceset [varname] [if] [in] [, options]
```

options	Description
Main	
`size`	tabulate size of choice sets
`observations`	tabulate by observations, not cases; the default
`altwise`	use alternativewise deletion instead of casewise deletion
`transpose`	transpose rows and columns in two-way tables
`missing`	include missing values of *varname* in tabulation
`time`	tabulate choice sets versus time variable (only for panel CM data)
`generate(`*newvar*`, ...)`	create new variable containing categories for the choice-set patterns
Options	
tab1_options	options for one-way tables
tab2_options	options for two-way tables

tab1_options	Description
`sort`	display table in descending order of frequency

tab2_options	Description
`column`	report column percentages
`row`	report row percentages
`cell`	report cell percentages
`rowsort`	list rows in order of observed frequency
`colsort`	list columns in order of observed frequency
`[no]key`	report or suppress cell contents key

You must `cmset` your data before using `cmchoiceset`; see [CM] **cmset**.
by is allowed; see [D] **by**.

Options

Main

`size` tabulates the size of the choice sets rather than the choice-set patterns.

`observations` specifies that the tabulation be done by observations instead of by cases, which is the default. If *varname* is specified and *varname* is a case-specific variable (values constant within case), a tabulation of choice sets versus *varname* by cases is displayed by default. If *varname* is not a case-specific variable, a tabulation by cases cannot be produced, so the option `observations` must be specified; otherwise, an error message is given.

`altwise` specifies that alternativewise deletion be used when omitting observations because of missing values in the alternatives variable or *varname*. 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 or the `by` prefix; these observations are always handled alternativewise regardless of whether `altwise` is specified.

`transpose` transposes rows and columns in displays of two-way tables.

`missing` specifies that the missing values of *varname* be treated like any other value of *varname*.

`time` tabulates choice sets versus the time variable when data are panel choice data. See [CM] **cmset**.

`generate(`*newvar* [`, replace label(`*lblname*`)`]`)` creates a new variable containing categories for the choice-set patterns. The variable *newvar* is numeric and valued 1, 2, ... Its value label contains the choice-set patterns as strings. If option `size` was specified, then *newvar* contains the sizes of the choice sets.

`replace` allows any existing variable named *newvar* to be replaced.

`label(`*lblname*`)` specifies the name of the value label created when `generate(`*newvar*`)` is specified. By default, the variable name *newvar* is also used for the name of the value label.

Options

`sort` puts the table in descending order of frequency in a one-way table.

`column` displays the relative frequency, as a percentage, of each cell within its column in a two-way table.

`row` displays the relative frequency, as a percentage, of each cell within its row in a two-way table.

`cell` displays the relative frequency, as a percentage, of each cell in a two-way table.

`rowsort` and `colsort` specify that the rows and columns, respectively, be presented in order of observed frequency in a two-way table.

[`no`]`key` displays or suppresses a key above two-way tables. The default is to display the key if more than one cell statistic is requested. `key` displays the key. `nokey` suppresses its display.

Remarks and examples

`cmchoiceset` is useful when choice sets are unbalanced, meaning different cases have different sets of alternatives. For balanced choice sets—when every case has the same set of alternatives—this command merely tells you every choice set is the same.

In particular, `cmchoiceset, generate(`*newvar*`)` can be useful when using the postestimation command `margins` for unbalanced designs. The variable *newvar* can be used with `margins`'s options `over()` or `subpop()`. This allows you to look at adjusted predictions, expected probabilities, and marginal effects grouped by the different choice sets. See example 3 below and [CM] **margins** for details.

Example 1: Cross-sectional choice data, one-way tables

Here is an example with cross-sectional choice data. First, we `cmset` our data. The variable `consumerid` is our case ID, and the variable `car` gives the alternatives.

```
. use https://www.stata-press.com/data/r19/carchoice
(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
```

`cmset` tells us the choice sets are unbalanced. To see the choice sets, we type `cmchoiceset`:

```
. cmchoiceset
Tabulation of choice-set possibilities
```

Choice set	Freq.	Percent	Cum.
1 2 3	380	42.94	42.94
1 2 3 4	505	57.06	100.00
Total	885	100.00	

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

The majority of choice sets are 1, 2, 3, 4, and the remaining ones are 1, 2, 3—missing alternative 4.

To see the correspondence between numeric values of alternatives and their labels, we list the value label of the alternatives variable `car`.

```
. describe car
```

Variable name	Storage type	Display format	Value label	Variable label
car	byte	%9.0g	nation	Nationality of car

```
. label list nation
nation:
           1 American
           2 Japanese
           3 European
           4 Korean
```

We see that alternative 4 is Korean automobiles. This is the alternative that some consumers do not have.

To get a tabulation by observations rather than by cases, we use the `observations` option.

```
. cmchoiceset, observations
Tabulation of choice-set possibilities
```

Choice set	Freq.	Percent	Cum.
1 2 3	1,140	36.08	36.08
1 2 3 4	2,020	63.92	100.00
Total	3,160	100.00	

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

▷ Example 2: Cross-sectional choice data, two-way tables

If you suspect that there is a relationship between the choice set and some variable in your dataset, you can examine a two-way tabulation. Here we tabulate the choice sets versus `gender`, which is a case-specific variable, meaning that it is constant within each case.

```
. cmchoiceset gender
Tabulation of choice-set possibilities by gender
```

Choice set	Gender: 0 = Female, 1 = Male Female	Male	Total
1 2 3	102	271	373
1 2 3 4	134	355	489
Total	236	626	862

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

We notice that this tabulation has only 862 cases, whereas the earlier one had 885 cases. The variable `gender` must have missing values. Are the observations with missing values related to the choice sets? We can look at this by specifying the options `missing` and `observations`.

```
. cmchoiceset gender, missing observations
Tabulation of choice-set possibilities by gender
```

Choice set	Gender: 0 = Female, 1 = Male Female	Male	.	Total
1 2 3	306	827	7	1,140
1 2 3 4	548	1,456	16	2,020
Total	854	2,283	23	3,160

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

Note that we did this tabulation by observations, not cases. If we omit the option `observations`, we get an error message:

```
. cmchoiceset gender, missing
casevar not constant within case
    Casevar gender is not constant within case for 23 cases (85 obs).
    Use option observations when gender is not a casevar.
r(459);
```

By default, `cmchoiceset` considers any *varname* passed as an argument to be a case-specific variable. The variable `gender` is a case-specific variable when cases with any missing values are omitted. But if you treat missing values like any other value, then `gender` is not a case-specific variable because when there are missing values, the missing values are not found in every observation of the case.

If you want to examine missing values in choice data, you may find the `cmsample` command useful.

The altwise option handles missing values differently. This option omits observations with missing values for *varname* (or the alternatives variable) and then creates choice sets based on the remaining observations.

```
. cmchoiceset gender, altwise
Tabulation of choice-set possibilities by gender
```

Choice set	Gender: 0 = Female, 1 = Male Female	Male	Total
1 2	0	2	2
1 2 3	104	274	378
1 2 3 4	134	355	489
1 2 4	0	4	4
1 3	0	2	2
1 3 4	1	1	2
2 3	0	3	3
2 3 4	1	4	5
Total	240	645	885

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

Using altwise with these data creates several additional choice sets. When we use a cm estimator with the option altwise and have variables with missing values, the same thing can happen. Here is an example:

```
. cmclogit purchase, casevar(i.gender) altwise
  (output omitted)
. cmchoiceset if e(sample) == 1
Tabulation of choice-set possibilities
```

Choice set	Freq.	Percent	Cum.
1 2	2	0.23	0.23
1 2 3	378	42.71	42.94
1 2 3 4	489	55.25	98.19
1 2 4	4	0.45	98.64
1 3	2	0.23	98.87
1 3 4	2	0.23	99.10
2 3	3	0.34	99.44
2 3 4	5	0.56	100.00
Total	885	100.00	

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

The altwise option with the estimator cmclogit and the *casevar* gender creates the same choice sets as the altwise option does with cmchoiceset gender. Before using the option altwise with a cm estimator, we may want to think whether it is appropriate. In this example, it means treating 2 cases as if their only available alternatives were 1 or 2, 4 cases as if their only available alternatives were 1, 2, or 4, etc.

When doing a tabulation of choice sets versus a variable with many values, the option transpose is helpful.

```
. cmchoiceset dealers, observations missing transpose
Tabulation of dealers by choice-set possibilities
```

No. of dealerships in community	Choice set 1 2 3	1 2 3 4	Total
0	0	2	2
1	28	372	400
2	135	247	382
3	155	273	428
4	107	186	293
5	121	157	278
6	132	141	273
7	84	145	229
8	107	136	243
9	113	150	263
10	90	134	224
11	45	53	98
12	17	16	33
13	6	8	14
Total	1,140	2,020	3,160

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

It creates a long display rather than a wide display in this instance.

◁

▷ Example 3: The generate( ) option

The option `generate()` can be used to create a variable containing the categories of choice-set patterns. Here we use it after running `cmclogit`.

```
. cmclogit purchase dealers, casevar(i.gender income)
  (output omitted)
. cmchoiceset if e(sample) == 1, generate(choiceset)
Tabulation of choice-set possibilities
```

Choice set	Freq.	Percent	Cum.
1 2 3	373	43.27	43.27
1 2 3 4	489	56.73	100.00
Total	862	100.00	

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

Variable name	Storage type	Display format	Value label	Variable label
choiceset	byte	%8.0g	choiceset	Choice set

```
. label list choiceset
choiceset:
           1 1 2 3
           2 1 2 3 4
```

Note that we specified `if e(sample) == 1` with `cmchoiceset` so that the sample used for `cmchoiceset` is the same as the estimation sample used by `cmclogit`.

generate() creates a variable with values 1 and 2. Its value label contains the strings "1 2 3" and "1 2 3 4", which make the output understandable.

If we want to look at average predicted probabilities for the alternatives separately for the two different choice sets, we can use the newly created variable choiceset with the over() option in margins.

```
. margins, over(choiceset)
Predictive margins                                      Number of obs = 3,075
Model VCE: OIM

Expression: Pr(car|1 selected), predict()
Over:       choiceset
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
_outcome# choiceset						
American # 1 2 3	.4610168	.0172363	26.75	0.000	.4272343	.4947992
American # 1 2 3 4	.4172612	.0169068	24.68	0.000	.3841246	.4503979
Japanese # 1 2 3	.3840219	.0168052	22.85	0.000	.3510843	.4169596
Japanese # 1 2 3 4	.3532921	.0161223	21.91	0.000	.3216929	.3848913
European # 1 2 3	.1549613	.0123628	12.53	0.000	.1307307	.1791919
European # 1 2 3 4	.1476471	.0117985	12.51	0.000	.1245225	.1707718
Korean # 1 2 3	.	(not estimable)				
Korean # 1 2 3 4	.0817996	.0122163	6.70	0.000	.0578562	.105743

In particular, looking at predicted probabilities and marginal effects by choice sets is often useful for intentionally unbalanced designs. See [CM] **margins** for a more lengthy discussion.

◁

▷ Example 4: Panel choice data

When you have panel choice data, cmchoiceset is useful to see how choice sets vary by time—if they do vary by time. Here is an example with an unbalanced dataset.

```
. use https://www.stata-press.com/data/r19/transport_unbalanced, clear
(Transportation choice data with unbalanced choice sets)

. cmset id t alt
note: case identifier _caseid generated from id and t.
note: panel by alternatives identifier _panelaltid generated from id and alt.
note: alternatives are unbalanced across choice sets; choice sets of
      different sizes found.

                Panel data: Panels id and time t
          Case ID variable: _caseid
     Alternatives variable: alt
Panel by alternatives variable: _panelaltid (unbalanced)
              Time variable: t, 1 to 3, but with gaps
                      Delta: 1 unit

Note: Data have been xtset.
```

The output from cmset is telling us the data are unbalanced. Do the choice sets vary by time? cmchoiceset with the option time will answer this question.

```
. cmchoiceset, time
Tabulation of choice-set possibilities by time t
```

Choice set	Time variable 1	2	3	Total
1 2	0	1	0	1
1 2 3	5	3	0	8
1 2 3 4	483	482	500	1,465
1 2 4	6	3	0	9
1 3 4	2	3	0	5
2 3 4	4	8	0	12
Total	500	500	500	1,500

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

The choice sets at time t = 3 are balanced but are unbalanced at each of the other times.

If there were many time periods and only a few choice sets, the option transpose would make a more readable tabulation.

◁

Stored results

cmchoiceset stores the following in r():

Scalars

r(N)	number of observations
r(r)	number of rows
r(c)	number of columns

Also see

[CM] **cmsample** — Display reasons for sample exclusion

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

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

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

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on citing Stata documentation.