Title

proportion — Estimate proportions

Description

`proportion` produces estimates of proportions, along with standard errors, for the categories identified by the values in each variable of *varlist*.

Quick start

Proportions, standard errors, and 95% CIs for each level of `v1`

```
proportion v1
```

Also compute statistics for `v2`

```
proportion v1 v2
```

Treat missing values of `v1` as a valid category

```
proportion v1, missing
```

As above, for each subpopulation defined by the levels of `catvar`

```
proportion v1, missing over(catvar)
```

Standardizing across strata defined by `svar` with stratum weight `wvar1`

```
proportion v1, stdize(svar) stdweight(wvar1)
```

Weighting by sampling weight `wvar2`

```
proportion v1 [pweight=wvar2]
```

Menu

Statistics > Summaries, tables, and tests > Summary and descriptive statistics > Proportions

Syntax

`proportion` *varlist* [*if*] [*in*] [*weight*] [, *options*]

options	Description
Model	
`stdize(`*varname*`)`	variable identifying strata for standardization
`stdweight(`*varname*`)`	weight variable for standardization
`nostdrescale`	do not rescale the standard weight variable
`nolabel`	suppress value labels from *varlist*
`missing`	treat missing values like other values
if/in/over	
`over(`*varlist* [, `nolabel`]`)`	group over subpopulations defined by *varlist*; optionally, suppress group labels
SE/Cluster	
`vce(`*vcetype*`)`	*vcetype* may be `analytic`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
Reporting	
`level(`*#*`)`	set confidence level; default is `level(95)`
`citype(logit` | `normal)`	method to compute limits of confidence intervals; default is `citype(logit)`
`noheader`	suppress table header
`nolegend`	suppress table legend
display_options	control column formats and line width
`coeflegend`	display legend instead of statistics

`bootstrap`, `jackknife`, `mi estimate`, `rolling`, `statsby`, and `svy` are allowed; see **[U] 11.1.10 Prefix commands**.
`vce(bootstrap)` and `vce(jackknife)` are not allowed with the `mi estimate` prefix; see **[MI] mi estimate**.
Weights are not allowed with the `bootstrap` prefix; see **[R] bootstrap**.
`vce()` and weights are not allowed with the `svy` prefix; see **[SVY] svy**.
`fweight`s, `iweight`s, and `pweight`s are allowed; see **[U] 11.1.6 weight**.
`coeflegend` does not appear in the dialog box.
See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

`stdize(`*varname*`)` specifies that the point estimates be adjusted by direct standardization across the strata identified by *varname*. This option requires the `stdweight()` option.

`stdweight(`*varname*`)` specifies the weight variable associated with the standard strata identified in the `stdize()` option. The standardization weights must be constant within the standard strata.

`nostdrescale` prevents the standardization weights from being rescaled within the `over()` groups. This option requires `stdize()` but is ignored if the `over()` option is not specified.

`nolabel` specifies that value labels attached to the variables in *varlist* be ignored.

`missing` specifies that missing values in *varlist* be treated as valid categories, rather than omitted from the analysis (the default).

if/in/over

`over(`*varlist* [`, nolabel`]`)` specifies that estimates be computed for multiple subpopulations, which are identified by the different values of the variables in *varlist*.

When this option is supplied with one variable name, such as `over(`*varname*`)`, the value labels of *varname* are used to identify the subpopulations. If *varname* does not have labeled values (or there are unlabeled values), the values themselves are used, provided that they are nonnegative integers. Noninteger values, negative values, and labels that are not valid Stata names are substituted with a default identifier.

When `over()` is supplied with multiple variable names, each subpopulation is assigned a unique default identifier.

`nolabel` requests that value labels attached to the variables identifying the subpopulations be ignored.

SE/Cluster

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`analytic`), that allow for intragroup correlation (`cluster` *clustvar*), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] ***vce_option***.

`vce(analytic)`, the default, uses the analytically derived variance estimator associated with the sample proportion.

Reporting

`level(`*#*`)`; see [R] **estimation options**.

`citype(logit` | `normal)` specifies how to compute the limits of confidence intervals.

`citype(logit)`, the default, uses the logit transformation to compute the limits of confidence intervals.

`citype(normal)` uses the normal approximation to compute the limits of confidence intervals.

`noheader` prevents the table header from being displayed. This option implies `nolegend`.

`nolegend` prevents the table legend identifying the subpopulations from being displayed.

display_options: `cformat(`*%fmt*`)` and `nolstretch`; see [R] **estimation options**.

The following option is available with `proportion` but is not shown in the dialog box:

`coeflegend`; see [R] **estimation options**.

Remarks and examples

▷ Example 1

We can estimate the proportion of each repair rating in auto2.dta:

```
. use http://www.stata-press.com/data/r14/auto2
(1978 Automobile Data)

. proportion rep78

Proportion estimation               Number of obs     =         69
```

	Proportion	Std. Err.	[95% Conf.	Interval]
rep78				
Poor	.0289855	.0203446	.0070061	.1121326
Fair	.115942	.0388245	.0580159	.2183014
Average	.4347826	.0601159	.3207109	.556206
Good	.2608696	.0532498	.1690271	.3798066
Excellent	.1594203	.0443922	.089188	.2686455

Here we use the missing option to include missing values as a category of rep78:

```
. proportion rep78, missing

Proportion estimation               Number of obs     =         74

        _prop_6: rep78 = .
```

	Proportion	Std. Err.	[95% Conf.	Interval]
rep78				
Poor	.027027	.0189796	.0065484	.1047932
Fair	.1081081	.0363433	.054094	.204402
Average	.4054054	.0574637	.2977369	.523012
Good	.2432432	.0502154	.1572724	.3563376
Excellent	.1486486	.0416364	.0831005	.2517065
_prop_6	.0675676	.0293776	.0278144	.1550743

◁

▷ Example 2

We can also estimate proportions over groups:

```
. proportion rep78, over(foreign)
Proportion estimation                Number of obs   =         69

         Poor: rep78 = Poor
         Fair: rep78 = Fair
      Average: rep78 = Average
         Good: rep78 = Good
    Excellent: rep78 = Excellent

     Domestic: foreign = Domestic
      Foreign: foreign = Foreign
```

Over	Proportion	Std. Err.	[95% Conf.	Interval]
Poor				
Domestic	.0416667	.0291477	.0100299	.1572433
Foreign	0	(no observations)		
Fair				
Domestic	.1666667	.0543607	.0839032	.3039797
Foreign	0	(no observations)		
Average				
Domestic	.5625	.0723605	.4169211	.6980553
Foreign	.1428571	.0782461	.0444941	.3736393
Good				
Domestic	.1875	.0569329	.0986718	.3272601
Foreign	.4285714	.1106567	.2333786	.6488451
Excellent				
Domestic	.0416667	.0291477	.0100299	.1572433
Foreign	.4285714	.1106567	.2333786	.6488451

◁

Stored results

proportion stores the following in e():

Scalars

e(N)	number of observations
e(N_over)	number of subpopulations
e(N_stdize)	number of standard strata
e(N_clust)	number of clusters
e(k_eq)	number of equations in e(b)
e(df_r)	sample degrees of freedom
e(rank)	rank of e(V)

Macros

e(cmd)	proportion
e(cmdline)	command as typed
e(varlist)	*varlist*
e(stdize)	*varname* from stdize()
e(stdweight)	*varname* from stdweight()
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(cluster)	name of cluster variable
e(over)	*varlist* from over()
e(over_labels)	labels from over() variables
e(over_namelist)	names from e(over_labels)
e(namelist)	proportion identifiers
e(label#)	labels from #th variable in *varlist*
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. Err.
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(marginsnotok)	predictions disallowed by margins

Matrices

e(b)	vector of proportion estimates
e(V)	(co)variance estimates
e(_N)	vector of numbers of nonmissing observations
e(_N_stdsum)	number of nonmissing observations within the standard strata
e(_p_stdize)	standardizing proportions
e(error)	error code corresponding to e(b)

Functions

e(sample)	marks estimation sample

Methods and formulas

Proportions are means of indicator variables; see [R] **mean**.

Confidence intervals

Confidence intervals for proportions are calculated using a logit transform so that the endpoints lie between 0 and 1. Let $\widehat{p}$ be an estimated proportion and $\widehat{s}$ be an estimate of its standard error. Let

$$f(\widehat{p}) = \ln\left(\frac{\widehat{p}}{1-\widehat{p}}\right)$$

be the logit transform of the proportion. In this metric, an estimate of the standard error is

$$\widehat{\mathrm{SE}}\{f(\widehat{p})\} = f'(\widehat{p})\widehat{s} = \frac{\widehat{s}}{\widehat{p}(1-\widehat{p})}$$

Thus a $100(1-\alpha)\%$ confidence interval in this metric is

$$\ln\left(\frac{\widehat{p}}{1-\widehat{p}}\right) \pm \frac{t_{1-\alpha/2,\nu}\,\widehat{s}}{\widehat{p}(1-\widehat{p})}$$

where $t_{1-\alpha/2,\nu}$ is the $(1-\alpha/2)$th quantile of Student's t distribution with ν degrees of freedom. The endpoints of this confidence interval are transformed back to the proportion metric by using the inverse of the logit transform

$$f^{-1}(y) = \frac{e^y}{1+e^y}$$

Hence, the displayed confidence intervals for proportions are

$$f^{-1}\left\{\ln\left(\frac{\widehat{p}}{1-\widehat{p}}\right) \pm \frac{t_{1-\alpha/2,\nu}\,\widehat{s}}{\widehat{p}(1-\widehat{p})}\right\}$$

References

Cochran, W. G. 1977. *Sampling Techniques*. 3rd ed. New York: Wiley.

Stuart, A., and J. K. Ord. 1994. *Kendall's Advanced Theory of Statistics: Distribution Theory, Vol I*. 6th ed. London: Arnold.

Also see

[R] **proportion postestimation** — Postestimation tools for proportion

[R] **mean** — Estimate means

[R] **ratio** — Estimate ratios

[R] **total** — Estimate totals

[MI] **estimation** — Estimation commands for use with mi estimate

[SVY] **direct standardization** — Direct standardization of means, proportions, and ratios

[SVY] **poststratification** — Poststratification for survey data

[SVY] **subpopulation estimation** — Subpopulation estimation for survey data

[SVY] **svy estimation** — Estimation commands for survey data

[SVY] **variance estimation** — Variance estimation for survey data

[U] **20 Estimation and postestimation commands**