Title

stata.com

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
```

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

```
proportion v1 v2, 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
if/in/over	
over(*varlist*$_o$)	group over subpopulations defined by *varlist*$_o$
SE/Cluster	
vce(*vcetype*)	*vcetype* may be analytic, cluster *clustvar*, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
citype(*citype*)	method to compute limits of confidence intervals; default is citype(logit)
percent	report estimated proportions as percentages
noheader	suppress table header
display_options	control column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
coeflegend	display legend instead of statistics

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

Only numeric, nonnegative, integer-valued variables are allowed in *varlist*.

bootstrap, collect, 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**.

fweights, iweights, and pweights 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.

citype	Description
logit	calculate logit-transformed confidence intervals; the default
agresti	calculate Agresti–Coull confidence intervals
exact	calculate exact (Clopper–Pearson) confidence intervals
jeffreys	calculate Jeffreys confidence intervals
normal	calculate normal (Wald) confidence intervals
wald	synonym for normal
wilson	calculate Wilson confidence intervals

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.

if/in/over

`over(`$varlist_o$`)` specifies that estimates be computed for multiple subpopulations, which are identified by the different values of the variables in $varlist_o$. Only numeric, nonnegative, integer-valued variables are allowed in `over(`$varlist_o$`)`.

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(`*citype*`)` specifies how to compute the limits of confidence intervals. *citype* may be one of `logit` (default), `agresti`, `exact`, `jeffreys`, `normal`, `wald`, or `wilson`.

`percent` specifies that the proportions be reported as percentages.

`noheader` prevents the table header from being displayed.

display_options: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `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

stata.com

Example 1

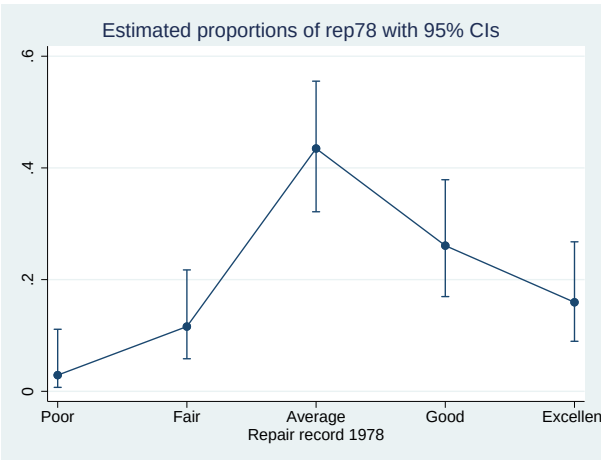
We can estimate the proportion of each repair rating in auto2.dta:

```
. use https://www.stata-press.com/data/r17/auto2
(1978 automobile data)
. proportion rep78
Proportion estimation                           Number of obs = 69
```

	Proportion	Std. err.	Logit [95% conf.	interval]
rep78				
Poor	.0289855	.0201966	.0070794	.1110924
Fair	.115942	.0385422	.058317	.2173648
Average	.4347826	.0596787	.3214848	.5553295
Good	.2608696	.0528625	.1695907	.3788629
Excellent	.1594203	.0440694	.0895793	.267702

marginsplot will produce a graph of the results from proportion:

```
. marginsplot
Variables that uniquely identify proportions: rep78
```



▷ Example 2

We can also estimate proportions over groups:

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

	Proportion	Std. err.	Logit [95% conf.	interval]
rep78@foreign				
Poor Domestic	.0416667	.0288424	.0101825	.1552326
Poor Foreign	0	(no observations)		
Fair Domestic	.1666667	.0537914	.084534	.3022522
Fair Foreign	0	(no observations)		
Average Domestic	.5625	.0716027	.4184154	.6967587
Average Foreign	.1428571	.0763604	.0458191	.3664757
Good Domestic	.1875	.0563367	.0993684	.3255432
Good Foreign	.4285714	.1079898	.2372889	.6438783
Excellent Domestic	.0416667	.0288424	.0101825	.1552326
Excellent Foreign	.4285714	.1079898	.2372889	.6438783

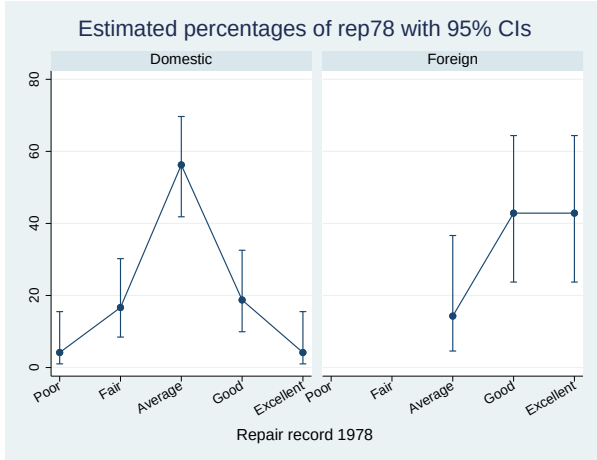
To see the results as percentages instead of proportions, we add the `percent` option:

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

	Percent	Std. err.	Logit [95% conf.	interval]
rep78@foreign				
Poor Domestic	4.17	2.88	1.02	15.52
Poor Foreign	0.00	(no observations)		
Fair Domestic	16.67	5.38	8.45	30.23
Fair Foreign	0.00	(no observations)		
Average Domestic	56.25	7.16	41.84	69.68
Average Foreign	14.29	7.64	4.58	36.65
Good Domestic	18.75	5.63	9.94	32.55
Good Foreign	42.86	10.80	23.73	64.39
Excellent Domestic	4.17	2.88	1.02	15.52
Excellent Foreign	42.86	10.80	23.73	64.39

We can now use `marginsplot` to graph the percentages for each group. We add the `bydimension(foreign)` option to plot the groups in separate graph panels. The `xlabel(, angle(30))` option prevents the x-axis labels from running into each other.

```
. marginsplot, bydimension(foreign) xlabel(, angle(30))
Variables that uniquely identify proportions: rep78 foreign
```



We estimate that only 19% of domestic cars have good repair records and only 4% have excellent repair records. For foreign cars, however, we find that 43% have good repair records and 43% have excellent repair records.

◁

▷ Example 3

Instead of estimating percentages within the foreign and domestic groupings, we might want to know overall percentages. For instance, what percentage of all cars are foreign and have excellent repair records? What percentage are domestic and have average records? We can obtain all such percentages by specifying an interaction between `rep78` and `foreign`.

```
. proportion rep78#foreign, percent
Percent estimation                                  Number of obs = 69
```

	Percent	Std. err.	Logit [95% conf.	interval]
rep78#foreign				
Poor#Domestic	2.90	2.02	0.71	11.11
Poor#Foreign	0.00	(no observations)		
Fair#Domestic	11.59	3.85	5.83	21.74
Fair#Foreign	0.00	(no observations)		
Average#Domestic	39.13	5.88	28.21	51.26
Average#Foreign	4.35	2.46	1.38	12.86
Good#Domestic	13.04	4.05	6.85	23.44
Good#Foreign	13.04	4.05	6.85	23.44
Excellent#Domestic	2.90	2.02	0.71	11.11
Excellent#Foreign	13.04	4.05	6.85	23.44

Looking at the last line of this output, we estimate that 13% of all cars are foreign with excellent repair records.

◁

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(clustvar)	name of cluster variable
e(over)	*varlist* from over()
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(freq)	vector of frequency estimates
e(error)	error code corresponding to e(b)

Functions

e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices

r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

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

Confidence intervals

For an overview of confidence interval methods for binomial proportions, see Dean and Pagano (2015).

Given k successes of n trials, the estimated proportion (probability of a success) is $\widehat{p} = k/n$ with estimated standard error $\widehat{s} = \sqrt{\widehat{p}(1-\widehat{p})/n}$.

The logit-transformed confidence interval is given by

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

where $t_{p,\nu}$ is the pth 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

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

The Wald-type $100(1-\alpha)\%$ confidence interval is given by

$$\widehat{p} \pm t_{1-\alpha/2,\nu}\,\widehat{s}$$

The Wilson interval is given by

$$\frac{\widehat{p} + z^2_{1-\alpha/2}/2n \pm z_{1-\alpha/2}\sqrt{\widehat{s} + z^2_{1-\alpha/2}/4n^2}}{1 + z^2_{1-\alpha/2}/n}$$

where z_p is the pth quantile of the standard normal distribution.

The exact (Clopper–Pearson) interval is given by

$$\left\{\widehat{p} - \frac{\nu_1 F_{\alpha/2,\nu_1,\nu_2}}{\nu_2 + \nu_1 F_{\alpha/2,\nu_1,\nu_2}};\ \ \widehat{p} + \frac{\nu_3 F_{\alpha/2,\nu_3,\nu_4}}{\nu_4 + \nu_3 F_{\alpha/2,\nu_3,\nu_4}}\right\}$$

where $\nu_1 = 2k$, $\nu_2 = 2(n-k+1)$, $\nu_3 = 2(k+1)$, $\nu_4 = 2(n-k)$, and F_{p,ν_1,ν_2} is the pth quantile of an F distribution with ν_1 and ν_2 degrees of freedom.

The Jeffreys interval is given by

$$\{\widehat{p} - \text{Beta}_{\alpha/2,\alpha_1,\beta_2};\ \ \widehat{p} + \text{Beta}_{1-\alpha/2,\alpha_1,\beta_2}\}$$

where $\alpha_1 = k + 0.5$, $\beta_1 = n - k + 0.5$, and $\text{Beta}_{p,\alpha_1,\beta_2}$ is the pth quantile of a Beta distribution with α_1 and β_1 degrees of freedom.

The Agresti–Coull interval is given by

$$\widetilde{p} \pm z_{1-\alpha/2}\sqrt{\widetilde{p}(1-\widetilde{p})/\widetilde{n}}$$

where $\widetilde{k} = k + z^2_{1-\alpha/2}/2$, $\widetilde{n} = n + z^2_{1-\alpha/2}$, and $\widetilde{p} = \widetilde{k}/\widetilde{n}$.

When degrees of freedom ν are posted to `e(df_r)`, the Wilson, exact, Jeffreys, and Agresti–Coull intervals use n^* in place of n, where

$$n^* = \frac{\widehat{p}(1-\widehat{p})}{\widehat{s}^2}\left\{\frac{z_{1-\alpha/2}}{t_{1-\alpha/2,\nu}}\right\}^2$$

References

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

Dean, N., and M. Pagano. 2015. Evaluating confidence interval methods for binomial proportions in clustered surveys. *Journal of Survey Statistics and Methodology* 3: 484–503. https://doi.org/10.1093/jssam/smv024.

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**