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

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

coefflegend  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, 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

logit  calculate logit-transformed confidence intervals; the default
agresti  calculate Agresti–Coul 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

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.

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).

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(citytype) specifies how to compute the limits of confidence intervals. citytype 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

Example 1

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

```
. use https://www.stata-press.com/data/r16/auto2
   (1978 Automobile Data)
. proportion rep78
```

Proportion estimation

<table>
<thead>
<tr>
<th></th>
<th>Proportion</th>
<th>Std. Err.</th>
<th>[95% Conf. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poor</td>
<td>.0289855</td>
<td>.0201966</td>
<td>.0070794 .1110924</td>
</tr>
<tr>
<td>Fair</td>
<td>.115942</td>
<td>.0385422</td>
<td>.058317 .2173648</td>
</tr>
<tr>
<td>Average</td>
<td>.4347826</td>
<td>.0596787</td>
<td>.3214848 .5553295</td>
</tr>
<tr>
<td>Good</td>
<td>.2608696</td>
<td>.0528625</td>
<td>.1695907 .3788629</td>
</tr>
<tr>
<td>Excellent</td>
<td>.1594203</td>
<td>.0440694</td>
<td>.0895793 .267702</td>
</tr>
</tbody>
</table>

`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)
```

<table>
<thead>
<tr>
<th>Proportion estimation</th>
<th>Number of obs = 69</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Logit</td>
</tr>
<tr>
<td></td>
<td>Proportion</td>
</tr>
<tr>
<td>rep78@foreign</td>
<td></td>
</tr>
<tr>
<td>Poor Domestic</td>
<td>0.0416667</td>
</tr>
<tr>
<td>Poor Foreign</td>
<td>0 (no observations)</td>
</tr>
<tr>
<td>Fair Domestic</td>
<td>0.1666667</td>
</tr>
<tr>
<td>Fair Foreign</td>
<td>0 (no observations)</td>
</tr>
<tr>
<td>Average Domestic</td>
<td>0.5625</td>
</tr>
<tr>
<td>Average Foreign</td>
<td>0.1428571</td>
</tr>
<tr>
<td>Good Domestic</td>
<td>0.1875</td>
</tr>
<tr>
<td>Good Foreign</td>
<td>0.4285714</td>
</tr>
<tr>
<td>Excellent Domestic</td>
<td>0.0416667</td>
</tr>
<tr>
<td>Excellent Foreign</td>
<td>0.4285714</td>
</tr>
</tbody>
</table>

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

```
. proportion rep78, over(foreign) percent
```

<table>
<thead>
<tr>
<th>Percent estimation</th>
<th>Number of obs = 69</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Logit</td>
</tr>
<tr>
<td></td>
<td>Percent</td>
</tr>
<tr>
<td>rep78@foreign</td>
<td></td>
</tr>
<tr>
<td>Poor Domestic</td>
<td>4.17</td>
</tr>
<tr>
<td>Poor Foreign</td>
<td>0.00 (no observations)</td>
</tr>
<tr>
<td>Fair Domestic</td>
<td>16.67</td>
</tr>
<tr>
<td>Fair Foreign</td>
<td>0.00 (no observations)</td>
</tr>
<tr>
<td>Average Domestic</td>
<td>56.25</td>
</tr>
<tr>
<td>Average Foreign</td>
<td>14.29</td>
</tr>
<tr>
<td>Good Domestic</td>
<td>18.75</td>
</tr>
<tr>
<td>Good Foreign</td>
<td>42.86</td>
</tr>
<tr>
<td>Excellent Domestic</td>
<td>4.17</td>
</tr>
<tr>
<td>Excellent Foreign</td>
<td>42.86</td>
</tr>
</tbody>
</table>

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.
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

<table>
<thead>
<tr>
<th></th>
<th>Percent</th>
<th>Std. Err.</th>
<th>[95% Conf. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>rep78#foreign</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Poor#Domestic</td>
<td>2.90</td>
<td>2.02</td>
<td>0.71    11.11</td>
</tr>
<tr>
<td>Poor#Foreign</td>
<td>0.00</td>
<td>(no observations)</td>
<td></td>
</tr>
<tr>
<td>Fair#Domestic</td>
<td>11.59</td>
<td>3.85</td>
<td>5.83    21.74</td>
</tr>
<tr>
<td>Fair#Foreign</td>
<td>0.00</td>
<td>(no observations)</td>
<td></td>
</tr>
<tr>
<td>Average#Domestic</td>
<td>39.13</td>
<td>5.88</td>
<td>28.21   51.26</td>
</tr>
<tr>
<td>Average#Foreign</td>
<td>4.35</td>
<td>2.46</td>
<td>1.38    12.86</td>
</tr>
<tr>
<td>Good#Domestic</td>
<td>13.04</td>
<td>4.05</td>
<td>6.85    23.44</td>
</tr>
<tr>
<td>Good#Foreign</td>
<td>13.04</td>
<td>4.05</td>
<td>6.85    23.44</td>
</tr>
<tr>
<td>Excellent#Domestic</td>
<td>2.90</td>
<td>2.02</td>
<td>0.71    11.11</td>
</tr>
<tr>
<td>Excellent#Foreign</td>
<td>13.04</td>
<td>4.05</td>
<td>6.85    23.44</td>
</tr>
</tbody>
</table>
```

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(freq)` vector of frequency estimates
- `e(pstdize)` standardizing proportions
- `e(error)` error code corresponding to `e(b)`

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

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 \( \hat{p} = k/n \) with estimated standard error \( \hat{s} = \sqrt{\hat{p}(1 - \hat{p})}/n \).

The logit-transformed confidence interval is given by

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

where \( t_{p,\nu} \) is the \( p \)th quantile of Student’s \( t \) distribution with \( \nu \) 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
\[ f^{-1}\left\{ \ln\left( \frac{\hat{p}}{1 - \hat{p}} \right) \pm t_{1-\alpha/2,\nu} \frac{\hat{s}}{\hat{p}(1 - \hat{p})} \right\} \]
The Wald-type 100(1 – α)% confidence interval is given by
\[ \hat{p} \pm t_{1-\alpha/2,\nu} \hat{s} \]
The Wilson interval is given by
\[ \frac{\hat{p} + z^2_{1-\alpha/2}/2n \pm z_{1-\alpha/2}\sqrt{\hat{s} + z^2_{1-\alpha/2}/4n^2}}{1 + z^2_{1-\alpha/2}/n} \]
where \( z_p \) is the \( p \)th quantile of the standard normal distribution.

The exact (Clopper–Pearson) interval is given by
\[ \left\{ \hat{p} - \frac{\nu_1 F_{\alpha/2,\nu_1,\nu_2}}{\nu_2 + \nu_1 F_{\alpha/2,\nu_1,\nu_2}}; \hat{p} + \frac{\nu_3 F_{\alpha/2,\nu_3,\nu_4}}{\nu_4 + \nu_3 F_{\alpha/2,\nu_3,\nu_4}} \right\} \]
where \( \nu_1 = 2k, \nu_2 = 2(n - k + 1), \nu_3 = 2(k + 1), \nu_4 = 2(n - k) \), and \( F_{p,\nu_1,\nu_2} \) is the \( p \)th quantile of an \( F \) distribution with \( \nu_1 \) and \( \nu_2 \) degrees of freedom.

The Jeffreys interval is given by
\[ \left\{ \hat{p} - \text{Beta}_{\alpha/2,\alpha_1,\beta_2}; \hat{p} + \text{Beta}_{1-\alpha/2,\alpha_1,\beta_2} \right\} \]
where \( \alpha_1 = k + 0.5, \beta_1 = n - k + 0.5 \), and \( \text{Beta}_{p,\alpha_1,\beta_2} \) is the \( p \)th quantile of a Beta distribution with \( \alpha_1 \) and \( \beta_1 \) degrees of freedom.

The Agresti–Coull interval is given by
\[ \hat{p} \pm z_{1-\alpha/2}\sqrt{\hat{p}(1 - \hat{p})/\tilde{n}} \]
where \( \tilde{k} = k + z^2_{1-\alpha/2}/2, \tilde{n} = n + z^2_{1-\alpha/2}, \) and \( \tilde{p} = \tilde{k}/\tilde{n}. \)

When degrees of freedom \( \nu \) are posted to \( e(df_\nu) \), the Wilson, exact, Jeffreys, and Agresti–Coull intervals use \( n^* \) in place of \( n \), where
\[ n^* = \left( \frac{\hat{p}(1 - \hat{p})}{\hat{s}^2} \right) \left\{ \frac{z_{1-\alpha/2}}{t_{1-\alpha/2,\nu}} \right\}^2 \]
References


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**